

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

WICHTIGE HINWEISE

Die Public Version der Protokoll Dokumentation enthält nur die wesentlichen Datagramme.

LAN VERBINDUNG

Ab MX10 Version 1.18.0090 stellt das MX10 auch eine Ethernet/LAN Schnittstelle zur Kommunikation zur Verfügung. Die Version 4.00 des Schnittstellen Protokolls enthält einen ersten Entwurf, wie diese zu nutzen ist.

KOMMUNIKATION ALLGEMEIN

Unbekannte Datagramme, bzw. Datagramme, welche vom MX10 nicht beantwortet werden können, muss eine PC-Software durch geeignete Timeouts abfangen.

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 1 von 62

INHALTSVERZEICHNIS

Wichtige Hinweise.....	1
LAN Verbindung	1
Kommunikation Allgemein.....	1
Inhaltsverzeichnis.....	2
Übersicht:	5
Translate Tabelle für Object's und Legacy Devices:.....	5
FILETYPES:	6
Genereller Aufbau der Telegramme	7
Grundsätzlicher ID Aufbau:.....	7
Beschreibung der Bit Felder:.....	7
Command Group's (Befehls Gruppen):.....	8
PC Interface:	9
PC USB Interface	9
Aufbau / Initialisierung der Verbindung:	9
Aufbau der Datentelegramme für das ZIMO 2.x Format für Virtual COM:	9
Ethernet/UDP Interface	10
Aufbau der Datentelegramme für das ZIMO 2.x Format per UDP:.....	10
Zusatz Befehle für PC Anbindung über LAN	11
Befehlssatz:.....	12
System Control Group [0x00].....	12
System Power [0x00.0x00].....	12
Accessory Command Group [0x01].....	14
Accessory State [0x01.0x00]	14
Accessory Mode [0x01.0x01]	15
Accessory GPIO [0x01.0x02]	17
Accessory Port4 [0x01.0x04].....	18
Accessory Data [0x01.0x05]	20
Accessory Port6 [0x01.0x06].....	21
Accessory Track MultiLimit [0x01.0x09]	24
Fahrzeug Control Group [0x02].....	25
Fahrzeug State [0x02.0x00].....	25
Fahrzeug Mode [0x02.0x01]	26
Fahrzeug Speed [0x02.0x02]	27
Fahrzeug Basis Funktionen Info [0x02.0x03]	27
Train Part List [0x05.0x01]	27
Fahrzeug Funktion Schalten [0x02.0x04]	28
Fahrzeug Aktiv [0x02.0x10].....	29
Fahrzeug Last Controller [0x02.0x12]	29
Free Group [0x03]	30
Free Group [0x04]	30

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

Train Control Group [0x05]	30
Train Part List [0x05.0x01]	30
Train Part Find [0x05.0x02]	30
Train Create [0x05.0x04]	30
Track Cfg Group [0x06/0x16]	32
TSE Track Mode [0x06/0x16.0x00]	32
TSE Info [0x06/0x16.0x02]	32
TSE Prog Clear [0x06/0x16.0x04]	33
TSE Prog Read [0x06/0x16.0x08]	33
TSE Prog Write [0x06/0x16.0x09]	34
TSE Search [0x06/0x16.0x0D]	34
TSE BiDi Raw Data, Broadcast [0x06/0x16.0x1D]	35
TSE BiDi Raw Data, Data Channel [0x06/0x16.0x1E]	35
TSE BiDi Raw Data, ACK/NACK [0x06/0x16.0x1F]	35
Data Group [0x07]	36
Group Count [0x07.0x00]	36
Item List by Index [0x07.0x01]	37
Item List by NId [0x07.0x02]	37
Hinweis zu Verwendung von 0x07.0x01/0x07.0x02:	38
Data Name [0x07.0x10]	39
Item Image Config [0x07.0x12]	39
PC ONLY: Data Name eXtended [0x07.0x21]	39
PC ONLY: Loco GUI eXtended [0x07.0x27]	39
Info / Config Group [0x08]	40
Modul Power Info [0x08.0x00]	40
Modul Info [0x08.0x08]	41
Modul Object/Property Config [0x08.0x0A]	42
Modul Tag List [0x08.0x0C]	43
PC ONLY: Modul Power Info [0x08.0x20]	44
Network Group [0x0A]	45
Ping [0x0A.0x00]	45
Port Open [0x0A.0x06]	46
LogOff / Port Close [0x0A.0x07]	46
Interface Option [0x0A.0x0A]	47
Interface Error [0x0A.0x0F]	48
Network Group [0x1A] über LAN	49
Port Open [0x1A.0x06]	49
Railway Control System (RCS) [0x0B]	50
RCS Options [0x0B.0x01]	50

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 3 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

File Control [0x0E]	50
File Transfer [0x0F].....	51
Funktionelle Eigenschaften	52
Ablauf Fahrzeug ‚aktivieren‘	52
Ablauf MX8, MX9	53
Tabellen:.....	54
Anhang:	55
Fahrzeug Status Flags.....	55
Eingetragene Markenzeichen	56
Haftungsausschluss.....	57
Glossar.....	58
Referenz Code in C# für PC Anbindung.....	59
Umwandlung von 16Bit Zahlen:.....	59
Changelog.....	60

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

ÜBERSICHT:

Das derzeit genutzte ZIMO CAN Protokoll ist mittlerweile ziemlich alt (> 10Jahre) und historisch gewachsen. Daher ist es kaum möglich, dieses Protokoll an neue Anforderungen anzupassen. Aus diesem Grunde werden die Geräte der ZS Serie (2010) parallel zum derzeitigen CAN Protokoll (ZCAN10) ein neues erweitertes Protokoll verwenden.

Das MX10 unterstützt beide Protokolle an seinen beiden CAN Buchsen.

Wobei die mit ZIMO beschrifteten CAN Buchsen defaultmäßig das ‚alte‘ Protokoll nutzen, die Fremdgeräte Buchse das neue CAN Protokoll. Dies kann aber im MX10 Menu jederzeit nach Bedarf geändert werden.

