



MBCAN

Fernsteuerung einer  - Modelleisenbahn

Nicht-kommerzielles Projekt – Alle Angaben ohne Gewähr

Bedienungsanleitung

Switchboard mbc-81

Version 1.2

HW 24.01.06, FW 1.00, Parametriercenter ab 2.1.0.0

©2007 – 2024 by Dr.-Ing. Thomas Wiesner

1 Inhalt

2	Disclaimer	4
3	Revision	5
3.1	Bedienungsanleitung	5
3.2	Firmware	5
4	Einleitung.....	6
5	Funktion.....	7
6	Schaltbild	8
7	Bestückung	10
8	Bauteileliste.....	11
9	Firmware	12
10	Steckverbindungen.....	13
11	Anschlussbeispiel.....	15
12	Inbetriebnahme.....	16
12.1	Modul in Betriebsbereitschaft versetzen.....	16
12.2	Modul konfigurieren	16
12.3	Modul mit der CS3 [®] verbinden.....	19
13	Modulbilder	22
14	Systemarray-Belegung für Eigenentwicklungen	23
14.1	Allgemeiner Bereich zum Modul.....	23
14.2	Modulspezifischer Bereich für Funktionsparameter	27
15	Befehlssatz zu den Modulen	28
15.1	PC_DH_H - Übertragung des Datenbanknamens mit 12 Bytes (1-4).....	30
15.2	PC_DH_M - Übertragung des Datenbanknamens mit 12 Bytes (5-8).....	31
15.3	PC_DH_L - Übertragung des Datenbanknamens mit 12 Bytes (9-12).....	32
15.4	PC_KENNER - Kenner und Identifier für die Module	33
15.5	PC_NEU - Neuanmeldeaufforderung	34
15.6	PC_NEU_DATA - Rückmeldung PC an das Modul während des Neuanmeldeprozesses	35
15.7	MD_NEU_DATA - Meldung des Moduls während des Neuanmeldeprozesses	36
15.8	PC_RESET - Durchführen eines Hardware-Resets auf dem Modul	37
15.9	PC_MD_SEL - Modul aus Datenbank entfernen	38
15.10	PC_ALIVE - ALIVE-Abfrage.....	39



15.11	MD_ALIVE - ALIVE-Abfrage.....	40
15.12	PC_ARRAY - Zugriff Systemarray anfragen	41
15.13	MD_ARRAY - Zugriff Systemarray freigegeben.....	42
15.14	PC_ARRAY_DATA - Zugriff Systemarray freigegeben	43
15.15	MD_ARRAY_DATA - Antwort des Moduls auf Systemarray-Zugriff	44
15.16	PC_UPGRADE - Firmware-Upgrade	45
15.17	MD_UPGRADE - Firmware-Upgrade freigeben	46
15.18	PC_UPGRADE_DATA - Schreibe Firmware.....	47
15.19	MD_UPGRADE_DATA - Antwort des Moduls auf Schreibe Firmware	48
15.20	PC_BOOT - Modul neu Booten	49
15.21	MD_S88 - Stellungsmeldung mbc-88 / mbc-90.....	50
16	Post-Code	51
17	Quellenverzeichnis	53
18	Allgemeine Hinweise zum MBCAN-Projekt.....	54

2 Disclaimer

ACHTUNG: Nur für erfahrene Elektronikbastler geeignet. KEIN Kinderspielzeug!

Bei Arbeiten an oder mit der aus dieser Dokumentation erstellten Leiterplatte beachten Sie bitte:

- Der Betrieb ist nur an Spannungen kleiner 24 V DC erlaubt. Verwenden Sie ausschließlich geprüfte und zugelassene Steckernetzteile
- Zusammenbau oder Instandsetzungen/Änderungen an der Leiterplatte sind immer im spannungsfreien Zustand durchzuführen
- Betreiben Sie das Gerät nur in trockenen Räumen. Beim Einsatz im Freien sollten Sie entsprechende Maßnahmen zum Schutz gegen Feuchtigkeit ergreifen
- Die zulässigen Ströme an den Schaltausgängen sind einzuhalten. Details finden Sie im jeweiligen Kapitel zur Funktion (vgl. Kapitel 5)
- Dieses Produkt ist nicht für die Nutzung durch Kinder unter 14 Jahren geeignet. Die Anforderungen an Kinderspielzeug werden NICHT erfüllt

Bitte beachten Sie außerdem das Kapitel „Allgemeine Hinweise zum MBCAN-Projekt“ bevor Sie mit dem Nachbau oder der Anwendung der Informationen für eigene Entwicklungen beginnen.

3 Revision

3.1 Bedienungsanleitung

1.0	06.01.2024	Neue Hardwareversion
1.1	01.10.2024	Redaktionelle Anpassungen
1.2	20.10.2024	Redaktionelle Anpassungen Befehlssatz

3.2 Firmware

FW: 1.00 - Erste Version

4 Einleitung

"Machine-to-Machine (M2M) steht für den automatisierten Informationsaustausch zwischen Endgeräten wie Maschinen, Automaten, Fahrzeugen oder Containern untereinander oder mit einer zentralen Leitstelle, zunehmend unter Nutzung des Internets und den verschiedenen Zugangsnetzen, wie dem Mobilfunknetz. Eine Anwendung ist die Fernüberwachung, -kontrolle und -wartung von Maschinen, Anlagen und Systemen, die traditionell als Telemetrie bezeichnet wird. Die M2M-Technologie verknüpft dabei Informations- und Kommunikationstechnik."

[Wikipedia, https://de.wikipedia.org/wiki/Machine_to_Machine]

Was für professionelle Systeme gilt, kann für die Automatisierung einer Modelleisenbahn nicht schlecht sein. Auch hier haben wir eine Leitstelle (bei Märklin® die CS2/3® oder MS2®) und verteilte Komponenten, die über den CAN-Bus verbunden sind. Auf dem CAN-Bus finden wir ein von Märklin® definiertes Protokoll vor. Der Austausch von Informationen erfolgt dann automatisch, wobei es keine reine Master-/Slave-Struktur auf dem Bus gibt, sondern ein Multi-Master-System. Das bedeutet, dass sich die mbc-Module bei Änderungen im Prozess, z.B. beim manuellen Umstellen der Weiche, selbständig bei der Leitstelle melden. Gleiches gilt für die Rückmelder.

Im Folgenden finden Sie hier die Beschreibung des Switchboards für die Erstellung eines Gleisbildstellpults.

5 Funktion

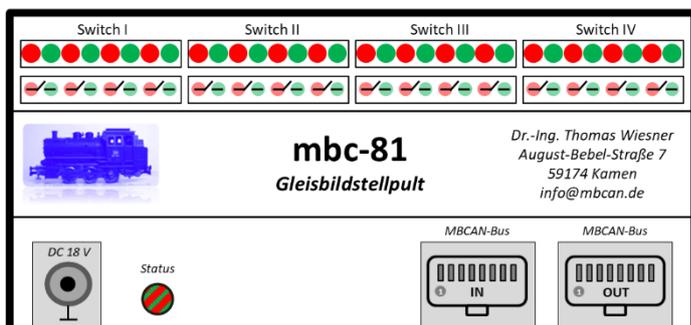


Abbildung 5-1:Modullabel

Dieses Modul stellt die Verbindung zwischen einem herkömmlichen Gleisbildstellpult und der MBCAN-Welt her.

Es ersetzt/doppelt im CS2/3[®]-/MS2[®]-Betrieb die Keyboard-Funktion da die Schaltkommandos im CAN-Nachrichtenformat von Märklin[®] übertragen werden.

Bei Märklin[®] Digital der 1. Generation oder anderer Hersteller wird ein herkömmliches Stellpult-/Keyboard-Modul ersetzt. 32 LED und 32 Taster können angeschlossen werden. Die LED dienen dabei als Stellungsrückmeldung und werden vom entsprechenden mbc-Modul (83, 84, und/oder 91) gemeldet.

In der Bus-0-Funktionalität stellt das mbc-81 ein Halbmodul zur Tastenmatrix am Links88[®] zur Verfügung.

Steckverbinder	2x RJ45 MBCAN-Bus / LED-Anzeigen und Schalter über PINHEAD-Verbinder
Stromversorgung	Steckernetzteil / RJ45-Versorgung
Statusanzeige	Betriebszustand und Traffic wird über Dreifarb-LED angezeigt
Adresszuordnung	Adresse per PC-Software einstellbar
Adressformat	Märklin [®] Motorola über Gleissignal / CAN-Bus-Protokoll (Loc-ID 0x00003000 - 0x000033ff) möglich
Features	Bei Anschluss an CS2/3 [®] resp. MS2 [®] automatische Rückmeldung / Parametrierung, Firmwareupdate und Auslesung über PC möglich / Anzeige des Moduls in der GUI der CS2/3 [®]

Das in Abbildung 6-1 gezeigte Schaltbild zeigt den bei allen Modulen identischen Prozessor-Kern mit Thermosensor DS 1820 als Seriennummer-Lieferant, einem externen EEPROM für die Upgrade-Fähigkeit und dem CAN-Bus-Interface nebst automatischer Terminierung des CAN-Bus.

Die Sonderfunktionen des Switchboards bestehen aus den Steckverbindungen für Taster und LEDs. Über einen I2C-Baustein PCF 8574 werden die I/O-Pins des eingesetzten Controllers erweitert. Bei den LEDs wird die Spannungsteilerregel angewandt. D.h. je nach Pin-Level ist nur eine von beiden LEDs aktiv, im inaktiven Zustand (z-Zustand) leuchten beide LEDs mit halber Helligkeit.

Wenn die Module als Tastenmatrix für die Emulation eines LinkS88® genutzt werden soll, haben die LED keine Funktion.

7 Bestückung

Die Bestückung erfolgt wie üblich von den Bauteilen mit der geringsten Höhe (z.B. Widerstände) bis hin zu den höchsten Bauteilen (z.B. Stecker).

