

Kommunikationsprotokoll
Graphical User Interface Prozessor (GUI)
<->
Gleisformat Prozessor (Gleis Format Prozessor)
über CAN
Transportierbar über Ethernet



märklin
digital

(Version vom 7.2.2012)

Gebr. Märklin & Cie. GmbH, Göppingen

Copyright (C) 2012 Gebr. Märklin & Cie. GmbH

Inhaltsverzeichnis:

| | | |
|---------|--|----|
| 1 | CAN Anschluss | 5 |
| 1.1 | CAN Grundformat | 5 |
| 1.2 | Feldergrundbeschreibung | 5 |
| 1.2.1 | Priorität (Prio) | 5 |
| 1.2.2 | Kommando | 5 |
| 1.2.3 | Response | 6 |
| 1.2.4 | Hash | 6 |
| 1.2.5 | Meldungsbestätigung | 6 |
| 1.2.6 | Sonstiges | 6 |
| 1.2.7 | Übertragung der CAN Kommandos via Ethernet | 7 |
| 1.3 | Allgemeines | 8 |
| 1.3.1 | Adressen im System – Die UID | 8 |
| 1.3.1.1 | Definition der Teilnehmerkennungen | 8 |
| 1.3.1.2 | Einbindung bestehender Gleisprotokolle, Bildung der „Loc-ID“ | 8 |
| 1.3.2 | Bildung von Automatik-UID und Rückmelde-UID | 9 |
| 1.3.2.1 | Gerätekennungen | 9 |
| 1.3.2.2 | Rückmeldekennungen und Automatikkennungen | 9 |
| 1.3.2.3 | Automatisierungsadressen / Rückmeldeadressen | 10 |
| 1.3.3 | Fahrstufen im System | 10 |
| 1.4 | Übersicht Werte der Kommandos | 11 |
| 2 | System-Befehle | 12 |
| 2.1 | Befehl: System Stopp | 12 |
| 2.2 | Befehl: System Go | 13 |
| 2.3 | Befehl: System Halt | 14 |
| 2.4 | Befehl: Lok Nothalt | 15 |
| 2.5 | Befehl: Lok Zyklus beenden | 16 |
| 2.6 | Befehl: Lok Datenprotokoll | 17 |
| 2.7 | Befehl: Schaltzeit Zubehördecoder festlegen | 18 |
| 2.8 | Befehl: Fast Read für mfx SID - Adresse | 19 |
| 2.9 | Befehl: Gleisprotokoll Frei Schalten | 20 |
| 2.10 | Befehl: System MFX Neuanmeldezähler setzen | 21 |
| 2.11 | Befehl: System Überlast | 22 |
| 2.12 | Befehl: System Status | 23 |
| 2.13 | Befehl: Gerätekennung | 24 |
| 2.14 | Befehl: System Reset | 25 |
| 3 | Verwaltung | 26 |
| 3.1 | Befehl: Lok Discovery | 26 |
| 3.2 | Befehl: MFX Bind | 28 |
| 3.3 | Befehl: MFX Verify | 29 |
| 3.4 | Befehl: Lok Geschwindigkeit | 30 |
| 3.5 | Befehl: Lok Richtung | 31 |
| 3.6 | Befehl: Lok Funktion | 32 |
| 3.7 | Befehl: Read Config | 33 |
| 3.8 | Befehl: Write Config | 35 |
| 4 | Zubehör-Befehle | 37 |
| 4.1 | Befehl: Zubehör Schalten | 37 |
| 5 | Rückmeldungen | 38 |
| 5.1 | Befehl: S88 Polling | 39 |
| 5.2 | Befehl: Rückmelde Event | 40 |
| 6 | Sonstige Befehle | 41 |
| 6.1 | Befehl: Softwarestand Anfrage / Teilnehmer Ping | 41 |
| 6.2 | Befehl: Statusdaten Konfiguration | 42 |

| | | |
|---------|--|----|
| 7 | GUI Informationsübertragung | 46 |
| 7.1 | Befehl: Anfordern Config Data | 46 |
| 7.2 | Befehl: Config Data Stream | 48 |
| 8 | Format der Konfigurationsdateien der CS2 | 50 |
| 8.1 | Konfigurationsdatei "Lokomotive.cs2" | 50 |
| 8.1.1 | Sektion "version" | 50 |
| 8.1.2 | Sektion "session" | 50 |
| 8.1.3 | Sektion "lokomotive" | 51 |
| 8.1.3.1 | 2. Ebene Sektion ".funktionen" | 51 |
| 8.1.3.2 | 2. Ebene Sektion ".prg" | 51 |
| 8.2 | Konfigurationsdatei "magnetartikel.cs2" | 53 |
| 8.2.1 | Sektion "version" | 53 |
| 8.2.2 | Sektion "artikel" | 53 |
| | Konfigurationsdatei "fahrstrasse.cs2" | 55 |
| 8.2.3 | Sektion "version" | 55 |
| 8.2.4 | Sektion "fahrstrasse" | 55 |
| 8.2.4.1 | 2. Ebene Sektion ".item" | 55 |
| 8.3 | Konfigurationsdatei "gleisbild.cs2" | 57 |
| 8.3.1 | Sektion "version" | 57 |
| 8.3.2 | Sektion "groesse" | 57 |
| 8.3.3 | Sektion "zuletztBenutzt" | 57 |
| 8.3.4 | Sektion "seite" | 57 |
| 8.3.5 | Sektion "element" | 58 |
| 8.4 | Konfigurationsstream "lokinfo" | 60 |
| 8.5 | Konfigurationsstream "loknamen" | 62 |
| 8.6 | Konfigurationsstream "maginfo" | 63 |
| 8.7 | Konfigurationsstream "lokdb" | 64 |
| 8.8 | Konfigurationsstream "ldbver" | 65 |
| 9 | Automatisierung | 66 |
| 9.1 | Befehl: Automatik schalten | 66 |
| 10 | Abkürzungen | 68 |

Haftungsausschluss:

Die in diesem Dokument enthaltenen Informationen beschreibt die Kommunikationssystematik der Komponenten von Märklin Digital.

Diese Dokumentation wird "wie sie ist" und ohne jede Gewährleistung für Funktion, Korrektheit oder Fehlerfreiheit zur Verfügung gestellt. Für jedweden direkten oder indirekten Schaden - insbesondere Schaden an anderer Software, Schaden an Hardware, Schaden durch Nutzungsausfall und Schaden durch Funktionsuntüchtigkeit der damit verbundenen Produkte, kann der Autor nicht haftbar gemacht werden.

Diese Beschreibung wurde mit größter Sorgfalt entwickelt, jedoch können Fehler niemals ausgeschlossen werden. Es kann daher keine Gewähr für die Richtigkeit übernommen werden.

Diese Beschreibung stellt keine Zusicherung von Eigenschaften des Märklin Digital Systems oder den damit verbundenen Produkten dar.

Technische Änderungen sind vorbehalten.

Urheberrecht

Die folgenden Seiten unterliegen dem deutschen Urheberrecht. Die Vervielfältigung, Bearbeitung, Verbreitung und jede Art der Verwertung außerhalb der Grenzen des Urheberrechtes bedürfen der schriftlichen Zustimmung der Fa. Gebr. Märklin & Cie GmbH.

1 CAN Anschluss

Der CAN Bus dient als Kommunikationsnetz der Steuergeräte bei Märklin Systems. 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.

Die Datenrate ist 250 KBit/s, die maximale Buslänge ist 100m.

1.1 CAN Grundformat

Das CAN Protokoll schreibt vor, dass Meldungen mit einer 29 Bit Meldungskennung, 4 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 | | | | | | | |

1.2 Feldergrundbeschreibung

1.2.1 Priorität (Prio)

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.

1.2.2 Kommando

Bestimmt das vom Endgerät auszuführende, bzw. das ausgeführte Kommando.

Kommandowerte sind eindeutig definiert.

Kennwertbereiche für die Kommandowerte:

| Bereich | Anzahl | Start | Ende |
|-----------------------------|--------|-------|------|
| Systembefehle | 1 | 0x00 | 0x00 |
| Verwaltung | 8 | 0x01 | 0x0A |
| Zubehör | 2 | 0x0B | 0x0D |
| Rückmeldungen | - | 0x10 | 0x12 |
| Software Update / Konfig | 6 | 0x18 | 0x1C |
| GUI Informationsübertragung | 3 | 0x20 | 0x22 |
| Automatisierungsbefehle | | 0x30 | |

1.2.3 Response

Bestimmt, ob CAN Meldung eine Anforderung oder Antwort oder 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. Jeder Teilnehmer am Bus, welche die Meldung ausgeführt hat, bestätigt ein Kommando.

1.2.4 Hash

Der Hash erfüllt eine Doppelfunktion:

Primär dient er zur Kollisionsauflösung der Meldungen und zur Sicherstellung der Kollisionsfreiheit zum CS1 Protokoll.

Sekundär kann er die Folgenummer einer Datenübertragung beinhalten.

Kollisionsfreiheit zum CS1 Protokoll:

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 (0bxxxxxx11), nicht benutzt. Diese Bitkombination wird daher zur Unterscheidung fest im Hash verwendet.

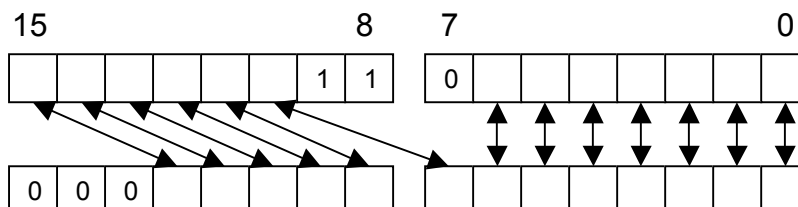
Kollisionsauflösung:

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

Jeder Teilnehmer am Bus hat den Hash empfangener CAN-Meldungen auf Kollisionsfreiheit zu prüfen. Wird der eigene Hash empfangen, so ist ein neuer zu wählen. Dieser darf mit keinem weiteren empfangenen übereinstimmen.

Folgenummer einer Datenübertragung:

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.



1.2.5 Meldungsbestätigung

Eine Initiator eine Meldung muss Sorge dafür tragen, dass die gewünschte Aktion tatsächlich ausgeführt wird. Die Meldungen werden nicht gesichert über den CAN-Bus übertragen. Der Empfang einer Meldung wird nicht bestätigt. Die Ausführung eines Kommandos wird bestätigt, bzw nur durch das Senden der Bestätigungsmeldung quittiert. Fehlt diese Quittierung, ist davon auszugehen, dass die Aktion nicht ausgeführt wurde.

1.2.6 Sonstiges

- In der Kommunikation werden keine Sender + Empfänger Adresse verwendet.
- 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.
- Byte-Order in den Meldungen ist immer Motorola Big Endian.

1.2.7 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 abzubildende 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, da das CAN - Ethernet - Gateway Pakete anderer Länge verwirft.

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.

1.3 Allgemeines

1.3.1 Adressen im System – Die UID

Der gesamte Adressraum hat 2^{32} Adressen (0x0000 0000 - 0xFFFF FFFF), dieses sind rund 4 Milliarden Adressen.

Diese werden auch als UID (Universal Identifier) bezeichnet. Je nach eingesetztem Protokoll hat jedoch eine UID eine andere Bedeutung.

1.3.1.1 Definition der Teilnehmerkennungen

Im System besitzt jeder adressierbare Teilnehmer eine eindeutige 32 Bit Adresse.

Dabei werden folgende UID unterschieden:

Geräte-UID Eindeutig vergebene Universal ID.
 Loc-ID (=Local ID, nicht Locomotive ID) Aus dem Protokoll und der Adresse berechnete Lokale ID.
 MFX-UID MFX Universal ID, eindeutige Kennung eines mfx Teilnehmers.

Bestimmte Geräte-UID besitzen eine besondere Bedeutung:

Die UID 0x00000000 ist die Broadcastadresse. Signalisiert, dass mehrere Teilnehmer denselben Befehl abarbeiten sollen.