TRANSLATE TABELLE FÜR OBJECT'S UND LEGACY DEVICES:

UID Word1	UID Word2 Min.	UID Word2 Max	Verfügbare Adressen	
0x0000	0x0000	0x27FF	10240	DCC Loks
0x0000	0x2800	0x28FF	256	MM1/MM2 Loks
0x0000	0x2900	0x2EFF	3072	Frei [1]
0x0000	0x2F00	0x2FFF	256	Multitractionen
0x0000	0x3000	0x31FF	512	DCC ‚Basic‘ Zubehördecoder
0x0000	0x3200	0x39FF	2048	DCC ‚eXtended‘ Zubehördecoder
0x0000	0x3A00	0x3DFF	1024	MM1 Zubehördecoder
0x0000	0x4000	0x47FF	2048	RESERVIERT (S88 Rückmelder, X-Net Decoder, X-Net Feedback)
0x0000	0x4800	0x4FFF	2048	Frei [2]
ZIMO Gerätegeneration 1				
0x0000	0x5000	0x503F	64	MX1
0x0000	0x5040	0x507F	64	MX8 Module, Channel 1
0x0000	0x5080	0x50BF	64	MX9 Module, Channel 1
0x0000	0x50C0	0x50CF	16	CSA Module
0x0000	0x50D0	0x50DF	16	MX31
0x0000	0x5100	0x513F	64	MX8 Module, Channel 2
0x0000	0x5140	0x517F	64	MX9 Module, Channel 2
0x0000	0x5800	0x5800	128	I2C eXtender, Unterscheidung siehe SubCmd
ZIMO System Database				
	0x6000	0x60FF	256	Panels / GBS / Stellwerke
	0x6100	0x63FF	768	Routes (Fahrstraße)
	0x6400	0x65FF		
	0x6600	0x66FF	256	Soundprojekte
	0x6700	0x7FFF		RESERVIERT
mfx Adressen				
0x0000	0x8000	0xBFFF	16384	Mfx Loks
ZIMO CAN 2.xx Geräte (Auch von nicht ZIMO-Herstellern)				
	0xC000	0xC0FF	256	MX10 Zentralen
	0xC100	0xC1FF	256	MX10 Booster
	0xC200	0xC2FF	256	Spezialgeräte (IF,)
	0xC300	0xC3FF	256	Fahrpulte

ZIMO CAN Protokoll 4.20 Public.doc	Erstellt von Mike F. Schwarzer 09.04.2020 20:16:45	Seite 5 von 62
------------------------------------	---	----------------

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

	0xC400	0xC4FF	256	MX32 Funkmodule
	0xCF00	0xCFFF	256	RESERVIERT (Fahrstraßen)
	0xD000	0xDFFF	4096	Module
	0xE000	0xEFFF	4096	RESERVIERT (Objekte)
	0xF000	0xFFFF	4096	Files (siehe File Types)

FILETYPES:

File Type	Beschreibung
0x010x	GUI Sprache (0...F)
0x011x	Bezeichnung der Sprache (0...F)
01x1501	MX10 Firmware
01x1502	MX10 XILINX
0x1503	MX10 Funkprozessor
0x3001	MX32, Rev. 5 Firmware
0x3301	MX32, Rev. 7 Firmware
0x3303	MX32, Rev. 7 Funkprozessor
0x9001	StEin Firmware
0x9002	StEin XILINX
0x9004	StEin Config
0x9005	StEin Sound
0x9201	Roco Melder 10808 Firmware
0x9204	Roco Melder 10808 Config
0x9206	Roco Booster 10806/10807

HINWEIS FILE TYPES:

Bei Bedarf wird die Liste ergänzt.

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

GENERELLER AUFBAU DER TELEGRAMME

ID Command Group	Counter	Data Byte 1	Data Byte 2	Data Byte 3	Data Byte 4	Data Byte 5	Data Byte 6	Data Byte 7	Data Byte 8
Befehls Gruppe		[Ziel-ID]		[Weitere Daten ja nach Befehl]					

Die Befehlsgruppen sind so aufgebaut, dass die jeweiligen CAN Geräte diese als Filterkriterium verwenden können und somit nicht alle Nachrichten am CAN Bus auswerten müssen.

Die Verwendung der Datenbytes ist vom jeweiligen Kommando abhängig.

Soweit sinnvoll, werden Sie in folgender Reihenfolge benutzt:

1. Ziel-ID Wird verwendet, wenn ein bestimmtes Gerät angesprochen werden soll (z.B.: Eine Weiche, ein Rückmelder oder eine Lok).

2. Restliche Datenbytes

Diese werden ja nach Befehl unterschiedlich benutzt; die genaue Verwendung ist bei den einzelnen Kommandos angeführt.

GRUNDSÄTZLICHER ID AUFBAU:

Hinweis: Es sind alle 29 ID Bits in Folge dargestellt, die 'CAN' internen Flags sind nicht dargestellt.

Bit 28	Bit 27 ... 24	Bit 23 ... 18	Bit 17 .. 16	Bit 15 ... 0
1	4	6	2	16
Flag ('1')	Group	Command	Mode	Network ID

BESCHREIBUNG DER BIT FELDER:

ID Feld Aufteilung bei Anfragen, Befehlen, Events und Bestätigungen	
Flag	Immer '1', dient zur Unterscheidung anderer Protokolle
Group	4 Bit für die jeweilige Kommandogruppe. Gibt die jeweilige Command Group an (Sys, FeedBack, Loco, ...)
Cmd	Dieses 6 Bit Feld enthält das jeweilige Command
Mode	0b00: Req (Abfragen) 0b01: Cmd (Steuerbefehle, Wert setzen,) 0b10: Evt (Events = Ungefragte Informationen) 0b11: ACK (Bestätigung)
NetworkID	Identifikationsnummer des 'Absenders'. Primär notwendig um Kollisionen am Bus zu vermeiden.

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

COMMAND GROUP'S (BEFEHLS GRUPPEN):

Alle Befehle sind in folgenden Gruppen zusammengefasst:

Gruppe	Code	Verwendung/Inhalt
System	0x00	systemkritische Aufgaben (Ein/Aus, Notstopp, ...)
Zubehör	0x01	Befehle zum Steuern des Zubehörs. Damit sind sowohl Encoder/Rückmelder wie auch Decoder gemeint.
Fahrzeuge	0x02	Befehle zum Steuern der Fahrzeuge (Mobile Decoder)
Frei	0x03	Derzeit noch unbenutzt
RCS	0x04	RESERVIERT
Config	0x05	Konfiguration von Geräten, ZIMO Command Language
TrackCfg	0x06	RESERVIERT
Data	0x07	Object-Daten Transfer
Info	0x08	Statusmeldungen, meist ungefragte Meldungen
Frei	0x09	Darf von Fremdsystemen je nach Bedarf verwendet werden.
Network	0x0A	Network Management, Modulanmeldung, ...

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

PC INTERFACE:

Die Verbindung zum PC kann per Virtuellen (USB-) Com-Port oder Ethernet erfolgen. In beiden Fällen wird das hier beschriebene (CAN) Protokoll verwendet.

Bitte beachten:

- Jeder Befehl hat eine Datenlängenangabe, diese ist einzuhalten.
Die Länge bezieht sich dabei IMMER auf die Nutzdaten, nicht auf den Header/Tail oder die Befehls Identifikation (Size, Group, Cmd, Mode, ...).
- Bis auf wenige Ausnahmen entspricht das Protokoll dem intern verwendeten CAN Protokoll. Daher werden typischerweise 8 Datenbytes genutzt.
- Um die Kommunikation mit dem PC zu optimieren, gibt es einige Befehle (Gruppe 0x10 ... 0x1F), welche bis zu 256 Datenbytes übertragen können.

PC USB INTERFACE

Zwischen dem ZIMO System USB-Interface und dem PC werden die CAN Datagramme mit folgender Methode übertragen.

