



# MBCAN

Fernsteuerung einer  - Modelleisenbahn

*Nicht-kommerzielles Projekt – Alle Angaben ohne Gewähr*

Bedienungsanleitung

**mf<sup>®</sup>-Sniffer mbc-82**

*Version 1.7*

*HW 21.03.01, FW 2.16, 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 .....	9
7	Bestückung .....	10
8	Bauteileliste.....	12
9	Firmware .....	14
10	Steckverbindungen.....	15
11	Anschlussbeispiele.....	17
12	Inbetriebnahme.....	19
12.1	Modul in Betriebsbereitschaft versetzen.....	19
12.2	Modul konfigurieren .....	19
12.3	Modul mit der CS2® verbinden.....	21
12.4	Modul mit der CS3® verbinden.....	23
13	Modulbilder .....	25
14	Systemarray-Belegung für Eigenentwicklungen .....	26
14.1	Allgemeiner Bereich zum Modul.....	26
14.2	Modulspezifischer Bereich für Funktionsparameter .....	30
15	Befehlssatz zu den Modulen .....	31
15.1	PC_DH_H - Übertragung des Datenbanknamens mit 12 Bytes (1-4).....	33
15.2	PC_DH_M - Übertragung des Datenbanknamens mit 12 Bytes (5-8).....	34
15.3	PC_DH_L - Übertragung des Datenbanknamens mit 12 Bytes (9-12).....	35
15.4	PC_KENNER - Kenner und Identifier für die Module .....	36
15.5	PC_NEU - Neuanmeldeaufforderung .....	37
15.6	PC_NEU_DATA - Rückmeldung PC an Modul während des Neuanmeldeprozesses .....	38
15.7	MD_NEU_DATA - Meldung des Moduls während des Neuanmeldeprozesses .....	39
15.8	PC_RESET - Durchführen eines Hardware-Resets auf dem Modul .....	40
15.9	PC_MD_SEL - Modul aus Datenbank entfernen .....	41



15.10	PC_ALIVE - ALIVE-Abfrage.....	42
15.11	MD_ALIVE - ALIVE-Abfrage.....	43
15.12	PC_ARRAY - Zugriff Systemarray anfragen .....	44
15.13	MD_ARRAY - Zugriff Systemarray freigegeben.....	45
15.14	PC_ARRAY_DATA - Zugriff Systemarray freigegeben .....	46
15.15	MD_ARRAY_DATA - Antwort des Moduls auf Systemarray-Zugriff .....	47
15.16	PC_UPGRADE - Firmware-Upgrade .....	48
15.17	MD_UPGRADE - Firmware-Upgrade freigegeben .....	49
15.18	PC_UPGRADE_DATA - Schreibe Firmware.....	50
15.19	MD_UPGRADE_DATA - Antwort des Moduls auf Schreibe Firmware.....	51
15.20	PC_BOOT - Modul neu Booten .....	52
15.21	MD_S88 - Stellungsmeldung mbc-88 / mbc-90.....	53
16	Post-Code .....	54
17	Quellenverzeichnis .....	56
18	Allgemeine Hinweise zum MBCAN-Projekt .....	57

## 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	22.11.2020	Erste Version
1.1	19.10.2021	Befehlssatz und Systemarray ergänzt
1.2	22.11.2021	Kommunikationsfunktionen angepasst
1.3	05.02.2022	Textanpassungen
1.4	18.10.2022	Anpassung Warenkorb DS 18B20
1.5	22.02.2024	Redaktionelle Anpassungen
1.6	01.10.2024	Redaktionelle Anpassungen
1.7	20.10.2024	Redaktionelle Anpassungen Befehlssatz

#### 3.2 Firmware

FW: 1.00 - Basisversion
FW: 2.00 - Kommunikationsfunktionen angepasst
FW: 2.11 - CAN-Empfindlichkeit angepasst
FW: 2.16 - DS 18B20 adaptiert

## 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](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).

Das mfx®-Protokoll von Märklin® ist auf bi-direktionale Kommunikation auf dem Gleis ausgelegt. D.h., die Lokdecoder empfangen nicht nur Steuerbefehle, sondern senden Daten zum Zustand auf dem Gleis zurück. Die Zentrale horcht am Gleis auf diese Daten, wertet sie aus und nutzt sie für den Systemablauf.

## 5 Funktion

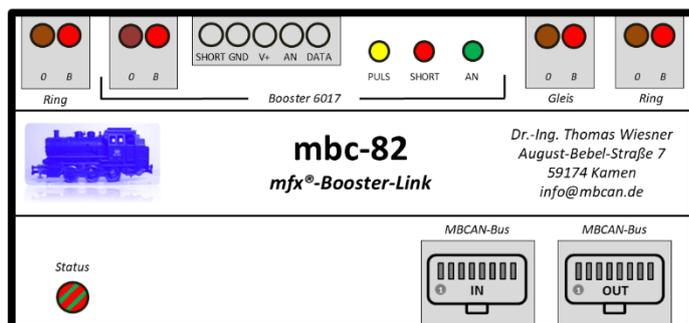


Abbildung 5-1: Modullabel

Dieses Modul stellt die Verbindung zwischen den älteren Boostern 6017<sup>®</sup> mit den modernen mfx<sup>®</sup>-Zentralen CS3<sup>®</sup> und MS2<sup>®</sup> bezüglich der Rückmeldung von Daten aus dem Gleis her.

In Kombination mit dem mbc-98 lassen sich alle Booster am Boosterbus um eine mfx<sup>®</sup>-Rückmeldung ergänzen. Im Stand-Alone-Betrieb – d.h. jeder Booster mit einem eigenen mbc-82 - kann jeder

einzelne Booster über die Not-Aus-Funktion inkl. Statusrückmeldung an die Zentralen ergänzt werden, was die Fehlerfindung im System erleichtert. Somit können auch diese historischen Bausteine der Märklinwelt in Zukunft bei Bedarf weitergenutzt werden.

Das Modul funktioniert auch an der CS2<sup>®</sup>; in dieser Zentrale ist bereits ein Boosterbus integriert. Die Verwendung des mbc-82 ist also nur dann erforderlich, wenn der Stand-Alone-Betrieb mit Einzelrückmeldung des Status gewünscht ist.

<b>Steckverbinder</b>	2x RJ45 MBCAN-Bus / Booster <sup>®</sup> 6017 <sup>®</sup> -Anschlüsse über Schraubklemmen (max. 3 A) und PINHEAD-Verbinder für den Boosterbus
<b>Stromversorgung</b>	Spannungsausgang des Boosters <sup>®</sup> 6017 <sup>®</sup> über DIL-Schalter am Booster/ RJ45-Versorgung
<b>Statusanzeige</b>	Betriebszustand und Traffic wird über Dreifarb-LED angezeigt
<b>Adresszuordnung</b>	-/-
<b>Betriebsarten</b>	Weitergabe der wichtigsten Eigenschaften (Not-Aus oder Kurzschluss) an die CS2/3 <sup>®</sup> resp. MS2 <sup>®</sup>
<b>Features</b>	Bei Anschluss an CS2/3 <sup>®</sup> resp. MS2 <sup>®</sup> automatische Rückmeldung / Parametrierung, Firmwareupdate und Auslesung über PC möglich / Anzeige des Moduls in der GUI der CS2/3 <sup>®</sup>

**Achtung:**

**Märklin® rät seit Einführung der CS3® von der Weiternutzung der Booster 6017® ab. Bis einschließlich der CS2® waren diese noch zulässig (eigene Schnittstelle) und in diversen Veröffentlichungen vom Anschluss her beschrieben (u.a. Märklin® Tipp 331 und dem Märklin® Buch 07420 „Grundlagen“). Dabei wurde auf die unbedingte Einhaltung der Trennstellen zwischen den Boosterkreisen entsprechend der einschlägigen Märklin®-Tipps hingewiesen!**

**Bei der Nutzung des mbc-82 resp. mbc-98 wird entsprechend des Abschnitts 18 wie bei allen anderen Modulen des MBCAN-Projektes keine Haftung übernommen.**

**Die Ringleitung aus den Zentralen bitte nur zur Verbindung der Module mbc-82 und mbc-98 untereinander verwenden und nicht als Boosterkreis nutzen. Dies soll die Zentralen soweit möglich vor Schäden bewahren.**