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

1.3.1.2 Einbindung bestehender Gleisprotokolle, Bildung der „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 | SX1 |
| 0x0C00 | 0x0FFF | Reserviert |
| 0x1000 | 0x13FF | Res. für MM1,2 Funktionsdecoder F1 - F4 (40 kHz, 80 & 255 Adressen) |
| 0x1400 | 0x17FF | Reserviert |
| 0x1800 | 0x1BFF | Frei für Privatanwender / Clubs |
| 0x1C00 | 0x1FFF | Frei für Firmen |
| 0x2000 | 0x23FF | Reserviert für MM1,2 Lokdecoder (20 kHz, 80 & 256 Adressen) |
| 0x2400 | 0x27FF | Reserviert |
| 0x2800 | 0x2BFF | SX1 - Zubehörartikel (Erweiterung) |
| 0x2C00 | 0x2FFF | Reserviert für Traktionen (interne GUI Kennungen) |
| 0x3000 | 0x33FF | MM1,2 Zubehörartikeldecoder (40 kHz, 320 & 1024 Adressen) |
| 0x3400 | 0x37FF | Reserviert |
| 0x3800 | 0x3BFF | DCC-Zubehörartikel |
| 0x3C00 | 0x3FFF | DCC-Zubehörartikel |
| | | |
| 0x4000 | 0x7FFF | MFX |
| 0x8000 | 0xBFFF | SX2 |
| 0xC000 | 0xFFFF | DCC |

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

1.3.2 Bildung von Automatik-UID und Rückmelde-UID

Im Gesamtsystem können sich mehrere Geräte mit der Fähigkeit einer Automatisierung oder einer Rückmeldemöglichkeit befinden.

Für eine Möglichkeit, diese Ressourcen Systemweit nutzen zu können, muss diese Fähigkeit angesprochen werden können. Dabei wird die Möglichkeit der Rückmeldung und der Automatisierung in einem gemeinsamen Adressraum zusammengefasst.

Diese Kennungen bestehen aus zwei 16Bit Teilkennungen: Einer zugeordneten Gerätekennung und einer Kennung für die Automatik/Rückmeldekennung in diesem Gerät. Somit können 65K Geräte mit jeweils 65K Funktionen zusammengeschaltet werden. Die Kontaktadresse sowie die Automatisierungsadresse wird somit über eine Kombination von 16 Bit Gerätekenner und 16Bit Kontakt- / Automatikkennung gebildet. Über diese Adressierung werden im System alle Kontakte und Automatikfunktionen angesprochen.

Durch diese Kennungen ist es möglich, Ressourcen von einem Gerät zur Steuerung in einem anderen Gerät zu verwenden.

1.3.2.1 Gerätekennungen

Ressourcen in Systeme, die einen S88 Bus / Rückmeldebus besitzen oder eine Automatisierungsfunktionen realisieren, bekommen eine 16Bit Gerätekenner zugewiesen.

Die Master - Zentrale weist den Endgeräten die Kennung beim Systemstart jeweils zu. Eine Resetfeste Speicherung findet in den Geräten nicht statt.

Über diese zentrale Verwaltung der Gerätekennungen wird erreicht, dass eine ausgefallenes Gerät im System ersetzt werden kann. Die gespeicherte und verwendete Adressierung in Automatikfunktionen kann somit unverändert übernommen werden.

Der Master – Zentrale speichert eine Liste mit allen bekannten Geräten im System, sowie deren NickNames als .cs2 Konfigurationsdatei. Der Name/Kenner kann sinnvoll vorbelegt werden, muss aber vom Anwender geändert werden können.

1.3.2.2 Rückmeldekennungen und Automatikkennungen

Über diese Kennung wird sowohl eine Rückmeldung wie auch eine Automatisierungsfunktion angesprochen.
S88 /

Rückmeldekennungen beginnen im Highbyte bei 0 bis maximal 63, Lowbyte im Bereich jeweils von 0 - 255. Somit maximal 16.384 Rückmeldekontakte pro Gerät möglich.

Der Wert 64 ist reserviert für SX1 Rückmelder.

Automatikfunktionen beginnen im Highbyte bei 65 der ASCII Darstellung von „A“. Im Lowbyte beginnen per Konvention zur Zeit die Werte bei ASCII "1", dezimal 49 (0x31). (Dies ist die ASCII Darstellung „A1“ – somit die erste memory Funktion). Die Lücke zwischen SX1 Rückmelder und Automatikfunktionen ist reserviert.

Dabei werden die jetzt schon vorhandenen Automatikfunktionen mit der schon vergebenen Bezeichnung A1 – z8 adressiert. Neue Kennungen können zur Realisation neuer Automatisierungsfunktionen benutzt werden.

Tabelle der sich daraus ergebenden Adressierung.

| Gerätekennung | Start | | Ende | | Verwendung |
|-----------------|------------|-----------|------------|-------------|---|
| 0x0000 - 0xFFFF | 0x00 0 | 0x00 0 | 0x3F 63 | 0xFF 255 | S88 Rückmelder 16384 Melder |
| 0x0000 - 0xFFFF | 0x40 64 | 0x00 0 | 0x40 64 | 0xFF 255 | Reserviert SX1 Rückmelder 128 Melder |

| | | | | | |
|--------------------|-------------|------------|-------------|-------------|--|
| 0x0000 - 0xFFFF | 0x41 65 | 0x00 0 | 0x41 65 | 0x30 48 | Frei / Reserve |
| 0x0000 - 0xFFFF | 0x41 65 | 0x31 49 | 0x7F 127 | 0xFF 255 | Memory – Automatikfunktionen. Beginnend bei Ascii Darstellung „A“ „1“ |
| 0x0000 - 0xFFFF | 0x80 128 | 0x00 0 | 0xFF 255 | 0xFF 255 | Reserviert |

1.3.2.3 Automatisierungsadressen / Rückmeldeadressen

Diese Adressen werden zusammengesetzt aus der Geräteerkennung (Höher wertig) und der entsprechenden Rückmeldeerkennung bzw Automatikerkennung. Die Geräteerkennung bildet dabei den höherwertigen Teil.

1.3.3 Fahrstufen im System

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 dürfen keinen Empfänger stören. Die Fahrgeschwindigkeit entspricht hierbei dem Maximum.

Die Umrechnung in reale möglich Fahrstufen ist anhand folgender Rechenvorschriften möglich:

Systemfahrstufe = 1 + (Gleisfahrstufe - 1) * Schrittweite

| Fahrstufen Anzahl | Schrittweite |
|----------------------|--------------|
| 14 | 77 |
| 27 | 38 |
| 28 | 37 |
| 31 | 33 |
| 126 | 8 |

Gleisfahrstufe 1 ist somit immer auch Systemfahrstufe 1.

Gleisfahrstufe 11 ist bei:

- 14 Fahrstufen: 771
- 27 Fahrstufen: 381
- 28 Fahrstufen: 371
- 31 Fahrstufen: 331
- 126 Fahrstufen: 81

1.4 Übersicht Werte der Kommandos

Übersichtstabelle der Kommandokenner

| Befehl | Wert | Wert in CAN-ID | mögliche DLC |
|--|------|----------------|--------------|
| System-Befehle | 0x00 | 0x00 | siehe unten |
| Lok Discovery | 0x01 | 0x02 | 0,1,5,6 |
| MFX Bind | 0x02 | 0x04 | 6 |
| MFX Verify | 0x03 | 0x06 | 6,7 |
| Lok Geschwindigkeit | 0x04 | 0x08 | 4,6 |
| Lok Richtung | 0x05 | 0x0A | 4,5 |
| Lok Funktion | 0x06 | 0x0C | 5,6,8 |
| Read Config | 0x07 | 0x0E | 6,7 |
| Write Config | 0x08 | 0x10 | 8 |
| Zubehör Schalten | 0x0B | 0x16 | 6,8 |
| Zubehör Konfig | 0x0C | 0x18 | |
| S88 Polling (Feedback) | 0x10 | 0x20 | 5,6,7,8 |
| S88 Event | 0x11 | 0x22 | 7,8 |
| SX1 Event | 0x12 | 0x24 | 5,6 |
| Softwarestand Anfrage/Teilnehmer Ping | 0x18 | 0x30 | 0,5,7,8 |
| Updateangebot | 0x19 | 0x32 | 8 |
| Read Config Data | 0x1A | 0x34 | 4,8 |
| Bootloader CAN gebunden, „Service“ | 0x1B | 0x36 | 0,5,6,7,8 |
| Bootloader Schienen gebunden, „Service“ | 0x1C | 0x38 | 4,5,6,7,8 |
| Statusdaten Konfiguration | 0x1D | 0x3A | 5,8 |
| Anfordern Config Data, „Data Query“ | 0x20 | 0x40 | 8 |
| Config Data Stream | 0x21 | 0x42 | 6,7,8 |
| 60128 (Connect 6021) Data Stream/ alte Bezeichnung „6021 adapter“ | 0x22 | 0x44 | 5,6 |

Übersichtstabelle Systembefehle und Wert Subcommand

| Befehl | Sub-CMD Wert | mögliche DLC |
|------------------------------------|--------------|--------------|
| System Stopp | 0x00 | 5 |
| System Go | 0x01 | 5 |
| System Halt | 0x02 | 5 |
| Lok Nothalt | 0x03 | 5 |
| Lok Zyklus Stopp(beenden) | 0x04 | 5 |
| Lok Datenprotokoll | 0x05 | 6 |
| Schaltzeit Zubehördecoder | 0x06 | 5 |
| Fast Read für mfx | 0x07 | 6 |
| Gleisprotokoll frei schalten | 0x08 | 6 |
| System MFX Neuanmeldezähler setzen | 0x09 | 7 |
| System Überlast | 0x0A | 6 |
| System Status | 0x0B | 6,8 |
| System Kennung | 0x0C | 5,7 |
| Mfx Seek | 0x30 | 6,7,8 |
| System Reset | 0x80 | 6 |

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 Stopp | | | |
| | | | | | High | | | Low | | | | |

Beschreibung:

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 muss auf einen generellen Stopp Befehl, welcher alle Gleis Format Prozessoren betrifft, geachtet werden. Hierzu wird eine spezielle Ziel Geräte UID verwendet (0x0000)

Beispiel:

00004711 5 00 00 00 00 00 Stopp an alle

00004711 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 Graphical User Interface Prozessor 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 Go | | | |
| | | | | | High | | | Low | | | | |

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 muss auf einen generellen Go Befehl, welcher alle Gleis Format Prozessor betrifft, geachtet werden. Hierzu wird eine spezielle Ziel Geräte UID verwendet (0x0000) .

Beispiel:

00004711 5 00 00 00 00 01 Go an alle

00004711 5 43 53 32 08 01 Go an bestimmten Teilnehmer (CS2 mit SNr. 04)

Antwort:

Ursprünglicher Befehl mit gesetztem Response Bit.

Besonderheiten:

Go wird immer von Graphical User Interface Prozessor 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 Halt | | | |
| | | | | | High | | | Low | | | | |

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 muss auf einen generellen System Halt Befehl, welcher alle Gleis Format Prozessor betrifft, geachtet werden. Hierzu wird eine spezielle Ziel Geräte UID verwendet (0x0000)

Beispiel:

00004711 5 00 00 00 00 02 Halt an alle

00004711 5 43 53 32 08 02 Halt an bestimmten Teilnehmer (CS2 mit SNr. 04)

Antwort:

Ursprünglicher Befehl mit gesetztem Response Bit.

Besonderheiten:

Halt wird immer von Graphical User Interface Prozessor 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 Lok Halt | | | |
| | | | | | High | | | Low | | | | |

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 nicht aufgenommen.

