



MBCAN

Fernsteuerung einer  - Modelleisenbahn

Nicht-kommerzielles Projekt – Alle Angaben ohne Gewähr

Bedienungsanleitung

8-fach Rückmelder mbc-88

Massekontakt / Stromsensorik

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	11
8	Bauteileliste	14
9	Firmware	16
10	Steckverbindungen.....	17
11	Anschlussbeispiel.....	20
12	Inbetriebnahme.....	21
12.1	Modul in Betriebsbereitschaft versetzen.....	21
12.2	Modul konfigurieren	21
12.3	Modul mit der CS2® verbinden.....	22
12.4	Modul mit der CS3® verbinden.....	24
12.5	Modul als Automatik-Gerät an der CS2®	25
12.6	Modul als Automatik-Gerät an der CS3®	26
13	Modulbilder	28
14	Systemarray-Belegung für Eigenentwicklungen	30
14.1	Allgemeiner Bereich zum Modul.....	30
14.2	Modulspezifischer Bereich für Funktionsparameter	33
15	Befehlssatz zu den Modulen	35
15.1	PC_DH_H - Übertragung des Datenbanknamens mit 12 Bytes (1-4).....	37
15.2	PC_DH_M - Übertragung des Datenbanknamens mit 12 Bytes (5-8).....	38
15.3	PC_DH_L - Übertragung des Datenbanknamens mit 12 Bytes (9-12).....	39
15.4	PC_KENNER - Kenner und Identifier für die Module	40
15.5	PC_NEU - Neuanmeldeaufforderung.....	41
15.6	PC_NEU_DATA - Rückmeldung des PC an das Modul im Neuanmeldeprozess.....	42
15.7	MD_NEU_DATA - Meldung des Moduls während des Neuanmeldeprozesses	43

15.8	PC_RESET - Durchführen eines Hardware-Resets auf dem Modul	44
15.9	PC_MD_SEL - Modul aus Datenbank entfernen	45
15.10	PC_ALIVE - ALIVE-Abfrage.....	46
15.11	MD_ALIVE - ALIVE-Abfrage	47
15.12	PC_ARRAY - Zugriff Systemarray anfragen	48
15.13	MD_ARRAY - Zugriff Systemarray freigegeben.....	49
15.14	PC_ARRAY_DATA - Zugriff Systemarray freigegeben	50
15.15	MD_ARRAY_DATA - Antwort des Moduls auf Systemarray-Zugriff	51
15.16	PC_UPGRADE - Firmware-Upgrade	52
15.17	MD_UPGRADE - Firmware-Upgrade freigegeben	53
15.18	PC_UPGRADE_DATA - Schreibe Firmware.....	54
15.19	MD_UPGRADE_DATA - Antwort des Moduls auf Schreibe Firmware	55
15.20	PC_BOOT - Modul neu Booten	56
15.21	MD_S88 - Stellungsmeldung mbc-88 / mbc-90.....	57
16	Post-Code	58
17	Quellenverzeichnis	60
18	Allgemeine Hinweise zum MBCAN-Projekt	61

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	05.12.2019	Erste Version
1.1	01.10.2024	Redaktionelle Anpassungen
1.2	20.10.2024	Redaktionelle Anpassungen Befehlssatz

3.2 Firmware

FW: 1.00 - Basisversion

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 Umstellen der Weiche, selbständig bei der Leitstelle melden (Aktoren). Gleiches gilt für die Rückmelder (Sensoren).

Neben den Aktoren sind vor allem Sensoren für eine ferngesteuerte Modelleisenbahn notwendig. Diese sind entweder Masse-bezogene Rückmelder oder Stromsensoren.

Im Folgenden finden Sie hier die Beschreibung eines 8-fach Rückmelders für Massekontakt oder Stromsensorik.

5 Funktion

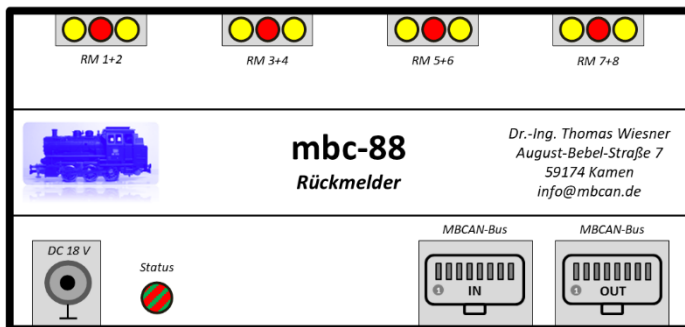


Abbildung 5-1:Modullabel

Dieses Modul stellt 8 Rückmeldekontakte für Massekontakt oder Stromsensorik bereit. Der Einsatz in digitalen 2-Leiter-DC-Anlagen von Trix® ist damit genauso möglich, wie die klassische Anwendung im Märklin® 3-Leiter-AC-System. Alle Kontakte sind untereinander galvanisch verbunden, zum MBCAN aber galvanisch entkoppelt. Die Bereitstellung des

Rückmeldepotenzials wird über die Gleisversorgung (Braun) sichergestellt. Je Eingang kann eine Haltezeit für das Ereignis als Filter programmiert werden. Der Diodenring zur Sicherung der Stromversorgung von Triebfahrzeugen in der AC-Anwendung mit Massekontakt ist integriert.

Gegenüber der CS2/3® tritt dieses Modul über den Terminaladapter als Rückmelder auf und kann entsprechend parametrierbar werden (Bus 1 bis 3, vgl. Bedienungsanleitung zum LinkS88® von Märklin®). Dabei repräsentiert es jeweils ein Halbmodul von acht Eingängen, je nach Parametrierung. Die Rückmeldungen werden auch an den MM®-Sniffer mbc-89 übertragen und dort im Standard-s88®-Bus entsprechend der Modulnummer eingereiht.

Steckverbinder	2x RJ45 MBCAN-Bus / Rückmeldeeingänge vom Gleis über Schraub-Steckverbinder
Stromversorgung	Steckernetzteil / RJ45-Versorgung
Statusanzeige	Betriebszustand und Traffic wird über Dreifarb-LED angezeigt
Modulzuordnung	Modulnummer im Sinne der klassischen s88®-Logik per PC-Software einstellbar
Automatik	CS2/3® vergibt den Automatik-Kenner / mbc-80 als Automatik-Gerät erforderlich
Features	Bei Anschluss an CS2/3® automatische Rückmeldung / Parametrierung, Firmwareupdate und Auslesung über PC möglich / Anzeige des Moduls in der GUI der CS2/3®

