

# Kommunikationsprotokoll

**Graphical User Interface Prozessor (GUI)**

<->

**Gleisformat Prozessor (GFP)**

**über CAN**

**transportierbar über Ethernet**

**(Version vom 26.9.2008)**



**Gebr. Märklin & Cie. GmbH, Göppingen**

Copyright (C) 2008 Gebr. Märklin & Cie. GmbH



## Inhaltsverzeichnis:

1	CAN Anschluss .....	4
1.1	Allgemeines.....	4
1.2	CAN Grundformat .....	4
1.3	Feldergrundbeschreibung.....	4
1.3.1	Priorität (Prio).....	4
1.3.2	Kommando.....	4
1.3.3	Response.....	4
1.3.4	Hash.....	5
1.3.5	Sonstiges .....	5
1.3.6	Übertragung der CAN Kommandos via Ethernet .....	5
1.4	Allgemeines.....	7
1.4.1	"Adressen" im System .....	7
1.4.2	Definition der Teilnehmerkennungen (UID) .....	7
1.4.3	Einbindung bestehender Gleisprotokolle, Loc-ID .....	7
2	System-Befehle .....	9
2.1	Befehl: System Stopp.....	9
2.2	Befehl: System Go .....	9
2.3	Befehl: System Halt.....	10
2.4	Befehl: Lok Nothalt.....	10
2.5	Befehl: Lok Zyklus Beenden .....	11
2.6	Befehl: System MFX Neuanmeldezähler setzen .....	12
2.7	Befehl: System Überlast.....	12
2.8	Befehl: System Status.....	13
3	Verwaltung .....	14
3.1	Befehl: Lok Discovery .....	14
3.2	Befehl: MFX Bind .....	15
3.3	Befehl: MFX Verify .....	15
3.4	Befehl: Lok Geschwindigkeit.....	16
3.5	Befehl: Lok Richtung .....	17
3.6	Befehl: Lok Funktion .....	18
3.7	Befehl: Read Config.....	18
3.8	Befehl: Write Config.....	19
4	Zubehör-Befehle.....	21
4.1	Befehl: Zubehör Schalten .....	21
5	Software Befehle .....	22
5.1	Befehl: Teilnehmer Ping.....	22
5.2	Befehl: Statusdaten Konfiguration .....	22
6	Abkürzungen .....	25

# 1 CAN Anschluss

## 1.1 Allgemeines

Der CAN Bus dient als Kommunikationsnetz der Steuergeräte bei Märklin Digital. Ziel ist, allen Geräten zur Steuerung einer Modellbahnanlage ein einheitliches Kommunikationsmedium zur Verfügung zu stellen.

Mittels CAN Meldungen werden Steueraufgaben übermittelt.  
Mittels CAN Streams werden Updates und Konfigurationsdaten übertragen.

Parameter für den Can - Anschluss: Datenrate ist 250 KBit/s, maximale Buslänge ist 100m.

## 1.2 CAN Grundformat

Das CAN Protokoll schreibt vor, dass Meldungen mit einer 29 Bit Meldungskennung, 5 Bit Meldungslänge sowie bis zu 8 Datenbyte bestehen. Die Meldungskennung wird aufgeteilt in die Unterbereiche Priorität (Prio), Kommando (Command), Response und Hash. Die Kommunikation basiert auf folgendem Datenformat:

Meldungskennung				DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Prio	Command	Resp.	Hash	DLC	D-Byte 0	D-Byte 1	D-Byte 2	D-Byte 3	D-Byte 4	D-Byte 5	D-Byte 6	D-Byte 7
2+2 Bit	8 Bit	1 Bit	16 Bit	4 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit
Message Prio	Kommando Kennzeichnung	Cmd / Resp.	Kollisionsauflösung	Anz. Datenbytes	Daten	....						

## 1.3 Feldergrundbeschreibung

### 1.3.1 Priorität (Prio)

Ist Bestandteil der Meldungskennung (Höchstwertigste Bits)  
[Die Priorität wird in V1.0 nicht verwendet. In der V1.0 muss 0b0000 verwendet werden.](#)

Für V2.0 oder später:  
Bestimmt die Priorisierung der Meldung auf dem CAN Bus:  
Prio 1: Stopp / Go / Kurzschluss-Meldung  
Prio 2: Rückmeldungen  
Prio 3: Lok anhalten (?)  
Prio 4: Lok / Zubehörbefehle  
Rest frei

Die Priorisierung wird von den Teilnehmern nicht als Teil der Meldung verstanden, sondern dient zum Priorisieren der Meldung auf dem CAN Bus.  
Empfänger sollen die Priorität nicht auswerten. Die Priorität darf nicht zur Unterscheidung von Befehlen verwendet werden.

### 1.3.2 Kommando

Ist Bestandteil der Meldungskennung  
Bestimmt das vom Endgerät auszuführende, bzw. das ausgeführte Kommando.  
Kommandowerte sind eindeutig definiert.

### 1.3.3 Response

Ist Bestandteil der Meldungskennung  
Bestimmt, ob CAN Meldung eine Anforderung oder Antwort auf eine vorhergehende Anforderung ist.  
Grundsätzlich wird eine Anforderung ohne ein gesetztes Response Bit angestoßen. Sobald ein Kommando ausgeführt wurde, wird es mit gesetztem Response Bit, sowie dem ursprünglichen Meldungsinhalt oder den

angefragten Werten, bestätigt. Der Teilnehmer am Bus, welcher die Meldung ausgeführt hat, bestätigt das Kommando.

Ein Initiator einer Meldung muss Sorge dafür tragen, dass die gewünschte Aktion tatsächlich ausgeführt wird. Meldungen werden nicht gesichert über den CAN-Bus übertragen. Der Empfang einer Meldung wird nicht bestätigt. Die Rückmeldung wird nicht gesichert übertragen.

#### 1.3.4 Hash

Ist Bestandteil der Meldungskennung.

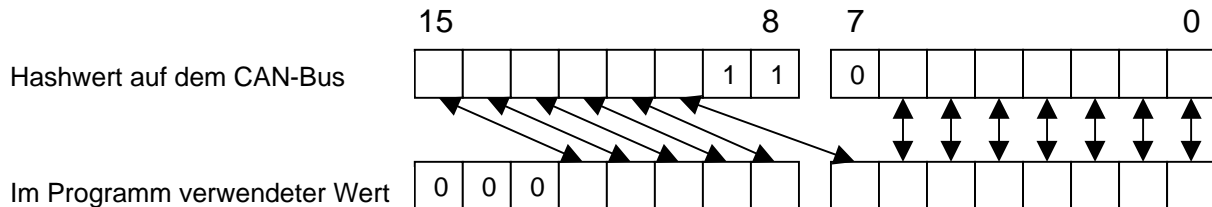
Der Hash erfüllt eine Doppelfunktion.

- Primär dient er zur Kollisionsauflösung der Meldungen und zur Sicherstellung der Kollisionsfreiheit zum CS1 Protokoll.
- Sekundär zeigt er die Folgenummer einer Datenübertragung an.

Der Hash dient dazu, die CAN Meldungen mit hoher Wahrscheinlichkeit kollisionsfrei zu gestalten. Dieser 16 Bit Wert wird gebildet aus der UID des Endgerätes. Berechnung: 16 Bit High UID XOR 16 Bit Low der UID. Danach werden die Bits entsprechend zur CS1 Unterscheidung gesetzt (S.U.).

Im CAN Protokoll der CS1 wird der Wert 6 für den "com-Bereich der ID"(?), dies sind die Bits 7..9, d.h. Highest Bit im Lowest-Byte (0b0xxxxxx) und die beiden Bits darüber (0bxxxxx11), nicht benutzt. Diese Bitkombination wird daher zur Unterscheidung fest im Hash verwendet.