## 6 Schaltbild

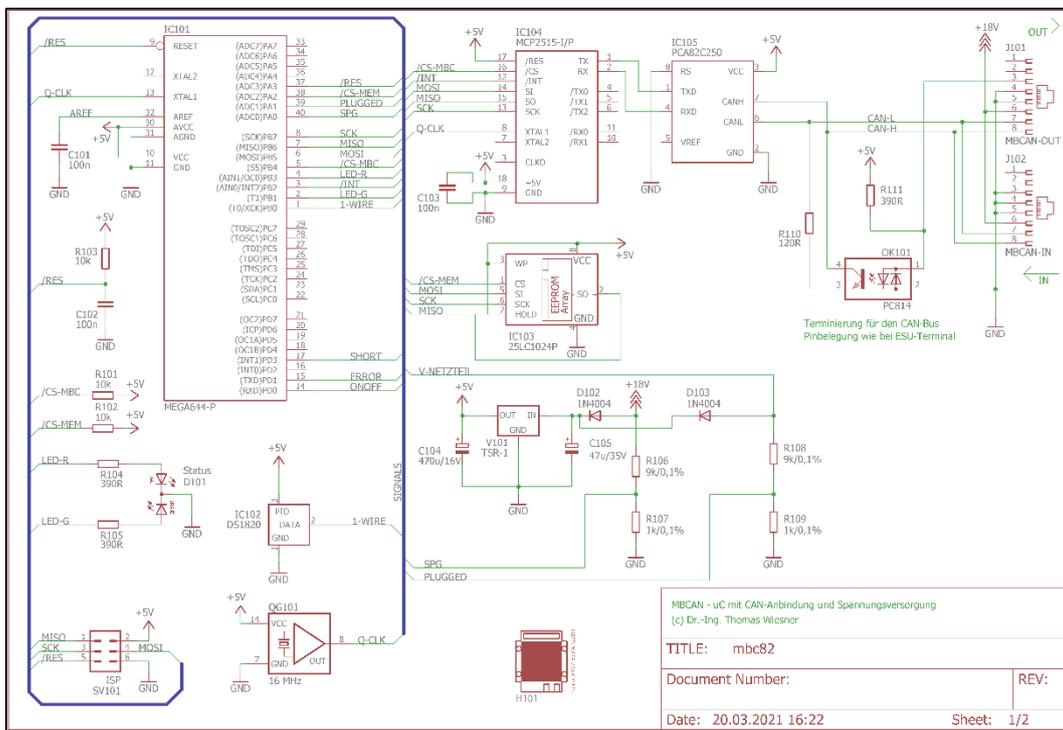


Abbildung 6-1: Schaltbild Seite 1

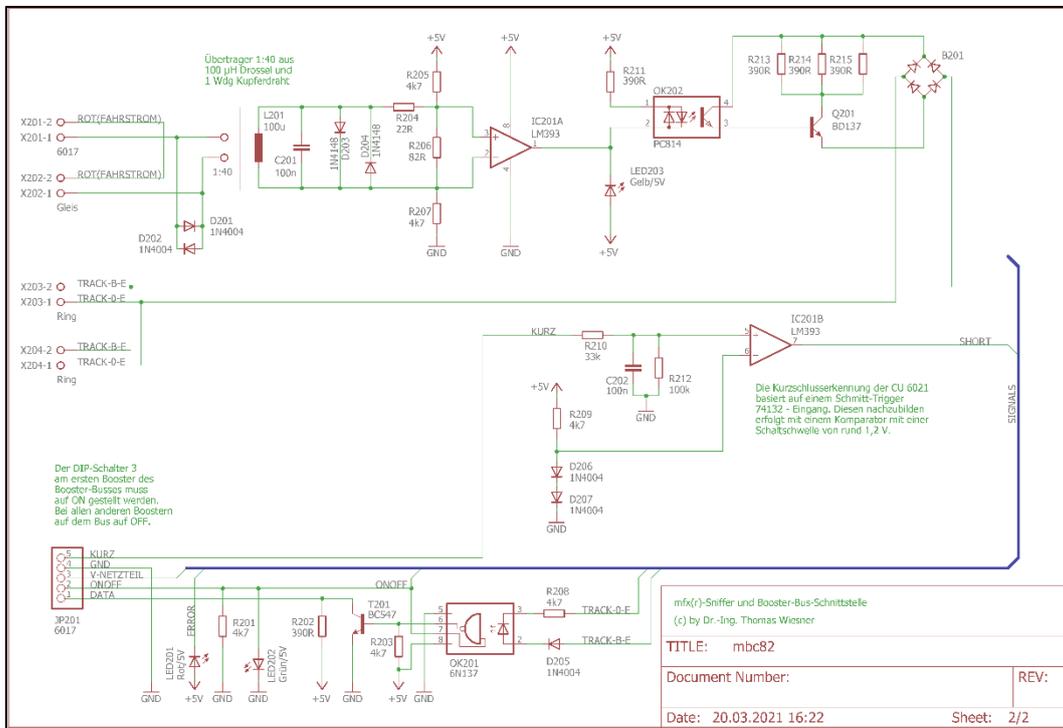


Abbildung 6-2: Schaltbild Seite 2

## 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).

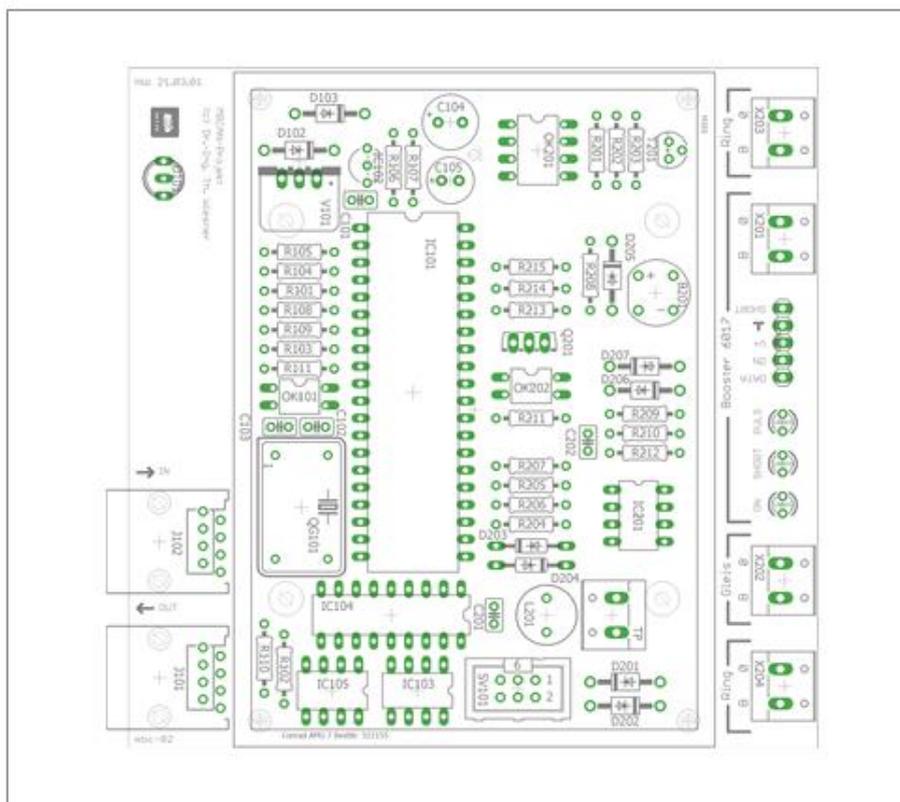


Abbildung 7-1: Bestückung Bauteilnummern



Abbildung 7-2: Übertrager

Der Übertrager bestehend aus L201 und den Schraubklemmen TP wird vor dem Einlöten aus einer Drossel 100  $\mu\text{H}$  und einem Stück lackiertem Kupferdraht mit 1  $\text{mm}^2$  Durchmesser und einer Länge von 6,5 cm selbst gewickelt, so dass ein Übersetzungsverhältnis von ca. 1,5:50 entsteht (die Drossel aus der Stückliste hat 50 Windungen).