6 Schaltbild

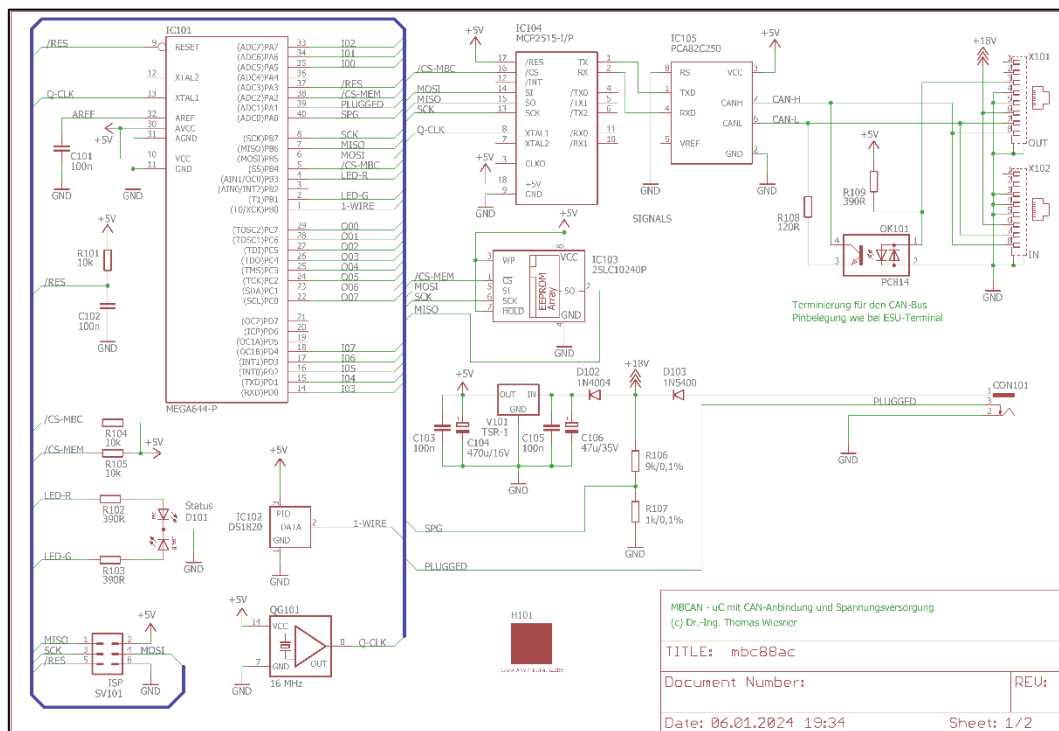


Abbildung 6-1: Prozessor-Schaltbild AD-Variante

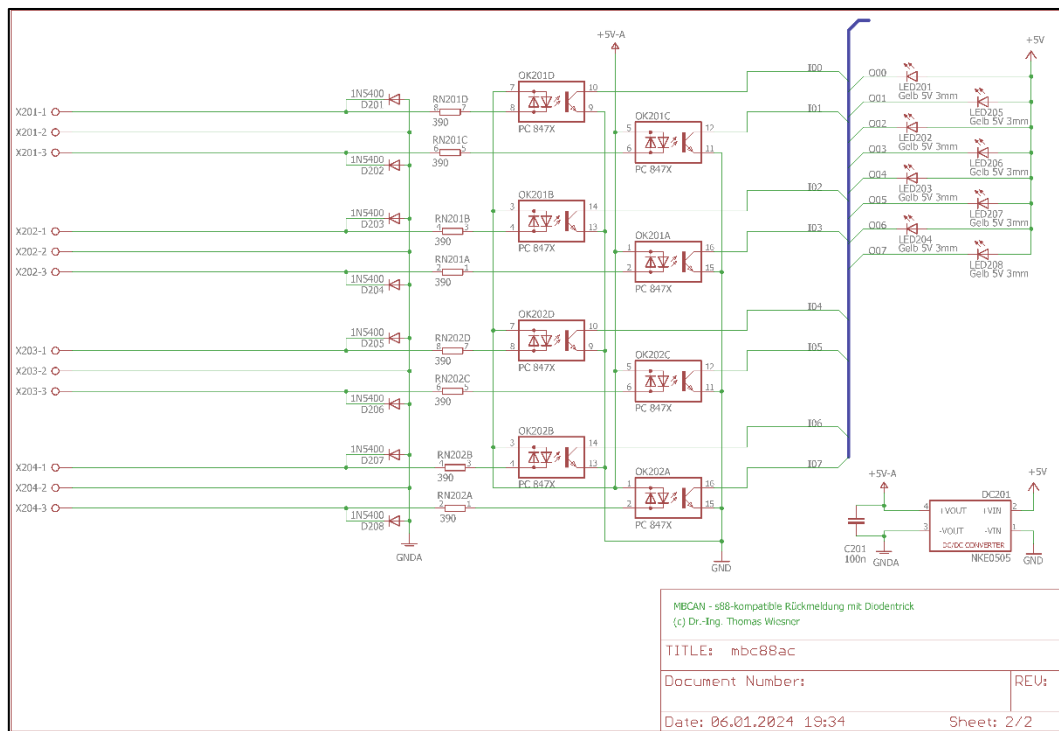


Abbildung 6-2: Sonderfunktionen AC-Variante

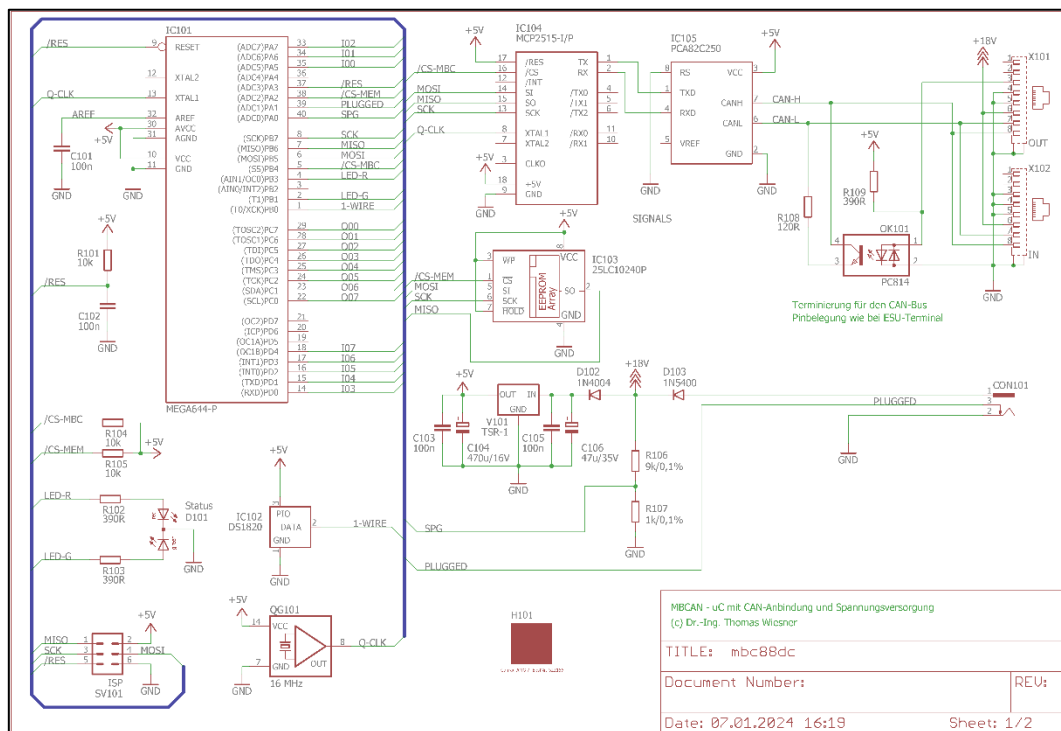


Abbildung 6-3: Prozessor-Schaltbild DC-Variante

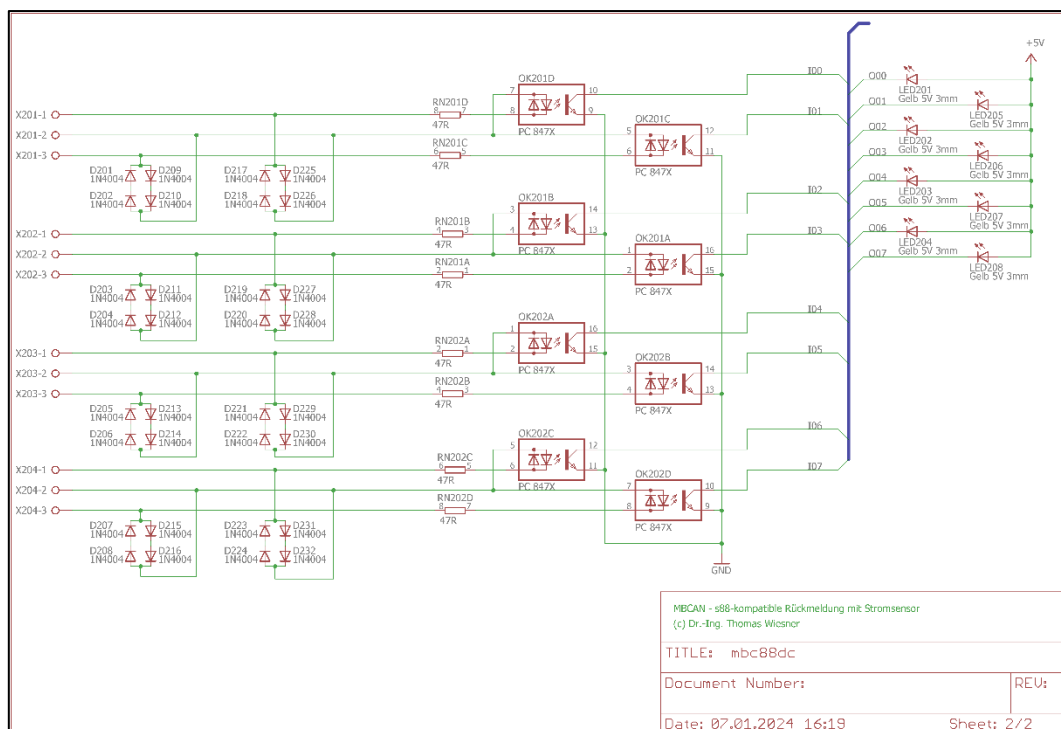


Abbildung 6-4: Sonderfunktionen DC-Variante

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 Sonderfunktion des Moduls besteht aus der Masse-bezogenen Besetztmeldung von Gleisabschnitten, alternativ über Stromsensorik.