AUFBAU / INITIALISIERUNG DER VERBINDUNG:

Wenn der PC eine Verbindung aufbauen will, so muss er den Aufbau durch Senden der Zeichenfolge ‚Z22Z‘ (=0x5A, 0x32, 0x32, 0x5A) initialisieren werden. Erst nachdem das MX10 eine derartige Zeichenfolge ‚verstanden‘ hat, antwortet es mit einem ‚Ping‘ Telegramm. Sollte der Ping für mehr als 500mS ausbleiben, so muss der Aufbaustring wiederholt werden. Wenn auch nach dem dritten Versuch kein Ping kommt, so ist von einem Fehler auszugehen.

AUFBAU DER DATENTELEGRAMME FÜR DAS ZIMO 2.X FORMAT FÜR VIRTUAL COM:

Für das neue CAN Protokoll werden als Telegramm Delimiter die Zeichen ‚Z2‘ / ‚2Z‘ verwendet. In diesem Falle wird der CAN ID Feldweise übertragen (Group, Direction, Command und NID).

Header	Size (DLC)	Group	Cmd+Mode	NID	Data 0 ... 8	CRC16	Tail
16 Bit	8 Bit	8Bit	8Bit	16 Bit	8x8Bit	16Bit	16 Bit
0x5A32							0x325A

Grundsätzlich werden die CAN Datagramme 1:1 in dem oben definierten Frame gesendet. Da aber eine USB- (VCom) -Verbindung keine fixe Limitierung auf 8 Datenbytes hat, können auch umfangreichere Datagramme gesendet bzw. empfangen werden.

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 9 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

ETHERNET/UDP INTERFACE

Das Ethernet Interface nutzt grundsätzlich die gleiche Methode zur Daten Übertragung.

Der App-Layer Datentransfer erfolgt im Ethernet (W-LAN) über IP/UDP Frames.

Eine PC Software (bzw. App) sendet Ihre Anfragen/Befehle über der UDP Port 14520 an das MX10, die Antworten des MX10 kommen am PC, Tab, ... am Port 14521 an.

**Hinweis: Die Ports können am MX10 auch auf andere Werte gestellt werden,
bitte Anleitung MX10 beachten.**

Um die Verbindung zu initiieren muss die Anwendung ein Port ‚Open‘ ([0x0A.0x06 bzw. 0x1A.0x06]) an das MX10 senden.

AUFBAU DER DATENTELEGRAMME FÜR DAS ZIMO 2.X FORMAT PER UDP:

Für die Datagramm Übertragung im Ethernet sind keine Delimiter erforderlich (‚Z2‘ ... ‚2Z‘) da dies ja durch die Ethernet Framelogik abgedeckt ist. Wie auch bei der USB (VCom-) Schnittstelle werden im Ethernet die Daten 1:1 wie am CAN Bus übertragen. Allerdings gibt es einige zusätzliche Ethernet Datagramme, welche deutlich mehr Daten an das System übertragen können bzw. kann das MX10 auch deutlich mehr Daten in einem Datagramm an den PC senden. Diese LAN Spezialbefehle sind gesondert angeführt.

Size (DLC)	Unused	Group	Cmd+Mode	NID	Data 0 ... x
16 Bit	16 Bit	8Bit	8Bit	16 Bit	

Da ein Ethernet Frame ja typischerweise 1536 Byte umfasst, ist die Längenangabe gegenüber der VCom Schnittstelle auf 16 Bit angewachsen. Zusätzlich gibt es, ein derzeit ungenutztes, 16 Bit Feld. Dieses ist für spätere Erweiterungen vorgesehen.

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 10 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

ZUSATZ BEFEHLE FÜR PC ANBINDUNG ÜBER LAN

Um die PC Kommunikation zu optimieren, stehen einige Datagramme in einer lang-Form zur Verfügung. Damit kann ein PC mehr Informationen in einem Datagramm abfragen und senden.

Die Ausgangsbasis dieser ‚Lang‘ Datagramme sind die jeweiligen CAN Datagramme in den Gruppe 0x10 bis 0x1F.

Beispiel Modul Power Info:

Das CAN Datagramme hat die Gruppe 0x08, Command 0x00.

Die PC (lang) Form hat die Gruppe 0x18, Command 0x00.

Auch inhaltsmäßig sind diese Datagramme extrem ähnlich. Da am CAN nur 8 Datenbytes zur Verfügung stehen, werden der Zustand und die aktuellen Strom/Spannungswerte abwechselnd je Schiene übertragen. Da UDP jedoch 1536 Nutzdatenbytes haben (genutzt werden jedoch maximal 1024) können beide Schienen Zustände und die beide Strom/Spannungswerte in einem Datagramm gesendet werden (lang Form).

Ähnliches gilt auch für das Übertragen von Namen, Zugnummern, usw. letztendlich transportieren die lang Datagramme den gleichen Inhalt wie die CAN Datagramme, fassen diese jedoch sinnvoll zusammen und reduzieren damit den Trafik.

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 11 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

BEFEHLSSATZ:

SYSTEM CONTROL GROUP [0X00]

Die Command Group 0x00 fasst alle System ‚High-Priority‘ Befehle zusammen und muss von allen Boostern und Fahrpulten implementiert werden.

SYSTEM POWER [0X00.0X00]

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x00	0x00	0b00		3	SystemNID	Port						
0x00	0x00	0b01		4	SystemNID	Port	Mode					
0x00	0x00	0b1x		4	SystemNID	Port	Mode					

Mit Cmd=0x00/M=0b00 kann der Power Status des jeweiligen Gerätes abgefragt werden.

ACHTUNG:

Eine Abfrage unmittelbar nach einem Power Command kann zu inkonsistenten Antworten führen! Nach einem Power Mode Command wechselt das MX10 in den jeweils gewünschten Mode, dieser wird aber erst NACHDEM die internen Regelschleifen den Wechsel ausgeführt und durch Messungen verifiziert haben auch gemeldet.

Dieser Vorgang kann je nach gewünschtem Wechsel mehrere 100ms dauern.

Mit Cmd=0x00/M=0b01 kann der Port Power Status des Gerätes gesetzt werden, nach ‚Ausführung‘ der Status- Änderung wird der aktuelle Status per Cmd=0x00/M=0b11 ‚quittiert‘.

Der jeweils gültige Status ist auch in der regelmäßigen (ca. 500ms) Power Meldung enthalten.

Das MX10 Port wird binär kodiert, Kombinationen sind erlaubt:

Port	Ausgang
0b00000001	Schiene 1
0b00000010	Schiene 2
0b0..... 00	Schiene 3 ... 7 (Weitere MX10 im Booster Mode)
0b10000000	Booster Ausgang

Um ALLE Ausgänge mit einem Befehl zu schalten ist daher als Port 255 (=0xFF, =0b11111111) zu verwenden.

ACHTUNG:

Wenn mehrere Ausgänge gleichzeitig geschaltet werden, so erfolgt die Bestätigung trotzdem jeweils einzeln für die ‚vorhandenen‘ Ports. Wenn also z.B.: kein weiteres MX10 im Booster Mode vorhanden ist, so gibt es KEIN ACK für diese nichtexistenten Schienen!

Anwendungen (egal ob per PC Interface oder an einem der internen Bussysteme) sollten nach einem Power-Modewechsel IMMER auf das jeweilige ACK des MX10 warten, wodurch sich im Grunde eine ‚Abfrage‘ erübrigt.