## 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

Part	Value	Lieferant	Bestellnummer	Anzahl
B101	RB1A	Reichelt	B80C2000RUND	1
C101 - C103, C201, C202	100n	Reichelt	MKS02-63 100N	5
C104	470u/16V	Reichelt	RAD 470/16	1
C105	47u/35V	Reichelt	RAD 47/35	3
D101	DUOLED R/G 5MM	Reichelt	LED 5 RG-3	1
D102, D103, D201, D202, D205-D207	1N4004	Reichelt	1N 4004	7
D203, D204	1N4148	Reichelt	1N 4148	2
IC101	MEGA644-P	Reichelt	ATMEGA 644P-20PU	1
IC103	25LC1024	Reichelt	25LC1024-I/P	1
IC102	DS1820	Reichelt	DS 18S20	1
oder	DS1820	Reichelt	DS 18B20	1
IC104	MCP2515-I/P	Reichelt	MCP 2515-I/P	1
IC105	PCA82C250	Reichelt	MCP 2551-I/P	1
IC201	LM393	Reichelt	LM 393 P	1
L201	100u	Reichelt	L-07HCP 100µ	1
LED201	LED 3mm G 5V	Reichelt	LED 3MM 5V GN	1
LED202	LED 3mm R 5V	Reichelt	LED 3MM 5V RT	1
LED203	LED 3mm Ge 5V	Reichelt	LED 3MM 5V GE	1
OK101, OK202	PC814	Reichelt	LTV 814	2
OK201	6N137	Reichelt	6N 137	1
QG101	16 MHz	Reichelt	OSZI 16,000000	1
T201	BC547	Reichelt	BC 547C	1
Q201	BD137	Reichelt	BD 137	1
V101	TSR 1-2450	Reichelt	TSR 1-2450	1
R101, R102, R103	10k	Reichelt	METALL 10,0K	3
R110	120R	Reichelt	METALL 120	1
R104, R105, R111, R202, R211, R213 - R215	390R	Reichelt	METALL 390	8
R201, R203, R205, R207 - R209	4k7	Reichelt	METALL 4,70K	6
R204	22R	Reichelt	METALL 22,0	1
R206	82R	Reichelt	METALL 82,0	1
R210	33k	Reichelt	METALL 33,0K	1

---

R212	100k	Reichelt	METALL 100K	1
R106, R108	9k/0,1%	Reichelt	MPR 9,10K	2
R107, R109	1k/0,1%	Reichelt	MPR 1,10K	2
JP201	Pinhead 6017	Reichelt	SL 1X50G 2,54	1
J101, J102	MBCAN	Reichelt	MEBP 8-8S	2
TP	AKL 101-02	Reichelt	AKL 101-02	1
X201 - X204	AKL 230-02	Reichelt	AKL 230-02	4
X201 - X204	AKL 249-02	Reichelt	AKL 249-02	4
SV101	ISP	Reichelt	SL 2x10G 2,54	1
ICS-8pol	Socket 8	Reichelt	GS 8P	3
ICS-18pol	Socket 18	Reichelt	GS 18P	1
ICS-40pol	Socket 40	Reichelt	GS 40P	1
Platine	mbc_82.brd	PCBPOOL	mbc_82.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.

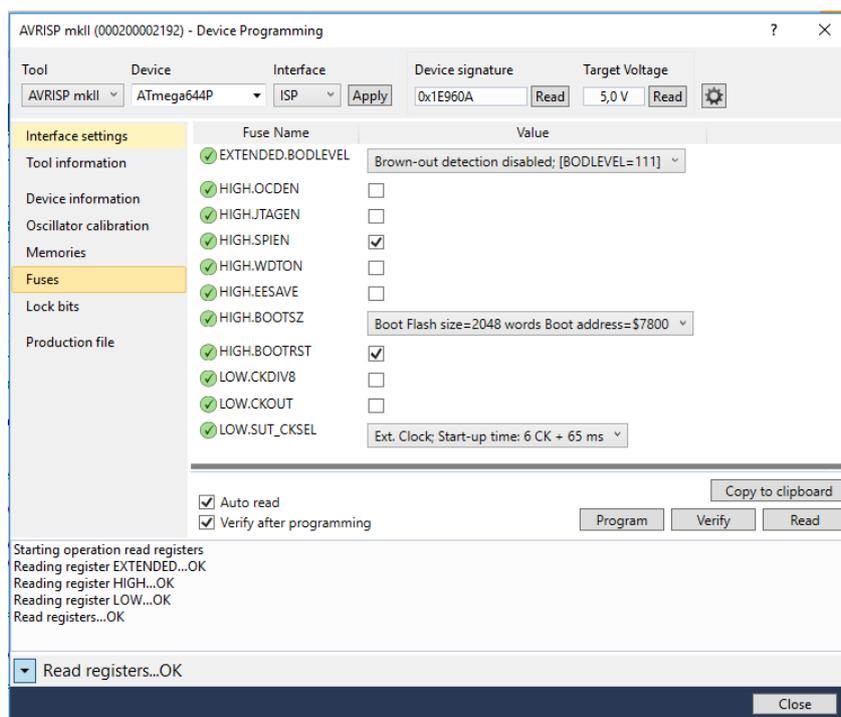


Abbildung 9-1: FUSES im AVR Studio

## 10 Steckverbindungen

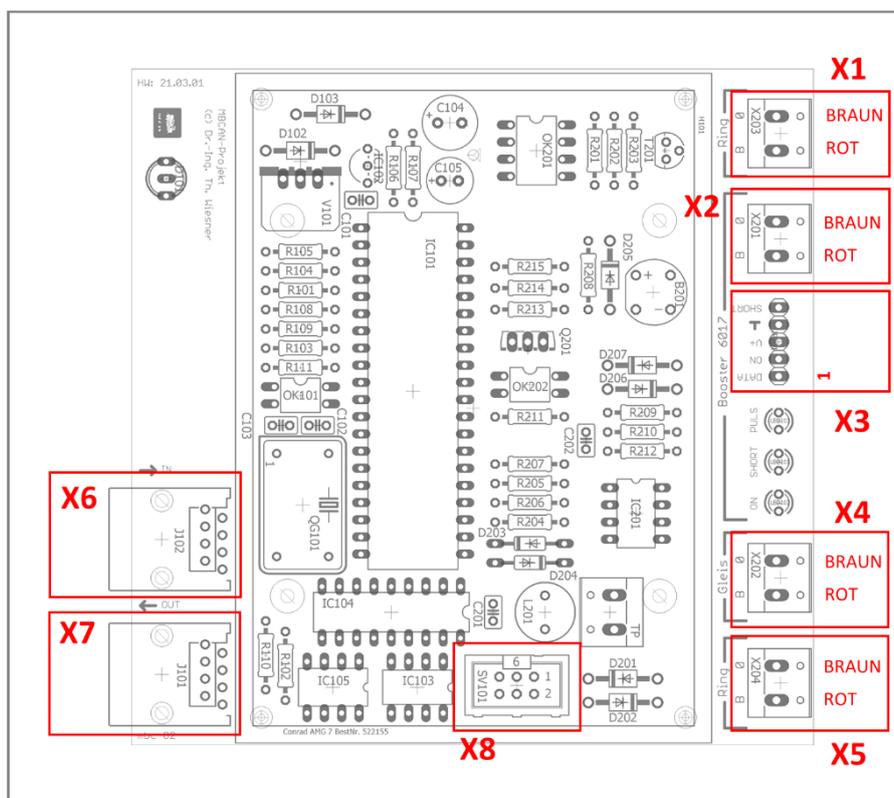


Abbildung 10-1: Steckverbinder

### **X1/5** Eingang/Ausgang von der Ringleitung der CS2/3<sup>®</sup> resp. MS2<sup>®</sup>

Diese Schraubklemmen werden genutzt, um die Leitungen der Zentrale für die Rückmeldung der mfx<sup>®</sup>-Signale an die Zentrale zu ermöglichen.

### **X2** Eingang vom Booster 6017<sup>®</sup>

Diese Schraubklemmen werden genutzt, um den Gleis Ausgang und den Versorgungsteil des Booster 6017<sup>®</sup> mit dem Modul zu verbinden.

### **X3** Boosterbus Booster 6017<sup>®</sup>

Diese Steckverbindung dient der Verknüpfung des Moduls mit dem Steuereingang des Boosters 6017<sup>®</sup>.

1 = DATA, 2 = ON/OFF, 3 = Spannungsversorgung, 4 = GND, 5 = SHORT

### **X4** Ausgang des Moduls zum Gleis

Diese Schraubklemmen werden genutzt, um das Modul mit dem Gleis zu verbinden.

**X6**    *MBCAN-IN*

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

**X7**    *MBCAN-OUT*

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

**X8**    *Optionale ISP-Schnittstelle*

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

## 11 Anschlussbeispiele

**Achtung:** Der Ausgang der CS3<sup>®</sup> resp. MS2<sup>®</sup> darf aus Sicherheitsgründen keinen eigenen Boosterkreis bilden und die Trennstellen sind gemäß Vorgaben von Märklin<sup>®</sup> auszuführen.