Die Rückmeldung an die CS2/3® erfolgt über das CAN-Kommando 0x22. Auch eine Einzelabfrage durch die CS2/3® wird beantwortet. Des Weiteren wird für Automaten auch die Zeit in *ms* seit dem letzten Ereignis übermittelt. Das Modul stellt damit zusammen mit dem Terminaladapter mbc-80 einen vollwertigen Ersatz für die s88®-Module von Märklin® dar.

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).

Ansonsten ist bei der Bestückung noch zu überlegen, ob der angegebene leistungselektronische Spannungsregler oder ein herkömmlicher Spannungsregler des Typs 7805T verwendet wird. Im letzteren Fall ist allerdings Kühlkörper vorzusehen. Der Platz kann je nach Kühlkörpertyp etwas beengt sein, hier ist auszuprobieren, welcher Kühlkörpertyp der geeignete ist. Außerdem sind dann zwei 100nF-Blockkondensatoren nachzurüsten, für die keine Steckplätze vorgesehen sind. Dies soll Schwingneigungen entgegenwirken. Bei den Modulen des Herstellers werden grundsätzlich die etwas teureren leistungselektronischen Spannungsregler von TRACO verwendet, diese haben die Block-Kondensatoren bereits integriert.

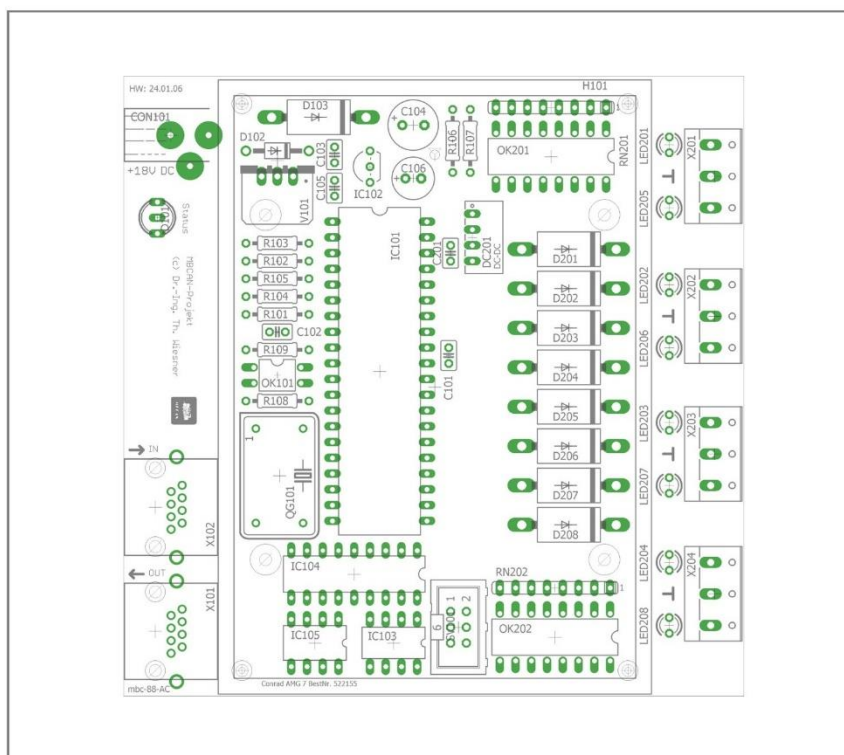


Abbildung 7-1: Bestückung Bauteilnummern AC-Variante





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.

Tabelle 8-1: Stückliste AC-Variante

Part	Value	Lieferant	Bestellnummer	Anzahl
C101 - C103, C105, C201	100n	Reichelt	MKS02-63 100N	5
C104	470u/16V	Reichelt	RAD 470/16	1
C106	47u/35V	Reichelt	RAD 47/35	1
D101	DUOLED R/G 5MM	Reichelt	LED 5 RG-3	1
D102	1N4004	Reichelt	1N 4004	1
D103, D201-D208	1N5400	Reichelt	1N 5400	9
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
V101	TSR 1-2450	Reichelt	TSR 1-2450	1
DC201	NKE0505	Reichelt	NKE0505SC	1
LED201 - LED208	LED 3mm Y 5V	Reichelt	LED 3MM 5V GE	8
QG101	QG5860	Reichelt	OSZI 16,000000	1
OK101	PC814	Reichelt	LTV 814	1
OK201, OK202	PC 847X	Reichelt	LTV 844	2
R101, R104, R105	10k	Reichelt	METALL 10,0K	3
R108	120R	Reichelt	METALL 120	1
R102, R103, R109	390R	Reichelt	METALL 390	3
R106, R108	9k/0,1%	Reichelt	MPR 9,10K	2
R107, R109	1k/0,1%	Reichelt	MPR 1,10K	2
RN201, RN202	390	Reichelt	SIL 8-4 390	2
RN203, RN204	4k7	Reichelt	SIL 8-4 4,7K	2
CON101	Power Jack	Reichelt	LUM NEB 21R	1
J101, J102	MBCAN	Reichelt	CAT5 T1U 2.8N4N	2
X101	AKL 101-02	Reichelt	AKL 101-02	1
SV101	ISP	Reichelt	WSL 6G	1
X201 - X204	AKL 101-03	Reichelt	AKL 101-03	4
ICS-8pol	Socket 8	Reichelt	GS 8P	2
ICS-16pol	Socket 16	Reichelt	GS 16P	2
ICS-18pol	Socket 18	Reichelt	GS 18P	1
ICS-40pol	Socket 40	Reichelt	GS 40P	1
Platine	mbc_88.brd	PCBPOOL	mbc_88_ac.brd	1

Tabelle 8-2: Stückliste DC-Variante

Part	Value	Lieferant	Bestellnummer	Anzahl
C101 - C103, C105	100n	Reichelt	MKS02-63 100N	4
C104	470u/16V	Reichelt	RAD 470/16	1
C106	47u/35V	Reichelt	RAD 47/35	1
D101	DUOLED R/G 5MM	Reichelt	LED 5 RG-3	1
D102, D201-D232	1N4004	Reichelt	1N 4004	33
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
V101	TSR 1-2450	Reichelt	TSR 1-2450	1
LED201 - LED208	LED 3mm Y 5V	Reichelt	LED 3MM 5V GE	8
QG101	QG5860	Reichelt	OSZI 16,000000	1
OK101	PC814	Reichelt	LTV 814	1
OK201, OK202	PC 847X	Reichelt	LTV 844	2
R101, R104, R105	10k	Reichelt	METALL 10,0K	3
R108	120R	Reichelt	METALL 120	1
R102, R103, R109	390R	Reichelt	METALL 390	3
R106, R108	9k/0,1%	Reichelt	MPR 9,10K	2
R107, R109	1k/0,1%	Reichelt	MPR 1,10K	2
RN201, RN202	47R	Reichelt	SIL 8-4 47	2
CON101	Power Jack	Reichelt	LUM NEB 21R	1
J101, J102	MBCAN	Reichelt	CAT5 T1U 2.8N4N	2
X101	AKL 101-02	Reichelt	AKL 101-02	1
SV101	ISP	Reichelt	WSL 6G	1
X201 - X204	AKL 101-03	Reichelt	AKL 101-03	4
ICS-8pol	Socket 8	Reichelt	GS 8P	2
ICS-16pol	Socket 16	Reichelt	GS 16P	2
ICS-18pol	Socket 18	Reichelt	GS 18P	1
ICS-40pol	Socket 40	Reichelt	GS 40P	1
Platine	mbc_88.brd	PCBPOOL	mbc_88_dc.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.