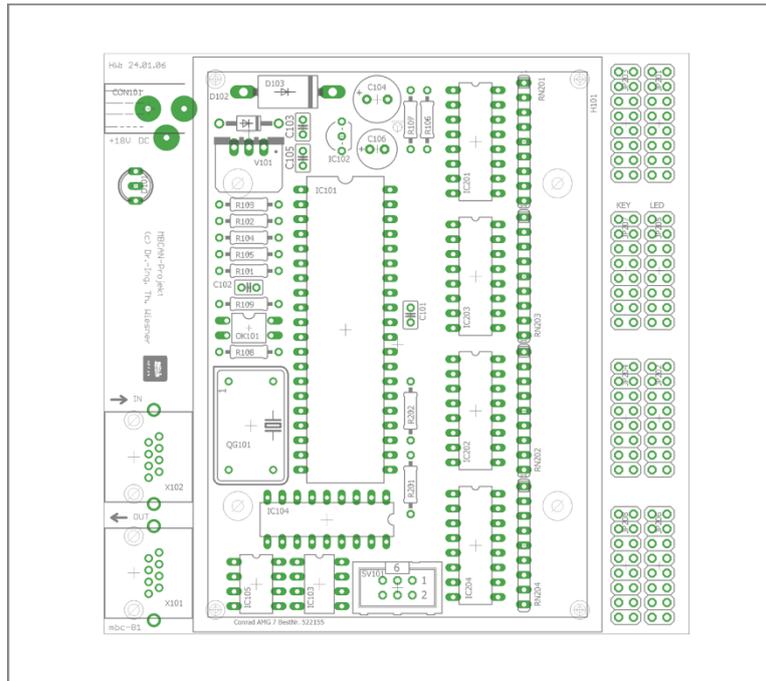


Abbildung 7-1: Bestückung Bauteilnummern

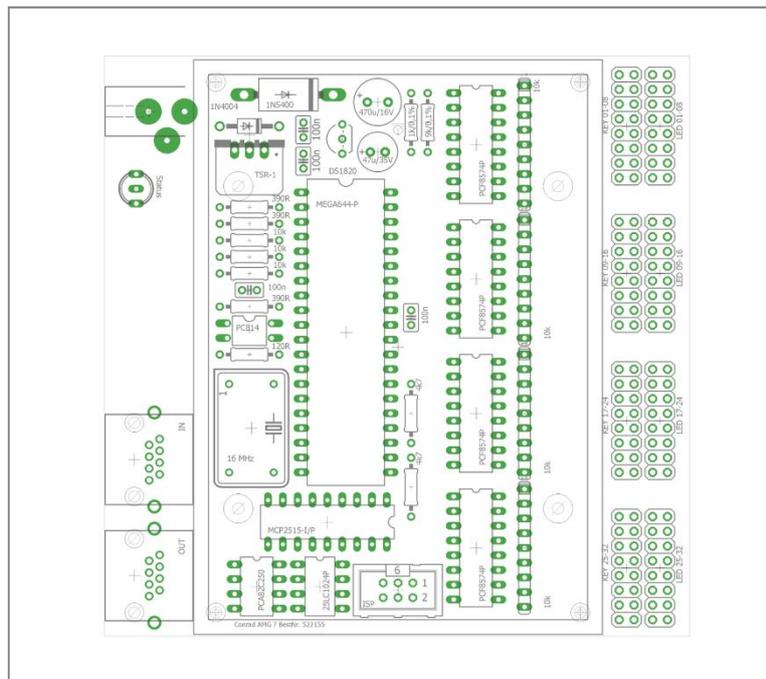


Abbildung 7-2: Bauteilwerte

8 Bauteileliste

Die für die Bestückung benötigten Bauteile sind in nachfolgender Tabelle aufgelistet. Ergänzt sind außerdem ein möglicher Lieferant sowie die zugehörige Bestellnummer. Der Lieferant ist nur ein Vorschlag und ist nicht bindend.

Das Gehäuse ist optional. Die Platine ist allerdings exakt auf das in der unteren Tabelle stehende bei Conrad erhältliche Gehäuse AMG 7 abgestimmt.

Tabelle 8-1: Stückliste

Part	Value	Lieferant	Bestellnummer	Anzahl
C101, C102	100n	Reichelt	MKS02-63 100N	2
C103	470u/16V	Reichelt	RAD 470/16	1
C104	47u/35V	Reichelt	RAD 47/35	1
D101	DUOLED R/G 5MM	Reichelt	LED 5 RG-3	1
D102, D104	1N4004	Reichelt	1N 4004	2
D103	1N5400	Reichelt	1N 5400	1
IC101	MEGA644-P	Reichelt	ATMEGA 644P-20PU	1
IC102	DS1820	Reichelt	DS 18S20	1
oder	DS1820	Reichelt	DS 18B20	1
IC103	25LC1024	Reichelt	25LC1024-I/P	1
IC104	MCP2515-I/P	Reichelt	MCP 2515-I/P	1
IC105	PCA82C250	Reichelt	MCP 2551-I/P	1
IC201-IC204	PCF8574P	Reichelt	PCF 8574 N	4
OK101	PC814	Reichelt	LTV 814	1
V101	TSR 1-2450	Reichelt	TSR 1-2450	1
QG101	QG5860	Reichelt	OSZI 16,000000	1
R101, R104, R105	10k	Reichelt	METALL 10,0K	3
R108	120R	Reichelt	METALL 120	1
R102, R103, R110	390R	Reichelt	METALL 390	3
R201, R202	4k7	Reichelt	METALL 4,70K	2
R106	9k/0,1%	Reichelt	MPR 9,10K	1
R107	1k/0,1%	Reichelt	MPR 1,10K	1
RN201-RN204	10k	Reichelt	SIL 9-8 10K	4
J101, J102	MBCAN	Reichelt	CAT5 T1U 2.8N4N	2
CON101	Power Jack	Reichelt	LUM NEB 21R	1
SV101	ISP	Reichelt	WSL 6G	1
JP201-JP208	PINHEAD	Reichelt	MPE 087-2-016	8
ICS-8pol	Socket 8	Reichelt	GS 8P	1
ICS-16pol	Socket 16	Reichelt	GS 16P	4
ICS-18pol	Socket 18	Reichelt	GS 18P	1
ICS-40pol	Socket 40	Reichelt	GS 40P	1
Platine	mbc_81.brd	PCBPOOL	mbc_81.brd	1

9 Firmware

Die Firmware zum Modul kann entweder direkt onboard via ISP-Schnittstelle oder extern auf den Controller gebracht werden (vgl. Bestückung).

Entsprechende Dateien können von der Webseite heruntergeladen werden. Die Dateitypen sind dabei wie folgt zu unterscheiden:

- Dateien des Typs **mbc_xx_xx_xx_xx_isp.hex** sind für die Erstprogrammierung zu verwenden und über die ISP-Schnittstelle aufzuspielen
- Dateien des Typs **mbc_xx_xx_xx_xx_upgrade.hex** sind für das Upgrade über das Parametriercenter gedacht. Sie funktionieren NICHT bei der Programmierung über die ISP-Schnittstelle

Die korrekte Einstellung der FUSES ist Abbildung 9-1 zu entnehmen, wenn das AVR Studio verwendet wird.

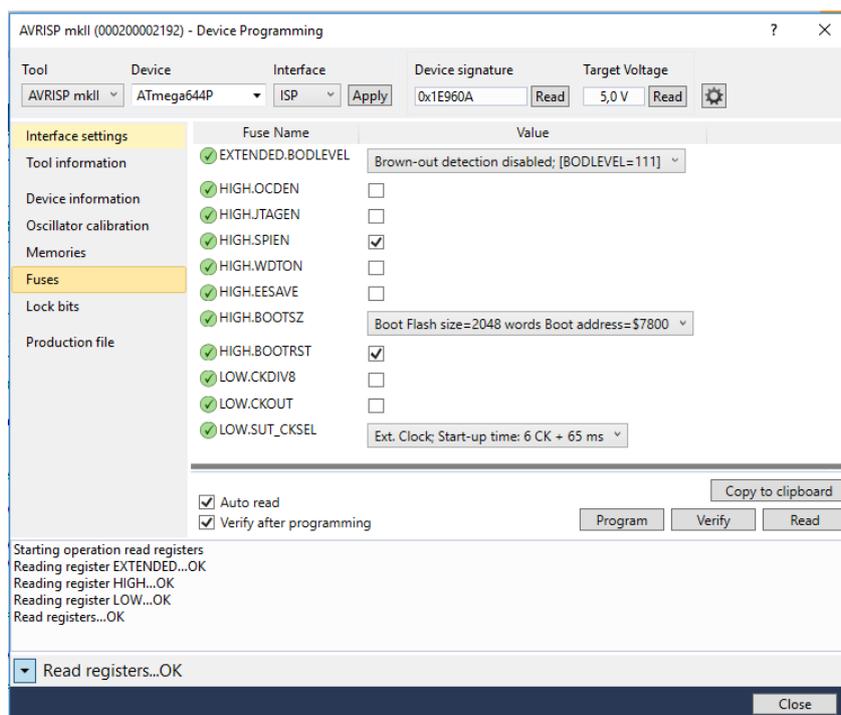


Abbildung 9-1: FUSES im AVR Studio

10 Steckverbindungen

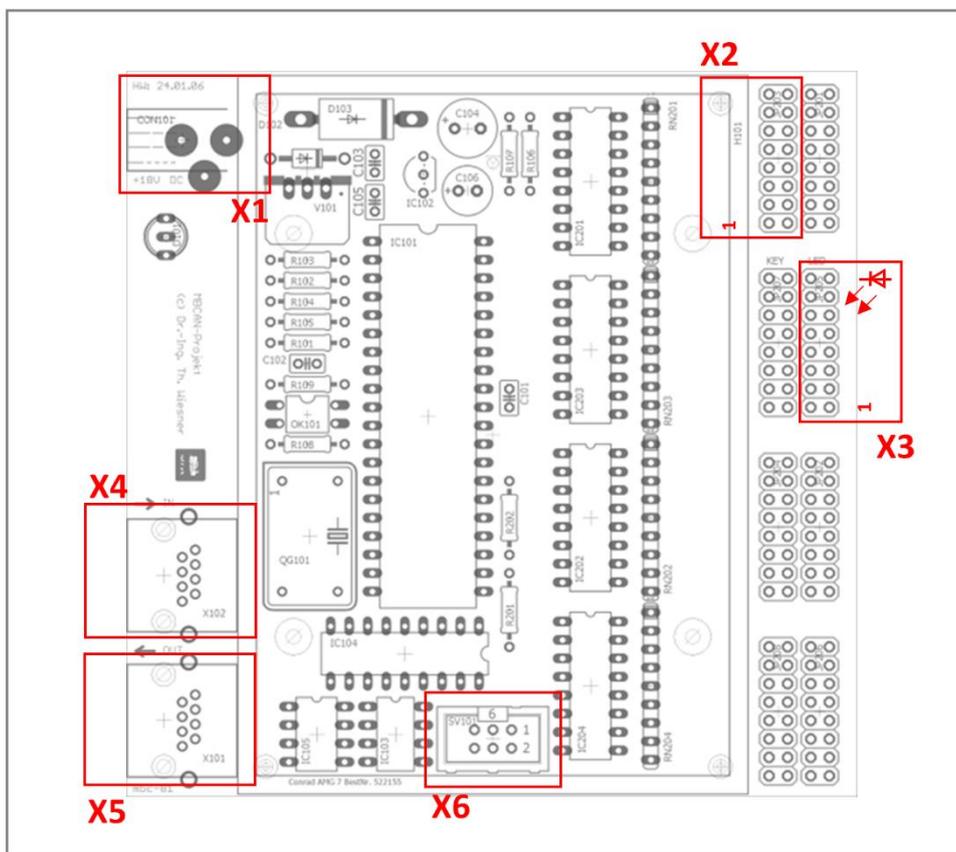


Abbildung 10-1: Steckverbinder

X1 dezentrale Spannungsversorgung

Diese Buchse wird genutzt, um die Basis-Spannungsversorgung in den MBCAN-Bus einzuspeisen. Der Eingang ist verpolungssicher.

