



Nicht-kommerzielles Projekt – Alle Angaben ohne Gewähr

# Bedienungsanleitung

MM2®-Sniffer mbc-89

Version 1.6

HW 22.03.01, FW 2.16, Parametriercenter ab 2.1.2.0

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



# 1 Inhalt

2	D	iscla	claimer	4
3	R	evis	vision	5
	3.1		Bedienungsanleitung	5
	3.2		Firmware	5
4	Ei	nlei	leitung	6
5	Fu	unkt	ıktion	7
6	So	chal	altbild	8
7	В	estü	tückung	10
8	В	aute	ıteileliste	12
9	Fi	rmv	nware	14
10	)	St	Steckverbindungen	15
11	L	Ar	Anschlussbeispiele	17
	11.1	-	Grundfunktion A	17
	11.2	2	Grundfunktion B	18
	11.3	3	Grundfunktion C	19
	11.4	ļ	Maximalausbau	21
12	2	In	Inbetriebnahme	22
	12.1	-	Modul in Betriebsbereitschaft versetzen	22
	12.2	<u>)</u>	Modul konfigurieren	22
	12	2.2.	2.1 Protokollauswahl Zubehörartikel	23
	12	2.2.2	2.2 Mapping/Direkt	23
	12	2.2.3	2.3 I2C-Busfrequenz	24
	12	2.2.4	2.4 Softwareversion des Connect6021®	24
	12	2.2.	2.5 Aktueller Zustand der möglichen 80 Lokomotiven	25
	12	2.2.0	2.6 Aktualisierung von Daten	25
	12.3	3	Modul mit der CS2® verbinden	26
	12.4	ļ	Modul mit der CS3® verbinden	28
13	3	M	Modulbilder	30
14	1	Sy	Systemarray-Belegung für Eigenentwicklungen	31
	14.1	-	Allgemeiner Bereich zum Modul	31
	14.2	2	Modulspezifischer Bereich für Funktionsparameter	35



15	Befehlssatz zu den Modulen	37		
15.1	PC_DH_H - Übertragung des Datenbanknamens mit 12 Bytes (1-4)	39		
15.2	PC_DH_M - Übertragung des Datenbanknamens mit 12 Bytes (5-8)	40		
15.3	PC_DH_L - Übertragung des Datenbanknamens mit 12 Bytes (9-12)	41		
15.4	PC_KENNER - Kenner und Identifier für die Module	42		
15.5	PC_NEU - Neuanmeldeaufforderung	43		
15.6	PC_NEU_DATA - Rückmeldung PC an Modul während des Neuanmeldeprozesses	44		
15.7	MD_NEU_DATA - Meldung des Moduls während des Neuanmeldeprozesses	45		
15.8	PC_RESET - Durchführen eines Hardware-Resets auf dem Modul	46		
15.9	PC_MD_SEL - Modul aus Datenbank entfernen	47		
15.1	0 PC_ALIVE - ALIVE-Abfrage	48		
15.1	1 MD_ALIVE - ALIVE-Abfrage	49		
15.1	2 PC_ARRAY - Zugriff Systemarray anfragen	50		
15.1	3 MD_ARRAY - Zugriff Systemarray freigegeben	51		
15.1	4 PC_ARRAY_DATA - Zugriff Systemarray freigegeben	52		
15.1	5 MD_ARRAY_DATA - Antwort des Moduls auf Systemarray-Zugriff	53		
15.1	6 PC_UPGRADE - Firmware-Upgrade	54		
15.1	7 MD_UPGRADE - Firmware-Upgrade freigeben	55		
15.1	8 PC_UPGRADE_DATA - Schreibe Firmware	56		
15.1	9 MD_UPGRADE_DATA - Antwort des Moduls auf Schreibe Firmware	57		
15.2	0 PC_BOOT - Modul neu Booten	58		
15.2	1 MD_S88 - Stellungsmeldung mbc-88 / mbc-90	59		
16	Post-Code	60		
17	Quellenverzeichnis	62		
18	Allgemeine Hinweise zum MBCAN-Projekt63			



## 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	08.05.2022	Erste Version
1.1	04.09.2022	Diverse Bugs behoben und I2C-Frequenz einstellbar gemacht
1.2	18.10.2022	Anpassung Warenkorb DS 18B20
1.3	04.08.2023	Ergänzung I2C-Bus Frequenztabelle
1.4	22.02.2024	Redaktionelle Anpassungen
1.5	01.10.2024	Redaktionelle Anpassungen
1.6	20.10.2024	Redaktionelle Anpassungen Befehlssatz

# 3.2 Firmware

FW: 1.00 - Erste Version				
FW: 1.01 - I2C-Kommunikation angepasst				
FW: 1.02 - PullUp am Eingang des CU-Sensors deaktiviert				
FW: 1.03 - Variable Frequenzen für I2C-Bus eingeführt				
FW: 1.04 - Programmierbare I2C-Bus-Frequenzen eingeführt				
FW: 1.05 - Rückmeldung Strom aus auf gewählten Ausgang 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]

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

Im Folgenden finden Sie hier die Beschreibung des MM2®-Sniffers. Er ermöglicht das Abtasten des Gleissignals der von alten MM®-Zentralen sowie der IR-Fernbedienung 00748 von Märklin® sowie der Übernahme der I2C-Systembus-Nachrichten der Märklin®-Digital-Geräte der 1. Generation.



#### **Funktion**

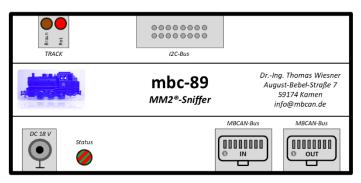


Abbildung 5-1:Modullabel

Dieses Modul stellt die Verbindung zwischen MM2®-Modellbahnzentrale und dem MBCAN her.

Werden die MM2®-Befehle direkt aus dem Gleis übernommen, erfolgt eine Übersetzung von Lok- und Schaltbefehlen in das CAN-Protokoll OHNE Rücksynchronisation. Schaltbefehle werden dann von den

Modulen mbc-83/84/87 ausgeführt. Eventuell angeschlossene Zentralen CS2/3® und MS2® können die CAN-Kommandos verarbeiten und zeigen die Schaltstellung an. Diese Funktion ist insbesondere für StartUp®-Fernbedienungen vorgesehen.

Bei Nutzung des I2C-Busses erfolgt eine vollständige Synchronisierung der Keyboards 6040® sowie der Zentrale 6021® nebst Steuerpulten 6036® mit den mbc-Modulen sowie den Zentralen CS2/3® und MS2®. Im Fall CS2/3® tritt das Modul ggü. einer CS2/3® als Connect6021® auf, Loks können gemappt werden. Das Mapping des Keyboards 6040® auf MM2® und DCC®-Adressbereiche ist blockweise (je vier Adressen) möglich.

Außerdem umfasst das Modul noch eine Stromversorgung für den MBCAN-Bus für bis zu vier Module. Eine unterstützende Stromversorgung ist je mbc-Modul jederzeit möglich.

Mehrere mbc-89 am MBCAN-Bus sind möglich. Es sind 2x mbc-89 notwendig, um mit 20 Keyboards den gesamten MM-Adressraum von 320 möglichen Zubehörartikeln adressieren zu können (DCC bis 1024).