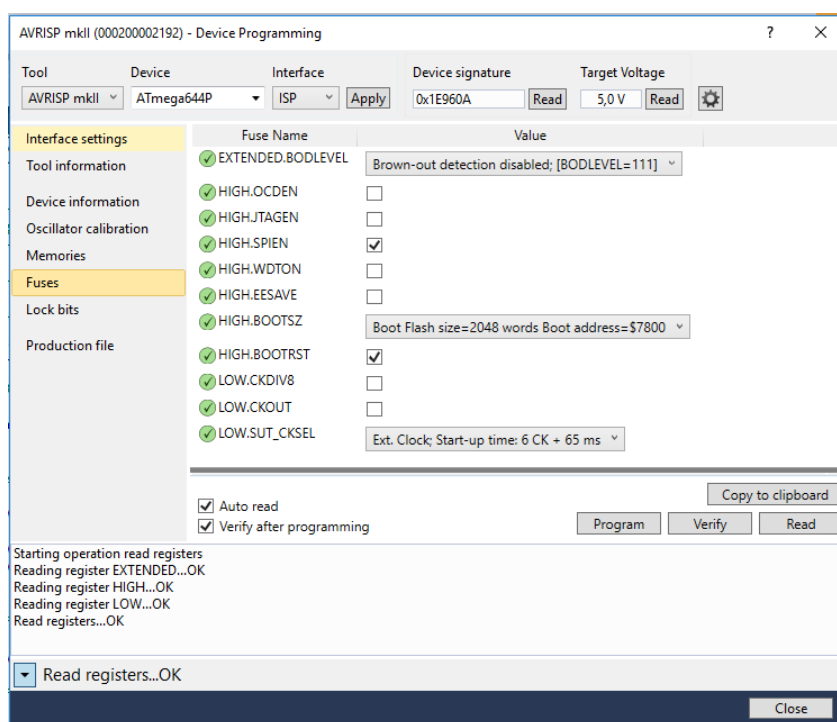


Abbildung 9-1: FUSES im AVR Studio

10 Steckverbindungen

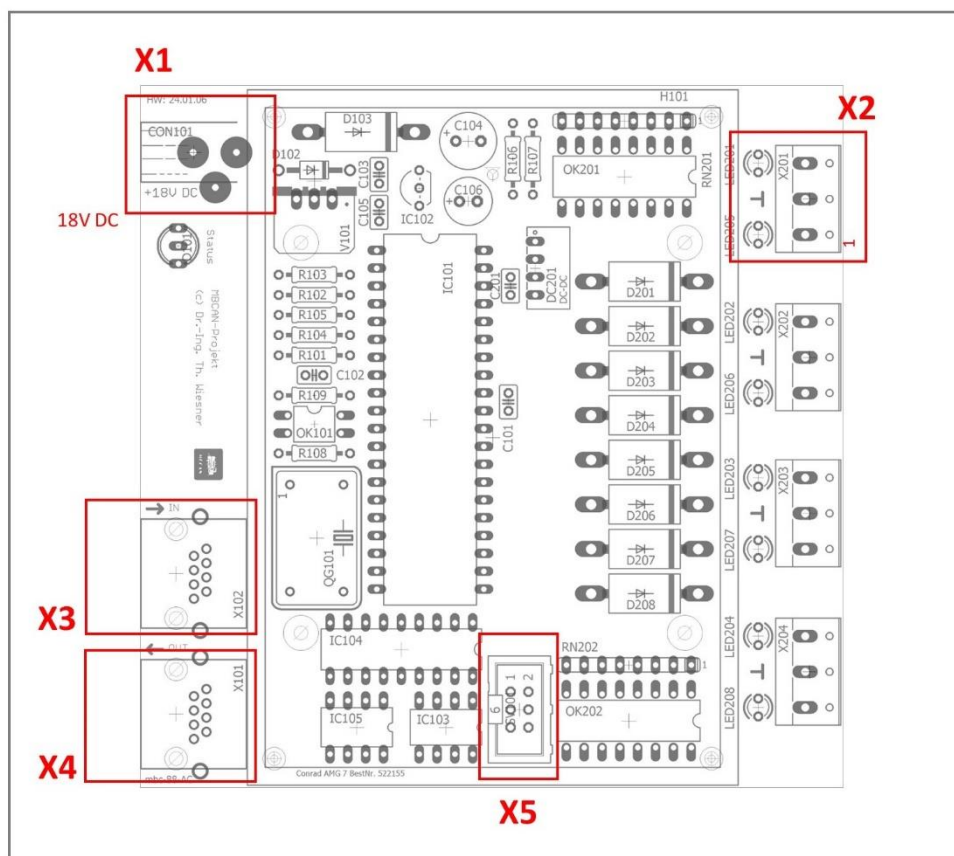


Abbildung 10-1: Steckverbinder AC-Variante

X1 dezentrale Spannungsversorgung

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

X2 Kontakteingänge

Hier kommen die Meldekontakte von den Gleisen an. Es wird über Masse geschaltet. Das Bezugspotenzial braucht nur einmal an einer der mit Braun gekennzeichneten Klemmen zugeführt werden. Voraussetzung ist in jedem Fall immer eine gemeinsame Masse im System. Die Pinbelegung lautet:

1 = Kontakt 2, 2 = Masse, 3 = Kontakt 1 (entsprechend 1 = Kontakt 4, 3 = Kontakt 3 an der zweiten Klemme usw.)

X3 MBCAN-IN

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

X4 MBCAN-OUT

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

X5 Optionale ISP-Schnittstelle

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

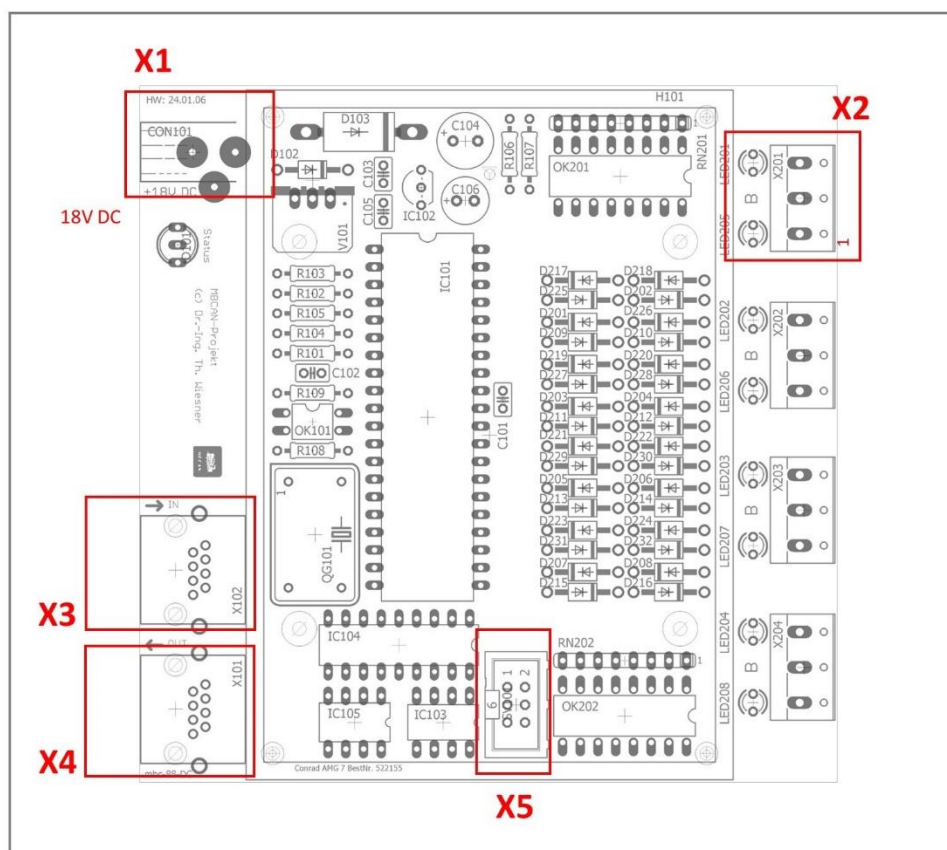


Abbildung 10-2: Steckverbinder DC-Variante

X1 dezentrale Spannungsversorgung

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

X2 Kontakteingänge

Hier kommen die Meldekontakte von den Gleisen an. Es wird der Stromfluss ausgewertet. Jeweils 2 Kontakte werden über eine zentralen Bahnstromeinspeisung versorgt.