Wird der Hash zur Kennzeichnung der Paketnummer verwendet, so werden diese Bits bei der Berechnung der Paketnummer ausgeblendet. D.H. bei der 16 Bit Zahl werden die Bits 7 bis 9 ausgeblendet, die obersten 3 Bits sind 0.



Der Wertebereich verringert sich entsprechend auf 8192.

#### 1.3.5 Sonstiges

- In der Kommunikation werden keine Sender + Empfänger Adresse verwendet. Jeder Teilnehmer des Kommunikationsbusses muss anhand des Inhalts der Meldung erkennen, welche Aktion zu tätigen ist.
- In der Kommunikation werden keine Remoteframes (=CAN-ID anfragen statt mit Daten senden) verwendet. Im allgemeinen sind die Teilnehmer so konfiguriert, dass diese nicht empfangen werden.
- Die Byte-Order in den Meldungen ist immer Motorola Big Endian.

#### 1.3.6 Übertragung der CAN Kommandos via Ethernet

Auf der CS2 kann - über das Setup / IP - Einstellungen - das Can-UDP-Gateway gestartet werden. Dort kann eine IP-Adresse (auch Broadcast) spezifiziert werden, an die das Gateway sendet. Die Portadressen sind über die Oberfläche nicht einstellbar und werden fest auf die Ports 15731 und 15730 gesetzt.

Funktionsweise: Wenn gestartet, lauscht das Gateway auf dem Ethernet Empfangsport 15731. Es verwirft alle UDP-Pakete, die eine Länge ungleich 13 haben. Pakete der Länge 13 werden als Can-Bus-Pakete interpretiert: 4 Byte Can-Bus-Id (BigEndian oder Network-Order), 1 Byte Länge und 8 Byte Daten, die ggf. mit Nullbytes aufzufüllen sind. Dieses Paket wird dann als Can-Bus-Botschaft auf den Can-Bus gegeben. Nicht abbildbare Bits oder Bytes auf dem CAN-Bus werden nicht beachtet und sollten auf "0" gesetzt werden.

Umgekehrt liest das Gateway alle Can-Bus-Botschaften, wandelt sie in analoger Weise in UDP-Pakete der Länge 13 um und verschickt diese an die spezifizierte IP-Adresse und den Sendeport (15730).

Beispielkonfiguration im lokalen Netz mit dem Netzwerksegment 192.168.2.0

CS2: (192.168.2.20) empfängt auf Port 15731, sendet an die Broadcast-Adresse 192.168.2.255:15730.

PC1: (192.168.2.10) empfängt auf Port 15730 sendet an CS2 (192.168.2.20:15731)

PC2: (192.168.2.11) empfängt auf Port 15730 sendet an CS2 (192.168.2.20:15731)

---

Im Ethernet werden immer Pakete mit 13 Bytes übertragen, unabhängig von der CAN - Datagramgröße  
Pakete anderer Länge verwirft der CAN - Ethernet - Gateway.

Die Bytes in der CAN- Botschaft werden folgendermaßen in dem UDP-Paket eingepackt:

- Bytes 1 bis 4 sind die Meldungskennung.
- Byte 5 entspricht dem DLC der CAN-Meldung.
- Bytes 6 - 13 sind die entsprechenden Nutzdaten. Dabei nicht benötigte Bytes sind mit 00 zu füllen.

## 1.4 Allgemeines

### 1.4.1 "Adressen" im System

Der gesamte Adressraum hat  $2^{32}$  Adressen (0x0000 0000 - 0xFFFF FFFF), diese sind rund 4 Milliarden Adressen. Diese werden auch als UID (Universal Identifier) bezeichnet. Je nach eingesetztem Protokoll hat jedoch eine UID eine andere Bedeutung. Die UIDs werden so vergeben, dass 2 Geräte oder sonstige Teilnehmer dieselbe Kennung haben. Die Verwaltung des Adressbereiches geschieht durch Märklin.

### 1.4.2 Definition der Teilnehmerkennungen (UID)

Im System besitzt jeder adressierbare Teilnehmer eine eindeutige 32 Bit Adresse, die UID. Dabei werden folgende UID unterschieden:

Geräte UID	Eindeutig vergebene Universal ID.
Loc ID	(=Local ID, nicht Lokomotive ID) Aus dem Protokoll und der Adresse berechnete Lokale ID.
MFX UID	MFX Universal UID.

#### Spezialfall einer Geräte - UID:

Es wird gefordert, dass jeder CAN Teilnehmer unter seiner UID ansprechbar sein muss. Bestimmte Befehle richten sich an eben diesen Teilnehmer. Neben dieser Teilnehmerkennung muss dieser ebenso auf die Broadcastadresse reagieren.

Die UID 0x00000000 ist die Broadcastadresse. Diese signalisiert, dass alle CAN Teilnehmer denselben Befehl abarbeiten sollen. Ist der Befehl für einen CAN Teilnehmer nicht gültig, so muss dieser den Befehl auch nicht beantworten oder ausführen.

Die UID 0xFFFFFFFF ist ungültig und steht für eine nicht initialisierte UID des Endgerätes.

### 1.4.3 Einbindung bestehender Gleisprotokolle, Loc-ID

Der Adressraum hat rund 4 Milliarden verfügbare Adressen. Von diesem Adressraum wird ein Teil (Adresse 0 - 65536) für die Einbindung bestehender Protokolle verwendet: In diesem reservierten Bereich werden die bestehenden Digitalprotokolle eingebettet, repräsentiert durch die Loc-ID. Durch Ihre Lage ergibt sich das Protokoll. Aufgeführt sind die unteren 2 Byte der Loc-ID bei diesen Protokollen, die oberen sind = 0x0000.

So ergibt sich folgendes Adressschema:

StartAdresse	EndAdresse	Protokoll
0x0000	0x03FF	MM1,2 Loks und Funktionsdecoder (20 & 40 kHz, 80 & 255 Adressen)
0x0400	0x07FF	Reserviert
0x0800	0x0BFF	Reserviert
0x0C00	0x0FFF	Reserviert
0x1000	0x13FF	Reserviert
0x1400	0x17FF	Reserviert
0x1800	0x1BFF	Frei für Privatanwender / Clubs
0x1C00	0x1FFF	Frei für Firmen
0x2000	0x23FF	Reserviert
0x2400	0x27FF	Reserviert
0x2800	0x2BFF	Reserviert
0x2C00	0x2FFF	Reserviert für Traktionen (interne GUI Kennungen)
0x3000	0x33FF	MM1,2 Zubehörtartikeldecoder (40 kHz, 320 & 1024 Adressen)
0x3400	0x37FF	Reserviert
0x3800	0x3BFF	Reserviert
0x3C00	0x3FFF	Reserviert
0x4000	0x7FFF	MFX
0x8000	0xBFFF	Reserviert
0xC000	0xFFFF	Reserviert

Beispiel (Hex):

Märklin Motorola mit Adresse 2:	Basis: 00 00 00 00	Plus Adresse:	00 00 00 02
MM2 Zubehör mit Adresse 3:	Basis: 00 00 30 00	Plus Adresse:	00 00 30 03



## 2 System-Befehle

Systembefehle betreffen den Gleis Format Prozessor direkt und bestimmen die Funktionsweise, bzw. die Zustände. Des weiteren werden Zustände des Gleis Format Prozessor weitergemeldet.

Die Befehle enthalten, neben einer UID, auch ein Kommandobyte. Dieses dient zur Kennzeichnung des auszuführenden Systembefehls.

### 2.1 Befehl: System Stopp

Kennung:

Systembefehl (0x00, in CAN-ID: 0x00)

Sub-CMD:

System Stopp (0x00)

Format:

Prio	Command	Resp.	Hash	DLC	D-Byte 0	D-Byte 1	D-Byte 2	D-Byte 3	D-Byte 4	D-Byte 5	D-Byte 6	D-Byte 7
2+2 Bit	8 Bit	1 Bit	16 Bit	4 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit
Message Prio	System-befehl	0/1		5	Ziel Geräte-UID				Sub-CMD			
					High			Low	Stop			

Beschreibung:

Der angesprochene Gleis Format Prozessor stoppt den Betrieb auf Haupt- und Programmiergleis. Es wird keine elektrische Energie mehr geliefert. Alle Fahrstufen/Funktionswerte und Einstellungen werden behalten.