Steckverbinder	2x RJ45 MBCAN-Bus / StartUp über Schraub-Steckverbinder / I2C-Stecker zur Verbindung zum Connect6021-Adapter
Stromversorgung	Steckernetzteil / RJ45-Versorgung
Statusanzeige	Betriebszustand und Traffic wird über Dreifarb-LED angezeigt
Adresszuordnung	-/-
Adressformat	CAN-Bus-Protokoll für MM/mfx/DCC bei Lokomotiven und MM/DCC bei Magnetartikeln möglich
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

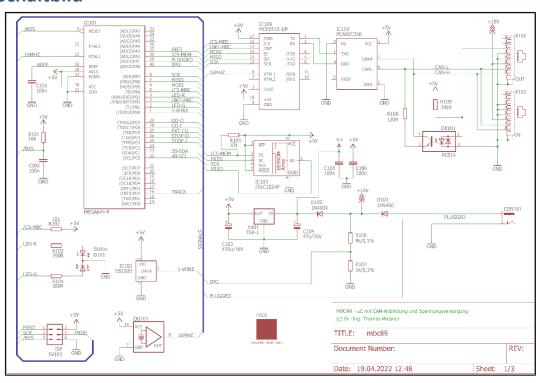


Abbildung 6-1: Prozessor-Schaltbild

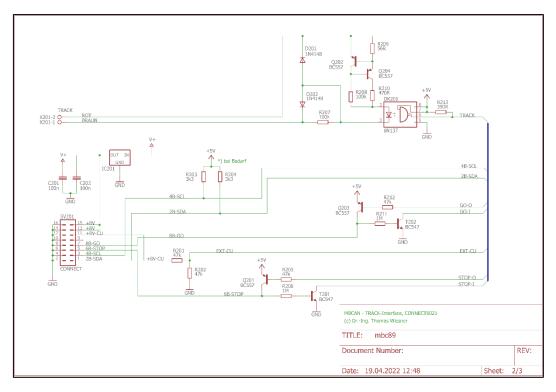


Abbildung 6-2: Sonderfunktionen

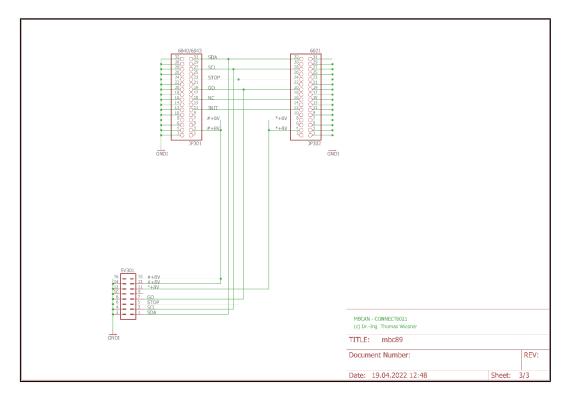


Abbildung 6-3: Sonderfunktionen I2C-Adapter

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

Die Sonderfunktionen des Sniffers bestehen aus der galvanisch getrennten Schnittstelle zum Gleis sowie aus einem I2C-Systembusadapter für Märklin®-Digital-Geräte der 1. Generation.



## 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).

Im Falle der ISP-Schnittstelle ist zu überlegen, ob bereits extern programmierte Controller eingesetzt werden oder es die Möglichkeit der OnBoard-Programmierung für spätere Zeitpunkte geben soll.

Für ein Upgrade der Module ist die ISP-Schnittstelle in der Regel nicht von Nöten, da dies über den CAN-Bus erfolgt. Es sei denn, beim Upgrade ist ein Fehler aufgetreten und das Modul ist nicht mehr ansprechbar. In jedem Falle ist der Controller aber mit der Ur-Software extern vorzubereiten.

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.

Das PCB ist mit einer gefrästen Trennlinie verbunden, um den I2C-Adapter von der Hauptplatine zu lösen. Bitte vorsichtig trennen und die überstehenden Nasen mit einer Feile entfernen.

Die Widerstände R203 und R204 werden nur benötigt, wenn das Verbindungkabel zwischen dem 6021-Adapter und dem mbc-89 länger als 30 cm ist.



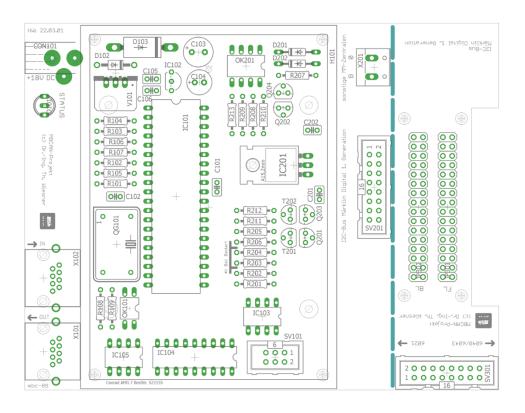


Abbildung 7-1: Bestückung Bauteilnummern

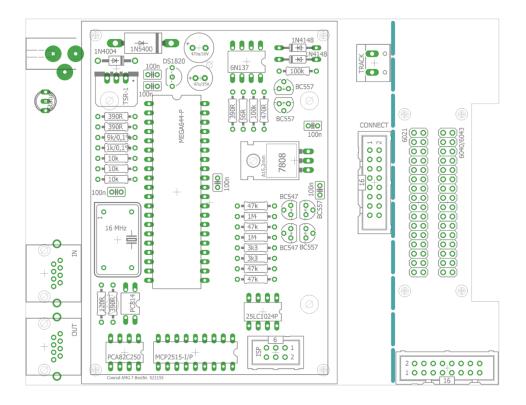


Abbildung 7-2: Bestückung Bauteilwerte



## 8 Bauteileliste

Die für die Bestückung benötigten Bauteile sind in nachfolgender Tabelle aufgelistet. Ergänzt sind außerdem ein möglicher Lieferant sowie die zugehörige Bestellnummer. Der Lieferant ist nur ein Vorschlag und ist nicht bindend. Das Gehäuse ist optional. Die Platine ist allerdings exakt auf das in der unteren Tabelle stehende bei Conrad erhältliche Gehäuse AMG 7 abgestimmt.

Tabelle 8-1: Stückliste

Part	Value	Lieferant	Bestellnummer	Anzahl
C101, C102, C105, C106,	100n	Reichelt	MKS02-63 100N	6
C201, C202	470/4.61/	Daishala	DAD 470/46	4
C103	470u/16V	Reichelt	RAD 470/16	1
C104	47u/35V	Reichelt	RAD 47/35	1
CON101	POWER-JACK	Reichelt	LUM NEB 21R	1
D101	Status	Reichelt	LED 5 RG-3	1
D102	1N4004	Reichelt	1N 4004	1
D103	1N5400	Reichelt	1N 5400	1
D201, D202	1N4148	Reichelt	1N 4148	2
IC101	MEGA644-P	Reichelt	ATMEGA 644P-20PU	1
IC102	DS1820	Reichelt	DS 18S20	1
oder	DS1820	Reichelt	DS 18B20	1
IC103	25LC1024P	Reichelt	25LC1024-I/P	1
IC104	MCP2515-I/P	Reichelt	MCP 2515-I/P	1
IC105	PCA82C250	Reichelt	MCP 2551-I/P	1
IC201	78L08	Reichelt	μΑ 7808	1
JP301	Federleiste 2x16	Reichelt	FL-Q2 32W	1
JP302	Messerleiste 2x16	Reichelt	ML-B2 32W	1
OK101	PC814	Reichelt	EL 814	1
OK201	6N137	Reichelt	6N 137	1
Q201 - Q204	BC557	Reichelt	BC 557C	4
QG101	16 MHz	Reichelt	OSZI 16,000000	1
R101, R102, R105	10k	Reichelt	METALL 10,0K	3
R103, R104, R109, R213	390R	Reichelt	METALL 390	4
R106	9k/0,1%	Reichelt	MPR 9,10K	1
R107	1k/0,1%	Reichelt	MPR 1,10K	1
R108	120R	Reichelt	METALL 120	1
R201, R202, R205, R212	47k	Reichelt	METALL 47,0k	4
R203, R204	3k3	Reichelt	METALL 3,30k	2
R206, R211	1M	Reichelt	METALL 1,00M	2
R207, R208	100k	Reichelt	METALL 100k	2
R209	56R	Reichelt	METALL 56	1
R210	470R	Reichelt	METALL 470	1
SV101	ML6	Reichelt	WSL 6G	1



