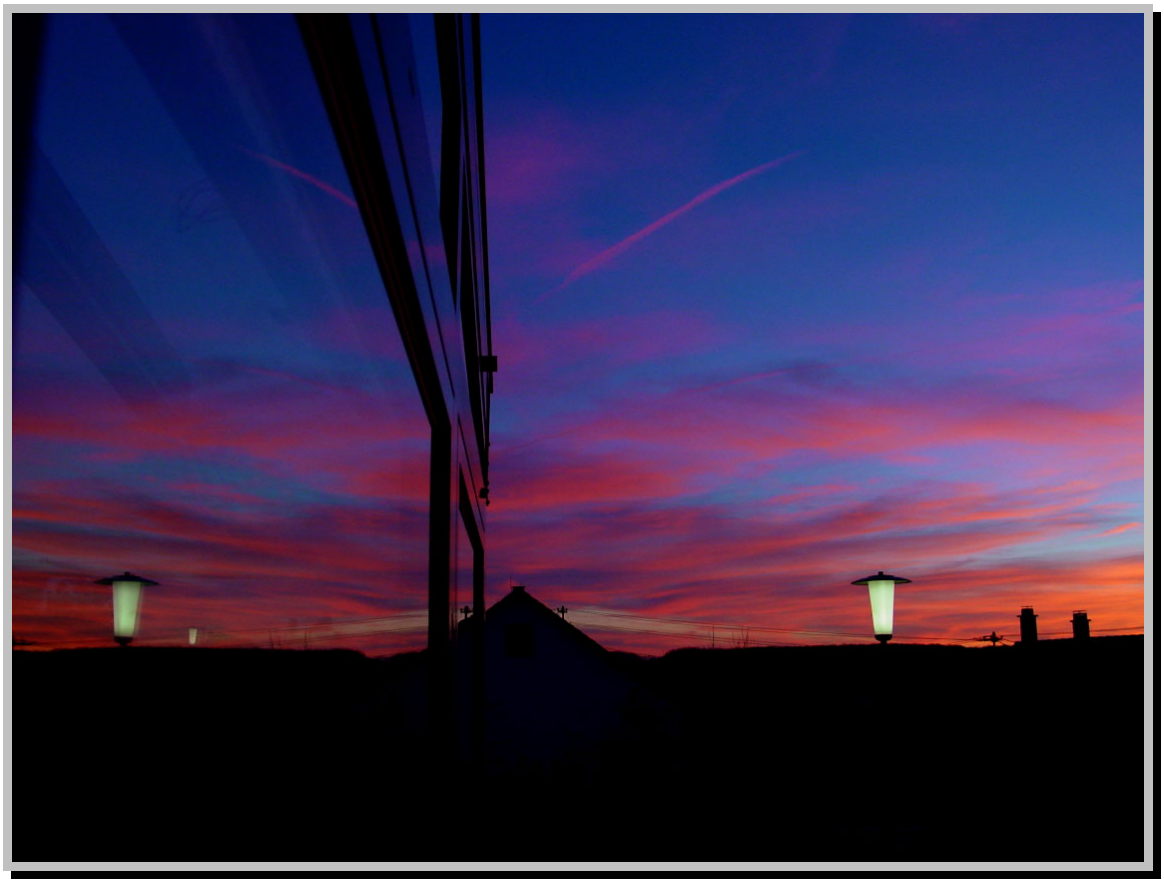
ConvCtrl

(conservatory control)

Version 1.0



Mikrokontroller gestützte Wintergarten-Steuerung für Licht-, Umwelt- und Beschattungsfunktionen

Lonsee im Dezember 2008

Markus Fulde
Finkenweg 3
D-89173 Lonsee
Telefon +49 (7336) 92 11 89
Fax +49 (7336) 92 10 68
Mobil +49 (160) 84 54 314

Email Markus.Fulde@t-online.de

© Markus Fulde, 2007

1 Sicherheitshinweis



Achtung:
Aufgrund der im Gerät frei geführten Netzspannung dürfen Aufbau und Inbetriebnahme ausschließlich von Fachkräften durchgeführt werden, die aufgrund ihrer Ausbildung dazu befugt sind. Die einschlägigen Sicherheits- und VDE-Bestimmungen sind unbedingt zu beachten.

Der Autor übernimmt keinerlei Haftung über Schaden jeglicher Art, die direkt oder indirekt durch das Gerät verursacht werden.



2 Inhaltsverzeichnis

1 Sicherheitshinweis 3
2 Inhaltsverzeichnis 4
3 Abbildungsverzeichnis 10
4 Tabellenverzeichnis 15
5 Schaltbilderverzeichnis 19
6 Softwareverzeichnis 21
7 Allgemeines 22
7.1 Leistungsumfang 22
7.2 Funktionskomponenten 23
7.3 Festlegungen und Entscheidungen 23
7.4 Gesamt-Checkliste für Funktionen, Eigenschaften und Features 24
7.4.1 Allgemeines zur HW-Umsetzung 24
7.4.2 Spannungsversorgung für +5V und +12V 25
7.4.3 Mikrocontroller ATmega32 oder ATmega128 25
7.4.4 LCD-Display 2x16 oder 3x16 25
7.4.5 Tastatur und Schalter 26
7.4.6 LED-Anzeigen für Status- und Funktionsanzeige 27
7.4.7 Windmesser 28
7.4.8 Regensensor 28
7.4.9 Temperatursensor 29
7.4.10 Luftfeuchtigkeit 29
7.4.11 Dimmungssteuerung für Beleuchtung 29
7.4.12 Motorsteuerung für Beschattung 30
7.4.13 Gehäusebelüftung mit Option Lüftersteuerung über ATmega 30
7.4.14 Datenspeicher / Datenlogger über EEPROM 31
7.4.15 DCF77 Funkuhrempfang (Conrad) 31
7.4.16 RTC – Real-Time-Clock 31
7.4.17 Passendes Selbstbaugehäuse aus Aluminium und rauchgläserner Front 32
7.4.18 Helligkeitsmessung 32
7.4.19 Anzeige von Zeit- und Datumsinformationen 32
7.4.20 Akustisches Feedback über Piezo Schallwandler 32
7.4.21 Selbstcheck bei StartUp und Warnmeldungen 33
7.4.22 Timersteuerung zeitkritischer Applikationen 33
7.5 Projektcheckliste für AVR Systemdesigns 33
7.5.1 Abblockkondensator(en) ordnungsgemäß installiert? 33
7.5.2 Spannungsversorgung richtig angeschlossen? 34
7.5.3 Reset-Pin korrekt beschaltet? 34
7.5.4 Alle Ground-Anschlüsse beschaltet? 34
7.5.5 Besonderheiten bei ATmega64 / ATmega128 34
7.6 Bedienelemente 35
7.6.1 Tasten 35
7.6.2 LCD-Display 36
7.7 Menüstruktur 37
8 Grundlagen 43
8.1 Mikrokontroller ATmega32 oder ATmega128 43
8.1.1 Ressourcenzuordnung ATmega32 zu Projekt ConvCtrl 44
8.1.2 Ressourcenzuordnung ATmega128 zu Projekt ConvCtrl 46
8.1.3 Interrupt-Vektor-Tabelle ATmega128 48

8.1.4 Basisbeschaltung des ATmega32 50
8.1.5 ISP-Schnittstelle / ISP-Programmierung 52
8.1.6 Ponyprog Programmieradapter 56
8.1.7 Schnittstellen und Adapter für ConvCtrl 63
8.1.8 Das JTEG Interface 65
8.1.9 PIN-Belegung des PC Parallel-Port 67
8.1.10 RS232-Interface 69
8.1.11 PIN-Belegung 9-polige RS232 71
8.1.12 Verbindungskabel RS232 PIN-Header / D-SUB9 71
8.1.13 Gesamt-Basisbeschaltung ATmega128 72
8.1.14 AVR Fusebits Tutorial 74
8.1.15 AVR Fuse Konfiguration ATmega32 80
8.1.16 AVR Fuse Konfiguration ATmega128 81
8.1.17 Die Fuse-Konfiguration von ConvCtrl 84
8.2 Spannungsversorgung 85
8.2.1 Hauptschalter 85
8.2.2 LED Vorschaltmodul für Hauptschalter 86
8.2.3 Abschätzung Stromverbrauch 89
8.2.4 Spannungen 5V und 12V 90
8.3 Netzteil mit Festspannungsregler 91
8.3.1 Schaltplan Netzteil 92
8.3.2 Zusammenfassung nach Inbetriebnahme 93
8.4 Netzteil mit Schaltregler 94
8.4.1 Schaltplan Netzteil 95
8.5 Steuerung Beschattungsanlage Motorantriebe 97
8.5.1 Absicherung Gegenläufigkeit der Motor-Relais 99
8.5.2 Relaisansteuerung 100
8.5.3 BASCOM Beispielcode für Beschattungsanlage 101
8.5.4 Prototyp für Beschattungsanlage 101
8.6 Steuerung Beleuchtungsanlage 101
8.6.1 BASCOM Beispielcode für Beleuchtungsanlage 102
8.6.2 Prototyp für Beleuchtungsanlage 102
8.7 Gesamtschaltplan für Relaisansteuerung 102
8.8 Lichtdimmung 105
8.8.1 Funktionsprinzip Phasenanschnittsteuerung 105
8.8.2 Phasenanschnitt- und Phasenabschnittsteuerung 106
8.8.3 Vor- und Nachteile der Phasenanschnittsteuerung 107
8.8.4 HW-Konzept für die Umsetzung im ConvCtrl 108
8.8.5 SW-Konzept für Umsetzung 112
8.8.6 Linearisierung der Helligkeitswerte 112
8.8.7 Detailbeschreibung SW 117
8.8.8 BASCOM Beispielcode für Relaisansteuerung und Dimmer 117
8.8.9 Prototyp Relaisansteuerung und Dimmer 127
8.9 LCD-Display 128
8.9.1 Beschaltung des LCD-Display 131
8.9.2 Zeichensatz und Befehlstabellen 133
8.9.3 BASCOM Beispielcode für die Displayansteuerung (Funktionsbibliothek) 136
8.9.4 Prototyp LCD-Display-Ansteuerung 143
8.10 EEPROM (intern und extern) 143
8.10.1 Initialisierung des EEPROM 24C512 148
8.10.2 Daten in das EEPROM 24C512 schreiben 148

8.10.3 Daten vom EEPROM 24C512 lesen 149
8.10.4 Adressraum und Datenfestlegung EEPROM 149
8.10.5 BASCOM Beispielcoder für EEPROM 152
8.10.6 Prototyp externes EEPROM 24C512 166
8.11 Sensorik für Temperatur und Luftfeuchtigkeit 166
8.11.1 Allgemeines zur Temperatur- und Feuchtigkeitsmessung 166
8.11.2 Beschaltung des Sensors SHT71 167
8.11.3 Wichtige Informationen und Notizen zum SHT71 170
8.11.4 BASCOM Treiber und Beispielcode für den SHT71 172
8.11.5 Prototyp zum SHT71 Sensor 185
8.12 Lüftersteuerung für Gehäusebelüftung 186
8.12.1 BASCOM Beispielcode für gemeinsame PWM LCD und Lüfter 187
8.12.2 Prototyp Lüfteransteuerung via PWM 193
8.13 Windgeber 194
8.13.1 Beschaltung Windgeber 194
8.13.2 BASCOM Beispielcode für Auswertung Windgeber 195
8.13.3 Prototyp Windgeberansteuerung 199
8.14 Beschaltung Regensensor 200
8.14.1 Conrad Regensensor: 200
8.14.2 Ankopplung Regensensor 210
8.14.3 BASCOM Treiber und Beispielcode für den Regensensor 213
8.14.4 Prototyp zum Regensensor 216
8.15 Beschaltung DCF77-Modul 217
8.15.1 BASCOM Treiber und Beispielcode für das DCF77-Modul 220
8.15.2 Das DCF77-Signal 229
8.15.3 Prototyp zur DCF77 Anbindung 230
8.16 RTC – Real-Time-Clock 231
8.16.1 Initialisierung (stellen) und aktivieren der Echtzeituhr 234
8.16.2 Auslesen der Uhrzeit 235
8.16.3 BASCOM Beispielcode zur Ansteuerung der RTC 235
8.16.4 Prototyp zur RTC DS1307 235
8.17 Lichtsensorik 236
8.17.1 Beschaltung des ADC0 238
8.17.2 BASCOM Beispielcode zum Lichtsensor TSL250R von TAOS 240
8.17.3 Prototyp zum TSL250R 242
8.18 Akustikausgabe und Sound 243
8.18.1 Ansteuerung des Piezo-Schallwandlers (Schwingkreis) 243
8.18.2 Ansteuerung eines Piezo-Schallwandlers mittels PWM 244
8.18.3 BASCOM Treibersoftware für Soundausgabe 245
8.18.4 Prototyp Piezo-Schwingkreis 248
8.19 Tastatur und Tastenmatrix 248
8.19.1 Allgemeines zu Tastenmatrix an Mikrocontrollern 248
8.19.2 Realisierung Tastenmatrix im Projekt 252
8.19.3 Mechanik der Tasten und Festlegungen 255
8.19.4 Prototyp Tastatur-Matrix 256
8.19.5 BASCOM Treibersoftware für Tastaturmatrix 256
8.20 Latched LED Matrix 262
8.20.1 Allgemeines zur LED-Matrix 262
8.20.2 Prototyp LED-Matrix 265
8.20.3 BASCOM Treibersoftware für Tastaturmatrix 266
8.21 LED-Zuordnung Schaltplan 273

8.22 Leiterplattenverbindungen 274
8.22.1 Leiterplattenverbindung Kontroller <-> Konsole 274
8.22.2 Leiterplattenverbindung Kontroller <-> Power 276
9 Prototypen und Inbetriebnahme 278
10 Hardware 281
10.1 Verbindungskabel und Konnektoren 281
10.1.1 Verbindung Außensensorik zu ConvCtrl 282
10.2 Festlegung von Netzklassen im Projekt 283
10.2.1 Netzklassen für das ATmega128 Evaluation-Board 283
10.2.2 Netzklassen für PCB1 Controller-Board 285
10.2.3 Netzklassen für PCB2 Konsolen-Board 287
10.2.4 Netzklassen für PCB3 Power-Board 288
10.3 Das ATmega128 Evaluation Board 290
10.3.1 Schematic 290
10.3.2 Layout Gesamt 294
10.3.3 Top-Layer 294
10.3.4 Bottom-Layer 294
10.3.5 Pad's und Via's 295
10.3.6 Bestückungsdruck 295
10.3.7 Restricted Areas 296
10.3.8 Eagle-BOM 296
10.3.9 Das EVA-Board 298
10.3.10 Testprogramm für Erstinbetriebnahme 299
10.4 Das Controller-Board PCB1 300
10.4.1 Schematic 300
10.4.2 Layout Gesamt 308
10.4.3 Top-Layer 308
10.4.4 Bottom-Layer 309
10.4.5 Pad's und Via's 309
10.4.6 Bestückungsdruck 310
10.4.7 Restricted Areas 311
10.4.8 Eagle-BOM 311
10.4.9 Das Controller-Board 314
10.4.10 Anmerkungen nach Bestückung und Erstinbetriebnahme 315
10.5 Das Konsolen-Board PCB2 315
10.5.1 Schematic 315
10.5.2 Layout Gesamt 320
10.5.3 Top-Layer 321
10.5.4 Bottom-Layer 322
10.5.5 Pad's und Via's 323
10.5.6 Bestückungsdruck 324
10.5.7 Restricted Areas 326
10.5.8 Eagle-BOM 326
10.5.9 Das Konsolen-Board 328
10.5.10 Anmerkungen nach Bestückung und Erstinbetriebnahme 329
10.6 Das Power-Board PCB3 329
10.6.1 Schematic 329
10.6.2 Layout Gesamt 332
10.6.3 Top-Layer 332
10.6.4 Bottom-Layer 333
10.6.5 Pad's und Via's 333

10.6.6 Bestückungsdruck 334
10.6.7 Restricted Areas 334
10.6.8 Eagle-BOM 335
10.6.9 Das Power-Board 335
10.7 Die fertige Entwicklungsumgebung 336
10.8 Versorgungsplatine für Labor 338
10.8.1 Schematic 339
10.8.2 Layout Gesamt 340
10.8.3 Bottom-Layer 340
10.8.4 Pad's und Via's 340
10.8.5 Bestückungsdruck 340
10.8.6 Restricted Areas 341
10.8.7 Eagle-BOM 341
10.8.8 Die Versorgungsplatine 341
11 Software 342
11.1 Systemfestlegungen und Definitionen 342
11.1.1 Timerfestlegungen 342
11.1.2 Der Watchdog-Timer 343
11.2 PWM-Signale mit Bascom erzeugen 344
11.2.1 Grundbegriffe 344
11.2.2 PWM-Arten 345
11.2.3 PWM-Ablauf 345
11.2.4 Genauere Erklärung 346
11.2.5 Grundprogramm 348
11.3 Der Source-Code zum Projekt ConvCtrl 349
12 Mechanik 519
12.1 Elektromechanische Komponenten 519
12.2 Maßgebende Prinzipzeichnungen Gehäuse 520
12.2.1 Prinzipielle Gesamtansicht 520
12.2.2 Draufsichten 523
12.2.3 Rückseite 524
12.2.4 Seitenansichten 525
12.2.5 Materialauswahl 526
12.2.6 Notizen 527
12.3 Das Gehäuse zur Wintergartensteuerung 528
12.3.1 Wandansicht 528
12.3.2 Bodeneinheit mit Lüfter und Schaltern 528
13 Verwendete Software 529
14 Mitgeltende Unterlagen 530
14.1 Bücher und Literatur 530
14.2 Internet 531
14.2.1 Firmen und Foren 531
14.2.2 ATmega SW und HW-Lösungen 532
14.2.3 Foren 532
14.2.4 Schaltnetzteile 533
14.2.5 Dimmer 533
15 Entwicklungsbegleitende Notizen und Informationen 534
15.1 Evertool 613
15.1.1 Intro 613
15.1.2 Changelog 614
15.1.3 License 614

15.1.4 Hardware 615
15.1.5 Software: 617
15.1.6 Step-by-Step Guide to Software-Installation: 618
15.1.7 Operation 619
15.1.8 Questions and Answers 620
15.1.9 "Evertool-Light" 620
15.1.10 Software-Installation on Evertool LIGHT 622
15.1.11 "Evertool-USB" 622
15.2 ispRE 623
15.2.1 Intro 623
15.2.2 Hardware 623
15.2.3 Software 625
15.2.4 Schitt für Schritt Anleitung 625
15.3 Das STK500 und STK501 626
15.3.1 Das STK500 mit STK501 (ATMEL) 626
15.3.2 STK501 anschließen für RS232-Kommunikation 627
15.4 Inbetriebnahme AVR ISP mkII 628
15.5 Pony-Prog Tutorial 629
15.5.1 Benötigte Ausstattung 629
15.5.2 PonyProg installieren und einrichten 630
15.5.3 Hardwareverbindung herstellen 631
15.5.4 Programmierung des Controllers 633
15.5.5 Fuses 633
15.5.6 Programmierung über den seriellen Port 633
15.5.7 Schaltbild 634
15.5.8 Vorgehensweise 634

3 Abbildungsverzeichnis

Abbildung 1: Entwicklungsumgebung ConvCtrl 22
Abbildung 2: Bedienelemente 36
Abbildung 3: LCD 3x16 37
Abbildung 6: Menüstrukturen ConvCtrl – Sheet 1 38
Abbildung 6: Menüstrukturen ConvCtrl – Sheet 2 39
Abbildung 6: Menüstrukturen ConvCtrl – Sheet 3 40
Abbildung 6: Menüstrukturen ConvCtrl – Sheet 4 41
Abbildung 6: Menüstrukturen ConvCtrl – Sheet 5 42
Abbildung 4: PINOut ATmega32 PDIP 43
Abbildung 5: PINOut ATmega32 TQFP 43
Abbildung 6: PINOut ATmega128 tqfp 44
Abbildung 7: PINOut ATmega32 PDIP 45
Abbildung 8: PINOut ATmega128 TQFP 46
Abbildung 9: PCB des ISP Programmierinterface 53
Abbildung 10: Layer Bottom ISP Programmierinterface 53
Abbildung 11: Bestückung ISP Programmierinterface 53
Abbildung 12: Einstellung Ponyprog für ISP-Programmierinterface 54
Abbildung 13: PFL10 Pfostenverbinder 10 polig 56
Abbildung 14: 2-poliger Platinensteckverbinder 57
Abbildung 15: PCB des SI-Prog Interface 58
Abbildung 16: Layer Bottom SI-Prog Interface 58
Abbildung 17: Bestückung SI-Prog Interface 58
Abbildung 18: PCB des PonyProg ISP-Parallel-Interface 61
Abbildung 19: Layer Bottom PonyProg ISP-Parallel-Interface 61
Abbildung 20: Bestückung PonyProg ISP-Parallel-Interface 62
Abbildung 21: PonyProg Interface Setup 63
Abbildung 22: AVRISP mkII Adapter 64
Abbildung 22: ATMEl JTAg Interface 65
Abbildung 23: D-SUB Buchse 9-polig für RS232 70
Abbildung 24: Fusebits im AVR Studio 78
Abbildung 25: PonyProg Fuse Bits 79
Abbildung 26: AVRDUDE 79
Abbildung 27: Fusebits bei Auslieferung und nach Programmierung 83
Abbildung 28: Hauptschalter 86
Abbildung 29: Bestückung der Conrad Vorschaltplatine 87
Abbildung 30: HAHN Trafo 91
Abbildung 31: Transformator Hahn BV UI 394 0088 92
Abbildung 32: PINOut Hahn Trafo BV UI 394 0088 (Ansicht von unten) 92
Abbildung 33: BLOCK Trafo 94
Abbildung 34: Transformator Block FL30/15 94
Abbildung 35: PINOut Block-Trafo FL30/15 (Ansicht von unten) 95
Abbildung 36: Schaltregler LM2576 96
Abbildung 37: Steuerungsrelais 98
Abbildung 38: Funktionsprinzipt Phasenanschnittsteuerung 106
Abbildung 39: Funktionsprinzipt Phasenanschnittsteuerung 107
Abbildung 40: Steuerlogik für Phasenanschnitt 108
Abbildung 41: Oszillogramm der Nulldurchgangserkennung 110
Abbildung 42: Oszillogramm Triac-Zündung „dunkel“ 111

Abbildung 43: Oszillogramm Triac-Zündung „hell“ 111
Abbildung 44: Oszillogramm aller Signal inkl. Netz-Sinus 111
Abbildung 45: Oszillogramm Triac-Ausgangssignal „hell“ 111
Abbildung 46: Oszillogramm Triac-Ausgangssignal „50%“ 111
Abbildung 47: Oszillogramm Triac-Ausgangssignal „dunkel“ 111
Abbildung 48: Angeschnittener Phasenverlauf 112
Abbildung 49: Sinus mit Timer-Rampen 113
Abbildung 50: Konvertierungstabelle TRIAC Zündzeitpunkte 114
Abbildung 51: Detailrechung Timer für Dimmer-Zündungssteuerung 117
Abbildung 52: Prototyp Relaisansteuerung und Dimmer mit STK 128
Abbildung 53: Prototyp Relaisansteuerung und Dimmer PCB 128
Abbildung 54: LCD-Display EA DOG-M 129
Abbildung 55: Abmessungen und technische Daten DOGM LCD-Display 130
Abbildung 56: Prototyp LCD-Display-Ansteuerung mit STK 143
Abbildung 57: Prototyp LCD-Display-Ansteuerung PCB 143
Abbildung 58: Adressierung des ST24C512 145
Abbildung 59: Daten in den ST24C512 schreiben 146
Abbildung 60: Daten aus dem ST24C512 lesen 146
Abbildung 61: Aktuelle Adresse aus dem ST24C512 lesen 146
Abbildung 62: Daten aus dem ST24C512 via Radom-Read lesen 146
Abbildung 63: Daten sequentiell aus dem ST24C512 auslesen 146
Abbildung 64: Oszillogramm einer I^2C Schreibseqenz 147
Abbildung 65: Oszillogramm I^2C mit 10kΩ bei 400k Baud 147
Abbildung 66: Oszillogramm I^2C mit 5kΩ bei 400 kBaud 147
Abbildung 67: Oszillogramm I^2C bei 100 kBaud 148
Abbildung 68: Oszillogramm I^2C bei 400 kBaud 148
Abbildung 69: Ein Byte in den ST24C512 schreiben 148
Abbildung 70: Ein Byte aus dem ST24C512 lesen 149
Abbildung 71: Prototyp EEPROM Ansteuerung 166
Abbildung 72: Sensor für Luftfeuchtigkeit und Temperatur 166
Abbildung 73: two wire interface für SHT1x / SHT7x 167
Abbildung 74: Oszillogramm SHT71 „connection reset sequence“ 168
Abbildung 75: Oszillogramm SHT71 „transmission start“ 168
Abbildung 76: SHT71 „transmission start“ und Byte-Wert (Reset) senden 169
Abbildung 77: SHT71 Softreset 169
Abbildung 78: SHT71 Statusregister lesen 169
Abbildung 79: SHT71 Statusregister schreiben 170
Abbildung 80: Die State machine des SHT71 170
Abbildung 81: Serial data (DATA) des SHT71 171
Abbildung 82: Transmission Sequence Start des SHT71 171
Abbildung 83: Connection Reset Sequence des SHT71 172
Abbildung 84: Mess-Sequence des SHT71 172
Abbildung 85: Statusregister des SHT71 schreiben und lesen 172
Abbildung 86: Traceausgabe SHT71 des BASCOM-Beispielcodes 173
Abbildung 87: Prototyp SHT71 Sensor-Anbindung 186
Abbildung 88: Prototyp Lüftersteuerung mit Lüfter 193
Abbildung 89: Prototyp Lüftersteuerung Detailansicht 193
Abbildung 90: Kleinwindgeber der Firma rb-Messtechnik 194
Abbildung 91: Kleinwindgeber mit Beschaltung 200
Abbildung 92: Beschaltung des Kleinwindgebers 200
Abbildung 93: Korrekte Verschaltung Regensensor 210

Abbildung 94: Kippschalter für Regensensorheizung! 213
Abbildung 95: EX-Kabelverschraubung K-M schwarz 213
Abbildung 96: Prototypaufbau Regensensor mit Beschaltung 216
Abbildung 97: Prototypaufbau Beschaltung für Regensensor im Detail 216
Abbildung 98: Innenleben Regensensor-Elektronik 217
Abbildung 99: Gesamtkomponente Regensensor inkl. Sensorfläche 217
Abbildung 100: Beschaltung Conrad DCF77-Modul 218
Abbildung 101: Conrad DCF77-Modul 218
Abbildung 102: Prototypaufbau DCF77 und RTC mit LCD-Display 230
Abbildung 103: Beschaltung DCF77 230
Abbildung 104: Beispielhafte Zeitausgabe über SoftClock mit DCF77 231
Abbildung 105: Das DCF77 Funkuhrmodul 231
Abbildung 106: TWI 2-Wire BUS-Konfiguration 233
Abbildung 107: Adressierung DATA READ – DS1307 233
Abbildung 108: Adressierung DATA WRITE – DS1307 233
Abbildung 109: DS1307 Adress-MAP 234
Abbildung 110: DS1307 Registers 234
Abbildung 111: Daten in DS1307 schreiben 234
Abbildung 112: Daten aus dem DS1307 lesen 235
Abbildung 113: Beschaltung RTC DS1307 236
Abbildung 114: Spektrale Empfindlichkeit Lichtsensor 236
Abbildung 115: Analog Noise Canceling Techniques 239
Abbildung 116: Beschaltung TAOS TSL250R 243
Abbildung 117: Detailansicht Beschaltung Lichtsensor 243
Abbildung 118: Prototyp des Piezo-Schwingkreises zur Soundausgabe 248
Abbildung 119: Tastenmatrix - 4x4-Tastenmatrix im Schlafzustand (rot=LOW) 248
Abbildung 120: Tastenmatrix - aufgeweckt, weil Taste gedrückt 249
Abbildung 121: Tastenmatrix - Linke Spalte wird abgefragt: Pech! 249
Abbildung 122: Tastenmatrix - Zweite Spalte wird abgefragt: Glück! 250
Abbildung 123: Tastenmatrix - Zwei Tasten sind problemlos 250
Abbildung 124: Tastenmatrix - Alle Tasten sind hiermit kein Problem 251
Abbildung 125: Tastenmatrix - Halbierte Portpin-Zahl, aber Dioden 251
Abbildung 126: Tastenmatrix - Keine Dioden 252
Abbildung 127: Tastendimensionen und Platbedarf 256
Abbildung 128: Prototyp der 3x3 Tastematrix 256
Abbildung 129: SN74116 Dual 4-Bit Latch 263
Abbildung 130: Prototyp der LED-Matrix 265
Abbildung 131: Verbindungskonfiguration PCB1 <-> PCB2 274
Abbildung 132: Connector PCB1<->PCB2 276
Abbildung 133: Verbindungskonfiguration PCB1 <-> PCB3 276
Abbildung 134: Connector PCB1<->PCB3 Niedervolt 277
Abbildung 135: Connector PCB1<->PCB3 Netzspannung 277
Abbildung 136: Außensensorik auf Aluminium-Träger 283
Abbildung 137: Definition der Netzklassen Eva-Board 284
Abbildung 138: Demoboard N etzklassen Eva-Board 285
Abbildung 139: Definition der Netzklassen PCB1 285
Abbildung 140: Demoboard N etzklassen PCB1 286
Abbildung 141: Definition der Netzklassen PCB2 287
Abbildung 142: Demoboard N etzklassen PCB2 288
Abbildung 143: Definition der Netzklassen PCB3 288
Abbildung 144: Demoboard N etzklassen PCB3 289

Abbildung 145: ATmega128 Evaluation-Board– Layout gesamt 294
Abbildung 146: ATmega128 Evaluation-Board– Top-Layer 294
Abbildung 147: ATmega128 Evaluation-Board– Bottom-Layer 295
Abbildung 148: ATmega128 Evaluation-Board– Pad's und Via's 295
Abbildung 149: ATmega128 Evaluation-Board– Bestückung 296
Abbildung 150: ATmega128 Evaluation-Board– Restricted Areas 296
Abbildung 151: Das ATmega128 EVA-Board 299
Abbildung 152: PCB1 Controler-Board– Layout gesamt 308
Abbildung 153: PCB1 Controller-Board– Top-Layer 309
Abbildung 154: PCB1 Controler-Board– Bottom-Layer 309
Abbildung 155: PCB1 Controler-Board– Pad's und Via's 310
Abbildung 156: PCB1 Controler-Board– Bestückungsdruck 310
Abbildung 157: PCB1 Controler-Board– Restricted Areas 311
Abbildung 154: PCB1 Controler-Board– PCB 314
Abbildung 154: PCB1 Controler-Board– fertig bestückt. 315
Abbildung 158: PCB2 Konsolen-Board– Layout gesamt 321
Abbildung 159: PCB2 Konsolen-Board– Top-Layer 322
Abbildung 160: PCB2 Konsolen-Board– Bottom-Layer 323
Abbildung 161: PCB2 Konsolen-Board– Pad's und Via's 324
Abbildung 162: PCB2 Konsolen-Board– Bestückungsdruck 325
Abbildung 163: PCB2 Konsolen-Board– Beschriftung 326
Abbildung 164: PCB2 Konsolen-Board– Restricted Areas 326
Abbildung 164: PCB2 Konsolen-Board– PCB 328
Abbildung 164: PCB2 Konsolen-Board– fertig bestückt 329
Abbildung 165: PCB3 Power-Board – Layout gesamt 332
Abbildung 166: PCB3 Power-Board – Top-Layer 332
Abbildung 167: PCB3 Power-Board – Bottom Layer 333
Abbildung 168: PCB3 Power-Board – Pad's and Via's 333
Abbildung 169: PCB3 Power-Board – Bestückung 334
Abbildung 170: PCB3 Power-Board – Restricted Areas 334
Abbildung 171: PCB3 Power-Board 336
Abbildung 171: ConvCtrl – Das vollständige Komplettsystem 337
Abbildung 171: ConvCtrl – Nummer EINS lebt! 338
Abbildung 171: ConvCtrl – Die Entwicklungsumgebung 338
Abbildung 165: PowerForDevelopment – Layout gesamt 340
Abbildung 165: PowerForDevelopment – Bottom-Layer 340
Abbildung 165: PowerForDevelopment – Pad's und Via's 340
Abbildung 165: PowerForDevelopment – Bestückungsdruck 341
Abbildung 165: PowerForDevelopment – Restricted Areas 341
Abbildung 171: ConvCtrl – PowerForDevelopment 342
Abbildung 172: PWM Tastverhältnis einmal von 10% und einmal von 50% 345
Abbildung 173: PWM Konfiguration des ATmega 346
Abbildung 174: PWM mit einem Tastverhältnis 20% 347
Abbildung 175: PWM mit einem Tastverhältnis von 80% 347
Abbildung 176: Oszillogramm der PWM mit Time 1a und 1b 348
Abbildung 177: Steckverbindungen und elektromechanische Komponenten 520
Abbildung 178: Gehäuse prinzipielle Gesamtansicht 521
Abbildung 179: Gehäuse prinzipielle Gesamtansicht Front 522
Abbildung 180: Gehäuse Draufsichten 523
Abbildung 181: Gehäuse Rückseite 524
Abbildung 182: Gehäuse Seitenansichten 525

Abbildung 183: Gehäuse Materialauswahl 526
Abbildung 184: Notizen 1 527
Abbildung 185: Notizen 2 528
Abbildung 185: Gehäuse Bodeneinheit von außen 529
Abbildung 185: Gehäuse Bodeneinheit von innen 529
Abbildung 186: Evertool V 0.6 JMP Board 615
Abbildung 187: Evertool Development Prototype Board 617
Abbildung 188: SMD Side 621
Abbildung 189: Connector Side 621
Abbildung 190: istRE Testplatine Lochraster 624
Abbildung 191: Das STK500 mit STK501 (ATMEL) 627
Abbildung 192: RS232-Verbindung STK500 und STK501 627
Abbildung 193: RS232-Verbindung STK500 mit STK501 628
Abbildung 194: AVR Studio Programmer Selection 628
Abbildung 195: AVR Studio Board-Settings 629
Abbildung 196: AVR Studio Program Settings 629
Abbildung 197: Pony Prog I/O port setup 630
Abbildung 198: Pony Prog Device Setup 631
Abbildung 199: Wannenstecker für 10-poligen ISP Anschluss 632
Abbildung 200: AVR ISP Anschlussbelegung 632

4 Tabellenverzeichnis

Tabelle 1: Festlegungen und Entscheidungen 24
Tabelle 2: Detailfunktionen Spannungsversorgung 24
Tabelle 3: Detailfunktionen Spannungsversorgung 25
Tabelle 4: Detailfunktionen Mikrocontroller 25
Tabelle 5: Detailfunktionen LCD-Display 26
Tabelle 6: Detailfunktionen Tastatur und Schalter 27
Tabelle 7: Detailfunktionen LED-Anzeigen 28
Tabelle 8: Detailfunktionen Windmessung 28
Tabelle 9: Detailfunktionen Regensensorik 28
Tabelle 10: Detailfunktionen Temperatursenorik 29
Tabelle 11: Detailfunktionen Luftfeuchtigkeit 29
Tabelle 12: Detailfunktionen Dimmersteuerung 30
Tabelle 13: Detailfunktionen Beschattung 30
Tabelle 14: Detailfunktionen Kühlung 31
Tabelle 15: Detailfunktionen Datenspeicherung 31
Tabelle 16: Detailfunktionen DCF77 Funkuhr 31
Tabelle 17: Detailfunktionen Real-Time-Clock 32
Tabelle 18: Detailfunktionen Gehäuse 32
Tabelle 19: Detailfunktionen Helligkeitssensorik 32
Tabelle 20: Detailfunktionen Zeit und Datumsinformation 32
Tabelle 21: Detailfunktionen Akustik 33
Tabelle 22: Detailfunktionen CheckUp 33
Tabelle 23: Detailfunktionen zeitkritische Timerfunktionen 33
Tabelle 24: Stückliste CPU-Unit ATmega32 43
Tabelle 25: Stückliste CPU-Unit ATmega128 44
Tabelle 26: Ressourcenzuordnung ATmega32 46
Tabelle 27: Ressourcenzuordnung ATmega128 48
Tabelle 28: Interrupt-Vektor-Tabelle TAmega128 50
Tabelle 29: Stückliste Basisbeschaltung ATmega 51
Tabelle 30: Kurzhubtaster für RESET 51
Tabelle 31: ISP connection Pinout 52
Tabelle 32: Stückliste Basisbeschaltung ATmega 54
Tabelle 33: Stückliste Basisbeschaltung ATmega mit ISP-Programmierinterface 55
Tabelle 34: Stückliste Basisbeschaltung ATmega mit ISP-Programmierinterface 56
Tabelle 35: Stückliste SI-Prog Interface 58
Tabelle 36: Serielle Kabel SI-Prog-Interface 59
Tabelle 37: Stückliste PonyProg ISP-Parallel-Interface 62
Tabelle 38: Stückliste JTAG Emu 67
Tabelle 39: PIN-Belegung des LPT-Ports 68
Tabelle 40: Stückliste Basisbeschaltung ATmega mit ISP uns RS232 70
Tabelle 41: PIN-Belegung der 9-poligen RS232 Stecker/Buchse 71
Tabelle 42: PIN-Belegung des seriellen RS232-Ports 71
Tabelle 43: Stückliste Basisbeschaltung ATmega mit ISP uns RS232 72
Tabelle 44: Stückliste Basisbeschaltung ATmega mit ISP uns RS232 73
Tabelle 45: Themenbezogene LED's für Gesamtbeschaltung 74
Tabelle 46: Fuse High Byte ATmega 128 81
Tabelle 47: Extended Fuse Byte ATmega 128 81
Tabelle 48: Fuse Low Byte ATmega 128 83

© Markus Fulde, 2007

Tabelle 49: Fuse High Byte ATmega 128 83
Tabelle 50: Extended Fuse Byte ATmega 128 83
Tabelle 51: AVR-Studio - Main 84
Tabelle 52: AVR-Studio - Program 84
Tabelle 53: AVR-Studio - Fuses 84
Tabelle 54: AVR-Studio - LockBits 84
Tabelle 55: AVR-Studio - Advanced 85
Tabelle 56: AVR-Studio – MainHW Settings 85
Tabelle 57: AVR-Studio – HW Info 85
Tabelle 58: AVR-Studio - Auto 85
Tabelle 59: Stückliste Vorschaltplatine 86
Tabelle 60: Stückliste Vorschaltplatine 87
Tabelle 61: Gesamtstromverbrauch für 5V Komponenten 89
Tabelle 62: Gesamtstromverbrauch für 12V Komponenten 89
Tabelle 63: Kühlkörper für Festspannungsregler 90
Tabelle 64: Vorgeschriebene Namensgebung für Spannungsversorgungen 90
Tabelle 65: Stückliste Nulldurchgangskennung 93
Tabelle 66: Leistungsdaten Bloeck Trafo Fl 30/15 95
Tabelle 67: Stückliste Schaltnetzteil +5V und +12V 97
Tabelle 68: Themenbezogene LED's für Schaltnetzteil 97
Tabelle 69: Adernbelegung Motoranschlüsse Beschattung 97
Tabelle 70: Beschreibung zum Relais 99
Tabelle 71: Stückliste Treiberstufe für Motor-Steuerrelais 101
Tabelle 72: Stückliste Treiberstufe für Lichtsteuerung AN/AUS 102
Tabelle 73: Stückliste Gesamtbeschaltung Relaisansteuerung 103
Tabelle 74: Stückliste Gesamtbeschaltung Relaisansteuerung 104
Tabelle 75: Themenbezogene LED's für Relaissteuerung 105
Tabelle 76: Ressourcenzuordnung ATmega128 für Relaisansteuerung 105
Tabelle 77: Stückliste Nulldurchgangskennung 109
Tabelle 78: Ressourcenzuordnung ATmega128 für Nulldurchgangserkennung 109
Tabelle 79: Stückliste Leistungsteil Dimmer 110
Tabelle 80: Ressourcenzuordnung ATmega128 für Triac-Zündstufe 111
Tabelle 81: LED-Hintergrundbeleuchtung für LCD-Display 131
Tabelle 82: Stückliste LCD Beschaltung 132
Tabelle 83: Ressourcenzuordnung ATmega128 für LCD-Display 133
Tabelle 84: Zeichensatz des LCD-Displays 133
Tabelle 85: LCD-Display EA DOGM Instruction Code 134
Tabelle 86: LCD-Display EA DOGM Instruction table 0 134
Tabelle 87: LCD-Display EA DOGM Instruction table 1 135
Tabelle 88: LCD-Display EA DOGM Instruction table 2 135
Tabelle 89: LCD-Display EA DOGM162 Initialisierungsbeispiel 135
Tabelle 90: LCD-Display EA DOGM163 Initialisierungsbeispiel 136
Tabelle 91: Stückliste externes EEPROM 145
Tabelle 92: Themenbezogene LED's für EEPROM 145
Tabelle 93: Ressourcenzuordnung ATmega128 für EEPROM 145
Tabelle 94: Adressraum und Speicherbedarf 149
Tabelle 95: Negative Zahlen im dualen System 150
Tabelle 96: Datenstrukturen und Datenfelder im EEPROM (intern und extern) 152
Tabelle 97: Stückliste externes EEPROM 167
Tabelle 98: Stückliste Temperatur- und Luftfeuchtigkeitssensor 168
Tabelle 99: Ressourcenzuordnung ATmega128 für EEPROM 168

Tabelle 100: Kommandos-Codes des SHT71 171
Tabelle 101: Statusregisterinhalte des SHT71 172
Tabelle 102: Stückliste Lüfteransteuerung 187
Tabelle 103: Themenbezogene LED's Lüftersteuerung 187
Tabelle 104: Ressourcenzuordnung ATmega128 für Lüftersteuerung 187
Tabelle 105: Stückliste Windgeberbeschaltung 195
Tabelle 106: Themenbezogene LED's Windimpuls 195
Tabelle 107: Ressourcenzuordnung ATmega128 für Lüftersteuerung 195
Tabelle 108: Stückliste Beschaltung Regensensor 212
Tabelle 109: Themenbezogene LED's Regensensor 212
Tabelle 110: Ressourcenzuordnung ATmega128 für Regensensor 213
Tabelle 111: Pinbelegung Steuerleitung Regensensor 213
Tabelle 112: Stückliste Beschaltung DCF77-Modul 220
Tabelle 113: Themenbezogene LED's DCF77 220
Tabelle 114: Ressourcenzuordnung ATmega128 für DTC77 220
Tabelle 115: Bit-Bedeutung DCF77-Signal 229
Tabelle 116: Stückliste Beschaltung RTC 232
Tabelle 117: Ressourcenzuordnung ATmega128 für RTC 233
Tabelle 118: Stückliste Beschaltung Lichtsensor 238
Tabelle 119: Ressourcenzuordnung ATmega128 für Lichtsensor 238
Tabelle 120: Stückliste Basisbeschaltung ATmega mit ISP, RS232 und ADC 240
Tabelle 121: Stückliste Beschaltung Piezo-Signalgeber 244
Tabelle 122: Ressourcenzuordnung ATmega128 für Piezo-Signalgeber 244
Tabelle 123: Piezo-Schallgeber 245
Tabelle 124: Stückliste Beschaltung Piezo-Signalgeber 245
Tabelle 125: Stückliste 3x3 Tastaturmatrix Grundprinzip 253
Tabelle 126: Stückliste 3x3 Tastaturmatrix Umsetzungsvariante 254
Tabelle 127: Themenbezogene LED's für Tastenmatrix 254
Tabelle 128: Ressourcenzuordnung ATmega128 für Tastaturmatrix 255
Tabelle 129: Eingabetaster 255
Tabelle 130: Asuwahl Eingabetaster 256
Tabelle 131: Latched LED's 263
Tabelle 132: None Latched LED's 263
Tabelle 133: Latched LED Array 264
Tabelle 134: Ressourcenzuordnung ATmega128 für Tastaturmatrix 265
Tabelle 135: Widerstands-Netzwerk 265
Tabelle 136: LED-Zuordnung Schaltplan 274
Tabelle 137: PIN-Belegung Leiterplattenverbindung PCB1<->PCB2 275
Tabelle 138: PIN-Belegung Leiterplattenverbindung PCB1<->PCB3 Niedervolt 277
Tabelle 139: PIN-Belegung Leiterplattenverbindung PCB1<->PCB3 Hochvolt 277
Tabelle 140: Bauelemente Conrad Electronic Prototyping 278
Tabelle 141: Bauelemente Reichelt Electronic Prototyping 281
Tabelle 142: Bauelemente Farnell Prototyping 281
Tabelle 143: Bauelemente RS-Online Prototyping 281
Tabelle 144: Pinbelegung Steuerleitung Regensensor 282
Tabelle 145: Pinbelegung Steuerleitung Windmesser 282
Tabelle 146: Pinbelegung Steuerleitung Außensensorik 282
Tabelle 147: Anschlussverbindungen Wind<->Regen<->ConvCtrl 283
Tabelle 148: Eagle BOM für Evaluation-Board 298
Tabelle 149: Bauteilspezialitäten ATmega128 Evalutation Board 298
Tabelle 150: Eagle BOM für PCB1 – Controller-Board 314

Tabelle 151: Eagle BOM für PCB2 – Konsolen-Board 328
Tabelle 152: Eagle BOM für Power-Board 335
Tabelle 152: Eagle BOM für PowerForDevelopment 341
Tabelle 153: Watchdog Konfiguration als Funktion von M103C und WDTON 343
Tabelle 154: Codierwerte für den Watchdog Timer 343
Tabelle 155: Stückliste ispRE 625

5 Schaltbilderverzeichnis

Schaltbild 1: Basisbeschaltung ATmega 51
Schaltbild 2: 10-Pin ISP connection Pinout 52
Schaltbild 3: ISP Programmierinterface 53
Schaltbild 4: ATmega Basisbeschaltung mit ISP-Programmierinterface 55
Schaltbild 5: ISP Adapter 6-Pin zu 10-Pin. 56
Schaltbild 6: PonyProg SI-Prog Interface 57
Schaltbild 7: PonyProg ISP-Parallel-Interface 60
Schaltbild 7: ConvCtrl Schnittstellen und Adapter 64
Schaltbild 8: Schaltplan AVR JTAG Emu 66
Schaltbild 9: Bestückungsplan AVR JTAg Emu 66
Schaltbild 10: Vorgeschriebene Belegung eines IEEE 1284 AB-Kabels 69
Schaltbild 11: ATmega Basisbeschaltung mit ISP und RS232-Schnittstelle 70
Schaltbild 12: Verbindungskabel zwischen PIN-Header und D-SUB9 Buchse 72
Schaltbild 13: Gesamt-Grundbeschaltung des ATmega128 73
Schaltbild 14: Vorschaltung für LED an 230V~ 86
Schaltbild 15: Schaltbild der Conrad Vorschaltung für LED's 87
Schaltbild 16: Symbolfestlegung für Spannungsversorgungen 91
Schaltbild 17: Spannungsversorgung +5V und +12V mit Festspannungsregler 93
Schaltbild 18: Schaltnetzteil +5V und +12V 96
Schaltbild 19: Verriegelung der Relais durch direkt Beschaltung 99
Schaltbild 20: Verriegelung der Relais durch Logikbausteine 100
Schaltbild 21: Treiberstufen für Motor-Steuerrelais Beschattung 100
Schaltbild 22: Treiberstufen für Lichtsteuerung AN/AUS 101
Schaltbild 23: Gesamtbeschaltung Relaisansteuerung 103
Schaltbild 24: Gesamtbeschaltung Relaisansteuerung 104
Schaltbild 25: Nulldurchgangserkennung (Dimmer) 109
Schaltbild 26: Leistungsteil (Dimmer) 110
Schaltbild 27: Beschaltung LCD-Display gemäß Datenblatt 131
Schaltbild 28: Beschaltung LCD-Display im Projekt ConvCtrl 132
Schaltbild 29: Beschaltung externes EEPROM 144
Schaltbild 30: Beschaltung des Temperatur- und Luftfeuchtigkeitssensors SHT71 167
Schaltbild 31: Schaltplan für Lüfteransteuerung inkl. Status-LED 186
Schaltbild 32: Beschaltung Windgeber 195
Schaltbild 33: Schaltbild Regensensor 211
Schaltbild 34: Schaltbild Beschaltung Regensensor 212
Schaltbild 35: Schaltplan DCF77-Modul von Conrad 219
Schaltbild 36: Aufschaltung DCF77 an ATmega 219
Schaltbild 37: Aufschaltung des DS1307 an ATmega 232
Schaltbild 38: Basisbeschaltung TAOS TSL250R 237
Schaltbild 39: Schaltplan zum Lichtsensor 238
Schaltbild 40: Beschaltung ATmega inkl. Störsignalreduktion 239
Schaltbild 41: Beschaltung Piezo-Signalgeber 244
Schaltbild 42: Beschaltung Piezo-Signalgeber mit PWM 245
Schaltbild 43: 3x3 Tastaturmatrix Grundprinzip 253
Schaltbild 44: 3x3 Tastaturmatrix Umsetzungsvariante 254
Schaltbild 45: Latch LED Array 264
Schaltbild 46: Schaltbild für Definition von Netzklassen – Eva-Board 284
Schaltbild 47: Schaltbild für Definition von Netzklassen PCB1 286

Schaltbild 48: Schaltbild für Definition von Netzklassen PCB2 287
Schaltbild 49: Schaltbild für Definition von Netzklassen PCB3 289
Schaltbild 50: Schaltbild Atmega128 Evaluation-Board - Sheet 1 290
Schaltbild 51: Schaltbild Atmega128 Evaluation-Board - Sheet 2 291
Schaltbild 52: Schaltbild Atmega128 Evaluation-Board - Sheet 3 291
Schaltbild 53: Schaltbild Atmega128 Evaluation-Board - Sheet 4 292
Schaltbild 54: Schaltbild Atmega128 Evaluation-Board - Sheet 5 292
Schaltbild 55: Schaltbild Atmega128 Evaluation-Board - Sheet 6 293
Schaltbild 56: Schaltbild Atmega128 Evaluation-Board - Sheet 7 293
Schaltbild 57: Schaltbild PCB1 Controller-Board - Sheet 1 301
Schaltbild 58: Schaltbild PCB1 Controller-Board - Sheet 2 301
Schaltbild 59: Schaltbild PCB1 Controller-Board - Sheet 3 302
Schaltbild 60: Schaltbild PCB1 Controller-Board - Sheet 4 302
Schaltbild 61: Schaltbild PCB1 Controller-Board - Sheet 5 303
Schaltbild 62: Schaltbild PCB1 Controller-Board - Sheet 6 303
Schaltbild 63: Schaltbild PCB1 Controller-Board - Sheet 7 304
Schaltbild 64: Schaltbild PCB1 Controller-Board - Sheet 8 304
Schaltbild 65: Schaltbild PCB1 Controller-Board - Sheet 9 305
Schaltbild 66: Schaltbild PCB1 Controller-Board - Sheet 10 305
Schaltbild 67: Schaltbild PCB1 Controller-Board - Sheet 11 306
Schaltbild 68: Schaltbild PCB1 Controller-Board - Sheet 12 306
Schaltbild 69: Schaltbild PCB1 Controller-Board - Sheet 13 307
Schaltbild 70: Schaltbild PCB1 Controller-Board - Sheet 14 307
Schaltbild 71: Schaltbild PCB2 Konsolen-Board - Sheet 1 316
Schaltbild 72: Schaltbild PCB2 Konsolen-Board - Sheet 2 316
Schaltbild 73: Schaltbild PCB2 Konsolen-Board - Sheet 3 317
Schaltbild 74: Schaltbild PCB2 Konsolen-Board - Sheet 4 317
Schaltbild 75: Schaltbild PCB2 Konsolen-Board - Sheet 5 318
Schaltbild 76: Schaltbild PCB2 Konsolen-Board - Sheet 6 318
Schaltbild 77: Schaltbild PCB2 Konsolen-Board - Sheet 7 319
Schaltbild 78: Schaltbild PCB2 Konsolen-Board - Sheet 8 319
Schaltbild 79: Schaltbild PCB2 Konsolen-Board - Sheet 9 320
Schaltbild 80: Schaltbild PCB3 Power-Board - Sheet 1 330
Schaltbild 81: Schaltbild PCB3 Power-Board - Sheet 2 330
Schaltbild 82: Schaltbild PCB3 Power-Board - Sheet 3 331
Schaltbild 83: Schaltbild PCB3 Power-Board - Sheet 4 331
Schaltbild 80: Schaltbild PowerForDevelopment – Sheet 1 339
Schaltbild 80: Schaltbild PowerForDevelopment – Sheet 2 340
Schaltbild 84: Evertool (JTAGICE/AVRISP) 615
Schaltbild 85: Evertool Light 621
Schaltbild 86: ispRE 624
Schaltbild 87: AVR ISP to COM1 634

6 Softwareverzeichnis

Software 1: Code zur Initialisierung der TRIAC-Zündzeitpunkte 115
Software 2: Code Zur Ansteuerung der Relaisstufen und Dimmereinheit 126
Software 3: Code für TRIAC Interrupt-Service-Routine aus BASCOM 127
Software 4: Code zur Ansteuerung des LCD-Displays 142
Software 5: Code zur Ansteuerung des EEPROM (extern und intern) 166
Software 6: Code zum Betrieb des SHT71 am ATmega 185
Software 7: Code zum Betrieb des Lüfters und LCD an einem Timer (PWM) 193
Software 8: Code zum Betrieb des Kleinwindgebers 199
Software 9: Code zum Betrieb des Regensensors 216
Software 10: Code zum Betrieb der DCF77 und RTC 229
Software 11: Code zum Betrieb des TSL250R 242
Software 12: Code für die 3x3 Tastenmatrix 262
Software 13: Code für die LED-Matrix 273
Software 14: EVA-Board COM2_TEST 299
Software 15: EVA-Board IO_TEST 300
Software 16: Code zum Bedienen des Watchdog-Timers 344
Software 16: Der Source-Code zum Projekt ConvCtrl 519

7 Allgemeines

Die Entwicklungsumgebung des Projekts ConCtrl:



Abbildung 1: Entwicklungsumgebung ConvCtrl

7.1 Leistungsumfang

im Folgenden wird der Leistungsumfang und die Teilfunktionalität beschrieben welche ConvCtrl besitzt und dem Benutzer zur Verfügung stellt:

- Hauptschalter
- LCD-Display
  - Menübedienung
  - Anzeige von StartUp- und Statusmeldungen
  - Anzeige von Messwerten und Logging-Daten
- Tastatur
  - Menüsteuerung
  - Direkt-Steuerung der Beschattungsanlage (Auf / Zu / Stop)
  - Direkt-Steuerung Wintergarten-Beleuchtung (Heller, Dunkler, Aus)
- Sensorik
  - Regensensor zur Detektion von Regen im Außenbereich
  - Windgeber zur Erfassung von Windgeschwindigkeiten
  - Luftfeuchtigkeit
  - Temperatur
  - Helligkeit
- Uhr
  - Echtzeituhr und Kopplung an DCF77-Modul für automatische Uhreinstellung
  - Programmierung von Öffnung- und Schließungszeiten Beschattung
  - Bereitstellung Uhrzeit für Datenerfassung
- Datenerfassung (Speicherung mit Uhrzeit und Datum)
  - Regensensordaten
  - Windgeschwindigkeitsdaten

  - Luftfeuchtigkeit
  - Temperatur
  - Helligkeit

- Sonstiges
  - Von außen zugänglicher Reset-Taster
  - Lüfter mit Lüftungsgitter für Kühlung
- Schnittstellen
  - RS232 Schnittstelle zum Tracing und zur Datenübertragung der Logging-Daten an PC
  - ISP Schnittstelle zur direkten Programmierung des Target
  - JTAG Interface für On-Chip-Debuging
- Dimmermodul für Lichtsteuerung der Wintergartenbeleuchtung
- Akustische Warnung und Signalisierungen über Piezo-Schallgeber

7.2 Funktionskomponenten

Das Projekt ConvCtrl verfügt über die folgenden einzelnen Funktions- / Teilkomponenten:

- Spannungsversorgung für +5V und +12V SUP
- Mikrocontroller ATmega32 oder ATmega128 (inkl. ISP, RS232 und Reset) µC
- LCD-Display 3x16 (inkl. Rahmen mit "Scheibe") LCD
- Tastatur und Schalter SW
- LED-Anzeigen für Status- und Funktionsanzeige LED
- Windmesser (pro Tick = 2,5m/sec Windgeschwindigkeit) WIND
- Regensensor (ja/nein Detektion inkl. Detektorheizung) RAIN
- Temperatursensor (Anschluss über I^2C Bus) TEMP
- Luftfeuchtigkeit (Anschluss über I^2C Bus) HUM
- Dimmungssteuerung für Beleuchtung (inkl. ein/aus Relais) DIM
- Motorsteuerung für Beschattung (inkl. Relaisverriegelung und Relais-Timeout) CONV
- Gehäusebelüftung mit Option Lüftersteuerung über ATmega FREEZ
- Datenspeicher / Datenlogger über EEPROM (Anschluss über I^2C Bus) LOG
- DCF77 Funkuhrempfang (Conrad) DCF
- Passendes Selbstbaugehäuse aus Aluminium und rauchgläserner Front BOX
- Helligkeitsmessung (Licht -> DC-Wandler mit OPAmp) LIGHT
- Anzeige von Zeit- und Datumsinformationen CLOCK
- Akustisches Feedback über Piezo Schallwandler SOUND
- Selbstcheck bei StartUp und ggf. Ausgabe von Warnmeldungen und Signalen CHECK
- Timersteuerung zeitkritischer Applikationen (Tasten, Dimmer, Helligkeit Display) TIME

7.3 Festlegungen und Entscheidungen

Datum	*Entscheidung*
23.07.2007	Bei der Ressourcenverteilung auf den ATmega32 wurde festgestellt, dass die IN-Pins und OUT-Pins nicht ausreichen. Hierbei ist das Ibterface für Tastatur und LED's nicht berücksich-

	tigt! Hiermit wird die Entscheidung getroffen Anstelle des ATmega32 den ATmega128 einzusetzen. Notwendige Beschaltunsgabweichungen und andere Vorgehensweisen werden im Dokument separat ausgewiesen und beschrieben!
03.09.2007	Aufgrund der in Kapitel „Zusammenfassung nach Inbetriebnahme“ unter „Netzteil mit Festspannungsregler“ beschriebenen Punkten wird auf ein Netzteil mit Schaltregler umgestellt.
04.09.2007	Aufgrund des Prototypenaufbaus wird die Entscheidung getroffen, das Schaltnetzteil in einer separaten PCB auszulagern und aufzubauen. Der Übergang vom Netzteil zur Hauptplatine erfolgt über Stiftbuchsen und Stiftleisten.
11.12.2007	Bei der Inbetriebnahme des Dimmers kam es zu Flackeffekten bei der Dimmung. Ursache hierfür war die Ausgabe von print-Informationen auf der RS232 als auch die Vergabe eines niedrig prioren Interrupts für die Zündung des Triac. Hierzu wurde Timer0 und Timer2 vertauscht. Die PWM des LCD-Displax läuft nun auf Timer0 und die Triac-Zündung auf Timer2. Sie ist damit höher Prior.
12.12.2007	Während der Dimmer arbeitet ist der Empfang des DCF77-Uhrenmoduls zu deaktivieren. Da es sonst zu zeitlichen Verschiebungen bei der Nulldurchgangserkennung kommen kann.
14.12.2007	Um da System bzgl. Interrupts, SW-Timer und ressourcen zu entlasten wird die Zeitrechnung für die Uhrzeit nicht in SW ausgeführt sondern es wird hiermit der Einsatz eines RTC-Bausteins (DS1307) festgelegt. Die Dokumentation wird ergänzt und dahingehend abgeändert.
22.12.2007	Das interne EEPROM des AVR kann nur bis zu 100.000 beschrieben werden. Bei einem Speicherrhythmus von 15 Minuten für die Umweltdaten wird die Grenze bereits schon nach 1041 Tagen (knapp 3 Jahre) erreicht. Dieser Zeitraum ist für eine Systemauslegung zu klein. Hiermit wird die Entscheidung getroffen, dass die Speicherung in Zusammenhang mit den Logging-Daten nur im externen EEPROM erfolgt. Die Setting-Daten only werden im internen EEPROM abgelegt.
14.01.2008	Auf die Erzeugung eines Piezo-Signaltones via PWM wird zugunsten der Lüftersteuerung verzichtet. Es kommtm die in Kapitel „Ansteuerung des Piezo-Schallwandlers (Schwingkreis)“ beschriebene Endstufe zum Einsatz welche über einen einzelnen GPIO-Pin gesteuert werden-kann ohne zusätzlich Ressourcen zu verbrauchen. Es erscheint vernünftiger den Lüfter über PWM am System anzubinden und verschiedene Lüfterstufen zu realisieren.
17.01.2008	Die bisherige Implementierung der PWM für die Displayhelligkeit mit Timer 0 wird umgestellt auf PWM mit Timer 3. Der Timer3 ist ein Mehrfachtimer mit dessen Hilfe mehrere PWM erzeugt werden können. LCD liegt auf OC3A. Gleichzeitig wird der gleiche Timer3 auf OC3B für das PWM-Signal der Lüftergeschwindigkeitregelung verwendet.

Tabelle 1: Festlegungen und Entscheidungen

7.4 Gesamt-Checkliste für Funktionen, Eigenschaften und Features

7.4.1 Allgemeines zur HW-Umsetzung

ID	*Funktion*	*Ressource*
HW001	Lösung in 3 getrennen EURO-PCB für: 1. µC Board / Steuerung PCB1 (Kontroller) 2. Display und Taststur (Bedienelemente und Anzeigen) PCB2 (Konsole) 3. Schaltnetzteil PCB3 (Power) *[Leiterplattenverbindungen]*	--
HW002	Verbindung zwischen PCB1 und PCB2 über PIN-Header und Flachbandkabel *[Leiterplattenverbindung Kontroller <-> Konsole]*	--
HW003	Position Lichtsensor auf Front-PCB2	--
HW004	Verbindung zwischen PCM1 und PCB3 erfolgt über Stiftleisten und Stiftbuchsen *[Leiterplattenverbindung Kontroller <-> Power]*	--

Tabelle 2: Detailfunktionen Spannungsversorgung

7.4.2 Spannungsversorgung für +5V und +12V

ID	*Funktion*	*Ressource*
SUP001	Spannungsversorgung 230V	--
SUP002	Feinsicherung 160 mA flink	--
SUP003	Hauptschalter (Kippschalter) mit LED Beleuchtet *[Hauptschalter]*	--
SUP004	LED Vorschaltplatine für Betrieb an 230V AC (Umsetzung auf PCB3) *[LED Vorschaltmodul für Hauptschalter]*	--
SUP005	Umsetzung Schaltnetzteil auf separater PCB3	--
SUP006	Zwei LED's für 5V und 12V zeigen das Vorhandensein der Betriebsspannung an. *LED1 und LED2 werden auf PCB3 untergebracht.*	LED1 LED2

Tabelle 3: Detailfunktionen Spannungsversorgung

Dokumentenlink Beschreibung: *[Netzteil]*

7.4.3 Mikrocontroller ATmega32 oder ATmega128

(inkl. IPS, RS232 und Reset)

ID	*Funktion*	*Ressource*
µC001	Einsatz des ATmega128	--
µC002	Betrieb des ATmega mit externem 16 MHz Quarz	XTAL1 XTAL2
µC003	Traces sollen über Jumper ein- und ausgeschaltet werden können *(1 IN-Pin)*	TRACE_ON
µC004	ISP-Schnittstelle (IP12) auf parallele Schnittstelle zur Onboard-Programmierung *[ISP-Schnittstelle / ISP-Programmierung]*	MISO MOSI SCK
µC005	Entwurf eines parallelen ISP-Prommers für Target (BASCOM + PonyProg)	--
µC006	Umsetzung der ISP-Schnittstelle über 10-poliges ATMEL Interface	--
µC007	RS232-Schnittstelle über MAX232 für serielles Logging und Traces	RxD TxD
µC008	Spannungsversorgung des ATmega32 oder ATmega128 mit +5V	VCC GND
µC009	Aktivierung und Betrieb des internen Watchdog	--
µC010	Reset über Programmierinterface	RESET
µC011	Reset über externen Taster (auf PCB2)	RESET
µC012	JTAG-Schnittstelle für On_Chip-Debugging *[Das JTEG Interface]*	TDI TDO TMS TCK
µC013	Die Bereitschaft des ATmega soll mit einer Low Current LED angezeigt werden. *!! Software !!*	LED3

Tabelle 4: Detailfunktionen Mikrocontroller

Dokumentenlink Beschreibung: *[Mikrokontroller ATmega32]*

7.4.4 LCD-Display 2x16 oder 3x16

(inkl. Rahmen mit "Scheibe")

ID	*Funktion*	*Ressource*
LCD001	Verwendung eines 2-zeiligen oder 3-zeiligen DOG-M LCD Displays	--
LCD002	Regelung von Kontrast und Helligkeit über Settings *!! Software !!*	--
LCD003	Persistente Speicherung der Einstellungsdaten in internem EEPROM	--

	!! Software !!	
LCD004	Bildschirmschoner (Deaktivierung durch beliebige Taste) zweistufig: Stufe1: Anzeige von Name und Uhrzeit Stufe2: LCD-Display aus *!! Software !!*	--
LCD005	LCD-Frontrahmen für Gehäuse	--
LCD006	Beleuchtung des Display durch weiße Hintergrundbeleuchtung	--
LCD007	Ansteuerung des Display über 4-Bit-Interface *(4 OUT-Pins)*	LCD_BIT0 LCD_BIT1 LCD_BIT2 LCD_BIT3
LCD008	Steuerung des LCD-Displays vom ATmega (Anschluss E und RS) *(2 OUT-Pins)*	LCD_E LCD_RS
LCD009	Realisierung einer PWM über Timer für Helligkeitseinstellung *Realisierung über Timer3*	OC3A *(Timer3)*
LCD010	Display soll vom ATmega aus- bzw. eingeschaltet werden können. *(1 OUT-Pin)* *!! Software !!*	LCD_ON

Tabelle 5: Detailfunktionen LCD-Display

Dokumentenlink Beschreibung: *[LCD-Display]*

7.4.5 Tastatur und Schalter

ID	*Funktion*	*Ressource*
SW001	LED's an Tasten signalisieren Datenübernahme (Reaktion) von Controller (Realisierung über LED-Matrix) *!! Software !!*	LED4 bis LED12
SW002	Hauptschalter für Spannungsversorgung *[Hauptschalter]*	--
SW003	Reset-Taster von außen zugänglich (Mikrotaster mit Loch in Frotplatte) *Der Reset-Taster wird auf PCB2 untergebracht.* *[Reset-Taster]*	RESET
SW004	Tasten für Dimmung [heller / dunkler] *[Tastatur und Tastenmatrix]*	2 Tasten *[Taste_2]*
SW005	Tasten für Beschattung [auf / zu]	2 Tasten *[Taste_1]*
SW006	Stoppen der Beschattungsantriebe durch Betätigen beider Beschattungstaster oder durch Betätigung von Enter. *[Tastatur und Tastenmatrix]* *!! Software !!*	--
SW007	Licht aus durch Betätigen beider Dimmungs-Taster	--
SW008	Tasten für Menüsteuerung [UP] [DOWN] [LEFT] [RIGHT] [ENTER] *[Tastatur und Tastenmatrix]*	5 Tasten *[Taste_3]* *[Taste_4]*
SW009	Langes Betätigen von [ENTER] führt aus jedem Menü zum Rücksprung in die Hauptmenüebene 1. Eintrag.	--
SW010	Umsetzung der Tastatur über 3x3 Tastenmatrix *(3 IN-Pins)* *(3 OUT-Pins)* *[Tastatur und Tastenmatrix]* *!! Software !!*	SW_IN_1 SW_IN_2 SW_IN_3 SW_OUT_1 SW_OUT_2 SW_OUT_3 INT7
SW011	Der Interrupt-Status der Tastatur soll über eine eigene Low Current LED angezeigt werden	LED31

Tabelle 6: Detailfunktionen Tastatur und Schalter

7.4.6 LED-Anzeigen für Status- und Funktionsanzeige

ID	*Funktion*	*Ressource*
LED001	Alive LED für ATmega das Applikation läuft (low current) *(1 OUT-Pin)*	ALIVE LED3
LED002	Spannungsversorgung 5V present (low current) *[Leuchtdioden]*	LED1
LED003	Spannungsversorgung 12V present (low current) *[Leuchtdioden]*	LED2
LED004	Relais Beschattung AUF angezogen (low current) *[Gesamtschaltplan für Relaisansteuerung]*	LED21
LED005	Relais Beschattung ZU angezogen (low current) *[Gesamtschaltplan für Relaisansteuerung]*	LED22
LED006	Relais für Lichtfunktion angezogen (low current) *[Gesamtschaltplan für Relaisansteuerung]*	LED23
LED007	Lüfter via SW zugeschaltet (low current) *[Lüftersteuerung für Gehäusebelüftung]*	LED25
LED008	Lüfter eingeschaltet (low current) *[Lüftersteuerung für Gehäusebelüftung]*	LED26
LED009	Umsetzung aller Status LED's mit low current LED's	--
LED010	Timer-Tick für Sekunden-Takt (low current) *(1 OUT-Pin + Timer3)* *!! Software !!*	SEC_TICK *(Timer3)* LED27
LED011	Impulseingang für Wind-Sensor *[Windgeber]*	LED28
LED012	Eingang Regensensor (Statusanzeige) *[Beschaltung Regensensor]*	LED29
LED013	LED's für jede Taste integriert in Taster 1-9 *[Latched LED Matrix]*	9 LED's: LED4 LED5 LED6 LED7 LED8 LED9 LED10 LED11 LED12
LED014	LED wenn Datenspeicher voll *[Latched LED Matrix]*	1 LED: LED13
LED015	LED's sollen im Gehäuse sichtbar angebracht sein	--
LED016	Anzeige des automatischen Auffahrgrundes für Beschattung: - Zeitsteuerung - Licht - Regen - Wind *[Latched LED Matrix]*	4 LED's: LED14 LED15 LED16 LED17
LED017	Anzeige des automatischen Zufahrgrundes für Beschattung: - Zeitsteuerung - Temperatur *[Latched LED Matrix]*	2 LED's: LED18 LED19
LED018	Entwicklung eines Konzeptes zur PIN schonenden Ansteuerung von LED's in MUX and Latch-Technik!	--
LED019	LED-Anzeige für Heizung AN/AUS über separaten Schalter	LED20
LED020	Für die Funktionen LED013 LED014 LED016	DATA_0 DATA_1 DATA_2

	LED017 wird eine ein gelatchtes LED-Array aufgebaut um IO-Pins zu sparen *(7 OUT-Pins)* *[Latched LED Matrix]*	DATA_3 ADR_0 ADR_1 CLEAR
LED021	Relais für Dauer-Licht angezogen (low current) *[Gesamtschaltplan für Relaisansteuerung]*	LED24
LED022	Der Impuls der DCF77 Uhr soll über eine Low Current LED visuell angezeigt werden	LED30

Tabelle 7: Detailfunktionen LED-Anzeigen

Dokumentenlink Beschreibung: *[Latched LED Matrix]*
Dokumentenlink Beschreibung: *[LED-Zuordnung Schaltplan]*

7.4.7 Windmesser

(pro Tick = 2,5m/sec Windgeschwindigkeit)

ID	*Funktion*	*Ressource*
WIND001	Anschluss Windgeber an µC (Umsetzung via Reedkontakt)	--
WIND002	Anschluss über +5V -> Windgeber -> InPort (Interrupt)	--
WIND003	Ggf. Entprellung von Eingangssignalen (50 µsec)	--
WIND004	Entprellung über Schmitt-Trigger-Stufe	--
WIND005	Umsetzung SW Entprellung über Interrupt-Routine „single shot“ *!! Software !!*	--
WIND006	Portfestlegung für externen Interrupt -> Impulszähler	INT6
WIND007	Signalauswertung alle 5 Sekunden *-> Steuerung über internen Timer bzw. SW-Timer, abgeleitet aus Timer3* *!! Software !!*	Timer3 Systemzeit
WIND008	Auswerteintervall für Windgeschwindigkeitberechnung über Setting einstellbar *!! Software !!*	--
WIND009	Persistente Speicherung der Settings in internem EEPROM	--
WIND0010	Anzeige aktuelle Windgeschwindigkeit in Windmenü	--
WIND0011	Anzeige Impuls Windsensor mit low current LED auf Konsolen-Board	LED28

Tabelle 8: Detailfunktionen Windmessung

Dokumentenlink Beschreibung: *[Windgeber]*

7.4.8 Regensensor

(ja/nein Detektion inkl. Detektorheizung)

ID	*Funktion*	*Ressource*
RAIN001	Betrieb des Regensensors an 12V	--
RAIN002	Regensensor steuert OUT-PIN bei Regen (Beschaltung über Spannungsteiler)	--
RAIN003	Einlesen der 12V Schaltspannung über Spannungsteiler und Schmitt-Trigger *(1 IN-Pin)*	RAIN
RAIN004	Signalauswertung jede Sekunde über Timer *-> Steuerung über internen Timer bzw. SW-Timer, abgeleitet aus Timer3*	Timer3 (Systemzeit)
RAIN005	Anzeige des aktuellen Regen-Status über Regen-Menü *!! Software !!*	--
RAIN006	Regendetektion blockiert manuelle und zeitlich gesteuerte Beschattung	--
RAIN007	Geschlossene Beschattung fährt bei erkanntem Regen automatisch auf	--
RAIN008	Heizung soll separat von Steuerung aus zugeschaltet werden können	--
RAIN009	Anzeige aktiver Regenstatus mit low current LED auf Konsolen Board	LED29
RAIN010	Anzeige ob Heizung des Regensensor eingeschaltet oder nicht	LED20

Tabelle 9: Detailfunktionen Regensensorik

Dokumentenlink Beschreibung: *[Beschaltung Regensensor]*

7.4.9 Temperatursensor

(Anschluss über two wire interface)

ID	*Funktion*	*Ressource*
TEMP001	Anschluss des Sensors über proprietäres two wire interface (Anschluss über zwei IO-Pins des ATmega) *(1 IN-Pin + 1 OUT-Pin)*	TEMP_DATA TEMP_SCK
TEMP002	Bestriebsspannung von 5V für Sensor bereitstellen	--
TEMP003	Sensor Platzierung auf PCB1 µC-Board	--
TEMP004	Sensor so in Luftstrom, dass angesaugte Luft direkt über Sensor streichen kann	--
TEMP005	Sensor alle 5 Sekunden über Timer auslesen *-> Steuerung über internen Timer bzw. SW-Timer, abgeleitet aus Timer3* *!! Software !!*	Timer3
TEMP006	Auswertungsintervall über Settings einstellbar *!! Software !!*	--
TEMP007	Anzeige der Temperatur über Menü Temperatur	--
TEMP008	Persistente Speicherung der Settingseinstellung	--

Tabelle 10: Detailfunktionen Temperatursenorik

Dokumentenlink Beschreibung: *[Sensorik für Temperatur und Luftfeuchtigkeit]*

7.4.10 Luftfeuchtigkeit

(Anschluss über two wire interface)

ID	*Funktion*	*Ressource*
HUM001	Anschluss des Sensors über proprietäres two wire interface (Anschluss über zwei IO-Pins des ATmega) *(1 IN-Pin + 1 OUT-Pin)*	TEMP_DATA TEMP_SCK
HUM002	Bestriebsspannung von 5V für Sensor bereitstellen *!! Software !!*	--
HUM003	Sensor Platzierung auf PCB1 µC-Board	--
HUM004	Sensor so in Luftstrom, dass angesaugte Luft direkt über Sensor streichen kann	--
HUM005	Sensor alle 5 Sekunden über Timer auslesen *-> Steuerung über internen Timer bzw. SW-Timer, abgeleitet aus Timer3*	Timer3 (Systemzeit)
HUM006	Auswertungsintervall über Settings einstellbar	--
HUM007	Anzeige der Luftfeuchtigkeit über Menü Luftfeuchtigkeit	--
HUM008	Persistente Speicherung der Settingseinstellungen	--

Tabelle 11: Detailfunktionen Luftfeuchtigkeit

Dokumentenlink Beschreibung: *[Sensorik für Temperatur und Luftfeuchtigkeit]*

7.4.11 Dimmungssteuerung für Beleuchtung

(inkl. EIN/AUS Relais)

ID	*Funktion*	*Ressource*
DIM001	Dimmungssteuerung mittels Triac für Wintergartenbeleuchtung *(1 OUT-Pin)*	Timer2 TRIAC
DIM002	2 Taster für Funktionen [heller / dunkler / an / aus] *(2 IN-Pins (in Kapitel tastatur berücksichtigt!))*	2 Tasten (Matrix)
DIM003	Tastverhältnis für Dimmung auch über Dimmer-Menü einstellbare *!! Software !!*	--
DIM004	Anzeige des aktuell eingestellten Tastverhältnisses über LCD-Display (Balken)	--
DIM005	Settings [min / max] für Tastverhältnis Hysterese über Settings einstellbar *!! Software !!*	--
DIM006	Letzte eingestellte Helligkeit wird persistent in internem EEPROM gespeichert	--

DIM007	Tasten für Dimmung entprellt umgesetzt	--
DIM008	Tastenfunktionen: [UP] heller, wenn AUS -> EIN [DOWN] dunkler, wenn AUS -> EIN [UP+DOWN] Licht AUS	--
DIM009	Absicherung der Lampensteuerung durch eigene Feinsicherung	--
DIM010	EIN/AUS der gesamten Dimmungssteuerung über eigenes Steuerrelais und geeignete Software-Entkopplung. *(1 OUT-Pin)*	DIMMUNG
DIM011	Nulldurchgangserkennung für Triac-Auslösung	INT5
DIM012	Anzeige ob Relais angezogen hat über Status low current LED auf Konsolen-Board	LED23
DIM013	Über ein eigenes weiteres Steuerrelais kann eingestellt werden ob das Licht dauerhaft leuchtet oder ob der Dimmer verwendet werden soll. Anzeige über Status-LED *(1 OUT-Pin)*	CON_LUM LED24

Tabelle 12: Detailfunktionen Dimmersteuerung

Dokumentenlink Beschreibung: *[Lichtdimmung]*

7.4.12 Motorsteuerung für Beschattung

(inkl. Relaisverriegelung und Relais-Timeout)

ID	*Funktion*	*Ressource*
CONV001	Absicherung der Motorsteuerung durch eigene Feinsicherung 3,15A mittelträge	--
CONV002	Richtungssteuerung der Motoren durch 2 getrennte Relais mit gegenseitiger HW-Verriegelung. *(2 OUT-Pins)*	CONV_AUF CONV_ZU
CONV003	Statusanzeige über LED's (low current)	--
CONV004	2 Taster für Funktionen [auf / zu / stop] *(2 IN-Pins (in Kapitel tastatur berücksichtigt!))*	2 Tasten (Matrix)
CONV005	Realisierung eines Timeout für automatischen AUF / ZU / STOP *-> Steuerung über internen Timer bzw. SW-Timer, abgeleitet aus Timer3*	Timer3 (Systemzeit)
CONV006	Timeout einstellbar über Settings	--
CONV007	Letzte eingestellte Settings persistent in internem EEPROM gespeichert	--
CONV008	Zeitmenüsteuerung für zeitlich gesteuertes zufahren / auffahren	--
CONV009	Automatisches Auffahren bei Dunkelheit → In Settings einstellbar. *!! Software !!*	--
CONV010	Automatisches Auffahren bei Regen → In Settings einstellbar. *!! Software !!*	--
CONV011	Automatisches Auffahren bei Wind → In Settings einstellbar. *!! Software !!*	--
CONV012	Windgrenzen einstellbar über Settings	--
CONV013	Optionales Zufahren bei zu größer Hitze	--
CONV014	Funktion und Temperaturschwelle einstellbar über Settings	--
CONV015	Abschaltung der Kreuztasten HotKeys über Settings	--
CONV016	Anzeige ob Relais AUF angezogen hat über low current LED auf Konsolen-Board	LED21
CONV017	Anzeige ob Relais ZU angezogen hat über low current LED auf Konsolen-Board	LED22

Tabelle 13: Detailfunktionen Beschattung

Dokumentenlink Beschreibung: *[Steuerung Beschattungsanlage Motorantriebe]*

7.4.13 Gehäusebelüftung mit Option Lüftersteuerung über ATmega

ID	*Funktion*	*Ressource*
FREEZ001	Das Gehäuse der Steuerelektronik wird aktiv durch einen Lüfter gekühlt	--
FREEZ002	Die Belüftungsrichtung erfolgt von unten nach oben	--
FREEZ003	Ggf. Belüftung des Gehäuses aktiv durch Auswertung Temperatursensor	--

FREEZ004	Steuerung der Lüftung durch ATmega *(1 OUT-Pin)*	FAN
FREEZ005	Über Jumper soll einstellbar sein ob Dauerplus oder µC-Control	--
FREEZ006	Laufende Belüftung soll durch LED angezeigt werden. - Lüftung via SW zugeschaltet - Lüftung via Dauerstrom eingeschaltet	 LED25 LED26
FREEZ007	ATmega 5V IO steuert 12V Last über Transistorstufe	--
FREEZ008	In Settings einstellbare ob Lüfter aktiv oder nicht	--
FREEZ009	Persistente Datenspeicherung der Einstellungen im internem EEPROM *!! Software !!*	--
FREEZ010	PWM für Geschwindigkeitsregelung	Timer3 (OC3B)

Tabelle 14: Detailfunktionen Kühlung

Dokumentenlink Beschreibung: *[Lüftersteuerung für Gehäusebelüftung]*

7.4.14 Datenspeicher / Datenlogger über EEPROM

ID	*Funktion*	*Ressource*
LOG001	Initialisierung des internen und externen EEPROM's auf Startwerte	--
LOG002	Kennzeichnung dass Default-Startwerte gesetzt mit Generationszähler	--
LOG003	Verwendung des internen EEPROM (1024 Byte) für Einstellungen / Settings	--
LOG004	Verwendung eines externen EEPROM für Datenlogging (Messwerte + Zeit + Datum)	--
LOG005	Messwerte aus externem EEPROM können über RS232 als Text ausgegeben werden (z.B. für Auswertung in Excel)	--
LOG006	Anschluss des externen EEPROM über I2C (extern)	SDA SCL
LOG007	Definition eines geeigneten Datenformats zur Datenablage	--
LOG008	Speichern von: Datum+Zeit+Wind+Feuchtigkeit+Temperatur+Regen+Licht	--
LOG009	Anzeige Speicher voll via Status LED (LED014) → Abhängig vom Overrun-Index	LED13

Tabelle 15: Detailfunktionen Datenspeicherung

Dokumentenlink Beschreibung: *[EEPROM]*

7.4.15 DCF77 Funkuhrempfang (Conrad)

ID	*Funktion*	*Ressource*
DCF001	Einsatz des DCF77-Moduls von Conrad an µC	DCF77
DCF002	Interne Uhr und Datum werden mit DCF77 synchronisiert	--
DCF003	Datum und Uhrzeit sollen im Menü Zeit angezeigt werden können	--
DCF004	Ggf. Uhrzeit auch über Bildschirmschoner	--
DCF005	Keine DCF77-Wandlung während Dimmer aktiv	--
DCF006	Der Impuls der DCF77 Uhr soll über eine Low Current LED visuell angezeigt werden	LED30
DCF007	Aus Empfangs- und Befestigungs-Gründen wird der DCF77-Empfänger auf dem Konsolen-Board PCB1 angebracht und befestigt. Das durch Schmitt-Trigger aufbereitete Signal wird über den Pfostenstecker auf Board PCB2 geführt.	--

Tabelle 16: Detailfunktionen DCF77 Funkuhr

Dokumentenlink Beschreibung: *[Beschaltung DCF77-Modul]*

7.4.16 RTC – Real-Time-Clock

ID	*Funktion*	*Ressource*
RTC001	Initialisierung der RTC aus Startwerte über DCF77	--

RTC002	Interne Uhr und Datum werden mit DCF77 synchronisiert	--
RTC003	Aktualisierung der RTC bei Systemstart und jede Stunde (optional alle 24h) wenn kein Dimmer aktiv	--
RTC004	Anschluss der externen RTC über I2C (extern)	SDA SCL
RTC005	Einsatz des Baustein DS1307 der auch als Referenz von BASCOM unterstützt wird.	

Tabelle 17: Detailfunktionen Real-Time-Clock

Dokumentenlink Beschreibung: *[RTC – Real-Time-Clock]*

7.4.17 Passendes Selbstbaugehäuse aus Aluminium und rauchgläserner Front

ID	*Funktion*	*Ressource*
BOX001	Gehäusedeckel aus Plexiglas ausgeführt wegen Optik	--
BOX002	Belüftung von unten nach oben	--
BOX003	Gehäuseausführung aus Aluminuium	--
BOX004	Kabeldurchführungen durch Gehäusewand oben und unten	--
BOX005	Schalter für Regensensorheizung in Gehäuse	--
BOX006	Hauptschalter in Gehäuse	--

Tabelle 18: Detailfunktionen Gehäuse

Dokumentenlink Beschreibung: *[Mechanik]*

7.4.18 Helligkeitsmessung

(Licht -> DC-Wandler mit OPAmp)

ID	*Funktion*	*Ressource*
LIGHT001	Helligkeitsmessung mittels TAOS Licht->Spannungs-Umsetzer	--
LIGHT002	„Aufspreizung“ des Messbereichs mittels PräzisionsOpAmp AD623	--
LIGHT003	Messung / Quantisierung über ADC	ADC0
LIGHT004	Automatisches Auffahren der Beschattung bei Dunkelheit (Einstellbar über Setting)	--
LIGHT005	Anzeige der gemessenen Helligkeit im Menü „Licht“	--
LIGHT006	Schwelle „Auffahren“ in Settings einstellbar. 0 bedeutet Funktion deaktiv.	--
LIGHT007	Peristente Speicherung der Einstellwerte in internem EEPROM	--
LIGHT008	Die Beschaltung der Lichtsensorik wird auf PCB3 umgesetzt und mittels Signalleitung über Steckverbinder zum ATmega geführt	ADC0

Tabelle 19: Detailfunktionen Helligkeitssensorik

Dokumentenlink Beschreibung: *[Lichtsensorik]*

7.4.19 Anzeige von Zeit- und Datumsinformationen

ID	*Funktion*	*Ressource*
CLOCK001	Anzeiger der Zeitinformation von DCF77 Uhr auf Display in Menü Zeit und Datum	--
CLOCK002	Anzeige der Datumsinformation auf Display in Menü Zeit und Datum	--
CLOCK003	Ggf. Anzeige von Uhrzeit und Datum als Bildschirmschoner	--

Tabelle 20: Detailfunktionen Zeit und Datumsinformation

7.4.20 Akustisches Feedback über Piezo Schallwandler

ID	*Funktion*	*Ressource*

SOUND001	Ansteuerung eines Piezo Schallwandlers	--
SOUND002	Entwicklung eines durch den ATmega gesteuerten Schwingkreises *(1 OUT-Pin)* *[Ansteuerung des Piezo-Schallwandlers]*	PEEP
SOUND003	Piepsen bei Auffahren und Zufahren	--
SOUND004	Geräusche und Töne sollen für die einzelnen Funktionen in Settings einstelbar sein	--
SOUND005	Persistente Speicherung der Settings	--
SOUND006	Frequenzeinstellung über Potentiometer	--
SOUND007	Bei Einsatz von ATmega128 Ansteuerung Piezo-Schallwandler über PWM Timer 3 oder über Schwingkreis wie bei SOUND002 *(1 Timer oder 1 GPIO-Pin)* *[Ansteuerung eines Piezo-Schallwandlers mittels PWM]* *[Ansteuerung des Piezo-Schallwandlers]*	SOUND *(Timer3)* oder GPIO

Tabelle 21: Detailfunktionen Akustik

Dokumentenlink Beschreibung: *[Ansteuerung des Piezo-Schallwandlers]*

7.4.21 Selbstcheck bei StartUp und Warnmeldungen

ID	*Funktion*	*Ressource*
CHECK001	Anzeige während Selbstcheck und Initialisierungsphase den Hochlauf auf Display	--
CHECK002	Erkannte Fehler und Fehlerzustände im System sollen abgefangen und zyklisch auf dem Display angezeigt werden	--
CHECK003	Bei Erkennung eines Fehler soll Warnton ausgegeben werden	--

Tabelle 22: Detailfunktionen CheckUp

7.4.22 Timersteuerung zeitkritischer Applikationen

(Tasten, Dimmer, Helligkeit Display)

ID	*Funktion*	*Ressource*
TIME001	Dimmer	--
TIME002	Timout für Beschattung auf / zu	--
TIME003	Uhrzeit	--
TIME004	Steuerung der Datenablage in externem EEPROM	--
TIME005	Auslesen der Daten aus den Sensoren	--
TIME006	Steuerung von Zeitliche gesteuerten Abläufen (automatisches auf/zu fahren)	--
TIME007	PWM für Displayhelligkeit	--
TIME008	Sekunden-Tick	--
TIME008	Aktivierung des Watchdog Timers und rücksetzen des Timers wegen Resets. *[Der Watchdog-Timer]*	--

Tabelle 23: Detailfunktionen zeitkritische Timerfunktionen

7.5 Projektcheckliste für AVR Systemdesigns

Diese Checkliste beinhaltet eingige grundlegende Regeln beim Design mit AVR Mikrokontrollern.

http://www.mikrocontroller.net/articles/AVR_Checkliste

Dies sind zusammengefasst in Kürze:

7.5.1 Abblockkondensator(en) ordnungsgemäß installiert?

Abblockkondensatoren ("Bunker-Kondensatoren") dienen dazu, sehr kurze Versorgungsspannungseinbrüche, die durch Schaltvorgänge verursacht werden können, zu kompensieren. Diesen Zweck erfüllen sie optimal, wenn folgende Regeln eingehalten werden:

- Ein Abblockkondensator sollte möglichst dicht am IC sitzen.
- Jedes IC in einer Schaltung sollte einen Abblockkondensator besitzen.
- Bei ICs mit mehreren Anschlüssen für VCC und GND sollte jedes VCC-GND-Paar mit einem eigenen Abblockkondensator beschaltet werden (z. B. AVRs in SMD-Bauform wie dem ATmega16A also mit vier Kondensatoren).
- Es sollten keramische Kondensatoren mit einer Kapazität von 100 nF verwendet werden. Größere Kondensatoren, etwa 10 µF-Elkos, sind für diese Aufgabe *nicht* geeignet, weil sie "zu langsam" sind!

7.5.2 Spannungsversorgung richtig angeschlossen?

Der AVCC-Pin ist der Versorgungsanschluss für den AD-Wandler und den zugehörigen Port. Er ist nicht an allen AVRs vorhanden; wenn er aber vorhanden ist, so muss er auf jeden Fall angeschlossen sein, auch wenn der AD-Wandler nicht benutzt wird. Wird der AD-Wandler verwendet, sollte zur Verbesserung der Genauigkeit der AVCC-Pin über einen Lowpass-Filter angeschlossen werden (siehe Datenblatt). Oft funktioniert die Programmierung des Controllers auch, wenn Vcc oder GND nicht richtig angeschlossen ist. Zur Sicherheit kann man mit einem Messgerät direkt an den Anschlüssen des AVRs kontrollieren (VCC-GND, AVCC-GND) prüfen, ob die Verbindungen korrekt sind. Es empfiehlt sich, vor dem Einsetzen bzw. Einlöten des Controllers die Versorgungsanschlüsse nochmals zu prüfen, um sicherzustellen, dass man den IC nicht durch eine zu hohe Spannung aufgrund eines Fehlers in der Versorgung zerstört.

7.5.3 Reset-Pin korrekt beschaltet?

Der Reset-Anschluss am AVR ist 'active-low', d. h. wenn man den Pin mit GND (Masse) verbindet, wird der Controller resettet. Zwar haben AVRs einen internen Pullup-Widerstand, der den Reset-Pin gegen VCC "zieht", dieser ist jedoch relativ hochohmig (ca. 50 kOhm, vgl. Datenblatt) und reicht unter Umständen nicht aus, um den Reset-Pin sicher "hochzuhalten". Als Mindestbeschaltung empfiehlt sich dringend, einen externen Pullup-Widerstand vorzusehen (typisch 10 kOhm), der den Reset-Pin mit VCC verbindet. Er sollte nicht kleiner als 4,7 kOhm sein, da der Programmieradapter sonst eventuell den Reset-Pin während des Programmiervorgangs nicht sicher auf "low" ziehen kann. Zusätzlich sollte man auch noch einen Kondensator 47 nF oder 100 nF zwischen Reset-Pin und GND anordnen. Dieses RC-Glied sorgt dafür, dass der Controller beim Einschalten der Versorgungsspannung für eine definierte Zeitspanne im Reset gehalten wird. Im laufenden Betrieb sorgt der Kondensator dafür, dass der Reseteingang unempfindlich gegenüber Spikes und Glitches wird. Er sollte deshalb unmittelbar in Pin-Nähe beim Prozessor untergebracht werden. Dieser Kondensator darf jedoch nicht verwendet werden, wenn DebugWire möglich sein soll.

Atmel empfiehlt zusätzlich noch zum Schutz vor Überspannungen eine externe Diode nach VCC ("Clamp-Diode"), da für den Reset-Pin keine interne vorhanden ist. Diese Diode bereitet jedoch bei manchen Programmieradaptern Schwierigkeiten.

7.5.4 Alle Ground-Anschlüsse beschaltet?

Bei AVRs mit mehreren Ground-Anschlüssen müssen alle Anschlüsse beschaltet werden.
Siehe

http://www.mikrocontroller.net/forum/read-1-107259.html

7.5.5 Besonderheiten bei ATmega64 / ATmega128

Der ATmega64 und der ATmega128 haben zwei besondere Fallstricke, über die man bei nicht ausreichendem Datenblattstudium leicht stolpert.

- Der erste betrifft den Anschluss der ISP-Signale (MISO, MOSI, SCK, RESET). Dieser erfolgt nicht wie bei den meisten anderen AVR-Controllern an den gleichnamigen Pins, sondern an PDI, PDO, SCK und RE-

SET. Die Pinzuordnung ist: MOSI->PE0 (Pin 2), MISO->PE1 (Pin 3), SCK->PB1 (Pin 11) und RESET->RESET (Pin 20). PEN (Pin 1) hat für normale ISP-Adapter keine Bedeutung und kann offen gelassen oder direkt mit Vcc verbunden werden. Die Benutzung der Pins PDI und PDO anstelle von MOSI und MISO gilt zusätzlich für einige ATMega128-Derivate im 64-Pin-TQFP-Gehäuse (darunter AT90CAN128, ATMega1281/2561)! Im Zweifelsfall im Datenblatt (Pin Configuration) nachsehen, ob PE0 und PE1 mit "PDI" bzw. "PDO" beschriftet sind.

- Die zweite kleine "Gemeinheit" betrifft die M103C-Fuse (Atmega103 Compatibility Mode). Diese ist bei fabrikneuen Atmega64 und Atmega128 gesetzt und sorgt dafür, dass sich diese wie ein Atmega103 verhalten.
  - Andererseits hat dies zur Folge, dass ein für den Atmega64 oder Atmega128 geschriebenes Programm beim ersten "ret" abstürzt, da der SRAM an einer anderen Stelle liegt als erwartet und somit der Prozessor keine gültige Rücksprungadresse vom Stack holen kann.
  - Außerdem funktionieren einige IO-Pins an PORTC, PORTF und PORTG anders.
  - Für weitere Infos bzgl. TWI, UART, Timer, Bootloader und Kalibrierung des internen RC-Oszillators unbedingt das Datenblatt lesen.

7.6 Bedienelemente

7.6.1 Tasten

Die Bedienung des Systems erfolgt über insgesamt 9 Tasten welche über eine LED verfügen. Die LED zeigt die korrekte Übernahme des Signals durch den Controller an. Eine genauer Beschreibung der Tastenmatrix befindet sich im Kapitel „Tastatur-Matrix“ beschrieben.

Das folgende Bild zeigt die Anordnung aller Bedienelemente und des Displays auf dem PCB2:

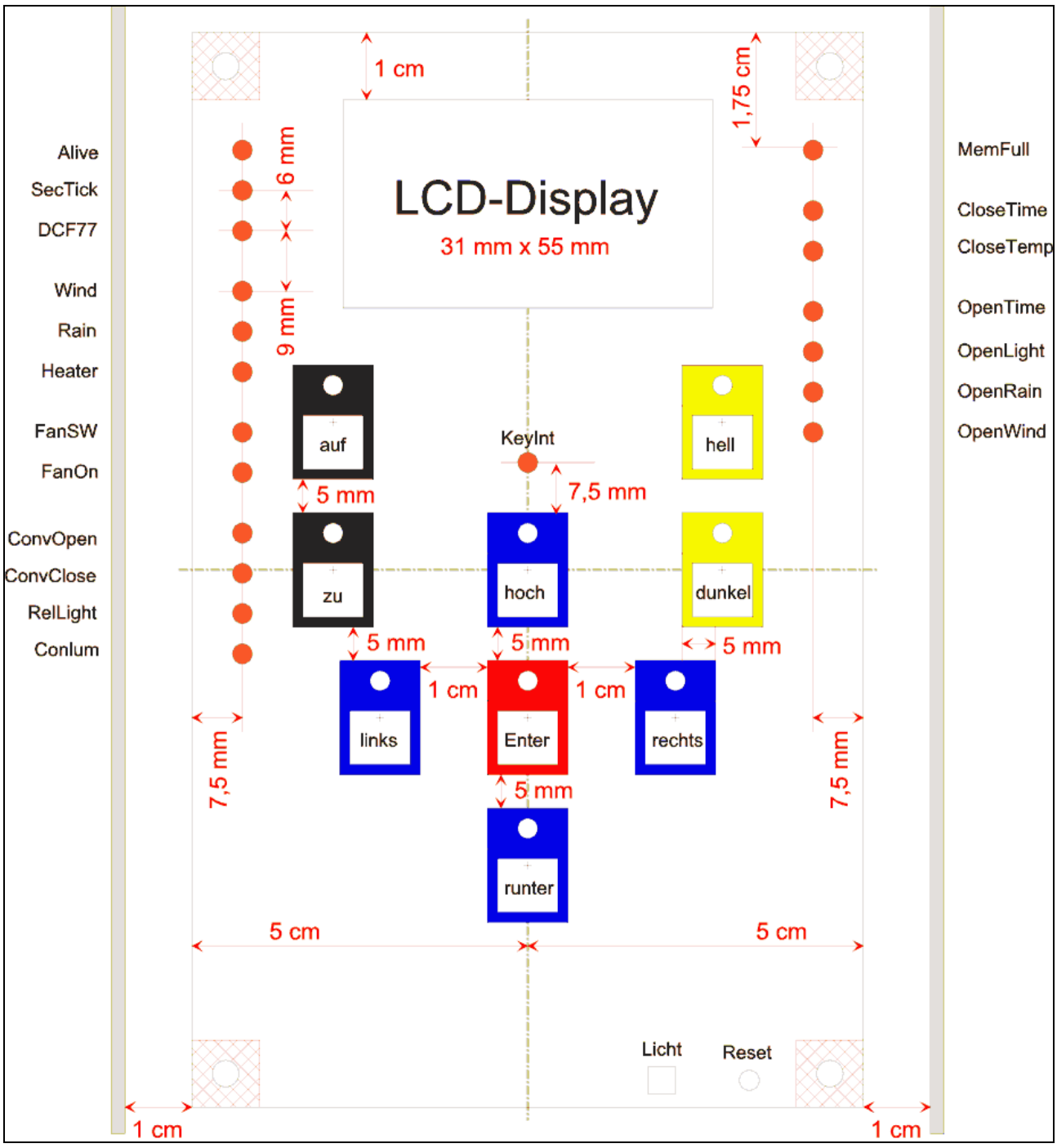


Abbildung 2: Bedienelemente

7.6.2 LCD-Display

Als Anzeigedisplay wird ein LCD-Modul der Firma ELECTRONIC ASSEMBLY der DOG_Serie 3,3V EA DOG-M Super Flach / 55x27 mm inkl. Kontroller ST7036 für 4-/8-BIT SPI (4-Draht) eingesetzt.
Als Hintergrundbeleuchtung wird LED-Beleuchtung Weiß eingesetzt.

Zum Einsatz kommt das folgenden Display:

LCD-Modul 3x16 – 3,65 mm	EA DOGM1634W-A
LED Hintergrundbeleuchtung Weiß	EA LED55X31W

EA DOGM1634W-A: Buchstabenhöhe 3,6 mm

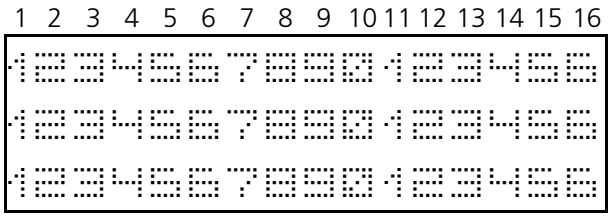


Abbildung 3: LCD 3x16

7.7 Menüstruktur

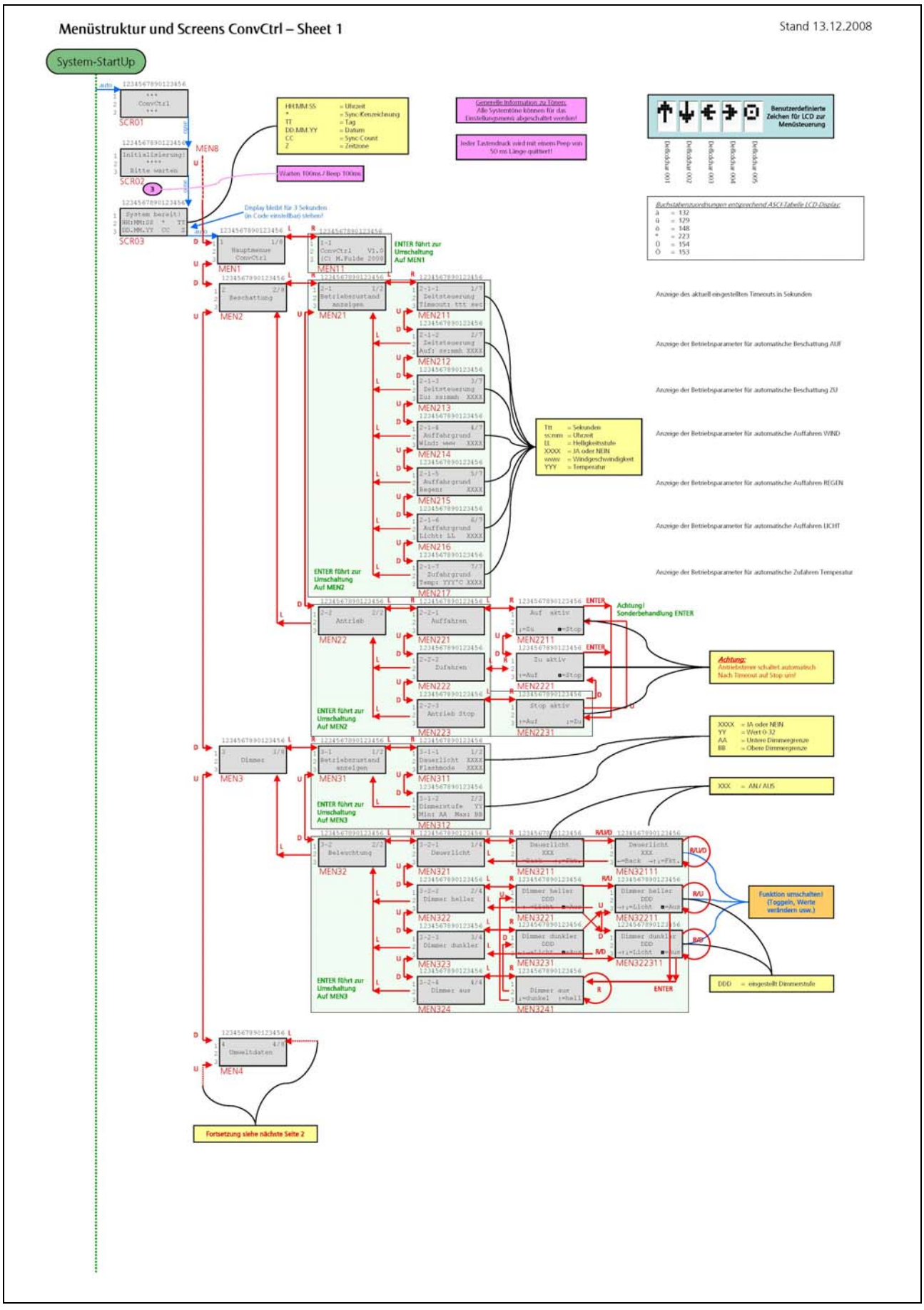


Abbildung 4: Menüstrukturen ConvCtrl – Sheet 1

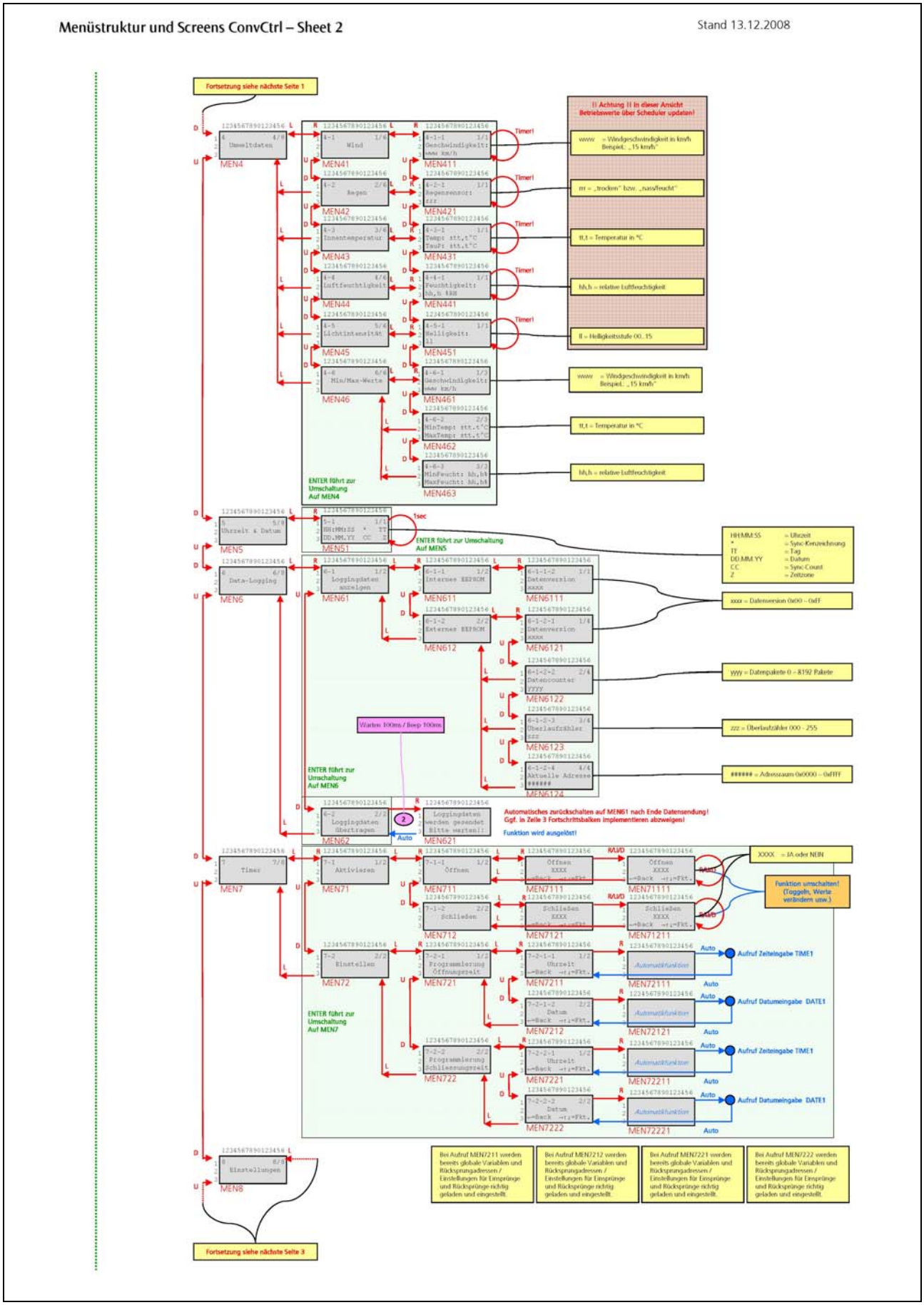


Abbildung 5: Menüstrukturen ConvCtrl – Sheet 2

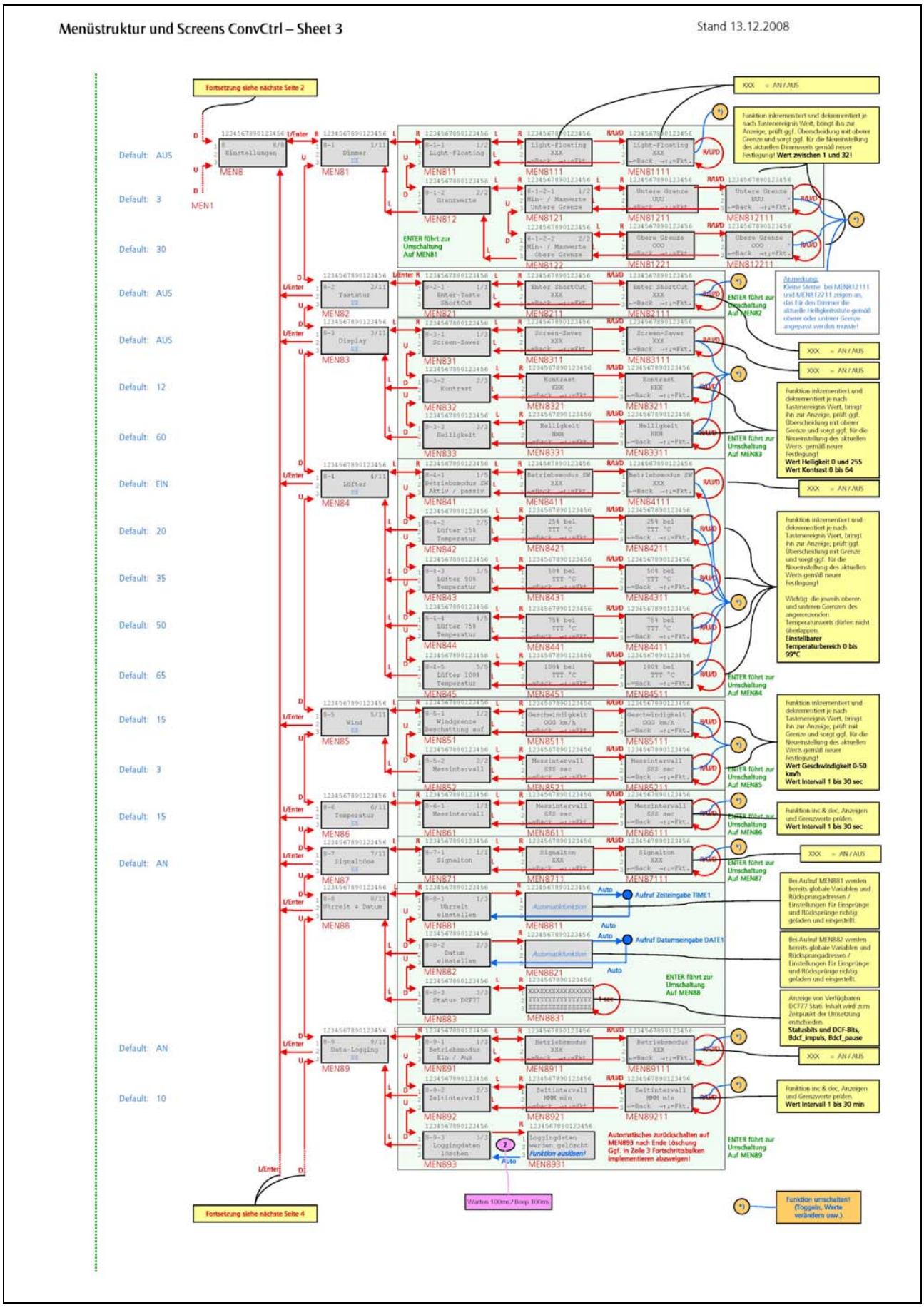


Abbildung 6: Menüstrukturen ConvCtrl – Sheet 3

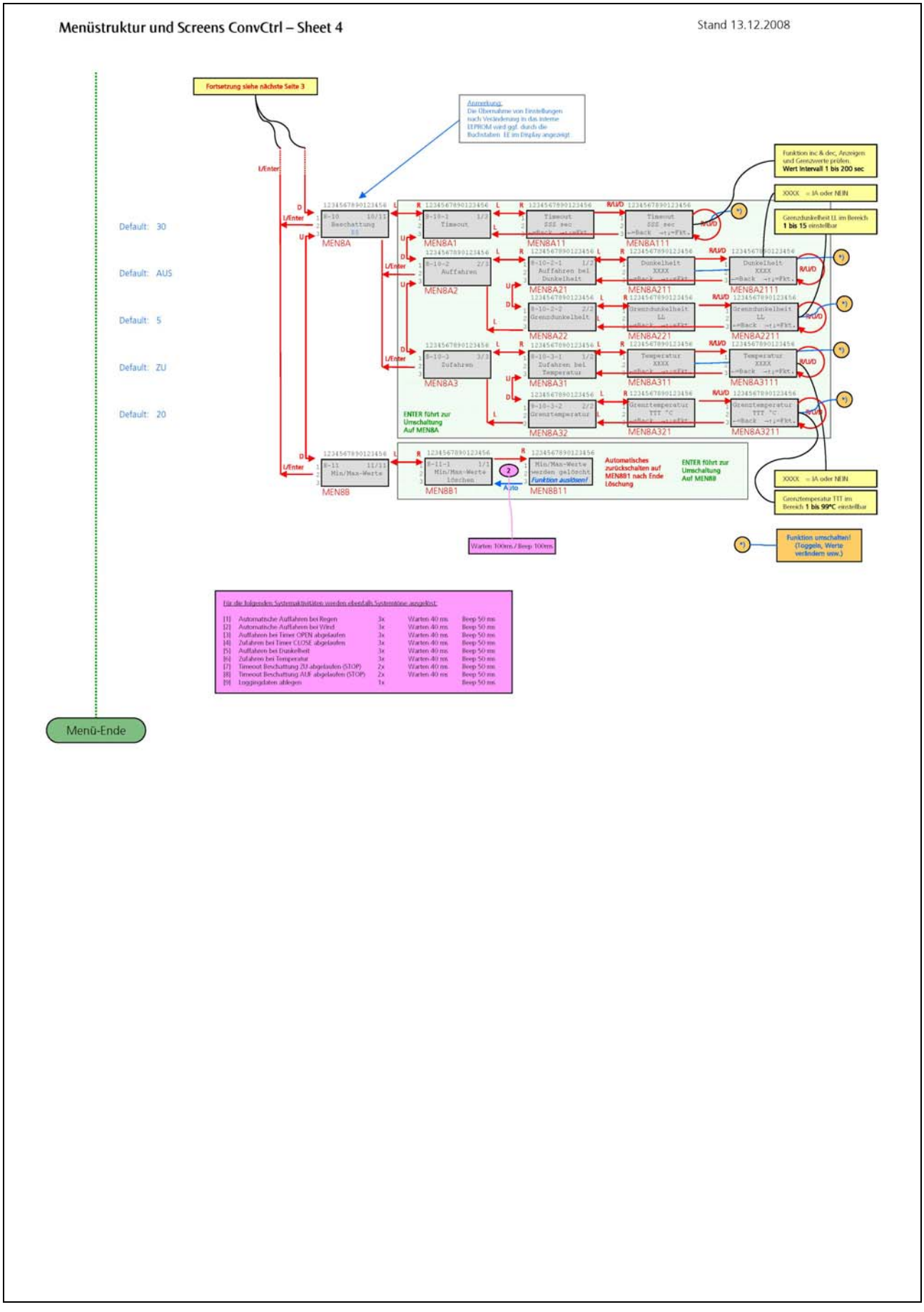
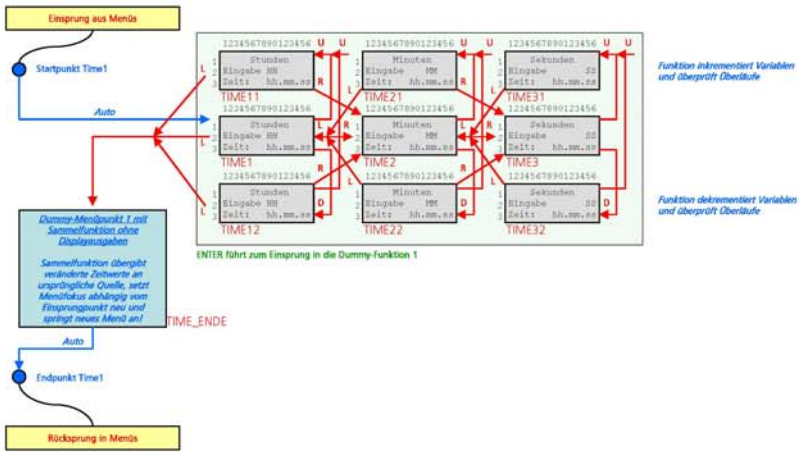


Abbildung 7: Menüstrukturen ConvCtrl – Sheet 4

Menüstruktur und Screens ConvCtrl – Sheet 5 Stand 13.12.2008

Bemerkung:
Die Funktion zum Stellen der Urzeit (Stunde, Minute, Sekunde) wird an mehreren Stellen im System benötigt. Aus diesem Grund wird im System eine generische Menüstruktur implementiert bei der die jeweiligen Einsprünge und die damit verbundenen Variablen zur Laufzeit umgehängt und angesprungen werden.



Bemerkung:
Die Funktion zum Stellen des Datum (Tag, Monat, Jahr) wird an mehreren Stellen im System benötigt. Aus diesem Grund wird im System eine generische Menüstruktur implementiert bei der die jeweiligen Einsprünge und die damit verbundenen Variablen zur Laufzeit umgehängt und angesprungen werden.

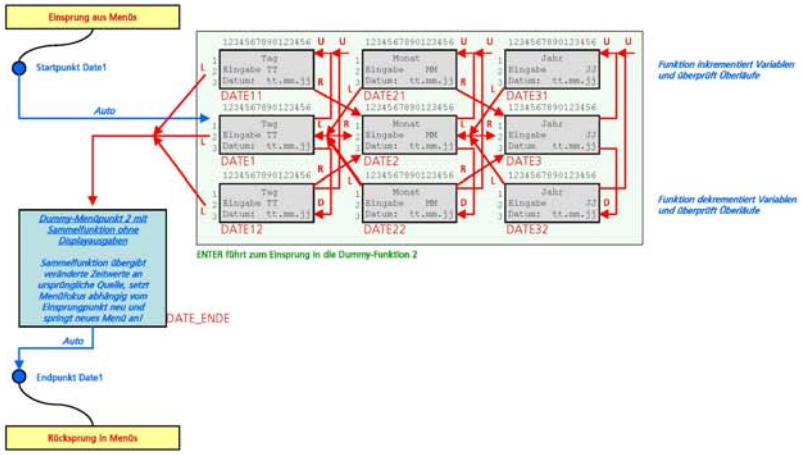


Abbildung 8: Menüstrukturen ConvCtrl – Sheet 5

8 Grundlagen

8.1 Mikrokontroller ATmega32 oder ATmega128

Im Projekt wird der Mikrokontroller ATmega32 oder ATmega128 von ATMEL mit einem externen Quarz 16 MHz eingesetzt.

Zum Zeitpunkt der Konzepterstellung dieses Dokuments steht die Wahl des Prozessors noch nicht fest. Erste Inbetriebnahmen und Versuche bzgl. Projektumsetzung werden mit dem ATmega32 auf dem STK500 von ATMEL realisiert.

Erste Hochrechungen für Codegrößen zeigen aber dass das Flash von 32 KByte nicht für die vollständige Umsetzung des Projekts ausreicht. Somit bleibt an dieser Stelle offen ob später auf den ATmega128 umgestellt wird. Am Grundkonzept ändert sich aber nichts. Es stehen lediglich mehr Funktionen und IO-Pins zur Verfügung welche die Arbeit speziell mit der Ein- und Ausgabe von Signalen (Tastaturmatrix und LED's erleichtert).

PINOut ATmega32:

PINOut ATmega32 PDIP

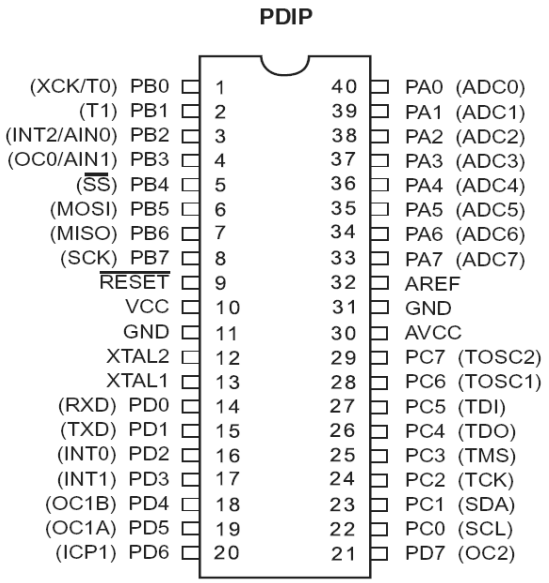


Abbildung 9: PINOut ATmega32 PDIP

PINOut ATmega32 TQFP

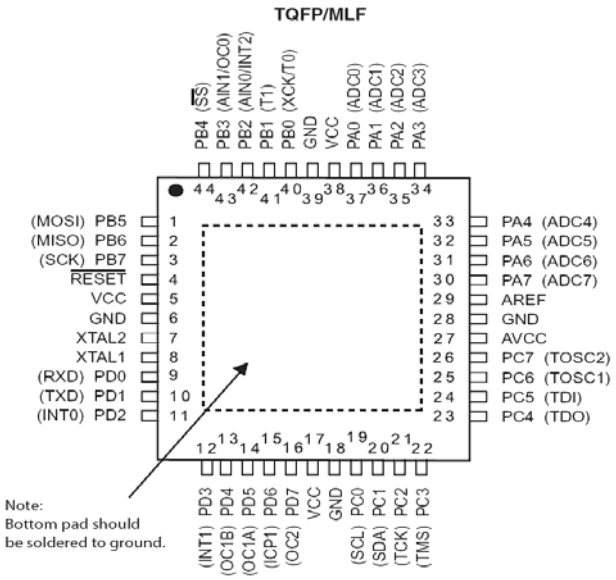


Abbildung 10: PINOut ATmega32 TQFP

Bauteile:

Stückliste: CPU-Unit ATmega32			
Sonstiges		*Halbleiter*	
Q1	Quarz 16,0 MHz	IC1	ATMEL ATmega32 RISC CPU

Tabelle 24: Stückliste CPU-Unit ATmega32

PINOut ATmega128:

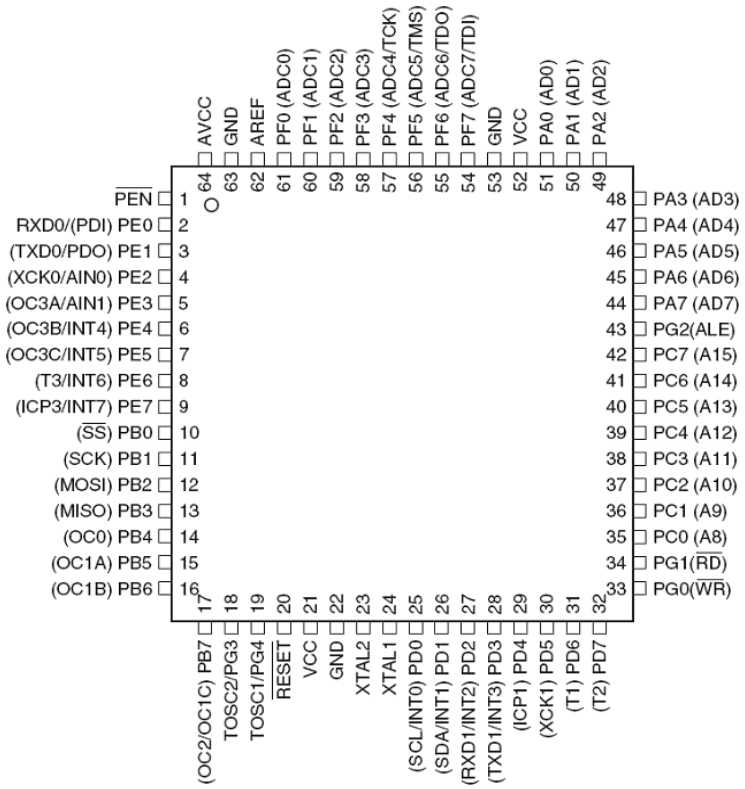


Abbildung 11: PINOut ATmega128 tqfp

Bauteile:

Stückliste: CPU-Unit ATmega128			
Sonstiges		*Halbleiter*	
Q1	Quarz 16,0 MHz	IC1	ATMEL ATmega128 RISC CPU

Tabelle 25: Stückliste CPU-Unit ATmega128

8.1.1 Ressourcenzuordnung ATmega32 zu Projekt ConvCtrl

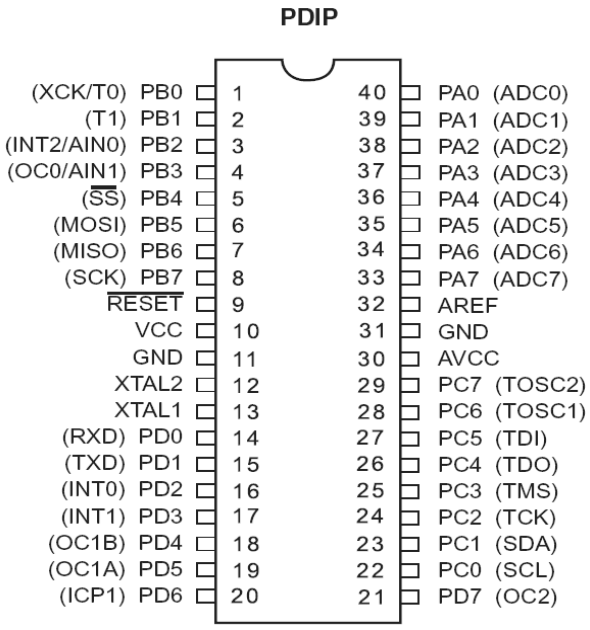


Abbildung 12: PINOut ATmega32 PDIP

PIN	*Port*	*Funktion*	*Used*	*Beschreibung*	*Definition*
1	PB0	XCK/T0	J	Trace ein/aus	TRACE_ON *[TRACE_ON]*
2	PB1	T1	J	Alive LED für ATmega	ALIVE *[ALIVE]*
3	PB2	INT2 AIN0	J	Anschluss DCF77-Modul	INT2 *[DCF77]*
4	PB3	OC0 AIN1	J	Timer-Tick für Sekunden-Impuls	SEC_TCIK *[SEC_TICK]*
5	PB4	SS	J	Schaltleitung Regensensor	RAIN *[RAIN]*
6	PB5	MOSI	J	ISP Programmierinterface	MOSI *[MOSI]*
7	PB6	MISO	J	ISP Programmierinterface	MISO *[MISO]*
8	PB7	SCK	J	ISP Programmierinterface	SCK *[SCK]*
9	RESET	RESET	J	Externer Reset Eingang	RESET *[RESET]*
10	VCC	VCC	J	Spannungsversorgung +5V	VCC *[VCC]*
11	GND	GND	J	Ground GND	GND *[GND]*
12	XTAL2	XTAL2	J	Externer 16 MHz Quarz	XTAL2 *[VCC]*
13	XTAL1	XTAL1	J	Externer 16 MHz Quarz	XTAL1 *[XTAL1]*
14	PD0	RXD	J	RS232 Schmittstelle – Receive Data	RXD *[RxD]*
15	PD1	TXD	J	RS232 Schmittstelle – Transmit Data	TXD *[TxD]*
16	PD2	INT0	J	Nulldurchgangserkennung Dimmung	INT_DIM *[INT5]*
17	PD3	INT1	J	Impulserfassung Windgeber	INT_WIND *[INT6]*
18	PD4	OC1B	J	Temperatursensor TWI Data	TEMP_DATA *[TEMP_DATA]*
19	PD5	OC1A	J	Temperatursensor TWI SCK	TEMP_SCK *[TEMP_SCK]*
20	PD6	ICP1	J	Zündleitung Triac	TRIAC *[TRIAC]*
21	PD7	OC2	J	PWM für Displayhelligkeit	LCD_PWM *[OC3A]*
22	PC0	SCL	J	Anschluss externes EEPROM über TWI / I^2C	SCL *[SCL]*
23	PC1	SDA	J	Anschluss externes EEPROM über TWI / I^2C	SDA *[SDA]*
24	PC2	TCK	J	Gesamtsteuerung Dimmung AN/AUS	DIMMUNG *[DIMMUNG]*
25	PC3	TMS	J	Beschattung AUF	CONF_AUF *[CONV_AUF]*
26	PC4	TDO	J	Beschattung ZU	CONF_ZU *[CONV_ZU]*
27	PC5	TDI	J	Lüftersteuerung	FAN *[FAN]*
28	PC6	TOSC1	J	Steuerung Schwinkreis für Schallgeber	PEEP *[PEEP]*
29	PC7	TOSC2	-	-	--
30	AVCC	AVCC	J	Versorgungsspannung ADC (AVCC)	AVCC *[ADC0]*
31	GND	GND	J	Ground GND für ADC (AGND)	AGND *[ADC0]*
32	AREF	AREF	J	Externe Referenzspannung (AREF)	AREF *[ADC0]*

33	PA7	ADC7	J	LCD ONOFF	LCD_ON	*[LCD_ON]*
34	PA6	ADC6	J	LCD Command/Data	LCD_RS	*[LCD_RS]*
35	PA5	ADC5	J	LCD Enable	LCD_E	*[LCD_E]*
36	PA4	ADC4	J	LCD Datenbit 3	LCD_BIT3	*[LCD_BIT3]*
37	PA3	ADC3	J	LCD Datenbit 2	LCD_BIT2	*[LCD_BIT2]*
38	PA2	ADC2	J	LCD Datenbit 1	LCD_BIT1	*[LCD_BIT1]*
39	PA1	ADC1	J	LCD Datenbit 0	LCD_BIT0	*[LCD_BIT0]*
40	PA0	ADC0	J	Lichtsensor über OpAmp AD623	ADC0	*[ADC0]*

Tabelle 26: Ressourcenzuordnung ATmega32

Zu Timer0 gehörende Pin's
Zu Timer1 gehörende Pin's
Zu Timer2 gehörende Pin's
Zu den Analog-Digital-Wandlern gehörend
JTAG interface
I²C Interface
ISP Programmierinterface
Serielle Schnittstelle RS232
Externe Interrupts und Reset
Spannungsversorgungen
Externe Betaktung
Analog Conmperator

Die unter „Definition" vergebenen Namen definieren die Namensgebung der Signalleitungen im Schaltplan und ggf. in der Software.

8.1.2 Ressourcenzuordnung ATmega128 zu Projekt ConvCtrl

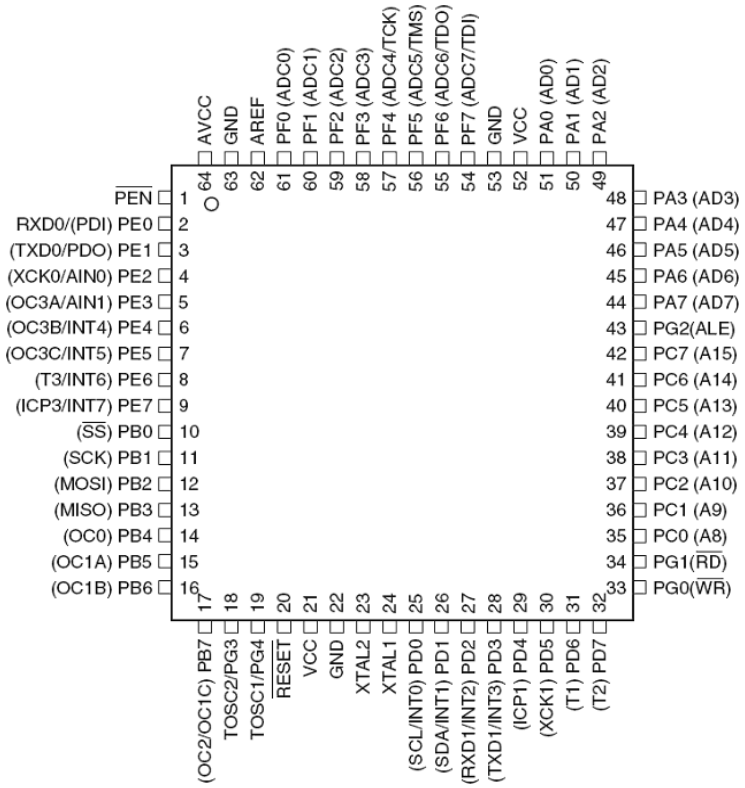


Abbildung 13: PINOut ATmega128 TQFP

PIN	Port	Funktion		Used	Beschreibung	Definition	
1	PEN	PEN		J	Programming Enable Bit für serielle Prog. Über 1k an VCC!	PEN	[PEN]
2	PE0	RXD0	PDI	J	ISP Programmierinterface	MOSI	[MOSI]
3	PE1	TXD0	PDO	J	ISP Programmierinterface	MISO	[MISO]
4	PE2	XCK0	AIN0	J	Licht dauerhaft EIN	CON_LUM	[CON_LUM]
5	PE3	OC3A	AIN1	J	LCD PWM für Displayhelligkeit	LCD_PWM	[OC3A]
6	PE4	OC3B	INT4	J	FAN PWM für Lüftersteuerung	FAN_PWM	[FAN]
7	PE5	OC3C	INT5	J	DIM Nulldurchgangserkennung	INT_DIM	[INT5]
8	PE6	T3	INT6	J	Impulserfassung Windgeber	INT_WIND	[INT6]
9	PE7	ICP3	INT7	J	Interrupt bei Betätigung einer Taste	KEY_INT	[INT7]
10	PB0	SS		J	Alive LED für ATmega	ALIVE	[ALIVE]
11	PB1	SCK		J	ISP Programmierinterface	SCK	[SCK]
12	PB2	MOSI		J	Anschluss DCF77-Modul	DCF77	[DCF77]
13	PB3	MISO		N			
14	PB4	OC0		N			
15	PB5	OC1A		N			
16	PB6	OC1B		J	Timer-Tick für Sekunden-Impuls	SEC_TCIK	[SEC_TICK]
17	PB7	OC2	OC1C	J	Trace ein/aus	TRACE_ON	[TRACE_ON]
18	PG3	TOSC2		J	Schaltleitung Regensensor	RAIN	[RAIN]
19	PG4	TOSC1		J	LED-Array Latch Clear	CLEAR	[CLEAR]
20	RESET	RESET		J	Externer Reset Eingang	RESET	[RESET]
21	VCC	VCC		J	Spannungsversorgung +5V	VCC	[VCC]
22	GND	GND		J	Ground GND	GND	[GND]
23	XTAL2	XTAL2		J	Externer 16 MHz Quarz	XTAL2	[VCC]
24	XTAL1	XTAL1		J	Externer 16 MHz Quarz	XTAL1	[XTAL1]
25	PD0	SCL	ÎNT0	J	Anschluss externes EEPROM über TWI / I^2C	SCL	[SCL]
26	PD1	SDA	INT1	J	Anschluss externes EEPROM über TWI / I^2C	SDA	[SDA]
27	PD2	RXD1	INT2	J	RS232 Schnittstelle – Receive Data	RXD	[RxD]
28	PD3	TXD1	INT3	J	RS232 Schnittstelle – Transmit Data	TXD	[TxD]
29	PD4	ICP1		J	Zündleitung Triac	TRIAC	[TRIAC]
30	PD5	XCK1		J	Signalleitung Piezo-Schallgeber	SOUND	[SOUND]
31	PD6	T1		J	Temperatursensor TWI Data	TEMP_DATA	[TEMP_DATA]
32	PD7	T2		J	Temperatursensor TWI SCK	TEMP_SCK	[TEMP_SCK]
33	PG0	WR		J	Gesamtsteuerung Dimmung AN/AUS	DIMMUNG	[DIMMUNG]
34	PG1	RD		J	LCD ONOFF	LCD_ON	[LCD_ON]
35	PC0	A8		J	LCD Datenbit 0	LCD_ BIT0	[LCD_BIT0]
36	PC1	A9		J	LCD Datenbit 1	LCD_BIT1	[LCD_BIT1]
37	PC2	A10		J	LCD Datenbit 2	LCD_BIT2	[LCD_BIT2]
38	PC3	A11		J	LCD Datenbit 3	LCD_BIT3	[LCD_BIT3]
39	PC4	A12		J	LCD Enable	LCD_E	[LCD_E]
40	PC5	A13		J	LCD Command/Data	LCD_RS	[LCD_RS]
41	PC6	A14		J	Beschattung AUF	CONV_AUF	[CONV_AUF]
42	PC7	A15		J	Beschattung ZU	CONV_ZU	[CONV_ZU]
43	PG2	ALE		J	LED-Array Adresse (Bit 1)	ADR_1	[ADR_1]
44	PA7	AD7		J	LED-Array Adresse (Bit 0)	ADR_0	[ADR_0]
45	PA6	AD6		J	LED-Array Daten (Bit 3)	DATA_3	[DATA_3]
46	PA5	AD5		J	LED-Array Daten /Bit 2)	DATA_2	[DATA_2]
47	PA4	AD4		J	LED-Array Daten (Bit 1)	DATA_1	[DATA_1]
48	PA3	AD3		J	LED-Array Daten (Bit 0)	DATA_0	[DATA_0]
49	PA2	AD2		J	Tastaturmatrix (Zeilen)	SW_IN_3	[SW_IN_3]
50	PA1	AD1		J	Tastaturmatrix (Zeilen)	SW_IN_2	[SW_IN_2]
51	PA0	AD0		J	Tastaturmatrix (Zeilen)	SW_IN_1	[SW_IN_1]
52	VCC	VCC		J	Spannungsversorgung +5V	VCC	[VCC]

53	GND	GND		J	Ground GND	GND	*[GND]*
54	PF7	ADC7	TDI	J	JTAG Interface	TDI	*[TDI]*
55	PF6	ADC6	TDO	J	JTAG Interface	TDO	*[TDO]*
56	PF5	ADC5	TMS	J	JTAG Interface	TMS	*[TMS]*
57	PF4	ADC4	TCK	J	JTAG Interface	TCK	*[TCK]*
58	PF3	ADC3		J	Tastaturmatrix (Spalten)	SW_OUT_3	*[SW_OUT_3]*
59	PF2	ADC2		J	Tastaturmatrix (Spalten)	SW_OUT_2	*[SW_OUT_2]*
60	PF1	ADC1		J	Tastaturmatrix (Spalten)	SW_OUT_1	*[SW_OUT_1]*
61	PF0	ADC0		J	Lichtsensor über OpAmp AD623	ADC0	*[ADC0]*
62	AREF	AREF		J	Externe Referenzspannung (AREF)	AREF	*[ADC0]*
63	GND	GND		J	Ground GND für ADC (AGND)	AGND	*[ADC0]*
64	AVCC	AVCC		J	Versorgungsspannung ADC (AVCC)	AVCC	*[ADC0]*

Tabelle 27: Ressourcenzuordnung ATmega128

Farbliche zuordnungen „Funktion“:

- Zu Timer0 gehörende Pin’s
- Zu Timer1 gehörende Pin’s
- Zu Timer2 gehörende Pin’s
- Zu Timer3 gehörende Pin’s
- Zu den Analog-Digital-Wandlern gehörend
- JTAG interface
- I^2C Interface
- ISP Programmierinterface
- Serielle Schnittstelle RS232
- Externe Interrupts und Reset
- Spannungsversorgungen
- Externe Betaktung
- Adressleitungen für externes Speicherinterface
- Seriell Programming Interface für
- Analog Conmperator

Farbliche Zuordnung „PIN“:

Mit Grau gekennzeichnete Pin-Felder werden über Verbindungsstecker auf PCB2 (Konsolen-Board) geführt weil sie dort in direktem Zusammen mit Eingabe, Ausgabe und Statusmeldung stehen.

8.1.3 Interrupt-Vektor-Tabelle ATmega128

Vector No.	Program Address(2)	Source	Interrupt Definition
1	$0000(1)	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	$0002	INT0	External Interrupt Request 0
3	$0004	INT1	External Interrupt Request 1
4	$0006	INT2	External Interrupt Request 2
5	$0008	INT3	External Interrupt Request 3
6	$000A	INT4	External Interrupt Request 4
7	$000C	INT5	External Interrupt Request 5
8	$000E	INT6	External Interrupt Request 6
9	$0010	INT7	External Interrupt Request 7
10	$0012	TIMER2 COMP	Timer/Counter2 Compare Match
11	$0014	TIMER2 OVF	Timer/Counter2 Overflow
12	$0016	TIMER1 CAPT	Timer/Counter1 Capture Event
13	$0018	TIMER1 COMPA	Timer/Counter1 Compare Match A
14	$001A	TIMER1 COMPB	Timer/Counter1 Compare Match B
15	$001C	TIMER1 OVF	Timer/Counter1 Overflow
16	$001E	TIMER0 COMP	Timer/Counter0 Compare Match
17	$0020	TIMER0 OVF	Timer/Counter0 Overflow
18	$0022	SPI, STC	SPI Serial Transfer Complete
19	$0024	USART0, RX	USART0, Rx Complete
20	$0026	USART0, UDRE	USART0 Data Register Empty
21	$0028	USART0, TX	USART0, Tx Complete
22	$002A	ADC	ADC Conversion Complete
23	$002C	EE READY	EEPROM Ready
24	$002E	ANALOG COMP	Analog Comparator
25	$0030(3)	TIMER1 COMPC	Timer/Countre1 Compare Match C
26	$0032(3)	TIMER3 CAPT	Timer/Counter3 Capture Event
27	$0034(3)	TIMER3 COMPA	Timer/Counter3 Compare Match A
28	$0036(3)	TIMER3 COMPB	Timer/Counter3 Compare Match B
29	$0038(3)	TIMER3 COMPC	Timer/Counter3 Compare Match C
30	$003A(3)	TIMER3 OVF	Timer/Counter3 Overflow

Vector No.	Program Address(2)	Source	Interrupt Definition
31	$003C(3)	USART1, RX	USART1, Rx Complete
32	$003E(3)	USART1, UDRE	USART1 Data Register Empty
33	$0040(3)	USART1, TX	USART1, Tx Complete
34	$0042(3)	TWI	Two-wire Serial Interface
35	$0044(3)	SPM READY	Store Program Memory Ready

Notes: 1. When the BOOTRST fuse is programmed, the device will jump to the Boot Loader address at reset, see "Boot Loader Support – Read-While-Write Self-Programming" on page 276.
2. When the IVSEL bit in MCUCR is set, interrupt vectors will be moved to the start of the Boot Flash section. The address of each interrupt vector will then be address in this table added to the start address of the boot Flash section.
3. The Interrupts on address $0030 - $0044 do not exist in ATmega103 compatibility mode.

Table 24 shows Reset and interrupt vectors placement for the various combinations of BOOTRST and IVSEL settings. If the program never enables an interrupt source, the interrupt vectors are not used, and regular program code can be placed at these locations. This is also the case if the Reset Vector is in the Application section while the interrupt vectors are in the Boot section or vice versa.

Table 24. Reset and Interrupt Vectors Placement

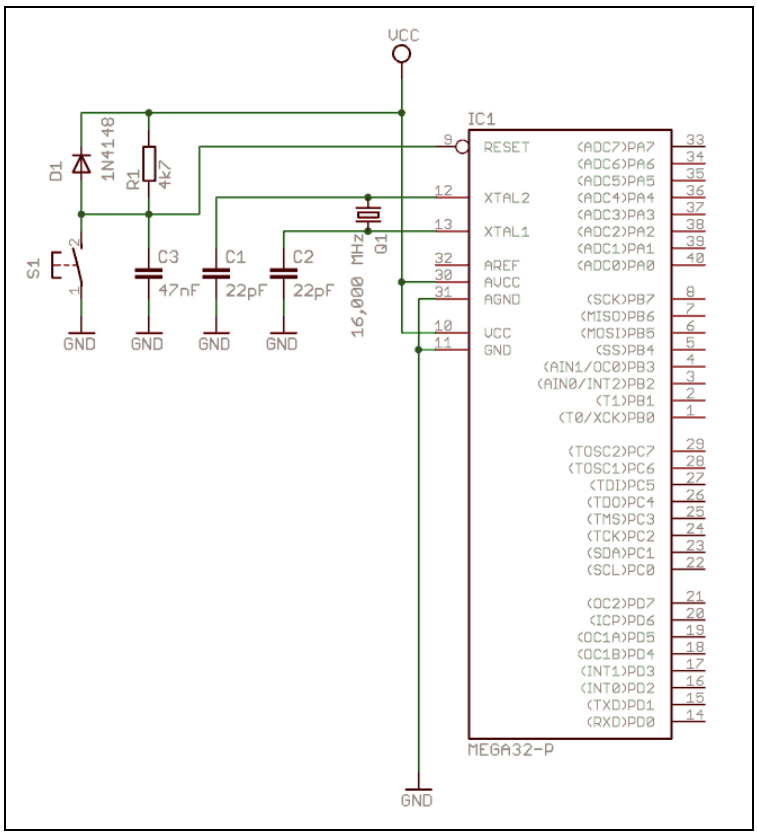
BOOTRST	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0	$0000	$0002
1	1	$0000	Boot Reset Address + $0002
0	0	Boot Reset Address	$0002
0	1	Boot Reset Address	Boot Reset Address + $0002

Note: The Boot Reset Address is shown in Table 112 on page 287. For the BOOTRST fuse "1" means unprogrammed while "0" means programmed.

Tabelle 28: Interrupt-Vektor-Tabelle TAmega128

8.1.4 Basisbeschaltung des ATmega32

Die im Folgenden dargestellte Schaltung bildet die Basisbeschaltung eines AVR ATmega. Zum Einsatz kommt ein externer Quarz mit 16.000.000 Hz (16 MHz). Weiter ist im Schaltplan der Anschluss eines externen RESET-Tasters vorgesehen.



Schaltbild 1: Basisbeschaltung ATmega

Bauteile:

Stückliste: Basisbeschaltung ATmega32			
Widerstände		*Halbleiter*	
R1	4k7 Ω	IC1	ATMEL AVR ATmega32-P
		D1	Diode 1N4148
Kondensatoren		*Sonstiges*	
C1, C2	Keramikkondensator 22 pF	S1	Kurzhubtaster
C3	Keramikkondensator 47 nF	Q1	Quarz 16.000.000 Hz

Tabelle 29: Stückliste Basisbeschaltung ATmega

Reset-Taster:

Als Reset-Taster wir ein Print-Kurzhubtaster verwendet welcher auf der Display- und Tastatur-PCB für die Frontblende untergebracht ist. Der Reset-Taster ist über eine Bohrung im Frontdeckel mit einem dünnen Gegenstand (z.B. Büroklammer) erreichbar und zu betätigen.



Kurzhubtaster 6x6mm, Höhe: 4,3mm, 12V, vertikal

Bestellnummer Reichelt elektronik: *TASTER 3301*

Tabelle 30: Kurzhubtaster für RESET

8.1.5 ISP-Schnittstelle / ISP-Programmierung

Zur In-System-Programmierung wird die ATMEL ISP-Schnittstelle umgesetzt. Die Programmierung im Projekt erfolgt direkt über BASCOM mit Hilfe von PonyProg oder STK500 von ATMEL.

Zum Ändern der Firmware wird „Pony Prog“ empfohlen. Diese Software und die Programmierer-Beschreibung kann als Freeware von

http://www.lancos.com

bezogen werden. Grundsätzlich ist der Mikrocontroller mit jedem Atmel - ISP - Kompatiblen Programmiergerät programmierbar.

ISP Programmierinterface von ATMEL:

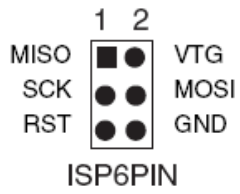
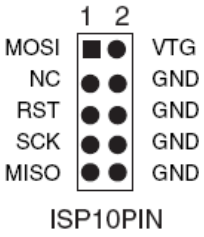
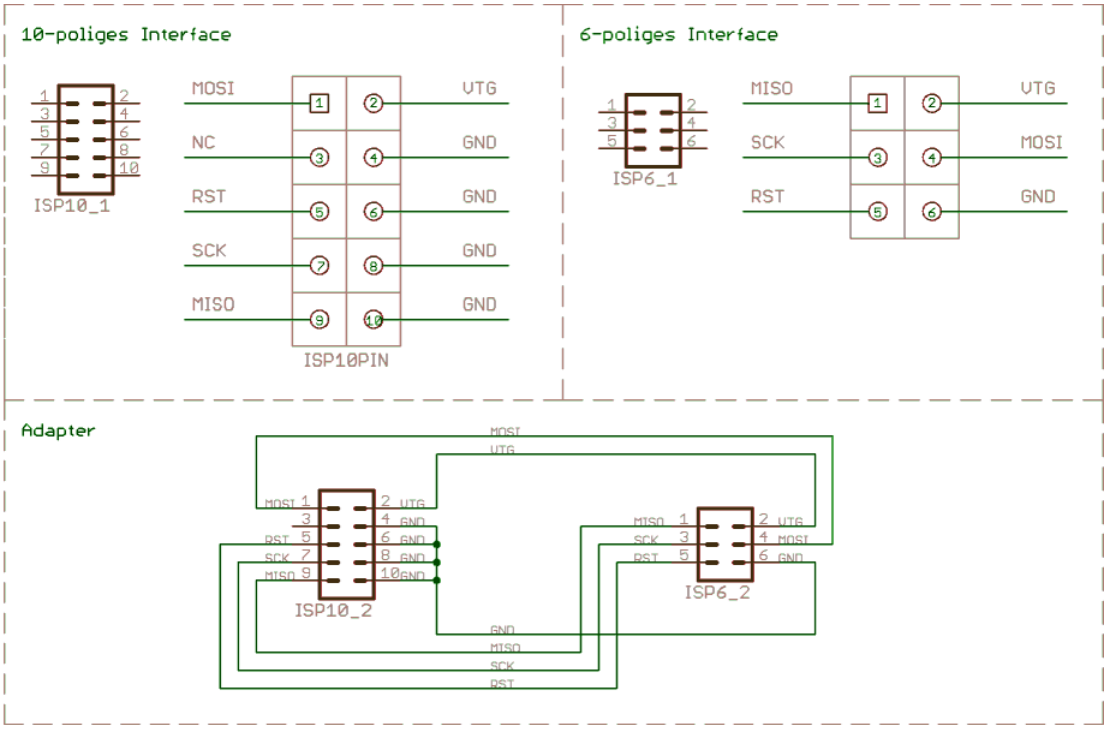
	
6-pin ISP connection Pinout	10-pin ISP connection Pinout

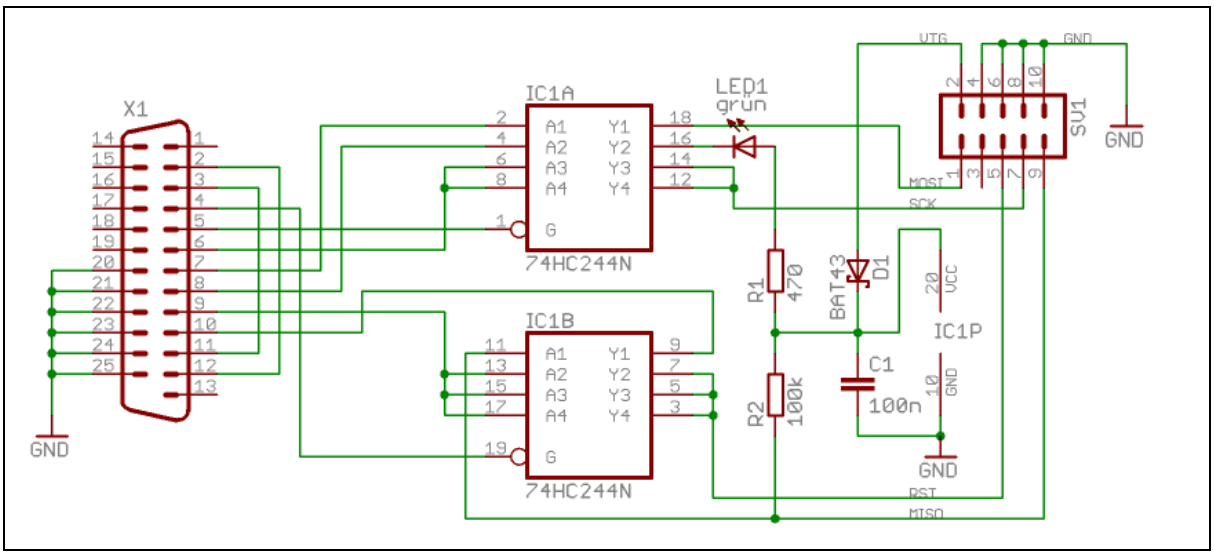
Tabelle 31: ISP connection Pinout

Für das Projekt ConvCtrl wir das 10-polige Interface in der folgenden Form umgesetzt:



Schaltbild 2: 10-Pin ISP connection Pinout

Die nachfolgende Schaltung zeigt einen Programmieradapter wie er für das Projekt umgesetzt wurde. Zur endgültigen Programmierung des Boards wurde PonyProg mit den dafür vorgesehenen Programmieradaptern verwendet. Eine Beschreibung der Programmieradapter befindet sich i Anschluss an dieses Kapitel.



Schaltbild 3: ISP Programmierinterface

Und hier das fertig geroutete Board in Version 1.0:

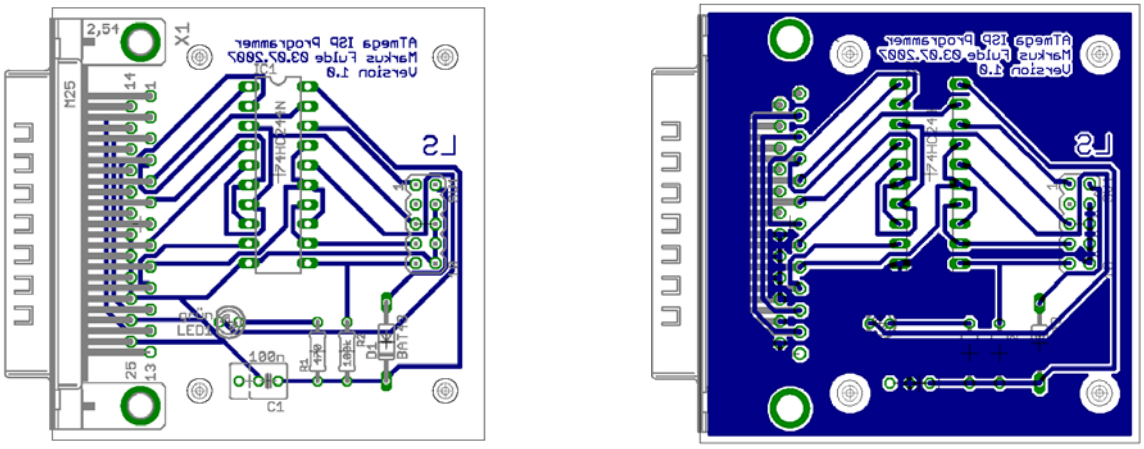


Abbildung 14: PCB des ISP Programmierinterface

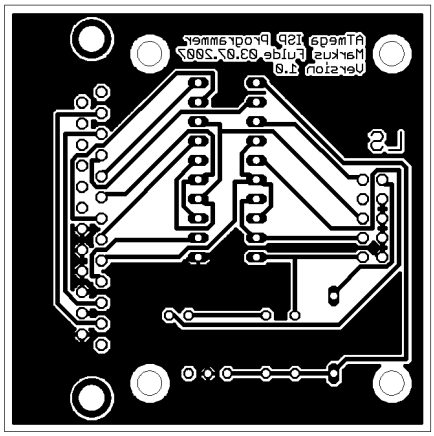


Abbildung 15: Layer Bottom ISP Programmierinterface

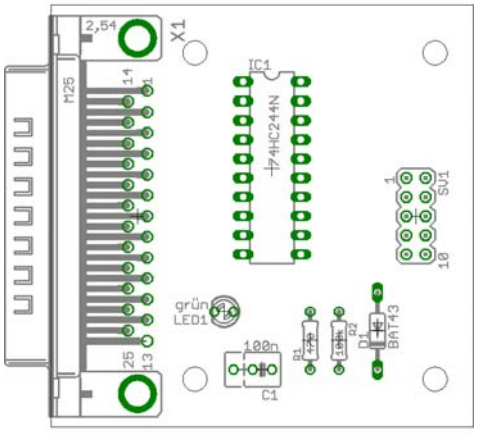


Abbildung 16: Bestückung ISP Programmierinterface

<u>Bauteile:</u>

Stückliste: ISP Programmierinterface			
Widerstände		*Halbleiter*	
R1	470 Ω	IC1	74HC244N
R2	100k Ω	D1	Diode BAT43
Kondensatoren		LED1	Leuchtdiode GRÜN
C1	100 nF	*Sonstiges*	
		X1	D-SUB-Stecker, 25-polig, gewinkelt, RM 9,4
		SV1	Pin Header MA05-2

Tabelle 32: Stückliste Basisbeschaltung ATmega

Der im Projekt verwendete D-SUB-Stecker hat bei Reichelt Elektronik die Bestellnummer: *D-SUB ST 25EU*

Programmiereinstellungen für Ponyprog:

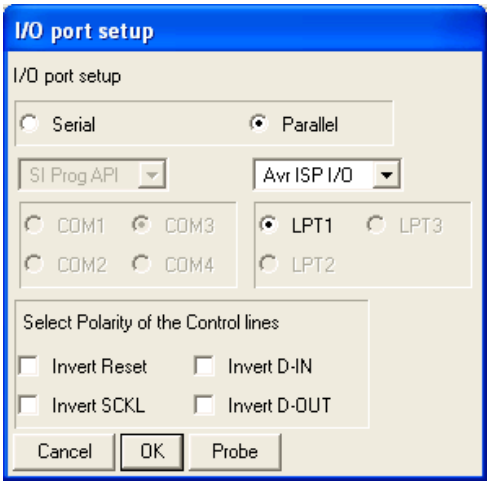
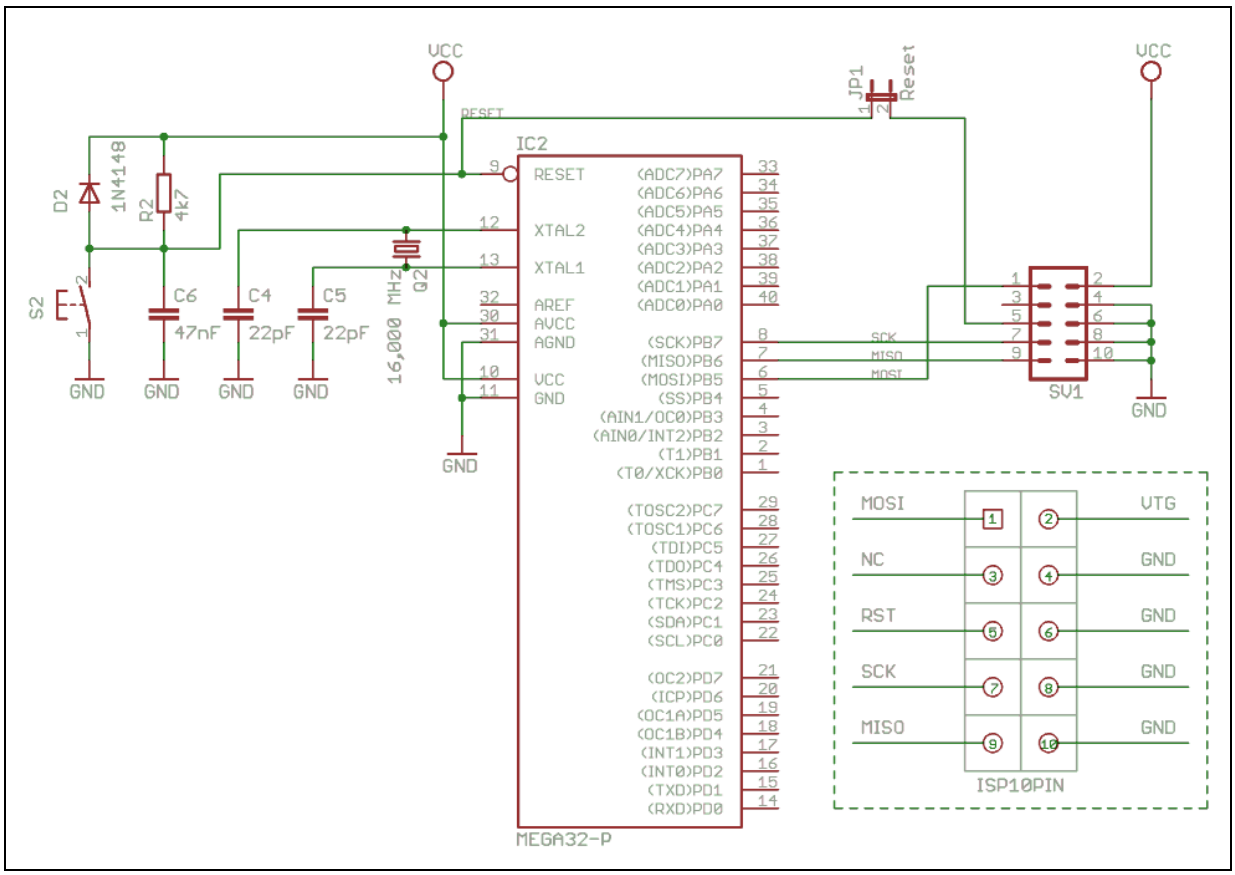


Abbildung 17: Einstellung Ponyprog für ISP-Programmierinterface

Ergänzung der ATmega Basisbeschaltung um ISP-Programmierinterface:



Schaltbild 4: ATmega Basisbeschaltung mit ISP-Programmierinterface

Bauteile:

Stückliste: Basisbeschaltung ATmega mit ISP-Programmierinterface			
Widerstände		*Halbleiter*	
R1	4k7 Ω	IC1	ATMEL AVR ATmega32
		D1	Diode 1N4148
Kondensatoren		*Sonstiges*	
C1, C2	Keramikkondensator 22 pF	S1	Taster
C3	Keramikkondensator 47 nF	Q1	Quarz 16.000.000 Hz
		SV1	Pin Header MA05-2
		JP1	Jumper 2-polig, 2,54mm Raster

Tabelle 33: Stückliste Basisbeschaltung ATmega mit ISP-Programmierinterface

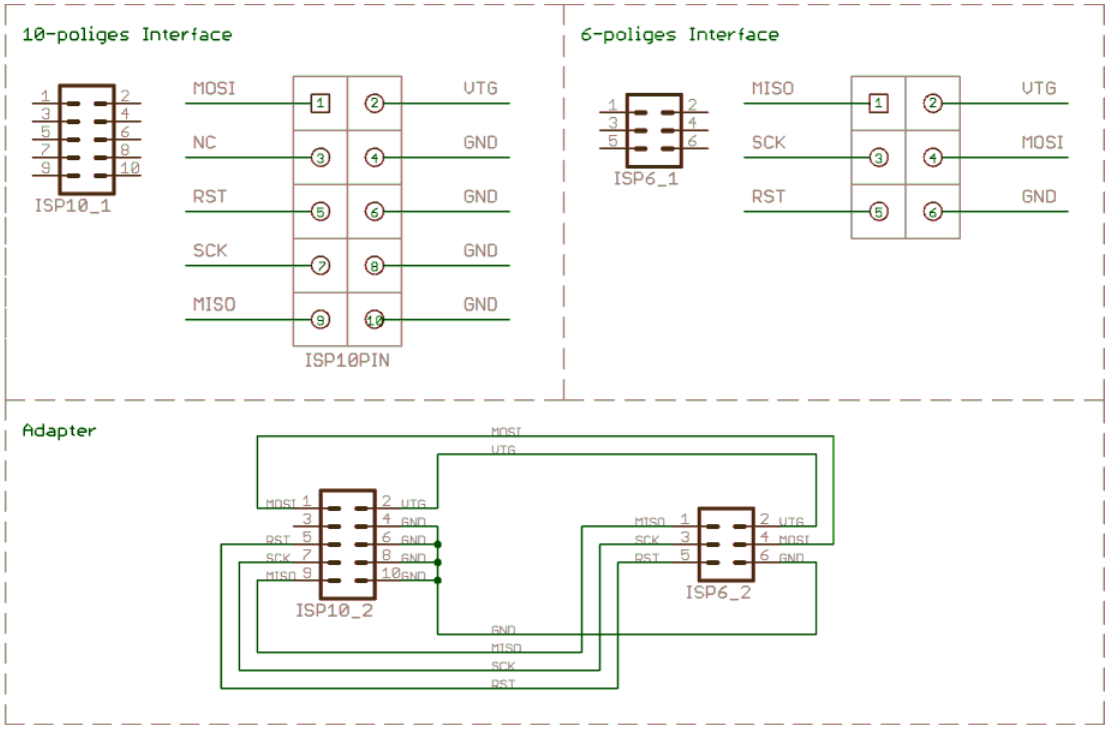
Als Kabelverbindung zwischen Target und ISP-Programmieradapter wird ein Pfostenverbinder in Schneidklemmtechnik für Flachbandkabel mit Zugentlastung eingesetzt.



Beschreibung:	PFL 10 Pfostenv. 10pol + Zugentl.
Rastermaß:	2,54 x 2,54 mm
Bestellnummer Reichelt:	*PFL 10*

Abbildung 18: PFL10 Pfostenverbinder 10 polig

Adapter 6-polig <-> 10-polig ISP



Schaltbild 5: ISP Adapter 6-Pin zu 10-Pin.

Bauteile:

Stückliste: ISP Adapter 6-Pin auf 10-Pin		
Sonstiges		
ISP10_2	Pin Header MA05-2	
ISP6_2	Pin Header MA03-2	

Tabelle 34: Stückliste Basisbeschaltung ATmega mit ISP-Programmierinterface

8.1.6 Ponyprog Programmieradapter

PonyProg unterstützt verschiedene Hardware Interfaces. Es ist zu beachten, das man nur mit dem SI-Prog Interface in der Lage ist, alle unterstützten Deives zu programmieren. Welche Desvies dies genau sind, ist der PonyProg Homepage

http://www.lancos.com/prog.html

oder der PonyProg Anleitung zu entnehmen.

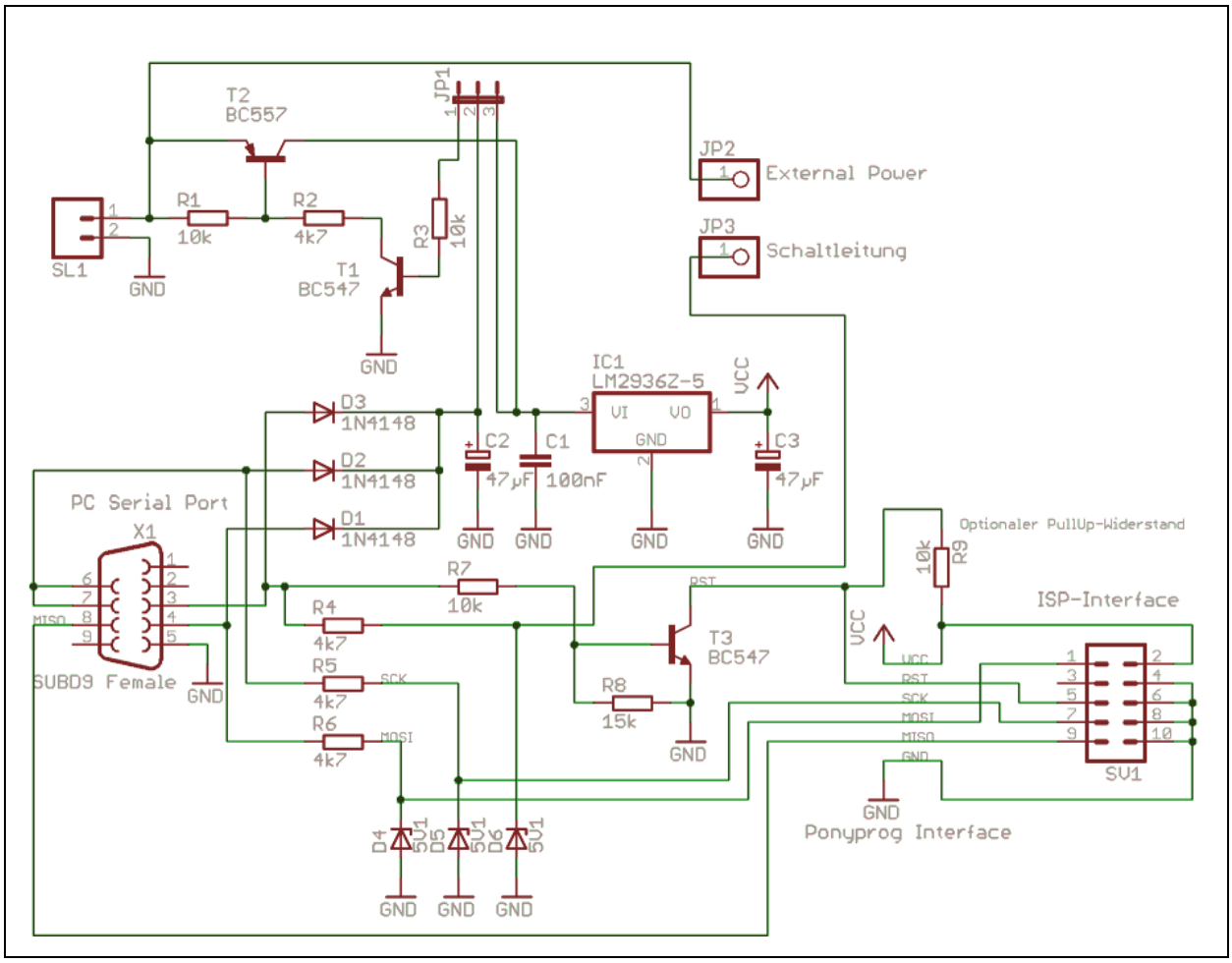
http://www.lancos.com/e2p/ponyprog2000.html

Die im folgenden abgebildeten Schaltbilder basieren auf den original PonyProg schematics welche dem Copyright © von Claudio Lanconelli unterliegen.

Für das Projekt ConvCtrl wurden die Schaltbilder auf das ATMEL ISP Interface (6-polig / 10-polig) angepasst. Weiter wurden Spannungsversorgungen und Teilmodule vereinheitlicht, entsprechend der Umsetzung für ConCtrl.

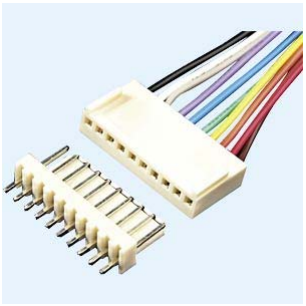
Das geistige Gut des urhebers bleibt dabei unangetastet.

[1] SI-Prog:



Schaltbild 6: PonyProg SI-Prog Interface

Für den externen Spannungsversorgungsanschluss wird ein Platinensteckverbinder eingesetzt:



Platinensteckverbinder gerade, weiß, 2-polig
2-poliger, gerader Platinen-Steckverbinder, bestehend aus Stift-und Buchsenleiste, fest verdrahtet zum Aufbau flexibler Platinenverbindungen.

Kabeltyp: AWG 24 (0,22mm²) für Platinenstärke bis 2,6mm

Rastermaß	2,54 mm
Kabellänge	0,25 m
Höhe	10,0 mm

Bestellnummer Reichelt: *PS 25/2G WS*

Abbildung 19: 2-poliger Platinensteckverbinder

Und hier das fertig geroutete Board in Version 1.2:

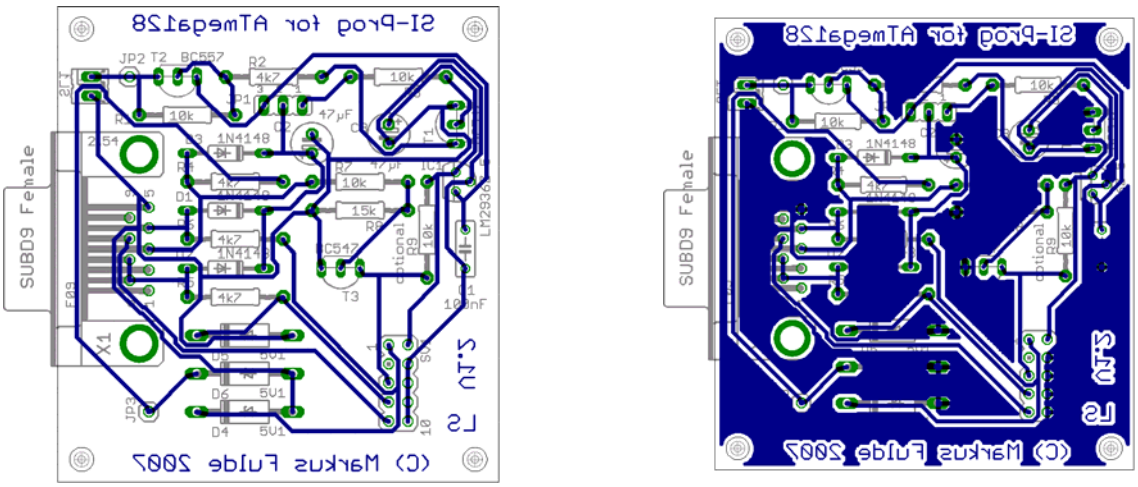


Abbildung 20: PCB des SI-Prog Interface

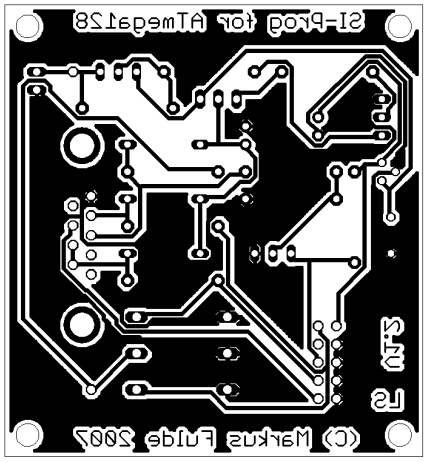


Abbildung 21: Layer Bottom SI-Prog Interface

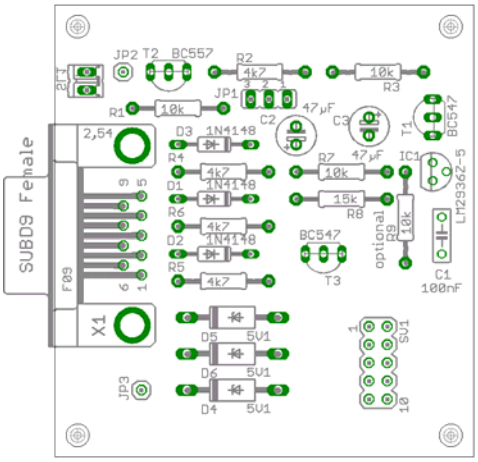


Abbildung 22: Bestückung SI-Prog Interface

Bauteile:

Stückliste: SI-Prog Interface			
Sonstiges		***Halbleiter***	
JP2, JP3	Pin Header 1X1 1-polig	D1, D2, D3	Diode 1N4148
JP1	Jumper JP2E 3-polig	T1, T3	Transistor BC547A
X1	SUB D9 Female	T2	Transistor BC557
SV1	Pin Header MA05-2 5x2-polig	IC1	Festspannungsregler LM2936Z
SL1	Platinensteckverbinder 2-polig	D4, D5, D6	Zener-Diode 5V1
Widerstände		***Kondensatoren***	
R1, R3, R7, R9	Metallschichtwiderstand 10k Ω	C1	Folien-Kondensator 100 nF
R2, R4, R5, R6	Metallschichtwiderstand 4k7 Ω	C2, C3	Elko 47µF / 35V
R8	Metallschichtwiderstand 15k Ω		

Tabelle 35: Stückliste SI-Prog Interface

Der in diesem Projekt verwendete D-SUB-Stecker hat bei Reichelt Elektronik die Bestellnummer: *D-SUB ST 09EU*

Umschaltung zwischen COM powered und external powered VDD mit Jumper JP1:

- Stellung 1-2: External powered
- Stellung 2-3: COM powered

Über SL1 kann eine externe Versorgungsspannung von 9V angelegt werden, falls die Spannung während der Programmierung unter 5B sinken sollte.

Die kann 3 Ursachen haben:

1. Der COM-Port liefert zu wenig Strom
2. Das Device verbraucht zu viel Strom
3. Der LM2936Z-5 wird durch einen anderen Spannungsreglertyp ausgetauscht (z.B. 78L05)

Über die 0-polige SUB-D-Buchse X1 wird das SI-Prog-Interface mit dem PC COM-Port verbunden. Hierzu wird ein normales 1:1 Standardkabel verwendet (kein Nullmodemkabel).

Achtung: Beim ISP-Interface SV1 ist auf die richtige Polartität zu achten (siehe hierzu Pin 1 und Pin 10). Eine falsche Polarität kann zur Zerstörung des Target-Device führen.

[1.1] Serielle Kabel:

Um PonyProg zu verwenden wird ein serielle Kabel DB9 Female zu DB9 Male mit einer 1:1 Verkabelung ALLER Pins inder folgenden Weise benötigt:

DB9 Female	*DB9 Male*
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9

Tabelle 36: Serielle Kabel SI-Prog-Interface

Es wird auch als Modem-Kabel bezeichnet. Wie auch immer – überprüfe dass alle Pins verbunden sind. Vermeide die Verwendung eines Null-Modem-Kabels!

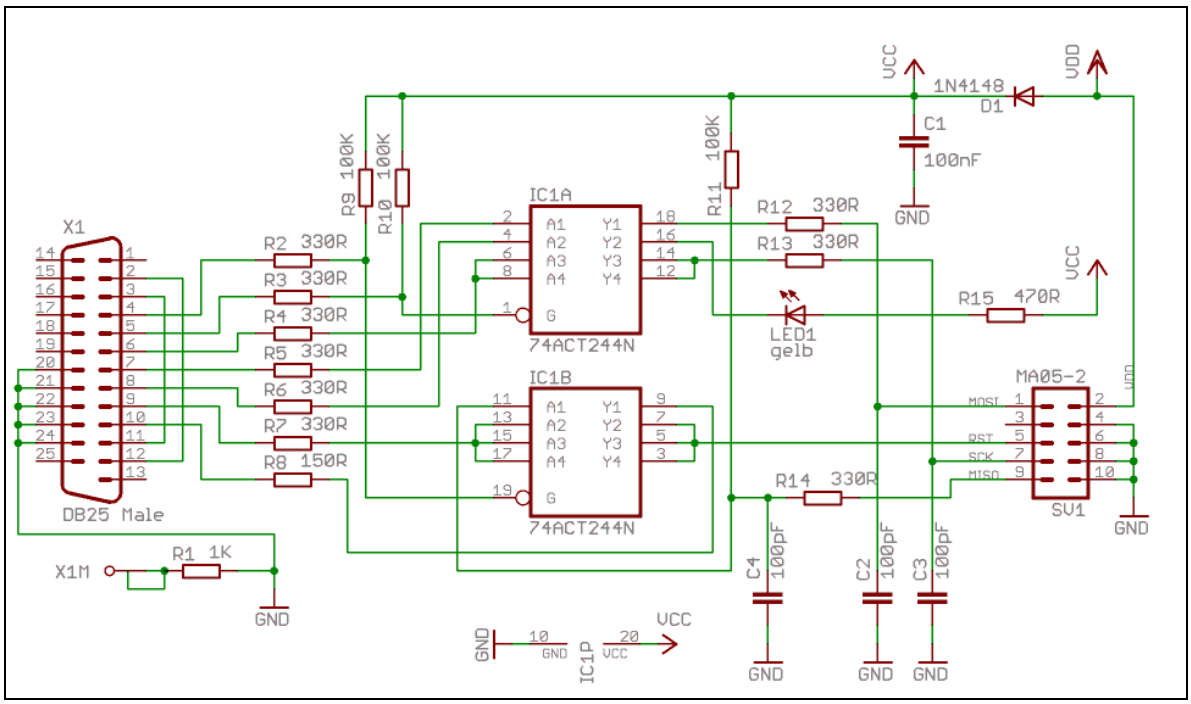
[2] PonyProg AVR ISP (STK200/300) Parallel-Port-Interface:

Dies ist der beste Weg, AVR ISP Programmierung durchzuführen. Das Interface arbeitet aich in Low Volt Systemen (3V).

Zur Auswahl dient in PonyProg das Options – Setup Menü „AVR ISP I/O) oder „AVR ISP API" in Verbindung mit der Parallel-Interface Checkbox.

Das Interface kann direkt auf dem Target-System (ISP) über den 10-poligen ISP-Anschluss angeschlossen werden.

AVR ISP mit ATMEL STK 500 ISP-.PinOut

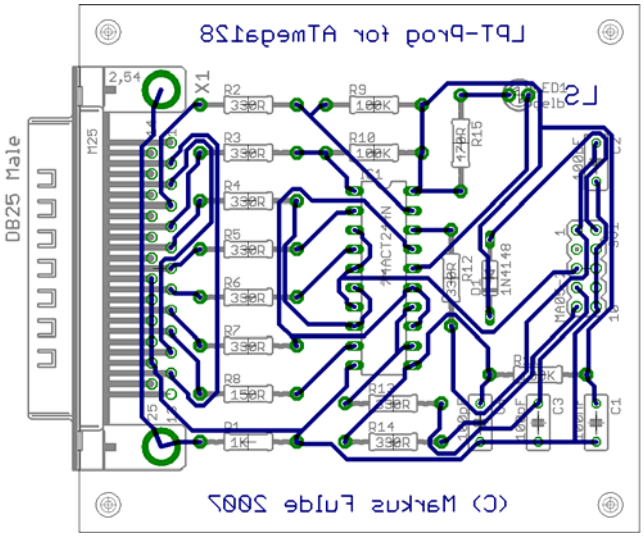


Schaltbild 7: PonyProg ISP-Parallel-Interface

Hinweise:

- Der Widerstand R1 hilft Fehlfunktionen oder Schäden durch elektrostatische Aufladungen zu verhindern.
- R2-R7 sind optional. R8 wird zum Schutz des Parallelport bzw. des Dongles benötigt.
- Die R9-C3, R10-C2 und R11-C4 Filter verhindern Masseschleifen und Resonanzen auf allen drei Datenleitungen.

Und hier das fertig geroutete Board in Version 1.0:



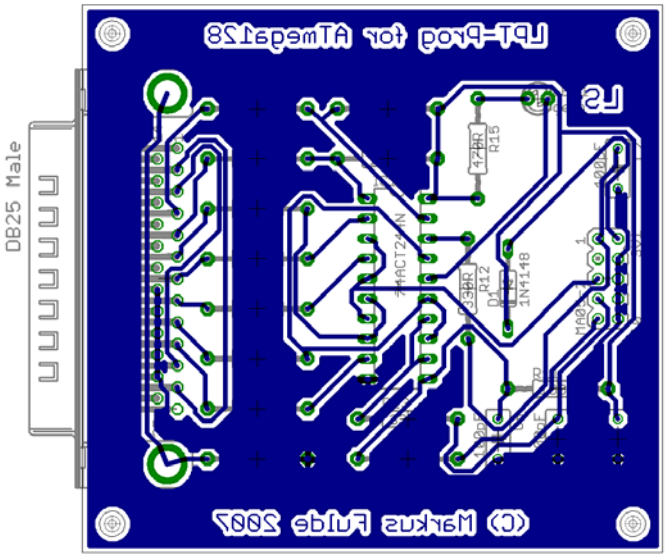


Abbildung 23: PCB des PonyProg ISP-Parallel-Interface

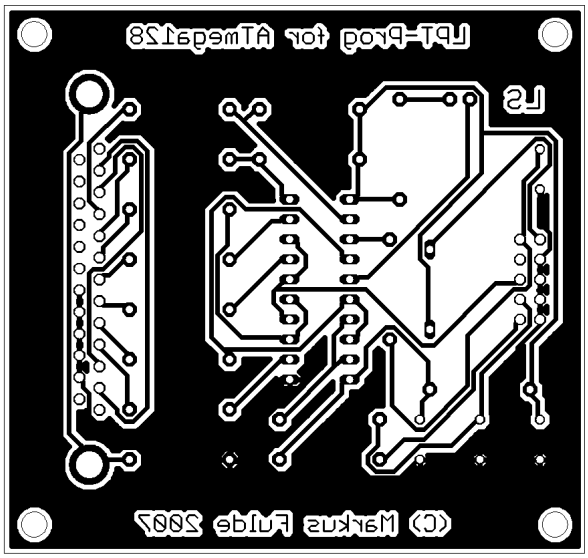


Abbildung 24: Layer Bottom PonyProg ISP-Parallel-Interface

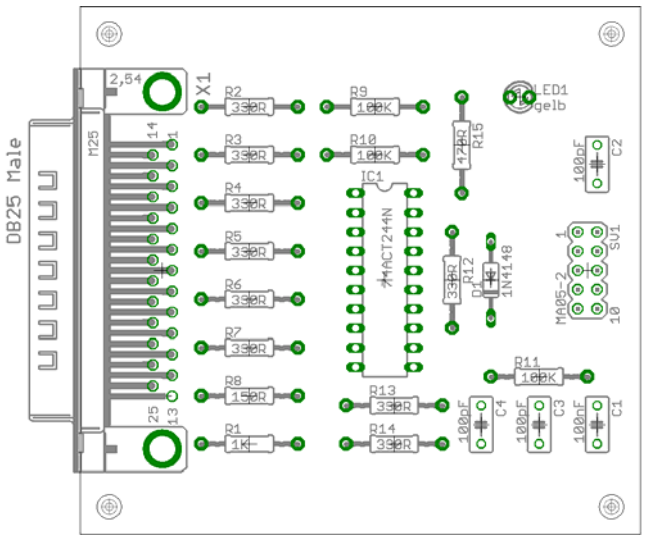


Abbildung 25: Bestückung PonyProg ISP-Parallel-Interface

Bauteile:

Stückliste: PonyProg ISP-Parallel-Interface			
Sonstiges		***Halbleiter***	
X1	25-poli SUBD Male gewinkelt	LED1	LED gelb, 3mm Standard
SV1	Postenstecker MA05-2, 5x2-polig	IC1	Treiber-Buffer SN74HCT244N
		D1	Diode 1N4148
Widerstände		***Kondensatoren***	
R1	Metallschichtwiderstand 1k Ω	C1	Folienkondensator 100nF
R2, R3, R4, R5, R6, R7,R12, R13, R14	Metallschichtwiderstand 330 Ω	C2, C3, C4	Keramtikkondesnator 100pF
R8	Metallschichtwiderstand 150 Ω		
R9, R10, R11	Metallschichtwiderstand 100k Ω		
R15	Metallschichtwiderstand 470 Ω		

Tabelle 37: Stückliste PonyProg ISP-Parallel-Interface

[3] PonyProg Interface Setup:

Öffnen der Dialog-Box "Interface Setup" wo man den Interface-Typ und die Port-Nummer einstellen kann, über die die Harware angeschlossen ist.

Hier können verschiedene Hardware-Interfaces ausgewählt werden. Es ist zu beachten, dass nur das SI Prog Interface in seiner ursprüglichen Form von PonyProg alle Interfaces unterstützt. Die hier abgebildeten Schaltungen in diesem Kapitel sind auf das AMTEL AVR ISP-Interface abgeändert!

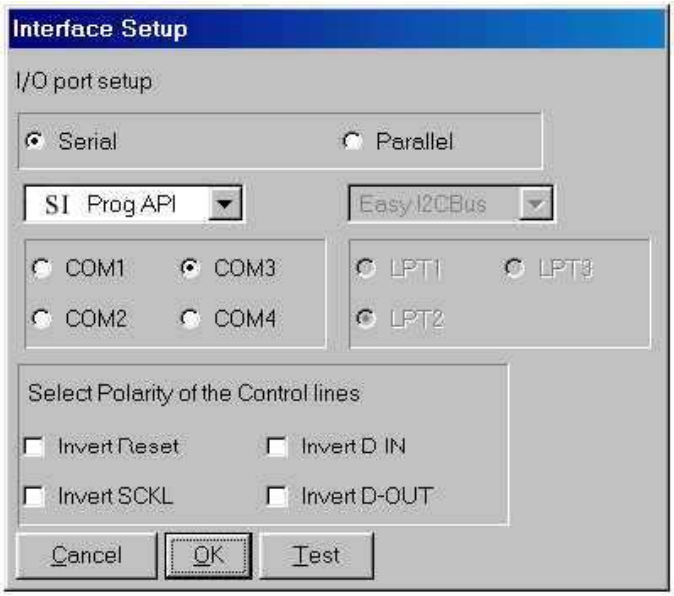


Abbildung 26: PonyProg Interface Setup

Hinweise (O-Ton aus der PonyProg Anleitung):

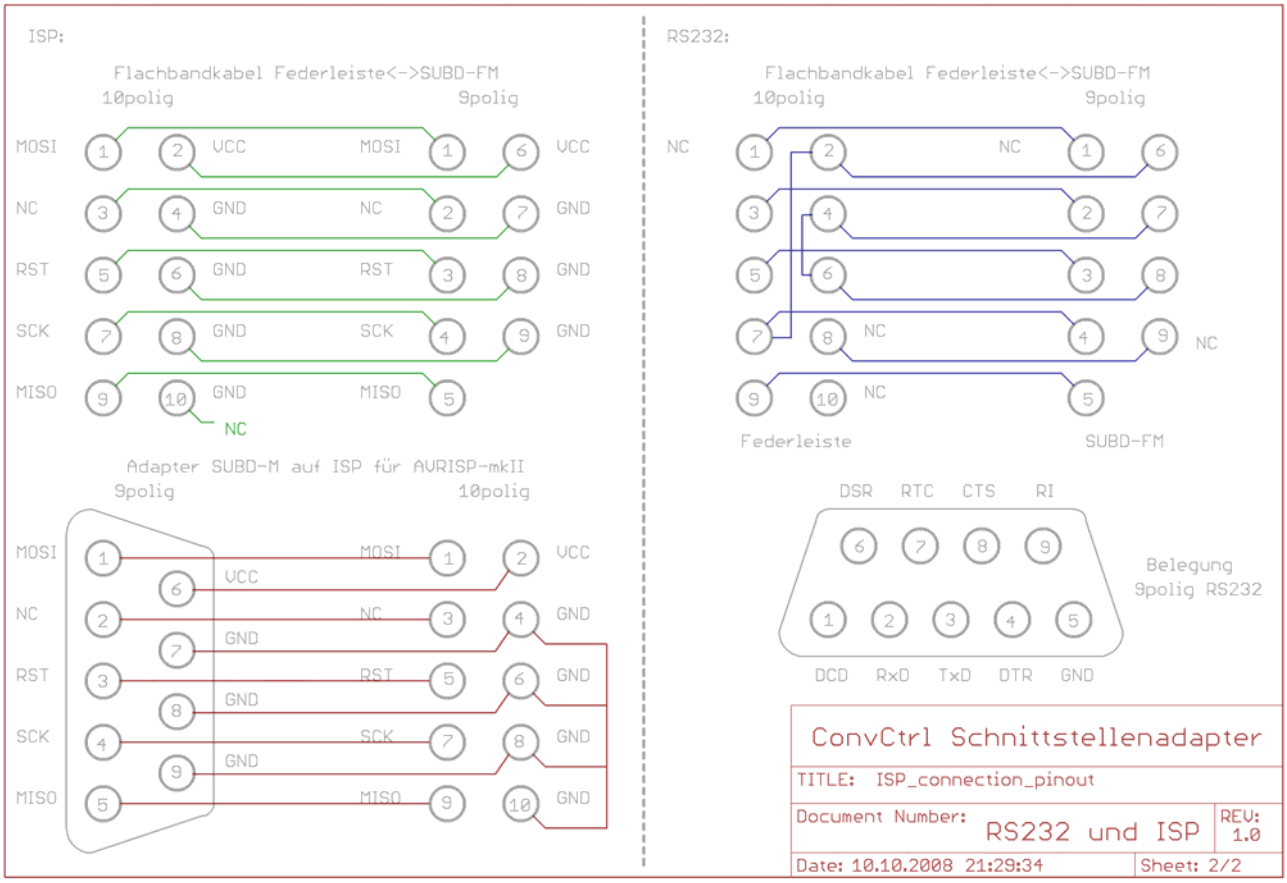
- SI-Prog interface select "serial" check-box and choose "SI-Prog API" or "SI-Prog I/O" in the combo-box. Windows 95/98/ME and NT/2000 work either with "PonyProg API" and "PonyProg I/O", but the latter is faster. To use the "SI-Prog I/O" with Linux you need to run PonyProg as root. Then select the COM port you want to use, if some COM ports are disabled it means that they are used by other programs (usually by the mouse driver or modem), or not installed.
- AVR ISP Parallel port interface select "parallel" check-box and choose "Avr ISP API" or "Avr ISP I/O" in the combo-box. WindowsNT/2000 work only with "Avr ISP I/O". To use the "Avr ISP I/O" with Linux you need to run PonyProg as root. Then select the LPT port you want to use. Note that if you want to program the AT89Sxx device you need to select the "Invert Reset" check-box. To use this interface with Linux you need a 2.4.x kernel and parport, parport_pc and ppdev kernel modules. You can use this interface to read/write the AVR micros and SPI eeproms.
- Ludipipo/JDM interface select "JDM API" or "JDM I/O". Then select the COM port you use. All the considerations for "SI-Prog" interface above are valid for JDM too. You can use this interface to read/write the PIC16x84 and 24Cxx eeproms. If you want to use JDM interface to program 24Cxx devices you have to connect pin 7 of the 24Cxx to GND (schematics are wrong).
- Easy I²CBus interface select the "parallel" check-box, then select the LPT port you want to use. All the considerations for "Avr ISP" abore are valid for "EasyI2CBus" too.
- DT-006 interface This are the simplest interface for AVR programming (you can find the board on Dontronics site). However I strongly suggest to use the buffered and safer "Avr ISP" interface above, especially for ISP, since the buffer go in Hi-Z state after programming.

8.1.7 Schnittstellen und Adapter für ConvCtrl

Im gehäuse der Wintergartensteuerung befinden sich zwei 9-polige SUB-D Schnittstellen eingebracht welche die RS232-Schnittstelle und das ISP-Interface nach außen führen und zugänglich machen. Das folgende Schaltbild zeigt die Beschaltung der 10-poligen Federleisten sowie deren Verbindung zum SUB-D-Stecker.

Für das ISP-Interface wird ein Adapter für 6-polig / 10-polig ISP benötigt damit das System mit dem AVRISP mkII von Atmel programmiert werden kann.

Beschaltung der Schnittstellen und Pinbelegung:



Schaltbild 8: ConvCtrl Schnittstellen und Adapter

Der ISP-Adapter für das AVRISP mkII:



Abbildung 27: AVRISP mkII Adapter

8.1.8 Das JTEG Interface

Zum Debugen der SW auf dem ATmega128 wird das JTAG-Interface inkl. AVR JTAG Emulator umgesetzt und auf dem Board mit integriert.

Die Beschaltung des JTAG Interface ist dem Kapitel „Gesamt-Basisbeschaltung ATmega128" zu entnehmen. In dem dort enthaltenen Schaltbild ist das Interface aufgeführt.

Als Kabelverbinder zwischen Target und JTAG-Interface wird der im vorherigen Kapitel beschriebene Pfostenverbinder in Schneidklemmtechnik inkl. Flachbandkabel eingesetzt.

JTAG Interface von ATMEL:

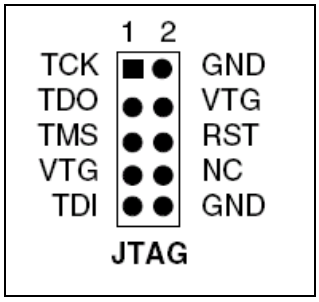


Abbildung 28: ATMEl JTAg Interface

Der JTAG Connector ist für den ATmega128(L) vorgesehen welcher ein built-in JTAG interface besitzt. Das Pinout des JTAG Connectors wird in obiger Abbildung dargestellt. Es ist kompatibel zum Pinout des ATMEL JTAG ICE Emulators.

Der Anschluss eines JTAG ICE Emulators auf diesen Anschluss erlaubt On-Chip-Debugging.

Mehr Informationen zum JTAG ICE und zum On-chip-Debugung ist dem JTAG ICE User Guide zu entnehmen welche über die Atmel Webseiten

http://www.atmel.com

erhältlich ist.

AVR JTAG Emulator:

Die Schaltung und die Informationen zum AVR JTAG Emulator sind aus der Webseite

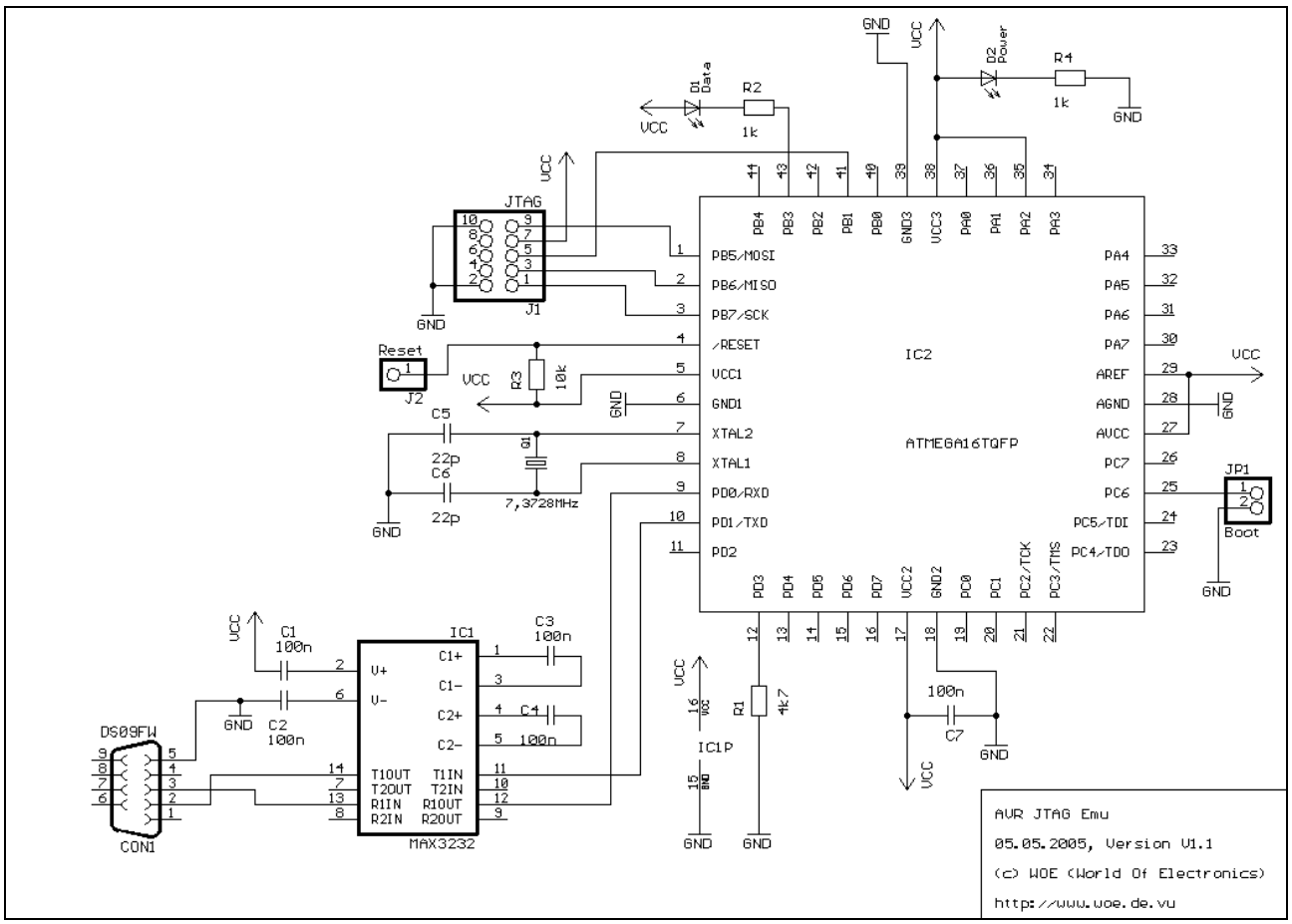
http://www.woe.onlinehome.de

Beim AVR JTAG Emu handelt es sich um einen Low-Cost JTAG-Debugger für die Mikrocontroller der ATmega-Serie, die On-Chip Debugging über eine JTAG-Schnittstelle unterstützen, z.B. unter anderem die Mikrocontroller ATmega16, ATmega162, ATmega32, ATmega128. Der AVR JTAG Emu wird über die serielle RS232-Schnittstelle am PC angeschlossen, das Debugging erfolgt über die kostenlose Software AVR-Studio von Atmel. Der AVR JTAG Emu muss zunächst mit einem geeigneten Bootloader (z.B. Evertool) programmiert werden, danach wird per AVR-Studio die aktuelle Firmware aufgespielt. Hierzu muss der Boot-Jumper JP1 gesteckt sein.

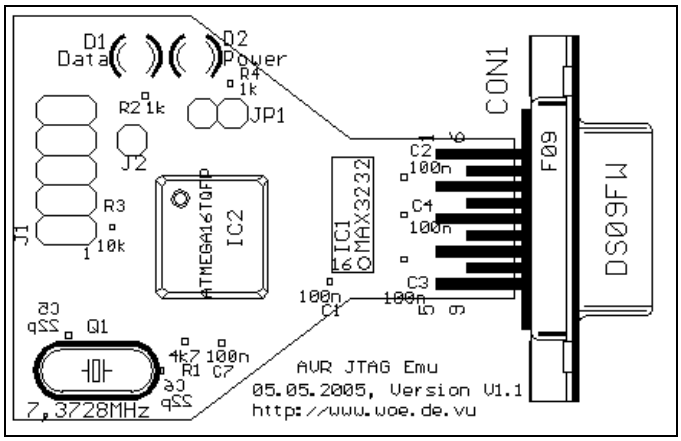
Die Schaltung sowie die Software stehen hier zum Download bereit:

Projektdateien für Hardware und Software im ZIP-Format:
http://www.woe.onlinehome.de/zip/ajemu.zip

Schaltplan und Bestückungsplan im GIF-Format:
http://www.woe.onlinehome.de/proj_gif/ajemu.gif
http://www.woe.onlinehome.de/proj_gif/ajemu_top_bot.gif



Schaltbild 9: Schaltplan AVR JTAG Emu



Schaltbild 10: Bestückungsplan AVR JTAg Emu

Leiterplatte:
Zu diesem Projekt ist eine professionell gefertigte unbestückte Leiterplatte unter der Bezeichnung AJE-MU.BRD erhältlich.

http://www.woe.onlinehome.de/woe_platinen.htm

Bauteile:

Stückliste: Basisbeschaltung ATmega mit ISP und RS232			
Widerstände		*Halbleiter*	
R1	SMD Widerstand 4k7 Ω	IC2	ATMEL AVR ATmega16 TQFT

R2, R4	SMD Widerstand 1k Ω	IC1	MAX3232 CSE
R3	SMD Widerstand 10k Ω	D1	Leuchtdiode GELB, Standard
		D2	Leuchtdiode GRÜN, Standard
Kondensatoren		*Sonstiges*	
C1, C2, C3, C4, C7	SMD Kondensator 100 nF	Q1	Quarz 7,3728 MhZ
C5, C6	SMD Kondensator 22pF	CON1	D-SUB-Buchse, 9-polig, gewinkelt, RM 9,4
		J1	Siftleiste 2x5 polig, RM2,54
		J2	Stiftleiste 1-polig
		JP1	Stiftleite 2-polig

Tabelle 38: Stückliste JTAG Emu

Die Widerstände und Kondensatoren sind im Rastermaß 0603 ausgeführt!

Hinweise zum AVR JTAG Emu:

Alle zu diesem Projekt gehörenden Informationen sind ohne Gewähr und ohne Anspruch auf Vollständigkeit. Der Nachbau erfolgt auf eigene Gefahr.

Die zu diesem Projekt gehörenden Dateien dürfen frei weitergegeben werden, jedoch ohne kommerzielle Nutzung und nur in vollständiger unveränderter Form, d.h. als *.ZIP-Datei.

Folgende Dateien sind in SED1335.ZIP enthalten:

* `AJEMU.SCH` Schaltplan des AVR JTAG Emu (im Eagle-Format)
* `AJEMU.BRD` Layout des AVR JTAG Emu (im Eagle-Format)
* `AJEMU.GIF` Schaltplan im GIF-Format
* `AJEMU_TOP_BOT.GIF` Bestückungsplan (top+bottom kombiniert) im GIF-Format
* `README.TXT` enthält Hinweise zum AVR JTAG Emu

Beschreibung:
Beim AVR JTAG Emu handelt es sich um einen Low-Cost JTAG-Debugger für die Mikrocontroller der ATmega-Serie, die On-Chip Debugging über eine JTAG-Schnittstelle unterstützen, z.B. unter anderem die Mikrocontroller ATmega16, ATmega162, ATmega32, ATmega128.

Der AVR JTAG Emu wird über die serielle RS232-Schnittstelle am PC angeschlossen, das Debugging erfolgt über die kostenlose Software AVR-Studio von Atmel. Der AVR JTAG Emu muss zunächst mit einem Bootloader programmiert werden, danach wird per AVR-Studio die aktuelle Firmware aufgespielt (hierzu Boot-Jumper JP1 setzen). Ein geeigneter Bootloader ist im Internet zu finden, z.B. Evertool.

Erweiterte Informationen vom Autor des AVR JTAG Emulators:

Man kann den ATmega16 TQFT in eingelöteten Zustand per ISP programmieren. Die meisten dafür benötigten Leitungen sind schon auf dem 10-poligen Steckverbinder auf der Platine vorhanden (MOSI, MISO, SCK, VCC, GND). Was fehlt ist der Resetpin, dieser ist über J2 zugänglich.

Als Programmiertool kann man z.B. den AVRISP von Atmel benutzen oder sonstige Tools wie z.B. Ponyprog.

Der AVR JTAG Emu muss zunächst mit einem geeigneten Bootloader (z.B. Evertool) programmiert werden, danach wird per AVR-Studio die aktuelle Firmware aufgespielt. Hierzu muss der Boot-Jumper JP1 gesteckt sein.

Evertool-Light gibt es z.B. hier:
http://www.siwawi.arubi.uni-kl.de/avr_projects/evertool/index.html

Für weitere Informationen zum Emulator siehe Kapitel „Evertool“ im Anhang.
[Evertool]

8.1.9 PIN-Belegung des PC Parallel-Port

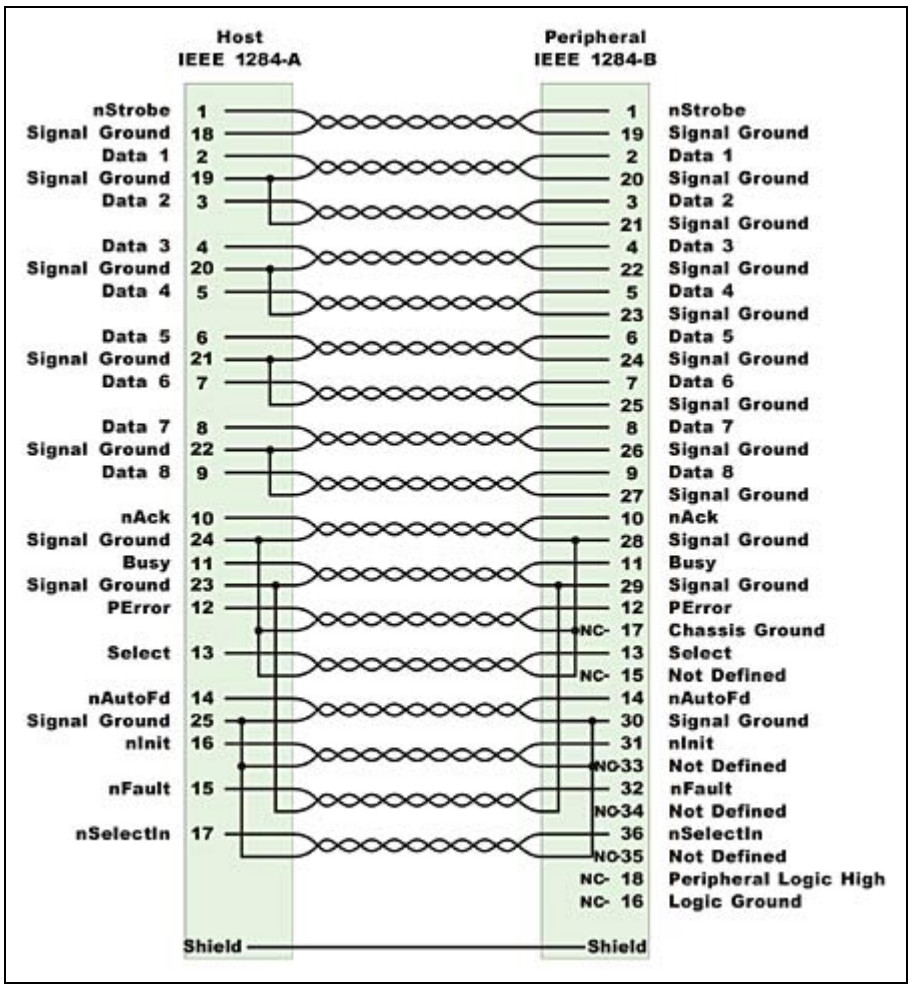
Pinbelegung des LPT-Ports			
Pin	*Name*	*Richtung*	*Funktion*
1	STROBE	>	Strobe
2	data0	<>	Address, Data or RLE Data Bit 0
3	data1	<>	Address, Data or RLE Data Bit 1
4	data2	<>	Address, Data or RLE Data Bit 2
5	data3	<>	Address, Data or RLE Data Bit 3
6	data4	<>	Address, Data or RLE Data Bit 4
7	data5	<>	Address, Data or RLE Data Bit 5
8	data6	<>	Address, Data or RLE Data Bit 6
9	data7	<>	Address, Data or RLE Data Bit 7
10	ACK	<	Acknowledge
11	BUSY	<	Busy
12	PError	<	Paper End
13	Select	<	Select
14	AutoFd	>	Autofeed
15	Fault	<	Error
16	Init	>	Initialize
17	SelectIn	>	Select In
18	GND	–	Signal Ground
19	GND	–	Signal Ground
20	GND	–	Signal Ground
21	GND	–	Signal Ground
22	GND	–	Signal Ground
23	GND	–	Signal Ground
24	GND	–	Signal Ground
25	GND	–	Signal Ground

Tabelle 39: PIN-Belegung des LPT-Ports

> bedeutet: vom PC zum Gerät, < bedeutet: vom Gerät zum PC, <> bedeutet: bidirektionale Signalleitung. Datenflussrichtung wird ausgehandelt; –: Masseleitung (ohne Datenflussrichtung)

Die Datenübertragung erfolgt nach folgendem Protokoll:

1. Der Sender überprüft anhand der Busy-Leitung, ob der Empfänger bereit ist. Falls ja, wird das Datenbyte auf die Datenleitungen gelegt.
2. Durch Aktivierung des Strobe-Signals (mind. 1 bis max. 50 Mikrosekunden) wird dem Empfänger die Gültigkeit des Datenbytes mitgeteilt.
3. Durch Aktivierung von Busy signalisiert der Empfänger, dass er mit der Verarbeitung der Daten beschäftigt ist.
4. Der erfolgreiche Empfang und Verarbeitung des Datenbytes wird vom Endgerät durch das Acknowledge-Signal bestätigt.



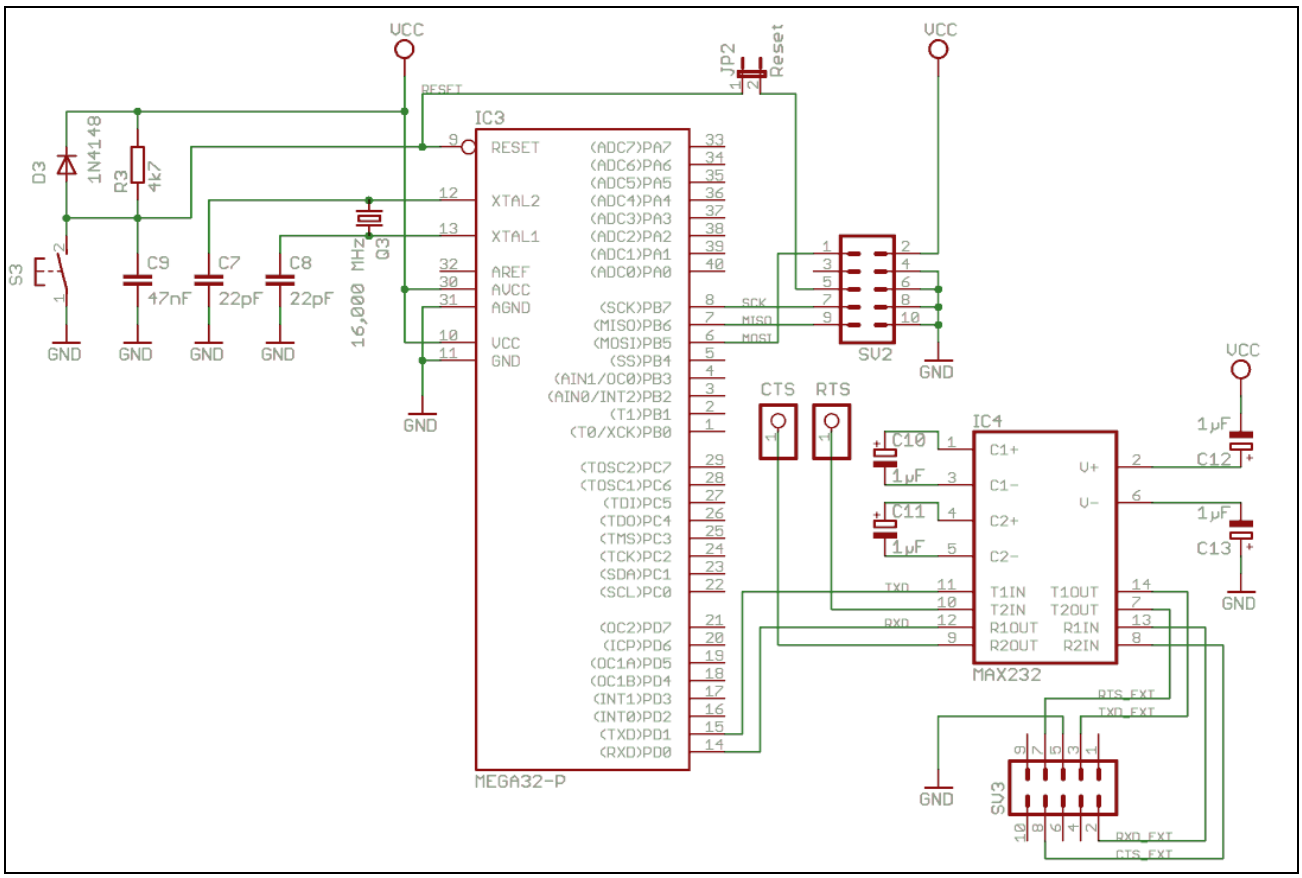
Schaltbild 11: Vorgeschriebene Belegung eines IEEE 1284 AB-Kabels

8.1.10 RS232-Interface

Für die serielle Datenübertragung zum und vom ATmega wird ein serielles Interface nach RS232 Standard mit einem Maxime MAX232 Baustein umgesetzt.

Die serielle Datenübertragung dient auch für die Möglichkeit, Loggingdaten aus dem externen EEPROM an einen PC zur Späteren Messdatenauswertung zu übertragen.

Ergänzung der ATmega Basisbeschaltung um ISP und RS232:



Schaltbild 12: ATmega Basisbeschaltung mit ISP und RS232-Schnittstelle

Bauteile:

Stückliste: Basisbeschaltung ATmega mit ISP und RS232			
Widerstände		*Halbleiter*	
R3	4k7 Ω	IC3	ATMEL AVR ATmega32
		D3	Diode 1N4148
		IC4	Maxim MAX232
Kondensatoren		*Sonstiges*	
C7, C8	Keramikkondensator 22 pF	S3	Taster
C9	Keramikkondensator 47 nF	Q3	Quarz 16.000.000 Hz
C10, C11, C12, C13	Elko 1µF/16V	SV2; SV3	Pin Header MA05-2
		JP2	Jumper 2-polig, 2,54mm Raster
		CTS, RTS	IC-Sockelleiste als Patchfeld

Tabelle 40: Stückliste Basisbeschaltung ATmega mit ISP uns RS232

Als Kabelverbinder für die serielle Schnittstelle mit Gehäusedurchführung wird eine 9-polige Buchsenleiste D-SUB für Flachbandkabel mit einem PIN-Header MA05-2 verwendet.



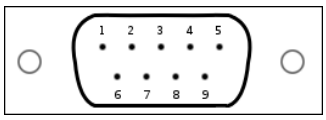
D-Sub-Steckverbinder, für Flachkabel, Flache Ausführung.
9-polige D-Subminiatur-Buchse für Flachbandkabel mit vergoldeten Kontakten.

Bestellnumer Reichelt Elektronik: *D-SUB BU 09FB*

Abbildung 29: D-SUB Buchse 9-polig für RS232

8.1.11 PIN-Belegung 9-polige RS232

RS232 Buchse 9-polig:



RS232 Stecker 9-polig:

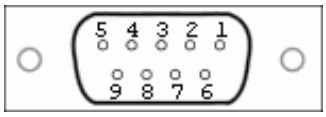


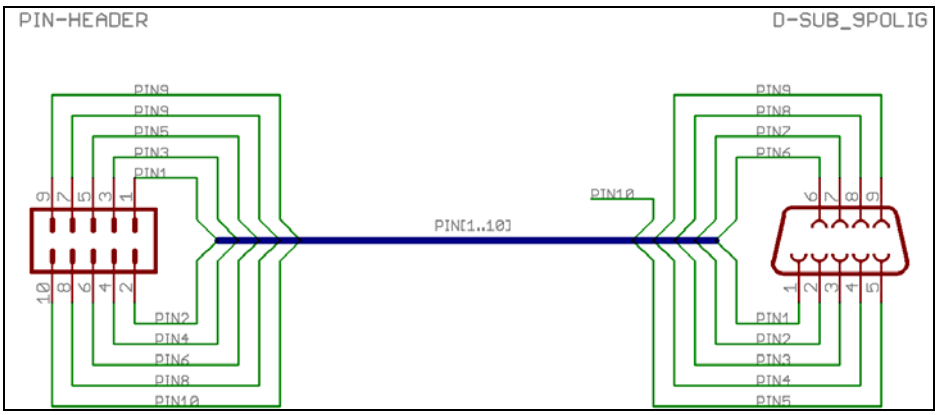
Tabelle 41: PIN-Belegung der 9-poligen RS232 Stecker/Buchse

Abkürzung	*Name*	*Beschreibung*	*Pin-Nr. 25-pol.*	*Pin-Nr. 9-pol.*	*In/Out*
--	Common Ground	Gemeinsame Abschirmmasse (nicht Datenmasse)	Pin 1	—	—
TxD,TX,TD	Transmit Data	Leitung für ausgehende (gesendete) Daten.	Pin 2	Pin 3	Out
RxD,RX,RD	Receive Data	Leitung für den Empfang von Daten.	Pin 3	Pin 2	In
RTS	Request to Send	„Sendeanforderung"; Eine logische Null an diesem Ausgang signalisiert der Gegenstelle, dass sie Daten Senden kann	Pin 4	Pin 7	Out
CTS	Clear to Send	Eine logische Null an diesem Eingang ist ein Signal der Gegenstelle, dass sie Daten entgegennehmen kann	Pin 5	Pin 8	In
DSR	Dataset Ready	Ein angeschlossenes Gerät signalisiert dem Computer, dass es einsatzbereit (nicht notwendigerweise empfangsbereit) ist, wenn eine logische Null auf dieser Leitung anliegt.	Pin 6	Pin 6	In
GND	Ground	Signalmasse. Die Signalspannungen werden gegen diese Leitung gemessen.	Pin 7	Pin 5	—
DCD,CD	(Data) Carrier Detect	Ein Gerät signalisiert dem Computer, dass es einlaufende Daten auf der Leitung erkennt	Pin 8	Pin 1	In
DTR	Data Terminal Ready	Über diese Leitung signalisiert der PC dem Gerät, dass er betriebsbereit ist. Damit kann ein Gerät eingeschaltet oder zurückgesetzt werden. (Üblicherweise schaltet ein Gerät z.B. Modem diese Leitung auf DSR durch, wenn es einsatzbereit ist)	Pin 20	Pin 4	Out
RI	Ring Indicator	Das Gerät zeigt dem PC an, dass ein Anruf ankommt ("ring" ist engl. für "klingeln"; besonders bei Modems)	Pin 22	Pin 9	In

Tabelle 42: PIN-Belegung des seriellen RS232-Ports

In/Out wird auch Sicht vom PC aus gesehen.

8.1.12 Verbindungskabel RS232 PIN-Header / D-SUB9



Schaltbild 13: Verbindungskabel zwischen PIN-Header und D-SUB9 Buchse

Bauteile:

Stückliste: Basisbeschaltung ATmega mit ISP und RS232		
Sonstiges		
PIN-HEADER	PFL10 Pfostensteckverbinder	
D-SUB_9POLIG	SUB-D-Buchse 9-polig.	

Tabelle 43: Stückliste Basisbeschaltung ATmega mit ISP uns RS232

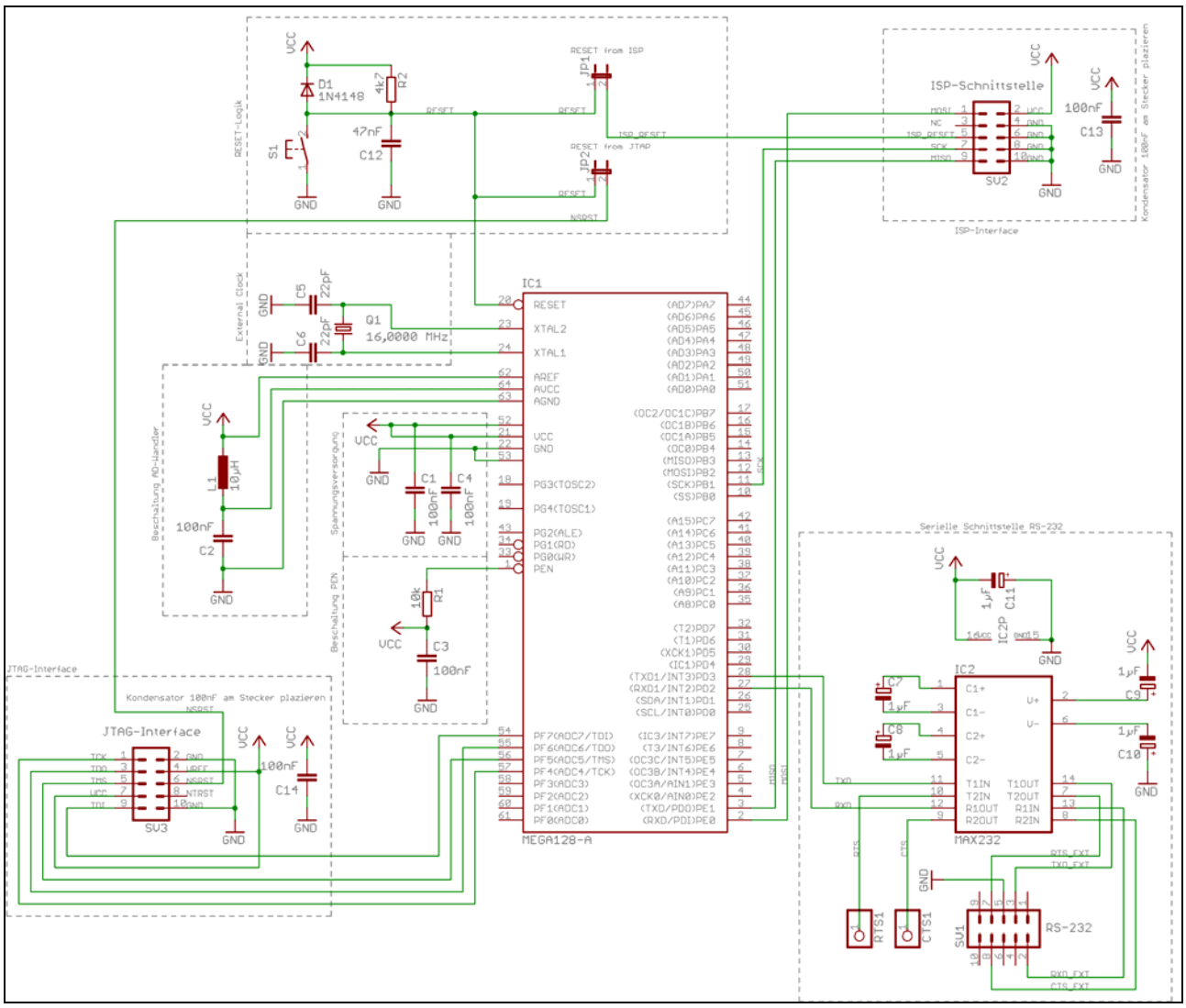
8.1.13 Gesamt-Basisbeschaltung ATmega128

Die Gesamtbeschaltung des ATmega128 weicht von der des ATmega32 ab. Beim ATmega32 können die Leitungen des ISP-Interface direkt an MISO und MOSI des Kontrollers angeschlossen werden. Beim ATmega128 sind diese nicht direkt belegt und intern nicht verdrahtet. Achtung! Beim ATmega128 findet die Beschaltung über Port E und die Leitungen RXT0 und TXD0 statt!

Weiter wurde für die Entwicklung mit dem ATmega128 das JTAG Interface ausgeführt damit ein Debugger eingesetzt werden kann.

Der PEN Eingang (Pin 1) wird mit einem 1k Wiederstand auf VCC gezogen!

Für die Beschaltung der ISP- und JTAG-interfaces wird eine Entkopplungskondensator am Stecker vorgesehen der als solches auch nahme beim/am Stecker bestückt werden muss!



Schaltbild 14: Gesamt-Grundbeschaltung des ATmega128

Bauteile:

Stückliste: Basisbeschaltung ATmega mit ISP und RS232			
Widerstände		***Halbleiter***	
R1	10k Ω	IC1	ATMEL AVR ATmega128
R2	4k7 Ω	D1	Diode 1N4148
		IC2	Maxim MAX232
Kondensatoren		***Sonstiges***	
C1, C2, C3, C4, C13, C14	100 nF	S1	Taster
C5, C6	Keramikkondensator 22 pF	Q1	Quarz 16.000.000 Hz
C7, C8, C9, C10, C11	Elko 1µF/16V	SV1, SV2; SV3	Pin Header MA05-2
C12	Keramikkondensator 47 nF	JP1, JP2	Jumper 2-polig, 2,54mm Raster
Induktivitäten		CTS1, RTS1	IC-Sockelleiste als Patchfeld
L1	Spule 10 µH		

Tabelle 44: Stückliste Basisbeschaltung ATmega mit ISP uns RS232

Themenbezogene LED's:

LED / Funktion / Beschreibung	*Abkürzung*	*Farbe*	
Alive LED	LED_ALIVE	grün	*[LED3]*

Timer-Tick für Sekundentakt	LED_SEC_TICK	grün	*[LED27]*

Tabelle 45: Themenbezogene LED's für Gesamtbeschaltung

Für die Gesamtbeschaltung eines ATmega128 wurde ein eigenes Evaluation-Board entwickelt. Siehe hierzu „Das ATmega128 Evaluation-Board" im Kapitel Hardware.

8.1.14 AVR Fusebits Tutorial

Einleitung - Was sind Fusebits?

Fusebits sind im Grunde genommen nichts anderes als Speicherzellen die man Programmieren und Löschen kann. Sie dienen jedoch nicht zur Speicherung von Daten sondern mit ihrer Hilfe kann das Verhalten des AVR beeinflusst werden. Zum Beispiel können bestimmte Funktionen aktiviert und deaktiviert werden.

Was ist zu beachten?

Bevor man zum ersten mal die Fusebits eines AVR verändert, sollte man wissen das man den Controller damit nicht zerstören kann. Es ist jedoch möglich den Controller so einzustellen das man mit den normalen Werkzeugen nicht mehr darauf zugreifen kann. Grundsätzlich kann ein "verfuseter" Controller mit dem richtigen Werkzeug aber wieder repariert werden. Ob dies wirtschaftlich jedoch sinnvoll ist steht auf einem anderen Blatt.

Was anfänglich Probleme verursacht ist die "invertierte Logik" der Fusebits. Ein programmiertes Fusebit ist nicht wie man annehmen könnte auf "1" gesetzt sondern auf "0". Eine unprogrammierte Fuse ist "1". Manche Programme stellen programmierte Fusebits mit einem gesetzten Häkchen dar (z.B. PonyProg), welches dann als eine 0 (=programmed) interpretiert wird. Möchte man das Fusebit "setzen" oder "programmieren" muss man das Häkchen setzen.

Handelt es sich um ein Enable Fusebit bedeutet ein programmiertes Fusebit (0), das die Funktion eingeschaltet ist. Ist es ein Disable Fusebit bedeutet ein programmiertes Fusebit (0), das die Funkiton ausgeschaltet ist.

Was braucht man?

Die Fusebits lassen sich über ISP, JTAG oder parallel programmieren. Man kann hierfür die gleiche Hardware und Software verwenden wie zum Programmieren der des Flash Speichers oder des Eeproms, zum Beispiel PonyProg oder das AVR Studio.

Die Fusebytes

Je nach Controller stehen bis zu drei Fusebytes zur Verfügung: Fuse Low Byte, Fuse High Byte und Extended Fuse Byte. Einige der älteren Controller haben nur ein Fuse Byte und sehr wenige Fusebits.

Die Fusebits

Die folgende Beschreibung listet alle Fusebits auf die es bei den AVR Controllern gibt. Es gibt jedoch keinen Controller in dem alle Fusebits gleichzeitig zu finden sind. Ältere Controller vom Typ AT90S kennen teilweise nur 2 Fusebits.

CKSEL

Die wohl am häufigsten geänderten Fusebits sind CKSEL0 bis CKSEL3 (Select Clock Source). Mit ihrer Hilfe wählt man die Taktquelle aus der der Controller seinen Takt erhält. Hier ist etwas Vorsicht geboten da eine falsche Einstellung den Controller lähmen kann. Eine falsche Einstellung lässt sich jedoch relativ leicht beheben. Die genauen Parameter können zwischen den einzelnen Typen variieren
Default: Interner RC Oszillator mit 1MHz (bzw 8MHz bei Typen mit Vorteiler)

```
CKSEL0 : 0 (programmiert)
CKSEL1 : 1 (unprogrammiert)
```

```
CKSEL2 : 1 (unprogrammiert)
CKSEL3 : 1 (unprogrammiert)
```

SUT

Mit SUT0 und SUT1 lässt sich die Zeit einstellen wie lange der Reset Impuls nach einem Reset oder Power Up verzögert wird. Je nach Umgebungsbedinung kann die Reset Zeit verlängert oder verkürzt werden. Zusammen mit der Brown Out Detection wird eine externe Resetschaltung (bis auf den üblichen 10kOhm PullUp Widerstand) meist überflüssig.

Default:

```
SUT0 : 0 (programmiert)
SUT1 : 1 (unprogrammiert)
```

CKDIV8

Divide Clock by 8 ist etwas irreführend. Wenn dieses Fusebit gesetzt ist wird ein Vorteiler aktiviert, der den Takt für den Controller durch 8 teilt. Es ist jedoch möglich diesen Vorteiler auf einen anderen Wert einzustellen. Dies ist dann sinnvoll wenn der Controller aus einer externen Taktquelle gespeist werden soll, die Frequenz aber zu hoch ist. Details dazu im Artikel Taktquelle.

Default:

```
CKDIV8 : 0 (programmiert)
```

CKOUT

Wird diese Fuse programmiert wird der CPU Takt an dem entsprechenden CLKO Pin ausgegeben.

Default:

```
CKOUT : 1 (unprogrammiert)
```

CKOPT

CKOPT kommt zum Einsatz wenn der AVR von einem externen Quarz getaktet wird. Wird CKOPT programmiert (0) schwingt der Oszillator mit der maximalen Amplitude. Dies kann notwendig werden wenn der AVR in einer Umgebung mit vielen Störsignalen betrieben werden soll. Ist CKOPT unprogrammiert (1) schwingt der Oszillator mit einer niedrigeren Amplitude. Dadurch verringert sich die Stromaufnahme und die Störabstrahlung.

Default:

```
CKOPT : 1 (unprogrammiert)
```

RSTDISBL

Dieses Fuse Bit steuert die Funktion des Reset Pin. Wird es programmiert kann man den Reset Pin als normalen IO Pin verwenden.

Achtung: Wird dieses Bit programmiert kann der Controller nicht mehr über die ISP Schnittstelle erreicht werden

Default:

```
RSTDISBL : 1 (unprogrammiert)
```

SPIEN

Mit SPIEN kann die ISP Schnittstelle abgeschaltet werden. Dieses Fusebit lässt sich nur über die parallele Programmierung ändern. Ist die ISP Schnittstelle einmal abgeschaltet kann der Controller nicht mehr über ISP erreicht werden.

Default:

```
SPIEN : 0 (programmiert)
```

JTAGEN

JTAGEN aktiviert/deaktiviert die JTAG Schnittstelle.

Default:
JTAGEN : 0 (programmiert)

DWEN

DWEN aktiviert/deaktiviert die debugWire Schnittstelle.

Default:
```
DWEN : 1 (unprogrammiert)
```

OCDEN

OCDEN aktiviert/deaktiviert das On-Chip Debug System. Das On-Chip Debug System kann unabhängig von der JTAG Schnittstelle deaktiviert werden. Bei abgeschaltetem OCD kann der Controller über JTAG nur programmiert werden.

Default:
```
OCDEN : 1 (unprogrammiert)
```

EESAVE

Wird EESAVE programmiert wird das EEprom bei einem Chip Erase vor dem Löschen geschützt. Ein Chip Erase löscht normalerweise den kompletten Speicher.

Default:
```
EESAVE : 1 (unprogrammiert)
```

BODEN

BODEN aktiviert/deaktiviert die Brown Out Detection. Bei manchen Controllern wird diese Funktion durch die BODLEVEL Fusebits übernommen.

Default:
```
BODEN : 1 (unprogrammiert)
```

BODLEVEL

Mit BODLEVEL kann der Spannungswert eingestellt werden bei dem der Unterspannungsschutz aktiv werden soll. Ältere Controller (zb ATmega128) haben nur zwei Schwellwerte. Mit BODLEVEL kann zwischen den Werten gewechselt werden, mit BODEN wird die Funktion komplett deaktiviert. Neuere Controller (zb ATmega168) haben 3 BODLEVEL Fusebits mit denen mehrere Schwellwerte eingestellt werden können bzw die gesamte Funktion deaktiviert wird. Ab Werk ist bei allen Typen die BOD Funktion abgeschaltet.

Default:
```
BODLEVEL : 1 (unprogrammed)
```

WDTON

Mit WDTON kann der Watchdog Timer permanent aktiviert werden. Ist dieses Fusebit nicht programmiert (1) kann der Watchdog per Software gesteuert werden.

Default:
```
WDTON : 1 (unprogrammiert)
```

BOOTRST

BOOTRST bestimmt an welche Adresse nach einem Reset gesprungen wird. Unprogrammiert (1) springt der Controller nach einem Reset an Adresse $0000. Wird das Fusebit programmiert springt der Controller nach einem Reset an den Beginn des Bootloaders. Die Adresse hängt vom Controller und von den Einstellungen der BOOTSZ Fusebits ab.

Default:
`BOOTRST : 1 (unprogrammiert)`

BOOTSZ

Mit BOOTSZ wird die Größe des Speicherbereiches bestimmt der für den Bootloader reserviert wird. Die Größe ist abhängig vom Controllertyp. Dieser Speicherbereich befindet sich immer am Ende des Flash Adressraumes..
Default:
`BOOTSZ : siehe Datenblatt`

Compatibility Bits

Viele Controller haben ein Compatibility Bit. Mit diesem Bit lässt sich der Controller in einen Modus versetzen in dem er sich exakt so verhält wie sein Vorgänger. Beim ATmega128 gibt es zB das M103C Bit. Der ATmega128 verhält sich also wie ein ATmega103.
Ob das Compatibility Bit ab Werk programmiert ist oder nicht hängt vom Controller ab.

SELFPRGEN

SELFPRGEN aktiviert/deaktiviert die Self Programming Funktion.
Default:
`SELFPRGEN : 1 (unprogrammiert)`

HWBEN

HWBEN aktiviert/deaktiviert die Hardware Boot Funktion
Default:
`HWBEN : 0 (programmiert)`

Sollte ich ein Fusebit vergessen haben oder neue dazukommen bitte ergänzen.

Fusebits mit dem AVR Studio programmieren:

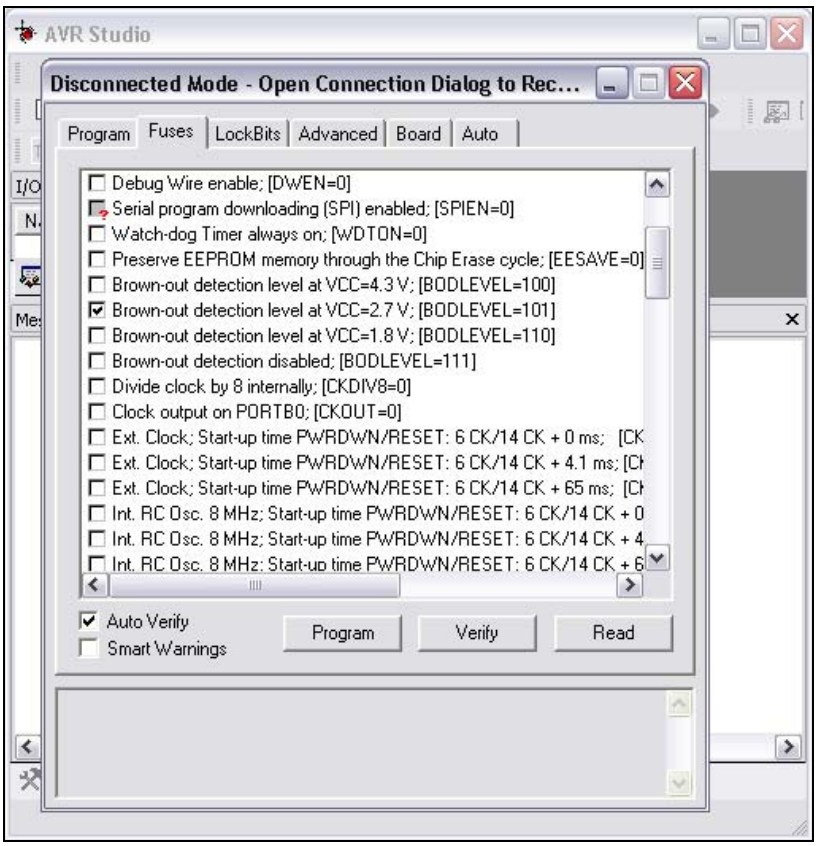


Abbildung 30: Fusebits im AVR Studio

Am einfachsten und intuitivsten lassen sich die Fusebits mit dem AVR Studio programmieren. Für jedes Fusebit gibt es eine kurze Beschreibung und den Default Wert. Gibt es für eine Funktion mehrere Fusebits wird für jede Kombination ein Häkchen mit Beschreibung und Bitkombination angezeigt.

In diesem Beispiel (ATmega168) sieht man zb das die Brown Out Detection auf 2,7V eingestellt wurde.

Fusebits mit PonyProg programmieren:

PonyProg ist schon etwas umständlicher beim Umgang mit Fusebits. Hier gibt es für jedes Fusebit ein eigenes Häkchen. Hier ist zu beachten das ein Häkchen bedeutet das das Bit programmiert (0) ist, kein Häkchen bedeutet unprogrammiert (1).

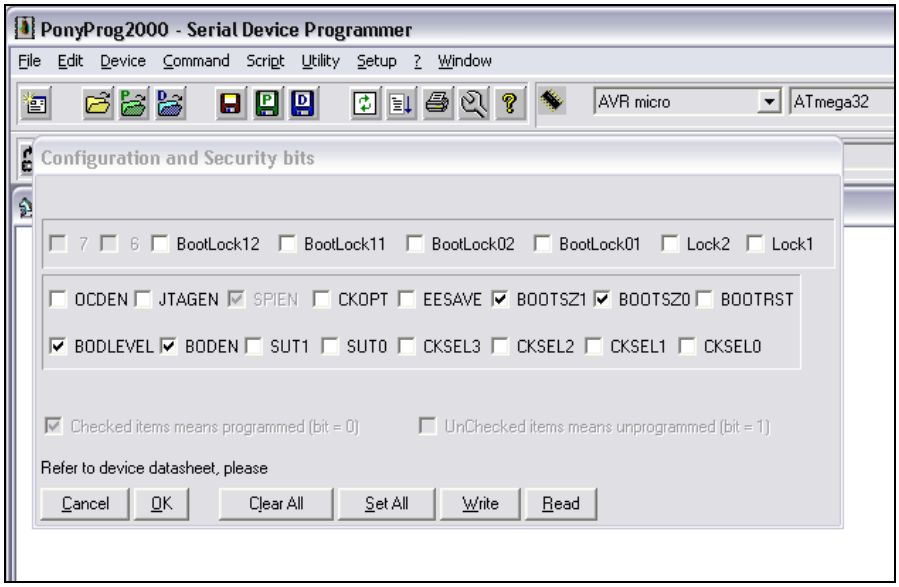


Abbildung 31: PonyProg Fuse Bits

In diesem Beispiel (ATmega32) wurden CKSEL und SUT so konfiguriert das ein externer 16MHz Quarz und maximale Resetverzögerung aktiviert ist.

Die AVRDUDE GUI:

Als Beispiel für ein Konsolenprogramm soll AVRDUDE dienen. Für dieses Programm gibt es eine grafische Oberfläche. Hier können einzelne Fusebits nicht direkt eingestellt werden. Statt dessen muss eine HEX Zahl für das gesamte Fusebyte angegeben werden.

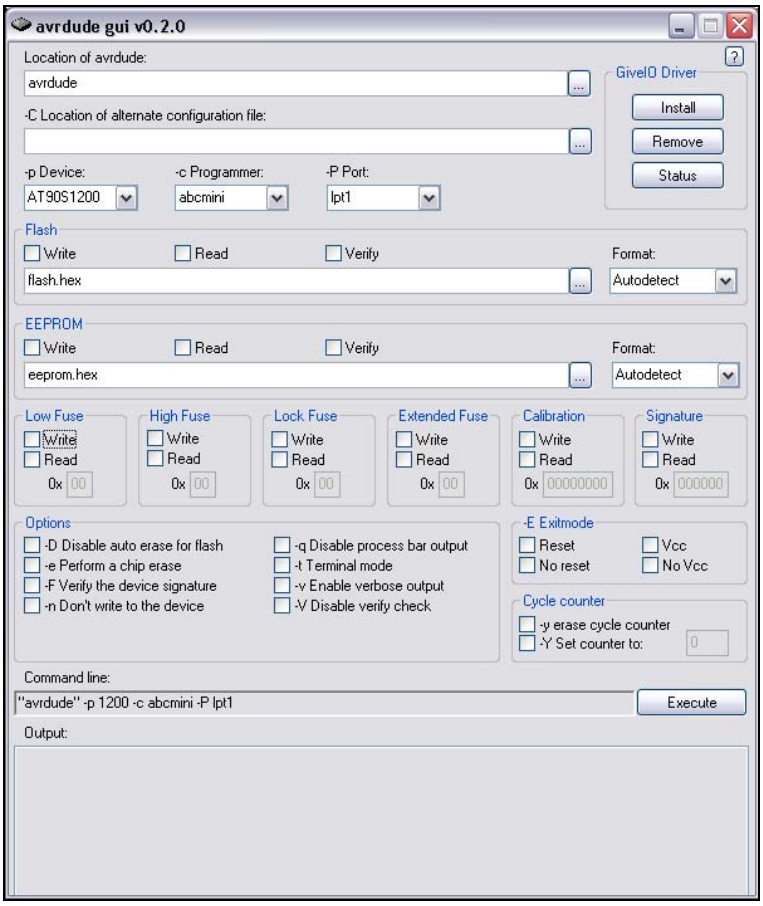


Abbildung 32: AVRDUDE

Sollen Fusebits beschrieben werden muss bei Low Fuse, High Fuse oder Extendes Fuse bei "Write" ein Haken gesetzt werden. In das Eingabefeld darunter muss der entsprechende HEX Wert für das gesamte Byte eingegeben werden. Bei er Ermittlung der richtigen Werte kann der AVR Fuse Calculator.

Zu Beachten ist hierbei, das alle Bits in dem Byte mit den vorgegebenen Werten beschrieben werden. Eventuelle andere Konfigurationen werden überschrieben.

Fusebits mit AVRDUDE programmieren:

Um mit AVRDUDE die Fusebits von der Kommandozeile aus zu programmieren muss man ebenfalls HEX Werte in die entsprechenden Fusebytes schreiben. Auch hierbei hilft der AVR Fuse Calculator.

Programmieren der Low, High und Extendes Fusebytes mit AVRDUDE, ATmega32 und STK200 Programmieradapter.

```
avrdude -p m32 -c pony-stk200 -P lpt1 -U lfuse:w:0xFF:m
avrdude -p m32 -c pony-stk200 -P lpt1 -U hfuse:w:0xD9:m
```

Dieses Beispiel mit dem AVR Fuse Calculator

AVR Fuse Calculator

Von http://www.wiki.elektronik-projekt.de/w/index.php/AVR_Fusebits_Tutorial

8.1.15 AVR Fuse Konfiguration ATmega32

Fuse Bytes des ATmega32:

Fuse High Byte	Bit No.	Description	Default Value
OCDEN[4]	7	Enable OCD	1 (unprogrammed, OCD disabled)
JTAGEN[5]	6	Enable JTAG	0 (programmed, JTAG enabled)
SPIEN[1]	5	Enable SPI Serial Program and Data Downloading	0 (programmed, SPI prog. enabled)
CKOPT[2]	4	Oscillator options	1 (unprogrammed)
EESAVE	3	EEPROM memory is preserved through the Chip Erase	1 (unprogrammed, EEPROM not preserved)
BOOTSZ1	2	Select Boot Size (see Table 99 for details)	0 (programmed)[3]
BOOTSZ0	1	Select Boot Size (see Table 99 for details)	0 (programmed)[3]
BOOTRST	0	Select reset vector	1 (unprogrammed)

Notes:
1. The SPIEN Fuse is not accessible in SPI Serial Programming mode.
2. The CKOPT Fuse functionality depends on the setting of the CKSEL bits. See See “Clock Sources” on page 25. for details.
3. The default value of BOOTSZ1..0 results in maximum Boot Size. See Table 99 on page 255.
4. Never ship a product with the OCDEN Fuse programmed regardless of the setting of Lock bits and the JTAGEN Fuse. A programmed OCDEN Fuse enables some parts of the clock system to be running in all sleep modes. This may increase the power consumption.
5. If the JTAG interface is left unconnected, the JTAGEN fuse should if possible be disabled. This to avoid static current at the TDO pin in the JTAG interface.

Fuse Low Byte	Bit No.	Description	Default Value
BODLEVEL	7	Brown-out Detector trigger level	1 (unprogrammed)
BODEN	6	Brown-out Detector enable	1 (unprogrammed, BOD disabled)
SUT1	5	Select start-up time	1 (unprogrammed)[1]
SUT0	4	Select start-up time	0 (programmed)[1]
CKSEL3	3	Select Clock source	0 (programmed)[2]
CKSEL2	2	Select Clock source	0 (programmed)[2]
CKSEL1	1	Select Clock source	0 (programmed)[2]
CKSEL0	0	Select Clock source	1 (unprogrammed)[2]

Notes:
1. The default value of SUT1..0 results in maximum start-up time. SeeTable 10 on page 30 for details.
2. The default setting of CKSEL3..0 results in internal RC Oscillator @ 1MHz. See Table 2 on page 25 for details.

Festlegung der Fuse-Bits für das Projekt ConvCtrl:

Fuse High Byte:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OCDEN	JTAGEN	SPIEN	CKOPT	EESAVE	BOOTSZ1	BOOTSZ0	BOOTRST
0	0	0	1	0	0	0	1

Tabelle 46: Fuse High Byte ATmega 128

Fuse Low Byte:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BODLEVEL	BODEN	SUT1	SUT0	CKSEL3	CKSEL2	CKSEL1	CKSEL0
1	1	1	1	1	1	1	1

Tabelle 47: Extended Fuse Byte ATmega 128

Bemerkung:
0 = programmiert (!! negative Logik !!)
1 = unprogrammiert (!! negative Logik !!)

Default (keine Programmierung)
Wichtige Programmierung (abweichend von Default)
Sonstige Systemeinstellung abweichend von Default)

CKSEL3-0 ist im Originalzustand laut Datenblatt auf 0001. Das entspricht dem internen Oszillator mit 1MHz. Uns interessiert aber der Zustand mit externem Quarz mit 16 MHz. Dazu muss CKSEL3-1 auf 111 gesetzt werden. Die Start-Up-Time wird durch CKSEL0 und SUT bestimmt. Eine 1 auf CKSEL0 bedeutet zusammen mit 11 auf SUT1-0 die längstmögliche Startverzögerung. So hat der Quarz die meiste Zeit zum Anschwingen. Wie beim Atmega üblich, bedeutet eine Null ein Häkchen und bei einer Eins bleibt das Feld leer.

Weitere Infos hierzu siehe im nächsten Kapitel für ATmega128.

8.1.16 AVR Fuse Konfiguration ATmega128

Der ATmega128 besitzt drei Fuse-Bytes

Fuse Bytes des ATmega128:

Extended Fuse Byte	Bit No.	Description	Default Value
–	7	–	1
–	6	–	1
–	5	–	1
–	4	–	1
–	3	–	1
–	2	–	1
M103C(1)	1	ATmega103 compatibility mode	0 (programmed)
WDTON(2)	0	Watchdog Timer always on	1 (unprogrammed)

Notes: 1. See "ATmega103 and ATmega128 Compatibility" on page 4 for details.
2. See "Watchdog Timer Control Register – WDTCR" on page 55 for details.

Fuse High Byte	Bit No.	Description	Default Value
OCDEN[4]	7	Enable OCD	1 (unprogrammed, OCD disabled)
JTAGEN[5]	6	Enable JTAG	0 (programmed, JTAG enabled)
SPIEN[1]	5	Enable Serial Program and Data Downloading	0 (programmed, SPI prog. enabled)
CKOPT[2]	4	Oscillator options	1 (unprogrammed)
EESAVE	3	EEPROM memory is preserved through the Chip Erase	1 (unprogrammed, EEPROM not preserved)
BOOTSZ1	2	Select Boot Size (see Table 112 for details)	0 (programmed)[3]
BOOTSZ0	1	Select Boot Size (see Table 112 for details)	0 (programmed)[3]
BOOTRST	0	Select Reset Vector	1 (unprogrammed)

Notes:
1. The SPIEN fuse is not accessible in SPI Serial Programming mode.
2. The CKOPT fuse functionality depends on the setting of the CKSEL bits. See "Clock Sources" on page 37 for details.
3. The default value of BOOTSZ1..0 results in maximum Boot Size. See Table 112 on page 287
4. Never ship a product with the OCDEN Fuse programmed regardless of the setting of lock bits and the JTAGEN Fuse. A programmed OCDEN Fuse enables some parts of the clock system to be running in all sleep modes. This may increase the power consumption.
5. If the JTAG interface is left unconnected, the JTAGEN fuse should if possible be disabled. This to avoid static current at the TDO pin in the JTAG interface.

Fuse Low Byte	Bit No.	Description	Default Value
BODLEVEL	7	Brown out detector trigger level	1 (unprogrammed)
BODEN	6	Brown out detector enable	1 (unprogrammed, BOD disabled)
SUT1	5	Select start-up time	1 (unprogrammed)[1]
SUT0	4	Select start-up time	0 (programmed)[1]
CKSEL3	3	Select Clock source	0 (programmed)[2]
CKSEL2	2	Select Clock source	0 (programmed)[2]
CKSEL1	1	Select Clock source	0 (programmed)[2]
CKSEL0	0	Select Clock source	1 (unprogrammed)[2]

Notes:
1. The default value of SUT1..0 results in maximum start-up time. See Table 14 on page 41 for details.
2. The default setting of CKSEL3..0 results in Internal RC Oscillator @ 1 MHz. See Table 6 on page 37 for details.

Festlegung der Fuse-Bits für das Projekt ConvCtrl:

Der Atmega schwingt mit den internen Oszillator auf ca. 1MHz. Das ist für die meisten Anwendungen zu ungenau und schlicht zu langsam.

Extended Fuse Byte:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	--	--	--	M103C	WDTON
1	1	1	1	1	1	1	1

Tabelle 48: Fuse Low Byte ATmega 128

Fuse High Byte:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OCDEN	JTAGEN	SPIEN	CKOPT	EESAVE	BOOTSZ1	BOOTSZ0	BOOTRST
0	0	0	1	0	0	0	1

Tabelle 49: Fuse High Byte ATmega 128

Fuse Low Byte:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BODLEVEL	BODEN	SUT1	SUT0	CKSEL3	CKSEL2	CKSEL1	CKSEL0
1	1	1	1	1	1	1	1

Tabelle 50: Extended Fuse Byte ATmega 128

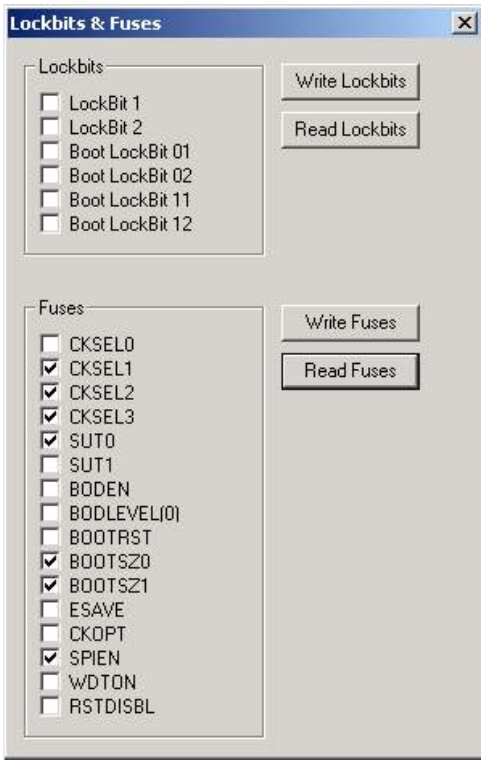
Bemerkung:
0 = programmiert (!! negative Logik !!)
1 = unprogrammiert (!! negative Logik !!)

	Default (keine Programmierung)
	Wichtige Programmierung (abweichend von Default)
	Sonstige Systemeinstellung abweichend von Default)

CKSEL3-0 ist im Originalzustand laut Datenblatt auf 0001. Das entspricht dem internen Oszillator mit 1MHz. Uns interessiert aber der Zustand mit externem Quarz mit 16 MHz. Dazu muss CKSEL3-1 auf 111 gesetzt werden. Die Start-Up-Time wird durch CKSEL0 und SUT bestimmt. Eine 1 auf CKSEL0 bedeutet zusammen mit 11 auf SUT1-0 die längstmögliche Startverzögerung. So hat der Quarz die meiste Zeit zum Anschwingen. Wie beim Atmega üblich, bedeutet eine Null ein Häkchen und bei einer Eins bleibt das Feld leer.

