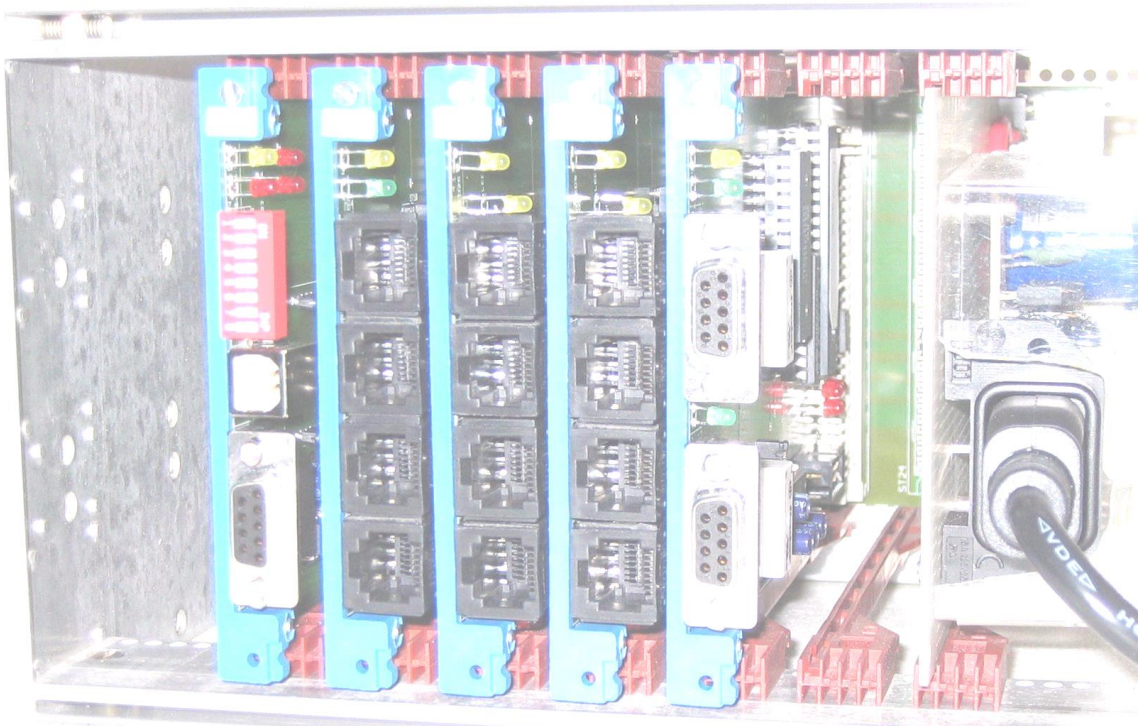


MoBaSbS

Modellbahn-Selbstbau-Steuerung

Handbuch

zur Version 4



Juli 2018

Grundlage des Handbuches sind die Internet-Seiten von

Patrick Urban
Lotharstr. 176
57057 Duisburg
0203 / 35 22 87

Karsten Thüner
Stettiner Str. 57
45739 Oer-
Erkenschwick
02368 / 696491

Uwe Mäusezahl
0621 / 4375658

sowie die Beiträge 1700 bis 10595 aus dem MoBaSbS-Forum.

Zusammenstellung und Bearbeitung durch:

Klaus Himstedt
Saffeweg 49
38304 Wolfenbüttel
05331 / 904319
klaus.himstedt@t-online.de

Folgende Themen werden im Handbuch nicht oder nur eingeschränkt beschrieben:

- Booster
- MoBaTL (Modellbahn-Teileliste) von Heinrich Bronstering / Edgar Berner
Datenbank zur Erfassung, Berechnung und Bestellung von elektronischen Bauteilen bei Reichelt
- SPC (StellPultController) von Uwe Mäusezahl
- PM_Dec_TRIAC von Marc Höner
Ansteuerung von Märklin-Weichen mit Wechselspannung
- PM_Enc_Opto von Marc Höner
- VRC (Voice Recognition Controller) von Günter Lintzmeyer
Steuerung der Modellbahn mittels Sprachbefehlen
- PM_SDEC_V3.0 Servodecoder mit Ansteuerung zur Herzstückpolarisierung von Norbert Steiniger
- 24-fach Decoder von Günter Lintzmeyer
- SDEC8 mit Herzstückpolarisation von Uwe Mäusezahl
- PM_TDOE (Triac-Decoder incl. Opto-Encoder) von Klemens Lichter
- SLC (Simple Lok Controller) von Günter Lintzmeyer
- WMC / WOM (Wireless Module Controller mit TV-Fernbedienung) von Günter Lintzmeyer
- Decoder mit WLAN von Günter Lintzmeyer

Es steht jedem MoBaSbS-Anwender frei, Kapitel zu ändern oder zu ergänzen. In diesen Fällen bitte ich um entsprechende Information.

Das Handbuch ist ausschließlich für den privaten Gebrauch bestimmt. Die kommerzielle Verwendung des Handbuches MoBaSbS oder Teilen davon ist nicht gestattet.

Inhaltsverzeichnis

0	Änderungsübersicht	1
1	Einleitung	4
2	Voraussetzungen	5
3	Platinen	5
4	Frontplatten	6
5	Linkliste	6
6	Architektur	7
6.1	Übersicht.....	7
6.2	Überblick Spannungsversorgung.....	9
6.3	RS485-Bus.....	11
7	Basis-Module	13
7.1	BMC.....	13
7.2	USC.....	17
7.2.1	Anschluss von Nicht-EDITS-Boostern.....	21
7.3	IFC.....	22
7.4	Netzteil.....	29
7.5	HDC.....	30
7.5.1	5V-Display-Variante.....	30
7.5.2	3,3V-Display-Variante.....	37
7.6	DPC.....	44
7.6.1	RC5-Variante.....	44
7.6.2	MoBaSbS-Variante.....	45
7.7	PMC.....	49
8	Peripherie-Module	51
8.1	Software-Varianten, Prozessortausch und Adapterplatinen.....	51
8.2	Decoder.....	55
8.2.1	FET-Decoder.....	55
8.2.2	ULN-Decoder.....	57
8.2.3	4-fach Servo-Decoder.....	58
8.2.4	8-fach Servo-Decoder.....	61
8.2.5	LDEC-Decoder.....	63
8.2.6	PM-Uni-Decoder.....	71
8.3	Encoder.....	74
9	Service	79
9.1	µC-Programmierboard.....	79
9.2	ISP-Programmer.....	82
9.3	Backplane.....	85
9.4	Verteiler Con888.....	86
10	Microcontroller programmieren	87
10.1	Hardwareumgebung.....	87
10.2	Programmierungsumgebung.....	87
10.2.1	Verzeichnisstruktur.....	87
10.2.2	Transfer der *.hex- und *.eep-Dateien zum Controller.....	87
10.2.3	Fuse-, Configuration- und Security-Bits.....	89
11	Inbetriebnahme	95
11.1	Vorgehensweise.....	95
11.2	Häufig auftretende Fehler.....	95
11.3	Testumgebung.....	96
12	MoBaSbS-Konfigurationstool	97
12.1	Installation.....	97
12.1.1	Tcl/Tk-Interpreter installieren.....	98
12.1.2	Konfigurationstool installieren.....	100
12.2	Programmstart.....	101
12.3	Menüs und Werkzeugleisten.....	104
12.3.1	Menüs.....	104
12.4	Konfiguration.....	105
12.4.1	Werkzeugleiste 1.....	105
12.4.2	Werkzeugleiste 2.....	105
12.4.3	Audio-Unterstützung.....	106

12.4.4	Autom. MBS laden.....	106
12.4.5	Konfig. ComPort (Schnittstelle zur MoBaSbS).....	106
12.4.6	Autom. Verbinden.....	106
12.4.7	Erw. ComPort Optionen.....	107
12.4.8	Lok-Datenbank.....	107
12.4.9	DCC Programmier Konfiguration.....	107
12.4.10	Debug.....	108
12.5	Basis-Module.....	109
12.5.1	BM-Report.....	109
12.5.2	BMC.....	111
12.5.3	BMC Lok. Daten-Basis.....	112
12.5.4	IFC.....	116
12.5.5	USC.....	117
12.5.6	PMC.....	118
12.5.7	DPC.....	120
12.5.8	HDC.....	121
12.5.9	SPC.....	122
12.5.10	WMC.....	122
12.5.11	VRC.....	123
12.6	Peripherie-Module.....	124
12.6.1	PM-Report.....	124
12.6.2	8-fach Encoder.....	125
12.6.3	24-fach Encoder.....	126
12.6.4	8-fach Decoder.....	127
12.6.5	4-fach Servo-Decoder.....	129
12.6.6	8-fach Servo-Decoder.....	130
12.6.7	Lichtsignal-Decoder-Universal.....	130
12.6.8	Lichtsignal-Decoder.....	135
12.6.9	Lichtsignal-Decoder Light.....	137
12.6.10	USC-Funktions-Decoder.....	138
12.6.11	Licht-Decoder ‚Gas-Laterne‘.....	139
12.6.12	Multi-Funktions-Dekoder (PM-Uni).....	140
12.7	Werkzeuge.....	142
12.7.1	Ändern einer Modul-Adresse.....	142
12.7.2	BM - Selektiver Lok-Daten Import.....	144
12.7.3	BM - Selektiver Lok-Daten Export.....	145
12.7.4	PM - Modul-Typ ändern.....	146
	PM - Kontakt Übersicht.....	147
12.7.5	Lok Kontroll Panel.....	148
12.7.6	DCC Programmier.....	149
12.7.7	Speicherinhalt BM/PM-Module (µC-Speicher-Dump).....	155
12.7.8	µC Flash Update (Bootlader).....	156
12.8	Decoder Schnellzugriff und Test.....	158
12.9	Monitor.....	159
13	Anhang.....	161
13.1	Funktionen des HDC.....	161
13.1.1	Kompatibilität.....	161
13.1.2	Einleitung.....	161
13.1.3	Navigation.....	162
13.1.4	Betriebsmodus.....	162
13.1.5	Hauptmenü („Menü“-Taste).....	164
13.1.6	Peripherie-Menü („Peri“-Taste).....	169
13.1.7	Lokmenü („Loco“-Taste).....	170
13.2	Programmierung und Inbetriebnahme des USB-Adapters.....	171
13.2.1	USB Tutorial (von Holger Bernhard).....	171
13.2.2	Installation FTDI (von Jörg Schindler, Stand 12.02.2017).....	174
13.2.3	Installation MoBa PC ab 2015 mit Win10 (von Roman Kircher).....	190
13.2.4	MoBaSbS Notizen USB (von Roman Kircher).....	193
13.3	Informationen zu Betrieb mit Traincontroller /-programmier.....	195
13.3.1	Traincontroller bis einschl. Version 5.5.....	195
13.3.2	Traincontroller ab Version 5.8.....	196
13.3.3	Traincontroller ab Version 7.0 Silber und Gold.....	196
13.3.4	Trainprogrammier ab V7.0.....	197

13.4	Bootlader-Bedienungs-Anleitung (Lintzmeyer-Variante)	198
13.5	Timing Diagramm Motorola II-Protokoll	201
13.6	Informationen zum PM-Uni	203
13.6.1	Einleitung	203
13.6.2	Hardware	203
13.6.3	Einstellungen	204
13.6.4	Bootladerfunktion.....	205
13.6.5	PM-Uni mit De- und Encoderfunktion.....	205
13.6.6	PM-Uni mit Servofunktion.....	208
13.6.7	PM-Uni mit LDEC-Funktion	209
13.7	Beschreibung des Servodecoders mit Herzstückpolarisierung	211
13.7.1	Allgemeine Hinweise	211
13.7.2	Warum diese Schaltung?	211
13.7.3	Funktionsbeschreibung	212
13.7.4	Die Software	213
13.7.5	Schaltbild	215
13.7.6	Hardware und Platinen-Layouts	217
13.7.7	Hard- und Software-Bereitstellung (Download).....	219
13.7.8	Materialliste	219

0 Änderungsübersicht

Version Juli 2018	
Kapitel	Änderung
6.1	HDC POM ergänzt
6.2	Spannungsversorgung. 3,3V Variante Jumperstellung abweichend. Hinweis eingefügt
7.1	Korrektur MoBaSbS unterstützt Funktionen schaltend und tastend
7.2	USC um Funktionen F13-F20 und tastende Funktionen ergänzt. Hinweis auf DCC Software von Uwe Mäusezahl und POM
7.5.2	HDC 3,3V Variante ergänzt um POM und Drehgeberrichtung. Abwärtskompatibilität zum HDC 5V. Weitere Aufbauhinweise
9.1	uC Programmierbord. Hinweis nicht für ATMega 644 geeignet
10	Mikrocontroller Programmieren, ergänzt um ATMega 644
10.2.3	Ergänzt um Fuse-Bits ATMega644
12.5.1	Mehrere Hinweise, Default-Adresse 31 bei Basismodulen gilt auch für HDC. Pollgrenze bei mehreren HDC beachten
13	Anhang, Unterschiede der HDC Varianten erläutert
13.1.1	HDC Kompatibilität. Mischbetrieb der HDC Varianten möglich
13.1.3	Bild 3,3V Variante ergänzt
13.1.4	HDC Betriebsmodus und Optionen komplett überarbeitet wegen 3,3V Variante. Tastfunktion, POM, Umschaltung Drehgeberrichtung und Ausschaltoption bei Funktionen tastend

Version Februar 2018 (nicht veröffentlicht)	
Kapitel	Änderung
Einleitung	24-fach Decoder, SDEC8 mit Herzstückpolarisation, PM_TDOE, SLC, WOM / WMC, Decoder mit WLAN ausgeschlossen Hinweis auf ‚mfx wird nicht unterstützt‘ ergänzt
3	Aktuelle EAGLE-Versionen und Hinweise ergänzt
5	Linkliste aktualisiert
6.1	PM_UNI ergänzt
6.3	Kapitel RS485-Bus neu eingefügt
7	Kapitel von Hardware in Basis-Module umbenannt
7.2	Hinweis auf Schaltdecoder-Booster ergänzt Hinweis auf ‚mfx wird nicht unterstützt‘ ergänzt Hinweise auf CV17/18/29 entfernt Hinweis auf kleineren Basiswiderstand für Schaltransistoren ergänzt
7.3	Hinweis auf Homepage von Holger Bernhard entfernt Funktionierende Treiber-Versionen ergänzt Hinweis auf Rücksetzen des USB-Cips ergänzt Hinweis auf Beschreibungen von R. Kircher und J. Schindler ergänzt Löschen von COM-Port-Einträgen ergänzt Hinweis auf aktuelle IFC-SW-Version ergänzt
7.5	HDC-Variante (3,3V Display) als zusätzliches Unterkapitel neu eingefügt Hinweis auf Spiralkabel ergänzt
7.6.2	Hinweis auf fehlenden Kondensator an PIN21 (Aref) ergänzt
7.7	Entfällt (war WMC / WOM), dadurch Verschiebung aller folgenden Kapitelnummern
7.8 → 8	Kapitelüberschrift Peripherie-Module eingefügt, dadurch Verschiebung aller folgenden Kapitelnummern
8.1	Abschnitt auf Bootlader-Version V2 aktualisiert,

	Software PM_universal und LDEC-T ergänzt
8.2.1	Hinweis auf zentrale Gleichrichtung ergänzt
8.2.3	Weitere Hinweise auf Servo-Zucken ergänzt
8.2.5	Hinweis auf Software LDEC-T ergänzt
8.2.6	PM-Uni Abschnitt neu eingefügt
7.3 → 9	Kapitelüberschrift Service eingefügt, dadurch Verschiebung aller folgenden Kapitelnummern
9.1	Bild für erste Version ausgetauscht Neue Version des µC-Programmierboards (mit ATmega8) ergänzt
9.2	Kapitel in ISP-Programmer umbenannt Beschreibung für zwei ISP-Programmer ergänzt
9.3	Bezugsquelle Reichelt ergänzt
10	Überschrift geändert ATMega644 ergänzt
10.1	Kapitel überarbeitet
10.2.2	Kapitel umbenannt und neu strukturiert ATMEL Studio ergänzt
10.2.3	Kapitel neu eingefügt Fusebits-Beschreibung ergänzt
12	Hinweis auf mobasbs.ini bei Startproblemen ergänzt
12.1.1	Hinweis auf 64-/32-bit Version von Tcl ergänzt. Audio-Installation unter WIN7 64-bit ergänzt
12.5.1	Adressbereich für PMC auf max. 6 geändert
12.6.5	Neues Feature ‚Inv. Drehrichtung‘ ergänzt
12.6.7	Hinweis auf Taktgenerator (LDEC-T) ergänzt
12.6.12	PM-Uni Abschnitt neu eingefügt
12.7.1	Peripherie Module: Bild aktualisiert
12.9	Link im Text entfernt
13.1.1	Hinweis auf HDC Version 4.03 ergänzt
13.2	Verweis auf Datei-Zulieferung geändert Kapitelstruktur geändert
13.2.2	Installation FTDI (von Jörg Schindler, Stand 12.02.2017) neu eingefügt
13.2.3	Installation MoBa PC ab 2015 mit Win10 (von Roman Kircher) neu eingefügt
13.2.4	MoBaSbS Notizen USB (von Roman Kircher) neu eingefügt
11.3 (alt)	Entfällt (war Konfiguration und Bedienung WMC / WOM), dadurch Verschiebung aller folgenden Kapitelnummern
13.3.1	Adressen ausführlicher benannt
13.3.2	Adressen ausführlicher benannt
13.4	Update der Beschreibung auf Version 2 (Stand 02.2012) BOOTRST korrigiert
13.6	PM-Uni Kapitel neu eingefügt
13.7	Beschreibung des Servodecoders mit Herzstückpolarisierung neu eingefügt

Version April 2011

Kapitel	Änderung
Einleitung	PM_Enc_Opto, VRC, PM_SDEC_V3.0 ausgeschlossen
7.3	FTDI-Treiberversion auf 2.6.2 aktualisiert
7.5	Verweis auf superflache Sockel und vergoldete Buchsenleiste eingefügt
7.6.2	Hinweis auf nur bitweises Auslesen ergänzt Hinweis auf Probleme bei Programmierung mit 12V ergänzt Hinweis auf Programmierung des USB-Chips ergänzt, Hinweis auf Programmierung des Funktionsdekoders ergänzt
7.9.1	Neues Kapitel SW-Varianten, Prozessortausch und Adapterplatinen eingefügt
7.9.4	Hinweis auf Einschaltzucken ergänzt

7.9.5	Hinweis auf Einschaltzucken ergänzt
7.11.1	Hinweis auf AVR Studio ergänzt, Adapterplatine Mega162 → Mega8 ergänzt
10.1.1	Kapitel auf neue Version 8.5.9.1 und ‚Snack‘ umgeschrieben
10.2.	Tabelle der Schaltbefehl-Symbole eingefügt
10.5	Reihenfolge der Kapitel an die Webseiten von K. Thüner angepasst
10.5	SPC und VRC ergänzt
10.6.5	Hinweis auf Speichern der eingestellten Werte in mobasbs.mbs ergänzt
10.7.6	Decoder Test: Bild und Text aktualisiert
10.7.8	Abschnitt neu eingefügt
11.1.4	Hinweis auf besonderes Verhalten der F12-Taste ergänzt
11.5	neues Kapitel Bootlader-Bedienungs-Anleitung eingefügt

Version Februar 2010

Kapitel	Änderung
Einleitung	PM_Triac ausgeschlossen
6. Architektur	Kapitel aktualisiert (wg. DPC)
7.6 DPC	Neues Kapitel DPC eingefügt
7.9.5 LDEC	Abschnitt komplett überarbeitet (neu: LDEC-Universal)
7.10 Encoder	Beschreibung Flackerschutzzeit (Entprellzeit) ergänzt
8.2.1 Verzeichnisstruktur	Hinweise auf entpacken der kompilierten Dateien ergänzt
10.4.9 DCC Programmer Konfiguration	Abschnitt eingefügt
10.5.8 DPC	Abschnitt eingefügt
10.6.7 LDEC- Universal	Abschnitt LDEC-Universal eingefügt
10.7.7 DCC- Programmer	DCC-Programmerbeschreibung eingefügt
11.4 Betrieb mit Traincontroller	Kapitel nach TC-Versionen aufgeteilt und Informationen ergänzt

Version April 2008

Kapitel	Änderung
alle	Überarbeitung aufgrund der Version 4

1 Einleitung

Die MoBaSbS richtet sich an engagierte Bastler, die Spaß an der Eisenbahn und an der Elektronik haben. Es wurde eine Elektronik entwickelt, die in Verbindung mit dem Steuerungsprogramm Railroad & Co (Traincontroller) Verbraucher einer Modellbahnanlage (Weichen, Lampen, Signale, usw.) ansteuert und Rückmeldungen von der Anlage oder dem Gleisbildstellpult an den Rechner zurückgibt. Die Steuerung wurde modular entwickelt. Es besteht eine klare Trennung zwischen Fahren, Schalten und Melden. Das Modul Fahren kann durch ein oder mehrere Handfahrpulte unterstützt werden. Damit ist ein komplettes System vorhanden, das seine Anfänge 1999 hatte und seitdem immer weiter verbessert wird.

Die MoBaSbS kann Lokdecoder für das MM1/2 und DCC ansteuern. Das MFX-Format wird nicht unterstützt. Lokomotiven mit MFX-Decoder werden als ‚normale‘ MM1/2-Lokomotiven betrachtet.

In einem geschützten MoBaSbS-Bereich sind alle notwendigen Schaltpläne, Bestückungslisten, Quellcodes und weitere Informationen abgelegt. In einem geschlossenen Forum gibt es die Möglichkeit andere MoBaSbS-Bastler zu kontaktieren.

Der geschützte MoBaSbS-Bereich und das Forum können nur nach Freigabe/Einladung von Patrick Urban erreicht werden.

Die MoBaSbS wird nicht als Produkt vertrieben, es besteht ausschließlich die Möglichkeit des Bausatzes.

2 Voraussetzungen

Für den Zusammenbau sind Grundkenntnisse der Elektronik hilfreich:

- Lesen von Schaltplänen
- Löten von elektronischen Bauteilen und ICs
- Kathode einer Diode
- gepolte und ungepolte Kondensatoren
- Umgang mit ICs

Informationen zu den Grundlagen der Elektronik und Löttechnik können im Web gefunden werden.

Ebenso nützlich sind Kenntnisse im Umgang mit Software-Quellen und die Programmierung von μ Controllern:

- *.asm und *.hex-Dateien
- Laden von Daten in μ Controller

Als Werkzeug hat sich als sinnvoll erwiesen:

- Vielfach-Messgerät (Widerstand, Spannung, Strom)
- Lötstation (ca. 40W) mit Temperaturregelung und
 - Kleine Lötspitze (ca. 0,8mm, an SMD-Bauteilen)
 - Normale Lötspitze (ca. 1,2mm, größere Lötunkte an Leistungsteilen)
- Elektronik-Lot 0,5mm
- Entlöt-Litze / Entlöt-Saugpumpe
- Silberdraht 0,8mm (oder abgeschnittene Beinchen für Durchkontaktierung und Lötbrücken)
- Mini-Bohrmaschine mit Bohrständer
- Platinen-Bohrer 0,8 bis 1,1mm
- Pinzette (für SMD-Bauteile, einfache Ausführung)
- Seitenschneider
- Balsa-Holz-Messer (zum Kratzen)
- Crimp-Zange für RJ45-Stecker (Empfehlung: MPCZ 8-8 von Reichelt), alternativ konfektionierte Kabel (z.B. MPFK 8-8 10m)

Hilfreich sind weiterhin:

- Lupe
- Platinenhalter („dritte Hand“)
- Lötlack SK10
- Plastik-Spray (zur Versiegelung)

Nicht notwendig ist:

- ESD-Matte (wenn Grundregeln des Umganges mit elektronischen Bauteilen eingehalten werden)

3 Platinen

Die Platinen der MoBaSbS wurden mit dem Programm EAGLE V3.55 der Firma cadsoft erstellt.

Der PM_UNI benötigt eine aktuellere Version:

Eagle 7.6.0 für 32bit-Systeme
Eagle 8.5.3 für 64bit-Systeme

Auf den Internetseiten von cadsoft kann man sich eine kostenlose Version von EAGLE downloaden. Die 64bit-Version erfordert eine Anmeldung mit E-Mail-Adresse.

4 Frontplatten

Das Projekt MoBaSbS bietet keine fertigen Frontplatten an. Sie können aber leicht in Eigeninitiative hergestellt werden. Einfach zu bearbeiten sind Polystyrol- oder Plexiglas-Platten, auf die man ein am PC entworfenes Layout überträgt.

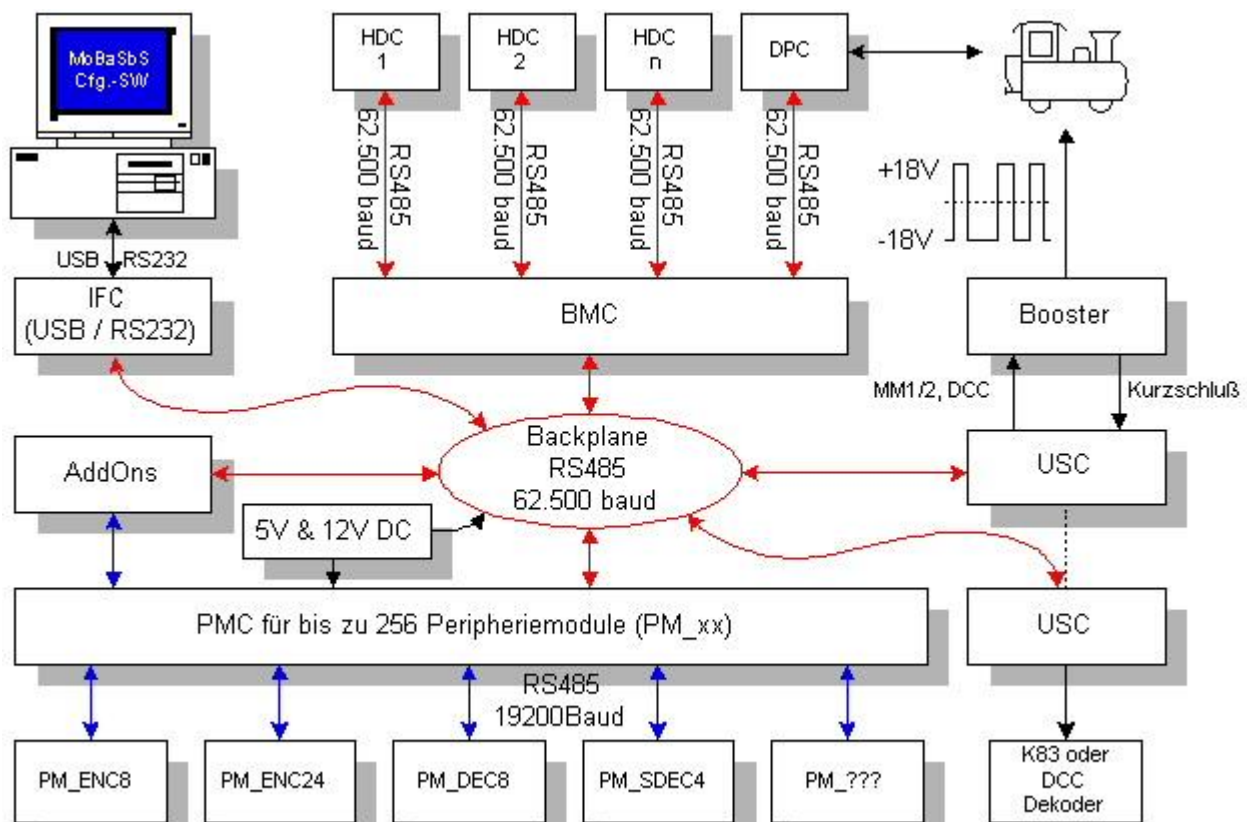
Für die professionelle Herstellung kann man sich zum Beispiel bei Fa. Schäffer informieren (siehe Kap. 5). Dort gibt es auch ein Programm zum Download, mit dem sich normgerechte Frontplatten erstellen lassen.

5 Linkliste

Homepage Patrick Urban	www.macurb.de
Homepage Karsten Thüner	http://mobasbs.thuener-oe.de
Archiv Uwe Mäusezahl	www.abhaeusler.de/Maeusi/www.die-maeusis.de/pmuni.html
ATMEL-Studio	https://www.atmel.com/products/AVR/ https://www.microchip.com/avr-support/atmel-studio-7
Conrad	https://www.conrad.de
DDW/DDDL	http://home.snafu.de/mgrafe/
EAGLE	https://www.autodesk.de/products/eagle/overview https://www.autodesk.de/products/eagle/free-download
Frontplatten	http://www.schaeffer-apparatebau.de
ISP-Adapter, Schaltplan	http://www.lancos.com/
ISP-Programmer Tremex/Diamex	https://www.diamex.de/dxshop/USB-ISP-Programmer-fuer-Atmel-AVR-Rev2
Lenz	http://www.digital-plus.de/index.php
PonyProg	http://www.lancos.com/
Prolok1	http://tt.borrmanns.de/prolok.html
Railroad & Co (Traincontroller)	http://www.freiwald.com/
Reichelt	https://www.reichelt.de
Virtueller COM-Port-Treiber (VCP)	http://www.ftdichip.com/
Tcl/Tk Interpreter	https://www.activestate.com/activetcl
Direkter Link zu Tcl/Tk-Interpreter	https://www.activestate.com/activetcl/downloads
Direkter Link zum Tcl-Zusatzpaket ‚Snack‘	http://www.speech.kth.se/snack/dist/snack2210-tcl.zip

6 Architektur

6.1 Übersicht



Architektur der MoBaSbS

Backplane

An die Backplane werden alle Basismodule (rote Linien) über Steckverbinder angeschlossen.

Bus-Systeme

Der Bus, der die Basismodule untereinander verbindet, heißt X-Bus (angelehnt an die LENZ-Beschreibung). Es handelt sich um einen RS485-Bus mit einer Datenrate von 62.500 Baud.

Der Bus, der die Peripherie-Module verbindet (blaue Linien, ausgehend vom PMC), heißt P-Bus (Peripherie-Bus). Es ist auch ein RS485-Bus mit einer Datenrate von 9.600 Baud.

Beschreibung RS485-Bus siehe Kap. 6.3

BMC

Verwaltet wird die Kommunikation durch den BMC (Basis-Modul-Controller). Dieser pollt zyklisch die am Bus angeschlossenen Komponenten und fragt dabei an, ob etwas "Neues" vorliegt. Meldet sich eine Komponente mit einer Information zurück, so wird diese an die anderen Komponenten und - über den IFC (Interface-Controller) an den PC weitergegeben.

USC

Der USC (Universal-Signal-Controller) erzeugt zum einen die Loksteuerbefehle (im Märklin-Motorola- und/oder DCC-Format). Zum anderen werden mit dem USC die Schaltbefehle für DCC- oder MM-Schaltdecoder generiert. Alle Befehle werden über einen Booster ans Gleis gelegt. Es wird empfohlen, für die Loksteuerbefehle und die Schaltbefehle je einen eigenen USC zu benutzen, damit es bei der Abarbeitung der Befehle zu möglichst geringen Verzögerungen kommt.

IFC

Die Kommunikation mit dem PC erfolgt über den IFC (Interface-Controller). Es können sowohl die serielle Schnittstelle (RS232) oder über eine Adapterplatine auf dem IFC eine USB-Schnittstelle angeschlossen werden.

Es wird das Lenz-Protokoll Ver.3 verwendet. Alle Befehle zur Decoder-Ansteuerung, Rückmeldung und Lokkontrolle werden unterstützt.

HDC

Der HDC (Hand-Daten-Controller) dient der Steuerung von Lokomotiven und Peripherie-Elementen. Es können:

- Lokomotiven ausgewählt und gesteuert werden
- Lokdecoder-Programmierung im laufenden Betrieb (DCC) mittels POM (programming on main)
- Peripherie-Elemente ausgewählt und geschaltet werden
- Rückmeldungen kontrolliert werden
- Servos eingestellt werden

DPC

Der DPC (Decoder-Programmier-Controller) dient zum Programmieren der ConfigurationVariables (CVs) von DCC-Decodern.

PMC

Der PMC (Peripherie-Modul-Controller) dient zum Anschluss von Peripheriemodulen, wie Encodern (für Rückmeldungen) und Decodern (für Schaltaufgaben).

PM_xxx

Über die PM (Peripherie-Module) werden die Elemente auf der Anlage angesteuert und Rückmeldungen eingelesen. Es gibt folgende Typen:

PM_ENC8 / PM_ENC24

Peripheriemodul-Encoder dienen zum Einlesen von Rückmeldungen. Es gibt Varianten für 8 und 24 Rückmeldungen pro Modul.

PM_DEC8

Peripheriemodul-Decoder steuern Schaltelemente. Es gibt eine MOSFET- und eine ULN-Variante. Beide haben jeweils 8 Ausgänge (damit lassen sich vier Weichen ansteuern) und unterscheiden sich durch den maximalen Schaltstrom.

PM_SDEC

Peripheriemodul-Servo-Decoder steuern Servo-Antriebe. Es gibt Varianten für 4 oder 8 Servos.

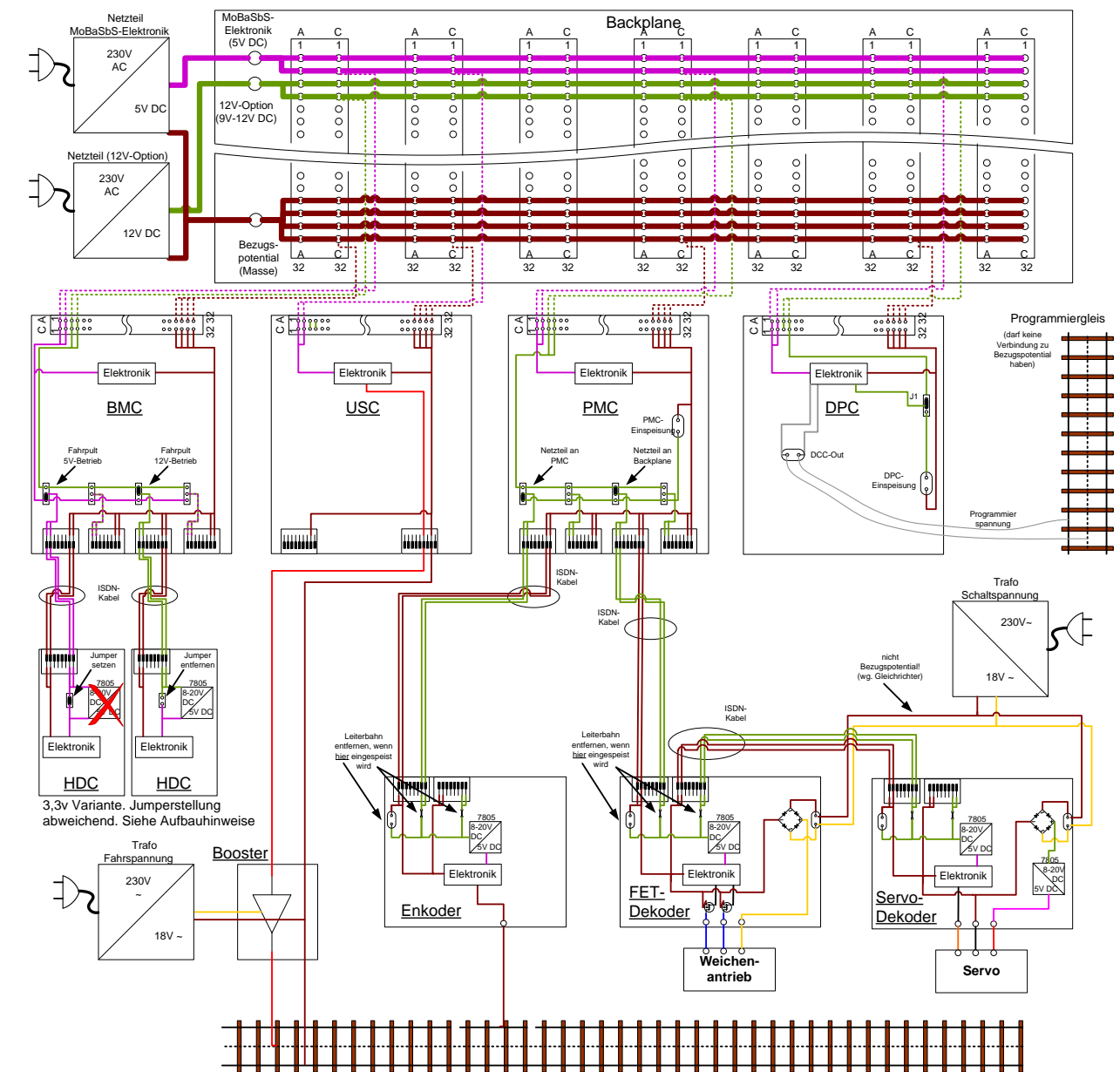
PM LDEC

Der Peripherie-Lichtsignaldecoder steuert die LEDs von Signalen an. Es können verschiedene Signalbilder dargestellt werden, wobei die Umschaltung über ein Ab- und Aufdimmen realisiert ist.

PM UNI

Mit dem Universal-Decoder können ebenfalls PM_ENC8, PM_DEC8 (MOSFET und ULN), PM_SDEC (4 und 8) und PM_LDEC aufgebaut werden. Zusätzlich können Erweiterungsplatinen für Stromfühler und Herzstückpolarisation montiert werden. Der Vorteil liegt darin, dass nur ein Platinentyp benötigt wird, der entsprechend der gewünschten Funktion zu bestücken ist.

6.2 Überblick Spannungsversorgung



Spannungsversorgung der MoBaSbS (Sicht auf die Bestückungsseite)

Die Spannungsversorgung gliedert sich in folgende Bereiche:

Bezugspotential (Massepotential)

Prinzipiell gilt, dass die gesamte Anlage ein Bezugspotential hat, das auf die Schiene bezogen ist. Durch Messung kann man feststellen, ob von der Schiene bis zum letzten Decoder eine niederohmige Verbindung besteht. Auf der Backplane liegt die Masse auf dem Steckverbinder A29 –A32 und C29 – C32.

MoBaSbS-Elektronik

Die Elektronik der Basismodule benötigt 5V DC. Diese wird über ein stabilisiertes Netzteil erzeugt. Das Netzteil sollte über eine genügende Spannungs-Steilheit verfügen, damit die Elektronik im Moment des Einschaltens nicht instabil wird. Die Einspeisung erfolgt über die Backplane (Steckverbinder A1, A2, C1 und C2).

Fahrspannung

Für die Fahrzeuge wird über eine Trafo- / Booster-Kombination die Digitalspannung ans Gleis geliefert. Abhängig vom Boostertyp können hier Ringleitungen sinnvoll sein.

Peripheriemodul-Versorgung

Die Elektronik der Peripheriemodule benötigt 5V DC. Es gibt drei Möglichkeiten, die Spannung zu den Bauteilen zu führen:

bei kurzen Kabeln (MoBaSbS-Elektronik)

Bei kurzen Kabeln ist der Spannungsverlust so gering, dass die Verbindung direkt von den 5V DC der MoBaSbS-Elektronik erfolgt. Hierfür werden die Jumper auf dem PMC auf die 12V-Option gesteckt. Auf der Backplane muss dann eine Verbindung von den 5V DC (MoBaSbS-Elektronik) zu der 12V-Option hergestellt werden. Auf den Decodern und Encodern ist der Spannungsregler (7805) nicht zu bestücken und mit einem Draht zu überbrücken. Die hierbei zu erreichende maximale Kabellänge kann durch Messungen oder Ausprobieren ermittelt werden.

bei langen Kabeln (12V-Option)

Bei langen Zuleitungen oder sehr vielen Peripheriemodulen ist der Spannungsverlust so groß, dass auf den Peripheriemodulen weniger als 5V ankommen. Für diesen Fall muss eine höhere Spannung eingespeist werden, die dann auf den Peripheriemodulen auf die benötigten 5V reduziert wird. Hierfür wird ein separates Netzteil benötigt. Es sind Werte ab 9V möglich. Als sinnvoll haben sich 12V erwiesen. Das Netzteil sollte 2A bis 4A liefern und über eine genügende Spannungs-Steilheit verfügen, damit die Elektronik im Moment des Einschaltens nicht instabil wird.

Die Einspeisung dieser Spannung erfolgt entweder auf der Backplane (Steckverbinder A3, A4, C3 und C4) oder dem PMC. Auf dem PMC ist über die Jumper auszuwählen, ob die 12V-Option von der Backplane oder dem PMC-Einspeisepunkt in das ISDN-Kabel geführt werden soll. Die Spannung wird dann zu den Peripheriemodulen geführt und dort per Spannungsregler auf 5V geregelt.

Ringleitung

Es kann sinnvoll sein, die 12V-Spannung nicht nur auf Backplane oder PMC einzuspeisen, sondern als Ringleitung unter der Anlage vorzusehen. Die Spannung wird dann direkt an den Peripheriemodulen eingespeist. In diesem Fall müssen zwei Leiterbahnen durchtrennt werden, die zum RJ45-Stecker führen.

HDC-Versorgung

Beide Versionen des HDC haben, wie die Peripheriemodule auch, eine 12V-Option. Im HDC kann ein 5V Festspannungsregler eingesetzt werden, der über einen Jumper aktiviert oder kurzgeschlossen wird. Die Jumperstellungen in der 5V und der 3,3V Variante sind unterschiedlich. Die Aufbauhinweise liefern detaillierte Auskunft.

Auf dem BMC befindet sich ebenfalls ein Jumper, mit dem ausgewählt wird, ob die 5V der MoBaSbS-Elektronik oder die 12V-Option genutzt wird.

Bei Nutzung der 12V-Option muss die Spannung über die Backplane zugeführt werden.

Meldespannung

(im Prinzipbild nicht eingetragen)

Für aktive Melder (z. B. Hallsensoren) kann am PM_ENCx eine externe Spannung angeschlossen werden.

Schaltspannung (Licht, Weichen, LEDs)

Für den Betrieb der Verbraucher wird eine Spannung benötigt, die direkt an den Peripheriedecodern angeschlossen wird.

Abhängig von den persönlichen Möglichkeiten und der Art der Verbraucher, sind folgende Varianten denkbar:

stabilisiertes Netzteil

Mit einem stabilisierten Netzteil erhält man eine betriebssichere Spannung, die bis zum angegebenen Ausgangsstrom kaum nachgibt. Damit lassen sich z.B. schwergängige Weichen zuverlässiger schalten. Auf den Peripheriemodulen braucht der Gleichrichter nicht bestückt zu werden. Ob ein Festspannungsregler einzusetzen ist, ist von der Ausgangsspannung des Netzteiles und der Spannung, die der Verbraucher benötigt, abhängig.

Trafo

Die üblichen Modellbahntrafos liefern ca. 18V ~. In der Regel haben diese Trafos eine so genannte "weiche" Kennlinie, d.h. bei Schalten der Weichenantriebe bricht die Spannung des Trafos zusammen und es kann bei schwergängigen Weichenantrieben zu Problemen im Schaltverhalten kommen. Die Wechselspannung des Trafos muss immer gleichgerichtet werden. Abhängig von den Verbrauchern ist ggf. ein Festspannungsregler einzusetzen.

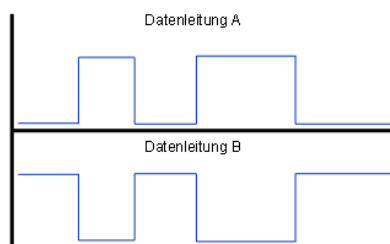
Bei beiden Möglichkeiten kann die Spannungsversorgung zentral aufgestellt sein. Ebenso kann die Gleichrichtung zentral erfolgen. Die Spannung kann dann als Ringleitung zu allen Peripheriemodulen geführt werden.

6.3 RS485-Bus

Der RS485 Bus ist extrem robust, nominell sollten 1800 Meter kein Problem sein. Aufgrund seiner Robustheit funktioniert es meist auch wenn man von der Spezifikation abweicht.

Im Zweifel (d.h. spätestens wenn Probleme auftauchen) hält man sich aber besser an die Spezifikation: Die Teilnehmer müssen in einer Linie liegen, unvermeidliche Stichleitungen sollten so kurz wie möglich sein (maximal 20cm). Also ist eine Baum oder Stern Topologie nicht erlaubt.

Die Leitung muss zwingend Twisted-Pair sein, weil die ganze Stabilität des Systems darauf beruht, dass eine Störung z.B. durch einen Lok-Antrieb auf beide Leitungen gleich einwirkt. Über die eine Leitung wird das Signal unverändert übertragen, auf der anderen Leitung wird das Signal invertiert. Wenn eine 1 übertragen werden soll hat die eine Leitung +5V, die andere Leitung 0V. Bei einer 0 hat die erste Leitung 0V, die andere +5V. Der Empfänger wertet nur die Differenz die beiden Signale aus. Durch dieses Verfahren werden Störungen nahezu eliminiert



RS485 Signalverlauf

Die Verbindung zwischen den Teilnehmern erfolgt bei der MoBaSbS über eine Zweidrahtverbindung. Das bedeutet, dass Daten nicht gleichzeitig in beide Richtungen übertragen werden können. Eine Übertragung erfolgt immer abwechselnd in die eine oder in die andere Richtung.

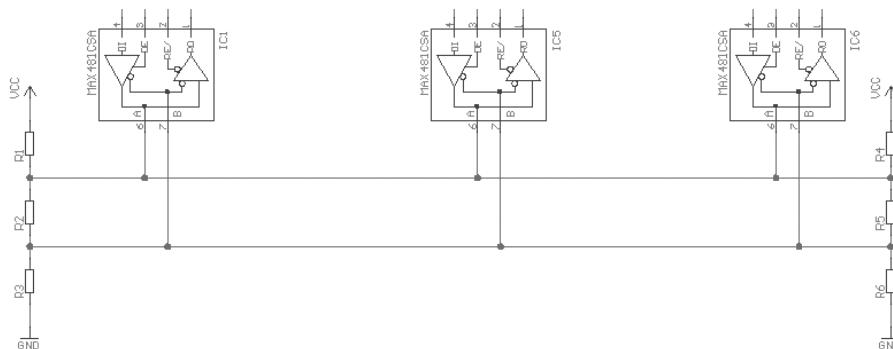
Die am häufigsten verwendeten RS485 Treiber sind der MAX485 von Maxim, der LTC485 von Linear Technology und der SN75176 von Texas Instruments.

Alle diese Treiber sind zueinander Pin- und Funktionskompatibel, unterscheiden sich jedoch in der maximalen Datenrate und der Anzahl der Teilnehmer auf einem Bus. Je nach Ausführung kann z.B. der MAX485 mit bis zu 2,5MBit arbeiten, der MAX483 dagegen nur mit 250kBit.

Die Anzahl der Teilnehmer am Bus ist durch die Eingangskapazität der RS485 Treiber begrenzt. Typisch funktionieren bis zu 32 Teilnehmer problemlos, dies hängt aber von der Marke/Type aller (!) Transceiver am Bus ab. D.h. man sollte nur Transceiver mit möglichst kleiner Last verwenden und Typen, die für die gewünschte Datenrate optimiert sind.

Verwendet man den 485-Chip können bis zu 32 Module angeschlossen werden. Alternativ kann ein 487-Chip (z.B. Max 487 ECPA) eingesetzt werden. Damit können an einen Peripherie-Strang bis zu 128 Module betrieben werden. Die 485er- können mit den 487er-Chips gemischt betrieben werden. Hierbei muss beachtet werden, dass das schwächste Glied der Kette die max. Anzahl bestimmt.

Die folgende Grafik zeigt wie der RS485 Bus aufgebaut ist.



Aufbau RS485-Bus

Man sieht hier die beiden Leitungen des RS485 Bus. Eine Leitung verbindet jeweils die A-Anschlüsse der Treiber, die zweite Leitung verbindet die B-Anschlüsse.

Die Widerstände R1 bis R6 sind Abschlusswiderstände und müssen am Ende der Leitungen montiert werden, damit Reflexionen vermieden werden. Bei geringen Entfernungen < 20m werden folgende Widerstände eingesetzt:

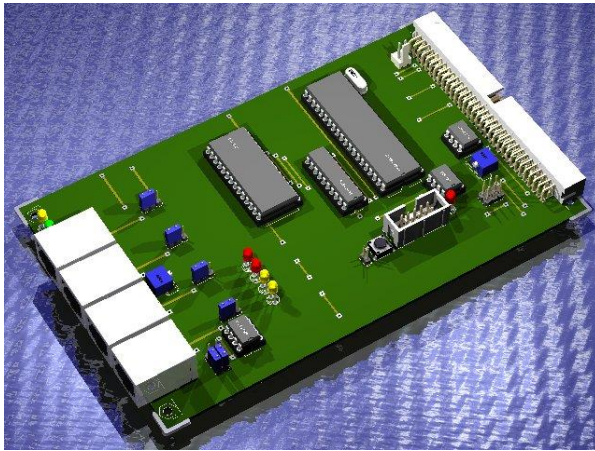
R1, R3, R4, R6: 100kOhm

R2, R5: 1kOhm

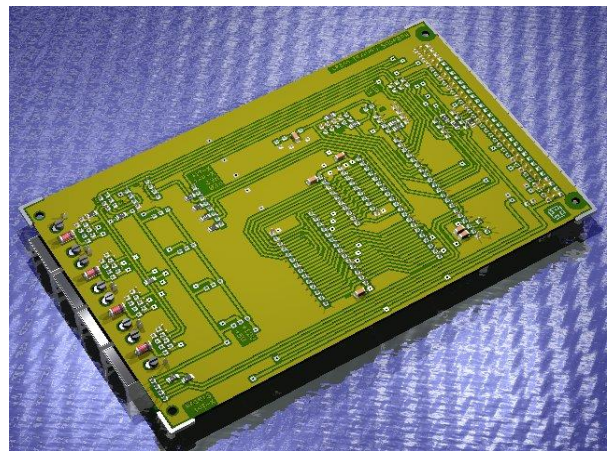
Je nach Datenrate und Leitungslänge kann es notwendig sein andere Werte einzusetzen. Sehr häufig werden für R2, R5 120 Ohm empfohlen. Diese sind bei der MoBaSbS an Pin 4+5 des RJ45 Steckers (die beiden mittleren Pins) anzuschließen.

7 Basis-Module

7.1 BMC



BMC (Bestückungsseite)



BMC (Lötseite)

Der Basis-Modul-Controller (BMC) erfüllt zwei wesentliche Aufgaben:

Zum einen sorgt er für den Datenaustausch zwischen den zentralen Modulen der MoBaSbS. Dazu polt der BMC zyklisch alle an ihm angeschlossenen Module und fragt deren Status ab.

Zum anderen verwaltet er die Lokdaten. Hierbei werden bei einer Änderung des Status' einer Lok deren Informationen im Speicher des BMCs abgelegt oder aber bei einer Anfrage eines Moduls z.B. nach dem Status einer Lok oder nach deren Namen die gewünschten Daten vom BMC an das anfragende Modul gesendet.

Aufgrund der erweiterten Funktionalität in Version 4 müssen das RAM (IC7 oder IC8), das Latch (IC6) und das EEPROM (IC15) bestückt sein.

Die Parameter eines Lokdatensatzes werden im seriellen EEPROM (ST24C256) gespeichert und nach dem Anschalten der MoBaSbS aus dem EEPROM in das statische RAM kopiert. Von dort wird während des laufenden Betriebs dann auf alle Parameter zugegriffen. Die Parameter einer Lok lassen sich in 4 Gruppen aufteilen:

1. Lokparameter, die zum Betrieb unbedingt notwendig (n) sind bzw. dem Komfort (k) dienen. Diese Parameter können direkt über den HDC oder das MoBaSbS-Konfigurationstool eingegeben werden. Zu diesen Parametern gehören:
 - (n) Adresse der Lok (bis 255 bei MM2, bis 9999 bei DCC)
 - (n) Protokolltyp (MM2 oder DCC)
 - (n) Fahrstufenanzahl (14, 28, 128)
 - (k) Loknamen (20 Zeichen)

2. Lokparameter, die während des Betriebs verändert werden und deshalb vom Anwender nicht vordefiniert werden können. Dazu gehören:
 - Lokgeschwindigkeit
 - Lokrichtung
 - Funktionstastenstatus

3. Lokparameter, die Kontrollzwecken dienen. Dazu gehören:
 - Definition, ob eine Funktionstaste tastend oder schaltend behandelt werden soll.
 - Definition, an welcher Stelle der Datensatz innerhalb aller Datensätze angelegt werden soll. Mit diesem Parameter ist es möglich, bestimmte Loks zu gruppieren. Als Folge erscheinen die Loks beim Aufruf im HDC in direkter Folge.

4. Lokparameter, die derzeit eine statistische Funktion haben und im MoBaSbS-Konfigurationstool ausgewertet werden. Dazu gehören:
 - Lokgeschwindigkeit
 - Artikelnummer (6stellig)
 - Herstellerbezeichner (Märklin, Roco, Fleischmann, ...)
 - Definition des Loktyps (z.B. Dampf, Diesel, Elektro und entsprechende Untergruppen wie Schnellzuglok, Güterzuglok, Triebwagen, etc.)
 - Definition der Lokfunktionen zugeordnet zu den Funktionstasten (z.B.: F1 = Licht vorne, F2 = Dampf, ... , F12 = Geräusch-Kohleschauflern)

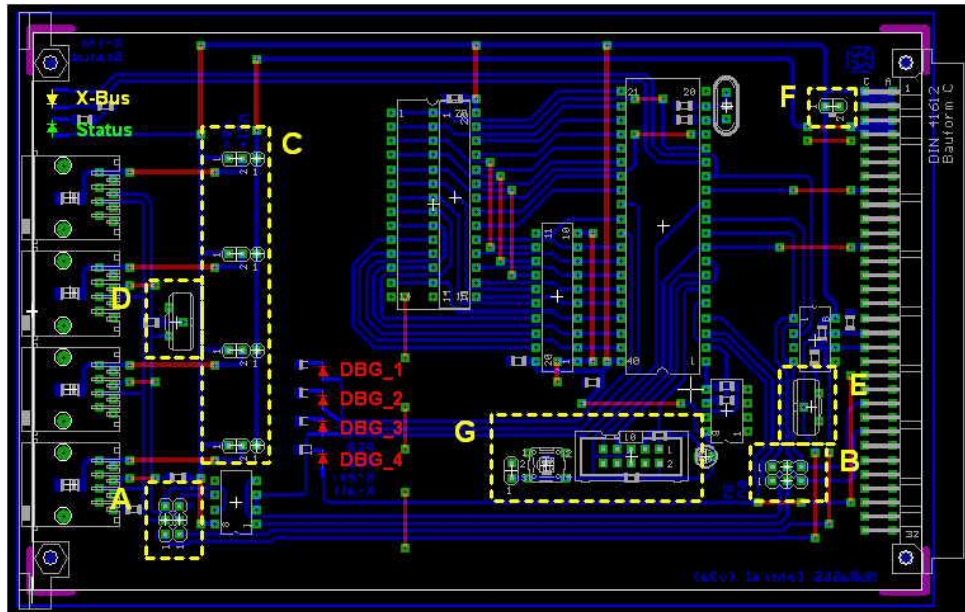
Im Speicher des BMCs können insgesamt ca. 500 Datensätze hinterlegt werden. Es können nur die Loks angesprochen werden, die auch über das MoBaSbS-Konfigurationstool definiert wurden. Weitere Informationen zur Konfiguration der Lokdatensätze sind der Beschreibung des MoBaSbS-Konfigurationstools (siehe Kap. 12) zu entnehmen.

Wie alle anderen Hauptmodule auch, ist er mit einem Mikrokontroller vom Typ ATmega162 bestückt. Über die vier RJ45 Buchsen links im Bild, lassen sich HDCs anschließen. Werden mehr als vier Anschlüsse benötigt, kann mit Verteilern (Y-Splitter) die Anzahl der Anschlussmöglichkeiten erhöht werden.

Aufbauhinweise

- Bei industriell gefertigten Platinen ist zu beachten, dass die Bauteilbezeichnungen teilweise falsch sind. Als Referenz ist deshalb immer die Bauteilbezeichnung der Platinen-Layouts aus dem geschützten MoBaSbS-Bereich zu verwenden.
- Als RAM kann sowohl eine schmale, als auch eine breite Version verwendet werden. Dafür ist hier eine alternative Bestückung vorgesehen.
- Obwohl zwei Arten von RS485 Treibern in der Bestückungsliste aufgeführt werden (75176 und MAX485), wird die Verwendung des MAX485 Bausteins empfohlen.
- Aufgrund der erweiterten Funktionalität in Version 4 müssen das RAM (IC7 oder IC8), das Latch (IC6) und das EEPROM (IC15) bestückt sein.
- Die beiden LEDs oben links im Bild zeigen den Kommunikationsstatus der Module untereinander (gelbe LED), sowie den aktiven Zustand (grüne LED) der MoBaSbS an. Die 4 weiteren Dioden (rechts unterhalb von "C") können ebenfalls in eine Frontplatte montiert werden und dienen Debugzwecken.
- Die Platine ist so ausgelegt, dass sie in ein 19"-Gehäuse mit Frontplatte passt.

Einstellmöglichkeiten



BMC. Platinen-Layout

Die gelben Markierungen im Layout zeigen folgendes an:

A

Hier kann definiert werden, ob die an der Frontseite befindlichen Buchsen von einem eigenen Treiber versorgt werden sollen oder ob der interne RS485 Bus nach vorn herausgeführt werden soll. Derzeit müssen hier zwei Jumper gesetzt werden, die die mittleren beiden Pins mit den unteren (mit 'i' gekennzeichneten) Pins vertikal verbinden. Der RS485 Treiber (IC 10/2), rechts neben den Jumpers, kann weggelassen werden.

B

Wird bei A der Treiber ausgewählt (die Jumper brücken die mittleren mit den oberen Pins), dann kann bei B festgelegt werden, ob beide Treiber (der für den internen und für den externen Bus) aus einer Quelle (2 Jumper bei B von Mitte nach Rechts (U0) gesteckt) oder von zwei unterschiedlichen Quellen (2 Jumper von der Mitte nach Links (U1) gesteckt) gespeist werden sollen. Derzeit wird dieser Jumper in keiner Position unterstützt und bleibt offen!

C

Hier kann eine Versorgungsspannung von +5V oder +9V bis +12V für jeden einzelnen Ausgang definiert werden. Die 4 Jumper werden dazu von der Mitte nach Links (+5V Option) oder nach rechts (höhere Spannung) gesteckt. Der HDC unterstützt beide Versorgungsoptionen. Um einen Spannungsabfall bei Kabellängen > 5m zu kompensieren, kann hier die höhere Spannung verwendet werden. Ein eigener Festspannungsregler im HDC sorgt dann wieder für die 5V Versorgung. Damit ein versehentliches Einstecken des HDCs in einen PMC anstatt des BMCs nicht zu einer Zerstörung des HDCs führt - an den PMC-Ausgängen liegen generell +9V bis +12V an -, sollte der HDC mit dem 5V Festspannungsregler ausgestattet werden und der BMC so gejumpt werden, dass an seinen RJ45 Buchsen die +9V bis +12V Spannung anliegt.

D und E

Diese Potis dienen zur Einstellung des Abschlusswiderstandes des internen und externen RS485 Busses. Die Einstellung sollte nicht in Mittenstellung sein, sondern besser mit Anschlag rechts (von den RJ45-Buchsen zur VGA-Leiste gesehen) betrieben werden. Das Poti an Position D hat durch die Jumperstellung von A jedoch keinen Einfluss.

F

Hier können +5V eingespeist werden, z. B. um ein Board außerhalb des Gehäuses zu programmieren.

G

Hier liegt die ISP Schnittstelle und der Reset-Taster. Über die ISP (In-System-Programmable) Schnittstelle lässt sich der μ C im System programmieren. Befindet sich die Karte nicht an der Backplane aufgesteckt, so kann am Punkt F eine 5V DC-Spannung eingespeist werden. Diese ISP-Schnittstelle findet man auf allen Basismodulen der MoBaSbS. Auf den Peripheriemodulen ist sie aus Platz- und Kostengründen nicht vorgesehen.

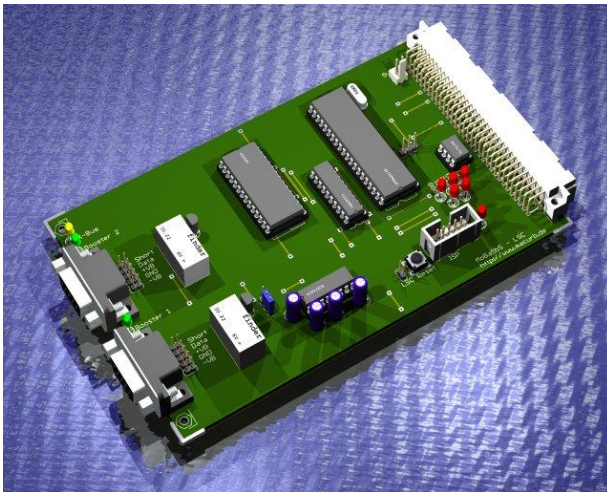
Reset

Der Reset-Taster auf jedem der Hauptmodule ist nur vorgesehen, um im Falle eines Falles einmal ein Modul separat zurücksetzen zu können. Auf dem BMC kommt dem Reset aber eine besondere Bedeutung zu, da hier das Auslösen des Rests zu einem Rücksetzen aller Basismodule führt. Dies ist bei den anderen Modulen nicht der Fall! Über den 2pol Pfostenstecker (links vom Reset-Taster) und ein Stück Kabel kann der Taster herausgeführt werden. Dadurch ist er leichter zu bedienen. Um ein sauberes Starten des μ Controllers nach dem Einschalten der Spannung zu gewährleisten, ist jeder μ C mit zwei Bauteilen beschaltet: Einem 10k Widerstand, der sich genau in der Mitte unterhalb des ISP-Wannensteckers befindet, und einem 100nF Kondensator, direkt unterhalb des Reset-Tasters. Angeschlossen (angesteckt) wird die BMC an die Backplane, die alle Module untereinander verbindet und über die die Kommunikation der einzelnen Module läuft.

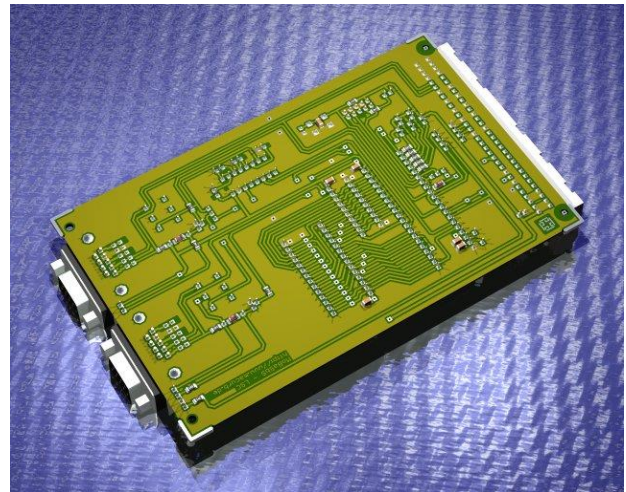
Software

Die aktuelle SW kann aus dem geschützten MoBaSbS-Bereich heruntergeladen werden.

7.2 USC



USC (Bestückungsseite)



USC (Lötseite)

Der USC wertet die Kommandos aus, die er über den zentralen RS485 Bus empfängt und setzt sie in ein Datenformat um, welches vom Booster verstärkt an die Lok- und Schaltdecoder, die aus der Gleisspannung versorgt werden, gesendet wird. Mögliche Quellen der Kommandos sind der PC und die Handfahrpulte (HDCs und WMCs).

Der USC löst die beiden bisherigen Module LSC (Lok/Loco-Signal-Controller) und SSC (Schalt/Switch-Signal-Controller) ab, da es nun mit einem Mikrocontroller-Code möglich ist, sowohl Lok- als auch Schaltdecoder-Protokolle zu generieren. Folgende Protokolle sind mit dem USC möglich:

- MM1-FD Funktionsdecoder (z.B. der Funktionsdecoder im Märklin Tanzwagen)
- MM1/2-SD Schaltdecoder (z.B. für die Märklin K83 Schaltdecoder)
- MM1/2-Lok Lokdecoder (14 FS, abwärtskompatibel zum MM1-Protokoll für Lokdecoder)
- DCC-SD DCC-Schaltdecoder
- DCC-Lok Lokdecoder (14, 28, 128 FS)

Der USC unterstützt kein MFX. Eine Implementierung ist auch nicht geplant.

Welche Protokolle der USC generieren soll, lässt sich über das MoBaSbS-Konfigurationstool einstellen. Die verschiedenen Konfigurationsmöglichkeiten sind in der nachfolgenden Tabelle aufgeführt.

Kombinationen\ Datenformate	MM1/2-Lok	DCC-Lok	MM-FD	MM-SD	DCC-SD
MM1/2- & DCC-Lok	Ja	Ja	Ja	Nein	Nein
MM-SD	Nein	Nein	Nein	Ja	Nein
DCC-SD	Nein	Nein	Nein	Nein	Ja
Kombi-MM	Ja	Ja	Ja	Ja	Nein
Kombi-DCC	Ja	Ja	Ja	Nein	Ja

USC-Konfigurationstabelle

Wie man der Tabelle entnehmen kann, ist es nicht möglich, den USC so zu konfigurieren, dass DCC- und MM-Schaltdecoder gleichzeitig unterstützt werden.

Von der Möglichkeit, Lok- und Schaltdecoderkommandos von nur einem Modul generieren zu lassen, wird abgeraten. Nur die getrennte Erzeugung durch zwei USCs kann gewährleisten, dass Lokbefehle in minimaler Zeit über das Gleis den Lokdecodern zugeführt werden können, ohne dass es durch Schaltkommandos zu einer verzögerten Abarbeitung von Lokbefehlen kommt.

Weiter sollte berücksichtigt werden, dass bei Konfiguration des USC als Schaltdecoder alle einlaufenden Schaltbefehle generiert werden, auch wenn diese an Schaltdecoder gerichtet sind, die gar nicht am USC angeschlossen sind. Für die Übertragung der Schaltbefehle ist ein separater (Schaltdecoder-) Booster notwendig, der nicht mit dem Ausgang des Lokdecoder-Boosters verbunden sein darf.

Der Mischbetrieb ‚MM1/2- & DCC-Lok‘ wurde mit folgenden Decodern getestet:

- MM Decoder: Märklin 6090x inkl. dem Decoder für den alten C-Sinus Motor)
- Decoder im Goliath Kran
- Spezialdecoder im neuen Kran mit Piezomotoren
- DCC Decoder: Kühn 145, Lopi 2, Lopi 3, Loksound 3.5 und einem Uhlenbrock Decoder. (Typ habe ich vergessen)

Testergebnisse mit dem neuen kompakten C-Sinus Motor und dessen Decodern liegen nicht vor.

Im DCC-Format können Lokadressen von 1 - 9999 genannt werden.

Es werden folgende DCC Formate unterstützt:

- 14 Fahrstufen verwendbar für Adresse 1-99 und der Basisadresse in CV 1)
- 28 Fahrstufen verwendbar für Adresse 1-99 mit der Basisadresse des Decoders in CV 1 verwendbar für Adresse 100-9999 mit erweitertem Adressbereich in CV 17 und 18
- 128 Fahrstufen verwendbar für Adresse 1-99 mit der Basisadresse des Decoders in CV 1 verwendbar für Adresse 100-9999 mit erweitertem Adressbereich in CV 17 und 18

Mögliche Funktionen

- F0,F1 – F4 (Funktionen der Gruppe 1)
- F5 – F8 (Funktionen der Gruppe 2)
- F9 – F12 (Funktionen der Gruppe 3)
- F12 – F20 (Funktionen der Gruppe 4)
- F21 – F28 (Funktionen der Gruppe 5)

Es werden das Ein- und Ausschalten der Funktionen sowie das Tasten der Funktionen unterstützt.

Es existiert eine reine DCC-Software von Uwe Mäusezahl. Diese besitzt eine größere Dynamik beim Behandeln der Zustandsänderungen. Mit dieser Software werden deutlich mehr als 16 Lokadressen zyklisch gepollt. Dieser Code unterstützt auch POM. Dieser Code sollte vorzugsweise verwendet werden, wenn auf der Modellbahn MM und MM2 nicht benötigt werden. Er eignet sich insbesondere für große Modellbahnen.

Von einem Lokdecoder-USC werden zyklisch Zustandsänderungen für 16 Lokadressen an den Booster gesendet. Sind auf einer großen Anlage mehr als 16 Lokomotiven in Zugfahrten unterwegs, kann es dazu kommen, dass eine Lok bereits wieder aus dem Speicher gelöscht wurde, bevor der Befehl abgearbeitet wurde. In diesem Fall kann der Wert bis 32 erhöht werden (nur im Quellcode über Patrick Urban möglich). Dadurch verlängert sich aber die Bearbeitungszeit der Lokbefehle. Es muss also ein Maß zwischen Bearbeitungszeit und Loks im Speicher gefunden werden.

Der USC-Kurzschlusserkennung ist für beide Ausgänge separat über das MoBaSbS-Konfigurationstool oder für den primären, unteren Ausgang über den HDC konfigurierbar.

Das Datensignal am Ausgang des µControllers wird über einen RS232-Treiber in ein symmetrisches Signal umgesetzt, das der Booster als Eingangssignal versteht. Über ein Relais kann das Signal zum Booster abgeschaltet werden. Setzt man den EDITS-Booster ein, schaltet das Schienenausgangssignal ab, wenn er kein Eingangssignal erhält. So kann z.B. bei einem aufgetretenen Kurzschluss die Anlage

stromlos geschaltet werden. Die Information über einen detektierten Kurzschluss wird an alle Komponenten des Systems weitergereicht; somit auch an den PC.

Der USC verfügt über eine automatische Abschaltung der Boosterausgänge für den Fall, dass er nicht mehr vom BMC zyklisch angefragt wird. So wird verhindert, dass bei einem Ausfall des BMCs dann unkontrolliert Daten generiert werden und somit Loks ohne Kontrolle fahren.

Kurzschlusserkennung:

Wird vom Booster die Leitung 'Short' auf Low-Pegel gehalten, liegt kein Kurzschluss vor. Bei einem Kurzschluss wird die Leitung 'Short' nicht mehr aktiv vom Booster auf dem Low-Pegel gehalten und ein Pull-Up-Widerstand (markiert als Punkte 'F') auf der Platine zieht die Leitung gegen +5V.

Im MoBaSbS-Konfigurationstool kann die Kurzschlusszeit eingestellt werden. Verfügt der angeschlossene Booster über kein Kurzschlusssignal, so kann die Kurzschlusserkennung deaktiviert werden.

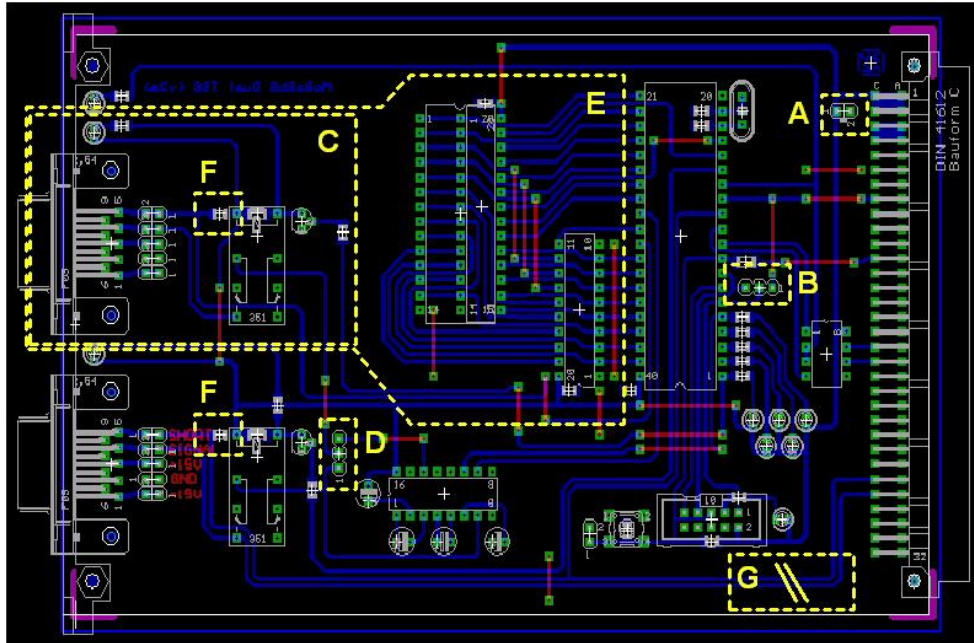
2ter Boosteranschluß:

Der 2te Boosteranschluß (im Layout als C bezeichnet) kann prinzipiell auch benutzt werden. Beide Booster werden dann über dasselbe Steuersignal angesprochen. Dafür muss ein Jumper auf die beiden unteren Pins des Pfostensteckers 'D' gesetzt werden.

Aufbauhinweise:

- Bei industriell gefertigten Platinen ist zu beachten, dass die Bauteilbezeichnungen teilweise falsch sind. Als Referenz ist deshalb immer die Bauteilbezeichnung der Platinen-Layouts aus dem geschützten MoBaSbS-Bereich zu verwenden.
- Die 5 LEDs rechts unten auf der Platine dienen lediglich Debugzwecken und brauchen nicht bestückt zu werden.
- Die LED oben links ist als gelbe LED zu bestücken. Sie zeigt die korrekte Datenkommunikation zum BMC an. Die beiden LEDs oberhalb der SUB-D Buchsen sollten als grüne LED bestückt werden und zeigen den aktiven Zustand des Relais an. Der Basiswiderstand der Schalttransistoren für die Relais ist im Schaltplan mit 47k angegeben. Bei dem verwendeten Relais (Reichelt: Fin 30.22.9 6V) sollte dieser Widerstand etwas kleiner (27k oder 10k) gewählt werden, damit die Relais sauber schalten.
- Das RAM (IC7 oder IC8) und das danebenliegende IC (IC6, Adresslatch 74HC573 brauchen ab Version 4 nicht mehr bestückt zu werden.
- Die 5x2pol Stiftheimbleiten für die SUB-D-Buchsen sollten als ein 10pol Wannenstecker - so wie für die ISP Schnittstelle - bestückt werden. Sie dienen dazu, eine (interne) Verbindung zum (EDITS-)Booster über ein 10pol Flachkabel vornehmen zu können, wenn Booster und MoBaSbS in einem Gehäuse untergebracht werden. So kann die Verbindung zum Booster entweder mit Wannenstecker oder SUB-D-Buchsen ausgeführt werden.
- Der Anschluss des USC findet über die Backplane statt, die alle Module untereinander verbindet.
- Die Platine ist für den Einbau in ein 19"-Gehäuse vorbereitet und kann mit einer Frontplatte versehen werden.

Einstellmöglichkeiten



USC, Platinen-Layout

Die gelben Markierungen im Layout zeigen folgendes an:

- A** Hier können +5V eingespeist werden, z. B. um ein Board außerhalb des Gehäuses zu programmieren.
- B** Optionaler Steckplatz für zukünftige Erweiterungen. Bitte keinen Jumper aufstecken.
- C** Zweiter Boosterausgang. Die Komponenten in Kasten C brauchen nicht bestückt zu werden, wenn kein 2ter Booster angeschlossen werden soll.
- D** Über diesen 3poligen Steckplatz wird ausgewählt, ob der 2te Boosterausgang mit demselben Signal wie der 1ste angesteuert werden soll. In diesem Fall ist ein Jumper auf die beiden unteren Pins zu stecken. Wird der Jumper auf die oberen beiden Pins gesteckt, so wird ein alternatives Boostersignal verwendet (z.B. ein Programmiersignal für Lokdecoder). Diese Option wird derzeit nicht unterstützt.
- E** Wird die Platine als reiner Schaltdecoder eingesetzt, so können alle Komponenten entfallen, die innerhalb der gestrichelten Linie liegen. Ab Version 4 entfallen bei Einsatz als LSC das Latch (IC6) und das RAM (IC7 oder IC8).
- F** Der hier gezeigte Widerstand zieht das Kurzschlusserkennungs-signal des (EDITS-) Boosters gegen VCC. Hat der verwendete Booster kein Kurzschluss-signal, so kann die Kurzschlusserkennung in dem MoBaSbS-Konfigurationstool deaktiviert werden.
- G** Werden mehrere USCs betrieben, müssen die im Bild eingezeichneten Leitungen getrennt werden.

Software

Die aktuelle SW kann aus dem geschützten MoBaSbS-Bereich heruntergeladen werden.

7.2.1 Anschluss von Nicht-EDITS-Boostern

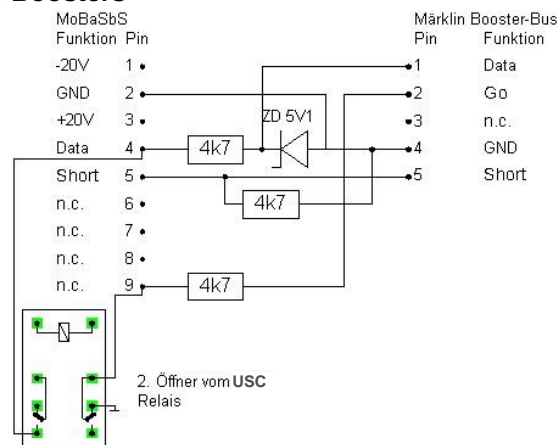
USC-Schnittstelle

Belegung

Pin1	- 18V
Pin2	GND
Pin3	+18V
Pin4	DATA / SIGNAL / TxD
Pin5	SHORT / STOP

Die +/-18V haben bei der MoBaSbS keine Bedeutung und enden bereits am Stecker. Sie sind ein historisches Überbleibsel, da die EDITS Schaltung, die den EDITS Booster ansteuerte 2 Hilfsspannungen von +/-18V benötigte. Die bekam sie aus dem Booster. Der jetzt eingesetzte RS232 hat aber sog. Ladungspumpen und kann sich die ~ +/-12V aus den 5V selbst generieren. Deshalb auch die Elkos am RS232.

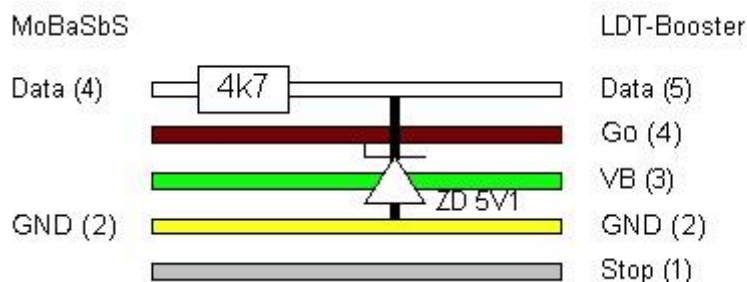
Anschluss eines Märklin-Boosters



Anschluss Märklin-Booster an USC

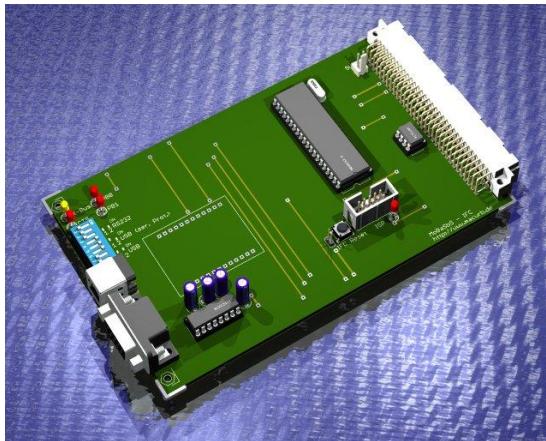
Die Datenleitung des Märklin-Boosters wird nur mit 5V Signal versorgt. Der USC gibt aber ca. 12V Signal ab. Deshalb muss in diese Leitung einen 4,7k Ohm Widerstand und eine Zenerdiode ZD 5V1 gesetzt werden.

Anschluss eines LDT-Boosters

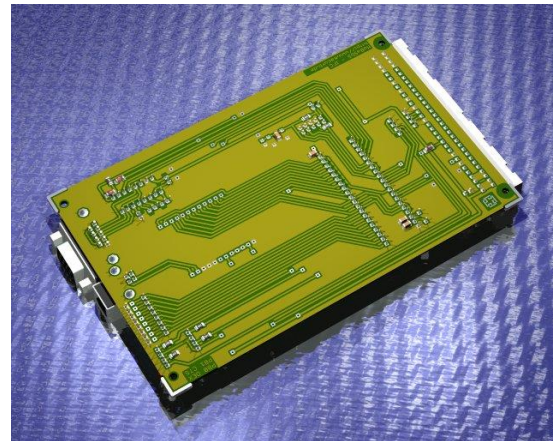


Anschluss LDT-Booster an USC

7.3 IFC



IFC (Bestückungsseite)

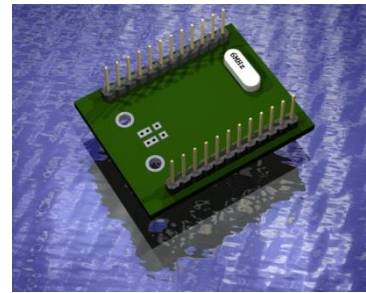


IFC (Lötseite)

Das PC-Interface stellt die Verbindung zwischen der MoBaSbS und einem PC her. Es kann zwischen einer RS232 Schnittstelle (19.200 Baud) mit Hardware-Handshake (CTS) oder -über eine USB-Adapter-Platine- einer USB-Schnittstelle gewählt werden. Zur Nutzung der USB-Schnittstelle muss im PC ein sog. 'Virtueller Com-Port Treiber (VCP)' installiert werden. Er emuliert eine serielle Schnittstelle im PC für die Steuerungssoftware; der Datenverkehr wird aber über die USB-Schnittstelle hergestellt. Den Treiber und weitere Informationen findet man bei FTDI. Es wird kein RS232 nach USB Konverter eingesetzt!



USB-Adapter (Bestückungsseite)

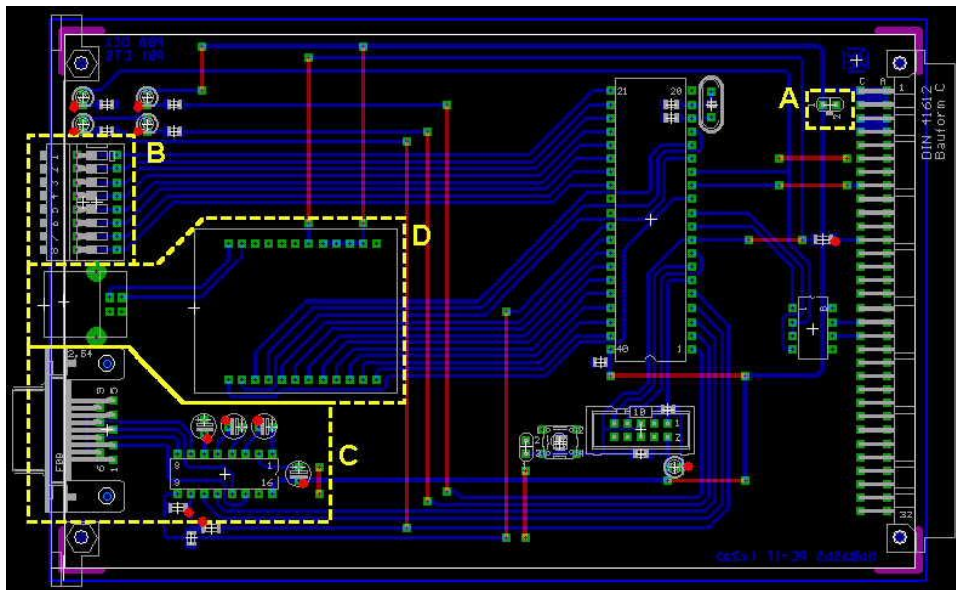


USB-Adapter (Lötseite)

Aufbauhinweise:

- Bei industriell gefertigten Platinen ist zu beachten, dass die Bauteilbezeichnungen teilweise falsch sind. Als Referenz ist deshalb immer die Bauteilbezeichnung der Platinen-Layouts aus dem geschützten MoBaSbS-Bereich zu verwenden.
- Der USB-Adapter wird nicht auf der Hauptplatine verlötet, sondern in zwei Fassungsreihen gesockelt.
- Für den DIP-Switch lassen sich zwei Varianten einbauen. Empfohlen wird die abgewinkelte Variante.
- Die im Layout zu sehenden roten Punkte bezeichnen die Kathode der eingesetzten Dioden bzw. den Minus-Anschluss der Elkos.
- Die beiden LEDs direkt am oberen linken Rand der Platine zeigen den Status der internen Verbindung zur BMC (oberste, gelbe LED) und den Status des CTS Signals (rote LED darunter) an. Die CTS-LED muss während des Betriebes leuchten. Die beiden weiteren LEDs dienen Debugzwecken und brauchen nicht bestückt oder in eine Frontplatte integriert zu werden.
- Der Anschluss des Interfaces findet über die Backplane statt, die alle Module untereinander verbindet.
- Die Platine ist als Einschubkarte für ein 19"-Gehäuse ausgelegt und kann mit einer Frontplatte versehen werden.

Einstellmöglichkeiten



IFC, Platinenlayout

Die gelben Markierungen zeigen folgendes an:

A

Hier können +5V eingespeist werden. Damit besteht die Möglichkeit ein Board außerhalb des Gehäuses zu programmieren.

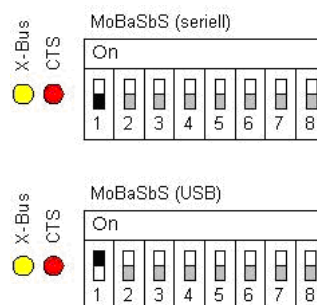
B

8-fach DIP-Switch zur Konfiguration

Der IFC unterstützt mehrere Kommunikationsarten, die sich über den DIP-Switch einstellen lassen.

Mit dem DIP-Switch 1 wird die Betriebsart eingestellt:

- OFF bedeutet serielle Kommunikation. Diese Verbindung entspricht dem Lenz-IF LI101F (in Traincontroller einzustellen). Es wird eine serielle Schnittstelle des PC (RS232) und ein serielles Kabel genutzt.
- ON aktiviert die USB-Kommunikation. Diese Verbindung entspricht dem Lenz IF LI101F (in Traincontroller einzustellen), wobei anstelle der seriellen RS232-Schnittstelle die USB-Schnittstelle mit einem USB-Kabel genutzt wird. Für den Betrieb muss auf dem PC ein VCP-Treiber installiert sein. Im MoBaSbS-Konfigurationstool ist hierzu der Menüpunkt Optionen -> Konfig. ComPort zu beachten (siehe Kap. 12.4.5).



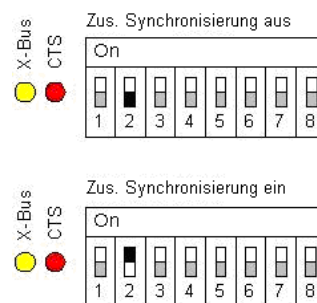
Einstellungen der Betriebsart auf IFC

Mit dem DIP-Switch 2 wird eine zusätzliche Synchronisierung eingestellt:

- OFF bedeutet, dass die zusätzliche Synchronisierung ausgeschaltet ist.
- ON aktiviert die zusätzliche Synchronisierung. Hierbei wird das serielle Protokoll erweitert. Vom PC aus werden immer 0xFF 0xFE dem Befehl vorweg gestellt. Je nach Antwort an den PC sendet das PC-IF dann 0xFF 0xFD (bei Broadcastmeldungen) und 0xFF 0xFE (bei allen anderen Meldungen) vorweg. Hintergrund der zusätzlichen Synchronisierung ist die Anbindung an TC, die bei der LENZ-USB-Unterstützung noch ein paar Extras unterstützt.

Seit Traincontroller V5.8 E1 ist diese Möglichkeit überflüssig, da die MoBaSbS direkt unterstützt wird. Sie kann als Digitalsystem direkt ausgewählt werden.

Es wird deshalb empfohlen, die Option nicht zu nutzen und den DIP-Switch 2 auf OFF zu stellen. Im MoBaSbS-Konfigurationstool ist hierzu der Menüpunkt Optionen -> Erw. ComPort Optionen -> Zus. Synchronisierung ein/aus zu beachten (siehe Kap. 12.4.7).