Power Modes:

Mode	Zustand
0	Als Command ungültig. Wird bei Request im ACK verwendet, wenn das MX10 einen Zustandswechsel ausführt, dieser aber zum Zeitpunkt der Abfrage noch unklar/bzw. noch nicht stabil ist.
1	Der ‚Port‘ wird in Normalbetrieb geschalten

ZIMO CAN Protokoll 4.20 Public.doc	Erstellt von Mike F. Schwarzer 09.04.2020 20:16:45	Seite 12 von 62
------------------------------------	---	-----------------

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

2	Der ,Port' wird in Sammelstopp mit Fahrstufe ,0' geschalten (SSP0)
3	Der ,Port' wird in Emergency Sammelstopp geschalten (SSPe)
4	Der ,Port' wird ,AUS' geschalten
5	Der ,Port' wird in Service Mode geschalten

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

ACCESSORY COMMAND GROUP [0X01]

Die Command Group 0x01 fasst die Zubehörbefehle zusammen. Als Zubehör gilt dabei jegliches stationäres Gerät angefangen bei simplen Weichendecodern oder S88 Rückmelder bis hin zu komplexen Modulen mit RailCom/mfx Empfängern.

ACCESSORY STATE [0X01.0X00]

Jedes Steuersystem (Fahrpult, PC-Software) sollte den Zubehörstatus immer als erste Initialabfrage ausführen. Insbesondere für die MX8 und MX9 Module.

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x01	0x00	0b00		2	ZubehörNID							
0x01	0x00	0b1x		8	ZubehörNID	State/Error		Data 1		Data2		

Wenn M = 0b00, DLC = 2, dann wird der Status des Zubehörs mit 'NID' angefragt.

Wenn M = 0b11, DLC = 8, dann sendet die Zentrale die Status Antwort für das jeweilige Zubehör.

Wenn ‚State/Error‘ = 0x0000, dann befindet sich das Modul in einem ‚normalen‘ Betriebszustand. In Data1 wird die CtrlNID von jenem Gerät gesendet, welches das jeweilige Zubehör zuletzt gesteuert hat. In Data2 wird die Anzahl der mS gesendet, welche seit dem letzten Steuerbefehl vergangen sind.

Alle ‚States/Errors‘ ungleich 0x0000 sind Fehlercodes.

Hinweis:

Sollte ein Steuergerät für mehr als 65 Sec. (65536ms) keinen Steuerbefehl senden, so wird es zu einem nicht aktiven Steuergerät. In dem Falle sendet die Zentrale nur mehr die CtrlNID und als CtrlTick 0xFFFF.

Die Status Flags sind im Anhang aufgelistet.

HINWEIS, STEIN:

Die StEin Module werden in den NID Bereich 0xD000 bis 0xDFFF gemappt.

Error	Verwendung	Data 1/2
0x0000	Kein Fehler	
0x0002	Keine Gleisspannung	
0x0003	Keine Zubehör Versorgung	
0x0004	Kein DCC Signal	
0x0005	Keine CAN Spannung	
0x0006	Keine +20V	
0x0007	Keine +5V	

ACCESSORY MODE [0X01.0X01]

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
Kurzform für Schienen Decoder (DCC/MMx) bzw. einfache Module												
0x01	0x01	0b00		2	ZubehörNID							
0x01	0x01	0b1x		4	ZubehörNID	Mode						
Langform, für komplexe Module												
0x01	0x01	0b00		3	ZubehörNID	Type						
0x01	0x01	0b1x		6/8	ZubehörNID	Type	Port	Mode1	Mode2			
Langform, für komplexe Module												
0x01	0x01	0b00		4	ZubehörNID	Type	Port					
0x01	0x01	0b1x		6/8	ZubehörNID	Type	Port	Mode1	Mode2			

Dieses Datagramm dient der Abfrage und der Einstellung der Zubehörbetriebsart.

HINWEIS, DCC BASIC DECODER:

Für DCC bzw. MMx Decoder gilt die Kurzform des Datagrammes.

DCC Basic Decoder haben eine Zubehör NID im Bereich 0x3000 ... 0x31FF (Adresse 1 ... 512).

Die Standard DCC Zubehör Decoder kennen 2 Betriebsarten:

Mode ,0': Default Mode (bzw. Betriebsart unbekannt)

Mode ,1': Paarbtrieb (Typischerweise Weichendecoder)

Mode ,2': Einzelbetrieb (Jeder Ausgang kann getrennt geschalten werden).

Wenn das MX10 Einschalten wird, befinden sich alle Decoder im ,Default' Mode (,0'), typischerweise arbeiten DCC Decoder dann im Paar (Weichen) Modus.

Wenn eine bestimmte Betriebsart gewünscht ist, so muss diese zuvor durch diesen Befehl für den jeweiligen Decoder festgelegt werden. Diese Festlegung wird im MX10 gespeichert und gilt bis diese geändert wird.

HINWEIS, MX8 MODULE:

Für die MX8 Module gilt die Kurzform des Datagrammes.

MX8 Module haben eine Zubehör NID im Bereich 0x5040 ... 0x507F (Adresse 0 ... 63).

Die MX8 Module kennen folgende Betriebsarten:

Mode ,0': Default Mode (bzw. Betriebsart unbekannt)

Mode ,1': Beide Ausgangsgruppen im Paar Betrieb (Paar/Paar)

Mode ,2': **Ausgangsgruppe 1: Einzelbetrieb, Ausgangsgruppe 2: Paarbtrieb oder
Ausgangsgruppe 1: Paarbtrieb, Ausgangsgruppe 2: Einzelbetrieb
Mode 2 wird auch gemischter Betrieb genannt, NICHT IMPLEMENTIERT!**

Mode ,3': Beide Ausgangsgruppen im Einzelbetrieb (Einzel/Einzel)

HINWEIS, MX9 MODULE:

Für die MX9 Module gilt die Kurzform des Datagrammes.

MX9 Module haben eine Zubehör NID im Bereich 0x5080 ... 0x50BF (Adresse 0 ... 63).

Die MX8 Module kennen keine besonderen Betriebsarten:

Mode ,0': Default Mode (bzw. Betriebsart unbekannt)

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

Mode ,1': MX9 Modul vorhanden

HINWEIS, MX10 ZENTRALE:

Für das MX10 gilt die jeweilige MX10 NId.

Type	Verwendung	Mode1	Mode2
0x00			
0x10	ABA Ausgänge	0x0000 = unbekannt 0xnnn1 = Ausgang kann ,offen' sein 0xnnn2 = Ausgang kann ,GND' schalten 0xnnn4 = Ausgang kann ,+5V' schalten	

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

ACCESSORY GPIO [0X01.0X02]

ACHTUNG:

Diese Datagramme sind nur als ‚Request‘ erlaubt und haben NUR den Zweck einer schnellen Informationsbeschaffung.

ECHTE Schaltvorgänge MÜSSEN über die Port Datagramme abgewickelt werden.