Beispiele von Fuse-Bits des ATmega128 mit YAAP:

Auslieferzustand:



Beispiel Programmiert:

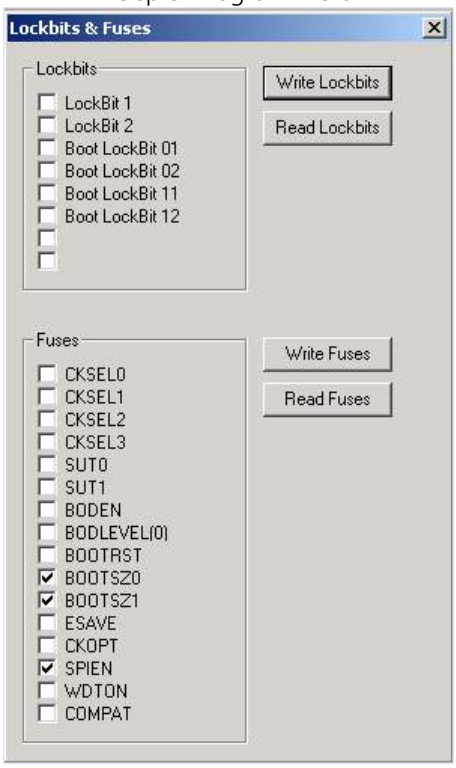


Abbildung 33: Fusebits bei Auslieferung und nach Programmierung

8.1.17 Die Fuse-Konfiguration von ConvCtrl

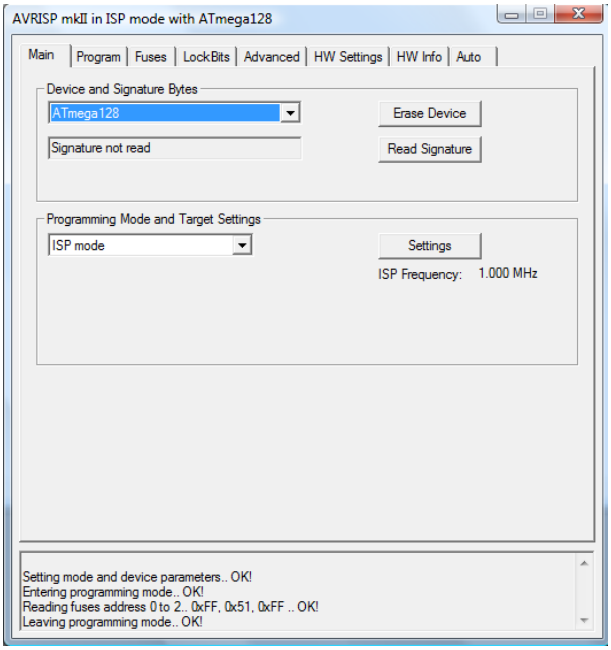


Tabelle 51: AVR-Studio - Main

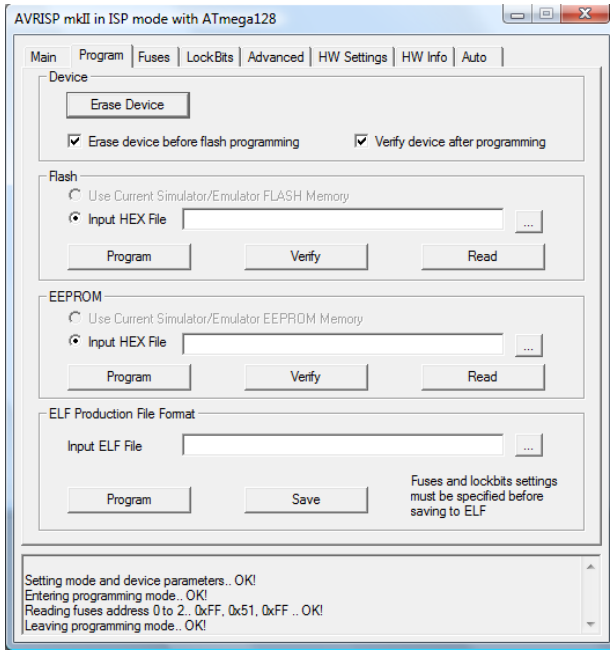


Tabelle 52: AVR-Studio - Program

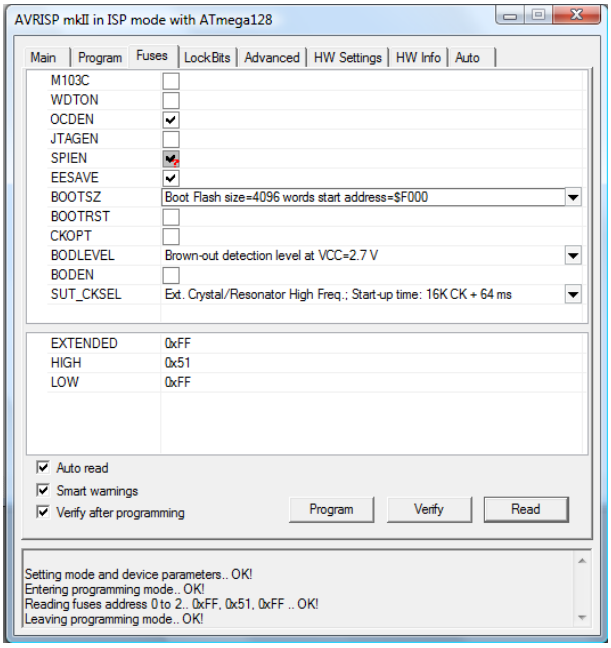


Tabelle 53: AVR-Studio - Fuses

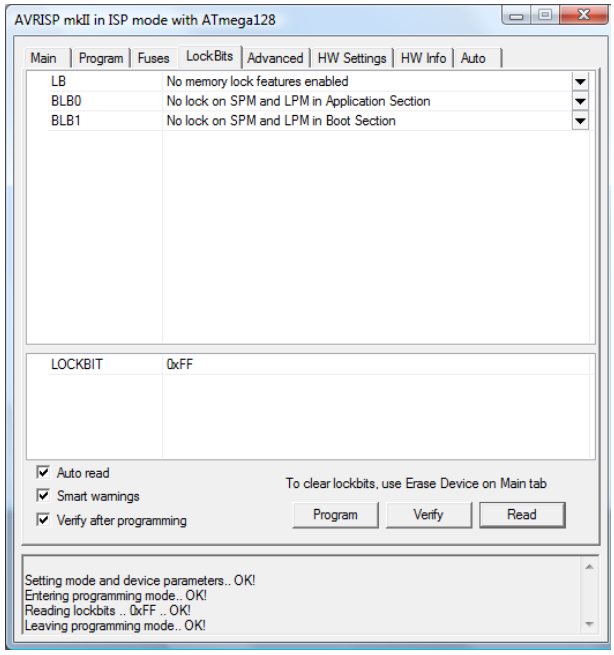


Tabelle 54: AVR-Studio - LockBits

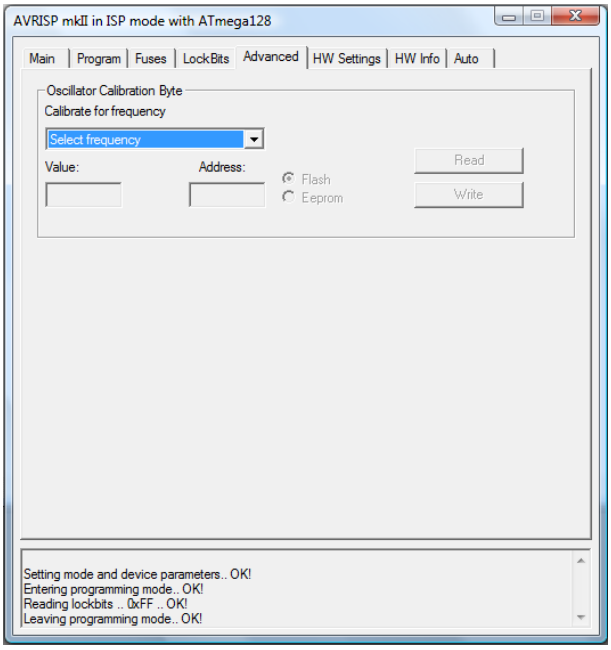


Tabelle 55: AVR-Studio - Advanced

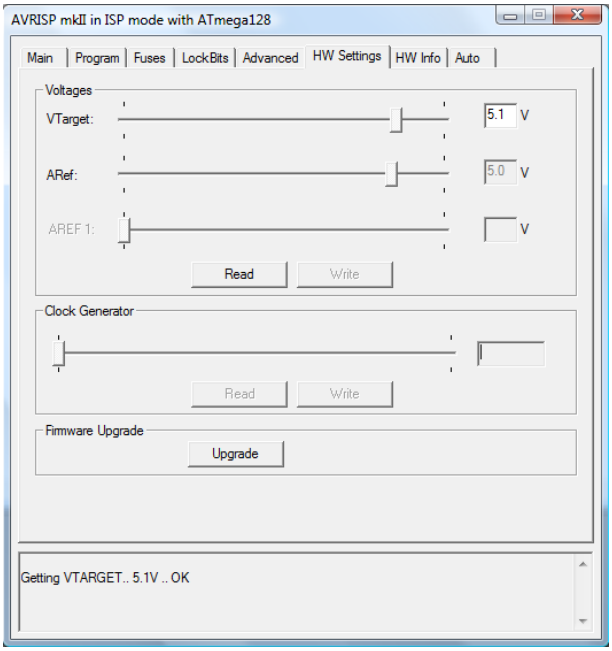


Tabelle 56: AVR-Studio – MainHW Settings

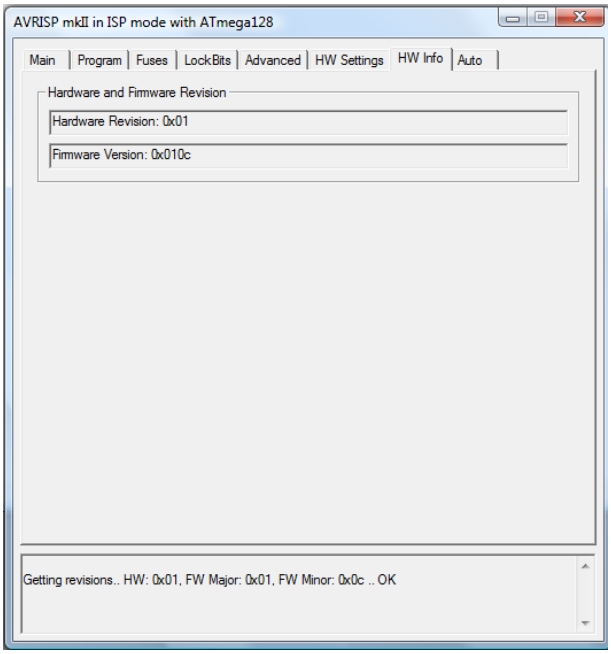


Tabelle 57: AVR-Studio – HW Info

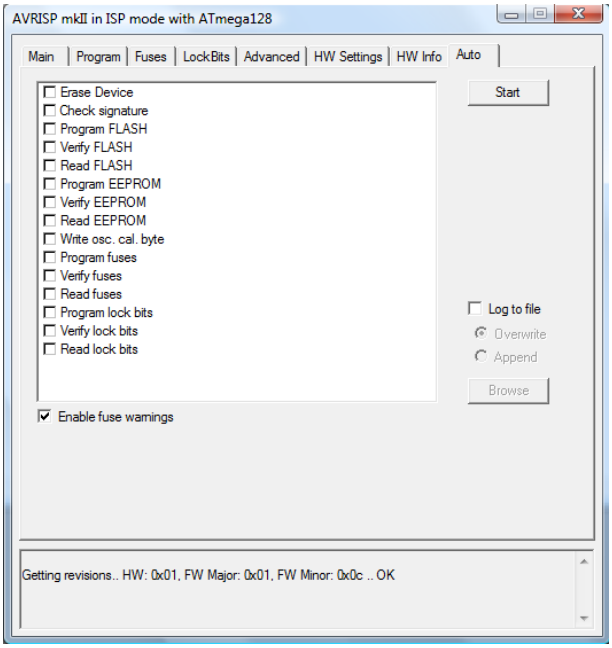


Tabelle 58: AVR-Studio - Auto

8.2 Spannungsversorgung

8.2.1 Hauptschalter

Die gesamte Anlage kann durch einen Hauptschalter ein- und ausgeschaltet werden. Folgender Hauptschalter wurde ausgewählt:



Ausführung:	1polig Ein/Aus
Schaltleistung:	10 A/125 V/AC/6A/250 V/AC
Anschluss:	Lötösen
Frontplatten-Stärke:	Max. 5 mm
Einbau-Ø:	18 mm
Isolations-Widerstand:	Min. DC 500 V/DC/100 MΩ
Kontakt-Widerstand:	Max. 50 mΩ
Spannungs-Festigkeit:	1500 V/AC/60 s
Hebel:	Schwarz mit LED

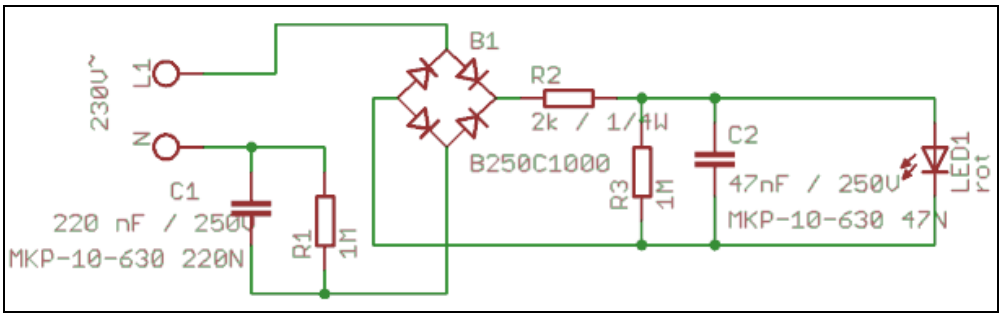
Weitere technische Daten:
Rahmen aus Polyamid. Lötanschlüsse Messing versilbert mit abgeflachten Seiten zur Frontplattenarretierung.

Conrad Bestellnummer: 701047 – 62

Abbildung 34: Hauptschalter

8.2.2 LED Vorschaltmodul für Hauptschalter

Für den direkten Betrieb der Schalter LED an 230V~ muss Elektronik in Form einer Vorschaltung umgesetzt werden. Das Konzept sieht die Umsetzung entsprechend dem folgenden Schaltplan vor:



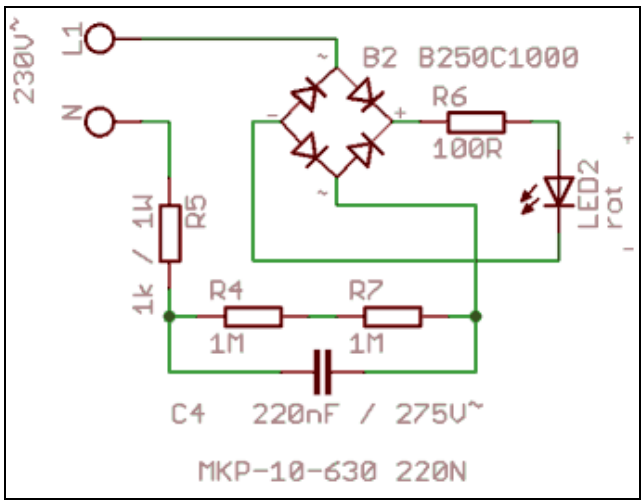
Schaltbild 15: Vorschaltung für LED an 230V~

Bemerkung:
Für die beiden Kondensatoren sind die Reichelt-Bestellnummern bereits im Schaltbild mit angegeben!

Bauteile:

Stückliste: LED Vorschaltung			
Widerstände		*Halbleiter*	
R1, R3	1M Ω / ¼ Watt	B1	Brückgleichrichter B250C1000
R2	2k Ω / ¼ Watt	LED1	LED rot (in Hauptschalter eingebaut)
Kondensatoren			
C1	Folienkondensator 220nF / 250V		
C2	Folienkondensator 47nF / 250V		

Tabelle 59: Stückliste Vorschaltplatine



Schaltbild 16: Schaltbild der Conrad Vorschaltung für LED's

Bauteile:

Stückliste: LED Vorschaltung Conrad			
Widerstände		*Halbleiter*	
R4, R7	1M Ω / ¼ Watt	B1	Brückgleichrichter B250C1000
R6	100 Ω	LED1	LED rot (in Hauptschalter eingebaut)
R2	1k Ω / 1 Watt		
Kondensatoren			
C1	Folienkondensator 220nF / 275V Bestellnummer Reichelt: MKP-10-630 220N		

Tabelle 60: Stückliste Vorschaltplatine

Die Conrad Bestellnummer lautet Best.-Nr. 19 11 78.

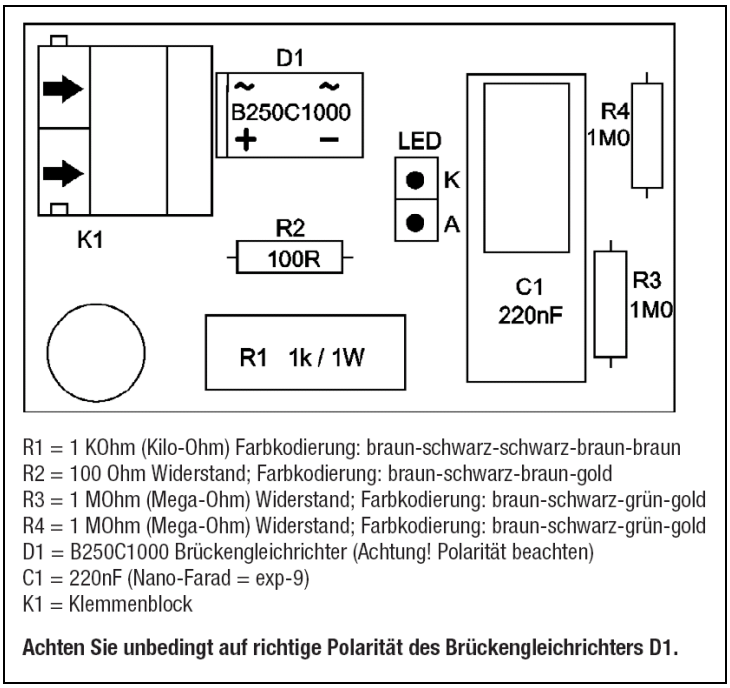


Abbildung 35: Bestückung der Conrad Vorschaltplatine

Ergänzungen zur Theorie der Vorschaltung:

LEDs müssen im Gegensatz zu normalen Glühlampen im Strom begrenzt werden. Betreibt man LEDs an niedrigen Gleichspannungen, reicht ein entsprechend dimensionierter Vorwiderstand aus. Bei 230V kommen gleich 2 Probleme dazu:

1. Die Verlustleistung an den Vorwiderständen werden relativ groß. Selbst wenn dies thermisch kein Problem ist, sinkt der Gesamtwirkungsgrad einer einzelnen LED an 230 V auf etwa 0,06%.
2. Wird eine LED verpolt angeschlossen so wird sie ab einer bestimmten Spannung zerstört (teilweise reichen 5V)

Ein drittes Problem könnte das Flimmern durch die Spannungsänderungen sein, wenn nur eine halbe Halbwelle verwendet wird.

Die Schaltung muss also 2 Dinge tun:

1. Den Strom begrenzen.
2. Einen Gleichstrom erzeugen, der mit einem Kondensator etwas geglättet wird.

Die Strombegrenzung kann man bei Wechselstrom am einfachsten mit einem Kondensator (C1) bewerkstelligen. Je größer die Kapazität ist, desto höher ist der Strom (weiterhin steigt der Strom proportional zur Frequenz und Spannung, was für uns aber weniger interessant ist, da beides durch das Stromnetz vorgegeben ist). Wichtig ist, das der Kondensator die richtige Kapazität hat und auf 400V dimensioniert wird. Ansonsten wird der Rest der Schaltung zerstört !!!

Wie groß muss der strombegrenzende Kondensator sein?

Mit 220nF, 250V~ ist man in der Regel gut bedient, der Blindwiderstand beträgt dann etwa 16KOhm. Wenn viele LEDS verwendet werden darf der Kondensator größer werden:

1-20	LEDs 220-330nF 250V~
20-40	LEDs 330-500nF 250V~
40-60	LEDS 500-600nF 250V~

Dies sind nur grobe Richtwerte, die aber in den meisten Fällen ausreichen sollten. Wenn die Helligkeit der LEDs zu gering ist kann C1 etwas vergrößert werden. Am besten ist ein Vergleich der Helligkeit mit einer anderen LED, die mit Vorwiderstand an einer Gleichtromquelle betrieben wird.

Es ist auch möglich Spannungsabfall an R2 zu Messen um so zu überprüfen, ob der Strom den richtigen Wert hat.

> Achtung!!
>
> Wer nicht im Umgang mit 230V~ geschult ist, sollte diese Schaltung auf keinen Fall ohne Berührungsschutz / Isolierung betreiben. Besonders beim Messen ist diese Bedingung nicht gegeben. Ich empfehle beim Messen Messkabel mit Krokoklemmen o.Ä. um eine versehendliche Berührung auszuschließen.

Der strombegrenzten Wechselstrom muss mit einem Brückengleichrichter gleichgerichtet werden. Es sind entweder Brückengleichrichter oder aus einzelnen Dioden aufgebaute Gleichrichter möglich.

Um zu verhindern, dass der Rest der Schaltung beim Einschalten durch einen Spannungsstoß beschädigt werden kann, wird der Widerstand R2 mit einem Wert von 2K eingebaut. Die Belastbarkeit sollte 1/4W sein.

Der Kondensator C2 hat zwei Funktionen: Er glättet die gleichgerichtete Wechselspannung und schluckt eventuelle Reste von Spannungsspitzen. C2 muss nicht besonders groß sein. 0,047µF sollte vollkommen ausreichen. Die Spannungsfestigkeit hängt von der Anzahl der LEDs ab. (Es schadet nicht einfach den gleichen Typen wie C1 zu nehmen). Zusätzlich sollte noch 2 Widerstande (R1 und R3) parallel zu C1 und C2 verbaut werden um sicher zustellen, dass diese sich bei abgeschalteter Leuchte vollständig entladen können. Die Widerstände sollte 1 MOhm bei 1/4W haben.

Nun fehlt noch das Wichtigste: Unsere LEDs. Es ist egal ob an dieser Schaltung 1 oder 50 LEDs in Reihe hängen, da diese Schaltung mit einem konstantem Strom arbeitet. Beliebig viele LEDs sind aber trotzdem nicht möglich. Wenn die komplette Arbeitspannung aller LEDs in die nähe der maximalen Amplitude von 380V kommt, kann kein Strom mehr fließen. Beachtet werden muss nur das die Kapazität von C1 ein wenig von der Summe aller

LED-Spannungen abhängt. Mehr als 60 LEDs würde ich nicht empfehlen. Bei mehr LEDs macht es schon Sinn die Schaltung zweifach zu bauen. Wenn der Wert von C1 verdoppelt wird, kann der Gleichrichter auch für 2 LED Stränge verwendet werden, allerdings müssen dann auch 2 gleiche Stränge angeschlossen werden, da sonst der doppelte Strom durch den Einzelnen fließt.

Wenn jetzt noch die Frage auftaucht, wo die Energie landet, die als Blindleistung "verbraten" wird. Sie wird wieder zurück in das Stromnetz gespeist. Die Stromversorger mögen das eigentlich gar nicht gerne, aber bei unserer Anwendung fließen eh nur 20mA . Dazu kommt aber das sich kapazitative und induktive Blindströme ausgleichen. Wer aber ein Stadion so mit LEDs ausleuchten will, wird mächtig Ärger mit seinem Stromlieferanten bekommen, denn diese Schaltung speist nicht nur Blindströme in das Netz, sondern produziert durch das nichtlinieare Verhalten der LEDs auch ein paar Oberwellen (Keine Angst, jedes Computernetzteil ohne PFC ist 100x schlimmer).

Eine weitere Frage wäre ob sich die LEDs mit einem Dimmer regeln lassen (Phasenanschnitt, verzert die Sinuswelle des Stromnetzten noch schlimmer: Es geht, aber es geht nicht gut. Die LEDs fangen erst an zu leuchten, wenn eine von der Anzahl der LEDs abhängige Spannung erreicht wird. Sobald das Maximum der Amplitude erreicht wird haben die LEDs schon ihre volle Leuchtkraft.

8.2.3 Abschätzung Stromverbrauch

Stromverbrauch für 5V-Verbraucher:

Lfd.-Nr.	*Verbraucher*	*Strom in Ampere (A)*
1	ATmega	0,0011
2	LCD Display	0,00025
3	Beleuchtung LCD Display	0,04
4		
5		
6		
7		
8		
9		
10		
--	*Gesamtverbrauch*	0,04135

Tabelle 61: Gesamtstromverbrauch für 5V Komponenten

Aus dem Stromverbrauch ergibt sich für den 5V Festspannungsregler welcher an 15AC betrieben wird (10V „Abfall“ an Regler) eine Verlustleistung von 0,4135 Watt welche in Wärme umgewandelt werden müssen.

Stromverbrauch für 12V-Verbraucher:

Lfd.-Nr.	*Verbraucher*	*Strom in Ampere (A)*
1	Regensensor (mit aktiver Heizung)	0,125
2	Lüfter (max.) (Miniaturlüfter 25mm nur 30 mA)	0,100
3	Relais für Beschattung auf / zu	0,054
4	Relais für Beleuchtung	0,054
5		
6		
7		
8		
9		
10		
--	*Gesamtverbrauch*	0,333

Tabelle 62: Gesamtstromverbrauch für 12V Komponenten

Aus dem Stromverbrauch ergibt sich für den 12V Festspannungsregler welcher an 15AC betrieben wird (3V „Abfall“ an Regler) eine Verlustleistung von 0,675 Watt welche in Wärme umgewandelt werden müssen.

Kühlkörpervarianten:

Für die TO-220 Gehäuse der Spannungsregler stehen bei Reichelt-Elektronik drei mögliche Kühlkörpervarianten zur Verfügung:

		
lötbar durch verzinnten Snap-In-Stift. Bestellnummer *V FI343* Aufsteckkühlkörper für Gehäuse TO-220, 25K/W	lötbar durch verzinnten Stift auf der Rückseite. Bestellnummer *V FI342* Aufsteckkühlkörper für Gehäuse TO-220, 25K/W	lötbar durch verzinnten Drahtbügel. Bestellnummer *V CK960/20* Aufsteckkühlkörper für Gehäuse TO-220 ,13K/W

Tabelle 63: Kühlkörper für Festspannungsregler

8.2.4 Spannungen 5V und 12V

Für die Spannungsversorgung des Projekts ConvCtrl werden die zwei Spannungen +5V und +12V mit gemeinsamen Massebezug GND bereitgestellt.

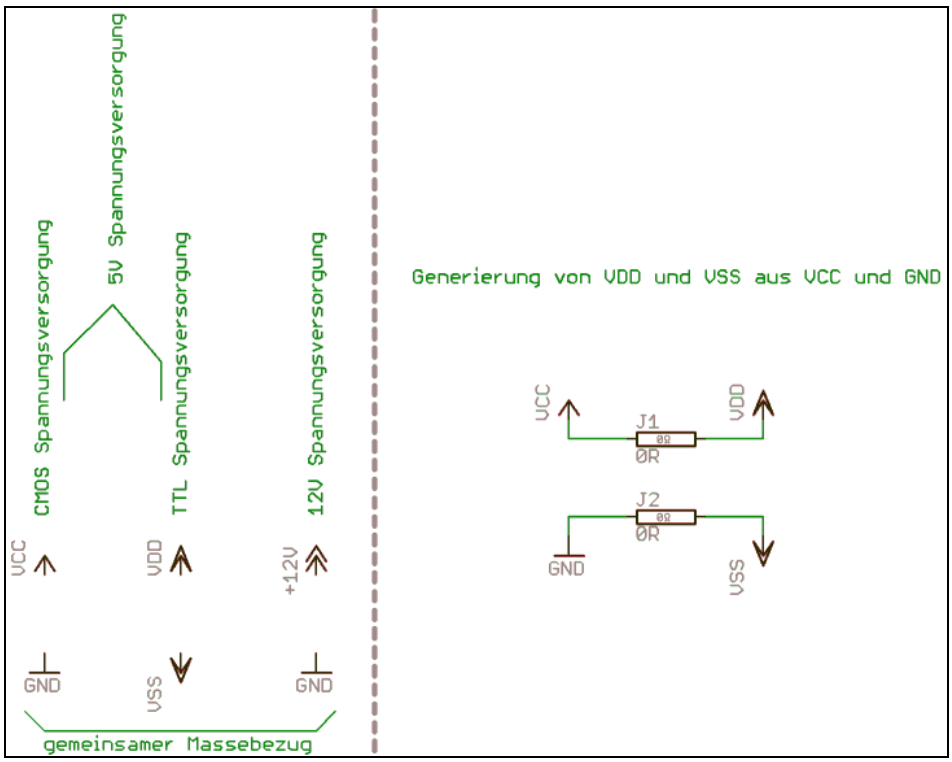
Unterschiedliche Technologien haben unterschiedliche Bezeichnungen für die notwendigen Spannungsversorgungen.

Für CMOS Bausteine gelten die Bezeichnungen VCC und GND.
TTL-Logik verwendet die Bezeichnungen VDD und VSS. VDD und VSS werden im Projekt durch 0Ω-Widerstände (Drahtbrücken) erzeugt um EAGLE die korrekte Behandlung der Spannungen zu ermöglichen.

Für das Projekt gelten die Folgenden Bezeichnungen in den Netzen:

Spannungspotential	*Verwendeter Name*
GND (Ground) = MASSE = 0V	GND VSS
+5V Spannung	VCC VDD
+12V	+12V

Tabelle 64: Vorgeschriebene Namensgebung für Spannungsversorgungen



Schaltbild 17: Symbolfestlegung für Spannungsversorgungen

8.3 Netzteil mit Festspannungsregler

Für die Spannungsversorgung wird ein Transformator der Firma HAHN – BV UI 394 0088 – mit nachgeschalteten, 2-stufigen Festspannungsreglern für die Spannungen +5V und +12V eingesetzt.

Transformator:



BV UI 394 0088
Temperaturklasse ta 70°C/B, nicht kurzschlussfest

Leistung:	18VA
Bauhöhe:	28,5 mm
Gewicht:	390 Gramm
Spannungen primär:	2 x 115V
Anschlussstifte primär:	1-3 / 8-10
Spannungen sekundär:	2 x 15V
Strom sekundär:	2x 600 mA
Anschlussstifte sekundär:	12-14 / 17-19
Leerlaufspannungen:	2 x 20,4 V

Abbildung 36: HAHN Trafo

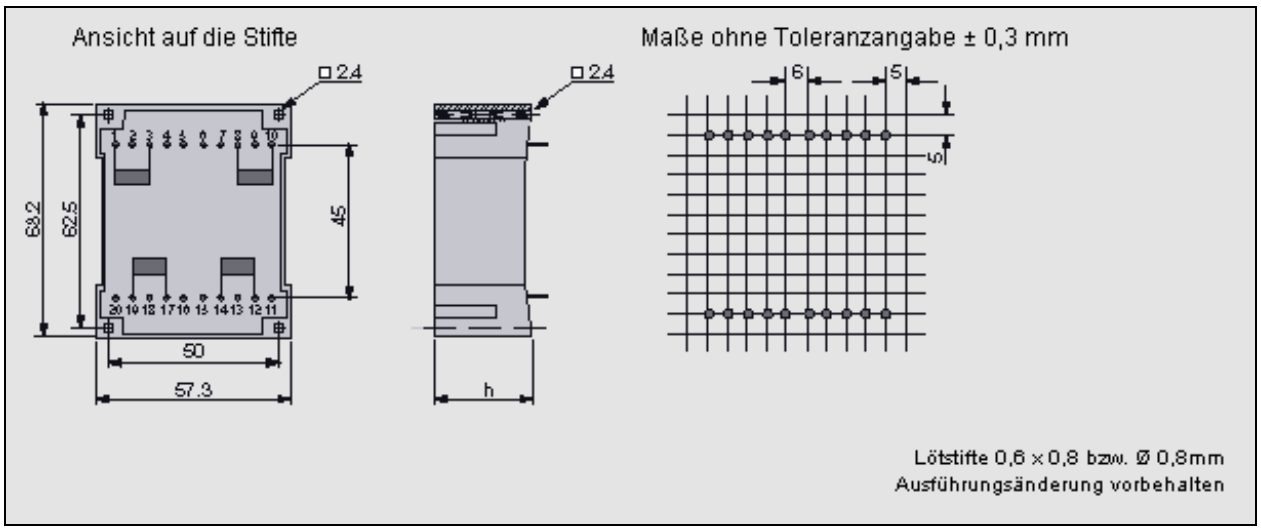


Abbildung 37: Transformator Hahn BV UI 394 0088

Beschaltung: Schaltung für 220 V, 230 V bzw. 240 V: Brücke: 3 – 8
Sek. doppelter Strom: Brücke: 12 - 17 und 14 - 19

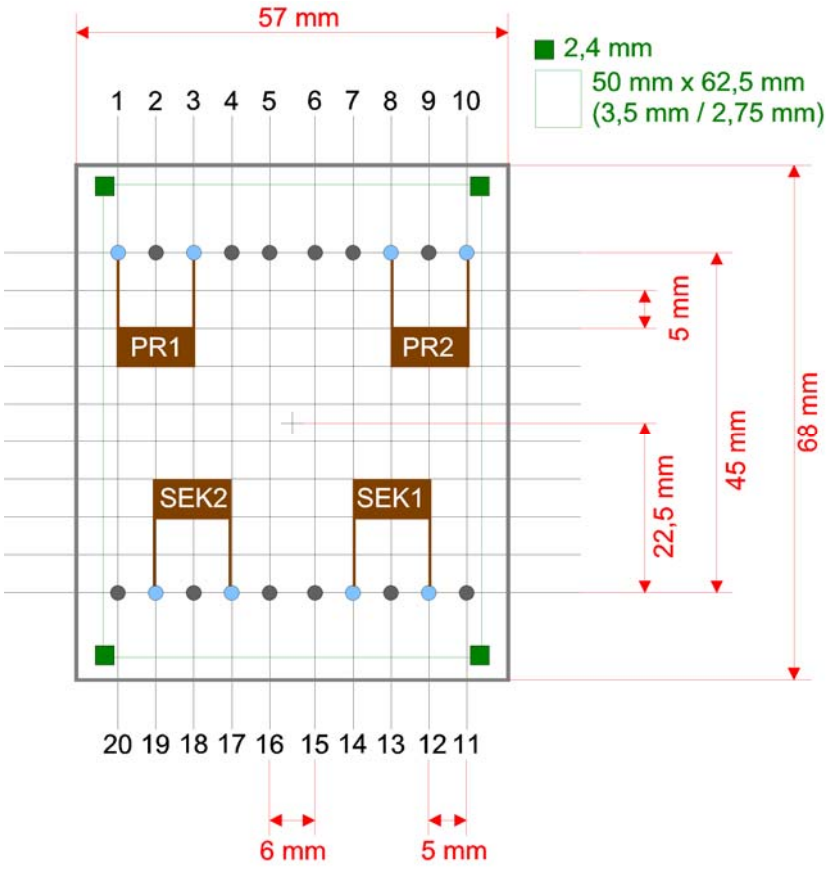
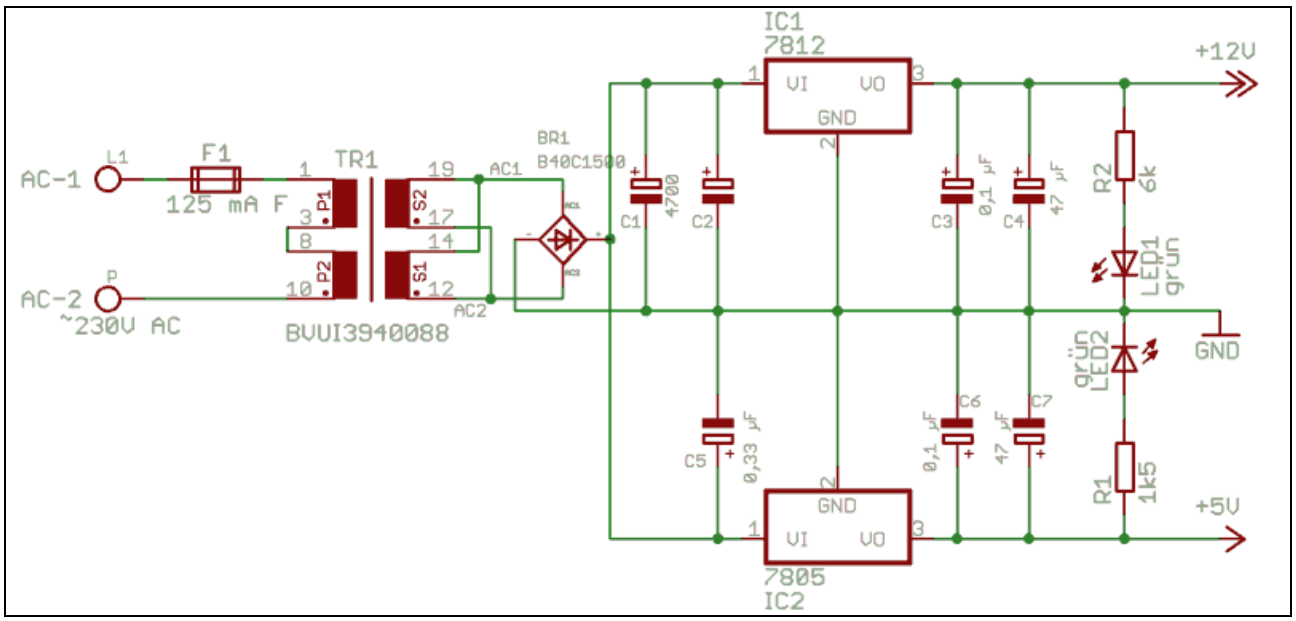


Abbildung 38: PINOut Hahn Trafo BV UI 394 0088 (Ansicht von unten)

8.3.1 Schaltplan Netzteil

Der folgende Schaltplan-Auszug zeigt die Beschaltung des Netzteils welches zur Erzeugung der Spannungen +5V und +12V im Projekt eingesetzt wird:



Schaltbild 18: Spannungsversorgung +5V und +12V mit Festspannungsregler

Bauteile:

Stückliste: Spannungsversorgung			
Sonstiges		***Halbleiter***	
F1	Feinsicherung 125 mA flink	IC1	Festspannungsregler 7812
AC-1/AC-2	Schraubklemmenblock Raster 7,5 mm	IC2	Festspannungsregler 7805
TR1	Transformator HAHN BV UI 394 0088	BR1	Brückengleichrichter B40C1500
Kondensatoren		LED1	LED 3mm, low current, grün
C1	Elko 4700 µF / 35V	LED2	LED 3mm, low current, grün
C2, C5	Elko 0,33 µF / 35V	***Widerstände***	
C3, C6	Elko 0,1 µF / 35V	R1	Metallschichtwiederstand 1k5 Ω
C4, C7	Elko 47 µF / 35V	R2	Metallschichtwiderstand 6k04 Ω

Tabelle 65: Stückliste Nulldurchgangskennung

Bemerkungen:
Der Brückengleichrichter B40C1500 ist für eine Spannung von 40V und einem Strom bis 2A ausgelegt.

Die verwendeten Festspannungsregler für 5V und 12V sind für einen Strom bis 2A ausgelegt. Zum Zeitpunkt der Dokumenterstellung liegt noch keine Angabe über Art und Ausprägung der Kühlung vor!

Die beiden verwendeten Leuchtdioden LED1 und LED2 zeigen die Verfügbarkeit der Betriebsspannungen +5V und +12V an und sind als Low Current LED ausgeführt.

Nachtrag 17.04.2008: Ausführung Kühlung ist nicht mehr relevant da Netzteil mit Festspannungsreglern nicht mehr zum Einsatz kommt sondern auf Schaltnetzteil umgestellt wurde.

8.3.2 Zusammenfassung nach Inbetriebnahme

Die Leerlaufspannung des Netgeräts beträgt 20,4 Volt.

Bei einer Belastung von 50 mA auf dem 5V-Zweig und 64 mA auf dem 12V-Zweig sing die Leerlaufspannung auf 19,4 Volt. Durch die Beschaltung mit einem 4.700 µF Glättungskondensator ergibt sich nach der Formel

$$U_C = U_{eff} * \sqrt{2} = 19{,}4V * 1{,}41 = 27{,}35V$$

eine Spannung am Kondensator von 27,35V.

Um daraus 5V bzw. 12V erzeugen zu können müssen 22,35V bzw. 15,35V „vernichtet werden. Bei Strömen von 50mA bzw. 64 mA ergeben sich die folgenden Verlustleistungen:

Für 5V: $$P = U * I = 22{,}35V * 50mA = 1{,}1175W$$

Für 12V: $P = U * I = 15{,}35V * 64mA = 0{,}9824W$

Diese Verlustleistungen führen bereits bei so geringen Strömen zu einer signifikaten Erwärmung der Festspannungsregler.

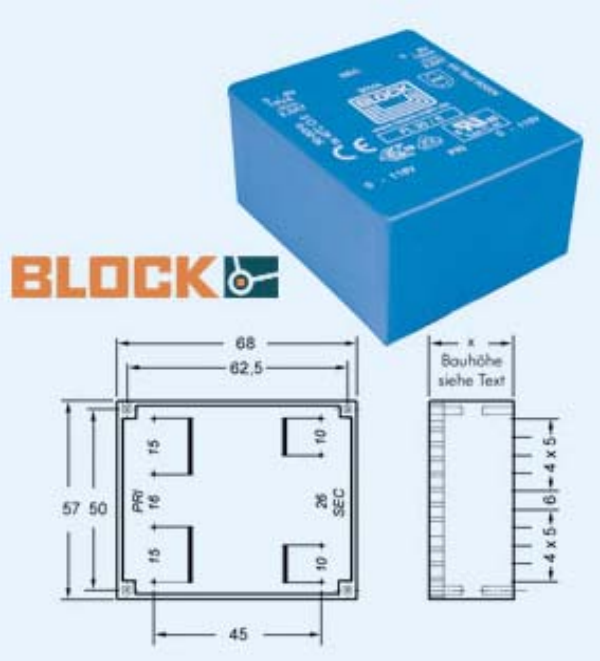
Bei höheren Strömen wird sicherlich die zu „vernichtende“ Spannung sinken, jedoch erhöht sich der damit verbundene Strom.

Aufgrund der gemachten Erfahrungen scheint der Einsatz eines festspannungsreglers über obige Schaltung für einen Dauerbetrieb als ungeeignet. Als Lösung empfiehlt sich ein Schaltregler welcher im folgenden Kapitel eingehend beschrieben wird.

8.4 Netzteil mit Schaltregler

Für die Spannungsversorgung wird ein Transformator der Firma BLOCK – FL30/15 mit 2 nachgeschalteten Schaltspannungsreglern für die Spannungen +5V und +12V eingesetzt.

Transformator:



FL30/15	
Temperaturklasse:	ta 40°/B
Leistung:	30VA
Länge:	68mm
Breite:	57mm
Bauhöhe:	35mm
Gewicht:	0,547 kg
Spannungen primär:	2 x 115V
Anschlussstifte primär:	1-4 / 9-6
Spannungen sekundär:	2 x 15,0V
Strom sekundär:	2x 1000 mA
Anschlussstifte sekundär:	12-14 / 19-17
Leerlaufspannungen:	2 x 17,2 V
Bestellnummer Reichelt:	UI 39/21 215

Abbildung 39: BLOCK Trafo

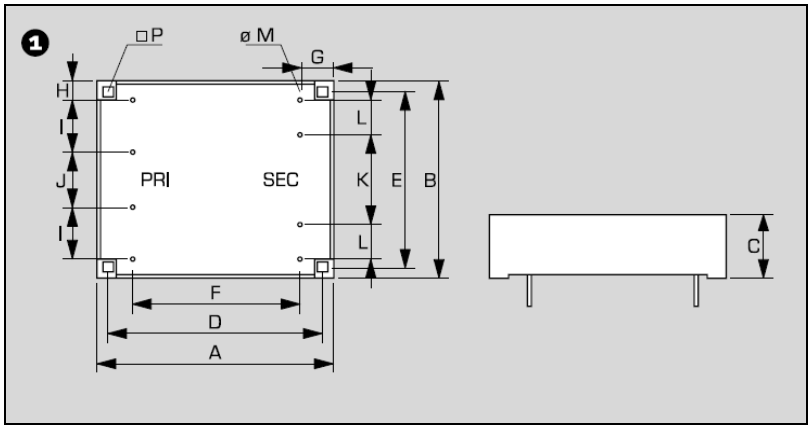


Abbildung 40: Transformator Block FL30/15

Beschaltung: Schaltung für 220 V, 230 V bzw. 240 V: Brücke: 3 – 8

Typ / Type	FL 30/..
Eingangsspannung / Input voltage	**2x115 V**
Frequenzbereich / Frequency range	50 - 60 Hz
Leerlaufverluste (typ.) / No-load loss (typ.)	1.7 W
Ausgangsspannung/Bestellnr. / Output voltage/Order no.	**2x5 V: FL 30/5 2x6 V: FL 30/6 2x8 V: FL 30/8 2x9 V: FL 30/9 2x12 V: FL 30/12 2x15 V: FL 30/15 2x18 V: FL 30/18 2x24 V: FL 30/24****
Leerlaufspannung (ca. x Faktor) / No-load voltage (ca. x factor)	1.17

Leistung / Power	**30.0 VA**
Prüfzeichen / Approvals	C-UL-US, ENEC 10 (VDE)
Kurzschlussfestigkeit / Short circuit strength	nicht kurzschlussfest / non-short-circuit-proof
Schutzart / Protection index	IP 00
Schutzklasse (vorbereitet) / Safety class (prepared)	II
Umgebungstemperatur max. / Ambient temperature max.	40 °C
Isolierstoffklasse / Class of Insulation System	VDE=E, UL=105
Prüfspannung / Test voltage	5000 V, 50 Hz
Wirkungsgrad / Efficiency	81 %

Bauart / Type	vergossen / encapsulated
Kerntyp / Core type	UI 39/21
Anschlüsse / Terminals	Lötstifte für Leiterplatten / Pins for printed circuit boards
Maße in mm A/B/C/D / Dimensions in mm A/B/C/D	68/57/35.8/62.5
Maße in mm E/F/G/H / Dimensions in mm E/F/G/H	50/45/11.5/5.5
Maße in mm I/J/K/L / Dimensions in mm I/J/K/L	15/16/26/10
Maße in mm M/N/O/P / Dimensions in mm M/N/O/P	0.8/-/-/2.5
Maßbild / Dimension drawing	1
Gewicht / Weight	0.53 kg
Bestellnummer / Order Number	**siehe Ausgangsspannung / see output voltage**

Tabelle 66: Leistungsdaten Bloeck Trafo Fl 30/15

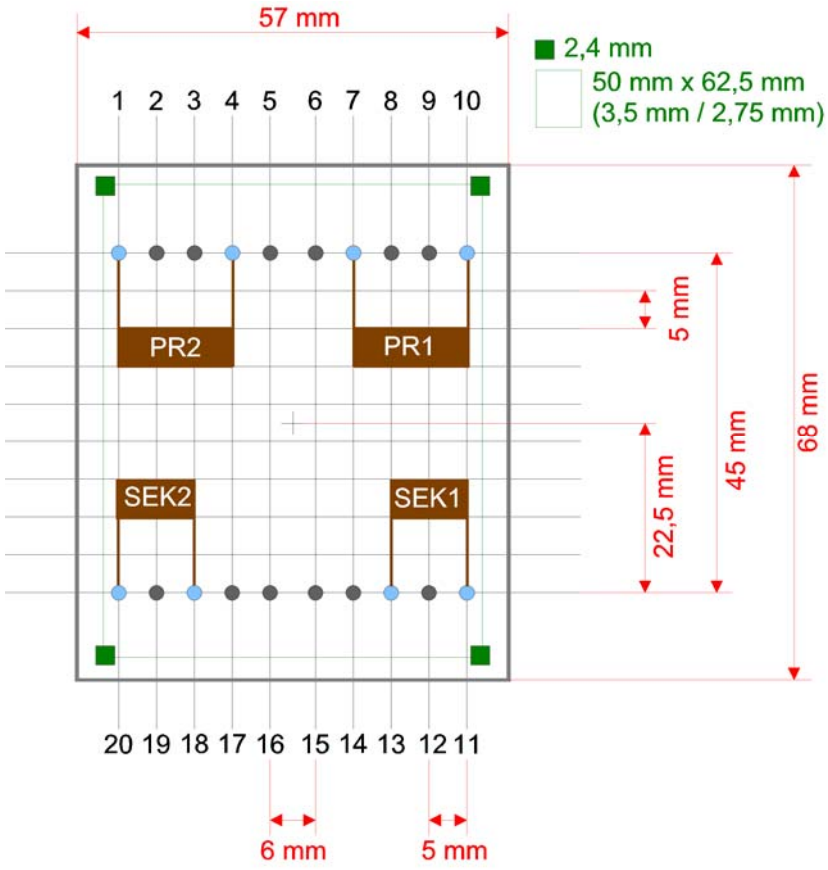


Abbildung 41: PINOut Block-Trafo FL30/15 (Ansicht von unten)

8.4.1 Schaltplan Netzteil

Der Schaltregler:

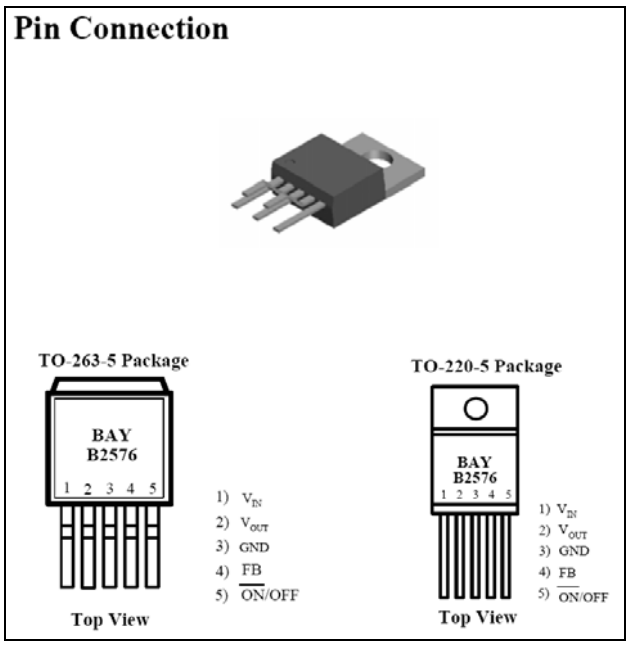
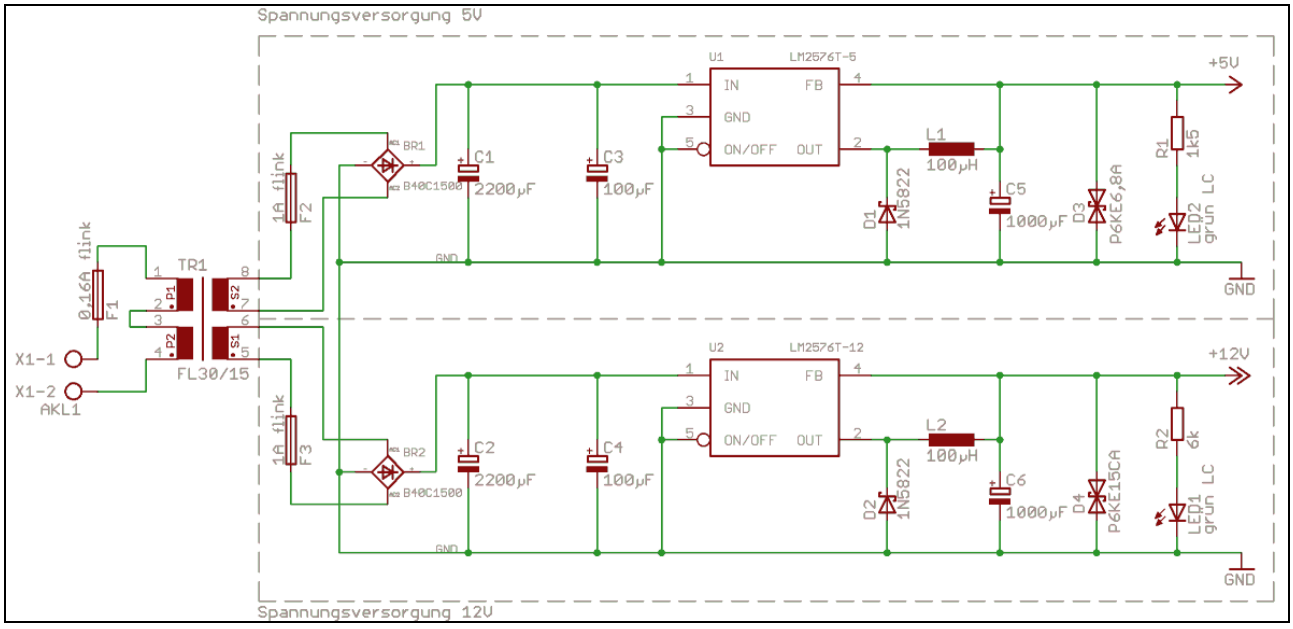


Abbildung 42: Schaltregler LM2576

Der Schaltplan:

Der folgende Schaltplan zeigt die Beschaltung des Netzteils welches zur Erzeugung der Spannungen +5V und +12V im Projekt eingesetzt wird:



Schaltbild 19: Schaltnetzteil +5V und +12V

Als Bestückunsalternative wird die Schottky-Diode 1N5821 und eine 220 µH Spule vorgesehen. Welcher der Bauteile eingesetzt wird steht zum Zeitpunkt der Konzepterstellung noch nicht fest. Die genaue Festlegung wird im hinteren Teil des Dokument im Kapitel „Hardware“ dokumentiert.

Der Kondensator C3 und C4, die Dioden D1 und D2 sowei die Spulen sind zur Vermeidung von Spannungsriffel so nah wie möglich am Schaltregler zu platzieren.

Bauteile:

Stückliste: Spannungsversorgung

Sonstiges		*Halbleiter*	
F1	Feinsicherung 160 mA flink		
F2, F3	Feinsicherung 1A flink	BR1, BR2	Brückengleichrichter B40C1500
AKL1	Schraubklemmenblock Raster 7,5 mm	U1	Schaltregler LM2576T-5 (5V)
TR1	Transformator Block FL30/15	U2	Schaltregler LM2576T-12 (12V)
		LED1	LED 3mm, low current, grün
Kondensatoren		LED2	LED 3mm, low current, grün
C1,C2	Elko 2200 µF / 35V	D1, D2	Diode 1N5822
C3, C4	Elko 100 µF / 35V	D3	Überspannungsschutzdiode 6,8V
C5, C6	Elko 1000 µF / 35V	D4	Überspannungsschutzdiode 15V
Induktivitäten		*Widerstände*	
L1, L2	Speicherdrossel 100µH	R1	Metallschichtwiederstand 1k5 Ω
		R2	Metallschichtwiderstand 6k04 Ω

Tabelle 67: Stückliste Schaltnetzteil +5V und +12V

Bemerkungen:
Der Brückengleichrichter B40C1500 ist für eine Spannung von 40V und einem Strom bis 2A ausgelegt.

Die verwendeten Schaltregler für 5V und 12V sind für einen Strom bis 3A ausgelegt. Zum Zeitpunkt der Dokumenterstellung liegt noch keine Angabe über Art und Ausprägung der Kühlung vor!

Die beiden verwendeten Leuchtdioden LED1 und LED2 zeigen die Verfügbarkeit der Betriebsspannungen +5V und +12V an und sind als Low Current LED ausgeführt.

Themenbezogene LED's:

LED / Funktion / Beschreibung	*Abkürzung*	*Farbe*	
5V present	LED_5V	grün	*[LED1]*
12V present	LED_12V	grün	*[LED2]*

Tabelle 68: Themenbezogene LED's für Schaltnetzteil

8.5 Steuerung Beschattungsanlage Motorantriebe

Die Beschattungsanlage wird durch Elektromotoren angetrieben welche je nach Ansteuerung die Beschattung zwangsgesteuert zuziehen oder aufziehen.

Der Anschluss der Elektromotoren erfolgt über eine 7-adrige Leitung mit 1,5 m^2 Kupferdurchmesser pro Leitung.

Die Leitungen teilen sich auf in:

Nummer	*Farben*	*Funktion*	
1	schwarz (1)	n.c.	= nicht belegt
2	schwarz (2)	N	= Nullleiter
3	schwarz (3)	S1	= zufahren (Beschattung zu)
4	schwarz (4)	S2	= auffahren (Beschattung auf)
5	schwarz (5)	n.c.	= nicht belegt
6	schwarz (6)	n.c.	= nicht belegt
7	grün / gelb	PE	= Schutzleiter

Tabelle 69: Adernbelegung Motoranschlüsse Beschattung

Die beiden Steuerleitungen S1 und S2 für „auf" und „zu" werden über zwei getrennte Relais des folgenden Typs gesteuert:

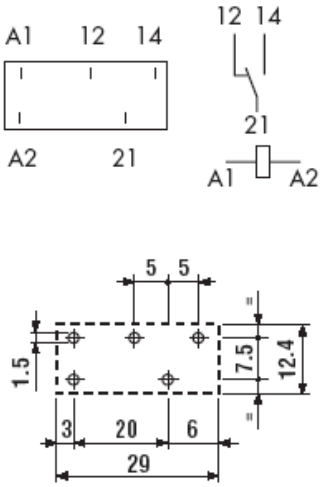


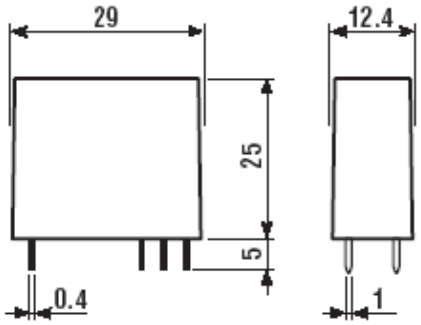
Print-Relais
Firma finder

Typ: 40.51
Raster: 5 mm
1 Wechsler mit 10A

Bestellnummer Reichelt: FIN 40.51.9 12V

Beschreibung:
Steckrelais, 1x UM, 250V/10A, 12V, RM 5,0mm





Ansicht auf die Anschlüsse

Abbildung 43: Steuerungsrelais

Beschreibung zum Relais aus Datenblatt:

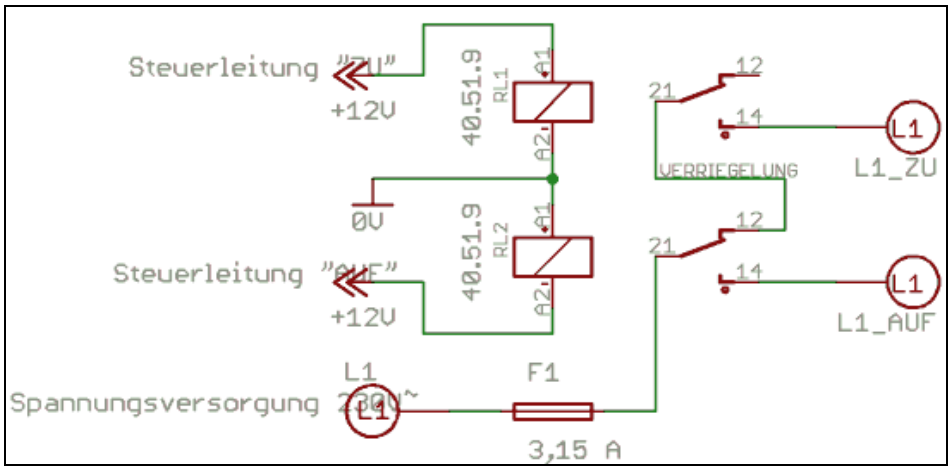
Kontakte				
Anzahl der Kontakte		1 Wechsler	1 Wechsler	2 Wechsler
Max. Dauerstrom/max. Einschaltstrom	A	10/20	10/20	8/15
Nennspannung/max. Schaltspannung	V AC	250/400*	250/400*	250/250
Max. Schaltleistung AC1	VA	2.500	2.500	2.000
Max. Schaltleistung AC15 (230 VAC)	VA	500	500	400
1-Phasenmotorlast, AC3 - Betrieb (230 VAC)	kW	0,37	0,37	0,3
Max. Schaltstrom DC1: 30/110/220V	A	10/0,3/0,12	10/0,3/0,12	8/0,3/0,12
Min. Schaltlast	mW (V/mA)	300 (5/5)	300 (5/5)	300 (5/5)
Kontaktmaterial Standard		AgNi	AgNi	AgNi
Spule				
Lieferbare	V AC (50/60 Hz)	6 - 12 - 24 - 48 - 60 - 110 - 120 - 230 - 240		
Nennspannungen (U_N)	V DC	5 - 6 - 7 - 9 - 12 - 14 - 18 - 21 - 24 - 28 - 36 - 48		60 - 90 - 110
Bemessungsleistung AC/DC/DC sens.	VA (50 Hz)/W/W	1,2/0,65/0,5	1,2/0,65/0,5	1,2/0,65/0,5
Arbeitsbereich	AC (50 Hz)	$(0,8...1,1)U_N$	$(0,8...1,1)U_N$	$(0,8...1,1)U_N$
	DC/DC sensitiv	$(0,73...1,5)U_N/(0,73...1,75)U_N$	$(0,73...1,5)U_N/(0,73...1,75)U_N$	$(0,73...1,5)U_N/(0,73...1,75)U_N$
Haltespannung	AC/DC	$0,8U_N/0,4U_N$	$0,8U_N/0,4U_N$	$0,8U_N/0,4U_N$
Rückfallspannung	AC/DC	$0,2U_N/0,1U_N$	$0,2U_N/0,1U_N$	$0,2U_N/0,1U_N$
Allgemeine Daten				
Mech. Lebensdauer AC/DC	Schaltspiele	$10 \cdot 10^6/20 \cdot 10^6$	$10 \cdot 10^6/20 \cdot 10^6$	$10 \cdot 10^6/20 \cdot 10^6$
Elektrische Lebensdauer AC1	Schaltspiele	$200 \cdot 10^3$	$200 \cdot 10^3$	$100 \cdot 10^3$
Ansprech-/Rückfallzeit (incl. Prellen)	ms	10/10 - (15/12 sensitiv)	10/10 - (15/12 sensitiv)	10/10 - (15/12 sensitiv)
Isolationskoordination EN 61810-5		3,6 kV/3	3,6 kV/3	3,6 kV/2
Spannungfestigkeit Spule/Kontakte (1,2/50 µs)		6 kV (8 mm)	6 kV (8 mm)	6 kV (8 mm)
Spannungfestigkeit offene Kontakte	V AC	1.000	1.000	1.000
Umgebungstemperatur	°C	–40...+85	–40...+85	–40...+85
Schutzart		IP 40	IP 40	IP 40
Zulassungen: (Details auf Anfrage)		GOST RINA		

Tabelle 70: Beschreibung zum Relais

Die Stromaufnahme der Relais bei einer Schaltspannung von 12V DC beträgt bei einer Bemessungsleistung von 0,65W ca. 54 mA.

Für die Steuerung der Relais ist eine beschaltungstechnische Absicherung gegen Gegenläufigkeit vorgesehen welche durch den Zustand der angesteuerten Relais ein gleichzeitiges Auf- und Zufahren verhindert.

8.5.1 Absicherung Gegenläufigkeit der Motor-Relais

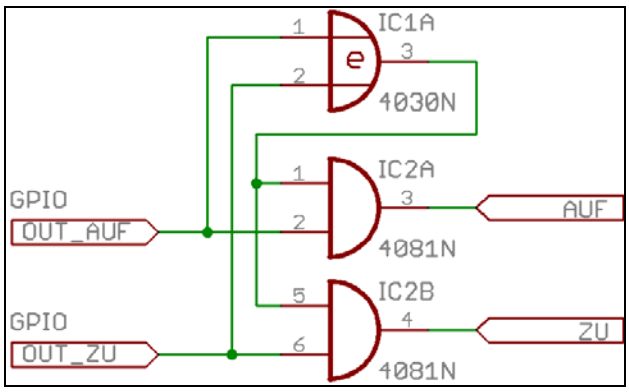


Schaltbild 20: Verriegelung der Relais durch direkt Beschaltung

Es gelten die folgenden Regeln:

- solange Motor „auf“ fährt bekommt Motor „zu“ keine Spannung.
- fährt Motor „zu“ und möchte Motor „auf“ fahren wird sofort Spannung für Motor „zu“ unterbrochen.

Durch die folgende Schaltung können die GPIO-Pins des Controllers durch digitale Verknüpfungen so verriegelt werden, dass immer nur ein Treiber das Relais steuert. Zum Zeitpunkt der Dokumenterstellung ist jedoch noch nicht klar, ob dieser Schaltungsteil eingesetzt wird oder ob auf eine Verriegelung in HW verzichtet wird.



Schaltbild 21: Verriegelung der Relais durch Logikbausteine

Funktionsbeschreibung:

Das XOR-Gatter erzeugt am Ausgang nur dann ein High-Pegel, wenn die beiden Eingänge (OUT_AUF und OUT_ZU) unterschiedliche sind. Somit kann nur AUF oder ZU gefahren werden. Nicht beides gleichzeitig.

Der XOR-Ausgang wird auf ein AND-Gatter zurückgekoppelt und mit dem eigentlichen Schaltsignal verknüpft. Die AND-Gatter-Funktion wird durch das XOR erst dann „freigegeben“ wenn entweder AUF exklusiv-oder ZU gefahren werden soll. Wird an beiden Steuerleitungen ein HIGH ausgegeben erzeugt das XOR ein LOW und HIGH AND LOW am AND-Gatter erzeugt LOW => Eine Ansteuerung der Relais-Stufen erfolgt nicht!

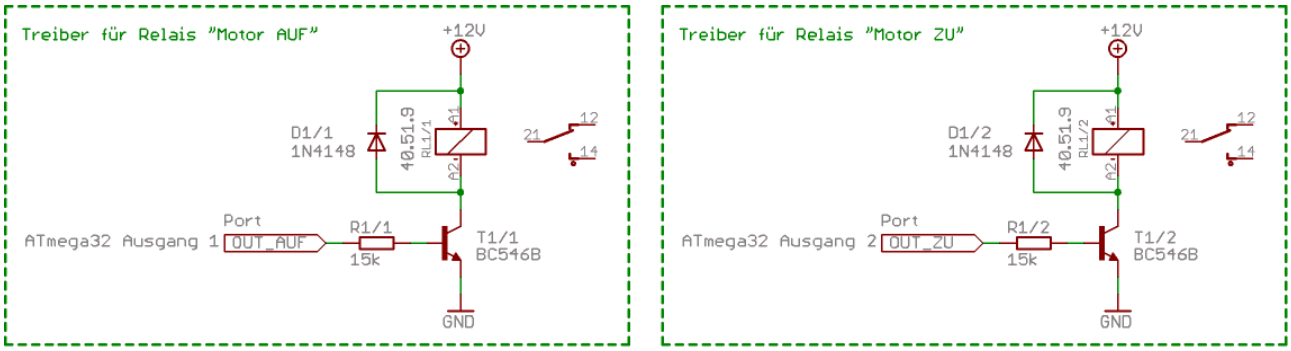
8.5.2 Relaisansteuerung

Die Ausgänge des ATmega sind nicht dafür geeignet ein Relais zu treiben da ihre Leistung dafür nicht reicht. Aus diesem Grund wird im Projekt ConvCtrl eine Transistorstufe zum Schalten des Relais umgesetzt.

Die Transistorstufe wird in ihrer Definition für alle drei Relais

- Motor AUF
- Motor ZU
- Beleuchtung EIN/AUS

gleich umgesetzt.



Schaltbild 22: Treiberstufen für Motor-Steuerrelais Beschattung

Bauteile:

Stückliste: Treiberstufe für Motor-Steuerrelais			
Sonstiges		*Halbleiter*	
RL1/1. RL1/2	finder Relais 40.51	T1/1, T1/2	Transistor BC546B / BC547B
Widerstände		D1/1, D1/2	Diode 1N4148
R1/1, R1/2	15k Ω		

Tabelle 71: Stückliste Treiberstufe für Motor-Steuerrelais

8.5.3 BASCOM Beispielcode für Beschattungsanlage

Der Beispielcode für die Beschattungsanlage ist im Gesamt-Beispielcode für Relais- und Dimmersteuerung enthalten. Siehe hierzu Kapitel „BASCOM Beispielcode für Relaisansteuerung und Dimmer“ im Kapitel „Lichtdimmer“

8.5.4 Prototyp für Beschattungsanlage

Der Prototyp für die Beschattungsanlage wurde im Rahmen des Gesamtprototypen für Relaissteuerung und Dimmung entwickelt. Siehe hierzu Kapitel „Prototyp für Relaisansteuerung und Dimmer“ im Kapitel „Lichtdimmer“

8.6 Steuerung Beleuchtungsanlage

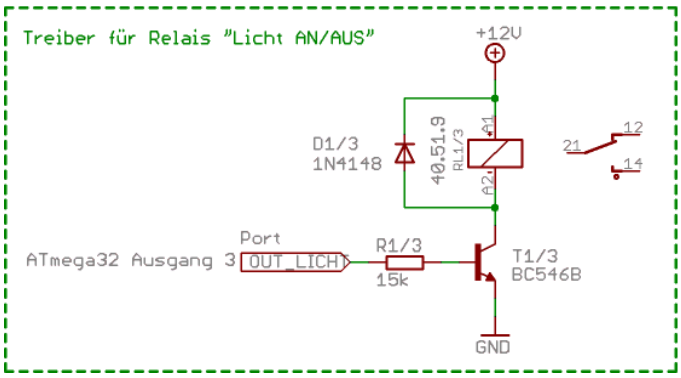
Neben der im nächsten Kapitel beschriebenen Dimmungs-Funktionalität soll die gesamte Beleuchtung ganzheitlich EIN oder AUS geschaltet werden können.

Dies soll über eine Relais-Stufe erfolgen welche der Dimmerstufe als Versorgungselement vorgeschaltet ist.

Die Ausgänge des ATmega sind nicht dafür geeignet ein Relais zu treiben da ihre Leistung dafür nicht reicht. Aus diesem Grund wird im Projekt ConvCtrl eine Transistorstufe zum Schalten des Relais umgesetzt.
Die Transistorstufe wird in ihrer Definition für alle drei Relais

- Motor AUF
- Motor ZU
- Beleuchtung EIN/AUS

gleich umgesetzt.



Schaltbild 23: Treiberstufen für Lichtsteuerung AN/AUS

Bauteile:

Stückliste: Treiberstufe für Lichtsteuerung AN/AUS			
Sonstiges		*Halbleiter*	
RL1/3	finder Relais 40.51	T1/3	Transistor BC546B
Widerstände		D1/3	Diode 1N4148
R1/3	15k Ω		

Tabelle 72: Stückliste Treiberstufe für Lichtsteuerung AN/AUS

8.6.1 BASCOM Beispielcode für Beleuchtungsanlage

Der Beispielcode für die Beleuchtungsanlage ist im Gesamt-Beispielcode für Relais- und Dimmersteuerung enthalten. Siehe hierzu Kapitel „BASCOM Beispielcoder für Relaisansteuerung und Dimmer“ im Kapitel „Lichtdimmer“

8.6.2 Prototyp für Beleuchtungsanlage

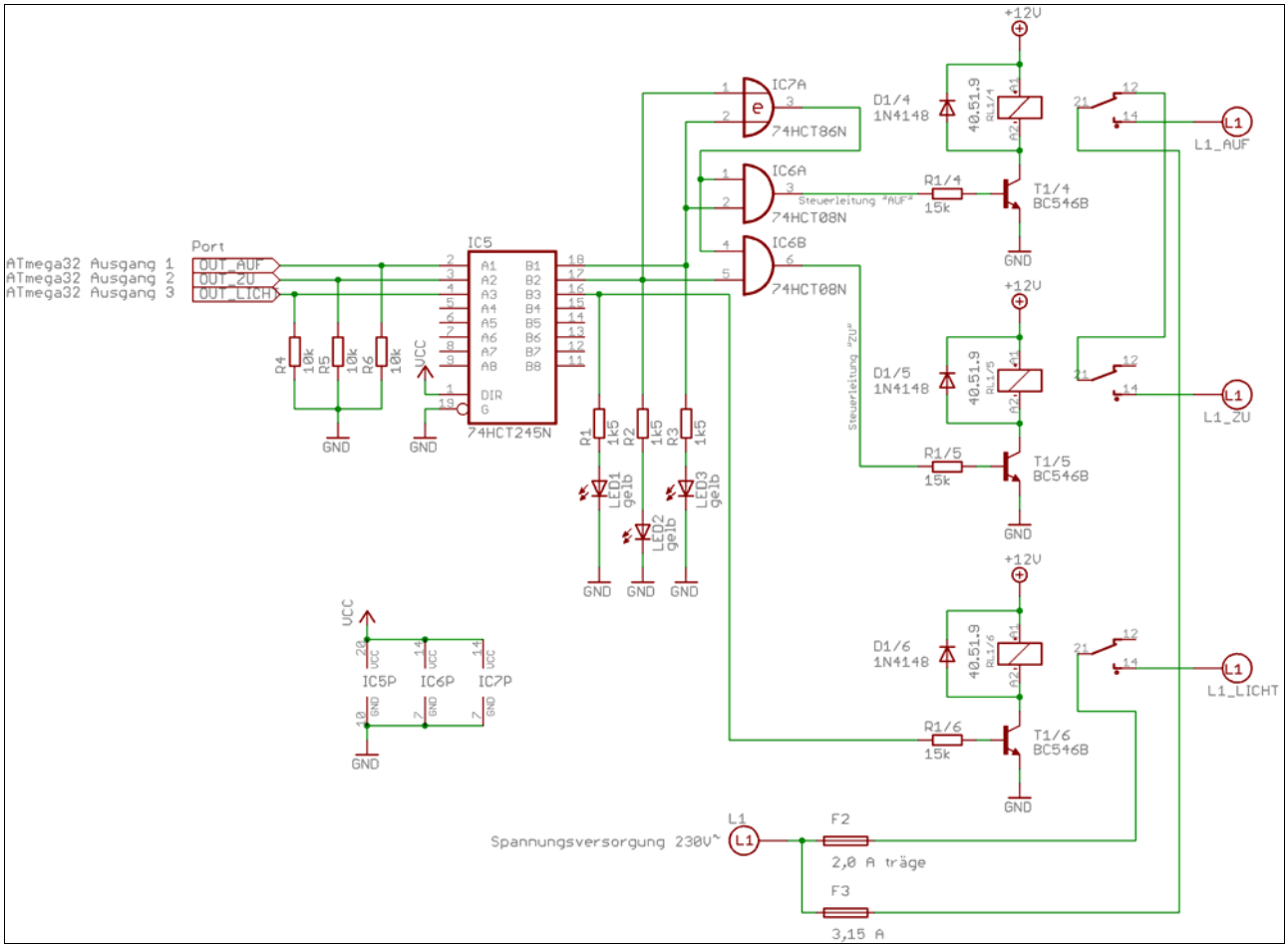
Der Prototyp für die Beleuchtungsanlage wurde im Rahmen des Gesamtprototypen für Relaissteuerung und Dimmung entwickelt. Siehe hierzu Kapitel „Prototyp für Relaisansteuerung und Dimmer“ im Kapitel „Lichtdimmer“

8.7 Gesamtschaltplan für Relaisansteuerung

Der folgende Schaltplan zeigt die gesamte Steuerung der drei im System vorhandenen Relais für Motor AUF-ZU und Spannung EIN für Dimmung inklusive Logikverriegelung und HW-Verriegelung für Motorsteuerteil sowie Treiberstufe für Signalleitungen zur Entlastung des ATmega und Ansteuerung von Status-LED’s.

Da im Reset- oder StartUp-Fall der ATmega - bis auf wenige Ausnahmen – seine Ports als hochohmige Eingänge schaltet, müssen die verwendeten Signalleitung OUT_AUF, OUT_ZU und OUT_LICHT mit 10k Ohm PullDown-Widerständen gegen floaten „abgesichert“ werden.

Dieser Sachverhalt viel bei der Inbetriebnahme des Prototypen auf bei dem es zu zufälligen Relais-Aktivitäten während der Initialisierung kam.



Schaltbild 24: Gesamtbeschaltung Relaisansteuerung

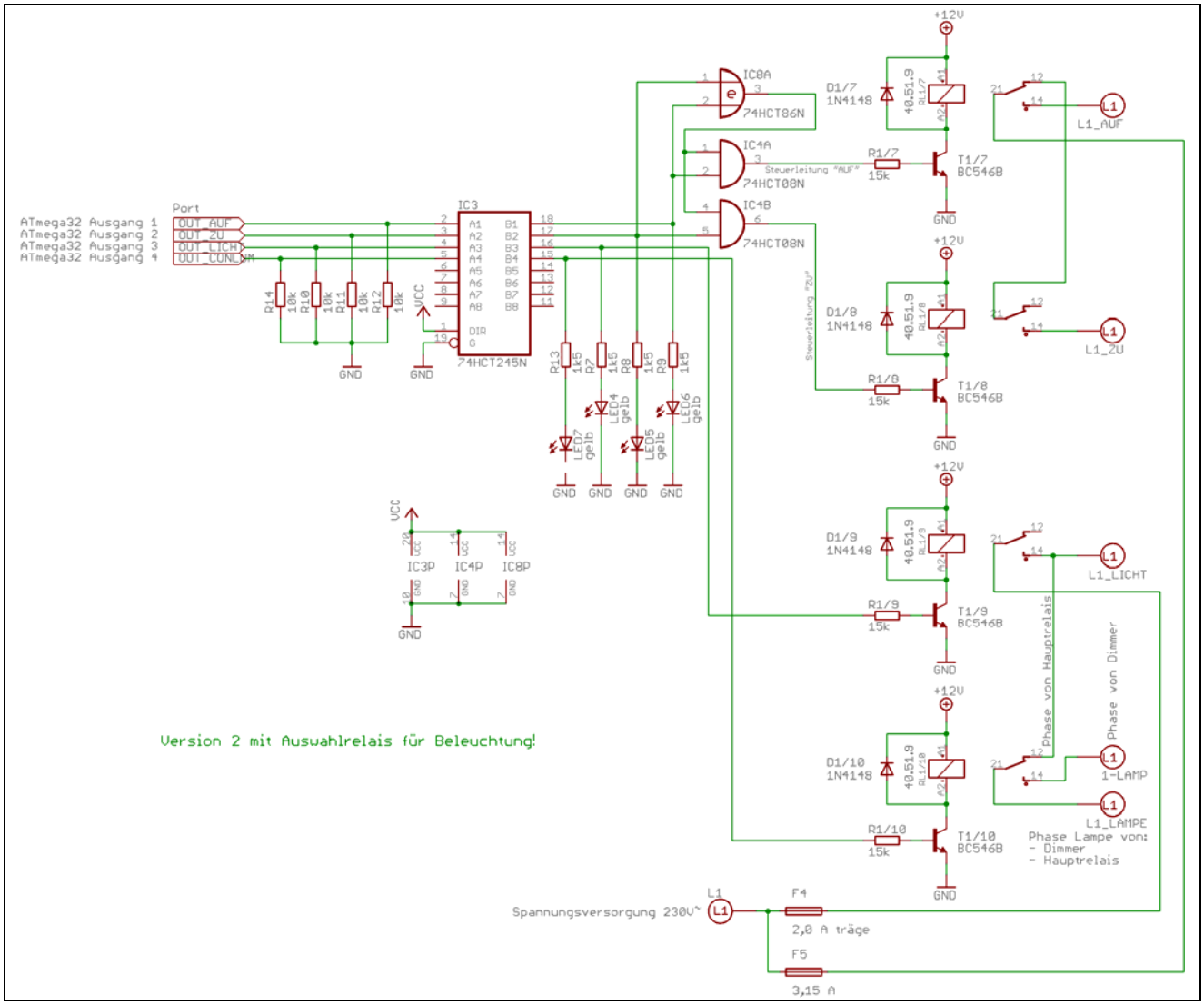
Bauteile:

Stückliste: Gesamtbeschaltung Relaisansteuerung			
Sonstiges		***Halbleiter***	
RL1/4, RL1/5, R1/6	finder Relais 40.51	T1/4, T1/5, T1/6	Transistor BC546B / BC547B
Widerstände		D1/4, D1/5, D1/6	Diode 1N4148
R1/4, R1/5, R1/6	15k Ω	IC5	Octal driver SN74HCT245N
R1, R2, R3	1k5 Ω	IC6	AND-Gatter SN74HCT08N
R4, R5, R6	10k Ω	IC7	XOR-Gatter SN74HCT86N
Sonstiges		LED1, LED2, LED3	LED 3mm, low current, grün
F2	Feinsicherung für Licht, 2,00 A träge		
F3	Feinsicherung Motorantrieb, 3,15A mitteträge		

Tabelle 73: Stückliste Gesamtbeschaltung Relaisansteuerung

Eine weitere Veariante kann so aussehen, dass ein zusätzliches Relai hinzugefügt wird mit dem eingestellt / bestimmt werden kann ob die Lampe konstant mit voller Leitungs brennt (der Dimmer gebypassed wird) oder ob die Lampe über den Dimmer versorgt wird.

Diese Variante ist im folgenden Schaltbild aufgeführt.



Schaltbild 25: Gesamtbeschaltung Relaisansteuerung

Bemerkung:
Version 2der Schaltung wird/wurde als Prototyp nicht aufgebaut da keine neue technische Funktion hinzugefügt wurde. Es wurde auf bereits existierende Funktionen zurückgegriffen die in Betrieb genommen sind.

Bauteile:

Stückliste: Gesamtbeschaltung Relaisansteuerung			
Sonstiges		***Halbleiter***	
RL1/7, RL1/8, R1/9, R1/10	finder Relais 40.51	T1/7, T1/8, T1/9, T1/10	Transistor BC546B / BC547B
Widerstände		D1/7, D1/8, D1/9, D1/10	Diode 1N4148
R1/7, R1/8, R1/9, R1/10	15k Ω	IC3	Octal driver SN74HCT245N
R7, R8, R9, R13	1k5 Ω	IC4	AND-Gatter SN74HCT08N
R10, R11, R12, R14	10k Ω	IC8	XOR-Gatter SN74HCT86N
Sonstiges		LED4-7	LED 3mm, low current, grün
F4	Feinsicherung für Licht, 2,00 A träge		
F5	Feinsicherung Motorantrieb, 3,15A mitteträge		

Tabelle 74: Stückliste Gesamtbeschaltung Relaisansteuerung

Themenbezogene LED's:

LED / Funktion / Beschreibung	*Abkürzung*	*Farbe*	
Relais Beschattung AUF	LED_REL_CONV_OPEN	gelb	*[LED21]*
Relais Beschattung ZU	LED_REL_CONV_CLOSE	gelb	*[LED22]*
Relais Licht (Versorgung Dimmer)	LED_REL_LIGHT	gelb	*[LED23]*
Relais Dauerlicht	LED_REL_CONLUM	gelb	*[LED24]*

Tabelle 75: Themenbezogene LED's für Relaissteuerung

Ressourcenzuordnung zum ATmega128:

Nummer	*Schaltbild*		*Ressource ATmega128*		
1	OUT_AUF	[CONV_AUF]	PortC.6	[PC6]	(GPIO)
2	OUT_ZU	[CONV_ZU]	PortC.7	[PC7]	(GPIO)
3	OUT_LICHT	[DIMMUNG]	PortG.0	[PG0]	(GPIO)
4	OUT_CONLUM	[CON_LUM]	PortE.2	[PE2]	(GPIO)

Tabelle 76: Ressourcenzuordnung ATmega128 für Relaisansteuerung

8.8 Lichtdimmung

8.8.1 Funktionsprinzip Phasenanschnittsteuerung

Je nach Lastcharakter gibt es verschiedene Arten der Dimmung. Bei Glüh- und Hochvolt-Halogenlampen (230 V) sowie bei konventionellen Transformatoren wird mit einer Phasenanschnittsteuerung gedimmt. Diese Verbraucher haben eine ohmsche oder eine induktive Lastcharakteristik.

Die Phasenanschnittsteuerung und die (seltener verwendete) Phasenabschnittsteuerung sind Methoden zur Leistungsregelung elektrischer Verbraucher, die mit Wechselspannung betrieben werden. Typische Anwendungen sind Dimmer und die Drehzahlregelung von Wechselstrommotoren (z.B. elektrische Handbohrmaschinen).

Bei der Phasenanschnittsteuerung wird der Stromfluss meist durch einen Triac gesteuert. Nach dem Nulldurchgang der Wechselspannung (und des Stromes) leitet der Triac den Strom so lange nicht, bis er einen Zündimpuls erhält; ab diesem Zeitpunkt (dieser "Phase" des Wechselstromsignals) wird der Verbraucher mit Energie versorgt (bis zum nächsten Nulldurchgang). Je später der Triac gezündet wird, desto geringer ist die (über eine Periode der Wechselspannung gemittelte) Leistung.

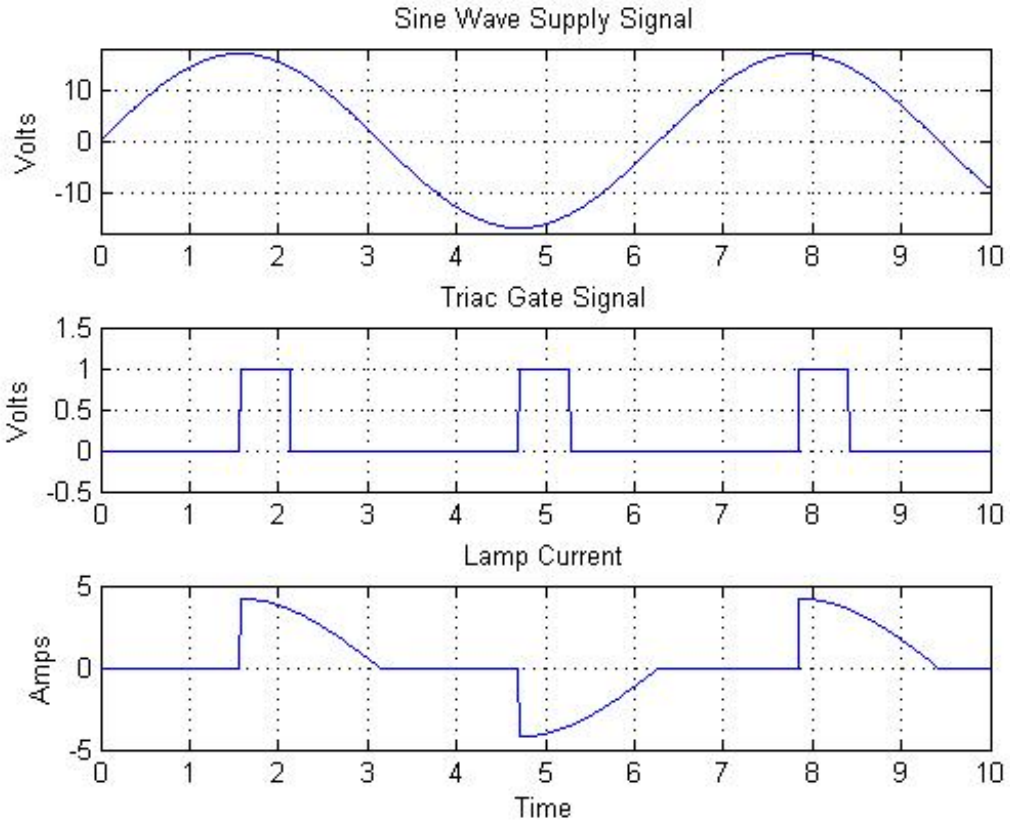


Abbildung 44: Funktionsprinzipt Phasenanschnittsteuerung

8.8.2 Phasenanschnitt- und Phasenabschnittsteuerung

Während bei der Phasenanschnittsteuerung der Strom verzögert nach dem Nulldurchgang der Wechselspannung eingeschaltet wird und bis zum nächsten Nulldurchgang anliegt, ist es bei der Phasenabschnittsteuerung genau umgekehrt: Der Strom wird nach dem Nulldurchgang sofort eingeschaltet, und noch vor dem nächsten Nulldurchgang ausgeschaltet. Der Schaltungsaufwand ist höher; es müssen entweder abschaltbare Thyristoren (GTO-Thyristor) oder (Power-MOSFET)-Transistoren verwendet werden, und auch die Steuerungselektronik ist aufwändiger.

Für kapazitive Lasten sind Phasenanschnittsteuerungen wegen des plötzlichen Spannungsanstiegs nicht geeignet (es würde ein extrem hoher Strom fließen), dafür werden Phasenabschnittsteuerungen eingesetzt. Hingegen sind Phasenabschnittsteuerungen nicht für induktive Lasten geeignet (Spannungsspitze beim Abschalten des Stromes).

Sogenannte Universaldimmer erkennen automatisch, ob eine induktive oder kapazitive Last vorliegt und wirken demnach als Phasenan- oder -abschnittsteuerung.

(a) Netzspannung als Funktion der Zeit (Sinusfunktion)

(b) Ausgangsspannung bei hoher Leistung.

(c) Ausgangsspannung bei niedriger Leistung.

Kleine rote Pfeile zeigen an, wann der Triac gezündet wird.

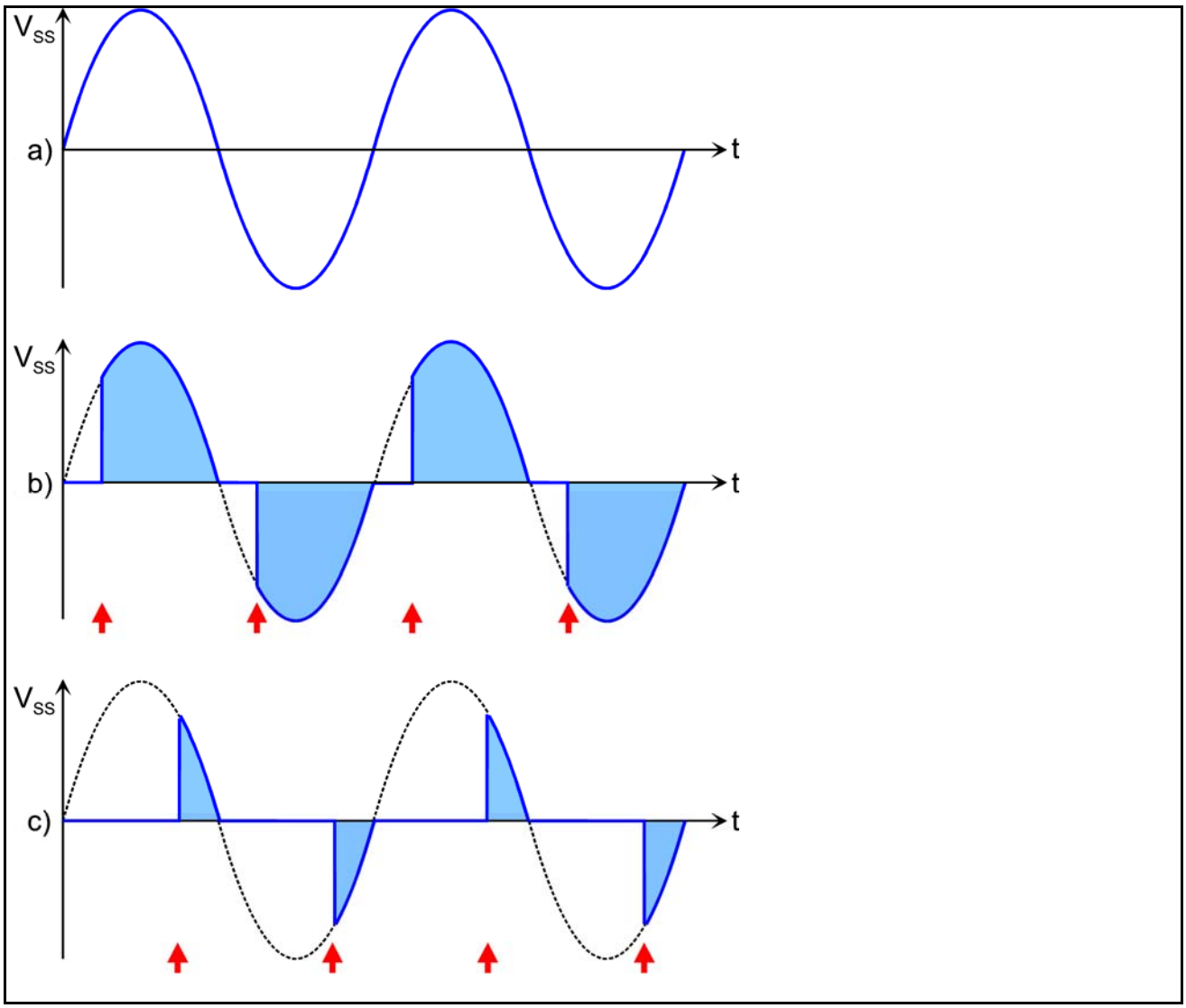


Abbildung 45: Funktionsprinzipt Phasenanschnittsteuerung

8.8.3 Vor- und Nachteile der Phasenanschnittsteuerung

Der Vorteil der Phasenanschnittsteuerung (und Phasenabschnittsteuerung) im Vergleich zu Schaltungen, bei denen die Spannung durch einen Widerstand geregelt wird, ist ihre sehr geringe Verlustleistung. Im Gegensatz zu komplizierten regelbaren Schaltnetzteilen, die auch geringe Verlustleistung haben, ist eine Phasenanschnittsteuerung wesentlich einfacher und kleiner aufzubauen und weniger störanfällig. Die ebenfalls zur Leistungssteuerung verwendeten Regeltransformatoren sind wesentlich größer und schwerer als Phasenanschnittsteuerungen.

Der größte Nachteil von Phasenanschnittsteuerungen (und Phasenabschnittsteuerung) ist der nicht-sinusförmige Verlauf des Stromes. Der Strom fließt ja nur in einem Teil jeder Halbwelle. Diese nicht-sinusformige Belastung ruft im Netz Störungen hervor. Je mehr die Kurvenform des Stromes von der Sinusform abweicht, desto größer sind die Störungen. Da Strom und Spannung nicht dieselbe Form besitzen, ergibt sich eine Blindleistung, die so genannte Steuer- oder Oberschwingungs-Blindleistung. Der Name Oberschwingungs-Blindleistung ergibt sich aus der Tatsache, dass ein von der Sinusform abweichendes Signal in der Darstellung als Spektrum Oberschwingungen besitzt.

Der Verbraucher erhält von der Phasenanschnittsteuerung oder Phasenabschnittsteuerung eine nicht-sinusförmige Spannung. Während das für ohmsche Verbraucher wie Glühlampen und Heizwiderstände kein Problem darstellt, können elektronische Geräte, die an eine Phasenanschnittsteuerung angeschlossen werden, dadurch beschädigt werden.

Auch bei Transformatoren, z.B. für Halogenlampen ist Vorsicht geboten: Es ist einerseits darauf zu achten, dass die Phasenanschnittsteuerung (d.h. der Dimmer) für diese Anwendung geeignet ist, andererseits darf man nur "echte" Transformatoren und keine elektronischen Spannungswandler, wie sie heute zunehmend eingesetzt werden, an Phasenanschnittsteuerungen anschließen. Es gibt jedoch Phasenabschnittsteuerungen, die für elektronische Spannungswandler geeignet sind. Elektronische Spannungswandler erkennt man daran, dass sie viel kleiner und leichter als Transformatoren gleicher Leistung sind. Mit Universaldimmern mit automatischer Lasterkennung können sowohl Transformatoren als auch elektronische Spannungswandler verwendet werden (aber nicht beide gleichzeitig!).

Die meisten Phasenanschnittsteuerungen (Dimmer) sind auch nicht für Leuchtstofflampen oder Energiesparlampen geeignet.

8.8.4 HW-Konzept für die Umsetzung im ConvCtrl

Bei der Phasenanschnittsteuerung ist der Verbraucher nach dem Nulldurchgang jeder Halbwelle zunächst stromlos.

Nach einer einstellbaren Zeit t wird dann ein Triac gezündet, der Strom fließen lässt. Ist eine gewisse Spannung unterschritten, sperrt der Triac wieder. Dieser Vorgang wiederholt sich mit einer Frequenz von 100 Hz bei positiver und negativer Halbwelle.

Je später der Triac gezündet wird, je niedriger ist die verfügbare Leistung und desto dunkler ist die Lampe.

Einstellbereich für Zeit t:

$$f = 2 * 50Hz = 100Hz$$

$$Hz = \frac{1}{\sec}$$

$$t = \frac{1}{f} = \frac{1}{100Hz} = 0{,}01\sec = 10ms$$

$$Einstellbereicht = 0 - 10ms$$

Schaltung:

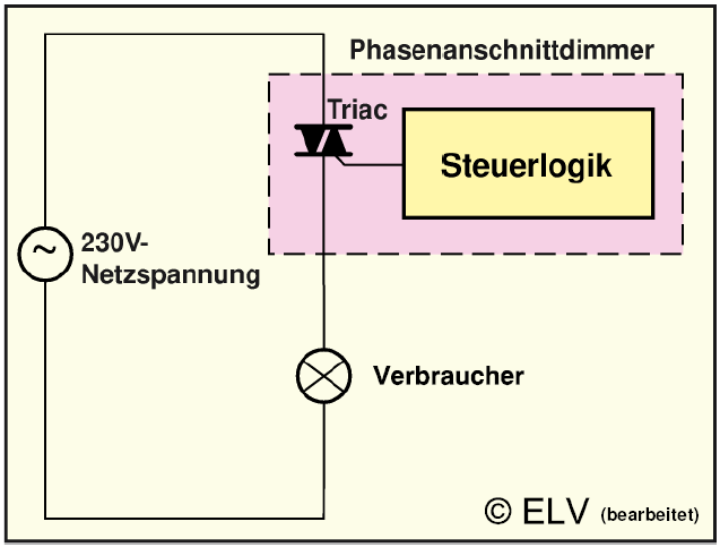
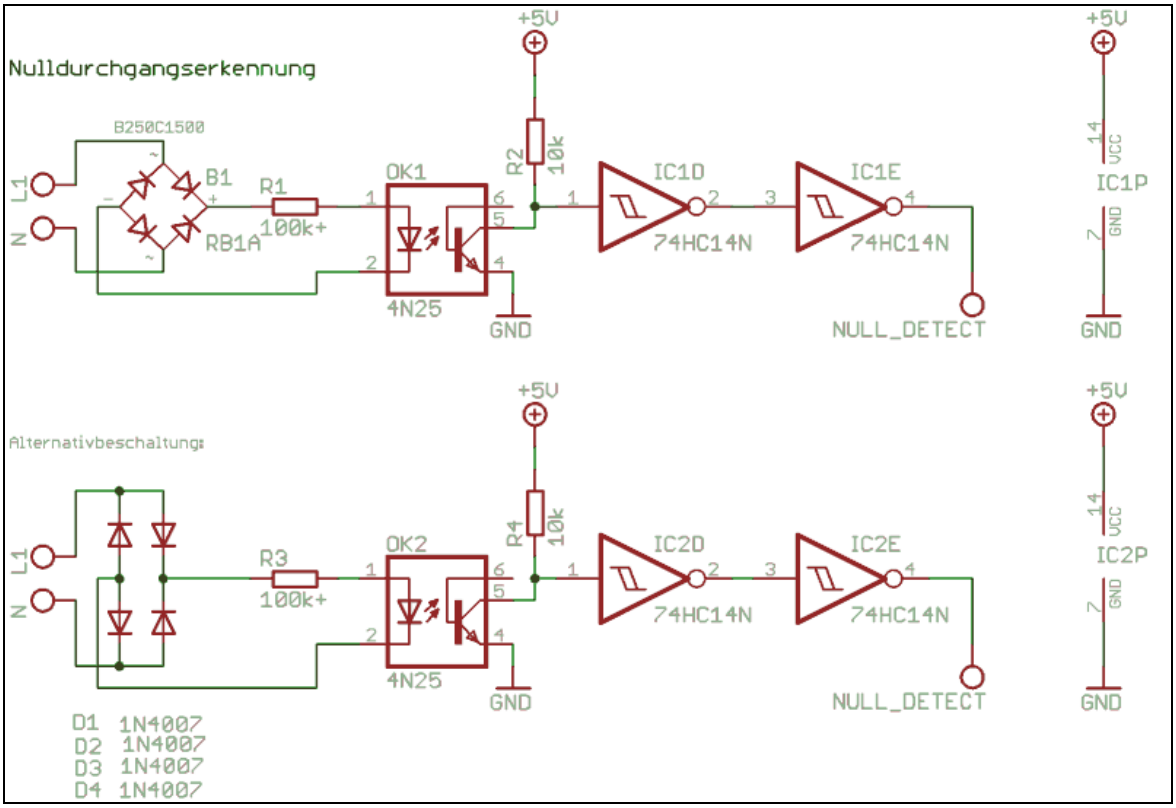


Abbildung 46: Steuerlogik für Phasenanschnitt

In einer elektronischen Schaltung wird der Triac von einer Steuerlogik gezündet wird. Diese Steuerung benötigt einen Nulldurchgangstest, damit sie den Nulldurchgang jeder Halbwelle registrieren kann.

Nulldurchgangserkennung:

Um den exakten Zündwinkel zu bestimmen, ist ein Nulldurchgangsdetektor unumgänglich. Dieser Detektor besteht aus einem (vom Netzteil unabhängigen) Gleichrichter (B1), einem Optokoppler 4N25 (OK1) und zwei Schmitt-Triggern (IC1A/B). Die im 100-Hz-Takt pulsierende gleichgerichtete Spannung hinter der Gleichrichterbrücke sorgt dafür, dass der Optokoppler bei jedem Nulldurchgang der Netzspannung kurzzeitig sperrt. In dieser Zeit liegen am Kollektor des Opto-Transistors +5 V, ansonsten 0 V. Dieses Signal – von den Schmitt-Triggern mit eindeutigen und steilen Flanken versehen - nutzt der Mikrocontroller, um die Software mit der Netzspannung zu synchronisieren.



Schaltbild 26: Nulldurchgangserkennung (Dimmer)

Bauteile:

Stückliste: Nulldurchgangserkennung			
Widerstände		*Halbleiter*	
R1	100 kΩ	B1	Brückengleichrichter B250C800DIP
R2	1 kΩ	OK1	Optokoppler 4N25
		IC1	Hex inverting Schmitt trigger 74HC14N
		Alternative: D1-D4	Diode 1N4007

Tabelle 77: Stückliste Nulldurchgangskennung

Ressourcenzuordnung zum ATmega128:

Nummer	*Schaltbild*	*Ressource ATmega128*
1	NULL_DETECT [INT_DIM] [INT0]	PortE.5 [INT5] (Interrupt 5)

Tabelle 78: Ressourcenzuordnung ATmega128 für Nulldurchgangserkennung

Zeitverhalten Nulldurchgangserkennung:

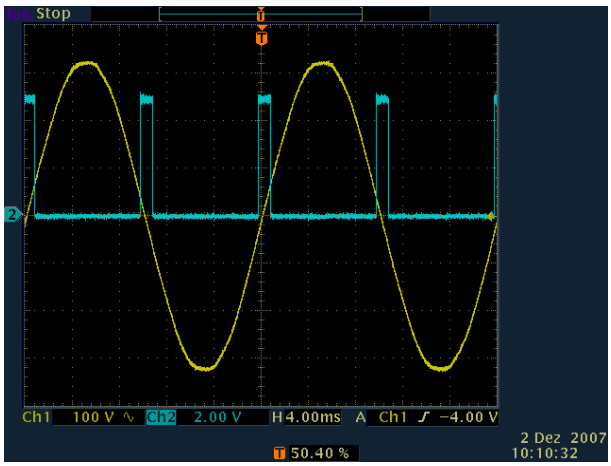
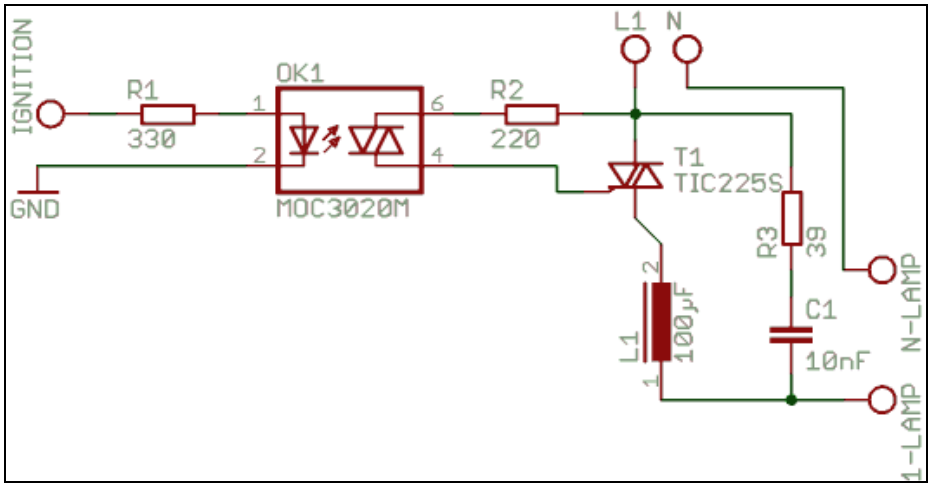


Abbildung 47: Oszillogramm der Nulldurchgangserkennung

Leistungsteil für Zündstufe:

Nach der Erkennung eines Nulldurchgangs fängt eine Stoppuhr oder Timer an zu zählen, bis zu der gewünschten Zeit, an der Triac gezündet werden soll.

Die Zündung des Triacs erfolgt über ein weiteres HW-Modul.



Schaltbild 27: Leistungsteil (Dimmer)

Bauteile:

Stückliste: Leistungsteil			
Widerstände		*Halbleiter*	
R1	330 Ω bzw. 680 Ω	OK1	Optokoppler MOC3020
R2	220 Ω bzw. 330 Ω	T1	Triac TIC225S
R3	39 Ω	*Sonstiges*	
Kondensatoren		L1	Speicherdrossel 100 µF
C1	10 nF MKS4 ~230V		

Tabelle 79: Stückliste Leistungsteil Dimmer

Ressourcenzuordnung zum ATmega128:

Nummer	*Schaltbild*	*Ressource ATmega128*
1	IGNITION [TRIAC]	PortD.4 [PD4] (GPIO)

2	--	Timer2	Zündzeitpunkt TRIAC

Tabelle 80: Ressourcenzuordnung ATmega128 für Triac-Zündstufe

Zeitverhalten TRIAC-Zündstufe und geschaltete Phase:

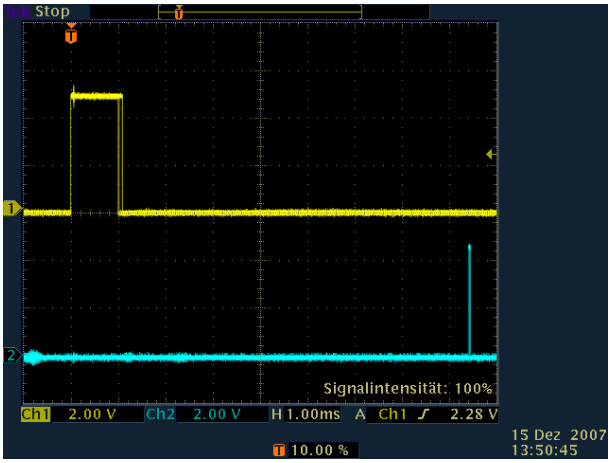


Abbildung 48: Oszillogramm Triac-Zündung „dunkel“

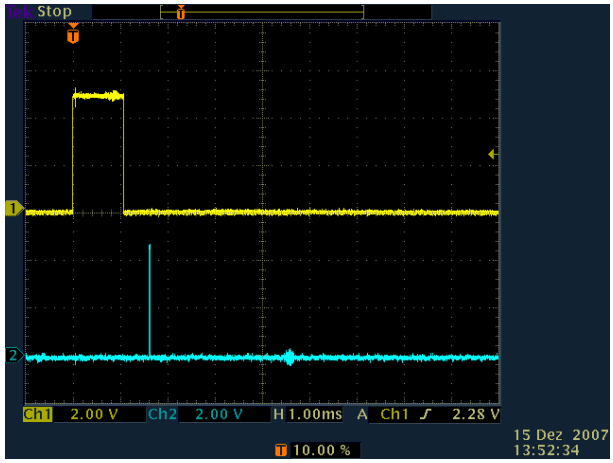


Abbildung 49: Oszillogramm Triac-Zündung „hell“

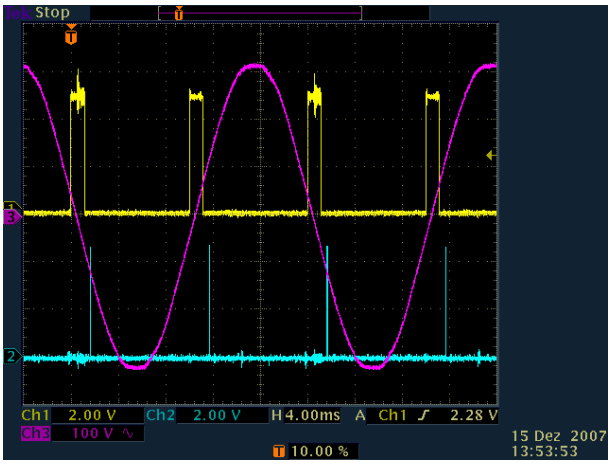


Abbildung 50: Oszillogramm aller Signal inkl. Netz-Sinus

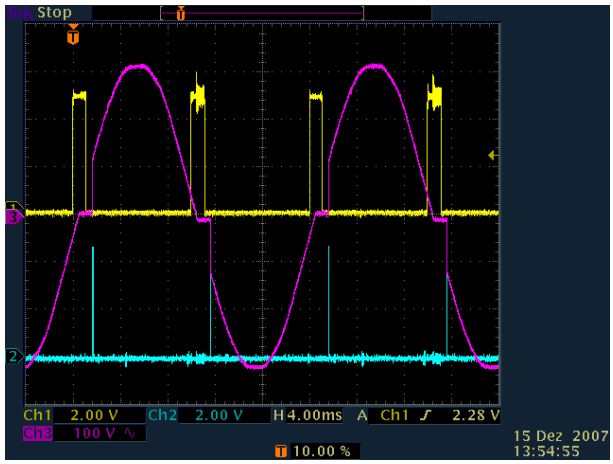


Abbildung 51: Oszillogramm Triac-Ausgangssignal „hell“

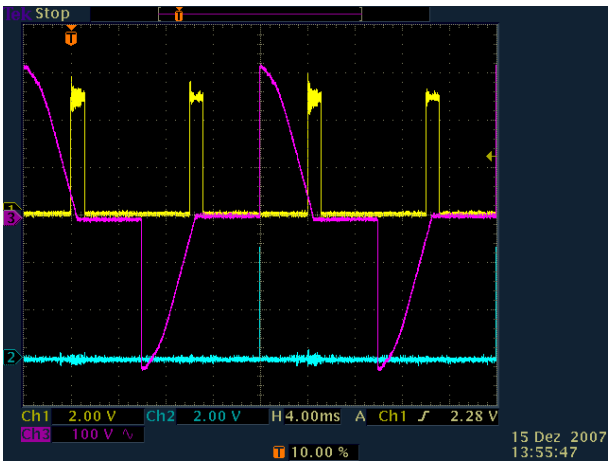


Abbildung 52: Oszillogramm Triac-Ausgangssignal „50%“

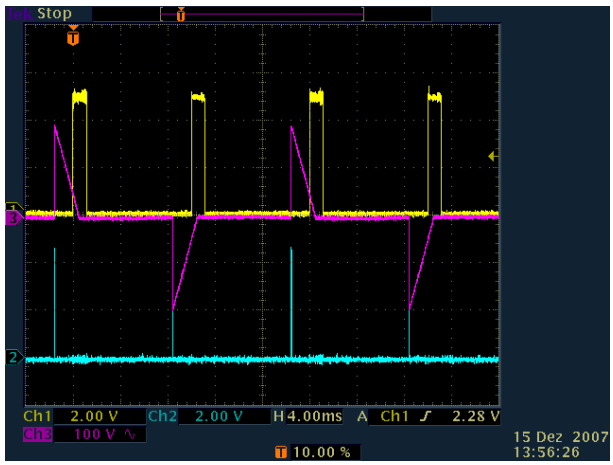


Abbildung 53: Oszillogramm Triac-Ausgangssignal „dunkel“

Bemerkung zu den Oszillogrammen:

Ch1 (gelb) ist das Signal der Nulldurchgangserkennung des Netz-Sinus.

Ch2 (hellblau) ist das Zündungssignal (20µs) für den Triac.
Ch3 (magenta) ist wahlweise die Abbildung des Netz-Sinus oder die durch den Triac „phasenangeschnittene" Sinuswelle.

8.8.5 SW-Konzept für Umsetzung

Die Programmierung der möglichen Zündzeit von max. 10 ms erfolgt in 32 Einzelschritten (Helligkeitsstufen) damit eine scheinbare stufenlose Dimmung ermöglicht wird. Der Zeitraum von 10 ms wird in 100 Stützstellen aufgeteilt. Eine Umsetzung der 32 Einzelschritte auf die Stüzstellen erfolgt über eine Tabelle (siehe hierzu Linearisierung der Helligkeitswerte und die dort enthaltene Konvertierungstabelle).

Mit Hilfe eines externen Interrupts wird in der SW der Nulldurchgang der Sinuswelle detektiert. Dieser „Detektions"-Interrupt startet einer Timer welcher mit dem Dimmungswert zwischen 0 und 10ms geladen ist.

Ist der Timer abgelaufen (= Auslösen des Timer-Interrupts) wird der Triac gezündet. Das Zünden des Triac zum entsprechenden Zeitpunkt nach dem letzten Nulldurchgang durch geschieht durch Setzen des der Lampe entsprechenden Bits im Port. Das Bit sollte recht schnell wieder gelöscht werden (minimal 25 µs), um nicht unnötig lange Strom durch das Triac-Gate fliessen zu lassen.

Der Kontroller wird mit diesem Verfahren nicht signifikant belastet. Die komplette Dimmungssteuerung erfolgt über diesen Mechanismus.

Mögliche Optionen:

In der SW wird ein Minimum-Wert und ein Maximum-Wert definiert. Der Min- und Max-Wert begrenzt die via Tasten veränderbare Timer-Zeit. Hierüber kann eine minimale und maximale Helligkeit vorgegeben werden.

8.8.6 Linearisierung der Helligkeitswerte

Hier ist eine Halbwelle der Netzspannung abgebildet. Die dickgezeichnete Funktion $u(t)$ ist phasenangeschnitten, die gestrichelte Linie stellt den nicht angeschnittenen Phasenverlauf dar:

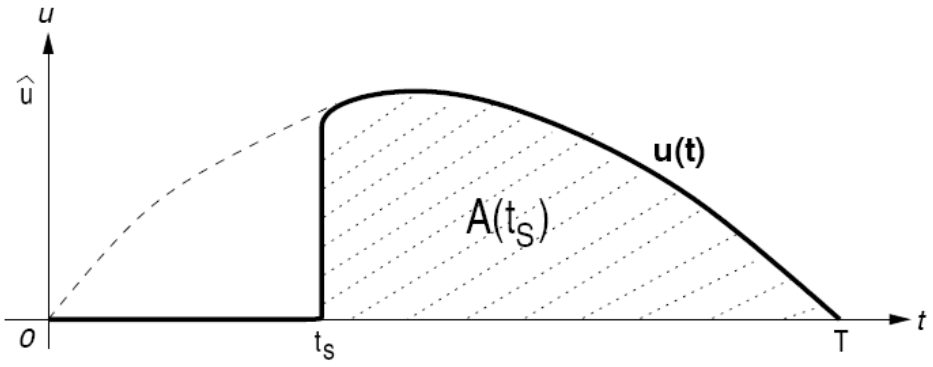


Abbildung 54: Angeschnittener Phasenverlauf

Dabei ist T die Dauer einer Halbwelle (z. B. in der vorliegenden Software 100) und t_s der Zündzeitpunkt nach dem Nulldurchgang, gemessen in der Einheit von T. $\hat{u}$ ist die Amplitude der Netzspannung, die aber aus Linearitätsgründen nicht von Belang ist.

Eine nichtangeschnittene Halbwelle wird durch $u(t) = \hat{u} \cdot \sin\left(\frac{\pi}{T} t\right)$ beschrieben. Die Fläche $A(t_S)$ ist demnach

$$A(t_S) = \int_{t_S}^{T} u(t)dt = \frac{\hat{u}T}{\pi}\left(1 + \cos\left(\frac{\pi}{T}t\right)\right)$$

Der lineare Helligkeitswert $p \in [0,1]$ berechnet sich zu $p = \frac{A(t_S)}{A(0)}$.

So ergibt sich durch Einsetzen und Umstellen die Formel, die aus einem gegebenen Helligkeitswert p den gesuchten Zündzeitpunkt t_S bestimmt:

$$t_S = \frac{T}{\pi}\arccos(2p - 1)$$

In der vorliegenden Software wurde eine Halbwelle mit Hilfe eines Timers in 100 Abschnitte zerlegt, d. h. dort ist $T = 100$. Die Berechnung der Zündzeitpunkte erfolgte mit einer Tabelle

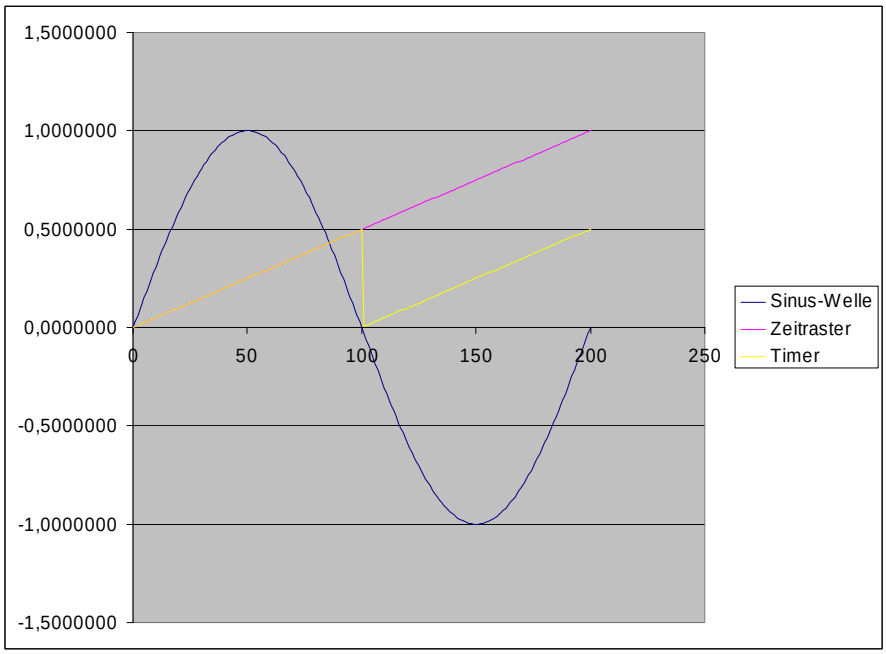


Abbildung 55: Sinus mit Timer-Rampen

Konvertierungstabelle:

Zeitraum Halbwelle	10 ms	0,010000	
Timer-Tick Timer0	64,0µs	0,000064	Prescaler 1024

Helligkeit	Stützstelle	Verzögerung	Counterregister
1 (dunkel)	100	0,01000	156
2	89	0,00890	139
3	84	0,00840	131
4	80	0,00800	125
5	77	0,00770	120
6	74	0,00740	116
7	71	0,00710	111
8	69	0,00690	108
9	66	0,00660	103
10	64	0,00640	100
11	61	0,00610	95
12	59	0,00590	92
13	57	0,00570	89
14	55	0,00550	86
15	53	0,00530	83
16	51	0,00510	80
17	49	0,00490	77
18	47	0,00470	73
19	45	0,00450	70
20	43	0,00430	67
21	41	0,00410	64
22	39	0,00390	61
23	36	0,00360	56
24	34	0,00340	53
25	32	0,00320	50
26	29	0,00290	45
27	26	0,00260	41
28	23	0,00230	36
29	20	0,00200	31
30	16	0,00160	25
31	11	0,00110	17
32 (hell)	0	0,00000	0
		[Sekunden]	

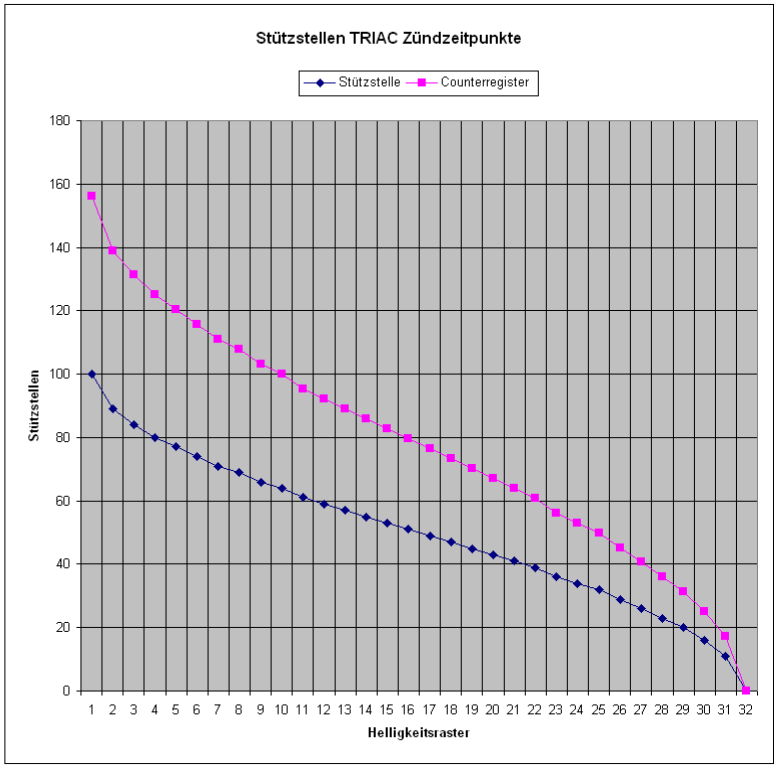


Abbildung 56: Konvertierungstabelle TRIAC Zündzeitpunkte

Da der Timer als bottom-up Timer umgesetzt ist und bei einem Überlauf einen Interrupt erzeugt werden nicht die Zahlenwerte im Counterregister in die Datenstruktur eingetragen sondern die zugehörigen Einerkomplemente (255 - Counterregister). Dies wir in der SW entsprechend berücksichtigt.

```
'-----------------------------------------------------------------------------
' Compilerinstruktionen und Compilerdirektiven
'-----------------------------------------------------------------------------
$regfile = "m128def.dat"
$crystal = 16000000
$hwstack = 32
$swstack = 32
$framesize = 40

$baud = 38400

' -----------------------------------------------------------------------------
' BASCOM Zündzeitpunkt über Tabelle in Datenstruktur laden
' -----------------------------------------------------------------------------

' --- Datendefinition ---

Dim Zuendzeitpunkte(32) As Byte

Dim Indexcounter As Byte


' --- Daten aus Speicher in Datenstruktur einlesen ---
Restore Triac_zuendtabelle

For Indexcounter = 1 To 32
   Read Zuendzeitpunkte(indexcounter)
Next Indexcounter

' --- Hauptschleife ---
Do
   ' Tue nix
Loop

End

' -----------------------------------------------------------------------------
' BASCOM Datendefinition für TRIAC
' -----------------------------------------------------------------------------
Triac_zuendtabelle:
Data 156 , 139 , 131 , 125 , 120 , 116 , 111 , 108
Data 103 , 100 ,  95 ,  92 ,  89 ,  86 ,  83 ,  80
Data  77 ,  73 ,  70 ,  67 ,  64 ,  61 ,  56 ,  53
Data  50 ,  45 ,  41 ,  36 ,  31 ,  25 ,  17 ,   0
```

```
' ------------------------------------------------------------------------
```

Software 1: Code zur Initialisierung der TRIAC-Zündzeitpunkte

Daten:

Helligkeitswerte über Konvertierungstabelle	Index 1 = Stützstelle 100 = dunkel Index 32 = Stützstelle 0 = hell
Frequenz	50 Hz
Frequenz pro Halbwelle	100 Hz
Nulldurchgänge	100 Hz
Zeit pro Halbwelle	500 ms
Zeit pro Stützstelle (100 pro Halbwelle)	5 ms

Die Werte aus der Konvertierungstabelle sind zum laden des Timers entweder mit der Timerfrequenz zu verrechnen oder der Timer ist so einzustellen, das er im 5 ms Raster arbeitet.

Detailrechung Timer:

Timer und Prescaler

Clock1	3.686.400
Clock2	**16.000.000**

Prescaler	Clock1	Clock2
1	3686400	**16000000**
8	460800	**2000000**
32	115200	**500000**
64	57600	**250000**
128	28800	**125000**
256	14400	**62500**
1024	3600	**15625**

Prescaler	1	8	64	256	1024
Clock1	3686400	460800	57600	14400	3600
Clock2	**16000000**	**2000000**	**250000**	**62500**	**15625**

Zeitdauer Zyklus Clock 1	2,712674E-07	= 0,0000002712674	271,267 ns	(Nanosekunden)
Zeitdauer Zyklus Clock 2	**6,250000E-08**	**= 0,0000000625000**	**62,500 ns**	**(Nanosekunden)**

Zykluszeiten				
Prescalerzeiten	**Clock1**	**Clock2**	**Clock1**	**Clock2**
1	0,00000027126736	**0,000000062500**	271,267 ns	**62,500 ns**
8	0,00000217013889	**0,000000500000**	2,170 µs	**500,00 ns**
32	0,00000868055556	**0,000002000000**	8,68 µs	**2,000 µs**
64	0,00001736111111	**0,000004000000**	17,361 µs	**4,000 µs**
128	0,00003472222222	**0,000008000000**	34,722 µs	**8,000 µs**
256	0,00006944444444	**0,000016000000**	69,444 µs	**16,000 µs**
1024	0,00027777777778	**0,000064000000**	277,778 µs	**64,000 µs**

Netzabhängiges Timing für Dimmer					
Netzfrequenz	50 Hz	0,02	20 ms	0,0002	200 µs
Halbwellen	**100 Hz**	**0,01**	**10 ms**	**0,0001**	**100 µs**
Stützstellen	100		Sinuswelle		pro Stützstelle

Timer Compareregistereinstellungen für Zeitraum 10 ms			
Prescaler	**Clock1**	**Clock2**	
1	36864	**160000**	
8	4608	**20000**	
32	1152	**5000**	
64	576	**2500**	
128	288	**1250**	
256	144	**625**	
1024	36	**156,25**	***=> Timerload 0-156 bzw. 99-255***

Bemerkung:
Es soll der Timer0 eingesetzt werden
Der Timer0 ist ein 8-Bit Timer = 0-255

Prescaler	Timerwert 1	Timerwert 2	Abweichung 1 in %	Abweichung 2 in %
1	368	1600	0,17	0,00
1	369	--	-0,10	--
8	46	200	0,17	0,00
64	5	25	13,19	0,00
64	6	--	-4,17	--
256	1	6	30,56	4,00
256	2	7	-38,89	-12,00
1024	1	1	-177,78	36,00
1024	--	2	--	-28,00

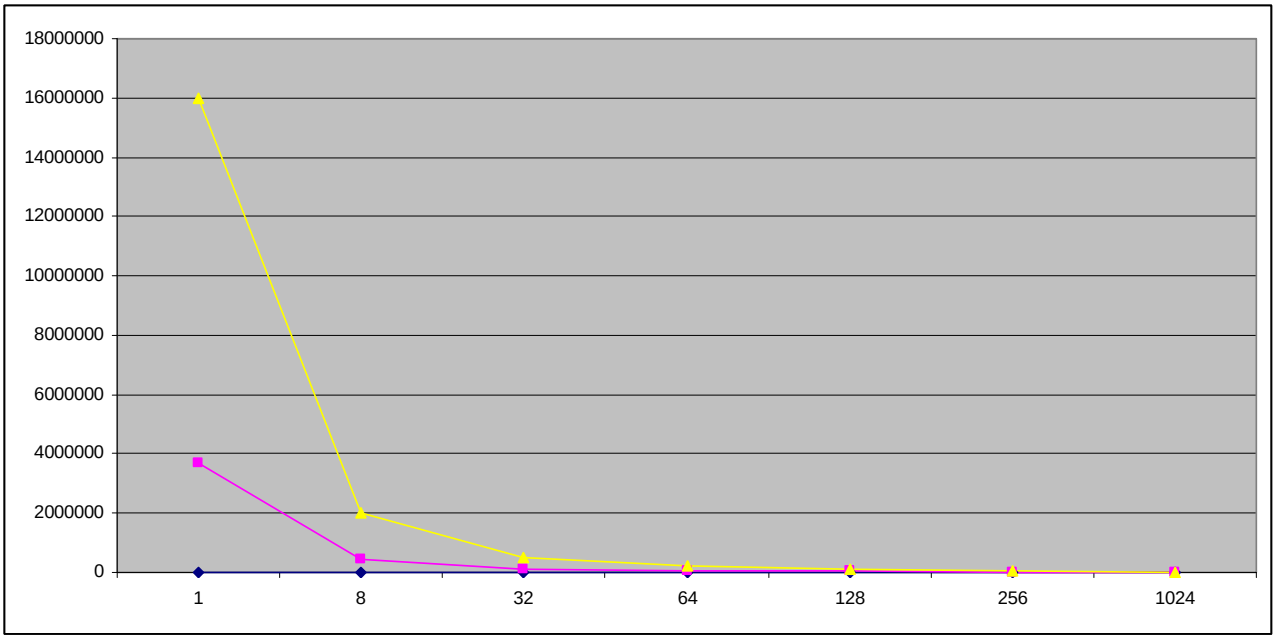


Abbildung 57: Detailrechung Timer für Dimmer-Zündungssteuerung

Bemerkung:

Im Projekt wird nach Inbetriebnahme des Dimmers nicht wie ursprüngliche geplant der Timer0 sondern der Timer2 eingesetzt. Damit erhöht sich die Interrupt-Priorität und es kann darüber eine „flackerfreie“ Zündung des Triac gewährleistet werden.

8.8.7 Detailbeschreibung SW

Die fallende Flanke des 100Hz Signals löst einen Interrupt aus. In der Interrup-Service-Routine der Nulldurchgangskennung wird ein Timer mit dem TRIAC-Zündzeitung aus der Zündtabelle entsprechend der aktuell eingestellten Helligkeit geladen.

Erreicht der Timer den Inhalt des Vergleichsregisters bzw. ist er abgelaufen, wird ein Timer-Interrupt ausgelöst. Dieser Timer-Interrupt gibt einen Impuls von ca. 80µs Dauer (minimal 25µs) auf einen PortD.4 aus.

Je nach Inhalt des Vergleichsregisters bzw. des Timer-Inhalts erscheint der Ausgangsimpuls also zu unterschiedlichen Zeiten realtiv zur fallenden Flanke des 100Hz Signals. Der Ausgangsimpuls ist der Zündimpuls für den Triac.

Bemerkung:

Bei dem eingesetzten Quarz mit einer Frequenz von 16 MhZ beträgt ein Clock-Zykle 62,50ns. Um ein Delay von 25µs zu erreichen ist es notwendig 400 Taktzyklen zu warten.

8.8.8 BASCOM Beispielcode für Relaisansteuerung und Dimmer

```
'#############################################################################
' RelaisUndLicht.BAS:                                        Stand 11.12.2007
' -------------------                                        (C) Markus Fulde
' Testprogramm zur Inbetriebnahme der Relais für Beschattung und
' Dimmung.
' Die Inbetriebnahme erfolgt mit dem Eva-Board STK500 + STK501
' Die für Ansteuerung von Relais und Dimmer notwendigen Routinen
' sind als solche gekennzeichnet und bereits zur späteren Verwendung
' ausgelagert.
'#############################################################################


'-----------------------------------------------------------------------------
' Compilerinstruktionen und Compilerdirektiven
'-----------------------------------------------------------------------------
$regfile = "m128def.dat"
$crystal = 16000000
```

```
$hwstack = 32
$swstack = 32
$framesize = 40

$baud = 38400

'-------------------------------------------------------------------------------
' Definition von Ressourcen
'-------------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========

' REL: Alias für Relais-Steuerung
Out_auf Alias Pinc.6                                    ' für Relais Beschattung auf
Out_zu Alias Pinc.7                                     ' für Relais Beschattung zu
Out_licht Alias Ping.0                                  ' für Relias Licht an

Rel_auf_on Alias Portc.6
Rel_zu_on Alias Portc.7
Rel_licht_on Alias Portg.0

' DIM: Alias für Dimmer_Steuerung
Null_detect Alias Pine.5                                ' für Input-PIN
Null_detect_pin Alias Porte.5                           ' für PullUp

Ignition Alias Pind.4                                   ' für Trias-Zündung
Triac_start Alias Portd.4


'========== Für Testumgebung  ==========

' Alias Namen für Tastaturabfragen
Taste_1 Alias Pina.0
Taste_2 Alias Pina.1
Taste_3 Alias Pina.2
Taste_4 Alias Pina.3
Taste_5 Alias Pina.4
Taste_6 Alias Pina.5
Taste_7 Alias Pina.6
Taste_8 Alias Pina.7

' Alias Namen für LED's
Led0 Alias Portb.0
Led1 Alias Portb.1
Led2 Alias Portb.2
Led3 Alias Portb.3
Led4 Alias Portb.4
Led5 Alias Portb.5
Led6 Alias Portb.6
Led7 Alias Portb.7



'-------------------------------------------------------------------------------
' Definition von Variablen, Konstaten und Datentypen
'-------------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========

' REL: Konstanen
Const Passiv = 0
Const Aktiv = 255
Const Beschattung_delay = 10
Const Relais_testmodus = 0

' DIM: Konstanten
Const Dimmer_testmodus = 0
Const Dim_helligkeit_default = 16
Const Dim_helligkeitstabellenwerte = 32
Const Dim_helligkeit_min = 3
Const Dim_helligkeit_max = 30

' REL: Variablen definieren
Dim Beschattung_auf_flag As Byte
Dim Beschattung_zu_flag As Byte
Dim Beschattung_auf_counter As Word
Dim Beschattung_zu_counter As Word

' DIM: Variablen definieren
Dim Sinuscounter As Word
```

```
Sinuscounter = 0

Dim Triac_zuenszeitpunkte(32) As Byte
Dim Dim_helligkeitsstufe As Byte
Dim Dim_index_counter As Byte


'========== Für Testumgebung ==========

' Konstanten
Const Sequencer_testmodus = 0
Const Timer_testmodus = 0
Const Main_testmodus = 0
Const Timervorgabe = 3036                                   ' Timer von 1 Sekunden

Const Led_aus = 1
Const Led_an = 0

' Sonstige Variablen definieren
Dim Timercounter As Word
Timercounter = 0

Dim Beleuchtung_flag As Byte
Beleuchtung_flag = Passiv

Dim Free_running_flag As Byte
Free_running_flag = Passiv


'--------------------------------------------------------------------------------
' Prototyping
'--------------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========

'========== Für Testumgebung ==========


'--------------------------------------------------------------------------------
' Konfiguration und Basiseinstellungen (Projekt und Testumgebung)
'--------------------------------------------------------------------------------

'========== Für Testumgebung ==========

'Konfiguration des Port A als Eingang für Tastaturabfrage
Config Porta = Input

'Konfiguration des Port B als Ausgang für Statusanzeige via LED
Config Portb = Output
Portb = &B11111111

'Konfiguration eines Timers für 1 Sekunden Timer-Tick
Config Timer1 = Timer , Prescale = 256                      ' Timer 1 verwenden
On Timer1 Timer_irq                                         ' Interrupt Routine
Timer1 = Timervorgabe


'========== Für Projekt ConvCtrl ==========

'-- Initialisierung Relais --

' REL: Routinen und Einstellungen
Gosub Relais_konfig

'Hier werden die Timer aktiviert
Enable Timer1                                               ' Timer 1 freigeben

'-- Initialisierung Dimmer --

' DIM: Interrupt für Nulldurchgangserkennung konfigurieren
Gosub Dimmer_konfig

' INT5 so konfiguriert das Trigger bei steigender  Flanke
Config Int5 = Rising
' Springe zur Interrupt-Routine Dimmer_irq
On Int5 Dimmer_null_irq
Enable Int5                                                 ' Interrupt5 freigeben


'Konfiguration Timer2 für Zündung TRIAC nach Vorgabe bzw. Nulldurchgang
```

```
Config Timer2 = Timer , Prescale = 1024
On Timer2 Dimmer_triac_irq
Enable Timer2


'-- Interrupts global freigeben --
Enable Interrupts

'-- Timer0 sofort anhalten und erst über SW anstossen
Stop Timer2

'========== Für Testumgebung ==========

#if Main_testmodus
   Print "** Enter MainLoop **"
   Print
#endif


'################################################################################
' Hauptprogramm
'################################################################################
'Hier ist die Programmhauptschleife
Do
   ' ===== Tasten für Relaissteuerung =====

   ' Tastaturabfrage Taste Beschattung ganz auf
   Debounce Taste_1 , 0 , Relais_beschattung_auf , Sub

   ' Tastaturabfrage Taste Beschattung ganz auf
   Debounce Taste_2 , 0 , Relais_beschattung_zu , Sub

   ' Tastaturabfrage Taste Beschattung stopp
   Debounce Taste_3 , 0 , Relais_beschattung_stop , Sub

   'Tastaturabfrage für Toggle-Funktion Licht EIN/AUS
   Debounce Taste_4 , 0 , Beleuchtung_ein_aus , Sub


   ' ===== Tasten für Dimmersteuerung =====
   ' Tastaturabfrage Taste Licht heller
   Debounce Taste_5 , 0 , Dimmer_heller , Sub

   ' Tastaturabfrage Taste Licht dunkler
   Debounce Taste_6 , 0 , Dimmer_dunkler , Sub

   ' Tastaturabfrage für Toglle-Funktion Dimmer EIN/AUS
   Debounce Taste_7 , 0 , Dimmer_free_running , Sub

   ' Taste für Triac ONE-Shot
   Debounce Taste_8 , 0 , Dimmer_one_shot , Sub

   ' ===== SW-Timer-Funktionen =====
   Gosub Sw_timer_sequencer

Loop
'## End Hauptprogramm ##########################################################


'********************************************************************************
' Interruptroutinen
'********************************************************************************
'-------------------------------------------------------------------------------
' Interrupt-Service-Routine (Timer1): Timer_irq
' Routine zur Auswertung des Timer Interrupts
'-------------------------------------------------------------------------------
Timer_irq:
   Incr Timercounter                                    ' Timercounter erhöhen
   Toggle Led7                                          ' Status LED toggeln
   Timer1 = Timervorgabe                                ' Timer neu laden

   #if Timer_testmodus
      Print Timercounter
   #endif

   #if Dimmer_testmodus
      Print Sinuscounter
   #endif

Return
```

```
'-- End Timer_irq ------------------------------------------------------------


'------------------------------------------------------------------------------
' Interrupt-Service-Routine (INT5): Dimmer_null_irq
' Routine zur Nulldurchgangserkennung des Sinussignals
'------------------------------------------------------------------------------
Dimmer_null_irq:
   ' In dieser IRQ-Routine wird wegen Geschwindigkeit auf Ausgaben verzichtet
   If Free_running_flag = Aktiv Then

   #if Dimmer_testmodus
      Incr Sinuscounter
   #endif

      Start Timer2
      Timer2 = Triac_zuenszeitpunkte(dim_helligkeitsstufe)
   End If

Return
'-- End Dimmer_null_irq ----------------------------------------------------


'------------------------------------------------------------------------------
' Interrupt-Service-Routine (Timer0): Dimmer_triac_irq
' Routine zum Zünden des TRIAC
'------------------------------------------------------------------------------
Dimmer_triac_irq:
   Stop Timer2
   Triac_start = 1
   Waitus 20
   Triac_start = 0
Return
'-- End Dimmer_triac_irq --------------------------------------------------


'******************************************************************************
' Subroutinen
'******************************************************************************


'******************************************************************************
' GOSubroutinen
'******************************************************************************

'------------------------------------------------------------------------------
' REL - Gosub-Routine: Relais_konfig
' Subroutine zur Konfiguration der Relais
'------------------------------------------------------------------------------
Relais_konfig:

   '----- Konfiguration Relais [AUF/ZU/Licht] (Transistorstufe) -----
   'Port G.0 (Licht) als Ausgang definieren und Relais ausschalten
   Config Out_licht = Output
   Rel_licht_on = 0

   'Port C.6 (AUF) als Ausgang definieren und Relais ausschalten
   Config Out_auf = Output
   Rel_auf_on = 0

   'Port C.7 (ZU) als Ausgang definieren und Relais ausschalten
   Config Out_zu = Output
   Rel_zu_on = 0

Return
'-- End Relais_konfig ----------------------------------------------------


'------------------------------------------------------------------------------
' REL - Gosub-Routine: Relais_auf_an
' Subroutine zum Eimschalten des Relais AUF
'------------------------------------------------------------------------------
Relais_auf_an:

   #if Relais_testmodus
      Print "Enter Relais_auf_an"
   #endif

   Rel_auf_on = 1                                          ' Relais AUF einschalten
Return
```

```
'-- End Relais_auf_an ------------------------------------------------------


'--------------------------------------------------------------------------------
' REL - Gosub-Routine: Relais_auf_aus
' Subroutine zum Ausschalten des Relais AUF
'--------------------------------------------------------------------------------
Relais_auf_aus:

   #if Relais_testmodus
      Print "Enter Relais_auf_aus"
   #endif

   Rel_auf_on = 0                                       ' Relais AUF ausschalten
Return
'-- End Relais_auf_aus ------------------------------------------------------


'--------------------------------------------------------------------------------
' REL - Gosub-Routine: Relais_zu_an
' Subroutine zum Eimschalten des Relais ZU
'--------------------------------------------------------------------------------
Relais_zu_an:

   #if Relais_testmodus
      Print "Enter Relais_zu_an"                       ' Relais ZU einschalten
   #endif

   Rel_zu_on = 1
Return
'-- End Relais_zu_an --------------------------------------------------------


'--------------------------------------------------------------------------------
' REL - Gosub-Routine: Relais_zu_aus
' Subroutine zum Ausschalten des Relais ZU
'--------------------------------------------------------------------------------
Relais_zu_aus:

   #if Relais_testmodus
      Print "Enter Relais_zu_aus"
   #endif

   Rel_zu_on = 0                                        ' Relais ZU ausschalten
Return
'-- End Relais_zu_aus -------------------------------------------------------


'--------------------------------------------------------------------------------
' REL - Gosub-Routine: Relais_licht_an
' Subroutine zum Eimschalten des Relais LICHT
'--------------------------------------------------------------------------------
Relais_licht_an:

   #if Relais_testmodus
      Print "Enter Relais_licht_an"
   #endif

   Rel_licht_on = 1                                     ' Relais LICHT einschalten
Return
'-- End Relais_licht_an -----------------------------------------------------


'--------------------------------------------------------------------------------
' REL - Gosub-Routine: Relais_licht_aus
' Subroutine zum Ausschalten des Relais LICHT
'--------------------------------------------------------------------------------
Relais_licht_aus:

   #if Relais_testmodus
      Print "Enter Relais_licht_aus"
   #endif

   Rel_licht_on = 0                                     ' Relais LICHT ausschalten
Return
'-- End Relais_licht_aus -----------------------------------------------------


'--------------------------------------------------------------------------------
' REL - Gosub-Routine: Rel_beschattung_auf
```

```
' Subroutine zum kompletten Auffahren der Beschattung mit Richtungsumschaltung
'-------------------------------------------------------------------------------
Relais_beschattung_auf:

   #if Relais_testmodus
      Print "Enter Relais_beschattung_auf"
   #endif

   ' Prüfung ob Antrieb sich aktuell in ZU-Richtung bewegt
   If Beschattung_zu_flag = Aktiv Then
      ' Wenn JA Antrieb abschalten und Flag zurücksetzen
      Beschattung_zu_flag = Passiv
      Gosub Relais_zu_aus
      Waitms 500                                        ' Zum Schutz des Antriebs 500 ms warten
   End If

   ' Antriebe AUF einschalten und Timerüberwachung aktivieren
   Gosub Relais_auf_an
   Beschattung_auf_flag = Aktiv
   Beschattung_auf_counter = Timercounter + Beschattung_delay
Return
'-- End Relais_beschattung_auf ------------------------------------------------


'-------------------------------------------------------------------------------
' REL - Gosub-Routine: Relais_beschattung_zu
' Subroutine zum kompletten Zufahren der Beschattung mit Richtungsumschaltung
'-------------------------------------------------------------------------------
Relais_beschattung_zu:

   #if Relais_testmodus
      Print "Enter Relais_beschattung_zu"
   #endif

   ' Prüfung ob Antrieb sich aktuell in AUF-Richtung bewegt
   If Beschattung_auf_flag = Aktiv Then
      ' Wenn JA Antrieb abschalten und Flag zurücksetzen
      Beschattung_auf_flag = Passiv
      Gosub Relais_auf_aus
      Waitms 500                                        ' Zum Schutz des Antriebs 500 ms warten
   End If

   ' Antriebe ZU einschalten und Timerüberwachung aktivieren
   Gosub Relais_zu_an
   Beschattung_zu_flag = Aktiv
   Beschattung_zu_counter = Timercounter + Beschattung_delay
Return
'-- End Relais_beschattung_zu -------------------------------------------------


'-------------------------------------------------------------------------------
' REL - Gosub-Routine: Relais_beschattung_stopp
' Subroutine zum Anhalten des aktuell Fahrvorgangs
'-------------------------------------------------------------------------------
Relais_beschattung_stop:

   #if Relais_testmodus
      Print "Enter Relais_beschattung_stop"
   #endif

   ' Antrieb ZU anhalten und Timerüberwachung zurücksetzen
   Beschattung_zu_flag = Passiv
   Gosub Relais_zu_aus

   ' Antrieb AUF anhalten und Timerüberwachung zurücksetzen
   Beschattung_auf_flag = Passiv
   Gosub Relais_auf_aus

Return
'-- End Relais_beschattung_stopp -------------------------------------------


'-------------------------------------------------------------------------------
' Gosub-Routine: Beleuchtung_ein_aus
' Subroutine schaltet Beleuchtung ein oder aus (Toggelbetrieb)
'-------------------------------------------------------------------------------
Beleuchtung_ein_aus:

   #if Relais_testmodus
      Print "Enter Beleuchtung_ein_aus"
```

```
   #endif

   ' Prüfung ob Beleuchtung aus
   If Beleuchtung_flag = Passiv Then
      ' JA, Beleuchtung ist aus und wird eingeschaltet
      Beleuchtung_flag = Aktiv
      Gosub Relais_licht_an
   Else
      ' NEIN, Beleuchtung ist ein und wird ausgeschaltet
      Beleuchtung_flag = Passiv
      Gosub Relais_licht_aus
   End If
Return
'-- End Beleuchtung_ein_aus ------------------------------------------------


'--------------------------------------------------------------------------------
' Gosub-Routine: Sw_timer_sequencer
' Subroutine Abprüfen der in RUNNING befindlichen SW-Timer für Beschattung auf
' oder zu
'--------------------------------------------------------------------------------
Sw_timer_sequencer:

   ' Überprüfung ob Timerüberwachung für Beschattung ZU aktiv
   If Beschattung_zu_flag = Aktiv Then

      ' Prüfung ob Timer abgelaufen
      If Timercounter = Beschattung_zu_counter Then

         #if Sequencer_testmodus
            Print "Automatische Abschaltung ZU"
         #endif

         ' Wenn ja Beschattung ZU abschalten und Timerüberwachung deaktivieren
         Beschattung_zu_flag = Passiv
         Gosub Relais_zu_aus
      End If
   End If

   ' Überprüfung ob Timerüberwachung für Beschattung AUF aktiv
   If Beschattung_auf_flag = Aktiv Then

      ' Prüfung ob Timer abgelaufen
      If Timercounter = Beschattung_auf_counter Then

         #if Sequencer_testmodus
            Print "Automatische Abschaltung AUF"
         #endif

         ' Wenn ja Beschattung AUF abschalten und Timerüberwachung deaktivieren
         Beschattung_auf_flag = Passiv
         Gosub Relais_auf_aus
      End If
   End If

Return
'-- End Sw_timer_sequencer --------------------------------------------------


'--------------------------------------------------------------------------------
' DIM - Gosub-Routine: Dimmer_konfig
' Subroutine zur Konfiguration des Dimemrs
'--------------------------------------------------------------------------------
Dimmer_konfig:

   #if Dimmer_testmodus
      Print "Enter Dimmer_konfig"
   #endif

   ' PortE.5 als Eingang mit aktivem PullUp konfigurieren
   Config Null_detect = Input
   Null_detect_pin = 1

   ' PortD.4 als Ausgang definieren und Triac ausschalten
   Config Ignition = Output
   Triac_start = 0


   ' Der Interrupt selbst kann nicht in der Gosub-Routine initialisiert werden
   ' sondern muss im Rumpf des Gesamtprogramms initialisiert werden.
```

```
   ' Dimmerhelligkeit auf Default einstellen
   Dim_helligkeitsstufe = Dim_helligkeit_default

   ' Triac Zündzeitpunkte aus Tabelle in Datenstruktur laden
   Restore Triac_zuendtabelle
   For Dim_index_counter = 1 To Dim_helligkeitstabellenwerte
      Read Triac_zuenszeitpunkte(dim_index_counter)
      #if Dimmer_testmodus
         Print "Zuendwert " ; Dim_index_counter ; " : " ; Triac_zuenszeitpunkte(dim_index_counter)
      #endif
   Next Dim_index_counter

Return
'-- End Dimmer_konfig ---------------------------------------------------------


'-------------------------------------------------------------------------------
' DIM - Gosub-Routine: Dimmer_heller
' Subroutine zum Erhöhen der Dimmerhelligkeit, Verschiebung Index Zündzeitpunkt
'-------------------------------------------------------------------------------
Dimmer_heller:

   #if Dimmer_testmodus
      Print "Enter Dimmer_heller"
   #endif

   If Dim_helligkeitsstufe < Dim_helligkeit_max Then
      Incr Dim_helligkeitsstufe
   End If

   #if Dimmer_testmodus
      Print "Helligkeit ist : " ; Dim_helligkeitsstufe
   #endif

Return
'-- End Dimmer_heller ----------------------------------------------------------


'-------------------------------------------------------------------------------
' DIM - Gosub-Routine: Dimmer_dunkler
' Subroutine zum reduzieren der Dimmerhelligkeit, Verschiebung Index
' Zündzeitpunkt
'-------------------------------------------------------------------------------
Dimmer_dunkler:

   #if Dimmer_testmodus
      Print "Enter Dimmer_dunkler"
   #endif

   If Dim_helligkeitsstufe > Dim_helligkeit_min Then
      Decr Dim_helligkeitsstufe
   End If

   #if Dimmer_testmodus
      Print "Helligkeit ist : " ; Dim_helligkeitsstufe
   #endif

Return
'-- End Dimmer_dunkler ---------------------------------------------------------


'-------------------------------------------------------------------------------
' Gosub-Routine: Dimmer_free_running
' Subroutine zum Setzen von Parametern für freien Dimmerlauf, gesteuert über
' Nulldurchgangskennung
'-------------------------------------------------------------------------------
Dimmer_free_running:

   #if Dimmer_testmodus
      Print "Enter Dimmer_free_running"
   #endif

   ' Prüfung ob free_running aktiv
   If Free_running_flag = Passiv Then
      ' JA, Free_runing ist aus und wird eingeschaltet
      Free_running_flag = Aktiv

   #if Dimmer_testmodus
      Print "Free Running ist aktiv"
```

```
   #endif

      Led6 = Led_an
   Else
      ' NEIN, Free_running ist ein und wird ausgeschaltet
      Free_running_flag = Passiv

   #if Dimmer_testmodus
      Print "Free Running ist passiv"
   #endif

      Led6 = Led_aus
   End If


Return
'-- End Dimmer_free_running ------------------------------------------------


'-------------------------------------------------------------------------------
' Gosub-Routine: Dimmer_one_shot
' Subroutine zum einmaligen Auslösen des Timer0 für TRIAC-Zündung
'-------------------------------------------------------------------------------
Dimmer_one_shot:

   #if Dimmer_testmodus
      Print "Enter Dimmer_one_shot"
   #endif

   Led0 = Led_an
   Waitus 5
   Led0 = Led_aus
   Start Timer2
   Timer2 = Triac_zuenszeitpunkte(dim_helligkeitsstufe)


Return
'-- End Dimmer_one_shot ---------------------------------------------------


End                                                    'end program


'-------------------------------------------------------------------------------
' Definition von globalen Konstantenfeldern
'-------------------------------------------------------------------------------

Triac_zuendtabelle:
' Tabelle mit Originalwerten
' Data 156 , 139 , 131 , 125 , 120 , 116 , 111 , 108
' Data 103 , 100 , 95 , 92 , 89 , 86 , 83 , 80
' Data 77 , 73 , 70 , 67 , 64 , 61 , 56 , 53,
' Data 50 , 45 , 41 , 36 , 31 , 25 , 17 , 0
' Für Timer0 werden Einerkompliment-Werte (255-Wert) benötigt
Data 99 , 116 , 124 , 130 , 135 , 139 , 144 , 147
Data 152 , 155 , 160 , 163 , 166 , 169 , 172 , 175
Data 178 , 182 , 185 , 188 , 191 , 194 , 199 , 202
Data 205 , 210 , 214 , 219 , 224 , 230 , 238 , 250


'#################################### END ####################################
```

Software 2: Code Zur Ansteuerung der Relaisstufen und Dimmereinheit

Optimierungsmöglichkeiten:

Bei einer Interrupt-Service-Routine in BASCOM werden alle Register am Anfang via PUSH und am Ende via POP auf den Stack gerettet bzw. vom Stack restauriert. Um Performance in in den Ablauf zu bekommen besteht die Möglichkeit mit NOSAVE den Itterupt aufzurufen. Dann muss aber das Retten von Registern von Hand erfolgen.

Die im folgenden abgebildete Code-Sequenz zeit die Interrupt-Service-Routine für die Zündung des TRIAC wie sie von BASCOM erzeugt wird. Hier besteht noch Optimierungspotenitial für PUSH und POP im Endprojekt.

```
+000001EA:   920F        PUSH    R0              Push register on stack
+000001EB:   921F        PUSH    R1              Push register on stack
+000001EC:   922F        PUSH    R2              Push register on stack
```

```
+000001ED:   923F        PUSH    R3              Push register on stack
+000001EE:   924F        PUSH    R4              Push register on stack
+000001EF:   925F        PUSH    R5              Push register on stack
+000001F0:   927F        PUSH    R7              Push register on stack
+000001F1:   92AF        PUSH    R10             Push register on stack
+000001F2:   92BF        PUSH    R11             Push register on stack
+000001F3:   930F        PUSH    R16             Push register on stack
+000001F4:   931F        PUSH    R17             Push register on stack
+000001F5:   932F        PUSH    R18             Push register on stack
+000001F6:   933F        PUSH    R19             Push register on stack
+000001F7:   934F        PUSH    R20             Push register on stack
+000001F8:   935F        PUSH    R21             Push register on stack
+000001F9:   936F        PUSH    R22             Push register on stack
+000001FA:   937F        PUSH    R23             Push register on stack
+000001FB:   938F        PUSH    R24             Push register on stack
+000001FC:   939F        PUSH    R25             Push register on stack
+000001FD:   93AF        PUSH    R26             Push register on stack
+000001FE:   93BF        PUSH    R27             Push register on stack
+000001FF:   93CF        PUSH    R28             Push register on stack
+00000200:   93DF        PUSH    R29             Push register on stack
+00000201:   93EF        PUSH    R30             Push register on stack
+00000202:   93FF        PUSH    R31             Push register on stack
+00000203:   B78F        IN      R24,0x3F        In from I/O location
+00000204:   938F        PUSH    R24             Push register on stack
+00000205:   2788        CLR     R24             Clear Register
+00000206:   BD85        OUT     0x25,R24        Out to I/O location
+00000207:   9A94        SBI     0x12,4          Set bit in I/O register
+00000208:   93EF        PUSH    R30             Push register on stack
+00000209:   93FF        PUSH    R31             Push register on stack
+0000020A:   E4EC        LDI     R30,0x4C        Load immediate
+0000020B:   E0F0        LDI     R31,0x00        Load immediate
+0000020C:   940E037A    CALL    0x0000037A      Call subroutine
+0000020E:   91FF        POP     R31             Pop register from stack
+0000020F:   91EF        POP     R30             Pop register from stack
+00000210:   9894        CBI     0x12,4          Clear bit in I/O register
+00000211:   918F        POP     R24             Pop register from stack
+00000212:   BF8F        OUT     0x3F,R24        Out to I/O location
+00000213:   91FF        POP     R31             Pop register from stack
+00000214:   91EF        POP     R30             Pop register from stack
+00000215:   91DF        POP     R29             Pop register from stack
+00000216:   91CF        POP     R28             Pop register from stack
+00000217:   91BF        POP     R27             Pop register from stack
+00000218:   91AF        POP     R26             Pop register from stack
+00000219:   919F        POP     R25             Pop register from stack
+0000021A:   918F        POP     R24             Pop register from stack
+0000021B:   917F        POP     R23             Pop register from stack
+0000021C:   916F        POP     R22             Pop register from stack
+0000021D:   915F        POP     R21             Pop register from stack
+0000021E:   914F        POP     R20             Pop register from stack
+0000021F:   913F        POP     R19             Pop register from stack
+00000220:   912F        POP     R18             Pop register from stack
+00000221:   911F        POP     R17             Pop register from stack
+00000222:   910F        POP     R16             Pop register from stack
+00000223:   90BF        POP     R11             Pop register from stack
+00000224:   90AF        POP     R10             Pop register from stack
+00000225:   907F        POP     R7              Pop register from stack
+00000226:   905F        POP     R5              Pop register from stack
+00000227:   904F        POP     R4              Pop register from stack
+00000228:   903F        POP     R3              Pop register from stack
+00000229:   902F        POP     R2              Pop register from stack
+0000022A:   901F        POP     R1              Pop register from stack
+0000022B:   900F        POP     R0              Pop register from stack
+0000022C:   9518        RETI                    Interrupt return
```

Software 3: Code für TRIAC Interrupt-Service-Routine aus BASCOM

8.8.9 Prototyp Relaisansteuerung und Dimmer

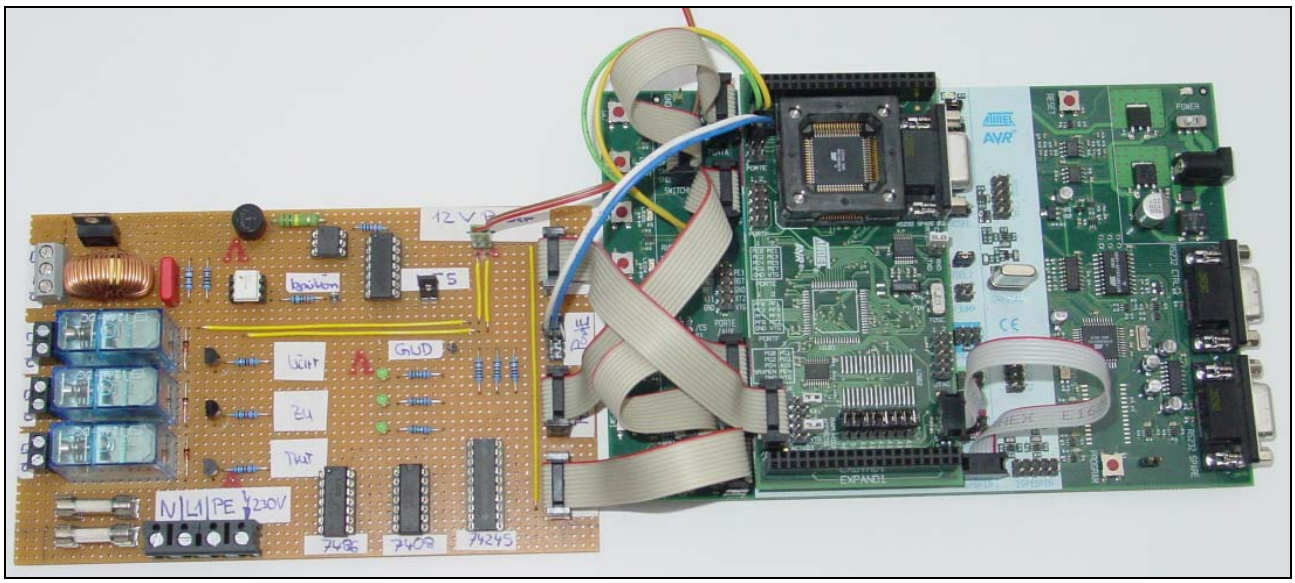


Abbildung 58: Prototyp Relaisansteuerung und Dimmer mit STK

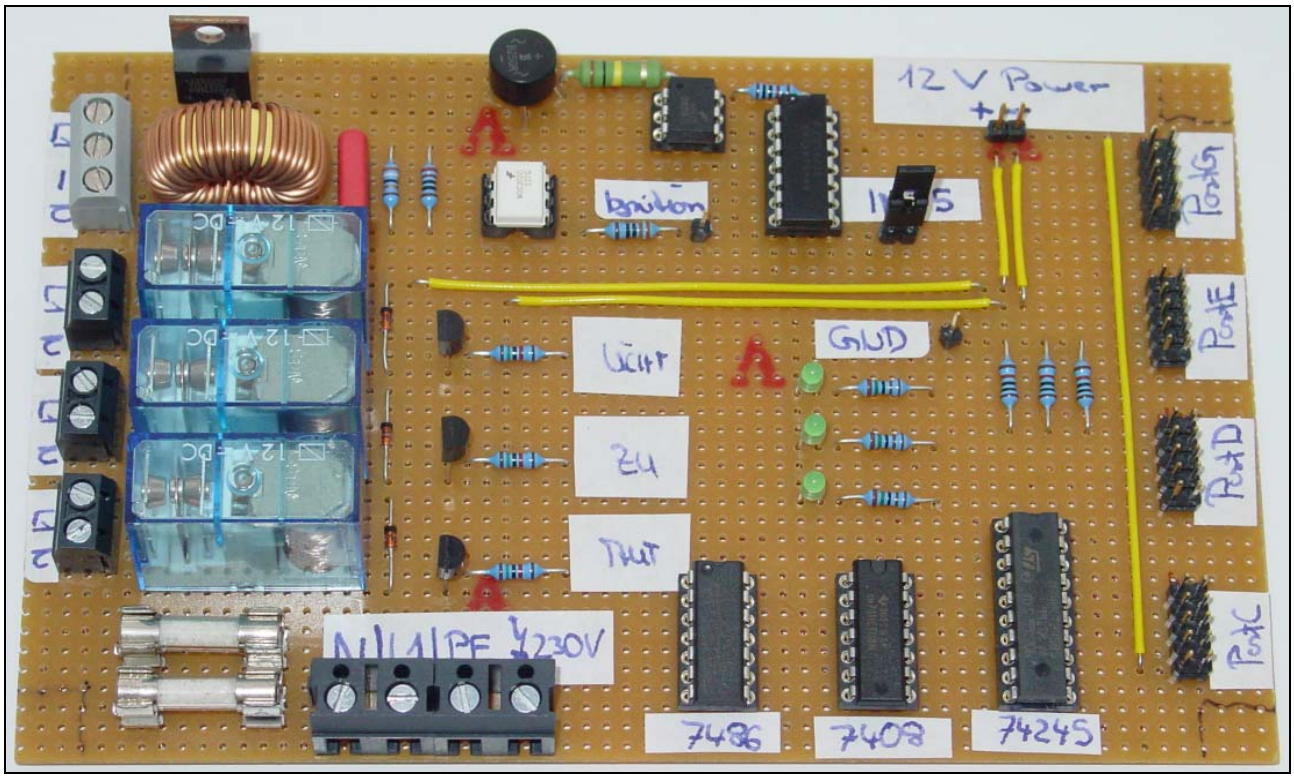


Abbildung 59: Prototyp Relaisansteuerung und Dimmer PCB

8.9 LCD-Display

Bemerkung:

Am 17.01.2008 wurde die Entscheidung getroffen die PWM für die LCD-Displayhelligkeit auf Timer3 und OC3A zu verlegen. Ursache dafür ist eine neue Verteilung der verfügbaren Ressourcen.

Die in diesem Kapitel enthaltenen Beschreibungen und Beispielcodes basieren auf der ursprünglichen Annahme die Umsetzung mit Timer0 zu realisieren. Weitere Beschreibungen und Code-Beispiele zu Timer3 befinden sich im Kapitel zur Lüfteransteuerung beschrieben. In diesem Kapitel wird der komplette Timer3 in Betrieb genommen und die HW auf Timer3 (PortE.3) umgestellt.

Allgemeines:

Im Projekt ConvCtrl soll ein LCD-Display der Firma electronic assembly der folgenden Typen eingesetzt werden:

- LCD-MODUL 3x16 - 3,65mm EA DOGM163W-A

Für das LCD-Display wird eine weiße LED-Hintergrundbeleuchtung eingesetzt.



Abbildung 60: LCD-Display EA DOG-M

Technische Daten:

- Kontrastreiche LCD-Supertwist Anzeige
- Optionale LED-Beleuchtungskörper in verschiedenen Farben
- 1x8, 2x16 und 3x16 Zeichen mit 12,0 mm / 5,6 mm und 3,6 mm Schrift
- Kontroller ST 7036 für 4-BIT, 8-BIT und SPI (4-DRAHT) Interface
- Spannungsversorgung +3,3V oder +5V single supply (typ. 250µA)
- Keine zus. Spannungen erforderlich
- Betriebstemperaturbereich -20..+70°C
- LED-Hintergrundbeleuchtung 3..80mA@3,3V oder 2..40mA@5V
- Keine Montage erforderlich: einfach nur in PCB einlöten.

Abmessungen:

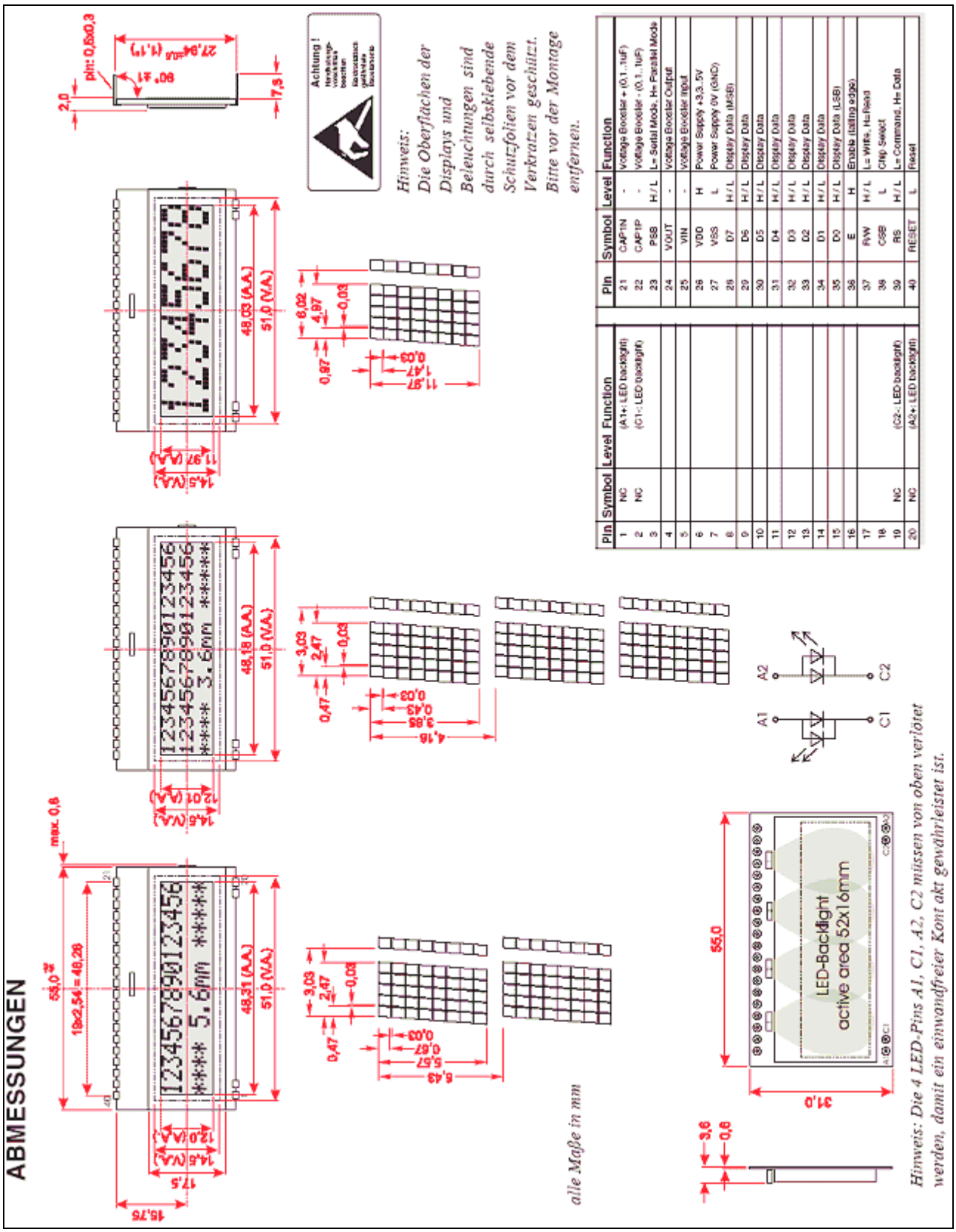


Abbildung 61: Abmessungen und technische Daten DOGM LCD-Display

Kontrasteinstellung:

Für alle Displays der EA DOG- Serie ist der Kontrast per Befehl einstellbar. Dies erfolgt über die Bits C0..C5 in den Befehlen "Contrast Set" und "Power/Icon Control/Contrast Set". In der Regel wird der Kontrast einmalig eingestellt und dann - dank integrierter Temperaturkompensation - über den gesamten Betriebstemperaturbereich (-20..+70°C) konstant gehalten.

Insgesamt benötigen die Displays selbst im 3,3V Betrieb keine zusätzliche negative Spannung!

LED-Hintergrundbeleuchtung für LCD-Display:

white EA LED55x31-W	Forward voltage	Current max.	Limiting resistor	
			@ 3,3 V	@ 5 V
Connected in parallel	3,2 V	60 mA	1,6 ohm	30 ohm
Connected in series	6,4 V	30 mA	-	-

Für die Beschaltung der Hintergrundbeleuchtung wird ein Vorwiderstand von 30 Ohm benötigt. Eine Serienschaltung der LED's ist nicht möglich weil hierfür eine Forward-Spannung von 6,4V benötigt wird, das Display aber nur mit 5V betrieben wird. Somit kommt nur Parallelschaltung in Frage!

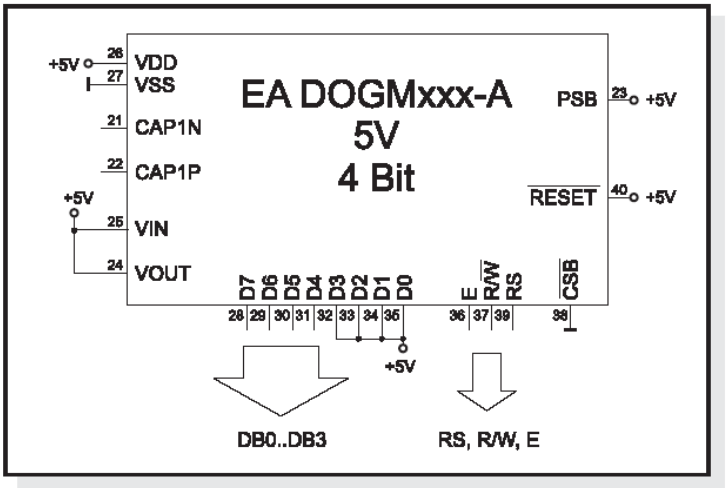
Tabelle 81: LED-Hintergrundbeleuchtung für LCD-Display

Für die Hintergrundbeleuchtung wird eine PWM über den ATmega realisiert mit Hilfe derer die Beleuchtungshelligkeit eingestellt werden kann. Siehe hierzu Beschaltung des LCD-Displays und separate Transistor-Stufe.

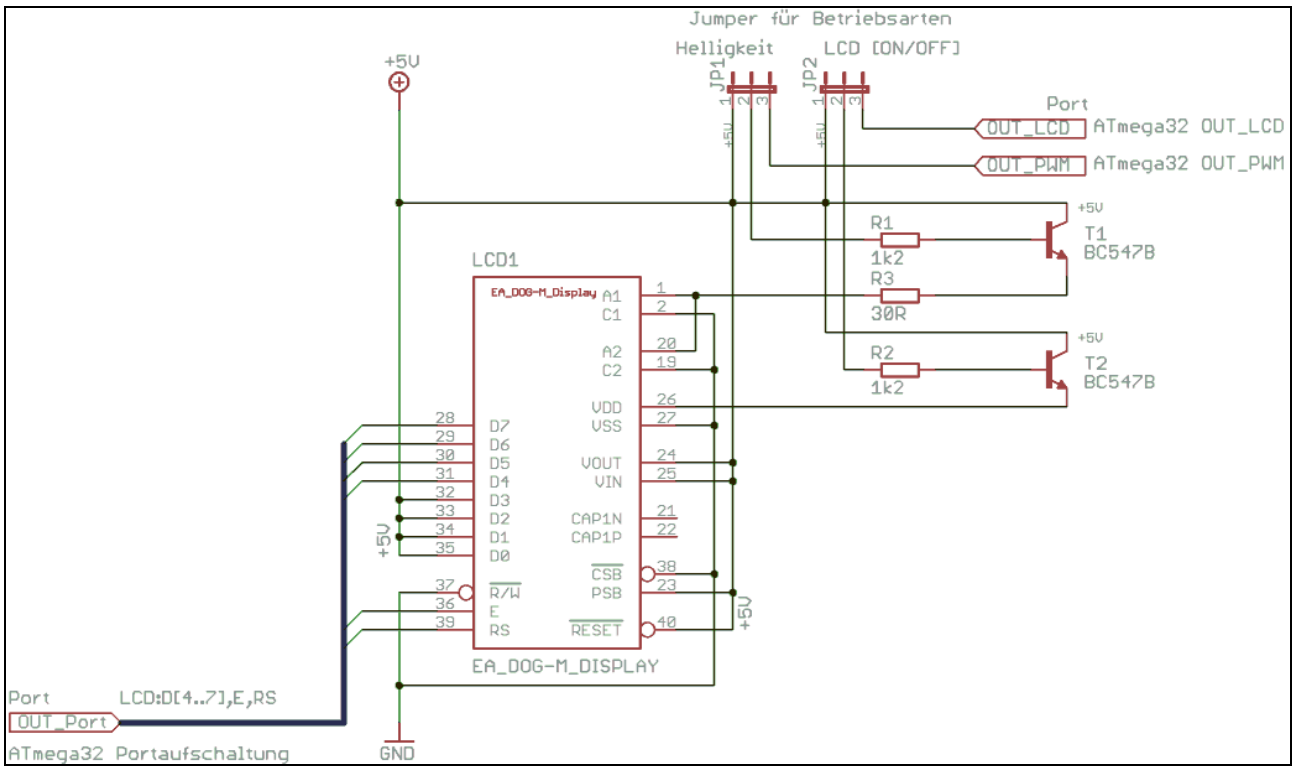
Das LCD-Display muss vor dem verlöten oder stecken auf die Beleuchtungseinheit gelötet werden, damit die LEDs der Beleuchtung über die Kontakte des LCDs Strom bekommen. Es ist ratsam, alle Pins der Beleuchtungseinheit anzulöten, da sich so der Druck beim Einsetzten des Displays in die Fassung besser verteilt. Elektrisch ist dies nicht notwendig. Bitte sehr sparsam mit dem Lötzinn umgehen, da es sonst an den Beinchen herunter läuft und somit das Display nicht in die Fassung passt.

8.9.1 Beschaltung des LCD-Display

Die Beschaltung des LCD-Display an den ATmega erfolgt wie im folgenden Schaltbild aus dem Datenblatt des LCD-Displays gefordert:



Schaltbild 28: Beschaltung LCD-Display gemäß Datenblatt



Schaltbild 29: Beschaltung LCD-Display im Projekt ConvCtrl

Bemerkung:

Änderung der PWM von Timer0 auf Timer3 und damit auf PortE.3!

Bauteile:

Stückliste: LCD Beschaltung			
Widerstände		*Halbleiter*	
R1, R2	1k2 Ω	LCD1	LCD Display EA DOGM163W
R3	30 Ω	T1, T2	Transistor BC547B
Sonstiges			
JP1, JP2	Jumper 3-polig, 2,54mm Raster		

Tabelle 82: Stückliste LCD Beschaltung

Bemerkung zur LCD-Beschaltung:

- Über den Jumper JP1 kann eingestellt werden, ob die Beleuchtung mit konstanter Spannung +5V und gleichbleibender Helligkeit betrieben wird oder ob eine Ankopplung an den ATmega zur PWM erfolgt.
- Über den Jumper JP2 kann eingestellt werden, ob das LCD-Display konstant eingeschaltet ist sobald die Betriebsspannung eingeschaltet ist oder ob das LCD-Display über eine Transistorstufe vom ATmega ein- und ausgeschaltet werden kann.

Ressourcenzuordnung zum ATmega128:

Nummer	*Schaltbild*	*Ressource ATmega128*
1	OUT_LCD [LCD_ON]	PortG.1 [PG1] (GPIO)
2	OUT_PWM [LCD_PWM]	PortB.4 [OC0] (Timer0) *Umstellung nach Test:* *PortE.3 [OC3A] (Timer3)*
3	Out_Port LCD:D4 [LCD_BIT0]	PortC.0 [PC0] (GPIO)

4	Out_Port LCD:D5	[LCD_BIT1]	PortC.1	[PC1]	(GPIO)
5	Out_Port LCD:D6	[LCD_BIT2]	PortC.2	[PC2]	(GPIO)
6	Out_Port LCD:D7	[LCD_BIT3]	PortC.3	[PC3]	(GPIO)
7	Out_Port LCD:E	[LCD_E]	PortC.4	[PC4]	(GPIO)
8	Out_Port LCD:RS	[LCD_RS]	PortC.5	[PC5]	(GPIO)

Tabelle 83: Ressourcenzuordnung ATmega128 für LCD-Display

8.9.2 Zeichensatz und Befehlstabellen

Der unten abgebildete Zeichensatz ist integriert. Zusätzlich können 8 eigene Zeichen frei definiert werden.



Tabelle 84: Zeichensatz des LCD-Displays

Befehlstabellen des Display:

Instruction	Instruction Code										Description	Instruction Execution Time		
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		OSC= 380kHz	OSC= 540kHz	OSC= 700kHz
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRAM. and set DDRAM address to "00H" from AC	1.08 ms	0.76 ms	0.59 ms
Return Home	0	0	0	0	0	0	0	0	1	x	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.08 ms	0.76 ms	0.59 ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.	26.3 µs	18.5 µs	14.3 µs
Display ON/OFF	0	0	0	0	0	0	1	D	C	B	D=1:entire display on C=1:cursor on B=1:cursor position on	26.3 µs	18.5 µs	14.3 µs
Function Set	0	0	0	0	1	DL	N	DH	IS2	IS1	DL: interface data is 8/4 bits N: number of line is 2/1 DH: double height font IS[2:1]: instruction table select	26.3 µs	18.5 µs	14.3 µs
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter	26.3 µs	18.5 µs	14.3 µs
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.	0	0	0
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM/ICONRAM)	26.3 µs	18.5 µs	14.3 µs
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM/ICONRAM)	26.3 µs	18.5 µs	14.3 µs

Tabelle 85: LCD-Display EA DOGM Instruction Code

Instruction table 0(IS[2:1]=[0,0])														
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	X	X	S/C and R/L: Set cursor moving and display shift control bit, and the direction, without changing DDRAM data.	26.3 µs	18.5 µs	14.3 µs
Set CGRAM	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter	26.3 µs	18.5 µs	14.3 µs

Tabelle 86: LCD-Display EA DOGM Instruction table 0

Instruction table 1(IS[2:1]=[0,1])														
Bias Set	0	0	0	0	0	1	BS	1	0	FX	BS=1:1/4 bias BS=0:1/5 bias FX: fixed on high in 3-line application and fixed on low in other applications.	26.3 µs	18.5 µs	14.3 µs
Set ICON Address	0	0	0	1	0	0	AC3	AC2	AC1	AC0	Set ICON address in address counter.	26.3 µs	18.5 µs	14.3 µs
Power/ICON Control/ Contrast Set	0	0	0	1	0	1	Ion	Bon	C5	C4	Ion: ICON display on/off Bon: set booster circuit on/off C5,C4: Contrast set for internal follower mode.	26.3 µs	18.5 µs	14.3 µs
Follower Control	0	0	0	1	1	0	Fon	Rab 2	Rab 1	Rab 0	Fon: set follower circuit on/off Rab2~0: select follower amplified ratio.	26.3 µs	18.5 µs	14.3 µs
Contrast Set	0	0	0	1	1	1	C3	C2	C1	C0	Contrast set for internal follower mode.	26.3 µs	18.5 µs	14.3 µs

Tabelle 87: LCD-Display EA DOGM Instruction table 1

Instruction table 2(IS[2:1]=[1,0])														
Double Height Position Select	0	0	0	0	0	1	UD	X	x	x	UD: Double height position select	26.3 µs	18.5 µs	14.3 µs
Reserved	0	0	0	1	X	X	X	X	X	X	Do not use (reserved for test)	26.3 µs	18.5 µs	14.3 µs

Tabelle 88: LCD-Display EA DOGM Instruction table 2

Eine detaillierte Beschreibung des hier integrierten Kontrollers ST7036 finden Sie im Internet unter
http://www.lcd-module.de/eng/pdf/zubehoer/st7036.pdf

Initialisierungsbeispiel für EA DOGM162 bei 8-Bit und 5V:

INITIALISIERUNGSBEISPIEL FÜR 8 Bit / 5V EA DOGM162												
Befehl	**RS**	**R/W**	**DB7**	**DB6**	**DB5**	**DB4**	**DB3**	**DB2**	**DB1**	**DB0**	**Hex**	**Bemerkung**
Function Set	0	0	0	0	1	1	1	0	0	1	$39	8-Bit Datenlänge, 2 Zeilen, Instruction table 1
Bias Set	0	0	0	0	0	1	1	1	0	0	$1C	BS: 1/4, 2-zeiliges LCD
Power Control	0	0	0	1	0	1	0	0	1	0	$52	Booster aus, Kontrast C5, C4setzen
Follower Control	0	0	0	1	1	0	1	0	0	1	$69	Spannungsfolger und Verstärkung setzen
Contrast Set	0	0	0	1	1	1	0	1	0	0	$74	Kontrast C3, C2, C1 setzen
Display ON/OFF	0	0	0	0	0	0	1	1	1	1	$0F	Display ein, Cursor ein, Cursor blinken
Clear Display	0	0	0	0	0	0	0	0	0	1	$01	Display löschen, Cursor Home
Entry Mode Set	0	0	0	0	0	0	0	1	1	0	$06	Cursor Auto-Increment

Tabelle 89: LCD-Display EA DOGM162 Initialisierungsbeispiel

Initialisierungsbeispiel für EA DOGM163 bei 8-Bit und 5V:

INITIALISIERUNGSBEISPIEL FÜR 8 Bit / 5V EA DOGM163												
Befehl	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Hex	Bemerkung
Function Set	0	0	0	0	1	1	1	0	0	1	$39	8-Bit Datenlänge, 2 Zeilen, Instruction table 1
Bias Set	0	0	0	0	0	1	1	1	0	1	$1D	BS: 1/4, 3-zeiliges LCD
Power Control	0	0	0	1	0	1	0	0	0	0	$50	Booster aus, Kontrast C5, C4setzen
Follower Control	0	0	0	1	1	0	1	1	0	0	$6C	Spannungsfolger und Verstärkung setzen
Contrast Set	0	0	0	1	1	1	1	1	0	0	$7C	Kontrast C3, C2, C1 setzen
Display ON/OFF	0	0	0	0	0	0	1	1	1	1	$0F	Display ein, Cursor ein, Cursor blinken
Clear Display	0	0	0	0	0	0	0	0	0	1	$01	Display löschen, Cursor Home
Entry Mode Set	0	0	0	0	0	0	0	1	1	0	$06	Cursor Auto-Increment

Tabelle 90: LCD-Display EA DOGM163 Initialisierungsbeispiel

8.9.3 BASCOM Beispielcode für die Displayansteuerung (Funktionsbibliothek)

```
'#################################################################################
' LCD-Display.BAS:                                                 Stand 11.12.2007
' ----------------                                                 (C) Markus Fulde
' Testprogramm zur Inbetriebnahme des Display DOGM163
' Die Inbetriebnahme erfolgt mit dem Eva-Board STK500 + STK501
' Die für die LCD-Display-Ansteuerung notwendigen Routinen sind
' als solche gekennzeichnet und bereits zur späteren Verwendung
' ausgelagert.
'#################################################################################


'--------------------------------------------------------------------------------
' Compilerinstruktionen und Compilerdirektiven
'--------------------------------------------------------------------------------
$regfile = "m128def.dat"
$crystal = 16000000
$hwstack = 32
$swstack = 32
$framesize = 40

$baud = 38400


'--------------------------------------------------------------------------------
' Definition von Variablen, Konstaten und Datentypen
'--------------------------------------------------------------------------------
' Konstanten
Const Lcd_testmodus = 1

' LCD: Konstanten für LCD-Helligkeit (Range 0-255 / 8 Bit)
Const Schrittweite_lcdhelligkeit = 5                    ' Schrittweite für Einzelschritte
Const Lcd_helligkeit_min = 0                            ' minimale Helligkeit
Const Lcd_helligkeit_max = 255                          ' maximale Helligkeit
Const Lcd_default_helligkeit = 60                       ' Default Helligkeit empirisch ermittelt

' LCD: Konstanten für LCD-Kontrast (Range 0-63 / 6 Bit)
Const Lcd_kontrast_min = 0                              ' minimaler kontrast
Const Lcd_kontrast_max = 63                             ' maximaler Kontrast
Const Lcd_kontrast_default = 12                         ' Default Kontrast empirisch ermittelt

' LCD: Variablen deklarieren und mit Defaultwert laden
Dim Pwmwert As Byte                                     ' Dimwert für LCD-Helligkeit
Pwmwert = Lcd_default_helligkeit

Dim Kontrastwert As Byte
Kontrastwert = Lcd_kontrast_default                     ' Kontrastwert

' LCD: Temporäre Arbeitsvariablen definieren
Dim Tempvar_1 As Byte                                   ' Temporäre Variable 1
Dim Tempvar_2 As Byte                                   ' Temporäre Variable 2
```

```
' Sonstige Variablen und Konsten für Demo-Programm
Dim Text(6) As String * 16                                  ' Texte für programmgesteuerte Demo-
Ausgaben
Text(1) = "1234567890123456"
Text(2) = "ABCDEFGHIJKLMNOP"
Text(3) = "   Hallo Welt   "
Text(4) = "Test  Test  Test"
Text(5) = "Menue ausw{132}hlen!"
Text(6) = "{}!§$%&/()=?+#-ß"

Dim Textstep As Byte                                        ' Zeiger für Demotext
Textstep = 1


'-------------------------------------------------------------------------------
' Prototyping
'-------------------------------------------------------------------------------
Declare Sub Lcd_writesingleline(line1 As Byte , Text1 As String)
Declare Sub Lcd_writedoubleline(line1 As Byte , Line2 As Byte , Text1 As String , Text2 As String)
Declare Sub Lcd_writetrippleline(text1 As String , Text2 As String , Text3 As String)


'-------------------------------------------------------------------------------
' Konfiguration und Basiseinstellungen
'-------------------------------------------------------------------------------
' Alias Namen für Tastaturabfragen
Lcd_pwm_up Alias Pina.0                                     ' heller
Lcd_pwm_down Alias Pina.1                                   ' dunkler
Lcd_pwm_aus Alias Pina.2                                    ' Beleuchtung aus
Lcd_pwm_ein Alias Pina.3                                    ' Beleuchtung ein
Lcd_disp_aus Alias Pina.4                                   ' Display aus
Lcd_disp_ein Alias Pina.5                                   ' Display an
Lcd_disp_text Alias Pina.7                                  ' Schalter für Texttoggel


' LCD: Alias für LCD-Steuerung
Lcd_on Alias Portg.1

' LCD: Routinen und Einstellungen
Gosub Lcd_konfig                                            ' LCD Konfigurieren

' LCD: Startwert für Dimmung festlegen und Dimmung starten
Gosub Lcd_beleuchtung_an


'Konfiguration des Port A als Eingang für Tastaturabfrage
Config Porta = Input

'Konfiguration des Port F als Ausgang für Statusanzeige via LED
Config Portd = Output
Portd = &B11111111


'###############################################################################
' Hauptprogramm
'###############################################################################
'Hier ist die Programmhauptschleife
Do
   ' Tastaturabfrage Taste LCD heller
   If Lcd_pwm_up = 0 Then
      Portd.0 = 0
      Gosub Lcd_beleuchtung_heller
   Else
      Portd.0 = 1
   End If

   ' Tastaturabfrage Tase LCD dunkler
   If Lcd_pwm_down = 0 Then
      Portd.1 = 0
      Gosub Lcd_beleuchtung_dunkler
   Else
      Portd.1 = 1
   End If

   ' Tastaturabfrage Taste Beleuchtung aus
   Debounce Lcd_pwm_aus , 0 , Lcd_beleuchtung_aus , Sub

   ' Tastaturabfrage Taste Beleuchtung ein
   Debounce Lcd_pwm_ein , 0 , Lcd_beleuchtung_an , Sub
```

```
   ' Tastaturabfrage Display aus
   ' Debounce Lcd_disp_aus , 0 , Lcd_display_aus , Sub

   ' Tastaturabfrage Display aus => umgeleitet auf Kontrast erhöhen
   If Lcd_disp_aus = 0 Then
      Portd.4 = 0
      Gosub Lcd_kontrast_up
   Else
      Portd.4 = 1
   End If

   ' Tastaturabfrage Display an
   ' Debounce Lcd_disp_ein , 0 , Lcd_display_an , Sub

   ' Tastaturabfrage Display an => umgeleitet auf Kontrast erniedrigen
   If Lcd_disp_ein = 0 Then
      Portd.5 = 0
      Gosub Lcd_kontrast_down
   Else
      Portd.5 = 1
   End If

   ' Tastaturabfrage für Text weitertoggeln
   Debounce Lcd_disp_text , 0 , Toggle_text , Sub

Loop
'## End Hauptprogramm ##########################################################


'*****************************************************************************
' Subroutinen
'*****************************************************************************


'-----------------------------------------------------------------------------
' LCD - Sub-Routine: Lcd_writesingleline
' Gibt einen einzeiligen Texte auf dem LCD-Display aus.
' Parameter: line1 = Zeile in der die Ausgabe erfolgen soll
'            text1 = Text der ausgegeben werden soll
'-----------------------------------------------------------------------------
Sub Lcd_writesingleline(line1 As Byte , Text1 As Sring)
   Cls                                                  ' Bildschirm löschen
   Locate Line1 , 1 : Lcd Text1                         ' Ausgabe Zeil 1
End Sub
'-- End Lcd_writesingleline -------------------------------------------------


'-----------------------------------------------------------------------------
' LCD - Sub-Routine: Lcd_writedoubleline
' Gibt einen zweizeiligen Texte auf dem LCD-Display aus.
' Die Reihenfolge und die Textposition kann frei gewählt werden
' Parameter: line1 = 1. Zeile in der die 1. Ausgabe erfolgen soll
'            line2 = 2. Zeile in der die 2. Ausgabe erfolgen soll
'            text1 = 1. Text der auf 1. Position ausgegeben werden soll
'            text2 = 2. Text der auf 2. Position ausgegeben werden soll
'-----------------------------------------------------------------------------
Sub Lcd_writedoubleline(line1 As Byte , Line2 As Byte , Text1 As String , Text2 As String)
   Cls                                                  ' Bildschirm löschen
   Locate Line1 , 1 : Lcd Text1                         ' Ausgabe Zeil 1
   Locate Line2 , 1 : Lcd Text2                         ' Ausgabe Zeile 2
End Sub
'-- End Lcd_writedoubleline -------------------------------------------------


'-----------------------------------------------------------------------------
' LCD - Sub-Routine: Lcd_writetrippleline
' Gibt einen dreizeiligen Texte auf dem LCD-Display aus.
' Parameter: text1 = 1. Text der in Zeile 1 ausgegeben werden soll
'            text2 = 2. Text der in Zeile 2 ausgegeben werden soll
'            text3 = 3. Text der in Teile 3 ausgegeben werden soll
'-----------------------------------------------------------------------------
Sub Lcd_writetrippleline(text1 As String , Text2 As String , Text3 As String)

   ' Variablen für Positionierung definieren und festlegen
   ' Hinweis: Interessanter Weise funktioniert Ausgabe nicht bei direkter Zuweisung!
   Dim Line1 As Byte , Line2 As Byte , Line3 As Byte
   Line1 = 1
   Line2 = 2
   Line3 = 3
```

```
   ' Ausgabe der Texte
   Cls                                                  'Bildschirm löschen
   Locate Line1 , 1 : Lcd Text1                         ' Ausgabe Zeile 1
   Locate Line2 , 1 : Lcd Text2                         ' Ausgabe Zeile 2
   Locate Line3 , 1 : Lcd Text3                         ' Ausgabe Zeile 3
End Sub
'-- End Lcd_writetrippleline ------------------------------------------------


'*******************************************************************************
' GOSubroutinen
'*******************************************************************************

'-------------------------------------------------------------------------------
' LCD - Gosub-Routine: Lcd_konfig
' Subroutine zur Konfiguration des LCD-Display inkl. der dafür
' notwendigen Timer.
'-------------------------------------------------------------------------------
Lcd_konfig:

#if Lcd_testmodus
   Print "Enter Lcd_konfig"
#endif

   '----- Konfiguration LCD-Helligkeit (PWM) -----
   'Variable für Helligkeit in Timerregister laden
   Ocr0 = Pwmwert

   'Konfiguration Timer 0 für Hardware-PWM an OC0 (B.4)
   Config Timer0 = Pwm , Prescale = 256 , Compare = Clear

   'Port B.4 ist Timer-Ausgang OC0
   Config Pinb.4 = Output

   '----- Konfiguration LCD [ein/aus] (Transistorstufe) -----
   'Port G.1 als Ausgang definieren und Display einschalten
   Lcd_on = 1
   Config Ping.1 = Output

   '----- Konfiguration LCD Display -----
   Config Lcdpin = Pin , Db4 = Portc.0 , Db5 = Portc.1 , Db6 = Portc.2 , Db7 = Portc.3 , E = Portc.4
, Rs = Portc.5
   Config Lcd = 16 * 3 , Chipset = Dogm163v5
   Initlcd
   Waitms 100
   Cursor Off Noblink
   Cls

   Gosub Lcd_kontrastset

Return
'-- End Lcd_konfig -----------------------------------------------------------


'-------------------------------------------------------------------------------
' LCD - Gosub-Routine: Lcd_beleuchtung_an
' Routine schaltet die Beleuchtung ein
'-------------------------------------------------------------------------------
Lcd_beleuchtung_an:

#if Lcd_testmodus
   Print "Enter Lcd_beleuchtung_an"
#endif

   'Start des PWM-Timers
   Enable Timer0
   Timer0 = 0
   Start Timer0
Return
'-- End Lcd_beleuchtung_an ----------------------------------------------------


'-------------------------------------------------------------------------------
' LCD - Gosub-Routine: Lcd_beleuchtung_aus
' Routine schaltet die Beleuchtung aus
'-------------------------------------------------------------------------------
Lcd_beleuchtung_aus:

#if Lcd_testmodus
   Print "Enter Lcd_beleuchtung_aus"
```

```
#endif

   'Stop des PWM-Timers
   Disable Timer0
   Stop Timer0
Return
'-- End Lcd_beleuchtung_aus ------------------------------------------------


'-------------------------------------------------------------------------------
' LCD - Gosub-Routine: Lcd_beleuchtung_heller
' Routine schaltet die Beleuchtung heller
'-------------------------------------------------------------------------------
Lcd_beleuchtung_heller:

#if Lcd_testmodus
   Print "Enter Lcd_beleuchtung_heller"
#endif

   If Pwmwert < Lcd_helligkeit_max Then
      Pwmwert = Pwmwert + Schrittweite_lcdhelligkeit
   End If

   Ocr0 = Pwmwert

#if Lcd_testmodus
   Print "Pwmwert: " ; Pwmwert
#endif

Return
'-- End Lcd_testmodus --------------------------------------------------------


'-------------------------------------------------------------------------------
' LCD - Gosub-Routine: Lcd_beleuchtung_dunkler
' Routine schaltet die Beleuchtung dunkler
'-------------------------------------------------------------------------------
Lcd_beleuchtung_dunkler:

#if Lcd_testmodus
   Print "Enter Lcd_beleuchtung_dunkler"
#endif

   If Pwmwert > Lcd_helligkeit_min Then
      Pwmwert = Pwmwert - Schrittweite_lcdhelligkeit
   End If

   Ocr0 = Pwmwert

#if Lcd_testmodus
   Print "Pwmwert: " ; Pwmwert
#endif

Return
'-- End Lcd_beleuchtung_dunkler ----------------------------------------------


'-------------------------------------------------------------------------------
' LCD - Gosub-Routine: Lcd_display_aus
' Routine schaltet die Spannungsversorgung für das Display aus
'-------------------------------------------------------------------------------
Lcd_display_aus:

#if Lcd_testmodus
   Print "Enter Lcd_display_aus"
#endif

   Lcd_on = 0

Return
'-- End Lcd_display_aus ------------------------------------------------------


'-------------------------------------------------------------------------------
' LCD - Gosub-Routine: Lcd_display_an
' Routine schaltet die Spannungsversorgung für das Display an
'-------------------------------------------------------------------------------
Lcd_display_an:

#if Lcd_testmodus
```

```
   Print "Enter Lcd_display_an"
#endif

   Lcd_on = 1

Return
'-- End Lcd_display_an ------------------------------------------------------


'--------------------------------------------------------------------------------
' LCD - Gosub-Routine: Lcd_kontrast_up
' Routine erhöht den Kontrast
'--------------------------------------------------------------------------------
Lcd_kontrast_up:

#if Lcd_testmodus
   Print "Enter Lcd_kontrast_up"
#endif

   If Kontrastwert > Lcd_kontrast_min Then
      Kontrastwert = Kontrastwert - 1
   End If

   Gosub Lcd_kontrastset
Return
'-- End Lcd_kontrast_up --------------------------------------------------------


'--------------------------------------------------------------------------------
' LCD - Gosub-Routine: Lcd_kontrast_down
' Routine veringert den Kontrast
'--------------------------------------------------------------------------------
Lcd_kontrast_down:

#if Lcd_testmodus
   Print "Enter Lcd_kontrast_down"
#endif

   If Kontrastwert < Lcd_kontrast_max Then
      Kontrastwert = Kontrastwert + 1
   End If

   Gosub Lcd_kontrastset
Return
'-- End Lcd_kontrast_down ------------------------------------------------------


'--------------------------------------------------------------------------------
' LCD - Gosub-Routine: Lcd_contrastset
' Routine berechnen neue Kontrastwerte und steuert direkt den
' Kontroller des Display an.
'--------------------------------------------------------------------------------
Lcd_kontrastset:                                        ' Kontrasteinstellung Display

#if Lcd_testmodus
   Print "Enter Lcd_kontrastset" ; Kontrastwert
#endif

   ' Verarbeitung des Kontrastwertes für High-Byte und Low-Byte
   Tempvar_1 = Kontrastwert And &B00001111
   Tempvar_1 = Tempvar_1 + &B01110000

   Tempvar_2 = Kontrastwert
   Shift Tempvar_2 , Right , 4
   Tempvar_2 = Tempvar_2 And &B00000011
   Tempvar_2 = Tempvar_2 + &B01010100

   ' Instruction Table 1 einstellen [0,1]
   _temp1 = &B00101001
   !rCall _Lcd_control

   ' Tempvar_1 = &B0111xxxx für Kontrast Set Instruction Table 1 - Low Byte
   _temp1 = Tempvar_1
   !rCall _Lcd_control

   ' Temovar_2 = &B010101xx für Kontrast Set Instruction Table 1 - High Byte
   _temp1 = Tempvar_2
   !rCall _Lcd_control

   ' Zurückschalten auf Instruction Table 0 [0,0]
```

```
   _temp1 = &B00101000
   !rCall _Lcd_control
Return
'-- End Lcd_kontrastset ------------------------------------------------------


'--------------------------------------------------------------------------------
' Gosub-Routine: Toggle_text
' Routine gibt je nach Zustand unterschiedliche Texte aus
'--------------------------------------------------------------------------------
Toggle_text:

#if Lcd_testmodus
   Print "Enter Toggle Text" ; Textstep
#endif

   Dim A As Byte , B As Byte

   Select Case Textstep

      Case 1:
         A = 1
         Call Lcd_writesingleline(a , Text(1))

      Case 2:
         A = 2
         Call Lcd_writesingleline(a , Text(1))

      Case 3:
         A = 3
         Call Lcd_writesingleline(a , Text(1))

      Case 4:
         A = 1
         B = 3
         Call Lcd_writedoubleline(a , B , Text(1) , Text(2))

      Case 5:
         A = 2
         Call Lcd_writesingleline(a , Text(3))

      Case 6:
         Call Lcd_writetrippleline(text(4) , Text(5) , Text(6))

      Case 7:
         Cls
         Locate 1 , 1
         Lcd Text(1)
         Locate 2 , 1
         Lcd Text(2)
         Locate 3 , 1
         Lcd Text(3)

      Case 8:
         Cls
         Locate 1 , 1 : Lcd "*"
         Locate 1 , 16 : Lcd "*"
         Locate 2 , 8 : Lcd "**"
         Locate 3 , 1 : Lcd "*"
         Locate 3 , 16 : Lcd "*"

   End Select

   ' Prüfung ob Textstep am Ende, wenn ja zurück auf Position 1
   Textstep = Textstep + 1
   If Textstep > 8 Then
      Textstep = 1
   End If

Return
'-- End Toggle_text -------------------------------------------------------------

'#####################################  END  ###################################
```

Software 4: Code zur Ansteuerung des LCD-Displays

8.9.4 Prototyp LCD-Display-Ansteuerung

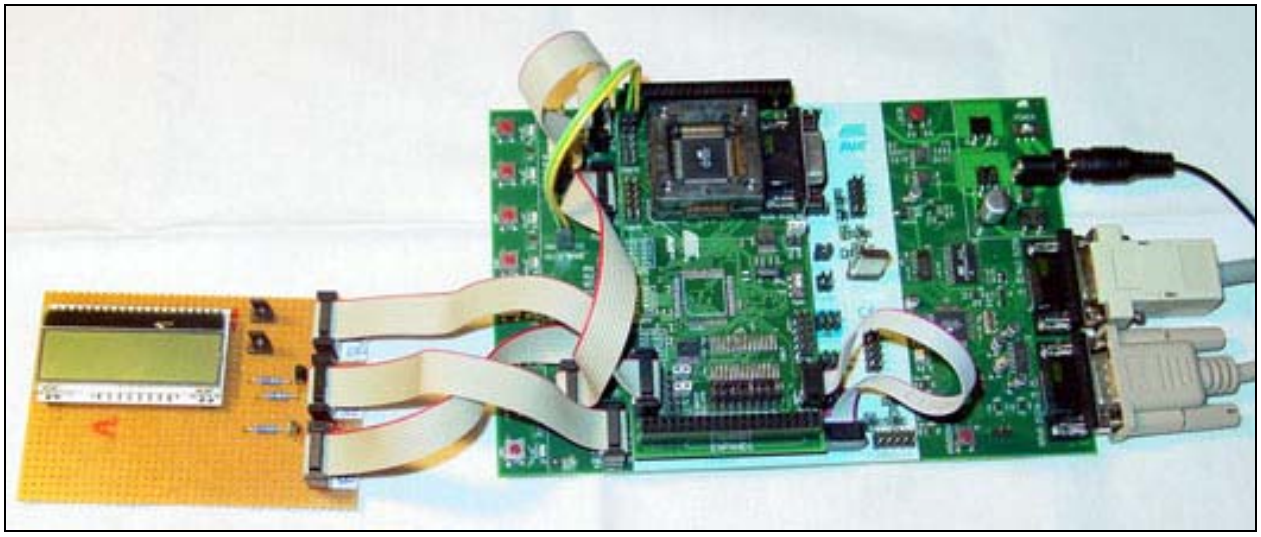


Abbildung 62: Prototyp LCD-Display-Ansteuerung mit STK

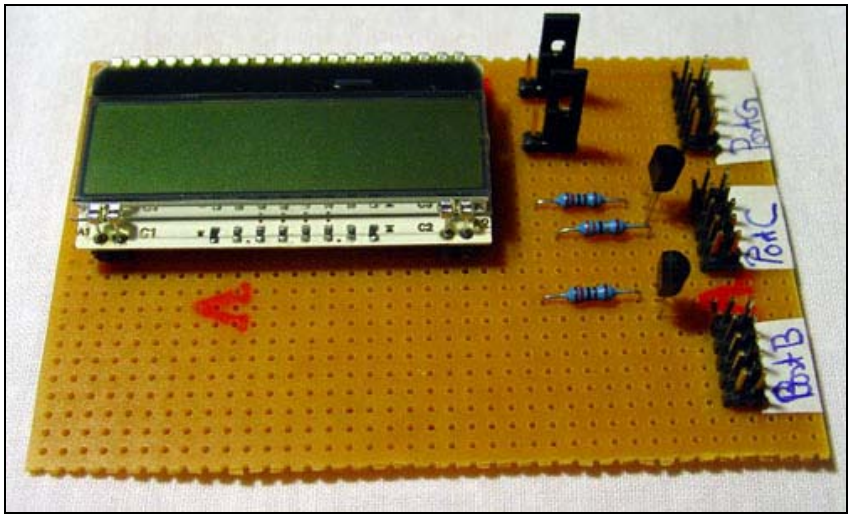


Abbildung 63: Prototyp LCD-Display-Ansteuerung PCB

8.10 EEPROM (intern und extern)

Für die Speicherung von Messwerten mit Zeit und Datumsinformationen wird ein externes EEPROM an den ATmega via I^2C-Bus angekoppelt. Die Setting-Daten und Einstellungen werden im 4k großen inernen EEPROM-Speicher des ATmega128 abgelegt. Das Kapitel beschreibt primär die Ansteuerung des externen EEPROM, geht aber auch auf die Datenstrukturen und Zugriffsmechanismen des internen EEPROM ein.

Für die Umsetzung kommen die drei EEPROM-Typen der Firma ST in Frage:

- ST24C128 Bestellnummer Firma Reichelt: ST 24C128 BN6
- ST24C256 Bestellnummer Firma Reichelt: ST 24C256 BN6
- *ST24C512* *Bestellnummer Firma Reichelt: ST 24C512 BN6*

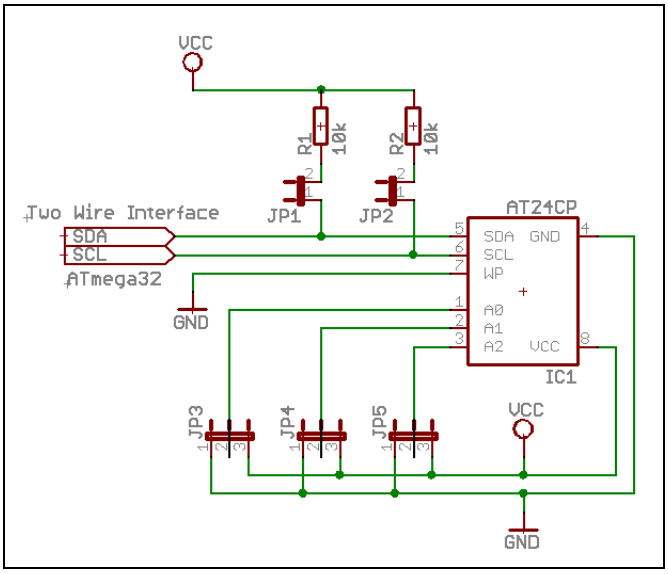
Im Projekt ConvCtrl wir der Baustein ST24C512 verwendet. Die Festlegung der Dateninhalte und Datenstrukturen wir zum Zeitpunkt der Systemprogrammierung getroffen und im Kapitel HW bzw. SW dokumentiert!

Der Baustein ST24C512 ist ein 512KBit Baustein. Er ist über 65.536 x 8 Bit organisiert. Er besitzt einen 128Byte Page-Write-Modus, erlaubt aber partielle Page-Zugriffe.

Aus dem Datenblatt des 24C512 geht hervor, welche Leistungsmerkmale er besitzt:

- Two Wite (I^2C) Interface
- Unterstützung von Busgeschwindigkeiten
- Byte- und Page-Write-Mode
- direkte Adressoerung
- automaisches Adress-Inkrement
- mehr als 1 Million Schreib/Lese Zyklen
- mindestens 40 Jahr Datensicherheit
- Geräte-ID 0b1010xxxx = 0xA0

Schaltbild:



Schaltbild 30: Beschaltung externes EEPROM

Zu Testzwecken können die gemäß Spezifikation geforderten PullUp-Widerstände des I^2C-Busses durch Jumper (JP1 und JP2) aktiviert / deaktiviert werden. Ggf. ist die Beschaltung auch durch Drahtbrücken im Layout zu ersetzen bzw. ausgeführt. In den Schaltplänen der weiteren I^2C Busteilnehmer sind die PullUp-Widerstände ebenfalls ausgeführt. In einem System mit mehreren Teilnehmern wird die Beschaltung jedoch nur einmal benötigt um den Bus zu treiben.

Der Abschlusswiderstand kann zwischen 5k und 10k variiert werden. Erste Versuche bei einer Baudrate von 400kBit zeigen keine sauberen Rechteckflanken auf dem Datenbus so dass der Einsatz von 5k vorbereitet und getestet wird.

Ein I^2C Gerät bildet aus seiner Geräte-ID und den möglichen Adresspins (A0-A2) seine Geräteadresse im Bus. Somit würden sich auch mehrere Geräte des gleichen Typs in einem Bus betreiben lassen. Für das Projekt ConvCtrl sind die Adressen via Jumper konfigurierbar.

Det gewählte Baustein ST24C512 besitzt nur zwei Adressleitung A0 und A1. Die Adressleitung A2 wie im obigen Schaltbild dargestellt (Pin3) ist beim ST24C512 not connected (NC).

Zum Betrieb des externen EEPROM wird die Adresse 0b00 gewählt. Daraus ergibt sich folgende Jumper-Stellung:

Jumper JP3 auf Position GND (1-2)

Jumper JP4 auf Position GND (1-2)

Jumper JP5 auf Position GND (1-2) (nicht relevant)

Bauteile:

Stückliste: Externes EEPROM			
Widerstände		*Halbleiter*	
R1, R2	10k Ω	IC1	Atmel EEPROM AT24C128/256/512
Sonstiges			
JP1, JP2	Jumper 2-polig, 2.54mm Raster		
JP3, JP4, JP5	Jumper 3-polig, 2.54mm Raster		

Tabelle 91: Stückliste externes EEPROM

Themenbezogene LED's:

LED / Funktion / Beschreibung	*Abkürzung*	*Farbe*	
Speicher voll	LED10_MEM_FULL	rot	*[LED13]*

Tabelle 92: Themenbezogene LED's für EEPROM

Ressourcenzuordnung zum ATmega128:

Nummer	*Schaltbild*		*Ressource ATmega128*		
1	SDA	[SDA]	PortD.1	[SDA]	(I^2C)
2	SCL	[SCL]	PortD.0	[SCL]	(I^2C)

Tabelle 93: Ressourcenzuordnung ATmega128 für EEPROM

Organisation des EEPROM AT24C512:

Das EEPROM AT24C512 besitzt 524.288 Bits welche seriell gelesen und geschrieben werden können. Das EEPROM ist in 65.536 Worte zu je 8 Bit organisiert.

Zusatzinformationen um Baustein:

Die Kommunikationsprotokolle sind dem Datenblatt des 24C512 zu entnehmen. Im folgenden sind die für das Projekt relevanten I^2C-Protokolle genauer beschrieben.

Die folgende Abbildung zeigt die Adressierung des EEPROM-Bausteins:

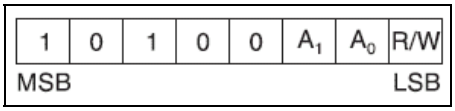


Abbildung 64: Adressierung des ST24C512

Byte schreiben:

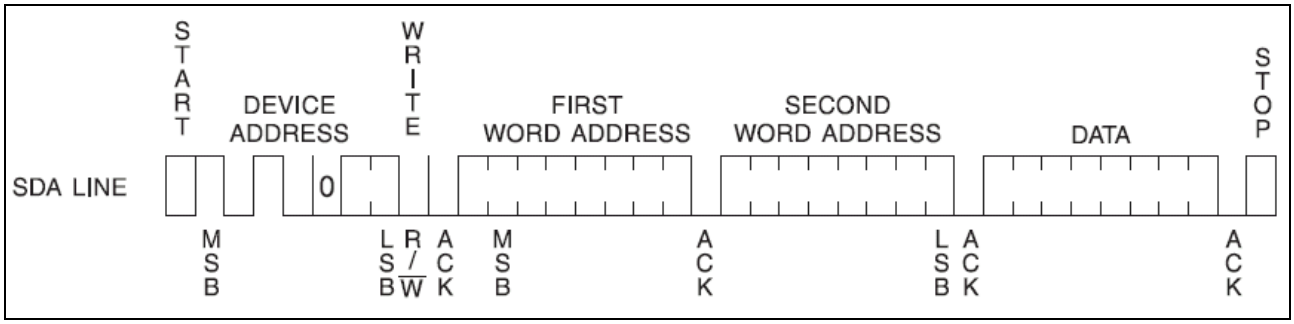


Abbildung 65: Daten in den ST24C512 schreiben

Byte lesen:

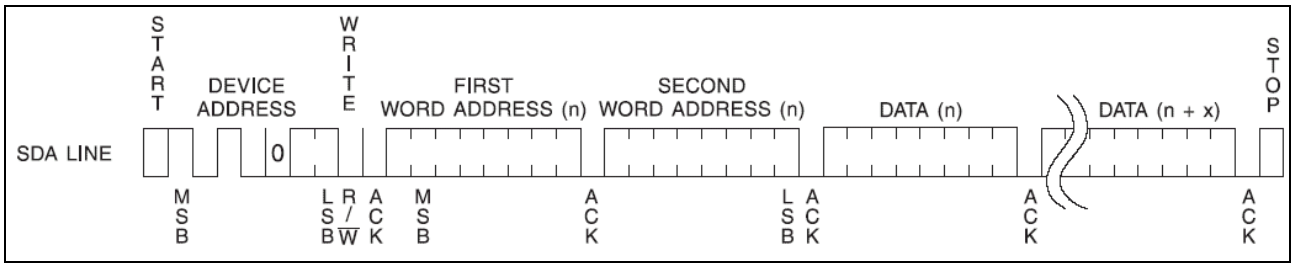


Abbildung 66: Daten aus dem ST24C512 lesen

Aktuelle Adressen auslesen:

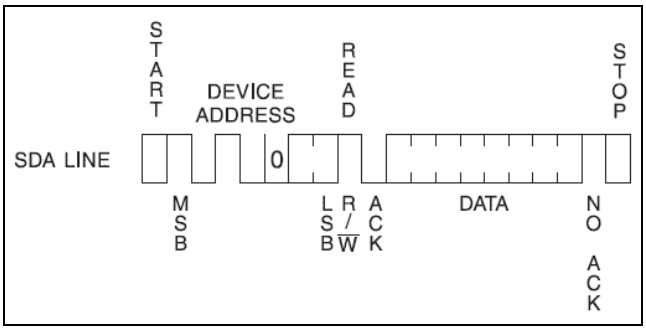


Abbildung 67: Aktuelle Adresse aus dem ST24C512 lesen

Random read:

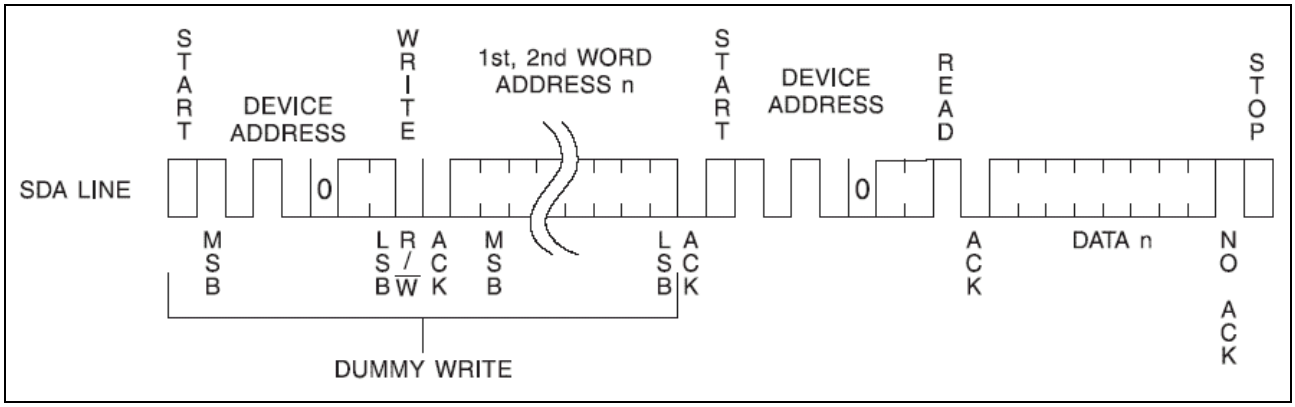


Abbildung 68: Daten aus dem ST24C512 via Radom-Read lesen

Sequentielles lesen:

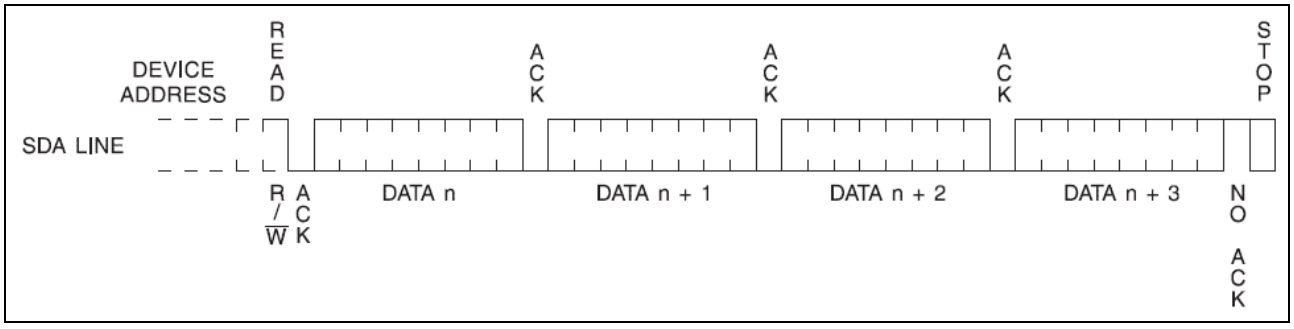


Abbildung 69: Daten sequentiell aus dem ST24C512 auslesen

Eine nützliche Hilfe ist der Beitrag über EEPROMS im Wiki
http://www.roboternetz.de/wissen/index.php/Bascom_und_I2C_EEprom

Beispiel einer Datensequenz im Projekt:

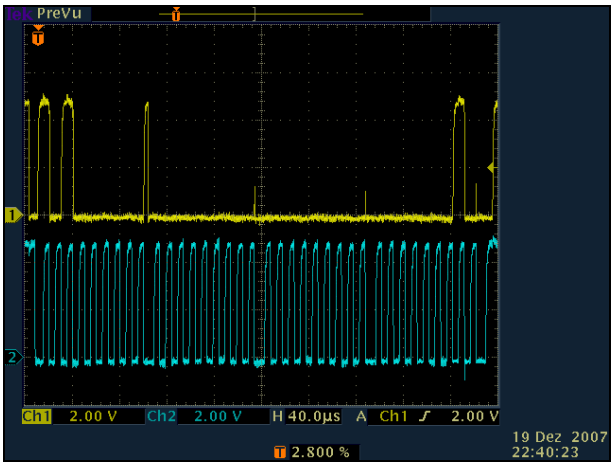


Abbildung 70: Oszillogramm einer I²C Schreibseqenz

Unterschiede bei der Beschaltung der Abschlusswiderstände:

Die folgenden Oszillogramme zeigen den Unterschied bei einer Beschaltung der I2C-Buss Abschlusswiderstände mit 5kΩ und 10kΩ.

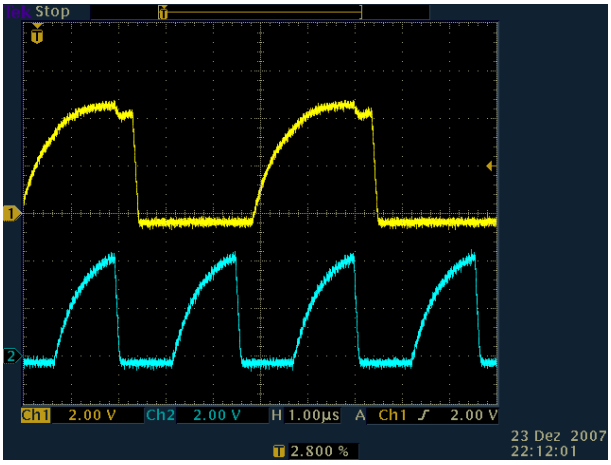


Abbildung 71: Oszillogramm I²C mit 10kΩ bei 400k Baud

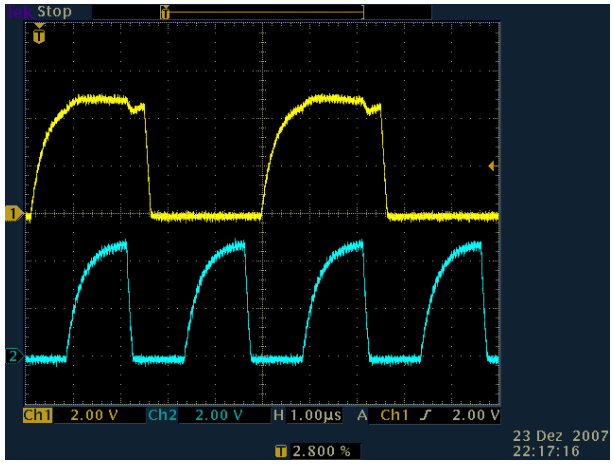


Abbildung 72: Oszillogramm I²C mit 5kΩ bei 400 kBaud

Bustaktrate 100k Baud gegenüber 400k Baud:

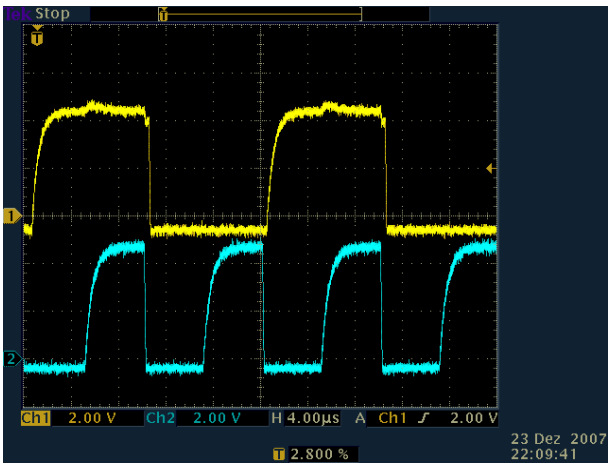


Abbildung 73: Oszillogramm I²C bei 100 kBaud

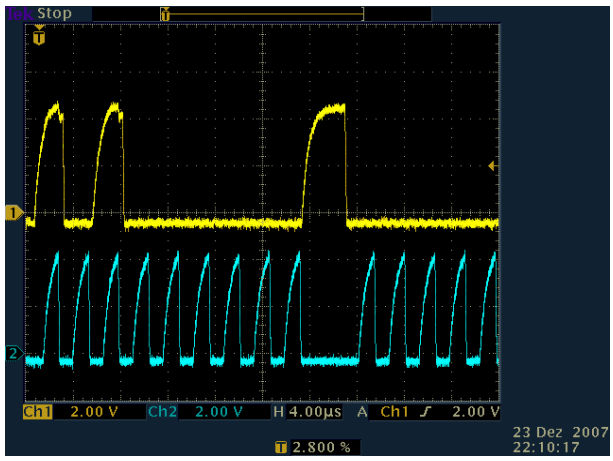


Abbildung 74: Oszillogramm I²C bei 400 kBaud

Bemerkung:

GELB (GELB) dargestellt ist die Datenleitung SDA
CYAN (CYAN) dargestellt ist die Clockleitung SCL

8.10.1 Initialisierung des EEPROM 24C512

Eine spezielle Initialisierung des EEPROM ist nicht erforderlich. Der Anwender kann aber durch das fortlaufende Schreiben von 0xFF oder 0x00 das EEPROM löschen.

8.10.2 Daten in das EEPROM 24C512 schreiben

Dem Datenblatt ent nehmen wir die Protokollsequenz zum Schreiben auf eine bestimmte Adresse des 24C512.

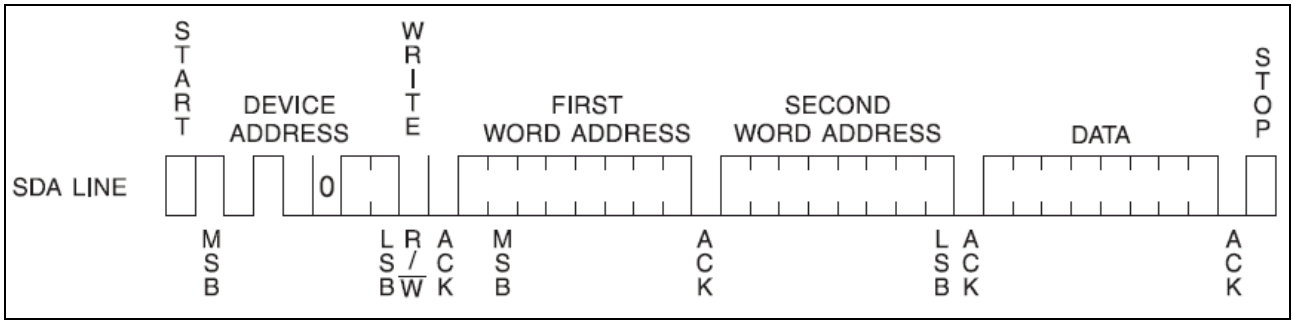


Abbildung 75: Ein Byte in den ST24C512 schreiben

Es sind also folgende I²C Befehle nötig um ein Byte auf Adresse X des EEPROM zu schreiben:

1. den ATmega als TWI-Master initialisieren
2. die START-Sequenzen auslösen
3. den EEPROM ansprechen (Geräte ID senden)
4. die gewünschte Adresse an das EEPROM senden (High+Low)
5. das Datenbyte senden
6. die STOP-Sequenze senden

Hierbei ist zu beachten, dass das EEPROM aufgrund seiner großen Speicherkapazität über zwei Adressbytes (First Word Adress, Second Word Address) verfügt.

Genauere Details sind dem Beispielcode am Ende des Kapitels zu entnehmen.

8.10.3 Daten vom EEPROM 24C512 lesen

Dem Datenblatt sentnehmen wir die Protokollsequenz zum Lesend eines Bytes von einer bestimmten Adresse des 24C512.

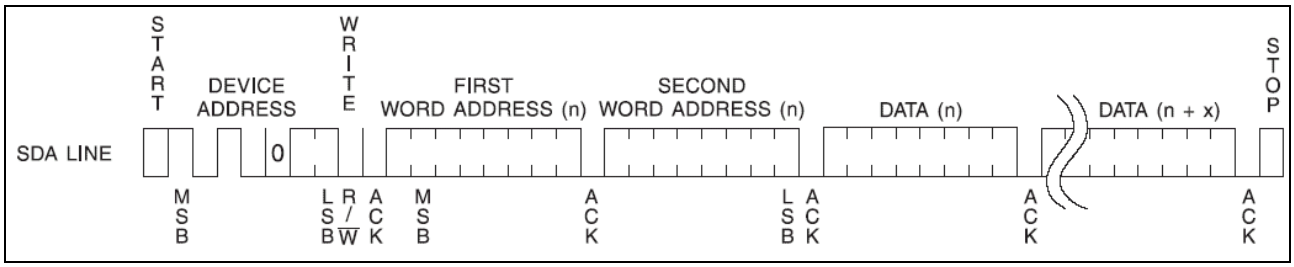


Abbildung 76: Ein Byte aus dem ST24C512 lesen

Es sind also folgende I^2C Befehle nötig um ein Byte von einer Adresse X des EEPROM zu lesen:

1. den ATmega als TWI-Master initialisieren
2. die START-Sequenz auslösen
3. Daten von der Adresse 0x0000 des EEPROM auslesen
   a. 24C512 adressieren
   b. Adresspointer auf Adresse 0 setzen (Al = 0x00, Ah = 0x00)
   c. Wiederholt TWI START senden
   d. 24C512 adressieren + R/W-Bit = Sendeaufforderung an Slave
   e. Slave sendet, Master empfängt
4. die STOP-Sequenz senden

Hierbei ist zu beachten, dass das EEPROM aufgrund seiner großen Speicherkapazität über zwei Adressbytes (First Word Adress, Second Word Address) verfügt.

Genauere Details sind dem Beispielcode am Ende des Kapitels zu entnehmen.

8.10.4 Adressraum und Datenfestlegung EEPROM

Der Adressraum des 24C512 umfasst die Adressen 0x0000 bis 0xFFFF (0 – 65.535).

Adressraum und Speichergröße:

MSB	Bits														LBS
F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
…															…
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
freier Adressraum													Daten-Struktur		

Adressraum: 2^13 = 8.192 Datenwerte

Werden alle 5 Minuten Umweltdaten in EEPROM abgelegt so reichen 8.192 Datenwerte für:
8.192 * 5min / 60 / 24 = ***28,4 Tage***

Werden alle 10 Minuten Umweltdaten in EEPROM abgelegt so reichen 8.192 Datenwerte für:
8.192 * 10min / 60 / 24 = ***56,9 Tage***

Werden alle 15 Minuten Umweltdaten in EEPROM abgelegt so reichen 8.192 Datenwerte für:
8.192 * 15min / 60 / 24 = ***85,3 Tage***

Tabelle 94: Adressraum und Speicherbedarf

Darstellung von negativen Zahlen in Binärsystemen:

positive Dez-Zahl	Binärdarstellung	negative Dez-Zahl	Binärdarstellung
0	00000000	0	
1	00000001	-1	1111111111
2	00000010	-2	1111111110
3	00000011	-3	1111111101
4	00000100	-4	1111111100
5	00000101	-5	1111111011
6	00000110	-6	1111111010
7	00000111	-7	1111111001
8	00001000	-8	1111111000
9	00001001	-9	1111110111
10	00001010	-10	1111110110
11	00001011	-11	1111110101
12	00001100	-12	1111110100
13	00001101	-13	1111110011
14	00001110	-14	1111110010
15	00001111	-15	1111110001

Tabelle 95: Negative Zahlen im dualen System

Datenstrukturen und Datenfelder im EEPROM:

Datenstruktur internes EEPROM (ATmega) und externes EEPROM (ST24C512)

Tabelle für Initialisierung und Basisdaten - internes EEPROM (Settings)

Adresse (von-bis)	HighAdr	LowAdr	Länge (in Byte)	Name	Bedeutung	Interpretation	Beschreibung
0x0000	0x00	0x00	1	Ee_data_version_intern	Version Datenstruktur	0x00 bis 0xFF	Legt die Version der initialisierten Datenstruktur fest *Default: 0x01*
0x0001	0x00	0x01	1	Ea_data_debug	Debug-Modus (Flag)	0x00 / 0x01	Debug Modus aktiv (0x01) oder deaktiv (0x00) *Default: 0x00*
0x0002	0x00	0x02	1	Ea_data_lcd_helligkeit	LCD Helligkeit	0x00 bis 0xFF	Vom Benutzer zuletzt eingestellte LCD Helligkeit *Default: 0x3C (=60)*
0x0003	0x00	0x03	1	Ea_data_lcd_kontrast	LCD Kontrast	0x00 bis 0x3F	Com Benutzer zuletzt eingestellter LCD Kontrast *Default: 0x0C (=12)*
0x0004	0x00	0x04	1	Ea_data_lcd_saver	LCD Screen Saver	0x00 bis 0xFF	Zeiteinstellung für LCD Screensaver in Sekunden. 0 bedeutet Funktion deaktiv. *Default: 0x00*
0x0005	0x00	0x05	1	Ee_data_wind_intervall	Zeit für Auswerteintervall Windgeschwindigkeit	0x01 / 0x1E	Auswerteintervall für Windgeschwindigkeit 5 - 30 Sekunden *Default: 0x03*
0x0006	0x00	0x06	1	Ea_data_wind_speed	Grenzgeschwindigkeit Wind	0x01 / 0x32	Grenzgeschwindigkeit für Wind zum automatischen Auffahren. *Default:0x0F*
0x0007	0x00	0x07	1	Ea_data_temp	Zeit für Auswerteintervall Temp und Feuchtigkeit	0x05 / 0x1E	Auswerteintervall für Temperatur und Feuchtigkeit 5 - 30 Sekunden *Default: 0x05*
0x0008	0x00	0x08	1	Ea_data_dimmer	Dimmer Helligkeit	0x00 / 0x1F	Vom Benutzer zuletzt eingestellte Dimmer-Helligkeit *Default: 0x10 (=16)*
0x0009	0x00	0x09	1	Ea_data_dim_min	Dimmer Min-Helligkeit	0x01 / 0x1F	Minimale einstellbare Dimmerhelligkeit *Default: 0x02*
0x000A	0x00	0x0A	1	Ea_data_dim_max	Dimmer Max-Helligkeit	0x01 / 0x1F	Maximale einstellbare Dimmerhelligkeit *Default: 0x1F (=31)*
0x000B	0x00	0x0B	1	Ee_data_sound	Soundflag	0x00 / 0x01	Einstellung ob Sound ausgegeben werden soll oder nicht. *Default: 0x01 (= EIN)*
0x000C	0x00	0x0C	1	Ee_data_logging_flag	Data-Logging	0x00 / 0x01	Einstellung ob Sound ausgegeben werden soll oder nicht. *Default: 0x01 (= EIN)*
0x000D	0x00	0x0D	2	Ee_data_logging_intervall	Logging-Intervall	0x01 / 0x708	Intervall in dem die Datenaufzeichnung durchgeführt werden soll. *Default: 0x0A (=10 min)*
0x000F	0x00	0x0F	1	Ea_data_conv_timeout	Motorabschaltung	0x00 bis 0xFF	Einstellbare Wartezeit bis automatische Abschaltung Beschattungsantrieb (auf / zu) *Default: 0x1E (=30 sec)*
0x0010	0x00	0x10	1	Ea_data_conv_temp_flag	Tempflag	0x00 / 0x01	Einstellung ob bei Temperatur zugefahren werden soll *Default: 0x00 (=AUS)*
0x0011	0x00	0x11	1	Ea_data_conv_temperatur	Temperaturgrenze	0x00 bis 0xFF	Komprimierter Temperaturwert ab dessen Überschreitung die Beschattung automatisch zugefahren wird. 0 bedeutet Funktion deaktiv! *Default: 0x00*
0x0012	0x00	0x12	1	Ea_data_conv_light_flag	Lightflag	0x00 / 0x01	Einstellung ob bei Licht aufgefahren werden soll *Default: 0x00 (=AUS)*
0x0013	0x00	0x13	1	Ea_data_conv_lichtwert	Helligkeitsgrenze	0x00 bis 0x0F	Einstellbare Grenze ab der bei Unterschreitung Beschattung wieder aufgefahren werden soll. 0 bedeutet Funktion deaktiv. *Default: 0x00*
0x0014	0x00	0x14	1	Ea_data_conv_hotkey	Hotkey Kreuztasten	0x00 / 0x01	Ein- bzw. Ausschalten der Kreuztasten-Hotkeys für Beschattung auf/zu *Default: 0x00*
0x0015	0x00	0x10	1	Ea_data_fan	Lüfter über ATmega	0x00 / 0x01	Einstellung ob Lüfter über SW gesteuert werden soll oder nicht. *Default: 0x01*
0x0016	0x00	0x16	1	Ee_data_fan_25_temp	Temperatur für 25% Lüfter	0x01 / 0x63	Einstellung für 25% Lüftergeschwindigkeit *Default: 20*
0x0017	0x00	0x17	1	Ee_data_fan_50_temp	Temperatur für 50% Lüfter	0x01 / 0x63	Einstellung für 50% Lüftergeschwindigkeit *Default: 35*
0x0018	0x00	0x18	1	Ee_data_fan_75_temp	Temperatur für 75% Lüfter	0x01 / 0x63	Einstellung für 75% Lüftergeschwindigkeit *Default: 50*
0x0019	0x00	0x19	1	Ee_data_fan_100_temp	Temperatur für 100% Lüfter	0x01 / 0x63	Einstellung für 100% Lüftergeschwindigkeit *Default: 65*
0x0020	0x00	0x20	1	Ee_data_dimmer_aktuell	Aktuelle Dimmereinstellung	0x01 / 0x1E	Speicherzelle für den aktuell gespeicherten Dimmerwert
0x0021	0x00	0x21	4	Ee_data_sht71_t_min	Minimaltemperatur	single	Speicherzelle für den minimalen Temperaturwert
0x0025	0x00	0x25	4	Ee_data_sht71_t_max	Maximaltemperatur	single	Speicherzelle für den maximalen Temperaturwert
0x0029	0x00	0x29	4	Ee_data_sht71_rhtrue_min	Minimalfeuchtigkeit	single	Speicherzelle für den minimalen Luftfeuchtigkeitswert
0x002D	0x00	0x2D	4	Ee_data_sht71_rhtrue_max	Maximalfeuchtigkeit	single	Speicherzelle für den maximalen Lüftfeuchtigkeitswert
0x0031	0x00	0x31	4	Ee_data_wind_kmh_max	Maximalwindgeschwindigkeit	0x00 / 0xFF	Speicherzelle für die maximale Windgeschwindigkeit

Tabelle für Basisdaten und Loggingdaten - externes EEPROM (Datenlogging)

Adresse (von-bis)	HighAdr	LowAdr	Länge (in Byte)	Name	Bedeutung	Interpretation	Beschreibung
0x0000	0x00	0x00	1	Ee_data_version_extern	Version Datenstruktur	0x00 bis 0xFF	Legt die Version der initialisierten Datenstruktur fest
0x0001	0x00	0x01	2	Ee_data_count	Datencounter High-Byte	0x00 bis 0xFF	Der Datencounter gibt an wie viele Datenpakete abgelegt wurden.
	0x00	0x02			Datencounter Low-Byte	0x00 bis 0xFF	
0x0003	0x00	0x03	1	Ee_data_overrun	Overrun-Zähler	0x00 bis 0xFF	Der Overrun-Counter zählt, wie oft das Speichern am Anfang begonnen wurde
0x0004	0x00	0x04	..	..	frei	..	..
0x0005	0x00	0x05	..	..	frei	..	..
0x0006	0x00	0x06	..	..	frei	..	..
0x0007	0x00	0x07	..	..	frei	..	..
0x0008	0x00	0x08	8	Ee_data_loging	Datenlogging	siehe unten	Erstes Datenpaket für Datenlogging
..	..	..	..	..	..	..	..
0xFFF8	0xFF	0xF8	8	Ee_data_loging	Datenlogging	siehe unten	Letztes Datenpaket für Datenlogging, Overrun++ und weiter bei 0x0008.

Bemerkung:
Der Speicher für das Datenlogging wird als Ringspeicher ausgelegt.
Ist das Ende erreicht, wir der Overrun-Zähler inkrementiert und das Beschreiben des Speicher beginnt wieder bei 0x0008.
Wann dieses Ereigniss eintritt ist abhängig von der Rhythmus in dem Umweltdaten gelogged werden.

Tabelle für Datenformate Datenlogging - externes EEPROM

Dateninhalt	Rel. Adr.	Pos	Länge	Name	Bedeutung	Interpretation	Beschreibung
Uhrzeit	0x00	0x00	1	Ee_data_hour	Stunde (h)	0x00 - 0x17	Beschreibt die Stunde des Datenpakets (0-23)
		0x01	1	Ee_data_minute	Minute (m)	0x00 - 0x3B	Beschreibt die Minute des Datenpakets (0-59)
Datum	0x02	0x02	1	Ee_data_day	Tag (T)	0x01 - 0x1F	Beschreibt den Tag des Datenpakets (1-31)
		0x03	Bit 7..4	Ee_data_month	Monat (low nibble) (M)	0x?1 - 0x?C	Beschreibt den Monat des Datenpakets (1-12) Bit [3..0] = low-nibble
			Bit 3..0	Ea_data_light	Helligkeit (high-nibbel) (lum)	0x0? - 0xF?	Gemessene Helligkeit reudziert auf 4 Bit Bit [7..4] = high-nibble
		0x04	1	Ee_data_year	Jahr (J)	0x00 - 0x63	Beschreibt das Jahr des Datenpakets (0-99)
Regen	0x05	0x05	Bit 7	Ee_data_rain	Regen (rain)	Bit 7.0 / 7.1	Flag ob zum Zeitpunkt der Datenerfassung Regen erkannt
Wind			Bit 6..0	Ee_data_wind	Wind (wind)	0x00 - 0x7F	Windgeschwindigkeit 0 - 128 m/s = 0 - 460,8 km/h
Temperatur	0x06	0x06	1	Ee_data_temp	Temperatur (lowbyte) (temp)	0x00 - 0xFF	Komprimiertes Temperatur-Low-Byte (siehe unten)
Luftfeuchtigkeit	0x09	0x07	1	Ee_data_humidity	Luftfeuchtigkeit (hum)	0x00 - 0x63	Luftfeuchtigkeit 0 - 99 % relative Luftfeuchte

Bemerkung:
Die Temperatur des Sensors SHT71 kann im Bereich von -40 bis +124 °C liegen. Hierzu reicht die Umsetzung in einem Byte. => Komprimierung Ea_data_temp

Komprimierung Ea_data_temp:
- Der Wertebereich 0x0000 - 0xFFFF, -32.768 bis +32.767 °C wird durch Abschneiden des High-Byte reduziert.
- Gespeichert wird nur das Low-Byte.
- Zur Restaurierung des originalen Integerwerts aus dem Byte wir Vorzeichnehaftet ergänzt.
- Ist Bit 7 = High wird das High-Byte auf 0b11111111 = 0xFF gesetzt
- Ist Bit 7 = low wird das High-Byte auf 0b00000000 = 0x00 gesetzt
- Die Darstellung von negativen Zahlen im Integer erfolgt über das Zweierkompliment

Ausgabeformat für RS-232 Datenübertragung:
Format lfd-nr;TT.MM.JJJJ;hh:mm;lum;rain;wind;temp;hum
Beispiel 1;27.12.2007;14:48;12;0;10;-12;51

Tabelle 96: Datenstrukturen und Datenfelder im EEPROM (intern und extern)

Es wird vorgesehen die Tabelle für die Initialisierung und die Basisdaten im internen EEPROM des ATmega128 abzulegen.

Für Das Datenlogging der Umweltdaten dient das externe EEPROM.