X2 Taster-Anschlüsse

In diese Stiftleiste können entweder Einzeltaster oder aber ein Stellpult von Märklin® angeschlossen werden. Die Pins 1, 3, 5, 7, 9, 11, 13, 15 sind dabei das Bezugspotenzial *Masse*, die Pins 2, 4, 6, 8, 10, 12, 14, 16 die jeweiligen Eingänge. Dabei gilt folgende Zuordnung:

16 = E01 Rot, 14 = E01 Grün, ...

X3 LED-Anschlüsse

Die Stellungsanzeige wird durch die entsprechenden Module mbc-83, 84, 87 und/oder 91 zurückgemeldet. Die LEDs sind in derselben Reihenfolge wie die Taster angeordnet:

15/16 = E01 Rot, 13/14 = E01 Grün, ...

Die Ausrichtung der LED folgt dem Schema 15 = Kathode, 16 = Anode.

Bei der Nutzung als Tastenmatrix am emulierten LinkS88® haben die LED keine Funktion.

X4 *MBCAN-IN*

Modulverbindung zum vorherigen Modul in der Kette über Cat.5-Netzwerkkabel.

X5 *MBCAN-OUT*

Modulverbindung zum nächsten Modul in der Kette über Cat.5-Netzwerkkabel.

X6 *Optionale ISP-Schnittstelle*

Programmierschnittstelle für Atmel-Programmieradapter. Wird nur zur initialen Installation oder im Falle eines Modulcrashes benötigt.

11 Anschlussbeispiel

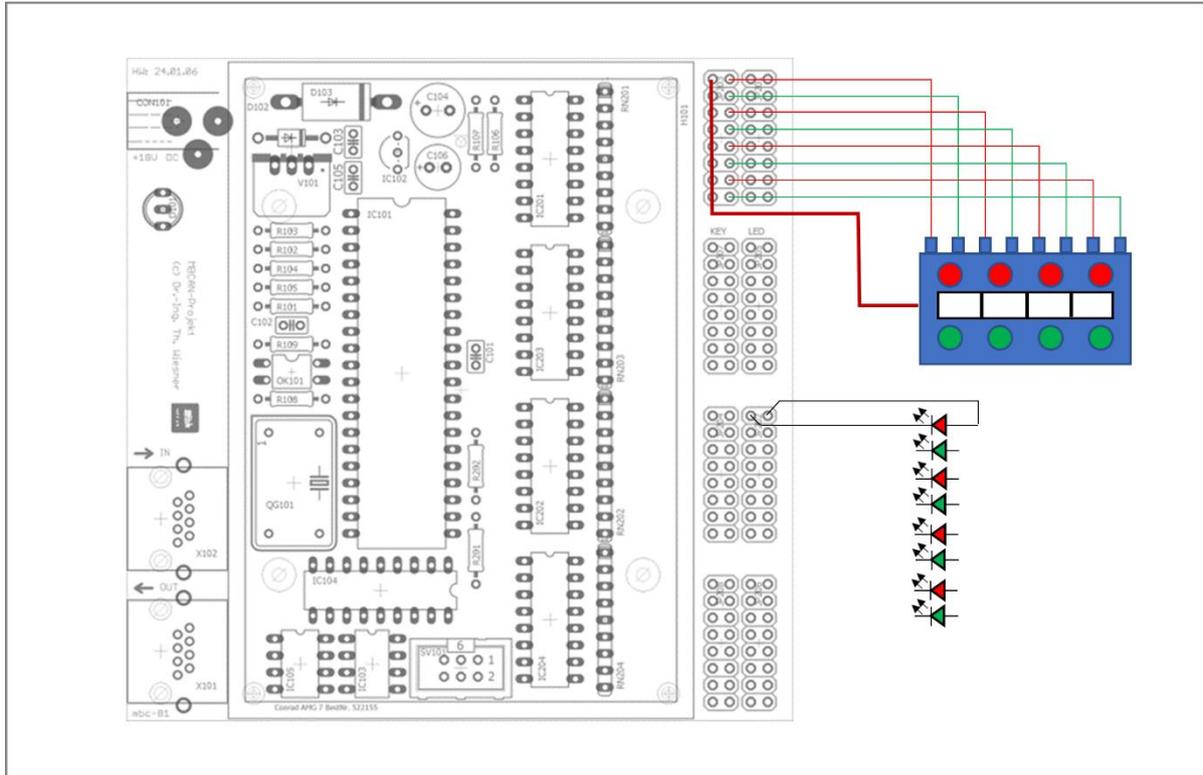


Abbildung 11-1: Anschluss von Stellpult und LEDs

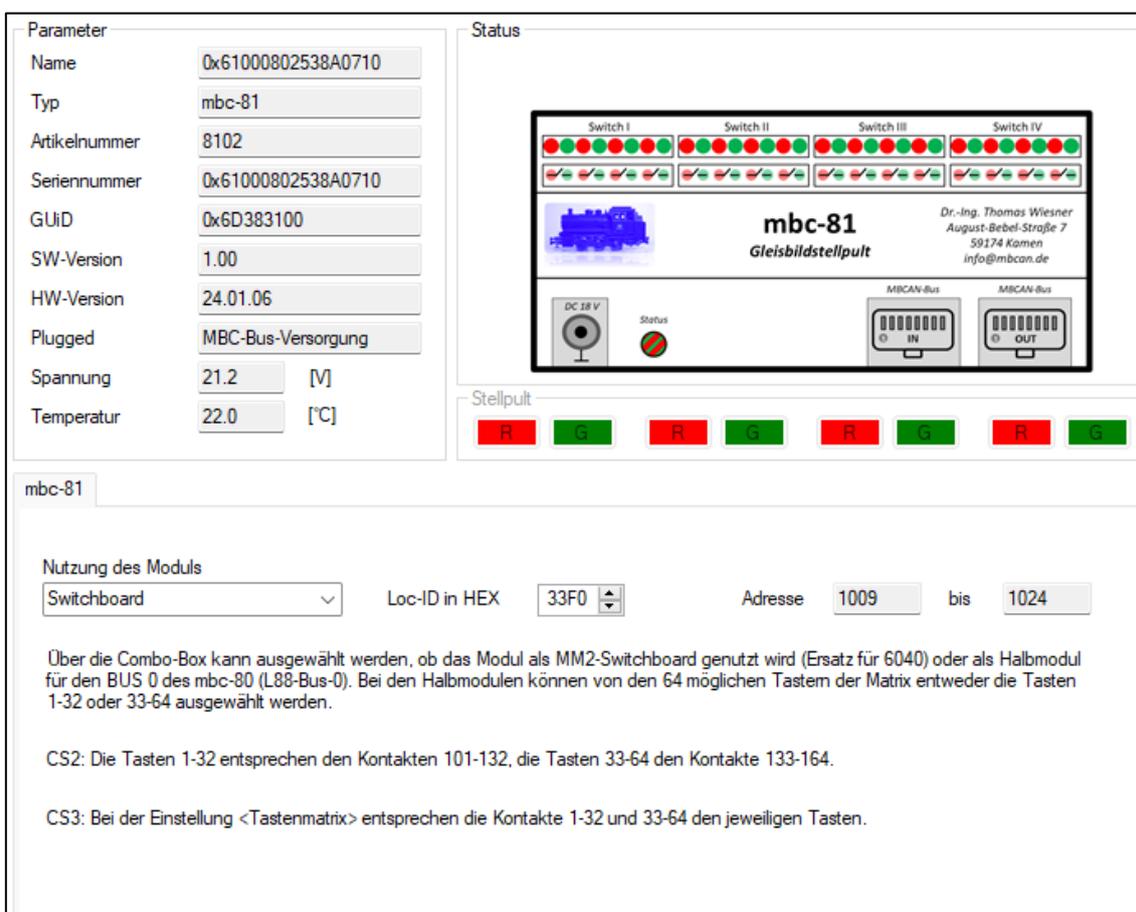
12 Inbetriebnahme

12.1 Modul in Betriebsbereitschaft versetzen

Nach dem Start des Parametriercenters und dem Anlegen der Spannungsversorgung an die **Steckverbindung X1** meldet sich das Modul automatisch an, sobald eine Verbindungsart zum MBCAN-Bus ausgewählt wurde (siehe Bedienungsanleitung zum Terminaladapter mbc-80).

12.2 Modul konfigurieren

Nach erfolgreicher Anmeldung am Parametriercenter und Auslesen des Moduls (vgl. Bedienungsanleitung zum Parametriercenter) erscheint folgender Konfigurationsbereich:



Parameter	
Name	0x61000802538A0710
Typ	mbc-81
Artikelnummer	8102
Seriennummer	0x61000802538A0710
GUID	0x6D383100
SW-Version	1.00
HW-Version	24.01.06
Plugged	MBC-Bus-Versorgung
Spannung	21.2 [V]
Temperatur	22.0 [°C]

Status

Switch I, Switch II, Switch III, Switch IV

mbc-81
Gleisbildstellpult

Dr.-Ing. Thomas Wiesner
August-Bebel-Straße 7
59174 Kamen
info@mbcan.de

DC 24V, Status, MBCAN-bus IN, MBCAN-bus OUT

Stellpult

R G R G R G R G

mbc-81

Nutzung des Moduls: Switchboard

Loc-ID in HEX: 33F0

Adresse: 1009 bis 1024

Über die Combo-Box kann ausgewählt werden, ob das Modul als MM2-Switchboard genutzt wird (Ersatz für 6040) oder als Halbmodul für den BUS 0 des mbc-80 (L88-Bus-0). Bei den Halbmodulen können von den 64 möglichen Tastern der Matrix entweder die Tasten 1-32 oder 33-64 ausgewählt werden.

CS2: Die Tasten 1-32 entsprechen den Kontakten 101-132, die Tasten 33-64 den Kontakte 133-164.

CS3: Bei der Einstellung <Tastenmatrix> entsprechen die Kontakte 1-32 und 33-64 den jeweiligen Tasten.

Abbildung 12-1: Konfigurationsbereich

Im obigen Beispiel wurde der Name des Moduls noch nicht geändert, hier wird die Seriennummer zunächst als Name verwendet. Er kann jederzeit über den Modul-Baum angepasst werden (siehe Bedienungsanleitung Terminaladapter mbc-80). Das Schaltpult ist deaktiviert.

Konfiguriert werden kann die **<Loc-ID>**, unter der das Modul seitens der Module mbc-83, 84, 87 und/oder 91 erreichbar ist resp. zu welchem Adressbereich die Stellkommandos weitergegeben werden, wenn im Combo-Feld **<Nutzung des Moduls>** *Switchboard* ausgewählt ist. Angegeben wird hier nur die Basisadresse. Bei Veränderung der **<Loc-ID>** über die Up-/Down-Buttons wird der Wert um 16 (0x10) angepasst. Dies entspricht immer genau dem Adressbereich von vier Modulen mbc-83, 84, 87 und/oder 91.