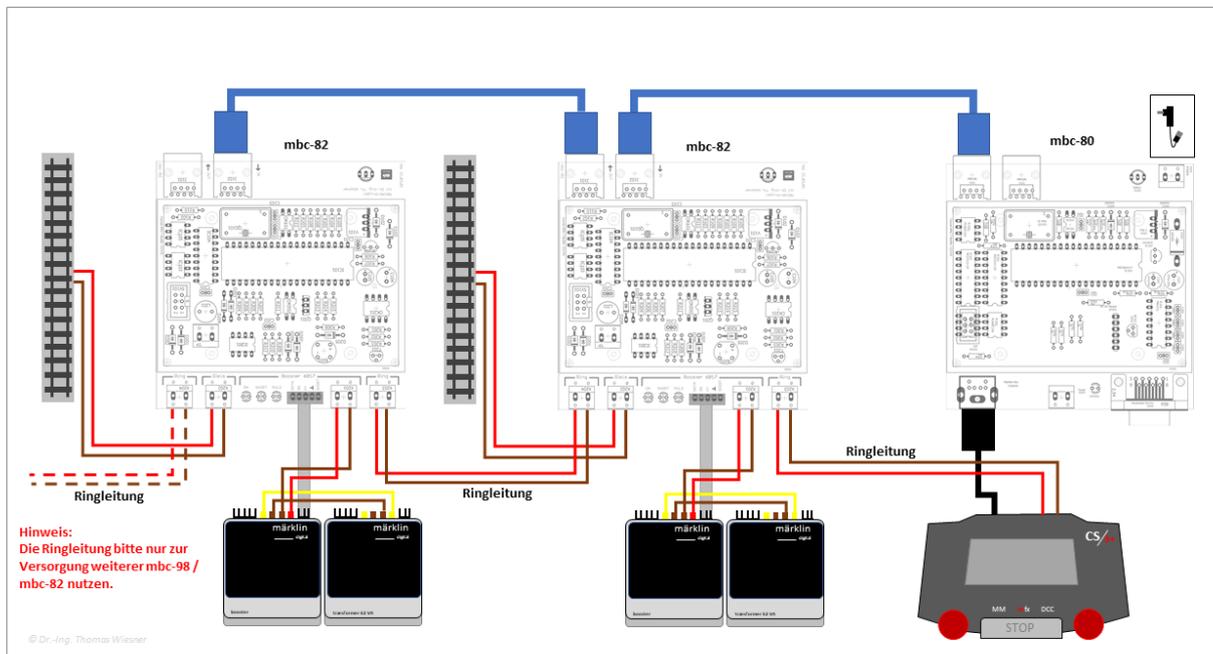


Abbildung 11-1: Anschluss Booster 6017<sup>®</sup> als Single-Modul an CS3<sup>®</sup>

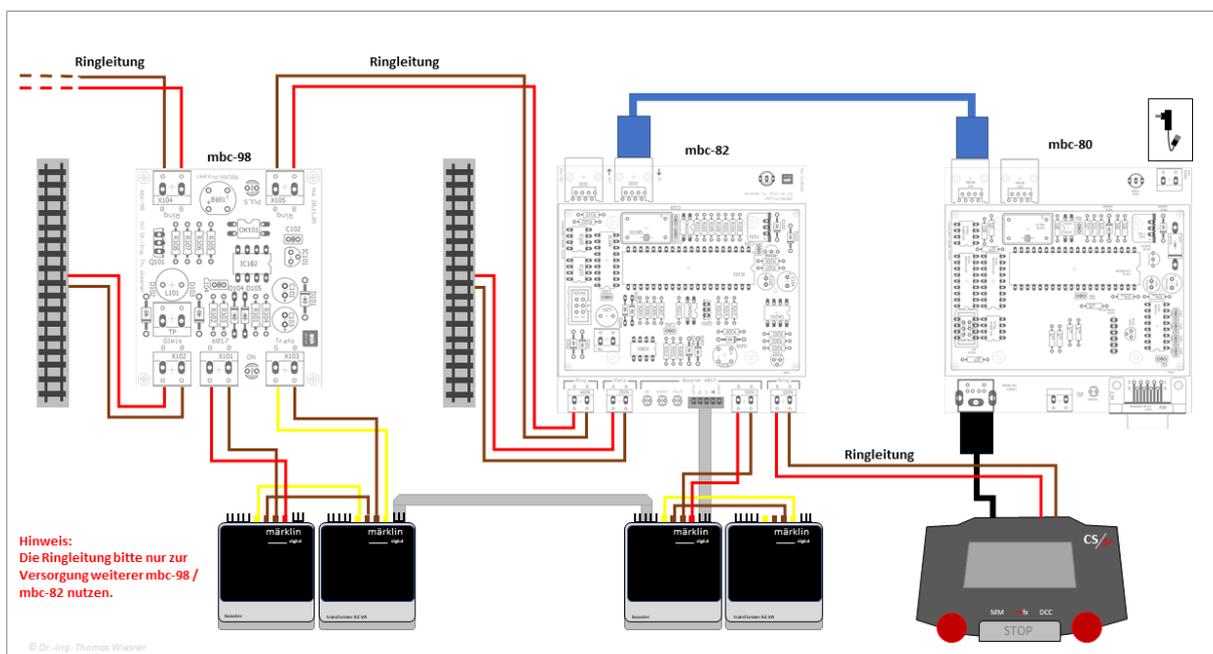


Abbildung 11-2: Anschluss Booster 6017<sup>®</sup> über Boosterbus und unter Verwendung des mbc-98 an CS3<sup>®</sup>

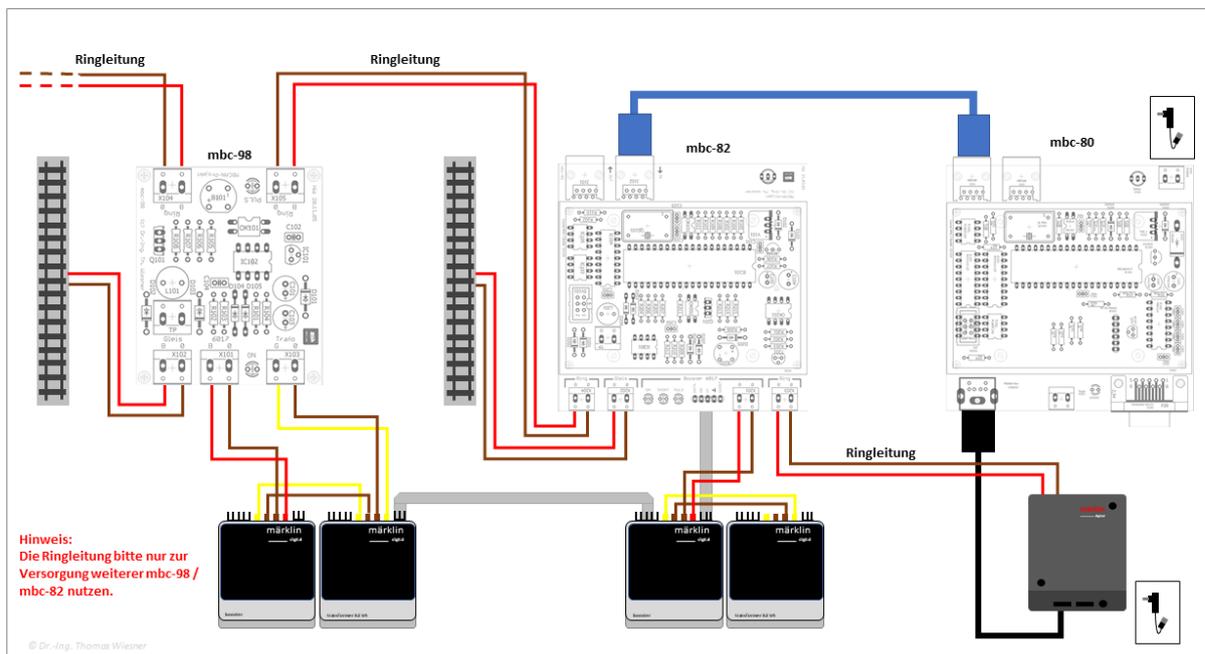


Abbildung 11-3: Anschluss Booster 6017<sup>®</sup> über Boosterbus und unter Verwendung des mbc-98 an MS2<sup>®</sup>

Der Anschluss nach Abbildung 11-1 erlaubt die Separierung jedes einzelnen Boosters mit Not-Aus und Kurzschlussanzeige. Diese Konstellation funktioniert auch an der CS2<sup>®</sup>, wenn der interne Boosterbus nicht verwendet wird. Eine gezielte Detektion des störungsbehafteten Boosters ist möglich.

Der Anschluss nach Abbildung 11-2 erlaubt die Nutzung des Boosterbusses mit zentraler Not-Aus und Kurzschlussanzeige an einer CS3<sup>®</sup>. Nur der erste Booster nutzt das mbc-82, alle weiteren werden über den Boosterbus und das mbc-98 zur Rückmeldung verbunden. Eine gezielte Detektion des störungsbehafteten Boosters ist nicht möglich.