Einstellungen der zusätzlichen Synchronisierung auf IFC

Weiterhin können Lokfahr- und Funktionsbefehle an den PC weitergereicht werden.

Mit dem DIP-Switch 8 wird die Befehlsweiterleitung eingestellt:

- OFF bedeutet, dass die Weiterleitung der Lokfahr- und Funktionsbefehle ausgeschaltet ist.
- ON aktiviert die Befehlsweiterleitung. In diesem Modus werden Lokfahr- und Funktionsbefehle, die von einem X-Bus Device (z.B. HDC) abgesetzt werden, an den PC durchgereicht. Das MoBaSbS-Konfigurationstool wertet diese Informationen aus und aktualisiert die Zustandsinformationen (Geschwindigkeit, Richtung und Status der Funktionstasten) in den Lok-Kontroll-Fenstern.

Seit Traincontroller V5.8 E1 wird die MoBaSbS direkt unterstützt. Sie kann als Digitalsystem ausgewählt werden. Es wird deshalb empfohlen, ab Nutzung des Traincontrollers V5.8 E1 den DIP-Switch 8 auf ‚ON‘ zu setzen. Im MoBaSbS-Konfigurationstool sind keine Optionen zu beachten.

Die Einstellung wirkt sich erst aus, wenn die MoBaSbS aus- und wieder eingeschaltet wird.



Einstellungen ‚Lokbefehle weiterreichen‘ auf IFC

C

Schnittstelle und Komponenten für den RS232 Betrieb. Das serielle Kabel zum PC ist 1:1 zu verbinden. Nachstehend die Pin-Nummern für 9-pol. und 25-pol. Stecker

Pin Nr.	Bez.	Aufgabe	E/A	Pin Nr. bei 25 P.
3	TxD	Sendedaten	A	2
2	RxD	Empfangsdaten	E	3
5	GND	Erde	-	7
7	RTS	Sendeteil einschalten	A	4
4	DTR	Endgerät bereit	A	20
8	CTS	Sendebereitschaft	E	5
6	DSR	Betriebsbereitschaft	E	6
1	DCD	Empfangssignalpegel	E	8
9	RI	Ankommender Ruf	E	22

D

Schnittstelle und USB-Adapter für den USB-Betrieb

Hinweise zum VCP (Virtuellen Com-Port) Treiber von FTDI:

Der auf dem USB-Adapter eingesetzte USB Chip hat die Bezeichnung FT 245 BM. Ebenso einsetzbar ist der FT 245 BL.

Für die Programmierung hat Holger Bernhard eine ausführliche Anleitung geschrieben (siehe Kap. 13.2.1). Alternativ können die Schritte auch wie folgt durchgeführt werden:

Aktuelle Treiber für den VCP findet man bei FTDI. Es ist möglich, dass nicht jede Treiberversion korrekt funktioniert. Bestätigt ist z. B. CDM 2.04.06 WHQL, 2.08.23 Beta und CDM 2.10.00 WHQL Certified (auch für WIN10). Die Version 2.12.00 scheint unter WIN10 nicht zu funktionieren.

Nachdem man das ZIP-Archiv von FTDI geladen hat, wird es in ein beliebiges Verzeichnis entpackt. FTDI hat der MoBaSbS eine eigene PID innerhalb der für FTDI vergebenen VID zugeteilt (VID: Vendor ID, PID: Product ID). Die VID lautet 0x0403, die zugewiesene PID 0xCB20. Diese beiden Werte, sowie die 'Product Description 'USB-IF MoBaSbS V4.0', müssen in zwei Dateien ftdibus.inf und ftdiport.inf, die in dem ZIP-Archiv enthalten sind, eingetragen werden.

Nachstehend die Einzelheiten die modifiziert werden müssen (dargestellt sind lediglich die Passagen in den beiden Dateien, in denen die rot eingefärbten Zeilen ergänzt werden müssen):

Ftdibus.inf aus VCP-Treiberpaket, Version 2.06.02 vom 30.03.2010

```
[FtdiHw]
%USB\VID_0403&PID_6001.DeviceDesc%=FtdiBus.NT,USB\VID_0403&PID_6001
%USB\VID_0403&PID_6010&MI_00.DeviceDesc%=FtdiBus.NT,USB\VID_0403&PID_6010&MI_00
%USB\VID_0403&PID_6010&MI_01.DeviceDesc%=FtdiBus.NT,USB\VID_0403&PID_6010&MI_01
%USB\VID_0403&PID_6011&MI_00.DeviceDesc%=FtdiBus.NT,USB\VID_0403&PID_6011&MI_00
%USB\VID_0403&PID_6011&MI_01.DeviceDesc%=FtdiBus.NT,USB\VID_0403&PID_6011&MI_01
%USB\VID_0403&PID_6011&MI_02.DeviceDesc%=FtdiBus.NT,USB\VID_0403&PID_6011&MI_02
%USB\VID_0403&PID_6011&MI_03.DeviceDesc%=FtdiBus.NT,USB\VID_0403&PID_6011&MI_03
%USB\VID_0403&PID_CB20.DeviceDesc%=FtdiBus.NT,USB\VID_0403&PID_CB20

[FtdiHw.NTamd64]
%USB\VID_0403&PID_6001.DeviceDesc%=FtdiBus.NTamd64,USB\VID_0403&PID_6001
%USB\VID_0403&PID_6010&MI_00.DeviceDesc%=FtdiBus.NTamd64,USB\VID_0403&PID_6010&MI_00
%USB\VID_0403&PID_6010&MI_01.DeviceDesc%=FtdiBus.NTamd64,USB\VID_0403&PID_6010&MI_01
%USB\VID_0403&PID_6011&MI_00.DeviceDesc%=FtdiBus.NTamd64,USB\VID_0403&PID_6011&MI_00
%USB\VID_0403&PID_6011&MI_01.DeviceDesc%=FtdiBus.NTamd64,USB\VID_0403&PID_6011&MI_01
%USB\VID_0403&PID_6011&MI_02.DeviceDesc%=FtdiBus.NTamd64,USB\VID_0403&PID_6011&MI_02
%USB\VID_0403&PID_6011&MI_03.DeviceDesc%=FtdiBus.NTamd64,USB\VID_0403&PID_6011&MI_03
%USB\VID_0403&PID_CB20.DeviceDesc%=FtdiBus.NTamd64,USB\VID_0403&PID_CB20

[Strings]
Ftdi="FTDI"
DESC="CDM
Driver Package"
DriversDisk="FTDI
USB Drivers Disk"
USB\VID_0403&PID_6001.DeviceDesc="USB Serial Converter"
USB\VID_0403&PID_6010&MI_00.DeviceDesc="USB Serial Converter A"
USB\VID_0403&PID_6010&MI_01.DeviceDesc="USB Serial Converter B"
USB\VID_0403&PID_6011&MI_00.DeviceDesc="USB Serial Converter A"
USB\VID_0403&PID_6011&MI_01.DeviceDesc="USB Serial Converter B"
USB\VID_0403&PID_6011&MI_02.DeviceDesc="USB Serial Converter C"
USB\VID_0403&PID_6011&MI_03.DeviceDesc="USB Serial Converter D"
USB\VID_0403&PID_CB20.DeviceDesc="USB-IF MoBaSbS V2.0"
SvcDesc="USB Serial Converter Driver"
ClassName="USB"
```

Ftdiport.inf aus VCP-Treiberpaket, Version 2.06.02 vom 30.03.2010

```
[FtdiHw]
%VID_0403&PID_6001.DeviceDesc%=FtdiPort232.NT,FTDIBUS\COMPORT&VID_0403&PID_6001
%VID_0403&PID_6010.DeviceDesc%=FtdiPort2232.NT,FTDIBUS\COMPORT&VID_0403&PID_6010
%VID_0403&PID_6011.DeviceDesc%=FtdiPort2232.NT,FTDIBUS\COMPORT&VID_0403&PID_6011
%VID_0403&PID_CB20.DeviceDesc%=FtdiPort232.NT,FTDIBUS\COMPORT&VID_0403&PID_CB20

[FtdiHw.NTamd64]
%VID_0403&PID_6001.DeviceDesc%=FtdiPort232.NTamd64,FTDIBUS\COMPORT&VID_0403&PID_6001
%VID_0403&PID_6010.DeviceDesc%=FtdiPort2232.NTamd64,FTDIBUS\COMPORT&VID_0403&PID_6010
%VID_0403&PID_6011.DeviceDesc%=FtdiPort2232.NTamd64,FTDIBUS\COMPORT&VID_0403&PID_6011
%VID_0403&PID_CB20.DeviceDesc%=FtdiPort232.NTamd64,FTDIBUS\COMPORT&VID_0403&PID_CB20

[Strings]
FTDI="FTDI"
DESC="CDM Driver Package"
DriversDisk="FTDI USB Drivers Disk"
PortsClassName = "Ports (COM & LPT)"
VID_0403&PID_6001.DeviceDesc="USB Serial Port"
VID_0403&PID_6010.DeviceDesc="USB Serial Port"
VID_0403&PID_6011.DeviceDesc="USB Serial Port"
VID_0403&PID_CB20.DeviceDesc="USB-IF MoBaSbS V2.0"
SvcDesc="USB Serial Port Driver"
SerEnum.SvcDesc="Serenum Filter Driver"
```


Die geänderten INF-Dateien sollten unbedingt gesichert werden, da diese bei einer erneuten Installation der MobaSbs benötigt werden! Sonst wird der USB-Treiber für die MobaSbs installiert!

Konfiguration des USB-Adapters:

Damit die oben beschriebenen Einstellungen auch zum USB-Adapter passen, muss dieser entsprechend konfiguriert werden.

Hierzu gibt es ausführliche Beschreibungen von Roman Kircher und Jörg Schindler für die Installation unter WIN7/8/10. (siehe Kap. 13.2.2 bis 13.2.4). Weiterhin können folgende Hinweise hilfreich sein:

Ist auf dem PC bereits ein VCP-Treiber installiert, muss dieser zuerst wieder deinstalliert werden. Im Archiv der FTDI Treiber gibt hierfür ein Deinstallationsprogramm mit Namen FTDIUNIN.exe.

Im ersten Schritt wird die MoBaSbS (mit USB-Adapter auf IFC) mit dem PC verbunden. Eine neue Hardware wird erkannt. Der Adapter erscheint als Eintrag unter den USB-Devices im Geräte-Manager der Systemsteuerung.

Nun den CDM-Treiber von FTDI, so wie oben beschrieben, installieren. Hat man alles richtig gemacht, dann sollte jetzt der USB-Adapter im Geräte-Manager des Betriebssystems auftauchen.

Wurden bereits einige Installationen durchgeführt, müssen nicht genutzte COM-Ports wieder gelöscht werden, da das Konfigurationstool nur COM-Ports bis COM7 unterstützt (siehe Kap. 12.4.5).

Unter Win XP: Start > Einstellungen > Systemsteuerung > System, dann Register ‚Hardware‘ > Geräte-Manager > Anschlüsse(COM&LPT), dann > Ansicht > Ausgeblendete Geräte anzeigen (‚Häkchen‘ setzen). Jetzt werden die nicht mehr belegten COM-Ports mit angezeigt. Diese dann löschen.

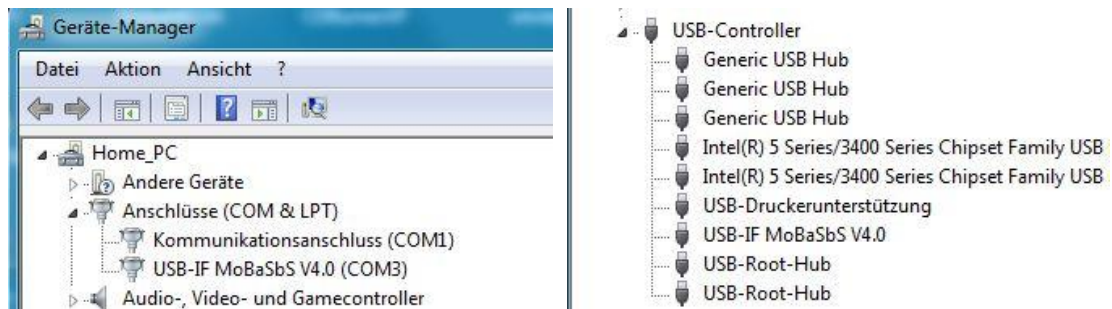
Unter WIN7 (mit Admin-Rechten): Start > Systemsteuerung > System und Sicherheit > System > Geräte-Manager > Anschlüsse (COM&LPT) > Anschlusseinstellungen (2. Reiter) und dann ‚erweitert‘. Danach eine Schnittstelle mit tieferer Anschlussnummer wählen. Nach jeder Deinstallation eines COM-Ports muss evtl. das System neu gebootet werden, da die Anzeige ‚Anschlüsse‘ im Geräte-Manager verschwindet.

Das FTDI clean tool (im Internet auf den FTDI-Seiten zu finden) löscht zwar die FTDI-Treiber, setzt aber die COM-Ports nicht zurück.

Ist alles richtig installiert, müssen bei eingeschalteter MoBaSbS unter ‚Anschlüsse (COM & LPT)‘ und unter ‚USB-Controller‘ die MoBaSbS-Treiber aufgeführt sein. Beim USB-Controller handelt es sich um den FTDI-Bustreiber und bei den Anschlüssen um den FTDI-Porttreiber. Beim Neuinstallieren der Treiber muss man dem Geräte-Manager manuell das Verzeichnis angeben, wo er die Treiber finden kann.

Tritt bei der Installation ein Fehler auf, liegt dies an der Treiberversion

So sieht es im Gerätemanager aus, wenn die MoBaSbS eingeschaltet und angeschlossen ist:



Nachfolgend wird ein Konfigtool gestartet (mprog), welches man ebenfalls bei FTDI findet. In 'mprog' wird die Defaultdatei 'ftdi.ept' geladen. Über den entsprechenden Knopf in 'mprog' (der Knopf hat das Aussehen eines 'blauen Blitzes'), wird nun der Adapter, d.h. der USB-Chip und das EEPROM konfiguriert. Damit sind alle notwendigen Einstellungen vorgenommen.

Wird der PC jetzt mit der MoBaSbS verbunden, erscheint sie mit Namen und korrekter Kennung im System.

Verwendet man das Tool MProg 3 ist folgendes zu beachten:

MProg 3 arbeitet evtl. nicht mit den Treibern, welche zu Zeiten vom MProg2 verwendet wurden und nicht unter WIN98/ME. Vor einem Treiber-Update müssen in den neuen INF-Dateien die Einträge, welche in den alten INF-Dateien manuell eingetragen wurden, bereits vorhanden sein! Andernfalls werden die neuen Treiber nicht installiert.

Software

Aufgrund von gelegentlich auftretenden Rückmeldehängern wurde der Code im IFC geändert. Die aktuelle Version ist 130624_IFC_EncFix. Sie kann von dem geschützten MoBaSbS-Bereich heruntergeladen werden.

Für die technisch Interessierten: Im IFC Code werden in einer Speicherstruktur, die als Ring ausgeführt ist, die Kommandos von der MoBaSbS zum PC zwischengespeichert. Diese Ringstruktur wurde vergrößert, so dass mehr Kommandos hineinpassen. Es besteht die Vermutung, dass bei hohem (Rückmelde-)Aufkommen noch nicht an den PC gesendete Daten überschrieben wurden.

7.4 Netzteil

Das Netzteil versorgt die zentralen Module - BMC, IFC, PMC, USC mit der Versorgungsspannung von +5V. Abhängig von der Gesamtkonfiguration der Modellbahnanlage wird noch ein +12V Netzteil benötigt.

Der von den Netzteilen zu liefernde Strom sollte mind. 2A betragen. Es kann auch jedes Netzteil verwendet werden, das höhere Ströme liefert. Dazu gehören z. B. alte PC-Netzteile.

Alle Netzteile sollten über eine genügende Spannungs-Steilheit verfügen, damit die Elektronik im Moment des Einschaltens nicht instabil wird. Dies ist oftmals die Ursache für unerklärliches Verhalten bei Einschalten.

Aufbauhinweise:

Zum einfachen Einbau wird empfohlen, das Netzteil auf eine 160x100mm Epoxyd-Platine aufzuschrauben. So lässt sich das Netzteil wie eine Einschubkarte verwenden.

Es wird ausdrücklich darauf hingewiesen, dass die Netzteile mit 230V versorgt werden. Diese Netzspannung ist lebensgefährlich! Es darf unter keinen Umständen eine Verbindung der Netzspannung mit den Schienen entstehen!

Es ist deshalb unbedingt auf Berührungsschutz, korrekten Anschluss des Schutzleiters und Absicherung der Primärseite zu achten. Die Abschaltung sollte über einen 2poligen Netzschalter erfolgen.

Die Gesamtstromaufnahme der MoBaSbS liegt bei 1,0 bis 1,5A. Der Wert erhöht sich, wenn Fahrpulte angeschlossen werden. Das LC-Display ist hierbei der größte Verbraucher. Bis zu vier Fahrpulte sollten kein Problem darstellen. Werden die Fahrpulte mit langen (>7m) Kabeln an der MoBaSbS angeschlossen, so kann bei einer 5V Versorgung der Spannungsabfall über die Leitung dazu führen, dass die Spannung am Fahrpult nicht mehr ausreichend ist.

Werden derart lange Kabel benötigt, bestehen zwei Möglichkeiten:

Ringleitung

Eine Ringleitung am Rand des Anlagenrahmens mit ausreichend dimensioniertem Kabelquerschnitt. In diese Ringleitung wird neben dem differentiellen Signal des BMC auch die Versorgungsspannung eingespeist. An verschiedenen Stellen in dieser Ringleitung können jetzt RJ45-Dosen eingeschleift werden, an denen dann - mit kurzen Kabeln - die HDCs angeschlossen werden können.

Auswählen der > 5V Option:

Der HDC hat einen integrierten 5V Festspannungsregler der über eine Jumperoption geschaltet werden kann. Wird diese Option im Fahrpult benutzt, so kann man über die Backplane eine Spannung zwischen 9V und 12V einspeisen. Auf dem BMC lässt sich durch Jumper bestimmen, ob ein angeschlossenes Fahrpult direkt mit +5V oder der höheren Spannung versorgt werden soll.

7.5 HDC

7.5.1 5V-Display-Variante



HDC

Der HDC dient einer Vielzahl von Aufgaben. Dazu gehören:

- Anzeige der Klartextnamen einer Lok/eines Zuges
- Anzeige der Geschwindigkeit, Richtung, Protokolltyp und Fahrstufenanzahl
- Anwahl der Geschwindigkeit und Richtung
- Anwahl der Lokfunktionen
- Eingabegerät zur Konfiguration eines Lokdatensatzes
- Eingabegerät zur Konfiguration von Basismodulen
- Eingabegerät zur Konfiguration von Servodecoder-Peripheriemodulen (PM)
- Eingabegerät zur Konfiguration von PM Adressen
- Steuergerät für Schaltausgänge von Decodern
- Anzeigegerät für Encoderstatus

Angeschlossen wird der HDC an der BMC über ein 8pol ISDN- oder Patchkabel. Die RJ45 Buchse am Fahrpult befindet sich unten rechts. An der oberen Stirnseite des Fahrpults befindet sich die ISP-Schnittstelle, um den µC auch ohne Öffnen des Gehäuses programmieren zu können.

Das HDC ist Hot-Pug-In-fähig. Es kann im laufenden Betrieb abgezogen und wieder eingesteckt werden.

Nach dem Programmieren des HDC ist die Default-Adresse 31 eingestellt. Diese muss mit dem MoBaSbS-Konfigurationstool oder am HDC selbst beim ersten Einsatz geändert werden.

Display

Bei dem Display handelt es sich um ein LCD-Modul mit 4x20 Zeichen. Die beiden oberen Zeilen enthalten die Menüführung. In der 3. Zeile wird im Betriebsmodus (dem Modus, in dem die Lokomotiven gesteuert werden) der Lokname angezeigt. Die Loknamen werden mit Hilfe des MoBaSbS-Konfigurationstools erstellt.

In der 4. Zeile stehen Informationen, auf welchem VirtuellenFahrpult (VFP) die Lok gerade aufgerufen ist, ob sie gerade aktiv gesteuert wird (oder von einem anderen Handregler oder dem PC übernommen worden ist), in welche Richtung sie fährt und die aktuelle Fahrstufe.

Tasten und LEDs

Stopp

Oben rechts bei der einzelnen roten LED befindet sich ein Notstop-Taster. Mit ihm wird die Spannung am Gleis abgeschaltet. Diese Meldung geht an alle angeschlossenen Komponenten - auch an die eventuell steuernde PC-Software.

Virtuelles Fahrpult (VFP)

Die 5 Taster in der 2ten Reihe sind VFP-Taster. Durch Drücken einer der VFP-Tasten, wird die dieser Taste momentan zugeordnete Lokadresse aufgerufen. Man hat damit also einen sehr schnellen Zugriff auf 5 Lokadressen. Möchte man eine Lokadresse aufrufen, die keiner VFP-Taste zugeordnet ist, so hält man die VFP-Taste gedrückt und wählt über den (Geschwindigkeits-)Drehregler die neue Lokadresse aus. Diese bleibt nun der entsprechenden VFP-Taste zugeordnet, bis entweder eine neue Adresse ausgewählt wird oder die Spannung des HDCs abgeschaltet wird. Alternativ kann man über einen Menüpunkt die aktuelle Zuordnung dauerhaft im Speicher des HDCs festhalten.

Schließt man die maximal möglichen vier Handregler an den Datengenerator an, so kann man über die Technik der jeweils 5 VFP-Tasten 20 Lokadressen direkt kontrollieren.

F0 bis F4

Die schwarzen Taster mit den roten LEDs bzw. der einen grünen dienen zum Aktivieren und Anzeigen der Lokfunktionen (Sonderfunktion, F1 bis F4).

Fahrtrichtung

Die Fahrtrichtung wird entweder durch Drücken des Drehreglers oder durch die jeweilige Lokrichtungstaste (6te Spalte, 3te und 4te Zeile '<--' & '-->') umgeschaltet (natürlich nur, wenn die Geschwindigkeit auf Null steht!).

F5 bis F12

Die Tasten können mit dem MoBaSbS-Konfigurationstool mit weiteren Lokadressen belegt werden. Sie sind fest zugeordnet und können nicht wie die VFP-Tasten im laufenden Betrieb geändert werden.

Drehregler

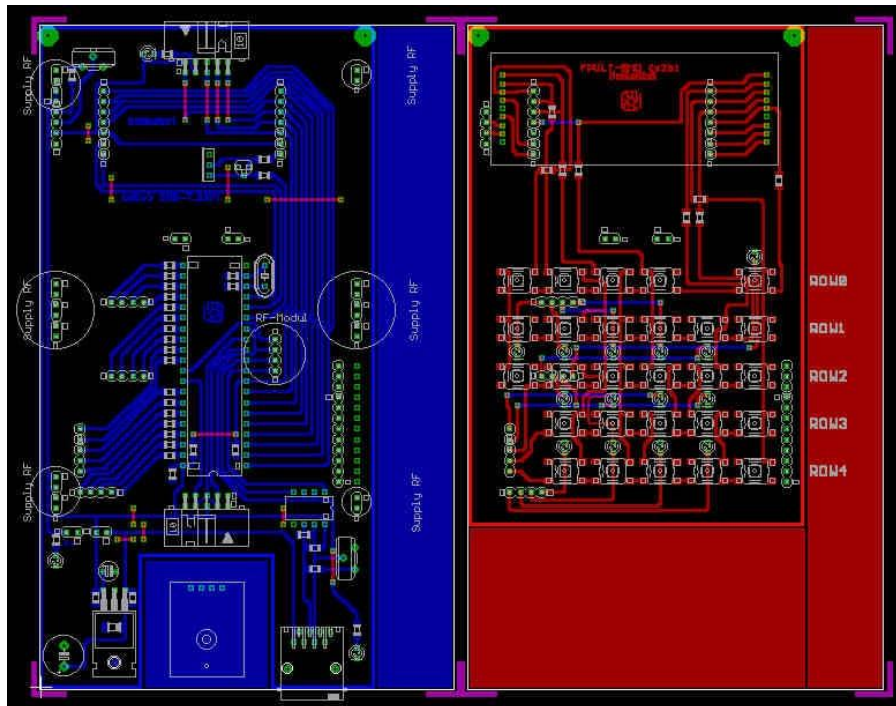
Der Drehregler ist ein opto-elektronischer Inkrementalgeber. Es gibt keine Nullstellung. Daher kann auch beliebig zwischen Lokadressen gewechselt werden, ohne auf die zuvor eingestellte Geschwindigkeit achten zu müssen. Wie oben erwähnt, dient der Drehknopf auch zur Anwahl einer neuen Lokadresse, die dann der jeweiligen gedrückten VFP-Taste zugeordnet wird.

Lok, Peri, Menu, Esc

Die Funktion der Tasten sind im Anhang Kap. 13.1 erklärt.

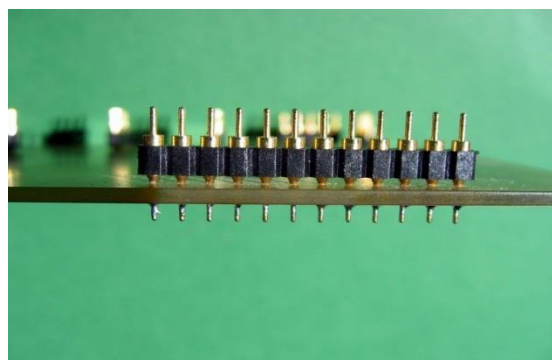
Aufbauhinweise:

- Bei industriell gefertigten Platinen ist zu beachten, dass die Bauteilbezeichnungen teilweise falsch sind. Als Referenz ist deshalb immer die Bauteilbezeichnung der Platinen-Layouts aus dem geschützten MoBaSbS-Bereich zu verwenden.
- Der HDC ist in Sandwichtechnik aufgebaut. Basis bildet eine Hauptplatine, auf die ein Bedienteil aufgesteckt wird. Dieses Bedienteil trägt nun seinerseits das LC-Display-Modul.
- Damit ein versehentliches Einstecken des HDCs in einen PMC nicht zu einer Zerstörung des HDCs führt - an den PMC-Ausgängen liegen +9V bis +12V an -, sollte der HDC mit dem 5V Festspannungsregler ausgestattet werden.

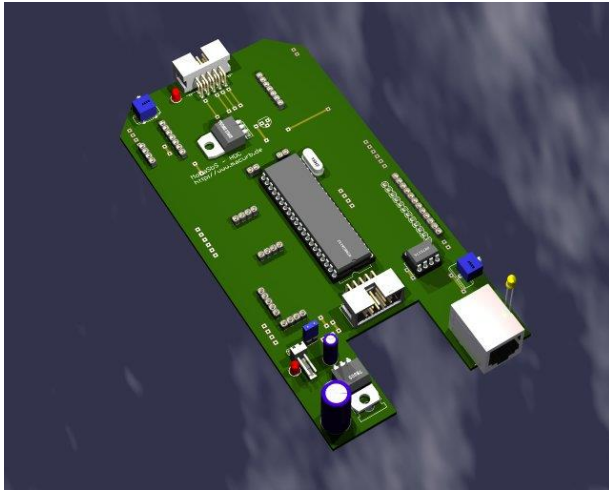


HDC, Platinen-Layout, Variante für Minitaster:

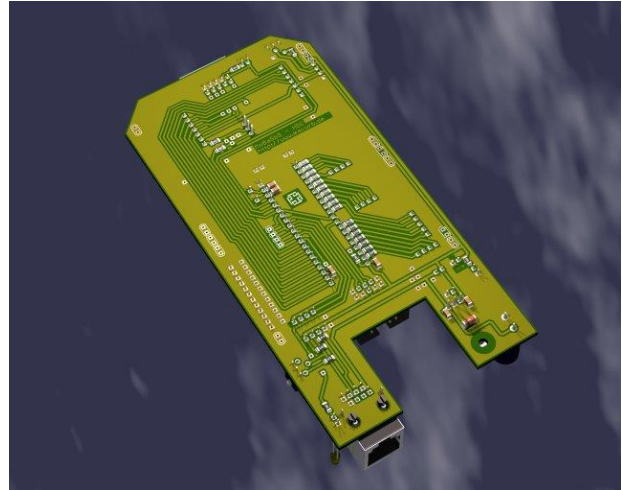
Die folgenden Bilder zeigen die Hauptplatine, das Bedienteil und eine Detailansicht des verwendeten Adapters (Bezug Reichelt: AW122/32). Wichtig ist es, die Kontaktbuchsen und Adapter so auszurichten, dass sie später genau ineinander gesteckt werden können. Wegen des geringen Zwischenraums müssen sehr flache Sockel für die beiden ICs verwendet werden, die direkt (1.1mm Bohrung) in die Platine versenkt werden. Die bisher empfohlenen Carrier-Buchsen von Conrad-Elektronik (Nr. 739049) sind nicht mehr lieferbar. Es kann auch ein superflacher Sockel (Reichelt GS 40P) verwendet werden. Ebenso wird die Buchsenleiste mit RM2.0 von Conrad Nr. 729243 durch die Nr. 741606 ersetzt.



HDC, Adapter (Detailansicht)

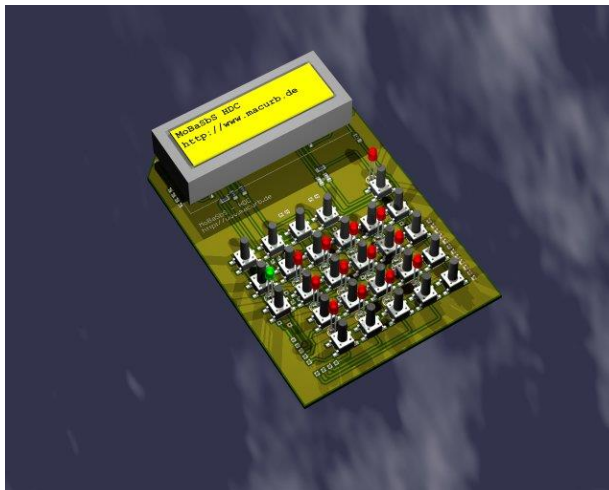


HDC, Basisplatine (Bestückungsseite)

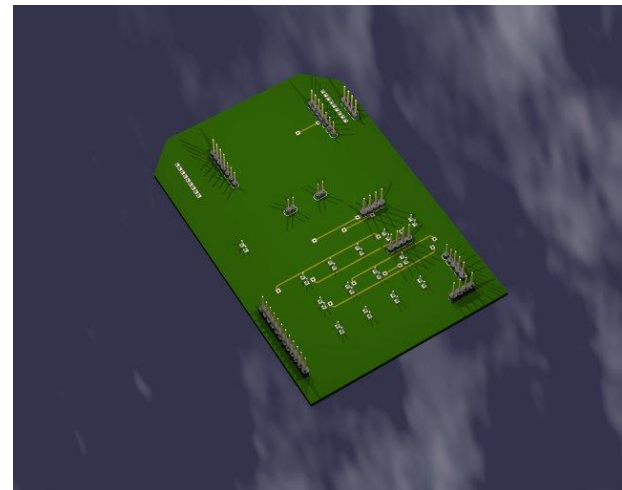


HDC, Basisplatine (Lötseite)

Bei der Basisplatine ist auf exakte Ausführung der Bohrungen zu achten. Sowohl beim Bedienteil als auch beim Motherboard müssen die oberen Ecken weggebohrt bzw. abgesägt werden, da diese sonst die Gehäuseverbindungen stören würden.



HDC, Bedienplatine (Bestückungsseite)



HDC, Bedienplatine (Lötseite)

Auf (oder besser unter) dem Bedienteil müssen einige Drahtbrücken gezogen werden. Diese sind leicht gewinkelt zu verlegen, da sie genau unter den LEDs verlaufen und so zu Kurzschlüssen führen können. Insgesamt sind auf dem Bedienteil 6 Lötbrücken vorzusehen. Bitte kontrollieren!

Der FET SPU xxx ist nicht mehr lieferbar. Als Ersatz-Typ kann der IRLR 024N (wie im Decoder) eingesetzt werden. Die I-Pak-Variante IRLU3410 ist auch möglich. Wird ein IRLU 3410 nicht eingesetzt, so kann dafür ein BC 337 direkt daneben eingelötet werden und der Widerstand R 35 ist von 0 Ohm auf 4k7 Ohm zu erhöhen.

Der Spannungsregler braucht nur dann montiert zu werden, wenn das Fahrpult mit einer höheren Spannung versorgt werden soll (9V - 12V, von der BMC). Dann darf der Jumper nicht gesetzt werden. Bei 5V Betrieb muss er gesetzt werden. Es wird empfohlen, den Spannungsregler zu bestücken und auch zu nutzen. Damit ist ausgeschlossen, dass der HDC Schaden nimmt, wenn es aus Versehen in eine der PMC-Ausgangsbuchsen gesteckt wird. Denn dort liegen zur Versorgung der Peripheriemodule (Encoder und Decoder) 9-12V DC Spannung an!

Bei einer der Adapterleisten müssen die durchstehenden Enden extrem gekürzt werden, da später dort ein Taster montiert werden muss. Zusätzlich muss von der Unterseite des besagten Tasters (3te Reihe, 2te Spalte) mit einem Minifräser etwas Material weggenommen werden; dann lässt sich der Taster wieder sauber aufsetzen, ohne zu wackeln. Bei der Bestückung des Bedienteils sollte man immer Zeilenweise vorgehen, sonst wird das Löten unnötig erschwert.

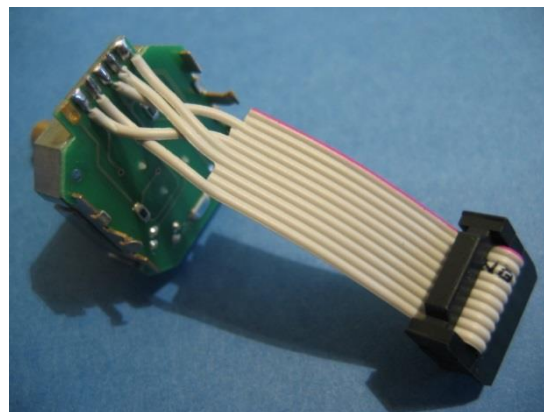
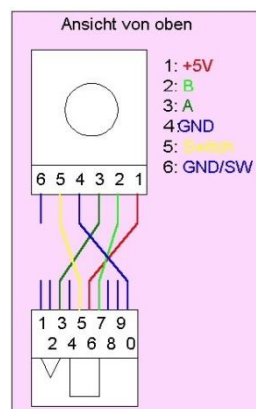
Die LEDs sollten nicht zu weit aus dem Gehäuse heraus schauen. Sie können ca. 1.5 bis 2mm tiefer als der Tasterknopf eingebaut werden. Bei 3mm verschwinden sie fast plan im Gehäuse.

Ein Tipp dazu: Wenn man einen 6mm breiten und 60mm langen Pappstreifen zwischen die Beinchen der LEDs einer Reihe steckt, so können die LEDs in der Höhe beim Verlöten optimal justiert werden. Am Besten ist es, erst nur ein Beinchen jeder LED einer Reihe anzulöten und dann den Pappstreifen wieder herauszuziehen.

Das LC-Display wird direkt auf das Bedienteil gelötet. Da es von oben verlötet werden muss, dürfen die Beinchen nicht vollständig in die Platine gesteckt werden. Optimal ist es, wenn sie unten gerade eben wieder herauskommen.

Die Tasten müssen sehr genau beim Festlöten positioniert werden. Ebenso sind deren Bohrungen und die der LEDs exakt auszuführen. Dazu kann man das Layout ausdrucken und von innen in das Gehäuse kleben. Danach bohrt man mit einem 1mm-Bohrer genau die angezeigten 'Fadenkreuze' vor. Von der Oberseite wurden die LED Bohrungen mit 3mm und die Tastenbohrungen mit 3.5mm erweitert (Bohrmaschine im Bohrständler langsam laufen lassen!). 3.5mm ist schon recht knapp und die Bestückung muss schon recht genau sein. Falls Tasten klemmen, kann man die Bohrungen von der Innenseite ansenken. Generell gilt für die Arbeit an den Aussparungen: Lieber ein paar Mal mehr zusammenbauen, als sich nachher über zu große Ausschnitte ärgern. Die unteren Befestigungsnocken im Gehäuse müssen etwas gekürzt werden, dann steht einem endgültigen Einbau der Platinen nichts mehr im Wege.

Zum Schluss fehlt noch die Bohrung für den Inkrementalgeber. Er wird über ein Stück Flachbandkabel angeschlossen. Das Bild zeigt den Anschluss des Drehgebers an den 10poligen Wannenstecker.



Inkrementalgeber, Schaltplan

Inkrementalgeber, Anschluss

Gelegentlich fehlt die interne Verbindung von GND zu GND/SW. Dadurch funktioniert der Inkrementalgeber unregelmäßig. Es ist dann eine Verbindung von PIN4 (GND) mit PIN6 (GND/SW) herzustellen.

Für die Verbindung vom HDC zur MoBaSbS kann neben dem 8-poligen Flachbandkabel auch ein Spiralkabel für Telefonhörer verwendet werden. Hierbei ist der RJ12 abzuschneiden und durch einen RJ45-Stecker zu ersetzen.

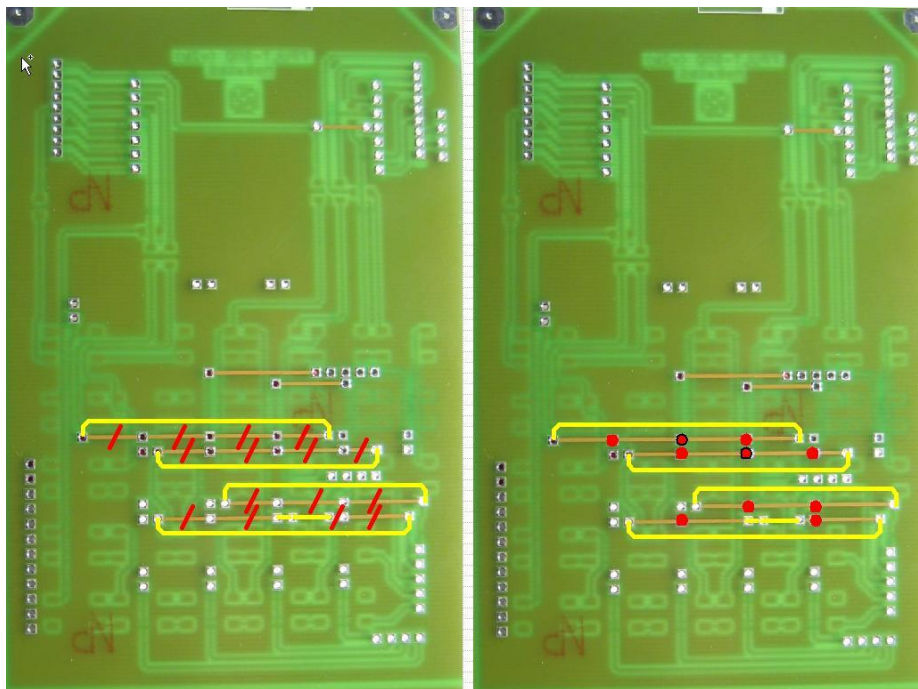
Weitere Hinweise zum Aufbau:Korrektur aus Sammelbestellung Anfang 2005

Es ist eine Korrektur auf den Platinen, die in einer Sammelbestellung Anfang 2005 in Industrie-Qualität gefertigt wurden, durchzuführen:

Der Fehler liegt in einem Kurzschluss verschiedener Leitungen, hervorgerufen dadurch, dass die Platinen wider erwartend doch doppelseitig gefertigt wurden. Die Drahtbrücken auf dem Bedienteil hätten dabei nicht einfach nur gerade, sondern so, wie in untenstehender Abbildung als gelbe Linien zu sehen, verlegt werden müssen.

Korrektur:

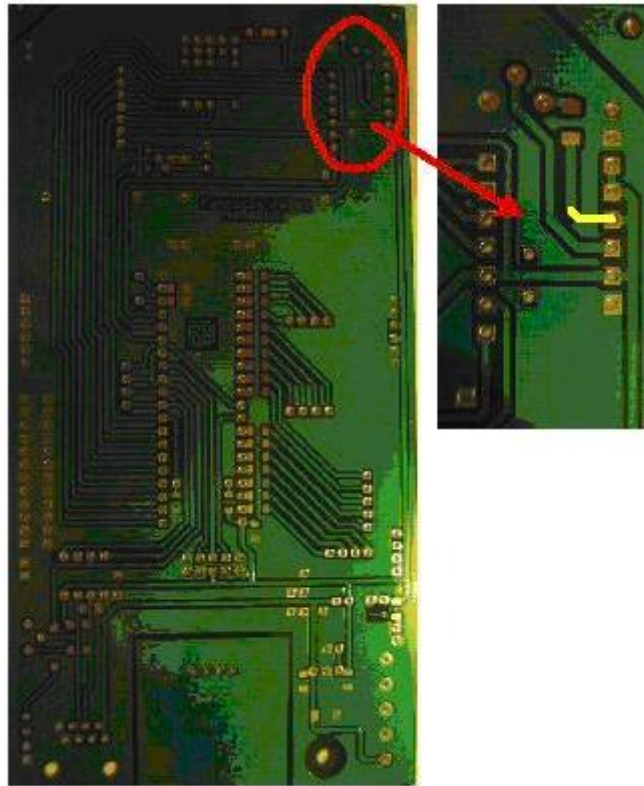
Die in Orange-Farbe zu sehenden gefertigten Brücken müssen an den durch die roten Striche gekennzeichneten Stellen aufgetrennt werden und stattdessen müssen die gelb eingezeichneten Brücken gelegt werden. Alternativ kann man auch die Durchkontaktierungen mit einem z.B. 2.5mm Bohrer an den durch die roten Punkte gekennzeichneten Stellen ansenken (nicht durchbohren!) Dabei kann man sogar auf die Punkte mit schwarzem Rand verzichten, denn diese werden durch die anderen Trennungen quasi freigelegt.



HDC: Korrekturen der Leiterbahnen (Sammelbestellung 2005)

Korrektur aus Sammelbestellung 2006/2007

Hier fehlt in der oberen rechten Ecke (siehe Abbildung) der HDC Hauptplatine eine Leiterbahnverbindung (in der Abbildung in gelb eingezeichnet). Kratzt man die grüne Lackschicht etwas auf der Leiterbahn weg, kann man diese Verbindung mit einem Stückchen Draht schnell wieder herstellen).



HDC-Korrektur 2006/2007

Software

Die aktuelle SW kann aus dem geschützten MoBaSbS-Bereich heruntergeladen werden.

7.5.2 3,3V-Display-Variante

Hierbei handelt es sich um eine Weiterentwicklung von Frank Kaiser des von Patrick Urban konstruierten und durch Uwe Mäusezahl um Funktionen ergänzten HDC. Die Überarbeitung verfolgte folgende Ziele:

- Umbau auf 3,3 V Technik, um wieder marktverfügbare LCD-Displays nutzen zu können
- Vereinfachung des Aufbaus
- Verwendung eines anderen Mikrocontrollers, um mehr Programmspeicherplatz verfügbar zu haben
- Integration von 28 Funktionen für das DCC Gleisformat
- Unterstützung der Tastfunktion der Funktionstasten
- Programming on Main (POM)
- Möglichkeiten zum Einbau eines kostengünstigen mechanischen marktverfügbaren Drehgebers
- Raum für weitere Erweiterungen

Die 3,3V Variante unterstützt alle Funktionen der 5V Variante. Zusätzlich stehen die Funktionen F13-F28 im DCC Protokoll zur Verfügung. Alle Funktionen können jetzt schaltend und tastend verwendet werden. Programming on Main wurde ebenfalls integriert. Hiermit ist es möglich alle CVs mit Ausnahme der CV 1 im laufenden Betrieb anzupassen. Ein Umsetzen der Lok auf ein Programmiergleis ist hierfür nicht erforderlich. Die 3,3V Variante verfügt über zwei zusätzliche Taster. Der obere dieser Taster wird für die Umschaltung der Funktionsblöcke benötigt. Der untere Taster ist für spätere Erweiterungen reserviert.

5V und 3,3V Varianten können ohne Einschränkung gleichzeitig betrieben werden (Abwärtskompatibilität).

Software

Die aktuelle SW kann aus dem geschützten MoBaSbS-Bereich heruntergeladen werden oder bei Frank Kaiser angefragt werden. Die 3,3V Variante kann nur in Betrieb genommen werden, wenn der BMC, der USC und das IFC den Softwarestand von 2018 aufweist.

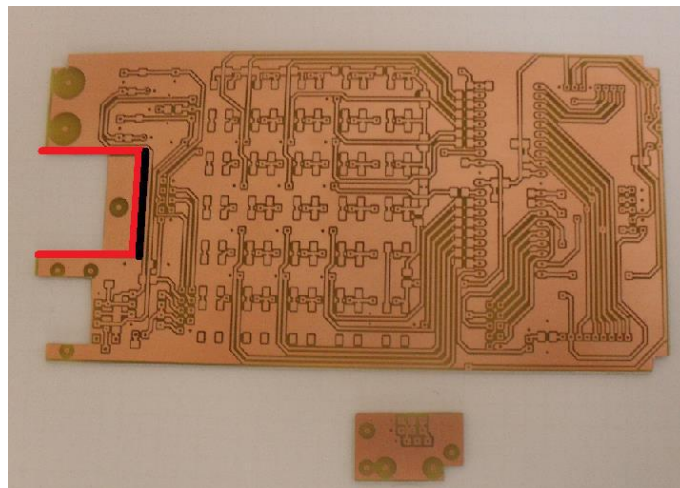


HDC 3,3V-Variante

Aufbauhinweise:

- Bei der Bauteilbestellung unbedingt darauf achten, dass die Widerstände für die Tastaturmatrix, das sind alle 200 Ohm, alle 1K Ohm und der 5,6 KOhm Widerstand, eine Toleranz von 1% oder weniger aufweisen. Widerstände mit 5 oder 10 % Toleranz sind ungeeignet. Es existiert eine Bauteilliste.

- Es wird ein Mikrocontroller vom Typ ATmega 644 verwendet. Die Anschlussbelegung weicht von der des sonst in der MoBaSbS verwendeten Typ ATmega 162 vollständig ab. Daher darf der Mikrocontroller nur eingebaut im HDC programmiert werden. Wird der Mikrocontroller ATmega 644 in das Programmierbord gesteckt droht Bauteilzerstörung.
 - Damit alle Komponenten des HDC in ein handliches Pultgehäuse passen, ist es erforderlich die RJ-45 Buchse auf einem separaten Platinenstück aufzubauen. Alle anderen Bauteile befinden sich auf der Hauptplatine.
 - Es empfiehlt sich die ICs mit Sockeln zu versehen.
 - Das LC-Display-Modul wird auf Buchsenleisten aufgesteckt.
 - Wird ein optoelektronischer Inkrementalgeber verwendet, ist die Aussparung in der Platine ganz herauszusägen. ACHTUNG: Die auf dem Bild mit schwarzer Linie gekennzeichnete Masseverbindung muss bestehen bleiben. Dieser Inkrementalgeber wird in den Gehäusedeckel eingebaut.
- Bei Verwendung eines mechanischen Inkrementalgebers wird nur die Hälfte der Aussparung aus der Platine gesägt und der Inkrementalgeber wird in die Platine eingebaut (sowie auf dem Bild dargestellt).
- Damit ein versehentliches Einstecken des HDCs in einen PMC nicht zu einer Zerstörung des HDCs führt - an den PMC-Ausgängen liegen +9V bis +12V an -, sollte der HDC mit dem 5V-Festspannungsregler ausgestattet werden. Auf den richtigen Sitz der Spannungswahlbrücke (Jumper) ist unbedingt zu achten. Falsches Stecken führt zur Zerstörung von Bauteilen.
 - Wird der HDC vom BMC ausschließlich mit 5 Volt versorgt, ist der Jumper umzustecken. In diesem Fall können die Bauteile C5, C6, C10, IC2 entfallen.



HDC 3.3V, Platinen-Layout

Das obige Bild zeigt die Hauptplatine und die Platine, die die RJ-45 Buchse aufnimmt. Beide werden am Schluss der Montage mit drei 2mm Schrauben verbunden. Elektrisch werden beide Teile mit vier Drahtbrücken verbunden. Sowohl auf der Platine für die RJ-45 Buchse wie auch bei der Hauptplatine müssen die Ecken weggesägt werden, da diese sonst die Gehäuseverbindungen stören würden (auf dem Bild gut zu sehen).

Der 5 Volt Spannungsregler braucht nur dann montiert zu werden, wenn das Fahrpult mit einer höheren Spannung versorgt werden soll (9V - 12V, vom BMC). Der Jumper ist entsprechend zu setzen. Es wird jedoch empfohlen, den Spannungsregler zu bestücken und auch zu nutzen. Damit ist ausgeschlossen, dass der HDC Schaden nimmt, wenn es aus Versehen in eine der PMC-Ausgangsbuchsen gesteckt wird. Denn dort liegen zur Versorgung der Peripheriemodule (Encoder und Decoder) 9-12V DC Spannung an!

Der Aufbau der Platine sollte in folgender Reihenfolge durchgeführt werden.

- Alle Brücken einlöten
- Alle SMD Bauteile einlöten
- Die beiden IC-Sockel einlöten
- Alle diskreten Bauteile einlöten

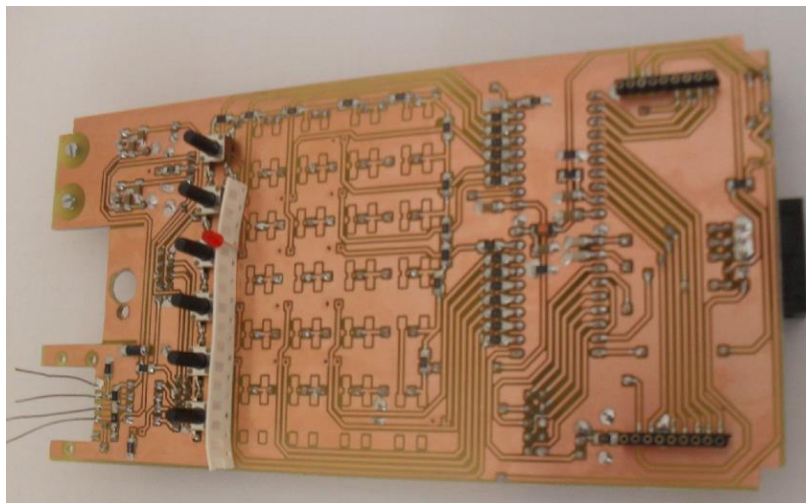
- Die ML 10 Buchsen sowie die Pfostenleisten für das Display einlöten (siehe kleine Platinkunde)
- Dann reihenweise die Taster mit den LEDs einlöten, von der Drehgeberseite anfangen.
- Die Platine mit der RJ-45 Buchse verschrauben und verlöten
- Inkrementalgeber vorbereiten und Display aufstecken

Bei der Bestückung der Platine mit den Tastern und den LEDs sollte man immer zeilenweise vom Inkrementalgeber ausgehend vorgehen, sonst wird das Löten unnötig erschwert.

Die LEDs sollten nicht zu weit aus dem Gehäuse heraus schauen. Sie können ca. 1 bis 2mm tiefer als der Tasterknopf eingebaut werden. Bei 2mm verschwinden sie fast plan im Gehäuse.

Ein Tipp dazu: Wenn man einen 6mm breiten und 60mm langen Pappstreifen zwischen die Beinchen der LEDs einer Reihe steckt, können die LEDs in der Höhe beim Verlöten optimal justiert werden. Am besten ist es, erst nur ein Beinchen jeder LED einer Reihe anzulöten und dann den Pappstreifen wieder herauszuziehen. Jetzt die LEDs nochmals ausrichten und das zweite Bein anlöten.

Die Tasten müssen sehr genau beim Festlöten positioniert werden.

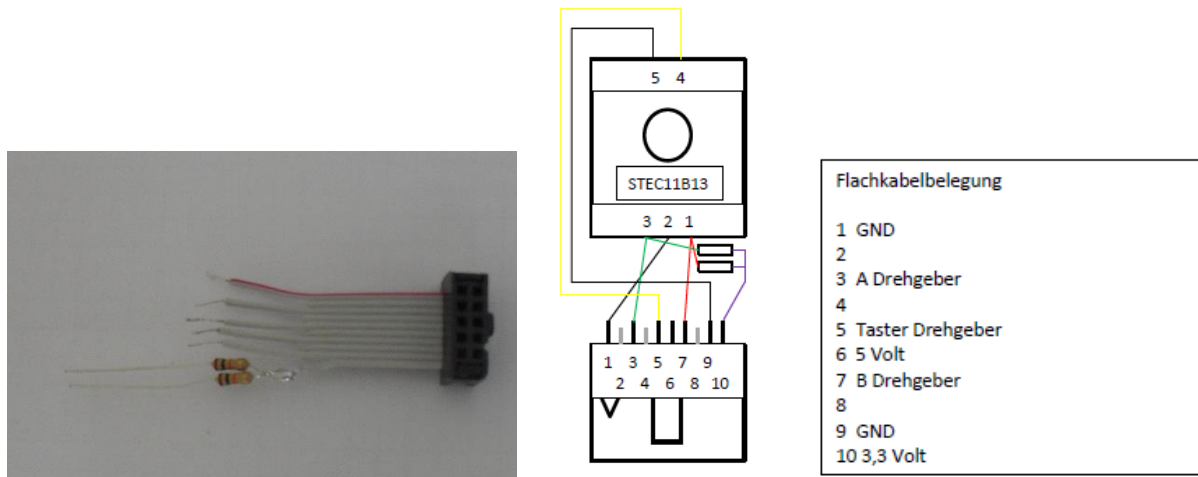


Einbau der Taster und LED

Ebenso sind deren Bohrungen im Gehäuse und die der LEDs exakt auszuführen. Dazu kann man das Layout ausdrucken und von innen in das Gehäuse kleben. Danach bohrt man mit einem 1mm-Bohrer genau die angezeigten 'Fadenkreuze' vor. Von der Oberseite werden die LED Bohrungen mit 3mm und die Tastenbohrungen mit 3.5mm erweitert (Bohrmaschine im Bohrstand langsam laufen lassen, damit die Bohrer nicht heiß werden und das Material schmilzt!). 3,5 mm ist schon recht knapp und die Bestückung muss schon recht genau sein.

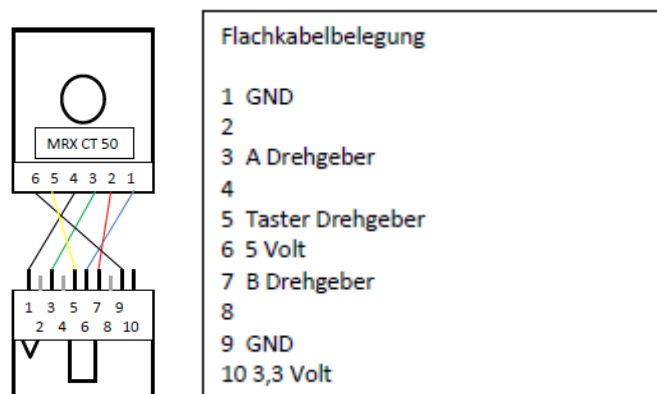
Falls Tasten klemmen, kann man die Bohrungen von der Innenseite ansenken. Generell gilt für die Arbeit an den Aussparungen: Lieber ein paar Mal mehr zusammenbauen, als sich nachher über zu große Ausschnitte ärgern. Die unteren Befestigungsnocken im Gehäuse müssen etwas gekürzt werden, dann steht einem endgültigen Einbau der Platinen nichts mehr im Wege.

Der Inkrementalgeber wird über ein Stück Flachbandkabel angeschlossen. Das nachstehende Bild zeigt die Vorbereitung des Anschlusskabels für die Verwendung des mechanischen Inkrementalgebers von ALPS. Für diesen, sind wie auf dem Bild zusehen, an die Leitung zehn zwei 10KOhm Widerstände zu löten. Ggf. ein Stück Schrumpfschlauch über die Widerstände ziehen. Auf der rechten Zeichnung wird die Anschlussbelegung dargestellt.



Anschluss des mechanischen Inkrementalgebers von ALPS

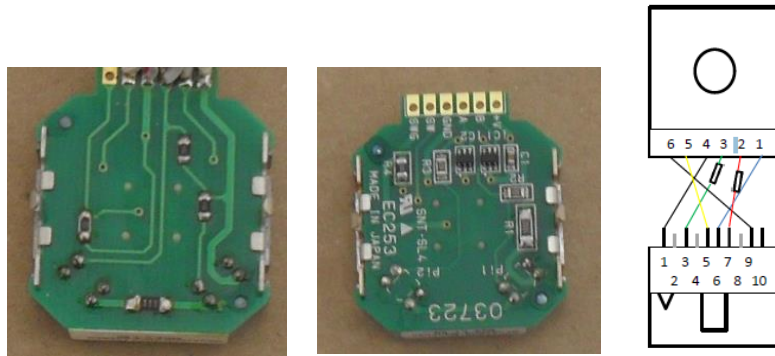
Das nachstehende Bild zeigt den Anschluss des 5 Volt optoelektronischen Inkrementalgebers von MEGATRON an den 10poligen Wannenstecker. Wird die 3,3 Volt Variante des optoelektronischen Inkrementalgebers von MEGATRON verwendet, so ist der Pin 1 des Inkrementalgebers (blaue Leitung) mit der Leitung 10 statt der Leitung 6 des Flachbandkabels zu verbinden.



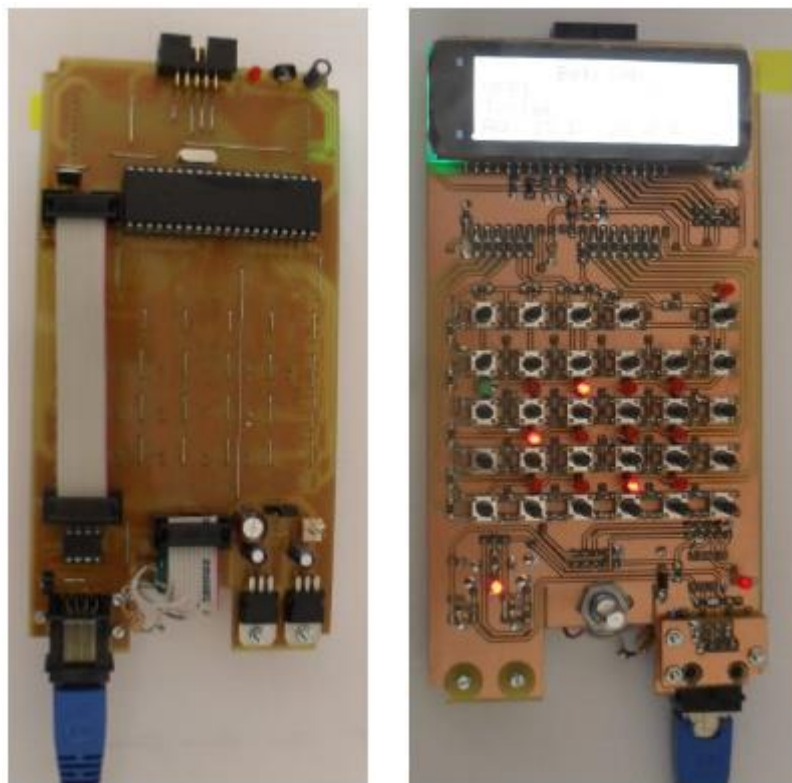
Anschluss des optoelektronischen Inkrementalgebers von Megatron

Beim optischen Drehgeber der Firma MEGATRON gibt es eine neuere und eine ältere Version. Die beiden Varianten sind auf den folgenden Bildern zu sehen. Links die ältere Version, in der Mitte die neuere Version.

Bei der neueren Version des Drehgebers müssen in die Leitungen A und B jeweils 10 KOhm Widerstände gelötet werden. Siehe rechtes Bild. Bei der älteren Version ist nur das Flachbandkabel anzulöten.



Optischer Inkrementalgeber von Megatron



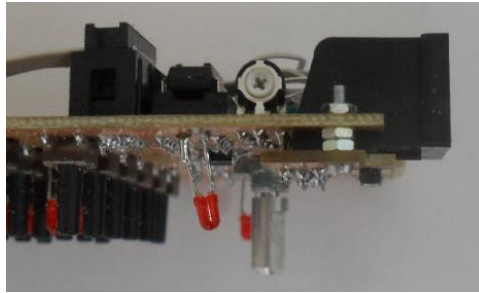
Fertig bestückte Platinenober und -unterseite

Die Einstellung des Displaykontrastes wird mit dem Poti am oberen Rand der Platine (neben der Programmierbuchse) durchgeführt. Mit dem Poti an der RJ-45 Buchse kann der Widerstand des RS-485 Busses angepasst werden. Dieser sollte zunächst, wenn man auf den Schraubenschlitz blickt ganz, nach links (entgegen dem Uhrzeigersinn gedreht sein). Sollte keine Buskommunikation zu Stande kommen, ist das Poti auf Mittelstellung zu stellen.

Ebenfalls ist auf dem linken Bild der vorherigen Seite die Stellung des Spannungswahljumpers zu sehen. So ist dieser zu stecken, wenn der HDC mit einer höheren Spannung als 5 V betrieben wird (Dieses ist die Empfehlung). ACHTUNG! Falsches Stecken kann Bauteilerstörung nach sich ziehen.

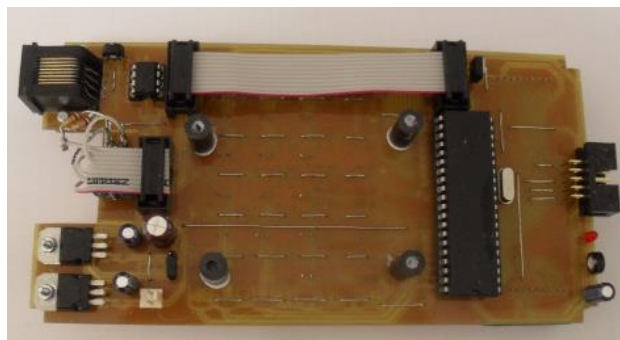
Rechts daneben befindet sich wie auf allen Basismodulen ein Stecker für die externe Spannungsversorgung.

Nachstehend ist die Verbindung zwischen der Platine und der RJ-45 Buchse zu sehen. Zunächst die 2 mm Schraube in die Platine mit der RJ-45 Buchse, dann zwei Muttern als Abstandshalter, dann die Platine und zum Abschluss wieder eine Mutter. Die Schrauben nicht anders herum montieren, da sie sonst im Gehäuse stören.



Verbindung Platine und RJ45-Buchse

Um die Platine mechanisch zu entlasten, empfiehlt es sich vor dem Einbau in das Gehäuse 4 Distanzhülsen im Bereich der Tastatur einzukleben. Die vier Distanzhülsen auf dem Bild wurden von 15 mm auf ca. 13 mm gekürzt.



Distanzhülsen

Aus dem Boden des Gehäuses müssen die Befestigungsnocken wegbohrt werden. Am besten mit 8 mm vorsichtig bohren.



Gehäuse ohne Befestigungsnocken

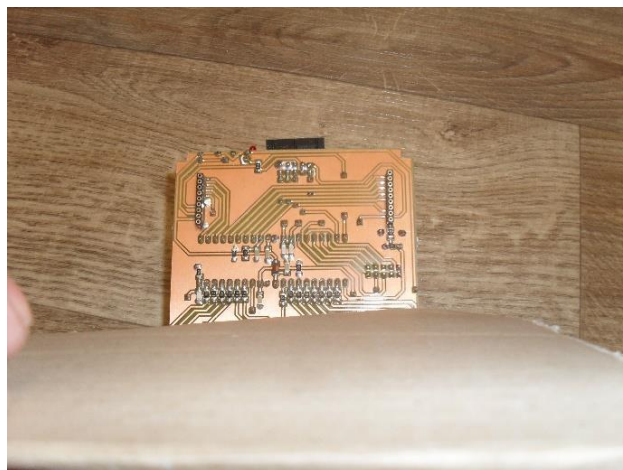
Hinweise zur Platinenherstellung

Selbstgefertigte Platinen sind nach dem Belichtungs-, Ätz- und Reinigungsvorgang sofort mit einem Schutzlack zu lackieren, da nur so eine schädliche Oxidation vermieden werden kann.

Meistens wird hierfür der Lötack SK 10 verwendet. Er besitzt neben dem oxidativen Schutz gleichzeitig ein Lötlösungsmittel. Dieses erleichtert den Lötprozess. Allerdings sind nach dem Löten die Lötreste und der Schutzlack wieder zu entfernen. Hierfür liefert die Industrie spezielle Reiniger.

Alternativ habe ich mit Aceton ebenfalls hervorragende Ergebnisse erzielt. Aceton gibt es in jedem Baumarkt oder Malerfachgeschäft zu kaufen. Das Reinigen der gelöteten Platinen erfolgt am besten mit einer handelsüblichen Zahnbürste und Aceton. Danach die gesamte Platine nochmals mit einer Ladung Aceton abspülen. Da Aceton stark verdunstet, entsteht Verdunstungskälte. Die Folge ist ein Tauwasserniederschlag auf der frisch gereinigten Platine. Dieser lässt sich mit einem Föhn schnell verdunsten. Danach ist die fertige Platine erneut vor Oxidation zu schützen, dies erfolgt am besten mit Plastiklack Plastik 70.

Beim Aufbau des HDC muss von dieser grundsätzlichen Behandlungsweise jedoch abgewichen werden. Der Grund dafür ist die Bauweise der Taster, da diese nicht Flüssigkeitsdicht sind. Also zunächst den HDC wie beschrieben aufbauen, jedoch ohne die Taster und die LEDs. Das Display nicht aufstecken und auch die IC Fassungen nicht bestücken. Jetzt die Platine ordentlich wie beschrieben mit Aceton reinigen und trocknen. Danach Taster und LEDs einlöten. Als nächstes den Plastiklack 70 aufbringen. Die Taster dürfen nicht mit besprüht werden. Wie auf dem Bild mit einem Stück Karton schützen. Wenn man die Platine dabei etwa 30 Grad neigt läuft von alleine zu viel aufgesprühter Schutzlack auf der Platine in die Zwischenräume der Taster. Die Platine dazu mehrfach drehen. Auf keinen Fall darf der Plastiklack von oben in die Taster hineinlaufen. Falls kleine Flächen zwischen den Tastern Lackfrei bleiben, ist dieses zu tolerieren.



Platine mit abgedeckten Tastern

Zum Einbau der beiden Buchsenleisten für das LCD-Display benötigt man etwas Löterfahrung. Als erstes sollte die Platine fest fixiert und um ca. 45 Grad geneigt sein. Pin für Pin löten. Lötcolben hierzu immer aus Richtung der Leiterbahn rechtwinkelig ansetzen. Dünne Lötspitze verwenden. Vor dem Einlöten der Buchsenleisten diese mit etwas SK 10 Lötack einsprühen und ca. 20 Minuten trocknen lassen, dass verringert die Wahrscheinlichkeit kalter Lötstellen. Jede Lötstelle mit der Lupe auf Brücken zu benachbarten Pins kontrollieren. Auch auf kalte Lötstellen prüfen. Danach nochmals mit Ohmmeter alles prüfen. Sorgfältiges, ruhiges und überlegtes Arbeiten beim Einlöten der Taster und der Buchsenleisten auf der Lötseite der Platine sichern einen störungsfreien Betrieb des HDC.

7.6 DPC

Der Decoder-Programmier-Controller (DPC) dient zum Programmieren der CVs (ConfigurationVariables) von DCC-Decodern. Da der DPC quasi zeitgleich in zwei Varianten entwickelt wurde, zeigt die nachfolgende Tabelle eine Gegenüberstellung der beiden Varianten.

	DPC (MoBaSbS)	DPC (RC5)
Anbindung & Schnittstellen	als Stand-Alone Gerät in Verbindung mit einem PC	als Stand-Alone Gerät, PC nicht notwendig
	als Modul der MoBaSbS (intern oder extern)	
	USB/RS232/XBus (RS485)	USB in Vorbereitung
Bedienung & Anzeige	MoBaSbS-Konfigurationssoftware	TV-Fernbedienung (RC5) & LCDisplay
	TrainProgrammer (Freiwald-Software)	
	DecoderProgrammer (Henning Voosen)	
Funktionen	CV lesen/schreiben/prüfen (Direct-Mode)	CV lesen/schreiben/prüfen (Direct-Mode)
	Speichern der Decoderdaten in einer Datei	Speichern der Decoderdaten in einem EEPROM
	Vorlagen für Decoder ladbar	
	Acknowledge Erkennung über Softwareabgleich	Acknowledge Erkennung über Hardwareabgleich
	Mini-Booster integriert	Mini-Booster integriert
unterstützte Decoder (getestet)	Kühn N025	Kühn N025
		Kühn T145/T125
	ESU Basic	ESU Basic
	ESU Lokpilot 3	ESU Lokpilot 3
	ESU Lokpilot V3.0 micro	ESU Lokpilot V3.0 micro
		ESU Lokpilot 3.5
		TRIX
	Zimo MX620	
	Lenz Gold mini	
	TAMS LD G-23	
TAMS LD33G		

7.6.1 RC5-Variante

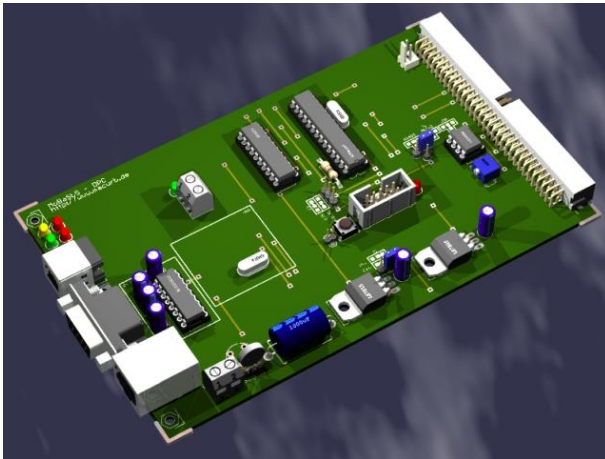
Die Konzeption und Programmierung wurde von Günter Lintzmeyer durchgeführt.

Das DPC_RC5_USB-Modul ist eine stand-alone-Lösung, die weder hardwareseitig noch softwareabhängig in die MoBaSbS integriert ist. Das DPC_RC5_USB-Modul wurde bewusst in dieser Form konzipiert und realisiert, damit man unabhängig (z.B. auf einer Test-Anlage / in einem anderen Raum, bei Modellbahn-Freunden, die eventuell noch keine MoBaSbS besitzen) die DCC-Programmierung ohne MoBaSbS und ohne PC-Anbindung (wahlweise) durchführen kann. Die Eingabe erfolgt per TV-Fernbedienung; die Ausgabe auf einem LCD-Display (2 x 20 Zeilen). Eine Funktionsbeschreibung und alle weiteren Informationen sind auf den MoBaSbS-Seiten von Patrick Urban unter DPC / DPC-RC5 hinterlegt.

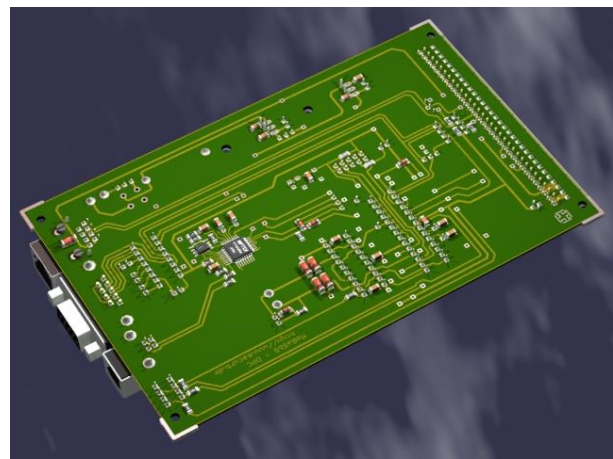
Der RC5-Lokprogrammer beinhaltet auch die Möglichkeit die CV-Daten von max. 250 Loks auf einem externen I2c-EEPROM zu speichern ("zu sichern") und wieder zurückzuholen.

Im gleichen Zusammenhang können auch "Lok-neutrale" Decoderdaten hinterlegt und für neue Loks weiterverwendet werden. Darüber hinaus besteht auch die Möglichkeit mit dem RC5-Lokprogrammer auch den vollen Funktionsumfang des Software-Paketes "DecoderProgrammer" (von Henning Voosen) zu verwenden; auch dort ist die Verwendung von Decoder-Dateien vorgesehen.

7.6.2 MoBaSbS-Variante



DPC (Bestückungsseite)



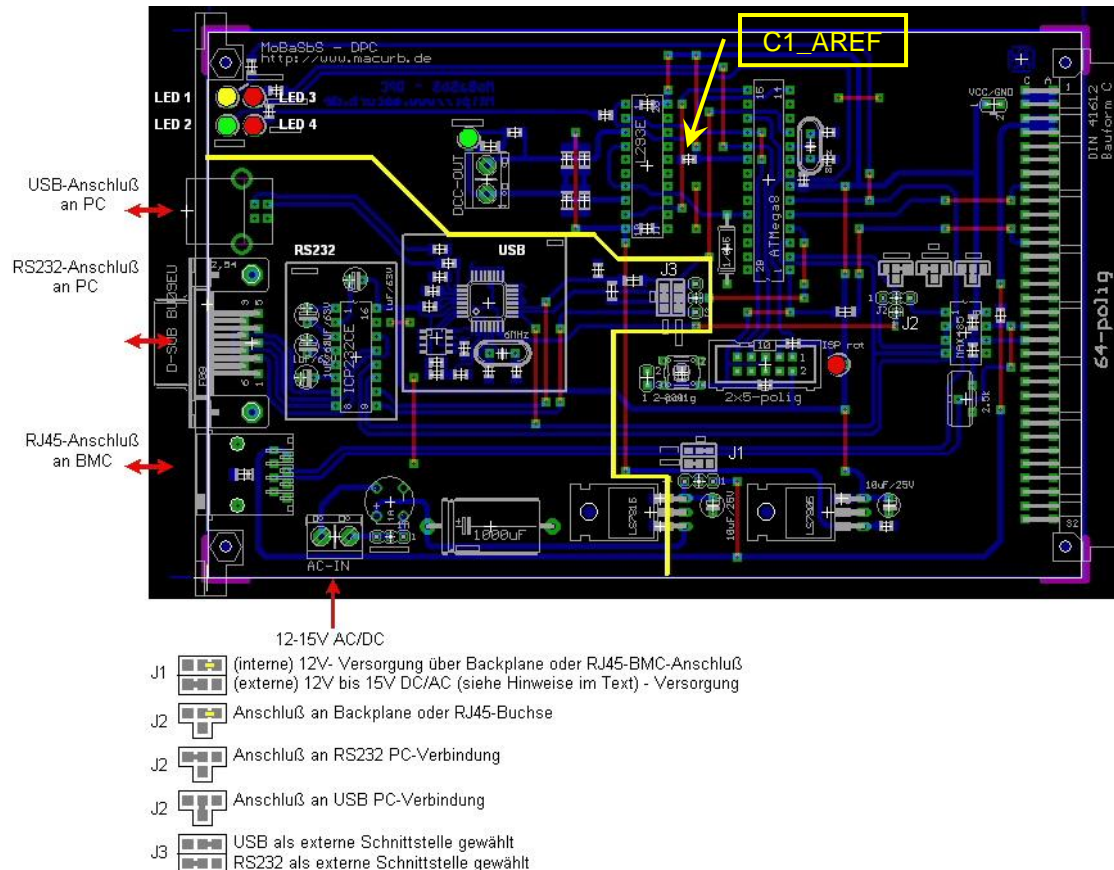
DPC (Lötseite)

Die Software und die Schaltung wurden von Uwe Mäusezahl entworfen; die Hardware hat Patrick Urban nach diesen Angaben entwickelt. An dieser Stelle ein Dank an Uwe für die freundliche Bereitstellung der Informationen und an Patrick für die Umsetzung.

Aufbauhinweise:

- An Pin 21 (Aref) des Mikrocontrollers fehlt ein Kondensator (10nF), der gegen Masse geschaltet werden muss. Der Kondensator C1_AREF (33pF) kann entfallen (siehe auch Fehlerbilder weiter unten)
- Bei industriell gefertigten Platinen ist zu beachten, dass die Bauteilbezeichnungen teilweise falsch sind. Als Referenz ist deshalb immer die Bauteilbezeichnung der Platinen-Layouts aus dem geschützten MoBaSbS-Bereich zu verwenden.
- Auf dem DPC wird ein Mikrocontroller vom Typ ATmega8 (Atmel) eingesetzt.
- LED 1 (gelb) oben links im Bild zeigt den Kommunikationsstatus des Moduls an, wenn der DPC als Modul der MoBaSbS betrieben wird (X-Bus Kommunikation).
- LED 2 (grün) zeigt den aktiven Zustand des DPC an. Leuchtet diese LED, so liegt eine Spannung am Ausgang des DPC an.
- LED 3 (rot) leuchtet bei Kurzschluss am Decoder (Strom >250mA).
- LED 4 (rot) zeigt durch kurzes Aufblitzen die Quittierung (Acknowledge) des Decoders auf einen Schreib-/Lesebefehl an.
- Die Platine ist so ausgelegt, dass sie in ein 19"-Gehäuse mit Frontplatte passt.

Einstellmöglichkeiten



DPC, Platinen-Layout

Die gelben Markierungen im Layout zeigen folgendes an:

Bei Anschluss des DPC als Modul der MoBaSbS über die Backplane brauchen die im unteren linken Bereich gelb-umrahmten Bauteile nicht bestückt zu werden (gilt auch für die Bauteile auf der Unterseite). Außerdem können die Bauteile zur externen Spannungsversorgung dann entfallen.

Hinweise:

Für den korrekten Betrieb des DPC müssen die Updates des BMC und USC vom 19.10.2008 eingespielt sein.

Zum Betrieb mit TrainProgrammer V7.0 siehe Kap. 13.3.4.

In der vorliegenden Version werden nur Decoder unterstützt, welche sich Bitweise auslesen lassen. Dies sollten alle neuen Decoder (seit 2002) beherrschen. Der Adress-Only-Modus und die seitenweise Programmierung werden nicht unterstützt.

Eine Verbindung des Programmiergleises über die Masse mit der Anlage muss ausgeschlossen werden, da der DPC seinen Ausgang während des Programmierens umpolt. Dies gilt auch für Fahrten über eine Trennstelle zum Programmiergleis, da sonst die Masseverbindung über die Lokomotive hergestellt wird. Der dann fließende hohe Strom würde zur Zerstörung des L293 führen.

Manche Decoder bereiten Probleme, wenn sie nur mit den 12V von der RJ45 Buchse versorgt werden. Dabei reagiert der Decoder überhaupt nicht oder nicht auf alle CV's. Abhilfe schafft hier eine externe Stromversorgung zu verwenden und den Jumper auf extern zu stecken.

Der DPC liest vor dem Auslesen einer CV die aktuelle Stromaufnahme der Lok aus und addiert anschließend 50mA zu diesem Wert hinzu. Überschreitet nun die Stromaufnahme beim Warten auf das ACK diesen errechneten Wert, so wird dies als positives ACK erkannt. Das ACK wird durch ein Zuschalten des Motors erzeugt. Deswegen ruckt die Lok auch immer ein wenig. Es ist deshalb auch darauf zu achten, die Lok auf einem sauberen Gleis zu programmieren.

Bei der Programmierung eines Funktionsdekoders muss diese Last künstlich erzeugt werden, da hier kein Motor vorhanden ist. Dies ist z. B. durch anschließen oder durch Parallelschalten eines Widerstandes zu den vorhandenen Verbrauchern zu realisieren. Sollte dies nicht funktionieren, kann die Mindeststromaufnahme auch im EEPROM geändert werden. Bei Bedarf kann U. Mäusezahl die Vorgehensweise erläutern.

Das Verfahren ist bis zu einer Stromaufnahme von 250mA realisiert. Dann wirkt die Kurzschlusserkennung um den Decoder zu schützen, wenn er evtl. verkehrt eingebaut wurde. Laut NMRA soll die Stromaufnahme 100mA nicht überschreiten und es sollen alle Verbraucher abgeschaltet werden.

Die LED an der DCC-OUT-Klemme wird direkt vom Controller über den Freigabeeingang des L293E angesteuert. Wird der Gleis Ausgang kurzgeschlossen, blinkt die LED mit der LED3 im Wechsel. Damit werden die 250mA überwacht. Aber Vorsicht: Ist ein Fehler im Aufbau vorhanden, so können Bauteile in Mitleidenschaft gezogen werden. Diesen Test bitte nicht nach einem Fahr- oder Funktionsbefehl durchführen, da dann der Kurzschlussstrom deutlich höher ist.

Der DPC kann CVs bis 1024 auslesen und schreiben.

Dies ist mit dem Trainprogrammer oder über die Befehlszeile im Konfigtool (Kap. 12.9) möglich (Die Möglichkeiten des DCC-Programmers im Konfigurationstool stehen im Abschnitt 12.7.6).

Der Trainprogrammer ab Version7 unterstützt das LENZ Protokoll in der Version 3.6 und somit auch die CVs bis 1024. Das Auslesen funktioniert auch mit der Demoversion für alle Decoder, nur für das Schreiben braucht man die Vollversion. Umgehen kann man dies durch die Verwendung des Direktprogrammers (ist ein Teil vom Trainprogrammer). Damit lassen sich alle CVs schreiben und lesen. Das Verfahren ist aber für die Programmierung einer Geschwindigkeitskennlinie sehr umständlich

Bei Auslesen von CVs blitzt die LED4 kurz auf. Dies zeigt an, dass der Decoder den Befehl verstanden hat und das es eine Übereinstimmung mit dem angefragten Bit gibt. Es wird insgesamt 8x pro CV durchgeführt. Kommt es hier zu Fehlern, so wird eine Fehlermeldung vom DPC abgesetzt. Diese Meldung interpretiert jede Software etwas anders. Es kann auch vorkommen, dass der Decoder die angefragte CV nicht unterstützt. Manche senden in diesem Fall eine Null zurück, andere geben keine Antwort auf diese Anfrage, was dann ebenfalls zu einer Fehlermeldung führt.

Der fehlende Kondensator am PIN21 (Aref) des ATmega8 kann zu folgenden Fehlerbildern führen:

Fehlerbild 1

Lok kann fahren, aber nicht programmiert werden

Die LEDs am DPC zeigen folgende Reaktionen:

LED3 leuchtet dauerhaft, wenn keine Lok auf dem Gleis steht; LED2 und LED am DCC-OUT blinken synchron, wenn keine Lok fährt. Das Blinken der beiden LEDs ist ungleichmäßig. Wird Spannung an der Platine zugeschaltet, blinken diese nicht so schnell (ca. 1 Sekundentakt), nach ein paar Sekunden wird das Blinken dann immer schneller.

Fehlerbild 2

Beim Auslesen wird ein Kurzschluss gemeldet, auch wenn keine Lok angeschlossen ist, also DCC-OUT offen bleibt.

Die LEDs am DPC zeigen folgende Reaktionen:

LED 1 (gelb X-Bus) in der MobaSbS reagiert auf den X-Bus, als Stand-alone ist sie natürlich aus

LED 2 (grün, aktiver DCC-Out) blitzt kurz auf

LED 3 (rot, Kurzschluss) leuchtet fast ständig, geht aus wenn LED 2 aufblitzt

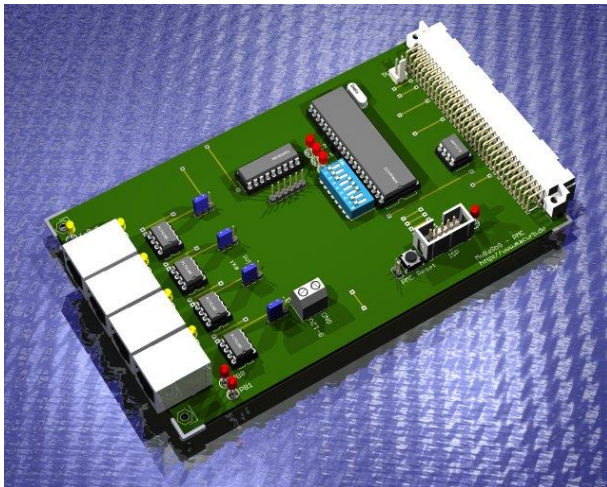
LED 4 ist aus.

LED an DCC-Out blitzt synchron mit LED 2 auf.

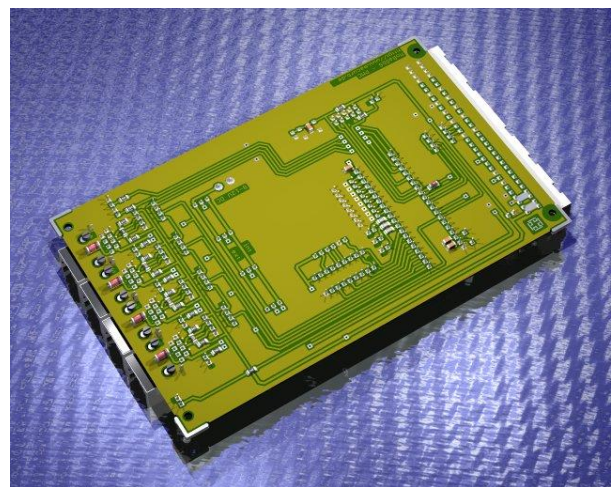
Programmierung des DPC-USB-Chips:

Die Programmierung des FT 245 B... erfolgt auf die gleiche Weise wie beim IFC (siehe Kap. 7.3).

7.7 PMC



PMC (Bestückungsseite)



PMC (Lötseite)

Der PMC ist das Modul, an dem Peripheriemodule (PM), also Decoder oder Encoder angeschlossen werden. Er gibt Schaltbefehle (z.B. vom PC) an die an ihn angeschlossenen Decoder weiter. Ebenso fragt er die angeschlossenen Encoder und rückmeldefähigen Decoder zyklisch ab und gibt Veränderungen des Zustandes an andere Module (z.B. dem PC) weiter.

Die Decoder und Encoder sind mit dem PMC über eine differentielle Leitung (RS485) verbunden. Dies führt zu einer sehr sicheren und sehr schnellen Datenverarbeitung.

Ein PMC unterstützt bis zu vier Stränge. Der Einbau eines zweiten PMC ist möglich. Der Anschluss von Peripheriemodulen kann an jedem Strang des PMC erfolgen. Auf eine Adress-Reihenfolge oder eine Typ-Abhängigkeit braucht nicht geachtet zu werden. Der PMC ermittelt den Typ bei einem Scan der PM automatisch.

An einen PMC-Strang dürfen bei Verwendung eines RS485-Treibers nicht mehr als 32 Peripheriemodule angeschlossen werden. Bei Einsatz eines RS487-Treibers sind bis zu 128 Peripheriemodule möglich.

Insgesamt können 256 PM adressiert werden. Damit stehen maximal 1024 Rückmeldekontakte oder bis zu 2048 Schaltkontakte zur Verfügung.

Bei Betrieb mit Traincontroller sind die Hinweise im Kapitel 13.3 zu beachten.

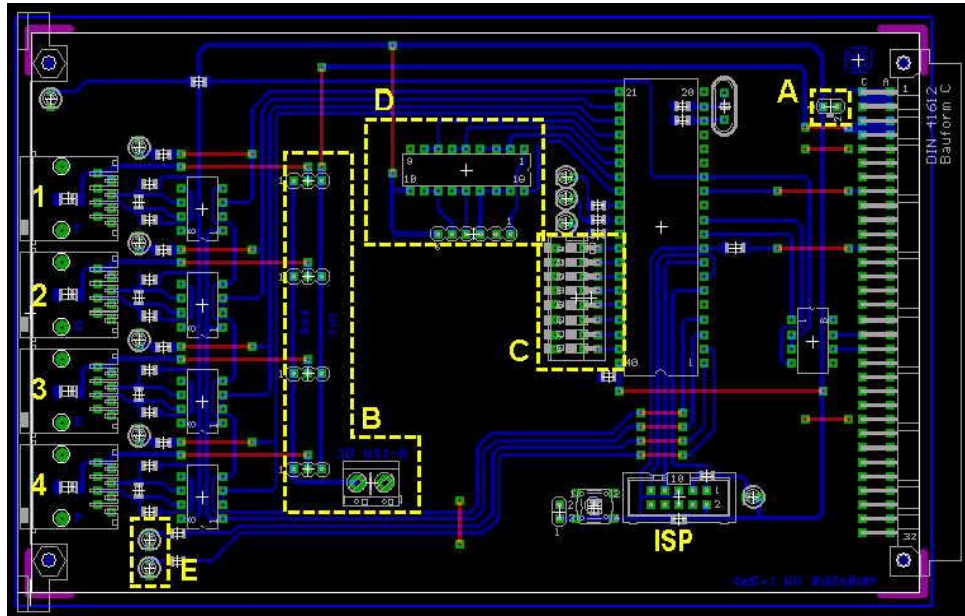
Hinweise:

- Nach dem Programmieren des PMC ist die Default-Adresse 31 eingestellt. Diese muss mit dem MoBaSbS-Konfigurationstool beim ersten Einsatz geändert werden.
- Beim Laden der Peripheriemodule mit der Software ist die Default-Adresse 255 eingestellt. Diese muss mit dem MoBaSbS-Konfigurationstool beim ersten Einsatz geändert werden. Die Adresse 255 hat somit eine Sonderfunktion und sollte für den Betrieb nicht vergeben werden.