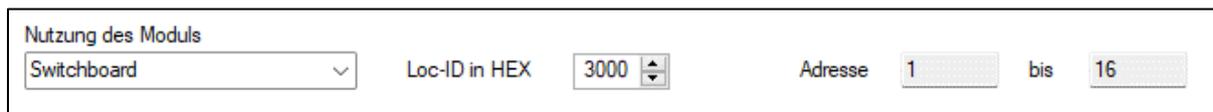


Abbildung 12-2: Auswahl als Switchboard

Die **<Adresse>** des korrespondierenden Ausgangs kann über folgende Gleichung der Modul-Loc-ID zugeordnet werden:

$$\text{Adresse Ausgang } X = \text{Loc-ID} - 0x3000 + X \text{ für } X = 1 \dots 16$$

Beispiele:

Loc-ID	Adresse Ein-/Ausgang 1	Adresse Ein-/Ausgang 2	Adresse ...	Adresse Ein-/Ausgang 16
0x3000	1	2		16
0x3010	17	18		32
0x3100	257	258		272
0x3120	289			304
0x33F0	1009	1022		1024

Alternativ kann das Modul als **Halbmodul einer Tastenmatrix am emulierten LinkS88® als Bus 0** parametrieren werden. Es sind entweder die Tasten 1 bis 32 oder alternativ die Tasten 33 bis 64 auswählbar, die Adresse des Switchboards wird automatisch deaktiviert. In diesem Modus sendet das Modul bei Betätigung des Tasteneingangs eine s88-Rückmeldenachricht für den Bus 0 an die CS2/3®.



Abbildung 12-3: Auswahl als Bus 0 Tastenmatrix Tasten 1..32



Abbildung 12-4: Auswahl als Bus 0 Tastenmatrix Tasten 33..64

Werden die Werte angepasst, wird das Feld *gelb* hinterlegt und alle Buttons deaktiviert. Dafür werden die Buttons **<Verwerfen>** und **<Update>** aktiviert. Mit **<Verwerfen>** kann die Eingabe rückgängig gemacht werden und der vorher gültige Wert wieder übernommen. Das Feld wird dann wieder in *weiß* hinterlegt. Mit dem Button **<Update>** wird der neue Parameter in das Modul geschrieben.

Die Bedeutung der anderen Buttons entnehmen Sie bitte der Bedienungsanleitung zum Terminaladapter mbc-80.

12.3 Modul mit der CS3® verbinden

Ist das Modul über den Terminaladapter mbc-80 mit der CS3® verbunden, meldet es sich über seine GUID an und ist im **<System/Einstellungen>-Bereich** unter **<Sonstige Geräte>** mit den Stammparametern anzeigbar.

Im **<Info>-Bereich** werden die Artikelnummer, die Version und die Seriennummer des Moduls in der Märklin®-Definition, die aktuelle Spannung und die Gehäuseinnentemperatur des Moduls angezeigt, letztere dynamisch.

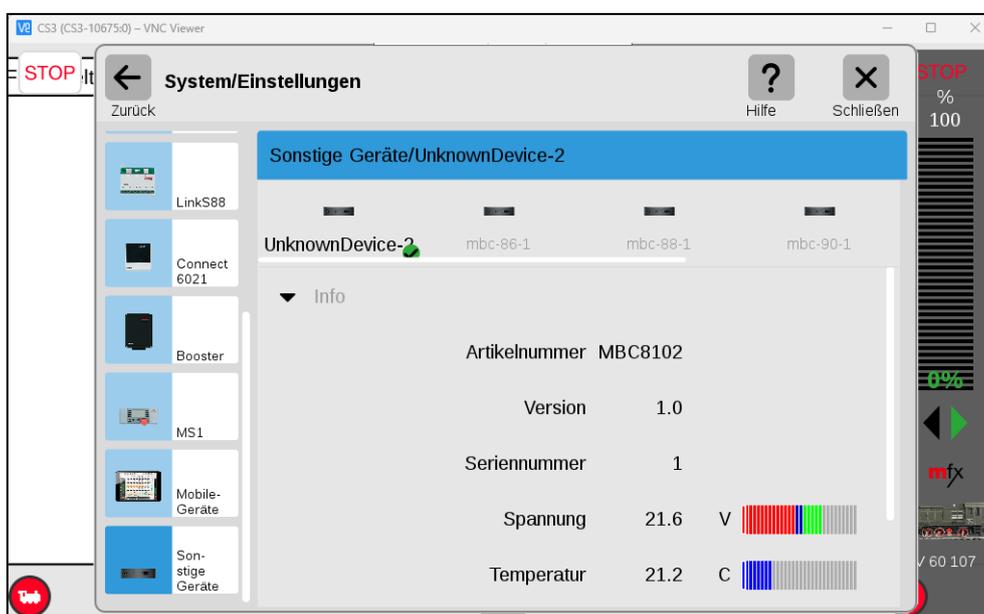


Abbildung 12-5: <Info>-Bereich

Die Seriennummer des Moduls in der Märklin®-Definition ist nicht identisch mit der Seriennummer der MBCAN-Module. Dies liegt darin begründet, dass Märklin® einen 32-Bit-Integer als Seriennummer zulässt und diese bei der Produktion der Geräte fest einprogrammiert. Bei MBCAN wird die Seriennummer aus dem eingesetzten DALLAS®-Chip verwendet, die 64 Bits umfasst. Um die zwingend durch Märklin® vorgegebene Kommunikation zur Seriennummer einzuhalten wird hier als Ersatzwert die Modulnummer des Modultyps aus GUID, erhöht um 1, eingetragen (die GUID-Modulnummer beginnt bei „0“).

Im **<Einstellungen>-Bereich** werden der Name des Moduls (Nickname), die Seriennummer in der MBCAN-Konvention, die Firmware- und die Hardwareversion des Moduls angezeigt (siehe Abbildung 12-6: <Einstellungen>-Bereich).

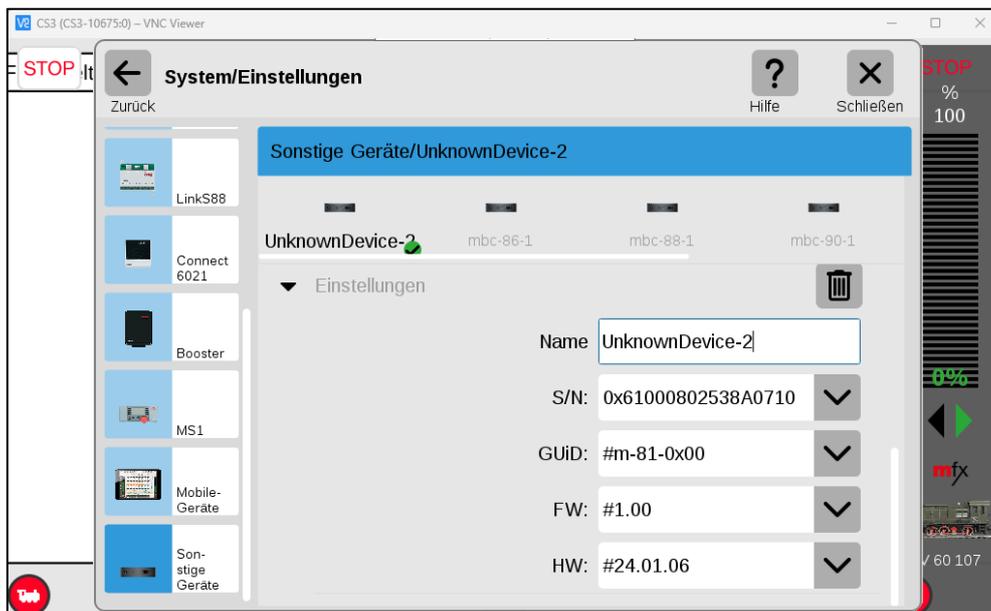


Abbildung 12-6: <Einstellungen>-Bereich

Das Feld <Name> ermöglicht einen anderen als den Default-Namen den die CS3® vergibt.

In der Funktion als **Halbmodul einer Tastenmatrix am Bus 0** werden s88®-Rückmeldenachrichten an die CS3® übertragen. Die Kontakte können über die Einstellung <Bus> in der GUI der CS3® bei den s88®-Rückmeldern im entsprechenden LinkS88® eingestellt werden, indem die Option *Tastenmatrix* ausgewählt wird. Die Kontaktnummern sind dann entweder 1 bis 32 oder 33 bis 64. In diesem Fall ist die Switchboardfunktion deaktiviert (vgl. Bedienungsanleitung mbc-90).

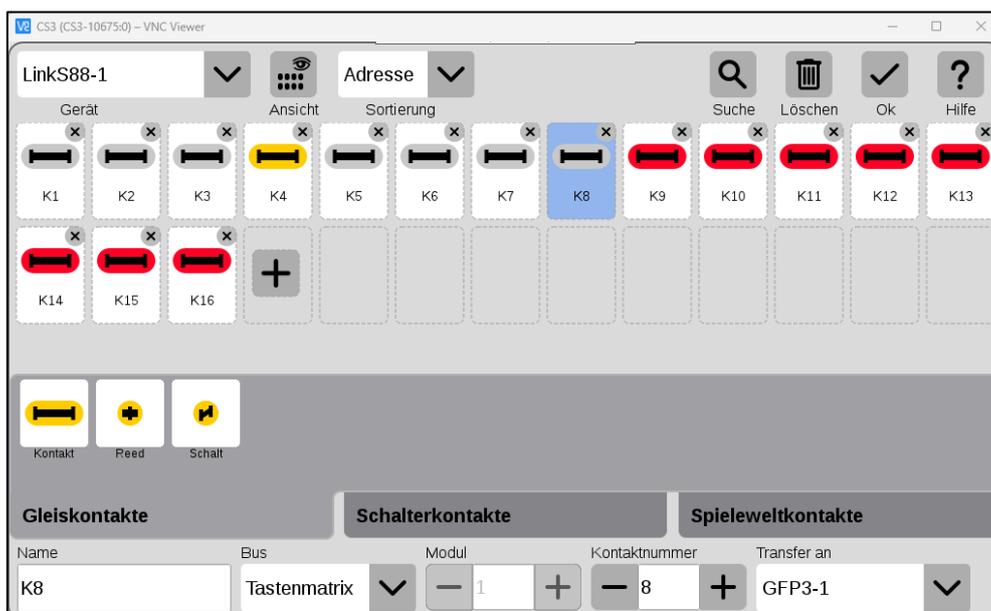


Abbildung 12-7: <Einstellungen>-Bereich s88-Rückmeldung, K1 .. K8 als Tastenmatrix festgelegt, K4 aktiviert



Die **Bus-0-Option Tastenmatrix** funktioniert anders als bei einem echten LinkS88® parallel zur **Option Direkt**, realisiert mit einem Modul mbc-90. Somit stehen sowohl die 16 direkten s88®-Kontakte über ein mbc-90 als auch die 64 Tasten über zwei mbc-81 als Rückmeldung zur Verfügung.