Der Anschluss nach Abbildung 11-3 erlaubt die Nutzung des Boosterbusses mit zentraler Not-Aus und Kurzschlussanzeige und externer Ringleitung für die Rückmeldung an einer MS2<sup>®</sup>. Eine gezielte Detektion des störungsbehafteten Boosters ist nicht möglich, da die MS2<sup>®</sup> über keine GUI für die angeschlossenen Geräte verfügt.

In allen Fällen signalisieren die LEDs auf der Gleis-/Booster-Seite des Moduls den Betriebszustand. Die grüne LED zeigt an, wann der Booster aktiv geschaltet ist. Die rote LED signalisiert einen Kurzschluss; die grüne LED ist dann aus und der Booster inaktiv. Eine pulsierende gelbe LED zeigt mfx-Rückmeldungen auf dem Gleis an.

## 12 Inbetriebnahme

### 12.1 Modul in Betriebsbereitschaft versetzen

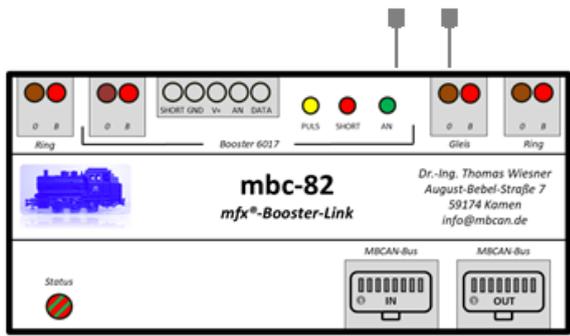


Abbildung 12-1: DIL 3 auf ON

Nach dem Start des Parametriercenters meldet sich das Modul automatisch an, sobald eine Verbindungsart zum MBCAN-Bus ausgewählt wurde (siehe Bedienungsanleitung zum Terminaladapter mbc-80). Bei Verwendung des mbc-82 ist immer der DIL-Schalter 3 für die externe Spannungsversorgung an dem Booster zu aktivieren, der mit dem mbc-82 direkt verbunden ist. Bei allen anderen Boostern am Boosterbus bitte auf OFF belassen. Dies gilt auch für die anderen DIL-Schalter.

### 12.2 Modul konfigurieren

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

<p><b>Parameter</b></p> <p>Name: <input type="text" value="0x95000803A4D97210"/></p> <p>Typ: <input type="text" value="mbc-82"/></p> <p>Artikelnummer: <input type="text" value="8201"/></p> <p>Seriennummer: <input type="text" value="0x95000803A4D97210"/></p> <p>GUID: <input type="text" value="0x6D383200"/></p> <p>SW-Version: <input type="text" value="2.16"/></p> <p>HW-Version: <input type="text" value="21.03.01"/></p> <p>Plugged: <input type="text" value="MBC-Bus-Versorgung"/></p> <p>Spannung: <input type="text" value="21.0"/> [V]</p> <p>Temperatur: <input type="text" value="22.0"/> [°C]</p>	<p><b>Status</b></p>  <p style="text-align: center;"><b>mbc-82</b> mfx®-Booster-Link</p> <p style="font-size: small; text-align: right;">Dr.-Ing. Thomas Wiesner August-Bebel-Straße 7 59174 Kamen info@mbcan.de</p> <p><b>Stellpult</b></p> <div style="display: flex; justify-content: space-around;"> <span style="background-color: red; color: white; padding: 2px 5px;">R</span> <span style="background-color: green; color: white; padding: 2px 5px;">G</span> <span style="background-color: red; color: white; padding: 2px 5px;">R</span> <span style="background-color: green; color: white; padding: 2px 5px;">G</span> <span style="background-color: red; color: white; padding: 2px 5px;">R</span> <span style="background-color: green; color: white; padding: 2px 5px;">G</span> <span style="background-color: red; color: white; padding: 2px 5px;">R</span> <span style="background-color: green; color: white; padding: 2px 5px;">G</span> </div>
---	--

**mbc-82**

**GFP-SW-Version**

Die aktuelle Softwareversion des GFP/Booster von Märklin, um die Update-Aufforderung von der CSx zu umgehen.

Nach dem <Update> das Modul bitte per <Reset> neu starten. Die neue Softwareversion wird dann in der CSx angezeigt.

GFP-SW-Version  .

**Latenz Kurzschlusserkennung**

Die Latenz bis zum Ansprechen der Kurzschlusserkennung kann hier in Schritten von 50 ms eingestellt werden.

Nach dem <Update> des Moduls bitte dieses aus der GUI der CS3 löschen. Der neue Latenzwert wird dann in der CS3 angezeigt.

Bei einer CS2 bitte diesen Wert ausschließlich in der CS2-GUI ändern und über den <Refresh>-Button aktualisieren.

Latenz [ms]

Abbildung 12-2: 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 Stellpult ist deaktiviert.

Konfiguriert werden muss die **<Software-Version>** zur Simulation eines Boosters mit Gleisformatprozessors (GFP). Die aktuelle Version wird angezeigt, wenn eine Aktualisierung des Moduls, welches mit einem roten Punkt versehen ist, angestoßen werden soll. In einem PopUp-Fenster zeigt die CS2/3<sup>®</sup> dann die neue Software-Version an. Diese bitte notieren, und im Parametriercenter eingeben. Danach das Modul resetten und parallel aus der GUI der CS2/3<sup>®</sup> löschen. Das Modul meldet sich mit den neuen Infos an.

Über die Vorgabe der **<Latenzzeit>** kann die Kurzschlussempfindlichkeit des Boosters eingestellt werden. In der Grundeinstellung sind dies 50 ms bei einem satten Kurzschluss. Dieser Wert hat sich als praktikabel gegenüber der Streuung der Empfindlichkeit der Märklin<sup>®</sup>-Booster aus den unterschiedlichen Produktionsjahren ergeben. Eine Erhöhung der Latenzzeit sollte aber nur in Ausnahmefällen vorgenommen werden. Den Wert bitte über **<Update>** an das Modul senden. Bei der CS3<sup>®</sup> das Modul danach aus der GUI der CS3<sup>®</sup> löschen (Papierkorbsymbol im Einstellungsbereich des Moduls). Das Modul meldet sich dann mit dem neuen Parameter an.

#### **Achtung bei der CS2<sup>®</sup>:**

Die CS2<sup>®</sup> kennt die Papierkorbfunktion leider nicht. D.h. der zuletzt in der CS2<sup>®</sup> gespeicherte Parameterwert wird wieder an das Modul nach dessen Neustart gemeldet und die Einstellung aus dem Parametriercenter überschrieben. Dies kann leider nicht umgangen werden. Daher ist dieser Wert bei Anwesenheit einer CS2<sup>®</sup> ausschließlich über die CS2<sup>®</sup> veränderbar und das Eingabefeld deaktiviert.

Wird einer der 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 beiden LED-Symbole zeigen den Status des Moduls an. Bitte vor der Interpretation einmal einen **<Refresh>** ausführen. Die linke LED zeigt an, ob der Booster sich im Kurzschluss befindet (ROT). Die rechte LED ob der Booster aktiv ist und das Gleis versorgt (GRÜN, vgl. Abbildung 12-2). Im Einschaltzustand des Moduls sollten beide LED GRAU sein. Bei einem Kurzschluss sollte die rechte LED ebenfalls GRAU anzeigen.

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 GUIID an und ist sowohl im **<Info>-Bereich** als auch im **<Info>-Konfigurationsbereich** mit den Stammparametern anzeigbar. Im **<Info>-Bereich** werden die Artikelnummer, die Version in der Märklin-Konvention, die aktuelle Spannung und die Gehäuseinnentemperatur des Moduls angezeigt, letztere dynamisch.



Abbildung 12-3: <Info>-Bereich allgemein

Hinzu kommt die Anzeige des Status des Moduls hinsichtlich **<ON Track>** und **<Kurzschluss>**. Bei **<ON Track>** ist die Anzeige hinter dem Menüpunkt GRÜN wenn der Booster aktiv ist (vgl. Abbildung 12-3). Bei einem **<Kurzschluss>** wird die Anzeige entsprechend ROT, **<ON Track>** wieder GRAU.

Im **<Info>-Konfigurationsbereich** werden Artikelnummer, die Seriennummer, die Firm- und die Hardwareversion, angezeigt.

Über die Vorgabe der Latenzzeit **<KS x 50 ms>** kann die Kurzschlussempfindlichkeit des Boosters eingestellt werden. In der Grundeinstellung sind dies 50 ms bei einem satten Kurzschluss. Dieser Wert hat sich als praktikabel gegenüber der Streuung der Empfindlichkeit der Märklin®-Booster aus den unterschiedlichen Produktionsjahren ergeben. Eine Erhöhung der Latenzzeit sollte aber nur in Ausnahmefällen vorgenommen werden.