Die Pinbelegung lautet:

1 = Kontakt 2, 2 = Bahnstrom, 3 = Kontakt 1 (entsprechend 1 = Kontakt 4, 3 = Kontakt 3 an der zweiten Klemme usw.)

X3 *MBCAN-IN*

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

X4 *MBCAN-OUT*

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

X5 *Optionale ISP-Schnittstelle*

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

12 Inbetriebnahme

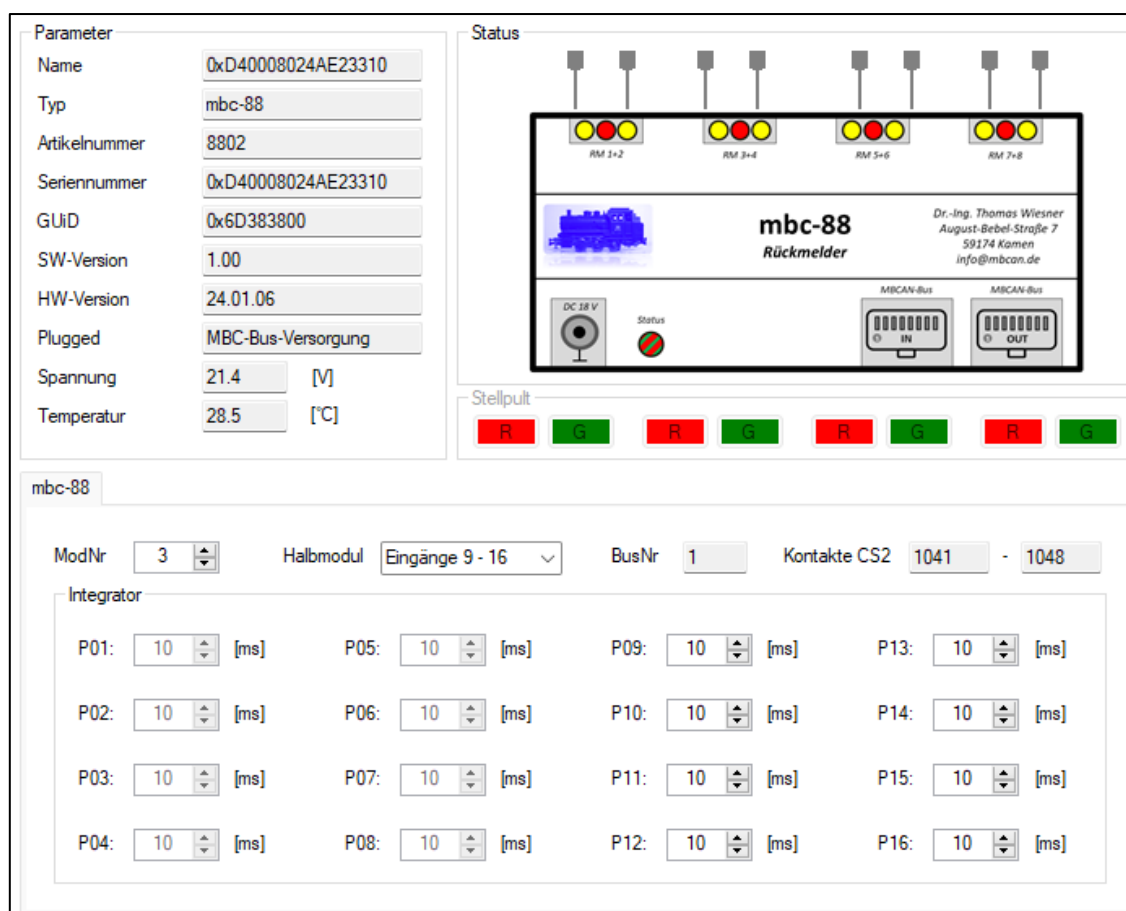
12.1 Modul in Betriebsbereitschaft versetzen

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 der in Abbildung 12-1 gezeigte Konfigurationsbereich.

Im gezeigten 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 Stellpult ist aktiviert.



The screenshot displays the configuration interface for the mbc-88 module. It is divided into several sections:

- Parameter:** A list of fields for configuring the module, including Name (0xD40008024AE23310), Typ (mbc-88), Artikelnummer (8802), Seriennummer (0xD40008024AE23310), GUID (0x6D383800), SW-Version (1.00), HW-Version (24.01.06), Plugged (MBC-Bus-Versorgung), Spannung (21.4 V), and Temperatur (28.5 °C).
- Status:** A visual representation of the module's physical components, including four RM (Relay Module) units (RM 1+2, RM 3+4, RM 5+6, RM 7+8) with yellow and red LEDs, a DC 18V input, a Status LED, and two MBCAN-bus ports (IN and OUT).
- Stellpult:** A row of eight buttons labeled R (Red) and G (Green), representing the relay states.
- mbc-88:** A section for integrator settings, including ModNr (3), Halbmodul (Eingänge 9 - 16), BusNr (1), and Kontakte CS2 (1041 - 1048). Below this is a grid of 16 integrator parameters (P01 to P16), each with a value of 10 and a unit of [ms].

Abbildung 12-1: Konfigurationsbereich

Konfiguriert werden kann die **<ModulNr>**, unter der das Modul seitens der CS2/3® oder einer PC-Anwendung erreichbar ist. Je Bus (1 bis 3) sind maximal 31 Module zulässig. D.h. Modulnummern bis einschließlich 31 gehören zum Bus 1, ab 32 bis 62 zu Bus 2, ab 63 zu Bus 3. Die Modulnummer des Bus 1 entspricht dem Modul in der s88®-Modulkette des meldenden mbc-89, respektive der Modulnummer korrespondierend zu den Rückmeldekontakten der CS2/3®. Die Modulnummern des Bus 2 und 3 werden durch das mbc-89 nicht ausgewertet.

Das Pull-Down-Menü **<Halbmodul>** legt fest, ob es sich bei dem Modul um die unteren acht Rückmeldekontakte oder die oberen acht Rückmeldekontakte eines vollwertigen s88®-Modul handelt. Entsprechend werden die Eingänge auch bei einer Meldung an die CS2/3® oder an das mbc-89 gemappt.

Die Felder **<BusNr>** und **<Kontakte CS2>** zeigen die korrespondierenden Einstellwerte für die CS2®. Bei der CS3® werden nur die Busnummer, die Modulnummer und die Kontaktnummer benötigt.

Die Integratorzeiten je Eingang in ms entsprechen der Länge des Belegtimpulses nach der letzten Aktivierung durch eine Achse des Zuges. Es entspricht in gewisser Weise einer Entprellung des Eingangs. Einstellmöglichkeiten bis 2500 ms sind möglich (in Abbildung 12-1 für Eingang P09 auf 2 s gesetzt). Je nach ausgewähltem Halbmodul werden die Eingabemöglichkeiten freigegeben. Dies ist nur eine programmtechnische Feinheit des VB-Programms um identische Masken für mbc-88 und mbc-90 verwenden zu können. Die eingestellten Werte werden dann entsprechend gemappt, wenn ein anderes Halbmodul ausgewählt wird.

Wird ein Wert 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 CS2® verbinden

Ist das Modul über den Terminaladapter mbc-80 mit der CS2® verbunden, meldet es sich über seine GUID an und ist sowohl im **<Info>-Bereich** als auch im **<Info>-Konfigurationsbereich** mit den Stammparametern anzeigbar.