Beispiel:

```
00004711 5 00 00 00 48 03   Lok Nothalt MM2 72
00004711 5 00 00 C0 03 03   Lok Nothalt DCC  Adr. 3
00004711 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 Graphical User Interface Prozessor ausgelöst.

Erster Befehl nimmt Lok in den Zyklus auf, Nothalt wird gesendet.

2.5 Befehl: Lok Zyklus beenden

Kennung:

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

Sub-CMD:

Lok Zyklus Stopp (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 High | | | Low | Sub-CMD Zyklus | | | |

Beschreibung:

Lok aus Verwaltungsliste des Gleis Format Prozessor löschen.

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:

00004711 5 00 00 00 48 04 Zyklus Ende MM2 72
 00004711 5 00 00 C0 03 04 Zyklus Ende DCC Adr. 3
 00004711 5 00 00 40 05 04 Zyklus Ende MFX SID 5
 00004711 5 43 53 32 08 04 Zyklus Ende alle Loks im Gleis Format Prozessor mit UID=43 53 32 08
 00004711 5 00 00 00 00 04 Zyklus Ende alle Loks auf allen Gleis Format Prozessoren
 (UID 00 00 00 00 Broadcast an alle)

Antwort:

Ursprünglicher Befehl mit gesetztem Response Bit.

Besonderheiten:

Wird immer von Graphical User Interface Prozessor ausgelöst.

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

2.6 Befehl: Lok Datenprotokoll

Kennung:

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

Sub-CMD:

Lok Datenprotokoll (0x05)

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 | | 6 | Loc-ID | | | | Sub-CMD Protokoll | Gleis- protokoll | | |
| | | | | | High | | | Low | | | | |

Beschreibung:

Gleisunterprotokoll ändern bzw. abweichend vom Defaultwert wählen, aber nicht das grundsätzliche Gleisprotokoll, da sonst Änderung der Loc-ID.

Also DCC bleibt DCC, MM2 bleibt MM2,

Folgende Gleisunterprotokolländerungen sind möglich (Werte für Parameter Gleisprotokoll):

Bei MM2:

- 0: MM2 2040, Ansteuerung Loks mit 20kHz und FDEC mit 40 kHz
- 1: MM2_LOK_20, Nur Ansteuerung Loks mit 20 kHz.
- 2: MM2_FKT_40, Nur Ansteuerung FDEC mit 40 kHz.

Bei DCC:

- 0: DCC Kurze Adresse, 28 Fahrstufen [=DCC-FS-Default],
- 1: DCC Kurze Adresse, 14 Fahrstufen
- 2: DCC Kurze Adresse, 126 Fahrstufen
- 3: DCC Lange Adresse, 28 Fahrstufen
- 4: DCC Lange Adresse, 126 Fahrstufen

Beispiel:

00004711 6 00 00 C0 03 05 02 DCC Adr 03 mit 126 Fahrstufen ansteuern

Antwort:

Ursprünglicher Befehl mit gesetztem Response Bit.

Bei ungültiger Loc-ID (nicht DCC nicht MM2) wird keine Antwort gesendet.

Besonderheiten:

Wird immer von Graphical User Interface Prozessor ausgelöst.

Erster Befehl nimmt Lok/Funktionsdecoder in Zyklus auf.

2.7 Befehl: Schaltzeit Zubehördecoder festlegen

Kennung:

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

Sub-CMD:

Schaltzeit (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 | System- | 1 | | 7 | Ziel Geräte-UID | | | | Sub-CMD | <Zeit> | | |
| Prio | befehl | | | | High | | | Low | Schaltzeit | High | Low | |

Beschreibung:

Defaultzeit zum Schalten von Zubehör festlegen.

Zeit kann auch beim Schaltbefehl mit angegeben werden.

Zeit in 10 ms Schritten, maximal 163 Sekunden rund 2:45 min, Wert 0 ist wieder Defaultschaltzeit.

Beispiel:

00004711 7 00 00 00 00 06 0A Schaltzeit 100 ms festlegen

Antwort:

Ursprünglicher Befehl mit gesetztem Response Bit.

Besonderheiten:

Wird immer von Graphical User Interface Prozessor ausgelöst.

2.8 Befehl: Fast Read für mfx SID - Adresse

Kennung:

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

Sub-CMD:

Fast Read (0x07)

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 | | 7 | Absender Geräte-UID | | | | Sub-CMD Fast Read | MFX-SID | | |
| | | | | | High | | | Low | | High | Low | |

Beschreibung:

Ab GFP Version 2 wird für diese MFX SID die Wartezeit zwischen den Lesebefehlen unterdrückt.

Fast Read ist nur mit Decodern der neueren Generation möglich. Mfx - Decoder der ersten Generation sind hierfür nicht ausgelegt und werden dadurch zerstört.

Fast Read kann erst nach einer Plausibilitätsprüfung durch den GFP aktiviert werden. Die UID zu dieser SID muss aus dem Märklin mfx Decoderbereich sein. (Beginnend mit 0x7F). Dies wird durch ein mfx Verify erreicht. Dieser Verify muss vor diesem Befehl ausgeführt werden und positiv beantwortet werden.

Antwort:

Immer: Ursprünglicher Befehl mit gesetztem Response Bit.

Besonderheiten:

In V1.0 wird Befehl verworfen.

Wird immer von Graphical User Interface Prozessor ausgelöst.

2.9 Befehl: Gleisprotokoll Frei Schalten

Kennung:

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

Sub-CMD:

Gleis Protokoll (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 | System-befehl | 0 1 | | 6 | Ziel Geräte-UID | | | | Sub-CMD GleisProt | Param | | |
| | | | | | High | | | Low | | | | |

Beschreibung:

Aktiviert oder sperrt entsprechende Protokolle auf dem Gleis. Parameter gibt Bitkodierte die entsprechenden Gleisprotokolle frei.

Ein gesetztes Bit gibt das entsprechende Protokoll frei, ein Gelöschtes unterdrückt es. Nach dem Reset werden alle Protokolle frei gegeben.

| Bit# | Protokoll |
|------|-----------|
| 0 | MM2 |
| 1 | MFX |
| 2 | DCC |
| 3 | Res |
| 4 | Res |
| 5 | Res |
| 6 | Res |
| 7 | Res |

Antwort:

Ursprünglicher Befehl mit gesetztem Response Bit

Besonderheiten:

GFP beherrscht nur MM2, mfx und DCC.

Bei abgeschaltetem mfx werden keine mfx-Loks mehr gesucht.

Ein Speichern der Einstellung findet im Gleis Format Prozessor nicht statt. Bei jedem Start ist dieser Parameter neu zu setzen.

2.10 Befehl: System MFX Neuanmeldezähler 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 | System- | 1 | | 7 | Ziel Geräte-UID | | | | Sub-CMD | Neuanmeldezähler | | |
| Prio | befehl | | | | High | | | Low | MFX NAZ | High | Low | |

Beschreibung:

Neuanmeldezähler des Gleis Format Prozessor verändern. Für MFX Subsystem.

Beispiel:

00004711 7 43 53 32 08 09 00 02 Setzen Neuanmeldezähler auf 2.

Antwort:

Ursprünglicher Befehl mit gesetztem Response Bit.

Besonderheiten:

Wird immer von Graphical User Interface Prozessor ausgelöst.

2.11 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 High | | | | Sub-CMD Überlast | Kanal-nummer | | |
| | | | | | | | | Low | | | | |

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. Kanalnummer wird bei der Abfrage mitgeteilt.

Beispiel:

00004711 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.

2.12 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 | Kanal-nummer | | |
| | | | | | High | | | Low | | | | |
| Message Prio | System-befehl | 0 | | 8 | Absender Geräte-UID | | | | Sub-CMD Status | Kanal-nummer | Konfigurationswert | |
| | | | | | High | | | Low | | | High | Low |
| Message Prio | System-befehl | 1 | | 7 | Ziel Geräte-UID | | | | Sub-CMD Status | Kanal-nummer | TRUE / FALSE | |
| | | | | | High | | | Low | | | | |
| Message Prio | System-befehl | 1 | | 8 | Absender Geräte-UID | | | | Sub-CMD Status | Kanal-nummer | 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.

Setzen eines Konfigurationswertes des Gleis Format Prozessor. Welche Werte gesetzt werden können, ist mittels des Befehls "Statusdaten Konfiguration" feststellbar. Erfolgreiches setzen eines Wertes wird mit einer Antwort quittiert.

Die Antwort auf Setzen eines Konfigurationswerts ist mit DLC = 7. Im Antwortbyte steht das Ergebniss der Operation: Bei Gültig wird mit TRUE (0x01), bei ungültig mit FALSE (0x00) geantwortet.

Beispiel:

00004711 6 43 53 32 08 0B 01 Abfrage des Messwertes Kanalnummer 1

00014711 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 Graphical User Interface Prozessor ausgelöst.

2.13 Befehl: Gerätekennung

Kennung:

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

Sub-CMD:

Kennung (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 | System-befehl | 0 | | 5 | Ziel Geräte-UID | | | | Sub-CMD | | | |
| | | | | | High | | | Low | Kennung | | | |
| Message Prio | System-befehl | 0 | | 7 | Ziel Geräte-UID | | | | Sub-CMD | System - Kenner | | |
| | | | | | High | | | Low | Kennung | High | Low | |
| Message Prio | System-befehl | 1 | | 7 | Ziel Geräte-UID | | | | Sub-CMD | System - Kenner | | |
| | | | | | High | | | Low | Kennung | High | Low | |

Beschreibung:

Für Geräte mit S88 Bus / Rückmeldebus und Automatisierungsmodule:

Setzen oder Mitteilen einer 16 Bit Kennung zur Bildung eindeutiger Automatik-UID.

Wird der Befehl mit fehlendem Gerätekenner gesendet, so wird nach einer gültigen Kennung gefragt.

Wird der Befehl mit Gerätekenner gesendet, so setzt dieser Befehl die Kennung.

Die Master - Zentrale weist den Endgeräten die Geräte - Kennung beim Systemstart jeweils zu. Eine Resetfeste Speicherung findet in den Geräten nicht statt.

Der Master -Zentrale speichert sich eine Liste mit allen bekannten Geräten im System, sowie deren NickNames als .cs2 Konfigurationsdatei.

Besonderheiten:

Systemkenner mit dem Wert "0" oder 0xFFFF" sind ungültig und sollten in dem Graphical User Interface Prozessor zu einer Meldung zur Eingabe einer gültigen Kennung führen.

Weiteres siehe Erläuterung der „Automatik-UID“.

2.14 Befehl: System Reset

Kennung:

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

Sub-CMD:

Set UID (**0x80**)

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 | | 6 | Ziel Geräte-UID | | | | Sub-CMD Reset | Reset-Ziel | | |
| | | | | | High | | | Low | | | | |

Beschreibung:

Zurücksetzen des Gerätes.

Besonderheiten:

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 | | 0 | | | | | | | | |
| 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 = 0: Starten "Erkennen alle Protokolle".
(später >= V2.0)

Form 2: DLC = 1: Erkennen nach Datenprotokoll, Protokoll - Kennung bestimmt Datenprotokoll, Gleis Format Prozessor macht MFX - Discovery / Lok - Erkennen 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 3: DLC = 5: 1.) Einzel - MFX - Discovery: Senden der Anforderung mit Range (=Info Länge der Bits) (Realisiert in V1.0)

2.) Sonstiges Erkennen: Aus Loc-ID werden Adresse und Protokoll extrahiert. Mit diesen Daten wird versucht, diese Adresse auf dem Gleis zu erkennen.

Antworten:

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

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

| Anfrage Programmiergleis: | | |
|---------------------------|-------------------------|--|
| Protokoll | Range/Protokoll-Kennung | Bemerkung |
| MFX: | 0-32 | =Range. Anfrage zur Erkennung der Lok auf dem Programmiergleis. Wenn die Lok antwortet, wird das Antwortbit gesetzt Sonderfall: Range=0: Es wird ein Zwangsneuanmelden eines Decoders auf dem PGL durchgeführt. |

| | | |
|---------------------|-------|---|
| MM2: | 33 | Erkennung der MM2-Lok auf dem Programmiergleis. In der Antwort wird die MM2-Adresse gemeldet. (20 kHz) |
| MM2: | 34 | Erkennung der MM2-Lok auf dem Programmiergleis. In der Antwort wird die MM2-Adresse gemeldet. (40 kHz) (>V3.0) |
| DCC: | 35 | Auslesen der DDC-Adresse auf dem Programmiergleis. Bewirkt denselben Vorgang wie 36, Antwort entweder 35 oder 36. In der Antwort wird die kurze DCC-Adresse gemeldet. |
| DCC: | 36 | Auslesen der DDC-Adresse auf dem Programmiergleis. Bewirkt denselben Vorgang wie 35, Antwort entweder 35 oder 36. In der Antwort wird die lange DCC-Adresse gemeldet. |
| DCC: | 37 | DCC-Erkennen (Intervall-Schachtelungs-Algorithmus mit DCC-K=127 Adr). In der Antwort wird die DCC-Adresse gemeldet. (>V3.0) |
| SX1: | 38 | Auslesen der SX1-Adresse auf dem Programmiergleis. In der Antwort wird die SX1-Adresse gemeldet. (>V3.0) |
| SX1: | 39 | Erkennen (Intervall-Schachtelungs-Algorithmus mit Adr 0-99) In der Antwort wird die SX1-Adresse gemeldet. (>V3.0) |
| | | |
| Anfrage Hauptgleis: | | Range = Wert mod 64. |
| MFX: | 64-96 | Anfrage zur Erkennung der Lok auf dem Hauptgleis. Wenn die Lok antwortet, wird das Antwortbit gesetzt |
| MM2: | 98 | Ungültig, da Hardwaretechnisch nicht möglich |
| DCC: | 99 | Ungültig, da Hardwaretechnisch nicht möglich |
| DCC: | 100 | Ungültig, da Hardwaretechnisch nicht möglich |
| SX1: | 101 | Ungültig, da Hardwaretechnisch nicht möglich |
| SX1: | 102 | Ungültig, da Hardwaretechnisch nicht möglich |

Beispiel:

00024711 1 20
00024711 5 FF FF FF FF 00
00024711 1 21

Voller mfx Discoveryzyklus
Range 0 MFX Discovery
MM2 Discovery

Besonderheiten:

Wird immer von Graphical User Interface Prozessor 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.

Bei diesem Kommando handelt es sich um ein sequenziell in der Abarbeitung befindlichen Programmierbefehl. Diese werden nicht in einer Befehlsqueue zwischengespeichert. Erst nach einer Antwort durch den Gleis Format Prozessor darf der nächste Programmierbefehl angefordert werden. Genau ein Programmierbefehl wird durch den Gleis Format Prozessor zeitgleich abgearbeitet.

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 | Bind | 0 1 | | 6 | MFX-UID | | | | MFX-SID | | | |
| Prio | | | | | 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.

Für mfx Decoder ist eine automatisierte Adressvergabe realisiert. Im der GUI werden diese durch den Loknamen ausgewählt und auch durch diese Kennung gefahren. Eine Schienenadresse ist hier nicht sichtbar. Das Steuergerät verwendet für die Ansteuerung der Decoder jedoch eine Adresse.

Des weiteren hat jeder Decoder eine eindeutige UID, mit welcher dieser sich im System bekannt macht. Die Kennung wird für das Anmeldeverfahren verwendet (Siehe Discovery). Durch das Bind Kommando wird dem Decoder eine kürzere Schienenadresse zugewiesen. Durch diese Schienenadresse wird der Decoder steuerbar und er nimmt am Anmeldeverfahren nicht mehr teil.

Beispiel:

00044711 6 FF FA 8C 43 00 05
00054711 6 FF FA 8C 43 00 05

Bind UID: FF FA 8C 43 auf SID: 05
Antwort

Antwort:

Ursprünglicher Befehl mit gesetztem Response Bit. Antwort zeigt das Ende der Ausführung an.

Besonderheiten:

Nur gültig für MFX.

Wird immer von Graphical User Interface Prozessor ausgelöst. Das Auslösen von Bind sollte nur vom Master - Bediengerät gemacht werden. Ansonsten sind Vorkehrungen für das gleichzeitige Auslösen von Binds zu treffen.

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

Bei diesem Kommando handelt es sich um ein sequenziell in der Abarbeitung befindlichen Programmierbefehl. Diese werden nicht in einer Befehlqueue zwischengespeichert. Erst nach einer Antwort durch den Gleis Format Prozessor darf der nächste Programmierbefehl angefordert werden. Genau ein Programmierbefehl wird durch den Gleis Format Prozessor zeitgleich abgearbeitet.

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-Verhältnis | |
| | | | | | High | | | Low | High | Low | | |

Beschreibung:

DLC = 6:

Anfrage auf Überprüfung, ob eine Lok unter der Kombination MFX-UID / MFX- SID vorhanden ist.

DLC = 7:

Sollte diese Kombination vorhanden sein, so wird eine Bestätigung mit MFX-UID und MFX-SID sowie dem ASK-Verhältnis (Qualität der Antwort) gesendet. Bei negativer Antwort wird die SID zu 0x0000 gesetzt.

Beispiel:

00064711 6 FF FA 8C 43 00 05 Verify FF FA 8C 43 auf SID 05