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

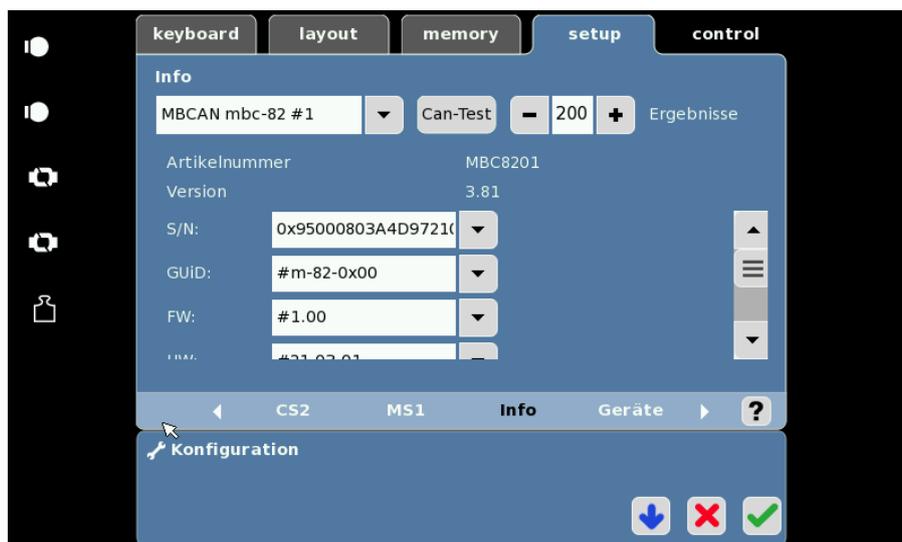


Abbildung 12-4: &lt;Info&gt;-Konfigurationsbereich

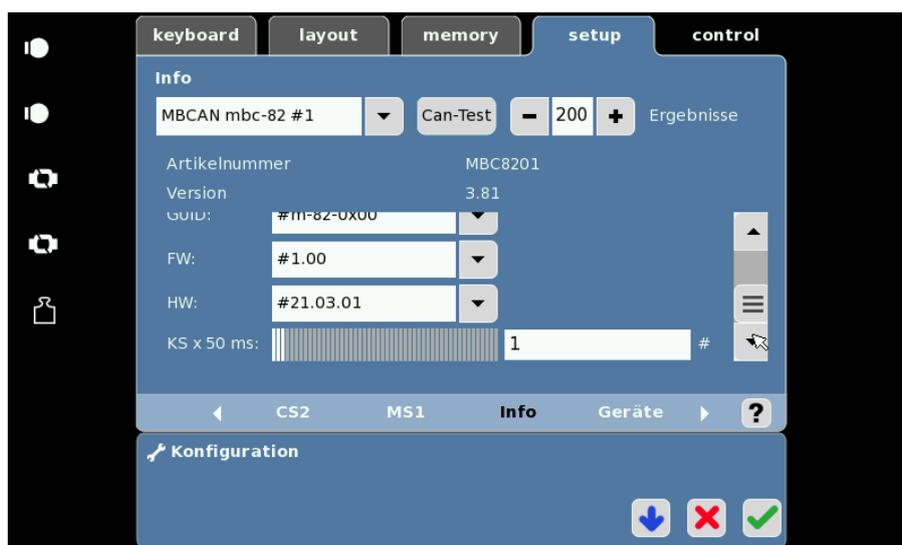


Abbildung 12-5: Latenzzeit Kurzschluss

## 12.4 Modul mit der CS3® verbinden

Ist das Modul über den Terminaladapter mbc-80 mit der CS3® verbunden, meldet es sich über seine GUIID an und ist im **<System/Einstellungen>-Bereich** unter **<Booster>** anzeigbar. Im **<Info>-Bereich** werden die Artikelnummer, die Versionsnummer, die interne Seriennummer, die aktuelle Spannung und die Gehäuseinnentemperatur des Moduls angezeigt.

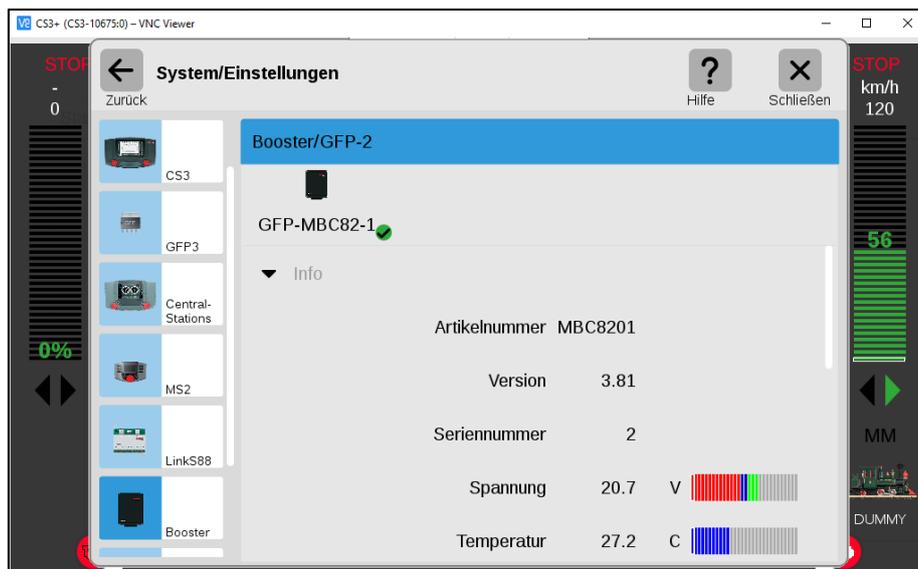


Abbildung 12-6: <Info>-Bereich allgemein

Hinzu kommt die Anzeige des Status des Moduls hinsichtlich **<ON Track>** und **<Kurzschluss>**. Bei **<ON Track>** ist die Anzeige hinter dem Menüpunkt GRÜN (vgl. Abbildung 12-7) wenn der Booster aktiv ist. Bei einem **<Kurzschluss>** wird die Anzeige entsprechend ROT, **<ON Track>** wieder GRAU.

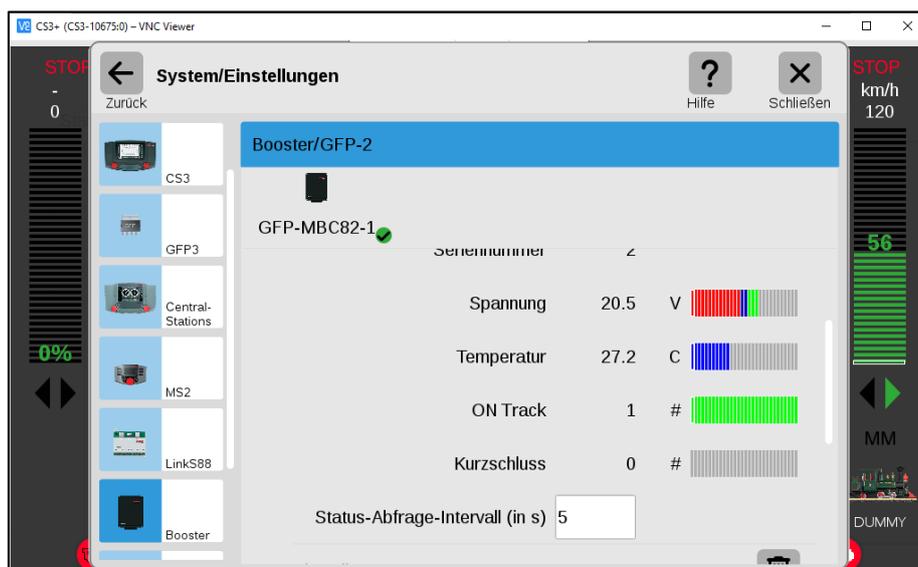


Abbildung 12-7: <Info>-Bereich ON Track und Kurzschluss

Das **<Status-Abfrage-Intervall>-Feld** regelt die Abfrage auf dem CAN-Bus zu diesem Modul. Es sollte auf dem vorgeschlagenen Wert belassen werden.

Im **<Einstellungen>-Bereich** werden der Name des Moduls, die Seriennummer, die GUID in der MBCAN-Konvention, die Firmware- und Hardwareversion angezeigt.