Als Sonderform wird auf einen generellen Stopp Befehl geachtet, welcher alle Gleis Format Prozessor betrifft. Hierzu wird die spezielle Ziel Geräte UID 0x00 00 00 00 verwendet.

Beispiel (Hex):

00000300 5 00 00 00 00 00 Stopp an alle

00000300 5 43 53 32 08 00 Stopp an bestimmten Teilnehmer (CS2 mit SNr. 08)

Antwort:

Ursprünglicher Befehl mit gesetztem Response Bit

Besonderheiten:

Stopp wird immer von einem Bediengerät ausgelöst.

Es wird keine Fahrstufe 0 oder Nothalt gesendet. Nach System Go fahren alle Lokomotiven wieder mit der alten Einstellung weiter oder bleiben stehen. Dieses Verhalten wird durch den Decoder bestimmt.

### 2.2 Befehl: System Go

Kennung:

Systembefehl (0x00, in CAN-ID: 0x00)

Sub-CMD:

System Go (0x01)

Format:

Prio	Command	Resp.	Hash	DLC	D-Byte 0	D-Byte 1	D-Byte 2	D-Byte 3	D-Byte 4	D-Byte 5	D-Byte 6	D-Byte 7
2+2 Bit	8 Bit	1 Bit	16 Bit	4 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit
Message Prio	System-befehl	0/1		5	Ziel Geräte-UID				Sub-CMD			
					High			Low	Stop			

Beschreibung:

Der Gleis Format Prozessor aktiviert den Betrieb und liefert elektrische Energie. Es werden alle evtl. noch vorhandenen bzw. gespeicherten Geschwindigkeitsstufen/Funktionen wieder gesendet.

Als Sonderform wird auf einen generellen Go Befehl geachtet, welcher alle Gleis Format Prozessor betrifft. Hierzu wird die spezielle Ziel Geräte UID 0x00 00 00 00 verwendet.

Beispiel (Hex):

00000300 5 00 00 00 00 01    Go an alle.  
 00000300 5 43 53 32 08 01    Go an bestimmten Teilnehmer.

Antwort:

Ursprünglicher Befehl mit gesetztem Response Bit.

Besonderheiten:

Go wird immer von einem Bediengerät ausgelöst.

## 2.3 Befehl: System Halt

Kennung:

Systembefehl (0x00, in CAN-ID: 0x00)

Sub-CMD:

System Halt (0x02)

Format:

Prio	Command	Resp.	Hash	DLC	D-Byte 0	D-Byte 1	D-Byte 2	D-Byte 3	D-Byte 4	D-Byte 5	D-Byte 6	D-Byte 7
2+2 Bit	8 Bit	1 Bit	16 Bit	4 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit
Message Prio	System-befehl	0/1		5	Ziel Geräte-UID				Sub-CMD			
					High			Low	Halt			

Beschreibung:

Allen Lokomotiven wird befohlen, inklusive ABV, anzuhalten (Fahrstufe 0). Digitalsignal weiterhin auf Gleis, danach werden keine weiteren Kommandos auf das Gleis gesendet. Elektrische Energie steht weiterhin zur Verfügung. Sinnvoll zum definierten Herunterfahren der Anlage.

Als Sonderform wird auf einen generellen Halt Befehl geachtet, welcher alle Gleis Format Prozessor betrifft. Hierzu wird die spezielle Ziel Geräte UID 0x00 00 00 00 verwendet.

Beispiel (Hex):

00000300 5 00 00 00 00 02    Halt an alle.  
 00000300 5 43 53 32 08 02    Halt an bestimmten Teilnehmer.  
 00010300 5 43 53 32 08 02    Antwort des Teilnehmers auf Halt.

Antwort:

Ursprünglicher Befehl mit gesetztem Response Bit.

Besonderheiten:

Halt wird immer von einem Bediengerät ausgelöst.

## 2.4 Befehl: Lok Nothalt

Kennung:

Systembefehl (0x00, in CAN-ID: 0x00)

Sub-CMD:

Lok Nothalt (0x03)

Format:

Prio	Command	Resp.	Hash	DLC	D-Byte 0	D-Byte 1	D-Byte 2	D-Byte 3	D-Byte 4	D-Byte 5	D-Byte 6	D-Byte 7
2+2 Bit	8 Bit	1 Bit	16 Bit	4 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit
Message Prio	System-befehl	0 1		5	Loc-ID				Sub-CMD			
					High			Low	Lok Halt			

Beschreibung:

Nothalt bzw. Soforthalt der Lokomotive, je nach Gleisprotokoll. Es muss eine Lokomotive angegeben werden, die schon durch ein Kommando angesprochen wurde. Ist diese Lok nicht im Zyklus, wird diese dadurch auch nicht aufgenommen.

Beispiel (Hex):

00000300 5 00 00 00 48 03 Lok Nothalt MM2 72  
 00000300 5 00 00 40 05 03 Lok Nothalt MFX SID 5

Antwort:

Ursprünglicher Befehl mit gesetztem Response Bit

Besonderheiten:

Nothalt ist nicht als Fahrstufe implementiert, damit keine Fehlinterpretation bei verschiedenen Protokollen geschieht.  
 Nothalt wird immer von einem Bediengerät ausgelöst.

## 2.5 Befehl: Lok Zyklus Beenden

Kennung:

Systembefehl (0x00, in CAN-ID: 0x00)

Sub-CMD:

Lok Zyklus Stop (0x04)

Format:

Prio	Command	Resp.	Hash	DLC	D-Byte 0	D-Byte 1	D-Byte 2	D-Byte 3	D-Byte 4	D-Byte 5	D-Byte 6	D-Byte 7
2+2 Bit	8 Bit	1 Bit	16 Bit	4 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit
Message Prio	System-befehl	0 1		5	Loc-ID				Sub-CMD			
					High			Low	Zyklus			

Beschreibung:

Lok aus Verwaltungsliste des Gleis Format Prozessor löschen. Die Lok erhält keine Befehle mehr (kein Refreshzyklus).  
 Erst wieder bei Setzen einer Fahrstufe oder Ansprechen mittels Funktionen wird Lok wieder angesprochen und mit Gleistelegammen versorgt.

Beispiel (Hex):

00000300 5 00 00 00 48 04 Zyklus Ende MM2 72  
 00000300 5 00 00 40 05 04 Zyklus Ende MFX SID 5  
 00000300 5 43 53 32 08 04 Zyklus Ende alle Loks im Gleis Format Prozessor mit UID=43 53 32 08  
 00000300 5 00 00 00 00 04 Zyklus Ende alle Loks auf allen Gleis Format Prozessors(UID 00 00 00 00 Broadcast an alle).

Antwort:

Ursprünglicher Befehl mit gesetztem Response Bit.

Besonderheiten:

Wird immer von einem Bediengerät ausgelöst.

Entspricht die Loc-ID der UID des Gleis Format Prozessor, so wird die gesamte Lokverwaltungstabelle des Gleis Format Prozessors gelöscht.

## 2.6 Befehl: System MFX Neuansmeldezähler setzen

Kennung:

Systembefehl (0x00, in CAN-ID: 0x00)

Sub-CMD:

Neuanmeldezähler (0x09)

Format:

Prio	Command	Resp.	Hash	DLC	D-Byte 0	D-Byte 1	D-Byte 2	D-Byte 3	D-Byte 4	D-Byte 5	D-Byte 6	D-Byte 7
2+2 Bit	8 Bit	1 Bit	16 Bit	4 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit
Message Prio	System-befehl	1		7	Ziel Geräte-UID				Sub-CMD	Neuanmeldezähler		
				High				Low	MFX NAZ	High	Low	

Beschreibung:

Neuanmeldezähler des Gleis Format Prozessor verändern. Für MFX Subsystem.