Aufbauhinweise

- Bei industriell gefertigten Platinen ist zu beachten, dass die Bauteilbezeichnungen teilweise falsch sind. Als Referenz ist deshalb immer die Bauteilbezeichnung der Platinen-Layouts aus dem geschützten MoBaSbS-Bereich zu verwenden.
- Von dem Reset-Taster verdeckt liegt ein 100nF SMD Kondensator, der mit einem 10k SMD Widerstand (unterhalb der ISP-Buchse) zusammen für ein sauberes Reset-Verhalten des μ Cs sorgt.

Einstellmöglichkeiten



PMC, Platinen-Layout

Die gelben Markierungen im Layout zeigen folgendes an:

- A** Hier können +5V eingespeist werden. Dadurch kann ein Board außerhalb des Gehäuses programmiert werden.
- B** Die angeschlossenen Encoder und Decoder (nur der Elektronik-Teil) können über die 4 Stränge entweder durch eine externe Spannungsquelle (9V bis 12V DC) oder intern über die Backplane versorgt werden. Dafür stehen auf dem PMC für jeden Strang getrennt Jumperoptionen zur Verfügung. Die Jumper werden an den Positionen "int" (Jumper auf die beiden rechten Pins) gesetzt, wohingegen über die Leiste "ext" (Jumper auf die beiden linken Pins) eine externe Versorgung eingespeist werden kann. (siehe Kap. 6.2)
- C** An dieser Stelle kann wahlweise ein stehender oder liegender 8pol DIP-Switch montiert werden. Der DIP-Switch ist für zukünftige Erweiterungen vorgesehen. Zurzeit wird er nicht unterstützt.
- D** Dieser Stecker erlaubt evtl. in einem kommenden Update der μ C-Software den Anschluss des S88 Rückmeldesystems. Derzeit hat er keine Funktion. Ebenso braucht der IC (ULNxxx) oberhalb des Steckers nicht bestückt zu werden.
- E** Die LEDs dienen Debugzwecken und brauchen nicht bestückt zu werden.

8 Peripherie-Module

8.1 Software-Varianten, Prozessortausch und Adapterplatinen

Ursprünglich waren alle Decoder mit einem ATtiny2313 ausgerüstet. Aufgrund weiterer Ergänzungen in der Software (Bootlader-Variante von G. Lintzmeyer) reichte der Speicher des ATtiny2313 nicht mehr aus und es wurde auf den ATmega8 ausgewichen. Da der ATtiny2313 und der ATmega8 nicht pin-kompatibel sind, muss bei einigen Decodern eine Adapterplatine verwendet werden, die in den Sockel des ATtiny2313 gesteckt wird. Damit lassen sich die bisherigen Decoder-/Encoder-Platinen weiterverwenden.

Zu beachten ist, dass bei der Umstellung auf den ATmega8 die eingestellten Werte für Servos, etc. nicht automatisch übertragen werden. Dies ist aber mit Hilfe der mobasbs.mbs-Datei möglich. Hierzu muss vor einem Wechsel der Prozessoren die mobasbs.mbs-Datei gespeichert worden sein (siehe Kap. 12.2 ff).

Nach dem Prozessortausch muss auf jeden Fall erst die eep-Datei des ATmega8 eingespielt werden, weil sich die Kennung (bisher: ATtiny2313 - jetzt: ATmega8) geändert hat. Damit sind alle Einstellungen (Servo-Ausschläge, etc) überschrieben. Danach muss die alte Adresse wieder eingestellt werden (255 > xxx). Nach Laden der mobasbs.mbs-Datei werden dann die alten Parameter wiedergefunden. Diese müssen dann noch in das EEPROM des ATmega8 geschrieben werden.

Eine Beschreibung, was beim Umstieg des Prozessors von ATtiny2313 auf ATmega8 und der Lintzmeyer-Bootlader-Funktion beachtet werden muss, steht im Anhang in Kap. 13.4

Software-Varianten

Aus der Weiterentwicklung der Software ergeben sich eine Vielzahl von Varianten, die nachfolgend dargestellt werden. Die Beschreibung der einzelnen Software-Funktionen steht im Abschnitt des jeweiligen Decoders.

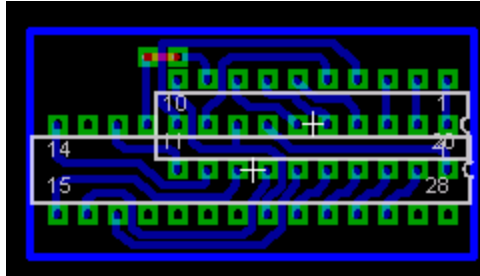
Platine	Adapter	Prozessor	Software	Bemerkung	
PM_FET PM_ULN	nein	AT90S2313	PM_DEC8_AT90S2313	Prozessor nicht mehr lieferbar, SW wird nicht weiter gepflegt	
		ATtiny2313	PM_DEC8	Software <i>ohne</i> Bootlader-Funktion, SW wird nicht weiter gepflegt	
			PM_LDEC_DB	SW wird nicht weiter gepflegt	
	ja	ATmega8		PM_LDEC_Light	SW wird nicht weiter gepflegt
				PM_DEC8_ATmega8_BL (Stand 2012-02-01)	Software <i>mit</i> Lintzmeyer-Bootlader-Funktion (siehe 13.4)
				PM_LDEC8	Software <i>ohne</i> Bootlader-Funktion, es sind nur 8 Ausgänge nutzbar, Ausgangsbez. und Konfig-Tool beachten, SW wird nicht weiter gepflegt
				PM_universal	Software <i>mit</i> Mäusezahl-Bootlader-Funktion (siehe 13.6.4)
		PM_LDEC_uni_T	Software <i>mit</i> Mäusezahl-Bootlader-Funktion (siehe 13.6.4)		
PM_Uni	nein	ATmega8	PM_universal	Software <i>mit</i> Mäusezahl-Bootlader-Funktion (siehe 13.6.4)	
			PM_LDEC_uni_T	Software <i>mit</i> Mäusezahl-Bootlader-Funktion (siehe 13.6.4)	
PM_LDEC wird nicht weiter gepflegt	nein	ATmega8	PM_LDEC8	Software <i>ohne</i> Bootlader-Funktion, SW wird nicht weiter gepflegt	
			PM_LDEC_uni_T	Software <i>mit</i> Mäusezahl-Bootlader-Funktion (siehe 13.6.4)	
PM_SDEC4	nein	AT90S2313	PM_SDEC4_AT90S2313	Prozessor nicht mehr lieferbar, SW wird nicht weiter gepflegt	
		ATtiny2313	PM_SDEC4	Software <i>ohne</i> Bootlader-Funktion, SW wird nicht weiter gepflegt	
	ja	ATmega8		PM_SDEC4_ATmega8_BL (Stand 2012-02-01)	Software <i>mit</i> Lintzmeyer-Bootlader-Funktion (siehe 13.4), enthält Erweiterung für Herzstück-Polarisierung und inv.Drehrichtung
				PM_universal	Software <i>mit</i> Mäusezahl-Bootlader-Funktion (siehe 13.6.4)
			PM_SDEC8_FB	SW kann eingesetzt werden, SDEC4-Platine ist für 8 Servos vorbereitet, SW wird nicht weiter gepflegt	
PM_SDEC8	nein	ATmega8		PM_SDEC8_FB	Software <i>ohne</i> Bootlader-Funktion, SW wird nicht weiter gepflegt
				PM_universal	Software <i>mit</i> Mäusezahl-Bootlader-Funktion (siehe 13.6.4)
				PM_SDEC8_FB_ATmega8_BL (Stand 2012-02-01)	Software <i>mit</i> Lintzmeyer-Bootlader-Funktion (siehe 13.4) <i>ohne</i> Erweiterung Herzstück-Polarisierung und inv.Drehrichtung
PM_ENC8	nein	AT90S2313	PM_ENC8T_AT90S2313	Prozessor nicht mehr lieferbar, SW wird nicht weiter gepflegt	
		ATtiny2313	PM_ENC8T	Software <i>ohne</i> Bootlader-Funktion	
	ja	ATmega8		PM_universal	Software <i>mit</i> Mäusezahl-Bootlader-Funktion (siehe 13.6.4)
			PM_ENC8T_ATmega8_BL (Stand 2012-02-01)	Software <i>mit</i> Lintzmeyer-Bootlader-Funktion (siehe 13.4)	
PM_ENC24	nein	ATmega162	PM_ENC24T	Software <i>ohne</i> Bootlader-Funktion	

Platinen-, Prozessor- und Software-Varianten

Adapterplatinen

Es gibt zwei Adapter-Varianten, die über den Online-Shop von Heinz-Jürgen Jorns bezogen werden können:

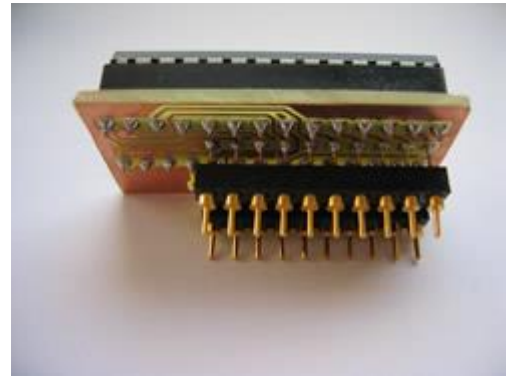
Adapter klein: (von Uwe Mäusezahl)



ATtiny2313/ATmega8-Adapter klein, Platinen-Layout



ATtiny2313/ATmega8-Adapter

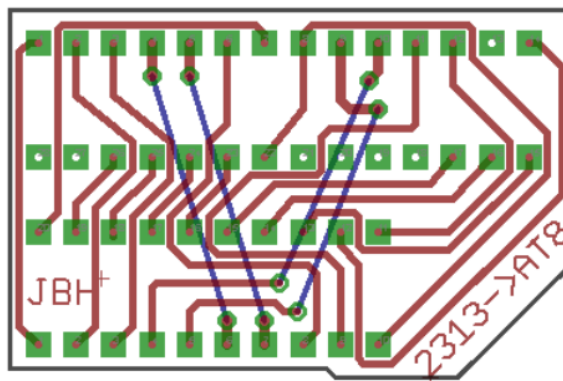


ATtiny2313/ATmega8-Adapter

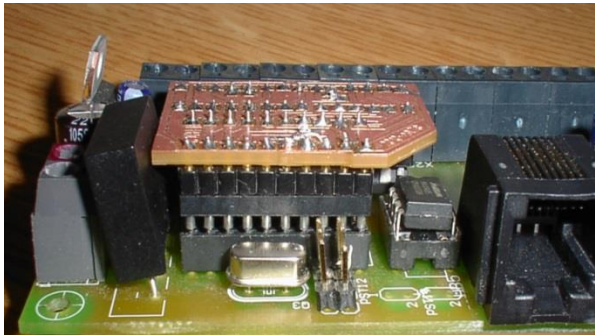
Die Stiftleisten (für das Einstecken in die "alte" ATtiny2313-IC-Fassung) müssen auf der Lötseite angebracht werden (Achtung: enger Abstand zwischen Stiftleiste und Lötseite!); die ATmega8-IC-Fassung wird auf der Bestückungsseite angebracht.

Nach dem Einbau im PM ist der ATmega8 von oben zugänglich

Adapter groß: (von Jörg-Bodo Häusler)



ATtiny2313/ATmega8-Adapter groß, Platinen-Lay-out



ATtiny2313/ATmega8-Adapter



ATtiny2313/ATmega8-Adapter

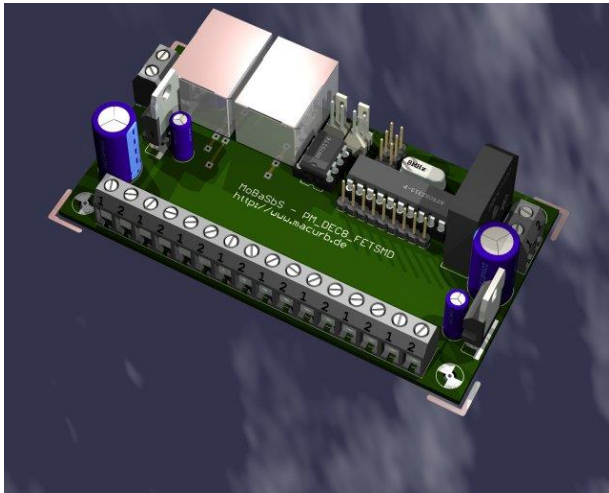
alle Bauteile werden auf der Bestückungsseite eingebracht. Es sind 4 Drahtbrücken erforderlich.

Nach dem Einbau im PM ist der ATmega8 nicht zugänglich (da nach unten gerichtet), nur die Lötseite ist sichtbar.

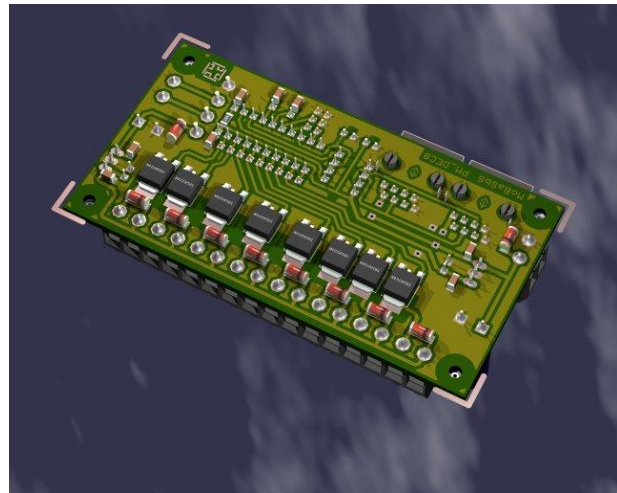
Damit der erforderliche Abstand zwischen Adapter und den Bauteilen auf der Platine gewährleistet ist, müssen entweder sog. Stapelleisten (Reichelt) oder eine IC-Fassung als Zwischenstecker verwendet werden.

8.2 Decoder

8.2.1 FET-Decoder



FET-Decoder (Bestückungsseite)



FET-Decoder (Lötseite)

Über die Decoder werden zu schaltende Verbraucher mit hohem Strombedarf wie etwa Weichen, Signale oder Lampen bedient. Pro Decoder lassen sich bis zu acht Einzelausgänge oder bis zu vier Ausgangspaare (z.B. für Weichen) schalten. Die Versorgungsspannung für den Elektronik-Teil wird auf jeder Decoderplatine separat erzeugt.

Über das MoBaSbS-Konfigurationstool können die Ausgänge mit unterschiedlichen Funktionen belegt werden:

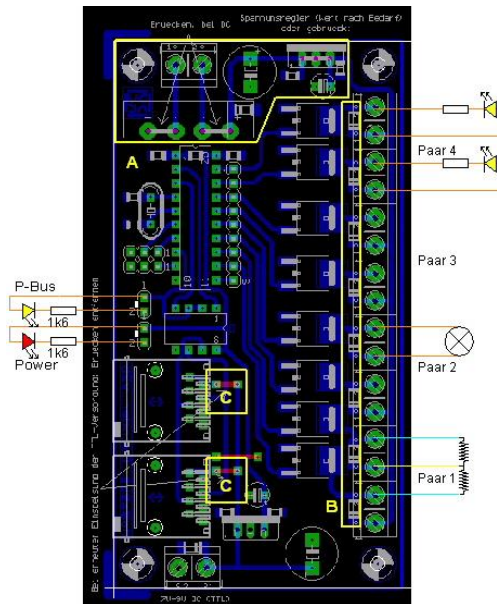
- Ausgänge werden durch Software ein- und ausgeschaltet (Normalfall)
- Ausgangspaar kann wechselseitig blinken. Die Blinkfrequenz kann für jedes Ausgangspaar unterschiedlich eingestellt werden.
Zum Beispiel können 2 oder 3 Ausgangspaare eines Decoders auf "Blinken" programmiert werden und damit z.B. das Baulicht einer Feuerwehr, eines Polizeiautos und eines Krankenwagens anzusteuern. Werden dann die Blinkfrequenzen leicht unterschiedlich gewählt und die Ausgangspaare auf die Fahrzeuge auch noch untereinander verteilt, so erhält man das typische Zusammen- und Auseinanderlaufen der Blinklichter.
Ebenso vorstellbar ist, dass ein Zug über einen Rückmeldekontakt die Blinklichter eines Bahnüberganges aktiviert. Es wird beim Einsatz des Decoders keine spezielle Blinkelektronik mehr benötigt.
- Ausgang kann nach einer festgelegten Zeit von selbst wieder ausschalten.
- Ausgangspaar kann sich wechselseitig ein- und ausschalten. Damit lassen sich einfache rot/grün Lichtsignale direkt vom Decoder ansprechen

Der FET-Decoder kann auch als LDEC-Decoder verwendet werden (siehe Kap. 8.2.5).

Aufbauhinweise:

- Bei industriell gefertigten Platinen ist zu beachten, dass die Bauteilbezeichnungen teilweise falsch sind. Als Referenz ist deshalb immer die Bauteilbezeichnung der Platinen-Layouts aus dem geschützten MoBaSbS-Bereich zu verwenden.
- Die beiden LEDs auf der linken Seite der Platine brauchen nicht bestückt zu werden. Sie dienen bei der Inbetriebnahme Diagnosezwecken. So 'blitzt' die P-Bus-LED einmal kurz beim Scannen der Peripheriemodule durch den PMC oder bei einigen Konfigurationskommandos auf. Mit der roten Power-LED kann schnell überprüft werden, ob die Versorgungsspannung am Decoder anliegt.

Einstellmöglichkeiten



FET-Decoder, Platinen-LAYOUT

Die gelben Markierungen im Layout zeigen folgendes an:

A

Hier können ein Gleichrichter und ein Festspannungsregler mit den zugehörigen Kondensatoren bestückt werden. An den zwei Klemmen oben wird eine unregulierte Wechselspannung angeschlossen, die gleichgerichtet und stabilisiert auf die Ausgangsklemmen geführt wird. Die Höhe der Spannung richtet sich nach der Eingangsspannung bzw. dem verwendeten Festspannungsregler.

Lässt man die Kondensatoren und den Festspannungsregler entfallen und setzt eine Brücke (Verbindung von Ein und Ausgang des Festspannungsreglers) ein, so steht eine unregulierte Gleichspannung zum Schalten zur Verfügung. Eine solche Spannung ist z.B. zum Schalten von Magnetartikeln ausreichend.

Eine weitere Möglichkeit besteht darin, dass man die Spannung an zentraler Stelle gleichrichtet (mit Trafo, Gleichrichter und Elkos oder ein Netzteil verwendet (z.B. aus einem PC oder Reichelt SNT FG-DP150-24)) und an den Eingangsklemmen einspeist. Dann können ebenfalls die Bauteile für Gleichrichter, Festspannungsregler, etc. entfallen.

B

Zwischen dem Massepotential und der Versorgungsspannung wird eine Freilaufdiode (nötig beim Schalten von Weichen mit Spulenantrieb, Relais) direkt auf der Decoderplatine aufgelötet.

C

Der μC wird über das Datenkabel mit Spannung versorgt. Diese wird an der Backplane angeschlossen und sollte ca. 9- 12 V DC betragen. Sie wird auf der Platine auf 5V stabilisiert. Bei sehr vielen Peripheriemodulen oder sehr langen Verbindungskabeln, kann der Spannungsabfall über die Versorgungsleitung zu groß werden. In diesem Fall kann eine DC-Spannung von 12V auf jedem Peripheriemodul neu zugeführt werden. Dafür muss eine der kurzen Drahtbrücken an der RJ45-Buchse entfernt werden, an der das Datenkabel vom vorherigen Modul ankommt. Über die andere Drahtbrücke an der 2ten RJ45-Buchse wird dann die Versorgungsspannung 'frisch' weitergegeben.

Hardware

Alle Verbraucher werden gegen Massepotential geschaltet. Auf der Decoderplatine sind dafür 8 Klemmen vorgesehen.

Es ist auch möglich, Verbraucher zu schalten, die durch unterschiedliche Spannungen versorgt werden. Es wird dann pro Schaltausgang nur die Masseklemme verwendet (die Klemme, die über den FET gegen Masse geschaltet wird). Das Versorgungspotential des Verbrauchers kommt jetzt von einer externen Quelle. Hier muss die Masse der externen Quelle mit der Masse der Decoderplatine verbunden werden.

Inbetriebnahme

Die Peripheriemodule werden über den PMC an die MoBaSbS angeschlossen. Wie bei allen anderen Peripheriemodulen auch, ist die μ C-Software per Default so konfiguriert, dass das PM-Modul auf Adresse 255 angesprochen wird. Um zu verhindern, dass ein Adresskonflikt auftritt, wenn ein weiteres neues Peripheriemodul an den PMC angeschlossen wird, muss die Adresse über das MoBaSbS-Konfigurationstool nach dem Anschluss des PM an den PMC geändert werden. Adressen von 0 bis 254 sind möglich. Ebenso wie die PM-Adresse werden auch alle weiteren Parameter des PM über das MoBaSbS-Konfigurationstool gesetzt.

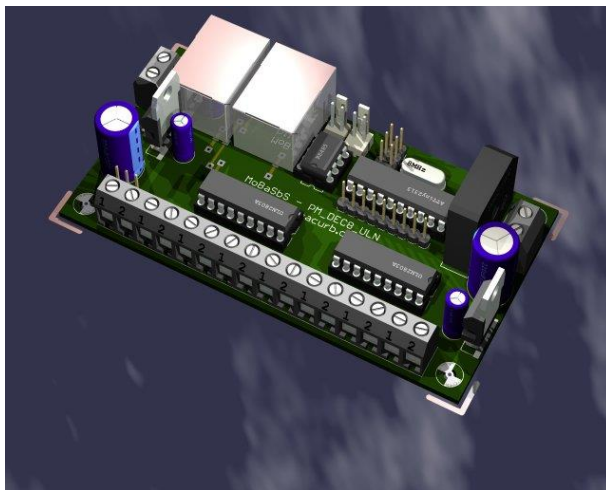
Software

Neben der Software PM_DEC8 für die Ansteuerung der 8 Ausgänge wurde von G. Lintzmeyer eine Variante für die Ansteuerung von Gaslicht-Laternen entwickelt.

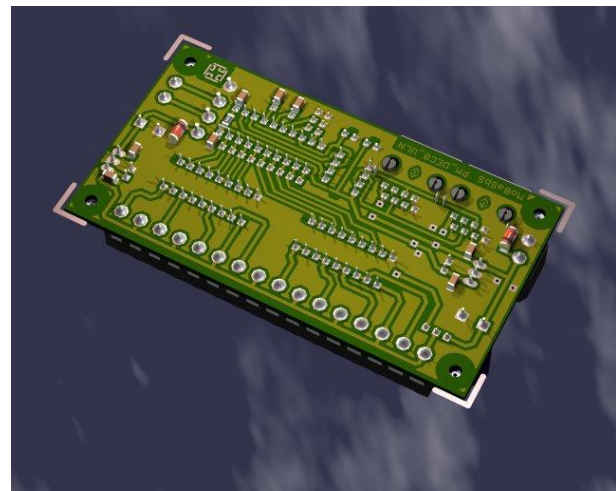
Damit können max. 3 Gaslicht-Ketten langsam ein- bzw. ausgeschaltet werden, ferner gibt es die Möglichkeit, die Lampen mal kurz flackern zu lassen. Andere Schaltmöglichkeiten auf dem gleichen Decoder sind nicht möglich.

Die Software kann bei ihm angefordert werden.

8.2.2 ULN-Decoder



ULN-Decoder (Bestückungsseite)

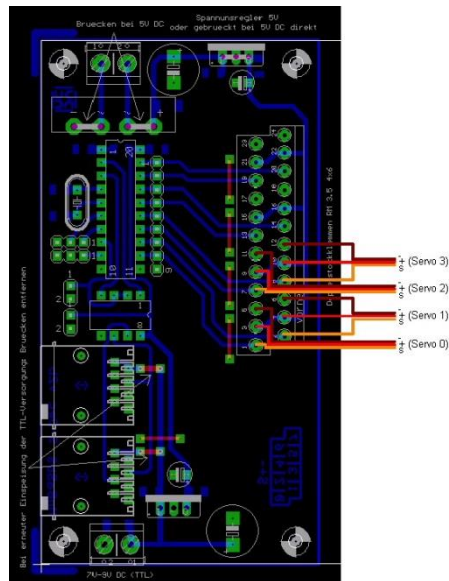


ULN-Decoder (Lötseite)

Im Gegensatz zum FET-Decoder wird bei diesem Decoder als Ausgangstreiber ein ULN2803 eingesetzt. Dies ist ein 18poliges-IC in dem sich 8 Darlington-Treiber befinden. Jeder Treiber hat eine Freilaufdiode, um induktive Lasten (Weichenspulenantriebe) zu schalten. Durch die Parallelschaltung von zwei ULN können Ströme bis 1A geschaltet werden.

Die Funktion des ULN-Decoders ist identisch mit der FET-Variante. Alle weiteren Informationen zu Spannungsversorgung etc. kann dort nachgelesen werden.

8.2.3 4-fach Servo-Decoder



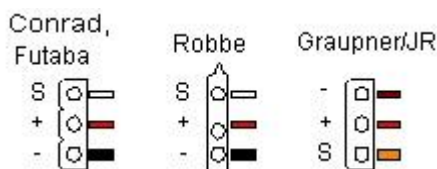
4-fach Servo-Decoder, Platinen-Lay-out

Neben dem (Schalt-)Decoder existiert ein Servo-Decoder, der vier Servos ansteuern kann. Die Servos werden direkt, also ohne weitere Elektronik, an den Servo-Decoder angeschlossen.

Servos werden im Funkfernsteuer-Modellbau verwendet und können auf der Modellbahn dort eingesetzt werden, wo Stellbewegungen ausgeführt werden sollen (Weichenantrieb, Bahnschranken, Formsignale, Lokschiuppentore, winkende Figuren, etc.).

Servos werden mit einer stabilisierten Spannung von +5V versorgt, diese Spannung liegt dauerhaft an. Über eine weitere Ader wird dem Servo ein Impuls zugeführt, dessen Impulslänge den Drehwinkel bestimmt. Der Impuls muss ca. alle 20ms wiederholt werden, bis der Servo seine Endlage erreicht hat. Danach ist der Impuls abzuschalten. Die Impulslänge und –polarität ist vom verwendeten Servo abhängig. Üblich sind positive Impulse mit Werten zwischen 0,8ms (Linksanschlag) und 2,2ms (Rechtsanschlag). Bei 1,5ms wäre der Servo in Mittelstellung. Die Impulslängen für den Ausschlag werden für jeden Servo einzeln mit dem MoBaSbS-Konfigurationstool festgelegt, man braucht keine externen Schalter/Taster zur Endabschaltung. Da es bezüglich der Spannungszuführung verschiedene Arten von Servos gibt, muss auf eine korrekte Belegung geachtet werden. Üblich ist es, für die Masse braun oder schwarz zu verwenden. Die +5V-Leitung ist im Allgemeinen rot und die Signalleitung hat eine helle Farbe (weiß, gelb, orange).

Beispiele:



Aufbauhinweise:

- Bei industriell gefertigten Platinen ist zu beachten, dass die Bauteilbezeichnungen teilweise falsch sind. Als Referenz ist deshalb immer die Bauteilbezeichnung der Platinen-Lay-outs aus dem geschützten MoBaSbS-Bereich zu verwenden.

- Die beiden LEDs auf der linken Seite der Platine brauchen nicht bestückt zu werden. Sie dienen bei der Inbetriebnahme Diagnosezwecken. So 'blitzt' die P-Bus-LED einmal kurz beim Scannen der Peripheriemodule durch den PMC oder bei einigen Konfigurationskommandos auf. Mit der roten Power-LED kann schnell überprüft werden, ob die Versorgungsspannung am Decoder anliegt.

Hardware

Spannungsversorgung der Verbraucher (hier der Servos):

Wie auf allen anderen Peripheriemodulen auch, kann die Spannungsversorgung als DC oder AC Versorgung erfolgen. Wird eine DC-Versorgung gewählt, so kann der Gleichrichter an den gekennzeichneten Stellen gebrückt werden.

Zu beachten ist, dass der Servo beim Einschalten der Versorgungsspannung kurz zucken kann, wenn noch kein Eingangssignal anliegt. Erst wenn das Eingangssignal kommt, rückt sich der Servo wieder zurecht.

Abhängig vom eingesetzten Servo kann es zu einem Zucken bis zum wilden hin- und herfahren des Servos beim Einschalten kommen. Die Ursachen sind zum Einen der jeweilige Servo-Typ und zum Anderen die Versorgungsspannung. Eine Abhilfe kann darin bestehen, die Versorgungsspannung für die Servos langsam hochzufahren oder die Versorgungsspannung der Servos erst einige Sekunden nach dem Einschalten der MoBaSbS einzuschalten.

Weiterhin kann durch Tausch des Netzteils das Zucken verhindert werden (z. B. anderes Schaltnetzteil verwenden oder ein 5V-PC-Netzteil einsetzen).

Sollten die Servos zucken, wenn eine Lok in der Nähe fährt, muss auf strikte Trennung von Busleitung und Servokabel geachtet werden. Weiterhin kann das Servokabel verlängert werden, da es möglich ist, dass das Servokabel wie eine Antenne auf das Gleissignal wirkt. Durch die Verlängerung wird die Antenne verstimmt.

Ein weiterer Punkt ist der Gleichrichter auf dem PM_SDEC. Da der Servo lediglich 5V benötigt kann die Eingangsspannung am Gleichrichter auf 7-9V reduziert werden, was die Verlustleistung deutlich reduziert.

Spannungsversorgung des μ Cs:

Der μ C wird über das Datenkabel mit Spannung versorgt. Diese wird an der Backplane angeschlossen und sollte ca. 9- 12 V DC betragen. Sie wird auf der Platine auf 5V stabilisiert. Bei sehr vielen Peripheriemodulen oder sehr langen Verbindungskabeln, kann der Spannungsabfall über die Versorgungsleitung zu groß werden. In diesem Fall kann eine DC-Spannung von 12V auf jedem Peripheriemodul neu zugeführt werden. Dafür muss eine der kurzen Drahtbrücken an der RJ45-Buchse entfernt werden, an der das Datenkabel vom vorherigen Modul ankommt. Über die andere Drahtbrücke an der 2ten RJ45-Buchse wird dann die Versorgungsspannung 'frisch' weitergegeben.

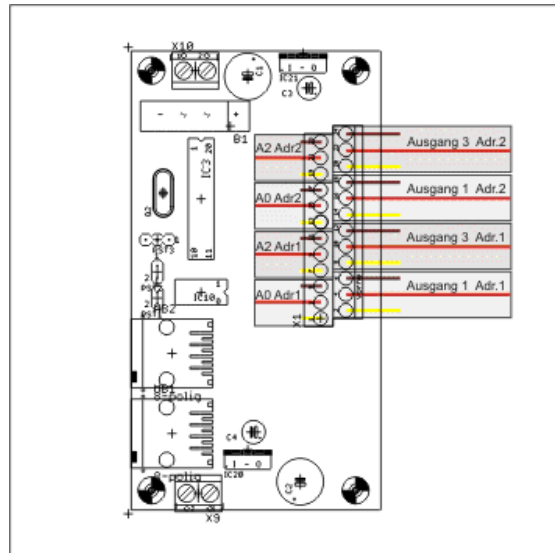
Software

Die Software des Servo-Decoders enthält eine Rückmeldung. Der Servo-Decoder sendet, sobald er die Endstellung erreicht hat, seine neue Endlage an die Steuersoftware weiter. Er verhält sich in dem Moment so, als hätte man einen Encoder auf die Adresse des Servo-Decoders eingestellt, d. h., dass für die Rückmeldung die gleiche Adresse verwendet werden muss, wie für die Ansteuerung.

Damit ist es möglich, erst dann Aktionen von der Steuersoftware auslösen zu lassen, wenn der Servo wirklich in der Endstellung eingelaufen ist (z. B. nach dem Öffnen des Schuppentores die Lok aus dem Lokschuppen automatisch herausfahren lassen).

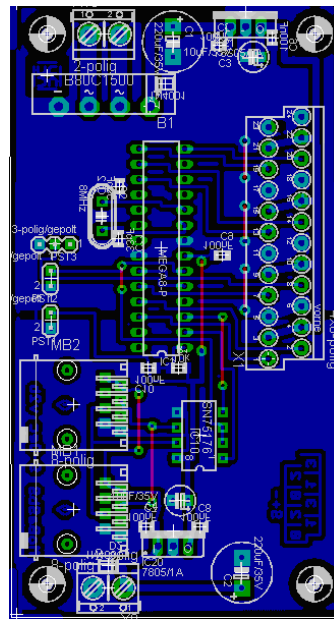
In Traincontroller bis einschl. Version 5.5 können hierfür nur die Adressen 0 bis 127 verwendet werden. Man kann also z. B. einen Servo auf Adresse 505 (in Traincontroller) ansteuern und gleichzeitig einen Melder auf Adresse 127-1(in Traincontroller) einrichten. Ab Traincontroller V 5.8 wird die MoBaSbS direkt unterstützt. Es gibt keine Adresseinschränkungen mehr.

Die Platine des 4-fach Servo-Decoders ist für den Anschluss von 8 Servos vorbereitet. Da aber in den ATtiny2313 nur der Code für 4 Servos hineinpasste, können nicht alle Anschlüsse genutzt werden. Um aber dennoch 8 Servos anschließen zu können, gibt es eine Adapterplatine, welche an Stelle des ATtiny2313 in den Sockel gesteckt wird. Damit ist der ATmega8 nutzbar und es kann dann die Software des 8-fach Servo-Decoders genutzt werden. Es sind dann die Hinweise zur Inbetriebnahme des 8-fach Servo-Decoders in Kap. 8.2.4 zu beachten.



Anschlussbelegung mit Adapter-Platine

8.2.4 8-fach Servo-Decoder



Servo-Decoder (8-fach mit ATmega8), Platinen-Layout

Von Uwe Mäusezahl wurde die Entwicklung des 8-fach Servo-Decoders vorangetrieben.

Mit dem 8-fachen Servo-Decoder ist es möglich bis zu 8 Servos mit einem Modul anzusteuern. Dies ist für alle interessant, die eine größere Anzahl von Servoantrieben innerhalb einer kleinen Fläche betreiben z. B. bei größeren Weichenstraßen. Das spart Verkabelungsaufwand und Kosten.

Da der Programmspeicher des ATtiny2313 für die Ansteuerung und Justage von 8 Servos nicht mehr ausreichend ist, wird bei diesem Modul der ATmega8 verwendet.

Aufbauhinweise:

- Die beiden LEDs auf der linken Seite der Platine brauchen nicht bestückt zu werden. Sie dienen bei der Inbetriebnahme Diagnosezwecken. So 'blitzt' die P-Bus-LED einmal kurz beim Scannen der Peripheriemodule durch den PMC oder bei einigen Konfigurationskommandos auf. Mit der roten Power-LED kann schnell überprüft werden, ob die Versorgungsspannung am Decoder anliegt.

Hardware

Spannungsversorgung der Verbraucher (hier der Servos):

Wie auf allen anderen Peripheriemodulen auch, kann die Spannungsversorgung als DC oder AC Versorgung erfolgen. Wird eine DC-Versorgung gewählt, so kann der Gleichrichter an den gekennzeichneten Stellen gebrückt werden. Bei einer AC-Versorgung kann die Eingangsspannung am Gleichrichter auf 7-9V reduziert werden, da der Servo lediglich 5V benötigt. Dadurch wird die Verlustleistung deutlich reduziert.

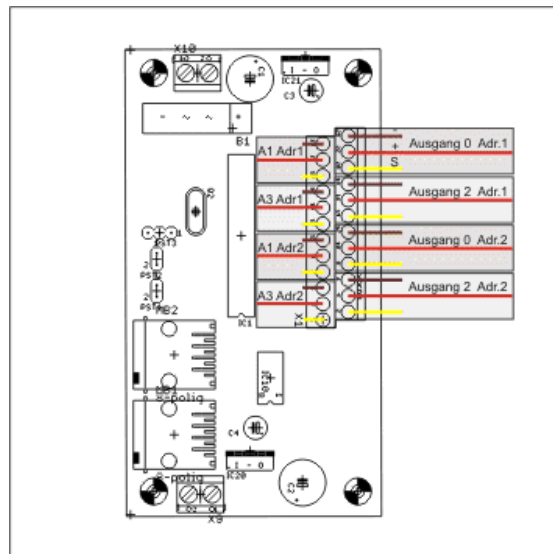
Für Abhilfe des Servo-Zuckens beim Einschalten stehen in Kap. 8.2.3 einige Tipps.

Aufgrund des Einschaltzuckens wurde in der Software für den 8-fach-Servo-Decoder als nicht ganz optimale Lösung eine Speicherung jedes Schaltbefehles ins Eeprom integriert. Damit wird sichergestellt, dass beim Einschalten des Servos sofort die letzte Stellung (die vor dem Abschalten) angesteuert wird und nur noch ein kurzes Zucken beim Einschalten vorhanden ist.

Spannungsversorgung des μ Cs:

Der μ C wird über das Datenkabel mit Spannung versorgt. Diese wird an der Backplane angeschlossen und sollte ca. 9- 12 V DC betragen. Sie wird auf der Platine auf 5V stabilisiert. Bei sehr vielen Peripheriemodulen oder sehr langen Verbindungskabeln, kann der Spannungsabfall über die Versorgungsleitung zu groß werden. In diesem Fall kann eine DC-Spannung von 12V auf jedem Peripheriemodul neu zugeführt werden. Dafür muss eine der kurzen Drahtbrücken an der RJ45-Buchse entfernt werden, an der das Datenkabel vom vorherigen Modul ankommt. Über die andere Drahtbrücke an der 2ten RJ45-Buchse wird dann die Versorgungsspannung 'frisch' weitergegeben.

Bei der Platine mit dem ATmega8 als Prozessor, wird eine andere Anschlussbelegung verwendet, als beim bisherigen Servo-Decoder.



Anschlussbelegung mit ATmega8 (ohne Adapterplatine)

Inbetriebnahme

- Für die Programmierung des ATmega8 müssen die Hinweise zu den Fusebits in Abschnitt 10.2.2 beachtet werden.
- Der 8-fachServodecoder wird im laufenden Betrieb der MoBaSbS als zwei 4-fach Servo-Decoder erkannt und auch so angesprochen.
- Durch die Erhöhung auf 8 Servos ist dem Spannungsregler für die Stromversorgung der Servos besondere Aufmerksamkeit zu schenken. Dabei kann es, je nach Eingangsspannung, Größe der Servos usw. notwendig werden, einen Kühlkörper zu verwenden.
- Das Modul ist nach der Programmierung auf die Peripheriemoduladresse 255 eingestellt. In diesem Zustand wird es vom PMC auch als nur eine Adresse erkannt. Ändert man nun die PM Adresse, erkennt das Konfigurationstool nach einem Rescan, zwei 4-fach Servo-Decoder, wobei einer die neue Adresse und der andere noch die Defaultadresse 255 hat. Diese Defaultadresse muss jetzt auch noch geändert werden. Dabei kann die Adresse frei gewählt werden. Es ist übrigens nicht notwendig nach der Programmierung der ersten Adresse einen Rescan durchzuführen. Es reicht vollkommen aus, dies nach dem zweiten Adresswechsel zu erledigen. Möchte man später die Moduladressen erneut ändern, so geschieht dies genauso, wie mit den anderen Peripheriemodulen. Um später noch zu wissen, welche Adresse als erste von 255 wegprogrammiert wurde, steht in der Servomaske des Konfigurationstools unter PM-Name am Ende Adr. 1 oder Adr. 2. Dies ist wichtig für die Anschlüsse der Servos.

8.2.5 LDEC-Decoder

Der Lichtsignaldecoder (LDEC) wurde von Uwe Mäusezahl entwickelt, um unterschiedliche Signalbilder vorbildgerecht umzuschalten.

Es gibt ihn in zwei Ausführungen:

- Lichtsignal-Decoder-Platine mit ATmega8
- Lichtsignal-Decoder unter Verwendung der FET- oder ULN-Decoder mit ATtiny2313

Für beide Versionen wurden unterschiedliche SW-Varianten entwickelt. Eine Übersicht der SW / HW-Kombinationen ist in Kap. 8.1 aufgelistet.

Beide Versionen werden nicht mehr gepflegt, Ersatz ist der PM-Uni-Decoder (siehe Kap. 8.2.6).

Nachstehend sind die alte Platinen-Version und die unterschiedlichen SW-Versionen beschrieben.

Lichtsignal-Decoder-Platine mit ATmega8

Mit dieser Version kann der Anwender seine Signalbilder selbst festlegen. Somit ist fast jedes Signalsystem darstellbar.

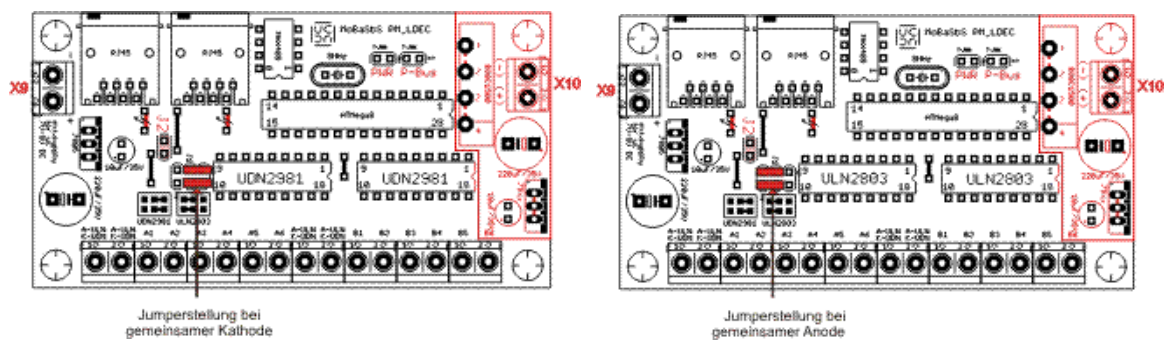
Funktionen

- Decoder für Signale mit gemeinsamer Anode oder Kathode verwendbar
- 12 Ausgänge frei konfigurierbar
- einstellbare Zeit für das Abdimmen und Aufdimmen
- einstellbare Zeit zwischen den Dimmvorgängen (Dunkelzeit)
- einstellbare Blinkfrequenz
- Helligkeit jeder LED veränderbar
- interne oder externe Spannungsversorgung der Signale über Jumper wählbar
- bis zu 8 Signalbilder einstellbar mit einem Modul
- Spezialmodus mit bis zu 16 Signalbildern pro Modul wählbar

Aufbauhinweise:

- Bei industriell gefertigten Platinen ist zu beachten, dass die Bauteilbezeichnungen teilweise falsch sind. Als Referenz ist deshalb immer die Bauteilbezeichnung der Platinen-Layouts aus dem geschützten MoBaSbS-Bereich zu verwenden.
- Die beiden LEDs PWR und P-Bus brauchen nicht bestückt zu werden. Sie dienen bei der Inbetriebnahme Diagnosezwecken. So 'blitzt' die P-Bus-LED einmal kurz beim Scannen der Peripheriemodule durch den PMC oder bei einigen Konfigurationskommandos auf. Mit der roten Power-LED kann schnell überprüft werden, ob die Versorgungsspannung am Decoder anliegt.
- Die Vorwiderstände für die LEDs R2 - R13 sind für 5V Spannung und die Verwendung von low-current LEDs berechnet. Sie können bei Bedarf angepasst werden.

Einstellmöglichkeiten



LDEC-Decoder (mit ATmega8), Platinen-Layout

Hardware

Spannungsversorgung der Verbraucher (hier der LEDs):

Die LEDs der Lichtsignale erhalten ihre Spannung über den Anschluss X10 auf der rechten Platinenhälfte.

Möchte man trotzdem mit den internen 5V des LDEC die Signale betreiben, so können die rot eingezeichneten Bauteile auf der Platine weggelassen werden und der Jumper J2 muss gesetzt werden. Ist dieser Jumper gesteckt darf keine Spannung am Eingang X10 angelegt werden.

Spannungsversorgung des μ Cs:

Der μ C wird über das Datenkabel mit Spannung versorgt. Die Spannung wird an der Backplane angeschlossen und sollte ca. 9- 12 V DC betragen. Sie wird auf der Platine auf 5V stabilisiert. Bei sehr vielen Peripheriemodulen oder sehr langen Verbindungskabeln, kann der Spannungsabfall über die Versorgungsleitung zu groß werden. In diesem Fall kann am Eingang X9 eine DC-Spannung von 12V auf dem Peripheriemodul neu zugeführt werden. Dafür dürfen die beiden Drahtbrücken, welche mit dem Blitzpfeil gekennzeichnet sind, nicht bestückt werden.

Es ist möglich das PM so zu konfigurieren, dass sowohl Lichtsignale mit gemeinsamer Kathode (Minuspol) oder gemeinsamer Anode (Pluspol) angeschlossen werden können

Gemeinsame Kathode

Hierzu muss für die Treiberbausteine IC 1 und IC 2 jeweils ein UDN 2981 eingesetzt werden und die Jumper wie in obigem Bild gesetzt werden.

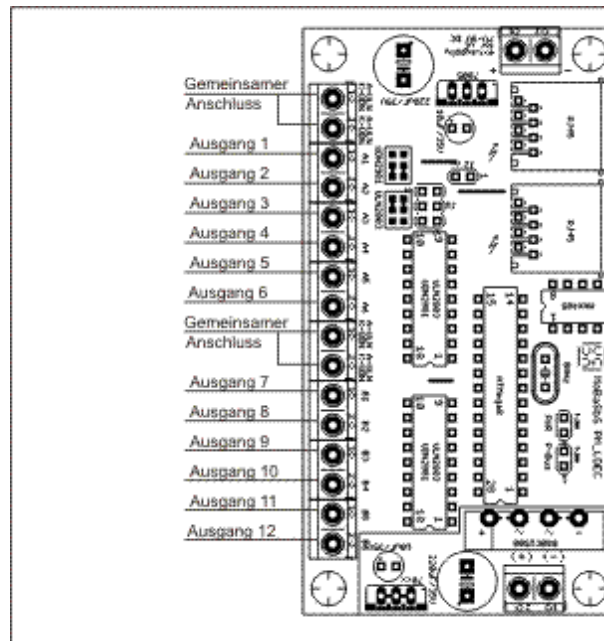
Alternativ können diese IC auch weggelassen werden und die Anschlüsse 1-6 mit den jeweils gegenüberliegenden Anschlüssen verbunden werden (Drahtbrücke). Die Lichtsignale werden dann direkt vom Prozessor angesteuert, so dass keine weitere Spannungsversorgung notwendig ist.

Ist die Leuchtkraft der LED zu niedrig und müssen deshalb die Vorwiderstände verkleinert werden (wenn keine low current LED, dann 220 Ohm Vorwiderstand), so ist diese Variante nicht empfehlenswert.

Gemeinsame Anode

Hier werden die IC 1 und IC 2 mit dem Treiberbaustein ULN 2803 bestückt und die Jumper wie in obigem Bild gesetzt.

Anschlüsse



LDEC mit ATmega8, Anschlüsse

Software PM_LDEC8

Die Software kann auf unterschiedlichen LDEC-Versionen eingesetzt werden (siehe Kap. 8.1).

Im Folgenden werden einige Beispiele für unterschiedliche Signalsysteme und Signalkombinationen beschrieben. Diese Beispieldateien können in der MoBaSbS-Konfigurationssoftware / Maske *Lichtsignal-Decoder Universal* über das Button "Laden" geladen werden.

Sie können noch angepasst oder direkt über das Button "SRAM oder Eeprom speichern" in den LDEC übertragen werden. Speichert man die Werte nur im SRAM so gehen sie nach dem Ausschalten des LDEC wieder verloren.

Die Symbole in den Einstelltabellen sind verschiedenen Steuerungssystemen nachempfunden (Pfeil, Plus-Minus, Rot-Grün).

Die Anschlussbelegungen sind an die Steuerungssoftware Traincontroller™ angelehnt. Es dürfte aber auch kein Problem darstellen, damit in anderen Steuerungsprogrammen die Anschlussbelegung zu erstellen.

Im Forum der MoBaSbS befindet sich ein Ordner ‚Daten für LDEC‘, in den jeder seine Daten mit einer kurzen Beschreibung ablegen und natürlich auch Daten von anderen Anwendern nutzen kann.

Beispiel: DB - Vierbegriffiges Hauptsignal mit Vorsignal am Mast

In diesem Beispiel wird die integrierte Dunkelschaltung bei den Signalen HP 0 und Sh 1 angewendet. Beide Signalbilder bekommen das Häkchen bei Dunkel. Damit wird das Vorsignal, wenn diese Signalbilder geschaltet werden, dunkel geschaltet. Ohne die Häkchen bei Dunkel können beide Signale auch separat verwendet werden.

In Traincontroller™ sind dazu zwei Signale zu verwenden und bei Anzahl der Kontakte 3/4 auszuwählen. In dieser Beispieldatei ist dann das erste Signal, das Hauptsignal und das zweite Signal das Vorsignal.

Sign:		HS
HP 0		Grün A 4
HP 1		Rot1 Rot2 A 1 + 2
HP 2		Weiß A 5
Sh 1		Gelb A 3
Vr 0		Grün2 A 10
Vr 1		Gelb2 A 8
Vr 2		Grün1 A 9
		VS
		Grün2 A 10
		Gelb2 A 8
		Grün1 A 9
		Gelb1 A 7
Dunkl		

Vierbegriffiges Hauptsignal mit Vorsignal am MastBeispiel: DR - HL System Komplettansteuerung aller Signalbegriffe

In diesem Beispiel kommt der 16 Signalmodus zur Anwendung. Das bedeutet, dass immer zwei Schaltbefehle gesendet werden müssen, um ein neues Signalbild aufzudimmen. Wie unten gut zu erkennen muss ein Schaltbefehl an die Adresse 1 oder 2 und der zweite Schaltbefehl an Adresse 3 oder 4 gesendet werden. Die Zeitspanne darf dabei 3 Sekunden nicht überschreiten, sonst wird der erste Befehl wieder verworfen.

Meistens werden nicht alle Signalbegriffe benötigt, so dass in einem nachfolgenden Beispiel die Ansteuerung zweier Signale des HL-Systems beschrieben wird. Wie schon erwähnt, sind dies nur Beispiele und jeder kann sich die Signale selber an seine Bedürfnisse anpassen.

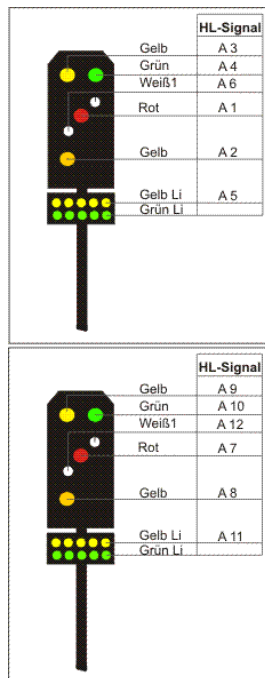
	HL-Signal
	Gelb A 7
	Grün A 8
	Weiß1 A 5
	Rot A 1
	Weiß2 A 11
	Gelb A 2
	Gelb Li A 4
	Grün Li A 3

Signalbild	Schaltbefehl an Adresse 1	Schaltbefehl an Adresse 2	Schaltbefehl an Adresse 3	Schaltbefehl an Adresse 4
HL 1	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input checked="" type="radio"/>	<input type="radio"/> <input type="radio"/>	<input checked="" type="radio"/> <input type="radio"/>
HL 2	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input checked="" type="radio"/>	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input checked="" type="radio"/>
HL 3a+b	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input checked="" type="radio"/>	<input checked="" type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>
HL 4	<input type="radio"/> <input type="radio"/>	<input checked="" type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>	<input checked="" type="radio"/> <input type="radio"/>
HL 5	<input type="radio"/> <input type="radio"/>	<input checked="" type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input checked="" type="radio"/>
HL 6a+b	<input type="radio"/> <input type="radio"/>	<input checked="" type="radio"/> <input type="radio"/>	<input checked="" type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>
HL 7	<input checked="" type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>	<input checked="" type="radio"/> <input type="radio"/>
HL 8	<input checked="" type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input checked="" type="radio"/>
HL 9a+b	<input checked="" type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>	<input checked="" type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>
HL 10	<input type="radio"/> <input checked="" type="radio"/>	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>	<input checked="" type="radio"/> <input type="radio"/>
HL 11	<input type="radio"/> <input checked="" type="radio"/>	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input checked="" type="radio"/>
HL 12a+b	<input type="radio"/> <input checked="" type="radio"/>	<input type="radio"/> <input type="radio"/>	<input checked="" type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>
HL 13	<input checked="" type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input checked="" type="radio"/>	<input type="radio"/> <input type="radio"/>
Ra 12	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input checked="" type="radio"/>	<input type="radio"/> <input checked="" type="radio"/>	<input type="radio"/> <input type="radio"/>
Zs 1	<input type="radio"/> <input type="radio"/>	<input checked="" type="radio"/> <input type="radio"/>	<input type="radio"/> <input checked="" type="radio"/>	<input type="radio"/> <input type="radio"/>

HL System Komplettansteuerung aller Signalbegriffe

Beispiel: DR - HL System Ansteuerung von zwei Hauptsignalen

Hier werden zwei HL Signale im 16-Signalmodus angesteuert.



Signalbild	Schaltbefehl an Adresse 1	Schaltbefehl an Adresse 2	Schaltbefehl an Adresse 3	Schaltbefehl an Adresse 4
HL 1	←	○ ○	←	○ ○
HL 3a+b	○ →+	○ ○	←	○ ○
HL 7	○ ○	←	←	○ ○
HL 9a+b	○ ○	○ →+	←	○ ○
HL 10	←	○ ○	○ →+	○ ○
HL 12a+b	○ →+	○ ○	○ →+	○ ○
HL 13	○ ○	←	○ →+	○ ○
Ra 12	○ ○	○ →+	○ →+	○ ○
HL 1	←	○ ○	○ ○	←
HL 3a+b	○ →+	○ ○	○ ○	←
HL 7	○ ○	←	○ ○	←
HL 9a+b	○ ○	○ →+	○ ○	←
HL 10	←	○ ○	○ ○	○ →+
HL 12a+b	○ →+	○ ○	○ ○	○ →+
HL 13	○ ○	←	○ ○	○ →+
Ra 12	○ ○	○ →+	○ ○	○ →+

HL System Ansteuerung von zwei Hauptsignalen**Software LDEC-uni-T**

Die Software kann auf unterschiedlichen LDEC-Versionen eingesetzt werden (siehe Kap.8.1).

Die LDEC-T-Variante ermöglicht es, einen Taktgenerator einzuschalten. In diesem Modus arbeitet der LDEC eigenständig und er braucht keine Schaltbefehle von außerhalb, um das Signalbild zu wechseln. Damit können automatisch gesteuerte Häuserbeleuchtungen, Verkehrsampeln oder auch einfache Ablaufsteuerungen realisiert werden.

Weitere Informationen siehe Kap. 12.6.7

Inbetriebnahme

Für die Programmierung des ATmega8 müssen die Hinweise zu den Fusebits in Abschnitt 10.2.2 beachtet werden.

Der Lichtsignaldecoder ist auf die Moduladresse 255 voreingestellt. Diese kann, wie bei den anderen Modulen auch, auf die gewünschte Adresse über das Konfigtool oder den HDC umprogrammiert werden.

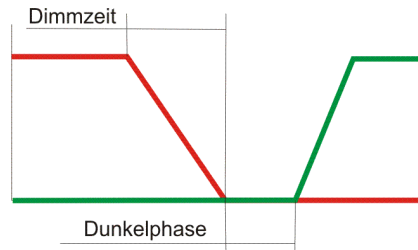
Lichtsignal-Decoder unter Verwendung der FET- oder ULN-Decoder mit ATtiny2313

Für den ATtiny2313 sind folgende Modi entwickelt worden:

- Modus 0 – vier zweibegriffige Signale
- Modus 1 – zwei unabhängige bis zu vierbegriffige Signale
- Modus 2 – ein Hauptsignal mit Vorsignal an einem Mast mit Dunkelschaltung

Alle Signalbilder können durch eine einstellbare Zeit abgedimmt und nach einer -ebenfalls einstellbaren- Dunkelphase wieder aufgedimmt werden. Alle Einstellungen lassen sich mit dem MoBaSbS-Konfigurationstool durchführen.

Aus Speichermangel im ATtiny2313 werden die weißen LED (SH1) nicht auf- bzw. abgedimmt, sondern nur ein- und ausgeschaltet.



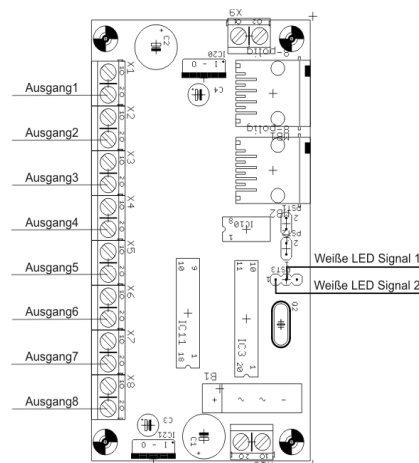
LDEC, Dimmphasen

Ein weiteres Feature ist die einstellbare Helligkeit jeder einzelnen LED des Signals durch das Konfigurationstool. Damit braucht nicht mehr mit Vorwiderständen herumexperimentiert zu werden.

Beim Einsatz der Decoder ist darauf zu achten, dass keine Vorwiderstände auf der Platine vorhanden sind und somit die LED nicht direkt angeschlossen werden dürfen.

Da die FET- und ULN-Decoder nach Masse schalten, können über diesen Baustein nur Lichtsignale mit gemeinsamer Anode angeschlossen werden. Möchte man auch Lichtsignale mit gemeinsamer Kathode anschließen, so ist der Baustein ULN 2803 zu entfernen. Dort sind dann die Anschlüsse 1-8 entweder mit Drahtbrücken mit den gegenüberliegenden Anschlüssen zu überbrücken oder durch passende Vorwiderstände zu ersetzen.

Bei Verwendung der FET- bzw. ULN-Decoder gilt folgendes Anschlussbild. Die Ausgänge für die weißen LEDs für jedes Signal müssen von dem Pfostenstecker PST 3 abgegriffen werden, wie im Bild unten zu sehen ist. Dies funktioniert aber nur bei Signalen mit gemeinsamer Kathode. Bei gemeinsamer Anode können mit einem Transistor (z. B. der Typ, der auch für FET-Decoder genutzt wird), der gegen Masse schaltet, die weißen LEDs angesteuert werden.



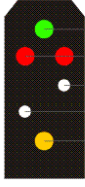
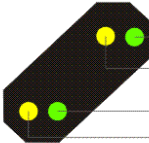
FET- oder ULN-Decoder als LDEC mit ATtiny2313, Anschlüsse

Modus 0

In diesem Modus können vier Signale angeschlossen werden. Er ist vorgesehen für zweibegriffige Signale, das heißt für alle Signale die nur zwischen zwei möglichen Signalbegriffen wechseln können. (z.B. Blocksignale, einfache Vorsignale, Sperrsignale usw.) Jedes Signal wird über eine Adresse angesteuert, so dass immer nur zwischen zwei LED hin- und hergeschaltet werden kann. Dabei schalten Befehle an die Adresse 0 die Ausgänge 1 und 2 um, Adresse 1 die Ausgänge 3 und 4 usw. Die Ausgänge 9 und 10 sind in diesem Modus wirkungslos.

















Modus 1

In diesem Modus ist es möglich, zwei bis zu vierbegriffige Signale anzuschließen. Diese Signale arbeiten unabhängig voneinander und belegen immer zwei zusammenhängende Kontaktpaare. Um ein hohe Flexibilität zu erreichen, kann jedes dieser zwei Signale als Vor- oder Hauptsignal betrieben werden. Dies wird über das MoBaSbS-Konfigurationstool eingestellt und im Eeprom abgespeichert.

		Signal1 als Hauptsignal	Signal2 als Hauptsignal
	Grün	Ausgang 4	Ausgang 8
	Rot1 Rot2	Ausgang 1 + 2	Ausgang 5 + 6
	Weiß	Ausgang 9	Ausgang 10
	Gelb	Ausgang 3	Ausgang 7
		Signal1 als Vorsignal	Signal2 als Vorsignal
	Grün2	Ausgang 4	Ausgang 8
	Gelb2	Ausgang 2	Ausgang 6
	Grün1	Ausgang 3	Ausgang 7
	Gelb1	Ausgang 1	Ausgang 5

Anschlussbelegung für Modus 1

In der folgenden Tabelle zeigen die Pfeile die Richtungstasten auf dem HDC, welche für die Auslösung des Schaltbefehles gedrückt werden müssen. Die Adressbezeichnung ist identisch mit der Zahl hinter der PM Adresse im Perimenu des HDC. Die anderen Symbole sind mit den Anschlussbelegungen im Steuerungsprogramm Traincontroller™ identisch.

















Signalbild	Schaltbefehl an Adresse 1 oder 3	Schaltbefehl an Adresse 2 oder 4
HP 0 	<input type="radio"/> 	<input type="radio"/> <input type="radio"/>
HP 1 	<input type="radio"/> <input type="radio"/>	
Hp 2 		<input type="radio"/> <input type="radio"/>
Sh 1 	<input type="radio"/> <input type="radio"/>	<input type="radio"/> 
Vr 0 	<input type="radio"/> 	<input type="radio"/> <input type="radio"/>
Vr 1 	<input type="radio"/> <input type="radio"/>	
Vr 2 		<input type="radio"/> <input type="radio"/>
Dunkel 	<input type="radio"/> <input type="radio"/>	<input type="radio"/> 

Schaltbefehle, Modus 1

Nach dem obigen Schema kann sich jetzt jeder ganz leicht die passende Anschlussbelegung herausuchen, egal ob das Signal 1 ein Vor- oder Hauptsignal ist. Dasselbe gilt auch für das Signal 2.

Modus 2

Für diesen Modus gilt ebenfalls das im vorigen Abschnitt geschriebene. Der Unterschied besteht darin, dass für das Signal 1 ein Hauptsignal und für das Signal 2 ein Vorsignal fest vorgegeben ist. Dieses Vorsignal ist an die Signalstellung des Hauptsignals gekoppelt. Daraus folgt, dass nur bei Fahrtstellung des Hauptsignals das Vorsignal einen Signalbegriff zeigt (Dunkelschaltung). Dies kommt beim Vorbild überall zur Anwendung, wenn das Signal HP0 (rot) zeigt. Die Ansteuerung ist zwar auch mit dem Modus 1 machbar, muss dann aber separat geschaltet werden, währenddessen dies hier automatisch geschieht.

Signalbild	Schaltbefehl an Adresse 1	Schaltbefehl an Adresse 2	Schaltbefehl an Adresse 3	Schaltbefehl an Adresse 4
HP 0 	<input type="radio"/> 	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>
HP 1 	<input type="radio"/> <input type="radio"/>	<input type="radio"/> 	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>
HP 2 	 <input type="radio"/>	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>
Sh 1 	<input type="radio"/> <input type="radio"/>	<input type="radio"/> 	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>
Vr 0 	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>	<input type="radio"/> 	<input type="radio"/> <input type="radio"/>
Vr 1 	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>	 <input type="radio"/>
Vr 2 	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>	 <input type="radio"/>	<input type="radio"/> <input type="radio"/>
Dunkel 	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>	<input type="radio"/> <input type="radio"/>	<input type="radio"/> 

Schaltbefehle, Modus 2

Software PM_LDEC_DB

Diese Variante kann nur in FET- und ULN-Decodern mit ATtiny2313 eingesetzt werden (siehe Kap.8.1). Sie unterstützt die Modus 0 bis 2.

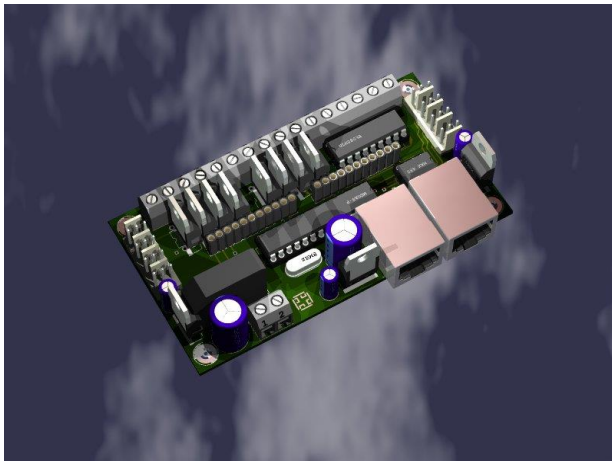
Software PM_LDEC_Light

Diese Variante kann nur in FET- und ULN-Decodern mit ATtiny2313 eingesetzt werden (siehe Kap.8.1). In dieser Variante ist es nur möglich den Modus 0 zu verwenden. Die Besonderheit liegt in der Verwendung der Signale 2 und 4 als Vorsignal zu den Signalen 1 und 3, wenn diese am selben Mast befestigt sind. Dies ist schon im Modus 2 als Dunkelschaltung beschrieben worden. Um auch hier eine hohe Flexibilität zu erreichen, können die Signale 2 und 4 jeweils jedes für sich, separat als Vorsignal eingeschaltet werden. Wählt man diese beiden Signale nicht als Vorsignal, so arbeitet der Lichtsignaldecoder, wie oben im Modus 0 beschrieben.

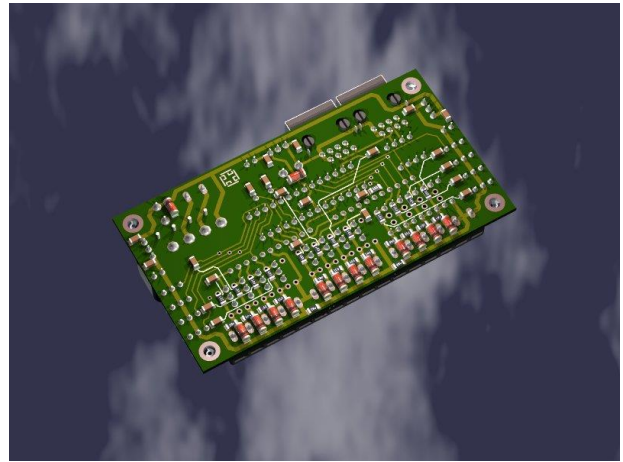
Inbetriebnahme

Der Lichtsignaldecoder ist auf die Moduladresse 255 voreingestellt. Diese kann, wie bei den anderen Modulen auch, auf die gewünschte Adresse über das Konfigtool oder den HDC umprogrammiert werden.

8.2.6 PM-Uni-Decoder



PM-Uni-Decoder (Bestückungsseite)



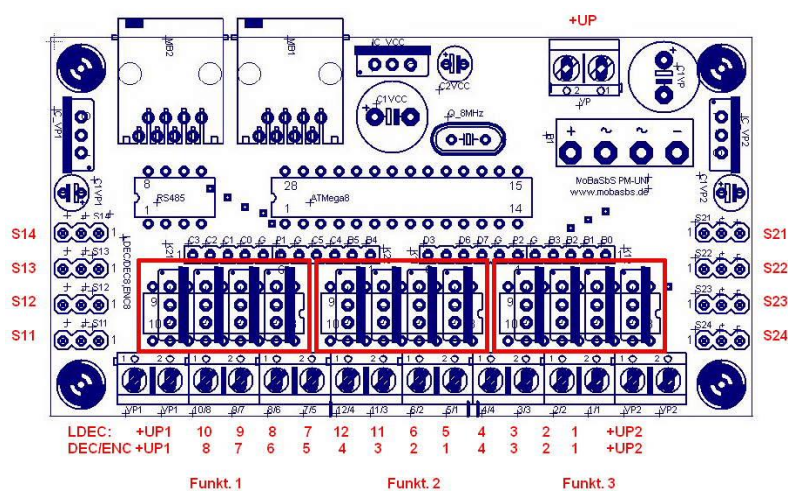
PM-Uni-Decoder (Lötseite)

Über den PM-Uni (PeripherieModul-Universal) werden zu schaltende Verbraucher wie etwa Weichen, Signale oder Lampen bedient, sowie Eingänge abgefragt. Auf dieser Platine können folgende Peripheriemodule aufgebaut werden:

DEC8 ULN und FET, ENC8, SDEC4 und 8, LDEC

Zusätzlich können Erweiterungsplatinen für Stromfühler und Herzstückpolarisation montiert werden. Die Schaltung entspricht den bisherigen Schaltungen der Einzelplatinen, die einzelnen Schaltungen wurden nur zusammengeführt. Beim LDEC gibt es eine Einschränkung: es sind nur LED-Signale mit gemeinsamer Anode anschließbar. Für Glühlampen gibt es keine Einschränkung. Die Platine gibt es in einer einseitigen Version für Selbstätzer sowie in einer industriell gefertigten doppelseitigen Version. Die einseitige Version bedarf guter Ätz- und Lötkenntnisse. Bis auf die Drahtbrücken und der Verwendung von bedrahteten Bauelementen ist die Bestückung quasi identisch.

Aufbauhinweise:



Übersicht der Schaltungsblöcke

Es werden nur die jeweils benötigten Schraubanschlüsse montiert! Nur beim LDEC wird die Vollbestückung benötigt!

Für eine bessere Übersichtlichkeit werden die Schaltungsteile unterhalb der Schraubanschlüsse als Funktionsblöcke 1 bis 3 (siehe Abbildung) bezeichnet.

Brücken

Zuerst zu den Brücken, die nur bei der einseitigen Version benötigt werden: Die Bestückung der Brücken hängt von dem zu erstellenden Peripheriemodul ab. 4 Brücken werden immer benötigt: die 2 horizontalen über den Servoanschlüssen für Masse und die 2 horizontalen unterhalb des ATmega8 und des MAX485 für Spannungsversorgung. So bestückt, können SDEC4 und SDEC8 betrieben werden. Die jeweils 4 vertikalen Brücken in den Funktionsblöcken 1 und 3 werden nur für DEC8 FET benötigt. Die 4 in Funktionsblock 2 vorerst gar nicht. Die vertikale Brücke zwischen den Funktionsblöcken 1 und 2 sowie die horizontale unter Funktionsblock 3 werden nur bei Montage einer Erweiterungsplatine für Herzstückpolarisation gebraucht. Die 3 vertikalen Brücken unter Funktionsblock 1 sowie die 4 unter Funktionsblock 3 werden für DEC8, ENC8 und LDEC benötigt. Die 2 Brücken unter dem Atmega8 müssen nur für den LDEC montiert werden.

Gemeinsames

Alle Versionen benötigen die Eingangsgruppe um den Atmega8 und MAX 485. Es sind die Bauteile im linken oberen Viertel mit Ausnahme der Stromversorgung 2 (VP2).

Stromversorgung

Die allgemeine Stromversorgung des Leistungsteils befindet sich im rechten oberen Viertel. Die Spannung wird an UP eingespeist. Wird eine DC-Spannung verwendet, dann kann der Brückengleichrichter entfallen. C1VP (der Elko rechts neben der Einspeisung) dient der Glättung und kann dann ebenfalls entfallen. Für DEC8, SDEC4 und LDEC reicht die Stromversorgung über die Elemente um VP1 (links oben) aus. Die zwei 0 Ohm Widerstände unterhalb der Schraubanschlüsse werden dann eingebaut (oder durch kleine Drahtbrücken ersetzt). Bei starken Zuckungen der Servos beim Einschalten sollte für SDEC8 auch die Stromversorgung VP2 (oben rechts) bestückt werden. Die 0 Ohm Widerstände fallen weg, der Gleichrichter sollte durch einen 4A Typ ersetzt werden (Reichelt FBU4A). Dies hängt im Wesentlichen von der Art der externen Stromversorgung ab. Vielleicht ist es mittlerweile klar geworden: Die Funktionsgruppen 1, 2 und 3 können von VP1 oder VP2 versorgt werden. Von welcher Versorgung hängt dann von den 0-Ohm Widerständen ab. I.d.R. wird man die Stromversorgung um VP1 aufbauen und damit die Funktionsgruppen 1 und 2 versorgen (DEC8 oder ENC8 oder SDEC4). Dazu wird der 0-Ohm Widerstand zwischen den Funktionsgruppen 1 und 2 bestückt. Für einen LDEC nimmt man die Funktionsgruppe 3 hinzu und bestückt noch den 0-Ohm Widerstand zwischen den Funktionsgruppen 2 und 3. Erst wenn die Stromversorgung zu knapp wird z.B. für einen SDEC8 dessen 2x4 Servos auf den Funktionsgruppen 1 und 3 liegen, wird die Stromversorgung um VP2 mit aufgebaut. VP1 versorgt dann nur noch die Funktionsgruppe 1 (die Servos S11 bis S14) und VP2 die Funktionsgruppe 3 (die Servos S21 bis S24). Die 0-Ohm Widerstände fallen dann weg!

Varianten

- SDEC
Hierfür sind nur noch die 4 dreifach Pfostenfeldstecker links (SDEC4) oder links und rechts (SDEC8) zu bestücken
- DEC8ULN
Auf die Funktionsblöcke 1 und 2 wird jeweils ein ULN 2803 (Kerbe nach rechts) gelötet
- DEC8FET
Hier wird die bedrahtete Version des FET verwendet (IRLU 024N). Die FETs werden mit der Kühlfläche nach rechts in den Funktionsblöcken 1 und 2 eingelötet, beginnend beim oberen Loch der 4er Reihe, das unterste Loch bleibt frei. Zusätzlich müssen noch die Freilaufdioden 1N4001 (SMD) eingebaut werden. Die Lötflächen befinden sich zwischen den Lötpins der Schraubanschlüsse, hier ist Gefühl beim Löten gefragt!

- **LDEC**
Die Funktionsblöcke 1-3 werden mit jeweils einem ULN2803 bestückt, Kerbe nach rechts, geänderte Anschlussbelegung beachten!
- **ENC8**
Die SMD-Päarchen für die Widerstände und Kondensatoren der Eingangs-RC-Glieder sind durch eine Verbindungslinie auf der Platinenrückseite gekennzeichnet. Bestückt werden die Funktionsblöcke 1 und 2.
Sollte für die Verwendung von Hallsensoren o.Ä. eine Stromversorgung dieser Bauteile benötigt werden, kann auf den Positionen der Servoanschlüsse ein oder mehrere Schraubanschlüsse gelötet werden. Die Lochabstände passen.
- **Erweiterungsplatine Herzstückpolarisation**
Hierfür werden auf den in der Mitte befindlichen horizontalen Kontakten die jeweils 6 inneren mit Buchsenleisten (von unten!) bestückt. Dies ist bei einseitiger Platine ein bisschen schwierig. Hier kann dann von unten jeweils eine Platine AD-SDEC4-H angesteckt werden. Die Anschlüsse für Herzstück und Zunge korrespondieren mit den Servoanschlüssen auf der PM-Uni-Platine. Diese Platinen werden mit 8mm Distanzrollen mit der PM-Uni-Platine verschraubt, die Befestigung auf der Platte erfolgt über die Zusatzplatinen. Hier sollte unbedingt die Stromversorgung mit dem 4A Gleichrichter bestückt werden, da die Relais noch zusätzlich Strom verbrauchen.
Von dieser Platine wird es voraussichtliche keine fertige Version geben, da der Nachbau recht einfach ist.
- **Erweiterungsplatine Stromfühler**
Hier werden auf der Kontaktreihe die jeweils 5 äußeren mit Buchsenleisten (von oben!) bestückt. Die Platine AD-ENC8-S wird von oben aufgesetzt und auch mit 8mm Distanzrollen verschraubt. Die Bestückung der RC-Glieder auf der PM-Uni-Platine kann entfallen.
Von dieser Platine wird es voraussichtliche keine fertige Version geben. Sie ist als Vorschlag zu verstehen, da viele Anwender ihre eigenen Stromfühler-Lösungen bevorzugen.

Sonstiges

Alle Verbraucher werden gegen Massepotential geschaltet. Auf der Dekoderplatine sind dafür entsprechende Klemmen vorgesehen. Für die Versorgung der zu schaltenden Verbraucher gibt es nun mehrere Möglichkeiten:

- **Spannungsversorgung der Verbraucher:**
Lässt man die Kondensatoren und den Festspannungsregler entfallen und setzt eine Brücke (Verbindung von Ein und Ausgang des Festspannungsreglers) ein, so steht eine unregelmäßige Gleichspannung zum Schalten zur Verfügung. Eine solche Spannung ist z.B. zum Schalten von Magnetartikeln ausreichend.
Möchte man Verbraucher schalten, die durch unterschiedliche Spannungen versorgt werden, so ist auch dies möglich. Es wird dann pro Schaltausgang nur die Masseklemme verwendet. Das Versorgungspotential des Verbrauchers kommt jetzt von einer externen Quelle. Hier muss jetzt aber die Masse der externen Quelle mit der Masse der Dekoderplatine verbunden werden.
- **Spannungsversorgung des μ Cs:**
Prinzipiell wird der μ C über das Datenkabel mit Spannung versorgt. Diese liegt bei ca. 9V bis 12V DC und wird auf jeder der Platinen auf 5V stabilisiert. Bei sehr vielen Peripheriemodulen oder sehr langen Verbindungskabeln kann der Spannungsabfall über die Versorgungsleitung zu groß werden. In diesem Fall kann eine DC-Spannung von 7V bis 12V auf Verbindungsadaptern zu den Peripheriemodulen neu zugeführt werden (siehe Kap. 9.4).

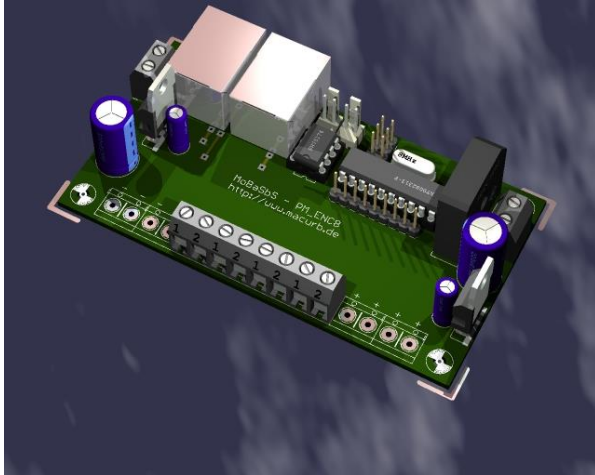
Inbetriebnahme

Die Peripheriemodule werden über den PMC an die MoBaSbS angeschlossen. Wie bei allen anderen Peripheriemodulen auch, ist die μ C-Software per Default so konfiguriert, dass das PM-Modul auf Adresse 255 angesprochen wird. Um zu verhindern, dass ein Adresskonflikt auftritt, wenn ein weiteres neues Peripheriemodul an den PMC angeschlossen wird, muss die Adresse über das MoBaSbS-Konfigurationstool nach dem Anschluss des PMs an den PMC umgesetzt werden. Adressen von 0 bis

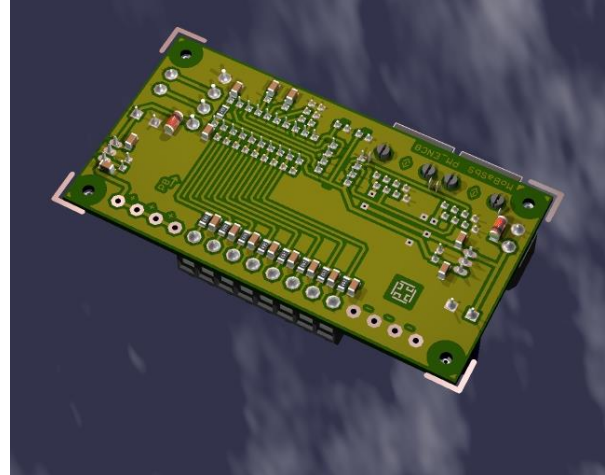
254 können definiert werden. Ebenso wie die PM-Adresse werden auch alle weiteren Parameter des PMs über das Konfigurationstool gesetzt.

Weitere Informationen zur Konfiguration stehen in der Beschreibung von Uwe Mäusezahl im Anhang (siehe Kap. 13.6).

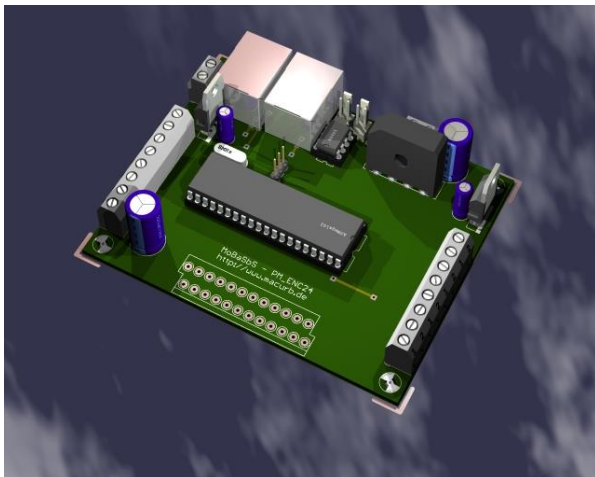
8.3 Encoder



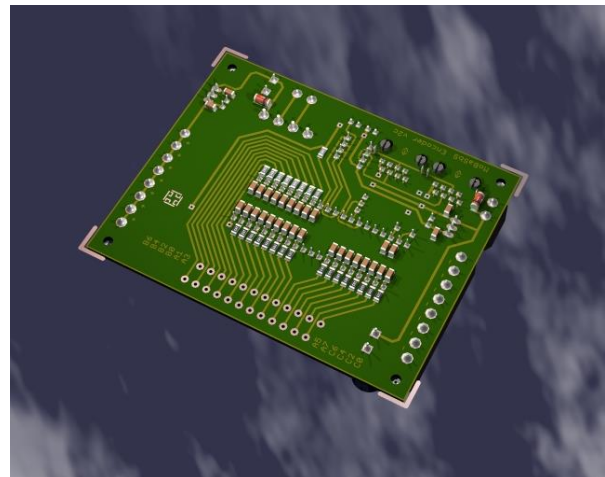
8-fach Encoder (Bestückungsseite)



8-fach Encoder (Lötseite)



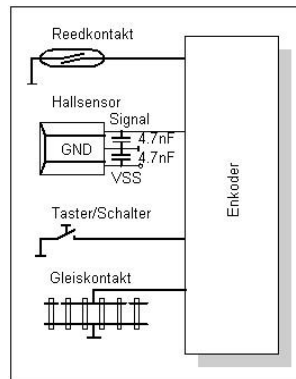
24-fach Encoder (Bestückungsseite)



24-fach Encoder (Lötseite)

Der Encoder dient zur Rückmeldung von Schaltzuständen. Diese Schaltzustände können durch ein Ereignis des fahrenden Zuges (Zug überfährt Kontaktmelder) oder aber auch durch das Betätigen eines Tasters oder Schalters ausgelöst werden (Gleisbildstellpult).

An den Encoder können Schaltelemente angeschlossen werden, die nach Masse hin schalten.



Element-Anschaltung an Encoder

Aufbauhinweise:

- Bei industriell gefertigten Platinen ist zu beachten, dass die Bauteilbezeichnungen teilweise falsch sind. Als Referenz ist deshalb immer die Bauteilbezeichnung der Platinen-Layouts aus dem geschützten MoBaSbS-Bereich zu verwenden.
- Die R-C-Kombination am Eingang jedes Rückmeldekontaktes ist besonders dann wichtig, wenn ein Massekontakt von der Schiene - wie im Märklin-Schienensystem möglich (unterste Möglichkeit in der Abbildung) - genutzt wird. Sie bildet einen Tiefpass 1. Ordnung. Damit werden Frequenzen bis ca. 340Hz durchgelassen. Höhere Frequenzen werden unterdrückt.
- Die Ausgänge sind wie folgt auf die Doppelstock-Klemmen aufgelegt:

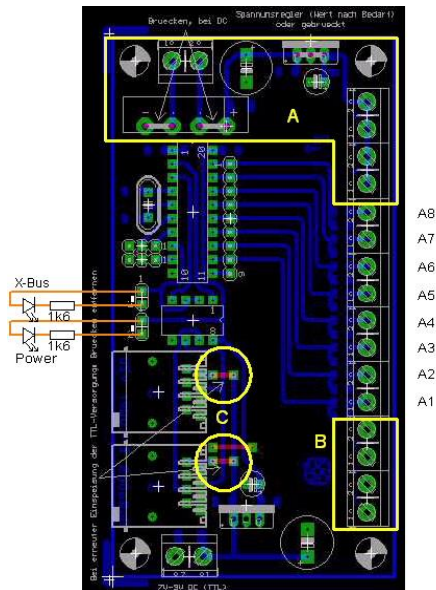
Ausgangsbelegung des 24-fach Encoder mit Doppelstock-Klemmreihe			
Ausgang am IC			Kontakte
Pin 8		X	Kontakt 2/8
Pin 7	X		Kontakt 2/7
Pin 6		X	Kontakt 2/6
Pin 5	X		Kontakt 2/5
Pin 4		X	Kontakt 2/4
Pin 3	X		Kontakt 2/3
Pin 2		X	Kontakt 2/2
Pin 1	X		Kontakt 2/1
Pin 39		X	Kontakt 1/1
Pin 38	X		Kontakt 1/2
Pin 37		X	Kontakt 1/3
Pin 36	X		Kontakt 1/4
Pin 35		X	Kontakt 1/5
Pin 34	X		Kontakt 1/6
Pin 33		X	Kontakt 1/7
Pin 32	X		Kontakt 1/8
Pin 28		X	Kontakt 3/8
Pin 27	X		Kontakt 3/7
Pin 26		X	Kontakt 3/6
Pin 25	X		Kontakt 3/5
Pin 24		X	Kontakt 3/4
Pin 23	X		Kontakt 3/3
Pin 22		X	Kontakt 3/2
Pin 21	X		Kontakt 3/1

Adresse 2

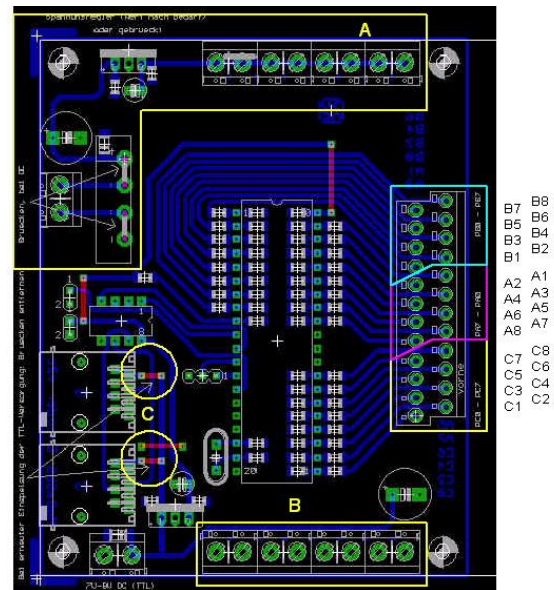
Adresse 1

Adresse 3

Einstellmöglichkeiten



8-fach Encoder, Platinen-Lay-out



24-fach Encoder, Platinen-Lay-out

Hardware

Spannungsversorgung der Verbraucher (im Falle von z.B. Hallsensoren oder Lichtschranken):

Wie auf allen anderen Peripheriemodulen auch, kann die Spannungsversorgung als DC oder AC Versorgung erfolgen. Wird eine DC-Versorgung gewählt, so kann der Gleichrichter an den gekennzeichneten Stellen (innerhalb der mit A bezeichneten gelben Markierungen) gebrückt werden. Gleichrichter, Festspannungsregler und Kondensatoren können entfallen, wenn keine zusätzliche Spannung benötigt wird. Alle Bauteile in den gelben Markierungen A und B brauchen dann nicht bestückt zu werden.