Im **<Info>-Bereich** werden die Artikelnummer, die Firmwareversion in der Märklin®-Definition, die aktuelle Spannung und die Gehäuseinnentemperatur des Moduls angezeigt, letztere dynamisch (siehe Abbildung 12-2: <Info>-Bereich).

Im **<Info>-Konfigurationsbereich** werden die Seriennummer, die Firm- und die Hardwareversion angezeigt (siehe Abbildung 12-3: <Info>-Konfigurationsbereich).

Der Button **<CAN-Test>** und das UpDown-Feld **<Ergebnisse>** haben keine besondere Funktion und werden standardmäßig in dieser Registerkarte angezeigt.

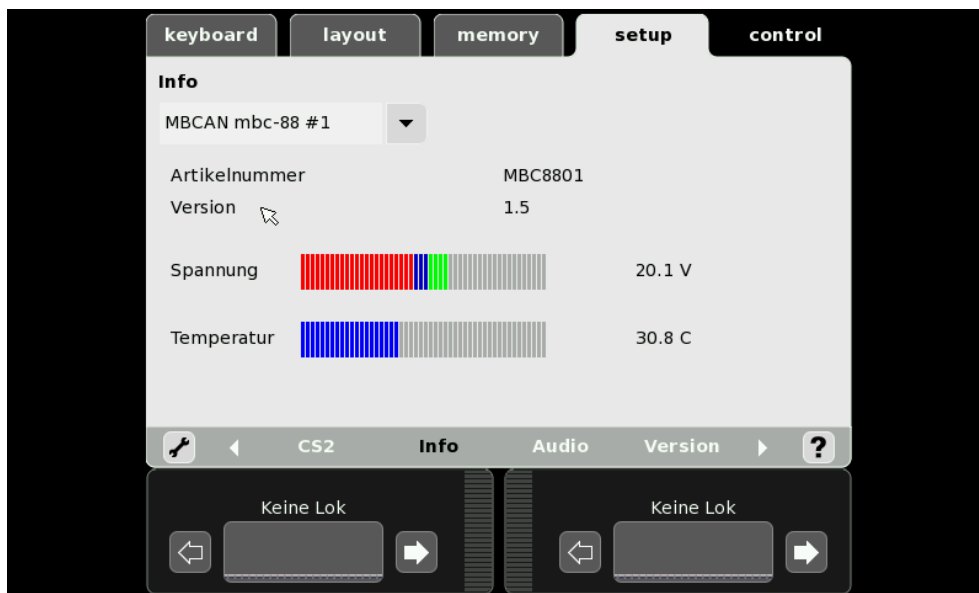


Abbildung 12-2: <Info>-Bereich

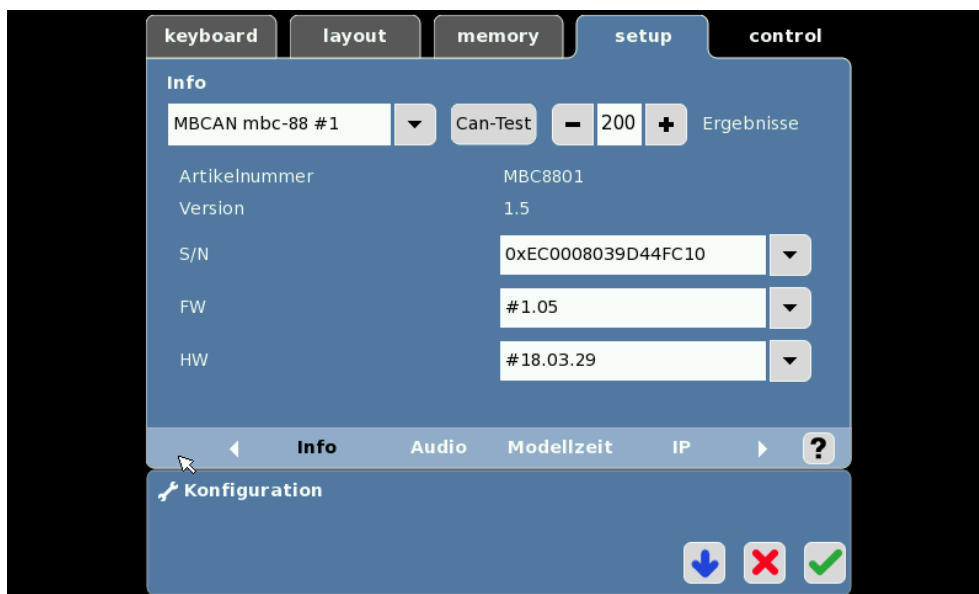


Abbildung 12-3: <Info>-Konfigurationsbereich

12.4 Modul mit der CS3® verbinden

Ist das Modul über den Terminaladapter 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 (siehe Abbildung 12-4: <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 Firm- und die Hardwareversion des Moduls angezeigt (siehe Abbildung 12-5: <Einstellungen>-Bereich).

Das Feld **<Name>** ermöglicht einen anderen als den Default-Namen den die CS3® vergibt. Unter diesem Namen ist das Modul dann bei der s88®-Konfiguration zu finden.