SV201, SV301	ML16	Reichelt	WSL 16G	2
T201, T202	BC547	Reichelt	BC 547C	2
V101	TSR-1	Reichelt	TSR 1-2450	1
X101, X102	CAT5 T1U 2.8N4N	Reichelt	CAT5 T1U 2.8N4N	2
X201	AKL 230-02	Reichelt	AKL 230-02	1
Y201	AKL 249-02	Reichelt	AKL 249-02	1
YV201, YV301	HAN 516 6803	Reichelt	HAN 516 6803	2
Z201-301	AWG 28-16G 3M	Reichelt	AWG 28-16G 3M	1
ICS-8pol	Sockel 8	Reichelt	GS 8P	4
ICS-18pol	Sockel 18	Reichelt	GS 18P	1
ICS-40pol	Sockel 40	Reichelt	GS 40P	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:

- a) Dateien des Typs *mbc\_xx\_xx\_xx\_isp.hex* sind für die Erstprogrammierung zu verwenden und über die ISP-Schnittstelle aufzuspielen
- b) Dateien des Typs mbc\_xx\_xx\_xx\_upgrade.hex sind für das Upgrade über das Parametriercenter gedacht. Sie funktionieren NICHT bei der Programmierung über die ISP-Schnittstelle

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

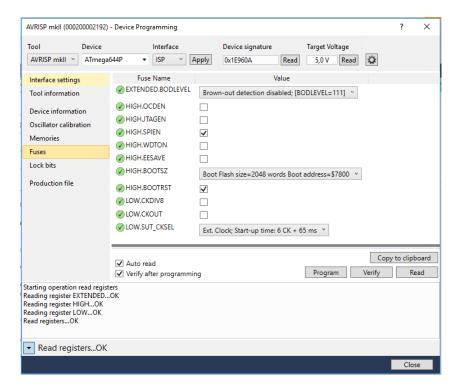


Abbildung 9-1: FUSES im AVR Studio



# 10 Steckverbindungen

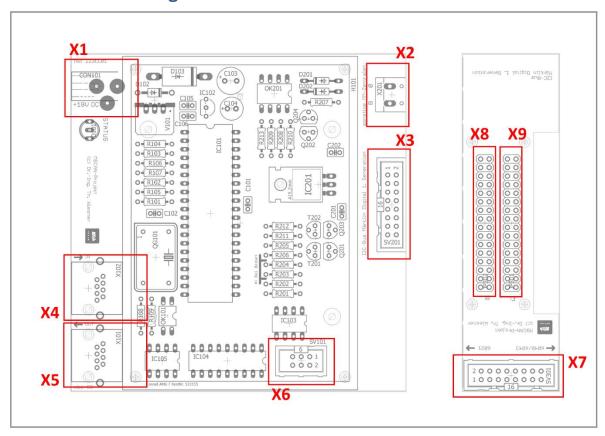


Abbildung 10-1: Steckverbinder

## **X1** dezentrale Spannungsversorgung

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

#### **X2** TRACK-Anschlüsse

Diese Schraubklemme ist für die Gleiseinspeisung von den Märklin® Digital® Zentralen 602x vorgesehen.

1 = Braun / 0, 2 = Rot / Bahnstrom

#### **X3/X7** *12C*<sup>®</sup>-Systembusstecker

Diese Pinleiste führt die I2C®-Systembussignale.

#### X4 MBCAN-IN

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



## **X5** MBCAN-OUT

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

## **X6** Optionale ISP-Schnittstelle

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

## **X8/X9** *I2C*<sup>®</sup>-Systembusstecker

Federbuchse und Federleiste zur Adaption an die Märklin®-Digital-Geräte 6021®/6040®/6043®



## 11 Anschlussbeispiele

Das Modul dient der Umsetzung des MM2®-Protokolls auf CAN-Kommandos im Sinne der Protokollbeschreibung von Märklin® entweder direkt aus dem TRACK-Signal oder über den I2C®-Systembus von Märklin®. Von daher kann vom Prinzip her jede Modellbahnzentrale, welche MM2® spricht, über dieses Modul Zugang zur großen Welt des CAN-Bus haben. Die folgenden Grundfunktionenkönnen beliebig kombiniert werden.

#### 11.1 Grundfunktion A

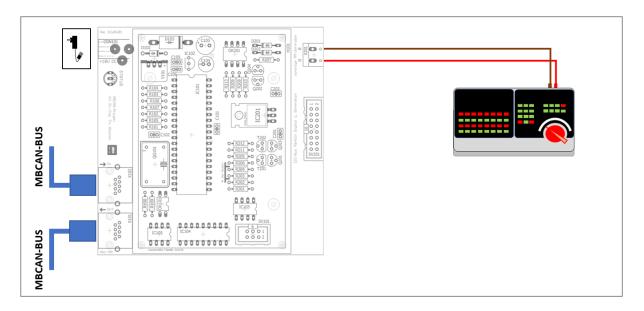


Abbildung 11-1: Grundfunktion A - Beliebige MM-Zentrale

Beim Anschluss eines Gleis-Ausgangs einer beliebigen MM®-Zentrale an das mbc-89 werden die TRACK-Signale ausgewertet und in korrespondierende CAN-Nachrichten übersetzt. Dies gilt sowohl für Lok- als auf Zubehörartikel-Nachrichten.

Befindet sich eine CS2/3® am MBCAN-Bus, können die Lokadressen der MM®-Zentrale auf beliebige Lokadressen der CS2/3® gemappt werden (vgl. Connect6021® von Märklin®).

Bei Nutzung der IR-Fernbedienung 00748® aus den StartUp®-Sets von Märklin® können bis zu vier Handsender an der StartUp-Gleisbox verwendet werden. Die Loks mit den MM-Adressen 24, 60, 72 und 78 können auf beliebige Lokadressen der CS2/3® gemappt werden (vgl. Connect6021® von Märklin®).

Bei Verwendung einer Gleisbox® mit MS2® als Zentrale ist ein Mapping auf beliebige Lokadressen auf Grund der fehlenden GUI auf der MS2® nicht möglich. Über das Parametriercenter ist in diesem Fall ein direktes Routing von MM®-Lokadresse der TRACK-Zentrale auf identische MM®-Adressen der MS2® möglich.