Beispiel (Hex):

00000300 7 43 53 32 08 09 00 02 Setzen Neuanmeldezähler auf 2.

Antwort:

Ursprünglicher Befehl mit gesetztem Response Bit.

Besonderheiten:

Wird immer von einem Bediengerät ausgelöst.

## 2.7 Befehl: System Überlast

Kennung:

Systembefehl (0x00, in CAN-ID: 0x00)

Sub-CMD:

Überlast (0x0A)

Format:

Prio	Command	Resp.	Hash	DLC	D-Byte 0	D-Byte 1	D-Byte 2	D-Byte 3	D-Byte 4	D-Byte 5	D-Byte 6	D-Byte 7
2+2 Bit	8 Bit	1 Bit	16 Bit	4 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit
Message Prio	System-befehl	1		6	Absender Geräte-UID (Gleis Format Prozessor)				Sub-CMD	Kanal-		
				High				Low	Überlast	nummer		

Beschreibung:

Ein Teilnehmer meldet hiermit die Überschreitung der im Betrieb zulässigen Werte. In Kanalnummer steht der Verursacher.

Die Kanalnummern sind pro Gerät eindeutig festgelegt. Mittels des Befehls "Statusdaten Konfiguration" kann die Bedeutung des Kanals festgestellt werden. Die Kanalnummer wird bei der Abfrage mitgeteilt.

Beispiel (Hex):

00000300 6 43 53 32 08 0A 01 Kanalnummer 1 meldet Überlast.

Antwort:

Dieses "Kommando" wird nur als Antwort gesendet.

Besonderheiten:

Wird immer von dem Gleis Format Prozessor ausgelöst.

## 2.8 Befehl: System Status

Kennung:

Systembefehl (*0x00*, in CAN-ID: *0x00*)

Sub-CMD:

Status (*0x0B*)

Format:

Prio	Command	Resp.	Hash	DLC	D-Byte 0	D-Byte 1	D-Byte 2	D-Byte 3	D-Byte 4	D-Byte 5	D-Byte 6	D-Byte 7
2+2 Bit	8 Bit	1 Bit	16 Bit	4 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit
Message Prio	System-befehl	0		6	Ziel Geräte-UID				Sub-CMD Status	Kanalnummer		
					High			Low				
Message Prio	System-befehl	1		8	Absender Geräte-UID				Sub-CMD Status	Kanalnummer	Messwert	
					High			Low			High	Low

Beschreibung:

Anfrage zu den aktuellen Verbrauchswerten des Gleis Format Prozessor. Gezielte Anfrage von einem Bediengerät. Nur der Gleis Format Prozessor, dessen UID im Frame ist, meldet seinen Wert. Die Kanalnummer sind pro Gerät eindeutig festgelegt. Mittels des Befehls "Statusdaten Konfiguration" kann Anzahl Kanäle, Bedeutung und Konfiguration des Kanals festgestellt werden.

Auslösen der Anfrage durch DLC = 6 und fehlendem Messwert & Resp. Bit.  
Antwort mit DLC = 8 und Messwert, sowie gesetztem Resp. Bit.  
Geräte UID bestimmt das gefragte Gerät.

Dieser Befehl wird dazu verwendet, den Status eines Teilnehmers darzustellen. Diese Abfragen sollten nur dann stattfinden, wenn die Daten wirklich benötigt und angezeigt werden. Zyklische Abfragen, welche nicht zur grafischen Darstellung benötigt werden, sollten in einem Zeitabstand <10s erfolgen.

Beispiel (Hex):

00000300 6 43 53 32 08 0B 01      Abfrage des Messwertes Kanalnummer 1  
00010300 8 43 53 32 08 0B 01 00 03      Antwort mit Messwert

Besonderheiten:

Ist die Kanalnummer nicht vorhanden, so wird die originale Anfrage mit DLC=6 und fehlendem Messwert bestätigt.  
Abfrage mittels der Broadcastadresse 0x00 00 00 00 ist möglich, liefert aber keine Zuordenbarkeit zum Endgerät. Befehl wird mit ursprünglicher Anfrageadresse bestätigt.

Wird immer von einem Bediengerät ausgelöst.

### 3 Verwaltung

Verwaltungsbefehle dienen zum Steuern der vom Gleisformatprozessor angesprochenen Empfänger. Hier sollten keine Befehle enthalten sein, die das Verhalten des Gleisformatprozessors beeinflussen.

#### 3.1 Befehl: Lok Discovery

Kennung:

Discovery (0x01, in CAN-ID: 0x02)

Format:

Prio	Command	Resp.	Hash	DLC	D-Byte 0	D-Byte 1	D-Byte 2	D-Byte 3	D-Byte 4	D-Byte 5	D-Byte 6	D-Byte 7
2+2 Bit	8 Bit	1 Bit	16 Bit	4 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit
Message Prio	Discovery	0		1	Protokoll-Kennung							
Message Prio	Discovery	0 1		5	MFX-UID / Loc-ID				Range/Protokoll-Kennung			
					High			Low				
Message Prio	Discovery	1		6	MFX-UID / Loc-ID				Range	ASK-Verhältnis		
					High			Low				

Beschreibung:

Suchen von Loks auf dem Gleis. Gleis wird durch Protokoll-Kennung bestimmt.

Anfragen:

Form 1: DLC = 1: Erkennen nach Datenprotokoll, Protokoll - Kennung bestimmt Datenprotokoll, Gleis Format Prozessor macht Lok - Erkennen / MFX - Discovery eigenständig. Die Discovery-Zwischenschritte werden in Form 3 kommuniziert. Pro Erkennungsvorgang wird nur ein Decoder erkannt.

Antwort: Positiv mit DLC = 5 und (bisher) gefundener Adresse  
Negativ: DLC = 0

Form 2: DLC = 5: Einzel - MFX - Discovery: Senden der Anforderung mit Range (=Info Länge der Bits).

Antworten:

Zur Anzeige der MFX Erkennung werden bei einem durch den Gleis Format Prozessor gesteuerten Zyklus auch die Zwischenschritte kommuniziert. Erst ein Range = 32 bestimmt die vollständige Dekoder-UID.

Zum Debug: DLC = 6: Einzel - MFX - Discovery: Antwort mit Range und ASK-Verhältnis.

Anfrage Programmiergleis:		
Protokoll	Range/Protokoll-Kennung	Bemerkung
MFX:	0-32	Entspricht Range. Anfrage zur Erkennung der Lok auf dem Programmiergleis. Wenn die Lok antwortet, wird das Antwortbit gesetzt
MM2:	33	Erkennung der MM2-Lok auf dem Programmiergleis. In der Antwort wird die erkannte MM2-Adresse gemeldet.
Anfrage Hauptgleis:		
MFX:	64-96	Range = Wert mod 64. Anfrage zur Erkennung der Lok auf dem Hauptgleis. Wenn die Lok antwortet, wird das Antwortbit gesetzt.

Beispiel (Hex):

00020300 1 20

00020300 5 FF FF FF FF 00

00020300 1 21

Voller mfx Discoveryzyklus

Range 0 MFX Discovery

MM2 Discovery

**Besonderheiten:**

Wird immer von einem Bediengerät ausgelöst. Das Auslösen eines Discoverys sollte nur vom Master - Bediengerät gemacht werden. Ansonsten sind Vorkehrungen für das gleichzeitige Auslösen von Discoverys zu treffen.

Antworten auf ein Discovery sollen und dürfen von allen Teilnehmern empfangen und auch entsprechend ausgewertet werden.