Die Anbindung an eine CS2® ist ebenfalls möglich und geschieht über einen vergleichbaren Prozess. Allerdings unterscheidet sich die GUI in den Details. Bitte in die Dokumentation der CS2® schauen für weitere Informationen.

13 Modulbilder



Abbildung 13-1: Fertiges Modul inkl. Gehäuse

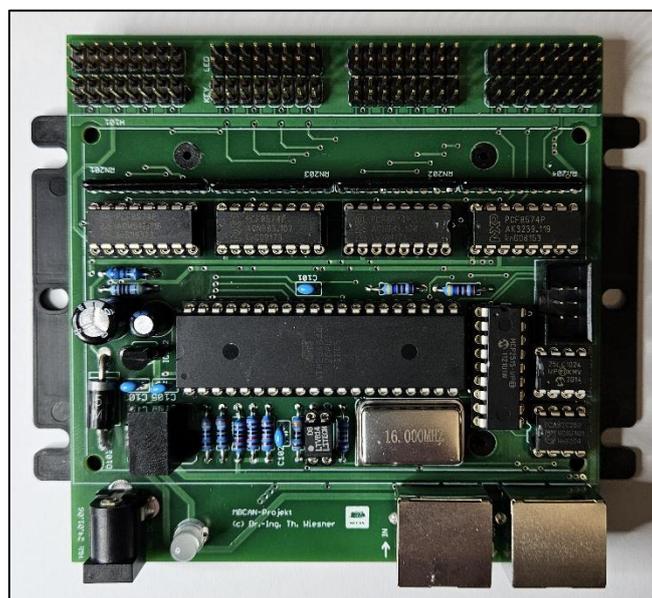


Abbildung 13-2: bestückte Platine

14 Systemarray-Belegung für Eigenentwicklungen

Nachfolgend ist die Belegung des Systemarrays abgebildet. Dies erleichtert bei Eigenentwicklungen von Software, die notwendigen Informationen des Moduls auslesen und parametrieren zu können.

14.1 Allgemeiner Bereich zum Modul

Der allgemeine Teil des Systemarrays ist bei allen mbc-Modulen gleich. Dargestellt ist die C-Schreibweise:

```
//=====
//= Systemarray mit Parametern zum Zustand =
//=====

// MBC_ARRAY_START          Start des belegten Systemarrays

#define MBC_ARRAY_START      1

// MBC_ALLG_START          Start des Allgemeinblocks

#define MBC_ALLG_START       MBC_ARRAY_START

// MBC_INFO_START          Start des Neuanmeldeblocks

#define MBC_INFO_START       MBC_ARRAY_START

// MBC_L_SNR               Laenge der Seriennummer
// MBC_S_SNR               Index Start Seriennummer
// MBC_SNR                 Seriennummer

#define MBC_L_SNR             8
#define MBC_S_SNR             MBC_ARRAY_START
#define MBC_SNR               sys_array[MBC_S_SNR]

// MBC_L_ART               Laenge der Artikelnummer
// MBC_S_ART               Index Start Artikelnummer
// MBC_ART_1               Artikelnummer Byte 4
// MBC_ART_2               Artikelnummer Byte 3
// MBC_ART_3               Artikelnummer Byte 2
// MBC_ART_4               Artikelnummer Byte 1

#define MBC_L_ART             4
#define MBC_S_ART             (MBC_L_SNR + MBC_S_SNR)
#define MBC_ART_1             sys_array[MBC_S_ART]
#define MBC_ART_2             sys_array[MBC_S_ART + 1]
#define MBC_ART_3             sys_array[MBC_S_ART + 2]
#define MBC_ART_4             sys_array[MBC_S_ART + 3]

// MBC_L_SW                Laenge Softwareversion
// MBC_S_SW                Index Start Softwareversion
// MBC_SW_1                Softwareversion Byte 3
// MBC_SW_2                Softwareversion Byte 2
// MBC_SW_3                Softwareversion Byte 1

#define MBC_L_SW              3
#define MBC_S_SW              (MBC_L_ART + MBC_S_ART)
#define MBC_SW_1              sys_array[MBC_S_SW]
#define MBC_SW_2              sys_array[MBC_S_SW + 1]
```

```
#define MBC_SW_3 sys_array[MBC_S_SW + 2]

// MBC_L_HW Laenge Hardwareversion
// MBC_S_HW Index Start Hardwareversion
// MBC_HW_1 Hardwareversion Byte 6
// MBC_HW_2 Hardwareversion Byte 5
// MBC_HW_3 Hardwareversion Byte 4
// MBC_HW_4 Hardwareversion Byte 3
// MBC_HW_5 Hardwareversion Byte 2
// MBC_HW_6 Hardwareversion Byte 1

#define MBC_L_HW 6
#define MBC_S_HW (MBC_L_SW + MBC_S_SW)
#define MBC_HW_1 sys_array[MBC_S_HW]
#define MBC_HW_2 sys_array[MBC_S_HW + 1]
#define MBC_HW_3 sys_array[MBC_S_HW + 2]
#define MBC_HW_4 sys_array[MBC_S_HW + 3]
#define MBC_HW_5 sys_array[MBC_S_HW + 4]
#define MBC_HW_6 sys_array[MBC_S_HW + 5]

// MBC_L_NAMEBLOCK Laenge des Modulnamens
// MBC_S_NAMEBLOCK Index Start Modulname
// MBC_NAME Name des Moduls

#define MBC_L_NAMEBLOCK 20
#define MBC_S_NAMEBLOCK (MBC_L_HW + MBC_S_HW)
#define MBC_NAME sys_array[MBC_S_NAMEBLOCK]

// MBC_INFO_ENDE Ende des Neuanmeldeblocks

#define MBC_INFO_ENDE (MBC_L_NAMEBLOCK + MBC_S_NAMEBLOCK - 1)

// MBC_L_GUID Laenge der GUID
// MBC_S_GUID Index Start GUID
// MBC_UiD_1 GUID Byte 4
// MBC_UiD_2 GUID Byte 3
// MBC_UiD_3 GUID Byte 2
// MBC_UiD_4 GUID Byte 1

#define MBC_L_GUID 4
#define MBC_S_GUID (MBC_L_NAMEBLOCK + MBC_S_NAMEBLOCK)
#define MBC_UiD_1 sys_array[MBC_S_GUID]
#define MBC_UiD_2 sys_array[MBC_S_GUID + 1]
#define MBC_UiD_3 sys_array[MBC_S_GUID + 2]
#define MBC_UiD_4 sys_array[MBC_S_GUID + 3]

// MBC_L_DB Laenge der Datenbankversion
// MBC_S_DB Index Start Datenbankversion
// MBC_DB Datenbanknummer des PC

#define MBC_L_DB 12
#define MBC_S_DB (MBC_L_GUID + MBC_S_GUID)
#define MBC_DB sys_array[MBC_S_DB]

// MBC_L_KN Laenge CS2-Geraetekennung
// MBC_S_KN Index Start CS2-Geraetekennung
// MBC_KN_H CS2-Geraetekennung HIGH
// MBC_KN_L CS2-Geraetekennung LOW
// MBC_AK_H CS2-Autokennung HIGH
```

```
// MBC_AK_L           CS2-Autokennung LOW
// MBC_CS2_GER        CS2-Geraetegruppe

#define MBC_L_KN       5
#define MBC_S_KN       (MBC_L_DB + MBC_S_DB)
#define MBC_KN_H       sys_array[MBC_S_KN]
#define MBC_KN_L       sys_array[MBC_S_KN + 1]
#define MBC_AK_H       sys_array[MBC_S_KN + 2]
#define MBC_AK_L       sys_array[MBC_S_KN + 3]
#define MBC_CS2_GER    sys_array[MBC_S_KN + 4]

// MBC_L_PARA         Laenge Parametersatz
// MBC_S_PARA         Index Start Parametersatz
// MBC_PLUGGED        Steckernetzteil gesteckt
// MBC_SPG_1          Spannung 10er Digit
// MBC_SPG_2          Spannung 1er Digit
// MBC_SPG_3          Spannung 0.1er Digit
// MBC_TMP_1          Temperatur 10er Digit
// MBC_TMP_2          Temperatur 1er Digit
// MBC_TMP_3          Temperatur 0.1er Digit

#define MBC_L_PARA     7
#define MBC_S_PARA     (MBC_L_KN + MBC_S_KN)
#define MBC_PLUGGED    sys_array[MBC_S_PARA]
#define MBC_SPG_1      sys_array[MBC_S_PARA + 1]
#define MBC_SPG_2      sys_array[MBC_S_PARA + 2]
#define MBC_SPG_3      sys_array[MBC_S_PARA + 3]
#define MBC_TMP_1      sys_array[MBC_S_PARA + 4]
#define MBC_TMP_2      sys_array[MBC_S_PARA + 5]
#define MBC_TMP_3      sys_array[MBC_S_PARA + 6]

// MBC_ALLG_ENDE      Ende des Allgemeinblocks

#define MBC_ALLG_ENDE  (MBC_L_PARA + MBC_S_PARA - 1)

// MBC_BOOT_START     Anfang des BOOT-Bereiches

#define MBC_BOOT_START (MBC_L_PARA + MBC_S_PARA)

// BOOT-Array-PAGE-Size Groesse der EEPROM-PAGE

#define MBC_BOOT_PGSZ  256

// Anzahl der Pakete fuer die BOOT-Array-Uebertragung

#define MBC_BOOT_PGMAX (MBC_BOOT_PGSZ / 4)

// MBC_L_BOOT         Laenge BOOT-Array
// MBC_S_BOOT         Index Start BOOT-Array
// MBC_BOOT           BOOT-Array

#define MBC_L_BOOT     (2 + MBC_BOOT_PGSZ)
#define MBC_S_BOOT     MBC_BOOT_START
#define MBC_BOOT_ARRAY sys_array[MBC_BOOT_START]

// MBC_BOOT_ENDE     Ende des BOOT-Bereiches

#define MBC_BOOT_ENDE (MBC_L_BOOT + MBC_S_BOOT - 1)
```

```
// Je nach Modultyp ist das sys_array unterschiedlich lang. Alle Module
// haben einen gemeinsamen Teil mit einer Laenge von 69 Bytes und einem Boot-Block
// von 256 Bytes. Bei den anderen Modulen kommt noch ein General-Purpose-Array mit
// einer Laenge von 2048 Bytes hinzu, respektive 512 Bytes beim mbc-80.

// MBC_L_GENPURP           Laenge des General Purpose Arrays
// MBC_S_GENPURP           Index Start des General-Purpose-Arrays
// MBC_ARRAY_MAX           Laenge des Systemarrays

#ifdef _mbc_80_
    #define MBC_L_GENPURP           512

    #define MBC_S_GENPURP           (MBC_BOOT_ENDE + 1)
    #define MBC_ARRAY_MAX           (MBC_L_GENPURP + MBC_S_GENPURP)
#else
    #define MBC_L_GENPURP           2048
    #define MBC_S_GENPURP           (MBC_BOOT_ENDE + 1)
    #define MBC_ARRAY_MAX           (MBC_L_GENPURP + MBC_S_GENPURP)
#endif

// MBC_ARRAY_ENDE           Ende des Systemarrays

#define MBC_ARRAY_ENDE           (MBC_ARRAY_MAX - 1)
```