8.10.5 BASCOM Beispielcoder für EEPROM

```
'#################################################################################
' Eeprom.BAS:                                                    Stand 28.12.2007
' -----------                                                    (C) Markus Fulde
' Testprogramm zur Inbetriebnahme des EEPROM ST24C512 mit I2C Bus über
' das TWI Interface des ATmega.
' Die Inbetriebnahme erfolgt mit dem Eva-Board STK500 + STK501
' Die für Ansteuerung des EEPROM notwendigen Rotuinen (Init, Read, Write)
' sind als solche gekennzeichnet und bereits zur späteren Verwendung
' ausgelagert.
'#################################################################################


'--------------------------------------------------------------------------------
' Compilerinstruktionen und Compilerdirektiven
'--------------------------------------------------------------------------------
$regfile = "m128def.dat"
```

```
$crystal = 16000000
$hwstack = 32
$swstack = 32
$framesize = 40

$baud = 38400

' Achtung: $sim beeinflusst Timer und Wait-Statements!
'$sim                                                    ' für reales Target abgeschaltet!

$lib "i2c_twi.lbx"                                      ' we do not use software emulated I2C
but the TWI


'-----------------------------------------------------------------------------
' Definition von Ressourcen
'-----------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========
Sda_port Alias Portd.1
Scl_port Alias Portd.0

'========== Für Testumgebung  ==========

' Alias Namen für Tastaturabfragen
Taste_1 Alias Pina.0
Taste_2 Alias Pina.1
Taste_3 Alias Pina.2
Taste_4 Alias Pina.3
Taste_5 Alias Pina.4
Taste_6 Alias Pina.5
Taste_7 Alias Pina.6
Taste_8 Alias Pina.7

' Alias Namen für LED's
Led0 Alias Portb.0
Led1 Alias Portb.1
Led2 Alias Portb.2
Led3 Alias Portb.3
Led4 Alias Portb.4
Led5 Alias Portb.5
Led6 Alias Portb.6
Led7 Alias Portb.7


'-----------------------------------------------------------------------------
' Definition von Variablen, Konstaten und Datentypen
'-----------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========

' EE: EEPROM Datenstruktur internes EEPROM - Setting
Dim Ee_data_version_intern As Eram Byte At &H0000
Dim Ea_data_debug As Eram Byte At &H0001
Dim Ea_data_lcd_helligkeit As Eram Byte At &H0002
Dim Ea_data_lcd_kontrast As Eram Byte At &H0003
Dim Ea_data_lcd_saver As Eram Byte At &H0004
Dim Ea_data_lcd_light As Eram Byte At &H0005
Dim Ea_data_wind As Eram Byte At &H0006
Dim Ea_data_temp As Eram Byte At &H0007
Dim Ea_data_dimmer As Eram Byte At &H0008
Dim Ea_data_dim_min As Eram Byte At &H0009
Dim Ea_data_dim_max As Eram Byte At &H000A
Dim Ea_data_conv_time As Eram Byte At &H000B
Dim Ea_data_conv_wind As Eram Byte At &H000C
Dim Ea_data_conv_temp As Eram Byte At &H000D
Dim Ea_data_conv_light As Eram Byte At &H000E
Dim Ea_data_conv_hotkey As Eram Byte At &H000F
Dim Ea_data_fan As Eram Byte At &H0010
Dim Ea_data_fan_temp As Eram Byte At &H0011
Dim Ea_data_fan_min_temp As Eram Byte At &H0012
Dim Ea_data_fan_max_temp As Eram Byte At &H0013


' EE: Grenzwerte für Setting-Einstellungen
Const Debug_an = &H01                                   ' Debugausgaben an
Const Debug_aus = &H00                                  ' Debugausgaben aus

Const Lcd_helligkeit_min = 0                            ' minimale Helligkeit
Const Lcd_helligkeit_max = 255                          ' maximale Helligkeit

Const Lcd_kontrast_min = 0                              ' minimaler kontrast
Const Lcd_kontrast_max = 63                             ' maximaler Kontrast

Const Lcd_screen_saver_min = &H00                       ' minimale Screensaver-Zeit (0=aus)
Const Lcd_screen_saver_max = &HFF                       ' maximale Screensaver-Zeit (255sec)

Const Lcd_beleuchtung_an = &H01                         ' Beleuchtung ein
Const Lcd_beleuchtung_aus = &H00                        ' Beleuchtung aus
```

```
Const Intervall_auswertung_min = &H05                    ' Auswertung alle 5 Sekunden
Const Intervall_auswertung_max = &H1E                    ' Auswertung alle 30 Sekunden

Const Dimmer_helligkeit_min = &H01                       ' minimaler Dimmerindex 1
Const Dimmer_helligkeit_max = &H1E                       ' maximaler Dimmerindex 31

Const Motorabschaltung_min = &H01                        ' Motorabschaltung nach 1 Sekunde
Const Motorabschaltung_max = &HFF                        ' Motorabschaltung anch 255 Sekunden

Const Windgrenzwert_min = &H00                           ' Wind bei 0 m/2 (0=aus)
Const Windgrenzwert_max = &H7F                           ' Wind bei 128 m/s

Const Temperaturgrenzwert_min = 1                        ' unterer Temperaturgrenzwert (0=aus)
Const Temperaturgrenzwert_max = 100                      ' oberer Temperaturgrenzwert

Const Lichtgrenzwert_min = 0                             ' unterer Lichtgrenzwert (Nacht) (0=aus)
Const Lichtgrenzwert_max = 15                            ' oberer Lichtgrenzwert(tag)

Const Hotkeykreuztaste_an = &H01                         ' Kreuztasten als Hotkey zulassen
Const Hotkeykreuztaste_aus = &H00                        ' Kreuzstasten als Hotkey deaktiv

Const Sw_fan_an = &H01                                   ' Lüfter über Temperatur gesteuert
Const Sw_fan_aus = &H00                                  ' Lüfter dauerhaft eingeschaltet

Const Atmega_fan_an = &H01                               ' Lüfter über ATmega gesteuert
Const Atmega_fan_aus = &H00                              ' Lüfter über Atmega nicht gesteuert

Const Fan_hysterese_abschalten_min = 30                  ' Unterer Hysteresepunkt von 30°
Const Fan_hysterese_abschalten_max = 45                  ' bis 45°

Const Fan_hysterese_einschalten_min = 50                 ' Oberer Hysteresepunkt von 50°
Const Fan_hysterese_einschalten_max = 80                 ' bis 80°


' EE: EEPROM Datenstruktur externes EEPROM - Datenlogging
Dim Ee_data_version_extern As Byte
Dim Ee_data_count As Word
Dim Ee_data_overrun As Byte

' Byte(1) Ee_data_hour
' Byte(2) Ee_data_minute
' Byte(3) Ee_data_day
' Byte(4) Ee_data_month [Bit7..4]
' Byte(4) Ea_data_light [Bit3..0]
' Byte(5) Ee_data_year
' Byte(6) Ee_data_rain  [Bit7]
' Byte(6) Ee_data_wind  [Bit6..0]
' Byte(7) Ee_data_temp
' Byte(8) Ee_data_humidity
Dim Ee_data_logging_struct(8) As Byte


' EE: Konstanten allgemein
Const Eeprom_testmodus = 1                               ' Testmodus für EEPROM

Const Akt_internal_ee_version = &H02                     ' Aktuelle EEPROM-Datenstruktur intern
Const Akt_external_ee_version = &H01                     ' Aktuelle EEPROM-Datenstruktur extern


' EE: Konstanten für Datenlogging (externes EEPROM)
Const Ee_data_version_extern_adress = &H0000
Const Ee_data_count_adress = &H0001
Const Ee_data_overrun_adress = &H0003
Const Ee_data_loging_start_adress = &H0008
Const Ea_data_loging_length = 8
Const Ea_data_loging_endadress = &H1FFF                  ' = 8191
'Const Ea_data_loging_endadress = &H000A                  ' = 10 for Test

Const Ee_data_count_default = &H0000
Const Ee_data_overrun_default = &H00


' EE: Konsten für Default-Settings der Einstellung Version 1
Const Ea_data_debug_default = &H00
Const Ea_data_lcd_helligkeit_default = &H3C
Const Ea_data_lcd_kontrast_default = &H0C
Const Ea_data_lcd_saver_default = &H00
Const Ea_data_lcd_light_default = &H01
Const Ea_data_wind_default = &H05
Const Ea_data_temp_default = &H05
Const Ea_data_dimmer_default = &H10
Const Ea_data_dim_min_default = &H02
Const Ea_data_dim_max_default = &H1F
Const Ea_data_conv_time_default = &H78
Const Ea_data_conv_wind_default = &H00
Const Ea_data_conv_temp_default = &H00
Const Ea_data_conv_light_default = &H00
Const Ea_data_conv_hotkey_default = &H00
Const Ea_data_fan_default = &H01
Const Ea_data_fan_temp_default = &H00
```

```
Const Ea_data_fan_min_temp_default = &H28
Const Ea_data_fan_max_temp_default = &H32

' EE:  Adressen für das exerne EEPROM
Const Address_write = &B10100000                          ' Adresse des 24C512 = &HA0 (schreiben)
Const Address_read = &B10100001                           ' Adresse des 24C512 = &HA1 (lesen)

' EE: Variablen
Dim Ee_adresse As Word


'========== Für Testumgebung ==========
Dim Temp_value As Byte
Dim Temp_adress As Word
Dim Temp_count As Byte

Dim Temp_byte1 As Byte
Dim Temp_byte2 As Byte

Dim Temp_word1 As Word
Dim Temp_word2 As Word
Dim Temp_word3 As Word

Dim Temp_int1 As Integer

Dim Direction As Byte

Dim Testarray_byte(4) As Byte
Testarray_byte(1) = 4
Testarray_byte(2) = 3
Testarray_byte(3) = 2
Testarray_byte(4) = 1

Dim Testarray_word(4) As Word
Testarray_word(1) = &HAFFE
Testarray_word(2) = &H0110
Testarray_word(3) = &HFFFF
Testarray_word(4) = &H9999


' Testdaten für Umweltdatenlogging
Dim Test_data_hour As Byte
Test_data_hour = 14

Dim Test_data_minute As Byte
Test_data_minute = 48

Dim Test_data_day As Byte
Test_data_day = 27

Dim Test_data_month As Byte                               ' [Bit7..4]
Test_data_month = 12

Dim Test_data_light As Byte                               ' [Bit3..0]
Test_data_light = 9

Dim Test_data_year As Word
Test_data_year = 2007

Dim Test_data_rain As Byte                                ' [Bit7]
Test_data_rain = 0

Dim Test_data_wind As Byte                                ' [Bit6..0]
Test_data_wind = 10

Dim Test_data_temp As Integer
'Test_data_temp = 17
Test_data_temp = -12

Dim Test_data_humidity As Byte
Test_data_humidity = 51


'-------------------------------------------------------------------------------
' Prototyping
'-------------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========
Declare Sub Ee_write_eeprom_byte(byval Adress As Word , Byval Value As Byte)
Declare Sub Ee_write_eeprom_word(byval Adress As Word , Byval Value As Word)

Declare Sub Ee_write_eeprom_struct(byval Adress As Word)
Declare Sub Ee_read_eeprom_struct(byval Adress As Word)

Declare Function Ee_read_eeprom_byte(byval Adress As Word) As Byte
Declare Function Ee_read_eeprom_word(byval Adress As Word) As Word

'========== Für Testumgebung ==========
```

```
'-----------------------------------------------------------------------------
' Konfiguration und Basiseinstellungen (Projekt und Testumgebung)
'-----------------------------------------------------------------------------

'========== Für Testumgebung ==========

'Konfiguration des Port A als Eingang für Tastaturabfrage
Config Porta = Input

'Konfiguration des Port B als Ausgang für Statusanzeige via LED
Config Portb = Output
Portb = &B11111111


'========== Für Projekt ConvCtrl ==========
Config Sda = Sda_port
Config Scl = Scl_port

I2cinit                                                 ' we need to set the pins in the proper
state

Config Twi = 400000                                     ' Taktfrequenz 400 kBaud
'Config Twi = 100000

' TWI gleich einschalten, das macht BASCOM ansonsten erst beim I2CStart
Twcr = &B00000100                                       ' nur TWEN setzen


Wait 5                                                  ' Zeit genug um Terminal zu starten

' Initialisierung des internen EEPROM aufrufen und ggf. Traces schreiben
Gosub Ee_init_internal_eeprom

' Initialisierung des externen EEPROM aufrufen und ggf. Traces schreiben
Gosub Ee_init_external_eeprom


'#############################################################################
' Hauptprogramm
'#############################################################################

Print "TWI master"

'Hier ist die Programmhauptschleife
Do

   ' ===== Tasten EEPROM-Testmenü =====

   ' Tastaturabfrage Taste 1
   Debounce Taste_1 , 0 , Write_testdata_byte_to_eeprom , Sub

   ' Tastaturabfrage Taste 2
   Debounce Taste_2 , 0 , Read_testdata_byte_from_eeprom , Sub

   ' Tastaturabfrage Taste 3
   Debounce Taste_3 , 0 , Write_und_read_gesamtes_eeprom , Sub

   ' Tastaturabfrage Taste 4
   Debounce Taste_4 , 0 , Write_testdata_word_to_eeprom , Sub

   ' Tastaturabfrage Taste 5
   Debounce Taste_5 , 0 , Read_testdata_word_from_eeprom , Sub

   ' Tastaturabfrage Taste 7
   Debounce Taste_7 , 0 , Write_testdata_datalogging , Sub

   ' Tastaturabfrage Taste 8
   Debounce Taste_8 , 0 , Read_testdata_datalogging , Sub

Loop
'## End Hauptprogramm ###########################################################


'*****************************************************************************
' Subroutinen
'*****************************************************************************


'-----------------------------------------------------------------------------
' EE - Subroutine: Ee_write_eeprom_byte
' Subroutine um ein einzelnes Byte an eine bestimmte Speicheradresse in das
' EEPROM zu schreiben.
'
' Parameter:   Adress = Adresse (Word) an die geschrieben werden soll
'              Value  = Bytevariable die geschrieben werden soll
'-----------------------------------------------------------------------------
Sub Ee_write_eeprom_byte(byval Adress As Word , Byval Value As Byte)

   ' ----- Lokale Variable definieren -----
   Local Ee_adrl As Byte
```

```
   Local Ee_adrh As Byte

   ' ----- Programmcode -----

   ' Adresse in High-Byte und Low-Byte zerlegen
   Ee_adrl = Low(adress)
   Ee_adrh = High(adress)

   ' Kommunikation über I2C-Bus
   I2cstart                                              ' START-Sequenze senden
   I2cwbyte Address_write                                ' Daten schreiben initiieren
   I2cwbyte Ee_adrh                                      ' High-Byte der Adresse übertragen
   I2cwbyte Ee_adrl                                      ' Low-Byte der Adresse übertragen
   I2cwbyte Value                                        ' Byte-Variable übertragen
   I2cstop                                               ' STOP-Sequenz

End Sub
'-- End Ee_write_eeprom_byte ------------------------------------------------


'-----------------------------------------------------------------------------
' EE - Subroutine: Ee_write_eeprom_word
' Subroutine um ein zwei Bytes (Word) an eine bestimmte Speicheradresse in das
' EEPROM zu schreiben.
'
' Parameter:   Adress = Adresse (Word) an die geschrieben werden soll
'              Value  = Wordvariable die geschrieben werden soll
'-----------------------------------------------------------------------------
Sub Ee_write_eeprom_word(byval Adress As Word , Byval Value As Word)

   ' ----- Lokale Variable definieren -----
   Local Ee_adrl As Byte
   Local Ee_adrh As Byte

   Local Ee_datal As Byte
   Local Ee_datah As Byte

   ' ----- Programmcode -----

   ' Adresse in High-Byte und Low-Byte zerlegen
   Ee_adrl = Low(adress)
   Ee_adrh = High(adress)

   ' Word in einzelne Bytes zerlegen
   Ee_datal = Low(value)
   Ee_datah = High(value)

   #if Eeprom_testmodus
      Print "Data high: " ; Hex(ee_datah)
      Print "Data low:  " ; Hex(ee_datal)
   #endif

   ' Kommunikation über I2C-Bus
   I2cstart                                              ' START-Sequenze senden
   I2cwbyte Address_write                                ' Daten schreiben initiieren
   I2cwbyte Ee_adrh                                      ' High-Byte der Adresse übertragen
   I2cwbyte Ee_adrl                                      ' Low-Byte der Adresse übertragen
   I2cwbyte Ee_datah                                     ' High-Byte aus Variable übertragen
   I2cwbyte Ee_datal                                     ' Low-byte aus Variable übertragen
   I2cstop                                               ' STOP-Sequenz

End Sub
'-- End Ee_write_eeprom_word ------------------------------------------------


'-----------------------------------------------------------------------------
' EE - Subroutine: Ee_write_eeprom_loggingstruct
' Subroutine schreibt die 8 Byte Loggingdaten an die übergebene Adresse
'
' Parameter: Adress        = Adresse (Word) ab der die Daten geschrieben werden
'-----------------------------------------------------------------------------
Sub Ee_write_eeprom_struct(byval Adress As Word)

   ' ----- Lokale Variable definieren -----
   Local Ee_adrl As Byte
   Local Ee_adrh As Byte

   ' ----- Programmcode -----

   #if Eeprom_testmodus
      Print "EE-write Adresse " ; Hex(adress)
   #endif

   ' Adresse in High-Byte und Low-Byte zerlegen
   Ee_adrl = Low(adress)
   Ee_adrh = High(adress)
   #if Eeprom_testmodus
      Print "EE-write low  Adresse " ; Hex(ee_adrl)
      Print "EE-write high Adresse " ; Hex(ee_adrh)
      Print "Daten-Struct : " ;
```

```
      Print Hex(ee_data_logging_struct(1)) ; " " ;
      Print Hex(ee_data_logging_struct(2)) ; " " ;
      Print Hex(ee_data_logging_struct(3)) ; " " ;
      Print Hex(ee_data_logging_struct(4)) ; " " ;
      Print Hex(ee_data_logging_struct(5)) ; " " ;
      Print Hex(ee_data_logging_struct(6)) ; " " ;
      Print Hex(ee_data_logging_struct(7)) ; " " ;
      Print Hex(ee_data_logging_struct(8))
   #endif

   ' Kommunikation über I2C-Bus
   I2cstart                                          ' START-Sequenze senden
   I2cwbyte Address_write                            ' Daten schreiben initiieren
   I2cwbyte Ee_adrh                                  ' High-Byte der Adresse übertragen
   I2cwbyte Ee_adrl                                  ' Low-Byte der Adresse übertragen
   I2cwbyte Ee_data_logging_struct(1)                ' Byte 1 übertragen
   I2cwbyte Ee_data_logging_struct(2)                ' Byte 2 übertragen
   I2cwbyte Ee_data_logging_struct(3)                ' Byte 3 übertragen
   I2cwbyte Ee_data_logging_struct(4)                ' Byte 4 übertragen
   I2cwbyte Ee_data_logging_struct(5)                ' Byte 5 übertragen
   I2cwbyte Ee_data_logging_struct(6)                ' Byte 6 übertragen
   I2cwbyte Ee_data_logging_struct(7)                ' Byte 7 übertragen
   I2cwbyte Ee_data_logging_struct(8)                ' Byte 8 übertragen
   I2cstop                                           ' STOP-Sequenz

End Sub
'-- End Ee_write_eeprom_loggingstruct ------------------------------------------


'-------------------------------------------------------------------------------
' EE - Subroutine: Ee_read_eeprom_loggingstruct
' Subroutine liest die 8 Byte Loggingdaten von der übergebene Adresse
'
' Parameter: Adress        = Adresse (Word) von der die Daten gelesen werden
'-------------------------------------------------------------------------------
Sub Ee_read_eeprom_struct(byval Adress As Word)

   ' ----- Lokale Variable definieren -----
   Local Ee_adrl As Byte
   Local Ee_adrh As Byte

   ' ----- Programmcode -----
   ' Adresse in High-Byte und Low-Byte zerlegen
   Ee_adrl = Low(adress)
   Ee_adrh = High(adress)

   ' Kommunikation über I2C-Bus
   I2cstart                                          ' START-Sequenz senden
   I2cwbyte Address_write                            ' Daten schreiben initiieren
   I2cwbyte Ee_adrh                                  ' High-Byte der Adresse übertragen
   I2cwbyte Ee_adrl                                  ' Low-Byte der Adresse übertragen

   I2cstart                                          ' START-Sequenz erneut senden
   I2cwbyte Address_read                             ' Daten lesen initiieren
   I2crbyte Ee_data_logging_struct(1) , Ack          ' Byte 1 lesen
   I2crbyte Ee_data_logging_struct(2) , Ack          ' Byte 2 lesen
   I2crbyte Ee_data_logging_struct(3) , Ack          ' Byte 3 lesen
   I2crbyte Ee_data_logging_struct(4) , Ack          ' Byte 4 lesen
   I2crbyte Ee_data_logging_struct(5) , Ack          ' Byte 5 lesen
   I2crbyte Ee_data_logging_struct(6) , Ack          ' Byte 6 lesen
   I2crbyte Ee_data_logging_struct(7) , Ack          ' Byte 7 lesen
   I2crbyte Ee_data_logging_struct(8) , Nack         ' Byte 8 lesen
   I2cstop                                           ' STOP-Sequenz

End Sub
'-- End Ee_write_eeprom_loggingstruct ------------------------------------------


'*******************************************************************************
' Funktionen
'*******************************************************************************

'-------------------------------------------------------------------------------
' EE - Funktion: Ee_read_eeprom_byte
' Funktion um ein einzelnes Byte von einer bestimmten Speicheradresse aus dem
' EEPROM zu lesen.
'
' Parameter:     Adress = Adresse (Word) von der das Byte gelesen werden soll
'
' Rückgabewert:  Bytevariable die von der Adresse ausgelesen wurde.
'-------------------------------------------------------------------------------
Function Ee_read_eeprom_byte(byval Adress As Word) As Byte

   ' ----- Lokale Variable definieren -----
   Local Ee_adrl As Byte
   Local Ee_adrh As Byte
   Local Temp_byte As Byte

   ' ----- Programmcode -----
```

```
    ' Adresse in High-Byte und Low-Byte zerlegen
    Ee_adrl = Low(adress)
    Ee_adrh = High(adress)

    ' Kommunikation über I2C-Bus
    I2cstart                                                ' START-Sequenz senden
    I2cwbyte Address_write                                  ' Daten schreiben initiieren
    I2cwbyte Ee_adrh                                        ' High-Byte der Adresse übertragen
    I2cwbyte Ee_adrl                                        ' Low-Byte der Adresse übertragen

    I2cstart                                                ' START-Sequenz erneut senden
    I2cwbyte Address_read                                   ' Daten lesen initiieren
    I2crbyte Temp_byte , Nack                               ' Byte auslesen und quittieren
    I2cstop                                                 ' STOP-Sequenz

    ' Temporäre Variable als Rückgabewert übergeben
    Ee_read_eeprom_byte = Temp_byte

End Function
'-- End Ee_read_eeprom_byte ------------------------------------------------


'-------------------------------------------------------------------------------
' EE - Funktion: Ee_read_eeprom_word
' Funktion um ein zwei Bytes (Word) von einer bestimmten Speicheradresse aus
' dem EEPROM zu lesen.
'
' Parameter:     Adress = Adresse (Word) von der das Word gelesen werden soll
'
' Rückgabewert:  Wordvariable die von der Adresse ausgelesen wurde.
'-------------------------------------------------------------------------------
Function Ee_read_eeprom_word(byval Adress As Word) As Word

    ' ----- Lokale Variable definieren -----
    Local Ee_adrl As Byte
    Local Ee_adrh As Byte

    Local Ee_datal As Byte
    Local Ee_datah As Byte

    Local Temp_word As Word

    ' ----- Programmcode -----

    ' Adresse in High-Byte und Low-Byte zerlegen
    Ee_adrl = Low(adress)
    Ee_adrh = High(adress)

    ' Kommunikation über I2C-Bus
    I2cstart                                                ' START-Sequenz senden
    I2cwbyte Address_write                                  ' Daten schreiben initiieren
    I2cwbyte Ee_adrh                                        ' High-Byte der Adresse übertragen
    I2cwbyte Ee_adrl                                        ' Low-Byte der Adresse übertragen

    I2cstart                                                ' START-Sequenz erneut senden
    I2cwbyte Address_read                                   ' Daten lesen initiieren
    I2crbyte Ee_datah , Ack                                 ' High-Byte auslesen und weiter
    I2crbyte Ee_datal , Nack                                ' Low-Byte auslesen und Ende
    I2cstop                                                 ' STOP-Sequenz

    #if Eeprom_testmodus
       Print "Data high: " ; Hex(ee_datah)
       Print "Data low:  " ; Hex(ee_datal)
    #endif

    ' Temporäre Variable als Rückgabewert übergeben
    Temp_word = Ee_datah
    Shift Temp_word , Left , 8
    Temp_word = Temp_word And &HFF00
    Temp_word = Temp_word + Ee_datal

    Ee_read_eeprom_word = Temp_word

End Function
'-- End Ee_read_eeprom_byte ------------------------------------------------


'*******************************************************************************
' GOSubroutinen
'*******************************************************************************

'-------------------------------------------------------------------------------
' GOSubroutine: Ee_init_internal_eeprom
' Subroutine überprüft die Datenkonsitenz aufgrund des verwendeten Versions-
' Index und führt ggf. eine Grundinitialisierung der Datenstrukturen durch.
' Die Settingdaten werden aus dem internen EEPROM ausgelesen.
'-------------------------------------------------------------------------------
Ee_init_internal_eeprom:
```

```
   #if Eeprom_testmodus
      Print "Enter Ee_init_internal_eeprom"
   #endif


   Temp_byte1 = Ee_data_version_intern
   If Temp_byte1 <> Akt_internal_ee_version Then

      #if Eeprom_testmodus
         Print "Datenversion stimmt nicht"
         Print "Initialisiere internes EEPROM neu!"
      #endif

      ' Datenstruktur internes EEPROM entspricht nicht dem aktuellen Stand
      ' => Datenstruktur neu anlegen bzw. neue Defaultwerte setzen.

      Ee_data_version_intern = Akt_internal_ee_version
      Ea_data_debug = Ea_data_debug_default
      Ea_data_lcd_helligkeit = Ea_data_lcd_helligkeit_default
      Ea_data_lcd_kontrast = Ea_data_lcd_kontrast_default
      Ea_data_lcd_saver = Ea_data_lcd_saver_default
      Ea_data_lcd_light = Ea_data_lcd_light_default
      Ea_data_wind = Ea_data_wind_default
      Ea_data_temp = Ea_data_temp_default
      Ea_data_dimmer = Ea_data_dimmer_default
      Ea_data_dim_min = Ea_data_dim_min_default
      Ea_data_dim_max = Ea_data_dim_max_default
      Ea_data_conv_time = Ea_data_conv_time_default
      Ea_data_conv_wind = Ea_data_conv_wind_default
      Ea_data_conv_temp = Ea_data_conv_temp_default
      Ea_data_conv_light = Ea_data_conv_light_default
      Ea_data_conv_hotkey = Ea_data_conv_hotkey_default
      Ea_data_fan = Ea_data_fan_default
      Ea_data_fan_temp = Ea_data_fan_temp_default
      Ea_data_fan_min_temp = Ea_data_fan_min_temp_default
      Ea_data_fan_max_temp = Ea_data_fan_max_temp_default
   End If

   #if Eeprom_testmodus
      Print "Setting-Daten internes EEPROM:"
      Print "=============================="

      Temp_byte1 = Ee_data_version_intern
      Print "Version Datenstruktur :                              " ; Hex(temp_byte1)

      Temp_byte1 = Ea_data_debug
      Print "Debug-Modus (Flag) :                                 " ; Hex(temp_byte1)

      Temp_byte1 = Ea_data_lcd_helligkeit
      Print "LCD Helligkeit :                                     " ; Hex(temp_byte1)

      Temp_byte1 = Ea_data_lcd_kontrast
      Print "LCD Kontrast :                                       " ; Hex(temp_byte1)

      Temp_byte1 = Ea_data_lcd_saver
      Print "LCD Screen Saver :                                   " ; Hex(temp_byte1)

      Temp_byte1 = Ea_data_lcd_light
      Print "LCD Beleuchtung :                                    " ; Hex(temp_byte1)

      Temp_byte1 = Ea_data_wind
      Print "Zeit fuer Auswerteintervall Windgeschwindigkeit :    " ; Hex(temp_byte1)

      Temp_byte1 = Ea_data_temp
      Print "Zeit fuer Auswerteintervall Temp und Feuchtigkeit : " ; Hex(temp_byte1)

      Temp_byte1 = Ea_data_dimmer
      Print "Dimmer Helligkeit :                                  " ; Hex(temp_byte1)

      Temp_byte1 = Ea_data_dim_min
      Print "Dimmer Min-Helligkeit :                              " ; Hex(temp_byte1)

      Temp_byte1 = Ea_data_dim_max
      Print "Dimmer Max-Helligkeit :                              " ; Hex(temp_byte1)

      Temp_byte1 = Ea_data_conv_time
      Print "Motorabschaltung :                                   " ; Hex(temp_byte1)

      Temp_byte1 = Ea_data_conv_wind
      Print "Windgrenzwert :                                      " ; Hex(temp_byte1)

      Temp_byte1 = Ea_data_conv_temp
      Print "Temperaturgrenze :                                   " ; Hex(temp_byte1)

      Temp_byte1 = Ea_data_conv_light
      Print "Helligkeitsgrenze :                                  " ; Hex(temp_byte1)

      Temp_byte1 = Ea_data_conv_hotkey
      Print "Hotkey Kreuztasten :                                 " ; Hex(temp_byte1)
```

```
      Temp_byte1 = Ea_data_fan
      Print "Luefter ueber ATmega :                              " ; Hex(temp_byte1)

      Temp_byte1 = Ea_data_fan_temp
      Print "Luefter Temperatur :                                " ; Hex(temp_byte1)

      Temp_byte1 = Ea_data_fan_min_temp
      Print "Luefter untere Abschaltschwelle :                   " ; Hex(temp_byte1)

      Temp_byte1 = Ea_data_fan_max_temp
      Print "Luefter obere Einschaltschwelle :                   " ; Hex(temp_byte1)
   #endif

Return
'-- End Ee_init_internal_eeprom ------------------------------------------------


'--------------------------------------------------------------------------------
' GOSubroutine: Ee_init_external_eeprom
' Subroutine überprüft die Datenkonsitenz aufgrund des verwendeten Versions-
' Index und führt ggf. eine Grundinitialisierung der Datenstrukturen durch.
' Die für das Datenlogging notwendigen Basisdaten werden in globale Variablen
' geladen
'--------------------------------------------------------------------------------
Ee_init_external_eeprom:

   #if Eeprom_testmodus
      Print "Enter Ee_init_external_eeprom"
   #endif


   Ee_data_version_extern = Ee_read_eeprom_byte(ee_data_version_extern_adress)

   If Ee_data_version_extern <> Akt_external_ee_version Then

      #if Eeprom_testmodus
         Print "Datenversion stimmt nicht"
         Print "Initialisiere externes EEPROM neu!"
      #endif

      ' Datenstruktur externes EEPROM entspricht nicht dem aktuellen Stand
      ' => Datenstruktur neu anlegen bzw. neue Defaultwerte setzen.

      Call Ee_write_eeprom_byte(ee_data_version_extern_adress , Akt_external_ee_version)
      Waitms 10                                                 'Wartezeit für sicheres Schreiben
      Call Ee_write_eeprom_word(ee_data_count_adress , Ee_data_count_default)
      Waitms 10                                                 'Wartezeit für sicheres Schreiben
      Call Ee_write_eeprom_byte(ee_data_overrun_adress , Ee_data_overrun_default)
      Waitms 10                                                 'Wartezeit für sicheres Schreiben

   End If

   #if Eeprom_testmodus
      Print "Setting-Daten externes EEPROM:"
      Print "=============================="

      Ee_data_version_extern = Ee_read_eeprom_byte(ee_data_version_extern_adress)
      Print "Version Datenstruktur : " ; Hex(ee_data_version_extern)

      Ee_data_count = Ee_read_eeprom_word(ee_data_count_adress)
      Print "Datencounter  :          " ; Hex(ee_data_count)

      Ee_data_overrun = Ee_read_eeprom_byte(ee_data_overrun_adress)
      Print "Overrun-Zaehler :        " ; Hex(ee_data_overrun)
   #endif

Return
'-- End Ee_init_external_eeprom ------------------------------------------------


'--------------------------------------------------------------------------------
' GOSubroutine: Write_testdata_byte_to_eeprom
' Subroutine schreibt zum Test 4 Byte aus einem Array in das EEPROM.
'--------------------------------------------------------------------------------
Write_testdata_byte_to_eeprom:

   #if Eeprom_testmodus
      Print "Enter Write_testdata_byte_to_eeprom"
   #endif

   Temp_adress = &H0000

   For Temp_count = 1 To 4
      Call Ee_write_eeprom_byte(temp_adress , Testarray_byte(temp_count))
      Print "writing : " ; Testarray_byte(temp_count)
      Incr Temp_adress
   Next Count

Return
'-- End Write_testdata_byte_to_eeprom -------------------------------------------
```

```
'--------------------------------------------------------------------------------
' GOSubroutine: Write_testdata_byte_to_eeprom
' Subroutine liest zum Test 4 Bytes aus dem EEPROM aus und gibt diese über
' aus RS232.
'--------------------------------------------------------------------------------
Read_testdata_byte_from_eeprom:

   #if Eeprom_testmodus
      Print "Enter Read_testdata_byte_from_eeprom"
   #endif

   Temp_adress = &H0000

   For Temp_count = 1 To 4
      Temp_byte1 = Ee_read_eeprom_byte(temp_adress )
      Print "reading : " ; Temp_byte1
      Incr Temp_adress
   Next Temp_count

Return
'-- End Read_testdata_byte_from_eeprom ---------------------------------------


'--------------------------------------------------------------------------------
' GOSubroutine: Write_testdata_word_to_eeprom
' Subroutine schreibt zum Test 4 Byte aus einem Array in das EEPROM.
'--------------------------------------------------------------------------------
Write_testdata_word_to_eeprom:

   #if Eeprom_testmodus
      Print "Enter Write_testdata_word_to_eeprom"
   #endif

   Temp_adress = &H0000

   For Temp_count = 1 To 4
      Call Ee_write_eeprom_word(temp_adress , Testarray_word(temp_count))
      Print "writing : " ; Hex(testarray_word(temp_count))
      Incr Temp_adress                                    ' Adresse da Word zwei weiter!
      Incr Temp_adress
   Next Count

Return
'-- End Write_testdata_word_to_eeprom ---------------------------------------


'--------------------------------------------------------------------------------
' GOSubroutine: Write_testdata_word_to_eeprom
' Subroutine liest zum Test 4 Bytes aus dem EEPROM aus und gibt diese über
' aus RS232.
'--------------------------------------------------------------------------------
Read_testdata_word_from_eeprom:

   #if Eeprom_testmodus
      Print "Enter Read_testdata_word_from_eeprom"
   #endif

   Temp_adress = &H0000

   For Temp_count = 1 To 4
      Temp_word1 = Ee_read_eeprom_word(temp_adress)
      Print "reading : " ; Hex(temp_word1)
      Incr Temp_adress                                    ' Adresse da Word zwei weiter!
      Incr Temp_adress
   Next Temp_count

Return
'-- End Read_testdata_word_from_eeprom ---------------------------------------


'--------------------------------------------------------------------------------
' GOSubroutine: Write_und_read_gesamtes_eeprom
' Subroutine beschreibt das komplette EEPROM mit dem Low-Byte der Adresse und
' liest anschließend den kompletten Inhalt gegen.
' Wir ein Fehler erkannt, so wird die Schleifenbearbeitung abgebrochen und es
' wird eine Fehlermeldung über RS232 ausgegeben.
'--------------------------------------------------------------------------------
Write_und_read_gesamtes_eeprom:

   #if Eeprom_testmodus
      Print "Enter Write_und_read_gesamtes_eeprom"
      Print "Beschreibe Adress 0x0000 bis 0xFFFF"
   #endif

   ' Erstes Byte aus EEPROM lesen
   Temp_byte1 = Ee_read_eeprom_byte(&H0000)

   ' Entscheidung ob 0-255 oder 255-0 geschrieben/gelesen werden soll
```

```
    If Temp_byte1 = 0 Then
       ' Wert ist 0 also 255-0 schreiben
       Direction = 1                                          ' Top down
    Else
       ' Wert ist 255 also 0-255 schreiben
       Direction = 0                                          ' Bottom up
    End If

    ' ----- Daten schreiben -----
    For Temp_adress = 0 To 65535
       ' Daten zum Schreiben aus Adresse extrahieren
       Temp_byte1 = Low(temp_adress)

       ' Korrektur auf TopDown oder BottomUp
       If Direction = 0 Then
          Temp_byte1 = 255 - Temp_byte1
       End If

       ' Datenwert an Adresse in EEPROM schreiben
       Call Ee_write_eeprom_byte(temp_adress , Temp_byte1 )

       ' ggf. Trace ausgeben
       #if Eeprom_testmodus
          Print "WA" ; Hex(temp_adress) ; "D" ; Hex(temp_byte1)
       #endif

       ' 10ms warten (ohne Wartezeit Probleme bei schreiben)
       Waitms 10

    Next Temp_adress

    ' Eine Sekunde zwischen schreiben und lesen warten
    Wait 1

    #if Eeprom_testmodus
       Print "Lese und vergleiche Adresse 0x0000 bis 0xFFFF"
    #endif

    ' ----- Daten lesen -----
    For Temp_adress = 0 To 65535

       ' Daten zum Vergleiche aus Adresse extrahieren
       Temp_byte1 = Low(temp_adress)

       ' Korrektur auf TopDown oder BottomUp
       If Direction = 0 Then
          Temp_byte1 = 255 - Temp_byte1
       End If

       ' Datenwert von Adresse aus EEPROM auslesen
       Temp_byte2 = Ee_read_eeprom_byte(temp_adress)

       ' ggf. Trace ausgeben
       #if Eeprom_testmodus
          Print "RA" ; Hex(temp_adress) ; "D" ; Hex(temp_byte2)
       #endif

       ' Fehlerbehandlung falsch SOLL<>IST
       If Temp_byte1 <> Temp_byte2 Then
          Print "Error auf Adresse " ; Hex(temp_adress)
          Print "Datenwert SOLL : " ; Hex(temp_byte1)
          Print "Datenwert IST  : " ; Hex(temp_byte2)
          Exit For                                            ' FOR-Schleife abbrechen
       End If

       ' 10ms warten (ohne Wartezeit Probleme bei lesen)
       Waitms 10

    Next Temp_adress

    #if Eeprom_testmodus
       Print "*Ende Test*"
    #endif

Return
'-- End Write_und_read_gesamtes_eeprom ---------------------------------------


'--------------------------------------------------------------------------------
' GOSubroutine: Write_testdata_datalogging
' Subroutine schreibt die Datenstruktur zum Umwetldatenlogging in den Speicher
' Hierbei wird auf ein Set an testdaten zurückgegriffen.
'--------------------------------------------------------------------------------
Write_testdata_datalogging:

    #if Eeprom_testmodus
       Print "Enter Write_testdata_datalogging"
       Print "Datenversion : " ; Hex(ee_data_version_extern)
       Print "Data Count :   " ; Hex(ee_data_count)
       Print "Data Overrun : " ; Hex(ee_data_overrun)
```

```
   #endif

   ' Prüfen ob Datenüberlauf und ggf. Variable zurücksetzen und Overrun erhöhen
   If Ee_data_count = Ea_data_loging_endadress Then
      Ee_data_count = 0                                    ' Counter auf Anfangsadresse setzen
      Incr Ee_data_overrun                                 ' Overrun-Zähler erhöhen

      #if Eeprom_testmodus
         Print "Speicherüberlauf!"
         Print "Setzte Data Count auf " ; Hex(ee_data_count)
         Print "Erhöhen Overrun auf " ; Hex(ee_data_overrun)
      #endif

      ' Overrun-Zähler gleich speichern
      Call Ee_write_eeprom_byte(ee_data_overrun_adress , Ee_data_overrun)

   End If

   ' Adresse zum Schreiben berechnen
   Temp_word1 = Ea_data_loging_length * Ee_data_count
   Temp_word1 = Temp_word1 + Ee_data_loging_start_adress

   #if Eeprom_testmodus
      Print "Berechnete Speicheradresse : " ; Hex(temp_word1)
   #endif

   ' Anzahl gespeicherter Daten um ein erhöhen
   Incr Ee_data_count
   #if Eeprom_testmodus
      Print "Erhöhe Data Count auf " ; Hex(ee_data_count)
   #endif

   ' Datenstruktur zum Speichern vorbereiten

   ' Byte(1) Ee_data_hour
   Ee_data_logging_struct(1) = Test_data_hour
   ' Byte(2) Ee_data_minute
   Ee_data_logging_struct(2) = Test_data_minute
   ' Byte(3) Ee_data_day
   Ee_data_logging_struct(3) = Test_data_day
   ' Byte(4) Ee_data_month [Bit7..4]
   ' Byte(4) Ea_data_light [Bit3..0]
   Temp_byte1 = Test_data_month
   Shift Temp_byte1 , Left , 4
   Temp_byte1 = Temp_byte1 And &B11110000
   Temp_byte2 = Test_data_light And &B00001111
   Ee_data_logging_struct(4) = Temp_byte1 + Temp_byte2
   ' Byte(5) Ee_data_year
   Temp_word2 = Test_data_year - 2000
   Ee_data_logging_struct(5) = Low(temp_word2)
   ' Byte(6) Ee_data_rain  [Bit7]
   ' Byte(6) Ee_data_wind  [Bit6..0]
   Temp_byte1 = Test_data_wind And &B01111111
   If Test_data_rain = &H01 Then
      Set Temp_byte1.7                                     ' Setzte Bit 7 auf "1"
   End If
   Ee_data_logging_struct(6) = Temp_byte1
   ' Byte(7) Ee_data_temp
   Ee_data_logging_struct(7) = Low(test_data_temp)
   ' Byte(8) Ee_data_humidity
   Ee_data_logging_struct(8) = Test_data_humidity

   #if Eeprom_testmodus
      Print "Daten-Struct : " ;
      Print Hex(ee_data_logging_struct(1)) ; " " ;
      Print Hex(ee_data_logging_struct(2)) ; " " ;
      Print Hex(ee_data_logging_struct(3)) ; " " ;
      Print Hex(ee_data_logging_struct(4)) ; " " ;
      Print Hex(ee_data_logging_struct(5)) ; " " ;
      Print Hex(ee_data_logging_struct(6)) ; " " ;
      Print Hex(ee_data_logging_struct(7)) ; " " ;
      Print Hex(ee_data_logging_struct(8))
   #endif

   ' Datenstruktur speichern
   Call Ee_write_eeprom_struct(temp_word1)

   ' aktuellen Data Count speichern
   Call Ee_write_eeprom_word(ee_data_count_adress , Ee_data_count)


Return
'-- End Write_testdata_datalogging ---------------------------------------


'-------------------------------------------------------------------------------
' GOSubroutine: Read_testdata_datalogging
' Subroutine liest die Datenstrkturen zu den Umweltdatenloggings aus und gibt
' sie auf der RS-232 als vorformatierter Datenstrom aus welche auch in Excel
' importiert werden kann.
```

```
'--------------------------------------------------------------------------------
Read_testdata_datalogging:

   #if Eeprom_testmodus
      Print "Enter Read_testdata_datalogging"
      Print "Datenversion : " ; Hex(ee_data_version_extern)
      Print "Data Count :   " ; Hex(ee_data_count)
      Print "Data Overrun : " ; Hex(ee_data_overrun)
   #endif

   ' Prüfen aufÜberlauf, wenn ja maximale Adresse annehmen
   If Ee_data_overrun > 0 Then
      Temp_word3 = Ea_data_loging_endadress - 1
   Else
      Temp_word3 = Ee_data_count
   End If

   ' Prüfen ob überhaupt Daten im EEPROM abgelegt und wenn ja ausgeben
   If Temp_word3 > 0 Then

      ' Schleife über alle vorhandenen Datenpakete
      For Temp_word1 = 0 To Temp_word3

         ' Adresse zum Lesen berechnen
         Temp_word2 = Ea_data_loging_length * Temp_word1
         Temp_word2 = Temp_word2 + Ee_data_loging_start_adress

         ' Indizierte Datenstruktur aus EEPROM lesen
         Call Ee_read_eeprom_struct(temp_word2)

         #if Eeprom_testmodus
            Print "Lese von Adresse " ; Hex(temp_word2)
            Print "Daten-Struct : " ;
            Print Hex(ee_data_logging_struct(1)) ; " " ;
            Print Hex(ee_data_logging_struct(2)) ; " " ;
            Print Hex(ee_data_logging_struct(3)) ; " " ;
            Print Hex(ee_data_logging_struct(4)) ; " " ;
            Print Hex(ee_data_logging_struct(5)) ; " " ;
            Print Hex(ee_data_logging_struct(6)) ; " " ;
            Print Hex(ee_data_logging_struct(7)) ; " " ;
            Print Hex(ee_data_logging_struct(8))
         #endif

         ' Gelesene Daten interpretieren und ausgeben
         ' Format: nr;day.month.year;hour:minute;light;rain;wind;temp;hum
         Temp_word2 = Temp_word1 + 1
         Print Temp_word2 ; ";" ;
         ' Byte(3) Ee_data_day
         Print Ee_data_logging_struct(3) ; "." ;
         ' Byte(4) Ee_data_month [Bit7..4]
         Temp_byte1 = Ee_data_logging_struct(4)
         Shift Temp_byte1 , Right , 4
         Temp_byte1 = Temp_byte1 And &B00001111
         Print Temp_byte1 ; "." ;
         ' Byte(5) Ee_data_year
         Temp_word2 = Ee_data_logging_struct(5) + 2000
         Print Temp_word2 ; ";" ;
         ' Byte(1) Ee_data_hour
         Print Ee_data_logging_struct(1) ; ":" ;
         ' Byte(2) Ee_data_minute
         Print Ee_data_logging_struct(2) ; ";" ;
         ' Byte(4) Ea_data_light [Bit3..0]
         Temp_byte1 = Ee_data_logging_struct(4) And &B00001111
         Print Temp_byte1 ; ";" ;
         ' Byte(6) Ee_data_rain  [Bit7]
         Temp_byte1 = Ee_data_logging_struct(6)
         Shift Temp_byte1 , Right , 7
         Temp_byte1 = Temp_byte1 And &B00000001
         Print Temp_byte1 ; ";" ;
         ' Byte(6) Ee_data_wind  [Bit6..0]
         Temp_byte1 = Ee_data_logging_struct(6) And &B01111111
         Print Temp_byte1 ; ";" ;
         ' Byte(7) Ee_data_temp
         Temp_byte1 = Ee_data_logging_struct(7)
         If Temp_byte1 > 127 Then
            Temp_word2 = Temp_byte1 + &HFF00
         Else
            Temp_word2 = Temp_byte1
         End If
         Temp_int1 = Temp_word2
         Print Temp_int1 ; ";" ;
         ' Byte(8) Ee_data_humidity
         Print Ee_data_logging_struct(8)
      Next Temp_word1
   End If

Return
'-- End Read_testdata_dataloggin ------------------------------------------------

End
```

```
'#################################### END ###################################
```

Software 5: Code zur Ansteuerung des EEPROM (extern und intern)

8.10.6 Prototyp externes EEPROM 24C512

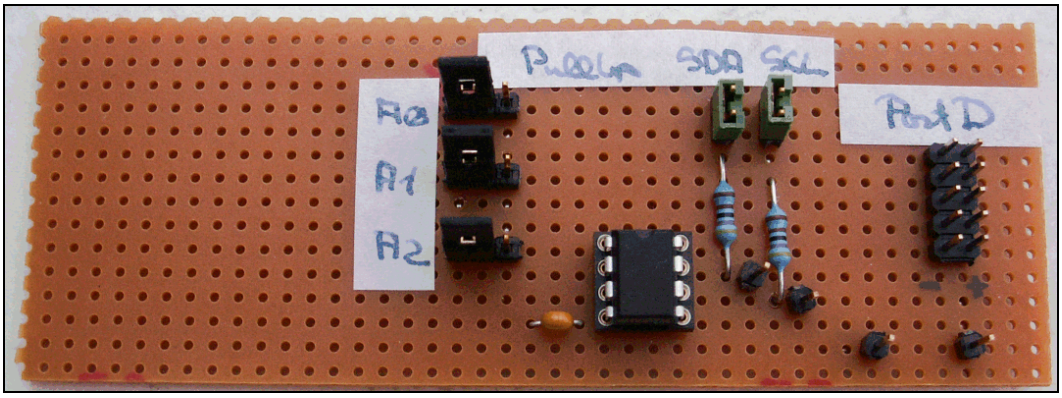


Abbildung 77: Prototyp EEPROM Ansteuerung

8.11 Sensorik für Temperatur und Luftfeuchtigkeit

8.11.1 Allgemeines zur Temperatur- und Feuchtigkeitsmessung



Zur Messwerterfassung von Temperatur und Luftfeuchtigkeit wird der Sensor SHT71 der Firma Sensirion eingesetzt.
Der Sensor ist ein Präzisionsfeuchtesensor SHT71: Digital, steckbar & vollständig kalibriert!

Abbildung 78: Sensor für Luftfeuchtigkeit und Temperatur

Der digitale SHT71 Feuchtesensor ist vollständig kalibriert und dank steckbarem Pin-Type Package (für 1.27mm Sockets) einfach zu montieren. Mit individueller Kalibration und digitalem Output (CMOSens® Technologie) ist der SHT71 vollständig austauschbar. Das erspart die mühsame und aufwändige Rekalibrierung im Falle eines Sensorwechsels/-austauschs.

Der Sensorkopf ist nur durch eine dünne Brücke mit den Pins verbunden und dadurch optimal thermisch isoliert, was nochmals für verbesserte Ansprechzeiten sorgt. Die Pins sind vergoldet und damit sehr korrosionsbeständig.

Leistungsmerkmale:

- 2 Sensoren für relative Feuchte und Temperatur
- Präzise Taupunktberechnung möglich
- Messbereich: 0-100% r.F.
- Absolute Genauigkeit (r.F.): +/- 3% r.F.
- Genauigkeit Temp.sensor: +/- 0.4°C @ 25 °C
- Kalibriert und digital (2-Wire Schnittstelle)
- Schnelle Ansprechzeit (r.F.): < 4 sec.

- Tiefer Stromverbrauch (typ. 30 µW)
- niedrige Kosten
- Präziser High-End Sensor, individuell kalibriert in Präzisionsfeuchtekammer (Taupunktspiegel)
- 100%ig austauschbar ohne Rekalibrierung!
- Führende (CMOSens® Technologie für überlegene Langzeitstabilität
- Einfachste Handhabung dank Kalibration und digitaler 2-Wire Schnittstelle preisgünstig

Interface:

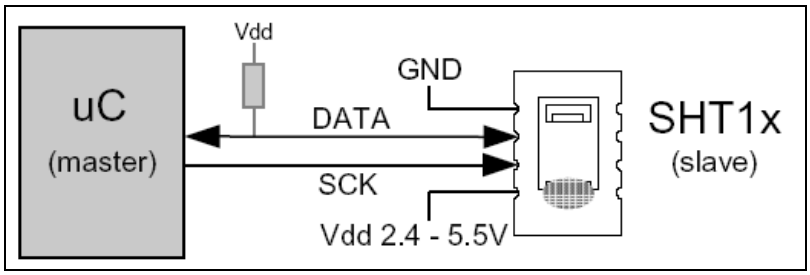


Abbildung 79: two wire interface für SHT1x / SHT7x

Pin	Name	Comment
1	SCK	Serial clock input
2	VDD	Supply
3	GND	Ground
4	DATA	Serial data bidirectional

Tabelle 97: Stückliste externes EEPROM

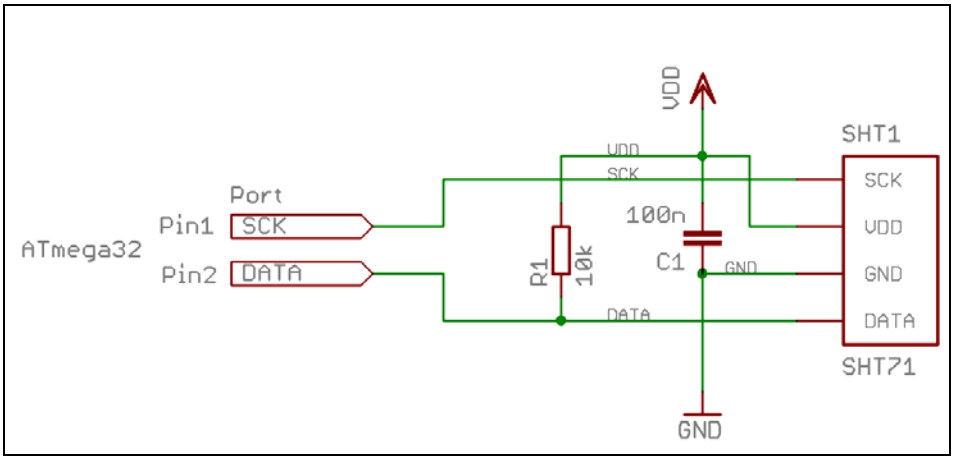
8.11.2 Beschaltung des Sensors SHT71

Der Sensor verfügt zwar – genaus so wie das EEPROM und der ATmega – über ein „two wire interface“. Er kann aber nicht gleichzeitig mit anderen I²C-Komponenten am TWI-Bus des ATmega betrieben werden.

Der SHT erfüllt nicht die I²C Spezifikation und wird anders initialisiert!!

Da es sich aber dennoch um einen sehr leistungsfähigen Baustein handelt wird er in diesem Projekt verwendet und über zwei separate Leitungen am ATmega angebunden.

Beschaltung:



Schaltbild 31: Beschaltung des Temperatur- und Luftfeuchtigkeitssensors SHT71

Bauteile:

Stückliste: Externes Temperatur- und Luftfeuchtigkeitssensor			
Widerstände		*Halbleiter*	
R1	10k Ω	SHT1	Sensor SHT71
Kondensator			
C1	100nF oder 10nF Keramikkondensator		

Tabelle 98: Stückliste Temperatur- und Luftfeuchtigkeitssensor

Das Datenblatt zum SHT71 schreibt einen 100 nF Kondensator als Abblockkondensator vor. Erfahrungen anderer Benutzer in Foren im Internet beschreiben aber die Verwendung eines 10 nF Keramik-Kondensators um das Springen von Messwerten und verhindern.

Zur Beschaltung des SHT71 ist auch der PullUp-Widerstand zwingend für eine saubere Funktion notwendig. Der ATmega interne PullUp reicht für die Implementierung nicht aus!

Originalinformation:

> "The DATA line of the SHT sensor needs an external pull-up resistor of 10 kohms to work properly, if you're using the ATmega's internal pull-up resistor, wrong values are read.
> Also a 0.01 uF decoupling capacitor from VCC to GND has to be placed as close as possible to the sensor."

Ressourcenzuordnung zum ATmega128:

Nummer	*Schaltbild*		*Ressource ATmega128*		
1	DATA	[TEMP_DATA]	PortD.6	[PD6]	(GPIO)
2	SCK	[TEMP_SCK]	PortD.7	[PD7]	(GPIO)

Tabelle 99: Ressourcenzuordnung ATmega128 für EEPROM

Oszillogramme der Kommunikation mit dem SHT71:

Im Folgenden sind einige typische Kommunikationsabläufe zwischen ATmega und SHT71 abgebildet.

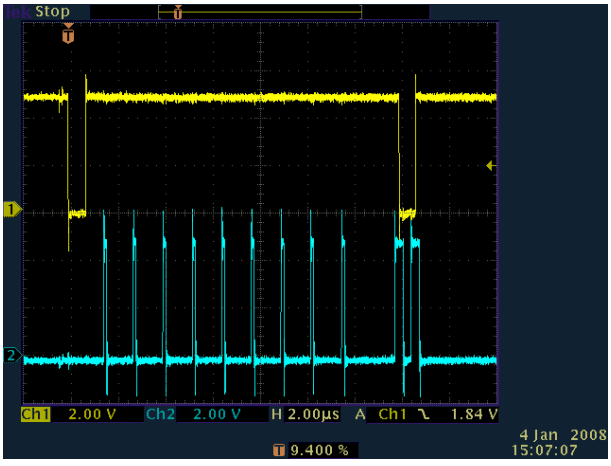


Abbildung 80: Oszillogramm SHT71 „connection reset sequence"

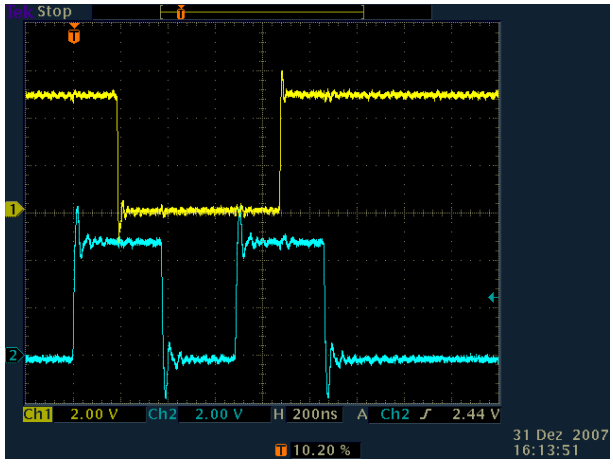


Abbildung 81: Oszillogramm SHT71 „transmission start"

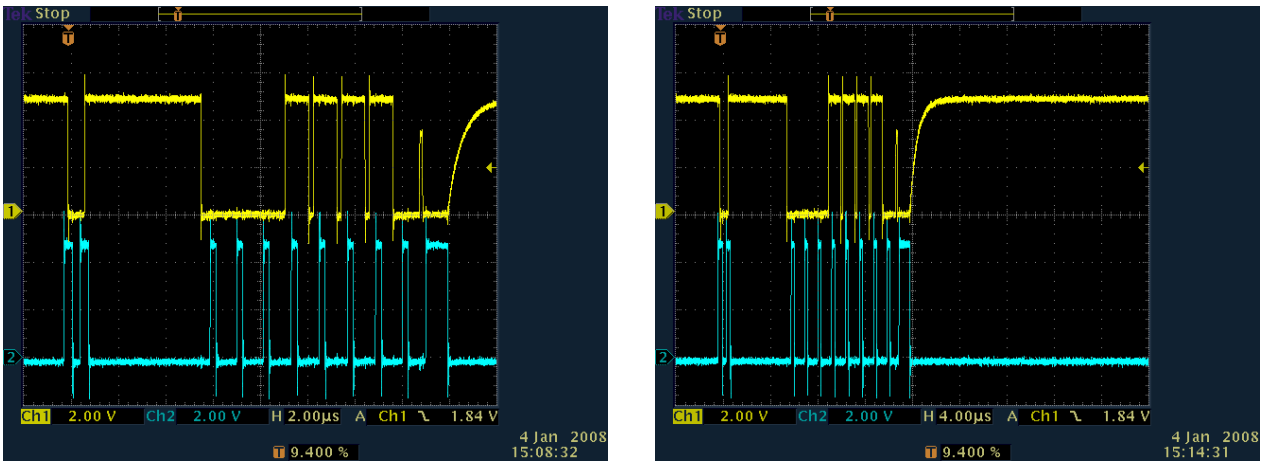


Abbildung 82: SHT71 „transmission start“ und Byte-Wert (Reset) senden

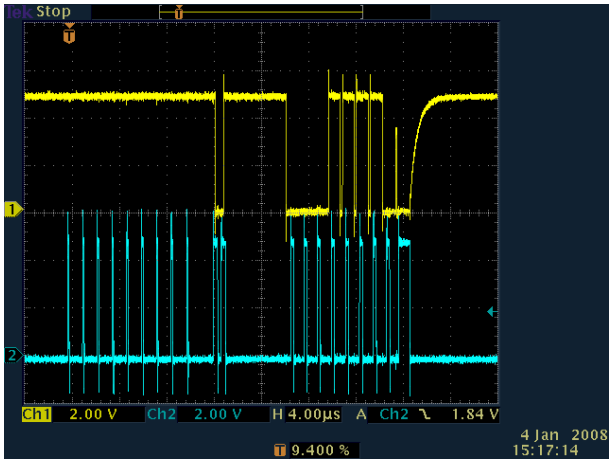


Abbildung 83: SHT71 Softreset

Der Softreser besteht aus:

- „connection reset sequence“
- „write reset“
- „connection reset sequence“

Die „connection reset sequence“ besteht aus:

- 9 x toggeln der SCK-Leitung bei DATA = 1
- „transmission start sequence“

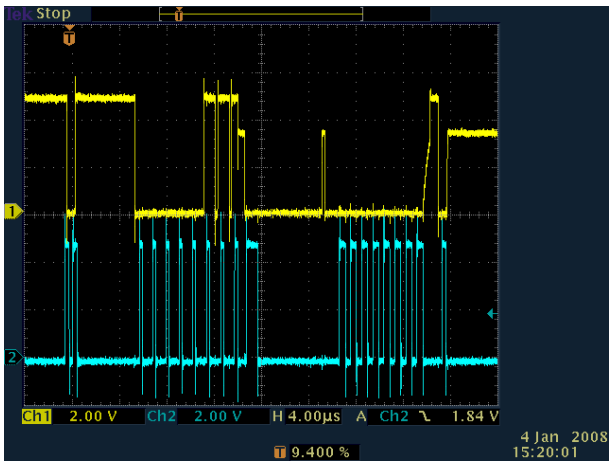


Abbildung 84: SHT71 Statusregister lesen

Es wird das Datenbyte &B00000000 gelesen (enstspricht dem SHT71 Default)

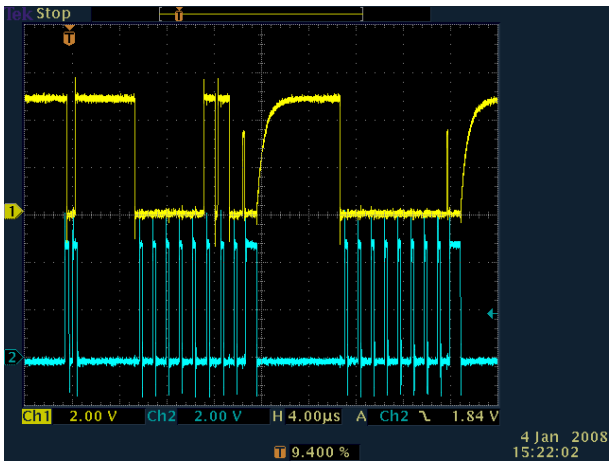


Abbildung 85: SHT71 Statusregister schreiben

Es wird das Datenbyte &B00000000 geschrieben (enstpsricht dem SHT71 Default)

Bemerkung:

GELB (GELB) dargestellt ist die Datenleitung DATA
CYAN (CYAN) dargestellt ist die Clockleitung SCK

8.11.3 Wichtige Informationen und Notizen zum SHT71

Nach einem Power-Up benötigt der Sensor 11 ms um den „sleep“ Mode zu erreichen. Während dieser Zeit darf kein Kommando an den SHT71 gesendet werden. Erst im Sleep-Mode kann mit dem Sensor kommuniziert werden.

Die State machine des Sensors sieht wie folgt aus:

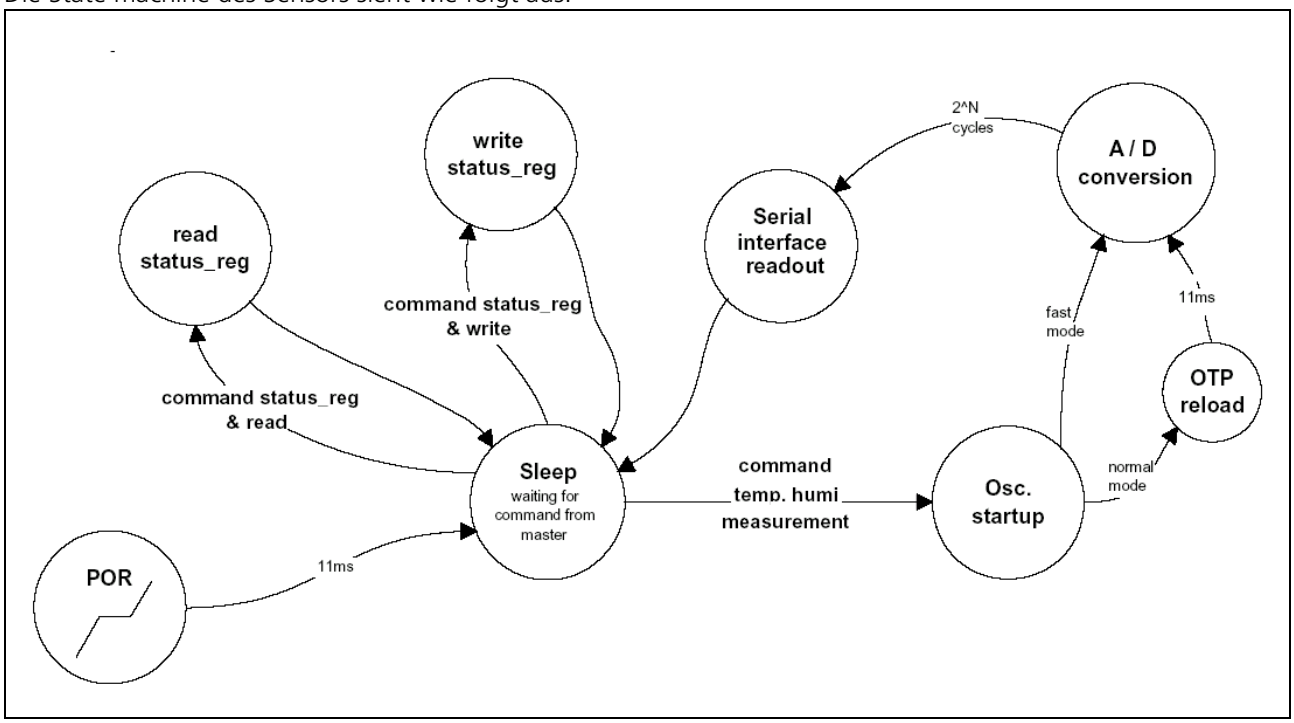


Abbildung 86: Die State machine des SHT71

Das Interface zum SHT71 ist speziell für den Sensor optimiert und nicht zum normalen I²C-Interface kompatibel.

DATA ändert sicht nach fallenden Flanken und die Daten werden bei steigenden Flanken des serial clock SCK übernommen.

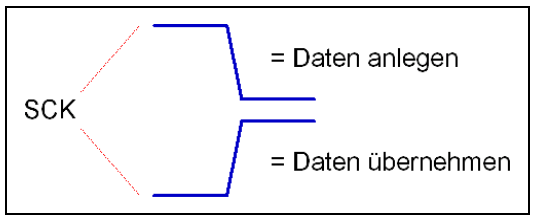


Abbildung 87: Serial data (DATA) des SHT71

Das Senden von Daten an den SHT71 muss durch eine „Transmission Start Sequence“ initiiert werden.

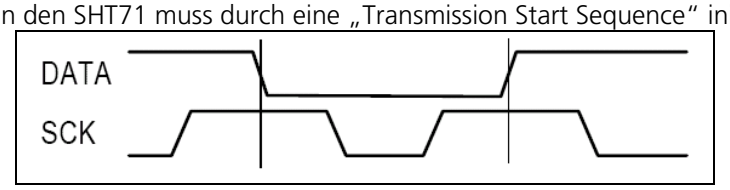


Abbildung 88: Transmission Sequence Start des SHT71

Im Kommando-Byte des SHT71 sind drei Bit (Bit 7..5) für die Adressierung des Sensors im Bus vorgesehen. Die Adresse ist jedoch auf „000“ fixiert und kann nicht geändert werden.

Der SHT71 verfügt über die folgenden Kommandos mit denen er gesteuert werden kann:

Command	Code
Reserved	0000x
Measure Temperature	**00011**
Measure Humidity	**00101**
Read Status Register	00111
Write Status Register	00110
Reserved	0101x-1110x
Soft reset, resets the interface, clears the status register to default values wait minimum 11 ms before next command	**11110**

Tabelle 100: Kommandos-Codes des SHT71

Ein Kommando zum SHT71 setzt sicht aus Adresse (Bit7..5) und dem Command-Code (Bit4..0) zusammen. Zur Übertragung des Bytes werden immer 8 Takte des SCK inkl. eonmes weiteren Acknowledge-Taktes (also 9 SCK-Takte) benötigt. Mit dem Acknowledge-Bit signalisert der Empfänger eines Bytes das korrekte Empfangen.

Für die Messung von Temperatur und Luftfeuchtigkeit benötigt der SHT71 je nach Auflösung 11 / 55 / 210 ms für 8 / 12 / 14 Bit Nessergebnis. Diese Zeit muss beim Auslesen des Messergebnisses nach Absenden des Kommandos gewartet werden. Sie kann aber im Bereich +- 15% schwanken.

Der Zustand „Data ready“ wird durch ein Low-Signal auf der Datenleitung DATA vom SHT71 angezeigt. Danach kann das Messergebnis (2 Byte) inkl. Checksumme ausgelsen werden. Soll die Checksumme nicht verwendet werde so kann das ACK ncht quittiert werden. Der SHT71 beendet die datenübertragung und geht in den Zustand „Sleep“ über.

Um den Sensor vor Eigenerwärmung zu schützen welche das Messergebnis verfälscht darf der SHT71 nur 10% der Betriebszeit aktiv sein. Dies bedeutet man sollte nicht mehr als 2 Messungen pro Sekunde bei einer Genauigkeit von 12 Bit durchführen.

Die gesamte digitale Datenübertragung wird durch eine CRC-8 Checksumme abgesichert welche auch im Projekt verwendet wird.

Bei der Datenübertragung vom SHT71 zum ATmega wird immer zuerst das MSByte dan das LSByte gesendet.

Eine „connection reset sequence“ besteht immer einer DATA high sequence bei der SCK mindestens 9 mal high-low getoggelt wird. Danach erfolgt eine Transmission Start Sequence.

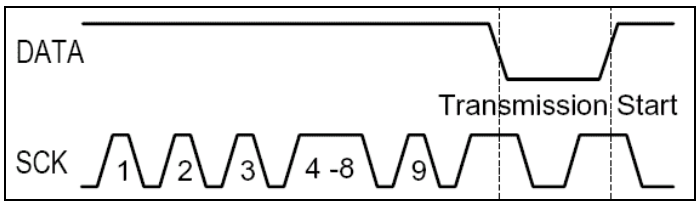


Abbildung 89: Connection Reset Sequence des SHT71

Eine Messsequence beim SHT71 sieht wie folgt aus:

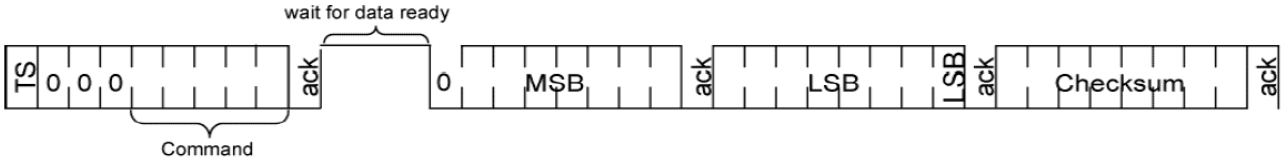


Abbildung 90: Mess-Sequence des SHT71

Das Schreiben und Lesen des Statusregisters sieht wie folgt aus:

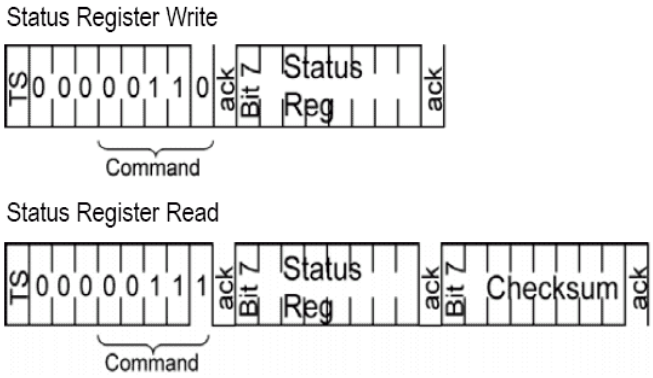


Abbildung 91: Statusregister des SHT71 schreiben und lesen

Das Status-Reister des SHT71:

Bit	Type	Description	Default	
7		reserved	0	
6	R	End of Battery (low voltage detection) '0' for Vdd > 2.47 '1' for Vdd < 2.47	X	No default value, bit is only updated after a measurement
5		reserved	0	
4		reserved	0	
3		For Testing only, do not use	0	
2	R/W	Heater	0	off
1	R/W	no reload from OTP	0	reload
0	R/W	'1' = 8bit RH / 12bit Temperature resolution '0' = 12bit RH / 14bit Temperature resolution	0	12bit RH 14bit Temp.

Tabelle 101: Statusregisterinhalte des SHT71

8.11.4 BASCOM Treiber und Beispielcode für den SHT71

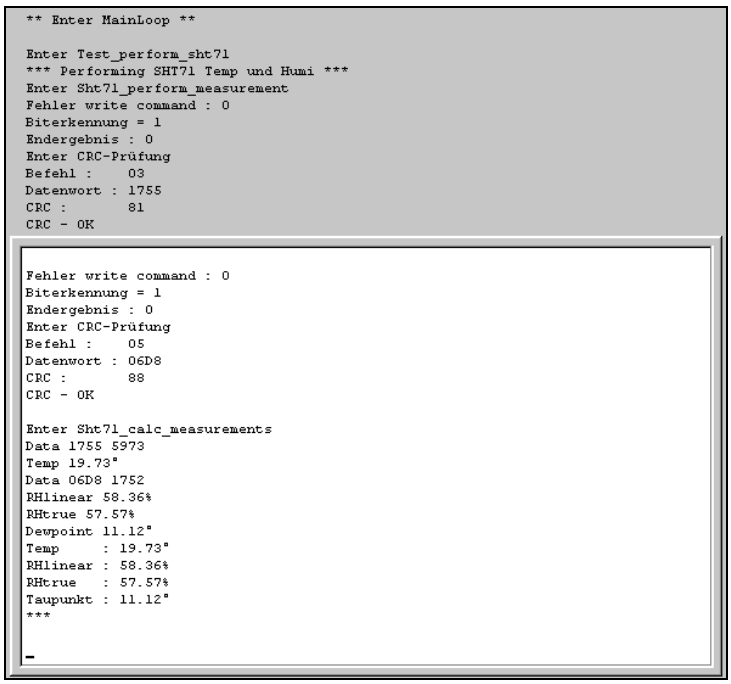


Abbildung 92: Traceausgabe SHT71 des BASCOM-Beispielcodes

```
'#################################################################################
' SHT71.BAS:                                                      Stand 04.01.2008
' --------------------                                            (C) Markus Fulde
'
' Testprogramm zur Inbetriebnahme des Feuchtigkeits- und Temperatursensor SHT71
' der Firma Sensirion.
' Das Testprogramm stellt alle notwendigen Routinen für den Zugriff auf den
' Sensor zur Verfügung. Es orientiert sich bei der Implementierung ab den
' Applikation Notes von Sensirion und stellt eine Umsetzung des Basiscodes auf
' BASCOM dar.
'#################################################################################


'--------------------------------------------------------------------------------
' Compilerinstruktionen und Compilerdirektiven
'--------------------------------------------------------------------------------
$regfile = "m128def.dat"
$crystal = 16000000
$hwstack = 64
$swstack = 64
$framesize = 80

$baud = 38400

'--------------------------------------------------------------------------------
' Definition von Ressourcen
'--------------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========

' Ressourcen für Zugriff
Sht71_sck Alias Portd.7                                 'Sck PortD.7
Sht71_dataout Alias Portd.6                             'Data PortD.6 Output
Sht71_datain Alias Pind.6                               'Data PortD.6 Input

' Ressourcen für Konfiguration
Temp_sck Alias Pind.7                                   'Sck PortD.7
Temp_data Alias Pind.6                                  'Data PortD.6


'========== Für Testumgebung  ==========

' Alias Namen für Tastaturabfragen
Taste_1 Alias Pina.0
Taste_2 Alias Pina.1
Taste_3 Alias Pina.2
Taste_4 Alias Pina.3
Taste_5 Alias Pina.4
Taste_6 Alias Pina.5
Taste_7 Alias Pina.6
Taste_8 Alias Pina.7

' Alias Namen für LED's
Led0 Alias Portb.0
```

```
Led1 Alias Portb.1
Led2 Alias Portb.2
Led3 Alias Portb.3
Led4 Alias Portb.4
Led5 Alias Portb.5
Led6 Alias Portb.6
Led7 Alias Portb.7

'-----------------------------------------------------------------------------
' Definition von Variablen, Konstaten und Datentypen
'-----------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========

' globale Hilfsvariablen
Dim Sht71_transceive_struct(3) As Byte
Dim Sht71_temp_byte1 As Byte
Dim Sht71_humi_word As Word
Dim Sht71_temp_word As Word
Dim Sht71_checksum As Byte
Dim Sht71_error_result As Byte

' globale Variable für Temperatur, Luftfeuchtigkeit und Taupunkt
Dim Sht71_rhlinear As Single                                ' Luftfeuchtigkeit linear
Dim Sht71_rhtrue As Single                                  ' Luftfeuchtigkeit mit Korrektur (real)
Dim Sht71_dewpoint As Single                                ' Taupunkt
Dim Sht71_t As Single                                       ' Temperatur

' Kommandocodes für SHT71-Befehle                             adr command r/w
Const Sht71_status_reg_write = &H06                         ' 000  0011    0
Const Sht71_status_reg_read = &H07                          ' 000  0011    1
Const Sht71_measure_temp = &H03                             ' 000  0001    1
Const Sht71_measure_humi = &H05                             ' 000  0010    1
Const Sht71_reset = &H1E                                    ' 000  1111    0


' Konstanten für die Berechnung von Luftfeuchtigkeit und Temperatur

' Konstante für Berechung von RHlinear
Const Sht71_c1 = -4                                         ' Parameter C1 für 12 Bit
Const Sht71_c2 = 0.0405                                     ' Parameter C2 für 12 Bit
Const Sht71_c3 = -0.0000028                                 ' Parameter C3 für 12 Bit
' Const Sht71_c1 = -4                                        ' Parameter C1 für 8 Bit
' Const Sht71_c2 = 0.648                                     ' Parameter C2 für 8 Bit
' Const Sht71_c3 = -0.00072                                  ' Parameter C3 für 8 Bit

' Konstanten für Temperaturberechnung
Const Sht71_d1 = -40.00                                     ' Parameter D1 für 5V und °C
'Const Sht71_d1 = -39.75                                    ' Parameter D1 für 4V und °C
'Const Sht71_d1 = -39.66                                    ' Parameter D1 für 3.5V und °C
'Const Sht71_d1 = -39.60                                    ' Parameter D1 für 3V und °C
'Const Sht71_d1 = -39.55                                    ' Parameter D1 für 2.5V und °C
Const Sht71_d2 = 0.01                                       ' Parameter D2 für 14 Bit und °C
'Const Sht71_d2 = 0.04                                      ' Parameter D2 für 12 Bit und °C

' Konstanten für Berechung von RHtrue
Const Sht71_t1 = 0.01                                       ' Parameter T1 für 8 und 12 Bit
Const Sht71_t2 = 0.00008                                    ' Parameter T2 für 12 Bit
'Const Sht71_t2 = 0.00008                                   ' Parameter T2 für 8 Bit

' Konstanten zur Berechnung des Taupunktes
Const Sht71_tn = 243.12
Const Sht71_m = 17.62
Const Sht71_alpha = 0.4343


' Sonstige Konstanten
Const Sht71_testmodus = 1                                   ' Für Trace-Output über RS232

Const Sht71_acknowledge = 1                                 ' Positive Quittierung = Weiter
Const Sht71_noacknowledge = 0                               ' Negative Quittierung = Ende

Const Sht71_modus_humi = 1                                  ' Parameter für measure Modus
Const Sht71_modus_temp = 2                                  ' Parameter für measure Modus

Const Sht71_communication_wait = 11                         ' 11 ms nach PowerUp/Reset warten

Const Sht71_response_wait = &H0004FFFF                      ' bei ATmega128 mit 16Mhz ca. 1,5 sec.

Const True = 1                                              ' Boolsche Variable
Const False = 0

Const Crc_ok = 0                                            ' CRC-Ergebnisse
Const Crc_nok = 1


'========== Für Testumgebung ==========

' globale Varaible für Testroutinen
Dim Temp_byte1 As Byte                                      ' Temporäre Varaible 1
```

```
Dim Temp_byte2 As Byte                                      ' Temporäre Varaible 2
Dim Temp_byte3 As Byte                                      ' Temporäre Varaible 3

Const Main_testmodus = 1                                    ' Für Trace-Output über RS232

Dim Temp_string1 As String * 20                             ' Ergebissstring Output


'-------------------------------------------------------------------------------
' Prototyping
'-------------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========

' ----- SUB-Routinen ----
Declare Sub Sht71_calc_measurements()

' ----- Funktionen -----
Declare Function Sht71_write_byte(byval Value As Byte) As Byte
Declare Function Sht71_read_byte(byval Acknowledge As Byte) As Byte
Declare Function Sht71_softreset() As Byte
Declare Function Sht71_read_statusregister(byref P_value As Byte , Byref P_checksum As Byte) As Byte
Declare Function Sht71_write_statusregister(byref P_value As Byte) As Byte
Declare Function Sht71_measure(byref P_value As Word , Byref P_checksum As Byte , Byval Modus As
Byte) As Byte
Declare Function Sht71_crc_check(byval Value As Word , Byval Crc As Byte , Byval Modus As Byte) As
Byte


'========== Für Testumgebung ==========


'-------------------------------------------------------------------------------
' Konfiguration und Basiseinstellungen (Projekt und Testumgebung)
'-------------------------------------------------------------------------------

'========== Für Testumgebung ==========

'Konfiguration des Port A als Eingang für Tastaturabfrage
Config Porta = Input

'Konfiguration des Port B als Ausgang für Statusanzeige via LED
Config Portb = Output
Portb = &B11111111


'========== Für Projekt ConvCtrl ==========
Config Temp_sck = Output
Config Temp_data = Output

Ddrd = &HFE

' Grundkonfiguration
Gosub Sht71_connection_reset_sequence


'========== Für Testumgebung ==========

#if Main_testmodus
   Print "** Enter MainLoop **"
   Print
#endif


'###############################################################################
' Hauptprogramm
'###############################################################################
'Hier ist die Programmhauptschleife
Do
   ' ===== Tasten EEPROM-Testmenü =====

   ' Tastaturabfrage Taste 2
   Debounce Taste_1 , 0 , Test_write_byte_to_sht71 , Sub

   ' Tastaturabfrage Taste 3
   Debounce Taste_2 , 0 , Test_perform_softreset , Sub

   ' Tastaturabfrage Taste 4
   Debounce Taste_3 , 0 , Test_statusregister_lesen , Sub

   ' Tastaturabfrage Taste 5
   Debounce Taste_4 , 0 , Test_statusregister_schreiben , Sub

   ' Tastaturabfrage Taste 6
   Debounce Taste_5 , 0 , Test_temp_und_humi_messung , Sub

   ' Tastaturabfrage Taste 7
   Debounce Taste_6 , 0 , Sht71_perform_measurement , Sub

   ' Tastaturabfrage Taste 8
```

```
   Debounce Taste_7 , 0 , Test_perform_sht71 , Sub

Loop
'## End Hauptprogramm ##############################################################


'********************************************************************************
' Interruptroutinen
'********************************************************************************

' Interrupts gibt es für den SHT71 keine!


'********************************************************************************
' Subroutinen
'********************************************************************************

'-------------------------------------------------------------------------------
' SHT71 - Subroutine: Sht71_calc_measurements
' Subroutine kalkuliert aus den Rohdaten in den globalen Variablen die tat-
' sächlichen Messwerte und legt diese erneut in globalen Enddaten-Variablen ab.
'
' Parameter:    keine
' Rückgabewert: keine
'-------------------------------------------------------------------------------
Sub Sht71_calc_measurements()

   ' ----- locale Variablen -----
   Local Local_resultstring As String * 20

   Local Local_temp_single1 As Single
   Local Local_temp_single2 As Single
   Local Local_temp_single3 As Single

   ' ----- Programmcode -----

   #if Sht71_testmodus = 1
      Print "Enter Sht71_calc_measurements"
   #endif


   ' ----- Berechnung der Temperatur T -----

   ' Formula temperature:
   ' Temperature = d1 + d2 * SOt
   Local_temp_single1 = Sht71_d2 * Sht71_temp_word
   Local_temp_single1 = Local_temp_single1 + Sht71_d1

   #if Sht71_testmodus = 1
      Local_resultstring = Fusing(local_temp_single1 , "###.##")
      Print "Data " ; Hex(sht71_temp_word) ; " " ; Sht71_temp_word
      Print "Temp " ; Local_resultstring ; "°"
   #endif

   ' gerechneter Temperaturwert an globale Variable übergeben
   Sht71_t = Local_temp_single1


   ' ----- Berechnung der Feuchtigkeit RHlinear -----

   ' Formula relative humidity:
   ' RHlinear = c1 + c2 * SOrh + c3 * SOrh^2

   Local_temp_single1 = Sht71_c2 * Sht71_humi_word

   Local_temp_single2 = Sht71_humi_word * Sht71_humi_word
   Local_temp_single2 = Sht71_c3 * Local_temp_single2

   Local_temp_single3 = Sht71_c1 + Local_temp_single1
   Local_temp_single3 = Local_temp_single3 + Local_temp_single2

   #if Sht71_testmodus = 1
      Local_resultstring = Fusing(local_temp_single3 , "###.##")
      Print "Data " ; Hex(sht71_humi_word) ; " " ; Sht71_humi_word
      Print "RHlinear " ; Local_resultstring ; "%"
   #endif

   ' gerechneter Feuchtigkeitswert RHlinear an globale Variable übergeben
   Sht71_rhlinear = Local_temp_single3


   ' ----- Berechnung der Feuchtigkeit RHtrue -----

   ' Formula humidity / temperature compensation:
   ' RHtrue = (Temperature - 25) * (t1 + t2 * SOrh) + RHlinear
   Local_temp_single1 = Sht71_t - 25

   Local_temp_single2 = Sht71_t2 * Sht71_humi_word
   Local_temp_single2 = Local_temp_single2 + Sht71_t1
```

```
   Local_temp_single3 = Local_temp_single1 * Local_temp_single2
   Local_temp_single3 = Local_temp_single3 + Sht71_rhlinear

   #if Sht71_testmodus = 1
      Local_resultstring = Fusing(local_temp_single3 , "###.##")
      Print "RHtrue " ; Local_resultstring ; "%"
   #endif

   ' gerechneter Feuchtigkeitswert RHtrue an globale Variable übergeben
   Sht71_rhtrue = Local_temp_single3


   ' ----- Berechnung des Taupunktes -----

   ' Formulas for Dew Point calculation:
   ' H = (log10(RHtrue) - 2) / alpha + (m * Temperature) / (tn + Temperature)
   ' Dp = tn * H / (m - H) // this is the dew point in Celsius
   Local_temp_single1 = Log10(sht71_rhtrue)
   Local_temp_single1 = Local_temp_single1 - 2
   Local_temp_single1 = Local_temp_single1 / Sht71_alpha

   Local_temp_single2 = Sht71_m * Sht71_t
   Local_temp_single3 = Sht71_tn + Sht71_t
   Local_temp_single2 = Local_temp_single2 / Local_temp_single3

   Local_temp_single3 = Local_temp_single1 + Local_temp_single2

   Local_temp_single1 = Sht71_tn * Local_temp_single3
   Local_temp_single2 = Sht71_m - Local_temp_single3

   Local_temp_single3 = Local_temp_single1 / Local_temp_single2

   #if Sht71_testmodus = 1
      Local_resultstring = Fusing(local_temp_single3 , "###.##")
      Print "Dewpoint " ; Local_resultstring ; "°"
   #endif

   ' gerechneter Taupunkt an globale Variable übergeben
   Sht71_dewpoint = Local_temp_single3

End Sub
'-- End Sht71_calc_measurements ------------------------------------------------


'*****************************************************************************
' Funktionen
'*****************************************************************************

'-----------------------------------------------------------------------------
' SHT71 - Function: Sht71_write_byte
' Funktion schreibt ein Byte auf den Sensibus und überpruft das Acknowledge
'
' Parameter:    Value = Byte welches zum SHT71 gesendet werden soll
' Rückgabewert: Rückgabewert error = 1 im Fall das kein Acknowledge gesendet
'-----------------------------------------------------------------------------
Function Sht71_write_byte(byval Value As Byte) As Byte

   ' ----- locale Variable -----
   Local Error As Byte

   ' ----- Programmcode -----

   ' Sicherstellen das Pin DATA auf Output
   Config Sht71_datain = Output

   ' Datenbyte senden
   ' MSB shifted out first when clock goes high
   Shiftout Sht71_dataout , Sht71_sck , Value , 1

   ' Release DATA-line
   Set Sht71_dataout

   ' Data auf Eingabe umschalten
   Config Sht71_datain = Input

   ' clk #9 für ACK absenden und Datain PIN auswerten
   Set Sht71_sck
   Error = Sht71_datain
   Reset Sht71_sck

   ' Rückgabewerte error = 1 wenn kein Acknowledge empfangen
   Sht71_write_byte = Error                                    ' Rückgabewert zuweisen

End Function
'-- End Sht71_write_byte -------------------------------------------------------


'-----------------------------------------------------------------------------
' SHT71 - Function: Sht71_read_byte
' Funktion liest ein Byte vom Sensibus und sende ein Acknowledge in Form von
```

```
' "ack=1"
'
' Parameter:    Acknowledge = Variable die als Acknowledge gesendet wird
' Rückgabewert: Bytewert des vom Sensibus gelesenen Bytes
'-------------------------------------------------------------------------------
Function Sht71_read_byte(byval Acknowledge As Byte) As Byte

   ' ----- locale Variable -----
   Local Databyte As Byte

   ' ----- Programmcode -----

   ' release DATA-line
   Config Sht71_datain = Output
   Set Sht71_dataout
   Config Sht71_datain = Input

   ' Datenbyte einlesen
   ' MSB shifted in first when clock goes high
   Shiftin Sht71_datain , Sht71_sck , Databyte , 1

   ' Acknowledge senden
   Config Sht71_datain = Output

   If Acknowledge = Sht71_acknowledge Then
      ' Botschaft mit ACK quittieren (Pegel = low)
      Reset Sht71_dataout
   Else
      ' Botschaft mit noACK quittieren (Pegel = high)
      Set Sht71_dataout
   End If

   ' clk #9 für ACK absenden
   Set Sht71_sck
   $asm : Nop : Nop : Nop : $end Asm
   Reset Sht71_sck

   ' Release DATA-line
   Set Sht71_dataout

   ' Rückgabewert Databyte zuweisen
   Sht71_read_byte = Databyte

End Function
'-- End Sht71_read_byte ---------------------------------------------------


'-------------------------------------------------------------------------------
' SHT71 - Function: Sht71_softreset
' Funktion resetet den sensor durch einen softreset
'
' Parameter:    keiner
' Rückgabewert: Rückgabewert error = 1 falls Sensor nicht antwortet
'-------------------------------------------------------------------------------
Function Sht71_softreset() As Byte

   ' ----- locale Variable -----
   Local Error As Byte

   ' ----- Programmcode -----

   ' Reset Communication
   Gosub Sht71_connection_reset_sequence

   ' Send RESET-Command an den Sensor
   Error = Sht71_write_byte(sht71_reset)

   ' Rückgabewerte zuweisen
   Sht71_softreset = Error

End Function
'-- End Sht71_softreset ---------------------------------------------------


'-------------------------------------------------------------------------------
' SHT71 - Function: Sht71_read_statusregister
' Funktion liest das Statusregister des Sht71 aus (inkl. 8 Bit Checksumme)
'
' Parameter:    P_Value    = Inhalt des Statusregister wird über
'                            "call by reference" zurückgegeben
'               P_Checksum = Cheksummenbyte wird über "call by reference"
'                            zurückgegeben
' Rückgabewert: Rückgabewert error = 1 falls Sensor nicht antwortet
'-------------------------------------------------------------------------------
Function Sht71_read_statusregister(byref P_value As Byte , Byref P_checksum As Byte) As Byte

   ' ----- locale Variable -----
   Local Error As Byte

   ' ----- Programmcode -----
```

```
   ' Transmission start
   Gosub Sht71_transmission_start

   ' Kommando für Statusregister auslesen senden
   Error = Sht71_write_byte(sht71_status_reg_read)
   P_value = Sht71_read_byte(sht71_acknowledge)             ' Read Statusregister (8-bit)
   P_checksum = Sht71_read_byte(sht71_noacknowledge)        ' Read Checksum (8-bit)

   ' Rückgabewert zuweisen
   Sht71_read_statusregister = Error

End Function
'-- End Sht71_read_statusregister ------------------------------------------


'--------------------------------------------------------------------------------
' SHT71 - Function: Sht71_write_statusregister
' Funktion schreibt in das Statusregister des Sht71 den übergebenen Bytewert
'
' Parameter:    P_Value    = Bytewert der in das Statusregister geschrieben
'                            werden soll. Achtung: "call by reference"
' Rückgabewert: Rückgabewert error = 1 falls Sensor nicht antwortet
'--------------------------------------------------------------------------------
Function Sht71_write_statusregister(byref P_value As Byte) As Byte

   ' ----- locale Variable -----
   Local Error As Byte

   ' ----- Programmcode -----

   ' Transmission start
   Gosub Sht71_transmission_start

   ' Kommando für Statusregister auslesen senden
    Error = Sht71_write_byte(sht71_status_reg_write)

   ' Datenbyte senden
   Sht71_temp_byte1 = Sht71_write_byte(p_value)
   Error = Error + Sht71_temp_byte1

   ' Rückgabewert zuweisen
   Sht71_write_statusregister = Error

End Function
'-- End Sht71_write_statusregister -----------------------------------------


'--------------------------------------------------------------------------------
' SHT71 - Function: Sht71_measure
' Funktion liest je nach Modus Temperature- oder Feuchtigkeitsdaten aus dem
' Sensor aus und übergibt die Messwerte an die jeweiligen Funktionen
'
' Parameter:    P_Value    = Word für 14 Bit Temperatur oder 12 Bit Feuchtigkeit
'                            Achtung: "call by reference"
'               P_checksum = Cheksummenbyte wird über "call by reference"
'                            zurück
'               Modus      = Parameter ob Temperatur oder Feuchtigkeit
' Rückgabewert: Rückgabewert error = 1 falls Sensor nicht antwortet
'--------------------------------------------------------------------------------
Function Sht71_measure(byref P_value As Word , Byref P_checksum As Byte , Byval Modus As Byte) As
Byte

   ' ----- locale Variable -----
   Local Error As Byte
   Local Local_temp_byte1 As Byte
   Local Local_temp_byte2 As Byte
   Local Local_temp_word As Word
   Local Local_temp_long1 As Long

   ' ----- Programmcode -----

   ' Transmission start
   Gosub Sht71_transmission_start

   Select Case Modus
      Case Sht71_modus_humi : Error = Sht71_write_byte(sht71_measure_humi)
      Case Sht71_modus_temp : Error = Sht71_write_byte(sht71_measure_temp)
   End Select

   #if Sht71_testmodus = 1
      Print "Fehler write command : " ; Error
   #endif

   ' Umschalten auf Eingabe
   Config Sht71_datain = Input

   ' So und nun warten bis PIN zappelt
   'Bitwait Sht71_datain , Reset                             ' Problem, Routine wartet ggf. unend-
lich!
```

```
   ' Umsetzung hier über FOR-Schleife mit Timeout
   Local_temp_byte1 = False
   For Local_temp_long1 = 0 To Sht71_response_wait
      If Sht71_datain = 0 Then
         ' Reaktion des SHT71 erkannt, FOR abbrechen und weiter
         Local_temp_byte1 = True
         Exit For
      End If
   Next Local_temp_long1

   #if Sht71_testmodus = 1
      Print "Biterkennung = " ; Local_temp_byte1
   #endif

   ' Wenn PIN zappelt weiter mit Einlesen ansonsten Fehler setzen und fertig
   If Local_temp_byte1 = True Then
      Local_temp_byte1 = Sht71_read_byte(sht71_acknowledge)  ' read the first byte (MSB)
      Local_temp_byte2 = Sht71_read_byte(sht71_acknowledge)  ' read the second byte (LBS)
      P_checksum = Sht71_read_byte(sht71_noacknowledge)      ' read checksum
      Local_temp_word = Local_temp_byte1
      Shift Local_temp_word , Left , 8
      Local_temp_word = Local_temp_word And &HFF00
      Local_temp_word = Local_temp_word + Local_temp_byte2
      P_value = Local_temp_word
   Else
      Incr Error
   End If

   ' Rückgabewert zuweisen
   Sht71_measure = Error

   #if Sht71_testmodus = 1
      Print "Endergebnis : " ; Error
   #endif

End Function
'-- End Sht71_measure ------------------------------------------------------


'-------------------------------------------------------------------------------
' SHT71 - Function: Sht71_crc_check
' Funktion prüft mit dem Übergebenen Daten-Word und der Checksumme ob CRC für
' Datenübertragung korrekt oder fehlerbehaftet.
' Im Fehlerfall wird empfohlen einen Reset auf den Sensor durchzuführen.
'
' Parameter:    Value    = Word für 14 Bit Temperatur oder 12 Bit Feuchtigkeit
'               Crc      = Cheksummenbyte
' Rückgabewert: Bytewert Crc_ok oder Crc_nok
'-------------------------------------------------------------------------------
Function Sht71_crc_check(byval Value As Word , Byval Crc As Byte , Byval Modus As Byte) As Byte

   ' ----- Locale Variablen -----
   Local Crc_reversed As Byte
   Local Local_temp_byte1 As Byte
   Local Local_temp_byte2 As Byte
   Local Local_temp_byte3 As Byte
   Local Crc_register As Byte
   Local Table_index As Byte

   ' ----- Programmcode -----
   ' Vorgaben zur CRC-Rechung aus Application Note CRC Sensirion
   '
   ' Perform the following operations:
   ' 1. Initialize the CRC register with the value of the lower nibble of the
   '    value of the status register (reversed (s0s1s2s3'0000)).
   '    (default '00000000' = 0)
   ' 2. XOR each (transmitted and received) byte with the previous CRC value.
   '    The result is the new byte that you need to calculate the CRC value from.
   ' 3. Use this value as the index to the table to obtain the new CRC value.
   ' 4. Repeat from 2.) until you have passed all bytes through the process.
   ' 5. The last byte retrieved from the table is the final CRC value.
   ' 6. The CRC value retrieved from the SHTxx must be reversed
   '    (bit 0 = bit 7, bit 1=bit 6..bit 7 = bit 0) and can then be compared to
   '    the final CRC value.(2)

   #if Sht71_testmodus = 1
      Print "Enter CRC-Prüfung"
   #endif

   ' Zur CRC-Prüfung muss das vom SHT71 empfangene CRC-Byte "gedreht" werden.
   ' bit 0 = bit 7, bit 1 = bit 6 ... bit 7 = bit 0
   Local_temp_byte1 = 0
   Local_temp_byte2 = 7
   Do
      Crc_reversed.local_temp_byte1 = Crc.local_temp_byte2
      Incr Local_temp_byte1
      Decr Local_temp_byte2
   Loop Until Local_temp_byte1 = 8
```

```
   ' Ab jetzt gilt zur Berechung das "gedrehte CRC-Byte Crc_reversed

   ' Initialisierung des Crc_registers mit dem lower nibble des Status-Registers
   ' hier werden Konstanten angenommen da Default = &B00000000 = 0 ist.
   ' (reversed(s0 s1 s2 s3 0 0 0))
   Crc_register = 0

   ' Jedes gesendete und empfangene Byte muss nun verarbeitet werden.
   ' Hierzu wird Datenstruktur aufgebaut

   Select Case Modus
      Case Sht71_modus_humi : Sht71_transceive_struct(1) = 5
      Case Sht71_modus_temp : Sht71_transceive_struct(1) = 3
   End Select

   Sht71_transceive_struct(2) = High(value)
   Sht71_transceive_struct(3) = Low(value)

   ' Nun kann CRC gerechnet werden
   For Local_temp_byte1 = 1 To 3

      ' Lese Byte aus Datenstruktur für Tabellenzugriff aus
      Local_temp_byte2 = Sht71_transceive_struct(local_temp_byte1)

      ' Setzte den Zeiger auf den Anfang des DATA-Statement Sht71_Crc_table
      Restore Sht71_crc_table

      ' Zugriff und Berechnung
      Table_index = Crc_register Xor Local_temp_byte2
      If Local_temp_byte2 = 0 Then
         Table_index = 3
      End If

      For Local_temp_byte3 = 0 To Table_index
         Read Crc_register
      Next Local_temp_byte3

   Next Local_temp_byte1

   #if Sht71_testmodus = 1
      Print "Befehl :     " ; Hex(sht71_transceive_struct(1))
      Print "Datenwort : " ; Hex(value)
      Print "CRC :        " ; Hex(crc_reversed)
   #endif

   If Crc_register = Crc_reversed Then
      Local_temp_byte1 = Crc_ok
      #if Sht71_testmodus = 1
         Print "CRC - OK"
         Print
      #endif
   Else
      Local_temp_byte1 = Crc_nok
      #if Sht71_testmodus = 1
         Print "CRC - NOK"
         Print
      #endif
   End If

   Sht71_crc_check = Local_temp_byte1

End Function
'-- End Sht71_crc_check ------------------------------------------------------


'*******************************************************************************
' GOSubroutinen
'*******************************************************************************

'-------------------------------------------------------------------------------
' SHT71 - Gosub-Routine: Sht71_connection_reset_sequence
' Bei StartUp, Verbindungsverlust, Reset und CRC-Fehler muss das serielle
' Interface zurückgesetzt werden. Hierzu dient diese Routine.
' Die Routine sende die Im Datenblatt beschriebene connection reset sequence.
' Hierzu wird während DATA=high SCK mindestens 9 Mal getoggelt. Anschließend
' wird eine Transmission Start Sequenz gesendet.
'-------------------------------------------------------------------------------
Sht71_connection_reset_sequence:

   ' Absicherung dass DATA und SCK Pin's auf Output stehen da Vorgeschichte unbekannt
   Config Temp_sck = Output
   Config Temp_data = Output

   ' Initial State
   Set Sht71_dataout
   Reset Sht71_sck

   ' Connection Reset Sequence
   '          _____________________________________________________________|_______|_______
   ' Data:
```

```
   '
   ' SCK:  __|¯¯|__|¯¯|__|¯¯|__|¯¯|__|¯¯|__|¯¯|__|¯¯|__|¯¯|_____|¯¯¯|__|¯¯¯|_____

   For Sht71_temp_byte1 = 1 To 9
      Set Sht71_sck
      ' Waitus 2
      Reset Sht71_sck
      ' Waitus 2
   Next Sht71_temp_byte1

   Gosub Sht71_transmission_start

Return
'-- End Sht71_connection_reset_sequence ------------------------------------


'---------------------------------------------------------------------------
' SHT71 - Gosub-Routine: Sht71_transmission_start
' Diese Routine initiiert das Senden von Kommandos an den SHT71. Jede Aussen-
' dung von Daten vom ATmega zum SHT71 muss mit dieser Sequenz eingeleitet
' werden
'---------------------------------------------------------------------------
Sht71_transmission_start:

   ' Absicherung dass DATA und SCK Pin's auf Output stehen da Vorgeschichte unbekannt
   Config Temp_sck = Output
   Config Temp_data = Output

   ' Initial state
   Set Sht71_dataout
   Reset Sht71_sck
   $asm : Nop : $end Asm

   ' Transmission Start Sequence
   '
   ' Data:  _____|_______|_______
   '
   '           ___     ___
   ' SCK:  ___|   |___|   |_____
   '
   Set Sht71_sck
   $asm : Nop : $end Asm
   Reset Sht71_dataout
   $asm : Nop : $end Asm
   Reset Sht71_sck
   $asm : Nop : Nop : Nop : $end Asm
   Set Sht71_sck
   $asm : Nop : $end Asm
   Set Sht71_dataout
   $asm : Nop : $end Asm
   Reset Sht71_sck

Return
'-- End Sht71_transmission_start -------------------------------------------


'---------------------------------------------------------------------------
' SHT71 - Gosub-Routine: Sht71_perform_measurement
' Diese Routine liest Luftfeuchtigkeit und Temperatur aus dem Sensor aus. Die
' Rohdaten werden in den entsprechenden globalen Variablen gespeichert.
' Anschließend wird eine Checksumme-Überprüfung durchgeführt.
'---------------------------------------------------------------------------
Sht71_perform_measurement:

   #if Sht71_testmodus = 1
      Print "Enter Sht71_perform_measurement"
   #endif

   ' globales Fehler-Gesamtergebnis zurücksetzen
   Sht71_error_result = 0

   ' Da die Temperatur für die Rechnung der relativen Luftfeuchtigkeit Basis ist
   ' wird diese zuerst ausgelesen und in den globalen Variablen abgelegt.

   ' Temperatur auslesen
   Sht71_temp_byte1 = Sht71_measure(sht71_temp_word , Sht71_checksum , Sht71_modus_temp)

   ' Prüfung ob ein Fehler bei der Kommunikation aufgetreten ist.
   If Sht71_temp_byte1 = 0 Then
      ' NEIN

      ' CRC-Prüfung der Temperaturmessdaten
      Sht71_temp_byte1 = Sht71_crc_check(sht71_temp_word , Sht71_checksum , Sht71_modus_temp)

      ' Prüfung ob CRC NOK
      If Sht71_temp_byte1 = Crc_nok Then

         ' JA, Messung wahr fehlerhaft -> Fehlerzähler erhöhen
         Incr Sht71_error_result
         #if Sht71_testmodus = 1
            Print "CRC-Fehler TEMP"
```

```
         #endif

      End If
   Else
      ' JA -> Fehlerzähler erhöhen
      Incr Sht71_error_result
      #if Sht71_testmodus = 1
         Print "Kommunikationsfehler TEMP"
      #endif
   End If

   ' Luftfeuchtigkeit auslesen
   Sht71_temp_byte1 = Sht71_measure(sht71_humi_word , Sht71_checksum , Sht71_modus_humi)

   ' Prüfung ob ein Fehler bei der Kommunikation aufgetreten ist.
   If Sht71_temp_byte1 = 0 Then
      ' NEIN

      ' CRC-Prüfung der Luftfeuchtigkeitsmessdaten
      Sht71_temp_byte1 = Sht71_crc_check(sht71_humi_word , Sht71_checksum , Sht71_modus_humi)

      ' Prüfung ob CRC NOK
      If Sht71_temp_byte1 = Crc_nok Then

         ' JA, Messung wahr fehlerhaft -> Fehlerzähler erhöhen
         Incr Sht71_error_result
         #if Sht71_testmodus = 1
            Print "CRC-Fehler HUMI"
         #endif

      End If
   Else
      ' JA -> Fehlerzähler erhöhen
      Incr Sht71_error_result
      #if Sht71_testmodus = 1
         Print "Kommunikationsfehler HUMI"
      #endif
   End If


   ' Wenn ein Fehler aufgetreten ist dann Sensor zurücksetzen
   If Sht71_error_result > 0 Then

      #if Sht71_testmodus = 1
         Print "Perform Sensor Reset wegen Fehlern"
      #endif

      ' Softreset mit Kommunikations-Restart durchführen
      Sht71_temp_byte1 = Sht71_softreset()

      ' Zeit "Sht71_communication_wait" warten nach Reset / PowerUp
      Waitms Sht71_communication_wait

      ' Nach Reset Kommunikation erneut starten
      Gosub Sht71_connection_reset_sequence

   End If

Return
'-- End Sht71_perform_measurement ----------------------------------------------


'--------------------------------------------------------------------------------
' Gosub-Routine: Test_write_byte_to_sht71
' Testroutine zum Senden eines Bytes an den SHT71
'--------------------------------------------------------------------------------
Test_write_byte_to_sht71:

   #if Main_testmodus = 1
      Print "Enter Main_testmodus"
      Print "Sende Sht71_transmission_start und Byte-Wert RESET"
   #endif

   Gosub Sht71_transmission_start

   Temp_byte1 = Sht71_write_byte(sht71_reset)

   #if Main_testmodus = 1
      Print "Fehlerrückgabewert : " ; Temp_byte1
      Print
   #endif

   ' Nach Reset Kommunikation erneut starten
   Gosub Sht71_connection_reset_sequence

Return
'-- End Test_write_byte_to_sht71 ----------------------------------------------


'--------------------------------------------------------------------------------
```

```
' Gosub-Routine: Test_perform_softreset
' Testroutine zum Aufrufen der Soft-Reset-Funktion
'-------------------------------------------------------------------------------
Test_perform_softreset:

   #if Main_testmodus = 1
      Print "Enter Test_performe_softreset"
   #endif

   Temp_byte1 = Sht71_softreset()

   #if Main_testmodus = 1
      Print "Fehlerrückgabewert : " ; Temp_byte1
      Print
   #endif

   ' Zeit "Sht71_communication_wait" warten nach Reset / PowerUp
   Waitms Sht71_communication_wait

   ' Nach Reset Kommunikation erneut starten
   Gosub Sht71_connection_reset_sequence

Return
'-- End Test_perform_softreset ------------------------------------------------


'-------------------------------------------------------------------------------
' Gosub-Routine: Test_statusregister_lesen
' Testroutine zum Auslesen des Statusregisters
'-------------------------------------------------------------------------------
Test_statusregister_lesen:

   #if Main_testmodus = 1
      Print "Enter Test_statusregister_lesen"
   #endif

   Temp_byte2 = &HFF
   Temp_byte3 = &HFF

   Temp_byte1 = Sht71_read_statusregister(temp_byte2 , Temp_byte3)

   #if Main_testmodus = 1
      Print "Error      : " ; Temp_byte1
      Print "Datenbyte  : " ; Bin(temp_byte2)
      Print "Checksumme : " ; Hex(temp_byte3)
      Print
   #endif

Return
'-- End Test_statusregister_lesen ---------------------------------------------


'-------------------------------------------------------------------------------
' Gosub-Routine: Test_statusregister_schreiben
' Testroutine zum Auslesen des Statusregisters
'-------------------------------------------------------------------------------
Test_statusregister_schreiben:

   #if Main_testmodus = 1
      Print "Enter Test_statusregister_schreiben"
   #endif

   Temp_byte2 = &B00000000                                  ' 12 Bit RH und 14 Bit T
   ' Temp_byte2 = &B00000001                                  ' 8 Bit RH und 12 Bit T


   Temp_byte1 = Sht71_write_statusregister(temp_byte2)

   #if Main_testmodus = 1
      Print "Error                  : " ; Temp_byte1
      Print "Datenbyte geschrieben : " ; Bin(temp_byte2)
      Print
   #endif

Return
'-- End Test_statusregister_schreiben ----------------------------------------


'-------------------------------------------------------------------------------
' Gosub-Routine: Test_temp_und_humi_messung
' Testroutine zum Auslesen der Temperatur und Luftfeuchtigkeit
'-------------------------------------------------------------------------------
Test_temp_und_humi_messung:

   #if Main_testmodus = 1
      Print "Enter Test_temp_und_humi_messung"
      Print "Starte Messung der Temperatur"
   #endif

   Temp_byte1 = Sht71_measure(sht71_temp_word , Sht71_checksum , Sht71_modus_temp)
```

```
   #if Main_testmodus = 1
      Print "Ergebnisword :" ; Hex(sht71_temp_word)
      Print "Cheksumme :" ; Hex(sht71_checksum)
      Print "Fehler :" ; Temp_byte1
   #endif

   Temp_byte2 = Sht71_measure(sht71_humi_word , Sht71_checksum , Sht71_modus_humi)
   #if Main_testmodus = 1
      Print "Ergebnisword :" ; Hex(sht71_humi_word)
      Print "Cheksumme :" ; Hex(sht71_checksum)
      Print "Fehler :" ; Temp_byte2
      Print
   #endif

Return
'-- End Test_temp_und_humi_messung ------------------------------------------

'--------------------------------------------------------------------------------
' Gosub-Routine: Test_perform_sht71
' Testroutine zum Auslesen und berechnen der Temperatur und Luftfeuchtigkeit
'--------------------------------------------------------------------------------
Test_perform_sht71:

   #if Main_testmodus = 1
      Print "Enter Test_perform_sht71"
   #endif

   Print "*** Performing SHT71 Temp und Humi ***"

   Gosub Sht71_perform_measurement
   Call Sht71_calc_measurements()

   Temp_string1 = Fusing(sht71_t , "###.##")
   Print "Temp     : " ; Temp_string1 ; "°"

   Temp_string1 = Fusing(sht71_rhlinear , "###.##")
   Print "RHlinear : " ; Temp_string1 ; "%"

   Temp_string1 = Fusing(sht71_rhtrue , "###.##")
   Print "RHtrue   : " ; Temp_string1 ; "%"

   Temp_string1 = Fusing(sht71_dewpoint , "###.##")
   Print "Taupunkt : " ; Temp_string1 ; "°"

   Print "***"
   Print

Return
'-- End Test_perform_sht71 ---------------------------------------------------

End                                                    'end program


'--------------------------------------------------------------------------------
' Definition von globalen Konstantenfeldern
'--------------------------------------------------------------------------------
Sht71_crc_table:
Data 0 , 49 , 98 , 83 , 196 , 245 , 166 , 151 , 185 , 136 , 219 , 234 , 125 , 76 , 31 , 46,
Data 67 , 114 , 33 , 16 , 135 , 182 , 229 , 212 , 250 , 203 , 152 , 169 , 62 , 15 , 92 , 109,
Data 134 , 183 , 228 , 213 , 66 , 115 , 32 , 17 , 63 , 14 , 93 , 108 , 251 , 202 , 153 , 168,
Data 197 , 244 , 167 , 150 , 1 , 48 , 99 , 82 , 124 , 77 , 30 , 47 , 184 , 137 , 218 , 235,
Data 61 , 12 , 95 , 110 , 249 , 200 , 155 , 170 , 132 , 181 , 230 , 215 , 64 , 113 , 34 , 19,
Data 126 , 79 , 28 , 45 , 186 , 139 , 216 , 233 , 199 , 246 , 165 , 148 , 3 , 50 , 97 , 80,
Data 187 , 138 , 217 , 232 , 127 , 78 , 29 , 44 , 2 , 51 , 96 , 81 , 198 , 247 , 164 , 149,
Data 248 , 201 , 154 , 171 , 60 , 13 , 94 , 111 , 65 , 112 , 35 , 18 , 133 , 180 , 231 , 214,
Data 122 , 75 , 24 , 41 , 190 , 143 , 220 , 237 , 195 , 242 , 161 , 144 , 7 , 54 , 101 , 84,
Data 57 , 8 , 91 , 106 , 253 , 204 , 159 , 174 , 128 , 177 , 226 , 211 , 68 , 117 , 38 , 23,
Data 252 , 205 , 158 , 175 , 56 , 9 , 90 , 107 , 69 , 116 , 39 , 22 , 129 , 176 , 227 , 210,
Data 191 , 142 , 221 , 236 , 123 , 74 , 25 , 40 , 6 , 55 , 100 , 85 , 194 , 243 , 160 , 145,
Data 71 , 118 , 37 , 20 , 131 , 178 , 225 , 208 , 254 , 207 , 156 , 173 , 58 , 11 , 88 , 105,
Data 4 , 53 , 102 , 87 , 192 , 241 , 162 , 147 , 189 , 140 , 223 , 238 , 121 , 72 , 27 , 42,
Data 193 , 240 , 163 , 146 , 5 , 52 , 103 , 86 , 120 , 73 , 26 , 43 , 188 , 141 , 222 , 239,
Data 130 , 179 , 224 , 209 , 70 , 119 , 36 , 21 , 59 , 10 , 89 , 104 , 255 , 206 , 157 , 172

'#################################### END ##################################
```

Software 6: Code zum Betrieb des SHT71 am ATmega

8.11.5 Prototyp zum SHT71 Sensor

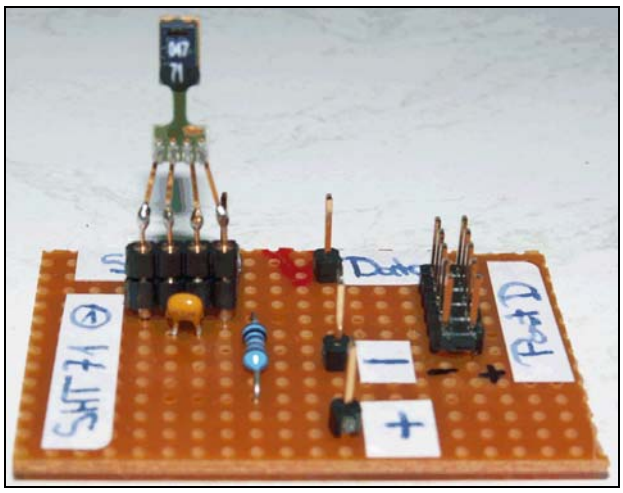


Abbildung 93: Prototyp SHT71 Sensor-Anbindung

8.12 Lüftersteuerung für Gehäusebelüftung

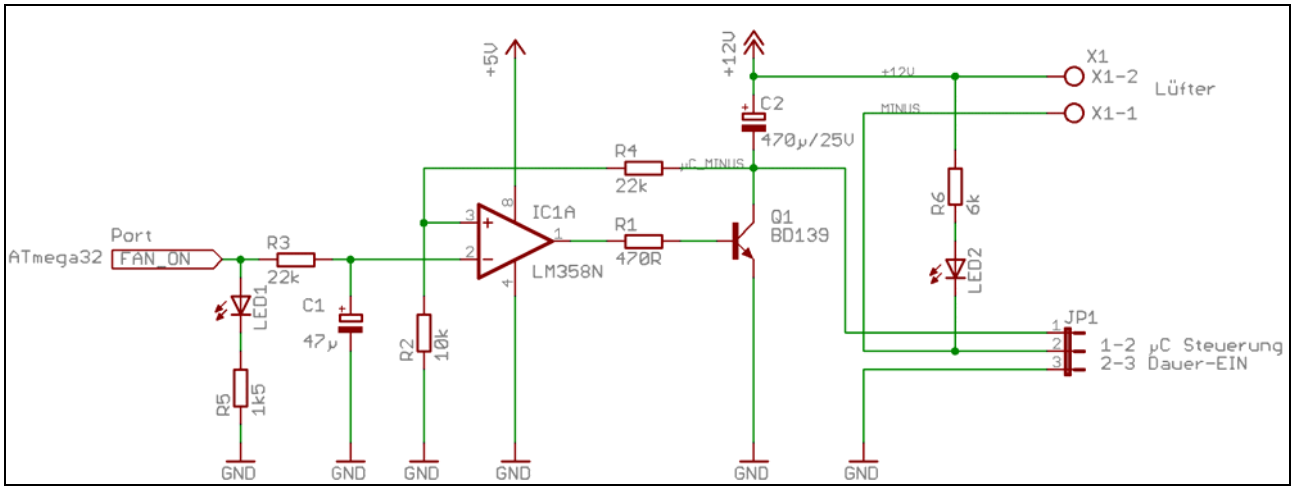
Zur Belüftung des Gehäuses um Wärme abzuleiten, aber auch zur Versorgung des Temperatur- und Luftfeuchtigkeitssensors wird eine Ventilator im Gehäuse eingebaut welcher direkt über Versorgungsspannung oder aber vom ATmega aus gesteuert werden kann.

Im folgenden wir die dafür notwendige Schaltung dargestellt und die Funktion erläutert.

Zur Spannungsversorgung werden die im System verfügbaren 12V verwendet. Bei der Auswahl des Lüfters ist auf geringen Stromverbrauch und kleine Dimensionierung zu achten!

Bei der Umsetzung wird der Lüfter ein einen PWM geregelten Ausgang des ATmega128 angeschlossen os dass mittels SW die Drehzahl des Lüfters geregelt werden kann.

Beschaltung:



Schaltbild 32: Schaltplan für Lüfteransteuerung inkl. Status-LED

Achtung:
FAN_ON ist Null-Aktiv. Lüfter läuft bei FAN_ON = 0 und steht bei FAN_ON = 1!

Bauteile:

Stückliste: Lüfteransteuerung	
Widerstände	*Halbleiter*

R1	470 Ω	IC1A	Operationsverstärker LM358
R2	10k Ω	Q1	Leistungstransistor BD139
R3, R4	22k Ω	LED1	low current LED 3mm, gelb
R5	1,5k Ω	LED2	low current LED 3mm, grün
R6	6K Ω	*Sonstiges*	
Kondensator		X1	Schraubklemme, Raster 5mm
C1, C2	470µF / 25V	JP1	Jumper 3-polig, 2,54mm Raster

Tabelle 102: Stückliste Lüfteransteuerung

Mit dem Jumper JP1 kann die Schaltung so verändert werden, dass der Lüfter unabhängig vom ATmega läuft. Hierzu muss lediglich der Hauptschalter eingeschaltet und die Spannung von 12V anliegen.

Jumperstellung 2-3 = Dauerbetrieb

Soll der Lüfter vom ATmega bei Bedarf und über SW gesteuert eingeschaltet werden so ist die folgende Jumperstellung zu verwenden:

Jumperstellung 1-2 = µC Steuerung

Themenbezogene LED's:

LED / Funktion / Beschreibung	*Abkürzung*	*Farbe*	
Lüfter über SW	LED_FAN_SW	gelb	*[LED25]*
Lüfter über Dauerplus	LED_FAN_12V	gelb	*[LED26]*

Tabelle 103: Themenbezogene LED's Lüftersteuerung

Ressourcenzuordnung zum ATmega128:

Nummer	*Schaltbild*	*Ressource ATmega128*	
1	FAN_ON [FAN_PWM]	PortE.4	[OC3A] (Timer3)

Tabelle 104: Ressourcenzuordnung ATmega128 für Lüftersteuerung

8.12.1 BASCOM Beispielcode für gemeinsame PWM LCD und Lüfter

```
'#################################################################################
' PWM_LCD_und_FAN.BAS                                          Stand 15.01.2007
' -------------------                                          (C) Markus Fulde
' Testprogramm zur Umstellung der PWM für das LCD-Display auf Timer3 und für
' die Inbetriebnahme der PWM für die Lüfter-Endstufe.
' Die für die PWM-Ansteuerungen notwendigen Routinen sind als solche
' gekennzeichnet und bereits zur späteren Verwendung ausgelagert.
'#################################################################################


'--------------------------------------------------------------------------------
' Compilerinstruktionen und Compilerdirektiven
'--------------------------------------------------------------------------------
$regfile = "m128def.dat"
$crystal = 16000000
$hwstack = 32
$swstack = 32
$framesize = 40

$baud = 38400


'--------------------------------------------------------------------------------
' Definition von Variablen, Konstaten und Datentypen
'--------------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========
```

```
Const Lcd_testmodus = 1
Const Fan_testmodus = 1

' LCD: Konstanten für LCD-Helligkeit (Range 0-255 / 8 Bit)
Const Schrittweite_lcdhelligkeit = 5                    ' Schrittweite für Einzelschritte
Const Lcd_helligkeit_min = 0                            ' minimale Helligkeit
Const Lcd_helligkeit_max = 255                          ' maximale Helligkeit
Const Lcd_default_helligkeit = 60                       ' Default Helligkeit empirisch ermittelt
Const Lcd_helligkeit_aus = 255                          ' Bytewert für LCD-Aus

' FAN: Konstaten für FAN-Geschwindigkeit (Range 0-255 / 8 Bit)
Const Schrittweite_fangeschwindigkeit = 5               ' Schrittweite für Einzelschritte
Const Fan_geschwindigkeit_min = 100                     ' minimale Geschwindigkeit
Const Fan_geschwindigkeit_max = 255                     ' maximale Geschwindigkeit
Const Fan_default_geschwindigkeit = 200                 ' Default Geschwindigkeit
Const Fan_anlaufgeschwindigkeit = 130                   ' Anlaufgeschwindigkeit
Const Fan_belueftung_aus = 0                            ' Bytewert für LCD-An

' LCD: Variablen deklarieren und mit Defaultwert laden
Dim Lcd_pwmwert As Byte                                 ' Dimwert für LCD-Helligkeit
Lcd_pwmwert = Lcd_default_helligkeit

' FAN: Variablen deklarieren und mit Defaultwert laden
Dim Fan_pwmwert As Byte                                 ' Geschwindigkeitswert für Lüfter
Fan_pwmwert = Fan_default_geschwindigkeit

'========== Für Testumgebung  ==========

' LCD: Temporäre Arbeitsvariablen definieren
Dim Tempvar_1 As Byte                                   ' Temporäre Variable 1
Dim Tempvar_2 As Byte                                   ' Temporäre Variable 2

Const Timervorgabe = 3036                               ' Timercounter für 1 Sek.


'-------------------------------------------------------------------------------
' Prototyping
'-------------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========
Declare Sub Fan_set_geschwindigkeit(byval Value As Byte)
Declare Sub Lcd_set_helligkeit(byval Value As Byte)

'========== Für Testumgebung  ==========


'-------------------------------------------------------------------------------
' Konfiguration und Basiseinstellungen
'-------------------------------------------------------------------------------

'========== Für Testumgebung  ==========

' Alias Namen für Tastaturabfragen
Taste_1 Alias Pina.0
Taste_2 Alias Pina.1
Taste_3 Alias Pina.2
Taste_4 Alias Pina.3
Taste_5 Alias Pina.4
Taste_6 Alias Pina.5
Taste_7 Alias Pina.6
Taste_8 Alias Pina.7

Led_1 Alias Porte.5
Led_2 Alias Porte.6
Led_3 Alias Porte.7


'========== Für Projekt ConvCtrl ==========


'-------------------------------------------------------------------------------
' Konfiguration und Basiseinstellungen (Projekt und Testumgebung)
'-------------------------------------------------------------------------------

'========== Für Testumgebung  ==========

'Konfiguration des Port A als Eingang für Tastaturabfrage
Config Porta = Input

' Konfiguration des Pin E.5-7 als Ausgang für LED-Kontrolleuchten
Config Pine.5 = Output
Config Pine.6 = Output
Config Pine.7 = Output

' Konfiguration des 1 Sekunden-Timers für LED-Toggle-Funktion
Config Timer1 = Timer , Prescale = 256
On Timer1 Timer_irq

' Timer aktivieren und Interrupts freigeben
Enable Timer1
```

```
Enable Interrupts


'========== Für Projekt ConvCtrl ==========

'Konfiguration Timer 3 für Hardware-PWM an OC3A (E.3) und OC3B (E.4)
' Konfiguration muss global geschehen da mit einem Timer zwei PWM realisiert
Config Timer3 = Pwm , Pwm = 8 , Prescale = 256 , Compare A Pwm = Clear Up , Compare B Pwm = Clear Up

' LCD: Routinen und Einstellungen
Gosub Lcd_konfig                                        ' LCD Konfigurieren

' FAN: Routinen und Einstellungen
Gosub Fan_konfig

' LCD: Startwert für Dimmung festlegen und Dimmung starten
Gosub Lcd_beleuchtung_an

' FAN: Startwert für Lüfteranlauf festlegen und Lüfter starten
Gosub Fan_lueftung_aus


'========== Für Testumgebung ==========

#if Main_testmodus
   Print "** Enter MainLoop **"
   Print
#endif


'#############################################################################
' Hauptprogramm
'#############################################################################
'Hier ist die Programmhauptschleife
Do

   ' ----- Testroutine für Einzeltasten -----

   ' Tastaturabfrage für Taste 1
   Debounce Taste_1 , 0 , Lcd_beleuchtung_heller , Sub

   ' Tastaturabfrage für Taste 2
   Debounce Taste_2 , 0 , Lcd_beleuchtung_dunkler , Sub

   ' Tastaturabfrage für Taste 3
   Debounce Taste_3 , 0 , Lcd_beleuchtung_aus , Sub

   ' Tastaturabfrage für Taste 4
   Debounce Taste_4 , 0 , Lcd_beleuchtung_an , Sub

   ' Tastaturabfrage für Taste 5
   Debounce Taste_5 , 0 , Fan_lueftung_an , Sub

   ' Tastaturabfrage für Taste 6
   Debounce Taste_6 , 0 , Fan_lueftung_aus , Sub

   ' Tastaturabfrage für Taste 7
   Debounce Taste_7 , 0 , Fan_lueftung_schneller , Sub

   ' Tastaturabfrage für Taste 8
   Debounce Taste_8 , 0 , Fan_lueftung_langsamer , Sub


   ' ----- Testroutinen für Parameterübergabe -----

   ' Werte über RS232 einlesen
   Input "Lüftergeschwindigkeit : " , Tempvar_1
   Input "LCD Helligkeit        : " , Tempvar_2

   ' Kontrollausgabe
   Print "Setze Lüftergeschwindigkeit auf " ; Tempvar_1
   Print "Setze LCD Helligkeit auf " ; Tempvar_2

   ' Subroutinen aufrufen
   Call Fan_set_geschwindigkeit(tempvar_1)
   Call Lcd_set_helligkeit(tempvar_2)

Loop
'## End Hauptprogramm ########################################################


'*****************************************************************************
' Interruptroutinen
'*****************************************************************************

'-----------------------------------------------------------------------------
' Interrupt-Service-Routine (Timer1): Timer_irq
' Routine zur Auswertung des Timer Interrupts
'-----------------------------------------------------------------------------
Timer_irq:
```

```
   ' Timervariable neu setzen
   Timer1 = Timervorgabe

   ' LED-Ausgaben toggeln
   Toggle Led_1
   Toggle Led_2
   Toggle Led_3

Return
'-- End Timer_irq ------------------------------------------------------------


'*******************************************************************************
' Subroutinen
'*******************************************************************************

'-------------------------------------------------------------------------------
' FAN - Subroutine: Fan_set_geschwindigkeit
' Subroutine setzt den als Übergabeparameter übergebenen Bytewert als Ge-
' schwindigkeitswert für die Lüftergeschwindigkeit.
' Kleinere Bytewerte als die Minimalgeschwindigkeit werden auf MIN begrenzt.
'
' Parameter:    Value = Bytewert für Lüftergeschwindigkeit
' Rückgabewert: keine
'-------------------------------------------------------------------------------
Sub Fan_set_geschwindigkeit(byval Value As Byte)

   #if Fan_testmodus
      Print "Enter Fan_set_geschwindigkeit"
   #endif

   ' Prüfung ob Bytewert zu klein, wenn ja auf MIN begrenzen
   If Value <= Fan_geschwindigkeit_min Then
      Value = Fan_geschwindigkeit_min
   End If

   ' PWM-Wert setzen und Variable in Timer-Count-Register laden
   Fan_pwmwert = Value
   Pwm3b = Fan_pwmwert

End Sub
'-- End Fan_set_geschwindigkeit -------------------------------------------------


'-------------------------------------------------------------------------------
' LCD - Subroutine: Lcd_set_helligkeit
' Subroutine setzt den als Übergabeparameter übergebenen Bytewert als
' LCD Helligkeitswert
'
' Parameter:    Value = Bytewert für LCD Helligkeit
' Rückgabewert: keine
'-------------------------------------------------------------------------------
Sub Lcd_set_helligkeit(byval Value As Byte)

   #if Lcd_testmodus
      Print "Enter Lcd_set_helligkeit"
   #endif

   ' PWM-Wert setzen und Variable in Timer-Count-Register laden
   Lcd_pwmwert = Value
   Pwm3a = Lcd_pwmwert

End Sub
'-- End Lcd_set_helligkeit ------------------------------------------------------


'*******************************************************************************
' Funktionen
'*******************************************************************************

' keine Funktionen implementiert

'*******************************************************************************
' GOSubroutinen
'*******************************************************************************

'-------------------------------------------------------------------------------
' LCD - Gosub-Routine: Lcd_konfig
' Subroutine zur Konfiguration des LCD-Display inkl. der dafür
' notwendigen Timer.
'-------------------------------------------------------------------------------
Lcd_konfig:

   #if Lcd_testmodus
      Print "Enter Lcd_konfig"
   #endif

   '----- Konfiguration LCD-Helligkeit (PWM) -----
   'Variable für Helligkeit in Timerregister laden
```

```
   Pwm3a = Lcd_pwmwert

   'Port E.3 ist Timer-Ausgang OC3A
   Config Pine.3 = Output

Return
'-- End Lcd_konfig -------------------------------------------------------------


'-----------------------------------------------------------------------------
' LCD - Gosub-Routine: Lcd_beleuchtung_an
' Routine schaltet die Beleuchtung ein
'-----------------------------------------------------------------------------
Lcd_beleuchtung_an:

   #if Lcd_testmodus
      Print "Enter Lcd_beleuchtung_an"
   #endif

   ' PWM-Wert in Timerregister laden
   Pwm3a = Lcd_pwmwert

Return
'-- End Lcd_beleuchtung_an ------------------------------------------------------


'-----------------------------------------------------------------------------
' LCD - Gosub-Routine: Lcd_beleuchtung_aus
' Routine schaltet die Beleuchtung aus
'-----------------------------------------------------------------------------
Lcd_beleuchtung_aus:

   #if Lcd_testmodus
      Print "Enter Lcd_beleuchtung_aus"
   #endif

   ' PWM-Wert in Timerregister laden
   Pwm3a = Lcd_helligkeit_aus

Return
'-- End Lcd_beleuchtung_aus -----------------------------------------------------


'-----------------------------------------------------------------------------
' LCD - Gosub-Routine: Lcd_beleuchtung_heller
' Routine schaltet die Beleuchtung heller
'-----------------------------------------------------------------------------
Lcd_beleuchtung_heller:

   #if Lcd_testmodus
      Print "Enter Lcd_beleuchtung_heller"
   #endif

   If Lcd_pwmwert > Lcd_helligkeit_min Then
      Lcd_pwmwert = Lcd_pwmwert - Schrittweite_lcdhelligkeit
   End If

   ' PWM-Wert in Timerregister laden
   Pwm3a = Lcd_pwmwert

   #if Lcd_testmodus
      Print "Lcd_pwmwert: " ; Lcd_pwmwert
   #endif

Return
'-- End Lcd_testmodus ------------------------------------------------------------


'-----------------------------------------------------------------------------
' LCD - Gosub-Routine: Lcd_beleuchtung_dunkler
' Routine schaltet die Beleuchtung dunkler
'-----------------------------------------------------------------------------
Lcd_beleuchtung_dunkler:

   #if Lcd_testmodus
      Print "Enter Lcd_beleuchtung_dunkler"
   #endif

   If Lcd_pwmwert < Lcd_helligkeit_max Then
      Lcd_pwmwert = Lcd_pwmwert + Schrittweite_lcdhelligkeit
   End If

   ' PWM-Wert in Timerregister laden
   Pwm3a = Lcd_pwmwert

   #if Lcd_testmodus
      Print "Lcd_pwmwert: " ; Lcd_pwmwert
   #endif

Return
```

```
'-- End Lcd_beleuchtung_dunkler ------------------------------------------


'--------------------------------------------------------------------------------
' FAN - Gosub-Routine: Fan_konfig
' Subroutine zur Konfiguration der PWM für die Lüftersteuerung
'--------------------------------------------------------------------------------
Fan_konfig:

   #if Fan_testmodus
      Print "Enter Fan_konfig"
   #endif

   '----- Konfiguration FAN-Geschwindigkeit (PWM) -----
   'Variable für Geschwindgkeit in Timerregister laden
   Pwm3b = Fan_pwmwert

   'Port E.4 ist Timer-Ausgang OC3B
   Config Pine.4 = Output

Return
'-- End Lcd_konfig --------------------------------------------------------


'--------------------------------------------------------------------------------
' FAN - Gosub-Routine: Fan_an
' Routine schaltet den Lüfter ein
'--------------------------------------------------------------------------------
Fan_lueftung_an:

   #if Fan_testmodus
      Print "Enter Fan_an"
      Print "Setze PWM-Wert " ; Fan_pwmwert
   #endif

   ' PWM-Wert in Timerregister laden
   Pwm3b = Fan_pwmwert

Return
'-- End Lcd_beleuchtung_an -------------------------------------------------


'--------------------------------------------------------------------------------
' FAN - Gosub-Routine: Fan_aus
' Routine schaltet den Lüfter aus
'--------------------------------------------------------------------------------
Fan_lueftung_aus:

   #if Fan_testmodus
      Print "Enter Fan_aus"
   #endif

   ' PWM-Wert in Timerregister laden
   Pwm3b = Fan_belueftung_aus

Return
'-- End Fan_aus ------------------------------------------------------------


'--------------------------------------------------------------------------------
' FAN - Gosub-Routine: Fan_lueftung_schneller
' Routine erhöht die Lüfterdrehzahlen
'--------------------------------------------------------------------------------
Fan_lueftung_schneller:

   #if Fan_testmodus
      Print "Enter Fan_lueftung_schneller"
   #endif

   If Fan_pwmwert < Fan_geschwindigkeit_max Then
      Fan_pwmwert = Fan_pwmwert + Schrittweite_fangeschwindigkeit
   End If

   ' PWM-Wert in Timerregister laden
   Pwm3b = Fan_pwmwert

   #if Fan_testmodus
      Print "Fan_pwmwert: " ; Fan_pwmwert
   #endif

Return
'-- End Fan_lueftung_schneller ----------------------------------------------


'--------------------------------------------------------------------------------
' FAN - Gosub-Routine: Fan_lueftung_langsamer
' Routine verringert die Lüfterdrehzahl
'--------------------------------------------------------------------------------
Fan_lueftung_langsamer:
```

```
    #if Fan_testmodus
       Print "Enter Fan_lueftung_langsamer"
    #endif

    If Fan_pwmwert > Fan_geschwindigkeit_min Then
       Fan_pwmwert = Fan_pwmwert - Schrittweite_fangeschwindigkeit
    End If

    ' PWM-Wert in Timerregister laden
    Pwm3b = Fan_pwmwert

    #if Fan_testmodus
       Print "Fan_pwmwert: " ; Fan_pwmwert
    #endif

Return
'-- End Fan_lueftung_langsamer ------------------------------------------------

End

'####################################### END ##################################
```

Software 7: Code zum Betrieb des Lüfters und LCD an einem Timer (PWM)

8.12.2 Prototyp Lüfteransteuerung via PWM

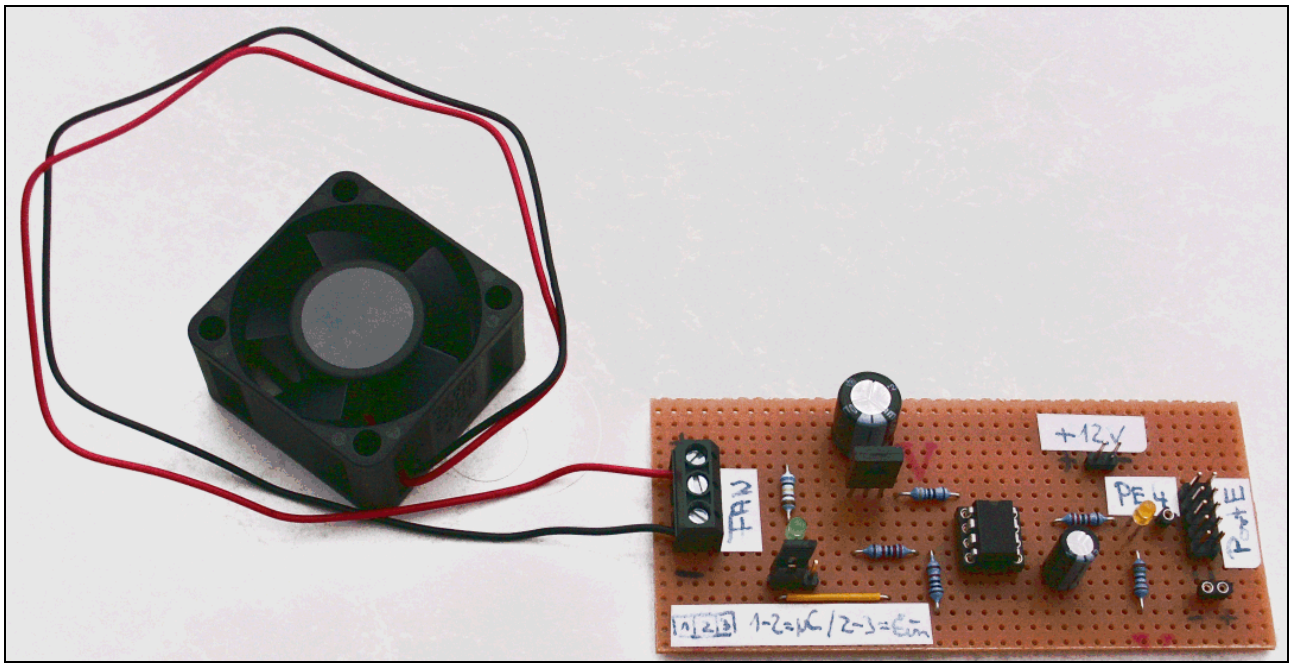


Abbildung 94: Prototyp Lüftersteuerung mit Lüfter

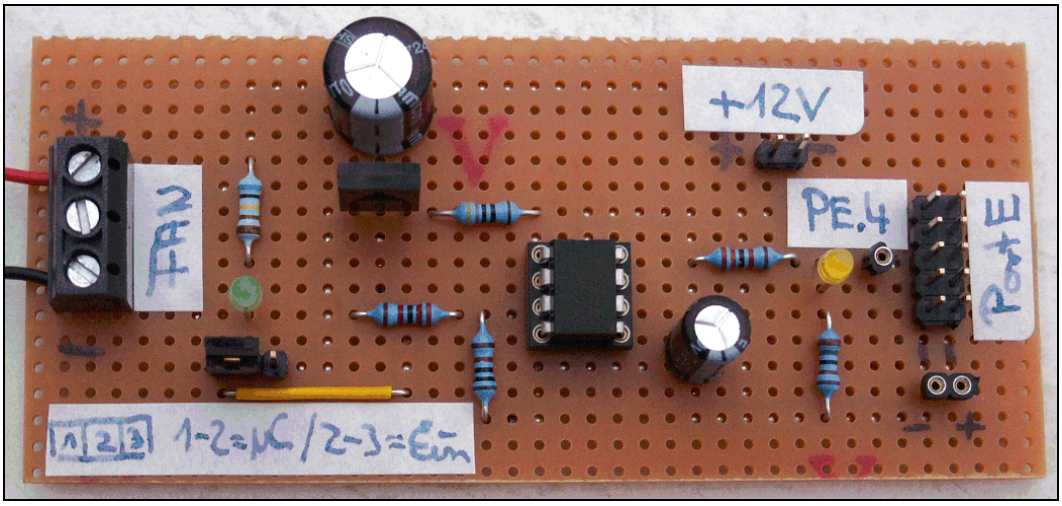


Abbildung 95: Prototyp Lüftersteuerung Detailansicht

8.13 Windgeber

Am System wird ein Windgeber angeschlossen welcher die Windgeschwindigkeit erfasst, anzeigt und ggf. ab einer einstellbaren Schwelle die Beschattung automatisch auffährt.

Bei dem Windgeber handelt es sich um einen kugelgelagerten Kleinwindgeber der Firma rb-Messtechnik Reinhard. Der Wingeber ist mit Kugellager und Reedkontakt ausgeführt.



Abbildung 96: Kleinwindgeber der Firma rb-Messtechnik

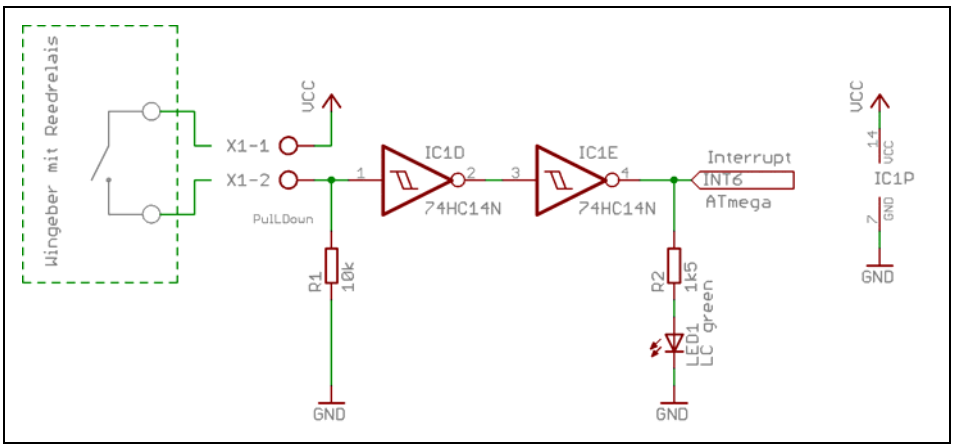
Beschreibung:

- Schalendurchmesser 120 mm
- Ausgang Reedkontakt
- 1 Impuls pro Sekunde entspricht 2,5 km/h → 0,6944 m/sec
- Schaltleistung max. 24V 5W

8.13.1 Beschaltung Windgeber

Beschaltung:

Mittels des Reedkontakts kann eine einfache Anbindung an den ATmega der folgenden Form erfolgen:



Schaltbild 33: Beschaltung Windgeber

Um das vom Windgeber gelieferte Impulssignal zu bereinigen wird es über zwei NAND-Schmitt-Trigger geführt die wiederum eine saubere Flanke zur Interrupterkennung für den ATmega erzeugen.

Bauteile:

Stückliste: Lüfteransteuerung			
Sonsitges		***Halbleiter***	
X1	Schraubklemme, Raster 5mm	IC1A, IC1B	NAND-Schnmittrigger 74HC14N
Kleinwindgeber der Firma rb-Messtechnik Reinhardt		LED1	Low current LED grün/gelb
		Widerstand	
		R1	Metallschicht-Widerstand 10k Ω
		R2	Metallschicht-Widerstand 1k5k Ω

Tabelle 105: Stückliste Windgeberbeschaltung

Themenbezogene LED's:

LED / Funktion / Beschreibung	*Abkürzung*	*Farbe*	
Status-LED für Windimpuls	LED_WIND_PULS	gelb	*[LED28]*

Tabelle 106: Themenbezogene LED's Windimpuls

Ressourcenzuordnung zum ATmega128:

Nummer	*Schaltbild*	*Ressource ATmega128*
1	INT6 [INT_WIND]	PortE.6 [INT6] (Interrupt)

Tabelle 107: Ressourcenzuordnung ATmega128 für Lüftersteuerung

8.13.2 BASCOM Beispielcode für Auswertung Windgeber

```
'#################################################################################
' WINDMESUNG.BAS:                                          Stand 01.02.2008
' ---------------                                          (C) Markus Fulde
'
' Testprogramm zur Inbetriebnahme des Schalenkreuzanemometers und der dafür
' vorgesehenen Eingangsbeschaltung für ATmega128.
' Die Inbetriebnahme erfolgt mit dem Eva-Board STK500 + STK501
' Die für die Berechnung der Windgeschwindigkeit notwendigen Routinen sind
' als solche gekennzeichnet.
'#################################################################################


'--------------------------------------------------------------------------------
' Compilerinstruktionen und Compilerdirektiven
```

```
'-----------------------------------------------------------------------------
$regfile = "m128def.dat"
$crystal = 16000000
$hwstack = 32
$swstack = 32
$framesize = 40

$baud = 38400


'-----------------------------------------------------------------------------
' Definition von Ressourcen
'-----------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========


'========== Für Testumgebung  ==========
Led_1 Alias Portc.0
Led_1_pin Alias Pinc.0


'-----------------------------------------------------------------------------
' Definition von Variablen, Konstaten und Datentypen
'-----------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========
Const Wind_testmodus = 1                                    ' Testmodus Projekt 1 = aktiv
Const Wind_auswertung = 5                                   ' Auswerteintervall in Sekunden
Const Wind_impuls_geschwindigkeit = 2.5                     ' Ein Impuls entspricht 2,5 km/h Wind
Const True = 1                                              ' Boolsche Variable für JA
Const False = 0                                             ' Boolsche Variable für NEIN

Dim Wind_sensorimpulse As Byte                              ' Zähler für Windimpulse
Dim Wind_geschwindigkeit_kmh As Single                      ' Variable für Windgeschwindigkeit in
km/h
Dim Wind_geschwindigkeit_ms As Single                       ' Variable für Windgeschwindigkeit in
m/s
Dim Wind_erfassung As Byte                                  ' Globale Varibale zum Status der Erfas-
sung

Dim Wind_tempbyte_1 As Byte                                 ' Temporärvariable für Windberechung
Byte


'========== Für Testumgebung ==========
Const Main_testmodus = 1                                    ' Testmodus Testumgebung 1 = aktiv

' Alias Namen für Tastaturabfragen
Taste_1 Alias Pina.0
Taste_2 Alias Pina.1
Taste_3 Alias Pina.2
Taste_4 Alias Pina.3
Taste_5 Alias Pina.4
Taste_6 Alias Pina.5
Taste_7 Alias Pina.6
Taste_8 Alias Pina.7

' Globale COunter-Variable für Auswertung
Dim Sec_counter As Byte

' Timercounter für 1 Sek.
Const Timervorgabe = 3036

' Ergebissstring Output
Dim Temp_string1 As String * 20


'-----------------------------------------------------------------------------
' Prototyping
'-----------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========


'========== Für Testumgebung ==========




'-----------------------------------------------------------------------------
' Konfiguration und Basiseinstellungen (Projekt und Testumgebung)
'-----------------------------------------------------------------------------

'========== Für Testumgebung ==========
' Konfiguration des 1 Sekunden-Timers für LED-Toggle-Funktion
Config Timer1 = Timer , Prescale = 256
On Timer1 Timer_irq
Sec_counter = 0                                             ' Gobale Counter-Variable auf NULL
```

```
' Timer aktivieren und Interrupts freigeben
Enable Timer1
Enable Interrupts

' Konfiguration des Pin C.0 als Ausgang für LED-Kontrolleuchten
Config Led_1_pin = Output

'Konfiguration des Port A als Eingang für Tastaturabfrage
Config Porta = Input

'========== Für Projekt ConvCtrl ==========

' INT6 so konfiguriert das Trigger bei steigender Flanke
Config Int6 = Rising

' Springe zur Interrupt-Routine Wind_irq
On Int6 Wind_irq

Gosub Wind_konfig


'========== Für Testumgebung ==========

#if Main_testmodus
   Print "** Enter MainLoop **"
   Print
#endif


'#####################################################################################
' Hauptprogramm
'#####################################################################################
'Hier ist die Programmhauptschleife
Do

   ' Tastaturabfrage Taste 1
   Debounce Taste_1 , 0 , Wind_erfassung_an , Sub

   ' Tastaturabfrage Taste 2
   Debounce Taste_2 , 0 , Wind_erfassung_aus , Sub


   ' Prüfen ob Auswerteinterfall abgelaufen
   If Sec_counter = Wind_auswertung Then
      ' Wenn ja dann Auswertung starten und Windgeschwindigkeit berechnen

      #if Main_testmodus
         Print "Erfassung     : " ; Wind_erfassung
         Print "Sec_counter   : " ; Sec_counter
         Print "Sensorimpulse : " ; Wind_sensorimpulse
      #endif

      ' Windgeschwindigkeit berechnen
      Gosub Wind_measurement

      ' Auswerteinterfall zurücksetzen
      Sec_counter = 0

      #if Main_testmodus
         Temp_string1 = Fusing(wind_geschwindigkeit_kmh , "###.##")
         Print "Wind in km/h  : " ; Temp_string1

         Temp_string1 = Fusing(wind_geschwindigkeit_ms , "###.##")
         Print "Wind in m/s   : " ; Temp_string1

         Print
      #endif


   End If

Loop
'## End Hauptprogramm ###############################################################


'*************************************************************************************
' Interruptroutinen
'*************************************************************************************

'-------------------------------------------------------------------------------------
' Interrupt-Service-Routine (Timer1): Timer_irq
' Routine zur Auswertung des Timer Interrupts
'-------------------------------------------------------------------------------------
Timer_irq:

   ' Timervariable neu setzen
   Timer1 = Timervorgabe
   Toggle Led_1

   ' globale Counter-Variable um Eins erhöhen
```

```
   Incr Sec_counter

Return
'-- End Timer_irq ------------------------------------------------------------


'--------------------------------------------------------------------------------
' Interrupt-Service-Routine (INT6): Wind_irq
' Routine zur Auswertung des INT6 Interrupts für Impulseingang Windsensor
'--------------------------------------------------------------------------------
Wind_irq:
   ' Impulszähler erhöhen
   Incr Wind_sensorimpulse
Return
'-- End Timer_irq ------------------------------------------------------------



'********************************************************************************
' Subroutinen
'********************************************************************************

' gibts keine

'********************************************************************************
' Funktionen
'********************************************************************************

' gibts keine

'********************************************************************************
' GOSubroutinen
'********************************************************************************

'--------------------------------------------------------------------------------
' WIND - Gosub-Routine: Wind_konfig
' Subroutine zur Konfiguration der für die Windmessung notwendigen Datenstruk-
' turen, Timer usw.
'--------------------------------------------------------------------------------
Wind_konfig:

   #if Wind_testmodus
      Print "Enter Wind_konfig"
   #endif

   Wind_sensorimpulse = 0

Return
'-- End Wind_konfig ------------------------------------------------------------


'--------------------------------------------------------------------------------
' WIND - Gosub-Routine: Wind_measurement
' Subroutine zur Berechung der aktuellen Windgeschwindigkeit im Auswerteinter-
' vall für km/ ud m/s.
' Die Ablage der errechneten Werte erfolgt in globalen Variablen
'--------------------------------------------------------------------------------
Wind_measurement:

   #if Wind_testmodus
      Print "Enter Wind_measurement"
   #endif

   ' Interrupt während Auslesen von Zähler wegen "Kollisionsgefahr" sperren
   Disable Int6                                          ' Interrupt 6 sperren

   Wind_tempbyte_1 = Wind_sensorimpulse                  ' Variable auslesen
   Wind_sensorimpulse = 0                                ' Variable zurücksetzen

   ' Interrupt nur freigeben wenn Erfassung auch gewünscht
   If Wind_erfassung = True Then
      Enable Int6                                        ' Interrupt 6 wieder freigeben
   End If

   ' Berechung der Windgeschwindigkeit in km/h
   Wind_geschwindigkeit_kmh = Wind_tempbyte_1 / Wind_auswertung
   Wind_geschwindigkeit_kmh = Wind_geschwindigkeit_kmh * Wind_impuls_geschwindigkeit

   ' Umrechung von km/h in m/s: Wert * 1000 / 3600 => Wert / 3,6
   Wind_geschwindigkeit_ms = Wind_geschwindigkeit_kmh / 3.6

Return
'-- End Wind_measurement ------------------------------------------------------


'--------------------------------------------------------------------------------
' WIND - Gosub-Routine: Wind_erfassung_an
' Subroutine aktiviert den Interrupt zur Erfassung der "Windimpulse"
'--------------------------------------------------------------------------------
Wind_erfassung_an:
```

```
   #if Wind_testmodus
      Print "Enter Wind_erfassung_an"
   #endif

   ' Interrupt freigeben -> Erfassung aktiv
   Enable Int6
   Wind_erfassung = True
   Wind_sensorimpulse = 0

Return
'-- End Wind_erfassung_an ------------------------------------------------------


'-------------------------------------------------------------------------------
' WIND - Gosub-Routine: Wind_erfassung_aus
' Subroutine deaktiviert den Interrupt zur Erfassung der "Windimpulse"
'-------------------------------------------------------------------------------
Wind_erfassung_aus:

   #if Wind_testmodus
      Print "Enter Wind_erfassung_aus"
   #endif

   ' Interrupt sperren -> Erfassung ausgeschaltet
   Disable Int6
   Wind_erfassung = False
   Wind_sensorimpulse = 0

Return
'-- End Wind_erfassung_an ------------------------------------------------------


End                                                    'end program


'-------------------------------------------------------------------------------
' Definition von globalen Konstantenfeldern
'-------------------------------------------------------------------------------

' gibt es keine

'##################################### END ######################################
```

Software 8: Code zum Betrieb des Kleinwindgebers

8.13.3 Prototyp Windgeberansteuerung



Abbildung 97: Kleinwindgeber mit Beschaltung

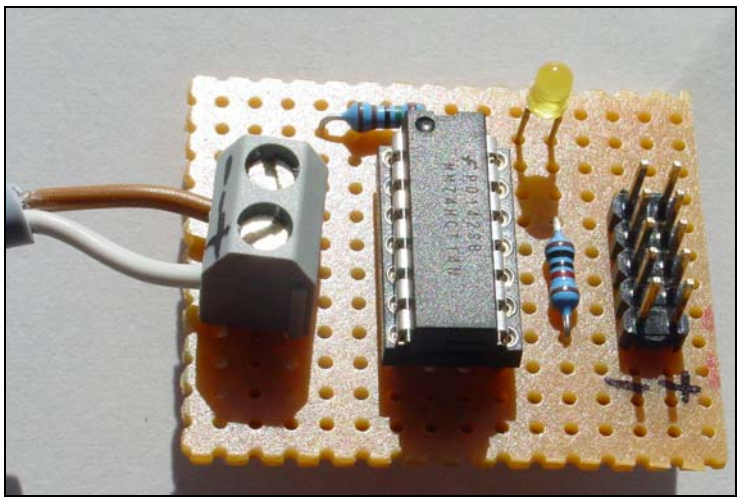


Abbildung 98: Beschaltung des Kleinwindgebers

8.14 Beschaltung Regensensor

Zur Regenerkennung wird im Projekt ConvCtrl der Regensensor von Conrad Electronik, Bestellnummer 11 52 40 in einem speziell dafür geeigneten Gehäuse verwendet.

Dieses Kapitel teilt sich auf in die Beschreibung des Regensensors und die die Beschreibung der Ankopplung an den ATmega.

8.14.1 Conrad Regensensor:

CONRAD IM INTERNET http://www.conrad.com

BEDIENUNGSANLEITUNG

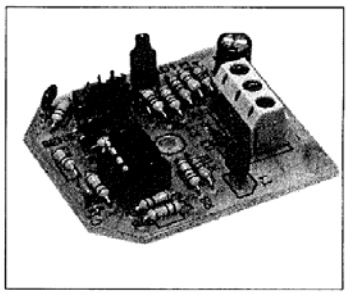
Version 12/03

CE

Regensensor

❑ Best.-Nr.: 11 52 40 Bausatz

❑ Best.-Nr.: 11 52 74 Fertiggerät im Gehäuse



Impressum

Diese Bedienungsanleitung ist eine Publikation der Conrad Electronic GmbH, Klaus-Conrad-Straße 1, D-92240 Hirschau.

Alle Rechte einschließlich Übersetzung vorbehalten. Reproduktionen jeder Art, z. B. Fotokopie, Mikroverfilmung, oder die Erfassung in EDV-Anlagen, bedürfen der schriftlichen Genehmigung des Herausgebers.

Nachdruck, auch auszugsweise, verboten.

Diese Bedienungsanleitung entspricht dem technischen Stand bei Drucklegung. Änderung in Technik und Ausstattung vorbehalten.

100% Recycling-papier.

Chlorfrei gebleicht.

© Copyright 2000 by Conrad Electronic GmbH. Printed in Germany.

*12-03/MZ

4 016138 115246

CONRAD ELECTRONIC

Wichtig! Unbedingt lesen!

Lesen Sie diese Anleitung sorgfältig durch. Bei Schäden, die durch Nichtbeachtung der Bedienungsanleitung entstehen, erlischt der Garantieanspruch. Für Folgeschäden, die daraus resultieren, übernehmen wir keine Haftung.

Inhaltsverzeichnis

	Seite
Betriebsbedingungen	3
Bestimmungsgemäße Verwendung	5
Sicherheitshinweis	5
Produktbeschreibung	7
Schaltungsbeschreibung	8
Technische Daten	12
Allgemeiner Hinweis zum Aufbau einer Schaltung	13
Lötanleitung	15
1. Baustufe I	18
2. Baustufe II	25
Schaltplan	26
Bestückungspläne	27
3. Baustufe III	28
Checkliste zur Fehlersuche	30
Tips für den Gehäuseeinbau	33
Störung	33
Garantie	34

Hinweis (Bausatz)

Der Bausatz darf nur von einer mit der Materie vertrauten Fachkraft aufgebaut und in Betrieb genommen werden!
Derjenige, der einen Bausatz fertigstellt oder eine Baugruppe durch Erweiterung bzw. Gehäuseeinbau betriebsbereit macht, gilt nach DIN VDE 0869 als Hersteller und ist verpflichtet, bei der Weitergabe des Gerätes alle Begleitpapiere mitzuliefern und auch seinen Namen und seine Anschrift anzugeben. Geräte, die aus Bausätzen selbst zusammengestellt werden, sind sicherheitstechnisch wie ein industrielles Produkt zu betrachten.

Hinweis (Fertiggerät)

Dieses Gerät hat das Werk in sicherheitstechnisch einwandfreiem Zustand verlassen. Um diesen Zustand zu erhalten und einen gefahrlosen Betrieb sicherzustellen, muß der Anwender die Sicherheitshinweise und Warnvermerke, die in dieser Anleitung enthalten sind, beachten!

Betriebsbedingungen

- Der Betrieb der Baugruppe darf nur an der dafür vorgeschriebenen Spannung erfolgen.
- Bei Geräten mit einer Betriebsspannung ≥ 35 Volt darf die Endmontage nur vom Fachmann unter Einhaltung der VDE-Bestimmungen vorgenommen werden.
- Das Gerät sollte so im Freien montiert werden, daß die Sensorfläche nach oben zeigt und um ca. 45° geneigt ist. So kann der Regen gut abfließen und sich keine „Pfützen" auf der Sensorfläche bilden.
- Es ist unbedingt auf die Einhaltung, der in dieser Anleitung an-

gegebenen technischen Daten zu achten. Das Überschreiten dieser Werte kann zu Schäden am Gerät oder Verbraucher führen.

- Bei der Installation des Gerätes ist auf ausreichenden Kabelquerschnitt der Anschlußleitungen zu achten!
- Die angeschlossenen Verbraucher sind entsprechend den VDE-Vorschriften zu verbinden.
- Das Gerät darf nicht in Verbindung mit leicht entflammbaren und brennbaren Flüssigkeiten verwendet werden!
- Baugruppen und Bauteile gehören nicht in Kinderhände!
- Die Baugruppen dürfen nur unter Aufsicht eines fachkundigen Erwachsenen oder eines Fachmannes in Betrieb genommen werden!
- In gewerblichen Einrichtungen sind die Unfallverhütungsvorschriften des Verbandes der gewerblichen Berufsgenossenschaften für elektrische Anlagen und Betriebsmittel zu beachten.
- In Schulen, Ausbildungseinrichtungen, Hobby- und Selbsthilfewerkstätten ist das Betreiben von Baugruppen durch geschultes Personal verantwortlich zu überwachen.
- Betreiben Sie die Baugruppe nicht in einer Umgebung, in welcher brennbare Gase, Dämpfe oder Stäube vorhanden sind oder vorhanden sein können.
- Falls das Gerät einmal repariert werden muß, dürfen nur Original-Ersatzteile verwendet werden! Die Verwendung abweichender Ersatzteile kann zu ernsthaften Sach- und Personenschäden führen!
- Eine Reparatur des Gerätes darf nur vom Fachmann durchgeführt werden!

- Das Gerät ist nach Gebrauch stets von der Versorgungsspannung zu trennen!
- Dringt irgendeine Flüssigkeit in das Gerät ein, so könnte es dadurch beschädigt werden. Sollten Sie irgendwelche Flüssigkeiten in, oder über die Baugruppe verschüttet haben, so muß das Gerät von einem qualifizierten Fachmann überprüft werden.

Bestimmungsgemäße Verwendung

Der bestimmungsgemäße Einsatz des Regensensors ist das Erfassen und Melden von Regen.
Ein anderer Einsatz als vorgegeben ist nicht zulässig!

Sicherheitshinweis

Beim Umgang mit Produkten, die mit elektrischer Spannung in Berührung kommen, müssen die gültigen VDE-Vorschriften beachtet werden, insbesondere VDE 0100, VDE 0550/0551, VDE 0700, VDE 0711 und VDE 0860.

- Vor Öffnen eines Gerätes stets den Netzstecker ziehen oder sicherstellen, daß das Gerät stromlos ist.
- Bauteile, Baugruppen oder Geräte dürfen nur in Betrieb genommen werden, wenn sie vorher berührungssicher in ein Gehäuse eingebaut wurden. Während des Einbaus müssen sie stromlos sein.
- Werkzeuge dürfen an Geräten, Bauteilen oder Baugruppen nur benutzt werden, wenn sichergestellt ist, daß die Geräte von der Versorgungsspannung getrennt sind und elektrische Ladungen, die in den im Gerät befindlichen Bauteilen gespeichert sind, vorher entladen wurden.

- Spannungsführende Kabel oder Leitungen, mit denen das Gerät, das Bauteil oder die Baugruppe verbunden ist, müssen stets auf Isolationsfehler oder Bruchstellen untersucht werden. Bei Feststellen eines Fehlers in der Zuleitung muß das Gerät unverzüglich aus dem Betrieb genommen werden, bis die defekte Leitung ausgewechselt worden ist.
- Bei Einsatz von Bauelementen oder Baugruppen muß stets auf die strikte Einhaltung der in der zugehörigen Beschreibung genannten Kenndaten für elektrische Größen hingewiesen werden.
- Wenn aus einer vorliegenden Beschreibung für den nichtgewerblichen Endverbraucher nicht eindeutig hervorgeht, welche elektrischen Kennwerte für ein Bauteil oder eine Baugruppe gelten, wie eine externe Beschaltung durchzuführen ist oder welche externen Bauteile oder Zusatzgeräte angeschlossen werden dürfen und welche Anschlußwerte diese externen Komponenten haben dürfen, so muß stets ein Fachmann um Auskunft ersucht werden.
- Es ist vor der Inbetriebnahme eines Gerätes generell zu prüfen, ob dieses Gerät oder Baugruppe grundsätzlich für den Anwendungsfall, für den es verwendet werden soll, geeignet ist!
  Im Zweifelsfalle sind unbedingt Rückfragen bei Fachleuten, Sachverständigen oder den Herstellern der verwendeten Baugruppen notwendig!
- Bitte beachten Sie, daß Bedien- und Anschlußfehler außerhalb unseres Einflußbereiches liegen. Verständlicherweise können wir für Schäden, die daraus entstehen, keinerlei Haftung übernehmen.
- Bausätze sollten bei Nichtfunktion mit einer genauen Fehlerbeschreibung (Angabe dessen, was nicht funktioniert... denn nur eine exakte Fehlerbeschreibung ermöglicht eine einwandfreie Reparatur!) und der zugehörigen Bauanleitung sowie ohne Gehäuse zurückgesandt werden. Zeitaufwendige Montagen oder Demontagen von Gehäusen müssen wir aus verständlichen Gründen zusätzlich berechnen. Bereits aufgebaute Bausätze sind vom Umtausch ausgeschlossen. Bei Installationen und beim Umgang mit Netzspannung sind unbedingt die VDE-Vorschriften zu beachten.

- Geräte, die an einer Spannung ≥ 35 V betrieben werden, dürfen nur vom Fachmann angeschlossen werden.
- In jedem Fall ist zu prüfen, ob der Bausatz für den jeweiligen Anwendungsfall und Einsatzort geeignet ist bzw. eingesetzt werden kann.
- Die Inbetriebnahme darf grundsätzlich nur erfolgen, wenn die Schaltung absolut berührungssicher in ein Gehäuse eingebaut ist.
- Sind Messungen bei geöffnetem Gehäuse unumgänglich, so muß aus Sicherheitsgründen ein Trenntrafo zwischengeschaltet werden, oder, wie bereits erwähnt, die Spannung über ein geeignetes Netzteil, (das den Sicherheitsbestimmungen entspricht) zugeführt werden.
- Alle Verdrahtungsarbeiten dürfen nur im spannungslosen Zustand ausgeführt werden.

Produktbeschreibung

Mit Hilfe dieser Regensensorik, die auf Regen, Schnee und Hagel reagiert, kann in Verbindung mit einer entsprechenden Elektronik bei Regen z. B. der Sonnenschutz oder die Markise angefahren werden, Dachflächenfenster geschlossen werden oder ganz einfach nur Regen gemeldet werden. Eine automatische Heizung

© Markus Fulde, 2007

verhindert zuverlässig z. B. Vereisung oder Betauung und bietet damit eine funktionssichere Niederschlagsdetektion.

Schaltungsbeschreibung

Genau genommen haben wir es hier mit einer Schaltung zur Flüssigkeits-Erkennung zu tun. Durch die Dimensionierung der externen Komponenten läßt sich aber eine Empfindlichkeit erreichen, die die Bezeichnung Feuchtesensor rechtfertigt; es braucht also kein Gewitterschauer zu sein, der zum Ansprechen führt, sondern es reichen dazu schon ein paar Regentropfen oder der Tau am Morgen aus.

Das Funktionsprinzip ist recht simpel: Zwei Fühler detektieren die Feuchtigkeit, und das IC vergleicht den Widerstand zwischen den beiden Sensordrähten mit einem Bezugswiderstand. Wenn in diesem Zusammenhang von leitenden Flüssigkeiten die Rede ist, dann bezieht sich das auf deren ohmschen Durchgangswiderstand.

Unabhängig davon werden die Fühlerleitungen mit einem Wechselspannungssignal beaufschlagt. Damit vermeidet man Polarisationseffekte, die bei Gleichspannung auftreten, und die zu einem schleichenden Anstieg des Übergangswiderstandes führen. Dadurch ließe sich mit der Zeit nicht mehr unterscheiden, ob die Flüssigkeit nicht vorhanden ist oder ob bloß ihr Leitvermögen abgenommen hat.

Da man normalerweise kein getaktetes Ausgangssignal haben möchte, bietet das IC 1 hierfür eine einfache Lösung an: Sofern man am Pin 6 einen externen Filterkondensator un deine Diode anschließt, wird der Oszillatortakt darüber geglättet. Der Ausgang schaltet dann unterbrechungsfrei aus, sobald die Fühlerdrähte am Eingang keine Flüssigkeit detektieren.
Bei vorhandener Flüssigkeit (und niedrigem Eingangswiderstand) schaltet der Pin-7-Ausgang nach Plus gezogen. Im letztgenannten Fall schaltet der angeschlossene Treibertransistor T1 durch, was am Aufleuchten der LED zu verfolgen ist; das Lämpchen leuchtet also beim Leiten.

Anders als bei allen herkömmlichen Füllstandschaltern ist hier noch eine Besonderheit vorhanden: Bei aktivem T1 bekommt der angeschlossene Transistor T2 über R12 Basisstrom und leitet ebenfalls. Damit legt er den mit 'Heizung' (+H) bezeichneten Platinen-Ausgang an Plus, so daß die daran angeschlossenen zehn SMD-Widerstände R6...15 stromdurchflossen werden.

Diese 1,2-kΩ-Widerstände befinden sich auf der Rückseite der Fühlerplatine und erzeugen an 12 V eine Verlustleistung von 120 mW pro Stück. Zusammen sind das also 1,2 W, die zur Heizung des Fühlers benutzt werden, sobald der einmal feucht bzw. naß geworden ist. Damit wird er gleichermaßen vor Betauung wie auch vor Frost geschützt und ist nach kurzer Zeit stets wieder betriebsbereit.

Die Heizung wurde folgendermaßen realisiert: Auf der Unterseite der Fühlerplatine sind die zehn SMD-Widerstände erkennbar, die alle mit einem Bein an Masse liegen (GND) und über T2 zum Heizen an Plus geschaltet werden. Der eigentliche Fühler befindet sich auf der Oberseite dieser Platine. Zur Erhöhung der Sensibilität besteht er nicht nur aus zwei einzelnen Leiterbahnen, sondern aus einer kammartig ineinander verschachtelten Struktur. Damit genügt bereits das bloße Handauflegen, um den Sensor auszulösen.

Die Leiterbahnen auf der Vorderseite sind vergoldet, damit sie einen Schutz gegen Korrosion besitzen. Die Durchkontaktierungen zur Rückseite sind aber mit Lötstopplack abgedeckt, damit hier keine Feuchtigkeit ins Gehäuse-Innere eindringen kann.

Beim vorgeschlagenen Konzept wird die integrierte Heizung nur dann eingeschaltet, wenn der Sensor auf Feuchtigkeit bzw. Nässe

reagiert. Wer die Heizung ständig in Betrieb lassen möchte (z.B. bei sehr hoher Luftfeuchte), kann bei T2 den Kollektor und Emitter kurzschließen; in diesem Fall fließt allerdings ständig ein Strom von ca. 125 mA, während er ansonsten im Ruhezustand auf unter 10 mA zurückgeht.

Nachbau

Anhand des Bestückungsplans und der Stückliste dürfte der Nachbau keine großen Probleme aufwerfen. Bestücken Sie zunächst die Basisplatine HB 461 und beginnen Sie dort mit den fünf Widerständen. Es empfiehlt sich immer wieder, auch die Widerstände einheitlich auszurichten, also die goldenen Toleranzringe z.B. stets nach rechts oder nach unten einzulöten. Obwohl dafür keine technische Notwendigkeit besteht, vereinfacht sich dadurch die spätere Sichtkontrolle.

Die beiden Kleinsignaldioden müssen allerdings wie vorgeschrieben eingebaut werden, d.h. D1 zeigt mit dem schwarzen Katodenring in Richtung R9, und D2 zeigt von der Klemme weg. Während D1 zur Entkopplung von Sensor- und Lastkreis dient, sorgt D3 für das Kurzschließen von Induktionsspitzen, wenn an Out ein Relais angeschlossen ist.
Das IC bekommt natürlich eine Fassung, deren Markierungskerbe zur Leuchtdiode zeigt. Nach dem Einlöten der drei keramischen Kondensatoren kommen die beiden Elkos an die Reihe; dabei ist wieder die richtige Polung zu beachten. Auch die Leuchtdiode darf man nicht falschherum einsetzen: Hier ist ebenfalls der Plus-Anschluß länger, der die Anodenseite kennzeichnet; die mit dem Querstrich versehene Katode hat das kürzere Bein, das im Bestückungsplan mit dem weißen Balken markiert ist (liegt auf der D1-Seite).

Beim Einlöten der dreipoligen Schraubklemme ist auf jeden Fall darauf zu achten, daß man mit heißem Kolben und ausreichendem, frischen Lötzinn zu Werke geht; andernfalls sind hier kalte Lötstellen geradezu vorprogrammiert.

Nach Abschluß dieser Arbeiten wird das IC eingesetzt, wozu man vorher die auseinandergespreizten Beinchen erst einmal parallel ausrichtet. Wichtig sind außerdem die richtige Lage (Markierungskerbe zu R1 hin zeigend) und die Vorsicht davor, daß keins der Beinchen umknickt. Es gibt zwar Pins, die nicht belegt sind, aber wenn einer umknickt, ist es immer der mit der wichtigsten Funktion...

Inbetriebnahme

Sie können in dieser Phase einen ersten Funktionstest vornehmen, indem Sie an +Uv und GND ca. 12 V anschließen und an der dreipoligen Stiftleiste die beiden äußeren Pins mit ca. 10 kΩ überbrücken. Damit simulieren Sie für das IC Feuchtigkeit, was dies mit dem Einschalten der roten LED quittieren muß.

Abschließend machen Sie sich an die Platine HB 461A und bestücken auf deren Rückseite die zehn kleinen SMD-Widerstände. Das geht unproblematischer vor sich, als es häufig angenommen wird: Sie brauchen für jeden Widerstand nur eins der beiden Lötpads vorzuverzinnen, das Bauteil mit dem Fingernagel zu plazieren und festzuhalten; nach dem Anschmelzen der vorverzinnten Stelle hält der Widerstand schon fest, so daß Sie sein anderes Bein auch noch anlöten können.

Die Verbindung zur Basisplatine stellen Sie mit einem dreiadrigen, steckbaren Kabel her. Dieses Kabel stecken Sie durch die Öffnung des Gehäusedeckels und streichen die Fühlerplatine mit Silikonkautschuk ein, ehe Sie sie am vorgesehenen Platz einsetzen.

Auch die Rückseite des Gehäusedeckels müssen Sie mit Silikon ausfüllen, damit es keine undichten Stellen gibt. Beachten Sie bitte, daß die Fühler-Kämme nach unten hin offen sind, damit die Wassertropfen dort abfließen können.

© Markus Fulde, 2007

Das vorgesehene, fertig gefräste Gehäuse hat eine innen umlaufende Dichtlippe, die es gegen Spritzwasser abdichtet. Auch die Versorgung sollte über eine dichtende Kabeldurchführung erfolgen. Einen abschließenden Test können Sie übrigens ganz einfach vornehmen: Sprühen Sie mit einer Haushaltsflasche einen dünnen Feuchtigkeitsfilm auf den Fühler und kontrollieren Sie das Ansprechen des Sensors; nach kurzer Zeit muß die SMD-Heizung wieder für Trockenheit sorgen.

Technische Daten:

Versorgungsspannung . : 11 ... 15 V=

Stromaufnahme . . . . . : 6...8 mA (Ruhe), 125 mA (mit Heizung)

Schaltausgang . . . . . : bis max. 100 mA belastbar (zum Steuern von Relais oder entsprechenden Elektroniken)

Abmessungen . . . . . : 40 x 45 mm (Basisplatine) 65 x 50 x 35 mm (im Gehäuse)

passendes Gehäuse, bearbeitet: Best.-Nr. 11 53 47.

Achtung!

Bevor Sie mit dem Nachbau beginnen, lesen Sie diese Bauanleitung erst einmal bis zum Ende in Ruhe durch, bevor Sie den Bausatz oder das Gerät in Betrieb nehmen (besonders den Abschnitt über die Fehlermöglichkeiten und deren Beseitigung!) und natürlich die Sicherheitshinweise. Sie wissen dann, worauf es ankommt und was Sie beachten müssen und vermeiden dadurch von vornherein Fehler, die manchmal nur mit viel Aufwand wieder zu beheben sind!

Führen Sie die Lötungen und Verdrahtungen absolut sauber und gewissenhaft aus, verwenden Sie kein säurehaltiges Lötzinn,

Lötfett o. ä. Vergewissern Sie sich, daß keine kalte Lötstelle vorhanden ist. Denn eine unsaubere Lötung oder schlechte Lötstelle, ein Wackelkontakt oder schlechter Aufbau bedeuten eine aufwendige und zeitraubende Fehlersuche und unter Umständen eine Zerstörung von Bauelementen, was oft eine Kettenreaktion nach sich zieht und der komplette Bausatz zerstört wird.

Beachten Sie auch, daß Bausätze, die mit säurehaltigem Lötzinn, Lötfett o. ä. gelötet wurden, von uns nicht repariert werden.

Beim Nachbau elektronischer Schaltungen werden Grundkenntnisse über die Behandlung der Bauteile, Löten und der Umgang mit elektronischen bzw. elektrischen Bauteilen vorausgesetzt.

Allgemeiner Hinweis zum Aufbau einer Schaltung

Die Möglichkeit, daß nach dem Zusammenbau etwas nicht funktioniert, läßt sich durch einen gewissenhaften und sauberen Aufbau drastisch verringern. Kontrollieren Sie jeden Schritt, jede Lötstelle zweimal, bevor Sie weitergehen! Halten Sie sich an die Bauanleitung! Machen Sie den dort beschriebenen Schritt nicht anders und überspringen Sie nichts! Haken Sie jeden Schritt doppelt ab: einmal fürs Bauen, einmal fürs Prüfen.
Nehmen Sie sich auf jeden Fall Zeit: Basteln ist keine Akkordarbeit, denn die hier aufgewendete Zeit ist um das dreifache geringer als jene bei der Fehlersuche.

Eine häufige Ursache für eine Nichtfunktion ist ein Bestückungsfehler, z. B. verkehrt eingesetzte Bauteile wie ICs, Dioden und Elkos. Beachten Sie auch unbedingt die Farbringe der Widerstände, da manche leicht verwechselbare Farbringe haben.

Achten Sie auch auf die Kondensator-Werte z. B. n 10 = 100 pF (nicht 10 nF). Dagegen hilft doppeltes und dreifaches Prüfen. Achten Sie auch darauf, daß alle IC-Beinchen wirklich in der Fassung stecken. Es passiert sehr leicht, daß sich eines beim Ein-

stecken umbiegt. Ein kleiner Druck, und das IC muß fast von selbst in die Fassung springen. Tut es das nicht, ist sehr wahrscheinlich ein Beinchen verbogen.

Stimmt hier alles, dann ist als nächstes eventuell die Schuld bei einer kalten Lötstelle zu suchen. Diese unangenehmen Begleiter des Bastlerlebens treten dann auf, wenn entweder die Lötstelle nicht richtig erwärmt wurde, so daß das Zinn mit den Leitungen keinen richtigen Kontakt hat, oder wenn man beim Abkühlen die Verbindung gerade im Moment des Erstarrens bewegt hat. Derartige Fehler erkennt man meistens am matten Aussehen der Oberfläche der Lötstelle. Einzige Abhilfe ist, die Lötstelle nochmals nachzulöten.

Bei 90 % der reklamierten Bausätze handelt es sich um Lötfehler, kalte Lötstellen, falsches Lötzinn usw.. So manches zurückgesandte "Meisterstück" zeugte von nicht fachgerechtem Löten.

Verwenden Sie deshalb beim Löten nur Elektronik-Lötzinn mit der Bezeichnung "SN 60 Pb" (60 % Zinn und 40 % Blei). Dieses Lötzinn hat eine Kolophoniumseele, welche als Flußmittel dient, um die Lötstelle während des Lötens vor dem Oxydieren zu schützen. Andere Flußmittel wie Lötfett, Lötpaste oder Lötwasser dürfen auf keinen Fall verwendet werden, da sie säurehaltig sind. Diese Mittel können die Leiterplatte und Elektronik-Bauteile zerstören, außerdem leiten sie den Strom und verursachen dadurch Kriechströme und Kurzschlüsse.

Ist bis hierher alles in Ordnung und läuft die Sache trotzdem noch nicht, dann ist wahrscheinlich ein Bauelement defekt. Wenn Sie Elektronik-Anfänger sind, ist es in diesem Fall das Beste, Sie ziehen einen Bekannten zu Rate, der in Elektronik ein bißchen versiert ist und eventuell nötige Meßgeräte besitzt.

Sollten Sie diese Möglichkeit nicht haben, so schicken Sie den Bausatz bei Nichtfunktion gut verpackt und mit einer genauen

Fehlerbeschreibung, sowie der zugehörigen Bauanleitung an unsere Service-Abteilung ein (nur eine exakte Fehlerangabe ermöglicht eine einwandfreie Reparatur!). Eine genaue Fehlerbeschreibung ist wichtig, da der Fehler ja auch bei Ihrem Netzgerät oder Ihrer Außenbeschaltung sein kann.

Hinweis

Dieser Bausatz wurde, bevor er in Produktion ging, viele Male als Prototyp aufgebaut und getestet. Erst wenn eine optimale Qualität hinsichtlich Funktion und Betriebssicherheit erreicht ist, wird er für die Serie freigegeben.

Um eine gewisse Funktionssicherheit beim Bau der Anlage zu erreichen, wurde der gesamte Aufbau in 3 Baustufen aufgegliedert:

1. **Baustufe I : Montage der Bauelemente auf der Hauptplatine**
2. **Baustufe II : Montage der Bauelemente auf der Sensorplatine**
3. **Baustufe III: Stückprüfung/Anschluß/Inbetriebnahme**

Achten Sie beim Einlöten der Bauelemente darauf, daß diese (falls nicht Gegenteiliges vermerkt) ohne Abstand zur Platine eingelötet werden. Alle überstehenden Anschlußdrähte werden direkt über der Lötstelle abgeschnitten.

Da es sich bei diesem Bausatz teilweise um sehr kleine, bzw. eng beieinanderliegende Lötpunkte handelt (Lötbrückengefahr), darf hier nur mit einem Lötkolben mit kleiner Lötspitze gelötet werden. Führen Sie die Lötvorgänge und den Aufbau sorgfältig aus.

Lötanleitung

Wenn Sie im Löten noch nicht so geübt sind, lesen Sie bitte zuerst

diese Lötanleitung, bevor Sie zum Lötkolben greifen. Denn Löten will gelernt sein.

1. Verwenden Sie beim Löten von elektronischen Schaltungen grundsätzlich nie Lötwasser oder Lötfett. Diese enthalten eine Säure, die Bauteile und Leiterbahnen zerstört.

2. Als Lötmaterial darf nur Elektronikzinn SN 60 Pb (d. h. 60 % Zinn, 40 % Blei) mit einer Kolophoniumseele verwendet werden, die zugleich als Flußmittel dient.

3. Verwenden Sie einen kleinen Lötkolben mit max. 30 Watt Heizleistung. Die Lötspitze sollte zunderfrei sein, damit die Wärme gut abgeleitet werden kann. Das heißt: Die Wärme vom Lötkolben muß gut an die zu lötende Stelle geleitet werden.

4. Die Lötung selbst soll zügig vorgenommen werden, denn durch zu langes Löten werden Bauteile zerstört. Ebenso führt es zum Ablösen der Lötaugen oder Kupferbahnen.

5. Zum Löten wird die gut verzinnte Lötspitze so auf die Lötstelle gehalten, daß zugleich Bauteildraht und Leiterbahn berührt werden.
Gleichzeitig wird (nicht zuviel) Lötzinn zugeführt, das mit aufgeheizt wird. Sobald das Lötzinn zu fließen beginnt, nehmen Sie es von der Lötstelle fort. Dann warten Sie noch einen Augenblick, bis das zurückgebliebene Lot gut verlaufen ist und nehmen dann den Lötkolben von der Lötstelle ab.

6. Achten Sie darauf, daß das soeben gelötete Bauteil, nachdem Sie den Kolben abgenommen haben, ca. 5 Sek. nicht bewegt wird. Zurück bleibt dann eine silbrig glänzende, einwandfreie Lötstelle.

7. Voraussetzung für eine einwandfreie Lötstelle und gutes

Löten ist eine saubere, nicht oxydierte Lötspitze. Denn mit einer schmutzigen Lötspitze ist es absolut unmöglich, sauber zu löten. Nehmen Sie daher nach jedem Löten überflüssiges Lötzinn und Schmutz mit einem feuchten Schwamm oder einem Silikon-Abstreifer ab.

8. Nach dem Löten werden die Anschlußdrähte direkt über der Lötstelle mit einem Seitenschneider abgeschnitten.

9. Beim Einlöten von Halbleitern, LEDs und ICs ist besonders darauf zu achten, daß eine Lötzeit von ca. 5 Sek. nicht überschritten wird, da sonst das Bauteil zerstört wird. Ebenso ist bei diesen Bauteilen auf richtige Polung zu achten.

10. Nach dem Bestücken kontrollieren Sie grundsätzlich jede Schaltung noch einmal darauf hin, ob alle Bauteile richtig eingesetzt und gepolt sind. Prüfen Sie auch, ob nicht versehentlich Anschlüsse oder Leiterbahnen mit Zinn überbrückt wurden. Das kann nicht nur zur Fehlfunktion, sondern auch zur Zerstörung von teuren Bauteilen führen.

11. Beachten Sie bitte, daß unsachgemäße Lötstellen, falsche Anschlüsse, Fehlbedienung und Bestückungsfehler außerhalb unseres Einflußbereiches liegen.

Löten an SMD-Bauteilen

Die besten Lötergebnisse erzielt man, wenn vor dem Plazieren des Bauteils ein Lötpunkt auf der Platine leicht verzinnt wird. Nun wird mit einer Pinzette das Bauteil plaziert und mit dieser festgedrückt. Gleichzeitig wird das Bauelement und der vorher verzinnte Lötpunkt mit dem Lötkolben erhitzt, bis das Lötzinn sauber verflossen und eine gute Verbindung zwischen Bauteil und Lötpunkt hergestellt ist. Achten Sie darauf, daß der Lötvorgang nicht zu lange dauert, da ansonsten das Bauteil bzw. die Leiterbahn zerstört wird. Nach dem Abkühlen dieser Lötstelle ist

das Bauteil fixiert, und es werden die restlichen Anschlüsse verlötet.

Hier muß mit gleicher Sorgfalt vorgegangen werden. Das Lötzinn und die Lötspitze werden gleichzeitig an die Lötstelle gelegt (Bauelement und Lötpunkt gleichzeitig erhitzen), etwas Zinn abgeschmolzen und gewartet, bis das Zinn sauber verflossen ist, erst dann wird die Lötspitze von der Lötstelle genommen.

1. Baustufe I:

Montage der Bauelemente auf der Hauptplatine

1.1 Widerstände

Zuerst werden die Anschlußdrähte der Widerstände entsprechend dem Rastermaß rechtwinklig abgebogen und in die vorgesehenen Bohrungen (lt. Bestückungsplan) gesteckt. Damit die Bauteile beim Umdrehen der Platine nicht herausfallen können, biegen Sie die Anschlußdrähte der Widerstände ca. 45° auseinander, und verlöten diese dann sorgfältig mit den Leiterbahnen auf der Rückseite der Platine. Anschließend werden die überstehenden Drähte abgeschnitten.

Die hier in diesem Bausatz verwendeten Widerstände sind Kohleschicht-Widerstände. Diese haben eine Toleranz von 5% und sind durch einen goldfarbigen „Toleranz-Ring" gekennzeichnet. Kohleschicht-Widerstände besitzen normalerweise 4 Farbringe. Zum Ablesen des Farbcodes wird der Widerstand so gehalten, daß sich der goldfarbige Toleranzring auf der rechten Seite des Widerstandskörpers befindet. Die Farbringe werden dann von links nach rechts abgelesen!

R 1 =	100 k	braun,	schwarz,	gelb
R 2 =	100 k	braun,	schwarz,	gelb
R 3 =	100 k	braun,	schwarz,	gelb
R 4 =	100 k	braun,	schwarz,	gelb
R 5 =	100 k	braun,	schwarz,	gelb
R 6 =	100 k	braun,	schwarz,	gelb
R 7 =	1M	braun,	schwarz,	grün
R 8 =	150 k	braun,	grün,	gelb
R 9 =	1 M	braun,	schwarz,	grün
R 10=	4,7 k	gelb,	viloett,	rot
R 11=	1 k	braun,	schwarz,	rot
R 12=	4,7 k	gelb,	viloett,	rot
R 13=	47 k	gelb,	viloett,	orange
R 14=	10 Ω	braun,	schwarz,	schwarz

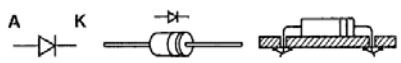


1.2 Dioden

Nun werden die Anschlußdrähte der Dioden entsprechend dem Rastermaß rechtwinklig abgebogen und in die vorgesehenen Bohrungen (lt. Bestückungsdruck) gesteckt. Achten Sie hierbei unbedingt darauf, daß die Dioden richtig gepolt eingebaut werden! Beachten Sie die Lage des Kathodenstriches!

Damit die Dioden beim Umdrehen der Platine nicht herausfallen können, biegen Sie die Anschlußdrähte ca. 45° auseinander, und verlöten diese bei kurzer Lötzeit mit den Leiterbahnen. Dann werden die überstehenden Drähte abgeschnitten.

D 1 = 1 N 4148	Silizium-Universaldiode
D 2 = 1 N 4148	Silizium-Universaldiode
D 3 = 1 N 4148	Silizium-Universaldiode



1.3 Kondensatoren

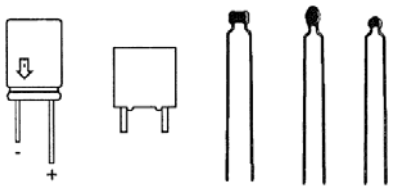
Stecken Sie die Kondensatoren in die entsprechend gekennzeich-

neten Bohrungen, biegen Sie die Drähte etwas auseinander und verlöten diese sauber mit den Leiterbahnen. Bei den Elektrolyt-Kondensatoren (Elkos) ist auf richtige Polarität zu achten (+ -).

Achtung!

Je nach Fabrikat weisen Elektrolyt-Kondensatoren verschiedene Polaritätskennzeichnungen auf. Einige Hersteller kennzeichnen „+", andere aber „-". Maßgeblich ist die Polaritätsangabe, die vom Hersteller auf den Elkos aufgedruckt ist.

C 1 = 0,01 µF = 10 nF = 10 000 pF = 103 Keramik-Kondensator
C 2 = 0,47 µF = 470 nF = 474 Keramik-Kondensator
C 3 = 0,47 µF = 470 nF = 474 Keramik-Kondensator
C 4 = 0,1 µF = 100 nF = 100 000 pF = 104 Keramik-Kondensator
C 5 = 47 µF 16 Volt Elko



1.4 IC-Fassung

Stecken Sie die Fassung für den integrierten Schaltkreis (IC) in die entsprechende Position auf der Bestückungsseite der Platine.

Achtung!

Beachten Sie die Einkerbung oder eine sonstige Kennzeichnung an einer Stirnseite der Fassung. Dies ist die Markierung (Anschluß 1) für das IC, welches später einzusetzen ist. Die Fassung muß so eingesetzt werden, daß diese Markierung mit der Markierung am Bestückungsaufdruck übereinstimmt!
Um zu verhindern, daß beim Umdrehen der Platine (zum Löten) die Fassung wieder herausfällt, werden zwei schräg gegenüberliegende Pins der Fassung umgebogen und danach alle Anschlußbeinchen verlötet.

1 x Fassung 8-pol.

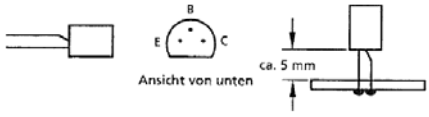


1.5 Transistoren

In diesem Arbeitsgang werden die Transistoren dem Bestückungsaufdruck entsprechend eingesetzt und auf der Leiterbahnseite verlötet.

Beachten Sie dabei die Lage: Die Gehäuse-Umrisse der Transistoren müssen mit denen des Bestückungsaufdruckes übereinstimmen. Orientieren Sie sich hierbei an der abgeflachten Seite der Transistorgehäuse. Die Anschlußbeine dürfen sich auf keinen Fall kreuzen, außerdem sollten die Bauteile mit ca. 5 mm Abstand zur Platine eingelötet werden.

Achten Sie auf kurze Lötzeit, damit die Transistoren nicht durch Überhitzung zerstört werden.
T 1 = BC 337, 338. 339 A, B oder C Kleinleistungs-Transistor
T 2 = BC 327, 328, 329 A, B oder C Kleinleistungs-Transistor



1.6 Leuchtdiode (LED)

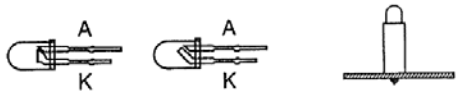
Jetzt löten Sie die 3 mm-LED polungsrichtig in die Schaltung ein. Das kürzere Anschlußbeinchen kennzeichnet die Kathode. Betrachtet man die Leuchtdiode gegen das Licht, so erkennt man die Kathode an der größeren Elektrode im Inneren der LED.

Am Bestückungsaufdruck wird die Lage der Kathode durch einen dicken Strich im Gehäuseumriß der Leuchtdiode dargestellt.

Zur Montage wird die LED in den beiliegenden LED-Abstandshalter gesteckt. Mit diesem Abstandshalter sind zwei unterschiedliche Befestigungsvarianten möglich. Wird die LED von der einen Seite in das Röhrchen gesteckt, so verschwindet von ihrem Gehäuse der untere Rand vollkommen in der Fassung und es schaut nur noch ein Teil ihres „Kopfes" aus dem Abstandshalter heraus.
Wird dagegen das Röhrchen um 180° gedreht, so sitzt die LED „oben auf". Ihr Gehäuse ragt vollständig aus dem Abstandshalter hervor.

Setzen Sie die LED so in den Abstandshalter ein, daß nur noch der „Kopf" aus dem Röhrchen herausragt.

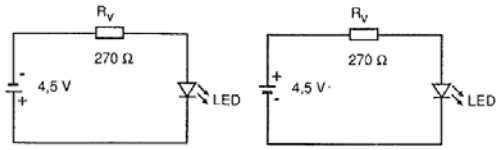
Die hier in diesem Bausatz verwendete Leuchtdiode ist eine „LOW CURRENT- LED", d. h. eine LED, die ihre volle Leuchtkraft bereits bei einer Stromaufnahme von 2 mA (grün 4 mA) erreicht.
LD 1 = rot Ø 3 mm Low Current



Fehlt eine eindeutige Kennzeichnung einer LED oder sind Sie sich mit der Polarität in Zweifel (da manche Hersteller unterschiedliche Kennzeichnungsmerkmale benutzen), so kann diese auch durch Probieren ermittelt werden. Dazu gehen Sie wie folgt vor:

Man schließt die LED über einen Widerstand von ca. 270 R (bei Low-Current-LED 4 k 7) an eine Betriebsspannung von ca. 5 V (4,5 V oder 9 V-Batterie) an.

Leuchtet dabei die LED, so ist die „Kathode" der LED richtigerweise mit Minus verbunden. Leuchtet die LED nicht, so ist diese in Sperrichtung angeschlossen (Kathode an Plus) und muß umgepolt werden.



LED wird in Sperrichtung angeschlossen und leuchtet demzufolge nicht. (Kathode an "+")

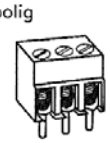
LED mit Vorwiderstand in Durchlaßrichtung angeschlossen, sie leuchtet (Kathode an "-")

1.7 Anschlußklemmen

Stecken Sie die Schraubklemmen in die entsprechende Position auf der Platine und verlöten Sie die Anschlußstifte sauber auf der Leiterbahnseite.

Bedingt durch die größere Massefläche von Leiterbahn und Anschlußklemme muß hier die Lötstelle etwas länger aufgeheizt werden, damit das Zinn gut fließt und sich eine saubere Lötstelle bildet.

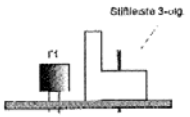
1 x Anschlußklemme 3-polig

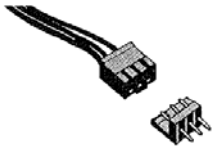


1.8 Stecker-Stiftleiste

Stecken Sie die Anschlußbeinchen der 3-pol. Stecker-Stiftleiste in die entsprechenden Bohrungen und verlöten Sie die Anschlußbeinchen mit den Leiterbahnen der Hauptplatine.

1x Stecker-Stiftleiste 3-polig





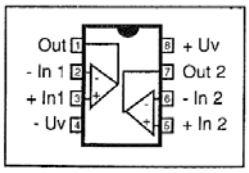
1.9 Integrierte Schaltung (IC)

Zum Schluß wird der integrierte Schaltkreis polungsrichtig in die vorgeshene Fassung gesteckt.

Achtung!

Integrierte Schaltungen sind sehr empfindlich gegen falsche Polung! Achten Sie deshalb auf die entsprechende Kennzeichnung des IC's (Kerbe oder Punkt).
Integrierte Schaltungen dürfen grundsätzlich nicht bei anliegender Betriebsspannung gewechselt oder in die Fassung gesteckt werden!

IC 1 = LM 358 OP
(Kerbe oder Punkt muß zur Leuchtdiode R 1 zeigen).



2. Baustufe II:

Montage der Bauelemente auf der Sensorplatine

2.1 SMD-Widerstände

Plazieren und verlöten Sie die SMD-Widerstände lt. Bestückungsplan auf der Platine. Der Wert der Widerstände ist in einem Zahlencode aufgedruckt.

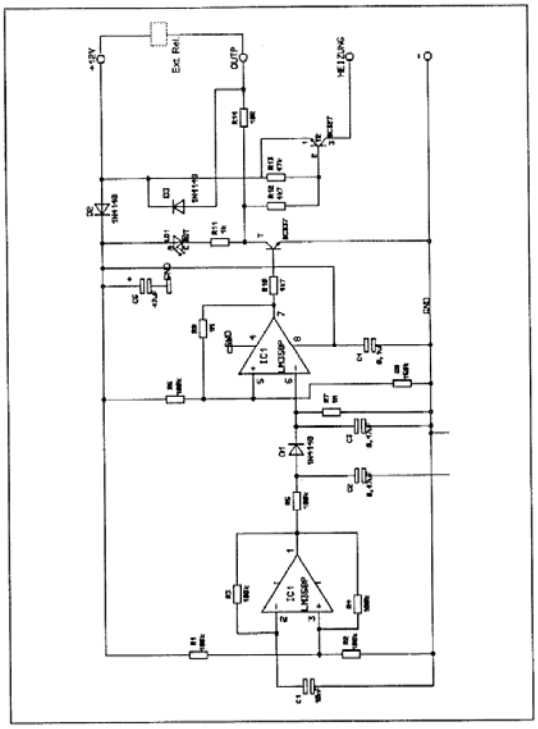
R 6 ...R 15 = 1 k 2 = 122



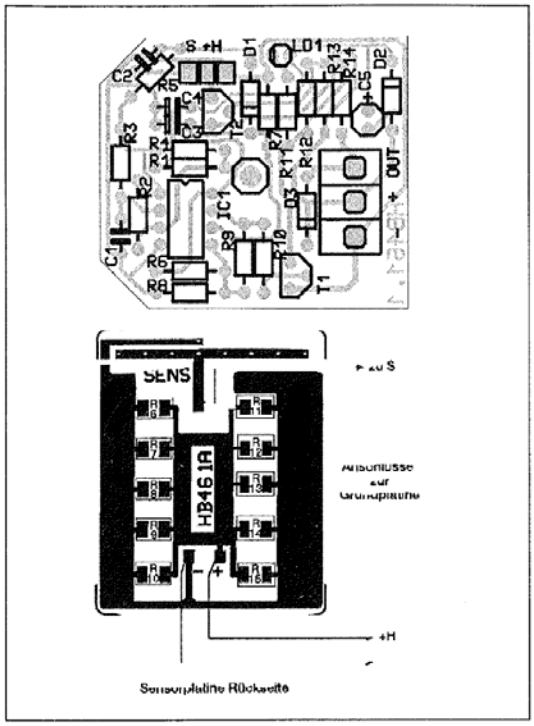
2.2 Stecker-Stiftleiste

Stecken Sie den Stecker mit den offenen Kabelenden auf die Stiftleiste auf, welche auf die Hauptplatine gelötet ist. Die Anschlußstifte der Stiftleiste sind mit „S", „+H" und „-" bezeichnet. Löten Sie das mittlere Anschlußkabel „+H" an den Anschlußpunkt „+H" auf der Sensorplatine an. Löten Sie ebenso das Anschlußkabel „S" (SENS) an die Leiterbahn „SENS" und das Anschlußkabel „-" an die „-"-Leiterbahn auf der Sensorplatine an.

Schaltplan



Bestückungspläne



3. Baustufe III:

Stückprüfung/Anschluß/Inbetriebnahme

3.1 Stückprüfung durch denjenigen, der das Gerät fertiggestellt hat!

Nach Fertigstellung des Gerätes muß als erstes eine Stückprüfung durchgeführt werden. Sinn dieser Stückprüfung ist es, Gefahren durch Materialschäden und durch unsachgemäßen Zusammenbau zu erkennen.

Sichtprüfung

Bei der Sichtprüfung darf das Gerät nicht mit seiner Stromversorgung verbunden sein.

Kontrollieren Sie nochmal, ob alle Bauteile richtig eingesetzt und gepolt sind. Sehen Sie auf der Lötseite (Leiterbahnseite) nach, ob durch Lötzinnreste Leiterbahnen überbrückt wurden, da dies zu Kurzschlüssen und zur Zerstörung von Bauteilen führen kann.

Ferner ist zu kontrollieren, ob abgeschnittene Drahtenden auf oder unter der Platine liegen, da dies ebenfalls zu Kurzschlüssen führen kann.

Etwaige Mängel sind zu beseitigen!

Anschluß/Inbetriebnahme

3.2 Nachdem die Stückprüfung durchgeführt wurde, kann ein erster Funktionstest durchgeführt werden.

Beachten Sie, daß dieser Bausatz nur mit gesiebter Gleichspannung aus einem Netzgerät oder mit einer Batterie/Akku versorgt werden darf, das bzw. die auch den nötigen Strom liefern kann. Autoladegeräte oder Spielzeugeisenbahntrafos sind als Spannungsquelle nicht geeignet und

führen zur Beschädigung von Bauteilen bzw. zur Nichtfunktion der Baugruppe.

Lebensgefahr!

Verwenden Sie ein Netzgerät als Spannungsquelle, so muß dies unbedingt den VDE-Vorschriften entsprechen!

3.3 Schließen Sie an den Anschlußklemmen „+" und „-" die Betriebsspannung (Gleichspannung), die im Bereich zwischen 11 und 15 V liegen kann, polungsrichtig an.

3.4 Berühren Sie die Sensorfläche mit einem leicht angefeuchteten Finger.

- Die rote Leuchtdiode LD 1 muß nun aufleuchten
- Die Sensorfläche muß sich langsam erwärmen (mit der LED wird gleichzeitig die Heizung für die Sensorfläche eingeschaltet).

3.5 Ist die Senorfläche abgetrocknet, so muß die rote LED erlöschen und sich die Heizung der Sensorfläche wieder abschalten.

3.6 An die Anschlußklemmen „+ 12 Volt" und „OUT" kann ein Kleinrelais mit einer max. Stromaufnahme von 100 mA angeschlossen werden. Leuchtet LD 1 auf, so muß auch das angeschlossene externe Relais anziehen. Eine Freilaufdiode, die normalerweise für solch eine induktive Last notwendig ist, braucht extern nicht beschaltet werden - diese Schutzdiode befindet sich bereits auf der Platine.

3.7 Ist bis hierher alles in Ordnung, so überspringen Sie die nachfolgende Fehler-Checkliste.

3.8 Sollte die rote LED wider Erwarten nicht oder ständig leuchten, oder eine andere Fehlfunktion zu erkennen sein, so

schalten Sie sofort die Betriebsspannung ab und prüfen die komplette Platine noch einmal nach folgender Checkliste.

Checkliste zur Fehlersuche

Haken Sie jeden Prüfungsschritt ab!

- ❑ Bevor Sie mit der Überprüfung der Schaltung beginnen, trennen Sie diese unbedingt von der Betriebsspannung!
- ❑ Ist die Betriebsspannung richtig gepolt?
- ❑ Ist die Betriebsspannung an den richtigen Anschlußklemmen angeschlossen?
- ❑ Liegt die Betriebsspannung bei eingeschaltetem Gerät noch im Bereich von 11 - 15 Volt?
- ❑ Betriebsspannung wieder ausschalten.
- ❑ Sind die Widerstände wertmäßig richtig eingelötet?
  Überprüfen Sie die Werte noch einmal nach **1.1** der Bauanleitung.
- ❑ Sind die Dioden richtig gepolt eingelötet? Stimmt der auf den Dioden angebrachte Kathodenring mit dem Bestückungsaufdruck auf der Platine überein?
  Der Kathodenring von D 1 muß zum Befestigungsloch zeigen.
  Der Kathodenring von D 2 muß von den Anschlußklemmen wegzeigen.
  Der Kathodenring von D 3 muß zu T1 zeigen
- ❑ Ist die LED richtig gepolt eingelötet?
  Betrachtet man die Leuchtdiode gegen das Licht, so erkennt man die Kathode an der größeren Elektrode im Inneren der

LED. Am Bestückungsaufdruck wird die Lage der Kathode durch einen Strich am Gehäuseumriß der Leuchtdiode dargestellt.
Die Kathode der LED LD 1 muß zur Beschriftung „D1" zeigen.

- ❑ Sind die Transistoren T 1 u. T 2 richtig herum eingelötet? Überkreuzen sich ihre Anschlußbeinchen? Stimmt der Bestückungsaufdruck mit den Umrissen der Transistoren überein?
- ❑ Sind jeweils die richtigen Transistor-Typen eingelötet?
  Überprüfen Sie die Typenbezeichnung noch einmal mit der Stückliste (zwei unterschiedliche Typen).
- ❑ Sind die Elektrolyt-Kondensatoren richtig gepolt?
  Vergleichen Sie die auf den Elkos aufgedruckte Polaritätsangabe noch einmal mit dem auf der Platine aufgebrachten Bestückungsaufdruck bzw. mit dem Bestückungsplan in der Bauanleitung. Beachten Sie, daß je nach Fabrikat der Elkos „+" oder „-" auf den Bauteilen gekennzeichnet sein kann!
- ❑ Ist der integrierte Schaltkreis polungsrichtig in der Fassung?
  Kerbe oder Punkt von IC 1 muß zu R1 zeigen.
- ❑ Sind alle IC-Beinchen wirklich in der Fassung? Es passiert sehr leicht, daß sich eines beim Einstecken umbiegt oder an der Fassung vorbei mogelt.
- ❑ Befindet sich eine Lötbrücke oder ein Kurzschluß auf der Lötseite?
  Vergleichen Sie Leiterbahnverbindungen, die eventuell wie eine ungewollte Lötbrücke aussehen mit dem Leiterbahnbild (Raster) des Bestückungsaufdrucks und dem Schaltplan in der Anleitung, bevor Sie eine Leiterbahnverbindung (vermeintliche Lötbrücke) unterbrechen!
- ❑ Prüfen Sie auch, ob jeder Lötpunkt gelötet ist; oft kommt es

vor, daß Lötstellen beim Löten übersehen werden.
Um Leiterbahnverbindungen oder -unterbrechungen leichter feststellen zu können, halten Sie die gelötete Printplatte gegen das Licht und suchen von der Lötseite her nach diesen unangenehmen Begleiterscheinungen.

- ❑ Ist eine kalte Lötstelle vorhanden?
  Prüfen Sie bitte jede Lötstelle gründlich!
  Prüfen Sie mit einer Pinzette, ob Bauteile wackeln!
  Kommt Ihnen eine Lötstelle verdächtig vor, dann löten Sie sie sicherheitshalber noch einmal nach!

- ❑ Prüfen Sie auch, ob jeder Lötpunkt gelötet ist; oft kommt es vor, daß Lötstellen beim Löten übersehen werden.

- ❑ Denken Sie auch daran, daß eine mit Lötwasser, Lötfett oder ähnlichen Flußmitteln oder mit ungeeignetem Lötzinn gelötete Platine nicht funktionieren kann. Diese Mittel sind leitend und verursachen dadurch Kriechströme und Kurzschlüsse.

  Desweiteren erlischt bei Bausätzen, die mit säurehaltigem Lötzinn, mit Lötfett oder ähnlichen Flußmitteln gelötet wurden die Garantie, bzw. diese Bausätze werden von uns nicht repariert oder ersetzt.

3.9 Sind diese Punkte überprüft und eventuelle Fehler korrigiert worden, so ist nach Baustufe **3.1** erneut die Stückprüfung durchzuführen. Erst danach darf die Baugruppe wieder in Betrieb genommen werden! Ist durch einen eventuell vorhandenen Fehler kein Bauteil in Mitleidenschaft gezogen worden, muß die Schaltung nun funktionieren.

Die vorliegende Schaltung kann nun nach erfolgtem Funktionstest in ein entsprechendes Gehäuse eingebaut, und unter Einhatung der VDE-Bestimmungen für den vorgesehenen Zweck in Betrieb genommen werden.

32

Tips für den Gehäuseeinbau

- Kleben Sie die Sensorplatine mit der vergoldeten Sensorfläche nach außen in den bereits ausgefräßten Gehäusedeckel ein. Verwenden Sie hierzu Silikonmasse, damit das Gehäuse wasserdicht wird.

- Verwenden Sie zur Kabeldurchführung eine wasserdichte PG-Kabelverschraubung!

- Das Gerät sollte so im Freien montiert werden, daß die Sensorfläche nach oben zeigt und um ca. 45° geneigt ist. So kann der Regen gut abfließen und sich keine „Pfützen" auf der Sensorfläche bilden.

- Achten Sie auch darauf, daß auch die Durchkontaktierungen auf der Platine wasserdicht verschlossen sind (mit Lack abgedeckt oder mit Silikon verschlossen).

Störung

Ist anzunehmen, daß ein gefahrloser Betrieb nicht mehr möglich ist, so ist das Gerät außer Betrieb zu setzen und gegen unbeabsichtigten Betrieb zu sichern.

Das trifft zu:
- wenn das Gerät sichtbare Beschädigungen aufweist
- wenn das Gerät nicht mehr funktionsfähig ist
- wenn Teile des Gerätes lose oder locker sind
- wenn die Verbindungsleitungen sichtbare Schäden aufweisen.

Falls das Gerät repariert werden muß, dürfen nur Original-Ersatzteile verwendet werden! Die Verwendung abweichender Ersatzteile kann zu ernsthaften Sach- und Personenschäden führen!

33

Eine Reparatur des Gerätes darf nur vom Fachmann durchgeführt werden!

Garantie

Auf dieses Gerät gewähren wir 1 Jahr Garantie. Die Garantie umfaßt die kostenlose Behebung der Mängel, die nachweisbar auf die Verwendung nicht einwandfreien Materials oder Fabrikationsfehler zurückzuführen sind.

Da wir keinen Einfluß auf den richtigen und sachgemäßen Aufbau haben, können wir aus verständlichen Gründen bei Bausätzen nur die Gewähr der Vollständigkeit und einwandfreien Beschaffenheit der Bauteile übernehmen.

Garantiert wird eine den Kennwerten entsprechende Funktion der Bauelemente im uneingebautem Zustand und die Einhaltung der technischen Daten der Schaltung bei entsprechend der Lötvorschrift, fachgerechter Verarbeitung und vorgeschriebener Inbetriebnahme und Betriebsweise.

Weitergehende Ansprüche sind ausgeschlossen.

Wir übernehmen weder eine Gewähr noch irgendwelche Haftung für Schäden oder Folgeschäden im Zusammenhang mit diesem Produkt. Wir behalten uns eine Reparatur, Nachbesserung, Ersatzteillieferung oder Rückerstattung des Kaufpreises vor.

Bei folgenden Kriterien erfolgt keine Reparatur bzw. es erlischt der Garantieanspruch:

- wenn zum Löten säurehaltiges Lötzinn, Lötfett oder säurehaltiges Flußmittel u. ä. verwendet wurde,
- wenn der Bausatz unsachgemäß gelötet und aufgebaut wurde.

34

Das gleiche gilt auch

- bei Veränderung und Reparaturversuchen am Gerät
- bei eigenmächtiger Abänderung der Schaltung
- bei der Konstruktion nicht vorgesehene, unsachgemäße Auslagerung von Bauteilen, Freiverdrahtung von Bauteilen wie Schalter, Potis, Buchsen usw.
- Verwendung anderer, nicht original zum Bausatz gehörender Bauteile
- bei Zerstörung von Leiterbahnen oder Lötaugen
- bei falscher Bestückung und den sich daraus ergebenden Folgeschäden
- Überlastung der Baugruppe
- bei Schäden durch Eingriffe fremder Personen
- bei Schäden durch Nichtbeachtung der Bedienungsanleitung und des Anschlußplanes
- bei Anschluß an eine falsche Spannung oder Stromart
- bei Falschpolung der Baugruppe
- bei Fehlbedienung oder Schäden durch fahrlässige Behandlung oder Mißbrauch
- bei Defekten, die durch überbrückte Sicherungen oder durch Einsatz falscher Sicherungen entstehen

In all diesen Fällen erfolgt die Rücksendung des Bausatzes zu Ihren Lasten.

35

Achtung!!

Die Beschaltung des Regensensors hat sich gegenüber der originalen Bedienungsanleitung geändert. Wie dem folgenden Schaltbild zu entnehmen ist wurde auch die PIN-Belegung des 3-poligen Schraubklemmenblocks geändert.

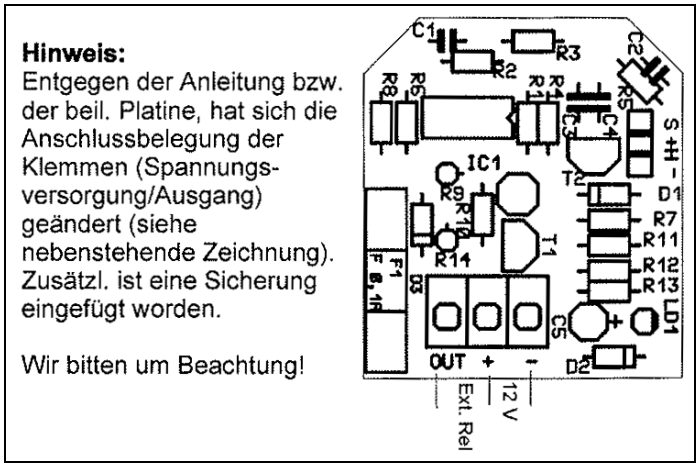


Abbildung 99: Korrekte Verschaltung Regensensor

Die Heizung des Regensensors wird über eine separate 3.-Leitung betrieben. Die Beschaltung in Verbindung mit der Gesamtsteuerung sieht vor, die Heizung durch Jumper separat zu steuern. Hierzu wird die Schaltspannung mit zwei Leitungen vom Regensensor zurück zum ConvCtrl geführt und dort über einen Jumper konfigurierbar gemacht.

Der Grundgedanke hierzu ist folgender: Ist der Regensensor der Nässe ausgesetzt und zeigt dieser Regen an so ist es auch nicht notwendig, dass die Beschattung zugefahren wird. Es kann jedoch notwendig sein, mit Hinblick auf Wintermonate den Regensensor zu beheizen um Eis und Schnee abtauen zu lassen.

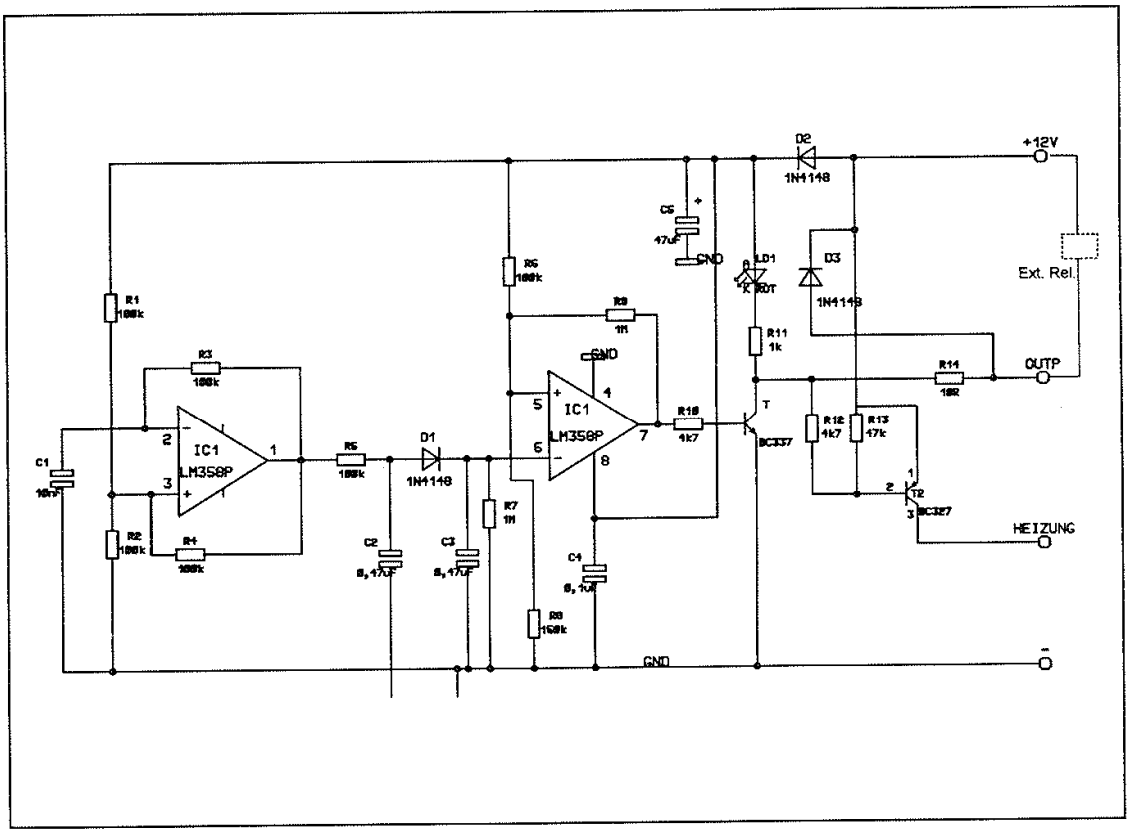
Eventuell wird speziell für die Heizung ein kleiner Schalter im Gehäuse vorgesehen welcher es ermöglicht die Konfiguration nach Wahl einzustellen.

8.14.2 Ankopplung Regensensor

Am Anschlussprinzip des Regensensors hat sich jedoch nichts geändert. Der 3-polige Schraubklemmenblock beinhaltet die Anschlusse.

- \- = GND (Spannungsversorgung)
- \+ = +12V (Spannungsversorgung)
- OUT = Geschalteter Ausgang

Wie dem Schaltplan des Regensensors zu entnehmen ist wird der Out Ausgang mit einem Transistor gegen Masse geschaltet so dass zwischen +Uv und Out eine Spannung von +12V anliegt.

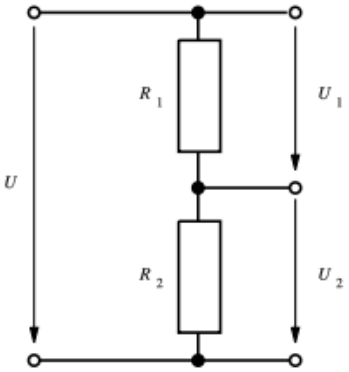


Schaltbild 34: Schaltbild Regensensor

Die Beschaltung sieht vor, die 12V mittels Spannungsteiler im Verhältnis 1:2,4 zu teilen und über eine NAND-Schmitt-Trigger-Kombination auf +5V Eingangspegel für den ATmega zu normieren.

Der Spannungsteiler wird hochohmig ausgelegt um den Stromverbrauch gering zu halten.

<u>Dimensionierung Spannungsteiler:</u>



$$U_2 = U \bullet \frac{R_2}{R_1 + R_2}$$

$$\frac{U_2}{U} = \frac{R_2}{R_1 + R_2}$$

$$R_1 = R_2 \rightarrow U_2 = \frac{U}{2}$$

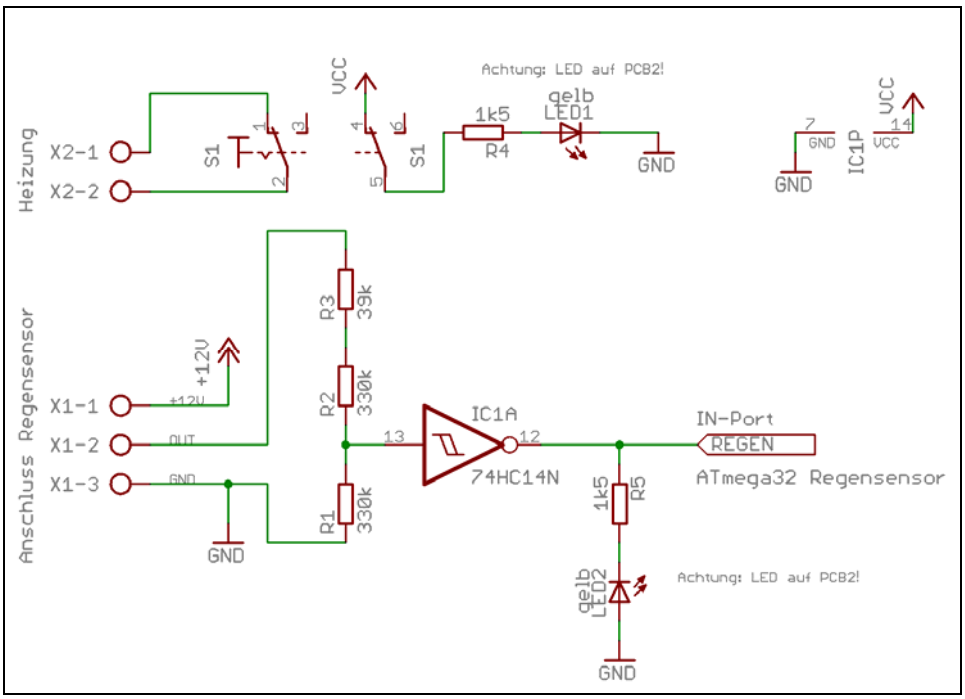
Für die Spannung U wird bei einem idealen Spannungsteiler (unbelastet) die Spannung mit 10,6V angenommen. Als Berechnungsbasis wurde der Widerstand R2 mit 330kΩ festgelegt.

$$R_1 = \left(\frac{U \bullet R_2}{U_2}\right) - R_2 = \left(\frac{10{,}6V \bullet 330k}{5V}\right) - 330k = 369{,}6k - 1k - 10R = 368k = 330k + 39k$$

Es werden folgende Widerstandswerte gewählt: R_1 = 330k + 39kΩ
R_2 = 330kΩ

Aus obiger Rechnung ergibt sich für eine unbelasteten Spannungsversorgung von ca. 12V ein Wert für U_2 von ca. 4,3 - 4,8V

Beschaltung:



Schaltbild 35: Schaltbild Beschaltung Regensensor

Bauteile:

Stückliste: Regensensor			
Widerstände		*Halbleiter*	
R1, R2	330k Ω	IC1A	Schmitt-Trigger-NAND 74HC14N
R3	39k Ω	LED1, LED2	low current LED 3mm, gelb
R4, R5	1k5 Ω	*Sonstiges*	
		X1	Schraubklemme 3-polig, Raster 5mm
		X2	Schraubklemme 2-polig, Raster 5mm
		S1	Mikroschalter 2xUm Kippschalter, 2-polig, 3A-250VAC, Ein-Ein (*MS 500F*)
		ca. 10 Meter 10-adrige Steuerleitung	
		Kabelverschraubung	

Tabelle 108: Stückliste Beschaltung Regensensor

Themenbezogene LED's:

LED / Funktion / Beschreibung	*Abkürzung*	*Farbe*	
Regen detektiert	LED_RAIN_DETECT	gelb	*[LED29]*
Heizung Regensensor aktiv	LED_RAIN_HEATER	gelb	*[LED20]*

Tabelle 109: Themenbezogene LED's Regensensor

Ressourcenzuordnung zum ATmega128:

Nummer	*Schaltbild*	*Ressource ATmega128*

1	RAIN	[RAIN]	PortG.3	[PG3]	(GPIO)

Tabelle 110: Ressourcenzuordnung ATmega128 für Regensensor

Beschaltung:

Zur Beschaltung und Ankopplung an die Wintergartensteuerung wird eine min. 5-adrige flexible Leitung benötigt. Über diese Leitung wird die Versorgungsspannung als auch die Schaltleitung für Regenerkennung und Sensorheizung geführt.

Bei der Realisierung wird eine 10-adrige Steuerleitung eingesetzt. Die Steuerleitung wird über spezielle Kabelverschraubungen in das Gehäuse geführt.



Abbildung 100: Kippschalter für Regensensorheizung!



Abbildung 101: EX-Kabelverschraubung K-M schwarz

Conrad Electronic
Artikel-Nr.: 525978 - 62

Pinbelegung

Die folgende Tabelle zeigt die Pinbelegung der 10-adrigen Steuerleitung für den Anschluss des Regensensors:

Pin	*Farbe*	*Funktion*
1	rot	+ 12V
2	schwaz	GND
3	braun	Regensignal
4	gelb	Heizung S1
5	blau	Heizung S2
6	weiß	nc
7	rosa	nc
8	grau	nc
9	grün	nc
10	lila	nc

Tabelle 111: Pinbelegung Steuerleitung Regensensor

8.14.3 BASCOM Treiber und Beispielcode für den Regensensor

```
'#####################################################################################
' REGENSENSOR.BAS:                                          Stand 16.02.2007
' ----------------                                          (C) Markus Fulde
'
' Testprogramm zur Inbetriebnahme der Regensensor-Beschaltung des Regensensor-
' Moduls von Corad Electronic Best.-Nr. 11 52 40.
'
' Die Inbetriebnahme erfolgt mit dem Eva-Board STK500 + STK501
' Die notwendigen Routinen für DCF77 und DS1307 sind als solch'
'
' Der wesentliche Programmteil besteht aus einer Funktion welche als Rückgabe-
' wert den Zustand des Regensensor (Regen = JA/NEIN) zurück liefert.
'
```

```
'################################################################################

'--------------------------------------------------------------------------------
' Compilerinstruktionen und Compilerdirektiven
'--------------------------------------------------------------------------------
$regfile = "m128def.dat"
$crystal = 16000000
$hwstack = 32
$swstack = 32
$framesize = 40

$baud = 38400

'--------------------------------------------------------------------------------
' Definition von Ressourcen
'--------------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========
Rain_sensor Alias Ping.3                                    ' Anschluss an PinG.3

' Achtung: Auf STK501 ist an PinG.3 via Default der Oszillator TOSC ange-
' schlossen => Schiebeschalter auf Stellung PIN


'--------------------------------------------------------------------------------
' Definition von Variablen, Konstaten und Datentypen
'--------------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========
Const Rain_testmodus = 1

Const Rain_no = 0                                           ' Boolsche Variable für NO
Const Rain_yes = 1                                          ' Boolsche Variable für YES

'========== Für Testumgebung ==========
Const Main_testmodus = 1

Dim Temp_byte_1 As Byte


'--------------------------------------------------------------------------------
' Prototyping
'--------------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========
Declare Function Rain_measurement() As Byte


'--------------------------------------------------------------------------------
' Konfiguration und Basiseinstellungen (Projekt und Testumgebung)
'--------------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========
Config Rain_sensor = Input

'========== Für Testumgebung ==========

#if Main_testmodus
   Print "** Enter MainLoop **"
   Print
#endif


'################################################################################
' Hauptprogramm
'################################################################################
'Hier ist die Programmhauptschleife
Do
   ' Funktion zur Abfrage des Regensensors aufrufen
   Temp_byte_1 = Rain_measurement();

   ' und auswerten
   If Temp_byte_1 = Rain_yes Then
      ' Rain_yes
      Print "Rain detected -> Regenschirme raus"
   Else
      ' Rain_no
      Print "No Rain detected -> Grillzeug raus"
   End If
```

```
   ' Leerzeile
   Print

   ' 1 Sekunde warten und nochmal
   Wait 1

Loop
'## End Hauptprogramm ###############################################################


'*****************************************************************************
' Interruptroutinen
'*****************************************************************************

' gibt's keine


'*****************************************************************************
' Subroutinen
'*****************************************************************************

' gibts keine


'*****************************************************************************
' Funktionen
'*****************************************************************************

'-----------------------------------------------------------------------------
' RAIN - Function: Rain_measurement
' Funktion liest den Zustand des PINs zurück an dem der Regensensor ange-
' schlossen ist und liefert den Wert als Rückgabewert
'
' Parameter:    keiner
' Rückgabewert: Bytewert &H00 = kein Regen
'                        &H01 = Regen vom Sensor erkannt
'-----------------------------------------------------------------------------
Function Rain_measurement() As Byte

   ' ----- lokale Variablen -----
   Local Temp_value As Byte

   ' ----- Programmcode -----
   #if Rain_testmodus
      Print "Enter Rain_measurement"
      ' Direkte Auswertung des PINs für Trace
      If Rain_sensor = 0 Then
         Print "PinG.3 - NO"
      Else
         Print "PinG.3 - YES"
      End If
   #endif

   ' Status Regensensor von Pin einlesen
   Temp_value = Rain_sensor

   ' Traceausgabe
   #if Rain_testmodus
      Print "Zustand PIN Regensensor = " ; Temp_value
   #endif

   ' temporäre Variable in Rückgabewert übergeben
   Rain_measurement = Temp_value

End Function
'-- End Rain_measurement ---------------------------------------------------

'*****************************************************************************
' GOSubroutinen
'*****************************************************************************

' gibt's nix

End                                                 'end program


'-----------------------------------------------------------------------------
' Definition von globalen Konstantenfeldern
'-----------------------------------------------------------------------------
```

```
' gibt's nix

'###################################### END ###################################
```

Software 9: Code zum Betrieb des Regensensors

8.14.4 Prototyp zum Regensensor

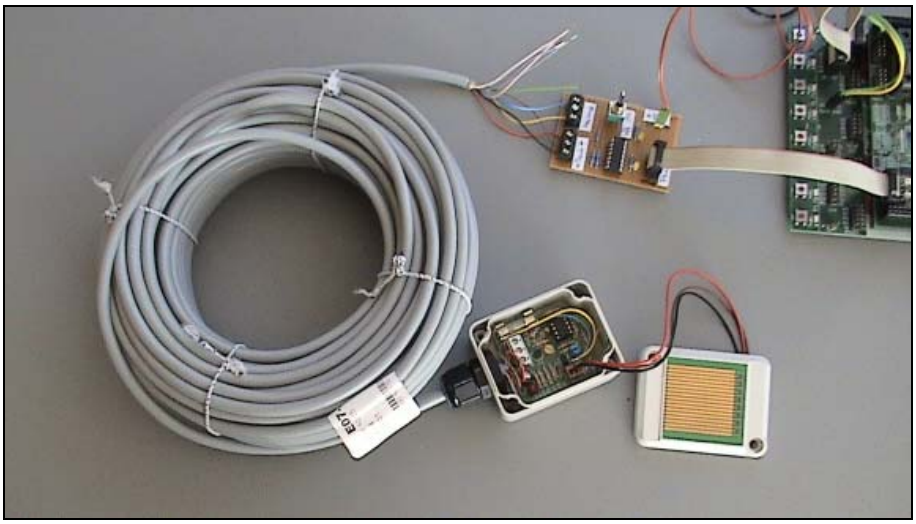


Abbildung 102: Prototypaufbau Regensensor mit Beschaltung

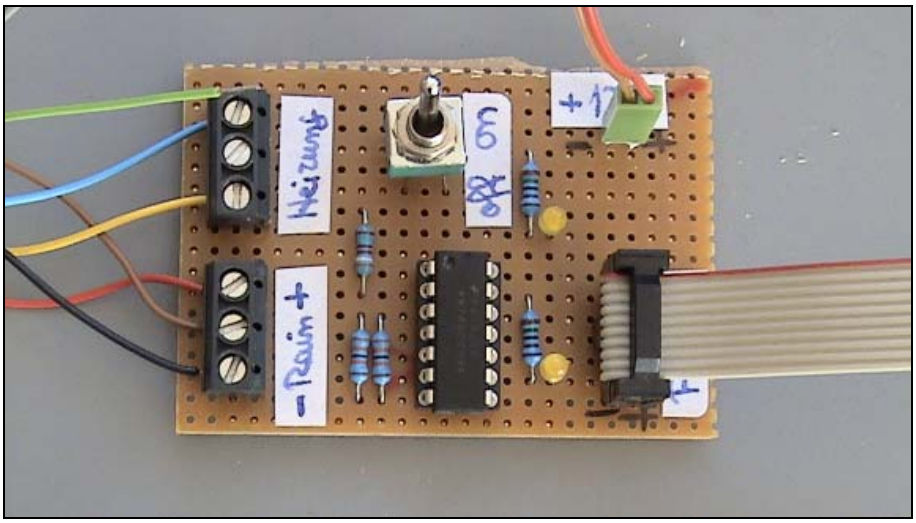


Abbildung 103: Prototypaufbau Beschaltung für Regensensor im Detail

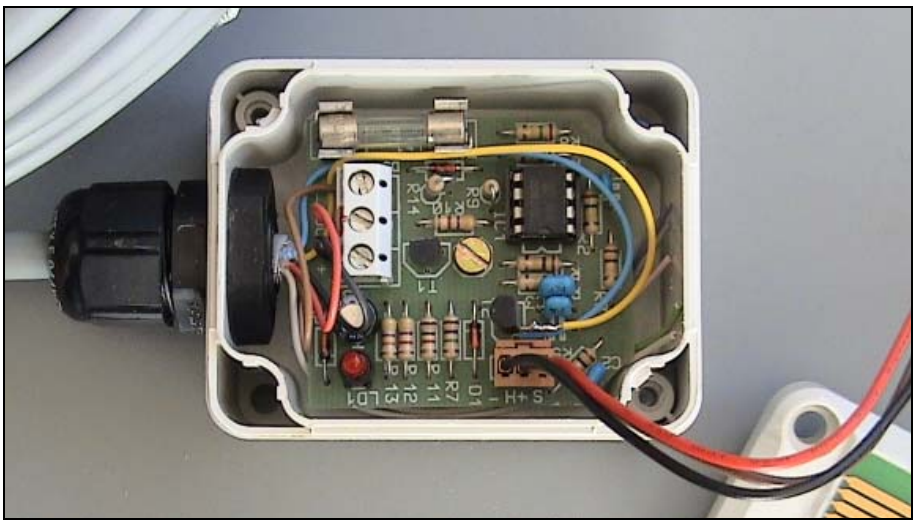


Abbildung 104: Innenleben Regensensor-Elektronik

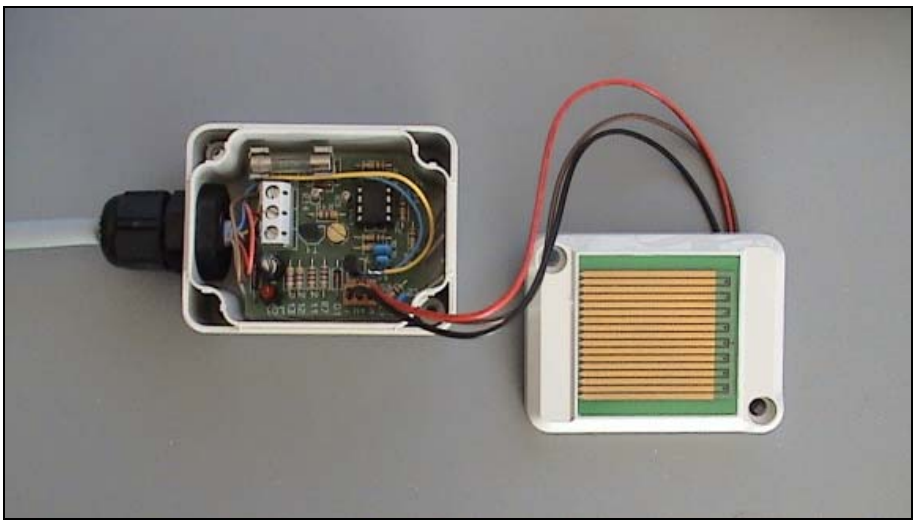


Abbildung 105: Gesamtkomponente Regensensor inkl. Sensorfläche

8.15 Beschaltung DCF77-Modul

Für den Empfang des DCF77-Signals aus Frankfurt zur Synchronisation der internen Uhr wird das DCF77-Modul von Conrad Electronic mit der Artikelnummer 641138 – LN eingesetzt.

Modulbeschreibung:

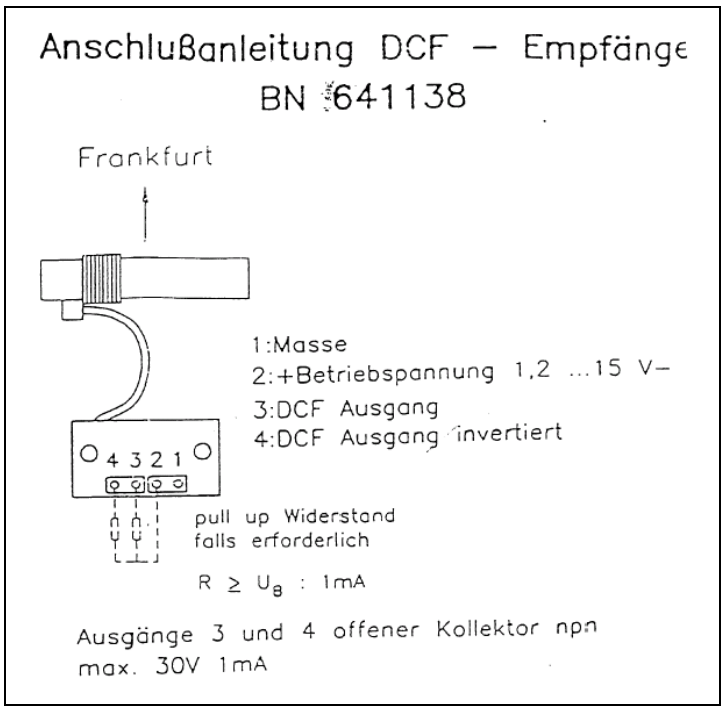


Abbildung 106: Beschaltung Conrad DCF77-Modul

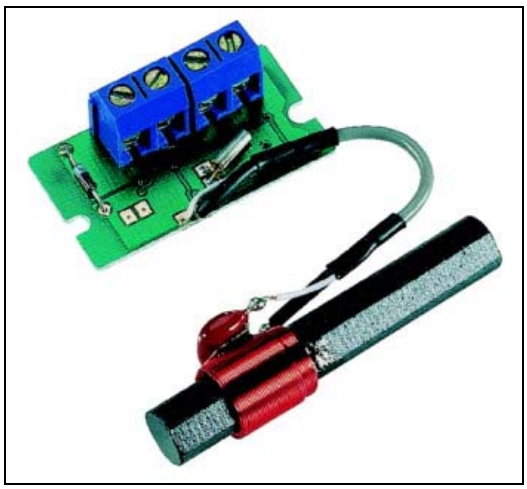
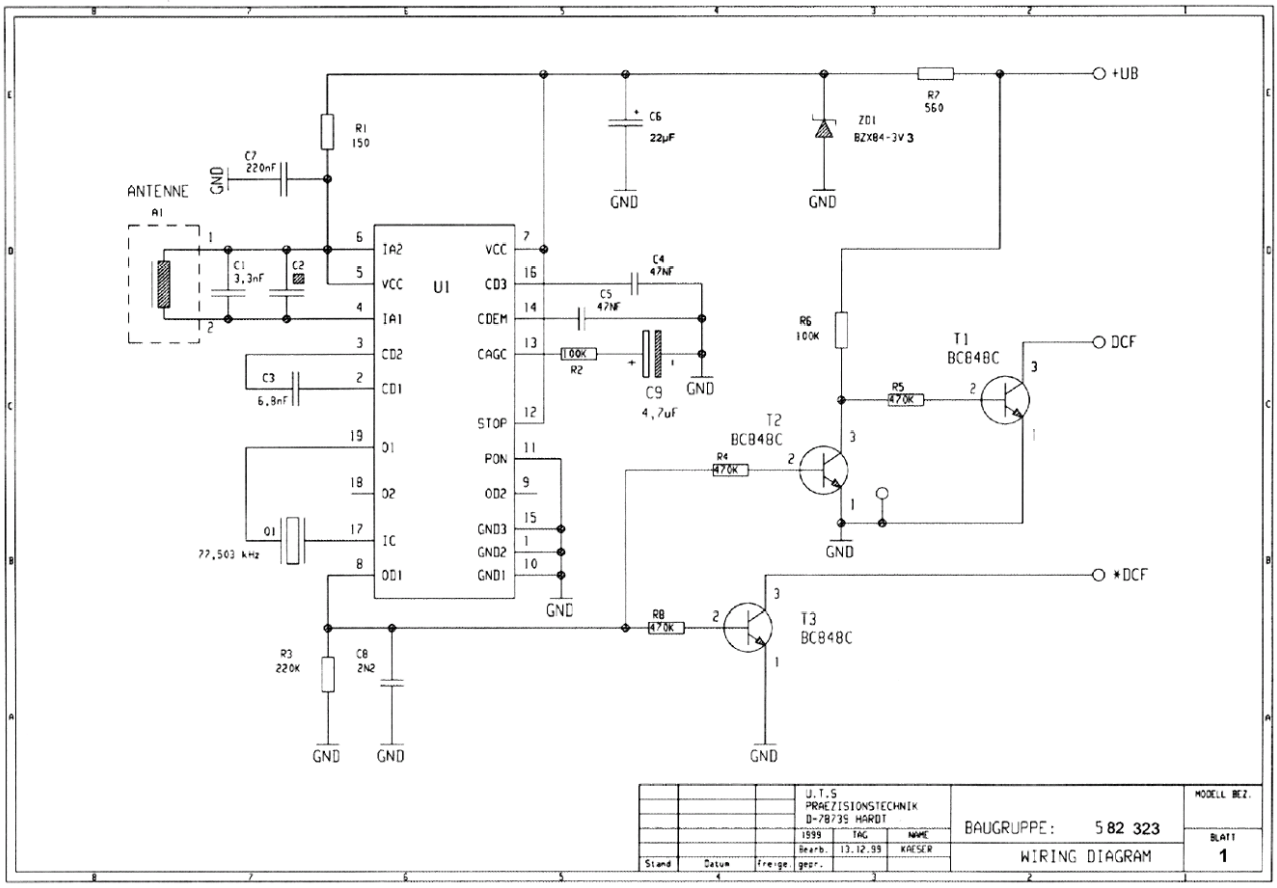
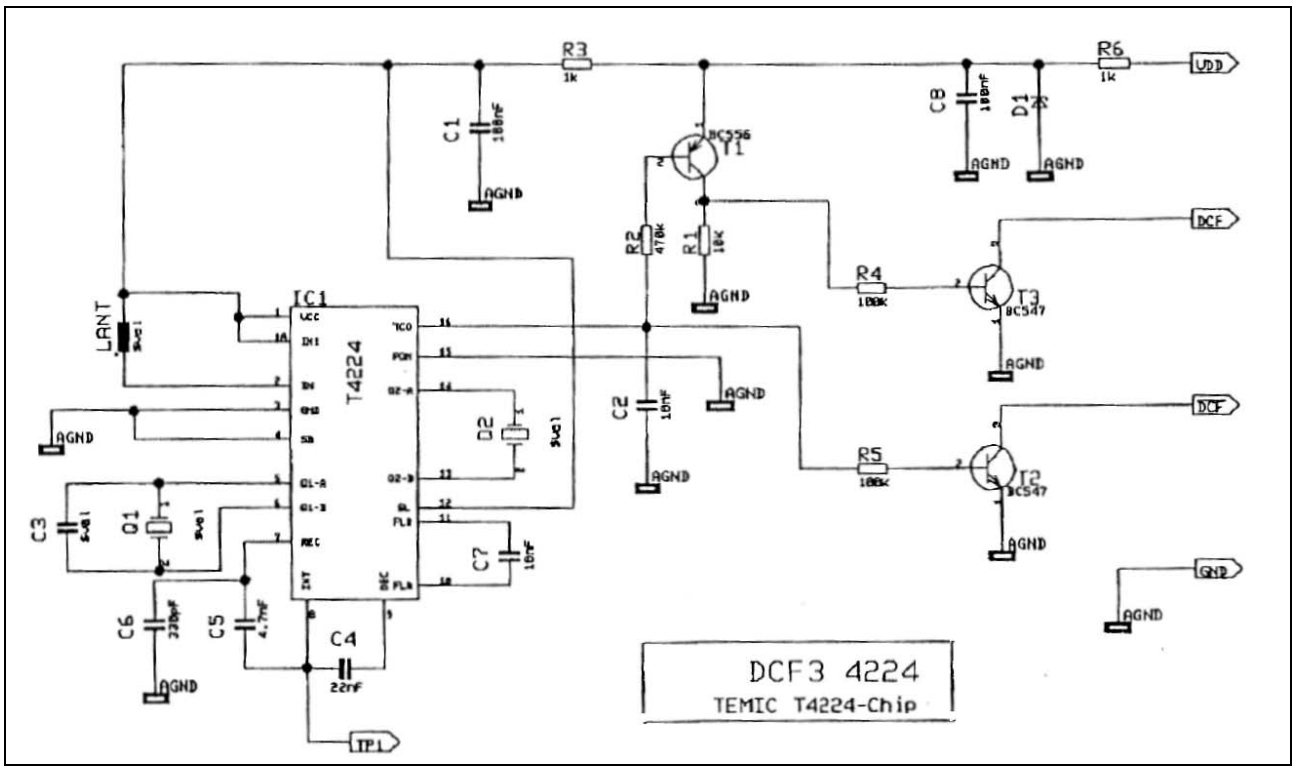


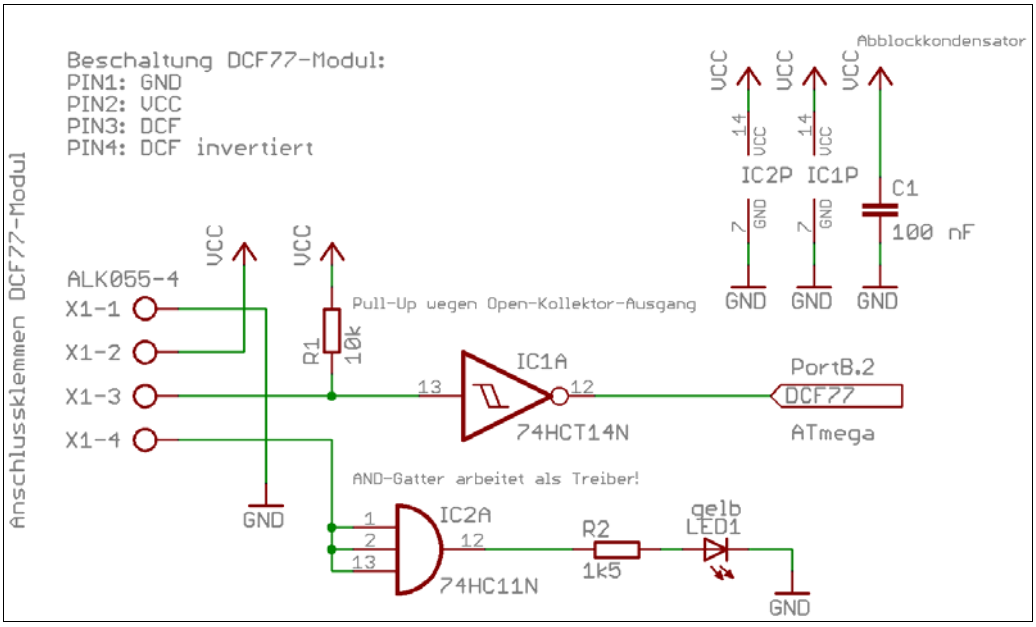
Abbildung 107: Conrad DCF77-Modul





Schaltbild 36: Schaltplan DCF77-Modul von Conrad

Aufschaltung an ATmega:



Schaltbild 37: Aufschaltung DCF77 an ATmega

Bauteile:

Stückliste: DCF77-Modul			
Widerstände		*Halbleiter*	
R1	Metallschicht-Widerstand 10k Ω	IC1A	Schmitt-Trigger-NAND 74HCT14N
R2	Metallschicht-Widerstand 1k5 Ω	IC2A	NAND 74HC11N
		LED1	Low corrent LED 3mm gelb

Kondensatoren		*Sonstiges*	
C1	Abblockkondensator 100 nF	X1	Schraubklemme 4-polig, Raster 5mm
			DCF77-Modul

Tabelle 112: Stückliste Beschaltung DCF77-Modul

Themenbezogene LED's:

LED / Funktion / Beschreibung	*Abkürzung*	*Farbe*	
DCF77 Impuls	LED_DCF77_TICK	gelb	*[LED30]*

Tabelle 113: Themenbezogene LED's DCF77

Ressourcenzuordnung zum ATmega128:

Nummer	*Schaltbild*	*Ressource ATmega128*		
1	DCF77 [DCF77]	PortB.2	[DCF77]	(GPIO)

Tabelle 114: Ressourcenzuordnung ATmega128 für DTC77

Bemerkung zur Umsetzung:

Aus Empfangs- und Befestigungs-Gründen wird der DCF77-Empfänger auf dem Konsolen-Board PCB1 angebracht und befestigt. Das durch Schmitt-trigger aufbereitete Signal wird über den Pfostenstecker auf Board PCB2 geführt.

8.15.1 BASCOM Treiber und Beispielcode für das DCF77-Modul

```
'#################################################################################
' DCF77_und_DS1307.BAS                                        Stand 13.02.2007
' --------------------                                        (C) Markus Fulde
'
' Testprogramm zur Inbetriebnahme des DCF77 Funkuhrmoduls von Conrad in Ver-
' bindung mit der BASCOM DCF77-Lösung.
' Es wird ein RTC (DS1307) über TWI-Interface angebunden welcher mit der DCF77
' synchronisiert wird und die Uhrzeit währedn der spannungslosen Zeit puffert.
'
' Die Inbetriebnahme erfolgt mit dem Eva-Board STK500 + STK501
' Die notwendigen Routinen für DCF77 und DS1307 sind als solche gekennzeichnet.
'
' Zur Erläuterung:
' Die DCF77 Anbindung wird nur zur Synchronisation der SoftClock verwendet. Das
' Rechnen und Arbeiten mit Zeit und Datumsinformationen geschieht über die
' Variablen der SoftClock.
' Die RTC dient dazu, Zeitinformationen bereitzustellen wenn kein DTF Empfang
' vorhanden ist oder wenn bei Systemstart noch nicht gesynct wurde. Die RTC
' wird nach erfolgreichem DCF77 Sync gesetzt und bei Systemstart ausgelesen um
' die SoftClock vorzuinitialisieren. Man kann sich überlegen - falls die
' SoftClock zu ungenau arbeitet jede Stunde die SoftClock mit der RTC zu
' synchronisieren.
'#################################################################################


'--------------------------------------------------------------------------------
' Compilerinstruktionen und Compilerdirektiven
'--------------------------------------------------------------------------------
$regfile = "m128def.dat"                                    ' ATmega 128
$crystal = 16000000                                         ' 16 MHz
$hwstack = 32
$swstack = 32
$framesize = 40

$lib "i2c_twi.lbx"                                          ' we do not use software emulated I2C
but the TWI


'--------------------------------------------------------------------------------
```

```
' Definition von Ressourcen
'--------------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========
Sda_port Alias Portd.1
Scl_port Alias Portd.0

'========== Für Testumgebung  ==========

' ----- Alias Namen für Tastaturabfragen -----
Taste_1 Alias Pina.0
Taste_2 Alias Pina.1
Taste_3 Alias Pina.2
Taste_4 Alias Pina.3
Taste_5 Alias Pina.4
Taste_6 Alias Pina.5
Taste_7 Alias Pina.6
Taste_8 Alias Pina.7


'--------------------------------------------------------------------------------
' Definition von Variablen, Konstaten und Datentypen
'--------------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========

' ----- Flags für Testmenüs und Trace-Ausgaben -----
Const Lcd_testmodus = 1                                 ' Testmodus für LCD
Const Dcf77_testmodus = 1                               ' Testmodus für DCF77
Const Rtc_testmodus = 1                                 ' Testmodus für RTC DS1307

' ----- DCF77 -----
Dim Dcf77_hour As Byte                                  ' Globale Stundenvariable
Dim Dcf77_min As Byte                                   ' Globale Minutenvariable
Dim Dcf77_sec As Byte                                   ' Globale Sekundenvariable

Dim Dcf77_weekday As Byte                               ' Globaler Wochentag

Dim Dcf77_day As Byte                                   ' Globale Tagvariable
Dim Dcf77_month As Byte                                 ' Globale Monatsvariable
Dim Dcf77_year As Byte                                  ' Globale Jahresvariable

' ----- RTC DS1307 -----
' RTC:  Adressen für das exerne EEPROM
Const Rtc_address_write = &B11010000                    ' Adresse des DS1307 = &HD0 (schreiben)
Const Rtc_address_read = &B11010001                     ' Adresse des DS1307 = &HD1 (lesen)

Const Rtc_address_seconds = &H00
Const Rtc_address_minutes = &H01
Const Rtc_address_hours = &H02
Const Rtc_address_day = &H03
Const Rtc_address_date = &H04
Const Rtc_address_month = &H05
Const Rtc_address_year = &H06
Const Rtc_address_control = &H07


'========== Für Testumgebung ==========

Const Main_testmodus = 1                                ' Testmodus für Testsystem

' ----- RTC DS1307 -----
Dim Rtc_control_test_value As Byte
' Rtc_control_test_value = &B00000011
Rtc_control_test_value = &B00000011
'                          |  |  ||               RS1  RS0  SQWE Output Frequency
'                          |  |  |+---- RS0        0    0    1 Hz
'                          |  |  +----- RS1        0    1    4.096 kHz
'                          |  +-------- SQWE       1    0    8.192 kHz
'                          +----------- OUT        1    1    32.768 kHz

' ----- DCF77 -----
Dim Sec_old As Byte                                     ' Schleifenvaraible SoftClock Sec
Sec_old = 99

Dim Dcfsec_old As Byte                                  ' Schleifenvariable DCF77 Sec
Dcfsec_old = 99

Dim Dcf_sync_count As Byte                              ' Zähler für Anzahl der Syncs
Dcf_sync_count = 0
```

```
' ----- Sonstiges -----
Dim Trace_out As Bit                                    ' Ausgabesteuerung Trace
Trace_out = 0

Dim Temp_byte_1 As Byte                                 ' Temporäre Byte Variable
Dim Temp_integer_1 As Integer                           ' Temporäre Integer Variable


'--------------------------------------------------------------------------------
' Prototyping
'--------------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========

' ----- Subroutinen -----
Declare Sub Rtc_write_control_reg(byval Value As Byte)

' ----- Funktionen -----
Declare Function Rtc_read_control_reg() As Byte

'========== Für Testumgebung ==========


'--------------------------------------------------------------------------------
' Konfiguration und Basiseinstellungen (Projekt und Testumgebung)
'--------------------------------------------------------------------------------


'========== Für Projekt ConvCtrl ==========

' ----- DCF77: Routinen und Einstellungen -----

'  CONFIG DCF77 = pin , timer = timer [ INVERTED=inv, CHECK=check, UPDATE=upd, UPDATETIME=updtime ,
TIMER1SEC=tmr1sec, SWITCHPOWER=swpwr, POWERPIN=pin, POWERLEVEL = pwrlvl , SECONDTICKS=sectick
,DEBUG=dbg , GOSUB = Sectic ]
'  PIN          = input pin that is connected to the DCF-77 signal
'  TIMER        = The timer that is used to generate the compare interrupts
'  INVERTED     = This value is 0 by default. When you specify 1, the compiler will assume you use
an inverted DCF signal.
'  CHECK        = The DCF-77 parity bits are checked
'  UPDATE       = Upd determines how often the internal date/time variables are updated with the DCF
received values.
'  UPDATETIME   = This value depends on the used UPDATE parameter.
'  TIMER1SEC    = 16 bit timers with the right crystal value can generate a precise interrupt that
fires every second.
'  SWITCHPOWER  = This option can be used to turn on/off the DCF-77 module with the control of a
port pin.
'  POWERPIN     = The name of a pin like pinB.2 that will be used to turn on/off the DCF module.
'  POWERLEVEL   = This option controls the level of the output pin that will result in a power ON
for the module.
'  SECONDTICKS  = The number of times that the DCF signal state is read.
'  DEBUG        = Optional value to fill 2 variables with debug info.
'  GOSUB        = The Sectic option will call a label in the main program every second.

' Config mit stündlichem Update
' Config Dcf77 = Pinb.2 , Timer = 1 , Inverted = 0 , Check = 1 , Update = 1 , Updatetime = 59 ,
Timer1sec = 1 , Debug = 1 , Gosub = Sectic

Config Dcf77 = Pinb.2 , Timer = 1 , Inverted = 0 , Check = 1 , Update = 0 , Timer1sec = 1 , Debug =
1 , Gosub = Sectic
'               |        |           |               |          |            |                |
|
' Use PinB.2 ---+        |           |               |          |            |                |
|
' Timer 1 is default ----+           |               |          |            |                |
|
' DCF77 signal wird nicht invertiert +               |          |            |                |
|
' Vergleich akt. Minutes mit vorheriger Minute ------+          |            |                |
|
' Perform kontinuierliches Update der empfangenen Daten --------+            |                |
|
' Updatetime ist nicht relevant da Update = 0 ------------------+            |                |
|
' Timer sorgt für SoftClock Update ------------------------------------------+                |
|
' Switchpower, Powerpin und Powerlevel wird nicht benutzt da Interface nicht vorhanden!       |
|
```

```
' Secondticks wird dem Compiler überlassen und nicht neu gesetzt!                          |
|
' Debug für die Variablen  bDCF_Pause und bDCF_Impuls ist aktiv ----------------------------+
|
' Jede Sekunde wird Gosub-Routine Dcf77_sectic angesprungen ----------------------------------------
------+

Enable Interrupts                                          ' Interrupts aktivieren
Config Date = Dmy , Separator = .                          ' Datumsformat einstellen


' ----- Konfiguration LCD Display -----
Config Lcdpin = Pin , Db4 = Portc.0 , Db5 = Portc.1 , Db6 = Portc.2 , Db7 = Portc.3 , E = Portc.4 ,
Rs = Portc.5
Config Lcd = 16 * 3 , Chipset = Dogm163v5
'Config Portc = Output                                      ' nicht zwingend erforderlich
'Config Lcdbus = 4                                         ' nicht zwingend erforderlich
Initlcd
Waitms 100
Cursor Off Noblink
Cls


' ----- Konfiguration für RTC DS1307 -----
Config Sda = Sda_port
Config Scl = Scl_port

I2cinit                                                    ' we need to set the pins in the proper
state

' Achtung: gemäß Datenblatt kann der DS1307 nur eine max frequency von 100 kHz
Config Twi = 100000                                        ' Taktfrequenz 100 kBaud

' TWI gleich einschalten, das macht BASCOM ansonsten erst beim I2CStart
Twcr = &B00000100                                          ' nur TWEN setzen

' Optionale Einstellungen für TWI Kommunikation die nicht zwingend erforderliche sind
' Twsr = 0                                                 ' Status und Prescaler Register
' Twbr = 152                                               ' Bit Rate Register, 72 = 100kHz   152 =
50kHz(0 bis 255)
' Config I2cdelay = 40


'========== Für Testumgebung ==========

' Com1 und Com2 für Trace-Output konfigurieren
Config Com1 = 38400 , Synchrone = 0 , Parity = None , Stopbits = 1 , Databits = 8 , Clockpol = 0
Config Com2 = 38400 , Synchrone = 0 , Parity = None , Stopbits = 1 , Databits = 8 , Clockpol = 0

Open "com1:" For Binary As #1                              ' Sitzt auf PE0 und PE1
Open "com2:" For Binary As #2                              ' Sitzt auf PD2 und PD3


'Konfiguration des Port A als Eingang für Tastaturabfrage
Config Porta = Input


' Überschrift ausgeben auf LCD-Display
Locate 2 , 1
Lcd "DCF77-Test MF V1"
Wait 2
Cls

Gosub Rtc_set_softclock

' Programmstart auf Trace ausgeben
#if Main_testmodus
   Print #1 , "** Enter MainLoop **"
#endif


'#####################################################################################
' Hauptprogramm
'#####################################################################################
'Hier ist die Programmhauptschleife
Do

   ' Tastaturabfrage Taste 1
   Debounce Taste_1 , 0 , Dcf77_trace_an , Sub
```

```
   ' Tastaturabfrage Taste 2
   Debounce Taste_2 , 0 , Dcf77_trace_aus , Sub

   ' Tastaturabfrage Taste 3
   Debounce Taste_3 , 0 , Control_reg_read , Sub

   ' Tastaturabfrage Taste 4
   Debounce Taste_4 , 0 , Control_reg_write , Sub

   If Trace_out = 1 Then

      ' Die Schleife ist so aufgebaut, dass zumindest einmal pro Sekunde eine Ausgabe erfolgt,
      ' falls auch der Timer 1 nicht funktionert z.B. wenn der Interrupt deaktiviert ist.
      ' Ändert sich _Sec (Soft-Clock Zeit) oder DCF_Sec (ausgewertetes DCF-Signal) wird die
      ' Schleife verlassen. Mit WAITMS 220 wird noch das Ende des Impulses (100 oder 200 mSec)
      ' abgewartet, damit auch dieser in der Ausgabe zur aktuellen DCF-Sekunden berücksichtigt wird.
      ' Eine Ausgabe erfolgt, wenn sich die Soft-Clock Sekunde oder die DCF-Sekunden ändert.
      ' Nach Synchronisation fallen diese beiden Zeitpunkte zusammen. Sofern die Soft-Clock
      ' nicht synchron läuft, können pro Sekunden 2 Zeilen erscheinen.

      For Temp_integer_1 = 1 To 78
         Waitms 10                                          ' Schleife läuft max. 780 ms
         If Sec_old <> _sec Then
            Exit For
         End If
         If Dcfsec_old <> Dcf_sec Then
            Exit For
         End If
      Next

      Waitms 220                                            ' Mit 220 ms ist Sekunde voll

      ' Sekunden updaten
      Sec_old = _sec                                        ' Update SoftClock Sekunden
      Dcfsec_old = Dcf_sec                                  ' Update DCF77 Sekunden

      ' Zusatzerläuterung:
      ' ------------------
      ' Die Impulslänge sollte zwischen 2 und 8 liegen
      ' Die Pausenlänge sollte zwischen 70 und 90 liegen

      ' Explanation dcf_status (aus BASCOm Hilfe)
      ' Bit 0 = The last reading of the DCF pin.
      ' Bit 1 = This bit is reserved.
      ' Bit 2 = This Bit is set, if after a complete time-stamp at second 58 the time-stamp
      '         is checked and it is OK. If after a minute mark (2 sec pause) this bit is
      '         set, the time from the DCF-Part is copied to the Clock-Part and this bit
      '         reset too. Every second mark also resets this bit. So time is only set,
      '         if after second 58 a minute mark follows. Normally this bit is only at
      '         value 1 from Second 58 to second 60/00.
      ' Bit 3 = This Bit indicates, that the DCF-Part should be stopped, if time is set.
      '         (at the option of updating once per hour or day).
      ' Bit 4 = This Bit indicated that the DCF-Part is stopped.
      ' Bit 5 = This bit indicates, that the CLOCK is configured the way,
      '         that during DCF-Clock is stopped, there is only one ISR-Call in one second.
      ' Bit 6 = This Bit determines the level of the DCF input-pin at the pulse (100/200 mSec part).
      ' Bit 7 = This bit indicates, that the DCF-Part has set the time of the Clock-part.

      ' Dcf77timezone()
      ' 0 : when there is no valid DCF77 data yet
      ' 1 : When In "Middle Europe Normal Time"
      ' 2 : When In "Middle Europe daylight saving Time"

      Print #1 , Time$ ; " " ; Date$ ; " " ; Time(dcf_sec) ; " " ; Date(dcf_day) ; " " ;
Bin(dcf_status) ; " " ; Bin(dcf_bits) ; " " ; Bdcf_impuls ; " " ; Bdcf_pause ; " " ; Dcf77timezone()

   Else
      ' Wenn Trace nicht aktiv trotzdem 1 Sekunde für LCD-Update warten
      Wait 1
   End If

   ' DCF77 Statusbits abfragen und maskieren
   Temp_byte_1 = Dcf_status And &H80

   ' Prüfung ob SoftClock durch DCF77 synchronisiert (Bit 7 aus Statusbyte)
   If Temp_byte_1 = &H80 Then

      Incr Dcf_sync_count                                   ' Sync-Counter erhören

      #if Dcf77_testmodus
```

```
         Print #1 , "SYNC: SoftClock wurde durch DCF77 synchronisiert!"
      #endif

      ' Zur Kontrolle bei erfolgreichem SYNC Status-Bit zurücksetzen
      Reset Dcf_status.7

      ' Sync-Kennzeichnung auf LCD ausgeben
      Locate 1 , 11
      Lcd "*"

      ' Anzahl der erfolgen Sync auf LCD ausgeben
      Temp_byte_1 = Makebcd(dcf_sync_count)
      Locate 2 , 11
      Lcd Bcd(temp_byte_1)

      ' DS1307 synchronisieren
      Gosub Rtc_set_rtc_clock

   End If

   ' ----- Datum und Uhrzeit BCD formatieren und formatiert auf LCD ausgeben -----
   Dcf77_hour = Makebcd(_hour)
   Dcf77_min = Makebcd(_min)
   Dcf77_sec = Makebcd(_sec)

   Locate 1 , 1
   Lcd Bcd(dcf77_hour) ; ":" ; Bcd(dcf77_min) ; ":" ; Bcd(dcf77_sec)

   Dcf77_weekday = Makebcd(_weekday)
   Locate 1 , 15
   Lcd Lookupstr(dcf77_weekday , Wochentag)

   Dcf77_day = Makebcd(_day)
   Dcf77_month = Makebcd(_month)
   Dcf77_year = Makebcd(_year)
   Locate 2 , 1
   Lcd Bcd(dcf77_day) ; "." ; Bcd(dcf77_month) ; "." ; Bcd(dcf77_year) ; " "

   ' Zeitzone ausgeben
   Locate 2 , 16 : Lcd Dcf77timezone()

   ' Statusbits und DCF-Bits auf LCD ausgeben
   Locate 3 , 1
   Lcd Bin(dcf_status)                                    ' Position 1 - 8
   Lcd Bin(dcf_bits)                                      ' Position 9 - 16

Loop


'## End Hauptprogramm ########################################################


'*******************************************************************************
' Interruptroutinen
'*******************************************************************************

'-------------------------------------------------------------------------------
' Service-Routine DCF77_sectic für DCF77 Softclock
' Routine wird als Parameter bei Config DCF77 übergeben und automatisch jede
' Sekunde ausgeführt. Damit ist SekundenTick für Gesamtsystem ohne zusätzlichen
' Timer möglich.
'-------------------------------------------------------------------------------
Sectic:
   #if Dcf77_testmodus
      Print #2 , "*"
   #endif
Return
'-- End Dcf77_sectic ---------------------------------------------------------


'*******************************************************************************
' Subroutinen
'*******************************************************************************

'-------------------------------------------------------------------------------
' RTC - Subroutine: Rtc_write_control_reg
' Funktion schreibt den übergebenen Bytewert in das Control-Register der
' RTC DS1307.
'
' Parameter:    Bytewert für neuen Control-Register-Inhalt
```

```
' Rückgabewert: keiner
'
'-------------------------------------------------------------------------------
Sub Rtc_write_control_reg(byval Value As Byte)

   #if Rtc_testmodus
      Print #1 , "Enter Rtc_write_control_reg"
   #endif

   I2cstart                                          ' START-Sequenz senden
   I2cwbyte Rtc_address_write                        ' Daten schreiben initiieren
   I2cwbyte Rtc_address_control                      ' Adresse übertragen
   I2cwbyte Value                                    ' Byte 1 übertragen
   I2cstop                                           ' STOP-Sequenz

End Sub
'-- End Rtc_write_control_reg ---------------------------------------------


'*******************************************************************************
' Funktionen
'*******************************************************************************

'-------------------------------------------------------------------------------
' RTC - Funktion: Rtc_read_control_reg
' Funktion liest das Control-Register der RTC DS1307.
'
' Parameter:    keiner
' Rückgabewert: Inhalt des Controllregisters als Byte-Wert
'
'-------------------------------------------------------------------------------
Function Rtc_read_control_reg() As Byte

   #if Rtc_testmodus
      Print #1 , "Enter Rtc_read_control_reg"
   #endif

   I2cstart                                          ' START-Sequenz senden
   I2cwbyte Rtc_address_write                        ' Daten schreiben initiieren
   I2cwbyte Rtc_address_control                      ' Adresse übertragen

   ' I2cstop                                         ' STOP-Bedingung wird nicht zwingend
gefordert
   I2cstart                                          ' START-Sequenz erneut senden
   I2cwbyte Rtc_address_read                         ' Daten lesen initiieren
   I2crbyte Temp_byte_1 , Nack                       ' Byte 1 lesen
   I2cstop                                           ' STOP-Sequenz

   ' Datenbyte als Rückgabewert übergeben
   Rtc_read_control_reg = Temp_byte_1

End Function
'-- End Rtc_read_control_reg ----------------------------------------------


'*******************************************************************************
' GOSubroutinen
'*******************************************************************************

'-------------------------------------------------------------------------------
' Gosub-Routine: Dcf77_trace_an
' Subroutine schaltet gobale Variable für Traceoutput für DCF77 und Softclock
' ein. Ausgabe des Trace erfolgt in Main-Loop
'-------------------------------------------------------------------------------
Dcf77_trace_an:

   #if Main_testmodus
      Print #1 , "Enter Dcf77_trace_an"
   #endif

   Trace_out = 1

Return
'-- End Dcf77_trace_an --------------------------------------------------


'-------------------------------------------------------------------------------
' Gosub-Routine: Dcf77_trace_aus
' Subroutine schaltet gobale Variable für Traceoutput für DCF77 und Softclock
' aus. Ausgabe des Trace erfolgt in Main-Loop
```

```
'--------------------------------------------------------------------------------
Dcf77_trace_aus:

   #if Main_testmodus
      Print #1 , "Enter Dcf77_trace_aus"
   #endif

   Trace_out = 0

Return
'-- End Dcf77_trace_aus ------------------------------------------------------


'--------------------------------------------------------------------------------
' RTC - Gosub-Routine: Rtc_set_rtc_clock
' Subroutine übernimmt bei erfolgreicher Synchronisation durch DCF77 die
' aktuellen Zeit und Datumsinformationen in die RTC DS1307
'--------------------------------------------------------------------------------
Rtc_set_rtc_clock:

   #if Rtc_testmodus
      Print #1 , "Enter Rcf_set_rtc_clock"
   #endif

   ' Daten zum Speichern in BCD Format aufbereiten
   Dcf77_hour = Makebcd(_hour)
   Dcf77_min = Makebcd(_min)
   Dcf77_sec = Makebcd(_sec)
   Dcf77_weekday = Makebcd(_weekday)
   Dcf77_day = Makebcd(_day)
   Dcf77_month = Makebcd(_month)
   Dcf77_year = Makebcd(_year)

   ' Traceoutput
   #if Rtc_testmodus
      Print #1 , "Schreibe Zeitinformationen in DS1307:"
      Print #1 , "Stunde    : " ; Bcd(dcf77_hour)
      Print #1 , "Minute    : " ; Bcd(dcf77_min)
      Print #1 , "Sekunde   : " ; Bcd(dcf77_sec)
      Print #1 , "Wochentag : " ; Bcd(dcf77_weekday)
      Print #1 , "Tag       : " ; Bcd(dcf77_day)
      Print #1 , "Monat     : " ; Bcd(dcf77_month)
      Print #1 , "Jahr      : " ; Bcd(dcf77_year)
   #endif

   ' RTC DS1307 synchronisieren
   I2cstart                                          ' START-Sequenz senden
   I2cwbyte Rtc_address_write                        ' Daten schreiben initiieren
   I2cwbyte Rtc_address_seconds                      ' Adresse übertragen = &H00

   I2cwbyte Dcf77_sec                                ' Byte 1 übertragen
   I2cwbyte Dcf77_min                                ' Byte 2 übertragen
   I2cwbyte Dcf77_hour                               ' Byte 3 übertragen
   I2cwbyte Dcf77_weekday                            ' Byte 4 übertragen
   I2cwbyte Dcf77_day                                ' Byte 5 übertragen
   I2cwbyte Dcf77_month                              ' Byte 6 übertragen
   I2cwbyte Dcf77_year                               ' Byte 7 übertragen
   I2cstop                                           ' STOP-Sequenz

Return
'-- End Rtc_set_rtc_clock ---------------------------------------------------


'--------------------------------------------------------------------------------
' RTC - Gosub-Routine: Rtc_set_softclock
' Subroutine übernimmt bei Aufruf der Routine die in der RTC "mitlaufenden"
' Zeitinformationen in die SoftClock.
' Damit kann bereits zu Systemstart ohne DCF77 Empfang mit "realer" Zeit
' gestartet werden.
'--------------------------------------------------------------------------------
Rtc_set_softclock:

   #if Rtc_testmodus
      Print #1 , "Enter Rcf_set_softclock"
   #endif

   ' RTC DS1307 auslesen
   I2cstart                                          ' START-Sequenz senden
   I2cwbyte Rtc_address_write                        ' Daten schreiben initiieren
   I2cwbyte Rtc_address_seconds                      ' Adresse übertragen = &H00
```

```
   ' I2cstop                                                ' STOP-Sequenz wird nicht zwingend benö-
tigt

   I2cstart                                                 ' START-Sequenz erneut senden
   I2cwbyte Rtc_address_read                                ' Daten lesen initiieren
   I2crbyte Dcf77_sec , Ack                                 ' Byte 1 lesen
   I2crbyte Dcf77_min , Ack                                 ' Byte 2 lesen
   I2crbyte Dcf77_hour , Ack                                ' Byte 3 lesen
   I2crbyte Dcf77_weekday , Ack                             ' Byte 4 lesen
   I2crbyte Dcf77_day , Ack                                 ' Byte 5 lesen
   I2crbyte Dcf77_month , Ack                               ' Byte 6 lesen
   I2crbyte Dcf77_year , Nack                               ' Byte 7 lesen und fertig (NACK)
   I2cstop                                                  ' STOP-Sequenz

   ' Traceoutput
   #if Rtc_testmodus
      Print #1 , "Lese Zeitinformationen aus DS1307:"
      Print #1 , "Stunde    : " ; Bcd(dcf77_hour)
      Print #1 , "Minute    : " ; Bcd(dcf77_min)
      Print #1 , "Sekunde   : " ; Bcd(dcf77_sec)
      Print #1 , "Wochentag : " ; Bcd(dcf77_weekday)
      Print #1 , "Tag       : " ; Bcd(dcf77_day)
      Print #1 , "Monat     : " ; Bcd(dcf77_month)
      Print #1 , "Jahr      : " ; Bcd(dcf77_year)
   #endif

   ' ausgelesene Daten in SoftClock Variablen übernehmen
   _hour = Makedec(dcf77_hour)
   _min = Makedec(dcf77_min)
   _sec = Makedec(dcf77_sec)
   _weekday = Makedec(dcf77_weekday)
   _day = Makedec(dcf77_day)
   _month = Makedec(dcf77_month)
   _year = Makedec(dcf77_year)

Return
'-- End Rtc_set_softclock ------------------------------------------------------


'--------------------------------------------------------------------------------
' Gosub-Routine: Control_reg_read
' Subroutine zum auslesen des Control Registers aus der RTC DS1307.
'--------------------------------------------------------------------------------
Control_reg_read:

   #if Main_testmodus
      Print #1 , "Enter Control_reg_read"
   #endif

   Temp_byte_1 = Rtc_read_control_reg()

   #if Main_testmodus
      Print #1 , "Inhalt Control Register " ; Bin(temp_byte_1)
   #endif

Return
'-- End Control_reg_read -----------------------------------------------------


'--------------------------------------------------------------------------------
' Gosub-Routine: Control_reg_write
' Subroutine zum schreiben des Control Registers in der RTC DS1307.
'--------------------------------------------------------------------------------
Control_reg_write:

   #if Main_testmodus
      Print #1 , "Enter Control_reg_write"
      Print #1 , "Schreibe Control Register " ; Bin(rtc_control_test_value)
   #endif

   Call Rtc_write_control_reg(rtc_control_test_value)

Return
'-- End Control_reg_write ----------------------------------------------------


'################################################################################

' Offene Devices COM1 und COM2 schließen
Close #2
```

```
Close #1


' Terminate program execution
End                                                    'end program


'--------------------------------------------------------------------------------
' Definition von globalen Konstantenfeldern
'--------------------------------------------------------------------------------

' ----- Daten für Wochentag -----
Wochentag:
Data "  " , "Mo" , "Di" , "Mi" , "Do" , "Fr" , "Sa" , "So"

'###################################### END ###################################
```

Software 10: Code zum Betrieb der DCF77 und RTC

8.15.2 Das DCF77-Signal

Das DCF77-Signal wird jede Sekunde einmal gesendet. Nun ist zu beachten, wie lang der zur jeweiligen Sekunde gesendete Impuls ist. Bei einer Länge von 0,1s handelt es sich um eine '0', bei einer Impulsdauer von 0,2s um eine '1'. Um den Beginn einer Sendung zu erkennen, entfällt das 59. Bit. Dies dient als Synchronisationsgrundlage. Insgesamt sieht das Bitmuster dann also folgendermaßen aus:

Sekunde	*Bedeutung*
0.	Minutenbeginn --> immer '0'
1.-14.	keine Bedeutung, normal '0' NEU Wetterdaten
15.	Reserveantenne aktiv
16.	Umstellung zwischen Sommer- bzw. Winterzeit
17.	Sommerzeit
18.	Winterzeit
19.	Schaltsekunde
20.	Zeitbeginn
21.-27.	Minute 1, 2, 4, 8, 10, 20, 40
28.	Prüfbit Minute
29.-34.	Stunde 1, 2, 4, 8, 10, 20
35.	Prüfbit Stunde
36.-41.	Tag 1, 2, 4, 8, 10, 20
42.-44.	Wochentag 1, 2, 4
45.-49.	Monat 1, 2, 4, 8, 10
50.-57.	Jahr 1, 2, 4, 8, 10, 20, 40, 80
58.	Prüfbit Datum (36. - 57.)
59.	wird nicht gesendet (siehe oben)

Tabelle 115: Bit-Bedeutung DCF77-Signal

Unten ein Beispiel für eine DCF77 Datenreihe (die untere Reihe dient lediglich der Orientierung):

```
00000000000000000010010000010111001010100011111000100010000001
01234567890123456789012345678901234567890123456789012345678
```

- Wie man sieht, ist das 17. Bit gesetzt, wir haben also Sommerzeit.
- Dann ist Bit 20 wieder gesetzt --> Zeitbeginn.
- Bei den Minuten ist lediglich Bit 26 gesetzt, es waren also 20 Minuten.
- Da die Anzahl der gesetzten Bits bei den Minuten ungerade war, ist das Prüfbit (28.) gesetzt.
- Da Bit 29, 30 und 33 gesetzt sind, ergibt sich die Stundenzahl zu 13.
- Bei einer Zahl von 3 gesetzten Bits (ungerade) muss das Prüfbit (35.) wieder gesetzt sein.
- Hiermit wäre bereits die Uhrzeit zu 13:20 Uhr bestimmt.

- Analog kann man sich auch das Datum ausrechnen.

Ausgewertet wird das Signal bei mir mittels eines Timers sowie des externen Interrupts INT0. Wird am Pin eine steigende Flanke erkannt, wird der Interrupt ausgelöst und vermerkt in einer Variablen den Signalbeginn. Ein kontinuierlich laufenden Timerinterrupt beobachtet nun den Pin und zählt die Signaldauer mittels einer Variablen hoch. Detektiert der Timerinterrupt, dass am Pin L-Pegel herrscht, wertet er die Signaldauer aus und setzt die Variablen zurück. Wird gerade kein Signal empfangen, zählt der Timerinterrupt die Pausenzeiten aus, um eine Synchronisation (siehe 59. Bit) zu erreichen. Alle Werte werden in ein Array geschrieben. Ist dieses voll, wurde ein komplettes Zeitsignal empfangen. Nun kann dieses ausgewertet und die Uhrzeit entsprechend aktualisiert werden.

Achtung:
Während der Dimmer arbeitet ist der Empfang des DCF77-Uhrenmoduls zu deaktivieren. Da es sonst zu zeitlichen Verschiebungen bei der Nulldurchgangserkennung kommen kann.