Spannungsversorgung des μ Cs:

Der μ C wird über das Datenkabel mit Spannung versorgt. Diese wird an der Backplane angeschlossen und sollte ca. 9- 12 V DC betragen. Sie wird auf der Platine auf 5V stabilisiert. Bei sehr vielen Peripheriemodulen oder sehr langen Verbindungskabeln, kann der Spannungsabfall über die Versorgungsleitung zu groß werden. In diesem Fall kann eine DC-Spannung von 12V auf jedem Peripheriemodul neu zugeführt werden. Dafür muss eine der kurzen Drahtbrücken an der RJ45-Buchse entfernt werden, an der das Datenkabel vom vorherigen Modul ankommt. Über die andere Drahtbrücke an der 2ten RJ45-Buchse wird dann die Versorgungsspannung 'frisch' weitergegeben.

Inbetriebnahme

- Der 24fach-Encoder verwaltet intern 3 **aufeinander folgende** Adressen. Es braucht aber lediglich seine Startadresse definiert zu werden.
- An den beiden 2poligen Pfostenleisten können zu Testzwecken LEDs aufgesteckt werden. Die obere LED dient zur Überprüfung des X-Busses und muss während des korrekten Betriebs leuchten. Die untere LED zeigt an, ob die Spannungsversorgung korrekt anliegt.

Software

Die Encoder-Codes tragen in Ihrer Bezeichnung ein angehängtes T (z. B. PM_ENC24T).

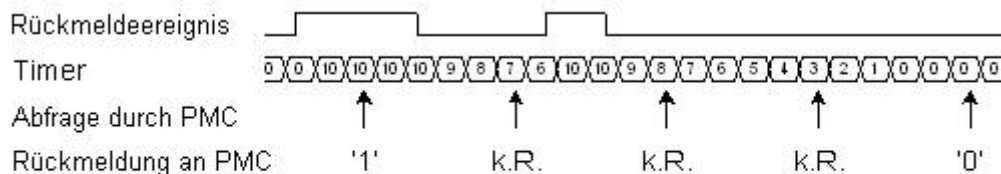
Das T steht für Timer und bedeutet, dass man zu jedem Eingang eine Flackerschutzzeit definieren kann. Der Vorteil dieses Flackerschutzes ist, dass kurze Eingangsimpulse, die dadurch entstehen, dass man einen nicht optimalen Rad-Schiene-Kontakt hat, nicht zu einem 'Rückmeldegewitter' in der angeschlossenen MoBa-Software führen. In Traincontroller kann man solch einen Flackerschutz ebenfalls definieren. Warum aber dann einmal in TC und einmal in der MoBaSbS oder besser im Encoder selbst?

Ganz einfach: Definiert man einen Flackerschutz direkt im Encoder, so meldet dieser erst gar nicht solche sich wiederholende, kurze Rückmeldungen an den PMC, sondern nur das erste Auftreten innerhalb der Flackerschutzzeit. Wiederholte Impulse an seinem Eingang setzen lediglich die Flackerschutzzeit wieder zurück.

Die Flackerschutzzeiten können in der Maske für den Encoder im Konfigurationstool für jeden Eingang eines Encoders im Bereich von 10ms bis 2,5sec eingestellt werden.

Beschreibung der Flackerschutzzeit (Entprellzeit) im Detail:

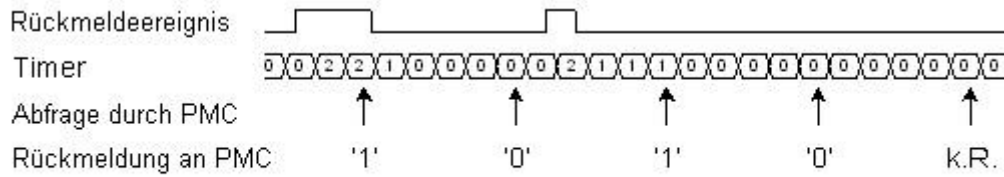
Für jeden Eingang getrennt, lässt sich eine Zeit zwischen 10ms und 2,5sec definieren, über die ein aufgetretenes Ereignis gehalten wird, auch wenn bereits wieder der Eingang inaktiv geworden ist. Außerdem wird der Zeitgeber mit jedem aktiven Ereignis neu gestartet. Mit dieser Option wird schon im Encoder verhindert, dass eine Vielzahl von Ereignissen an den PMC und von dort weiter über den IFC an den PC geleitet werden, die lediglich durch z.B. einen schlechten Rad-Schiene-Kontakt hervorgerufen werden. Die nachstehende Grafik veranschaulicht nochmals die Funktion:



k.R.: keine Rückmeldung, da keine Zustandsänderung

Wie man dem Bild entnehmen kann, fragt der PMC den Zustand des PM in zyklischen Abständen (symbolisiert durch die Pfeile) ab. Da vor dem ersten Rückmeldeereignis offensichtlich ein inaktiver Zustand anlag, wird bei der ersten Abfrage des PMC eine '1' an diesen zurückgemeldet. Obwohl der aktive Zustand am Eingang des PM bei der zweiten Abfrage durch den PMC nicht mehr anliegt, wird durch den Timer verhindert, dass der inaktive Zustand weitergeleitet wird. Da zwischen der 2ten und 3ten Abfrage erneut ein aktiver Zustand am Eingang des Encoders aufgetreten ist, wurde der Timer zurückgesetzt. Damit erfolgt weder bei der dritten, noch bei der vierten Abfrage eine Meldung über eine Zustandsänderung. Erst zur letzten Abfrage - hier ist der Zähler nun komplett abgelaufen - erfolgt die Rückmeldung des inaktiven Zustands.

Ein besonderer Fall wird im nächsten Bild behandelt. Die Zeit bis zum Senden eines inaktiven Zustands ist hier extrem kurz eingestellt. Alternativ kann man sich auch ein Szenario mit sehr vielen Encodern an einem PMC vorstellen, so dass der einzelne Encoder in größeren zeitlichen Abständen abgefragt wird. Es kann sich dann folgendes Szenario ergeben:



k.R.: keine Rückmeldung, da keine Zustandsänderung

Bei der ersten Abfrage durch den PMC wird eine '1' zurückgemeldet. Bei der zweiten Abfrage ist der Zähler bereits abgelaufen und deshalb wird nun eine '0' weitergegeben. Bis hierhin entspricht der Ablauf dem schon Gezeigten der ersten Darstellung. Nun tritt aber ein kurzer Impuls zwischen der zweiten und dritten Abfrage auf. Er ist gerade lang genug, um erkannt zu werden. Obwohl der Zähler bis zur nächsten Abfrage bis auf 0 herunterlaufen könnte, wird er aber jetzt beim Stand von 1 festgehalten, da der Encoder 'weiß', dass er noch einen '1' Zustand melden muss! Genau der wird nun auch bei der dritten Abfrage gemeldet und erst jetzt kann der Zähler bis auf 0 zurückgehen. Dies führt dann in der vierten Abfrage zu der Zustandsmeldung '0'. Bei der fünften Abfrage liegt keine Änderung vor und so wird nichts gemeldet. Anhand dieses Beispiels wird deutlich, dass auch ein nur sehr kurzes aufgetretenes Ereignis unter allen Umständen weitergegeben wird!

9 Service

9.1 μ C-Programmierboard

Um das Programmieren der μ Cs zu erleichtern, wurde ein Board entwickelt, auf dem die μ Cs über die ISP-Schnittstelle bequem programmiert werden können. Außerdem befindet sich auf dem Board eine Möglichkeit, selbstkonfektionierte Kabel zu testen.

Hierzu dienen die beiden RJ45-Buchsen, die Jumper und die acht LED. Nachdem ein Kabel konfektioniert wurde, wird es in die Buchsen gesteckt. Mit den Jumpers kann jede Ader des Kabels geprüft werden. Der Jumper schließt gegen Masse und die entsprechende LED muss aufleuchten.

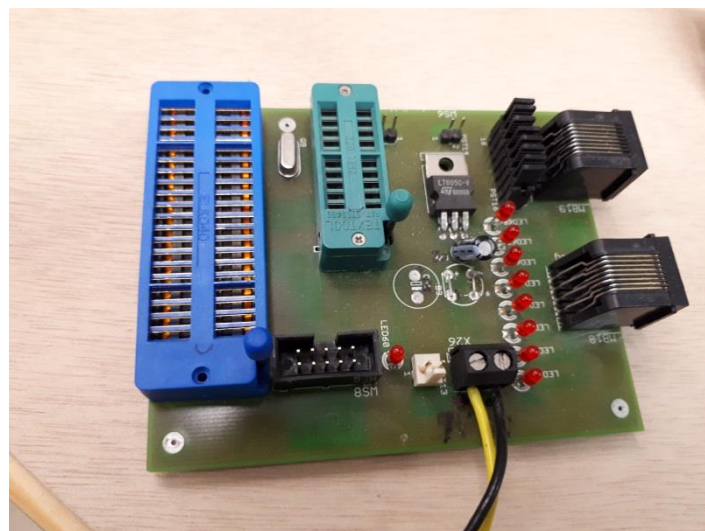
Das μ C-Programmierboard wird über den 10-poligen Wannenstecker an einen ISP-Programmer (siehe Kap. 9.2) angeschlossen.

An den Schraubklemmen muss eine Gleichspannung von ca. 9-12V DC angelegt werden, damit die μ Controller programmiert werden können.

Der 16-polige Sockel ist für die Programmierung der ATtiny-Controller geeignet. Auf dem 40-poligen Sockel können die ATmega162-Controller programmiert werden. ATmega644 Controller können nicht auf diesem Board programmiert werden. Dieser Controller wird nur in der 3,3V Variante des HDC eingesetzt und sollte nur über die ML-10 Buchse im HDC programmiert werden.

Da im Laufe der Erweiterungen an der Dekoder-Software ein neuer Prozessortyp verwendet werden musste (ATmega8), musste das μ C-Programmierboard erweitert werden. Es existieren deshalb mehrere Versionen:

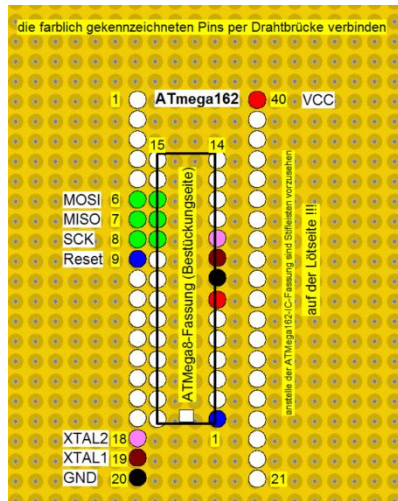
Erste Version



μ C-Programmierboard (erste Version)

Adaptersockel für ATmega8

Bei der o.g. Version steht für den ATmega8-Controller kein geeigneter Sockel zur Verfügung. Um ihn mit dem µC-Board programmieren zu können, muss eine Adapterplatine verwendet werden. Diese lässt sich auf einer Lochrasterplatine aufbauen.

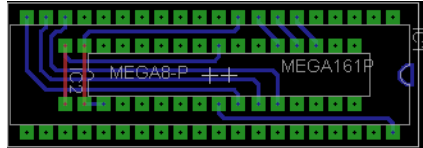


ATmega162 zu ATmega8 Adapter auf Lochrasterplatine

Hierbei sind die farbigen Punkte mit einer Drahtbrücke zu verbinden.

Hinweis: Es dürfen nicht alle sechs grünen Punkte miteinander verbunden werden, sondern nur die jeweils gegenüberliegenden.

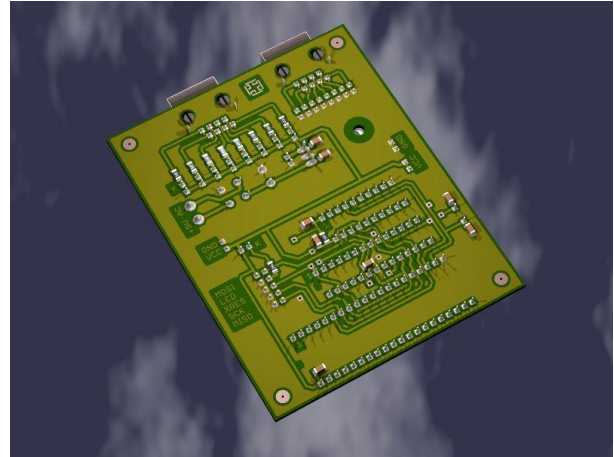
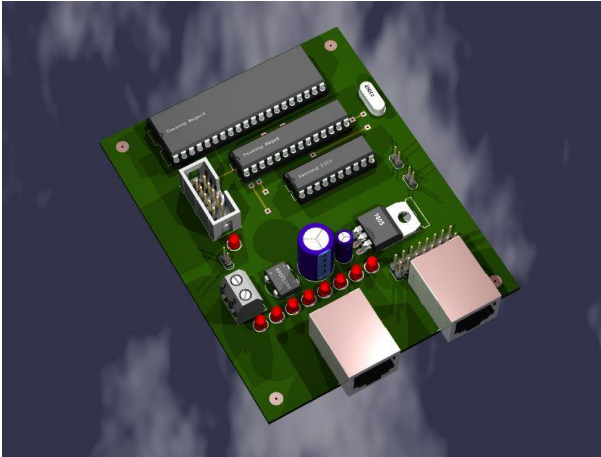
Eine andere Variante sieht so aus:



ATmega162 zu ATmega8 Adapter Platinenlyout

Die Verbindungen im Einzelnen:

ATmega162 PIN-Nr.	ATmega8 PIN-Nr.	Signal
6	17	MOSI
7	18	MISO
8	19	SCK
9	1	Reset
18	10	XTAL2
19	9	XTAL1
20	8 und 22	GND
40	7	VCC

Zweite Version (ohne Adapterplatine)

µC-Programmierboard (dritte Version)

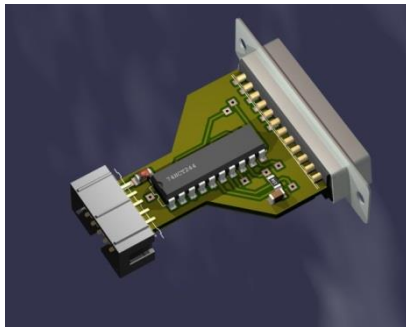
Hier sitzt der ATmega8-Sockel direkt auf der Platine.

9.2 ISP-Programmer

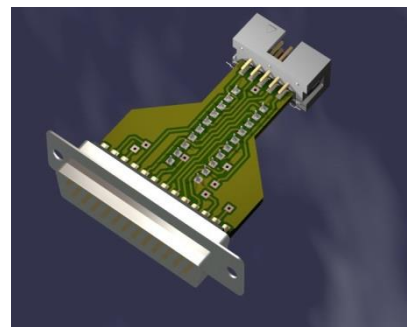
Ein ISP-Programmer erlaubt das Programmieren der μ Cs im System (ISP = In-System-Programming). Der Anschluss erfolgte über die parallele Schnittstelle. Da diese Schnittstelle von USB abgelöst wurde und aktuelle PCs keine parallele Schnittstelle besitzen, kann der nachstehend beschriebene ISP-Adapter nicht mehr eingesetzt werden. Aktuell werden ISP-Programmer verwendet, die über USB an den PC angeschlossen werden. Diese sind im Abschnitt nach dem ISP-Adapter beschrieben.

ISP-Adapter (für parallele PC-Schnittstelle)

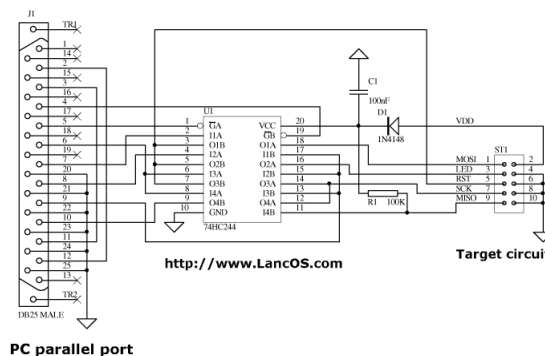
Der ISP-Adapter ist ein Programmierstecker, der an die parallele Schnittstelle des PCs angeschlossen wird. Basis ist die Vorlage AVR ISP-STK200 von Claudio Lanconelli, der in seinem Web ein Platinenlayout abgelegt hat.



ISP-Adapter, Bestückungsseite



ISP-Adapter, Lötseite



ISP-Adapter, Schaltbild

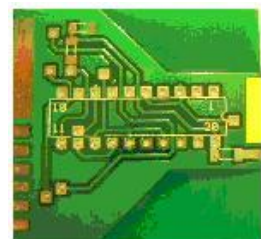
Der ISP-Adapter wird dabei vom μ C-Programmierboard (siehe Kap. 9.1) versorgt. Die Verbindung zwischen ISP-Adapter und μ C-Programmierboard wird durch ein 10poliges Flachbandkabel mit 10pol Aufquetschbuchsen hergestellt. Die Verbindung erfolgt 1:1, d. h. Stift1 wird mit Stift 1 verbunden, usw. Der ISP-Adapter selbst wird an die parallele Schnittstelle des PCs angeschlossen.

Programme, die auf die parallele Schnittstelle zugreifen, müssen geschlossen werden.

Für die Programmierung kann auf dem PC die Software PonyProg verwendet werden (siehe Kap. 10.2.2).

Korrektur aus Sammelbestellung 2006/2007

Hier fehlt die auf der in der Abbildung gelb markierte Massefläche, auf der die Beinchen des Wannensteckers angelötet werden. Kratzt man die grüne Lackschicht etwas weg, dann kann man den Wannenstecker anlöten.



ISP-Programmer

Aktuelle PCs haben keine parallele Schnittstelle mehr. Deshalb kann der ISP-Adapter nicht weiter eingesetzt werden. Als Ersatz stehen verschiedene ISP-Programmer zur Verfügung.

ATMEL Programmer AVR ISP mkII



ATMEL Programmer AVR ISP mkII

PonyProg arbeitet nur über Umwege mit diesem Programmer zusammen. Es wird deshalb das ATMEL Studio (siehe Kap. 10.2.2) oder eine andere Programmierumgebung empfohlen. Die Software kann man in der Regel kostenlos downloaden (siehe Kap. 5).

Der Programmer hat ausgangsseitig eine 6-polige Buchse, die man einfach auf die 10-polige Buchse des μ C-Programmierboards umbauen kann.

Wenn man von oben auf die Buchsen schaut (also nicht in die Löcher) und so, dass die Nase an den Buchsen nach links weist, dann sieht man an dem Farbschema, wie das Flachbandkabel von der 6-poligen Buchse mit der 10-poligen Buchse zu verdrahten ist.



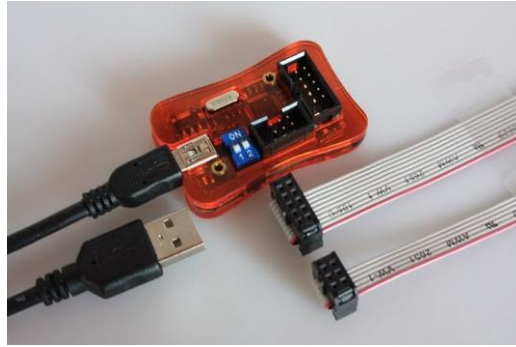
Buchse AVR ISP MKII und μ C-Programmierboard

Der Windows 7/8/10-Treiber für AVR ISP mkII ist manchmal nur mit sehr viel Mühe zu installieren. Ggf. muss im Internet nach einer Lösung gesucht werden.

Folgende Anleitung zur Installation des Treibers für den Programmer AVR ISP mkII unter Windows 7 64bit kann helfen:

Man installiere mittels des Gerätemanagers unter "Aktion" "Legacyhardware hinzufügen" "Hardware manuell aus einer Liste wählen und installieren", dann ,weiter' navigieren zu i.d.R C:\Program Files (x86)\Atmel\AVR Tools\usb64\windrvr6.inf den Treiber WinDriver.

Hiernach wird die USB-Software des AVRISP mkII nach Einstecken des USB-Steckers sauber erkannt und installiert. Sollte bereits vorher der USB-Treiber installiert worden sein, dann im Gerätemanager auf "Treiber aktualisieren" klicken oder das USB-Device deinstallieren.

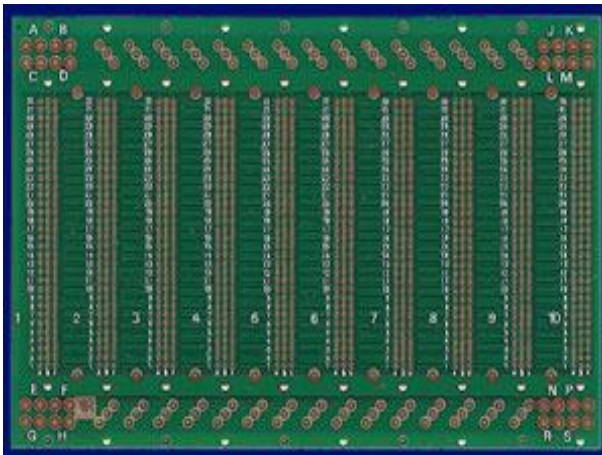
ISP-Programmer der Fa. TREMEX (wird auch von DIAMEX angeboten)ISP-Programmer der Fa. TREMEX/DIAMEX

Er ist besonders für Win10 zu empfehlen, da er keinen zusätzlichen Treiber benötigt. Für WIN7/8 kann ein Treiber auf der Lieferantenseite heruntergeladen werden (siehe Kap. 5).

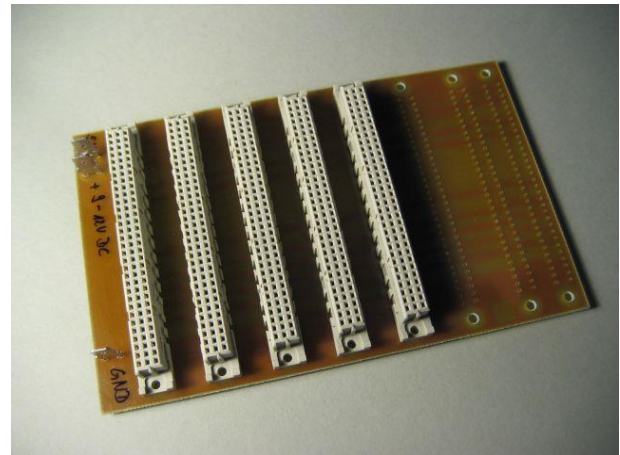
PonyProg arbeitet nur über Umwege mit diesem Programmer zusammen. Es wird deshalb das ATMEL Studio (siehe Kap. 10.2.2) oder eine andere Programmierumgebung empfohlen. Die Software kann man in der Regel kostenlos downloaden (siehe Kap. 5).

Der ISP-Programmer hat ausgangsseitig sowohl eine 6-polige als auch eine 10-polige Buchse.

9.3 Backplane



Backplane



Backplane, teilbestückt

Die Backplane dient der Verbindung der Platinen untereinander. Sie wird mit 2x32poligen Buchsen (A-C) bestückt. Sie kann bei Reichelt unter UP 940EP bestellt werden.

Auf dem Foto der Backplane sind 5 der maximal 7 Buchsenleisten eingelötet. Ganz rechts im Bild erkennt man die Bohrungen für einen Adapter.

Alle für die MoBaSbS relevanten Signale sind auf den Buchsenleisten untereinander verdrahtet. Besondere Rücksicht wurde auf die Masse (GND) und Versorgungsanschlüsse (VCC und +15V) genommen. Die Anschlüsse für die Spannungen sind als 6.3mm Flachstecker ausgeführt.

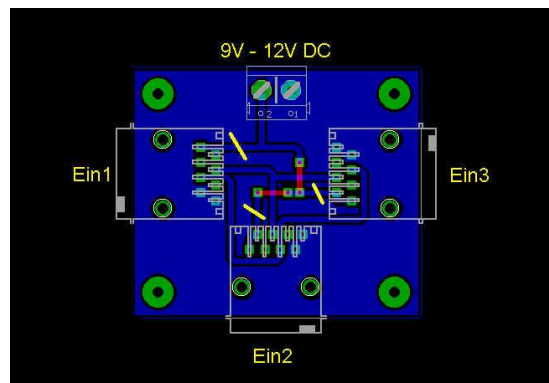
Für die MoBaSbS sind die 7 Steckplätze auf einer Platine ausreichend. Möchte man die Platinen mit etwas größerem Abstand einbauen, so kann die linke Platine über einen Adapter zur rechten Platine erweitert werden. Der Rastermaßabstand für eventuelle Frontplatten wird dabei eingehalten.

Beide großen Platinen werden ausschließlich mit der gleichen Federleiste bestückt (FL-C 64G4). Die Adapterplatine wird 'über Kopf' auf die Hauptplatinen gesteckt. Auf ihr befinden sich Messerleisten (gerade) vom Typ (ML-R 64G4).



MoBaSbS-Backplane erweitert, Platinen-Layout

9.4 Verteiler Con888



Verteiler Con888, Platinen-Layout

Der Verteiler Con888 ist Hilfsplatine für die Verkabelung der Decoder und Encoder.

Er erfüllt zwei Aufgaben:

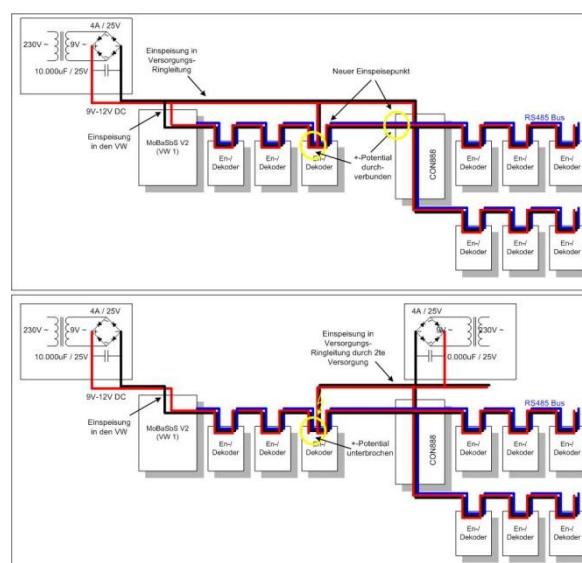
Aufteilung der Decoder-/Encoder-Stränge

Ein Decoder-/Encoder-Strang kann zwei Stränge aufgeteilt werden

Einspeisung

Eine erneute Einspeisung der ca. 9-12V DC zur Versorgung des TTL-Teils der Encoder und Decoder ist hier möglich. Dafür muss die eingehende Leitung von der neuen Einspeisung getrennt werden. Die Trennung kann entfallen, wenn die Versorgung aus derselben Spannungsquelle kommt. Auf dem Platinenlayout sind die möglichen Trennstellen eingezeichnet (gelbe Striche), die je nach ankommender Leitung vorgenommen werden müssen.

Auf das Problem des Spannungsabfalls der Versorgungsspannung der TTL-Logik der Encoder und Decoder innerhalb eines Stranges ist schon an anderen Stellen hingewiesen worden. In der Grafik wird beispielhaft die Neueinspeisung und Verteilung über den Verteiler Con888 beispielhaft gezeigt.



Verteiler Con888, Beispiel Einspeisung

10 Microcontroller programmieren

In der MoBaSbS werden ATMEL- μ Controller eingesetzt. Die aktuellen Typen sind ATmega644, ATmega162, ATtiny2313 und ATmega8. Nicht mehr lieferbar, aber auch einsetzbar ist AT90S2313.

10.1 Hardwareumgebung

Für die Programmierung der ATMEL- μ Controller wird ein ISP-Programmer benötigt (siehe Kap. 9.2). Der ISP-Programmer wird an den PC angeschlossen und über ein Flachbandkabel wird entweder das μ C-Programmierboard (siehe Kap. 9.1) oder die jeweilige Platine (z. B. HDC, BMC,...) verbunden.

Auf den Platinen brauchen die ATMEL- μ Controller eine stabilisierte 5V-Gleichspannung. Diese muss über den 2poligen Stecker mit der Führungsnase (Verpolschutz) zugeführt werden.

Das μ C-Programmierboard ist mit einem 5 Volt-Stabilisator ausgerüstet. Hier müssen über die Schraubklemmen 9-12V DC zugeführt werden.

Der ATMEL- μ Controller behält nach Abschalten der Versorgungsspannung seinen programmierten Inhalt, so dass eine Pufferung entfällt.

10.2 Programmierumgebung

10.2.1 Verzeichnisstruktur

Die MoBaSbS-Dateien liegen als gepackte Versionen im geschützten MoBaSbS-Bereich. Die zip-Archive der MoBaSbS-Mikrocontroller-Codes werden in ein neu angelegtes Verzeichnis (z.B. MoBaSbS_uC_Code) kopiert und entpackt. Die Verzeichnisstruktur muss mit entpackt werden! Bei vielen Packprogrammen ist das eine Option, die beim Entpacken explizit angegeben werden muss.

Nach dem Entpacken enthält die entstandene Verzeichnisstruktur die bereits compilierten Mikrocontroller-Codes (hex- und eep-Dateien).

10.2.2 Transfer der *.hex- und *.eep-Dateien zum Controller

Zum Laden der *.hex und *.eep-Dateien in den μ Controller benötigt man ein Transfer-Programm. Es stehen mehrere Möglichkeiten zur Verfügung, von denen nachfolgend zwei Programme beschrieben werden.

PonyProg

PonyProg kann nur zusammen mit dem ISP-Adapter (siehe Kap. 9.2) eingesetzt werden. ISP-Programmer mit USB sind für Ponyprog nicht mit Einschränkungen geeignet.

Mit dem Programm lässt sich auch der μ C auslesen, löschen, verifizieren und die Fuse-, Configuration- und Security-Bits setzen. Es besteht die Möglichkeit den zu programmierenden Code zu editieren, z.B. um Konfigurationswerte zu ändern. Dafür muss die Option ‚EDIT Buffer Enabled‘ eingeschaltet werden.

Zum Umgang mit PonyProg wurde von Karsten Thüner ein Video erstellt, das aus dem geschützten MoBaSbS-Bereich heruntergeladen werden kann.

Sollte keine Verbindung über die parallele Schnittstelle möglich sein, ist zu prüfen, dass auf dem PC die DLPORTIO.SYS aktiv ist.

Es ist darauf zu achten, mindestens die Version PonyProg2000 – Serial Device Programmer V2.06e Beta Feb 3 2005 einzusetzen. Ältere Versionen funktionieren nicht.

Hinweis zu PonyProg unter Win7 32-bit:

Das Programm muss mit Admin-Rechten ausgeführt werden. Einfach auf die PonyProg.exe mit rechts klicken und Eigenschaften wählen. In dem Reiter Kompatibilität einen Haken bei "Programm als Administrator ausführen" setzen und mit "Übernehmen" bestätigen

Nach der Installation muss die Software konfiguriert werden. Dazu wird das Menü "Setup" - "Interface Setup" aufgerufen. In dem erscheinenden Fenster werden die Punkte "Parallel", sowie "Avr ISP I/O" und der entsprechende LPT-Port ausgewählt. Alle anderen Selektionsboxen werden nicht markiert.

Oben rechts in PonyProg kann noch Avmicro und der μ Controller ausgewählt werden (ATmega162, AT90S2313, ATtiny2313, ATmega8) oder man lässt hier "AVR Auto" stehen.

Nun wird der Code geladen. Dies geschieht über das Menü "File" und dort über die beiden Punkte "Open Program FLASH File" und "Open Data EEPROM File". Es sind dann die jeweiligen *.hex und *.eep-Dateien auszuwählen und in PonyProg zu laden.

Danach wird programmiert.

Dazu muss der ISP-Adapter an der parallelen (Drucker-)Schnittstelle angeschlossen sein. Das andere Ende steckt in der ISP-Buchse am μ C-Programmierboard. Der ATMEL- μ Controller befindet sich in seiner Fassung und die Versorgungsspannung ist eingeschaltet.

Bevor ein neues Programm (FLASH) geladen werden soll, muss der ATMEL- μ Controller erst einmal gelöscht werden. Dafür gibt es im Menü ‚Command‘ das Kommando "Erase". Danach wird zuerst das Flash geschrieben "Write Program (FLASH)", dann eventuell existierende EEPROM Daten. Mit den Optionen "Verify" lässt sich der Programmiervorgang anschließend noch einmal kontrollieren.

ATMEL Studio

Für die Programmierung mit einem ISP-Programmer kann ATMEL Studio (früher AVR Studio) eingesetzt werden. Die Software umfasst eine komplette Entwicklungsumgebung und kann auf den ATMEL-Seiten (siehe Kap. 5) heruntergeladen werden.

Für den Umgang und die Programmierung gibt es im Internet zahlreiche Beschreibungen. Empfehlenswert sind die Seiten von mikrocontroller.net und youtube-Videos.

10.2.3 Fuse-, Configuration- und Security-Bits

Die Fuse-Bits eines Mikrocontrollers dienen zu dessen Konfiguration und können von der darauf laufenden Software nicht verändert werden. Durch die Fuse-Bits können Funktionen eines Mikrocontrollers ein- oder ausgeschaltet werden und sind nichts anderes als programmierbare Speicherzellen.

Bei der Programmierung der Fuse-Bits muss darauf geachtet werden, dass ein programmiertes beziehungsweise aktiviertes/gesetztes Fuse-Bit auf den Wert 0 (Ponyprog: Haken) gesetzt wird.

Die Fuse-Bits haben folgende Bedeutung:

BODEN

BODEN steht für *Brown out detector enable*. Dadurch kann die Brown Out Detection aktiviert beziehungsweise deaktiviert werden.

BODLEVEL0 bis BODLEVEL2

BODLEVEL steht für *Brown-out Detector trigger level*. Mit diesen Fuse-Bits kann der Spannungswert eingestellt werden, bei welchem der Unterspannungsschutz aktiv werden soll. Mit dem Fuse-Bit BODEN kann diese Funktion komplett deaktiviert werden.

BOOTRST

BOOTRST steht für *Select Reset Vector*. Mit diesem Fuse-Bit wird bestimmt, an welche Adresse nach einem Reset gesprungen werden soll. Ist das Fuse-Bit deaktiviert, springt der Mikrocontroller nach einem Reset an Adresse 0000. Bei aktiviertem Fuse-Bit springt der Mikrocontroller nach einem Reset an den Beginn des Bootloaders.

BOOTSZ0 und BOOTSZ1

BOOTSZ steht für *Select Boot Size*. Mit BOOTSZ wird die Größe des Speicherbereiches bestimmt, welcher für den Bootloader reserviert wird. Die Größe des Speicherbereiches ist abhängig vom Controllertyp und befindet sich immer am Ende des Flash Adressraumes. BOOTSZ wird nur benötigt, wenn das Fuse-Bit BOOTRST aktiviert ist.

DWEN

DWEN steht für *debugWIRE Enable* und aktiviert beziehungsweise deaktiviert die debugWIRE Schnittstelle.

CKDIV8

CKDIV8 steht für *Divide Clock by 8*. Wenn CKDIV8 aktiviert ist, wird ein Vorteiler aktiviert, welche den Takt für den Mikrocontroller durch 8 teilt.

CKOPT

CKOPT steht für *Oscillator Options*. Dieses Fuse-Bit wird benötigt, wenn der Mikrocontroller von einem externen Quarz getaktet wird. Wird CKOPT aktiviert, schwingt der Oszillator mit der maximalen Amplitude. Ist CKOPT deaktiviert, schwingt der Oszillator mit einer niedrigeren Amplitude. Dadurch kann die Stromaufnahme und die Störabstrahlung verringert werden.

CKOUT

CKOUT steht für *Clock output*. Ist dieses Fuse-Bit aktiviert, wird das CPU-Taktsignal an den CLKO Pin ausgegeben.

CKSEL0 bis CKSEL3

CKSEL steht für *Select Clock Source*. Durch diese Fuse-Bits legt man fest, von welcher Taktquelle der Mikrocontroller seinen Takt erhält. Eine fehlerhafte Einstellung kann den Mikrocontroller lahm legen, ist jedoch leicht wieder zu beheben.

EESAVE

Bei aktiviertem EESAVE wird das EEPROM bei einem Chip Erase vor dem Löschen geschützt. Normalerweise löscht ein Chip Erase den kompletten Speicher.

JTAGEN

JTAGEN steht für *Enable JTAG*. Durch JTAGEN kann die JTAG-Schnittstelle aktiviert beziehungsweise deaktiviert werden.

OCDEN

OCDEN steht für *Enable OCD*. Damit kann das On-Chip Debug System aktiviert beziehungsweise deaktiviert werden. Das On-Chip Debug System kann unabhängig von der JTAG-Schnittstelle deaktiviert werden. Wurde das On-Chip Debug System deaktiviert, kann der Mikrocontroller über die JTAG-Schnittstelle nur programmiert werden.

SPIEN

Durch das SPIEN Fuse-Bit kann die ISP-Schnittstelle deaktiviert werden. Dieses Fuse-Bit kann nur mit einem Parallel-Programmierer (z.B. STK500) verändert werden.

SUT0 und SUT1

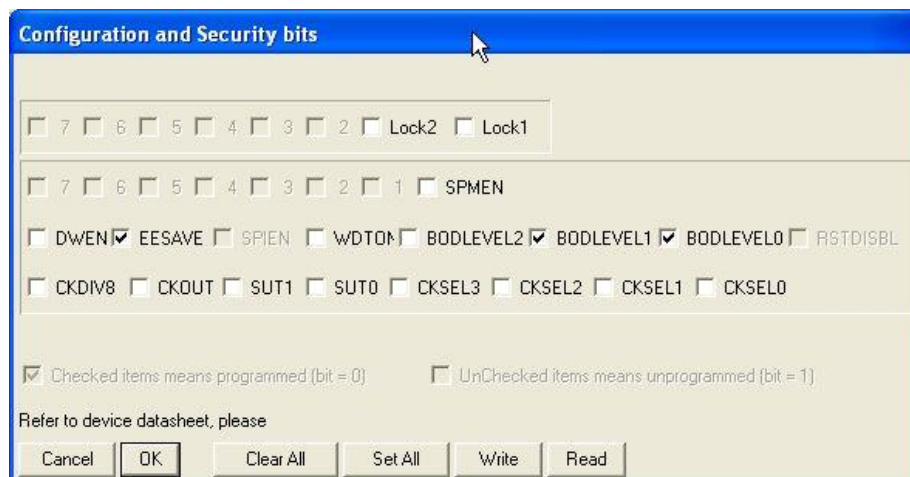
SUT steht für *Select Start-Up Time*. Durch SUT0 und SUT1 lässt sich die Zeit einstellen, wie lange der Reset Impuls nach einem Reset beziehungsweise Power Up verzögert wird.

WDTON

WDTON steht für *Watchdog timer always on*. Dieses Fuse-Bit aktiviert den Watchdog Timer permanent. Ist dieses Fuse-Bit nicht aktiviert, kann der Watchdog per Software gesteuert werden.

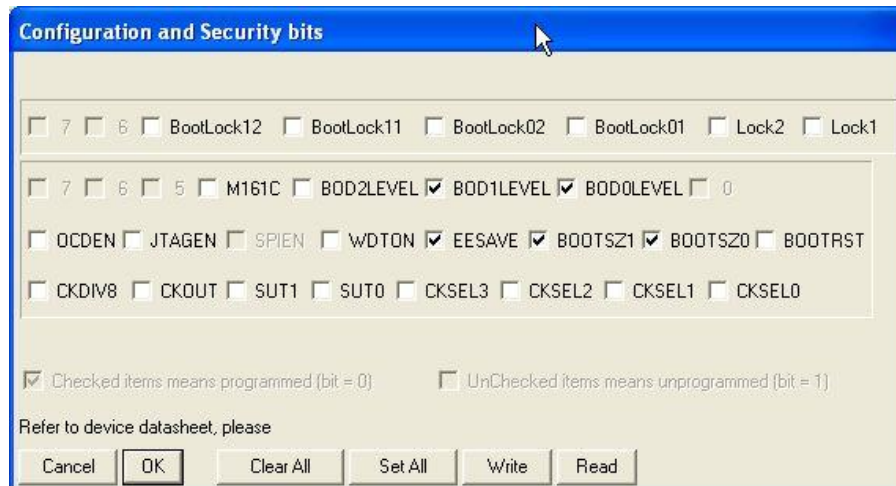
AT90S2313

Beim AT90S2313 brauchen keine Bits gesetzt zu werden.

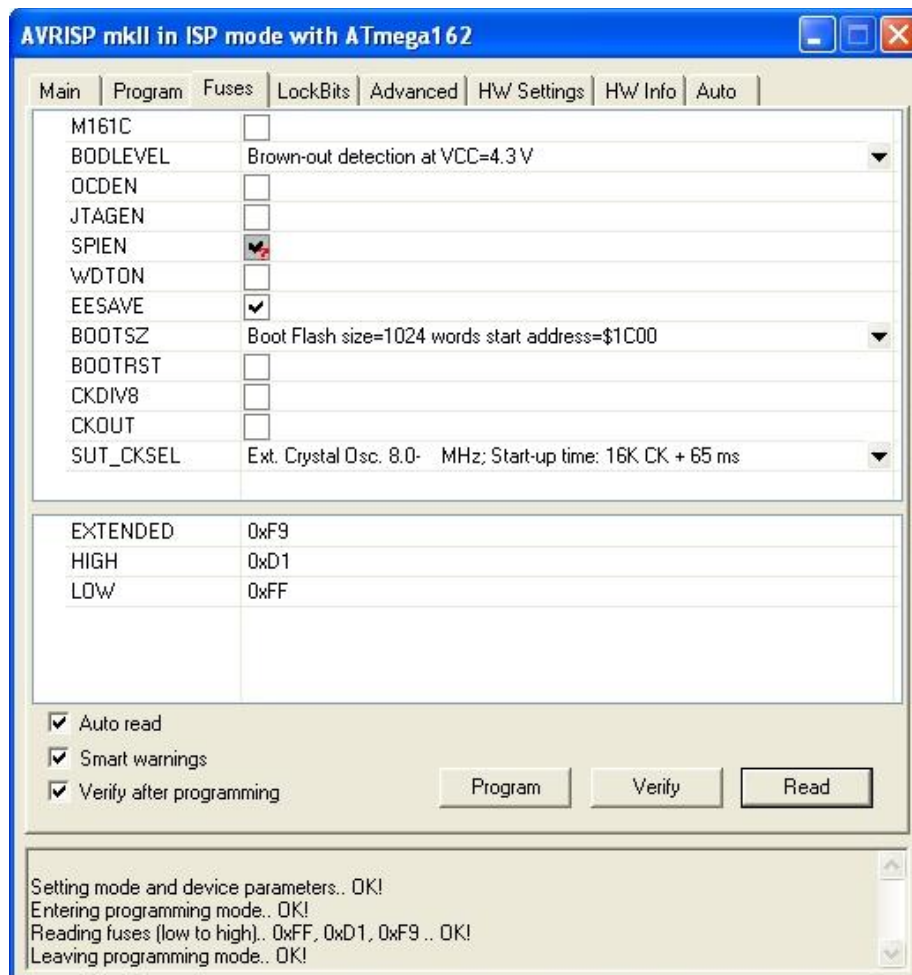
ATtiny2313

ATtiny2313, Fusebits in PonyProg

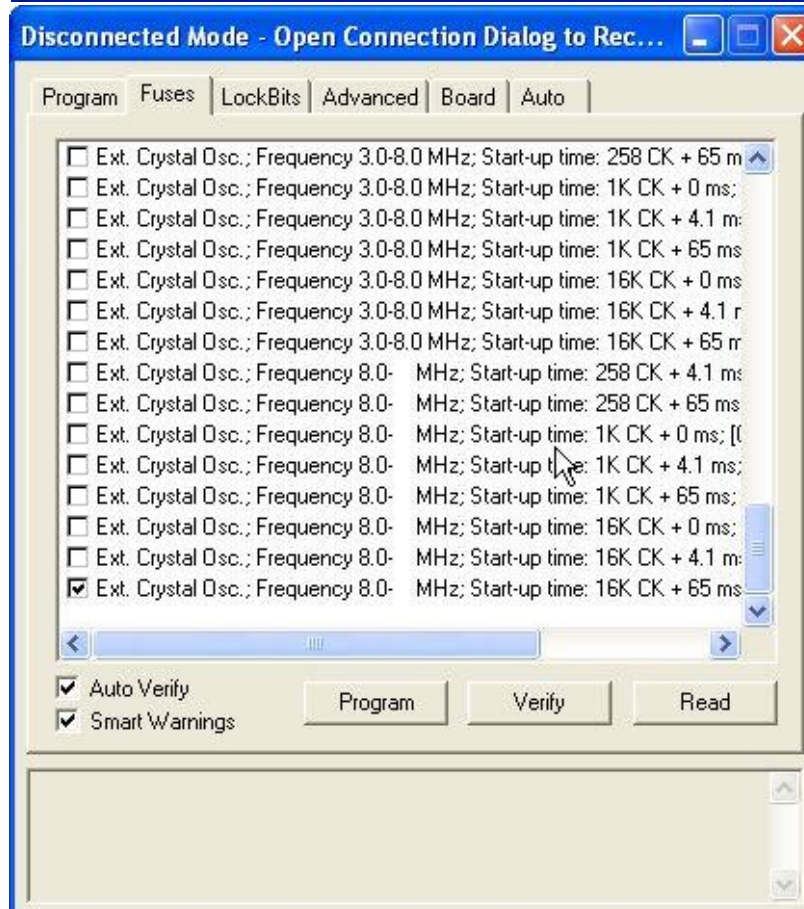
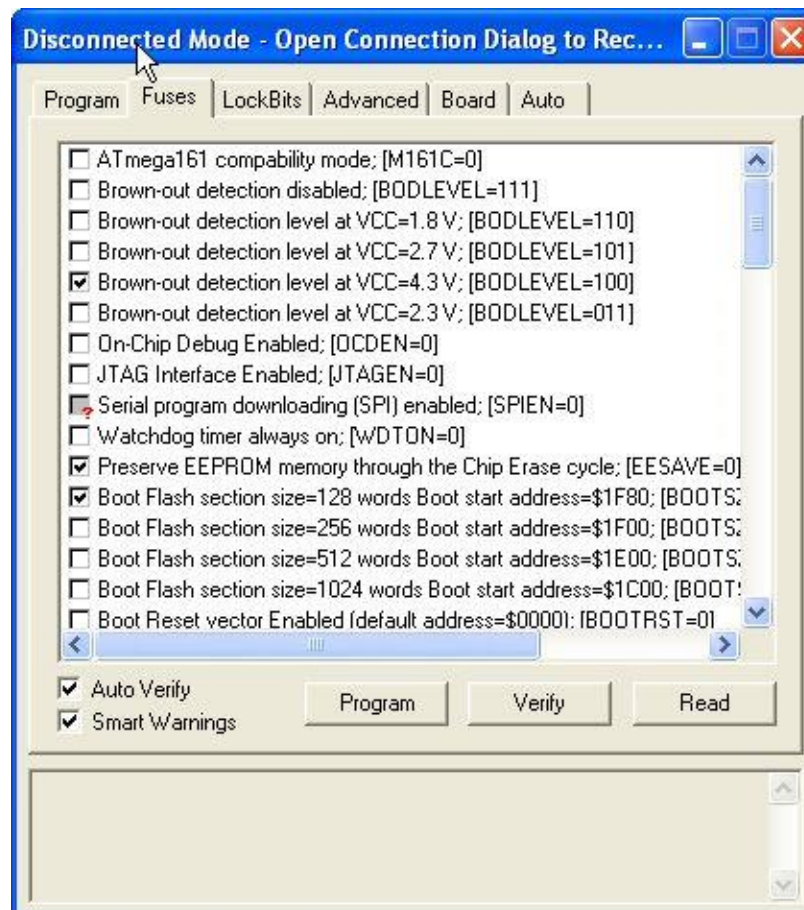
ATmega162



ATmega162, Fusebits in PonyProg ohne /mit Mäusezahl-Bootlader



ATmega162, Fusebits in ATMEL Studio ohne /mit Mäusezahl-Bootlader



ATmega162, Fusebits in ATMEL Studio ohne /mit Mäusezahl-Bootlader

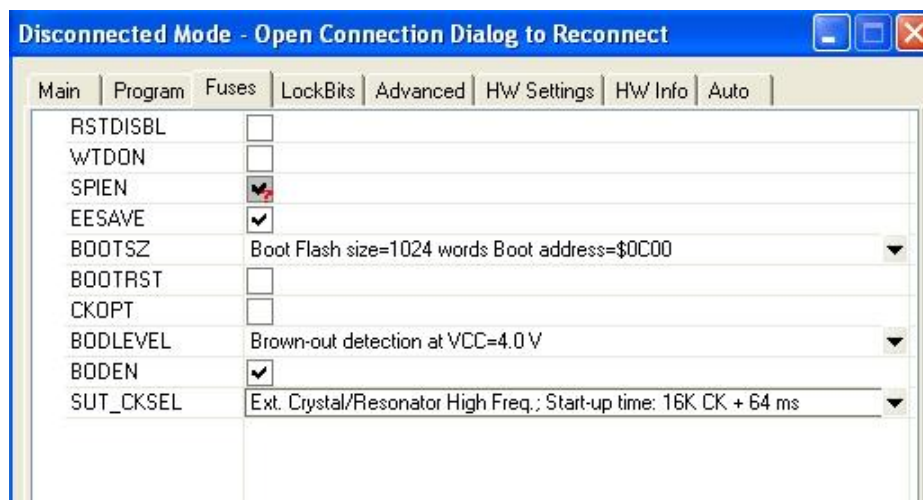
ATmega8

Beim ATmega8 müssen vor der ersten Programmierung des Speichers die Fusebits richtig gesetzt werden. Mit den ausgelieferten Defaulteinstellungen lässt sich der Prozessor nach der ersten Programmierung nur noch über einen extern angelegten Takt wiederbeleben. Wichtig sind dabei die Einstellungen CKSEL = 1111 und SUT = 01. Dies bedeutet, dass in PonyProg in den Kästchen CKSEL 0-3 und in SUT 0 keine Häkchen gesetzt werden dürfen.

Hat man dies einmal nicht beachtet, muss extern ein Takt angelegt werden (z. B. Signalgenerator mit einem Rechtecksignal und einer Amplitude von 5V).

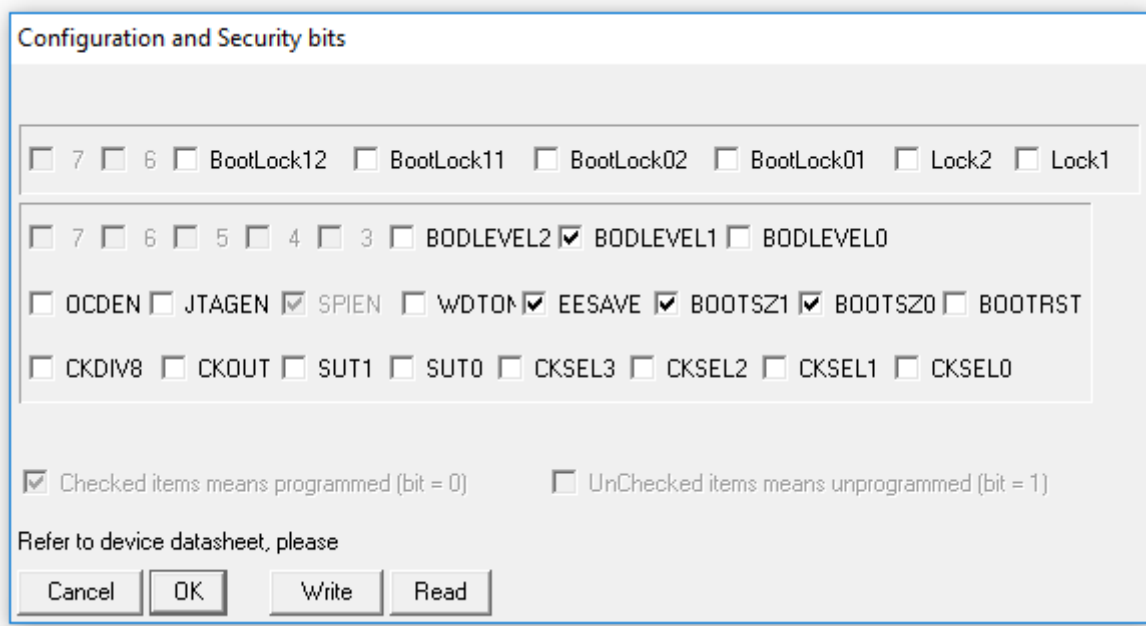


ATmega8, Fusebits in PonyProg ohne / mit Mäusezahl-Bootlader
(Lintzmeyer-Bootlader siehe Kap. 13.4)



ATmega8, Fusebits in ATMEL Studio ohne / mit Mäusezahl-Bootlader

ATmega644



ATmega644, Fusebits in PonyProg

Fuse	Value
BODLEVEL	Brown-out detection at VCC=2.7 V
OC DEN	<input type="checkbox"/>
JTAGEN	<input type="checkbox"/>
SPIEN	<input checked="" type="checkbox"/>
WDTON	<input type="checkbox"/>
EESAVE	<input checked="" type="checkbox"/>
BOOTSZ	Boot Flash size=4096 words Boot address=\$7000
BOOTRST	<input type="checkbox"/>
CKDIV8	<input type="checkbox"/>
CKOUT	<input type="checkbox"/>
SUT_CKSEL	Ext. Crystal Osc. 8.0- MHz; Start-up time: 16K CK + 65 ms

Lock Bit Byte 0xFF

Fuse Bits

- Extended 0xFD
- High Byte 0xD1
- Low Byte 0xFF

ATmega644, Fusebits in ATMEL Studio

Insbesondere auf die Brown-out detection achten.

ACHTUNG!

Den ATMEGA 644 hat eine vom ATMEGA162 abweichende Pin-Belegung, daher nicht in das MoBaSbS Programmierbord stecken. Es droht Bauteilerstörung. Den ATMEGA644 nur eingebaut im HDC programmieren.

11 Inbetriebnahme

11.1 Vorgehensweise

1. Nach dem Zusammenbau **einer** Platine und noch **bevor** die ICs eingesetzt werden und die 5V Versorgung eingeschaltet wird, sollte mit einem Multimeter getestet werden, ob zwischen GND und +5V ein Kurzschluss vorliegt.
2. Jetzt wird die Versorgungsspannung angelegt. Nach wie vor wurden **keine** ICs bestückt! Nun wird abermals mit einem Multimeter getestet, ob an den entsprechenden Pins der IC Sockel auch wirklich die 5V anliegen (4.75V bis 5.25V sind in Ordnung.). Wenn alles OK ist, dann wird die Spannung wieder **ausgeschaltet**. Bei der 3,3V Variante des HDC ist die Versorgungsspannung 3,3V.
3. Nun werden die ICs eingesetzt und der μ C über den ISP-Dongle und PonyProg programmiert. Dabei werden **zuerst** die Fusebits des μ Cs korrekt gesetzt und auch geschrieben. Danach wird der Programm (*.hex) und - sofern vorhanden - der EEPROM (*.eep)-Code in den μ C geladen.
4. Zuerst einmal nur BMC und USC auf die Backplane stecken. Wenn hier die X-Bus LED nicht leuchtet -> Checke Pkt. 3.
5. Wenn die X-Bus LEDs blinken, dann wird der IFC dazu gestreckt. Es sollten nun wieder die X-Bus LEDs leuchten (obwohl der HDC noch auf Adresse 31 steht - der BMC ist so intelligent und fragt neben den Adressen 1 bis 9 auch 31 mit ab). Es kann sein, dass die Xbus-LED des IFC erst dann anfängt zu blinken, wenn das USB-Kabel eingesteckt wird. Blinkt die IFC LED nicht -> Checke Pkt. 3
6. Nun wird der HDC angeschlossen. Blinkt die IFC LED nicht -> Checke Pkt.3. Bricht nun beim Betätigen einer VFP Taste am HDC der X-Bus zusammen, dann liegt ein Fehler im SRAM oder Eeprom des BMCs vor. Falls es sich um einen Eeprom Fehler handelt, kann man dies herausfinden, indem man über das MoBaSbS-Konfigurationstool einen Datensatz anlegt, bevor man eine VFP-Taste drückt. Dieser wird im SRAM des BMCs hinterlegt. Kann jetzt eine Lok im HDC aufgerufen werden, dann liegt ein Fehler im Eeprom vor, andernfalls ist das SRAM defekt. Achtung: Diese Prozedur ist über den HDC nicht möglich!
7. Weitere Fehler kann man über die Testumgebung von Günter (siehe Kap. 11.3) herausfinden.
8. Wenn die X-Bus LEDs alle leuchten, dann wird die MoBaSbS wieder ausgeschaltet, mit dem PC verbunden und wieder eingeschaltet.
9. In Traincontroller wird als Digitalsystem 'Lenz Digital Plus / LI100(F)' oder 'MOBASBS' und die entsprechende serielle Schnittstelle eingetragen.

11.2 Häufig auftretende Fehler

- Die Spannungsversorgung wurde nicht angeschaltet
- Dioden wurden verkehrt bestückt
- Kondensatoren wurden verkehrt bestückt
- (Masse-)Lötbrücken wurden vergessen
- Die Fusebits der μ C wurden zwar gesetzt aber nicht geschrieben
- Auf dem BMC wurden zum Anschluss des Fahrpults die entsprechenden Jumper nicht gesetzt
- Sowohl auf dem BMC, als auch auf dem PMC wurden die Jumper zur Spannungsversorgung der HDCs bzw. der Peripheriemodule nicht oder falsch gesetzt
- Im Traincontroller wurden Elemente definiert, die von der MoBaSbS nicht unterstützt werden. Z.B wurden Rückmeldekontakte auf Adressen der MoBaSbS gelegt, für die kein PMC vorhanden war.
- Es wurde die falsche serielle Schnittstelle in Traincontroller definiert

11.3 Testumgebung

Günter Lintzmeyer hat eine Testumgebung erstellt, mit der man losgelöst von der MoBaSbS die korrekte Funktion des IFC einerseits und der Kombination aus BMC und IFC andererseits testen kann.

Über ein Terminalprogramm wird eine Zahl an den IFC gesendet. Als Antwort erhält man vom IFC daraufhin den um 10 erhöhten Wert der gesendeten Zahl zurück. So lässt sich sehr schnell feststellen, ob der Aufbau des IFC und die Verbindung zum PC korrekt sind, bzw. in einem 2ten Schritt, ob das IFC mit der BMC korrekt zusammen läuft und vom PC aus angesprochen werden kann.

Mit dem Programm TEST_TSG können einzelne Funktions-Prüfungen des USC (z.B. ob das Relais schaltet, Verbindungen zwischen M162-Ports und den anderen ICs ok, usw.) durchgeführt werden (Einzelheiten stehen in der aktualisierten Programm-Beschreibung).

In einem 2ten Schritt (TEST_PCIF) kann geprüft werden, ob das IFC mit dem BMC korrekt zusammen läuft und vom PC aus angesprochen werden kann. Gleichzeitig können die auf dem BMC vorhandenen ICs (RAM-Speicher, bzw. EEPROM 24C512) getestet werden.

Ein weiteres Programm TEST_DVW prüft den PMC. Insbesondere können dabei die korrekten Verbindungen der einzelnen Stränge zu den Decodern kontrolliert werden.

Für die Decoder ist dabei ein weiteres Programm TEST_DEC beigefügt (für Atiny2313 / 8MHz) mit dem das Zusammenspiel >PMC – Decoder< überprüft werden kann.

Alle benötigten Dateien, Programme und Dokumentation können im geschützten MoBaSbS-Bereich als ZIP-File heruntergeladen werden.

Das Archiv enthält:

AVRTERM.EXE	Terminalprogramm, um Zeichen an das PC-IF zu senden und zu empfangen
AVRTERM.CFG	Konfigurationsprogramm zu AVRTERM
TEST-Programme.pdf	Dokumentation
TEST_PCIF.HEX	spezieller Mikrokontroller Code für den Funktionstest des IFC
TEST_CE.HEX	spezieller Mikrokontroller Code für den Funktionstest des BMC
TEST_TSG.HEX	spezieller Mikrokontroller Code für den Funktionstest des USC
TEST_DVW.HEX	spezieller Mikrokontroller Code für den Funktionstest des PMC
TEST_DEC.HEX	Direkt bei G. Lintzmeyer anfordern

Das Archiv wird in ein leeres Verzeichnis entpackt. Weitere Informationen enthält die Programmbeschreibung.

Die Testprogramme funktionieren ohne Berücksichtigung der MoBaSbS-spezifischen Einstellungen am IFC-Dippschalter.

Auf dem Bildschirm AVRTERMINAL muss >Decimal< selektiert werden, die Bildschirmanzeige >CTS-Status< muss nach einem Reset rot / blau blinken.

Die Parameter für die COM-Schnittstelle müssen im Bildschirm AVRTERMINAL >Settings< wie folgt gewählt werden: 19200,8,1,N

Die Testprogramme funktionieren nur über die COM-Schnittstelle. Das COM-Verbindungskabel (9-poliges Kabel) muss 1:1 durchverbunden werden. Ein NULL-Modem-Kabel ist nicht verwendbar.

12 MoBaSbS-Konfigurationstool

Zum Umgang mit dem MoBaSbS-Konfigurationstool wurden von Karsten Thüner vier Videos erstellt, die auf den geschützten Seiten von Patrick abgerufen werden können.

Mit der Version 4 der MoBaSbS werden alle Module im System konfiguriert. Jedes Peripherie-Modul kann bei Änderungen in der Anlage verbleiben. Lediglich das erstmalige Programmieren der Mikrocontroller mit den bereitgestellten, kompilierten *.hex und *.eep Dateien ist noch notwendig. Alle weiteren Einstellungen werden über das MoBaSbS-Konfigurationstool durchgeführt.

Die benötigte Software liegt im geschützten MoBaSbS-Bereich und wurde von Karsten Thüner programmiert.

Die nachfolgende beschriebene Version des MoBaSbS-Konfigurationstools ist V4.60 vom 10.03.2013.

Gelegentlich treten nach Benutzung des MoBaSbS-Konfigurationstools Störungen beim Stellen mit autom. Weichenstraßen in Traincontroller auf. Deshalb sollte die MoBaSbS nach Benutzung mit dem Konfigurationstool immer aus- und wieder eingeschaltet werden, bevor sie mit Traincontroller verbunden wird. Ein Reset an der MoBaSbS ist nicht ausreichend.

Bei Startproblemen der Software kann eine minimale und universelle Initialisierungsdatei MoBaSbS.ini helfen. Sie kann unter <http://mobasbs.thuener-oe.de/download/download.htm> heruntergeladen werden. Diese muss in das Start-Verzeichnis der Konfigurationssoftware parallel zur Datei MoBaSbS.tcl gespeichert werden. Danach sollte die Software starten und es können weitere Einstellungen vorgenommen werden.

12.1 Installation

Es ist darauf zu achten, dass im Windows-System keine großen Schriftarten aktiviert werden. Andernfalls werden die Text-Tabellen des PMC sowie die des BM- und PM-Reports deutlich vergrößert und verdecken damit rechts liegende Elemente.

12.1.1 Tcl/Tk-Interpreter installieren

Das MoBaSbS-Konfigurationstool ist komplett in Tcl/Tk geschrieben. Um sie benutzen zu können, muss ein Tcl/Tk-Interpreter auf dem PC installiert werden.

Die bisherige Version 8.4.13 von ActiveState kann weiter benutzt werden, wird aber nicht mehr zum Download angeboten.

Ab der Version 8.5.8.1 bietet ActiveState nicht mehr alle Tcl-Zusatzpakete in seinem Softwarepaket an. Für die MoBaSbS-Konfigurationssoftware bedeutet das, dass keine Audio-Ausgaben mehr möglich sind. Hierfür wird das Audio-Paket ‚Snack‘ benötigt, das manuell nachinstalliert werden kann (siehe unten). Ab der Version 4.42 der Konfigurationssoftware wird die Verfügbarkeit des Audio-Pakets ‚Snack‘ geprüft und bei Fehlen werden nach einem Hinweis die Audio-Ausgaben deaktiviert.

Auch mit ActiveTcl 8.5.14 läuft die Konfigurationssoftware einwandfrei.

Der Download der Software ist frei. Ein direkter Link ist in der Linkliste (siehe Kap. 5) aufgeführt.

Es stehen sowohl 64-als auch 32bit-Versionen zur Verfügung. Sollte das Config-Tool nicht mit der 64bit-Version laufen, muss die 32bit-Version verwendet werden.

Das selbstentpackende Archiv hat eine Größe von ca. 25 Mbyte. Sinnvoll ist es, die Datei als lokale Datei auf einem lokalen Speichermedium (Festplatte) zu speichern und dann von dort aus zu starten.

Durch Ausführen der Datei (z. Zt.: ActiveTcl8.5.9.1.294121-win32-ix86-threaded.exe) öffnet sich das Installationsprogramm mit der Überschrift "ActiveState ActiveTcl <Version> Installer.

1. Der erste Bildschirm gibt eine Übersicht über die enthaltenen Pakete und deren Version. Durch Drücken des Knopfes 'Next' gelangt man zum 'License Agreement', das in Englisch verfasst.
2. Akzeptieren durch Anklicken des Ausdrucks 'I accept the terms in the License Agreement' und weiter geht's mit einem Druck auf den Knopf 'Next'
3. Im nächsten Bildschirm gibt es nun mehrere Auswahlmöglichkeiten. Im oberen Kasten kann die Software für den aktuellen Benutzer ausschließlich (obere Option aktiviert) oder alle Benutzer des PCs (untere Option aktiviert, default) installiert werden. Die beiden Optionen im darunterliegenden Kasten bitte beide aktiviert lassen, da sonst das MoBaSbS Konfigurationstool später nicht korrekt gestartet werden kann. Nachdem schließlich ein geeignetes Verzeichnis ausgewählt worden ist (z.B. 'C:/Programme/ActiveTcl8.5.9.1') gelangt man durch Drücken des Knopfes 'Next' auf den nächsten Bildschirm.
4. Hier wird er Pfad für Demo-Dateien abgefragt, der per Default als Unterverzeichnis 'demos' im zuvor zugewiesenen Installationsverzeichnis angelegt wird. In aller Regel kann diese Einstellung durch Druck auf den Knopf 'Next' einfach akzeptiert werden.
5. Auf dem jetzt erscheinenden Fenster werden alle Pfade noch einmal angezeigt und die eigentliche Installation wird durch den Druck auf den Kopf 'Next' gestartet.
6. Die Installation beginnt und benötigt einige Zeit...
7. Nach Abschluss der Installation erscheint ein beiger Kasten mit der Überschrift "Serious about Programming in Tcl?" und der Knopf unten rechts in der Ecke wechselt zu 'Finish'. Nach Druck auf diesen Knopf verschwindet das Installationsfenster und die Installation ist abgeschlossen.
8. Aus dem Windows-Explorer können nun Tcl/Tk Applikationen mit der Default-Erweiterung '.tcl' direkt durch Doppelklick gestartet werden.

Tcl Zusatzpaket 'Snack' für Audio-Ausgaben

Das Zusatzpaket 'Snack' für Tcl kann direkt von der Webseite des Anbieters ‚Department of Speech, Music and Hearing‘ (KTH - Royal Institute of Technology, Stockholm, Schweden) geladen werden.

Der Download der Software ist frei. Ein direkter Link ist in der Linkliste (siehe Kap. 5) aufgeführt.

Das Zusatzpaket 'Snack' kann grundsätzlich nur nach der Installation von Tcl/Tk erfolgen, da für die Installation der Tcl/Tk Interpreter verwendet wird. Die Installation des Paketes erfolgt grundsätzlich in die Tcl/Tk Version, aus der der Tcl/Tk Interpreter beim Aufruf verwendet wird.

Für Win XP gilt:

1. Entpacken des Zip-Archivs in einen leeren Ordner
2. Start der Installation durch Ausführen der Datei 'install.tcl' im Windows-Explorer.
Achtung: Nicht die Datei 'setup.bat' verwenden, da diese fest einen Tcl/Tk Interpreter der Version 8.4x sucht und starten will!

Für WIN7 64-bit gilt:

Sieht man sich das Script "install.tcl" an, macht es lediglich das folgende:

```
[D:/Install.XP/ActiveTcl-8.5.9.0/Snack2210]> tclsh install_dbg.tcl
file mkdir C:/Program Files/ActiveTcl-8.5.9.0/lib/tcl8.5/snack2.2
file copy -force bin/windows/libsnack.dll C:/Program Files/ActiveTcl-8.5.9.0/lib/tcl8.5/snack2.2
file copy -force bin/windows/libsound.dll C:/Program Files/ActiveTcl-8.5.9.0/lib/tcl8.5/snack2.2
file copy -force bin/windows/snack.tcl C:/Program Files/ActiveTcl-8.5.9.0/lib/tcl8.5/snack2.2
file copy -force bin/windows/pkgIndex.tcl C:/Program Files/ActiveTcl-8.5.9.0/lib/tcl8.5/snack2.2
file copy -force bin/windows/snackstub22.lib C:/Program Files/ActiveTcl-8.5.9.0/lib/tcl8.5/snack2.2
file copy -force bin/windows/libsnackogg.dll C:/Program Files/ActiveTcl-8.5.9.0/lib/tcl8.5/snack2.2
file copy -force bin/windows/libsnacksphere.dll C:/Program Files/ActiveTcl-8.5.9.0/lib/tcl8.5/snack2.2
```

Man muss also nur die Dateien aus einem lokalen Verzeichnis ".../Snack2210/bin/windows" in ein Zielverzeichnis "C:/Program Files/ActiveTcl-8.5.9.0/lib/tcl8.5/snack2.2" kopieren, welches zuvor aus dem Script angelegt wird. "C:/Program Files/ActiveTcl-8.5.9.0" ist hierbei das Verzeichnis, in das TCL installiert wurde.

Für Win7 64-bit funktioniert es mit folgenden Versionen:

Active Tcl 8.5.11.1.295590-win32-x86_64threaded.exe und snack 2210-tcl.zip als Audio-Modul, wobei der Unterordner "tcl\lib\tk8.5\snack2210-tcl" heißt

Nach Abschluß der Installation ist das Paket in die Tcl/Tk Umgebung integriert und kann verwendet werden.

Sollte beim ersten Start der MoBaSbS-Konfigurationssoftware die Audio-Unterstützung abgeschaltet worden sein, kann sie über das Menu ‚Setup‘ wieder aktiviert werden.

Bei aktivierter Audio-Unterstützung ist (wenn das Häkchen für Audio-Unterstützung in den Optionen gesetzt ist) beim Verbinden mit der MoBaSbS eine kurze Tonsequenz zu hören.

12.1.2 Konfigurationstool installieren

Neuinstallation

Das MoBaSbS-Konfigurationstool wird als selbstentpackendes ZIP Archiv bereitgestellt. Die Installation erfolgt in folgenden Schritten:

1. Download des MoBaSbS Konfigurationstools (empfohlen: Speichern auf einen lokalen Medium, Festplatte)
2. Der Entpack- und Installationsvorgang wird durch ausführen der Datei 'MBSCfgSW_<version>.exe' gestartet.
Während der Installation öffnet sich eine DOS-Box in der der Installationsprozess abläuft. Alle Dateien und Unterverzeichnisse des Archivs werden in dem Verzeichnis entpackt und installiert, in dem sich das Archiv befindet!
3. Die Installation ist abgeschlossen sobald die DOS-Box geschlossen wurde.

Die Software kann durch Ausführen der Datei 'MoBaSbS.tc!' im neuen Unterverzeichnis MBSCfgSW gestartet werden.

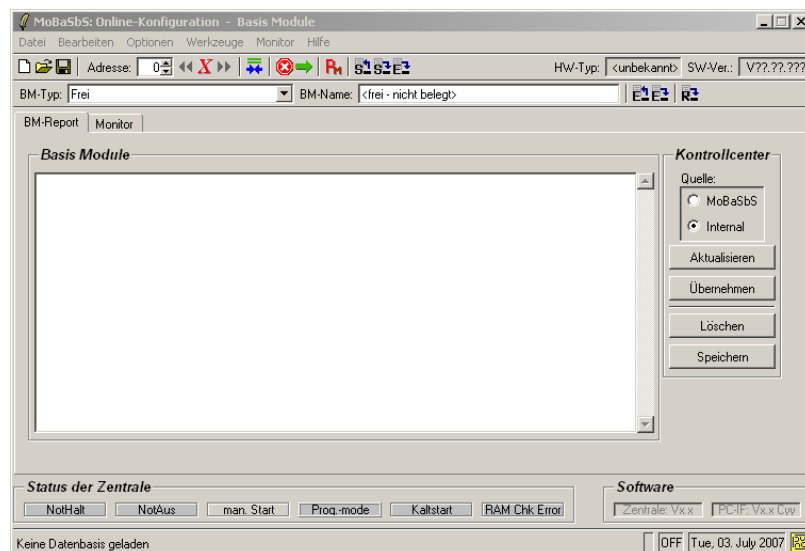
Update

Wenn nicht anders beschrieben, sollte ein Update der Software nach dem folgenden Schema erfolgen:

1. Sichern der vorhandenen MoBaSbS.ini Datei aus dem Installationsverzeichnis
2. Alte MBSCfgSW-Version sichern oder löschen
3. Neues MBSCfgSW-Version herunterladen und im gewünschten Verzeichnis ablegen
4. Entpacken der *.zip-Datei durch Doppelklick auf die Datei. Bei der Installation wird automatisch ein neues Unterverzeichnis 'MBSCfgSW' angelegt und die erforderlichen Dateien dort hinein installiert.
5. MoBaSbS.ini Datei zurückspielen
6. Konfigurationstool wie gewohnt starten und Basis-Konfiguration überprüfen (COM-Port, etc.)

12.2 Programmstart

Nach dem Start des Programms erscheint das unten abgebildete Fenster. Man erkennt im oberen Teil die Menüleiste, darunter die Werkzeugleisten 1 (allgemeine Funktionen) und 2 (je nach Ansicht mit Funktion für Basis- oder Peripherie-Module). Im mittleren Bereich des Fensters befindet sich das Feld mit den verschiedenen ‚Registerkarten‘, die die Masken für die verschiedenen MoBaSbS-Module sowie den Monitor enthalten. Darunter befinden sich diverse Status-Informationen der Zentrale sowie die Software-Versionen (IFC und BMC). Am unteren Fensterrand befindet sich die Status-Zeile des Konfigurationstools, in der verschiedene Informationen während der Arbeit mit dem Programm angezeigt werden. Diese enthält auch einen ‚Progress-Bar‘, der den Fortschritt von länger andauernden Operationen visualisiert.



Start-Ansicht

Icons und Funktionen



Standard Windows Icons für 'Neu', 'Öffnen' und 'Speichern'; bezieht sich ausschließlich auf die interne Datenbasis.



Springe zur nächsten definierten Adresse 'abwärts' bzw. 'aufwärts'; gilt sowohl für die Basis-Modul-Ansicht als auch die Peripherie-Modul-Ansicht



Lösche das Modul unter der aktuellen Adresse; gilt sowohl für die Basis-Modul-Ansicht als auch die Peripherie-Modul-Ansicht



MoBaSbS 'Verbinden' bzw. 'Verbindung lösen' (Icon wechselt die Ansicht je nach aktuellem Zustand)



Kommando 'Alles Aus' an MoBaSbS senden



Kommando 'Alles Ein' an MoBaSbS senden



Wechsel in die Peripherie- bzw. Basis-Modul-Ansicht



Modulspezifische Daten aus SRAM lesen (BM und PM)



Modulspezifische Daten in SRAM schreiben (BM und PM)



Modulspezifische Daten aus EEPROM lesen (BM und PM)









Modulspezifische Daten in EEPROM schreiben (BM und PM)

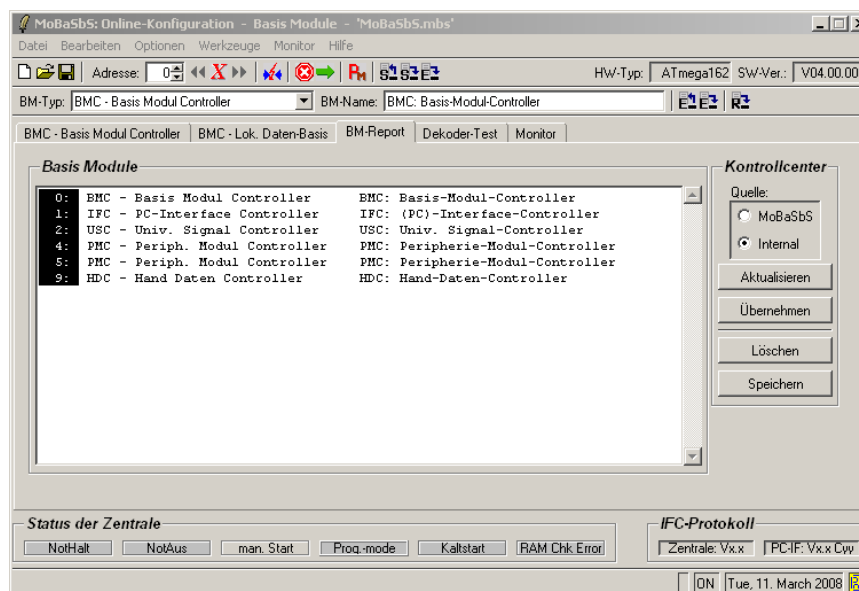


Software-Reset für aktuelles Modul auslösen (nur BM)

Die nachfolgende Tabelle gibt eine Übersicht über die ab Version 4.30 eingeführten Schaltsymbole. Diese korrespondieren mit den Symbolen in Traincontroller 7.

	Ausgang 1, Schaltbefehl für links und rechts, nach Richtung in die die Spitze des blauen Dreiecks zeigt
	Ausgang 2, Schaltbefehl für links und rechts, nach Richtung in die die Spitze des blauen Dreiecks zeigt
	Ausgang 3, Schaltbefehl für links und rechts, nach Richtung in die die Spitze des blauen Dreiecks zeigt
	Ausgang 4, Schaltbefehl für links und rechts, nach Richtung in die die Spitze des blauen Dreiecks zeigt
	Löst eine Sequenz von Schaltbefehlen aus: Ausgang 1 rechts, danach Ausgang 4 links
	Modusabhängig gesperrte Taste (z. B. LDEC-Uni)

Existiert bereits eine Datei für die gesamte MoBaSbS-Konfiguration ('MoBaSbS.mbs' im Unterverzeichnis 'data'), so wird diese beim Start automatisch geladen. Im Startbildschirm erscheint der BM-Report mit allen bekannten und gespeicherten BM-Modulen:

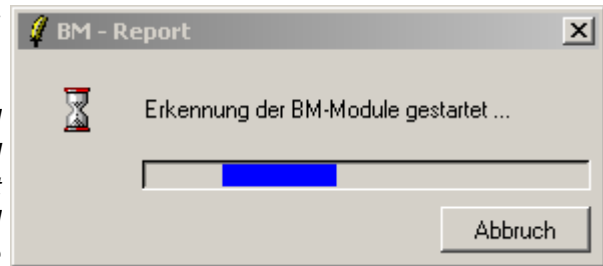


Start-Ansicht mit bereits gespeicherten Modulen

Wichtige Bedienungshinweise

1. Während die blaue Fortschritts-Anzeige in der Status-Zeile des Programms sichtbar ist, dürfen keine weiteren Aktionen gestartet oder andere Knöpfe gedrückt werden.

Das Programm ist in Tcl/Tk erstellt, was grundsätzlich eine Parallelverarbeitung von verschiedenen Aufgaben erlaubt, sofern diese unabhängig voneinander sind (Stichwort 'multi-threaded'). Sinn und Aufgabe des Konfigurationstools ist es, mit der MoBaSbS zu kommunizieren und Antworten auf Anfragen zu verarbeiten. Die modulare Struktur der Software erfordert ein striktes FiFo-Prinzip (FiFo = 'First in First out', oder in Deutsch 'Wer zuerst kommt, mahlt zuerst'), d.h. stellt ein Modul einen Request an die MoBaSbS-Hardware wird die nächste Antwort an dieses anfragende Software-Modul weitergeleitet. Greifen nun mehrere Abläufe ineinander, ist diese zwingend notwendige Abhängigkeit nicht mehr gegeben und die Stabilität der Software ist stark in Frage gestellt. Aus diesem Grund wird gerade bei Operationen mit intensiver und umfangreicher Hardware-Kommunikation der blaue Fortschritts-Balken eingeblendet, um den Anwender von übereilten Aktionen abzuhalten



2. Nach einem vermeintlichen Software-Fehler (Applikations-Fehler) bitte das Konfigurationstool schließen und neu starten.

Tcl/Tk stürzt nicht wie andere Programme bei einem Fehler einfach ab, sondern versucht dem Anwender den Fehler zu erklären und ihm die Möglichkeit der Behebung zu geben. Dieses kann sowohl als Vorteil als auch als Nachteil gesehen werden. Von Vorteil ist, dass die Software nicht unbedingt neu gestartet werden muss und man weiterarbeiten kann. Nachteilig ist aber die Tatsache, dass bei einem Fehler die Abarbeitung des Ablaufs (Threads), bei dem der Fehler aufgetreten ist, unterbrochen wird. Damit werden nicht alle erwarteten Operationen korrekt und komplett ausgeführt und das Programm ist möglicherweise instabil geworden, auch wenn die äußerlich intakte Oberfläche dieses nicht vermuten lässt. Darum bitte nach einem Applikations-Fehler immer das Konfigurationstool schließen und neu starten.

12.3 Menüs und Werkzeugleisten

12.3.1 Menüs

Die folgende Tabelle enthält eine Übersicht über die zentrale Menustruktur und deren Unterpunkte.

Datei	Neu	Interne Datenbasis löschen und neu initialisieren
	Öffnen	Datenbasis aus Datei laden
	Speichern	Geladene Datenbasis speichern
	Speichern unter	Aktuelle Datenbasis unter neuem Namen speichern
	Beenden	beendet das MoBaSbS-Konfigurationstool
	Bearbeiten	BM-Daten löschen
PM-Daten löschen		löscht nur den für Peripherie-Module relevanten Teil der aktuell geladenen Datenbasis
Zähler zurücksetzen		Setzt alle Zähler des aktuellen Moduls (aktuelle Adresse) zurück (z. Zt. nur Encoder)
Default-Konfiguration		Stellt Default-Einstellungen für das aktuelle Modul (aktuelle Adresse) wieder her (BM und PM)
SW-Reset		Löst für das aktuelle Modul (aktuelle Adresse) einen SW-Reset aus (BM und PM)
Optionen		<SW-Konfiguration>
Werkzeuge	Basis-Module	Werkzeuge speziell für Basis-Module
	BM-Adresse ändern	Adresse eines BM-Moduls ändern (nur für Adressen größer 3!)
	BM-Selektiver Lok-Daten Import	Selektiver Import von Lok-Daten aus einer externen Datei in die interne Lok-Datenbasis
	BM-Selektiver Lok-Daten Export	Selektiver Export von Lok-Daten in eine externe Datei aus die interne Lok-Datenbasis
	BM-Label anzeigen	liest interne Informationen des Basis-Moduls aus (SW-Version, etc.)
	Peripherie-Module	Werkzeuge speziell für Peripherie-Module
	PM-Adresse ändern	Adresse eines PM-Moduls ändern
	PM-Adresse ändern neu	Adresse eines PMC- und/oder PM-Moduls ändern
	PM-Modultyp ändern	Typ eines PM-Moduls ändern (z. Zt. nur in Bezug auf Rückmeldefähigkeit)
	PM-Kontakt Übersicht	Übersicht über alle Rückmelde-Kontakte öffnen (separates Übersichtsfenster)
	Lok Kontroll Panel	steuern von Lokomotiven direkt aus dem Config-Tool
	DCC-Programmer	Programmierung der CVs
	UC-Speicher-Dump	auslesen des Eeprom und SRAM von BM/PM
Monitor	Anwender Kommando 1-8	Anwender Kommando definieren, öffnet separates Bearbeitungsfenster (1-8)
	Monitor Log speichern	Speichert den aktuellen Monitor Log in einer ASCII-Datei
	Monitor Log löschen	Löscht das aktuelle Monitor Log Textfeld
Hilfe	Konsole ein	Tcl/Tk Konsole einblenden (nur für Debug- und LowLevel Aktionen zu öffnen)
	Konsole aus Über	Tcl/Tk Konsole ausblenden Zeigt Version und Datum der Release an

12.4 Konfiguration

12.4.1 Werkzeugleiste 1

Im Menu 'Optionen' kann die Werkzeugleiste aktiviert werden. Es wird empfohlen, sie zu aktivieren.



Die Werkzeugleiste 1 ist für die zentrale Navigation und einige globale Funktionen konzipiert. Am linken Rand befinden sich die Standard-Knöpfe für 'Neu', 'Laden' und 'Speichern' (vgl. Menu 'Datei'). Daran rechts anschließend befindet sich das Feld zur Navigation bzw. Adress-Auswahl. Je nach Ansicht kann eine Adresse aus dem zulässigen Raum für Basis-Module (0-31) oder Peripherie-Module (0-255) gewählt werden.

Dieses kann durch direktes Eintippen im Feld Adresse oder durch betätigen der kleinen Pfeile am rechten Rand des Feldes geschehen (incr. bzw. decr. jeweils um 1). Ungültige Eingaben werden nicht akzeptiert und mit einem Signalton quittiert. Mit den Knöpfen bzw. kann schnell zur nächsten Adresse gesprungen werden. Bei nur insgesamt einem für diese Ansicht definierten Modul erscheint eine Fehlermeldung. Ein Druck auf den Knopf löscht nach Rückfrage die aktuelle Adresse und markiert diese wieder als 'Frei'.

Über die Knöpfe bzw. kann eine Verbindung hergestellt bzw. gelöst werden. Das Bild des Symbols ändert sich je nach Verbindungszustand.

Für einen eventuellen Notfall sind in dieser Leiste auch Knöpfe für die Befehle 'Alles Aus' () bzw. 'Alles An' () definiert.

Die drei folgenden Knöpfe bewirken das Lesen bzw. Schreiben der Konfigurationsdaten des unter 'Adresse' eingestellten Moduls. Hierbei lesen und schreiben die 'S'-Knöpfe die Daten aus dem SRAM des BMC. Mit dem 'E'-Knopf werden alle Einstellungen in das Eeprom eines Peripheriemoduls geschrieben.

Am rechten Rand dieser Werkzeugleiste befinden sich noch die Angaben zum HW-Typ der aktuellen Adresse und der Softwareversion, sofern anwendbar (d.h. Adresse einem Modultyp zugewiesen und bereits aus dem Modul ausgelesen)

12.4.2 Werkzeugleiste 2

Im Menu 'Optionen' kann die Werkzeugleiste aktiviert werden. Es wird empfohlen, sie zu aktivieren. Die Werkzeugleiste 2 wechselt je nach eingestellter Ansicht.

Für die Basis-Module enthält sie im linken Teil das Auswahl-Menü für den Modultyp sowie den Modul-Namen (Descriptor). Der Descriptor kann über den Knopf gelesen werden. Er kann dann nach Belieben geändert (max. 32 Zeichen) und über den Knopf geschrieben werden. Mit Hilfe des Knopfes kann ein Reset am Basis-Modul ausgelöst werden.



Für die Peripherie-Module enthält sie im linken Teil das Auswahl-Menü für den PM-Modultyp sowie den Modul-Namen, der über den Knopf gelesen und über den Knopf geschrieben werden kann. Auf der rechten Seite werden im Feld 'PMC' die Adresse des PMC-Moduls angezeigt, an dem das aktuelle eingblendete Peripherie-Modul angeschlossen ist. Das Feld 'Str.' zeigt den zugehörigen Strang des PMC-Moduls an.

12.4.3 Audio-Unterstützung

Im Menü 'Optionen' kann die Audio-Unterstützung ein- bzw. abgeschaltet werden. Sie gilt für alle im Programm verfügbaren Audio-Ausgaben.

12.4.4 Autom. MBS laden

Mit der Option wird festgelegt, ob beim Start des MoBaSbS-Konfigurationstools eine bereits gespeicherte Konfiguration (*.MBS-Datei) automatisch geladen werden soll.