Listing 14-1: Modulparameter

Um die geänderten Parameter in das interne EEPROM zu schreiben werden folgende Indizes verwendet:

```
//=====
//= Parameter fuer die EEPROM-Steuerung =
//=====

#define EE_NAME           0x01           // ID Block Name
#define EE_GUID           0x02           // ID Block GUID
#define EE_KENNUNG           0x03           // ID Block Geraetekennung
#define EE_DB           0x04           // ID Block Datenbank
```

Listing 14-2: EEPROM-Indizes Modul

14.2 Modulspezifischer Bereich für Funktionsparameter

Der modulspezifische Teil des Systemarrays beinhaltet die für die Funktion des Moduls vom Standard abweichenden Parameter. Dieser liegt im GENPURP-Bereich des Systemarrays. Dargestellt ist die C-Schreibweise:

```
//=====
//= Sonderbelegung des Systemarrays nach dem Allgemeinblock =
//=====

// Laenge modulspezifische Parameter
// Index Start modulspezifische Parameter
// MAddr HIGH
// MAddr LOW

#define MBC_81_L_MM          2

#define MBC_81_S_MM          MBC_S_GENPURP
#define MBC_81_MM_H          sys_array[MBC_81_S_MM]
#define MBC_81_MM_L          sys_array[MBC_81_S_MM + 1]
```

Listing 14-3: Funktionsparameter

Um die geänderten Parameter in das interne EEPROM zu schreiben werden folgende Indizes verwendet:

```
//=====
//= Parameter fuer die EEPROM-Steuerung =
//=====

#define EE_81_MAddr          0x0A          // ID Block Adresse
```

Listing 14-4: EEPROM-Indizes Funktionen

15 Befehlssatz zu den Modulen

Um die Module des MBCAN-Projektes auf dem CAN-Bus ansprechen und parametrieren zu können, bedarf es neben der Geräte-UID auch einen adäquaten Befehlssatz. Der Befehlssatz von Märklin® setzt sich aus Kommandos zusammen, die im CAN-Header integriert sind. Da dieser Header sehr sensibel auf Fehler reagiert, fällt er für eigene Befehlsübertragungen aus.

Märklin® hat aber eine Möglichkeit geschaffen, dass Privatpersonen, Vereine o.ä. freie Adressräume in der Loc-ID (Local ID, nicht Lokomotiv-ID) nutzen können. Diese liegen im Adressraum 0x00001800 bis 0x00001BFF (Datenbytes 1 bis 4 der CAN-Nachricht) und sind u.a. über das Schaltkommando 0x0B (= 0x16 im CAN-Header) verfügbar.

Der Befehlssatz von MBCAN baut auf diesem Adressraum und das Märklin®-Schaltkommando auf. Anders als bei Märklin® üblich, werden nur uni-direktionale Befehle generiert. D.h., dass das Response-Bit im CAN-Header nicht genutzt wird.

```
//=====
//= CAN-Befehlsnummern PC-Kommunikation initialisieren           =
//= Dieser ist der zweite Teil in der Addr der CAN-Nachricht 0x18xx =
//=====

// PC_DB_H           PC - Datenbanknummer HIGH
// PC_DB_M           PC - Datenbanknummer MID
// PC_DB_L           PC - Datenbanknummer LOW
// PC_KENNER         PC - Geraetekenner und Identifier
// PC_NEU            PC - Neuanmeldungsanforderung des PC
// PC_NEU_DATA       PC - Neuanmeldungs kanal des PC
// MD_NEU_DATA       MD - Neuanmeldungs kanal des Moduls
// PC_RESET          PC - Reset durch PC
// PC_MD_DEL         PC - Modul wurde aus Datenbank geloescht
// PC_ALIVE          PC - Alivemeldung durch PC angefordert
// MD_ALIVE          MD - Acknowledge des Moduls auf PC_ALIVE
// PC_ARRAY          PC - Anfordern, auf das Systemarray des Moduls
//                  zuzugreifen
// MD_ARRAY          MD - Acknowledge des Moduls auf PC_ARRAY
// PC_ARRAY_DATA     PC - Schreiben/Lesen und ggf. Wert und Systemarray-
//                  index uebergeben
// MD_ARRAY_DATA     MD - Ack des Moduls auf PC_ARRAY_DATA und Wert aus
//                  dem Systemarray uebergeben
// PC_UPGRADE        PC - Anfordern, auf das Systemarray des Moduls
//                  zuzugreifen
// MD_UPGRADE        MD - Acknowledge des Moduls auf PC_UPGRADE
// PC_UPGRADE_DATA   PC - Schreiben/Lesen und ggf. Wert und Systemarray-
//                  index uebergeben
// MD_UPGRADE_DATA   MD - Ack des Moduls auf PC_UPGRADE_DATA und Wert
//                  aus dem Systemarray uebergeben
// PC_BOOT           PC - Modul mit neuer Firmware starten
// MD_S88            MD - S88-Stellungsmeldung

#define PC_DB_H      0x00
#define PC_DB_M      0x01
#define PC_DB_L      0x02
#define PC_KENNER    0x03
#define PC_NEU       0x04
#define PC_NEU_DATA  0x05
```

```
#define MD_NEU_DATA      0x06
#define PC_RESET        0x07
#define PC_MD_DEL       0x08
#define PC_ALIVE        0x09
#define MD_ALIVE        0x0A
#define PC_ARRAY        0x0B
#define MD_ARRAY        0x0C
#define PC_ARRAY_DATA   0x0D
#define MD_ARRAY_DATA   0x0E
#define PC_UPGRADE      0x0F
#define MD_UPGRADE      0x10
#define PC_UPGRADE_DATA 0x11
#define MD_UPGRADE_DATA 0x12
#define PC_BOOT         0x13
#define MD_S88          0x14
```

Listing 15-1: Befehlssatz der MBCAN-Module

Die Nachrichten auf dem MBCAN-Bus zur Kommunikation der Module untereinander und zum Parametriercenter entsprechen wie beschrieben der Märklin®-Konvention mit einer Datenlänge von 8 Byte (vgl. UDP-Datenformat bei Kopplung mit der CS2®):

Beispiel: **00 16 5F 38 08 00 00 18 09 6D 38 34 01**

Übersetzung:

PRI0: 0x00 = Normale Priorität der Nachricht

KOMMANDO: 0x16 = Schaltkommando

HASH: 0x5F38 = HASH des Senders aus der GUID gemäß Märklin®

DLC: 0x08 = Länge der Nachricht

Loc-ID: 0x00001809 = ALIVE-Anfrage (0x1800 als Basis und 0x0009 als Befehl MD_ALIVE)

GUID: 0x6D383401 = Anfrage an mbc-84 #1

Weitere Informationen zu den CAN-Nachrichten gemäß Märklin®-Konvention siehe Quellenangabe unten.

Nachfolgend sind die Befehle und ihre Funktionen aufgeführt.

15.1 PC_DH_H - Übertragung des Datenbanknamens mit 12 Bytes (1-4)

Befehl	PC_DB_H
Sender	PC
Loc-ID	0x00001800
Funktion	Übertragung des Datenbanknamens mit 12 Bytes (1-4)
Beschreibung	Bytes 1 bis 6 stellen das Datum, Bytes 7 bis 12 die Uhrzeit dar. Beispielstring: "070917235340" = am 07.09.2017 um 23:53:40 wurde die Datenbank erstellt. Die einzelnen Bytes werden in ASCII-Werte übersetzt und dann übertragen. Dieser String findet sich auch im Dateinamen der exportierten Datenbank aus dem Parametriercenter.
Genutzte Datenbytes	D0 – D3: Loc-ID 0x00001800
	D4 – D7: Bytes 1-4 des Datum-/Uhrzeit-Strings
Nachricht	00 16 5F 38 08 00 00 18 00 30 37 30 39
Antwort	-/-

15.2 PC_DH_M - Übertragung des Datenbanknamens mit 12 Bytes (5-8)

Befehl	PC_DB_M
Sender	PC
Loc-ID	0x00001801
Funktion	Übertragung des Datenbanknamens mit 12 Bytes (5-8)
Beschreibung	Bytes 1 bis 6 stellen das Datum, Bytes 7 bis 12 die Uhrzeit dar. Beispielstring: "070917235340" = am 07.09.2017 um 23:53:40 wurde die Datenbank erstellt. Die einzelnen Bytes werden in ASCII-Werte übersetzt und dann übertragen. Dieser String findet sich auch im Dateinamen der exportierten Datenbank aus dem Parametriercenter.
Genutzte Datenbytes	D0 – D3: Loc-ID 0x00001801
	D4 – D7: Bytes 5-8 des Datum-/Uhrzeit-String
Nachricht	00 16 5F 38 08 00 00 18 00 31 37 32 33
Antwort	-/-

15.3 PC_DH_L - Übertragung des Datenbanknamens mit 12 Bytes (9-12)

Befehl	PC_DB_L
Sender	PC
Loc-ID	0x00001802
Funktion	Übertragung des Datenbanknamens mit 12 Bytes (9-12)
Beschreibung	Bytes 1 bis 6 stellen das Datum, Bytes 7 bis 12 die Uhrzeit dar. Beispielstring: "070917235340" = am 07.09.2017 um 23:53:40 wurde die Datenbank erstellt. Die einzelnen Bytes werden in ASCII-Werte übersetzt und dann übertragen. Dieser String findet sich auch im Dateinamen der exportierten Datenbank aus dem Parametriercenter.
Genutzte Datenbytes	D0 – D3: Loc-ID 0x00001802
	D4 – D7: Bytes 5-8 des Datum-/Uhrzeit-Strings
Nachricht	00 16 5F 38 08 00 00 18 00 35 33 34 30
Antwort	-/-

15.4 PC_KENNER - Kenner und Identifier für die Module

Befehl	PC_KENNER
Sender	PC
Loc-ID	0x00001803
Funktion	Kenner und Identifier für die Module
Beschreibung	<p>Die Kennung der MBCAN-Module folgt strikt dem Format der Geräte-UiD von Märklin. In der GUID stellt die erste Stelle die Kennung dar. Zurzeit verwendet MBCAN die Kennung "m" (0x6D). Im Parametriercenter kann die Kennung angepasst werden, falls Märklin den Kenner "m" für seine eigene Module reklamiert.</p> <p>Darüber hinaus bekommt jedes Modul noch einen Identifier, mit dem es sich an der GUID der CS2/3 als „Sonstige Geräte“ anmelden kann. Zurzeit ist dies „AAAA“ (0xAAAA). Im Parametriercenter kann die Kennung angepasst werden, falls Märklin den Identifier "m" für seine eigene Module reklamiert. Ausgenommen sind die Module mbc-80 (Identifier 0x0040) und mbc-82 (Identifier 0x0000) die von Märklin fest vorgegeben sind. Diese Identifier sind in der Firmware der Module bereits fest integriert.</p>
Genutzte Datenbytes	D0 - D3: Loc-ID 0x00001803 D4: 0x00 D5: Kennung D6 – D7: Identifier
Nachricht	00 16 5F 38 08 00 00 18 03 00 6D AA AA
Antwort	-/-