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x01	0x02	0b00		4	ZubehörNID	Type						
0x01	0x02	0b1x		8	ZubehörNID	Type	GPIO States					

Diese Datagramme dienen der effizienten Statusabfrage von simplen Ein-/Ausgängen. Es werden je Gruppe bis zu 32 Ein-/Ausgangszustände übertragen.

Durch M=0b00 kann vom Gerät ‚ZubehörNID‘ der GPIO Bereich ‚Type‘ abgefragt werden.

Diese werden mit M=0b11 (ACK) beantwortet. Sollte irrtümlicherweise ein Command (0b01) an die Zentrale gesendet werden, so wird die mit einem ‚Command Error‘ beantwortet.

HINWEIS, DCC BASIC DECODER:

Die DCC Basic Decoder sind in den NID Bereich 0x3000 bis 0x31FF gemappt.

Diese Datagramme werden NUR für DCC Decoder im Einzel Betrieb (Mode=2) unterstützt. Diese Betriebsart MUSS VOR Verwendung entsprechend gesetzt werden.

HINWEIS, MX8:

Die MX8 Module werden in den NID Bereich 0x5040 bis 0x507F gemappt.

Type=‘0‘: MX 8 in Betriebsart unbekannt → FEHLER

Type=‘1‘: MX 8 in Paar/Paar Mode

ACHTUNG:

Ein Request wird mit einem Acknowledge aus dem internen MX10 Speicher beantwortet!!

Es kann daher bei fehlerhaften Weichen bzw. MX8 Modulen zu falschen Antworten kommen.

Eine PC Software muss daher die Differenzen zwischen ACK und EVT berücksichtigen und entsprechende Maßnahmen setzen. Z.B.: Befehl wiederholen, Anwender Informieren, ...

Je nach verwendet MX8 und/oder Weichenantrieb sind Fehlmeldungen (Fehlerhafte ACK's) mehr oder minder wahrscheinlich. Bei Motor, Servo Antrieben stimmen die Stellungen typischerweise immer, bei ‚Klick-Klack‘ Antrieben ist dies extrem vom jeweiligen Antrieb abhängig.

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 17 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

HINWEIS, MX9:

Die MX9 Module werden in den NID Bereich 0x5080 bis 0x50BF gemappt.

Type	Verwendung	Value
0x0000	Gleisabschnitt 1 ... 16, Besetzmeldungen Da ein MX9 über 16 Abschnitte verfügt, sind nur die ersten 16 Bits benutzt.	,0' = Frei, ,1' = Besetzt
0x0002	MX9 Signal Ausgänge, sofern das jeweilige MX9 mit ALA Platinen bestückt ist.	,0' = AUS, ,1' = EIN

Sofern die Daten des MX9 zum Zeitpunkt der Abfrage ,unklar' sind, wird die Abfrage mit einer Accessory Error Meldung (Grp=0x01, Cmd=0x00) beantwortet. Da das MX9 die Zustände der ALA nicht speichert und auch beim MX9 nicht abgefragt werden können, erhält man nur den gültigen Status von zuvor gesetzten MX9ALA Zuständen, weil nur das MX10 diese speichert.

ACCESSORY PORT4 [0X01.0X04]

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x01	0x04	0b00		3	ZubehörNID	Port						
0x01	0x04	0b0x		4	ZubehörNID	Port	Value					
0x01	0x04	0b1x		4	ZubehörNID	Port	Value					

Wenn M = 0b00, DLC = 3, dann wird der Zustand des Ein/Ausganges (Port) vom Zubehör NID abgefragt.

Durch M = 0b01, DLC = 4 wird der Ausgang des Zubehörs (NID) auf den angegebenen Wert eingestellt. Die Abfrage wird durch 0b1x beantwortet, ebenso Änderungen welche durch andere Einflüsse verursacht werden (z.B.: Manuelles Verstellen, Zeitablauf, ... oder andere events)

Jedes Zubehör (Egal ob 'Schienen' gebunden oder am Bus System) darf bis zu 128 Ein/Ausgänge haben. Jeder Ausgang darf bis zu 256 'Stellungen' haben. Wie viele dieser Möglichkeiten genutzt sind, ist vom jeweiligen Modul abhängig.

Bit 7 der Port Nummer gibt an ob der Port Wert gültig (Valid) oder nur ,virtuell' (gespeichert) ist.

HINWEIS, DCC BASIC DECODER:

DCC Basic Decoder haben eine Zubehör NID im Bereich 0x3000 ... 0x31FF (Adresse 1 ... 512).

Die Ports werden von 0 ... 7 gezählt, bei Weichendecodern sind nur die geraden Port Nummern gültig (0 = Weiche 1; 2 = Weiche 2; 4 = Weiche 3; 6 = Weiche 4;).

Ein Value von ,0' bedeutet, dass der jeweilige Ausgang (Port) abgeschaltet sein soll, ein Value von ,1', das dieser eingeschalten sein soll.

ACHTUNG:

Die tatsächliche Funktion bei DCC Decodern ist extrem vom jeweiligen Decoder und dessen Konfiguration abhängig. Im Grunde bewirkt dieser Befehl nur, dass die Zentrale einen DCC Befehl gemäß NMRA Norm ,Basic Accessory Decoder Packet Format' ans Gleis sendet.

Folgende Bitzuordnung wird dabei verwendet:

NMRA Befehl: 10AAAAA 0 1AAACDDD (siehe NMRA Norm 9.2.1:

http://www.nmra.org/sites/default/files/s-9.2.1_2012_07.pdf)

A = 9 Bit Adresse des Decoders

D = Port Nummer

C = Port Zustand

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 18 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

HINWEIS, MX8:

Die bekannten MX8 Module werden entsprechend Ihrer Adressen in den Zubehör NID Bereich gemappt.

Die MX8 Module belegen dabei den Bereich 0x5040 bis 0x507F (Max. 64).

Die MX8 Module haben 32 Ausgänge, welche je nach MX8 Konfiguration getrennt oder paarweise angesteuert werden können.

Im Paar Betrieb im Paar Betrieb gelten für die Ansteuerung jeweils die geraden Port Nummern (0, 2, 4, 6, ...).

Die Antwort über die Schnittstelle kommt mit der gleichen Port Nummer (0, 2, 4, ...) und jeweils einem Bit für die beiden Ausgänge. Dabei bedeutet 0b00 und 0b11 eine fehlerhafte Stellung, 0b01 bzw. 0b10 die jeweils gültige Stellung.

HINWEIS, MX9:

Die MX9 Module werden in den NID Bereich 0x5080 bis 0x50BF gemappt.

Die Port Nummer wird für die verschiedenen MX9 Funktionen wie folgt genutzt:

Port	Verwendung
0 ... 15	Gleisabschnitt 1 ... 16, Besetzmeldungen
32 ... 63	ALA Ausgänge
128 ... 143	HLU Geschwindigkeit. Wobei die HLU Geschwindigkeit immer für Hauptabschnitt und Folgeabschnitt gemeinsam gilt. Bit 0 ... 5 → HLU Speed.