 00074711 6 FF FA 8C 43 00 00 Antwort negativ
 00074711 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.

Empfängt ein gebundener MFX-Decoder eine falsche Kombination aus MFX-UID und MFX-SID, so wird ein UnBIND im Lokdecoder mit der entsprechenden 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 Verifys zu treffen.

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

Bei diesem Kommando handelt es sich um ein sequenziell in der Abarbeitung befindlichen Programmierbefehl. Diese werden nicht in einer Befehlsqueue zwischengespeichert. Erst nach einer Antwort durch den Gleis Format Prozessor darf der nächste Programmierbefehl angefordert werden. Genau ein Programmierbefehl wird durch den Gleis Format Prozessor zeitgleich abgearbeitet.

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 | Lok | 0 1 | | 4 | Loc-ID | | | | | | | |
| Prio | Geschw. | | | | High | | | Low | | | | |
| Message | Lok | 0 1 | | 6 | Loc-ID | | | | Geschwindigkeit | | | |
| Prio | Geschw. | | | | High | | | Low | High | Low | | |

Beschreibung:

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.

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.

DLC = 6, Setzen der Fahrstufe:

Lok mit LOC-ID wird mit Geschwindigkeit angesteuert. Geschwindigkeit im Bereich von 0 bis 1024 (10 Bit). Für alle Protokolle wird die Geschwindigkeit auf die reale mögliche Fahrstufe umgerechnet. Fahrstufe 0 ist Lok - Haltebefehl mit eingestellter Anfahr- und Bremsverzögerung (Nicht Nothalt).

DLC = 4, Abfrage der Fahrstufe:

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

Beispiel:

| | |
|------------------------------|--|
| 00084711 6 00 00 08 03 03 20 | Lok Geschwindigkeit SX1 Adr 3, V=0x0320=800 von 1024 |
| 00084711 6 00 00 08 03 00 A0 | Lok Geschwindigkeit SX1 Adr 3, V=0x00A0=10 von 1024 |
| 00084711 6 00 00 40 01 03 20 | Lok Geschwindigkeit mfx Adr 1, V=0x0320=800 von 1024 |
| 00084711 6 00 00 C0 03 01 20 | Lok Geschwindigkeit DCC Adr 3 |
| 00084711 6 00 00 C0 03 00 A0 | Lok Geschwindigkeit DCC Adr 3 |

Antwort:

Setzen einer Geschwindigkeit:

Ursprünglicher Befehl mit gesetztem Response Bit.

Lesen der Geschwindigkeit:

Antwort in Form Geschwindigkeit setzen, wenn Lok bekannt.

Sonst fehlende Geschwindigkeitsinformation (Ursprüngliche Anfrage).

Besonderheiten:

Erster Befehl nimmt Lok/Funktionsdecoder in Zyklus auf.

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:

Abfragen oder setzen der Fahrtrichtung einer Lok

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
- 1 = Fahrtrichtung vorwärts
- 2 = Fahrtrichtung rückwärts
- 3 = Fahrtrichtung umschalten
- Rest: Richtung bleibt

Antwort:

Setzen der Richtung:

Ursprünglicher Befehl mit gesetztem Response Bit.

Abfrage der Richtung:

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

Besonderheiten:

Erster Befehl nimmt Lok/Funktionsdecoder in Zyklus auf.

Je nach Lokdecoder kann es sein, dass die Richtungsumkehr nur im Stand stattfindet. Die neue Richtung wird auf jeden Fall vom Gleis Format Prozessor gesendet.

Wird immer von Graphical User Interface Prozessor ausgelöst.

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 | Lok | 0 | | 5 | Loc-ID | | | | Funktion | | | |
| Prio | Funktion | | | | High | | | Low | | | | |
| Message | Lok | 0 1 | | 6 | Loc-ID | | | | Funktion | Wert | | |
| Prio | Funktion | | | | High | | | Low | | | | |
| Message | Lok | 0 1 | | 8 | Loc-ID | | | | Funktion | Wert | Funktionswert | |
| Prio | Funktion | | | | High | | | Low | | | | |

Beschreibung:

Lok Funktion auslösen / ausschalten oder abfragen.

Funktion ist im Bereich 0-31. 0 entspricht F0, 31 entspricht F31. Der Status weiterer Funktionen wird im Gleis Format Prozessor nicht gespeichert.

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. Dimmzustand Status aus Gleis Format Prozessor, nicht aus der Abfrage des Decoders.

DLC = 6: Aktivieren einer Funktion.

Einschalten einer Funktion. Je nach Gleisprotokoll wird die entsprechende Funktion eingeschaltet oder mit dem entsprechenden Dimmwert angesteuert. Der gültige Bereich eines Funktionswertes richtet sich nach den Lokdecodern und dem Gleisprotokoll.

DLC = 8: Aktivieren eines Wertes mit Sonderfunktionen. (Später, derzeit nicht implementiert).

Wie bei DLC = 6. Funktionswert wird an Lokdecoder übermittelt.

Besonderheiten:

Erster Befehl nimmt Lok/Funktionsdecoder in Zyklus auf.

Dimmfunktionen und Dimmwert wird nicht im Gleis Format Prozessor gehalten (Speicher). Solange der Dimmwert aktiv sein soll, wird aktiv (1) zurückgemeldet.

Wird immer von Graphical User Interface Prozessor ausgelöst.

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) | | Anzahl | |
| | | | | | High | | | Low | CV-Nummer (10 Bit) | | | |
| Message Prio | Read Config | 1 | | 7 | Loc-ID | | | | CV-Index (6 Bit) | | Wert | |
| | | | | | High | | | Low | CV-Nummer (10 Bit) | | | |
| Message Prio | Read Config | 1 | | 6 | Loc-ID | | | | CV-Index (6 Bit) | | | |
| | | | | | High | | | Low | CV-Nummer (10 Bit) | | | |

Beschreibung:

Lesen von Werten aus rückmeldefähigen 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.

CV-Nummer bestimmt, welche Konfigurationsvariable 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. Platzhalter Anzahl beinhaltet nun den gelesenen Wert.

Konnte kein Wert ausgelesen werden, so wird dies mit der negativen Quittung mit DLC = 6, also fehlendem Wert mitgeteilt.

Protokolltypische Wertebereiche und Verhalten:

DCC:

CV-Index wird nicht behandelt. CV-Nummer liegt im Bereich 1 - 1024.

Parameter "Anzahl" bestimmt die Anzahl der zu lesenden Bytes ab der angegebenen Start - CV-Nummer. Wert "00" für "Anzahl" liest 256 Bytes aus dem Decoder.

Als Antwort wird das gelesene Byte angehängt. CV-Nummer 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 nur auf dem Programmiergleis gelesen werden.

MFX:

CV-Nummer liegt im Bereich zwischen 1 und 1024.

CV-Index wird beachtet und 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. Der Gleis Format Prozessor zerlegt die Leseanfrage in Mehrbytelesebefehle und liest maximal 4 Byte gleichzeitig aus dem Decoder aus.

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.

SX1:

CV-Index wird nicht behandelt. CV-Nummer liegt im Bereich 1 - 5.

Parameter "Anzahl" bestimmt die Anzahl der zu lesenden Bytes ab der angegebenen Start - CV-Nummer "CV-Nummer". Es können maximal 5 Bytes gelesen werden.

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

Da SX1 keine Nummerierung der Konfigurationswerte kennt, wird folgende Umsetzung vorgenommen:

| CV-Nummer | Bedeutung |
|-----------|--------------------------------|
| 1 | Adresse |
| 2 | Höchstgeschwindigkeit |
| 3 | Beschleunigung |
| 4 | Impulsbreite des Motorimpulses |
| 5 | 1 / 2 – Halteabschnitte |

Ein Decoder kann nur auf dem Programmiergleis gelesen werden.

MM2:

MM2 Loks können nicht ausgelesen werden.

Besonderheiten:

Wird immer von Graphical User Interface Prozessor ausgelöst.

Der Befehl nimmt Lok/Funktionsdecoder nicht in Zyklus auf.

Je nach Protokoll nur auf dem Programmiergleis möglich.

Bei diesem Kommando handelt es sich um ein sequenziell in der Abarbeitung befindlichen Programmierbefehl. Diese werden nicht in einer Befehlqueue zwischengespeichert. Erst nach einer Antwort durch den Gleis Format Prozessor darf der nächste Programmierbefehl angefordert werden. Genau ein Programmierbefehl wird durch den Gleis Format Prozessor zeitgleich abgearbeitet.

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 | | | | CV-Index (6 Bit) | | Wert | Ctrl(2Bit) |
| | | | | | High | | | Low | CV-Nummer (10Bit) | | | |
| Message Prio | Write Config | 1 | | 8 | Loc-ID | | | | CV-Index (6 Bit) | | Wert | Rslt(2Bit) |
| | | | | | High | | | Low | CV-Nummer (10Bit) | | | |

Beschreibung:

Schreiben von CV-Werten in einen programmierbaren 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. (> V3.0)

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

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

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

Parameter Wert enthält das zu schreibende Byte. Bei Programmierart "DCC-Bitprogrammierung" hat das Datenbyte die Information im entsprechenden DCC Format also:

1111DBBB D: Wert des Bits, BBB: Bitposition.

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 Gleis Hauptgleis (Wert = 1) oder Programmiergleis (Wert = 0)

Bit 7: Multibyteschreiben Es folgen weitere Schreibbefehle.

Bit 6: DCC Programmierart 1 (DCC Register oder Direct / Bitprogrammierung)

Bit 5: DCC Programmierart 2

Anz (Bits 4 bis 0) ist Reserviert

DCC Programmierartauswahl:

| DCC1 | DCC2 | Art |
|------|------|-------------------------|
| 0 | 0 | Direct Programmierung |
| 0 | 1 | Register Programmierung |
| 1 | 0 | Bitprogrammierung |
| 1 | 1 | Reserviert |

Bestätigung:

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

Bit 8: Schreiben erfolgreich betätigt durch Controller.

Bit 7: Verifiy erfolgreich verlaufen.

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

Protokolltypische Wertebereiche und Verhalten:

DCC:

CV-Index wird nicht behandelt.

CV-Nummer liegt im Bereich 1 - 1024.

Schreiben auf Programmiergleis und Hauptgleis

Decoder kann vollständig auf dem Programmiergleis beschrieben werden.

Parameter, die mit "POM" manipuliert werden können, sind auch auf dem Hauptgleis veränderbar.

MFX:

CV-Nummer liegt im Bereich zwischen 1 und 1024.

CV-Index wird beachtet und liegt im Bereich zwischen 1 und 63.

Das Schreiben findet grundsätzlich auf Hauptgleis und Programmiergleis statt.

SX1:

CV-Index wird nicht behandelt. CV-Nummer liegt im Bereich 1 - 5.

Ein Decoder kann nur auf dem Programmiergleis programmiert werden.

Bei SX1 haben die Parameter keine Nummern. Folgende Umsetzung wird vorgenommen:

| CV-Nummer | Bedeutung |
|-----------|--------------------------------|
| 1 | Adresse |
| 2 | Höchstgeschwindigkeit |
| 3 | Beschleunigung |
| 4 | Impulsbreite des Motorimpulses |
| 5 | 1 / 2 – Halteabschnitte |

MM2:

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

Die Programmierung kann sowohl nur auf dem Programmiergleis 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 Graphical User Interface Prozessor ausgelöst.

Der Befehl nimmt Lok/Funktionsdecoder nicht in Zyklus auf.

Bei MM2 bestimmt diese Adresse die Programmieradresse, unter welcher das Programmieren stattfindet. Also auch von Adresse 80 abweichende Adressen.

Bei diesem Kommando handelt es sich um ein sequenziell in der Abarbeitung befindlichen Programmierbefehl. Diese werden nicht in einer Befehlqueue zwischengespeichert. Erst nach einer Antwort durch den Gleis Format Prozessor darf der nächste Programmierbefehl angefordert werden. Genau ein Programmierbefehl wird durch den Gleis Format Prozessor zeitgleich abgearbeitet.

4 Zubehör-Befehle

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

Eine Funktion ist immer (auch vor Zeit - Ablauf) abschaltbar.

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:

Schaltbefehl an Zubehörcodecorder senden.

DLC = 6:

Schalten mit Default Timeout / Systemtimeout des Gleis Format Prozessor (Siehe Befehl: "Schaltzeit Zubehörcodecorder")

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

- Strom 0 - 31 (Dimmfunktion, je nach Unterstützung durch Protokoll)

0: Ausschalten.

1: Einschalten.

>1: Einschalten mit Sonderfunktionswert sofern Protokoll dies unterstützt.

- 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 Zubehörcodecorder festlegen" bestimmbar, Default ist ca. 1s.)

Protokollspezifische Stellungskodierung MM3:

Bit 0,1: Stellung:

00: Aus, Rund, Rot, Rechts, HP0

01: Ein, Grün, Gerade, HP1

10: Gelb, Links, HP2

11: Weis, SH0

DCC: Nur Stellung 0 (aus) oder 1 (ein) möglich.

SX1: Nur Stellung 0 (aus) oder 1 (ein) möglich.

Besonderheiten:

Wird immer von Graphical User Interface Prozessor ausgelöst.

Schaltbefehle werden durch den Gleis Format Prozessor priorisiert gesendet, vor Fahr- / Funktionsbefehlen.

5 Rückmeldungen

Adressierung von Kontakten im System Märklin digital

S88 Kontakte können sich an mehreren Geräten im System befinden. Also an jeder CS2 oder an mehreren S88 Devices. Ziel ist die Bildung einer eindeutigen 32 Bit Kontaktkenung, welche ebenso den Austausch eines Gerätes ermöglicht.

Bildung einer Kontaktkenung aus 2 Teilen:

16 Bit "Gerätekenung", 16 Bit Kontaktnummer -> Somit maximal eine 32 Bit "UID" - Zahl.

Die Kontaktkenung wird gebildet aus einer konfigurierten Bezeichnung (dem Geräte - Kenner) und Kontaktnummer im Gerät. (Erfordert die Einrichtung der Geräte, siehe Systembefehl "Geräte Kenung") Durch diese Systematik ist der Austausch eines Gerätes möglich, ohne dass die Automatisierungsfunktionen auf die geänderten Geräteadressen angepasst werden müssten. Siehe auch Beschreibung der Rückmelde-UIDs

Kompatibilitätsmodus (S88 Polling):

Registerbasierend. Analog der Ursystematik der S88 Rückmeldung löscht das Lesen die Register.

Verwendbar, wenn genau ein Steuergerät die Rückmeldungen auswertet. Als Kompatibilitätsmodus für bestehende Steuerungsrealisierungen.

Neue Meldungssystematik im Märklin Digital System

Die Grundlage dieser Erweiterung ist:

- Änderungen an Kontakten führen zu Mitteilungen auf dem Kommunikationsbus. Pro Kontakt wird eine Meldung für das Betätigen und eine für das Freiwerden erzeugt.
- Abfrage des aktuellen Belegt - Zustandes eines Kontakts. Das Lesen behält den Zustand bei. Mehrmaliges Lesen ist möglich, auch von verschiedenen Geräten.
- *Erweiterung beim Lesen: Hierbei wird die Zeit mitgeteilt, seit der letzten Änderung des Zustandes. Wunsch: Zeit zur letzten Flanke und Zeit zur vorletzten Flanke. (Kann für Geschwindigkeits-Bestimmung verwendet werden.)?*

Maßnahmen zur Reduktion von Meldungsfluten

Diese Maßnahmen dienen der Reduktion der Busbelastung durch viele Meldungen:

- Filter für "flackernde" Kontakte:
"Frei - Entprellung" Kurze Freipausen werden gefiltert, also der Event "Frei" wird nur dann gesendet, wenn innerhalb eines Timeouts keine erneute Belegung erfolgt.
Diese Funktion kann, je nach Rückmeldegerät ebenso als generelle Entprellung stattfinden.
- Größe der Anlage -> hohe Anzahl Rückmeldekontakte
Werden Rückmeldekontakte von Busteilnehmern ausgewertet? Also nur Senden, wenn Interessent vorhanden? Dies könnte nur für CS2 interessant sein. CAN-S88 Device eher nicht, da dieses keine Kontakte auswertet.

5.1 Befehl: S88 Polling

Kennung:

S88 Polling (0x10, in CAN-ID: 0x20)

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 | S88 Polling | 0 | | 5 | Geräte UID | | | | Modul-anzahl | | | |
| | | | | | High | | | Low | | | | |
| Message Prio | S88 Polling | 1 | | 7 | Geräte UID | | | | Modul | Zustand | | |
| | | | | | High | | | Low | | | | |

Beschreibung:

Kompatibilitätsmodus zum bestehenden S88 System.