12.4.5 Konfig. ComPort (Schnittstelle zur MoBaSbS)

RS232

Für die serielle Kommunikation muss der serielle Port und die Geschwindigkeit eingestellt werden. Dazu öffnet sich über Menü 'Optionen > Konfig. ComPort' das nebenstehende Fenster mit den beiden Einstellungs-Optionen als Pull-Down-Menu-Feld. Bei Nutzung einer RS232-Schnittstelle muss nach der Auswahl des Ports (COMx) die erwartete Geschwindigkeit eingestellt werden.

Die Konfig-Software kann nur COM-Ports bis einschl. COM7 verwalten, ggf. müssen die COM-Ports über die Windows-System-Verwaltung neu nummeriert werden.



Weitere Hinweise zur Installation siehe Kap. 7.3.

Über das Menü 'Optionen > USB Mode' ist der USB Mode auszuschalten. Der Status ist mit den Dip1-Switch Einstellung des IFC abzugleichen.

USB

Für die Konfiguration der USB-Schnittstelle mit FTDI-Treiber werden beim Start des MoBaSbS-Konfigurationstools alle auf dem PC verfügbaren seriellen Ports ermittelt und in einer Auswahlliste aufgenommen (Sortierung alphabetisch absteigend). Sind FTDI Treiber auf dem PC installiert **und** diese mit der **MoBaSbS-Kennung** versehen, werden sie erkannt und der Name des seriellen Ports, der via FTDI-Treiber mit der MoBaSbS-Hardware verknüpft ist, wird durch 'USB-IF MoBaSbS V4.0' ersetzt. Dieser geänderte Eintrag erscheint gemäß Sortierung direkt am Anfang der Liste. Wird dieser Port ausgewählt, wird die Geschwindigkeit sofort auf '<automatisch>' gesetzt.

Über das Menü 'Optionen > USB Mode' ist der USB Mode einzuschalten. Der Status ist mit den Dip1-Switch Einstellung des IFC abzugleichen.

Nach der Konfiguration der Schnittstelle kann die Verbindung mit der MoBaSbS hergestellt werden. Bei erfolgreichem Verbindungsaufbau erscheint in der Status-Zeile für den BMC die Version von BMC-Software und IFC, sowie in der Programm-Status-Zeile der Hinweis 'ON'.

12.4.6 Autom. Verbinden

Im Menü 'Optionen' kann dieser Punkt aktiviert werden. Es wird dann bei Start des Konfigurationstools automatisch eine Verbindung zur MoBaSbS hergestellt.

12.4.7 Erw. ComPort Optionen

Im Menu 'Optionen' kann klappt unter dem Menüpunkt 'Erw. ComPort Optionen' ein Untermenü auf, das folgende Punkte enthält:

- **Zus. Synchronisierung ein/aus**
Ist dieser Modus aktiviert werden jedem Befehl von und zur MoBaSbS zwei weitere Bytes vorangestellt: 0xFF 0xFE. Diese dienen zur besseren Synchronisation und enthalten keine weiteren Informationen. Es ist wichtig, das IFC und Konfigurationstools immer gleich konfiguriert sind, um einen Verbindungsaufbau zu ermöglichen (Stellung von DIP-Switch 2 auf IFC beachten, siehe Kap. 7.3). Diese Option ist per Default nicht gesetzt. Sie wird in der MoBaSbS.ini Datei dauerhaft gespeichert.
- **Autom. COM-Port Detektion**
Dadurch wird beim Starten der SW die Windows Registry untersucht und für den Begriff COM3 die aus dem USB-Interface erkannte Kennung eingesetzt (siehe auch Abschnitt 12.4.5). Bei Problemen mit den FTDI-Treibern (z.B. fehlerhafte oder unvollständige Installation, evtl. 2 IFC mit USB gleichzeitig in Betrieb) kann das Abschalten der automatischen COM-Port Detektion sehr hilfreich sein. Nach dem Abschalten werden 15 COM-Ports zur Auswahl vorgegeben und es obliegt dem Anwender, den richtigen (USB oder RS232) auszuwählen). Diese Option wird in der MoBaSbS.ini Datei dauerhaft gespeichert.
- **Slow Port Modus**
Wenn diese Option aktiviert ist, lässt sich das Konfigurationstool bei einem Verbindungsaufbau deutlich mehr Zeit zwischen den einzelnen Befehlen. Diese Option macht nur Sinn, wenn mit einer langsamen RS232-Verbindung gearbeitet wird. In Verbindung mit einem langsamen PC kann dann sicher eine Verbindung zur MoBaSbS aufgebaut werden. Bei Verwendung der USB-Schnittstelle sollte diese Option nicht aktiviert sein. Diese Option wird in der MoBaSbS.ini Datei dauerhaft gespeichert.

12.4.8 Lok-Datenbank

Unter dem Menüpunkt 'Lok-Datenbank' klappt ein Untermenü auf, das zurzeit die folgenden Konfigurationsmöglichkeiten bietet:

Konfig. Hersteller
Konfig. Funktionstyp

Unter dem Punkt 'Konfig. Hersteller' öffnet sich ein Fenster, in dem 16 Herstellernamen frei konfiguriert werden können. Die Länge des Namens ist auf max. 20 Zeichen begrenzt. Für den zweiten Punkt gilt ähnliches, jedoch können hier bis zu 16 Funktionstypen frei definiert werden. Die Länge der Bezeichnung beträgt hierbei max. 25 Zeichen.

12.4.9 DCC Programmierer Konfiguration

Im Menu 'DCC Programmierer' können die folgenden Parameter eingestellt werden:

- Lange Adressen verwenden
- CV Schreiben verifizieren

Ist der erste Punkt 'Lange Adressen verwenden' aktiviert, ist das Feld für das Editieren der CV1718 (Lange Lok-Adresse) immer zum Editieren für den Decoder-Typ 'einfach' (Default-Einstellung) freigegeben. Ist der untere Punkt 'CV-Schreiben verifizieren' aktiviert, liest die Konfigurationssoftware nach jedem Schreiben die gleiche CV sofort bitweise aus dem Decoder zurück und vergleicht das Ergebnis. Diese Funktion kann bei Bedarf eingeschaltet werden, wenn ein Bit-Vergleich nach dem Schreiben erwünscht ist. Per Default vergleicht die DCC-Hardware nach dem Schreiben einer CV das komplette 8-bit Wort.

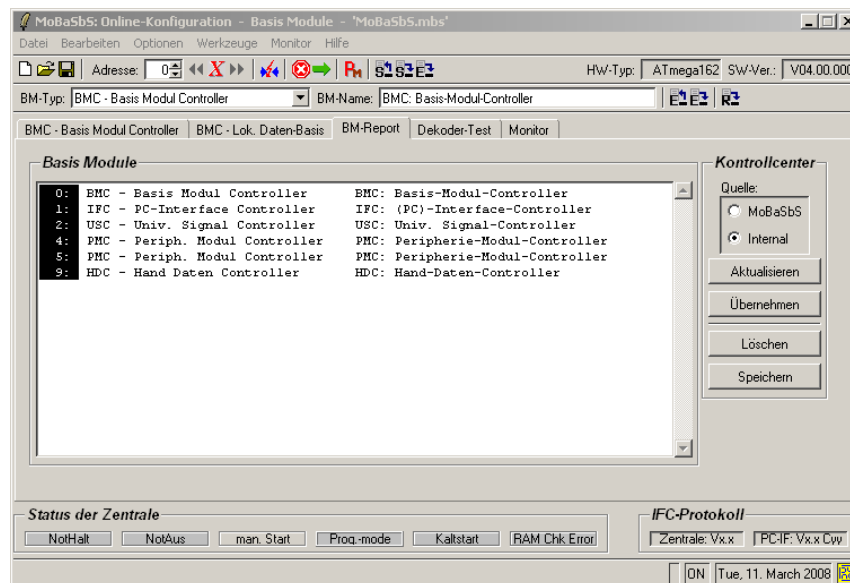
12.4.10 Debug

Unter dem Menüpunkt 'Debug' klappt ein Untermenü auf, das die folgenden Punkte enthält:

- LoopBack Modus
- CmdLoopBack Modus
- RawData Modus
Für den normalen Betrieb und die Anwendung ist keine dieser drei Optionen zu setzen. Diese Optionen können auch nicht in der MoBaSbS.ini Datei gespeichert werden und sind bei jedem Neustart grundsätzlich deaktiviert.
- LokNamen ASM - Volle Liste
Mit Hilfe dieser Option können LokNamen-Dateien aus älteren Versionen in die Lok-Datenbank der Version 4 importiert werden. Diese Option wird in der MoBaSbS.ini Datei dauerhaft gespeichert.
- Ext. Hex-Monitor
- Autom. Ext. Hex-Monitor öffnen
Für den normalen Betrieb und die Anwendung ist keiner der beiden Menüpunkte zu verwenden. Für einfacheres Debuggen ist zusätzlich zum integrierten Hex-Monitor ein externer Hex-Monitor integriert worden, der als eigenständiges Fenster ausgelegt ist und somit immer parallel zum Hauptfenster sichtbar gehalten werden kann. In Aussehen, Inhalt und Funktion entspricht es exakt dem internen Hex-Monitor. Wird der Monitor nachträglich geöffnet, erscheinen alte Einträge des integrierten Monitors in schwarz, neue werden dagegen wie gewohnt farbig dargestellt. Mit der unteren Option kann ausgewählt werden, ob das ext. Hex-Monitor-Fenster automatisch beim Start geöffnet werden soll. Diese Option wird in der MoBaSbS.ini Datei dauerhaft gespeichert.
- Autom. Konsole öffnen
Wie das ext. Hex-Monitor-Fenster kann auch bei Bedarf das Konsolen-Fenster autom. beim Start geöffnet werden, sofern dieser Menüpunkt markiert ist. Diese Option wird in der MoBaSbS.ini Datei dauerhaft gespeichert.

12.5 Basis-Module

12.5.1 BM-Report



BM-Report

Nach dem Start befindet sich das Konfigurationstool in der Ansicht 'Basis-Module' (ein Wechsel der Ansicht ist über das Menu 'Ansicht' jederzeit möglich). Im Register 'BM-Report' kann auf der rechten Seite die Quelle ('MoBaSbS'=Hardware, 'Internal'=interne Datenbasis) ausgewählt werden. Mit dem Knopf 'Aktualisieren' werden dann die Daten der ausgewählten Quelle aktualisiert. Bei 'Internal' wird der momentane Inhalt der Datenbank ausgegeben, bei 'MoBaSbS' (falls verbunden) werden die BM-Module gescannt und angezeigt.



Dabei werden die gefundenen Informationen nicht automatisch in die interne Datenbasis übernommen, sondern lediglich im Report-Fenster angezeigt. Für die Übernahme muss explizit der Knopf 'Übernehmen' gedrückt werden, der aus diesem Grund nur zur Verfügung steht, wenn als Quelle 'MoBaSbS' ausgewählt ist.

Bei der Übernahme werden alle vorhandenen Informationen zu BM-Modulen mit den gerade gelesenen Daten überschrieben. Das sind ausschließlich Adresse, BM-Typ, BM-HW, BM-SW-Version sowie der Descriptor. Alle weiteren Informationen wie VFP Konfiguration beim HDC, BMC-Lok-DB, etc. werden überschrieben /gelöscht und müssen neu aus der MoBaSbS zurück gelesen werden

Als Hinweis wird der nicht gespeicherte Report als rote Schrift auf grauem Grund dargestellt.

Die Adresse in der linken Spalte als 'Link' ausgelegt. Durch Anklicken des Links wird zu der entsprechenden Adresse gewechselt und das entsprechende Register für das Basis-Modul eingeblendet.

Mit Hilfe der beiden Knöpfe 'Löschen' und 'Speichern' kann das aktuelle Analyse-Ergebnis gelöscht oder in einer ASCII-Datei zwecks Archivierung gespeichert werden.

Für alle Basis-Module (einzige Ausnahme: IFC) kann ein beliebiger BM-Name (Descriptor) vergeben werden. Die Länge ist auf max. 32 Zeichen beschränkt. Auslesen und Speichern erfolgt über die Knöpfe  bzw. .

Die Zuweisung von Basis-Modulen ist beschränkt und erfolgt gemäß der nachfolgenden Tabelle.

BMC	0	zwingend notwendig
IFC	1	zwingend notwendig für das erste IFC
USC	2 - 30	empfohlen: 2 bis 3
WMC	6	(wenn vorhanden)
PMC	2 - 6	empfohlen: 4 bis 6
DPC	2 - 30	empfohlen: 4 bis 7
HDC	2 - 30	empfohlen: > 8 (die Einstellung der Adresse direkt über den HDC lässt nur Werte größer oder gleich 8 zu)
weitere	4 - 30	



Ein neu programmiertes BM oder HDCs haben per Default die Adresse 31. Es darf deshalb immer nur ein neues BM bzw. HDC in das Gesamtsystem aufgenommen werden. Bei diesem muss erst die Default-Adresse geändert werden, bevor ein weiteres BM oder HDC gesteckt werden kann. Dieses Vorgehen gilt auch für die Konfiguration der PMs mit der Adresse 255

Der PMC steht ebenfalls per Default auf Adresse 31 und wird daher nicht vom BMC angesprochen, da der BMC per Default 'nur' bis Adresse 9 pollt. Über Werkzeuge > Basismodule > BM-Adresse ändern kann die Adresse geändert werden. Erst dann geht die X-Bus LED am PMC an. Auch bei mehreren HDC beachten (Adressraum 9, 10, 11, etc).

Über Werkzeuge > Basismodule > BM-Adresse ändern können auch die anderen Adressen geändert werden.

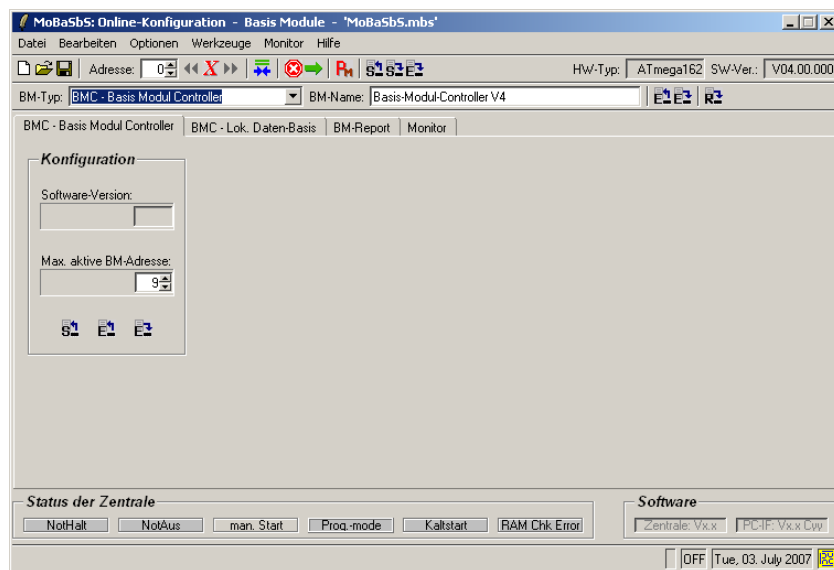
Die Konfiguration kann mit Hilfe des Knopfes 'Übernehmen' nach Beantwortung der Sicherheitsabfrage in die interne Datenbasis übernommen werden.

Will man eine bereits gespeicherte Datenbasis (*.mbs) laden, so muss dies vor einer Aktualisierung geschehen. Also MoBaSbS und Tool starten -> Datenbasis (sofern vorhanden) einlesen -> BM-Report -> Aktualisieren -> Übernehmen. Jetzt stehen alle Daten aus der gespeicherten Datenbasis und die frisch ermittelten Werte zur Verfügung.

Nach erfolgreicher Übernahme wird das Register des BMC sichtbar. Mit den Navigationstasten  und  kann jetzt zwischen allen definierten Basis-Modulen gewechselt werden.




Die Adresse in der linken Spalte ist als ‚Link‘ ausgelegt. Durch Klicken auf die entsprechende Adresse wechselt das Register zum jeweiligen Basismodul und die verschiedenen Informationen und Konfigurationseinstellungen werden sichtbar.

12.5.2 BMC

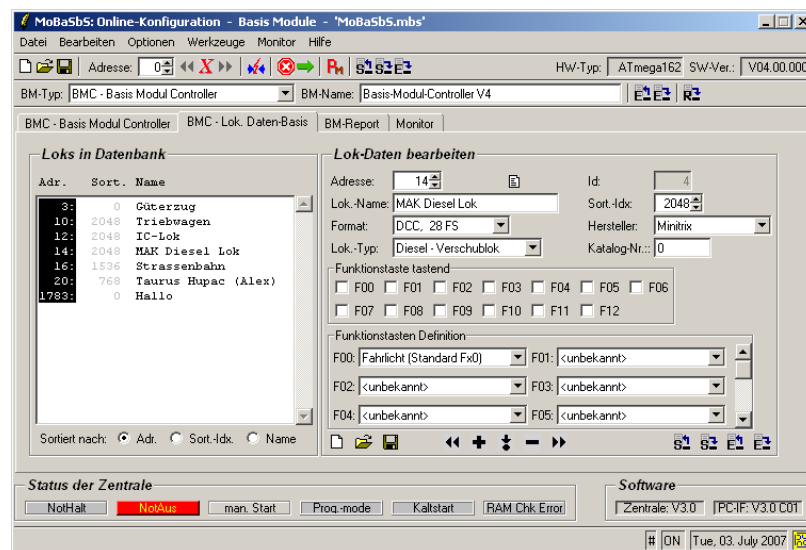


Register BMC

Neben dem frei wählbaren BM-Namen (Descriptor) kann die höchste aktive Adresse definiert werden, die der BMC während eines Abfragezyklus abfragt. Der Default-Wert ist 9. Dies ist bei der Adressvergabe von HDC und PMC zu berücksichtigen.

Über die Knöpfe  und  kann die aktuelle Konfiguration aus dem RAM bzw. EEPROM gelesen werden, mit  wird ein neuer Wert ins EEPROM geschrieben, der dann erst nach einem HW-Reset wirksam wird.

12.5.3 BMC Lok. Daten-Basis



BMC Lok. Daten-Basis

Mit dem Konfigurationstool ist es möglich, seine Lok-Daten-Basis komfortabel zu verwalten und zu speichern. Dieses erfolgt im Normalfall in der globalen Sicherungsdatei (Endung '.mbs'), es besteht aber auch die Möglichkeit, die Lok-Daten-Basis in einer separaten Datei (Endung '.mbdb') unabhängig zu speichern und ggf. zu jeder beliebigen MoBaSbS-Konfiguration hinzu zu laden. Über das Menu 'Tools' ist ebenfalls ein selektiver Export sowie ein selektiver Import möglich, über den Teile der Lok-Datenbasis in eine Datei exportiert und dann in eine andere MoBaSbS importiert werden können.

Mit der MoBaSbS können nur die Loks angesprochen werden, die auch über das MoBaSbS-Konfigurationstool definiert wurden.

Das Register teilt sich in zwei Bereiche auf: auf der linken Seite die Übersicht über alle in der Datenbank enthaltenen Loks (Überschrift: Loks in Datenbank) und auf der rechten Seite die Datenfelder mit den Funktionsknöpfen am unteren Rand (Lok-Daten bearbeiten).

Loks in Datenbank

Das Textfenster mit den aktuell in der Datenbank vorhandenen Loks gliedert sich in drei Spalten:

Adr. Adresse der Lok
Sort. Sortierungsindex der Lok
Name Name der Lok

Durch die drei Auswahlmöglichkeiten am unteren Rand des Fensters können die Loks in der Ansicht sortiert werden. Neben der Sortierung nach Name und Adresse kann nach dem Sortierungsindex sortiert werden. Er ist eine vom Anwender zu vergebende Zahl im Bereich der möglichen Adressen, mit dessen Hilfe man verschiedene Lokomotiven unter einem anderen Gesichtspunkt sortieren kann (z.B. Sortierungsindex zwischen 100 und 200 bezeichnet alle E-Loks, etc.).


Eine Lok aus der Datenbank kann durch Anklicken auf die Adresse zum Bearbeiten im rechten Fenster ausgewählt werden. Sämtliche zurzeit gespeicherten Parameter werden nach dem Anklicken in den entsprechenden Feldern angezeigt.


Lok Datenbank bearbeiten





In dieser Maske können die Daten einer Lok bearbeitet werden und danach wieder in der internen Datenbank gespeichert werden.



Die Grundlage für eine neue Datenbank kann durch das Importieren von vorhandenen Lok-Namen-Dateien (Erweiterung '.mbl') oder vorhandener ASM-Dateien mit Lok-Namen geschaffen werden. Dabei ist das Format zu beachten und ggf. korrekt einzustellen (Menu 'Optionen > Debug > LokNamen ASM - volle Liste'). Alle importierten Lok-Daten erscheinen dann im linken Textfenster und können durch Anklicken der Adresse zum Editieren ausgewählt werden. Beim Anklicken werden die für die gewählte Adresse gespeicherten Daten in den Feldern im rechten Teil des Registers angezeigt.

Minimale Daten, die für den Betrieb zwingend erforderlich sind, sind die Adresse der Lok und das Format. Der Name ist hilfreich für die Steuerung über das HDC, aber nicht unbedingt erforderlich. Lokomotiven werden durch die Adresse unterschieden und können gleiche Namen haben.

Achtung: Nach erfolgter Änderung müssen alle getätigten Änderungen in den Feldern in die Datenbank des Konfigurationstools zurück geschrieben werden, um sie dauerhaft zu speichern. Dieses erfolgt durch Drücken des Knopfes . Wurde die Lok-Adresse eines bestehenden Datensatzes geändert, erscheint ein Hinweis mit den Optionen 'Neuanlegen', 'Ändern' oder 'Abbrechen'. 'Neuanlegen' erzeugt einen neuen Lok-Datensatz mit der geänderten Adresse, 'Ändern' ändert die Adresse des aktuellen Datensatzes ab und Abbrechen sowie das Schließen des Hinweisfenster verändert nichts.

Hinweis: *An Hand der ID kann man erkennen, ob ein neuer Datensatz angelegt wird oder ob ein vorhandener geändert wird. Solange als ID der Wert 0 steht, wird auch ohne vorheriges Drücken des Knopfes  ein neuer Datensatz angelegt. Ist der Wert ungleich 0 wird der Datensatz mit der entsprechenden ID verändert, also in aller Regel der zuletzt durch Klicken im linken Fenster zum Bearbeiten ausgewählte Datensatz.*













Mit Hilfe des Knopfes  kann ein leerer Datensatz eingefügt (Nachbearbeiten und Speichern erforderlich) bzw. mit dem Knopf  gelöscht werden.  bzw.  wählen den jeweils nächsten Datensatz gemäß aktueller Sortierung zum Bearbeiten aus.

Neben den sichtbaren Feldern können zusätzlich für jede Lok noch in einem separaten Textfenster weitere persönliche Infos abgelegt werden. Dazu muss zunächst die Lok zum Bearbeiten angewählt werden (klicken auf die Adresse im linken Übersichtsfenster -Feldinhalte werden aktualisiert) und anschließend auf das Symbol  oben zwischen den Feldern 'Adresse' und 'Id' geklickt werden. Es öffnet sich ein Textfenster, in dem beliebige weitere Zusatzinformationen abgelegt werden können, für die bislang kein Feld vorgesehen ist. Diese Informationen werden wie alle anderen Änderungen erst mit Drücken des Kopfes  in der Datenbank des Konfigurationstools gespeichert.



Übersicht über die Datenbankfelder



Feld	Typ	Beschreibung
Adresse	Zahl, Bereich 0-9999	Lok-Adresse wie im Decoder programmiert,
Id	Zahl	Datensatznummer, nicht veränderbar
Lok.-Name	Text	Name der Lok (max. 32 Zeichen, Sonderzeichen erlaubt)
Sort.-Idx	Zahl, Bereiche 0-9999	Sortierindex; vom Anwender frei zu vergebende Zahl, nach der sortiert werden kann
Format	Liste	Decoder-Format mit Fahrstufen
Hersteller	Liste	Hersteller
Lok.-Typ	Liste	Typ der Lok, Klassifizierung
Katalog-Nr.	Zahl mit 6 Ziffern	Katalog-Nummer
Funktionstaste tastend	Markierung	Funktionstaste hat tastende Funktion, je möglicher Funktionstaste ein Markierung
Funktionstasten Definition	Liste	Art und Funktion der jeweiligen Funktionstaste, für jede Funktionstaste vorgesehen (scrollbares Feld, je zwei Funktionstasten in einer Zeile dargestellt), Eintragungen werden derzeit von MoBaSbS nicht genutzt

Übersicht über die Icons im Register BMC Lok-Daten-Basis

Icon	Bedeutung und Funktion
Standard Funktionen für Speichern und Lesen von Dateien	
	Neue (leere) Datenbank erzeugen erzeugt eine neue und leere Datenbank im Speicher des Konfigurationstools
	Datenbank öffnen bzw. importieren über dieses Icon kann eine komplette Lok-Datenbank, die zuvor als einzelne Datei separat von anderen Konfigurationsdaten gespeichert wurde, eingelesen werden (Erweiterung: .mbdb). Zusätzlich besteht hier die Möglichkeit, vorhandene Informationen über Loks (z.B. MoBaSbS Lok-Namen Datei, Erweiterung: .mbl, oder ASM-Datei) einzulesen. Das Konfigurationstool legt dabei für jede gefundene Lok einen neuen Datensatz an, das anschließend vom Anwender bearbeitet werden muss.
	Datenbank komplett exportieren exportiert eine komplette Datenbank zwecks Archivierung oder Übertragung auf ein anderes Hardware-System.
Funktionen zum Bearbeiten und Navigieren in der Datenbank	
	Wähle vorhergehenden Datensatz wählt den vorhergehenden Datensatz gemäß aktuell gewählter Sortierung Achtung: Ungespeicherte Änderungen des aktuellen Datensatzes gehen dabei unwiderruflich verloren!
	Neuen (leeren) Datensatz in die Datenbank einfügen erzeugt einen neuen und leeren Datensatz in der Datenbank, der danach mit den minimal notwendigen Information versehen und gespeichert werden muss. Notwendige Informationen sind Adresse, Name und Format, alle anderen Felder sind optional und dienen lediglich der Information.
	Änderungen im aktuellen Datensatz speichern speichert die aktuell angezeigten Änderungen im Datensatz durch Überschreiben. Wurde die Adresse in einem bestehenden Datensatz geändert, kann diese geändert oder eine neue Lok unter der geänderten Adresse angelegt werden.
	Löscht den aktuell angezeigten Datensatz nach Bestätigung der aktuelle Datensatz wird dadurch unwiderruflich gelöscht.
	Wähle nachfolgenden Datensatz wählt den nachfolgenden Datensatz gemäß aktuell gewählter Sortierung Achtung: Ungespeicherte Änderungen des aktuellen Datensatzes gehen dabei unwiderruflich verloren!
Funktionen zum Austausch der Datenbank mit der MoBaSbS Hardware	
	SRAM lesen liest die z. Zt. im BMC vorhandene Datenstruktur aus und erzeugt eine neue Datenbasis basierend auf den gelesenen Daten. Achtung: Datenbank in dem Konfigurationstool wird dabei komplett überschrieben!
	SRAM schreiben schreibt die aktuelle Datenbank des Konfigurationstools in das SRAM des BMC Achtung: Datenbank im SRAM des BMC wird dabei komplett überschrieben!
	Seriellles BMC EEPROM schreiben überträgt die Datenbank im BMC SRAM dauerhaft in den ext. EEPROM Speicher auf dem BMC Achtung: Datenbank im SRAM des BMC wird dabei komplett überschrieben!
	Seriellles BMC EEPROM auslesen über trägt die Datenbank aus dem ext. EEPROM Speicher in das SRAM des BMC Achtung: Datenbank im SRAM des BMC wird dabei komplett überschrieben!

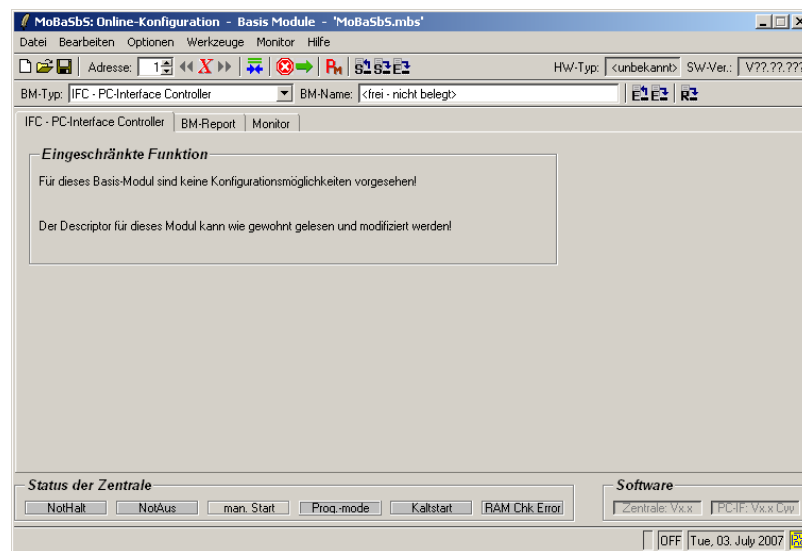
Lok-Datenbank mit MoBaSbS-Hardware austauschen

Über den Knopf  kann die gesamte aktuell im BMC SRAM vorhandene Datenbasis in das Konfigurationstool geladen werden. Evtl. bereits vorhandene Daten in dem Konfigurationstool werden dabei überschrieben. Der Knopf  schreibt die Datenbasis zurück in das BMC-SRAM und macht sie automatisch sofort für die MoBaSbS Hardware verfügbar.

Der Druck auf den Knopf  löst ein automatisches Übertragen der im BMC-SRAM gespeicherten Daten in das auf der BMC-Platine vorhandene externe serielle EEPROM aus. Dabei werden alle im EEPROM vorhandenen Daten überschrieben. Über den Knopf  kann ein automatisches Übertragen der im externen EEPROM enthaltenen Daten in das BMC-SRAM gestartet werden. Dabei werden die im SRAM vorhandenen Daten komplett mit dem EEPROM Inhalt überschrieben. Dieses kann u. a. genutzt werden, um temporäre Änderungen im SRAM wieder rückgängig zu machen.

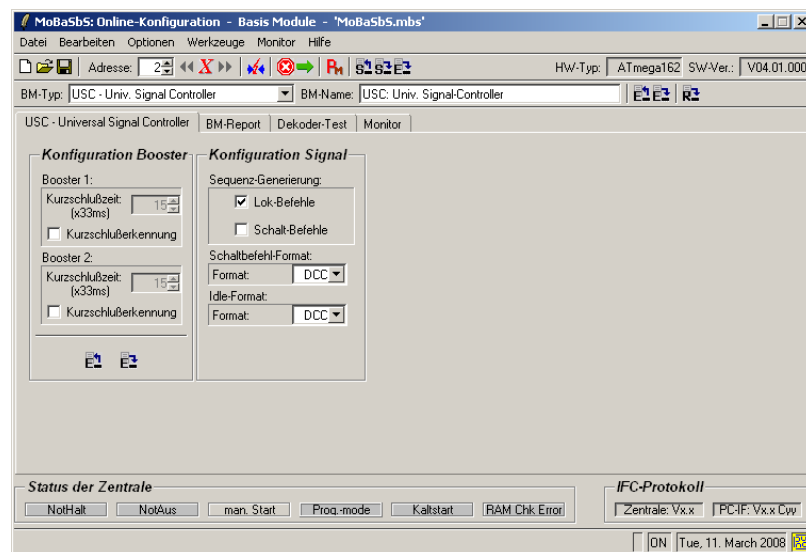
12.5.4 IFC

Der PC-Interface Controller bildet die Schnittstelle zwischen PC und der auf ihm laufenden Software und der MoBaSbS-Hardware. Das PC-Interface kann nicht direkt per Software konfiguriert werden, sondern muss über DIP-Schalter auf der Vorderseite konfiguriert werden.



IFC

12.5.5 USC



USC

Das USC-Modul vereint zwei Aufgaben, die vorher durch den LSC und SSC ausgeführt wurden:

- Erzeugung von Steuerbefehlen für die verschiedenen Lokomotiven
- Erzeugung von Steuerbefehlen für MM- oder DCC-Format basierte Funktionsdecoder

Beide Arten von Steuerbefehlen werden über einen entsprechenden Booster an die Lokomotiven bzw. angeschlossene Funktionsdecoder gesendet.

Im linken Teil des Registers mit der Überschrift "Konfiguration Booster" kann die Kurzschlusserkennung der Booster für beide USC-Ausgänge unabhängig aktiviert und wenn aktiviert auch deren Verzögerung unabhängig eingestellt werden.

Im rechten Teil mit der Überschrift "Konfiguration Signal" kann festgelegt werden, ob der USC Lok-Befehle und/oder Schalt-Befehle generieren soll. Für die Schaltbefehle ist das Protokoll (DCC/MM2, Default-Wert ist DCC) festzulegen.

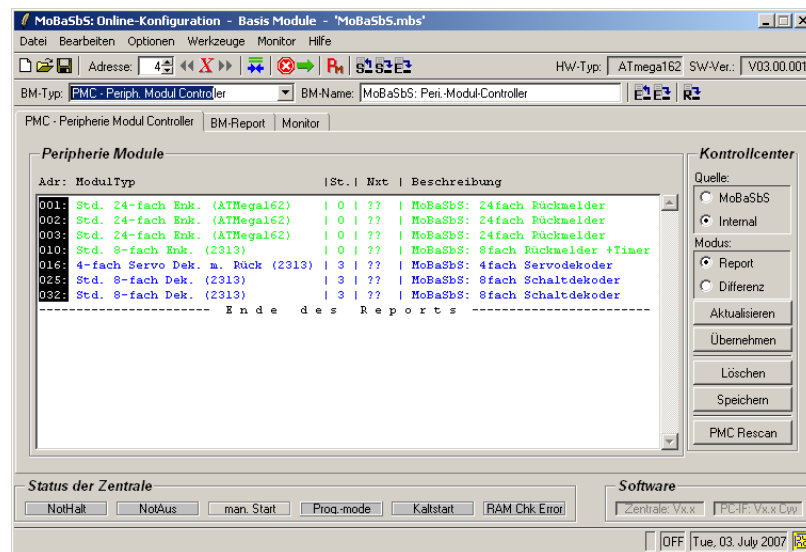
Zusätzlich kann noch das IDLE-Format festgelegt werden. Je nach eingesetzten Lok-Decodern kann der Anwender jetzt zwischen dem DCC und dem MM IDLE-Format auswählen. Die Defaulteinstellung für das Idle-Pattern steht auf DCC und kann so auch unverändert für MM verwendet werden.

Erklärung:

Bevor überhaupt eine einzige Lok aufgerufen wird und der USC (oder jede andere Digitalsteuerung) weiß, welches Signalmuster ans Gleis angelegt werden muss, wird das sog. Idle-Pattern gesendet. Das Idle-Pattern im MM-Format wird dabei von keinem Lokdecoder verstanden, sorgt aber dafür, dass eben nicht einfach nur eine Gleichspannung ans Gleis angelegt wird, sondern ein (wenn auch von keinem Decoder zu verstehendes) gültiges Signalmuster. Der Unterschied zwischen MM und DCC-Idle ist dabei, dass das DCC-Idle im Mittel 'Gleichspannungs-Offsetfrei' ist, d.h. die negativen und positiven Anteile im Muster sind gleich oft (lang) vorhanden. Außerdem ist das DCC-Idle schneller abgearbeitet, so dass die nächste zu bearbeitende Lok wieder schneller an der Reihe ist. Deshalb auch der Rat, das per Default bereits gesetzte DCC-Idle so stehen zu lassen. Im Falle unseres USCs werden ja 16 Loks in einem zyklisch abzuarbeitenden Speicher gehalten. Am Anfang stehen die Protokoll-Formatinfos aller 16 Einträge noch auf 'Idle' und werden erst nach Aufruf einer Lok durch deren konkrete Protokollinfo ersetzt.

Aktuelle Einstellungen können mit Hilfe Knöpfe  aus dem USC gelesen bzw. mit  in den USC zurück geschrieben werden.

12.5.6 PMC



PMC

In diesem Register können alle an diesen PMC angeschlossenen Peripherie-Module analysiert und die Ergebnisse in die interne Datenbasis übernommen werden.

Es werden nur Informationen für die Adressen in die Datenbasis übernommen, die nach dem letzten PMC-Rescan als nicht frei erkannt wurden. Vormals belegte Adressen, die jetzt als 'frei' erkannt werden bleiben in der internen DB als belegt markiert. In diesem Fall ist es sinnvoll, nach allen Adress-Änderungen mit einer komplett neuen Hardware-Analyse zu beginnen oder die nicht mehr belegten Module von Hand aus der Datenbasis zu löschen.

Je nach gewählter Quelle ('MoBaSbS'=Hardware, 'Internal'=interne Datenbasis) wird der unter Modus gewählte Report erstellt und im linken Textfenster angezeigt.

Wird beim Start automatisch die Default-Datenbasis 'MoBaSbS.mbs' eingelesen, wird automatisch die Ansicht auf 'Internal' gesetzt. Wird keine Datenbasis eingelesen, wird die Ansicht auf MoBaSbS gesetzt und vom Anwender wird erwartet, die MoBaSbS neu zu analysieren und zu speichern.

Rückmeldekontakte bzw. Encoder werden dabei in grün, Schaltdecoder in blau dargestellt, die Spalte 'St.' enthält die Information über den Strang (0-3), an dem sich das Peripherie-Modul elektrisch befindet. Für Encoder wird zusätzlich die Information über die nächste abzufragende Adresse angezeigt ('Linked List'), die der PMC bei der zyklischen Abfrage der an ihn angeschlossenen Encoder durchläuft (üblicherweise rückwärts).

Die Adresse in der linken Spalte ist als 'Link' ausgelegt. Durch Anklicken wird das Register / die Optionen des Peripherie-Moduls eingeblendet.

Eine Abfrage wird durch Druck auf den Knopf 'Aktualisieren' gestartet und kann durch Drücken des Knopfes 'Übernehmen' nach Rückfrage in die interne Datenbasis übernommen werden. Während der Analyse zeigt ein blauer Balken den Fortschritt an.

'Löschen' bzw. 'Speichern' ermöglicht ein manuelles Löschen der gesamten Liste bzw. ein Speichern der aktuellen Anzeige in eine ASCII-Datei.

Wird in den Modus 'Differenz' geschaltet, werden alle Informationen der aktuellen Ansicht rot eingefärbt, die in der jeweils anderen Ansicht nicht vorhanden oder anders sind. Dieses funktioniert in beiden Ansichten und dient zum Aufspüren von Unterschieden. Dabei ist zu beachten, dass immer aktuell gelesene (aus der MoBaSbS analysierte Informationen **nicht** in die interne Datenbasis übernommen werden dürfen, da dann ja beide Vergleichsquellen auf jeden Fall identisch sind.

In der untersten Zeile im Feld links neben ‚ON / OFF‘ wird ein # angezeigt, wenn sich an der internen Datenbasis seit dem letzten Abgleich mit dem Archiv etwas geändert hat. Archiv meint hier eine gespeicherte Datei aber auch eine frisch initialisierte Datenbasis (z.B. nach dem Start oder nach Ausführen des Befehls 'Neu'). Nach jedem Speichern in eine Datei oder nach dem Befehl 'Neu' verschwindet das Zeichen. Übrigens kommen Abfragen nach Speichern oder Verwerfen von Daten nur, wenn das Zeichen # sichtbar ist.

Inbetriebnahme / Tausch eines Peripherie-Moduls

Bei Änderungen (Modul am Strang getauscht, neues Modul bzw. Strang hinzugefügt) sind immer folgende Schritte an jedem PMC durchzuführen, an dem sich etwas geändert hat:

1. Modul wird an einen Strang des PMC angeschlossen.
2. PMC-Rescan auslösen (via Konfig-SW oder HDC)
3. PMC: Konfig-SW löschen (Knopf 'Löschen' drücken)
4. PMC: Konfig-SW aktualisieren (Knopf 'Aktualisieren' drücken)
5. Bei einem neuen Peripherie-Modul ist nach dem Laden der Software die Default-Adresse 255 eingestellt. Sie ist zu ändern (PM-Adresse ändern) und die Schritte 2 -4 sind nochmals durchzuführen

Danach sollten alle PM-Module in der neuen Topologie wieder erkannt und ansprechbar sein.

War ein PM-Modul bereits einmal bekannt und ist jetzt nicht mehr erreichbar, erscheint die Meldung ‚Peripherie Modul hat nicht geantwortet (BMC NAK empfangen)‘.

Ursachen können sein:

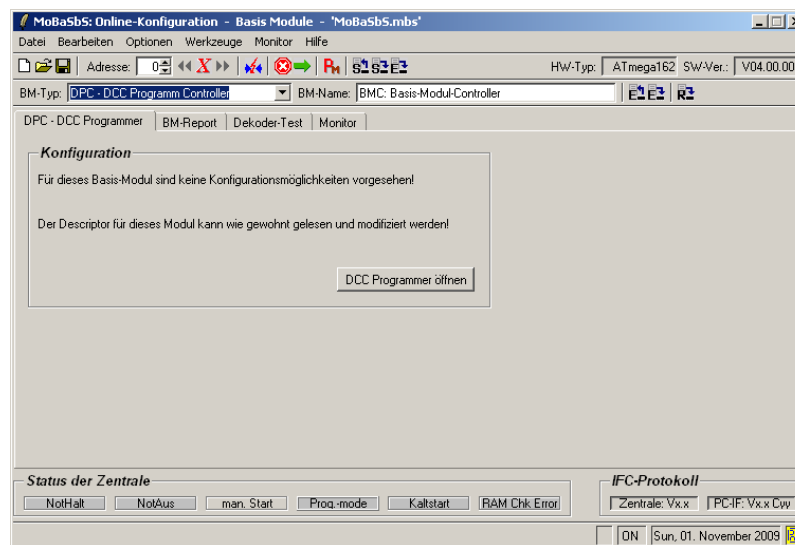
am falschen (ungelernten) Strang eine PMCs hängt

- an einem anderem als angelernten PMC hängt
- das RJ45 Kabel defekt ist
- das PM-Modul keine oder nicht ausreichende Spannung hat
- mehrere Module mit der gleichen Adresse plötzlich im System sind
- ...

Erklärung: Der BMC lauscht am Bus und überwacht, ob sich ein angesprochenes Modul (PM oder BM) innerhalb einer vorgegebenen Zeit meldet. Wenn nicht, übernimmt er die Initiative und sendet BMC NAK (BMC Not Acknowledge) an die SW zurück. Damit entsteht kein Deadlock in der Kommunikation zwischen PC und der MoBaSbS.

Im Kapitel 13.3 stehen Hinweise zu den PM-Adressen bei Betrieb mit Traincontroller.

12.5.7 DPC



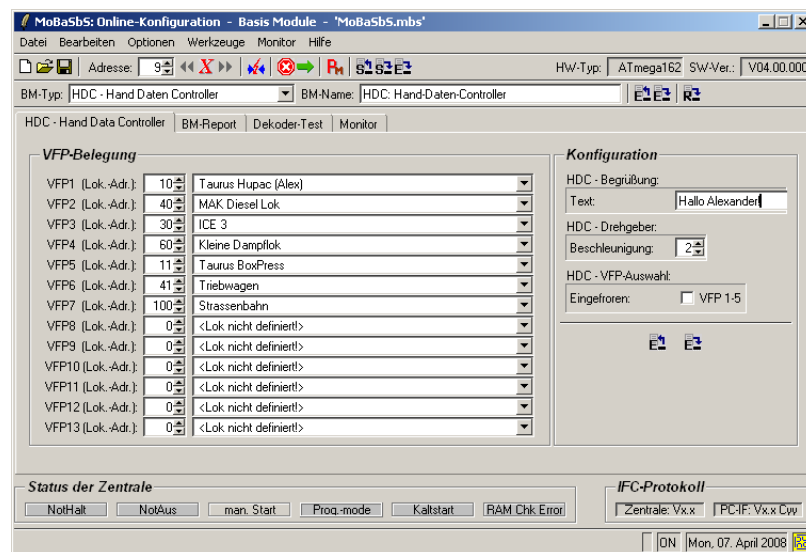
DPC

Der DCC Programmier Controller bildet die Schnittstelle zwischen PC und den Lok-Decodern im DCC Format. Mit dem DPC können CVs des Lok-Decoders bearbeitet und archiviert werden. Durch die weitgehend automatische Konfiguration im Betrieb ('auto detection') gibt es keinerlei Konfigurationsmöglichkeiten. Über den Knopf unten rechts kann ein separates Fenster für den DPC geöffnet werden. Ein Öffnen ist auch über das Menü unter "Werkzeuge -> DCC-Programmer" möglich.

Der DPC kann auch als eigenständiges Modul außerhalb der MoBaSbS-Backplane über die integrierte RS232 bzw. USB Schnittstelle betrieben werden, d. h. es werden keine anderen BM-Module der MoBaSbS benötigt. In diesem Fall kann die Konfigurationssoftware ebenfalls benutzt werden, es stehen aber nur die Konfiguration der seriellen Schnittstelle sowie das Werkzeug zum Programmieren von DCC Decodern incl. aller Funktionen zur Verfügung.

Hierfür wird empfohlen, die Konfigurationssoftware ein zweites Mal parallel zu installieren und durch Ausführen der Datei 'MBSDccPrg.tcl' im Unterverzeichnis MBSCfgSW zu starten. In diesem Fall wird das Werkzeug zum Programmieren von Decodern sofort geöffnet, das bekannte Hauptfenster wird in diesem Fall minimiert gestartet. Ein Zugriff auf das Hauptfenster ist beim ersten Start für die Konfiguration der seriellen Schnittstelle notwendig und jederzeit möglich.

12.5.8 HDC

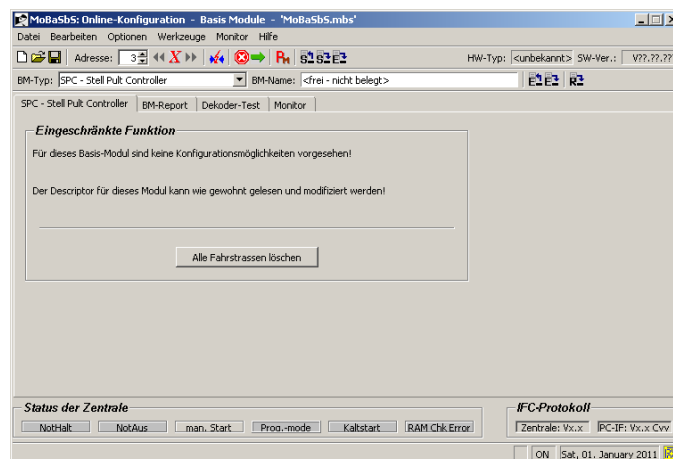
HDC

Über das Register für den Hand Daten Controller können neben dem Begrüßungstext auch die Lok-Adressen auf die max. 13 virtuellen Fahrpulte (VFP) zugewiesen werden. Dieses kann wahlweise durch die Auswahl einer Adresse oder die Auswahl eines Lok-Namens erfolgen. Adresse oder Name werden dann entsprechend ergänzt, sofern die Lok-Datenbasis geladen ist.

Der Begrüßungstext kann max. 16. Zeichen enthalten. Weiterhin besteht die Möglichkeit, die 'Beschleunigung' des Drehgebers (größere Schrittweite bei schnellerem Drehen) einzustellen. Dieser Wert kann im Bereich von 1-6 definiert werden.

Mit dem Schalter 'Eingefroren' werden die VFP-Tasten 1-5 fixiert. Dadurch können sie nicht mehr versehentlich geändert werden.

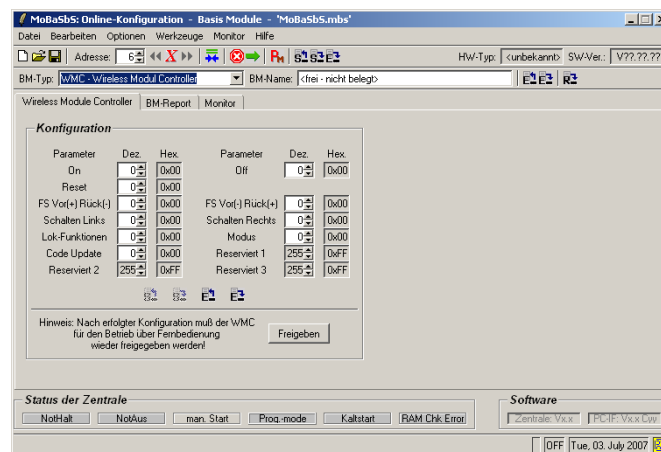
12.5.9 SPC



SPC

Der Stell-Pult Controller SPC kann nicht direkt per Software konfiguriert werden sondern muss über eine eingebaute Lernfunktion programmiert werden. Lediglich ein Knopf zum Löschen aller im SPC abgelegten Fahrstrassen erleichtert das Löschen.



12.5.10 WMC



WMC

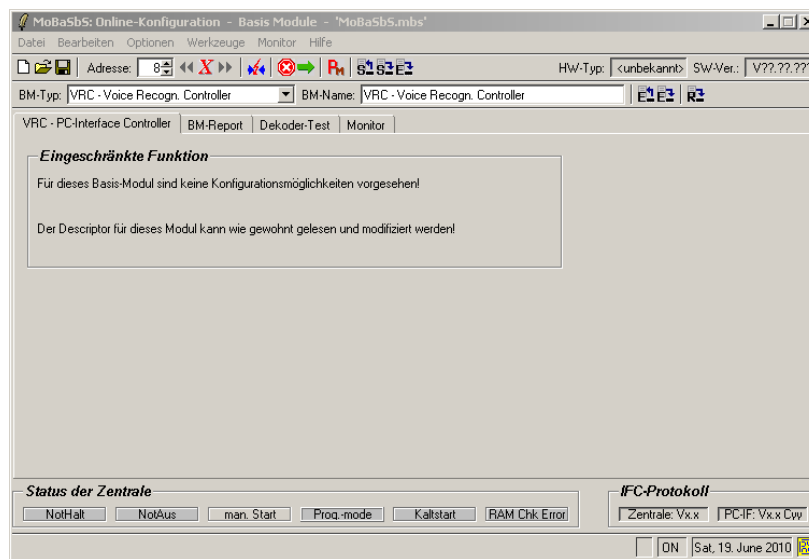
Der Wireless Module Controller (WMC) bietet die Möglichkeit, die Modellbahn über eine drahtlose handelsübliche Fernbedienung (RC5 codiert) bedienen. Neben dem Schalten von Weichen und Signalen ist auch die Steuerung von Lokomotiven möglich.

Über das Konfigurationsfeld können die Tasten-Codes für die verschiedenen Tasten definiert werden.

Diese müssen dann im EEPROM gespeichert werden. Aktuelle Einstellungen können mit Hilfe Knöpfe  aus dem WMC gelesen bzw. mit  in den WMC zurück geschrieben werden.

In der aktuellen Version besteht nur Kommunikation mit dem WMC, wenn diese via Fernbedienung aktiviert worden ist. Nach dem Speichern der Konfiguration im WMC-Eeprom muss durch Druck auf den Knopf 'Freigabe' wieder die Kommunikation an die Fernbedienung zurückgegeben werden.

12.5.11 VRC

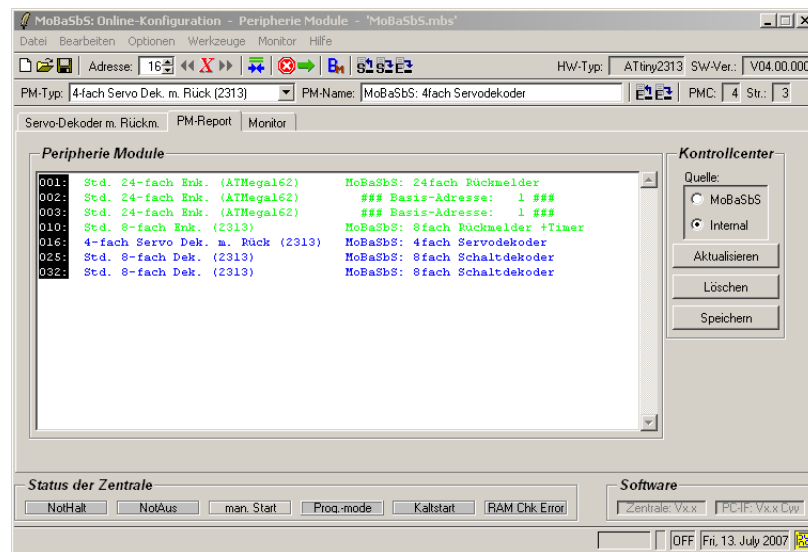


VRC

Der Voice Recognition Controller bildet die Schnittstelle zwischen PC, der auf ihm laufenden Spracherkennungssoftware und der MoBaSbS-Hardware. Der Voice Recognition Controller kann nicht direkt per Software konfiguriert werden.

12.6 Peripherie-Module

12.6.1 PM-Report



PM-Report

Mit Hilfe des Registers PM-Report wird eine schnelle Übersicht über alle zurzeit in der MoBaSbS-Hardware oder der internen Datenbasis bekannten Peripherie-Module generiert. Alle erkannten Module werden neben ihrer Adresse unter Angabe von Modul-Typ und Modul-Name (Descriptor) im linken Textfenster gelistet. Für alle PM-Module kann ein beliebiger Name vergeben werden. Die Länge ist auf 32 Zeichen beschränkt.

Das Auslesen und Speichern erfolgt über die Knöpfe  bzw. .

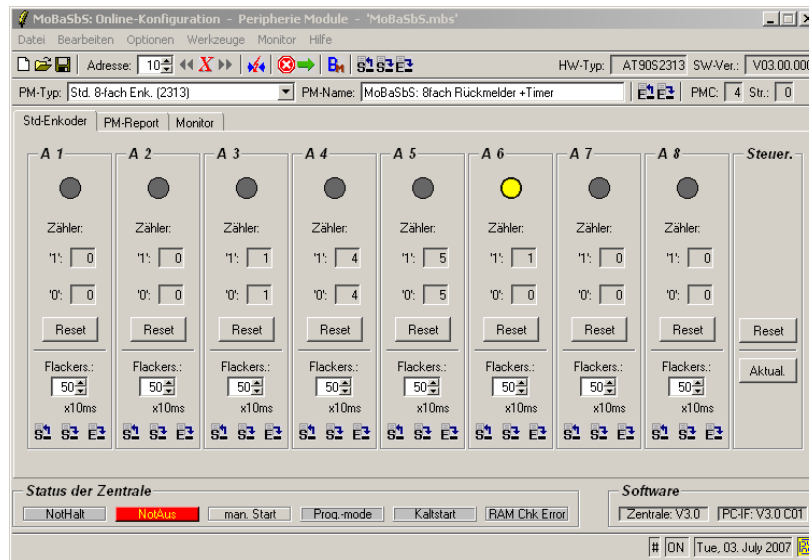
Hinweis: Bei ‚MoBaSbS-fremden‘-Decodern (z.B. LDT) muss bei einer Änderung erst der PM-Name und danach der PM-Typ eingegeben werden.

Über das rechte Kontrollcenter kann die Quelle ('MoBaSbS'=Hardware, 'Internal'=interne Datenbasis) ausgewählt und die Analyse gestartet werden (Knopf 'Aktualisieren'). Im Gegensatz zum BM-Report können die Daten nicht in der internen Datenbasis gespeichert werden. Eine Analyse zur Übernahme in die interne DB ist ausschließlich über das Register des PMC in der BM-Ansicht möglich.

Die Adresse in der linken Spalte ist als 'Link' ausgelegt. Durch Anklicken wird das Register des jeweiligen Peripherie-Moduls eingeblendet.

Mit Hilfe der beiden Knöpfe 'Löschen' und 'Speichern' kann das aktuelle Analyse-Ergebnis gelöscht oder in einer ASCII-Datei zwecks Archivierung gespeichert werden.

12.6.2 8-fach Encoder



8-fach Encoder, Register




Im Register für den 8-fach-Encoder sind 8 Felder definiert, die je einen Rückmeldekontakt behandeln und mit A1 bis A8 überschrieben sind. Im Feld PM-Name kann eine individuelle Bezeichnung vergeben werden (lesen und speichern über die Knöpfe in der Werkzeugleiste 2).

Der aktuelle Kontaktzustand wird im oberen Teil durch eine stilisierte LED angezeigt: '1' bzw. geschlossen entspricht einer gelb leuchtenden LED, '0' bzw. offen entspricht einer nicht leuchtenden LED.

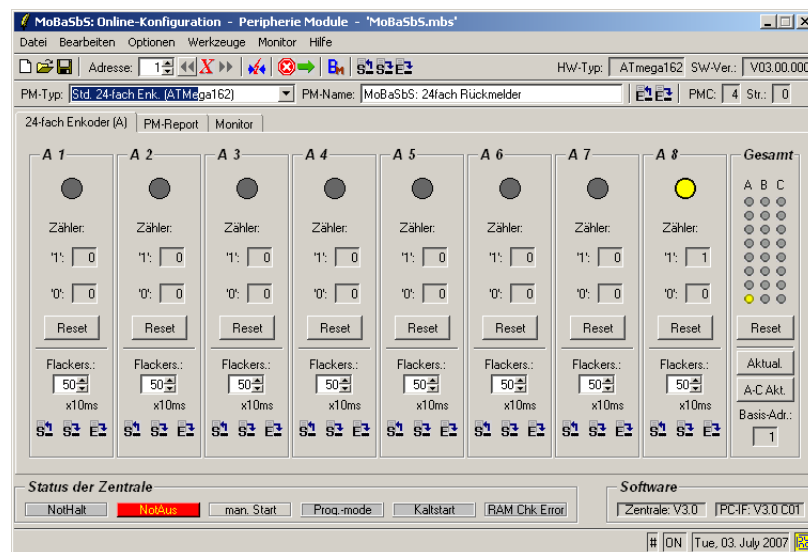
Der Zustand der LEDs ändert sich nur bei sichtbarem Register für die aktuell eingestellte Adresse. Ein Wechsel der Adresse löscht den aktuellen Zustand und setzt alle LEDs auf '0' (offener Kontakt) und alle Zähler zurück auf 0. Der aktuelle Zustand kann über den Knopf 'Aktual' im rechten Feld 'Steuer.' wieder angezeigt werden (liest aktuellen Status aus PMC-Modul aus).

Jeder Wechsel auf '1' bzw. '0' wird von den beiden Zählern in jedem Feld registriert, die über den Knopf 'Reset' in jedem Feld zurückgesetzt werden können. Der Knopf 'Reset' im Feld 'Steuer.' setzt alle 16 Zähler zurück.

Im unteren Bereich eines jeden Feldes befindet sich der Konfigurationsbereich, bei dem die Flackerschutzzeit eingestellt werden kann. Der Wert kann für jeden Kontakt in Schritten von 10 ms eingestellt werden kann. Dieses entspricht im Prinzip einer Kontakt-Entprellung, da frühestens nach der eingestellten Zeit eine erneute Änderung gemeldet werden kann. Ein Wert von 1 entspricht einem Abschalten.

Zur Einstellung/Konfiguration kann die Flackerschutzzeit für jeden Kontakt getrennt aus dem SRAM gelesen (Knopf ) bzw. geschrieben werden (Knopf ). Endgültige Werte können dann mit Hilfe des Knopfes  in das PM-Modul geschrieben werden und werden erst nach einen SW-Reset oder einem Hardware-Reset aktiv; Änderungen am SRAM wirken sich dagegen immer sofort aus.

12.6.3 24-fach Encoder



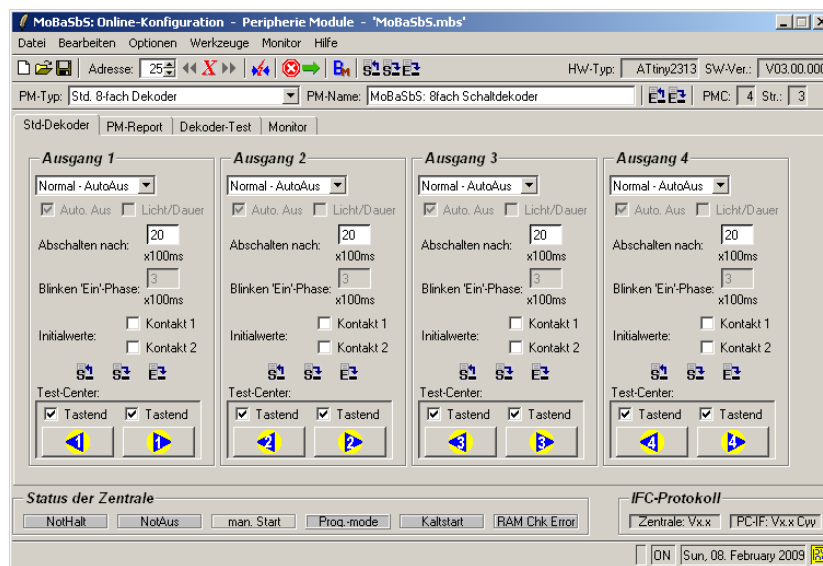
24-fach Encoder

Die Register für den 24-fach Encoder entsprechen in Aufbau und Funktion dem 8-fach Encoder mit 3 aufeinander folgenden Adressen (Basis-Adresse, Basis-Adresse+1 und Basis-Adresse+2). Zur Verdeutlichung sind die Kontaktfelder für die Basis-Adresse mit dem Buchstaben A, für die Basis-Adresse+1 mit dem Buchstaben 'B' und für die Basis-Adresse+2 mit einem C benannt. Das Feld auf der rechten Seite mit der Überschrift 'Gesamt' zeigt unter jeder der drei möglichen Adressen eine Gesamtübersicht über alle Kontakte. Wie für alle PM-Module kann auch für den 24-fach-Encoder im Feld PM-Name ein individueller Name vergeben werden (lesen und speichern über die Knöpfe in der Werkzeugleiste 2).

Der Zustand der LEDs ändert sich nur bei sichtbarem Register für die aktuell eingestellte Adresse sowie für die beiden zurzeit unsichtbaren Register in der Gesamtansicht. Ein Wechsel der Adresse auf eine Adresse außerhalb des für diesen Encoder definierten Bereichs löscht den aktuellen Zustand und setzt alle LEDs auf '0' (offener Kontakt) und alle Zähler zurück auf 0. Der aktuelle Zustand des Registers kann über den Knopf 'Aktual' im rechten Feld 'Steuer.' wieder angezeigt werden (liest aktuellen Status aus PMC-Modul aus).




Die Flackerschutzzeit ist in Abschnitt 8.3 beschrieben.

12.6.4 8-fach Decoder



8-fach Decoder

Das Register für den 8-fach-Decoder enthält 4 Abschnitte für 4x2 Kontaktpaare ('Ausgang A' bis 'Ausgang D'). Wie für alle PM-Module kann auch für jeden 8-fach Decoder im Feld PM-Name ein individueller Name vergeben werden (lesen und speichern über die Knöpfe in der Werkzeugleiste 2).

Jeder Abschnitt konfiguriert und testet ein Kontaktpaar, wobei die Konfiguration unabhängig von den anderen Kontaktpaaren über die Knöpfe  (aktuelle Konfiguration aus SRAM lesen) und  (aktuelle Konfiguration in SRAM speichern) verändert werden kann. Dauerhaft können Änderungen über den Knopf  in das PM-Modul geschrieben werden und wirken sich nach dem nächsten SW-Rest bzw. HW-Reset aus, Änderungen im SRAM wirken sich dagegen immer sofort aus.

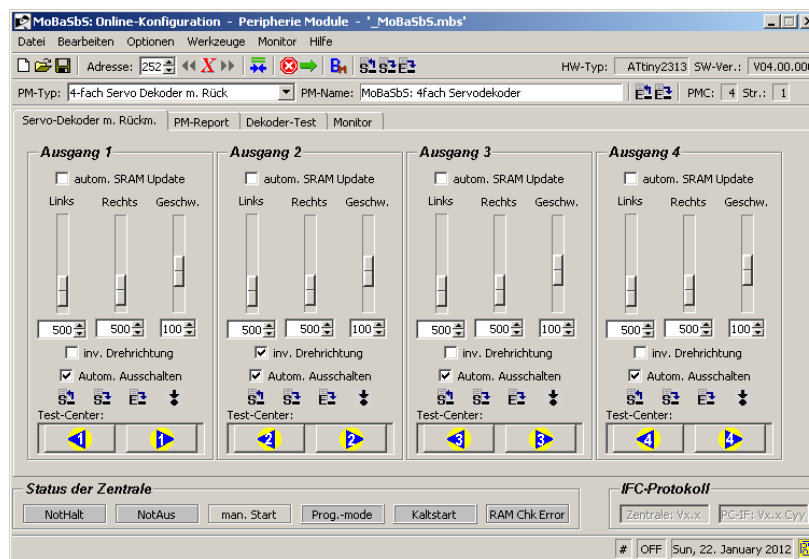
Für jeden Ausgang wird über das obere Pull-down-Feld ein Funktionsmodus ausgewählt. Je nach gewähltem Modus werden dann die verschiedenen Elemente zur Bearbeitung freigegeben bzw. der aktuelle Status der Felder/Optionen aktualisiert. Die verschiedenen Kombinationen sind in folgender Tabelle aufgelistet:

Funktionsmodus	Autom. Abschalten	Licht/Dauer	Abschaltzeit	Blinken ,ein'	Beschreibung
Normal AutoAus	X	---	aktiv	---	Normale Funktion mit autom. Abschaltung gemäß Zeitvorgabe, Ausschaltkommando wird ausgewertet
Normal	---	---	---	---	Normale Funktion, Ein- und Ausschaltimpuls notwendig
Wechsel	---	X	---	---	Wechselfunktion, Einschalten Kontakt 2 bewirkt Abschalten von Kontakt 1, Dauer- oder Lichtsignalmodus
Blinken	---	---	---	aktiv	Blinken wird bei Betätigen eines Kontaktes aktiviert (Reihenfolge je nachdem welcher Kontakt betätigt wurde), Frequenz nach Zeitvorgabe
Trigger AutoAus	X	---	aktiv	---	Normale Funktion mit autom. Abschaltung gemäß Zeitvorgabe, Ausschaltkommando wird ignoriert
Puls - AutoAus	X	---	aktiv	aktiv	Normale Funktion mit autom. Abschaltung nach Zeitvorgabe, bis zum autom. Abschalten wird der Kontakt nach Zeitvorgabe 'Blinken Ein' ein- und ausgeschaltet (Erzeugung von wiederholten kurzen Einschaltimpulsen)

Über die beiden Felder 'Kontakt 1' und 'Kontakt 2' können Initialwerte für jeden der beiden Kontaktausgänge vorgegeben werden. Ein markiertes Feld bedeutet dabei, dass der Kontakt nach einem Reset geschlossen bzw. aktiv ist.

Mit dem 'Test-Center' am unteren Ende des Feldes kann die aktuelle Konfiguration getestet werden. Drücken des Knopfes 'Kontakt x' sendet den Befehl zum Schließen des Kontaktes x, das Loslassen des Knopfes sendet den Befehl zum Öffnen des Kontaktes (Tastfunktion). Wird die Option 'Tastend' für einen Kontakt deaktiviert, bewirkt jedes Drücken und Loslassen eines Knopfes 'Kontakt x' eine Zustandsänderung zwischen 'geschlossen' und 'offen' für den entsprechenden Kontakt (Schaltfunktion).

12.6.5 4-fach Servo-Decoder



Servo-Decoder

Mit Hilfe des Registers für den Servo-Decoder können die einzelnen Servos eines Decoders individuell eingestellt werden. Wie für alle PM-Module kann auch für jeden Servo Decoder im Feld PM-Name ein individueller Name vergeben werden (lesen und speichern über die Knöpfe in der Werkzeugleiste 2).

Für jeden Servo ist ein Konfigurationsfeld definiert, das mit 'Ausgang A' bis 'Ausgang D' überschrieben ist. Für jeden Ausgang kann der linke und der rechte Anschlag sowie die Umlaufzeit (-geschwindigkeit) definiert werden. Dazu stehen für die grobe Justierung Schieberegler zur Verfügung, die Feinjustierung kann dann über die darunterliegenden Felder mit den Spin-Knöpfen erfolgen. Die Auflösung hier beträgt dabei 1.

Sind alle Schieberegler nach unten geschoben, steht der Servo in Mittenstellung und ist auf die kürzeste Umlaufzeit eingestellt.

Die eingestellten Werte können in der mobasbs.mbs gespeichert werden (Menu Datei/Speichern) und stehen dann nach einem Prozessor-Wechsel wieder zur Verfügung. Hierbei muss darauf geachtet werden, dass die alte Moduladresse wieder eingestellt wird.

Ist oberhalb der Schieberegler die Option 'autom. SRAM Update' aktiviert, wird jede Änderung eines der drei Parameter sofort in das SRAM geschrieben und es wird sofort eine der folgenden Aktionen ausgeführt:

Anschlag Links geändert	Servo fährt auf die neu definierte Endposition
Anschlag Rechts geändert	Servo fährt auf die neu definierte Endposition
Umlaufzeit geändert	Servo beginnt im Wechsel zwischen Endposition links und Endposition rechts zu pendeln

Die Software des Decoders ist rückmeldefähig. Die Tasten 'Links' bzw. 'Rechts' werden bei einer Rückmeldung gelb eingefärbt und zeigen damit den aktuellen Zustand des Servos an. Während der Bewegung wird der alte Zustand angezeigt bis das Erreichen des neuen gemeldet wird.

Mit der Option 'Inv. Drehrichtung' ist es möglich, die Drehrichtung des Servos umzukehren, wenn das entsprechende Häkchen in der Maske gesetzt ist und im SRAM sowie im EEPROM Bereich gespeichert ist. Ist das Feature eingeschaltet, bewirkt es, dass der Servo beim Kommando 'Drehe nach Links' nach rechts dreht und umgekehrt. Es wird also nur das Kommando im Servo-Dekoder neu interpretiert. Die

Rückmeldung ist nicht invertiert. Das bedeutet, dass bei einem Servo-Dekoder mit Rückmeldung der 'falsche' sprich invertierte Knopf andersfarbig neu hinterlegt wird.

Ist die Option 'Autom. Ausschalten' aktiviert, wird der Servo nach Erreichen seiner neuen Sollposition plus einer zusätzlichen Abschaltzeit abgeschaltet. In diesem Zustand wird der Stellimpuls für den Servo unterdrückt und es erfolgt keine Nachregelung. Bei aktivierter Option wird der Servo permanent nachgeregelt, was äußere Einwirkungen wirksam kompensiert.

Über die beiden Knöpfe 'Links' und 'Rechts' kann der Servo gezielt in die eingestellte linke bzw. rechte Endposition verfahren und belassen werden, wobei je nach Option 'Autom. Ausschalten' dann der Stellimpuls ggf. nach einiger Zeit unterdrückt wird.

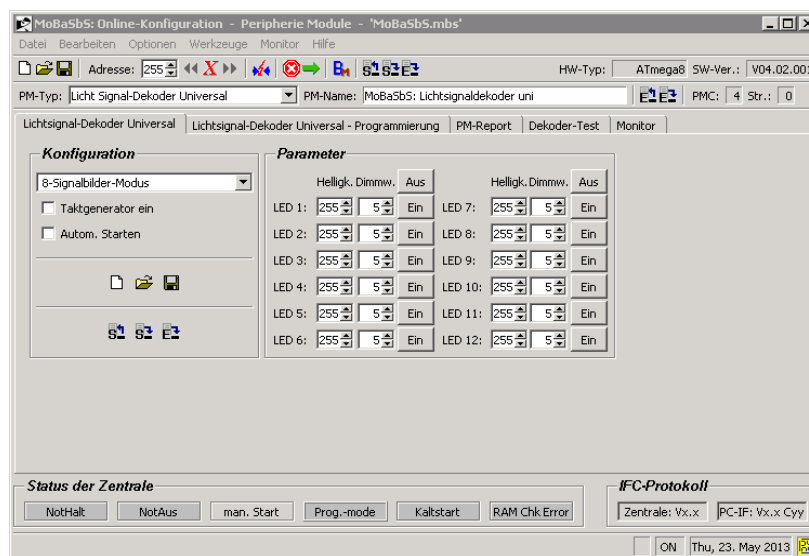
Über die Knöpfe direkt oberhalb der des 'Test-Center' kann die Konfiguration für jeden Servo individuell aus dem SRAM des Moduls ausgelesen (S1), in das SRAM des Moduls geschrieben (S2) bzw. permanent in den EEPROM-Bereich des Moduls geschrieben werden (E3). Ein Druck auf den Knopf S3 fährt den entsprechenden Servo in die Mittelstellung

12.6.6 8-fach Servo-Decoder

Der 8-fach Servo-Decoder verhält sich wie zwei eigenständige 4-fach Servo-Decoder und wird vom MoBaSbS-Konfigurationstool so behandelt.

Es gilt hier deshalb die gleiche Beschreibung wie für den 4-fach Servo-Decoder.

12.6.7 Lichtsignal-Decoder-Universal





LDEC-Decoder-Universal

Der Lichtsignal-Decoder-Universal ist die konsequente Weiterentwicklung des Lichtsignal-Decoders. Waren beim Lichtsignal-Decoder alle Signalbilder fest vorgegeben können beim Lichtsignal-Decoder-Universal bis zu 8 bzw. 16 verschiedene Signalbilder frei programmiert werden. Der Lichtsignal-Decoder-Universal verwendet die Platine auf Basis des ATmega8 und unterstützt zwei verschiedene Betriebsmodi:

- 8-Signalbilder-Modus
- 16-Signalbilder-Modus

Der Signalbild-Modus kann im linken Feld mit der Überschrift "Konfiguration" ausgewählt werden. Wird '8-Signalbilder-Modus' ausgewählt, können im Register 'Lichtsignal-Decoder-Universal-Programmierung' 8 Signalbilder incl. Blinkfunktion frei definiert werden. Im '16-Signalbilder-Modus' sind es 16.

Signalbilder können mit Hilfe der Knöpfe   als Datei gespeichert bzw. geladen werden. Ein Druck auf das linke Symbol löscht die aktuelle Programmierung und stellt wieder Default-Werte her. Die Programmierung wird in einer Datei im Intel-HEX Format und der Erweiterung '.mbe' (MBS EEP Konfigurationsdatei) abgelegt und enthält nur Ausschnitte aus einem für die Peripherie-Modul-Programmierung notwendigen Datensatz.



Über die bekannten Symbole kann die Konfiguration aus dem SRAM gelesen  bzw. geschrieben  werden. Beim Schreiben ins EEPROM mittels Knopf  wird zudem noch automatisch ein SW-Reset ausgelöst.

In der mittleren Spalte können die Parameter für jeden der 12 LED-Ausgänge eingestellt werden. Für jede LED kann individuell die Helligkeit sowie der Dimmwert eingestellt werden. Sinnvolle Werte für den Dimmwert liegen im Bereich von 1-10 (max. 255). Bei dem voreingestellten Defaultwert von 5 beträgt die Zeit bis zum Aufdimmen 650 ms. Eine Verkleinerung des Wertes erhöht die Zeit bis maximal 3,3 s. Über den rechten Knopf mit der Bezeichnung 'Ein' kann eine einzelne LED zum Leuchten gebracht werden. In diesem Zustand werden Veränderungen der Parameter für diese LED sofort übertragen und wirksam. Über den Knopf 'Aus' können alle LEDs dunkel geschaltet werden. Es kann immer nur eine LED in diesem Modus aktiv sein, d.h. beim Einschalten einer neuen LED verlischt die zuvor aktive LED.


Hinweis: Vor dem Einschalten einer einzelnen LED muss erst das aktuelle Signalbild ausgeschaltet werden (Knopf 'Aus' drücken).

Taktgenerator

Mit der Codeversion LDEC-T (siehe Kap. 13.6) ist es möglich einen Taktgenerator einzuschalten. In diesem Modus arbeitet der LDEC eigenständig und er braucht keine Schaltbefehle von außerhalb, um das Signalbild zu wechseln (Nutzung für Häuserbeleuchtungen, Verkehrsampeln, etc.).

Eingeschaltet wird der Taktgenerator in der Maske Lichtsignal-Dekoder Universal. Dabei besteht zusätzlich die Möglichkeit den Taktgenerator beim Einschalten der Spannung automatisch starten zu lassen. Ansonsten wird der Taktgenerator mit einem Schaltbefehl nach links  ein- und nach  rechts ausgeschaltet.

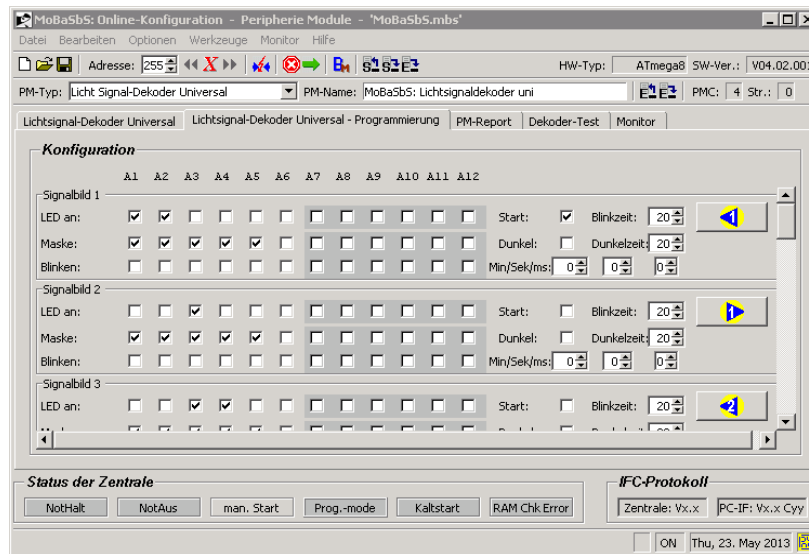
Mit dem Auswahlmeneü 8- oder 16 Signalbild-Modus wird die Anzahl der möglichen Takte (Signalbilder) ausgewählt. Die wirklich benötigten Takte werden dann durch die Zeitstufen in der Maske Lichtsignal-Dekoder Universal Programmierung festgelegt. Wird dabei in allen drei Zeitwerten (Minuten - Sekunden - Millisekunden) der Wert 0 geschrieben, so wird dieser Takt übersprungen. Somit ist es möglich nur die benötigten Takte festzulegen.

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12				
Signalbild 1													Start: <input type="checkbox"/>	Blinkzeit: <input type="text" value="23"/>		
LED an:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Dunkel: <input type="checkbox"/>	Dunkelzeit: <input type="text" value="20"/>		
Maske:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Min/Sek/ms	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Blinken:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				

Die Zeitdauer, wie lange das Signalbild gezeigt wird, wird durch die drei Zeitwerte festgelegt. Es sind Zeitwerte von 100 Millisekunden bis 255 Minuten auswählbar. Eine 1 in Millisekunde bedeutet 100 Millisekunden.



Diese eingestellte Zeitdauer berücksichtigt noch nicht den gewählten Dimmwert und die Dunkelzeit. Möchte man z.B. ein Lauflicht mit kurzen Zeiten verwirklichen, so ist der Dimmwert auf einen hohen Wert und die Dunkelzeit auf 1 zu stellen. Damit wird die Dimm- und die Dunkelzeit fast ausgeschaltet und wirkt sich nicht auf den eingestellten Zeitwert aus.


Register Programmierung



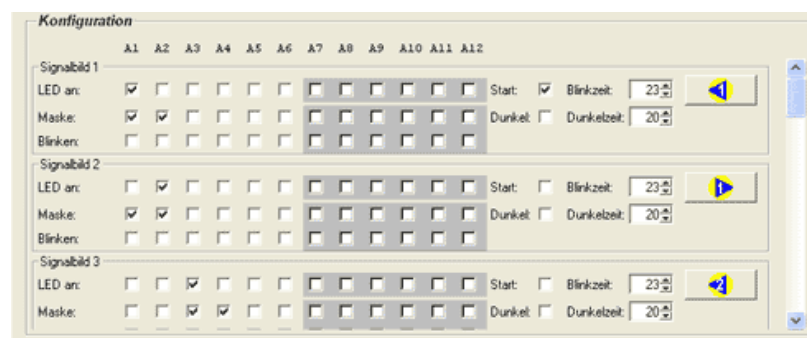
LDEC-Universal-Programmierung

Im zweiten Register für den Lichtsignal-Decoder-Universal können die einzelnen Signalbilder individuell programmiert werden. Prinzipiell besteht die Programmierung aus einer großen Matrix. Mit der Matrix kann für jede LED (bzw. jeden Ausgang) das Verhalten in einem bestimmten Signalbild konfiguriert werden.




Zu Testzwecken kann jedes Signalbild über den rechten farbigen Knopf aufgerufen werden. Der Aufdruck beschreibt den zum Anzeigen notwendigen Befehl (z. B. ) bzw. bei 16 Signalbildern die Kombination von Befehlen (z.B. )

Signalbilder, die im aktuell eingestellten Modus nicht verfügbar sind, haben die Überschrift **"!! >>> nicht verfügbar in diesem Modus <<< !!"** und  als Testkopf



Register Programmierung - Maskierung



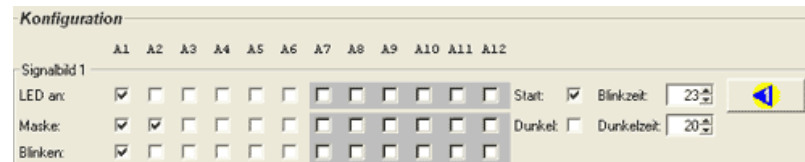
LDEC-Maskierung

Alle LEDs, welche zum selben Signal gehören, bekommen einen Haken bei Maske. Diese Maskierung muss bei allen verwendeten Adressen (Signalbildern) des Signals gleich sein. Bei dem Beispiel ist im Signalbild 1 (Adresse ) und im Signalbild 2 (Adresse ) ein zweibegriffiges Signal maskiert. Im Signalbild 3 (Adresse ) folgt dann das nächste Signal, mit einer neuen Maskierung.

Register Programmierung - Signalbilder

Durch Auswahl der gewünschten LEDs bei jeder Adresse die zum Signal gehören, werden die unterschiedlichen Signalbilder erzeugt. In unserem obigen Beispiel werden bei Adresse  die LED A1 auf- und die LED A2 abgedimmt. Beim Schalten der Adresse  ist es genau umgekehrt. Dieses Beispiel ist im aktuellen Code im Eeprom hinterlegt. Es werden dort zweibegriffige Signale jeweils an den Ausgängen A 1 - 2, A 3 - 4, A 7 - 8 und A 9 - 10 gesteuert.

Register Programmierung - Blinken



LDEC-Maskierung, Blinken

Setzt man zusätzlich in der unteren Zeile bei Blinken ein Häkchen bei dem gewünschten Ausgang, so blinkt diese LED mit der einstellbaren Blinkzeit.

Das Verhältnis zwischen Hell- und Dunkelphase ist dabei gleich. Durch Veränderung der Blinkzeit (Hell- und Dunkelphase) bei der jeweiligen Adresse und /oder des Dimmwertes der jeweiligen LED in der Maske Lichtsignal-Decoder-Universal ist eine individuelle Blinkfrequenz einstellbar.

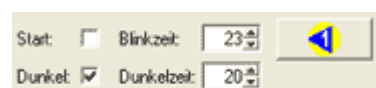
Register Programmierung –Wechselblinken



LDEC-Maskierung, Wechselblinken

Möchte man ein Wechselblinken erzeugen, so setzt man bei allen beteiligten LEDs bei Blinken ein Häkchen. Um das Wechselblinken zu erreichen, wird bei allen LEDs, welche zuerst mit der Dunkelphase beginnen sollen, das Häkchen bei "LED an" entfernt.

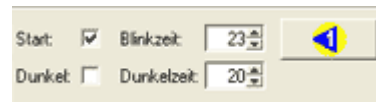
Register Programmierung - Dunkelschaltung



LDEC-Maskierung, Dunkelschaltung

Die bisherige Dunkelschaltung eines Vorsignals am selben Mast des Hauptsignals, wird durch die Auswertung der Maskierung erreicht. Setzt man das Häkchen bei Dunkelschaltung, so wird bei der Ansteuerung dieser Adresse das nachfolgende Signal dunkel geschaltet. Es können auch mehrere Adressen eines Signals die Dunkelschaltung auslösen z.B. Hp0 und Sh1.

Der LDEC erkennt das nachfolgende Signal an der Veränderung der Maskierung. Daher muss das Signal, welches dunkel geschaltet werden soll, in der Adressierungsreihenfolge dem Hauptsignal folgen. Belegt als Beispiel das Hauptsignal die Adressen 1 bis 2, so muss das Vorsignal bei Adresse 3 ← beginnen.

Register Programmierung - StartadresseLDEC-Maskierung, Startadresse

Um dem LDEC mitzuteilen, welche Signalbilder nach dem Einschalten aufgedimmt werden sollen, setzt man bei der gewünschten Adresse ein Häkchen bei Start. Es wird immer nur eine Startadresse pro Signal ausgewertet. Wird die Auswahl der Startadresse vergessen, so wird immer die erste Adresse des Signals aufgedimmt.

16-Signalbilder-Modus

In diesem Spezialmodus ist es möglich, mit den 4 Adressen des Peripheriemoduls bis zu 16 Signalbilder zu schalten.

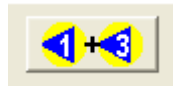
Dies ist einerseits bei Signalsystemen mit mehr als 8 Signalbegriffen erforderlich (z.B. HL-System) oder wenn die 12 LED-Anschlüsse optimal ausgenutzt werden sollen (Anschluss von 4 dreibegriffigen Signalen).

Um dies zu erreichen, werden die 4 Adressen als eine Einheit betrachtet. Dies wird durch die Auswertung von zwei Schaltbefehlen an dieses Modul erreicht.

In der Steuerungssoftware, wie Traincontroller™, sind dazu zwei Signale notwendig. Dabei muss bei beiden Signalen die Anzahl der Kontakte auf 3/4 stehen und beide Signale müssen dieselbe Moduladresse besitzen. Dem ersten Signal werden dabei die Ausgänge 1 und 2 und dem zweiten Signal die Ausgänge 3 und 4 zugewiesen. Steuert man das Modul in diesem Modus über den HDC an, so muss immer ein Schaltbefehl an Ausgang 1 oder 2 und ein Schaltbefehl an Ausgang 3 oder 4 folgen.

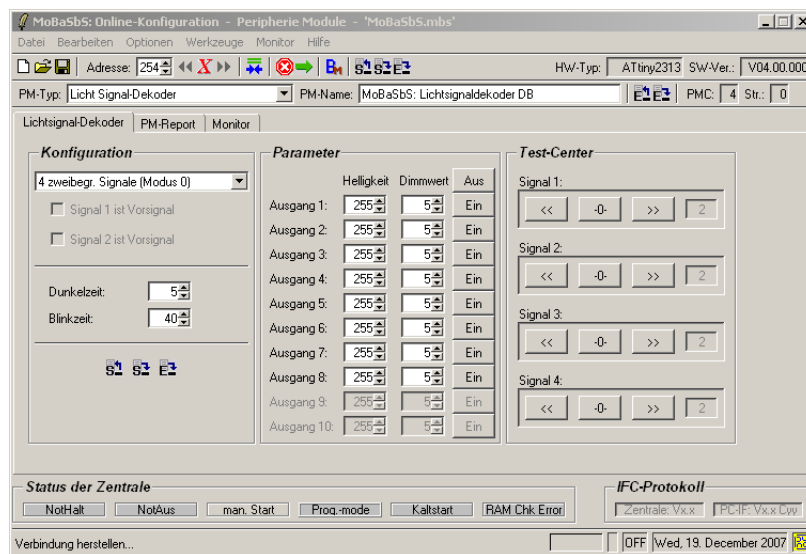
Diese beiden Schaltbefehle müssen innerhalb von 3 Sekunden gesendet werden, sonst werden diese nicht ausgewertet. Dies ist erforderlich, damit nicht durch einen älteren Schaltbefehl ein falsches Signalbild aufgedimmt wird. Diese 3 Sekunden sind auch mit einer Steuerungssoftware nicht zu überschreiten.

Beim Testen der eingestellten Werte im Konfigurationstool werden diese beiden Befehle bereits



automatisch gesendet.

12.6.8 Lichtsignal-Decoder



Lichtsignal-Decoder

Der Lichtsignal-Decoder (LDEC) steuert ein- und mehrbegriffige Lichtsignale an. Er setzt Signale mit LEDs voraus.