Lokomotiven, die gemappt sind, können nur noch von den Steuerpulten der MM®-Zentrale oder der Infrarot-Fernsteuerungen gesteuert werden. Auf den Steuerpulten der CS2/3® und der MS2® werden die aktuellen Lokinfos angezeigt, können aber nicht geändert werden da dies vom mbc-89 gesperrt wird. Das macht Sinn, da nicht immer eine bidirektionale Synchronisierung der Zentralen möglich ist. Aus diesem Grund werden auch alle gemappten Loks nach dem Start des Moduls auf Geschwindigkeit "0", Fahrtrichtung auf "vorwärts" und alle Funktionen auf "aus" gesetzt. Somit gibt es immer einen definierten Anfangszustand.

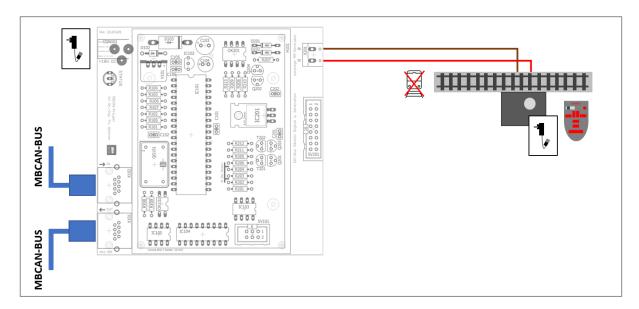


Abbildung 11-2: Grundfunktion A - Infrarot-Fernbedienung 00748 von Märklin®

Zubehörartikeladressen werden entweder 1:1 in den MM®-CAN-Schaltbefehl übersetzt oder auf die gleiche Adresse im DCC-Adressraum gemappt, was sowohl bei der CS2/3® als auch bei der MS2® gilt.

#### 11.2 Grundfunktion B

Über die 6021®-Adapterplatine kann das mbc-89 an die 6021® angeschlossen werden. Alle Lokbefehle des 6021® werden in korrespondierende CAN-Nachrichten übersetzt.

Befindet sich eine CS2/3® am MBCAN-Bus, können die Lokadressen der 6021 auf beliebige Lokadressen der CS2/3® gemappt werden (vgl. Connect6021® von Märklin®).

Bei Verwendung einer Gleisbox® mit MS2® als Zentrale ist ein Mapping auf beliebige Lokadressen auf Grund der fehlenden GUI auf der MS2® nicht möglich. Über das Parametriercenter ist in diesem Fall ein direktes Routing von MM®-Lokadresse der 6021® auf identische MM®-Adressen der MS2® möglich.

Lokomotiven, die gemappt sind, können nur noch von den Steuerpulten der 6021®/6036® gesteuert werden. Auf den Steuerpulten der CS2/3® und der MS2® werden die aktuellen Lokinfos angezeigt, können aber nicht geändert werden da dies vom mbc-89 gesperrt wird. Zwar wäre die



Synchronisierung grundsätzlich möglich, aber warum sollte eine Lok auf ein Steuerpult übergeben werden, wenn es durch die CS2/3® oder MS2® gesteuert werden soll. Alle gemappten Loks werden nach dem Start des Moduls auf Geschwindigkeit "0", Fahrtrichtung auf "vorwärts" und alle Funktionen auf "aus" gesetzt. Somit gibt es immer einen definierten Anfangszustand.

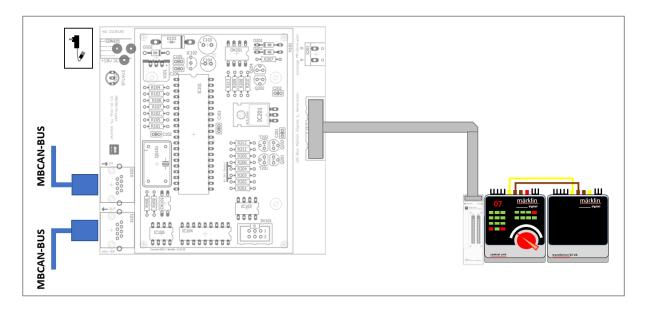


Abbildung 11-3: Grundfunktion B - 6021® am I2C-Systembus®

### **HINWEIS:**

Auf Grund des Eingriffs in den I2C-Bus ist eine Übernahme von Loks von einem Steuerpult auf das andere Steuerpult wie von Märklin vorgesehen leider nicht möglich. Dieser Modus wird durch Blinken des eigentlichen Steuerpultes angezeigt, wo hingegen das anfordernde Steuerpult nur die Lokadresse aber ohne Richtungspfeil anzeigt. Wird versucht, die Übernahme dennoch zu erzwingen gehen beide Steuerpulte in den Freeze-Modus (bei beiden Steuergeräten wird nur die Lokadresse aber ohne Richtungspfeil angezeigt). In diesem Fall bitte durch gleichzeitiges Drücken der STOP- und GO-Tasten die 6021 resetten. Danach sollte der Freeze-Modus wieder aufgehoben sein.

#### 11.3 Grundfunktion C

Über die 6021®-Adapterplatine kann das mbc-89 an das 6040® und oder 6043® angeschlossen werden. Alle Zubehörartikelbefehle des 6040®/6043® werden in korrespondierende CAN-Nachrichten übersetzt.

Zubehörartikeladressen werden entweder 1:1 in den MM®-CAN-Schaltbefehl übersetzt oder auf die gleiche Adresse im DCC-Adressraum gemappt, was sowohl bei der CS2/3® als auch bei der MS2® gilt. Eingestellt werden kann dies im Parametriercenter. Dies gilt aber nur für die Richtung 6040® -> CS2/3® resp. MS2. ® Die umgekehrte Richtung wird immer in den MM®-Adressraum des 6040 gemappt. Es stehen damit 16 Keyboards und somit 256 Zubehörartikel zur Verfügung. Die freien MM®-Adressen oberhalb 256 können durch das mbc-89 nicht gemappt werden.



Das 6043® kann entweder Stand-Alone oder in Kombination mit dem 6040® eingesetzt werden. Bitte beachten sie die nicht mögliche Verriegelung von Fahrstraßen, welche auf der CS2/3® eingerichtet wurden und denen der 6043® da sie nicht überprüft werden. Eine Programmierung des 6043® ohne 6040® ist leider nicht möglich.