Mfx Discovery unterscheidet nicht nach Hauptgleis oder Programmiergleis.

Wird eine MFX Lok nach Form 1 gesucht, so wird bei einer Antwort der komplette Zyklus mit 32 Stufen durchgeführt.

ASK Verhältnis ist eine Kenngröße für die Qualität der MFX Rückmeldesignale.

Bei einem kompletten Zyklus wird die letzte Meldung mit und ohne ASK Verhältnis gesendet.

### 3.2 Befehl: MFX Bind

**Kennung:**

Bind (0x02, in CAN-ID: 0x04)

**Format:**

Prio	Command	Resp.	Hash	DLC	D-Byte 0	D-Byte 1	D-Byte 2	D-Byte 3	D-Byte 4	D-Byte 5	D-Byte 6	D-Byte 7
2+2 Bit	8 Bit	1 Bit	16 Bit	4 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit
Message Prio	Bind	0/1		6	MFX-UID				MFX-SID			
					High			Low	High	Low		

**Beschreibung:**

Einem MFX-Decoder mit <MFX-UID> mittels mfx-BIND die <MFX-SID> zuweisen, anmelden einer per Discovery gefundenen mfx-Lok. Der Decoder kann danach mit einer Loc-ID angesprochen werden.

**Beispiel (Hex):**

00040300 6 FF FA 8C 43 00 05

Bind FF FA 8C 43 auf SID 05

00050300 6 FF FA 8C 43 00 05

Antwort

**Antwort:**

Ursprünglicher Befehl mit gesetztem Response Bit

**Besonderheiten:**

Nur gültig für MFX.

Wird immer von einem Bediengerät ausgelöst. Das Auslösen des BIND sollte nur vom Master - Bediengerät gemacht werden. Ansonsten sind Vorkehrungen für das gleichzeitige Auslösen von Discoverys zu treffen.

Antworten auf ein Discovery sollen und dürfen von allen Teilnehmern empfangen und auch entsprechend ausgewertet werden.

### 3.3 Befehl: MFX Verify

**Kennung:**

Verify (0x03, in CAN-ID: 0x06)

**Format:**

Prio	Command	Resp.	Hash	DLC	D-Byte 0	D-Byte 1	D-Byte 2	D-Byte 3	D-Byte 4	D-Byte 5	D-Byte 6	D-Byte 7
2+2 Bit	8 Bit	1 Bit	16 Bit	4 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit
Message Prio	Verify	0/1		6	MFX-UID				MFX-SID			
					High			Low	High	Low		
Message Prio	Verify	1		7	MFX-UID				MFX-SID		ASK-	
					High			Low	High	Low	Verhältnis	

**Beschreibung:**

DLC = 6: Anfrage auf Überprüfung, ob eine Lok unter der Kombination MFX-UID / SID vorhanden ist. Sollte diese Kombination vorhanden sein, so wird eine Bestätigung mit DLC = 7, MFX-UID und MFX-SID und dem ASK-Verhältnis (Qualität der Antwort) gesendet.

**Beispiel (Hex):**

00060300 6 FF FA 8C 43 00 05      Verify FF FA 8C 43 auf SID 05  
 00070300 6 FF FA 8C 43 00 00      Antwort negativ  
 00070300 7 FF FA 8C 43 00 05 D1    Antwort positiv mit ASK-Verhältnis 0xD1

**Antwort:**

DLC = 6: Sollte die Kombination nicht vorhanden sein, so wird in der Antwort mit gesetztem Response Bit die SID auf 0x0000 gesetzt.

DLC = 7: Sollte die Kombination vorhanden sein, so wird als Antwort der ursprünglicher Befehl gesendet mit gesetztem Response Bit und in D-Byte 6 das ASK-Verhältnis.

**Besonderheiten:**

Nur gültig für MFX. Sollte diese Kombination aus MFX-UID und MFX-SID die falsche MFX-UID sein, so wird ein UnBIND im Lokdecoder mit der MFX-SID ausgelöst.

Wird immer von einem Bediengerät ausgelöst. Das Auslösen des VERIFYS sollte nur vom Master - Bediengerät gemacht werden. Ansonsten sind Vorkehrungen für das gleichzeitige Auslösen von Discoverys zu treffen.

Antworten auf ein Discovery sollen und dürfen von allen Teilnehmern empfangen und auch entsprechend ausgewertet werden.

### 3.4 Befehl: Lok Geschwindigkeit

**Kennung:**

Lok Geschwindigkeit (0x04, in CAN-ID: 0x08)

**Format:**

Prio	Command	Resp.	Hash	DLC	D-Byte 0	D-Byte 1	D-Byte 2	D-Byte 3	D-Byte 4	D-Byte 5	D-Byte 6	D-Byte 7
2+2 Bit	8 Bit	1 Bit	16 Bit	4 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit
Message Prio	Lok Geschw.	0		4	Loc-ID							
					High			Low				
Message Prio	Lok Geschw.	0 1		6	Loc-ID			Geschwindigkeit				
					High			Low	High	Low		

**Beschreibung:**

Geschwindigkeiten im gesamten System werden als 10 - Bit Werte behandelt. Dieser Wert ist unabhängig vom real zur Lok (über das Gleis) gesendeten Wert. Der verwendete Wertebereich sollte von 0 bis 1000 gehen, 0 entspricht einer stehenden Lok, 1000 der maximalen Geschwindigkeit einer Lok.

Werte oberhalb 1000 (bis 1023) dürfen vorkommen und sollten keinen Empfänger stören. Die Fahrgeschwindigkeit entspricht hierbei dem Maximum.

Dies ist der Fahrbefehl für Lokomotiven. Bei bekannter Loc-ID kann eine Lokomotive sofort gefahren werden. Durch Auswerten der Fahrbefehle (Antworten) kann eine Anzeige stetig aktualisiert werden.

DLC = 6: Setzen der Fahrstufe

Lok mit LOC-ID wird mit Geschwindigkeit angesteuert. Geschwindigkeit im Bereich von 0 bis 1000 (10 Bit). Für alle Protokolle wird die Geschwindigkeit auf die reale mögliche Fahrstufe umgerechnet. Fahrstufe 0 ist Lok - Haltebefehl mit eingestellter ABV (Nicht Nothalt).

DLC = 4: Abfrage der Fahrstufe

Abfrage der aktuellen Geschwindigkeit bei fehlendem Geschwindigkeitswert (DLC = 4)

Durch Abfrage der aktuellen Geschwindigkeit kann ein Fahrgerät die aktuelle Geschwindigkeit erfragen, und damit die aktuelle Darstellung richtig stellen.



Beispiel (Hex):

00080300 6 00 00 40 01 03 20

Lok Geschwindigkeit mfx Adr 1, V=0x0320=800 von 1024

Antwort:

Setzen einer Geschwindigkeit:  
Ursprünglicher Befehl mit gesetztem Response Bit.

Lesen der Geschwindigkeit:  
Antwort in Form Geschwindigkeit setzen, wenn Lok bekannt. Ansonsten keine Antwort.

Besonderheiten:

Wird immer von einem Bediengerät ausgelöst.

Lok/Funktionsdecoder: Erster Fahrstufen / Funktionsbefehl -> Lok wird mit Adressbereich -> Default-Gleisprotokoll, Default-Richtung vorwärts, alle sonstigen Funktionen = 0 angelegt und sofort gefahren.

GUI und Gleis Format Prozessor skalieren Fahrstufe in etwa auf Promille:

14 Fahrstufen entsprechen  $14 * 72 = 1008$

126 Fahrstufen entsprechen  $126 * 8 = 1008$

### 3.5 Befehl: Lok Richtung

Kennung:

Lok Richtung (0x05, in CAN-ID: 0x0A)

Format:

Prio	Command	Resp.	Hash	DLC	D-Byte 0	D-Byte 1	D-Byte 2	D-Byte 3	D-Byte 4	D-Byte 5	D-Byte 6	D-Byte 7
2+2 Bit	8 Bit	1 Bit	16 Bit	4 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit
Message	Lok	0		4	Loc-ID							
Prio	Richtung				High			Low				
Message	Lok	0/1		5	Loc-ID			Richtung				
Prio	Richtung				High			Low				

Beschreibung:

DLC = 4: Abfrage der Richtung:

Aktuelle Richtung erfragen. Antwort mit 5 Bytes in Form von „Setzen Richtung“.