Am LDEC können keine Initialwerte eingestellt werden, deshalb muss die Beschaltung der LEDs ggf. an der Decoder-Hardware geändert werden.

Den Lichtsignal-Decoder gibt es in zwei Ausführungen:

- Lichtsignal-Decoder-Platine mit ATmega8
- FET- oder ULN-Decoder mit ATtiny2313

Für den LDEC können verschiedene Modi eingestellt werden:




Modus 0: 4 zweibegriffige Signale

Modus 1: 2 mehrbegriffige Signale

Modus 2: Hauptsignal und Vorsignal mit Dunkelschaltung

Der Standardfall für das MoBaSbS-Konfigurationstool ist die ATmega8-Variante. Diese Version wird ausgewählt, wenn über das Konfigurationstool eine Adresse mit dem LDEC belegt wird. Wurde der LDEC mit einem ATtiny2313 realisiert, muss dies von dem Konfigurationstool erst 'gelernt' werden. Dazu wird die HW-Kennung aus dem FET- oder ULN-Decoder ausgelesen.

Dies geschieht beim Rescan der MoBaSbS. Wird ein ATtiny2313 erkannt, werden die Einschränkungen in der Maske wirksam. Eine Möglichkeit, die 2313er Variante von Hand direkt einzugeben besteht nicht.

Der Modus kann im linken Feld mit der Überschrift "Konfiguration" ausgewählt werden. Wird der Modus 1 (zwei mehrbegriffige Signale) ausgewählt, kann das Signal 1 und Signal 2 als Vorsignal konfiguriert werden (beeinflusst das Signalbild des angeschlossenen Signals). Darunter befinden sich die Felder für die Dunkelzeit (Zeit zwischen dem Verlöschen und dem Neuaufleuchten des neuen Signalbilds) sowie die Blinkzeit (war nur im nicht mehr unterstützten Modus 3 möglich). Über die bekannten Symbole kann die Konfiguration aus dem SRAM gelesen  bzw. geschrieben  werden. Beim Schreiben ins EEPROM mittels Knopf  wird zudem noch automatisch ein SW-Reset ausgelöst.

In der mittleren Spalte können die Parameter für jeden der 10 Ausgänge (8 bei ATtiny2313 Decoder) eingestellt werden. Für jede LED kann individuell die Helligkeit sowie der Dimmwert (eingestellt werden). Sinnvolle Werte für den Dimmwert liegen im Bereich von 1-10, max. kann hier 255 eingestellt werden. Über den rechten Knopf mit der Bezeichnung 'Ein' kann eine einzelne LED zum Leuchten gebracht werden. In diesem Zustand werden Veränderungen der Parameter für diese LED sofort übertragen und wirksam. Über den Knopf 'Aus' könne alle LEDs Dunkel geschaltet werden. Es kann immer nur eine LED in diesem Modus eingeschaltet werden, d.h. beim Einschalten einer neuen LED verlischt die zuvor eingeschaltete LED.

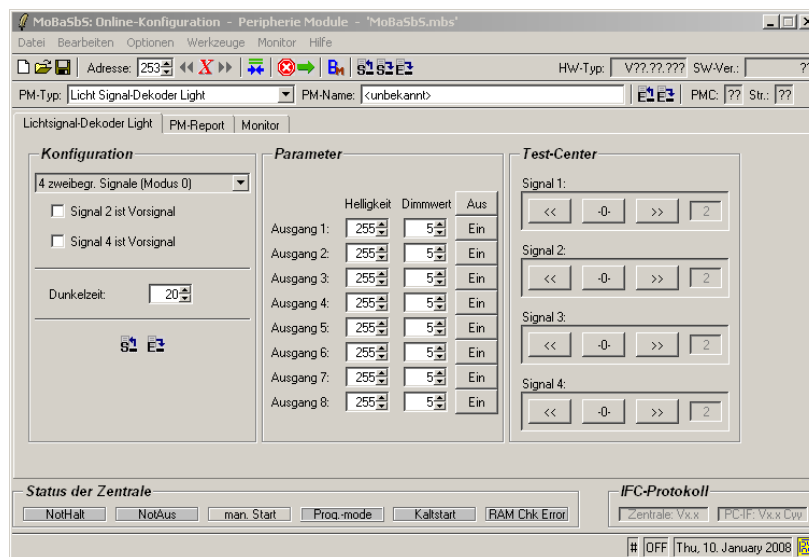
Vor dem Einschalten einer einzelnen LED muss erst das aktuelle Signalbild ausgeschaltet werden (Knopf 'Aus' drücken).

Auf der rechten Seite können die eingestellten Werte getestet werden. Hier gibt es, abhängig vom Modus, für jedes Signal drei Knöpfe:

- << Einen Schritt zurück im Signalbild-Zyklus
- 0- Default- bzw. Startzustand setzen
- >> Einen Schritt vorwärts im Signalbild-Zyklus

Im rechten Feld für jedes Signal wird der aktuelle Zustand als dezimale Zahl angezeigt.

12.6.9 Lichtsignal-Decoder Light



Lichtsignal-Decoder Light

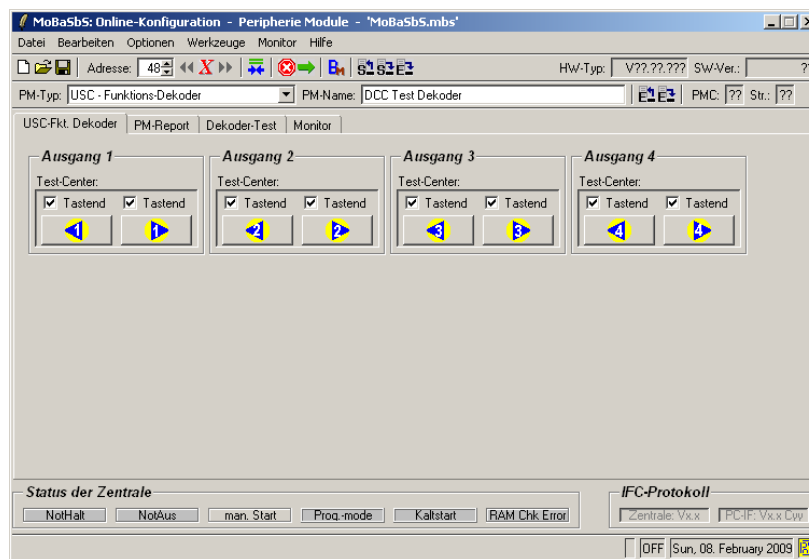
Vom Lichtsignal-Decoder gibt es auch eine Version mit reduziertem Funktionsumfang. Auf Basis eines Standard FET- oder ULN-Decoders (mit ATiny2313) gibt es in dieser Version nur den Modus 0 für bis zu 4 zweibegriffige Signale.

Der Decoder verwaltet zwei Signalgruppen

- Gruppe 1: Signal 1 und Signal 2
- Gruppe 2: Signal 3 und Signal 4

In jeder Gruppe kann das Signal mit der höheren Zahl als Vorsignal (VS) deklariert werden. In diesem Fall wird das Signal dunkel geschaltet, wenn das zugehörige Hauptsignal (HS) auf 'Halt' steht.

12.6.10 USC-Funktions-Decoder



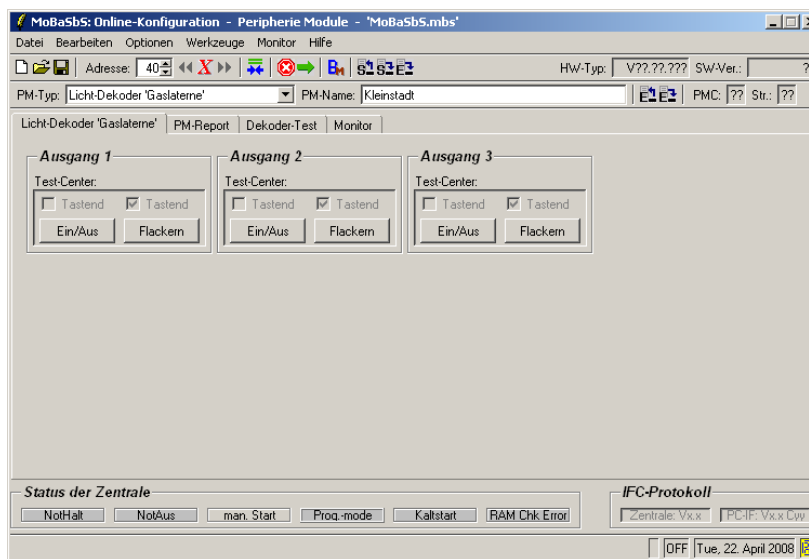
DCC/MM-Format-basierte Funktions-Decoder

Über dieses Register können ausschließlich manuell USC-gesteuerte Funktionsdecoder kontrolliert werden; ein Erkennen der Decoder im System ist nicht möglich

Es steht für diese Art von Decoder ausschließlich das Test-Center zur Verfügung, ein Konfiguration sowie das Auslesen von möglichen Konfigurationen ist nicht möglich. Ein Descriptor kann innerhalb der Konfigurations-Software vergeben und in der der zentralen Datei *.mbs gespeichert werden. Ein Übertragen in die Decoder-Hardware ist nicht möglich.

Das Feld 'Test-Center' stellt eine einfache Möglichkeit dar, die aktuelle Konfiguration zu testen. Drücken des Knopfes 'Kontakt x' sendet den Befehl zum Schließen des Kontaktes x, das Loslassen des Knopfes sendet den Befehl zum Öffnen des Kontaktes (Tastfunktion). Wird die Option 'Tastend' für einen Kontakt deaktiviert, bewirkt jedes Drücken und Loslassen eines Knopfes 'Kontakt x' eine Zustandsänderung zwischen 'geschlossen' und 'offen' für den entsprechenden Kontakt (Schaltfunktion).

12.6.11 Licht-Decoder ‚Gas-Laterne‘



Licht-Decoder ‚Gas-Laterne‘

Der Lichtdecoder für Gas-Laternen stellt 3 getrennte Kreise für eine Gruppe von Gas-Laternen bereit. Innerhalb eines Kreises verhalten sich alle Laternen bzgl. Flackern sowie Ein-/Ausschaltverhalten gleich.

Für diesen Decoder gibt es keinerlei spezielle Konfigurationsmöglichkeiten, lediglich der Descriptor kann vergeben bzw. verändert werden.

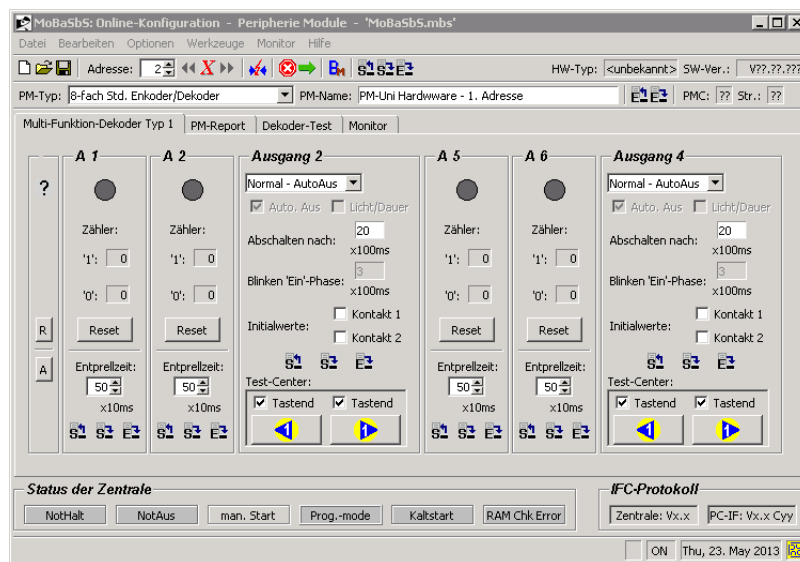
Das Feld 'Test-Center' stellt eine einfache Möglichkeit dar, jede der drei Ketten unabhängig zu testen. Drücken des Knopfes 'Ein/Aus' (rastende Funktion) sendet den Befehl zum Ein- bzw. Ausschalten der Kette. Durch den Druck auf die Taste Flackern (tastende Funktion) kann manuell ein Flackern ausgelöst werden.

12.6.12 Multi-Funktions-Dekoder (PM-Uni)

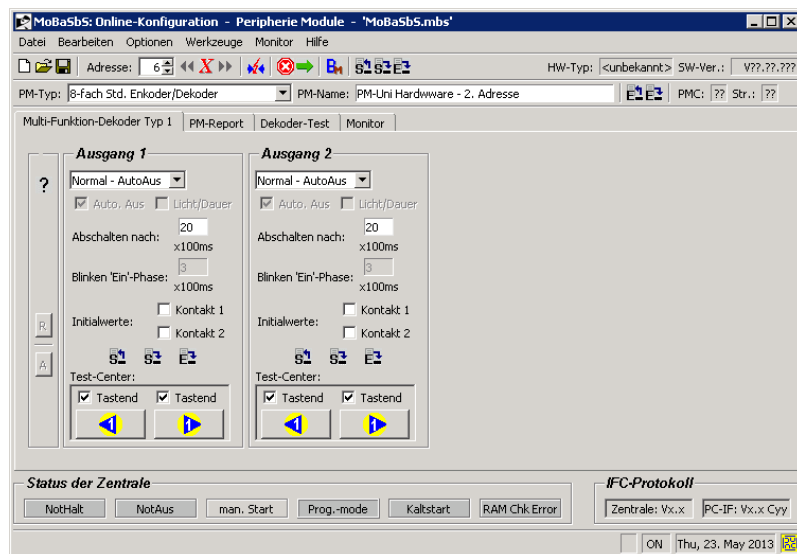
Basierend auf eine neuen Universal PM-Modul Platine, genannt PM-Uni, ist es jetzt möglich, verschiedene Dekoder-Typen auf ein und derselben Hardware zu implementieren. Dabei sind folgende Kombinationen möglich:

Adresse	Servo-Dekoder	Standard Dekoder	Standard Enkoder
1. Adresse	4-fach Servo-Dekoder		
		max. 4 Paare	max. 4 Paare
2. Adresse	4-fach Servo-Dekoder		
		max.2 Paare	max.2 Paare

Hierbei ist es möglich, für jede Adresse entweder die Funktion "4-fach Servo-Dekoder" zu programmieren oder eine Kombination aus Standard-Dekoder-Funktion und Standard-Enkoder-Funktion. Hierbei sind für die 1. Adresse max. 4 Paare in beliebiger Kombination möglich, für die 2. Adresse können max. 2 Paare konfiguriert werden (bedingt durch die Gegebenheiten der PM-Uni Hardware).



PM-Uni 1.Adresse (Primär-Adresse)



PM-Uni 2.Adresse (Sekundär-Adresse)

Die Konfiguration für jede Adresse erfolgt über ein spezielles Werkzeug unter "Werkzeuge > Peripherie Module > Multi-Funktions Module > MFM-1 konfigurieren. Hier ist es auch möglich, die 2. Adresse zu aktivieren oder deaktivieren

Von Ulrich Born wurden Zusatzplatinen für den Einsatz von Stromfühlern und Herzstückpolarisation entwickelt.

Die Platinen passen auf die Anschlüsse von PM_UNI.

Für Stromfühler habe ich eine 8-fach Platine, somit wird nur eine Adresse belegt. Die Platine wird auf PM_UNI gesteckt. Als Software kann die „alte“ ENC8 für Atmega8 verwendet werden, der 5-polige Pfostenfeldstecker ist dafür an der bezeichneten Stelle von unten an die Platine zu löten. Für die neue UNI-Software ist der andere bezeichnete Anschluss zu verwenden, ebenfalls von unten zu löten. Der 3. Anschluss ist immer auch von unten zu bestücken. Auf PM_UNI sind die 5-poligen Buchsen an den korrespondierenden Stellen von oben einzulöten. Eine 12-fach Variante habe ich zwar auch entworfen, das ist aber ein ziemliches Monstrum, ich werde sie nicht benutzen.

Die Platine für die Herzstückpolarisation hat ein etwas ungewöhnliches Format, dies deshalb, weil 2 Platinen unter PM_UNI montiert werden können (eine um 180 Grad gedreht). Deshalb werden die 6-fach Pfostenfeldstecker auf der Platine von oben montiert, die 6-fach Buchsen auf PM_UNI von unten. Damit können 4 oder 8 Herzstücke polarisiert werden. Als Software ist SDEC8 mit Herzstückpolarisierung (von Uwe) oder der UNI-Code verwendbar, die Bestückung ändert sich nicht.

Die Platinen sind einseitig und werden mit 8mm Abstandsrollen und M3 Schrauben auf bzw. unter PM_UNI befestigt.

Weitere Informationen zur Konfiguration stehen im Anhang (siehe Kap. 13.6).

12.7 Werkzeuge

Das MoBaSbS-Konfigurationstool bietet eine ganze Reihe von hilfreichen Tools, mit denen das gesamte MoBaSbS-System analysiert und einzelne MoBaSbS-Komponenten konfiguriert und getestet werden können.

12.7.1 Ändern einer Modul-Adresse

Dieses Werkzeug steht sowohl für Basis-Module als auch für Peripherie-Module zur Verfügung. Es kann die Adresse eines Moduls im eingebauten Zustand geändert und das entsprechende Modul durch einen Software-Reset neu gestartet werden. Dieses ist insbesondere für bereits in der Modellbahnanlage eingebaute PM-Module sehr von Vorteil.

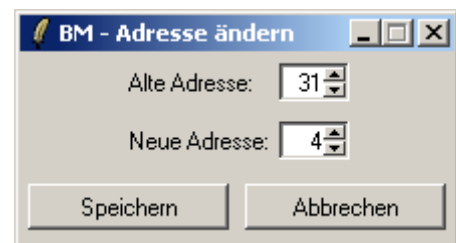
Basis-Module

Den Modulen BMC (Adresse 0), IFC (Adresse 1) kann keine neue Adresse zugewiesen werden. Andere Module können ebenfalls nicht auf diese Adressen umprogrammiert werden.

Die Adresse des USC, PMC sowie des HDC kann im Bereich 2-30 verändert werden. Der Adresse 31 kommt ein Sonderstatus zu und sollte nicht verwendet werden.

Für die Neuprogrammierung einer Basis-Modul-Adresse sind die alte, existierende Adresse und die neue Adresse für das zu verändernde Modul einzugeben. In beiden Eingabefeldern sind nur Eingaben im maximal zulässigen Adressbereich möglich.

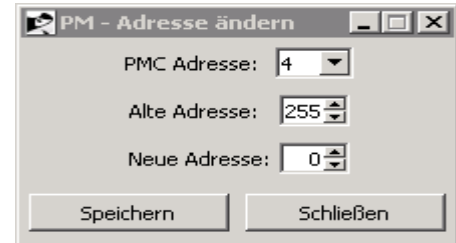
Durch Druck auf den Knopf 'Speichern' wird der Umprogrammierungsvorgang ausgelöst. Dabei wird zunächst Existenz und Modultyp des Moduls unter der aktuellen ('alte') Adresse ermittelt und anschließend geprüft, ob die neue Adresse noch nicht belegt ist. Wenn beide Prüfungen erfolgreich waren, wird die neue Adresse programmiert und ein Software-Reset in diesem Modul ausgelöst. In allen anderen Fällen wird der Vorgang abgebrochen und eine entsprechende Fehlermeldung ausgegeben.



Peripherie-Module

Die Adresse eines Peripherie-Moduls kann im Bereich 0-255 beliebig verändert werden. Für die Umprogrammierung einer Peripherie-Modul-Adresse sind die alte, existierende Adresse und die neue Adresse für das zu verändernde Modul einzugeben. In beiden Eingabefeldern sind nur Eingaben Bereich 0-255 möglich.

Durch den Druck auf den Knopf 'Speichern' wird der Umprogrammierungsvorgang ausgelöst. Dabei wird zunächst Existenz und Modultyp des Moduls unter der aktuellen Adresse ermittelt und anschließend geprüft, ob die neue Adresse noch nicht belegt ist. Wenn beide Prüfungen erfolgreich waren, wird die neue Adresse programmiert und ein Software-Reset in diesem Modul ausgelöst, in allen anderen Fällen wird der Vorgang abgebrochen und eine entsprechende Fehlermeldung ausgegeben. Für Encoder wird zusätzlich ein Warnhinweis ausgegeben, wenn die neue Adresse nicht im Adressbereich von 0-127 liegt (Einschränkung in Traincontroller bis einschl. V5.5). Nach erfolgreicher Umprogrammierung muss ggf. der/die Peripherie-Modul-Controller neu initialisiert werden (Hardware-Reset).



Hinweise:

- Es muss darauf geachtet werden, dass zur selben Zeit nicht zwei Module mit der gleichen Adresse im System aktiv sind.
- Alle Peripherie-Module sind nach der ersten Mikrocontroller-Programmierung immer auf die Adresse 255 eingestellt.
- Neue Module, bei denen der Mikrocontroller frisch programmiert wurde, dürfen nur nacheinander eingebaut werden. Nach jedem Einbau ist immer als erster Schritt die Adresse zu ändern!
- Nach einer Adressänderung muss ein Rescan ausgelöst werden.

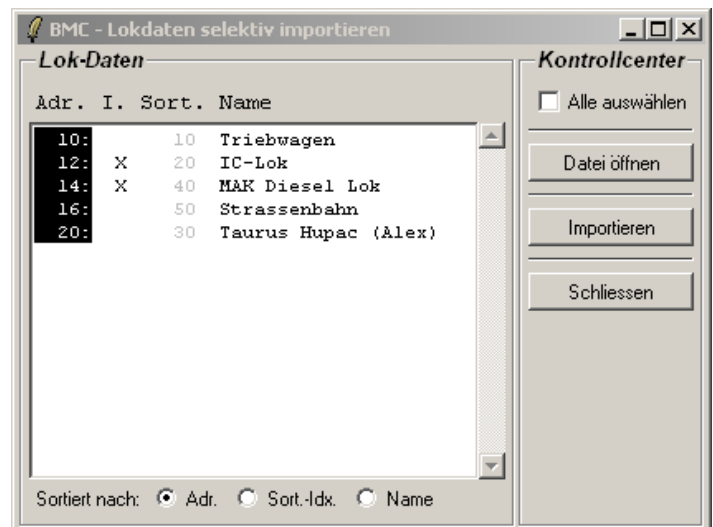
12.7.2 BM - Selektiver Lok-Daten Import

Mit Hilfe der selektiven Import-Funktion von Lok-Daten ist es auf einfache Weise möglich, Daten von einem MoBaSbS-System auf ein anderes System zu übertragen. Als Datenträger kann jedes beliebige Medium (z.B. Diskette, USB-Stick, etc.) dienen, aber auch Online können solche Daten übertragen werden (z.B. Mail).

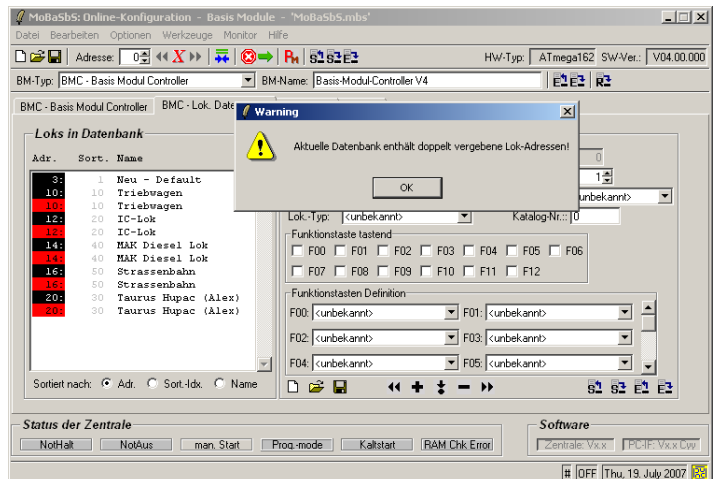
Beim Importieren von Lok-Daten muss zunächst die Datei mit den zu importierenden Lok-Daten über den Knopf 'Datei öffnen' eingelesen werden. Alle gelesenen Datensätze erscheinen danach im linken Textfenster gemäß aktueller Sortierung. Am unteren Ende des Text-Fensters kann zwischen verschiedenen Sortierungen gewählt werden:

Adr. Sortierung aufsteigend nach Lok-Adresse
 Sort.-Idx. Sortierung aufsteigend nach Sortierungs-Index
 Name Alphabetisch nach Namen

Es werden grundsätzlich nur die mit einem 'X' in der Tabelle markierten Einträge importiert. Ein Eintrag kann durch Klicken auf die Adresse (links Spalte) selektiert oder deselektiert werden. Durch Anklicken der Option 'Alle auswählen' werden entweder alle Einträge selektiert (Häkchen gesetzt) oder alle Einträge deselektiert (Häkchen nicht gesetzt). Der Eigentliche Importvorgang wird dann durch Drücken des Knopfes 'Importieren' gestartet.



Sind nach Abschluss des Importvorgangs Fehler aufgetreten (es wird zuerst der Importvorgang komplett abgeschlossen und danach die dann entstandene Datenbasis auf Fehler untersucht) erscheint die nebenstehende Meldung. Fehlerhafte Datensätze können jetzt vom Anwender zum Ändern ausgewählt und bearbeitet werden. Nach dem Abspeichern jeder Änderung wird die Datenbasis erneut überprüft. Solange dabei entscheidende Fehler gefunden werden, werden diese gemäß nebenstehendem Bild erneut gemeldet.



12.7.3 BM - Selektiver Lok-Daten Export

Mit Hilfe der selektiven Export-Funktion von Lok-Daten ist es auf einfache Weise möglich, Daten von einem MoBaSbS-System auf ein anderes System zu übertragen. Als Datenträger kann jedes beliebige Medium (z.B. Diskette, USB-Stick, etc.) dienen, aber auch Online können solche Daten übertragen werden (z.B. Mail).

Voraussetzung für den Export der Lok-Daten ist, dass diese dem Konfigurationsprogramm bekannt sind. Dies kann entweder durch Einladen einer Komplett-Datenbasis (Endung '.mbs') oder dem Auslesen der aktuellen BMC-Lok-Daten-Information erfolgen. Über die Export-Funktion können dann ausgewählte Datensätze in eine Datei (bekommt die Endung '.mbdb') geschrieben werden.

Beim Exportieren werden Datensätze aus der internen Lok-Datenbasis (d.h. die gerade der dem MoBaSbS-Konfigurationstool bekannten Datensätze) in eine Datei exportiert. Dabei werden wiederum nur markierte Datensätze beim Export berücksichtigt. Sollten sich in der Zwischenzeit Änderungen ergeben haben (das Fenster kann beim Bearbeiten geöffnet bleiben), kann durch Drücken des Knopfes 'Aktualisieren' die Auflistung der Datensätze neu generiert werden,

Ein Datensatz kann durch Klicken auf die Adresse (links Spalte) selektiert oder deselektiert werden.

Durch Anklicken der Option 'Alle auswählen' werden entweder alle Einträge selektiert (Häkchen gesetzt) oder alle Einträge deselektiert (Häkchen nicht gesetzt).



Lok-Daten exportieren

Nach dem Drücken des Knopfes 'Exportieren' öffnet sich ein Datei-Menü zur Auswahl des Dateinamens. Der Default-Name ist dabei 'BMC_LocoData.mdb' im Unterverzeichnis 'data'.

12.7.4 PM - Modul-Typ ändern

Für den 4-fach-Servo-Decoder kann der Modultyp geändert werden. Er kann wahlweise mit oder ohne Rückmeldungen eingesetzt werden.

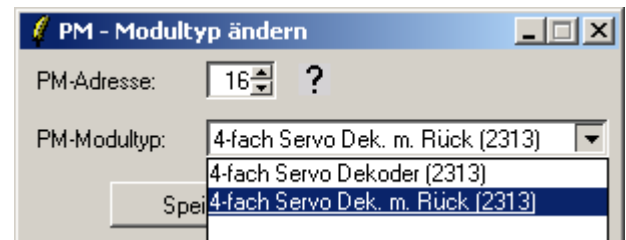
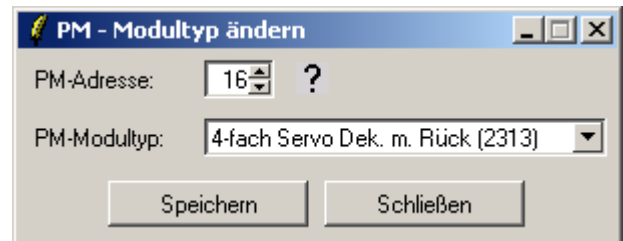
Hierzu ist lediglich eine Umkonfiguration im Eeprom notwendig, der Flash-Speicher braucht nicht neu beschrieben zu werden.

Der Typ eines Moduls kann nur im Online-Modus geändert werden, d.h. das MoBaSbS-Konfigurationstool muss mit der MoBaSbS verbunden sein und das zu ändernde Modul muss dem PMC bekannt sein. Bei Eingabe der Adresse wird der Typ des PM-Moduls unter der Adresse abgefragt und im Feld PM-Modultyp dargestellt. Nur bei Typen, bei dem Rückmeldefähigkeit ein- bzw. ausschaltet werden kann, erscheint dieses Feld nicht ausgegraut und man kann einen anderen Typ auswählen.

Sollte für eine Adresse der Modultyp nicht korrekt angezeigt werden, kann das Ermitteln des Typs für die aktuelle eingestellte Adresse durch drücken auf den Knopf **?** erzwungen werden.

Durch Drücken auf den Knopf 'Speichern' wird der neue Modul-Typ in das EEPROM des ausgewählten Moduls geschrieben. Das erfolgreiche Schreiben wird durch ein Fenster bestätigt, in dem auch der Rescan des PMC ausgelöst werden kann. Ein Rescan ist unbedingt notwendig, damit im Falle einer neu

eingestellten Rückmeldefähigkeit dieses rückmeldefähige Modul auch zyklisch vom PMC abgefragt wird.



PM - Kontakt Übersicht

The screenshot displays the 'PM - Kontakt Übersicht' window. The top section is a grid of 128 LEDs, organized into 16 rows (0-15) and 8 columns (1-8). The LEDs are represented by small circles, some of which are lit (yellow or grey). A tooltip is visible over LED 37, indicating its status. The bottom section is a 'Verlauf Monitor' window showing a log of contact events. The log entries are as follows:

Adresse	Bit	Wert	Std.	24-fach	Enk.	(ATMega162)	MoBaSbS	24fach	Rückmelder
000014	19:18:51	Adresse: 1	Bit: 8	Wert: 1	Std.	24-fach	Enk.	(ATMega162)	MoBaSbS: 24fach Rückmelder
000015	19:19:00	Adresse: 1	Bit: 8	Wert: 0	Std.	24-fach	Enk.	(ATMega162)	MoBaSbS: 24fach Rückmelder
000016	19:19:00	Adresse: 3	Bit: 5	Wert: 1	Std.	24-fach	Enk.	(ATMega162)	MoBaSbS: 24fach Rückmelder
000017	19:19:00	Adresse: 3	Bit: 5	Wert: 0	Std.	24-fach	Enk.	(ATMega162)	MoBaSbS: 24fach Rückmelder
000018	19:19:01	Adresse: 3	Bit: 5	Wert: 1	Std.	24-fach	Enk.	(ATMega162)	MoBaSbS: 24fach Rückmelder
000019	19:19:01	Adresse: 3	Bit: 5	Wert: 0	Std.	24-fach	Enk.	(ATMega162)	MoBaSbS: 24fach Rückmelder
000020	19:19:01	Adresse: 3	Bit: 5	Wert: 1	Std.	24-fach	Enk.	(ATMega162)	MoBaSbS: 24fach Rückmelder
000021	19:19:01	Adresse: 3	Bit: 5	Wert: 0	Std.	24-fach	Enk.	(ATMega162)	MoBaSbS: 24fach Rückmelder
000022	19:19:01	Adresse: 3	Bit: 5	Wert: 1	Std.	24-fach	Enk.	(ATMega162)	MoBaSbS: 24fach Rückmelder

The 'Steuerung' panel on the right contains the following buttons: Aktualisieren, Löschen, Speichern, and Schließen.

Kontaktübersicht

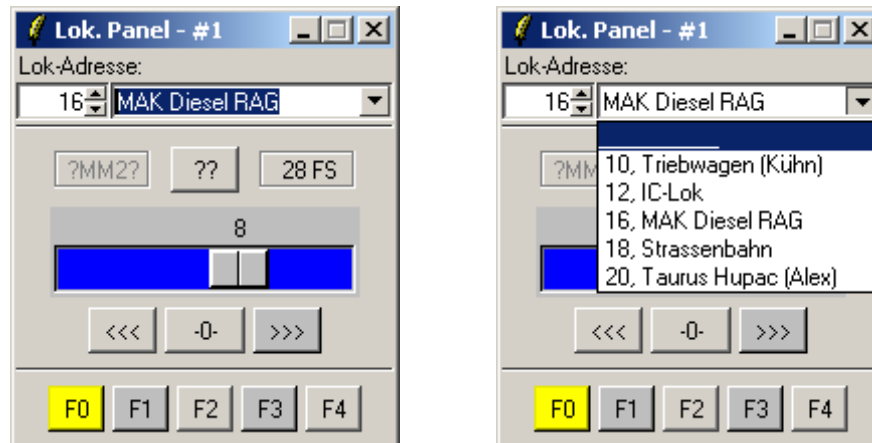
Die Kontaktübersicht ermöglicht einen genauen Überblick über den Zustand aller an die MoBaSbS angeschlossenen Rückmeldekontakte (Encoder). Das Fenster enthält zur besseren Übersicht zwei Teile, die je in einem eigenen Register realisiert sind. Gemäß klassischer Definition sind per Default die Kontakte der Adressen 0-127 im ersten Register sichtbar, das zweite Register enthält die Kontakte der Adressen 128-255.

Das Werkzeug kann zu beliebiger Zeit geöffnet werden und offen belassen werden. Es dient lediglich der Kontrolle und greift nicht in die aktive Datenkommunikation zwischen Konfigurationstool und MoBaSbS-Hardware ein. Nach dem Öffnen werden alle Schaltbefehl-Informationen protokolliert und im Textfenster unten links angezeigt. Gleichzeitig wird an entsprechender Stelle im Register die symbolische LED auf gelb ('1', Kontakt geschlossen) bzw. grau ('0', Kontakt offen) gesetzt. Bei aktivierter akustischer Unterstützung (Menu 'Optionen > Audio Unterstützung') wird das Schließen eines Kontaktes auch durch die Software akustisch unter Angabe der Modul-Adresse und des jeweiligen Kontaktes angezeigt (z.B. 8-fach Encoder auf Adresse 112, Bit 3: "PM-1-1-2<pause>A-3").

Das Protokoll im Textfenster kann zu jeder Zeit in einer Textdatei gespeichert (Knopf 'Speichern') oder gelöscht werden (Knopf 'Löschen'). Drücken des Knopfes 'Rücksetzen' bewirkt ein Rücksetzen aller Kontakte auf den Wert '0' (Kontakt offen) ohne das Protokollfenster zu verändern.

12.7.5 Lok Kontroll Panel

Das Loco Kontroll Panel (LCP) ermöglicht die Steuerung einer Lokomotive direkt aus dem Konfigurationstool. Es können beliebig viele LCPs geöffnet werden, die unabhängig voneinander sind. Ist in zwei LCPs die gleiche Adresse eingestellt, sind sie gleichberechtigt und zeigen beide identische Informationen.



Lok Kontroll Panel

In jedem Panel kann die Lok-Adresse direkt oder über das Pull-Down-Menu (rechtes Bild) eingegeben werden. Im Pull-Down-Menu werden alle mit Lok-Namen eingegebenen Lokomotiven mit ihrer Adresse aufgeführt. Der aktuelle Status wird bei der Anwahl direkt aus der MoBaSbS HW abgefragt.

Im mittleren Teil kann über den Schieberegler die Geschwindigkeit der Lokomotive eingestellt werden. Ein Druck auf die mittlere Taste "-0-" bewirkt ein normales Anhalten der Lok, während ein Druck auf die Fahrtrichtungstaste "<<<" bzw. ">>>" einen Not-Stopp auslöst.

Im unteren Teil des Panels sind die wichtigsten Funktionstasten (ausschließlich F0-F4) erreichbar. Ist eine Taste gedrückt, wechselt sie die Farbe zum Zeichen der Aktivierung. F0 ist dabei gelb unterlegt, die anderen Tasten dunkelgrau.

Alle Änderungen am HDC werden direkt im LCP dargestellt, wenn bei beiden die gleiche Lok-Adresse eingestellt ist. Ebenso zeigt das HDC sofort Änderungen über das LCP auf Display und den LEDs an.

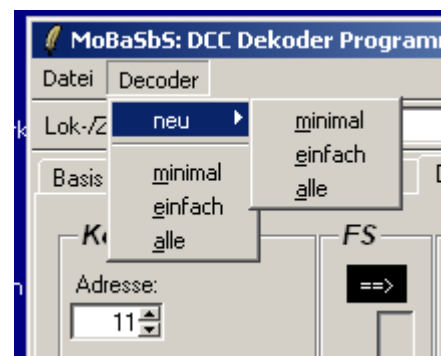
12.7.6 DCC Programmier

Mit Hilfe dieses Werkzeugs ist es möglich, einzelne oder alle CVs eines Lok-Decoders im DCC Format aus dem Decoder zu lesen, zu verändern und wieder in den Decoder zurück zu schreiben. In der aktuellen Version werden CVs nur im Bereich von 1 bis 256 unterstützt. Über verschiedene Register im Werkzeug können unterschiedliche Aufgaben erledigt werden:

- Ändern der wichtigsten CV-Werte („Basis“)
- Ändern der Zuordnung von Decoder-Funktionen auf Funktionstasten
- Ändern von bitweise definierten CVs
- Ändern von einzelnen CVs
- Übersicht über alle CVs
- Decoder-Test
- Vorlagen für DCC-Lok-Decoder

Über das Menu 'Datei' ist es möglich, DCC-Decoder-Informationen (CV-Werte) einer bestimmten Lok bzw. eines Zuges zu speichern und wieder zu laden. DCC-Decoder-Dateien erhalten die Datei-Erweiterung '*.dcc' und werden im Unterverzeichnis 'data' gespeichert. Zusätzlich können Vorlagen für DCC-Decoder (Datei-Erweiterung '*.dct') geladen werden, die Informationen über die für einen bestimmten Decoder-Typ definierten CVs enthalten.

Weitere Informationen zu den Vorlagen-Dateien und deren Format stehen am Ende dieses Abschnittes.



Über das Menu 'Decoder' kann ein neuer Decoder-Typ eingestellt werden (Untermenu 'neu') bzw. der Typ des aktuellen Decoders verändert werden. Je nach gewähltem Typ ('minimal', 'einfach', 'alle') werden mehr oder weniger CV Felder aus der Kategorie 'Standard CVs' freigegeben oder gesperrt. Gesperrte CVs werden weder gelesen, noch geschrieben noch geprüft

minimal	nur CV1 kann neben CV29 bearbeitet werden, gültig für Funktionsdecoder, etc.
einfach	alle CVs außer CV6 und 19 können editiert werden. CV17&18 sind optional editierbar, wenn im unter Optionen die Verwendung von langen Adressen voreingestellt ist.
alle	alle CVs dieses Registers können editiert werden

Über den verschiedenen Registern befindet sich die Zeile

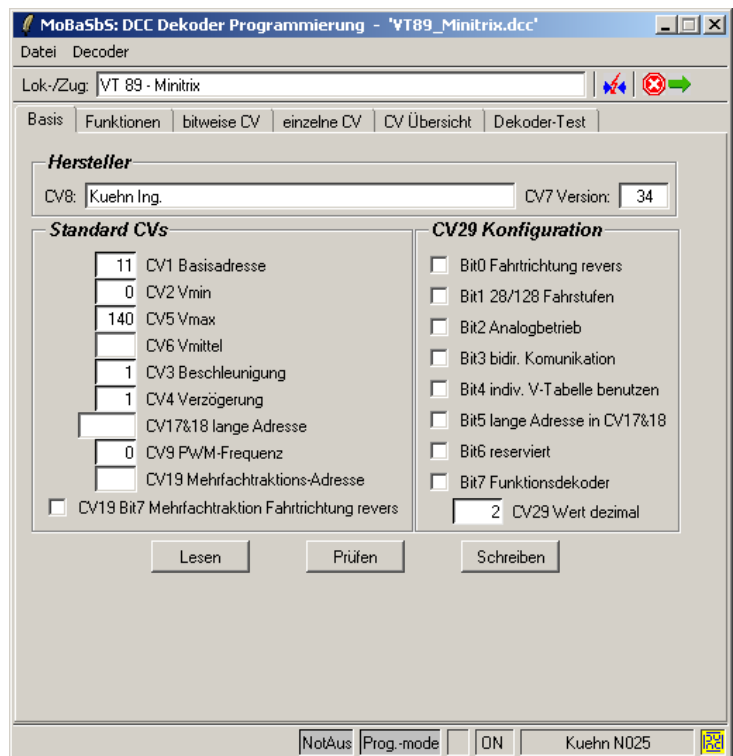


Darin kann zusätzlich zu den einzelnen CV-Werten eine allgemeine Beschreibung der Lok bzw. des Zuges eingegeben werden. Diese Zeile ist auf allen Registern permanent sichtbar und wird zusammen mit allen CV-Werten in der DCC-Decoder-Datei gespeichert. Am rechten Rand befinden sich zusätzlich noch die Knöpfe, um die Verbindung zwischen Konfigurations-Software und DPC- bzw. MoBaSbS-Hardware herzustellen bzw. zu lösen (⚡) sowie für 'AllesAus' (⊗) bzw. 'Alles An' (➡).

In der Statuszeile am unteren Ende des Fensters befindet sich etwa in der Mitte zwei Felder 'NotAus' sowie 'Prog.-Mode', die den aktuellen Status des DPCs bzw. der MoBaSbS anzeigen. Diese Felder sind Kopien der Felder aus dem Hauptfenster und zeigen immer den gleichen Status an.

Ändern der wichtigsten CV-Werte („Basis“)

Über das Register 'Basis' kann auf die wichtigsten CV-Werte zugegriffen werden, die im Wesentlichen die Grundfunktionen des Decoders betreffen. Über die drei Knöpfe am unteren Rand des Fensters können alle CVs in diesem Register ausgelesen, geprüft oder geschrieben werden. Gelesene CV-Werte werden sofort nach dem Lesen im Feld für die CV dargestellt und können anschließend bearbeitet werden. Über den Knopf "Schreiben" werden alle CVs der Reihe nach wieder in den Decoder geschrieben, wobei die DPC-Hardware jeden Schreibvorgang selbstständig durch einen Vergleich auf Erfolg überprüft. 'Prüfen' bewirkt ein Lesen der CV-Werte, die mit den aktuell angezeigten Werten verglichen werden. Dabei werden die angezeigten Werte nicht verändert, nach jedem CV wird aber ein Unterschied in einem Fenster als Fehler angezeigt.

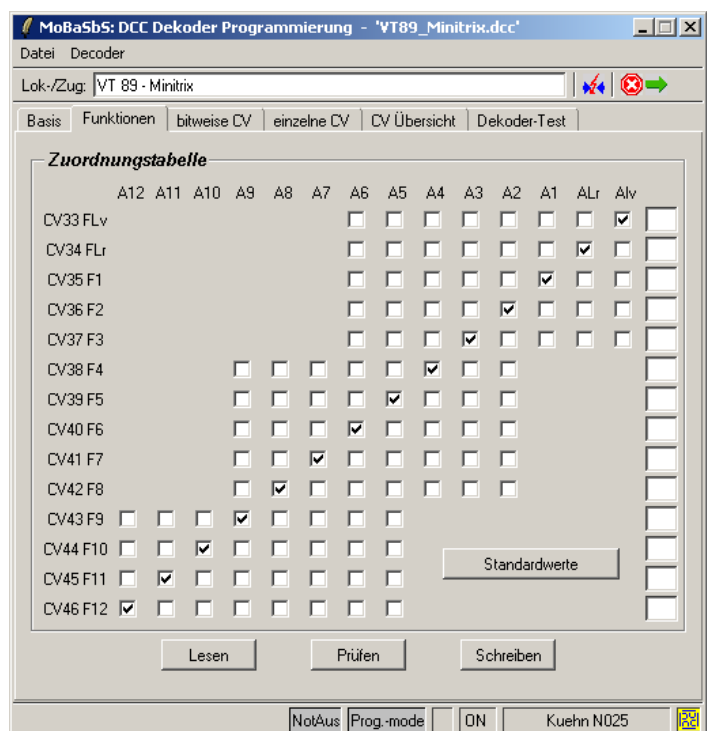


Die Herstellerkennung (Feld 'Hersteller', CV7&8) kann nur aus dem Decoder ausgelesen werden, nicht aber von Hand editiert werden. Alle anderen CV-Werte können durch Editieren bzw. klicken mit der Maus gesetzt oder rückgesetzt werden

Ändern der Zuordnung von Decoder-Funktionen auf Funktionstasten

Mit diesem Register kann die Zuordnung von Decoder-Funktionen auf die Funktionstasten analysiert und verändert werden. Dabei werden gemäß DCC-Spezifikation immer alle CV im Bereich 33 bis 46 gelesen, geschrieben und geprüft. Die jeweilige Aktion kann durch Druck auf einen der Knöpfe am unteren Ende des Registers ausgelöst werden. Drücken des Knopfes 'Standardwerte' erzeugt die im Bild dargestellten Zuordnungen.

Dieses Register orientiert sich strikt an der DCC Spezifikation RP-9.2.2 der NMRA. Bei einigen Decodern verschiedener Hersteller kann die Funktionszuordnung bzw. die Belegung der Register-Bits abweichen und das Bearbeiten der Zuordnungen über dieses Register teilweise oder ganz unmöglich sein. Dann bitte auf das Register für eine einzelne CV ausweichen und die Register des entsprechenden Decoder manuell lesen.

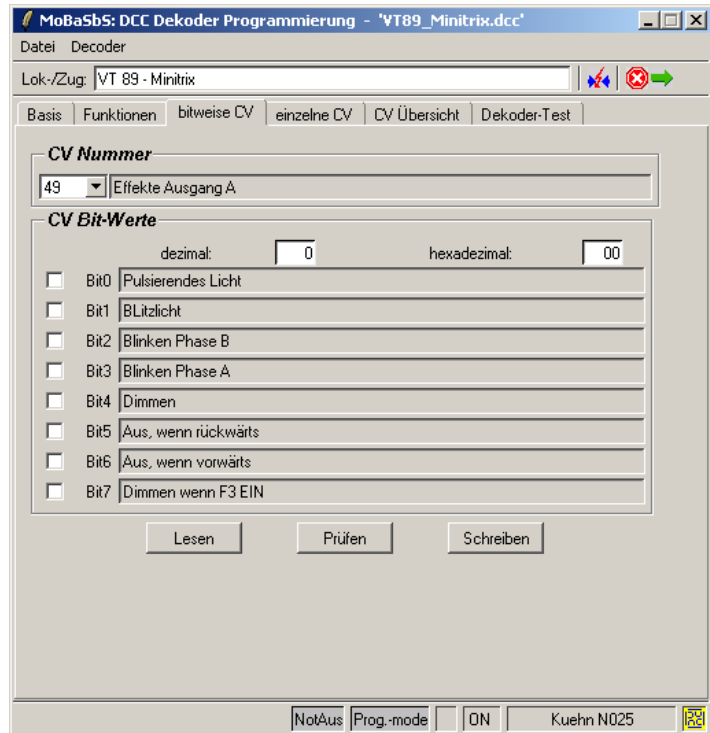


Ändern von bitweise definierten CVs

Über das Register 'bitweise CV' können ausschließlich CV Register gelesen, bearbeitet und geschrieben werden, die auch als bitweise Register definiert sind.

Im oberen Abschnitt unter CV-Nummer kann aus einer Liste von bitweise definierten Registern ausgewählt werden. Diese Werte können aus dem Decoder gelesen und geschrieben werden.

In dieser Maske könne die CV-Werte nur durch Setzen oder Rücksetzen der Kästchen verändert werden. In den Feldern 'dezimal' bzw. 'hexadezimal' wird die aktuelle Bit-Kombination lediglich dezimal bzw. hexadezimal ausgegeben.



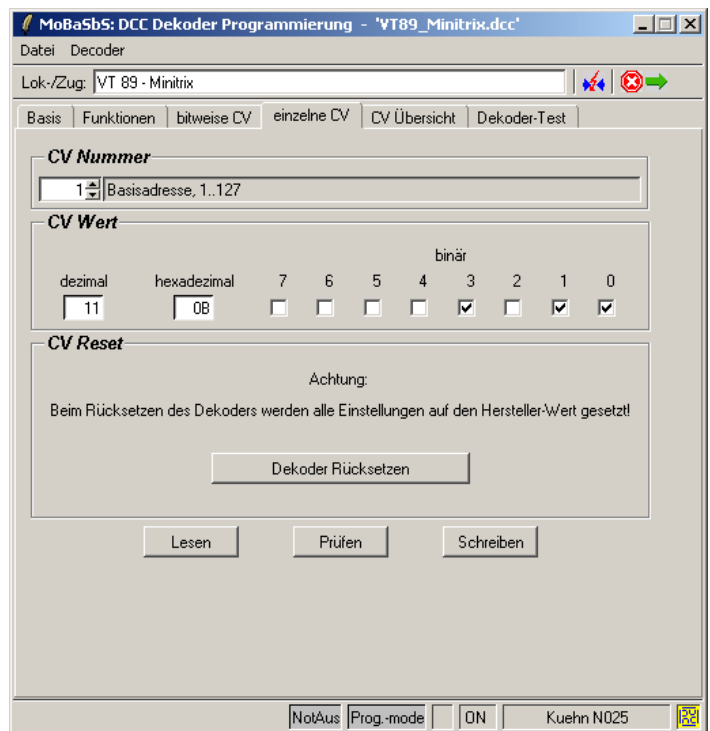
Ändern von einzelnen CVs

Einzelne CVs können im Register 'einzelne CV' gelesen, editiert und wieder geschrieben werden. Ein Prüfen ist ebenfalls hier möglich.

Im oberen Abschnitt unter CV-Nummer kann die zu bearbeitende CV ausgewählt werden. Wenn vorhanden erscheint die Beschreibung der CV automatisch im rechten (nicht editierbaren Feld).

Ist die CV ausgewählt und ggf. ihr Wert aus dem Decoder gelesen, kann wahlweise der Wert dezimal, hexadezimal sowie binär editiert werden. Alle drei Methoden sind gleichberechtigt und können beliebig kombiniert werden.

Das Feld mit der Überschrift 'CV Reset' bietet die Möglichkeit, alle CVs des Decoders auf die ursprünglichen Herstellerwerte zurückzusetzen.



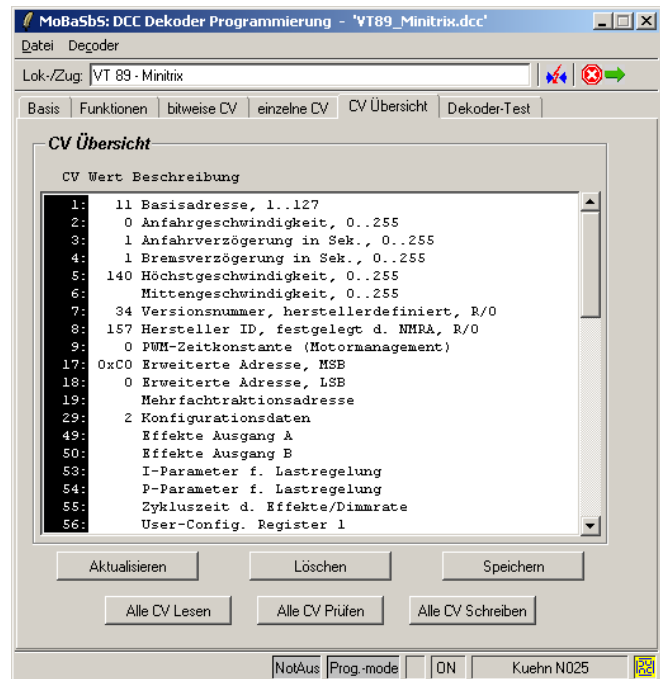
Übersicht über alle CVs

Eine Übersicht über alle CVs mit Beschreibung und Werten bietet dieses Register. Über die drei Knöpfe am unteren Rand des Registers stehen dem Anwender folgende Funktionen zur Verfügung:

Aktualisieren - Löscht das aktuelle Fenster, liest die CV-Werte aus der internen CV-Datenbank innerhalb der Konfigurationssoftware neu aus und stellt das Ergebnis dar.

Löschen - Löscht das aktuelle Fenster. Danach kann mit 'Aktualisieren' das Fenster neu aufgebaut werden.

Speichern - Hiermit kann der aktuelle Fensterinhalt der CV-Übersicht in eine Text-Datei ausgegeben werden. Zusätzlich wird am Anfang der Datei die Information über die Lok. bzw. den Zug sowie den verwendeten Decoder-Typ mit ausgegeben



Wenn eine Decoder-Vorlage geladen ist (also eine *.dct-Vorlage oder eine Decoderdatei *.dcc basierend auf einer Vorlage) stehen drei weitere Knöpfe zur Verfügung. Je nach Komplexität kann das Lesen mehrere Minuten dauern und der Fortschritt wird links unten in der Statuszeile angezeigt. Während dieser lang laufenden Aktionen erscheint ein separates Fortschrittsfenster, wodurch alle weiteren Aktionen verriegelt sind. Das Drücken auf den Knopf 'Abbruch' im Fortschrittsfenster beendet die aktuelle lang laufende Aktion (CV lesen, prüfen oder schreiben). Dabei wird ein gerade begonnener CV-Zugriff noch vollständig zu Ende geführt.

Alle CV Lesen - Nach Drücken dieses Knopfes werden alle für den aktuellen Decoder-Typ definierten CV-Werte ausgelesen.

Alle CV Prüfen - Bei dieser Funktion werden die Werte aller in der Software gesetzten CV-Werte mit den aktuellen Werten im Decoder verglichen. Bei unterschiedlichen Werten, wird eine Fehlermeldung ausgegeben.

Alle CV Schreiben - Alle CV-Werte des Decoders werden mit den in der Software gespeicherten bzw. geänderten CV-Werte überschrieben.




Hinweis: Die Funktion 'Speichern' kann für das Drucken benutzt werden, da eine Druckfunktion in der Konfigurations-Software nicht zur Verfügung steht

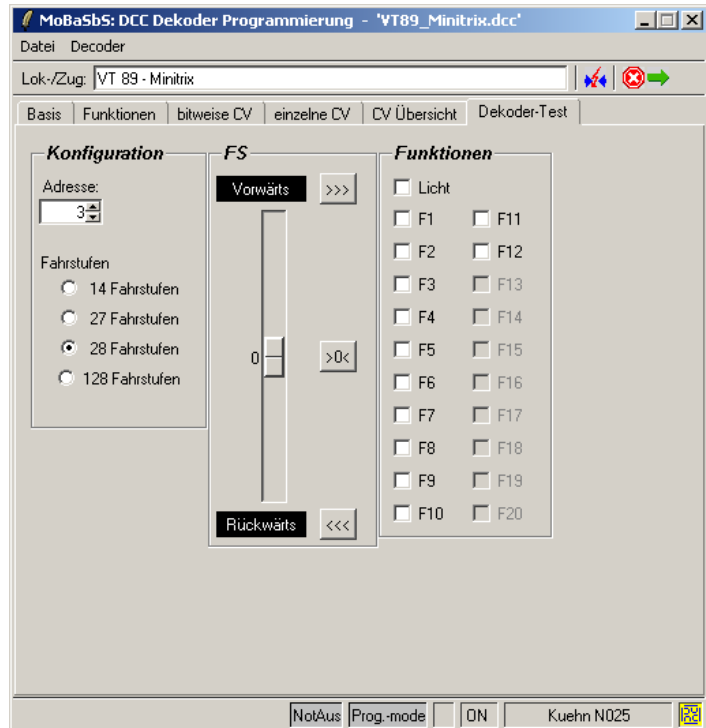
Decoder-Test

In diesem Register kann schnell mal der so eben programmierte Decoder getestet werden. Ähnlich dem Lok-Kontroll-Panel besteht hier die Möglichkeit, Adresse und Anzahl der FS einzustellen (Achtung: BMC Lok-DB wird hier nicht verwendet!).

Danach hat man wie gewohnt die Möglichkeit, die Lok vorwärts wie rückwärts fahren zu lassen. Auch können die Funktionstasten F0 (Licht) bis F12 ein- und ausgeschaltet werden.

Der Zugriff auf die Funktionstasten F13-F20 ist vorbereitet, wird aber in der aktuellen Version des DPC noch nicht unterstützt.

Über die Tasten  bzw.  wird die Geschwindigkeit der Lok mit jedem Drücken um 1 erhöht bzw. um 1 verkleinert. Somit kann schrittweise beschleunigt oder gebremst werden. Durch Drücken der Taste  wird die Lok auf Fahrstufe 0 gesetzt



Vorlagen für DCC-Lok-Decoder

Im Unterverzeichnis 'data' wird eine Vorlagendatei bereitgestellt, die von jedem Anwender als Grundlage für die Erstellung von Decoder-spezifischen CV-Informationen genutzt werden kann. Wird eine solche Datei verwendet, sind ausschließlich die in der Datei definierten CVs in den Fenstern aktiviert und können geschrieben und gelesen werden. Weitere Einzelheiten sind im Kopf der Vorlagendatei zu finden.

Eine Kontrolle der Datei kann durch Einlesen in den DPC erfolgen. Dabei werden Syntax-Fehler direkt ausgegeben. Danach kann in der Übersicht das erfolgreiche und fehlerfreie Einlesen kontrolliert werden. Als Hilfe gibt es hier die Möglichkeit, im DPC Fenster unter dem Menu-Punkt 'Decoder' die Option 'Debug-Template' anzuklicken. Ist diese Option aktiviert, wird beim Einlesen jede gelesene Zeile im Konsolenfenster (zu öffnen über das Menu 'Hilfe -> Konsole ein' im Hauptfenster des Programms) angezeigt. Beim Neustart der Software bzw. des DPC ist diese Option automatisch wieder gelöscht und muss erneut gesetzt werden.

Für Dekoder-Templates gibt es im Forum unter 'Dateien / DCC Dekoder Templates' die Möglichkeit, bereits ausgefüllte Templates auszutauschen.

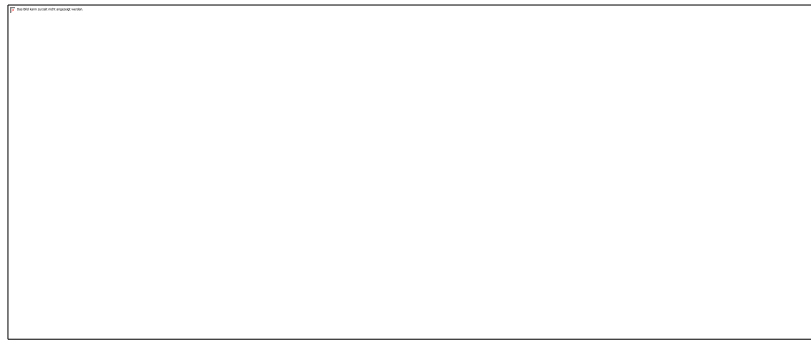
Datei ‚DCC DecTemplate.dct‘

```

#-----
# MoBaSbS V4.0 - D C C D e c o d e r C V D a t e n
#
# Datum: <Datum der letzten Änderung, z.B. "Dienstag, 13. Oktober 2009 17:42:59">
# Autor: <Name des Autor>
# Format: .dct
# Hinweis: Diese Datei enthält Informationen über alle CV Parameter eines
# speziellen DCC Dekoder Typs. Sie bestimmt, welche CVs für den
# Dekoder definiert und programmiert werden können. Für CVs,
# die nicht definiert sind, darf es keine Einträge geben. Diese
# Datei ist von Hand zu erstellen und kann in den DCC Programmer
# zur weiteren Verwendung geladen werden. Alle Zeilen, die nicht
# mit '#' beginnen, werden von der Software gelesen und analysiert.
# Entscheidend ist das Gleichheitszeichen, denn links davon steht
# der Name des Parameters, rechts davon sein Wert. Wichtig: Das
# '=' darf recht und links NICHT von Leerzeichen umgeben sein!
# Jede Zeile muß in der ersten Spalte mit dem ersten Zeichen be-
# ginnen (keine Leerzeichen).
# Teil 1: Basic CV Data und CV Werte
# Hier bitte XXX durch den DCC Dekoder Typ ersetzen, z.B.
# 'Kuehn N025' oder 'ESU LokPilot V3.0 micro' (Hochkommata
# bitte weglassen). Sonst in diesem Bereich bitte nichts
# ändern!
# Teil 2: CV Beschreibung für diesen DCC Decoder Typ
# Dieser Teil enthält die Liste mit CVs, die für diesen
# Dekoder definiert sind. Jede Zeile dieses Abschnitts
# muß mit "CVDescr=" beginnen. Danach kommt die Nummer
# der CV direkt nach dem Gleichheitszeichen. Das nächste
# Element enthält die Beschreibung der CV, die durch '{}'
# eingeschlossen sein muß. Muß die Beschreibung selbst
# obige Klammern enthalten, müssen diese im Text durch
# einen vorangestellten BackSlash (\) markiert werden.
# Die Länge des Beschreibungstextes sollte ~55-60 Zeichen
# nicht überschreiten.
# Teil 3: CV Bit Map Beschreibungen für diesen Dekoder Typ
# Dieser Teil enthält zusätzliche Informationen zu CVs,
# die nicht einen gesamten Wert enthalten sondern Info's
# zu einzelnen Bits enthalten. Jede dieser Zeilen muß mit
# 'CVBitMap=' beginnen und nach dem Gleichheitszeichen
# die Nummer der CV enthalten. Die einzelnen Beschreibungen
# je Bit müssen immer durch Anführungsstriche Markiert sein,
# als Trennzeichen zwischen den Bits ist '|' zu setzen. Es
# müssen immer 8-Bits definiert werden, ist ein Bit unbenutzt,
# kann der Beschreibungstext entfallen, die Trennzeichen '|'
# müssen aber gesetzt werden (siehe CV50 unten).
#-----
# Basic CV Data und CV Werte
LocName=<Template für XXX Dekoder>
DecType=XXX
DCCType=einfach
CVData=35 {} 36 {} 37 {} 38 {} 40 {} 39 {} ....
#
# CV Beschreibung für diesen DCC Decoder Typ
CVDescr=1 {Basisadresse, 1..127}
CVDescr=2 {Anfahrgeschwindigkeit, 0..255}
#
# CV Bit Map Beschreibungen für diesen Dekoder Typ
CVBitMap=13 {"Funktion F1"}{"Funktion F2"}{"Funktion F3"}{"Funktion F4"}{"Funktion F5"}{...}
CVBitMap=14 {"Funktion FL(f)"}{"Funktion FL(r)"}{"Funktion F9(f)"}{"Funktion F10(f)"}{...}
CVBitMap=50 {"AC Analog Modus aus / ein"}{"DC Analog Modus aus / ein"}{||||}
#-----

```


12.7.7 Speicherinhalt BM/PM-Module (μ C-Speicher-Dump)



Speicherauszug

Mit Hilfe dieses Werkzeuges ist es möglich, sich den aktuellen Speicherinhalt (SRAM oder EEPROM) von Basis- oder Peripherie-Modulen anzusehen und ggf. für Debugzwecke zu speichern. Im Kontrollcenter rechts müssen zunächst die folgenden Optionen eingestellt werden, bevor der Speicher dann durch Drücken des Knopfes 'Speicher auslesen' gelesen und dargestellt wird:

Quelle: Quelle, aus der der Speicherinhalt gelesen werden soll
 ?????: keine Quelle ausgewählt
 BM-SRAM: Basis-Modul SRAM lesen
 BM-EEP: Basis-Modul EEPROM lesen
 PM-SRAM: Peripherie-Modul SRAM lesen
 PM_EEP: Peripherie-Modul EEPROM lesen

Mod.-Adr. Modul-Adresse
 BM-Module: 0 - 31
 PM-Module: 0 - 255

Mem.-Adr. Speicher-Adresse (Memory-Adresse), 2-Byte Zahl im hexadezimal-Format ohne führendes '0x' mit Leerzeichen dazwischen ('abcdef' ist ebenso erlaubt wie 'ABCDEF')

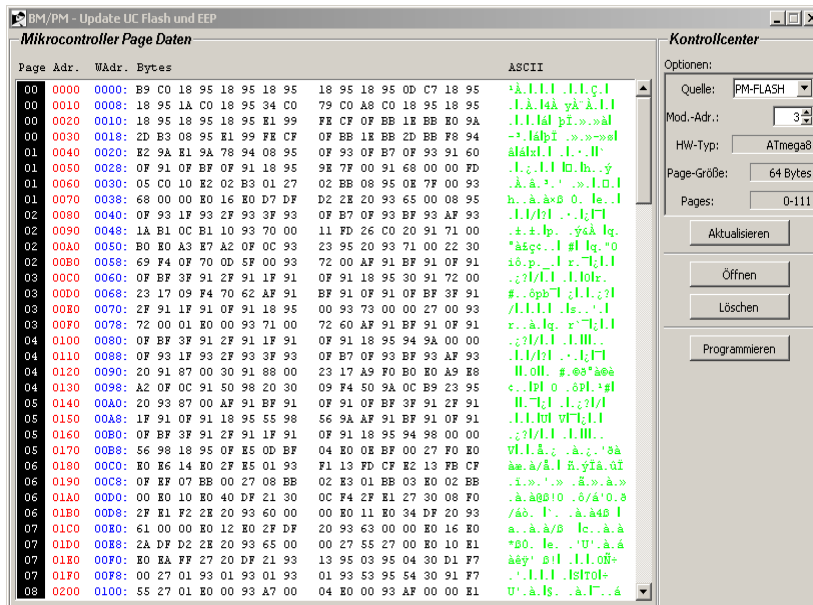
Zeilen: Anzahl der Zeilen in der Ausgabe, entspricht im Prinzip der Anzahl der Speicherstellen, die ausgegeben werden (16 Speicherstellen je Zeile), zur Auswahl stehen 8, 16, 32, 48 und 64 Zeilen

Der Speicherinhalt wird im Hexadezimal-Format zu je zwei Päckchen á 8 Bytes ausgegeben, zur Information wird am rechten Rand jeder Zeile noch das zu jedem Byte-Wert gehörende ASCII-Zeichen ausgegeben. ASCII-Steuerzeichen (ASCII-Wert < 32) werden dabei als '.' dargestellt. Das Fenster wird automatisch vor jedem neuen Auslesevorgang gelöscht und die Daten werden verworfen.

Der so gemachte aktuelle Speicherauszug kann über den Knopf 'Speichern' in einer ASCII-Text-Datei gespeichert und zu Debugzwecken weiterverwendet werden. Mit dem Knopf 'Löschen' kann der aktuelle Inhalt manuell gelöscht werden.

12.7.8 µC Flash Update (Bootlader)

Mit Hilfe der µC Flash Update-Funktion kann eine neue SW-Version in die Prozessoren übertragen werden. Dazu muss bereits eine Software-Version mit Bootlader-Funktion im µChip vorhanden sein (siehe Kap. 8.1 und 13.4)



Flash-Update

Im Kontrollcenter rechts müssen zunächst die folgenden Optionen eingestellt werden, bevor der UC aktualisiert werden kann

Quelle: Quelle bzw. Ziel, das aktualisiert werden soll:
 BM-FLASH: Basis-Modul FLASH aktualisieren (hex-File)
 BM-EEPROM: Basis-Modul EEPROM aktualisieren (eep-File)
 PM-FLASH: Peripherie-Modul FLASH aktualisieren (hex-File)
 PM-EEPROM: Peripherie-Modul EEPROM aktualisieren (eep-File)

Mod.-Adr. Modul-Adresse
 BM-Module: 0 - 31
 PM-Module: 0 - 255

Die darunter befindlichen Felder für HW-Typ, Page-Größe sowie Pages sind rein informativ und zeigen notwendige Parameter an, die für die ordnungsgemäße Programmierung erforderlich sind. Durch Drücken des Knopfes 'Aktualisieren' wird der Typ des Mikroprozessors ermittelt (d.h. Auslesen des EEPROMs des adressierten Moduls). Gleiches erfolgt auch noch automatisch direkt zu Beginn des Flash-Programmierungsvorgangs.

Wie im obigen Beispiel zu sehen, wird der Inhalt der Programmier-Datei (Intel-HEX-Format, Endung *.hex bzw *. eep) komplett eingelesen und in Pages umgebrochen. Je nach notwendiger Page-Größe (μ C abhängig) werden entweder je 4 Zeilen (Page-Größe 64 Bytes, ATmega8) oder 8 Zeilen zu einer Page zusammengefasst. Die erste Spalte zeigt an, zu welcher Page die jeweilige Zeile gehört. Spalte 2 und 3 zeigen die jeweilige Startadresse (erstes Byte) als reale Adresse (rot) bzw. als Schreib-Adresse (blau) an. Rechts davon sind die 16 Bytes zu je 2x8 Bytes zu sehen (schwarz) sowie die gleichen Bytes in ASCII-Form (grün).

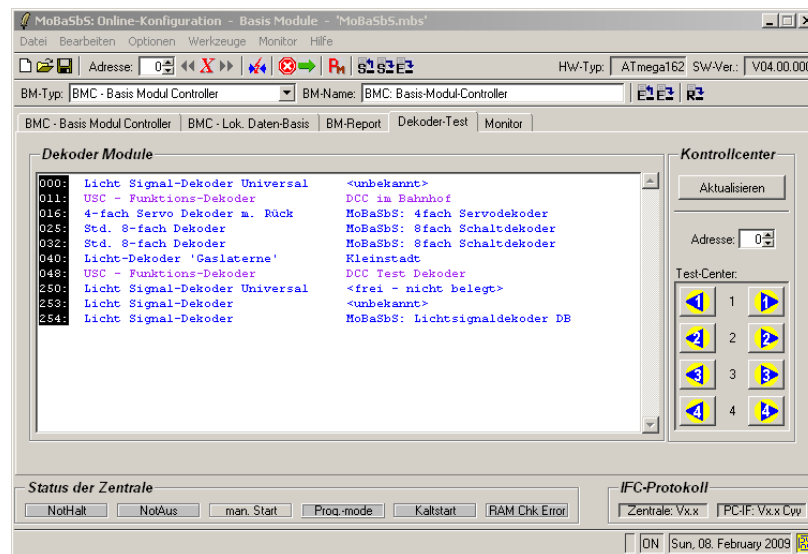
Hinweis: Wird die Page als rote Zahl auf schwarzem Grund angezeigt, handelt es sich um geschützte Bereiche (z. B. 'BootLader'), der nicht überschrieben werden darf.

Durch Drücken des Knopfes 'Öffnen' kann eine Datei zum Programmieren ausgewählt und geladen werden. Update-Dateien werden dabei im Verzeichnis 'patch' erwartet. Drücken auf den Knopf 'Löschen' entfernt alle geladenen Daten aus dem Speicher und setzt alles auf Start-Bedingungen zurück. Gleiches passiert ebenfalls, wenn die Quelle geändert wird. Bei Änderung der Adresse werden lediglich die Programmier-Parameter zurückgesetzt und es muss die HW unter der neuen Adresse wieder gelesen werden (Druck auf Knopf 'Aktualisieren').

Der eigentliche Programmier-Vorgang wird durch Drücken auf den Knopf 'Programmieren' ausgelöst. Der erfolgreiche Abschluss der Programmierung wird am Ende durch ein Informations-Fenster angezeigt.

Hinweis: Beim Update des EEP Bereiches eines Basis- oder Peripherie-Moduls wird die am Beginn des Update-Vorgangs eingestellte Modul-Adresse wiederhergestellt und ein SW-Reset ausgelöst, sodass die neu eingespielten EEPROM Werte gültig werden!

12.8 Decoder Schnellzugriff und Test



Decoder-Test

Mit diesem Register besteht die Möglichkeit, schnell auf Decoder-Ausgänge zugreifen zu können. Dabei kann sowohl auf MoBaSbS-System-Komponenten als auch auf USC-gesteuerte Decoder zugegriffen werden. Voraussetzung ist, dass alle Komponenten in der internen Datenbasis bekannt sind.

Durch Drücken auf den Knopf 'Aktualisieren' werden alle verfügbaren (und bekannten) Decoder im linken Textfenster aufgelistet. MoBaSbS-System-Komponenten werden dabei blau, USC gesteuerte in lila dargestellt. Klicken auf die Adresse im linken Fenster oder die direkte Auswahl der Adresse im Feld 'Adresse' wählt einen Decoder für den Schnellzugriff aus. Danach kann durch Drücken der Knöpfe "<<" bzw. ">>" eine Aktion am gewählten Decoder ausgelöst werden.

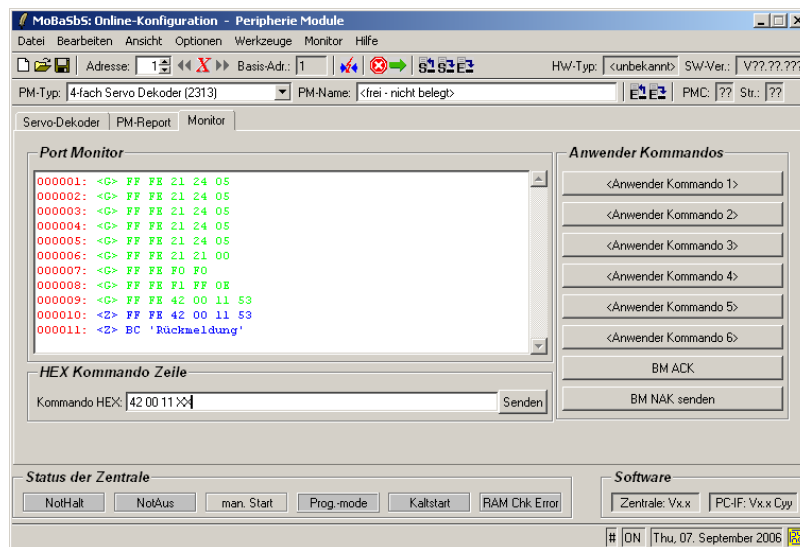
In diesem Fenster steht ausschließlich eine tastende Funktion zur Verfügung, d.h. beim Drücken eines Knopfes wird ein Einschalt-Befehl gesendet, beim Loslassen sofort der Ausschalt-Befehl.

12.9 Monitor

Das Register für den Monitor ist sowohl in der Basis- als auch in der Peripherie-Modul-Ansicht sichtbar und ermöglicht ein Überwachen des Datenstroms zwischen Konfigurationstool und MoBaSbS sowie ein manuelles Eingreifen zu Debug- und Testzwecken.

Port-Monitor

Der Port Monitor überwacht die gesamte Kommunikation zwischen Konfigurationstool und MoBaSbS-Hardware. Jeder Befehl/Befehlssequenz wird in einer eigenen Zeile dargestellt, wobei Befehle von der Software an die MoBaSbS-Hardware in grün und Antworten von der MoBaSbS-Hardware an die Software in blau dargestellt werden. Zusätzlich werden Antworten noch in einer weiteren Zeile kommentiert.



Port-Monitor

Über das Menü 'Monitor' kann der aktuelle Port-Monitor-Inhalt in eine ASCII-Text-Datei geschrieben werden (Menü 'Monitor > Monitor Log speichern') oder gelöscht werden (Menu 'Monitor > Monitor Log löschen'). Letzteres ist hilfreich, um uninteressante und/oder historische Kommunikationseinträge zu löschen, bevor interessante Kommunikationsabläufe protokolliert werden sollen.

Hinweis: Der Monitor ist stets aktiv und protokolliert alle Kommunikationsabläufe zwischen Software und MoBaSbS-Hardware, auch wenn das Register 'Monitor' nicht im Vordergrund zu sehen ist.

HEX Kommando Zeile

Über die Hex-Kommando Zeile kann der Anwender direkt Befehlssequenzen zur MoBaSbS-Hardware senden. Dazu muss der Befehl byteweise durch Leerzeichen getrennt im Hexadezimal-Format ohne führendes '0x' eingegeben werden. Wird am Ende ein 'XX' bzw. 'xx' angehängt, berechnet die Software vor dem Absenden der Befehlssequenz das XOR-Byte selbständig und ersetzt das 'XX'/'xx' durch dieses. Das Kommando wird sowohl durch Drücken der Taste 'Return' während der Programm-Fokus noch in der Eingabezeile steht, als auch durch Drücken auf den Knopf 'Senden' an die MoBaSbS-Hardware gesendet. Auch hier sind neben den üblichen Ziffern nur die Zeichen 'ABCDEFX' bzw. 'abcdefx' erlaubt.

Hinweis: Für den korrekten Aufbau der Befehlssequenz (Byte-Werte, Länge, etc.) ist ausschließlich der Anwender verantwortlich!

Anwender-Kommandos

Für häufig wiederkehrende Befehlssequenzen an die MoBaSbS-Hardware hat der Anwender die Möglichkeit sich bis zu 8 Befehlssequenzen zu definieren. Diese werden in der Datei 'MoBaSbS.ini' gespeichert und stehen beim nächsten Start des Konfigurationstools wieder zur Verfügung. Jedes dieser max. 8 Kommandos kann der Anwender über das Menu 'Monitor' frei mit Hilfe der nebenstehenden Maske konfigurieren.

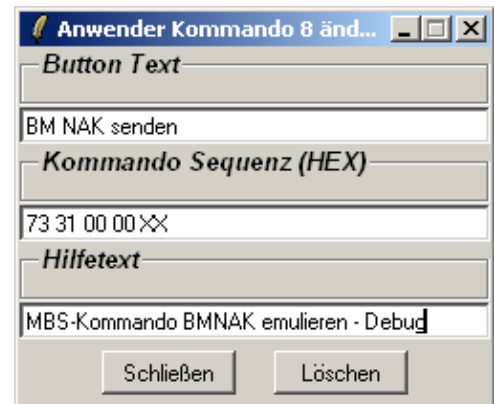
Kommando Sequenz (HEX)

Hexadezimale Kommandosequenz (für Details siehe Abschnitt 'Hex Kommando Zeile')

Hilfetext

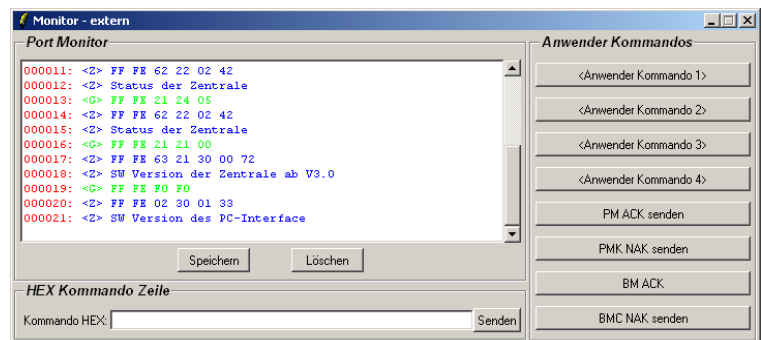
Hinweis, der beim Überfahren des Knopfes mit der Maus ausgegeben wird.

Ein Kommando wird durch Drücken des Knopfes 'Schließen' gespeichert und aktiviert, 'Löschen' setzt das Kommando auf Defaultwerte zurück.



Externer Hex-Monitor

Für Debug-Aufgaben kann auch ein externer Hex-Monitor in einem separaten Fenster aktiviert werden. Er ist im Prinzip eine 1:1 Kopie des internen und zeigt dieselben Informationen wie der interne Monitor. Funktionen wie Hex-Kommando-Zeile und Anwender-Kommandos verhalten sich ebenfalls identisch



Beispiel für die Ansteuerung der Funktionstasten

Es gibt folgende Lenz-Befehle für die Funktionstasten (zwischen den Klammern steht jeweils die Bedeutung des Bytes Daten als Bit, Fx=1 bedeutet, dass die Funktion eingeschaltet ist):
<AH> <AL> ist die Adresse der Lok mit AH = Adresse / 256 und AL = Adresse mod 256

Gruppe 1: Funktionstasten F0-F4 Data = [0 0 0 F0 F4 F3 F2 F1]
E4 20 <AH> <AL> <Data> XX

Gruppe 2: Funktionstasten F5-F8 Data = [0 0 0 0 F8 F7 F6 F5]
E4 21 <AH> <AL> <Data> XX

Gruppe 3: Funktionstasten F9-F12 Data = [0 0 0 0 F12 F11 F10 F9]
E4 22 <AH> <AL> <Data> XX

Gruppe 4: Funktionstasten F13-F20 Data = [F20 F19 F18 F17 F16 F15 F14 F13]
E4 23 <AH> <AL> <Data> XX

Gruppe 5: Funktionstasten F21-F28 Data = [F28 F27 F26 F25 F24 F23 F22 F21]
E4 28 <AH> <AL> <Data> XX

Um F0 und F3 für Adresse 32 zu setzen ist im HexMonitor folgendes einzugeben: E4 20 00 20 12 XX
Alle ausschalten geht dann mit E4 20 00 20 00 XX

Diese Befehle kann man auch auf die Benutzerkommandos legen, dann geht es schneller mit ein- ausschalten.

13 Anhang

13.1 Funktionen des HDC

Die ursprünglichen Funktionen des HDC wurden mehrmals ergänzt. Mit der Variante 3,3V (HDC 3,3V) des HDC wurde die Funktionalität des HDC erneut erweitert. Hier wird die Funktionalität des HDC 3,3V beschrieben. Funktionen, die in der Variante 5V nicht verfügbar sind, werden im Text separat gekennzeichnet mit dem Hinweis „nicht in HDC 3,3V verfügbar“. Zur Unterscheidung der beiden Varianten ist der verwendete uC maßgeblich. 162-er gleich Variante 5V, 644-er gleich HDC 3,3V. HDCs 3,3V und HDCs 5V lassen sich auch ohne weiteres gemischt an der MoBaSbS verwenden.

13.1.1 Kompatibilität

Der hier beschriebene HDC-Code beruht auf der ModellBahnSelbstbauSteuerung von Patrick Urban und ist auch nur mit selbiger verwendbar. Dabei muss der Hand-Daten-Controller in der HDC 3,3V oder HDC 5V, wie er zurzeit auf der Homepage von Patrick beschrieben wird, eingesetzt werden. In allen verwendeten Modulen muss die Softwareversion 4 aufgespielt sein.

Sollen alle Funktionen der HDC 3,3V verfügbar sein, muss im BMC, IFC und USC der Code aus dem Jahr 2018 eingespielt sein. Die Funktion POM steht nur für DCC-Decoder zur Verfügung und bisher nur mit dem USC Code von Uwe Mäusezahl. (Siehe Text-Datei im Verzeichnis USC Code)

Ab der HDC-Version vom 26.01.2009 ändert sich die Zählweise der Elemente an Decoder und Encoder. Sie beginnt bei den einzelnen Elementen nicht mehr bei ‚0‘, sondern bei ‚1‘. Bei den Encodern werden aus 0-0 bis 0-7 dann 0-1 bis 0-8. Aus den Decoder-Elementen 128-0 bis 128-3 werden dann die Elemente 128-1 bis 128-4. Damit ist die Zählweise identisch mit Traincontroller ab Version 7.0.

Weiterhin gibt es seit Oktober 2011 eine HDC Version 4.03, die eine geänderte Schnelligkeit des Drehreglers enthält.

Die Software ist im Archiv von Uwe Mäusezahl zu finden (siehe Kap. 5)

Die Geschwindigkeitsschwelle ist weiterhin über das Konfigtool änderbar. Allerdings ist das Verhalten dieses Wertes jetzt genau anders herum. Zum besseren Verständnis ein kleines Beispiel:

Ist der Wert 1, so wird wie folgt in- bzw. dekrementiert (Zeitwert beim Drehen des Drehgebers von einer Raststellung zur nächsten Raststellung in ms)

bis 6,4 , 12,8 , 19,2 , 25,6 , 32 , 38,4 ab 102,4ms
15 , 14 , 13 , 12 , 11 , 10 1

Bei einem Wert von 2 wird der Zeitwert halbiert, bei 3 gedrittelt usw. Möchte man diese Funktion nicht nutzen, so sollte ein hoher Wert gewählt werden.

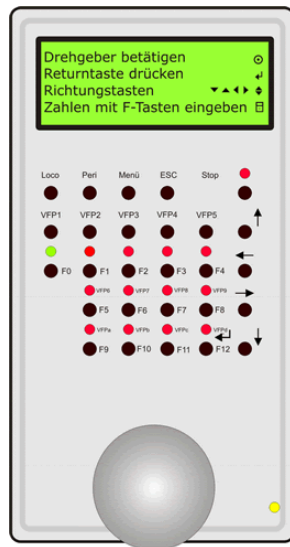
13.1.2 Einleitung

Die Lokdaten werden zentral im Basismodulcontroller (BMC) erfasst. Sie werden dort als Lokdatensätze abgelegt und können von jedem angeschlossenen HDC verwendet werden, ohne die Loknamen in den Programmspeicher des HDC einzuprogrammieren.

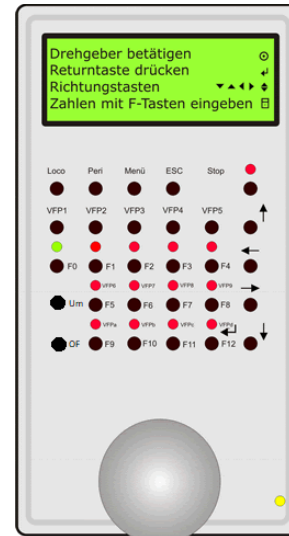
Für das Anlegen, Editieren und Löschen der Datensätze gibt es ein eigenes Menü im HDC. Im Hauptmenü wurden viele neue Menüpunkte aufgenommen, um Einstellarbeiten auch an Basismodulen durchzuführen.

Verbesserungen gibt es auch im Einstellmenü für die Servos. Es werden jetzt bei jedem Aufruf eines Servos alle bisherigen Einstellwerte von dem Modul geholt und man kann selbst entscheiden, ob mit diesen Werten gearbeitet oder alles zurückgestellt werden soll.

13.1.3 Navigation



5V Variante



3,3V Variante

HDC-Navigation

Zur Navigation werden die Tasten mit den Pfeilen verwendet. Diese werden am rechten Rand in der Zeile angezeigt, welche z. Zt. aktiv ist. Es sind mehrere Navigationssymbole möglich. Für die Zahleneingabe bei der Lokadresseneingabe und im POM-Menü zur CV Eingabe werden die Funktionstasten F0 bis F9 verwendet. In den sonstigen Menüs behalten sie ihre bisherige Funktion bei. Die Returntaste liegt auf der Funktionstaste F12. Wird das Symbol Returntaste angezeigt, so wird die Aktion ausgelöst, welche in dieser Zeile steht. Dabei kann es durchaus möglich sein, dass noch eine andere Zeile als aktiv angezeigt wird. Auch hier gilt wieder, dass die Funktionstaste F12 im Betriebsmenü ihre bisherige Funktion beibehält. Mit der ESC – Taste gelangt man immer wieder in die nächsthöhere Menüebene bis zum Startmenü.

13.1.4 Betriebsmodus

Durch Drücken einer der VFP-Tasten (VFP1 – VFP5) gelangt man in den Betriebsmodus. Hier können die Geschwindigkeit, die Funktionen F0 – F28 (HDC 5V nur Funktionen F0- F12) und die Richtung der jeweils ausgewählten Lok geändert werden. Ist für eine Lok das Datenformat MM ausgewählt, stehen nur die Funktionen F0 - F4 bereit.

Durch gleichzeitiges Drücken der VFP-Taste und dem Drehen am Drehgeber ist es möglich eine neue Lok auszuwählen. Dies wird nur von den oberen fünf virtuellen Fahrpulttasten VFP1 – VFP5 unterstützt.

Auf den Tasten F5 bis F8 und / oder F9 bis F12 lassen sich weitere VFP einrichten. Loks können hier jedoch nicht mit dem Drehgeber ausgewählt werden, sondern müssen im MoBaSbS Konfigurationstool belegt werden. Siehe Menü HDC Optionen. Dann stehen allerdings nur die Funktionen F0 – F4 bzw. F8 zur Verfügung.

Die Funktionen F13 – F20 und die Funktionen F21 – F28 werden durch Betätigen der Taste Um erreicht. Nach Aufrufen eines VFP sind zunächst immer die Funktionen F0 – F12 aktiv. Der jeweilige Funktionsblock wird im Display angezeigt.