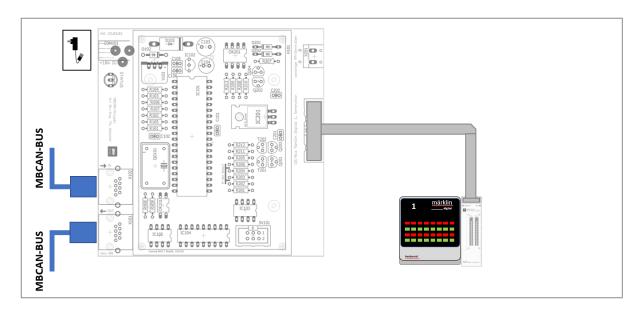


Abbildung 11-4: Grundfunktion C - Schalten im MM/DCC-Format über 6040®

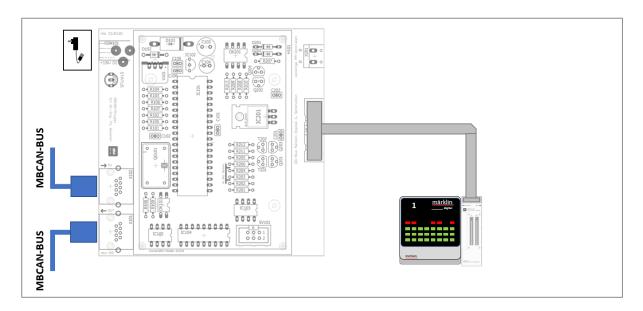


Abbildung 11-5: Grundfunktion C - Einsatz eines Memory 6043®



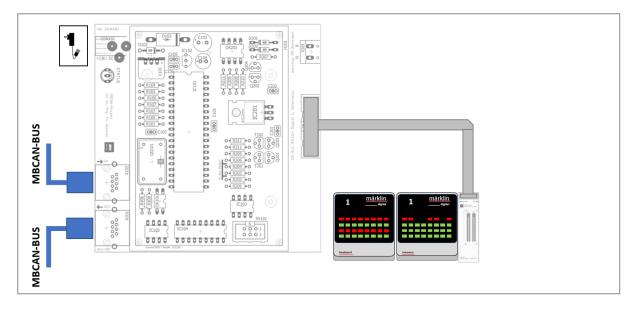


Abbildung 11-6: Grundfunktion C - Einsatz von 6040® und 6043®

### 11.4 Maximalausbau

Alle drei Grundfunktionen lassen sich beliebig kombinieren.

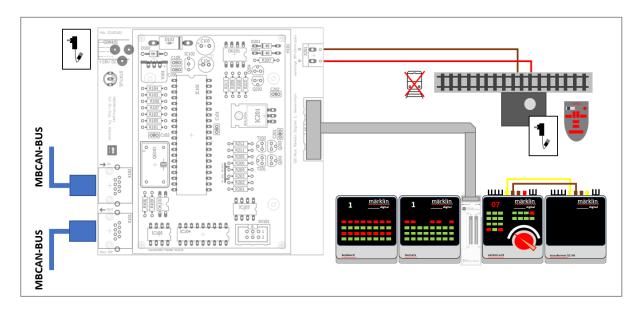


Abbildung 11-7: Kombination aller Grundfunktionen



### 12 Inbetriebnahme

#### 12.1 Modul in Betriebsbereitschaft versetzen

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

## 12.2 Modul konfigurieren

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

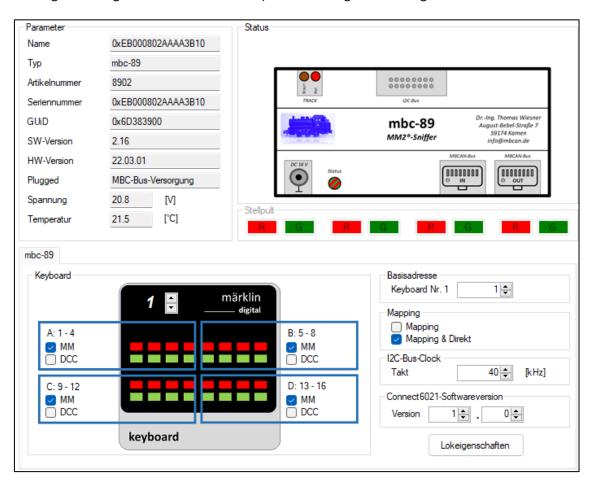
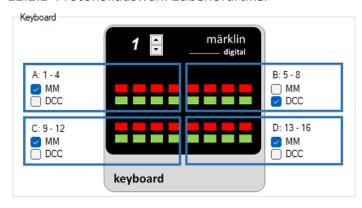


Abbildung 12-1: Konfigurationsbereich

Das Modul mbc-89 arbeitet in der Grundfunktion als Connect6021®-Derivat. Es gibt aber darüber hinaus weitere Funktionen, die das mbc-89 universeller machen.



#### 12.2.1 Protokollauswahl Zubehörartikel



Zubehörartikel werden über die Keyboardtasten gesteuert. Je vier Tastenpaare sind zu einem Block zusammengefasst, was dem üblichen Funktionsumfang von gängigen Zubehörartikeldecodern entspricht. Die Grundadresse je Keyboard wird über die beiden <Pfeile> oben links im Keyboardbild ausgewählt (1 bis 16 = 256 Zubehörartikel). Die

korrespondierenden Adressen der Blöcke werden neben <A bis D> in den blauen Boxen angezeigt. Nun kann ausgewählt werden, ob ein Tastendruck auf dem Keyboard einen MM®- oder einen DCC®-Schaltbefehl über den CAN-Bus auslösen soll. Natürlich muss der entsprechende Decoder in der CS2/3® und MS2® auf dasselbe Protokoll eingestellt sein. Der Gleisformatprozessor® gibt dann die entsprechenden MM®- oder DCC®-Nachrichten auf dem Gleis aus, um die Zubehörartikeldecoder anzusprechen. Sollten Sie nur MBCAN-Schaltdecoder verwenden, brauchen Sie nichts anzupassen, da die Basiseinstellung immer auf dem MM®-Protokoll beruht.



Durch Auswahl der **<Basis>** des mbc-89 können bis zu 4 Connect6021®-Derivate und damit 1024 möglichen MM®-Zubehörartikel adressiert werden.

CS2/3® und MS2® lassen allerdings nur 320 zu, bei DCC® sind 1024 Zubehörartikel möglich.

### 12.2.2 Mapping/Direkt



Das Connect6021® von Märklin® erlaubt ein beliebiges Mapping von Lokadressen der 6021® auf Loks der CS2/3® (vgl. Connect6021®). Entsprechend soll eine Lok der CS2/3 nur dann

auf Kommandos der 6021® reagieren, wenn die Lok gemappt wurde. Dies gilt für die Basiseinstellung. Um auch bei einer MS2® ein Mapping ermöglichen zu können, gibt es die Option **<Mapping & Direkt>**. In diesem Fall werden alle 80 möglichen Loks der 6021® mit den MM®-Lokadressen 1 bis 80 auf MS2®-MM®-Loks mit den identischen Adressen gemappt. Eine Steuerung von mfx®- oder DCC®-Loks ist bei der MS2® nicht möglich.

Wird diese Option bei einer CS2/3® gewählt, werden alle 6021®-Loks auf MM®-Loks der CS2/3® geroutet oder aber, wenn Mapping vorliegt, die entsprechende Lok angesteuert. Somit ergibt sich eine Vielzahl von Mapping-Möglichkeiten.



#### 12.2.3 I2C-Busfrequenz

Märklin® hat in den Produktionsjahren unterschiedliche Generationen seiner Digital-Serie auf den Markt gebracht. Z.B. gibt es eine Serie, die zusammen mit Arnold® produziert wurde.



Leider sind die Einzelgeräte untereinander aus verschiedenen Produktionsjahren zum Teil nicht kompatibel zueinander.

Beim Test verschiedener Geräte-Baujahre und Kombinationen haben sich Inkompatibilitäten bei der Frequenz auf dem I2C-Bus ergeben. Aus diesem Grund kann über dieses Auswahlfeld die I2C-Bus-Frequenz angepasst und so ggf. eine Nichtfunktionalität aufgelöst werden.

Der Modellbaukollege Rainer Knittlmeier hat verschiedene Digital-Geräte von Märklin® und Arnold® getestet und folgende Kombinationen von Bauserie und I2C-Busfrequenz identifiziert. Diese Tabelle soll als Anhaltspunkt für die eigene Parametrierung dienen.