Sofern die Daten des MX9 zum Zeitpunkt der Abfrage ‚unklar‘ sind, wird die Abfrage mit einer Accessory Error Meldung (Grp=0x01, Cmd=0x00) beantwortet.

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

ACCESSORY DATA [0X01.0X05]

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x01	0x05	0b00		4	ZubehörNID	Port	Type					
0x01	0x05	0b01		8	ZubehörNID	Port	Type	Data	Data	Data	Data	
0x01	0x05	0b11		8	ZubehörNID	Port	Type	Data	Data	Data	Data	

Mit diesen Datagrammen können Objektdaten abgefragt und gesetzt werden.

In einigen Fällen sind die Objektdaten ‚read only‘, z.B. Zugnummern können immer nur abgefragt werden bzw. werden bei Änderung als ‚Event‘ gemeldet.

HINWEIS, MX9:

Die MX9 Module werden in den NID Bereich 0x5080 bis 0x50BF gemappt. Die Port Nummer wird fortlaufend von ‚0‘ bis ‚15‘ gezählt. Der ‚alte‘ Hauptabschnitt 1 hat daher die Port Nummer ‚0‘ und ‚1‘, usw.

Type	Verwendung	Data (DB5 ... DB8)
0x11	MX9 Fahrzeug 1, 2	Fahrzeugadresse 1 in DB5, DB6 Fahrzeugadresse 2 in DB7, DB8
0x12	MX9 Fahrzeug 3, 4	Fahrzeugadresse 3 in DB5, DB6 Fahrzeugadresse 4 in DB7, DB8

Hinweis zu den MX9 Zugnummern:

Siehe Bedienungsanleitung MX9

HINWEIS, STEIN:

Die StEin Module werden in den NID Bereich 0xD000 bis 0xDFFF gemappt.

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 20 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

ACCESSORY PORT6 [0X01.0X06]

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x01	0x06	0b00		4	ZubehörNID	Port	Type					
0x01	0x06	0b1x		6	ZubehörNID	Port	Type	Value				

Wenn M = 0b00, DLC = 4 dann wird der Zustand des Ein-/Ausganges (Port) vom Zubehör NID abgefragt.

Durch M = 0b01, DLC = 6 wird der Ausgang des Zubehörs (NID) auf den angegeben Wert eingestellt. Die Abfrage wird durch 0b1x beantwortet, ebenso Änderungen welche durch andere Einflüsse verursacht werden (z.B.: Manuelles Verstellen, Zeitablauf, ...)

HINWEIS, MX10:

Die ABA Ein-/Ausgänge des MX10 sind über die Nid des MX10 ansprechbar.
Die Port Nummer gibt den Ein- bzw. Ausgang an.

Port	Type	Verwendung
0x00 ... 0x07	0x20	ABA Eingänge Value enthält den jeweiligen Analog Wert des Eingangs
0x00 ... 0x06	0x21	ABA Ausgänge Value ,0x00' → Ausgang ,offen', ,0x10' → Ausgang ,Low', ,0x11' → Ausgang ,High'

HINWEIS, STEIN:

Port	Type	Verwendung																
0x00	0	Spezial Fälle																
0x01	0	StEin Display, Nur COMMAND, Antwort ACK DB5 = Digit 1, DB6 = Digit 2.																
0x00 ... 0x07	0x01	Block, Status: Value = 0x0000 → Frei, keine Spannung Value = 0x0100 → Frei, mit Spannung Value = 0x1000 → Besetzt, keine Spannung Value = 0x1100 → Besetzt, mit Spannung Value = 0x1201 → Besetzt, Überlast 1 Value = 0x1202 → Besetzt, Überlast 2 Value = 0x1203 → Besetzt, Überlast 3																
0x00 ... 0x07	0x02	HLU, Value 0 ... 15 → HLU Speed <table border="1" style="margin: 5px auto; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 12.5%;">15</td> <td style="width: 12.5%;">14</td> <td style="width: 12.5%;">12 ... 13</td> <td style="width: 12.5%;">8 ... 11</td> <td style="width: 12.5%;">7</td> <td style="width: 12.5%;">6</td> <td style="width: 12.5%;">4 ... 5</td> <td style="width: 12.5%;">0 ... 3</td> </tr> <tr> <td>A2</td> <td>Frei</td> <td>Dir 2</td> <td>HLU 2</td> <td>A1</td> <td>Frei</td> <td>Dir 1</td> <td>HLU 1</td> </tr> </table> HLU 1/2: 0x0 → AUS, 0x2 → Halt, 0x4 → UH, 0x6 → U, 0x8 → LU, 0xA → L, 0xC → FL, 0xE → Fahrt Dir 1/2: 0x00 → KEIN Richtungsbit, 0x01 → Vorwärts, 0x02 → Rückwärts A (Aktiv): Bei Command wird IMMER Dir1/HLU1 ans Gleis gesendet, A1 ist daher im Command ACK immer ‚1‘ und A2 immer ‚0‘. Bei Abfrage bzw. Event ist je nach aktuellem Status A1 oder A2 = ‚1‘.	15	14	12 ... 13	8 ... 11	7	6	4 ... 5	0 ... 3	A2	Frei	Dir 2	HLU 2	A1	Frei	Dir 1	HLU 1
15	14	12 ... 13	8 ... 11	7	6	4 ... 5	0 ... 3											
A2	Frei	Dir 2	HLU 2	A1	Frei	Dir 1	HLU 1											
0x00 ... 0x07	0x03	HLU Funktion																
0x00 ... 0x07	0x08	Aktueller Anschluss Strom, Value = Strom in mA																
0x00 ... 0x0F	0x10	Output Ports: Value = 0x0000 → Unbekannt Value = 0x0001 → AUS/OFFEN Value = 0x0002 → Masse Value = 0x0004 → VCC Value = 0x10nn → Mehrfach Stellungen 1 bis Anzahl Mode Command																
0x00 ... 0x0F	0x20	Input Ports: Value = 0x0000 → Unbekannt (Z.B. Wechsellspannung, Pulse, ...) Value = 0x0001 → AUS/OFFEN Value = 0x0002 → Masse Value = 0x0004 → Positiv (> 3V3)																
0x80 ...	0xFF ...	I2C Adresse: 0 ... 127																

StEin Anzeige Logik.

Mit Port 0x01, Type 0x00 können die beiden Anzeige Digits vom StEin für ‚externe‘ Anzeigen verwendet werden. Jede Digitbyte ist dazu in 3 High Bits und 5 Low Bits unterteilt.

Die 5 Low Bits ergeben das gewünschte Zeichen: 0 = ‚0‘, 1 = ‚1‘, 9 = ‚9‘, 10 = ‚A‘, 11 = ‚b‘, 12 = ‚C‘, 13 = ‚d‘, 14 = ‚E‘, 15 = ‚F‘, 16 = ‚H‘, 17 = ‚h‘, 18 = ‚I‘, 19 = ‚L‘, 20 = ‚n‘, 21 = ‚P‘, 22 = ‚S‘, 23 = ‚U‘, 24 = ‚u‘, 25 = ‚_‘, 26 = ‚-‘, 27 = ‚‘,

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

Wenn Bit ,7' bei einem der beiden Digit Bytes gesetzt ist, so wird die ,remote' Anzeigefunktion beendet (Gilt immer für beide Digits!!!), wenn Bit ,6' gesetzt ist, blinkt das jeweilige Digit, Bit ,5' zeigt den Dezimalpunkt an.