8.15.3 Prototyp zur DCF77 Anbindung

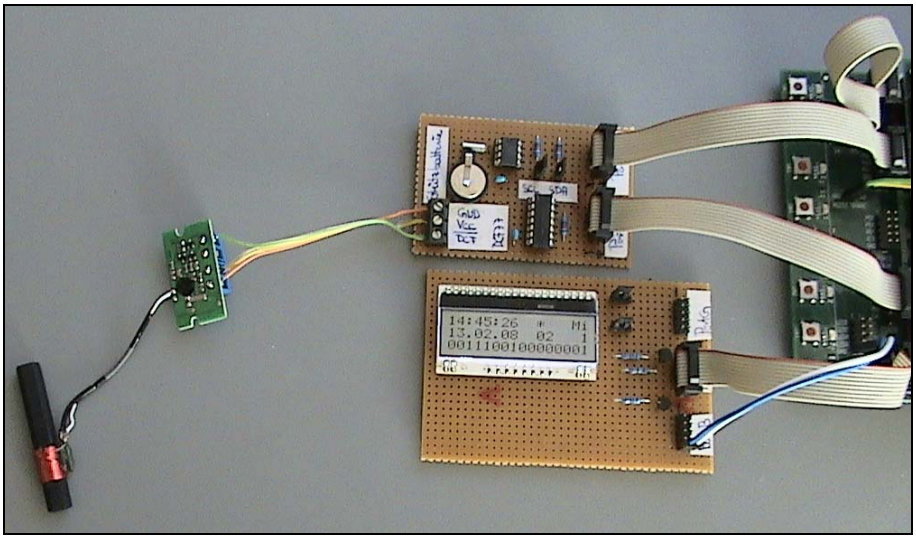


Abbildung 108: Prototypaufbau DCF77 und RTC mit LCD-Display

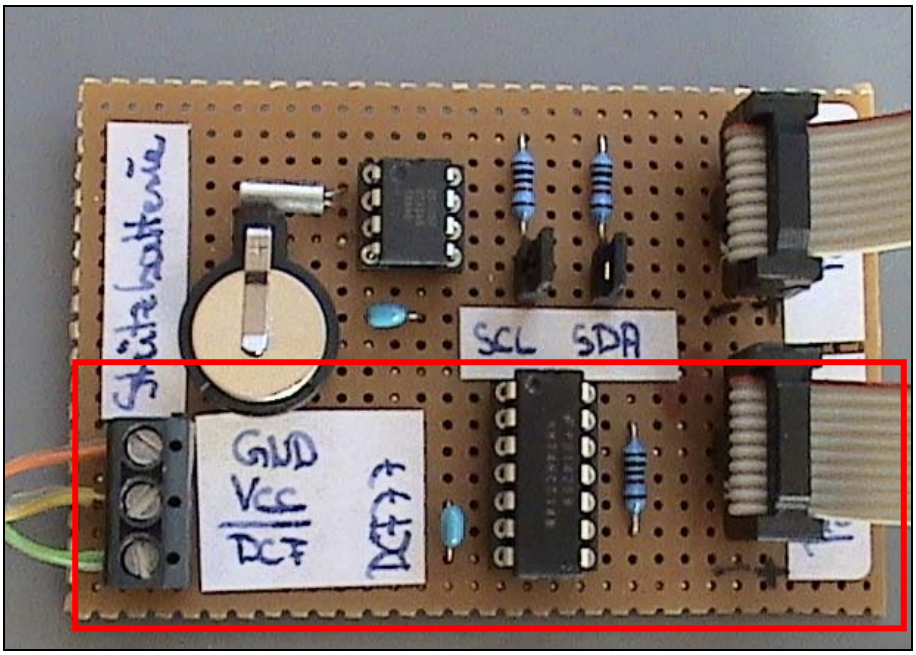


Abbildung 109: Beschaltung DCF77

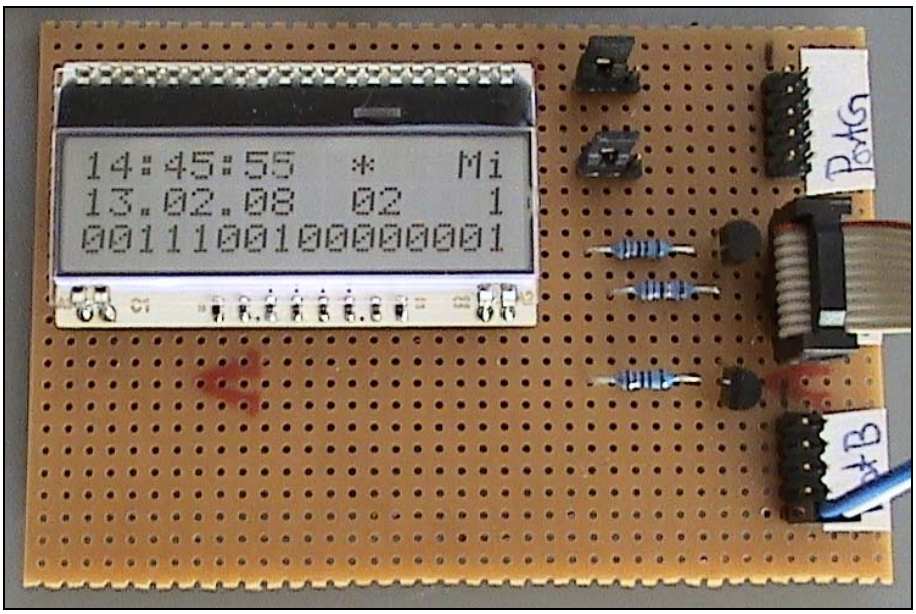


Abbildung 110: Beispielhafte Zeitausgabe über SoftClock mit DCF77

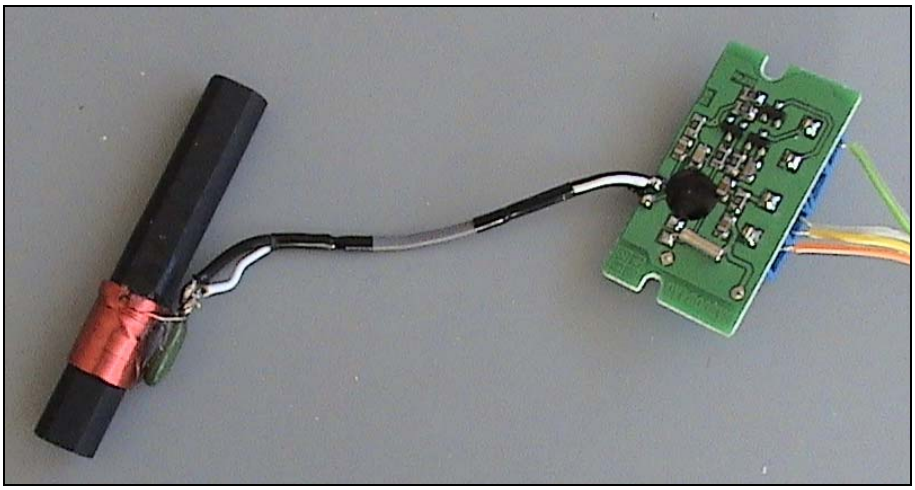


Abbildung 111: Das DCF77 Funkuhrmodul

8.16 RTC – Real-Time-Clock

Für die Verwaltung / Berechung der Urzeit im Projekt ConvCtrl wir der externe RTC-Baustein DS 1307 (Echtzeit-uhr) mit Stützbatterie eingesetzt. Diese Vorgehensweise dient zur Entlastung des Gesamtsystem gegenüber der ursprünglichenVorgehensweise bei der die Zeitrechung über SW und Timer-Interrupt geplant war.

Über diesen Weg muss er RTC nur einmal bei Systemstart über den DCF77-Empfänger aktualisiert werden. Als weitere Synchronisation wird optional jede Stunde bzw. alle 24 Stunden einmal neu synchronisiert.

Die RTC wird über eine 3V Stützbatterie mit Spannungs versorgt so dass die Urzeit auch gehalten werden kann wenn die Steuerung selbst abgeschaltet ist.

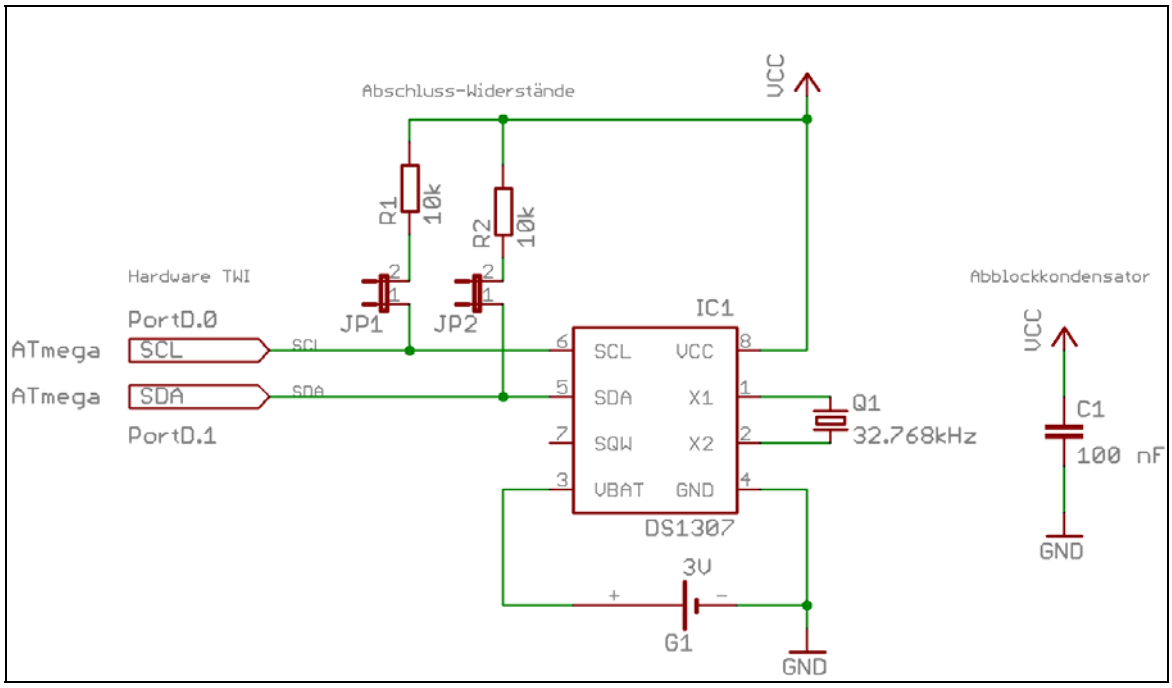
Die RTC DS1307 verfügt über einen I^2C-Bus so dass hier keine weiteren Ressourcen benötigt werden. Die Uhr kann parallel zum I^2C-EEPROM adressiert und betrieben werden.

Detailinformationen zum DS1307:

- TWI (I^2C)-Schnittstelle
- 100 kHz Protokoll

- Echtzeituhr für Jahr, Monat, Stunde, Minute und Sekunde
- geringe Stromaufnahme
- batteriegestützte Datenhaltung, weniger als 500 nA im Batteriebetrieb
- Daten liegen im BCD Format vor
- Register 0 = Sekunde, BCD, Bit7 Stop/Run RTC
- Register 1 = Minute, BCD
- Register 2 = Stunde, BCD
- berücksichtigt Schaltjahre bis 2100
- GerätelD 0B1101xxxx = 0xD0
- keine Adresserweiterung, also nur eine RTC in einem Bus

Die Beschaltung des DS1307 RTC:



Schaltbild 38: Aufschaltung des DS1307 an ATmega

Bauteile:

Stückliste: RTC			
Widerstände		*Halbleiter*	
R1, R2	10k Ω	IC1	Real-Time-Clock DS1307
Kondensator		*Sonstiges*	
C1	Abblockkondesnator 100nF	JP1, JP2	Jumper 2-polig
		Q1	Quarz 32.768 kHz
		G1	Stützbatterie 3V mit Halter

Tabelle 116: Stückliste Beschaltung RTC

Ressourcenzuordnung zum ATmega128:

Nummer	*Schaltbild*		*Ressource ATmega128*	
1	SDA	[SDA]	PortD.1	[SDA] (I^2C)
2	SCL	[SCL]	PortD.0	[SCL] (I^2C)

Tabelle 117: Ressourcenzuordnung ATmega128 für RTC

Basics zum I^2C-Bus:

Der I^2C-Bus muss mit PullUp-Widerständen auf High gezogen werden. Dies darf aber nur einmal geschehen und sollte nur von einem Bus-Teilnehmer-Modul bzw. am Anfang / am Ende der Busleitung passieren. Um Reflexionen auf den Leitungen zu vermeiden empfielt es sich die PullUps am Ende des I^2C-Busses anzubringen.

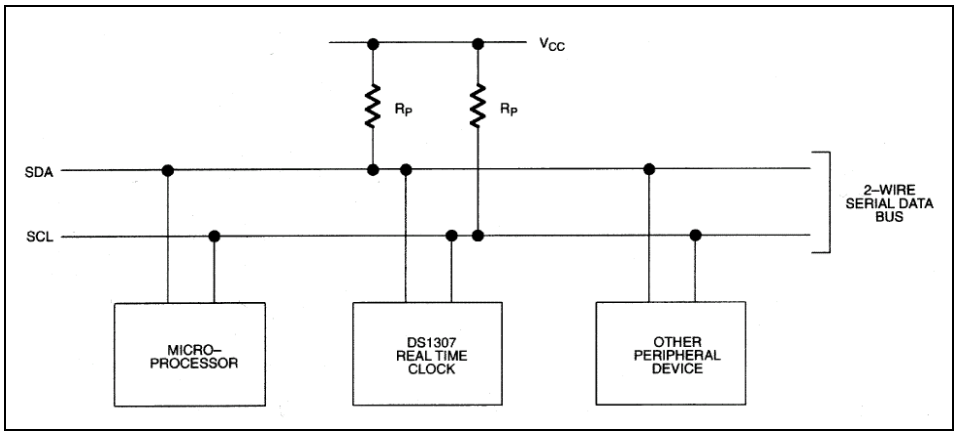


Abbildung 112: TWI 2-Wire BUS-Konfiguration

Adressierung DATA READ:

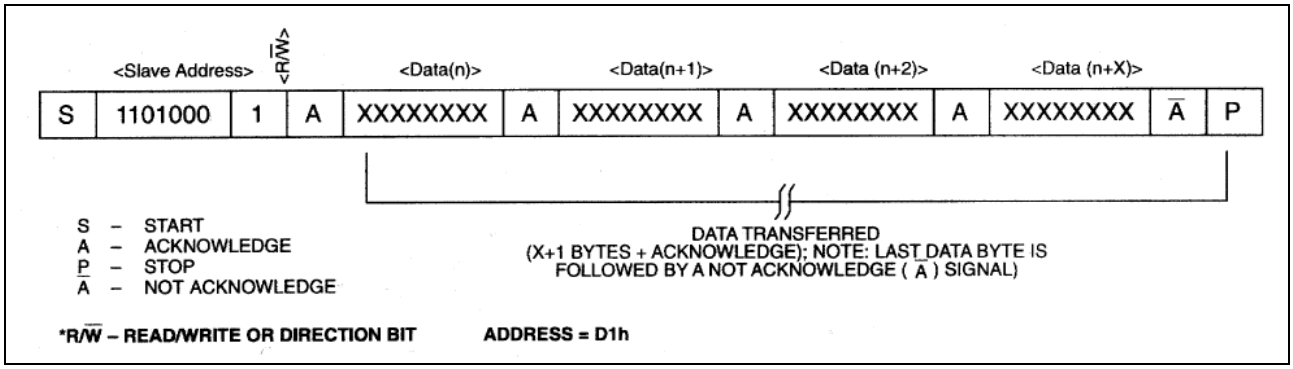


Abbildung 113: Adressierung DATA READ – DS1307

Adressierung DATA WRITE:

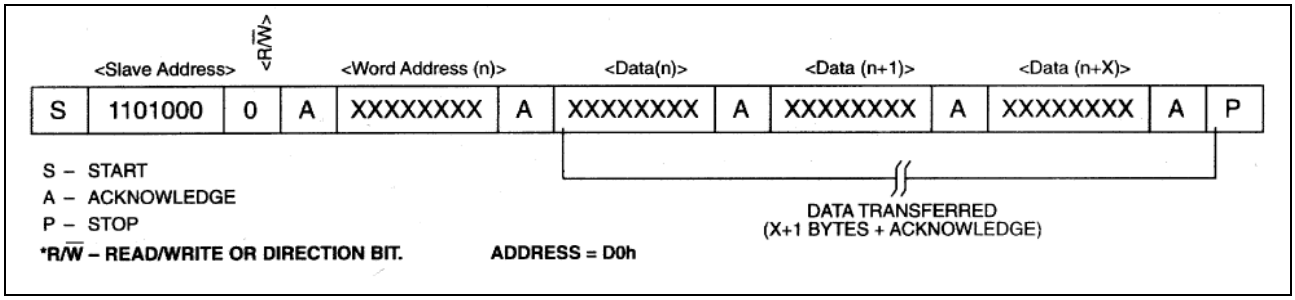


Abbildung 114: Adressierung DATA WRITE – DS1307

Die Kommunikationsprotokolle des DS1307 sind dem Datenblatt zu entnehmen. Im folgenden werden die I²C-Protokolle für die Echtzeituhr betrachtet:

- Initialisierung, aktivieren
- Uhr stellen (Sekunde, Miute, Stunde)
- Zeit auslesen (Sekunde, Minute, Stunde)

8.16.1 Initialisierung (stellen) und aktivieren der Echtzeituhr

Die Initialisierung des DS1307 ist verhältnismäßig einfach. Die aktuellen Zeitdaten sind fortlaufend ab Register 0 in der Echtzeituhr abgelegt.

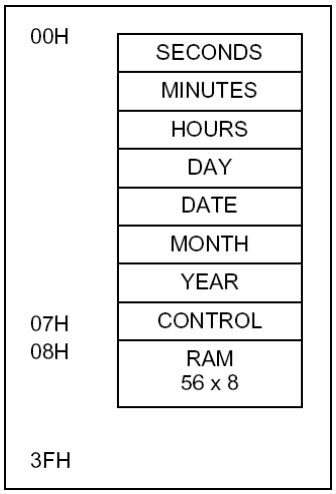


Abbildung 115: DS1307 Adress-MAP

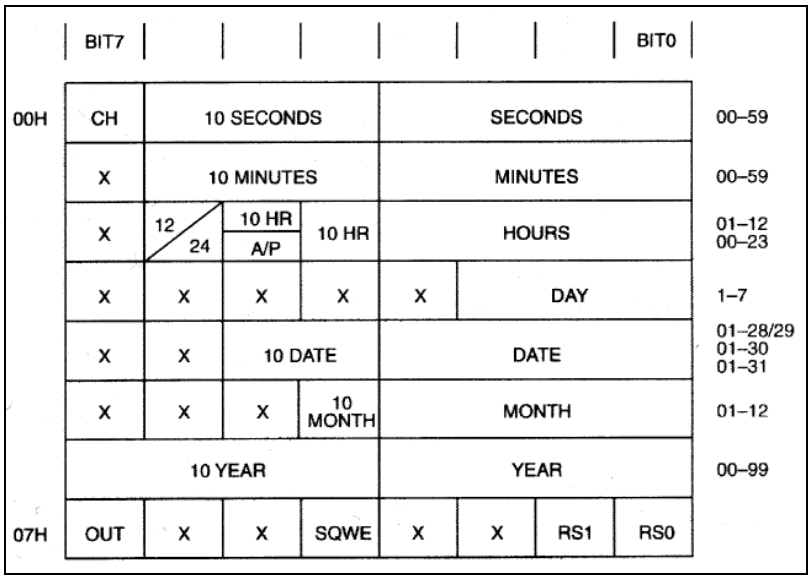


Abbildung 116: DS1307 Registers

Das höchstwertige Bit des Registers 0 wird mit CH bezeichnet (= SECONDS). CH steht für CLOCK HALT. Damit lässt sich die Uhr anhalten (CH=1) und auch starten (CH=0). Das Einfachste ist, beim Stellen der Uhr den Wert 0x00 in das Sekundenregister zu schreiben; damit startet die Uhr.

Dem Datenblatt ist die folgende Protokollsequenz zum Schreiben des Registers des DS1307 entnommen:

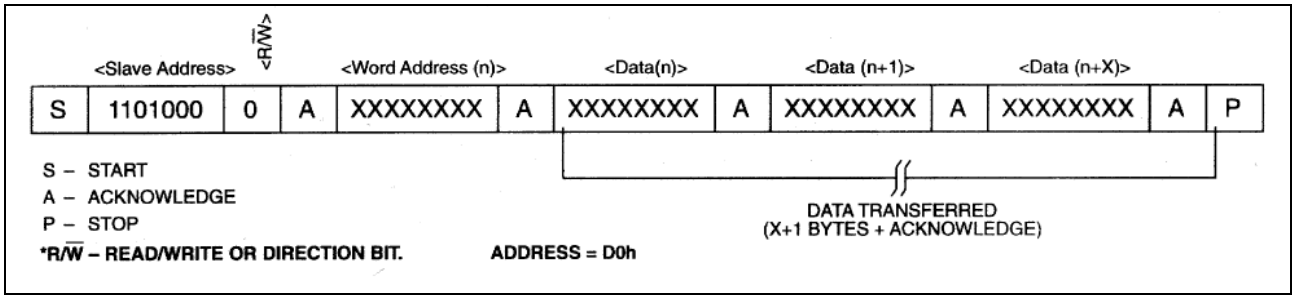


Abbildung 117: Daten in DS1307 schreiben

Zu beachten ist, dass nur die erste Adresse (Startadresse) der Schreiboperation angegeben werden muss und dann fortlaufend Datenbytes gesendet werden könnnen. Der DS1307 stellt den Adresspointer automatisch weiter (autoincrement). Es sind also folgende I²C Befehle nötig, um die Echtzeituhr für die Messungen der Uhrzeit zu konfigurieren:

1. den ATmega als TWI-Master initialisieren
2. die START-Sequenz auslösen
3. Register Nummer 0,1 und 2 (Sekunde, Minute und Stunde) laden

   a. DS1307 adressieren
   b. Register 0 adressieren
   c. Sekunden schreiben (Uhr starten)
   d. Minuten schreiben
   e. Stunde schreiben
4. die STOP-Sequenz senden

Genauere Details sind dem Beispielcode am Ende des Kapitel zu entnehmen. Im Beispielcode wird ausserdem auf Tag und Jahr eingegangen.

8.16.2 Auslesen der Uhrzeit

Nachdem die Uhr gestellt und gestartet wurde, läuft diese mit hoher Genauigkeit und bei eingelegter Batterie auch ohne Spannungsversorgung über das Board. Für das Auslesen der Uhrzeit gibt das Datenblatt der Echtzeituhrfolgendes Protokoll vor:

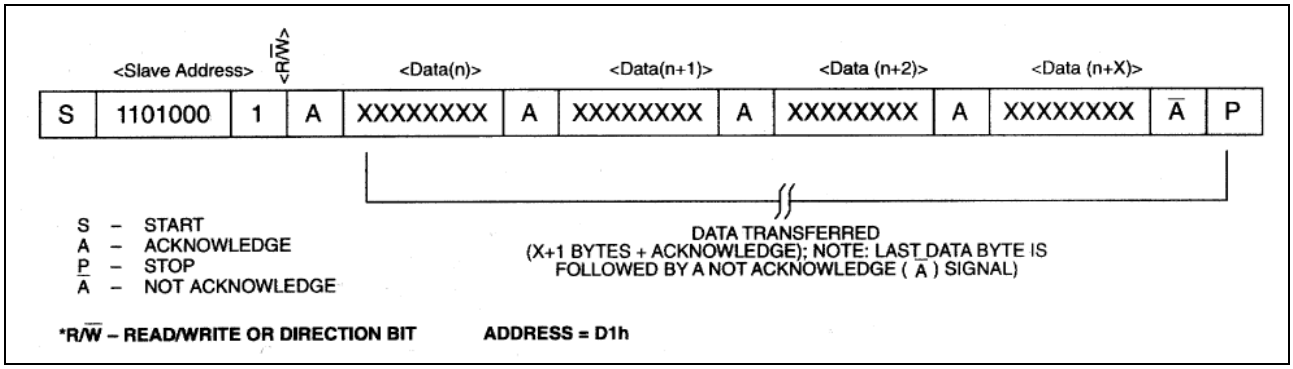


Abbildung 118: Daten aus dem DS1307 lesen

Es sind die folgenden I²C-Befehle nötig, um die Uhrzeit zu ermitteln:

1. den ATmega als TWI-Master initialisieren
2. die STARTSequenz auslösen
3. Register Nummer 0-2 (Stunde, Minute, Sekunde) auslesen
   a. DS1307 adessieren
   b. Register 0 adressieren, Registerpointer setzen ab Adresse 0 lesen
   c. Wiederholt TWI START senden
   d. DS1307 adressieren + Write-Bit = Sendeaufforderung an Slave
   e. Slave sendet Sekunde, Master empfängt und bestätigt Bereitschaft das nächstes Byte zu empfangen (Acknowledge)
   f. Slave sendet Minute, Master empfängt und bestätigt Bereitschaft das nächstes Byte zu empfangen (Acknowledge)
   g. Slave sendet Stunde, Master empfängt und negiert Bereitschaft das nächstes Byte zu empfangen (No), keine weiteren daten gewünscht
4. die STOP-Sequenz senden

8.16.3 BASCOM Beispielcode zur Ansteuerung der RTC

Die RTC DS1307 wurde im Rahmen der DCF77 Anbindung in Betrieb genommend. Die für die RTC DA1307 notwendigen Code-Teile befinden sich im BASCOM Codeteil zur DCF77.

8.16.4 Prototyp zur RTC DS1307

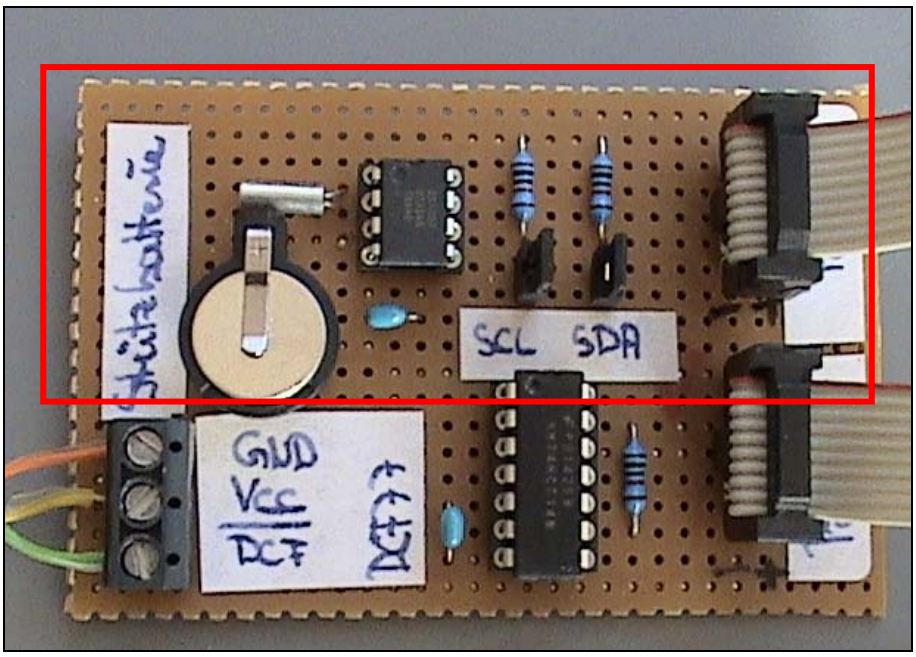


Abbildung 119: Beschaltung RTC DS1307

8.17 Lichtsensorik

Hintergrundinformationen:

Es gibt eine Vielzahl von Lichtsensoren die auf dem Markt erhältlich sind. Im Projekt wird der Lichtsensor TSL250R-LF der Firma TAOS eingesetzt.

Hinter TAOS verbirgt sicht die Firma Texas Advanced Optical Solutions Inc., ein Ableger der bekannten Texas Instruments Inc.

Der TSL250R besteht aus einer Fotodiode und einem integrierten Verstärker. Zum Anschluss an die Schaltung benötigt der TSL250R ausschließlich Versorgungsspannung (Pin 2) und Masse (Pin 1). An Pin 3 liefert der Sensor eine Ausgangsspannung welche proportional zum einfallenden Licht ist.

Die spektrale Empfindlichkeit ist in folgender Abbildung dargestellt, sie reicht von Ultraviolett (links) bis Infrarot (rechts) und hat ihr Maximum im sichtbaren Bereich des Lichtspektrums.

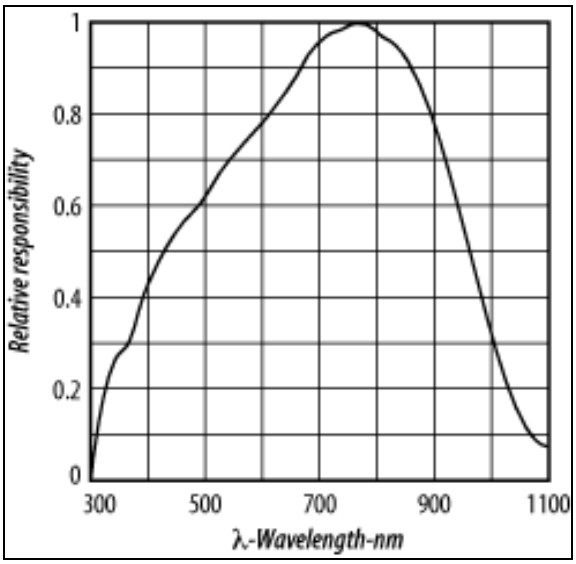
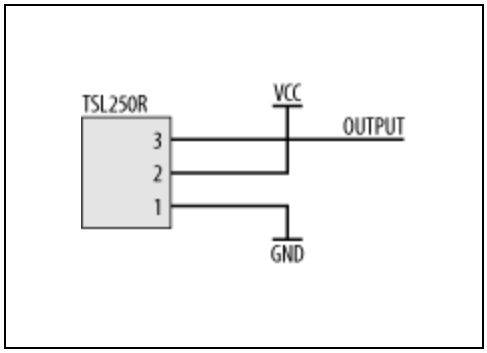


Abbildung 120: Spektrale Empfindlichkeit Lichtsensor

Der TSL250R arbeitet mit Betriebspannungen zwischen 2,70 und 5,50 V und einem typischen Strom von 1,1 mA. Die Basisbeschaltung des TSL250R ist ebenfalls sehr simpel und wird in der nächsten Schaltung dargestellt.



Schaltbild 39: Basisbeschaltung TAOS TSL250R

Das Maximum der Ausgangsspannung (unter voller Bestrahlung) für diesen Sensor liegt unter 4V, wenn der Baustein mit 5V betrieben wird. Somit haben wir die Wahl, den Sensor direkt ohne zusätzliche Beschaltung mit einer Referenzspannung von 5V über einen ADC-Kanal an den Kontroller anzuschließen. Ohne zusätzliche Verstärkung verlieren wir aber über den möglichen Spannungsbereich hinweg 1V Auflösung.

Für einen 8-Bit-ADC bedeutet eine 4V Eingangsspannung einen Messwert von 0xCC. Der Messwertebereich geht damit von 0x00 bis 0xCC. Abhängig von der Applikation ist dies kein Problem. Wenn es aber darum geht präzisere Lichtmessungen durchzuführen sollte der gesamte verfügbare ADC-Bereich ausgenutzt werden.

Für diesen Zweck wurde im Projekt ein Operationsverstärker verwendet welcher das Sensor-Ausgangssignal mit einem Verstärkungsfaktor von 1,25 auf die vom ADV möglichen 5V Maximum verstärkt.

Ein guter geeigneter und allgemein eingesetzter Prazisions-OpAMP ist der AD623 von Analog Devices. Er hat eine rail-to-rail Funktion, kann mit einer einzigen Betriebsspannung betrieben werden, benötigt wenig Strom und ist sehr leicht verständlich. Außerdem benötigt der AD623 nur einen einzigen Widerstand zum Einstellen des Verstärkungsfaktors.

Der Widerstandswert wird wie folgt berechnet:

$$R_G = \frac{100k\Omega}{(Gain-1)}$$

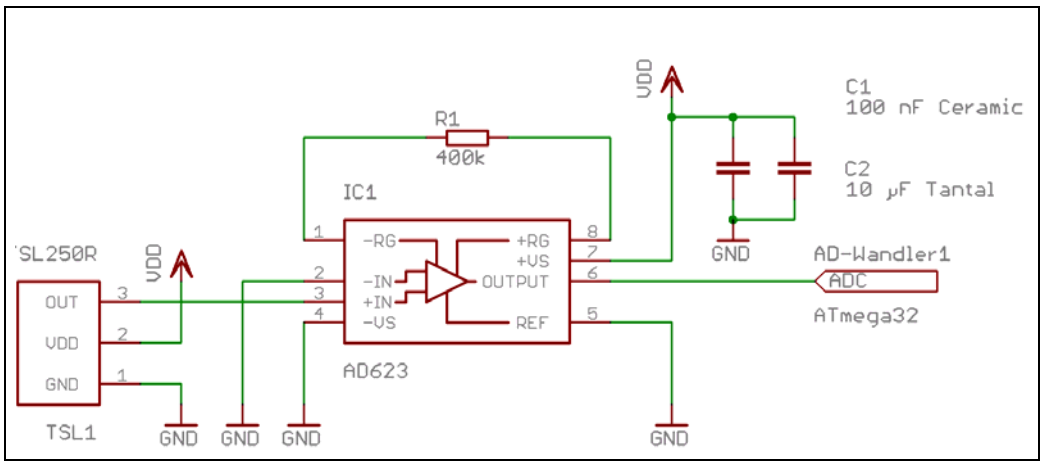
Für den benötigten Verstärkungsfaktor von 1,25 wird folgenden Widerstand benötigt:

$$R_G = \frac{100k\Omega}{(Gain-1)} = \frac{100k\Omega}{(1{,}25-1)} = \frac{100k\Omega}{0{,}25} = 400k\Omega$$

Der Widerstand sollte eine Toleranz von 1% oder besser besitzen. Prinzipiell eignen sich hier Metallschichtwiderstände. Die klassische Toleranz von Standardwiderständen beträgt 5%. Diese Toleranz reicht für den hier geplanten Einsatz nicht aus!

Beschaltung:

Die Schaltung mit dem TSL250R Interface an den AD623 ist in folgendem Schaltbild abgebildet:



Schaltbild 40: Schaltplan zum Lichtsensor

Der Ausgang des TSL250R Sensors (Pin 3) ist verbunden mit dem nichtinvertierenden Eingang des AD624 OpAmp (Pin 3). Der invertierende Eingang ist auf Masse gelegt. Der Widerstände welcher den Verstärkungsfaktor festlegt ist zwischen den Pins 1 und 8 beschaltet.

Die negative Spannungsversorgung des Bausteins (-VS) ist wegen des Betriebs mit einer einzigen Spannung auf Masse gelegt. Die positive Spannungsversorgung (+VS) wird mit VCC = +5V verbunden und mit zwei Kondensatoren gegen Masse gekoppelt.

Der Referenzeingang (REF) des OpAmp wird ebenso an Masse angeschlossen. Der Ausgang (Pin 6) wird direkt mit dem Analogeingang ADC1 des ATmega verschaltet.

Bauteile:

Stückliste: Lichsensor			
Widerstände		*Halbleiter*	
R1	400k Ω	TSL1	Lichtsensor TAOS TSL250R-LF
		IC1	Operationsverstärker AD623
		Kondensatoren	
		C1	Keramik-Kondensator 100 nF
		C2	Tantal-Kondensator 10µF

Tabelle 118: Stückliste Beschaltung Lichtsensor

Nummer	*Schaltbild*		*Ressource ATmega128*		
1	ADC0	[ADC0]	PortF.0	[ADC0]	(AD-Wandler)

Tabelle 119: Ressourcenzuordnung ATmega128 für Lichtsensor

8.17.1 Beschaltung des ADC0

Für die Analog-Digital-Umsetzung mit dem ATmega wird der Eingang ADC0 des ATmega verwendet.

Als Referenzspannung wird die Betriebsspannung des ATmega VCC auf Pin AVCC verwendet. Der externe Referenzspannungseingang AREF wird nicht verwendet.

Da sich im Projekt einige Störungsausstrahlende Komponenten befinden (z.B. Triac-Steuerung für Dimmung) wird eine spezielle im Datenblatt vorgeschlagene analoge Störsignalreduktion umgesetzt.

Umsetzung Analog Noise Canceling Techniques:

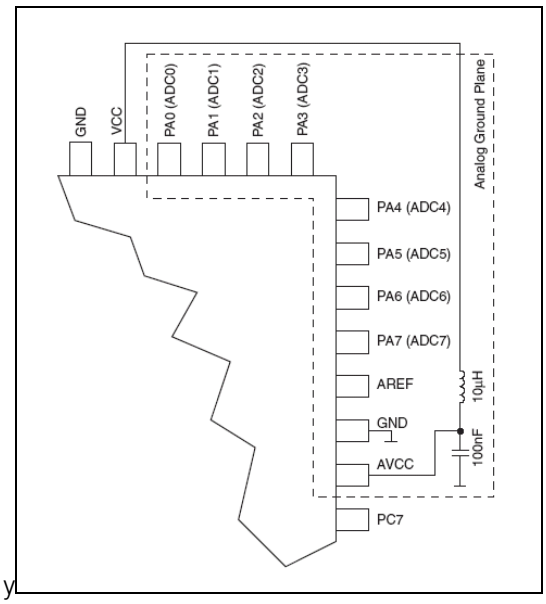
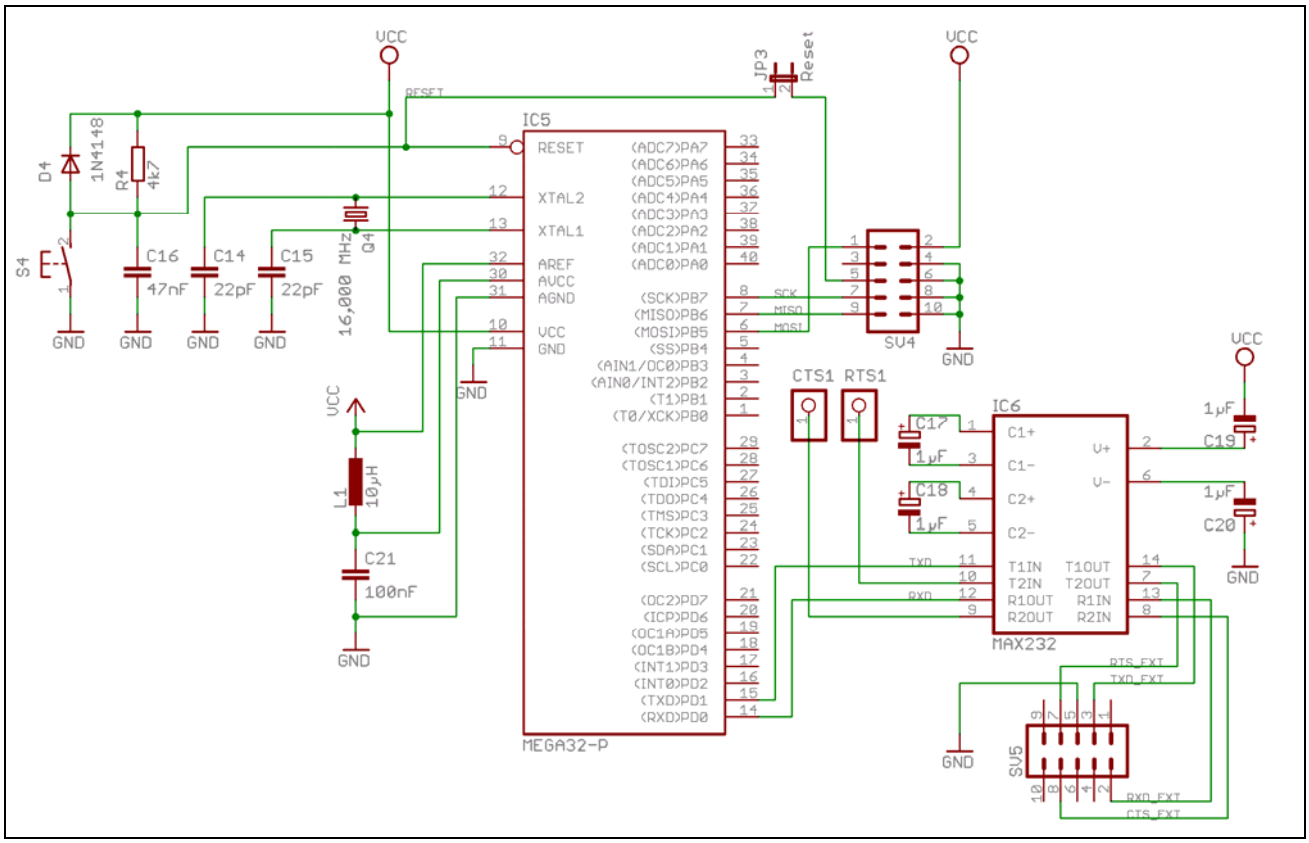


Abbildung 121: Analog Noise Canceling Techniques

Zur Umsetzung der Störsignalreduktion wird die bereits schon mehrfach abgebildete Basisbeschaltnug des ATmega für dieses Projekt um die notwendigen Komponenten angepasst.

Beschaltung:



Schaltbild 41: Beschaltung ATmega inkl. Störsignalreduktion

Bauteile:

Stückliste: Basisbeschaltung ATmega mit ISP, RS232 und ADC	
Widerstände	*Halbleiter*

R4	4k7 Ω	IC5	ATMEL AVR ATmega32
		D4	Diode 1N4148
		IC6	Maxim MAX232
Kondensatoren		*Sonstiges*	
C14, C15	Keramikkondensator 22 pF	S4	Taster
C16	Keramikkondensator 47 nF	Q4	Quarz 16.000.000 Hz
C17, C18, C19, C20	Elko 1µF/16V	SV4; SV5	Pin Header MA05-2
C21	Keramikkondensator 100 nF	JP3	Jumper 2-polig, 2,54mm Raster
Induktivitäten		CTS, RTS	IC-Sockelleiste als Patchfeld
L1	Spule 10 µH		

Tabelle 120: Stückliste Basisbeschaltung ATmega mit ISP, RS232 und ADC

8.17.2 BASCOM Beispielcode zum Lichtsensor TSL250R von TAOS

```
'#################################################################################
' LICHTSENSOR_TSL250R.BAS                                    Stand 11.12.2007
' -----------------------                                    (C) Markus Fulde
'
' Testprogramm zur Inbetriebnahme des Lichtsensors TSL250R von TAOS.
' Der Sensor arbeitet als Lich-zu-Spannungsumsetzer dessen Messbereich von 4V
' mittels Präzisions-OPAmp auf 5V full scale range erweitert wird.
' Das Testprogramm stellt alle notwendigen Routinen für den Zugriff bzw. die
' dafür notwendig AD-Wandlung und die Messertnormierung zur Verfügung.
'
'#################################################################################


'--------------------------------------------------------------------------------
' Compilerinstruktionen und Compilerdirektiven
'--------------------------------------------------------------------------------
$regfile = "m128def.dat"
$crystal = 16000000
$hwstack = 32
$swstack = 32
$framesize = 40

$baud = 38400

'--------------------------------------------------------------------------------
' Definition von Ressourcen
'--------------------------------------------------------------------------------

'========== Für Testumgebung  ==========

' LED's zum Test ob BASCOM-ADC-Routine andere PIN's von PortF beeinflusst
Led1 Alias Portf.1
Led2 Alias Portf.2
Led3 Alias Portf.3
Led4 Alias Portf.4
Led5 Alias Portf.5
Led6 Alias Portf.6
Led7 Alias Portf.7


'--------------------------------------------------------------------------------
' Definition von Variablen, Konstaten und Datentypen
'--------------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========
Const Light_testmodus = 1                                  ' Testmodus für Produktiv-Traceausgabe

Dim Light_channel As Byte                                  ' Variable für ADC-Kanal festlegen
Light_channel = 0                                          ' TSL250R hängt an ADC0

'========== Für Testumgebung ==========
Const Main_testmodus = 1                                   ' Testmodus für Testumgebung

Dim Temp_byte_1 As Byte                                    ' Temporärer Bytewert definieren


'--------------------------------------------------------------------------------
' Prototyping
```

```
'-------------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========
Declare Function Light_measurement() As Byte


'-------------------------------------------------------------------------------
' Konfiguration und Basiseinstellungen (Projekt und Testumgebung)
'-------------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========

' ----- Konfiguration ADC0 -----

' Konfiguration ADC Single-Mode und automatische Prescaler Setting
' Der Single-Mode wir bei BASCOM in Verbindung mit der Funktion GETADC() verwendet
' Der Prescaler teilt den internen Tackt durch 2, 4, 8,16,32,64 or 128 da der ADC
' einen Takt zwischen 50-200 kHz benötigt.
' Das AUTO Feature von BASCOM, setzt automatisch die höchste mögliche Taktrate
Config Adc = Single , Prescaler = Auto


'========== Für Testumgebung ==========
' LED-Ausgäng konfigurieren
Config Pinf.1 = Output
Config Pinf.2 = Output
Config Pinf.3 = Output
Config Pinf.4 = Output
Config Pinf.5 = Output
Config Pinf.6 = Output
Config Pinf.7 = Output


' Trace Output
#if Main_testmodus
   Print "** Enter MainLoop **"
   Print
#endif


'#################################################################################
' Hauptprogramm
'#################################################################################
'Hier ist die Programmhauptschleife
Do
   ' Funktion zum Lesen der Lichtstärke aufrufen und Werte ausgeben
   Temp_byte_1 = Light_measurement()
   Print "Lichtwert = " ; Temp_byte_1
   Print

   ' 2 Sekunde warten, das macht das Hauptprogramm ruhiger :-)
   Wait 5

   ' LEDs toggeln
   Toggle Led1
   Toggle Led2
   Toggle Led3
   Toggle Led4
   Toggle Led5
   Toggle Led6
   Toggle Led7

Loop
'## End Hauptprogramm ##########################################################


'*********************************************************************************
' Interruptroutinen
'*********************************************************************************

' gibts keine

'*********************************************************************************
' Subroutinen
'*********************************************************************************

' gibts keine

'*********************************************************************************
' Funktionen
```

```
'*******************************************************************************

'-------------------------------------------------------------------------------
' LIGHT - Function: Light_measurement
' Funktion liefert einen normierten Lichtwert im Wertebereich zwischen 0-15
' zurück welcher proportional der aktuell vom TAOS gelieferten Lichtintensität
' entspricht.
'
' Parameter:    keiner
' Rückgabewert: Bytewert 0 - 16 für Helligkeit
'-------------------------------------------------------------------------------
Function Light_measurement() As Byte

   ' ----- lokale Variablen -----
   Local Temp_word As Word
   Local Temp_byte As Byte
   ' ----- Programmcode -----

   ' Trace output
   #if Light_testmodus
      Print "Enter Light_measurement"
   #endif

   ' Die Spannungsversorgung für den ADC einschalten und den ADC starten
   Start Adc

   ' Single-Shot für ADC-Wert (Licht sampeln)
   Temp_word = Getadc(light_channel)

   ' Mit STOP ADC kann die Spannung abgeschaltet und der ADC gestopped werden
   Stop Adc

   ' Trace output
   #if Light_testmodus
      Print "ADC-Wert = " ; Temp_word
   #endif

   ' ----- Normierung durchfühen auf Wertebereich 0-15 -----

   ' ADC0 im ATmega128 arbeitet mit 10-Bit Auflösung = 0..1023
   ' Normierung auf 0-15 (4-Bit) = ADC / 64 bedeutet ShiftRight um 6 Bit
   Shift Temp_word , Right , 6
   Temp_byte = Temp_word And &H0F

   ' Messwert in Rückgabewert übergeben
   Light_measurement = Temp_byte

End Function
'-- End Light_measurement ------------------------------------------------------


'*******************************************************************************
' GOSubroutinen
'*******************************************************************************

' gibts keine

End                                                     'end program


'-------------------------------------------------------------------------------
' Definition von globalen Konstantenfeldern
'-------------------------------------------------------------------------------

' gibts keine

'###################################### END ######################################
```

Software 11: Code zum Betrieb des TSL250R

8.17.3 Prototyp zum TSL250R

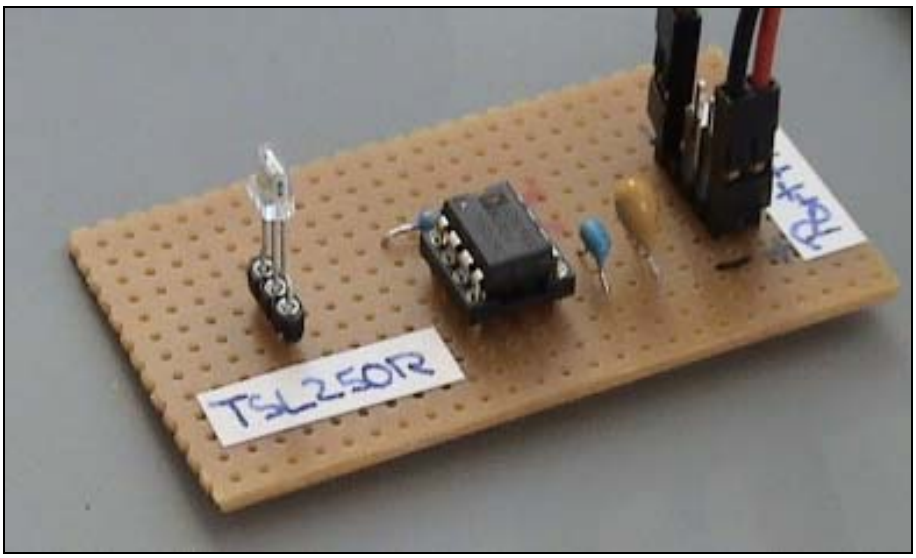


Abbildung 122: Beschaltung TAOS TSL250R

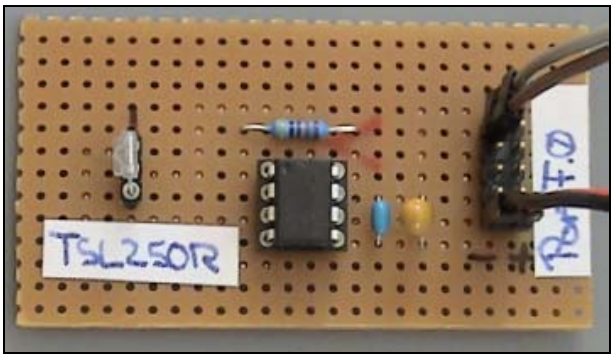


Abbildung 123: Detailansicht Beschaltung Lichtsensor

8.18 Akustikausgabe und Sound

Es gibt für die Realisierung einer Sonudausgabe im Projekt zwie möglich Lösungen.

1. Variante 1 mit Schwingkreis
2. Variante 2 mit PWM über den ATmega

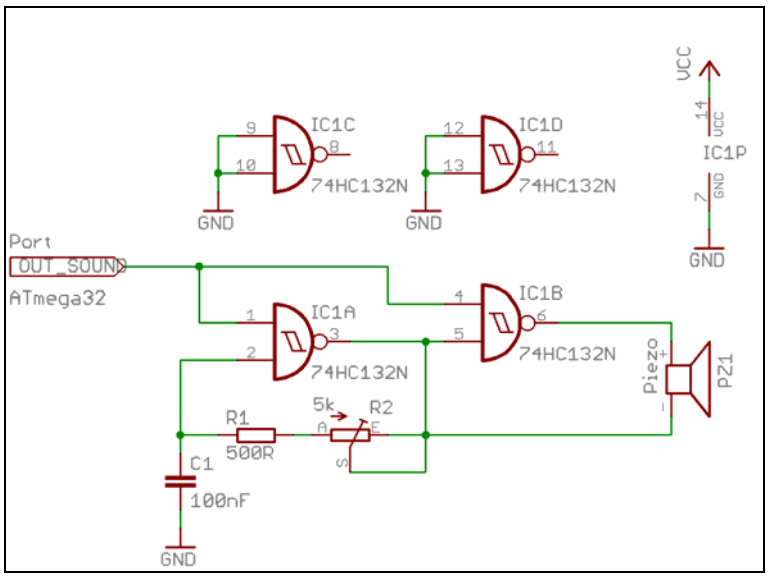
Da für Variante 2 keine Ressourcen mehr verfügbar sind um diesen Punkt sauber umzusetzen wir Variante 1 umgesetzt. Zur Vollständigkeit wird aber Variante 2 auch beschrieben.

8.18.1 Ansteuerung des Piezo-Schallwandlers (Schwingkreis)

Das zur Ansteuerung des Piezo-Signalgebers erforderliche Rechtecksignal wird nicht direkt vom Controller erzeugt. Zum einen würde dies zu Timing-Problemen mit anderen, zeitkritischen Programmteilen führen, zum anderen wären für eine ausreichende Amplitude bzw. Flankensteilheit ohnehin externe Treiberkomponenten erforderlich. Deshalb ist mit zwei NANDGattern (IC1) und R1+R2, C1 ein Oszillator aufgebaut, der bei Ausgabe eines High-Signals an Port 1.4 mit ca. 4 kHz schwingt.

Zur Justage der Frequenz wird ein stehendes Potentiometer der Firma Piher mit 5kΩ eingesetzt (Bestellnummer Firma Reichelt: *PIHER 10-S 5,0K*)

Beschaltung:



Schaltbild 42: Beschaltung Piezo-Signalgeber

Bauteile:

Stückliste: Beschaltung Piezo-Signalgeber			
Widerstände		*Halbleiter*	
R1	Widerstand 500 Ω	IC5	NAND 74HC132N
R2	Trimmpotentiometer stehen 5kΩ		
Kondensatoren		*Sonstiges*	
C1	Keramikkondensator 100 nF	PZ1	Piezo-Signalgeber, print

Tabelle 121: Stückliste Beschaltung Piezo-Signalgeber

Ressourcenzuordnung zum ATmega128:

Nummer	*Schaltbild*	*Ressource ATmega128*
1	SOUND [OUT_SOUND]	PortD.5 [PD5] (GPIO)

Tabelle 122: Ressourcenzuordnung ATmega128 für Piezo-Signalgeber

8.18.2 Ansteuerung eines Piezo-Schallwandlers mittels PWM

Durch den Einsatz des ATmega128 stehen mehr Ressourcen und technische Möglichkeiten zur Verfügung. Ein weitere Timer ermöglicht den Einsatz einer Beschaltung die in der Lage ist die Tonhöhe mittels Puls-Weiten-Modulation (PWM) zu verändern.

Zum Einsatz kommt der Piezo-Schallwandler RMP-14SP der Firma Ekulit.
Conrad Bestellnummer: *710107 - LN*

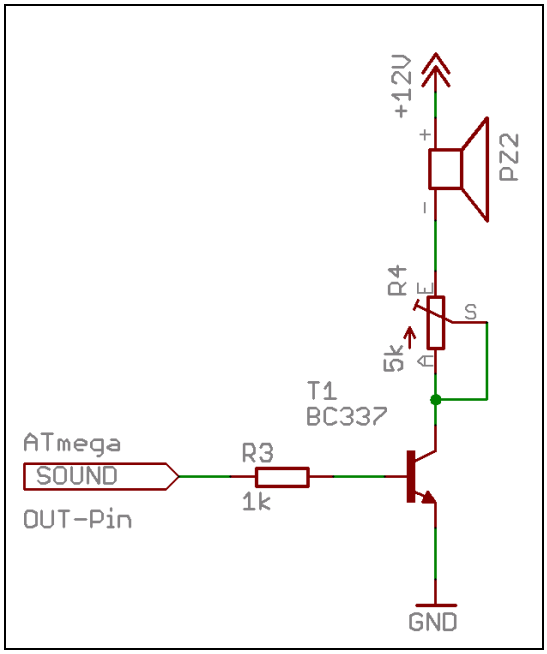
Der Piezo-Signalgeber ist für Print- und Anschlusslitzenmontage geeignet und hat ein extrem niedrige Stromaufnahme.



Typ:	RMP-14SP
Betriebstemperatur:	-20 - +70 °C
RM:	7.6 mm
Ausführung:	integrierte Elektronik
Betriebsspannung:	3 - 16 V/AC
Resonanzfreq.:	~4000 Hz
Stromaufnahme:	< 7 mA
Weitere technische Daten:	Abm. (AxB): 13.8 x 7.5 mm

Tabelle 123: Piezo-Schallgeber

Der Piezo wird über eine Transistor-Treiberstufe an 12V betrieben.



Schaltbild 43: Beschaltung Piezo-Signalgeber mit PWM

Bauteile:

Stückliste: Beschaltung Piezo-Signalgeber mit PWM			
Widerstände		*Halbleiter*	
R1	Widerstand 1k Ω	T1	Transistor BC337
R4	Potentiometer 5k Ω		
Sonstiges			
PZ2	Piezo-Schallgeber RMP-14SP		

Tabelle 124: Stückliste Beschaltung Piezo-Signalgeber

8.18.3 BASCOM Treibersoftware für Soundausgabe

```
'#################################################################################
' Piezo_Sound.BAS:                                          Stand 13.01.2008
' ----------------                                          (C) Markus Fulde
'
' Testprogramm zur Inbetriebnahme des Piezo-Schwingkreises zur Ausgabe eines
' über Potentiometer einsellbaren Signaltons.
' Das Testprogramm stellt alle notwendigen Routinen für den Zugriff zur
' Verfügung.
'#################################################################################


'--------------------------------------------------------------------------------
```

```
' Compilerinstruktionen und Compilerdirektiven
'-----------------------------------------------------------------------------
$regfile = "m128def.dat"
$crystal = 16000000
$hwstack = 32
$swstack = 32
$framesize = 40

$baud = 38400


'-----------------------------------------------------------------------------
' Definition von Ressourcen
'-----------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========
Sound_pin Alias Pind.5                                  ' GPIO Pin PortD.5
Sound_dataout Alias Portd.5                             ' GPIO Pin PortD.5

'========== Für Testumgebung  ==========
Taste_1 Alias Pina.0
Taste_2 Alias Pina.1
Taste_3 Alias Pina.2


'-----------------------------------------------------------------------------
' Definition von Variablen, Konstaten und Datentypen
'-----------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========
Const Sound_testmodus = 1

'========== Für Testumgebung ==========
Const Main_testmodus = 1

Dim Beep_time As Word
Beep_time = 100


'-----------------------------------------------------------------------------
' Prototyping
'-----------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========
Declare Sub Sound_beep(time_value As Word)

'========== Für Testumgebung ==========


'-----------------------------------------------------------------------------
' Konfiguration und Basiseinstellungen (Projekt und Testumgebung)
'-----------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========
Gosub Sound_konfig

'========== Für Testumgebung ==========

#if Main_testmodus
   Print
   Print "** Enter MainLoop **"
#endif


'#############################################################################
' Hauptprogramm
'#############################################################################
'Hier ist die Programmhauptschleife
Do
   ' Tastaturabfrage Taste 1
   Debounce Taste_1 , 0 , Sound_an , Sub

   ' Tastaturabfrage Taste 2
   Debounce Taste_2 , 0 , Sound_aus , Sub

   ' Tastaturabfrage Taste 3
   Debounce Taste_3 , 0 , Test_beep , Sub

Loop
'## End Hauptprogramm ######################################################


'*****************************************************************************
' Subroutinen
'*****************************************************************************

'-----------------------------------------------------------------------------
' SOUND - Subroutine: Sound_beep
' Subroutine gibt für die in ms übergebene Zeit einen Signalton aus
' Achtung, Routine wartet so lange bis Ton ausgegeben ist!
```

```
'
' Parameter:    time_value = Zeit in ms die der Beepton ausgegeben werden soll
' Rückgabewert: keine
'-------------------------------------------------------------------------------
Sub Sound_beep(time_value As Word)

   #if Sound_testmodus = 1
      Print "Enter Sound_beep mit Wartezeit " ; Time_value ; " ms"
   #endif

   Gosub Sound_an                                          ' Ton einschalten
   Waitms Time_value                                       ' warten
   Gosub Sound_aus                                         ' Ton ausschalten

End Sub
'-- End Sound_beep ------------------------------------------------------------


'*******************************************************************************
' GOSubroutinen
'*******************************************************************************

'-------------------------------------------------------------------------------
' SOUND - Gosub-Routine: Sound_konfig
' Konfiguration der notwendigen PIN's und Ports für die Ansteuerung des Piezo-
' Schwingkreises.
'-------------------------------------------------------------------------------
Sound_konfig:

   #if Sound_testmodus = 1
      Print "Enter Sound_konfig"
   #endif

   Config Sound_pin = Output                               ' PortD.5 als Output
Return
'-- End Sound_konfig -------------------------------------------------------


'-------------------------------------------------------------------------------
' SOUND - Gosub-Routine: Sound_an
' Routinen schaltet die Ausgabe des Signaltons ein.
'-------------------------------------------------------------------------------
Sound_an:

   #if Sound_testmodus = 1
      Print "Enter Sound_an"
   #endif

   Set Sound_dataout                                       ' Bit setzen
Return
'-- End Sound_an -----------------------------------------------------------


'-------------------------------------------------------------------------------
' SOUND - Gosub-Routine: Sound_aus
' Routinen schaltet die Ausgabe des Signaltons aus.
'-------------------------------------------------------------------------------
Sound_aus:

   #if Sound_testmodus = 1
      Print "Enter Sound_aus"
   #endif

   Reset Sound_dataout                                     ' Bit rücksetzen
Return
'-- End Sound_aus ----------------------------------------------------------


'-------------------------------------------------------------------------------
' Gosub-Routine: Test_beep
' Routinen ruft zum Test due Beep-Routine auf.
'-------------------------------------------------------------------------------
Test_beep:

   #if Main_testmodus
      Print "Enter Test_beep"
      Print "Parameter : " ; Beep_time
   #endif

   Call Sound_beep(beep_time)                              ' Soubroutine aufrufen
Return
'-- End Sound_aus ----------------------------------------------------------

End                                                        'end program

'####################################### END ####################################
```

8.18.4 Prototyp Piezo-Schwingkreis

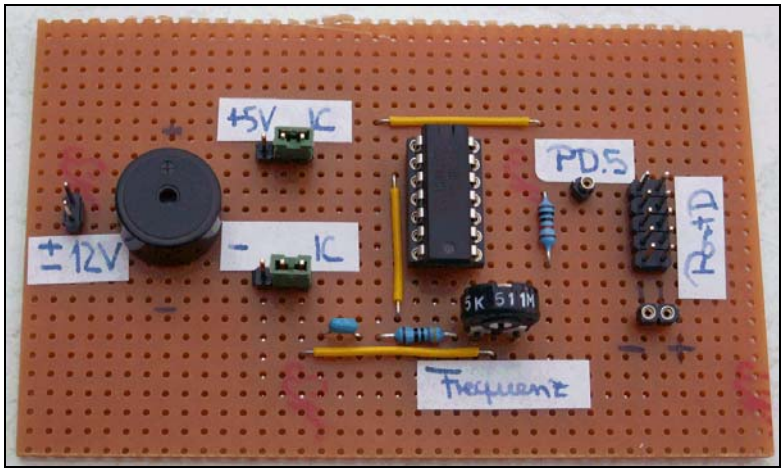


Abbildung 124: Prototyp des Piezo-Schwingkreises zur Soundausgabe

8.19 Tastatur und Tastenmatrix

8.19.1 Allgemeines zu Tastenmatrix an Mikrocontrollern

Es geht um den Anschluss vieler Tasten an einen Mikrocontroller. Grundsätzlich werden diese im Zeitmultiplexverfahren (zyklisch) abgefragt.

8.19.1.1 Grundsätzliches Schema

Die Tasten werden in einem mxn-Matrix angeordnet. Um möglichst wenig Anschlüsse für möglichst viele Tasten zu verwenden, wird eine quadratische Matrix angestrebt.

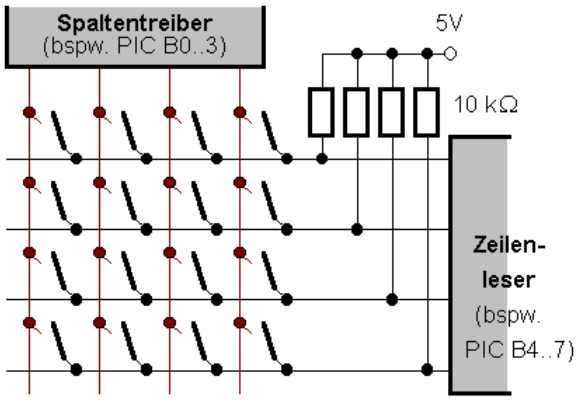


Abbildung 125: Tastenmatrix - 4x4-Tastenmatrix im Schlafzustand (rot=LOW)

Der Spaltentreiber ist ein Ausgabeport, der Zeilenleser ein Eingabeport des Mikrocontrollers. Die stets erforderlichen Widerstände sind oft bereits im Eingabeport des Mikrocontrollers integriert oder zuschaltbar. Deshalb können sie in der Schaltung entfallen!

Daraus ergibt sich allerdings auch die Notwendigkeit, mit Low-aktiven Signalen zu arbeiten, da alle Mikrocontroller Hochzieh-Widerstände (Pull-Up) und nie Tiefzieh-Widerstände (Pull-Down) haben. Gelegentlich zu sehende Tastenmatrizen mit externen Pull-Up- oder gar Pull-Down-Widerständen zeugen von der Stümperhaftigkeit des Entwicklers.

Wichtig: Ohne Entkopplungsdioden ist nur die Erkennung einer oder zweier Tasten sicher möglich, nicht mehr! Für ein klavierartiges Musikinstrument siehe unten.

Die Trivialmatrix erfordert 2n Leitungen für n2 Tasten. Bei rechteckicken Matrizen ist das Verhältnis etwas ungünstiger, m+n Leitungen für m*n Tasten.

8.19.1.2 Ruhestromfreie Abfrage

Dazu wird in oben gezeigter Schaltung ein Eingabeport benötigt, welches auf allen Leitungen interruptfähig ist, und das im Tiefschlafmodus mit abgeschaltetem Oszillator. Diese Eigenschaft hat das Port B (Bits 4..7) des PIC-Mikrocontrollers. (Mir erscheint spontan kein anderer Mikrocontroller geeignet.)

In Fernbedienungen wird ausschließlich diese Abfrageart verwendet.

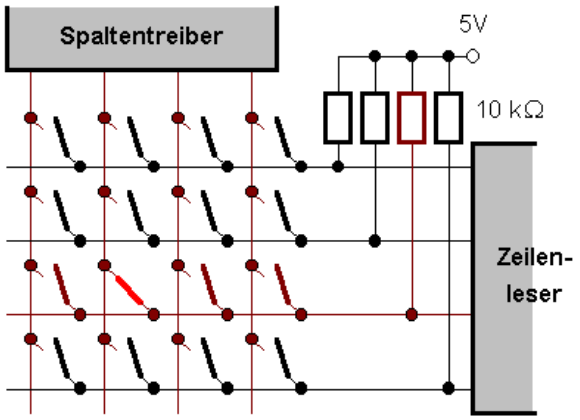


Abbildung 126: Tastenmatrix - aufgeweckt, weil Taste gedrückt

Wird irgendeine Taste gedrückt, wird ein Eingabe-Portpin nach LOW gezogen, der Microcontroller erwacht aus seinem Tiefschlaf, der Oszillator läuft an, und der Befehl nach SLEEP bzw. die Interruptroutine wird angesprungen.

Diese beginnt nun mit der spaltenweisen Abfrage.

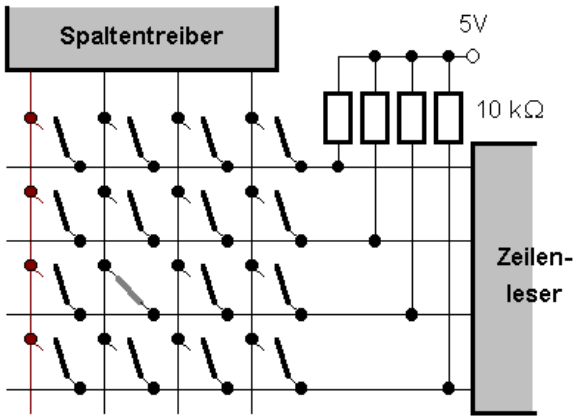


Abbildung 127: Tastenmatrix - Linke Spalte wird abgefragt: Pech!

Die nicht abgefragten Spalten bekommen ein schwaches HIGH oder werden hochohmig geschaltet, damit weitere gedrückte Tasten keine Kurzschlüsse zwischen Treiberausgängen bewirken!

Dabei wird so lange vorgegangen, bis eine gedrückte Taste erkannt wird.

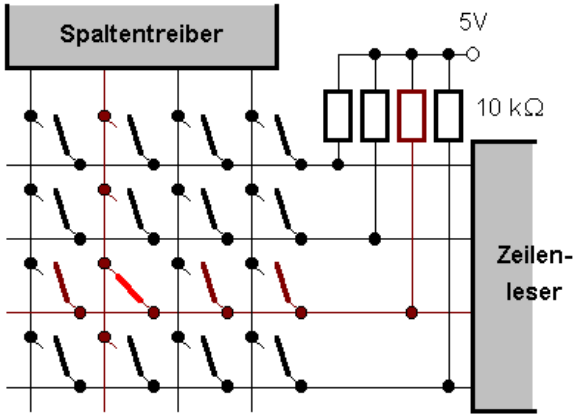


Abbildung 128: Tastenmatrix - Zweite Spalte wird abgefragt: Glück!

Eventuell ist gar keine Taste (mehr) gedrückt, dann ist nichts mehr zu tun.

Im Regelfall wird der Tastenkode verarbeitet und der entsprechende Befehl ausgeführt.

Das Erkennen des Loslassens aller Tasten erfolgt, indem der Ursprungszustand (alles LOW) des Spaltentreibers hergestellt wird. Dann müssen alle Zeilenleitungen HIGH werden.

8.19.1.3 Zwei gedrückte Tasten erkennen

Zwei Tasten lassen sich durch fortwährendes Spalten-Abfragen ermitteln. Sind die beiden Tasten in einer Spalte, ist auch mal mehr als eine Zeilenleitung aktiv.

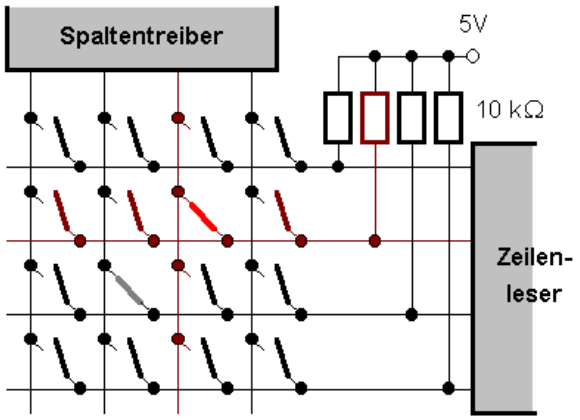


Abbildung 129: Tastenmatrix - Zwei Tasten sind problemlos

Mehr als 2 Tasten?
Mehr als 2 Tasten lassen sich nicht sicher auswerten!

Wenn im Bild 5 die Taste (2;2) gedrückt wird, ist es auf Grund des Kurzschlusses zwischen Spalte 2 und 3 sowie Zeile 2 und 3 für den Mikrocontroller nicht möglich, zwischen (2;2) oder (3;3) als dritte Taste zu unterscheiden.

Wird dagegen jede andere Taste gedrückt, ist deren Detektion nur eine Frage intelligenter Software.

8.19.1.4 Beliebige Tasten gleichzeitig

Dies erfordert Entkopplungsioden an jeder Taste. Es gibt Einzelkontakte, die diese Diode bereits integriert haben.

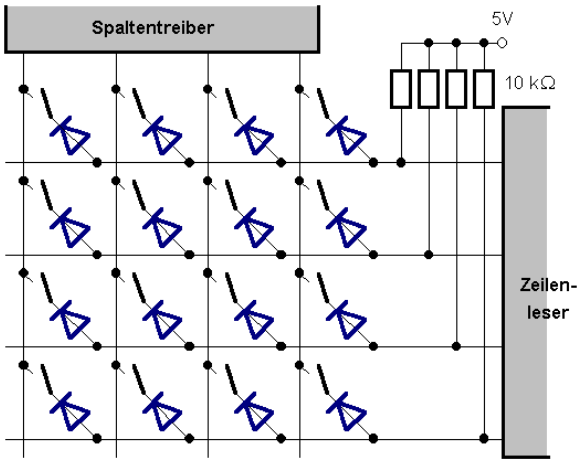


Abbildung 130: Tastenmatrix - Alle Tasten sind hiermit kein Problem

Hierbei genügt ein reines Ausgabeport (bspw. eines Port-Expanders oder eines Schieberegisters) als Spaltentreiber, weil keine Kurzschlüsse durch mehrfache Tastendrücke möglich sind.

Ob die Software tatsächlich alle Tastenkombinationen auswertet ist Frage des Problems und des Geschmacks. Bei einem Klavier bspw. genügt es sicherlich, max. 10 gedrückte Tasten (für 10 Finger) zuzulassen.

8.19.1.5 Matrizen mit weniger Portpins

Bei Verzicht auf die Hauptdiagonale und Ersatz durch Dioden kann die Anzahl der Portpins halbiert werden.

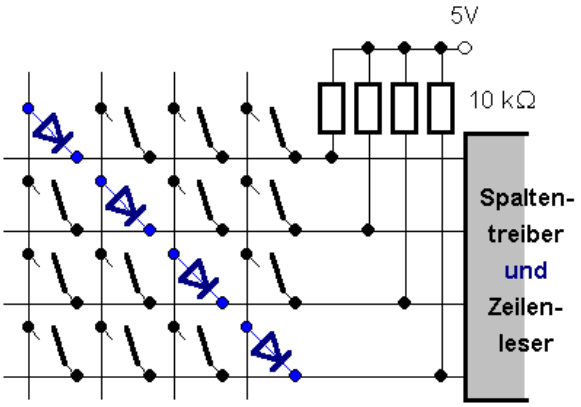


Abbildung 131: Tastenmatrix - Halbierte Portpin-Zahl, aber Dioden

Das Abfrageschema ist wie oben, nur dass die gerade aktivierte Spalte stets als aktive Zeile zurückkommt (muss ausmaskiert werden).

Zwei Tasten gleichzeitig gehen hier nicht!

Ein Schlafmodus ist mit dieser Schaltung nicht möglich! (Bestenfalls nur für einige bestimmte Tasten.)

Auf n Portpins kommen n2-n = n*(n-1) Tasten. Günstig für Kodeschlösser u. ä. ist die hier gezeigte Matrix mit 12 Tasten.

Eine Anreihung einer nach Masse gehenden Tastenreihe ist hier genauso möglich wie im folgenden Fall, das ergäbe n2, hier 16 Tasten.

8.19.1.6 Weniger Portpins ohne Dioden

Bei Verzicht auf eine halbe Matrix kann man, bei geschickter Software, auf die Dioden verzichten.

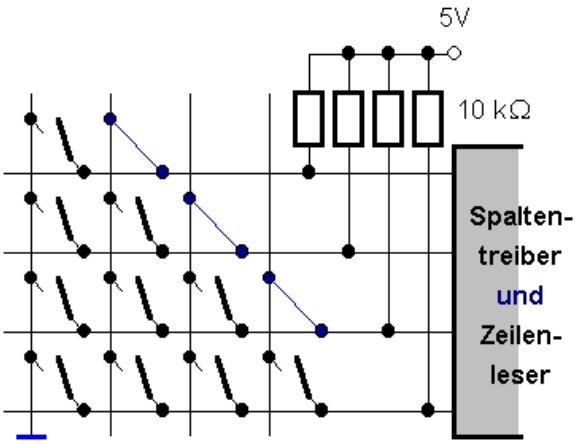


Abbildung 132: Tastenmatrix - Keine Dioden

Hiermit ist keine Erkennung zweier Tasten möglich, Schlafmodus sowieso nicht. (Bestenfalls für die masseseitigen Tasten.) Es ist zwischen jeder Kombination zweier Portpins sowie nach Masse genau eine Taste.

Auf n Portpins kommen n*(n+1)/2 Tasten, hier sind es 10 Tasten.

8.19.1.7 Tasten, eingebettet in eine Zeitmultiplexsteuerung

Wird am Mikrocontroller eine LED-Anzeige im Zeitmultiplexbetrieb angeschlossen, empfiehlt sich die Tastenabfrage über einen oder mehrere Pseudo-Stellentreiber. Einmal pro Auffrischzyklus (nicht weniger, sonst flimmert's) werden die LED-Zeilentreiber auf Eingabe geschaltet und der Pseudo-Stellentreiber aktiviert. Alles andere ist dann wie in o. g. Grundschaltung, mit meist stark rechteckiger Auslegung.

8.19.1.8 Tasten mit gestuften Widerständen am A/D-Wandler

Diese Möglichkeit kommt bei kurzen Leitungsverbindungen in Betracht, weil sie enorm Verkabelungsaufwand im Gerät spart.

Um nicht lauter verschiedene Widerstände verwenden zu müssen, kann man eine Kette aus lauter gleichen Widerständen bilden und jede Anzapfung mit einem Taster nach Masse versehen. Am A/D-Wandler wird ein (etwa) Konstantstrom eingespeist.

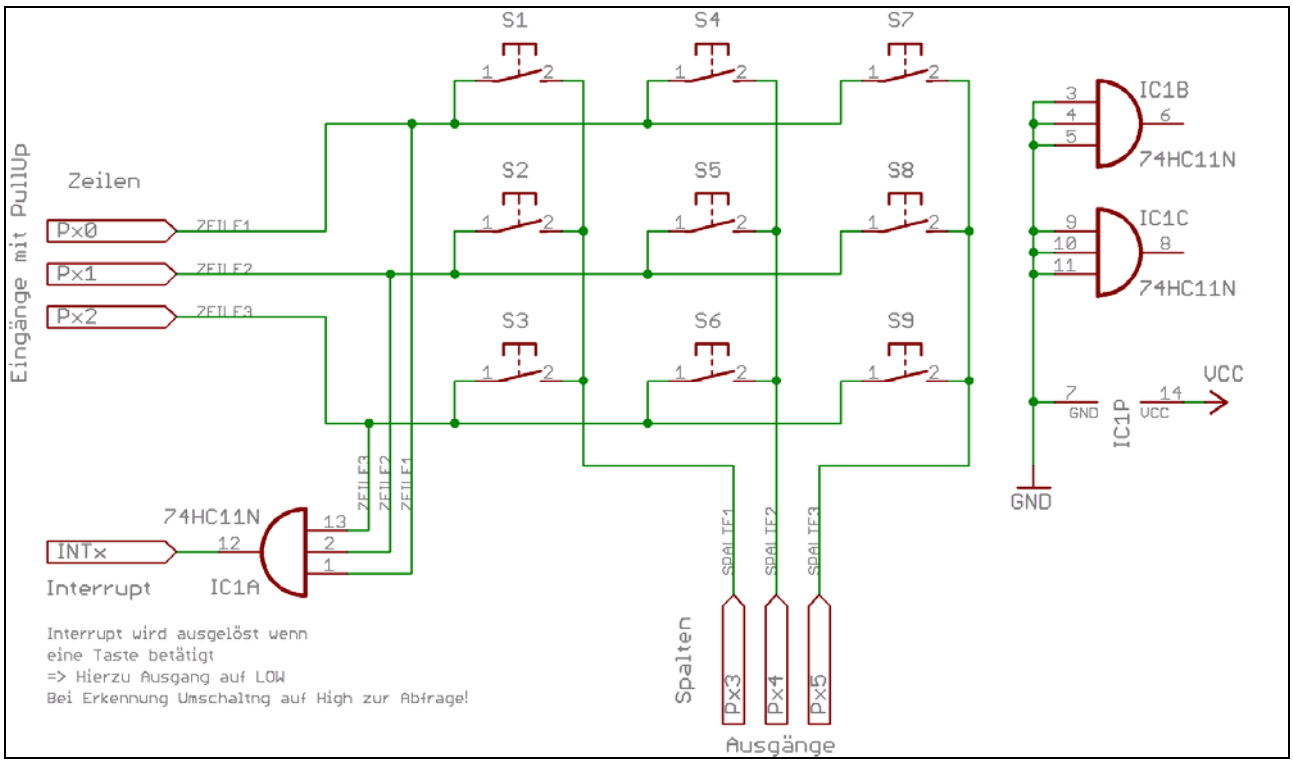
Logisch, dass der Einschaltwiderstand der Taster wesentlich kleiner sein sollte als der Einzelwiderstand der Kette. Diese Möglichkeit verlässt jedoch den Rahmen der Tasten-Matrix.

8.19.2 Realisierung Tastenmatrix im Projekt

Prinzipielle Umsetzung:

Die Tastatur im Projekt wird als 3x3 Tastenmatrix ausgeführt. Auf diesem Weg werden IO-Leitungen des Controllers gespart.

Die Tasten sind spalten- und zweilenweise zusammengafasst. Unter Verwendung der ATmega internen PullUp-Widerstände erhält man das folgende Anschlussschema einer 3x3 Tastaturmatrix.



Schaltbild 44: 3x3 Tastaturmatrix Grundprinzip

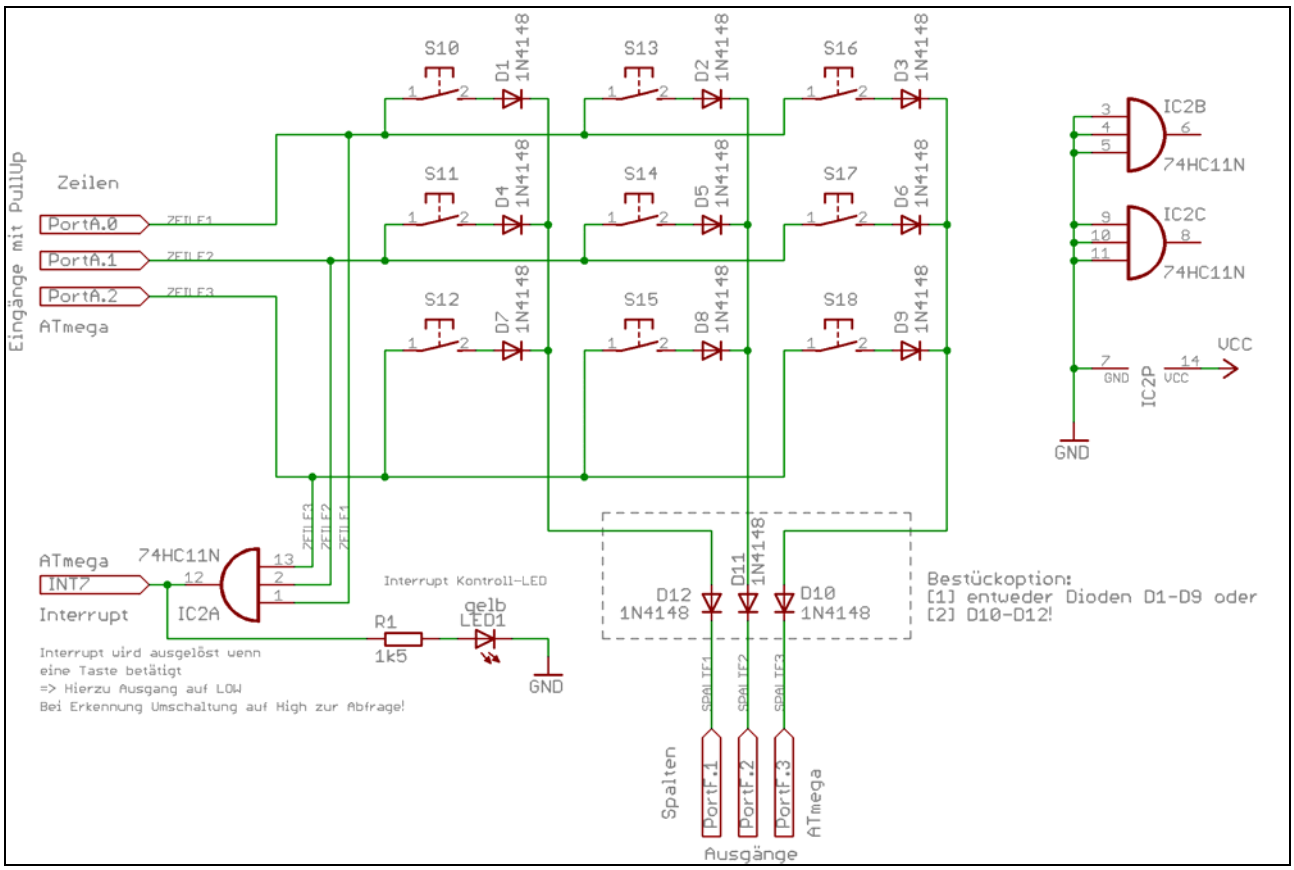
Bauteile:

Stückliste 3x3 Tastaturmatrix			
Sonstiges		*Halbleiter*	
S1-S9	Taster	IC1	3-fach AND Gatter 74HC11N

Tabelle 125: Stückliste 3x3 Tastaturmatrix Grundprinzip

Umsetzung nach Erstinbetriebnahme (optiert):

Mit der Tastenmatrix soll es möglich sein, mehrere Tasten gleichzeitig zu betätigen. Hierzu sind Entkopplungsdioden vorzusehen. Das folgende Schaltbild zeigt die optimierte Variante mit den Entkopplungsdioden. Hierbei ist eine Bestückvariante vorgesehen.



Schaltbild 45: 3x3 Tastaturmatrix Umsetzungsvariante

Bauteile:

Stückliste 3x3 Tastaturmatrix			
Sonstiges		***Halbleiter***	
S10-S19	Taster	IC2	3-fach AND Gatter 74HC11N
Widerstände		D1-D12	Universal-Diode 1N4148
R1	Metallschicht-Widerstand 1k5 Ω	LED1	LED low current 3mm gelb

Tabelle 126: Stückliste 3x3 Tastaturmatrix Umsetzungsvariante

Themenbezogene LED's:

LED / Funktion / Beschreibung	*Abkürzung*		*Farbe*	
Taste 1	LED1_SW1	(auf)	gelb	*[LED4]*
Taste 2	LED2_SW2	(zu)	gelb	*[LED5]*
Taste 3	LED3_SW3	(heller)	gelb	*[LED6]*
Taste 4	LED4_SW4	(dunkler)	gelb	*[LED7]*
Taste 5	LED5_SW5	(left)	gelb	*[LED8]*
Taste 6	LED6_SW6	(right)	gelb	*[LED9]*
Taste 7	LED7_SW7	(up)	gelb	*[LED10]*
Taste 8	LED8_SW8	(down)	gelb	*[LED11]*
Taste 9	LED9_SW9	(enter)	gelb	*[LED12]*
Status-LED für Tasteninterrupt	INT7		gelb	*[LED31]*

Tabelle 127: Themenbezogene LED's für Tastenmatrix

Ressourcenzuordnung zum ATmega128:

Nummer	*Schaltbild*		*Ressource ATmega128*		
1	SW_IN_1	[SW_IN_1]	PortA.0	[PA0]	(GPIO)
2	SW_IN_2	[SW_IN_2]	PortA.1	[PA1]	(GPIO)
3	SW_IN_3	[SW_IN_3]	PortA.2	[PA2]	(GPIO)
4	SW_OUT_1	[SW_OUT_1]	PortF.1	[PF1]	(GPIO)
5	SW_OUT_2	[SW_OUT_2]	PortF.2	[PF2]	(GPIO)
6	SW_OUT_3	[SW_OUT_3]	PortF.3	[PF3]	(GPIO)
7	KEY_INT	[INT7]	PortE.7	[INT7]	(Interrupt 7)

Tabelle 128: Ressourcenzuordnung ATmega128 für Tastaturmatrix

Die Pins Px0 bis Px2 werden als Eingänge mit PullUp-Widerstand konfiguriert, während die Pins Px3 bis Px5 jeweils die abzufragenden Spalten auf LOW legt.

Die Abfrage dieser Tasttaur teilt sich on mehrere Abfragen von Spalten auf. Um die Rechenlast der zyklische Abfrage für den Atmega zu senken wurden die Zeilenspalten über ein AND-Gatter geführt welches mit einem INT des ATmegas verbunden ist.

Die Ausgänge sind hierzu auf LOW konfiguriert. Wir nun eine Taste beätigt so wird ein Interrupt ausgelöst. In der Interrupt-Service-Routine werden die Ausgänge zunächste auf HIGH umkonfiguriert umd nach und nach die Spalten zru Abfrage auf LOW zu legen.

Achtung: Nicht verwendete Eingänge eines AND-Gatters sind mit GND zu beschalten.

Das folgende Listing (© by Claus Kühnle) aus dem Buch „Programmieren der AVR RISC Mikrocontroller“ zeugt in der Funktion query_matrix() zwei verschachtelte Schleifen zur Abfrage einer 3x4 Tastaturmatrix.

8.19.3 Mechanik der Tasten und Festlegungen

Auswahl der einzusetzenden Taster:



Eingabetaster mit fühl- und hörbarem Schaltpunkt. (Impulstaster, Digitast)
Verpackungsgewicht: 0.0022kg
RoHS: konform

Eingabetaster SW, Schaltspannung: 24V,fürLED (schwarz)
Reichelt Bestellnummer: *DTL 2 SW*

Eingabetaster rot, Schaltspannung: 24V,fürLED (rot)
Reichelt Bestellnummer: *DTL 2 RT*

Eingabetaster blau, Schaltspannung: 24V,fürLED (blau)
Reichelt Bestellnummer: *DTL 2 BL*

Eingabetaster, Schaltspannung: 24V, für led, ge (gelb)
Reichelt Bestellnummer: *DTL 2 GE*

Eingabetaster, Schaltspannung: 24V, für led, gn (grün)
Reichelt Bestellnummer: *DTL 2 GN*

Tabelle 129: Eingabetaster

Für die Steuerung wird die folgende Farbwahl der Tasten festgelegt:

Funktion		*Farbe / Taster*
Beschattung auf/zu/Stopp	(Taste_1)	2 x DTL 2 SW
Licht an/aus/heller/dunkler	(Taste_2)	2 x DTL 2 GE
Taste UP / LEFT / DOWN / RIGHT	(Taste_3)	4 x DTL 2 BL

Taste ENTER	(Taste_4)	1 x DRL 2 RT

Tabelle 130: Asuwahl Eingabetaster

Hinweise zum Einbau der Eingabetasten:

Die Eingabetasten werden auf der oberen PCB2 auf der Bestückungsseite bestückt. Für die Gaheäuseabdeckung sind Aussparungen vorgesehen durch diese di Tasten hidurch ragen. Aufgrund der Tastengröße kann die Tasten- und LED-Elektronik auf der PCB2 Seite ebenfalls auf der Bestückseite bestückt werden. Bei ca. 2mm Deckelstärke ist ca. 8mm Platz für die Bestückung vorhanden.

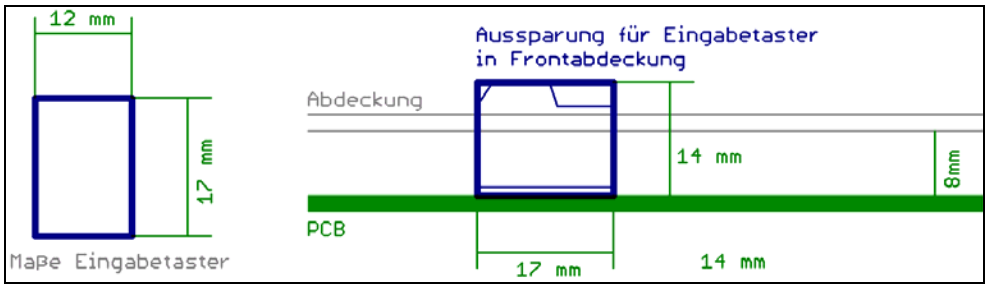


Abbildung 133: Tastendimensionen und Platbedarf

8.19.4 Prototyp Tastatur-Matrix

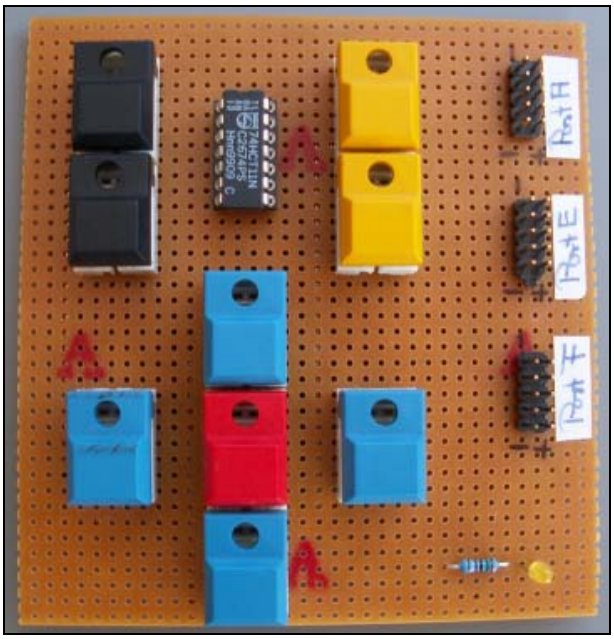


Abbildung 134: Prototyp der 3x3 Tastematrix

8.19.5 BASCOM Treibersoftware für Tastaturmatrix

```
'################################################################################
' TASTENMATRIX.BAS:                                          Stand 18.03.2008
' -----------------                                          (C) Markus Fulde
'
' Testprogramm zur Inbetriebnahme einer 3x3 Tastenmatrix.
'
' Das Konzept sieht das Einlesen von 9 Tasten über eine 3x3 Tastenmatrix am
' einem ATmega vor. Der Vorteil einer Tastematrix besteht in der Ressourcen-
' schonenden Umsetzung da nicht für alle Tasten ein eigener IO-Pin benötigt
' wird.
```

```
' Bei der hier vorliegenden Umsetzung wird zusätzlich eine Interrupt-Steuerung
' eingesetzt welche es ermöglicht, die Tastatur erst dann abzufragen wenn
' wirklich ein Eingabeereignis über Tastatur vorliegt. Damit ist es möglich
' auch Rechenzeit zu sparen.
' Eine BASCOM-Lösungen sehen den Einsatz der Funktion DEBOUNCE zur Abfrage der
' Zeilen vor. Da dadurc die Gleichzeitige Drücken von mehreren Tasten innerhalb
' einer Zeile nicht erkannt werden kann, werden die PINs in der hier entstan-
' denen Lösung einzeln Abgefragt.
' Weiter haben die einzelnen Spalten und Zeilen von Ihrer Wertigkeit die Reihe
' 1-2-4. Dadurch wird es möglich, ggf. aus einem Tasten-Gesamtwert
' (Key_query_matrix) die einzelnen betätigten Tasten herauszurechnen.
'
' Das Tastaturkonzept soll in einer Wintergartensteuerung eingesetzt werden
' mit der auch eine komplette Menüsteuerung realisiert wird. Die zugrunde
' gelegte Intension ist so wenig wie möglich Rechenzeit in die Tastaturabfrage
' einzusetzen da die Hauptaufgaben der Steuerung die eigentlichen Steuerfunk-
' tionen sind.
'
' Die Testroutinen und die Auswertung der Tastatur, die Wahl der Reihen und
' Spalten ist bereits auf das Gesamtkonzept der Wintergartensteuerung zuge-
' schnitten. Das zugrundegelegte Konzept einer Matrixtastatur wird aber dennoch
' deutlich.
'
' Die Inbetriebnahme erfolgt mit dem Eva-Board STK500 + STK501.
' Die notwendigen Routinen für die Tastatur sind als solche mit KEY gekenn-
' zeichnet.
'
'################################################################################


'--------------------------------------------------------------------------------
' Compilerinstruktionen und Compilerdirektiven
'--------------------------------------------------------------------------------
$regfile = "m128def.dat"
$crystal = 16000000
$hwstack = 32
$swstack = 32
$framesize = 60

$baud = 38400

'--------------------------------------------------------------------------------
' Definition von Ressourcen
'--------------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========
Key_zeile_1_pin Alias Pina.0                          ' PortA.0 ist Zeile 1
Key_zeile_2_pin Alias Pina.1                          ' PortA.1 ist Zeile 2
Key_zeile_3_pin Alias Pina.2                          ' PortA.2 ist Zeile 3

Key_zeile_1_port Alias Porta.0                        ' PortA.0 ist Zeile 1
Key_zeile_2_port Alias Porta.1                        ' PortA.1 ist Zeile 2
Key_zeile_3_port Alias Porta.2                        ' PortA.2 ist Zeile 3

Key_spalte_1_pin Alias Pinf.1                         ' PortF.1 ist Spalte 1
Key_spalte_2_pin Alias Pinf.2                         ' PortF.2 ist Spalte 2
Key_spalte_3_pin Alias Pinf.3                         ' PortF.3 ist Spalte 3

Key_spalte_1_port Alias Portf.1                       ' PortF.1 ist Spalte 1
Key_spalte_2_port Alias Portf.2                       ' PortF.2 ist Spalte 2
Key_spalte_3_port Alias Portf.3                       ' PortF.3 ist Spalte 3

External_interrupt_flag_register Alias Eifr


'========== Für Testumgebung  ==========
' LED's zum Test ob BASCOM-ADC-Routine andere PIN's von PortF beeinflusst
Led_1 Alias Portc.0
Led_2 Alias Portc.1
Led_3 Alias Portc.2
Led_4 Alias Portc.3
Led_5 Alias Portc.4
Led_6 Alias Portc.5
Led_7 Alias Portc.6
Led_8 Alias Portc.7

'--------------------------------------------------------------------------------
' Definition von Variablen, Konstaten und Datentypen
'--------------------------------------------------------------------------------
```

```
'========== Für Projekt ConvCtrl ==========
' ----- Konstanten -----
Const Key_testmodus = 0                                    ' Flag für Testmodus

Const Key_zeilen_pullup = &B00000111                       ' PullUp Definition für PORTA
'                              |||
'                              ||+--- PullUp für PortA.0 Zeile 1
'                              |+---- PullUp für PortA.1 Zeile 2
'                              +----- PullUp für PortA.2 Zeile 3

Const Key_spalten_ddrf = &B00001110                        ' Definition für Data Direction
'                            |||
'                            ||+--- DDR für PortF.1 Spalte 1
'                            |+---- DDR für PortF.2 Spalte 2
'                            +----- DDR für PortF.3 Spalte 3

Const Key_interrupt_mask = &B0111111                       ' Interrupt Mask für Bit7

Const True = &H01                                          ' Boolsche Variable TRUE
Const False = &H00                                         ' Boolsche Variable FALSE

Const Key_waiting = 50                                     ' Konstante in ms für Entprellung

' ----- Globale Variablen -----
Dim Key_keypressed As Byte                                 ' Globale Variable zur Tastensignalis-
ierung
Key_keypressed = False                                     ' Vorinitialisierung

Dim Key_taste As Byte                                      ' Variable enthält Tastennummer
Dim Key_spalte As Byte                                     ' Variable für Spalten
Dim Key_spalten_index As Byte                              ' Variable für Spaltenindex Tastenre-
chung
Dim Key_zeile As Byte                                      ' Variable für Zeilen

Dim Temp_byte_1 As Byte                                    ' Temporäre Byte-Variable


'========== Für Testumgebung ==========

' ----- Konstanten -----
Const Main_testmodus = 1                                   ' Flag für Testmodus
Const Wait_for_start = 3                                   ' Wartezeit vor Eintritt in LOOP

' In diesem Testprogramm werden zwei mögliche Lösung untersucht und vorgestellt:
' 1. Lösung: Tastensteuerung via Interrupt, Auswertung aber innerhalb Hauptschleife
'            (Variante1)
' 2. Lösung: Auswertung der Taste gleich in Interruptroutine und Bereitstellung
'            Tastencode über globale Variable
'            (Variante2)
Const Variante1 = 1                                        ' Achtung: entweder oder
Const Variante2 = 0                                        ' Achtung: entweder oder

' ----- Globale Variablen -----
Dim Test_int_counter As Word                               ' Testcounter für Tasteninterrupts


'-------------------------------------------------------------------------------
' Prototyping
'-------------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========
Declare Function Key_query_matrix() As Byte

'========== Für Testumgebung ==========


'-------------------------------------------------------------------------------
' Konfiguration und Basiseinstellungen (Projekt und Testumgebung)
'-------------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========

' ----- Konfiguration der Tastenmatrix -----
' Konfiguration Port A Zeilen (Reihen)
Config Key_zeile_1_pin = Input                             ' Zeile 1 ist Eingang
Config Key_zeile_2_pin = Input                             ' Zeile 2 ist Eingang
Config Key_zeile_3_pin = Input                             ' Zeile 3 ist Eingang
Porta = Key_zeilen_pullup                                  ' PullUp aktiv für PA0-PA2

' Konfiguration Port F Spalten
```

```
Config Key_spalte_1_pin = Output                              ' Spalte 1 ist Ausgang
Config Key_spalte_2_pin = Output                              ' Spalte 2 ist Ausgang
Config Key_spalte_3_pin = Output                              ' Spalte 3 ist Ausgang

' Basis-Initialisierung der Spalten-PINs (alle Spalten auf LOW)
' Dies ist nach Systemstart normalerweise Default, aber man weis ja nie :-)
Reset Key_spalte_1_port
Reset Key_spalte_2_port
Reset Key_spalte_3_port

' Alternative Konfiguration für die Spalten
' Ddrf = Key_spalten_ddrf                                     ' Alle Spalten auf einen Schlag
' Nachteil dieser Lösung ist, dass gesamtes Register geladen wird. Im Gesamtprojekt
' haben aber die anderen BIT's andere Aufgaben und Bedeutung. Daher explizites Setzen
' gewählt.


' ----- Konfiguration der Interrupts für die Tastenabfrage -----
On Int7 Key_isr_routine                                       ' Definition ISR
Test_int_counter = &H0000                                     ' Testcounter vorinitialisieren
Enable Int7                                                   ' INT7 freigeben
Config Int7 = Falling                                         ' INT bei falling Edge
Enable Interrupts                                             ' Alle Interrupts aktivieren


'========== Für Testumgebung ==========
' LED-Ausgäng konfigurieren (only for Test with STK500)
Config Portc = Output                                         ' Kompletter PortC ist Ausgang
Portc = &HFF                                                  ' Alle LED aus

' Wait_for_start warten bis Programm beginnt (ermöglich Umschaltung auf Terminalprogramm)
Wait Wait_for_start                                           ' For Test only

' Ausgabe Trace-Text
#if Main_testmodus
   Print "** Enter MainLoop **"
   Print
#endif


'#############################################################################
' Hauptprogramm
'#############################################################################
'Hier ist die Programmhauptschleife
Do
   ' Variante 1:
   ' Detektion der Taste über Interrupt, Signalisierung über globale Variable
   ' Auswertung der Tastenmatrix ausserhalb Interrupt und übergabe Tastencode
   ' über globale Variable.
   #if Variante1
      If Key_keypressed = True Then
         ' 50ms Zeit soll das Betätigen von zwei Tasten gleichzeitig vereinfachen
         ' -> wird benötigt für "Beschattung stopp" und "Licht aus"
         ' Außerdem dient die Zeit zum Entprellen der Tasten!!
         Waitms Key_waiting
         ' Tastenmatrix auslesen (Achtung, gleichzeigiges Betätigen innerhalb Reihe problematisch)
         Key_taste = Key_query_matrix()                       ' Abfrage der Tastaturmatrix
         ' Auswertung der Tastatur und Ausgabe
         If Key_taste <> 0 Then
            Print "Taste " ; Key_taste                       ' Anzeige der gelesenen Taste
         End If

         ' globales Flag für Signalisierung zurücksetzen da Auswertung jetzt durchgeführt
         Key_keypressed = False

         ' Nach Auswertung Tastatur für erneute Eingabe wieder freigeben
         Gosub Key_enable_keyinterrupt

      End If
   #endif

   ' Variante 2:
   ' Detektion der Taste über Interrupt, Auswertung der Tastenmatrix innerhalb
   ' Interrupt, Signalisierung und Übergabe Tastencode über globale Variablen
   #if Variante2
      ' Auswertung der Tastatur und Ausgabe
      If Key_keypressed = True Then
         If Key_taste <> 0 Then
            Print "Taste " ; Key_taste                       ' Anzeige der gelesenen Taste
         End If
```

```
        ' globales Flag für Signalisierung zurücksetzen da Auswertung jetzt durchgeführt
        Key_keypressed = False

        ' Nach Auswertung Tastatur für erneute Eingabe wieder freigeben
        Gosub Key_enable_keyinterrupt
      End If
   #endif

Loop
'## End Hauptprogramm ###########################################################


'*******************************************************************************
' Interruptroutinen
'*******************************************************************************
'-------------------------------------------------------------------------------
' Interrupt-Service-Routine (INT7): Key_isr_routine
'
'-------------------------------------------------------------------------------
Key_isr_routine:
   ' INT7 sperren damit keine weitere Tasteneingabe mehr ausgelöst wird.
   Disable Int7
   ' globales Flag für Tasteeingabe setzen
   Key_keypressed = True
   ' Ggf. Trace Output schreiben
   #if Key_testmodus
      ' Interrupt-Counter erhöhen
      Incr Test_int_counter
      Print "* " ; Test_int_counter
      ' LED1 einschalten - for test only
      Reset Led_1
   #endif

   ' Für Variante 2 hier noch direkt gleich Tastenmatrix abfrage
   #if Variante2
      ' 50ms Zeit soll das Betätigen von zwei Tasten gleichzeitig vereinfachen
      ' -> wird benötigt für "Beschattung stopp" und "Licht aus"
      ' Außerdem dient die Zeit zum Entprellen der Tasten!!
      Waitms Key_waiting
      ' Tastenmatrix auslesen
      Key_taste = Key_query_matrix()                        ' Abfrage der Tastaturmatrix
   #endif

Return
'-- End Key_isr_routine ------------------------------------------------------


'*******************************************************************************
' Subroutinen
'*******************************************************************************

' gibt's keine

'*******************************************************************************
' Funktionen
'*******************************************************************************

'-------------------------------------------------------------------------------
' KEY - Funktion: Key_query_matrix()
' Funktion zur Abfrage der einzelnen Tasten innerhalb der Matrix.
' Die Abfrage erfolgt über zwei Schleifenkonstrukte welche die einzelnen Spal-
' ten und darin die einzelnen Zeilen einer Spalte auf eine gedrückte Taste
' absucht.
'
' Parameter:    keine
' Rückgabewert: Bytewert = Summe aller erkannten Tastencodes
'                          Tastencode ist wie folgt aufgebaut (Ty.x):
'                          T1.1 = 11     T1.2 = 12      T1.3 = 14
'                          T2.1 = 21     T2.2 = 22      T2.3 = 24
'                          T3.1 = 41     T3.2 = 42      T3.3 = 44
'-------------------------------------------------------------------------------
Function Key_query_matrix() As Byte

   ' Ggf. Trace Output schreiben
   #if Key_testmodus
      Print "Enter Key_query_matrix"
   #endif

   ' Taste NIX vorbelegen
```

```
   Key_taste = 0

   ' Spaltenabfragen (Spalte für Spalte durchklappern)
   For Key_spalte = 0 To 2

      ' ----- Einzelnes "abklappern" der Spalten -----
      ' Spalte 1 auf LOW und Spaltenindex für Tastencoderechung setzen
      If Key_spalte = 0 Then
         Reset Key_spalte_1_port
         Set Key_spalte_2_port
         Set Key_spalte_3_port
         Key_spalten_index = 1
      End If

      ' Spalte 2 auf LOW und Spaltenindex für Tastencoderechung setzen
      If Key_spalte = 1 Then
         Set Key_spalte_1_port
         Reset Key_spalte_2_port
         Set Key_spalte_3_port
         Key_spalten_index = 2
      End If

      ' Spalte 3 auf LOW und Spaltenindex für Tastencoderechung setzen
      If Key_spalte = 2 Then
         Set Key_spalte_1_port
         Set Key_spalte_2_port
         Reset Key_spalte_3_port
         Key_spalten_index = 4
      End If

     ' Zeilenabfragen (Zeile für Zeile durchklappern)
     ' In der Ursprünglichen Version wurde die Abfrage der Zeilen über den
     ' BASCOM-Befehl debounce realisiert. Das Problem hier ist, das Debounce
     ' wartet bis eine "Taste=Zeile" wieder losgelassen wurde. Damit ist die
     ' Abfrage von mehreren Tasten einer gleichen Zeile nicht möglich!
     ' Die Entprellung der Tasten erfolgt durch die Zeitkonstante Key_waiting
     For Key_zeile = 1 To 3
       Select Case Key_zeile
         ' Prüfung für Zeile 1 vornehmen und ggf. Tastencodes rechen
         Case 1 :
            ' Debounce Key_zeile_1_pin , 0 , Key_calc_taste , Sub
            If Key_zeile_1_pin = 0 Then
               Temp_byte_1 = Key_spalten_index + 10
               Key_taste = Key_taste + Temp_byte_1
            End If
         ' Prüfung für Zeile 2 vornehmen und ggf. Tastencodes rechen
         Case 2 :
            ' Debounce Key_zeile_2_pin , 0 , Key_calc_taste , Sub
            If Key_zeile_2_pin = 0 Then
               Temp_byte_1 = Key_spalten_index + 20
               Key_taste = Key_taste + Temp_byte_1
            End If
         ' Prüfung für Zeile 3 vornehmen und ggf. Tastencodes rechen
         Case 3 :
            ' Debounce Key_zeile_3_pin , 0 , Key_calc_taste , Sub
            If Key_zeile_3_pin = 0 Then
               Temp_byte_1 = Key_spalten_index + 40
               Key_taste = Key_taste + Temp_byte_1
            End If
       End Select
     Next Key_zeile
   Next Key_spalte

   ' Urspungszustand wie bei Init wieder herstellen damit Interruptsteuerung
   ' wieder möglich wird.
   Reset Key_spalte_1_port
   Reset Key_spalte_2_port
   Reset Key_spalte_3_port

   ' Tastencode an Rückgabewert übergeben
   Key_query_matrix = Key_taste

End Function
'-- End Key_Query_matrix ------------------------------------------------------


'*******************************************************************************
' GOSubroutinen
'*******************************************************************************
```

```
'-------------------------------------------------------------------------------
' KEY - Gosub-Routine: Test_enable_keyinterrupt
' Routine aktiviert den INT7 für die Tastaturereigniserfassung erneut. Zuvor
' wird jedoch das Bit für INT7 im External-Interrupt-Flag-Register gelöscht
' da bereits (durch prellen der Tasten) ein neuer Interrupt anstehen kann.
'-------------------------------------------------------------------------------
Key_enable_keyinterrupt:

   #if Key_testmodus
      Print "Enter Test_enable_keyinterrupt"
      Print "ORIG EIFR : " ; Bin(eifr)

      ' LED1 ausschalten - for test only
      Set Led_1
   #endif

   ' Dieses Bit wird gesetzt, wenn am INT7-Pin eine Interrupt-Kondition,
   ' entsprechend der Konfiguration, erkannt wird. Wenn das External  Enable
   ' Interrupt Flag gesetzt ist, wird die Interruptroutine angesprungen.
   ' Das Flag wird automatisch gelöscht , wenn die Interruptroutine beendet ist.
   ' Alternativ kann das Flag gelöscht werden , indem der Wert 1(!)
   ' eingeschrieben wird. ACHTUNG: Wert 1 !!!!!

   ' Lösche Flankeninterrupt INT7 wenn bereits mehrfach aufgetreten.
   Eifr.intf7 = 1

   #if Key_testmodus
      Print "RST EIFR  : " ; Bin(eifr)
   #endif

   Enable Int7

Return
'-- End Key_enable_keyinterrupt ------------------------------------------------


End                                                    'end program


'-------------------------------------------------------------------------------
' Definition von globalen Konstantenfeldern
'-------------------------------------------------------------------------------


'#################################### END ####################################
```

Software 12: Code für die 3x3 Tastenmatrix

8.20 Latched LED Matrix

8.20.1 Allgemeines zur LED-Matrix

Beim Design des Projekts – ürsprunglich auf ATmega32 geplant – wurde sehr schnell klar, dass eine Lösung benötigt wird welches es ermöglicht, die benötigten LED's Ressourcenschonend an den ATmega anzubinden.

Obwohl der ATmega128 einiges an PIN's und IO-Leitungen mehr bietet wird dennoch die Ressourcen schonende Umsetzung designed und verfolgt. Ziel ist es alle benötigten LED's mit so wenig Ressourcen wie möglich an den ATmega anzubinden.

Hierbei denkt mal als erstes an eine Schieberegister-Funktion in den die Bitmuster eingetacktet werden. Da dies jedoch einen höheren Anspruch an Timing und Programmierung stellt wird eine parallele Lösung verfolgt.

Zur Umsetzung des Projekts bietet sich der Baustein 74116 (54116) an. Der Baustein ist ein Dual 4-Bit Latch.

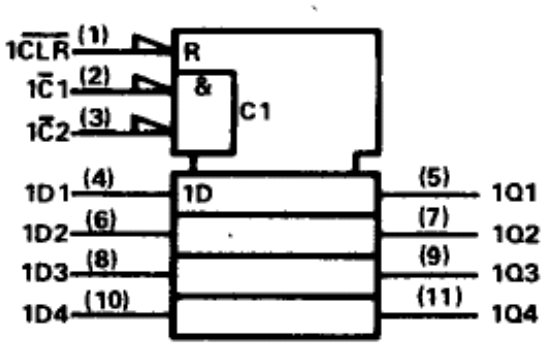


Abbildung 135: SN74116 Dual 4-Bit Latch

Alle LED's werden im System als 3mm low current LED ausgelegt!

Die folgenden LED's werden über das Latch (kaskadierte Latches) betrieben:

LED / Funktion / Beschreibung	*Abkürzung*	*Farbe*	
Taste 1	LED1_SW1 (auf)	gelb	*[LED4]*
Taste 2	LED2_SW2 (zu)	gelb	*[LED5]*
Taste 3	LED3_SW3 (heller)	gelb	*[LED6]*
Taste 4	LED4_SW4 (dunkler)	gelb	*[LED7]*
Taste 5	LED5_SW5 (left)	gelb	*[LED8]*
Taste 6	LED6_SW6 (right)	gelb	*[LED9]*
Taste 7	LED7_SW7 (up)	gelb	*[LED10]*
Taste 8	LED8_SW8 (down)	gelb	*[LED11]*
Taste 9	LED9_SW9 (enter)	gelb	*[LED12]*
Speicher voll	LED10_MEM_FULL	rot	*[LED13]*
Auffahrgrund Zeitsteuerung	LED11_OPEN_TIME	gelb	*[LED14]*
Auffahrgrund Licht	LED12_OPEN_LIGHT	gelb	*[LED15]*
Auffahrgrund Regen	LED13_OPEN_RAIN	rot	*[LED16]*
Auffahrgrund Wind	LED14_OPEN_WIND	rot	*[LED17]*
Zufahrgrund Zeitsteuerung	LED15_CLOSE_TIME	gelb	*[LED18]*
Zufahrgrund Temperatur	LED16_CLOSE_TEMP	gelb	*[LED19]*

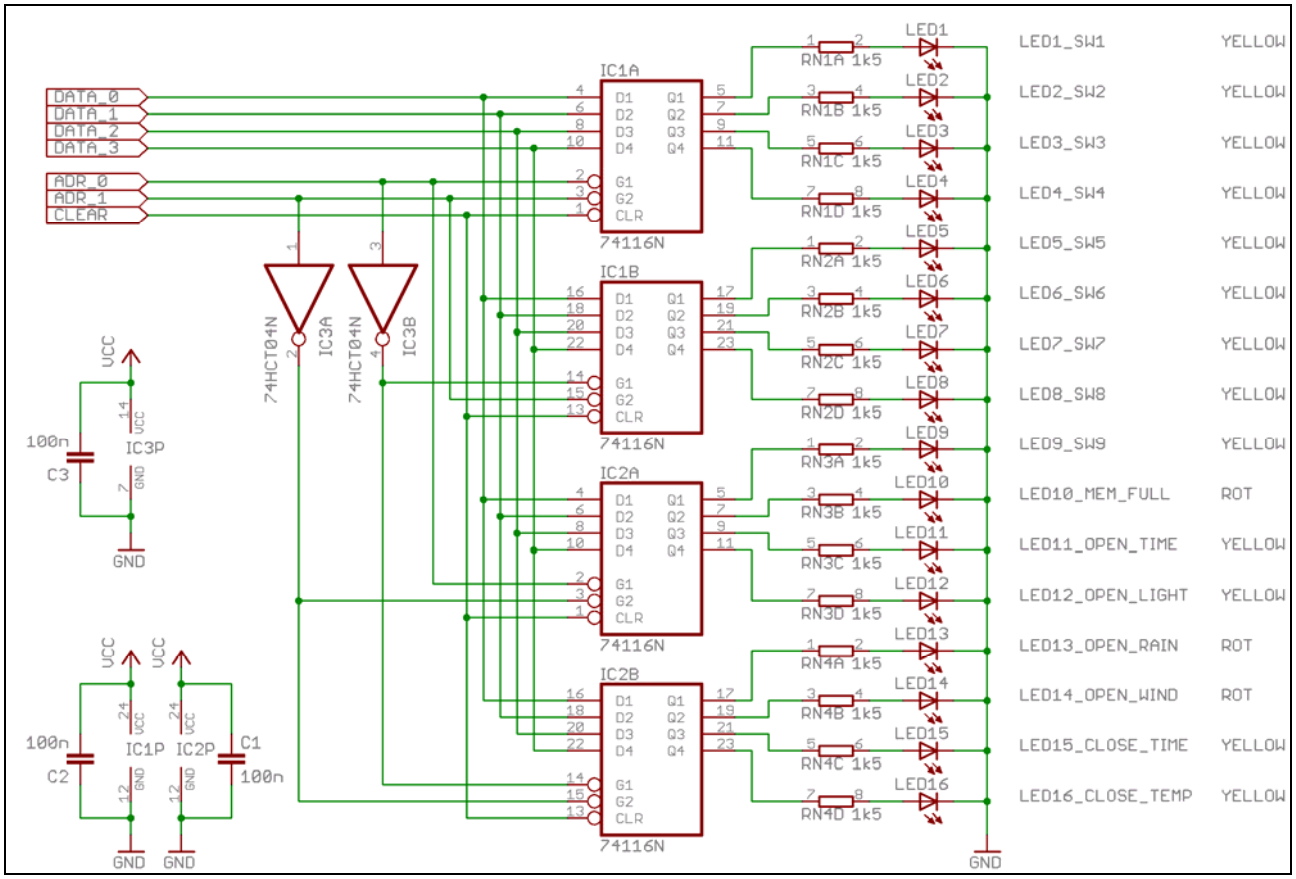
Tabelle 131: Latched LED's

Die folgenden LED's werden NICHT über das Latch (kaskadierte Latches) betrieben sondern hängen diskret an anderen Funktionen:

LED / Funktion / Beschreibung	*Abkürzung*	*Farbe*	
Alive LED	LED_ALIVE	grün	*[LED3]*
5V present	LED_5V	grün	*[LED1]*
12V present	LED_12V	grün	*[LED2]*
Relais Beschattung AUF	LED_REL_CONV_OPEN	gelb	*[LED21]*
Relais Beschattung ZU	LED_REL_CONV_CLOSE	gelb	*[LED22]*
Relais Licht (Versorgung Dimmer)	LED_REL_LIGHT	gelb	*[LED23]*
Relais Dauerlicht	LED_REL_CONLUM	gelb	*[LED24]*
Lüfter über SW	LED_FAN_SW	gelb	*[LED25]*
Lüfter über Dauerplus	LED_FAN_12V	gelb	*[LED26]*
Timer-Tick für Sekundentakt	LED_SEC_TICK	grün	*[LED27]*
Regen detektiert	LED_RAIN_DETECT	gelb	*[LED29]*
Heizung Regensensor aktiv	LED_RAIN_HEATER	gelb	*[LED20]*
Status-LED für Tasteninterrupt	INT7	gelb	*[LED31]*
Status-LED für Windimpuls	LED_WIND_PULS	gelb	*[LED28]*
DCF77 Impuls	LED_DCF77_TICK	gelb	*[LED30]*

Tabelle 132: None Latched LED's

Umsetzung der latched Steuerung:



Schaltbild 46: Latch LED Array

Bauteile:

Stückliste: Latched LED Array			
Widerstände		*Halbleiter*	
RN1, RN2, RN3, RN4	Widerstand-Netzwerk 1k5 Ω	IC1, IC2	Dual 4-Bit Latch SN74116N
		IC3	Inverter 74HCT04N
		LED1-9	3mm low current LED gelb
		LED11-12	3mm low current LED gelb
		LED15-16	3mm low current LED gelb
		LED10, LED13, LED14	3mm low current LED rot

Tabelle 133: Latched LED Array

Der Baustein SN74116N im DIL24 Gehäuse ist sehr schwierig zu bekommen. Bei der Internetrecherche hatte die Firma ELW noch einige dieser Bausteine. Hierz die folgenden Daten:

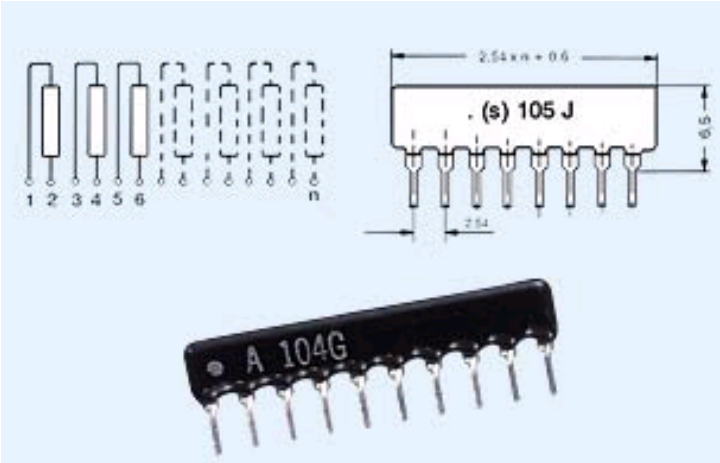
ELW Elektronik Handels GmbH
Am Kalkofen 10
61206 Wöllstadt / Germany
Gerichtsstand : HRB Friedberg Nr. 1655
Geschäftsführer Reinhold Mühlnickel
Umsatz – ID – Nr.: DE 154 318 581
Tel +49 6034 4411
Fax +49 6034 5739
Internet: http://www.elw-shop.de/Impressum.aspx

Ressourcenzuordnung zum ATmega128:

Nummer	*Schaltbild*		*Ressource ATmega128*		
1	DATA_0	[DATA_0]	PortA.3	[PA3]	(GPIO)
2	DATA_1	[DATA_1]	PortA.4	[PA4]	(GPIO)
3	DATA_2	[DATA_2]	PortA.5	[PA5]	(GPIO)
4	DATA_3	[DATA_3]	PortA.6	[PA6]	(GPIO)
5	ADR_0	[ADR_0]	PortA.7	[PA7]	(GPIO)
6	ADR_1	[ADR_1]	PortG.2	[PG2]	(GPIO)
7	CLEAR	[CLEAR]	PortG.4	[PG4]	(GPIO)

Tabelle 134: Ressourcenzuordnung ATmega128 für Tastaturmatrix

Widerstandsnetzwerk:



Typ: Widerstands-Netzwerk
Aufbau: 8 pins / 4 Widerstände
Spezifikation: Sternschaltung

Bestellnummer Reichelt: *SIL 8-4 1,5K*

Tabelle 135: Widerstands-Netzwerk

8.20.2 Prototyp LED-Matrix

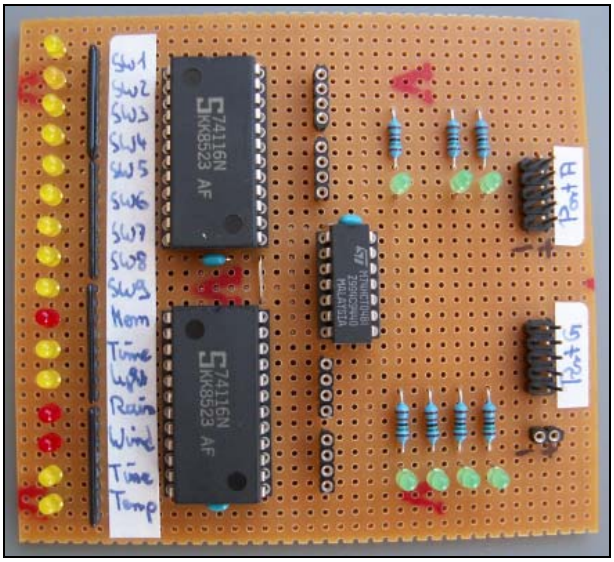


Abbildung 136: Prototyp der LED-Matrix

8.20.3 BASCOM Treibersoftware für Tastaturmatrix

```
'###############################################################################
' LED-Array.BAS:                                          Stand 29.03.2008
' --------------                                          (C) Markus Fulde
'
' Testprogramm zur Inbetriebnahme eines LED-Arrays mit 16 LED's.
'
' Das Konzept sieht eine Ressourcen schonende Ansteuerung von 16 LED's für
' bliebige Aufgaben vor. Hierbei sollen nur 4 PIN's zzgl. 3 PIN's für die
' Steuerleitungen verwendet werden.
'
' Die Ausgabe von Daten auf die LED's funktioniert über vier 4-Bit Latches die
' sequentiell nacheinander adressiert werden und die Daten übernehmen.
'
' Das LED-Array soll in einer Wintergartensteuerung eingesetzt werden
' und zur Ausgabe diverses Betriebszusände und Informationen dienen.
'
' Die Inbetriebnahme erfolgt mit dem Eva-Board STK500 + STK501.
' Die notwendigen Routinen für das LED-Array sind als solche mit LED gekenn-
' zeichnet.'
'
'###############################################################################


'-------------------------------------------------------------------------------
' Compilerinstruktionen und Compilerdirektiven
'-------------------------------------------------------------------------------
$regfile = "m128def.dat"
$crystal = 16000000
$hwstack = 32
$swstack = 32
$framesize = 30

$baud = 38400

'-------------------------------------------------------------------------------
' Definition von Ressourcen
'-------------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========

' ----- Port Input Pin Adress für Datenbits -----
Led_bit0_pin Alias Pina.3                           ' PortA.3 ist Daten-Bit0
Led_bit1_pin Alias Pina.4                           ' PortA.4 ist Daten-Bit1
Led_bit2_pin Alias Pina.5                           ' PortA.5 ist Daten-Bit2
Led_bit3_pin Alias Pina.6                           ' PortA.6 ist Daten-Bit3

' ----- Port Data Register für Datenbits -----
Led_bit0_port Alias Porta.3                         ' PortA.3 ist Daten-Bit0
Led_bit1_port Alias Porta.4                         ' PortA.4 ist Daten-Bit1
Led_bit2_port Alias Porta.5                         ' PortA.5 ist Daten-Bit2
Led_bit3_port Alias Porta.6                         ' PortA.6 ist Daten-Bit3

' ----- Port Input Pin Adress für Adressbits -----
Led_adr0_pin Alias Pina.7                           ' PortA.7 ist Adressen-Bit 0
Led_adr1_pin Alias Ping.2                           ' PortG.2 ist Adressen-Bit 1

' ----- Port Data Register für Adressbits -----
Led_adr0_port Alias Porta.7                         ' PortA.7 ist Adressen-Bit 0
Led_adr1_port Alias Portg.2                         ' PortG.2 ist Adressen-Bit 1

' Bemerkung:
' Durch die Verteilung der Adressen auf zwei Ports und damit nicht zusammenhängende
' Bits kommt es bei der Umschaltung der Adresse &B01 nach &B10 zu "Flackereffekten"
' bei denen entweder Adresse &B00 oder &B11 betroffen sind. Es ist im Einzelfall
' zu enscheiden ob zuerst Bit 0 auf LOW oder Bit 1 auf HIGH gezogen wird.
' Dieser Effekt ist nur dadurch zu vermeiden, indem beide Adressen in einem Port
' zusammengefasst werden. Der Zugriff darf dann nicht mit SET oder RESET erfolgen,
' sondern durch direkte Port-Ausgaben bei denen die 2-Bit-Adresse direkt gesetzt wird.

' ----- Port Input Pin Adress für Latch Clear -----
Led_clr_pin Alias Ping.4                            ' PortG.4 ist Latch Clear

' ----- Port Data Register für Latch Clear -----
Led_clr_port Alias Portg.4                          ' PortG.4 ist Latch Clear

' ----- Alias Namen für einzelne LED's im LED-Array -----
Led_01_sw1 Alias Led_array.0                        ' Bit 0
```

```
Led_02_sw2 Alias Led_array.1                                ' Bit 1
Led_03_sw3 Alias Led_array.2                                ' Bit 2
Led_04_sw4 Alias Led_array.3                                ' Bit 3
Led_05_sw5 Alias Led_array.4                                ' Bit 4
Led_06_sw6 Alias Led_array.5                                ' Bit 5
Led_07_sw7 Alias Led_array.6                                ' Bit 6
Led_08_sw8 Alias Led_array.7                                ' Bit 7
Led_09_sw9 Alias Led_array.8                                ' Bit 8
Led_10_mem_full Alias Led_array.9                           ' Bit 9
Led_11_open_time Alias Led_array.10                         ' Bit 10
Led_12_open_light Alias Led_array.11                        ' Bit 11
Led_13_open_rain Alias Led_array.12                         ' Bit 12
Led_14_open_wind Alias Led_array.13                         ' Bit 13
Led_15_close_time Alias Led_array.14                        ' Bit 14
Led_16_close_temp Alias Led_array.15                        ' Bit 15


'========== Für Testumgebung  ==========

' ----- Alias Namen für LED-Test-Ausgabe -----
Led_1 Alias Portc.0
Led_2 Alias Portc.1
Led_3 Alias Portc.2
Led_4 Alias Portc.3
Led_5 Alias Portc.4
Led_6 Alias Portc.5
Led_7 Alias Portc.6
Led_8 Alias Portc.7

' ----- Alias Namen für Tastaturabfragen -----
Taste_1 Alias Pind.0
Taste_2 Alias Pind.1
Taste_3 Alias Pind.2
Taste_4 Alias Pind.3
Taste_5 Alias Pind.4
Taste_6 Alias Pind.5
Taste_7 Alias Pind.6
Taste_8 Alias Pind.7


'-----------------------------------------------------------------------------
' Definition von Variablen, Konstaten und Datentypen
'-----------------------------------------------------------------------------

' ----- Konstanten -----
'========== Für Projekt ConvCtrl ==========
Const Led_testmodus = 1                                     ' Flag für Testmodus

'========== Für Testumgebung ==========
Const Main_testmodus = 1                                    ' Flag für Testmodus
Const Wait_for_start = 3                                    ' Wartezeit vor Eintritt in LOOP

' ----- Variablen -----
Dim Led_array As Word                                       ' Globale Variable für LED's
' Der Zugriff auf das Array erfolgt Bitweise. Übertragen wird das gesamte Word
' in das Latch!

Dim Temp_byte_1 As Byte                                     ' globale temporäre Variable
Dim Temp_byte_2 As Byte                                     ' globale temporäre Variable
Dim Temp_byte_3 As Byte                                     ' globale temporäre Variable

Dim Temp_word_1 As Word                                     ' globale temporäre Variable


'-----------------------------------------------------------------------------
' Prototyping
'-----------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========


'========== Für Testumgebung ==========


'-----------------------------------------------------------------------------
' Konfiguration und Basiseinstellungen (Projekt und Testumgebung)
'-----------------------------------------------------------------------------

'========== Für Projekt ConvCtrl ==========
```

```
' Konfiguration der Ausgänge für Datenbits, Adressbits und Latch Clear
Config Led_bit0_pin = Output                                ' Daten-Bit 0 ist Ausgang
Config Led_bit1_pin = Output                                ' Daten-Bit 1 ist Ausgang
Config Led_bit2_pin = Output                                ' Daten-Bit 2 ist Ausgang
Config Led_bit3_pin = Output                                ' Daten-Bit 3 ist Ausgang

Config Led_adr0_pin = Output                                ' Adress-Bit 0 ist Ausgang
Config Led_adr1_pin = Output                                ' Adress-Bit 1 ist Ausgang

Config Led_clr_pin = Output                                 ' Latch Clear ist Ausgang

' Basis-Initialisierung aller Signalleitungen auf LOW
' Dies ist nach Systemstart normalerweise Default, aber man weis ja nie :-)
Reset Led_bit0_port
Reset Led_bit1_port
Reset Led_bit2_port
Reset Led_bit3_port

Reset Led_adr0_port
Reset Led_adr1_port

Reset Led_clr_port

' Funktionstabelle des Latch:
' +-------------------------------------+
' | Inputs:     ENABLE          | OUTPUT |
' |    _____    __  __          |        |
' |    CLEAR    C1  C2    DATA  |  Q     |
' |=====================================|
' |      H      L   L       L   |  L     |
' |      H      L   L       H   |  H     |
' |      H      x   H       x   |  Qo    |
' |      H      H   x       x   |  Qo    |
' |      L      x   x       x   |  L     |
' +-------------------------------------+


'========== Für Testumgebung ==========
' LED-Ausgäng konfigurieren (only for Test with STK500)
Config Portc = Output                                       ' Kompletter PortC ist Ausgang
Portc = &HFF                                                ' Alle LED aus

' Wait_for_start warten bis Programm beginnt (ermöglich Umschaltung auf Terminalprogramm)
Wait Wait_for_start                                         ' For Test only
Reset Led_1                                                 ' Led 1 für Statusanzeig an

' Ausgabe Trace-Text
#if Main_testmodus
   Print "** Enter MainLoop **"
   Print
#endif


'#############################################################################
' Hauptprogramm
'#############################################################################
'Hier ist die Programmhauptschleife
Do
   ' Tastaturabfrage Taste 1
   Debounce Taste_1 , 0 , Led_clearlatch_set , Sub

   ' Tastaturabfrage Taste 2
   Debounce Taste_2 , 0 , Led_clearlatch_reset , Sub

   ' Tastaturabfrage Taste 3
   Debounce Taste_3 , 0 , Test_set_nibbles , Sub

   ' Tastaturabfrage Taste 4
   Debounce Taste_4 , 0 , Test_reset_nibbles , Sub

   ' Tastaturabfrage Taste 7
   Debounce Taste_7 , 0 , Test_update_array , Sub

   ' Tastaturabfrage Taste 8
   Debounce Taste_8 , 0 , Led_reset_array , Sub

Loop
'## End Hauptprogramm ######################################################
```

```
'********************************************************************************
' Interruptroutinen
'********************************************************************************

' gibt's keine

'********************************************************************************
' Subroutinen
'********************************************************************************

' gibt's keine

'********************************************************************************
' Funktionen
'********************************************************************************

' gibt's keine

'********************************************************************************
' GOSubroutinen
'********************************************************************************

'------------------------------------------------------------------------------
' LED-Gosub-Routine: Led_clearlatch_set
' Subroutine setzt das Clear Latch Bit am Ausgang. Clear Input überschreibt
' unabhängig von den Input-Pins die Output-PIN's mit Low.
'
' Bemerkung:
' Das Clear Latch Bit muss auf HIGH gesetzt werden, damit überhaupt mit dem
' Latch gearbeitet werden kann! Beim Übergang von LOW auf HIGH werden die
' angelegten Datenbits auf dem jeweilig adressierten Nibble sofort aktiv und
' Sichtbar. Ggf. sollte ein korrektes Datenmuster und eine Adresse vor dem
' "loslassen" des Clear Latch Bits erfolgen.
'------------------------------------------------------------------------------
Led_clearlatch_set:

   ' Ggf. Trace ausgeben
   #if Main_testmodus
      Print "Enter Led_clearlatch_set"
   #endif

   ' Clear Latch auf HIGH setzen
   Set Led_clr_port

Return
'-- End Led_clearlatch_set -----------------------------------------------------


'------------------------------------------------------------------------------
' LED-Gosub-Routine: Led_clearlatch_reset
' Subroutine setzt das Clear Latch Bit am Ausgang zurück. Clear Input
' überschreibt unabhängig von den Input-Pins die Output-PIN's mit Low.
'
' Bemerkung:
' Wird das Clear Latch auf LOW gesetzt so werden alle Ausgänge LOW. Die zuvor
' gespeicherten (gelatchten) Inhalte gehen verloren!
'------------------------------------------------------------------------------
Led_clearlatch_reset:

   ' Ggf. Trace ausgeben
   #if Main_testmodus
      Print "Enter Led_clearlatch_reset"
   #endif

   ' Clear Latch auf LOW setzen
   Reset Led_clr_port

Return
'-- End Led_clearlatch_reset ---------------------------------------------------


'------------------------------------------------------------------------------
' LED-Gosub-Routine: Led_reset_array
' Subroutine löscht alle Dateninhalte und damit alle LED, setzt den Datenbus
' auf &B0000 und setzt Adresse &B00 für das Low-Nibble. Das Clear Latch Bit
' wird nach der Löschung automatisch auf HIGH und damit auf Normalbetrieb
' geschaltet.
' Achtung: Alle gelatchten Dateninhalte gehen verloren
'------------------------------------------------------------------------------
Led_reset_array:
```

```
   ' Ggf. Trace ausgeben
   #if Main_testmodus
      Print "Enter Led_reset_array"
   #endif

   ' Clear Latch auf LOW setzen und damit alle Dateninhalte löschen
   Reset Led_clr_port

   ' Daten auf 0000
   Reset Led_bit0_port : Reset Led_bit1_port : Reset Led_bit2_port : Reset Led_bit3_port

   ' Adresse &B00 einstellen
   Reset Led_adr0_port : Reset Led_adr1_port

   ' Clear Latch auf HIGH setzen und damit Normalbetrieb freigeben
   Set Led_clr_port

Return
'-- End Led_reset_array ------------------------------------------------------


'--------------------------------------------------------------------------------
' LED-Gosub-Routine: Led_update_array
' Subroutine übertrag die globale Array-Variable Led_array in das Latch und
' gibt damit den aktuellen Inhalt aus.
'--------------------------------------------------------------------------------
Led_update_array:

   ' Ggf. Trace ausgeben
   #if Main_testmodus
      Print "Enter Led_update_array"
   #endif

   ' Löschen aller LED's, zurücksetzen Datenbits und Adresse auf &B00 stellen
   Gosub Led_reset_array

   ' LED-Array zur weiteren Verarbeitung in temp. Variable laden
   Temp_word_1 = Led_array

   ' Schleife über alle 4 Nibbles des LED-Arrays
   For Temp_byte_1 = 1 To 4
      ' Adresse für jeweiliges Nibble sinstellen. &B00 ist bereits durch Reset
      ' eingestellt, daher keine Veränderung
      Select Case Temp_byte_1
         Case 2:
            ' Setzte Adresse &B01
            Set Led_adr0_port
         Case 3:
            ' Setzte Adresse &B10
            Set Led_adr1_port : Reset Led_adr0_port
            ' Achtung: Reihenfolge Bits setzen für Adresse hat ggf. Einfluss auf
            ' niedrigeres oder höherwertiges Nibble da Adressengleichheit!!!
         Case 4:
            ' Setzte Adresse &B11
            Set Led_adr0_port
      End Select

      ' Extrahiere jeweiliges Nibble aus 16-Bit LED-Array
      Temp_byte_2 = Temp_word_1 And &H000F

      ' Verschiebe extrahiertes Nibble auf korrekte Datenposition für Port-Byte
      Shift Temp_byte_2 , Left , 3

      ' Hole aktuelles Portregister inkl. aktuell eingesteller Adresse und
      ' maskiere Datenbits für Nibble aus
      Temp_byte_3 = Porta And &B10000111
      ' Addiere oben extrahiertes Nibble in Portbyte zur Ausgabe udn Portbyte
      ' schreiben -> Ausgabe der jeweiligen LED's erfolgt
      Porta = Temp_byte_3 + Temp_byte_2

      ' Verschiebe nächstes Nibble in 16-Bit LED-Array auf Arbeitsposition
      Shift Temp_word_1 , Right , 4

   Next Temp_byte_1                                    ' Schleife ggf. wiederholen

Return
'-- End Led_update_array ------------------------------------------------------
```

```
'--------------------------------------------------------------------------
' Gosub-Routine: Test_set_nibbles
' Testsequenz zum sequentiellen Durchfahren der 16 LED über die 4 Adresse
' &B00, &B01, &B10 und &B11 mit den Bitmuster &B0101 und &B1111.
' Die Routine steuert 4 Nibbles.
' Achtung: Die 4 Datenbits sind immer für eines der 4 Adressen aktiv. Dies
' muss beim Verändern der Daten bzw. der Adressen berücksichtigt werden. Zur
' Sicherheit immer zuerst die Adressen umschalten um sicherzustellen das die
' restlichen Nibbles nicht beinflusst werden.
' Bei "- finish -" sollten alle LED's leuchten.
'--------------------------------------------------------------------------
Test_set_nibbles:

   ' Ggf. Trace ausgeben
   #if Main_testmodus
      Print "Enter Test_set_lownibble"
   #endif

   ' ----- Nibble 0 -----
   ' Adresse &B00 einstellen (Bits werden einzeln angesprochen)
   Reset Led_adr0_port : Reset Led_adr1_port
   ' Daten auf 0000 - alle LED's aus (Bits werden einzeln angesprochen)
   Reset Led_bit0_port : Reset Led_bit1_port : Reset Led_bit2_port : Reset Led_bit3_port
   ' Sequenz &B0101 -> &B1111 abfahren (Bits werden einzeln angesprochen)
   Set Led_bit0_port : Set Led_bit2_port                    ' Setze Bit 0 und 1
   Wait 1                                                   ' 1 Sekunde warten
   Set Led_bit1_port : Set Led_bit3_port                    ' Setzte Bit 1 und 3

   Wait 1                                                   ' 1 Sekunde warten

   ' ----- Nibble 1 -----
   ' Adresse &B01 einstellen (Bits werden einzeln angesprochen)
   Set Led_adr0_port
   ' Daten auf 0000 - alle LED's aus (Bits werden einzeln angesprochen)
   Reset Led_bit0_port : Reset Led_bit1_port : Reset Led_bit2_port : Reset Led_bit3_port
   ' Sequenz &B0101 -> &B1111 abfahren (Bits werden einzeln angesprochen)
   Set Led_bit0_port : Set Led_bit2_port                    ' Setze Bit 0 und 1
   Wait 1                                                   ' 1 Sekunde warten
   Set Led_bit1_port : Set Led_bit3_port                    ' Setzte Bit 1 und 3

   Wait 1                                                   ' 1 Sekunde warten

   ' ----- Nibble 2 -----
   ' Adresse &B10 einstellen (Bits werden einzeln angesprochen)
   Reset Led_adr0_port : Set Led_adr1_port
   ' Achtung: Reihenfolge Bits setzen für Adresse hat ggf. Einfluss auf niedrigeres
   ' oder höherwertiges Nibble da Adressengleichheit!!!

   ' Daten auf 0000 - alle LED's aus (Bits werden einzeln angesprochen)
   Reset Led_bit0_port : Reset Led_bit1_port : Reset Led_bit2_port : Reset Led_bit3_port
   ' Sequenz &B0101 -> &B1111 abfahren (Bits werden einzeln angesprochen)
   Set Led_bit0_port : Set Led_bit2_port                    ' Setze Bit 0 und
   Wait 1                                                   ' 1 Sekunde warten
   Set Led_bit1_port : Set Led_bit3_port                    ' Setzte Bit 1 und 3

   Wait 1                                                   ' 1 Sekunde warten

   ' ----- Nibble 3 -----
   ' Adresse &B11 einstellen (Bits werden einzeln angesprochen)
   Set Led_adr0_port
   ' Daten auf 0000 - alle LED's aus (Bits werden einzeln angesprochen)
   Reset Led_bit0_port : Reset Led_bit1_port : Reset Led_bit2_port : Reset Led_bit3_port
   ' Sequenz &B0101 -> &B1111 abfahren (Bits werden einzeln angesprochen)
   Set Led_bit0_port : Set Led_bit2_port                    ' Setze Bit 0 und
   Wait 1                                                   ' 1 Sekunde warten
   Set Led_bit1_port : Set Led_bit3_port                    ' Setzte Bit 1 und 3

   ' Ggf. Trace ausgeben
   #if Main_testmodus
      Print "- finished -"
   #endif

Return
'-- End Test_set_nibbles ------------------------------------------------------


'--------------------------------------------------------------------------
' Gosub-Routine: Test_reset_nibbles
' Testsequenz setzt über die 4 Adresse &B00, &B01, &B10 und &B11 nacheinander
' mit den Bitmuster &B0000 alle 4 Nibbles zurück.
```

```
' Bei "- finish -" sollten keine LED's mehr leuchten.
'--------------------------------------------------------------------------
Test_reset_nibbles:

   ' Ggf. Trace ausgeben
   #if Main_testmodus
      Print "Enter Test_reset_lownibble"
   #endif

   ' ----- Nibble 0 -----
   ' Adresse &B00 einstellen (Bits werden einzeln angesprochen)
   Reset Led_adr0_port : Reset Led_adr1_port

   ' Daten auf 0000 - alle LED's aus
   ' Da alle Bits in allen Nibbles ausgeschaltet werden sollen muss Daten-Nibble
   ' nicht verändert werden -> Ausgänge bleiben konstant geschaltet!
   Reset Led_bit0_port : Reset Led_bit1_port : Reset Led_bit2_port : Reset Led_bit3_port

   Wait 1                                                  ' 1 Sekunde warten

   ' ----- Nibble 1 -----
   ' Adresse &B01 einstellen (Bits werden einzeln angesprochen)
   Set Led_adr0_port

   Wait 1                                                  ' 1 Sekunde warten

   ' ----- Nibble 2 -----
   ' Adresse &B10 einstellen (Bits werden einzeln angesprochen)
   Reset Led_adr0_port : Set Led_adr1_port
   ' Achtung: Reihenfolge Bits setzen für Adresse hat ggf. Einfluss auf niedrigeres
   ' oder höherwertiges Nibble da Adressengleichheit!!!

   Wait 1                                                  ' 1 Sekunde warten

   ' ----- Nibble 3 -----
   ' Adresse &B11 einstellen (Bits werden einzeln angesprochen)
   Set Led_adr0_port

   ' Ggf. Trace ausgeben
   #if Main_testmodus
      Print "- finished -"
   #endif

Return
'-- End Test_reset_nibbles ------------------------------------------------


'--------------------------------------------------------------------------
' Gosub-Routine: Test_update_array
' Testsequenz steuer LED array über globale Variable und Update-Sequenz an
'--------------------------------------------------------------------------
Test_update_array:

   ' Ggf. Trace ausgeben
   #if Main_testmodus
      Print "Enter Test_update_array"
   #endif

   ' Muster 1: LED-Array (Bit-Matrix) mit Wert laden, ausgeben und 1 Sekunde warten
   Led_array = &H6666 : Gosub Led_update_array : Wait 1

   ' Muster 2: LED-Array (Bit-Matrix) mit Wert laden, ausgeben und 1 Sekunde warten
   Led_array = &H9999 : Gosub Led_update_array : Wait 1

   ' Muster 3: LED-Array (Bit-Matrix) mit Wert laden, ausgeben und 1 Sekunde warten
   Led_array = &H1111 : Gosub Led_update_array : Wait 1

   ' Muster 4: LED-Array (Bit-Matrix) mit Wert laden, ausgeben und 1 Sekunde warten
   Led_array = &H3333 : Gosub Led_update_array : Wait 1

   ' Muster 5: LED-Array (Bit-Matrix) mit Wert laden, ausgeben und 1 Sekunde warten
   Led_array = &H7777 : Gosub Led_update_array : Wait 1

   ' Muster 6: LED-Array (Bit-Matrix) mit Wert laden, ausgeben und 1 Sekunde warten
   Led_array = &HFFFF : Gosub Led_update_array : Wait 1

   ' LED-Array auf NULL setzten und alle LED aus
   Led_array = &H0000 : Gosub Led_update_array

   ' Nacheinander alle LED über LED-Array einschalten. Zeitraster 1 Sekunde
```

```
   Set Led_01_sw1 : Gosub Led_update_array : Wait 1
   Set Led_02_sw2 : Gosub Led_update_array : Wait 1
   Set Led_03_sw3 : Gosub Led_update_array : Wait 1
   Set Led_04_sw4 : Gosub Led_update_array : Wait 1
   Set Led_05_sw5 : Gosub Led_update_array : Wait 1
   Set Led_06_sw6 : Gosub Led_update_array : Wait 1
   Set Led_07_sw7 : Gosub Led_update_array : Wait 1
   Set Led_08_sw8 : Gosub Led_update_array : Wait 1
   Set Led_09_sw9 : Gosub Led_update_array : Wait 1
   Set Led_10_mem_full : Gosub Led_update_array : Wait 1
   Set Led_11_open_time : Gosub Led_update_array : Wait 1
   Set Led_12_open_light : Gosub Led_update_array : Wait 1
   Set Led_13_open_rain : Gosub Led_update_array : Wait 1
   Set Led_14_open_wind : Gosub Led_update_array : Wait 1
   Set Led_15_close_time : Gosub Led_update_array : Wait 1
   Set Led_16_close_temp : Gosub Led_update_array

   ' Ggf. Trace ausgeben
   #if Main_testmodus
      Print "- finished -"
   #endif

Return
'-- End Test_update_array ------------------------------------------------------


End                                                    'end program

'------------------------------------------------------------------------------
' Definition von globalen Konstantenfeldern
'------------------------------------------------------------------------------

' gibt's keine

'####################################### END ####################################
```

Software 13: Code für die LED-Matrix

8.21 LED-Zuordnung Schaltplan

In der folgenden Tabelle befinden sich alle LED's des Projekt beschrieben und den HW-Rssourcen im Schaltplan inkl. Funktion zugeordnet:

LED	*Funktion*	*Abkürzung*	*Schaltplan*	*Farbe*	*PCB*
1	5V present	--	Netzteil	grün	3
2	12V present	--	Netzteil	grün	3
3	Alive	LED_ALIVE	LED8	grün	2
4	Taste 1	LED1_SW1	S2D1	gelb	2
5	Taste 2	LED2_SW2	S3D1	gelb	2
6	Taste 3	LED3_SW3	S8D1	gelb	2
7	Taste 4	LED4_SW4	S9D1	gelb	2
8	Taste 5	LED5_SW5	S4D1	gelb	2
9	Taste 6	LED6_SW6	S10D1	gelb	2
10	Taste 7	LED7_SW7	S5D1	gelb	2
11	Taste 8	LED8_SW8	S6D1	gelb	2
12	Taste 9	LED9_SW9	S7D1	gelb	2
13	Speicher voll	LED10_MEM_FULL	LED13	rot	2
14	Auffahrgrund Zeit	LED11_OPEN_TIME	LED14	gelb	2
15	Auffahrgrund Licht	LED12_OPEN_LIGHT	LED15	gelb	2
16	Auffahrgrund Regen	LED13_OPEN_RAIN	LED16	rot	2
17	Auffahrgrund Wind	LED14_OPEN_WIND	LED17	rot	2
18	Zufahrgrund Zeit	LED15_CLOSE_TIME	LED18	gelb	2
19	Zufahrgrund Temperatur	LED16_CLOSE_TEMP	LED19	gelb	2
20	Heizung aktiv/inanktiv	LED_RAIN_HEATER	LED9	gelb	2
21	Relais Beschattung AUF	LED_REL_CONV_OPEN	LED4	gelb	2
22	Relais Beschattung ZU	LED_REL_CONV_CLOSE	LED3	gelb	2
23	Relais Lichtfunktion	LED_REL_LIGHT	LED2	gelb	2

24	Relais Dauerlicht	LED_REL_CONLUM	LED1	gelb	2
25	Lüfter SW	LED_FAN_SW	LED5	gelb	2
26	Lüfter ON	LED_FAN_12V	LED6	gelb	2
27	Sekundentakt	LED_SEC_TICK	LED11	grün	2
28	Windimpuls	LED_WIND_PULS	LED7	gelb	2
29	Regenerkennung	LED_RAIN_DETECT	LED10	gelb	2
30	DCF77 Impuls	LED_DCF77_TICK	LED20	gelb	2
31	Tastatur-Interrupt	INT7	LED12	gelb	2

Tabelle 136: LED-Zuordnung Schaltplan

Bedeutung der Farben:

LED's über LED-Array
LED's über Stand-Alone-LED'

8.22 Leiterplattenverbindungen

Im gesamten Projekt existieren drei einzelne PCB's welche miteinander verbunden werden müssen. Bei den PCB's handelt es sich um:

1. PCB1 = Kontroller-Board (gesamte Steuerelektronik für Funktionen)
2. PCB2 = Konsolen-Board (Taststus, Display und LED's inkl. dafür notwendige Steuerlogik)
3. PCB3 = Schaltnetzteil

Die Verbindung der einzelnen PCB's wird in den folgenden Unterkapiteln definiert und beschrieben.

8.22.1 Leiterplattenverbindung Kontroller <-> Konsole

Für die Verbindung des Kontroller-Boards mit dem Konsolen-Board wird eine 1:1 Verbindung über Pfostenstecker und Pfostenbuchsen mit Zugentlastung realisiert. Aus Platzgründen wird auf verriegelbare Steckverbinder verzichtet.

Der Pfostenstecker auf PCB1 (Controller-Board) wird gerade ausgeführt. Der Pfostenstecker auf PCB2 (Konsolen-Board) wird gewinkelt ausgeführt um Platz gegenüber der oberen Plexiglasabdeckung zu gewinnen.

Die dafür vorgehaltene Anzahl an Pins beträgt 40-Pin's.

Material:



Stiftleiste, 2x20-polig, gewinkelt



Flachbandkabel AWG28, 40-pol.,



Pfostenbuchse, 40-polig, mit Zugentlastung

Abbildung 137: Verbindungskonfiguration PCB1 <-> PCB2

Pinbelegung:

Pin Nr.	*Beschriftung*	*Funktion*	*Bemerkung*

1	VCC	+5V			Versorgungsspannung +5V
2	+12V	+12V			Versorgungsspannung +12V
3	RESET_SW	RESET_SW	*[µC011]*	*[20]*	Resettastersignale (schaltet gegen GND)
4	LCD_BIT0	LCD_BIT0	*[LCD007]*	*[35]*	LCD Datenbit 0
5	LCD_BIT1	LCD_BIT1	*[LCD007]*	*[36]*	LCD Datenbit 1
6	LCD_BIT2	LCD_BIT2	*[LCD007]*	*[37]*	LCD Datenbit 2
7	LCD_BIT3	LCD_BIT3	*[LCD007]*	*[38]*	LCD Datenbit 3
8	LCD_E	LCD_E	*[LCD008]*	*[39]*	LCD Enable
9	LCD_RS	LCD_RS	*[LCD008]*	*[40]*	LCD Command/Data
10	LCD_PWM	LCD_PWM	*[LCD009]*	*[5]*	PWM für Displayhelligkeit
11	LCD_ON	LCD_ON	*[LCD010]*	*[34]*	LCD ONOFF
12	SW_IN_1	SW_IN_1	*[SW010]*	*[51]*	Tastenmatrix 3x3 – Zeile 1
13	SW_IN_2	SW_IN_2	*[SW010]*	*[50]*	Tastenmatrix 3x3 – Zeile 2
14	SW_IN_3	SW_IN_3	*[SW010]*	*[49]*	Tastenmatrix 3x3 – Zeile 3
15	SW_OUT_1	SW_OUT_1	*[SW010]*	*[60]*	Tastenmatrix 3x3 – Spalte 1
16	SW_OUT_2	SW_OUT_2	*[SW010]*	*[59]*	Tastenmatrix 3x3 – Spalte 2
17	SW_OUT_3	SW_OUT_3	*[SW010]*	*[58]*	Tastenmatrix 3x3 – Spalte 3
18	KEY_INT	KEY_INT	*[SW010]*	*[9]*	Tastaturinterrupt
19	ALIVE	ALIVE	*[ALIVE]*	*[10]*	Alive LED ATmega *LED_ALIVE (LED3)*
20	LED21	LED004	*[LED004]*	*[CONV016]*	Relais Beschattung AUF angezogen *LED_REL_CONV_OPEN (LED21)*
21	LED22	LED005	*[LED005]*	[CONV017]	Relais Beschattung ZU angezogen *LED_REL_CONV_CLOSE (LED22)*
22	LED23	LED006	*[LED006]*	*[DIM012]*	Relais für Lichtfunktion angezogen *LED_REL_LIGHT (LED23)*
23	LED24	LED021	*[LED021]*	*[DIM013]*	Relais für Dauerlicht *LED_REL_CONLUM (LED24)*
24	DCF77	DCF77	*[DCF007]*	*[DCF77]*	DCF77 Impuls *Rem; DCF77 Empfänger sitzt auf PCB1!!!*
25	LED25	LED007	*[LED007]*	*[FREEZ006]*	Lüfter via SW zugeschaltet *LED_FAN_SW (LED25)*
26	LED26	LED008	*[LED008]*	*[FREEZ006]*	Lüfter via Dauerstrom eingeschaltet *LED_FAN_12V (LED26)*
27	LED27	LED010	*[LED010]*	*[16]*	Timer-Tick 1 sec *LED_SEC_TICK (LED27)*
28	LED28	LED011	*[LED011]*	*[WIND0011]*	Impulseingang Windsensor *LED_WIND_PULS (LED28)*
29	LED29	LED012	*[LED012]*	*[RAIN009]*	Statusanzeige Regensensor *LED_RAIN_DETECT (LED29)*
30	DATA_0	DATA_0	*[LED020]*	*[48]*	LED-Array Daten (Bit 0)
31	DATA_1	DATA_1	*[LED020]*	*[47]*	LED-Array Daten (Bit 1)
32	DATA_2	DATA_2	*[LED020]*	*[46]*	LED-Array Daten (Bit 2)
33	DATA_3	DATA_3	*[LED020]*	*[45]*	LED-Array Daten (Bit 3)
34	ADR_0	ADR_0	*[LED020]*	*[44]*	LED-Array Adresse (Bit 0)
35	ADR_1	ADR_1	*[LED020]*	*[43]*	LED-Array Adresse (Bit 1)
36	CLEAR	CLEAR	*[LED020]*	*[19]*	LED-Array Latch Clear
37	ADC0	ADC0	*[LIGHT003]*	*[61]*	Lichtsensor über OpAmp AD623
38	LED20	LED019	*[LED019]*	*[RAIN010]*	Heizung Regensensor aktiv/passiv *LED_RAIN_HEATER (LED20)*
39	GND	GND			Ground GND doppelt gebundelt
40	GND	GND			Ground GND doppelt gebundelt

Tabelle 137: PIN-Belegung Leiterplattenverbindung PCB1<->PCB2

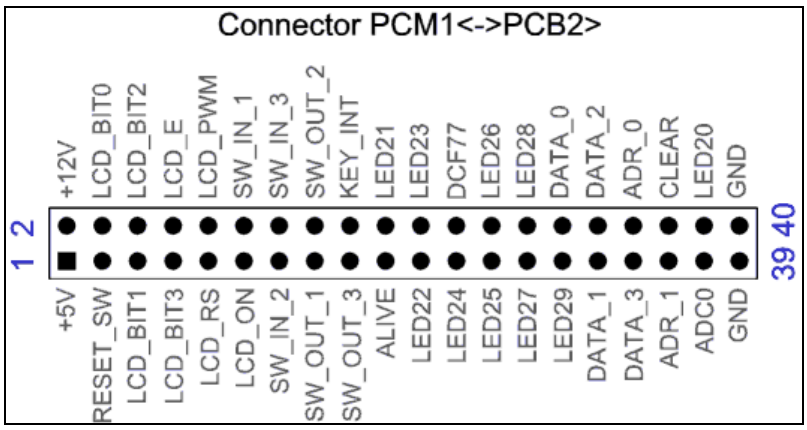


Abbildung 138: Connector PCB1<->PCB2

Bemerkung:
Der erste Link in Klammern *[blau kursiv]* ist ein Link auf das Requirement in der Gesamt-Checkliste. Der zweite Link in Klammern *[blau kursiv]* bezeichnet den jeweiligen PIN des ATmega128 und führt in die entsprechende Tabelle.

8.22.2 Leiterplattenverbindung Kontroller <-> Power

Für die Verbindung des Kontroller-Boards mit dem Schaltnetzteil wird eine 1:1 Verbindung über 13-polige Buchsenleisten mit zugehörigen 13-poligen Stiftleisten realisiert. Die dafür vorgehaltene Anzahl an Pins beträgt 13-Pin's.
Es wird zwischen Hochspannung und Niedervolt unterschieden. Da auf dem Kontrollerboard auch Netzspannung geschaltetw erden soll, wird dieser Leiterplattenverbindungstyp 2 Mal vorgesehen für:

1. PCB Verbindung für Niedervol +5V, +12V und Ground
2. PCB Verbindung für Netzspannung 230V~ (L1, N, PE)

Material:



13pol. Stiftleiste, gewinkelt, RM 2,54



13pol. Buchsenleiste, gewinkelt, RM 2,54

Abbildung 139: Verbindungskonfiguration PCB1 <-> PCB3

Pinbelegung Niedervolt:

Pin-Paar	*Funktion*	*Bemerkung*
1-2	+5V	Versorgungsspannung +5V doppelt gebundelt
3-4	+5V	Versorgungsspannung +5V doppelt gebundelt
5-6	--	frei
7-8	--	frei
9-10	+12V	Versorgungsspannung +12V doppelt gebundelt

11-12	+12V	Versorgungsspannung +12V doppelt gebundelt
13-14	--	frei
15-16	--	frei
17-18	--	frei
19-20	--	frei
21-22	GND	Ground GND dreifach gebundelt
23-24	GND	Ground GND dreifach gebundelt
25-26	GND	Ground GND dreifach gebundelt

Tabelle 138: PIN-Belegung Leiterplattenverbindung PCB1<->PCB3 Niedervolt

Achtung: Die Pin's werden immer in Paaren (ungerade+gerade Pins) beschaltet!

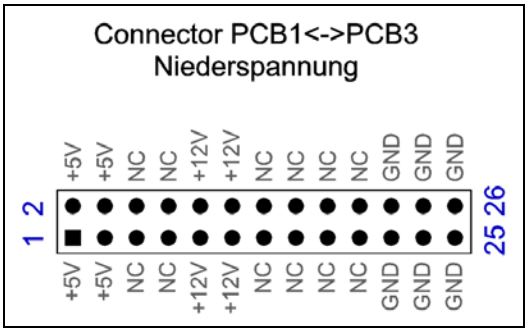


Abbildung 140: Connector PCB1<->PCB3 Niedervolt

Pinbelegung Hochvolt:

Pin-Paar	*Funktion*	*Bemerkung*
1-2	L1	Phase dreifach gebundelt
3-4	L1	Phase dreifach gebundelt
5-6	L1	Phase dreifach gebundelt
7-8	--	frei
9-10	--	frei
11-12	N	Nullleiter dreifach gebundelt
13-14	N	Nullleiter dreifach gebundelt
15-16	N	Nullleiter dreifach gebundelt
17-18	--	frei
19-20	--	frei
21-22	PE	Schutzleiter dreifach gebundetl
23-24	PE	Schutzleiter dreifach gebundetl
25-26	PE	Schutzleiter dreifach gebundetl

Tabelle 139: PIN-Belegung Leiterplattenverbindung PCB1<->PCB3 Hochvolt

Achtung: Die Pin's werden immer in Paaren (ungerade+gerade Pins) beschaltet!

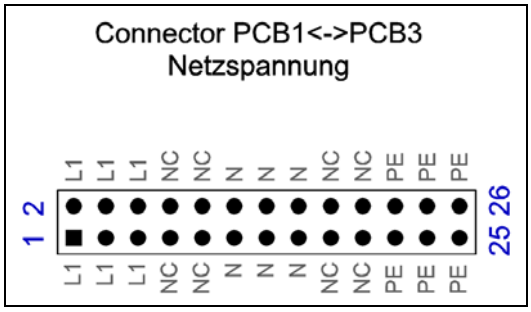


Abbildung 141: Connector PCB1<->PCB3 Netzspannung

9 Prototypen und Inbetriebnahme

Für das Prototyping und die Inbetriebnahme der im obigen Kapitel „Grundlagen" beschriebene Einzelthemen werden die Einzelteile in Verbindung mit dem STK500 und Labor-Lochrasterplatinen aufgebaut und in Betrieb genommen.

Die folgenden Stücklisten geben einen Überblick über die benötigten Bauelemente und ergeben gleichzeitig die Bestellisten von Conrad Electronic, Reichelt Electronic und Farnell.

Bauelemente Conrad Electronic:

Stk.	*Beschreibung*	*Bestellnummer*	*Einzelpreis*	*Gesamtpreis*
		Sonstiges		
1	KIPPSCHALTER R13-405AL-05 RT LED	701047-62	1,500	1,500
1	Piezo Schallwandler RMP-14SP	710107-LN	1,870	1,870
1	Axiallüfter 12V 40X40+	537314-62	14,950	14,950
1	Brückengleichrichter B40C1000=B40D	501204–62	0,910	0,910
1	SCHALTBUCHSE PRINTMONTAGE	737992-13	2,560	2,560
1	DC-STECKER 2,1MM / 5,0MM / 9,5MM GERADE	530020–62	2,180	2,180

Tabelle 140: Bauelemente Conrad Electronic Prototyping

Bauelemente Reichelt Electronic:

Stk.	*Beschreibung*	*Bestellnummer*	*Einzelpreis*	*Gesamtpreis*
		Kondensatoren		
2	Keramik-Kondensator 47nF	KERKO 47N	0,070	0,140
30	Keramik-Kondensator 100nF	KERKO 100N	0,072	0,144
2	Keramik-Kondensator 22pF	KERKO 22P	0,050	0,100
2	Keramik-Kondensator 1,2N	KERKO 1,2N	0,041	0,082
1	FOLIEN-KONDENSATOR	MKS-4-630 10N	0,110	0,110
7	WIMA Folienkondensator, Rm 5mm, 100nF	MKS-2 100N	0,070	0,420
1	WIMA, Folienkondensator, Impulsfest; 220nF / 275V	MKP-10-630 220N	0,940	0,940
2	Elektrolytkondensator, 5x11mm, RM 2,0mm	RAD 0,1/100	0,040	0,080
2	Elektrolytkondensator, 5x11mm, RM 2,0mm	RAD 0,47/63	0,040	0,080
15	Elektrolytkondensator 1µF, 5x11mm, RM 2,0mm	RAD 1/63	0,040	0,200
4	Elektrolytkondensator, 6,3x11mm, RM 2,5mm	RAD 47/35	0,040	0,080
2	Elektrolytkondensator, 8x11mm, RM 3,5mm	RAD 100/35	0,072	0,144
2	Elektrolytkondensator, 10x16mm, RM 5,0mm	RAD 470/25	0,110	0,220
2	Elektrolytkondensator, 12,5x25mm, RM 5,0mm	RAD 1.000/35	0,310	0,620
2	Elektrolytkondensator, 16x25mm, RM 7,5mm	RAD 2.200/35	0,470	0,94
1	Elektrolytkondensator, 18x40mm, RM 7,5mm	RAD 4.700/35	1,050	1,050
1	Tantal-Kondensator, Rm 2,5, 10µF/25V	TANTAL 10/25	0,250	0,250
		Widerstände		
1	Metallschichtwiderstand 30 Ohm	METALL 30,0	0,082	0,082
1	Metallschichtwiderstand 39 Ohm	METALL 39,0	0,080	0,080
1	Metallschichtwiderstand 100 Ohm	METALL 100	0,080	0,080
1	Metallschichtwiderstand 220 Ohm	METALL 220	0,080	0,080
1	Metallschichtwiderstand 330 Ohm	METALL 330	0,080	0,080
2	Metallschichtwiderstand 470 Ohm	METALL 470	0,080	0,160
1	Metallschichtwiderstand 500 Ohm (499 Ohm)	METALL 499	0,080	0,080
3	Metallschichtwiderstand 1,00 K-Ohm	METALL 1,00K	0,080	0,240
1	Metalloxidschicht-Widerstand 2W, 5% 1,0 K-Ohm	2W METALL 1,0K	0,090	0,090

2	Metallschichtwiderstand 1,20 K-Ohm	METALL 1,20K	0,080	0,160
7	Metallschichtwiderstand 1,50 K-Ohm	METALL 1,50K	0,080	0,400
6	Metallschichtwiderstand 4,70 K-Ohm	METALL 4,70K	0,080	0,080
2	Metallschichtwiderstand 6,00 K-Ohm (6,04K)	METALL 6,04K	0,080	0,080
12	Metallschichtwiderstand 10,0 K-Ohm	METALL 10,0K	0,080	0,400
4	Metallschichtwiderstand 15,0 K-Ohm	METALL 15,0K	0,080	0,240
2	Metallschichtwiderstand 22,0 K-Ohm	METALL 22,0K	0,080	0,160
1	Metallschichtwiderstand 39,0 K-Ohm	METALL 39,0K	0,080	0,080
2	Metallschichtwiderstand 100 K-Ohm	METALL 100K	0,080	0,160
1	Metallschichtwiderstand 330 K-Ohm	METALL 330K	0,080	0,080
1	Metallschichtwiderstand 400 K-Ohm (402k)	METALL 402K	0,080	0,080
2	Metallschichtwiderstand 1,00 M-Ohm	METALL 1,00M	0,080	0,160
2	Einstellpotentiometer, stehend, 10mm, 5,0 K-Ohm	PIHER 10-S 5,0K	0,190	0,380
8	Widerstands-Netzwerk, 4Wid./8Pins, 1,5 K-Ohm	SIL 8-4 1,5K	0,140	0,560
	Halbleiter			
7	Diode	1N 4148	0,020	0,080
1	Diode	BAT 43	0,110	0,110
2	Diode	1N 5822	0,160	0,320
1	Überspannungsschutzdiode 6,8V	P6KE 6,8CA	0,170	0,170
1	Überspannungsschutzdiode15V	P6KE 15CA	0,170	0,170
3	Z-Diode 0,5W	ZF 5,1	0,031	0,093
1	ATMega AVR-RISC-Controller, TQFP-44	ATMEGA 16-16 TQ	2,600	2,600
1	ATMega AVR 128-16 TQFP64	ATMEGA 128-16 TQ	8,050	8,050
2	RS232E Interface, DIL-16	MAX 232 EPE	1,900	1,900
1	MAXIM-IC,Schnittstellenbaustein, SO-16	MAX 3232 CSE	3,400	3,400
1	8-BIT BUSTREIBER, DIL-20	74HCT 244	0,260	0,260
1	Octal Driver (IC-Schsltung), DIL20	74HCT 245	0,310	0,310
1	Inverter, DIL14	74HCT 04	0,180	0,180
1	AND-Gatter (IC-Schaltung), DIL14	74HCT 08	0,180	0,180
1	XOR-Gatter (IC-Schaltung), DIL14	74HCT 86	0,220	0,220
4	Hex inv Schmitt-Trigger, DIL14	74HCT 14	0,210	0,840
2	3-fach AND-Gatter, DIL14	74HCT 11	0,180	0,360
1	NAND, DIL14	74HCT 132	0,280	0,280
2	LED, 3mm, Standard, grün	LED 3MM ST GN	0,082	0,164
1	LED, 3mm, Standard, gelb	LED 3MM ST GE	0,082	0,082
10	LED 3mm, low-Current, grün	LED 3MM 2MA GN	0,090	0,900
29	LED 3mm, low-Current, gelb	LED 3MM 2MA GE	0,090	1,800
4	LED 3mm, low-Current, rot	LED 3MM 2MA RT	0,090	0,360
2	TRANSISTOR	BC 547A	0,031	0,062
5	TRANSISTOR	BC 547B	0,031	0,155
1	TRANSISTOR	BC 557A	0,031	0,031
1	TRANSISTOR	BC 337-16	0,041	0,041
1	TRANSISTOR	BD 139	0,120	0,120
1	TRIAC 8A/400V	TIC 225D	0,790	0,790
1	Optokoppler, DIL8	4N 25	0,170	0,170
1	Optokoppler, DIL8	MOC 3020	0,290	0,000
2	DIL-GLEICHRICHTER, DIL8	B250C800DIP	0,240	0,480
3	BRÜCKEN-GLEICHRICHTE	B40C1500-WW+	0,280	0,280
1	Stromregler, TO-92	LM 2936 Z5,0	1,650	1,650
1	Spannungsregler 2A positiv, TO-220	µA 78S12	0,360	0,360
1	Spannungsregler 2A positiv, TO-220	µA 78S05	0,360	0,360
1	Spannungsregler, TO-220-5	LM 2576 T5,0	1,650	1,650
1	Schaltregler 3A, 12V, 45Vs, TO-220-5	LM 2576 T12	1,950	1,950
1	DOG-Serie 3,3V Hintergrund: weiss	EA DOGM162W-A	11,25	11,250
1	Led-Beleuchtung für EA DOGM..Farbe: weiss	EA LED55X31-W	10,20	10,200
1	CMOS seriell EEPROM, 5V, DIP-8	ST 24C512 BN6	2,600	2,600
1	RTC Time Keeper, DIP-8	DS 1302	4,500	4,500
1	Op-amp, DIP-8	LM 358 DIP	0,092	0,092

Sonstiges				
10	Kurzhubtaster	Taster 3301	0,082	0,082
1	Miniatur-Schiebeschalter	T 681	2,700	2,700
1	Standardquarz, Grundton, 16,0 MHz	16,0000-HC49U-S	0,240	0,240
1	Standardquarz, Grundton, 7,3728 MHz	7,3728-HC49U-S	0,240	0,240
1	Mini Uhrenquarz 32,768 kHz	0,032768	0,140	0,140
1	50pol. Stiftleiste, gerade, RM 2,54	SL 1X50G 2,54	0,680	0,680
1	2x50pol.-Stiftleiste, gerade, RM 2,54	SL 2X50G 2,54	1,400	1,400
9	Kurzschlussbrücke, schwarz, RM 2,54, vergol-det	JUMPER 2,54 SW	0,040	0,360
1	IC-Fassung, 64-polig, einreihig, RM 2,54, gera-de	SPL 64	0,890	0,890
1	D-SUB-Buchse, 9-polig f. Flachbandkabel	D-SUB BU 09FB	0,460	0,460
1	D-SUB-Stecker, 25-polig, gewinkelt, RM 9,4	D-SUB ST 25EU	0,430	0,430
1	D-SUB-Buchse, 9-polig, gewinkelt, RM 9,4	D-SUB BU 09GW	0,560	0,560
1	D-SUB-Buchse, 9-polig, gewinkelt, TM9,4	D-SUB BU 09EU	0,280	0,280
10	Pfostenbuchse, 10-polig, mit Zugentlastung	PFL 10	0,092	0,920
3	Anreihklemme 2-polig, RM5,0	AKL 055-02	0,140	0,420
1	Anreihklemme 3-polig, RM5,0	AKL 055-03	0,140	0,140
1	Anreihklemme 4-polig, RM5,0	AKL 055-04	0,140	0,140
1	Anschlussklemme 2-polig, RM7,5mm	AKL 171-02	0,260	0,260
6	Sicherungshalter, 5x20mm, max. 6,3A-250V	PL 120000	0,041	0,246
1	Feinsicherung 5x20mm, flink 0,125A	FLINK 0,125A	0,120	0,120
1	Feinsicherung 5x20mm, flink 0,160A	FLINK 0,16A	0,072	0,072
2	Feinsicherung 5x20mm, flink 1,0A	FLINK 1,0A	0,072	0,144
1	Feinsicherung 5x20mm, träge 2,0A	TRAEG 2,0A	0,072	0,072
1	Feinsicherung 5x20mm, mittelträge 3,15A	MTR. 3,15A	0,072	0,072
2	Aufsteckkühlkörper für Gehäuse TO-220, 25K/W	V FI342	1,150	2,300
3	Steckrelais, 1x UM, 250V/10A, 12V, RM 5,0mm	FIN 40.51.9 12V	1,600	4,800
1	Funk-Entstördrossel, 50 Windungen, 100µ	FED 100µ	1,600	1,600
2	High Current Funkentstördrossel, 100µ	77A 100µ	0,920	1,840
1	Drosselspule, Festinduktivität, axial, 10µ	SMCC 10µ	0,160	0,160
2	IC-Fassung, 20-polig, einreihig, RM 2,54, gera-de	SPL20	0,310	0,620
1	Kippschalter, 2-polig, 3A-250VAC, Ein-Ein	MS 500F	1,700	1,700
2	Eingabetaster SW, Schaltspannung: 24V,fürLED	DTL 2 SW	1,850	3,700
2	Eingabetaster, Schaltspannung: 24V, für led, ge	DTL 2 GE	1,850	3,700
4	Eingabetaster blau, Schaltspannung: 24V, f.LED	DTL 2 BL	1,850	7,400
1	Eingabetaster rot, Schaltspannung: 24V, fürLED	DTL 2 RT	1,850	1,850
2	IC-Sockel, 24-polig, superflach, gedreht, ver-gold.	GS 24P	0,270	0,540
1	IC-Sockel, 16-polig, superflach, gedreht, ver-gold.	GS 16P	0,180	0,180
2	IC-Sockel, 20-polig, superflach, gedreht, ver-gold.	GS 20P	0,240	0,480
10	IC-Sockel, 14-polig, superflach, gedreht, ver-gold.	GS 14P	0,170	1,700
10	IC-Sockel, 8-polig, superflach, gedreht, ver-gold.	GS 8P	0,100	1,000
1	IC-Sockel, 6-polig, superflach, gedreht, ver-gold.	GS 6P	0,090	0,090
1	Trafo 30VA, 2x 15V, 2x 1000mA	UI 39/21 215	11,20	11,20
2	Pfostenstecker, 40-polig, mit Verrieglung, ge-rade	PSL 40	0,650	1,300
2	Pfostenbuchse, 40-polig, mit Zugentlastung	PFL 40	0,120	0,240
2	2x13pol. Buchsenleiste, gewinkelt, RM 2,54	BL 2X13W8 2,54	0,640	1,280
2	2x13pol. Stiftleiste, gewinkelt, RM 2,54	SL 2X13W 2,54	0,130	0,260

1	Stiftleiste	Handanfertigung	--	--
1	Flachbandkabel AWG28, 40-pol.,	AWG 28-40F 3M	11,800	11,800
1	Platinensteckverbinder gerade, weiß	PS 25/2G WS	0,250	0,250
1	Lithium-Knopfzelle, 3 Volt, 38 mAh, 12,5x2,5mm	BR 1225	0,850	0,850
1	Knopfzellenhalter für Ø 12 mm	KZH 12-1	1,750	1,750

Tabelle 141: Bauelemente Reichelt Electronic Prototyping

Die in blauer Schrift gekennzeichneten Zeilen wurden nach der Entscheidung, das Netzteil mit Schaltreglern und als abgesetzte Platine umzusetzen hinzugefügt bzw. die Stückzahlen ergänzt.

Die in grüner Schrift gekennzeichneten Zeilen wurden nach Inbetriebnahme der Prototypen und Ergänzung der Schaltpläne hinzugefügt bzw. die Stückzahlen ergänzt.

Die in oranger Schrift gekennzeichneten Zeilen wurden nach Entwicklung des ATmega128 Evaluation-Board hinzugefügt bzw. die Stückzahlen ergänzt.

Bauelemente Farnell:

Stk.	*Beschreibung*	*Bestellnummer*	*Einzelpreis*	*Gesamtpreis*
		Kondensatoren		
2	SMD CAPACITOR, 0603 22PF 50V	8819831	0,051	0,102
5	SMD CAPACITOR, 100NF X7R 50V 0603	1414028	0,076	0,380
		Widerstände		
2	SMD RESISTOR, RC22H 0603 1K	9238484	0,064	0,128
1	SMD RESISTOR, RC22H 0603 4K7	9238565	0,064	0,064
1	SMD RESISTOR, RC22H 0603 10K	9238603	0,064	0,064
		Halbleiter		
1	Sensirion Feuchtesensor SHT71	4130686	22,30	22,300
1	SENSOR, L/VOLTS	1182346	4,690	4,690
1	AD623ANZ	9603700	6,450	6,450
		Induktivitäten		
2	PE-52626 Schaltregler Induktivität	1209548	9,950	19,900

Tabelle 142: Bauelemente Farnell Prototyping

Bauelemente RS-Online:

Stk.	*Beschreibung*	*Bestellnummer*	*Einzelpreis*	*Gesamtpreis*
		Sonstiges		
1	Steckverbinder,Leiterplatten,MODU,Crimpsockelgehäuse,nicht abgeschirmt,einreihig,2-polig	233-1817 (grün) 360-5967	4,150	4,150
1	Steckverbinder,MODUIV/V,Crimpkontakt,Snap-In,32-28AWG,0.63mm,vergoldet	233-1889	9,700	9,700
2	Diode	1N 5821	1,640	1,640

Tabelle 143: Bauelemente RS-Online Prototyping

10 Hardware

10.1 Verbindungskabel und Konnektoren

10.1.1 Verbindung Außensensorik zu ConvCtrl

Für die Verbindung des Wintgebers und des Regensensors ist eine 10-adrige Steuerleitung durch den Wintergarten verlegt.

Pinbelegung vom Regensensor zum Verteiler:

Die folgende Tabelle zeigt die Pinbelegung der 10-adrigen Steuerleitung für den Anschluss des Regensensors:

Pin	*Farbe*	*Funktion*
1	rot	+ 12V
2	schwaz	GND
3	braun	Regensignal
4	gelb	Heizung S1
5	blau	Heizung S2
6	weiß	nc
7	rosa	nc
8	grau	nc
9	grün	nc
10	lila	nc

Tabelle 144: Pinbelegung Steuerleitung Regensensor

Pinbelegung vom Windmesser zum Verteiler:

Die folgende Tabelle zeigt die Pinbelegung der 2-adrigen Steuerleitung für den Anschluss des Windmessers:

Pin	*Farbe*	*Funktion*
1	braun	S1
2	weiß	S2

Tabelle 145: Pinbelegung Steuerleitung Windmesser

Pinbelegung der Hauptleitung vom Verteiler zum Board:

Die folgende Tabelle zeigt die Pinbelegung der 10-adrigen Steuerleitung für den Anschluss der Sensorik:

Pin	*Farbe*	*Funktion*
1	rot	+ 12V
2	schwaz	GND
3	braun	Regensignal
4	gelb	Heizung S1
5	blau	Heizung S2
6	weiß	nc
7	rosa	nc
8	grau	nc
9	grün	Wind S1
10	lila	Wind S2

Tabelle 146: Pinbelegung Steuerleitung Außensensorik

Anschlussverbindungen:

Windmesser		*Regensensor*		*Hauptleitung*	
Pin	*Farbe*	*Pin*	*Farbe*	*Pin*	*Farbe*

--	--	1	rot	1	rot
--	--	2	schwaz	2	schwaz
--	--	3	braun	3	braun
--	--	4	gelb	4	gelb
--	--	5	blau	5	blau
--	--	--	--	6	weiß
--	--	--	--	7	rosa
--	--	--	--	8	grau
1	braun	--	--	9	grün
2	weiß	--	--	10	lila

Tabelle 147: Anschlussverbindungen Wind<->Regen<->ConvCtrl

Das folgende Bild zeigt die auf einem Aluminium-Träger montierte Außensensorik bereits am Wintergarten montiert:



Abbildung 142: Außensensorik auf Aluminium-Träger

10.2 Festlegung von Netzklassen im Projekt

10.2.1 Netzklassen für das ATmega128 Evaluation-Board

Für die Umsetzung der PCB in EAGLE werden die folgenden Netzklassen definiert:

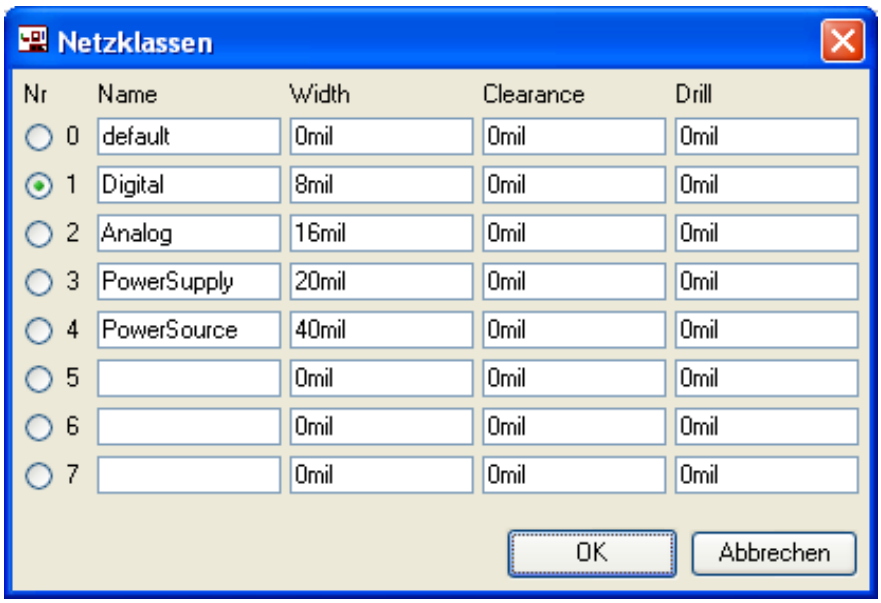


Abbildung 143: Definition der Netzklassen Eva-Board

Diese Vorgaben führen bei einem einfachen Schaltbild zur folgenden Umsetzung auf dem Board:



Schaltbild 47: Schaltbild für Definition von Netzklassen – Eva-Board

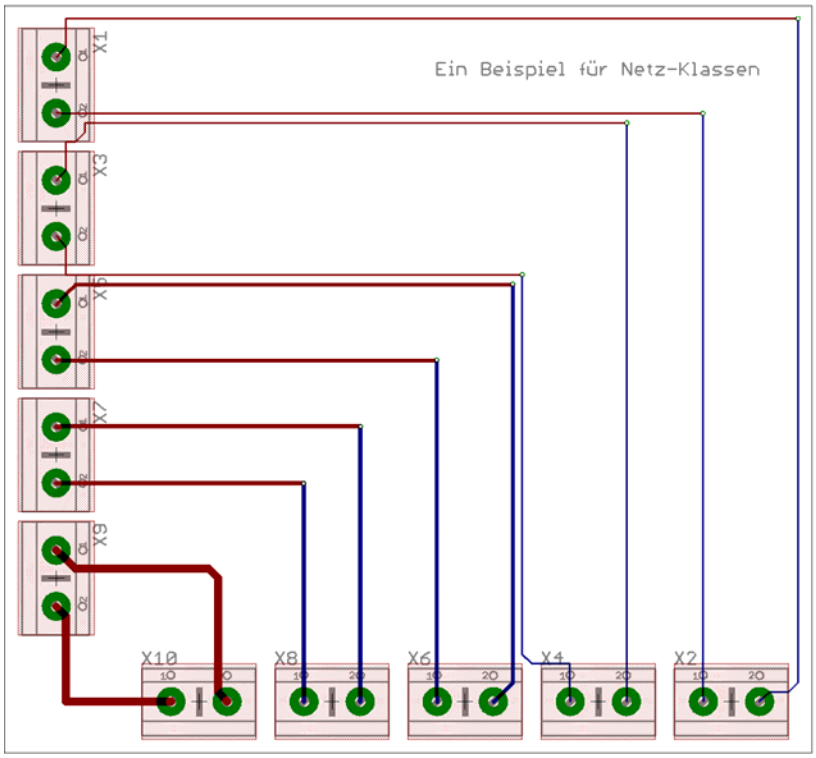


Abbildung 144: Demoboard N etzklassen Eva-Board

10.2.2 Netzklassen für PCB1 Controller-Board

Für die Umsetzung der PCB1 in EAGLE werden die folgenden Netzklassen definiert:

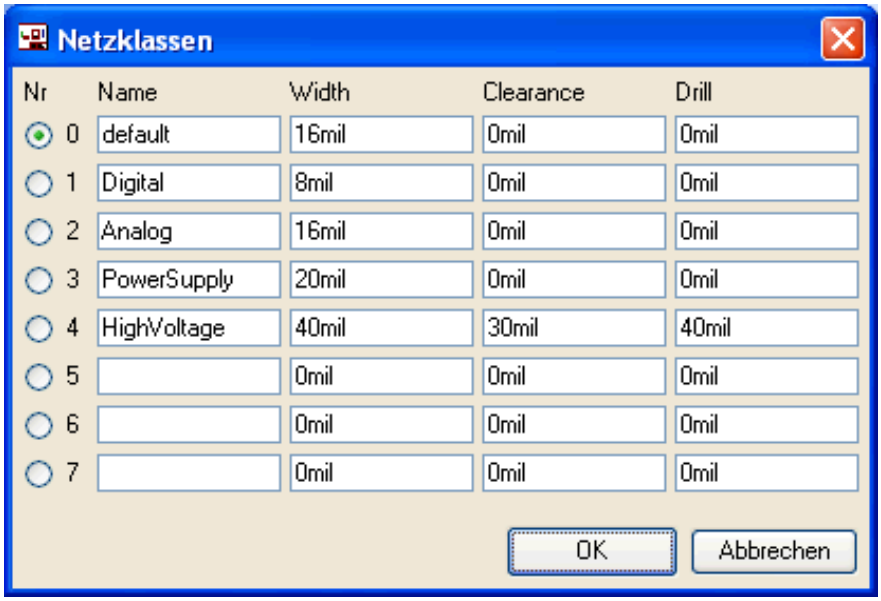


Abbildung 145: Definition der Netzklassen PCB1

Diese Vorgaben führen bei einem einfachen Schaltbild zur folgenden Umsetzung auf dem Board:



Schaltbild 48: Schaltbild für Definition von Netzklassen PCB1

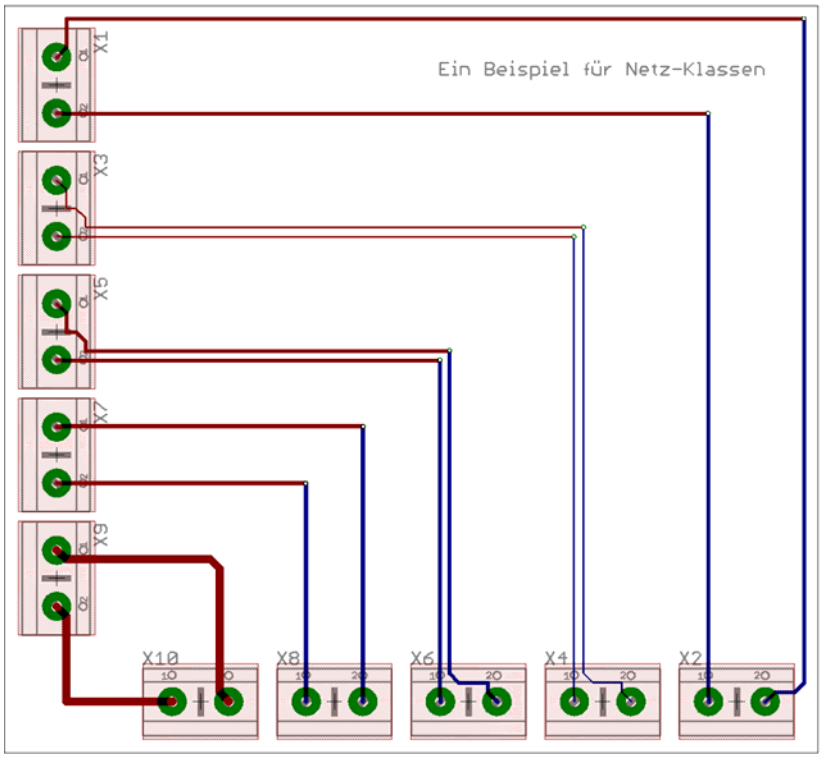


Abbildung 146: Demoboard N etzklassen PCB1

Weiter gilt die Vorgabe von 150 mil für alle High Voltage Signalleitungen. In den Klassen wird für High Voltage eine Breit von min. 40 mil festgelegt damit Anschlüsse an PIN-Header ohne Verletzung der Design-Rules möglich ist. Die zusammengeführten Signale auf dem PCB werden jedoch mit 150 mil geroutet um eine ausreichende Stärke für die notwendigen Strombelastungen zu erreichen.

Für die Strombelastung gibt es die Fausregel:

- 1 mm Leiterbahn bei 35µ Kupferstärke für 1 Ampere Strom

10.2.3 Netzklassen für PCB2 Konsolen-Board

Für die Umsetzung der PCB2 in EAGLE werden die folgenden Netzklassen definiert:

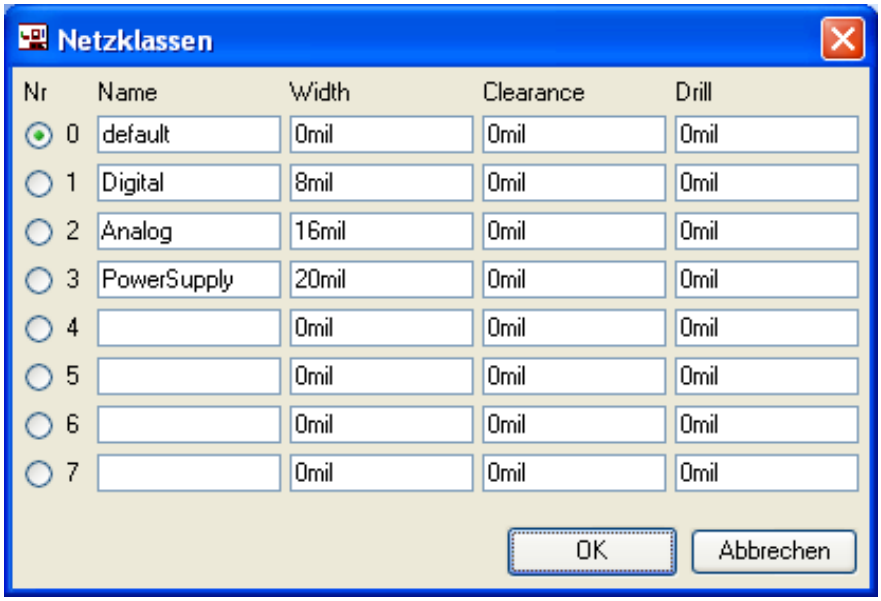


Abbildung 147: Definition der Netzklassen PCB2

Diese Vorgaben führen bei einem einfachen Schaltbild zur folgenden Umsetzung auf dem Board:



Schaltbild 49: Schaltbild für Definition von Netzklassen PCB2

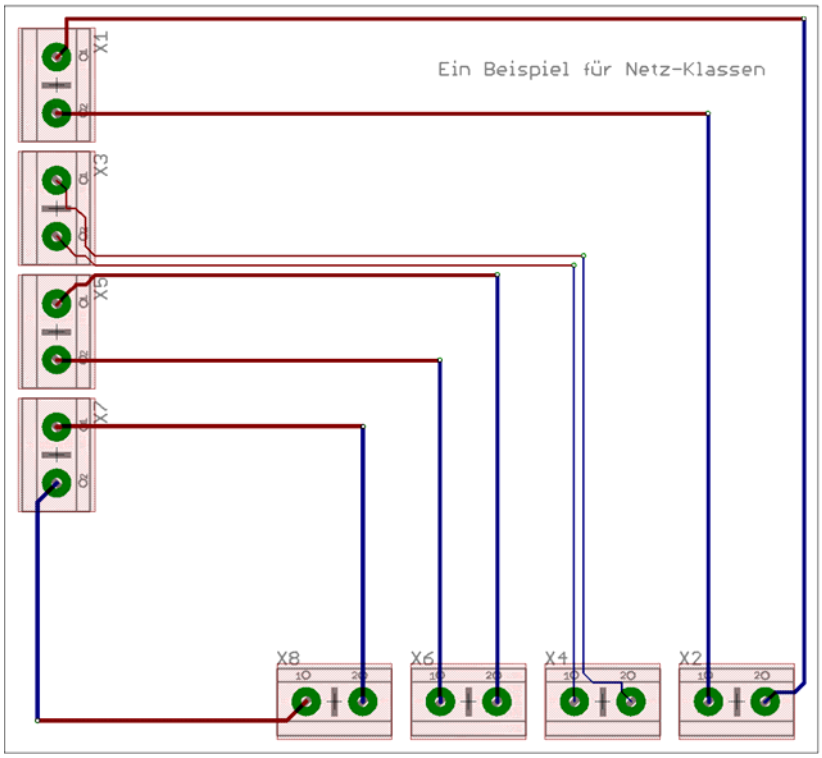


Abbildung 148: Demoboard N etzklassen PCB2

10.2.4 Netzklassen für PCB3 Power-Board

Für die Umsetzung der PCB in EAGLE werden die folgenden Netzklassen definiert:

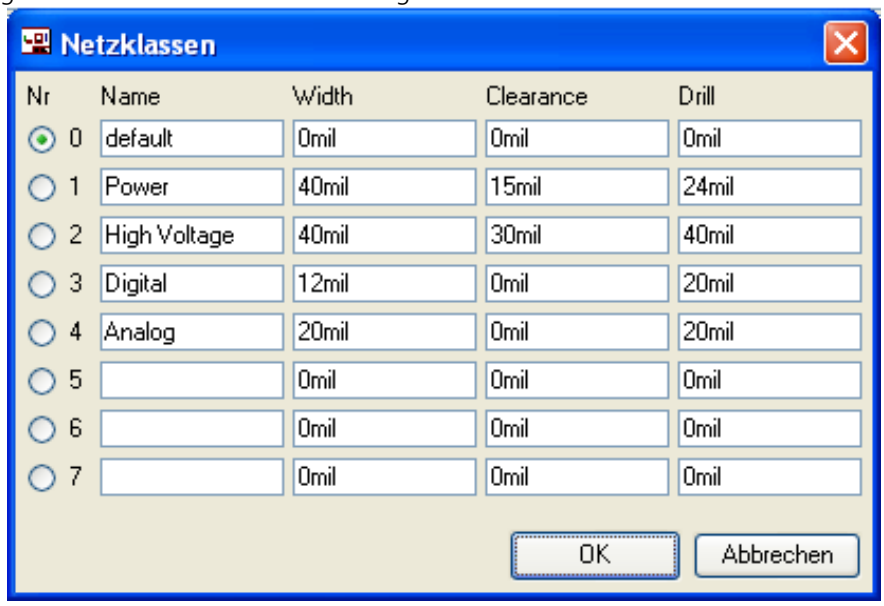


Abbildung 149: Definition der Netzklassen PCB3

Diese Vorgaben führen bei einem einfachen Schaltbild zur folgenden Umsetzung auf dem Board:



Schaltbild 50: Schaltbild für Definition von Netzklassen PCB3

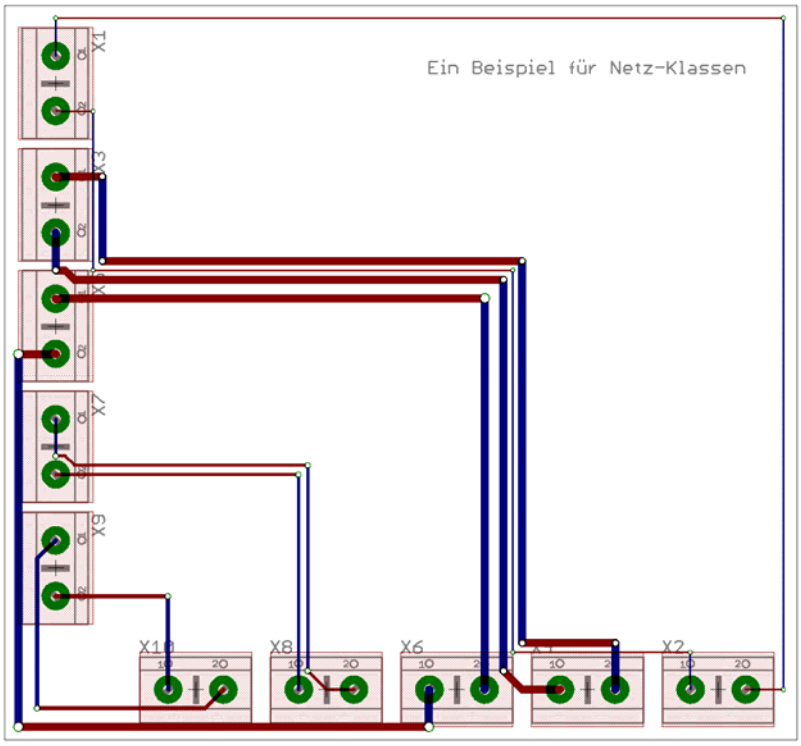


Abbildung 150: Demoboard N etzklassen PCB3

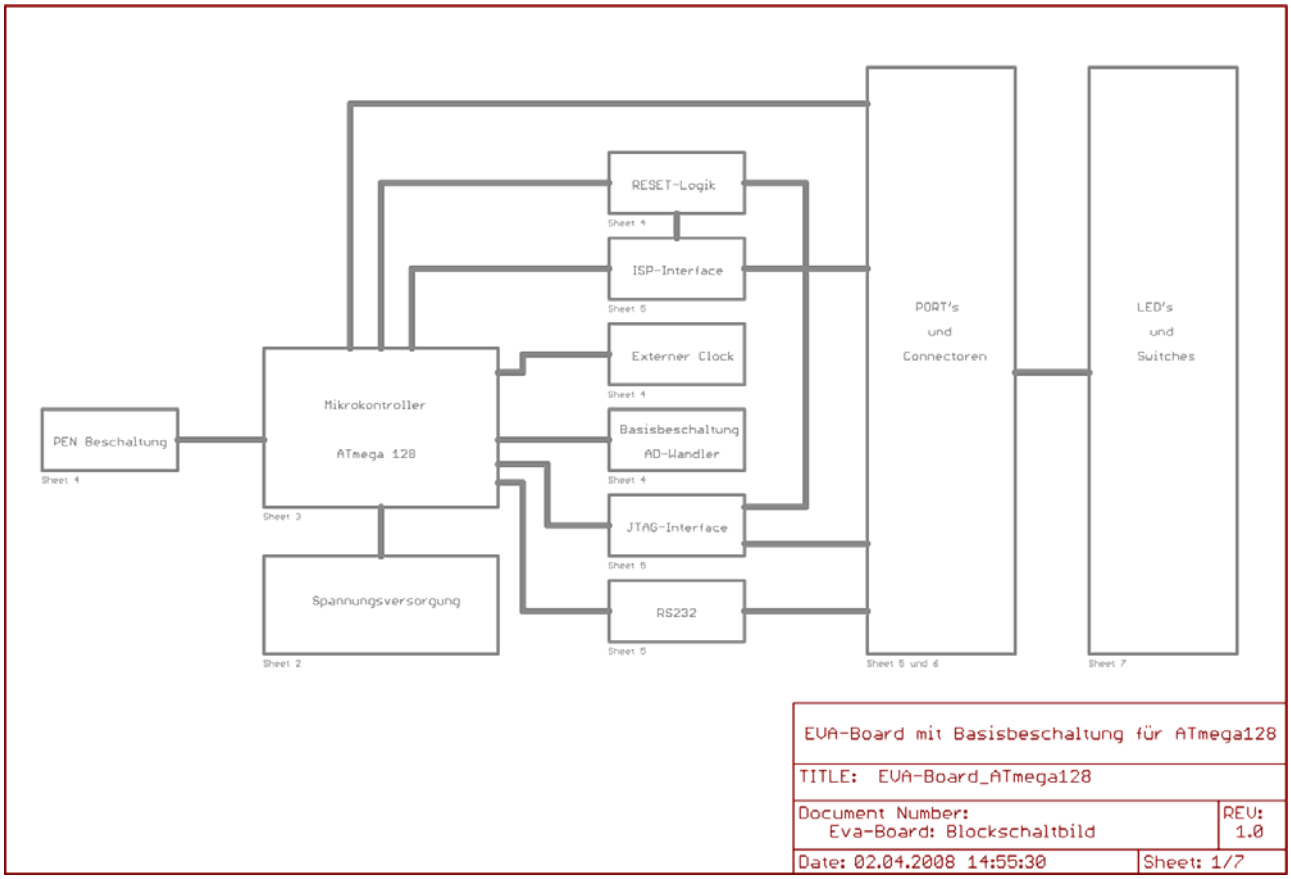
Weiter gilt die Vorgabe von 150 mil für alle High Voltage Signalleitungen. In den Klassen wird für High Voltage eine Breit von min. 40 mil festgelegt damit Anschlüsse an PIN-Header ohne Verletzung der Design-Rules möglich ist. Die zusammengeführten Signale auf dem PCB werden jedoch mit 150 mil geroutet um eine ausreichende Stärke für die notwendigen Strombelastungen zu erreichen.

Für die Strombelastung gibt es die Fausregel:

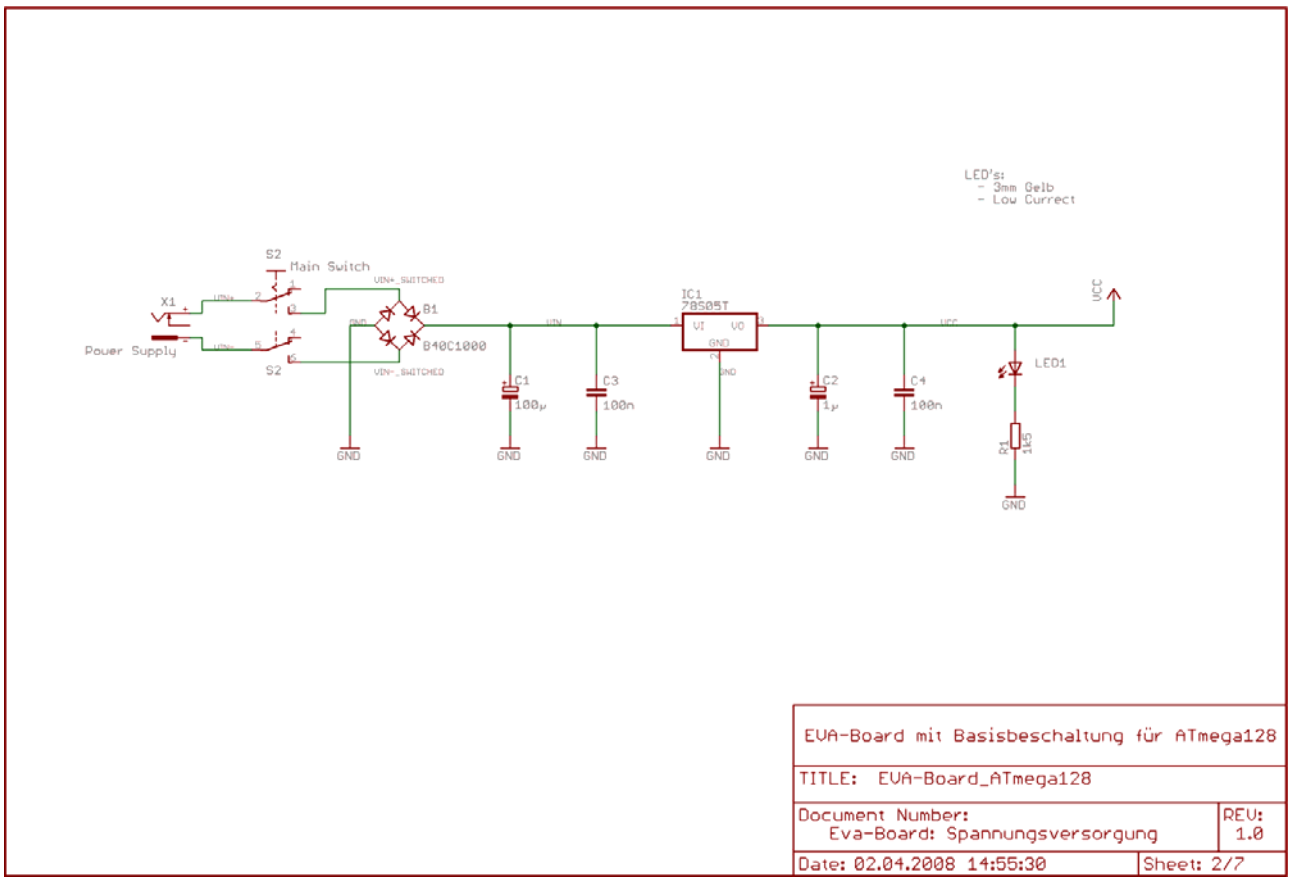
- 1 mm Leiterbahn bei 35µ Kupferstärke für 1 Ampere Strom

10.3 Das ATmega128 Evaluation Board

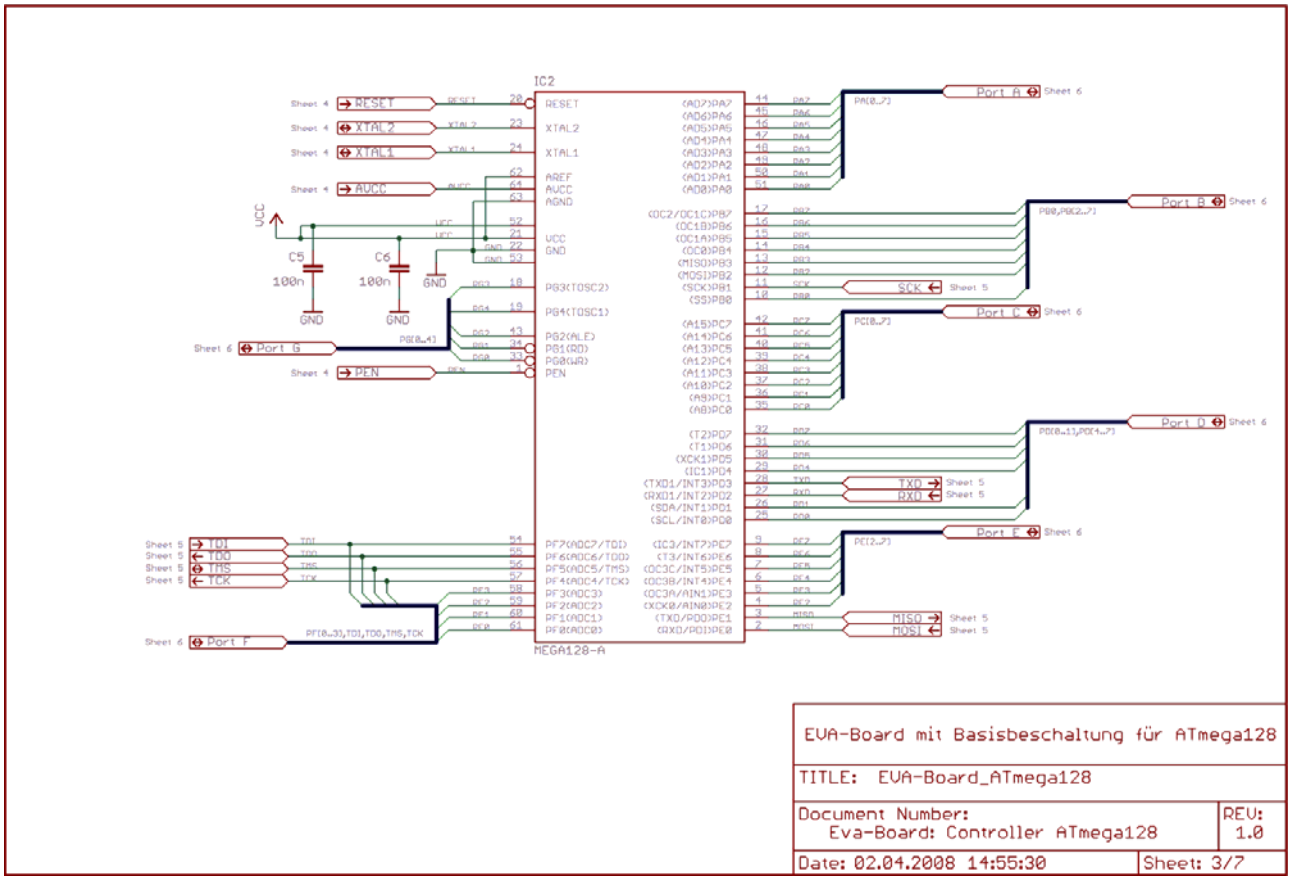
10.3.1 Schematic



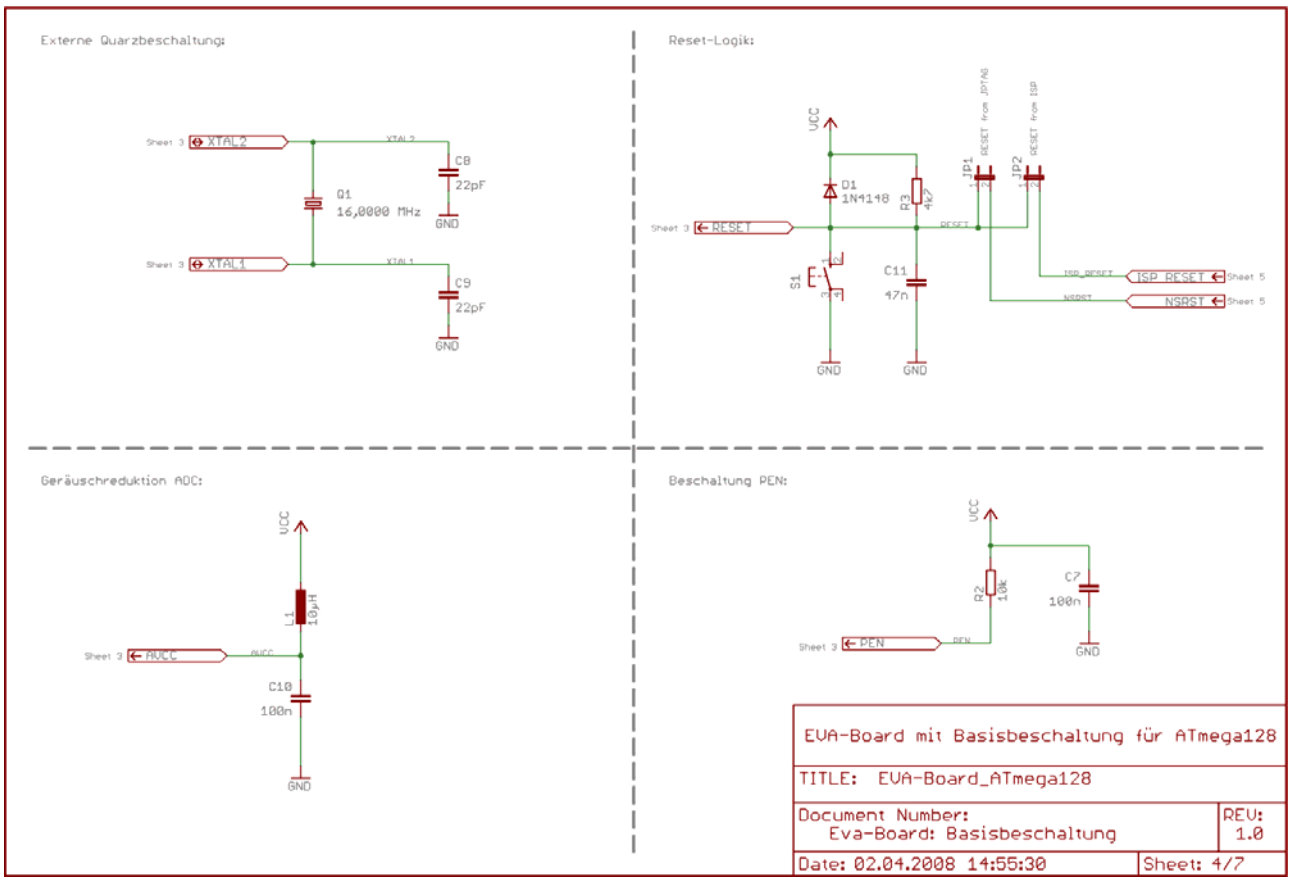
Schaltbild 51: Schaltbild Atmega128 Evaluation-Board - Sheet 1



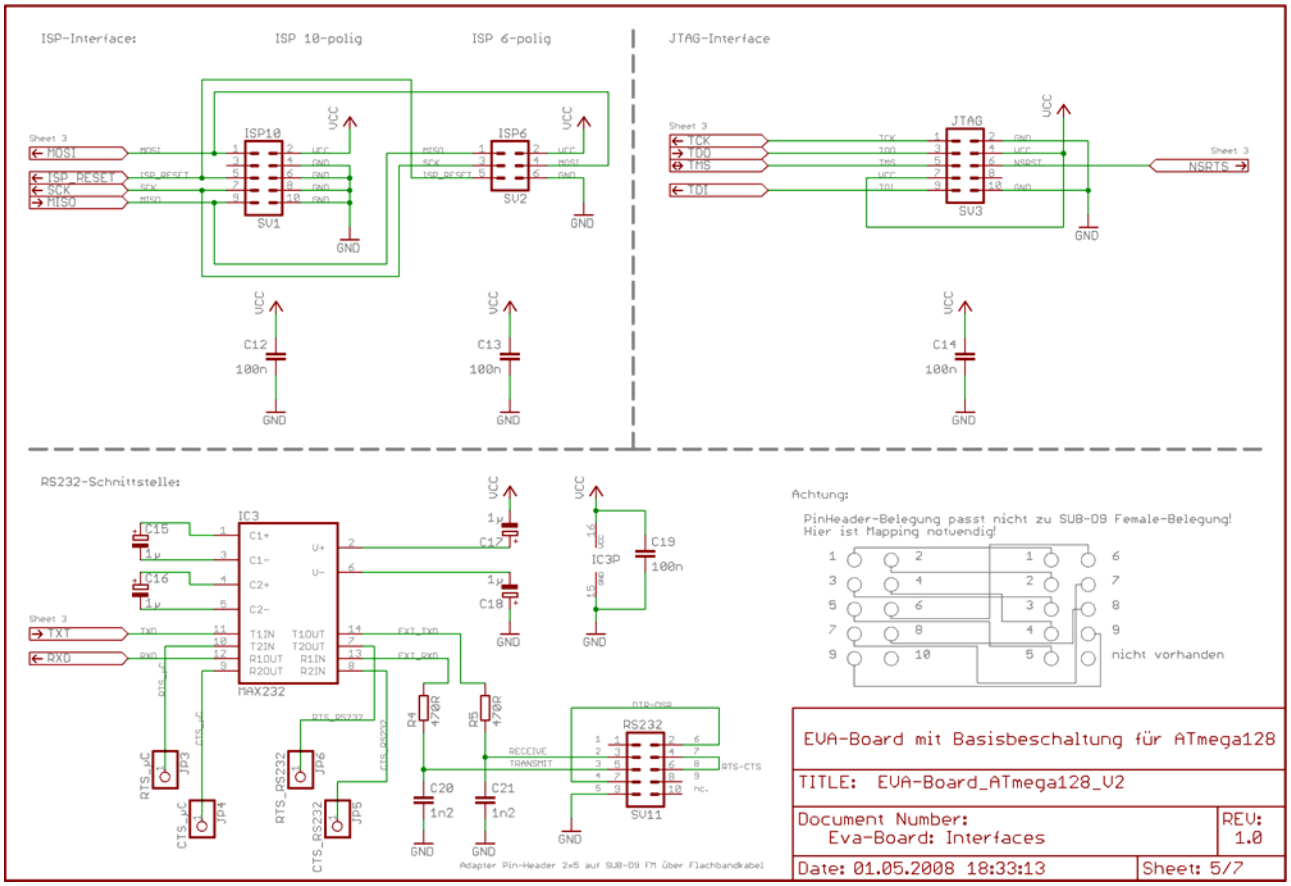
Schaltbild 52: Schaltbild Atmega128 Evaluation-Board - Sheet 2



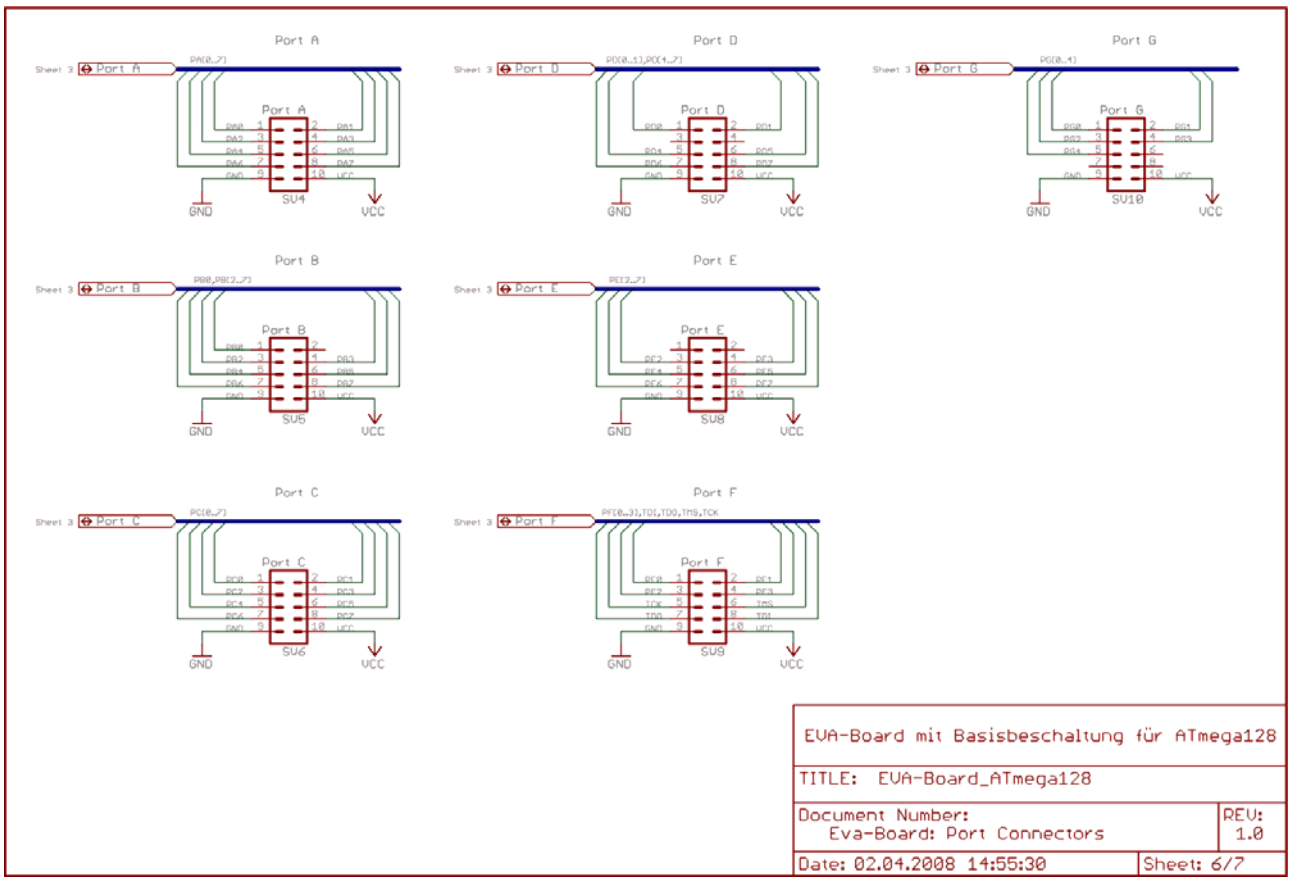
Schaltbild 53: Schaltbild Atmega128 Evaluation-Board - Sheet 3



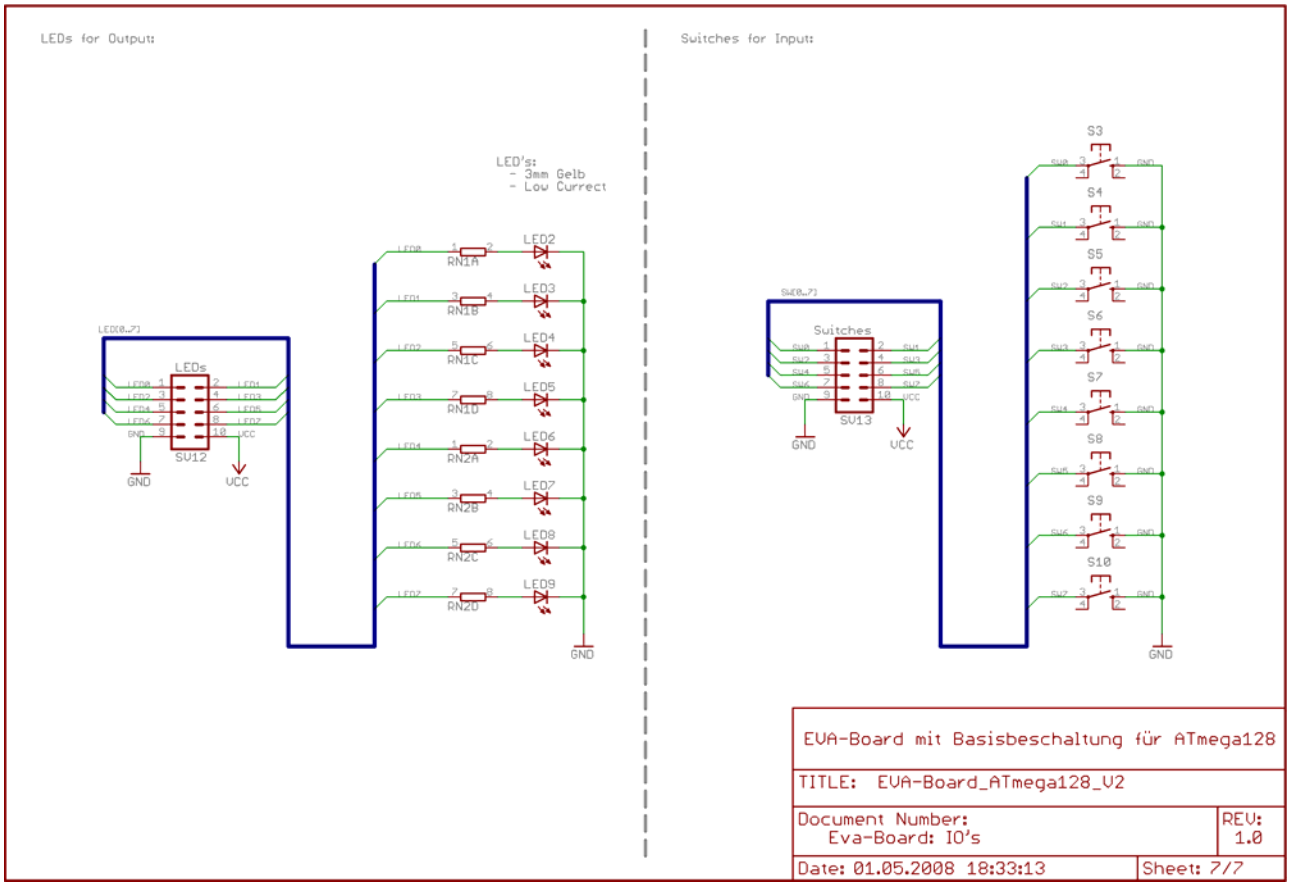
Schaltbild 54: Schaltbild Atmega128 Evaluation-Board - Sheet 4



Schaltbild 55: Schaltbild Atmega128 Evaluation-Board - Sheet 5



Schaltbild 56: Schaltbild Atmega128 Evaluation-Board - Sheet 6



Schaltbild 57: Schaltbild Atmega128 Evaluation-Board - Sheet 7

10.3.2 Layout Gesamt

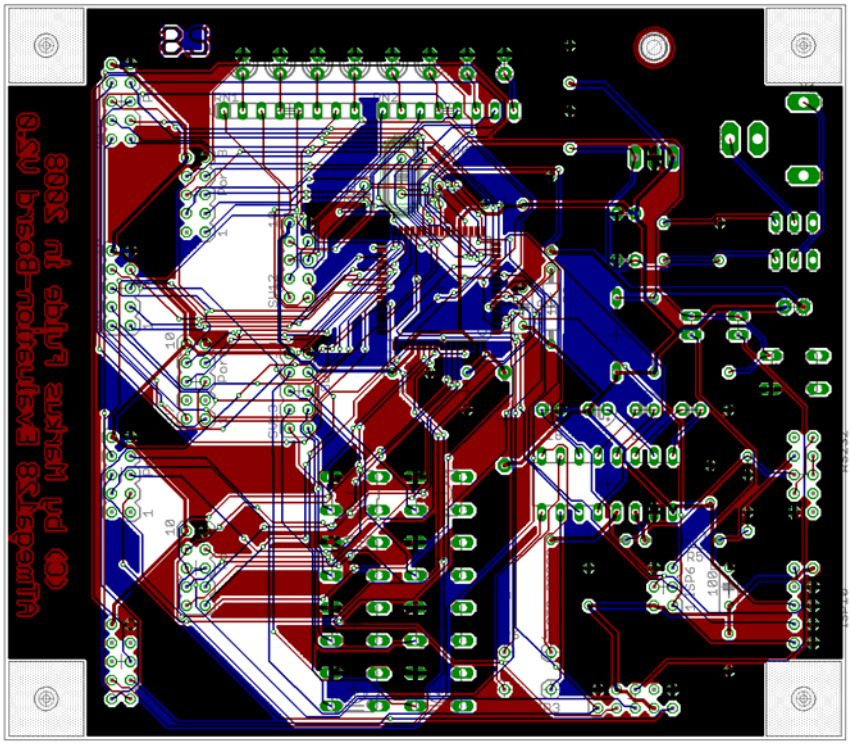


Abbildung 151: ATmega128 Evaluation-Board– Layout gesamt

10.3.3 Top-Layer

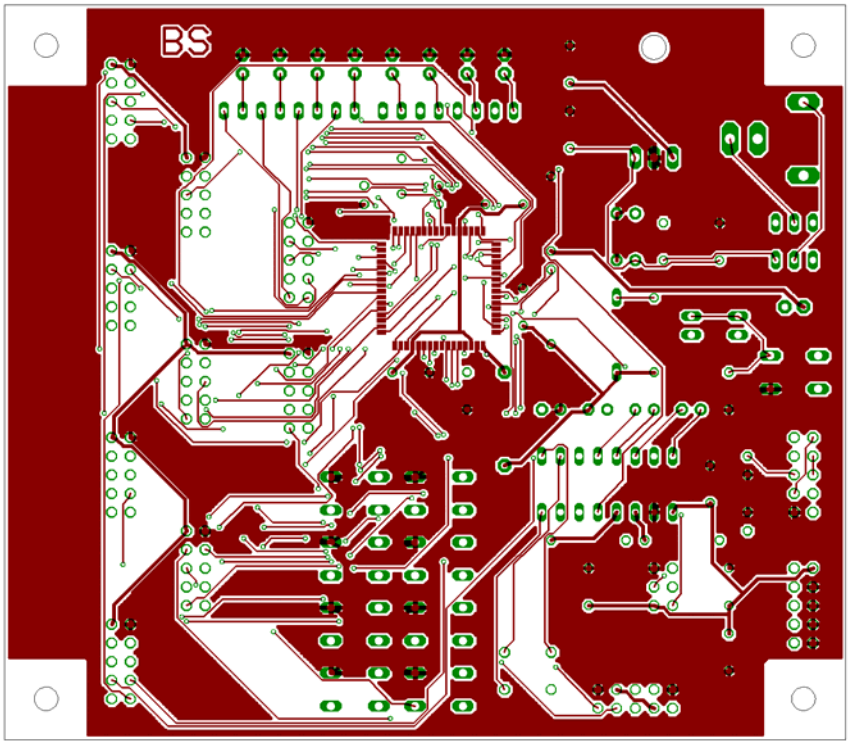


Abbildung 152: ATmega128 Evaluation-Board– Top-Layer

10.3.4 Bottom-Layer

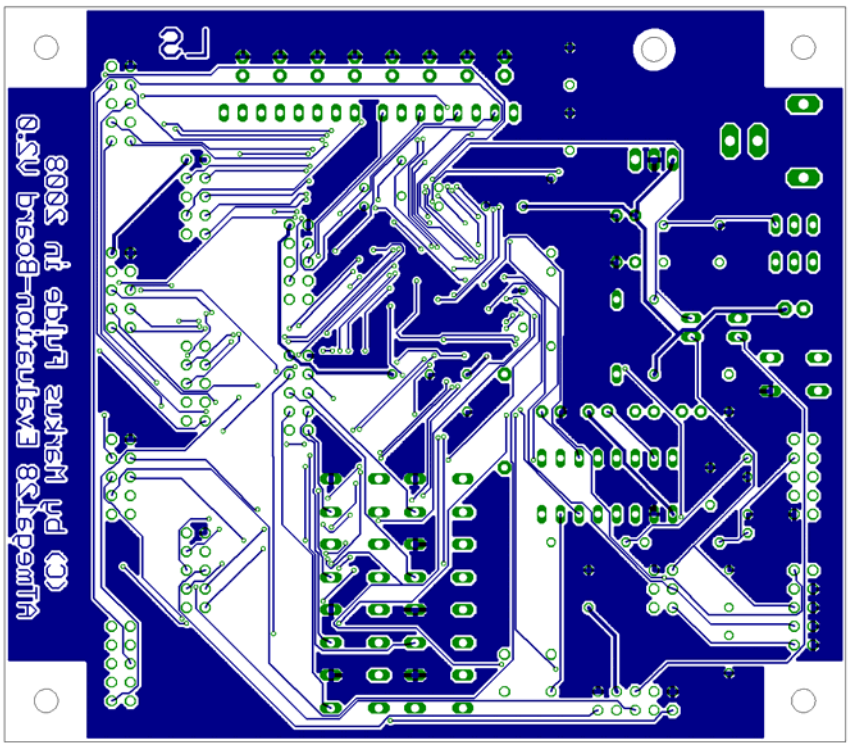


Abbildung 153: ATmega128 Evaluation-Board– Bottom-Layer

10.3.5 Pad's und Via's

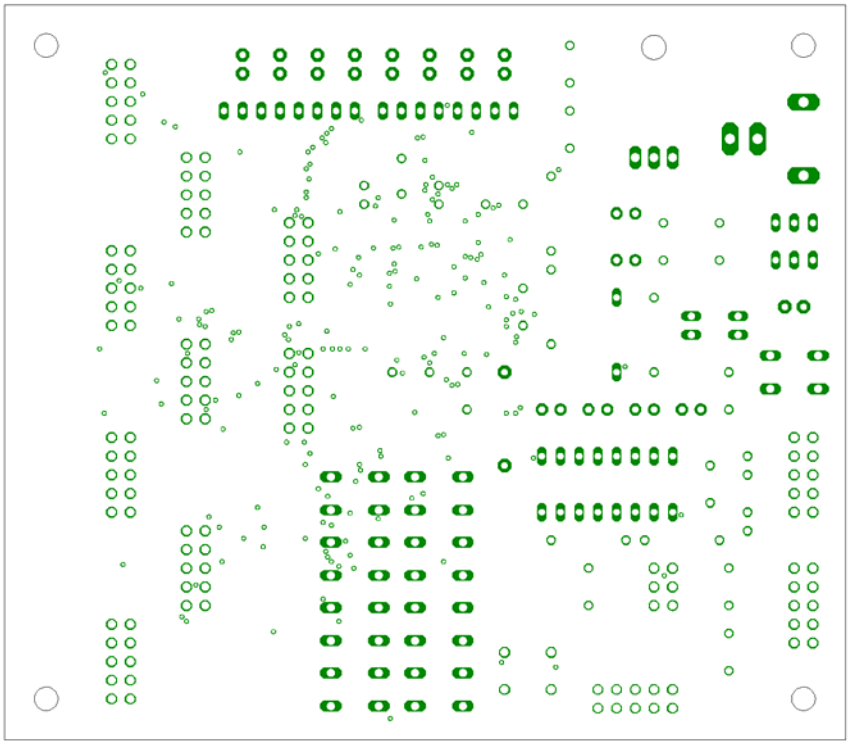


Abbildung 154: ATmega128 Evaluation-Board– Pad's und Via's

10.3.6 Bestückungsdruck

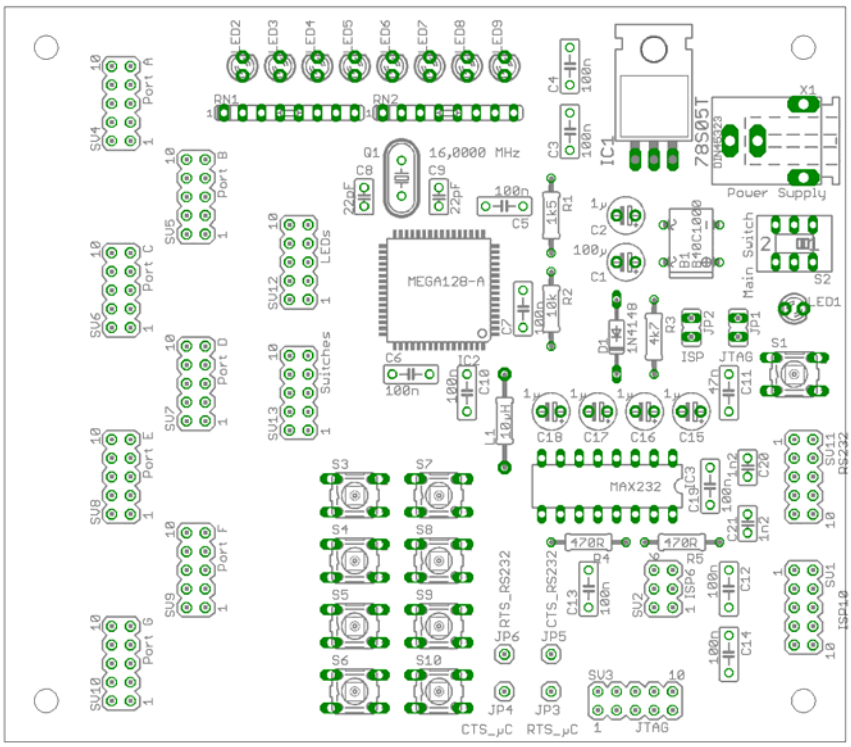


Abbildung 155: ATmega128 Evaluation-Board– Bestückung

10.3.7 Restricted Areas

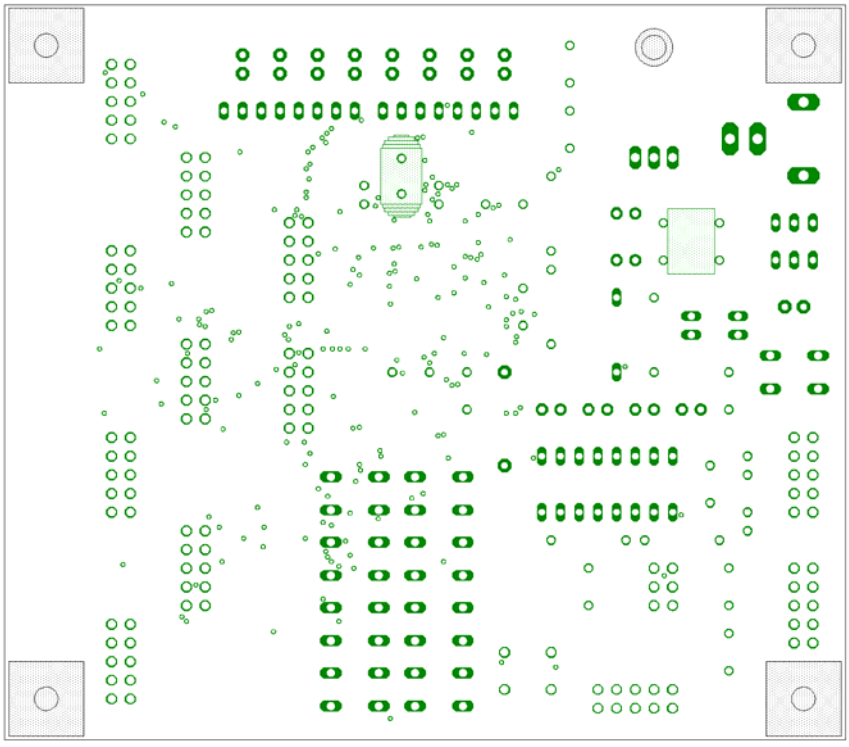


Abbildung 156: ATmega128 Evaluation-Board– Restricted Areas

10.3.8 Eagle-BOM

Pos.		Bauteil	Wert	Device	Package
1.		B1	B40C1000	B-DIL	B-DIL

2.		C1	100µ	POLARIZEDCAPACITORE2.5-5	E2,5-5
3.		C2	1µ	POLARIZEDCAPACITORE2.5-5	E2,5-5
4.		C3	100n	CAPACITOR050-025X075	C050-025X075
5.		C4	100n	CAPACITOR050-025X075	C050-025X075
6.		C5	100n	CAPACITOR050-025X075	C050-025X075
7.		C6	100n	CAPACITOR050-025X075	C050-025X075
8.		C7	100n	CAPACITOR050-025X075	C050-025X075
9.		C8	22pF	CAPACITOR025-025X050	C025-025X050
10.		C9	22pF	CAPACITOR025-025X050	C025-025X050
11.		C10	100n	CAPACITOR050-025X075	C050-025X075
12.		C11	47n	CAPACITOR050-025X075	C050-025X075
13.		C12	100n	CAPACITOR050-025X075	C050-025X075
14.		C13	100n	CAPACITOR050-025X075	C050-025X075
15.		C14	100n	CAPACITOR050-025X075	C050-025X075
16.		C15	1µ	POLARIZEDCAPACITORE2.5-5	E2,5-5
17.		C16	1µ	POLARIZEDCAPACITORE2.5-5	E2,5-5
18.		C17	1µ	POLARIZEDCAPACITORE2.5-5	E2,5-5
19.		C18	1µ	POLARIZEDCAPACITORE2.5-5	E2,5-5
20.		C19	100n	CAPACITOR050-025X075	C050-025X075
21.		C20	1n2	CAPACITOR025-025X050	C025-025X050
22.		C21	1n2	CAPACITOR025-025X050	C025-025X050
23.		D1	1N4148	1N4148	DO35-10
24.		IC1	78S05T	7805T	TO220H
25.		IC2	MEGA128-A	MEGA128-A	TQFP64
26.		IC3	MAX232	MAX232	DIL16
27.		JP1		JP1E	JP1
28.		JP2		JP1E	JP1
29.		JP3	RTS_µC	PINHD-1X1	1X01
30.		JP4	CTS_µC	PINHD-1X1	1X01
31.		JP5	CTS_RS232	PINHD-1X1	1X01
32.		JP6	RTS_RS232	PINHD-1X1	1X01
33.		L1	10µH	INDUCTOR0207/12	0207/12
34.		LED1		LED3MM	LED3MM
35.		LED2		LED3MM	LED3MM
36.		LED3		LED3MM	LED3MM
37.		LED4		LED3MM	LED3MM
38.		LED5		LED3MM	LED3MM
39.		LED6		LED3MM	LED3MM
40.		LED7		LED3MM	LED3MM
41.		LED8		LED3MM	LED3MM
42.		LED9		LED3MM	LED3MM
43.		Q1	16,0000 MHz	CRYTALHC49U-V	HC49U-V
44.		R1	1k5	RESISTOR0207/10	0207/10
45.		R2	10k	RESISTOR0207/10	0207/10
46.		R3	4k7	RESISTOR0207/10	0207/10
47.		R4	470R	RESISTOR0207/10	0207/10
48.		R5	470R	RESISTOR0207/10	0207/10
49.		RN1		E4R	SIL8
50.		RN2		E4R	SIL8

51.		S1		MJTP1230	MJTP1230
52.		S2	Main Switch	A68-A31	A68-A31
53.		S3		MJTP1230	MJTP1230
54.		S4		MJTP1230	MJTP1230
55.		S5		MJTP1230	MJTP1230
56.		S6		MJTP1230	MJTP1230
57.		S7		MJTP1230	MJTP1230
58.		S8		MJTP1230	MJTP1230
59.		S9		MJTP1230	MJTP1230
60.		S10		MJTP1230	MJTP1230
61.		SV1	ISP10	MA05-2	MA05-2
62.		SV2	ISP6	MA03-2	MA03-2
63.		SV3	JTAG	MA05-2	MA05-2
64.		SV4	Port A	MA05-2	MA05-2
65.		SV5	Port B	MA05-2	MA05-2
66.		SV6	Port C	MA05-2	MA05-2
67.		SV7	Port D	MA05-2	MA05-2
68.		SV8	Port E	MA05-2	MA05-2
69.		SV9	Port F	MA05-2	MA05-2
70.		SV10	Port G	MA05-2	MA05-2
71.		SV11	RS232	MA05-2	MA05-2
72.		SV12	LEDs	MA05-2	MA05-2
73.		SV13	Switches	MA05-2	MA05-2
74.		X1	Power Supply	737992-55	737992-5

Tabelle 148: Eagle BOM für Evaluation-Board

Die folgende Tabelle zeigt die Bauteilspezialitäten für das ATmega128 Evaluation-Board:

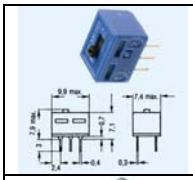
	T 681 Schiebeschalter-Miniatur, 2x UM
	SCHALTBUCHSE PRINTMONTAGE Conrad: 737992-13
	DC-STECKER 2,1MM / 5,0MM / 9,5MM GERADE Conrad: 530020 - 62
	Brückengleichrichter B40C1000=B40D Conrad: 501204 - 62

Tabelle 149: Bauteilspezialitäten ATmega128 Evalutation Board

10.3.9 Das EVA-Board

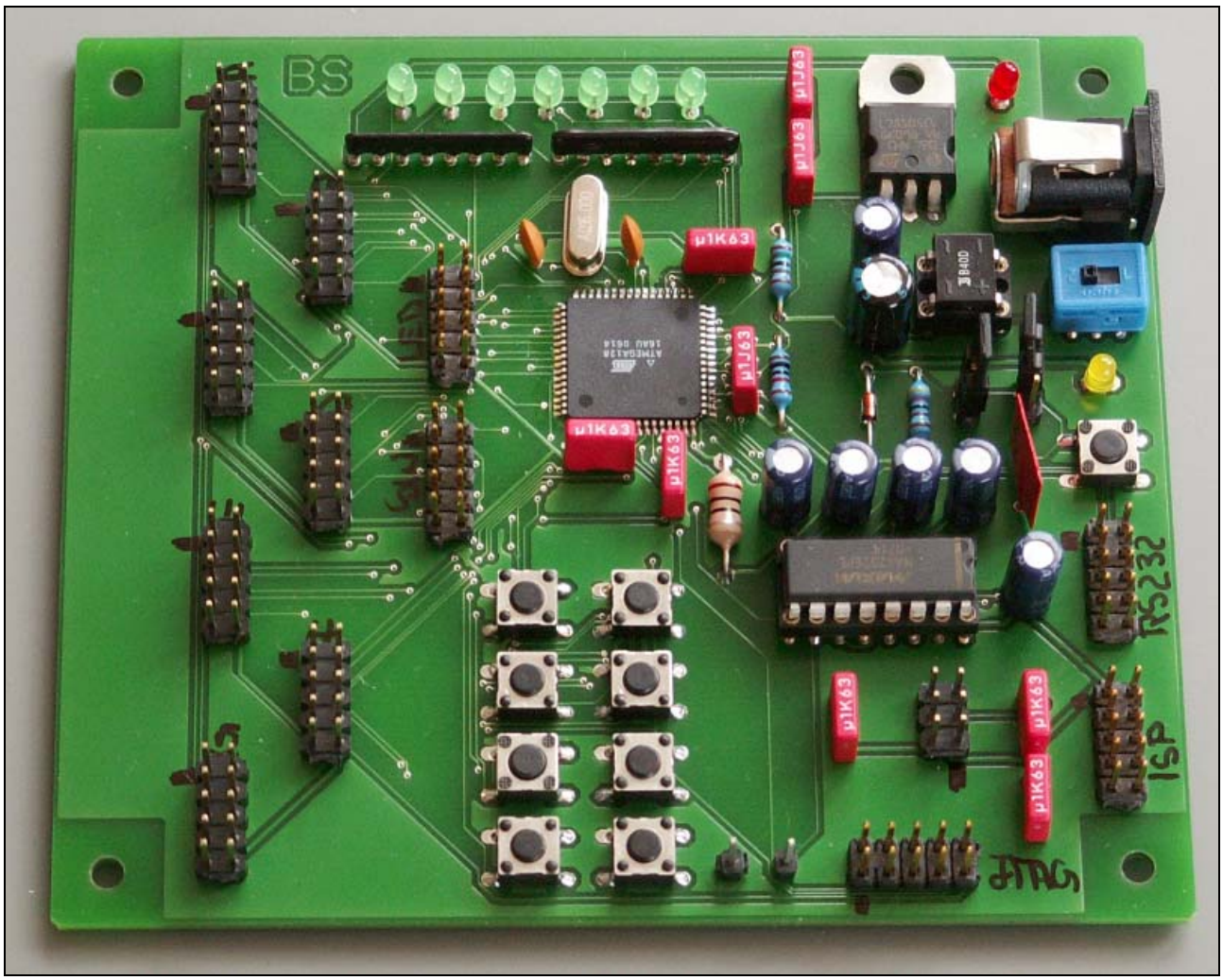


Abbildung 157: Das ATmega128 EVA-Board

10.3.10 Testprogramm für Erstinbetriebnahme

```
$regfile = "m128def.dat"
$crystal = 16000000

'Config Com1 = 38400 , Synchrone = 0 , Parity = None , Stopbits = 1 , Databits = 8 , Clockpol = 0
Config Com2 = 38400 , Synchrone = 0 , Parity = None , Stopbits = 1 , Databits = 8 , Clockpol = 0

'Open "com1:" For Binary As #1
Open "com2:" For Binary As #2

Config Porta = Output
Porta = &B00000000

Dim Count As Word
Count = 0

Do
'    Print #1 , "test serial port 1"
    Print #2 , Count ; ": test serial port 2"
    Porta = Count
    Wait 1
    Incr Count
Loop

Close #2
'Close #1

End
```

Software 14: EVA-Board COM2_TEST

```
$regfile = "m128def.dat"
$crystal = 16000000
```

```
$hwstack = 32
$swstack = 32
$framesize = 40

$baud = 38400


Taste_1 Alias Pinf.0
Taste_2 Alias Pinf.1
Taste_3 Alias Pinf.2
Taste_4 Alias Pinf.3
Taste_5 Alias Pinf.4
Taste_6 Alias Pinf.5
Taste_7 Alias Pinf.6
Taste_8 Alias Pinf.7

Config Portf = Input
Portf = &B11111111

Config Porta = Output
Porta = &B00000000

Do
   If Taste_1 = 0 Then
      Porta.0 = 1
   Else
      Porta.0 = 0
   End If

   If Taste_2 = 0 Then
      Porta.1 = 1
   Else
      Porta.1 = 0
   End If

   If Taste_3 = 0 Then
      Porta.2 = 1
   Else
      Porta.2 = 0
   End If

   If Taste_4 = 0 Then
      Porta.3 = 1
   Else
      Porta.3 = 0
   End If

   If Taste_5 = 0 Then
      Porta.4 = 1
   Else
      Porta.4 = 0
   End If

   If Taste_6 = 0 Then
      Porta.5 = 1
   Else
      Porta.5 = 0
   End If

   If Taste_7 = 0 Then
      Porta.6 = 1
   Else
      Porta.6 = 0
   End If

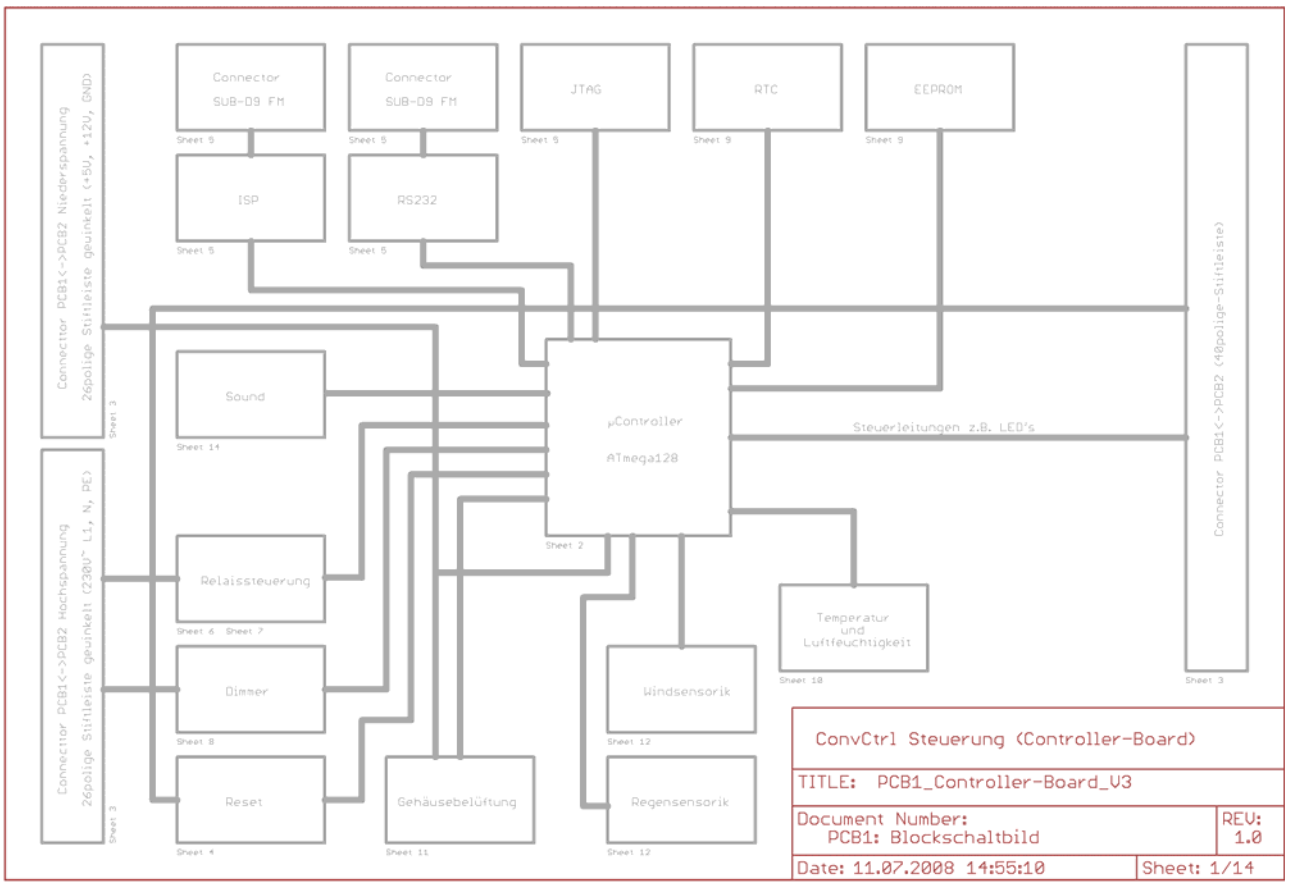
   If Taste_8 = 0 Then
      Porta.7 = 1
   Else
      Porta.7 = 0
   End If

Loop
```

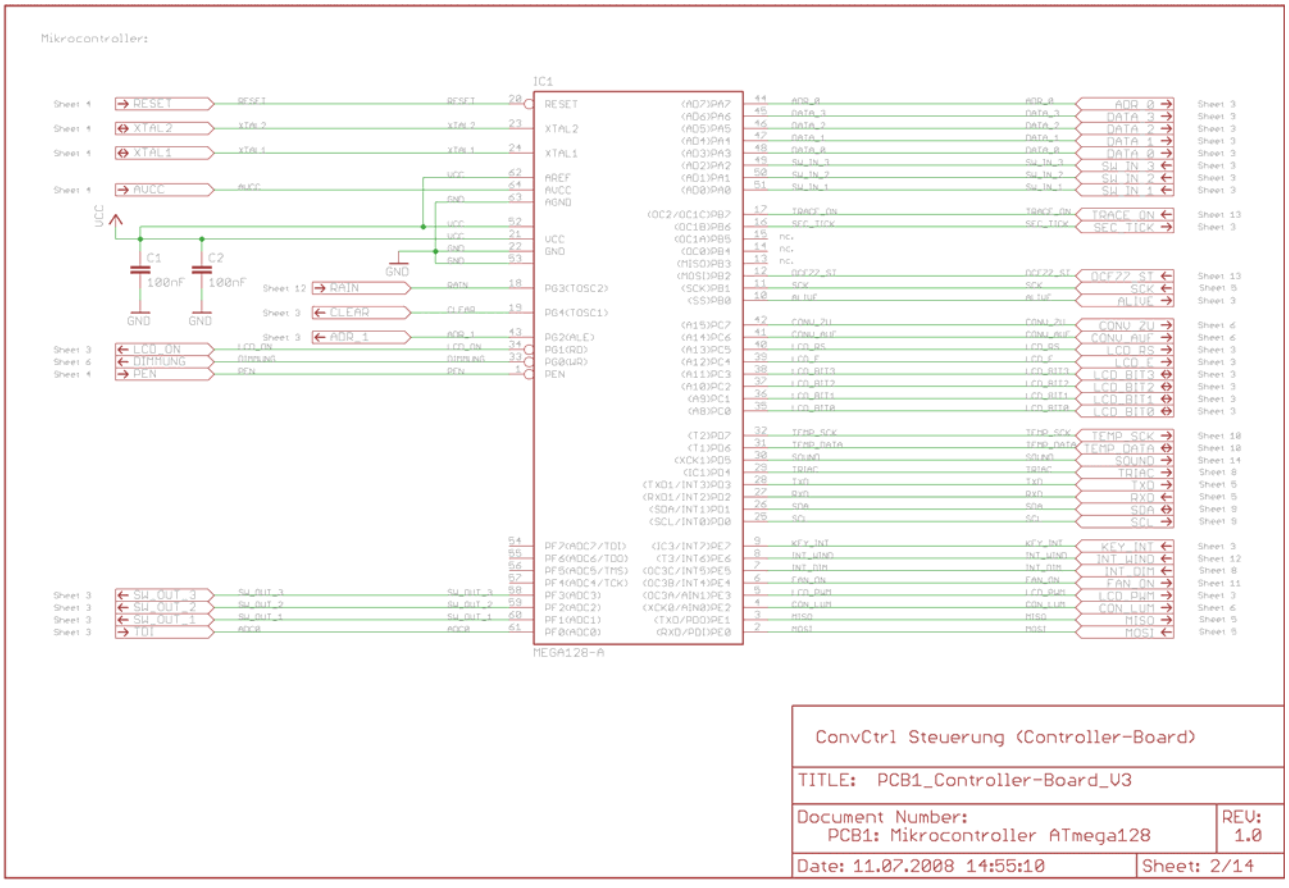
Software 15: EVA-Board IO_TEST

10.4 Das Controller-Board PCB1

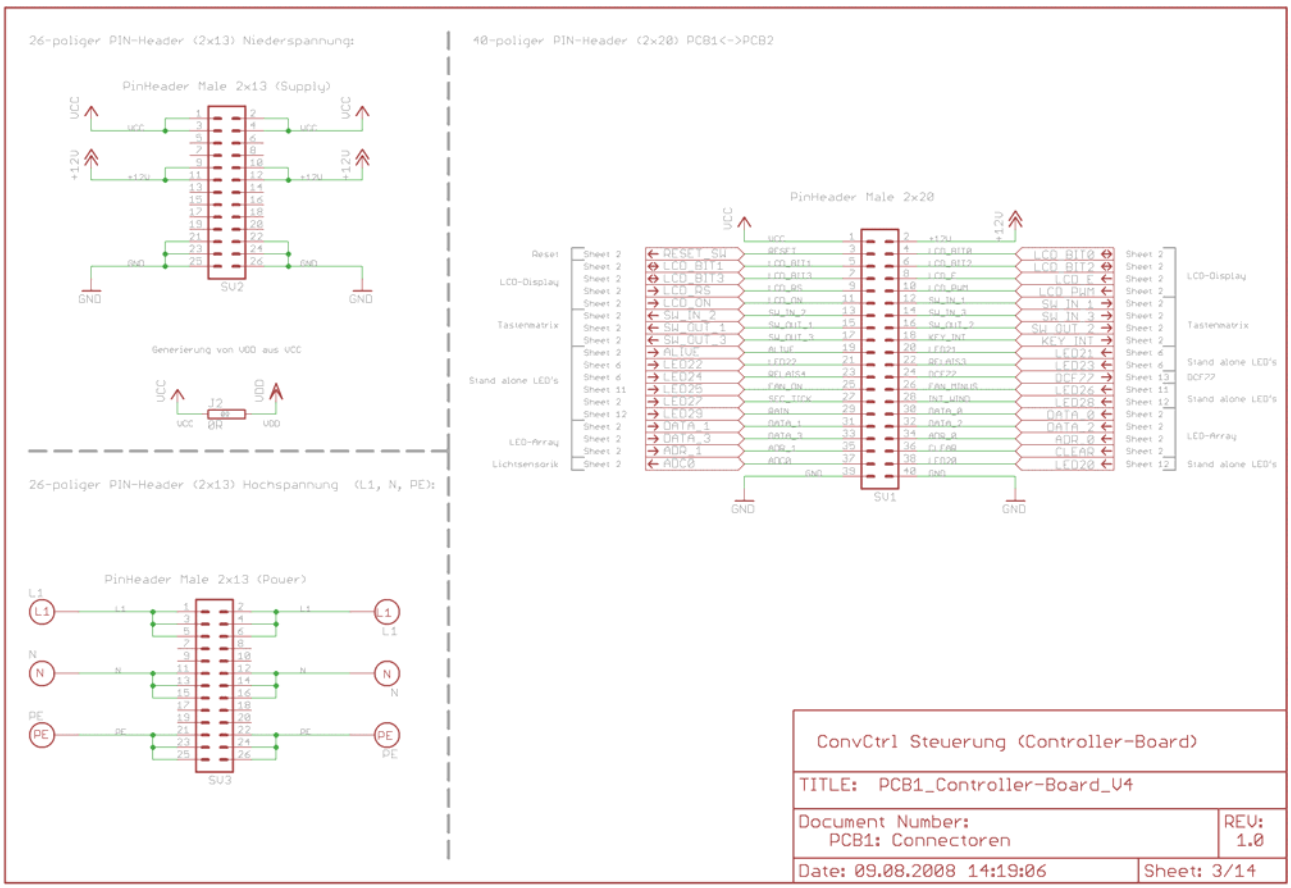
10.4.1 Schematic



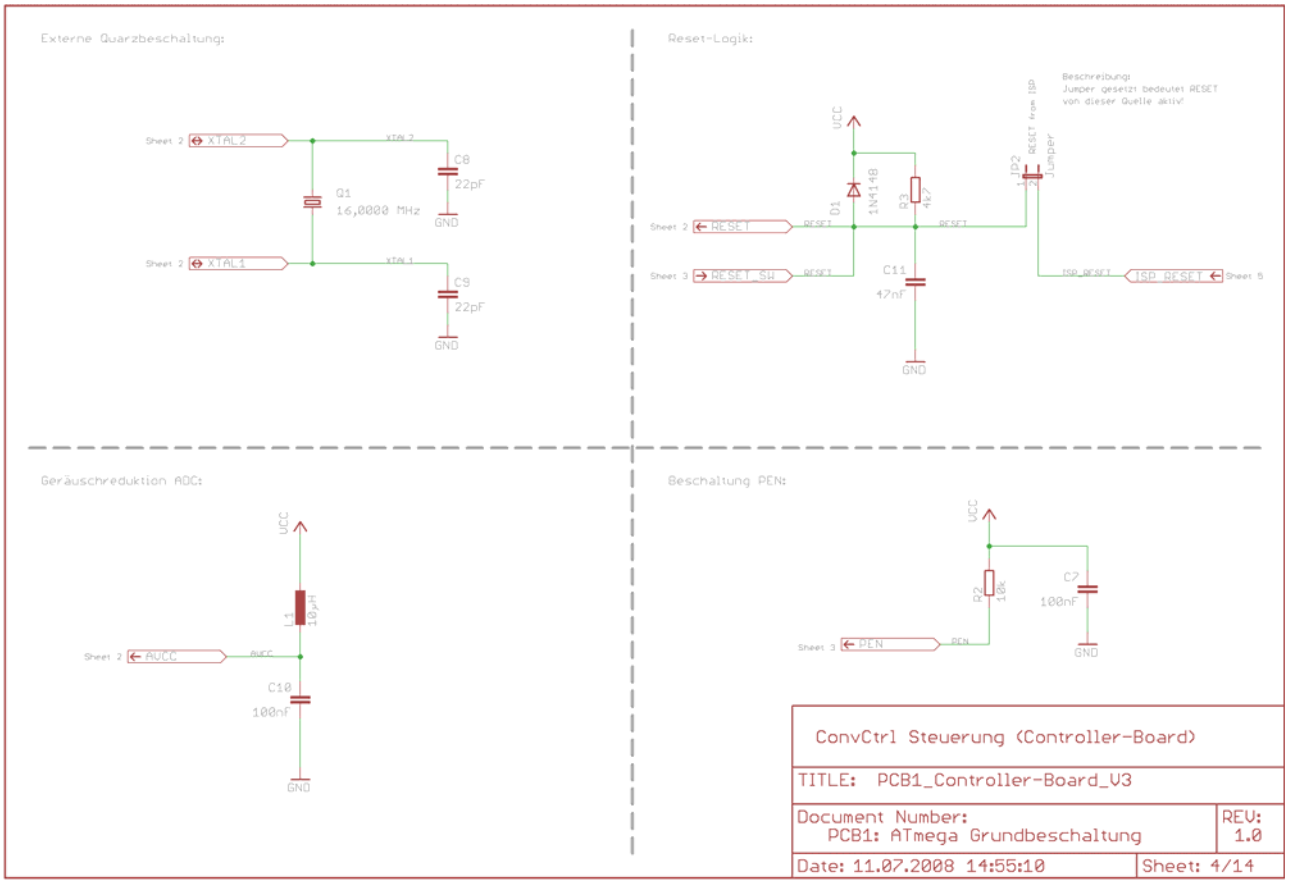
Schaltbild 58: Schaltbild PCB1 Controller-Board - Sheet 1