DLC = 5: Setzen der Richtung:

Ändern der Fahrtrichtung gemäß dem Parameter „Richtung“. Bei einer Änderung der Richtung sorgt Gleis Format Prozessor für Fahrstufe 0, Lok wird mit Decoder „ABV“ gebremst. Wird Richtung nicht geändert, erfolgt keine Änderung der Fahrstufe.

Bedeutung Parameter Richtung:

- 0 = Fahrtrichtung bleibt (auf für ungültige Werte)
- 1 = Fahrtrichtung vorwärts
- 2 = Fahrtrichtung rückwärts
- 3 = Fahrtrichtung umschalten

Antwort:

Form: Setzen der Richtung

Ursprünglicher Befehl mit gesetztem Response Bit.

Form: Abfrage der Richtung:

Antwort in Form "Richtung setzen". Eine Antwort erfolgt auf jeden Fall, auch wenn Lok nicht bekannt ist.

Besonderheiten:

Wird immer von einem Bediengerät ausgelöst.

Bei unbekannter Lok: Lok wird mit Adress-Bereich -> Default-Gleisprotokoll, alle sonstigen Fkt=0 angelegt und nicht gefahren (Fahrstufe =0)!

### 3.6 Befehl: Lok Funktion

Kennung:

Lok Funktion (0x06, in CAN-ID: 0x0C)

Format:

Prio	Command	Resp.	Hash	DLC	D-Byte 0	D-Byte 1	D-Byte 2	D-Byte 3	D-Byte 4	D-Byte 5	D-Byte 6	D-Byte 7
2+2 Bit	8 Bit	1 Bit	16 Bit	4 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit
Message Prio	Lok Funktion	0/1		6	Loc-ID			Funktion	Wert			
					High			Low				

Beschreibung:

Lok Funktion auslösen / ausschalten oder abfragen.

Funktion ist im Bereich 0-31. 0 entspricht F0, 31 entspricht F31.

Wert im Bereich von 0 bis 31, 0 = aus, 1 – 31 an. Bei Protokollen welche einen Funktionswert unterstützen wird dieser an Decoder gesendet. Funktionswerte werden nicht im Gleis Format Prozessor gespeichert, nur der Status.

DLC = 5: Abfrage Zustand der Funktion.

Abfrage des Zustandes einer Funktion. Geliefert wird nur Aktiv oder Inaktiv, nicht ein evtl. Funktionswert. Status kommt aus dem Gleis Format Prozessor, nicht aus Abfrage des Dekoders.

Besonderheiten:

Wird immer von einem Bediengerät ausgelöst.

Bei ersterem Fkt - Befehl: Lok wird mit Default-Gleisprotokoll, Default-Richtung vorwärts, alle sonstigen Fkt=0 angelegt und nicht gefahren!

### 3.7 Befehl: Read Config

Kennung:

Read Config (0x07, in CAN-ID: 0x0E)

Format:

Prio	Command	Resp.	Hash	DLC	D-Byte 0	D-Byte 1	D-Byte 2	D-Byte 3	D-Byte 4	D-Byte 5	D-Byte 6	D-Byte 7
2+2 Bit	8 Bit	1 Bit	16 Bit	4 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit
Message Prio	Read Config	0		7	Loc-ID			CV-Index (6 Bit)	CV-Nummer (10 Bit)	Anzahl		
					High			Low				
Message Prio	Read Config	1		7	Loc-ID			CV-Index (6 Bit)	CV-Nummer (10 Bit)	Wert		
					High			Low				
Message Prio	Read Config	1		6	Loc-ID			CV-Index (6 Bit)	CV-Nummer (10 Bit)			
					High			Low				

Beschreibung:

Lesen von Werten aus (Lok)Decodern.

Bei einer Anfrage wird die CV-Nummer und der Startindex angegeben. Durch die Anzahl der zu lesenden Bytes wird bestimmt, wie viele gelesen werden sollen. Eine Anfrage kann, bedingt durch die Anzahl der zu lesenden Bytes, mehrere Antworten auslösen.

Durch Loc-ID werden Protokoll und Adresse des Decoders bestimmt.

In CV-Nummer steht die CV-Nummer, welche gelesen werden soll. Möglich sind insgesamt 1024 Adressen. CV-Nummer steht in D-Byte 5 und den 2 niedrigstwertigen Bits von D-Byte 4.

CV-Index bestimmt den Index der zu lesenden CV-Nummer. CV-Index ist nur für Mfx zulässig. CV-Index steht in den 6 höchstwertigen Bits von D-Byte 4.

Bei der Antwort wird Byteweise ein Wert gesendet. CV-Nummer und CV-Index bestimmen dabei, um welches Bytes es sich handelt. D-Byte 6 beinhaltet nun den gelesenen Wert.

Protokolltypische Wertebereiche und Verhalten:

MFX:

CV-Nummer liegt im Bereich zwischen 0 und 1023.

CV-Index wird beachtet. Index liegt im Bereich zwischen 0 und 63.

Parameter "Anzahl" bestimmt die Anzahl der zu lesenden Bytes ab dem angegebenen Start - CV-Index.

Parameter CV-Nummer bleibt konstant. Es können maximal 63 Byte pro Befehl ausgelesen werden.

Als Antwort wird das gelesene Byte angehängt. CV-Index wird in der Antwort aktualisiert. Konnte kein Wert ausgelesen werden, so wird ein Frame mit DLC=6 und fehlendem D-Byte 6 ausgegeben.

Decoder kann auf Hauptgleis und Programmiergleis gelesen werden. Eine Unterscheidung findet nicht statt.

MM2:

MM2 Loks können nicht ausgelesen werden.

Besonderheiten:

Wird immer von einem Bediengerät ausgelöst.

Je nach Protokoll nur auf dem Programmiergleis möglich.

### 3.8 Befehl: Write Config

Kennung:

Write Config (0x08, in CAN-ID: 0x10)

Format:

Prio	Command	Resp.	Hash	DLC	D-Byte 0	D-Byte 1	D-Byte 2	D-Byte 3	D-Byte 4	D-Byte 5	D-Byte 6	D-Byte 7
2+2 Bit	8 Bit	1 Bit	16 Bit	4 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit
Message Prio	Write Config	0		8	Loc-ID				CvIndex (6 Bit)		Wert	Ctrl(2Bit)
					High			Low	CvNr (10Bit)			
Message Prio	Write Config	1		8	Loc-ID				CvIndex (6 Bit)		Wert	Rslt(2Bit)
					High			Low	CvNr (10Bit)			

Beschreibung:

Schreiben von CV-Werten in einen Lok - Decoder.

Abhängig vom Protokoll des zu programmierenden Decoders sind unterschiedliche Parameter möglich. Das Schreiben der CV-Werte wird, sofern möglich, durch ein nachträgliches Lesen verifiziert. Im Bestätigungsframe wird das Ergebnis der Verifikation mitgeteilt. (mit Verifikation ab Version V2.0)

In Loc-ID stehen Protokoll und Adresse des zu programmierenden Decoders.

In CV-Nr steht die CV-Nummer, welche verändert werden soll. Möglich sind insgesamt 1024 Adressen. CV-Nr steht in D-Byte 5 und den 2 niedrig wertigen Bits von D-Byte 4.

CV-Index bestimmt den Index der zu verändernden CV-Nummer. CV-Index ist nur für Mfx zulässig. CV-Index steht in den 6 höchst wertigen Bits von D-Byte 4.