- Agiert wie die alten Befehle 6050 / 6051. "Liest den S88 Bus ein" und löscht damit die Zwischenpuffer / Register.
- Abfrage des aktuellen Zustands eines Eingangs. Liefert den Zustand des letzten Lesezyklus vom S88 Bus.

In der Anfrage wird durch die Start - Kontaktkennung das zugehörige System und S88 - Modul spezifiziert. Im Parameter Modulanzahl wird die Anzahl der abzufragenden Module spezifiziert.

Die Antwort ist immer Modulbasierend. Pro Antwort wird der Zustand von den 16 Eingängen eines S88 Moduls zurückgemeldet. Werden mehr als ein Modul abgefragt, so antwortet der Empfänger mit mehreren Antworttelegrammen, pro Modul ein Telegramm. In Kontaktkennung und Modul stehen die entsprechenden Angaben, in Zustand der aktuelle S88 Belegzustand.

Besonderheiten:

Durch das Löschen des Zwischenpuffers muss in einer Mehrgeräteumgebung jeder Teilnehmer die Antworten entsprechend "mithören"

Durch das neue Konzept des Märklin Digital Systems sollte dieser Befehl nicht verwendet werden. Es ermöglicht mehreren Teilnehmern die Auswertung von Rückmeldungen und die Behandlung mehrerer Rückmeldebuse.

5.2 Befehl: Rückmelde Event

Kennung:

S88 Event (*0x11*, in CAN-ID: *0x22*)

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 | Rückmelde Event | 0 | | 4 | Gerätekennung | | Kontaktkennung | | | | | |
| | | | | | High | Low | High | Low | | | | |
| Message Prio | Rückmelde Event | 0 | | 5 | Gerätekennung | | Kontaktkennung | | Parameter | | | |
| | | | | | High | Low | High | Low | | | | |
| Message Prio | Rückmelde Event | 1 | | 8 | Gerätekennung | | Kontaktkennung | | Zustand alt | Zustand neu | Zeit | |
| | | | | | High | Low | High | Low | | | | |

Beschreibung:

Meldung für Zustandsänderungen im System. Dabei ist die Meldung so gestaltet, dass auch Analogwerte mitgeteilt werden können.

Form 1, DLC = 4: Abfrage des aktuellen Status eines Eingangs, Antwort mit DLC = 8

Form 2, DLC = 5: Einen Eingang für verteilte Anwendung anmelden und das Versenden von Statusänderungen einschalten, Antwort mit DLC = 8.

Die Antwort auf ein Kommando erfolgt immer mit DLC=8. In der Antwort wird der aktuelle Status des Rückmelder mitgeteilt. Weitere Teilnehmer können diese Meldung verwenden, damit der interne Status richtig gestellt werden kann.

Welche Eigenschaften ein Rückmelder hat, kann derzeit noch nicht festgestellt werden. Angedacht sind hier Zählmodule, Geschwindigkeitsmodule, etc.

6 Sonstige Befehle

6.1 Befehl: Softwarestand Anfrage / Teilnehmer Ping

Kennung:

Softwarestand / Teilnehmer 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 1 | | 0 8 | Absender Geräte UID | | | | SW-Versionsnummer | | Geräteerkennung | |
| | | | | | High | | | Low | | | | |

Beschreibung:

Jedes Gerät antwortet mit den entsprechenden Daten. Damit wird die Konfigurationsabfrage aller am CAN Bus erreichbarer Teilnehmer erreicht.

DLC = 0:

Abfrage aller Teilnehmer am Bus.

DLC = 8

Bei der Antwort wird die UID durch die des antwortenden Gerätes ersetzt. Somit kann die Graphical User Interface Prozessor bestimmen, welche Geräte angeschlossen sind.

Versionsnummer ist ein Kennung der Softwareversion.

In Byte 6 und Byte 7 (DB) steht Big-Endian codiert die Typ-Information des Geräts. Zur Zeit sind folgende Gerätekennungen festgelegt:

| Geräteerkennung | Gerät |
|-----------------|---|
| 0x00 0x00 | Gleis Format Prozessor 60213,60214 / Booster 60173, 60174 |
| 0x00 0x10 | Gleisbox 60112 und 60113 |
| 0x00 0x20 | Connect 6021 Art-Nr.60128 |
| 0x00 0x30 | MS 2 60653, Txxxxx |
| 0xFF 0xE0 | Wireless Devices |
| 0xFF 0xFF | CS2-GUI (Master) |

Besonderheiten:

Es antworten nur die Steuergeräte. Keine Lokdecoder.

6.2 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 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 | | 6 | Geräte-UID | | | | Index | Paket-anzahl | | |
| | | | | | High | | | Low | | | | |

Beschreibung:

Abfrage der Beschreibung der Messwertdaten und der Konfigurationsdaten eines Gerätes.

Messwertdaten eines Endgerätes:

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 Indices für Messwerte beginnen bei 1 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 der 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:

| Typ | Bedeutung |
|--------|----------------------------------|
| Char | Anzahl der Messwerte im Gerät. |
| Char | Anzahl der Konfigurationskanäle |
| 2 Byte | frei. |
| U32 | Seriennummer CS2. |
| String | 8 Byte Artikelnummer. |
| String | Gerätebezeichnung, \0 Terminiert |

Format des Datenblocks zur Beschreibung eines Messwert:

Die folgenden Indices enthalten eine zum Messwert gehörende Beschreibung:

Format Messwertbeschreibung

| Typ | Bedeutung | Beispiel |
|------|----------------------|--------------------------------------|
| Char | Abfragekanalnummer | 0x01: Abfrage unter Kanal 1 |
| Char | Potenz des Messwerts | -3 Bedeutet: 10^{-3} |
| Char | Farbe Bereich 1 | 0x00: Farbdarst. Sw, SW-Darst: Sw |
| Char | Farbe Bereich 2 | 0x31: Farbdarst. Grün, SW-Darst: Gr1 |
| Char | Farbe Bereich 3 | 0xF2: Farbdarst. Gelb, SW-Darst: Gr2 |

| | | |
|--------|---------------------|-----------------------------------|
| Char | Farbe Bereich 4 | 0xC3: Farbdarst. Rot SW-Darst: Ws |
| U16 | Nullpunkt | 0 |
| U16 | Ende Bereich 1 | 1800 |
| U16 | Ende Bereich 2 | 2000 |
| U16 | Ende Bereich 3 | 2100 |
| U16 | Ende Bereich 4 | 2250 |
| String | Messwertbezeichnung | Hauptgleisstrom\0 |
| String | Bezeichnung Start | 0.000\0 |
| String | Bezeichnung Ende | 2.500\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 |
|-----------|-----------|-----------|-----------|
| | | | |

Werte aus Strings " Bezeichnung Start" und "Bezeichnung Ende":
 0 Ende aus String 2.500

Mitgeteilte Messwerte aus dem obigen Bereichsangaben:

Nullpunkt
 0 300 325 375 400 Ende Bereich 4

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 |
|-----------|-----------|-----------|-----------|
| | | | |

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.

Nullpunkt bestimmt, ab welchem Messwert der Wert angezeigt werden soll.

Messwertbezeichnung:

Liefert eine Bezeichnung zum Messwert.

Internationalisierung:

Messwerte müssen in der Graphical User Interface Prozessor internationalisiert werden können. Damit dies realisiert werden kann, sind folgende vordefinierte Begriffe festgelegt:

| Messwertbezeichnung | Bezeichnung Deutsch | Bez. Englisch |
|---------------------|---------------------|---------------|
| HGL | Hauptgleis | Maintrack |
| PGL | Programmiergleis | Programtrack |
| TRACK | Boostergleis | Boostertrack |
| VOLT | Versorgungsspannung | Supplyvoltage |
| ACHSEN | Achsen | Axes |
| 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.

Format des Datenblocks zur Beschreibung eines Konfigurationskanals:

Ein Konfigurationskanal liefert eine Möglichkeit zum Einstellen von Betriebsparametern eines Endgeätes. Die Abfragenummern der Konfigkanäle folgen unmittelbar auf die Messwertkanäle.

Ein Konfigurationskanal hat eine Konfigkanalnummer, welche mit einem Messwertkanal korrespondieren kann. Die GUI hat damit die Möglichkeit, den Messwertkanal einer Einstellmöglichkeit zuzuordnen. Somit ist es möglich, z.B. die maximale Stromabgabe des HGL durch einen Konfigkanal zu beeinflussen.

Ist die Konfigkanalnummer ausserhalb des Bereiches der Messwertkanäle, so existiert zu dieser Einstellmöglichkeit keine Messwertabfrage.

Für Konfigkanäle sind 2 Grundformate definiert:

- 1.) Eine Auswahlliste mit deren Bezeichnern zur Auswahl von diskreten Werten (Drop Down)
- 2.) Eine Analoge Auswahl mit der Möglichkeit der gleitenden Eingabe von Werten (Slider).

Format eines Datenblocks mit der Möglichkeit eine Auswahl zu treffen:

| Typ | Bedeutung | Beispiel |
|--------|------------------------------|-------------------------------|
| Char | Konfigkanalnummer | 0x01: Setzen unter Kanal 1 |
| Char | Kenner Auswahlliste | Wert 1 |
| Char | Anzahl der Auswahlpunkte | 2 - 255 |
| Char | Jetzige (Default)Einstellung | |
| Char | 4 Res | |
| String | Auswahlbezeichnung | Betrieb mit Stromversorgung\0 |
| String | Auswahl 1 | Trafo 60 VA 60052\0 |
| String | Auswahl 2 | Schaltnetzteil 60061 \0 |
| String | Auswahl 3 | Schaltnetzteil 60101\0 |

Es müssen so viele Auswahlstrings wie Optionen mitgeteilt werden.

Auswahl 0 wird der Wert 0 zugeordnet, Auswahl 1 der Wert 1. Ausgewählte Optionen werden im Endgerät entsprechend umgesetzt.

Das Setzen eines Auswahlpunktes geschieht mittels dem Befehl: „System Status“ und einer DLC = 8. Der hierzu in diesem Befehl benötigte Kanal ist die Konfigurationskanalnummer.

Format eines Datenblocks mit der Möglichkeit einen Wert einzustellen:

| Typ | Bedeutung | Beispiel |
|--------|---------------------------|----------------------------|
| Char | Konfigurationskanalnummer | 0x05: Setzen unter Kanal 5 |
| Char | Kenner Slider | Wert 2 |
| Word | Unterer Wert | 0 |
| Word | Oberer Wert | 660 |
| Word | Aktuelle Einstellung | 500 |
| String | Auswahlbezeichnung | Variable Strombegrenzung\0 |
| String | Bezeichnung Start | 0.000\0 |
| String | Bezeichnung Ende | 2.500\0 |
| String | Einheit | Achsen\0 oder A\0 |

Die Umrechnung der Eingabewerte zur Anzeige geschieht analog der Anzeige von Messwerten.

Setzen der Einstellung entsprechend dem obigen System.

Besonderheiten:

Wird immer von Graphical User Interface Prozessor ausgelöst.

Kanalnummer und Kanalindex müssen nicht übereinstimmen.

Bei diesem Kommando handelt es sich um ein sequenziell in der Abarbeitung befindlichen Programmierbefehl. Diese werden nicht in einer Befehlsqueue zwischengespeichert. Erst nach einer Antwort durch den Gleis Format Prozessor darf der nächste Programmierbefehl angefordert werden. Genau ein Programmierbefehl wird durch den Gleis Format Prozessor zeitgleich abgearbeitet.

7 GUI Informationsübertragung

Konfigurationsabgleich mehrerer Graphical User Interface Prozessors.

7.1 Befehl: Anfordern Config Data

Kennung:

Anfordern Config Data (0x20, in CAN-ID: 0x40)

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 | Konfig Data | | Hash | 8 | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |

Beschreibung:

Anfordern einer Configdatei. Darin enthalten ist ein 8 Byte Dateiname.

Bestätigen der Anforderung mit gesetztem Resp. Bit.

Die folgenden Dateinamen werden von der CS2 unterstützt:

| Dateiname | Ab-sender | Bedeutung | Folge-Info | Stream-Datenformat |
|------------|----------------|---|---|---|
| "lokinfo" | MS2 | Daten einer Lok | Anfang des Lok-Namens 2 Pakete mit zusammen max. 16 Zeichen | Text (Ausschnitt aus "lokomotive.cs2") |
| "loknamen" | MS2 | Namen von n Loks und Gesamtzahl | 1 Datagram mit Nr. der ersten Lok und Anzahl durch Leerzeichen getrennt | Text: Lok-Nr, Lok-Name und Gesamtanzahl der Loks im .cs2-Format. |
| "maginfo" | MS2 | Daten von n Magnetartikeln | 1 Datagram mit Adresse des ersten Magnet- artikels und Anzahl durch Leerzeichen getrennt | Text (Ausschnitt aus "magnetartikel.cs2") |
| "lokdb" | MS2 | Lokdatenbank | - | Binär (MS2-Lokdb-Format) |
| „lang“ | MS2 | Sprachendatei | - | Binär (MS2-Sprachen-Format) |
| „ldbver“ | MS2 | Versionsinfo der Lokdatenbank | - | Text: Versionsnummer, Monat, Jahr, Anzahl der Einträge im .cs2-Format |
| „langver“ | MS2 | Versionsinfo der Sprachendatei | - | Text: Versionsnummer und Größe der Sprachendatei im .cs2-Format |
| „loks“ | CS2 (Slave) | Komplette Lokliste | - | Konfigurationsdatei „lokomotive.cs2“ Zlib-komprimiert |
| „mags“ | CS2 (Slave) | Komplette Magnetartikel- konfiguration | - | Konfigurationsdatei „magnetartikel.cs2“ Zlib-komprimiert |
| „gbs“ | CS2 (Slave) | Komplette Gleisbildgrund- konfiguration | - | Konfigurationsdatei „gleisbild.cs2“ Zlib-komprimiert |
| „gbs-#“ | CS2 | Konfiguration einer | - | Konfigurationsdatei einer |

| | (Slave) | Gleisbildseite | | Gleisbildseite Zlib-komprimiert |
|------------|----------------|--|---|--|
| „fs“ | CS2 (Slave) | Komplette Fahrstraßen- konfiguration | - | Konfigurationsdatei „fahrstrassen.cs2“ Zlib-komprimiert |
| „loksstat“ | CS2 (Slave) | Aktueller Zustand der Loks | - | Statusdatei „lokomotive.sr2“ Zlib-komprimiert |
| „magstat“ | CS2 (Slave) | Aktueller Zustand der Magnetartikel | - | Statusdatei „magnetartikel.sr2“ Zlib-komprimiert |
| „gbsstat“ | CS2 (Slave) | Aktueller Zustand einer Gleisbildseite | - | Statusdatei „gleisbild.sr2“ Zlib-komprimiert |
| fsstat | CS2 (Slave) | Reserve für Fahrstrassen | - | <i>Nicht implementiert</i> |

Wenn in der Spalte Folge-Info etwas angegeben ist, erwartet die Master-CS2 nach der Quittung **eine(?)** weitere Config-Data-Anfrage, in der die erste Anfrage weiter qualifiziert wird.

Besonderheiten:

Kann vom PC auch angefordert werden (für Datensicherung).

7.2 Befehl: Config Data Stream

Kennung:

Config Data Stream (**0x21**, in CAN-ID: **0x42**)

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 | CFG Data Stream | | | 6 | Dateilänge / Streamlänge in Bytes | | | | CRC | | | |
| | | | | | High | | | Low | High | Low | | |
| Message Prio | CFG Data Stream | | | 7 | Dateilänge / Streamlänge in Bytes | | | | CRC | | Byte 6 | |
| | | | | | High | | | Low | High | Low | | |
| Message Prio | CFG Data Stream | | | 8 | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |

Beschreibung:

Überträgt den Datenstrom einer Konfigurationsdatei.

Folgt unmittelbar auf das (letzte) Antwortframe der "Config Data Anforderung" oder zum Mitteilen einer Konfigurationsänderung per Broadcast. Durch die DLC wird der Inhalt unterschieden:

DLC = 6:

Ist das erste Paket im Datenstrom. Es definiert den Datenstrom als Antwort auf eine Config Data Anfrage. Es enthält die Dateilänge in Bytes, sowie die CRC der zu übertragenden Bytes.

Bei diesem Typ des Datenstroms wird zur Kollisionsauflösung der Hash des **Empfängers** verwendet.

Erweiterung: Dies kann auch die Antwort auf eine "blokinfo" Frage sein. Dieser Text wird zu einem Rückmeldekontakt / Block dargestellt. Bietet somit die Möglichkeit einer freien Textdarstellung im Gleisbild.

DLC = 7:

Ist das erste Paket im Datenstrom. Es definiert den Datenstrom als Broadcast einer Konfigänderung. Es enthält die Dateilänge in Bytes, sowie die CRC der zu übertragenden Bytes, sowie einem reservierten Byte. Der Datenstrom ist bei dieser Übertragungsform immer(?) Zlib-Komprimiert.

Bei diesem Typ des Datenstroms wird zur Kollisionsauflösung der Hash des **Senders** verwendet.

DLC = 8:

Datenpaket: Es enthält immer 8 Byte an Daten. Stehen weniger Daten zur Verfügung, so wird mit 0x00 gepaddet. Wird immer ab dem 2.ten Datenpaket gesendet. Hash richtet sich nach obiger Systematik.

Alle anderen DLC-Längen:

Tritt im Datenstrom eine andere Länge auf, so ist der Empfang abubrechen bzw. zu beenden.

Besonderheiten:

Pakete werden nicht bestätigt und werden mit niedriger Prio im CAN übertragen.

Der CRC ist ein 16-Bit-CRC mit dem Startwert 0xFFFF und dem Polynom 0x1021, der nach folgendem Algorithmus berechnet wird.