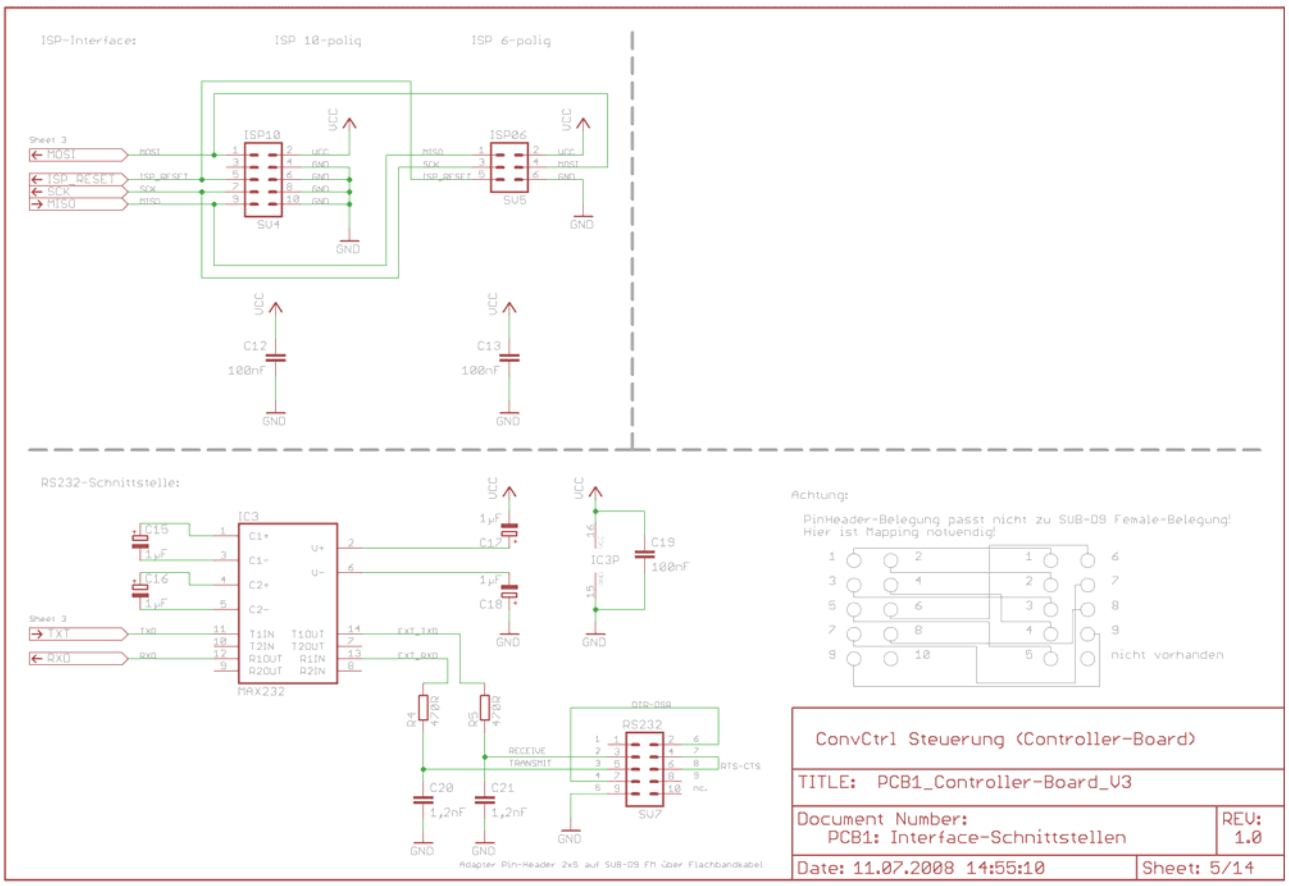
Schaltbild 59: Schaltbild PCB1 Controller-Board - Sheet 2



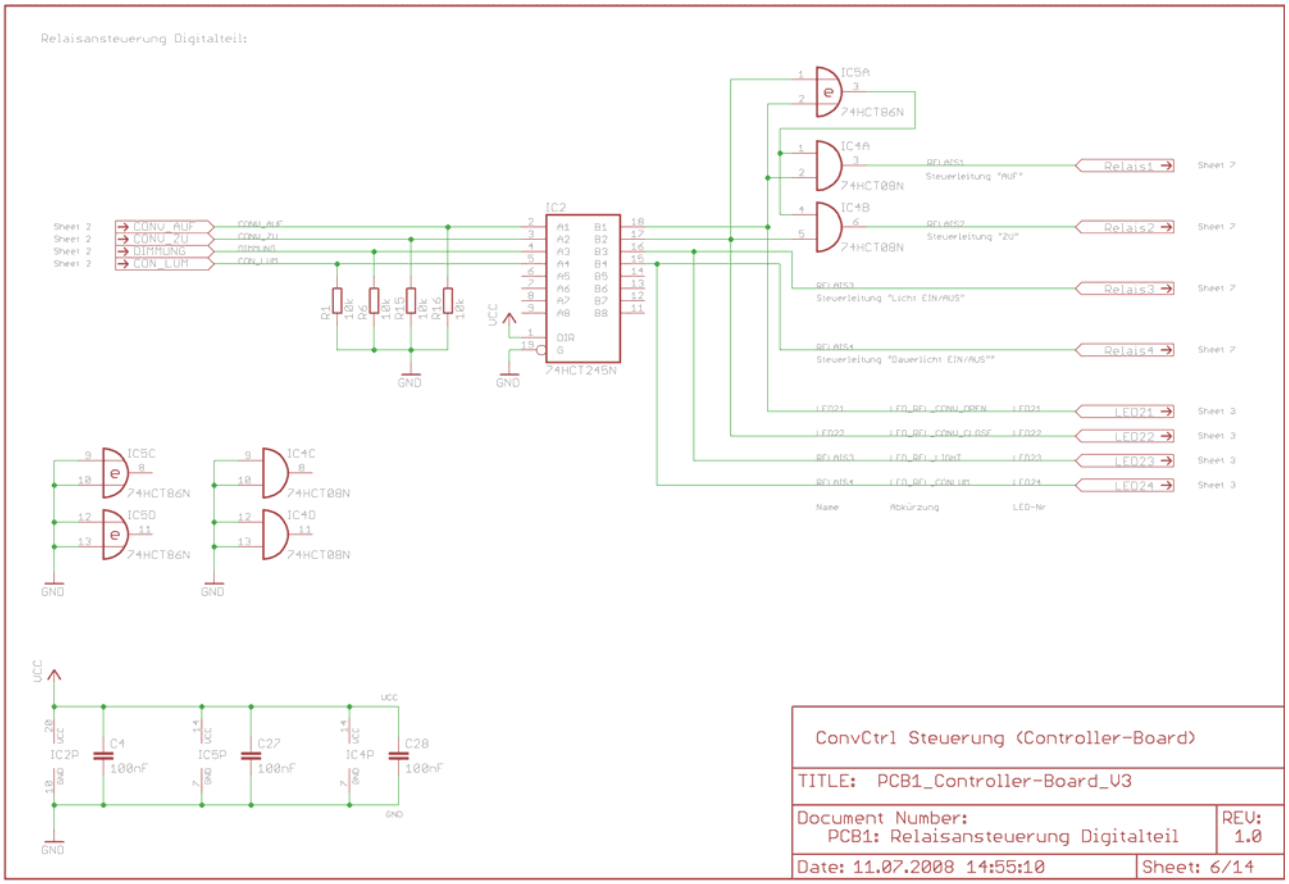
Schaltbild 60: Schaltbild PCB1 Controller-Board - Sheet 3



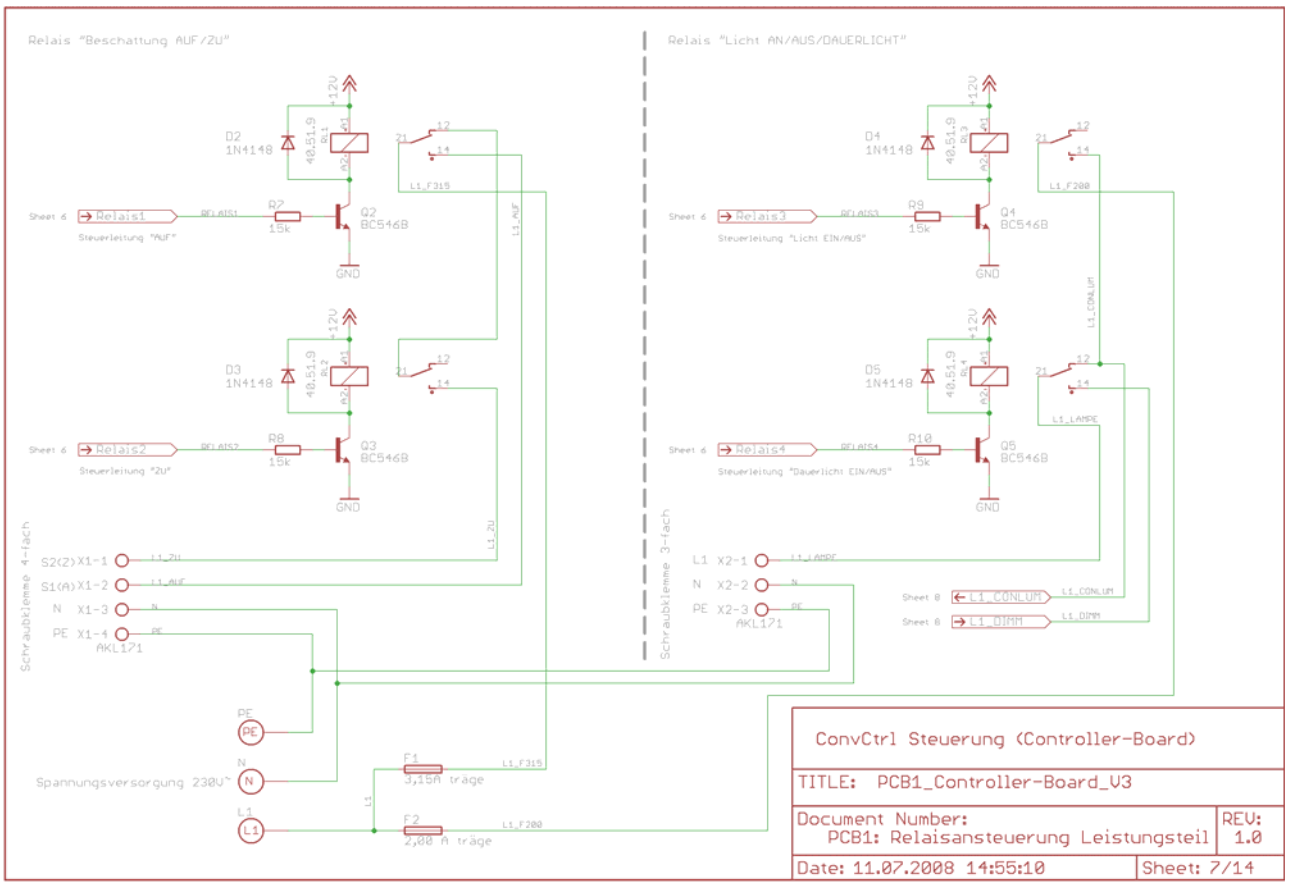
Schaltbild 61: Schaltbild PCB1 Controller-Board - Sheet 4



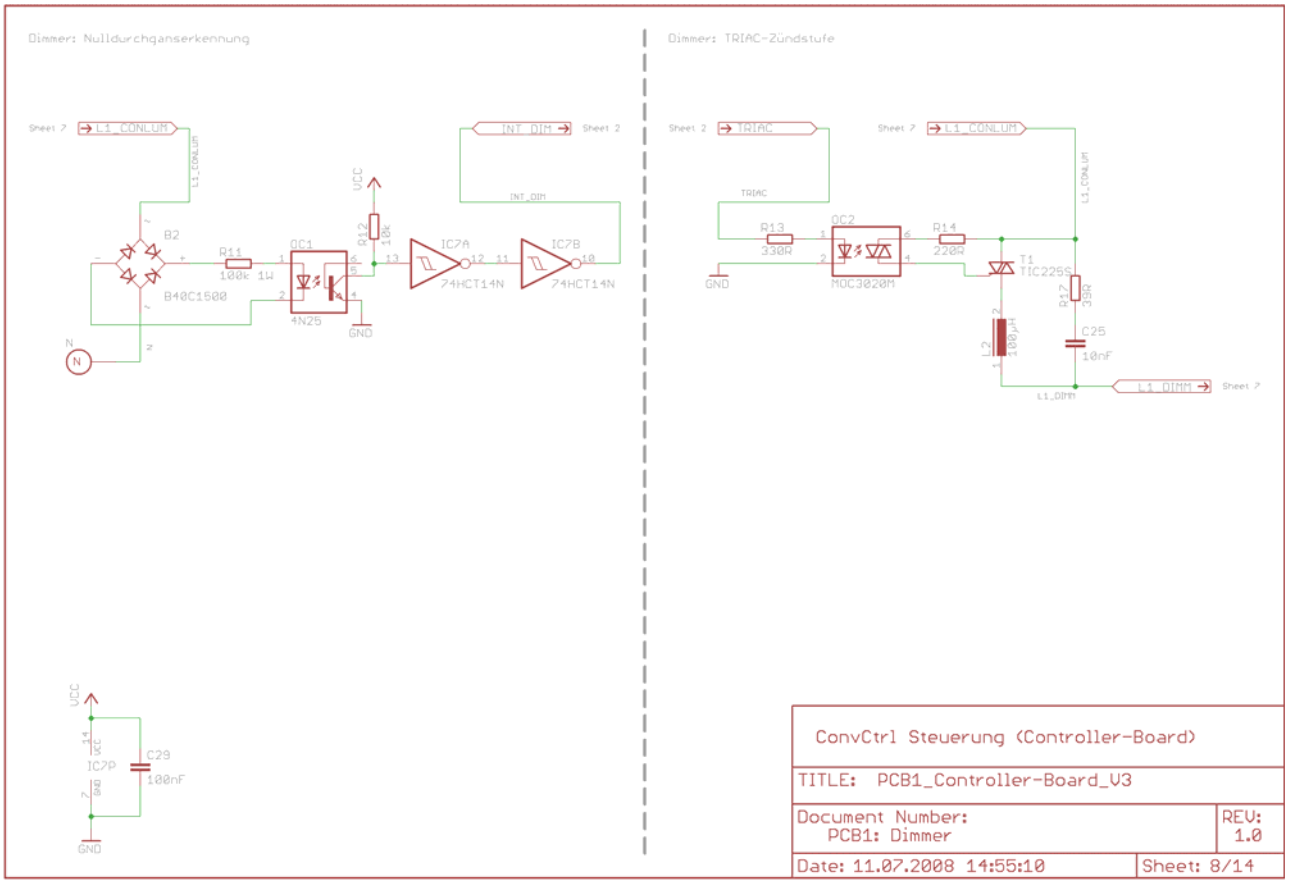
Schaltbild 62: Schaltbild PCB1 Controller-Board - Sheet 5



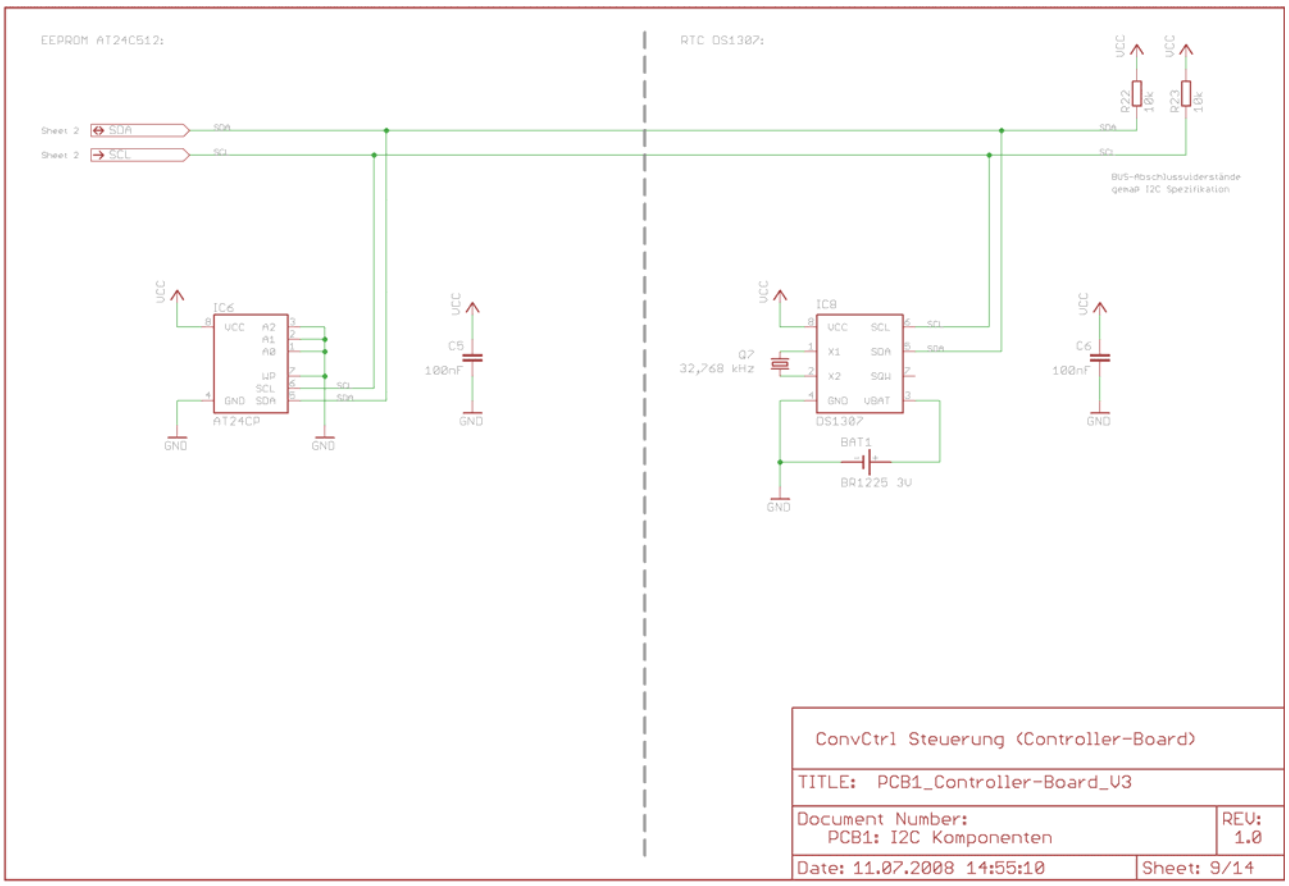
Schaltbild 63: Schaltbild PCB1 Controller-Board - Sheet 6



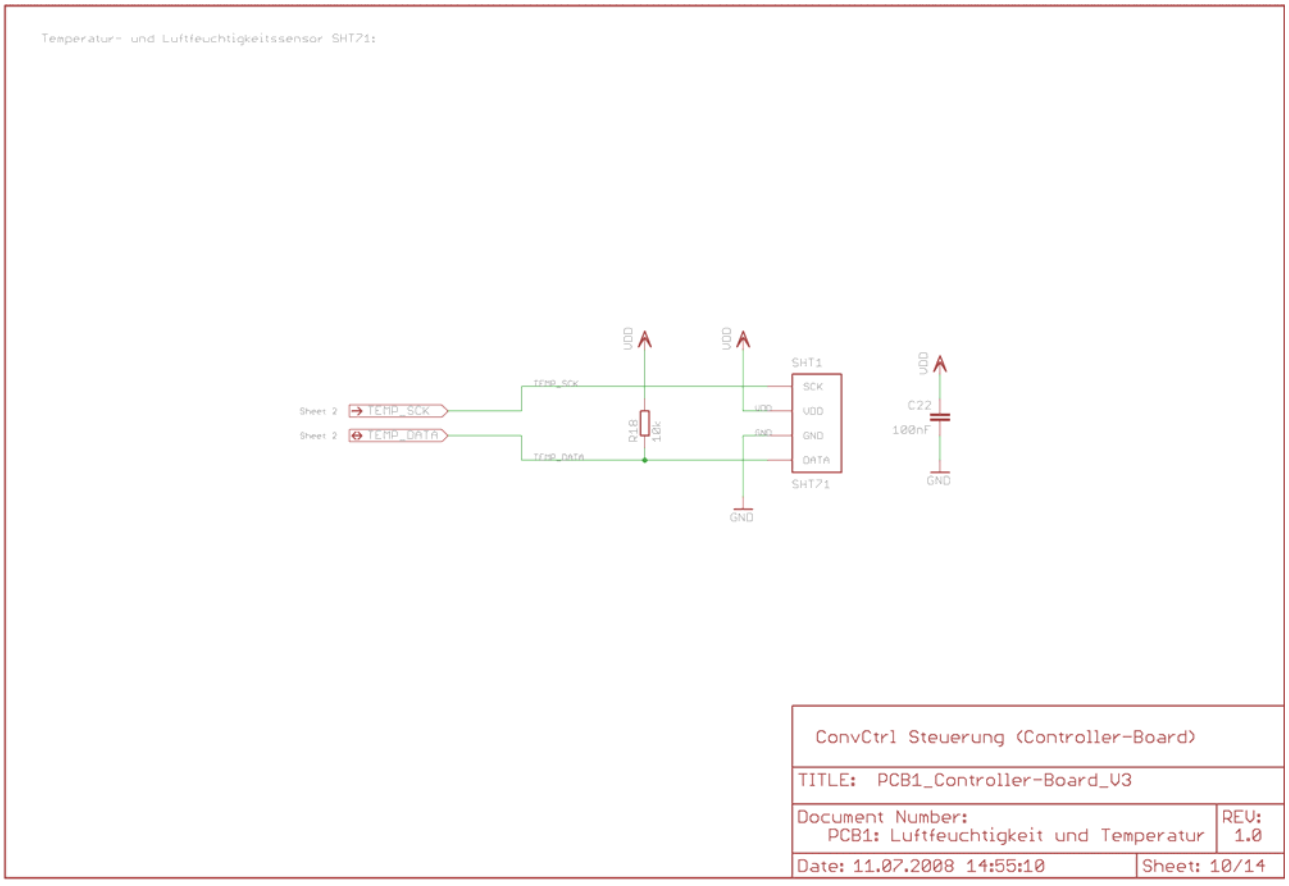
Schaltbild 64: Schaltbild PCB1 Controller-Board - Sheet 7



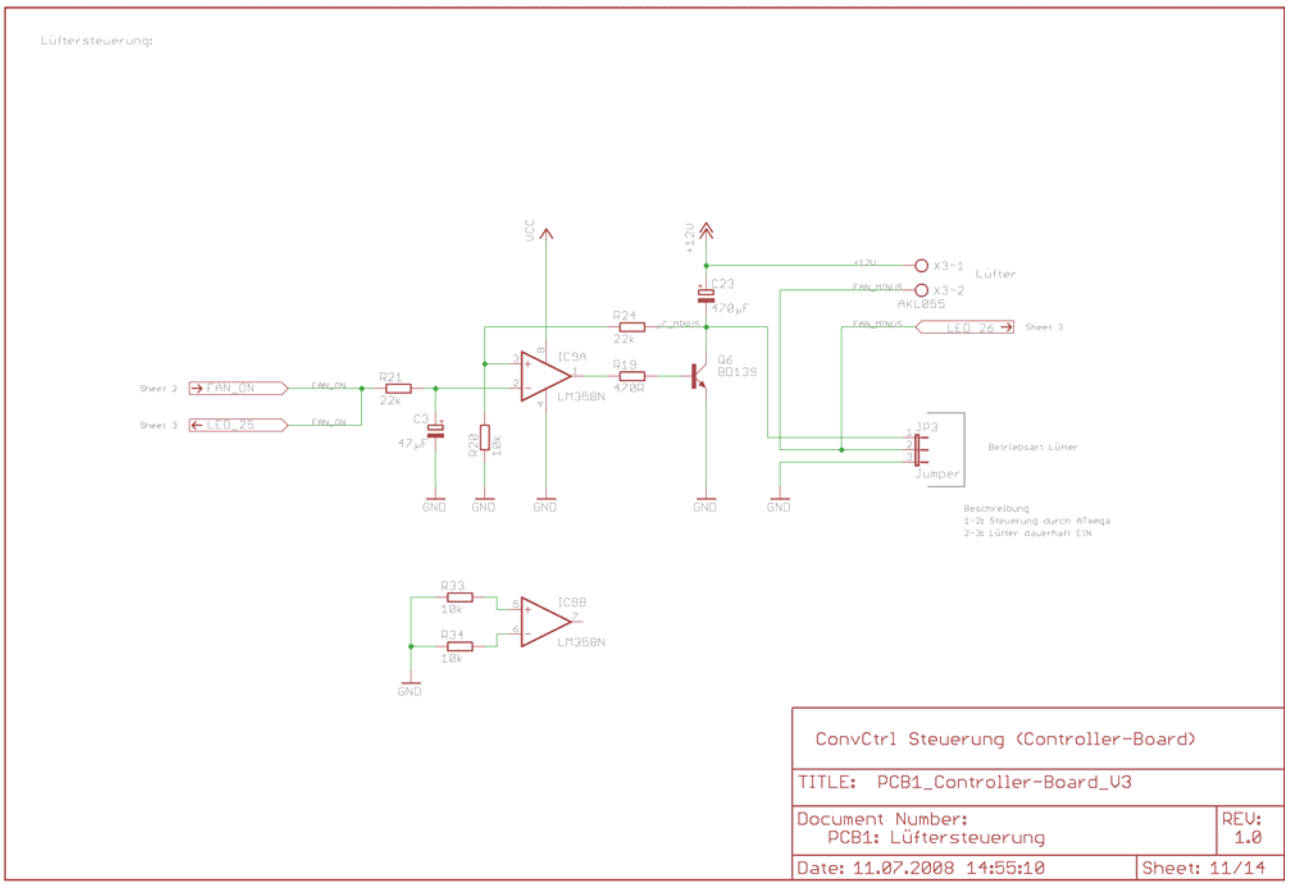
Schaltbild 65: Schaltbild PCB1 Controller-Board - Sheet 8



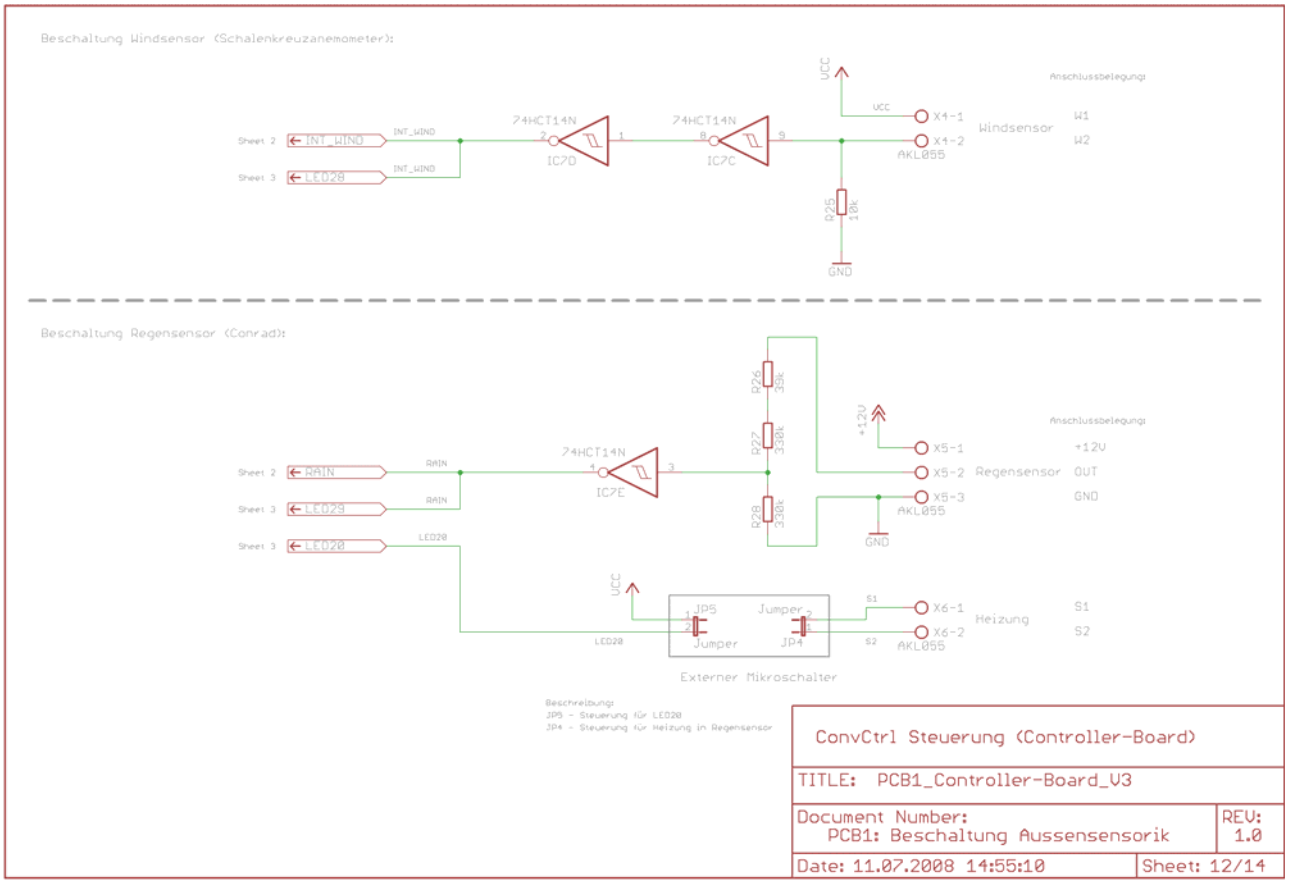
Schaltbild 66: Schaltbild PCB1 Controller-Board - Sheet 9



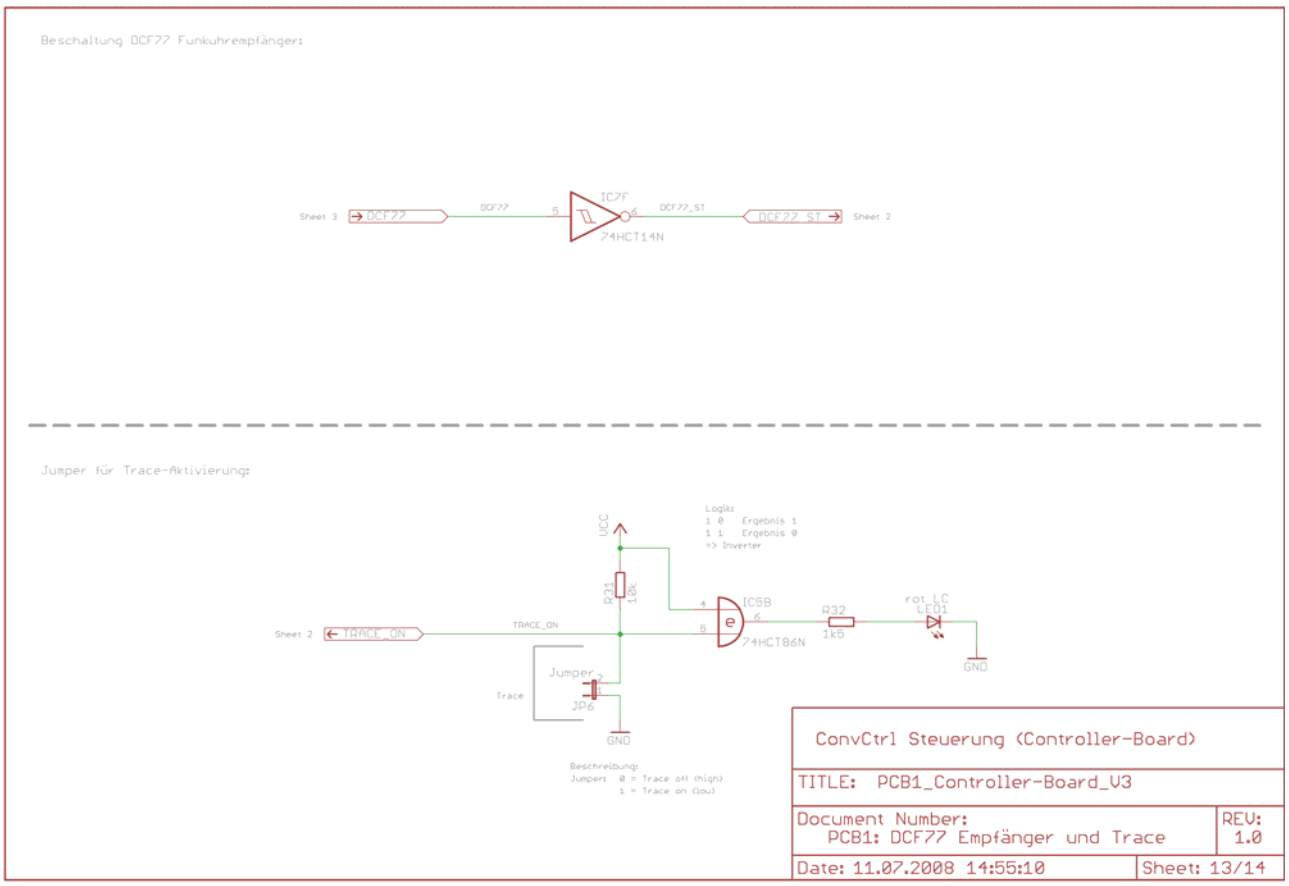
Schaltbild 67: Schaltbild PCB1 Controller-Board - Sheet 10



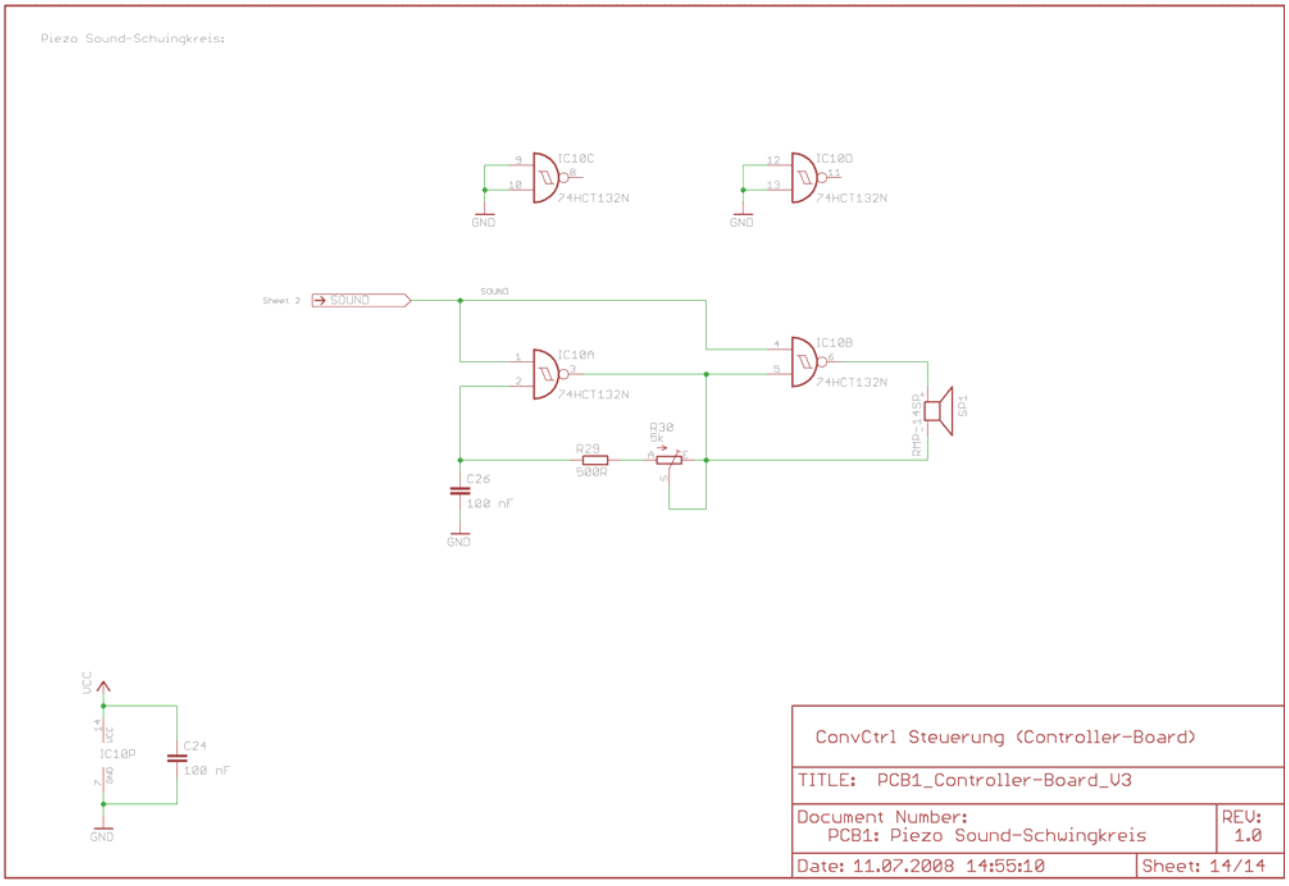
Schaltbild 68: Schaltbild PCB1 Controller-Board - Sheet 11



Schaltbild 69: Schaltbild PCB1 Controller-Board - Sheet 12



Schaltbild 70: Schaltbild PCB1 Controller-Board - Sheet 13



Schaltbild 71: Schaltbild PCB1 Controller-Board - Sheet 14

10.4.2 Layout Gesamt

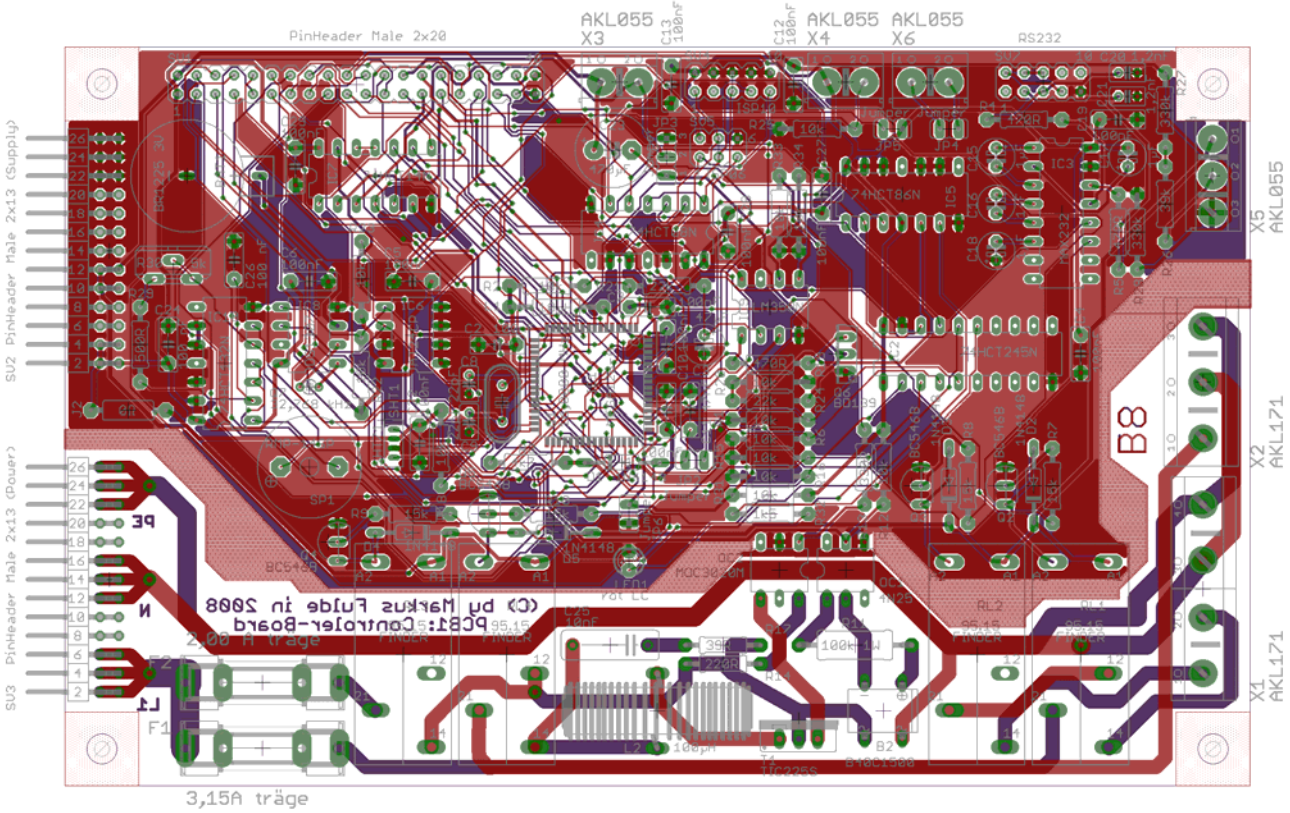


Abbildung 158: PCB1 Controler-Board– Layout gesamt

10.4.3 Top-Layer

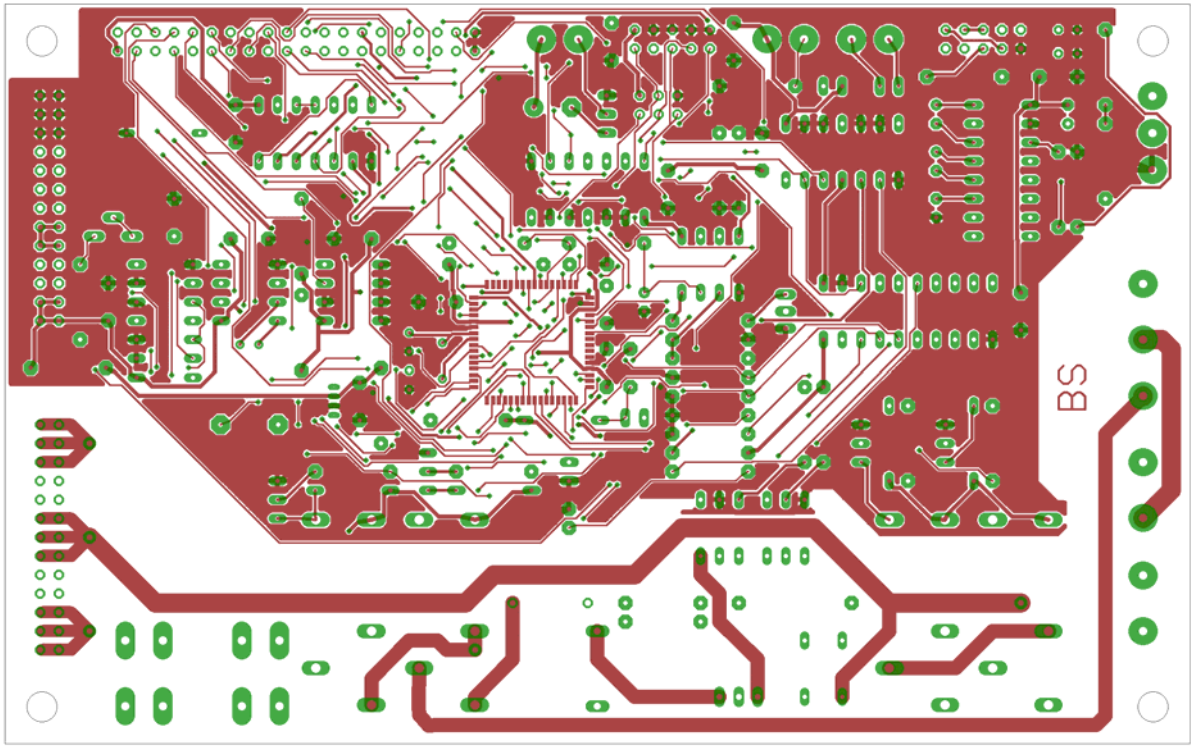


Abbildung 159: PCB1 Controller-Board– Top-Layer

10.4.4 Bottom-Layer

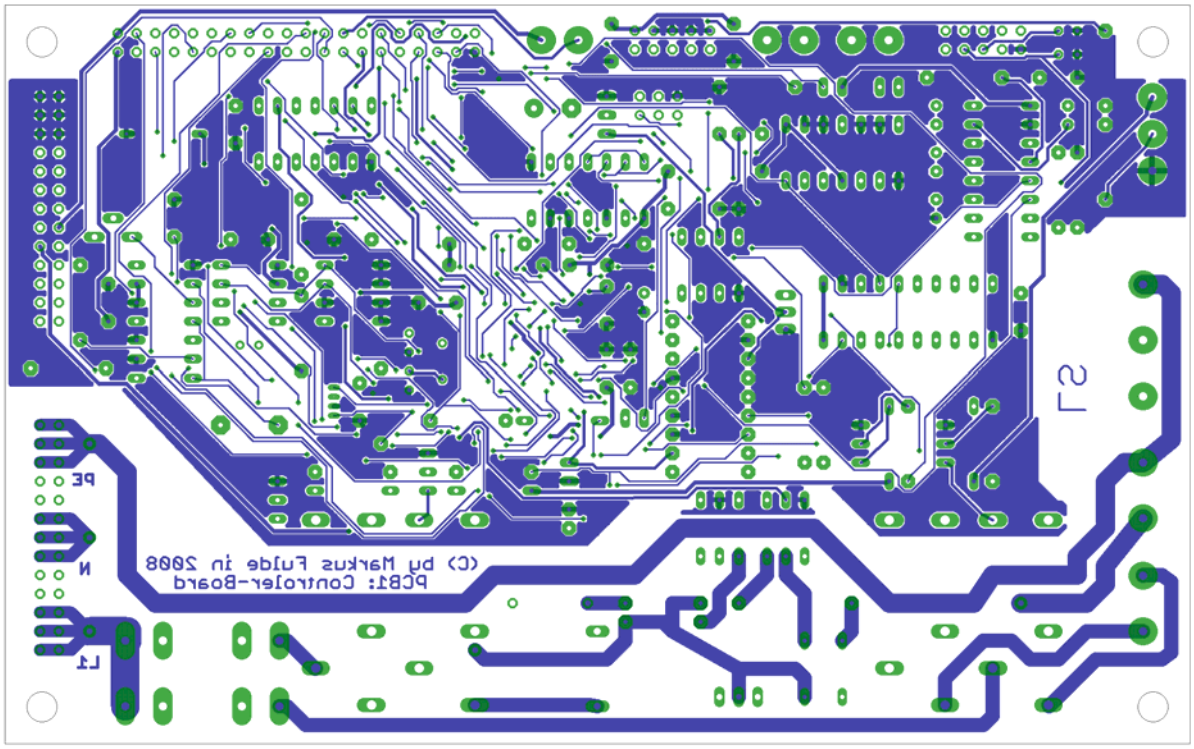


Abbildung 160: PCB1 Controler-Board– Bottom-Layer

10.4.5 Pad's und Via's

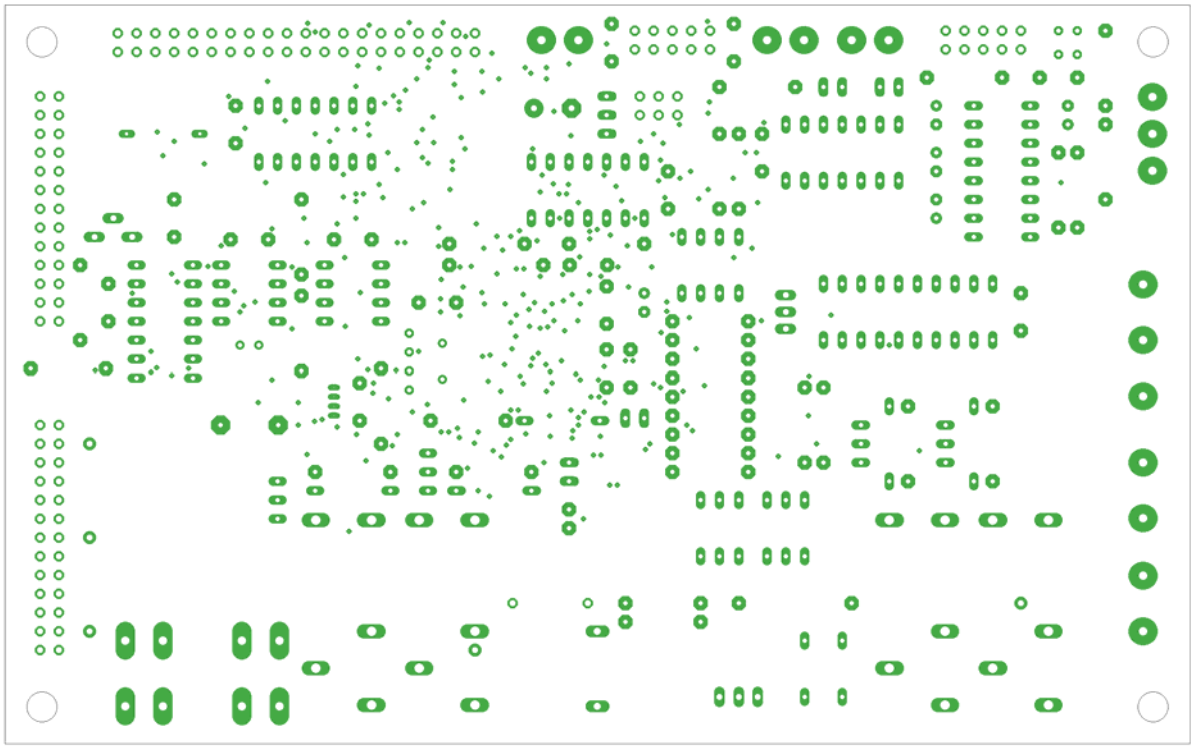


Abbildung 161: PCB1 Controler-Board– Pad's und Via's

10.4.6 Bestückungsdruck

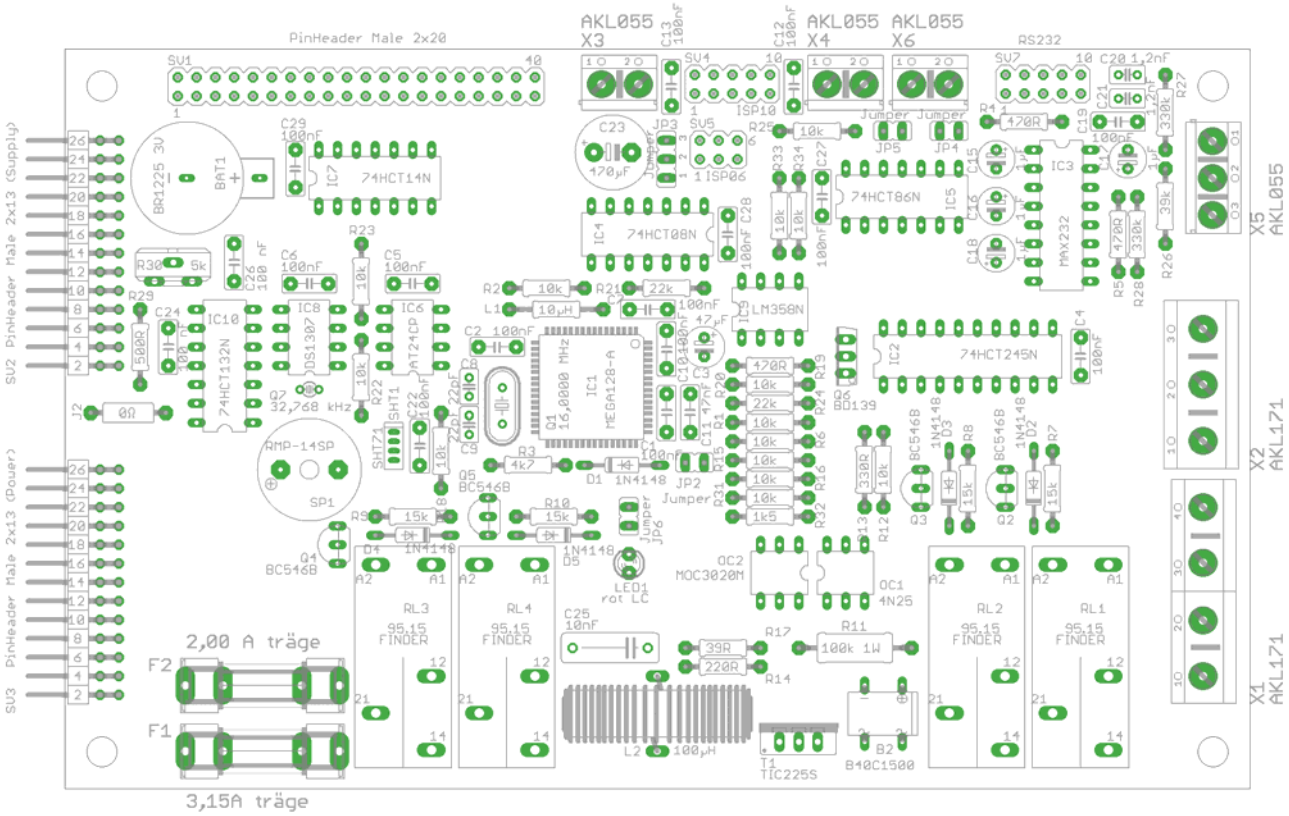


Abbildung 162: PCB1 Controler-Board– Bestückungsdruck

10.4.7 Restricted Areas

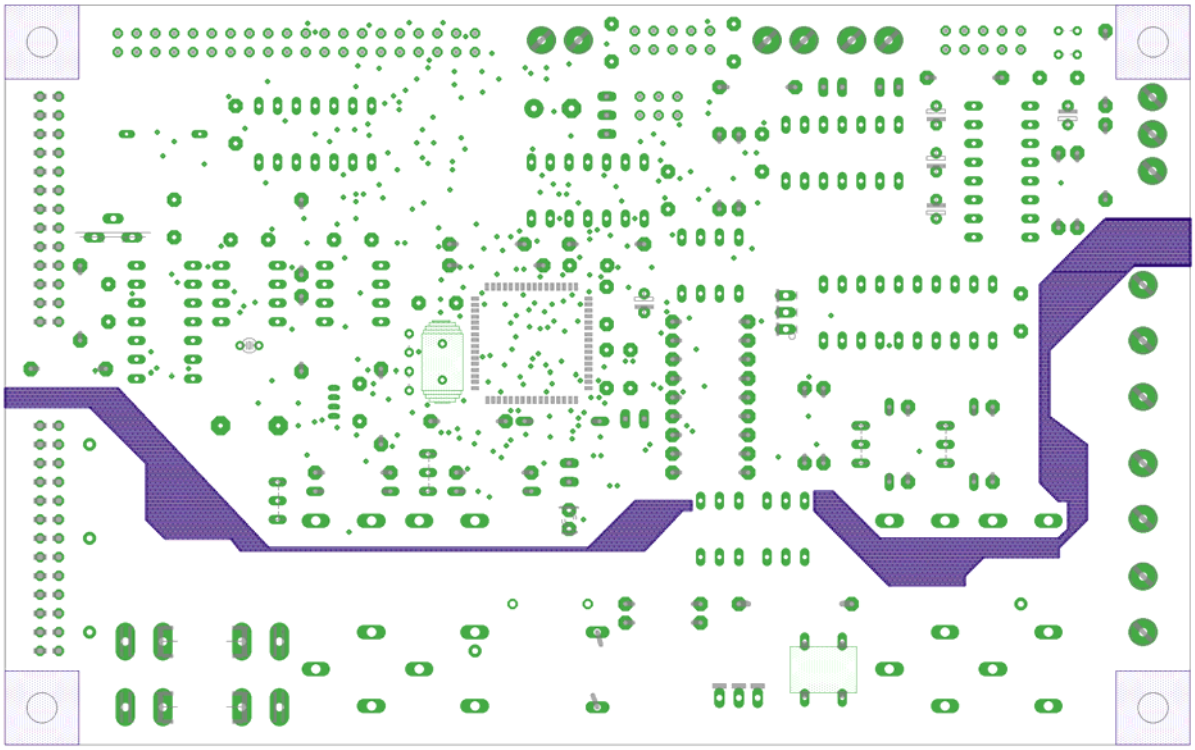


Abbildung 163: PCB1 Controler-Board– Restricted Areas

10.4.8 Eagle-BOM

Pos.		Bauteil	Wert	Device	Package
1.		B2	B40C1500	B250C1000	B-DIL
2.		BAT1	BR1225 3V	BR1225	BR1225
3.		C1	100nF	CAPACITOR050-025X075	C050-025X075
4.		C2	100nF	CAPACITOR050-025X075	C050-025X075
5.		C3	47µF	POLARIZEDCAPACITORE2.5-6	E2,5-6
6.		C4	100nF	CAPACITOR050-025X075	C050-025X075
7.		C5	100nF	CAPACITOR050-025X075	C050-025X075
8.		C6	100nF	CAPACITOR050-025X075	C050-025X075
9.		C7	100nF	CAPACITOR050-025X075	C050-025X075
10.		C8	22pF	CAPACITOR025-025X050	C025-025X050
11.		C9	22pF	CAPACITOR025-025X050	C025-025X050
12.		C10	100nF	CAPACITOR050-025X075	C050-025X075
13.		C11	47nF	CAPACITOR050-025X075	C050-025X075
14.		C12	100nF	CAPACITOR050-025X075	C050-025X075
15.		C13	100nF	CAPACITOR050-025X075	C050-025X075
16.		C15	1µF	POLARIZEDCAPACITORE2.5-5	E2,5-5
17.		C16	1µF	POLARIZEDCAPACITORE2.5-5	E2,5-5
18.		C17	1µF	POLARIZEDCAPACITORE2.5-5	E2,5-5
19.		C18	1µF	POLARIZEDCAPACITORE2.5-5	E2,5-5
20.		C19	100nF	CAPACITOR050-025X075	C050-025X075
21.		C20	1,2nF	CAPACITOR025-025X050	C025-025X050

22.		C21	1,2nF	CAPACITOR025-025X050	C025-025X050
23.		C22	100nF	CAPACITOR050-025X075	C050-025X075
24.		C23	470µF	POLARIZEDCAPACITORE5-10.5	E5-10,5
25.		C24	100 nF	CAPACITOR050-025X075	C050-025X075
26.		C25	10nF	CAPACITOR102-043X133	C102-043X133
27.		C26	100 nF	CAPACITOR050-025X075	C050-025X075
28.		C27	100nF	CAPACITOR050-025X075	C050-025X075
29.		C28	100nF	CAPACITOR050-025X075	C050-025X075
30.		C29	100nF	CAPACITOR050-025X075	C050-025X075
31.		D1	1N4148	1N4148	DO35-10
32.		D2	1N4148	1N4148	DO35-10
33.		D3	1N4148	1N4148	DO35-10
34.		D4	1N4148	1N4148	DO35-10
35.		D5	1N4148	1N4148	DO35-10
36.		F1	3,15A träge	SHK20L	SHK20L
37.		F2	2,00 A träge	SHK20L	SHK20L
38.		IC1	MEGA128-A	MEGA128-A	TQFP64
39.		IC2	74HCT245N	74HCT245N	DIL20
40.		IC3	MAX232	MAX232	DIL16
41.		IC4	74HCT08N	74HCT08N	DIL14
42.		IC5	74HCT86N	74HCT86N	DIL14
43.		IC6	AT24CP	AT24CP	DIL08
44.		IC7	74HCT14N	74HCT14N	DIL14
45.		IC8	DS1307	DS1307	DIL08
46.		IC9	LM358N	LM358N	DIL08
47.		IC10	74HCT132N	74HCT132N	DIL14
48.		J1	0R	0R	0R/10
49.		J2	0R	0R	0R/10
50.		JP2	Jumper	JP1E	JP1
51.		JP3	Jumper	JP2E	JP2
52.		JP4	Jumper	JP1E	JP1
53.		JP5	Jumper	JP1E	JP1
54.		JP6	Jumper	JP1E	JP1
55.		L1	10µH	INDUCTOR0207/12	0207/12
56.		L2	100µH	SFT830D	SFT830D
57.		LED1	rot LC	LED3MM	LED3MM
58.		OC1	4N25	4N25	DIL06
59.		OC2	MOC3020M	MOC3020M	DIL06
60.		Q1	16,0000 MHz	CRYTALHC49U-V	HC49U-V
61.		Q2	BC546B	BC546B	TO92-EBC
62.		Q3	BC546B	BC546B	TO92-EBC
63.		Q4	BC546B	BC546B	TO92-EBC
64.		Q5	BC546B	BC546B	TO92-EBC
65.		Q6	BD139	BD139	TO126AV
66.		Q7	32,768 kHz	CRYTALTC26V	TC26V
67.		R1	10k	RESISTOR0207/10	0207/10
68.		R2	10k	RESISTOR0207/10	0207/10
69.		R3	4k7	RESISTOR0207/10	0207/10
70.		R4	470R	RESISTOR0207/10	0207/10

71.		R5	470R	RESISTOR0207/10	0207/10
72.		R6	10k	RESISTOR0207/10	0207/10
73.		R7	15k	RESISTOR0207/10	0207/10
74.		R8	15k	RESISTOR0207/10	0207/10
75.		R9	15k	RESISTOR0207/10	0207/10
76.		R10	15k	RESISTOR0207/10	0207/10
77.		R11	100k 1W	RESISTOR0411/15	0411/15
78.		R12	10k	RESISTOR0207/10	0207/10
79.		R13	330R	RESISTOR0207/10	0207/10
80.		R14	220R	RESISTOR0207/10	0207/10
81.		R15	10k	RESISTOR0207/10	0207/10
82.		R16	10k	RESISTOR0207/10	0207/10
83.		R17	39R	RESISTOR0207/10	0207/10
84.		R18	10k	RESISTOR0207/10	0207/10
85.		R19	470R	RESISTOR0207/10	0207/10
86.		R20	10k	RESISTOR0207/10	0207/10
87.		R21	22k	RESISTOR0207/10	0207/10
88.		R22	10k	RESISTOR0207/10	0207/10
89.		R23	10k	RESISTOR0207/10	0207/10
90.		R24	22k	RESISTOR0207/10	0207/10
91.		R25	10k	RESISTOR0207/10	0207/10
92.		R26	39k	RESISTOR0207/10	0207/10
93.		R27	330k	RESISTOR0207/10	0207/10
94.		R28	330k	RESISTOR0207/10	0207/10
95.		R29	500R	RESISTOR0207/10	0207/10
96.		R30	5k	TRIM_EU-PT10S	PT-10S
97.		R31	10k	RESISTOR0207/10	0207/10
98.		R32	1k5	RESISTOR0207/10	0207/10
99.		R33	10k	RESISTOR0207/10	0207/10
100.		R34	10k	RESISTOR0207/10	0207/10
101.		RL1	40.51.9	40.51.9	95.15
102.		RL2	40.51.9	40.51.9	95.15
103.		RL3	40.51.9	40.51.9	95.15
104.		RL4	40.51.9	40.51.9	95.15
105.		SHT1	SHT71	SHT71	SHT7X
106.		SP1	RMP-14SP	RMP-14SP	RMP-14SP
107.		SV1	PinHeader Male 2x20	MA20-2	MA20-2
108.		SV2	PinHeader Male 2x13 (Supply)	MA13-2W	MA13-2W
109.		SV3	PinHeader Male 2x13 (Power)	MA13-2W	MA13-2W
110.		SV4	ISP10	MA05-2	MA05-2
111.		SV5	ISP06	MA03-2	MA03-2
112.		SV7	RS232	MA05-2	MA05-2
113.		T1	TIC225S	TIC225S	TO220DS
114.		X1	AKL171	AKL171-04	AKL171-04
115.		X2	AKL171	AKL171-03	AKL171-03
116.		X3	AKL055	AKL055-02	AKL055-02
117.		X4	AKL055	AKL055-02	AKL055-02

© Markus Fulde, 2007

118.		X5	AKL055	AKL055-03	AKL055-03
119.		X6	AKL055	AKL055-02	AKL055-02

Tabelle 150: Eagle BOM für PCB1 – Controller-Board

10.4.9 Das Controller-Board

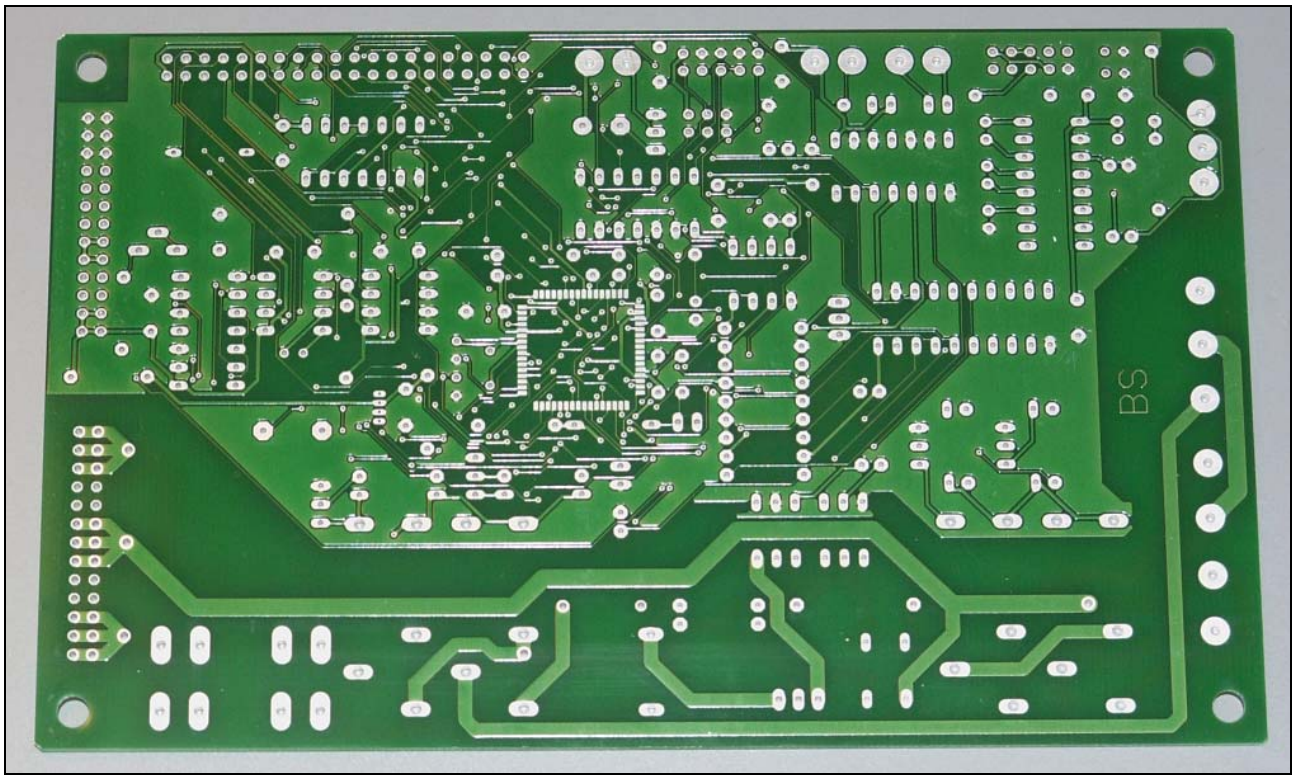


Abbildung 164: PCB1 Controler-Board– PCB

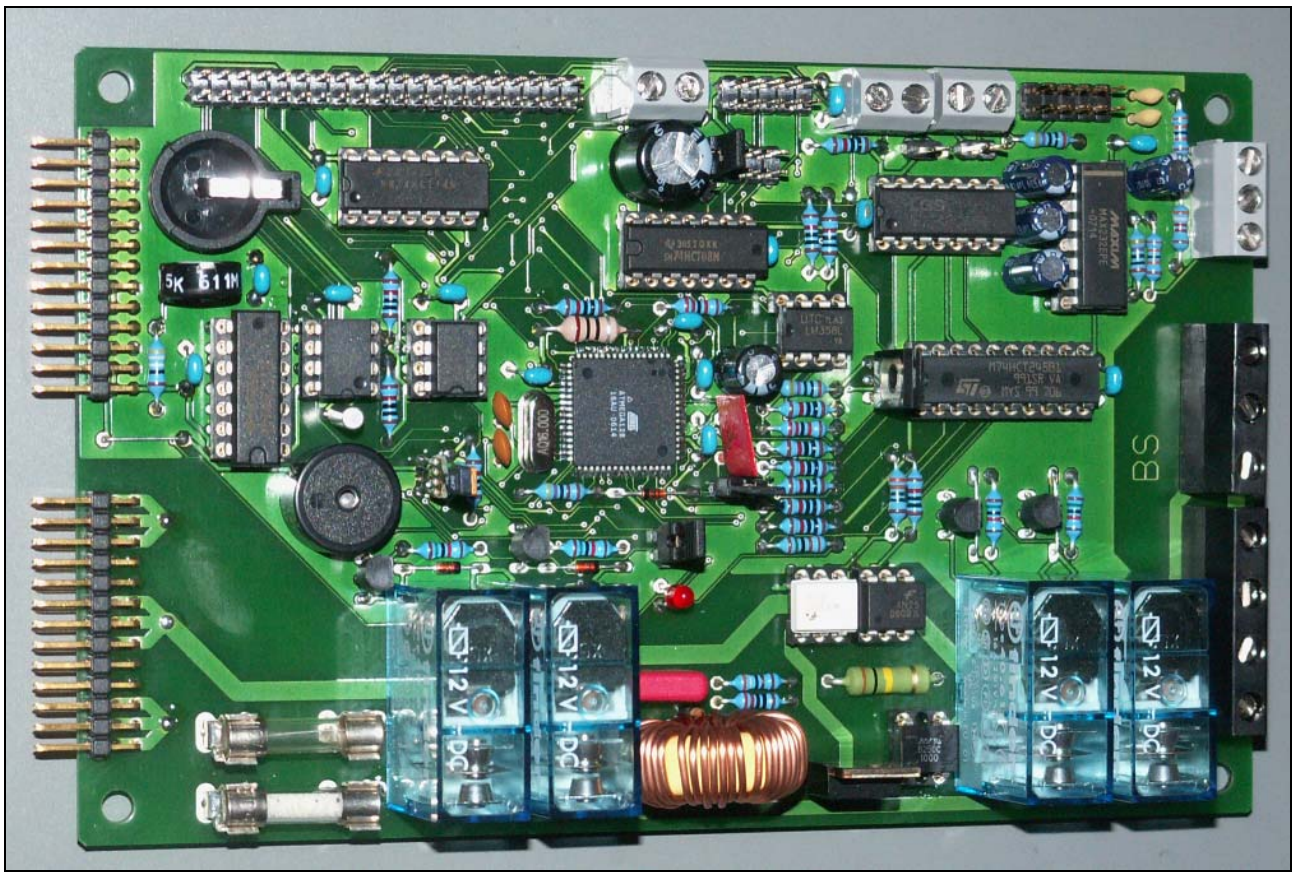


Abbildung 165: PCB1 Controler-Board– fertig bestückt.

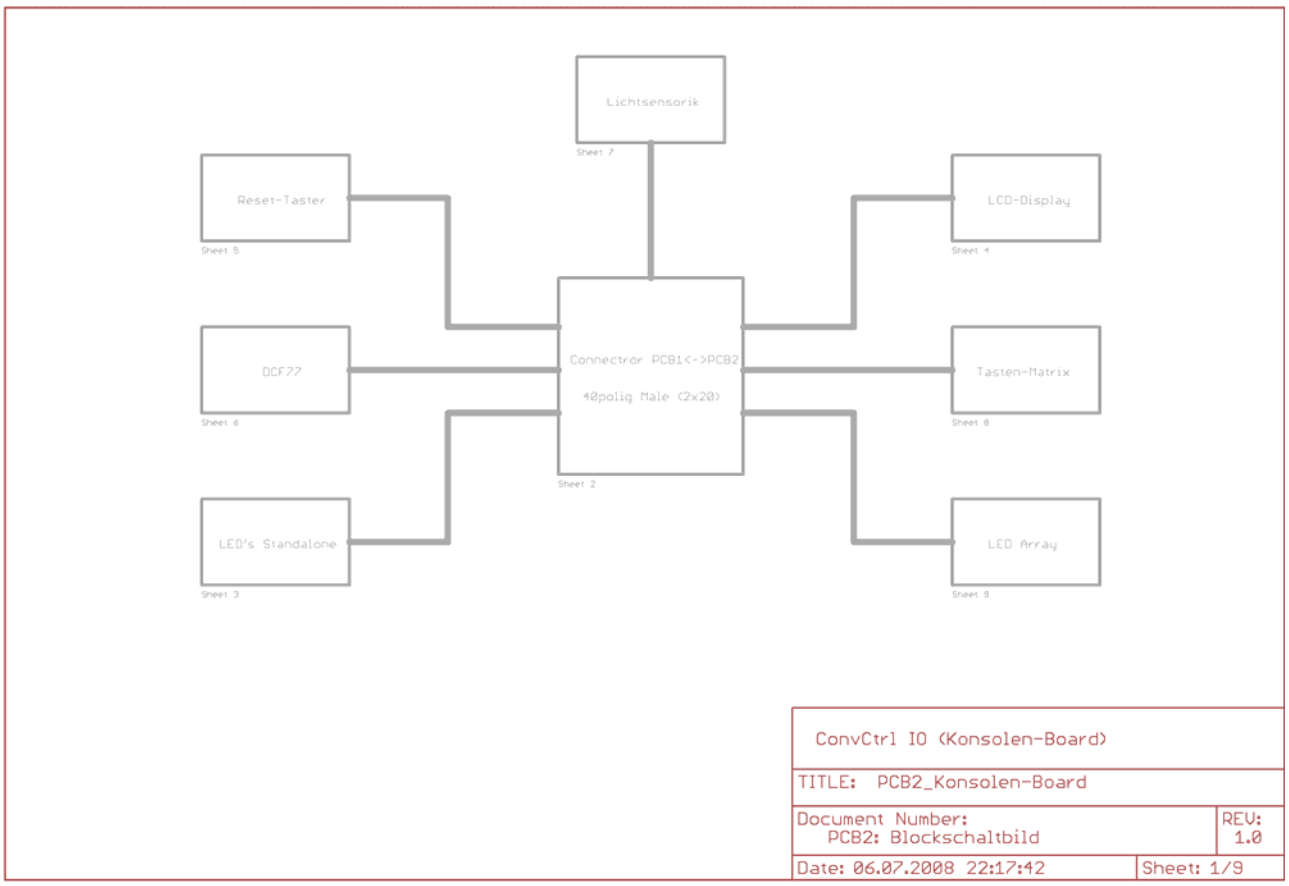
10.4.10 Anmerkungen nach Bestückung und Erstinbetriebnahme

Die folgenden Themen müssen weiter beachtet werden. Sie sind bei der Erstinbetriebnahme und Bestückung aufgefallen:

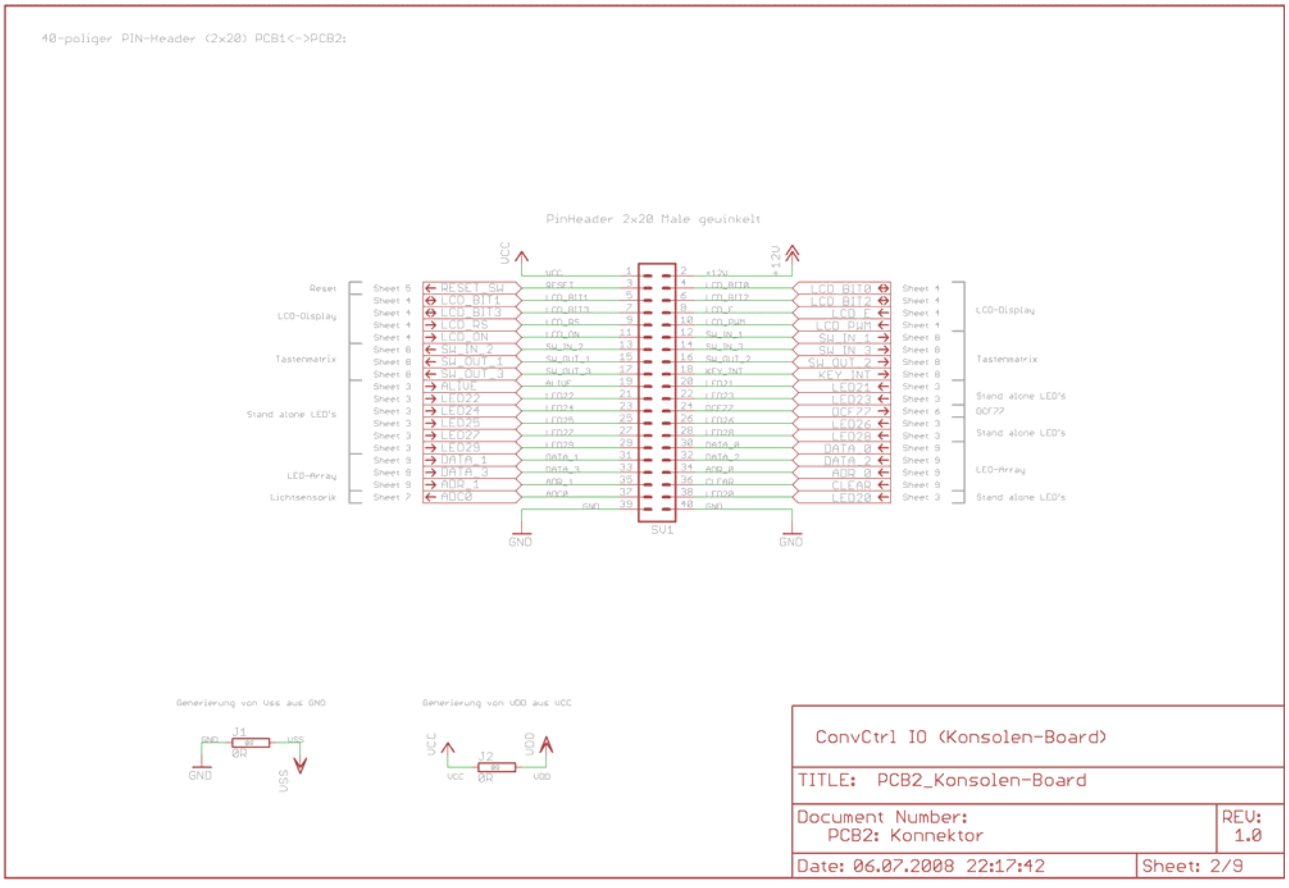
1. Das PIN-Out der Relais war spiegelverkehrt. Im Datenblatt wurde der Hinweis „Ansicht auf die Kontakte“ beim Anlegen des Package übersehen. Es wurde zum Bestücken eine Variante des Relais verwendet welches für mehr Leistung ausgelegt ist und Doppel-Pin-Paare anbietet. Die nicht benutzen wurden abgetrennt. Das Package in der LIB wurde abgeändert. Nachdem die PCB aber fertig vorliegt bleiben die PCB’s in originalem Zustand und werden nicht überarbeitet.
2. Die Drills für den SHT71 sind zu klein so dass der Sensor nicht im Board „durchgesteckt“ werden kann. Er wurde von oben aufgelötet. Das Package in der LIB wurde abgeändert.
3. Die Drills für den Batteriehalter sind zu klein so dass der Halter nicht im Board „durchgesteckt“ werden kann. Er wurde von oben aufgelötet. Das Package in der LIB wurde geändert.

10.5 Das Konsolen-Board PCB2

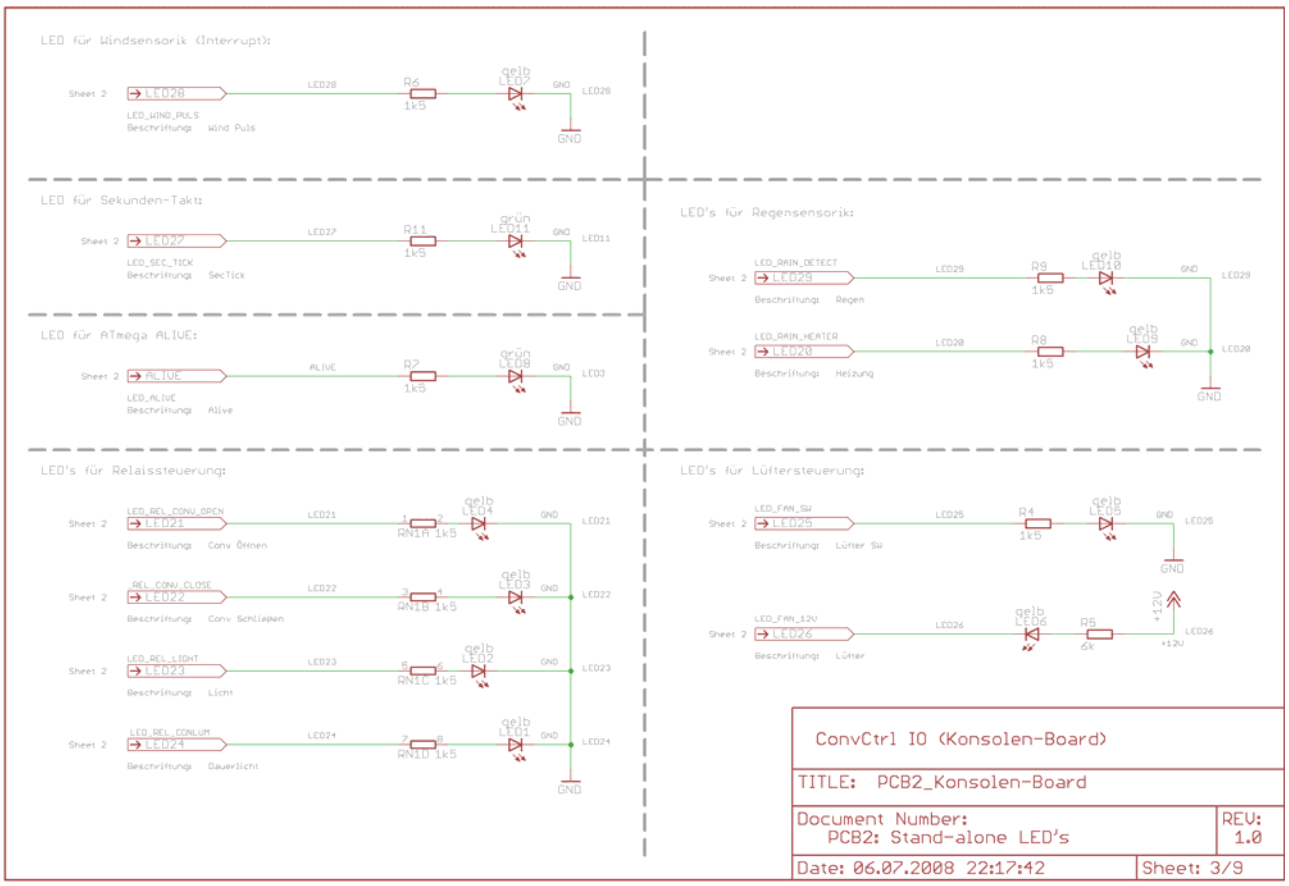
10.5.1 Schematic



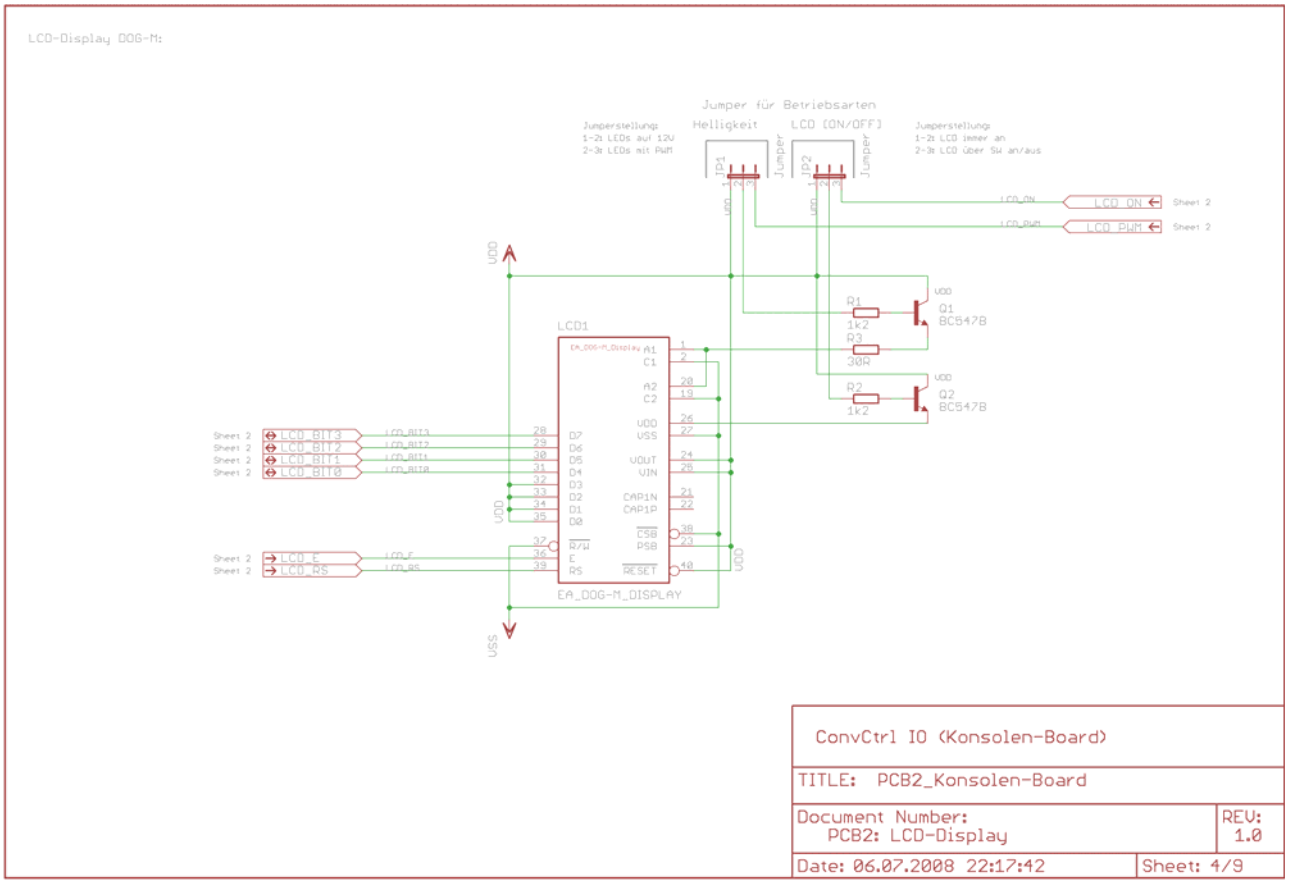
Schaltbild 72: Schaltbild PCB2 Konsolen-Board - Sheet 1



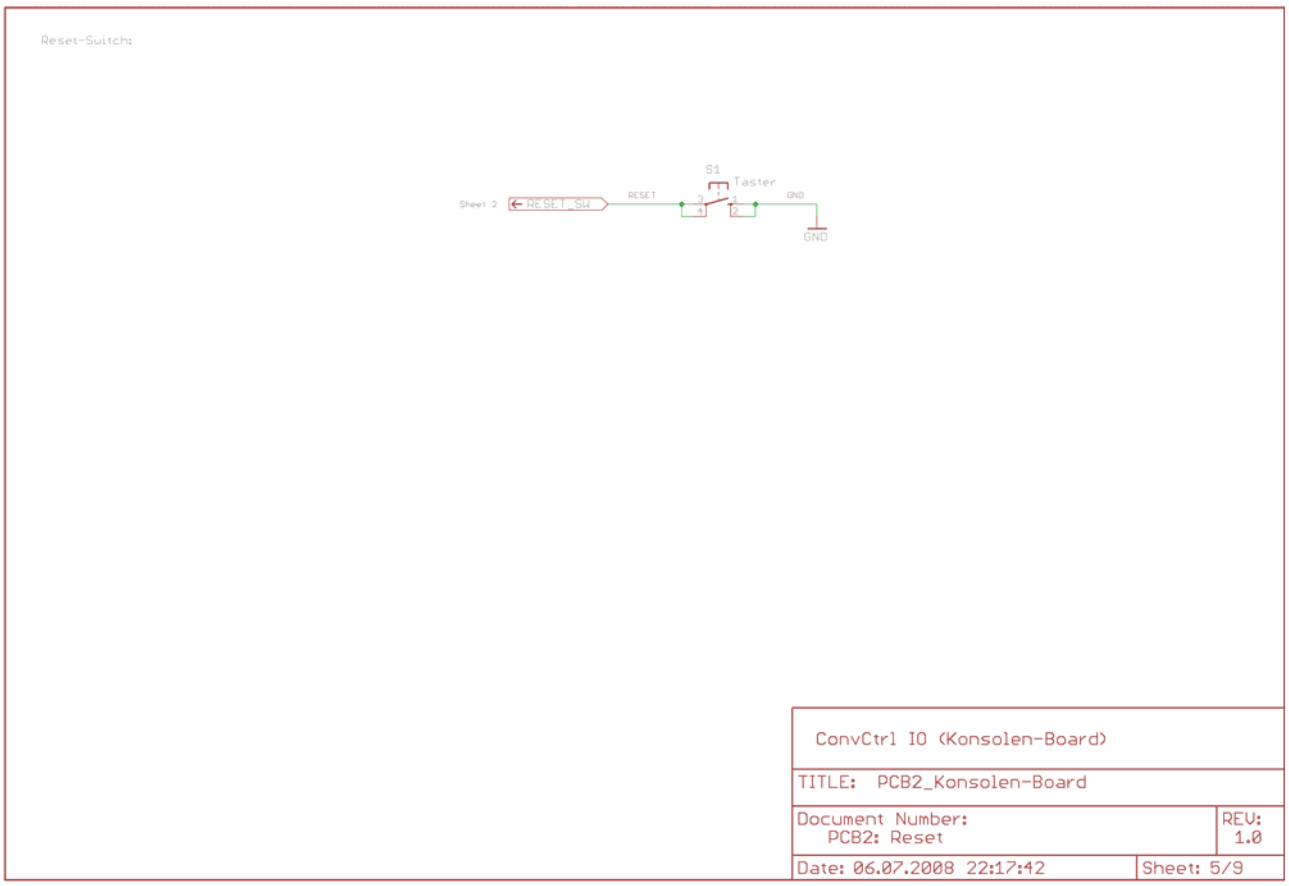
Schaltbild 73: Schaltbild PCB2 Konsolen-Board - Sheet 2



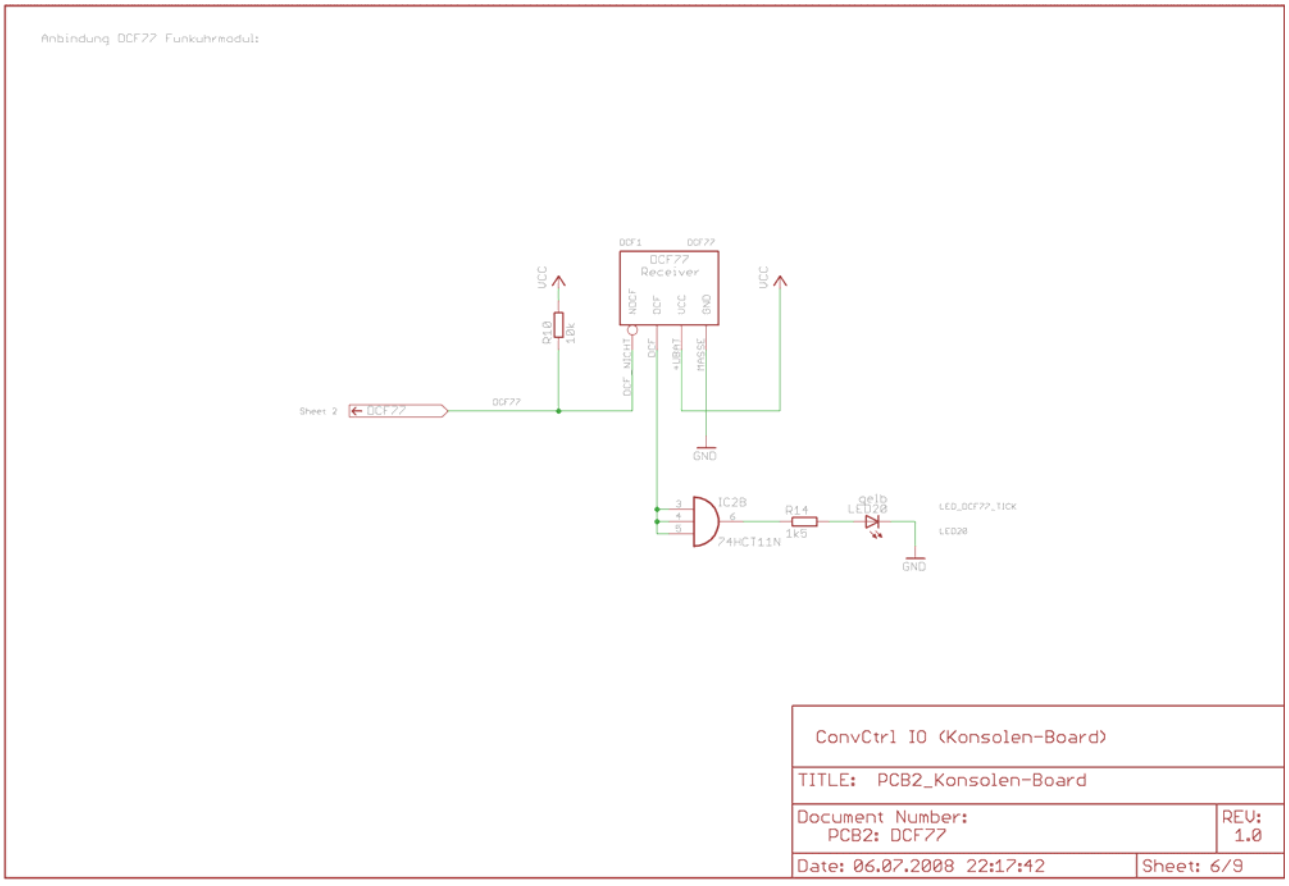
Schaltbild 74: Schaltbild PCB2 Konsolen-Board - Sheet 3



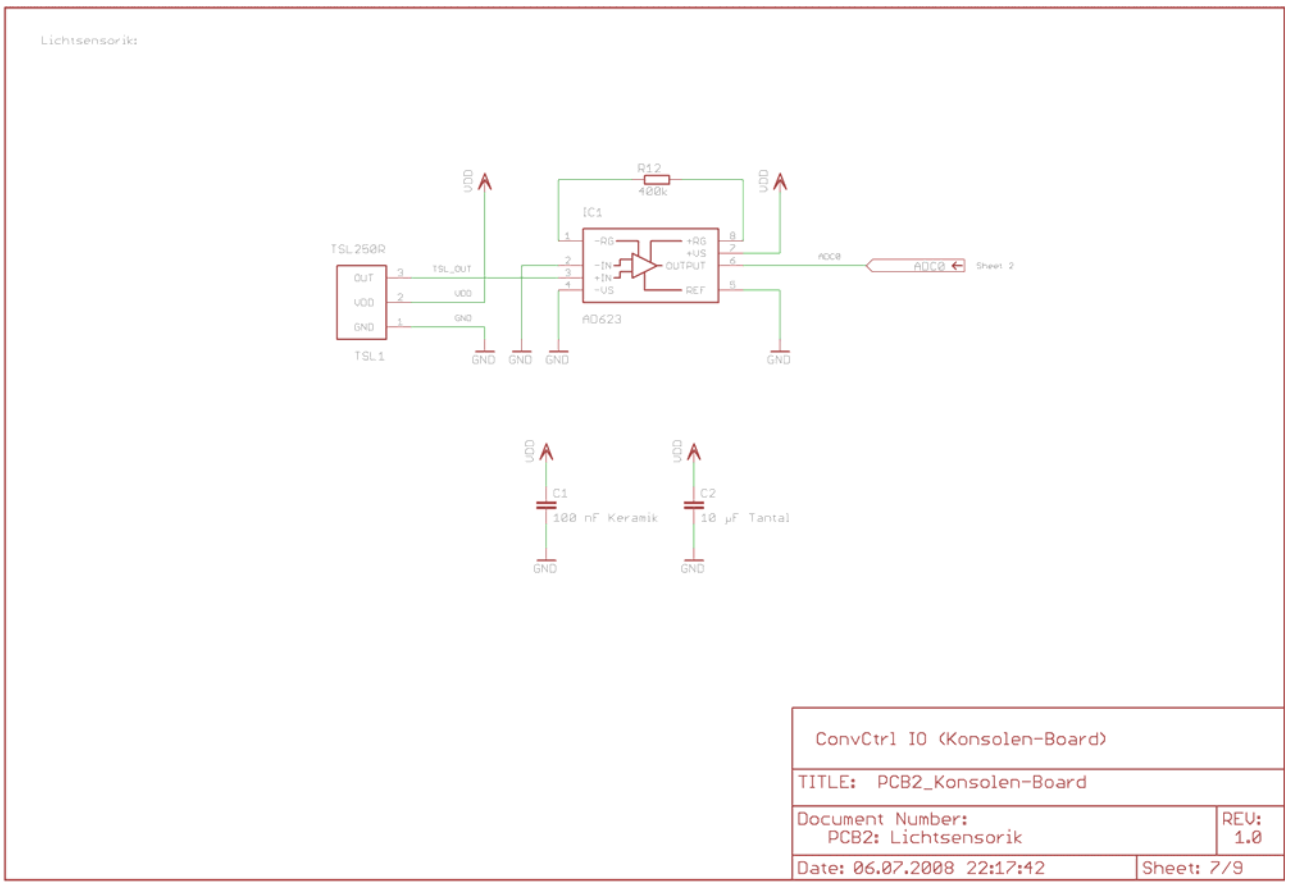
Schaltbild 75: Schaltbild PCB2 Konsolen-Board - Sheet 4



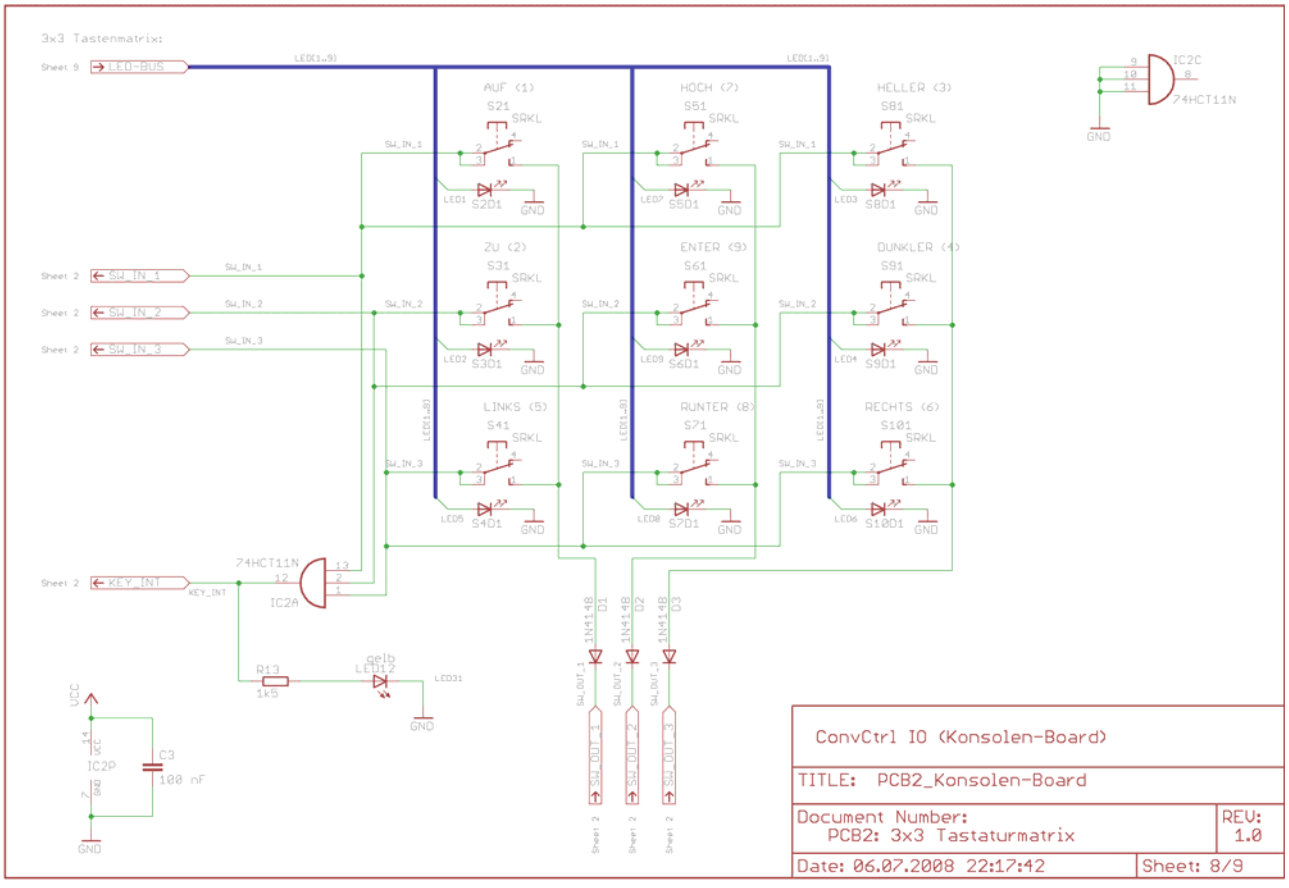
Schaltbild 76: Schaltbild PCB2 Konsolen-Board - Sheet 5



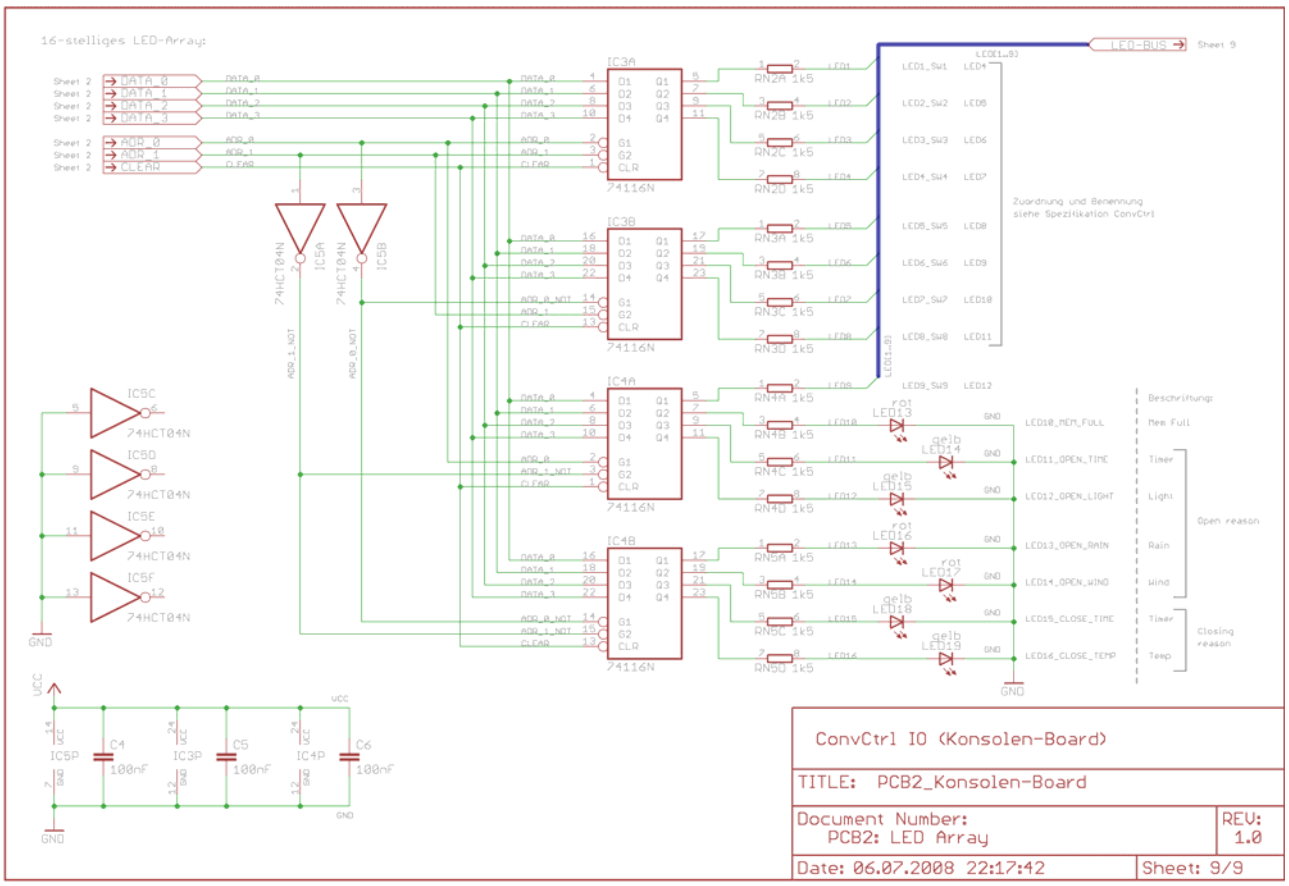
Schaltbild 77: Schaltbild PCB2 Konsolen-Board - Sheet 6



Schaltbild 78: Schaltbild PCB2 Konsolen-Board - Sheet 7



Schaltbild 79: Schaltbild PCB2 Konsolen-Board - Sheet 8



Schaltbild 80: Schaltbild PCB2 Konsolen-Board - Sheet 9

10.5.2 Layout Gesamt

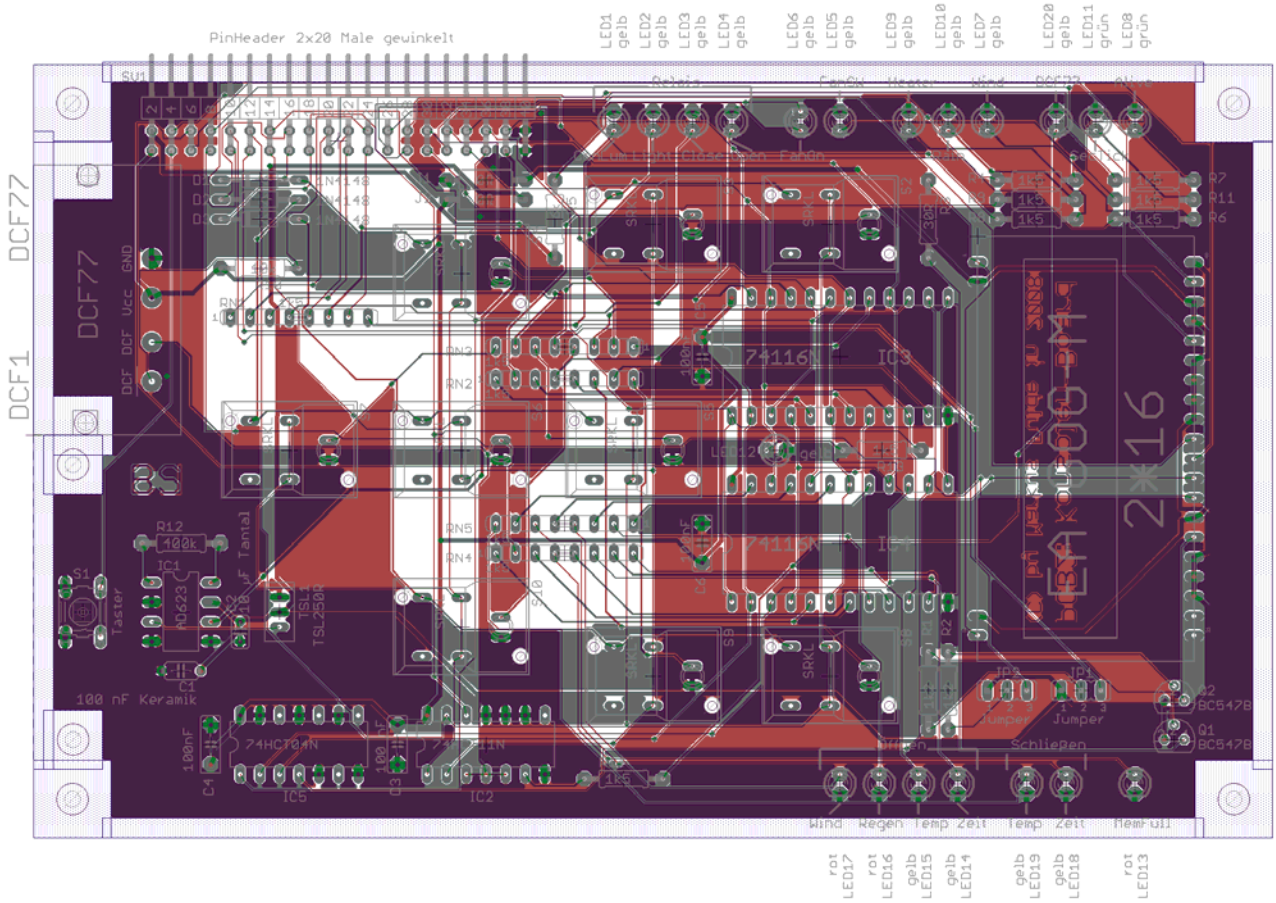


Abbildung 166: PCB2 Konsolen-Board– Layout gesamt

10.5.3 Top-Layer

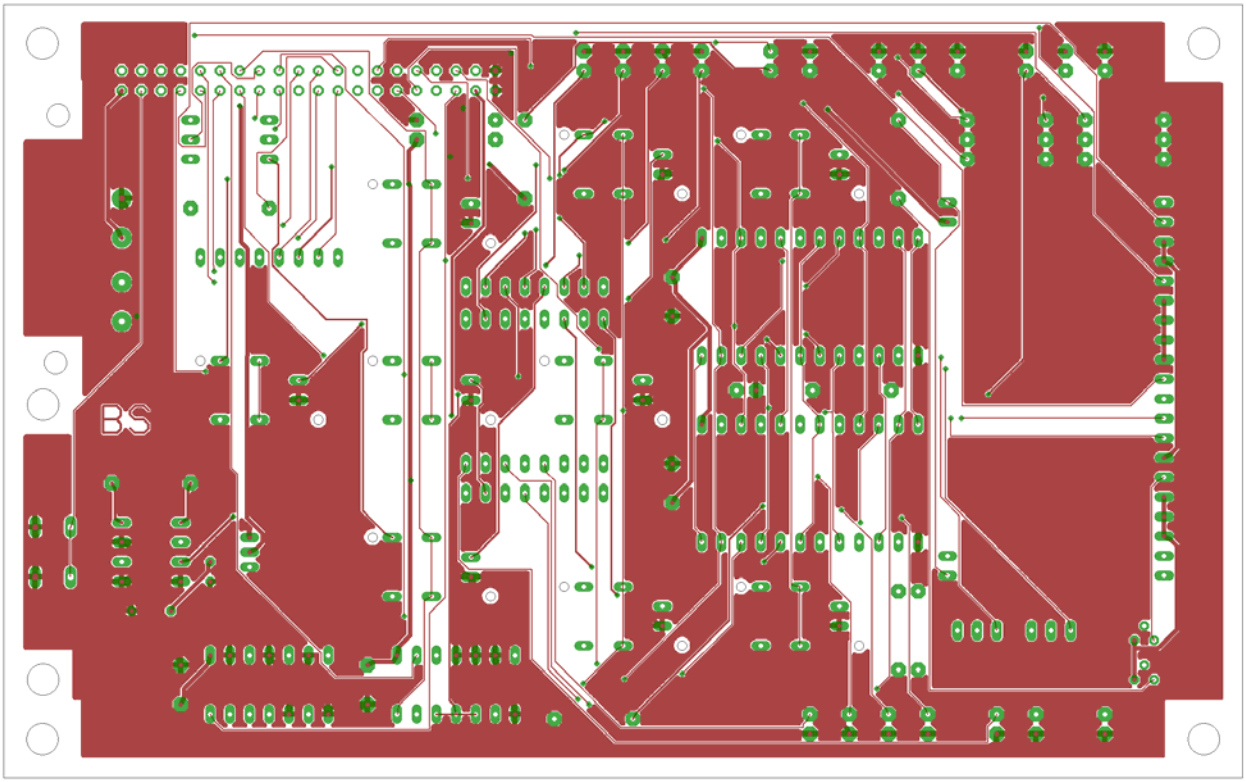


Abbildung 167: PCB2 Konsolen-Board– Top-Layer

10.5.4 Bottom-Layer

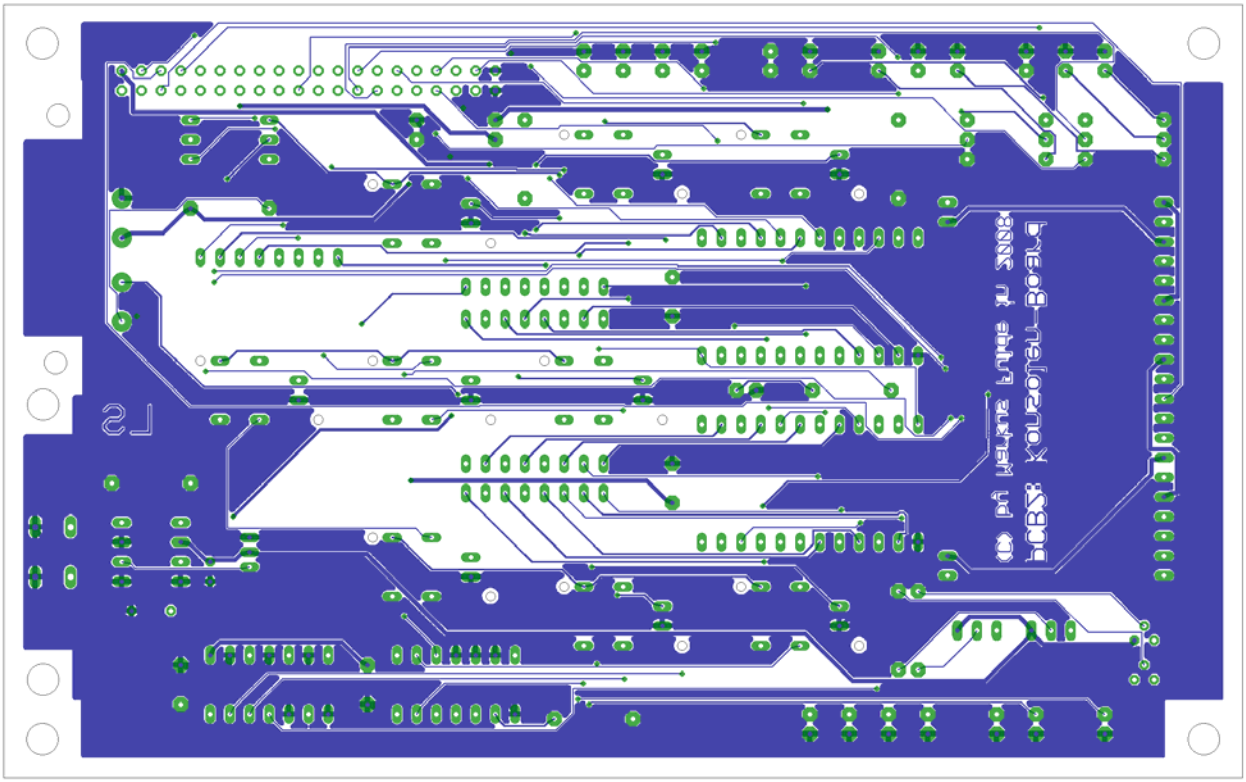


Abbildung 168: PCB2 Konsolen-Board– Bottom-Layer

10.5.5 Pad's und Via's

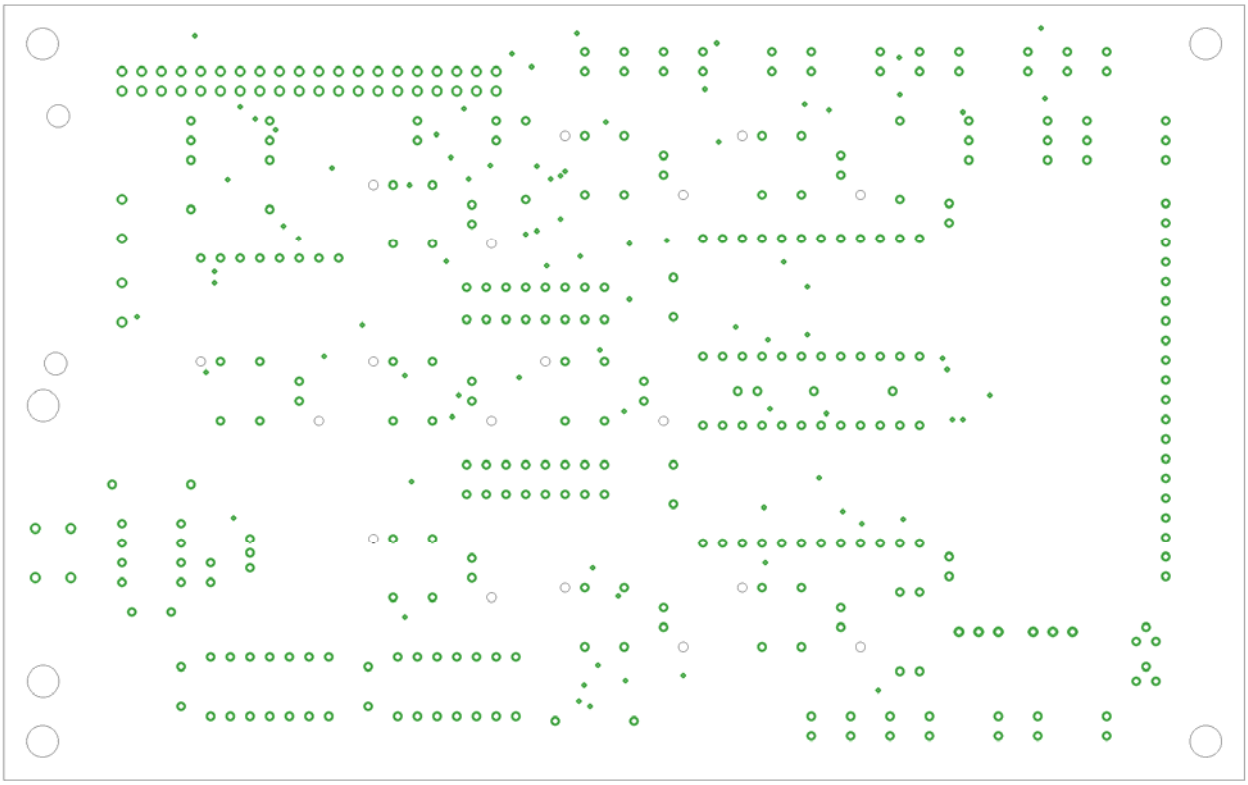


Abbildung 169: PCB2 Konsolen-Board– Pad's und Via's

10.5.6 Bestückungsdruck

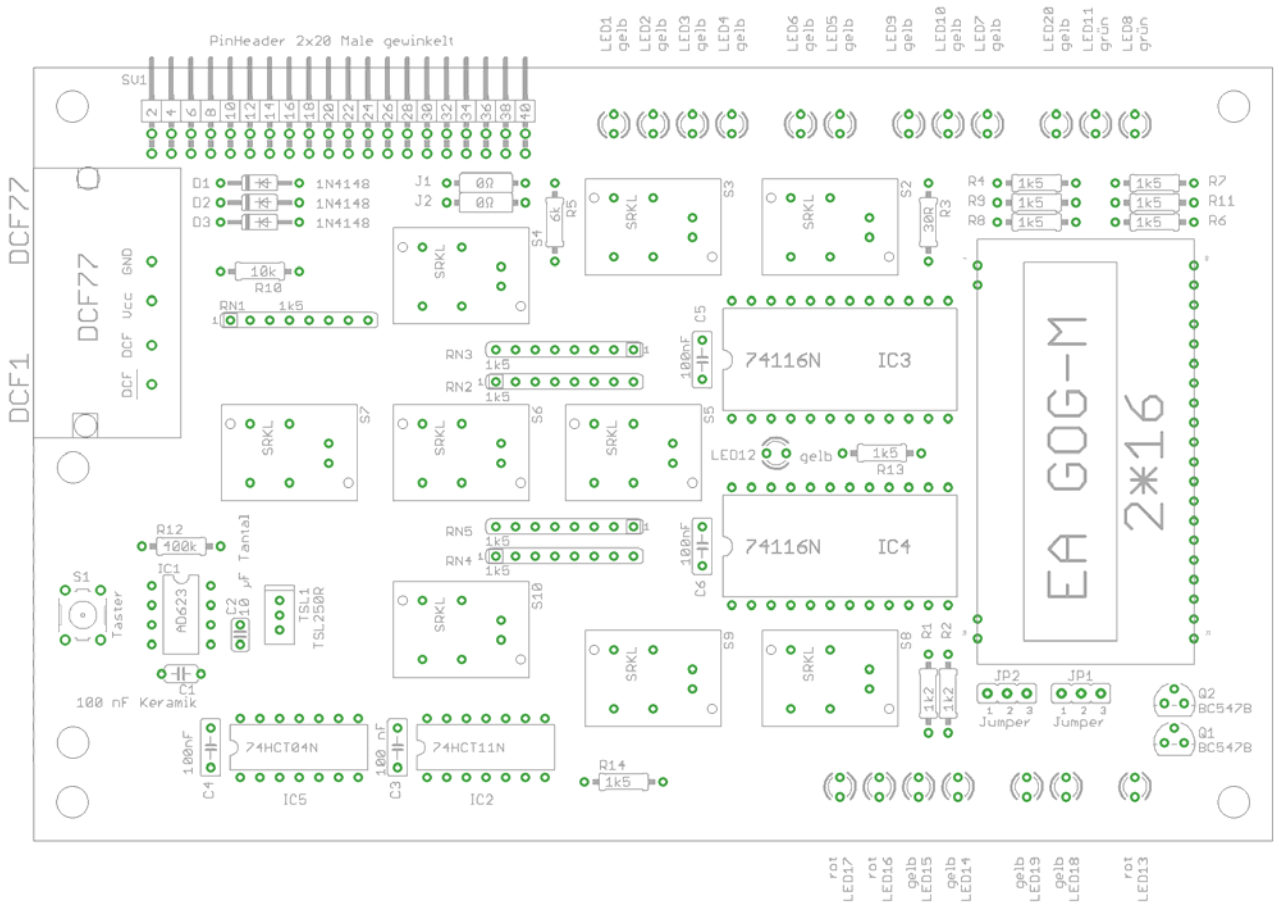


Abbildung 170: PCB2 Konsolen-Board– Bestückungsdruck

Abbildung 171: PCB2 Konsolen-Board– Beschriftung

10.5.7 Restricted Areas

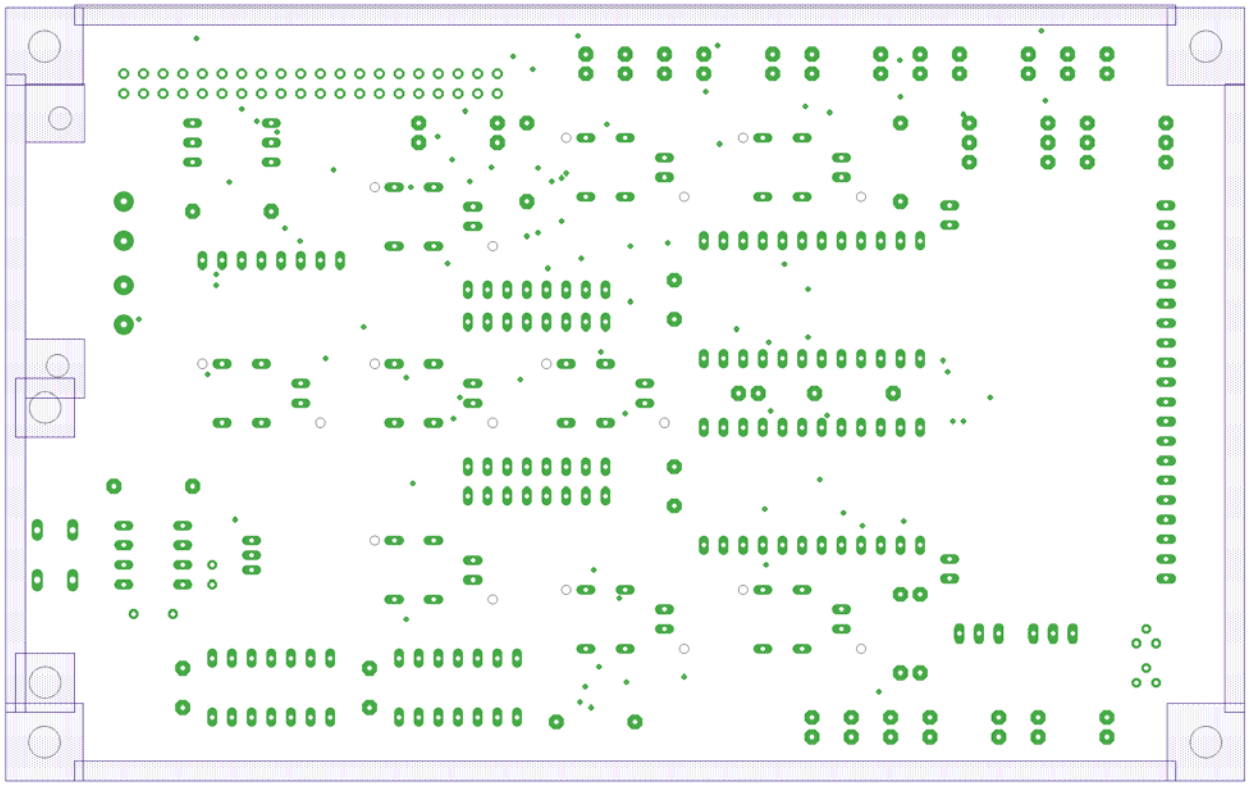


Abbildung 172: PCB2 Konsolen-Board– Restricted Areas

10.5.8 Eagle-BOM

Pos.		Bauteil	Wert	Device	Package
1.		C1	100 nF Keramik	CAPACITOR050-024X044	C050-024X044
2.		C2	10 µF Tantal	CAPACITOR025-024X044	C025-024X044
3.		C3	100 nF	CAPACITOR050-025X075	C050-025X075
4.		C4	100nF	CAPACITOR050-025X075	C050-025X075
5.		C5	100nF	CAPACITOR050-025X075	C050-025X075
6.		C6	100nF	CAPACITOR050-025X075	C050-025X075
7.		D1	1N4148	1N4148	DO35-10
8.		D2	1N4148	1N4148	DO35-10
9.		D3	1N4148	1N4148	DO35-10
10.		DCF1	DCF77	DCF77	DCF77
11.		IC1	AD623	AD623	DIL08
12.		IC2	74HCT11N	74HCT11N	DIL14
13.		IC3	74116N	74116N	DIL24-6
14.		IC4	74116N	74116N	DIL24-6
15.		IC5	74HCT04N	74HCT04N	DIL14
16.		J1	0R	0R	0R/10

17.		J2	0R	0R	0R/10
18.		JP1	Jumper	JP2E	JP2
19.		JP2	Jumper	JP2E	JP2
20.		LCD1	EA_DOG-M_DISPLAY	EA_DOG-M_DISPLAY	E_DOG-M_DISPLAY
21.		LED1	gelb	LED3MM	LED3MM
22.		LED2	gelb	LED3MM	LED3MM
23.		LED3	gelb	LED3MM	LED3MM
24.		LED4	gelb	LED3MM	LED3MM
25.		LED5	gelb	LED3MM	LED3MM
26.		LED6	gelb	LED3MM	LED3MM
27.		LED7	gelb	LED3MM	LED3MM
28.		LED8	grün	LED3MM	LED3MM
29.		LED9	gelb	LED3MM	LED3MM
30.		LED10	gelb	LED3MM	LED3MM
31.		LED11	grün	LED3MM	LED3MM
32.		LED12	gelb	LED3MM	LED3MM
33.		LED13	rot	LED3MM	LED3MM
34.		LED14	gelb	LED3MM	LED3MM
35.		LED15	gelb	LED3MM	LED3MM
36.		LED16	rot	LED3MM	LED3MM
37.		LED17	rot	LED3MM	LED3MM
38.		LED18	gelb	LED3MM	LED3MM
39.		LED19	gelb	LED3MM	LED3MM
40.		LED20	gelb	LED3MM	LED3MM
41.		Q1	BC547B	BC547B	TO92
42.		Q2	BC547B	BC547B	TO92
43.		R1	1k2	RESISTOR0207/10	0207/10
44.		R2	1k2	RESISTOR0207/10	0207/10
45.		R3	30R	RESISTOR0207/10	0207/10
46.		R4	1k5	RESISTOR0207/10	0207/10
47.		R5	6k	RESISTOR0207/10	0207/10
48.		R6	1k5	RESISTOR0207/10	0207/10
49.		R7	1k5	RESISTOR0207/10	0207/10
50.		R8	1k5	RESISTOR0207/10	0207/10
51.		R9	1k5	RESISTOR0207/10	0207/10
52.		R10	10k	RESISTOR0207/10	0207/10
53.		R11	1k5	RESISTOR0207/10	0207/10
54.		R12	400k	RESISTOR0207/10	0207/10
55.		R13	1k5	RESISTOR0207/10	0207/10
56.		R14	1k5	RESISTOR0207/10	0207/10
57.		RN1	1k5	E4R	SIL8
58.		RN2	1k5	E4R	SIL8
59.		RN3	1k5	E4R	SIL8
60.		RN4	1k5	E4R	SIL8
61.		RN5	1k5	E4R	SIL8
62.		S1	Taster	MJTP1230	MJTP1230
63.		S2	SRKL	SRKL	SRKL
64.		S3	SRKL	SRKL	SRKL

65.		S4	SRKL	SRKL	SRKL
66.		S5	SRKL	SRKL	SRKL
67.		S6	SRKL	SRKL	SRKL
68.		S7	SRKL	SRKL	SRKL
69.		S8	SRKL	SRKL	SRKL
70.		S9	SRKL	SRKL	SRKL
71.		S10	SRKL	SRKL	SRKL
72.		SV1	PinHeader 2x20 Male gewinkelt	MA20-2W	MA20-2W
73.		TSL1	TSL250R	TSL250R	S

Tabelle 151: Eagle BOM für PCB2 – Konsolen-Board

10.5.9 Das Konsolen-Board

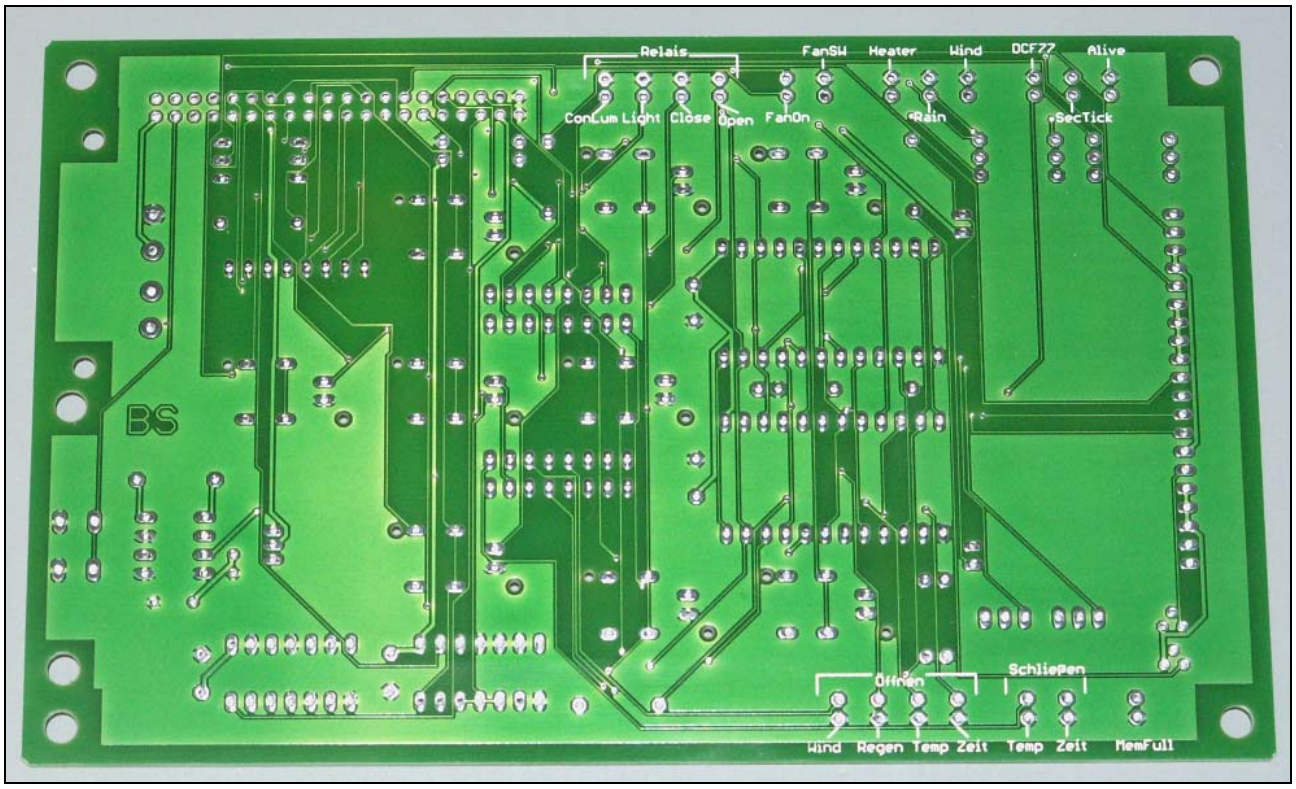


Abbildung 173: PCB2 Konsolen-Board– PCB

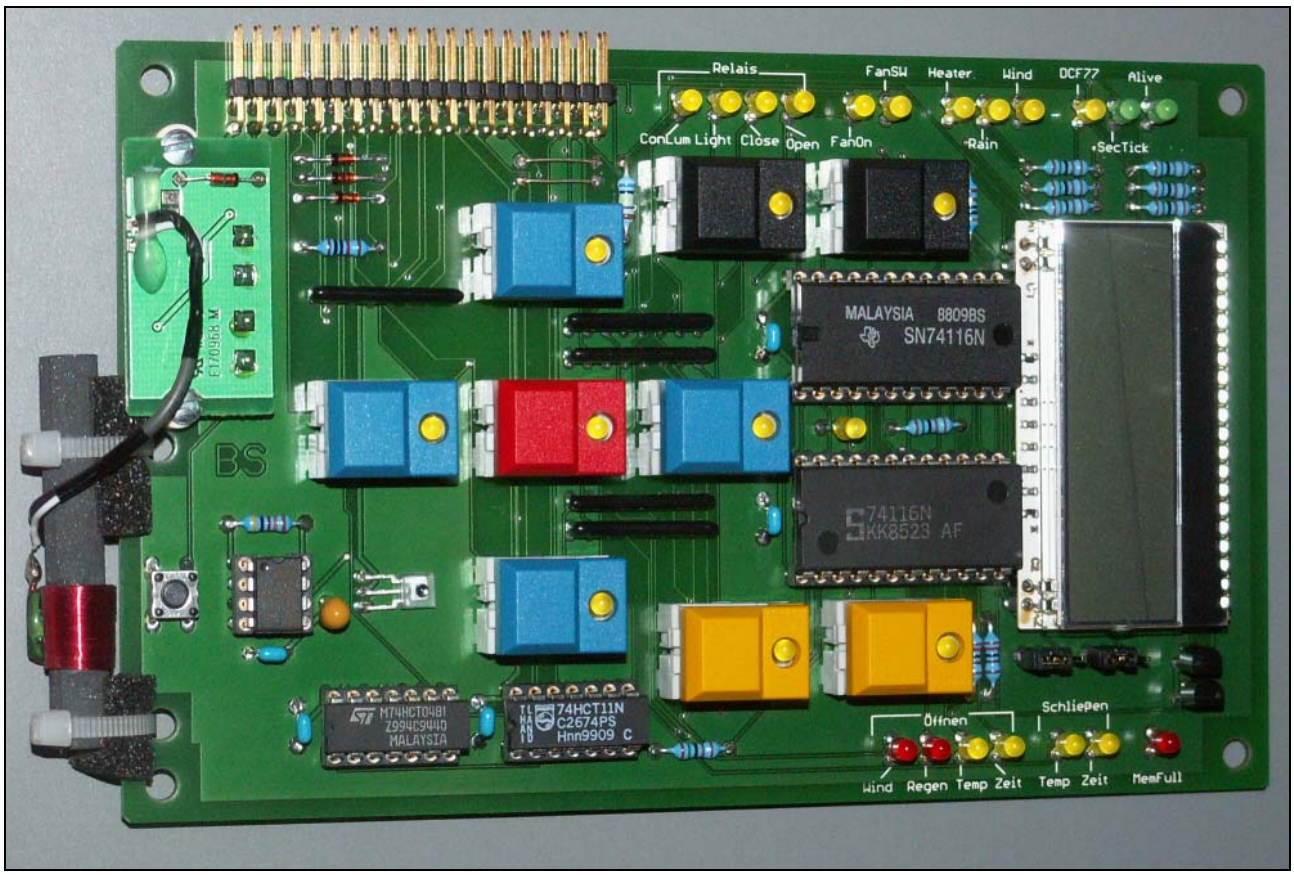


Abbildung 174: PCB2 Konsolen-Board– fertig bestückt

10.5.10 Anmerkungen nach Bestückung und Erstinbetriebnahme

Die folgenden Themen müssen weiter beachtet werden. Sie sind bei der Erstinbetriebnahme und Bestückung aufgefallen:

1. Der Tantal-Elko C2 welcher polarisiert ist wurde mit falscher Polung eingebaut. Die Inbetriebnahme erfolgte ohne Probleme. Nach ca. 3 Wochen „starb" der Tantal-Elko und verursachte andauernden Kurzschluss.
2. In Schaltbild 6/9 fehlt in der Beschaltung des DCF77-Moduls ein 10kOhm-Widerstand zwischen Vcc und dem Ausgang DCF. Die Ausgänge des Moduls sind Open-Collector-Ausgänge. Ohne Widerstand ist das Signal immer NULL. Da die PCB schon fertig produziertund bestückt war wird auf eine Änderung des Schematic und der Dokumentation verzichtet. Der fehlende Widerstand wird direkt auf dem DCF77-Modul verlötet.

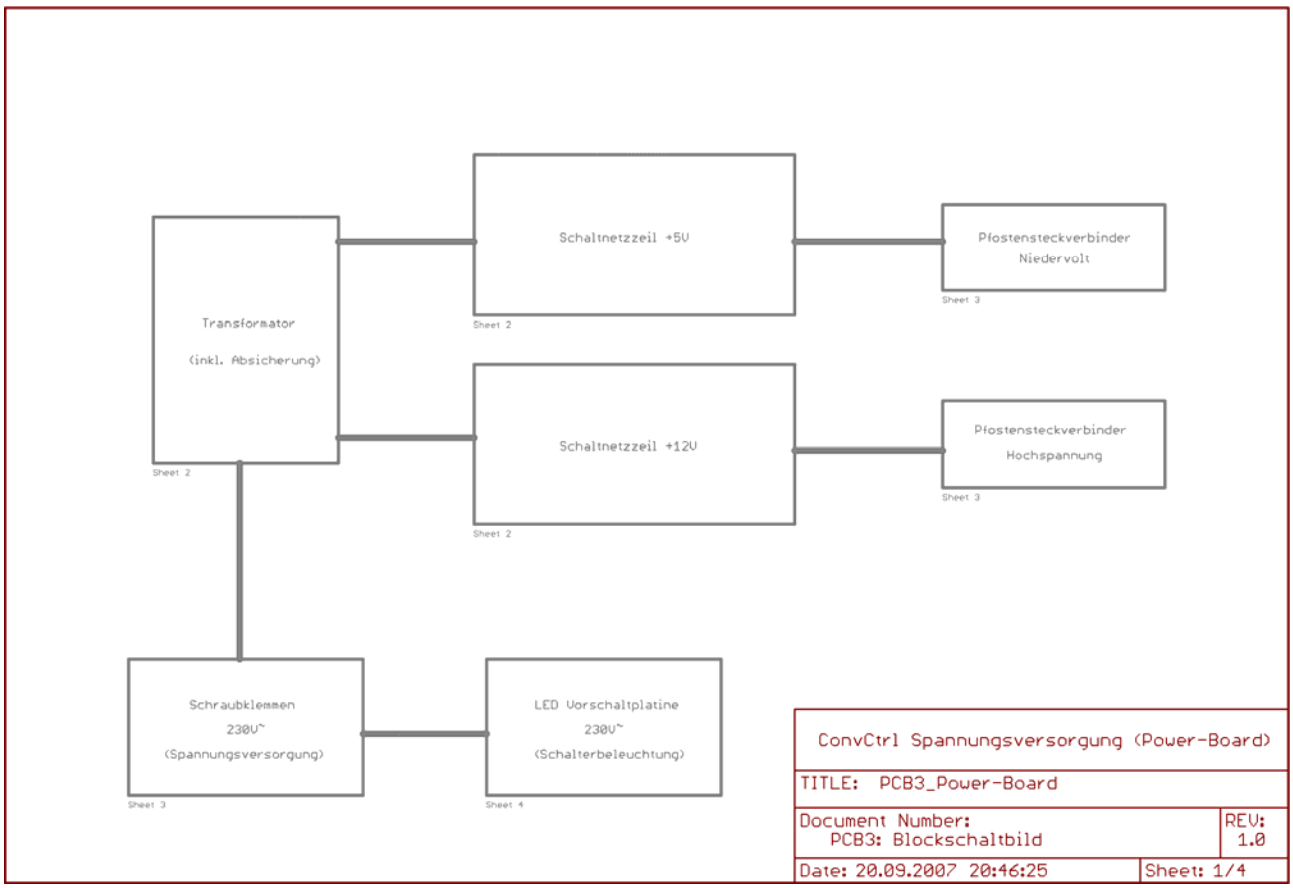
10.6 Das Power-Board PCB3

Als Richtwert für die Auslegung der 230V Leiterbahnbreiten und die geschätzten Ströme gilt:

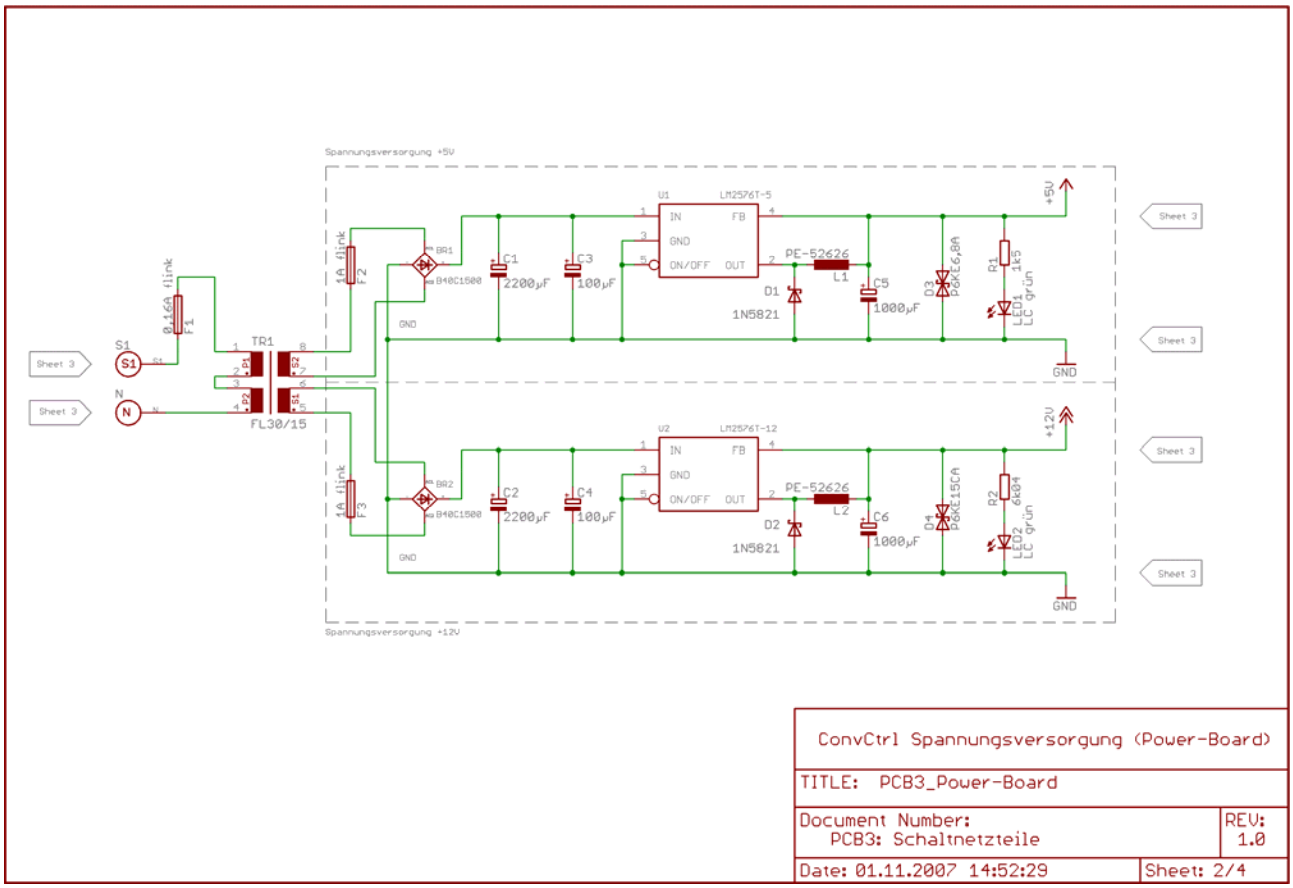
1 mm Leiterbahnbreite bei 35µ Kupfer pro Ampere

Die in der Schaltung für 230V vorhandenen Leiterbahnen zum Transport der Leistungen werden mit eine Bahnbreite von 4mm und einer Kupferstärke von 70µ gefertigt. Damit ist die Schaltung für einen Strom von 8 Ampere ausgelegt.

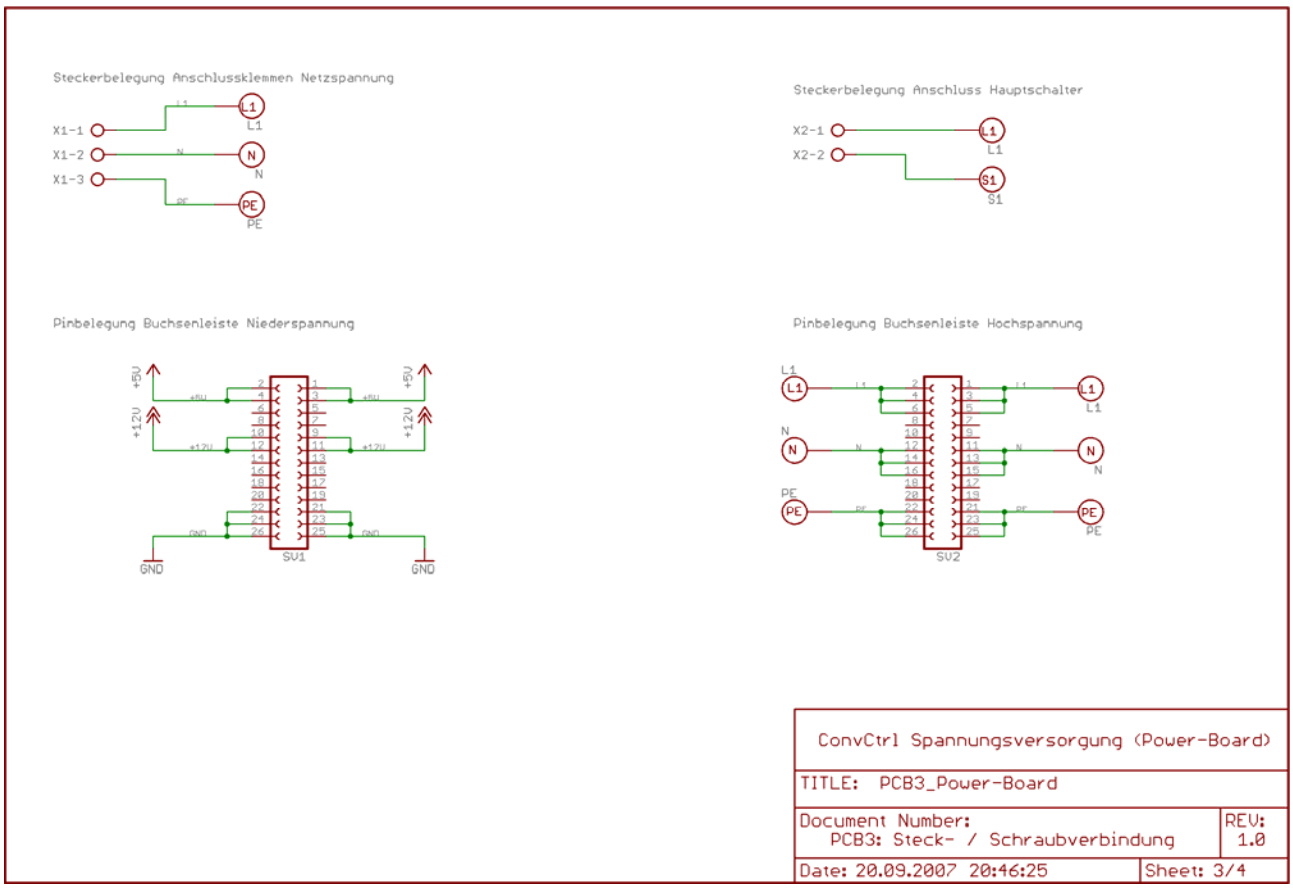
10.6.1 Schematic



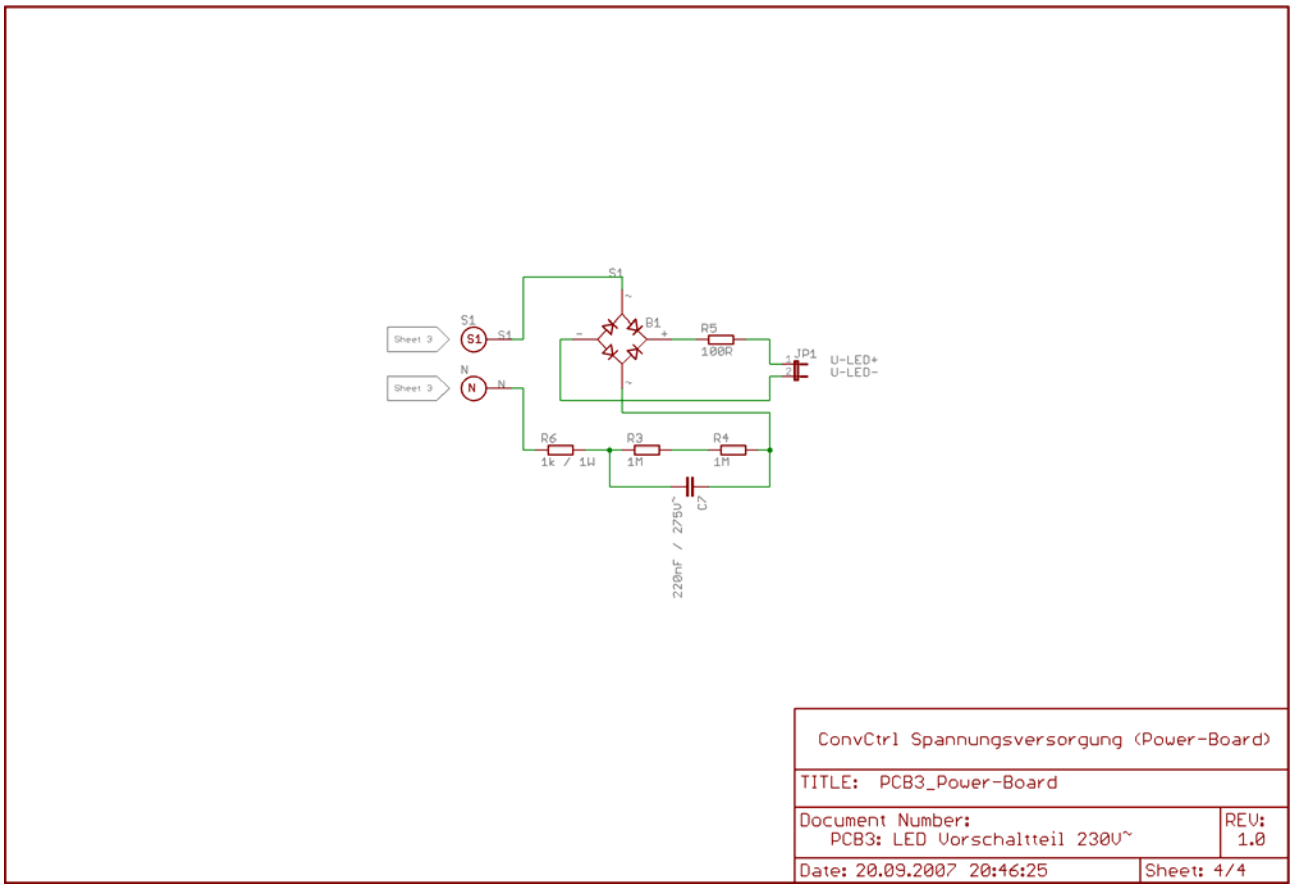
Schaltbild 81: Schaltbild PCB3 Power-Board - Sheet 1



Schaltbild 82: Schaltbild PCB3 Power-Board - Sheet 2



Schaltbild 83: Schaltbild PCB3 Power-Board - Sheet 3



Schaltbild 84: Schaltbild PCB3 Power-Board - Sheet 4

10.6.2 Layout Gesamt

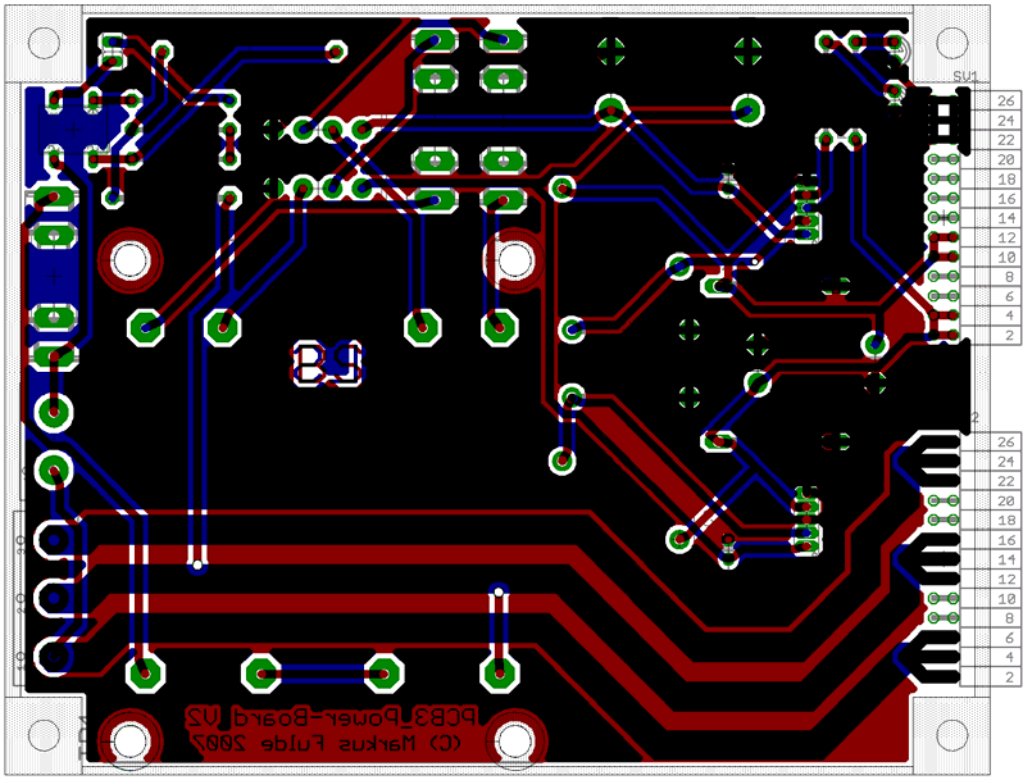


Abbildung 175: PCB3 Power-Board – Layout gesamt

10.6.3 Top-Layer

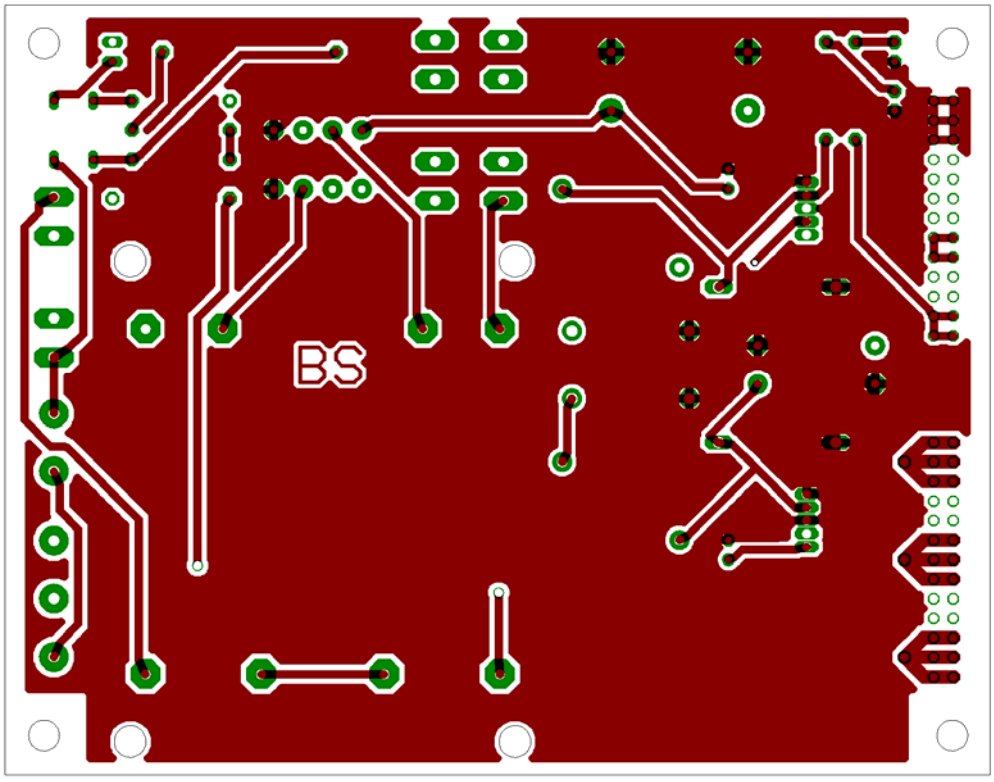


Abbildung 176: PCB3 Power-Board – Top-Layer

10.6.4 Bottom-Layer

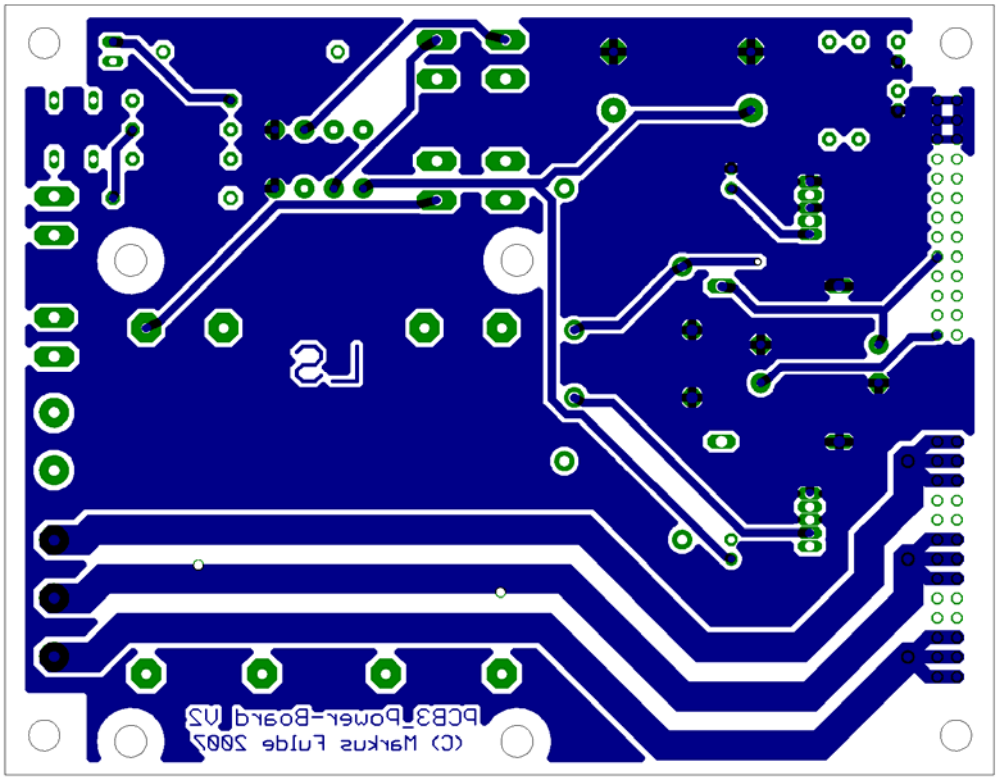


Abbildung 177: PCB3 Power-Board – Bottom Layer

10.6.5 Pad's und Via's

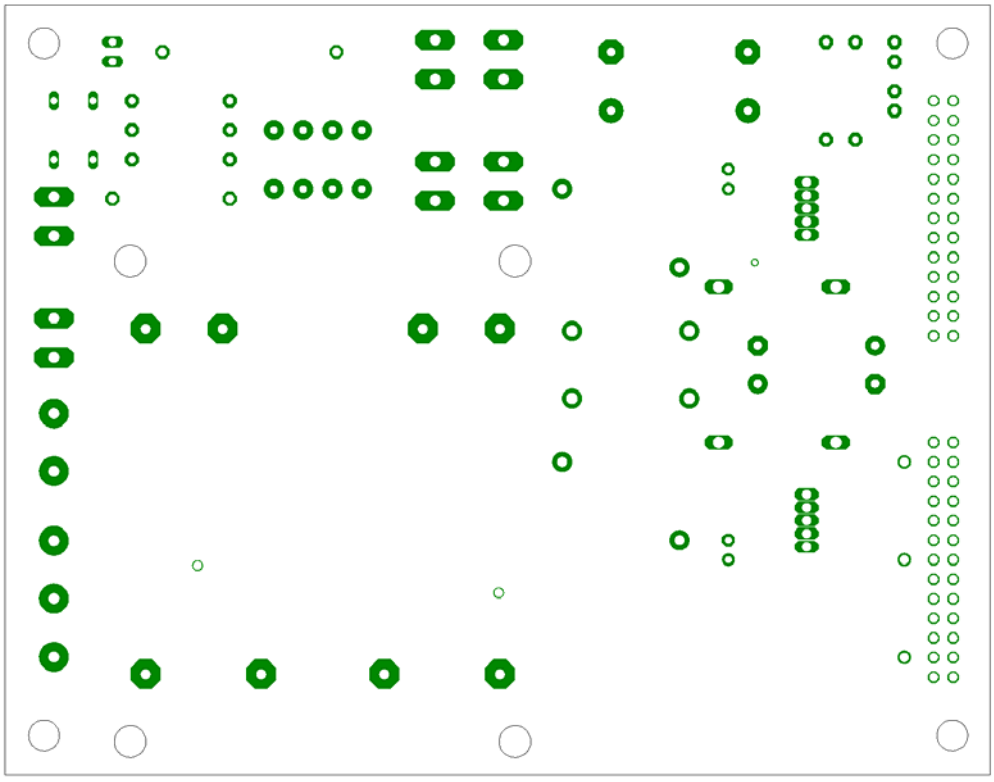


Abbildung 178: PCB3 Power-Board – Pad's and Via's

10.6.6 Bestückungsdruck

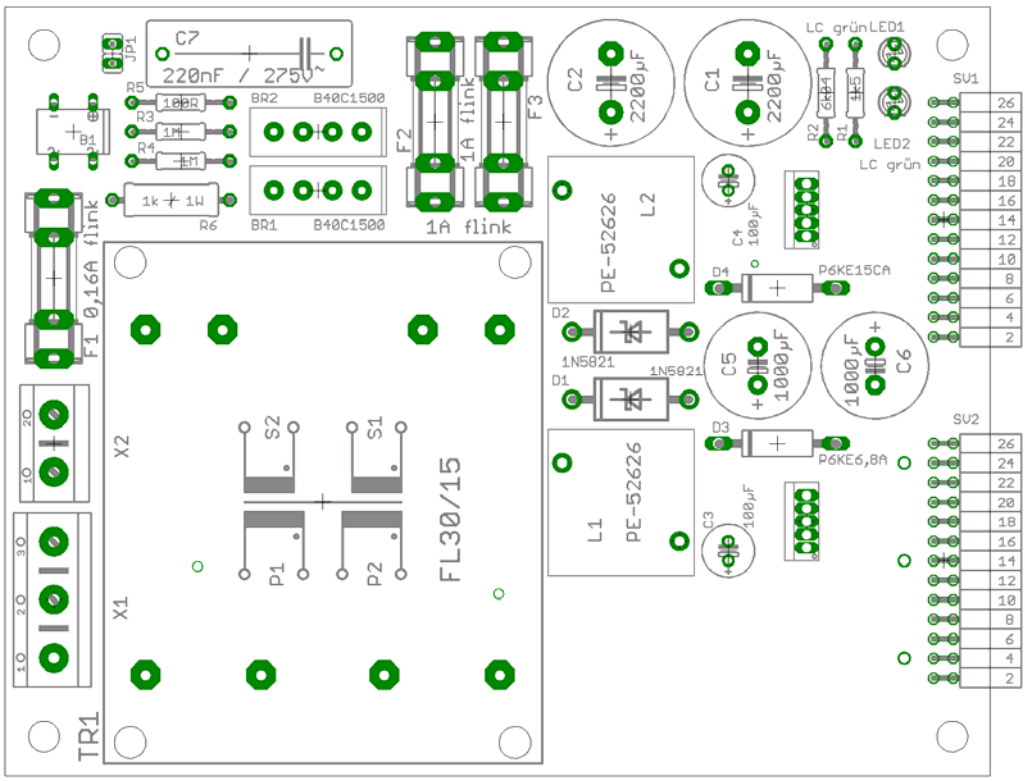


Abbildung 179: PCB3 Power-Board – Bestückung

10.6.7 Restricted Areas

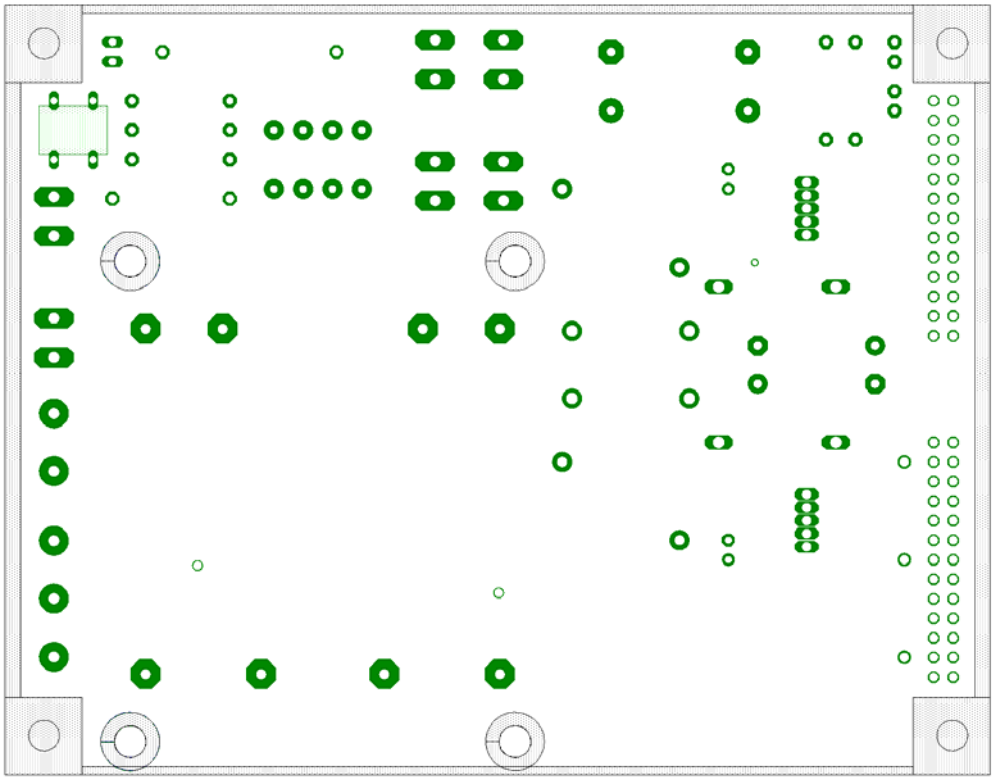


Abbildung 180: PCB3 Power-Board – Restricted Areas

10.6.8 Eagle-BOM

Pos.		Menge	Wert	Device	Bauteile
1.		2	1N5821	1N5822RM15	D1, D2
2.		1		AKL171-02	X2
3.		1		AKL171-03	X1
4.		2	B40C1500	B40C1500	BR1, BR2
5.		1		B250C1000	B1
6.		1	220nF / 275V~	CAPACITOR225-087X268	C7
7.		2		FE13-2W	SV1, SV2
8.		1	FL30/15	FL30/15	TR1
9.		2	220µH	PE-52626	L1, L2
10.		1		JP1E	JP1
11.		2	LC grün	LED3MM	LED1, LED2
12.		1	LM2576T-5	LM2576T-5	U1
13.		1	LM2576T-12	LM2576T-12	U2
14.		1	P6KE6,8A	P6KEXXC	D3
15.		1	P6KE15CA	P6KEXXC	D4
16.		2	100µF	POLARIZEDCAPACITORE2.5-7	C3, C4
17.		2	1000µF	POLARIZEDCAPACITORE5-13	C5, C6
18.		2	2200µF	POLARIZEDCAPACITORE7.5-16	C1, C2
19.		2	1M	RESISTOR0207/12	R3, R4
20.		1	1k5	RESISTOR0207/12	R1
21.		1	6k04	RESISTOR0207/12	R2
22.		1	100R	RESISTOR0207/12	R5
23.		1	1k / 1W	RESISTOR0414/15	R6
24.		1	0,16A flink	SHK20L	F1
25.		2	1A flink	SHK20L	F2, F3

Tabelle 152: Eagle BOM für Power-Board

10.6.9 Das Power-Board

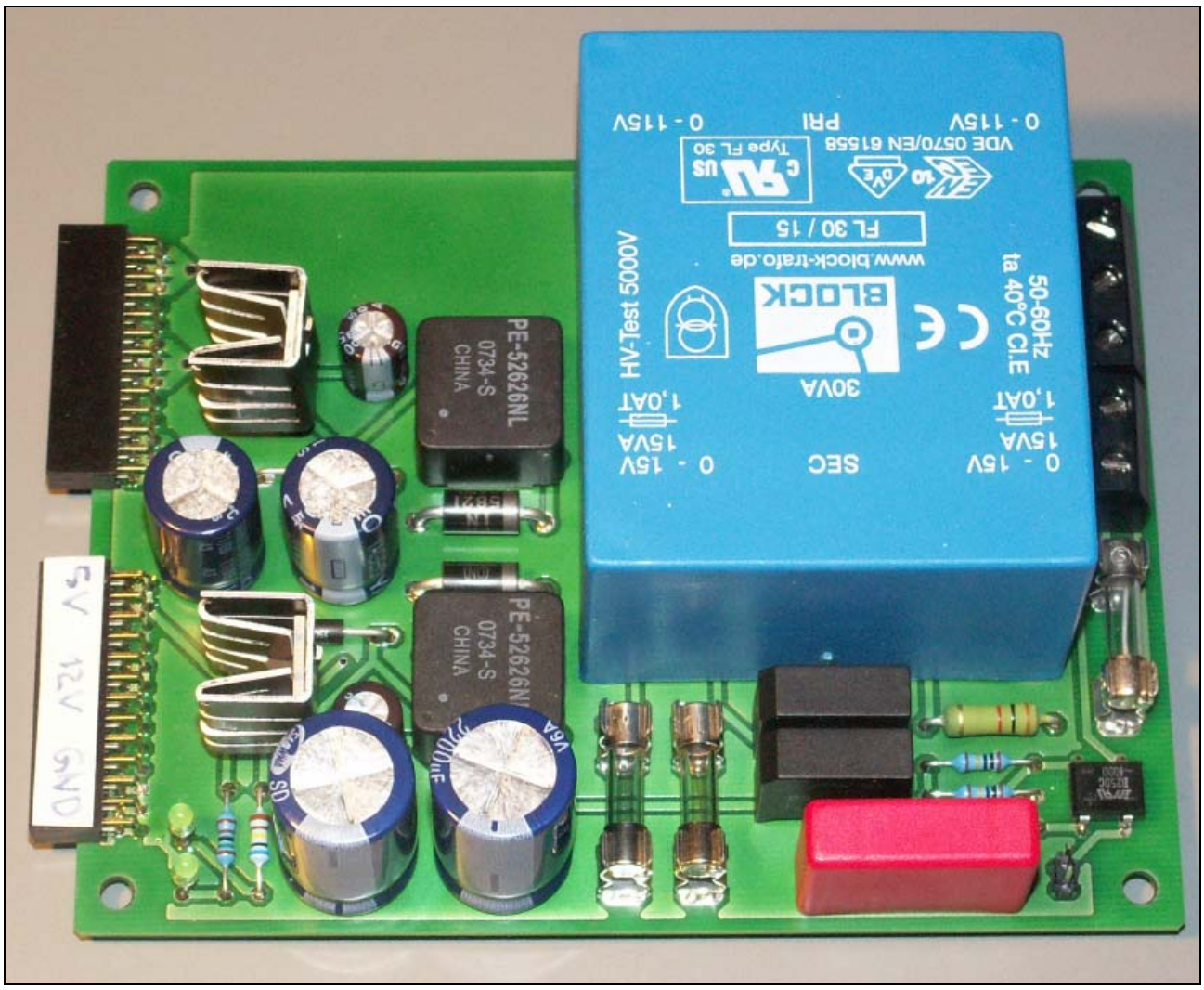


Abbildung 181: PCB3 Power-Board

10.7 Die fertige Entwicklungsumgebung

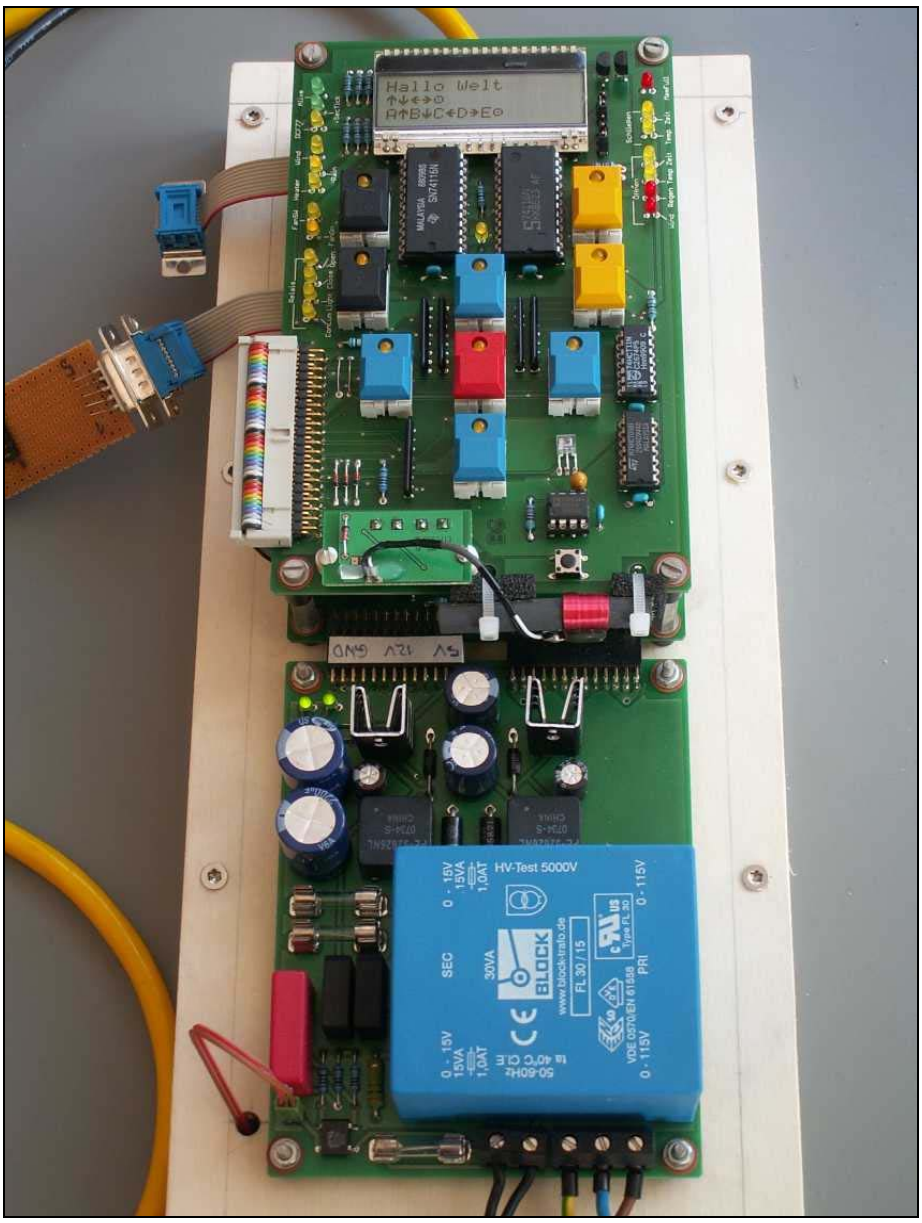


Abbildung 182: ConvCtrl – Das vollständige Komplettsystem

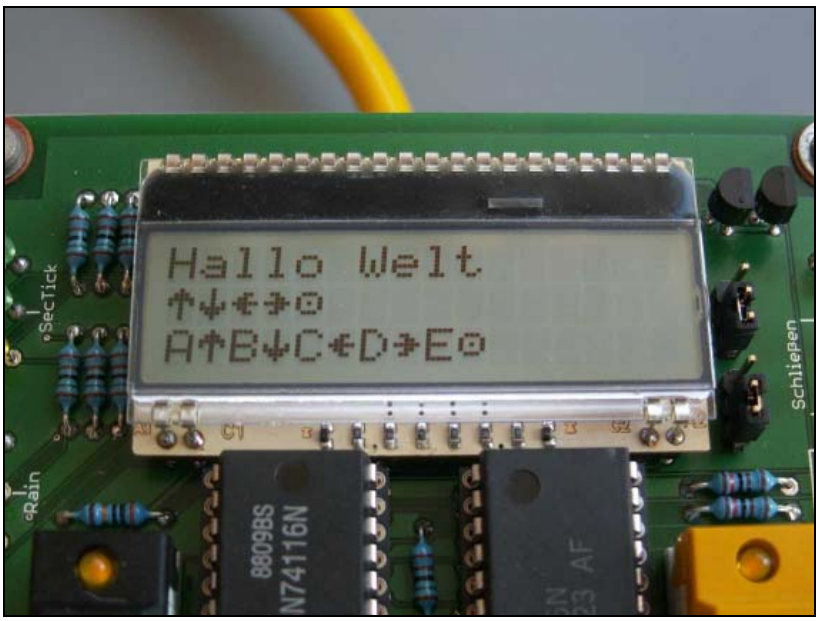


Abbildung 183: ConvCtrl – Nummer EINS lebt!

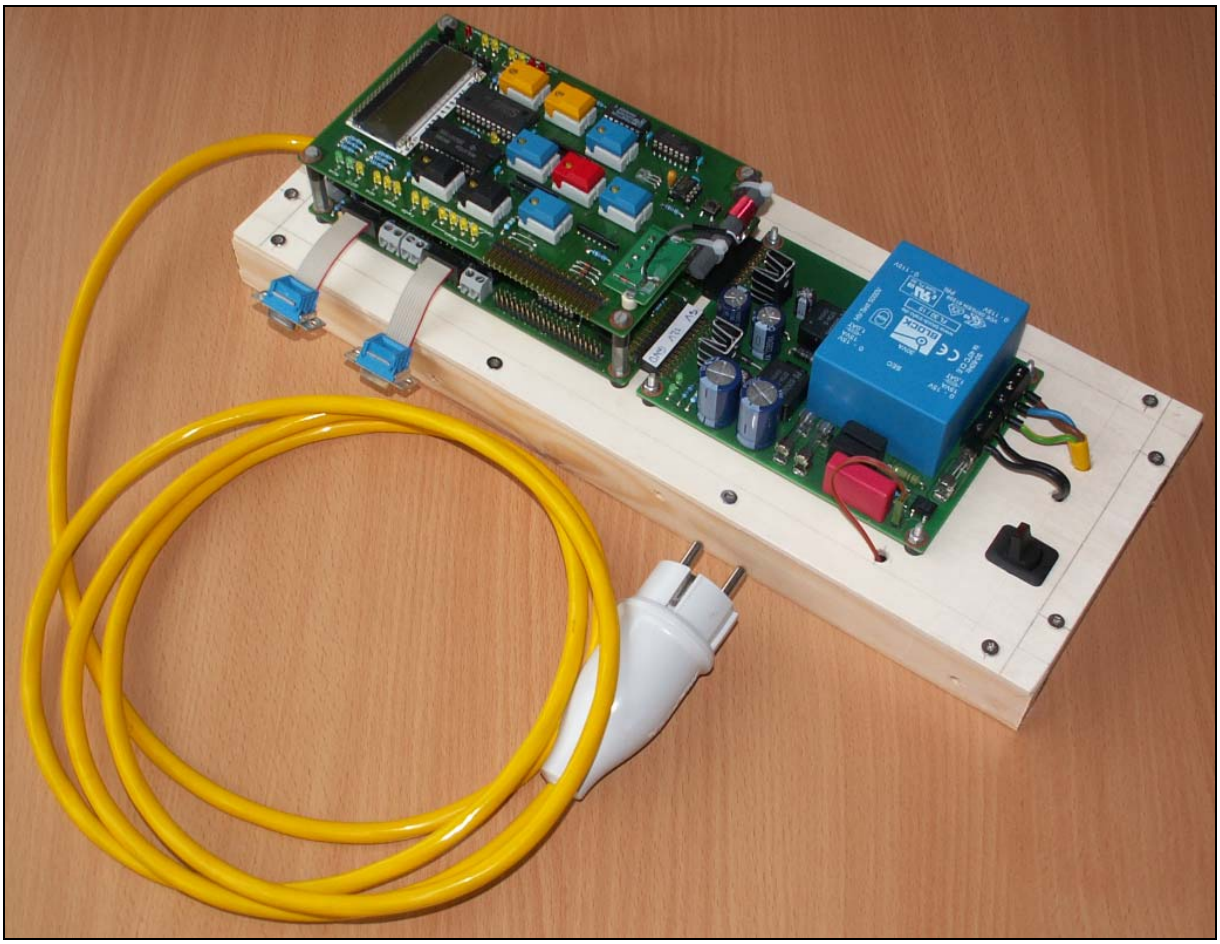
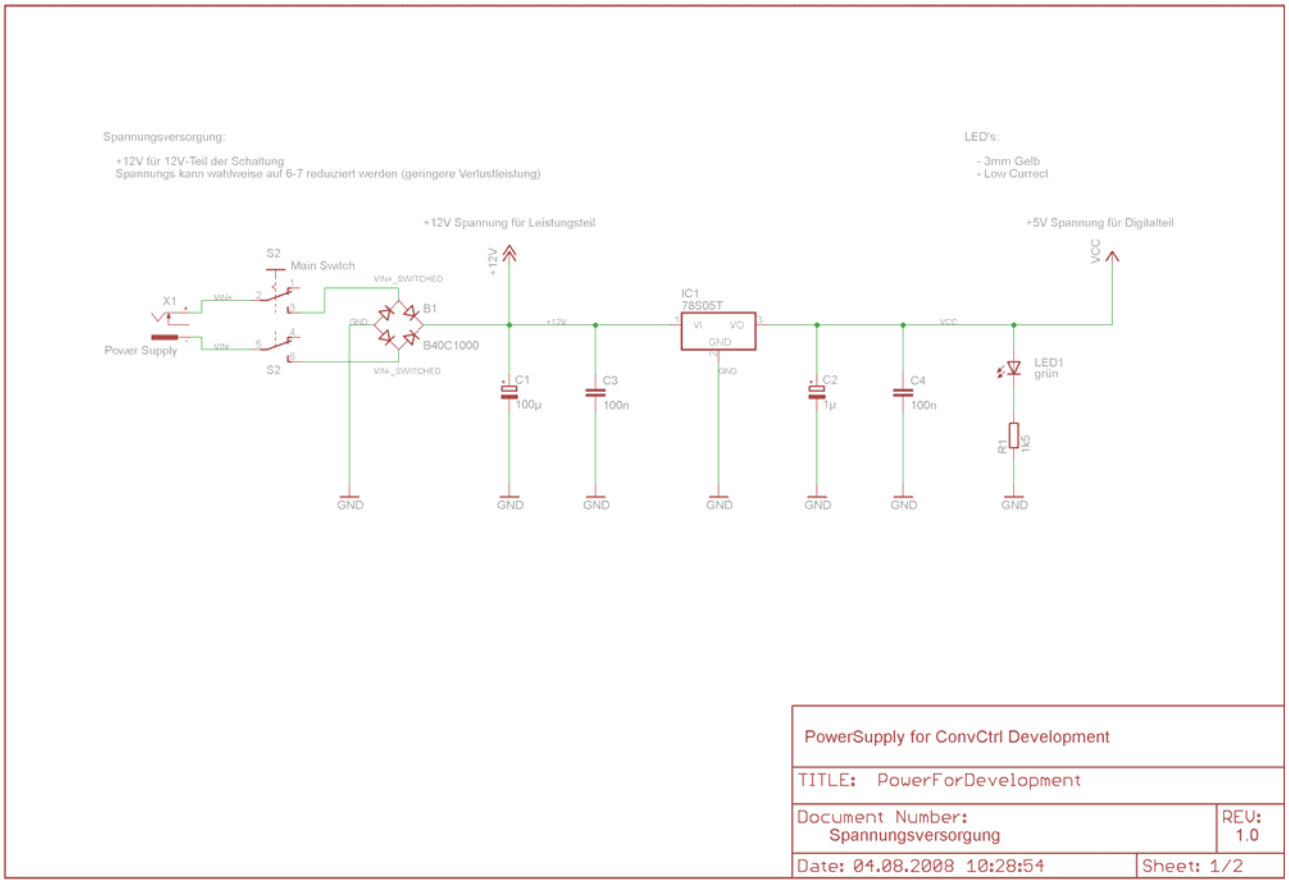


Abbildung 184: ConvCtrl – Die Entwicklungsumgebung

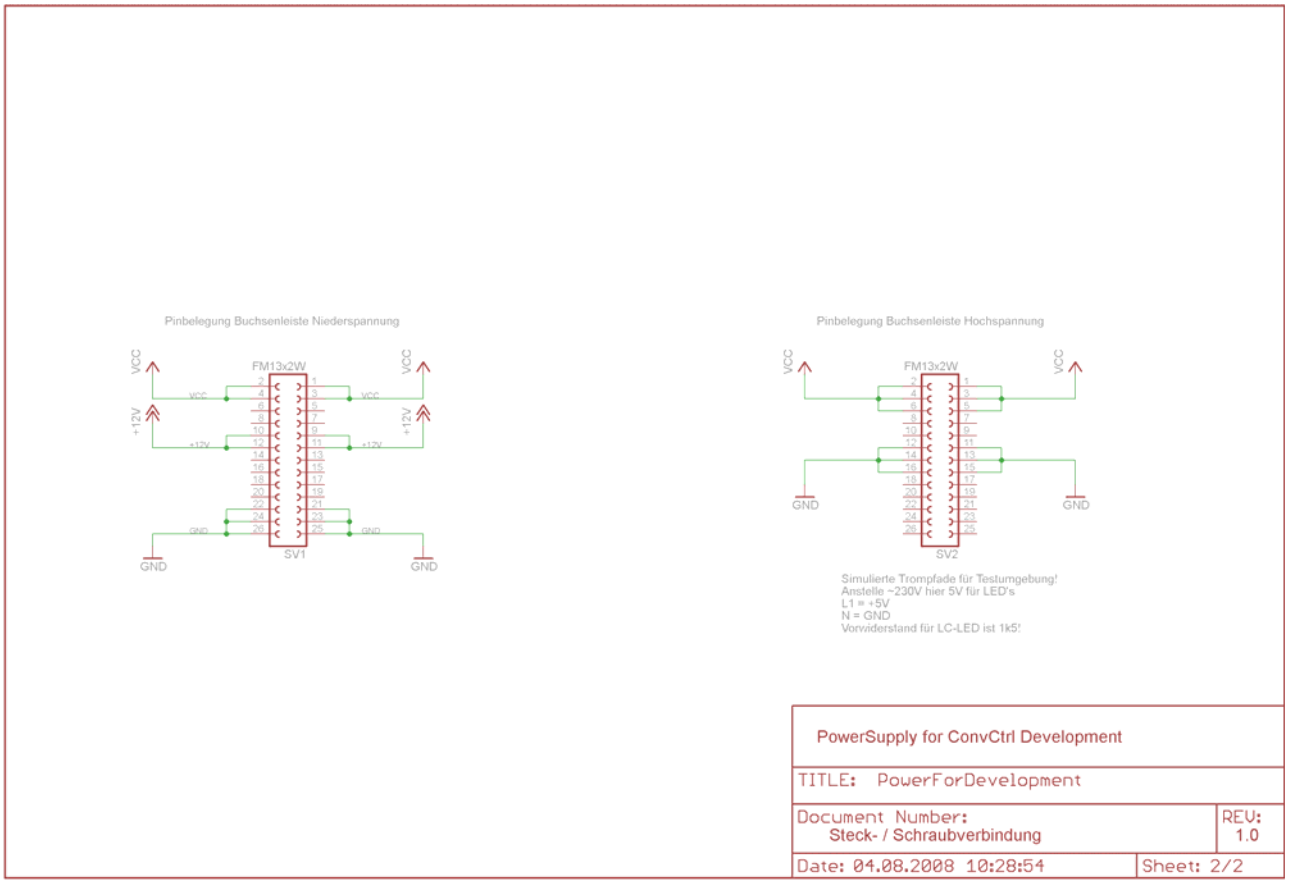
10.8 Versorgungsplatine für Labor

Zur Versogung von PCB1 und PCB2 ohne Netzspannung wurde eine kleine Adapterplatine entwickelt welche kurz zur Vollständigkeit mitdokumentiert wird.

10.8.1 Schematic



Schaltbild 85: Schaltbild PowerForDevelopment – Sheet 1



Schaltbild 86: Schaltbild PowerForDevelopment – Sheet 2

10.8.2 Layout Gesamt

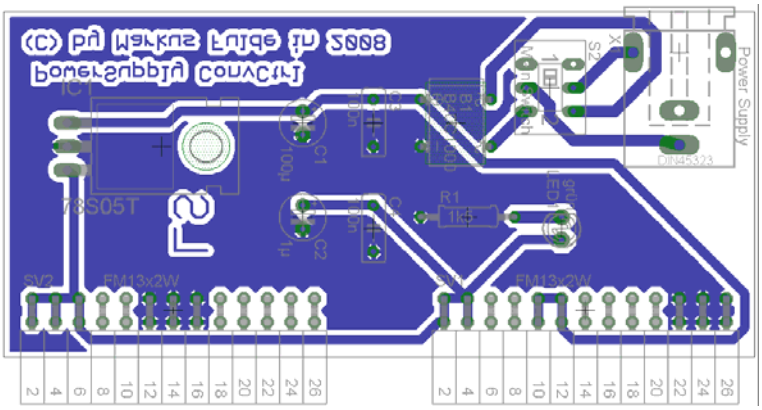


Abbildung 185: PowerForDevelopment – Layout gesamt

10.8.3 Bottom-Layer

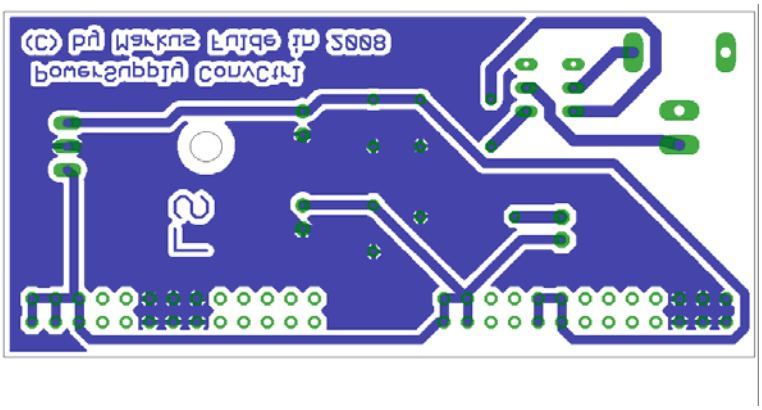


Abbildung 186: PowerForDevelopment – Bottom-Layer

10.8.4 Pad's und Via's

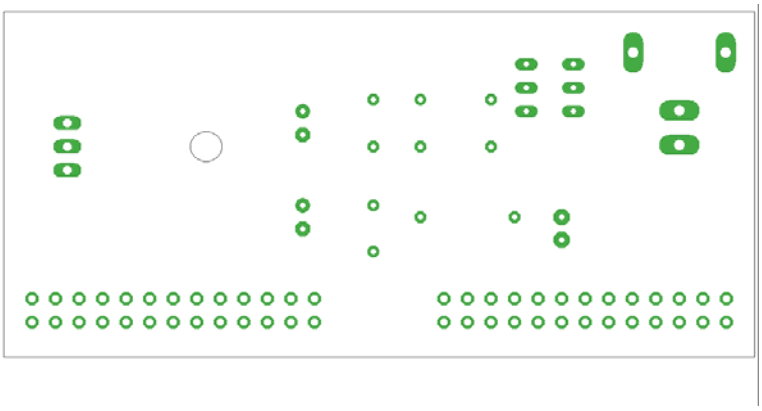


Abbildung 187: PowerForDevelopment – Pad's und Via's

10.8.5 Bestückungsdruck

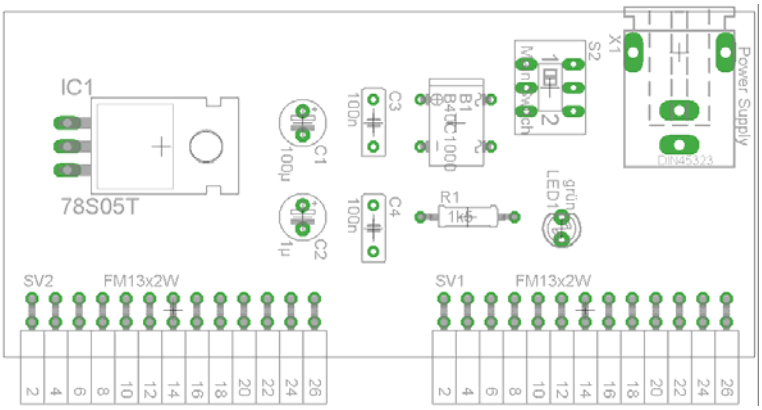


Abbildung 188: PowerForDevelopment – Bestückungsdruck

10.8.6 Restricted Areas

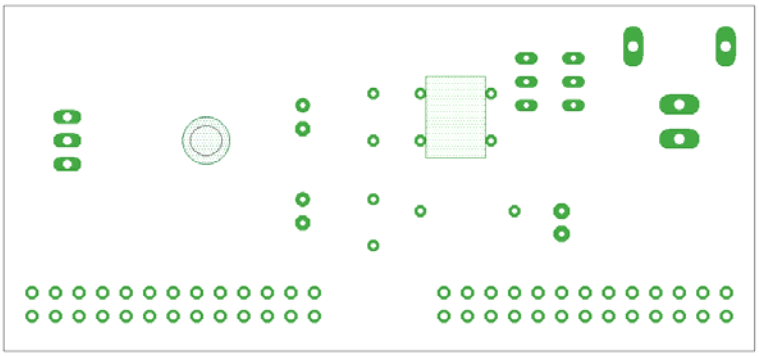


Abbildung 189: PowerForDevelopment – Restricted Areas

10.8.7 Eagle-BOM

Pos.		Bauteil	Wert	Device	Package
1.		B1	B40C1000	B-DIL	B-DIL
2.		C1	100µ	POLARIZEDCAPACITORE2.5-5	E2,5-5
3.		C2	1µ	POLARIZEDCAPACITORE2.5-5	E2,5-5
4.		C3	100n	CAPACITOR050-025X075	C050-025X075
5.		C4	100n	CAPACITOR050-025X075	C050-025X075
6.		IC1	78S05T	7805T	TO220H
7.		LED1	grün	LED3MM	LED3MM
8.		R1	1k5	RESISTOR0207/10	0207/10
9.		S2	Main Switch	A68-A31	A68-A31
10.		SV1	FM13x2W	FE13-2W	FE13-2W
11.		SV2	FM13x2W	FE13-2W	FE13-2W
12.		X1	Power Supply	737992-55	737992-5

Tabelle 153: Eagle BOM für PowerForDevelopment

10.8.8 Die Versorgungsplatine

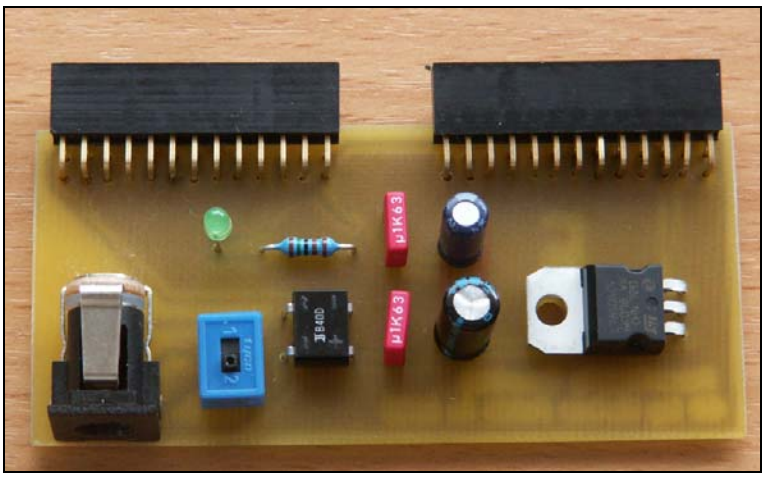


Abbildung 190: ConvCtrl – PowerForDevelopment

11 Software

11.1 Systemfestlegungen und Definitionen

11.1.1 Timerfestlegungen

Die Timer im Gesamtsystem haben der Priorität nach die folgende Reihenfolge:

1. Timer2
2. Timer1
3. Timer0
4. Timer3

Timer0:

~~Der Timer3 ist im ATmega128 der Timer mit der niedrigsten Priorität. Mit seiner Hilfe wird ein SW-Timer aufgebaut. Der Timer3 versorgt das Gesamtsystem mit einem 1-Sekunden-Timertick. Zum Zeitpunkt der Implementierung ist es auch möglich den Tick auf 2 oder 5 Sekunden zu erhöhen um die Last von ISR und SW-Timer-Sequencer zu senken.~~

Timer1:

Der Timer1 wird für die Konfiguration und Ansteuerung der DCF77 Funkuhr verwendet.

Timer2:

Der Timer2 hat die höchste Priorität im System. Mit ihm wird der Dimmer, genauer gesagt der TRIAC-Zündimpuls gerechnet. Nach einer Nulldurchgangserkennung des Netspannungs-Sinus wird der Timer2 mit dem zur Helligkeit korrespondierenden Wert geladen. Sobald der Timer abgelaufen ist wird über die Timer-ISR der Triac gezündet.

Genauere Erläuterungen und Beschreibungen sowie der Beispielcode ist im Kapitel zum Dimmer enthalten.

Timer3:

Der Timer3 wird für die Puls-Weiten-Modulation (PWM) verwendet.

OC3A: Er dient zur Steuerung der Display-Helligkeit. Genauere Erläuterungen und Beschreibungen sowie der Beispielcode ist im Kapitel zum LCD-Display enthalten.

OC3B: Er dient zur Steuerung der Lüftergeschwindigkeit. Genauere Erläuterungen und Beschreibungen sowie der Beispielcode ist im Kapitel zum Lüfter enthalten.

11.1.2 Der Watchdog-Timer

Der verwendete ATmega128 besitzt einen Watchdog-Timer. Der Watchdog-Timer wird durch einen internen On-chip Oszillator mit 1 MHz Taktfrequenz getacktet. Die Taktfrequenz ist aber abhängig von der Versorgungsspannung des Atmegas. Die Angaben von 1 MHz beziehen sich auf V_{CC} = 5V.

Über den Watchdog Timer Prescaler kann das Timer Intervall in dem der Timer den Watchdog auslöst eingestellt werden. Der Bereich liegt zwischen 16 und 2048 ms. Einige AVRs bieten die Möglichkeit an, den Prescaler auf 4098 oder 8192 einzustellen.

Der Basic-Compiler BASCOM bietet einige Befehle für die Konfiguration des Watchdog an:

```
CONFIG WATCHDOG = time
START WATCHDOG
STOP WATCHDOG
RESET WATCHDOG
```

Die Veränderungen via SW (Start und Stop) setzen aber voraus, dass die Fuse WDTON nicht programmiert ist. Mit dieser Fuse wird festgelegt, dass der Watchdog immer enabled ist.

M103C	WDTON	Safety Level	WDT Initial State	How to Disable the WDT	How to Change Time-out
Unprogrammed	Unprogrammed	1	Disabled	Timed sequence	Timed sequence
Unprogrammed	Programmed	2	Enabled	Always enabled	Timed sequence
Programmed	Unprogrammed	0	Disabled	Timed sequence	No restriction
Programmed	Programmed	2	Enabled	Always enabled	Timed sequence

Tabelle 154: Watchdog Konfiguration als Funktion von M103C und WDTON

Es können 8 verschiedene Takt-Zyklen ausgewählt werden welche zum Auslösen des Resets führen.

WDP2	WDP1	WDP0	Number of WDT Oscillator Cycles	Typical Time-out at V_{CC} = 3.0V	Typical Time-out at V_{CC} = 5.0V
0	0	0	16K (16,384)	14.8 ms	14.0 ms
0	0	1	32K (32,768)	29.6 ms	28.1 ms
0	1	0	64K (65,536)	59.1 ms	56.2 ms
0	1	1	128K (131,072)	0.12 s	0.11 s
1	0	0	256K (262,144)	0.24 s	0.22 s
1	0	1	512K (524,288)	0.47 s	0.45 s
1	1	0	1,024K (1,048,576)	0.95 s	0.9 s
1	1	1	2,048K (2,097,152)	1.9 s	1.8 s

Tabelle 155: Codierwerte für den Watchdog Timer

Da die Programmierung des Projekts in BASCOM BASIC erfolgt wird an dieser Stelle auf das ATmega 128 Data Sheet verwiesen in dem die genauen Inhalte der Register des Watchdog Timers und die Zusammenhänge beschrieben sind.

Beispielcode für die Verwendung / Programmierung des Watchdog:

```
'-----------------------------------------------------------------------------------------
'name                     : watchd.bas
'copyright                : (c) 1995-2005, MCS Electronics
'purpose                  : demonstrates the watchdog timer
'micro                    : Mega48
'suited for demo          : yes
'commercial addon needed  : no
'-----------------------------------------------------------------------------------------

$regfile = "m48def.dat"                                    ' specify the used micro
$crystal = 4000000                                         ' used crystal frequency
$baud = 19200                                              ' use baud rate
$hwstack = 32                                              ' default use 32 for the hardware stack
$swstack = 10                                              ' default use 10 for the SW stack
$framesize = 40                                            ' default use 40 for the frame space

Config Watchdog = 2048                                     'reset after 2048 mSec
Start Watchdog                                             'start the watchdog timer
Dim I As Word
For I = 1 To 1000

  Print I                                                  'print value
  'Reset Watchdog
  'you will notice that the for next doesnt finish because of the reset
  'when you unmark the RESET WATCHDOG statement it will finish because the
  'wd-timer is reset before it reaches 2048 msec
Next
End
```

Software 16: Code zum Bedienen des Watchdog-Timers

11.2 PWM-Signale mit Bascom erzeugen

11.2.1 Grundbegriffe

Bei der Puls-Weiten-Modulation (PWM) wird ein digitales Ausgangssignal erzeugt, dessen Tastverhältnis moduliert wird.
Das Tastverhältnis gibt das Verhältnis der Länge des eingeschalteten Zustands zur Periodendauer an. Dabei bleiben die Frequenz und der Pegel des Signals immer gleich! Es ändert sich nur die Länge von High zu Low.

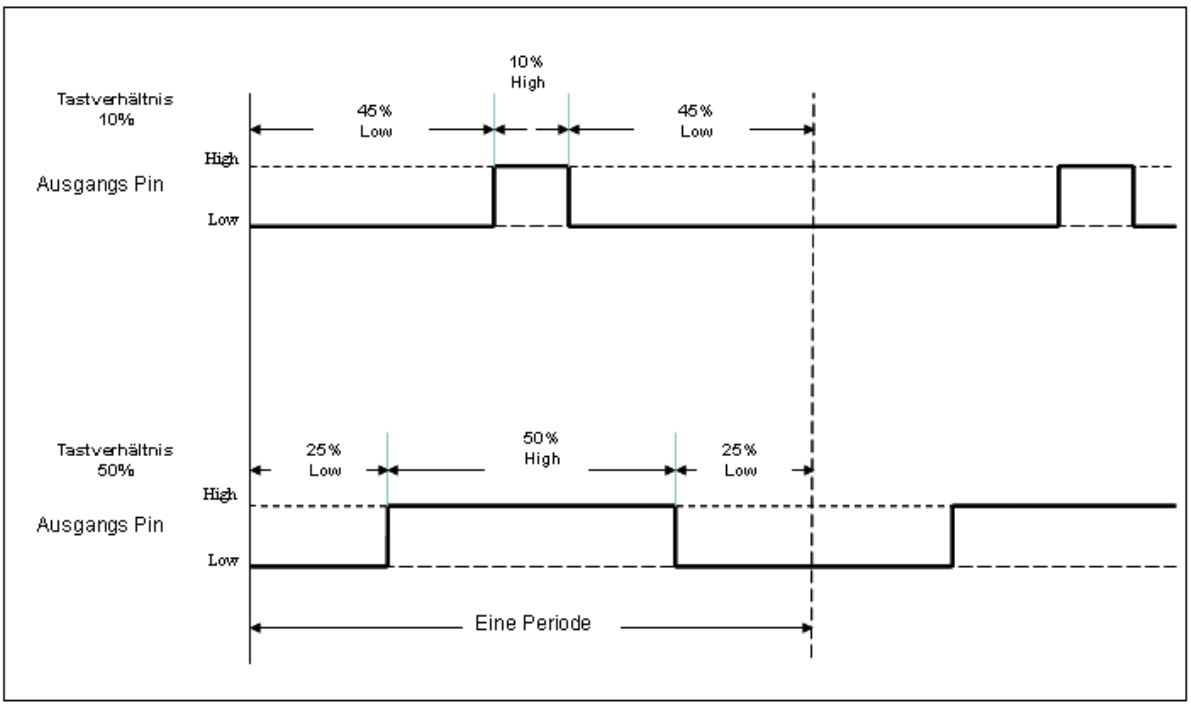


Abbildung 191: PWM Tastverhältnis einmal von 10% und einmal von 50%

Man könnte das in etwa mit einem Wasserhahn vergleichen, den man z.B. alle Minuten betätigt. Wenn man ihn in jeder Minute nur kurz aufdreht und dann gleich wieder zumacht, kommt in Summe nur wenig Wasser raus. Wenn man ihn aber in jeder dieser Minuten länger offen lässt, kommt mehr Wasser raus.

Der Rhythmus bleibt immer gleich, aber es ändert sich in Summe die Wassermenge, die raus kommt.

Mit dem PWM-Signal kann man nun tolle Sachen machen.
Zum Beispiel:

- eine LED (oder über einen Transistor auch eine Lampe) an den PWM-Ausgang anschliessen und mit der Länge des PWM-Signal's die Helligkeit der LED verändern.
- einen Motor in der Geschwindigkeit regeln.
- mittels nachgeschaltetem RC-Filter, welcher das PWM-Signal glättet, kann eine Gleichspannung erzeugt werden die zwischen 0V und 5V geregelt werden kann.

11.2.2 PWM-Arten

Es gibt zwei Arten PWM-Signale in Bascom zu erzeugen:

Software PWM

Vorteile:

- es kann (fast) jeder Ausgabe-Pin des AVR benutzt werden.
- unter zuhilfename eines (freien) Timers können sogar mehrere verschiedene PWM-Signale auf verschiedene Pins erzeugt werden.

Nachteil:

- Etwas grösserer Programmaufwand, da der PortPin per Software verändert werden muss.

Hardware PWM

Vorteile:

- Sehr schnell (Maximal die Quarzfrequenz / Periode)
- unabhängig vom Programmablauf des AVR

Nachteile:

- Je nach AVR können nur bestimmte Timer mit bestimmten Ausgangspins dafür verwendet werden.
- belegt den Timer, der für keine weiteren funktionen verwendet werden kann.

Beim einem ATMega8 stehen drei Hardware-PWM-Ausgänge verteilt auf zwei Timer zur Verfügung.

Mit Timer1 können zwei PWM Signale erzeugt werden (Compare A => OC1A - Pin 15 und Compare B => OC1B - Pin 16).

Die Auflösung kann auf 8, 9 und 10 Bit eingestellt werden, also max. 1024 Abstufungen.

Timer2 kann ein PWM-Signal mit einer Auflösung von 8 Bit erzeugen (Compare Register => OC2 - Pin 17)

11.2.3 PWM-Ablauf

Das folgende Bild zeigt den Ablauf bei Timer1. Als Taktquelle dient die CPU-Frequenz, dessen Frequenz im Prescaler (Vorteiler) nochmal verkleinert werden kann. Je nach eingestelltem Wert in den Output Compare Registern wird der Status des Ausgangs-Pin entsprechend oft umgeschaltet, und erzeugt somit das PWM-Signal.

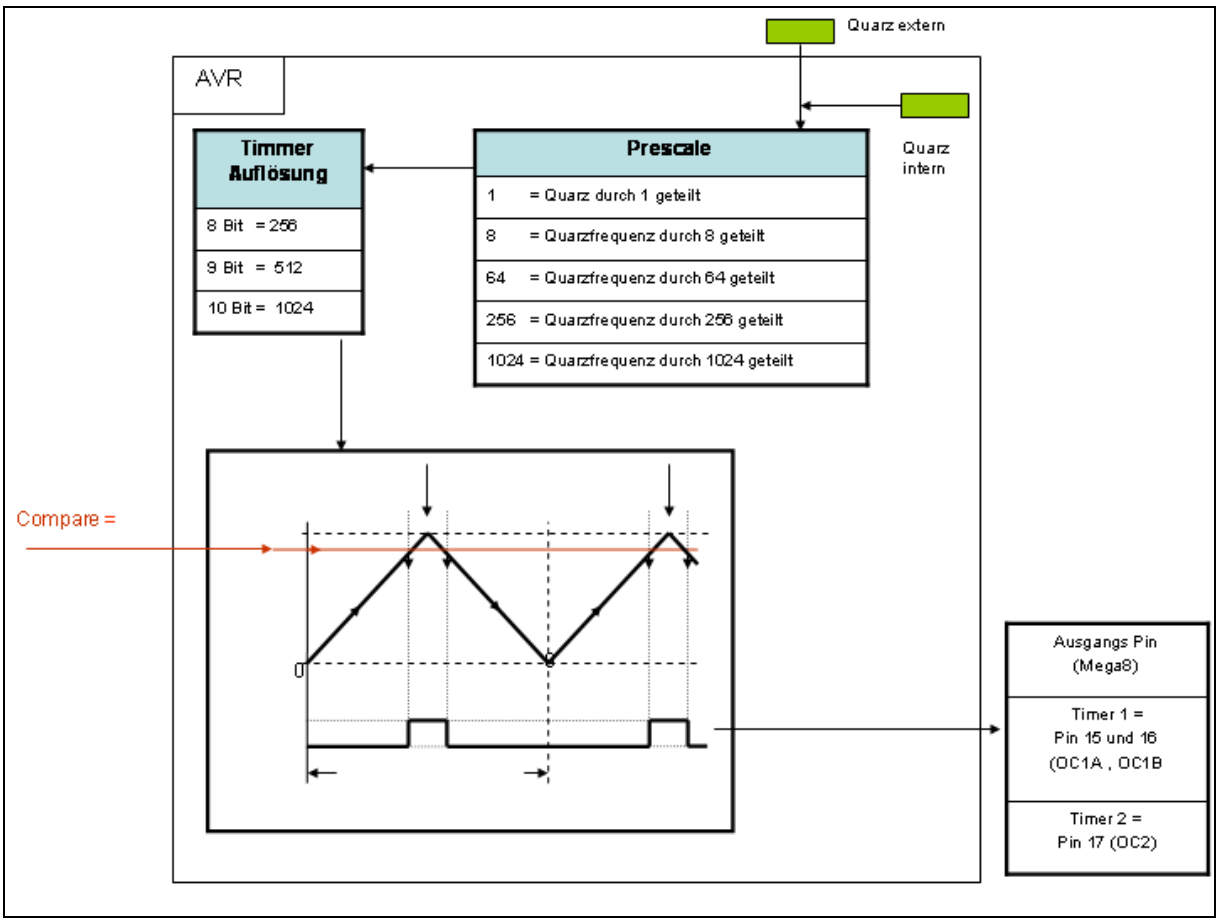


Abbildung 192: PWM Konfiguration des ATmega

11.2.4 Genauere Erklärung

Der Prescaler teilt die Frequenz die vom Quarz kommt! Bei Teilung 1 geht die vollständige Frequenz zum Timer. Bei Teilung 8 nur ein Achtel der Quarzfrequenz. (Also bei 8 MHz Quarz kommen zum Timer dann nur 1 MHz) Bei Teilung 1024 sind es dann z.B. 8 MHz/1024 = 7,8125kHz

Der Timer ist hier der Zähler für die PWM. Er zählt mit der Frequenz, die vom Prescaler kommt, einmal von 0 bis zu der eingestellten Timerauflösung rauf, dann wieder auf 0 zurück. (dann wieder von 0 auf Timerauflösung u.s.w.) Einmal rauf- und runterzählen, ergibt ein Periode. Die Periode ist gleich die Ausgangsfrequenz des PWM-Signals. Ausgangsfrequenz = (Quarzfrequenz/Prescale) /(Timerauflösung*2)

z.B.: Quarz = 8 MHz ; Prescaler = 1 ; Timer = 8 Bit ergibt: (8000000Hz/1) / (256*2) = 15,625 kHz

oder: Quarz = 8 MHz ; Prescaler = 8 ; Timer = 10 Bit ergibt: (8000000Hz/8) / (1024*2) = 244,14 Hz

Mit dem Compare Register definiert man nun das Tastverhältnis! Überall, wo nun der Timer diese Compare Linie schneidet, schaltet der Ausgang! Beim raufzählen des Timers auf EIN, beim runterzählen auf AUS.

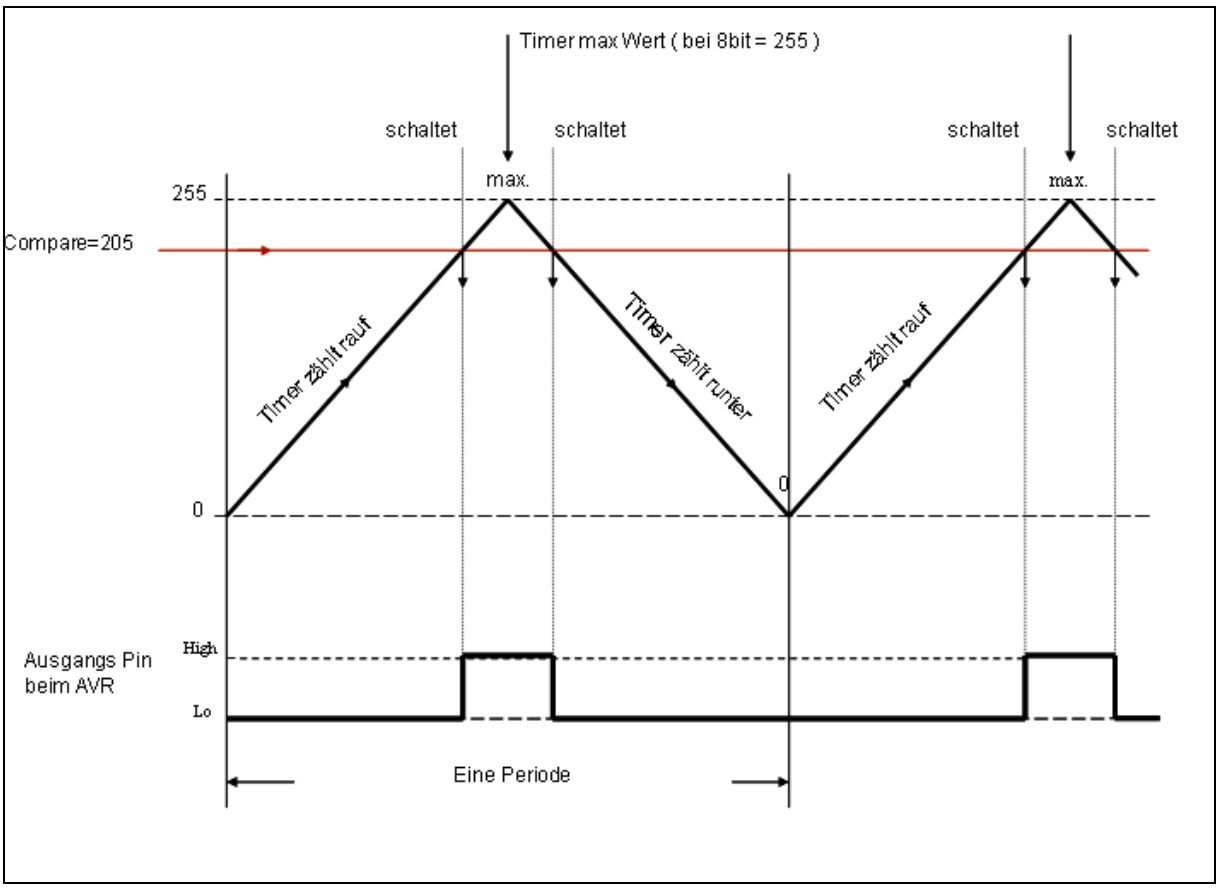


Abbildung 193: PWM mit einem Tastverhältnis 20%

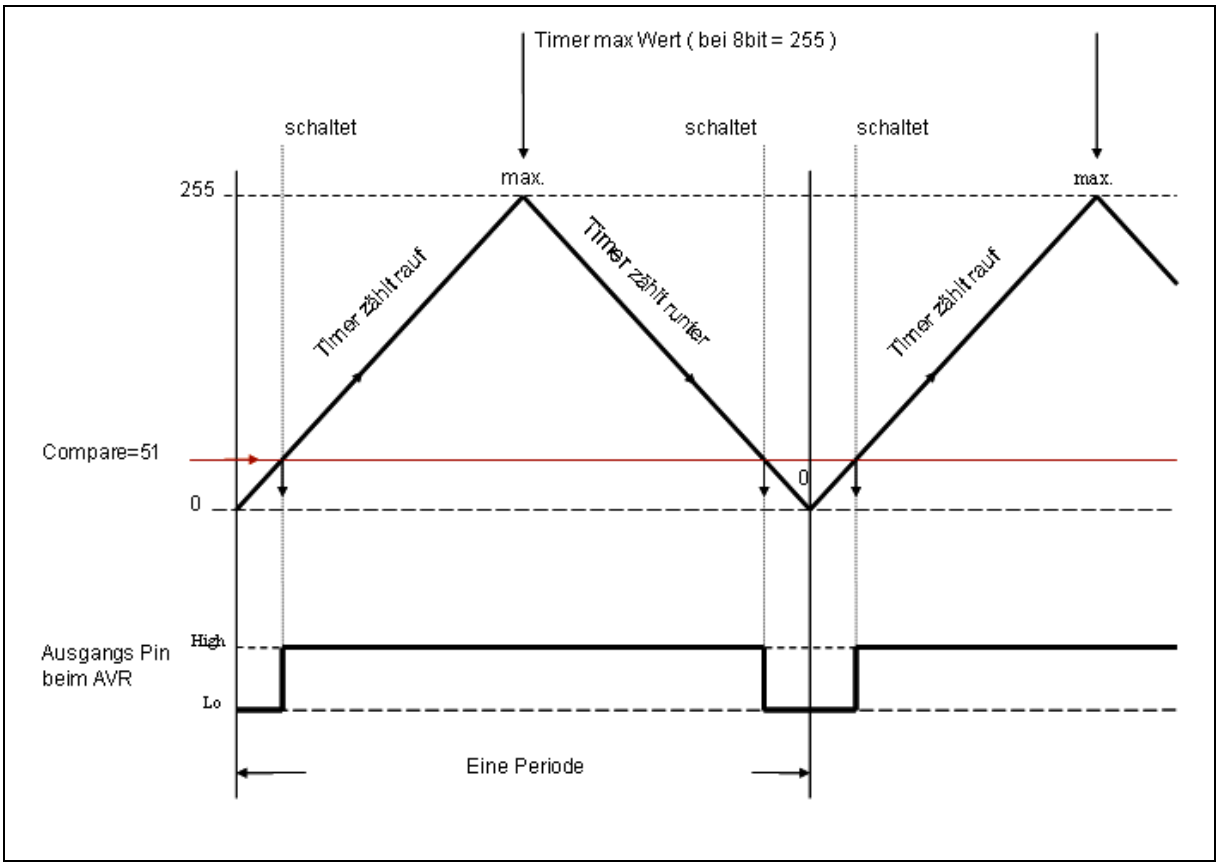


Abbildung 194: PWM mit einem Tastverhältnis von 80%

Hier sieht man, wie die Signale auf einem Oszilloskop ausschauen. Oben das Signal von Pin15 (Compare A), unten das von Pin 16 (Compare B)

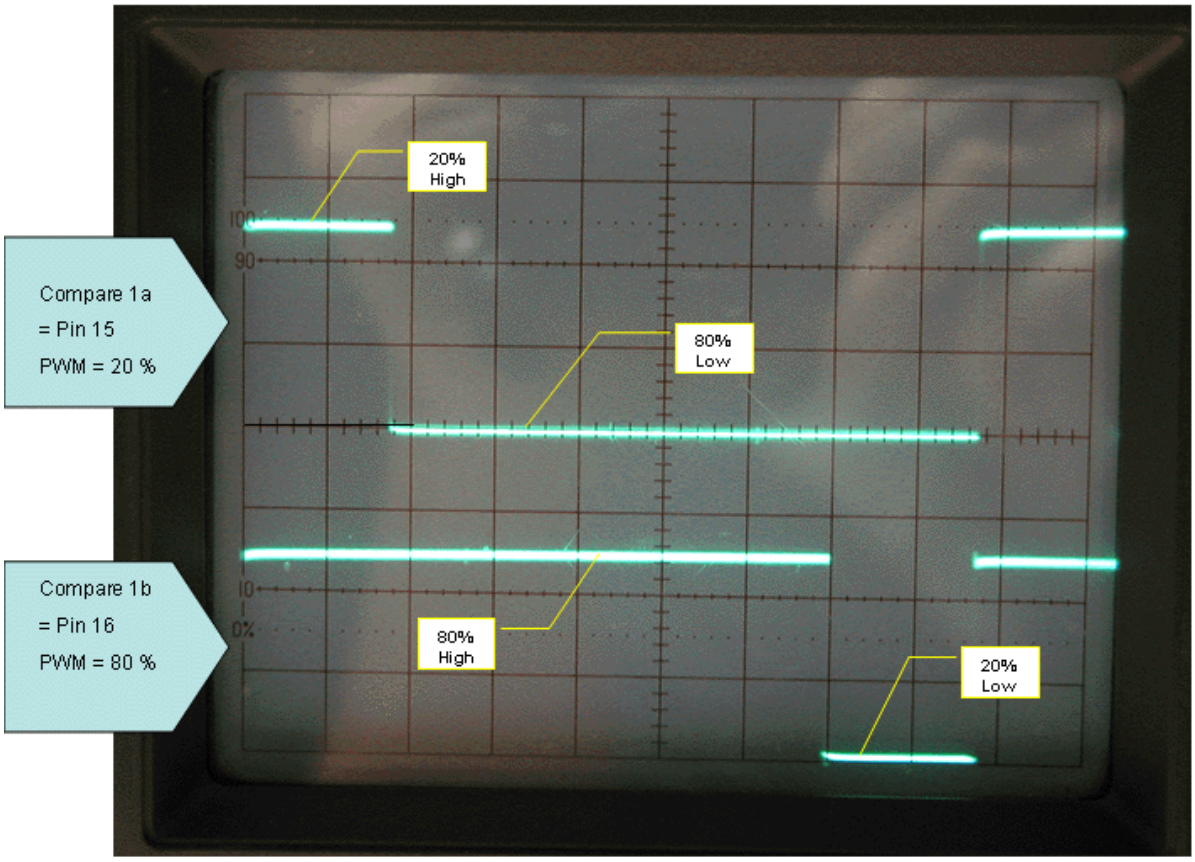


Abbildung 195: Oszillogramm der PWM mit Time 1a und 1b

11.2.5 Grundprogramm

Hier nun ein Grundprogramm für die Ausgabe von zwei PWM Signalen mit dem Timer1
' Hardware PWM mit Timer1

```
$regfile = "m8def.dat"
$crystal = 4000000

Config Portb.1 = Output
Config Portb.2 = Output

Config Timer1 = Pwm , Pwm = 8 , Compare A Pwm = Clear Up , Compare B Pwm =
Clear Up , Prescale = 1

Do
 Compare1a = 205
 Compare1b = 51
Loop

End
```

Erklärung:

```
$regfile = "m8def.dat"
$crystal = 8000000
```

Definiert den Mega8 und den 8MHz Quarz

```
Config Portb.1 = Output
Config Portb.2 = Output
```

Definiert die zwei Ausgänge von Timer1 auf Ausgabe.
Portb.1 = für Compare1a (= Compare A) = Pin 15
Portb.2 = für Compare1b (= Compare B) = Pin 16

```
Config Timer1 = Pwm , Pwm = 8 , Compare A Pwm = Clear Up , Compare B Pwm =
Clear Up , Prescale = 1
```

Config Timer1 = Pwm
: Timer1 auf PWM einstellen

Pwm = 8
: Timer Auflösung = 8 Bit einstellen

Compare A Pwm = Clear Up
: Definiert, wie der Compare A schalten soll. Bei „Clear Up" schaltet der Ausgang beim erreichen des Compare-Wertes zuerst auf High und dann auf Low. Bei „Clear Down", umgekehrt.

Compare B Pwm = Clear Up
: Das gleiche noch mal mit Compare B

Prescale = 1
: Hier wird der Prescaler auf 1 eingestellt.
(Wert 1 heißt, direkte Frequenz vom Quarz zum Timer.)
Weitere Teilungen, wie z.B.: 8, 64, 256 und 1024 sind möglich.

```
Compare1a = 205
Compare1b = 51
```

Hier kann man nun die Werte für das Tastverhältnis, in dem Register Compare1a und Compare1b übergeben. Oder man kann, statt Compare1a und 1b, auch die Bezeichnungen Pwm1a und Pwm1b verwenden, Bascom nimmt beides.