ACHTUNG!!!
NOCH SIND NICHT ALLE KOMBINATIONEN GEPRÜFT!!

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 23 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

ACCESSORY TRACK MULTILIMIT [0X01.0X09]

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x01	0x09	0b00		3	AccessoryNId		Track					
0x01	0x09	0b01		5	AccessoryNId		Track	HLU-Code	O/W-Code			
0x01	0x09	0b1x		5	AccessoryNId		Track	HLU-Code	O/W-Code			

[0x01.0x09] Eingeführt mit Softwareversion 01.26.0001

DLC sollte klarerweise 2 bzw. > 4 sein. Somit stehen 4 ‚Nutzdatenbytes‘ zur Verfügung.

Diese Datagramme arbeiten ähnlich dem Accessory PIN6 Datagramm[0x01; 0x06], jedoch können bis zu 2 Folge HLU Stufen (Zusätzlich zur ‚Basis‘ HLU Stufe) und der Ost/West Code an einem Gleisabschnitt gesendet werden.

Die HLU Codes sind wie folgt (maskiert auf den unteren 4 Bits)

0 = A = HLU AUS

2 = H = HLU Halt

4 = UH = Zwischenstufe

6 = U = HLU Ultralangsam

8 = LU = Zwischenstufe

10 = L = HLU Langsam

12 = FL = Zwischenstufe

14 = F = HLU Freie Fahrt

Bei ungeraden Werten wird abgerundet.

Per Datagramm mit Mode ‚0b00‘ kann der aktuelle HLU und O/W Code für den jeweiligen Gleisabschnitt (Track) abgefragt werden.

Per Mode ‚0b01‘ wird ein neuer HLU und O/W Code an das Modul (StEin) gesendet. Dieser wird per Ack (0b10) bestätigt, ebenso Abfragen.

Sobald der StEin im ‚Betrieb‘ autonom (eben aufgrund der Balisen Meldungen) die HLU Stufen wechselt, erfolgt die Meldung per Event (0b11).

Hinweis:

Wenn der StEin per Multilimit Datagramm [0x01.0x09] gesteuert wird, so:

- Erfolgen die Rückmeldungen ebenfalls durch dieses Datagramm
- Balisen Meldungen sind in dem Falle NICHT mehr vorhanden, da ja intern schon abgearbeitet.

Das MX10 prüft ab Version 01.26.0001 bei KEINEM StEin Datagramm mehr ob der DLC der CAN2.0-Doku entspricht oder nicht. Nur mehr, das der DLC kleiner/gleich 8, weil dies schlicht das CAN Limit ist.

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 24 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

FAHRZEUG CONTROL GROUP [0X02]

Die Command Group 0x02 fasst alle Fahrzeugsteuerbefehle zusammen und muss von allen Boostern und Fahrpulten implementiert werden. Diese Gruppe enthält jedoch KEINE Programmierbefehle.

FAHRZEUG STATE [0X02.0X00]

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x02	0x00	0b00		2	FahrzeugNID							
0x02	0x00	0b11		8	FahrzeugNID		StateFlags		CtrlNID		CtrlTick	

Mit diesen Datagrammen kann ein Gerät (Fahrpult, PC-Software, ...) den aktuellen Status eines Fahrzeuges abfragen. Dies ist insbesondere sinnvoll um Steuer-Konflikte zu erkennen.
 Wenn M = 0b00, DLC = 2, dann wird der Status des Fahrzeuges mit 'NID' angefragt.
 Wenn M = 0b11, DLC = 8, dann sendet die Zentrale die Status Antwort für das jeweilige Fahrzeug.
 In der Antwort gibt die CtrlNID an welches Gerät das jeweilige Fahrzeug zuletzt gesteuert hat, der Wert CtrlTick enthält dabei die Anzahl der vergangenen ms seit dem letzten Steuerbefehl des CtrlNID Gerätes.

Hinweis:

Sollte ein Steuergerät für mehr als 65 Sec. (65536ms) keinen Steuerbefehl senden, so wird es zu einem nicht aktiven Steuergerät. In dem Falle sendet die Zentrale nur mehr die CtrlNID und als CtrlTick 0xFFFF.

Die Status Flags sind im Anhang aufgelistet.

Hinweis:

Bei einem dem MX10 unbekanntem Fahrzeug kommt keine Meldung.

STATE FLAGS

Bit	Info Command
0 .. 7	Consist Nid & 0xFF
8	BiDi
9	ZACK
10	Richtungs Anweisung (Client an MX10)
11	Aktuelle Gleisrichtung (MX10 an Client)
12	
13	
14	
15	E-Stopp

FAHRZEUG MODE [0X02.0X01]

Diese Datagramme dienen der Abfrage bzw. dem Einstellen der Fahrzeug-Betriebsart. Damit das MX10 ein Fahrzeug steuern kann, müssen ihm die Betriebsparameter bekannt sein.

Eine PC-Software kann/muss die Fahrzeug Betriebs-Parameter festlegen!!!

Dazu kann sie entweder diese zuerst abfragen und nur ‚unbekannte‘ Parameter ergänzen, oder schlicht und einfach ‚Ihre‘ Parameter rücksichtslos als Command senden.

In jedem Falle arbeitet das MX10 mit den zuletzt definierten (empfangenen) Fahrzeugparametern und speichert diese bei Betriebsende.

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x02	0x01	0b00		2	FahrzeugNID							
0x02	0x01	0b01		5	FahrzeugNID	M1	M2	M3				
0x02	0x01	0b11		5	FahrzeugNID	M1	M2	M3				

Durch Mode=0b00 kann das Steuergerät die aktuellen Betriebsparameter für ein Fahrzeug abfragen. Bei einer Antwort mit M1=0x00 und M2=0x00 ist dem MX10 das jeweilige Fahrzeug unbekannt.

MODE 1 FLAGS

Bit	Info Command
0 .. 3	Speed Steps: 0: ‚unbekannt‘ 1: 14FS 2: 27FS 3: 28FS 4: 128FS 5: 1024FS 6 - 7: nicht definiert
4 .. 7	Schienen Format: 0: unbekannt 1: DCC 2: MM2 3: nicht definiert 4:mfx

MODE 2 FLAGS

Bit	Info Command
0 .. 7	Max. Anzahl an Funktionen: Keine (0) bis derzeit max. 32

MODE 3 FLAGS

Bit	Info Command
0	Puls Fx (Funktionen werden also Pulschette gesendet, LGB)
1	Analog Fx (Analog Funktionen)
2 .. 3	Speed Limit ZIMO / Speed Limit NMRA 0b00 = Kein Speed Limit aussenden, 0b01 = NMRA Speed Limit senden 0b10 = ZIMO Speed Limit senden

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

4	nicht definiert
5	nicht definiert
6	nicht definiert
7	nicht definiert

FAHRZEUG SPEED [0X02.0X02]

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x02	0x02	0b00		2	FahrzeugNID							
0x02	0x02	0b01		6	FahrzeugNID		Speed		Div	0		
0x02	0x02	0b11		6	FahrzeugNID		Speed		Div	0		

Wenn M = 0b00, DLC = 2, dann wird die Geschwindigkeit der Lok mit 'NID' angefragt.