15.5 PC_NEU - Neuanmeldeaufforderung

Befehl	PC_NEU
Sender	PC
Loc-ID	0x00001804
Funktion	Neuanmeldeaufforderung
Beschreibung	Zyklische Aufforderung an neu am MBCAN-Bus angeschlossene und noch nicht angemeldete Module, sich am Parametriercenter anzumelden. Dies gilt auch für Module, die über das Parametriercenter zurückgesetzt wurden.
Genutzte Datenbytes	D0 - D3: Loc-ID 0x00001804
	D4 - D7: 0x00
Nachricht	00 16 5F 38 08 00 00 18 04 00 00 00 00
Antwort	MD_NEU_DATA

15.6 PC_NEU_DATA - Rückmeldung PC an das Modul während des Neuanmeldeprozesses

Befehl	PC_NEU_DATA
Sender	PC
Loc-ID	0x00001805
Funktion	Rückmeldung des PC an das Modul während des Neuanmeldeprozesses
Beschreibung	Der PC sendet das empfangene Seriennummer-Byte auf den MBCAN-Bus zurück als Quittierung. Das entsprechende Modul reagiert dann mit dem nächsten Byte der Seriennummer, alle anderen Module schalten in den Listen-Modus und reagieren erst nach einer weiteren PC_NEU-Nachricht, falls sie noch nicht erfolgreich angemeldet waren.
Genutzte Datenbytes	D0 - D3: Loc-ID 0x00001805
	D4 - D6: 0x00
	D7: n-tes Byte xx der Seriennummer
Nachricht	00 16 5F 38 08 00 00 18 05 00 00 00 xx
Antwort	MD_NEU_DATA

15.7 MD_NEU_DATA - Meldung des Moduls während des Neuanmeldeprozesses

Befehl	MD_NEU_DATA
Sender	Modul
Loc-ID	0x00001806
Funktion	Meldung des Moduls während des Neuanmeldeprozesses
Beschreibung	Wenn das Modul noch nicht am Parametriercenter angemeldet war, reagiert es mit dieser Nachricht an den PC. Es sendet sein erstes Byte seiner Seriennummer an den PC. Reagiert der PC mit der Nachricht PC_NEU_DATA mit exakt dem gleichen Byte, sendet es weitere Bytes seiner Seriennummer, bis entweder alle Bytes übertragen wurden (erfolgreiche Anmeldung) oder der PC gerade ein anderes Modul initiiert. Stimmt das Byte nicht überein, geht es in den Listen-Modus und wartet auf eine weitere PC_NEU-Nachricht.
Genutzte Datenbytes	D0 - D3: Loc-ID 0x00001806
	D4 - D6: 0x00
	D7: n-tes Byte xx der Seriennummer
Nachricht	00 16 2B 17 08 00 00 18 06 00 00 00 xx
Antwort	PC_NEU_DATA

15.8 PC_RESET - Durchführen eines Hardware-Resets auf dem Modul

Befehl	PC_RESET
Sender	PC
Loc-ID	0x00001807
Funktion	Durchführen eines Hardware-Resets auf dem Modul
Beschreibung	Über das Parametriercenter können Module gezielt einem RESET unterzogen werden. Die Identifizierung der Module geschieht über ihre GUID.
Genutzte Datenbytes	D0 - D3: Loc-ID 0x00001807 D4 - D7: GUID des Moduls
Nachricht	GUID mbc-84 #1 = 6D 38 34 01 00 16 5F 38 08 00 00 18 07 6D 38 34 01
Antwort	-/-

15.9 PC_MD_SEL - Modul aus Datenbank entfernen

Befehl	PC_MD_DEL
Sender	PC
Loc-ID	0x00001808
Funktion	Modul aus Datenbank entfernen
Beschreibung	Das Modul wurde aus der Datenbank entfernt und kann sich an dieser Datenbank auch nicht mehr neu anmelden. Wird in der Regel nur bei Modulen verwendet, die sich in der Datenbank befinden aber nicht mehr am Bus angeschlossen werden sollen. Wird nur einmal gesendet, wenn das Modul im Parametriercenter gelöscht wird. Ist das Modul nicht am Bus und wird nach einem Neustart der Software wieder am Bus angeschlossen, meldet es sich nicht mehr neu an, es sei denn, die Datenbank wird neu erstellt.
Genutzte Datenbytes	D0 - D3: Loc-ID 0x00001808
	D4 - D7: GUID des Moduls
Nachricht	GUID mbc-84 #1 = 6D 38 34 01
	00 16 5F 38 08 00 00 18 08 6D 38 34 01
Antwort	-/-

15.10 PC_ALIVE - ALIVE-Abfrage

Befehl	PC_ALIVE
Sender	PC
Loc-ID	0x00001809
Funktion	ALIVE-Abfrage
Beschreibung	Zyklische Abfrage über die GUID, ob das betreffende Modul sich noch am MBCAN-Bus befindet. Es antwortet mit der Nachricht MD_ALIVE.
Genutzte Datenbytes	D0 - D3: Loc-ID 0x00001809
	D4 - D7: GUID des Moduls
Nachricht	GUID mbc-84 #1 = 6D 38 34 01
	00 16 5F 38 08 00 00 18 09 6D 38 34 01
Antwort	MD_ALIVE

15.11 MD_ALIVE - ALIVE-Abfrage

Befehl	MD_ALIVE
Sender	Modul
Loc-ID	0x0000180A
Funktion	ALIVE-Abfrage
Beschreibung	Zyklische Abfrage über die GUID, ob das betreffende Modul sich noch am MBCAN-Bus befindet. Es antwortet mit der Nachricht MD_ALIVE.
Genutzte Datenbytes	D0 - D3: Loc-ID 0x0000180A
	D4 - D7: GUID des Moduls
Nachricht	GUID mbc-84 #1 = 6D 38 34 01
	00 16 2B 17 08 00 00 18 0A 6D 38 34 01
Antwort	-/-

15.12 PC_ARRAY - Zugriff Systemarray anfragen

Befehl	PC_ARRAY
Sender	PC
Loc-ID	0x0000180B
Funktion	Zugriff Systemarray anfragen
Beschreibung	Der PC fragt über die GUID an, ob er auf das Systemarray des Moduls zugreifen darf.
Genutzte Datenbytes	D0 - D3: Loc-ID 0x0000180B
	D4 - D7: GUID des Moduls
Nachricht	GUID mbc-84 #1 = 6D 38 34 01
	00 16 5F 38 08 00 00 18 0B 6D 38 34 01
Antwort	MD_ARRAY

15.13 MD_ARRAY - Zugriff Systemarray freigeben

Befehl	MD_ARRAY
Sender	Modul
Loc-ID	0x0000180C
Funktion	Zugriff Systemarray freigeben
Beschreibung	Antwort des durch die GUID im Befehl PC_ARRAY adressierten Moduls mit Freigabe des Zugriffs. Das Modul geht dann in die Wartestellung, alle anderen Module werden die folgenden Anfragen des PC nicht mehr aus. Ausgenommen sind Anfragen des PC außerhalb des Befehls PC_ARRAY_DATA.
Genutzte Datenbytes	D0 - D3: Loc-ID 0x0000180C
	D4 - D7: GUID des Moduls
Nachricht	GUID mbc-84 #1 = 6D 38 34 01
	00 16 2B 17 08 00 00 18 0C 6D 38 34 01
Antwort	-/-

15.14 PC_ARRAY_DATA - Zugriff Systemarray freigegeben

Befehl	PC_ARRAY_DATA
Sender	PC
Loc-ID	0x0000180D
Funktion	Zugriff Systemarray freigegeben
Beschreibung	<p>Der PC stellt die Zugriffsanfrage. Dies kann entweder ein Lese- oder ein Schreibzugriff sein. Außerdem ist der Systemarray-Index enthalten, der gelesen oder beschrieben werden soll.</p> <p>Beispiel: D4 = 0 -> Lesen, 1 -> Schreiben D5 + D6 = Systemarray-Index D7 = zu schreibender Wert, bei lesendem Zugriff irrelevant</p> <p>Über den Index des Systemarrays wird außerdem das Ende einer Datenübertragung angezeigt. Liegt der Index über der Maximallänge des Systemarrays und entspricht es einem bestimmten Wert, wird die Wartestellung des Modus für weitere Datenübertragungen aufgehoben und alle anderen Module können wieder auf einen PC_ARRAY-Zugriff angesprochen werden.</p>
Genutzte Datenbytes	D0 - D3: Loc-ID 0x0000180D D4: 0 -> Lesen, 1 -> Schreiben D5 - D6: Systemarray-Index D7: zu schreibender Wert, beim Lesen n.c.
Nachricht	Wert 0x0A an die Stelle 0x0001 im Systemarray schreiben 00 16 5F 38 08 00 00 18 0D 01 00 01 0A
Antwort	MD_ARRAY_DATA

15.15 MD_ARRAY_DATA - Antwort des Moduls auf Systemarray-Zugriff

Befehl	MD_ARRAY_DATA
Sender	Modul
Loc-ID	0x0000180E
Funktion	Antwort des Moduls auf Systemarray-Zugriff
Beschreibung	Bei einem lesenden Zugriff übergibt das Modul auf D7 den Inhalt des Systemarrays, bei einem schreibenden Zugriff ist D7 irrelevant. Die anderen Datenbytes der Nachricht (D4 ... D6) sind identisch mit der Nachricht des PC.
Genutzte Datenbytes	D0 - D3: Loc-ID 0x0000180E
	D4: 0 -> Lesen, 1 -> Schreiben
	D5 - D6: Systemarray-Index
	D7: gelesener Inhalt des Systemarrays, beim Schreiben n.c.
Nachricht	Gelesener Wert 0x07 aus der Stelle 0x0108 im Systemarray
	00 16 2B 17 08 00 00 18 0E 00 01 08 07
Antwort	-/-

15.16 PC_UPGRADE - Firmware-Upgrade

Befehl	PC_UPGRADE
Sender	PC
Loc-ID	0x0000180F
Funktion	Firmware-Upgrade
Beschreibung	Der PC fragt über die GUID an, ob er die Firmware des Moduls upgraden darf. Ist nur aktiv bei Modulen der 3. Generation und nicht gültig für die Module des Typs mbc-91.
Genutzte Datenbytes	D0 - D3: Loc-ID 0x0000180F
	D4 - D7: GUID des Moduls
Nachricht	GUID mbc-84 #1 = 6D 38 34 01
	00 16 5F 38 08 00 00 18 0F 6D 38 34 01
Antwort	MD_UPGRADE