Mit diesem kurzen Programm, hat man nun zwei PWM Signale erzeugt, bei dem eines ein Tastverhältnis von 20% (Compare1a) und das andere 80 % hat. :-)

11.3 Der Source-Code zum Projekt ConvCtrl

```
'#####################################################################################
' ConvCtrl.BAS:                                                    Stand 11.12.2008
' -------------                                                    (C) Markus Fulde
'
' Gesamtsoftware Steuerprogramm für Wintergartensteuerung ConvCtrl
'
' Das Projekt ConvCtrl verfügt über die folgenden einzelnen Funktions- und
' Teilkomponenten:
'      - Spannungsversorgung für +5V und +12V
'      - Mikrocontroller ATmega32 oder ATmega128 (inkl. ISP, RS232 und Reset)
'      - LCD-Display 3x16 (inkl. Rahmen mit "Scheibe")
'      - Tastatur und Schalter
'      - LED-Anzeigen für Status- und Funktionsanzeige
'      - Windmesser (pro Tick = 2,5m/sec Windgeschwindigkeit)
'      - Regensensor (ja/nein Detektion inkl. Detektorheizung)
'      - Temperatursensor (Anschluss über I2C Bus)
'      - Luftfeuchtigkeit (Anschluss über I2C Bus)
'      - Dimmungssteuerung für Beleuchtung (inkl. ein/aus Relais)
'      - Motorsteuerung für Beschattung (inkl. Verriegelung und Relais-Timeout)
```

```
'     - Gehäusebelüftung mit Option Lüftersteuerung über ATmega
'     - Datenspeicher / Datenlogger über EEPROM (Anschluss über I2C Bus)
'     - DCF77 Funkuhrempfang (Conrad)
'     - Passendes Selbstbaugehäuse aus Aluminium und rauchgläserner Front
'     - Helligkeitsmessung (Licht -> DC-Wandler mit OPAmp)
'     - Anzeige von Zeit- und Datumsinformationen
'     - Akustisches Feedback über Piezo Schallwandler
'     - Selbstcheck bei StartUp und ggf. Ausgabe von Warnmeldungen und Signalen
'     - Timersteuerung zeitkritischer Applikationen (Dimmer, LCD-Helligkeit,..)
'
'###############################################################################


'-------------------------------------------------------------------------------
' Compilerinstruktionen und Compilerdirektiven
'-------------------------------------------------------------------------------
$regfile = "m128def.dat"                                    ' Definitionsdatei für ATmega128 laden
$crystal = 16000000                                         ' Quarzfrequenz für 16 MHz festlegen
$hwstack = 128                                              ' HW-Stack auf 128 Bytes erweitern
$swstack = 64                                               ' SW-Stack auf 64 Bytes erweitern
$framesize = 80                                             ' Framesize auf 80 Byte festlegen

' Serielle Schnittstelle wird für Tracedatenausgabe verwendet
$baud = 38400                                               ' Baudrate für RS232 auf 38400 Bauf
festlegen

' Einbinden der I2C-Library für das Hardware-Interface des ATmega
$lib "i2c_twi.lbx"                                          ' we do not use software emulated I2C
but the TWI


'-------------------------------------------------------------------------------
' Definition von Ressourcen
'-------------------------------------------------------------------------------

' ----- TRACE:  Trace-Jumper
Trace_pin Alias Pinb.7                                      ' für Trace-Jumper-Abfrage (für DDR oder
Input)
Trace Alias Portb.7                                         ' für Trace-Jumper-Abfrage (für Output
oder PullUp)


' ----- SOUND:  Piezo Soundausgabe -----
' Port und Pin für Soundsteuerung
Sound_pin Alias Pind.5                                      ' für Piezo-Sound (für DDR oder Input)
Sound_dataout Alias Portd.5                                 ' für Piezo-Sound (für Output oder Pul-
lUp)


' ----- LCD: Display -----
' Port und Pin für LCD Display einschalten
Lcd_on_pin Alias Ping.1                                     ' um LCD einzuschalten (Transistorstufe)
(für DDR oder Input)
Lcd_on_dataout Alias Portg.1                                ' um LCD einzuschalten (Transistorstufe)
(für Output oder PullUp)


' ----- KEY: Tastatur -----
' Zeilen
Key_zeile_1_pin Alias Pina.0                                ' PortA.0 ist Zeile 1 (für DDR oder
Input)
Key_zeile_2_pin Alias Pina.1                                ' PortA.1 ist Zeile 2 (für DDR oder
Input)
Key_zeile_3_pin Alias Pina.2                                ' PortA.2 ist Zeile 3 (für DDR oder
Input)

Key_zeile_1_port Alias Porta.0                              ' PortA.0 ist Zeile 1 (für Output oder
PullUp)
Key_zeile_2_port Alias Porta.1                              ' PortA.1 ist Zeile 2 (für Output oder
PullUp)
Key_zeile_3_port Alias Porta.2                              ' PortA.2 ist Zeile 3 (für Output oder
PullUp)

' Spalten
Key_spalte_1_pin Alias Pinf.1                               ' PortF.1 ist Spalte 1 (für DDR oder
Input)
Key_spalte_2_pin Alias Pinf.2                               ' PortF.2 ist Spalte 2 (für DDR oder
Input)
Key_spalte_3_pin Alias Pinf.3                               ' PortF.3 ist Spalte 3 (für DDR oder
Input)
```

```
Key_spalte_1_port Alias Portf.1                          ' PortF.1 ist Spalte 1 (für Output oder
PullUp)
Key_spalte_2_port Alias Portf.2                          ' PortF.2 ist Spalte 2 (für Output oder
PullUp)
Key_spalte_3_port Alias Portf.3                          ' PortF.3 ist Spalte 3 (für Output oder
PullUp)


' ----- REL: Licht und Relais -----
' REL: Alias für Port und PIN der Relais-Steuerung
Out_auf Alias Pinc.6                                     ' Relais Beschattung auf (für DDR oder
Input)
Out_zu Alias Pinc.7                                      ' Relais Beschattung zu (für DDR oder
Input)
Out_licht Alias Ping.0                                   ' Relais Licht an (für DDR oder Input)
Out_dauerlicht Alias Pine.2                              ' Relais Dauerlicht an (für DDR oder
Input)

Rel_auf_on Alias Portc.6                                 ' Relais Beschattung auf (für Output
oder PullUp)
Rel_zu_on Alias Portc.7                                  ' Relais Beschattung zu (für Output oder
PullUp)
Rel_licht_on Alias Portg.0                               ' Relais Licht an (für Output oder Pul-
lUp)
Rel_dauerlicht_on Alias Porte.2                          ' Relais Dauerlicht an (für Output oder
PullUp)

' DIM: Alias für Dimmer_Steuerung
Null_detect Alias Pine.5                                 ' für Input-PIN (für DDR oder Input)
Null_detect_pin Alias Porte.5                            ' für PullUp (für Output oder PullUp)

Ignition Alias Pind.4                                    ' für Triac-Zündung (für DDR oder Input)
Triac_start Alias Portd.4                                ' für Triac-Zündung (für Output oder
PullUp)


' ----- LED:  Stand-Alone-LED's und LED Array -----
' Stand-Alone-LED's
Alive_led_pin Alias Pinb.0                               ' GPIO für Alive-LED (für DDR oder In-
put)
Alive_led Alias Portb.0                                  ' GPIO für Alive-LED (für Output oder
PullUp)

Sectick_led_pin Alias Pinb.6                             ' GPIO für Sekundentakt-LED (für DDR
oder Input)
Sectick_led Alias Portb.6                                ' GPIO für Sekundentakt-LED (für Output
oder PullUp)
'Anmerkung: LED wird über SECTIC getoggelt. Frequenz ist damit 0,5 Hz!


' ----- RAIN: Regensensor -----
Rain_sensor Alias Ping.3                                 ' Anschluss an PinG.3 (für DDR oder
Input)
' Anmerkung: Regensensor wird über Schmitt-Rigger geleitet. PullUp ist prinzipiell nicht notwendig.
'            Pull ist aber nach Reset und Input-Config via Default aktiv!


' ----- SHT71:  Temperatur- und Luftfeuchtigkeit -----
' Ressourcen für Zugriff (Nachbildung für I2C Bus
Sht71_sck Alias Portd.7                                  ' Sck PortD.7 (für Output oder PullUp)
Sht71_dataout Alias Portd.6                              ' Data PortD.6 Output (für Output oder
PullUp)
Sht71_datain Alias Pind.6                                ' Data PortD.6 Input (für DDR oder In-
put)

' Ressourcen für Konfiguration
Temp_sck Alias Pind.7                                    ' Sck PortD.7 (für DDR oder Input)
Temp_data Alias Pind.6                                   ' Data PortD.6 (für DDR oder Input)


' ----- MEGA: Controller-Spezialitäten
' Interrupt-Flag-Register (relevant für Tastatureingabe)
External_interrupt_flag_register Alias Eifr              ' Um Register zurück zu setzen

' TWI-Bus Hardware für RTC und EEPROM
Sda_port Alias Portd.1                                   ' TWI Data = SDA-Pin (für Output oder
PullUp)
Scl_port Alias Portd.0                                   ' TWI Clock = SCL-Pin (für Output oder
PullUp)
```

```
' ----- LED:  LED-Array -----
' Port Input Pin Adress für Datenbits
Led_bit0_pin Alias Pina.3                                   ' PortA.3 ist Daten-Bit0 (für DDR oder
Input)
Led_bit1_pin Alias Pina.4                                   ' PortA.4 ist Daten-Bit1 (für DDR oder
Input)
Led_bit2_pin Alias Pina.5                                   ' PortA.5 ist Daten-Bit2 (für DDR oder
Input)
Led_bit3_pin Alias Pina.6                                   ' PortA.6 ist Daten-Bit3 (für DDR oder
Input)

' Port Data Register für Datenbits
Led_bit0_port Alias Porta.3                                 ' PortA.3 ist Daten-Bit0 (für Output
oder PullUp)
Led_bit1_port Alias Porta.4                                 ' PortA.4 ist Daten-Bit1 (für Output
oder PullUp)
Led_bit2_port Alias Porta.5                                 ' PortA.5 ist Daten-Bit2 (für Output
oder PullUp)
Led_bit3_port Alias Porta.6                                 ' PortA.6 ist Daten-Bit3 (für Output
oder PullUp)

' Port Input Pin Adress für Adressbits
Led_adr0_pin Alias Pina.7                                   ' PortA.7 ist Adressen-Bit 0 (für DDR
oder Input)
Led_adr1_pin Alias Ping.2                                   ' PortG.2 ist Adressen-Bit 1 (für DDR
oder Input)

' Port Data Register für Adressbits
Led_adr0_port Alias Porta.7                                 ' PortA.7 ist Adressen-Bit 0 (für Output
oder PullUp)
Led_adr1_port Alias Portg.2                                 ' PortG.2 ist Adressen-Bit 1 (für Output
oder PullUp)

' Bemerkung:
' Durch die Verteilung der Adressen auf zwei Ports und damit nicht zusammenhängende
' Bits kommt es bei der Umschaltung der Adresse &B01 nach &B10 zu "Flackereffekten"
' bei denen entweder Adresse &B00 oder &B11 betroffen sind. Es ist im Einzelfall
' zu enscheiden ob zuerst Bit 0 auf LOW oder Bit 1 auf HIGH gezogen wird.
' Dieser Effekt ist nur dadurch zu vermeiden, indem beide Adressen in einem Port
' zusammengefasst werden. Der Zugriff darf dann nicht mit SET oder RESET erfolgen,
' sondern durch direkte Port-Ausgaben bei denen die 2-Bit-Adresse direkt gesetzt wird.

' Port Input Pin Adress für Latch Clear
Led_clr_pin Alias Ping.4                                    ' PortG.4 ist Latch Clear (für DDR oder
Input)

' Port Data Register für Latch Clear
Led_clr_port Alias Portg.4                                  ' PortG.4 ist Latch Clear (für Output
oder PullUp)

' Alias Namen für einzelne LED's im LED-Array
Led_01_sw1 Alias Led_array.0                                ' Bit 0 - LED1
Led_02_sw2 Alias Led_array.1                                ' Bit 1 - LED2
Led_03_sw3 Alias Led_array.2                                ' Bit 2 - LED3
Led_04_sw4 Alias Led_array.3                                ' Bit 3 - LED4
Led_05_sw5 Alias Led_array.4                                ' Bit 4 - LED5
Led_06_sw6 Alias Led_array.5                                ' Bit 5 - LED6
Led_07_sw7 Alias Led_array.6                                ' Bit 6 - LED7
Led_08_sw8 Alias Led_array.7                                ' Bit 7 - LED8
Led_09_sw9 Alias Led_array.8                                ' Bit 8 - LED9
Led_10_mem_full Alias Led_array.9                           ' Bit 9 - LED10
Led_11_open_time Alias Led_array.10                         ' Bit 10 - LED11
Led_12_open_light Alias Led_array.11                        ' Bit 11 - LED12
Led_13_open_rain Alias Led_array.12                         ' Bit 12 - LED13
Led_14_open_wind Alias Led_array.13                         ' Bit 13 - LED14
Led_15_close_time Alias Led_array.14                        ' Bit 14 - LED15
Led_16_close_temp Alias Led_array.15                        ' Bit 15 - LED16


'-------------------------------------------------------------------------------
' Definition von Konstaten
'-------------------------------------------------------------------------------

' Hinweis:
' Zur Ausgabe von Traces wurde auf der PCB1 der Wintergartensteuerung ein
' Jumper mit Kontroll-LED implementiert welcher abgefragt werden kann. Dieser
' Jumper ist in der bisherigen Systemimplementierung jedoch nicht berücksichtigt.
' Er wird initialisiert und abefragt und steht über die globale Variable
```

```
' TRACE_JUMPER_STATUS mit den Werten True und False zur Verfügung. Die bisherigen
' Implementierungen basieren jedoch auf definierten Konstanten.
' Weiter wurde in den Einstellungen auch eine im EEPROM ablegbare Variable
' Ee_data_debug vorgesehen und implementiert. Auch diese Variable wird aktuell
' nicht verwendet.

' ----- Für Testumgebung bzw. Traceausgaben -----
Const Main_testmodus = 0                                    ' Flag für Testmodus Allgemeinsystem
Const Lcd_testmodus = 0                                     ' Flag für Testmodus LCD-Dispaly
Const Fan_testmodus = 0                                     ' Flag für Testmodus Lüfter
Const Sound_testmodus = 0                                   ' Flag für Testmodus Piezo Sound
Const Key_testmodus = 0                                     ' Flag für Testmodus der Tastatur
Const Menue_testmodus = 0                                   ' Flag für Testmodus der Menüs
Const Dcf77_testmodus = 0                                   ' Testmodus für DCF77
Const Rtc_testmodus = 0                                     ' Testmodus für RTC DS1307
Const Dimmer_testmodus = 0                                  ' Testmodus für Dimmer
Const Sequencer_testmodus = 0                               ' Testmodus für Sequencer (Scheduler)
Const Relais_testmodus = 0                                  ' Testmodus für Relaisfunktionen
Const Rain_testmodus = 0                                    ' Testmodus für Regensensor
Const Wind_testmodus = 0                                    ' Testmodus Projekt 1 = aktiv
Const Sht71_testmodus = 0                                   ' Für Trace-Output über RS232
Const Light_testmodus = 0                                   ' Testmodus für Produktiv-Traceausgabe
Const Led_testmodus = 0                                     ' Flag für Testmodus
Const Eeprom_testmodus = 0                                  ' Testmodus für EEPROM
Const Datalogging_testmodus = 0                             ' Testmodus für Datalogging
Const Tracejumper_testmodus = 0                             ' Testmodus für Tracejumper
Const Timer_testmodus = 0                                   ' Testmodus für Timerfunktionen
Const Screensaver_testmodus = 0                             ' Testmodus für Screensaver
Const Shortcut_testmodus = 0                                ' Testmodus für ShortCut-Funktion
Const Flashmode_testmodus = 0                               ' Testmodus für Flashmode


' ----- LCD:  Für LCD-Display -----

' LCD: Konstanten für LCD-Helligkeit (Range 0-255 / 8 Bit)
Const Lcd_schrittweite_helligkeit = 5                       ' Schrittweite für Einzelschritte
Const Lcd_helligkeit_min = 0                                ' minimale Helligkeit
Const Lcd_helligkeit_max = 255                              ' maximale Helligkeit
Const Lcd_default_helligkeit = 60                           ' Default Helligkeit empirisch ermittelt
Const Lcd_helligkeit_aus = 255                              ' Bytewert für LCD-Aus

' LCD: Konstanten für LCD-Kontrast (Range 0-63 / 6 Bit)
Const Lcd_kontrast_min = 0                                  ' minimaler kontrast
Const Lcd_kontrast_max = 63                                 ' maximaler Kontrast
Const Lcd_kontrast_default = 12                             ' Default Kontrast empirisch ermittelt

Const Lcd_screensaver_default = 0                           ' Defaultwert für Screensaver
Const Lcd_screensaver_delay = 60                            ' 60 Sekunden Delay für Screensaver


' ----- FAN:  Für Lüfter -----

' FAN: Konstaten für FAN-Geschwindigkeit (Range 0-255 / 8 Bit)
Const Fan_schrittweite_geschwindigkeit = 5                  ' Schrittweite für Einzelschritte
Const Fan_geschwindigkeit_min = 100                         ' minimale Geschwindigkeit
Const Fan_geschwindigkeit_max = 255                         ' maximale Geschwindigkeit
Const Fan_default_geschwindigkeit = 200                     ' Default Geschwindigkeit
Const Fan_anlaufgeschwindigkeit = 130                       ' Anlaufgeschwindigkeit
Const Fan_belueftung_aus = 0                                ' Bytewert für Belüftung aus

' FAN: PWM-Werte für Lüftersteuerung
' Anmerkung: Geschwindigkeitswerte sind nicht errechnet sondern empirisch ermittelt
Const Fan_speed_25 = 120                                    ' Geschwindigkeitskonstante für 25%
Lüfter
Const Fan_speed_50 = 165                                    ' Geschwindigkeitskonstante für 50%
Lüfter
Const Fan_speed_75 = 210                                    ' Geschwindigkeitskonstante für 75%
Lüfter
Const Fan_speed_100 = 255                                   ' Geschwindigkeitskonstante für 100%
Lüfter

' FAN: Temperaturgrenzwerte für Lüfterstufen
Const Fan_25_default = 20                                   ' 25% Lüftung ab 20 °C
Const Fan_50_default = 35                                   ' 50% Lüftung ab 35 °C
Const Fan_75_default = 50                                   ' 75% Lüftung ab 50 °C
Const Fan_100_default = 65                                  ' 100% Lüftung ab 65 °C

' FAN: Temperaturgrenzen für Menüeinstellungen
Const Fan_min_temp = 0                                      ' niedrigste Temperatur für Lüftersteue-
rung in °C
```

```
Const Fan_max_temp = 99                                 ' höchste Temperatur für Lüftersteuerung
in °C


' ----- SOUND:  Piezo Soundausgabe -----
Const Sound_beep_time_const = 100                       ' Festlegung auf 100 ms
Const Sound_beep_time_short = 50                        ' Festlegung auf 50 ms
Const Sound_flag_default = 1                            ' Sound via Default eingeschaltet


' ----- KEY:  Tastatur -----
Const Key_zeilen_pullup = &B00000111                    ' PullUp Definition für PORTA
'                              |||
'                              ||+--- PullUp für PortA.0 Zeile 1
'                              |+---- PullUp für PortA.1 Zeile 2
'                              +----- PullUp für PortA.2 Zeile 3

Const Key_spalten_ddrf = &B00001110                     ' Definition für Data Direction
'                             |||
'                             ||+--- DDR für PortF.1 Spalte 1
'                             |+---- DDR für PortF.2 Spalte 2
'                             +----- DDR für PortF.3 Spalte 3

Const Key_interrupt_mask = &B0111111                    ' Interrupt Mask für Bit7
Const Key_waiting = 50                                  ' Konstante in ms für Entprellung

Const Key_shortcut_status_default = 0                   ' Konstante für Shortcut-Default


' ----- MENUE:  Menue -----
' Die Menüsteuerung erfolgt nur über die Tasten Hoch | Runter | Links | Rechts | Enter
' Für die Funktionen Heller | Dunkler und Auf | Zu werden ShortCut-Funktionen implementiert
Const Key_buttons = 5                                   'Anzahl der Tasten für das Menü

' Die logische Zustands-NULL für das Menue
Const Key_null = 0                                      'keine Taste gedrückt

' Wartezeit von Scr03 vor Einstieg in Hauptmenue
Const Menue_scr03_delay = 3                             ' Delay-Time 3 Sekunden


' ----- RTC:   DS1307 -----
' RTC:  Steuerkommandos für die exerne RTC DS1307
Const Rtc_address_write = &B11010000                    ' Adresse des DS1307 = &HD0 (schreiben)
Const Rtc_address_read = &B11010001                     ' Adresse des DS1307 = &HD1 (lesen)

' RTC:  Adressen für die exerne RTC DS1307
Const Rtc_address_seconds = &H00                        ' Adresse der Sekunden im RTC-Speicher
Const Rtc_address_minutes = &H01                        ' Adresse der Minuten im RTC-Speicher
Const Rtc_address_hours = &H02                          ' Adresse der Stunden im RTC-Speicher
Const Rtc_address_day = &H03                            ' Adresse der Wochentag im RTC-Speicher
Const Rtc_address_date = &H04                           ' Adresse der Tag im RTC-Speicher
Const Rtc_address_month = &H05                          ' Adresse der Monat im RTC-Speicher
Const Rtc_address_year = &H06                           ' Adresse der Jahr im RTC-Speicher
Const Rtc_address_control = &H07                        ' Adresse der Controllregister im RTC-
Speicher
' Anmerkung: Die Adressen können einzeln übergeben werden. Durch Block-Lesen
' können aber alle weiteren Daten hintereinander ausgelesen werden. Ein neues
' Adressieren ist nicht notwendig. Gestartet wird üblich mit Rtc_address_seconds


' ----- REL:  Relais -----
Const Passiv = 0                                        ' Definition Boolscher Zustand Passiv
Const Aktiv = 1                                         ' Definition Boolscher Zustand Aktiv

Const Rel_beschattung_min = 1                           ' Minimale Antriebs-Delay-Zeit in sec
Const Rel_beschattung_max = 200                         ' Maximale Antriebs-Delay-Zeit in sec

Const Rel_auf_dunkelheit_default = 5                    ' Grenzdunkelheit [0..15]
Const Rel_auf_dunkelheit_min = 1                        ' Einstellbare Helligkeit Minimum
Const Rel_auf_dunkelheit_max = 15                       ' Einstellbare Helligkeit Maximum

Const Rel_zu_temperatur_default = 20                    ' Grenztemperatur in °C
Const Rel_zu_temperatur_min = 1                         ' Einstellbare Temperatur Minimum °C
Const Rel_zu_temperatur_max = 99                        ' Einstellbare Temperatur Maximum °C

Const Rel_delay_default = 30                            ' Default-Wert für Antriebszeitüberwa-
chung in sec
' Anmerkung: Rel_beschattung_delay ist die Zeitverzögerung mit der der Antrieb
' automatisch wieder abgeschaltet wird. Der Antrieb selbst verfügt über einen
```

```
' "Endanschlag" der den Antrieb selbst abschaltet. Hier geht es nur noch darum
' nicht dauerhaft Spannung auf den Leitungen und Antrieben zu haben.


' ----- DIM:  Dimmer und Lichtfunktionen -----
' für Betrieb
Const Dim_helligkeit_default = 16                       ' Standardwert für Dimmerhelligkeit
[1..32]
Const Dim_helligkeitstabellenwerte = 32                 ' Anzahl an Dimmerstufen
Const Dim_helligkeit_min_default = 3                    ' kleinste Dimmerhelligkeit
Const Dim_helligkeit_max_default = 30                   ' hellste Dimmerhelligkeit
Const Dim_flash_grenzhelligkeit = 10                    ' Grenzhelligkeit für Flash-Mode [0..15]

' für Einstellungen
Const Dim_helligkeit_set_min = 1                        ' unterste einstellbare Grenze [1..32]
Const Dim_helligkeit_set_max = 32                       ' oberste einstellbare Grenze [1..32]


' ----- RAIN:  Regensensor -----
Const Rain_no = 0                                       ' Boolsche Variable für NO
Const Rain_yes = 1                                      ' Boolsche Variable für YES


' ----- WIND:  Windsensor -----
Const Wind_auswertung_default = 3                       ' Default Auswerteintervall in Sekunden
Const Wind_auswertung_min = 1                           ' Auswerteinterval untere Grenze in sec
Const Wind_auswertung_max = 30                          ' Auswerteinterval obere Grenze in sec
Const Wind_impuls_geschwindigkeit = 2.5                 ' Ein Impuls entspricht 2,5 m/sec Wind

Const Wind_beschattung_auf_default = 15                 ' Beschattung auf bei 15 km/h
Const Wind_beschattung_min = 1                          ' kleinste Windgeschwindigkeit in km/h
Const Wind_beschattung_max = 50                         ' größte Windgeschwindigkeit in km/h


' ----- LIGHT:  Lichtsensor -----
Const Light_auswertung_default = 10                     ' Default Auswerteintervall in Sekunden
Const Light_messwert_zahl = 5                           ' Anzahl wie of Light für Mittelwert
gemessen werden soll


' ----- SHT71:  Temperatur- und Luftfeuchtigkeit -----

' SHT71: Konstanten für Messungssteuerung
Const Sht71_auswertung_default = 15                     ' Default Auswerteintervall in Sekunden
Const Sht71_auswertung_min = 1                          ' Auswerteintervall untere Grenze
Const Sht71_auswertung_max = 30                         ' Auswerteinterval obere Grenze

' SHT71: Kommandocodes für SHT71-Befehle                  adr command r/w
Const Sht71_status_reg_write = &H06                     ' 000  0011    0
Const Sht71_status_reg_read = &H07                      ' 000  0011    1
Const Sht71_measure_temp = &H03                         ' 000  0001    1
Const Sht71_measure_humi = &H05                         ' 000  0010    1
Const Sht71_reset = &H1E                                ' 000  1111    0


' SHT71: Konstanten für die Berechnung von Luftfeuchtigkeit und Temperatur

' SHT71: Konstante für Berechung von RHlinear
Const Sht71_c1 = -4                                     ' Parameter C1 für 12 Bit
Const Sht71_c2 = 0.0405                                 ' Parameter C2 für 12 Bit
Const Sht71_c3 = -0.0000028                             ' Parameter C3 für 12 Bit
' Const Sht71_c1 = -4                                     ' Parameter C1 für 8 Bit
' Const Sht71_c2 = 0.648                                  ' Parameter C2 für 8 Bit
' Const Sht71_c3 = -0.00072                               ' Parameter C3 für 8 Bit

' SHT71: Konstanten für Temperaturberechnung
Const Sht71_d1 = -40.00                                 ' Parameter D1 für 5V und °C
'Const Sht71_d1 = -39.75                                 ' Parameter D1 für 4V und °C
'Const Sht71_d1 = -39.66                                 ' Parameter D1 für 3.5V und °C
'Const Sht71_d1 = -39.60                                 ' Parameter D1 für 3V und °C
'Const Sht71_d1 = -39.55                                 ' Parameter D1 für 2.5V und °C
Const Sht71_d2 = 0.01                                   ' Parameter D2 für 14 Bit und °C
'Const Sht71_d2 = 0.04                                   ' Parameter D2 für 12 Bit und °C

' SHT71: Konstanten für Berechung von RHtrue
Const Sht71_t1 = 0.01                                   ' Parameter T1 für 8 und 12 Bit
Const Sht71_t2 = 0.00008                                ' Parameter T2 für 12 Bit
'Const Sht71_t2 = 0.00008                                ' Parameter T2 für 8 Bit

' SHT71: Konstanten zur Berechnung des Taupunktes
```

```
Const Sht71_tn = 243.12                                 ' Wert lambda für Taupunktberechnung
Const Sht71_m = 17.62                                   ' Wert BETA für Taupunktberechnung
Const Sht71_alpha = 0.4343                              ' Parameter für Logaritmusberechnung

' SHT71: Sonstige Konstanten
Const Sht71_acknowledge = 1                             ' Positive Quittierung = Weiter
Const Sht71_noacknowledge = 0                           ' Negative Quittierung = Ende

Const Sht71_modus_humi = 1                              ' Parameter für measure Modus Luftfeuch-
tigkeit
Const Sht71_modus_temp = 2                              ' Parameter für measure Modus Temperatur

Const Sht71_communication_wait = 11                     ' 11 ms nach PowerUp/Reset warten

Const Sht71_response_wait = &H0004FFFF                  ' bei ATmega128 mit 16Mhz ca. 1,5 sec.

Const Crc_ok = 0                                        ' CRC-Ergebnisse OK
Const Crc_nok = 1                                       ' CRC-Ergebnis nicht OK

' SHT71: Konstanten für Beschattungssteuerung
Const Sht71_hysterese = 5                               ' Implementierung einer 5° Hysterese


' ----- LED:  LED-Array -----
Const Led_key_mask = &B1111111000000000                 ' Maske um alle Tasten-LED's abzuschal-
ten


' ----- EE: internes & externes EEPROM -----
Const Ee_auswertung_default = 600                       ' Default Auswerteintervall = 10 Minuten
Const Ee_auswertung_min = 60                            ' Auswerteinterval untere Grenze = 1 Min
Const Ee_auswertung_max = 1800                          ' Auswerteinterval obere Grenze = 30 Min

' Versionsverwaltung der Datenstrukturen
Const Akt_internal_ee_version = &H05                    ' Aktuelle EEPROM-Datenstruktur intern
Const Akt_external_ee_version = &H02                    ' Aktuelle EEPROM-Datenstruktur extern
' Hinweis: Über die Versionsverwaltung der Datenstrukturen wird vermieden, dass
' ablegete Daten Missinterpretiert werden. Auf diesem Weg können die Daten an
' jeder Stelle und zu jeder Zeit geändert werden. Bei jeder Änderung muss die
' Konstante erhöht werden. Bei Systemstart wird geprüft ob Initialisierung noch
' stimmt. Wenn neun werden alle Daten mit Default-Werten neu geladen.

' EE: Konstanten für Datenlogging (externes EEPROM)
Const Ee_data_version_extern_adress = &H0000            ' Byte-Variable: Datenversion
Const Ee_data_count_adress = &H0001                     ' Word-Variable: Anzahl Datenpakete
Const Ee_data_overrun_adress = &H0003                   ' Byte-Variable: Überlaufzähler
Const Ee_data_loging_start_adress = &H0008              ' Word-Variable: Startadresse für Ring-
buffer
Const Ea_data_loging_length = 8                         ' Anzahl der Datenbytes in einem Daten-
Record
Const Ea_data_loging_endadress = &H1FFF                 ' = 8191
'Const Ea_data_loging_endadress = &H000A                 ' = 10 for Test

Const Ee_data_count_default = &H0000                    ' Startwert für Ringbufferzähler
Const Ee_data_overrun_default = &H00                    ' Startwert für Überlaufzähler

' EE:  Adressen für das exerne EEPROM
Const Address_write = &B10100000                        ' Adresse des 24C512 = &HA0 (schreiben)
Const Address_read = &B10100001                         ' Adresse des 24C512 = &HA1 (lesen)


' ----- Timer: Timer -----

' Konstanten für Defaultwerte
Const Timer_open_flag_default = 0                       ' Funktion automatisches Öffnen ist
Default AUS
Const Timer_close_flag_default = 0                      ' Funktion automatisches Schließen ist
Default AUS


' ----- Allgemeine Konstanten -----
' Boolsche Variablen
Const True = 1                                          ' Boolsche Variable TRUE
Const False = 0                                         ' Boolsche Variable FALSE

' Variablen zur Klassifizierung ob Funktion über Menue oder Hot_Key ausgelöst
Const Menue_activated = &H01                            ' Funktion über Menü ausgelöst
Const Hotkey_activated = &H02                           ' Funktion über Hotkey ausgelöst
```

```
'-------------------------------------------------------------------------
' Definition von Variablen und Datentypen
'-------------------------------------------------------------------------

' ----- TRACE:  Trace-Jumper -----
Dim Trace_jumper_value As Byte                          ' globales Trace-Flag


' ----- LCD:  Für LCD-Display -----

' Variable für Helligkeits-PWM deklarieren und mit Defaultwert laden
Dim Lcd_pwmwert As Byte                                 ' Dimwert für LCD-Helligkeit
Dim Lcd_pwmwert_buffer As Byte                          ' Speicherung des eingestellten Hellig-
keitswerts

Dim Lcd_screensaver As Byte                             ' Flag für Screen-Saver-Modus
Dim Lcd_screensaver_counter As Word                     ' Sectic counter für Screensaver
Dim Lcd_screensaver_aktiv As Byte                       ' Flag ob Screensaver aktiv ist


' Variablen für Lcd_kontrastwerte deklarieren und mit Default laden
Dim Lcd_kontrastwert As Byte                            ' Kontrastwert für LCD-Display
Dim Lcd_helligkeit As Byte                              ' Helligkeitswert is Einerkomplement von
Lcd_pwmwert!


' ----- FAN:  Für Lüfter -----

' Variablen für Geschwindigkeits-PWM deklarieren und mit Defaultwert laden
Dim Fan_pwmwert As Byte                                 ' Geschwindigkeitswert für Lüfter
Dim Fan_sw_betrieb As Byte                              ' Flag ob Fan via SW gesteuert werden
soll
Dim Fan_25_temperatur As Byte                           ' Temperaturvariable für 25% Lüftung
Dim Fan_50_temperatur As Byte                           ' Temperaturvariable für 50% Lüftung
Dim Fan_75_temperatur As Byte                           ' Temperaturvariable für 75% Lüftung
Dim Fan_100_temperatur As Byte                          ' Temperaturvariable für 100% Lüftung


' ----- SOUND:  Piezo Soundausgabe -----
Dim Sound_beep_time As Word                             ' Dauer eines Beep
Dim Sound_flag As Byte                                  ' Flag ob Töne ausgegeben werden sollen


' ----- KEY:  Tastatur -----
Dim Key_keypressed As Byte                              ' Globale Variable zur Tastensignalisie-
rung

Dim Key_taste As Byte                                   ' Variable enthält Tastennummer
Dim Key_taste_buffer As Byte                            ' Buffer für verstätete Tastenauswertung
Dim Key_spalte As Byte                                  ' Variable für Spalten
Dim Key_spalten_index As Byte                           ' Variable für Spaltenindex Tastenre-
chung
Dim Key_zeile As Byte                                   ' Variable für Zeilen

' Hinweis:
' In verschiedenen Menüs ist es notwendig die Taste zu kennen welche betätigt
' wurde. Aus diesem grund wird unabhängig von der Menüsteuerung jeder Tasten-
' druck in der Buffer-Variable Key_taste_buffer gespeichert.


' ----- MENUE:  Menue -----
' Tastencodes
Dim Key_codes(key_buttons) As Byte                      ' Array für Tastencodes
Dim Key_shortcut_status As Byte                         ' Statusflag für Short-Cut Modus

' Textausgaben
Dim Menue_lcd_textbuffer(3) As String * 17              ' Auslesebuffer für die Menü-Strings
Dim Menue_ausgabeflags As Byte                          ' Ausgabeflags wie viele Zeilen

' Zustandsvariablen
Dim Menue_state As Word                                 ' aktueller State
Dim Menue_state_buffer As Word                          ' gebufferter Menüstatus
Dim Menue_state_renew As Byte                           ' Flag
Dim Menue_state_gosub As Word                           ' aktuelles Unterprogramm

Dim Menue_indimmenue_flag As Byte                       ' Flag für Dimmermenü wegen Shortcuts
Dim Menue_inrelmenue_flag As Byte                       ' Flag für Beschattungsmenue wegen
Shortcuts
Dim Menue_insensmenue_flag As Byte                      ' Flag für zyklisches Update von Menüs
```

```
' Arbeitsvariablen für universelles Zeit-Einstellmenue
Dim Menue_time_sekunde As Byte                          ' Sekunde
Dim Menue_time_minute As Byte                           ' Minute
Dim Menue_time_stunde As Byte                           ' Stunde

' Arbeitsvariablen für universelles Datum-Einstellmenue
Dim Menue_date_tag As Byte                              ' Tag
Dim Menue_date_monat As Byte                            ' Monat
Dim Menue_date_jahr As Byte                             ' Jahr

' Sonstige Arbeitsvariablen
Dim Menue_tagesanzahl As Byte                           ' Globale Variable für Bufferung von
Tagen im Monat
Dim Menue_datenuebernahme_flag As Byte                  ' Datenübernahmeflag für Time und Date
Dim Menue_data_changed_flag As Byte                     ' Flag für Datenübernahme in Settings


' ----- DCF77:  Funkuhr -----
Dim Dcf77_hour As Byte                                  ' Globale Stundenvariable
Dim Dcf77_min As Byte                                   ' Globale Minutenvariable
Dim Dcf77_sec As Byte                                   ' Globale Sekundenvariable

Dim Dcf77_weekday As Byte                               ' Globaler Wochentag

Dim Dcf77_day As Byte                                   ' Globale Tagvariable
Dim Dcf77_month As Byte                                 ' Globale Monatsvariable
Dim Dcf77_year As Byte                                  ' Globale Jahresvariable

Dim Sectic_count As Word                                ' Globaler SecTic (für Scheduler)
Dim Dcf_sync_count As Byte                              ' Zähler für Anzahl der Syncs


' ----- REL: Relais und Licht -----
Dim Beschattung_auf_flag As Byte                        ' Flag ob Beschattung auf aktiv
Dim Beschattung_zu_flag As Byte                         ' Flag ob Beschattung zu aktiv
Dim Beschattung_auf_counter As Word                     ' Sectic-Counter für Abschaltung auf
Dim Beschattung_zu_counter As Word                      ' Sectic-Counter für Abschaltung zu

Dim Rel_beschattung_delay As Byte                       ' Gloable Varaible für Beschattungsdelay

Dim Rel_auf_dunkel_flag As Byte                         ' Flag ob automatisches Auffahren bei
Dunkelheit
Dim Rel_auf_dunkelheit As Byte                          ' Variable für Dunkelheitswert

Dim Rel_zu_temp_flag As Byte                            ' Flag ob automatisches Zufahren bei
Temperatur
Dim Rel_zu_temperatur As Byte                           ' Variable für Grenztemperatur


' ----- DIM:  Dimmer -----
Dim Triac_zuenszeitpunkte(32) As Byte                   ' Array für Triac Zündzeitpunkte
Dim Dim_helligkeitsstufe As Byte                        ' Globale Variable für Dimmerhelligkeit
Dim Dim_index_counter As Byte                           ' Indexvariable für Auslesen der Zünd-
zeitpunkte
Dim Free_running_flag As Byte                           ' Flag ob Dimmer als single shot oder
free running
Dim Beleuchtung_flag As Byte                            ' Flag für Beleuchtungsrelais
Dim Dauerlicht_flag As Byte                             ' Flag für Dauerlicht-Relais
Dim Dim_flash_mode As Byte                              ' Flag für Flash-Mode
Dim Dim_flash_mode_locked As Byte                       ' Flag für Verriegelung des Flash-Mode
Dim Dim_helligkeit_min As Byte                          ' Aktuelle minimale Helligkeit
Dim Dim_helligkeit_max As Byte                          ' Aktuelle maximale Helligkeit
' Anmerkung: Der Flasmode beschreibt das automatische Anfahren des Dimmers bis
' zur aktuell eingestellten Dimmerstufe bei absinkender Raumhelligkeit


' ----- WIND:  Windsensor -----
Dim Wind_sensorimpulse As Byte                          ' Zähler für Windimpulse
Dim Wind_geschwindigkeit_kmh As Single                  ' Variable für Windgeschwindigkeit in
km/h
Dim Wind_kmh_max As Single                              ' Maximale gemessene Windgeschwindigkeit
Dim Wind_geschwindigkeit_ms As Single                   ' Variable für Windgeschwindigkeit in
m/s
Dim Wind_erfassung As Byte                              ' Globale Variable zum Status der Erfas-
sung
                                                        ' True = Erfassung Wind läuft, False =
Abgeschaltet
Dim Wind_auswertung As Byte                             ' Einstellung für Auswerteintervall
(Zeit)
Dim Wind_erfassungs_counter As Word                     ' Globaler Zähler für Windauswertung
```

```
Dim Wind_beschattung_auf As Byte                        ' Globale Varible für kritische Windge-
schwindigkeit

Dim Wind_autoauffahren As Byte                          ' Globales Flag für automatisches Auf-
fahren


' ----- SHT71:  Temperatur- und Luftfeuchtigkeit -----
Dim Sht71_erfassung As Byte                             ' Globale Variable zum Status der Erfas-
sung
                                                        ' True = Erfassung Wind läuft, False =
Abgeschaltet
Dim Sht71_auswertung As Byte                            ' Einstellung für Auswerteintervall
(Zeit)
Dim Sht71_erfassungs_counter As Word                    ' Globaler Zähler für Temp&Hum

' globale Hilfsvariablen
Dim Sht71_transceive_struct(3) As Byte                  ' Sende-Datenstruktur für SHT71
Dim Sht71_temp_byte1 As Byte                            ' Temporäre Arbeitsvariable Byte für
SHT71
Dim Sht71_humi_word As Word                             ' Rohdatenwert (16-Bit) für Luftfeuch-
tigkeit
Dim Sht71_temp_word As Word                             ' Rohdatenwert (16-Bit) für Temperatur
Dim Sht71_checksum As Byte                              ' Arbeitsvariable für SHT71 Checksummen-
rechnung
Dim Sht71_error_result As Byte                          ' Arbeitsvariable für SHT71 Fehlerbe-
handlung

' globale Variable für Temperatur, Luftfeuchtigkeit und Taupunkt
Dim Sht71_rhlinear As Single                            ' Luftfeuchtigkeit linear
Dim Sht71_rhtrue As Single                              ' Luftfeuchtigkeit mit Korrektur (real)
Dim Sht71_rhtrue_min As Single                          ' Minimaler gemessener Feuchtigkeitswert
Dim Sht71_rhtrue_max As Single                          ' Maximaler gemessener Feuchtigkeitswert
Dim Sht71_dewpoint As Single                            ' Taupunkt
Dim Sht71_t As Single                                   ' Temperatur
Dim Sht71_t_min As Single                               ' Minimale gemessene Temperatur
Dim Sht71_t_max As Single                               ' Maximale gemessene Temperatur

Dim Sht71_autozufahren As Byte                          ' Globales Flag für automatisches Zufah-
ren
Dim Sht71_autozufahren_locked As Byte                   ' Verriegelung gegen erneutes Zufahren


' ----- LIGHT:  Lichtsensor -----
Dim Light_channel As Byte                               ' Variable für ADC-Kanal festlegen
Dim Light_erfassung As Byte                             ' Globale Variable zum Status der Erfas-
sung
                                                        ' True = Erfassung Licht läuft, False =
Abgeschaltet
Dim Light_erfassungs_counter As Word                    ' Globaler Zähler für Lichtwerterfassung
Dim Light_value As Byte                                 ' Globaler Lichtmesswert
Dim Light_update_flag As Byte                           ' Flag für Werteupdate (Wird bei Flash
benötigt)

Dim Light_autoauffahren As Byte                         ' Globales Flag für automatisches Auf-
fahren
Dim Light_autoauffahren_locked As Byte                  ' Verriegelung gegen erneutes Auffahren
bei Dunkelheit


' ----- RAIN:  Regensensor -----
Dim Rain_autoauffahren As Byte                          ' Globales Flag für automatisches Auf-
fahren


' ----- LED:  LED-Array -----
Dim Led_array As Word                                   ' Globale Variable für LED's
Dim Led_array_save As Word                              ' Buffer um Änderungen zu tracken
' Der Zugriff auf das Array erfolgt Bitweise. Übertragen wird das gesamte Word
' in das Latch!
Dim Led_key_flag As Byte                                ' Flag zur Steuerung der LED-Überwachung
Dim Led_key_counter As Word                             ' Merker für aktuellen SecTic


' ----- EE: internes & externes EEPROM -----

' generelle Einstellungen für Datenlogging und Auswertungsintervall
Dim Ee_datalogging_flag As Byte                         ' Globales Flag ob Daten gelogged werden
sollen
```

```
Dim Ee_auswertung As Word                                  ' Zeitintervall für Loggingdaten

Dim Ee_erfassungs_counter As Word                          ' Globaler Zähler für Logging-Daten
schreiben

' EE: EEPROM Datenstruktur internes EEPROM - Setting-Daten
Dim Ee_data_version_intern As Eram Byte At &H0000          ' Speicherstelle für Datenversion
Dim Ee_data_debug As Eram Byte At &H0001                   ' Byte-Variable: Test_debug_flag
Dim Ee_data_lcd_helligkeit As Eram Byte At &H0002          ' Byte-Variable: Lcd_helligkeit
Dim Ee_data_lcd_kontrast As Eram Byte At &H0003            ' Byte-Variable: Lcd_kontrastwert
Dim Ee_data_lcd_saver As Eram Byte At &H0004               ' Byte-Variable: Lcd_screensaver
Dim Ee_data_wind_intervall As Eram Byte At &H0005          ' Byte-Variable: Wind_auswertung
Dim Ee_data_wind_speed As Eram Byte At &H0006              ' Byte-Variable: Wind_beschattung_auf
Dim Ee_data_temp_intervall As Eram Byte At &H0007          ' Byte-Variable: Sht71_auswertung
Dim Ee_data_dimmer As Eram Byte At &H0008                  ' Byte-Variable: Dim_flash_mode
Dim Ee_data_dim_min As Eram Byte At &H0009                 ' Byte-Variable: Dim_helligkeit_min
Dim Ee_data_dim_max As Eram Byte At &H000A                 ' Byte-Variable: Dim_helligkeit_max
Dim Ee_data_sound As Eram Byte At &H000B                   ' Byte-Variable: Sound_flag
Dim Ee_data_logging_flag As Eram Byte At &H000C            ' Byte-Variable: Ee_datalogging_flag
Dim Ee_data_logging_intervall As Eram Word At &H000D       ' Byte-Variable: Ee_auswertung
Dim Ee_data_conv_timeout As Eram Byte At &H000F            ' Byte-Variable: Rel_beschattung_delay
Dim Ee_data_conv_temp_flag As Eram Byte At &H0010          ' Byte-Variable: Rel_zu_temp_flag
Dim Ee_data_conv_temperatur As Eram Byte At &H0011         ' Byte-Variable: Rel_zu_temperatur
Dim Ee_data_conv_light_flag As Eram Byte At &H0012         ' Byte-Variable: Rel_auf_dunkel_flag
Dim Ee_data_conv_lichtwert As Eram Byte At &H0013          ' Byte-Variable: Rel_auf_dunkelheit
Dim Ee_data_conv_hotkey As Eram Byte At &H0014             ' Byte-Variable: Key_shortcut_status
Dim Ee_data_fan As Eram Byte At &H0015                     ' Byte-Variable: Fan_sw_betrieb
Dim Ee_data_fan_25_temp As Eram Byte At &H0016             ' Byte-Variable: Fan_25_temperatur
Dim Ee_data_fan_50_temp As Eram Byte At &H0017             ' Byte-Variable: Fan_50_temperatur
Dim Ee_data_fan_75_temp As Eram Byte At &H0018             ' Byte-Variable: Fan_75_temperatur
Dim Ee_data_fan_100_temp As Eram Byte At &H0019            ' Byte-Variable: Fan_100_temperatur
Dim Ee_data_dimmer_aktuell As Eram Byte At &H0020          ' Byte-Variable: Dim_helligkeitsstufe
Dim Ee_data_sht71_t_min As Eram Single At &H0021           ' Single-Variable: Sht71_t_min
Dim Ee_data_sht71_t_max As Eram Single At &H0025           ' Single-Variable: Sht71_t_max
Dim Ee_data_sht71_rhtrue_min As Eram Single At &H0029      ' Single-Variable: Sht71_rhtrue_min
Dim Ee_data_sht71_rhtrue_max As Eram Single At &H002D      ' Single-Variable: Sht71_rhtrue_max
Dim Ee_data_wind_kmh_max As Eram Single At &H0031          ' Single-Variable: Wind_kmh_max

' EE: EEPROM Datenstruktur externes EEPROM - Datenlogging
Dim Ee_data_version_extern As Byte                         ' Datenversion für externe Daten
Dim Ee_data_count As Word                                  ' Zähler abgelegter Daten
Dim Ee_data_overrun As Byte                                ' Überlaufzähler

' Aufbau Loggingdaten-Record
' Byte(1) Ee_data_hour
' Byte(2) Ee_data_minute
' Byte(3) Ee_data_day
' Byte(4) Ee_data_month [Bit7..4]
' Byte(4) Ea_data_light [Bit3..0]
' Byte(5) Ee_data_year
' Byte(6) Ee_data_rain  [Bit7]
' Byte(6) Ee_data_wind  [Bit6..0]
' Byte(7) Ee_data_temp
' Byte(8) Ee_data_humidity
Dim Ee_data_logging_struct(8) As Byte                      ' Datenstruktur

Dim Ee_output_flag As Byte                                 ' Ausgabeflag der Loggingdaten
Dim Ee_output_counter As Word                              ' Ausgabezähler
Dim Ee_output_endadress As Word                            ' Endadresse für Ausgabe

' Anmerkung: Damit während der Ausgabe der Daten die normale Bearbeitung des
' Programms möglich ist, werden die einzelnen Datenpakete über den Sequencer
' ausgegeben.


' ----- Timer: Timer -----

' generelle Einstellungen für automatischen Öffnen und Schließen -----
Dim Timer_open_flag As Byte                                ' Flag für Funktion automatisches Öffnen
Dim Timer_close_flag As Byte                               ' Flag für Funktion automatisches
Schließen

' Variablen für Zeit und Datumsinformationen
Dim Timer_open_sekunde As Byte                             ' Sekundenvariable für Öffnungszeit
Dim Timer_open_minute As Byte                              ' Minutenvariable für Öffnungszeit
Dim Timer_open_stunde As Byte                              ' Stundenvariable für Öffnungszeit

Dim Timer_open_tag As Byte                                 ' Tagvariable  für Öffnungszeit
Dim Timer_open_monat As Byte                               ' Monatsvariable für Öffnungszeit
Dim Timer_open_jahr As Byte                                ' Jahresvariable für Öffnungszeit
```

```
Dim Timer_close_sekunde As Byte                             ' Sekundenvraiable für Schließungszeit
Dim Timer_close_minute As Byte                              ' Minutenvariable für Schließungszeit
Dim Timer_close_stunde As Byte                              ' Stundenvariable für Schließungszeit

Dim Timer_close_tag As Byte                                 ' Tagesvariable für Schließungszeit
Dim Timer_close_monat As Byte                               ' Monatsvariable für Schließungszeit
Dim Timer_close_jahr As Byte                                ' Jahresvariable für Schließungszeit


' ----- Allgemeine systembedingte globale Variablen -----


' temporäre globale Arbeitsvariablen
Dim Temp_byte_1 As Byte                                     ' Temporäre Byte-Variable 1
Dim Temp_byte_2 As Byte                                     ' Temporäre Byte-Variable 2
Dim Temp_byte_3 As Byte                                     ' Temporäre Byte-Variable 3

Dim Temp_word_1 As Word                                     ' Temporäre Word-Variable 1
Dim Temp_word_2 As Word                                     ' Temporäre Word-Variable 2

Dim Temp_integer_1 As Integer                               ' Temporäre Integer Variable 1

Dim Temp_string_1 As String * 20                            ' Temporäre String Variable 1
Dim Temp_string_2 As String * 20                            ' Temporäre String Variable 2

Dim Temp_single_1 As Single                                 ' Temporäre Single Variable 1


' Globale Variablen für Testzwecke bei Traceausgaben
Dim Test_int_counter As Word                                ' Testcounter für Tasteninterrupts
Dim Test_debug_flag As Byte


'-------------------------------------------------------------------------------
' Prototyping
'-------------------------------------------------------------------------------

' ========== Subroutinen ==========

' ----- LCD:  Für LCD-Display -----
Declare Sub Lcd_set_helligkeit(byval Value As Byte)

Declare Sub Lcd_writesingleline(byval Line1 As Byte , Text1 As String)
Declare Sub Lcd_writedoubleline(byval Line1 As Byte , Byval Line2 As Byte , Text1 As String , Text2
As String)
Declare Sub Lcd_writetrippleline(text1 As String , Text2 As String , Text3 As String)


' ----- FAN:  Für Lüfter -----
Declare Sub Fan_set_geschwindigkeit(byval Value As Byte)


' ----- SOUND: Piezo Soundausgabe -----
Declare Sub Sound_beep(byval Time_value As Word)


' ----- SHT71: Temperatur- und Luftfeuchtigkeit -----
Declare Sub Sht71_calc_measurements()


' ----- EEPROM (intern und extern) -----
Declare Sub Ee_write_eeprom_byte(byval Adress As Word , Byval Value As Byte)
Declare Sub Ee_write_eeprom_word(byval Adress As Word , Byval Value As Word)

Declare Sub Ee_write_eeprom_struct(byval Adress As Word)
Declare Sub Ee_read_eeprom_struct(byval Adress As Word)


' ========== Funktionen ==========

' ----- KEY:  Tastatur -----
Declare Function Key_query_matrix() As Byte


' ----- RAIN:  Regensensor -----
Declare Function Rain_measurement() As Byte


' ----- SHT71: Temperatur- und Luftfeuchtigkeit -----
```

```
Declare Function Sht71_write_byte(byval Value As Byte) As Byte
Declare Function Sht71_read_byte(byval Acknowledge As Byte) As Byte
Declare Function Sht71_softreset() As Byte
Declare Function Sht71_read_statusregister(byref P_value As Byte , Byref P_checksum As Byte) As Byte
Declare Function Sht71_write_statusregister(byref P_value As Byte) As Byte
Declare Function Sht71_measure(byref P_value As Word , Byref P_checksum As Byte , Byval Modus As
Byte) As Byte
Declare Function Sht71_crc_check(byval Value As Word , Byval Crc As Byte , Byval Modus As Byte) As
Byte


' ----- LIGHT:  Lichtsensor -----
Declare Function Light_measurement(byval Value As Byte) As Byte


' ----- EEPROM (intern und extern) -----
Declare Function Ee_read_eeprom_byte(byval Adress As Word) As Byte
Declare Function Ee_read_eeprom_word(byval Adress As Word) As Word


' ----- Allgemeine Funktionen -----
Declare Function Byte2string(byval Value As Byte , Byval Character As Byte) As String
Declare Function Time2string(byval Value1 As Byte , Byval Value2 As Byte , Byval Value3 As Byte ,
Byval Character As Byte) As String
Declare Function Schaltjahr(byval Value As Word) As Byte


'-----------------------------------------------------------------------------
' Konfiguration und Basiseinstellungen (Projekt und Testumgebung)
'-----------------------------------------------------------------------------

' ----- vorgezogene wichtige Gosub's (der Rest erfolgt hinten) -----

' SOUND: Routine und Einstellungen
   Gosub Sound_konfig


' ----- CONFIG -----

' Device COM2 mounten und im System zur Verfügung stellen
   Config Com2 = 38400 , Synchrone = 0 , Parity = None , Stopbits = 1 , Databits = 8 , Clockpol = 0
   Open "com2:" For Binary As #2

   ' Systemstart auf jeden Fall auf RS232 ausgeben
   Print #2 , "** Iginiton sequence start **"


' LCD-Display konfigurieren

   ' Port G.1 als Ausgang definieren und Display einschalten
   Config Lcd_on_pin = Output
   Lcd_on_dataout = 1

   ' Konfiguration LCD Display
   Config Lcdpin = Pin , Db4 = Portc.0 , Db5 = Portc.1 , Db6 = Portc.2 , Db7 = Portc.3 , E = Portc.4
, Rs = Portc.5
   Config Lcd = 16 * 3 , Chipset = Dogm163v5                ' DOG-M Treiber laden
   Config Lcdbus = 4                                        ' LCD arbeitet über 4-Bit
   Initlcd                                                  ' LCD initialisieren
   Waitms 100                                               ' 100ms nach Init warten
   Cursor Off Noblink                                       ' Blinkenden Cursor abschalten
   Cls                                                      ' Clear Screen

   ' Definition benutzerdefinierter Zeichen
   ' Benutzerdefiniertes LCD-Zeichen Pfeil Hoch
   Deflcdchar 1 , 4 , 14 , 31 , 21 , 4 , 4 , 4 , 32
   ' Benutzerdefiniertes LCD-Zeichen Pfeil Runter
   Deflcdchar 2 , 4 , 4 , 4 , 21 , 31 , 14 , 4 , 32
   ' Benutzerdefiniertes LCD-Zeichen Pfeil Links
   Deflcdchar 3 , 32 , 6 , 12 , 31 , 12 , 6 , 32 , 32
   ' Benutzerdefiniertes LCD-Zeichen Pfeil Rechts
   Deflcdchar 4 , 32 , 12 , 6 , 31 , 6 , 12 , 32 , 32
   ' Benutzerdefiniertes LCD-Zeichen Enter
   Deflcdchar 5 , 32 , 14 , 17 , 21 , 17 , 14 , 32 , 32

   ' Erstes Lebenszeichen auf Display schreiben
   Locate 1 , 7 : Lcd "***"
   Locate 2 , 5 : Lcd "ConvCtrl"
   Locate 3 , 7 : Lcd "***"
   Wait 2                                                   ' 2 Sec warten
```

```
   Cls                                                    ' Clear Screen
   Locate 1 , 1 : Lcd "Initialisierung!"
   Locate 2 , 7 : Lcd "****"
   Locate 3 , 3 : Lcd "Bitte warten"

' Konfiguration Timer 3 für Hardware-PWM an OC3A (E.3) und OC3B (E.4)
   ' Konfiguration muss global geschehen da mit einem Timer zwei PWM realisiert
   Config Timer3 = Pwm , Pwm = 8 , Prescale = 256 , Compare A Pwm = Clear Up , Compare B Pwm = Clear
Up

'Konfiguration Timer2 für Zündung TRIAC nach Vorgabe bzw. Nulldurchgang
   Config Timer2 = Timer , Prescale = 1024
   On Timer2 Dimmer_triac_irq                             ' ISR festlegen
   Enable Timer2

' DCF77: Routinen und Einstellungen

   '  CONFIG DCF77 = pin , timer = timer [ INVERTED=inv, CHECK=check, UPDATE=upd, UPDATETIME=updtime
, TIMER1SEC=tmr1sec, SWITCHPOWER=swpwr, POWERPIN=pin, POWERLEVEL = pwrlvl , SECONDTICKS=sectick
,DEBUG=dbg , GOSUB = Sectic ]
   '  PIN          = input pin that is connected to the DCF-77 signal
   '  TIMER        = The timer that is used to generate the compare interrupts
   '  INVERTED     = This value is 0 by default. When you specify 1, the compiler will assume you
use an inverted DCF signal.
   '  CHECK        = The DCF-77 parity bits are checked
   '  UPDATE       = Upd determines how often the internal date/time variables are updated with the
DCF received values.
   '  UPDATETIME   = This value depends on the used UPDATE parameter.
   '  TIMER1SEC    = 16 bit timers with the right crystal value can generate a precise interrupt
that fires every second.
   '                 Wichtig: Wird der Parameter TIMER1SEC = 1 gesetzt, wird während der synchroni-
sationsfreien Zeit der Compare-Wert so verändert, dass nur mehr 1 Interrupt pro Sekunde ausgelöst
wird.
   '  SWITCHPOWER  = This option can be used to turn on/off the DCF-77 module with the control of a
port pin.
   '  POWERPIN     = The name of a pin like pinB.2 that will be used to turn on/off the DCF module.
   '  POWERLEVEL   = This option controls the level of the output pin that will result in a power ON
for the module.
   '  SECONDTICKS  = The number of times that the DCF signal state is read.
   '  DEBUG        = Optional value to fill 2 variables with debug info.
   '  GOSUB        = The Sectic option will call a label in the main program every second.

   ' Config mit stündlichem Update
   ' Config Dcf77 = Pinb.2 , Timer = 1 , Inverted = 0 , Check = 1 , Update = 1 , Updatetime = 59 ,
Timer1sec = 1 , Debug = 1 , Gosub = Sectic

   ' Config mit kontinuierlichem Update
   ' Config Dcf77 = Pinb.2 , Timer = 1 , Inverted = 0 , Check = 1 , Update = 0 , Timer1sec = 1 ,
Debug = 1 , Gosub = Sectic

   Config Dcf77 = Pinb.2 , Timer = 1 , Inverted = 0 , Check = 1 , Update = 2 , Updatetime = 12 ,
Timer1sec = 1 , Debug = 1 , Gosub = Sectic
   '              |        |           |               |          |            |                 |
|            |
   ' Use PinB.2 ---+        |           |               |          |            |                 |
|            |
   ' Timer 1 is default ----+           |               |          |            |                 |
|            |
   ' DCF77 signal wird nicht invertiert +               |          |            |                 |
|            |
   ' Vergleich akt. Minutes mit vorheriger Minute ------+          |            |                 |
|            |
   ' Perform Update einmal am Tag ---------------------------------+            |                 |
|            |
   ' Zielzeit für Zeitresynchronistion ist 12:00 Uhr ----------------------------+                 |
|            |
   ' Updatetime ist nicht relevant da Update = 0 ------------------+                               |
|            |
   ' Timer1sec = 1 verändert in sync. freier Zeit den Compare Wert auf nur -----------------------+
|            |
   '              Interrupt pro Sekunde = Reduktion der Prozessor-Last!!!
|            |
   ' Switchpower, Powerpin und Powerlevel wird nicht benutzt da Interface nicht vorhanden!
|            |
   ' Secondticks wird dem Compiler überlassen und nicht neu gesetzt!
|            |
   ' Debug für die Variablen  bDCF_Pause und bDCF_Impuls ist aktiv ---------------------------------
--------------+            |
   ' Jede Sekunde wird Gosub-Routine Dcf77_sectic angesprungen -------------------------------------
--------------------------+
```

```
   ' Datumsformat einstellen
   Config Date = Dmy , Separator = .                        ' German  dd.mm.yy


' LIGHT: Lichtsensor
   ' Konfiguration ADC Single-Mode und automatische Prescaler Setting
   ' Der Single-Mode wir bei BASCOM in Verbindung mit der Funktion GETADC() verwendet
   ' Der Prescaler teilt den internen Tackt durch 2, 4, 8,16,32,64 or 128 da der ADC
   ' einen Takt zwischen 50-200 kHz benötigt.
   ' Das AUTO Feature von BASCOM, setzt automatisch die höchste mögliche Taktrate
   Config Adc = Single , Prescaler = Auto , Reference = Avcc


' ----- Port's und Pin's -----

' Konfiguration des Trace-Jumpers
   Config Trace_pin = Input                                 ' Trace-Jumper soll gelesen werden
   Trace = 0                                                ' Widerstand abschalten


' Konfiguration der Tastenmatrix
   ' Konfiguration Port A Zeilen (Reihen)
   Config Key_zeile_1_pin = Input                           ' Zeile 1 ist Eingang
   Config Key_zeile_2_pin = Input                           ' Zeile 2 ist Eingang
   Config Key_zeile_3_pin = Input                           ' Zeile 3 ist Eingang
   Porta = Key_zeilen_pullup                                ' PullUp aktiv für PA0-PA2

   ' Konfiguration Port F Spalten
   Config Key_spalte_1_pin = Output                         ' Spalte 1 ist Ausgang
   Config Key_spalte_2_pin = Output                         ' Spalte 2 ist Ausgang
   Config Key_spalte_3_pin = Output                         ' Spalte 3 ist Ausgang

   ' Basis-Initialisierung der Spalten-PINs (alle Spalten auf LOW)
   ' Dies ist nach Systemstart normalerweise Default, aber man weis ja nie :-)
   Reset Key_spalte_1_port                                  ' Spalte 1 auf LOW
   Reset Key_spalte_2_port                                  ' Spalte 2 auf LOW
   Reset Key_spalte_3_port                                  ' Spalte 3 auf LOW

   ' Alternative Konfiguration für die Spalten
   ' Ddrf = Key_spalten_ddrf                                  ' Alle Spalten auf einen Schlag
   ' Nachteil dieser Lösung ist, dass gesamtes Register geladen wird. Im Gesamtprojekt
   ' haben aber die anderen BIT's andere Aufgaben und Bedeutung. Daher explizites Setzen
   ' gewählt.


' Konfiguration I2C-Bus
   Config Sda = Sda_port                                    ' TWI SDA = Data
   Config Scl = Scl_port                                    ' TWI SCL = Clock

   I2cinit                                                  ' we need to set the pins in the proper
state

   ' Achtung: gemäß Datenblatt kann der DS1307 nur eine max frequency von 100 kHz
   Config Twi = 100000                                      ' Taktfrequenz 100 kBaud

   ' TWI gleich einschalten, das macht BASCOM ansonsten erst beim I2CStart
   Twcr = &B00000100                                        ' nur TWEN setzen

   ' Optionale Einstellungen für TWI Kommunikation die nicht zwingend erforderliche sind
   ' Twsr = 0                                                 ' Status und Prescaler Register
   ' Twbr = 152                                               ' Bit Rate Register, 72 = 100kHz   152
= 50kHz(0 bis 255)
   ' Config I2cdelay = 40


' Stand-Alone-LED's
   Config Alive_led_pin = Output                            ' Ausgang definieren
   Alive_led = 0                                            ' LED ausschalten

   Config Sectick_led_pin = Output                          ' Ausgang definieren
   Sectick_led = 0                                          ' LED ausschalten


' RAIN: Regensensor konfigurieren
   Config Rain_sensor = Input                               ' Eingang definieren


' SHT71 Sensor
   Config Temp_sck = Output                                 ' Clock für SHT71
```

```
   Config Temp_data = Output                                ' Data für SHT71
   ' Anmerkung: Der SHT71 besitz eine TWI Interface welches sich aber nicht nach
   ' Spezifikation verhält. Aus diesem Grund kann nicht das HW TWI des Mega
   ' verwenden werden sondern die Ansteuerung muss von Hand implementiert werden.


' LED: LED-Array
   ' Konfiguration der Ausgänge für Datenbits, Adressbits und Latch Clear
   Config Led_bit0_pin = Output                             ' Daten-Bit 0 ist Ausgang
   Config Led_bit1_pin = Output                             ' Daten-Bit 1 ist Ausgang
   Config Led_bit2_pin = Output                             ' Daten-Bit 2 ist Ausgang
   Config Led_bit3_pin = Output                             ' Daten-Bit 3 ist Ausgang

   Config Led_adr0_pin = Output                             ' Adress-Bit 0 ist Ausgang
   Config Led_adr1_pin = Output                             ' Adress-Bit 1 ist Ausgang

   Config Led_clr_pin = Output                              ' Latch Clear ist Ausgang

   ' Basis-Initialisierung aller Signalleitungen auf LOW
   ' Dies ist nach Systemstart normalerweise Default, aber man weiss ja nie :-)
   Reset Led_bit0_port                                      ' Datenbit 0 auf Low
   Reset Led_bit1_port                                      ' Datenbit 1 auf Low
   Reset Led_bit2_port                                      ' Datenbit 2 auf Low
   Reset Led_bit3_port                                      ' Datenbit 3 auf Low

   Reset Led_adr0_port                                      ' Adressbit 0 auf Low
   Reset Led_adr1_port                                      ' Adressbit 1 auf Low

   Reset Led_clr_port                                       ' Latch-Clear auf Low

   ' Funktionstabelle des Latch:
   ' +-------------------------------------+
   ' | Inputs:     ENABLE          | OUTPUT |
   ' |    _____    __  __          |        |
   ' |    CLEAR    C1  C2    DATA  |  Q     |
   ' |=====================================|
   ' |      H      L   L      L    |  L     |
   ' |      H      L   L      H    |  H     |
   ' |      H      x   H      x    |  Qo    |
   ' |      H      H   x      x    |  Qo    |
   ' |      L      x   x      x    |  L     |
   ' +-------------------------------------+


' ----- Variablen und Werte -----

' Hinweis:
' Alle Variablen werden in diesem Abschnitt zunächste mit Default-Werten geladen.
' Zu einem Späteren Zeitpuntk wird geprüft, ob die im Flash abgelegten Daten-
' sturkturen passen, neu initialisiert werden müssen oder ob die hier vorgenom-
' menen Default-Einstellungen durch User-Settings aus dem EEPROM überschrieben
' werden können.


' TRACE:  Trace Jumper
   Trace_jumper_value = Trace_pin                           ' Bitwert von Port/Pin lesen
   Trace_jumper_value = Trace_jumper_value Xor &B00000001   ' Bit invertieren


' LCD:  LCD-Display
   Lcd_pwmwert = Lcd_default_helligkeit                     ' Dimmwert mit Default laden
   Lcd_helligkeit = 255 - Lcd_pwmwert                       ' Inhalte umdrehen
   Lcd_kontrastwert = Lcd_kontrast_default                  ' Kontrastwert
   Lcd_screensaver = Lcd_screensaver_default                ' Screensaver auf AUS setzen

   Lcd_screensaver_aktiv = False                            ' Screensaver ist nicht aktiv


' FAN:  Lüfter
   Fan_pwmwert = Fan_default_geschwindigkeit                ' PWM-Wert für Lüfter laden
   Fan_sw_betrieb = True                                    ' SW Steuerung eingeschaltet

   Fan_25_temperatur = Fan_25_default                       ' Variable für 25% mit default laden
   Fan_50_temperatur = Fan_50_default                       ' Variable für 50% mit default laden
   Fan_75_temperatur = Fan_75_default                       ' Variable für 75% mit default laden
   Fan_100_temperatur = Fan_100_default                     ' Variable für 100% mit default laden


' SOUND:  Piezo-Sound
   Sound_beep_time = Sound_beep_time_const                  ' Beep Zeitkonstante laden
```

```
   Sound_flag = Sound_flag_default                          ' Sound-Flag mit Default-Variable vorbe-
legen


' MENUE: Strukturen und Variablen
   ' Tastencodes
   Key_codes(1) = 12                                        ' Taste Hoch
   Key_codes(2) = 42                                        ' Taste Runter
   Key_codes(3) = 41                                        ' Taste Links
   Key_codes(4) = 44                                        ' Taste Rechts
   Key_codes(5) = 22                                        ' Taste Enter

   Key_keypressed = False                                   ' Vorinitialisierung

   'Initial state variables
   Menue_state = Loadlabel(men1)                            ' Startbildschirm
   Menue_state_renew = 1                                    ' Menüstatus wurde geändert
   Key_taste = Key_null                                     ' keine Taste gedrückt
   Key_shortcut_status = Key_shortcut_status_default        ' Shortcutstatus auf Default setzen

   ' Erkennung für ShortCut ob Display auf Dimmer-Menues steht damit Werte
   ' korrigiert / ergänzt werden können
   ' +-----------------------------------------------+
   ' | Wert |  Bedeutung                             |
   ' |======|========================================|
   ' |  0   |  Menü nicht in relevanten Dimmer-Menüs |
   ' |  1   |  Menü steht auf Dimmer heller          |
   ' |  2   |  Menü steht auf Dimmer dunkler         |
   ' |  3   |  Menü steht auf Dimmer aus             |
   ' |  4   |  Menü steht auf Dimmerstufenanzeige    |
   ' +-----------------------------------------------+
   Menue_indimmenue_flag = 0                                ' Flag für Dimmermenü wegen Shortcuts

   ' Erkennung für ShortCut ob Display auf Beschattungs-Menues steht damit Werte
   ' korrigiert / ergänzt werden können
   ' +-----------------------------------------------+
   ' | Wert |  Bedeutung                             |
   ' |======|========================================|
   ' |  0   |  Menü nicht in relevanten Relais-Menüs |
   ' |  1   |  Menü steht auf Beschattung auf        |
   ' |  2   |  Menü steht auf Beschattung zu         |
   ' |  3   |  Menü steht auf Beschattung stopp      |
   ' +-----------------------------------------------+
   Menue_inrelmenue_flag = 0                                ' Flag für Beschattungs-Menue wegen
Shortcuts

   ' +-----------------------------------------------+
   ' | Wert |  Bedeutung                             |
   ' |======|========================================|
   ' |  0   |  Kein zyklische Update von Menüdaten   |
   ' |  1   |  Update von Windgeschwindigkeit        |
   ' |  2   |  Update zustand Regensensor            |
   ' |  3   |  Update Innentemperatur                |
   ' |  4   |  Update Luftfeuchtigkeit               |
   ' |  5   |  Update Lichintensität                 |
   ' |  6   |  Update Uhrzeit von DCF77 Empfänger    |
   ' |  7   |  Update DCF77 Statusbits und Infos     |
   ' +-----------------------------------------------+
   Menue_insensmenue_flag = 0                               ' Flag für zyklisches Update Menüdaten

   Menue_datenuebernahme_flag = 0                           ' Datenübernahmeflag initialisieren
   Menue_data_changed_flag = 0                              ' Flag ob sich Daten wirklich geändert
haben
   ' Anmerkung: Mittels Menue_data_change_flag soll unnötiges Schreiben in das
   '             EEPROM verhindert werden.


' DCF77: Voreinstelung Sectic_counter und Sync_counter
   Sectic_count = 0                                         ' Counter mit NULL vorinitialisieren
   Dcf_sync_count = 0                                       ' Synchronisationszähler auf NULL


' DIM: Voreinstellung für Dimmer. Bei Free_running_flag = passiv ist Dimmer aus!
   Free_running_flag = Passiv                               ' Dimmer auf passiv schalten
   Beleuchtung_flag = Passiv                                ' Statusflag für Beleuchtung Relais
passiv
   Dauerlicht_flag = Passiv                                 ' Statusflag für Dauerlich-Relais passiv
   Dim_flash_mode = Passiv                                  ' Flash-Mode erstmal passiv
   Dim_flash_mode_locked = False                            ' Flas-Mode-Verriegelung false
```

```
   Dim_helligkeit_min = Dim_helligkeit_min_default          ' Dimmerhelligkeit MIN erstmal mit De-
fault laden
   Dim_helligkeit_max = Dim_helligkeit_max_default          ' Dimmerhelligkeit MAX erstmal mit De-
fault laden


' LIGHT: Lichtsensor
   Light_channel = 0                                       ' TSL250R hängt an ADC0
   Light_erfassungs_counter = 0                            ' Globaler Zähler vorinitialisieren
   Light_erfassung = False                                 ' Lichtwerterfassung AUS
   Light_value = 15                                        ' Lichtwert initialisieren
   Light_update_flag = False                               ' Update Flag
   Light_autoauffahren = 0                                 ' Globales Lichtflag initialisieren
   Light_autoauffahren_locked = 0                          ' Verriegelung zunächste auf 0 initiali-
sieren


' WIND:  Windsensor
   Wind_autoauffahren = 0                                  ' Globales Windflag initialisieren


' RAIN:  Regensesnor
   Rain_autoauffahren = 0                                  ' Globales Regenflag initialisieren


' LED-Array
   Led_array = 0                                           ' Vorinitialisierung des 16-Bit LED-
Array
   Led_key_flag = False                                    ' Zeitüberwachung aus


' SHT71: Sensor
   Sht71_auswertung = Sht71_auswertung_default             ' Zeitintervall für SHT71 auf Default
   Sht71_t_min = 0.0                                       ' Minimale Temperatur initialisieren
   Sht71_t_max = 0.0                                       ' Maximale Temperatur initialisieren
   Sht71_rhtrue_min = 0.0                                  ' Minimale Feuchtigkeit initialisieren
   Sht71_rhtrue_max = 0.0                                  ' Maximale Feuchtigkeit initialisieren
   Sht71_autozufahren = False                              ' Globales Temperaturflag initialisieren
   Sht71_autozufahren_locked = False                       ' Globales verriegeln, erst Hysterese


' EE: externes EEPROM
   Ee_datalogging_flag = True                              ' Datenlogging in externes EEPROM de-
fault EIN
   Ee_auswertung = Ee_auswertung_default                   ' Zeitintervall auf default setzen

   Ee_output_flag = False                                  ' Ausgabeflag initialisieren


' Timer: Timer
   Timer_open_flag = Timer_open_flag_default               ' Default-Wert für Funktion automati-
sches Öffnen laden
   Timer_close_flag = Timer_close_flag_default             ' Default-Wert für Funktion automati-
sches Schließen laden

   Timer_open_sekunde = 0                                  ' Sekundenvariable für Öffnungszeit
initialisieren
   Timer_open_minute = 0                                   ' Minutenvariable für Öffnungszeit ini-
tialisieren
   Timer_open_stunde = 0                                   ' Stundenvariable für Öffnungszeit ini-
tialisieren

   Timer_open_tag = 1                                      ' Tagvariable  für Öffnungszeit initia-
lisieren
   Timer_open_monat = 1                                    ' Monatsvariable für Öffnungszeit initi-
alisieren
   Timer_open_jahr = 8                                     ' Jahresvariable für Öffnungszeit initi-
alisieren

   Timer_close_sekunde = 0                                 ' Sekundenvraiable für Schließungszeit
initialisieren
   Timer_close_minute = 0                                  ' Minutenvariable für Schließungszeit
initialisieren
   Timer_close_stunde = 0                                  ' Stundenvariable für Schließungszeit
initialisieren

   Timer_close_tag = 1                                     ' Tagesvariable für Schließungszeit
initialisieren
   Timer_close_monat = 1                                   ' Monatsvariable für Schließungszeit
initialisieren
```

```
   Timer_close_jahr = 8                                    ' Jahresvariable für Schließungszeit
initialisieren


' Sonstiges
   Test_debug_flag = False                                 ' Globales test-Debug-Flag
   ' Anmerkung: Test_debug_flag wird als globales Steuerungsflag für Testausgaben
   ' vorgesehen, findet aber aktuell keine Verwendung da Steuerung von Traceausgaben
   ' über if-endif-Konstrukt mit globalen Konstanten erfolgt!


' ----- Gosub's -----

' LCD: Routinen und Einstellungen
   Gosub Lcd_konfig                                        ' Beleuchtung und der Rest
   Gosub Lcd_kontrastset                                   ' Kontrast noch einstellen


' FAN: Routinen und Einstellungen
   Gosub Fan_konfig                                        ' Konfiguration für Belüftung fertig


' LCD: Startwert für Dimmung festlegen und Dimmung starten
   Gosub Lcd_beleuchtung_an                                ' LCD-Beleuchtung einschalten


' FAN: Startwert für Lüfteranlauf festlegen und Lüfter starten
   Gosub Fan_lueftung_aus                                  ' Lüfter ausschalten


' DCF77: Softclock gemäß RTC einstellen
   Gosub Rtc_set_softclock                                 ' Softclock aus RTC laden


' REL: Routinen und Einstellungen
   Gosub Relais_konfig                                     ' Relais fertig konfigurieren


' DIM: Interrupt für Nulldurchgangserkennung konfigurieren
   Gosub Dimmer_konfig                                     ' Dimmer fertig konfigurieren


' WIND: Windsensensor konfigurieren
   Gosub Wind_konfig                                       ' Windsensor fertig konfigurieren


' SHT71: Temperatur- und Luftfeuchtigkeit
' Grundkonfiguration
   Gosub Sht71_connection_reset_sequence                   ' SHT71 initialisieren


' LED:  LED-Array initialisieren
   Gosub Led_clearlatch_set                                ' Latch aktivieren
   Led_array = &HFFFF                                      ' Globale Variable setzen (alle LED an!)
   Gosub Led_update_array                                  ' Latch updaten


' EE:  Internes EEPROM
   Gosub Ee_init_internal_eeprom                           ' Internes EEPROM konfigurieren


' EE:  externes EEPROM
   Gosub Ee_init_external_eeprom                           ' Externes I2C EEPROM konfigurieren


' ----- Interrupts -----

' Konfiguration der Interrupts für die Tastenabfrage
   On Int7 Key_isr_routine                                 ' Definition ISR
   Test_int_counter = &H0000                               ' Testcounter vorinitialisieren
   Enable Int7                                             ' INT7 freigeben
   Config Int7 = Falling                                   ' INT bei falling Edge


' INT5 konfigurieren für Dimmer-Nulldurchgangserkennung
   Config Int5 = Rising                                    ' Trigger bei steigender Flanke
   On Int5 Dimmer_null_irq                                 ' Springe zur Interrupt-Routine Dim-
mer_irq
   Enable Int5                                             ' Interrupt5 freigeben
```

```
' INT6 konfiguriert für Windimpulserfassung des Schalenkreuzanemometers
   Config Int6 = Rising                                    ' Trigger bei steigender Flanke
   On Int6 Wind_irq                                        ' Springe zur Interrupt-Routine Wind_irq

   Enable Interrupts                                       ' Alle Interrupts aktivieren


'-- Timer0 sofort anhalten und erst über SW anstossen
   ' Hinweis: Timer 2 wird zur Zündung des Triac verwendet.
   Stop Timer2


' ################################################################################
'
'                              Hauptprogramm ConvCtrl
'
' ################################################################################

' Controll-Output schreiben und los's gehts
#if Main_testmodus                                         ' Trace
   Print #2 , ""
   Print #2 , "** Enter MainLoop **"
   Print #2 , ""
#endif                                                     ' Trace

' Wenn Konstante gesetzt Trace Output schreiben
#if Tracejumper_testmodus                                  ' Trace
   Select Case Trace_jumper_value
      Case True:
         Print #2 , "Trace-Jumper ist gesetzt"
      Case False:
         Print #2 , "Trace-Jumper ist NICHT gesetzt"
   End Select                                              ' Trace-Jumper
#endif                                                     ' Trace

' Globales Test-Debug-Flag überprüfen und ggf. Traceoutput schreiben
If Test_debug_flag = True Then
   Print #2 , "Globales Test-Debug-Flag ist gesetzt!!"
End If                                                     ' Test-Debug-Flag

' Alive-LED einschalten
Alive_led = 1

' Eintritt in Hauptprogramm durch 2xBeep signalisieren wenn Soundflag gesetzt
If Sound_flag = True Then

   Call Sound_beep(sound_beep_time)                        ' Soubroutine Beep aufrufen
   Waitms 100                                              ' 100 ms warten
   Call Sound_beep(sound_beep_time)                        ' Soubroutine Beep aufrufen

End If                                                     ' Sound-Flag

' Display löschen
Cls                                                        ' Clear Screen

' Anzeige des Startbildschirms mit Uhrzeit und Datum
For Temp_byte_1 = 1 To Menue_scr03_delay

   ' Überschrift
   Locate 1 , 2 : Lcd "System bereit!"

   ' Uhrzeit und Tag ausgeben
   Locate 2 , 1 : Lcd Time$
   Locate 2 , 15 : Lcd Lookupstr(_weekday , Wochentag)

   ' Datum ausgeben
   Locate 3 , 1 : Lcd Date$

   ' Anzahl der erfolgten Synchronisationen auf LCD ausgeben
   Temp_byte_2 = Makebcd(dcf_sync_count)
   Locate 3 , 11 : Lcd Bcd(temp_byte_2)

   ' Zeitzone ausgeben
   Locate 3 , 16 : Lcd Dcf77timezone()

   ' Eine Sekunde warten!
   Wait 1

Next Temp_byte_1
```

```
' Display löschen
Cls                                                     ' Clear Screen

' LED-Array initialisieren
   ' Alle LED's des LED-Arrays aussschalten
   Led_array = 0

   ' ggf. MEM-Full-LED bereits setzen
   If Ee_data_overrun > 0 Then
      ' Externes EEPROM mindestens 1x übergelaufen => LED einschalten
      Set Led_10_mem_full
   End If                                               ' Überlaufzähler

   ' LED-Array updaten
   Gosub Led_update_array

' Display auf richtigen Bildschirm schalten
Gosub Menue_change_state_by_sub

' Auswertung der Umweltdaten starten

' Auswertung für Wind starten
Gosub Wind_erfassung_an                                 ' Interrupts freigeben und los gehts
Wind_erfassungs_counter = Sectic_count + Wind_auswertung    ' Auswerteintervall neu setzen

' Auswertung für Temperatur und Luftfeuchtigkeit starten
Sht71_erfassung = True                                  ' Auswertung Sht71 zulassen
Sht71_erfassungs_counter = Sectic_count + Sht71_auswertung  ' Auswerteintervall neu setzen

' ggf. Auswertung der Data-Logging-Methode starten
If Ee_datalogging_flag = True Then
   Ee_erfassungs_counter = Sectic_count + Ee_auswertung     ' Auswerteintervall neu setzen
End If                                                  ' Data-Logging-Flag

' Abhängig von den Einstellungen Lichtauswertung starten
If Rel_auf_dunkel_flag = True Then

   ' Licht einmalig erfassen
   Light_value = Light_measurement(light_messwert_zahl )

   ' zyklisches Update aktivieren
   Light_erfassung = True

   ' Auswerteintervall setzen
   Light_erfassungs_counter = Sectic_count + Light_auswertung_default

End If                                                  ' Automatisch auf bei Dunkelheit

' Abhängig von den Einstellung ggf. Screensaver starten
If Lcd_screensaver = True Then
   Lcd_screensaver_counter = Sectic_count + Lcd_screensaver_delay
   Lcd_pwmwert_buffer = Lcd_pwmwert                     ' Helligkeit buffern
End If                                                  ' Screensaver

' ------------------------------------------------
' ----- Hier ist die Programmhauptschleife -----
' ------------------------------------------------

Do

   ' ----- Tastatur abfragen und auswerten -----
   '
   ' +----+                        +----+
   ' | 11 |SW1                     | 14 |SW7
   ' +----+LED1                    +----+LED3
   ' +----+                        +----+
   ' | 21 |SW2      +----+         | 24 |SW8
   ' +----+LED2     | 12 |SW4      +----+LED4
   '                +----+LED7
   '    +----+      +----+      +----+
   '    | 41 |SW3   | 22 |SW5   | 44 |SW9
   '    +----+LED5  +----+LED9  +----+LED6
   '                +----+
   '                | 42 |SW6
   '                +----+LED8


   Led_array_save = Led_array                           ' LED-Array-Inhalt retten
```

```
   If Key_keypressed = True Then
      ' 50ms Zeit soll das Betätigen von zwei Tasten gleichzeitig vereinfachen
      ' -> wird benötigt für "Beschattung stopp" und "Licht aus"
      ' Außerdem dient die Zeit zum Entprellen der Tasten!!
      Waitms Key_waiting
      ' Tastenmatrix auslesen (Achtung, gleichzeigiges Betätigen innerhalb Reihe problematisch)
      Key_taste = Key_query_matrix()                        ' Abfrage der Tastaturmatrix
      Key_taste_buffer = Key_taste                          ' Taste buffern


      ' Ggf. Signalton ausgeben
      If Sound_flag = True Then
         Call Sound_beep(sound_beep_time_short)             ' Subroutine Beep aufrufen
      End If                                                ' Signalton

      ' **** Auswertung der Tastatur und Ausgabe ****

      ' Wenn Konstante gesetzt Trace Output schreiben
      #if Key_testmodus                                     ' Trace
         Print #2 , "Taste " ; Key_taste                    ' Anzeige der gelesenen Taste
      #endif                                                ' Trace

      ' Prüfung ob aktuell Screensaver verwendet
      If Lcd_screensaver = True Then
         ' JA, Screensaver ist eingeschaltet

         ' Zunächste Zeitintervall neu setzen
         Lcd_screensaver_counter = Sectic_count + Lcd_screensaver_delay

         ' Prüfen ob Screensaver aktiv oder nicht
         If Lcd_screensaver_aktiv = True Then
            ' JA, Screensaver ist aktiv

            ' Ggf. Trace ausgeben
            #if Screensaver_testmodus                       ' Trace
               Print #2 , "Screensaver wird passiv"
            #endif                                          ' Trace

            ' Taste ignorieren, zunächste Display einschalten
            Key_taste = 0

            ' Screensaver deaktivieren
            Lcd_screensaver_aktiv = False

            ' Helligkeit einstellen
            Call Lcd_set_helligkeit(lcd_pwmwert_buffer)

            ' Display einschalten
            Gosub Lcd_display_on

         End If                                             ' Screensaver aktiv
      End If                                                ' Screensaver eingeschaltet

      ' LED's für die jeweiligen Tasten einschalten
      Select Case Key_taste
         Case 11:                                           ' Taste "Beschattung auf"
            Set Led_01_sw1
         Case 21:                                           ' Taste "Beschattung zu"
            Set Led_02_sw2
         Case 41:                                           ' Taste Link
            Set Led_05_sw5
         Case 12:                                           ' Taste Hoch
            Set Led_07_sw7
         Case 22:                                           ' Taste Enter
            Set Led_09_sw9
         Case 42:                                           ' Taste Runter
            Set Led_08_sw8
         Case 14:                                           ' Taste "Licht heller"
            Set Led_03_sw3
         Case 24:                                           ' Taste "Licht dunkler"
            Set Led_04_sw4
         Case 44:                                           ' Taste Rechts
            Set Led_06_sw6
         Case 32:                                           ' Tastenkombination Beschattung Stopp
            Set Led_01_sw1
            Set Led_02_sw2
         Case 38:                                           ' Tastenkombination Licht Aus
            Set Led_03_sw3
            Set Led_04_sw4
      End Select                                            ' Key_taste
```

```
      Gosub Led_update_array
      Led_key_flag = True
      Led_key_counter = Sectic_count

      ' Hot-Keys prüfen
      Select Case Key_taste

         ' ### Tastencodes für Beschattung ###

         ' Beschattung auf betätigt
         Case 11:
            Key_taste = 0

            ' Prüfen auf Regen
            Temp_byte_1 = Rain_measurement()
            If Temp_byte_1 = Rain_no Then

               ' Es regnet nicht, also weiter

               ' Prüfen ob maximale Windgeschwindigkeit erreicht
               Temp_byte_2 = Wind_geschwindigkeit_kmh
               If Temp_byte_2 < Wind_beschattung_auf Then

                  ' Es windet nicht also weiter

                  Gosub Relais_beschattung_auf

                  If Menue_inrelmenue_flag <> 0 Then
                     Menue_state = Loadlabel(men2211)
                     Gosub Menue_change_state_by_sub
                  End If                                    ' Menüüberprüfung

               End If                                       ' Ende Windprüfung
            End If                                          ' Ende Regenprüfung

         ' Beschattung zu betätigt
         Case 21:
            Key_taste = 0

            ' Prüfen auf Regen
            Temp_byte_1 = Rain_measurement()
            If Temp_byte_1 = Rain_no Then

               ' Es regnet nicht, also weiter

               ' Prüfen ob maximale Windgeschwindigkeit erreicht
               Temp_byte_2 = Wind_geschwindigkeit_kmh
               If Temp_byte_2 < Wind_beschattung_auf Then

                  ' Es windet nicht also weiter

                  Gosub Relais_beschattung_zu

                  If Menue_inrelmenue_flag <> 0 Then
                     Menue_state = Loadlabel(men2221)
                     Gosub Menue_change_state_by_sub
                  End If                                    ' Menüüberprüfung

               End If                                       ' Ende Windprüfung
            End If                                          ' Ende Regenprüfung

         ' Beschattung auf/zu betätigt = Stop
         Case 32:
            Key_taste = 0

            ' Prüfen auf Regen
            Temp_byte_1 = Rain_measurement()
            If Temp_byte_1 = Rain_no Then

               ' Es regnet nicht, also weiter

               ' Prüfen ob maximale Windgeschwindigkeit erreicht
               Temp_byte_2 = Wind_geschwindigkeit_kmh
               If Temp_byte_2 < Wind_beschattung_auf Then

                  ' Es windet nicht also weiter

                  Gosub Relais_beschattung_stop
```

```
            If Menue_inrelmenue_flag <> 0 Then
               Menue_state = Loadlabel(men2231)
               Gosub Menue_change_state_by_sub
            End If                                          ' Menüüberprüfung

         End If                                             ' Ende Windprüfung
      End If                                                ' Ende Regenprüfung


   ' ### Tastencodes für Dimmer ###

   ' Licht heller betätigt
   Case 14:
      Key_taste = 0

      ' Wegen manuellem Eingriff in Licht Flashmode verriegeln
      Dim_flash_mode_locked = True

      ' Prüfen ob Beleuchtung bereits eingeschaltet
      If Beleuchtung_flag = Passiv Then
         ' Beleuchtung läuft noch nicht => Beleuchtung einschalten
         Gosub Beleuchtung_ein
         Gosub Beleuchtung_dauerlicht_aus
         ' Dimmer frei laufen lasse
         Free_running_flag = Aktiv
      Else                                                  ' Beleuchtungsflag
         ' Dimmer heller schalten und gegen Endwert prüfen
         If Dim_helligkeitsstufe < Dim_helligkeit_max Then
            ' Helligkeitsstufe erhöhen
            Incr Dim_helligkeitsstufe
         End If                                             ' Überprüfung Helligkeit
      End If                                                ' Beleuchtungsflag

      If Menue_indimmenue_flag <> 0 Then

         Select Case Menue_indimmenue_flag
            Case 1 To 3:
               Menue_state = Loadlabel(men3221)          'Bildschirm Dimmer heller
               Gosub Menue_change_state_by_sub
               Gosub Lcd_print
               Gosub Menue_men3221
               Menue_state_renew = 0
            Case 4:
               Gosub Menue_men312
         End Select                                         ' Überprüfung Dimmermenü

       End If                                               ' Menüüberprüfung

   ' Licht dunkler betätigt
   Case 24:
      Key_taste = 0

      ' Wegen manuellem Eingriff in Licht Flashmode verriegeln
      Dim_flash_mode_locked = True

      ' Prüfen ob Beleuchtung bereits eingeschaltet
      If Beleuchtung_flag = Passiv Then
         ' Beleuchtung läuft noch nicht => Beleuchtung einschalten
         Gosub Beleuchtung_ein
         Gosub Beleuchtung_dauerlicht_aus
         ' Dimmer frei laufen lasse
         Free_running_flag = Aktiv
      Else                                                  ' Beleuchtungsflag
         ' Dimmer dunkler schalten und gegen Endwert prüfen
         If Dim_helligkeitsstufe > Dim_helligkeit_min Then
            ' Helligkeitsstufe veringern
            Decr Dim_helligkeitsstufe
         End If                                             ' Überprüfung Helligkeit
      End If                                                ' Beleuchtungsflag

      If Menue_indimmenue_flag <> 0 Then

         Select Case Menue_indimmenue_flag
            Case 1 To 3:
               Menue_state = Loadlabel(men3231)          'Bildschirm Dimmer dunkler
               Gosub Menue_change_state_by_sub
               Gosub Lcd_print
               Gosub Menue_men3231
               Menue_state_renew = 0
            Case 4:
```

```
                Gosub Menue_men312
            End Select                                          ' Überprüfung Dimmermenü

         End If                                                 ' Menüüberprüfung

      ' Licht heller/dunkler betätigt = Aus
      Case 38:
         Key_taste = 0

         ' Wegen manuellem Eingriff in Licht Flashmode verriegeln
         Dim_flash_mode_locked = True

         ' Dimmer ausschalten
         Free_running_flag = Passiv
         If Dauerlicht_flag = Passiv Then
            Gosub Beleuchtung_dauerlicht_ein
            Gosub Beleuchtung_aus
         End If                                                 ' Überprüfung Dauerlicht

         If Menue_indimmenue_flag <> 0 Then

            Select Case Menue_indimmenue_flag
               Case 1 To 3:
                  Menue_state = Loadlabel(men3241)              'Startbildschirm
                  Gosub Menue_change_state_by_sub
                  Gosub Lcd_print
                  Menue_state_renew = 0
            End Select                                          ' Überprüfung Dimmermenü

         End If                                                 ' Menüüberprüfung

         ' Dimmer wurde sauber ausgeschaltet => aktueller Helligkeitswert speichern

         ' Prüfung ob sich Werte (geseichert / aktuell) unterscheiden
         Temp_byte_1 = Ee_data_dimmer_aktuell
         If Temp_byte_1 <> Dim_helligkeitsstufe Then
            ' Es wird nur in das EEPROM geschrieben, wenn sich die aktuellen
            ' Werte von den gespeicherten unterscheiden!
            Ee_data_dimmer_aktuell = Dim_helligkeitsstufe
         End If                                                 ' EEPROM-Zellenprüfung

   End Select                                                   ' Key_taste


   ' Prüfung auf ShortCut-Funktion
   If Key_shortcut_status = True Then
      ' JA, ShortCut soll aktiv sein

      ' Ggf. Trace ausgeben
      #if Shortcut_testmodus                                    ' Trace
         Print #2 , "ShortCut ist aktiv!"
      #endif                                                    ' Trace

      If Key_taste = 22 Then

         ' Beschattung aktuell aktiv
         Temp_byte_1 = Beschattung_auf_flag + Beschattung_zu_flag
         If Temp_byte_1 <> 0 Then
            ' JA, Beschattung wird aktuell entweder auf oder zu gefahren und
            ' kann damit gestopt werden.

            ' Ggf. Trace ausgeben
            #if Shortcut_testmodus                              ' Trace
               Print #2 , "Beschattung ist aktiv"
            #endif                                              ' Trace

            Key_taste = 0

            ' Prüfen auf Regen
            Temp_byte_1 = Rain_measurement()
            If Temp_byte_1 = Rain_no Then

               ' Es regnet nicht, also weiter

               ' Prüfen ob maximale Windgeschwindigkeit erreicht
               Temp_byte_2 = Wind_geschwindigkeit_kmh
               If Temp_byte_2 < Wind_beschattung_auf Then

                  ' Es windet nicht also weiter
```

```
                    Gosub Relais_beschattung_stop

                    If Menue_inrelmenue_flag <> 0 Then
                       Menue_state = Loadlabel(men2231)
                       Gosub Menue_change_state_by_sub
                    End If                                          ' Menüüberprüfung

                 End If                                             ' Ende Windprüfung
              End If                                                ' Ende Regenprüfung

           Else                                                     ' Regenüberprüfung
              ' Prüfen ob aktuell Dimmer arbeitet
              If Free_running_flag = True Then
                 ' JA, Dimmer ist aktuell aktiv

                 ' Ggf. Trace ausgeben
                 #if Shortcut_testmodus                             ' Trace
                    Print #2 , "Dimmer ist aktiv"
                 #endif                                             ' Trace

                 Key_taste = 0
                 ' Dimmer ausschalten
                 Free_running_flag = Passiv
                 If Dauerlicht_flag = Passiv Then
                    Gosub Beleuchtung_dauerlicht_ein
                    Gosub Beleuchtung_aus
                 End If                                             ' Überprüfung Dauerlicht

                 If Menue_indimmenue_flag <> 0 Then

                    Select Case Menue_indimmenue_flag
                       Case 1 To 3:
                          Menue_state = Loadlabel(men3241) 'Startbildschirm
                          Gosub Menue_change_state_by_sub
                          Gosub Lcd_print
                          Menue_state_renew = 0
                    End Select                                      ' Überprüfung Dimmermenü

                 End If                                             ' Menüüberprüfung

                 ' Dimmer wurde sauber ausgeschaltet => aktueller Helligkeitswert speichern
                 Ee_data_dimmer_aktuell = Dim_helligkeitsstufe

              End If                                                ' Dimmer
           End If                                                   ' Beschattung
        End If                                                      ' Enter
     End If                                                         ' Shortcut Funktion

     ' Wenn Taste noch nicht durch Hot-Key abgefangen dann Menü
     If Key_taste <> 0 Then

        'Menüeintrag und Tastencodes finden und ggf. State wechseln
        'hier nur Tastendruck auswerten
        Gosub Menue_change_state

        'Pointer nach Statuswechsel neu setzen
        If Menue_state_renew = 1 Then
           Gosub Menue_change_state
        End If                                                      ' Menü-Update angefordert

        ' Bereits schon LCD refresh auslösen bevor Unterprogramm angesprungen
        ' wird da im unterprogramm ggf. noch Inhalte ergänzt werden.
        If Menue_state_renew = 1 Then
           Menue_state_renew = 0
           Gosub Lcd_print
        End If                                                      ' Erneutes Menü-Update prüfen

        'Unterprogramm des aktuellen State anspringen
        Gosub Menue_call_subroutine

     End If                                                         ' Menüsteuerung

     ' globales Flag für Signalisierung zurücksetzen da Auswertung jetzt durchgeführt
     Key_keypressed = False

     ' Nach Auswertung Tastatur für erneute Eingabe wieder freigeben
     Gosub Key_enable_keyinterrupt

  End If                                                            ' Tastatureingabe
```

```
   ' Nochmals LCD refresh falls Menü von außen verändert
   If Menue_state_renew = 1 Then
      Menue_state_renew = 0
      Gosub Lcd_print
   End If                                                  ' Menü-Update angefordert

   '----- Ende Menue-modul -----


   ' ----- DCF77-Modul zur Behandlung von Synchronisationen -----
   ' DCF77 Statusbits abfragen und maskieren
   Temp_byte_1 = Dcf_status And &H80

   ' Prüfung ob SoftClock durch DCF77 synchronisiert (Bit 7 aus Statusbyte)
   If Temp_byte_1 = &H80 Then

      ' Trace-Output schreiben
      #if Dcf77_testmodus                                  ' Trace
         Print #2 , "SYNC: SoftClock wurde durch DCF77 synchronisiert!"
      #endif                                               ' Trace

      ' Sync-Counter erhöhen
      Incr Dcf_sync_count

      ' Zur Kontrolle bei erfolgreichem SYNC Status-Bit zurücksetzen
      Reset Dcf_status.7

      ' DS1307 synchronisieren
      Gosub Rtc_set_rtc_clock

   End If                                                  ' Synchronisation erfolgt?

   ' ----- Ende DCF77-Modul  -----


   ' ===== SW-Timer-Funktionen =====
   Gosub Sw_timer_sequencer


   ' ===== Steuerungs- und Regelfunktionen =====

   ' [1] Prüfung auf Regen und ggf. Beschattung auffahren und verriegeln
   Temp_byte_3 = Rain_measurement()                        ' Rain_no / Rain_yes
   If Temp_byte_3 = Rain_yes Then

      ' Regen aktuell erkannt
      If Rain_autoauffahren = Passiv Then

         ' Bisher hatte es noch nicht geregnet aber Regen wurde jetzt erkannt
         Rain_autoauffahren = Aktiv                        ' Regenflag setzen
         Set Led_13_open_rain                              ' Regen-LED setzen

         ' Beschattung auffahren
         Gosub Relais_beschattung_auf

         If Menue_inrelmenue_flag <> 0 Then
            Menue_state = Loadlabel(men2211)
            Gosub Menue_change_state_by_sub
         End If                                            ' Menüflag

         ' Ggf. Signalton ausgeben
         If Sound_flag = True Then
            Call Sound_beep(sound_beep_time_short)         ' Subroutine Beep aufrufen
            Waitms 40
            Call Sound_beep(sound_beep_time_short)         ' Subroutine Beep aufrufen
            Waitms 40
            Call Sound_beep(sound_beep_time_short)         ' Subroutine Beep aufrufen
         End If                                            ' Signalton
      End If                                               ' Regen-Auffahr-Flag
   Else                                                    ' Regenmessung
      ' Es regnet nicht mehr => Auffahrgrund existiert nicht mehr
      Reset Led_13_open_rain                               ' Regen-LED zurücksetzen
   End If                                                  ' Regenmessung


   ' [2] Prüfung auf Windgrenze und ggf. Beschattung auffahren und verriegeln
   Temp_byte_1 = Wind_geschwindigkeit_kmh
   If Temp_byte_1 >= Wind_beschattung_auf Then

      ' Aktuell zu starker Wind erkannt, Grenzwert erreicht
```

```
      If Wind_autoauffahren = Passiv Then

         ' Bisher Windgrenze noch nicht rreicht aber nur Grenzwert überschritten
         Wind_autoauffahren = Aktiv                          ' Windflag setzen
         Set Led_14_open_wind                                ' Wind-LED setzen

         ' Beschattung auffahren
         Gosub Relais_beschattung_auf

         If Menue_inrelmenue_flag <> 0 Then
            Menue_state = Loadlabel(men2211)
            Gosub Menue_change_state_by_sub
         End If                                              ' Menüflag

         ' Ggf. Signalton ausgeben
         If Sound_flag = True Then
            Call Sound_beep(sound_beep_time_short)           ' Subroutine Beep aufrufen
            Waitms 40
            Call Sound_beep(sound_beep_time_short)           ' Subroutine Beep aufrufen
            Waitms 40
            Call Sound_beep(sound_beep_time_short)           ' Subroutine Beep aufrufen
         End If                                              ' Signalton

      End If                                                 ' Wind-Auffahr-Flag
   Else                                                      ' Windmessung

      ' Wind ist kleiner als Grenzwert => Auffahrgrund existiert nicht mehr
      Reset Led_14_open_wind                                 ' Wind-LED zurücksetzen

   End If                                                    ' Windmessung


   ' [3] Prüfen auf abgelaufenen OPEN Timer
   If Timer_open_flag = Aktiv Then
      ' Timerüberwachung ist aktiv, es kann geprüft werden ob Uhrzeit erreicht
      Temp_string_1 = Date$

      ' Prüfen ob Datum korrekt

      ' Datum für Vergleich zusammenbauen
      Temp_byte_1 = Makebcd(timer_open_tag)
      Temp_byte_2 = Makebcd(timer_open_monat)
      Temp_byte_3 = Makebcd(timer_open_jahr)

      ' Anscheinend kann BASCOM nun auch verschachtelte Operationen :-)
      Temp_string_2 = Bcd(temp_byte_1) + "." + Bcd(temp_byte_2) + "." + Bcd(temp_byte_3)
      ' Sonst:
      'Temp_string_2 = Temp_string_2 + Bcd(temp_byte_2)
      'Temp_string_2 = Temp_string_2 + "."
      'Temp_string_2 = Temp_string_2 + Bcd(temp_byte_3)

      ' Ggf. Trace ausgeben
      #if Timer_testmodus                                    ' Trace
         Print #2 , "Systemdatum : " ; Temp_string_1
         Print #2 , "Timer Open  : " ; Temp_string_2
      #endif                                                 ' Trace

      If Temp_string_1 = Temp_string_2 Then

         ' Ja, Datum ist korrekt => prüfen ob Zeit korrekt

         Temp_string_1 = Time$

         ' Zeit zum Vergleich zusammenbauen
         Temp_byte_1 = Makebcd(timer_open_stunde)
         Temp_byte_2 = Makebcd(timer_open_minute)
         Temp_byte_3 = Makebcd(timer_open_sekunde)

         ' Verschachtelten String zusammenbauen

         Temp_string_2 = Bcd(temp_byte_1) + ":" + Bcd(temp_byte_2) + ":" + Bcd(temp_byte_3)

         ' Ggf. Trace ausgeben
         #if Timer_testmodus                                 ' Trace
            Print #2 , "Systemzeit : " ; Temp_string_1
            Print #2 , "Timer Open : " ; Temp_string_2
         #endif                                              ' Trace

         If Temp_string_1 = Temp_string_2 Then
```

```
            ' JA, Zeit ist auch korrekt => geplante Aktion kann ausgeführt werden

            ' Ggf. Trace ausgeben
            #if Timer_testmodus                                 ' Trace
               Print #2 , "Timer OPEN aktiv => Beschattung auffahren"
            #endif                                              ' Trace

            ' Anmerkung: Unabhängg davon, das jederzeit aufgefahren werden kann
            ' soll durch die Überprüfung von Wind und Regen sichergestellt werden,
            ' dass die Antriebe unnötig nochmal durch den OPEN-Timer bestromt
            ' werden, obwohl dieser schon geöffnet ist.

            ' Prüfen auf Regen
            Temp_byte_1 = Rain_measurement()
            If Temp_byte_1 = Rain_no Then

               ' Es regnet nicht, also weiter

               ' Prüfen ob maximale Windgeschwindigkeit erreicht
               Temp_byte_2 = Wind_geschwindigkeit_kmh
               If Temp_byte_2 < Wind_beschattung_auf Then

                  ' Es windet nicht also weiter

                  Gosub Relais_beschattung_auf

                  ' ggf. Menü anpassen
                  If Menue_inrelmenue_flag <> 0 Then
                     Menue_state = Loadlabel(men2211)
                     Gosub Menue_change_state_by_sub
                  End If                                        ' Menü-Update
'

                  ' Ggf. Signalton ausgeben
                  If Sound_flag = True Then
                     Call Sound_beep(sound_beep_time_short) ' Subroutine Beep aufrufen
                     Waitms 40
                     Call Sound_beep(sound_beep_time_short) ' Subroutine Beep aufrufen
                     Waitms 40
                     Call Sound_beep(sound_beep_time_short) ' Subroutine Beep aufrufen
                  End If                                        ' Signalton

                  ' Status LED setzen
                  Set Led_11_open_time

               End If                                           ' Ende Windprüfung
            End If                                              ' Ende Regenprüfung

            ' Timer flag zurücksetzen da Aktion nur einmal ausgeführt werden soll
            Timer_open_flag = Passiv

         End If                                                 ' Zeit
      End If                                                    ' Datum
   End If                                                       ' Timer_open_flag


   ' [4] Prüfen auf abgelaufenen CLOSE Timer
   If Timer_close_flag = Aktiv Then
      ' Timerüberwachung ist aktiv, es kann geprüft werden ob Uhrzeit erreicht
      Temp_string_1 = Date$

      ' Prüfen ob Datum korrekt

      ' Datum für Vergleich zusammenbauen
      Temp_byte_1 = Makebcd(timer_close_tag)
      Temp_byte_2 = Makebcd(timer_close_monat)
      Temp_byte_3 = Makebcd(timer_close_jahr)

      ' Anscheinend kann BASCOM nun auch verschachtelte Operationen :-)
      Temp_string_2 = Bcd(temp_byte_1) + "." + Bcd(temp_byte_2) + "." + Bcd(temp_byte_3)

      ' Ggf. Trace ausgeben
      #if Timer_testmodus                                       ' Trace
         Print #2 , "Systemdatum : " ; Temp_string_1
         Print #2 , "Timer Close : " ; Temp_string_2
      #endif                                                    ' Trace

      If Temp_string_1 = Temp_string_2 Then

         ' Ja, Datum ist korrekt => prüfen ob Zeit korrekt
```

```
      Temp_string_1 = Time$

      ' Zeit zum Vergleich zusammenbauen
      Temp_byte_1 = Makebcd(timer_close_stunde)
      Temp_byte_2 = Makebcd(timer_close_minute)
      Temp_byte_3 = Makebcd(timer_close_sekunde)

      ' Verschachtelten String zusammenbauen

      Temp_string_2 = Bcd(temp_byte_1) + ":" + Bcd(temp_byte_2) + ":" + Bcd(temp_byte_3)

      ' Ggf. Trace ausgeben
      #if Timer_testmodus                                  ' Trace
         Print #2 , "Systemzeit : " ; Temp_string_1
         Print #2 , "Timer Close: " ; Temp_string_2
      #endif                                               ' Trace

      If Temp_string_1 = Temp_string_2 Then

         ' JA, Zeit ist auch korrekt => geplante Aktion kann ausgeführt werden

         ' Ggf. Trace ausgeben
         #if Timer_testmodus                               ' Trace
            Print #2 , "Timer CLOSE aktiv => Beschattung zufahren"
         #endif                                            ' Trace


         ' Prüfen auf Regen
         Temp_byte_1 = Rain_measurement()
         If Temp_byte_1 = Rain_no Then

            ' Es regnet nicht, also weiter

            ' Prüfen ob maximale Windgeschwindigkeit erreicht
            Temp_byte_2 = Wind_geschwindigkeit_kmh
            If Temp_byte_2 < Wind_beschattung_auf Then

               ' Es windet nicht also weiter

               ' Beschattung zufahren
               Gosub Relais_beschattung_zu

               ' ggf. Menü anpassen
               If Menue_inrelmenue_flag <> 0 Then
                  Menue_state = Loadlabel(men2221)
                  Gosub Menue_change_state_by_sub
               End If                                      ' Menüanpassung

               ' Ggf. Signalton ausgeben
               If Sound_flag = True Then
                  Call Sound_beep(sound_beep_time_short) ' Subroutine Beep aufrufen
                  Waitms 40
                  Call Sound_beep(sound_beep_time_short) ' Subroutine Beep aufrufen
                  Waitms 40
                  Call Sound_beep(sound_beep_time_short) ' Subroutine Beep aufrufen
               End If                                      ' Signalton

               ' Status LED setzen
               Set Led_15_close_time

            End If                                         ' Ende Windprüfung
         End If                                            ' Ende Regenprüfung

         ' Timer flag zurücksetzen da Aktion nur einmal ausgeführt werden soll
         Timer_close_flag = Passiv

      End If                                               ' Zeit
   End If                                                  ' Datum
End If                                                     ' Timer_close_flag


' [5] Prüfen auf Licht für automatisches Auffahren
If Rel_auf_dunkel_flag = True Then
   ' Helligkeitsüberprüfung ist aktiv, es kann geprüft werden ob Dunkelheit erreicht

   ' Gedanken zum lichtgesteuerten Zufahren:
   ' - Zufahren nur erlaubt wenn Lichtfunktionen deaktiv (kein Dimmer eingeschaltet)
   ' - Fuktion darf nur pro Tag einmal "auffahren" auslösen
```

```
      ' Prüfen ob Dunkelheitssteuerung nicht verriegelt
      If Light_autoauffahren_locked = False Then

         ' Prüfen ob künstliche Beleuchtung eingeschaltet
         If Beleuchtung_flag = False Then

            ' NEIN, künstliche Beleuchtung welche messung stören würde ist nicht
            ' eingeschaltet, somit kann Routine fortgesetzt werden.

            ' Prüfen ob Dunkelheitsgrenze unterschritten
            If Light_value <= Rel_auf_dunkelheit Then
               ' JA, Beschattung auffahren

               ' Anmerkung: Unabhängg davon, das jederzeit aufgefahren werden kann
               ' soll durch die Überprüfung von Wind und Regen sichergestellt werden,
               ' dass die Antriebe unnötig nochmal durch den OPEN-Timer bestromt
               ' werden, obwohl dieser schon geöffnet ist.

               ' Prüfen auf Regen
               Temp_byte_1 = Rain_measurement()
               If Temp_byte_1 = Rain_no Then

                  ' Es regnet nicht, also weiter

                  ' Prüfen ob maximale Windgeschwindigkeit erreicht
                  Temp_byte_2 = Wind_geschwindigkeit_kmh
                  If Temp_byte_2 < Wind_beschattung_auf Then

                     ' Es windet nicht also weiter

                     ' Beschattung auffahren
                     Gosub Relais_beschattung_auf

                     ' ggf. Menü anpassen
                     If Menue_inrelmenue_flag <> 0 Then
                        Menue_state = Loadlabel(men2211)
                        Gosub Menue_change_state_by_sub
                     End If                                    ' Menüanpassung

                     ' Ggf. Signalton ausgeben
                     If Sound_flag = True Then
                        Call Sound_beep(sound_beep_time_short)       ' Subroutine Beep aufrufen
                        Waitms 40
                        Call Sound_beep(sound_beep_time_short)       ' Subroutine Beep aufrufen
                        Waitms 40
                        Call Sound_beep(sound_beep_time_short)       ' Subroutine Beep aufrufen
                     End If                                    ' Signalton

                     ' Status-LED einschalten
                     Set Led_12_open_light

                     ' Zustand für Menü richtig setzen
                     Light_autoauffahren = Aktiv

                  End If                                       ' Ende Windprüfung
               End If                                          ' Ende Regenprüfung

               ' Verriegelung gegen erneutes Auffahren setzen
               Light_autoauffahren_locked = True

            End If                                             ' Licht
         End If                                                ' Künstliche Beleuchtung
      End If                                                   ' Verriegelung
   End If                                                      ' Dunkel flag


   ' [6] Prüfen auf Temperatur für automatisches Zufahren
   If Rel_zu_temp_flag = True Then
      ' Temperaturüberprüfung ist aktiv, es kann geprüft werden ob Temperatur erreicht
      ' ACHTUNG: Nur zufahren bei kein Regen und kein Wind!

      ' Gedanken zum temperaturgesteuerten Zufahren:
      ' - Für Funktion wird Hysterese implementiert mit der "toggeln" verhindert
      '   werden soll

      ' Prüfen ob Temperatursteuerung nicht verriegelt
      If Sht71_autozufahren_locked = False Then
         ' JA, Beschattung kann zugefahren werden

         Temp_byte_1 = Sht71_t                                 ' Casting von Single in Byte
```

```
      ' Prüfen ob Temperaturgrenze überschritten
      If Temp_byte_1 >= Rel_zu_temperatur Then

         'Ja, Temperaturgrenze erreicht

         ' Prüfen auf Regen
         Temp_byte_1 = Rain_measurement()
         If Temp_byte_1 = Rain_no Then

            ' Es regnet nicht, also weiter

            ' Prüfen ob maximale Windgeschwindigkeit erreicht
            Temp_byte_2 = Wind_geschwindigkeit_kmh
            If Temp_byte_2 < Wind_beschattung_auf Then

               ' Es windet nicht also weiter

               ' Beschattung zufahren
               Gosub Relais_beschattung_zu

               ' ggf. Menü anpassen
               If Menue_inrelmenue_flag <> 0 Then
                  Menue_state = Loadlabel(men2221)
                  Gosub Menue_change_state_by_sub
               End If                                    ' Menüanpassung

               ' Ggf. Signalton ausgeben
               If Sound_flag = True Then
                  Call Sound_beep(sound_beep_time_short) ' Subroutine Beep aufrufen
                  Waitms 40
                  Call Sound_beep(sound_beep_time_short) ' Subroutine Beep aufrufen
                  Waitms 40
                  Call Sound_beep(sound_beep_time_short) ' Subroutine Beep aufrufen
               End If                                    ' Signalton

               ' Status-LED einschalten
               Set Led_16_close_temp

               ' Zustand für Menü richtig setzen
               Sht71_autozufahren = Aktiv

            End If                                       ' Ende Windprüfung
         End If                                          ' Ende Regenprüfung

         ' Verriegelung gegen erneutes Zufahren setzen
         Sht71_autozufahren_locked = True

      End If                                             ' Temperatur
   End If                                                ' Verriegelung
End If                                                   ' Temp Flag


' [7] Prüfen ob Verriegelung für lichtgesteuertes Auffahren zurückgesetzt werden kann

' Anmerkung: Das Flag für die Dunkelheitssteuerung soll verhindern, dass bei
' Helligkeitsschwankungen oder bei Einschalten der künstlichen Beleuchtung
' und bei erneutem Unterschreiten der eingestellten Helligkeitsschwelle das
' Relais erneut anspricht.
' Die Lösung sieht vor, ggf. erst ab erreichen der Uhrzeit 12:00 Uhr das Flag
' erneut für die Steuerung der Beschattung zuzulassen.

If Light_autoauffahren_locked = True Then
   ' JA, lockflag ist gesetzt

   ' Prüfen on Uhrzeit 12:00 Uhr Mittags, dann lockbit zurücksetzen
   If _hour = 12 Then
      ' Ja, 12 Uhr Mittag erreicht

      ' Flag für Dunkelheitseinstellung zurücksetzen
      Light_autoauffahren_locked = False

      ' Status-LED ausschalten
      Reset Led_12_open_light

   End If                                                ' Uhrzeitprüfung
End If                                                   ' Flagüberprüfung


' [8] Prüfug ob Verriegelung für temperaturgesteuertes Zufahren zurückgesetzt werden kann
```

```
   If Sht71_autozufahren_locked = True Then
      ' Ja, lockflag ist gesetzt

      Temp_byte_1 = Rel_zu_temperatur - Sht71_hysterese
      Temp_byte_2 = Sht71_t                                  ' Casting Single to Byte
      If Temp_byte_2 < Temp_byte_1 Then
         ' Ja, hysterese ist unterschritten

         ' Flag für Temperatur zurücksetzen
         Sht71_autozufahren_locked = False

         ' Status-LED ausschalten
         Reset Led_16_close_temp

      End If                                                 ' Temperaturprüfung
   End If                                                    ' Flagüberprüfung


   ' [9] Prüfung ob Lüftersteuerung und Lüfter einregeln

   If Fan_sw_betrieb = True Then
      ' JA, Lüfter soll via SW gesteuert werden

      ' Geschwindigkeitsintervalle festlegen
      Temp_byte_1 = Sht71_t                                  ' Casting Single to Byte

      ' Ggf. Trace ausgeben
      #if Main_testmodus                                     ' Trace
         Print #2 , "Luefterssteuerung via SW"
         Print #2 , "Bezugstemperatur : " ; Temp_byte_1
      #endif                                                 ' Trace

      If Temp_byte_1 >= Fan_100_temperatur Then
         Call Fan_set_geschwindigkeit(fan_speed_100)

         ' Ggf. Trace ausgeben
         #if Main_testmodus                                  ' Trace
            Print #2 , "Luefterssteuerung 100%"
         #endif                                              ' Trace
      Else                                                   ' Stufe 100%
         If Temp_byte_1 >= Fan_75_temperatur Then
            Call Fan_set_geschwindigkeit(fan_speed_75)

            ' Ggf. Trace ausgeben
            #if Main_testmodus                               ' Trace
               Print #2 , "Luefterssteuerung 75%"
            #endif                                           ' Trace
         Else                                                ' Stufe 75%
            If Temp_byte_1 >= Fan_50_temperatur Then
               Call Fan_set_geschwindigkeit(fan_speed_50)

               ' Ggf. Trace ausgeben
               #if Main_testmodus                            ' Trace
                  Print #2 , "Luefterssteuerung 50%"
               #endif                                        ' Trace
            Else                                             ' Stufe 50%
               If Temp_byte_1 >= Fan_25_temperatur Then
                  Call Fan_set_geschwindigkeit(fan_speed_25)

                  ' Ggf. Trace ausgeben
                  #if Main_testmodus                         ' Trace
                     Print #2 , "Luefterssteuerung 25%"
                  #endif                                     ' Trace
               Else                                          ' Stufe 25%
                  Call Fan_set_geschwindigkeit(fan_belueftung_aus)

                  ' Ggf. Trace ausgeben
                  #if Main_testmodus                         ' Trace
                     Print #2 , "Luefterssteuerung NULL"
                  #endif                                     ' Trace
               End If                                        ' Stufe 25%
            End If                                           ' Stufe 50%
         End If                                              ' Stufe 75%
      End If                                                 ' Stufe 100%
   End If                                                    ' Flag für SW Lüfterbetrieb


   ' [10] Flashmode

   ' Gedanken und Anmerkungen zur Implementierung des Flash-Mode:
```

```
' Der Flashmode beschreibt die Funktion, dass bei fortschreitender Dunkelheit
' der Dimmer auf kleinster Stufe eingeschaltet und "angefahren" wird, bis
' der Lichtsensor wieder seinen Maximalwert liefert.
' Folgende Regel sollen dabei gelten:
' - Flashmode erst ab Uhrzeit 15:00 Uhr aktiv
' - Manueller Eingriff über Tasten oder Menü deaktiiert sofort Flashmode
' - Helligkeitsrampe Flashmode ersetzt die bisherige Helligkeitseinstellung
' - Helligkeitswert der Lampe kann durch Flashmode nur erhöht aber nicht
'   verringert werden!
' - Flashmode kann nur aktiv werden wenn Beleuchtung noch aus.

' Prüfen ob Flashmode aktiv
If Dim_flash_mode = Aktiv Then
   ' JA, Flashmode ist aktiv

   ' Prüfen ob Flash-Mode nicht durch manuellen Eingriff verriegelt
   If Dim_flash_mode_locked = False Then
      ' NEIN, Flashmode ist nicht veriegelt

      ' Prüfen ob Zeit nach 15:00 Uhr
      Temp_byte_1 = _hour
      If Temp_byte_1 > 15 Then
         ' JA, es ist nach 15:00 Uhr

         ' Ggf. Trace Output schreiben
         #if Flashmode_testmodus                          ' Trace
            Print #2 , "Alle Bedingungen für Flashmode sind erfuellt!"
         #endif                                           ' Trace

         ' Prüfen ob Update auf Licht-Wert stattgefunden hat
         If Light_update_flag = True Then
            ' Helligkeitswert überwachen
            If Light_value < Dim_flash_grenzhelligkeit Then
               ' JA, definierte Grenzhelligkeit ist unterschritten

               ' Prüfen ob Beleuchtung noch aus
               If Beleuchtung_flag = Passiv Then

                  ' Ja, Beleuchtung ist noch aus

                  ' Für Flash-Mode mit kleinster Einstellung starten
                  Dim_helligkeitsstufe = Dim_helligkeit_min

                  ' Beleuchtung läuft noch nicht => Beleuchtung einschalten
                  Gosub Beleuchtung_ein
                  Gosub Beleuchtung_dauerlicht_aus
                  ' Dimmer frei laufen lasse
                  Free_running_flag = Aktiv

               Else                                       ' Überprüfung Beleuchtung

                  ' So lange über Algorithmus Helligkeit der Dimmund erhöhen bis
                  ' entweder Maximalwert erreicht oder Helligkeit ausreichend

                  ' Ist Maximalwert schon erreicht?
                  If Dim_helligkeitsstufe < Dim_helligkeit_max Then
                     Incr Dim_helligkeitsstufe
                  End If                                  ' Maximale Helligkeitsstufe

               End If                                     ' Beleuchtung

               ' Flag wieder zurücksetzen
               Light_update_flag = False

            End If                                        ' Update Lichtwert
         End If                                           ' Grenzhelligkeit
      End If                                              ' 15:00 Uhr Grenze
   End If                                                 ' Verriegelung Flash-Mode
End If                                                    ' Flag für Flashmode


' [11] Prüfen ob Flash-Mode-Verriegelung entriegelt werden kann
' Prüfen ob Flashmode aktiv
If Dim_flash_mode = Aktiv Then
   ' JA, Flashmode ist aktiv

   ' Prüfen ob Zeit gleich 15:00 Uhr
   Temp_byte_1 = _hour
   If Temp_byte_1 = 15 Then
      ' JA, es ist 15:00 Uhr
```

```
         ' Prüfen ob Beleuchtung noch aus
         If Beleuchtung_flag = False Then
            ' Ja, Beleuchtung ist noch aus

            ' Verriegelung des Flash-Mode aufheben
            Dim_flash_mode_locked = False

            ' Ggf. Trace Output schreiben
            #if Flashmode_testmodus                          ' Trace
               Print #2 , "Verriegelung Flashmode wurde aufgehoben"
            #endif                                           ' Trace

         End If                                              ' Beleuchtung
      End If                                                 ' 15:00 Uhr Grenze
   End If                                                    ' Flag für Flashmode


   ' ===== Restarbeiten =====

   ' Prüfen ob LED-Array upgedated werden muss
   If Led_array_save <> Led_array Then
      Gosub Led_update_array                                 ' LED-Anzeige updaten
   End If                                                    ' Array updaten
Loop

'## End Hauptprogramm ###########################################################


'*******************************************************************************
' Interruptroutinen
'*******************************************************************************

'-------------------------------------------------------------------------------
' Interrupt-Service-Routine (INT7): Key_isr_routine
'
'-------------------------------------------------------------------------------
Key_isr_routine:

   ' INT7 sperren damit keine weitere Tasteneingabe mehr ausgelöst wird.
   Disable Int7
   ' globales Flag für Tasteeingabe setzen
   Key_keypressed = True

   ' Ggf. Trace Output schreiben
   #if Key_testmodus                                         ' Trace
      ' Interrupt-Counter erhöhen
      Incr Test_int_counter
      Print #2 , "* " ; Test_int_counter
   #endif                                                    ' Trace

Return
'-- End Key_isr_routine ------------------------------------------------------


'-------------------------------------------------------------------------------
' Service-Routine DCF77_sectic für DCF77 Softclock
' Routine wird als Parameter bei Config DCF77 übergeben und automatisch jede
' Sekunde ausgeführt. Damit ist SekundenTick für Gesamtsystem ohne zusätzlichen
' Timer möglich.
'-------------------------------------------------------------------------------
Sectic:

   ' Sectic_counter ist Zähler für Scheduler / Sequencer -> zählt immer weiter
   Incr Sectic_count

   ' Sectic-LED toggelt mit einer Frequenz von 0,5 Hz
   Toggle Sectick_led

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Dcf77_testmodus                                       ' Trace
      Print #2 , ""
      Print #2 , "tic :" ; Sectic_count
      Print #2 , ""
   #endif                                                    ' Trace

Return
'-- End Dcf77_sectic ----------------------------------------------------------
```

```
'-------------------------------------------------------------------------------
' Interrupt-Service-Routine (INT5): Dimmer_null_irq
' Routine zur Nulldurchgangserkennung des Sinussignals
'-------------------------------------------------------------------------------
Dimmer_null_irq:

   ' In dieser IRQ-Routine wird wegen Geschwindigkeit auf Ausgaben verzichtet
   If Free_running_flag = Aktiv Then
      Start Timer2
      Timer2 = Triac_zuenszeitpunkte(dim_helligkeitsstufe)
   End If                                                  ' Dimmer aktiv?

Return
'-- End Dimmer_null_irq -------------------------------------------------------


'-------------------------------------------------------------------------------
' Interrupt-Service-Routine (Timer0): Dimmer_triac_irq
' Routine zum Zünden des TRIAC
'-------------------------------------------------------------------------------
Dimmer_triac_irq:

   Stop Timer2
   Triac_start = 1
   Waitus 20
   Triac_start = 0

Return
'-- End Dimmer_triac_irq -------------------------------------------------------


'-------------------------------------------------------------------------------
' Interrupt-Service-Routine (INT6): Wind_irq
' Routine zur Auswertung des INT6 Interrupts für Impulseingang Windsensor
'-------------------------------------------------------------------------------
Wind_irq:

   ' Impulszähler erhöhen
   Incr Wind_sensorimpulse

Return
'-- End Timer_irq -------------------------------------------------------------


'*******************************************************************************
' Subroutinen
'*******************************************************************************

' ********************
' * LCD:  Display    *
' ********************

'-------------------------------------------------------------------------------
' LCD - Subroutine: Lcd_set_helligkeit
' Subroutine setzt den als Übergabeparameter übergebenen Bytewert als
' LCD Helligkeitswert
'
' Parameter:    Value = Bytewert für LCD Helligkeit
' Rückgabewert: keine
'-------------------------------------------------------------------------------
Sub Lcd_set_helligkeit(byval Value As Byte)

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Lcd_testmodus                                       ' Trace
      Print #2 , "Enter Lcd_set_helligkeit"
   #endif                                                  ' Trace

   ' PWM-Wert setzen und Variable in Timer-Count-Register laden
   Lcd_pwmwert = Value
   Pwm3a = Lcd_pwmwert

End Sub
'-- End Lcd_set_helligkeit ------------------------------------------------------


'-------------------------------------------------------------------------------
' LCD - Sub-Routine: Lcd_writesingleline
' Gibt einen einzeiligen Texte auf dem LCD-Display aus.
' Parameter: line1 = Zeile in der die Ausgabe erfolgen soll
'            text1 = Text der ausgegeben werden soll
```

```
'-------------------------------------------------------------------------------
Sub Lcd_writesingleline(line1 As Byte , Text1 As Sring)

   Cls                                                  ' Bildschirm löschen
   Locate Line1 , 1 : Lcd Text1                         ' Ausgabe Zeil 1

End Sub
'-- End Lcd_writesingleline -------------------------------------------------


'-------------------------------------------------------------------------------
' LCD - Sub-Routine: Lcd_writedoubleline
' Gibt einen zweizeiligen Texte auf dem LCD-Display aus.
' Die Reihenfolge und die Textposition kann frei gewählt werden
' Parameter: line1 = 1. Zeile in der die 1. Ausgabe erfolgen soll
'            line2 = 2. Zeile in der die 2. Ausgabe erfolgen soll
'            text1 = 1. Text der auf 1. Position ausgegeben werden soll
'            text2 = 2. Text der auf 2. Position ausgegeben werden soll
'-------------------------------------------------------------------------------
Sub Lcd_writedoubleline(line1 As Byte , Line2 As Byte , Text1 As String , Text2 As String)

   Cls                                                  ' Bildschirm löschen
   Locate Line1 , 1 : Lcd Text1                         ' Ausgabe Zeil 1
   Locate Line2 , 1 : Lcd Text2                         ' Ausgabe Zeile 2

End Sub
'-- End Lcd_writedoubleline -------------------------------------------------


'-------------------------------------------------------------------------------
' LCD - Sub-Routine: Lcd_writetrippleline
' Gibt einen dreizeiligen Texte auf dem LCD-Display aus.
' Parameter: text1 = 1. Text der in Zeile 1 ausgegeben werden soll
'            text2 = 2. Text der in Zeile 2 ausgegeben werden soll
'            text3 = 3. Text der in Teile 3 ausgegeben werden soll
'-------------------------------------------------------------------------------
Sub Lcd_writetrippleline(text1 As String , Text2 As String , Text3 As String)

   ' Variablen für Positionierung definieren und festlegen
   ' Hinweis: Interessanter Weise funktioniert Ausgabe nicht bei direkter Zuweisung!
   Dim Line1 As Byte , Line2 As Byte , Line3 As Byte
   Line1 = 1
   Line2 = 2
   Line3 = 3

   ' Ausgabe der Texte
   Cls                                                  ' Bildschirm löschen
   Locate Line1 , 1 : Lcd Text1                         ' Ausgabe Zeile 1
   Locate Line2 , 1 : Lcd Text2                         ' Ausgabe Zeile 2
   Locate Line3 , 1 : Lcd Text3                         ' Ausgabe Zeile 3

End Sub
'-- End Lcd_writetrippleline ------------------------------------------------


' ********************
' * FAN:  Lüfter     *
' ********************

'-------------------------------------------------------------------------------
' FAN - Subroutine: Fan_set_geschwindigkeit
' Subroutine setzt den als Übergabeparameter übergebenen Bytewert als Ge-
' schwindigkeitswert für die Lüftergeschwindigkeit.
' Kleinere Bytewerte als die Minimalgeschwindigkeit werden auf MIN begrenzt.
'
' Parameter:    Value = Bytewert für Lüftergeschwindigkeit
' Rückgabewert: keine
'-------------------------------------------------------------------------------
Sub Fan_set_geschwindigkeit(byval Value As Byte)

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Fan_testmodus                                    ' Trace
      Print #2 , "Enter Fan_set_geschwindigkeit"
   #endif                                               ' Trace

   ' Prüfung ob Bytewert zu klein, wenn ja auf MIN begrenzen
   ' If Value <= Fan_geschwindigkeit_min Then
   '   Value = Fan_geschwindigkeit_min
   ' End If
   ' Anmerkung: Prüfung wird in Praxis nicht gemachtda auch abgeschaltet werden
```

```
   ' soll.

   ' PWM-Wert setzen und Variable in Timer-Count-Register laden
   Fan_pwmwert = Value
   Pwm3b = Fan_pwmwert

End Sub
'-- End Fan_set_geschwindigkeit ---------------------------------------------


' ********************
' * SOUND: Piezo     *
' ********************

'-------------------------------------------------------------------------------
' SOUND - Subroutine: Sound_beep
' Subroutine gibt für die in ms übergebene Zeit einen Signalton aus
' Achtung, Routine wartet so lange bis Ton ausgegeben ist!
'
' Parameter:    time_value = Zeit in ms die der Beepton ausgegeben werden soll
' Rückgabewert: keine
'-------------------------------------------------------------------------------
Sub Sound_beep(time_value As Word)

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Sound_testmodus = 1                                  ' Trace
      Print #2 , "Enter Sound_beep mit Wartezeit " ; Time_value ; " ms"
   #endif                                                   ' Trace

   Gosub Sound_an                                           ' Ton einschalten
   Waitms Time_value                                        ' warten
   Gosub Sound_aus                                          ' Ton ausschalten

End Sub
'-- End Sound_beep ----------------------------------------------------------


' ********************
' * SHT71:  Sensor   *
' ********************

'-------------------------------------------------------------------------------
' SHT71 - Subroutine: Sht71_calc_measurements
' Subroutine kalkuliert aus den Rohdaten in den globalen Variablen die tat-
' sächlichen Messwerte und legt diese erneut in globalen Enddaten-Variablen ab.
'
' Parameter:    keine
' Rückgabewert: keine
'-------------------------------------------------------------------------------
Sub Sht71_calc_measurements()

   ' ----- locale Variablen -----
   Local Local_resultstring As String * 20

   Local Local_temp_single1 As Single
   Local Local_temp_single2 As Single
   Local Local_temp_single3 As Single

   ' ----- Programmcode -----

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Sht71_testmodus = 1                                  ' Trace
      Print #2 , "Enter Sht71_calc_measurements"
   #endif                                                   ' Trace


   ' ----- Berechnung der Temperatur T -----

   ' Formula temperature:
   ' Temperature = d1 + d2 * SOt
   Local_temp_single1 = Sht71_d2 * Sht71_temp_word
   Local_temp_single1 = Local_temp_single1 + Sht71_d1

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Sht71_testmodus = 1                                  ' Trace
      Local_resultstring = Fusing(local_temp_single1 , "###.##")
      Print #2 , "Data " ; Hex(sht71_temp_word) ; " " ; Sht71_temp_word
      Print #2 , "Temp " ; Local_resultstring ; "C"
   #endif                                                   ' Trace
```

```
   ' gerechneter Temperaturwert an globale Variable übergeben
   Sht71_t = Local_temp_single1


   ' ----- Berechnung der Feuchtigkeit RHlinear -----

   ' Formula relative humidity:
   ' RHlinear = c1 + c2 * SOrh + c3 * SOrh^2

   Local_temp_single1 = Sht71_c2 * Sht71_humi_word

   Local_temp_single2 = Sht71_humi_word * Sht71_humi_word
   Local_temp_single2 = Sht71_c3 * Local_temp_single2

   Local_temp_single3 = Sht71_c1 + Local_temp_single1
   Local_temp_single3 = Local_temp_single3 + Local_temp_single2

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Sht71_testmodus = 1                                    ' Trace
      Local_resultstring = Fusing(local_temp_single3 , "###.##")
      Print #2 , "Data " ; Hex(sht71_humi_word) ; " " ; Sht71_humi_word
      Print #2 , "RHlinear " ; Local_resultstring ; "%"
   #endif                                                     ' Trace

   ' gerechneter Feuchtigkeitswert RHlinear an globale Variable übergeben
   Sht71_rhlinear = Local_temp_single3


   ' ----- Berechnung der Feuchtigkeit RHtrue -----

   ' Formula humidity / temperature compensation:
   ' RHtrue = (Temperature - 25) * (t1 + t2 * SOrh) + RHlinear
   Local_temp_single1 = Sht71_t - 25

   Local_temp_single2 = Sht71_t2 * Sht71_humi_word
   Local_temp_single2 = Local_temp_single2 + Sht71_t1

   Local_temp_single3 = Local_temp_single1 * Local_temp_single2
   Local_temp_single3 = Local_temp_single3 + Sht71_rhlinear

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Sht71_testmodus = 1                                    ' Trace
      Local_resultstring = Fusing(local_temp_single3 , "###.##")
      Print #2 , "RHtrue " ; Local_resultstring ; "%"
   #endif                                                     ' Trace

   ' gerechneter Feuchtigkeitswert RHtrue an globale Variable übergeben
   Sht71_rhtrue = Local_temp_single3


   ' ----- Berechnung des Taupunktes -----

   ' Formulas for Dew Point calculation:
   ' H = (log10(RHtrue) - 2) / alpha + (m * Temperature) / (tn + Temperature)
   ' Dp = tn * H / (m - H) // this is the dew point in Celsius
   Local_temp_single1 = Log10(sht71_rhtrue)
   Local_temp_single1 = Local_temp_single1 - 2
   Local_temp_single1 = Local_temp_single1 / Sht71_alpha

   Local_temp_single2 = Sht71_m * Sht71_t
   Local_temp_single3 = Sht71_tn + Sht71_t
   Local_temp_single2 = Local_temp_single2 / Local_temp_single3

   Local_temp_single3 = Local_temp_single1 + Local_temp_single2

   Local_temp_single1 = Sht71_tn * Local_temp_single3
   Local_temp_single2 = Sht71_m - Local_temp_single3

   Local_temp_single3 = Local_temp_single1 / Local_temp_single2

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Sht71_testmodus = 1                                    ' Trace
      Local_resultstring = Fusing(local_temp_single3 , "###.##")
      Print #2 , "Dewpoint " ; Local_resultstring ; "C"
   #endif                                                     ' Trace

   ' gerechneter Taupunkt an globale Variable übergeben
   Sht71_dewpoint = Local_temp_single3

End Sub
```

```
'-- End Sht71_calc_measurements ------------------------------------------------


' **********************
' * EE:  EEPROM extern *
' **********************

'-------------------------------------------------------------------------------
' EE - Subroutine: Ee_write_eeprom_byte
' Subroutine um ein einzelnes Byte an eine bestimmte Speicheradresse in das
' EEPROM zu schreiben.
'
' Parameter:   Adress = Adresse (Word) an die geschrieben werden soll
'              Value  = Bytevariable die geschrieben werden soll
'-------------------------------------------------------------------------------
Sub Ee_write_eeprom_byte(byval Adress As Word , Byval Value As Byte)

   ' ----- Lokale Variable definieren -----
   Local Ee_adrl As Byte
   Local Ee_adrh As Byte

   ' ----- Programmcode -----

   ' Adresse in High-Byte und Low-Byte zerlegen
   Ee_adrl = Low(adress)
   Ee_adrh = High(adress)

   ' Kommunikation über I2C-Bus
   I2cstart                                                ' START-Sequenze senden
   I2cwbyte Address_write                                  ' Daten schreiben initiieren
   I2cwbyte Ee_adrh                                        ' High-Byte der Adresse übertragen
   I2cwbyte Ee_adrl                                        ' Low-Byte der Adresse übertragen
   I2cwbyte Value                                          ' Byte-Variable übertragen
   I2cstop                                                 ' STOP-Sequenz

End Sub
'-- End Ee_write_eeprom_byte -------------------------------------------------


'-------------------------------------------------------------------------------
' EE - Subroutine: Ee_write_eeprom_word
' Subroutine um ein zwei Bytes (Word) an eine bestimmte Speicheradresse in das
' EEPROM zu schreiben.
'
' Parameter:   Adress = Adresse (Word) an die geschrieben werden soll
'              Value  = Wordvariable die geschrieben werden soll
'-------------------------------------------------------------------------------
Sub Ee_write_eeprom_word(byval Adress As Word , Byval Value As Word)

   ' ----- Lokale Variable definieren -----
   Local Ee_adrl As Byte
   Local Ee_adrh As Byte

   Local Ee_datal As Byte
   Local Ee_datah As Byte

   ' ----- Programmcode -----

   ' Adresse in High-Byte und Low-Byte zerlegen
   Ee_adrl = Low(adress)
   Ee_adrh = High(adress)

   ' Word in einzelne Bytes zerlegen
   Ee_datal = Low(value)
   Ee_datah = High(value)

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Eeprom_testmodus                                    ' Trace
      Print #2 , "Data high: " ; Hex(ee_datah)
      Print #2 , "Data low:  " ; Hex(ee_datal)
   #endif                                                  ' Trace

   ' Kommunikation über I2C-Bus
   I2cstart                                                ' START-Sequenze senden
   I2cwbyte Address_write                                  ' Daten schreiben initiieren
   I2cwbyte Ee_adrh                                        ' High-Byte der Adresse übertragen
   I2cwbyte Ee_adrl                                        ' Low-Byte der Adresse übertragen
   I2cwbyte Ee_datah                                       ' High-Byte aus Variable übertragen
   I2cwbyte Ee_datal                                       ' Low-byte aus Variable übertragen
   I2cstop                                                 ' STOP-Sequenz
```

```
End Sub
'-- End Ee_write_eeprom_word ------------------------------------------------


'--------------------------------------------------------------------------------
' EE - Subroutine: Ee_write_eeprom_loggingstruct
' Subroutine schreibt die 8 Byte Loggingdaten an die übergebene Adresse
'
' Parameter: Adress        = Adresse (Word) ab der die Daten geschrieben werden
'--------------------------------------------------------------------------------
Sub Ee_write_eeprom_struct(byval Adress As Word)

   ' ----- Lokale Variable definieren -----
   Local Ee_adrl As Byte
   Local Ee_adrh As Byte

   ' ----- Programmcode -----

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Eeprom_testmodus                                    ' Trace
      Print #2 , "EE-write Adresse " ; Hex(adress)
   #endif                                                  ' Trace

   ' Adresse in High-Byte und Low-Byte zerlegen
   Ee_adrl = Low(adress)
   Ee_adrh = High(adress)

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Eeprom_testmodus                                    ' Trace
      Print #2 , "EE-write low  Adresse " ; Hex(ee_adrl)
      Print #2 , "EE-write high Adresse " ; Hex(ee_adrh)
      Print #2 , "Daten-Struct : " ;
      Print #2 , Hex(ee_data_logging_struct(1)) ; " " ;
      Print #2 , Hex(ee_data_logging_struct(2)) ; " " ;
      Print #2 , Hex(ee_data_logging_struct(3)) ; " " ;
      Print #2 , Hex(ee_data_logging_struct(4)) ; " " ;
      Print #2 , Hex(ee_data_logging_struct(5)) ; " " ;
      Print #2 , Hex(ee_data_logging_struct(6)) ; " " ;
      Print #2 , Hex(ee_data_logging_struct(7)) ; " " ;
      Print #2 , Hex(ee_data_logging_struct(8))
   #endif                                                  ' Trace

   ' Kommunikation über I2C-Bus
   I2cstart                                                ' START-Sequenze senden
   I2cwbyte Address_write                                  ' Daten schreiben initiieren
   I2cwbyte Ee_adrh                                        ' High-Byte der Adresse übertragen
   I2cwbyte Ee_adrl                                        ' Low-Byte der Adresse übertragen
   I2cwbyte Ee_data_logging_struct(1)                      ' Byte 1 übertragen
   I2cwbyte Ee_data_logging_struct(2)                      ' Byte 2 übertragen
   I2cwbyte Ee_data_logging_struct(3)                      ' Byte 3 übertragen
   I2cwbyte Ee_data_logging_struct(4)                      ' Byte 4 übertragen
   I2cwbyte Ee_data_logging_struct(5)                      ' Byte 5 übertragen
   I2cwbyte Ee_data_logging_struct(6)                      ' Byte 6 übertragen
   I2cwbyte Ee_data_logging_struct(7)                      ' Byte 7 übertragen
   I2cwbyte Ee_data_logging_struct(8)                      ' Byte 8 übertragen
   I2cstop                                                 ' STOP-Sequenz

End Sub
'-- End Ee_write_eeprom_loggingstruct ----------------------------------------


'--------------------------------------------------------------------------------
' EE - Subroutine: Ee_read_eeprom_loggingstruct
' Subroutine liest die 8 Byte Loggingdaten von der übergebene Adresse
'
' Parameter: Adress        = Adresse (Word) von der die Daten gelesen werden
'--------------------------------------------------------------------------------
Sub Ee_read_eeprom_struct(byval Adress As Word)

   ' ----- Lokale Variable definieren -----
   Local Ee_adrl As Byte
   Local Ee_adrh As Byte

   ' ----- Programmcode -----
   ' Adresse in High-Byte und Low-Byte zerlegen
   Ee_adrl = Low(adress)
   Ee_adrh = High(adress)

   ' Kommunikation über I2C-Bus
```

```
   I2cstart                                                ' START-Sequenz senden
   I2cwbyte Address_write                                  ' Daten schreiben initiieren
   I2cwbyte Ee_adrh                                        ' High-Byte der Adresse übertragen
   I2cwbyte Ee_adrl                                        ' Low-Byte der Adresse übertragen

   I2cstart                                                ' START-Sequenz erneut senden
   I2cwbyte Address_read                                   ' Daten lesen initiieren
   I2crbyte Ee_data_logging_struct(1) , Ack                ' Byte 1 lesen
   I2crbyte Ee_data_logging_struct(2) , Ack                ' Byte 2 lesen
   I2crbyte Ee_data_logging_struct(3) , Ack                ' Byte 3 lesen
   I2crbyte Ee_data_logging_struct(4) , Ack                ' Byte 4 lesen
   I2crbyte Ee_data_logging_struct(5) , Ack                ' Byte 5 lesen
   I2crbyte Ee_data_logging_struct(6) , Ack                ' Byte 6 lesen
   I2crbyte Ee_data_logging_struct(7) , Ack                ' Byte 7 lesen
   I2crbyte Ee_data_logging_struct(8) , Nack               ' Byte 8 lesen
   I2cstop                                                 ' STOP-Sequenz

End Sub
'-- End Ee_write_eeprom_loggingstruct ------------------------------------



'*****************************************************************************
' Funktionen
'*****************************************************************************

' ********************
' * KEY:  Tastatur   *
' ********************

'-----------------------------------------------------------------------------
' KEY - Funktion: Key_query_matrix()
' Funktion zur Abfrage der einzelnen Tasten innerhalb der Matrix.
' Die Abfrage erfolgt über zwei Schleifenkonstrukte welche die einzelnen Spal-
' ten und darin die einzelnen Zeilen einer Spalte auf eine gedrückte Taste
' absucht.
'
' Parameter:    keine
' Rückgabewert: Bytewert = Summe aller erkannten Tastencodes
'                          Tastencode ist wie folgt aufgebaut (Ty.x):
'                          T1.1 = 11     T1.2 = 12      T1.3 = 14
'                          T2.1 = 21     T2.2 = 22      T2.3 = 24
'                          T3.1 = 41     T3.2 = 42      T3.3 = 44
'-----------------------------------------------------------------------------
Function Key_query_matrix() As Byte

   ' Ggf. Trace Output schreiben
   #if Key_testmodus                                       ' Trace
      Print #2 , "Enter Key_query_matrix"
   #endif                                                  ' Trace

   ' Taste NIX vorbelegen
   Key_taste = 0

   ' Spaltenabfragen (Spalte für Spalte durchklappern)
   For Key_spalte = 0 To 2

      ' ----- Einzelnes "abklappern" der Spalten -----
      ' Spalte 1 auf LOW und Spaltenindex für Tastencoderechung setzen
      If Key_spalte = 0 Then
         Reset Key_spalte_1_port
         Set Key_spalte_2_port
         Set Key_spalte_3_port
         Key_spalten_index = 1
      End If                                               ' Spalte 0 geprüft

      ' Spalte 2 auf LOW und Spaltenindex für Tastencoderechung setzen
      If Key_spalte = 1 Then
         Set Key_spalte_1_port
         Reset Key_spalte_2_port
         Set Key_spalte_3_port
         Key_spalten_index = 2
      End If                                               ' Spalte 1 geprüft

      ' Spalte 3 auf LOW und Spaltenindex für Tastencoderechung setzen
      If Key_spalte = 2 Then
         Set Key_spalte_1_port
         Set Key_spalte_2_port
         Reset Key_spalte_3_port
         Key_spalten_index = 4
```

```
      End If                                                ' Spalte 2 geprüft

     ' Zeilenabfragen (Zeile für Zeile durchklappern)
     ' In der Ursprünglichen Version wurde die Abfrage der Zeilen über den
     ' BASCOM-Befehl debounce realisiert. Das Problem hier ist, das Debounce
     ' wartet bis eine "Taste=Zeile" wieder losgelassen wurde. Damit ist die
     ' Abfrage von mehreren Tasten einer gleichen Zeile nicht möglich!
     ' Die Entprellung der Tasten erfolgt durch die Zeitkonstante Key_waiting
     For Key_zeile = 1 To 3
       Select Case Key_zeile

         ' Prüfung für Zeile 1 vornehmen und ggf. Tastencodes rechen
         Case 1 :
            ' Debounce Key_zeile_1_pin , 0 , Key_calc_taste , Sub
            If Key_zeile_1_pin = 0 Then
               Temp_byte_1 = Key_spalten_index + 10
               Key_taste = Key_taste + Temp_byte_1
            End If                                          ' Zeile 1 geprüft

         ' Prüfung für Zeile 2 vornehmen und ggf. Tastencodes rechen
         Case 2 :
            ' Debounce Key_zeile_2_pin , 0 , Key_calc_taste , Sub
            If Key_zeile_2_pin = 0 Then
               Temp_byte_1 = Key_spalten_index + 20
               Key_taste = Key_taste + Temp_byte_1
            End If                                          ' Zeile 2 geprüft

         ' Prüfung für Zeile 3 vornehmen und ggf. Tastencodes rechen
         Case 3 :
            ' Debounce Key_zeile_3_pin , 0 , Key_calc_taste , Sub
            If Key_zeile_3_pin = 0 Then
               Temp_byte_1 = Key_spalten_index + 40
               Key_taste = Key_taste + Temp_byte_1
            End If                                          ' Zeile 3 geprüft

       End Select                                           ' Zeilenprüfung
     Next Key_zeile                                         ' Zeile
   Next Key_spalte                                          ' Spalte

   ' Urspungszustand wie bei Init wieder herstellen damit Interruptsteuerung
   ' wieder möglich wird.
   Reset Key_spalte_1_port
   Reset Key_spalte_2_port
   Reset Key_spalte_3_port

   ' Tastencode an Rückgabewert übergeben
   Key_query_matrix = Key_taste

End Function
'-- End Key_Query_matrix ------------------------------------------------------


' ********************
' * RAIN: Regensensor *
' ********************

'-------------------------------------------------------------------------------
' RAIN - Function: Rain_measurement
' Funktion liest den Zustand des PINs zurück an dem der Regensensor ange-
' schlossen ist und liefert den Wert als Rückgabewert
'
' Parameter:    keiner
' Rückgabewert: Bytewert &H00 = kein Regen
'                        &H01 = Regen vom Sensor erkannt
'-------------------------------------------------------------------------------
Function Rain_measurement() As Byte

   ' ----- lokale Variablen -----
   Local Temp_value As Byte

   ' ----- Programmcode -----

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Rain_testmodus                                   ' Trace
      Print #2 , "Enter Rain_measurement"
      ' Direkte Auswertung des PINs für Trace
      If Rain_sensor = 0 Then
         Print #2 , "PinG.3 - NO"
      Else
         Print #2 , "PinG.3 - YES"
```

```
      End If
   #endif                                                ' Trace

   ' Status Regensensor von Pin einlesen
   Temp_value = Rain_sensor

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Rain_testmodus                                    ' Trace
      Print #2 , "Zustand PIN Regensensor = " ; Temp_value
   #endif                                                ' Trace

   ' temporäre Variable in Rückgabewert übergeben
   Rain_measurement = Temp_value

End Function
'-- End Rain_measurement ------------------------------------------------------


' ********************
' * SHT71:  Sensor   *
' ********************

'-------------------------------------------------------------------------------
' SHT71 - Function: Sht71_write_byte
' Funktion schreibt ein Byte auf den Sensibus und überpruft das Acknowledge
'
' Parameter:    Value = Byte welches zum SHT71 gesendet werden soll
' Rückgabewert: Rückgabewert error = 1 im Fall das kein Acknowledge gesendet
'-------------------------------------------------------------------------------
Function Sht71_write_byte(byval Value As Byte) As Byte

   ' ----- locale Variable -----
   Local Error As Byte

   ' ----- Programmcode -----

   ' Sicherstellen das Pin DATA auf Output
   Config Sht71_datain = Output

   ' Datenbyte senden
   ' MSB shifted out first when clock goes high
   Shiftout Sht71_dataout , Sht71_sck , Value , 1

   ' Release DATA-line
   Set Sht71_dataout

   ' Data auf Eingabe umschalten
   Config Sht71_datain = Input

   ' clk #9 für ACK absenden und Datain PIN auswerten
   Set Sht71_sck
   Error = Sht71_datain
   Reset Sht71_sck

   ' Rückgabewerte error = 1 wenn kein Acknowledge empfangen
   Sht71_write_byte = Error                              ' Rückgabewert zuweisen

End Function
'-- End Sht71_write_byte ------------------------------------------------------


'-------------------------------------------------------------------------------
' SHT71 - Function: Sht71_read_byte
' Funktion liest ein Byte vom Sensibus und sende ein Acknowledge in Form von
' "ack=1"
'
' Parameter:    Acknowledge = Variable die als Acknowledge gesendet wird
' Rückgabewert: Bytewert des vom Sensibus gelesenen Bytes
'-------------------------------------------------------------------------------
Function Sht71_read_byte(byval Acknowledge As Byte) As Byte

   ' ----- locale Variable -----
   Local Databyte As Byte

   ' ----- Programmcode -----

   ' release DATA-line
   Config Sht71_datain = Output
   Set Sht71_dataout
   Config Sht71_datain = Input
```

```
   ' Datenbyte einlesen
   ' MSB shifted in first when clock goes high
   Shiftin Sht71_datain , Sht71_sck , Databyte , 1

   ' Acknowledge senden
   Config Sht71_datain = Output

   If Acknowledge = Sht71_acknowledge Then
      ' Botschaft mit ACK quittieren (Pegel = low)
      Reset Sht71_dataout
   Else                                                    ' I2C-Acknowledge?
      ' Botschaft mit noACK quittieren (Pegel = high)
      Set Sht71_dataout
   End If                                                  ' I2C-Acknowledge?

   ' clk #9 für ACK absenden
   Set Sht71_sck
   $asm : Nop : Nop : Nop : $end Asm
   Reset Sht71_sck

   ' Release DATA-line
   Set Sht71_dataout

   ' Rückgabewert Databyte zuweisen
   Sht71_read_byte = Databyte

End Function
'-- End Sht71_read_byte ---------------------------------------------------


'-------------------------------------------------------------------------------
' SHT71 - Function: Sht71_softreset
' Funktion resetet den sensor durch einen softreset
'
' Parameter:    keiner
' Rückgabewert: Rückgabewert error = 1 falls Sensor nicht antwortet
'-------------------------------------------------------------------------------
Function Sht71_softreset() As Byte

   ' ----- locale Variable -----
   Local Error As Byte

   ' ----- Programmcode -----

   ' Reset Communication
   Gosub Sht71_connection_reset_sequence

   ' Send RESET-Command an den Sensor
   Error = Sht71_write_byte(sht71_reset)

   ' Rückgabewerte zuweisen
   Sht71_softreset = Error

End Function
'-- End Sht71_softreset ---------------------------------------------------


'-------------------------------------------------------------------------------
' SHT71 - Function: Sht71_read_statusregister
' Funktion liest das Statusregister des Sht71 aus (inkl. 8 Bit Checksumme)
'
' Parameter:    P_Value    = Inhalt des Statusregister wird über
'                            "call by reference" zurückgegeben
'               P_Checksum = Cheksummenbyte wird über "call by reference"
'                            zurückgegeben
' Rückgabewert: Rückgabewert error = 1 falls Sensor nicht antwortet
'-------------------------------------------------------------------------------
Function Sht71_read_statusregister(byref P_value As Byte , Byref P_checksum As Byte) As Byte

   ' ----- locale Variable -----
   Local Error As Byte

   ' ----- Programmcode -----

   ' Transmission start
   Gosub Sht71_transmission_start

   ' Kommando für Statusregister auslesen senden
   Error = Sht71_write_byte(sht71_status_reg_read)
```

```
   P_value = Sht71_read_byte(sht71_acknowledge)             ' Read Statusregister (8-bit)
   P_checksum = Sht71_read_byte(sht71_noacknowledge)        ' Read Checksum (8-bit)

   ' Rückgabewert zuweisen
   Sht71_read_statusregister = Error

End Function
'-- End Sht71_read_statusregister ------------------------------------------


'--------------------------------------------------------------------------
' SHT71 - Function: Sht71_write_statusregister
' Funktion schreibt in das Statusregister des Sht71 den übergebenen Bytewert
'
' Parameter:    P_Value    = Bytewert der in das Statusregister geschrieben
'                            werden soll. Achtung: "call by reference"
' Rückgabewert: Rückgabewert error = 1 falls Sensor nicht antwortet
'--------------------------------------------------------------------------
Function Sht71_write_statusregister(byref P_value As Byte) As Byte

   ' ----- locale Variable -----
   Local Error As Byte

   ' ----- Programmcode -----

   ' Transmission start
   Gosub Sht71_transmission_start

   ' Kommando für Statusregister auslesen senden
    Error = Sht71_write_byte(sht71_status_reg_write)

   ' Datenbyte senden
   Sht71_temp_byte1 = Sht71_write_byte(p_value)
   Error = Error + Sht71_temp_byte1

   ' Rückgabewert zuweisen
   Sht71_write_statusregister = Error

End Function
'-- End Sht71_write_statusregister -----------------------------------------


'--------------------------------------------------------------------------
' SHT71 - Function: Sht71_measure
' Funktion liest je nach Modus Temperature- oder Feuchtigkeitsdaten aus dem
' Sensor aus und übergibt die Messwerte an die jeweiligen Funktionen
'
' Parameter:    P_Value    = Word für 14 Bit Temperatur oder 12 Bit Feuchtigkeit
'                            Achtung: "call by reference"
'               P_checksum = Cheksummenbyte wird über "call by reference"
'                            zurück
'               Modus      = Parameter ob Temperatur oder Feuchtigkeit
' Rückgabewert: Rückgabewert error = 1 falls Sensor nicht antwortet
'--------------------------------------------------------------------------
Function Sht71_measure(byref P_value As Word , Byref P_checksum As Byte , Byval Modus As Byte) As
Byte

   ' ----- locale Variable -----
   Local Error As Byte
   Local Local_temp_byte1 As Byte
   Local Local_temp_byte2 As Byte
   Local Local_temp_word As Word
   Local Local_temp_long1 As Long

   ' ----- Programmcode -----

   ' Transmission start
   Gosub Sht71_transmission_start

   Select Case Modus
      Case Sht71_modus_humi : Error = Sht71_write_byte(sht71_measure_humi)
      Case Sht71_modus_temp : Error = Sht71_write_byte(sht71_measure_temp)
   End Select

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Sht71_testmodus = 1                                  ' Trace
      Print #2 , "Fehler write command : " ; Error
   #endif                                                   ' Trace

   ' Umschalten auf Eingabe
```

```
   Config Sht71_datain = Input

   ' So und nun warten bis PIN zappelt
   'Bitwait Sht71_datain , Reset                              ' Problem, Routine wartet ggf. un-
endlich!

   ' Umsetzung hier über FOR-Schleife mit Timeout
   Local_temp_byte1 = False
   For Local_temp_long1 = 0 To Sht71_response_wait
      If Sht71_datain = 0 Then
         ' Reaktion des SHT71 erkannt, FOR abbrechen und weiter
         Local_temp_byte1 = True
         Exit For
      End If                                                 ' Noch Warten
   Next Local_temp_long1                                     ' Warteschleife

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Sht71_testmodus = 1                                   ' Trace
      Print #2 , "Biterkennung = " ; Local_temp_byte1
   #endif                                                    ' Trace

   ' Wenn PIN zappelt weiter mit Einlesen ansonsten Fehler setzen und fertig
   If Local_temp_byte1 = True Then

      Local_temp_byte1 = Sht71_read_byte(sht71_acknowledge) ' read the first byte (MSB)
      Local_temp_byte2 = Sht71_read_byte(sht71_acknowledge) ' read the second byte (LBS)
      P_checksum = Sht71_read_byte(sht71_noacknowledge)     ' read checksum
      Local_temp_word = Local_temp_byte1
      Shift Local_temp_word , Left , 8
      Local_temp_word = Local_temp_word And &HFF00
      Local_temp_word = Local_temp_word + Local_temp_byte2
      P_value = Local_temp_word

   Else                                                      ' Einlesen
      Incr Error
   End If                                                    ' Einlesen

   ' Rückgabewert zuweisen
   Sht71_measure = Error

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Sht71_testmodus = 1                                   ' Trace
      Print #2 , "Endergebnis : " ; Error
   #endif                                                    ' Trace

End Function
'-- End Sht71_measure ------------------------------------------------------


'--------------------------------------------------------------------------------
' SHT71 - Function: Sht71_crc_check
' Funktion prüft mit dem Übergebenen Daten-Word und der Checksumme ob CRC für
' Datenübertragung korrekt oder fehlerbehaftet.
' Im Fehlerfall wird empfohlen einen Reset auf den Sensor durchzuführen.
'
' Parameter:    Value    = Word für 14 Bit Temperatur oder 12 Bit Feuchtigkeit
'               Crc      = Cheksummenbyte
' Rückgabewert: Bytewert Crc_ok oder Crc_nok
'--------------------------------------------------------------------------------
Function Sht71_crc_check(byval Value As Word , Byval Crc As Byte , Byval Modus As Byte) As Byte

   ' ----- Locale Variablen -----
   Local Crc_reversed As Byte
   Local Local_temp_byte1 As Byte
   Local Local_temp_byte2 As Byte
   Local Local_temp_byte3 As Byte
   Local Crc_register As Byte
   Local Table_index As Byte

   ' ----- Programmcode -----
   ' Vorgaben zur CRC-Rechung aus Application Note CRC Sensirion
   '
   ' Perform the following operations:
   ' 1. Initialize the CRC register with the value of the lower nibble of the
   '    value of the status register (reversed (s0s1s2s3‘0000)).
   '    (default ‘00000000’ = 0)
   ' 2. XOR each (transmitted and received) byte with the previous CRC value.
   '    The result is the new byte that you need to calculate the CRC value from.
   ' 3. Use this value as the index to the table to obtain the new CRC value.
   ' 4. Repeat from 2.) until you have passed all bytes through the process.
```

```
' 5. The last byte retrieved from the table is the final CRC value.
' 6. The CRC value retrieved from the SHTxx must be reversed
'    (bit 0 = bit 7, bit 1=bit 6..bit 7 = bit 0) and can then be compared to
'    the final CRC value.(2)

' Wenn Konstante gesetzt Trace Output schreiben
#if Sht71_testmodus = 1                                 ' Trace
   Print #2 , "Enter CRC-Pruefung"
#endif                                                  ' Trace

' Zur CRC-Prüfung muss das vom SHT71 empfangene CRC-Byte "gedreht" werden.
' bit 0 = bit 7, bit 1 = bit 6 ... bit 7 = bit 0
Local_temp_byte1 = 0
Local_temp_byte2 = 7
Do
   Crc_reversed.local_temp_byte1 = Crc.local_temp_byte2
   Incr Local_temp_byte1
   Decr Local_temp_byte2
Loop Until Local_temp_byte1 = 8                         ' Byte spiegeln

' Ab jetzt gilt zur Berechung das "gedrehte CRC-Byte Crc_reversed

' Initialisierung des Crc_registers mit dem lower nibble des Status-Registers
' hier werden Konstanten angenommen da Default = &B00000000 = 0 ist.
' (reversed(s0 s1 s2 s3 0 0 0 0))
Crc_register = 0

' Jedes gesendete und empfangene Byte muss nun verarbeitet werden.
' Hierzu wird Datenstruktur aufgebaut

Select Case Modus
   Case Sht71_modus_humi : Sht71_transceive_struct(1) = 5
   Case Sht71_modus_temp : Sht71_transceive_struct(1) = 3
End Select                                              ' Modusauswahl

Sht71_transceive_struct(2) = High(value)
Sht71_transceive_struct(3) = Low(value)

' Nun kann CRC gerechnet werden
For Local_temp_byte1 = 1 To 3

   ' Lese Byte aus Datenstruktur für Tabellenzugriff aus
   Local_temp_byte2 = Sht71_transceive_struct(local_temp_byte1)

   ' Setzte den Zeiger auf den Anfang des DATA-Statement Sht71_Crc_table
   Restore Sht71_crc_table

   ' Zugriff und Berechnung
   Table_index = Crc_register Xor Local_temp_byte2
   If Local_temp_byte2 = 0 Then
      Table_index = 3
   End If                                               ' Auswahl Tabellenindex

   For Local_temp_byte3 = 0 To Table_index
      Read Crc_register
   Next Local_temp_byte3                                ' Datenwerte lesen

Next Local_temp_byte1                                   ' CRC-Wert rechnen

' Wenn Konstante gesetzt Trace Output schreiben
#if Sht71_testmodus = 1                                 ' Trace
   Print #2 , "Befehl :    " ; Hex(sht71_transceive_struct(1))
   Print #2 , "Datenwort : " ; Hex(value)
   Print #2 , "CRC :       " ; Hex(crc_reversed)
#endif                                                  ' Trace

If Crc_register = Crc_reversed Then
   Local_temp_byte1 = Crc_ok

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Sht71_testmodus = 1                              ' Trace
      Print #2 , "CRC - OK"
      Print #2 , ""
   #endif                                               ' Trace
Else                                                    ' CRC-Prüfung
   Local_temp_byte1 = Crc_nok

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Sht71_testmodus = 1                              ' Trace
      Print #2 , "CRC - NOK"
```

```
         Print #2 , ""
      #endif                                                ' Trace
   End If                                                   ' CRC-Prüfung

   ' Rückgabeparameter an Funktionswert übergeben
   Sht71_crc_check = Local_temp_byte1

End Function
'-- End Sht71_crc_check ------------------------------------------------------


' **********************
' * LIGHT: Lichtsensor *
' **********************

'-------------------------------------------------------------------------------
' LIGHT - Function: Light_measurement
' Funktion liefert einen normierten Lichtwert im Wertebereich zwischen 0-15
' zurück welcher proportional der aktuell vom TAOS gelieferten Lichtintensität
' entspricht.
'
' Parameter:    Value    = Byte für Anzahl der durchzuführenden Messungen
' Rückgabewert: Bytewert 0 - 16 für Helligkeit
'-------------------------------------------------------------------------------
Function Light_measurement(byval Value As Byte) As Byte

   ' ----- lokale Variablen -----
   Local Result_word As Word                                ' Arbeitsvariable für Akkumulierte Mess-
werte
   Local Temp_word As Word                                  ' Arbeitsvariable für ADC-Messwert
   Local Temp_byte As Byte                                  ' Teporäre Byte-Arbeitsvariable

   ' ----- Programmcode -----

   ' 5.11.2008: Modifikation
   ' Es zeigt sich, dass der Messwert Aufgrund des Quantisierungsrauschens
   ' sehr schnell / stark schwanken kann. Aus diesem Grund wird die Funktion
   ' um einen Parameter ergänzt der angiebt, wie oft die Messung durchgeführt
   ' und gemittelt werden soll.

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Light_testmodus                                      ' Trace
      Print #2 , "Enter Light_measurement"
      Print #2 , "Anzahl der Messungen : " ; Value
   #endif                                                   ' Trace

   ' akkumulierte Arbeitsvariable auf NULL setzen
   Result_word = 0

   ' Die Spannungsversorgung für den ADC einschalten und den ADC starten
   Start Adc

   ' Schleife über Anzahl an Messwerten fahren
   For Temp_byte = 1 To Value

      ' Single-Shot für ADC-Wert (Licht sampeln)
      Temp_word = Getadc(light_channel)

      ' Wenn Konstante gesetzt Trace Output schreiben
      #if Light_testmodus                                   ' Trace
         Print #2 , "ADC-Wert " ; Temp_byte ; " : " ; Temp_word
      #endif                                                ' Trace

      ' ----- Normierung durchfühen auf Wertebereich 0-15 -----

      ' ADC0 im ATmega128 arbeitet mit 10-Bit Auflösung = 0..1023
      ' Normierung auf 0-15 (4-Bit) = ADC / 64 bedeutet ShiftRight um 6 Bit
      Shift Temp_word , Right , 6

      ' Messwerte akumulieren
      Result_word = Result_word + Temp_word

      ' Wenn Konstante gesetzt Trace Output schreiben
      #if Light_testmodus                                   ' Trace
         Print #2 , "Normiert : " ; Temp_word
      #endif                                                ' Trace

   Next Temp_byte                                           ' Messwertsummation

   ' Wenn Konstante gesetzt Trace Output schreiben
```

```
   #if Light_testmodus                                          ' Trace
      Print #2 , "Gesamt   : " ; Result_word
   #endif                                                       ' Trace

   ' Mit STOP ADC kann die Spannung abgeschaltet und der ADC gestopped werden
   Stop Adc

   ' Messwerte mitteln
   Temp_word = Result_word / Value
   Temp_byte = Temp_word And &H000F

   ' Trace output
   #if Light_testmodus                                          ' Trace
      Print #2 , "Ergebnis " ; Temp_byte
   #endif                                                       ' Trace

   ' Updateflag setzen
   Light_update_flag = True

   ' Messwert in Rückgabewert übergeben
   Light_measurement = Temp_byte

End Function
'-- End Light_measurement ------------------------------------------------------


' **********************
' * EE:  EEPROM extern *
' **********************

'-------------------------------------------------------------------------------
' EE - Funktion: Ee_read_eeprom_byte
' Funktion um ein einzelnes Byte von einer bestimmten Speicheradresse aus dem
' EEPROM zu lesen.
'
' Parameter:     Adress = Adresse (Word) von der das Byte gelesen werden soll
'
' Rückgabewert:  Bytevariable die von der Adresse ausgelesen wurde.
'-------------------------------------------------------------------------------
Function Ee_read_eeprom_byte(byval Adress As Word) As Byte

   ' ----- Lokale Variable definieren -----
   Local Ee_adrl As Byte
   Local Ee_adrh As Byte
   Local Temp_byte As Byte

   ' ----- Programmcode -----

   ' Adresse in High-Byte und Low-Byte zerlegen
   Ee_adrl = Low(adress)
   Ee_adrh = High(adress)

   ' Kommunikation über I2C-Bus
   I2cstart                                                     ' START-Sequenz senden
   I2cwbyte Address_write                                       ' Daten schreiben initiieren
   I2cwbyte Ee_adrh                                             ' High-Byte der Adresse übertragen
   I2cwbyte Ee_adrl                                             ' Low-Byte der Adresse übertragen

   I2cstart                                                     ' START-Sequenz erneut senden
   I2cwbyte Address_read                                        ' Daten lesen initiieren
   I2crbyte Temp_byte , Nack                                    ' Byte auslesen und quittieren
   I2cstop                                                      ' STOP-Sequenz

   ' Temporäre Variable als Rückgabewert übergeben
   Ee_read_eeprom_byte = Temp_byte

End Function
'-- End Ee_read_eeprom_byte ----------------------------------------------------


'-------------------------------------------------------------------------------
' EE - Funktion: Ee_read_eeprom_word
' Funktion um ein zwei Bytes (Word) von einer bestimmten Speicheradresse aus
' dem EEPROM zu lesen.
'
' Parameter:     Adress = Adresse (Word) von der das Word gelesen werden soll
'
' Rückgabewert:  Wordvariable die von der Adresse ausgelesen wurde.
'-------------------------------------------------------------------------------
Function Ee_read_eeprom_word(byval Adress As Word) As Word
```

```
   ' ----- Lokale Variable definieren -----
   Local Ee_adrl As Byte
   Local Ee_adrh As Byte

   Local Ee_datal As Byte
   Local Ee_datah As Byte

   Local Temp_word As Word

   ' ----- Programmcode -----

   ' Adresse in High-Byte und Low-Byte zerlegen
   Ee_adrl = Low(adress)
   Ee_adrh = High(adress)

   ' Kommunikation über I2C-Bus
   I2cstart                                          ' START-Sequenz senden
   I2cwbyte Address_write                            ' Daten schreiben initiieren
   I2cwbyte Ee_adrh                                  ' High-Byte der Adresse übertragen
   I2cwbyte Ee_adrl                                  ' Low-Byte der Adresse übertragen

   I2cstart                                          ' START-Sequenz erneut senden
   I2cwbyte Address_read                             ' Daten lesen initiieren
   I2crbyte Ee_datah , Ack                           ' High-Byte auslesen und weiter
   I2crbyte Ee_datal , Nack                          ' Low-Byte auslesen und Ende
   I2cstop                                           ' STOP-Sequenz

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Eeprom_testmodus                              ' Trace
      Print #2 , "Data high: " ; Hex(ee_datah)
      Print #2 , "Data low:  " ; Hex(ee_datal)
   #endif                                            ' Trace

   ' Temporäre Variable als Rückgabewert übergeben
   Temp_word = Ee_datah
   Shift Temp_word , Left , 8
   Temp_word = Temp_word And &HFF00
   Temp_word = Temp_word + Ee_datal

   ' Rückgabeparameter an Funktionswert übergeben
   Ee_read_eeprom_word = Temp_word

End Function
'-- End Ee_read_eeprom_byte ------------------------------------------------


' ************************
' * Allgemeine Funktionen *
' ************************

'-------------------------------------------------------------------------------
' Function: Byte2string
' Funktion liefert als Ergebnis eine "rechtsbündig" formatierten String einer
' übergebenen Byte-Variablen. Der String ist max. 3 Zeichen groß und beinhaltet
' ggf. bei kleineren Zahlen das übergebene Character als Füllzeichen von links
' führend aufgefüllt.
'
' Parameter:    Value     = Byte das als SString formatiert werden soll
'               Character = ASCII-Wert des Zeichens mit dem ggf. aufgefüllt
'               werden soll.
' Rückgabewert: String der Byte-Variablen inkl. Füllzeichen
'-------------------------------------------------------------------------------
Function Byte2string(byval Value As Byte , Byval Character As Byte) As String

   ' ----- lokale Variablen -----
   Local String_length As Byte
   Local Process_string As String * 4

   ' ----- Programmcode -----

   ' Basis-String anlegen
   Process_string = Str(value)
   ' Länge ermitteln
   String_length = Len(process_string)

   ' Abhängig von Länge Füllzeichen ergänzen
   Select Case String_length
      Case 1:
         ' Wenn Länge 1 müssen zwei Füllzeichen links ergänzt werden
```

```
         Process_string = Chr(character) + Process_string
         Process_string = Chr(character) + Process_string
      Case 2:
         ' Wenn Länge 2 muss ein Füllzeichen links ergänzt werden
         Process_string = Chr(character) + Process_string
   End Select                                              ' Stringlänge

   ' Bearbeiteter String an Übergabeparameter übergeben
   Byte2string = Process_string

End Function
'-- End Byte2string ------------------------------------------------------


'--------------------------------------------------------------------------
' Function: Time2string
' Funktion liefert als Ergebnis einen formatierten String welcher die drei
' übergebenen Zeitinformationen durch den übergebenen Character getrennt
' beinhaltet.
'
' Parameter:    Value1    = Zeitparameter 1 (Byte-Wert)
'               Value2    = Zeitparameter 2 (Byte-Wert)
'               Value3    = Zeitparameter 3 (Byte-Wert)
'               Character = ASCII-Wert des Zeichens mit dem getrennt werden soll
' Rückgabewert: String der Zeitinformation
'--------------------------------------------------------------------------

Function Time2string(byval Value1 As Byte , Byval Value2 As Byte , Byval Value3 As Byte , Byval
Character As Byte) As String

   ' ----- lokale Variablen -----
   Local Process_string As String * 20
   Local Byte_temp_1 As Byte
   Local Byte_temp_2 As Byte
   Local Byte_temp_3 As Byte

   ' ----- Programmcode -----

   ' Zunächste alles in schönes BCD-Format wandeln
   Byte_temp_1 = Makebcd(value1)
   Byte_temp_2 = Makebcd(value2)
   Byte_temp_3 = Makebcd(value3)

   ' Dann String nach Muster xx?yy?zz zusammebauen. z.B. 12.34.56 oder 12:34:56
   Process_string = Bcd(byte_temp_1) + Chr(character) + Bcd(byte_temp_2) + Chr(character) +
Bcd(byte_temp_3)

   ' Process_string = Bcd(byte_temp_1) + Chr(character)
   ' Process_string = Process_string + Bcd(byte_temp_2)
   ' Process_string = Process_string + Chr(character)
   ' Process_string = Process_string + Bcd(byte_temp_3)

   ' Bearbeiteter String an Übergabeparameter übergeben
   Time2string = Process_string

End Function
'-- End Time2string ------------------------------------------------------


'--------------------------------------------------------------------------
' Function: Schaltjahr
' Funktion überprüft ob übergebenes Jahr ein Schaltjahr ist.
' Das Jahr muss als vollständiges Jahr z.B. 2008 übergeben werden.
'
' Parameter:    Value    = Jahr welches überprüft werden soll (Word-Wert)
' Rückgabewert: Bytevariable mit True (&H01) oder False (&H00)
'--------------------------------------------------------------------------
Function Schaltjahr(byval Value As Word) As Byte

   ' ----- lokale Variablen -----
   Local Word_temp_1 As Word
   Local Result_value As Byte

   ' ----- Programmcode -----

   ' Regel für Schaltjahre:
   ' Das Jahr ist durch vier teilbar.
   ' Das Jahr darf nicht durch 100 teilbar sein,
   ' ausser es ist durch 400 teilbar.
```

```
   ' Prüfung ob Jahr durch vier teilbar
   Word_temp_1 = Value Mod 4

   If Word_temp_1 = 0 Then
      ' Ja, Jahr könnte Schaltjahr sein
      ' Prüfung ob Jahr durch 100 teilbar
      Word_temp_1 = Value Mod 100
      If Word_temp_1 = 0 Then
         ' Jahr ist durch 100 teilbar
         ' Prüfung ob Jahr durch 400 teilbar
         Word_temp_1 = Value Mod 400
         If Word_temp_1 = 0 Then
            Result_value = True
         Else                                              ' Jahr durch 400 teilbar?
            Result_value = False
         End If                                            ' Jahr durch 400 teilbar?
      Else                                                 ' Jahr durch 100 teilbar?
         ' Jahr ist nicht durch 100 teilbar
         Result_value = True
      End If                                               ' Jahr durch 100 teilbar?
   Else                                                    ' Jahr durch 4 teilbar?
      ' Jahr ist definitiv kein Schaltjahr
      Result_value = False
   End If                                                  ' Jahr durch 4 teilbar?

   ' Rückgabeparameter an Funktionswert übergeben
   Schaltjahr = Result_value

End Function
'-- End Schaltjahr --------------------------------------------------------


'*****************************************************************************
' GOSubroutinen
'*****************************************************************************

' ********************
' * LCD:  Display    *
' ********************

'-----------------------------------------------------------------------------
' LCD - Gosub-Routine: Lcd_konfig
' Subroutine zur Konfiguration des LCD-Display inkl. der dafür
' notwendigen Timer.
'-----------------------------------------------------------------------------
Lcd_konfig:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Lcd_testmodus                                     ' Trace
      Print #2 , "Enter Lcd_konfig"
   #endif                                                ' Trace

   '----- Konfiguration LCD-Helligkeit (PWM) -----
   'Variable für Helligkeit in Timerregister laden
   Pwm3a = Lcd_pwmwert

   'Port E.3 ist Timer-Ausgang OC3A
   Config Pine.3 = Output

Return
'-- End Lcd_konfig ------------------------------------------------------


'-----------------------------------------------------------------------------
' LCD - Gosub-Routine: Lcd_display_an
' Routine schaltet die Spannungsversorgung für das Display an
'-----------------------------------------------------------------------------
Lcd_display_an:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Lcd_testmodus                                     ' Trace
      Print #2 , "Enter Lcd_display_an"
   #endif                                                ' Trace

   Lcd_on_dataout = 1

Return
'-- End Lcd_display_an -------------------------------------------------
```

```
'-------------------------------------------------------------------------------
' LCD - Gosub-Routine: Lcd_display_aus
' Routine schaltet die Spannungsversorgung für das Display aus
'-------------------------------------------------------------------------------
Lcd_display_aus:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Lcd_testmodus                                        ' Trace
      Print #2 , "Enter Lcd_display_aus"
   #endif                                                   ' Trace

   Lcd_on_dataout = 0

Return
'-- End Lcd_display_aus ---------------------------------------------------


'-------------------------------------------------------------------------------
' LCD - Gosub-Routine: Lcd_beleuchtung_an
' Routine schaltet die Beleuchtung ein
'-------------------------------------------------------------------------------
Lcd_beleuchtung_an:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Lcd_testmodus                                        ' Trace
      Print #2 , "Enter Lcd_beleuchtung_an"
   #endif                                                   ' Trace

   ' PWM-Wert in Timerregister laden
   Pwm3a = Lcd_pwmwert

Return
'-- End Lcd_beleuchtung_an ------------------------------------------------


'-------------------------------------------------------------------------------
' LCD - Gosub-Routine: Lcd_beleuchtung_aus
' Routine schaltet die Beleuchtung aus
'-------------------------------------------------------------------------------
Lcd_beleuchtung_aus:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Lcd_testmodus                                        ' Trace
      Print #2 , "Enter Lcd_beleuchtung_aus"
   #endif                                                   ' Trace

   ' PWM-Wert in Timerregister laden
   Pwm3a = Lcd_helligkeit_aus

Return
'-- End Lcd_beleuchtung_aus -----------------------------------------------


'-------------------------------------------------------------------------------
' LCD - Gosub-Routine: Lcd_display_ein
' Routine schaltet das gesamte Display ein
'-------------------------------------------------------------------------------
Lcd_display_on:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Lcd_testmodus                                        ' Trace
      Print #2 , "Enter Lcd_display_ein"
   #endif                                                   ' Trace

   ' Instruction Display ON/OFF gemäß LCD Befehlskatalog
   _temp1 = &B00001100
   !Call _Lcd_control

Return
'-- End Lcd_display_ein ---------------------------------------------------


'-------------------------------------------------------------------------------
' LCD - Gosub-Routine: Lcd_display_aus
' Routine schaltet das gesamte Display aus
'-------------------------------------------------------------------------------
Lcd_display_off:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Lcd_testmodus                                        ' Trace
```

```
      Print #2 , "Enter Lcd_display_aus"
   #endif                                                  ' Trace

   ' Instruction Display ON/OFF gemäß LCD Befehlskatalog
   _temp1 = &B00001000
   !Call _Lcd_control

Return
'-- End Lcd_display_ein ------------------------------------------------------


'--------------------------------------------------------------------------------
' LCD - Gosub-Routine: Lcd_contrastset
' Routine berechnen neue Kontrastwerte und steuert direkt den
' Kontroller des Display an.
'--------------------------------------------------------------------------------
Lcd_kontrastset:                                           ' Kontrasteinstellung Display

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Lcd_testmodus                                       ' Trace
      Print #2 , "Enter Lcd_kontrastset" ; Lcd_kontrastwert
   #endif                                                  ' Trace

   ' Verarbeitung des Kontrastwertes für High-Byte und Low-Byte
   Temp_byte_1 = Lcd_kontrastwert And &B00001111
   Temp_byte_1 = Temp_byte_1 + &B01110000

   Temp_byte_2 = Lcd_kontrastwert
   Shift Temp_byte_2 , Right , 4
   Temp_byte_2 = Temp_byte_2 And &B00000011
   Temp_byte_2 = Temp_byte_2 + &B01010100

   ' Achtung: rCall muss durch Call ersetzen werden. rCall ist ein relavtiver
   ' Sprung welcher nur über 2kByte Zeilen geht. Die LCD-Lib liegt aber Auf-
   ' grund der Code-Größe außerhalb des Bereichs und kann deshalb nicht mehr
   ' angesprungen werden.
   ' Error 220: Forward jump out of range

   ' Instruction Table 1 einstellen [0,1]
   _temp1 = &B00101001
   '!rCall _Lcd_control
   !Call _Lcd_control

   ' Tempvar_1 = &B0111xxxx für Kontrast Set Instruction Table 1 - Low Byte
   _temp1 = Temp_byte_1
   '!rCall _Lcd_control
   !Call _Lcd_control

   ' Temovar_2 = &B010101xx für Kontrast Set Instruction Table 1 - High Byte
   _temp1 = Temp_byte_2
   '!rCall _Lcd_control
   !Call _Lcd_control

   ' Zurückschalten auf Instruction Table 0 [0,0]
   _temp1 = &B00101000
   '!rCall _Lcd_control
   !Call _Lcd_control

Return
'-- End Lcd_kontrastset -------------------------------------------------------


'--------------------------------------------------------------------------------
' LCD - Gosub-Routine: Lcd_print
' Die Routine gibt den Lcd_textbuffer für das Menü auf dem LCD-Display aus
'Result:     LCD
'--------------------------------------------------------------------------------
Lcd_print:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Lcd_testmodus                                       ' Trace
      Print #2 , "Enter Lcd_print"
   #endif                                                  ' Trace

   Cls                                                     ' Clear Screen

   Select Case Menue_ausgabeflags
      Case 1:
         ' Nur 1. Zeile ausgaben
         ' Locate 1 , 1 : Lcd Menue_lcd_textbuffer(1)
```

```
         Call Lcd_writesingleline(1 , Menue_lcd_textbuffer(1))
      Case 2:
         ' Nur 2. Zeile ausgaben
         ' Locate 2 , 1 : Lcd Menue_lcd_textbuffer(2)
         Call Lcd_writesingleline(2 , Menue_lcd_textbuffer(2))
      Case 4:
         ' Nur 3. Zeile ausgaben
         ' Locate 3 , 1 : Lcd Menue_lcd_textbuffer(3)
         Call Lcd_writesingleline(3 , Menue_lcd_textbuffer(3))
      Case 3:
         ' 1. und 2. Zeile ausgeben
         ' Locate 1 , 1 : Lcd Menue_Lcd_textbuffer(1)
         ' Locate 2 , 1 : Lcd Menue_Lcd_textbuffer(2)
         Call Lcd_writedoubleline(1 , 2 , Menue_lcd_textbuffer(1) , Menue_lcd_textbuffer(2) )
      Case 5:
         ' 1. und 3. Zeile ausgeben
         ' Locate 1 , 1 : Lcd Menue_Lcd_textbuffer(1)
         ' Locate 3 , 1 : Lcd Menue_Lcd_textbuffer(3)
         Call Lcd_writedoubleline(1 , 3 , Menue_lcd_textbuffer(1) , Menue_lcd_textbuffer(3) )
      Case 6:
         ' 2. und 3. Zeile ausgeben
         ' Locate 2 , 1 : Lcd Menue_Lcd_textbuffer(2)
         ' Locate 3 , 1 : Lcd Menue_Lcd_textbuffer(3)
         Call Lcd_writedoubleline(2 , 3 , Menue_lcd_textbuffer(2) , Menue_lcd_textbuffer(3) )
      Case 7:
         ' Alle Zeilen ausgeben
         ' Locate 1 , 1 : Lcd Menue_Lcd_textbuffer(1)
         ' Locate 2 , 1 : Lcd Menue_Lcd_textbuffer(2)
         ' Locate 3 , 1 : Lcd Menue_Lcd_textbuffer(3)
         Call Lcd_writetrippleline(menue_lcd_textbuffer(1) , Menue_lcd_textbuffer(2) , Me-
nue_lcd_textbuffer(3) )
   End Select                                                ' Menüausgabeflag

   Menue_state_renew = 0

Return
'-- End Lcd_print ------------------------------------------------------------


' ********************
' * FAN:  Lüfter     *
' ********************

'-------------------------------------------------------------------------------
' FAN - Gosub-Routine: Fan_konfig
' Subroutine zur Konfiguration der PWM für die Lüftersteuerung
'-------------------------------------------------------------------------------
Fan_konfig:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Fan_testmodus                                         ' Trace
      Print #2 , "Enter Fan_konfig"
   #endif                                                    ' Trace

   '----- Konfiguration FAN-Geschwindigkeit (PWM) -----
   'Variable für Geschwindgkeit in Timerregister laden
   Pwm3b = Fan_pwmwert

   'Port E.4 ist Timer-Ausgang OC3B
   Config Pine.4 = Output

Return
'-- End Lcd_konfig ---------------------------------------------------------


'-------------------------------------------------------------------------------
' FAN - Gosub-Routine: Fan_an
' Routine schaltet den Lüfter ein
'-------------------------------------------------------------------------------
Fan_lueftung_an:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Fan_testmodus                                         ' Trace
      Print #2 , "Enter Fan_an"
      Print #2 , "Setze PWM-Wert " ; Fan_pwmwert
   #endif                                                    ' Trace

   ' PWM-Wert in Timerregister laden
   Pwm3b = Fan_pwmwert
```

```
Return
'-- End Lcd_beleuchtung_an ------------------------------------------------


'--------------------------------------------------------------------------------
' FAN - Gosub-Routine: Fan_aus
' Routine schaltet den Lüfter aus
'--------------------------------------------------------------------------------
Fan_lueftung_aus:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Fan_testmodus                                    ' Trace
      Print #2 , "Enter Fan_aus"
   #endif                                               ' Trace

   ' PWM-Wert in Timerregister laden
   Pwm3b = Fan_belueftung_aus

Return
'-- End Fan_aus ------------------------------------------------------------


' ********************
' * SOUND: Piezo     *
' ********************

'--------------------------------------------------------------------------------
' SOUND - Gosub-Routine: Sound_konfig
' Konfiguration der notwendigen PIN's und Ports für die Ansteuerung des Piezo-
' Schwingkreises.
'--------------------------------------------------------------------------------
Sound_konfig:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Sound_testmodus = 1                              ' Trace
      Print #2 , "Enter Sound_konfig"
   #endif                                               ' Trace

   Config Sound_pin = Output                            ' PortD.5 als Output

Return
'-- End Sound_konfig ------------------------------------------------------


'--------------------------------------------------------------------------------
' SOUND - Gosub-Routine: Sound_an
' Routinen schaltet die Ausgabe des Signaltons ein.
'--------------------------------------------------------------------------------
Sound_an:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Sound_testmodus = 1                              ' Trace
      Print #2 , "Enter Sound_an"
   #endif                                               ' Trace

   Set Sound_dataout                                    ' Bit setzen

Return
'-- End Sound_an ----------------------------------------------------------


'--------------------------------------------------------------------------------
' SOUND - Gosub-Routine: Sound_aus
' Routinen schaltet die Ausgabe des Signaltons aus.
'--------------------------------------------------------------------------------
Sound_aus:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Sound_testmodus = 1                              ' Trace
      Print #2 , "Enter Sound_aus"
   #endif                                               ' Trace

   Reset Sound_dataout                                  ' Bit rücksetzen

Return
'-- End Sound_aus ----------------------------------------------------------


' ********************
' * KEY: Tastatur    *
```

```
' ********************

'-------------------------------------------------------------------------------
' KEY - Gosub-Routine: Test_enable_keyinterrupt
' Routine aktiviert den INT7 für die Tastaturereigniserfassung erneut. Zuvor
' wird jedoch das Bit für INT7 im External-Interrupt-Flag-Register gelöscht
' da bereits (durch prellen der Tasten) ein neuer Interrupt anstehen kann.
'-------------------------------------------------------------------------------
Key_enable_keyinterrupt:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Key_testmodus                                        ' Trace
      Print #2 , "Enter Test_enable_keyinterrupt"
      Print #2 , "ORIG EIFR : " ; Bin(eifr)
   #endif                                                   ' Trace

   ' Dieses Bit wird gesetzt, wenn am INT7-Pin eine Interrupt-Kondition,
   ' entsprechend der Konfiguration, erkannt wird. Wenn das External  Enable
   ' Interrupt Flag gesetzt ist, wird die Interruptroutine angesprungen.
   ' Das Flag wird automatisch gelöscht , wenn die Interruptroutine beendet ist.
   ' Alternativ kann das Flag gelöscht werden , indem der Wert 1(!)
   ' eingeschrieben wird. ACHTUNG: Wert 1 !!!!!

   ' Lösche Flankeninterrupt INT7 wenn bereits mehrfach aufgetreten.
   Eifr.intf7 = 1

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Key_testmodus                                        ' Trace
      Print #2 , "RST EIFR  : " ; Bin(eifr)
   #endif                                                   ' Trace

   ' Interrupt wieder freigeben
   Enable Int7

Return
'-- End Key_enable_keyinterrupt ------------------------------------------------


' ********************
' * MENUE:  Menue    *
' ********************

'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_change_state
' Routine stellt den Status der Menue-State-Machine fest und führt ggf. Wechsel
' durch. Hierbei werden die Pointer auf die States neu befüllt und die notwen-
' digen Variablen neu gesetzt.
'-------------------------------------------------------------------------------
Menue_change_state:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_change_state"
   #endif                                                   ' Trace

   ' 16-Bit-Adresse des aktuellen Labels / Menüzustandes laden
   lds R8, {Menue_State}
   lds R9, {Menue_State + 1}

   ' Mittels Schleife über das Tastatur-Array Verzweigung prüfen
   For Temp_byte_1 = 1 To Key_buttons
      ' Ansprungsadresse laden
      Read Temp_word_1
      ' Prüfen ob zugehörige Taste betätigt
      If Key_taste = Key_codes(temp_byte_1) Then
         ' Wenn Taste betätigt dann prüfen ob Ansprungziel gültig
         If Temp_word_1 <> Loadlabel(null) Then
            ' Menü muss erneuert werden - Flag setzen
            Menue_state_renew = 1
            ' Tastenstatus zurücksetzen um Einfluss von GOSUBs zu verhindern
            Key_taste = Key_null
            ' Neuer Menüstatus in Zustandsmaschine laden
            Menue_state = Temp_word_1
         End If                                             ' Ansprungziel
      End If                                                ' gültige Taste
   Next Tempb_byte_1                                        ' Array mit Tasten-Codes

   ' Adresse des anzuspringenden Unterprogramms einlesen
   Read Menue_state_gosub
```

```
   ' Ausgabeflags einlesen
   Read Menue_ausgabeflags

   ' Textzeilen für LCD-Ausgabe einlesen
   Read Menue_lcd_textbuffer(1)                              'read LCD text
   Read Menue_lcd_textbuffer(2)                              'read LCD text
   Read Menue_lcd_textbuffer(3)                              'read LCD text

Return
'-- End Menue_chane_state ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_changes_state_by_sub
' Routine wird von Subroutinen aufgerufen falls eine Menüänderungen händisch
' erzwungen werden soll.
' Hier wird die Statemachine auf einen neuen Zustand umgeschaltet. Der Zustand
' kann in der Variablen Menue_state hinterlegt werden.
'--------------------------------------------------------------------------------
Menue_change_state_by_sub:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Menue_testmodus                                       ' Trace
      Print #2 , "Enter Menue_change_state_by_sub"
   #endif                                                    ' Trace

   ' Menü muss erneuert werden - Flag setzen
   Menue_state_renew = 1

   ' Eigentliche Menüänderung durchführen
   Gosub Menue_change_state

Return
'-- End Menue_chane_state_by_sub ------------------------------------------------

'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_call_subroutine
' Routine liest aus aktuellem Menue-State den Sprungadresse und springt angege-
' benes Ziel an.
'--------------------------------------------------------------------------------
Menue_call_subroutine:

         LDS R31, {Menue_State_gosub+1}                      'High see Bascom-Doc Mixing ASM and
BASIC
         LDS R30, {Menue_State_gosub}                        'Low
         LSR R31                                             'Division durch 2 (Alternativ ADR ver-
wenden)
         ROR R30
         'Call zum Sub
         ICALL

Return
'-- End Menue_call_subroutine --------------------------------------------------



'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men2 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men2:

   ' Trace Output schreiben
   #if Menue_testmodus                                       ' Trace
      Print #2 , "Enter Menue_men2"
   #endif                                                    ' Trace

   ' Menue befndet sich außerhalb der anzeigepflichtigen Untermenüs
   Menue_inrelmenue_flag = 0

Return
'-- End Menue_men2 -------------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men211 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
```

```
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men211:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men211"
   #endif                                                 ' Trace

   ' Zeitsteuerung Timeout ausgeben

   Temp_string_1 = Byte2string(rel_beschattung_delay , &H30 )       ' 0 als Auffüllzeichen verwenden
   Locate 3 , 10 : Lcd Temp_string_1 ; " sec"

Return
'-- End Menue_men211 ------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men212 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men212:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men212"
   #endif                                                 ' Trace

   ' Zeitsteuerung Uhrzeit AUF ausgeben
   Temp_byte_1 = Makebcd(timer_open_stunde)
   Temp_byte_2 = Makebcd(timer_open_minute)
   Locate 3 , 6 : Lcd Bcd(temp_byte_1) ; ":" ; Bcd(temp_byte_2) ; "h"

   Locate 3 , 13
   Select Case Timer_open_flag
      Case Aktiv:
         Lcd "  JA"
      Case Passiv:
         Lcd "NEIN"
   End Select                                             ' Timer-Open-Flag

Return
'-- End Menue_men212 ------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men213 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men213:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men213"
   #endif                                                 ' Trace

   ' Zeitsteuerung Uhrzeit ZU ausgeben
   Temp_byte_1 = Makebcd(timer_close_stunde)
   Temp_byte_2 = Makebcd(timer_close_minute)
   Locate 3 , 6 : Lcd Bcd(temp_byte_1) ; ":" ; Bcd(temp_byte_2) ; "h"

   Locate 3 , 13
   Select Case Timer_close_flag
      Case Aktiv:
         Lcd "  JA"
      Case Passiv:
         Lcd "NEIN"
   End Select                                             ' Timer-Open-Flag

Return
'-- End Menue_men213 ------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men214 (Menue-Unterstützung-Routine)
```

```
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------
Menue_men214:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men214"
   #endif                                                 ' Trace

   ' Auffahrgrund Wind ja/nein ausgeben
   Temp_string_1 = Byte2string(wind_beschattung_auf , &H30)
   Locate 3 , 7 : Lcd Temp_string_1
   Locate 3 , 13
   Select Case Wind_autoauffahren
      Case True:
         Lcd "  JA"
      Case False:
         Lcd "NEIN"
   End Select                                             ' Wind-Auffahren-Flag

Return
'-- End Menue_men214 ---------------------------------------------------------


'-------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men215 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------
Menue_men215:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men215"
   #endif                                                 ' Trace

   ' Auffahrgrund Regen ja/nein ausgeben
   Locate 3 , 13
   Select Case Rain_autoauffahren
      Case True:
         Lcd "  JA"
      Case False:
         Lcd "NEIN"
   End Select                                             ' Regen-Auffahren-Flag

Return
'-- End Menue_men215 ---------------------------------------------------------


'-------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men216 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------
Menue_men216:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men216"
   #endif                                                 ' Trace

   ' Auffahrgrund Licht Helligkeit und ja/nein ausgeben
   Temp_byte_1 = Makebcd(rel_auf_dunkelheit)
   Locate 3 , 7 : Lcd Bcd(temp_byte_1)

   Locate 3 , 13
   Select Case Light_autoauffahren
      Case True:
         Lcd "  JA"
      Case False:
         Lcd "NEIN"
   End Select                                             ' Licht-Auffahren-Flag

Return
'-- End Menue_men216 ---------------------------------------------------------
```

```
'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men217 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men217:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men217"
   #endif                                                 ' Trace

   ' Zufahrgrund Temperatur ausgeben
   Temp_string_1 = Byte2string(rel_zu_temperatur , &H30)
   Locate 3 , 7 : Lcd Temp_string_1
   Locate 3 , 13
   Select Case Sht71_autozufahren
      Case True:
         Lcd "  JA"
      Case False:
         Lcd "NEIN"
   End Select                                             ' Temperatur-Zufahren-Flag

Return
'-- End Menue_men217 ------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men221 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men221:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men221"
   #endif                                                 ' Trace

   ' Flag für Dimmermenü wegen Shortcuts
   Menue_inrelmenue_flag = 0

Return
'-- End Menue_men221 ------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men2211 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men2211:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men2211"
   #endif                                                 ' Trace

   ' Prüfen ob es Regnet
   Temp_byte_1 = Rain_measurement()
   If Temp_byte_1 = Rain_no Then

   ' NEIN, es Regnet nicht, Funktion kann ausgeführt werden

      ' Prüfen ob maximale Windgeschwindigkeit erreicht
      Temp_byte_2 = Wind_geschwindigkeit_kmh
      If Temp_byte_2 < Wind_beschattung_auf Then

         ' NEIN, Windgeschwindigkeit ist noch unterhalb kritischem Wert

         ' Steuerung Antrieb Auf
         Gosub Relais_beschattung_auf

      Else                                                ' Windgeschwindigkeit

         Cls
```

```
         Locate 2 , 4 : Lcd "Windmodus"

      End If                                                ' Ende Wind
   Else                                                     ' Regen

      Cls
      Locate 2 , 4 : Lcd "Regenmodus"

   End If                                                   ' Ende Regen

   ' Menue befndet sich außerhalb der anzeigepflichtigen Untermenüs
   Menue_inrelmenue_flag = 1

Return
'-- End Menue_men2211 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men222 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men222:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_men222"
   #endif                                                   ' Trace

   ' Flag für Dimmermenü wegen Shortcuts
   Menue_inrelmenue_flag = 0

Return
'-- End Menue_men222 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men2221 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men2221:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_men2221"
   #endif                                                   ' Trace

   ' Prüfen ob es Regnet
   Temp_byte_1 = Rain_measurement()
   If Temp_byte_1 = Rain_no Then

      ' NEIN, es Regnet nicht, Funktion kann ausgeführt werden

      ' Prüfen ob maximale Windgeschwindigkeit erreicht
      Temp_byte_2 = Wind_geschwindigkeit_kmh
      If Temp_byte_2 < Wind_beschattung_auf Then

         ' NEIN, Windgeschwindigkeit ist noch unterhalb kritischem Wert

         ' Steuerung Antrieb Zu
         Gosub Relais_beschattung_zu

      Else                                                  ' Windgeschwindigkeit

         Cls
         Locate 2 , 4 : Lcd "Windmodus"

      End If                                                ' Ende Wind
   Else                                                     ' Regen

      Cls
      Locate 2 , 4 : Lcd "Regenmodus"

   End If                                                   ' Ende Regen

   ' Menue befndet sich außerhalb der anzeigepflichtigen Untermenüs
   Menue_inrelmenue_flag = 2
```

```
Return
'-- End Menue_men2221 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men223 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men223:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men223"
   #endif                                                 ' Trace

   ' Flag für Dimmermenü wegen Shortcuts
   Menue_inrelmenue_flag = 0

Return
'-- End Menue_men223 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men2211 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men2231:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men2231"
   #endif                                                 ' Trace

   ' Prüfen ob es Regnet
   Temp_byte_1 = Rain_measurement()
   If Temp_byte_1 = Rain_no Then

   ' NEIN, es Regnet nicht, Funktion kann ausgeführt werden

      ' Prüfen ob maximale Windgeschwindigkeit erreicht
      Temp_byte_2 = Wind_geschwindigkeit_kmh
      If Temp_byte_2 < Wind_beschattung_auf Then

         ' NEIN, Windgeschwindigkeit ist noch unterhalb kritischem Wert

         ' Steuerung Antrieb Stopp
         Gosub Relais_beschattung_stop

      Else                                                ' Windgeschwindigkeit

         Cls
         Locate 2 , 4 : Lcd "Windmodus"

      End If                                              ' Ende Wind
   Else                                                   ' Regen

      Cls
      Locate 2 , 4 : Lcd "Regenmodus"

   End If                                                 ' Ende Regen

   ' Menue befndet sich außerhalb der anzeigepflichtigen Untermenüs
   Menue_inrelmenue_flag = 3

Return
'-- End Menue_men2231 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men3 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men3:
```

```
   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men3"
   #endif                                                 ' Trace

   ' Flag für Dimmermenü wegen Shortcuts
   Menue_indimmenue_flag = 0

Return
'-- End Menue_men3 ------------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men31 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men31:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men31"
   #endif                                                 ' Trace

   ' Flag für Dimmermenü wegen Shortcuts
   Menue_indimmenue_flag = 0

Return
'-- End Menue_men31 -----------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men311 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men311:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men311"
   #endif                                                 ' Trace

   ' Ausgabe von Dauerlicht und Flashmode
   Locate 2 , 13
   If Beleuchtung_flag = False Then
      Lcd "NEIN"
   Else                                                   ' Beleuchtung
      Select Case Dauerlicht_flag
         Case Aktiv : Lcd "  JA"
         Case Passiv : Lcd "NEIN"
      End Select
   End If                                                 ' Beleuchtung

   Locate 3 , 13
   Select Case Dim_flash_mode
      Case Aktiv : Lcd "  JA"
      Case Passiv : Lcd "NEIN"
   End Select                                             ' Flash-Mode

Return
'-- End Menue_men311 ------------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men312 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men312:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men312"
   #endif                                                 ' Trace
```

```
   ' Ausgabe von Dimmerstufe und Dimmergrenzen
   Temp_byte_1 = Makebcd(dim_helligkeitsstufe)
   Locate 2 , 15 : Lcd Bcd(temp_byte_1)

   Temp_byte_1 = Makebcd(dim_helligkeit_min)
   Locate 3 , 6 : Lcd Bcd(temp_byte_1)

   Temp_byte_1 = Makebcd(dim_helligkeit_max)
   Locate 3 , 15 : Lcd Bcd(temp_byte_1)

   ' Flag für Dimmermenü wegen Shortcuts
   Menue_indimmenue_flag = 4

Return
'-- End Menue_men312 ------------------------------------------------------


' Erläuterung zur Relaissteuerung
'  ---------------------------------------------------------------------
'  |    Relais 1    |      Relais 2         |         Funktion         |
'  |(Licht EIN/AUS) |  (Dauerlicht EIN/AUS) |                          |
'  |================|=======================|==========================|
'  |    abgefallen  |     abgefallen        |      Licht aus           |
'  |    abgefallen  |     angezogen         |      Licht aus           |
'  |    angezogen   |     abgefallen        |      Licht auf Dauerlicht|
'  |    angezogen   |     angezogen         |      Licht von Dimmer    |
'  ---------------------------------------------------------------------
'  Achtung: Bei Dimmer wird Free-Running benötigt!

'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men3211 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men3211:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men3211"
   #endif                                                 ' Trace

   ' Ausgabe ob Dauerlicht JA/NEIN

   Locate 2 , 7

   If Beleuchtung_flag = Aktiv Then
      Select Case Dauerlicht_flag
         Case Aktiv :
            Lcd "AN "
         Case Passiv :
            Lcd "AUS"
      End Select
   Else                                                   ' Beleuchtung
      Lcd "AUS"
   End If                                                 ' Beleuchtung

Return
'-- End Menue_men3211 ------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men32111 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men32111:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men32111"
   #endif                                                 ' Trace

   ' Wegen manuellem Eingriff in Licht Flashmode verriegeln
   Dim_flash_mode_locked = True

   ' Ausgabe ob Dauerlicht JA/NEIN inkl. Umschaltung

   ' Prüfung ob Beleuchtung überhaupt eingeschaltet
```

```
   If Beleuchtung_flag = Passiv Then

      ' Beleuchtung ist nicht eingeschaltet und läuft auch nicht auf Dimmer
      ' => Beleuchtung einschalten und Dauerlicht aktivieren
      Gosub Beleuchtung_ein
      Gosub Beleuchtung_dauerlicht_ein

   Else                                                    ' Beleuchtung

      ' Beleuchtung läuft bereits, entweder als Dimmer oder bereits im Dauerlicht

      ' Prüfen ob Dimmer im Hintergrund aktiv
      If Free_running_flag = Passiv Then

         ' NEIN, Dimmer läuft nicht => Beleuchtung aus und Relais auf NULL
         Gosub Beleuchtung_aus
         Gosub Beleuchtung_dauerlicht_ein

      Else                                                 ' Dimmer aktiv
         ' Ja, Free-Running ist aktiv und damit ist Dimmer aktiv
         ' => prüfen ob Dauerlicht an oder aus
         If Dauerlicht_flag = Aktiv Then
            Gosub Beleuchtung_dauerlicht_aus
         Else                                              ' Dauerlicht
            Gosub Beleuchtung_dauerlicht_ein
         End If                                            ' Dauerlicht
      End If                                               ' Dimmer aktiv
   End If                                                  ' Beleuchtung

   Locate 2 , 7
   If Beleuchtung_flag = Aktiv Then
      Select Case Dauerlicht_flag
         Case Aktiv :
            Lcd "AN "
         Case Passiv :
            Lcd "AUS"
      End Select                                           ' Dauerlicht
   Else                                                    ' Beleuchtung
      Lcd "AUS"
   End If                                                  ' Beleuchtung

Return
'-- End Menue_men32111 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men322 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men322:

   ' Trace Output schreiben
   #if Menue_testmodus                                   ' Trace
      Print #2 , "Enter Menue_men322"
   #endif                                                ' Trace

   ' Flag für Dimmermenü wegen Shortcuts
   Menue_indimmenue_flag = 0

Return
'-- End Menue_men322 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men3221 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men3221:

   ' Trace Output schreiben
   #if Menue_testmodus                                   ' Trace
      Print #2 , "Enter Menue_men3221"
   #endif                                                ' Trace

   ' Wegen manuellem Eingriff in Licht Flashmode verriegeln
   Dim_flash_mode_locked = True
```

```
   ' Dimmerhelligkeit ausgeben (Dimmer heller)

   ' Flag für Dimmermenü wegen Shortcuts
   Menue_indimmenue_flag = 1

   ' Prüfen ob Beleuchtung bereits eingeschaltet
   If Beleuchtung_flag = Passiv Then

      ' Beleuchtung läuft noch nicht => Beleuchtung einschalten
      Gosub Beleuchtung_ein
      Gosub Beleuchtung_dauerlicht_aus

      ' Dimmer frei laufen lasse
      Free_running_flag = Aktiv

   End If                                                  ' Beleuchtung

   ' Helligkeitsstufe auf Display ausgeben
   Locate 2 , 8
   Temp_byte_1 = Makebcd(dim_helligkeitsstufe)
   Lcd Bcd(temp_byte_1)

Return
'-- End Menue_men3221 ------------------------------------------------------


'---------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men32211 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'---------------------------------------------------------------------------
Menue_men32211:

   ' Trace Output schreiben
   #if Menue_testmodus                                     ' Trace
      Print #2 , "Enter Menue_men32211"
   #endif                                                  ' Trace

   ' Dimmer heller
   ' Dimmerhelligkeit ausgeben und Helligkeit um eins erhöhen bis Grenze erreicht

   ' Dimmer heller schalten und gegen Endwert prüfen
   If Dim_helligkeitsstufe < Dim_helligkeit_max Then
      Incr Dim_helligkeitsstufe
   End If                                                  ' Helligkeitsstufe

   ' Helligkeitsstufe auf Display ausgeben
   Locate 2 , 8
   Temp_byte_1 = Makebcd(dim_helligkeitsstufe)
   Lcd Bcd(temp_byte_1)

Return
'-- End Menue_men32211 ------------------------------------------------------


'---------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men323 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'---------------------------------------------------------------------------
Menue_men323:

   ' Trace Output schreiben
   #if Menue_testmodus                                     ' Trace
      Print #2 , "Enter Menue_men323"
   #endif                                                  ' Trace

   ' Flag für Dimmermenü wegen Shortcuts
   Menue_indimmenue_flag = 0

Return
'-- End Menue_men323 ------------------------------------------------------


'---------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men3231 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
```

```
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men3231:

   ' Trace Output schreiben
   #if Menue_testmodus                                        ' Trace
      Print #2 , "Enter Menue_men3231"
   #endif                                                     ' Trace

   ' Wegen manuellem Eingriff in Licht Flashmode verriegeln
   Dim_flash_mode_locked = True

   ' Dimmerhelligkeit ausgeben (Dimmer dunkler)

   ' Flag für Dimmermenü wegen Shortcuts
   Menue_indimmenue_flag = 2

   ' Prüfen ob Beleuchtung bereits eingeschaltet
   If Beleuchtung_flag = Passiv Then

      ' Beleuchtung läuft noch nicht => Beleuchtung einschalten
      Gosub Beleuchtung_ein
      Gosub Beleuchtung_dauerlicht_aus

      ' Dimmer frei laufen lasse
      Free_running_flag = Aktiv

   End If                                                     ' Beleuchtung

   ' Helligkeitsstufe auf Display ausgeben
   Locate 2 , 8
   Temp_byte_1 = Makebcd(dim_helligkeitsstufe)
   Lcd Bcd(temp_byte_1)

Return
'-- End Menue_men3231 ------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men32311 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men32311:

   ' Trace Output schreiben
   #if Menue_testmodus                                        ' Trace
      Print #2 , "Enter Menue_men32311"
   #endif                                                     ' Trace

   ' Dimmer dunkler
   ' Dimmerhelligkeit ausgeben und Helligkeit um eins verringern bis Grenze erreicht

   ' Dimmer dunkler schalten und gegen Endwert prüfen
   If Dim_helligkeitsstufe > Dim_helligkeit_min Then
      Decr Dim_helligkeitsstufe
   End If                                                     ' Helligkeitsstufe

   ' Helligkeitsstufe auf Display ausgeben
   Locate 2 , 8
   Temp_byte_1 = Makebcd(dim_helligkeitsstufe)
   Lcd Bcd(temp_byte_1)

Return
'-- End Menue_men32311 ------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men324 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men324:

   ' Trace Output schreiben
   #if Menue_testmodus                                        ' Trace
      Print #2 , "Enter Menue_men324"
```

```
   #endif                                                    ' Trace

   ' Flag für Dimmermenü wegen Shortcuts
   Menue_indimmenue_flag = 0

Return
'-- End Menue_men324 ------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men3241 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_men3241:

   ' Trace Output schreiben
   #if Menue_testmodus                                       ' Trace
      Print #2 , "Enter Menue_men3241"
   #endif                                                    ' Trace

   ' Wegen manuellem Eingriff in Licht Flashmode verriegeln
   Dim_flash_mode_locked = True

   ' Dimmer ausschalten

   ' Flag für Dimmermenü wegen Shortcuts
   Menue_indimmenue_flag = 3

   ' Dimmerfreilauf beenden
   Free_running_flag = Passiv

   ' Relais ggf. abschalten bzw. umschalten
   If Dauerlicht_flag = Passiv Then
      Gosub Beleuchtung_dauerlicht_ein
      Gosub Beleuchtung_aus
   End If                                                    ' Dauerlicht

Return
'-- End Menue_men3241 ------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men4 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_men4:

   ' Trace Output schreiben
   #if Menue_testmodus                                       ' Trace
      Print #2 , "Enter Menue_men4"
   #endif                                                    ' Trace

   ' Zeitüberwachung aus Untermenüs abschalten damit Menüs nicht mehr zyklisch
   ' aufgerufen werden.

   ' Flag für zeitlich upzudatendes Menue zurücksetzen
   Menue_insensmenue_flag = 0

   ' Abhängig vom Flash-Mode und von der Funktion lichtgesteuert zuzufahren
   ' Lichterfassung ausschalten
   If Dim_flash_mode = Passiv Then
      If Rel_auf_dunkel_flag = False Then
         Light_erfassung = False
      End If                                                 ' Lichtfunktion Beschattung AUF
   End If                                                    ' Flash-Mode

Return
'-- End Menue_men4 ------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men41 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
```

```
Menue_men41:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men41"
   #endif                                                 ' Trace

   ' Zeitüberwachung aus Untermenüs abschalten damit Menüs nicht mehr zyklisch
   ' aufgerufen werden.

   ' Flag für zeitlich upzudatendes Menue zurücksetzen
   Menue_insensmenue_flag = 0

Return
'-- End Menue_men41 ------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men411 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_men411:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men411"
   #endif                                                 ' Trace

   ' Zeitüberwachung aus Untermenüs einschalten damit Menüs zyklisch
   ' aufgerufen werden.

   ' Windgeschwindigkeit ausgeben!


   ' Flag für zeitlich upzudatendes Menue setzen
   Menue_insensmenue_flag = 1

   Temp_string_1 = Fusing(wind_geschwindigkeit_kmh , "###.##")
   Locate 3 , 1 : Lcd Temp_string_1 ; " km/h   "

   ' Achtung: 3 Leerezeichen bei der Ausgabe verhindern das Stehenbleiben von
   ' einzenen Buchstaben bei großen Schwankungen der Windgeschwindigkeit in
   ' einer Zeile. Auf diese Weise kan auf das Löschen der vorhergehenden Zeile
   ' verzichtet werden.

Return
'-- End Menue_men411 ------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men42 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_men42:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men42"
   #endif                                                 ' Trace

   ' Zeitüberwachung aus Untermenüs abschalten damit Menüs nicht mehr zyklisch
   ' aufgerufen werden.

   ' Flag für zeitlich upzudatendes Menue zurücksetzen
   Menue_insensmenue_flag = 0

Return
'-- End Menue_men42 ------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men421 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
```

```
Menue_men421:

   ' Trace Output schreiben
   #if Menue_testmodus                                          ' Trace
      Print #2 , "Enter Menue_men421"
   #endif                                                       ' Trace

   ' Zeitüberwachung aus Untermenüs einschalten damit Menüs zyklisch
   ' aufgerufen werden.
   ' Status Regensensor ausgeben!

   ' Flag für zeitlich upzudatendes Menue setzen
   Menue_insensmenue_flag = 2

   ' Funktion zur Abfrage des Regensensors aufrufen
   ' Rückgabewert: &H00 = kein Regen
   '               &H01 = Regen vom Sensor erkannt
   Temp_byte_1 = Rain_measurement();

   Locate 3 , 1

   ' Variable auswerten und Ausgabetext anzeigen
   Select Case Temp_byte_1
      Case &H00 : Lcd "trocken    "
      Case &H01 : Lcd "nass/feucht"
   End Select                                                   ' Regensensorwert

Return
'-- End Menue_men421 ------------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men4 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men43:

   ' Trace Output schreiben
   #if Menue_testmodus                                          ' Trace
      Print #2 , "Enter Menue_men43"
   #endif                                                       ' Trace

   ' Zeitüberwachung aus Untermenüs abschalten damit Menüs nicht mehr zyklisch
   ' aufgerufen werden.

   ' Flag für zeitlich upzudatendes Menue zurücksetzen
   Menue_insensmenue_flag = 0

Return
'-- End Menue_men43 -------------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men431 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men431:

   ' Trace Output schreiben
   #if Menue_testmodus                                          ' Trace
      Print #2 , "Enter Menue_men431"
   #endif                                                       ' Trace

   ' Zeitüberwachung aus Untermenüs einschalten damit Menüs zyklisch
   ' aufgerufen werden.
   ' Temperatur und Taupunkt ausgeben!

   ' Flag für zeitlich upzudatendes Menue setzen
   Menue_insensmenue_flag = 3

   ' Ausgabe der Temperatur
   Locate 2 , 7
   Temp_string_1 = Fusing(sht71_t , "###.##")
   Lcd Temp_string_1 ; "{223}C  "

   ' Ausgabe des Taupunktes
```

```
   Locate 3 , 7
   Temp_string_1 = Fusing(sht71_dewpoint , "###.##")
   Lcd Temp_string_1 ; "{223}C  "

Return
'-- End Menue_men431 ------------------------------------------------------


'------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men44 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'------------------------------------------------------------------------------
Menue_men44:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_men44"
   #endif                                                   ' Trace

   ' Zeitüberwachung aus Untermenüs abschalten damit Menüs nicht mehr zyklisch
   ' aufgerufen werden.

   ' Flag für zeitlich upzudatendes Menue zurücksetzen
   Menue_insensmenue_flag = 0

Return
'-- End Menue_men44 ------------------------------------------------------


'------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men441 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'------------------------------------------------------------------------------
Menue_men441:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_men441"
   #endif                                                   ' Trace

   ' Zeitüberwachung aus Untermenüs einschalten damit Menüs zyklisch
   ' aufgerufen werden.
   ' Luftfeuchtigkeit ausgeben!

   ' Abhängig vom Flash-Mode und von der Funktion lichtgesteuert zuzufahren
   ' Lichterfassung ausschalten
   If Dim_flash_mode = Passiv Then
      If Rel_auf_dunkel_flag = False Then
         Light_erfassung = False
      End If                                                ' Lichtfunktion Beschattung AUF
   End If                                                   ' Flash-Mode

   ' Flag für zeitlich upzudatendes Menue setzen
   Menue_insensmenue_flag = 4

   ' Ausgabe der relativen Luftfeuchtigkeit
   Locate 3 , 1
   Temp_string_1 = Fusing(sht71_rhtrue , "###.##")
   Lcd Temp_string_1 ; " %RH"

Return
'-- End Menue_men441 ------------------------------------------------------


'------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men45 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'------------------------------------------------------------------------------
Menue_men45:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_men45"
   #endif                                                   ' Trace
```

```
   ' Zeitüberwachung aus Untermenüs abschalten damit Menüs nicht mehr zyklisch
   ' aufgerufen werden.

   ' Flag für zeitlich upzudatendes Menue zurücksetzen
   Menue_insensmenue_flag = 0

   ' Abhängig vom Flash-Mode und von der Funktion lichtgesteuert zuzufahren
   ' Lichterfassung ausschalten
   If Dim_flash_mode = Passiv Then
      If Rel_auf_dunkel_flag = False Then
         Light_erfassung = False
      End If                                              ' Lichtfunktion Beschattung AUF
   End If                                                 ' Flash-Mode

Return
'-- End Menue_men45 ------------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men451 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men451:

   ' Trace Output schreiben
   #if Menue_testmodus                                   ' Trace
      Print #2 , "Enter Menue_men451"
   #endif                                                ' Trace

   ' Zeitüberwachung aus Untermenüs einschalten damit Menüs zyklisch
   ' aufgerufen werden.
   ' Helligkeitswert ausgeben

   ' Flag für zeitlich upzudatendes Menue setzen
   Menue_insensmenue_flag = 5

   ' Prüfen ob Lichterfassung ausgeschaltet
   If Light_erfassung = False Then
      ' JA, Lichterfassung ist aus

      ' Licht einmalig erfassen
      Light_value = Light_measurement(light_messwert_zahl )

      ' zyklisches Update aktivieren
      Light_erfassung = True

      ' Auswerteintervall setzen
      Light_erfassungs_counter = Sectic_count + Light_auswertung_default
   End If                                                ' Lichterfassung

   ' Ausgabe der vom Sensor TAOS gemessenen Helligkeit normiert auf 0-15
   Temp_byte_1 = Makebcd(light_value)
   Locate 2 , 15 : Lcd Bcd(temp_byte_1)

Return
'-- End Menue_men451 ------------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men461 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men461:

   ' Trace Output schreiben
   #if Menue_testmodus                                   ' Trace
      Print #2 , "Enter Menue_men461"
   #endif                                                ' Trace

   ' Maximale gemessene Windgeschwindigkeit ausgeben

   ' Flag für zeitlich upzudatendes Menue zurücksetzen
   ' Anmerkung: Aufruf könnte von Men451 mit Zeitüberwachung sein.
   Menue_insensmenue_flag = 0
```

```
   ' Ausgabe der aktuell gespeicherten maximalen Windgeschwindigkeit
   Temp_string_1 = Fusing(wind_kmh_max , "###.#")
   Locate 3 , 1 : Lcd Temp_string_1 ; " km/h"

Return
'-- End Menue_men461 ------------------------------------------------------


'------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men462 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'------------------------------------------------------------------------------
Menue_men462:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_men462"
   #endif                                                   ' Trace

   ' Minimale und maximale gemessene Temperaturen ausgeben

   ' Ausgabe der minimalen Temperatur
   Temp_string_1 = Fusing(sht71_t_min , "###.#")
   Locate 2 , 10 : Lcd Temp_string_1 ; "{223}C"

   ' Ausgabe der maximalen Temperatur
   Temp_string_1 = Fusing(sht71_t_max , "###.#")
   Locate 3 , 10 : Lcd Temp_string_1 ; "{223}C"

Return
'-- End Menue_men462 ------------------------------------------------------


'------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men463 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'------------------------------------------------------------------------------
Menue_men463:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_men463"
   #endif                                                   ' Trace

   ' Minimale und maximale gemessene Lufttfeuchtigkeit ausgeben

   ' Ausgabe der minimalen relativen Luftfeuchtigkeit
   Temp_string_1 = Fusing(sht71_rhtrue_min , "###.#")
   Locate 2 , 12 : Lcd Temp_string_1 ; "%"

   ' Ausgabe der maximalen relativen Luftfeuchtigkeit
   Temp_string_1 = Fusing(sht71_rhtrue_max , "###.#")
   Locate 3 , 12 : Lcd Temp_string_1 ; "%"

Return
'-- End Menue_men463 ------------------------------------------------------


'------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men5 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'------------------------------------------------------------------------------
Menue_men5:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_men5"
   #endif                                                   ' Trace

   ' Zeitüberwachung aus Untermenüs ausschalten damit Menüs nicht mehr zyklisch
   ' aufgerufen werden.

   ' Flag für zeitlich upzudatendes Menue zurücksetzen
   Menue_insensmenue_flag = 0
```

```
Return
'-- End Menue_men5 ------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men51 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_men51:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_men51"
   #endif                                                   ' Trace

   ' Zeitüberwachung aus Untermenüs einschalten damit Menüs zyklisch
   ' aufgerufen werden.

   ' Flag für zeitlich upzudatendes Menue setzen
   Menue_insensmenue_flag = 6

   ' Zeit und Datumsinformationen zyklisch ausgeben
   ' Uhrzeit und Tag ausgeben
   Locate 2 , 1 : Lcd Time$
   Locate 2 , 15 : Lcd Lookupstr(_weekday , Wochentag)

   ' Wenn Zeit synchronisiert wurde dann * ausgeben
   If Dcf_sync_count > 0 Then
      Locate 2 , 11 : Lcd "*"
   End If                                                   ' Zeit synchronisiert?

   ' Datum ausgeben
   Locate 3 , 1 : Lcd Date$

   ' Anzahl der erfolgten Synchronisationen auf LCD ausgeben
   Temp_byte_2 = Makebcd(dcf_sync_count)
   Locate 3 , 11 : Lcd Bcd(temp_byte_2)

   ' Zeitzone ausgeben
   Locate 3 , 16 : Lcd Dcf77timezone()

   ' Eine Sekunde warten!
   ' Wait 1 ' nicht hier!!

Return
'-- End Menue_men51 ------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men6111 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_men6111:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_men6111"
   #endif                                                   ' Trace

   ' Ausgabe der Datenversion Internes EEPROM

   Temp_word_1 = Akt_internal_ee_version
   Locate 3 , 1 : Lcd "&H" ; Hex(temp_word_1)

Return
'-- End Menue_men6111 ------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men6121 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_men6121:
```

```
   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men6121"
   #endif                                                 ' Trace

   ' Ausgabe der Datenversion Externes EEPROM
   Temp_word_1 = Akt_external_ee_version
   Locate 3 , 1 : Lcd "&H" ; Hex(temp_word_1)

Return
'-- End Menue_men6121 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men6122 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men6122:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men6122"
   #endif                                                 ' Trace

   ' Ausgabe Datencounter in externem EEPROM
   Locate 3 , 1 : Lcd "&H" ; Hex(ee_data_count) ; " = " ; Ee_data_count

Return
'-- End Menue_men6122 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men6123 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men6123:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men6123"
   #endif                                                 ' Trace

   ' Ausgabe Überlaufzähler externes EEPROM
   Locate 3 , 1 : Lcd "&H" ; Hex(ee_data_overrun) ; " = " ; Ee_data_overrun

Return
'-- End Menue_men6123 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men6124 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men6124:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men6124"
   #endif                                                 ' Trace

   ' Ausgabe der aktuellen Adresse im externen EEPROM

   Temp_word_1 = Ea_data_loging_length * Ee_data_count
   Temp_word_1 = Temp_word_1 + Ee_data_loging_start_adress

   Locate 3 , 1 : Lcd "&H" ; Hex(temp_word_1) ; " = " ; Temp_word_1

Return
'-- End Menue_men6124 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men621 (Menue-Unterstützung-Routine)
```

```
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men621:

   ' Trace Output schreiben
   #if Menue_testmodus                                          ' Trace
      Print #2 , "Enter Menue_men621"
   #endif                                                       ' Trace

   ' Loggingdaten übertragen

   Print #2 , "Logging-Daten ConvCtrl V1.0"
   Print #2 , "Uebertragung :;" ; Date$ ; ";" ; Time$
   Print #2 , ""
   Print #2 , "Datenversion :;" ; Hex(ee_data_version_extern)
   Print #2 , "Data Count :;" ; Ee_data_count
   Print #2 , "Data Overrun :;" ; Ee_data_overrun
   Print #2 , ""
   Print #2 , "Lfd.-Nr.;Datum;Zeit;Licht;Regen;Wind;Temperatur;Luftfeuchtigkeit"

   ' Gelesene Daten interpretieren und ausgeben
   ' Format: nr;day.month.year;hour:minute;light;rain;wind;temp;hum


   ' Prüfen auf Überlauf, wenn ja maximale Adresse annehmen
   If Ee_data_overrun > 0 Then
      Ee_output_endadress = Ea_data_loging_endadress - 1
   Else                                                         ' Überlauf
      Ee_output_endadress = Ee_data_count
   End If                                                       ' Überlauf

   ' Anmerkung: Wenn Überlauf stattgefunden hat kann gesamtes EEPROM ausgegeben
   ' werden. Andernfalls wird nur von Adress NULL bis zum aktuellen Datenzeiger
   ' ausgegeben.

   ' Prüfen ob überhaupt Daten im EEPROM abgelegt und wenn ja ausgeben
   If Ee_output_endadress > 0 Then

      ' Ausgabeflag setzen
      Ee_output_flag = True

      ' Ausgabecounter setzen
      Ee_output_counter = Ee_data_count_default

   Else                                                         ' Daten im EEPROM?

      Print #2 , "*** Fehler! keine Daten enthalten ***"
      Print #2 , "Ende der Daten"

   End If                                                       ' Daten im EEPROM?

Return
'-- End Menue_men621 ------------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men7111 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men7111:

   ' Trace Output schreiben
   #if Menue_testmodus                                          ' Trace
      Print #2 , "Enter Menue_men7111"
   #endif                                                       ' Trace

   ' Timerstatus für Öffnung ausgeben
   Locate 2 , 7

   Select Case Timer_open_flag
      Case Aktiv :
         Lcd "AN "
      Case Passiv :
         Lcd "AUS"
   End Select                                                   ' Timer-Open-Flag
```

```
Return
'-- End Menue_men7111 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men71111 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men71111:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men71111"
   #endif                                                 ' Trace

   ' Timerstatus für Öffnung ausgeben und Status umschalten
   Locate 2 , 7

   Select Case Timer_open_flag
      Case Aktiv :
         Timer_open_flag = Passiv
         Lcd "AUS"
      Case Passiv :
         Timer_open_flag = Aktiv
         Lcd " AN"
   End Select                                             ' Timer-Open-Flag

Return
'-- End Menue_men71111 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men7121 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men7121:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men7121"
   #endif                                                 ' Trace

   ' Timerstatus für Schließung ausgeben
   Locate 2 , 7

   Select Case Timer_close_flag
      Case Aktiv :
         Lcd "AN "
      Case Passiv :
         Lcd "AUS"
   End Select                                             ' Timer-Close-Flag

Return
'-- End Menue_men7121 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men71211 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men71211:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men71211"
   #endif                                                 ' Trace

   ' Timerstatus für Schließung ausgeben und Status umschalten
   Locate 2 , 7

   Select Case Timer_close_flag
      Case Aktiv :
         Timer_close_flag = Passiv
         Lcd "AUS"
```

```
      Case Passiv :
         Timer_close_flag = Aktiv
         Lcd " AN"
   End Select                                          ' Timer-Close-Flag

Return
'-- End Menue_men71211 ------------------------------------------------------


'------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men7211 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'------------------------------------------------------------------------------
Menue_men7211:

   ' Trace Output schreiben
   #if Menue_testmodus                                 ' Trace
      Print #2 , "Enter Menue_men7211"
   #endif                                              ' Trace

   ' ggf. Zeitwerte aus Unterfunktion übernehmen - Öffnungszeiten

   If Menue_datenuebernahme_flag = True Then

      ' Flag zurücksetzen
      Menue_datenuebernahme_flag = False

      ' Daten uebernehmen
      Timer_open_sekunde = Menue_time_sekunde          ' Interne Sekundenvariable
      Timer_open_minute = Menue_time_minute            ' Interne Minutenvariable
      Timer_open_stunde = Menue_time_stunde            ' Interne Stundenvariable

   End If                                              ' Datenübernahme?

Return
'-- End Menue_men7211 ---------------------------------------------------------


'------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men72111 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'------------------------------------------------------------------------------
Menue_men72111:

   ' Trace Output schreiben
   #if Menue_testmodus                                 ' Trace
      Print #2 , "Enter Menue_men72111"
   #endif                                              ' Trace

   ' Einstellen der Uhrzeit vorbereiten und relevante Variablen laden

   ' Uhrzeit laden wurde von Menü MEN7211 aufgerufen. Es geht hier um Öffnungszeit

   ' Hilfvariablen für Time-Menü laden

   Menue_time_sekunde = Timer_open_sekunde             ' Interne Sekundenvariable
   Menue_time_minute = Timer_open_minute               ' Interne Minutenvariable
   Menue_time_stunde = Timer_open_stunde               ' Interne Stundenvariable

   ' Rücksprung-Ziel festlegen
   Menue_state_buffer = Loadlabel(men7211)             ' Uhrzeit einstellen Menü 7-2-1-1

   ' Menü neu laden und einspringen
   Menue_state = Loadlabel(time1)                      ' Einsprung in erstes Einstellfenster
   Gosub Menue_change_state_by_sub
   Gosub Lcd_print
   Menue_state_renew = 0
   Gosub Menue_time1                                   ' zugehörige Unterfunktiona anspringen

Return
'-- End Menue_men72111 ------------------------------------------------------


'------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men7212 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
```

```
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------
Menue_men7212:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men7212"
   #endif                                                 ' Trace

   ' ggf. Datumswert aus Unterfunktion übernehmen - Öffnungszeiten
   If Menue_datenuebernahme_flag = True Then

      ' Flag zurücksetzen
      Menue_datenuebernahme_flag = False

      ' Daten üebernehmen
      Timer_open_tag = Menue_date_tag                     ' Interne Tagvariable
      Timer_open_monat = Menue_date_monat                 ' Interne Monatsvariable
      Timer_open_jahr = Menue_date_jahr                   ' Interne Jahresvariable

   End If                                                 ' Datenübernahme

Return
'-- End Menue_men7212 ---------------------------------------------------


'-------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men72121 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------
Menue_men72121:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men72121"
   #endif                                                 ' Trace

   ' Einstellen des Datums vorbereiten und relevante Variablen laden

   ' Datum laden wurde von Menü MEN7212 aufgerufen. Es geht hier um Öffnungszeit

   ' Hilfvariablen für Time-Menü laden

   Menue_date_tag = Timer_open_tag                        ' Interne Tagvariable der SoftClock
   Menue_date_monat = Timer_open_monat                    ' Interne Monatsvariable der SoftClock
   Menue_date_jahr = Timer_open_jahr                      ' Interne Jahresvariable der Softclock

   ' Rücksprung-Ziel festlegen
   Menue_state_buffer = Loadlabel(men7212)                ' Datum einstellen Menü 7-2-1-2

   ' Menü neu laden und einspringen
   Menue_state = Loadlabel(date1)                         ' Einsprung in erstes Einstellfenster
   Gosub Menue_change_state_by_sub
   Gosub Lcd_print
   Menue_state_renew = 0
   Gosub Menue_date1                                      ' zugehörige Unterfunktion anspringen

Return
'-- End Menue_men72121 --------------------------------------------------


'-------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men7221 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------
Menue_men7221:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men7221"
   #endif                                                 ' Trace

   ' ggf. Zeitwerte aus Unterfunktion übernehmen - Schließungszeiten
      If Menue_datenuebernahme_flag = True Then
```

```
      ' Flag zurücksetzen
      Menue_datenuebernahme_flag = False

      ' Daten übernehmen
      Timer_close_sekunde = Menue_time_sekunde              ' Interne Sekundenvariable
      Timer_close_minute = Menue_time_minute                ' Interne Minutenvariable
      Timer_close_stunde = Menue_time_stunde                ' Interne Stundenvariable

   End If                                                   ' Datenübernahme

Return
'-- End Menue_men7221 ------------------------------------------------------


'------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men72211 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'------------------------------------------------------------------------------
Menue_men72211:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_men72211"
   #endif                                                   ' Trace

   ' Einstellen der Uhrzeit vorbereiten und relevante Variablen laden

   ' Uhrzeit laden wurde von Menü MEN7221 aufgerufen. Es geht hier um Schließungszeit

   ' Hilfvariablen für Time-Menü laden

   Menue_time_sekunde = Timer_close_sekunde                 ' Interne Sekundenvariable
   Menue_time_minute = Timer_close_minute                   ' Interne Minutenvariable
   Menue_time_stunde = Timer_close_stunde                   ' Interne Stundenvariable

   ' Rücksprung-Ziel festlegen
   Menue_state_buffer = Loadlabel(men7221)                  ' Uhrzeit einstellen Menü 7-2-2-1

   ' Menü neu laden und einspringen
   Menue_state = Loadlabel(time1)                           ' Einsprung in erstes Einstellfenster
   Gosub Menue_change_state_by_sub
   Gosub Lcd_print
   Menue_state_renew = 0
   Gosub Menue_time1                                        ' zugehörige Unterfunktion anspringen

Return
'-- End Menue_men72211 ------------------------------------------------------


'------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men7222 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'------------------------------------------------------------------------------
Menue_men7222:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_men7222"
   #endif                                                   ' Trace

   ' ggf. Datumswert aus Unterfunktion übernehmen - Schließungszeiten
   If Menue_datenuebernahme_flag = True Then

      ' Flag zurücksetzen
      Menue_datenuebernahme_flag = False

      ' Daten übernehmen
      Timer_close_tag = Menue_date_tag                      ' Interne Tagvariable
      Timer_close_monat = Menue_date_monat                  ' Interne Monatsvariable
      Timer_close_jahr = Menue_date_jahr                    ' Interne Jahresvariable

   End If                                                   ' Datenübernahme

Return
'-- End Menue_men7222 ------------------------------------------------------
```

```
'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men72221 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men72221:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men72221"
   #endif                                                 ' Trace

   ' Einstellen des Datums vorbereiten und relevante Variablen laden

   ' Datum laden wurde von Menü MEN882 aufgerufen. Es geht hier um Schließungszeit

   ' Hilfvariablen für Time-Menü laden

   Menue_date_tag = Timer_close_tag                       ' Interne Tagvariable
   Menue_date_monat = Timer_close_monat                   ' Interne Monatsvariable
   Menue_date_jahr = Timer_close_jahr                     ' Interne Jahresvariable

   ' Rücksprung-Ziel festlegen
   Menue_state_buffer = Loadlabel(men7222)                ' Datum einstellen Menü 7-2-2-2

   ' Menü neu laden und einspringen
   Menue_state = Loadlabel(date1)                         ' Einsprung in erstes Einstellfenster
   Gosub Menue_change_state_by_sub
   Gosub Lcd_print
   Menue_state_renew = 0
   Gosub Menue_date1                                      ' zugehörige Unterfunktion anspringen

Return
'-- End Menue_men72121 ------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men81 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men81:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men81"
   #endif                                                 ' Trace

   ' Ggf. Dimmereinstellungen in EEPROM zurückspeichern

   If Menue_data_changed_flag = True Then

      ' Änderungs-Flag zurücksetzen
      Menue_data_changed_flag = False

      ' Daten aus Dimmermenü in EEPROM zurückschreiben
      Ee_data_dimmer = Dim_flash_mode
      Ee_data_dim_min = Dim_helligkeit_min
      Ee_data_dim_max = Dim_helligkeit_max

      ' Speicherung durch EE auf Display anzeigen
      Locate 3 , 8 : Lcd "EE"

   End If                                                 ' Datenänderung

Return
'-- End Menue_men81 ----------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8111 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men8111:
```

```
   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men8111"
   #endif                                                 ' Trace

   ' Zustand Lightfloating ausgeben
   ' Wertebereich AN/AUS

   Locate 2 , 7
   Select Case Dim_flash_mode
      Case Aktiv:
         Lcd "AN"
      Case Passiv:
         Lcd "AUS"
   End Select                                             ' Flash-Mode

Return
'-- End Menue_men8111 ------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men81111 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men81111:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men81111"
   #endif                                                 ' Trace

   ' Zustand Lightfloating umschalten und Zustand ausgeben
   ' Wertebereich AN/AUS

   Menue_data_changed_flag = True

   Locate 2 , 7
   Select Case Dim_flash_mode
      Case Aktiv:
         Dim_flash_mode = Passiv
         Lcd "AUS"
      Case Passiv:
         Dim_flash_mode = Aktiv
         Lcd "AN"
   End Select                                             ' Flash-Mode

Return
'-- End Menue_men81111 ------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men81211 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men81211:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men81211"
   #endif                                                 ' Trace

   ' Dimmer untere Grenze ausgeben
   ' Wertebereich 01-32

   Temp_byte_1 = Makebcd(dim_helligkeit_min)
   Locate 2 , 8 : Lcd Bcd(temp_byte_1)

Return
'-- End Menue_men81211 ------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men812111 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
```

```
'-------------------------------------------------------------------------------
Menue_men812111:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men812111"
   #endif                                                 ' Trace

   ' Dimmer untere Grenze berechnen und neu ausgeben
   ' Wertebereich 01-32 aber gegen obere Grenze verriegelt

   Menue_data_changed_flag = True

   ' Tasten auswerten welche betätigt wurden
   Select Case Key_taste_buffer

      Case 12:
         ' hoch = erhöhen
         Temp_byte_1 = Dim_helligkeit_max - 1
         If Dim_helligkeit_min < Temp_byte_1 Then
            Incr Dim_helligkeit_min
         End If                                           ' Helligkeitsstufe

      Case 42:
         ' runter = erniedrigen
         If Dim_helligkeit_min > Dim_helligkeit_set_min Then
            Decr Dim_helligkeit_min
         End If                                           ' Helligkeitsstufe

      Case 44:
         'rechts = erhöhen und bei null anfangen
         Incr Dim_helligkeit_min
         If Dim_helligkeit_min >= Dim_helligkeit_max Then
            Dim_helligkeit_min = Dim_helligkeit_set_min
         End If                                           ' Helligkeitsstufe

   End Select                                             ' Tastenauswertung

   ' Neuen Helligkeitswert ausgeben
   Temp_byte_1 = Makebcd(dim_helligkeit_min)
   Locate 2 , 8 : Lcd Bcd(temp_byte_1)

   ' Prüfen ob aktuell eingestellter Helligkeitswert verschoben werden muss
   ' => ggf. Ausgabe eines Sternchens!

   ' Prüfen ob Verschiebung bei unterer Grenze?
   If Dim_helligkeitsstufe < Dim_helligkeit_min Then
      ' eingestellte Helligkeit auf untere Grenze setzen
      Dim_helligkeitsstufe = Dim_helligkeit_min
      ' Stern zur Kennzeichnug ausgeben
      Locate 2 , 16 : Lcd "*"
   End If                                                 ' Helligkeitsstufen

Return
'-- End Menue_men812111 -------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men81221 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men81221:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men81221"
   #endif                                                 ' Trace

   ' Dimmer obere Grenze ausgeben
   ' Wertebereich 01-32

   Temp_byte_1 = Makebcd(dim_helligkeit_max)
   Locate 2 , 8 : Lcd Bcd(temp_byte_1)

Return
'-- End Menue_men81221 -------------------------------------------------------
```

```
'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men812211 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men812211:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men812211"
   #endif                                                 ' Trace

   ' Dimmer obere Grenze berechnen und neu ausgeben
   ' Wertebereich 01-32 aber gegen untere Grenze verriegelt.

   Menue_data_changed_flag = True

   ' Tasten auswerten welche betätigt wurden
   Select Case Key_taste_buffer

      Case 12:
         ' hoch = erhöhen
         If Dim_helligkeit_max < Dim_helligkeit_set_max Then
            Incr Dim_helligkeit_max
         End If                                           ' Helligkeitsstufe

      Case 42:
         ' runter = erniedrigen
         Temp_byte_1 = Dim_helligkeit_min + 1
         If Dim_helligkeit_max > Temp_byte_1 Then
            Decr Dim_helligkeit_max
         End If                                           ' Helligkeitsstufe

      Case 44:
         'rechts = erhöhen und bei null anfangen
         Incr Dim_helligkeit_max
         If Dim_helligkeit_max > Dim_helligkeit_set_max Then
            Dim_helligkeit_max = Dim_helligkeit_min + 1
         End If                                           ' Helligkeitsstufe

   End Select                                             ' Tastenauswertung

   ' Neuen Helligkeitswert ausgeben
   Temp_byte_1 = Makebcd(dim_helligkeit_max)
   Locate 2 , 8 : Lcd Bcd(temp_byte_1)

   ' Prüfen ob aktuell eingestellter Helligkeitswert verschoben werden muss
   ' => ggf. Ausgabe eines Sternchens!

   ' Prüfen ob Verschiebung bei oberer Grenze?
   If Dim_helligkeitsstufe > Dim_helligkeit_max Then
      ' eingestellte Helligkeit auf obere Grenze setzen
      Dim_helligkeitsstufe = Dim_helligkeit_max
      ' Stern zur Kennzeichnug ausgeben
      Locate 2 , 16 : Lcd "*"
   End If                                                 ' Helligkeitsstufe

Return
'-- End Menue_men812211 -------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men82 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men82:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men82"
   #endif                                                 ' Trace

   ' Ggf. Tastatureinstellungen in EEPROM zurückspeichern

   If Menue_data_changed_flag = True Then

      ' Änderungsflag zurücksetzen
```

```
      Menue_data_changed_flag = False

      ' Einstellungen aus Tastatur-Menü in EEPROM schreiben
      Ee_data_conv_hotkey = Key_shortcut_status

      ' Speicherung durch EE auf Display anzeigen
      Locate 3 , 8 : Lcd "EE"

   End If                                                  ' Daten verändert?

Return
'-- End Menue_men82 ------------------------------------------------------


'--------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8211 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------
Menue_men8211:

   ' Trace Output schreiben
   #if Menue_testmodus                                     ' Trace
      Print #2 , "Enter Menue_men8211"
   #endif                                                  ' Trace

   ' Status Enter ShortCut ausgeben
   ' Wertebereich EIN/AUS

   Locate 2 , 8
   Select Case Key_shortcut_status
      Case Aktiv:
         Lcd "AN"
      Case Passiv:
         Lcd "AUS"
   End Select                                              ' Short_Cut aktiv?

Return
'-- End Menue_men8211 ----------------------------------------------------


'--------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men82111 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------
Menue_men82111:

   ' Trace Output schreiben
   #if Menue_testmodus                                     ' Trace
      Print #2 , "Enter Menue_men82111"
   #endif                                                  ' Trace

   ' Status Enter ShortCut umschalten und Status ausgeben
   ' Wertebereich EIN/AUS

   Menue_data_changed_flag = True

   Locate 2 , 8
   Select Case Key_shortcut_status
      Case Aktiv:
         Key_shortcut_status = Passiv
         Lcd "AUS"
      Case Passiv:
         Key_shortcut_status = Aktiv
         Lcd "AN"
   End Select                                              ' Short-Cut aktiv?

Return
'-- End Menue_men82111 ----------------------------------------------------


'--------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men83 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------
```

```
Menue_men83:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men83"
   #endif                                                 ' Trace

   ' Ggf. Displayeinstellungen in EEPROM zurückspeichern

   If Menue_data_changed_flag = True Then

      ' Änderungsflag zurücksetzen
      Menue_data_changed_flag = False

      ' Einstellungen aus Display-Menü in EEPROM schreiben
      Ee_data_lcd_helligkeit = Lcd_helligkeit
      Ee_data_lcd_kontrast = Lcd_kontrastwert
      Ee_data_lcd_saver = Lcd_screensaver

      ' Speicherung durch EE auf Display anzeigen
      Locate 3 , 8 : Lcd "EE"

   End If                                                 ' Daten verändert?

Return
'-- End Menue_men83 ------------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8311 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men8311:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men8311"
   #endif                                                 ' Trace

   ' Status ScreenSaver ausgeben
   ' Wertebereich EIN/AUS

   Locate 2 , 8
   Select Case Lcd_screensaver
      Case True:
         Lcd "AN"
      Case False:
         Lcd "AUS"
   End Select                                             ' Screensaver aktiv?

Return
'-- End Menue_men8311 ----------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men83111 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men83111:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men83111"
   #endif                                                 ' Trace

   ' Status ScreenSaver umschalten und ausgeben
   ' Wertebereich EIN/AUS

   Menue_data_changed_flag = True

   Locate 2 , 8
   Select Case Lcd_screensaver

      Case True:
         Lcd_screensaver = False
         Lcd "AUS"
```

```
         Lcd_screensaver_aktiv = False

         ' Ggf. Trace ausgeben
         #if Screensaver_testmodus                          ' Trace
            Print #2 , "Screensaver ausgeschaltet"
         #endif                                             ' Trace

      Case False:
         Lcd_screensaver = True
         Lcd "AN"
         Lcd_screensaver_counter = Sectic_count + Lcd_screensaver_delay

         ' Ggf. Trace ausgeben
         #if Screensaver_testmodus                          ' Trace
            Print #2 , "Screensaver eingeschaltet"
         #endif                                             ' Trace

   End Select                                               ' Screensaver aktiv

Return
'-- End Menue_men83111 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8321 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men8321:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_men8321"
   #endif                                                   ' Trace

   ' Eingestellter Kontrast für LCD-Display ausgeben
   ' Wertebereich 00-64

   ' Kontrast auf Display ausgeben
   Temp_byte_1 = Makebcd(lcd_kontrastwert)
   Locate 2 , 8 : Lcd Bcd(temp_byte_1)

Return
'-- End Menue_men8321 --------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men83211 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men83211:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_men83211"
   #endif                                                   ' Trace

   ' Eingestellter Kontrast für LCD-Display ändern und ausgeben
   ' Wertebereich 00-64

   Menue_data_changed_flag = True

   ' Tasten auswerten welche betätigt wurden
   Select Case Key_taste_buffer

      Case 12:
         ' hoch = erhöhen
         If Lcd_kontrastwert < Lcd_kontrast_max Then
            Incr Lcd_kontrastwert
         End If                                             ' Kontrastwert

      Case 42:
         ' runter = erniedrigen
         If Lcd_kontrastwert > Lcd_kontrast_min Then
            Decr Lcd_kontrastwert
         End If                                             ' Kontrastwert
```

```
      Case 44:
         'rechts = erhöhen und bei null anfangen
         Incr Lcd_kontrastwert
         If Lcd_kontrastwert > Lcd_kontrast_max Then
            Lcd_kontrastwert = Lcd_kontrast_min
         End If                                                ' Kontrastwert

   End Select                                                  ' Tastenauswertung

   ' Kontrast neu setzen
   Gosub Lcd_kontrastset

   ' Kontrast auf Display ausgeben
   Temp_byte_1 = Makebcd(lcd_kontrastwert)
   Locate 2 , 8 : Lcd Bcd(temp_byte_1)

Return
'-- End Menue_men83211 ------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8331 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_men8331:

   ' Trace Output schreiben
   #if Menue_testmodus                                         ' Trace
      Print #2 , "Enter Menue_men8331"
   #endif                                                      ' Trace

   ' Eingestellte Helligkeit für LCD-Display ausgeben
   ' Wertebereich 000-255

   ' Helligkeit des Display aisgeben
   Temp_string_1 = Byte2string(lcd_helligkeit , &H30 )         ' String mit 0 auffüllen
   Locate 2 , 8 : Lcd Temp_string_1

Return
'-- End Menue_men8331 ------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men83311 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_men83311:

   ' Trace Output schreiben
   #if Menue_testmodus                                         ' Trace
      Print #2 , "Enter Menue_men83311"
   #endif                                                      ' Trace

   ' Eingestellte Helligkeit für LCD-Display ändern ausgeben
   ' Wertebereich 000-255

   Menue_data_changed_flag = True

   ' Tasten auswerten welche betätigt wurden
   Select Case Key_taste_buffer

      Case 12:
         ' hoch = heller
         If Lcd_pwmwert > Lcd_helligkeit_min Then
            Lcd_pwmwert = Lcd_pwmwert - Lcd_schrittweite_helligkeit
         End If                                                ' LCD Helligkeit

      Case 42:
         ' runter = dunkler
         If Lcd_pwmwert < Lcd_helligkeit_max Then
            Lcd_pwmwert = Lcd_pwmwert + Lcd_schrittweite_helligkeit
         End If                                                ' LCD-Helligkeit

      Case 44:
         'rechts = heller und bei null anfangen
         If Lcd_pwmwert > Lcd_helligkeit_min Then
```

```
            Lcd_pwmwert = Lcd_pwmwert - Lcd_schrittweite_helligkeit
         Else                                                    ' LCD-Helligkeit
            Lcd_pwmwert = Lcd_helligkeit_max
         End If                                                  ' LCD-Helligkeit

   End Select                                                    ' Tastenauswertung

   ' PWM-Wert in Timerregister laden
   Pwm3a = Lcd_pwmwert
   Lcd_helligkeit = 255 - Lcd_pwmwert

   ' Helligkeit des Display aisgeben
   Temp_string_1 = Byte2string(lcd_helligkeit , &H30 )      ' String mit 0 auffüllen
   Locate 2 , 8 : Lcd Temp_string_1

Return
'-- End Menue_men83311 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men84 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men84:

   ' Trace Output schreiben
   #if Menue_testmodus                                           ' Trace
      Print #2 , "Enter Menue_men84"
   #endif                                                        ' Trace

   ' Ggf. Lüftereinstellungen in EEPROM zurückspeichern

   If Menue_data_changed_flag = True Then

      ' Änderungsflag zurücksetzen
      Menue_data_changed_flag = False

      ' Einstellungen aus Lüftermenu in EEPROM schreiben
      Ee_data_fan = Fan_sw_betrieb
      Ee_data_fan_25_temp = Fan_25_temperatur
      Ee_data_fan_50_temp = Fan_50_temperatur
      Ee_data_fan_75_temp = Fan_75_temperatur
      Ee_data_fan_100_temp = Fan_100_temperatur

      ' Speicherung durch EE auf Display anzeigen
      Locate 3 , 8 : Lcd "EE"

   End If                                                        ' Daten geändert?

Return
'-- End Menue_men84 ----------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8411 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men8411:

   ' Trace Output schreiben
   #if Menue_testmodus                                           ' Trace
      Print #2 , "Enter Menue_men8411"
   #endif                                                        ' Trace

   ' Status Lüfter Betriebsmodus ausgeben
   ' Wertebereich AN / AUS

   Locate 2 , 8

   Select Case Fan_sw_betrieb
      Case True:
         Lcd "AN"
      Case False:
         Lcd "AUS"
   End Select                                                    ' Lüfterbetrieb?
```

```
Return
'-- End Menue_men8411 ------------------------------------------------------


'------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men84111 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'------------------------------------------------------------------------------
Menue_men84111:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men84111"
   #endif                                                 ' Trace

   ' Status Lüfter Betriebsmodus umschalten und ausgeben
   ' Wertebereich AN / AUS

   Menue_data_changed_flag = True

   Locate 2 , 8
   Select Case Fan_sw_betrieb

      Case True:
         Fan_sw_betrieb = False
         Lcd "AUS"
         ' Auf jeden Fall PWM-Wert auf AUS setzen!
         Call Fan_set_geschwindigkeit(fan_belueftung_aus)

      Case False:
         Fan_sw_betrieb = True
         Lcd "AN"

   End Select                                             ' Lüfterbetrieb?

Return
'-- End Menue_men84111 ------------------------------------------------------


'------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8421 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'------------------------------------------------------------------------------
Menue_men8421:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men8421"
   #endif                                                 ' Trace

   ' Temperatureinstellung für Lüfter 25% ausgeben
   ' Wertebereich 0 bis 100 °C

   ' Temperatur ausgeben
   Locate 2 , 7
   Temp_byte_1 = Makebcd(fan_25_temperatur)
   Lcd Bcd(temp_byte_1) ; "{223}C"

Return
'-- End Menue_men8421 ------------------------------------------------------


'------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men84211 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'------------------------------------------------------------------------------
Menue_men84211:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men84211"
   #endif                                                 ' Trace

   ' Temperatureinstellung für Lüfter 25% verändern, auf Grenzen prüfen und
```

```
   ' ausgeben. Wertebereich 0 bis 100 °C

   Menue_data_changed_flag = True

   ' Tasten auswerten welche betätigt wurden
   Temp_byte_1 = Fan_50_temperatur - 1

   Select Case Key_taste_buffer

      Case 12:
         ' hoch = erhöhen
         If Fan_25_temperatur < Temp_byte_1 Then
            Incr Fan_25_temperatur
         End If                                              ' Temperatur 25%

      Case 42:
         ' runter = erniedrigen
         If Fan_25_temperatur > Fan_min_temp Then
            Decr Fan_25_temperatur
         End If                                              ' Temperatur 25%

      Case 44:
         'rechts = erhöhen und bei null anfangen
         Incr Fan_25_temperatur
         If Fan_25_temperatur > Temp_byte_1 Then
            Fan_25_temperatur = Fan_min_temp
         End If                                              ' Temperatur 25%

   End Select                                                ' Tastenauswertung

   ' Temperatur ausgeben
   Locate 2 , 7
   Temp_byte_1 = Makebcd(fan_25_temperatur)
   Lcd Bcd(temp_byte_1) ; "{223}C"

Return
'-- End Menue_men84211 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8431 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men8431:

   ' Trace Output schreiben
   #if Menue_testmodus                                       ' Trace
      Print #2 , "Enter Menue_men8431"
   #endif                                                    ' Trace

   ' Temperatureinstellung für Lüfter 50% ausgeben
   ' Wertebereich 0 bis 100 °C

   ' Temperatur ausgeben
   Locate 2 , 7
   Temp_byte_1 = Makebcd(fan_50_temperatur)
   Lcd Bcd(temp_byte_1) ; "{223}C"

Return
'-- End Menue_men8431 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men84311 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men84311:

   ' Trace Output schreiben
   #if Menue_testmodus                                       ' Trace
      Print #2 , "Enter Menue_men84311"
   #endif                                                    ' Trace

   ' Temperatureinstellung für Lüfter 50% verändern, auf Grenzen prüfen und
   ' ausgeben. Wertebereich 0 bis 100 °C
```

```
   Menue_data_changed_flag = True

   ' Tasten auswerten welche betätigt wurden
   Temp_byte_1 = Fan_25_temperatur + 1
   Temp_byte_2 = Fan_75_temperatur - 1

   Select Case Key_taste_buffer

      Case 12:
         ' hoch = erhöhen
         If Fan_50_temperatur < Temp_byte_2 Then
            Incr Fan_50_temperatur
         End If                                          ' Temperatur 50%

      Case 42:
         ' runter = erniedrigen
         If Fan_50_temperatur > Temp_byte_1 Then
            Decr Fan_50_temperatur
         End If                                          ' Temperatur 50%

      Case 44:
         'rechts = erhöhen und bei null anfangen
         Incr Fan_50_temperatur
         If Fan_50_temperatur > Temp_byte_2 Then
            Fan_50_temperatur = Temp_byte_1
         End If                                          ' Temperatur 50%

   End Select                                            ' Tastenauswertung

   ' Temperatur ausgeben
   Locate 2 , 7
   Temp_byte_1 = Makebcd(fan_50_temperatur)
   Lcd Bcd(temp_byte_1) ; "{223}C"

Return
'-- End Menue_men84311 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8441 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men8441:

   ' Trace Output schreiben
   #if Menue_testmodus                                 ' Trace
      Print #2 , "Enter Menue_men8441"
   #endif                                              ' Trace

   ' Temperatureinstellung für Lüfter 75% ausgeben
   ' Wertebereich 0 bis 100 °C

   ' Temperatur ausgeben
   Locate 2 , 7
   Temp_byte_1 = Makebcd(fan_75_temperatur)
   Lcd Bcd(temp_byte_1) ; "{223}C"

Return
'-- End Menue_men8441 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men84411 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men84411:

   ' Trace Output schreiben
   #if Menue_testmodus                                 ' Trace
      Print #2 , "Enter Menue_men84411"
   #endif                                              ' Trace

   ' Temperatureinstellung für Lüfter 75% verändern, auf Grenzen prüfen und
   ' ausgeben. Wertebereich 0 bis 100 °C

   Menue_data_changed_flag = True
```

```
   ' Tasten auswerten welche betätigt wurden
   Temp_byte_1 = Fan_50_temperatur + 1
   Temp_byte_2 = Fan_100_temperatur - 1

   Select Case Key_taste_buffer

      Case 12:
         ' hoch = erhöhen
         If Fan_75_temperatur < Temp_byte_2 Then
            Incr Fan_75_temperatur
         End If                                          ' Temperatur 75%

      Case 42:
         ' runter = erniedrigen
         If Fan_75_temperatur > Temp_byte_1 Then
            Decr Fan_75_temperatur
         End If                                          ' Temperatur 75%

      Case 44:
         'rechts = erhöhen und bei null anfangen
         Incr Fan_75_temperatur
         If Fan_75_temperatur > Temp_byte_2 Then
            Fan_75_temperatur = Temp_byte_1
         End If                                          ' Temperatur 75%

   End Select                                            ' Tastenauswertung

   ' Temperatur ausgeben
   Locate 2 , 7
   Temp_byte_1 = Makebcd(fan_75_temperatur)
   Lcd Bcd(temp_byte_1) ; "{223}C"

Return
'-- End Menue_men84411 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8451 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men8451:

   ' Trace Output schreiben
   #if Menue_testmodus                                   ' Trace
      Print #2 , "Enter Menue_men8451"
   #endif                                                ' Trace

   ' Temperatureinstellung für Lüfter 100% ausgeben
   ' Wertebereich 0 bis 100 °C

   ' Temperatur ausgeben
   Locate 2 , 7
   Temp_byte_1 = Makebcd(fan_100_temperatur)
   Lcd Bcd(temp_byte_1) ; "{223}C"

Return
'-- End Menue_men8451 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men84511 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men84511:

   ' Trace Output schreiben
   #if Menue_testmodus                                   ' Trace
      Print #2 , "Enter Menue_men84511"
   #endif                                                ' Trace

   ' Temperatureinstellung für Lüfter 100% verändern, auf Grenzen prüfen und
   ' ausgeben. Wertebereich 0 bis 100 °C

   Menue_data_changed_flag = True
```

```
   ' Tasten auswerten welche betätigt wurden
   Temp_byte_1 = Fan_75_temperatur + 1

   Select Case Key_taste_buffer

      Case 12:
         ' hoch = erhöhen
         If Fan_100_temperatur < Fan_max_temp Then
            Incr Fan_100_temperatur
         End If                                              ' Temperatur 100%

      Case 42:
         ' runter = erniedrigen
         If Fan_100_temperatur > Temp_byte_1 Then
            Decr Fan_100_temperatur
         End If                                              ' Temperatur 100%

      Case 44:
         'rechts = erhöhen und bei null anfangen
         Incr Fan_100_temperatur
         If Fan_100_temperatur > Fan_max_temp Then
            Fan_100_temperatur = Temp_byte_1
         End If                                              ' Temperatur 100%

   End Select                                                ' Tastenauswertung

   ' Temperatur ausgeben
   Locate 2 , 7
   Temp_byte_1 = Makebcd(fan_100_temperatur)
   Lcd Bcd(temp_byte_1) ; "{223}C"

Return
'-- End Menue_men84511 ------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men85 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_men85:

   ' Trace Output schreiben
   #if Menue_testmodus                                       ' Trace
      Print #2 , "Enter Menue_men85"
   #endif                                                    ' Trace

   ' Ggf. Windeinstellungen in EEPROM zurückspeichern

   If Menue_data_changed_flag = True Then

      ' Änderungsflag zurücksetzen
      Menue_data_changed_flag = False

      ' Einstellungen aus Windmenü in EEPROM schreiben
      Ee_data_wind_intervall = Wind_auswertung
      Ee_data_wind_speed = Wind_beschattung_auf

      ' Speicherung durch EE auf Display anzeigen
      Locate 3 , 8 : Lcd "EE"

   End If                                                    ' Daten geändert?

Return
'-- End Menue_men85 --------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8511 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_men8511:

   ' Trace Output schreiben
   #if Menue_testmodus                                       ' Trace
      Print #2 , "Enter Menue_men8511"
   #endif                                                    ' Trace
```

```
   ' Einstellung Windgeschwindigkeit für automatisches Auffahren ausgeben
   ' Wertebereich 1-50 km/h

   ' Windgeschwindigkeit bei der die Beschattung automatisch geöffnet wird ausgeben
   Temp_byte_1 = Makebcd(wind_beschattung_auf)
   Locate 2 , 5 : Lcd ">" ; Bcd(temp_byte_1) ; " km/h"

Return
'-- End Menue_men8511 ------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men85111 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_men85111:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men85111"
   #endif                                                 ' Trace

   ' Einstellung Windgeschwindigkeit für automatisches Auffahren eingeben
   ' und ausgeben. Wertebereich 1-50 km/h

   Menue_data_changed_flag = True

   Select Case Key_taste_buffer

      Case 12:
         ' hoch = erhöhen
         If Wind_beschattung_auf < Wind_beschattung_max Then
            Incr Wind_beschattung_auf
         End If                                           ' Windgeschwindigkeit

      Case 42:
         ' runter = erniedrigen
         If Wind_beschattung_auf > Wind_beschattung_min Then
            Decr Wind_beschattung_auf
         End If                                           ' Windgeschwindigkeit

      Case 44:
         'rechts = erhöhen und bei null anfangen
         Incr Wind_beschattung_auf
         If Wind_beschattung_auf > Wind_beschattung_max Then
            Wind_beschattung_auf = Wind_beschattung_min
         End If                                           ' Windgeschwindigkeit

   End Select                                             ' Tastenauswertung

   ' Windgeschwindigkeit bei der die Beschattung automatisch geöffnet wird ausgeben
   Temp_byte_1 = Makebcd(wind_beschattung_auf)
   Locate 2 , 5 : Lcd ">" ; Bcd(temp_byte_1) ; " km/h"

Return
'-- End Menue_men85111 ------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8521 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_men8521:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men8521"
   #endif                                                 ' Trace

   ' Einstellung Messintervall für Windgeschwindigkeit ausgeben.
   ' Wertebereich 1 bis 30 Sekunden.

   ' Messintervall für Windgeschwindigkeit ausgeben
   Temp_byte_1 = Makebcd(wind_auswertung)
   Locate 2 , 6 : Lcd Bcd(temp_byte_1) ; " sec"
```

```
Return
'-- End Menue_men8521 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men85211 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men85211:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_men85211"
   #endif                                                   ' Trace

   ' Einstellung Messintervall für Windgeschwindigkeit verändern und ausgeben.
   ' Wertebereich 1 bis 30 Sekunden.

   Menue_data_changed_flag = True

   Select Case Key_taste_buffer

      Case 12:
         ' hoch = erhöhen
         If Wind_auswertung < Wind_auswertung_max Then
            Incr Wind_auswertung
         End If                                             ' Windgeschwindigkeit

      Case 42:
         ' runter = erniedrigen
         If Wind_auswertung > Wind_auswertung_min Then
            Decr Wind_auswertung
         End If                                             ' Windgeschwindigkeit

      Case 44:
         'rechts = erhöhen und bei null anfangen
         Incr Wind_auswertung
         If Wind_auswertung > Wind_auswertung_max Then
            Wind_auswertung = Wind_auswertung_min
         End If                                             ' Windgeschwindigkeit

   End Select                                               ' Tastenauswertung

   ' Messintervall für Windgeschwindigkeit ausgeben
   Temp_byte_1 = Makebcd(wind_auswertung)
   Locate 2 , 6 : Lcd Bcd(temp_byte_1) ; " sec"

Return
'-- End Menue_men85211 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men86 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men86:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_men86"
   #endif                                                   ' Trace

   ' Ggf. Temperatureinstellungen in EEPROM zurückspeichern

   If Menue_data_changed_flag = True Then

      ' Änderungsflag zurücksetzen
      Menue_data_changed_flag = False

      ' Einstellungen aus Temperaturmenü in EEPROM schreiben
      Ee_data_temp_intervall = Sht71_auswertung

      ' Speicherung durch EE auf Display anzeigen
      Locate 3 , 8 : Lcd "EE"
```

```
   End If                                                ' Daten geändert?

Return
'-- End Menue_men86 ------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8611 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_men8611:

   ' Trace Output schreiben
   #if Menue_testmodus                                  ' Trace
      Print #2 , "Enter Menue_men8611"
   #endif                                               ' Trace

   ' Einstellung Messintervall für Temperaturmessungen ausgeben.
   ' Wertebereich 1 bis 30 Sekunden.

   ' Auswerteintervall für Temperatur und Luftfeuchtigkeit ausgeben
   Temp_byte_1 = Makebcd(sht71_auswertung)
   Locate 2 , 6 : Lcd Bcd(temp_byte_1) ; " sec"

Return
'-- End Menue_men8611 ------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men86111 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_men86111:

   ' Trace Output schreiben
   #if Menue_testmodus                                  ' Trace
      Print #2 , "Enter Menue_men86111"
   #endif                                               ' Trace

   ' Einstellung Messintervall für Temperaturmessungen verändern und ausgeben.
   ' Wertebereich 1 bis 30 Sekunden.

   Menue_data_changed_flag = True

   Select Case Key_taste_buffer

      Case 12:
         ' hoch = erhöhen
         If Sht71_auswertung < Sht71_auswertung_max Then
            Incr Sht71_auswertung
         End If                                         ' Auswerteintervall

      Case 42:
         ' runter = erniedrigen
         If Sht71_auswertung > Sht71_auswertung_min Then
            Decr Sht71_auswertung
         End If                                         ' Auswerteintervall

      Case 44:
         'rechts = erhöhen und bei null anfangen
         Incr Sht71_auswertung
         If Sht71_auswertung > Sht71_auswertung_max Then
            Sht71_auswertung = Sht71_auswertung_min
         End If                                         ' Auswerteintervall

   End Select                                           ' Tastenauswertung

   ' Auswerteintervall für Temperatur und Luftfeuchtigkeit ausgeben
   Temp_byte_1 = Makebcd(sht71_auswertung)
   Locate 2 , 6 : Lcd Bcd(temp_byte_1) ; " sec"

Return
'-- End Menue_men86111 ------------------------------------------------------


'-----------------------------------------------------------------------------
```

```
' MENUE - Gosub-Routine: Menue_men87 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men87:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_men87"
   #endif                                                   ' Trace

   ' Ggf. Signaltoneinstellungen in EEPROM zurückspeichern

   If Menue_data_changed_flag = True Then

      ' Änderungsflag zurücksetzen
      Menue_data_changed_flag = False

      ' Änderungen aus Signaltonmenü in EEPROM schreiben
      Ee_data_sound = Sound_flag

      ' Speicherung durch EE auf Display anzeigen
      Locate 3 , 8 : Lcd "EE"

   End If                                                   ' Daten geändert?

Return
'-- End Menue_men87 ------------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8711 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men8711:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_men8711"
   #endif                                                   ' Trace

   ' Einstellung Signaltöne ausgeben.
   ' Wertebereich AN/AUS.

   Locate 2 , 8

   Select Case Sound_flag
      Case True:
         Lcd "AN"
      Case False:
         Lcd "AUS"
   End Select                                               ' Soundausgabe?

Return
'-- End Menue_men8711 ----------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men87111 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men87111:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_men87111"
   #endif                                                   ' Trace

   ' Einstellung Signaltöne verändern und ausgeben.
   ' Wertebereich AN/AUS.

   Menue_data_changed_flag = True

   Locate 2 , 8
   Select Case Sound_flag
```

```
      Case True:
         Sound_flag = False
         Lcd "AUS"
      Case False:
         Sound_flag = True
         Lcd "An"
   End Select                                              ' Soundausgabe

Return
'-- End Menue_men87111 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men88 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men88:

   ' Trace Output schreiben
   #if Menue_testmodus                                     ' Trace
      Print #2 , "Enter Menue_men88"
   #endif                                                  ' Trace

   ' Ggf. zeitliche Überwachung von Menue MEN881 abschalten

   ' Flag für zeitlich upzudatendes Menue setzen
   Menue_insensmenue_flag = 0

Return
'-- End Menue_men88 ---------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men881 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men881:

   ' Trace Output schreiben
   #if Menue_testmodus                                     ' Trace
      Print #2 , "Enter Menue_men881"
   #endif                                                  ' Trace

   ' Ggf. zeitliche Überwachung von Menue MEN881 abschalten

   ' ggf. Zeitwerte aus Unterfunktion übernehmen
   If Menue_datenuebernahme_flag = True Then

      ' Flag zurücksetzen
      Menue_datenuebernahme_flag = False

      ' Daten üebernehmen
      _sec = Menue_time_sekunde                            ' Interne Sekundenvariable der SoftClock
      _min = Menue_time_minute                             ' Interne Minutenvariable der SoftClock
      _hour = Menue_time_stunde                            ' Interne Stundenvariable der Softclock

   End If                                                  ' Datenübernahme

Return
'-- End Menue_men881 ---------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8811 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men8811:

   ' Trace Output schreiben
   #if Menue_testmodus                                     ' Trace
      Print #2 , "Enter Menue_men8811"
   #endif                                                  ' Trace

   ' Einstellen der Uhrzeit vorbereiten und relevante Variablen laden
```

```
   ' Uhrzeit laden wurde von Menü MEN881 aufgerufen. Es geht hier um Systemzeit

   ' Hilfvariablen für Time-Menü laden

   Menue_time_sekunde = _sec                                ' Interne Sekundenvariable der SoftClock
   Menue_time_minute = _min                                 ' Interne Minutenvariable der SoftClock
   Menue_time_stunde = _hour                                ' Interne Stundenvariable der Softclock

   ' Rücksprung-Ziel festlegen
   Menue_state_buffer = Loadlabel(men881)                   ' Uhrzeit einstellen Menü 8-8-1

   ' Menü neu laden und einspringen
   Menue_state = Loadlabel(time1)                           ' Einsprung in erstes Einstellfenster
   Gosub Menue_change_state_by_sub
   Gosub Lcd_print
   Menue_state_renew = 0
   Gosub Menue_time1                                        ' zugehörige Unterfunktiona anspringen

Return
'-- End Menue_men8811 ------------------------------------------------------


'------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men882 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'------------------------------------------------------------------------------
Menue_men882:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_men882"
   #endif                                                   ' Trace

   ' Einstellen des Datum vorbereiten und relevante Variablen laden

   ' ggf. Datumswerte aus Unterfunktion übernehmen
   If Menue_datenuebernahme_flag = True Then

      ' Flag zurücksetzen
      Menue_datenuebernahme_flag = False

      ' Daten üebernehmen
      _day = Menue_date_tag                                 ' Interne Tagvariable der SoftClock
      _month = Menue_date_monat                             ' Interne Monatsvariable der SoftClock
      _year = Menue_date_jahr                               ' Interne Jahresvariable der Softclock

   End If                                                   ' Datenübernahme

Return
'-- End Menue_men882 ------------------------------------------------------


'------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8821 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'------------------------------------------------------------------------------
Menue_men8821:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_men8821"
   #endif                                                   ' Trace

   ' Einstellen des Datums vorbereiten und relevante Variablen laden

   ' Datum laden wurde von Menü MEN882 aufgerufen. Es geht hier um Systemzeit

   ' Hilfvariablen für Time-Menü laden

   Menue_date_tag = _day                                    ' Interne Tagvariable der SoftClock
   Menue_date_monat = _month                                ' Interne Monatsvariable der SoftClock
   Menue_date_jahr = _year                                  ' Interne Jahresvariable der Softclock

   ' Rücksprung-Ziel festlegen
   Menue_state_buffer = Loadlabel(men882)                   ' Datum einstellen Menü 8-8-2
```

```
   ' Menü neu laden und einspringen
   Menue_state = Loadlabel(date1)                          ' Einsprung in erstes Einstellfenster
   Gosub Menue_change_state_by_sub
   Gosub Lcd_print
   Menue_state_renew = 0
   Gosub Menue_date1                                       ' zugehörige Unterfunktiona anspringen

Return
'-- End Menue_men8821 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men883 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men883:

   ' Trace Output schreiben
   #if Menue_testmodus                                     ' Trace
      Print #2 , "Enter Menue_men883"
   #endif                                                  ' Trace

   ' ggf. zyklisches Updta der DCF77 Statusausgabe deaktivieren

   ' Flag für zeitlich upzudatendes Menue setzen
   Menue_insensmenue_flag = 0

Return
'-- End Menue_men883 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8831 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men8831:

   ' Trace Output schreiben
   #if Menue_testmodus                                     ' Trace
      Print #2 , "Enter Menue_men8831"
   #endif                                                  ' Trace

   ' DCF77 Statusbits ausgeben

   ' Flag für zeitlich upzudatendes Menue setzen
   Menue_insensmenue_flag = 7

   Temp_byte_1 = Makebcd(bdcf_impuls)
   Temp_byte_2 = Makebcd(bdcf_pause)

   ' Impuls und Pause ausgeben
   Locate 1 , 1
   Lcd "Imp: " ; Bcd(temp_byte_1) ; " " ; "Paus: " ; Bcd(temp_byte_2)

   Locate 2 , 1 : Lcd "Status:"
   Locate 2 , 9 : Lcd "Bits:"

   ' Statusbits und DCF-Bits auf LCD ausgeben
   Locate 3 , 1
   Lcd Bin(dcf_status)                                     ' Position 1 - 8
   Lcd Bin(dcf_bits)                                       ' Position 9 - 16

Return
'-- End Menue_men8831 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men89 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men89:

   ' Trace Output schreiben
```

```
   #if Menue_testmodus                                          ' Trace
      Print #2 , "Enter Menue_men89"
   #endif                                                       ' Trace

   ' Ggf. Loggingeinstellungen in EEPROM zurückspeichern

   If Menue_data_changed_flag = True Then

      ' Änderungsflag zurücksetzen
      Menue_data_changed_flag = False

      ' Einstellungen aus Logging-Menü in EEPROM zurückschreiben
      Ee_data_logging_flag = Ee_datalogging_flag
      Ee_data_logging_intervall = Ee_auswertung

      ' Speicherung durch EE auf Display anzeigen
      Locate 3 , 8 : Lcd "EE"

   End If                                                       ' Datenübernahme

Return
'-- End Menue_men89 ------------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8911 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men8911:

   ' Trace Output schreiben
   #if Menue_testmodus                                          ' Trace
      Print #2 , "Enter Menue_men8911"
   #endif                                                       ' Trace

   ' Betriebsmodus für Datenlogging ausgeben
   ' Wertebereich AN/AUS

   Locate 2 , 8

   Select Case Ee_datalogging_flag
      Case True:
         Lcd "AN"
      Case False:
         Lcd "AUS"
   End Select                                                   ' Datenlogging

Return
'-- End Menue_men8911 ----------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men89111 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men89111:

   ' Trace Output schreiben
   #if Menue_testmodus                                          ' Trace
      Print #2 , "Enter Menue_men89111"
   #endif                                                       ' Trace

   ' Betriebsmodus für Datenlogging umschalten und ausgeben
   ' Wertebereich AN/AUS

   Menue_data_changed_flag = True

   Locate 2 , 8
   Select Case Ee_datalogging_flag
      Case True:
         Ee_datalogging_flag = False
         Lcd "AUS"
      Case False:
         Lcd "AN"
         Ee_datalogging_flag = True
         Ee_erfassungs_counter = Sectic_count + Ee_auswertung
```

```
   End Select                                                ' Datenlogging

Return
'-- End Menue_men89111 ------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8921 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men8921:

   ' Trace Output schreiben
   #if Menue_testmodus                                       ' Trace
      Print #2 , "Enter Menue_men8921"
   #endif                                                    ' Trace

   ' Zeitintervall für Daten-Logging ausgeben
   ' Wertebereich 1 - 30 minuten

   Temp_word_1 = Ee_auswertung / 60                          ' Wert auf Minuten hochrechnen
   Temp_byte_1 = Temp_word_1

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Menue_testmodus                                       ' Trace
      Print #2 , "Ee_auswertung : " ; Ee_auswertung
      Print #2 , "Temp_word_1   : " ; Temp_word_1
      Print #2 , "Temp_byte_1   : " ; Temp_byte_1
   #endif                                                    ' Trace

   Temp_byte_1 = Makebcd(temp_byte_1)                        ' In BCD umformatieren
   Locate 2 , 6 : Lcd Bcd(temp_byte_1) ; " min"              ' Auf LCD ausgeben

Return
'-- End Menue_men8921 -------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men89211 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men89211:

   ' Trace Output schreiben
   #if Menue_testmodus                                       ' Trace
      Print #2 , "Enter Menue_men89211"
   #endif                                                    ' Trace

   ' Zeitintervall für Daten-Logging verändern und ausgeben
   ' Wertebereich 1 - 30 minuten

   Menue_data_changed_flag = True

   Select Case Key_taste_buffer

      Case 12:
         ' hoch = erhöhen
         If Ee_auswertung < Ee_auswertung_max Then
            Ee_auswertung = Ee_auswertung + 60
         End If                                              ' Auswerteintervall

      Case 42:
         ' runter = erniedrigen
         If Ee_auswertung > Ee_auswertung_min Then
            Ee_auswertung = Ee_auswertung - 60
         End If                                              ' Auswerteintervall

      Case 44:
         'rechts = erhöhen und bei null anfangen
         Ee_auswertung = Ee_auswertung + 60
         If Ee_auswertung > Ee_auswertung_max Then
            Ee_auswertung = Ee_auswertung_min
         End If                                              ' Auswerteintervall

   End Select                                                ' Tastenauswertung
```

```
   Temp_word_1 = Ee_auswertung / 60                          ' Wert auf Minuten hochrechnen
   Temp_byte_1 = Temp_word_1

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Menue_testmodus                                       ' Trace
     Print #2 , "Ee_auswertung : " ; Ee_auswertung
     Print #2 , "Temp_word_1   : " ; Temp_word_1
     Print #2 , "Temp_byte_1   : " ; Temp_byte_1
   #endif                                                    ' Trace

   Temp_byte_1 = Makebcd(temp_byte_1)                        ' In BCD umformatieren
   Locate 2 , 6 : Lcd Bcd(temp_byte_1) ; " min"              ' Auf LCD ausgeben

Return
'-- End Menue_men89211 ------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8931 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men8931:

   ' Trace Output schreiben
   #if Menue_testmodus                                       ' Trace
      Print #2 , "Enter Menue_men8931"
   #endif                                                    ' Trace

   ' EEPROM mit Loggingdaten löschen und neu initialisieren

   ' Datenstruktur externes EEPROM entspricht nicht dem aktuellen Stand
   ' => Datenstruktur neu anlegen bzw. neue Defaultwerte setzen.

   Ee_data_count = Ee_data_count_default
   Ee_data_overrun = Ee_data_overrun_default

   Call Ee_write_eeprom_word(ee_data_count_adress , Ee_data_count_default)
   Waitms 10                                                 'Wartezeit für sicheres Schreiben
   Call Ee_write_eeprom_byte(ee_data_overrun_adress , Ee_data_overrun_default)

   ' Datenlöschen durch 2xBeep signalisieren wenn Soundflag gesetzt
   If Sound_flag = True Then
      Call Sound_beep(sound_beep_time)                       ' Soubroutine Beep aufrufen
      Waitms 100                                             ' 100 ms warten
      Call Sound_beep(sound_beep_time)                       ' Soubroutine Beep aufrufen
   End If                                                    ' Soundflag

   ' Menü neu laden und einspringen
   Menue_state = Loadlabel(men893)                           ' Einsprung in erstes Einstellfenster
   Gosub Menue_change_state_by_sub

Return
'-- End Menue_men8931 ----------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8a (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men8a:

   ' Trace Output schreiben
   #if Menue_testmodus                                       ' Trace
      Print #2 , "Enter Menue_men8a"
   #endif                                                    ' Trace

   ' Ggf. Beschattungseinstellungen in EEPROM zurückspeichern

   If Menue_data_changed_flag = True Then

      ' Änderungsflag zurücksetzen
      Menue_data_changed_flag = False

      ' Einstellungen aus Beschattungsmenü in EEPROM schreiben
      Ee_data_conv_timeout = Rel_beschattung_delay
      Ee_data_conv_temp_flag = Rel_zu_temp_flag
```

```
      Ee_data_conv_temperatur = Rel_zu_temperatur
      Ee_data_conv_light_flag = Rel_auf_dunkel_flag
      Ee_data_conv_lichtwert = Rel_auf_dunkelheit

      ' Speicherung durch EE auf Display anzeigen
      Locate 3 , 8 : Lcd "EE"

   End If                                                ' Daten geändert?

Return
'-- End Menue_men8a ------------------------------------------------------


'---------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8a11 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'---------------------------------------------------------------------------
Menue_men8a11:

   ' Trace Output schreiben
   #if Menue_testmodus                                   ' Trace
      Print #2 , "Enter Menue_men8a11"
   #endif                                                ' Trace

   ' Beschattugstimeout ausgeben
   ' Wertebereich 1 bis 200 Sekunden

   ' String ausgeben
   Temp_string_1 = Byte2string(rel_beschattung_delay , &H30)       ' Mit 0 füllen
   Locate 2 , 5 : Lcd Temp_string_1 ; " sec"

Return
'-- End Menue_men8a11 ----------------------------------------------------


'---------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8a111 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'---------------------------------------------------------------------------
Menue_men8a111:

   ' Trace Output schreiben
   #if Menue_testmodus                                   ' Trace
      Print #2 , "Enter Menue_men8a111"
   #endif                                                ' Trace

   ' Beschattugstimeout verändern und ausgeben
   ' Wertebereich 1 bis 200 Sekunden

   Menue_data_changed_flag = True

   Select Case Key_taste_buffer

      Case 12:
         ' hoch = erhöhen
         If Rel_beschattung_delay < Rel_beschattung_max Then
            Incr Rel_beschattung_delay
         End If                                          ' Delayzeit

      Case 42:
         ' runter = erniedrigen
         If Rel_beschattung_delay > Rel_beschattung_min Then
            Decr Rel_beschattung_delay
         End If                                          ' Delayzeit

      Case 44:
         'rechts = erhöhen und bei null anfangen
         Incr Rel_beschattung_delay
         If Rel_beschattung_delay > Rel_beschattung_max Then
            Rel_beschattung_delay = Rel_beschattung_min
         End If                                          ' Delayzeit

   End Select                                            ' Tastenauswertung

   ' String ausgeben
   Temp_string_1 = Byte2string(rel_beschattung_delay , &H30)       ' Mit 0 füllen
```

```
   Locate 2 , 5 : Lcd Temp_string_1 ; " sec"

Return
'-- End Menue_men8a111 ------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8a211 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men8a211:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men8a211"
   #endif                                                 ' Trace

   ' Status "Auffahren bei Dunkelheit" ausgeben
   ' Wertebereich JA/NEIN

   Locate 2 , 8

   Select Case Rel_auf_dunkel_flag
      Case True:
         Lcd "An"
      Case False:
         Lcd "Aus"
   End Select                                             ' Auffahren bei Dunkelheit?

Return
'-- End Menue_men8a211 ------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8a2111 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men8a2111:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men8a2111"
   #endif                                                 ' Trace

   ' Status "Auffahren bei Dunkelheit" umschalten und ausgeben
   ' Wertebereich JA/NEIN

   Menue_data_changed_flag = True

   Locate 2 , 8
   Select Case Rel_auf_dunkel_flag
      Case True:
         Rel_auf_dunkel_flag = False
         Lcd "Aus"

         ' zyklisches Update deaktivieren
         Light_erfassung = False

      Case False:
         Rel_auf_dunkel_flag = True
         Lcd "An"

         ' Licht einmalig erfassen
         Light_value = Light_measurement(light_messwert_zahl )
         ' zyklisches Update aktivieren
         Light_erfassung = True
         ' Auswerteintervall setzen
         Light_erfassungs_counter = Sectic_count + Light_auswertung_default

   End Select                                             ' Auffahren bei Dunkelheit

Return
'-- End Menue_men8a2111 ------------------------------------------------------


'-------------------------------------------------------------------------------
```

```
' MENUE - Gosub-Routine: Menue_men8a221 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men8a221:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men8a221"
   #endif                                                 ' Trace

   ' Grenzdunkelheit ausgeben
   ' Wertebereich 1 bis 15

   Temp_byte_1 = Makebcd(rel_auf_dunkelheit)
   Locate 2 , 8 : Lcd Bcd(temp_byte_1)

Return
'-- End Menue_men8a221 -------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8a2111 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men8a2211:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men8a2211"
   #endif                                                 ' Trace

   ' Grenzdunkelheit ändern und ausgeben
   ' Wertebereich 1 bis 15

   Menue_data_changed_flag = True

   Select Case Key_taste_buffer

      Case 12:
         ' hoch = erhöhen
         If Rel_auf_dunkelheit < Rel_auf_dunkelheit_max Then
            Incr Rel_auf_dunkelheit
         End If                                           ' Dunkelheit

      Case 42:
         ' runter = erniedrigen
         If Rel_auf_dunkelheit > Rel_auf_dunkelheit_min Then
            Decr Rel_auf_dunkelheit
         End If                                           ' Dunkelheit

      Case 44:
         'rechts = erhöhen und bei null anfangen
         Incr Rel_auf_dunkelheit
         If Rel_auf_dunkelheit > Rel_auf_dunkelheit_max Then
            Rel_auf_dunkelheit = Rel_auf_dunkelheit_min
         End If                                           ' Dunkelheit

   End Select                                             ' Tastenauswertung

   Temp_byte_1 = Makebcd(rel_auf_dunkelheit)
   Locate 2 , 8 : Lcd Bcd(temp_byte_1)

Return
'-- End Menue_men8a2211 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8a311 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_men8a311:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
```

```
      Print #2 , "Enter Menue_men8a311"
   #endif                                                ' Trace

   ' Status "Zufahren bei Temperatur" ausgeben
   ' Wertebereich JA/NEIN

   Locate 2 , 8

   Select Case Rel_zu_temp_flag
      Case True:
         Lcd "An"
      Case False:
         Lcd "Aus"
   End Select                                            ' Zufahren bei Temperatur

Return
'-- End Menue_men8a311 ------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8a3111 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men8a3111:

   ' Trace Output schreiben
   #if Menue_testmodus                                   ' Trace
      Print #2 , "Enter Menue_men8a3111"
   #endif                                                ' Trace

   ' Status "Zufahren bei Temperatur" ändern und ausgeben
   ' Wertebereich JA/NEIN

   Menue_data_changed_flag = True

   Locate 2 , 8
   Select Case Rel_zu_temp_flag
      Case True:
         Rel_zu_temp_flag = False
         Lcd "Aus"
      Case False:
         Rel_zu_temp_flag = True
         Lcd "An"
   End Select                                            ' Zufahren bei Temperatur

Return
'-- End Menue_men8a3111 ------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8a321 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_men8a321:

   ' Trace Output schreiben
   #if Menue_testmodus                                   ' Trace
      Print #2 , "Enter Menue_men8a321"
   #endif                                                ' Trace

   ' Grenztemperatur ausgeben
   ' Wertebereich 1 bis 99°C

   Temp_byte_1 = Makebcd(rel_zu_temperatur)
   Locate 2 , 8 : Lcd Bcd(temp_byte_1)

Return
'-- End Menue_men8a321 ------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8a3211 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
```

```
Menue_men8a3211:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men8a3211"
   #endif                                                 ' Trace

   ' Grenztemperatur ändern und ausgeben
   ' Wertebereich 1 bis 99°C

   Menue_data_changed_flag = True

   Select Case Key_taste_buffer

      Case 12:
         ' hoch = erhöhen
         If Rel_zu_temperatur < Rel_zu_temperatur_max Then
            Incr Rel_zu_temperatur
         End If                                           ' Temperatur

      Case 42:
         ' runter = erniedrigen
         If Rel_zu_temperatur > Rel_zu_temperatur_min Then
            Decr Rel_zu_temperatur
         End If                                           ' Temperatur

      Case 44:
         'rechts = erhöhen und bei null anfangen
         Incr Rel_zu_temperatur
         If Rel_zu_temperatur > Rel_zu_temperatur_max Then
            Rel_zu_temperatur = Rel_zu_temperatur_min
         End If                                           ' Temperatur

   End Select

   Temp_byte_1 = Makebcd(rel_zu_temperatur)
   Locate 2 , 8 : Lcd Bcd(temp_byte_1)

Return
'-- End Menue_men8a3211 -------------------------------------------------------



'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_men8b11 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_men8b11:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_men8b11"
   #endif                                                 ' Trace

   ' Min- und Maxwerte in globalen Variablen löschen
   Sht71_t_min = 0.0
   Sht71_t_max = 0.0

   Sht71_rhtrue_min = 0.0
   Sht71_rhtrue_max = 0.0

   Wind_kmh_max = 0.0


   ' Min- und Maxwerte in internem EEPROM löschen
   Ee_data_sht71_t_min = 0.0
   Ee_data_sht71_t_max = 0.0

   Ee_data_sht71_rhtrue_min = 0.0
   Ee_data_sht71_rhtrue_max = 0.0

   Ee_data_wind_kmh_max = 0.0


   ' Datenlöschen durch 2xBeep signalisieren wenn Soundflag gesetzt
   If Sound_flag = True Then
      Call Sound_beep(sound_beep_time)                    ' Soubroutine Beep aufrufen
      Waitms 100                                          ' 100 ms warten
```

```
      Call Sound_beep(sound_beep_time)                    ' Soubroutine Beep aufrufen
   End If                                                 ' Soundausgabe

   ' Menü neu laden und einspringen
   Menue_state = Loadlabel(men8b1)                        ' Einsprung in erstes Einstellfenster
   Gosub Menue_change_state_by_sub

Return
'-- End Menue_men8b11 ------------------------------------------------------



'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_time1 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_time1:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_time1"
   #endif                                                 ' Trace

   ' Stunden ausgeben

   ' Zeitstring mit ":" aufbauen
   Temp_string_1 = Time2string(menue_time_stunde , Menue_time_minute , Menue_time_sekunde , &H3A)
   ' Zeitstring ausgeben
   Locate 3 , 9 : Lcd Temp_string_1

Return
'-- End Menue_time1 -------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_time11 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_time11:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_time11"
   #endif                                                 ' Trace

   ' Stunden erhöhen und ausgeben

   Incr Menue_time_stunde
   If Menue_time_stunde > 23 Then Menue_time_stunde = 0

   ' Zeitstring mit ":" aufbauen
   Temp_string_1 = Time2string(menue_time_stunde , Menue_time_minute , Menue_time_sekunde , &H3A)
   ' Zeitstring ausgeben
   Locate 3 , 9 : Lcd Temp_string_1

Return
'-- End Menue_time11 ------------------------------------------------------


'-------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_time12 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-------------------------------------------------------------------------------
Menue_time12:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_time12"
   #endif                                                 ' Trace

   ' Stunden verringern und ausgeben

   Decr Menue_time_stunde
   If Menue_time_stunde = 255 Then Menue_time_stunde = 23
```

```
   ' Zeitstring mit ":" aufbauen
   Temp_string_1 = Time2string(menue_time_stunde , Menue_time_minute , Menue_time_sekunde , &H3A)
   ' Zeitstring ausgeben
   Locate 3 , 9 : Lcd Temp_string_1

Return
'-- End Menue_time12 ------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_time2 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_time2:

   ' Trace Output schreiben
   #if Menue_testmodus                                  ' Trace
      Print #2 , "Enter Menue_time2"
   #endif                                               ' Trace

   ' Minuten ausgeben

   ' Zeitstring mit ":" aufbauen
   Temp_string_1 = Time2string(menue_time_stunde , Menue_time_minute , Menue_time_sekunde , &H3A)
   ' Zeitstring ausgeben
   Locate 3 , 9 : Lcd Temp_string_1

Return
'-- End Menue_time2 -------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_time21 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_time21:

   ' Trace Output schreiben
   #if Menue_testmodus                                  ' Trace
      Print #2 , "Enter Menue_time21"
   #endif                                               ' Trace

   ' Minuten erhöhen und ausgeben

   Incr Menue_time_minute
   If Menue_time_minute > 59 Then Menue_time_minute = 0

   ' Zeitstring mit ":" aufbauen
   Temp_string_1 = Time2string(menue_time_stunde , Menue_time_minute , Menue_time_sekunde , &H3A)
   ' Zeitstring ausgeben
   Locate 3 , 9 : Lcd Temp_string_1

Return
'-- End Menue_time21 ------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_time22 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_time22:

   ' Trace Output schreiben
   #if Menue_testmodus                                  ' Trace
      Print #2 , "Enter Menue_time22"
   #endif                                               ' Trace

   ' Minuten verringern und ausgeben

   Decr Menue_time_minute
   If Menue_time_minute = 255 Then Menue_time_minute = 59

   ' Zeitstring mit ":" aufbauen
```

```
   Temp_string_1 = Time2string(menue_time_stunde , Menue_time_minute , Menue_time_sekunde , &H3A)
   ' Zeitstring ausgeben
   Locate 3 , 9 : Lcd Temp_string_1

Return
'-- End Menue_time22 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_time3 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_time3:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_time3"
   #endif                                                   ' Trace

   ' Sekunden ausgeben

   ' Zeitstring mit ":" aufbauen
   Temp_string_1 = Time2string(menue_time_stunde , Menue_time_minute , Menue_time_sekunde , &H3A)
   ' Zeitstring ausgeben
   Locate 3 , 9 : Lcd Temp_string_1

Return
'-- End Menue_time3 -------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_time31 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_time31:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_time31"
   #endif                                                   ' Trace

   ' Sekunden erhöhen und ausgeben

   Incr Menue_time_sekunde
   If Menue_time_sekunde > 59 Then Menue_time_sekunde = 0

   ' Zeitstring mit ":" aufbauen
   Temp_string_1 = Time2string(menue_time_stunde , Menue_time_minute , Menue_time_sekunde , &H3A)
   ' Zeitstring ausgeben
   Locate 3 , 9 : Lcd Temp_string_1

Return
'-- End Menue_time31 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_time32 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_time32:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_time32"
   #endif                                                   ' Trace

   ' Sekunden verringern und ausgeben

   Decr Menue_time_sekunde
   If Menue_time_sekunde = 255 Then Menue_time_sekunde = 59

   ' Zeitstring mit ":" aufbauen
   Temp_string_1 = Time2string(menue_time_stunde , Menue_time_minute , Menue_time_sekunde , &H3A)
   ' Zeitstring ausgeben
```

```
   Locate 3 , 9 : Lcd Temp_string_1

Return
'-- End Menue_time32 ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_time_ende (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_time_ende:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_time_ende"
   #endif                                                   ' Trace

   ' Rücksprung in relevantes Menü
   Menue_datenuebernahme_flag = True

   ' Menüstatus zurückschalten
   Menue_state = Menue_state_buffer

   Gosub Menue_change_state_by_sub
   Gosub Lcd_print
   Menue_state_renew = 0
   Gosub Menue_call_subroutine

Return
'-- End Menue_time_ende ------------------------------------------------------


'--------------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_date1 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'--------------------------------------------------------------------------------
Menue_date1:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_date1"
   #endif                                                   ' Trace

   ' Für Date11 und Date12 bereits schon die Anzahl der Tage im Monat
   ' berechnen und den aktuellen Tag ausgeben

   ' Anzahl der Tages des aktiven Monats aus Tabelle auslesen
   Menue_tagesanzahl = Lookup(menue_date_monat , Monate)

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Aktiver Monat : " ; Menue_date_monat
      Print #2 , "Tageszahl     : " ; Menue_tagesanzahl
   #endif                                                   ' Trace

   ' Ist der aktuelle Monat Februar
   If Menue_date_monat = 2 Then
      ' JA -> es könnte ein Schaltjahr sein!

      ' Jahr auf volles Jahr z.B. 2008 ergänzen
      Temp_word_1 = 2000 + Menue_date_jahr
      ' Schaltjahr prüfen
      Temp_byte_2 = Schaltjahr(temp_word_1)

      ' Trace Output schreiben
      #if Menue_testmodus                                   ' Trace
         Select Case Temp_byte_2
            Case True:
               Print #2 , "Jahr ist Schaltjahr"
            Case False:
               Print #2 , "Jahr ist kein Schaltjahr"
         End Seclect
      #endif                                                ' Trace

      ' Ist aktuelles Jahr ein Schaltjahr
      If Temp_byte_2 = True Then
```

```
         ' JA, dann Tagesanzahl auf 29 setzen
         Menue_tagesanzahl = 29
      End If                                                ' Schaltjahr

   End If                                                   ' Februar?

   ' Aktuelle Tagesanzahl steht in Variable Temp_byte_1!

   ' Datumstring mit "." aufbauen
   Temp_string_1 = Time2string(menue_date_tag , Menue_date_monat , Menue_date_jahr , &H2E)
   ' Datumstring ausgeben
   Locate 3 , 9 : Lcd Temp_string_1

Return
'-- End Menue_date1 ------------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_date11 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_date11:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_date11"
   #endif                                                   ' Trace

   ' Tag erhöhen und ausgeben

   Incr Menue_date_tag
   If Menue_date_tag > Menue_tagesanzahl Then Menue_date_tag = 1

   ' Datumstring mit "." aufbauen
   Temp_string_1 = Time2string(menue_date_tag , Menue_date_monat , Menue_date_jahr , &H2E)
   ' Datumstring ausgeben
   Locate 3 , 9 : Lcd Temp_string_1

Return
'-- End Menue_date11 ------------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_date12 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_date12:

   ' Trace Output schreiben
   #if Menue_testmodus                                      ' Trace
      Print #2 , "Enter Menue_date12"
   #endif                                                   ' Trace

   ' Tag verringern und ausgeben

   Decr Menue_date_tag
   If Menue_date_tag = 0 Then Menue_date_tag = Menue_tagesanzahl

   ' Datumstring mit "." aufbauen
   Temp_string_1 = Time2string(menue_date_tag , Menue_date_monat , Menue_date_jahr , &H2E)
   ' Datumstring ausgeben
   Locate 3 , 9 : Lcd Temp_string_1

Return
'-- End Menue_date12 ------------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_date2 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_date2:

   ' Trace Output schreiben
```

```
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_date2"
   #endif                                                 ' Trace

   ' Monat ausgeben

   ' Datumstring mit "." aufbauen
   Temp_string_1 = Time2string(menue_date_tag , Menue_date_monat , Menue_date_jahr , &H2E)
   ' Datumstring ausgeben
   Locate 3 , 9 : Lcd Temp_string_1

Return
'-- End Menue_date2 ------------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_date21 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_date21:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_date21"
   #endif                                                 ' Trace

   ' Monat erhöhen und ausgeben

   Incr Menue_date_monat
   If Menue_date_monat > 12 Then Menue_date_monat = 1

   ' Datumstring mit "." aufbauen
   Temp_string_1 = Time2string(menue_date_tag , Menue_date_monat , Menue_date_jahr , &H2E)
   ' Datumstring ausgeben
   Locate 3 , 9 : Lcd Temp_string_1

Return
'-- End Menue_date21 ------------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_date22 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_date22:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_date22"
   #endif                                                 ' Trace

   ' Monat verringern und ausgeben

   Decr Menue_date_monat
   If Menue_date_monat = 0 Then Menue_date_monat = 12

   ' Datumstring mit "." aufbauen
   Temp_string_1 = Time2string(menue_date_tag , Menue_date_monat , Menue_date_jahr , &H2E)
   ' Datumstring ausgeben
   Locate 3 , 9 : Lcd Temp_string_1

Return
'-- End Menue_date22 ------------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_date3 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_date3:

   ' Trace Output schreiben
   #if Menue_testmodus                                    ' Trace
      Print #2 , "Enter Menue_date3"
```

```
   #endif                                                ' Trace

   ' Jahr ausgeben

   ' Datumstring mit "." aufbauen
   Temp_string_1 = Time2string(menue_date_tag , Menue_date_monat , Menue_date_jahr , &H2E)
   ' Datumstring ausgeben
   Locate 3 , 9 : Lcd Temp_string_1

Return
'-- End Menue_date3 ------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_date31 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_date31:

   ' Trace Output schreiben
   #if Menue_testmodus                                   ' Trace
      Print #2 , "Enter Menue_date31"
   #endif                                                ' Trace

   ' Jahr erhöhen und ausgeben

   Incr Menue_date_jahr
   If Menue_date_jahr > 99 Then Menue_date_jahr = 1

   ' Datumstring mit "." aufbauen
   Temp_string_1 = Time2string(menue_date_tag , Menue_date_monat , Menue_date_jahr , &H2E)
   ' Datumstring ausgeben
   Locate 3 , 9 : Lcd Temp_string_1

Return
'-- End Menue_date31 ------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_date32 (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_date32:

   ' Trace Output schreiben
   #if Menue_testmodus                                   ' Trace
      Print #2 , "Enter Menue_date32"
   #endif                                                ' Trace

   ' Jahr verringern und ausgeben

   Decr Menue_date_jahr
   If Menue_date_jahr = 0 Then Menue_date_jahr = 99

   ' Datumstring mit "." aufbauen
   Temp_string_1 = Time2string(menue_date_tag , Menue_date_monat , Menue_date_jahr , &H2E)
   ' Datumstring ausgeben
   Locate 3 , 9 : Lcd Temp_string_1

Return
'-- End Menue_date32 ------------------------------------------------------


'-----------------------------------------------------------------------------
' MENUE - Gosub-Routine: Menue_date_ende (Menue-Unterstützung-Routine)
' Routine wird von Menue-State-Machine bei eintritt in Menue aufgerufen.
' Funktion dient zum Errechnen von Werten, Ausgeben von Informationsn usw. die
' ergänzend zum jeweiligen Menueschritt benötigt werden.
'-----------------------------------------------------------------------------
Menue_date_ende:

   ' Trace Output schreiben
   #if Menue_testmodus                                   ' Trace
      Print #2 , "Enter Menue_date_ende"
   #endif                                                ' Trace
```

```
   ' Alle Arbeitsvariablen zurückschreiben und Date neu setzen

   ' Rücksprung in relevantes Menü
   Menue_datenuebernahme_flag = True

   ' Menüstatus zurückschalten
   Menue_state = Menue_state_buffer

   Gosub Menue_change_state_by_sub
   Gosub Lcd_print
   Menue_state_renew = 0
   Gosub Menue_call_subroutine

Return
'-- End Menue_date_ende ------------------------------------------------------


' ********************
' * RTC:  Echtzeit   *
' ********************

'-------------------------------------------------------------------------------
' RTC - Gosub-Routine: Rtc_set_rtc_clock
' Subroutine übernimmt bei erfolgreicher Synchronisation durch DCF77 die
' aktuellen Zeit und Datumsinformationen in die RTC DS1307
'-------------------------------------------------------------------------------
Rtc_set_rtc_clock:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Rtc_testmodus                                    ' Trace
      Print #2 , "Enter Rcf_set_rtc_clock"
   #endif                                               ' Trace

   ' Daten zum Speichern in BCD Format aufbereiten
   Dcf77_hour = Makebcd(_hour)
   Dcf77_min = Makebcd(_min)
   Dcf77_sec = Makebcd(_sec)
   Dcf77_weekday = Makebcd(_weekday)
   Dcf77_day = Makebcd(_day)
   Dcf77_month = Makebcd(_month)
   Dcf77_year = Makebcd(_year)

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Rtc_testmodus                                    ' Trace
      Print #2 , "Schreibe Zeitinformationen in DS1307:"
      Print #2 , "Stunde    : " ; Bcd(dcf77_hour)
      Print #2 , "Minute    : " ; Bcd(dcf77_min)
      Print #2 , "Sekunde   : " ; Bcd(dcf77_sec)
      Print #2 , "Wochentag : " ; Bcd(dcf77_weekday)
      Print #2 , "Tag       : " ; Bcd(dcf77_day)
      Print #2 , "Monat     : " ; Bcd(dcf77_month)
      Print #2 , "Jahr      : " ; Bcd(dcf77_year)
   #endif                                               ' Trace

   ' RTC DS1307 synchronisieren
   I2cstart                                             ' START-Sequenz senden
   I2cwbyte Rtc_address_write                           ' Daten schreiben initiieren
   I2cwbyte Rtc_address_seconds                         ' Adresse übertragen = &H00

   I2cwbyte Dcf77_sec                                   ' Byte 1 übertragen
   I2cwbyte Dcf77_min                                   ' Byte 2 übertragen
   I2cwbyte Dcf77_hour                                  ' Byte 3 übertragen
   I2cwbyte Dcf77_weekday                               ' Byte 4 übertragen
   I2cwbyte Dcf77_day                                   ' Byte 5 übertragen
   I2cwbyte Dcf77_month                                 ' Byte 6 übertragen
   I2cwbyte Dcf77_year                                  ' Byte 7 übertragen
   I2cstop                                              ' STOP-Sequenz

Return
'-- End Rtc_set_rtc_clock ---------------------------------------------------


'-------------------------------------------------------------------------------
' RTC - Gosub-Routine: Rtc_set_softclock
' Subroutine übernimmt bei Aufruf der Routine die in der RTC "mitlaufenden"
' Zeitinformationen in die SoftClock.
' Damit kann bereits zu Systemstart ohne DCF77 Empfang mit "realer" Zeit
' gestartet werden.
'-------------------------------------------------------------------------------
Rtc_set_softclock:
```

```
   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Rtc_testmodus                                    ' Trace
      Print #2 , "Enter Rcf_set_softclock"
   #endif                                               ' Trace

   ' RTC DS1307 auslesen
   I2cstart                                             ' START-Sequenz senden
   I2cwbyte Rtc_address_write                           ' Daten schreiben initiieren
   I2cwbyte Rtc_address_seconds                         ' Adresse übertragen = &H00
   ' I2cstop                                            ' STOP-Sequenz wird nicht zwingend benö-
tigt

   I2cstart                                             ' START-Sequenz erneut senden
   I2cwbyte Rtc_address_read                            ' Daten lesen initiieren
   I2crbyte Dcf77_sec , Ack                             ' Byte 1 lesen
   I2crbyte Dcf77_min , Ack                             ' Byte 2 lesen
   I2crbyte Dcf77_hour , Ack                            ' Byte 3 lesen
   I2crbyte Dcf77_weekday , Ack                         ' Byte 4 lesen
   I2crbyte Dcf77_day , Ack                             ' Byte 5 lesen
   I2crbyte Dcf77_month , Ack                           ' Byte 6 lesen
   I2crbyte Dcf77_year , Nack                           ' Byte 7 lesen und fertig (NACK)
   I2cstop                                              ' STOP-Sequenz

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Rtc_testmodus                                    ' Trace
      Print #2 , "Lese Zeitinformationen aus DS1307:"
      Print #2 , "Stunde    : " ; Bcd(dcf77_hour)
      Print #2 , "Minute    : " ; Bcd(dcf77_min)
      Print #2 , "Sekunde   : " ; Bcd(dcf77_sec)
      Print #2 , "Wochentag : " ; Bcd(dcf77_weekday)
      Print #2 , "Tag       : " ; Bcd(dcf77_day)
      Print #2 , "Monat     : " ; Bcd(dcf77_month)
      Print #2 , "Jahr      : " ; Bcd(dcf77_year)
   #endif                                               ' Trace

   ' ausgelesene Daten in SoftClock Variablen übernehmen
   _hour = Makedec(dcf77_hour)
   _min = Makedec(dcf77_min)
   _sec = Makedec(dcf77_sec)
   _weekday = Makedec(dcf77_weekday)
   _day = Makedec(dcf77_day)
   _month = Makedec(dcf77_month)
   _year = Makedec(dcf77_year)

Return
'-- End Rtc_set_softclock -------------------------------------------------


' ********************
' * REL:  Relais     *
' ********************

'-------------------------------------------------------------------------------
' REL - Gosub-Routine: Relais_konfig
' Subroutine zur Konfiguration der Relais
'-------------------------------------------------------------------------------
Relais_konfig:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Relais_testmodus                                 ' Trace
      Print #2 , "Enter Relais_konfig"
   #endif                                               ' Trace

   '----- Konfiguration Relais [AUF/ZU/Licht] (Transistorstufe) -----
   'Port G.0 (Licht) als Ausgang definieren und Relais ausschalten
   Config Out_licht = Output
   Rel_licht_on = 0

   'Port E.2 (Dauer-Licht) als Ausgang definieren und Relais ausschalten
   Config Out_dauerlicht = Output
   Rel_dauerlicht_on = 0

   'Port C.6 (AUF) als Ausgang definieren und Relais ausschalten
   Config Out_auf = Output
   Rel_auf_on = 0

   'Port C.7 (ZU) als Ausgang definieren und Relais ausschalten
   Config Out_zu = Output
   Rel_zu_on = 0
```

```
   ' Variablen initialisieren
   Rel_beschattung_delay = Rel_delay_default                ' Globale Variable mit Default lasen

   Rel_auf_dunkel_flag = False                              ' Flag mit Default-Wert laden
   Rel_zu_temp_flag = False                                 ' Flag mit Default-Wert laden

   Rel_auf_dunkelheit = Rel_auf_dunkelheit_default          ' Globale Vaiable mit Default laden
   Rel_zu_temperatur = Rel_zu_temperatur_default

Return
'-- End Relais_konfig ------------------------------------------------------


'--------------------------------------------------------------------------------
' REL - Gosub-Routine: Relais_auf_an
' Subroutine zum Eimschalten des Relais AUF
'--------------------------------------------------------------------------------
Relais_auf_an:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Relais_testmodus                                     ' Trace
      Print #2 , "Enter Relais_auf_an"
   #endif                                                   ' Trace

   Rel_auf_on = 1                                           ' Relais AUF einschalten

Return
'-- End Relais_auf_an ------------------------------------------------------


'--------------------------------------------------------------------------------
' REL - Gosub-Routine: Relais_auf_aus
' Subroutine zum Ausschalten des Relais AUF
'--------------------------------------------------------------------------------
Relais_auf_aus:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Relais_testmodus                                     ' Trace
      Print #2 , "Enter Relais_auf_aus"
   #endif                                                   ' Trace

   Rel_auf_on = 0                                           ' Relais AUF ausschalten

Return
'-- End Relais_auf_aus ------------------------------------------------------


'--------------------------------------------------------------------------------
' REL - Gosub-Routine: Relais_zu_an
' Subroutine zum Eimschalten des Relais ZU
'--------------------------------------------------------------------------------
Relais_zu_an:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Relais_testmodus                                     ' Trace
      Print #2 , "Enter Relais_zu_an"
   #endif                                                   ' Trace

   Rel_zu_on = 1                                            ' Relais ZU einschalten

Return
'-- End Relais_zu_an ------------------------------------------------------


'--------------------------------------------------------------------------------
' REL - Gosub-Routine: Relais_zu_aus
' Subroutine zum Ausschalten des Relais ZU
'--------------------------------------------------------------------------------
Relais_zu_aus:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Relais_testmodus                                     ' Trace
      Print #2 , "Enter Relais_zu_aus"
   #endif                                                   ' Trace

   Rel_zu_on = 0                                            ' Relais ZU ausschalten

Return
'-- End Relais_zu_aus ------------------------------------------------------
```

```
'-------------------------------------------------------------------------------
' REL - Gosub-Routine: Relais_licht_an
' Subroutine zum Eimschalten des Relais LICHT
'-------------------------------------------------------------------------------
Relais_licht_an:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Relais_testmodus                                    ' Trace
      Print #2 , "Enter Relais_licht_an"
   #endif                                                  ' Trace

   Rel_licht_on = 1                                        ' Relais LICHT einschalten

Return
'-- End Relais_licht_an ------------------------------------------------------


'-------------------------------------------------------------------------------
' REL - Gosub-Routine: Relais_licht_aus
' Subroutine zum Ausschalten des Relais LICHT
'-------------------------------------------------------------------------------
Relais_licht_aus:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Relais_testmodus                                    ' Trace
      Print #2 , "Enter Relais_licht_aus"
   #endif                                                  ' Trace

   Rel_licht_on = 0                                        ' Relais LICHT ausschalten

Return
'-- End Relais_licht_aus -----------------------------------------------------


'-------------------------------------------------------------------------------
' REL - Gosub-Routine: Relais_dauerlicht_an
' Subroutine zum Eimschalten des Relais LICHT
'-------------------------------------------------------------------------------
Relais_dauerlicht_an:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Relais_testmodus                                    ' Trace
      Print #2 , "Enter Relais_dauerlicht_an"
   #endif                                                  ' Trace

   Rel_dauerlicht_on = 1                                   ' Relais LICHT einschalten

Return
'-- End Relais_dauerlicht_an -------------------------------------------------


'-------------------------------------------------------------------------------
' REL - Gosub-Routine: Relais_dauerlicht_aus
' Subroutine zum Ausschalten des Relais LICHT
'-------------------------------------------------------------------------------
Relais_dauerlicht_aus:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Relais_testmodus                                    ' Trace
      Print #2 , "Enter Relais_dauerlicht_aus"
   #endif                                                  ' Trace

   Rel_dauerlicht_on = 0                                   ' Relais LICHT ausschalten

Return
'-- End Relais_dauerlicht_aus ------------------------------------------------

'-------------------------------------------------------------------------------
' REL - Gosub-Routine: Rel_beschattung_auf
' Subroutine zum kompletten Auffahren der Beschattung mit Richtungsumschaltung
'-------------------------------------------------------------------------------
Relais_beschattung_auf:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Relais_testmodus                                    ' Trace
      Print #2 , "Enter Relais_beschattung_auf"
   #endif                                                  ' Trace
```

```
   ' Prüfung ob Antrieb sich aktuell in ZU-Richtung bewegt
   If Beschattung_zu_flag = Aktiv Then
      ' Wenn JA Antrieb abschalten und Flag zurücksetzen
      Beschattung_zu_flag = Passiv
      Gosub Relais_zu_aus
      Waitms 500                                          ' Zum Schutz des Antriebs 500 ms warten
   End If                                                 ' Beschattung zu aktiv

   ' Antriebe AUF einschalten und Timerüberwachung aktivieren
   Gosub Relais_auf_an
   Beschattung_auf_flag = Aktiv
   Beschattung_auf_counter = Sectic_count + Rel_beschattung_delay

   Sht71_autozufahren = Passiv                            ' Flag zurücksetzen

   Reset Led_15_close_time

Return
'-- End Relais_beschattung_auf ------------------------------------------------


'--------------------------------------------------------------------------------
' REL - Gosub-Routine: Relais_beschattung_zu
' Subroutine zum kompletten Zufahren der Beschattung mit Richtungsumschaltung
'--------------------------------------------------------------------------------
Relais_beschattung_zu:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Relais_testmodus                                   ' Trace
      Print #2 , "Enter Relais_beschattung_zu"
   #endif                                                 ' Trace

   ' Prüfung ob Antrieb sich aktuell in AUF-Richtung bewegt
   If Beschattung_auf_flag = Aktiv Then
      ' Wenn JA Antrieb abschalten und Flag zurücksetzen
      Beschattung_auf_flag = Passiv
      Gosub Relais_auf_aus
      Waitms 500                                          ' Zum Schutz des Antriebs 500 ms warten
   End If                                                 ' Beschattung auf aktiv

   ' Antriebe ZU einschalten und Timerüberwachung aktivieren
   Gosub Relais_zu_an
   Beschattung_zu_flag = Aktiv
   Beschattung_zu_counter = Sectic_count + Rel_beschattung_delay

   Rain_autoauffahren = Passiv                            ' globales Flag zurücksetzen
   Wind_autoauffahren = Passiv                            ' globales Flag zurücksetzen
   Light_autoauffahren = Passiv                           ' globales Flag zurücksetzen

   Reset Led_11_open_time

Return
'-- End Relais_beschattung_zu ------------------------------------------------


'--------------------------------------------------------------------------------
' REL - Gosub-Routine: Relais_beschattung_stopp
' Subroutine zum Anhalten des aktuell Fahrvorgangs
'--------------------------------------------------------------------------------
Relais_beschattung_stop:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Relais_testmodus                                   ' Trace
      Print #2 , "Enter Relais_beschattung_stop"
   #endif                                                 ' Trace

   ' Antrieb ZU anhalten und Timerüberwachung zurücksetzen
   Beschattung_zu_flag = Passiv
   Gosub Relais_zu_aus

   ' Antrieb AUF anhalten und Timerüberwachung zurücksetzen
   Beschattung_auf_flag = Passiv
   Gosub Relais_auf_aus

Return
'-- End Relais_beschattung_stopp ----------------------------------------------


'--------------------------------------------------------------------------------
' Gosub-Routine: Beleuchtung_ein
```

```
' Subroutine schaltet Beleuchtung ein
'-------------------------------------------------------------------------------
Beleuchtung_ein:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Relais_testmodus                                    ' Trace
      Print #2 , "Enter Beleuchtung_ein"
   #endif                                                  ' Trace

   ' Relais einschalten
   Gosub Relais_licht_an

   ' Flag setzen
    Beleuchtung_flag = Aktiv

Return
'-- End Beleuchtung_ein -------------------------------------------------------


'-------------------------------------------------------------------------------
' Gosub-Routine: Beleuchtung_aus
' Subroutine schaltet Beleuchtung aus
'-------------------------------------------------------------------------------
Beleuchtung_aus:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Relais_testmodus                                    ' Trace
      Print #2 , "Enter Beleuchtung_aus"
   #endif                                                  ' Trace

   ' Relais einschalten
   Gosub Relais_licht_aus

   ' Flag setzen
    Beleuchtung_flag = Passiv

Return
'-- End Beleuchtung_aus -------------------------------------------------------


'-------------------------------------------------------------------------------
' Gosub-Routine: Beleuchtung_dauerlicht_ein
' Subroutine schaltet Dauerlicht ein
'-------------------------------------------------------------------------------
Beleuchtung_dauerlicht_ein:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Relais_testmodus                                    ' Trace
      Print #2 , "Enter Beleuchtung_dauerlicht_ein"
   #endif                                                  ' Trace

   ' Bemerkung: Dauerlicht ist eingeschaltet, wenn Relais nicht angezongen!!!!!

   ' Relais ausschalten
   Gosub Relais_dauerlicht_aus

   ' Flag setzen
   Dauerlicht_flag = Aktiv

Return
'-- End Beleuchtung_ein -------------------------------------------------------


'-------------------------------------------------------------------------------
' Gosub-Routine: Beleuchtung_dauerlicht_aus
' Subroutine schaltet Dauerlicht ein
'-------------------------------------------------------------------------------
Beleuchtung_dauerlicht_aus:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Relais_testmodus                                    ' Trace
      Print #2 , "Enter Beleuchtung_dauerlicht_aus"
   #endif                                                  ' Trace

   ' Bemerkung: Dauerlicht ist eingeschaltet, wenn Relais nicht angezongen!!!!!

   ' Relais ausschalten
   Gosub Relais_dauerlicht_an

   ' Flag setzen
```

```
   Dauerlicht_flag = Passiv

Return
'-- End Beleuchtung_aus ------------------------------------------------------


'--------------------------------------------------------------------------------
' Gosub-Routine: Sw_timer_sequencer
' Subroutine Abprüfen der in RUNNING befindlichen SW-Timer für Beschattung auf
' oder zu
'--------------------------------------------------------------------------------
Sw_timer_sequencer:

   ' *** Sequence [1] ***
   ' Überprüfung ob Timerüberwachung für Beschattung ZU aktiv
   If Beschattung_zu_flag = Aktiv Then

      ' Prüfung ob Timer abgelaufen
      If Sectic_count = Beschattung_zu_counter Then

         ' Traceausgabe wenn Sequencer Aufgabe [6] ausführt
         #if Sequencer_testmodus                              ' Trace
            Print #2 , "[1] Automatische Abschaltung ZU"
         #endif                                               ' Trace

         ' Wenn ja Beschattung ZU abschalten und Timerüberwachung deaktivieren
         Beschattung_zu_flag = Passiv
         Gosub Relais_zu_aus

         ' Ggf. Signalton ausgeben
         If Sound_flag = True Then
            Call Sound_beep(sound_beep_time_short)            ' Subroutine Beep aufrufen
            Waitms 40
            Call Sound_beep(sound_beep_time_short)            ' Subroutine Beep aufrufen
         End If                                               ' Signalton

         ' Prüfung von wo Funktions ausgeführt
         If Menue_inrelmenue_flag <> 0 Then
            ' Wenn Funktion von Menue ausgeführt an Umschaltung auf Stopbildschirm
            Menue_state = Loadlabel(men2231)                  ' Stopbildschirm
            Menue_state_renew = 1
            Gosub Menue_change_state_by_sub
         End If                                               ' Menü-Update

      End If                                                  ' Timer abgelaufen?
   End If                                                     ' Beschattung zu?


   ' *** Sequence [2] ***
   ' Überprüfung ob Timerüberwachung für Beschattung AUF aktiv
   If Beschattung_auf_flag = Aktiv Then

      ' Prüfung ob Timer abgelaufen
      If Sectic_count = Beschattung_auf_counter Then

         ' Traceausgabe wenn Sequencer Aufgabe [6] ausführt
         #if Sequencer_testmodus                              ' Trace
            Print #2 , "[2] Automatische Abschaltung AUF"
         #endif                                               ' Trace

         ' Wenn ja Beschattung AUF abschalten und Timerüberwachung deaktivieren
         Beschattung_auf_flag = Passiv
         Gosub Relais_auf_aus

         ' Ggf. Signalton ausgeben
         If Sound_flag = True Then
            Call Sound_beep(sound_beep_time_short)            ' Subroutine Beep aufrufen
            Waitms 40
            Call Sound_beep(sound_beep_time_short)            ' Subroutine Beep aufrufen
         End If                                               ' Signalton

         ' Prüfung von wo Funktions ausgeführt
         If Menue_inrelmenue_flag <> 0 Then
            Menue_state = Loadlabel(men2231)                 'Startbildschirm
            Menue_state_renew = 1
            Gosub Menue_change_state_by_sub
         End If                                               ' Menü-Update

      End If                                                  ' Timer abgelaufen?
   End If                                                     ' Beschattung auf?
```

```
' *** Sequence [3] ***
' Überprüfung ob Timerüberwachung für Windsensor aktiv
If Wind_erfassung = True Then

   ' Überprüfung ob Zeitintervall abgelaufen
   If Sectic_count = Wind_erfassungs_counter Then

      ' Traceausgabe wenn Sequencer Aufgabe [6] ausführt
      #if Sequencer_testmodus                              ' Trace
         Print #2 , "[3] Wind messen"
      #endif                                               ' Trace

      ' Windgeschwindigkeit berechnen
      Gosub Wind_measurement

      ' Überprüfung auf maximale Windgeschwindigkeit
      If Wind_geschwindigkeit_kmh > Wind_kmh_max Then
         ' Maximale Windgeschwindigkeit neu setzen
         Wind_kmh_max = Wind_geschwindigkeit_kmh
         ' Werte im EEPROM updaten
         Ee_data_wind_kmh_max = Wind_kmh_max
      End If                                               ' Windgeschwindigkeit

      ' Auswerteintervall neu setzen
      Wind_erfassungs_counter = Wind_erfassungs_counter + Wind_auswertung

   End If                                                  ' Timer abgelaufen?
End If                                                     ' Winderfassung


' *** Sequence [4] ***
' Überprüfung ob Timerüberwachung für SHT71 aktiv
If Sht71_erfassung = True Then

   ' Überprüfung on Zeitintercall abgelaufen
   If Sectic_count = Sht71_erfassungs_counter Then

      ' Traceausgabe wenn Sequencer Aufgabe [6] ausführt
      #if Sequencer_testmodus                              ' Trace
         Print #2 , "[4] Temp&Hum messen"
      #endif                                               ' Trace

      ' Temperatur und Luftfeuchtigkeit berechnen und in globalen Variablen ablegen
      Gosub Sht71_perform_measurement
      Call Sht71_calc_measurements()

      ' Prüfung Temperatur-Wert auf Minimum
      If Sht71_t < Sht71_t_min Then
         Sht71_t_min = Sht71_t                             ' Gobale Variable zuweisen
         Ee_data_sht71_t_min = Sht71_t_min                 ' Wert in Eeprom speichern

         ' Ggf. Trace ausgeben
         #if Sequencer_testmodus                           ' Trace
            Print #2 , "- Update Sht71_t_min"
         #endif                                            ' Trace

      End If                                               ' Temperatur Minimum

      ' Prüfung Temperatur-Wert auf Maximum
      If Sht71_t > Sht71_t_max Then
         Sht71_t_max = Sht71_t                             ' Gobale Variable zuweisen
         Ee_data_sht71_t_max = Sht71_t_max                 ' Wert in Eeprom speichern

         ' Ggf. Trace ausgeben
         #if Sequencer_testmodus                           ' Trace
            Print #2 , "- Update Sht71_t_max"
         #endif                                            ' Trace

      End If                                               ' Temperatur Maximum

      ' Prüfung Feuchtigkeits-Wert auf Minimum
      If Sht71_rhtrue < Sht71_rhtrue_min Then
         Sht71_rhtrue_min = Sht71_rhtrue                   ' Gobale Variable zuweisen
         Ee_data_sht71_rhtrue_min = Sht71_rhtrue_min       ' Wert in Eeprom speichern

         ' Ggf. Trace ausgeben
         #if Sequencer_testmodus                           ' Trace
            Print #2 , "- Update Sht71_rhtrue_min"
```

```
            #endif                                              ' Trace

         End If                                                 ' Feuchtigkeit Minimum

         ' Prüfung Feuchtigkeits-Wert auf Maximum
         If Sht71_rhtrue > Sht71_rhtrue_max Then
            Sht71_rhtrue_max = Sht71_rhtrue                     ' Gobale Variable zuweisen
            Ee_data_sht71_rhtrue_max = Sht71_rhtrue_max         ' Wert in Eeprom speichern

            ' Ggf. Trace ausgeben
            #if Sequencer_testmodus                             ' Trace
               Print #2 , "- Update Sht71_rhtrue_max"
            #endif                                              ' Trace

         End If                                                 ' Feuchtigkeit Maximum

         ' Prüfung auf Urinitialisierung
         If Sht71_rhtrue_min = 0.0 Then
            If Sht71_t_min = 0.0 Then
               ' JA; beiden Werte sind noch nicht initialisiert, stehen auf NULL
               Sht71_t_min = Sht71_t_max                        ' Min-Wert = Max-Wert setzen
               Ee_data_sht71_t_min = Sht71_t_min                ' Wert in Eeprom speichern
               Sht71_rhtrue_min = Sht71_rhtrue_max              ' Min-Wert = Max-Wert setzen
               Ee_data_sht71_rhtrue_min = Sht71_rhtrue_min      ' Wert in Eeprom speichern

               ' Ggf. Trace ausgeben
               #if Sequencer_testmodus                          ' Trace
                  Print #2 , "- Update aller Min-Werte"
               #endif                                           ' Trace

            End If                                              ' Urinitialisierung Temperatur
         End If                                                 ' Urinitialisierung Feuchtigkeit

         ' Auswerteintervall neu setzen
         Sht71_erfassungs_counter = Sht71_erfassungs_counter + Sht71_auswertung

      End If                                                    ' Timer abgelaufen
   End If                                                       ' Temp und Feuchtigkeit


   ' *** Sequence [5] ***
   ' Überprüfung ob ein Menüinhalt upgedated werden muss
   If Menue_insensmenue_flag <> 0 Then

      ' Traceausgabe wenn Sequencer Aufgabe [5] ausführt
      #if Sequencer_testmodus                                   ' Trace
         Print #2 , "[5] Menue aktualisieren"
      #endif                                                    ' Trace

      ' JA, es gibt einen aktiven Inhalt => Inhalt anzeigen durch Aufruf Subroutine
      Gosub Menue_call_subroutine

   End If                                                       ' Menü-Update


   ' *** Sequence [6] ***
   ' Überprüfung ob Licht gemessen werden soll
   If Light_erfassung = True Then
      ' JA, das Statusflag ist auf EIN geschaltet

      ' Überprüfen on Zeitintervall für Messwerterfassung abgelaufen ist
      If Sectic_count = Light_erfassungs_counter Then
         ' JA, Counter ist abgelaufen

         ' Traceausgabe wenn Sequencer Aufgabe [6] ausführt
         #if Sequencer_testmodus                                ' Trace
            Print #2 , "[6] Licht messen"
         #endif                                                 ' Trace

         ' Licht messen und in globale Varible übertragen
         Light_value = Light_measurement(light_messwert_zahl)

         ' Auswerteintervall neu setzen
         Light_erfassungs_counter = Light_erfassungs_counter + Light_auswertung_default
      End If                                                    ' Timer abgelaufen
   End If                                                       ' Lichterfassung


   ' *** Sequence [7] ***
   ' Überprüfen ob LED's für Tasten zurückgesetzt werden müssen
```

```
   If Led_key_flag = True Then
      ' JA, Flag ist True

      ' Prüfen ob auch Counter
      If Sectic_count > Led_key_counter Then

         ' Traceausgabe wenn Sequencer Aufgabe [7] ausführt
         #if Sequencer_testmodus                          ' Trace
            Print #2 , "[7] Led's zuruecksetzen"
         #endif                                           ' Trace

         ' LED's für Tasten wieder abschalten
         Led_array = Led_array And Led_key_mask
         ' Gosub Led_update_array                          ' LED-Update erfolgt zentral in Haupt-
schleife

         ' Überwachungsflag wieder zurücksetzen
         Led_key_flag = False

      End If                                              ' Timer abgelaufen
   End If                                                 ' LED


   ' *** Sequence [8] ***
   ' Überwachung des Datalogging-Mechanismus

   ' Überrüfung ob Datalogging überhaupt aktiv
   If Ee_datalogging_flag = True Then
      ' JA, Flag ist True

      ' Prüfen ob Counter auch abgelaufen bzw. erreicht
      If Sectic_count = Ee_erfassungs_counter Then
         ' JA, Count ist abgelaufen

         ' Traceausgabe wenn Sequencer Aufgabe [8] ausführt
         #if Sequencer_testmodus                          ' Trace
            Print #2 , "[8] Data-Logging"
         #endif                                           ' Trace

         Ee_erfassungs_counter = Sectic_count + Ee_auswertung       ' Auswerteintervall neu setzen

         ' Wenn Konstante gesetzt Trace Output schreiben
         #if Datalogging_testmodus                        ' Trace
            Print #2 , "Ueberwachung Datalogging:"
            Print #2 , "  aktueller Sectic :      " ; Sectic_count
            Print #2 , "  neu gesetzter Counter : " ; Ee_erfassungs_counter
         #endif                                           ' Trace

         ' Prüfen ob Datenüberlauf und ggf. Variable zurücksetzen und Overrun erhöhen
         If Ee_data_count = Ea_data_loging_endadress Then

            Ee_data_count = 0                             ' Counter auf Anfangsadresse setzen
            Incr Ee_data_overrun                          ' Overrun-Zähler erhöhen

            ' Ggf. Trace ausgeben
            #if Eeprom_testmodus                          ' Trace
               Print #2 , "Speicherueberlauf!"
               Print #2 , "Setzte Data Count auf " ; Hex(ee_data_count)
               Print #2 , "Erhoehen Overrun auf " ; Hex(ee_data_overrun)
            #endif                                        ' Trace

            ' Overrun-Zähler gleich speichern
            Call Ee_write_eeprom_byte(ee_data_overrun_adress , Ee_data_overrun)

         End If                                           ' Datenüberlauf

         ' Adresse zum Schreiben berechnen
         Temp_word_1 = Ea_data_loging_length * Ee_data_count
         Temp_word_1 = Temp_word_1 + Ee_data_loging_start_adress

         ' Ggf. Trace ausgeben
         #if Eeprom_testmodus                             ' Trace
            Print #2 , "Berechnete Speicheradresse : " ; Hex(temp_word_1)
         #endif                                           ' Trace

         ' Anzahl gespeicherter Daten um ein erhöhen
         Incr Ee_data_count

         ' Ggf. Trace ausgeben
         #if Eeprom_testmodus                             ' Trace
```

```
            Print #2 , "Erhoehe Data Count auf " ; Hex(ee_data_count)
         #endif                                                ' Trace

         ' Datenstruktur zum Speichern vorbereiten

         ' Byte(1) Ee_data_hour
         Ee_data_logging_struct(1) = _hour

         ' Byte(2) Ee_data_minute
         Ee_data_logging_struct(2) = _min

         ' Byte(3) Ee_data_day
         Ee_data_logging_struct(3) = _day

         ' Byte(4) Ee_data_month [Bit7..4]
         ' Byte(4) Ea_data_light [Bit3..0]
         Temp_byte_1 = _month
         Shift Temp_byte_1 , Left , 4
         Temp_byte_1 = Temp_byte_1 And &B11110000
         Temp_byte_2 = Light_measurement(light_messwert_zahl)
         Temp_byte_2 = Temp_byte_2 And &B00001111
         Ee_data_logging_struct(4) = Temp_byte_1 + Temp_byte_2

         ' Byte(5) Ee_data_year
         Temp_byte_1 = _year
         Ee_data_logging_struct(5) = Temp_byte_1

         ' Byte(6) Ee_data_rain  [Bit7]
         ' Byte(6) Ee_data_wind  [Bit6..0]
         Temp_byte_1 = Wind_geschwindigkeit_kmh                ' Casting von Single auf Byte
         Temp_byte_1 = Temp_byte_1 And &B01111111
         Temp_byte_2 = Rain_measurement()
         If Temp_byte_2 = Rain_yes Then
            Set Temp_byte_1.7                                  ' Setzte Bit 7 auf "1"
         End If                                                ' Regen
         Ee_data_logging_struct(6) = Temp_byte_1

         ' Byte(7) Ee_data_temp
         Temp_integer_1 = Sht71_t                              ' Casting von Single auf Integer
         Ee_data_logging_struct(7) = Low(temp_integer_1)

         ' Byte(8) Ee_data_humidity
         Temp_byte_1 = Sht71_rhtrue                            ' Casting von Single auf Byte
         Ee_data_logging_struct(8) = Temp_byte_1

         ' Ggf. Trace ausgeben
         #if Eeprom_testmodus                                  ' Trace
            Print #2 , "Daten-Struct : " ;
            Print #2 , Hex(ee_data_logging_struct(1)) ; " " ;
            Print #2 , Hex(ee_data_logging_struct(2)) ; " " ;
            Print #2 , Hex(ee_data_logging_struct(3)) ; " " ;
            Print #2 , Hex(ee_data_logging_struct(4)) ; " " ;
            Print #2 , Hex(ee_data_logging_struct(5)) ; " " ;
            Print #2 , Hex(ee_data_logging_struct(6)) ; " " ;
            Print #2 , Hex(ee_data_logging_struct(7)) ; " " ;
            Print #2 , Hex(ee_data_logging_struct(8))
         #endif                                                ' Trace

         ' Datenstruktur speichern
         Call Ee_write_eeprom_struct(temp_word_1)

         ' aktuellen Data Count speichern
         Call Ee_write_eeprom_word(ee_data_count_adress , Ee_data_count)

         ' Ggf. Signalton ausgeben
         If Sound_flag = True Then
            Call Sound_beep(sound_beep_time_short)             ' Subroutine Beep aufrufen
         End If                                                ' Signalton

      End If                                                   ' Timer abgelaufen?
   End If                                                      ' Datenlogging


   ' *** Sequence [9] ***
   ' Datenlogging ausgeben

   ' Überprüfung ob Datenlogging ausgegeben werden muss
   If Ee_output_flag = True Then
      ' JA, Datenlogging-Ausgabe wurde angefordert
```

```
' Traceausgabe wenn Sequencer Aufgabe [9] ausführt
#if Sequencer_testmodus                                 ' Trace
   Print #2 , "[9] Logging-Data Output"
#endif                                                  ' Trace


' Überprüfen ob Ende der Daten schon erreicht ist
If Ee_output_counter < Ee_output_endadress Then
   ' NEIN, Ende ist noch nicht erreicht

   ' Adresse zum Lesen berechnen
   Temp_word_2 = Ea_data_loging_length * Ee_output_counter
   Temp_word_2 = Temp_word_2 + Ee_data_loging_start_adress

   ' Indizierte Datenstruktur aus EEPROM lesen
   Call Ee_read_eeprom_struct(temp_word_2)

   ' Ggf. Trace ausgeben
   #if Eeprom_testmodus                                 ' Trace
      Print #2 , "Lese von Adresse " ; Hex(temp_word_2)
      Print #2 , "Daten-Struct : " ;
      Print #2 , Hex(ee_data_logging_struct(1)) ; " " ;
      Print #2 , Hex(ee_data_logging_struct(2)) ; " " ;
      Print #2 , Hex(ee_data_logging_struct(3)) ; " " ;
      Print #2 , Hex(ee_data_logging_struct(4)) ; " " ;
      Print #2 , Hex(ee_data_logging_struct(5)) ; " " ;
      Print #2 , Hex(ee_data_logging_struct(6)) ; " " ;
      Print #2 , Hex(ee_data_logging_struct(7)) ; " " ;
      Print #2 , Hex(ee_data_logging_struct(8))
   #endif                                               ' Trace

   ' Gelesene Daten interpretieren und ausgeben

   ' Format: nr;day.month.year;hour:minute;light;rain;wind;temp;hum
   Temp_word_2 = Ee_output_counter + 1
   Print #2 , Temp_word_2 ; ";" ;

   ' Byte(3) Ee_data_day
   Print #2 , Ee_data_logging_struct(3) ; "." ;

   ' Byte(4) Ee_data_month [Bit7..4]
   Temp_byte_1 = Ee_data_logging_struct(4)
   Shift Temp_byte_1 , Right , 4
   Temp_byte_1 = Temp_byte_1 And &B00001111
   Print #2 , Temp_byte_1 ; "." ;

   ' Byte(5) Ee_data_year
   Temp_word_2 = Ee_data_logging_struct(5) + 2000
   Print #2 , Temp_word_2 ; ";" ;

   ' Byte(1) Ee_data_hour
   Print #2 , Ee_data_logging_struct(1) ; ":" ;

   ' Byte(2) Ee_data_minute
   Print #2 , Ee_data_logging_struct(2) ; ";" ;

   ' Byte(4) Ea_data_light [Bit3..0]
   Temp_byte_1 = Ee_data_logging_struct(4) And &B00001111
   Print #2 , Temp_byte_1 ; ";" ;

   ' Byte(6) Ee_data_rain  [Bit7]
   Temp_byte_1 = Ee_data_logging_struct(6)
   Shift Temp_byte_1 , Right , 7
   Temp_byte_1 = Temp_byte_1 And &B00000001
   Print #2 , Temp_byte_1 ; ";" ;

   ' Byte(6) Ee_data_wind  [Bit6..0]
   Temp_byte_1 = Ee_data_logging_struct(6) And &B01111111
   Print #2 , Temp_byte_1 ; ";" ;

   ' Byte(7) Ee_data_temp
   Temp_byte_1 = Ee_data_logging_struct(7)
   If Temp_byte_1 > 127 Then
      Temp_word_2 = Temp_byte_1 + &HFF00
   Else
      Temp_word_2 = Temp_byte_1
   End If
   Temp_integer_1 = Temp_word_2
   Print #2 , Temp_integer_1 ; ";" ;
```

```
         ' Byte(8) Ee_data_humidity
         Print #2 , Ee_data_logging_struct(8)

         ' Ausgabezähler erhöhen
         Incr Ee_output_counter

      Else                                                  ' Logging-Counter

         ' JA, Ende erreicht. Noch Endzeile ausgeben und fertig
         Print #2 , "Ende der Daten"
         Ee_output_flag = False

         ' Ggf. Signalton ausgeben
         If Sound_flag = True Then
            Call Sound_beep(sound_beep_time_short)          ' Subroutine Beep aufrufen
            Waitms 40
            Call Sound_beep(sound_beep_time_short)          ' Subroutine Beep aufrufen
         End If                                             ' Signalton

      End If                                                ' Logging-Counter
   End If                                                   ' Daten ausgeben


   ' *** Sequence [10] ***
   ' Screensaver überprüfen

   ' Überprüfung ob Screensaver eigeschaltet
   If Lcd_screensaver = True Then
      ' JA, Screensaver ist eingeschaltet

      ' Prüfen ob der Screen bereits aktiv ist
      If Lcd_screensaver_aktiv = False Then
         ' Nein, Screensaver ist noch nicht abgelaufen

         ' Ist es Zeit für den Screensaver zu handeln?
         If Lcd_screensaver_counter = Sectic_count Then
            ' JA, Zeit abgelaufen

            ' Traceausgabe wenn Sequencer Aufgabe [10] ausführt
            #if Sequencer_testmodus                         ' Trace
               Print #2 , "[10] Screensaver"
            #endif                                          ' Trace

            ' Helligkeitswert retten
            Lcd_pwmwert_buffer = Lcd_pwmwert

            ' LCD dunkel
            Call Lcd_set_helligkeit(lcd_helligkeit_aus)

            ' Display ausschalten
            Gosub Lcd_display_off

            ' Flag setzen
            Lcd_screensaver_aktiv = True

            ' Ggf. Trace ausgeben
            #if Screensaver_testmodus                       ' Trace
               Print #2 , "PWM-Wert        : " ; Lcd_pwmwert
               Print #2 , "PWM-Wert Buffer : " ; Lcd_pwmwert_buffer
               Print #2 , "LCD Screensaver aktiv"
            #endif                                          ' Trace

         End If                                             ' Timer abgelaufen
      End If                                                ' Screensaver aktiv
   End If                                                   ' Screensaver eingeschaltet

Return
'-- End Sw_timer_sequencer ------------------------------------------------------


' ********************
' * DIM:  Dimmer     *
' ********************

'-------------------------------------------------------------------------------
' DIM - Gosub-Routine: Dimmer_konfig
' Subroutine zur Konfiguration des Dimmers
'-------------------------------------------------------------------------------
Dimmer_konfig:
```

```
   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Dimmer_testmodus                                    ' Trace
      Print #2 , "Enter Dimmer_konfig"
   #endif                                                  ' Trace

   ' PortE.5 als Eingang mit aktivem PullUp konfigurieren
   Config Null_detect = Input
   Null_detect_pin = 1

   ' PortD.4 als Ausgang definieren und Triac ausschalten
   Config Ignition = Output
   Triac_start = 0


   ' Der Interrupt selbst kann nicht in der Gosub-Routine initialisiert werden
   ' sondern muss im Rumpf des Gesamtprogramms initialisiert werden.

   ' Dimmerhelligkeit auf Default einstellen
   Dim_helligkeitsstufe = Dim_helligkeit_default

   ' Triac Zündzeitpunkte aus Tabelle in Datenstruktur laden
   Restore Triac_zuendtabelle
   For Dim_index_counter = 1 To Dim_helligkeitstabellenwerte

      Read Triac_zuenszeitpunkte(dim_index_counter)

      ' Wenn Konstante gesetzt Trace Output schreiben
      #if Dimmer_testmodus                                 ' Trace
         Print #2 , "Zuendwert " ; Dim_index_counter ; " : " ; Tri-
ac_zuenszeitpunkte(dim_index_counter)
      #endif                                               ' Trace

   Next Dim_index_counter                                  ' Helligkeitsindex

Return
'-- End Dimmer_konfig -------------------------------------------------------


'--------------------------------------------------------------------------
' DIM - Gosub-Routine: Dimmer_heller
' Subroutine zum Erhöhen der Dimmerhelligkeit, Verschiebung Index Zündzeitpunkt
'--------------------------------------------------------------------------
Dimmer_heller:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Dimmer_testmodus                                    ' Trace
      Print #2 , "Enter Dimmer_heller"
   #endif                                                  ' Trace

   If Dim_helligkeitsstufe < Dim_helligkeit_max Then
      Incr Dim_helligkeitsstufe
   End If                                                  ' Helligkeitsstufe

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Dimmer_testmodus                                    ' Trace
      Print #2 , "Helligkeit ist : " ; Dim_helligkeitsstufe
   #endif                                                  ' Trace

Return
'-- End Dimmer_heller -------------------------------------------------------


'--------------------------------------------------------------------------
' DIM - Gosub-Routine: Dimmer_dunkler
' Subroutine zum reduzieren der Dimmerhelligkeit, Verschiebung Index
' Zündzeitpunkt
'--------------------------------------------------------------------------
Dimmer_dunkler:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Dimmer_testmodus                                    ' Trace
      Print #2 , "Enter Dimmer_dunkler"
   #endif                                                  ' Trace

   If Dim_helligkeitsstufe > Dim_helligkeit_min Then
      Decr Dim_helligkeitsstufe
   End If                                                  ' Helligkeitsstufe

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Dimmer_testmodus                                    ' Trace
```

```
      Print #2 , "Helligkeit ist : " ; Dim_helligkeitsstufe
   #endif                                                  ' Trace

Return
'-- End Dimmer_dunkler ------------------------------------------------------


'-------------------------------------------------------------------------------
' Gosub-Routine: Dimmer_free_running
' Subroutine zum Setzen von Parametern für freien Dimmerlauf, gesteuert über
' Nulldurchgangskennung
'-------------------------------------------------------------------------------
Dimmer_free_running:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Dimmer_testmodus                                    ' Trace
      Print #2 , "Enter Dimmer_free_running"
   #endif                                                  ' Trace

   ' Prüfung ob free_running aktiv
   If Free_running_flag = Passiv Then
      ' JA, Free_runing ist aus und wird eingeschaltet
      Free_running_flag = Aktiv

      ' Wenn Konstante gesetzt Trace Output schreiben
      #if Dimmer_testmodus                                 ' Trace
         Print #2 , "Free Running ist aktiv"
      #endif                                               ' Trace

   Else                                                    ' Dimmer aktiv
      ' NEIN, Free_running ist ein und wird ausgeschaltet
      Free_running_flag = Passiv

      ' Wenn Konstante gesetzt Trace Output schreiben
      #if Dimmer_testmodus                                 ' Trace
         Print #2 , "Free Running ist passiv"
      #endif                                               ' Trace

   End If                                                  ' Dimmer aktiv


Return
'-- End Dimmer_free_running -------------------------------------------------


' ********************
' * WIND: Windsensor *
' ********************

'-------------------------------------------------------------------------------
' WIND - Gosub-Routine: Wind_konfig
' Subroutine zur Konfiguration der für die Windmessung notwendigen Datenstruk-
' turen, Timer usw.
'-------------------------------------------------------------------------------
Wind_konfig:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Wind_testmodus                                      ' Trace
      Print #2 , "Enter Wind_konfig"
   #endif                                                  ' Trace

   Wind_sensorimpulse = 0                                  ' Anzahl Sensorimpulse auf 0
   Wind_auswertung = Wind_auswertung_default               ' Auswerteintervall mit Defaultwert
laden
   Wind_kmh_max = 0.0                                      ' Maximaler Wind initialisieren

   ' Kritische Windgeschwindigkeit mit Default-Wert laden
   Wind_beschattung_auf = Wind_beschattung_auf_default

Return
'-- End Wind_konfig ----------------------------------------------------------


'-------------------------------------------------------------------------------
' WIND - Gosub-Routine: Wind_measurement
' Subroutine zur Berechung der aktuellen Windgeschwindigkeit im Auswerteinter-
' vall für km/ ud m/s.
' Die Ablage der errechneten Werte erfolgt in globalen Variablen
'-------------------------------------------------------------------------------
Wind_measurement:
```

```
   ' Traceausgabe schreiben wenn Testmodus gesetzt ist
   #if Wind_testmodus                                          ' Trace
      Print #2 , "Enter Wind_measurement"
      Print #2 , ""
      Print #2 , "Erfassung     : " ; Wind_erfassung
      Print #2 , "Sectic_count  : " ; Sectic_count
      Print #2 , "Sensorimpulse : " ; Wind_sensorimpulse
   #endif                                                      ' Trace

   ' Interrupt während Auslesen von Zähler wegen "Kollisionsgefahr" sperren
   Disable Int6                                                ' Interrupt 6 sperren

   Temp_byte_1 = Wind_sensorimpulse                            ' Variable auslesen
   Wind_sensorimpulse = 0                                      ' Variable zurücksetzen

   ' Interrupt nur freigeben wenn Erfassung auch gewünscht
   If Wind_erfassung = True Then
      Enable Int6                                              ' Interrupt 6 wieder freigeben
   End If                                                      ' Winderfassung

   ' Berechung der Windgeschwindigkeit in km/h

   ' Messwert über Zeit auf eine Sekunde normieren
   Wind_geschwindigkeit_kmh = Temp_byte_1 / Wind_auswertung
   Wind_geschwindigkeit_kmh = Wind_geschwindigkeit_kmh * Wind_impuls_geschwindigkeit

   ' Umrechung von km/h in m/s: Wert * 1000 / 3600 => Wert / 3,6
   Wind_geschwindigkeit_ms = Wind_geschwindigkeit_kmh / 3.6


   ' Traceausgabe schreiben wenn Testmodus gesetzt ist
   #if Wind_testmodus                                          ' Trace

      ' Windgeschwindigkeit in km/h ausgeben
      Temp_string_1 = Fusing(wind_geschwindigkeit_kmh , "###.##")
      Print #2 , "Wind in km/h  : " ; Temp_string_1

      ' Windgeschwindigkeit in m/s ausgeben
      Temp_string_1 = Fusing(wind_geschwindigkeit_ms , "###.##")
      Print #2 , "Wind in m/s   : " ; Temp_string_1

      Print #2 , ""
   #endif                                                      ' Trace

Return
'-- End Wind_measurement ------------------------------------------------------


'-------------------------------------------------------------------------------
' WIND - Gosub-Routine: Wind_erfassung_an
' Subroutine aktiviert den Interrupt zur Erfassung der "Windimpulse"
'-------------------------------------------------------------------------------
Wind_erfassung_an:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Wind_testmodus                                          ' Trace
      Print #2 , "Enter Wind_erfassung_an"
   #endif                                                      ' Trace

   ' Interrupt freigeben -> Erfassung aktiv
   Enable Int6
   Wind_erfassung = True
   Wind_sensorimpulse = 0

Return
'-- End Wind_erfassung_an ------------------------------------------------------


'-------------------------------------------------------------------------------
' WIND - Gosub-Routine: Wind_erfassung_aus
' Subroutine deaktiviert den Interrupt zur Erfassung der "Windimpulse"
'-------------------------------------------------------------------------------
Wind_erfassung_aus:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Wind_testmodus                                          ' Trace
      Print #2 , "Enter Wind_erfassung_aus"
   #endif                                                      ' Trace
```

```
   ' Interrupt sperren -> Erfassung ausgeschaltet
   Disable Int6
   Wind_erfassung = False
   Wind_sensorimpulse = 0

Return
'-- End Wind_erfassung_an -----------------------------------------------------


' ********************
' * SHT71:  Sensor   *
' ********************

'-------------------------------------------------------------------------------
' SHT71 - Gosub-Routine: Sht71_connection_reset_sequence
' Bei StartUp, Verbindungsverlust, Reset und CRC-Fehler muss das serielle
' Interface zurückgesetzt werden. Hierzu dient diese Routine.
' Die Routine sende die Im Datenblatt beschriebene connection reset sequence.
' Hierzu wird während DATA=high SCK mindestens 9 Mal getoggelt. Anschließend
' wird eine Transmission Start Sequenz gesendet.
'-------------------------------------------------------------------------------
Sht71_connection_reset_sequence:

   ' Absicherung dass DATA und SCK Pin's auf Output stehen da Vorgeschichte unbekannt
   Config Temp_sck = Output
   Config Temp_data = Output

   ' Initial State
   Set Sht71_dataout
   Reset Sht71_sck

   ' Connection Reset Sequence
   '          _____________________________________________________         _______
   ' Data:                                                        |_______|
   '        __    __    __    __    __    __    __    __    __    ___     ___
   ' SCK:  __|  |__|  |__|  |__|  |__|  |__|  |__|  |__|  |__|  |_____|   |___|   |_____

   For Sht71_temp_byte1 = 1 To 9
      Set Sht71_sck
      ' Waitus 2
      Reset Sht71_sck
      ' Waitus 2
   Next Sht71_temp_byte1                                    ' Bitcounter

   Gosub Sht71_transmission_start

Return
'-- End Sht71_connection_reset_sequence ---------------------------------------


'-------------------------------------------------------------------------------
' SHT71 - Gosub-Routine: Sht71_transmission_start
' Diese Routine initiiert das Senden von Kommandos an den SHT71. Jede Aussen-
' dung von Daten vom ATmega zum SHT71 muss mit dieser Sequenz eingeleitet
' werden
'-------------------------------------------------------------------------------
Sht71_transmission_start:

   ' Absicherung dass DATA und SCK Pin's auf Output stehen da Vorgeschichte unbekannt
   Config Temp_sck = Output
   Config Temp_data = Output

   ' Initial state
   Set Sht71_dataout
   Reset Sht71_sck
   $asm : Nop : $end Asm

   ' Transmission Start Sequence
   '        _____         _______
   ' Data:       |_______|
   '           ___     ___
   ' SCK:  ___|   |___|   |_____
   '
   Set Sht71_sck
   $asm : Nop : $end Asm
   Reset Sht71_dataout
   $asm : Nop : $end Asm
   Reset Sht71_sck
   $asm : Nop : Nop : Nop : $end Asm
   Set Sht71_sck
```

```
   $asm : Nop : $end Asm
   Set Sht71_dataout
   $asm : Nop : $end Asm
   Reset Sht71_sck

Return
'-- End Sht71_transmission_start ---------------------------------------------


'--------------------------------------------------------------------------------
' SHT71 - Gosub-Routine: Sht71_perform_measurement
' Diese Routine liest Luftfeuchtigkeit und Temperatur aus dem Sensor aus. Die
' Rohdaten werden in den entsprechenden globalen Variablen gespeichert.
' Anschließend wird eine Checksumme-Überprüfung durchgeführt.
'--------------------------------------------------------------------------------
Sht71_perform_measurement:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Sht71_testmodus = 1                                  ' Trace
      Print #2 , "Enter Sht71_perform_measurement"
   #endif                                                   ' Trace

   ' globales Fehler-Gesamtergebnis zurücksetzen
   Sht71_error_result = 0

   ' Da die Temperatur für die Rechnung der relativen Luftfeuchtigkeit Basis ist
   ' wird diese zuerst ausgelesen und in den globalen Variablen abgelegt.

   ' Temperatur auslesen
   Sht71_temp_byte1 = Sht71_measure(sht71_temp_word , Sht71_checksum , Sht71_modus_temp)

   ' Prüfung ob ein Fehler bei der Kommunikation aufgetreten ist.
   If Sht71_temp_byte1 = 0 Then
      ' NEIN

      ' CRC-Prüfung der Temperaturmessdaten
      Sht71_temp_byte1 = Sht71_crc_check(sht71_temp_word , Sht71_checksum , Sht71_modus_temp)

      ' Prüfung ob CRC NOK
      If Sht71_temp_byte1 = Crc_nok Then

         ' JA, Messung wahr fehlerhaft -> Fehlerzähler erhöhen
         Incr Sht71_error_result

         ' Wenn Konstante gesetzt Trace Output schreiben
         #if Sht71_testmodus = 1                            ' Trace
            Print #2 , "CRC-Fehler TEMP"
         #endif                                             ' Trace

      End If                                                ' CRC NOK
   Else                                                     ' Fehlerprüfung
      ' JA -> Fehlerzähler erhöhen
      Incr Sht71_error_result

      ' Wenn Konstante gesetzt Trace Output schreiben
      #if Sht71_testmodus = 1                               ' Trace
         Print #2 , "Kommunikationsfehler TEMP"
      #endif                                                ' Trace
   End If                                                   ' Fehlerprüfung

   ' Luftfeuchtigkeit auslesen
   Sht71_temp_byte1 = Sht71_measure(sht71_humi_word , Sht71_checksum , Sht71_modus_humi)

   ' Prüfung ob ein Fehler bei der Kommunikation aufgetreten ist.
   If Sht71_temp_byte1 = 0 Then
      ' NEIN

      ' CRC-Prüfung der Luftfeuchtigkeitsmessdaten
      Sht71_temp_byte1 = Sht71_crc_check(sht71_humi_word , Sht71_checksum , Sht71_modus_humi)

      ' Prüfung ob CRC NOK
      If Sht71_temp_byte1 = Crc_nok Then

         ' JA, Messung wahr fehlerhaft -> Fehlerzähler erhöhen
         Incr Sht71_error_result

         ' Wenn Konstante gesetzt Trace Output schreiben
         #if Sht71_testmodus = 1                            ' Trace
            Print #2 , "CRC-Fehler HUMI"
         #endif                                             ' Trace
```

```
      End If                                                ' CRC NOK
   Else                                                     ' Fehlerprüfung
      ' JA -> Fehlerzähler erhöhen
      Incr Sht71_error_result

      ' Wenn Konstante gesetzt Trace Output schreiben
      #if Sht71_testmodus = 1                               ' Trace
         Print #2 , "Kommunikationsfehler HUMI"
      #endif                                                ' Trace
   End If                                                   ' Fehlerprüfung


   ' Wenn ein Fehler aufgetreten ist dann Sensor zurücksetzen
   If Sht71_error_result > 0 Then

      ' Wenn Konstante gesetzt Trace Output schreiben
      #if Sht71_testmodus = 1                               ' Trace
         Print #2 , "Perform Sensor Reset wegen Fehlern"
      #endif                                                ' Trace

      ' Softreset mit Kommunikations-Restart durchführen
      Sht71_temp_byte1 = Sht71_softreset()

      ' Zeit "Sht71_communication_wait" warten nach Reset / PowerUp
      Waitms Sht71_communication_wait

      ' Nach Reset Kommunikation erneut starten
      Gosub Sht71_connection_reset_sequence

   End If                                                   ' Fehlerprüfung

Return
'-- End Sht71_perform_measurement ---------------------------------------------


' ********************
' * LED:  LED-Array  *
' ********************

'-------------------------------------------------------------------------------
' LED-Gosub-Routine: Led_clearlatch_set
' Subroutine setzt das Clear Latch Bit am Ausgang. Clear Input überschreibt
' unabhängig von den Input-Pins die Output-PIN's mit Low.
'
' Bemerkung:
' Das Clear Latch Bit muss auf HIGH gesetzt werden, damit überhaupt mit dem
' Latch gearbeitet werden kann! Beim Übergang von LOW auf HIGH werden die
' angelegten Datenbits auf dem jeweilig adressierten Nibble sofort aktiv und
' Sichtbar. Ggf. sollte ein korrektes Datenmuster und eine Adresse vor dem
' "loslassen" des Clear Latch Bits erfolgen.
'-------------------------------------------------------------------------------
Led_clearlatch_set:

   ' Ggf. Trace ausgeben
   #if Main_testmodus                                       ' Trace
      Print #2 , "Enter Led_clearlatch_set"
   #endif                                                   ' Trace

   ' Clear Latch auf HIGH setzen
   Set Led_clr_port

Return
'-- End Led_clearlatch_set ---------------------------------------------------


'-------------------------------------------------------------------------------
' LED-Gosub-Routine: Led_clearlatch_reset
' Subroutine setzt das Clear Latch Bit am Ausgang zurück. Clear Input
' überschreibt unabhängig von den Input-Pins die Output-PIN's mit Low.
'
' Bemerkung:
' Wird das Clear Latch auf LOW gesetzt so werden alle Ausgänge LOW. Die zuvor
' gespeicherten (gelatchten) Inhalte gehen verloren!
'-------------------------------------------------------------------------------
Led_clearlatch_reset:

   ' Ggf. Trace ausgeben
   #if Main_testmodus                                       ' Trace
      Print #2 , "Enter Led_clearlatch_reset"
```

```
   #endif                                                ' Trace

   ' Clear Latch auf LOW setzen
   Reset Led_clr_port

Return
'-- End Led_clearlatch_reset ------------------------------------------------


'--------------------------------------------------------------------------------
' LED-Gosub-Routine: Led_reset_array
' Subroutine löscht alle Dateninhalte und damit alle LED, setzt den Datenbus
' auf &B0000 und setzt Adresse &B00 für das Low-Nibble. Das Clear Latch Bit
' wird nach der Löschung automatisch auf HIGH und damit auf Normalbetrieb
' geschaltet.
' Achtung: Alle gelatchten Dateninhalte gehen verloren
'--------------------------------------------------------------------------------
Led_reset_array:

   ' Ggf. Trace ausgeben
   #if Main_testmodus                                    ' Trace
      Print #2 , "Enter Led_reset_array"
   #endif                                                ' Trace

   ' Clear Latch auf LOW setzen und damit alle Dateninhalte löschen
   Reset Led_clr_port

   ' Daten auf 0000
   Reset Led_bit0_port : Reset Led_bit1_port : Reset Led_bit2_port : Reset Led_bit3_port

   ' Adresse &B00 einstellen
   Reset Led_adr0_port : Reset Led_adr1_port

   ' Clear Latch auf HIGH setzen und damit Normalbetrieb freigeben
   Set Led_clr_port

Return
'-- End Led_reset_array ------------------------------------------------


'--------------------------------------------------------------------------------
' LED-Gosub-Routine: Led_update_array
' Subroutine übertrag die globale Array-Variable Led_array in das Latch und
' gibt damit den aktuellen Inhalt aus.
'--------------------------------------------------------------------------------
Led_update_array:

   ' Ggf. Trace ausgeben
   #if Main_testmodus                                    ' Trace
      Print #2 , "Enter Led_update_array"
   #endif                                                ' Trace

   ' Löschen aller LED's, zurücksetzen Datenbits und Adresse auf &B00 stellen
   Gosub Led_reset_array

   ' LED-Array zur weiteren Verarbeitung in temp. Variable laden
   Temp_word_1 = Led_array

   ' Schleife über alle 4 Nibbles des LED-Arrays
   For Temp_byte_1 = 1 To 4

      ' Adresse für jeweiliges Nibble sinstellen. &B00 ist bereits durch Reset
      ' eingestellt, daher keine Veränderung
      Select Case Temp_byte_1
         Case 2:
            ' Setzte Adresse &B01
            Set Led_adr0_port
         Case 3:
            ' Setzte Adresse &B10
            Set Led_adr1_port : Reset Led_adr0_port
            ' Achtung: Reihenfolge Bits setzen für Adresse hat ggf. Einfluss auf
            ' niedrigeres oder höherwertiges Nibble da Adressengleichheit!!!
         Case 4:
            ' Setzte Adresse &B11
            Set Led_adr0_port

      End Select                                         ' Nibble-Adresse

      ' Extrahiere jeweiliges Nibble aus 16-Bit LED-Array
      Temp_byte_2 = Temp_word_1 And &H000F
```

```
      ' Verschiebe extrahiertes Nibble auf korrekte Datenposition für Port-Byte
      Shift Temp_byte_2 , Left , 3

      ' Hole aktuelles Portregister inkl. aktuell eingesteller Adresse und
      ' maskiere Datenbits für Nibble aus
      Temp_byte_3 = Porta And &B10000111
      ' Addiere oben extrahiertes Nibble in Portbyte zur Ausgabe udn Portbyte
      ' schreiben -> Ausgabe der jeweiligen LED's erfolgt
      Porta = Temp_byte_3 + Temp_byte_2

      ' Verschiebe nächstes Nibble in 16-Bit LED-Array auf Arbeitsposition
      Shift Temp_word_1 , Right , 4

   Next Temp_byte_1                                    ' Schleife ggf. wiederholen

Return
'-- End Led_update_array -------------------------------------------------


' ********************************
' * EE: EEPROM (intern und extern *
' ********************************

'-------------------------------------------------------------------------------
' EE-Gosubroutine: Ee_init_internal_eeprom
' Subroutine überprüft die Datenkonsitenz aufgrund des verwendeten Versions-
' Index und führt ggf. eine Grundinitialisierung der Datenstrukturen durch.
' Die Settingdaten werden aus dem internen EEPROM ausgelesen.
'-------------------------------------------------------------------------------
Ee_init_internal_eeprom:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Eeprom_testmodus                                ' Trace
      Print #2 , "Enter Ee_init_internal_eeprom"
   #endif                                              ' Trace

   ' Zur Vollständigkeit hier nochmals die Datendefinitionen:
   ' Dim Ee_data_version_intern As Eram Byte At &H0000          ' Speicherstelle für Datenversion
   ' Dim Ee_data_debug As Eram Byte At &H0001                   ' Byte-Variable: Test_debug_flag
   ' Dim Ee_data_lcd_helligkeit As Eram Byte At &H0002          ' Byte-Variable: Lcd_helligkeit
   ' Dim Ee_data_lcd_kontrast As Eram Byte At &H0003            ' Byte-Variable: Lcd_kontrastwert
   ' Dim Ee_data_lcd_saver As Eram Byte At &H0004               ' Byte-Variable: Lcd_screensaver
   ' Dim Ee_data_wind_intervall As Eram Byte At &H0005          ' Byte-Variable: Wind_auswertung
   ' Dim Ee_data_wind_speed As Eram Byte At &H0006              ' Byte-Variable:
Wind_beschattung_auf
   ' Dim Ee_data_temp_intervall As Eram Byte At &H0007          ' Byte-Variable: Sht71_auswertung
   ' Dim Ee_data_dimmer As Eram Byte At &H0008                  ' Byte-Variable: Dim_flash_mode
   ' Dim Ee_data_dim_min As Eram Byte At &H0009                 ' Byte-Variable: Dim_helligkeit_min
   ' Dim Ee_data_dim_max As Eram Byte At &H000A                 ' Byte-Variable: Dim_helligkeit_max
   ' Dim Ee_data_sound As Eram Byte At &H000B                   ' Byte-Variable: Sound_flag
   ' Dim Ee_data_logging_flag As Eram Byte At &H000C            ' Byte-Variable:
Ee_datalogging_flag
   ' Dim Ee_data_logging_intervall As Eram Word At &H000D       ' Word-Variable: Ee_auswertung
   ' Dim Ee_data_conv_timeout As Eram Byte At &H000F            ' Byte-Variable:
Rel_beschattung_delay
   ' Dim Ee_data_conv_temp_flag As Eram Byte At &H0010          ' Byte-Variable: Rel_zu_temp_flag
   ' Dim Ee_data_conv_temperatur As Eram Byte At &H0011         ' Byte-Variable: Rel_auf_dunkelheit
   ' Dim Ee_data_conv_light_flag As Eram Byte At &H0012         ' Byte-Variable:
Rel_auf_dunkel_flag
   ' Dim Ee_data_conv_lichtwert As Eram Byte At &H0013          ' Byte-Variable: Rel_zu_temperatur
   ' Dim Ee_data_conv_hotkey As Eram Byte At &H0014             ' Byte-Variable:
Key_shortcut_status
   ' Dim Ee_data_fan As Eram Byte At &H0015                     ' Byte-Variable: Fan_sw_betrieb
   ' Dim Ee_data_fan_25_temp As Eram Byte At &H0016             ' Byte-Variable: Fan_25_temperatur
   ' Dim Ee_data_fan_50_temp As Eram Byte At &H0017             ' Byte-Variable: Fan_50_temperatur
   ' Dim Ee_data_fan_75_temp As Eram Byte At &H0018             ' Byte-Variable: Fan_75_temperatur
   ' Dim Ee_data_fan_100_temp As Eram Byte At &H0019            ' Byte-Variable: Fan_100_temperatur
   ' Dim Ee_data_dimmer_aktuell As Eram Byte At %H0020          ' Byte-Variable:
Dim_helligkeitsstufe
   ' Dim Ee_data_sht71_t_min As Eram Single At &H0021           ' Single-Variable: Sht71_t_min
   ' Dim Ee_data_sht71_t_max As Eram Single At &H0025           ' Single-Variable: Sht71_t_max
   ' Dim Ee_data_sht71_rhtrue_min As Eram Single At &H0029      ' Single-Variable: Sht71_rhtrue_min
   ' Dim Ee_data_sht71_rhtrue_max As Eram Single At &H002D      ' Single-Variable: Sht71_rhtrue_max
   ' Dim Ee_data_wind_kmh_max As Eram Single At &H0031          ' Single-Variable: Wind_kmh_max

   ' Aktuelle Datenversion des internen EEPROM lesen
   Temp_byte_1 = Ee_data_version_intern

   ' Ist die Datenversion korrekt
```

```
   If Temp_byte_1 <> Akt_internal_ee_version Then

      ' NEIN, aktuelle Daten neu speichern

      ' Wenn Konstante gesetzt Trace Output schreiben
      #if Eeprom_testmodus                                    ' Trace
         Print #2 , "Datenversion intern stimmt nicht"
         Print #2 , "Initialisiere internes EEPROM neu!"
      #endif                                                  ' Trace

      ' Datenstruktur internes EEPROM entspricht nicht dem aktuellen Stand
      ' => Datenstruktur neu anlegen bzw. neue Defaultwerte setzen.
      Gosub Ee_save_data_internal_eeprom

    Else                                                      ' Datenversion?

      ' JA, gespeicherte Werte laden und Variablen initialisieren

      ' Wenn Konstante gesetzt Trace Output schreiben
      #if Eeprom_testmodus                                    ' Trace
         Print #2 , "Datenversion intern stimmt"
         Print #2 , "Initialisiere Variablen aus internem EEPROM"
      #endif                                                  ' Trace

      Gosub Ee_load_data_internal_eeprom

   End If                                                     ' Datenversion?

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Eeprom_testmodus                                       ' Trace
      Print #2 , "Setting-Daten internes EEPROM:"
      Print #2 , "=============================="

      ' Wert 01
      Temp_byte_1 = Ee_data_version_intern
      Print #2 , "Version Datenstruktur :      " ; Hex(temp_byte_1)

      ' Wert 02
      Temp_byte_1 = Ee_data_debug
      Print #2 , "Debug-Modus (Flag) :         " ; Hex(temp_byte_1)

      ' Wert 03
      Temp_byte_1 = Ee_data_lcd_helligkeit
      Print #2 , "LCD Helligkeit :             " ; Temp_byte_1

      ' Wert 04
      Temp_byte_1 = Ee_data_lcd_kontrast
      Print #2 , "LCD Kontrast :               " ; Temp_byte_1

      ' Wert 05
      Temp_byte_1 = Ee_data_lcd_saver
      Print #2 , "LCD Screen Saver :           " ; Hex(temp_byte_1)

      ' Wert 06
      Temp_byte_1 = Ee_data_wind_intervall
      Print #2 , "Auswerteintervall Wind :     " ; Temp_byte_1

      ' Wert 07
      Temp_byte_1 = Ee_data_wind_speed
      Print #2 , "Grenzgeschwindigkeit Wind :  " ; Temp_byte_1

      ' Wert 08
      Temp_byte_1 = Ee_data_temp_intervall
      Print #2 , "Auswerteintervall SHT71 :    " ; Temp_byte_1

      ' Wert 09
      Temp_byte_1 = Ee_data_dimmer
      Print #2 , "Flash-Mode Dimmer :          " ; Hex(temp_byte_1)

      ' Wert 10
      Temp_byte_1 = Ee_data_dim_min
      Print #2 , "Dimmer Min-Helligkeit :      " ; Temp_byte_1

      ' Wert 11
      Temp_byte_1 = Ee_data_dim_max
      Print #2 , "Dimmer Max-Helligkeit :      " ; Temp_byte_1

      ' Wert 12
      Temp_byte_1 = Ee_data_sound
      Print #2 , "Soundausgabe :               " ; Hex(temp_byte_1)
```

```
    ' Wert 13
    Temp_byte_1 = Ee_data_logging_flag
    Print #2 , "Logginwerte schreiben :      " ; Hex(temp_byte_1)

    ' Wert 14
    Temp_word_1 = Ee_data_logging_intervall
    Print #2 , "Zeitintervall Loggingdaten : " ; Temp_word_1

    ' Wert 15
    Temp_byte_1 = Ee_data_conv_timeout
    Print #2 , "Motorabschaltung :           " ; Temp_byte_1

    ' Wert 16
    Temp_byte_1 = Ee_data_conv_temp_flag
    Print #2 , "Autom. ZU bei Temperatur :   " ; Hex(temp_byte_1)

    ' Wert 17
    Temp_byte_1 = Ee_data_conv_temperatur
    Print #2 , "Grenztemperatur :            " ; Temp_byte_1

    ' Wert 18
    Temp_byte_1 = Ee_data_conv_light_flag
    Print #2 , "Autom. Auf bei Licht :       " ; Hex(temp_byte_1)

    ' Wert 19
    Temp_byte_1 = Ee_data_conv_lichtwert
    Print #2 , "Grenzhelligkeit :            " ; Temp_byte_1

    ' Wert 20
    Temp_byte_1 = Ee_data_conv_hotkey
    Print #2 , "Hotkey Kreuztasten :         " ; Hex(temp_byte_1)

    ' Wert 21
    Temp_byte_1 = Ee_data_fan
    Print #2 , "Luefter ueber ATmega :       " ; Hex(temp_byte_1)

    ' Wert 22
    Temp_byte_1 = Ee_data_fan_25_temp
    Print #2 , "Temp. fuer 25% Luefter :     " ; Temp_byte_1

    ' Wert 23
    Temp_byte_1 = Ee_data_fan_50_temp
    Print #2 , "Temp. fuer 50% Luefter :     " ; Temp_byte_1

    ' Wert 24
    Temp_byte_1 = Ee_data_fan_75_temp
    Print #2 , "Temp. fuer 75% Luefter :     " ; Temp_byte_1

    ' Wert 25
    Temp_byte_1 = Ee_data_fan_100_temp
    Print #2 , "Temp. fuer 100% Luefter :    " ; Temp_byte_1

    ' Wert 26
    Temp_byte_1 = Ee_data_dimmer_aktuell
    Print #2 , "Aktuelle Helligkeitsstufe :  " ; Temp_byte_1

    ' Wert 27
    Temp_single_1 = Ee_data_sht71_t_min
    Temp_string_1 = Fusing(temp_single_1 , "###.##")
    Print #2 , "Minimale Temperatur :        " ; Temp_string_1

    ' Wert 28
    Temp_single_1 = Ee_data_sht71_t_max
    Temp_string_1 = Fusing(temp_single_1 , "###.##")
    Print #2 , "Maximale Temperatur :        " ; Temp_string_1

    ' Wert 29
    Temp_single_1 = Ee_data_sht71_rhtrue_min
    Temp_string_1 = Fusing(temp_single_1 , "###.##")
    Print #2 , "Minimale Feuchtigkeit :      " ; Temp_string_1

    ' Wert 30
    Temp_single_1 = Ee_data_sht71_rhtrue_max
    Temp_string_1 = Fusing(temp_single_1 , "###.##")
    Print #2 , "Maximale Feuchtigkeit :      " ; Temp_string_1

    ' Wert 31
    Temp_single_1 = Ee_data_wind_kmh_max
    Temp_string_1 = Fusing(temp_single_1 , "###.##")
```

```
      Print #2 , "Max. Windgeschwindigkeit :   " ; Temp_string_1

   #endif                                                    ' Trace

Return
'-- End Ee_init_internal_eeprom -----------------------------------------------


'-------------------------------------------------------------------------------
' EE-Gosubroutine: Ee_save_data_internal_eeprom
' Subroutine speichert alle Arbeitsvariablen in die angelegte Struktur des
' internen EEPROMs.
'-------------------------------------------------------------------------------
Ee_save_data_internal_eeprom:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Eeprom_testmodus                                      ' Trace
      Print #2 , "Enter Ee_save_data_internal_eeprom"
   #endif                                                    ' Trace

   Ee_data_version_intern = Akt_internal_ee_version
   Ee_data_debug = Test_debug_flag
   Ee_data_lcd_helligkeit = Lcd_helligkeit
   Ee_data_lcd_kontrast = Lcd_kontrastwert
   Ee_data_lcd_saver = Lcd_screensaver
   Ee_data_wind_intervall = Wind_auswertung
   Ee_data_wind_speed = Wind_beschattung_auf
   Ee_data_temp_intervall = Sht71_auswertung
   Ee_data_dimmer = Dim_flash_mode
   Ee_data_dim_min = Dim_helligkeit_min
   Ee_data_dim_max = Dim_helligkeit_max
   Ee_data_sound = Sound_flag
   Ee_data_logging_flag = Ee_datalogging_flag
   Ee_data_logging_intervall = Ee_auswertung
   Ee_data_conv_timeout = Rel_beschattung_delay
   Ee_data_conv_temp_flag = Rel_zu_temp_flag
   Ee_data_conv_temperatur = Rel_zu_temperatur
   Ee_data_conv_light_flag = Rel_auf_dunkel_flag
   Ee_data_conv_lichtwert = Rel_auf_dunkelheit
   Ee_data_conv_hotkey = Key_shortcut_status
   Ee_data_fan = Fan_sw_betrieb
   Ee_data_fan_25_temp = Fan_25_temperatur
   Ee_data_fan_50_temp = Fan_50_temperatur
   Ee_data_fan_75_temp = Fan_75_temperatur
   Ee_data_fan_100_temp = Fan_100_temperatur
   Ee_data_dimmer_aktuell = Dim_helligkeitsstufe
   Ee_data_sht71_t_min = Sht71_t_min
   Ee_data_sht71_t_max = Sht71_t_max
   Ee_data_sht71_rhtrue_min = Sht71_rhtrue_min
   Ee_data_sht71_rhtrue_max = Sht71_rhtrue_max
   Ee_data_wind_kmh_max = Wind_kmh_max

Return
'-- End Ee_save_data_internal_eeprom --------------------------------------------


'-------------------------------------------------------------------------------
' EE-Gosubroutine: Ee_load_data_internal_eeprom
' Subroutine lädt alle Arbeitsvariablen aus den angelegte Struktur des
' internen EEPROMs.
'-------------------------------------------------------------------------------
Ee_load_data_internal_eeprom:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Eeprom_testmodus                                      ' Trace
      Print #2 , "Enter Ee_load_data_internal_eeprom"
   #endif                                                    ' Trace

   Test_debug_flag = Ee_data_debug
   Lcd_helligkeit = Ee_data_lcd_helligkeit
   Lcd_kontrastwert = Ee_data_lcd_kontrast
   Lcd_screensaver = Ee_data_lcd_saver
   Wind_auswertung = Ee_data_wind_intervall
   Wind_beschattung_auf = Ee_data_wind_speed
   Sht71_auswertung = Ee_data_temp_intervall
   Dim_flash_mode = Ee_data_dimmer
   Dim_helligkeit_min = Ee_data_dim_min
   Dim_helligkeit_max = Ee_data_dim_max
   Sound_flag = Ee_data_sound
   Ee_datalogging_flag = Ee_data_logging_flag
```

```
   Ee_auswertung = Ee_data_logging_intervall
   Rel_beschattung_delay = Ee_data_conv_timeout
   Rel_zu_temp_flag = Ee_data_conv_temp_flag
   Rel_zu_temperatur = Ee_data_conv_temperatur
   Rel_auf_dunkel_flag = Ee_data_conv_light_flag
   Rel_auf_dunkelheit = Ee_data_conv_lichtwert
   Key_shortcut_status = Ee_data_conv_hotkey
   Fan_sw_betrieb = Ee_data_fan
   Fan_25_temperatur = Ee_data_fan_25_temp
   Fan_50_temperatur = Ee_data_fan_50_temp
   Fan_75_temperatur = Ee_data_fan_75_temp
   Fan_100_temperatur = Ee_data_fan_100_temp
   Dim_helligkeitsstufe = Ee_data_dimmer_aktuell
   Sht71_t_min = Ee_data_sht71_t_min
   Sht71_t_max = Ee_data_sht71_t_max
   Sht71_rhtrue_min = Ee_data_sht71_rhtrue_min
   Sht71_rhtrue_max = Ee_data_sht71_rhtrue_max
   Wind_kmh_max = Ee_data_wind_kmh_max

Return
'-- End Ee_load_data_internal_eeprom ------------------------------------------


'--------------------------------------------------------------------------------
' EE-Subroutine: Ee_init_external_eeprom
' Subroutine überprüft die Datenkonsitenz aufgrund des verwendeten Versions-
' Index und führt ggf. eine Grundinitialisierung der Datenstrukturen durch.
' Die für das Datenlogging notwendigen Basisdaten werden in globale Variablen
' geladen
'--------------------------------------------------------------------------------
Ee_init_external_eeprom:

   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Eeprom_testmodus                                    ' Trace
      Print #2 , "Enter Ee_init_external_eeprom"
   #endif                                                  ' Trace

   ' EEPROM-Version auslesen
   Ee_data_version_extern = Ee_read_eeprom_byte(ee_data_version_extern_adress)

   ' Prüfen ob gelesene Version der aktuellen Version entspricht
   If Ee_data_version_extern <> Akt_external_ee_version Then
      ' NEIN, EEPROM-Version ist ungleich aktueller Version

     ' Wenn Konstante gesetzt Trace Output schreiben
     #if Eeprom_testmodus                                  ' Trace
         Print #2 , "Datenversion extern stimmt nicht"
         Print #2 , "Initialisiere externes EEPROM neu"
      #endif                                               ' Trace

      ' Datenstruktur externes EEPROM entspricht nicht dem aktuellen Stand
      ' => Datenstruktur neu anlegen bzw. neue Defaultwerte setzen.

      Ee_data_version_extern = Akt_external_ee_version
      Ee_data_count = Ee_data_count_default
      Ee_data_overrun = Ee_data_overrun_default

      Call Ee_write_eeprom_byte(ee_data_version_extern_adress , Akt_external_ee_version)
      Waitms 10                                            'Wartezeit für sicheres Schreiben
      Call Ee_write_eeprom_word(ee_data_count_adress , Ee_data_count_default)
      Waitms 10                                            'Wartezeit für sicheres Schreiben
      Call Ee_write_eeprom_byte(ee_data_overrun_adress , Ee_data_overrun_default)

   Else                                                    ' Datenversion
      ' JA, Daten entsprechen den aktuellen Versionen

      ' Wenn Konstante gesetzt Trace Output schreiben
      #if Eeprom_testmodus                                 ' Trace
         Print #2 , "Datenversion extern stimmt"
         Print #2 , "Initialisiere Daten aus externem EEPROM"
      #endif                                               ' Trace

      ' Datenversion aus externem EEPROM auslesen
      Ee_data_version_extern = Ee_read_eeprom_byte(ee_data_version_extern_adress)
      ' Data-Counter auslesen
      Ee_data_count = Ee_read_eeprom_word(ee_data_count_adress)
      ' Overrun-Counter auslesen
      Ee_data_overrun = Ee_read_eeprom_byte(ee_data_overrun_adress)

   End If                                                  ' Datenversion
```

```
   ' Wenn Konstante gesetzt Trace Output schreiben
   #if Eeprom_testmodus                                    ' Trace
      Print #2 , "Setting-Daten externes EEPROM:"
      Print #2 , "=============================="
      Print #2 , "Version Datenstruktur : " ; Hex(ee_data_version_extern)
      Print #2 , "Datencounter  :         " ; Hex(ee_data_count)
      Print #2 , "Overrun-Zaehler :       " ; Hex(ee_data_overrun)
   #endif                                                  ' Trace

Return
'-- End Ee_init_external_eeprom ------------------------------------------------


'--------------------------------------------------------------------------------
' Devices schließend und ggf. "Terminate Programm execution"
'--------------------------------------------------------------------------------
Close #2
End                                                      'end program


'--------------------------------------------------------------------------------
' Definition von globalen Konstantenfeldern
'--------------------------------------------------------------------------------

'--------------------------------------------------------------------------------
' DATA:           Bedatung der Wochentage für DCF77 und Softclock
' Rückgabewerte:  Zeiger auf Wochentage
'--------------------------------------------------------------------------------

Wochentag:
Data "  " , "Mo" , "Di" , "Mi" , "Do" , "Fr" , "Sa" , "So"

'----- Ende Wochentag -----------------------------------------------------------


'--------------------------------------------------------------------------------
' DATA:           Bedatung der Tagesanzahl innerhalb der Monate
' Rückgabewerte:  Zeiger auf Tagesanzahl
'--------------------------------------------------------------------------------

Monate:
Data 0 , 31 , 28 , 31 , 30 , 31 , 30 , 31 , 31 , 30 , 31 , 30 , 31

'----- Ende Wochentag -----------------------------------------------------------


'--------------------------------------------------------------------------------
' DATA:           Bedatung der Triac-Zündtabelle
' Rückgabewerte:  Zeiger auf Zündzeitpunkte des Dimmers
'--------------------------------------------------------------------------------
Triac_zuendtabelle:
' Tabelle mit Originalwerten
' Data 156 , 139 , 131 , 125 , 120 , 116 , 111 , 108
' Data 103 , 100 , 95 , 92 , 89 , 86 , 83 , 80
' Data 77 , 73 , 70 , 67 , 64 , 61 , 56 , 53,
' Data 50 , 45 , 41 , 36 , 31 , 25 , 17 , 0
' Für Timer0 werden Einerkompliment-Werte (255-Wert) benötigt
Data 99 , 116 , 124 , 130 , 135 , 139 , 144 , 147
Data 152 , 155 , 160 , 163 , 166 , 169 , 172 , 175
Data 178 , 182 , 185 , 188 , 191 , 194 , 199 , 202
Data 205 , 210 , 214 , 219 , 224 , 230 , 238 , 250
'----- Ende Triac_zuendtabelle -------------------------------------------------


'--------------------------------------------------------------------------------
' DATA:           Bedatung der SHT71 Checksummen
' Rückgabewerte:  Zeiger auf CRC-Checksummen
'--------------------------------------------------------------------------------
Sht71_crc_table:
Data 0 , 49 , 98 , 83 , 196 , 245 , 166 , 151 , 185 , 136 , 219 , 234 , 125 , 76 , 31 , 46,
Data 67 , 114 , 33 , 16 , 135 , 182 , 229 , 212 , 250 , 203 , 152 , 169 , 62 , 15 , 92 , 109,
Data 134 , 183 , 228 , 213 , 66 , 115 , 32 , 17 , 63 , 14 , 93 , 108 , 251 , 202 , 153 , 168,
Data 197 , 244 , 167 , 150 , 1 , 48 , 99 , 82 , 124 , 77 , 30 , 47 , 184 , 137 , 218 , 235,
Data 61 , 12 , 95 , 110 , 249 , 200 , 155 , 170 , 132 , 181 , 230 , 215 , 64 , 113 , 34 , 19,
Data 126 , 79 , 28 , 45 , 186 , 139 , 216 , 233 , 199 , 246 , 165 , 148 , 3 , 50 , 97 , 80,
Data 187 , 138 , 217 , 232 , 127 , 78 , 29 , 44 , 2 , 51 , 96 , 81 , 198 , 247 , 164 , 149,
Data 248 , 201 , 154 , 171 , 60 , 13 , 94 , 111 , 65 , 112 , 35 , 18 , 133 , 180 , 231 , 214,
Data 122 , 75 , 24 , 41 , 190 , 143 , 220 , 237 , 195 , 242 , 161 , 144 , 7 , 54 , 101 , 84,
Data 57 , 8 , 91 , 106 , 253 , 204 , 159 , 174 , 128 , 177 , 226 , 211 , 68 , 117 , 38 , 23,
```

```
Data 252 , 205 , 158 , 175 , 56 , 9 , 90 , 107 , 69 , 116 , 39 , 22 , 129 , 176 , 227 , 210,
Data 191 , 142 , 221 , 236 , 123 , 74 , 25 , 40 , 6 , 55 , 100 , 85 , 194 , 243 , 160 , 145,
Data 71 , 118 , 37 , 20 , 131 , 178 , 225 , 208 , 254 , 207 , 156 , 173 , 58 , 11 , 88 , 105,
Data 4 , 53 , 102 , 87 , 192 , 241 , 162 , 147 , 189 , 140 , 223 , 238 , 121 , 72 , 27 , 42,
Data 193 , 240 , 163 , 146 , 5 , 52 , 103 , 86 , 120 , 73 , 26 , 43 , 188 , 141 , 222 , 239,
Data 130 , 179 , 224 , 209 , 70 , 119 , 36 , 21 , 59 , 10 , 89 , 104 , 255 , 206 , 157 , 172
'----- Ende SHT71_Checksummen -------------------------------------------------


'-------------------------------------------------------------------------------
' DATA:           Bedatung der Menue-State-Machine
' Rückgabewerte:  Zeiger auf Zustände , Unterprogramme, Displaytexte, usw.
' Beschreibung:   1. Zeile: für jede Taste genau eine Verzeigung ADR2
'                           (siehe auch Anzahl der Tasten (Key_buttons = xx))
'                           Reihenfolge der Tasten in Zeile 1:
'                                Hoch | Runter | Links | Rechts | Enter
'                 2. Zeile: Sprunglabel als ADR2 für ein ggf. anzuspringendes
'                           Unterprogramm
'                 3. Zeile: Binaermuster für Ausgabeflag von LCD-Zeile 1-3
'                 4. Zeile: Displaytext für LCD-Zeile 1 des akt. Status als DATA Feld
'                 5. Zeile: Displaytext für LCD-Zeile 2 des akt. Status als DATA Feld
'                 6. Zeile: Displaytext für LCD-Zeile 3 des akt. Status als DATA Feld
' Hinweis:        wenn nichts passieren soll wird das Label NULL eingetragen
'-------------------------------------------------------------------------------

' Null ist ein Dummy-Flag für Zustand und Gosub -> tue nichts
Null:
         Return


' Menü MEN1 - Hauptmenue ConvCtrl
Men1:
         ' Hoch       Runter      Links       Rechts      Enter
         Adr2 Men8 : Adr2 Men2 : Adr2 Null : Adr2 Men11 : Adr2 Null
         Adr2 Null
         Data &B00000111
         '     1234567890123456
         Data "1             1/8"
         Data "   Hauptmenue"
         Data "    ConvCtrl"

Men11:
            ' Hoch       Runter      Links       Rechts      Enter
            Adr2 Null : Adr2 Null : Adr2 Men1 : Adr2 Null : Adr2 Men1
            Adr2 Null
            Data &B00000111
            '     1234567890123456
            Data "1-1"
            Data "ConvCtrl    V1.0"
            Data "(C) M.Fulde 2008"


' Menü MEN2 - Beschattung
Men2:
         ' Hoch       Runter      Links       Rechts       Enter
         Adr2 Men1 : Adr2 Men3 : Adr2 Null : Adr2 Men21 : Adr2 Null
         Adr2 Menue_men2
         Data &B00000011
         '     1234567890123456
         Data "2             2/8"
         Data "  Beschattung"
         Data ""

Men21:
            ' Hoch        Runter       Links       Rechts        Enter
            Adr2 Men22 : Adr2 Men22 : Adr2 Men2 : Adr2 Men211 : Adr2 Men2
            Adr2 Null
            Data &B00000111
            '     1234567890123456
            Data "2-1           1/2"
            Data "Betriebszustand "
            Data "   anzeigen"

Men211:
               ' Hoch         Runter        Links        Rechts      Enter
               Adr2 Men217 : Adr2 Men212 : Adr2 Men21 : Adr2 Null : Adr2 Men2
               Adr2 Menue_men211
               Data &B00000111
               '     1234567890123456
               Data "2-1-1         1/7"
```

```
                Data " Zeitsteuerung"
                Data "Timeout:"

Men212:
                ' Hoch          Runter         Links         Rechts       Enter
                Adr2 Men211 : Adr2 Men213 : Adr2 Men21 : Adr2 Null : Adr2 Men2
                Adr2 Menue_men212
                Data &B00000111
                '     1234567890123456
                Data "2-1-2         2/7"
                Data " Zeitsteuerung"
                Data "Auf:"

Men213:
                ' Hoch          Runter         Links         Rechts       Enter
                Adr2 Men212 : Adr2 Men214 : Adr2 Men21 : Adr2 Null : Adr2 Men2
                Adr2 Menue_men213
                Data &B00000111
                '     1234567890123456
                Data "2-1-3         3/7"
                Data " Zeitsteuerung"
                Data "Zu:"

Men214:
                ' Hoch          Runter         Links         Rechts       Enter
                Adr2 Men213 : Adr2 Men215 : Adr2 Men21 : Adr2 Null : Adr2 Men2
                Adr2 Menue_men214
                Data &B00000111
                '     1234567890123456
                Data "2-1-4         4/7"
                Data " Auffahrgrund"
                Data "Wind:"

Men215:
                ' Hoch          Runter         Links         Rechts       Enter
                Adr2 Men214 : Adr2 Men216 : Adr2 Men21 : Adr2 Null : Adr2 Men2
                Adr2 Menue_men215
                Data &B00000111
                '     1234567890123456
                Data "2-1-5         5/7"
                Data " Auffahrgrund"
                Data "Regen:"

Men216:
                ' Hoch          Runter         Links         Rechts       Enter
                Adr2 Men215 : Adr2 Men217 : Adr2 Men21 : Adr2 Null : Adr2 Men2
                Adr2 Menue_men216
                Data &B00000111
                '     1234567890123456
                Data "2-1-6         6/7"
                Data " Auffahrgrund"
                Data "Licht:"

Men217:
                ' Hoch          Runter         Links         Rechts       Enter
                Adr2 Men216 : Adr2 Men211 : Adr2 Men21 : Adr2 Null : Adr2 Men2
                Adr2 Menue_men217
                Data &B00000111
                '     1234567890123456
                Data "2-1-7         7/7"
                Data "  Zufahrgrund"
                Data "Temp:"

Men22:
              ' Hoch         Runter        Links        Rechts         Enter
              Adr2 Men21 : Adr2 Men21 : Adr2 Men2 : Adr2 Men221 : Adr2 Men2
              Adr2 Null
              Data &B00000011
              '     1234567890123456
              Data "2-2           2/2"
              Data "    Antrieb"
              Data ""

Men221:
                ' Hoch          Runter         Links         Rechts          Enter
                Adr2 Men223 : Adr2 Men222 : Adr2 Men22 : Adr2 Men2211 : Adr2 Men2
                Adr2 Menue_men221
                Data &B00000011
                '     1234567890123456
                Data "2-2-1         1/3"
```

```
                Data "   Auffahren"
                Data ""

Men2211:
                   ' Hoch       Runter         Links         Rechts      Enter
                   Adr2 Null : Adr2 Men2221 : Adr2 Men221 : Adr2 Null : Adr2 Men2231
                   Adr2 Menue_men2211
                   Data &B00000101
                   '     1234567890123456
                   Data "   Auf  aktiv"
                   Data ""
                   Data "{002}=Zu      {005}=Stop"

Men222:
                ' Hoch         Runter        Links        Rechts         Enter
                Adr2 Men221 : Adr2 Men223 : Adr2 Men22 : Adr2 Men2221 : Adr2 Men2
                Adr2 Menue_men222
                Data &B00000011
                '     1234567890123456
                Data "2-2-2        2/3"
                Data "    Zufahren"
                Data ""

Men2221:
                   ' Hoch          Runter      Links         Rechts      Enter
                   Adr2 Men2211 : Adr2 Null : Adr2 Men222 : Adr2 Null : Adr2 Men2231
                   Adr2 Menue_men2221
                   Data &B00000101
                   '     1234567890123456
                   Data "    Zu aktiv"
                   Data ""
                   Data "{001}=Auf     {005}=Stop"

Men223:
                ' Hoch         Runter        Links        Rechts         Enter
                Adr2 Men222 : Adr2 Men221 : Adr2 Men22 : Adr2 Men2231 : Adr2 Men2
                Adr2 Menue_men223
                Data &B00000011
                '     1234567890123456
                Data "2-2-3        3/3"
                Data "  Antrieb Stop"
                Data ""

Men2231:
                   ' Hoch          Runter         Links         Rechts      Enter
                   Adr2 Men2211 : Adr2 Men2221 : Adr2 Men223 : Adr2 Null : Adr2 Men2
                   Adr2 Menue_men2231
                   Data &B00000101
                   '     1234567890123456
                   Data "   Stop aktiv"
                   Data ""
                   Data "{001}=Auf       {002}=Zu"


' Menü MEN3 - Dimmer
Men3:
          ' Hoch       Runter      Links       Rechts      Enter
          Adr2 Men2 : Adr2 Men4 : Adr2 Null : Adr2 Men31 : Adr2 Null
          Adr2 Menue_men3
          Data &B00000011
          '     1234567890123456
          Data "3            3/8"
          Data "     Dimmer"
          Data ""

Men31:
             ' Hoch        Runter      Links       Rechts        Enter
             Adr2 Men32 : Adr2 Men32 : Adr2 Men3 : Adr2 Men311 : Adr2 Men3
             Adr2 Null
             Data &B00000111
             '     1234567890123456
             Data "3-1          1/2"
             Data "Betriebszustand"
             Data "    anzeigen"

Men311:
                ' Hoch         Runter        Links        Rechts      Enter
                Adr2 Men312 : Adr2 Men312 : Adr2 Men31 : Adr2 Null : Adr2 Men3
                Adr2 Menue_men311
                Data &B00000111
```

```
                '     1234567890123456
                Data "3-1-1        1/2"
                Data "Dauerlicht:"
                Data "Flashmode:"

Men312:
                ' Hoch         Runter        Links        Rechts      Enter
                Adr2 Men311 : Adr2 Men311 : Adr2 Men31 : Adr2 Null : Adr2 Men3
                Adr2 Menue_men312
                Data &B00000111
                '     1234567890123456
                Data "3-1-2        2/2"
                Data "Dimmerstufe:"
                Data "Min:     Max:"

Men32:
             ' Hoch        Runter       Links       Rechts        Enter
             Adr2 Men31 : Adr2 Men31 : Adr2 Men3 : Adr2 Men321 : Adr2 Men3
             Adr2 Null
             Data &B00000011
             '     1234567890123456
             Data "3-2          2/2"
             Data "  Beleuchtung"
             Data ""

Men321:
                ' Hoch         Runter       Links        Rechts         Enter
                Adr2 Men324 : Adr2 Men322 : Adr2 Men32 : Adr2 Men3211 : Adr2 Men3
                Adr2 Null
                Data &B00000011
                '     1234567890123456
                Data "3-2-1        1/4"
                Data "   Dauerlicht"
                Data ""

Men3211:
                ' Hoch           Runter          Links         Rechts          Enter
                Adr2 Men32111 : Adr2 Men32111 : Adr2 Men321 : Adr2 Men32111 : Adr2 Men3
                Adr2 Menue_men3211
                Data &B00000101
                '     1234567890123456
                Data "   Dauerlicht"
                Data ""
                Data "{003}=Back  {004}{001}{002}=Fkt."

Men32111:
                   ' Hoch           Runter          Links         Rechts          Enter
                   Adr2 Men32111 : Adr2 Men32111 : Adr2 Men321 : Adr2 Men32111 : Adr2 Men3
                   Adr2 Menue_men32111
                   Data &B00000101
                   '     1234567890123456
                   Data "   Dauerlicht"
                   Data ""
                   Data "{003}=Back  {004}{001}{002}=Fkt."

Men322:
             ' Hoch         Runter        Links        Rechts         Enter
             Adr2 Men321 : Adr2 Men323 : Adr2 Men32 : Adr2 Men3221 : Adr2 Men3
             Adr2 Menue_men322
             Data &B00000011
             '     1234567890123456
             Data "3-2-2        2/4"
             Data " Dimmer heller"
             Data ""

Men3221:
                ' Hoch           Runter          Links         Rechts          Enter
                Adr2 Men32211 : Adr2 Men32311 : Adr2 Men322 : Adr2 Men32211 : Adr2 Men3241
                Adr2 Menue_men3221
                Data &B00000101
                '     1234567890123456
                Data " Dimmer heller"
                Data ""
                Data "{001}{002}{004}=Licht  {005}=Aus"

Men32211:
                   ' Hoch           Runter          Links         Rechts          Enter
                   Adr2 Men32211 : Adr2 Men32311 : Adr2 Men322 : Adr2 Men32211 : Adr2 Men3241
                   Adr2 Menue_men32211
                   Data &B00000101
```

```
                '     1234567890123456
                Data " Dimmer heller"
                Data ""
                Data "{001}{002}{004}=Licht  {005}=Aus"


Men323:
            ' Hoch         Runter        Links        Rechts         Enter
            Adr2 Men322 : Adr2 Men324 : Adr2 Men32 : Adr2 Men3231 : Adr2 Men3
            Adr2 Menue_men323
            Data &B00000011
            '     1234567890123456
            Data "3-2-3        3/4"
            Data " Dimmer dunkler"
            Data ""

Men3231:
              ' Hoch          Runter           Links         Rechts          Enter
              Adr2 Men32211 : Adr2 Men32311 : Adr2 Men323 : Adr2 Men32311 : Adr2 Men3241
              Adr2 Menue_men3231
              Data &B00000101
              '     1234567890123456
              Data " Dimmer dunkler"
              Data ""
              Data "{001}{002}{004}=Licht  {005}=Aus"

Men32311:
              ' Hoch          Runter           Links         Rechts          Enter
              Adr2 Men32211 : Adr2 Men32311 : Adr2 Men323 : Adr2 Men32311 : Adr2 Men3241
              Adr2 Menue_men32311
              Data &B00000101
              '     1234567890123456
              Data " Dimmer dunkler"
              Data ""
              Data "{001}{002}{004}=Licht  {005}=Aus"

Men324:
            ' Hoch         Runter        Links        Rechts         Enter
            Adr2 Men323 : Adr2 Men321 : Adr2 Men32 : Adr2 Men3241 : Adr2 Men3
            Adr2 Menue_men324
            Data &B00000011
            '     1234567890123456
            Data "3-2-4        4/4"
            Data "   Dimmer aus"
            Data ""

Men3241:
              ' Hoch          Runter         Links         Rechts         Enter
              Adr2 Men3221 : Adr2 Men3231 : Adr2 Men324 : Adr2 Men3241 : Adr2 Men3
              Adr2 Menue_men3241
              Data &B00000110
              '     1234567890123456
              Data ""
              Data "   Dimmer aus"
              Data "{002}=dunkel  {001}=hell"


' Menü MEN4 - Umweltdaten
Men4:
          ' Hoch       Runter      Links       Rechts       Enter
          Adr2 Men3 : Adr2 Men5 : Adr2 Null : Adr2 Men41 : Adr2 Null
          Adr2 Menue_men4
          Data &B00000011
          '     1234567890123456
          Data "4            4/8"
          Data "  Umweltdaten"
          Data ""


Men41:
            ' Hoch        Runter       Links       Rechts        Enter
            Adr2 Men46 : Adr2 Men42 : Adr2 Men4 : Adr2 Men411 : Adr2 Men4
            Adr2 Menue_men41
            Data &B00000011
            '     1234567890123456
            Data "4-1          1/6"
            Data "      Wind"
            Data ""

Men411:
```

```
                ' Hoch       Runter        Links        Rechts        Enter
               Adr2 Null : Adr2 Men421 : Adr2 Men41 : Adr2 Null : Adr2 Men4
               Adr2 Menue_men411
               Data &B00000011
               '     1234567890123456
               Data "4-1-1        1/1"
               Data "Geschwindigkeit:"
               Data ""

Men42:
            ' Hoch        Runter       Links       Rechts        Enter
           Adr2 Men41 : Adr2 Men43 : Adr2 Men4 : Adr2 Men421 : Adr2 Men4
           Adr2 Menue_men42
           Data &B00000011
           '     1234567890123456
           Data "4-2          2/6"
           Data "      Regen"
           Data ""

Men421:
               ' Hoch         Runter        Links        Rechts        Enter
               Adr2 Men411 : Adr2 Men431 : Adr2 Men42 : Adr2 Null : Adr2 Men4
               Adr2 Menue_men421
               Data &B00000011
               '     1234567890123456
               Data "4-2-1        1/1"
               Data "Regensensor:"
               Data ""

Men43:
            ' Hoch        Runter       Links       Rechts        Enter
           Adr2 Men42 : Adr2 Men44 : Adr2 Men4 : Adr2 Men431 : Adr2 Men4
           Adr2 Menue_men43
           Data &B00000011
           '     1234567890123456
           Data "4-3          3/6"
           Data "Innentemperatur"
           Data ""

Men431:
               ' Hoch         Runter        Links        Rechts        Enter
               Adr2 Men421 : Adr2 Men441 : Adr2 Men43 : Adr2 Null : Adr2 Men4
               Adr2 Menue_men431
               Data &B00000111
               '     1234567890123456
               Data "4-3-1        1/1"
               Data "Temp:"
               Data "TauP:"

Men44:
            ' Hoch       Runter        Links       Rechts        Enter
           Adr2 Men43 : Adr2 Men45 : Adr2 Men4 : Adr2 Men441 : Adr2 Men4
           Adr2 Menue_men44
           Data &B00000011
           '     1234567890123456
           Data "4-4          4/6"
           Data "Luftfeuchtigkeit"
           Data ""

Men441:
               ' Hoch         Runter        Links        Rechts        Enter
               Adr2 Men431 : Adr2 Men451 : Adr2 Men44 : Adr2 Null : Adr2 Men4
               Adr2 Menue_men441
               Data &B00000011
               '     1234567890123456
               Data "4-4-1        1/1"
               Data "Feuchtigkeit:"
               Data ""

Men45:
            ' Hoch        Runter       Links       Rechts        Enter
           Adr2 Men44 : Adr2 Men46 : Adr2 Men4 : Adr2 Men451 : Adr2 Men4
           Adr2 Menue_men45
           Data &B00000011
           '     1234567890123456
           Data "4-5          5/6"
           Data "Lichtintensit{132}t"
           Data ""

Men451:
```

```
                ' Hoch         Runter        Links        Rechts        Enter
               Adr2 Men441 : Adr2 Men461 : Adr2 Men45 : Adr2 Null : Adr2 Men4
               Adr2 Menue_men451
               Data &B00000011
               '     1234567890123456
               Data "4-5-1        1/1"
               Data "Helligkeit:"
               Data ""

Men46:
            ' Hoch         Runter       Links       Rechts        Enter
           Adr2 Men45 : Adr2 Men41 : Adr2 Men4 : Adr2 Men461 : Adr2 Men4
           Adr2 Null
           Data &B00000011
           '     1234567890123456
           Data "4-6          6/6"
           Data "  Min/Max-Werte"
           Data ""

Men461:
               ' Hoch         Runter        Links        Rechts      Enter
               Adr2 Men451 : Adr2 Men462 : Adr2 Men46 : Adr2 Null : Adr2 Men4
               Adr2 Menue_men461
               Data &B00000011
               '     1234567890123456
               Data "4-6-1        1/3"
               Data "Geschwindigkeit:"
               Data ""

Men462:
               ' Hoch         Runter        Links        Rechts      Enter
               Adr2 Men461 : Adr2 Men463 : Adr2 Men46 : Adr2 Null : Adr2 Men4
               Adr2 Menue_men462
               Data &B00000111
               '     1234567890123456
               Data "4-6-2        2/3"
               Data "MinTemp:"
               Data "MaxTemp:"

Men463:
               ' Hoch         Runter      Links        Rechts      Enter
               Adr2 Men462 : Adr2 Null : Adr2 Men46 : Adr2 Null : Adr2 Men4
               Adr2 Menue_men463
               Data &B00000111
               '     1234567890123456
               Data "4-6-3        3/3"
               Data "MinFeucht:"
               Data "MaxFeucht:"


' Menü MEN5 - Uhrzeit&Datum
Men5:
         ' Hoch       Runter      Links       Rechts       Enter
         Adr2 Men4 : Adr2 Men6 : Adr2 Null : Adr2 Men51 : Adr2 Null
         Adr2 Menue_men5
         Data &B00000011
         '     1234567890123456
         Data "5            5/8"
         Data "Uhrzeit & Datum"
         Data ""

Men51:
            ' Hoch       Runter      Links       Rechts      Enter
            Adr2 Null : Adr2 Null : Adr2 Men5 : Adr2 Null : Adr2 Men5
            Adr2 Menue_men51
            Data &B00000001
            '     1234567890123456
            Data "5-1          1/1"
            Data ""
            Data ""


' Menü MEN6 - Datalogging
Men6:
         ' Hoch       Runter      Links       Rechts       Enter
         Adr2 Men5 : Adr2 Men7 : Adr2 Null : Adr2 Men61 : Adr2 Null
         Adr2 Null
         Data &B00000011
         '     1234567890123456
         Data "6            6/8"
```

```
        Data "  Data-Logging"
        Data ""

Men61:
          ' Hoch        Runter       Links       Rechts        Enter
          Adr2 Men62 : Adr2 Men62 : Adr2 Men6 : Adr2 Men611 : Adr2 Men6
          Adr2 Null
          Data &B00000111
          '     1234567890123456
          Data "6-1          1/2"
          Data "  Loggingdaten"
          Data "    anzeigen"

Men611:
            ' Hoch         Runter        Links        Rechts         Enter
            Adr2 Men612 : Adr2 Men612 : Adr2 Men61 : Adr2 Men6111 : Adr2 Men6
            Adr2 Null
            Data &B00000011
            '     1234567890123456
            Data "6-1-1        1/2"
            Data "Internes EEPROM"
            Data ""

Men6111:
              ' Hoch       Runter      Links         Rechts      Enter
              Adr2 Null : Adr2 Null : Adr2 Men611 : Adr2 Null : Adr2 Men6
              Adr2 Menue_men6111
              Data &B00000011
              '     1234567890123456
              Data "6-1-1-1      1/1"
              Data "Datenversion"
              Data ""

Men612:
            ' Hoch         Runter        Links        Rechts         Enter
            Adr2 Men611 : Adr2 Men611 : Adr2 Men61 : Adr2 Men6121 : Adr2 Men6
            Adr2 Null
            Data &B00000011
            '     1234567890123456
            Data "6-1-2        2/2"
            Data "Externes EEPROM"
            Data ""

Men6121:
              ' Hoch          Runter         Links         Rechts      Enter
              Adr2 Men6124 : Adr2 Men6122 : Adr2 Men612 : Adr2 Null : Adr2 Men6
              Adr2 Menue_men6121
              Data &B00000011
              '     1234567890123456
              Data "6-1-2-1      1/4"
              Data "Datenversion"
              Data ""

Men6122:
              ' Hoch          Runter         Links         Rechts      Enter
              Adr2 Men6121 : Adr2 Men6123 : Adr2 Men612 : Adr2 Null : Adr2 Men6
              Adr2 Menue_men6122
              Data &B00000011
              '     1234567890123456
              Data "6-1-2-2      2/4"
              Data "Datencounter"
              Data ""

Men6123:
              ' Hoch          Runter         Links         Rechts      Enter
              Adr2 Men6122 : Adr2 Men6124 : Adr2 Men612 : Adr2 Null : Adr2 Men6
              Adr2 Menue_men6123
              Data &B00000011
              '     1234567890123456
              Data "6-1-2-3      3/4"
              Data "{154}berlaufz{132}hler"
              Data ""

Men6124:
              ' Hoch          Runter         Links         Rechts      Enter
              Adr2 Men6123 : Adr2 Men6121 : Adr2 Men612 : Adr2 Null : Adr2 Men6
              Adr2 Menue_men6124
              Data &B00000011
              '     1234567890123456
              Data "6-1-2-4      4/4"
```

```
                Data "Aktuelle Adresse"
                Data ""

Men62:
            ' Hoch        Runter       Links       Rechts        Enter
            Adr2 Men61 : Adr2 Men61 : Adr2 Men6 : Adr2 Men621 : Adr2 Men6
            Adr2 Null
            Data &B00000111
            '     1234567890123456
            Data "6-2          2/2"
            Data "  Loggingdaten"
            Data "   {129}bertragen"

Men621:
              ' Hoch        Runter     Links       Rechts      Enter
              Adr2 Null : Adr2 Null : Adr2 Null : Adr2 Null : Adr2 Null
              Adr2 Menue_men621
              Data &B00000111
              '     1234567890123456
              Data "  Loggingdaten"
              Data "werden  gesendet"
              Data " Bitte warten!!"


' Menü MEN7 - Timer
Men7:
         ' Hoch       Runter      Links       Rechts      Enter
         Adr2 Men6 : Adr2 Men8 : Adr2 Null : Adr2 Men71 : Adr2 Null
         Adr2 Null
         Data &B00000011
         '     1234567890123456
         Data "7            7/8"
         Data "     Timer"
         Data ""

Men71:
            ' Hoch        Runter       Links       Rechts       Enter
            Adr2 Men72 : Adr2 Men72 : Adr2 Men7 : Adr2 Men711 : Adr2 Men7
            Adr2 Null
            Data &B00000011
            '     1234567890123456
            Data "7-1          1/2"
            Data "   Aktivieren"
            Data ""

Men711:
              ' Hoch         Runter        Links        Rechts         Enter
              Adr2 Men712 : Adr2 Men712 : Adr2 Men71 : Adr2 Men7111 : Adr2 Men7
              Adr2 Null
              Data &B00000011
              '     1234567890123456
              Data "7-1-1        1/2"
              Data "     {153}ffnen"
              Data ""

Men7111:
                ' Hoch          Runter          Links         Rechts          Enter
                Adr2 Men71111 : Adr2 Men71111 : Adr2 Men711 : Adr2 Men71111 : Adr2 Men7
                Adr2 Menue_men7111
                Data &B00000101
                '     1234567890123456
                Data "     {153}ffnen"
                Data ""
                Data "{003}=Back  {004}{001}{002}=Fkt."

Men71111:
                  ' Hoch          Runter          Links         Rechts          Enter
                  Adr2 Men71111 : Adr2 Men71111 : Adr2 Men711 : Adr2 Men71111 : Adr2 Men7
                  Adr2 Menue_men71111
                  Data &B00000101
                  '     1234567890123456
                  Data "     {153}ffnen"
                  Data ""
                  Data "{003}=Back  {004}{001}{002}=Fkt."

Men7121:
                ' Hoch          Runter          Links         Rechts          Enter
                Adr2 Men71211 : Adr2 Men71211 : Adr2 Men712 : Adr2 Men71211 : Adr2 Men7
                Adr2 Menue_men7121
                Data &B00000101
```

```
                '     1234567890123456
                Data "   Schliessen"
                Data ""
                Data "{003}=Back  {004}{001}{002}=Fkt."

Men71211:
                   ' Hoch           Runter          Links         Rechts          Enter
                   Adr2 Men71211 : Adr2 Men71211 : Adr2 Men712 : Adr2 Men71211 : Adr2 Men7
                   Adr2 Menue_men71211
                   Data &B00000101
                   '     1234567890123456
                   Data "   Schliessen"
                   Data ""
                   Data "{003}=Back  {004}{001}{002}=Fkt."

Men712:
              ' Hoch         Runter        Links        Rechts         Enter
              Adr2 Men711 : Adr2 Men711 : Adr2 Men71 : Adr2 Men7121 : Adr2 Men7
              Adr2 Null
              Data &B00000011
              '     1234567890123456
              Data "7-1-2        2/2"
              Data "  Schliessen"
              Data ""

Men72:
            ' Hoch        Runter       Links       Rechts       Enter
            Adr2 Men71 : Adr2 Men71 : Adr2 Men7 : Adr2 Men721 : Adr2 Men7
            Adr2 Null
            Data &B00000011
            '     1234567890123456
            Data "7-2          2/2"
            Data "   Einstellen"
            Data ""

Men721:
              ' Hoch         Runter        Links        Rechts         Enter
              Adr2 Men722 : Adr2 Men722 : Adr2 Men72 : Adr2 Men7211 : Adr2 Men7
              Adr2 Null
              Data &B00000111
              '     1234567890123456
              Data "7-2-1        1/2"
              Data " Programmierung"
              Data "  {153}ffnungszeit"

Men7211:
                ' Hoch          Runter         Links         Rechts      Enter
                Adr2 Men7212 : Adr2 Men7212 : Adr2 Men721 : Adr2 Men72111 : Adr2 Men7
                Adr2 Menue_men7211
                Data &B00000111
                '     1234567890123456
                Data "7-2-1-1      1/2"
                Data "    Uhrzeit"
                Data "{003}=Back  {004}{001}{002}=Fkt."

Men72111:
                ' Hoch       Runter      Links       Rechts      Enter
                Adr2 Null : Adr2 Null : Adr2 Null : Adr2 Null : Adr2 Null
                Adr2 Menue_men72111
                Data &B00000000
                '     1234567890123456
                Data ""
                Data ""
                Data ""

Men7212:
                ' Hoch          Runter         Links         Rechts      Enter
                Adr2 Men7211 : Adr2 Men7211 : Adr2 Men721 : Adr2 Men72121 : Adr2 Men7
                Adr2 Menue_men7212
                Data &B00000111
                '     1234567890123456
                Data "7-2-1-2      2/2"
                Data "     Datum"
                Data "{003}=Back  {004}{001}{002}=Fkt."

Men72121:
                ' Hoch       Runter      Links       Rechts      Enter
                Adr2 Null : Adr2 Null : Adr2 Null : Adr2 Null : Adr2 Null
                Adr2 Menue_men72121
                Data &B00000000
```

```
                '     1234567890123456
                Data ""
                Data ""
                Data ""

Men722:
              ' Hoch         Runter        Links        Rechts         Enter
              Adr2 Men721 : Adr2 Men721 : Adr2 Men72 : Adr2 Men7221 : Adr2 Men7
              Adr2 Null
              Data &B00000111
              '     1234567890123456
              Data "7-2-2        2/2"
              Data " Programmierung"
              Data "Schliessungszeit"

Men7221:
                ' Hoch          Runter         Links         Rechts      Enter
                Adr2 Men7222 : Adr2 Men7222 : Adr2 Men722 : Adr2 Men72211 : Adr2 Men7
                Adr2 Menue_men7221
                Data &B00000111
                '     1234567890123456
                Data "7-2-2-1      1/2"
                Data "    Uhrzeit"
                Data "{003}=Back  {004}{001}{002}=Fkt."

Men72211:
                ' Hoch       Runter      Links       Rechts      Enter
                Adr2 Null : Adr2 Null : Adr2 Null : Adr2 Null : Adr2 Null
                Adr2 Menue_men72211
                Data &B00000000
                '     1234567890123456
                Data ""
                Data ""
                Data ""

Men7222:
                ' Hoch          Runter         Links         Rechts      Enter
                Adr2 Men7221 : Adr2 Men7221 : Adr2 Men722 : Adr2 Men72221 : Adr2 Men7
                Adr2 Menue_men7222
                Data &B00000111
                '     1234567890123456
                Data "7-2-2-2      2/2"
                Data "     Datum"
                Data "{003}=Back  {004}{001}{002}=Fkt."

Men72221:
                ' Hoch       Runter      Links       Rechts      Enter
                Adr2 Null : Adr2 Null : Adr2 Null : Adr2 Null : Adr2 Null
                Adr2 Menue_men72221
                Data &B00000000
                '     1234567890123456
                Data ""
                Data ""
                Data ""


' Menü MEN8 - Einstellungen
Men8:
         ' Hoch       Runter      Links       Rechts      Enter
         Adr2 Men7 : Adr2 Men1 : Adr2 Null : Adr2 Men81 : Adr2 Null
         Adr2 Null
         Data &B00000011
         '     1234567890123456
         Data "8            8/8"
         Data " Einstellungen"
         Data ""


Men81:
            ' Hoch        Runter       Links       Rechts        Enter
            Adr2 Men8b : Adr2 Men82 : Adr2 Men8 : Adr2 Men811 : Adr2 Men8
            Adr2 Menue_men81
            Data &B00000011
            '     1234567890123456
            Data "8-1         1/11"
            Data "     Dimmer"
            Data ""

Men811:
              ' Hoch         Runter        Links        Rechts         Enter
```

```
                Adr2 Men812 : Adr2 Men812 : Adr2 Men81 : Adr2 Men8111 : Adr2 Men81
                Adr2 Null
                Data &B00000011
                '     1234567890123456
                Data "8-1-1        1/2"
                Data " Light-Floating"
                Data ""

Men8111:
                   ' Hoch           Runter          Links         Rechts          Enter
                   Adr2 Men81111 : Adr2 Men81111 : Adr2 Men811 : Adr2 Men81111 : Adr2 Men81
                   Adr2 Menue_men8111
                   Data &B00000101
                   '     1234567890123456
                   Data " Light-Floating"
                   Data ""
                   Data "{003}=Back  {004}{001}{002}=Fkt."

Men81111:
                   ' Hoch           Runter          Links         Rechts          Enter
                   Adr2 Men81111 : Adr2 Men81111 : Adr2 Men811 : Adr2 Men81111 : Adr2 Men81
                   Adr2 Menue_men81111
                   Data &B00000101
                   '     1234567890123456
                   Data " Light-Floating"
                   Data ""
                   Data "{003}=Back  {004}{001}{002}=Fkt."

Men812:
                ' Hoch         Runter        Links        Rechts         Enter
                Adr2 Men811 : Adr2 Men811 : Adr2 Men81 : Adr2 Men8121 : Adr2 Men81
                Adr2 Null
                Data &B00000011
                '     1234567890123456
                Data "8-1-2        2/2"
                Data "   Grenzwerte"
                Data ""

Men8121:
                   ' Hoch          Runter         Links         Rechts          Enter
                   Adr2 Men8122 : Adr2 Men8122 : Adr2 Men812 : Adr2 Men81211 : Adr2 Men81
                   Adr2 Null
                   Data &B00000111
                   '     1234567890123456
                   Data "8-1-2-1      1/2"
                   Data "Min- / Maxwerte"
                   Data " Untere Grenze"

Men81211:
                   ' Hoch           Runter          Links          Rechts           Enter
                   Adr2 Men812111 : Adr2 Men812111 : Adr2 Men8121 : Adr2 Men812111 : Adr2 Men81
                   Adr2 Menue_men81211
                   Data &B00000101
                   '     1234567890123456
                   Data " Untere Grenze"
                   Data ""
                   Data "{003}=Back  {004}{001}{002}=Fkt."

Men812111:
                      ' Hoch           Runter          Links          Rechts           Enter
                      Adr2 Men812111 : Adr2 Men812111 : Adr2 Men8121 : Adr2 Men812111 : Adr2 Men81
                      Adr2 Menue_men812111
                      Data &B00000101
                      '     1234567890123456
                      Data " Untere Grenze"
                      Data ""
                      Data "{003}=Back  {004}{001}{002}=Fkt."

Men81221:
                   ' Hoch           Runter          Links          Rechts           Enter
                   Adr2 Men812211 : Adr2 Men812211 : Adr2 Men8122 : Adr2 Men812211 : Adr2 Men81
                   Adr2 Menue_men81221
                   Data &B00000101
                   '     1234567890123456
                   Data "  Obere Grenze"
                   Data ""
                   Data "{003}=Back  {004}{001}{002}=Fkt."

Men812211:
                      ' Hoch           Runter          Links          Rechts           Enter
```

```
                    Adr2 Men812211 : Adr2 Men812211 : Adr2 Men8122 : Adr2 Men812211 : Adr2 Men81
                    Adr2 Menue_men812211
                    Data &B00000101
                    '     1234567890123456
                    Data "  Obere Grenze"
                    Data ""
                    Data "{003}=Back  {004}{001}{002}=Fkt."


Men8122:
                ' Hoch          Runter         Links         Rechts          Enter
                Adr2 Men8121 : Adr2 Men8121 : Adr2 Men812 : Adr2 Men81221 : Adr2 Men81
                Adr2 Null
                Data &B00000111
                '     1234567890123456
                Data "8-1-2-2      2/2"
                Data "Min- / Maxwerte"
                Data "  Obere Grenze"

Men82:
            ' Hoch        Runter       Links       Rechts        Enter
            Adr2 Men81 : Adr2 Men83 : Adr2 Men8 : Adr2 Men821 : Adr2 Men8
            Adr2 Menue_men82
            Data &B00000011
            '     1234567890123456
            Data "8-2          2/11"
            Data "    Tastatur"
            Data ""

Men821:
              ' Hoch       Runter      Links        Rechts         Enter
              Adr2 Null : Adr2 Null : Adr2 Men82 : Adr2 Men8211 : Adr2 Men82
              Adr2 Null
              Data &B00000111
              '     1234567890123456
              Data "8-2-1        1/1"
              Data "   Enter-Taste"
              Data "    ShortCut"

Men8211:
                ' Hoch           Runter          Links         Rechts          Enter
                Adr2 Men82111 : Adr2 Men82111 : Adr2 Men821 : Adr2 Men82111 : Adr2 Men82
                Adr2 Menue_men8211
                Data &B00000101
                '     1234567890123456
                Data " Enter ShortCut"
                Data ""
                Data "{003}=Back  {004}{001}{002}=Fkt."

Men82111:
            ' Hoch           Runter          Links         Rechts          Enter
            Adr2 Men82111 : Adr2 Men82111 : Adr2 Men821 : Adr2 Men82111 : Adr2 Men82
            Adr2 Menue_men82111
            Data &B00000101
            '     1234567890123456
            Data " Enter ShortCut"
            Data ""
            Data "{003}=Back  {004}{001}{002}=Fkt."

Men83:
            ' Hoch        Runter       Links       Rechts        Enter
            Adr2 Men82 : Adr2 Men84 : Adr2 Men8 : Adr2 Men831 : Adr2 Men8
            Adr2 Menue_men83
            Data &B00000011
            '     1234567890123456
            Data "8-3          3/11"
            Data "    Display"
            Data ""

Men831:
               ' Hoch         Runter        Links        Rechts         Enter
               Adr2 Men833 : Adr2 Men832 : Adr2 Men83 : Adr2 Men8311 : Adr2 Men83
               Adr2 Null
               Data &B00000011
               '     1234567890123456
               Data "8-3-1        1/3"
               Data "  Screen-Saver"
               Data ""

Men8311:
```

```
                ' Hoch          Runter          Links         Rechts          Enter
                Adr2 Men83111 : Adr2 Men83111 : Adr2 Men831 : Adr2 Men83111 : Adr2 Men83
                Adr2 Menue_men8311
                Data &B00000101
                '     1234567890123456
                Data "  Screen-Saver"
                Data ""
                Data "{003}=Back  {004}{001}{002}=Fkt."

Men83111:
                ' Hoch          Runter          Links         Rechts          Enter
                Adr2 Men83111 : Adr2 Men83111 : Adr2 Men831 : Adr2 Men83111 : Adr2 Men83
                Adr2 Menue_men83111
                Data &B00000101
                '     1234567890123456
                Data "  Screen-Saver"
                Data ""
                Data "{003}=Back  {004}{001}{002}=Fkt."

Men832:
              ' Hoch         Runter        Links        Rechts         Enter
              Adr2 Men831 : Adr2 Men833 : Adr2 Men83 : Adr2 Men8321 : Adr2 Men83
              Adr2 Null
              Data &B00000011
              '     1234567890123456
              Data "8-3-2        2/3"
              Data "    Kontrast"
              Data ""

Men8321:
                ' Hoch          Runter          Links         Rechts          Enter
                Adr2 Men83211 : Adr2 Men83211 : Adr2 Men832 : Adr2 Men83211 : Adr2 Men83
                Adr2 Menue_men8321
                Data &B00000101
                '     1234567890123456
                Data "    Kontrast"
                Data ""
                Data "{003}=Back  {004}{001}{002}=Fkt."

Men83211:
                ' Hoch          Runter          Links         Rechts          Enter
                Adr2 Men83211 : Adr2 Men83211 : Adr2 Men832 : Adr2 Men83211 : Adr2 Men83
                Adr2 Menue_men83211
                Data &B00000101
                '     1234567890123456
                Data "    Kontrast"
                Data ""
                Data "{003}=Back  {004}{001}{002}=Fkt."

Men833:
              ' Hoch         Runter        Links        Rechts         Enter
              Adr2 Men832 : Adr2 Men831 : Adr2 Men83 : Adr2 Men8331 : Adr2 Men83
              Adr2 Null
              Data &B00000011
              '     1234567890123456
              Data "8-3-3        3/3"
              Data "   Helligkeit"
              Data ""

Men8331:
                ' Hoch          Runter          Links         Rechts          Enter
                Adr2 Men83311 : Adr2 Men83311 : Adr2 Men833 : Adr2 Men83311 : Adr2 Men83
                Adr2 Menue_men8331
                Data &B00000101
                '     1234567890123456
                Data "   Helligkeit"
                Data ""
                Data "{003}=Back  {004}{001}{002}=Fkt."

Men83311:
                ' Hoch          Runter          Links         Rechts          Enter
                Adr2 Men83311 : Adr2 Men83311 : Adr2 Men833 : Adr2 Men83311 : Adr2 Men83
                Adr2 Menue_men83311
                Data &B00000101
                '     1234567890123456
                Data "   Helligkeit"
                Data ""
                Data "{003}=Back  {004}{001}{002}=Fkt."

Men84:
```

```
            ' Hoch        Runter       Links       Rechts        Enter
           Adr2 Men83 : Adr2 Men85 : Adr2 Men8 : Adr2 Men841 : Adr2 Men8
           Adr2 Menue_men84
           Data &B00000011
           '     1234567890123456
           Data "8-4         4/11"
           Data "     L{129}fter"
           Data ""

Men841:
              ' Hoch         Runter        Links        Rechts         Enter
             Adr2 Men845 : Adr2 Men842 : Adr2 Men84 : Adr2 Men8411 : Adr2 Men84
             Adr2 Null
             Data &B00000111
             '     1234567890123456
             Data "8-4-1        1/5"
             Data "Betriebsmodus SW"
             Data " Aktiv / Passiv"

Men8411:
                ' Hoch           Runter          Links         Rechts          Enter
               Adr2 Men84111 : Adr2 Men84111 : Adr2 Men841 : Adr2 Men84111 : Adr2 Men84
               Adr2 Menue_men8411
               Data &B00000101
               '     1234567890123456
               Data "Betriebsmodus SW"
               Data ""
               Data "{003}=Back  {004}{001}{002}=Fkt."

Men84111:
                  ' Hoch           Runter          Links         Rechts          Enter
                 Adr2 Men84111 : Adr2 Men84111 : Adr2 Men841 : Adr2 Men84111 : Adr2 Men84
                 Adr2 Menue_men84111
                 Data &B00000101
                 '     1234567890123456
                 Data "Betriebsmodus SW"
                 Data ""
                 Data "{003}=Back  {004}{001}{002}=Fkt."

Men842:
              ' Hoch         Runter        Links        Rechts         Enter
             Adr2 Men841 : Adr2 Men843 : Adr2 Men84 : Adr2 Men8421 : Adr2 Men84
             Adr2 Null
             Data &B00000111
             '     1234567890123456
             Data "8-4-2        2/5"
             Data "   L{129}fter 25%"
             Data "   Temperatur"

Men8421:
                ' Hoch           Runter          Links         Rechts          Enter
               Adr2 Men84211 : Adr2 Men84211 : Adr2 Men842 : Adr2 Men84211 : Adr2 Men84
               Adr2 Menue_men8421
               Data &B00000101
               '     1234567890123456
               Data "     25% bei"
               Data ""
               Data "{003}=Back  {004}{001}{002}=Fkt."

Men84211:
                  ' Hoch           Runter          Links         Rechts          Enter
                 Adr2 Men84211 : Adr2 Men84211 : Adr2 Men842 : Adr2 Men84211 : Adr2 Men84
                 Adr2 Menue_men84211
                 Data &B00000101
                 '     1234567890123456
                 Data "     25% bei"
                 Data ""
                 Data "{003}=Back  {004}{001}{002}=Fkt."

Men843:
              ' Hoch         Runter        Links        Rechts         Enter
             Adr2 Men842 : Adr2 Men844 : Adr2 Men84 : Adr2 Men8431 : Adr2 Men84
             Adr2 Null
             Data &B00000111
             '     1234567890123456
             Data "8-4-3        3/5"
             Data "   L{129}fter 50%"
             Data "   Temperatur"

Men8431:
```

```
                ' Hoch          Runter          Links         Rechts          Enter
               Adr2 Men84311 : Adr2 Men84311 : Adr2 Men843 : Adr2 Men84311 : Adr2 Men84
               Adr2 Menue_men8431
               Data &B00000101
               '     1234567890123456
               Data "     50% bei"
               Data ""
               Data "{003}=Back  {004}{001}{002}=Fkt."

Men84311:
                   ' Hoch          Runter          Links         Rechts          Enter
                  Adr2 Men84311 : Adr2 Men84311 : Adr2 Men843 : Adr2 Men84311 : Adr2 Men84
                  Adr2 Menue_men84311
                  Data &B00000101
                  '     1234567890123456
                  Data "     50% bei"
                  Data ""
                  Data "{003}=Back  {004}{001}{002}=Fkt."

Men844:
              ' Hoch         Runter        Links        Rechts         Enter
              Adr2 Men843 : Adr2 Men845 : Adr2 Men84 : Adr2 Men8441 : Adr2 Men84
              Adr2 Null
              Data &B00000111
              '     1234567890123456
              Data "8-4-4        4/5"
              Data "   L{129}fter 75%"
              Data "   Temperatur"

Men8441:
               ' Hoch          Runter          Links         Rechts          Enter
               Adr2 Men84411 : Adr2 Men84411 : Adr2 Men844 : Adr2 Men84411 : Adr2 Men84
               Adr2 Menue_men8441
               Data &B00000101
               '     1234567890123456
               Data "     75% bei"
               Data ""
               Data "{003}=Back  {004}{001}{002}=Fkt."

Men84411:
                   ' Hoch          Runter          Links         Rechts          Enter
                  Adr2 Men84411 : Adr2 Men84411 : Adr2 Men844 : Adr2 Men84411 : Adr2 Men84
                  Adr2 Menue_men84411
                  Data &B00000101
                  '     1234567890123456
                  Data "     75% bei"
                  Data ""
                  Data "{003}=Back  {004}{001}{002}=Fkt."

Men845:
              ' Hoch         Runter        Links        Rechts         Enter
              Adr2 Men844 : Adr2 Men841 : Adr2 Men84 : Adr2 Men8451 : Adr2 Men84
              Adr2 Null
              Data &B00000111
              '     1234567890123456
              Data "8-4-5        5/5"
              Data "   L{129}fter 100%"
              Data "   Temperatur"

Men8451:
               ' Hoch          Runter          Links         Rechts          Enter
               Adr2 Men84511 : Adr2 Men84511 : Adr2 Men845 : Adr2 Men84511 : Adr2 Men84
               Adr2 Menue_men8451
               Data &B00000101
               '     1234567890123456
               Data "    100% bei"
               Data ""
               Data "{003}=Back  {004}{001}{002}=Fkt."

Men84511:
                   ' Hoch          Runter          Links         Rechts          Enter
                  Adr2 Men84511 : Adr2 Men84511 : Adr2 Men845 : Adr2 Men84511 : Adr2 Men84
                  Adr2 Menue_men84511
                  Data &B00000101
                  '     1234567890123456
                  Data "    100% bei"
                  Data ""
                  Data "{003}=Back  {004}{001}{002}=Fkt."

Men85:
```

```
            ' Hoch        Runter       Links       Rechts        Enter
           Adr2 Men84 : Adr2 Men86 : Adr2 Men8 : Adr2 Men851 : Adr2 Men8
           Adr2 Menue_men85
           Data &B00000011
           '     1234567890123456
           Data "8-5         5/11"
           Data "      Wind"
           Data ""

Men851:
              ' Hoch         Runter        Links        Rechts         Enter
             Adr2 Men852 : Adr2 Men852 : Adr2 Men85 : Adr2 Men8511 : Adr2 Men85
             Adr2 Null
             Data &B00000111
             '     1234567890123456
             Data "8-5-1        1/2"
             Data "   Windgrenze"
             Data "Beschattung auf"

Men8511:
                ' Hoch           Runter          Links         Rechts          Enter
               Adr2 Men85111 : Adr2 Men85111 : Adr2 Men851 : Adr2 Men85111 : Adr2 Men85
               Adr2 Menue_men8511
               Data &B00000101
               '     1234567890123456
               Data "Geschwindigkeit"
               Data ""
               Data "{003}=Back  {004}{001}{002}=Fkt."

Men85111:
                  ' Hoch           Runter          Links         Rechts          Enter
                 Adr2 Men85111 : Adr2 Men85111 : Adr2 Men851 : Adr2 Men85111 : Adr2 Men85
                 Adr2 Menue_men85111
                 Data &B00000101
                 '     1234567890123456
                 Data "Geschwindigkeit"
                 Data ""
                 Data "{003}=Back  {004}{001}{002}=Fkt."

Men852:
              ' Hoch         Runter        Links        Rechts        Enter
             Adr2 Men851 : Adr2 Men851 : Adr2 Men85 : Adr2 Men8521 : Adr2 Men85
             Adr2 Null
             Data &B00000011
             '     1234567890123456
             Data "8-5-2        2/2"
             Data "  Messintervall"
             Data ""

Men8521:
                ' Hoch           Runter          Links         Rechts          Enter
               Adr2 Men85211 : Adr2 Men85211 : Adr2 Men852 : Adr2 Men85211 : Adr2 Men85
               Adr2 Menue_men8521
               Data &B00000101
               '     1234567890123456
               Data " Messintervall"
               Data ""
               Data "{003}=Back  {004}{001}{002}=Fkt."

Men85211:
                  ' Hoch           Runter          Links         Rechts          Enter
                 Adr2 Men85211 : Adr2 Men85211 : Adr2 Men852 : Adr2 Men85211 : Adr2 Men85
                 Adr2 Menue_men85211
                 Data &B00000101
                 '     1234567890123456
                 Data " Messintervall"
                 Data ""
                 Data "{003}=Back  {004}{001}{002}=Fkt."

Men86:
           ' Hoch        Runter       Links       Rechts        Enter
          Adr2 Men85 : Adr2 Men87 : Adr2 Men8 : Adr2 Men861 : Adr2 Men8
          Adr2 Menue_men86
          Data &B00000011
          '     1234567890123456
          Data "8-6         6/11"
          Data "   Temperatur"
          Data ""

Men861:
```

```
                ' Hoch       Runter      Links        Rechts         Enter
               Adr2 Null : Adr2 Null : Adr2 Men86 : Adr2 Men8611 : Adr2 Men86
               Adr2 Null
               Data &B00000011
               '     1234567890123456
               Data "8-6-1        1/1"
               Data "  Messintervall"
               Data ""

Men8611:
                  ' Hoch           Runter          Links         Rechts          Enter
                  Adr2 Men86111 : Adr2 Men86111 : Adr2 Men861 : Adr2 Men86111 : Adr2 Men86
                  Adr2 Menue_men8611
                  Data &B00000101
                  '     1234567890123456
                  Data " Messintervall"
                  Data ""
                  Data "{003}=Back  {004}{001}{002}=Fkt."

Men86111:
                     ' Hoch           Runter          Links         Rechts          Enter
                     Adr2 Men86111 : Adr2 Men86111 : Adr2 Men861 : Adr2 Men86111 : Adr2 Men86
                     Adr2 Menue_men86111
                     Data &B00000101
                     '     1234567890123456
                     Data " Messintervall"
                     Data ""
                     Data "{003}=Back  {004}{001}{002}=Fkt."

Men87:
            ' Hoch        Runter       Links       Rechts        Enter
            Adr2 Men86 : Adr2 Men88 : Adr2 Men8 : Adr2 Men871 : Adr2 Men8
            Adr2 Menue_men87
            Data &B00000011
            '     1234567890123456
            Data "8-7         7/11"
            Data "   Signalt{148}ne"
            Data ""

Men871:
               ' Hoch       Runter      Links        Rechts         Enter
               Adr2 Null : Adr2 Null : Adr2 Men87 : Adr2 Men8711 : Adr2 Men87
               Adr2 Null
               Data &B00000011
               '     1234567890123456
               Data "8-1-1        1/1"
               Data "   Signalton"
               Data ""

Men8711:
                  ' Hoch           Runter          Links         Rechts          Enter
                  Adr2 Men87111 : Adr2 Men87111 : Adr2 Men871 : Adr2 Men87111 : Adr2 Men87
                  Adr2 Menue_men8711
                  Data &B00000101
                  '     1234567890123456
                  Data "    Signalton"
                  Data ""
                  Data "{003}=Back  {004}{001}{002}=Fkt."

Men87111:
                     ' Hoch           Runter          Links         Rechts          Enter
                     Adr2 Men87111 : Adr2 Men87111 : Adr2 Men871 : Adr2 Men87111 : Adr2 Men87
                     Adr2 Menue_men87111
                     Data &B00000101
                     '     1234567890123456
                     Data "    Signalton"
                     Data ""
                     Data "{003}=Back  {004}{001}{002}=Fkt."

Men88:
            ' Hoch        Runter       Links       Rechts        Enter
            Adr2 Men87 : Adr2 Men89 : Adr2 Men8 : Adr2 Men881 : Adr2 Men8
            Adr2 Menue_men88
            Data &B00000011
            '     1234567890123456
            Data "8-8         8/11"
            Data "Uhrzeit & Datum"
            Data ""

Men881:
```

```
                ' Hoch          Runter         Links         Rechts          Enter
               Adr2 Men883 : Adr2 Men882 : Adr2 Men88 : Adr2 Men8811 : Adr2 Men88
               Adr2 Menue_men881
               Data &B00000111
               '     1234567890123456
               Data "8-8-1         1/3"
               Data "    Uhrzeit"
               Data "   einstellen"

Men8811:
                  ' Hoch       Runter      Links       Rechts      Enter
                  Adr2 Null : Adr2 Null : Adr2 Null : Adr2 Null : Adr2 Null
                  Adr2 Menue_men8811
                  Data &B00000000
                  '     1234567890123456
                  Data ""
                  Data ""
                  Data ""


Men882:
               ' Hoch          Runter         Links         Rechts          Enter
               Adr2 Men881 : Adr2 Men883 : Adr2 Men88 : Adr2 Men8821 : Adr2 Men88
               Adr2 Menue_men882
               Data &B00000111
               '     1234567890123456
               Data "8-8-2         2/3"
               Data "     Datum"
               Data "   einstellen"

Men8821:
                  ' Hoch       Runter      Links       Rechts      Enter
                  Adr2 Null : Adr2 Null : Adr2 Null : Adr2 Null : Adr2 Null
                  Adr2 Menue_men8821
                  Data &B00000000
                  '     1234567890123456
                  Data ""
                  Data ""
                  Data ""

Men883:
               ' Hoch          Runter         Links         Rechts          Enter
               Adr2 Men882 : Adr2 Men881 : Adr2 Men88 : Adr2 Men8831 : Adr2 Men88
               Adr2 Menue_men883
               Data &B00000011
               '     1234567890123456
               Data "8-8-3         3/3"
               Data "  Status DCF77"
               Data ""

Men8831:
               ' Hoch       Runter      Links         Rechts      Enter
               Adr2 Null : Adr2 Null : Adr2 Men883 : Adr2 Null : Adr2 Men88
               Adr2 Menue_men8831
               Data &B00000000
               '     1234567890123456
               Data ""
               Data ""
               Data ""


Men89:
            ' Hoch         Runter        Links        Rechts         Enter
            Adr2 Men88 : Adr2 Men8a : Adr2 Men8 : Adr2 Men891 : Adr2 Men8
            Adr2 Menue_men89
            Data &B00000011
            '     1234567890123456
            Data "8-9          9/11"
            Data "  Data-Logging"
            Data ""

Men891:
               ' Hoch          Runter         Links         Rechts          Enter
               Adr2 Men893 : Adr2 Men892 : Adr2 Men89 : Adr2 Men8911 : Adr2 Men89
               Adr2 Null
               Data &B00000111
               '     1234567890123456
               Data "8-9-1         1/3"
               Data " Betriebsmodus"
               Data "   Ein / Aus"
```

```
Men8911:
                ' Hoch          Runter          Links         Rechts          Enter
                Adr2 Men89111 : Adr2 Men89111 : Adr2 Men891 : Adr2 Men89111 : Adr2 Men89
                Adr2 Menue_men8911
                Data &B00000101
                '     1234567890123456
                Data " Betriebsmodus"
                Data ""
                Data "{003}=Back  {004}{001}{002}=Fkt."

Men89111:
                ' Hoch          Runter          Links         Rechts          Enter
                Adr2 Men89111 : Adr2 Men89111 : Adr2 Men891 : Adr2 Men89111 : Adr2 Men89
                Adr2 Menue_men89111
                Data &B00000101
                '     1234567890123456
                Data " Betriebsmodus"
                Data ""
                Data "{003}=Back  {004}{001}{002}=Fkt."

Men892:
              ' Hoch         Runter        Links        Rechts         Enter
              Adr2 Men891 : Adr2 Men893 : Adr2 Men89 : Adr2 Men8921 : Adr2 Men89
              Adr2 Null
              Data &B00000011
              '     1234567890123456
              Data "8-9-2        2/3"
              Data " Zeitintervall"
              Data ""

Men8921:
                ' Hoch          Runter          Links         Rechts          Enter
                Adr2 Men89211 : Adr2 Men89211 : Adr2 Men892 : Adr2 Men89211 : Adr2 Men89
                Adr2 Menue_men8921
                Data &B00000101
                '     1234567890123456
                Data "  Zeitintervall"
                Data ""
                Data "{003}=Back  {004}{001}{002}=Fkt."

Men89211:
                  ' Hoch          Runter          Links         Rechts          Enter
                  Adr2 Men89211 : Adr2 Men89211 : Adr2 Men892 : Adr2 Men89211 : Adr2 Men89
                  Adr2 Menue_men89211
                  Data &B00000101
                  '     1234567890123456
                  Data "  Zeitintervall"
                  Data ""
                  Data "{003}=Back  {004}{001}{002}=Fkt."

Men893:
              ' Hoch         Runter        Links        Rechts         Enter
              Adr2 Men892 : Adr2 Men891 : Adr2 Men89 : Adr2 Men8931 : Adr2 Men89
              Adr2 Null
              Data &B00000111
              '     1234567890123456
              Data "8-9-3        3/3"
              Data "  Loggingdaten"
              Data "   l{148}schen"

Men8931:
                ' Hoch       Runter      Links       Rechts      Enter
                Adr2 Null : Adr2 Null : Adr2 Null : Adr2 Null : Adr2 Null
                Adr2 Menue_men8931
                Data &B00000111
                '     1234567890123456
                Data "  Loggingdaten"
                Data "werden  gel{148}scht"
                Data " Bitte warten !"

Men8a:
            ' Hoch        Runter       Links       Rechts        Enter
            Adr2 Men89 : Adr2 Men8b : Adr2 Men8 : Adr2 Men8a1 : Adr2 Men8
            Adr2 Menue_men8a
            Data &B00000011
            '     1234567890123456
            Data "8-10       10/11"
            Data "  Beschattung"
            Data ""
```

```
Men8a1:
               ' Hoch         Runter        Links        Rechts         Enter
              Adr2 Men8a3 : Adr2 Men8a2 : Adr2 Men8a : Adr2 Men8a11 : Adr2 Men8a
              Adr2 Null
              Data &B00000011
              '     1234567890123456
              Data "8-10-1       1/3"
              Data "    Timeout"
              Data ""

Men8a11:
                 ' Hoch           Runter          Links         Rechts          Enter
                Adr2 Men8a111 : Adr2 Men8a111 : Adr2 Men8a1 : Adr2 Men8a111 : Adr2 Men8a
                Adr2 Menue_men8a11
                Data &B00000101
                '     1234567890123456
                Data "    Timeout"
                Data ""
                Data "{003}=Back  {004}{001}{002}=Fkt."

Men8a111:
                   ' Hoch           Runter          Links         Rechts          Enter
                  Adr2 Men8a111 : Adr2 Men8a111 : Adr2 Men8a1 : Adr2 Men8a111 : Adr2 Men8a
                  Adr2 Menue_men8a111
                  Data &B00000101
                  '     1234567890123456
                  Data "    Timeout"
                  Data ""
                  Data "{003}=Back  {004}{001}{002}=Fkt."

Men8a2:
               ' Hoch         Runter        Links        Rechts         Enter
              Adr2 Men8a1 : Adr2 Men8a3 : Adr2 Men8a : Adr2 Men8a21 : Adr2 Men8a
              Adr2 Null
              Data &B00000011
              '     1234567890123456
              Data "8-10-2       2/3"
              Data "    Auffahren"
              Data ""

Men8a21:
                 ' Hoch          Runter         Links         Rechts          Enter
                Adr2 Men8a22 : Adr2 Men8a22 : Adr2 Men8a2 : Adr2 Men8a211 : Adr2 Men8a
                Adr2 Null
                Data &B00000111
                '     1234567890123456
                Data "8-10-2-1     1/2"
                Data " Auffahren bei"
                Data "  Dunkelheit"

Men8a211:
                   ' Hoch           Runter          Links          Rechts          Enter
                  Adr2 Men8a2111 : Adr2 Men8a2111 : Adr2 Men8a21 : Adr2 Men8a2111 : Adr2 Men8a
                  Adr2 Menue_men8a211
                  Data &B00000101
                  '     1234567890123456
                  Data "   Dunkelheit"
                  Data ""
                  Data "{003}=Back  {004}{001}{002}=Fkt."

Men8a2111:
                     ' Hoch           Runter          Links          Rechts          Enter
                    Adr2 Men8a2111 : Adr2 Men8a2111 : Adr2 Men8a21 : Adr2 Men8a2111 : Adr2 Men8a
                    Adr2 Menue_men8a2111
                    Data &B00000101
                    '     1234567890123456
                    Data "   Dunkelheit"
                    Data ""
                    Data "{003}=Back  {004}{001}{002}=Fkt."

Men8a22:
                 ' Hoch          Runter         Links         Rechts          Enter
                Adr2 Men8a21 : Adr2 Men8a21 : Adr2 Men8a2 : Adr2 Men8a221 : Adr2 Men8a
                Adr2 Null
                Data &B00000011
                '     1234567890123456
                Data "8-10-2-2     2/2"
                Data "Grenzdunkelheit"
                Data ""
```

```
Men8a221:
                    ' Hoch            Runter           Links          Rechts           Enter
                   Adr2 Men8a2211 : Adr2 Men8a2211 : Adr2 Men8a22 : Adr2 Men8a2211 : Adr2 Men8a
                   Adr2 Menue_men8a221
                   Data &B00000101
                   '     1234567890123456
                   Data "Grenzdunkelheit"
                   Data ""
                   Data "{003}=Back  {004}{001}{002}=Fkt."

Men8a2211:
                      ' Hoch            Runter           Links          Rechts           Enter
                     Adr2 Men8a2211 : Adr2 Men8a2211 : Adr2 Men8a22 : Adr2 Men8a2211 : Adr2 Men8a
                     Adr2 Menue_men8a2211
                     Data &B00000101
                     '     1234567890123456
                     Data "Grenzdunkelheit"
                     Data ""
                     Data "{003}=Back  {004}{001}{002}=Fkt."

Men8a3:
               ' Hoch         Runter        Links        Rechts         Enter
              Adr2 Men8a2 : Adr2 Men8a1 : Adr2 Men8a : Adr2 Men8a31 : Adr2 Men8a
              Adr2 Null
              Data &B00000011
              '     1234567890123456
              Data "8-10-3       3/3"
              Data "    Zufahren"
              Data ""

Men8a31:
                  ' Hoch          Runter         Links         Rechts          Enter
                 Adr2 Men8a32 : Adr2 Men8a32 : Adr2 Men8a3 : Adr2 Men8a311 : Adr2 Men8a
                 Adr2 Null
                 Data &B00000111
                 '     1234567890123456
                 Data "8-10-3-1     1/2"
                 Data "  Zufahren bei"
                 Data "   Temperatur"

Men8a311:
                    ' Hoch            Runter           Links          Rechts           Enter
                   Adr2 Men8a3111 : Adr2 Men8a3111 : Adr2 Men8a31 : Adr2 Men8a3111 : Adr2 Men8a
                   Adr2 Menue_men8a311
                   Data &B00000101
                   '     1234567890123456
                   Data "   Temperatur"
                   Data ""
                   Data "{003}=Back  {004}{001}{002}=Fkt."

Men8a3111:
                      ' Hoch            Runter           Links          Rechts           Enter
                     Adr2 Men8a3111 : Adr2 Men8a3111 : Adr2 Men8a31 : Adr2 Men8a3111 : Adr2 Men8a
                     Adr2 Menue_men8a3111
                     Data &B00000101
                     '     1234567890123456
                     Data "   Temperatur"
                     Data ""
                     Data "{003}=Back  {004}{001}{002}=Fkt."

Men8a32:
                  ' Hoch          Runter         Links         Rechts          Enter
                 Adr2 Men8a31 : Adr2 Men8a31 : Adr2 Men8a3 : Adr2 Men8a321 : Adr2 Men8a
                 Adr2 Null
                 Data &B00000011
                 '     1234567890123456
                 Data "8-10-3-2     2/2"
                 Data "Grenztemperatur"
                 Data ""

Men8a321:
                    ' Hoch            Runter           Links          Rechts           Enter
                   Adr2 Men8a3211 : Adr2 Men8a3211 : Adr2 Men8a32 : Adr2 Men8a3211 : Adr2 Men8a
                   Adr2 Menue_men8a321
                   Data &B00000101
                   '     1234567890123456
                   Data "Grenztemperatur"
                   Data ""
                   Data "{003}=Back  {004}{001}{002}=Fkt."
```

```
Men8a3211:
                        ' Hoch           Runter          Links          Rechts          Enter
                       Adr2 Men8a3211 : Adr2 Men8a3211 : Adr2 Men8a32 : Adr2 Men8a3211 : Adr2 Men8a
                       Adr2 Menue_men8a3211
                       Data &B00000101
                       '     1234567890123456
                       Data "   Temperatur"
                       Data ""
                       Data "{003}=Back  {004}{001}{002}=Fkt."

Men8b:
            ' Hoch        Runter       Links       Rechts        Enter
           Adr2 Men8a : Adr2 Men81 : Adr2 Men8 : Adr2 Men8b1 : Adr2 Men8
           Adr2 Null
           Data &B00000011
           '     1234567890123456
           Data "8-11       11/11"
           Data "  Min/Max-Werte"
           Data ""

Men8b1:
               ' Hoch       Runter      Links        Rechts         Enter
              Adr2 Null : Adr2 Null : Adr2 Men8b : Adr2 Men8b11 : Adr2 Men8b
              Adr2 Null
              Data &B00000111
              '     1234567890123456
              Data "8-11-1       1/1"
              Data "  Min/Max-Werte"
              Data "   l{148}schen"

Men8b11:
                  ' Hoch       Runter      Links       Rechts      Enter
                 Adr2 Null : Adr2 Null : Adr2 Null : Adr2 Null : Adr2 Null
                 Adr2 Menue_men8b11
                 Data &B00000111
                 '     1234567890123456
                 Data "  Min/Max-Werte"
                 Data "werden  gel{148}scht"
                 Data " Bitte warten !"


'-- Bedatung von wiederverwendbaren Spezialmenüs ------------------------------


' Statemachine für universelle Zeiteingabe
Time1:
          ' Hoch         Runter        Links             Rechts       Enter
         Adr2 Time11 : Adr2 Time12 : Adr2 Time_ende : Adr2 Time2 : Adr2 Time_ende
         Adr2 Menue_time1
         Data &B00000111
         '     1234567890123456
         Data "     Stunden"
         Data "Eingabe HH"
         Data "Zeit:"

Time11:
          ' Hoch         Runter        Links             Rechts       Enter
         Adr2 Time11 : Adr2 Time12 : Adr2 Time_ende : Adr2 Time2 : Adr2 Time_ende
         Adr2 Menue_time11
         Data &B00000111
         '     1234567890123456
         Data "     Stunden"
         Data "Eingabe HH"
         Data "Zeit:"

Time12:
          ' Hoch         Runter        Links             Rechts       Enter
         Adr2 Time11 : Adr2 Time12 : Adr2 Time_ende : Adr2 Time2 : Adr2 Time_ende
         Adr2 Menue_time12
         Data &B00000111
         '     1234567890123456
         Data "     Stunden"
         Data "Eingabe HH"
         Data "Zeit:"

Time2:
          ' Hoch         Runter        Links        Rechts       Enter
         Adr2 Time21 : Adr2 Time22 : Adr2 Time1 : Adr2 Time3 : Adr2 Time_ende
         Adr2 Menue_time2
```

```
        Data &B00000111
        '     1234567890123456
        Data "     Minuten"
        Data "Eingabe    MM"
        Data "Zeit:"

Time21:
        ' Hoch         Runter        Links        Rechts       Enter
        Adr2 Time21 : Adr2 Time22 : Adr2 Time1 : Adr2 Time3 : Adr2 Time_ende
        Adr2 Menue_time21
        Data &B00000111
        '     1234567890123456
        Data "     Minuten"
        Data "Eingabe    MM"
        Data "Zeit:"

Time22:
        ' Hoch         Runter        Links        Rechts       Enter
        Adr2 Time21 : Adr2 Time22 : Adr2 Time1 : Adr2 Time3 : Adr2 Time_ende
        Adr2 Menue_time22
        Data &B00000111
        '     1234567890123456
        Data "     Minuten"
        Data "Eingabe    MM"
        Data "Zeit:"

Time3:
        ' Hoch         Runter        Links        Rechts      Enter
        Adr2 Time31 : Adr2 Time32 : Adr2 Time2 : Adr2 Null : Adr2 Time_ende
        Adr2 Menue_time3
        Data &B00000111
        '     1234567890123456
        Data "    Sekunden"
        Data "Eingabe       SS"
        Data "Zeit:"

Time31:
        ' Hoch         Runter        Links        Rechts      Enter
        Adr2 Time31 : Adr2 Time32 : Adr2 Time2 : Adr2 Null : Adr2 Time_ende
        Adr2 Menue_time31
        Data &B00000111
        '     1234567890123456
        Data "    Sekunden"
        Data "Eingabe       SS"
        Data "Zeit:"

Time32:
        ' Hoch         Runter        Links        Rechts      Enter
        Adr2 Time31 : Adr2 Time32 : Adr2 Time2 : Adr2 Null : Adr2 Time_ende
        Adr2 Menue_time32
        Data &B00000111
        '     1234567890123456
        Data "    Sekunden"
        Data "Eingabe       SS"
        Data "Zeit:"

Time_ende:
        ' Hoch       Runter      Links       Rechts      Enter
        Adr2 Null : Adr2 Null : Adr2 Null : Adr2 Null : Adr2 Null
        Adr2 Menue_time_ende
        Data &B00000000
        '     1234567890123456
        Data ""
        Data ""
        Data ""


' Statemachine für universelle Datumeingabe
Date1:
        ' Hoch         Runter        Links             Rechts       Enter
        Adr2 Date11 : Adr2 Date12 : Adr2 Date_ende : Adr2 Date2 : Adr2 Date_ende
        Adr2 Menue_date1
        Data &B00000111
        '     1234567890123456
        Data "       Tag"
        Data "Eingabe TT"
        Data "Datum:"

Date11:
        ' Hoch         Runter        Links             Rechts       Enter
```

```
        Adr2 Date11 : Adr2 Date12 : Adr2 Date_ende : Adr2 Date2 : Adr2 Date_ende
        Adr2 Menue_date11
        Data &B00000111
        '     1234567890123456
        Data "       Tag"
        Data "Eingabe TT"
        Data "Datum:"

Date12:
        ' Hoch         Runter        Links             Rechts       Enter
        Adr2 Date11 : Adr2 Date12 : Adr2 Date_ende : Adr2 Date2 : Adr2 Date_ende
        Adr2 Menue_date12
        Data &B00000111
        '     1234567890123456
        Data "       Tag"
        Data "Eingabe TT"
        Data "Datum:"

Date2:
        ' Hoch         Runter        Links        Rechts       Enter
        Adr2 Date21 : Adr2 Date22 : Adr2 Date1 : Adr2 Date3 : Adr2 Date_ende
        Adr2 Menue_date2
        Data &B00000111
        '     1234567890123456
        Data "      Monat"
        Data "Eingabe    MM"
        Data "Datum:"

Date21:
        ' Hoch         Runter        Links        Rechts       Enter
        Adr2 Date21 : Adr2 Date22 : Adr2 Date1 : Adr2 Date3 : Adr2 Date_ende
        Adr2 Menue_date21
        Data &B00000111
        '     1234567890123456
        Data "      Monat"
        Data "Eingabe    MM"
        Data "Datum:"

Date22:
        ' Hoch         Runter        Links        Rechts       Enter
        Adr2 Date21 : Adr2 Date22 : Adr2 Date1 : Adr2 Date3 : Adr2 Date_ende
        Adr2 Menue_date22
        Data &B00000111
        '     1234567890123456
        Data "      Monat"
        Data "Eingabe    MM"
        Data "Datum:"

Date3:
        ' Hoch         Runter        Links        Rechts      Enter
        Adr2 Date31 : Adr2 Date32 : Adr2 Date2 : Adr2 Null : Adr2 Date_ende
        Adr2 Menue_date3
        Data &B00000111
        '     1234567890123456
        Data "      Jahr"
        Data "Eingabe       JJ"
        Data "Datum:"

Date31:
        ' Hoch         Runter        Links        Rechts      Enter
        Adr2 Date31 : Adr2 Date32 : Adr2 Date2 : Adr2 Null : Adr2 Date_ende
        Adr2 Menue_date31
        Data &B00000111
        '     1234567890123456
        Data "      Jahr"
        Data "Eingabe       JJ"
        Data "Datum:"

Date32:
        ' Hoch         Runter        Links        Rechts      Enter
        Adr2 Date31 : Adr2 Date32 : Adr2 Date2 : Adr2 Null : Adr2 Date_ende
        Adr2 Menue_date32
        Data &B00000111
        '     1234567890123456
        Data "      Jahr"
        Data "Eingabe       JJ"
        Data "Datum:"

Date_ende:
        ' Hoch       Runter      Links       Rechts      Enter
```

```
        Adr2 Null : Adr2 Null : Adr2 Null : Adr2 Null : Adr2 Null
        Adr2 Menue_date_ende
        Data &B00000000
        '     1234567890123456
        Data ""
        Data ""
        Data ""


'-- End Bedatung der Menue-State-Machine -----------------------------------



'#################################### END ###################################


'################################## Historie ################################
' 09.12.2008 : Version 1.0
'              Erste offizielle und vollständig umgesetzte Produktivsoftware
'---------------------------------------------------------------------------
' 11.12.2008 : Version 1.1
'              Kommentare und Traceausgaben im gesamten Code ergänzt
'############################################################################
```