```
u16 CtDataSender::updateCRC (u16 CRC_acc, u8 CRC_input)
{
    #define POLY 0x1021
    // Create the CRC "dividend" for polynomial arithmetic (binary arithmetic with no carries)
    CRC_acc = CRC_acc ^ (CRC_input << 8);
    // "Divide" the poly into the dividend using CRC XOR subtraction CRC_acc holds the
    // "remainder" of each divide. Only complete this division for 8 bits since input is 1 byte
    for (int i = 0; i < 8; i++) {
        // Check if the MSB is set (if MSB is 1, then the POLY can "divide" into the "dividend")
        if ((CRC_acc & 0x8000) == 0x8000) {
            // if so, shift the CRC value, and XOR "subtract" the poly
            CRC_acc = CRC_acc << 1;
            CRC_acc ^= POLY;
        }
        else {
            // if not, just shift the CRC value
            CRC_acc = CRC_acc << 1;
        }
    }
    // Return the final remainder (CRC value)
    return CRC_acc;
}
```

8 Format der Konfigurationsdateien der CS2

Die Formatdateien der CS2 unterliegen einer beständigen Weiterentwicklung. Erweiterungen zu den bisherigen Inhalten sind jederzeit möglich. Unbekannte / bisher nicht dokumentierte Einträge sind daher möglich.

8.1 Konfigurationsdatei "Lokomotive.cs2"

Diese Datei enthält die Konfigurationsdaten aller Lokomotiven der CS2.

Grundformat:

```
[lokomotive]
version
.major=0
.minor=3
session
.id=24
lokomotive
.uid=0xc002
.name=Dcc-2
.vorname=Dcc-2
.adresse=0x3e8
.typ=dcc
.sid=0xcdcd
.mfxuid=0xffffffff
.icon=
.symbol=0
.av=30
.bv=30
.volume=2
.progmask=0x2
.velocity=0
.richtung=0
.tachomax=350
.vmax=255
.vmin=1
.xprot=3
.mfxtyp=205
.funktionen
..nr=0
..typ=1
..dauer=0
..wert=0
.inTraktion=0xffffffff
```

8.1.1 Sektion "version"

Beschreibung:

Entält die Versionsnummer der Konfigurationsdatei. Wird verwendet um die Rückwärtskompatibilität herzustellen. Diese Sektion darf nur einmal vorhanden sein.

Felder:

```
.major={Wert}
.minor={Wert}
```

Aktuelle Werte sind .major=0 und .minor=3

8.1.2 Sektion "session"

Beschreibung:

Der Wert des aktuellen mfx Neuanmeldezähler. Der MFX Neuanmeldezähler wird zur Anmeldung von mfx-lokomotiven benötigt. Diese Sektion darf nur einmal vorhanden sein.

Felder:

```
.id={Wert}
```

Beginnend bei 1. Max ist 65536.

8.1.3 Sektion "lokomotive"

Beschreibung:

Enthält die Konfigurationsdaten einer Lokomotive.

Felder:

| | |
|------------------------|---|
| .uid=0xc002 | UID in Hexadezimaler Darstellung. |
| .name=Dcc-2 | aktueller Name der Lokomotive. |
| .vorname=Dcc-2 | Name vor der letzten Änderung (erlaubt Namensänderungen an andere CS2-en zu kommunizieren) |
| .adresse=0x3e8 | Adresse als Hex. |
| .typ=dcc | Decodertyp: mm2_prg, mm2_dil8, dcc, mfx, sx1 |
| .sid=0xcdcd | bei mfx: Die SID |
| .mfuid=0xffffffff | bei mfx: UID des Decoders |
| .icon= | Leer: noch kein Icon gewählt, sonst: Dateiname ohne Endung |
| .symbol=0 | Lok-Symbol für MS1 (0=Elok, 1=Diesellok, 2=Dampflo, 3=kein Icon) |
| .av=30 | Anfahrverzögerung |
| .bv=30 | Bremsverzögerung |
| .volume=2 | Lautstärke |
| .progmask=0x2 | Interner Merker für Lokprogrammierung, zu programmierende Werte |
| .velocity=0 | Fahrgeschwindigkeit der Lok |
| .richtung=0 | Fahrtrichtung der Lok |
| .tachomax=350 | Endausschlag des Tachos |
| .vmax=255 | Maximalgeschwindigkeit der Lok |
| .vmin=1 | Minimalgeschwindigkeit der Lok |
| .xprot=3 | Erweitertes Protokoll (Fahrstufen, Unterprotokoll) |
| .mfxtyp=205 | Typ des mfx Decoders |
| .funktionen | Eröffnung einer 2.Ebene. Es folgen die 16 Funktionen. Pro Erweiterung ein Feld Parameter der Ebene siehe Kapitel funktionen |
| .inTraktion=0xffffffff | kein Mitglied einer Traktion |
| .prg | Eröffnung einer 2.Ebene. Pro Erweiterung ein Feld Erweiterter Programmiermodus. Es folgen die Definitionen des Konfigurationsdialogs. Pro Zeile ein Eintrag. Parameter der Ebene siehe Kapitel prg |

8.1.3.1 2. Ebene Sektion ".funktionen"

Beschreibung:

Beschreibt das Verhalten und Aussehen einer Funktion auf der Oberfläche.

Felder:

| | |
|-----------|---|
| ..nr=0 | Beschreibung 0 -> F0 15 -> F15 |
| ..typ=1 | Datei-Nummer des F-Icons (hier Licht) |
| ..dauer=0 | Auslösedauer, 0: Dauerfunktion, -1 Momentfunktion, sonst Zeitfunktion |
| ..wert=0 | aktueller Zustand |

8.1.3.2 2. Ebene Sektion ".prg"

Beschreibung:

Beschreibt eine Zeile in der erweiterten CV-Programmierung. Diese finden Verwendung innerhalb des erweiterten CV-Zugriffs.

Felder:

| | |
|----------------|----------------------------|
| ..adresse=1 | Nummer der CV |
| ..name=Cv Name | Name der CS |
| ..wert=3 | aktuell eingetragener Wert |
| ..maske=0 | Bitmaske |

8.2 Konfigurationsdatei "magnetartikel.cs2"

Enthält die Konfiguration und Stellungen der Magnetartikel

Grundformat:

```
[magnetartikel]
version
.major=0
.minor=1
artikel
.id=1
.name=src
.typ=lichtsignal_HP012_SH01
.stellung=2
.schaltzeit=200
.ungerade=0
.dectyp=mm2
```

8.2.1 Sektion "version"

Beschreibung

Enthält die Versionsnummer der Konfigurationsdatei. Wird verwendet um die Rückwärtskompatibilität herzustellen. Diese Sektion darf nur einmal vorhanden sein.

Felder:

.major={Wert}

.minor={Wert}

Aktuelle Werte sind .major=0 und .minor=1

8.2.2 Sektion "artikel"

Beschreibung

Enthält die Beschreibung genau eines Magnetartikels.

Felder:

| | |
|---------------------|------------------------------------|
| artikel | Beginn Beschreibung eines Artikels |
| .id=1 | Adresse des Artikels |
| .name=Signal 1 | Name/Bezeichnung des Artikels |
| .typ= std_rot_gruen | Typ des Artikels. Mögliche Werte: |
| | std_rot_gruen |
| | std_rot |
| | std_gruen |
| | entkupplungsgleis |
| | entkupplungsgleis_1 |
| | rechtsweiche |
| | linksweiche |
| | y_weiche |
| | k84_ausgang |
| | k84_doppelausgang |
| | dreiwegweiche |
| | DKW 2 Antriebe |
| | DKW 1 Antrieb |
| | lichtsignal_HP01 |
| | lichtsignal_HP02 |
| | lichtsignal_HP012 |
| | lichtsignal_HP012_SH01 |
| | lichtsignal_SH01 |
| | formsignal_HP01 |

| | |
|-----------------|---|
| | formsignal_HP02 |
| | formsignal_HP012 |
| | formsignal_HP012_SH01 |
| | formsignal_SH01 |
| | urc_lichtsignal_HP01 |
| | urc_lichtsignal_HP012 |
| | urc_lichtsignal_HP012_SH01 |
| | urc_lichtsignal_SH01 |
| | schiebebuehne |
| | drehscheibe_alt |
| | digitaldrehscheibe |
| .stellung=1 | aktuelle Stellung |
| .schaltzeit=200 | eingestellte Schaltzeit |
| .ungerade=0 | Bei Artikeln mit einem Begriff relevant |
| .dectyp=mm2 | Decodertyp, entweder dcc oder mm2 |

Konfigurationsdatei "fahrstrasse.cs2"

Enthält die Konfiguration und Stellungen der Magnetartikel. Jede Konfiguration kann eine der vorkonfigurierten Plätze belegen.

Grundformat:

```
[fahrstrassen]
version
  .major=0
  .minor=1
fahrstrasse
  .id=1
  .name=FS1ein
  .s88=1
  .s88Ein=1
  .extern=1
  .item
  ..fsverweis=-1
  ..magnetartikel=23
  ..stellung=1
```

8.2.3 Sektion "version"

Beschreibung

Entält die Versionsnummer der Konfigurationsdatei. Wird verwendet um die Rückwärtskompatibilität herzustellen. Diese Sektion darf nur einmal vorhanden sein.

Felder:

.major={Wert}
.minor={Wert}

Aktuelle Werte sind .major=0 und .minor=1

8.2.4 Sektion "fahrstrasse"

Beschreibung

Listet einen Eintrag einer Fahrstrasse auf

Felder:

| | |
|--------------|--|
| .id=1 | Nummer der Fahrstrasse |
| .name=FS1ein | Name der Fahrstrasse |
| .s88=1 | S88-Kontakt, der diese Fahrstrasse beim Schließen auslöst, 0 wenn keine S88-Steuerung gewünscht |
| .s88Ein=1 | S88-Kontakt, der diese Fahrstrasse beim Öffnen auslöst, ansonsten 0 |
| .extern=1 | Handbetrieb (0) oder gemischter Automatikbetrieb. |
| .item | Beginn einer weiteren Subsektion. Bereich mit zu schaltendem Artikeln oder Fahrstrassen. Pro Artikel ein Abschnitt. |

8.2.4.1 2. Ebene Sektion ".item"

Beschreibung:

Listet die zu schaltenden Elemente der Fahrstrasse auf.

Felder:

| | |
|--------------------|--|
| ..fsverweis= -1 | Verweis auf andere Fahrstrasse. Entweder Nummer der Fahrstrasse oder -1 für Magnetartikel. Bei Verweis auf weitere Fahrstrasse ist ..magnetartikel und ..stellung auf 0 zu setzen. |
| ..magnetartikel=23 | Magnetartikel, Nummer aus Keyboard, 0 bei Verweis auf weitere Fahrstrasse |

..stellung=1

Zu schaltende Stellung, je nach Artikel zwischen 1 und 4 oder 0 bei Verweis auf weitere Fahrstrasse.

8.3 Konfigurationsdatei "gleisbild.cs2"

Enthält die Konfiguration des Gleisbildstellpultes

Grundformat:

```
[gleisbild]
version
  .major=0
  .minor=6
groesse
  .width=20
  .height=15
zuletztBenutzt
  .name=Haupt-alt
seite
  .id=0
  .name=Haupt-alt
seite
  .id=1
  .name=Hauptbahnhof
element
  .id=0x301
  .typ=s88kontakt
  .drehung=0
  .artikel=2
  .text=
```

8.3.1 Sektion "version"

Beschreibung

Entält die Versionsnummer der Konfigurationsdatei. Wird verwendet um die Rückwärtskompatibilität herzustellen. Diese Sektion darf nur einmal vorhanden sein.

Felder:

.major={Wert}
.minor={Wert}

Aktuelle Werte sind .major=0 und .minor=6

8.3.2 Sektion "groesse"

Beschreibung

Anzahl der Elemente pro Seite. Diese Sektion darf nur einmal vorhanden sein.

Felder:

.width=20 Feste Einstellung
.height=15 Feste Einstellung

8.3.3 Sektion "zuletztBenutzt"

Beschreibung

Definition der zuletzt benutzten Gleisbildseite

Felder:

.seite=0 Name der Seite

8.3.4 Sektion "seite"

Beschreibung

Definition einer Stellpultseite. Die Definitionen der Stellpultseiten müssen hintereinander folgen. Pro Seite eine Sektion.

Felder:

| | |
|--------------|------------------|
| .id=0 | Nummer der Seite |
| .name=Seite0 | Namen der Seite |

8.3.5 Sektion "element"

Beschreibung

Festlegung eines Elementes einer Stellpultseite. Pro Element eine Definition.

Felder:

| | |
|--------------|---|
| .id=0x010301 | Hexadezimal ssxxyy: ss Seite, xx x-Koord., yy y-Koord (also Seite=1, x=3, y=1) |
| .typ=gerade | Typ des Elements. Mögliche Werte: |
| | leer gerade kreuzung unterfuehrung prellbock bogen doppelbogen tunnel linksweiche rechtsweiche dreiwegweiche yweiche dkweiche dkweiche_2 signal s88kontakt s88bogen pfeil fahrstrasse text signal_hp02 signal_hp012 signal_hp01s signal_p_hp012s signal_f_hp012s signal_p_hp012 signal_f_hp01 signal_f_hp02 signal_f_hp012 signal_sh01 k84_einfach k84_doppelt entkuppler entkuppler_1 std_rot std_gruen std_rot_gruen_0 std_rot_gruen_1 schiebebuehne_0 schiebebuehne_1 schiebebuehne_2 schiebebuehne_3 drehscheibe_alt_0 drehscheibe_alt_1 |

(Verweis auf andere Gleisbildseite)

| | |
|-------------|---|
| | drehscheibe_alt_2 |
| | drehscheibe_alt_3 |
| | drehscheibe_dig_0 |
| | .. bis |
| | drehscheibe_dig_31 |
| .drehung=1 | Anzeigerichtung, im Uhrzeigersinn von 0 bis 3. |
| .artikel=-1 | Adresse des Magnetartikels im Keyboard, -1 für keine Äquivalenz |
| .text= | Beschriftung der Elementes |

8.4 Konfigurationsstream "lokinfo"

Diese und die folgenden Informationen liegen nicht oder nicht dauerhaft auf der CS2 in Dateiform vor, sie werden temporär über das Streaming-Protokoll an die anfordernden Stationen gesandt.

"lokinfo" enthält einen – für die anfordernde Station (z.B.) MS2 ausreichenden – Teilausschnitt aus der Lokokonfigurationsdatei der CS2.

Grundformat:

```
[lokomotive]
lok
  .uid=0x4005
  .name=M4
  .adresse=0x9
  .typ=mfx
  .mfxuid=0xff001234
  .av=64
  .bv=48
  .volume=64
  .vmax=255
  .vmin=12
  .fkt
  ..nr=0
  ..typ=32
  .fkt
  ..nr=1
  ..typ=12
  .fkt
  ..nr=2
  ..typ=38
  .fkt
  ..nr=3
  ..typ=43
  .fkt
  ..nr=4
  ..typ=91
  .fkt
  ..nr=5
  ..typ=22
  .fkt
  ..nr=6
  ..typ=39
  .fkt
  ..nr=7
  ..typ=31
  .fkt
  ..nr=8
  ..typ=19
  .fkt
  ..nr=9
  ..typ=133
  .fkt
  ..nr=10
  ..typ=27
  .fkt
  ..nr=11
  ..typ=29
  .fkt
  ..nr=12
  ..typ=89
  .fkt
  ..nr=13
  ..typ=16
  .fkt
  ..nr=14
  ..typ=14
  .fkt
  ..nr=15
  ..typ=4
  .mfxAdr
```

```
..target=2
..name=3
..addr=136
..xcel=78
..speedtable=79
..volume=117
..numfunc=16
..func=21
```

Als einzigen Zusatz enthält "lokinfo" für mfx-Loks in der Untersektion .mfxAdr die Adress-Information der benötigten mfx-Konfig-Einträge.

8.5 Konfigurationsstream "loknamen"

"loknamen" enthält die angeforderte Anzahl von Loks aus der Lokliste der CS2 zusammen mit der Information, wie viele Loks insgesamt auf der CS2 verfügbar sind. Die MS2 z.B. fordert jeweils 2 Einträge an, die sie auf ihrem Display anzeigt; beim Hoch- oder Herunterscrollen werden jeweils die vorangehenden oder nachfolgenden Loknamen angefordert.

Grundformat:

```
[lokomotive]
lok
  .nr=3
  .name=MaK 1206 ACTS
lok
  .nr=4
  .name=M4-RIOs
numloks
  .wert=8
```

8.6 Konfigurationsstream "maginfo"

"maginfo" enthält den angeforderten Ausschnitt der Magnetartikel-Konfigurationsdatei.

Grundformat:

```
[magnetartikel]
artikel
  .id=3
  .name=1.3
  .typ=rechtsweiche
  .schaltzeit=200
  .dectyp=mm2
artikel
  .id=4
  .name=1.4
  .typ=entkupplungsgleis
  .schaltzeit=200
  .dectyp=mm2
```

8.7 Konfigurationsstream "lokdb"

lokdb enthält die Lokdatenbank als Binär-Stream

8.8 Konfigurationsstream "ldbver"

ldbver enthält die Versionsinformation der auf der CS2 verfügbaren Lok-Datenbank

Grundformat:

```
.version=9  
.monat=2  
.jahr=10  
.anzahl=1078
```

9 Automatisierung

Automatisierung bezieht sich auf derzeitige "memory" Funktionen und auch auf die Erweiterung mit Blöcken und komplexen Ablaufsteuerungen.

Automatisierungsbefehle befinden sich derzeit in der Realisation. Daher kann sich das exakte Format der Befehle ändern.

9.1 Befehl: Automatik schalten

Kennung:

Automatik schalten (0x30, in CAN-ID: 0x60)

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 | | 0 | | 6 | Gerätekenner | Automatik Funktion | Stellung | Parameter | | | | |
| | | | | | High | Low | High | Low | | | | |
| Message Prio | | 1 | | 6 | Gerätekenner | Automatik Funktion | Status | Parameter | | | | |
| | | | | | High | Low | High | Low | | | | |
| Message Prio | | 1 | | 8 | Gerätekenner | Automatik Funktion | Loc-ID | | | | | |
| | | | | | High | | Low | High | | | | Low |

Beschreibung:

Mitteilung von Statusinformationen zu Automatisierungsblöcken, bzw "memory"-Funktion, Automatikfunktion in CS2 auslösen.

Mit Gerätekenner und Automatik Funktion wird bestimmt, wer welche "Memory Funktion" auslösen soll.

Gerätekenner wird mit der Systemfunktion "Geräte Kennung" zugeordnet.

"Memory Funktion" ist die Automatikfunktion zum Auslösen. Dabei ist im Highbyte der Zeilen-Index der Automatikfunktion (Wertebereich: A-Z, a-z) und im Lowbyte der Spaltenindex (Wertebereich:1-8). Wird eine Erweiterung benötigt, so wird der Wertebereich erweitert.

Anfordern einer Automatikfunktion, Stellung bestimmt die Auslösung:

0 für Freigeben / Deaktivieren

1 für Aktivieren.

Beim Ausbau der Automatikfunktionen können weitere Stati eingeführt werden (wie Sperren, Reservieren oder Freigeben).

Mitteilen einer Statusänderung, Verwendung für Anzeige und Ablaufkontrolle:

Bei der Antwort wird der aktuelle Aktivierungsstatus mitgeteilt:

0: Automatikfunktion deaktiviert,

(Derzeitig: Nicht markiert)

1: Automatikfunktion aktiviert,

(Derzeitig: Grün markiert)

2: Automatikfunktion im Ablauf, beim Einlaufen.

(Derzeitig: Gelb Markiert)

3: Block frei

(Darstellung: Frei)

4: Block belegt

(Darstellung: Belegt)

5: Block / Automatikfunktion Suspendiert, Wartet.

(Darstellung: Wartend)

6: Block / Automatikfunktion Reserviert, Vorbelegt.

(Darstellung: Reserviert)

254: Automatikfunktion wartet

255: Automatikfunktion Ablauffehler.

(Darstellung: Status unbekannt)

Weitere Stati können im Laufe der Entwicklung hinzukommen. Daher muss ein Endgerät unbekannte Stati und Memoryfunktionen als "Automatikfunktion Ablauffehler" mit Darstellung "Status unbekannt" interpretieren.

Die Farben zu den Block/Automatikstatus lässt sich durch den Benutzer einstellen.

Besonderheiten:

Wird nur von Graphical User Interface Prozessor verarbeitet.

Auslösung erfolgt durch Bediengerät oder weiterem Automatisierungsgerät.

Pro Auslösebefehl können mehrere Statusbefehle(Antworten) folgen, damit der entsprechende Status in den angeschlossenen Steuergeräten aktualisiert werden kann.

Sonderfall Broadcastadresse: Die Funktion wird NUR vom Master ausgeführt.

10 Abkürzungen

| oder

& und

Nummer

Adr = Adresse

Anz = Anzahl

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

CAN = Bussystem

CMD = Command

CS1 = Central Station 1 von ESU

CS2 = Central Station 2, neu

CV = Configuration-Variables.

D-Byte = Daten-Byte

DCC = Digital Command Protocoll, Gleis-Protokoll

DLC = Data Length Code, Zahl der Datenbytes im CAN-Telegramm (0-8)

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

Fkt = Funktion

FS = Fahrstufe.

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

GUI = Graphical User Interface Prozessor.

High = höherwertige bits/bytes

ID = Identifikation/Identifier

Len = Length, Länge

Loc-ID = Local Identifier. Adresse, mit welcher 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

MSB = Most Significant/ höherwertige bits/bytes

PGL = Programmiergleis

POM = Programming On the Main, Hauptgleis-Programmierung.

Prio = Priorität

Resp. = Response-Bit

S88 = Rückmelde-Hardware/Protokoll

SID = Schienen-Identifier, den mfx vergibt

SNr. = Seriennummer

Sub-CMD = Unter-Kommando

SX1 = Selectrix Gleis-Protokoll

SX2 = Weiterentwicklung von SX1

UID = Universal Identifier

V = Velocity, Geschwindigkeit