15.17 MD_UPGRADE - Firmware-Upgrade freigeben

Befehl	MD_UPGRADE
Sender	Modul
Loc-ID	0x00001810
Funktion	Firmware-Upgrade freigeben
Beschreibung	Antwort des durch die GUID im Befehl PC_UPGRADE adressierten Moduls mit Freigabe des Zugriffs. Das Modul geht dann in die Wartestellung, alle anderen Module werden die folgenden Anfragen des PC nicht mehr aus. Ausgenommen sind Anfragen des PC außerhalb des Befehls PC_UPGRADE_DATA.
Genutzte Datenbytes	D0 - D3: Loc-ID 0x00001810
	D4 - D7: GUID des Moduls
Nachricht	GUID mbc-84 #1 = 6D 38 34 01
	00 16 2B 17 08 00 00 18 10 6D 38 34 01
Antwort	-/-

15.18 PC_UPGRADE_DATA - Schreibe Firmware

Befehl	PC_UPGRADE_DATA
Sender	PC
Loc-ID	0x00001811
Funktion	Schreibe Firmware
Beschreibung	Der PC übermittelt die Upgrade-Daten. Das Modul speichert diese in das externe EEPROM zur Vorbereitung der Neuprogrammierung. Die Daten werden PAGE-weise (je 64 Byte) vom Parametriercenter übertragen, so dass der BOOTLOADER hinterher die Daten aus dem externen EEPROM auch korrekt auslesen kann.
Genutzte Datenbytes	HASH: Laufende Nummer in der jeweiligen PAGE
	D0 - D3: Loc-ID 0x00001811
	D4 - D7: 4 zu schreibende Bytes
Nachricht	Schreibe im laufenden Index 2 die Werte 0x01, 0x00, 0x01 und 0x0A fortlaufend in das externe EEPROM
	00 16 03 02 08 00 00 18 11 01 00 01 0A
Antwort	MD_UPGRADE_DATA

15.19 MD_UPGRADE_DATA - Antwort des Moduls auf Schreibe Firmware

Befehl	MD_UPGRADE_DATA
Sender	Modul
Loc-ID	0x00001812
Funktion	Antwort des Moduls auf Schreibe Firmware
Beschreibung	Das Modul antwortet mit der exakten Datenstruktur der gesendeten Nachricht und signalisiert damit, dass es die Upgrade-Daten im externen EEPROM gespeichert hat.
Genutzte Datenbytes	HASH: Laufende Nummer in der jeweiligen PAGE
	D0 - D3: Loc-ID 0x00001812
	D4 - D7: 4 zu schreibende Bytes
Nachricht	Schreibe im laufenden Index 2 die Werte 0x01, 0x00, 0x01 und 0x0A fortlaufend in das externe EEPROM
	00 16 03 02 08 00 00 18 12 01 00 01 0A
Antwort	-/-

15.20 PC_BOOT - Modul neu Booten

Befehl	PC_BOOT
Sender	PC
Loc-ID	0x00001813
Funktion	Modul neu Booten
Beschreibung	Nach erfolgreicher Übertragung der neuen Firmware signalisiert der PC einen Hardwarerest des Moduls. Dies geschieht nicht über den Befehl PC_RESET, da vorher noch Identifier in das externe EEPROM gespeichert werden müssen die anzeigen, dass eine neue Firmware vorliegt.
Genutzte Datenbytes	D0 - D3: Loc-ID 0x00001813 D4 - D7: GUID des Moduls
Nachricht	GUID mbc-84 #1 = 6D 38 34 01 00 16 5F 38 08 00 00 18 13 6D 38 34 01
Antwort	-/-

15.21 MD_S88 - Stellungsmeldung mbc-88 / mbc-90

Befehl	MD_S88
Sender	Modul
Loc-ID	0x00001814
Funktion	Stellungsmeldung mbc-88 / mbc-90
Beschreibung	Sendet bei Statusänderung eines PINs die Stellung auf den MBCAN-Bus, so dass sowohl des Parametriercenter als auch andere Module diese ggf. weiterverarbeiten können. Ist ein Relikt aus den ersten beiden Generationen der MBCAN-Modulreihe und sollte bei Eigenentwicklungen durch Auswertung der 0x22/23-CAN-Kommandos von Märklin® ersetzt werden.
Genutzte Datenbytes	D0 - D3: Loc-ID 0x00001814
	D4 - D5: Modulnummer (BUS 1 1...31, BUS 2 32...62, BUS 3 63...93)
	D6: Kontaktnummer (1...16)
	D7: Stellung
Nachricht	Modul 16, Kontakt 2 hat Stellung 1
	00 16 2B 17 08 00 00 18 14 00 10 02 01
Antwort	-/-

16 Post-Code

Jedes Modul besitzt eine Dreifarb-LED zur Anzeige des Betriebsstatus. Dies ist notwendig, da die Module ansonsten ohne Bus-Verbindungen keine Möglichkeiten haben zu sagen "wie es ihnen gerade geht". Ähnlich dem Post-Code bei den PC, wo über Töne beim Booten die einzelnen Schritte bestätigt oder Fehler akustisch ausgegeben wurden, habe ich mir einen Licht-Code für die Dreifarb-LED einfallen lassen.

Die LED-Anzeige wird mit 500 ms getaktet und ist je Botschaft 7 s lang; d.h., dass im Grundsatz 6 Blinkschematas zu je einer der drei Farben ROT, ORANGE und GRÜN möglich sind, Mischungen mal ausgenommen. Die Farbe der LED sind drei Klassen von Botschaften resp. Stati zugeordnet:

ROT: Fehler im Modul *ORANGE: Konfiguration des Moduls* *GRÜN: Bestätigung von Prozessen*

Unregelmäßiges Aufflackern der orangenen LED-Farbe bei ansonsten grüner LED zeigt Datentrain auf dem CAN-Bus an bzw. während des Upgrades aus dem externen EEPROM entsprechende Schreib-/Lesezugriffe. Damit ist erkennbar, ob der auf dem Modul implementierte CAN-Baustein Nachrichten verarbeitet.

Stand heute sind folgende Post-Codes implementiert:

Tabelle 16-1: LED-Signalbedeutung

Normalbetrieb (kein Blinken)												
												
Neuanmeldung PC erfolgreich (1x grün blinken)												
												
Neuanmeldung CS2/3 erfolgreich (2x grün blinken)												
												
Modulupdate erfolgreich (3x grün blinken)												
												
BT-Schreiben erfolgreich (4x grün blinken)												
												

FW-Upgrade erfolgreich (5x grün blinken)												
												
MCP-CAN-Baustein defekt oder nicht vorhanden (1x rot blinken)												
												
Versorgungsspannung zu niedrig (2x rot blinken)												
												
Firmware-Upgrade abgebrochen, da Fehler beim Parsen des neuen Programms. Das im Controller gespeicherte Programm wird wieder ausgeführt. (4x rot blinken)												
												
Firmware-Upgrade hat einen allgemeinen Systemfehler erzeugt. Das Modul muss über die ISP-Schnittstelle komplett neu aufgesetzt werden. (5x rot blinken)												
												
Interne Firmware wird gestartet (nur bei Modulen mit Upgrade-Funktion) (1x orange blinken)												
												
Firmware-Upgrade - Probedurchlauf wird durchgeführt (2x orange blinken)												
												
Firmware-Upgrade - Programmierung des internen EEPROM wird durchgeführt (3x orange blinken)												
												
Modul konfiguriert die interne Hardware (10x orange blinken)												
												

17 Quellenverzeichnis

Bei der Erstellung der Hard- und Software sowie der Dokumente und Texte zum MBCAN-Projekt sind u.a. folgende Fundstellen verwendet worden:

- [01] Märklin: „Kommunikationsprotokoll CAN transportierbar über Ethernet“, 2012
- [02] Märklin: „Einstieg in Märklin Digital“, 1994
- [03] Atmel: „ATMega644P - 8-bit AVR“, 2008
- [04] Microchip: „MCP2515 - Stand-Alone CAN Controller With SPI™ Interface“, 2003
- [05] Schmitt: „Mikrocomputertechnik mit Controllern der Atmel AVR-RISC-Familie“, 2008
- [06] Luis: „C/C++ - Das komplette Programmierwissen für Studium und Job“, 2004
- [07] CAN: „<http://www.kreatives-chaos.com/artikel/can>“
- [08] MM-Protokoll: „<http://home.snafu.de/mgrafe/Programme/Signalerzeugung - Froitzheim.pdf>“
- [09] Eagle: „<http://www.cadsoft.de>“
- [10] Microsoft: „<https://www.visualstudio.com/products/visual-studio-dev-essentials-vs>“
- [11] Atmel: „<http://www.atmel.com/microsite/atmel-studio/>“
- [12] Forum: „<http://www.mikrocontroller.net>“
- [13] Wolff: „HTML5 und CSS3 - Das umfassende Handbuch“, 2016
- [14] SelfHTML: „<https://wiki.selfhtml.org/wiki/CSS/Tutorials/Bildergalerie>“, 2018

18 Allgemeine Hinweise zum MBCAN-Projekt

Dies ist eine Dokumentation zu meiner privaten, nicht-kommerziellen Internetseite zum MBCAN-Projekt und dient ausschließlich der Darstellung meines Hobbys. Dazu gehören auch die dort zum Download angebotenen Dokumente und Softwarepakete.

Die Ausführungen beziehen sich auf die Internetpräsenz "mbcan.de".

Herausgeber:



Dr.-Ing. Thomas Wiesner
August-Bebel-Str. 7
59174 Kamen
eMail: info@mbcan.de

Haftungshinweis:

Die Inhalte der Internetpräsenz "mbcan.de", die Dokumentation, deren Inhalt sowie die Ideen dürfen nur für den privaten Gebrauch genutzt werden. Der Nachbau der gezeigten Schaltungen oder Anwendung der Software geschieht auf eigene Gefahr. Ich übernehme keine Haftung für eventuell durch die Anwendung entstandenen Sach-, Vermögens- oder Personenschäden.

Copyrights:

Die auf den Internetseiten und in den Dokumenten ggf. verwendeten jeweiligen Warenzeichen sind Eigentum der jeweiligen Unternehmen. Alle ggf. damit verbundenen Rechte werden durch mich uneingeschränkt anerkannt.

Soweit nicht durch Copyrights Dritter geschützt, liegt das Copyright bei allen hier gezeigten Texten, Bildern, Schaltungen und Quellcode bei Dr.-Ing. Thomas Wiesner. Eine Verwendung auf anderen Webseiten oder jegliche andere Veröffentlichung, auch auszugsweise, wird hiermit ausdrücklich untersagt.

Kamen, 20.10.2024

gez. Dr.-Ing. Thomas Wiesner