Für jede Funktion kann separat im MoBaSbS Konfigurationstool eingestellt werden, ob die Funktion schaltend oder tastend erfolgen soll. Für die tastenden Funktionen eines HDC kann im Menü HDC Optionen ausgewählt werden, ob der „Aus“-Befehl nach ca. 0,3 Sekunden automatisch erfolgen soll oder ob der „Aus“ Befehl beim Loslassen der Taste erfolgen soll.

Dabei ist für die Taste F12 zu beachten, dass direkt nach dem Einschalten des HDC noch keine Lok aufgerufen werden kann. Dies ist erst nach Auswahl einer anderen Lok möglich. Der Grund liegt in der Verwendung als ENTER-Taste.

Betrieb
VFP b F8-
F12
E 103
Ad 1435 M 14 A ▶ 9

Betrieb
VFP 1 F13-
F20
BR 232
Ad 65 D 28 A ▶ 20

1. Zeile Hier wird angezeigt, dass man sich im Betriebsmodus befindet
2. Zeile Hier befinden sich die Information über grundsätzliche Einstellungen im HDC
Dies sind im Folgenden:
- es wurde das virtuelle Fahrpult (b) 1 ausgewählt
 - es sind die Funktionstasten (F8-F12) F13-F20 für die Verwendung als Funktion eingeschaltet
3. Zeile Hier wird der Lokname angezeigt, welcher für die Lokadresse vergeben wurde.
4. Zeile Hier werden alle Informationen über die gesteuerte Lok angezeigt.
- Adresse der Lok
 - D oder M für das Format, mit welchen die Lok gesteuert wird
 - die Fahrstufenanzahl, mit welcher die Lok betrieben wird (14, 28 oder 128)
 - ob die Lok von einem anderen HDC bzw. PC Steuerprogramm gesteuert wird
 - A – Lok wird vom HDC gesteuert
 - B – Lok wird von einem anderen Gerät gesteuert
 - P – Lok ist bisher noch nicht gesteuert worden
 - die Fahrtrichtung der Lokomotive
 - die jetzige Fahrstufe der Lok

Wird im Betriebsmodus die Cursortaste „hoch betätigt“, gelangt man in das POM-Menü. IM POM-Menü lassen sich CV-Werte eines Lokdecoders im laufenden Betrieb ändern. Ausgenommen hiervon ist die CV1 Lokadresse. Ein Umsetzen der Lok auf ein Programmiergleis ist für POM nicht erforderlich. Ebenso können beliebig viele Loks gerade in Betrieb sein. POM eignet sich insbesondere um Loks einzumessen oder die Geschwindigkeitskennlinie zu optimieren. Ebenso lässt sich im Betrieb die Lautstärke von Sounddecodern optimal abstimmen. Vor der Benutzung der Funktion POM sollte sich der Anwender der Bedeutung des Veränderns von CV im Klaren sein.

```

POM           Adr 254
Neue BR 212
CV 0002      Wert 128
Bitweise     10000000
  
```

In der 1. Zeile steht die Adresse des Lokdecoders.

In der 2. Zeile der definierte Lokname.

In der 3. Zeile kann mit den Tasten 0-9 der CV eingegeben werden. Der Wertebereich ist mit 0002 – 1024 definiert. Weder größere noch kleinere Werte lassen sich eingeben. Der CV-Wert wird mit dem Drehgeber eingegeben. Wird der Drehgeber schnell gedreht, ändern sich die Werte um den Wert 10. Der mögliche Wertebereich ist 0 – 255.

In der 4. Zeile wird der CV Wert Bitweise dargestellt. Sollen nur einzelne Bits einer CV geändert werden, spart dieses das notwendige umrechnen ins Dualsystem. Ob POM für einen Lokdecoder verfügbar ist oder nicht, ist der Betriebsanleitung desselben zu entnehmen. Alle neueren Lokdecoder verfügen werksseitig bereits über POM. POM steht nur für das DCC Protokoll zur Verfügung.

13.1.5 Hauptmenü („Menü“-Taste)

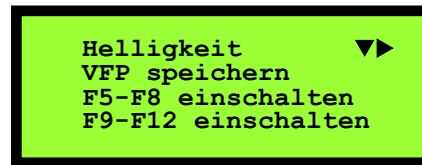
In das Hauptmenü gelangt man durch das Drücken der „Menü“-Taste.

```

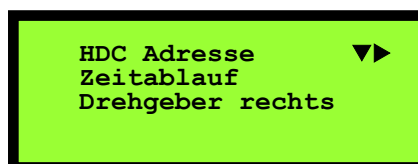
HDC Optionen  ▼▶
Basis Module
Peripherie Module
  
```

HDC-Optionen

In diesem Menüpunkt werden alle Einstellungen vorgenommen, die mit dem HDC in Verbindung stehen. Dieser Menüpunkt besteht aus zwei Seiten



- **Helligkeit**
In das Menü gelangt man durch die „Nach Rechts“-Taste. Hier wird durch Drehen am Drehgeber die Hintergrundbeleuchtung heller oder dunkler gestellt. Das Ganze kann dann durch Drücken der Return-Taste fest im HDC abgespeichert werden.
- **VFP speichern**
Die aktuell ausgewählten Lokadressen auf den virtuellen Fahrpulten 1-5 werden nach drücken der „Return“-Taste dauerhaft im EEPROM hinterlegt und stehen dann auch nach einem Neustart sofort wieder zur Verfügung. Die VFP 6-13 können nur über das MoBaSbS Konfigurationstool geändert werden.
- **Funktion F5-F8 an**
Um eine optimale Ausnutzung der Funktionstasten zu ermöglichen, ist es möglich, die Tasten F5-F8 mit den Funktionsaufrufen F5-F8 zu belegen oder weiterhin als VFP zu verwenden. Erscheint hinter F5-F8 ein „einschalten“, so lassen sich durch Drücken auf „Return“ die Tasten mit den Funktionen belegen.
Steht hinter F5 – F8 „ausschalten“, so wurden die Tasten bereits als Funktionstasten verwendet und man kann sie umschalten. Die Anzeige welche Tasten als Funktionen verwendet werden, steht im Betriebsmenü in Zeile 2. Man kann auch kurzzeitig die Funktionsbelegung wählen, die jeweilige Funktion schalten und anschließend das Ganze wieder als VFP verwenden. Somit wird eine optimale Auslastung der Funktionstasten erreicht.
- **Funktion F9-F12 an**
Wie schon im obigen Kapitel beschrieben lässt sich das Ganze auch mit den Funktionstasten F9 - F12 anwenden.
- **Nur 3,3V Variante.** Sind F5 – F8 und / oder F9 – F12 ausgeschaltet, lassen sich die Funktionstasten mit der „Um“ Taste nicht auf die Bereiche F13 – F20 und F21 - F28 umschalten. In diesen Fällen ist die „Um“ Taste ohne Funktion.



- **HDC Adresse**
Dieser Menüpunkt erscheint erst, nachdem in Zeile 4 des HDC Menüs, die Taste „Nach Unten“ gedrückt wurde. Hier muss dann durch die „Nach Rechts“-Taste noch in das eigentliche Menü gewechselt werden. Durch das Drehen am Drehgeber wird die neue HDC Adresse ausgewählt. Ein Druck auf die Return-Taste speichert die neue Adresse im Eeprom und löst einen Reset am HDC aus. Der Anwender muss sicherstellen, dass Adressen nicht doppelt vergeben werden. Wird das EEPROM im HDC neu programmiert, erhält jeder HDC zunächst die Adresse 31.
Wird der Drehgeber schnell gedreht, verändern sich die Werte pro Rastung nicht um 1 sondern um 10. So lassen sich auch große Wertveränderungen bequem realisieren. Die Geschwindigkeit für schnelles Drehen kann im MoBaSbS Konfigurationstool justiert werden.
- **Zeitablauf (nur 3,3V Variante)**
Mit dieser Funktion kann durch betätigen der Eingabetaste zwischen zwei Zuständen gewechselt werden. Steht Zeitablauf im Display, werden tastende Funktionen wie folgt bearbeitet. Durch Betätigen der Funktionstaste wird die Funktion eingeschaltet. Unabhängig davon wie lange der Anwender die

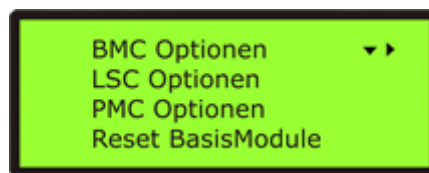
Taste tatsächlich noch betätigt, wird die Funktion nach ca. 0,3 Sekunden wieder ausgeschaltet. Eine so ausgelöste Soundfunktion wird vom Lok- / Sounddecoder vollständig abgespielt. Steht im Display Loslassen, wird die Funktion auch mit Betätigen der Taste eingeschaltet. Ausgeschaltet wird die Funktion jedoch erst, wenn der Anwender die Taste wieder loslässt. Diese Option lässt sich für jeden HDC unterschiedlich einstellen, gilt dann aber am ausgewählten HDC für alle tastenden Funktionen. Welche der jeweiligen Funktionen F0 – F28 schaltend oder tastend sein soll, muss im MoBaSbS Konfigurationstool eingestellt werden.

- Drehgeber rechts (nur 3,3V Variante)

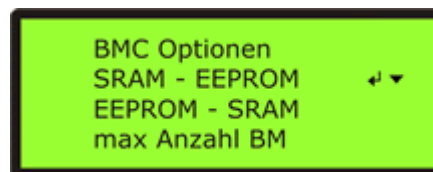
Auch dieses ist eine Hin- und Herschaltfunktion, zwischen denen ebenfalls mit der Eingabetaste gewechselt werden kann. Diese ist ausschließlich eine Komfortfunktion, mit der gewählt werden kann, ob der Drehgeber beim rechtsherum Drehen oder beim linksherum Drehen größere Werte liefert. Damit soll der Ergonomie der Rechts- sowie Linkshänder Rechnung getragen werden.

Basis-Module

Hier können Einstellungen am BMC, USC, PMC und ein Reset an allen BM durchgeführt werden



- BMC Optionen

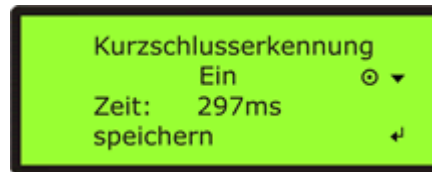


In diesem Menü ist es möglich, die Lokdatensätze vom SRAM des BMC in dessen Eeprom zu schreiben oder aus diesem zu lesen. Dabei ist besonders das Schreiben dieser Datensätze ins Eeprom wichtig, da sonst die gemachten Änderungen an den Lokdatensätzen beim nächsten Neustart verloren gehen. Dieser Menüpunkt ist auch in Zeile 5 des Locomenüs vorhanden, um dort schon die gemachten Änderungen dauerhaft zu speichern.

Das Lesen der Lokdatensätze aus dem Eeprom in das SRAM des BMC geschieht beim Neustart automatisch. Dieser Punkt wird nur benötigt, wenn Änderungen während des Betriebes rückgängig gemacht werden sollen.

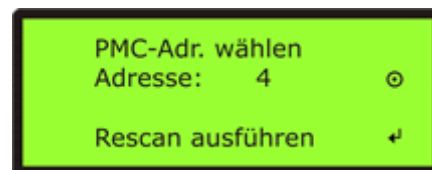
Im Menüpunkt „maximale Anzahl BM“ kann die Anzahl der zu pollenden Basismodule mit dem Drehgeber erhöht oder erniedrigt und anschließend gespeichert werden. Dabei sollte man darauf achten, dass auch nur so viele BM abgefragt werden, wie vorhanden sind, um die Geschwindigkeit des Systems nicht zu beeinflussen.

- USC Optionen



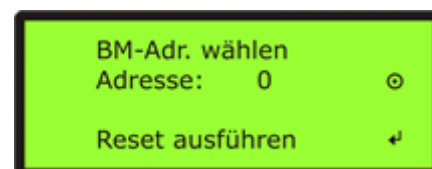
Im USC Menü kann in Zeile 2 mit dem Drehgeber ausgewählt werden, ob die Kurzschlusserkennung des Boosters vom USC ausgewertet werden soll oder nicht. Wird Kurzschlusserkennung „Ein“ gewählt, ist in Zeile 3 eine Verzögerungszeit zu wählen, bis der Kurzschluss vom USC ausgewertet wird. Ein Druck auf die Return-Taste speichert die Einstellungen.

- PMC Optionen



Nach dem Hinzufügen neuer Peripheriemodule wie z.B. Servodecoder oder nach einer Adressänderung, muss dem PMC dies mitgeteilt werden. Dies geschieht am einfachsten über einen Rescan des PMC. Dabei werden alle Stränge des PMC abgefragt und die gefundenen Module im PMC hinterlegt. Somit weiß der PMC wieder, ob er angeschlossene Encoder abfragen oder Schaltbefehle an angeschlossene Module weiterleiten muss. Die Adresse 4 ist als Standardadresse voreingestellt. Sind mehrere PMCs in der MoBaSbS eingebaut, kann die Adresse durch den Drehgeber geändert werden.

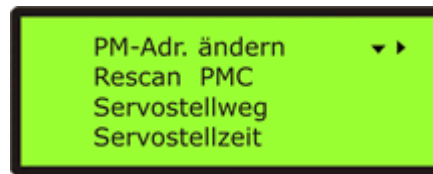
- Reset Basismodule



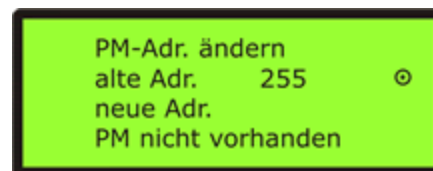
Mit dieser Option ist es möglich einen Reset an einzelnen Basismodulen auszulösen, ohne dass ein Neustart der MoBaSbS notwendig ist. Dazu wird mit dem Drehgeber die BM-Adresse gewählt und anschließend durch die „Return“-Taste der Reset ausgelöst

Peripherie-Module

Das Menü ist für die Einstellung eines Servomotors vorgesehen.

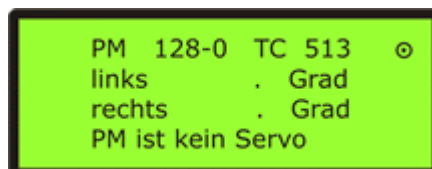


- PM-Adr. Ändern



Zuerst wird die alte Adresse mit dem Drehgeber ausgewählt. Ist dieses Modul nicht vorhanden, so erscheint in Zeile 4 „PM nicht vorhanden“. Wurde ein Modul gefunden, kommt man mit der „Nach Unten“-Taste in Zeile 3, in der sich mit dem Drehgeber die neue Adresse auswählen lässt. Gespeichert wird die neue Adresse mit der „Return“-Taste. Anschließend erscheint in Zeile 4 „Rescan durchführen“. Dies ist notwendig damit der Peripheriemodulcontroller (PMC) das Modul unter der neuen Adresse ansprechen kann.

- Rescan PMC
Nach Anwahl dieses Menüpunktes kann durch den Drehgeber der jeweilige PMC ausgewählt werden, an dessen Strängen das geänderte Peripheriemodul angeschlossen ist. Es ist dann der Rescan auszulösen. Einzelheiten dazu im Menüpunkt „PMC Optionen“.
- Servostellweg



Darstellung bis HDC-Version 081019



Darstellung ab HDC-Version 090126

Nach Anwahl dieses Menüpunktes, kann man durch den Drehgeber das Servomodul auswählen.

Dabei wird, um Fehlerquellen bei der Umrechnung in das TC-Format zu vermeiden, die TC-Adresse mit angezeigt. Die Zahl nach dem Bindestrich zeigt den Ausgang an, wie er auf den Leiterplatten angegeben ist. Ist das ausgewählte Modul kein Servodecoder, so erscheint in Zeile 4 „PM ist kein Servo“.

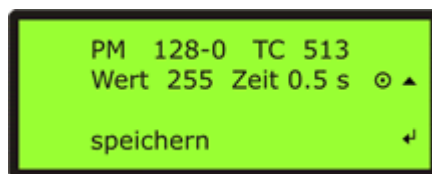
Ist an dieser Adresse ein Servodecoder angeschlossen, wechselt die Anzeige in Zeile 4 wechselt nach „Mittelstellung Ta 0“. Mit Druck auf die Taste F0 beginnt die Justage der Mittelstellung des Servos.

Möchte man nur den Servo etwas nachjustieren, so wechselt man direkt in Zeile 2, in der die gespeicherten Werte des Servodecoder-Eeproms angezeigt werden. Durch den Drehgeber wird der Servo in beide Richtungen eingestellt.

Will man die Servostellzeit noch ändern, so kann der Speichervorgang auch erst in diesem Menü vorgenommen werden.

Zum Testen ist es möglich die „Peri“-Taste zu drücken. Dort kann dann der Servodecoder wie üblich geschaltet werden. Durch die ESC Taste gelangt man wieder in das Servoeinstellmenü zurück.

- Servostellzeit



Darstellung bis HDC-Version 081019



Darstellung ab HDC-Version 090126

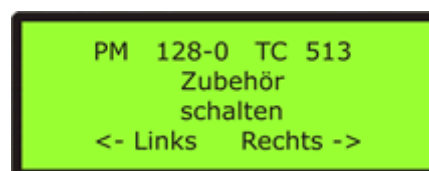
Im Menüpunkt Servostellzeit wird die Zeit eingegeben, die der Servo für die eingestellten Justagewerte benötigt.

Zuerst wird in der oberen Zeile durch Drehen des Drehgebers die Adresse ausgewählt. Bei jeder Anwahl einer neuen Adresse, werden alle erforderlichen Daten vom Eeprom des Servodecoders geholt.

In der zweiten Zeile wird durch Drehen am Drehgeber der Wert von 1-255 verändert. Dabei erscheint gleichzeitig ein errechneter Zeitwert. Dieser Wert ist in ungefähr die Zeit, die der Servo vom gewählten Linksanschlag bis zum gewählten Rechtsanschlag benötigt. Nach dem Einstellen der Zeit kann man diese über die „Peri“-Taste testen. Mit der ESC-Taste gelangt man ins Servostellzeitmenü zurück. Anschließend das Speichern nicht vergessen, sonst sind alle Einstellarbeiten nach einem Neustart wieder hinfällig.

13.1.6 Peripherie-Menü („Peri“-Taste)

Durch Drücken der „Peri“-Taste gelangt man in dieses Menü.



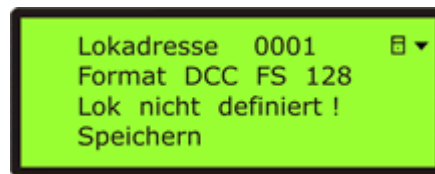
In diesem Menü ist es möglich Peripheriedecoder anzusteuern oder Encoder abzufragen.

Nach dem ersten Druck erscheint die Ansteuerung für die Peripheriemodule („Zubehör“). Mit dem Drehgeber wird die gewünschte Adresse ausgewählt und durch Drücken der <- oder -> Tasten das jeweilige Modul geschaltet.

Mit einem weiteren Druck auf die „Peri“-Taste gelangt man in das Encodermenü. Hier wird für jeden Eingang der nach Masse geschaltet wurde, ein Punkt über seiner Adresse angezeigt.

13.1.7 Lokmenü („Loco“-Taste)

In das Menü gelangt man durch das Drücken der „Loco“-Taste. Es ist möglich Lokdatensätze anzulegen, zu editieren und zu löschen.



Lokadresseingabe

In Zeile 1 dieses Menüs muss die Lokadresse zwingend vierstellig eingegeben werden. Dazu werden die Funktionstasten F0 – F9 verwendet. Nach jeder Eingabe wird ein Zeichen weiter nach rechts gesprungen. Zur Korrektur kann mit dem Navigationstasten „Links“ oder „Rechts“ zu jeder Ziffer gewechselt werden.

Nachdem die Lokadresse fertig eingegeben wurde, wird durch Drücken der „Nach Unten“ Taste in Zeile 2 gewechselt. Dabei holt sich der HDC alle Informationen über diese Lok aus der Lokdatenbank im BMC. Ist dieser Datensatz noch nicht vorhanden, steht in Zeile 3 „Lok nicht definiert!“. Im folgendem kann man nun diesen Datensatz neu anlegen, editieren oder löschen.

Lokformat

In Zeile 2 wird mit Hilfe des Drehgebers eines der vier voreingestellten Formate ausgewählt. Dies sind:

- MM 14 FS
- DCC 14 FS
- DCC 28 FS
- DCC 128 FS

Mit der „Nach Unten“ Taste geht es nun zur Loknameneingabe in Zeile 3

Loknameneingabe

War dieser Datensatz noch nicht vorhanden und stand bisher hier „Lokdatensatz nicht definiert!“, so wird diese Zeile gelöscht und man kann den Loknamen eingeben. Wenn dieser Datensatz schon angelegt war, so kann er hier editiert werden. Mit Hilfe des Drehgebers ist es möglich, aus einem voreingestellten Zeichensatz ein Zeichen oder eine Zahl auszuwählen. Durch die „Nach Rechts“ und die „Nach Links“ Taste kann man sich in der Zeile bewegen und jedes Zeichen ändern. Ist man mit seiner Eingabe zufrieden geht es mit der „Nach Unten“ Taste ins Menü Speichern.

Menü Speichern

Hier hat man nun die Möglichkeit durch Betätigen der Returnntaste seine Daten zum BMC zu senden, wo sie in einer Lokdatenbank abgelegt werden. Möchte man einen vorhanden Datensatz löschen so erscheint rechts unten das Symbol für die „Nach Rechts“ Taste. Durch Auswahl dieses Menüpunktes ändert sich die Zeile 4 in Datensatz löschen und durch die Returnntaste kann dieser Datensatz nun gelöscht werden.

ACHTUNG !

Die versendeten Lokdatensätze sind noch nicht in das Eeprom des BMC übertragen worden, das heißt, durch Neustart der MoBaSbS gehen diese Änderungen an den Datensätzen wieder verloren. Um dies zu verhindern, wechselt man durch erneutes Drücken der „Nach Unten“ Taste in Zeile 5 des Locomenüs. Dort wird durch Betätigen der Returnntaste der Transfer der Lokdatensätze vom SRAM des BMC in das Eeprom des BMC ausgelöst.

13.2 Programmierung und Inbetriebnahme des USB-Adapters

13.2.1 USB Tutorial (von Holger Bernhard)

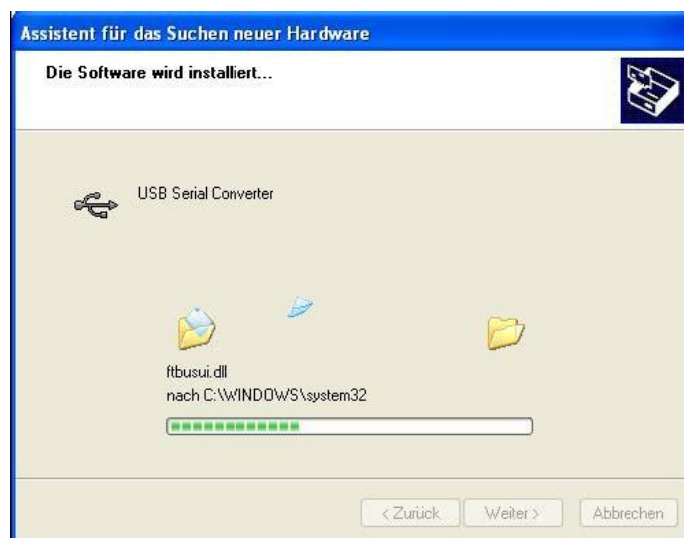
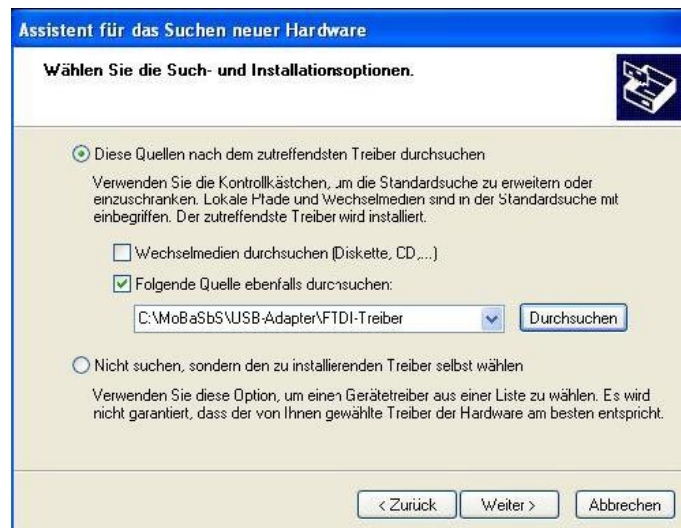
Von Holger Bernhard wurde die folgende Beschreibung erstellt (Stand 27.09.2007). Die im Text erwähnten Dateien und Tools können auf Nachfrage zur Verfügung gestellt werden. Sie sind alle in einem ZIP-File enthalten.

Im ZIP-File sind folgende Dateien enthalten:

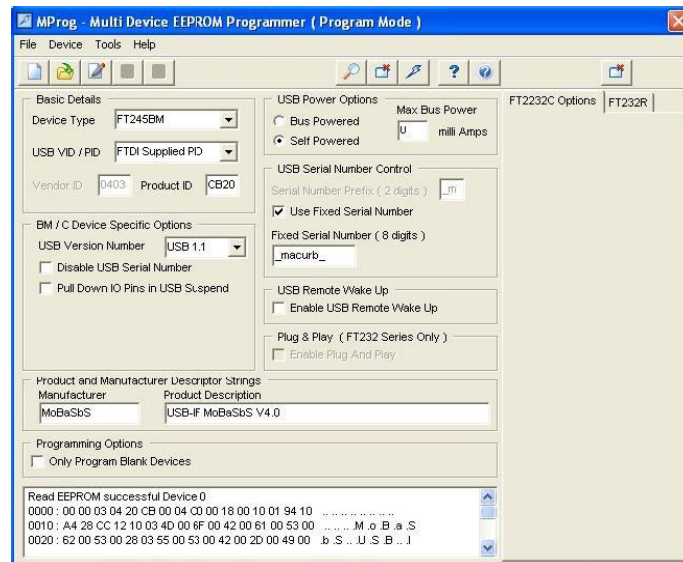
- Tutorial.pdf / dieses Dokument
- Mprog3.0_Setup.exe / Programm zum beschreiben des USB Chips
- ftdi.ept / Einstellungen für Mprog
- FTDI-Treiber / aktuelle, an die MoBaSbS V4 angepasste, USB-Treiber

1. alle Dateien in einen Ordner nach Wahl entpacken (z. B. C:\MoBaSbS\USB-Adapter)
2. Mprog installieren: Mprog3.0_Setup.exe ausführen und den Anweisungen am Bildschirm folgen
3. USB-Modul auf das IFC aufstecken und DIP-Switch 1 am IFC auf ON (USB-Betrieb) stellen
4. MoBaSbS mit dem PC verbinden und MoBaSbS einschalten
5. neue Hardware wird gefunden (USB <-> Serial), und es wird nach den Treibern gefragt





- es wird erneut neue Hardware gefunden (USB Serial Port). Hier erfolgt die gleiche Vorgehensweise wie unter Punkt 5 beschrieben.



13. die Programmierung ist nun abgeschlossen und die MoBaSbS kann ihren USB-Betrieb antreten.

13.2.2 Installation FTDI (von Jörg Schindler, Stand 12.02.2017)

Nachfolgend soll eine Anleitung zum Einrichten der FTDI USB Treiber unter Win 8.x/10 sowohl für 32-bit als auch 64-bit Installationen gegeben werden. Da jede automatische Installation irgendwie immer wieder Probleme bereitet, wird hier in Anlehnung an die FTDI Applikations-Notiz

http://www.ftdichip.com/Support/Documents/InstallGuides/AN_396%20FTDI%20Drivers%20Installation%20Guide%20for%20Windows%2010.pdf

die manuelle Installation gewählt.

Es wäre noch zu erwähnen, dass ich immer alle automatisch angebotenen Updates installieren lasse, mich also praktisch nicht speziell um Updates kümmere. Sie sind da und ich habe bisher auch keine schlechten Erfahrungen damit gemacht, auch in Bezug auf die FTDI Treiber und die Ankopplung der MoBaSbs. Aktuell ist folgende Version Windows Version (hier der 64-bit Büro Rechner) installiert:

Edition	Windows 10 Pro
Version	1607
Betriebssystembuild	14393.693

Treiber laden

Als Voraussetzung für die manuelle Installation der FTDI-Treiber müssen die Treiber von der FTDI-Seite geladen werden.

<http://www.ftdichip.com/Drivers/.../CDM v2.12.24 WHQL Certified.zip>

Operating System	Release Date	Processor Architecture					Comments
		x86 (32-bit)	x64 (64-bit)	ARM	MIPS	SH4	
Windows*	2016-10-10	2.12.24	2.12.24	-	-	-	WHQL Certified. Includes VCP and D2XX. Available as a setup executable Please read the Release Notes and Installation Guides.

Nach allem was ich recherchieren konnte, bekommt man bei einem Windows Treiber, egal ob man über VCP oder D2xx reingeht, immer dasselbe Softwarepaket geliefert.

Ich versuche immer aktuelle Treiber zu benutzen, der oben angegebene Dateiname ist nur als Beispiel zu sehen (Version vom 16.1.2017). Das Paket beinhaltet sowohl die 32-bit als auch die 64-bit Treiber.

Bei früheren Versionen des FTDI Treibers bekam man eine exe-Datei zum Download angeboten. In dem Fall bestand der erste Trick darin, die Datei umzubenennen. Der Name konnte erhalten bleiben, die Erweiterung muss aber in „.zip“ umbenannt werden. Denn um nichts anderes als um ein selbstentpackendes Zip-Archiv handelte es sich auch da schon bei der von FTDI bereitgestellten Datei. In einem zweiten Schritt entpackte man nun den Inhalt der Zip-Datei in ein beliebiges Verzeichnis, nachfolgend Ablageort genannt.

Inf-Dateien anpassen

Gehen wir davon aus, dass wir die Zip-Datei (oder die frisierte exe-Datei) entpackt haben und den Ablageort kennen.

Bevor an das Installieren der Treiber gegangen werden kann, erfolgt zunächst die Bekanntmachung der MoBaSbS-PID durch Anpassen der Inf-Dateien (siehe Kap. 7.3). Im Detail scheint FTDI den String „FtdiPort232.NT“ in den entspr. Sektionen der „Ftdioprt.inf“ durch den String „FtdiPort.NT“ ersetzt zu haben.

Im Klartext – die „roten“ Zeilen sind in den Dateien „Ftdibus.inf“ bzw. „Ftdioprt.inf“ zu ergänzen

Ftdibus.inf (aus Treiberpaket, Version CDM v2.12.24 WHQL Certified)

```
[FtdiHw]
%USB\VID_0403&PID_6001.DeviceDesc%=FtdiBus.NT,USB\VID_0403&PID_6001
%USB\VID_0403&PID_6010&MI_00.DeviceDesc%=FtdiBus.NT,USB\VID_0403&PID_6010&MI_00
%USB\VID_0403&PID_6010&MI_01.DeviceDesc%=FtdiBus.NT,USB\VID_0403&PID_6010&MI_01
%USB\VID_0403&PID_6011&MI_00.DeviceDesc%=FtdiBus.NT,USB\VID_0403&PID_6011&MI_00
.....
%USB\VID_0403&PID_603E.DeviceDesc%=FtdiBus.NT,USB\VID_0403&PID_603E
%USB\VID_0403&PID_0000.DeviceDesc%=FtdiBus.NT,USB\VID_0403&PID_0000
%USB\VID_0403&PID_CB20.DeviceDesc%=FtdiBus.NT,USB\VID_0403&PID_CB20

[FtdiHw.NTamd64]
%USB\VID_0403&PID_6001.DeviceDesc%=FtdiBus.NTamd64,USB\VID_0403&PID_6001
%USB\VID_0403&PID_6010&MI_00.DeviceDesc%=FtdiBus.NTamd64,USB\VID_0403&PID_6010&MI_00
%USB\VID_0403&PID_6010&MI_01.DeviceDesc%=FtdiBus.NTamd64,USB\VID_0403&PID_6010&MI_01
%USB\VID_0403&PID_6011&MI_00.DeviceDesc%=FtdiBus.NTamd64,USB\VID_0403&PID_6011&MI_00
.....
%USB\VID_0403&PID_603E.DeviceDesc%=FtdiBus.NTamd64,USB\VID_0403&PID_603E
%USB\VID_0403&PID_0000.DeviceDesc%=FtdiBus.NTamd64,USB\VID_0403&PID_0000
%USB\VID_0403&PID_CB20.DeviceDesc%=FtdiBus.NTamd64,USB\VID_0403&PID_CB20

.....
[Strings]
Ftdi="FTDI"
DESC="CDM Driver Package - Bus/D2XX Driver"
DriversDisk="FTDI USB Drivers Disk"
USB\VID_0403&PID_6001.DeviceDesc="USB Serial Converter"
USB\VID_0403&PID_6010&MI_00.DeviceDesc="USB Serial Converter A"
USB\VID_0403&PID_6010&MI_01.DeviceDesc="USB Serial Converter B"
USB\VID_0403&PID_6011&MI_00.DeviceDesc="USB Serial Converter A"
.....
USB\VID_0403&PID_603E.DeviceDesc="UMFTPD3A Programming Board"
USB\VID_0403&PID_0000.DeviceDesc="USB Serial Converter"
USB\VID_0403&PID_CB20.DeviceDesc="USB-IF MoBaSbS V4.0"
SvcDesc="USB Serial Converter Driver"
ClassName="USB"
```

Ftdiport.inf (aus Treiberpaket, Version CDM v2.12.24 WHQL Certified)

```
[FtdiHw]
%VID_0403&PID_6001.DeviceDesc%=FtdiPort.NT,FTDIBUS\COMPORT&VID_0403&PID_6001
%VID_0403&PID_6010.DeviceDesc%=FtdiPort.NT,FTDIBUS\COMPORT&VID_0403&PID_6010
%VID_0403&PID_6011.DeviceDesc%=FtdiPort.NT,FTDIBUS\COMPORT&VID_0403&PID_6011
.....
%VID_0403&PID_603A.DeviceDesc%=FtdiPort.NT,FTDIBUS\COMPORT&VID_0403&PID_603A
%VID_0403&PID_0000.DeviceDesc%=FtdiPort.NT,FTDIBUS\COMPORT&VID_0403&PID_0000
%VID_0403&PID_CB20.DeviceDesc%=FtdiPort.NT,FTDIBUS\COMPORT&VID_0403&PID_CB20

[FtdiHw.NTamd64]
%VID_0403&PID_6001.DeviceDesc%=FtdiPort.NTamd64,FTDIBUS\COMPORT&VID_0403&PID_6001
%VID_0403&PID_6010.DeviceDesc%=FtdiPort.NTamd64,FTDIBUS\COMPORT&VID_0403&PID_6010
%VID_0403&PID_6011.DeviceDesc%=FtdiPort.NTamd64,FTDIBUS\COMPORT&VID_0403&PID_6011
.....
%VID_0403&PID_603A.DeviceDesc%=FtdiPort.NTamd64,FTDIBUS\COMPORT&VID_0403&PID_603A
%VID_0403&PID_0000.DeviceDesc%=FtdiPort.NTamd64,FTDIBUS\COMPORT&VID_0403&PID_0000
%VID_0403&PID_CB20.DeviceDesc%=FtdiPort.NTamd64,FTDIBUS\COMPORT&VID_0403&PID_CB20

[Strings]
FTDI="FTDI"
DESC="CDM Driver Package - VCP Driver"
DriversDisk="FTDI USB Drivers Disk"
PortsClassName = "Ports (COM & LPT)"
VID_0403&PID_6001.DeviceDesc="USB Serial Port"
VID_0403&PID_6010.DeviceDesc="USB Serial Port"
VID_0403&PID_6011.DeviceDesc="USB Serial Port"
.....
VID_0403&PID_0000.DeviceDesc="USB Serial Port"
VID_0403&PID_CB20.DeviceDesc="USB-IF MoBaSbS V4.0"
SvcDesc="USB Serial Port Driver"
SerEnum.SvcDesc="Serenum Filter Driver"
```

Ich selbst nutze bei Updates etc. genau diese PDF-Datei, indem ich die „roten“ Zeilen per Copy&Paste an die entsprechenden Stellen der „inf“-Dateien einer neuen FTDI-Treiber-Version kopiere. Es sollte ausreichen, wenn die „roten“ Zeilen in der jeweiligen Sektion, z.B. [Strings] an eine beliebige Stelle reinkopiert werden.

Achtung: die „inf“-Dateien enthalten weitere durch Wörter in eckigen Klammern eröffnete Sektionen. Die sind für uns nicht relevant und tauchen deshalb oben auch nicht auf. Wir müssen uns jeweils nur um [FtdiHw], [FtdiHw.NTamd64] und [Strings] kümmern.

Treiber installieren

Jetzt ist der Zeitpunkt gekommen, den IFC per USB-Kabel mit dem PC zu verbinden und sich mit dem Programm „USBView“ kurz zu vergewissern, dass der IFC mit der MoBaSbS eigenen PID gefüttert ist. Muss man nicht machen, dient aber der Sicherheit! Es ist für eine spätere Fehlersuche immer gut zu wissen, bis zu welchem Schritt man noch fehlerfrei dabei war.

In meinem Fall habe ich die USB-Platine fertig programmiert von Patrick bezogen. Wie man die Programmierung selber vornimmt, ist im „USB-Tutorial“ von Holger Bernard ausführlich beschrieben (siehe Kap. 13.2.1).

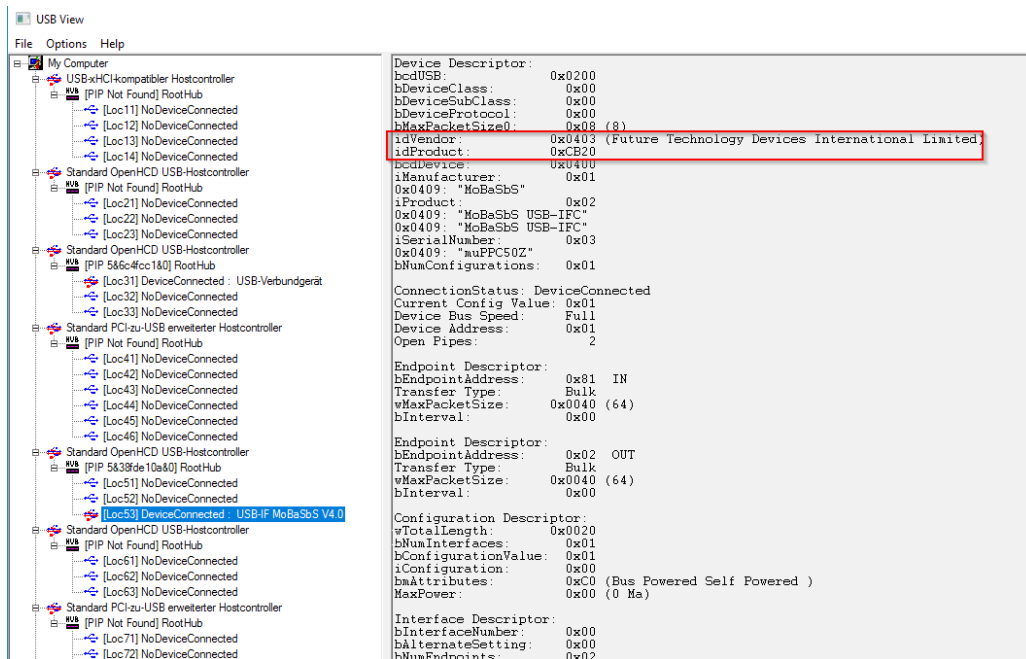


Bild 1 Ausgabe von USB View unter Win10

Passt alles, geht man in den Geräte-Manager und findet dort unter „Andere Geräte“ einen Eintrag „USB <-> Serial Converter“ oder auch „MoBaSbS USB-IF“ mit dem bekannten gelben Ausrufezeichen versehen, was auf eine noch nicht abgeschlossene oder falsche Konfiguration hinweist (linker Teil von Bild 4). Klar, wir haben ja noch gar keinen Treiber installiert. Läuft das „MoBaSbS USB-IF“ aber bereits, wird es unter „Anschlüsse (Com & LPT)“ eingetragen (Bild 2). Das wäre bei einem Treiberupdate-Szenario der Fall.

Den Geräte-Manager erreicht man unter Win10/8.1. mit einem Rechts-Klick auf das Windows Symbol in der linken unteren Ecke (Desktop-Ansicht vorausgesetzt), unter Win 7 z.B. über „Computer“ à „Verwalten“.

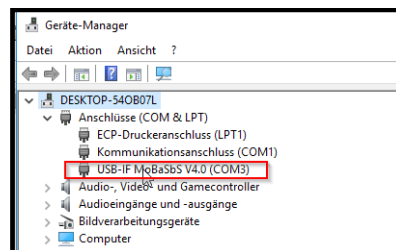


Bild 2 Ausgabe von USB View unter Win10

Aber hoppla, jetzt wird's brenzlig

Die Vorgehensweise bei der Treiberinstallation unterscheidet sich gravierend je nachdem, ob wir uns in einer 32-bit oder 64-bit Welt bewegen.

Wer es nicht genau weiß, eine von mehreren Möglichkeiten es herauszubekommen geht über die Systemsteuerung

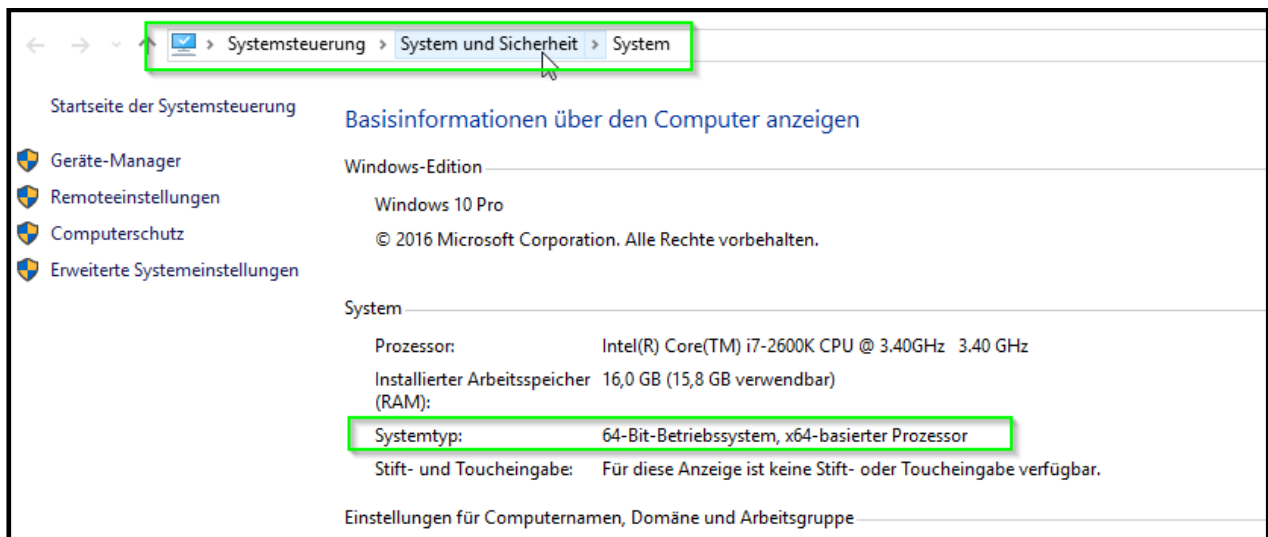


Bild 3 Systeminfo

32-bit Welt ...

.. würde ich für unseren Anwendungsfall immer bevorzugen, da es sich einfach viel besser handhaben lässt. Warum? Microsoft sagt sich wohl, 32-bit ist für zu Hause, da wollen wir mal nicht so streng sein. 64-Bit ist was für's Business, da muss alles mega-stabil laufen und da akzeptieren wir nur Treiber, die wir gesehen und zertifiziert haben (das ist meine ganz persönliche Formulierung 😊). Wie's dennoch – Stand 16.01.2017 – über eine offiziell von Microsoft dokumentierte Hintertür funktioniert steht, weiter unten.

Mit einem Rechts-Klick auf den Eintrag „MoBaSbS USB-IF“ öffnet sich ein Dialog, aus dem wir den Menüpunkt „Treibersoftware aktualisieren“ auswählen. Der Begriff könnte verwirren, da im Falle einer Neuinstallation noch gar kein Treiber installiert wurde. Im sich nun öffnendem Dialog wählen wir „Auf dem Computer nach Treibersoftware suchen“ aus und gelangen zu einem Menü, wo wir das Verzeichnis angeben müssen, in welchem wir unsere Treiber abgespeichert haben (der oben erwähnte Ablageort). Abschließend noch den Button „Weiter“ drücken und los geht's.

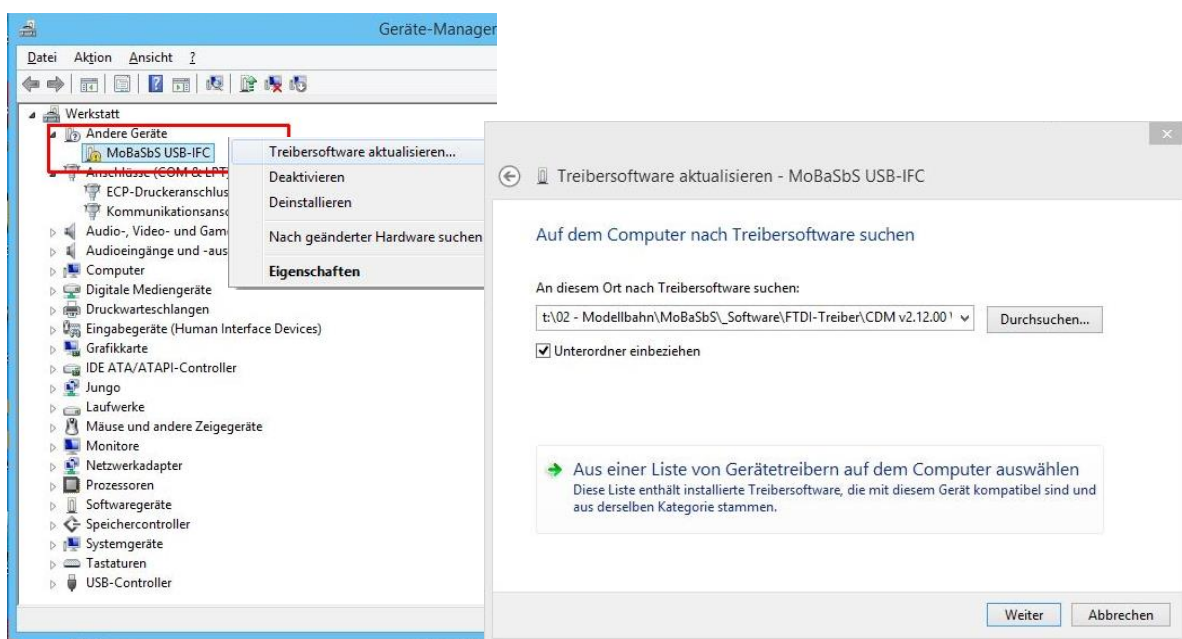


Bild 4 Dialoge im Geräte-Manager Win8.1 am Beispiel einer kompletten Neuinstallation

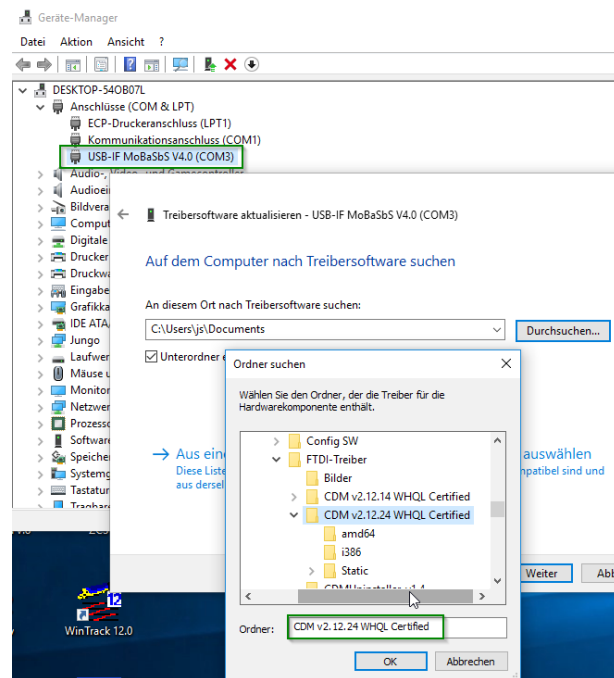


Bild 5 ... und im Geräte-Manager Win10 am Beispiel eines bereits laufenden Treibers (Update)

.... aber noch nicht ganz. Microsoft nimmt es mit der Überprüfung der Treiber hinsichtlich Zertifizierung eben auch schon bei 32-bit sehr genau. Normalerweise handelt es sich ja bei dem FTDI Treiber um einen von Microsoft zertifizierten Treiber, das geht aus dem „WHQL Certified“ im Dateinamen des Treibers hervor. Da wir aber die „inf“-Dateien manipuliert haben um dem Treiber unsere MoBaSbS PID beizubringen, ist das Zertifikat ungültig geworden. Im Falle einer 32-Bit Win8/10 Installation reicht es, die folgende Abfrage mit „Treibersoftware trotzdem installieren“ zu beantworten.

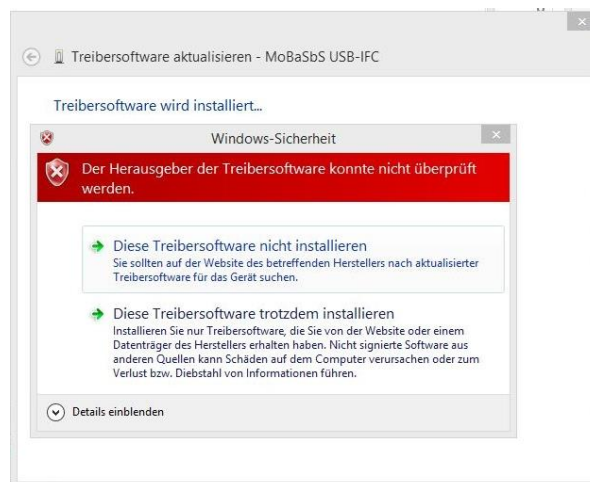


Bild 6 Sicherheitsabfrage bei fehlendem Zertifikat

Nun ist der erste wichtige Schritt geschafft und je nach Rechnerleistung bekommt man früher oder später folgende Bestätigung über den Erfolg der Treiberinstallation zu Gesicht.

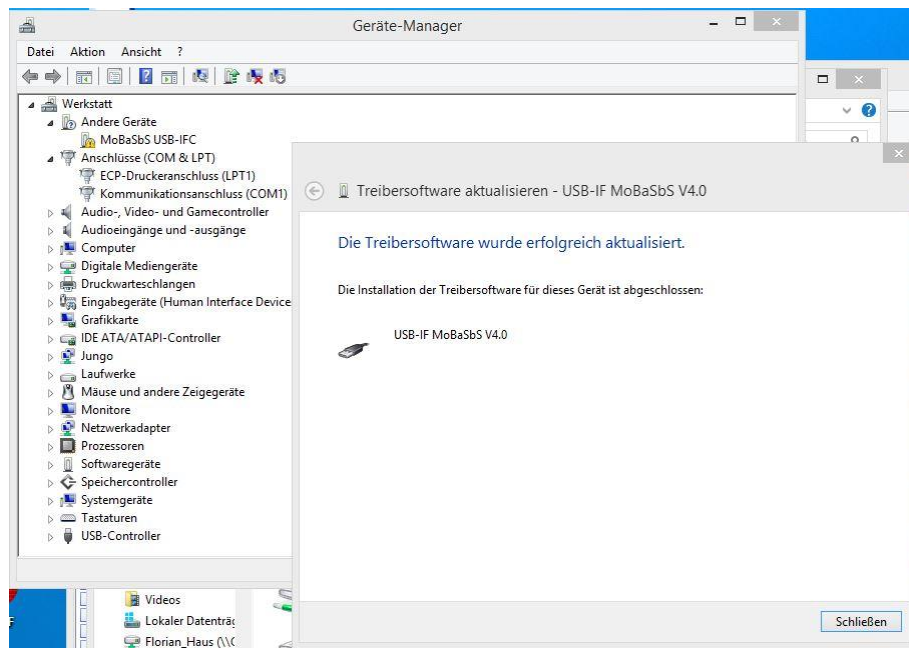


Bild 7 USB-Treiber erfolgreich installiert

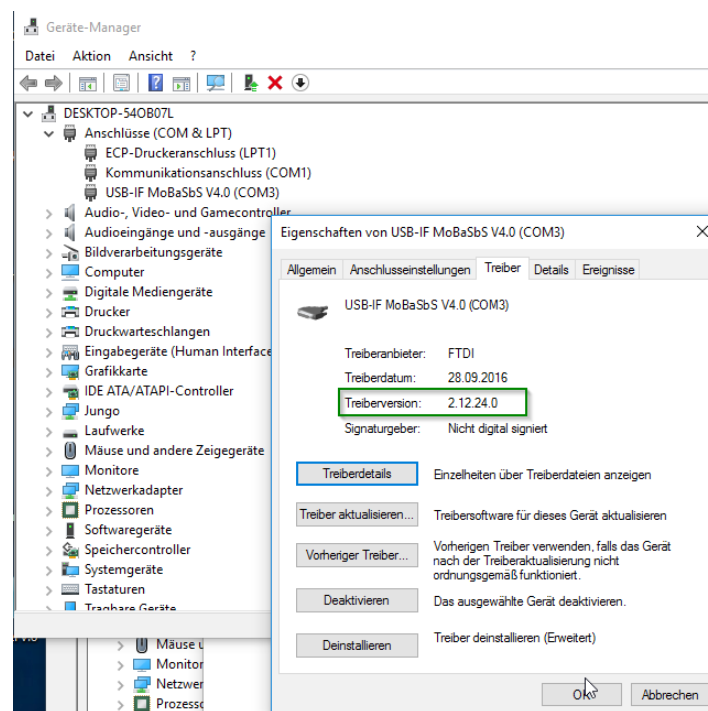


Bild 8 Unter "USB-IF MoBaSbS V4.0"-Eigenschaften kann man sehen, dass der Treiber erfolgreich installiert wurde.

Im zweiten Schritt muss nun noch der virtuelle COM-Port eingerichtet werden. Dazu rufen wir erneut den Geräte-Manager auf und finden unter „Andere Geräte“ einen Eintrag „USB Serial Port“ mit einem gelben Warnzeichen (Bild 9), welches – nun keine Überraschung mehr – wieder auf ein Problem hinweist. Denn es fehlt auch diesmal der Treiber.

Erneut ist wie oben beschrieben vorzugehen: Rechts-Klick auf den Eintrag „USB Serial-Port“, „Treibersoftware aktualisieren“ und im darauffolgenden Dialog „Auf dem Computer nach Treibersoftware suchen“ auswählen sowie das Verzeichnis angeben, in welchem wir unsere Treiber abgespeichert haben und den Button „Weiter“ drücken.

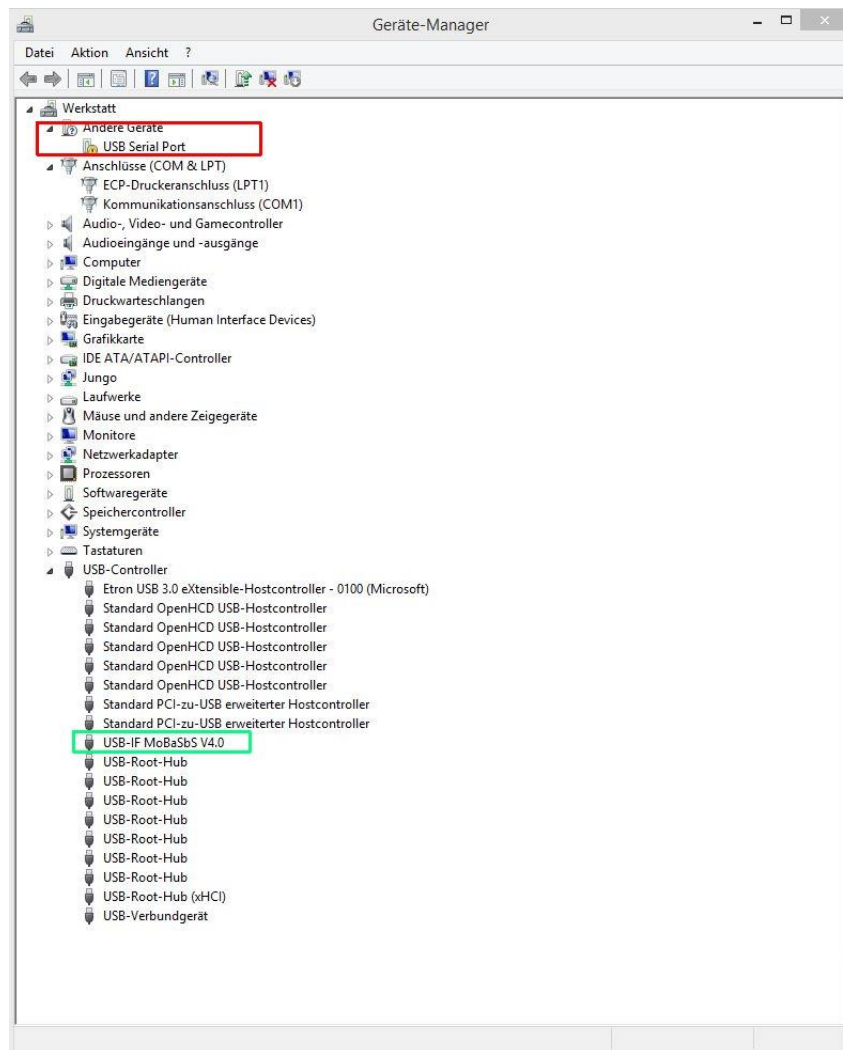


Bild 9 Virtuellen seriellen Port einrichten

Auch hier wird früher oder später der Erfolg der Treiberinstallation bestätigt (Bild 10). Die Nummer des COM-Port wird vom System automatisch vergeben. Den COM-Port muss man sich für die Konfiguration der MoBaSbS-Config SW merken.

Evtl. zeigt der Geräte-Manager mehrerer virtuelle COM-Ports. Ein Grund können anderweitig verwendete „USB -> Serial“ Adapter sein. Ein anderer Grund kann aber auch in zuvor fehlgeschlagenen Installationsversuchen liegen. In dem Fall sollte man nicht benötigte virtuelle COM-Ports deinstallieren – sie verwirren nur.

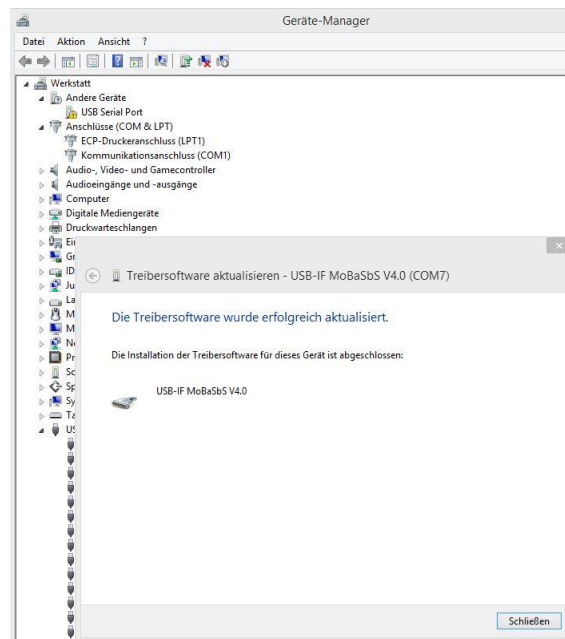


Bild 10 Virtueller COM Port erfolgreich installiert

Vom vollständigen Erfolg der Treiberinstallation zeugen die beiden grün markierten Einträge im Geräte-Manager (Bild 11)

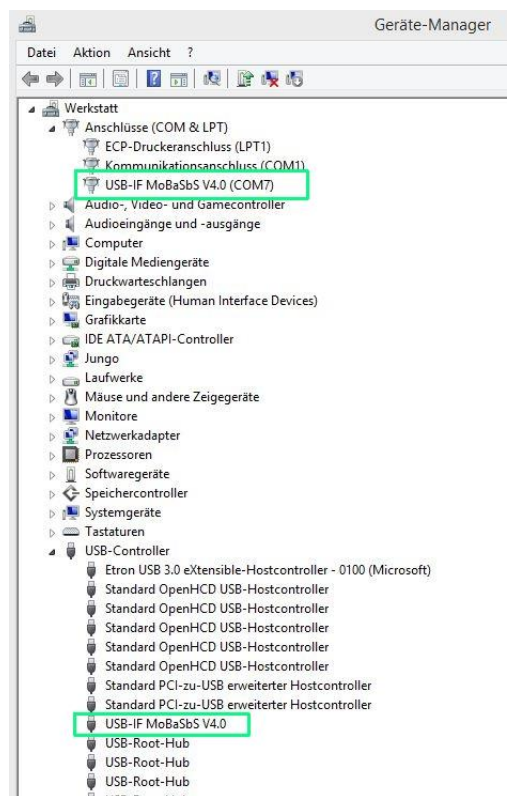


Bild 11 Virtueller COM Port erfolgreich installiert

64-bit Welt ...

.. wer es sich antun will ...

Es gibt einige Erläuterungen, wie man unter einem 64-bit System nicht zertifizierte Treiber installieren kann – Google ist dein Freund. Ich beschreibe hier die Holzhammermethode – und zum Zwecke des Ausprobierens habe ich die Treiber auf meinem „Büro“-Rechner nach diesem Verfahren installiert. Ich bleibe dabei – im Hobbykeller dürfte ein 32-bit System die bessere Wahl sein.

Nach Verbinden der MoBaSbs mit einem USB Anschluss des PC's ergibt sich das folgende (erwartete) Bild (Bild 12)

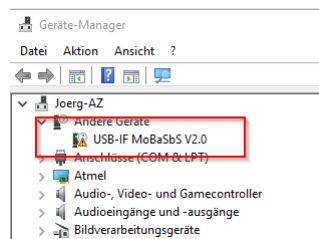


Bild 12 Geräte-Manager nach erstmaligem Anschluss des MoBaSbs IFC an einem PC USB Anschluss

Jetzt versuchen wir wie oben beschrieben, den Treiber zu aktualisieren und bekommen folgende (Bild 13) uns nicht überraschende Fehlermeldung:

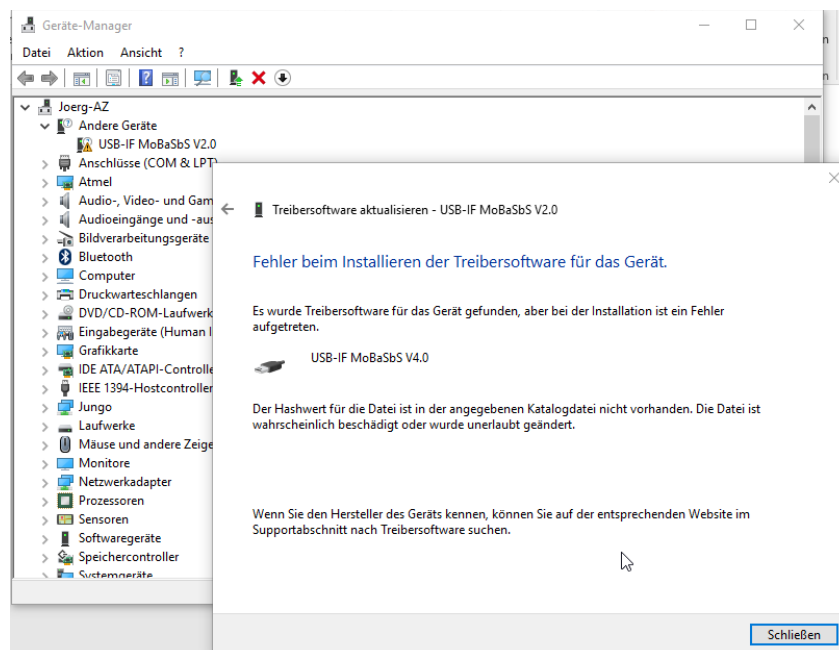




Bild 13 Erfolgreicher Versuch, die Treiber wie unter einem 32-bit Windows zu installieren

Um dennoch den Treiber erfolgreich installieren zu können, müssen wir die Kontrolle der Treibersignatur (temporär) ausschalten. Dazu gehen wir wie folgt vor: links unten auf das Windows-Symbol  und anschließend auf den Power-Button  klicken. Jetzt die Shift-Taste drücken, halten und auf „Neu starten“ klicken (Bild 14). Es wird davon ausgegangen, dass sich die Taskleiste am unteren Bildschirmrand befindet. Falls sich jemand den Desktop anders strukturiert hat, wird es für ihn kein Problem sein, die Position der Buttons dennoch zu finden ;-)

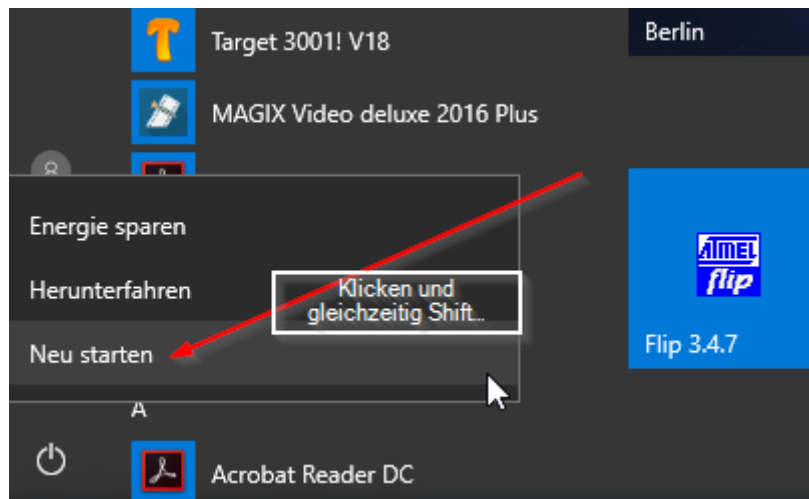


Bild 14 Neustart um Treibersignatur abschalten zu können

Nach wenigen Augenblicken erscheint folgender Bildschirm (Bild 15), wir klicken auf „Problembehandlung“ und im dann erscheinenden Menü auf „Erweiterte Optionen“ (Bild 16).

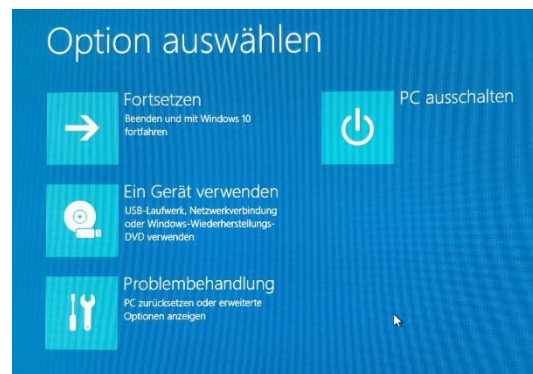


Bild 15 Menü Optionen



Bild 16 Menü Problembehandlung

Weiter geht's mit einem Klick auf den Menüpunkt „Starteinstellungen“ (Bild 17). Im dann erscheinenden Bildschirm (Bild 18) haben wir es zunächst geschafft – „Neu starten“ lautet die Devise



Bild 17 Menü Erweiterte Optionen



Bild 18 Menü Neu starten

Je nachdem was unser PC so leistet geht es mehr oder weniger schnell, bis uns erneut ein Menü anschaut, die „Starteinstellungen“ (Bild 19)



Bild 19 Starteinstellungen

Wir müssen nicht raten, das Menü verrät uns, dass es mit der Taste „7“ (oder „F7“) weitergeht. Nun startet unser PC wie gewohnt und wir können ab sofort so tun, als wären wir in einem 32-bit System, also Geräte-Manager öffnen, Rechts-Klick auf den Eintrag „MoBaSbS USB-IF ...“ (siehe Abschnitt 32-bit Welt...). Die nachfolgenden Screenshots entstammen alle der 64-bit Installation die zu einer problemlosen Verbindungsaufnahme der Config-SW mit dem IFC führte. In Bildern zeigen das die Bilder 20 - 26.

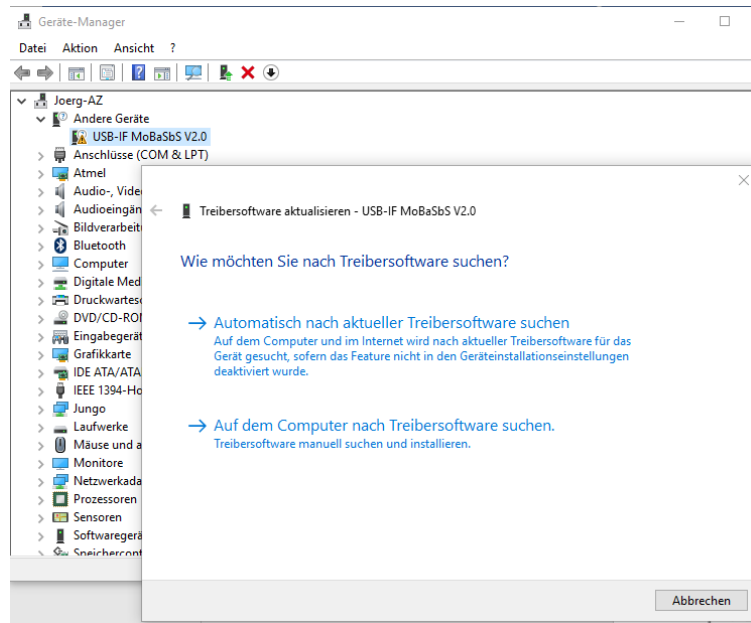


Bild 20

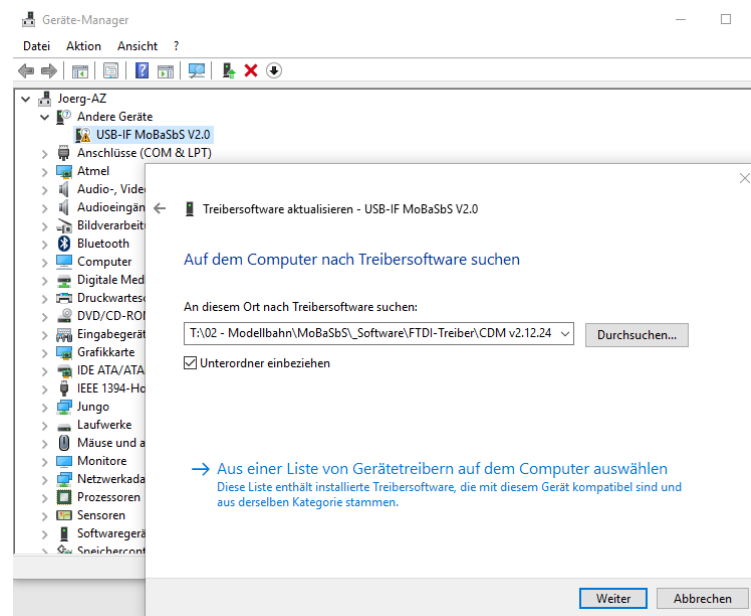


Bild 21

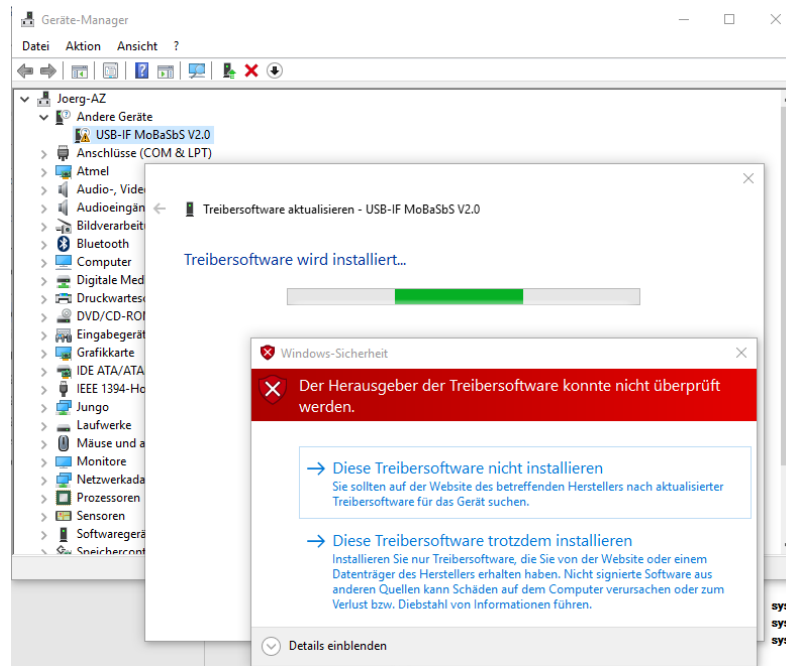


Bild 22

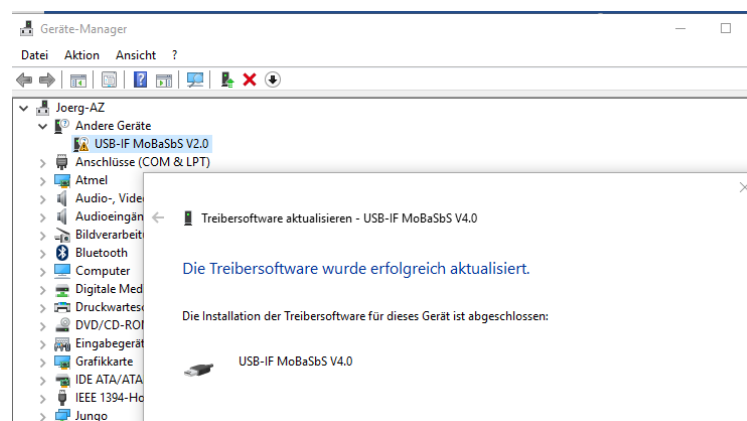


Bild 23

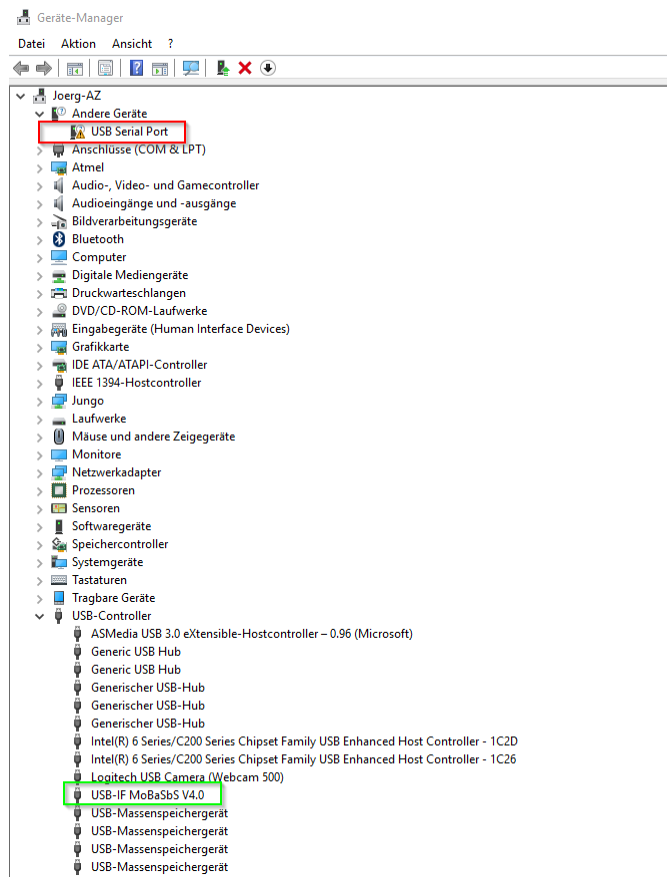


Bild 24

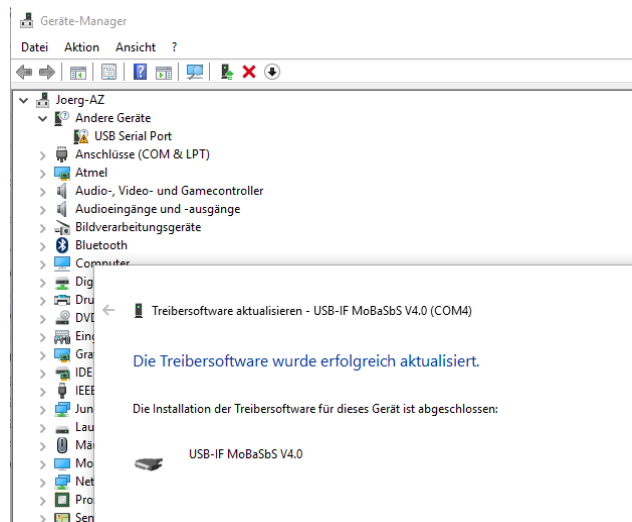


Bild 25

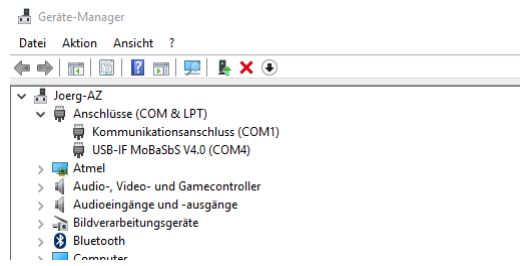


Bild 26

Ach, ja – wenn man jetzt den Rechner neu bootet sieht man rechts unten in etwa folgendes:

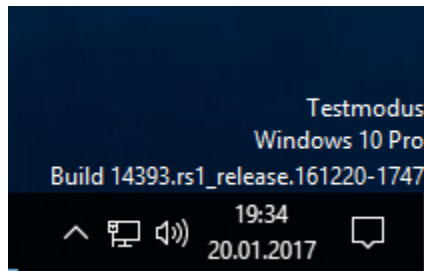


Bild 27

Der Grund – solange die Treibersignatur ausgeschaltet ist, befindet sich Windows eben im Testmodus. Um auch das noch glatt zu ziehen, öffnen wir eine Shell mit Administratorrechten (siehe Bild 28, Rechts-Klick auf „Eingabeaufforderung“) und geben folgendes Kommando ein:
„Bcdedit /Set Testsigning OFF“

Nach dem nächsten Rechnerstart ist alles wie immer – nur, dass wir nun die MoBaSbs über USB anschließen können.

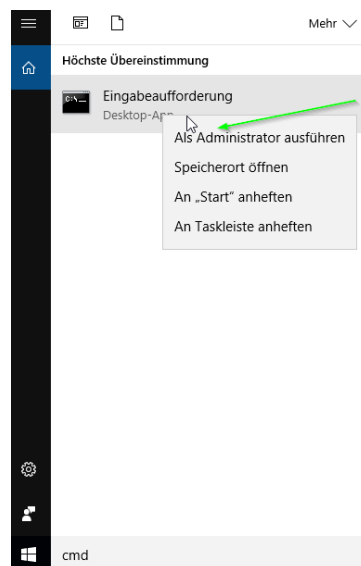


Bild 28

Zusammenfassung

Es ist kein Problem, auch unter aktuellen Windows 10 Versionen unsere MoBaSbs spezifische Schnittstelle zu aktivieren. Insbesondere ist es nicht notwendig, den FTDI Chip wieder in den Ursprungszustand hinsichtlich seiner PID zu versetzen, wenn er einmal mit der MoBaSbs spezifischen PID versehen wurde. Wäre ja auch schade drum.

Das beschriebene Verfahren führt auch unter Win7 und Win 8 zum Erfolg, ggfs. sehen die Menüs bzw. Dialoge optisch etwas anders aus. Weiterhin kann ich nicht überprüfen, ob sich das Startmenü für die 64-Bit Variante unter älterer Hardware, die insbesondere nicht über einen UEFI Bootloader sondern ein klassisches BIOS verfügt, wie beschrieben verhält.

Es kann auch nicht ganz ausgeschlossen werden, dass sich Microsoft in zukünftigen Win10 Versionen weitere „Schikanen“ einfallen lässt. Allerdings bin ich überzeugt davon, dass es immer einen dokumentierten Weg geben wird, proprietäre Treiber zu aktivieren. Allein schon deshalb, weil für exotische ältere Hardware eben nicht zwangsläufig noch neue Treiber entwickelt oder gar (kostenpflichtig) bei Microsoft zertifiziert werden, Microsoft aber letztendlich es sich mit solchen Anwendern doch nicht ganz verscherzen will (wieder meine ganz persönliche Meinung).

Für Rückfragen bin ich unter der Mail Adresse joerg.schi@gmx.de erreichbar.

13.2.3 Installation MoBa PC ab 2015 mit Win10 (von Roman Kircher)

MoBaSbS Treiber USB Chip FTDI

Die Installation des FTDI-Treibers gestaltete sich sehr schwierig. Im Nachhinein ist nicht klar, welcher der vielen Versuche zum funktionierenden Ergebnis führten.

Die aus dem Internet von der FTDI-Seite geladene Datei enthält sowohl 32- als auch 64-Bit Treiber und wird als „Certified“ und für Windows 10 geeignet bezeichnet. Damit müsste der Treiber problemlos installierbar sein. Für die MoBaSbS müssen allerdings die 2 INF-Dateien angepasst werden.

Unter anderem wurde dem Betriebssystem mit folgenden Befehlen die Installation von unsignierten Treibern beigebracht. Ob nach dem Zulassen ein Reboot notwendig ist, ist ebenfalls nicht geklärt.

Win10 unsignierte Treiber zulassen

```
BCDEDIT –Set LoadOptions DDISABLE_INTEGRITY_CHECKS  
BCDEDIT –Set TESTSIGNING ON
```

Einstellung zurücknehmen

```
BCDEDIT –Set LoadOptions EENABLE_INTEGRITY_CHECKS  
BCDEDIT –Set TESTSIGNING OFF
```

Mit dem vom Netzlaufwerk kopierten Treiberverzeichnis ließ sich der Treiber nicht installieren. Meldungen wie „Hash Werte stimmen nicht – Paket korrupt“ und „Treiber für Windows 95 oder höher erstellt“ werden ausgeworfen.

Die Lösung hier war:

das Ausführen der heruntergeladenen Treiberdatei

```
CDM v2.12.06 WHQL Certified.exe
```

am Windows 10 PC direkt.

Der Treiber wird dabei unter

```
C:\Users\Roman\AppData\Local\Temp\FTDI-Driver
```

entpackt und kann zwischendrin von dort wegkopiert werden.

Mit dem Ausführen am PC direkt wird der Teil für den USB-Anschluss „USB-IF MoBaSbS V4.0“ richtig installiert und Windows 10 akzeptiert den Treiber plötzlich. Der Name im Geräte manager wurde bereits früher per MProg in den USB Chip programmiert. Die angepassten INF Dateien enthalten denselben Namen.

Jetzt erscheint im Geräte manager ein unbekannter COMx-Anschluss. Im Normalfall wird diesem Anschluss das eben kopierte Treiberverzeichnis noch mal zur Verfügung gestellt. Aber auch hier brach die Installation mit z. T. oben genannten Meldungen ab. Nach jedem Fehlversuch wird der COMx Anschluss um eine Zahl hochgezählt.

Inzwischen wurde noch versucht, die INF-Dateien am Windows 10 PC direkt zu ändern. Dabei gab es auch die Variante, die originalen INF-Dateien erst mal umzubenennen, dann je eine Kopie zu erstellen und diese zu ändern. Die umbenannten Dateien bleiben im Verzeichnis stehen.

Inzwischen war der COMxx Zähler schon im doppelstelligen Bereich. Nach zurücksetzen auf COM5 wurde die Schnittstelle plötzlich akzeptiert. Irgendeine Installationsvariante hatte also gezogen, nur die zweistellige COMNummer hat das weitere Vorgehen blockiert.

Im Geräte manager unter Anschlüsse steht nun „USB-IF MoBaSbS V4.0 (COM5)“.

Der Windows eigene Zähler des nächsten freien COM-Anschlusses konnte mit dem Tool „COM_Name_Arbitr_Setter.exe“ wieder auf die effektiv nicht benutzten COM-Anschlüsse zurückgestellt werden.

Nachtrag Juni 2016

Windows 10 hat sich auf die Version „1511“ aktualisiert. Das Update hat den Treiber für die MoBaSbS unbrauchbar gemacht (COM5 im Geräte manager, aber mit Ausrufezeichen).

Die oben beschriebenen Aktionen führten nicht zum Erfolg. Die bereits bekannte Fehlermeldung (siehe Screenshot) bleibt hartnäckig bestehen. Von der FTDI Webseite wurde die aktuellste Treiberversion geladen, die INF Dateien erneut angepasst, immer mit demselben Ergebnis.



Um wieder einen stabilen Ausgangszustand zu erhalten, habe ich über das Tool „CDM Uninstaller“ von der FTDI Seite die bestehenden Gerätetreiber entfernt. Über das GUI-Tool zunächst im Fenster die VID (Vendor ID) eintragen, hier „0403“, dann die PID, hier „CB20“. Dieses Gerät dann „entfernen“. Im Geräte manager verschwindet der falsche Eintrag, PC neu booten.

Als nächstes habe ich über den bereits vorhandenen „COM_Name_Arbitr_Setter.exe“ die verbrauchten USB Ports wieder frei gemacht.

Beide Installationen zusammenführend komme ich zu folgendem Ergebnis:

Die FTDI Treiber sind durch die Anpassung der INF-Dateien nicht mehr zertifiziert. Die folgende Befehlsfolge ist mit Windows 10 zwingend erforderlich.

Win10 unsigned Treiber zulassen

BCDEDIT –Set LoadOptions DDISABLE_INTEGRITY_CHECKS

BCDEDIT –Set TESTSIGNING ON

Einstellung zurücknehmen

BCDEDIT –Set LoadOptions EENABLE_INTEGRITY_CHECKS

BCDEDIT –Set TESTSIGNING OFF

Des Weiteren ist am MoBa PC die Benutzerkontensteuerung (UAC) abgeschaltet. Alle Aktionen müssen sonst mit der Option „Als Administrator ausführen“ durchgeführt werden. Man kann die Benutzerkontensteuerung auch nachher wieder einschalten.

Von der FTDI Seite wurde der Virtual COM Port Treiber in der aktuellsten Version als ZIP File geladen

<http://www.ftdichip.com/Drivers/VCP.htm>

Die im Handbuch beschriebenen notwendigen Anpassungen der INF Dateien (siehe Kap. 7.3) führen nicht zum Erfolg. Dieselben Anpassungen an INF-Dateien aus dem Jahr 2014 (älterer Treiber) funktionieren mit der aktuellen Konfiguration.

Die INF Dateien haben in der aktuellen Version mehr Einträge, irgendwas stört sich daran nach der Ergänzung durch die MoBaSbS Zusätze.

Nach der Deinstallation des Treibers wurde die MoBaSbS ausgeschaltet, der PC neu gestartet.

MoBaSbS einschalten, im Gerätemanager am erscheinenden Eintrag auf Treiber aktualisieren klicken. Option „Auf dem lokalen Computer nach Treibern suchen“, das Verzeichnis mit den angepassten INF-Dateien aus dem Jahr 2014 angeben, Treiber installieren.

Der nachstehende Hinweis erscheint nur, wenn unsigned Treiber erlaubt wurden.

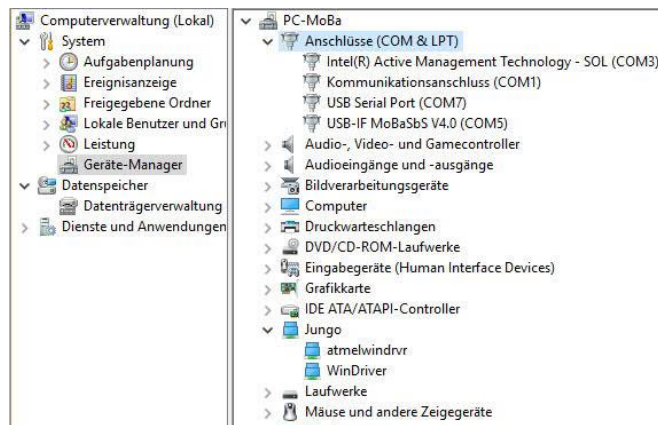


Den Treiber installieren. Im Gerätemanager verschwindet nun MoBaSbS und es erscheint ein unbekannter USB-serial Converter.