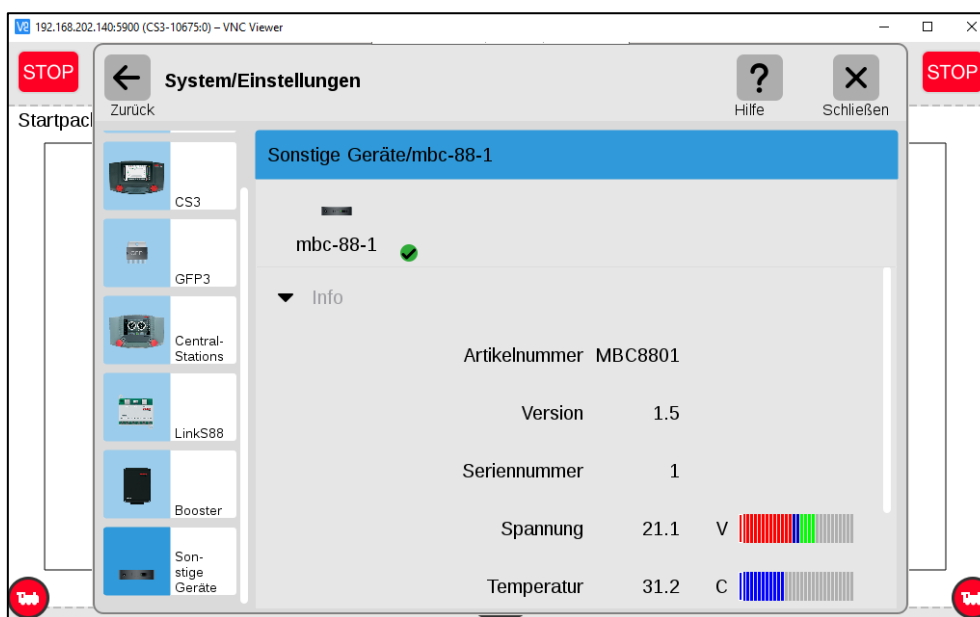


Abbildung 12-4: <Info>-Bereich eines mbc-88

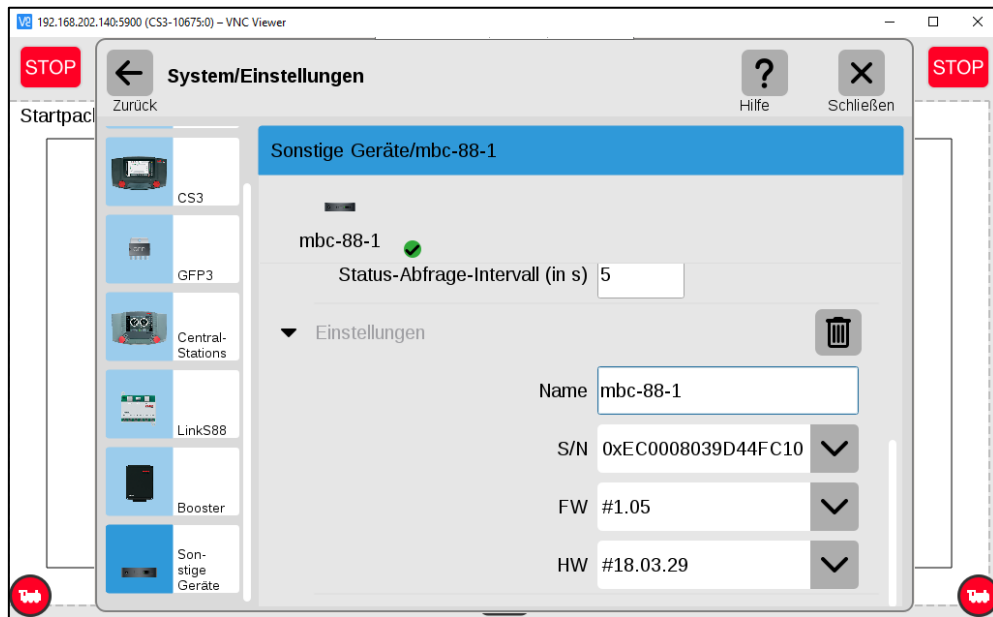


Abbildung 12-5: <Einstellungen>-Bereich eines mbc-88

12.5 Modul als Automatik-Gerät an der CS2®

Im Konfigurationsmenü der CS2® (siehe Abbildung 12-6) wird der Terminaladapter angezeigt. In der Regel steht beim Neustart der CS2® bei **<Typ (Name)>** „LinkS88 (1)“. Hier kann auch ein anderer Automatik-Kenner vergeben werden, dies sollte allerdings allein der CS2® vorbehalten bleiben.

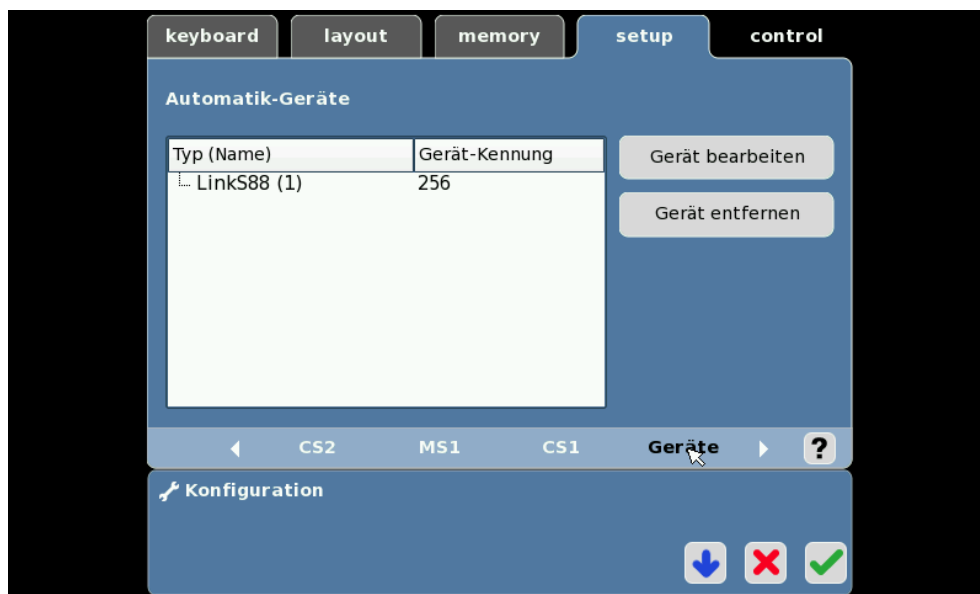


Abbildung 12-6: Der Terminaladapter als Automatik-Gerät

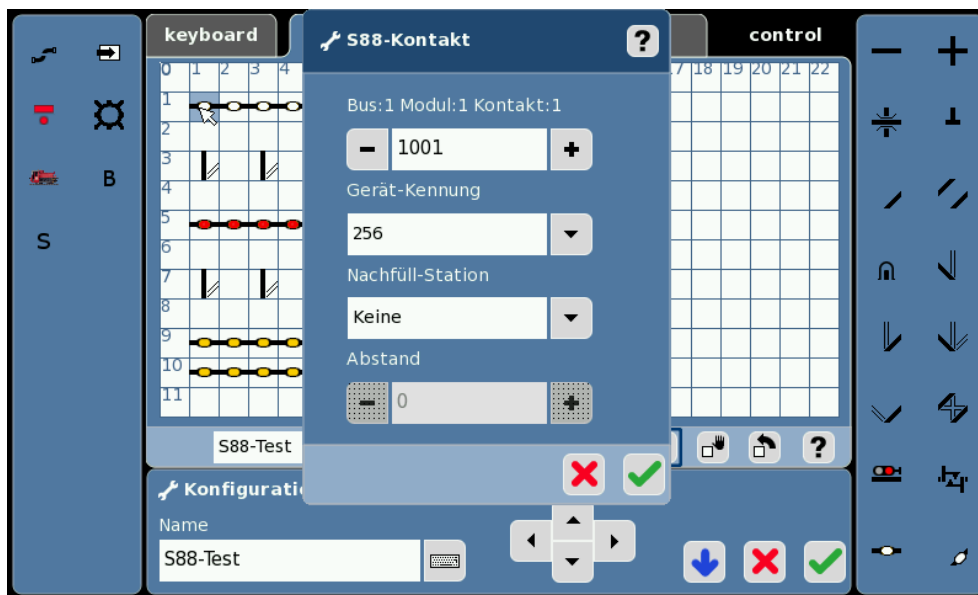


Abbildung 12-7: Zuordnung von Kontakten zum Automatik-Gerät "256"

In Abbildung 12-7 ist ein Beispiel zur Zuordnung eines Rückmelders im Layout der CS2® zum Automatik-Gerät „256“, also dem Terminaladapter mbc-80, dargestellt.

Im oberen Feld zu den **<S88-Kontakt>** wird die Kontaktnummer vergeben. Diese startet bei „1001“, die Modulnummer wird automatisch hochgezählt, wenn der <+>-Button gedrückt wird und die Kontaktzahl ein Vielfaches von 16 überschreitet. Über dem Eingabefeld ist als Merker die Busnummer des simulierten LinkS88® **<Bus: 1>**, die Modulnummer **<Modul: 1>** sowie die Kontaktnummer zum zugehörigen Modul **<Kontakt: 1>** angezeigt.

Die „1001“ signalisiert der CS2®, dass es sich um den simulierten Bus 1 eines LinkS88® handelt. Die einzutragende Kontaktnummern lässt sich wie folgt berechnen (analog für Bus 2 mit 2000 und Bus 3 als 3000 als Startwert):

$$\text{Kontaktnummer CS2®} = 1000 + (\text{mbc-88-Modulnummer} - 1) * 16 + \text{mbc-88-Kontaktnummer}$$

Die angezeigte **< mbc-88-Modulnummer >** entspricht dabei der Modul-Nummer die im Abschnitt 12.2 vergeben wurde. Bei Bus 2 ist diese jedoch um den Wert 31 zu verringern, bei Bus 3 um 62. Die **<mbc-88-Kontaktnummer>** ist entsprechend dem eingestellten **<Halbmodul>** auszuwählen, also 1-8 resp. 9-16.

12.6 Modul als Automatik-Gerät an der CS3®

Bei der CS3® funktioniert die Verwaltung der Rückmelder mbc-88 als Automatik-Gerät ähnlich. Das Automatik-Gerät mbc-80 wird als LinkS88® in der CS3® registriert, die untergeordneten mbc-88-Module antworten entsprechend, wenn der mbc-80 angesprochen wird.

Dazu ist im Pulldown-Menü im **<s88®-Artikelfenster>** der CS3® oben links im Konfigurationsfenster zu den **<Gleiskontakten>** das mbc-80 auszuwählen. Bitte beachten Sie, dass mbc-80 hier der **<Nickname>** des Links88® ist den Sie wie in Abschnitt 12.4 gezeigt vergeben haben.