Hersteller	Merkmal	Frequenz	Im Verbund	Wechsel der	Frequenz mit
	Keyboard	ohne	mit anderen	Anordnung	Märklin®
		Märklin®	Keyboards		6021
		6021			(DIP rot)
Arnold®	Schrift rot	1 - 28 kHz	DIP rot	Keine	30 - 100 kHz
			1 - 58 kHz	Auswirkungen	
Arnold®	Schrift	1 - 25 kHz	DIP rot	Keine	30 - 100 kHz
	schwarz		1 - 58 kHz	Auswirkungen	
Arnold®	Schrift rot	1 - 28 kHz	DIP schwarz	Keine	30 - 100 kHz
			1 - 28 kHz	Auswirkungen	
Arnold®	Schrift	1 - 25 kHz	DIP schwarz	Keine	30 - 100 kHz
	schwarz		1 - 28 kHz	Auswirkungen	
Arnold®	Schrift rot	1 - 28 kHz	Schrift	Keine	30 - 100 kHz
			schwarz	Auswirkungen	
			1 - 28 kHz		
Märklin®	DIP schwarz	1 - 28 kHz	DIP rot	Keine	30 - 100 kHz
			1 - 58 kHz	Auswirkungen	
Märklin®	DIP rot	1 - 75 kHz	DIP schwarz	Keine	30 - 100 kHz
			1 - 58 kHz	Auswirkungen	

Tabelle 12-1: I2C-Bus Frequenzübersicht

#### 12.2.4 Softwareversion des Connect6021®

Die <Connect6021®-Softwareversion> ermöglicht die Unterdrückung der Märklin-

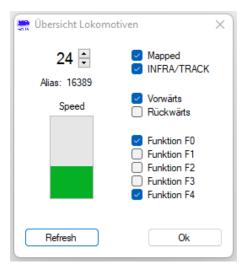


Softwareaktualisierung. Das Modul imitiert ein originales Connect6021®. Sollte Märklin ein Softwareupdate durchführen, wird dies in der CS2/3® angezeigt. Wird die Connect6021®-Softwareversion auf diese Märklin®-

Versionsnummer programmiert, unterbleibt ein weiterer Aufruf zur Aktualisierung.



#### 12.2.5 Aktueller Zustand der möglichen 80 Lokomotiven



Dieses Fenster wird über <Lokeigenschaften> aktiviert. Es bietet die Möglichkeit, die aktuellen Infos zu den maximal 80 Lokomotiven der 6021® auszulesen.

Gezeigt wird neben der Lokadresse 1 bis 80 eine Alias-Adresse. Diese weicht von der Lokadresse ab, wenn die Lok gemappt wurde (vgl. Loc-ID in der CAN-

Protokollbeschreibung). Wird die Lok über den TRACK-Eingang gesteuert, wird dies ebenfalls angezeigt. Die weiteren Checkboxen zeigen die Infos zu Fahrtrichtung und Funktion. Eine Progressbar visualisiert die Geschwindigkeit.

Wird die Lokadresse über die beiden Pfeile angepasst, wird die entsprechende Lok aktualisiert aus dem Modul ausgelesen. Gleiches gilt beim Drücken von **<Refresh>**.

#### 12.2.6 Aktualisierung von Daten

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. 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.

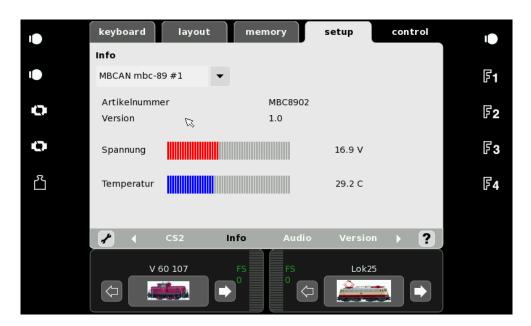


Abbildung 12-2: <Info>-Bereich



Abbildung 12-3: <Info>-Konfigurationsbereich



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.

Im <Info>-Konfigurationsbereich unter der Rubrik 6021 befindet sich die Mapping-Funktion von Loks auf die 80 möglichen Lokadressen der 6021® (siehe Abbildung 12-4: <Lokauswahl>-Bereich). Wie dies funktioniert, ist der Bedienungsanleitung des Connect6021® zu entnehmen.



Abbildung 12-4: <Lokauswahl>-Bereich



#### 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 **<Connect6021>** 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 12-5: <Info>-Bereich).

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

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

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

In einem weiteren Untermenü befindet sich die Mapping-Funktion von Loks auf die 80 möglichen Lokadressen der 6021® (siehe Abbildung 12-7: <Lokauswahl>-Bereich). Wie dies funktioniert, ist der Bedienungsanleitung des Connect6021® zu entnehmen.