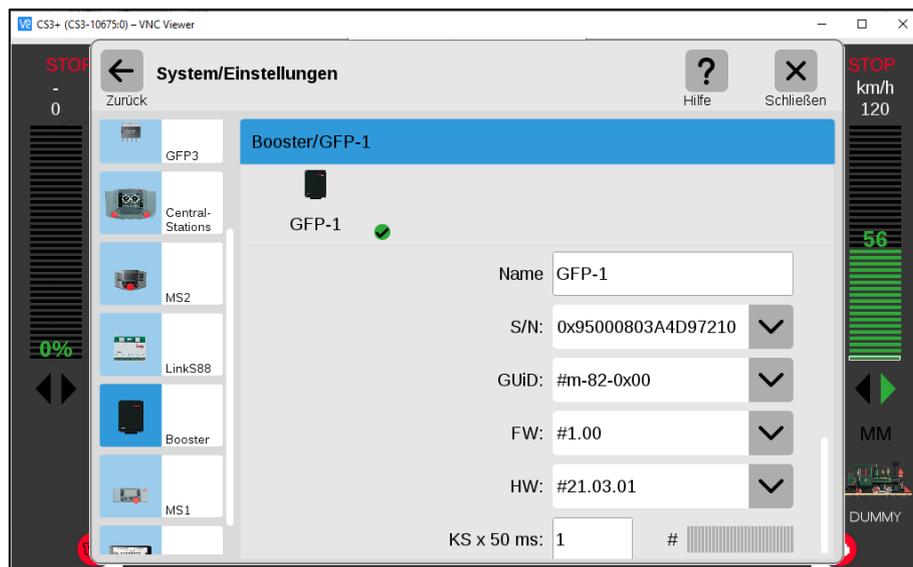


Abbildung 12-8: <Einstellungen>-Bereich

Über die Vorgabe der Latenzzeit **<KS x 50 ms>** kann die Kurzschlussempfindlichkeit des Boosters eingestellt werden. In der Grundeinstellung sind dies 50 ms bei einem satten Kurzschluss. Dieser Wert hat sich als praktikabel gegenüber der Streuung der Empfindlichkeit der Märklin®-Booster aus den unterschiedlichen Produktionsjahren ergeben. Eine Erhöhung der Latenzzeit sollte aber nur in Ausnahmefällen vorgenommen werden.

## 13 Modulbilder



Abbildung 13-1: Fertiges Modul

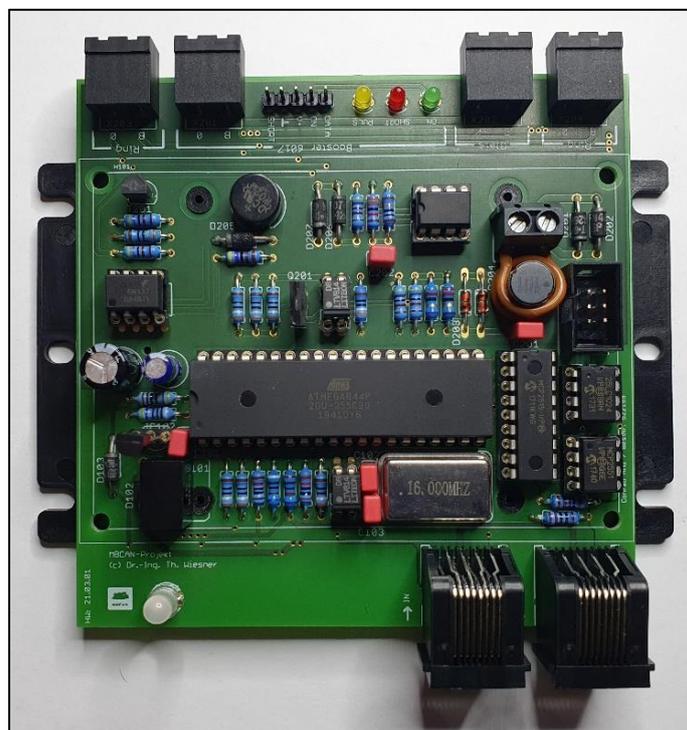


Abbildung 13-2: Bestückte Platine

## 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. Diese liegen im GENPURP-Bereich des Systemarrays.

Dargestellt ist die C-Schreibweise:

```
//=====
//= Sonderbelegung des Systemarrays nach dem Allgemeinblock           =
//=====
//= Der Beginn des Sonderblocks ist mit MBC_S_GENPURP definiert       =
//= und ergibt sich aus dem Allgemeinblock                             =
//=====

// MBC_82_L_PARA                Laenge Parametersatz
// MBC_82_S_PARA                Index Start Parametersatz
// MBC_82_STATUS                Booster aktiv BIT x, Kurzschluss BIT y

#define MBC_82_L_PARA           1
#define MBC_82_S_PARA           MBC_S_GENPURP
#define MBC_82_STATUS           sys_array[MBC_82_S_PARA]

// MBC_82_L_SW                 Laenge der Softwareversion des L88
// MBC_82_S_SW                 Index Start Softwareversion des L88
// MBC_82_SW_x                 Array mit der Softwareversion des L88

#define MBC_82_L_SW             2
#define MBC_82_S_SW             (MBC_82_L_PARA + MBC_82_S_PARA)
#define MBC_82_SW_1             sys_array[MBC_82_S_SW]
#define MBC_82_SW_2             sys_array[MBC_82_S_SW + 1]

// MBC_82_L_SH                 Laenge Kurzschlusslaenge
// MBC_82_S_SH                 Index Start Kurzschlusslaenge
// MBC_82_SH                   Kurzschlusslaenge

#define MBC_82_L_SH             1
#define MBC_82_S_SH             (MBC_82_L_SW + MBC_82_S_SW)
#define MBC_82_SH               sys_array[MBC_82_S_SH]
```