Software 17: Der Source-Code zum Projekt ConvCtrl

12 Mechanik

12.1 Elektromechanische Komponenten

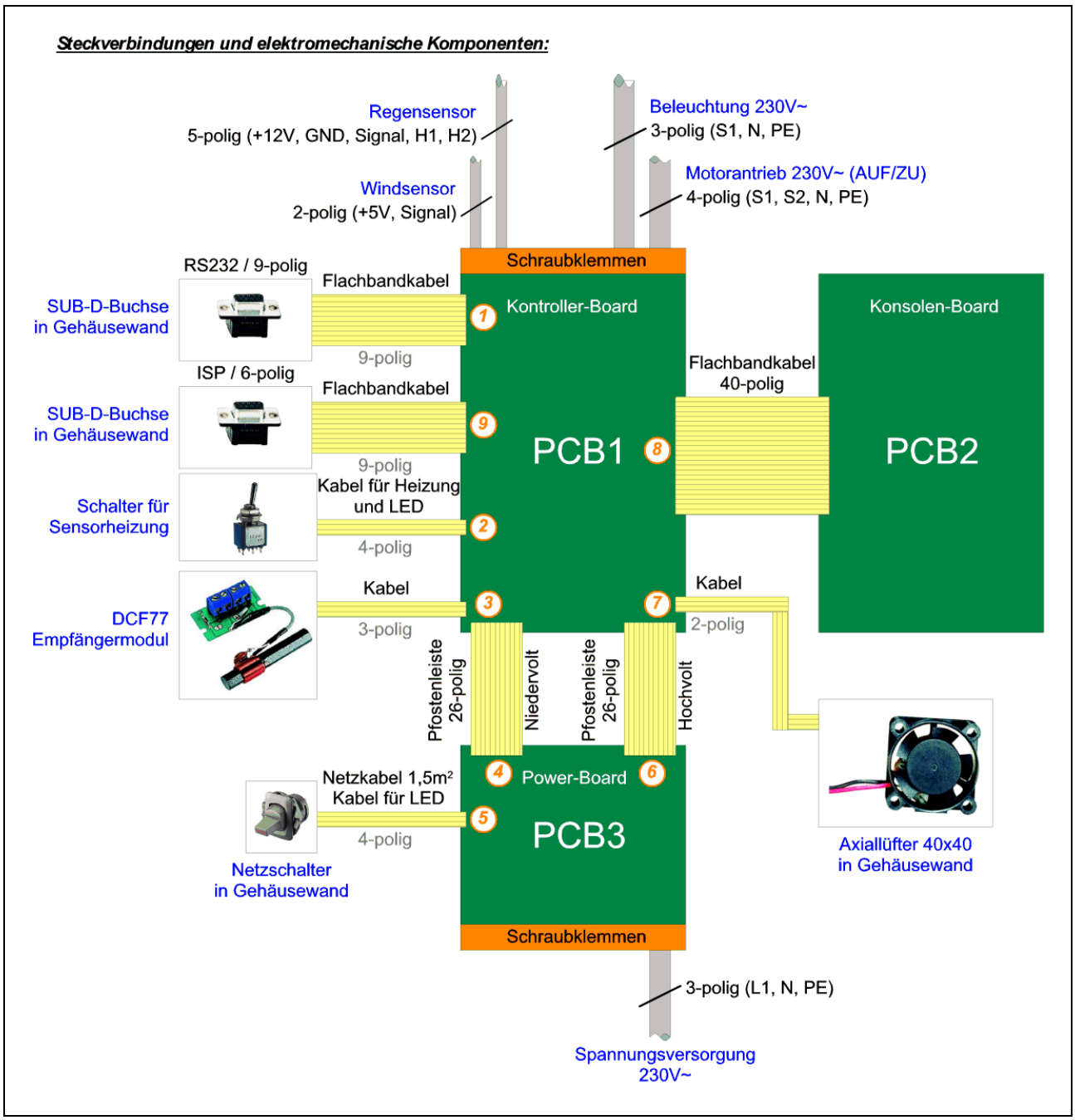


Abbildung 196: Steckverbindungen und elektromechanische Komponenten

12.2 Maßgebende Prinzipzeichnungen Gehäuse

12.2.1 Prinzipielle Gesamtansicht

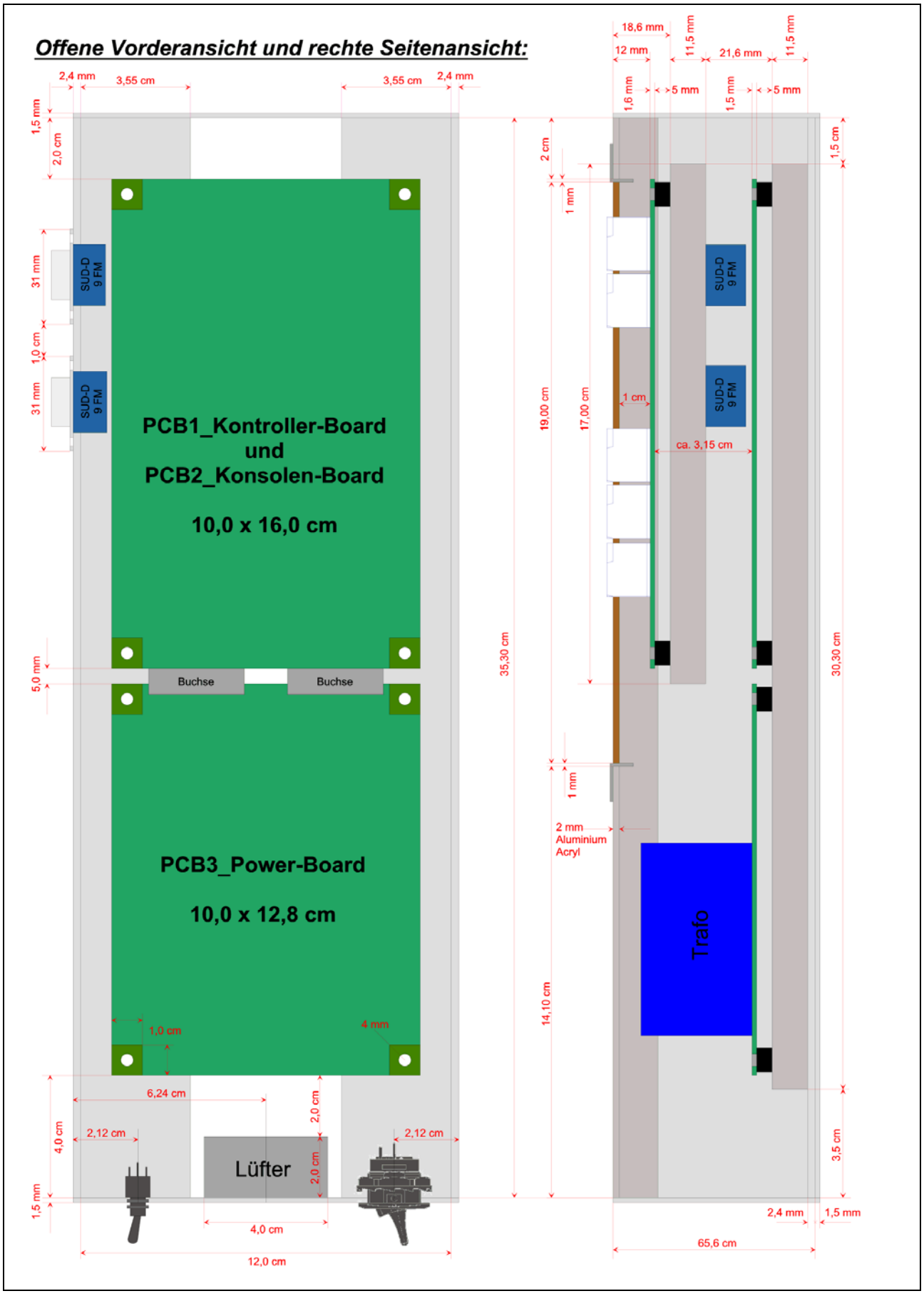


Abbildung 197: Gehäuse prinzipielle Gesamtansicht

Gesamtansicht geschlossenes Gehäuse:

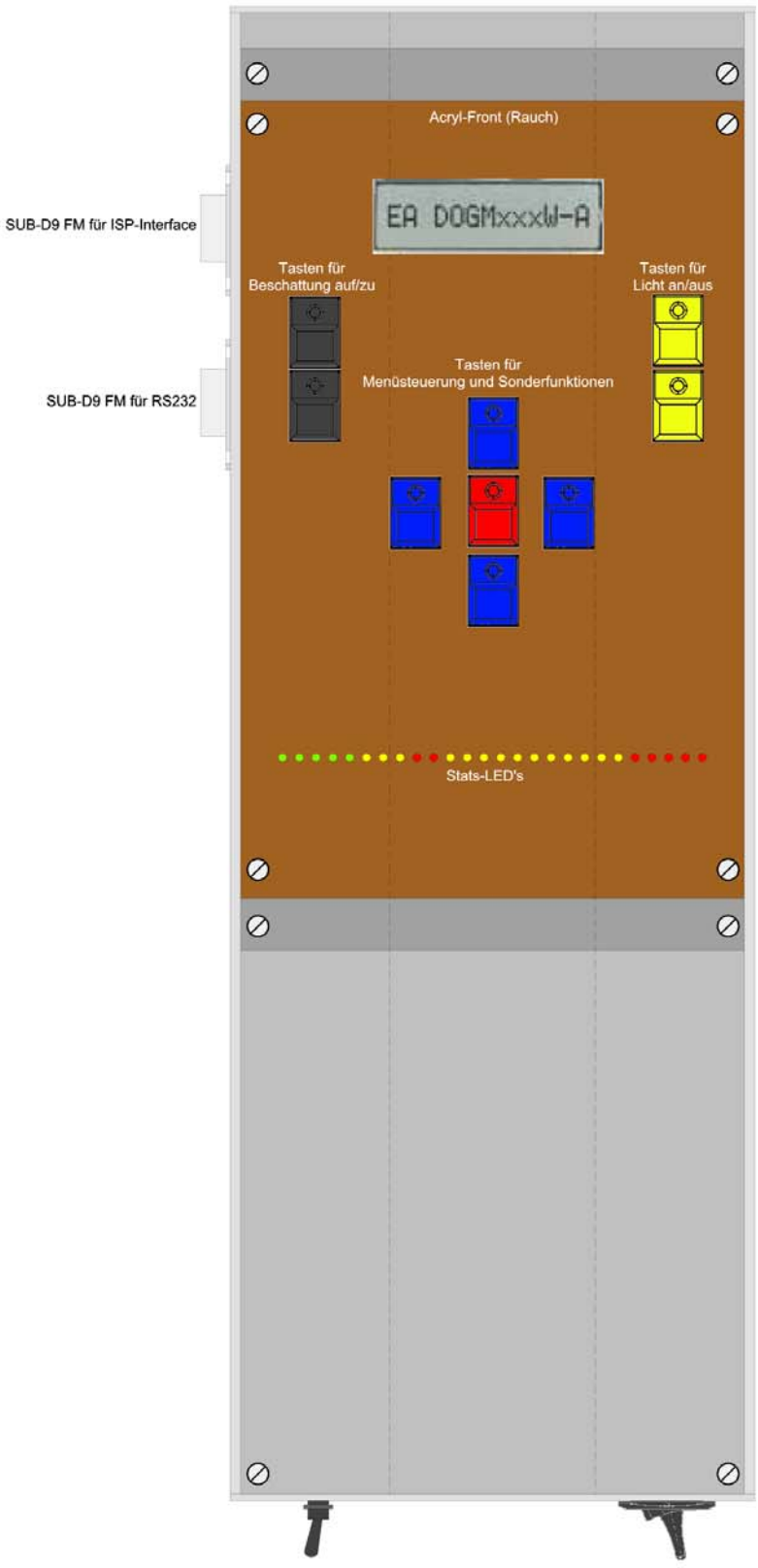


Abbildung 198: Gehäuse prinzipielle Gesamtansicht Front

12.2.2 Draufsichten

Untere und obere Deckelansicht:

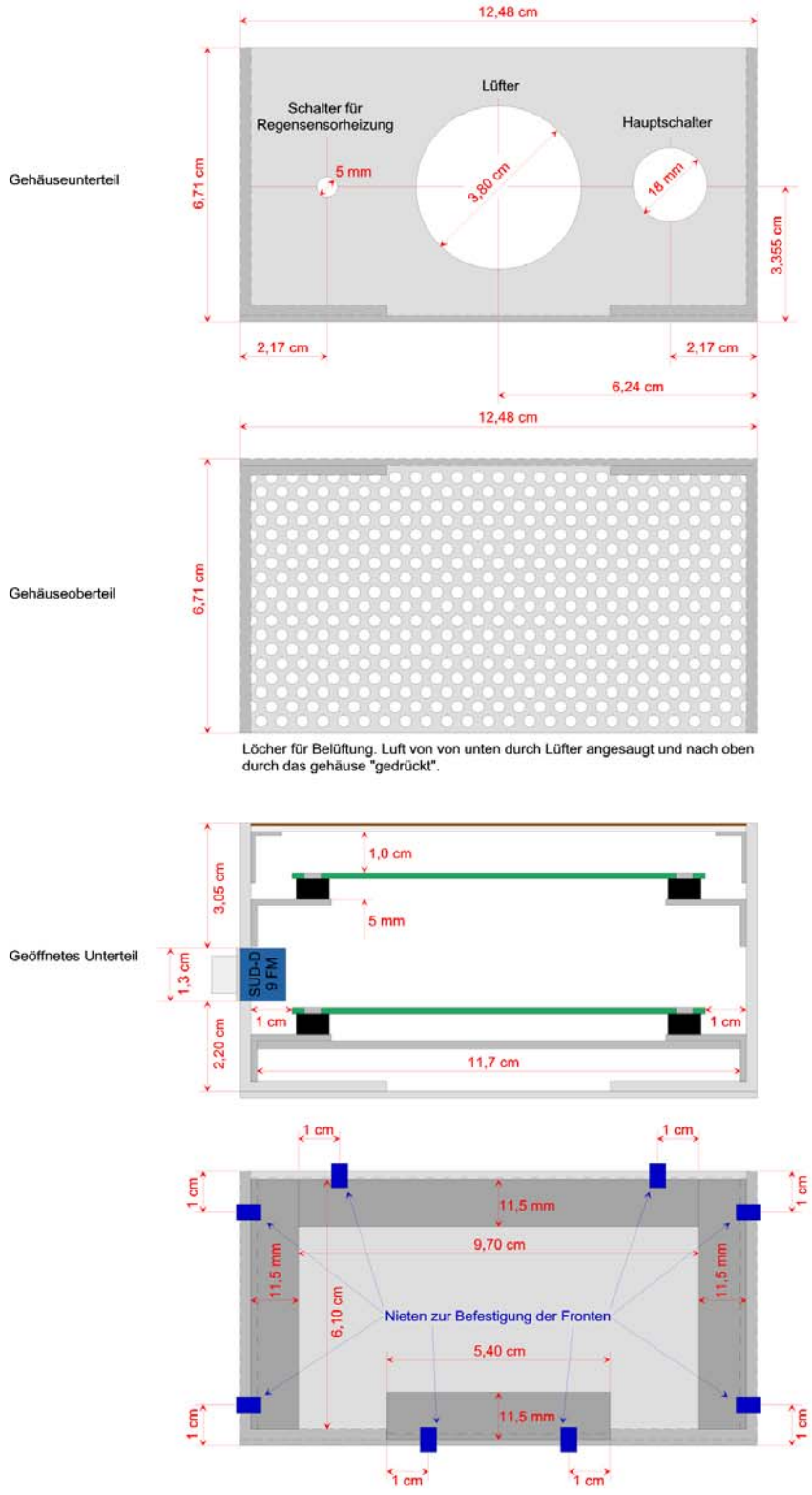


Abbildung 199: Gehäuse Draufsichten

12.2.3 Rückseite

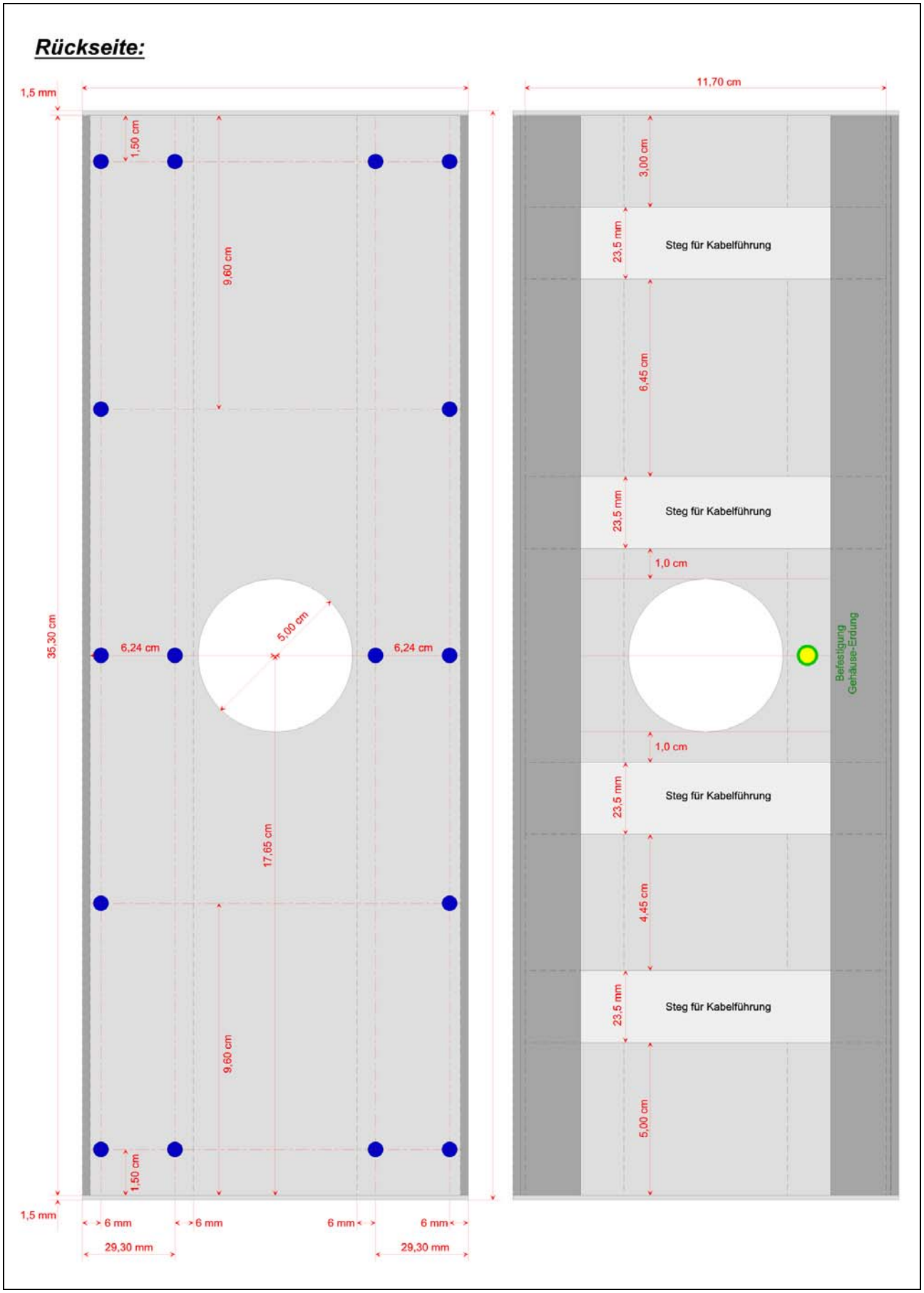


Abbildung 200: Gehäuse Rückseite

12.2.4 Seitenansichten

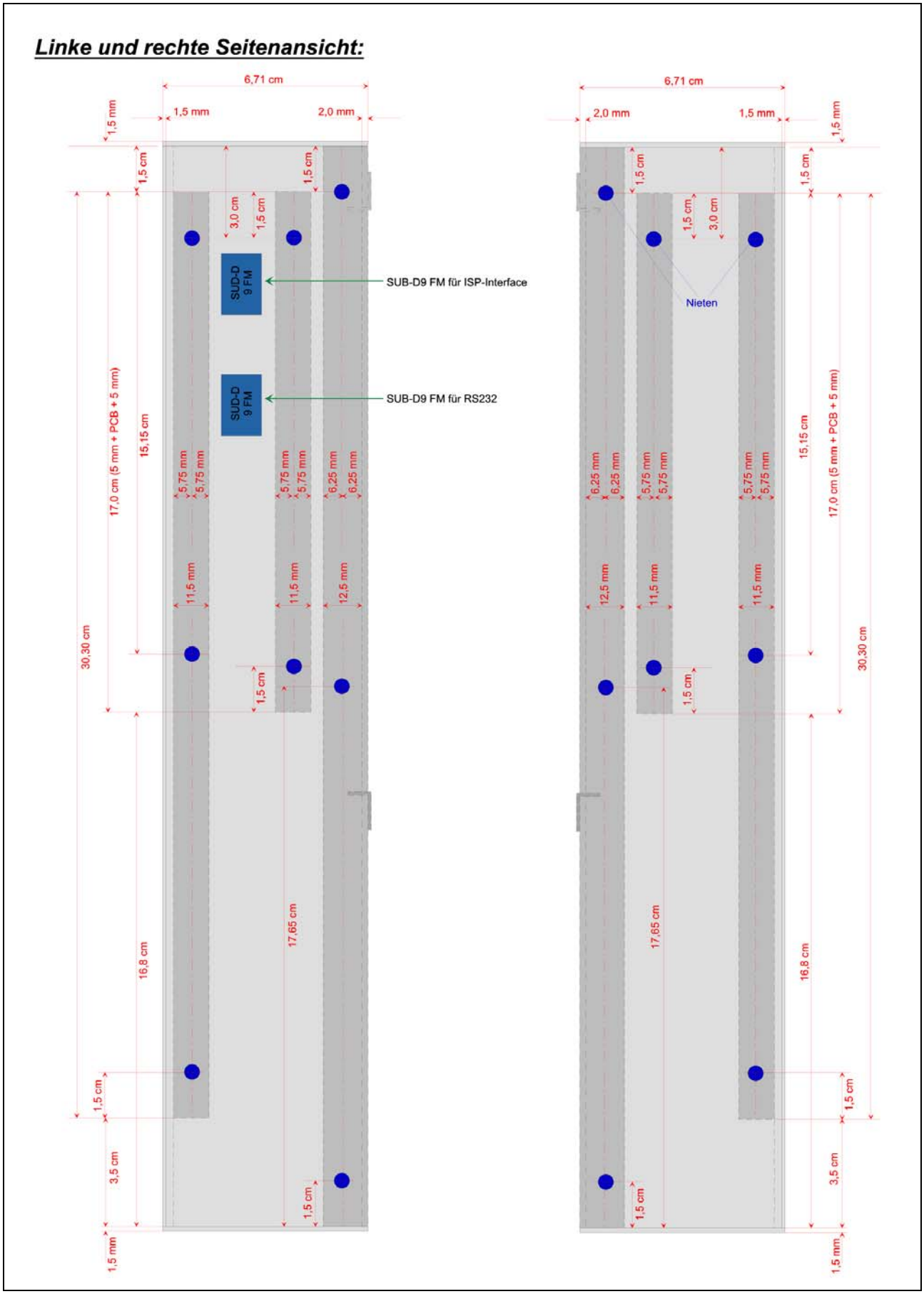


Abbildung 201: Gehäuse Seitenansichten

12.2.5 Materialauswahl

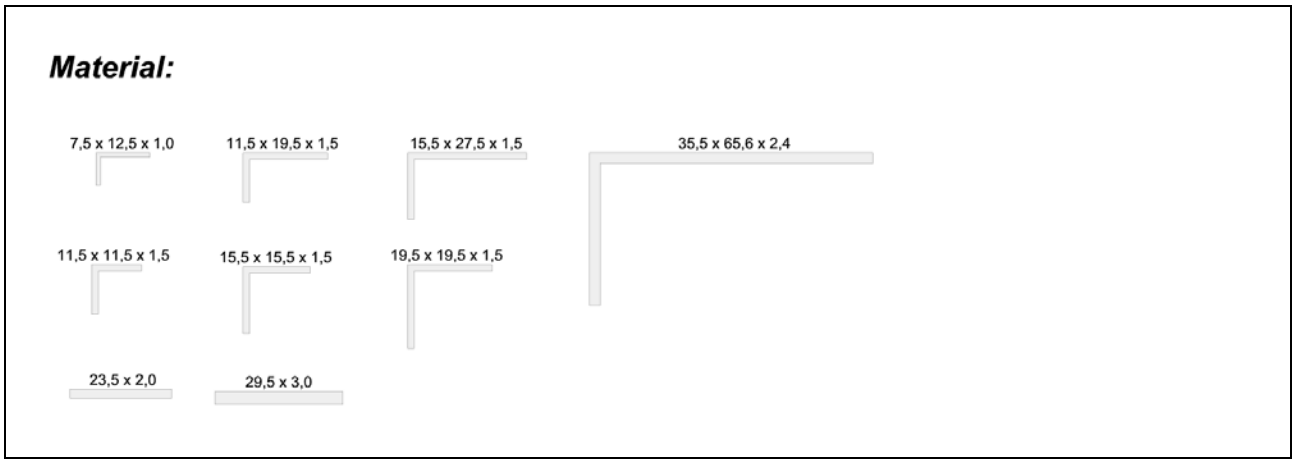


Abbildung 202: Gehäuse Materialauswahl

12.2.6 Notizen

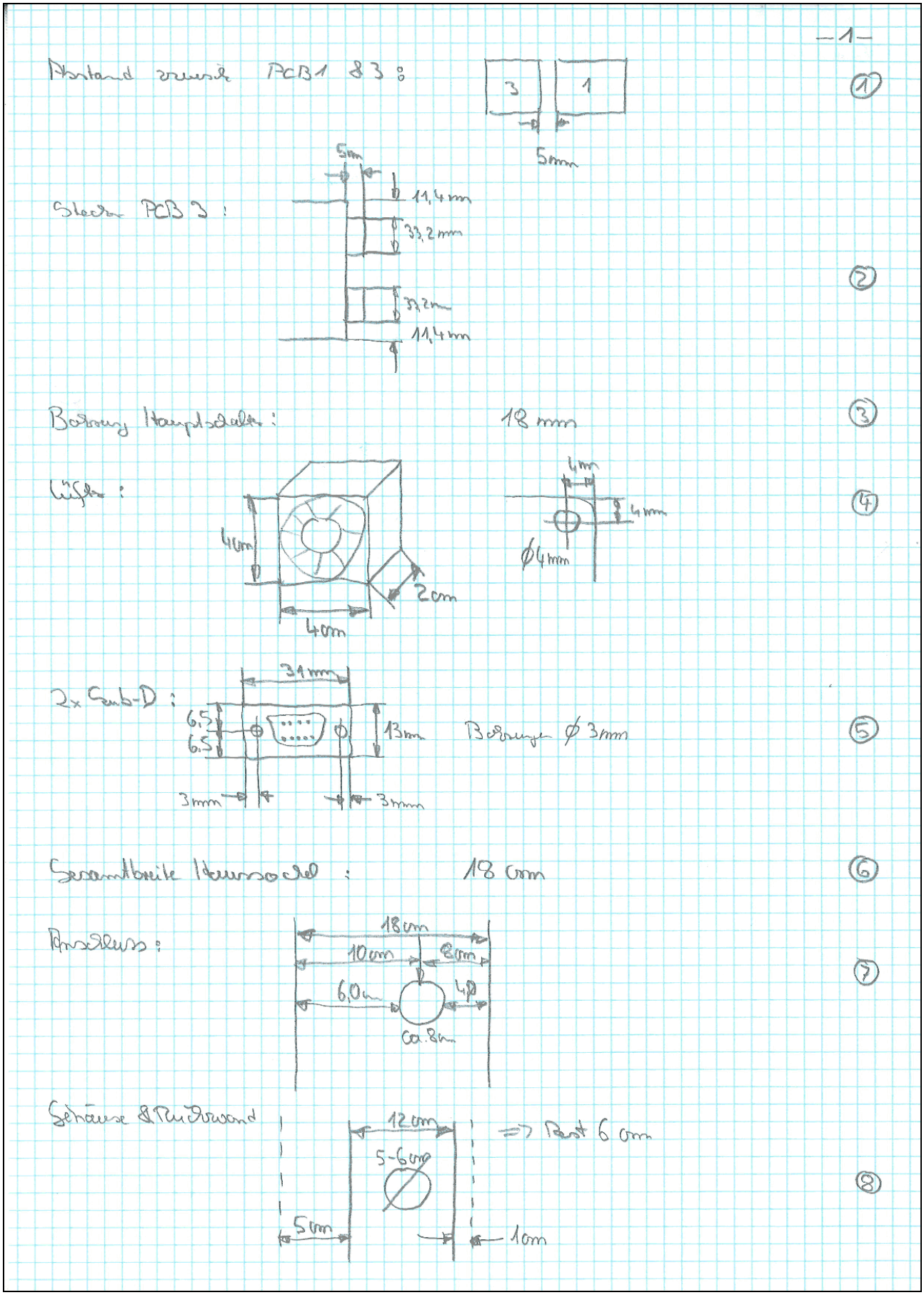


Abbildung 203: Notizen 1

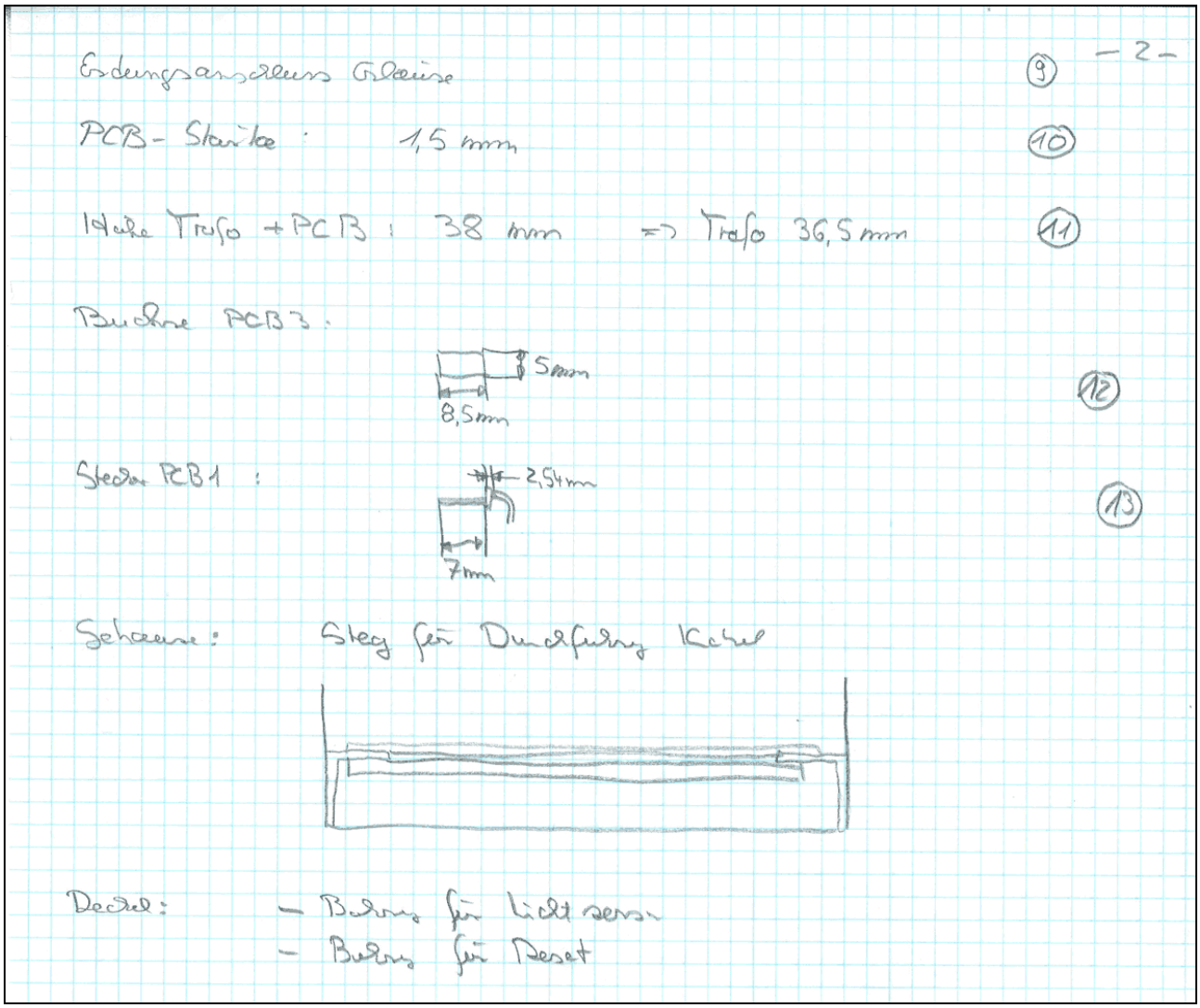


Abbildung 204: Notizen 2

12.3 Das Gehäuse zur Wintergartensteuerung

12.3.1 Wandansicht

Wandansicht

12.3.2 Bodeneinheit mit Lüfter und Schaltern



Abbildung 205: Gehäuse Bodeneinheit von außen

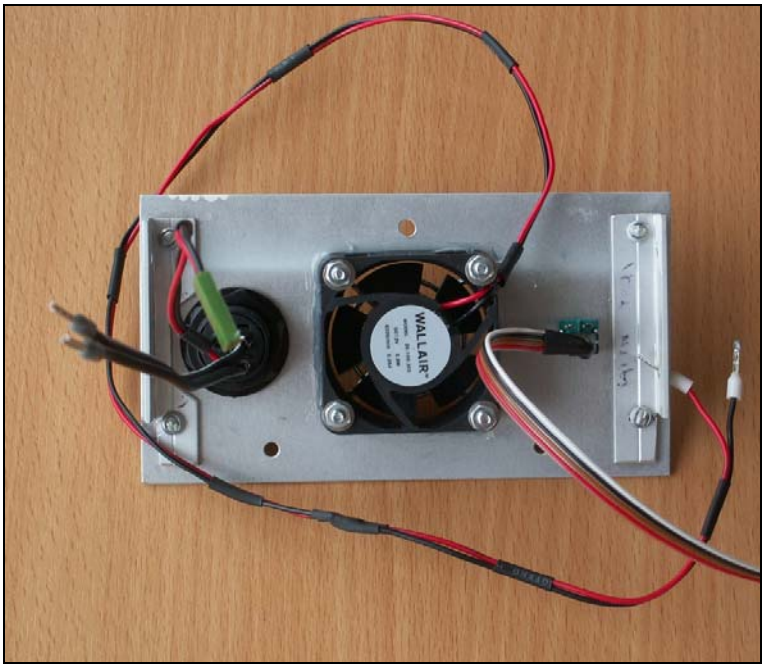


Abbildung 206: Gehäuse Bodeneinheit von innen

13 Verwendete Software

Zur Erstellung dieses Projekts kam folgende Software zum Einsatz:

- Workstation: Betriebsystem Windows NT 4.0 SP 6a
- Notebook: Betriebsystem Windows XP Professional

- myAVR Workpad PLUS
  Laser & Co. Solutions GmbH
- BASCOM-AVR Basic Compiler
  MCS Electronics
- EAGLE 4.16r2
- ATMEL AVRStudio

- WinAVR

Zur Erstellung dieses Projekts kam folgende HW-Umgebung und SDK's zum Einsatz:

- ATMEL STK500
- USB-Programmer mySmartUSB
  Laser & Co. Solutions GmbH
- myAVR Bausatz PLUS 1 LPT
  Laser & Co. Solutions GmbH
- myAVR TWI Add-On EEPROM
  Laser & Co. Solutions GmbH
- myAVR TWI Add-On Echtzeituhr
  Laser & Co. Solutions GmbH
- myAVR TWI Temperatursensor
  Laser & Co. Solutions GmbH
- myMultiProg LPT
  Laser & Co. Solutions GmbH
- myMultiProg USB
  Laser & Co. Solutions GmbH
- myDigitalOut
  Laser & Co. Solutions GmbH
- Atmel-Evaluations-Board V2.0
  Pollin Electronic GmbH
- Conrad Regensensor 115240
- Conrad DCF77 Empfängerplatine 641138
- Windgeber
  rb-Meßtechnik Reinhardt

14 Mitgeltende Unterlagen

14.1 Bücher und Literatur

- myAVR Lehrbuch Mikrocontroller-Programmierung
  Laser & Co. Solutions GmbH
- LCD Lehrheft
  Laser & Co. Solutions GmbH
- Projekt „myTWI"
  Laser & Co. Solutions GmbH
- Mikrocomputertechnik mit Controllern der Atmel AVR-RISC-Familie
  G. Schmitt, Oldenburgverlag, ISBN 3-486-58016-7
- Leiterplattendesign mit EAGLE
  André Ketler, Marc Neujahr, mitp Verlag, ISBN 978-3-8266-1340-1
- Messen, Steuern und Regeln mit AVR_Controllern
  Wolfgang Trampert, Franzis Verlag, ISBN 3-7723-4298-1
- Programmierung der AVR RISC Mikrocontroller mit BASCOM-AVR
  Claus Kühnel, Books on Demand, ISBN 3907857046

14.2 Internet

14.2.1 Firmen und Foren

SW

MCS Electronics
URL: http://www.mcselec.com/
- BASCOM-AVR BASIC Compiler und Forum

HW

Bauelemente:

Conrad Electronic
URL: http://www.conrad.de
- Bauelemente und Zubehör (zum Teil aber sehr teuer)

Pollin Electronic
URL: http://www.pollin.de/shop/shop.php
- Bauelemente
- Atmega Evaluationsboard

Reichelt elektronik
URL: http://www.reichelt.de/
- Günstige Bauelemente
- STK500 von ATMEL

Farnell Deutschland
URL: http://de.farnell.com/jsp/home/homepage.jsp
- Baulelemente

ELV
URL: http://shop.elv.de/output/controller.aspx
- Günstige Bauelemente
- LCD-Displays

RS Components GmbH
URL: http://www.rsonline.de
- Große Auswahl an Bauelementen zu guten Preisen

Hersteller und Spezialitäten:

ATMEL Corporation
URL: http://www.atmel.com/
- ATmega
- STK500

Sensirion
URL: http://www.sensirion.com/
- Halbleiterhersteller für Temperatursensoren und Feuchtigkeitssensoren

TAOS
URL: http://www.taosinc.com
- Halbleiterhersteller für Lichtsensoren

Riesen + Kern GmbH

URL: http://www.driesen-kern.de
- Deutscher Distributor für Sensirion Halbleiter

rb-Messtechnik Reinhardt
URL: http://www.rb-messtechnik.de
- Windgeber

DatasheetCatalog.COM
URL: http://www.datasheetcatalog.com
- Datenblätter zu fast allen bekannten elektr. Bauelementen

Worls Of Electronic – Elektronikprojekte
URL: http://www.woe.onlinehome.de/projekte.htm
- AVR JTAG Emulator

Plantinenservice und Hersteller:

GS Electronic
Sven Schult
Spillbähnstraße 19a
53844 Troisdorf

Tel. 02241-3010465
Fax 02241-3010469

eMail gselectronic@gsel.de
URL: http://www.gsel.com/
- Platinenservice
- Einzelstücke
- Kleinserien

LeitOn GmbH
Gottlieb-Dunkel-Str. 47-48
12099 Berlin

Tel.: +49-(0)30-701 73 49-0
Fax: +49-(0)30-701 73 49-19

E-Mail: kontakt@leiton.de
URL: http://www.leiton.de
- Platinenservice
- Einzelstücke
- Kleinserien
- Sehr günstige Preise
- Leiton stellt Download für Eagle DesignRules zur Verfügung

14.2.2 ATmega SW und HW-Lösungen

Laser & Co. Solutions GmbH
URL: http://www.myavr.de
- Bausätze zum ATmega8
- SW-Lösungen zum Selbststudium
- Dokumente

14.2.3 Foren

RoboterNetz.de
URL: http://www.roboternetz.de

- Großer Portal für Robotik, Elektronik und Mikrocontroller

MCS Electronics
URL: http://www.mcselec.com/
- Forum rund um BASCOM-AVR

AVR feaks
URL: http://www.avrfreaks.net/
- Forum AVRFREAKS.NET

Pony-Prog Tutorial
URL: http://www.mikrocontroller.net/articles/Pony-Prog_Tutorial
- Pony-Prog Tutorial

QSLnet
URL: http://www.qsl.net
URL: http://www.qsl.net/pa3ckr/index.html
URL: http://www.qsl.net/pa3ckr/bascom%20and%20avr/arrays%20and%20data/index.html
- Forum für Elektronik und SW-Lösungen (entstanden aus Radio Amateur Community)
- Zusammenfassung BASCOM und AVR Lösungen (Arrays usw.)

AVR_Praxis
URL: http://www.avr-praxis.de/index.php
AVR-PRAXiS ist ein Forum, das ausschließlich für einen Gedankenaustausch und als Diskussionsplattform für Interessierte bereitstelle, welche sich privat, durch das Studium oder beruflich mit der AVR-Mikrocontrollerfamilie beschäftigen wollen oder müssen.

Microcontroller.net
URL: http://www.mikrocontroller.net
Großes Portal mit Forum und Chat

BASCOM-Forum
URL: http://www.bascom-forum.de
Forum für Projekte, Hardware und Diskussionen

Infos rund um den ATmega:
URL: http://www.dieelektronikerseite.de/uC%20Ecke/Module/Ports%20-%20Wenn%20der%20AVR%20steuert.htm
URL: http://www.kreatives-chaos.com/artikel/avr-grundschaltungen

Informationen zum TWI / I²C-Bus:
URL: http://www.roboternetz.de/wissen/index.php/I2C
URL: http://www.roboternetz.de/wissen/index.php/TWI
URL: http://www.roboternetz.de/wissen/index.php/TWI_Praxis
URL: http://www.roboternetz.de/wissen/index.php/TWI_Praxis_Multimaster
URL: http://www.roboternetz.de/wissen/index.php/Bascom_I2C_Master
URL: http://www.roboternetz.de/wissen/index.php/Bascom_und_USI-Kommunikation
URL: http://www.roboternetz.de/wissen/index.php/Bascom_Soft-I2c_Library
URL: http://www.roboternetz.de/wissen/index.php/Bascom_Inside-Code

14.2.4 Schaltnetzteile

Spannungswandler fürs Auto & Co.
URL: http://www.blafusel.de/misc/spannung.html

14.2.5 Dimmer

8-fach 230V-Dimmer
URL: http://www.piware.de/hardware.shtml

Diese Schaltung realisiert eine Bühnenbeleuchtung mit 8 frei dimmbaren, über eine serielle Schnittstelle steuerbaren Lampen. Dabei wurde auf Einfachheit, Robustheit und vor allem auf einen günstigen Preis (in der Größenordnung von EUR 50) geachtet.

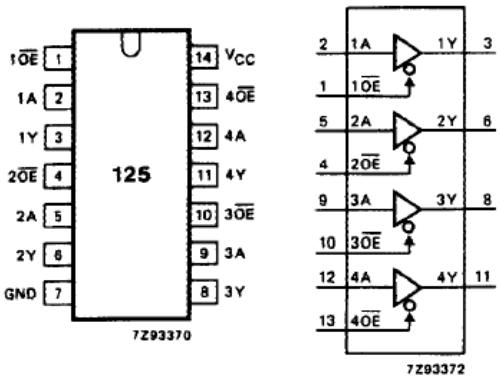
Lichtsteuerung
URL: http://www-projekte.emsp.tu-berlin.de/ele/ws0506/ele41/das_projekt.html
Wir bauen ein 3x6 kanaliges Dimmpack zur Steuerung von Lichtanlagen
und Effektgeräten mit Beatdetektor

Steckdosenleiste mit Dimmer
URL: http://www.umnicom.de/Elektronik/Mikrokontroller/Atmel/AtVolt/AtVolt.html
Entwickelt wird die Steuerung für eine Steckdosenleiste. Zwei Steckdosen verfügen über ein Dimmerfunktion und zwei Steckdosen über eine Schaltfunktion. Alle Steckdosen verfügen über eine Rückkopplung, d.h. es wird detektiert, ob die angeschlossenen Lampen auch wirklich leuchten wenn sie eingeschaltet werden. Die Steuerung erfolgt über eine Infrarot Fernbedienung und/oder über eine serielle Schnittstelle.

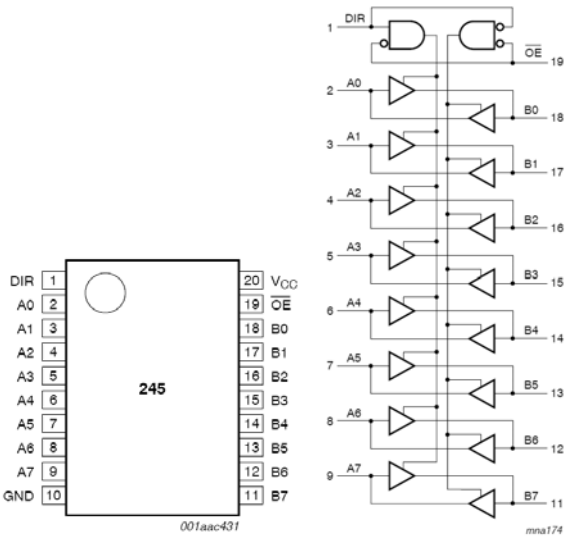
DMX Dimmerpack
URL: http://www.hoelscher-hi.de/hendrik/light/dmxdimmer.htm
Das hier beschriebene einphasige Dimmerpack ist durch seine minimalistische Ausstattung sehr günstig und robust. Dennoch ist es dank seiner Leistungslinearisierung, der vollen Auflösung von 256 Stufen und der Möglichkeit, einzelne Kanäle im Switchmodus zu betreiben, den meisten käuflichen Dimmerpacks im Billigstsegment an Performance überlegen. Mit bis zu acht Kanälen à 2kWmax. und einem Übertemperaturschutz sind auch größere Systeme problemlos zu realisieren. Durch das modulare Design und HotSwap-Möglichkeit ist der Dimmer sehr servicefreundlich und lässt sich einfach upgraden.

15 Entwicklungsbegleitende Notizen und Informationen

IC-Schaltung 74HC/HCT125N
Quad buffer/line driver; 3-state



IC-Schaltung 74HC/HCT245
Octal bus transceiver; 3-state



Transistor BC546B und BC547B

NPN general purpose transistor

NPN general purpose transistors BC546; BC547

FEATURES

- Low current (max. 100 mA)
- Low voltage (max. 65 V).

APPLICATIONS

- General purpose switching and amplification.

DESCRIPTION

NPN transistor in a TO-92; SOT54 plastic package.
PNP complements: BC556 and BC557.

PINNING

PIN	DESCRIPTION
1	emitter
2	base
3	collector

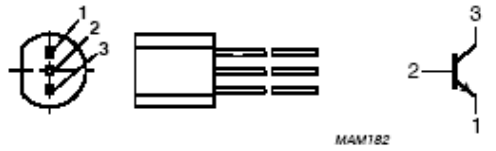


Fig.1 Simplified outline (TO-92; SOT54) and symbol.

LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V_{CBO}	collector-base voltage	open emitter			
	BC546		–	80	V
	BC547		–	50	V
V_{CEO}	collector-emitter voltage	open base			
	BC546		–	65	V
	BC547		–	45	V
V_{EBO}	emitter-base voltage	open collector			
	BC546		–	6	V
	BC547		–	6	V
I_C	collector current (DC)		–	100	mA
I_{CM}	peak collector current		–	200	mA
I_{BM}	peak base current		–	200	mA
P_{tot}	total power dissipation	$T_{amb} \leq 25$ °C; note 1	–	500	mW
T_{stg}	storage temperature		–65	+150	°C
T_j	junction temperature		–	150	°C
T_{amb}	operating ambient temperature		–65	+150	°C

NPN general purpose transistors BC546; BC547

THERMAL CHARACTERISTICS

SYMBOL	PARAMETER	CONDITIONS	VALUE	UNIT
$R_{th\ j\text{-}a}$	thermal resistance from junction to ambient	note 1	0.25	K/mW

Note

1. Transistor mounted on an FR4 printed-circuit board.

CHARACTERISTICS

T_j = 25 °C unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
I_{CBO}	collector cut-off current	I_E = 0; V_{CB} = 30 V	–	–	15	nA
		I_E = 0; V_{CB} = 30 V; T_j = 150 °C	–	–	5	µA
I_{EBO}	emitter cut-off current	I_C = 0; V_{EB} = 5 V	–	–	100	nA
h_{FE}	DC current gain BC546A BC546B; BC547B BC547C	I_C = 10 µA; V_{CE} = 5 V; see Figs 2, 3 and 4	 – – –	 90 150 270	 – – –	
	DC current gain BC546A BC546B; BC547B BC547C BC547 BC546	I_C = 2 mA; V_{CE} = 5 V; see Figs 2, 3 and 4	 110 200 420 110 110	 180 290 520 – –	 220 450 800 800 450	
V_{CEsat}	collector-emitter saturation voltage	I_C = 10 mA; I_B = 0.5 mA	–	90	250	mV
		I_C = 100 mA; I_B = 5 mA	–	200	600	mV
V_{BEsat}	base-emitter saturation voltage	I_C = 10 mA; I_B = 0.5 mA; note 1	–	700	–	mV
		I_C = 100 mA; I_B = 5 mA; note 1	–	900	–	mV
V_{BE}	base-emitter voltage	I_C = 2 mA; V_{CE} = 5 V; note 2	580	660	700	mV
		I_C = 10 mA; V_{CE} = 5 V	–	–	770	mV
C_c	collector capacitance	$I_E = i_e$ = 0; V_{CB} = 10 V; f = 1 MHz	–	1.5	–	pF
C_e	emitter capacitance	$I_C = i_c$ = 0; V_{EB} = 0.5 V; f = 1 MHz	–	11	–	pF
f_T	transition frequency	I_C = 10mA; V_{CE} = 5 V; f = 100 MHz	100	–	–	MHz
F	noise figure	I_C = 200 µA; V_{CE} = 5 V; R_S = 2 kΩ; f = 1 kHz; B = 200 Hz	–	2	10	dB

Notes

1. V_{BEsat} decreases by about 1.7 mV/K with increasing temperature.
2. V_{BE} decreases by about 2 mV/K with increasing temperature.

Diode 1N4148
Hight Speed Diodes

High-speed diodes 1N4148; 1N4448

FEATURES

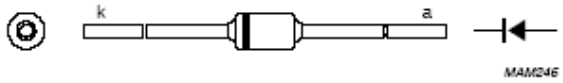
- Hermetically sealed leaded glass SOD27 (DO-35) package
- High switching speed: max. 4 ns
- General application
- Continuous reverse voltage: max. 75 V
- Repetitive peak reverse voltage: max. 100 V
- Repetitive peak forward current: max. 450 mA.

APPLICATIONS

- High-speed switching.

DESCRIPTION

The 1N4148 and 1N4448 are high-speed switching diodes fabricated in planar technology, and encapsulated in hermetically sealed leaded glass SOD27 (DO-35) packages.



The diodes are type branded.

Fig.1 Simplified outline (SOD27; DO-35) and symbol.

LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 60134).

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V_{RRM}	repetitive peak reverse voltage		–	100	V
V_R	continuous reverse voltage		–	75	V
I_F	continuous forward current	see Fig.2; note 1	–	200	mA
I_{FRM}	repetitive peak forward current		–	450	mA
I_{FSM}	non-repetitive peak forward current	square wave; T_j = 25 °C prior to surge; see Fig.4			
		t = 1 µs	–	4	A
		t = 1 ms	–	1	A
		t = 1 s	–	0.5	A
P_{tot}	total power dissipation	T_{amb} = 25 °C; note 1	–	500	mW
T_{stg}	storage temperature		–65	+200	°C
T_j	junction temperature		–	200	°C

Note

1. Device mounted on an FR4 printed circuit-board; lead length 10 mm.

ELECTRICAL CHARACTERISTICS

T_j = 25 °C unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V_F	forward voltage	see Fig.3			
	1N4148	I_F = 10 mA	–	1	V
	1N4448	I_F = 5 mA	0.62	0.72	V
		I_F = 100 mA	–	1	V
I_R	reverse current	V_R = 20 V; see Fig.5		25	nA
		V_R = 20 V; T_j = 150 °C; see Fig.5	–	50	µA
I_R	reverse current; 1N4448	V_R = 20 V; T_j = 100 °C; see Fig.5	–	3	µA
C_d	diode capacitance	f = 1 MHz; V_R = 0; see Fig.6	–	4	pF
t_{rr}	reverse recovery time	when switched from I_F = 10 mA to I_R = 60 mA; R_L = 100 Ω; measured at I_R = 1 mA; see Fig.7	–	4	ns
V_{fr}	forward recovery voltage	when switched from I_F = 50 mA; t_r = 20 ns; see Fig.8	–	2.5	V

THERMAL CHARACTERISTICS

SYMBOL	PARAMETER	CONDITIONS	VALUE	UNIT
$R_{th\ j\text{-}tp}$	thermal resistance from junction to tie-point	lead length 10 mm	240	K/W
$R_{th\ j\text{-}a}$	thermal resistance from junction to ambient	lead length 10 mm; note 1	350	K/W

Note

1. Device mounted on a printed circuit-board without metallization pad.

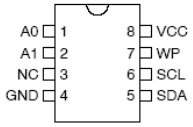
EEPROM ATMEL AT24C512:

Features:
• Low-voltage and Standard-voltage Operation
– 2.7 (VCC = 2.7V to 5.5V)
– 1.8 (VCC = 1.8V to 3.6V)
• Internally Organized 65,536 x 8
• 2-wire Serial Interface
• Schmitt Triggers, Filtered Inputs for Noise Suppression
• Bidirectional Data Transfer Protocol
• 1 MHz (5V), 400 kHz (2.7V) and 100 kHz (1.8V) Compatibility
• Write Protect Pin for Hardware and Software Data Protection
• 128-byte Page Write Mode (Partial Page Writes Allowed)
• Self-timed Write Cycle (5 ms Typical)
• High Reliability
– Endurance: 100,000 Write Cycles
– Data Retention: 40 Years
• Automotive Grade and Extended Temperature Devices Available
• 8-lead PDIP, 8-lead EIAJ SOIC, 8-lead JEDEC SOIC, 8-lead TSSOP,
8-lead LAP and 8-ball dBGATM Packages

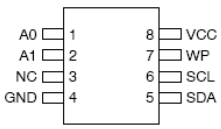
Pin Configurations

Pin Name	Function
A0 - A1	Address Inputs
SDA	Serial Data
SCL	Serial Clock Input
WP	Write Protect
NC	No Connect

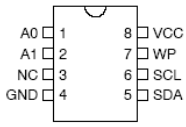
8-lead TSSOP



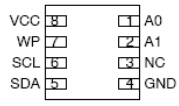
8-lead SOIC



8-lead PDIP

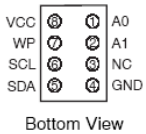


8-lead Leadless Array



Bottom View

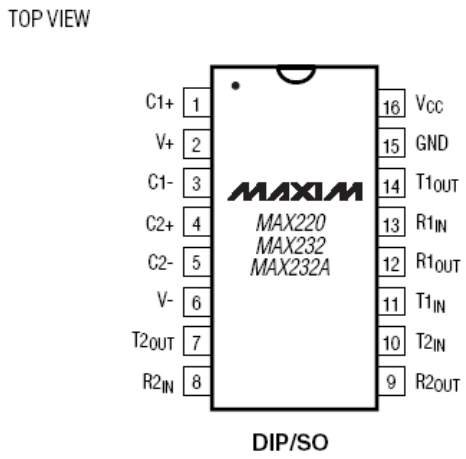
8-ball dBGA



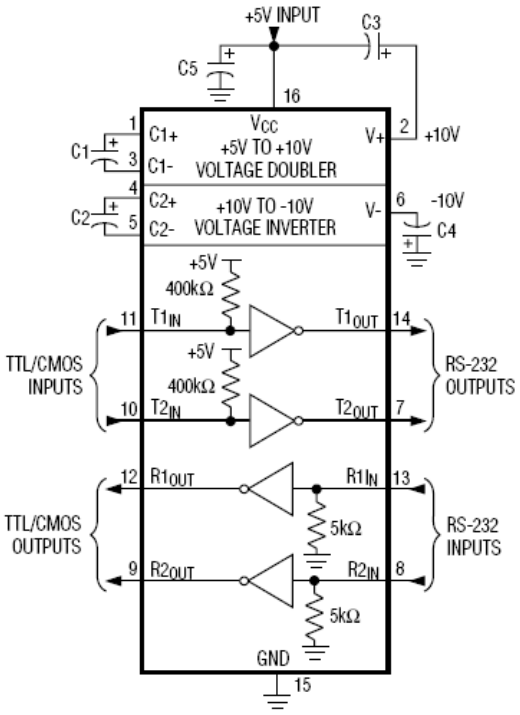
Bottom View

IC-Schaltung MAX232

+5V-Powered, Multichannel RS-232 Drivers/Receivers



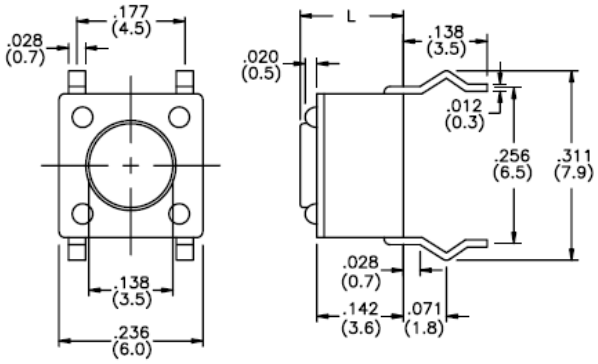
CAPACITANCE (µF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	4.7	4.7	10	10	4.7
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1



Kurzhubtaster Taster3301

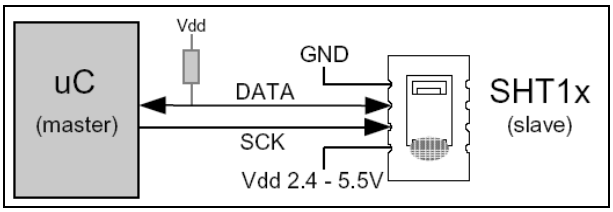
EU # PHAP3301



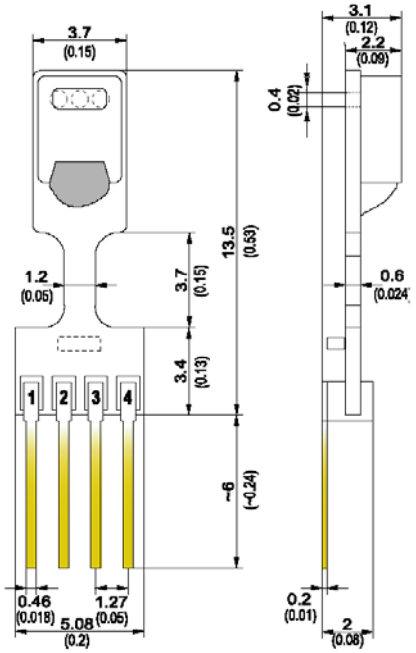


Dim. L (in./mm)	EU Model No.
.170 (4,3)	PHAP3301
.197 (5,0)	PHAP3301A
.374 (9,5)	PHAP3301B
.315 (8,0)	PHAP3301C
.512 (13,0)	PHAP3301D
.287 (7,3)	PHAP3301E
.335 (8,5)	PHAP3301F
.275 (7,0)	N/A

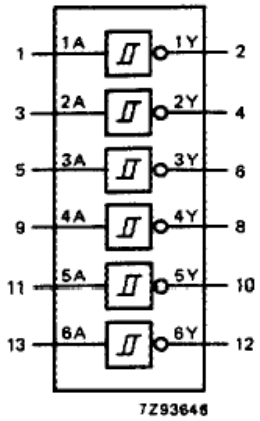
Temperatur- & Lüftfeuchtigkeitssensor SHT71



Pin	Name	Comment
1	SCK	Serial clock input
2	VDD	Supply
3	GND	Ground
4	DATA	Serial data bidirectional



IC-Schaltung 74HC14N
Hex inverting Schmitt trigger



Metallschichtwiderstände E12-Reihe:
(alle Werte in Ohm)

1,0	1,2	1,5	1,8	2,2	2,7	3,3	3,9	4,7	5,6	6,8	8,2
10	12	15	18	22	27	33	39	47	56	68	82
100	120	150	180	220	270	330	390	470	560	680	820
1k	1k2	1k5	1k8	2k2	2k7	3k3	3k9	4k7	5k6	6k8	8k2
10k	12k	15k	18k	22k	27k	33k	39k	47k	56k	68k	82k
100k	120k	150k	180k	220k	270k	330k	390k	470k	560k	680k	820k
1M											

Lichtsensor TAOS TSL250R-LF

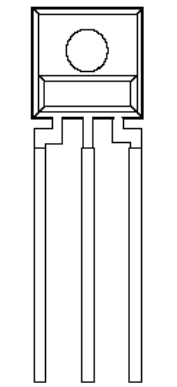
TAOS TEXAS ADVANCED OPTOELECTRONIC SOLUTIONS®

TSL250R, TSL251R, TSL252R
LIGHT-TO-VOLTAGE OPTICAL SENSORS

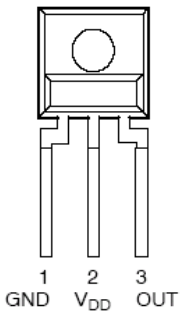
TAOS028G – JUNE 2007

- Monolithic Silicon IC Containing Photodiode, Operational Amplifier, and Feedback Components
- Converts Light Intensity to a Voltage
- High Irradiance Responsivity, Typically 137 mV/(μW/cm²) at λ_p = 635 nm (TSL250R)
- Compact 3-Lead Clear Plastic Package
- Single Voltage Supply Operation
- Low Dark (Offset) Voltage....10 mV Max
- Low Supply Current.....1.1 mA Typical
- Wide Supply-Voltage Range.... 2.7 V to 5.5 V
- Replacements for TSL250, TSL251, and TSL252
- RoHS Compliant (–LF Package Only)

PACKAGE S
SIDELOOKER
(FRONT VIEW)



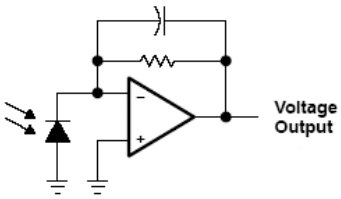
PACKAGE SM
SURFACE MOUNT
SIDELOOKER
(FRONT VIEW)



Description

The TSL250R, TSL251R, and TSL252R are light-to-voltage optical sensors, each combining a photodiode and a transimpedance amplifier (feedback resistor = 16 MΩ, 8 MΩ, and 2.8 MΩ respectively) on a single monolithic IC. Output voltage is directly proportional to the light intensity (irradiance) on the photodiode. These devices have improved amplifier offset-voltage stability and low power consumption and are supplied in a 3-lead clear plastic sidelooker package with an integral lens. When supplied in the lead (Pb) free package, the device is RoHS compliant.

Functional Block Diagram



TSL250R, TSL251R, TSL252R
LIGHT-TO-VOLTAGE OPTICAL SENSORS

TAOS028G – JUNE 2007

Terminal Functions

TERMINAL NAME	TERMINAL NO.	DESCRIPTION
GND	1	Ground (substrate). All voltages are referenced to GND.
OUT	3	Output voltage
V_{DD}	2	Supply voltage

Absolute Maximum Ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage, V_{DD} (see Note 1) . . . . . 6 V
Output current, I_O . . . . . ±10 mA
Duration of short-circuit current at (or below) 25°C (see Note 2) . . . . . 5 s
Operating free-air temperature range, T_A . . . . . –25°C to 85°C
Storage temperature range, T_{stg} . . . . . –25°C to 85°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds . . . . . 260°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. All voltages are with respect to GND.
2. Output may be shorted to supply.

Recommended Operating Conditions

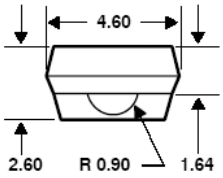
	MIN	NOM	MAX	UNIT
Supply voltage, V_{DD}	2.7		5.5	V
Operating free-air temperature, T_A	0		70	°C

MECHANICAL INFORMATION

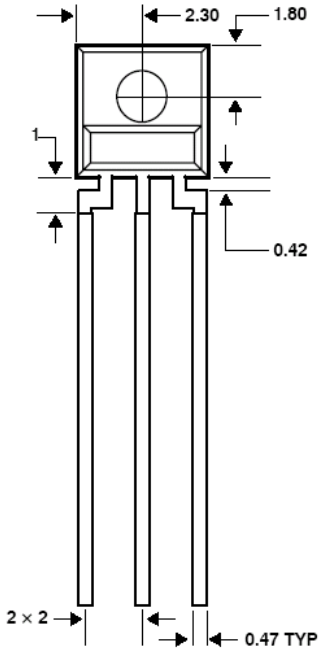
The devices are supplied in a clear plastic three-lead package (S). The integrated photodiode active area is typically 1,0 mm^2 (0.0016 in^2) for TSL250R, 0,5 mm^2 (0.00078 in^2) for the TSL251R, and 0,26 mm^2 (0.0004 in^2) for the TSL252R.

PACKAGE S **PLASTIC SINGLE-IN-LINE SIDE-LOOKER PACKAGE**

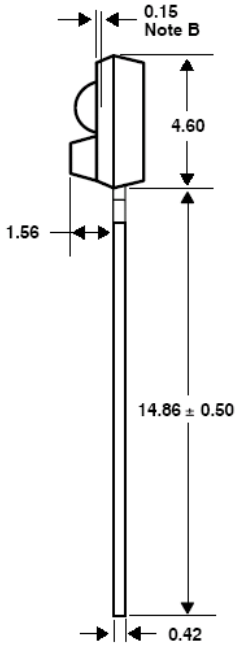
TOP VIEW



FRONT VIEW



SIDE VIEW



Pb
Lead Free
Available

NOTES:
A. All linear dimensions are in millimeters; tolerance is ± 0.25 mm unless otherwise stated.
B. Dimension is to center of lens arc, which is located below the package face.
C. The integrated photodiode active area is typically located in the center of the lens and 0.97 mm below the top of the lens surface.
D. Index of refraction of clear plastic is 1.55.
E. Lead finish for TSL25xR: solder dipped, 63% Sn/37% Pb. Lead finish for TSL25xR–LF: solder dipped, 100% Sn.
F. This drawing is subject to change without notice.

Piezo-Signalgeber Ekulit RMP-14SP

RMP-14SP (Artikel-Nr. 185010)

EKULIT

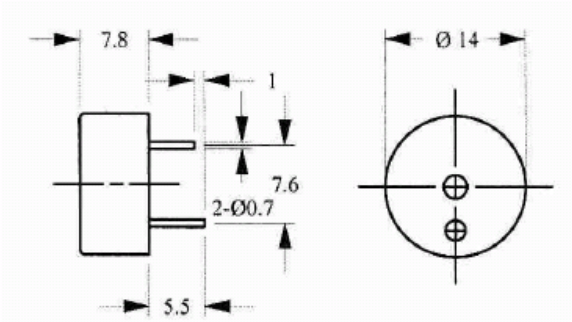
SPECIFICATIONS:

TYPE	UNIT	RMP-14SP
• Min. Sound Output at 10cm	dB	85
Rated Voltage	V	12
Operating Voltage	V	3~16
Resonant Frequency	Hz	4.000±500
• Max. Current Consumption	mA	8
Tone		Continuous
Operating Temperature	°C	-20~+70
Storage Temperature	°C	-30~+80

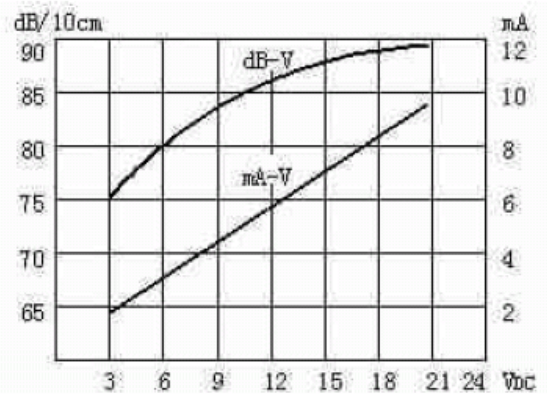
• Value applying at rated voltage

DIMENSIONS :

(Unit: mm)



Frequency response :



Eingabetaster

DIGITAST
Microminiature SPDT, Key Switches



Features/Benefits

- *High reliability / long life*
- *Designed for low level switching*
- *Ready to implement on PCB*
- *Latching function available*
- *Several button options*

Typical Applications

- *Medical*
- *Instrumentation*
- *Industrial electronics*
- *Audio electronics*

Construction

FUNCTION: momentary or push-push
CONTACT ARRANGEMENT: 1 change over contact SPDT, NO
MODE OF SWITCHING: Non-shorting
DISTANCE BETWEEN BUTTON CENTERS:
HORIZONTAL MINIMUM: 12,7 (0.500) or 17,78 (0.700)
VERTICAL MINIMUM: 17,78 (0.700)
TERMINALS: PC pins
MOUNTING: Soldering, centering pins

Mechanical

TOTAL TRAVEL: ≤ 3 (.0018)
SWITCHING TRAVEL: 1,5 (0.0591)
LATCHING TRAVEL: 1,8 (0.0709)
OPERATING FORCE: 1.5 +/- 0.5 N (150 +/- 50 grams)

Process

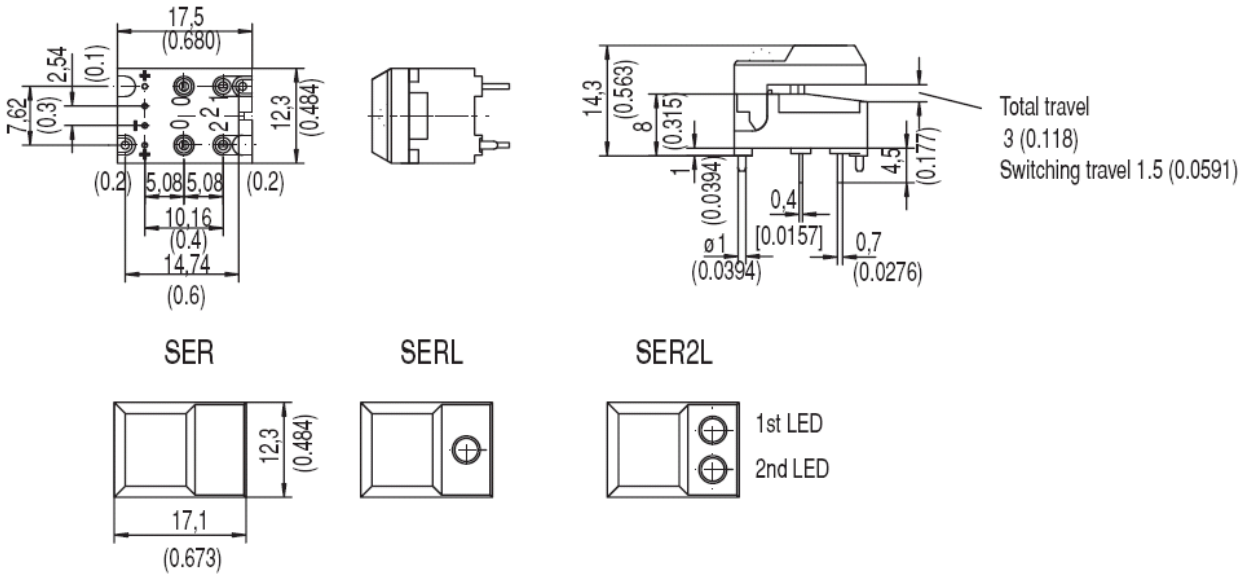
SOLDERABILITY: Sealed contacts
Max. soldering time and temperature: 5 s at 260°C

Electrical

SWITCHING POWER MAX.: 240 mW DC
SWITCHING VOLTAGE MAX.: 24 V DC
SWITCHING CURRENT MAX.: 10 mA DC
SPACING CURRENT AT 20°C (push-push version): 100 mA
DIELECTRIC STRENGTH (50 Hz, 1 min): 500 V
OPERATING LIFE with or without max. switching power
Momentary: ≥ 5 x 10^6 operations
Push-push: ≥ 5 x 10^5 operations
CONTACT RESISTANCE: Initial ≤ 50 mΩ
After 5 x 10^6 operations: ≤ 100 mΩ
INSULATION RESISTANCE: ≥ 10^{10} Ω
CAPACITANCE at f= 10 kHz: ≤ 0.8pF
BOUNCE TIME: ≤ 2.5 ms
Operating speed 400 mm/s (15.75/s)

Environmental

OPERATING TEMPERATURE: -25°C to 85°C.



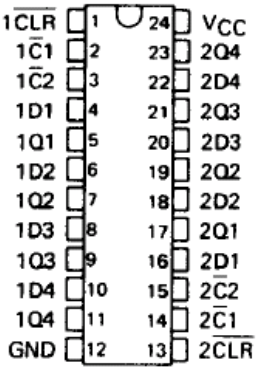
IC-Schaltung Dual 4-Bit Latch SN74116

SN54116, SN74116
DUAL 4-BIT LATCHES WITH CLEAR

DECEMBER 1972 – REVISED MARCH 1988

- Two Independent 4-Bit Latches in a Single Package
- Separate Clear Inputs Provide One-Step Clearing Operation
- Dual Gated Enable Inputs Simplify Cascading Register Implementations
- Compatible for Use with TTL Circuits
- Input Clamping Diodes Simplify System Design

SN54116 . . . J OR W PACKAGE
SN74116 . . . N PACKAGE
(TOP VIEW)



description

These monolithic TTL circuits utlize D-type bistables to implement two independent four-bit latches in a single package. Each four-bit latch has an independent asynchronous clear input and a gated two-input enable circuit. When both enable inputs are low, the output levels will follow the data input levels. When either or both of the enable inputs are taken high, the outputs remain at the last levels setup at the inputs prior to the low-to-high-level transition at the enable input(s). After this, the data inputs are locked out.

The clear input is overriding and when taken low will reset all four outputs low regardless of the levels of the enable inputs.

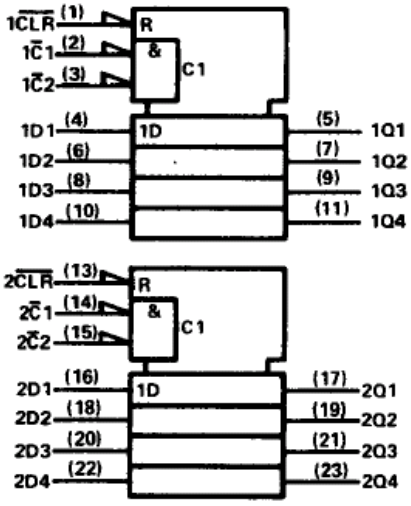
The SN54116 is characterized for operation over the full military temperature range of −55°C to 125°C; the SN74116 is characterized for operation from 0°C to 70°C.

FUNCTION TABLE
(EACH LATCH)

INPUTS				OUTPUT
CLEAR	ENABLE $\overline{C1}$	ENABLE $\overline{C2}$	DATA	Q
H	L	L	L	L
H	L	L	H	H
H	X	H	X	Q_0
H	H	X	X	Q_0
L	X	X	X	L

H = high level, L = low level, X = irrelevant
Q_0 = the level of Q before these input conditions were established.

logic symbol†

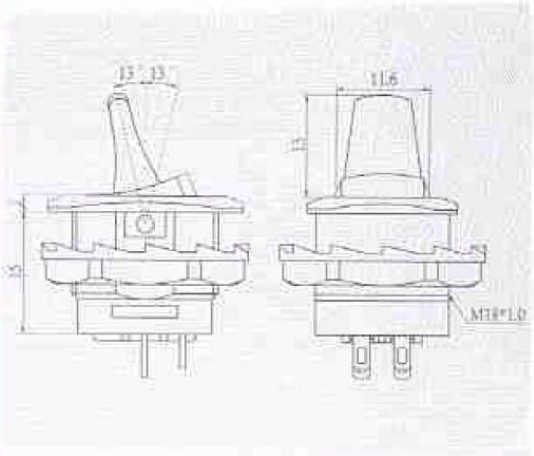


†This symbol is in accordance with ANSI/IEEE Std. 91-1984 and IEC Publication 617-12.

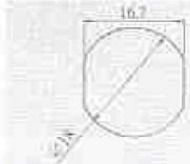
Kippschalter R13-405AL

R13-405





Rating
10A 125V AC
6A 250V AC
Contact resistance
50mΩ Max
Insulation resistance
DC 500V 100MΩ Min
Dielectric strength
AC 1500V 1 minute
Cut-out



5 mm max

FEATURES

* MATERIAL: Frames are nylon (UL flame: 94V-2). Terminals are silver plated brass.
* LED indicator on the top of lever. (AL , CL, DL type only). LED available with red, amber, and green colors.

Axiallüfter 12V 40x40

WALLAIR

Niedervolt-Axiallüfter



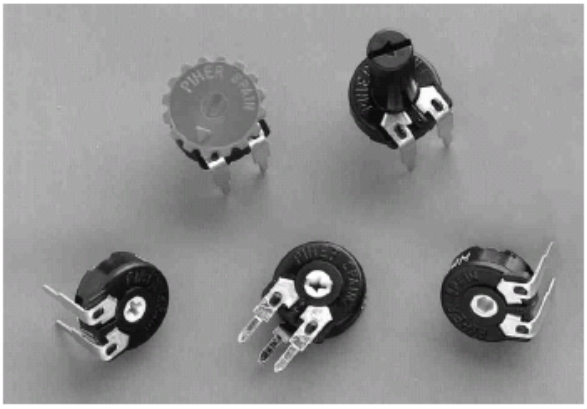
Universell einsetzbar.
Für Intervall- und Dauerbetrieb in elektronischen Geräten sowie im Sanitär- und Klimabereich.
Einfacher und wirksamer Schutz für PC´s vor Überhitzung.

CE

Maße	Sp.	Upm	m^3/h	A	dBA	VE	Typen-Bezeich-nung	Art.-Nr.	
25*25*10	5	7700	2,98	0,14	24,5	10	0502PFS2	20100300	
25*25*10	12	10500	3,00	0,095	26,4	10	0502PFB2	20100301	
40*40*20	12	6000	11,04	0,07	25,5	10	1204PKS2	20100302	
40*40*20	5	6000	11,04	0,07	25,5	10	0504PKS2	20100303	
60*60*25	12	4500	36,54	0,19	34,0	10	1206PTS1	20100304	
80*80*25	12	3000	63,00	0,22	33,0	10	1208PTS2	20100305	
92*92*25	12	2400	71,40	0,12	25,5	10	1209PTS3	20100306	
120*120*25	12	2400	117,00	0,23	42,3	10	1212PTS4	20100307	

Einstellpotentiometer PIHER 10-S 5,0K

PIHER



PT-10 10 mm Carbon Potentiometer

FEATURES

- Carbon resistive element
- Dust proof enclosure
- Polyester substrate
- Also upon request:
  - Wiper positioned at 50% or fully clockwise.
  - Supplied in magazines for automatic insertion.
  - Long life model for low cost control potentiometer applications
  - Self extinguishable plastic UL 94V-0
  - Cut track option
  - Special tapers
  - Mechanical detents
  - Low & extra low torque versions
  - Special switch option

MECHANICAL SPECIFICATIONS

– Mechanical rotation angle:	235° ± 5°
– Electrical rotation angle:	220° ± 20°
– Torque:	0.4 to 2 Ncm. (0.6 to 2.7 in-oz)
– Stop torque:	> 5 Ncm. (>7 in-oz)

ELECTRICAL SPECIFICATIONS

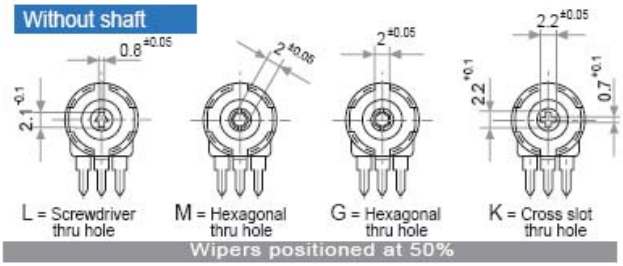
- Range of values (*)
  100Ω ≤ Rn ≤ 5 M (Decad. 1.0 - 2.0 - 2.2 - 2.5 - 4.7 - 5.0)
- Tolerance (*): 100Ω ≤ Rn ≤ 1M Ω ------ ± 20%
  1MΩ < Rn ≤ 5M Ω ------ ± 30%
- Max. Voltage: 200 VDC (lin) 100 VDC (no lin)
- Nominal Power 50°C (122°F) (see power rating curve)
  0.15 W (lin) 0.07 W (no lin)
- Taper (*) (Log. & Alog. only Rn ≥ 1K) Lin ; Log; Alog.
- Residual resistance: ≤ 5.10 $^{-3}$ Rn (2 Ω min.)
- Equivalent Noise Resistance: ≤ 3% Rn (3 Ω min.)
- Operating temperature**: -25°C + 70°C (-13°F + 158°F)

(*) Others upon request

** Up to 85°C depending on application

ROTORS

Without shaft

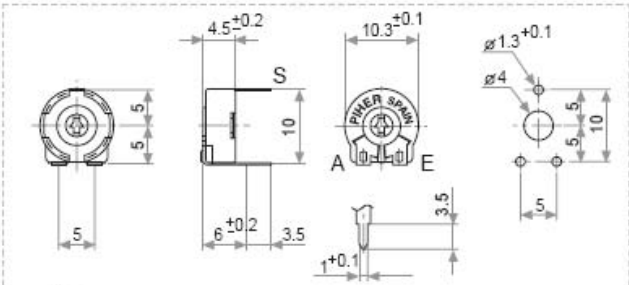


With shaft

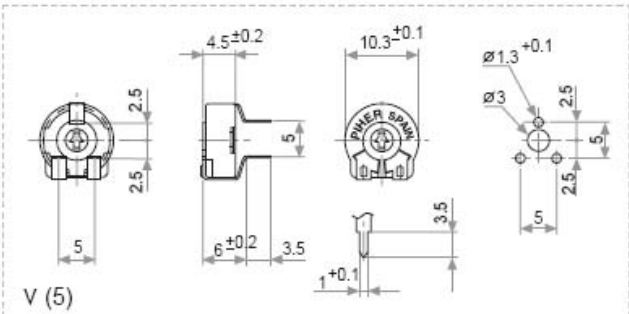


MOUNTING METHODS

V = horizontal mount – vertical adjust

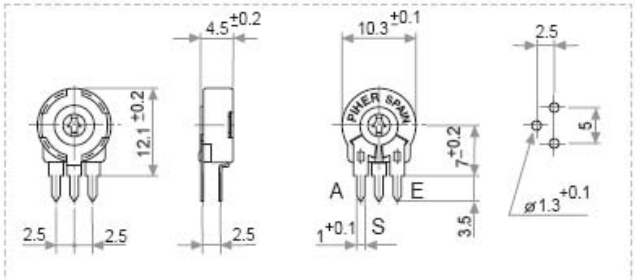


V (10) A = Initial S = Wiper E = Final

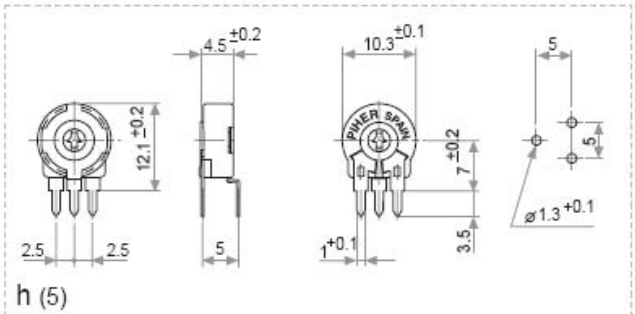


V (5)

h = vertical mount – horizontal adjust



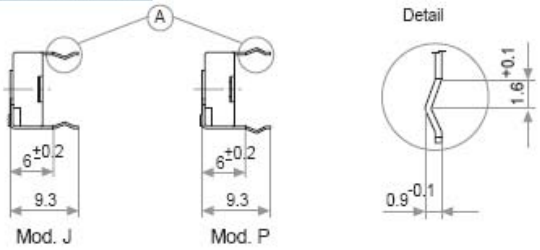
h (2.5) A = Initial S = Wiper E = Final



h (5)

NOTE = Please note relative terminal positions when ordering non linear tapers.