Parameter Wert enthält das zu schreibende Byte. Bitmanipulationen sind nicht vorgesehen.

Die Bedeutung des letzten Bytes im Telegramm ist für Anforderung und Bestätigung unterschiedlich:  
Anforderung:

Ctrl (Bit 8 & Bit 7) enthält Anweisungen zum Befehl.

Bit 8: Unterscheidung Hauptgleis (HGL, Wert = 1) / Programmiergleis (PGL, Wert = 0)

**Bestätigung:**

In Result (Bit 8 & Bit 7) stehen die Ergebnisse zu Schreiben und Verifizieren:

Bit 8: Schreiben erfolgreich betätigt durch Controller

Bit 7: Verify erfolgreich verlaufen

Je nach Controllertyp können negative Ergebnisse trotz allem erfolgreich verlaufen sein. Z.B. wenn diese weder Auslesen noch Bestätigen beherrschen.

**Protokolltypische Wertebereiche und Verhalten:****MFx:**

CV-Nummer liegt im Bereich zwischen 0 und 1023.

CV-Index wird beachtet. Index liegt im Bereich zwischen 0 und 63.

Das Schreiben findet grundsätzlich auf HGL und PGL statt.

**MM2:**

CV-Index wird nicht behandelt. CV-Nummer liegt im Bereich 1 - 256.

Die Programmierung kann in V1.0 nicht auf dem HGL stattfinden.

Durch die in der Loc-ID angegebene MM2 Adresse lässt sich die "MM2 Programmieradresse" festlegen.

Ein programmierbarer MM2 - Decoder kann entweder unter seiner eigenen Adresse oder auf Adresse 80 programmiert werden. Dies ist auf dem Hauptgleis zu beachten.

**Besonderheiten:**

Wird immer von einem Bediengerät ausgelöst.

## 4 Zubehör-Befehle

Befehle zur Behandlung von Zubehördecodern.

Die Stellung des Zubehörs wird im Gleis Format Prozessor nicht gespeichert.

### 4.1 Befehl: Zubehör Schalten

Kennung:

Zubehör schalten (**0x0B**, in CAN-ID: **0x16**)

Format:

Prio	Command	Resp.	Hash	DLC	D-Byte 0	D-Byte 1	D-Byte 2	D-Byte 3	D-Byte 4	D-Byte 5	D-Byte 6	D-Byte 7
2+2 Bit	8 Bit	1 Bit	16 Bit	4 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit
Message Prio	Zubehör Schalten	0 1		6	Loc-ID				Stellung	Strom		
					High			Low				
Message Prio	Zubehör Schalten	0 1		8	Loc-ID				Stellung	Strom	Schaltzeit / Sonderfunktionswert	
					High			Low				

Beschreibung:

DLC = 6: Schalten mit Default Timeout / Systemtimeout des Gleis Format Prozessor (Siehe Schaltzeit Zubehördecoder)

- STELLUNG 0 - 255, je nach Protokoll unterschiedliche Werte möglich.

- STROM 0 - 31 (Dimmfunktion bei unterstützendem Protokoll)

0: Ausschalten (Ohne Zeitfaktor)

1 - 31: Einschalten mit Dimmwert (sofern Protokoll dies unterstützt), 1 = ein.

- Schaltzeit: Gleis Format Prozessor gesteuerte Schaltzeit in 10ms Schritten.

Bei Schaltzeit = 0: Gleis Format Prozessor schaltet nur ein.

Gleis Format Prozessor kann Schaltzeit nur im Telegrammrahmenraster realisieren. Schaltzeitangaben sind nur rudimentär einzuhalten.

Der Gleis Format Prozessor schaltet bei STROM = 0 ab ODER nach Default-Zeit. (Default ist durch Systembefehl "Schaltzeit Zubehördecoder festlegen" bestimmbar).

Protokollspezifische Stellungskodierung MM3:

Bit 0,1: Stellung:

00: Aus, Rund, Rot, Rechts, HP0

01: Ein, Grün, Gerade, HP1

10: Gelb, Links, HP2

11: Weiß, SH0

Protokolltypische Wertebereiche und Verhalten:

MM2:

Bediengeräte müssen darauf achten, dass im Gesamtsystem nur ein Artikel geschaltet werden darf. Erst wenn dieser Artikel wieder ausgeschaltet wurde, darf der nächste Einschaltbefehl folgen.

Bei MM2 werden die Artikel auf der Grundadresse des Decoders ausgeschaltet. D.H. mit einem Ausschaltbefehl werden alle 8 Ausgänge des Decoders ausgeschaltet. Durch dieses Verhalten können unerwünschte Schaltfolgen entstehen. Weiterhin wird bei MM2 vorausgesetzt, dass das Steuergerät einen Ausgang zurücksetzt und nicht der Decoder selbst. Dadurch entsteht die Gefahr, dass ein Artikel durchbrennen kann.

Besonderheiten:

Wird immer von einem Bediengerät ausgelöst.

## 5 Software Befehle

### 5.1 Befehl: Teilnehmer Ping

Kennung:

Softwarestand / Ping (*0x18*, in CAN-ID: *0x30*)

Format:

Prio	Command	Resp.	Hash	DLC	D-Byte 0	D-Byte 1	D-Byte 2	D-Byte 3	D-Byte 4	D-Byte 5	D-Byte 6	D-Byte 7
2+2 Bit	8 Bit	1 Bit	16 Bit	4 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit
Message Prio	SW-Stand Anfrage	0		0								
Message Prio	SW-Stand Anfrage	1		8	Absender Geräte UID			VERS		DB		
					High			Low				

Beschreibung:

Anfrage mit Len = 0

Jedes Gerät antwortet mit seinen entsprechenden Daten.

Damit wird die Konfigurationsabfrage aller am CAN Bus erreichbarer Teilnehmer erreicht. Bei der Antwort wird die UID des antwortenden Gerätes mitgeteilt. Somit kann das Bediengerät bestimmen, welche Geräte angeschlossen sind.

Besonderheiten:

Es antworten nur die Steuergeräte. Keine Lokdecoder.

### 5.2 Befehl: Statusdaten Konfiguration

Kennung:

Statusdaten (*0x1D*, in CAN-ID: *0x3A*)

Format:

Prio	Command	Resp.	Hash	DLC	D-Byte 0	D-Byte 1	D-Byte 2	D-Byte 3	D-Byte 4	D-Byte 5	D-Byte 6	D-Byte 7
2+2 Bit	8 Bit	1 Bit	16 Bit	4 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit
Message Prio	Statusdaten	0		5	Ziel Geräte-UID			Index				
					High			Low				
Message Stream Prio	Statusdaten Stream	1	Paket#	8	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Message Prio	Statusdaten	1		5	Geräte-UID			Index		Paketanzahl		
					High			Low				

Beschreibung:

Abfrage der Konfigurationswerte der Beschreibung der Messwertdaten eines Gerätes.

Messwertdaten eines Gerätes sind in Indices gegliedert. Zu jedem Index gehört eine Beschreibung. In der Antwort sind für jeden Index die im Gerät gespeicherten Konfigurationsdaten enthalten. Index 0 ist die Gerätebeschreibung und enthält unter anderem die Anzahl der zur Verfügung gestellten Messwerte. Jeder weitere Index liefert eine Beschreibung, wie der entsprechende Messkanal dargestellt werden kann. Die Indice für die Messwerte müssen bei 1 beginnen und dürfen hier keine Lücken enthalten.

Die Übertragung der Daten erfolgt als Datenstrom, im Hash befindet sich die Paketnummer. Der Abschluss der Übertragung erfolgt durch Bestätigung des ursprünglichen Datagramms und der Information der gesendeten Pakete.