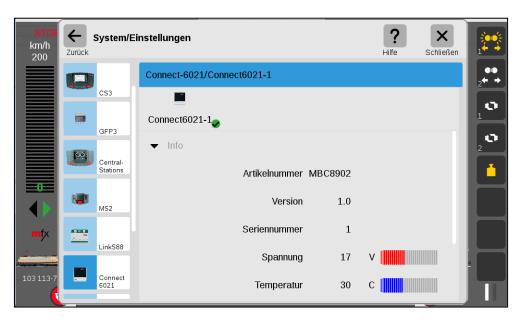


Abbildung 12-5: <Info>-Bereich



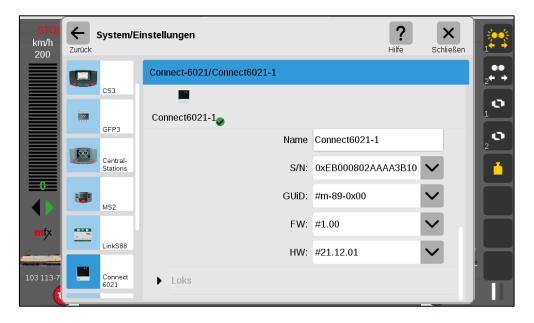


Abbildung 12-6: <Einstellungen>-Bereich

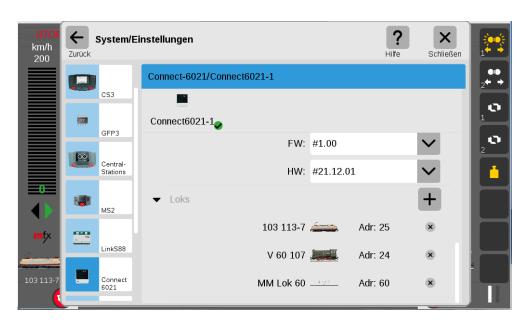


Abbildung 12-7: <Lokauswahl>-Bereich



# 13 Modulbilder



Abbildung 13-1: Fertiges Modul inkl. Gehäuse

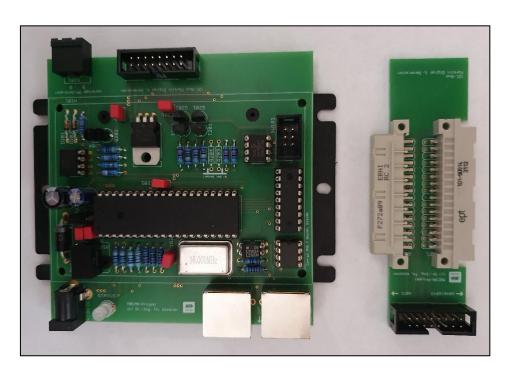


Abbildung 13-2: bestückte Platine



## 14 Systemarray-Belegung für Eigenentwicklungen

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

### 14.1 Allgemeiner Bereich zum Modul

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

```
//= Systemarray mit Parametern zum Zustand
// MBC_ARRAY_START
                              Start des belegten Systemarrays
#define MBC ARRAY START
// 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
                              Index Start Seriennummer
// MBC S SNR
                              Seriennummer
// MBC_SNR
#define MBC L SNR
#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
                              (MBC_L_SNR + MBC_S_SNR)
#define MBC_S_ART
                              sys_array[MBC_S_ART]
#define MBC_ART_1
#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
#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
#define MBC_S_HW
                                   (MBC_L_SW + MBC_S_SW)
#define MBC_HW_1
                                   sys_array[MBC_S_HW]
                                   sys_array[MBC_S_HW + 1]
sys_array[MBC_S_HW + 2]
#define MBC HW 2
#define MBC HW 3
                                   sys array[MBC S HW + 3]
#define MBC HW 4
                                   sys array[MBC S HW + 4]
#define MBC HW 5
                                   sys_array[MBC_S_HW + 5]
#define MBC_HW_6
// MBC L NAMEBLOCK
                                   Laenge des Modulnamens
// MBC S NAMEBLOCK
                                   Index Start Modulname
                                   Name des Moduls
// MBC NAME
#define MBC_L_NAMEBLOCK
#define MBC S NAMEBLOCK
                                   (MBC_L_HW + MBC_S_HW)
                                   sys_array[MBC_S_NAMEBLOCK]
#define MBC_NAME
// MBC INFO ENDE
                                   Ende des Neuanmeldeblocks
                                   (MBC_L_NAMEBLOCK + MBC_S_NAMEBLOCK - 1)
#define MBC_INFO_ENDE
// 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
#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
#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
                                   CS2-Geraetekennung LOW
// MBC_KN_L
// MBC_AK_H
                                   CS2-Autokennung HIGH
```



```
// MBC_AK_L
                                  CS2-Autokennung LOW
// MBC_CS2_GER
                                  CS2-Geraetegruppe
#define MBC_L_KN
#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
// MBC_SPG_2
                                  Spannung 10er Digit
                                  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
                                  (MBC L KN + MBC S KN)
#define MBC S PARA
#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]
                                  sys_array[MBC_S_PARA + 3]
#define MBC SPG 3
                                  sys_array[MBC_S_PARA + 4]
#define MBC TMP 1
                                  sys_array[MBC_S_PARA + 5]
#define MBC TMP 2
#define MBC_TMP_3
                                  sys_array[MBC_S_PARA + 6]
                                  Ende des Allgemeinblocks
// MBC_ALLG_ENDE
#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_PARA
                                  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)
                                  (MBC L GENPURP + MBC S GENPURP)
       #define MBC ARRAY MAX
#endif
// MBC_ARRAY_ENDE
                                  Ende des Systemarrays
                                  (MBC ARRAY MAX - 1)
#define MBC ARRAY ENDE
Listing 14-1: Modulparameter
```

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

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.

Es gibt eine Ablagestruktur für die Lokdaten (80 Lokomotiven), der Mapping-Tabelle für die Zubehörartikeladressen MM/DCC sowie der Info, ob nur eine spezifisches Mapping oder ein direktes Routing erlaubt ist. Außerdem gibt es eine Speicherzelle für die Blocknummer der angeschlossenen Keyboards von 1 bis 4 (1 = Keyboard 1 ... 16 bis 4 = Keyboard 49 ... 64, entspricht dem Adressraum bis 1024)

Dargestellt ist die C-Schreibweise:

```
//= Sonderbelegung des Systemarrays nach dem Allgemeinblock
//-----
                    Laenge der Softwareversion des L88
// MBC 89 L 6021 SW
// MBC 89 S 6021 SW
                       Index Start Softwareversion des L88
// MBC_89_6021_SW
                       Array mit der Softwareversion des L88
#define MBC 89 L 6021 SW
#define MBC 89 S 6021 SW
                              MBC S GENPURP
#define MBC 89 6021 SW 1
                               modul.sys array[MBC 89 S 6021 SW]
#define MBC_89_6021_SW_2
                              modul.sys_array[MBC_89_S_6021_SW + 1]
                     Laenge lokspezifische Parameter
Index Start lokspezifische Parameter
// MBC_89_L_LOKS
// MBC_89_S_LOKS
// MBC_89_STEP
                        Schrittweite Protokoll
// MBC_89_LOKS_ITEMS
                        Anzahl der Eigenschaften von MBC_89_LOKS
// MBC_89_LOKS
                        Lok-Speicher
//
                               *( mm_lok )
                                              = 0x00; Speed
                               *(mm_lok + 1) = 0x00; Dir
//
                               *( mm_lok + 2 ) = 0x00; F0
//
//
                               *( mm_lok + 3 ) = 0x00; F1...F4
                               *( mm_lok + 4 ) = 0x00; GUiD 4
//
                               *( mm_lok + 5 ) = 0x00; GUiD 3
//
                               *( mm_lok + 6 ) = 0x00; GUiD 2
//
//
                               *( mm_lok + 7 ) = 0x00; GUiD 1
                               *( mm_lok + 8 ) = 0x00; 6021Connect
//
//
                               *( mm_lok + 9 ) = 0x00; INFRAConnect
#define MBC_89_L_LOKS
#define MBC_89_S_LOKS
                              (MBC_89_S_6021_SW + MBC_89_L_6021_SW)
#define MBC_89_LOKS
                              modul.sys_array[MBC_89_S_LOKS]
#define MBC_89_STEP
                              68UL
#define MBC_89_LOKS_ITEMS
                              10
#define MBC_89_LOK_SPEED
                               *( mm lok )
#define MBC_89_LOK_DIR
                              *( mm lok + 1 )
#define MBC_89_LOK_F0
                              *(mm lok + 2)
#define MBC_89_LOK_FX
                              *(mm lok + 3)
#define MBC_89_LOK_ADDR_4
                              *(mm lok + 4)
#define MBC 89 LOK ADDR 3
                              *(mm lok + 5)
#define MBC 89 LOK ADDR 2
                              *(mm lok + 6)
#define MBC 89 LOK ADDR 1
                              *( mm_lok + 7
#define MBC 89 LOK MAPPED
                              *( mm_lok + 8 )
```



```
#define MBC_89_LOK_INFRA
                                  *(mm_lok + 9)
                           Laenge magnetartikelspezifische Parameter
// MBC_89_L_MAGNET
// MBC_89_S_MAGNET
                           Index Start magnetartikelspezifische Parameter
// MBC_89_MAGNET
                           Protokollspeicher MM2/DCC fuer Keyboard
#define MBC_89_L_MAGNET
                                  (MBC_89_L_LOKS + MBC_89_S_LOKS)
#define MBC_89_S_MAGNET
#define MBC_89_MAGNET
                                  modul.sys_array[MBC_89_S_MAGNET]
// MBC 89 L ZENTRALE Laenge Angaben zur Zentrale
// MBC_89_S_ZENTRALE Index Angaben zur Zentrale
// MBC_89_ZENTRALE
                           Angaben zur Zentrale
#define MBC_89_L_ZENTRALE
#define MBC 89 S ZENTRALE
                                  (MBC 89 L MAGNET + MBC 89 S MAGNET)
#define MBC 89 ZENTRALE
                                  modul.sys array[MBC 89 S ZENTRALE]
// MBC 89 L BASIS
                                  Laenge Angabe zur MM-Basis
// MBC 89 S BASIS
                                  Index Angaben zur MM-Basis
// MBC_89_BASIS
                                  Angaben zur MM-Basis
#define MBC 89 L BASIS
#define MBC 89 S BASIS
                                  (MBC 89 L ZENTRALE + MBC 89 S ZENTRALE)
#define MBC_89_BASIS
                                  modul.sys_array[MBC_89_S_BASIS]
// MBC 89 L I2C
                                  Laenge Angabe zur I2C-Frequenz
// MBC 89 S I2C
                                  Index Angaben zurI2C-Frequenz
// MBC_89_I2C
                                  Angaben zur I2C-Frequenz
#define MBC 89_L_I2C
#define MBC 89 S I2C
                                  (MBC 89 L BASIS + MBC 89 S BASIS)
#define MBC_89_I2C
                                  modul.sys_array[MBC_89_S_I2C]
```