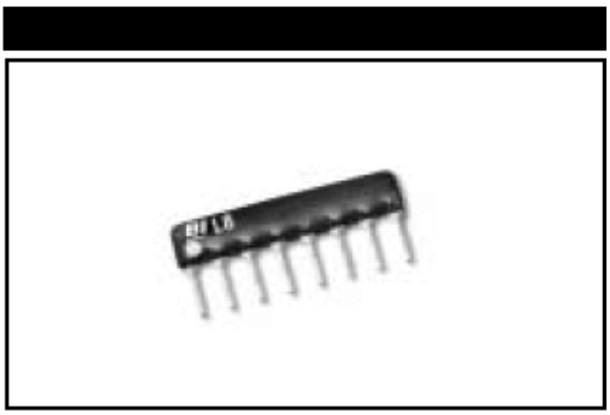
Crimped terminals



Widerstandsnetzwerk 4Wid./8Pins

MODEL L SERIES

Thick Film
Low Profile SIP
Conformal Coated
Resistor Networks



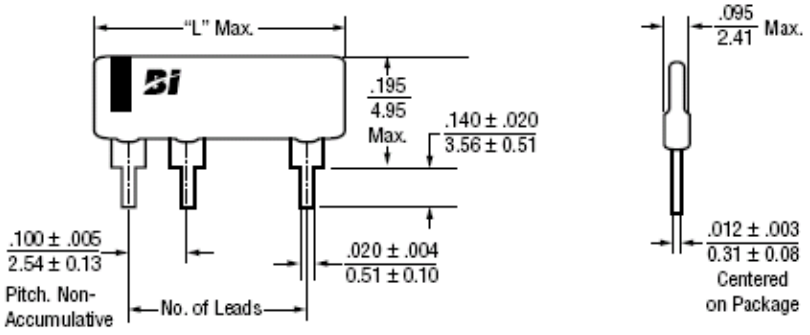
ELECTRICAL

Standard Resistance Range, Ohms	22 to 1Meg
Standard Resistance Tolerance, at 25°C	±2%
	Optional: ±1% (F Tol.), ±5% (J Tol.)
Operating Temperature Range	-55°C to +125°C
Temperature Coefficient of Resistance	±100ppm/°C (<100 Ohms = ±250ppm/°C)
Temperature Coefficient of Resistance, Tracking	±50ppm/°C
Maximum Operating Voltage	100Vdc or √PR
Insulation Resistance	≥10,000 Megohms

MECHANICAL

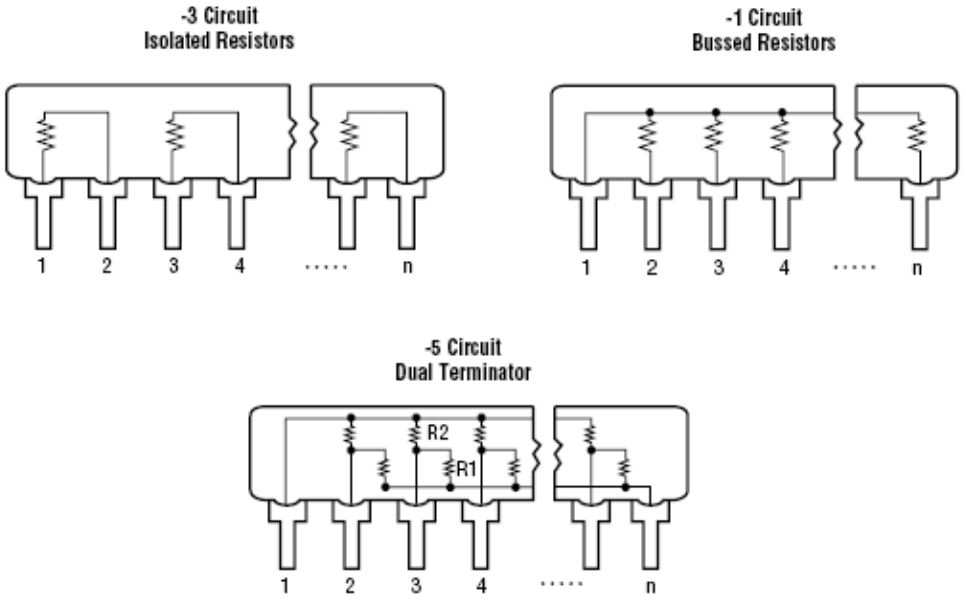
Lead Material	Steel Alloy (Standard) Copper Alloy (Optional)
Lead Finish	90/10 Tin-Lead
Substrate Material	Alumina
Resistor Material	Cermet
Body Material	Conformal Epoxy Resin

OUTLINE DIMENSIONS (Inch/mm)



# of Leads	4	5	6	7	8	9	10	11	12	13	14
Dim. "L" Max.	.40 / 10.16	.50 / 12.70	.60 / 15.24	.70 / 17.78	.80 / 20.32	.90 / 22.86	1.00 / 25.40	1.10 / 27.94	1.20 / 30.48	1.30 / 33.02	1.40 / 35.56

SCHEMATICS



Diode BAT 43

VISHAY

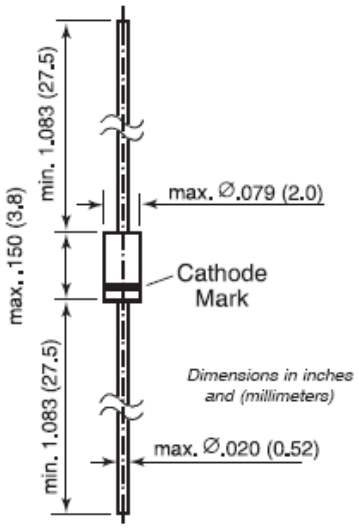
BAT42, BAT43

Vishay Semiconductors
formerly General Semiconductor

Schottky Diodes



DO-204AH
(DO-35 Glass)



Features

- For general purpose applications
- These diodes feature very low turn-on voltage and fast switching. These devices are protected by a PN junction guard ring against excessive voltage, such as electro-static discharges
- These diodes are also available in the SOD-123 case with the type designations BAT42W to BAT43W and in designations LL42 to LL43.

Mechanical Data

Case: DO-35 Glass Case
Weight: approx. 0.13g
Packaging Codes/Options:
D7/10K per 13" reel (52mm tape), 20K/box
D8/10K per Ammo tape (52mm tape), 20K/box

Maximum Ratings & Thermal Characteristics

Ratings at 25°C ambient temperature unless otherwise specified.

Parameter	Symbol	Value	Unit
Repetitive Peak Reverse Voltage	V_{RRM}	30	V
Forward Continuous Current at T_{amb} = 25°C	I_F	200(1)	mA
Repetitive Peak Forward Current at t_p < 1s, δ < 0.5, T_{amb} = 25°C	I_{FRM}	500(1)	mA
Surge Forward Current at t_p < 10ms, T_{amb} = 25°C	I_{FSM}	4(1)	A
Power Dissipation(1) at T_{amb} = 65°C	P_{tot}	200(1)	mW
Thermal Resistance Junction to Ambient Air	$R_{\theta JA}$	300(1)	°C/W
Junction Temperature	T_j	125	°C
Ambient Operating Temperature Range	T_{amb}	−65 to +125	°C
Storage Temperature Range	T_S	−65 to +150	°C

Electrical Characteristics

(T_J = 25°C unless otherwise noted)

Parameter		Symbol	Test Condition	Min	Typ	Max	Unit
Reverse Breakdown Voltage		$V_{(BR)R}$	I_R = 100µA (pulsed)	30	—	—	V
Leakage Current Pulse Test t_p < 300µs, δ < 2%		I_R	V_R = 25V V_R = 25V, T_j = 100°C	— —	— —	0.5 100	µA
Forward Voltage Pulse Test t_p < 300µs, δ < 2%	BAT42, 43 BAT42 BAT43 BAT43 BAT43	V_F	I_F = 200mA I_F = 10mA I_F = 50mA I_F = 2mA I_F = 15mA	— — — 0.26 —	— — — — —	1 0.4 0.65 0.33 0.45	V
Capacitance		C_{tot}	V_R = 1V, f = 1MHz	—	7	—	pF
Reverse Recovery Time		t_{rr}	I_F = 10mA, I_R = 10mA I_{rr} = 1mA, R_L = 100Ω	—	—	5	ns
Detection Efficiency		η_v	R_L = 15KΩ, C_L = 300pF f = 45MHz, V_{RF} = 2V	80	—	—	%

Note: (1) Valid provided that leads at a distance of 4mm from case are kept at ambient temperature

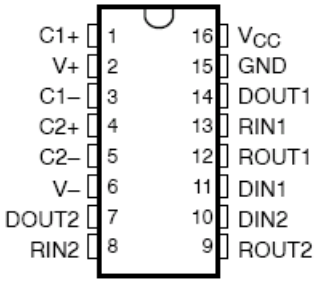
IC-Schaltung MAX 3232 CSE

MAX3232
3-V TO 5.5-V MULTICHANNEL RS-232 LINE DRIVER/RECEIVER

SLLS410F – JANUARY 2000 – REVISED AUGUST 2002

- **Meets or Exceeds the Requirements of TIA/EIA-232-F and ITU v.28 Standards**
- **Operates With 3-V to 5.5-V V_{CC} Supply**
- **Operates up to 250 kbit/s**
- **Two Drivers and Two Receivers**
- **Low Supply Current . . . 300 μA Typical**
- **External Capacitors . . . 4 × 0.1 μF**
- **Accepts 5-V Logic Input With 3.3-V Supply**
- **Designed to Be Interchangeable With Maxim MAX3232**
- **RS-232 Bus-Pin ESD Protection Exceeds ±15 kV Using Human-Body Model (HBM)**
- **Applications**
  - **Battery-Powered Systems, PDAs, Notebooks, Laptops, Palmtop PCs, and Hand-Held Equipment**

D, DB, DW, OR PW PACKAGE
(TOP VIEW)



description/ordering information

The MAX3232 device consists of two line drivers, two line receivers, and a dual charge-pump circuit with ±15-kV ESD protection pin to pin (serial-port connection pins, including GND). The device meets the requirements of TIA/EIA-232-F and provides the electrical interface between an asynchronous communication controller and the serial-port connector. The charge pump and four small external capacitors allow operation from a single 3-V to 5.5-V supply. The devices operate at data signaling rates up to 250 kbit/s and a maximum of 30-V/μs driver output slew rate.

Function Tables

EACH DRIVER

INPUT DIN	OUTPUT DOUT
L	H
H	L

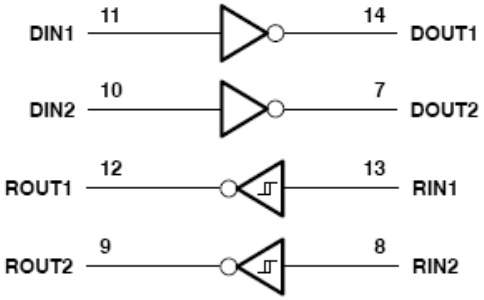
H = high level, L = low level

EACH RECEIVER

INPUT RIN	OUTPUT ROUT
L	H
H	L
Open	H

H = high level, L = low level, Open = input disconnected or connected driver off

logic diagram (positive logic)



IC-Schaltung 74HCT244

SGS-THOMSON MICROELECTRONICS

M54HCT240/241/244
M74HCT240/241/244

OCTAL BUS BUFFER WITH 3 STATE OUTPUTS
HCT240: INVERTED - HCT241/244 NON INVERTED

- HIGH SPEED
  t_{PD} = 13 ns (TYP.) at V_{CC} = 5V
- LOW POWER DISSIPATION
  I_{CC} = 4 µA (MAX.) at T_A = 25 °C
- COMPATIBLE WITH TTL OUTPUTS
  V_{IH} = 2V (MIN.) V_{IL} = 0.8V (MAX)
- OUTPUT DRIVE CAPABILITY
  15 LSTTL LOADS
- SYMMETRICAL OUTPUT IMPEDANCE
  $|I_{OH}| = I_{OL}$ = 6 mA (MIN)
- BALANCED PROPAGATION DELAYS
  $t_{PLH} = t_{PHL}$
- PIN AND FUNCTION COMPATIBLE
  WITH 54/74LS240/241/244



B1R
(Plastic Package)



F1R
(Ceramic Package)



M1R
(Micro Package)



C1R
(Chip Carrier)

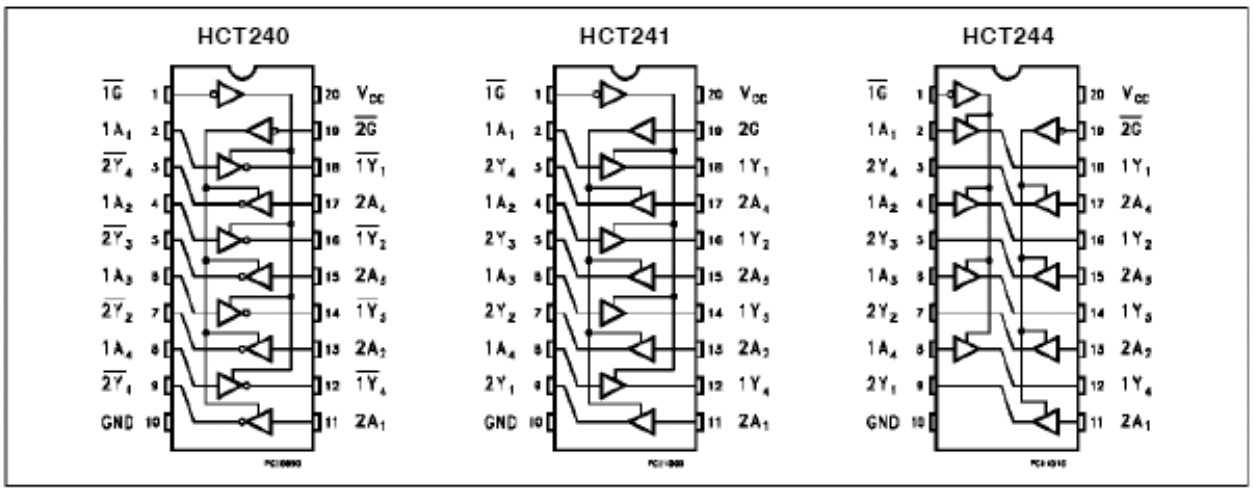
ORDER CODES :

M54HCTXXXF1R	M74HCTXXXM1R
M74HCTXXXB1R	M74HCTXXXC1R

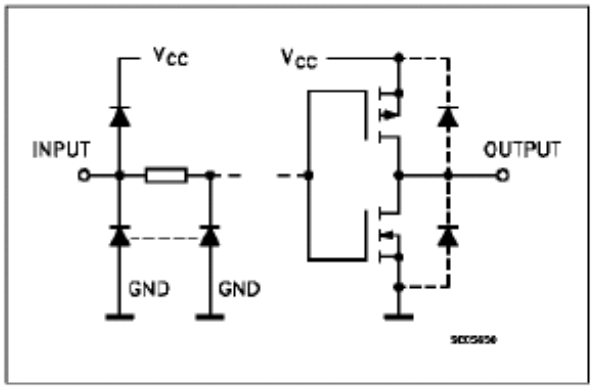
DESCRIPTION

The M54/74HCT240, HCT241 and HCT244 are high speed CMOS OCTAL BUS BUFFERs fabricated in silicon gate C^2MOS technology.
They have the same high speed performance of LSTTL combined with true CMOS low power consumption. The designer has a choise of select combination of inverting and non-inverting outputs, symmetrical $\overline{G}$ (active low output control) input, and complementary G and $\overline{G}$ inputs. Each control input governs four BUS BUFFERs.
This integrated circuit has input and output characteristics that are fully compatible with 54/74 LSTTL logic families. M54/74HCT devices are designed to directly interface HSC^2MOS systems with TTL and NMOS components.
They are also plug in replacements for LSTTL devices giving a reduction of power consumption. These devices are designed to be used with 3 state memory address drivers, etc.
All inputs are equipped with protection circuits against static discharge and transient excess voltage.

PIN CONNECTION (top view)



INPUT AND OUTPUT EQUIVALENT CIRCUIT



PIN DESCRIPTION (HCT240)

PIN No	SYMBOL	NAME AND FUNCTION
1	$\overline{1G}$	Output Enable Input
2, 4, 6, 8	1A1 to 1A4	Data Inputs
9, 7, 5, 3	2Y1 to 2Y4	Data Outputs
11, 13, 15, 17	2A1 to 2A4	Data Inputs
18, 16, 14, 12	1Y1 to 1Y4	Data Outputs
19	$\overline{2G}$	Output Enabel Input
10	GND	Ground (0V)
20	Vcc	Positive Supply Voltage

PIN DESCRIPTION (HCT241)

PIN No	SYMBOL	NAME AND FUNCTION
1	$\overline{1G}$	Output Enable Input
2, 4, 6, 8	1A1 to 1A4	Data Inputs
9, 7, 5, 3	2Y1 to 2Y4	Data Outputs
11, 13, 15, 17	2A1 to 2A4	Data Inputs
18, 16, 14, 12	1Y1 to 1Y4	Data Outputs
19	2G	Output Enabel Input
10	GND	Ground (0V)
20	Vcc	Positive Supply Voltage

PIN DESCRIPTION (HCT244)

PIN No	SYMBOL	NAME AND FUNCTION
1	$\overline{1G}$	Output Enable Input
2, 4, 6, 8	1A1 to 1A4	Data Inputs
9, 7, 5, 3	$\overline{2Y1}$ to $\overline{2Y4}$	Data Outputs
11, 13, 15, 17	2A1 to 2A4	Data Inputs
18, 16, 14, 12	$\overline{1Y1}$ to $\overline{1Y4}$	Data Outputs
19	$\overline{2G}$	Output Enabel Input
10	GND	Ground (0V)
20	Vcc	Positive Supply Voltage

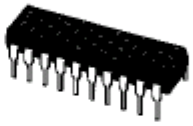
IC-Schaltung 74HCT245

SGS-THOMSON MICROELECTRONICS M54/74HCT245/640/643

M54/74HCT245/640/643

OCTAL BUS TRANSCEIVER (3-STATE): HCT245 NON INVERTING HCT640 INVERTING, HCT643 INVERTING/NON INVERTING

- HIGH SPEED
  t_{PD} = 10 ns (TYP.) at V_{CC} = 5V
- LOW POWER DISSIPATION
  I_{CC} = 4 μA (MAX.) at T_A = 25 °C
- COMPATIBLE WITH TTL OUTPUTS
  V_{IH} = 2V (MIN.) V_{IL} = 0.8V (MAX.)
- OUTPUT DRIVE CAPABILITY
  15 LSTTL LOADS
- SYMMETRICAL OUTPUT IMPEDANCE
  $|I_{OH}| = I_{OL}$ = 6 mA (MIN)
- BALANCED PROPAGATION DELAYS
  $t_{PLH} = t_{PHL}$
- PIN AND FUNCTION COMPATIBLE
  WITH 54/74LS245/640/643



B1R
(Plastic Package)



F1R
(Ceramic Package)



M1R
(Micro Package)



C1R
(Chip Carrier)

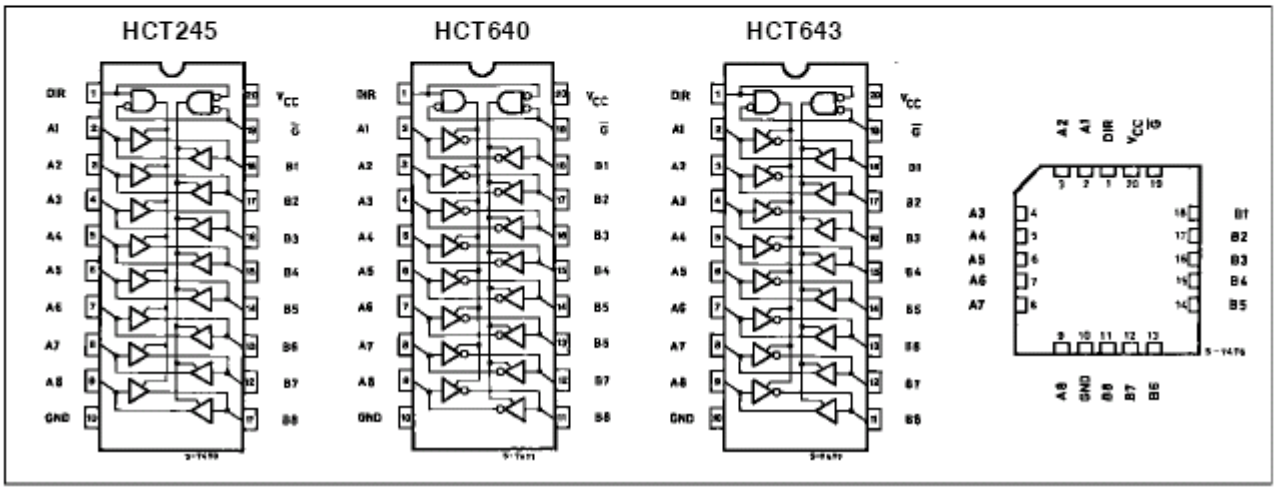
ORDER CODES :

M54HCTXXXF1R M74HCTXXXM1R
M74HCTXXXB1R M74HCTXXXC1R

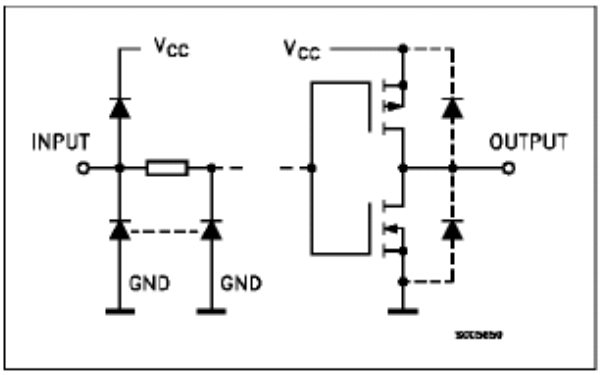
DESCRIPTION

The M54/74HCT245, HCT640 and HCT643 utilise silicon gate C^2MOS technology to achive operating speeds eqivalent to LSTTL devices.
Along with the low power dissipation and high noise immunity of standad CMOS integrated circuit, it possesses the capability to drive 15 LSTTL loads. These IC's are intended for two-way asynchronous communication between data buses, and the direction of data trasmission is determined by DIR input. The enable input ($\overline{G}$) can be used to disable the device so that the buses are effectively isolated. All input are equipped with protection circuits against static discharge and transient discharge. These integrated circuits have input and output characteristics that are fully compatible with 54/74 LSTTL logic families. M54/74HCT devices are designed to directly interface HSC^2MOS systems with TTL and NMOS components. They are also plug in replacements for LSTTL devices giving a reduction of power consumption.
IT IS PROHIBITED TO APPLY A SIGNAL TO A BUS TERMINAL WHEN IT IS IN OUTPUT MODE AND WHEN A BUS THERMINAL IS FLOATING (HIGH IMPEDANCE STATE), IT IS REQUESTED TO FIX THE INPUT LEVEL BY MEANS OF EXTERNAL PULL DOWN OR PULL UP RESISTOR.

PIN CONNECTION (top view)



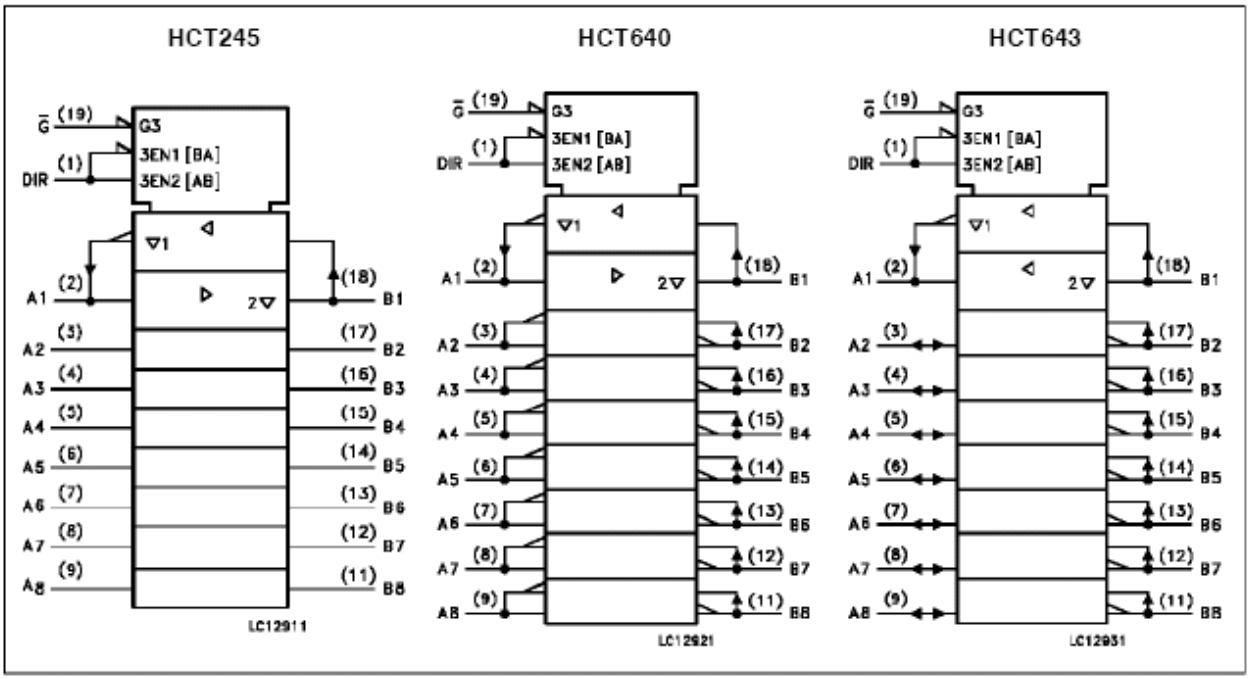
INPUT AND OUTPUT EQUIVALENT CIRCUIT



PIN DESCRIPTION

PIN No	SYMBOL	NAME AND FUNCTION
1	DIR	Directional Control
2, 3, 4, 5, 6, 7, 8, 9	A1 to A8	Data Inputs/Outputs
18, 17, 16, 15, 14, 13, 12, 11	B1 to B8	Data Inputs/Outputs
19	$\overline{G}$	Output Enabel Input (Active LOW)
10	GND	Ground (0V)
20	Vcc	Positive Supply Voltage

IEC LOGIC SYMBOLS



TRUTH TABLE

INPUT		FUNCTION		OUTPUT		
$\overline{G}$	DIR	A BUS	B BUS	HCT245	HCT640	HCT643
L	L	OUTPUT	INPUT	A = B	A = $\overline{B}$	A = B
L	H	INPUT	OUTPUT	B = A	B = $\overline{A}$	B = $\overline{A}$
H	X	Z	Z	Z	Z	Z

X: "H" or "L"
Z: High impedance

IC-Schaltung 74HCT04

SGS-THOMSON MICROELECTRONICS

M54HCT04
M74HCT04

HEX INVERTER

- HIGH SPEED
  t_{PD} = 8 ns (TYP.) AT V_{CC} = 5 V
- LOW POWER DISSIPATION
  I_{CC} = 1 µA (MAX.) AT T_A = 25 °C
- COMPATIBLE WITH TTL OUTPUTS
  V_{IH} = 2V (MIN.) V_{IL} = 0.8V (MAX)
- OUTPUT DRIVE CAPABILITY
  10 LSTTL LOADS
- SYMMETRICAL OUTPUT IMPEDANCE
  $|I_{OH}| = I_{OL}$ = 4 mA (MIN.)
- BALANCED PROPAGATION DELAYS
  $t_{PLH} = t_{PHL}$
- PIN AND FUNCTION COMPATIBLE
  WITH 54/74LS04



B1R
(Plastic Package)



F1R
(Ceramic Package)



M1R
(Micro Package)



C1R
(Chip Carrier)

ORDER CODES :

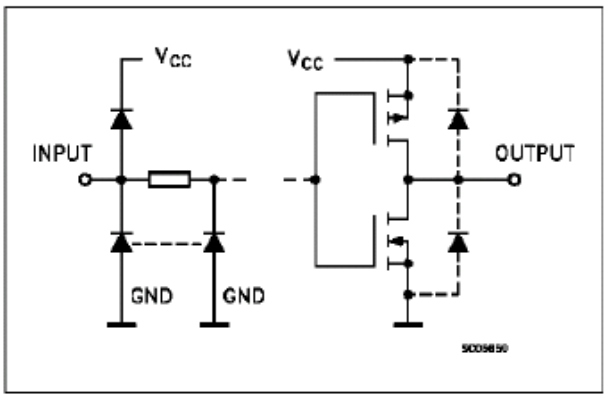
M54HCT04F1R M74HCT04M1R
M74HCT04B1R M74HCT04C1R

DESCRIPTION

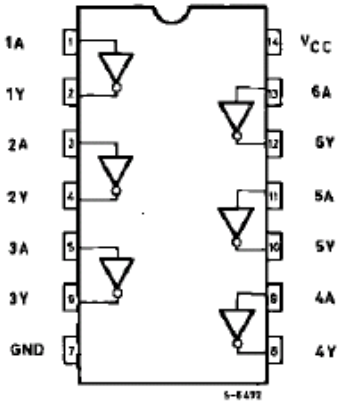
The M54/74HCT04 is a high speed CMOS INVERTER fabricated in silicon gate C^2MOS technology. It has the same high speed performance of LSTTL combined with true CMOS low power consumption. The internal circuit is composed of 3 stages including buffered output, which gives high noise immunity and a stable output.

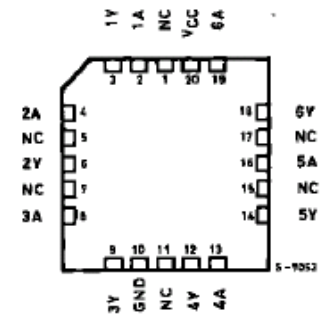
All inputs are equipped with protection circuits against static discharge or transient excess voltage. This integrated circuit has input and output characteristic that are fully compatible with 54/74 LSTTL logic families. M54HCT/74HCT devices are designed to directly interface HSC^2MOS systems with TTL and NMOS components. They are also plug in replacements for LSTTL devices giving a reduction of power consumption.

INPUT AND OUTPUT EQUIVALENT CIRCUIT



PIN CONNECTIONS (top view)



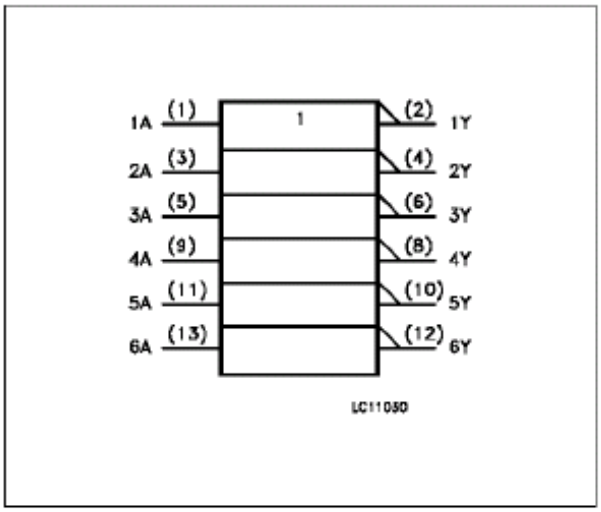


NC =
No Internal
Connection

TRUTH TABLE

A	Y
L	H
H	L

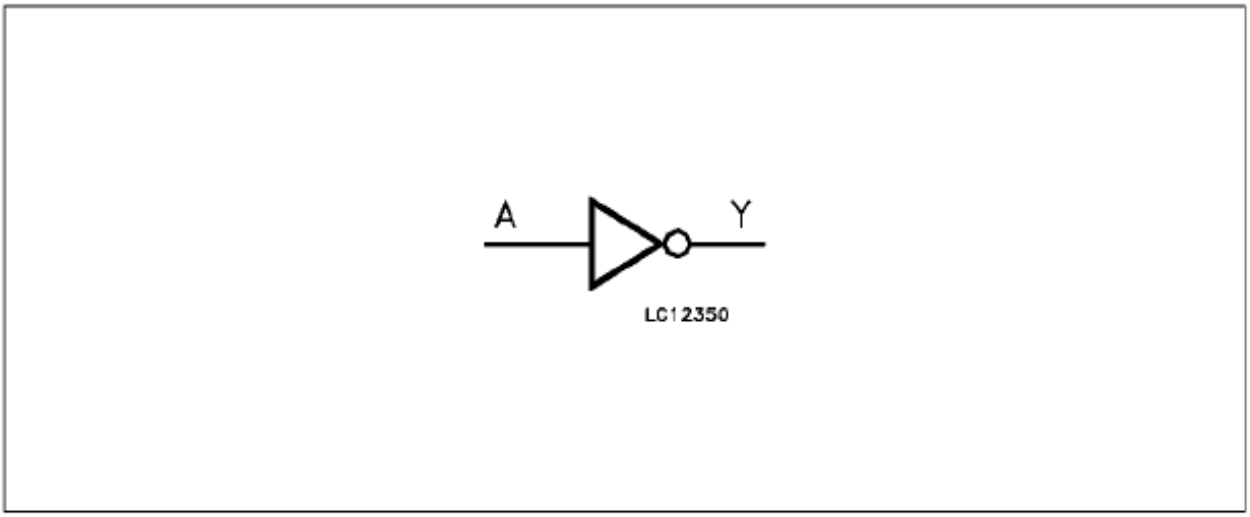
IEC LOGIC SYMBOL



PIN DESCRIPTION

PIN No	SYMBOL	NAME AND FUNCTION
1, 3, 5, 9, 11, 13	1A to 6A	Data Inputs
2, 4, 6, 8, 10, 12	1Y to 6Y	Data Outputs
7	GND	Ground (0V)
14	V_{CC}	Positive Supply Voltage

LOGIC DIAGRAM (Per Gate)



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CC}	Supply Voltage	-0.5 to +7	V
V_I	DC Input Voltage	-0.5 to V_{CC} + 0.5	V
V_O	DC Output Voltage	-0.5 to V_{CC} + 0.5	V
I_{IK}	DC Input Diode Current	± 20	mA
I_{OK}	DC Output Diode Current	± 20	mA
I_O	DC Output Source Sink Current Per Output Pin	± 25	mA
I_{CC} or I_{GND}	DC V_{CC} or Ground Current	± 50	mA
P_D	Power Dissipation	500 (*)	mW
T_{stg}	Storage Temperature	-65 to +150	°C
T_L	Lead Temperature (10 sec)	300	°C

Absolute Maximum Ratings are those values beyond which damage to the device may occur. Functional operation under these condition is not implied.
(*) 500 mW: ≙ 65 °C derate to 300 mW by 10mW/°C: 65 °C to 85 °C

IC-Schaltung 74HCT08

SGS-THOMSON MICROELECTRONICS

M54HCT08
M74HCT08

QUAD 2-INPUT AND GATE

- HIGH SPEED
  t_{PD} = 12 ns (TYP.) AT V_{CC} = 5 V
- LOW POWER DISSIPATION
  I_{CC} = 1 µA (MAX.) AT T_A = 25 °C
- COMPATIBLE WITH TTL OUTPUTS
  V_{IH} = 2V (MIN.) V_{IL} = 0.8V (MAX)
- OUTPUT DRIVE CAPABILITY
  10 LSTTL LOADS
- SYMMETRICAL OUTPUT IMPEDANCE
  $|I_{OH}| = I_{OL}$ = 4 mA (MIN.)
- BALANCED PROPAGATION DELAYS
  $t_{PLH} = t_{PHL}$
- PIN AND FUNCTION COMPATIBLE
  WITH 54/74LS08



B1R
(Plastic Package)



F1R
(Ceramic Package)



M1R
(Micro Package)



C1R
(Chip Carrier)

ORDER CODES :

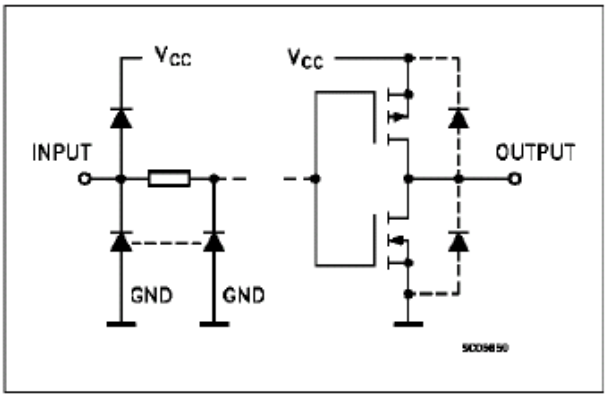
M54HCT08F1R	M74HCT08M1R
M74HCT08B1R	M74HCT08C1R

DESCRIPTION

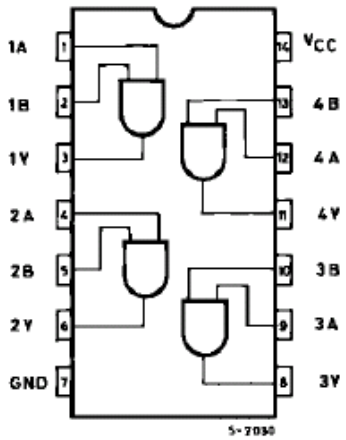
The M54/74HCT08 is a high speed CMOS QUAD 2-INPUT AND GATE fabricated in silicon gate C^2MOS technology. It has the same high speed performance of LSTTL combined with true CMOS low power consumption. The internal circuit is composed of 2 stages including buffer output, which gives high noise immunity and stable output. All inputs are equipped with protection circuits against static discharge and transient excess voltage.

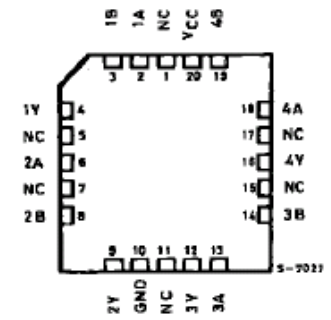
This integrated circuit has input and output characteristics that are fully compatible with 54/74 LSTTL logic families. M54/74HC devices are designed to directly interface HSC^2MOS systems with TTL and NMOS components. They are also plug in replacements for LSTTL devices giving a reduction of power consumption.

INPUT AND OUTPUT EQUIVALENT CIRCUIT



PIN CONNECTIONS (top view)



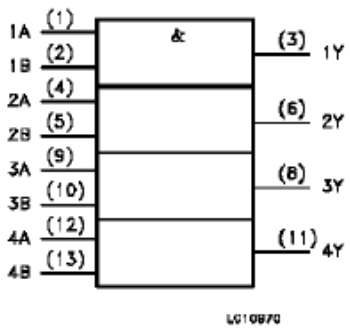


NC =
No Internal
Connection

TRUTH TABLE

A	B	Y
L	L	L
L	H	L
H	L	L
H	H	H

IEC LOGIC SYMBOL



PIN DESCRIPTION

PIN No	SYMBOL	NAME AND FUNCTION
1, 4, 9, 12	1A to 4A	Data Inputs
2, 5, 10, 13	1B to 4B	Data Inputs
3, 6, 8, 11	1Y to 4Y	Data Outputs
7	GND	Ground (0V)
14	V_{CC}	Positive Supply Voltage

SCHEMATIC CIRCUIT (Per Gate)



ABSOLUTE MAXIMUM RATING

Symbol	Parameter	Value	Unit
V_{CC}	Supply Voltage	-0.5 to +7	V
V_I	DC Input Voltage	-0.5 to V_{CC} + 0.5	V
V_O	DC Output Voltage	-0.5 to V_{CC} + 0.5	V
I_{IK}	DC Input Diode Current	± 20	mA
I_{OK}	DC Output Diode Current	± 20	mA
I_O	DC Output Source Sink Current Per Output Pin	± 25	mA
I_{CC} or I_{GND}	DC V_{CC} or Ground Current	± 50	mA
P_D	Power Dissipation	500 (*)	mW
T_{stg}	Storage Temperature	-65 to +150	°C
T_L	Lead Temperature (10 sec)	300	°C

Absolute Maximum Ratings are those values beyond which damage to the device may occur. Functional operation under these condition is not implied.
(*) 500 mW: ≅ 65 °C derate to 300 mW by 10mW/°C: 65 °C to 85 °C

IC-Schaltung 74HCT86

SGS-THOMSON MICROELECTRONICS

M54HCT86
M74HCT86

QUAD EXCLUSIVE OR GATE

- HIGH SPEED
  t_{PD} = 15 ns (TYP.) AT V_{CC} = 5 V
- LOW POWER DISSIPATION
  I_{CC} = 1 µA (MAX.) AT T_A = 25 °C
- COMPATIBLE WITH TTL OUTPUTS
  V_{IH} = 2V (MIN.) V_{IL} = 0.8V (MAX)
- OUTPUT DRIVE CAPABILITY
  10 LSTTL LOADS
- SYMMETRICAL OUTPUT IMPEDANCE
  $|I_{OH}| = I_{OL}$ = 4 mA (MIN.)
- BALANCED PROPAGATION DELAYS
  $t_{PLH} = t_{PHL}$
- PIN AND FUNCTION COMPATIBLE
  WITH 54/74LS86



B1R
(Plastic Package)



F1R
(Ceramic Package)



M1R
(Micro Package)



C1R
(Chip Carrier)

ORDER CODES :

M54HCT86F1R	M74HCT86M1R
M74HCT86B1R	M74HCT86C1R

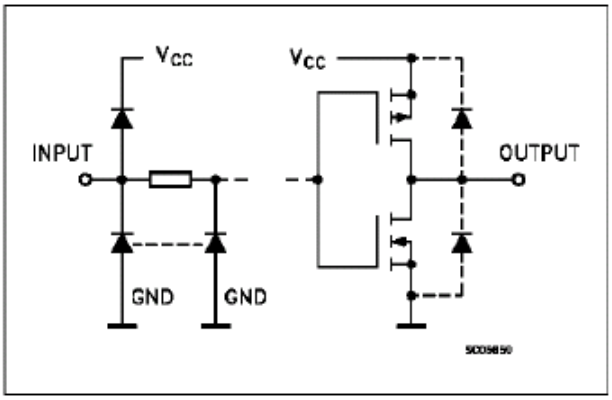
DESCRIPTION

The M54/74HCT86 is a high speed CMOS QUAD EXCLUSIVE OR GATE fabricated in silicon gate C^2MOS technology.

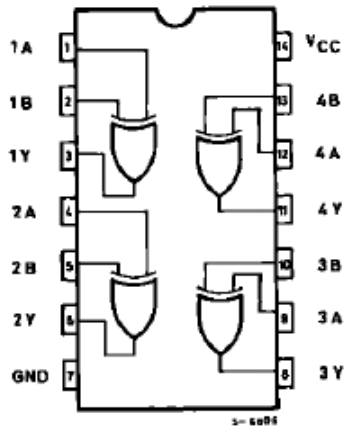
It has the same high speed performance of LSTTL combined with true CMOS low power consumption. Input and output buffer are installed, which enables high noise immunity and stable output.

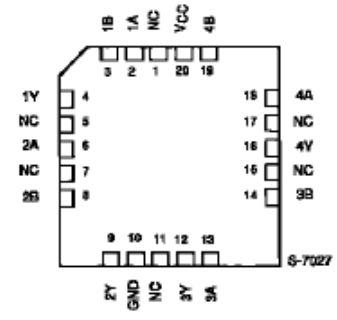
All inputs are equipped with protection circuits against static discharge and transient excess voltage.This integrated circuit has input and output characteristics that are fully compatible with 54/74 LSTTL logic families. M54/74HCT devices are designed to directly interface HSC^2MOS systems with TTL and NMOS components. They are also plug in replacements for LSTTL devices giving a reduction of power consumption.

INPUT AND OUTPUT EQUIVALENT CIRCUIT



PIN CONNECTIONS (top view)



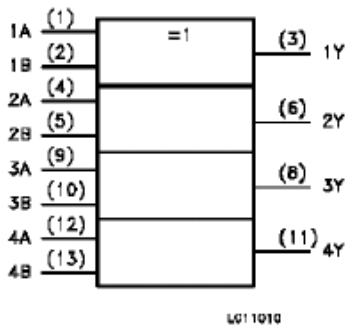


NC =
No Internal
Connection

TRUTH TABLE

A	B	Y
L	L	L
L	H	H
H	L	H
H	H	L

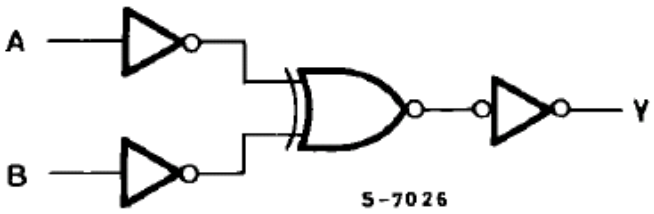
IEC LOGIC SYMBOL



PIN DESCRIPTION

PIN No	SYMBOL	NAME AND FUNCTION
1, 4, 9, 12	1A to 4A	Data Inputs
2, 5, 10, 13	1B to 4B	Data Inputs
3, 6, 8, 11	1Y to 4Y	Data Outputs
7	GND	Ground (0V)
14	V_{CC}	Positive Supply Voltage

SCHEMATIC CIRCUIT (Per Gate)



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CC}	Supply Voltage	-0.5 to +7	V
V_I	DC Input Voltage	-0.5 to V_{CC} + 0.5	V
V_O	DC Output Voltage	-0.5 to V_{CC} + 0.5	V
I_{IK}	DC Input Diode Current	± 20	mA
I_{OK}	DC Output Diode Current	± 20	mA
I_O	DC Output Source Sink Current Per Output Pin	± 25	mA
I_{CC} or I_{GND}	DC V_{CC} or Ground Current	± 50	mA
P_D	Power Dissipation	500 (*)	mW
T_{stg}	Storage Temperature	-65 to +150	°C
T_L	Lead Temperature (10 sec)	300	°C

Absolute Maximum Ratings are those values beyond which damage to the device may occur. Functional operation under these condition is not implied.
(*) 500 mW: ≅ 65 °C derate to 300 mW by 10mW/°C: 65 °C to 85 °C

IC-Schaltung 74HCT14

SGS-THOMSON MICROELECTRONICS

M54HCT14
M74HCT14

HEX SCHMITT INVERTER

- HIGH SPEED
  t_{PD} = 16 ns (TYP.) AT V_{CC} = 5 V
- LOW POWER DISSIPATION
  I_{CC} = 1 µA (MAX.) AT T_A = 25 °C
- HIGH NOISE IMMUNITY
  V_H = 0.7 V (TYP.) AT V_{CC} = 5 V
- OUTPUT DRIVE CAPABILITY
  10 LSTTL LOADS
- SYMMETRICAL OUTPUT IMPEDANCE
  $|I_{OH}| = I_{OL}$ = 4 mA (MIN.)
- BALANCED PROPAGATION DELAYS
  $t_{PLH} = t_{PHL}$
- PIN AND FUNCTION COMPATIBLE WITH 54/74LS14



B1R
(Plastic Package)



F1R
(Ceramic Package)



M1R
(Micro Package)



C1R
(Chip Carrier)

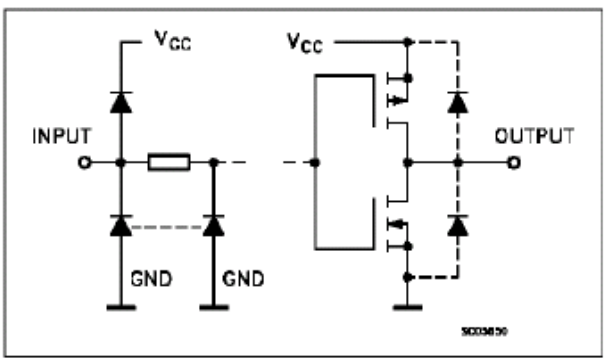
ORDER CODES :

M54HCT14F1R M74HCT14M1R
M74HCT14B1R M74HCT14C1R

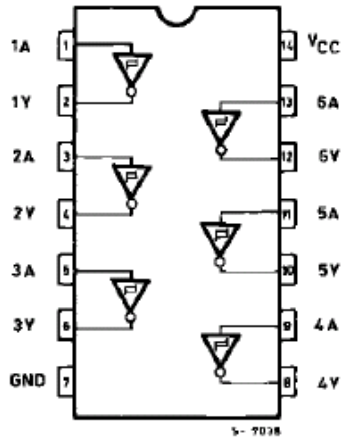
DESCRIPTION

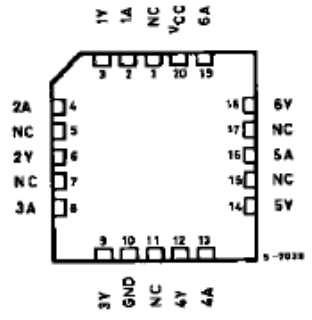
The M54/74HCT14 is a high speed CMOS HEX SCHMITT INVERTER fabricated in silicon gate C^2MOS technology. It has the same high speed performance of LSTTL combined with true CMOS low power consumption. Pin configuration and function are the same as those of the HCT04 but all inputs have 0.7 V hysteresis level.This together with its schmitt trigger function allows it to be used on line receivers with slow rise/fall input signals. This integrated circuit has input and output characteristics that are fully compatible with 54/74 LSTTL logic families. M54/74HCT devices are designed to directly interface HSC^2MOS systems with TTL and NMOS components. They are also plug in replacements for LSTTL devices giving a reduction of power consumption.All inputs are equipped with pro-

INPUT AND OUTPUT EQUIVALENT CIRCUIT



PIN CONNECTIONS (top view)



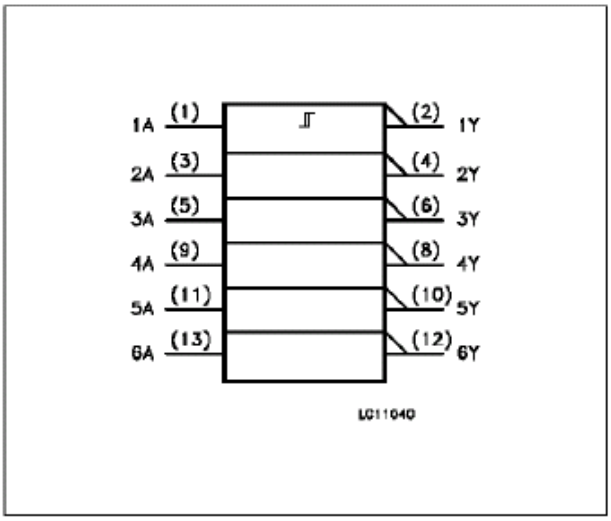


NC = No Internal Connection

TRUTH TABLE

A	Y
L	H
H	L

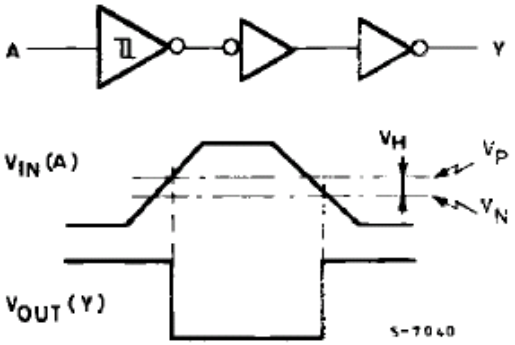
IEC LOGIC SYMBOL



PIN DESCRIPTION

PIN No	SYMBOL	NAME AND FUNCTION
1, 3, 5, 9, 11, 13	1A to 6A	Data Inputs
2, 4, 6, 8, 10,12	1Y to 6Y	Data Outputs
7	GND	Ground (0V)
14	V_{CC}	Positive Supply Voltage

LOGIC DIAGRAM/WAVEFORM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CC}	Supply Voltage	-0.5 to +7	V
V_I	DC Input Voltage	-0.5 to V_{CC} + 0.5	V
V_O	DC Output Voltage	-0.5 to V_{CC} + 0.5	V
I_{IK}	DC Input Diode Current	± 20	mA
I_{OK}	DC Output Diode Current	± 20	mA
I_O	DC Output Source Sink Current Per Output Pin	± 25	mA
I_{CC} or I_{GND}	DC V_{CC} or Ground Current	± 50	mA
P_D	Power Dissipation	500 (*)	mW
T_{stg}	Storage Temperature	-65 to +150	°C
T_L	Lead Temperature (10 sec)	300	°C

Absolute Maximum Ratings are those values beyond which damage to the device may occur. Functional operation under these condition is not implied.
(*) 500 mW: ≅ 65 °C derate to 300 mW by 10mW/°C: 65 °C to 85 °C

IC-Schaltung 74HCT11

Triple 3-input AND gate 74HC/HCT11

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
1, 3, 9	1A to 3A	data inputs
2, 4, 10	1B to 3B	data inputs
7	GND	ground (0 V)
12, 6, 8	1Y to 3Y	data outputs
13, 5, 11	1C to 3C	data inputs
14	V_{CC}	positive supply voltage

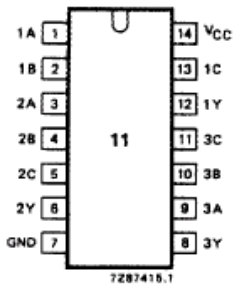


Fig.1 Pin configuration.

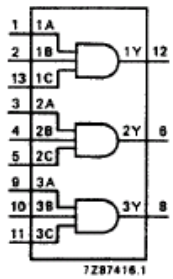


Fig.2 Logic symbol.

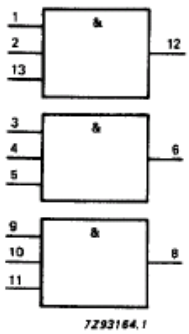


Fig.3 IEC logic symbol.

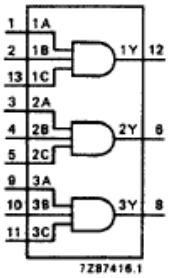


Fig.4 Functional diagram.

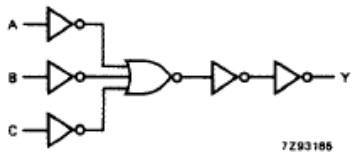


Fig.5 Logic diagram (one gate).

FUNCTION TABLE

INPUTS			OUTPUT
nA	nB	nC	nY
L	L	L	L
L	L	H	L
L	H	L	L
L	H	H	L
H	L	L	L
H	L	H	L
H	H	L	L
H	H	H	H

Notes

1. H = HIGH voltage level
   L = LOW voltage level

IC-Schaltung 74HCT132

TEXAS INSTRUMENTS

Data sheet acquired from Harris Semiconductor
SCHS145A

August 1997 - Revised May 2000

CD54/74HC132, CD54/74HCT132

High Speed CMOS Logic Quad 2-Input NAND Schmitt Trigger

Features

- Unlimited Input Rise and Fall Times
- Exceptionally High Noise Immunity
- Typical Propagation Delay: 10ns at V_{CC} = 5V, C_L = 15pF, T_A = 25°C
- Fanout (Over Temperature Range)
  - Standard Outputs . . . . . 10 LSTTL Loads
  - Bus Driver Outputs . . . . . 15 LSTTL Loads
- Wide Operating Temperature Range . . . -55°C to 125°C
- Balanced Propagation Delay and Transition Times
- Significant Power Reduction Compared to LSTTL Logic ICs
- HC Types
  - 2V to 6V Operation
  - High Noise Immunity: N_{IL} = 37%, N_{IH} = 51% of V_{CC} at V_{CC} = 5V
- HCT Types
  - 4.5V to 5.5V Operation
  - Direct LSTTL Input Logic Compatibility, V_{IL} = 0.8V (Max), V_{IH} = 2V (Min)
  - CMOS Input Compatibility, $I_I \leq 1\mu A$ at V_{OL}, V_{OH}

Description

The 'HC132 and 'HCT132 each contain four 2-input NAND Schmitt Triggers in one package. This logic device utilizes silicon gate CMOS technology to achieve operating speeds similar to LSTTL gates with the low power consumption of standard CMOS integrated circuits. All devices have the ability to drive 10 LSTTL loads. The HCT logic family is functionally pin compatible with the standard LS logic family.

Ordering Information

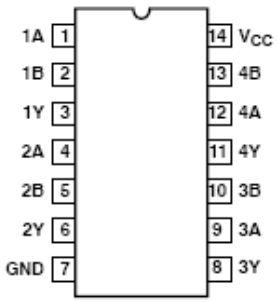
PART NUMBER	TEMP. RANGE (°C)	PACKAGE
CD54HC132F3A	-55 to 125	14 Ld CERDIP
CD74HC132E	-55 to 125	14 Ld PDIP
CD74HC132M	-55 to 125	14 Ld SOIC
CD54HCT132F	-55 to 125	14 Ld CERDIP
CD54HCT132F3A	-55 to 125	14 Ld CERDIP
CD74HCT132E	-55 to 125	14 Ld PDIP
CD74HCT132M	-55 to 125	14 Ld SOIC

NOTES:

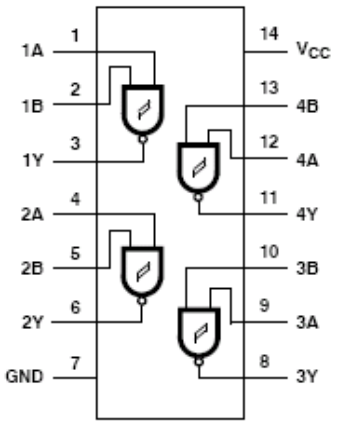
1. When ordering, use the entire part number. Add the suffix 96 to obtain the variant in the tape and reel.
2. Die for this part number is available which meets all electrical specifications. Please contact your local TI sales office or customer service for ordering information.

Pinout

CD54HC132, CD54HCT132
(CERDIP)
CD74HC132, CD74HCT132
(PDIP, SOIC)
TOP VIEW



Functional Diagram

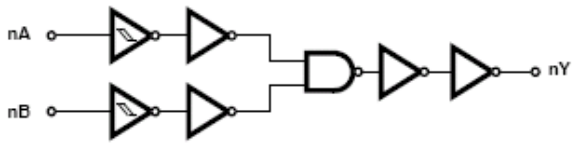


TRUTH TABLE

INPUTS		OUTPUT
nA	nB	nY
L	L	H
L	H	H
H	L	H
H	H	L

NOTE: H = High Voltage Level, L = Low Voltage Level

Logic Symbol



3mm LED Standard

Kingbright®

T-1 (3mm) SOLID STATE LAMPS

L-934H BRIGHT RED
L-934I HIGH EFFICIENCY RED
L-934N PURE ORANGE
L-934PG PURE GREEN
L-934E ORANGE
L-934G GREEN
L-934Y YELLOW

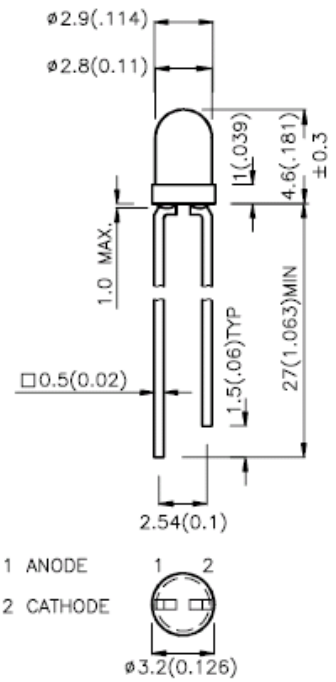
Features

- HIGH INTENSITY.
- LOW POWER CONSUMPTION.
- POPULAR T-1 DIAMETER PACKAGE.
- GENERAL PURPOSE LEADS.
- RELIABLE AND RUGGED.
- AVAILABLE ON TAPE AND REEL.
- DIFFUSED, TRANSPARENT AND WATER CLEAR TYPE.

Description

The Bright Red source color devices are made with Gallium Phosphide Red Light Emitting Diode.
The Green source color devices are made with Gallium Phosphide Green Light Emitting Diode.
The High Efficiency Red and Orange source color devices are made with Gallium Arsenide Phosphide on Gallium Phosphide Orange Light Emitting Diode.
The Yellow source color devices are made with Gallium Arsenide Phosphide on Gallium Phosphide Yellow Light Emitting Diode.
The Pure Orange source color devices are made with Gallium Arsenide Phosphide on Gallium Phosphide Pure Orange Light Emitting Diode.
The Pure Green source color devices are made with Gallium Phosphide Pure Green Light Emitting Diode.

Package Dimensions



Notes:
1. All dimensions are in millimeters (inches).
2. Tolerance is ±0.25(0.01") unless otherwise noted.
3. Lead spacing is measured where the lead emerge package.
4. Specifications are subjected to change without notice.

Selection Guide

Part No.	Dice	Lens Type	Iv (mcd) @ 10 mA		Viewing Angle
			Min.	Max.	2θ1/2
L-934HD	BRIGHT RED (GaP)	RED DIFFUSED	1.3	5	60°
L-934ID	HIGH EFFICIENCY RED (GaAsP/GaP)	RED DIFFUSED	8	50	60°
L-934IT		RED TRANSPARENT	20	125	50°
L-934EC		WATER CLEAR	20	125	50°
L-934ED	ORANGE (GaAsP/GaP)	ORANGE DIFFUSED	8	50	60°
L-934GD	GREEN (GaP)	GREEN DIFFUSED	8	32	60°
L-934GT		GREEN TRANSPARENT	20	80	50°
L-934GC		WATER CLEAR	20	80	50°
L-934YD	YELLOW (GaAsP/GaP)	YELLOW DIFFUSED	8	32	60°
L-934YT		YELLOW TRANSPARENT	10	50	50°
L-934YC		WATER CLEAR	10	50	50°
L-934ND	PURE ORANGE	ORANGE DIFFUSED	8	50	60°
L-934NT		ORANGE TRANSPARENT	20	125	50°
L-934NC		WATER CLEAR	20	125	50°
L-934PGD	PURE GREEN (GaP)	GREEN DIFFUSED	2	8	60°
L-934PGT		GREEN TRANSPARENT	3.2	20	50°
L-934PGC		WATER CLEAR	3.2	20	50°

Note:
1. θ1/2 is the angle from optical centerline where the luminous intensity is 1/2 the optical centerline value.

3mm LE low current

Kingbright®

T-1 (3mm) LOW CURRENT LED LAMPS

L-934LI HIGH EFFICIENCY RED L-934LY YELLOW

L-934LG GREEN

L-934LSR SUPER BRIGHT RED

Features

- MINIMUM LUMINOUS INTENSITY SPECIFIED AT 2 mA.
- HIGH LIGHT OUTPUT AT LOW CURRENTS.
- LOW POWER CONSUMPTION.
- LOW CURRENT REQUIREMENTS.
- WIDE VIEWING ANGLE.
- I.C. COMPATIBLE.
- RELIABLE AND RUGGED.

Description

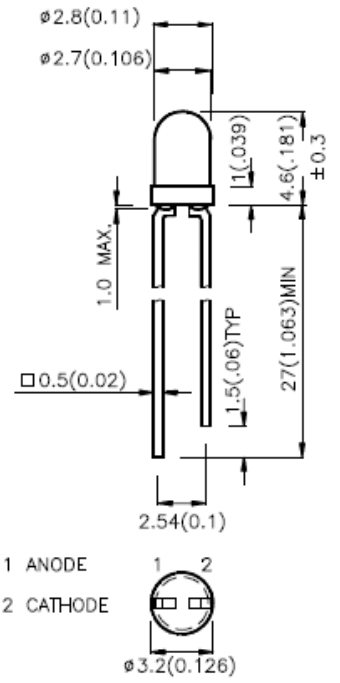
The Green source color devices are made with Gallium Phosphide Green Light Emitting Diode.

The High Efficiency Red source color devices are made with Gallium Arsenide Phosphide on Gallium Phosphide Orange Light Emitting Diode.

The Yellow source color devices are made with Gallium Arsenide Phosphide on Gallium Phosphide Yellow Light Emitting Diode.

The Super Bright Red source color devices are made with Gallium Aluminum Arsenide Red Light Emitting Diode.

Package Dimensions



Notes:
1. All dimensions are in millimeters (inches).
2. Tolerance is ±0.25(0.01") unless otherwise noted.
3. Lead spacing is measured where the lead emerge package.
4. Specifications are subjected to change without notice.

Selection Guide

Part No.	Dice	Lens Type	Iv (mcd) @ 2 mA		Viewing Angle
			Min.	Max.	2θ1/2
L-934LID	HIGH EFFICIENCY RED (GaAsP/GaP)	RED DIFFUSED	0.8	5	60°
L-934LGD	GREEN (GaP)	GREEN DIFFUSED	0.8	3.2	60°
L-934LYD	YELLOW (GaAsP/GaP)	YELLOW DIFFUSED	0.8	3.2	60°
L-934LSRD	SUPER BRIGHT RED (GaAlAs)	RED DIFFUSED	8	20	60°

Note:
1. θ1/2 is the angle from optical centerline where the luminous intensity is 1/2 the optical centerline value.

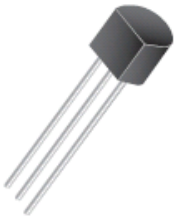
Transistor BC337-16

VISHAY

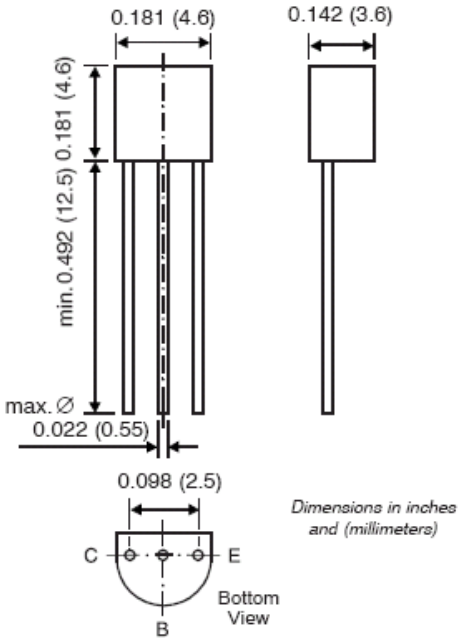
BC337 and BC338

Vishay Semiconductors
formerly General Semiconductor

Small Signal Transistors (NPN)



TO-226AA (TO-92)



Features

- NPN Silicon Epitaxial Planar Transistors for switching and amplifier applications. Especially suited for AF-driver stages and low power output stages.
- These types are also available subdivided into three groups -16, -25, and -40, according to their DC current gain. As complementary types, the PNP transistors BC327 and BC328 are recommended.
- On special request, this transistor is also manufactured in the pin configuration TO-18.

Mechanical Data

Case: TO-92 Plastic Package

Weight: approx. 0.18g

Packaging Codes/Options:
E6/Bulk – 5K per container, 20K/box
E7/4K per Ammo mag., 20K/box

Maximum Ratings & Thermal Characteristics

Ratings at 25°C ambient temperature unless otherwise specified.

Parameter		Symbol	Value	Unit
Collector-Emitter Voltage	BC337 BC338	V_{CES}	50 30	V
Collector-Emitter Voltage	BC337 BC338	V_{CEO}	45 25	V
Emitter-Base Voltage		V_{EBO}	5	V
Collector Current		I_C	800	mA
Peak Collector Current		I_{CM}	1	A
Base Current		I_B	100	mA
Power Dissipation at T_{amb} = 25°C		P_{tot}	625(1)	mW
Thermal Resistance Junction to Ambient Air		$R_{\theta JA}$	200(1)	°C/W
Junction Temperature		T_j	150	°C
Storage Temperature Range		T_S	–65 to +150	°C

Note:
(1) Valid provided that leads are kept at ambient temperature at a distance of 2 mm from case.

Transistor BD 139

SGS-THOMSON MICROELECTRONICS

BD135
BD137/BD139

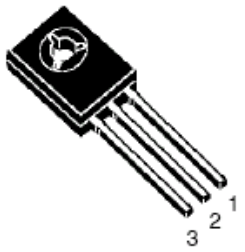
NPN SILICON TRANSISTOR

- SGS-THOMSON PREFERRED SALESTYPES

DESCRIPTION

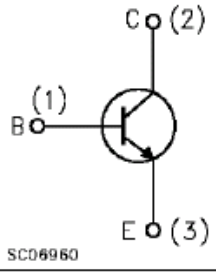
The BD135, BD137 and BD139 are silicon epitaxial planar NPN transistors in Jedec SOT-32 plastic package, designed for audio amplifiers and drivers utilizing complementary or quasi compementary circuits.

The complementary PNP types are the BD136 BD138 and BD140.



SOT-32

INTERNAL SCHEMATIC DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value			Unit
		BD135	BD137	BD139	
V_{CBO}	Collector-Base Voltage (I_E = 0)	45	60	80	V
V_{CEO}	Collector-Emitter Voltage (I_B = 0)	45	60	80	V
V_{EBO}	Emitter-Base Voltage (I_C = 0)	5			V
I_C	Collector Current	1.5			A
I_{CM}	Collector Peak Current	3			A
I_B	Base Current	0.5			A
P_{tot}	Total Dissipation at $T_c \leq 25$ °C	12.5			W
P_{tot}	Total Dissipation at $T_{amb} \leq 25$ °C	1.25			W
T_{stg}	Storage Temperature	-65 to 150			°C
T_j	Max. Operating Junction Temperature	150			°C

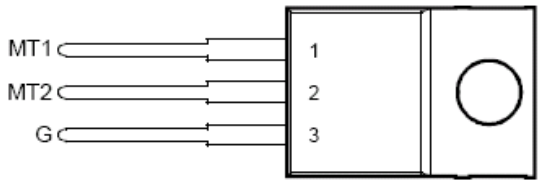
TRIAC TIC 255-D

TIC225 SERIES
SILICON TRIACS

Copyright © 1997, Power Innovations Limited, UK JULY 1975 - REVISED MARCH 1997

- **Sensitive Gate Triacs**
- **8 A RMS, 70 A Peak**
- **Glass Passivated Wafer**
- **400 V to 800 V Off-State Voltage**
- **Max I_{GT} of 5 mA (Quadrant 1)**

TO-220 PACKAGE
(TOP VIEW)



Pin 2 is in electrical contact with the mounting base.

MDC2ACA

absolute maximum ratings over operating case temperature (unless otherwise noted)

RATING		SYMBOL	VALUE	UNIT
Repetitive peak off-state voltage (see Note 1)	TIC225D	V_{DRM}	400	V
	TIC225M		600	
	TIC225S		700	
	TIC225N		800	
Full-cycle RMS on-state current at (or below) 70°C case temperature (see Note 2)		$I_{T(RMS)}$	8	A
Peak on-state surge current full-sine-wave (see Note 3)		I_{TSM}	70	A
Peak on-state surge current half-sine-wave (see Note 4)		I_{TSM}	80	A
Peak gate current		I_{GM}	±1	A
Peak gate power dissipation at (or below) 85°C case temperature (pulse width ≤ 200 μs)		P_{GM}	2.2	W
Average gate power dissipation at (or below) 85°C case temperature (see Note 5)		$P_{G(AV)}$	0.9	W
Operating case temperature range		T_C	-40 to +110	°C
Storage temperature range		T_{stg}	-40 to +125	°C
Lead temperature 1.6 mm from case for 10 seconds		T_L	230	°C

NOTES: 1. These values apply bidirectionally for any value of resistance between the gate and Main Terminal 1.
2. This value applies for 50-Hz full-sine-wave operation with resistive load. Above 70°C derate linearly to 110°C case temperature at the rate of 200 mA/°C.
3. This value applies for one 50-Hz full-sine-wave when the device is operating at (or below) the rated value of on-state current. Surge may be repeated after the device has returned to original thermal equilibrium. During the surge, gate control may be lost.
4. This value applies for one 50-Hz half-sine-wave when the device is operating at (or below) the rated value of on-state current. Surge may be repeated after the device has returned to original thermal equilibrium. During the surge, gate control may be lost.
5. This value applies for a maximum averaging time of 20 ms.

electrical characteristics at 25°C case temperature (unless otherwise noted)

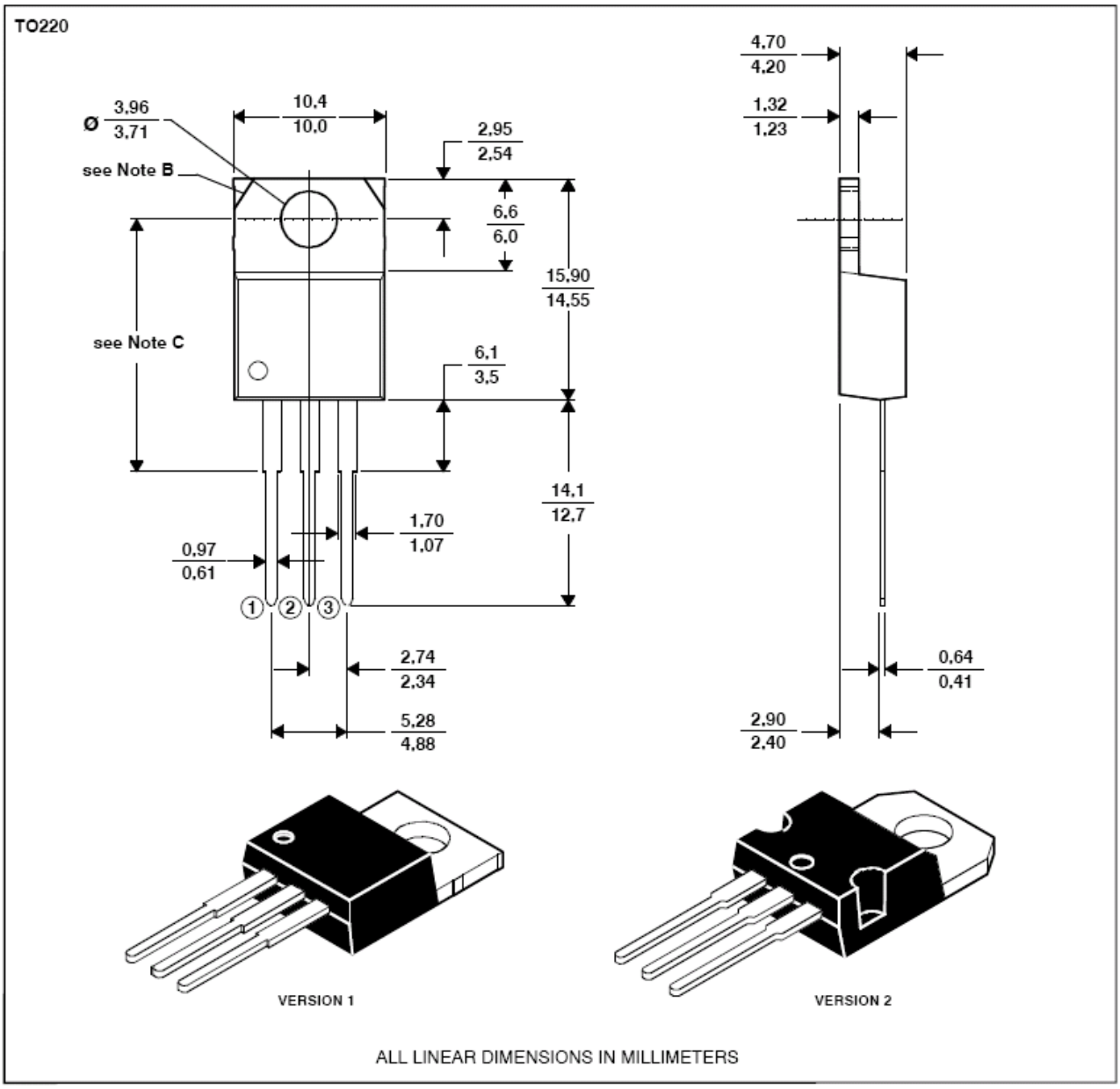
PARAMETER		TEST CONDITIONS			MIN	TYP	MAX	UNIT
I_{DRM}	Repetitive peak off-state current	V_D = rated V_{DRM}	I_G = 0	T_C = 110°C			±2	mA
I_{GTM}	Peak gate trigger current	V_{supply} = +12 V†	R_L = 10 Ω	$t_{p(g)}$ > 20 μs		0.8	5	mA
		V_{supply} = +12 V†	R_L = 10 Ω	$t_{p(g)}$ > 20 μs		-4.5	-20	
		V_{supply} = -12 V†	R_L = 10 Ω	$t_{p(g)}$ > 20 μs		-3.5	-10	
		V_{supply} = -12 V†	R_L = 10 Ω	$t_{p(g)}$ > 20 μs		11.7	30	
V_{GTM}	Peak gate trigger voltage	V_{supply} = +12 V†	R_L = 10 Ω	$t_{p(g)}$ > 20 μs		0.7	2	V
		V_{supply} = +12 V†	R_L = 10 Ω	$t_{p(g)}$ > 20 μs		-0.7	-2	
		V_{supply} = -12 V†	R_L = 10 Ω	$t_{p(g)}$ > 20 μs		-0.8	-2	
		V_{supply} = -12 V†	R_L = 10 Ω	$t_{p(g)}$ > 20 μs		0.9	2	

† All voltages are with respect to Main Terminal 1.

TO-220

3-pin plastic flange-mount package

This single-in-line package consists of a circuit mounted on a lead frame and encapsulated within a plastic compound. The compound will withstand soldering temperature with no deformation, and circuit performance characteristics will remain stable when operated in high humidity conditions. Leads require no additional cleaning or processing when used in soldered assembly.



NOTES: A. The centre pin is in electrical contact with the mounting tab. MDXXBE
B. Mounting tab corner profile according to package version.
C. Typical fixing hole centre stand off height according to package version. Version 1, 18.0 mm. Version 2, 17.6 mm.

PRODUCT INFORMATION

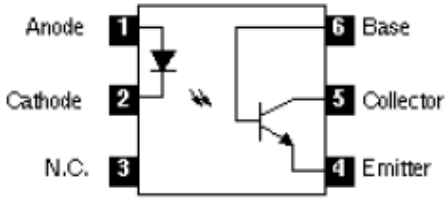
Power INNOVATIONS

Optokoppler 4N25

4N25
Optoisolator
NPN Transistor Output

Description:

The 4N25 is a gallium arsenide, infrared emitting diode in a 6-Lead DIP type package coupled with a silicon phototransistor.



Absolute Maximum Ratings: (T_A = +25°C C unless otherwise specified)

Infrared Emitting Diode	
Reverse Voltage, V_R	3V
Continuous Forward Current, I_F	60mA
Peak Forward Current (Pulse Width 1µs, 300pps), I_F	3A
Power Dissipation, P_D	200mW
Derate Above +25°C	2.6mW/°C
Phototransistor	
Collector-Emitter Voltage, V_{CEO}	30V
Emitter-Collector Voltage, V_{ECO}	7V
Collector-Base Voltage, V_{CBO}	70V
Continuous Collector Current, I_C	100mA
Detector Power Dissipation, P_D	200mW
Derate Above +25°C	2.6mW/°C
Total Device	
Isolation Source Voltage (Input-to-Output), V_{ISO}	
Peak	1500V
RMS	1060V
Operating Temperature Range, T_{opr}	-55° to +100°C
Storage Temperature Range, T_{stg}	-55° to +150°C
Lead Temperature (During Soldering, 10sec), T_L	+260°C

Agilent 4N25
Phototransistor Optocoupler
General Purpose Type
Data Sheet

Description
The 4N25 is an optocoupler for general purpose applications. It contains a light emitting diode optically coupled to a phototransistor. It is packaged in a 6-pin DIP package and available in wide-lead spacing option and lead bend SMD option. Response time, t_r, is typically 3 µs and minimum CTR is 20% at input current of 10 mA.

Ordering Information
Specify part number followed by Option Number (if desired).

4N25-XXXX

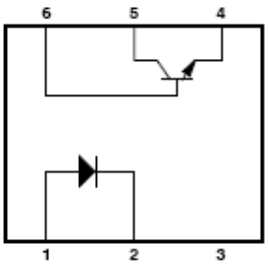
Option Number

060 = IEC/EN/DIN EN 60747-5-2 Option
W00 = 0.4" Lead Spacing Option
300 = Lead Bend SMD Option
500 = Tape and Reel Packaging Option
XXXE = Lead Free Option

Features
- Response time (t_r: typ., 3 µs at V_{CE} = 10 V, I_C = 2 mA, R_L = 100 Ω)
- Current Transfer Ratio (CTR: min. 20% at I_F = 10 mA, V_{CE} = 10 V)
- Input-output isolation voltage (V_{iso} = 2500 Vrms)
- Dual-in-line package
- UL approved
- CSA approved
- IEC/EN/DIN EN 60747-5-2 approved
- Options available:
  - Leads with 0.4" (10.16 mm) spacing (W00)
  - Leads bends for surface mounting (300)
  - Tape and reel for SMD (500)
  - IEC/EN/DIN EN 60747-5-2 approvals (060)

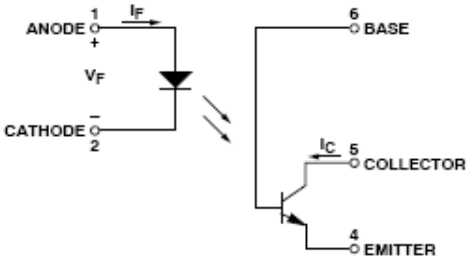
Functional Diagram

PIN NO. AND INTERNAL CONNECTION DIAGRAM



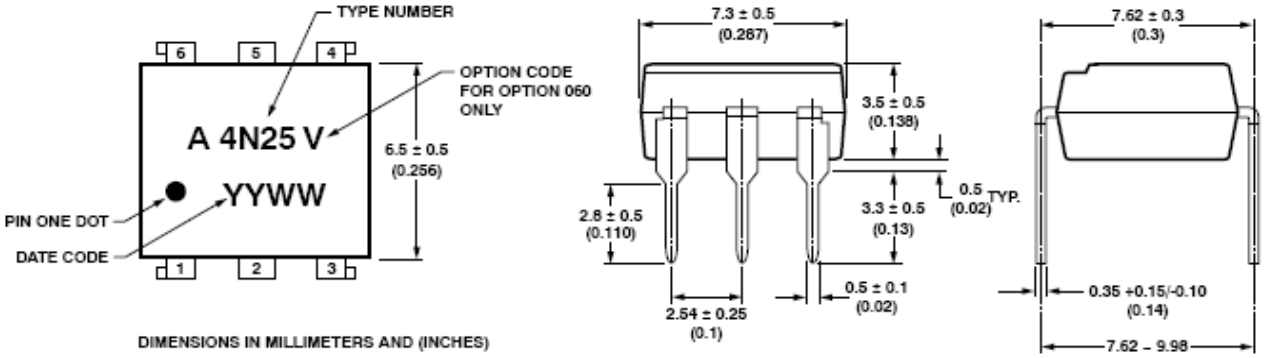
1. ANODE
2. CATHODE
3. NC
4. EMITTER
5. COLLECTOR
6. BASE

Schematic



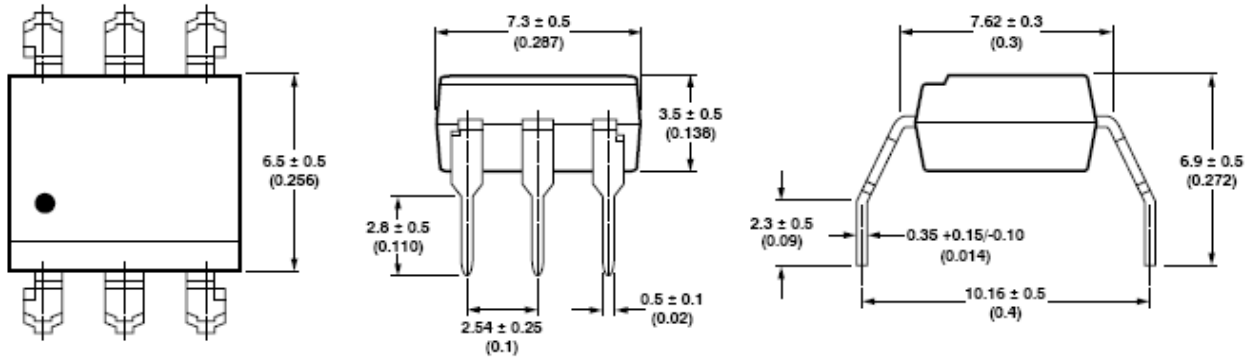
Applications
- I/O interfaces for computers
- System appliances, measuring instruments
- Signal transmission between circuits of different potentials and impedances

Package Outline Drawings



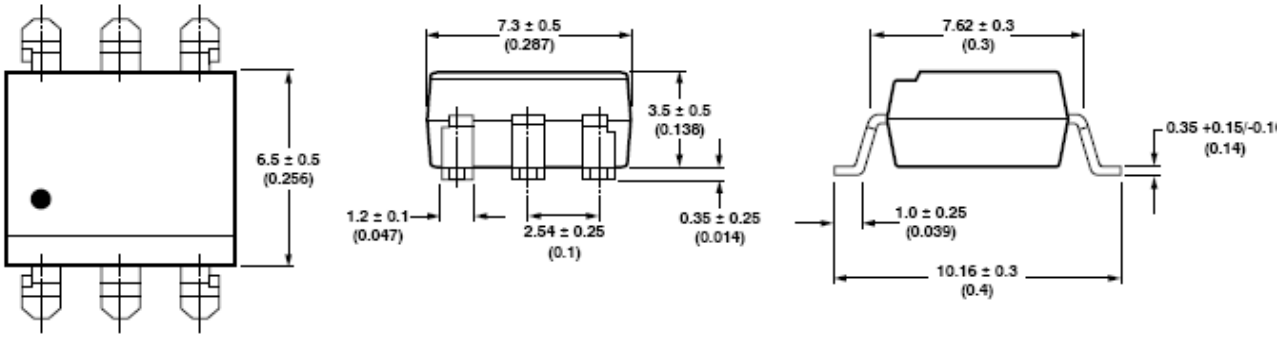
DIMENSIONS IN MILLIMETERS AND (INCHES)

Package Outline – Option W00



DIMENSIONS IN MILLIMETERS AND (INCHES)

Package Outline – Option 300



DIMENSIONS IN MILLIMETERS AND (INCHES)

Optokoppler MOC 3020

MOC3020 THRU MOC3023
OPTOCOUPLERS/OPTOISOLATORS

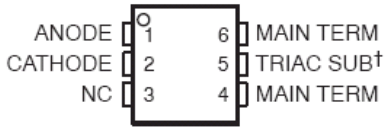
SOES025A – OCTOBER 1986 – REVISED APRIL 1998

- **400 V Phototriac Driver Output**
- **Gallium-Arsenide-Diode Infrared Source and Optically-Coupled Silicon Traic Driver (Bilateral Switch)**
- **UL Recognized . . . File Number E65085**
- **High Isolation . . . 7500 V Peak**
- **Output Driver Designed for 220 Vac**
- **Standard 6-Terminal Plastic DIP**
- **Directly Interchangeable with Motorola MOC3020, MOC3021, MOC3022, and MOC3023**

typical 115/240 Vac(rms) applications

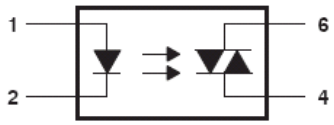
- **Solenoid/Valve Controls**
- **Lamp Ballasts**
- **Interfacing Microprocessors to 115/240 Vac Peripherals**
- **Motor Controls**
- **Incandescent Lamp Dimmers**

MOC3020 – MOC3023 . . . PACKAGE
(TOP VIEW)



† Do not connect this terminal
NC – No internal connection

logic diagram



absolute maximum ratings at 25°C free-air temperature (unless otherwise noted)†

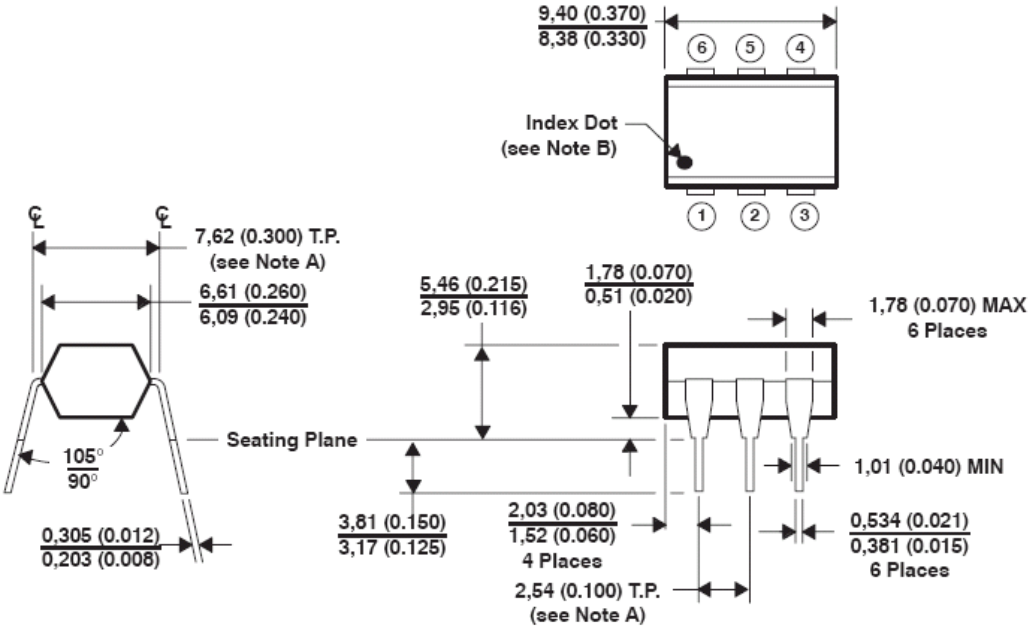
Input-to-output peak voltage, 5 s maximum duration, 60 Hz (see Note 1)	7.5 kV
Input diode reverse voltage	3 V
Input diode forward current, continuous	50 mA
Output repetitive peak off-state voltage	400 V
Output on-state current, total rms value (50-60 Hz, full sine wave): T_A = 25°C	100 mA
T_A = 70°C	50 mA
Output driver nonrepetitive peak on-state current (t_w = 10 ms, duty cycle = 10%, see Figure 7)	1.2 A
Continuous power dissipation at (or below) 25°C free-air temperature:	
Infrared-emitting diode (see Note 2)	100 mW
Phototriac (see Note 3)	300 mW
Total device (see Note 4)	330 mW
Operating junction temperature range, T_J	–40°C to 100°C
Storage temperature range, T_{stg}	–40°C to 150°C
Lead temperature 1,6 (1/16 inch) from case for 10 seconds	260°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. Input-to-output peak voltage is the internal device dielectric breakdown rating.
2. Derate linearly to 100°C free-air temperature at the rate of 1.33 mW/°C.
3. Derate linearly to 100°C free-air temperature at the rate of 4 mW/°C.
4. Derate linearly to 100°C free-air temperature at the rate of 4.4 mW/°C.

MECHANICAL INFORMATION

Each device consists of a gallium-arsenide infrared-emitting diode optically coupled to a silicon phototriac mounted on a 6-terminal lead frame encapsulated within an electrically nonconductive plastic compound. The case can withstand soldering temperature with no deformation and device performance characteristics remain stable when operated in high-humidity conditions.



NOTES: A. Leads are within 0,13 (0.005) radius of true position (T.P.) with maximum material condition and unit installed.
B. Pin 1 identified by index dot.
C. The dimensions given fall within JEDEC MO-001 AM dimensions.
D. All linear dimensions are given in millimeters and parenthetically given in inches.

Brückengleichrichter B250C800DIP

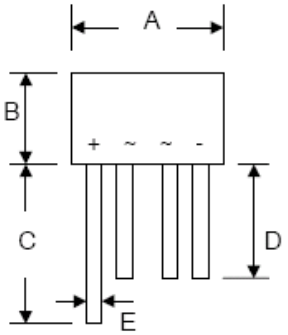
WTE
POWER SEMICONDUCTORS

B40C800 – B500C800

0.8A BRIDGE RECTIFIER

Features

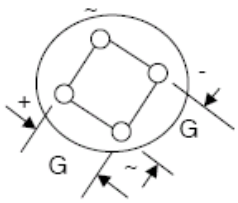
- Diffused Junction
- Low Forward Voltage Drop
- High Current Capability
- High Reliability
- High Surge Current Capability
- Ideal for Printed Circuit Boards



WOB		
Dim	Min	Max
A	8.60	9.10
B	5.0	5.50
C	27.9	—
D	25.4	—
E	0.71	0.81
G	4.60	5.60
All Dimensions in mm		

Mechanical Data

- Case: Molded Plastic
- Terminals: Plated Leads Solderable per MIL-STD-202, Method 208
- Polarity: As Marked on Body
- Weight: 1.1 grams (approx.)
- Mounting Position: Any
- Marking: Type Number



Maximum Ratings and Electrical Characteristics @T_A=25°C unless otherwise specified

Single Phase, half wave, 60Hz, resistive or inductive load.
For capacitive load, derate current by 20%.

Characteristic	Symbol	B40C 800	B80C 800	B125C 800	B250C 800	B380C 800	B500C 800	Unit
Peak Repetitive Reverse Voltage Working Peak Reverse Voltage DC Blocking Voltage	V_{RRM} V_{RWM} V_R	100	200	300	600	900	1200	V
Input Voltage Recommended	$V_{R(RMS)}$	40	80	125	250	380	500	V
Average Rectified Output Current (Note 1) @T_A = 50°C	Io	0.8						A
Non-Repetitive Peak Forward Surge Current 8.3ms Single half sine-wave superimposed on rated load (JEDEC Method)	I_{FSM}	45						A
Forward Voltage (per element) @I_F = 0.8A	V_{FM}	1.0						V
Peak Reverse Current @T_A = 25°C At Rated DC Blocking Voltage @T_A = 100°C	I_{RM}	10 500						μA
Operating Temperature Range	T_j	-55 to +125						°C
Storage Temperature Range	T_{STG}	-55 to +150						°C

Note: 1. Leads maintained at ambient temperature at a distance of 9.5mm from the case.

Brückengleichrichter B40C1400-WW+

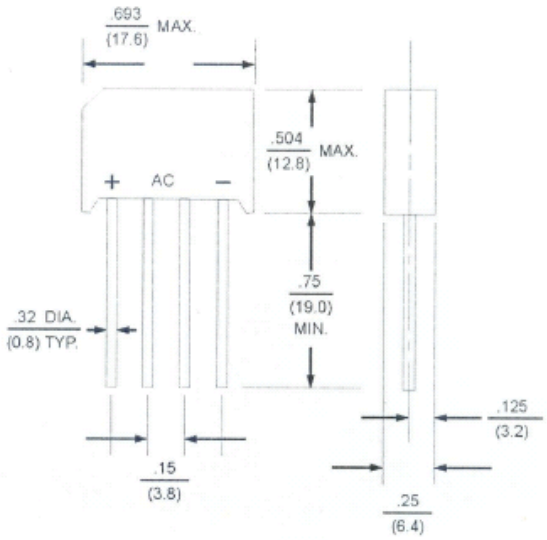
B40C thru B500C2300/1500

SILICON BRIDGE RECTIFIERS

REVERSE VOLTAGE - **50** to **1000** Volts
FORWARD CURRENT - **2.0** Amperes

FEATURES

- Surge overload rating - 50 amperes peak
- Ideal for printed circuit board
- Plastic material has Underwriters Laboratory Flammability Classification 94V-0
- Mounting Position: Any



Dimensions in inches and (millimeters)

MAXIMUM RATINGS AND ELECTRICAL CHARACTERISTICS

Rating at 25°C ambient temperature unless otherwise specified.
Resistive or inductive load, 60Hz.
For capacitive load, derate current by 20%

CHARACTERISTICS		B40C 2300/1500	B80C 2300/1500	B125C 2300/1500	B250C 2300/1500	B380C 2300/1500	B500C 2300/1500	UNIT
Maximum Recurrent Peak Reverse Voltage	50	100	200	400	600	800	1000	V
Maximum RMS Bridge Input Voltage	35	70	140	280	420	560	700	V
Maximum DC Blocking Voltage	50	100	200	400	600	800	1000	V
Maximum Average Forward Rectified Output Current @T_A=50°C	2.0							A
Peak Forward Surge Current, 8.3ms single half sine-wave super imposed on rated load	50.0							A
Maximum Forward Voltage Drop per Bridge Element at 2.0A Peak	1.1							V
Maximum Reverse Current at Rated DC Blocking Voltage per Element	10.0							μA
Maximum Reverse Current at Rated DC Blocking Voltage per Element T_A=100°C	1.0							mA
Operating Temperature Range T_J	-55 to +125							°C
Storage Temperature Range T_A	-55 to +150							°C

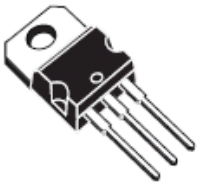
Spannungsregler LS78S05 und LS78S12

L78S00 series

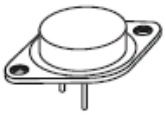
2A Positive voltage regulators

Feature summary

- Output current to 2A
- Output voltages of 5; 7.5; 9; 10; 12; 15; 18; 24V
- Thermal overload protection
- Short circuit protection
- Output transition SOA protection



TO-220

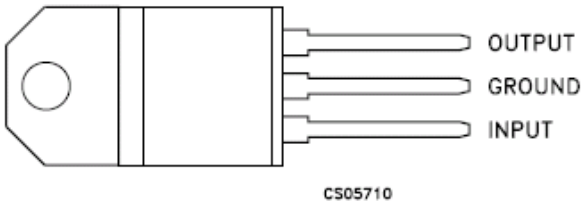


TO-3

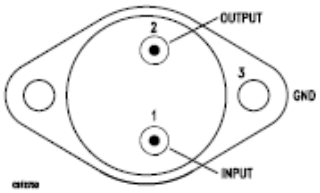
Description

The L78S00 series of three-terminal positive regulators is available in TO-220 and TO-3 packages and with several fixed output voltages, making it useful in a wide range of applications. These regulators can provide local on-card regulation, eliminating the distribution problems associated with single point regulation. Each type employs internal current limiting, thermal shut-down and safe area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 2A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.

Figure 1. Pin connections (top view)



TO-220



TO-3

Table 1. Absolute maximum ratings

Symbol	Parameter		Value	Unit
V_I	DC Input voltage	for V_O= 5 to 18V	35	V
		for V_O= 24V	40	
I_O	Output current		Internally Limited	
P_D	Power dissipation		Internally Limited	
T_{STG}	Storage temperature range		-65 to 150	°C
T_{OP}	Operating junction temperature range	for L78S00	-55 to 150	°C
		for L78S00C	0 to 150	

Note: Absolute Maximum Ratings are those values beyond which damage to the device may occur. Functional operation under these condition is not implied.

Table 2. Thermal Data

Symbol	Parameter	TO-220	TO-3	Unit
R_{thJC}	Thermal resistance junction-case	5	4	°C/W
R_{thJA}	Thermal resistance junction-ambient	50	35	°C/W

Figure 3. Application circuits

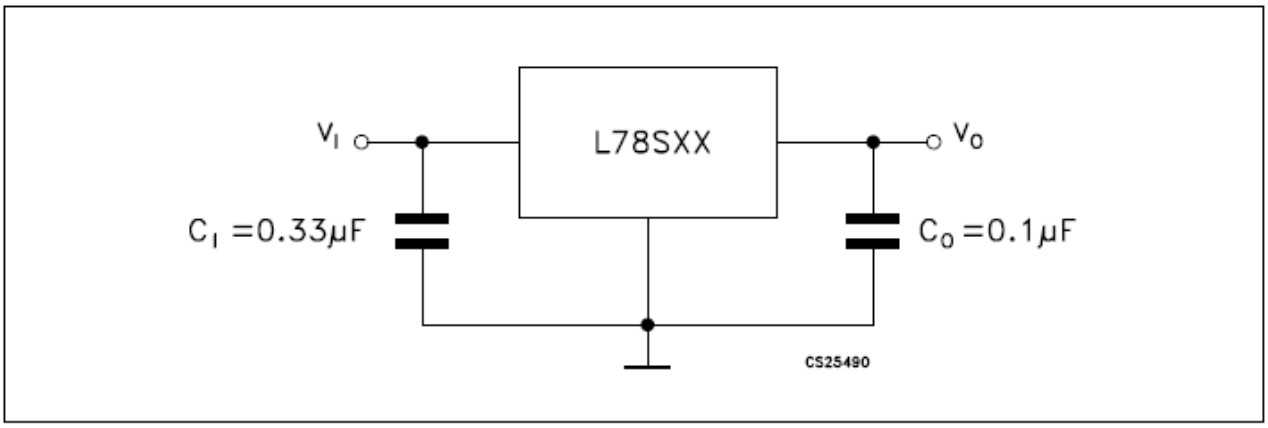


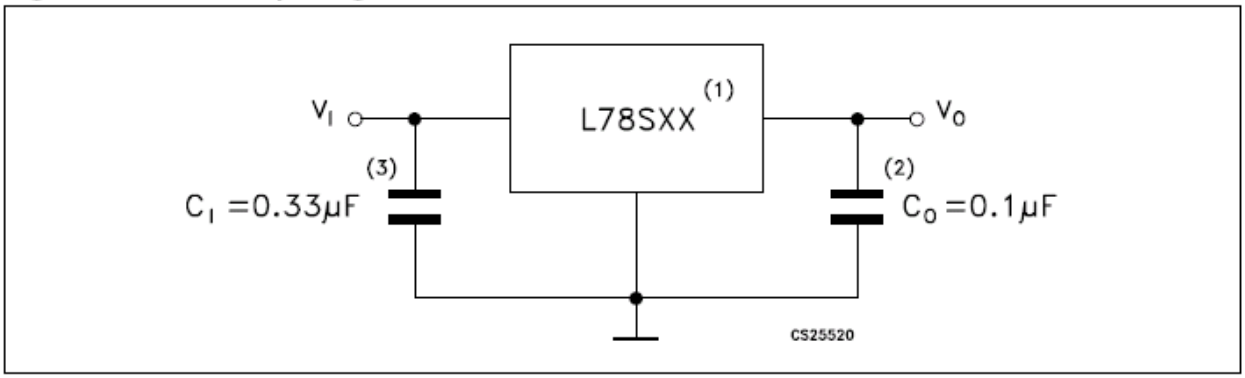
Table 3. Electrical characteristics of L78S05 (refer to the test circuits, T_J = 25°C, V_I = 10V, I_O = 500 mA, unless otherwise specified)

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
V_O	Output voltage		4.8	5	5.2	V
V_O	Output voltage	I_O = 1A, V_I = 7V	4.75	5	5.25	V
ΔV_O	Line regulation	V_I = 7 to 25V			100	mV
		V_I = 8 to 25V			50	
ΔV_O	Load regulation	I_O = 20 mA to 2A			100	mV
I_d	Quiescent current				8	mA
ΔI_d	Quiescent current change	I_O = 20mA to 1A			0.5	mA
		V_I = 7 to 25 V, I_O = 20mA			1.3	
$\Delta V_O/\Delta T$	Output voltage drift	I_O = 5mA, T_J = -55°C to 150°C		-1.1		mV/°C
eN	Output noise voltage	B =10Hz to 100KHz		40		µV
SVR	Supply voltage rejection	f = 120Hz	60			dB
V_d	Dropout voltage	I_O ≤1A	8			V
R_O	Output resistance	f = 1 KHz		17		mΩ
I_{sc}	Short circuit current	V_I = 27V		500		mA
I_{scp}	Short circuit peak current			3		A

Table 7. Electrical characteristics of L78S12 (refer to the test circuits, T_J = 25°C, V_I = 19V, I_O = 500 mA, unless otherwise specified)

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
V_O	Output voltage		11.5	12	12.5	V
V_O	Output voltage	I_O = 1A, V_I = 14.5V	11.4	12	12.6	V
ΔV_O	Line regulation	V_I = 14.5 to 30V			240	mV
		V_I = 16 to 22V			120	
ΔV_O	Load regulation	I_O = 20 mA to 2A			160	mV
I_d	Quiescent current				8	mA
ΔI_d	Quiescent current change	I_O = 20mA to 1A			0.5	mA
		V_I = 14.5 to 30 V, I_O = 20mA			1	
$\Delta V_O/\Delta T$	Output voltage drift	I_O = 5mA, T_J = -55°C to 150°C		-1		mV/°C
eN	Output noise voltage	B =10Hz to 100KHz		75		µV
SVR	Supply voltage rejection	f = 120Hz	53			dB
V_d	Dropout voltage	I_O ≤1.5A	15			V
R_O	Output resistance	f = 1 KHz		18		mΩ
I_{sc}	Short circuit current	V_I = 27V		500		mA
I_{scp}	Short circuit peak current			3		A

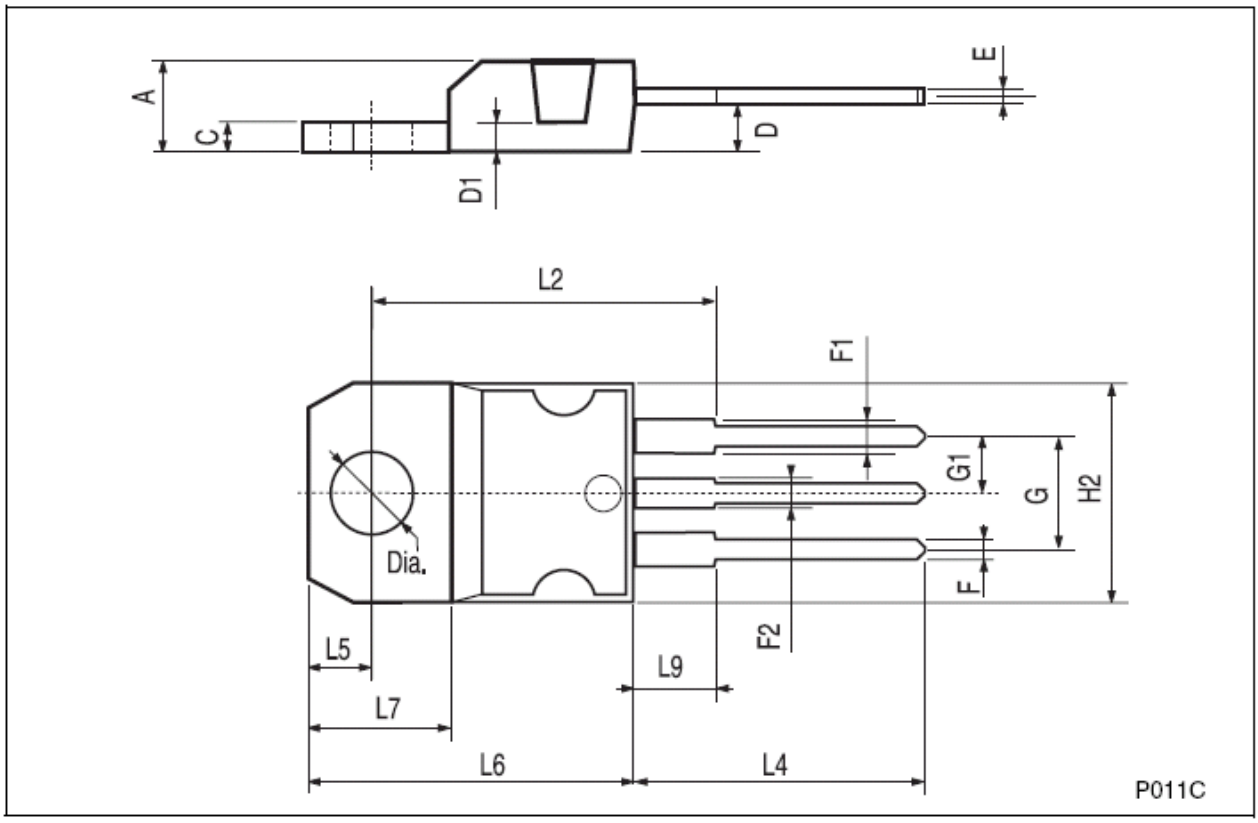
Figure 16. Fixed output regulator



1. To specify an output voltage, substitute voltage value for "XX".
2. Although no output capacitor is need for stability, it does improve transient response.
3. Required if regulator is locate an appreciable distance from power supply filter.

TO-220 MECHANICAL DATA

DIM.	mm.			inch		
	MIN.	TYP	MAX.	MIN.	TYP.	MAX.
A	4.40		4.60	0.173		0.181
C	1.23		1.32	0.048		0.051
D	2.40		2.72	0.094		0.107
D1		1.27			0.050	
E	0.49		0.70	0.019		0.027
F	0.61		0.88	0.024		0.034
F1	1.14		1.70	0.044		0.067
F2	1.14		1.70	0.044		0.067
G	4.95		5.15	0.194		0.203
G1	2.4		2.7	0.094		0.106
H2	10.0		10.40	0.393		0.409
L2		16.4			0.645	
L4	13.0		14.0	0.511		0.551
L5	2.65		2.95	0.104		0.116
L6	15.25		15.75	0.600		0.620
L7	6.2		6.6	0.244		0.260
L9	3.5		3.93	0.137		0.154
DIA.	3.75		3.85	0.147		0.151



CMOS EEPROM ST 24C512 BN6

Features

- **Low-voltage and Standard-voltage Operation**
  - **2.7 (V_{CC} = 2.7V to 5.5V)**
  - **1.8 (V_{CC} = 1.8V to 3.6V)**
- **Internally Organized 65,536 x 8**
- **2-wire Serial Interface**
- **Schmitt Triggers, Filtered Inputs for Noise Suppression**
- **Bidirectional Data Transfer Protocol**
- **1 MHz (5V), 400 kHz (2.7V) and 100 kHz (1.8V) Compatibility**
- **Write Protect Pin for Hardware and Software Data Protection**
- **128-byte Page Write Mode (Partial Page Writes Allowed)**
- **Self-timed Write Cycle (5 ms Typical)**
- **High Reliability**
  - **Endurance: 100,000 Write Cycles**
  - **Data Retention: 40 Years**
- **Automotive Grade and Extended Temperature Devices Available**
- **8-lead PDIP, 8-lead EIAJ SOIC, 8-lead JEDEC SOIC, 8-lead TSSOP, 8-lead LAP and 8-ball dBGA™ Packages**

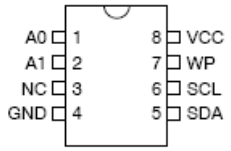
Description

The AT24C512 provides 524,288 bits of serial electrically erasable and programmable read only memory (EEPROM) organized as 65,536 words of 8 bits each. The device's cascadable feature allows up to 4 devices to share a common 2-wire bus. The device is optimized for use in many industrial and commercial applications where low-power and low-voltage operation are essential. The devices are available in space-saving 8-pin PDIP, 8-lead EIAJ SOIC, 8-lead JEDEC SOIC, 8-lead TSSOP, 8-lead Leadless Array (LAP) and 8-ball dBGA packages. In addition, the entire family is available in 2.7V (2.7V to 5.5V) and 1.8V (1.8V to 3.6V) versions.

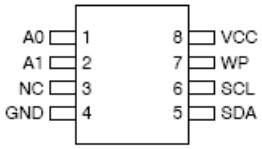
Pin Configurations

Pin Name	Function
A0 - A1	Address Inputs
SDA	Serial Data
SCL	Serial Clock Input
WP	Write Protect
NC	No Connect

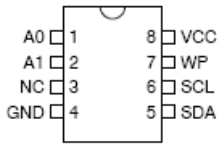
8-lead TSSOP



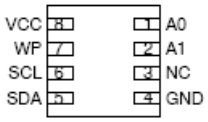
8-lead SOIC



8-lead PDIP

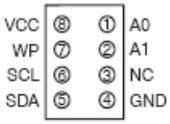


8-lead Leadless Array



Bottom View

8-ball dBGA



Bottom View

2-wire Serial EEPROM

512K (65,536 x 8)

AT24C512

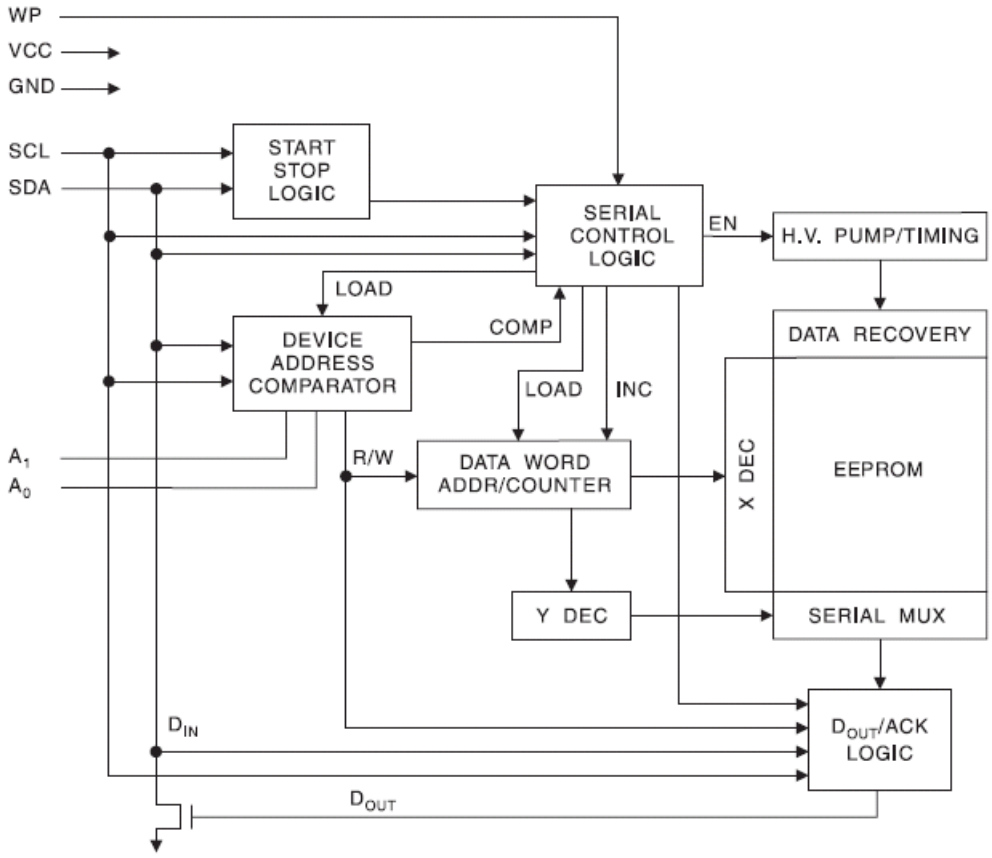
Rev. 1116I–SEEPR–3/03

Absolute Maximum Ratings*

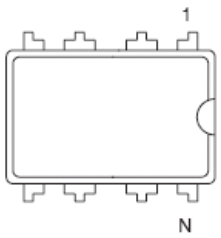
Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.25V
DC Output Current	5.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

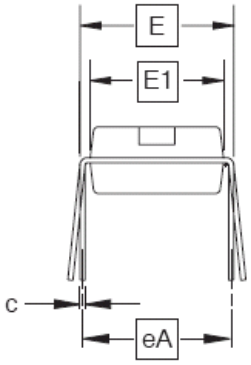
Block Diagram



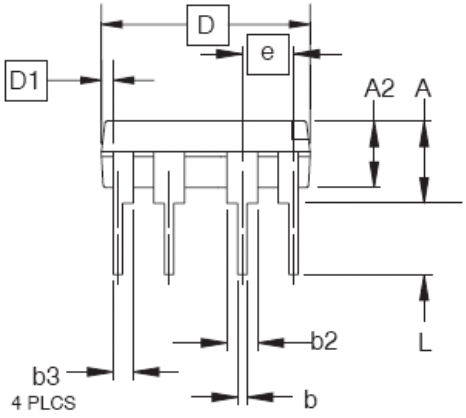
8P3 – PDIP



Top View



End View



Side View

COMMON DIMENSIONS
(Unit of Measure = inches)

SYMBOL	MIN	NOM	MAX	NOTE
A			0.210	2
A2	0.115	0.130	0.195	
b	0.014	0.018	0.022	5
b2	0.045	0.060	0.070	6
b3	0.030	0.039	0.045	6
c	0.008	0.010	0.014	
D	0.355	0.365	0.400	3
D1	0.005			3
E	0.300	0.310	0.325	4
E1	0.240	0.250	0.280	3
e	0.100 BSC			
eA	0.300 BSC			4
L	0.115	0.130	0.150	2

Notes:
1. This drawing is for general information only; refer to JEDEC Drawing MS-001, Variation BA for additional information.
2. Dimensions A and L are measured with the package seated in JEDEC seating plane Gauge GS-3.
3. D, D1 and E1 dimensions do not include mold Flash or protrusions. Mold Flash or protrusions shall not exceed 0.010 inch.
4. E and eA measured with the leads constrained to be perpendicular to datum.
5. Pointed or rounded lead tips are preferred to ease insertion.
6. b2 and b3 maximum dimensions do not include Dambar protrusions. Dambar protrusions shall not exceed 0.010 (0.25 mm).

01/09/02

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**8P3**, 8-lead, 0.300" Wide Body, Plastic Dual In-line Package (PDIP)	8P3	B

Operationsverstärker LM 358 DIP

LM158,A-LM258,A LM358,A

LOW POWER DUAL OPERATIONAL AMPLIFIERS

- INTERNALLY FREQUENCY COMPENSATED
- LARGE DC VOLTAGE GAIN : 100dB
- WIDE BANDWIDTH (unity gain) : 1.1MHz (temperature compensated)
- VERY LOW SUPPLY CURRENT/OP (500µA) - ESSENTIALLY INDEPENDENT OF SUPPLY VOLTAGE
- LOW INPUT BIAS CURRENT : 20nA (temperature compensated)
- LOW INPUT OFFSET VOLTAGE : 2mV
- LOW INPUT OFFSET CURRENT : 2nA
- INPUT COMMON-MODE VOLTAGE RANGE INCLUDES GROUND
- DIFFERENTIAL INPUT VOLTAGE RANGE EQUAL TO THE POWER SUPPLY VOLTAGE
- LARGE OUTPUT VOLTAGE SWING 0V TO (Vcc – 1.5V)



N
DIP8
(Plastic Package)



D
SO8
(Plastic Micropackage)



P
TSSOP8
(Thin Shrink Small Outline Package)

ORDER CODES

Part Number	Temperature Range	Package		
		N	D	P
LM158,A	–55°C, +125°C	•	•	•
LM258,A	–40°C, +105°C	•	•	•
LM358,A	0°C, +70°C	•	•	•
Example : LM258N				

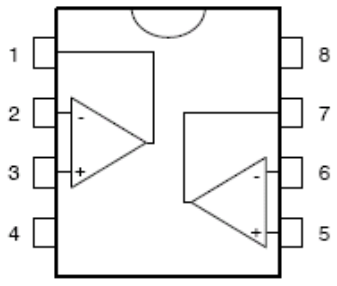
DESCRIPTION

These circuits consist of two independent, high gain, internally frequency compensated which were designed specifically to operate from a single power supply over a wide range of voltages. The low power supply drain is independent of the magnitude of the power supply voltage.

Application areas include transducer amplifiers, dc gain blocks and all the conventional op-amp circuits which now can be more easily implemented in single power supply systems. For example, these circuits can be directly supplied with the standard + 5V which is used in logic systems and will easily provide the required interface electronics without requiring any additional power supply.

In the linear mode the input common-mode voltage range includes ground and the output voltage can also swing to ground, even though operated from only a single power supply voltage.

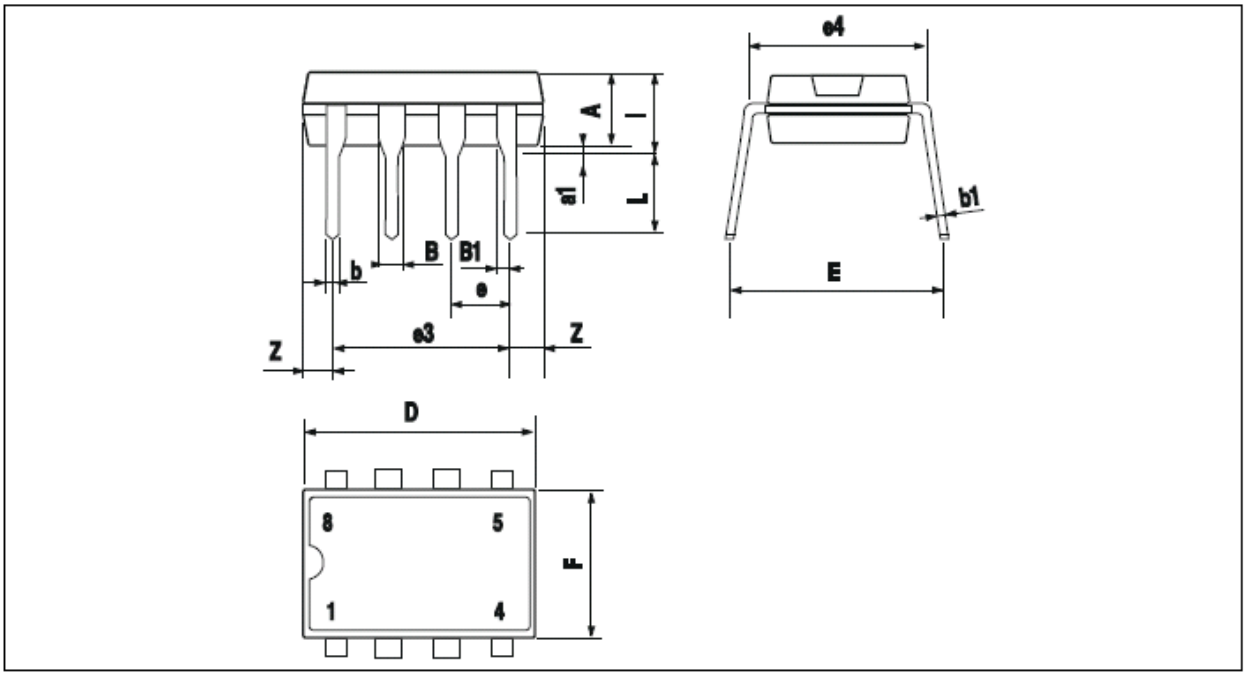
PIN CONNECTIONS (top view)



1 - Output 1
2 - Inverting input 1
3 - Non-inverting input 1
4 - V_{CC}^-
5 - Non-inverting input 2
6 - Inverting input 2
7 - Ouput 2
8 - V_{CC}^+

LM158,A - LM258,A - LM358,A

PACKAGE MECHANICAL DATA
8 PINS - PLASTIC DIP



Dim.	Millimeters			Inches		
	Min.	Typ.	Max.	Min.	Typ.	Max.
A		3.32			0.131	
a1	0.51			0.020		
B	1.15		1.65	0.045		0.065
b	0.356		0.55	0.014		0.022
b1	0.204		0.304	0.008		0.012
D			10.92			0.430
E	7.95		9.75	0.313		0.384
e		2.54			0.100	
e3		7.62			0.300	
e4		7.62			0.300	
F			6.6			0260
i			5.08			0.200
L	3.18		3.81	0.125		0.150
Z			1.52			0.060

DIP8.TBL

n

Schottky Diode 1N5822

1N5820, 1N5821, 1N5822

1N5820 and 1N5822 are Preferred Devices

Axial Lead Rectifiers

This series employs the Schottky Barrier principle in a large area metal–to–silicon power diode. State–of–the–art geometry features chrome barrier metal, epitaxial construction with oxide passivation and metal overlap contact. Ideally suited for use as rectifiers in low–voltage, high–frequency inverters, free wheeling diodes, and polarity protection diodes.

Features

- Extremely Low V_F
- Low Power Loss/High Efficiency
- Low Stored Charge, Majority Carrier Conduction
- Shipped in plastic bags, 500 per bag
- Available in Tape and Reel, 1500 per reel, by adding a "RL" suffix to the part number
- Pb–Free Packages are Available*

Mechanical Characteristics:

- Case: Epoxy, Molded
- Weight: 1.1 Gram (Approximately)
- Finish: All External Surfaces Corrosion Resistant and Terminal Leads are Readily Solderable
- Lead Temperature for Soldering Purposes: 260°C Max. for 10 Seconds
- Polarity: Cathode indicated by Polarity Band

*For additional information on our Pb–Free strategy and soldering details, please download the ON Semiconductor Soldering and Mounting Techniques Reference Manual, SOLDERRM/D.

ON Semiconductor®

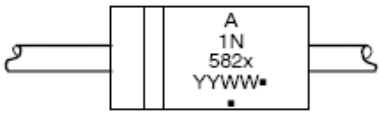
http://onsemi.com

SCHOTTKY BARRIER RECTIFIERS 3.0 AMPERES 20, 30, 40 VOLTS



AXIAL LEAD
CASE 267–05
(DO–201AD)
STYLE 1

MARKING DIAGRAM



A	= Assembly Location
1N582x	= Device Code
x	= 0, 1, or 2
YY	= Year
WW	= Work Week
▪	= Pb–Free Package

(Note: Microdot may be in either location)

ORDERING INFORMATION

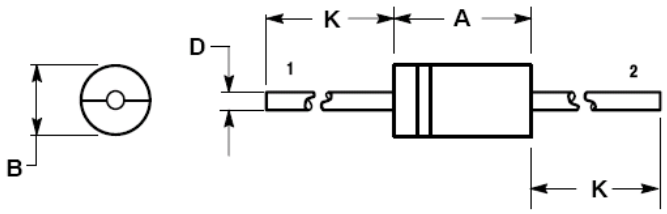
See detailed ordering and shipping information on page 3 of this data sheet.

Preferred devices are recommended choices for future use and best overall value.

1N5820, 1N5821, 1N5822

PACKAGE DIMENSIONS

AXIAL LEAD
CASE 267–05
(DO–201AD)
ISSUE G



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	0.287	0.374	7.30	9.50
B	0.189	0.209	4.80	5.30
D	0.047	0.051	1.20	1.30
K	1.000	---	25.40	---

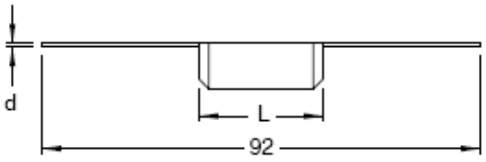
STYLE 1:
PIN 1. CATHODE (POLARITY BAND)
2. ANODE

High Current Funkentstördrossel

RFI Suppression Inductors Entstördrosseln

Series
Bauform

Large (Type 77A) High Current Suppression Coil **77A**



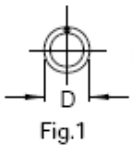
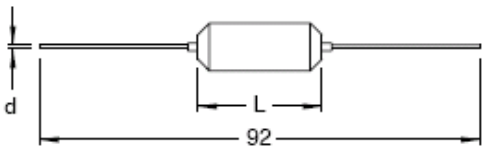


Fig.1





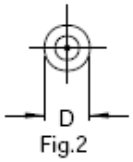
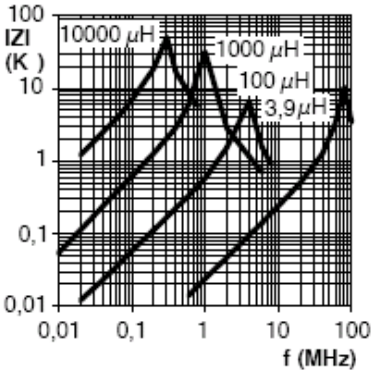


Fig.2





Ordering Code Bestellbezeichnung	Inductance Induktivitäten L (µH)	fL (MHz)	Tol %	Current Nennstrom (A)	Resistance Widerstand ( )	Dimension Abmessungen D x L	 d	Fig.	Material
*77A-3R9M-00	3,9	1,0	±20	12,00	0,009	8x30	1,0	1	Ferrite (Ferrit)
77A-5R6M-01	5,6			8,00	0,017	8x26	0,8	2	
77A-100M-01	10			5,00	0,035				
77A-150M-01	15	0,10		4,00	0,050				
77A-330M-01	33			3,50	0,070	9x26			
77A-680M-01	68			3,00	0,100	10x26			
77A-101M-01	100			2,50	0,150	11x26			
77A-151M-01	150			1,80	0,300				
77A-331M-01	330			1,40	0,500				
77A-681M-01	680			1,00	1,000				
77A-102M-01	1000			0,80	1,500				
77A-152M-01	1500	0,01		0,70	2,000				
77A-332M-01	3300			0,50	4,000				
77A-682M-01	6800			0,35	8,000				
77A-103M-01	10000			0,30	12,000				

Option without insulating material protection (77A/B...)
* Only available in loose packing

Sonderausführung ohne Isolation (77A/B...)
* Kann nur lose geliefert werden

Schaltregler LM2576T

National Semiconductor

June 1999

LM2576/LM2576HV Series
SIMPLE SWITCHER® 3A Step-Down Voltage Regulator

General Description

The LM2576 series of regulators are monolithic integrated circuits that provide all the active functions for a step-down (buck) switching regulator, capable of driving 3A load with excellent line and load regulation. These devices are available in fixed output voltages of 3.3V, 5V, 12V, 15V, and an adjustable output version.

Requiring a minimum number of external components, these regulators are simple to use and include internal frequency compensation and a fixed-frequency oscillator.

The LM2576 series offers a high-efficiency replacement for popular three-terminal linear regulators. It substantially reduces the size of the heat sink, and in some cases no heat sink is required.

A standard series of inductors optimized for use with the LM2576 are available from several different manufacturers. This feature greatly simplifies the design of switch-mode power supplies.

Other features include a guaranteed ±4% tolerance on output voltage within specified input voltages and output load conditions, and ±10% on the oscillator frequency. External shutdown is included, featuring 50 μA (typical) standby current. The output switch includes cycle-by-cycle current limiting, as well as thermal shutdown for full protection under fault conditions.

Features

- 3.3V, 5V, 12V, 15V, and adjustable output versions
- Adjustable version output voltage range, 1.23V to 37V (57V for HV version) ±4% max over line and load conditions
- Guaranteed 3A output current
- Wide input voltage range, 40V up to 60V for HV version
- Requires only 4 external components
- 52 kHz fixed frequency internal oscillator
- TTL shutdown capability, low power standby mode
- High efficiency
- Uses readily available standard inductors
- Thermal shutdown and current limit protection
- P+ Product Enhancement tested

Applications

- Simple high-efficiency step-down (buck) regulator
- Efficient pre-regulator for linear regulators
- On-card switching regulators
- Positive to negative converter (Buck-Boost)

Typical Application (Fixed Output Voltage Versions)

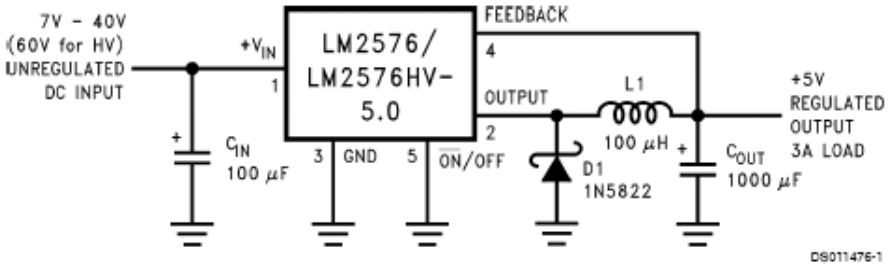


FIGURE 1.

Definition of Terms (Continued)

Most standard aluminum electrolytic capacitors in the 100 µF–1000 µF range have 0.5Ω to 0.1Ω ESR. Higher-grade capacitors ("low-ESR", "high-frequency", or "low-inductance") in the 100 µF–1000 µF range generally have ESR of less than 0.15Ω.

EQUIVALENT SERIES INDUCTANCE (ESL)

The pure inductance component of a capacitor (see *Figure 16*). The amount of inductance is determined to a large extent on the capacitor's construction. In a buck regulator, this unwanted inductance causes voltage spikes to appear on the output.

OUTPUT RIPPLE VOLTAGE

The AC component of the switching regulator's output voltage. It is usually dominated by the output capacitor's ESR multiplied by the inductor's ripple current (ΔI_{IND}). The peak-to-peak value of this sawtooth ripple current can be determined by reading the Inductor Ripple Current section of the Application hints.

CAPACITOR RIPPLE CURRENT

RMS value of the maximum allowable alternating current at which a capacitor can be operated continuously at a specified temperature.

STANDBY QUIESCENT CURRENT (I_{STBY})

Supply current required by the LM2576 when in the standby mode ($\overline{ON}$ /OFF pin is driven to TTL-high voltage, thus turning the output switch OFF).

INDUCTOR RIPPLE CURRENT (ΔI_{IND})

The peak-to-peak value of the inductor current waveform, typically a sawtooth waveform when the regulator is operating in the continuous mode (vs. discontinuous mode).

CONTINUOUS/DISCONTINUOUS MODE OPERATION

Relates to the inductor current. In the continuous mode, the inductor current is always flowing and never drops to zero, vs. the discontinuous mode, where the inductor current drops to zero for a period of time in the normal switching cycle.

INDUCTOR SATURATION

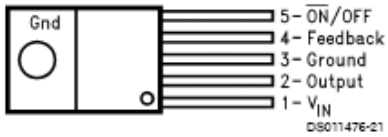
The condition which exists when an inductor cannot hold any more magnetic flux. When an inductor saturates, the inductor appears less inductive and the resistive component dominates. Inductor current is then limited only by the DC resistance of the wire and the available source current.

OPERATING VOLT MICROSECOND CONSTANT ($E \cdot T_{op}$)

The product (in Volt•µs) of the voltage applied to the inductor and the time the voltage is applied. This $E \cdot T_{op}$ constant is a measure of the energy handling capability of an inductor and is dependent upon the type of core, the core area, the number of turns, and the duty cycle.

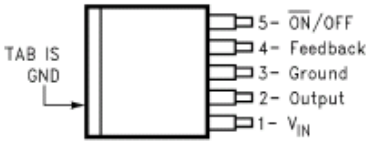
Connection Diagrams (Note 15)

Straight Leads
5-Lead TO-220 (T)
Top View

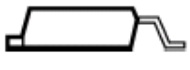


LM2576T-XX or LM2576HVT-XX
NS Package Number T05A

TO-263 (S)
5-Lead Surface-Mount Package
Top View

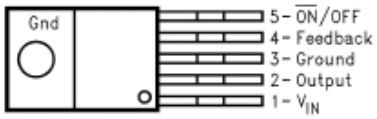


Side View

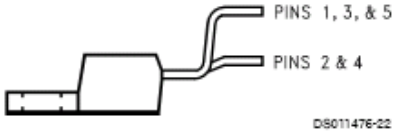


LM2576S-XX or LM2576HVS-XX
NS Package Number TS5B
LM2576SX-XX or LM2576HVSX-XX
NS Package Number TS5B, Tape and Reel

Bent, Staggered Leads
5-Lead TO-220 (T)
Top View



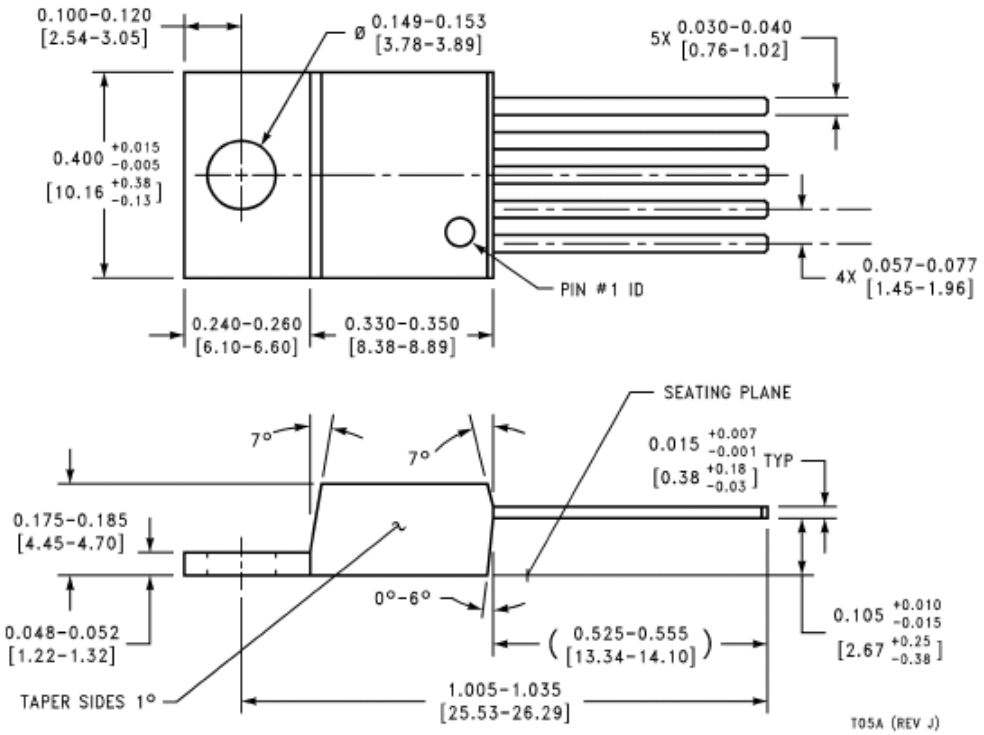
Side View



LM2576T-XX Flow LB03
or LM2576HVT-XX Flow LB03
NS Package Number T05D

Note 15: (XX indicates output voltage option. See ordering information table for complete part number.)

Physical Dimensions inches (millimeters) unless otherwise noted



5-Lead TO-220 (T)
Order Number LM2576T-3.3, LM2576HVT-3.3, LM2576T-5.0, LM2576HVT-5.0, LM2576T-12, LM2576HVT-12, LM2576T-15, LM2576HVT-15, LM2576T-ADJ or LM2576HVT-ADJ
NS Package Number T05A

Schutzdioden:

P6KE6V8A/440A
P6KE6V8CA/440CA

TRANSIL™

FEATURES

- PEAK PULSE POWER : 600 W (10/1000µs)
- BREAKDOWN VOLTAGE RANGE : From 6.8V to 440 V.
- UNI AND BIDIRECTIONAL TYPES.
- LOW CLAMPING FACTOR.
- FAST RESPONSE TIME.
- UL RECOGNIZED.



CB417

DESCRIPTION

Transil diodes provide high overvoltage protection by clamping action. Their instantaneous response to transient overvoltages makes them particularly suited to protect voltage sensitive devices such as MOS Technology and low voltage supplied IC's.

ABSOLUTE MAXIMUM RATINGS (Tamb = 25°C)

Symbol	Parameter		Value	Unit
P_{PP}	Peak pulse power dissipation (see note 1)	Tj initial = T_{amb}	600	W
P	Power dissipation on infinite heatsink	T_{amb} = 75°C	5	W
I_{FSM}	Non repetitive surge peak forward current For Unidirectional types.	Tj initial = T_{amb} tp = 10 ms	100	A
T_{stg} T_j	Storage temperature range Maximum junction temperature		- 65 to + 175 175	°C °C
T_L	Maximum lead temperature for soldering during 10s at 5mm from case		230	°C

Note 1 : For a surge greater than the maximum values, the diode will fail in short-circuit.

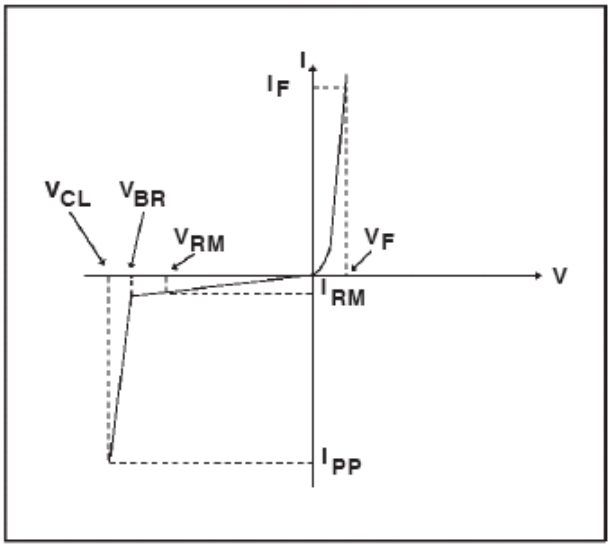
THERMAL RESISTANCES

Symbol	Parameter	Value	Unit
R_{th} (j-l)	Junction-leads	20	°C/W
R_{th} (j-a)	Junction to ambient on printed circuit. L_{lead} = 10 mm	75	°C/W

P6KExx

ELECTRICAL CHARACTERISTICS (Tamb = 25°C)

Symbol	Parameter
V_{RM}	Stand-off voltage.
V_{BR}	Breakdown voltage.
V_{CL}	Clamping voltage.
I_{RM}	Leakage current @ V_{RM}.
I_{PP}	Surge current.
αT	Voltage temperature coefficient.
V_F	Forward Voltage drop.



TYPES		I_{RM} @ V_{RM}		V_{BR} @ I_R				V_{CL} @ I_{PP}		V_{CL} @ I_{PP}		αT	C
		max		min	nom	max note2		max 10/1000µs		max 8/20µs		max note3	typ note4
Unidirectional	**Bidirectional**	**µA**	**V**	**V**	**V**	**V**	**mA**	**V**	**A**	**V**	**A**	**10^{-4}/°C**	**pF**
P6KE6V8A	P6KE6.8CA	1000	5.8	6.45	6.8	7.14	10	10.5	57	13.4	298	5.7	4000
P6KE7V5A	P6KE7.5CA	500	6.4	7.13	7.5	7.88	10	11.3	53	14.5	276	6.1	3700
P6KE10A	P6KE10CA	10	8.55	9.5	10	10.5	1	14.5	41	18.6	215	7.5	2800
P6KE12A	P6KE12CA	5	10.2	11.4	12	12.6	1	16.7	36	21.7	184	7.8	2300
P6KE15A	P6KE15CA	1	12.8	14.3	15	15.8	1	21.2	28	27.2	147	8.4	1900
P6KE18A	P6KE18CA	1	15.3	17.1	18	18.9	1	25.2	24	32.5	123	8.8	1600
P6KE22A	P6KE22CA	1	18.8	20.9	22	23.1	1	30.6	20	39.3	102	9.2	1350
P6KE24A	P6KE24CA	1	20.5	22.8	24	25.2	1	33.2	18	42.8	93	9.4	1250
P6KE27A	P6KE27CA	1	23.1	28.7	27	28.4	1	37.5	16	48.3	83	9.6	1150
P6KE30A	P6KE30CA	1	25.6	28.5	30	31.5	1	41.5	14.5	53.5	75	9.7	1075
P6KE33A	P6KE33CA	1	28.2	31.4	33	24.7	1	45.7	13.1	59	68	9.8	1000
P6KE36A	P6KE36CA	1	30.8	34.2	36	37.8	1	49.9	12	64.3	62	9.9	950
P6KE39A	P6KE39CA	1	33.3	37.1	39	41.0	1	53.9	11.1	69.7	57	10.0	900
P6KE47A	P6KE47CA	1	40.2	44.7	47	49.4	1	64.8	9.3	84	48	10.1	800
P6KE56A	P6KE56CA	1	47.8	53.2	56	58.8	1	77	7.8	100	40	10.3	700
P6KE68A	P6KE68CA	1	58.1	64.6	68	71.4	1	92	6.5	121	33	10.4	625
P6KE82A	P6KE82CA	1	70.1	77.9	82	86.1	1	113	5.3	146	27	10.5	550
P6KE100A	P6KE100CA	1	85.5	95.0	100	105	1	137	4.4	178	22.5	10.6	500
P6KE120A	P6KE120CA	1	102	114	120	126	1	165	3.6	212	19	10.7	450
P6KE150A	P6KE150CA	1	128	143	150	158	1	207	2.9	265	15	10.8	400
P6KE180A	P6KE180CA	1	154	171	180	189	1	246	2.4	317	12.6	10.8	360
P6KE200A	P6KE200CA	1	171	190	200	210	1	274	2.2	353	11.3	10.8	350

TYPES		I_{RM} @ V_{RM} max		V_{BR} @ I_R min nom max note2				V_{CL} @ I_{PP} max 10/1000µs		V_{CL} @ I_{PP} max 8/20µs		αT max note3	C typ note4
Unidirectional	**Bidirectional**	**µA**	**V**	**V**	**V**	**V**	**mA**	**V**	**A**	**V**	**A**	**10^{-4}/°C**	**pF**
P6KE220A	P6KE220CA	1	188	209	220	231	1	328	2	388	10.3	10.8	330
P6KE250A	P6KE250CA	1	213	237	250	263	1	344	2	442	9	11	310
P6KE300A	P6KE300CA	1	256	285	300	315	1	414	1.6	529	7.6	11	290
P6KE350A	P6KE350CA	1	299	332	350	368	1	482	1.6	618	6.5	11	270
P6KE400A	P6KE400CA	1	342	380	400	420	1	548	1.3	706	5.7	11	360
P6KE440A	P6KE440CA	1	376	418	440	462	1	603	1.3	776	5.2	11	350

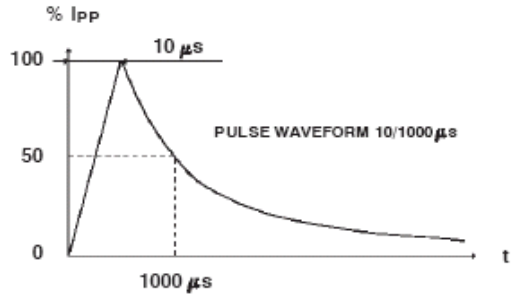
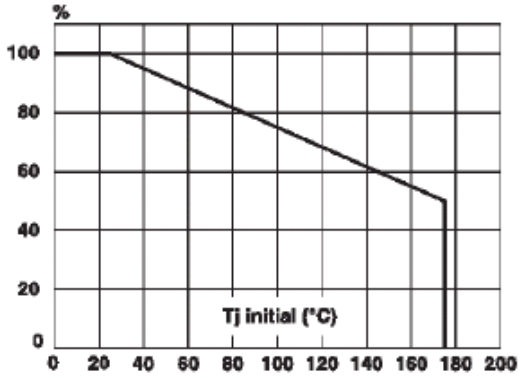


Fig 1: Peak pulse power dissipation versus initial junction temperature (printed circuit board).



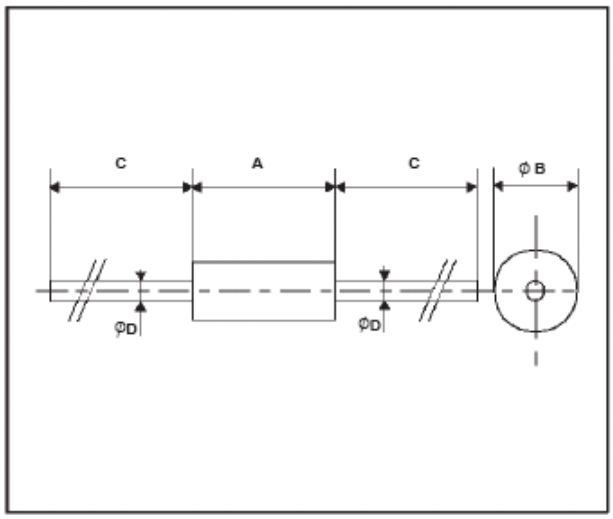
Note 2 : Pulse test : tp < 50 ms.

Note 3 : $\Delta V_{BR} = \alpha T * (Ta - 25) * V_{BR}(25°C)$.

Note 4 $V_R = 0$ V, F = 1 MHz. For bidirectional types, capacitance value is divided by 2.

PACKAGE MECHANICAL DATA

CB417 (Plastic)



REF.	DIMENSIONS			
	Millimetres		Inches	
	Min.	Max.	Min.	Max.
A		8.89		0.350
B		3.683		0.145
C	25.4		1.000	
D		1.092		0.043

Packaging : standard packaging is in tape and reel.

Weight = 0.65 g.

BC 557 A:

PNP general purpose transistors BC556; BC557

FEATURES

- Low current (max. 100 mA)
- Low voltage (max. 65 V).

APPLICATIONS

- General purpose switching and amplification.

DESCRIPTION

PNP transistor in a TO-92; SOT54 plastic package.
NPN complements: BC546 and BC547.

PINNING

PIN	DESCRIPTION
1	emitter
2	base
3	collector

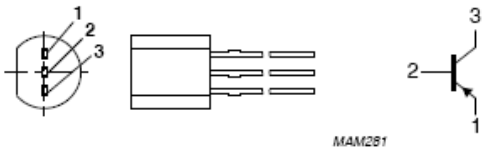


Fig.1 Simplified outline (TO-92; SOT54) and symbol.

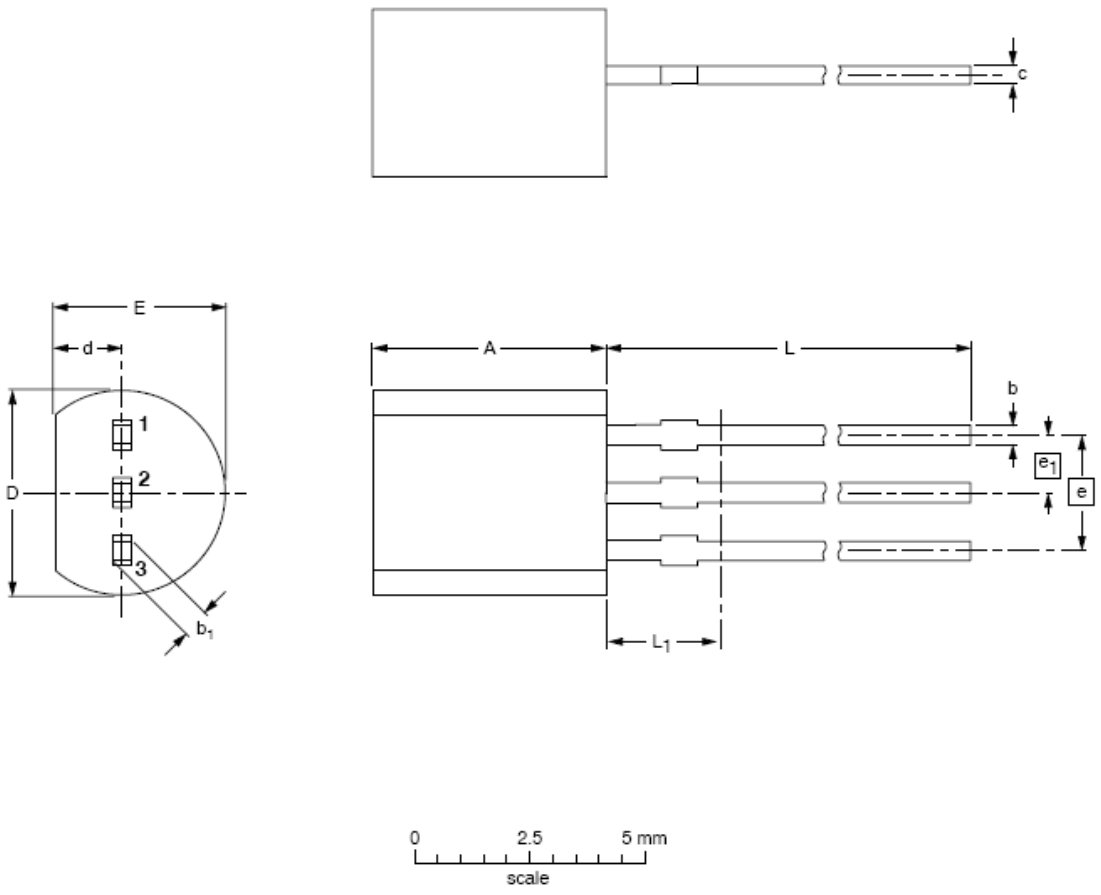
LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V_{CBO}	collector-base voltage	open emitter			
	BC556		–	–80	V
	BC557		–	–50	V
V_{CEO}	collector-emitter voltage	open base			
	BC556		–	–65	V
	BC557		–	–45	V
V_{EBO}	emitter-base voltage	open collector	–	–5	V
I_C	collector current (DC)		–	–100	mA
I_{CM}	peak collector current		–	–200	mA
I_{BM}	peak base current		–	–200	mA
P_{tot}	total power dissipation	$T_{amb} \leq 25$ °C	–	500	mW
T_{stg}	storage temperature		–65	+150	°C
T_j	junction temperature		–	150	°C
T_{amb}	operating ambient temperature		–65	+150	°C

PACKAGE OUTLINE

Plastic single-ended leaded (through hole) package; 3 leads **SOT54**



DIMENSIONS (mm are the original dimensions)

UNIT	A	b	b_1	c	D	d	E	e	e_1	L	L_1(1)
mm	5.2 5.0	0.48 0.40	0.66 0.56	0.45 0.40	4.8 4.4	1.7 1.4	4.2 3.6	2.54	1.27	14.5 12.7	2.5

Note

1. Terminal dimensions within this zone are uncontrolled to allow for flow of plastic and terminal irregularities.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT54		TO-92	SC-43			97-02-28

Festspannungsregler LM 2936 Z-5,0:

National Semiconductor

March 2001

LM2936-5.0
Ultra-Low Quiescent Current 5V Regulator

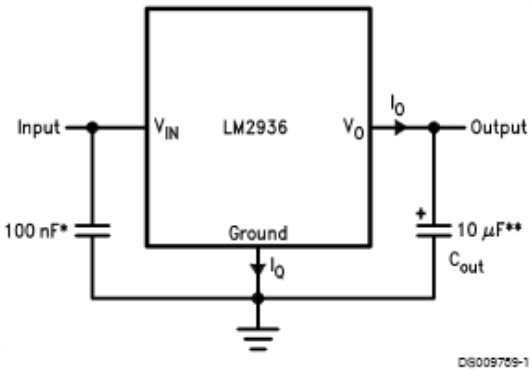
General Description

The LM2936 ultra-low quiescent current regulator features low dropout voltage and low current in the standby mode. With less than 15 μA quiescent current at a 100 μA load, the LM2936 is ideally suited for automotive and other battery operated systems. The LM2936 retains all of the features that are common to low dropout regulators including a low dropout PNP pass device, short circuit protection, reverse battery protection, and thermal shutdown. The LM2936 has a 40V maximum operating voltage limit, a −40˚C to +125˚C operating temperature range, and ±3% output voltage tolerance over the entire output current, input voltage, and temperature range. The LM2936 is available in a TO-92 package, a SO-8 surface mount package, and a TO-252 surface mount power package.

Features

- Ultra low quiescent current ($I_Q \leq 15\ \mu A$ for $I_O = 100\ \mu A$)
- Fixed 5V, 50 mA output
- ±2% Initial output tolerance
- ±3% Output tolerance over line, load, and temperature
- Dropout voltage typically 200 mV @ I_O = 50 mA
- Reverse battery protection
- −50V reverse transient protection
- Internal short circuit current limit
- Internal thermal shutdown protection
- 40V operating voltage limit
- Shutdown Pin available with LM2936BM package

Typical Application

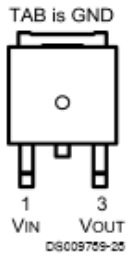


* Required if regulator is located more than 2" from power supply filter capacitor.

** Required for stability. Must be rated for 10 μF minimum over intended operating temperature range. Effective series resistance (ESR) is critical, see curve. Locate capacitor as close as possible to the regulator output and ground pins. Capacitance may be increased without bound.

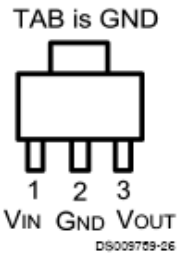
Connection Diagrams

TO-252



Top View
Order Number LM2936DT-5.0
See NS Package Number TD03B

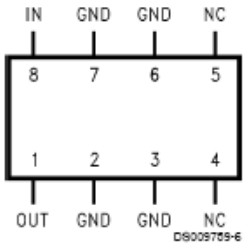
SOT-223



Top View
Order Number LM2936MP-5.0
See NS Package Number MA04A

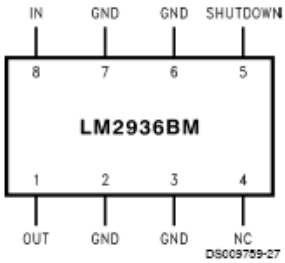
Connection Diagrams (Continued)

8-Pin SO (M)



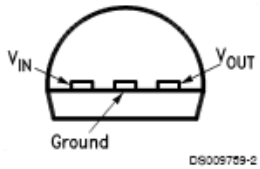
Top View
Order Number LM2936M-5.0
See NS Package Number M08A

8-Pin SO (M)



Top View
Order Number LM2936BM-5.0
See NS Package Number M08A

TO-92



Bottom View
Order Number LM2936Z-5.0
See NS Package Number Z03A

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Input Voltage (Survival)	+60V, −50V
ESD Susceptibility (Note 2)	2000V
Power Dissipation (Note 3)	Internally limited
Junction Temperature (T_{Jmax})	150˚C
Storage Temperature Range	−65˚C to +150˚C
Lead Temperature (Soldering, 10 sec.)	260˚C

Operating Ratings

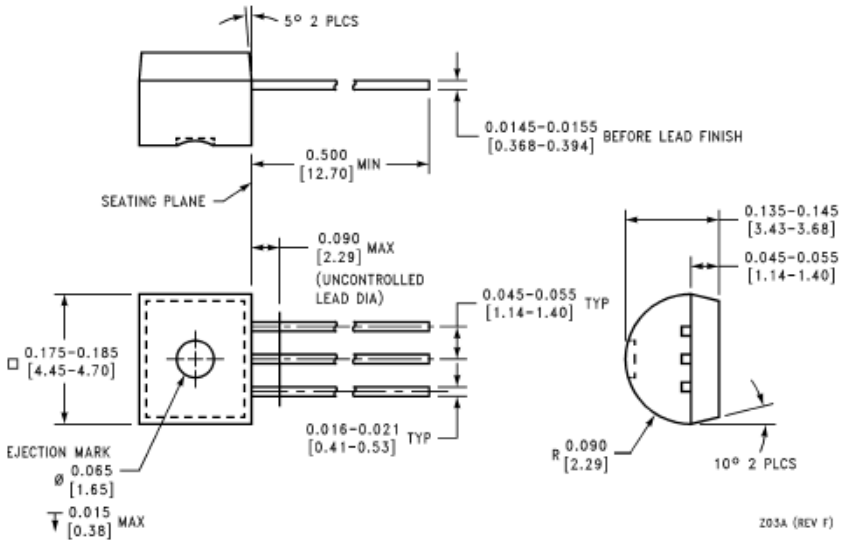
Operating Temperature Range	−40˚C to +125˚C
Maximum Input Voltage (Operational)	40V
Maximum Shutdown Pin Voltage	0V to 40V
TO-92 (Z03A) θ_{JA}	195˚C/W
SO-8 (M08A) θ_{JA}	140˚C/W
SO-8 (M08A) θ_{JC}	45˚C/W
TO-252 (TD03B) θ_{JA}	136˚C/W
TO-252 (TD03B) θ_{JC}	6˚C/W
SOT-223 (MA04A) θ_{JA}	149˚C/W
SOT-223 (MA04A) θ_{JC}	36˚C/W

Electrical Characteristics

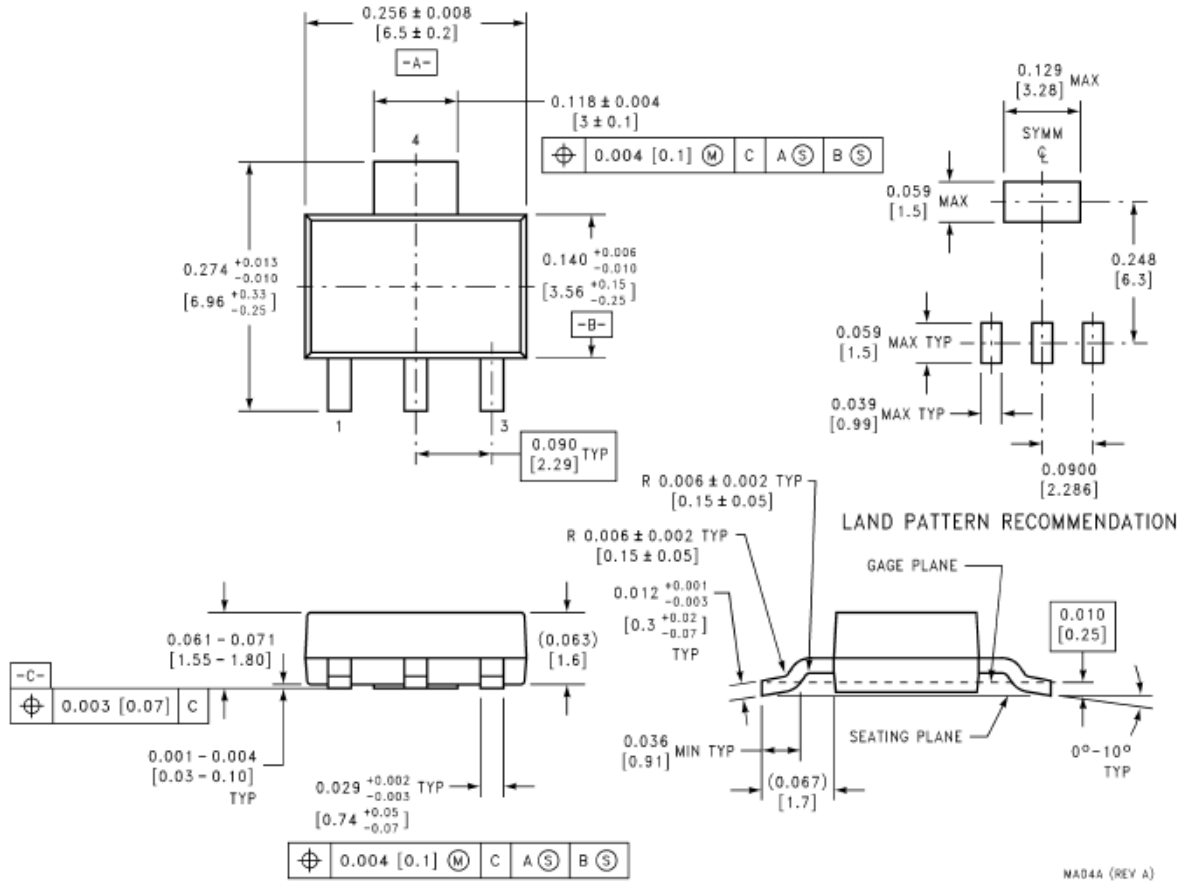
V_{IN} = 14V, I_O = 10 mA, T_J = 25˚C, unless otherwise specified. **Boldface** limits apply over entire operating temperature range

Parameter	Conditions	Min (Note 5)	Typical (Note 4)	Max (Note 5)	Units
Output Voltage		4.900	5.000	5.100	V
	5.5V ≤ V_{IN} ≤ 26V, 100 μA ≤ I_O ≤ 50 mA (Note 6)	**4.85**	**5.00**	**5.15**	
Quiescent Current	I_O = 100 μA, 8V ≤ V_{IN} ≤ 24V		9	15	μA
	I_O = 10 mA, 8V ≤ V_{IN} ≤ 24V		0.20	0.50	mA
	I_O = 50 mA, 8V ≤ V_{IN} ≤ 24V		1.5	2.5	mA
Line Regulation	9V ≤ V_{IN} ≤ 16V		5	10	mV
	6V ≤ V_{IN} ≤ 40V, I_O = 1 mA		10	30	
Load Regulation	100 μA ≤ I_O ≤ 5 mA		10	30	mV
	5 mA ≤ I_O ≤ 50 mA		10	30	
Dropout Voltage	I_O = 100 μA		0.05	0.10	V
	I_O = 50 mA		0.20	0.40	V
Short Circuit Current	V_O = 0V	65	120	250	mA
Output Impedance	I_O = 30 mAdc and 10 mArms, $_f$ = 1000 Hz		450		mΩ
Output Noise Voltage	10 Hz–100 kHz		500		μV
Long Term Stability			20		mV/1000 Hr
Ripple Rejection	V_{ripple} = 1V_{rms}, $_{fripple}$ = 120 Hz	−40	−60		dB
Reverse Polarity DC Input Voltage	R_L = 500Ω, V_O ≥ −0.3V	−15			V
Reverse Polarity Transient Input Voltage	R_L = 500Ω, T = 1 ms	−50	−80		V
Output Leakage with Reverse Polarity Input	V_{IN} = −15V, R_L = 500Ω		−0.1	−600	μA
Maximum Line Transient	R_L = 500Ω, V_O ≤ 5.5V, T = 40ms	60			V
Shutdown Input – LM2936BM Only					
Output Voltage, V_{OUT}	Output Off, V_{SD}=2.4V, R_{LOAD}=499Ohms		0	0.010	V
Shutdown High Threshold Voltage, V_{IH}	Output Off, R_{LOAD}=499Ohms	2.00	1.1		V
Shutdown Low Threshold Voltage, V_{IL}	Output On, R_{LOAD}=499Ohms		1.1	0.60	V
Shutdown High Current, I_{IH}	Output Off, V_{SD}=2.4V, R_{LOAD}=499Ohms		12		μA

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



3-Lead TO-92 Plastic Package (Z)
NS Package Number Z03A



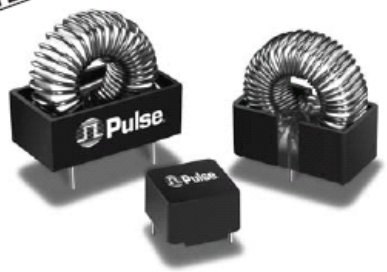
SOT-223 Package (MP)
NS Package Number MA04A

PE-52626 Schaltregler-Induktivität:

INDUCTORS DESIGNED FOR NATIONAL'S 50 kHz SIMPLE SWITCHER™

Pulse® A TECHNITROL COMPANY

RoHS COMPLIANT



- Designed for use with National's device numbers LM2574/LM2575/LM2576
- Base material meets flammability requirements of UL 94V-0
- Performance verified by National Semiconductor

Electrical Specifications @ 25°C— Operating Temperature -40° to +130° C

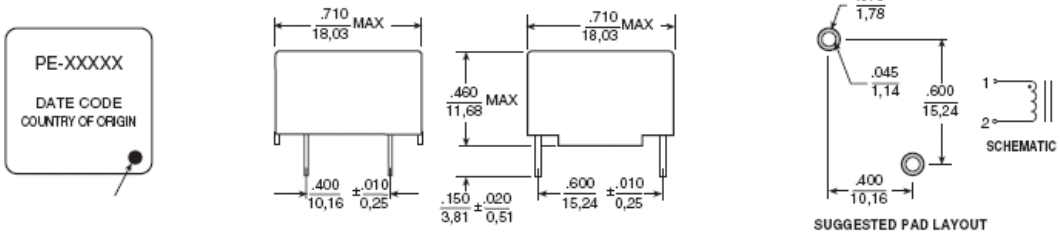
Part Identification		Reference Operating Values[1]			Design Control Values			
Part Number	Inductor Code	Inductance Typical (µH)	IDC (Amps)	ETOP (V-µSec)	Inductance No DC[2] (µH ± 20%)	DCR (Ω MAX)	Package Style	Lead Diameter
PE-52626	L220	220	1.4	90	230	0.38	Low Profile	.025 SQ.

NOTES:

1. Typical inductance occurs at the IDC and ETOP values shown.
2. The control value of inductance is measured at BOP equal to or less than 10 gauss (10 mV @ 20 kHz) without DC current.
3. Inductance decreases with higher values of DC current and increases with lower values of DC current.
4. Inductance increases with increase in BOP or ETOP.
5. SIMPLE SWITCHER™ is a trademark of National Semiconductor Corporation.
6. RoHS compliant parts are available. Order RoHS compliant parts by adding the suffix "**NL**" to the part number (i.e. PE-53112 becomes PE-53112**NL**).

Mechanicals

Low Profile Series



For More Information:

Pulse Worldwide Headquarters
12220 World Trade Drive
San Diego, CA 92128
U.S.A.
www.pulseeng.com
TEL: 858 674 8100
FAX: 858 674 8262

Pulse Northern Europe
3 Huxley Road
Surrey Research Park
Guildford, Surrey GU2 5RE
United Kingdom
TEL: 44 1483 401700
FAX: 44 1483 401701

Pulse Southern Europe
Zone Industrielle
F-39270
Orgelet
France
TEL: 33 3 84 35 04 04
FAX: 33 3 84 25 46 41

Pulse China Headquarters
No. 1
Industrial District
Changan, Dongguan
China
TEL: 86 769 5538070
FAX: 86 769 5538870

Pulse North China
Room 1002
No. 819
Nanjing West Rd
Shanghai
China
TEL: 86 21 32181071
FAX: 86 21 32181396

Pulse South Asia
150 Kampong Ampat
#07-01/02
KA Centre
Singapore 368324
TEL: 65 6287 8998
FAX: 65 6280 0080

Pulse North Asia
3F-4, No. 81, Sec. 1
Hsin Tai Wu Road
Hsi-Chih
Taipei Hsien
Taiwan
TEL: 886 2 26980228
FAX: 886 2 26980948

Performance warranty of products offered on this data sheet is limited to the parameters specified. Data is subject to change without notice. Other brand and product names mentioned herein may be trademarks or registered trademarks of their respective owners.
© Copyright, 2005. Pulse Engineering, Inc. All right reserved.

www.pulseeng.com P596.B (10/05

MAX238:

19-4323; Rev 11; 2/03

MAXIM

+5V-Powered, Multichannel RS-232 Drivers/Receivers

General Description

The MAX220–MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where ±12V is not available.

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than 5µW. The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

Applications

Portable Computers

Low-Power Modems

Interface Translation

Battery-Powered RS-232 Systems

Multidrop RS-232 Networks

Features

Superior to Bipolar

- **Operate from Single +5V Power Supply (+5V and +12V—MAX231/MAX239)**
- **Low-Power Receive Mode in Shutdown (MAX223/MAX242)**
- **Meet All EIA/TIA-232E and V.28 Specifications**
- **Multiple Drivers and Receivers**
- **3-State Driver and Receiver Outputs**
- **Open-Line Detection (MAX243)**

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX220CPE	0°C to +70°C	16 Plastic DIP
MAX220CSE	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220C/D	0°C to +70°C	Dice*
MAX220EPE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

Ordering Information continued at end of data sheet.

**Contact factory for dice specifications.*

Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value (µF)	SHDN & Three-State	Rx Active in SHDN	Data Rate (kbps)	Features
MAX220	+5	2/2	4	0.1	No	—	120	Ultra-low-power, industry-standard pinout
MAX222	+5	2/2	4	0.1	Yes	—	200	Low-power shutdown
MAX223 (MAX213)	+5	4/5	4	1.0 (0.1)	Yes	✔	120	MAX241 and receivers active in shutdown
MAX225	+5	5/5	0	—	Yes	✔	120	Available in SO
MAX230 (MAX200)	+5	5/0	4	1.0 (0.1)	Yes	—	120	5 drivers with shutdown
MAX231 (MAX201)	+5 and +7.5 to +13.2	2/2	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; same functions as MAX232
MAX232 (MAX202)	+5	2/2	4	1.0 (0.1)	No	—	120 (64)	Industry standard
MAX232A	+5	2/2	4	0.1	No	—	200	Higher slew rate, small caps
MAX233 (MAX203)	+5	2/2	0	—	No	—	120	No external caps
MAX233A	+5	2/2	0	—	No	—	200	No external caps, high slew rate
MAX234 (MAX204)	+5	4/0	4	1.0 (0.1)	No	—	120	Replaces 1488
MAX235 (MAX205)	+5	5/5	0	—	Yes	—	120	No external caps
MAX236 (MAX206)	+5	4/3	4	1.0 (0.1)	Yes	—	120	Shutdown, three state
MAX237 (MAX207)	+5	5/3	4	1.0 (0.1)	No	—	120	Complements IBM PC serial port
MAX238 (MAX208)	+5	4/4	4	1.0 (0.1)	No	—	120	Replaces 1488 and 1489
MAX239 (MAX209)	+5 and +7.5 to +13.2	3/5	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; single-package solution for IBM PC serial port
MAX240	+5	5/5	4	1.0	Yes	—	120	DIP or flatpack package
MAX241 (MAX211)	+5	4/5	4	1.0 (0.1)	Yes	—	120	Complete IBM PC serial port
MAX242	+5	2/2	4	0.1	Yes	✔	200	Separate shutdown and enable
MAX243	+5	2/2	4	0.1	No	—	200	Open-line detection simplifies cabling
MAX244	+5	8/10	4	1.0	No	—	120	High slew rate
MAX245	+5	8/10	0	—	Yes	✔	120	High slew rate, int. caps, two shutdown modes
MAX246	+5	8/10	0	—	Yes	✔	120	High slew rate, int. caps, three shutdown modes
MAX247	+5	8/9	0	—	Yes	✔	120	High slew rate, int. caps, nine operating modes
MAX248	+5	8/8	4	1.0	Yes	✔	120	High slew rate, selective half-chip enables
MAX249	+5	6/10	4	1.0	Yes	✔	120	Available in quad flatpack package

ABSOLUTE MAXIMUM RATINGS—MAX223/MAX230–MAX241

V_{CC} .....-0.3V to +6V
V+ .....(V_{CC} - 0.3V) to +14V
V- .....+0.3V to -14V
Input Voltages
T_{IN} .....-0.3V to (V_{CC} + 0.3V)
R_{IN}.....±30V
Output Voltages
T_{OUT} .....(V+ + 0.3V) to (V- - 0.3V)
R_{OUT}.....-0.3V to (V_{CC} + 0.3V)
Short-Circuit Duration, T_{OUT} .....Continuous
Continuous Power Dissipation (T_A = +70°C)
14-Pin Plastic DIP (derate 10.00mW/°C above +70°C)....800mW
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C)....842mW
20-Pin Plastic DIP (derate 11.11mW/°C above +70°C)....889mW
24-Pin Narrow Plastic DIP
(derate 13.33mW/°C above +70°C) 1.07W
24-Pin Plastic DIP (derate 9.09mW/°C above +70°C).....500mW
16-Pin Wide SO (derate 9.52mW/°C above +70°C).....762mW
20-Pin Wide SO (derate 10 00mW/°C above +70°C).....800mW
24-Pin Wide SO (derate 11.76mW/°C above +70°C).....941mW
28-Pin Wide SO (derate 12.50mW/°C above +70°C) 1W
44-Pin Plastic FP (derate 11.11mW/°C above +70°C).....889mW
14-Pin CERDIP (derate 9.09mW/°C above +70°C).....727mW
16-Pin CERDIP (derate 10.00mW/°C above +70°C).....800mW
20-Pin CERDIP (derate 11.11mW/°C above +70°C).....889mW
24-Pin Narrow CERDIP
(derate 12.50mW/°C above +70°C) 1W
24-Pin Sidebraze (derate 20.0mW/°C above +70°C).....1.6W
28-Pin SSOP (derate 9.52mW/°C above +70°C).....762mW
Operating Temperature Ranges
MAX2 _ _ C _ _.....0°C to +70°C
MAX2 _ _ E _ _.....-40°C to +85°C
MAX2 _ _ M _ _ .....-55°C to +125°C
Storage Temperature Range .....-65°C to +160°C
Lead Temperature (soldering, 10s) .....+300°C

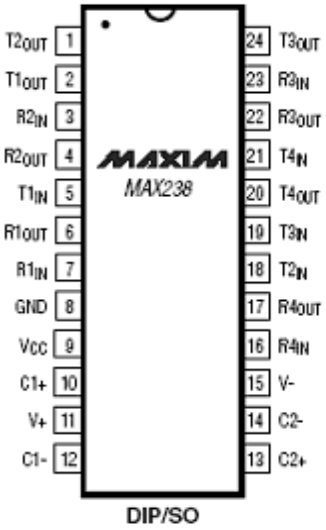
Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

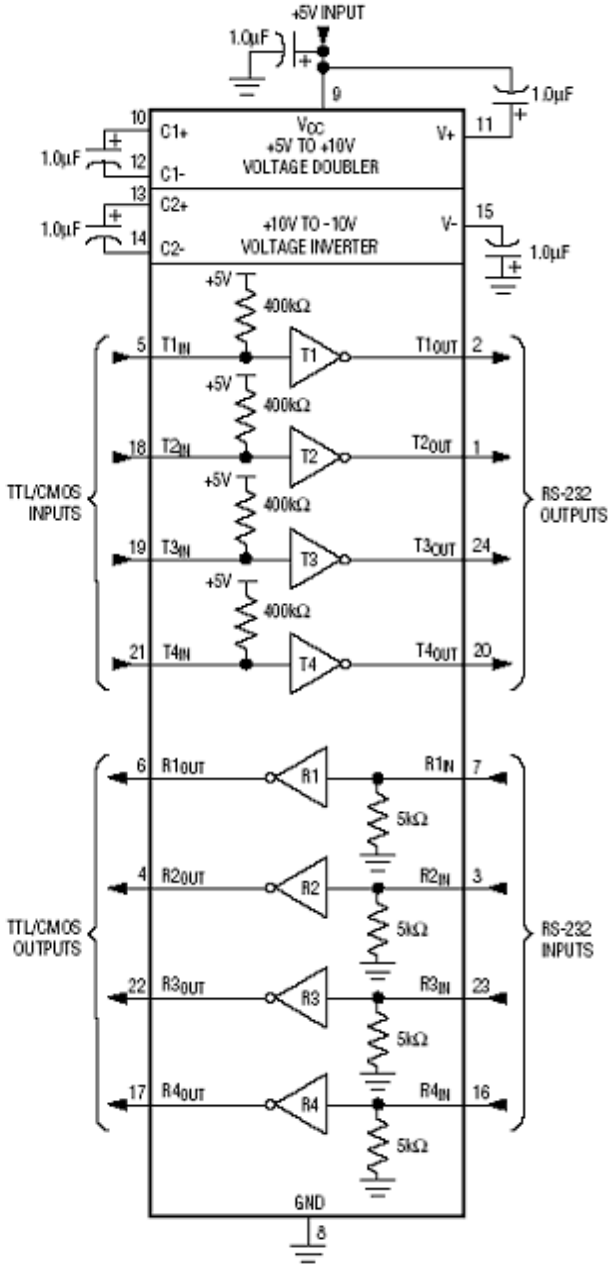
ELECTRICAL CHARACTERISTICS—MAX223/MAX230–MAX241

(MAX223/230/232/234/236/237/238/240/241, V_{CC} = +5V ±10; MAX233/MAX235, V_{CC} = 5V ±5%, C1–C4 = 1.0µF; MAX231/MAX239, V_{CC} = 5V ±10%; V+ = 7.5V to 13.2V; T_A = T_{MIN} to T_{MAX}; unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to ground		±5.0	±7.3		V
V_{CC} Power-Supply Current	No load, T_A = +25°C	MAX232/233		5	10	mA
		MAX223/230/234–238/240/241		7	15	
		MAX231/239		0.4	1	
V+ Power-Supply Current		MAX231		1.8	5	mA
		MAX239		5	15	
Shutdown Supply Current	T_A = +25°C	MAX223		15	50	µA
		MAX230/235/236/240/241		1	10	
Input Logic Threshold Low	T_{IN}; EN, $\overline{SHDN}$ (MAX233); $\overline{EN}$, SHDN (MAX230/235–241)				0.8	V
Input Logic Threshold High	T_{IN}		2.0			V
	EN, $\overline{SHDN}$ (MAX223); $\overline{EN}$, SHDN (MAX230/235/236/240/241)		2.4			
Logic Pull-Up Current	T_{IN} = 0V			1.5	200	µA
Receiver Input Voltage Operating Range			-30		30	V

TOP VIEW





RTC – Reald-Time-Clock DS 1307:

DALLAS SEMICONDUCTOR

DS1307/DS1308
64 X 8 Serial Real Time Clock

www.dalsemi.com

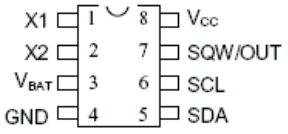
FEATURES

- Real time clock counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap year compensation valid up to 2100
- 56 byte nonvolatile RAM for data storage
- 2-wire serial interface
- Programmable squarewave output signal
- Automatic power-fail detect and switch circuitry
- Consumes less than 500 nA in battery backup mode with oscillator running
- Optional industrial temperature range -40°C to +85°C (IND) available for DS1307 and DS1308
- DS1307 available in 8-pin DIP or SOIC
- DS1308 available in 36-pin SMD BGA (Ball Grid Array)
- DS1308 accuracy is better than ±2 minute/month at 25°C
- Recognized by Underwriters Laboratory

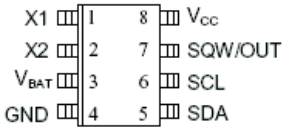
ORDERING INFORMATION

DS1307	8-Pin DIP
DS1307Z	8-Pin SOIC (150 mil)
DS1307N	8-Pin DIP (Industrial)
DS1307ZN	8-Pin SOIC (Industrial)
DS1308	36-Pin BGA
DS1308N	36-Pin BGA (Industrial)

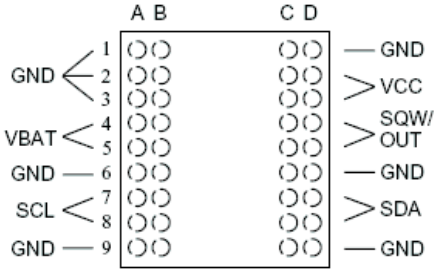
PIN ASSIGNMENT



DS1307 8-Pin DIP (300 mil)



DS1307Z 8-Pin SOIC (150 mil)



DS1308 36-Pin SMD BGA
(Top View)

PIN DESCRIPTION DS1307/DS1308

V_{CC}	- Primary Power Supply
X1, X2	- 32.768 kHz Crystal Connection
V_{BAT}	- +3V Battery Input
GND	- Ground
SDA	- Serial Data
SCL	- Serial Clock
SQW/OUT	- Square wave/Output Driver

DS1308 PIN IDENTIFIER

V_{CC}	- C2, C3, D2, D3
V_{BAT}	- A4, A5, B4, B5
SDA	- C7, C8, D7, D8
SCL	- A7, A8, B7, B8
SQW/OUT	- C4, C5, D4, D5
GND	- All Remaining Balls

DESCRIPTION

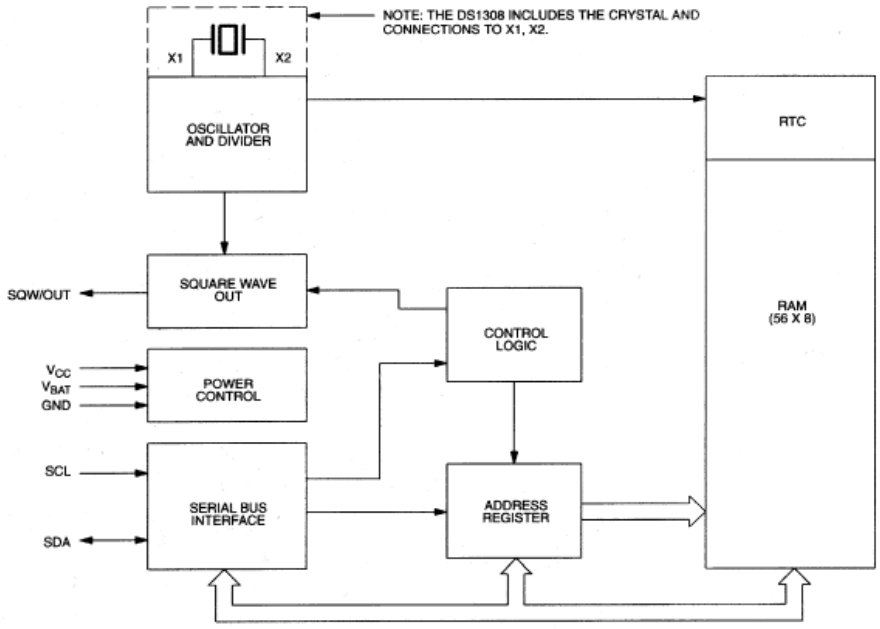
The DS1307 Serial Real Time Clock is a low power, full BCD clock/calendar plus 56 bytes of nonvolatile SRAM. Address and data are transferred serially via a 2-wire bi-directional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with less than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power sense circuit which detects power failures and automatically switches to the battery supply.

The DS1308 incorporates the DS1307 chip with a 32.768 kHz crystal in a surface mountable, 36-pin ball grid array package (BGA). The close proximity of the embedded crystal to the high impedance crystal input pins on the DS1307 minimizes capacitive loading and noise injection problems associated with many other oscillator designs. The total area required for installation is less than that of one United States dime: thus, minimizing PCB space required.

OPERATION

The DS1307/1308 operates as a slave device on the serial bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When V_{CC} falls below 1.25 x V_{BAT} the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out of tolerance system. When V_{CC} falls below V_{BAT} the device switches into a low current battery backup mode. Upon power up, the device switches from battery to V_{CC} when V_{CC} is greater than V_{BAT} +0.2V and recognizes inputs when V_{CC} is greater than 1.25 x V_{BAT}. The block diagram in Figure 1 shows the main elements of the Serial Real Time Clock.

DS1307/DS1308 BLOCK DIAGRAM Figure 1



15.1 Evertool

Evertool is an AVRISP/STK500-protocol and JTAGICE compatible Programmer/JTAG debugger.

- ISP Programmer compatible with Atmel AVRISP, directly accessable with AVRStudio and avrdude
- JTAG debugger compatible with Atmel JTAGICE, directly accessable with AVRStudio and AVaRice
- Evertool supports all AVR devices Atmel supports with their AVRISP and JTAGICE.
- Easy update of the programmer for new target AVRs via RS232/serial port
- Rescue clock generaration to recover from wrong fuse-bit-settings
- Simple hardware, standard parts

15.1.1 Intro

The ATMEL AVRISP In-System Programmer is a small device to program ATMEL AVR microcontrollers. The original ATMEL AVRISP device is described here. To avoid misunderstandings: ATMEL's AVRISP is a "small STK500". Evertool which emulates an AVRISP is not a "bit-bang-programmer" or an AVR910-type-programmer. Evertool "talks" STK500-protocol on the RS232-port since the STK500-firmware ATMEL delivers with the AVRStudio is used. From the programming interface the AVRISP is compatible with the Atmel STK500 in low-voltage serial programming mode.

The ATMEL JTAGICE is a device to program and debug ATMEL AVR microcontrollers with JTAG-Interface (ATmega16, ATmega32, ATmega64, ATmega128...) The ATMEL JTAGICE is described here:
http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2737

This is a rebuild of both devices on a single board. The main advantage of such a programmer is the compatibility with the ATMEL AVRstudio since it uses the original ATMEL firmware code supplied with AVRStudio on an ATmega16 (JTAG) and ATmega8535 (ISP). So the front-end software (AVRStudio) and programmer firmware are supported by the makers of the target-controllers and new devices should be supported as soon as a new AVRStudio gets released with the new firmware-files for STK500/AVRISP or JTAGICE. With the AVRStudio AVRISP/STK500-plug-in programming of flash, eeprom, fuse- and lockbits is an easy task. The JTAG part of Evertool is compatible with AVRstudio's JTAGICE plug-in. And yes, it is possible "to jtag" the AVR Butterfly with Evertool.

Before you start you may concider this:

- The original ATMEL AVRISP device is cheap (prices May 2004: ca. 40 EUR incl. Tax at www.reichelt.de, ca. 26 EUR+TAX at de.digi-key.com). So if you don't need the JTAG-part this universal programmer may not be worth the effort.
- The presented device has no ESD-protection IC or level-shifters on board. Refer to the schematic of the original ATMEL AVRISP device.(Sorry, the schematic of the Atmel JTAGICE is not available here.) But since the AVR IO-pins have a rather good internal protection this should not be a problem.

  http://www.atmel.com/dyn/resources/prod_documents/Avrisp_Sch.pdf
- You need a programmer to build this programmer. A simple STK200-type dongle will do. Refer to the lancos.com PonyProg page which describes a simple programming interface. The dongle is only needed for 10 minutes so you may borrow one from a friend. Another AVRISP, an AVR910-type programmer or a STK500 will do too, of cause. The programmer is just needed to program the special Evertool bootloader into the ATmega16 once. Updating of Evertool to new firmware(s) is done with the Evertool-bootloader - no additional progammer needed.

  http://www.lancos.com/prog.html
- If you are just interested in the JTAG-part: Rainer Radow made a board based on the Evertool JTAG-section. This board only emulates the JTAGICE. More information on "Evertool-Light" further down on this page.

  http://www.siwawi.arubi.uni-kl.de/avr_projects/evertool/index.html#etlight

- If you are just interested in the AVRISP-part (although the original by ATMEL is cheap): see the ispRE page.

  http://www.siwawi.arubi.uni-kl.de/avr_projects/avrispre/index.html

Evertool has some extra functions:

- Rescue mode, so AVR controllers where the clock fuse bits have been set incorrectly can be reseted and reprogrammed
- Power-supply to the target AVR-controller-board either from a regulated source or from the onboard voltage regulator (3,3V and 5V).
- Reset Button for target AVR (connected to the AVRISP-part)

15.1.2 Changelog

- 29. Jun. 2004 - updated schematic, picture gallery.
- 30. Jun. 2004 - added "Evertool-light"
- 16. Sept. 2004 - added remark about "L"-type AVRs
- 20. Sept. 2004 - updated schematic (power-supply), Hardware Version 0.9
- 14. Jan. 2005 - updated Evertool schematic (JTAG VTG voltage-Divider). Thanks to Martin Kohout for spotting this.
- 28. Jan. 2005 - Remark about ATmega8535 chip-erase problem with bootloader version 0.2 in section "Software".
- 11. Feb. 2005 - Updated Bootloader binary (version 0.7)
- 7. Mar. 2005 - Remark about avrdude and STK500/AVRISP Firmware 2 in section "operation"
- 19. Apr. 2005 - moved Pull-up from PD4 to PD6. Thanks to Björn B.
- 17. Dec. 2005 - A current-limiting resistor (around 200 Ohm) should be placed between the Target-Reset-Button and GND (Schematic will be updated soon). Thanks to Dirk Büttner
- 24. Jan. 2006 - Information about how to program the AVR fuses with Ponyprog. Provided by Jeff Havens. Thank you, Jeff.
- Feb. 2006 - Added Step-by-Step guide for softare-installation on Evertool-LIGHT
- 8. Sept. 2006 - Evertool-based device with USB by Florian Schenk

15.1.3 License

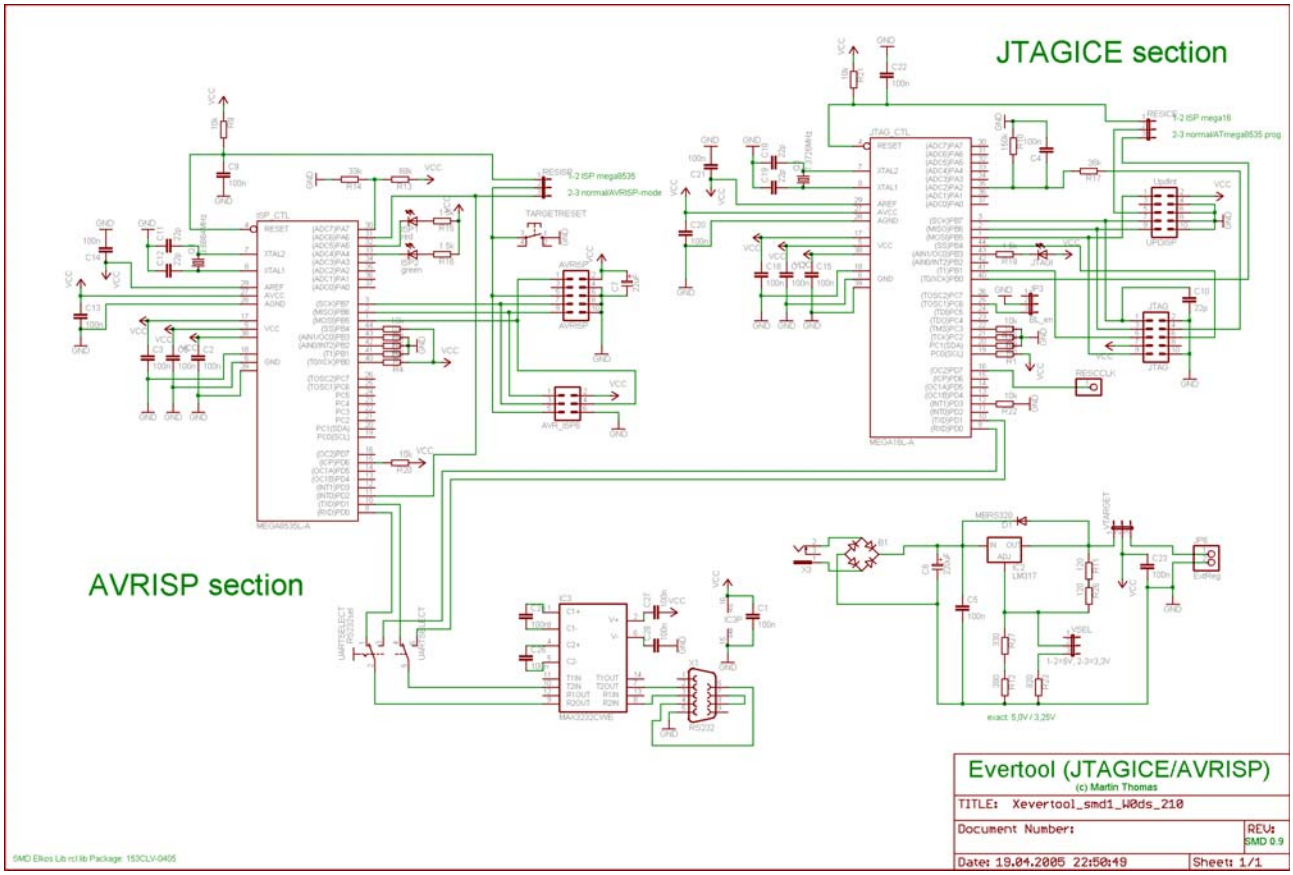
- The schematic is as free as ATMEL made their schematics freely available.
- The layout is copyrighted. Free for non-commercial use. For commercial use please contact me by e-mail.
- The bootloader is also copyrighted. Free for non-commercial use. For commercial use please contact me by e-mail.
- It's also free to build yourself an Evertool for your work and just **use** the tool for programming/debugging.

Here commercial use is meant as: building and/or selling devices with the Evertool PCB-Layout with or without a preinstalled Evertool-Bootloader or selling the Evertool-Bootloader itself without hardware (as binary, source-code or programmed in a controller). Or just simply: If you earn money with my work, you owe me a part of your profit.

15.1.4 Hardware

Schematic:

Evertool V 0.9 Schematic



Schaltbild 87: Evertool (JTAGICE/AVRISP)

Board:

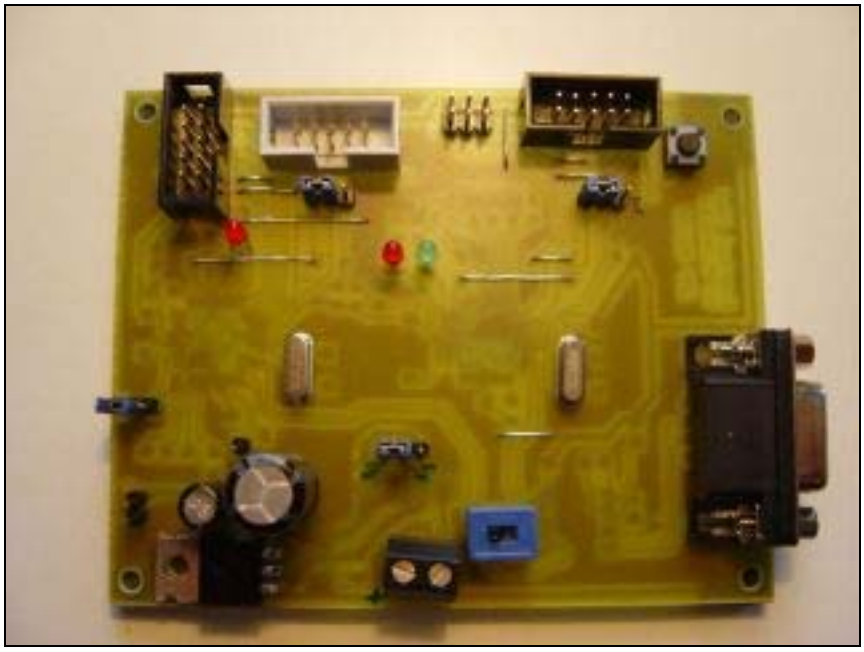


Abbildung 207: Evertool V 0.6 JMP Board

Remarks:
In this layout all "thru-hole"-parts are located on the backside. So a one-sided PCB with some "wire-bridges" will do. Still around 100 holes to drill...
Florian Schenk has created a device based on the Evertool-design with an USB-Interface (FTDI232-IC) and created a layout. See below.

Main Parts:
- AVRISP-section
  - ATmega8535 or AT90S8535
  - MAX(3)232 (3,3V Version preferred)
  - 2*5 pin-header
  - SUBD-9 Socket
- JTAGICE-section
  - ATmega16
  - 2 2*5 pin-headers

During development:
- any ISP capable AVR programmer (STK200-dongle, AVRISP, AVR910, STK500)

Since Evertool can be powered by the target-system the MAX3232 (or compatible 3.3V RS-232-interface-IC) is preferred for reliable programming and debugging of 3.3V target systems. Some "5Volt-MAX232" (or compatible) ICs are known to work at 3.3V. But I do not have any reliable source of information.

Two remarks about low-voltage ("L"-type) AVRs:

- Here is an information from the AVR Technical support.
  http://www.robotgames.net/atmel/mega8_or_mega8l.htm

  > Is there actually a difference between the chips?
  >
  > Someone told me that the Mega8L chips were Mega 8L's that failed the 16MHz speed tests. I asked Atmel about this and this was their engineer's response:
  >
  > *Dear Customer,*
  > *All versions are basically the same device. The only difference is that they are tested with different voltage levels and speed grads. There is no requalification of failing devices from one classification to another. As you can see by the definition of the voltage level and speed grades of the AVR devices, there is a clear relation between these settings. It is therefore very likely that a specific device from one classification will work within the specification limits of another classification. But this can never be guarantied from Atmel, due to the fact that that specific device is in production only tested within its own classification limits.*
  > *Best Regards,*
  > *Atmel AVR Technical Support*
  >
  > So failed chips don't get sold off under a different label and it's very likely that a Mega8L will run at 16MHz and a Mega8 will run at 2.7V. Just note that they haven't been tested outside their specs and remember it's unlikely that either will run at 16MHz and 2.7V at the same time.
  >
  > Of course if you're buying a chip buy the one tested for the conditions you expect to use it under. For me the decision was easy because on a Robot voltage tolerance can be an issue. You don't want your Bot to coats to a halt in the middle of a bout because the voltage dipped below 4V. I also don't care about the higher speed because I plan on using the internal clock which only goes up to 8MHz anyway

- Another non-offical information: The datasheets state that only L-type AVRs should be used for voltages below 4.5V. According to the information I've got from some "insiders" (who do not work for ATMEL) there is no selection for "low-voltage" operation. The selection is done for controllers which do not work reliably at 16MHz. Those are labeled as "L"-type. So there should be no problem using "normal" (non-"L") AVRs with Evertool at 3.3V operation voltage. I have tested "normal" AVRs with my prototypes and had no problems with Vcc=3.3V. Since XTAL-frequencies for both Evertool-controllers are below 8MHz you may just select the type you can get cheaper.

Development-Prototype-Board:

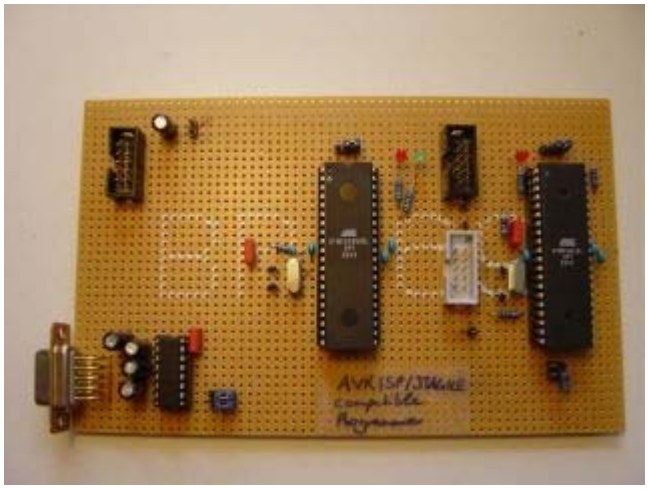


Abbildung 208: Evertool Development Prototype Board

A Picture Gallery you can found under
http://www.siwawi.arubi.uni-kl.de/avr_projects/evertool/gallery/index.html

15.1.5 Software:

- The special Evertool-firmware (a "bootloader"). This bootloader gets programmed into the ATmega16. The distribution-archive includes a Windows batch-file for easier installation (see below). The installer uses "AVRDUDE W32" which is also included. Make sure the driver "give-io" is installed when a par-port-programmer (i.e. a STK200-type-dongle) is used with WinNT/2k/XP. Give-io is also included. Functions of the Evertool-bootloader:
  - UART-Bootloader to upload the ATMEL JTAGICE firmware into the flash of the ATmega16 on the Evertool-board thru AVRStudio/AVRProg.
  - AVR910-style ISP programmer to program the ATMEL AVRISP firmware into the ATmega8535 on the Evertool-board thru AVRProg. The Evertool ATmega16 acts like the AVR90S1200 in the original AVR910-design but the hardware UART and hardware SPI of the ATmega16 are used instead of the "bit-bang" SPI and software UART in the original design with an AVR90S1200. The Evertool's bootloader programming support has only been tested with an ATmega8535 (also in AT90S8535-compatibility-mode) since this is the only Atmel AVRISP/STK500 compatible controller.
  - High speed clock generation for rescue clock.
- Binary-distributions of the Evertool-bootloader. Only free for non-commercial use!
  - Version 0.7_HW09 (Timestamp 20050211). For the Hardware-Versions 0.7->1.1.
    http://www.siwawi.arubi.uni-kl.de/avr_projects/evertool/evertool_SW07_HW09_PB0_20050211.zip
  - Version 0.7_HW06 (Timestamp 20050211). For the Hardware-Version 0.6 JMP.
    http://www.siwawi.arubi.uni-kl.de/avr_projects/evertool/evertool_SW07_HW06_PA0_20050211.zip
  - Old Version 0.2 (Timestamp 20040601). This version has two issues in the ATmega8535 programming-module: (a) Sometimes AVRProg looses sync. Just press the OK-Button if the Sync-Dialog appears on the screen. (b) The ATmega8535 can't be erased from the Evertool-Bootloader which causes errors during upgrade of the AVRISP-firmware when there is already some code in the ATmega8535 flash. Both issues have no influence on the ATmega16 JTAG-firmware update and have been fixed in version 0.7 and 1.1a.
    http://www.siwawi.arubi.uni-kl.de/avr_projects/evertool/evertool_0_2_bin_20040601.zip
- Licensed users will find the commercial version 1.1a on the web-page mentioned in the Evertool manual.

- ISP-Programming software compatible to the programmer you've got from your friend (AVRDUDE, ponyprog, STK500-plungin etc.). To upload the bootloader into the ATmega16 and set the fuses and lock-bits in the ATmega8535 and ATmega16
- AVRStudio 4 - AVRProg (from the tools-menu) is needed to upload the AVRISP-firmware and the JTAGICE-firmware which also come with AVRStudio.

15.1.6 Step-by-Step Guide to Software-Installation:

1. Build the board and double-check it.
2. Set the reset-select jumper near the ATMega16 to "ISP ATmega16".
3. Set the reset-select jumper near the ATMega8535 to "ISP ATmega8535".
4. Connect your programmer (STK200-dongle or whatever) to the ATmega16 ISP-Port.
5. Connect an external power supply to the Evertools board if STK200-PPI, AVRISP or AVR910-type programmer is used.
6. Upload the Evertool bootloader into the ATmega16 flash using PonyProg or avrdude or whatever software can talk to you programmer.
7. Program the ATmega16 fuses and lock-bits (external high freq. crystal, boot-reset, boot-size 1024 words, SPM in boot-section locked).
8. Connect your programmer (STK200-dongle or whatever) to the ATmega8535 ISP-Port.
9. Program the ATmega8535 fuses (external high freq. crystal, AT90S8535-compatibility flag (not needed - but can not do any harm)).
10. Set the reset-select jumper near the ATmega16 to "Normal/ATmega8535 programming".
11. Power-down the Evertool board.
12. Connect your PCs serial port to the Evertool board with a serial/RS232 cable.
13. Select the UART-Jumpers/Switch to connect to the ATmega16.
14. Close the "enable Bootloader (BL)" Jumper near the ATmega16.
15. Connect the ATmega16 ISP Pin-Header ("ISP mega16") to the ATmega8535 ISP Pin-Header ("ISP mega8535") with a 10-wire cable (You will need such a cable to connect Evertool to you target boards - so this is no extra part).
16. Power-up the board.
17. Start AVRStudio, start Tools/AVRProg. (2)
18. Select "ATmega16 boot" from the AVRProg device selection.
19. Select [Browse] and browse to /programs/Atmel/AVR Tools/JTAGICE in the file-dialog, select the file Upgrade.ebn (set filter to *.*).
20. Press the [Program] button in the "Flash" frame. Now the JTAGICE firmware gets uploaded into the ATmega16 via "self-programming" aka bootloader-programming and verified from the Evertool-bootloader.
21. Select "ATmega8515 boot" (yes: no typo. It's 8515 not 8535) from the AVRProg device selection.(3)
22. Select [Browse] and browse /programs/Atmel/AVR Tools/STK500, select the file stk500.ebn (set filter to *.*).
23. Press the [Program]-Flash button. AVRProg will ask to continue since the Flash "overwrites boot", select "continue". The next message box will ask if the file should be truncated: say "no". Now the Evertool-bootloader uploads and verifies the AVRISP firmware into the ATmega8535 via serial-(in-system-)programming.(4)
24. Power-down the board
25. Remove the cable between ATmega16 ISP-connector and ATmega8535 ISP-connector.
26. Open the Bootloader-Enable ("enable BL") Jumper near the ATmega16

27. Set the reset-select jumper near the ATMega8535 to "normal/AVRISP mode"
28. Done.

Remarks:
(1) n.t.
(2) If the message box "No supported board found" appears on the screen verify that no user software uses the COM-port, the UART-select jumpers/switch are/is set to ATmega16/JTAG, the Evertool bootloader is programmed into the ATmega16, the board is powered and the "enable Bootloader"-jumper is set. You may also read the AVRProg user guide.
(3) You may verify that the ISP connection between the ATmega16 and the ATmega8535 is established by pressing the [Advanced] button while the device "ATmega8515 boot" is selected. The Advanced Screen should show the ATmega8535 device signature (see the ATmega8535 datasheet).
(4) If you get verify-errors after the ATmega8535 programming first verify that all connections and jumper settings are correct and the signature-bytes can be read (Advanced Button). If there are still problems press the Chip-Erase button in the advance stettings Chip Erase for the Mega8535 is implemented in bootloader version 0.7 and above.
The steps 1-10 are only needed for the first time to prepare the "virgin" ATmegas. The binary-distribution of the Evertool bootloader includes a setup-routine which will partly automate the steps 1-10 (flash upload and fuse-settings). All other steps are needed every time the AVRStudio whants to update an AVRISP or JTAGICE to a new firmware. Avoid letting AVRStudio do the updates automatically (it will not work). Just cancel the "upgrade-dialogs" and follow the "AVRProg-procedure" (Steps 11 to end) above.
A description how to program the ATmega16- and ATmega8535-fuses with PonyProg provided by Jeff Havens can be found on this page.
The commercial version of the bootloader comes preprogrammed on an ATMega16. No additional programmer ist needed for the initial Evertool setup. There are no "overwrite" and "truncate" dialogs during ATmega8535 programming when using the commercial version and the ATmega8535 fuses are set to the correct values by the bootloader.

15.1.7 Operation

Depending on the UART-Jumpers/switch and the connection to the target the board now acts as an AVRISP or as a JTAGICE.

- Set the UART-jumpers to ATmega8535 and connect the target to the ATmega8535 ISP pin-header (either 10pin "Kanda-standard" or 6pin "Atmel standard") and the board will act as an AVRISP/STK500-protocol-programmer which can be accessed from AVRStudio's Tools-menue (or from AVRDUDE as programmer-type STK500). See the Application-note AVR 042 for pin-assigments. avrdude 4.3.0 does not support the AVRISP/STK500 firmware version 2.x which is shipped in the stk500.ebn-file of AVRStudio 4.11. An experimental avrdude-Version with firmware version 2 support is available from my AVR-projects-page.

  http://www.atmel.com/dyn/resources/prod_documents/doc2521.pdf
  http://www.siwawi.arubi.uni-kl.de/avr_projects/index.html#avrdudew32

- Set the UART-jumpers/switch to ATmega16 and connect the target to the JTAG-header and the board will act as a JTAGICE which can be accessed from the AVRStudio Tools-menu.

- Special "Rescue"-Mode: When the Bootloader-Enable jumper is set during power-up a clock-signal (0,73MHz) is available at the "Rescue-Clock" pin-header. With this clock-signal connected to the XTAL1-pin of an AVR a misconfigured controller can be accessed and reprogrammed. (helps with fuses: external clock-signal instead of external crystal/quarz et.). Use the AVRISP part of Evertool to reprogram the fuses with the AVRStudio STK500/AVRISP plug-in (the JTAGICE part is not available if the ATmega16 is in bootloader/rescue-mode).

The operation is described in the AVRStudio online-help. The AVRISP parts acts exactly as the original. The only extension is the "Target Reset" button.

The JTAGICE-Part works as described in the AVRStudio online-help. You may also use the JTAG-section with the AVaRICE interface-software to the gnu-debugger (gdb) on non-MS-Windows machines.

http://avarice.sourceforge.net/

15.1.8 Questions and Answers

Are ready-made Evertools available?
Yes, see the Myevertool-Page. The page is in German but you can contact Mr. Büttner by e-mail in English. Myevertool is a sligthly modified version of the Evertool-design and has a licensed version of the "commercial" Evertool-Bootloader preinstalled.

Are printed circuit-boards available?
Yes, see the Myevertool-Page. The page is in German but you can contact Mr. Büttner by e-mail in English (info (AT) myevertool [D O T] de).

Can I have the Eagle-files?
No, I do not offer Eagle-Files neither schematic nor layout.

Why two controllers?
Since the original Atmel firmwares are used, there have to be two controllers. One for each task. It may be possible to extend the Evertool-bootloader to emulate an STK500/AVRISP or AVR910-type programmer but I do not have enough time and knowledge to keep this software up-to-date with every new AVR-device.

Why no USB interface?
An USB to serial (RS232) converter has been tested succesfully with the Evertool prototyp-board. This converter is avaialable for only 13EUR inlcuding cable at www.reichelt.de. A FT232/245 with USB-socket and cable is not cheaper in small amounts. For a device similar to the Evertool but with USB-Interace see below.

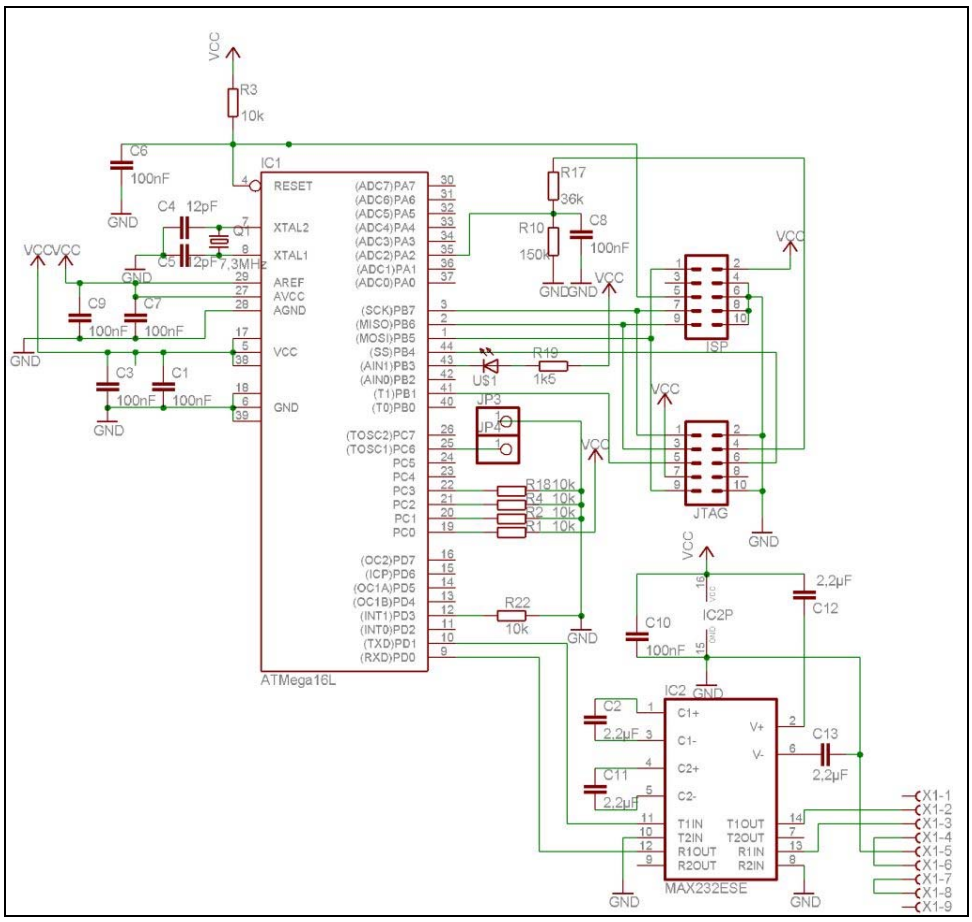
Why SMD?
Well, no reason. It's my first SMD-project. Thought Evertool might be "nicer" in SMD.

Does Evertool support Debugwire?
No. So far no MKII schematic is available for "inspiration" and the MKII Firmware coming with AVRStudio seems to have another format than the ebn-files.

AVRProg shows "No supported board found" - what's wrong?
Please double check your board and double check that all configuration-steps have been done as described above before sending me an e-mail. You can test if the Bootloader is installed, the fuses are set correctly and the bootloader is active by using a terminal software (Hyperterm, Bray Terminal...). Set the terminal to: 19200, 8N1, no flowcontrol, mount the "enable Bootloader (BL)"-Jumper, apply power to the board, make sure the RS232-level-shifter is connected to the ATmega16 and send a S (capital S) to the Evertool. If everything is o.k. the bootloader will answer with the characters: AVRBOOT. If not: double check...

15.1.9 "Evertool-Light"

"Evertool-Light" is a stripped down version of Evertool which emulates the JTAGICE only. The design has been done by Rainer Rakow. The schematic is based on my Evertool design. Basicly this is an extended version of the "BOOTICE"-design which can be found somewhere else.



Schaltbild 88: Evertool Light

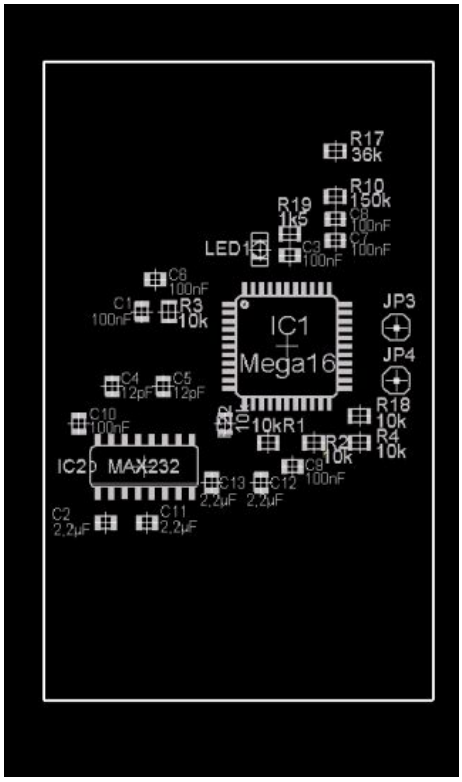


Abbildung 209: SMD Side

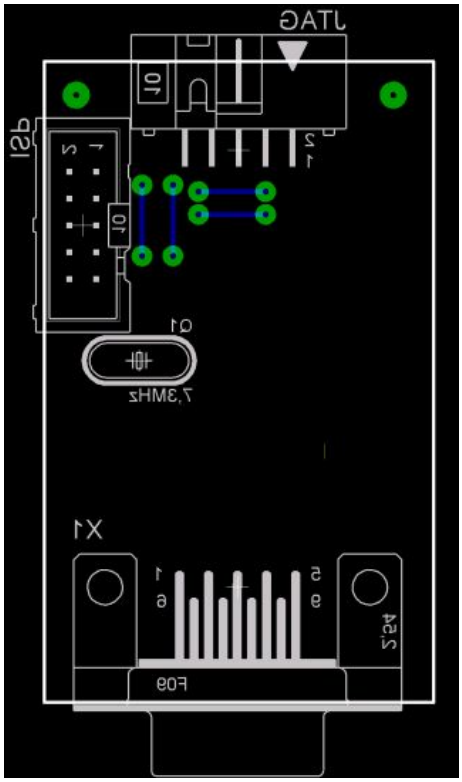


Abbildung 210: Connector Side

zip-Archive including schematic and PCB-layout (Postscript-files) of Evertool-Light (ca. 180kB)
http://www.siwawi.arubi.uni-kl.de/avr_projects/evertool/light/evertool_light_rr.zip

(9.Jul.2004: Message from Rainer: the free end of resistor R2 should be connected to GND in his layout.)

15.1.10 Software-Installation on Evertool LIGHT

1. Build the board and double-check it.
2. Connect your programmer (STK200-dongle or whatever) to the ISP-Port.
3. Connect an external power supply to the Evertool-Light board if STK200-PPI, AVRISP or AVR910-type programmer is used.
4. Upload the Evertool bootloader (which can also be used for Evertool-Light) into the ATmega16 flash using PonyProg or avrdude or whatever software can talk to you programmer.
5. Program the ATmega16 fuses and lock-bits (external high freq. crystal, boot-reset, boot-size 1024 words, SPM in boot-section locked - lfuse: 0xff hfuse: 0xd8:m lock: 0x2f).
6. Power-down the Evertool-Light board.
7. Connect your PCs serial port to the board with a serial/RS232 cable.
8. Close the "enable Bootloader (BL)" (JP3) Jumper
9. Power-up the board.
10. Start AVRStudio, start Tools/AVRProg. **(1)**
11. Select "ATmega16 boot" from the AVRProg device selection.
12. Select [Browse] and browse to /programs/Atmel/AVR Tools/JTAGICE in the file-dialog, select the file Upgrade.ebn (set filter to *.*).
13. Press the [Program] button in the "Flash" frame. Now the JTAGICE firmware gets uploaded into the ATmega16 via "self-programming" aka bootloader-programming and verified from the Evertool-bootloader.
14. Power-down the board
15. Open the Bootloader-Enable ("enable BL"/JP3) Jumper
16. Done.

Remarks:
(1) If the message box "No supported board found" appears on the screen verify that no user software uses the COM-port, the UART-select jumpers/switch are/is set to ATmega16/JTAG, the Evertool bootloader is programmed into the ATmega16, the board is powered and the "enable Bootloader"-jumper is set. You may also read the AVRProg user guide.
http://www.atmel.com/dyn/resources/prod_documents/DOC1021.PDF

The steps 1-5 are only needed for the first time to prepare the new ATmega16. The binary-distribution of the Evertool bootloader includes a setup-routine which will partly automate the steps 1-5 (flash upload and fuse-settings). All other steps are needed every time the AVRStudio whants to update the JTAGICE to a new firmware. Avoid letting AVRStudio do the updates automatically (it will not work). Just cancel the "upgrade-dialogs" and follow the "AVRProg-procedure" (Steps 6 to end) above.

15.1.11 "Evertool-USB"

Florian Schenk has extended the Evertool with an USB-Interface (FTDI232-USB-to-serial IC). He has also created a layout. The schematic and the Eagle-files (.brd and .sch) can be downloaded from [here] (timestamp 20060908). If you have questions you may ask Florian Schenk directly by e-mail (schenk (At) schema2000 (Dot) de).
http://www.siwawi.arubi.uni-kl.de/avr_projects/evertool/evertool_usb_v1.zip

I (M.Thomas) do not offer any support for this ET-USB-design. I have not reviewed the layout and I do not own such a device. Anyway, at least a remark from my side: Take care about the voltage-supply. When the jumper near the USB-connector of "ET-USB" is closed VCC for the complete device is 5V and the levels at the ISP- and

JTAG-connectors are also near 5V. This is a "no-no" when connecting to a 3,3V target. If in doubt leave the supply-jumper near the USB-connector open and supply the ET-USB from the target-board.

15.2 ispRE

ispRE: AVRISP-kompatibles Programmiergerät für Atmel AVR Mikrocontroller

Eine weiterentwickelte Version des hier vorgestellten Geräts mit Programmier- und Debugging-Unterstützung, die zumindest ich interessanter finde, wird auf meiner Evertool-Seite vorgestellt.

15.2.1 Intro

Das ATMEL AVRISP Programmiergerät zur Im-System-Programmierung (ISP) ist ein Gerät zur Programmierung (Flash/EEPROM/Fusebit/Lockbits) von Atmel AVR Mikrocontrollern. Informationen zum Originalgerät von ATMEL finden sich auf einer Atmel-Seite.

http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2726

Hier wird ein Nachbau dieses Gerätes mit (fast) minimaler Hardware beschrieben. Der Hauptvorteil dieses Nachbaus ist die Kompatibilität zum Atmel AVRStudio und der darin enthaltenen sehr anwendungsfreundlichen Software. Da sowohl die genutzte Software auf dem Programmiergerät und die Bedienungssoftware vom Hersteller gepflegt und aktualisiert werden, sind beide immer auf sehr aktuellem Stand. Das Programmieren von Fuses verliert seinen "Schrecken" durch klare Beschreibungen in der Software (AVRStudio/Tools/STK500, AVRISP).

Bevor Sie den Nachbau dieses Geräts in Betracht ziehen, bedenken Sie bitte:

- Das original von Atmel ist sehr günstig (Preise 5/2004, ca. 40 Euro inkl. MwSt. bei Reichelt.de, ca. 26 EUR+MwSt. bei de.digi-key.com). Der Nachbau des hier vorgestellten Geräts lohnt sich also möglicherweise nicht. Die Matrialkosten des vorgestellten Nachbaus belaufen sich auf ca. 15EUR. Ich habe das Geräte anfangs nur als Testplattform für Evertool entwickelt und später die Update-Funktion mit Code aus einem anderen Projekt nachgerüstet.
- Die vorstellte Hardware enthält nur die nötigsten Bauteile (z.B. kein ESD-Schutz, Nachrüstung mit ICs von Littlefuse oder TI u.a. ist möglich). Der Vergleich des hier wiedergegebenen Schaltplans mit dem des Originals zeigt die Unterschiede. Einige Widerstände können entfallen, da diese aber so günstig sind, wurden sie im Schaltplan belassen (markiert als "optional")

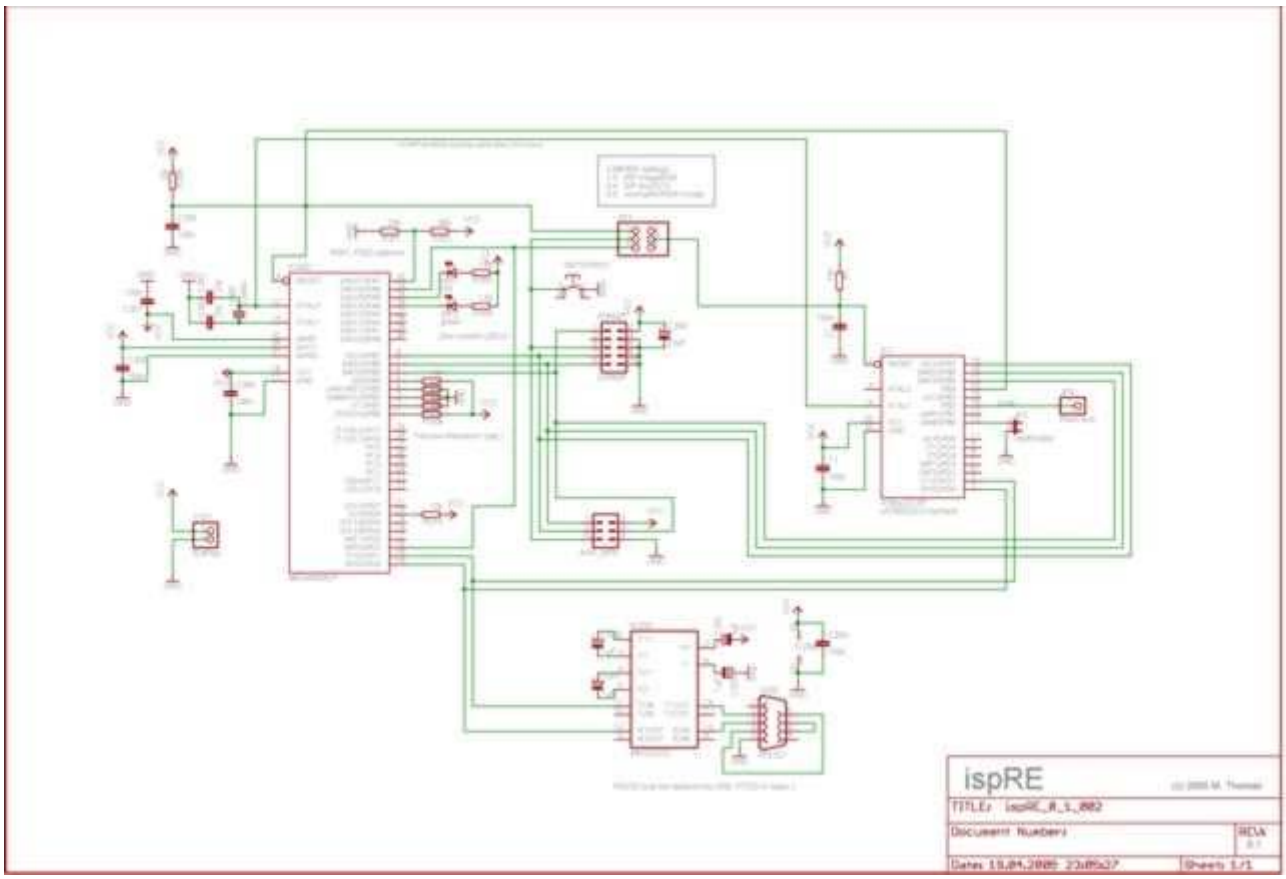
  http://www.atmel.com/dyn/resources/prod_documents/Avrisp_Sch.pdf
- Zum Bau diese Geräts wird kurzzeitig ein weiteres Programmiergerät benötigt (ja, "Henne/Ei"). Ein einfacher Parallelportadapter reicht jedoch vollkommen aus. Siehe dazu z.B. die lancos.com PonyProg Seite. Man kann natürlich auch einfach ein AVRISP/AVR910/STK500 ausleihen. Der zweite Programmieradapter wird nur für die Ersteinrichtung benötigt. Updates der Progammiersoftware (bei neuem AVRStudio) erfolgen durch den internen ispRE Programmieradapter (auf dem ATtiny2313) - das Gerät kann sich also "selbst updaten".

  http://www.lancos.com/prog.html
- Auf den Seiten von Tobi wird ein ähnliches Gerät beschrieben, das neben "serial download" ("normalem" ISP) auch high-voltage ISP und "parallel programming" ermöglicht. Etwas mehr Schaltungsaufwand, kein internes Update, dafür aber weitestgehend kompatibel zur Programmierhardware auf dem Atmel STK500.

  http://www.der-hammer.info/hvprog/index.htm

15.2.2 Hardware

Schaltplan:



Schaltbild 89: ispRE

Hauptkomponenten:

- ATmega8535 (vorzugsweise L-Typ). möglicherweise auch AT90S8535, aber nicht getestet
- ATtiny2313 für internes Update
- MAX(3)232 (vorzugsweise 3V-Version)

Testplatine (Lochraster):

Abbildung 211: istRE Testplatine Lochraster

Bauteilliste:

Part	*Pieces*	*Function*
ATMega8535(L)	1	Programming Controller, L-type for reliable programming of 3V targets
DIL40 Socket	1	for ATmega8535
MAX3232	1	RS232 level shifter and inverter, 3V-type for reliable programming of 3V targets
DIL14 Socket	1	for MAX(3)232
3,6864MHz Quarz	1	
ATtiny2313	1	internal update controller
DIL20 Socket	1	for ATtiny2313
1uF Electrolyt-Capacitor	4	4 for MAX(3)232 charge pumps
22uF Electrolyt-Capacitor	1	stabilisation near ISP-connector
100nF Capacitor	5	"block capacitors" 4 for AVRs, 1 for MAX(3)232
22pF Capacitor	2	near Quarz-Oscillator
100nF Cap.	2	Filter at AVR Reset
10kOhm Resistor	2	AVR Reset Pull-Ups

LED low-current (2mA)	2	Indicator LEDs
1.5kOhm Resistor	2	LED current limiter
DSUB connector "female"	1	PC RS232 connection
Pin-Header 2*5	1	ISP connection "Kanda-Standard" (see schematic)
Pin-Header 2*3	1	ISP connection "Atmel-Standard" (see schematic)
Pin-Header 2*3	1	ISP Resetline selector (see schematic)
Pin-Header 1*3	1	Update-enable Jumper
Pin-Header 1*1	1	Rescue-Clock connection (ca. 0,7 MHz)
Jumper	2	for reset-selector and update-enable
Pin-Header 1*2	1	external power supply for internal programming with STK200-PPI (stable DC)
Prototype-Board 2,54 mm grid	1	

Tabelle 156: Stückliste ispRE

15.2.3 Software

- Zur Erstinbetriebnahme wird eine zum vorhandenen zweiten Programmadapter kompatible Software benötigt. Z.B. avrdude, ponyprog, avrprog - was auch immer zum Programmer "passt".
- AVRStudio 4.x
- Die ATtiny2313 Update-Firmware. Diese ist z.Zt. nur in Binärform (Intel-hex) verfügbar. Download der ATtiny2313 firmware ("Version" 0.2 20060205).
- (Eine Version der Update-Firmware für den alten AT90S2313 (20060205). Die AT90S2313-Firmware wurde nicht von mir getestet, aber von Dritten erfolgreich angewendet.)

Die *2313-Firmware darf nur zu **nicht-kommerziellen** Zwecken genutzt werden! Falls Sie eine kommerzielle Version benötigen: Nehmen Sie bitte via e-mail mit mir Kontakt auf (Adresse s.u.)

Die Update-Firmware für den ATtiny2313 ist auch für andere Zielprozessoren ausser dem ATmega8535 erhältlich. Falls sie also eine Update-Funktion auf Ihren Eigenentwicklungen benötigen, aber keinen Bootloader in den Hauptprozessor integrieren wollen: nehmen Sie bitte Kontakt auf, angepasste Versionen sind bereits fertig oder kurzfristig lieferbar. Die eigentliche ATmega8535-Firmware ist nur von Atmel lieferbar, das Angebot gilt nur für die Update-Firmware des ATtiny2313. (Eine STK500.hex-Datei ist hier **nicht** erhältlich).

15.2.4 Schitt für Schritt Anleitung

Erstinbetriebnahme:

1. Diese Schritte müssen nur einmal abgearbeitet werden.
2. Platine herstellen und mehrmals prüfen
3. Steckbrücke (Jumper) Update-Enable offen
4. Reset-Select Jumper auf "ISP mega8535" (siehe Schaltplan)
5. Anschluss einer externen Spannungsquelle, falls der genutzte zweite Programmieradapter nicht über eine eigene verfügt (wie bei üblichen Parallelport-Dongles der Fall)
6. Setzen der ATmega8535-fuses: External high-freq-Crystal, CKOPT enabled ("programmed"), AT908535 compatibility. Für avrdude-Nutzer: lfuse = 0xff, hfuse = 0x49, locks = 0xff
7. Stromversorgung ausschalten
8. Reset-Select Jumper auf "ISP tiny2313"
9. Stormversorgung herstellen
10. Setzen der ATtiny2313-fuses. External Clock (everything else disabled: devided, brownout etc.). Für avrdude-Nutzer: lfuse = 0xe0, hfuse = 0xdf, efuse = 0xff, locks = 0xff
11. Programmieren des ATtiny2313's flash-Speichers mit dem Inhalt der Datei ispre*.hex
12. Ausschalten
13. Reset-Select-Jumper auf "normal/AVRISP-mode"

Update der Programmierhardware:

Die folgenden Schritte müssen zur Einrichtung und Aktualisierung der Programmierfirmware auf dem ATmega8535 durchgeführt werden. Also bei der Erstinbetriebnahme oder dann wenn die Firmware, die mit einer neuen Version von AVRStudio geliefert wird genutzt werden soll.

1. Auschalten
2. kein Gerät an den ISP-Anschluessen.
3. Herstellen einer seriellen Verbindung (oder einer USB-Verbindung, falls ein FT232 oder ähnlicher IC genutzt wird (Treiber mit "virtuellem COM-Port"))
4. Update-Enable Jumper schliessen
5. Stromversorgung herstellen
6. Start von AVRstudio->Tools->avrprog (kein anderes Programm darf auf den COM-Port zugreifen)
7. Flashen der Datei STK500.ebn aus dem Verzeichnis /programme/atmel/AVRTools/stk500 (Filter im Dateidialog auf *.*, evtl. "Datei suchen"). Der ATtiny2313 empfängt den Atmel-Firmwarecode und "brennt" damit den ATmega8535
8. Ausschalten
9. Update-Enable Jumper öffnen

Benutzung:

Das ispRE ist kompatibel zum Atmel AVRSIP (entspr. STK500 in low-voltage serial-programm/serial download). Die Zielplattform (Platine mit zu programmierendem AVR) wird mit einem der ISP-Anschlüssen auf dem AVR-ISP verbunden. Typisch erfolgt die Stromversorgung durch die Zielhardware über den ISP-Stecker. In AVRstudio Menü Tools->AVRISP/STK500 wählen: Es erscheint das avrisp/stk500-Plugin. Für weitere Informationen siehe AVR-Studio online-Hilfe. Wichtig (es steht auch da): die ISP-Frequenz darf 1/4 der Frequenz des zu programmierenen AVR nicht überschreiten (z.B. Ziel "läuft mit" 1MHz, ISP-Frequenz max. 250kHz).

ispRE kann auch von der Open-Source Software avrdude angesteuert werden. Man beachte, dass mit AVR-Studio 4.11 ein neues STK500-Protokoll eingeführt wurde, wenn die stk500.ebn-Datei aus neuen AVRStudio-Versionen (>=4.11) genutzt wird, ist für avrISP (wie auch das Orignal Atmel AVRISP und das STK500 mit Firmware 2.x) bei avrdude stk500v2 als Programmer anzugeben.

Ein 0,7MHz Takt für am RescueClock-Pin bereitgestellt (am ATtiny). Mit diesem Takt können AVRs "wiederbelebt" werden, bei denen die Fuse-Bits für die Taktgenerierung falsch eingestellt wurden. Einfach den Taktpin mit dem XTAL1-Anschluss des "zerschossenen" AVR verbinden, ISP-Verbindung herstellen und im stk500/avrisp-Plugin die richtige Fuse-Einstellung wählen. (Die Reset-Disable- oder Debugwire-Enable Fuse kann auf diese Weise jedoch nicht zurückgesetzt werden, nur die üblichen "Clock statt Quarz" Fehler). Bei Verwendung der 0,7MHz Takts ist zur Programmierung des "zu rettenden" AVR eine maximale ISP-Frequenz kleiner 0,7MHz/4 einzustellen.

15.3 Das STK500 und STK501

15.3.1 Das STK500 mit STK501 (ATMEL)

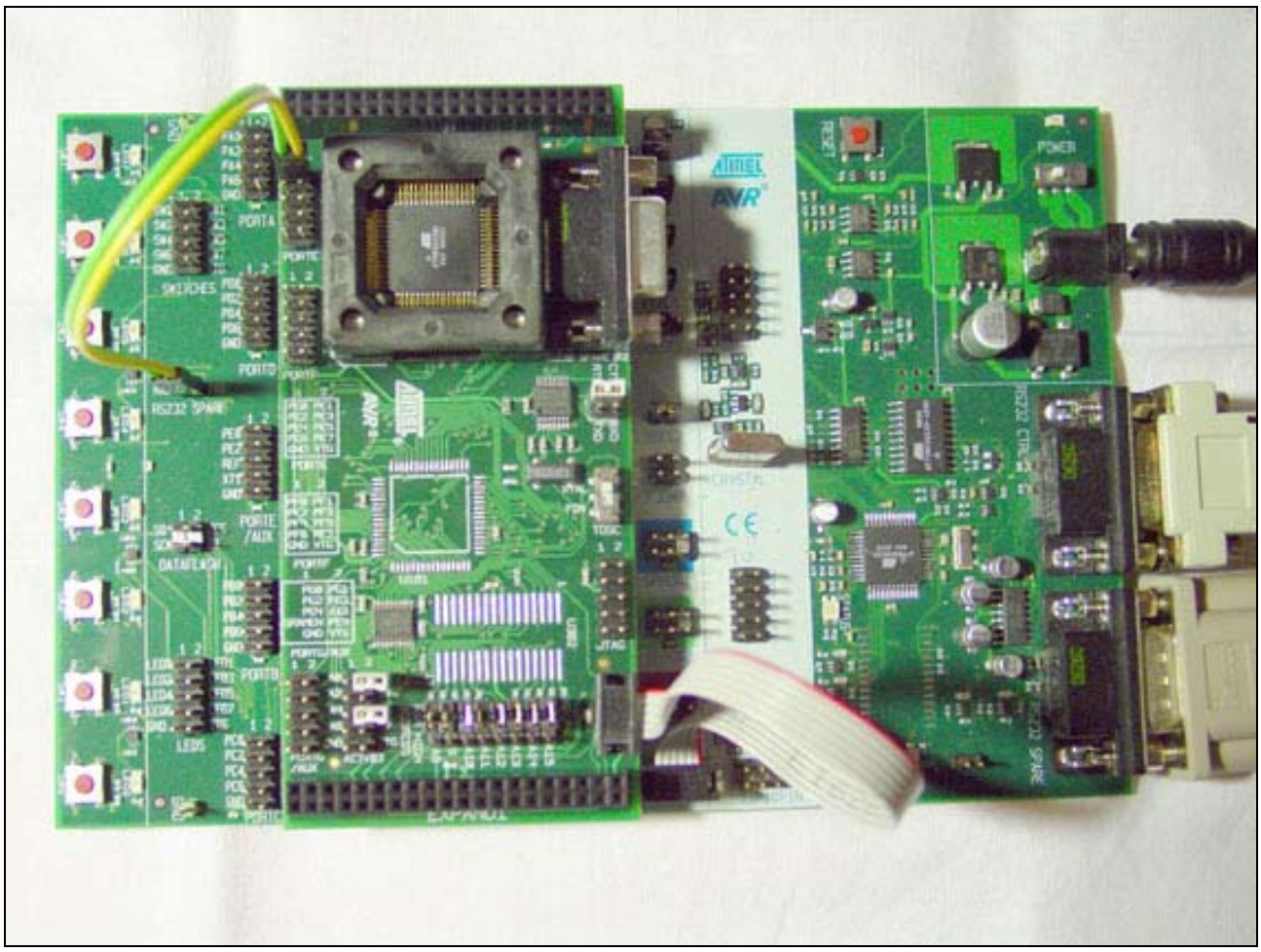


Abbildung 212: Das STK500 mit STK501 (ATMEL)

15.3.2 STK501 anschließen für RS232-Kommunikation

Der 9-polige Sub-D Stecker mit der Beschriftung „RS232 Spare“ wird mit einem freien COM-Port des PC verbunden.

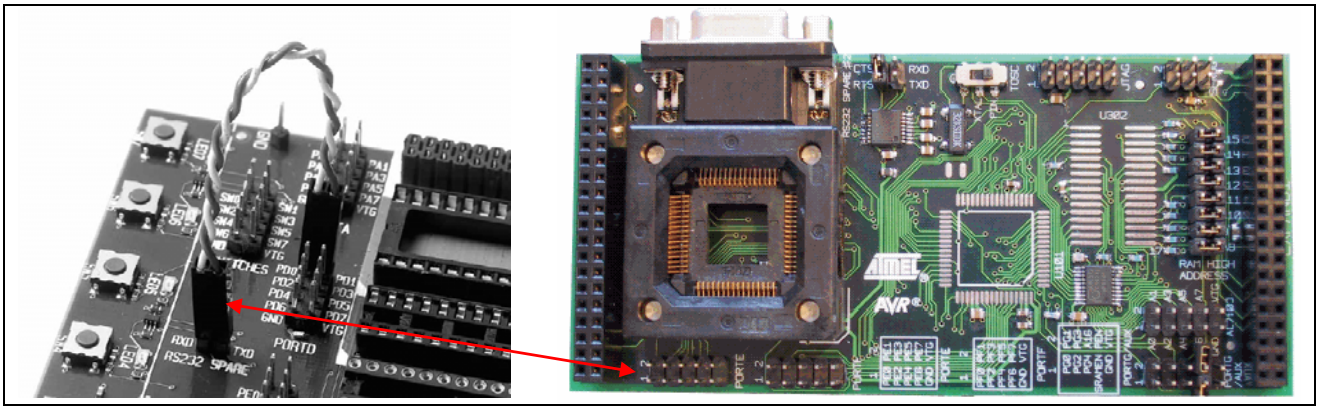


Abbildung 213: RS232-Verbindung STK500 und STK501

Der 9pol. RS232 Spare Stecker ist über einen Pegelwandler mit den beiden Pins im IO-Bereich verbunden (heist auch RS232 Spare). Diese zwei Pins (RxD und TxD) müssen mit dem mitgelieferten 2pol Kabel mit den Pins PE0 und PE1 (PortE) auf dem STK501 verbunden werden.

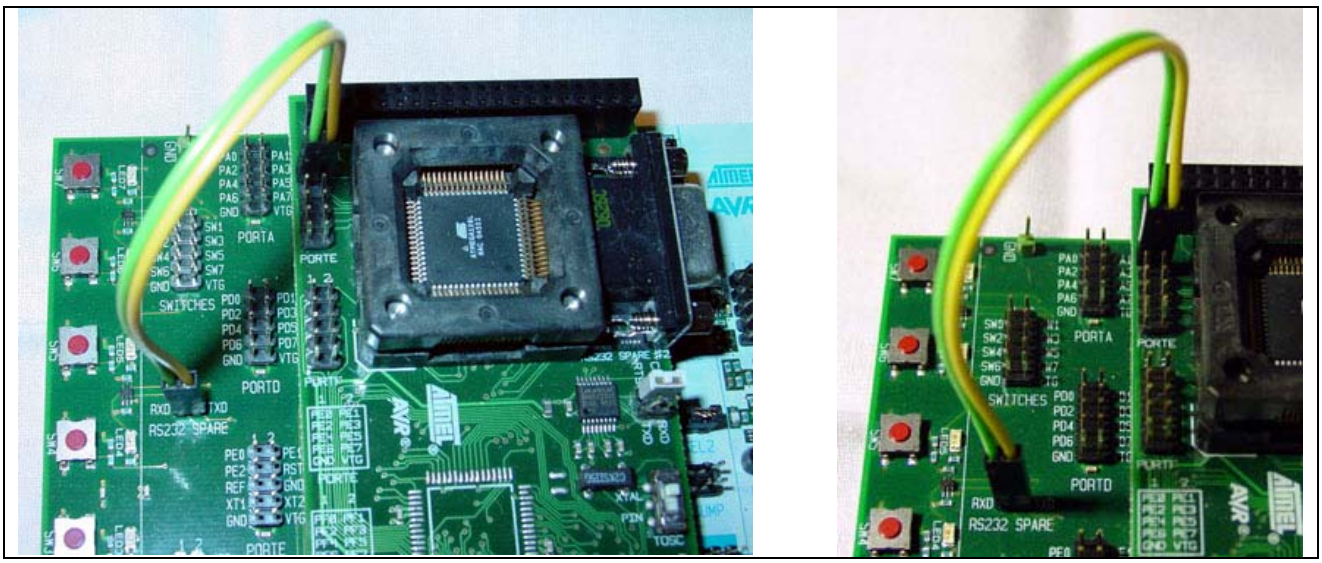


Abbildung 214: RS232-Verbindung STK500 mit STK501

15.4 Inbetriebnahme AVR ISP mkII

Für die Inbetriebnahme des STK500 + STK501 muss der RESET-Jumper auf dem STK500-Board entfernt werden (Dieser Hinweis stammt aus der Dokumentation des mkII).

Alle weiteren Einstellungen zeigen die folgenden Dialog-Boxen des AVR Studios 4.0:

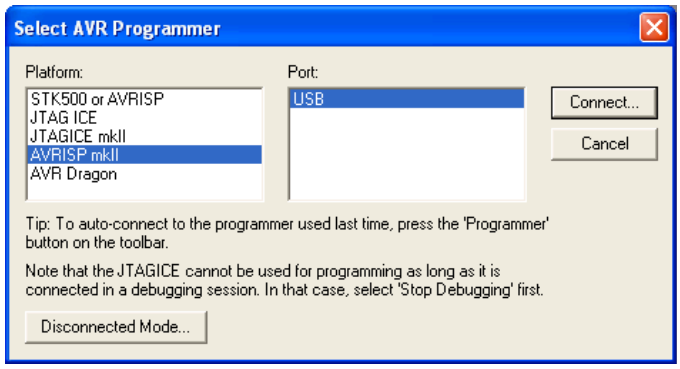


Abbildung 215: AVR Studio Programmer Selection

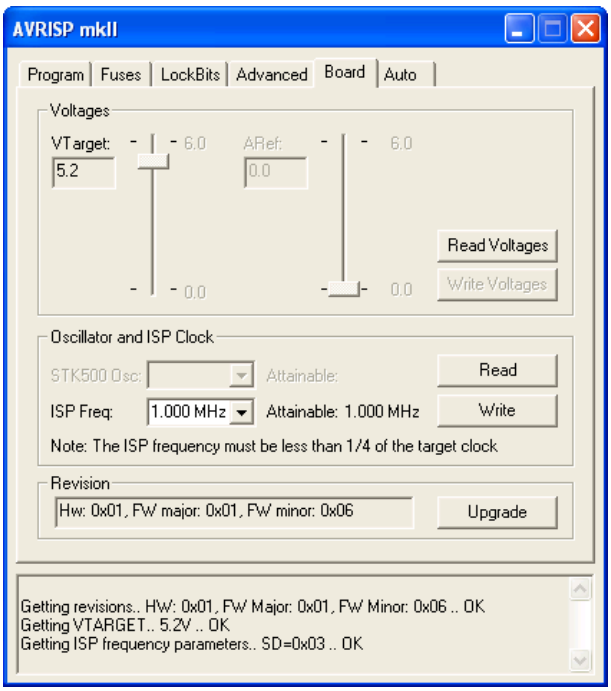


Abbildung 216: AVR Studio Board-Settings

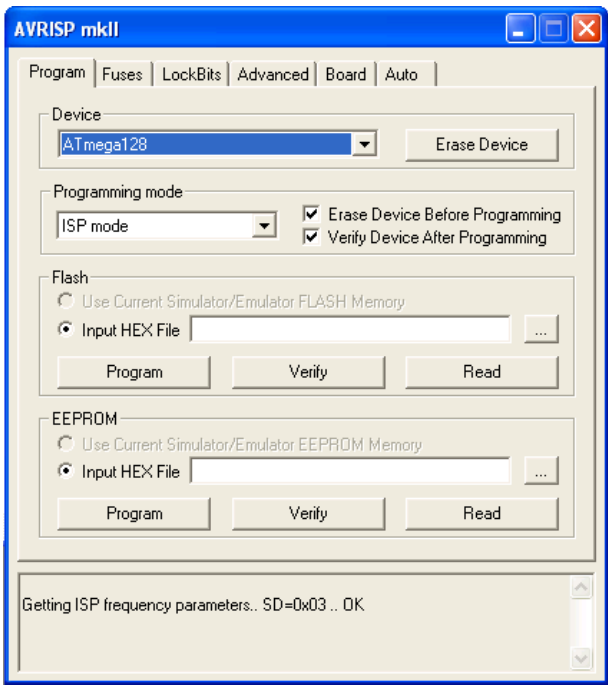


Abbildung 217: AVR Studio Program Settings

Hinweis: Die ISO Frequenz muss 1/4 bzw. 1/6 der Taktfrequenz des ATmegas betragen. Bei 16 MhZ des ATmega128 wurde der ISP Prommer erfolgreich mit 1,000 MHz in Betrieb genommen.

15.5 Pony-Prog Tutorial

Diese Anleitung beschreibt, wie vom Assembler (z.B. AVR-Studio) oder BASCOM-Basic-Compiler erzeugte HEX-Dateien auf einen Atmel AVR-Mikrocontroller übertragen werden. Die vorgestellte Lösung basiert auf freier oder kostenloser Software und low-cost-Hardware. Anfänger sollten sich bevor sie loslegen zuerst einmal das AVR-Tutorial zu Gemüte führen.

Die Ponyprog - Homepage ist http://www.lancos.com/prog.html
Das Ponyprog - Diskussionsforum ist http://ponyprog.sourceforge.net/phorum/list.php?2

15.5.1 Benötigte Ausstattung

- Programmier-Software PonyProg 2000 von www.lancos.com. Für diese Anleitung wurde PonyProg 2000 Version 2.06 beta genutzt.
- STK200-kompatible Programmier-Hardware (Programmieradapter).
  - Fertig zu kaufen bei:
    - http://www.microcontroller-starterkits.de
    - http://shop.mikrocontroller.net
    - http://www.embedit.de
  - Platine zum Selbstbestücken bei:
    - http://www.nc-step.de/ unter "Sonstige Schaltungen"
  - oder selbst bauen nach:
    - http://rumil.de/hardware/avrisp.html
    - www.lancos.com/prog.html Abschnitt "AVR ISP (STK200/300) parallel port interface"

- AVR-Controller-Board mit ISP-Anschluss (siehe weiter unten)

Um schon in der Anfangsphase Probleme zu vermeiden, sollte der Controller beim Einsatz eines STK200-Programmierers mit 5 Volt betrieben werden. Geringere Spannungen werden nicht von allen Programmieradaptern unterstützt. Höhere Spannungen zerstören möglicherweise den Controller oder den Parallelport des PCs.

15.5.2 PonyProg installieren und einrichten

Pronyprog kommt mit einem Installationsprogramm, einfach starten. Einstellung der Programmierhardware (STK200 ISP DONGLE) in PonyProg unter Setup/Interface-Setup nach folgender Abbildung:

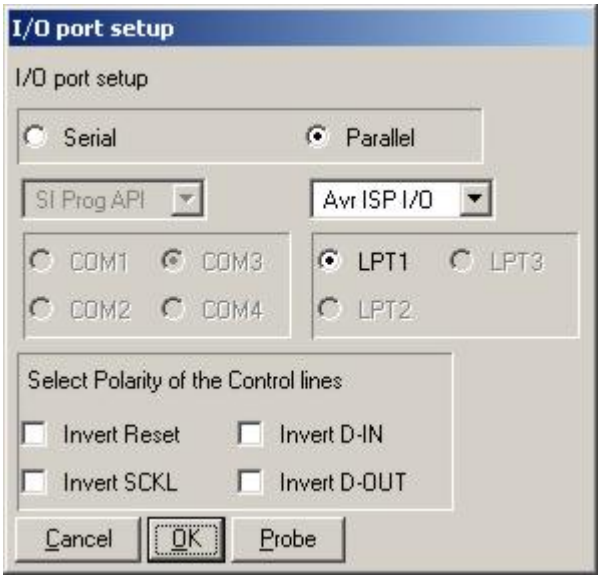


Abbildung 218: Pony Prog I/O port setup

Im Anschluss PonyProg die Zeitschleifen kalibrieren lassen (Menüpunkt Setup/Calibration).
PonyProg unterstützt die automatische Erkennung von AVR-Controllern, um Probleme zu vermeiden, sollte man den Controllertyp jedoch manuell einstellen. Für einen ATmega16 z.B. so:

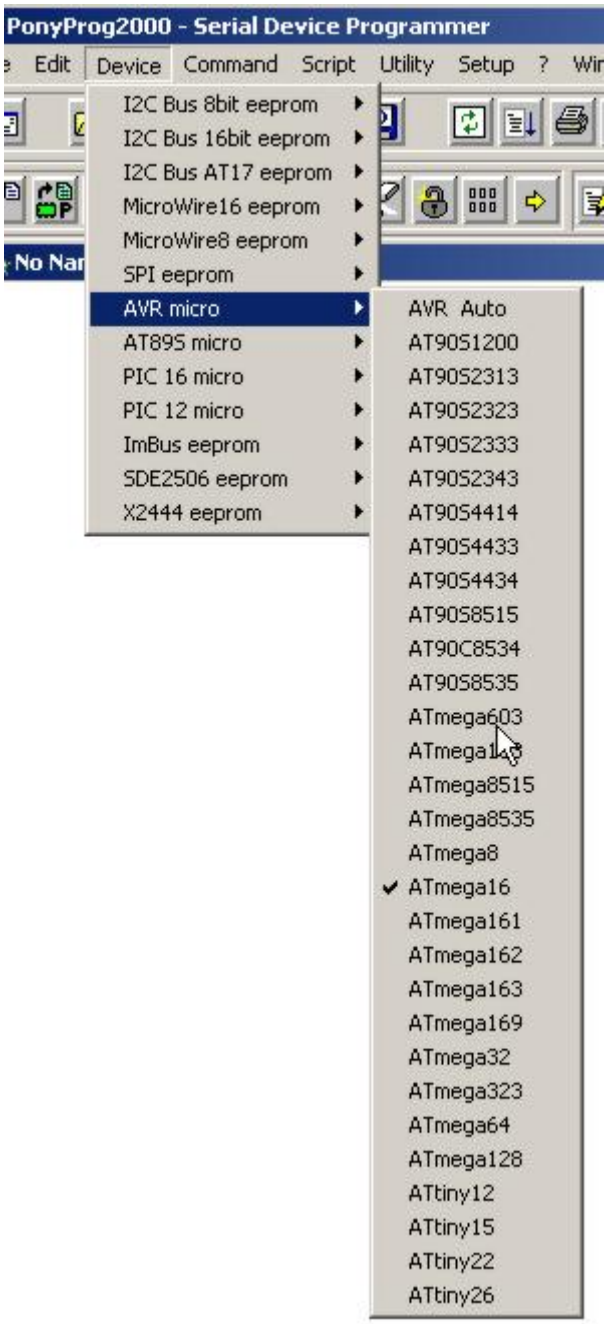


Abbildung 219: Pony Prog Device Setup

15.5.3 Hardwareverbindung herstellen

Auf der Controller-Platine (Controllerboard) ist ein Anschluss für den Programmieradapter erforderlich. Überlicherweise wird dazu eine 10-polige "Pfostenwanne" (andere Bezeichnung: Wannenstecker, Wannen-Stiftleisten) (5 Pole in 2 Reihen, "WSL 10G" bei Reichelt) genutzt (siehe Abbildung). Dieser Anschluss ist auch kompatibel zum ATMEL AVRISP Programmiergerät.



Abbildung 220: Wannenstecker für 10-poligen ISP Anschluss

Die Anschlussbelegung (entnommen aus der AVRISP Online-Benutzeranleitung) siehe folgende Abbildung. Pin 1 ist an Wannensteckern üblicherweise mit einem Pfeil markiert.

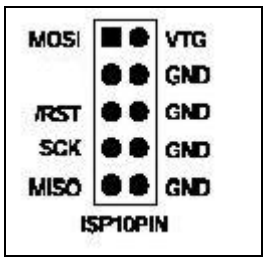


Abbildung 221: AVR ISP Anschlussbelegung

Signal	*Pins*	*E/A (aus "Sicht" des Programmieradapters)*	*Beschreibung*
VTG	2	(Eingang)	Stromversorgung des Programmiergeräts, mit +5V Versorgung des Controllerboards verbinden
GND	4,6,8,10	(Eingang)	Masse, Ground, mit Ground des Controllerboards verbinden
MOSI	1	Ausgang	Datenleitung vom Programmieradapter zum Controller, mit MOSI des AVR verbinden
MISO	9	Eingang	Datenleitung vom Controller zum Programmieradapter, mit MISO des AVR verbinden
SS	3	nicht benutzt	Reserviert für Auswahl eines Slave, wenn über diesen Stecker Master-Slave mit SPI Daten austauschen sollen. - Wird beim Programmieren nicht benutzt und ist nicht verbunden.
SCK	7	Ausgang	Programmier-Takt-Leitung (PA->AVR), mit SCK des AVR verbinden
RESET	5	Ausgang	Resetleitung, wird durch das Programmiergerät bei der Programmierung auf "LOW gezogen", mit RESET des AVR verbinden

Sogenannte "value-added" Programmieradapter unterstützen an PIN 3 den Anschluss einer LED, mit der ein Zugriff auf den Controller durch den Programmieradapter angezeigt werden kann. Weiters dazu in den Beschreibungen der Programmierhardware auf den o.g. Seiten.

Zur Programmierung mittels ISP muss der Controller mit einer "Taktquelle" ausgestattet sein, modernere AVRs sind mit einem internen Oszillator (1 MHz Takt) ausgestattet, der ab Werk aktiviert ist. Bei diesen sind keine zusätzliche externen Bauteile erforderlich. An AVRs ohne internen Oszillator (z.B. die veraltete "AT90S"-Serie) muss zur ISP-Programmierung ein externer Taktgeber (z.B. Quarz mit Kondensatoren) angeschlossen werden.

Der ISP Anschluss wird zur Programmierung mit dem Programmieradapter verbunden, der PC-seitige Anschluss erfolgt an die Druckerschnittstelle (Parallelport, LPT). Es empfiehlt sich, die Stomversorgung des Controllerboards beim an- und abstecken abzuschalten.

15.5.4 Programmierung des Controllers

Schritt 1: Hex-Datei in Ponyprog laden
In PonyProg Menü: File/Open-Device-File... wählen, im Dialog die HEX-Datei auswählen und mit [OK] in PonyProg laden..

Schritt 2: Daten zum Controller übertragen
In PonyProg Menü: Command/Write All wählen. PonyProg überträgt die Daten aus der HEX-Datei dann auf den Controller ("flashen") und verfiziert im Anschluss, ob dieser Schreibvorgang fehlerfrei durchgeführt wurde, indem der Inhalt des Controller-Flash-Speichers gelesen und mit dem Inhalt der HEX-Datei verglichen wird. Der Inhalt des EEPROM wird bei der Programmierung gelöscht, d.h. alle Speicherstellen des EEPROM-Speichers im AVR werden auf $FF gesetzt.

15.5.5 Fuses

Mit Ponyprog kann man auch die Fuses bei den AVRs setzen, muss dabei aber *sehr* aufpassen: Die Fuses sind im AVR invertiert, d.h. 0 = ein und 1 = aus. Das steht auch so im Datenblatt. Ponyprog invertiert das wieder (steht auch da, allerdings in dunkelgrau auf hellgrauem Grund), d.h. 0 = aus und 1 = ein. Jetzt scheint wieder alles normal zu sein.

Allerdings steht im Datenblatt nicht immer ein und aus, sondern manchmal (z.B. bei den Clock sources) steht da auch einfach 0 und 1. Da Ponyprog dieses Fuses ebenfalls invertiert, muss man diese selbst nochmals invertieren.

> Wo im Datenblatt eine 0 steht muss man in Ponyprog ein Häkchen setzen, bei einer 1 im Datenblatt das Häkchen löschen.

Verwirrt? Tja, hier hilft nur sorgfältiges arbeiten. Ganz wichtig ([1]): Erstmal die bisherigen Einstellungen auslesen. Und dann nur das ändern, was unbedingt nötig ist. Finger weg von RSTDSBL. Wenn man diese ändert, dann kann man den AVR nicht mehr über das STK200 programmieren. Ohne alternativen Bootloader oder einem AVR HV-Programmer (z.B. STK500) ist der Chip dann wertlos.

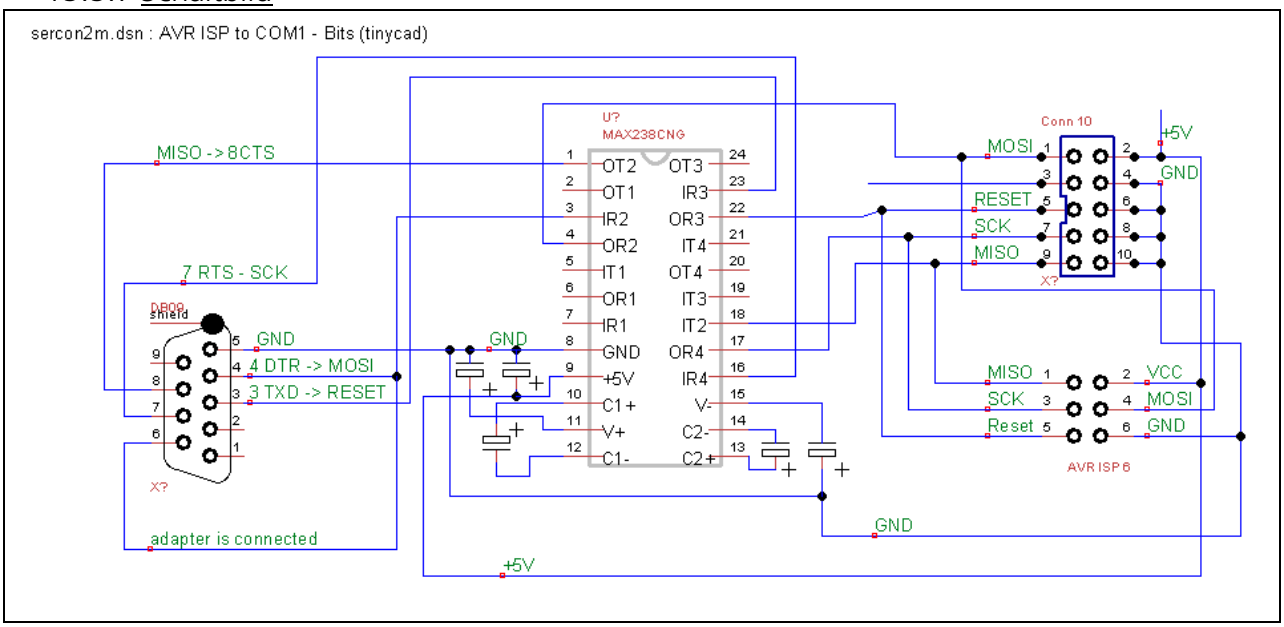
Siehe auch: Vergleich der Fuses bei verschiedenen Programmen

15.5.6 Programmierung über den seriellen Port

Modische Laptops haben nicht immer ein Parallelport. Dann können die 3 Ausgangsleitungen des seriellen Ports COM1 von Ponyprog angesteuert werden. Es werden dabei keine Daten seriell herausgeschickt, sondern die 3 Leitungen werden durch entsprechende Befehle High und Low geschaltet.

Eine Warnung: Es gibt im Handel diverse USB-Seriell Wandler. Ihre Aufgabe ist es am USB-Port eine serielle Schnittstelle zur Verfügung zu stellen. Da PonyProg den seriellen Port aber nicht wie einen standardmässigen seriellen Port benutzt, funktioniert die Programmierung über so einen USB Wandler in den seltesten Fällen.

15.5.7 Schaltbild



Schaltbild 90: AVR ISP to COM1

Die Spannungsversorgung für den MAX238 erfolgt von der Schaltung. Deswegen ist der Programmieradapter nur für Schaltungen brauchbar, die mit 5 V +-5% versorgt werden.

Ähnliche Adapter, die mit Dioden, Z-Dioden und Widerständen aufgebaut sind, sind unter dem Namen "sercon programming adapter" im Internet zu finden. Diese benötigen dann andere Einstellung der Invertierungsbits! Siehe hierzu [[2]]

15.5.8 Vorgehensweise

- Im Ponyprog-Menü : [Setup] [Interface Setup...]

```
(o) Serial [ SI Prog I/O ]
(o) COM1
[ ] Invert Reset [v] Invert D-in
[v] Invert SCKL  [v] Invert D-Out
```

- Zeitkalibrierung durchführen: Im Ponyprog-Menü : [Setup] [Calibration]
- Anwahl des Mikroporzessortyps AVR und ATmega8535
- Laden der HEX-Datei C:\home\cc\qq\avr\qqtslcd.hex
- Verbinde Programmieradapter mit Mikroprozessor
- Spannung an Mikroprozessor ein (5V, versorgt MAX238)
- Wähle an AVR und ATmega8535

Damit lassen sich die Konfigurationsbits auslesen.

Und damit lässt sich ATmega8535, ATmega32 programmieren.

Hinweis: Der MAX238 invertiert die RS232-Signale (das wird für RS232-Verbindungen so gebraucht.) Deswegen die Invertierungsbits im Ponyprog entsprechend setzen. **Das muss bei jedem Start von Ponyprog (2.07a Beta) neu eingestellt werden.**