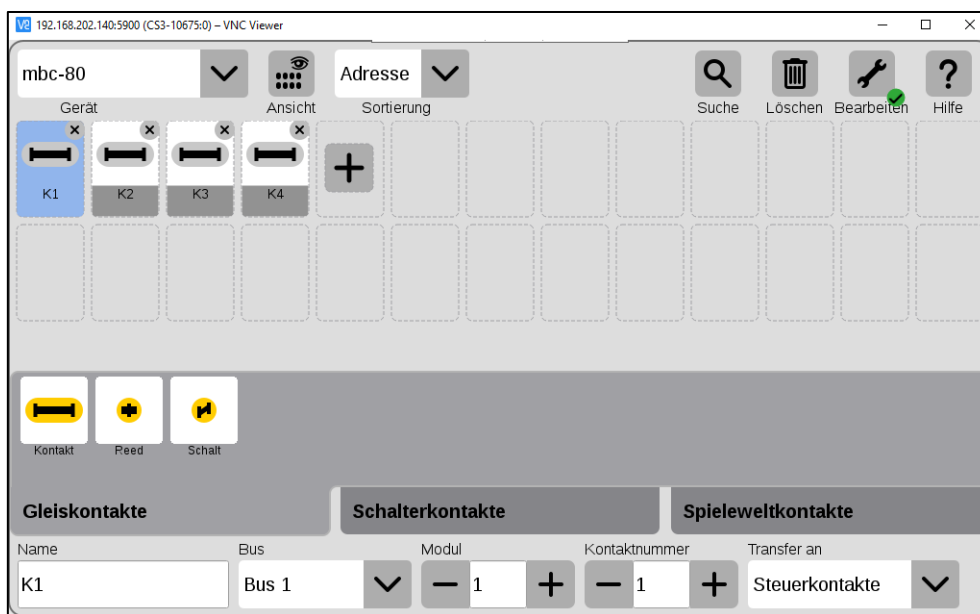


Abbildung 12-8: Zuordnung von Rückmeldekontakten

Neben dem **<Name>** des Kontaktes ist die Busnummer **<Bus>** auszuwählen (1 bis 3). Die Nummer des **<Modul>** entspricht der Modulnummer, die Sie in Abschnitt 12.2 vergeben haben. Bei Bus 2 ist diese jedoch um den Wert 31 zu verringern, bei Bus 3 um 62. Die **<mbc-88-Kontaktnummer>** ist entsprechend dem eingestellten **<Halbmodul>** auszuwählen, also 1-8 resp. 9-16.

Die **<Kontaktnummer>** ist entsprechend dem eingestellten **<Halbmodul>** auszuwählen, also 1-8 resp. 9-16. Die Einstellung **<Transfer an>** sollte auf Steuerkontakte eingestellt bleiben.

13 Modulbilder



Abbildung 13-1: Fertiges AC-Modul inkl. Gehäuse

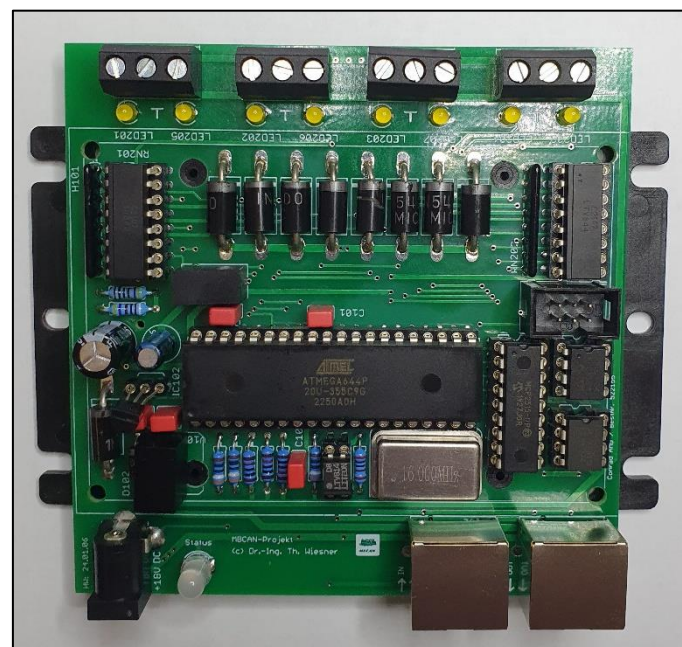


Abbildung 13-2: bestückte AC-Platine



Abbildung 13-3: Fertiges DC-Modul inkl. Gehäuse

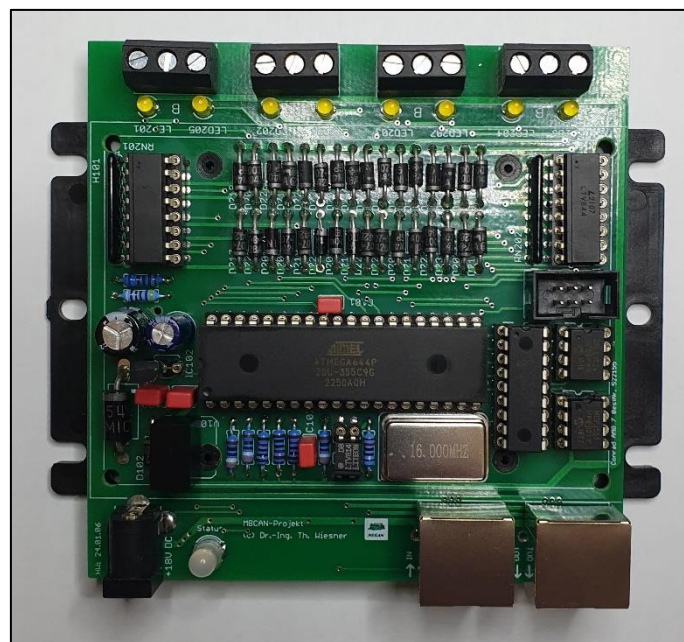


Abbildung 13-4: bestückte DC-Platine

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-Geraetekenennung
// MBC_S_KN Index Start CS2-Geraetekenennung
// MBC_KN_H CS2-Geraetekenennung HIGH
// MBC_KN_L CS2-Geraetekenennung 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. Diese liegen im GENPURP-Bereich des Systemarrays.

Neben der Moduladresse und der Kennzeichnung des Halbmoduls (Eingänge 1-8 oder 9-16 im Sinne der s88-Konvention) sind dies die Integratorkonstanten je Eingang für die Länge des Impulses nach dem letzten Trigger. Die Stellung der Eingänge kann ebenfalls ausgelesen werden.

Dargestellt ist die C-Schreibweise:

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

```
// Laenge modulspezifische Parameter
// Index Start modulspezifische Parameter
// S88Addr HIGH
// S88Addr LOW
// Halbmodul
// Eingang 8-1

#define MBC_88_L_S88          4
#define MBC_88_S_S88          MBC_S_GENPURP
#define MBC_88_S88_H          sys_array[MBC_88_S_S88]
#define MBC_88_S88_L          sys_array[MBC_88_S_S88 + 1]
#define MBC_88_S88_MOD        sys_array[MBC_88_S_S88 + 2]
#define MBC_88_S88_RM         sys_array[MBC_88_S_S88 + 3]

// Anzahl der S88PINS
// Index Start S88PINS
// Integrationszeit s88

#define MBC_88_L_INTS88        8
#define MBC_88_S_INTS88        (MBC_88_L_S88 + MBC_88_S_S88)
#define MBC_88_INTS88         sys_array[MBC_88_S_INTS88]
```

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_88_S88Addr          0x0A          // ID Block Adresse
#define EE_88_S88Timing        0x0B          // ID Block Timing
#define EE_88_S88Mod           0x0C          // ID Block Halbmodul
```

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 Identifizier
// PC_NEU                 PC - Neuanmeldungsanforderung des PC
// PC_NEU_DATA            PC - Neuanmeldungskanal des PC
// MD_NEU_DATA            MD - Neuanmeldungskanal 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	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. 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.
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 des PC an das Modul im Neuanmeldeprozess

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 freigegeben

Befehl	MD_ARRAY
Sender	Modul
Loc-ID	0x0000180C
Funktion	Zugriff Systemarray freigegeben
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	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. Beispiel: D4 = 0 -> Lesen, 1 -> Schreiben D5 + D6 = Systemarray-Index D7 = zu schreibender Wert, bei lesendem Zugriff irrelevant 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.
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 Hardwarereset 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