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_82_SW                0x0A                // Softwareversion
#define EE_82_SH                0x0B                // Kurzschlusslaenge
```

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:

*PRIO: 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)

<b>Befehl</b>	PC_DB_H
<b>Sender</b>	PC
<b>Loc-ID</b>	0x00001800
<b>Funktion</b>	Übertragung des Datenbanknamens mit 12 Bytes (1-4)
<b>Beschreibung</b>	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.
<b>Genutzte Datenbytes</b>	D0 – D3: Loc-ID 0x00001800
	D4 – D7: Bytes 1-4 des Datum-/Uhrzeit-Strings
<b>Nachricht</b>	00 16 5F 38 08 00 00 18 00 30 37 30 39
<b>Antwort</b>	-/-

## 15.2 PC\_DH\_M - Übertragung des Datenbanknamens mit 12 Bytes (5-8)

<b>Befehl</b>	PC_DB_M
<b>Sender</b>	PC
<b>Loc-ID</b>	0x00001801
<b>Funktion</b>	Übertragung des Datenbanknamens mit 12 Bytes (5-8)
<b>Beschreibung</b>	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.
<b>Genutzte Datenbytes</b>	D0 – D3: Loc-ID 0x00001801
	D4 – D7: Bytes 5-8 des Datum-/Uhrzeit-String
<b>Nachricht</b>	00 16 5F 38 08 00 00 18 00 31 37 32 33
<b>Antwort</b>	-/-

### 15.3 PC\_DH\_L - Übertragung des Datenbanknamens mit 12 Bytes (9-12)

<b>Befehl</b>	PC_DB_L
<b>Sender</b>	PC
<b>Loc-ID</b>	0x00001802
<b>Funktion</b>	Übertragung des Datenbanknamens mit 12 Bytes (9-12)
<b>Beschreibung</b>	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.
<b>Genutzte Datenbytes</b>	D0 – D3: Loc-ID 0x00001802
	D4 – D7: Bytes 5-8 des Datum-/Uhrzeit-Strings
<b>Nachricht</b>	00 16 5F 38 08 00 00 18 00 35 33 34 30
<b>Antwort</b>	-/-

## 15.4 PC\_KENNER - Kenner und Identifier für die Module

<b>Befehl</b>	PC_KENNER
<b>Sender</b>	PC
<b>Loc-ID</b>	0x00001803
<b>Funktion</b>	Kenner und Identifier für die Module
<b>Beschreibung</b>	<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>
<b>Genutzte Datenbytes</b>	D0 - D3: Loc-ID 0x00001803 D4: 0x00 D5: Kennung D6 – D7: Identifier
<b>Nachricht</b>	00 16 5F 38 08 00 00 18 03 00 6D AA AA
<b>Antwort</b>	-/-

## 15.5 PC\_NEU - Neuanmeldeaufforderung

<b>Befehl</b>	PC_NEU
<b>Sender</b>	PC
<b>Loc-ID</b>	0x00001804
<b>Funktion</b>	Neuanmeldeaufforderung
<b>Beschreibung</b>	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.
<b>Genutzte Datenbytes</b>	D0 - D3: Loc-ID 0x00001804
	D4 - D7: 0x00
<b>Nachricht</b>	00 16 5F 38 08 00 00 18 04 00 00 00 00
<b>Antwort</b>	MD_NEU_DATA

## 15.6 PC\_NEU\_DATA - Rückmeldung PC an Modul während des Neuanmeldeprozesses

<b>Befehl</b>	PC_NEU_DATA
<b>Sender</b>	PC
<b>Loc-ID</b>	0x00001805
<b>Funktion</b>	Rückmeldung des PC an das Modul während des Neuanmeldeprozesses
<b>Beschreibung</b>	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.
<b>Genutzte Datenbytes</b>	D0 - D3: Loc-ID 0x00001805
	D4 - D6: 0x00
	D7: n-tes Byte xx der Seriennummer
<b>Nachricht</b>	00 16 5F 38 08 00 00 18 05 00 00 00 xx
<b>Antwort</b>	MD_NEU_DATA

## 15.7 MD\_NEU\_DATA - Meldung des Moduls während des Neuanmeldeprozesses

<b>Befehl</b>	MD_NEU_DATA
<b>Sender</b>	Modul
<b>Loc-ID</b>	0x00001806
<b>Funktion</b>	Meldung des Moduls während des Neuanmeldeprozesses
<b>Beschreibung</b>	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.
<b>Genutzte Datenbytes</b>	D0 - D3: Loc-ID 0x00001806
	D4 - D6: 0x00
	D7: n-tes Byte xx der Seriennummer
<b>Nachricht</b>	00 16 2B 17 08 00 00 18 06 00 00 00 xx
<b>Antwort</b>	PC_NEU_DATA

## 15.8 PC\_RESET - Durchführen eines Hardware-Resets auf dem Modul

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

## 15.9 PC\_MD\_SEL - Modul aus Datenbank entfernen

<b>Befehl</b>	PC_MD_DEL
<b>Sender</b>	PC
<b>Loc-ID</b>	0x00001808
<b>Funktion</b>	Modul aus Datenbank entfernen
<b>Beschreibung</b>	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.
<b>Genutzte Datenbytes</b>	D0 - D3: Loc-ID 0x00001808
	D4 - D7: GUID des Moduls
<b>Nachricht</b>	GUID mbc-84 #1 = 6D 38 34 01
	00 16 5F 38 08 00 00 18 08 6D 38 34 01
<b>Antwort</b>	-/-

## 15.10 PC\_ALIVE - ALIVE-Abfrage

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

## 15.11 MD\_ALIVE - ALIVE-Abfrage

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

## 15.12 PC\_ARRAY - Zugriff Systemarray anfragen

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

### 15.13 MD\_ARRAY - Zugriff Systemarray freigegeben

<b>Befehl</b>	MD_ARRAY
<b>Sender</b>	Modul
<b>Loc-ID</b>	0x0000180C
<b>Funktion</b>	Zugriff Systemarray freigegeben
<b>Beschreibung</b>	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.
<b>Genutzte Datenbytes</b>	D0 - D3: Loc-ID 0x0000180C
	D4 - D7: GUID des Moduls
<b>Nachricht</b>	GUID mbc-84 #1 = 6D 38 34 01
	00 16 2B 17 08 00 00 18 0C 6D 38 34 01
<b>Antwort</b>	-/-

## 15.14 PC\_ARRAY\_DATA - Zugriff Systemarray freigegeben

<b>Befehl</b>	PC_ARRAY_DATA
<b>Sender</b>	PC
<b>Loc-ID</b>	0x0000180D
<b>Funktion</b>	Zugriff Systemarray freigegeben
<b>Beschreibung</b>	<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 -&gt; Lesen, 1 -&gt; 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>
<b>Genutzte Datenbytes</b>	D0 - D3: Loc-ID 0x0000180D D4: 0 -> Lesen, 1 -> Schreiben D5 - D6: Systemarray-Index D7: zu schreibender Wert, beim Lesen n.c.
<b>Nachricht</b>	Wert 0x0A an die Stelle 0x0001 im Systemarray schreiben 00 16 5F 38 08 00 00 18 0D 01 00 01 0A
<b>Antwort</b>	MD_ARRAY_DATA

### 15.15 MD\_ARRAY\_DATA - Antwort des Moduls auf Systemarray-Zugriff

<b>Befehl</b>	MD_ARRAY_DATA
<b>Sender</b>	Modul
<b>Loc-ID</b>	0x0000180E
<b>Funktion</b>	Antwort des Moduls auf Systemarray-Zugriff
<b>Beschreibung</b>	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.
<b>Genutzte Datenbytes</b>	D0 - D3: Loc-ID 0x0000180E
	D4: 0 -> Lesen, 1 -> Schreiben
	D5 - D6: Systemarray-Index
	D7: gelesener Inhalt des Systemarrays, beim Schreiben n.c.
<b>Nachricht</b>	Gelesener Wert 0x07 aus der Stelle 0x0108 im Systemarray
	00 16 2B 17 08 00 00 18 0E 00 01 08 07
<b>Antwort</b>	-/-

## 15.16 PC\_UPGRADE - Firmware-Upgrade

<b>Befehl</b>	PC_UPGRADE
<b>Sender</b>	PC
<b>Loc-ID</b>	0x0000180F
<b>Funktion</b>	Firmware-Upgrade
<b>Beschreibung</b>	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.
<b>Genutzte Datenbytes</b>	D0 - D3: Loc-ID 0x0000180F
	D4 - D7: GUID des Moduls
<b>Nachricht</b>	GUID mbc-84 #1 = 6D 38 34 01
	00 16 5F 38 08 00 00 18 0F 6D 38 34 01
<b>Antwort</b>	MD_UPGRADE

### 15.17 MD\_UPGRADE - Firmware-Upgrade freigeben

<b>Befehl</b>	MD_UPGRADE
<b>Sender</b>	Modul
<b>Loc-ID</b>	0x00001810
<b>Funktion</b>	Firmware-Upgrade freigeben
<b>Beschreibung</b>	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.
<b>Genutzte Datenbytes</b>	D0 - D3: Loc-ID 0x00001810
	D4 - D7: GUID des Moduls
<b>Nachricht</b>	GUID mbc-84 #1 = 6D 38 34 01
	00 16 2B 17 08 00 00 18 10 6D 38 34 01
<b>Antwort</b>	-/-

## 15.18 PC\_UPGRADE\_DATA - Schreibe Firmware

<b>Befehl</b>	PC_UPGRADE_DATA
<b>Sender</b>	PC
<b>Loc-ID</b>	0x00001811
<b>Funktion</b>	Schreibe Firmware
<b>Beschreibung</b>	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.
<b>Genutzte Datenbytes</b>	HASH: Laufende Nummer in der jeweiligen PAGE D0 - D3: Loc-ID 0x00001811 D4 - D7: 4 zu schreibende Bytes
<b>Nachricht</b>	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
<b>Antwort</b>	MD_UPGRADE_DATA

### 15.19 MD\_UPGRADE\_DATA - Antwort des Moduls auf Schreibe Firmware

<b>Befehl</b>	MD_UPGRADE_DATA
<b>Sender</b>	Modul
<b>Loc-ID</b>	0x00001812
<b>Funktion</b>	Antwort des Moduls auf Schreibe Firmware
<b>Beschreibung</b>	Das Modul antwortet mit der exakten Datenstruktur der gesendeten Nachricht und signalisiert damit, dass es die Upgrade-Daten im externen EEPROM gespeichert hat.
<b>Genutzte Datenbytes</b>	HASH: Laufende Nummer in der jeweiligen PAGE
	D0 - D3: Loc-ID 0x00001812
	D4 - D7: 4 zu schreibende Bytes
<b>Nachricht</b>	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
<b>Antwort</b>	-/-

## 15.20 PC\_BOOT - Modul neu Booten

<b>Befehl</b>	PC_BOOT
<b>Sender</b>	PC
<b>Loc-ID</b>	0x00001813
<b>Funktion</b>	Modul neu Booten
<b>Beschreibung</b>	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.
<b>Genutzte Datenbytes</b>	D0 - D3: Loc-ID 0x00001813 D4 - D7: GUID des Moduls
<b>Nachricht</b>	GUID mbc-84 #1 = 6D 38 34 01 00 16 5F 38 08 00 00 18 13 6D 38 34 01
<b>Antwort</b>	-/-

## 15.21 MD\_S88 - Stellungsmeldung mbc-88 / mbc-90

<b>Befehl</b>	MD_S88
<b>Sender</b>	Modul
<b>Loc-ID</b>	0x00001814
<b>Funktion</b>	Stellungsmeldung mbc-88 / mbc-90
<b>Beschreibung</b>	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.
<b>Genutzte Datenbytes</b>	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
<b>Nachricht</b>	Modul 16, Kontakt 2 hat Stellung 1
	00 16 2B 17 08 00 00 18 14 00 10 02 01
<b>Antwort</b>	-/-

## 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*

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

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

## 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](mailto: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