PC neu booten.

Im Gerätemanager nun erneut am unvollständigen Gerät auf Treiber aktualisieren klicken, denselben Pfad noch mal angeben. Jetzt wird das MoBaSbS USB Interface wieder angezeigt und funktioniert auch.

Im Gerätemanager schaut es nun aus, wie im Screenshot unten.



Der IFC der MoBaSbS meldet sich als USB Device, im TrainController kann deshalb bei Anschluss „USB“ angewählt werden. Der richtige virtuelle Port wird selbst erkannt. Die Angabe von COM5 funktioniert aber auch. Der LocoBuffer USB hat später die COM7 bekommen.

Nachtrag Februar 2018

Nach dem letzten großen Win10 Update (1709) war zumindest auf meine 64-Bit Arbeitszimmer PC der Treiber weg und die Neuinstallation nach der Anleitung hat sofort funktioniert. Auf meinem 32-Bit Werkstatt PC hat der Treiber das Windows Update ohne Probleme überstanden.

13.2.4 MoBaSbS Notizen USB (von Roman Kircher)

MoBaSbS Installation USB Interface

Vorbereitung

IFC: Switch 1 auf "ON"

MPROG 3.5 downloaden und installieren

FTDI CDM V2.04.06 Treiber downloaden und bereitstellen

ftdibus.inf und ftdiport.inf anpassen (siehe Kap. 7.3 oder auf der Webseite von Patrick)

Achtung, heruntergeladene Dateien hatten den Eintrag "...V2.0" im String Abschnitt --> kontrollieren

USB Kabel anschließen

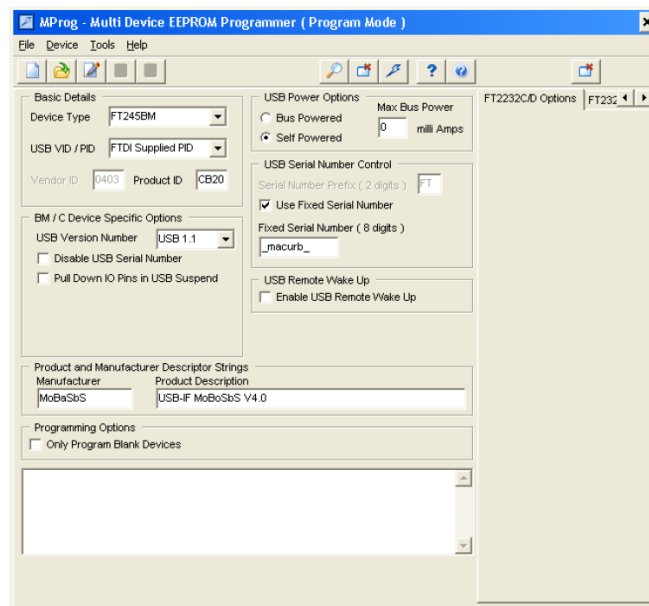
neue Hardware gefunden

2 x das Verzeichnis mit den angepassten inf Dateien angeben

MPROG aufrufen

Achtung, die geladene .ept Datei passte nicht

Nach Beschreibung USB Tutorial (siehe Kap. 13.2.1) die Daten selbst eingegeben und unter MoBaSbS-RK.ept gespeichert



Das erste Lesen (Tools / Read) des Chips bringt lauter Nullen, nach dem Schreiben stehen die entsprechenden Werte drin.

Nach Ab- und Anstöpseln des USB Kabels wird wieder neue Hardware gefunden. Jetzt wird der CDM Treiber gefordert (Pfad angeben).

13.3 Informationen zu Betrieb mit Traincontroller /-programmer

13.3.1 Traincontroller bis einschl. Version 5.5

Die MoBaSbS kann 256 Adressen verwalten. Hierbei können Decoder- und Encoderadressen beliebig vergeben werden.

Traincontroller bis einschl. Version 5.5 erwartet für die Adressen 0 bis 127 Encoder (Melder) und von 0 bis 255 Decoder (ansteuerbare Elemente). Die Adressen von 0-127 können also sowohl für Encoder als auch für Decoder benutzt werden.

Da in Traincontroller die Adressen nicht direkt eingetragen werden können, müssen sie wie folgt umgerechnet werden:

Decoder

TC-Adr = PM-Adr x 4 + 1

TC-Adr = PM-Adr x 4 + 2

TC-Adr = PM-Adr x 4 + 3

TC-Adr = PM-Adr x 4 + 4

Beispiel: PM_DEC8 ist auf Adresse 128 konfiguriert.

Dessen 4 Ausgangspaare lassen sich in TC über die TC-Adressen 513, 514, 515 und 516 ansprechen.

Encoder

Bei den Encodern ist es immer die um 1 erhöhte Adresse. Also: Ein PM_ENC8T auf Adresse 3 konfiguriert, wird in TC über die Adresse 4 (mit den Kontakten 1 bis 8) abgefragt.

Die Tabelle zeigt die Konvertierung einiger Adressen.

MoBaSbS-Encoder-Adresse		Meldernummern in Traincontroller	
von	bis	von	bis
0-0	0-7	1-1	1-8
1-0	1-7	2-1	2-8
2-0	2-7	3-1	3-8
3-0	3-7	4-1	4-8
4-0	4-7	5-1	5-8
5-0	5-7	6-1	6-8
6-0	6-7	7-1	7-8
7-0	7-7	8-1	8-8
8-0	8-7	9-1	9-8
9-0	9-7	10-1	10-8
10-0	10-7	11-1	11-8
11-0	11-7	12-1	12-8
12-0	12-7	13-1	13-8
13-0	13-7	14-1	14-8
14-0	14-7	15-1	15-8
...	...		
...	...		
126-0	126-7	127-1	127-8
127-0	127-7	128-1	128-8

MoBaSbS-Decoder-Adresse	Elementnummern in Traincontroller	
dez	von	bis
0	1	4
1	5	8
2	9	12
3	13	16
4	17	20
5	21	24
6	25	28
7	29	32
...		
...		
128	513	516
129	517	520
130	521	524
131	525	528
132	529	532
...		
...		
254	1017	1020
255	1021	1024

13.3.2 Traincontroller ab Version 5.8

Ab Traincontroller V5.8E1 wird die MoBaSbS direkt unterstützt. Sie muss als Digitalsystem ausgewählt werden. Es gibt keine Einschränkungen bezüglich der Adressen für Encoder und Decoder mehr.

Die Umrechnung der Adressen muss wie bei der Version 5.5 erfolgen.

MoBaSbS-Encoder-Adresse		Meldernummern in Traincontroller	
von	bis	von	bis
0-0	0-7	von	bis
1-0	1-7	1-1	1-8
2-0	2-7	2-1	2-8
2-0	2-7	3-1	3-8
3-0	3-7	4-1	4-8
4-0	4-7	5-1	5-8
5-0	5-7	6-1	6-8
6-0	6-7	7-1	7-8
...			
...			
128-0	128-7	129-1	129-8
129-0	129-7	130-1	129-8
130-0	130-7	131-1	131-8
131-0	131-7	132-1	132-8
132-0	132-7	133-1	133-8
133-0	133-7	134-1	134-8
134-0	134-7	135-1	135-8
...			
...			
254-0	254-7	255-1	255-8
255-0	255-7	256-1	256-8

MoBaSbS-Decoder-Adresse	Elementnummern in Traincontroller	
dez	von	bis
0	1	4
1	5	8
2	9	12
3	13	16
4	17	20
5	21	24
6	25	28
7	29	32
...		
...		
128	513	516
129	517	520
130	521	524
131	525	528
132	529	532
133	533	536
134	537	540
...		
...		
254	1017	1020
255	1021	1024

13.3.3 Traincontroller ab Version 7.0 Silber und Gold

(TC 7.0 Bronze wird nicht betrachtet. In der Bronze-Version sind keine Melder (Encoder) möglich)

Die MoBaSbS wird direkt unterstützt. Sie muss als Digitalsystem ausgewählt werden.

Adressierung

Ab Traincontroller V7.0 werden die MoBaSbS-Adressen direkt in TC angezeigt. Eine Konvertierung entfällt damit.

Z.B. wird eine Weiche in TC nun über die (PM-)Adresse und in einem separaten Feld über die Adresse des Ausgangspaares (1, 2, 3 oder 4) definiert.

Bei einem Einzelkontakt sieht das genauso aus. Zuerst kommt die Moduladresse und dann folgt die (Sub-)Adresse des Ausgangspaares. Jetzt kann man noch definieren, ob man Kontakt 1 oder 2 des Paares ansprechen möchte.

Werden für ein Element mehr als zwei Ausgänge benötigt, werden die Folge-Adresse und deren Ausgänge automatisch mit angezeigt.

Bei einem Encoder verhält es sich ähnlich. Erst wird die (PM-)Adresse angegeben und danach wird der Eingang (1 bis 8) festgelegt.

Symbole

Bei den Weichen und Schaltern/Tastern gibt es MoBaSbS eigene Symbole in der Kontaktdarstellung. Durch einen blauen Pfeil auf gelbem Grund wird symbolisiert, in welche Richtung (links/rechts) geschaltet wird. Damit deckt sich auch diese Darstellung mit der im HDC.

Synchronisieren der Lokdatenbank

Die folgenden Eigenschaften stehen nur mit der MoBaSbS-Software-Version vom 19.10.2008 oder aktueller zur Verfügung.

Ab TrainController 7.0 D1 findet man unter dem Menu 'Zug' den Eintrag 'Lokdatenbank der Zentrale'.

Wenn man diesen Eintrag anklickt, öffnet sich ein Fenster mit einer Tabelle, in der alle bereits in Traincontroller definierten/hinterlegten Loks aufgelistet sind. Im rechten Teil des Fensters gibt es 4 Knöpfe. Der Knopf 'Lesen vom Digitalsystem' liest alle Datensätze aus der MoBaSbS aus. Nun kann man Loks aus dieser Liste anklicken und der Knopf 'Anwenden auf TrainController' wird aktiv.

Drückt man diesen Knopf, so werden die zuvor ausgewählten Loks in TrainController angelegt. Dabei gibt es noch die Abfrage, ob auch eine bereits in der MoBaSbS hinterlegte Funktionstastenzuordnung mit übernommen werden soll (was man i.d.R. wohl auch mit Ja bestätigen wird).

Umgekehrt lassen sich über die Tabelle auch Loks, die bereits in TC angelegt wurden, aber noch nicht in der MoBaSbS gespeichert sind, in der MoBaSbS hinterlegen. Aber Achtung! Sie werden nur im flüchtigen Speicher abgelegt und stehen nach Power-Off/Reset nicht mehr zur Verfügung.

Über den HDC kann man aber in den BMC Optionen das SRAM (der flüchtige Speicher) in das EEPROM (den nicht-flüchtigen Speicher) der MoBaSbS übertragen.

Weiterhin kann auch das Datenformat in der Tabelle für eine Lok angegeben werden (Pull-Down-Menu). TrainController kommt ohne die Info aus, mit welchem Format (DCC/MM-Fahrstufen) angesteuert wird, denn TC liest diese Info aus der MoBaSbS bei jedem Neustart der Software aus.

Wurde nun aber eine Lok in TrainController angelegt, muss dies der MoBaSbS mitgeteilt werden. Deshalb muss diese Info zumindest einmal an die MoBaSbS gesendet werden.

13.3.4 Trainprogrammer ab V7.0

Programmierung der DCC-Variablen

Trainprogrammer bietet die MoBaSbS nicht als Digitalsystem zur Programmierung an. Wenn aber die MoBaSbS als Zentrale konfiguriert wurde, muss man das Häkchen bei 'Zentrale zum Programmieren verwenden' setzen. Alternativ kann das Lenz Interface direkt in Trainprogrammer gewählt werden.

13.4 Bootlader-Bedienungs-Anleitung (Lintzmeyer-Variante)

Von Günter Lintzmeyer wurde die Funktionalität einiger Peripheriemodule erweitert. Hierfür hat er die folgende Beschreibung (Stand Version 2 von Februar 2012) erstellt.

Mit Hilfe eines Bootladers ist es möglich, hex- oder eep-Dateien (z.B. neue Software-Versionen) über den Xbus in den Prozessor zu laden, ohne den Prozessor aus der Platine zu entfernen.

Hierfür ist es allerdings Voraussetzung, dass der Prozessor bereits das Programm mit der Bootlader-Routine enthält. Dies ist mit Hilfe eines Programmieradapters (STK500 oder μ C-Programmierboard/PonyProg) zu laden.

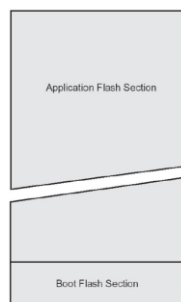
zur Verfügung steht folgende Decoder-/Encoder-Software:

- PM_DEC8_Atmega8_BL . hex / *. eep
- PM_SDEC4_Atmega8_BL . hex / *. eep
(enthält Funktion für invertierte Drehrichtung und Erweiterung für die Ansteuerung der Herzstück-Polarisierung)
- PM_SDEC8_Atmega8_BL . hex / *. eep (enthält keine Funktion für invertierte Drehrichtung und keine Erweiterung für die Ansteuerung der Herzstück-Polarisierung)
- PM_ENC8_Atmega8_BL . hex / *. eep

Für den PM-Uni existieren nicht kompatible Bootlader (siehe Kap.13.6.4)

Es ist zu beachten, dass der für einen Modul-Typ (z.B. PM_DEC8) programmierte Prozessor späterhin auch nur mit dem hex-file für den gleichen Modul-Typ über das Config-Tool und Bootlader beschrieben werden kann, d.h. ein (z.B.) für den Modul-Typ PM_DEC8 programmierter Prozessor kann nicht mit der Bootlader-Software z.B. für den PM_ENC8 über das Config-Tool (um-)programmiert werden. In diesem Fall muss zuerst über den Programmieradapter und PonyProg das hex-file (mit BL-Routine) für den Modul-Typ (PM_ENC8) eingespielt werden

Die Boot Flash Section wurde für die MoBaSbS-Programme einheitlich festgelegt:

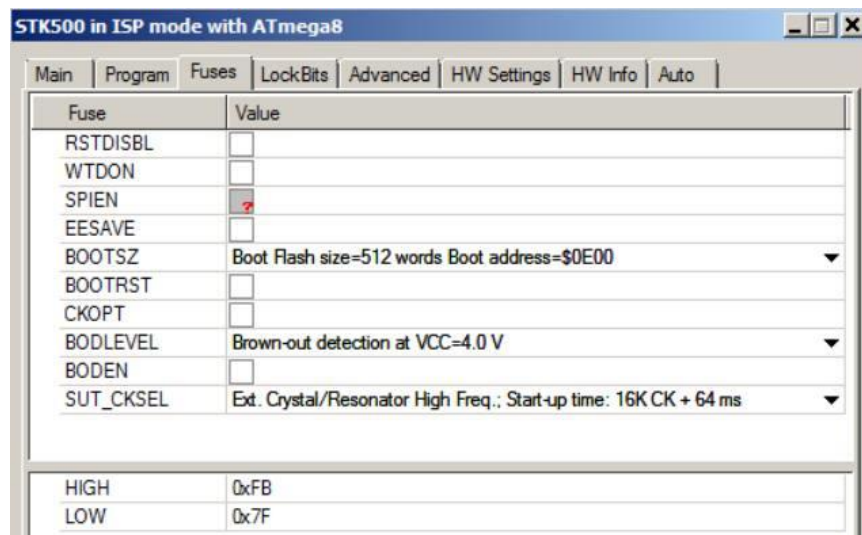


Einteilung des Flash-Speichers

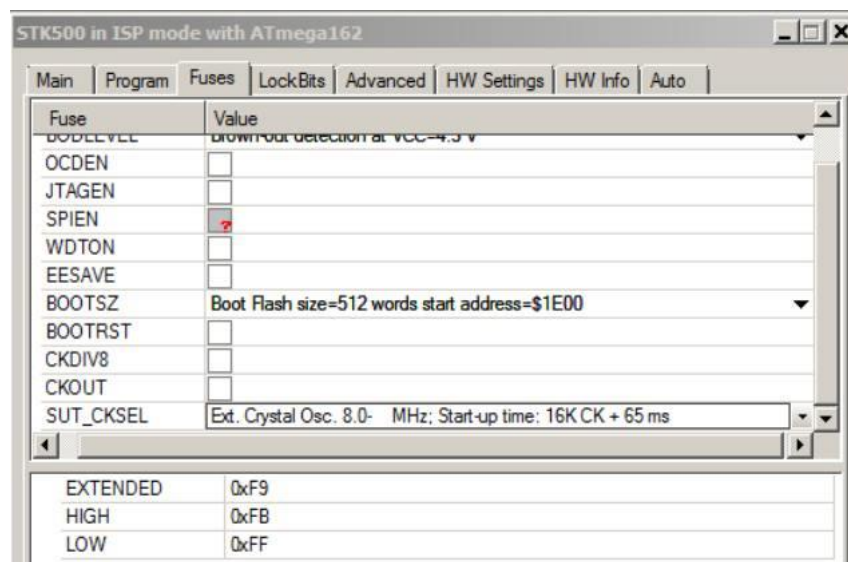
- beim ATmega8 beginnt die Boot-Flash-Section bei 0x0E00
- beim ATmega162 beginnt die Boot-Flash-Section bei 0x1E00

Hierfür sind die entsprechenden Fusebits zu setzen:

- BOOTSZ1 = 0 (gesetzt / programmiert, Ponyprog Haken gesetzt)
- BOOTSZ0 = 1 (nicht gesetzt / nicht programmiert, Ponyprog kein Haken gesetzt)
- BOOTRST = 0 (gesetzt / programmiert, Ponyprog Haken gesetzt)

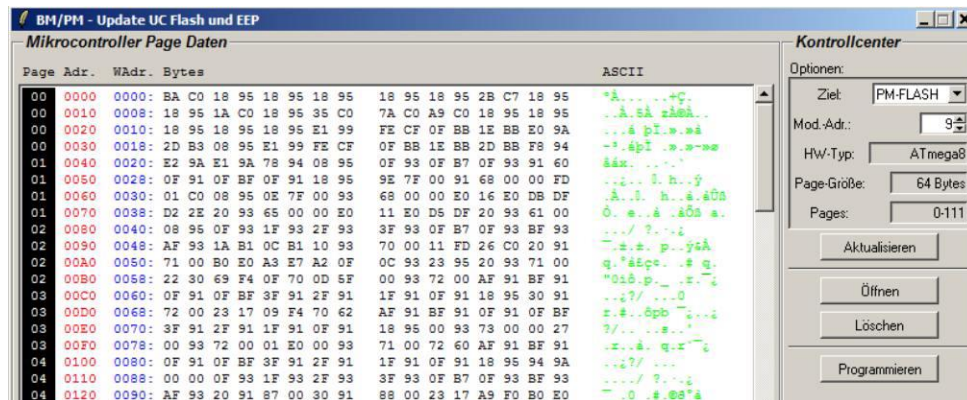


ATmega8, Fusebits mit STK500 bei Nutzung des Lintzmeyer-Bootladers



ATmega162, Fusebits mit STK500 bei Nutzung des Lintzmeyer-Bootladers

Die Datenbereitstellung erfolgt mit Hilfe des Konfigurationstools ab Version 4.5. Dazu müssen zunächst die zu übertragenden Files (hex bzw. eep) in den Ordner "patch" des Konfigurationstools (im Allgemeinen >MBSBcfgSW <) kopiert werden. Danach ist der Dialog ‚Werkzeuge > UC Update Flash‘ ist aufzurufen:



Flash-Update, hex-Datei

- >>> Ziel auswählen (hex-File >> PM-FLASH / eep-file >> PM-EEP)
 - >>> Mod.Adr. auswählen
 - >>> Aktualisieren (die dunkel gesteuerten Felder werden aktualisiert)
 - >>> Öffnen: die zu übertragende Datei (*.hex bzw. *.eep) selektieren (Order "patch" - s. oben)
Warten, bis der Inhalt der hex-Datei auf dem Dialog-Bildschirm dargestellt ist
 - >>> Programmieren
- Der erfolgreiche Abschluss der Programmierung **muss** mit folgender Meldung quittiert werden



Quittierungsmeldung

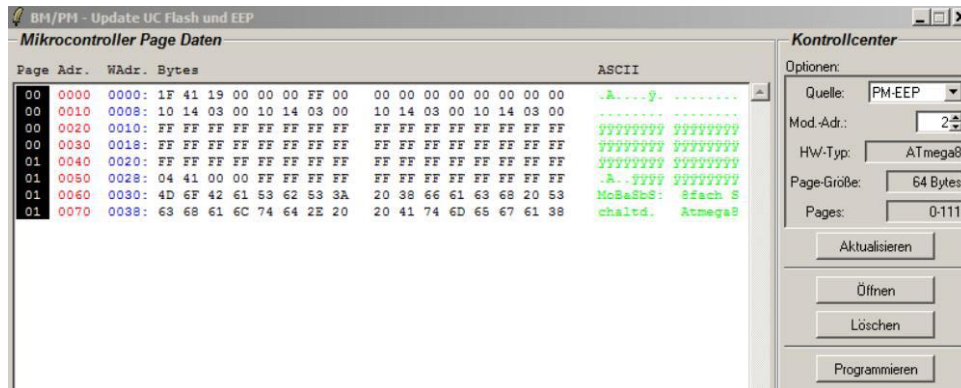
Falls ein Fehlverhalten des Decoders festgestellt wird, hilft ein Hardware-Reset.

Nach dem Programmieren der *.eep-Datei wird die bisher im eeprom vorhandene Modul-Adresse wieder rekonstruiert.

Beispiel:

aktuelle Modul-Adresse = 2

Modul-Adr. eeprom-file = FF (255)



Flash-Update, eep-Datei

Nach dem Programmieren ist das Modul weiterhin mit Modul-Adr. 2 anzusprechen. Es werden nur die Speicherbereiche überschrieben, die in der eep-Datei signifikant sind. Speicherbereiche, die nicht in der eep-Datei enthalten sind, bleiben im eeprom erhalten. Soll der gesamte eeprom-Bereich gelöscht werden, ist dies z.B. mit PonyProg oder STK500 durchzuführen.

Der erfolgreiche Abschluss der Programmierung **muss** mit folgender Meldung quittiert werden



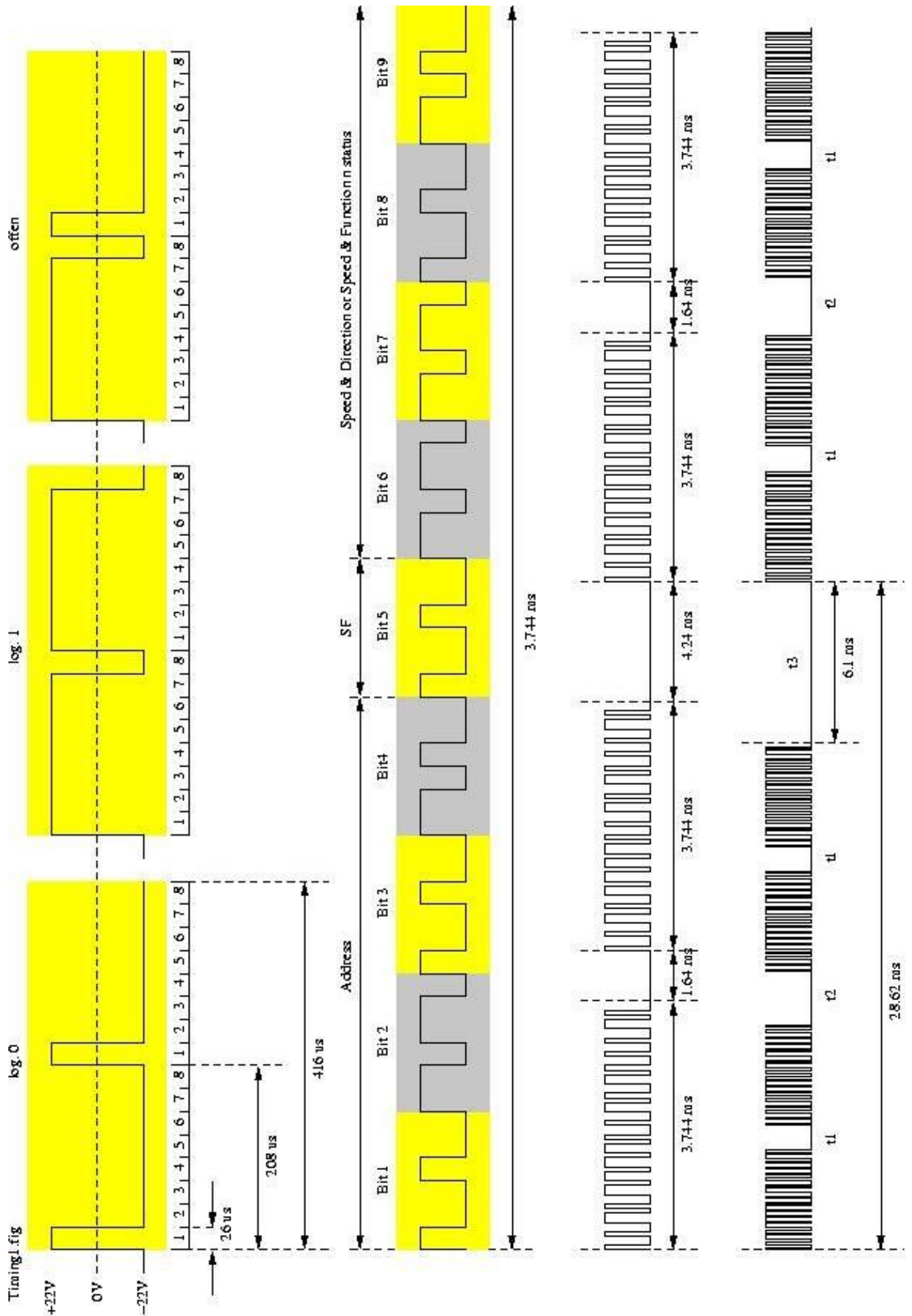
Quittierungsmeldung

Achtung:

Mit einer neuen eep-Datei werden natürlich die vorher evtl. eingestellten Parameter überschrieben, diese können aber mit der Datei MoBaSbS.mbs rekonstruiert werden

13.5 Timing Diagramm Motorola II-Protokoll

Von Patrick Urban wurde das folgende Timing-Diagramm erstellt.



Timing-Protokoll, Motorola2-Format

13.6 Informationen zum PM-Uni

Von Uwe Mäusezahl wurde die Platine und der Code für den PM-Uni (siehe Kap. 8.2.6) erstellt. Hierfür hat er die folgende Beschreibung erstellt.

Der PM-Uni ist ein Universalperipheriemodul für die MoBaSbS. Der Grundgedanke für die schon lange geplante Umsetzung ist die gleichzeitige Verwendung von Servo- und Dekoderausgängen sowie Encodereingängen in einem Modul. Dies macht sich besonders bei Modul- und Segmentanlagen positiv bemerkbar, da nicht für jeden Anwendungsfall ein eigenes Peripheriemodul verwendet werden muss. Außerdem verringert sich die Lagerhaltung für Servo, Encoder und Dekoder auf eine Platine. Für dieses Modul habe ich unter teilweiser Verwendung der bisherigen Routinen einen Universalcode entwickelt. Diesen möchte ich hier vorstellen und Tipps für den Aufbau und die Verwendung geben.

13.6.1 Einleitung

Mit dem neuen Code ergeben sich folgende Konfigurationsmöglichkeiten. Im Normalfall wird nur eine Adresse verwendet und damit kann dann das PM-Uni als 4-fach Servo- oder 8-fach En-/Dekoder verwendet werden.

Dabei können in der Einstellung En-/Dekoder die Ein- bzw. Ausgänge des Moduls paarweise gemischt werden. Bei der Verwendung eines ULN2803 als Treiber ist das Mischen bauteilebedingt auf jeweils 4 zusammenhängende Ein- bzw. Ausgänge beschränkt.

Bei der Verwendung als 4-fach Servo gibt es die zuschaltbare Möglichkeit eine Rückmeldung bei Erreichen der Endstellung auszulösen und die Möglichkeit festgelegte Ausgänge des PM-Uni für die Herzstückpolarisation zu verwenden (siehe Kap. 13.6.6).

Für den Fall, dass noch weitere Servo-, Dekoder- oder Encoderfunktionen benötigt werden, hat man die Möglichkeit eine zweite Adresse in diesem PM zu aktivieren. Dann stehen noch einmal zusätzlich ein 4-fach Servo- oder 8-fach En-/Dekoder zur Verfügung.

Übrigens erhält man die optimale Ausnutzung des PM-Uni bei Verwendung der Servofunktion auf Adresse 2 und der En-/Dekoderfunktion auf Adresse 1.

Was ich leider nicht realisieren konnte war die Einbindung der komplexen Routinen des LDEC in den Universalcode. Es besteht aber die Möglichkeit den Universalcode gegen den modifizierten PM_LDEC_uni_T Code (mit Taktgenerator) und auch umgekehrt über eine Bootladerfunktion auszutauschen.

Der PM_universal Code ist im Archiv von Uwe Mäusezahl (siehe Kap. 5) unter der Lasche PM Uni ganz am Ende der Beschreibung zu finden.

Der Code enthält einen Fehler beim Speichern der EEPROM-Daten. Es gibt deshalb eine neue Version, die im Config Tool mit V04.00.003 angezeigt wird.

Der Code kann von dem geschützten MoBaSbS-Bereich heruntergeladen werden.

Der PM_LDEC_uni_T Code (V4.02) im Archiv von Uwe Mäusezahl (siehe Kap. 5) unter der Lasche LDEC ganz am Ende der Beschreibung zu finden.

Hierzu auch die Hinweise in 13.6.4 beachten.

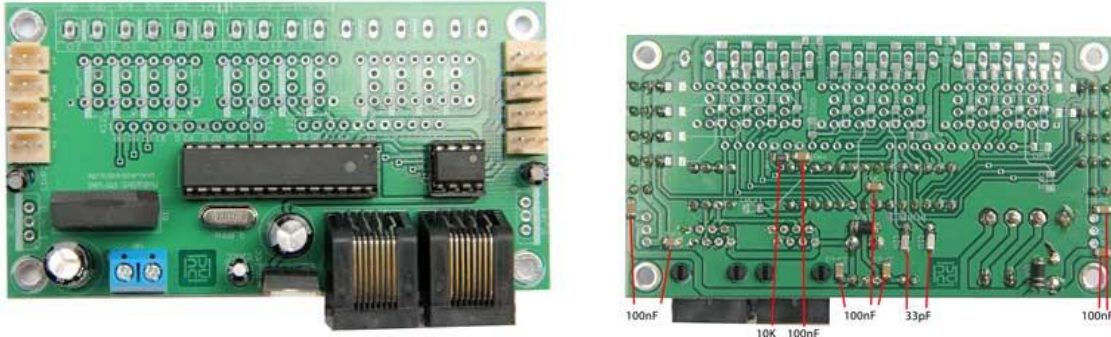
13.6.2 Hardware

Die Hardware für den PM Uni wurde parallel zur Software entwickelt, so dass viele Ideen auf beiden Seiten sofort umgesetzt werden konnten. Mein besonderer Dank gilt Uli, ohne den dieses Projekt noch viele Jahre in der Schublade geschlummert hätte. Von der Universalplatine gibt es zwei Ausführungen, einmal die professionell hergestellte doppelseitige und eine einseitige Platine zum selbst ätzen. Die erforderlichen Unterlagen und die Beschreibung der Hardware kann man auf der MoBaSbS-Homepage anschauen und herunterladen.

Im Folgenden gehe ich nur auf Besonderheiten in Bezug auf den Universalcode ein. Dabei beziehen sich die Ausführungen auf die doppelseitige Platine.

Da der tatsächliche Verwendungszweck erst kurz vor dem Einbau festgelegt wird, habe ich meine PM-Uni Platine mit den Grundbauelementen versehen.

Auf den Fotos ist das Herzstück des PM-Uni, der Atmega8, die Komponenten für die Kommunikation mit dem PMC und schon erste Bauteile für die Spannungsversorgung zu sehen. Soll der komplette PM-Uni nur als Encoder verwendet werden, können die Bauteile auf der linken Seite für die Spannungsversorgung weggelassen werden. Auf dem rechten Foto ist die Anordnung der SMD Bausteine auf der Unterseite zu sehen.



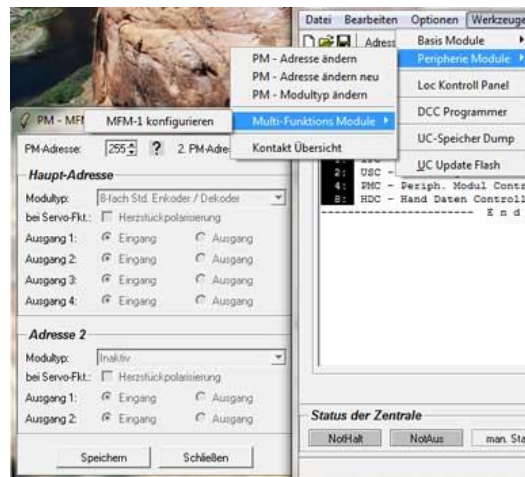
PM-Uni, Platine mit Grundelementen

13.6.3 Einstellungen

Die Grundeinstellung für das Universalmodul ist Moduladresse 255. Diese sollte man gleich mit Inbetriebnahme des Moduls ändern, um später nicht in Konflikt mit neu hinzugefügten Modulen zu kommen. Dies geschieht mit dem Menüpunkt PM Adresse ändern im Konfigurationstool. Nach dem anschließenden Rescan, der Aktualisierung und der Übernahme, steht das Modul jetzt im Konfigtool zur Verfügung.

Kommen wir nun zur Auswahl der Aus- bzw. Eingänge des Modules.

Wie bereits weiter oben erwähnt, können die Anschlüsse des Modules sowohl als Ein- bzw. Ausgänge oder Servo verwendet werden. Die Konfiguration wird über den Menüpunkt Werkzeuge, MFM-1 konfigurieren ausgewählt.



Bei der nun erscheinenden Maske gibt man oben die Adresse 1 (Hauptadresse) an und klickt auf das Fragezeichen. Ist die ausgewählte Adresse ein Universalmodul, kann man nun die Einstellungen dafür vornehmen. Sollte man versehentlich die Adresse 2 des PM ausgewählt haben, erscheint eine Fehlermeldung und die Adresse 1 wird vorgeschlagen. Bei Modultyp stehen zur Auswahl

- 8-fach Std. Encoder / Dekoder
- 4-fach Servodekoder mit und ohne Rückmeldung

Wählt man 8-fach Std. Encoder / Dekoder können jetzt noch die Ausgänge für die zukünftige Verwendung entweder auf Eingang (Encoder) oder auf Ausgang (Dekoder) gesetzt werden. Anschließend die Auswahl noch speichern und nach Aufforderung den RESCAN durchführen. Bei der anschließenden Aktualisierung des PMC und der Übernahme der neuen Daten steht nach der Auswahl des Universalmoduls die Maske mit den zuvor gewählten Ausgängen zur Verfügung. Dort werden dann die gewünschten Einstellungen für den jeweiligen Ausgang, so wie bisher schon von den Encoder- bzw. Dekodern bekannt, durchgeführt (siehe Kap. 13.6.5 und 13.6.6).

Reichen 4 Servo oder 8 Ausgänge nicht aus, so kann in diesem Menüpunkt die zweite Adresse aktiviert werden. Die Einstellungen dafür sind dieselben wie bisher für die erste Adresse beschrieben. Einziger Unterschied ist, dass für die Einstellung im oben gezeigten Menü die Adresse von der zuerst geänderten Adresse (Hauptadresse) verwendet werden muss. Bei einer Fehleingabe wird dies vom Konfigtool mit einer Fehlermeldung quittiert und die richtige Adresse vorgeschlagen.

13.6.4 Bootladerfunktion

Der PM-Uni ist mit einer Bootladerfunktion ausgestattet. Somit ist es möglich, evtl Updates des Programmcodes ohne Ausbau des Moduls aus der Anlage durchzuführen.

Weiterhin kann ohne Neuprogrammierung des Prozessors, der nicht mehr im Universalcode untergebrachte LDEC-T Code eingespielt werden. Auch der LDEC-T Code kann ohne Neuprogrammierung mit dem Bootlader wieder mit dem Universalcode überschrieben werden.

Bitte beachten:

Die Nutzung der Bootladerfunktion ist nur zwischen dem LDEC-T Code (siehe Kap. 12.6.7) und dem Universalcode möglich. Hierbei darf das Fusebit BOOTRST nicht gesetzt sein (siehe Kap. 10.2.3).

Alle bisherigen Codeversionen mit Bootladerfunktion (von G. Lintzmeyer) sind nicht kompatibel (siehe **Fehler! Verweisquelle konnte nicht gefunden werden.** und 13.4). Die Lintzmeyer-Bootlader benötigen in gesetztes BOOTRST.

Dies ist bedingt durch die Neuprogrammierung der Bootladerfunktion, da die bisher verwendete Version nicht passend zur Größe des neuen Codes ist. Nach der Programmierung muss kurz die Spannung zum Universalmodul getrennt werden.

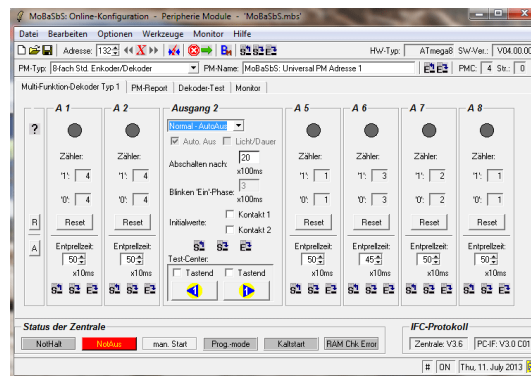
13.6.5 PM-Uni mit De- und Encoderfunktion

Hat man bei der jeweiligen Adresse 8-fach Std. Encoder/Dekoder (4-fach Std. Encoder/Dekoder) ausgewählt und dies anschließend gespeichert, einen Rescan ausgeführt, die PMC Maske aktualisiert und die Daten übernommen, gelangt man im Konfigurationstool auf die sich nun öffnende Maske für Encoder/Dekoder.

Dort erscheint jetzt die vorher gemachte Auswahl bei Ein- Ausgang, in unserem Beispiel die Ausgänge 1,3 und 4 als Encoder und Ausgang 2 als Dekoder.

Für dieses Beispiel habe ich eine Universalplatine bestückt. Dort verwende ich Adresse 1 gemischt als Encoder/ Dekoder mit 2 MOSFET und Adresse 2 als Dekoder mit ULN2803 als Ausgangstreiber.

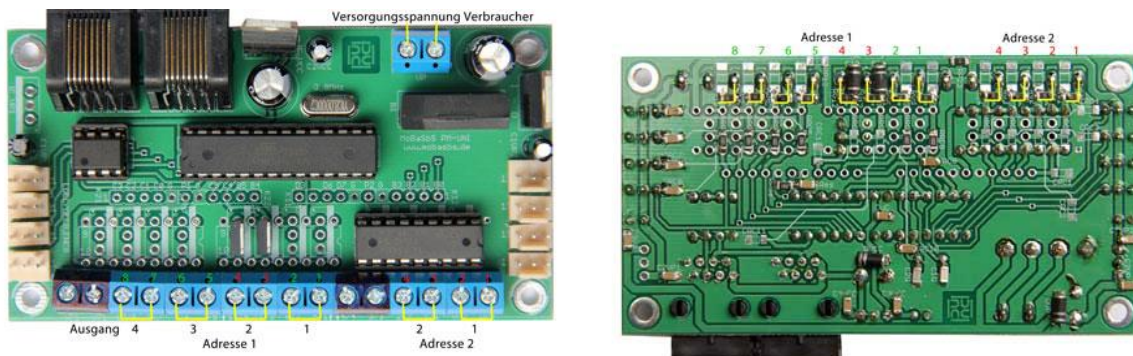
Im Konfigurationstool ist nur die Adresse 1 dargestellt. Die Einstellungen für Adresse 2 sind selbst-erklärend.



In dieser Maske können nun die bereits bekannten Einstellungen für Encoder oder Dekoder vorgenommen werden. Die Funktionen sind identisch mit den bisher verwendeten Encoder- bzw. Dekodereinstellungen bei den bisherigen Peripheriemodulen.

In den folgenden Bildern wird die Zuordnung zu der obigen Einstellung gezeigt.

ACHTUNG: In der ersten Serie der professionell hergestellten Platinen ist ein Fehler beim Aufdruck vorhanden. Dieser bezieht sich auf die Ausgänge für Adresse 2. In den Bildern hier ist die Anschlussbelegung richtig dargestellt.



Versorgungsspannungen

Bei der Versorgungsspannung für externe Verbraucher gibt es viele Möglichkeiten. Man hat die Möglichkeit diese Spannung bzw. Spannungen extern einzuspeisen oder intern erzeugen zu lassen. Dies ist abhängig vom Verwendungszweck. Dabei werden die Verbraucher vom ULN 2803 oder den MOSFET immer nach Masse geschaltet.

Externe Spannungen können an den im oberen Bild schwarz markierten Anschlussklemmen eingespeist werden z.B. die Spannung für die Weichenantriebe. Dabei wird immer nur der Funktionsblock (2 Ausgangspaare) eingespeist, wo sich die Klemme befindet. Möchte man auch den mittleren Funktionsblock mit der Spannung versorgen, so muss wie im unteren Bild ein 0 Ohm Widerstand oder eine Drahtbrücke eingelötet werden. Dieser Punkte befinden sich oberhalb zwischen der Anschlussklemme für Ausgang 5 Adresse 1 und zwischen Adresse 1 und 2. Möchte man mit nur einer Einspeisung beide Adressen (alle 3 Funktionsblöcke) versorgen, überbrückt man beide Trennstellen.

Ähnlich verhält es sich mit der intern erzeugten Spannung. Diese wird zentral an der Anschlussklemme für Versorgungsspannung eingespeist und bei Bedarf mit dem Brückengleichrichter gleichgerichtet. Dann hat man die Möglichkeit jedem Strang über einen Spannungsregler zu stabilisieren. Diese Spannungen können dann wie bei der externen Spannung beschrieben, auf einzelne oder alle Funktionsblöcke aufgeteilt werden. Die schwarz markierten Anschlussklemmen, dienen dann der Ausgabe der Versorgungsspannung für die externen Verbraucher.

Wichtig ist, dass bei Verwendung einer externen Spannung das Massepotential mit dem Universalperipheriemodul verbunden werden muss. Diese Leitung wird an eine der internen Klemmen für Versorgungsspannung angeschlossen. Anschließend ist eine Drahtbrücke vom nicht bestückten Wechselspannungseingang des Gleichrichters zum Massepotential desselben zu löten. Alternativ kann auch das Massepotential der Servoanschlüsse genutzt werden.

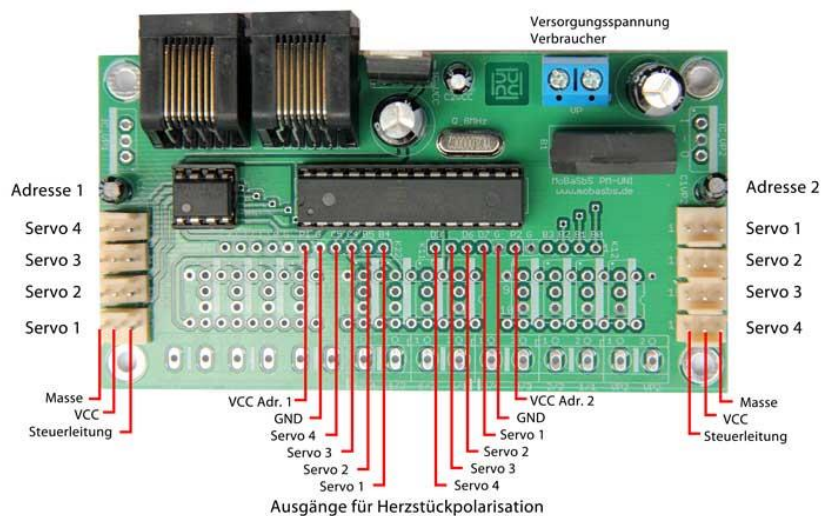
13.6.6 PM-Uni mit Servofunktion

Die Auswahl für die Verwendung als Servo, wird - wie bei En-/ Dekoder beschrieben - über den Menüpunkt Werkzeuge, MFM-1 konfigurieren ausgewählt.

Als Auswahl für Servo steht die Variante mit oder ohne Rückmeldung zur Verfügung. Wählt man Servo mit Rückmeldung, so wird der Dekoder rückmeldefähig und sendet bei jedem Erreichen der Endstellung die aktuelle Position an die Steuerungssoftware. Dies kann genutzt werden, um weitere Schalthandlungen auszulösen. Außerdem lässt sich auf diese Weise, eine mit dem HDC umgeschaltete Weiche auch im Steuerungsprogramm anzeigen.

Zusätzlich ist es möglich über den Menüpunkt Werkzeuge, MFM-1 eine Herzstückpolarisierung zuzuschalten. Auf diese Funktion gehe ich weiter unten in einem eigenen Abschnitt näher ein.

Anschlüsse




Im Bild oben ist für beide Adressen die Anschlussbelegung dargestellt. Die Spannungsregler von 5V für die Servos, sind dort noch nicht bestückt. Verwendet man die Herzstückpolarisierung, müssen diese die Spannung für die Relais mit bereitstellen und sollten dann dementsprechend dimensioniert werden.

Ich verwende dann 2A Typen (78S05) und versee sie bei einer Einspeisung von 12V Versorgungsspannung mit einem kleinem Kühlkörper

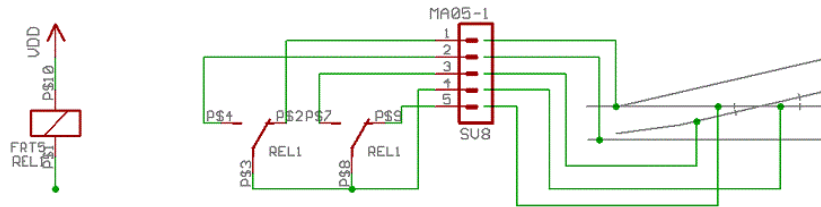
Herzstückpolarisierung

Seit dem Umbau meiner Anlage und dem Einbau von neuen Weichen ist die Herzstückpolarisierung auch für mich ein Thema. Das Herzstück der Weiche ist nun nicht mehr aus Kunststoff sondern aus Metall und macht damit eine stellungsabhängige Einspeisung notwendig.

Im obigen Bild sind die Steuerleitungen für die Relais aufgeführt, so dass jeder nach eigenen belieben eine eigene Platine entwerfen kann.

Die Spannung für das Relais wird im Nulldurchgang des Servo zu- oder abgeschaltet. Das Relais zieht dabei immer an, wenn nach <- geschaltet wird. Also nach Drücken der Taste  auf dem HDC oder im Konfigtool.

Bei meiner im Folgenden vorgestellten Variante wird pro Weiche nur ein Relais mit 2x Umschalter verwendet. Da ich öfter darauf angesprochen werde, dass man für die Herzstückpolarisation 2 Relais benötigt, gehe ich kurz auf die Wirkungsweise mit einem Relais ein.



Bei meinen verwendeten Weichen sind die Weichenzungen und das Herzstück voneinander isoliert. Die Weichenzungen bekommen im Normalfall die Spannung wenn sie an der Stammschiene anliegen, ansonsten sind sie spannungslos. Das Herzstück bekommt immer die zur passende Stellung der Weiche benötigte Spannung.

Dies geschieht durch das Relais der Herzstückpolarisation. Wann die Umschaltung erfolgt spielt keine Rolle. Dies könnte schon vor dem eigentlichen Umschaltvorgang, bei Mittelstellung oder auch erst nach dem Umschalten erfolgen. Es darf nur keine Lok in dem Moment auf dem Herzstück stehen. Nur dann kommt es zum Kurzschluss, wenn die Polarisation nicht mit der Weichenstellung übereinstimmt.

Um die Weichenzungen sicher mit Spannung zu versorgen, nehme ich sie in die Schaltung mit auf und sie bekommen immer dann Spannung, wenn sie an der Stammschiene anliegen. Wie schon erwähnt, wird diese Spannung nicht polarisiert sondern nur an- bzw. abgeschaltet.

In Kürze folgt hier die Beschreibung meiner Platine mit Herzstückpolarisierung. Dies ist bereits fertiggestellt und befindet sich in der Testphase. Sie ist nicht identisch mit der Herzstückpolarisierung von Lintzmeyer/Häusler (siehe Kap. 13.7)

13.6.7 PM-Uni mit LDEC-Funktion

Mit dem neuen Universalperipheriemodul wird auch die bisherige LDEC Platine ersetzt. Bei der Anwendung ergeben sich zur bisherigen Funktionsweise keine Änderungen.

Wie schon im allgemeinen Teil der Beschreibung für das Universalmodul geschrieben habe, war es nicht möglich den LDEC Code mit in den Universalcode zu integrieren. Dies wäre nicht einmal ein Platzproblem im Programmcode gewesen, sondern durch die Anzahl der vielen Timeroutinen lassen sich die beiden Versionen nicht zusammenführen.

Um aber doch ein gewisses Universalcode Feeling zu bekommen, wurde auch in die LDEC-Version eine Bootladerfunktion integriert. Damit ist es relativ einfach, diese beiden Versionen zu tauschen. Einzelheiten zur Bootladerfunktion sind in der allgemeinen Beschreibung zum PM Uni (siehe Kap. 13.6.4) nachzulesen.

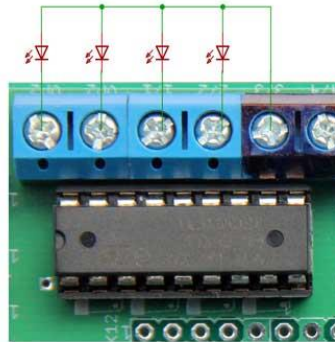
Anschlussbelegung beim Einsatz als LDEC

Im folgenden Bild ist die Belegung der Anschlüsse bei Verwendung des PM Uni als LDEC zu sehen. Die Reihenfolge der Anschlüsse ist dabei nicht fortlaufend. Die Anschlüsse A11 und A12 kommen bereits nach Anschluss A6.

Außerdem weicht der Bestückungsaufdruck auf der ersten Platinenserie bei Anschluss A1 bis VCC 2 von der Realität ab.



Bei der Bestückung mit ULN2803 als Treiberbaustein, lassen sich nur Lichtsignale mit gemeinsamer Anode anschließen. Die Spannungsversorgung kann man - wie schon bei der Dekoderfunktion vom PM Uni beschrieben - intern erzeugen oder extern einspeisen. Im nachfolgenden Bild ist ein Funktionsblock mit angeschlossenen LED dargestellt.

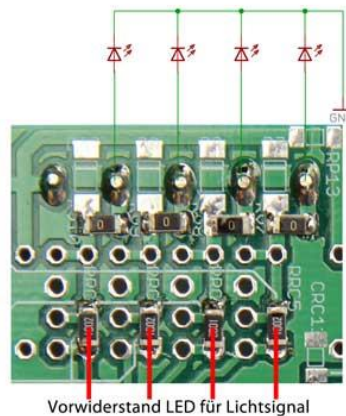


Möchte man Lichtsignale mit gemeinsamer Kathode verwenden, so kann man dies mit einem kleinen Trick auch verwirklichen.

Die LED werden dann direkt vom Prozessor gespeist. Alle Bausteine für die Spannungsversorgung der Verbraucher und die ULN2803 werden auf der Platine nicht bestückt. Man kann also eine Platine mit Minimalbestückung nehmen und diese durch 2 Widerstände oder 1 Widerstand und eine Drahtbrücke zum LDEC mit gemeinsamer Kathode umfunktionieren.

Im nachstehenden Bild sind die beiden Widerstände gut zu erkennen. Zum einen der 0 Ohm Widerstand als Verbindung zur Anschlussklemme und zum anderen der Vorwiderstand der LED als Verbindung zum Prozessor.

Bei dieser Schaltung werden die LED von der Stromversorgung des Prozessors mit gespeist. Da dieser seine Spannung über die RJ45 Buchse bezieht, sollte man nur Lichtsignale mit Low Current LED verwenden (ca. 2mA Stromaufnahme, 1,6kOhm Widerstand).



Der Atmega8 ist zwar auch in der Lage normale LED anzusteuern, die Gesamtbelastung wäre aber bei 12 LED zu hoch. Genauere Informationen dazu findet man im Datenblatt des Atmega8 unter "Electrical Characteristics".

13.7 Beschreibung des Servodecoders mit Herzstückpolarisierung

Von Günter Lintzmeyer und Jörg-Bodo Häusler wurde der Servodecoder um eine Herzstückpolarisierung erweitert. Sie ist nicht identisch mit der Herzstückpolarisation von Uwe Mäusezahl (siehe Kap. 13.6). Nachstehend die Beschreibung der einzelnen Varianten (Stand: Version 1.1 von 2011-04-04)

13.7.1 Allgemeine Hinweise

Die hier beschriebene Schaltung beruht auf der ModellbahnSelbstbauSteuerung (MoBaSbS) von Patrick Urban und ist auch nur mit selbiger verwendbar. Ergänzend zu Patricks´ Servodecoderplatine SDEC4 kann die Adapterplatine zur Polarisierung der Weichenherzstücke eingesetzt werden. Der hier verwendete Code wurde für die MobaSbS mit Softwarestand Version 4 entwickelt. Aufbauend auf dieser Schaltung und Software sind 2 zusätzliche Hardwarevarianten entstanden, wo zusätzlich jede einzelne Weichenzunge geschaltet wird. Software-Entwurf und Realisierung kommt von: Günter Lintzmeyer, Hardware-Entwurf und Realisierung kommt von: Jörg-Bodo Häusler.

13.7.2 Warum diese Schaltung?

Es gibt im Modellbahnhobby 2 relevante Arten der Stromzuführung. Alle Modellbahner mit einem Mittelleiter (z.B. Märklin) werden diese Schaltung nicht brauchen und können diese Beschreibung entspannt, mehr informativ, lesen.

Wer keinen Mittelleiter hat und die Stromzuführung über die Gleise macht, ist ein 2 Leiterfahrer und wird diese Schaltung benötigen, je mehr seine Weichen sich dem vorbildhaftem Aussehen annähern sollen. Er wird diese Schaltung auch benötigen, wenn er die Betriebssicherheit im Weichenbereich erhöhen will.

Die folgenden Erklärungen beziehen sich hier also nur auf die 2 Leiterfahrer.

Auch wenn die Digitaltechnik bei unserem Hobby Einzug gehalten hat, so werden unsere Loks auch weiterhin ihren Strom über die Schiene abgreifen. Es gibt im Weichenbereich eine Stelle, Herzstück genannt, wo es zu Stromunterbrechungen kommen kann (siehe Bild1). Dies resultiert daraus, weil dort das linke und rechte Gleis zusammengeführt werden. Das Herzstück muss isoliert werden damit es zu keinem Kurzschluss kommt. Entweder wird ein Herzstück aus Plastik verwendet, oder das linke und rechte Gleis werden isoliert.

Eine weitere Problemstelle ist der Bereich der Weichenzungen. Bei einer Weiche liegt immer eine Weichenzunge an einer Schiene an und hat daher die gleiche Polarität wie Diese. Die gegenüberliegende Weichenzunge hat aber auch dieselbe Polarität und steht mit einem gewissem Abstand der anderen Schiene gegenüber. Diese Schiene hat aber die entgegengesetzte Polarität. Zwischen dieser Schiene und der Weichenzunge kann es zu einem Kurzschluss kommen, wenn ein Rad durch unrundem Lauf oder schlackern beide elektrisch kurzzeitig verbindet.

Es gibt bei der Weichenzunge aber auch die Möglichkeit, dass diese keinen Strom führt, weil der Anpressdruck der Zunge an die Schiene nicht für sicheren elektrischen Kontakt ausreicht. Sei es durch Verschmutzung oder Materialermüdung. Um diese Problemstellen zu beheben, haben sich bei den Herstellern der Weichen verschiedene Lösungen etabliert. Ich will hier nicht alle Industrielösungen auflisten, aber das Ergebnis ist immer dasselbe. Wir müssen uns entscheiden, zwischen einer betriebsicheren Weiche oder einer originalgetreuen Weiche, mit allen Zwischenlösungen die es gibt.

Mit dieser Schaltung wollen wir es allen 2Leiterfahrern ermöglichen mit einer originalgetreuen Weiche eine hohe Betriebssicherheit zu erlangen. Das Ganze natürlich wie gewohnt im Selbstbau.

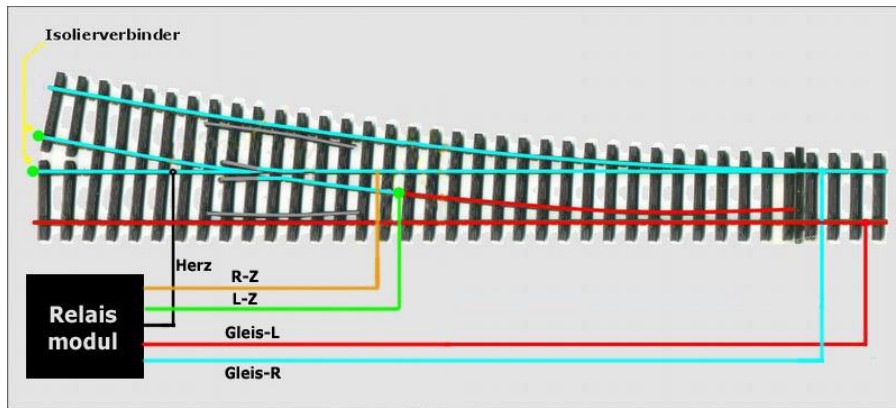
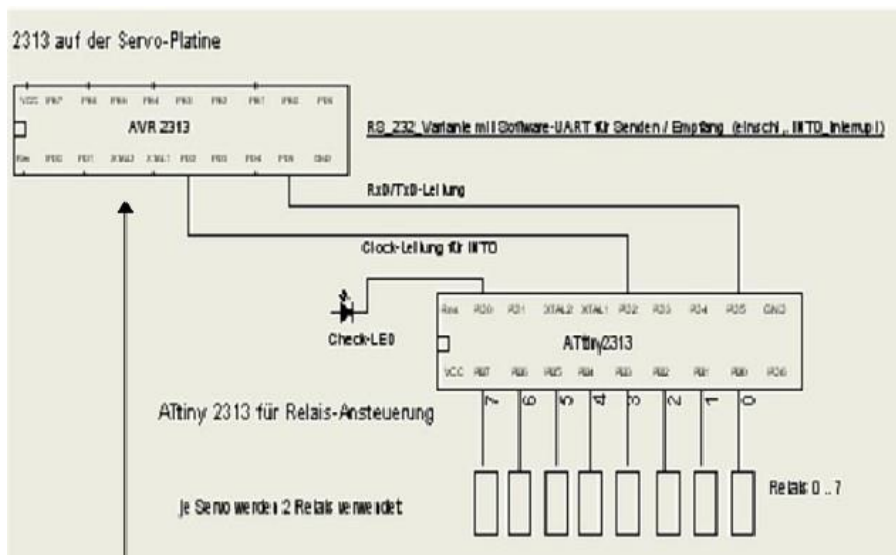


Bild 1

13.7.3 Funktionsbeschreibung

Herzstückpolarisation

Ist die Modellbahn abgeschaltet sind alle Kontakte offen. Das Herzstück bekommt keinen Strom. Grundstellung. Dies ist wichtig, da beim Einschalten der Steuerung alle Servomotoren in einen undefinierten Zustand stehen. Deshalb bekommen sie dann erst alle einen Stellimpuls von RR&Co.



Für die Herzstückpolarisation benötigen wir jeweils 2 Relais. Das Blockschaltbild zeigt die Datenverbindungen zwischen der Servo- und Adapterplatine.

Die Funktionsweise ist bei allen vorgestellten Hardwarevarianten dieselbe. Deshalb wird auch bei allen Varianten die gleiche Software verwendet.

Schritt 1

Unmittelbar nach dem Eintreffen eines Schaltkommandos (0x52-Adr-SchaltInfo-xor) wird das Herzstück der betreffenden Weiche stromlos geschaltet und der Umschaltvorgang für das Herzstück der Weiche (Umpolarisierung) vorbereitet (aber noch nicht geschaltet). Die Weichenzungen sind dann ebenfalls stromlos.

Schritt 2

In Patrick's Original-Code wird jetzt die Durchführung des Servo-Laufes vorbereitet und durchgeführt. Die Stelldauer (Stellgeschwindigkeit) der Weiche ist in Patrick's Code einstellbar und gilt somit hier auch. Diese Einstellung kann wie gewohnt mit dem Konfigtool bearbeitet werden.

Schritt 3

Nach einer - je Weiche individuell einstellbaren - Verzögerungszeit wird die Umpolarisierung aktiviert (relais_rs232.eep).

Die Verzögerungszeit sollte mindestens die Stellzeit der Servo's, plus eine Sicherheitszeit beinhalten. Die Verzögerungszeit kann in Abständen von 200 ms bis 3 Sek. eingestellt werden (in 200ms-Stufen, typische Einstellzeit bei den bisherigen Anwendern 800 ms).

Diese Zeit wird hinterlegt im EEPROM des ATtiny2313 auf der Adapter-Platine. Bei den anderen Varianten ist dies der Attiny2313 ohne ext. Quarzbeschaltung.

Hardware

Die Hardware hat einige zusätzliche Bauteile, die ausschließlich für eine höhere Ausfallsicherheit vorhanden sind. Diese Maßnahmen dienen auch gleichzeitig dafür, die Elektromagnetische Verträglichkeit (EMV) der Schaltung zu verbessern. Folgende Maßnahmen werden angewendet:

- 1) Getrennte Stromversorgung für die Ansteuerung und den Relaiseteil. Entkopplung durch die Stabilisierungs IC's.
- 2) Getrennte sternförmige Stromversorgung.
- 3) Breite Leiterbahnen für die pos. Stromleitungen und große Masseflächen.
- 4) Resetschaltung direkt am Prozessoranschluß.
- 5) Abblockkondensatoren an den Ic's.
- 6) Entstörkondensatoren an den Relaiskontakten (auch zum Entprellen).
- 7) Zusätzliche Freilaufdioden direkt am Relais (sind auch im ULN2803 integriert)
- 8) Entstörkondensatoren direkt an der Relaispule.

Außerdem sind alle Herzstücke mit Polymersicherungen versehen. Diese selbsttätig rückstellende Sicherung ist für einen Dauerstrom von 0,4 A (andere Ströme sind auch möglich) ausgelegt. Bei einem Kurzschluß wird das Bauteil hochohmig und unterbricht damit den Strom. Sobald der Kurzschluß beseitigt ist, kühlt sich das Bauteil ab und wird wieder leitend.

13.7.4 Die Software

Eine kurze Erklärung für alle Programmierer unter uns.

Anpassungen im Original-Code des Servodecoder (PM_SDEC4)

die beiden Programme für die Servodecoder
PM_SDEC4 und
PM_SDEC4_AT90S2313
wurden jeweils an 3 Stellen angepasst:

- Erweiterung der Port's (PORTD.x) für die RS232- und die CLOCK-Leitung (INT0)
- Aufruf der Programm-Routine > SD0_command_901 < (rcall ...)
- Einbindung des Programmes > SD0_command_asm < (.include ...)

In der Programm-Routine >SD0_command_901< werden die Daten mittels SW-UART gesendet und die CLOCK-Leitung aktiviert.

Die bereitgestellten Programmfiles können auch bei >Nichtverwendung< der Herzstückpolarisierung für die >normalen< Servodecoder verwendet werden.

Die geänderten Programme werden als
PM_SDEC4_AT90S2313_relais_rs232
PM_SDEC4_ATtiny2313_relais_rs232
als HEX- und EEP-File bereitgestellt .

Adressen und Parameter-Anpassungen können wie bisher über das Konfigtool gemacht werden.

Der Code für die Relaissteuerung.

Dieses Programm wurde neu erstellt.

Beim Programm-RESET werden die eingetragenen Verzögerungswerte aus dem EEPROM gelesen und einer Plausibilitäts-Prüfung unterzogen.

Sind fehlerhafte Werte vorhanden, blinkt die CHECK-LED. Die vorgesehenen Programm-Routinen werden nicht aktiviert (Programm-STOP)

Nach einem ext. INT0-Interrupt (auf der CLOCK-Leitung) werden die dann auf dem Datenbus' bereitgestellten Informationen mittels SW-Uart gelesen und die entsprechenden Port's für das Ansteuern der Relais vorbereitet.

Nach Ablauf der eingestellten Verzögerung (aus EEPROM gelesen) werden die Port's aktiviert und die Relais geschaltet.

Nach dem Schalten (also nach Ablauf der Verzögerungszeit) wird im eeprom das gerade aktivierte Relais (einer Weiche) gespeichert.

Beim RESET bzw. Einschalten des tiny2313 wird pro Weiche das zuletzt geschaltete Relais ermittelt (aus dem eeprom) und das zugehörige Relais aktiviert (der entsprechende Port wird eingeschaltet).

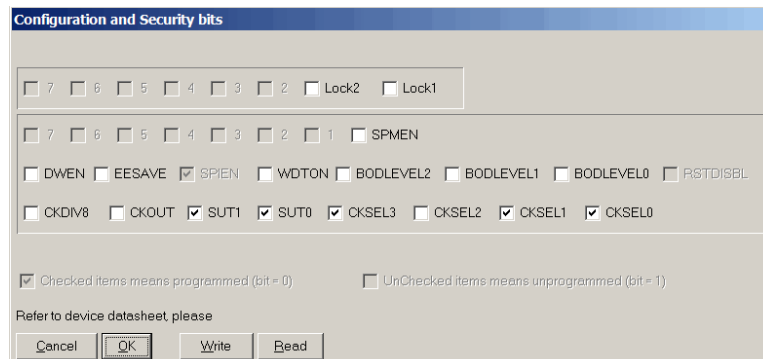
Im Programm-Code > relais.eep < können die entsprechenden Verzögerungswerte (0x02 = 200 ms , 0x04 = 400 ms ...0x1E = 3000 ms) eingestellt werden.

Diese evtl. per Ponyprog abändern!

In der bereitgestellten Version sind für alle 4 Servo 800 ms eingestellt.

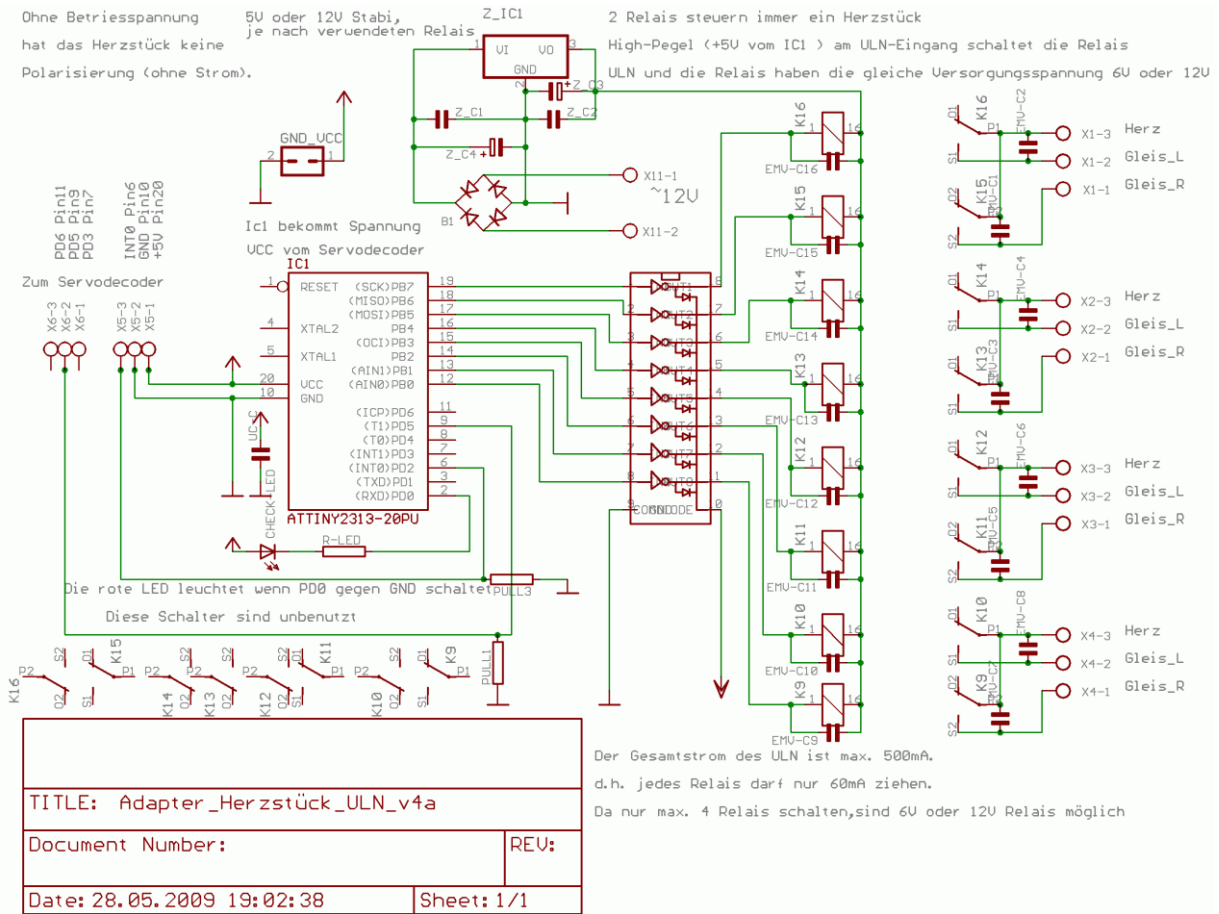
Das neue Programm wird als relais_rs232_reset (HEX- und EEP-File) bereitgestellt.

Beim ATTiny2313 gibt es auch Einstellungen für die Fusebits. Diese so belassen, wie sie im Auslieferungszustand sind. Damit wird dann der interne Takt verwendet und es ist kein ext. Quarz notwendig. Das folgende Bild zeigt die Fusebits, wie sie in Ponyprog angezeigt werden.

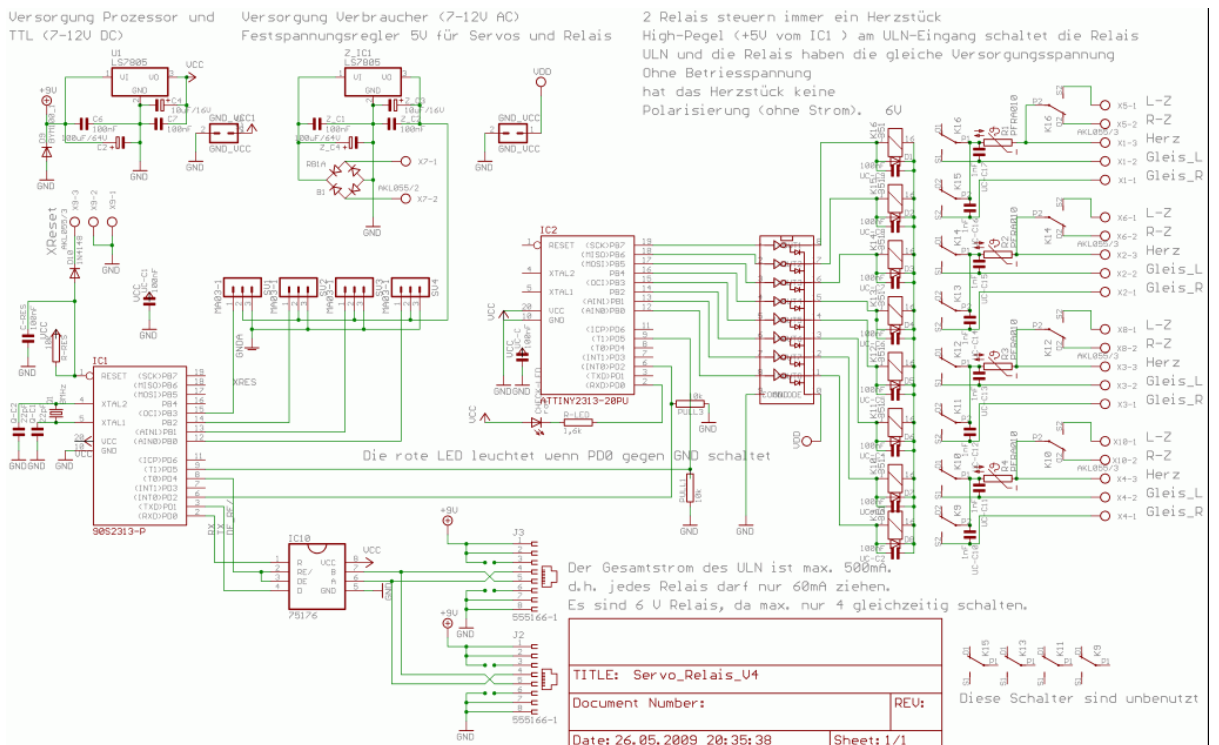


13.7.5 Schaltbild

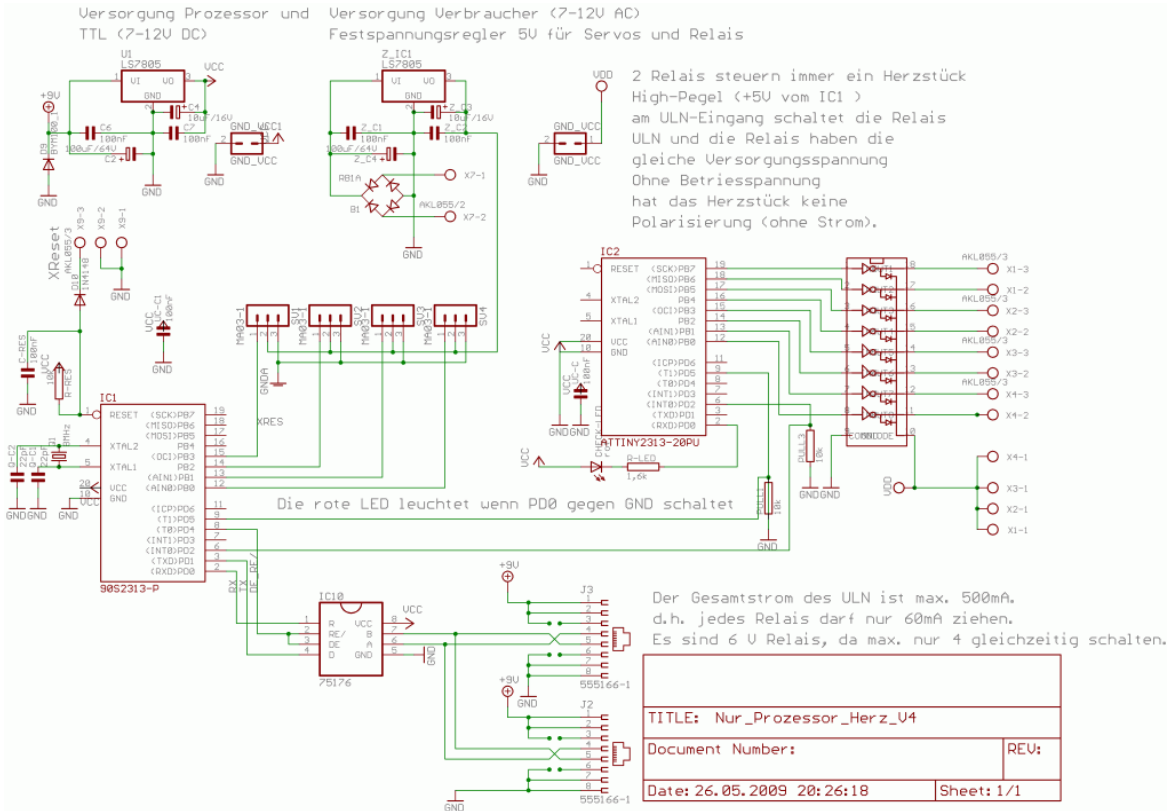
Variante1. Adapterplatte für einen Servodecoder SDEC4



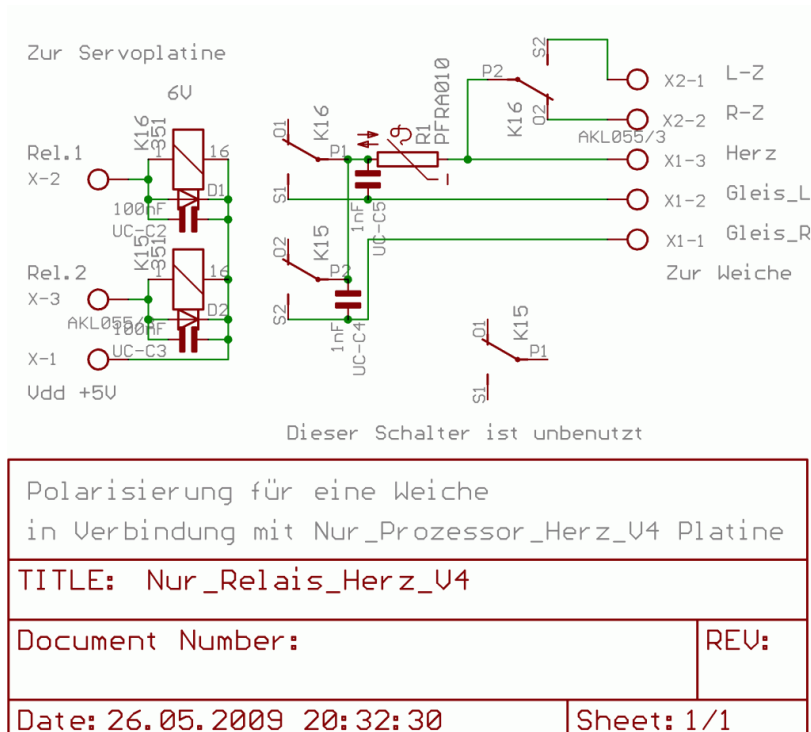
Variante2. Servodecoder mit Herzstück- und Weichenzungenpolarisierung



Variante3. Ansteuerung und Relaiseteil sind getrennt



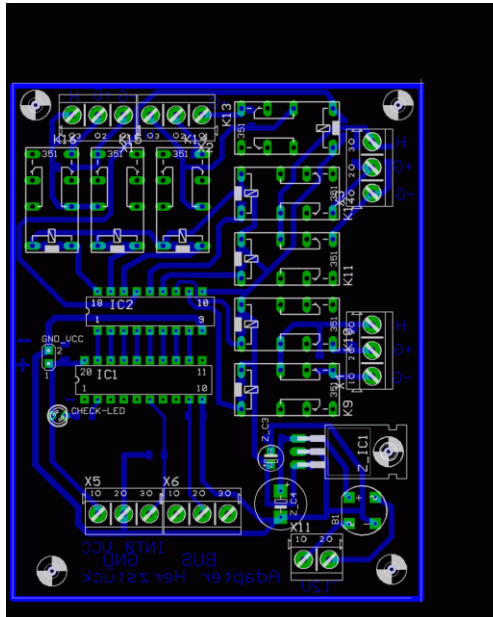
zu der getrennten Variante , hier der Relaiseteil.



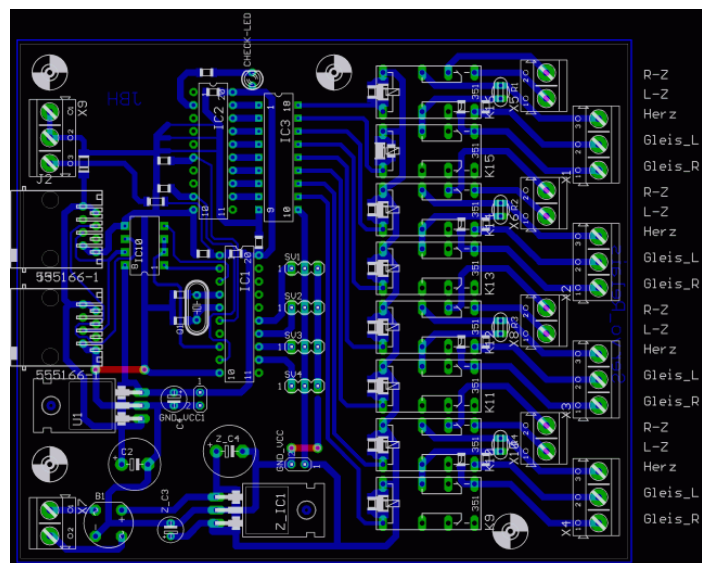
13.7.6 Hardware und Platinen-Layouts

Zur besseren Übersicht alle Platinen hier ohne Masseflächen dargestellt.

Variante1. Adapterplatine für einen Servodecoder SDEC4

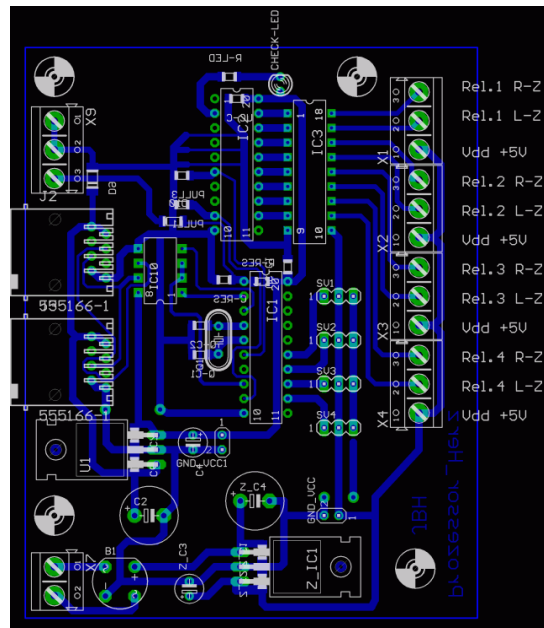


Variante2. Servodecoder mit Herzstück- und Weichenzungenpolarisierung

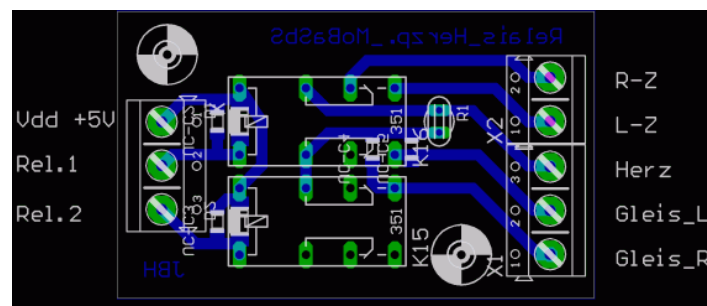


Achtung: GND zu den 4 Servoanschlüssen ist aus Routing- und Optionsgründen nicht angeschlossen. Bitte mit einer Drahtbrücke verbinden.

Variante3. Ansteuerung und Relaiseteil sind getrennt



und hier der Relaiseteil zu Variante 3:



Aufbauhinweise

- Allgemein
Zum Aufbau gibt es nicht viel zu sagen. Ich verweise hier auf die Internetseiten der MoBaSbS und den dortigen Hinweisen zum Aufbau des Servodecoder und den allgemeinen Hinweisen zum Aufbau und Inbetriebnahme von Schaltungen.
- Adapterplatine
Die Adapterplatine wurde für bereits aufgebaute Servodecoder entworfen. Es wurde darauf geachtet, dass nicht unter der Anlage gelötet und keine zusätzlichen Leitungen gelegt werden müssen. Die Relais bekommen eine ext. Stromversorgung. Diese kann Gleichstrom oder Wechselstrom sein. Je nach verwendeten Relais Typen muss die eingespeiste Spannung 5V höher sein. Dies kann z.B. dieselbe Versorgungsspannung für den Leistungsteil der Servoplatine sein. Der ATtiny2313 auf der Adapterplatine bekommt seine Daten und Versorgungsspannung direkt von der Servoplatine. Dazu kann ein IC-Sockel verwendet werden. Einfach die Leitungen an den entsprechenden Stellen anlöten und diesen Sockel auf den Sockel der Servoplatine dazwischen stecken. Der Prozessor ATtiny2313 wird mit den Files relais_rs232_reset programmiert.
- Servodecoder mit Herzstückpolarisierung
Dieser Servodecodertyp besteht im Prinzip aus 2 Schaltungen. Schaltung1 ist die Originalschaltung des Servodecoder SDEC4 und Teil2 ist die Adapterplatine für die Herzstückpolarisierung. Beide Schaltungen wurden auf einer Platine vereint, damit wir uns die Verbindungsleitungen dazwischen einsparen können. Denn wer mit dem Einsatz von Servodecodern anfängt, kann gleich diese Variante nutzen. Hier können nur die 6V -Relais

verwendet werden. Der Grund ist die gleiche Versorgungsspannung für die Servos und die Relais. Die verwendeten Servos haben alle 5V.

Der Prozessor ATTiny2313 (IC2) wird mit dem Files relais_rs232_reset programmiert.

Der Prozessor ATTiny2313 (IC1 mit 8Mhz Quarz) wird mit den Files PM_SDEC4_ATTiny2313_relais_rs232 programmiert.

Wer den Vorgängerprozessor AT90S2313 einsetzt, muss die Files PM_SDEC4_AT90S2313_relais_rs232 verwenden.

- Ansteuerung und Relaiseteil sind getrennt

Um beim Aufbau mehr Flexibilität zu bekommen, wurde hier eine Trennung vorgenommen. Dadurch ist es möglich, auch einzelne Weichen an entfernten Stellen der Modellbahn mit weniger Kabelaufwand zu polarisieren. Hier können nur die 6V - Relais verwendet werden. Der Grund ist die gleiche Versorgungsspannung für die Servos und die Relais. Die verwendeten Servos haben alle 5V.

Der Prozessor ATTiny2313 (IC2) wird mit dem Files relais_rs232_reset programmiert

Der Prozessor ATTiny2313 (IC1 mit 8Mhz Quarz) wird mit den Files PM_SDEC4_ATTiny2313_relais_rs232 programmiert

Wer den Vorgängerprozessor AT90S2313 einsetzt, muss die Files PM_SDEC4_AT90S2313_relais_rs232 verwenden.

13.7.7 Hard- und Software-Bereitstellung (Download)

Die Hard- und Software wird im MoBaSbS Anwenderforum im Ordner Dateien als ZIPFile bereitgestellt. Zu finden unter: Alles von Bodo / Polarisierung

13.7.8 Materialliste

Elektronikbauteile gibt es bei vielen Händlern online. Es wurden nur leicht beschaffbare Bauteile verwendet. Schaltbild und Layouts wurden mit Eagle 4.11 erstellt. Es lassen sich damit auch Bauteillisten erstellen. Das CAD-Programm Eagle gibt es als käufliche Variante und als Demoversion. Mit der Demoversion können diese Layouts gedruckt werden.

Mein Online-Händler ist Reichelt und für ein paar Bauteile will ich hier die Bestellnr. auflisten. Für einen Nachbauer, der den LötKolben an der richtigen Seite anfasst stellt es sicher kein Problem dar, sich die Liste selbst zu erstellen.

- | | | |
|---------------------------|-----|--|
| • rote LED | LED | 3MM 2MA RT |
| • rückstellende Sicherung | | PFRA 040 |
| • SMD Widerstand | xxx | SMD 1/4W xxx (xxx durch Wert ersetzen) |
| • Relais Treiber | | ULN 2803A |
| • SMD Keramikkondens | | NPO-G1206 33P (33p für Quarzbeschaltung) |
| • Prozessor | | ATTINY2313 DIP |
| • Sockel 20 polig | | GS 20P |
| • Modular Einbaubuchse | | MEBP 8-8S |
| • Spannungsregler 5V | | µA 7805 |
| • Anreihklemme 3pol. | | AKL 073-03 |
| • Fotoplatine einseitig | | BEL 160X100-1 |
| • Relais 6V | | FIN 30.22.9 6V |
| • Gleichrichter | | B80C1500RUND |
| • Quarz 8Mhz | | 8-HC49U-S |
| • 100 nF Kondensator | | X7R-G1206 100N |

Fragen zum Nachbau, Anwendung oder Anregungen zu dieser Anleitung, bitte im Anwenderforum der MoBaSbS stellen. Wir sind dort regelmäßig virtuell anzutreffen. Wir wünschen allen 2Leiterfahrern Erfolg beim Nachbau und viel Spaß mit unserem gemeinsamen Hobby, der Modellbahn und Ihrer super Steuerung MoBaSbS.