Listing 14-3: Funktionsparameter

Listing 14-4: EEPROM-Indizes Funktionen

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

```
//-----
//= Parameter fuer die EEPROM-Steuerung
//-----
#define EE 6021 SW
                  0x0A
                           // ID Block SW-Version 6021Connect
                           // ID Block Protokoll Magnetartikel
#define EE MM DCC
                  0x0B
                           // ID Block Angabe zur Zentrale
#define EE_ZENTRALE
                  0x0C
#define EE_BASIS
                  0x0D
                           // ID Block Basis x von 4
#define EE_I2C
                  0x0E
                           // ID Block I2C-Frequenz
```

© Dr.-Ing. Thomas Wiesner 2007 – 2024 ● August-Bebel-Straße 7 ● 59174 Kamen ● info@mbcan.de



#### 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 - 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 - Alivemeldung durch PC angefordert
MD - Acknowledge des Moduls auf PC_ALIVE
// PC ALIVE
// MD 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 - Acknowledge des Moduls auf PC_UPGRADE
// MD 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-Stellungsmeldung
// MD_S88
#define PC_DB_H
                         0x00
#define PC_DB_M
                         0x01
                         0x02
#define PC_DB_L
#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)

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 PC an Modul während des Neuanmeldeprozesses

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



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

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



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

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



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

Befehl	PC_MD_DEL
Sender	PC
Loc-ID	0x00001808
Funktion	Modul aus Datenbank entfernen
Beschreibung	Das Modul wurde aus der Datenbank entfernt und kann sich an dieser Datenbank auch nicht mehr neu anmelden. Wird in der Regel nur bei Modulen verwendet, die sich in der Datenbank befinden aber nicht mehr am Bus angeschlossen werden sollen. Wird nur einmal gesendet, wenn das Modul im Parametriercenter gelöscht wird. Ist das Modul nicht am Bus und wird nach einem Neustart der Software wieder am Bus angeschlossen, meldet es sich nicht mehr neu an, es sei denn, die Datenbank wird neu erstellt.
Genutzte Datenbytes	D0 - D3: Loc-ID 0x00001808 D4 - D7: GUiD des Moduls
Nachricht	
inachricht	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 werten 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 werten 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 1 131, BUS 2 3262, BUS 3 6393)
	D6: Kontaktnummer (116)
	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.

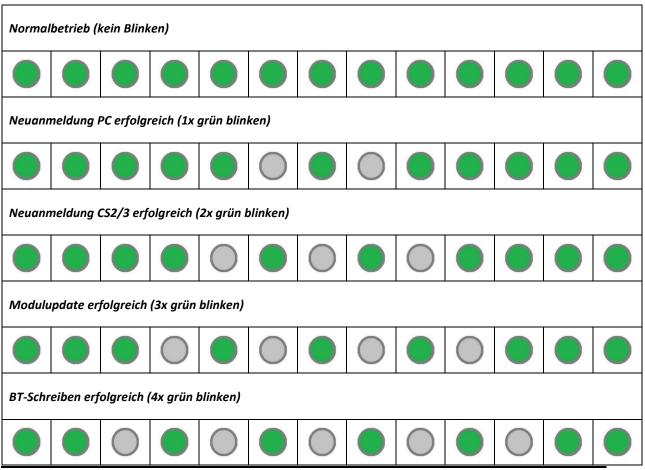
Die LED-Anzeige wird mit 500 ms getaktet und ist je Botschaft 7 s lang; d.h., dass im Grundsatz 6 Blinkschematas zu je einer der drei Farben ROT, ORANGE und GRÜN möglich sind, Mischungen mal ausgenommen. Die Farbe der LED sind drei Klassen von Botschaften resp. Stati zugeordnet:

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

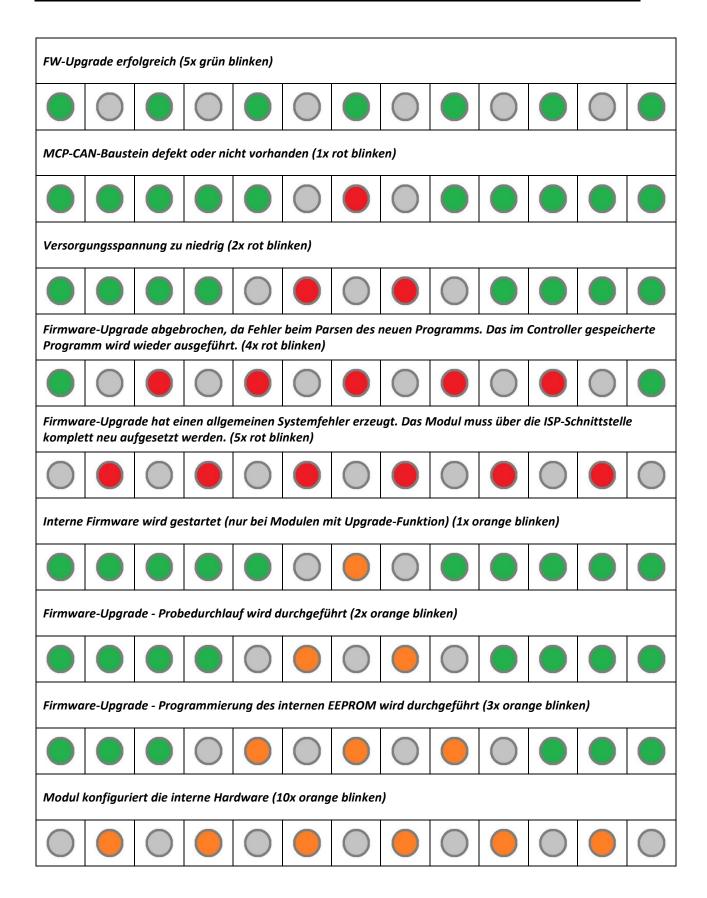
Unregelmäßiges Aufflackern der orangenen LED-Farbe bei ansonsten grüner LED zeigt Datentraffic auf dem CAN-Bus an bzw. während des Upgrades aus dem externen EEPROM entsprechende Schreib-/Lesezugriffe. Damit ist erkennbar, ob der auf dem Modul implementierte CAN-Baustein Nachrichten verarbeitet.

Stand heute sind folgende Post-Codes implementiert:

Tabelle 16-1: LED-Signalbedeutung









#### 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, Bilder, 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