#### Format Gerätebeschreibung

Unter Index 0 sind die Gerätebeschreibung abrufbar. Primär ist dies die Anzahl der zur Verfügung gestellten Messkanäle. Weiterhin enthalten sind Angaben zur Identifikation des Gerätes.

Format Gerätebeschreibung:

- 1 Byte Anzahl der Messwerte im Gerät.
- 3 Byte frei.
- 4 Byte Seriennummer Gerät.
- 8 Byte Artikelnummer.
- 1 String Gerätebezeichnung 0 - Terminiert oder Längenbegrenzt auf 24 Zeichen.

Format des Datenblocks:

Unter den darauf folgenden Indices sind die Messwertbeschreibungen abrufbar. Jeder Index liefert eine zum Messwert gehörende Beschreibung:

Format Messwertbeschreibung

Typ	Bedeutung	Beispiel
u8	Abfragekanalnummer	0x01: Abfrage unter Kanal 1
u8	Potenz des Messwerts	-3 Bedeutet: 10 <sup>-3</sup>
u8	Farbe Bereich 1	0x00: Farbdarst. Sw, SW-Darst: Sw
u8	Farbe Bereich 2	0x31: Farbdarst. Grün, SW-Darst: Gr1
u8	Farbe Bereich 3	0xF2: Farbdarst. Gelb, SW-Darst: Gr2
u8	Farbe Bereich 4	0xC3: Farbdarst. Rot SW-Darst: Ws
U16	Messwert Start	0
U16	Messwert Ende Bereich 1	300
U16	Messwert Ende Bereich 2	325
U16	Messwert Ende Bereich 3	375
U16	Messwert Ende Bereich 4	400
String	Messwertbezeichnung	"Hauptgleisstrom\0"
String	Bezeichnung und Wert Start	"0.00\0"
String	Bezeichnung und Wert Ende	"3.20\0"
String	Einheit	"Achsen\0" oder "A\0"

Abfragekanalnummer:

Wert bestimmt, unter welcher Kanalnummer mit dem Befehl "System Status" der entsprechende Messwert abgerufen werden kann. Messwerte sind als vorzeichenlose 16 Bit Messwerte normiert. Ebenso wird bei einer Überlastmeldung diese Kanalnummer verwendet.

Potenz des Messwerts:

Bestimmt, welche Potenz der zugehörige Messwert hat, also Messwert 1000 entspricht 1 bei Potenz -3

Farben der Bereiche eines Kanals:

Für eine farbliche Darstellung eines Messkanals benötigt man Bereichsgrenzen und die Informationen für die Farbdarstellung des Bereichs.

Beispiel Strommessung

Grün ist der normale Lastbereich, Gelb ist grenzwertig und bei Rot sollte nicht zu lange Betrieb sein. Max entspricht hier dem Überlastbereich, wenn dieser zu lange anhält wird abgeschaltet.

Bereich 1	Bereich 2	Bereich 3	Bereich 4
Grün	Gelb	Orange	Rot
Start aus String: 0			Ende aus String 2.500
Zugehöriger Messwert: 0	300	325	375      400

Der Messwert muss entsprechend umgerechnet werden!

Beispiel Spannungsmessung

Unterspannung und Überspannung sind Rot. Gelb ist ein Bereich, in dem Betrieb möglich aber nicht Empfehlenswert, Grün ist der normale Betriebsbereich.

Bereich 1	Bereich 2	Bereich 3	Bereich 4
Rot	Gelb	Grün	Rot

Die Farbdarstellung wird als 8 Bitwert, im Format RGB und Graudarstellung, repräsentiert. Dabei werden je 2 Bit für jede Information verwendet.  
Die Werte für Nullpunkt und Bereiche 1-4 bestimmen den Messwert ab welchem diese Farbe gültig ist. Anfang bestimmt, ab welchem Messwert der Wert angezeigt werden soll.

Messwertbezeichnung:

Liefert eine zum Messwert gehörende Bezeichnung.

Internationalisierung:

Messwerte müssen in einem Bediengerät internationalisiert werden können. Damit dies realisiert werden kann, sind folgende vordefinierte Begriffe festgelegt:

Messwertbezeichnung	Bezeichnung Deutsch	Bez. Englisch
HGL	Hauptgleis	Maintrack
PGL	Programmiergeis	Programtrack
TRACK	Boostergleis	Boostertrack
VOLT	Versorgungsspannung	Supplyvoltage
ACHSEN	Achsen	Axles
TEMP	Temperatur	Temperature
TEMPO	Geschwindigkeit	Speed

Ist ein Begriff in dieser Tabelle nicht enthalten, so wird er wie empfangen dargestellt und nicht internationalisiert.

Bezeichnung Start:

Format einer Gleitkommazahl. Liefert sowohl die Information, für den Startwert der Darstellung als auch eine Information, wie der Messwert dargestellt werden soll. Die Anzahl der Nachkommastellen sind auch die Anzahl der Nachkommastellen des Messwerts (mA also 3 Nachkommastellen). Abgefragter Messwert muss bei der Darstellung durch 1000 geteilt werden. Die Anzahl der Nachkommastellen und die Potenz des Messwerts müssen identisch sein.

Bezeichnung Ende:

Format und Funktion wie Bezeichnung Start. Für das Ende der Darstellung.

Einheit:

Liefert für die Darstellung des Messwertes die entsprechende Einheit.

**Besonderheiten:**

- Wird immer von einem Bediengerät ausgelöst.
- Kanalnummer und Kanalindex müssen nicht übereinstimmen
- Kanalnummer wird als Kennung bei Überlastmeldung verwendet



## 6 Abkürzungen

| oder

& und

# Nummer

8051F044 = Gleis Format Prozessor-Micro-Prozessor von SiLabs

ABV = Anfahrt und Bremsverzögerung.

Adr = Adresse

Anz = Anzahl

ASK = Amplitude Shift Keying, 1-bit-Rückmeldung bei mfx

CAN = Bussystem

Cmd, CMD = Command

CS1 = Central Station 1

CS2 = Central Station 2

CV = Configuration-Variables.

D-Byte = Daten-Byte

F0, F1, F2, ... Funktion Nr...

Fkt = Funktion

FS = Fahrstufe.

Geräte-UID = Eindeutige Identifier aller am CAN Bus angeschlossenen Geräte.

Gleis Format Prozessor = Gleisformat Prozessor.

High = höherwertige bits/bytes

HGL = Hauptgleis

ID = Identifikation/Identifyer

K83 = Zubehörartikeldecoder

Len = Length, Länge

Loc-ID = Local Identifier. Adresse, unter der Lok-/ Zubehördecoder auf dem CAN angesprochen werden.

Low = niederwertige bits/bytes

LSB = Least Significant/ niederwertige bits/bytes

mfx = Gleis-Protokoll mit automatische Anmeldung und vielen Funktionen

MFX-SID = Schienen-ID = mfx-Adresse, unter der gefahren wird.

MFX-UID = Universal-Identifier des mfx-Decoders.

MM1 = alte einfache Variante von MM2, nur Licht-Funktion F0

MM2 = Märklin Motorola 2, Gleis-Protokoll, F0-F4

MM3 = MM2 mit 255 statt nur 80 Adressen und mehr Schaltzuständen

ms = Milli-Sekunden

MSB = Most Significant/ höherwertige bits/bytes

PGL = Programmiergleis

POM = Programming On the Main, Hauptgleis-Programmierung, während andere Loks auf Gleis.

Prio = Priorität

PSK = Phase Shift Keying, 1-8 byte Rückmeldung bei mfx

Resp. = Response-Bit

S88 = Rückmelde-Hardware/Protokoll

SID = Schienen-Identifyer, den mfx vergibt

SNr. = Seriennummer

Sub-CMD = Unter-Kommando

UID = Universal Identifyer

URC = Under Rail Control (für MM2 Lichtsignale).

V = Velocity, Geschwindigkeit

Zub = Zubehör-Artikel, z.B. Weiche, Signal