Wenn M = 0b01, DLC = 6, dann wird die Geschwindigkeit der Lok mit 'NID' auf den übergeben Wert gesetzt.

Wenn M = 0b11, DLC = 6, dann Antwortet die Lok mit 'NID' auf die Abfrage nach Ihrer Geschwindigkeit.

Geschwindigkeiten werden immer mit 10Bit gesendet, in den obersten 6Bit werden zusätzliche Flags gesendet (siehe Anhang Fahrzeug Status Flags). Die jeweilige Umrechnung ins Schienenformat erfolgt im TSE.

Der ‚Div‘ (Byte 5) Wert gibt einen Rangierdivisor an (z.B.: /2 oder /4) für eine feinere Rangierauflösung.

FAHRZEUG BASIS FUNKTIONEN INFO [0X02.0X03]

TRAIN PART LIST [0X05.0X01]

Grp	Cmd	D	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x05	0x01	0b00		3	ZugNid		Idx					
0x05	0x01	0b11		7	ZugNid		DecoderNid		OwnerNid	Flag	Idx	

Mit diesem Datagramm können die ‚Teile‘ eines Zuges abgefragt werden.

Die Antwort enthält die ‚DecoderNid‘ (Fahrzeug/Funktionsdecoder Adresse) und wer den jeweiligen Zug angelegt hat.

Wenn M = 0b00, DLC = 2, dann wird der Funktionsstatus der Lok mit 'NID' abgefragt.

Wenn M = 0b11, DLC = 6, dann antwortet die Lok auf eine Statusabfrage.

Dieses Datagramm dient nur der ‚schnellen‘ Abfrage des aktuellen Fahrzeug Funktionszustandes (ähnlich dem Accessory Datagrammen ‚GPIO‘).

Ein ‚Cmd‘ (Befehl) ist NICHT vorgesehen!!

Die 32 Bits enthalten in den Datenbytes 3 bis 6 den jeweiligen Status der Lok-Funktionen 0 bis 31. Wobei das höchste Bit im DB3 die Funktion 0 enthält und das niedrigste Bit in DB6 die Funktion 31 darstellt.

Die eigentlichen Schaltbefehle und Bestätigungen erfolgen über die ‚Fahrzeug Funktion schalten‘ Datagramme.

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 27 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

FAHRZEUG FUNKTION SCHALTEN [0X02.0X04]

Grp	Cmd	D	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x02	0x04	0b00		4	FahrzeugNID		FxNr					
0x02	0x04	0b01		6	FahrzeugNID		FxNr		FxVal			
0x02	0x04	0b11		6	FahrzeugNID		FxNr		FxVal			

Wenn M = 0b00, DLC = 4, dann wird die Funktion der Lok mit 'NID' und der Funktion 'Nr.' abgefragt.
 Wenn M = 0b01, DLC = 6, dann wird die Lokfunktion 'FxNr' der Lok 'NID' auf den angegebenen Wert gesetzt.

Wenn M = 0b11, DLC = 6, dann antwortet die Lok auf eine Funktionswert Abfrage.

Wobei FxVal = 0x00 immer 'Aus' bedeutet, FxVal ungleich 0x00 sind vom jeweiligen Lok-Decoder abhängig, für 'normale' DCC und MM Lok Decoder werden diese als Funktion 'Ein' interpretiert.

Die ‚FxNr‘ ist für diesen Befehl in mehrere Bereiche aufgeteilt:

Von FxNr	Bis FxNr	Beschreibung	Gültige Werte
0	31	Die bekannten ‚normalen‘ Funktionen, die maximal Fx Nummer ist dabei vom jeweiligen Format abhängig.	Ein/Aus
254	254	Rangierfunktion	0 ... 255
255	255	MAN Funktion Ein/Aus	0/1

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

FAHRZEUG AKTIV [0X02.0X10]

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x02	0x10	0b00		2	Ziel-NID							
0x02	0x10	0b01		4	Ziel-NID	Mode						

Jedes Steuergerät (Fahrpult/Computer) sollte diesen Befehl für ‚aktive‘ Fahrzeug etwa alle 500 ... 1000ms aussenden. Dieses Kommando bewirkt, dass dieses Fahrzeug zumindest in Priorität 1 im MX10 verbleibt.

Wenn ein Steuergerät ein Fahrzeug aktiv steuern will, so muss es zuerst abfragen, ob das Fahrzeug nicht schon von einem anderen Gerät gesteuert wird. Wenn die Abfrage NICHT innerhalb von 500ms beantwortet wird, so wird das Fahrzeug von keinem anderen Gerät gesteuert und kann aktiviert werden.

Wenn die Abfrage beantwortet wird (Mode = 1), so ist das Fahrzeug auf einem anderen Gerät aktiv. Dies ist an sich eine reine Absicherung, da jedes Steuergerät sowieso periodisch für die von ihm gesteuerten Fahrzeuge eine ‚Aktiv‘ Meldungen senden muss.

Wenn ein Steuergerät ein ‚aktives‘ Fahrzeug übernehmen will, so muss es das ‚Übernahme‘ Kommando senden (Mode = 0x10).

Jede ‚Fahrzeug Aktiv‘ Meldung wird vom MX10 mit dem Status beantwortet.

ACHTUNG!!! UNTERSCHIEDUNG STELLWERK/FAHRPULT

Dieser Befehl ist die WESENTLICHSTE Unterscheidung zwischen einer Stellwerks- und einer Fahrpult-Anwendung (egal ob am PC oder Tab, ...).

Eine Fahrpultanwendung MUSS die Übernahme/Übergabe Prozedur implementieren, da sonst andere Fahrpulte kommentarlos gegensteuern können.

Eine Stellwerksoftware muss Fahrzeuge nicht zwangsweise ‚aktiv‘ melden, sofern sie mit manuellen Eingriffen umgehen kann.

Unter Anwendung der Übergabe/Übernahme Technik, kann ein Fahrpult immer nur jenes Fahrzeug steuern, für welches es den ‚aktiv Focus‘ hat.

Ohne der Übergabe/Übernahme Logik, inkl. der aktiv Meldung, kann jederzeit ein anderes Steuergerät Fahrstufen und/oder Funktionen ändern. Es ist dann Aufgabe der jeweiligen Software mit solchen Änderungen umzugehen. Abweichungen zwischen eigenen ‚SOLL‘ Zustand und gemeldeten ‚IST‘ Zustand müssen entsprechend abgebildet werden bzw. im weiteren Ablauf der Software berücksichtigt werden.

FAHRZEUG LAST CONTROLLER [0X02.0X12]

Reserviert.

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 29 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

FREE GROUP [0X03]

Derzeit frei für Protokoll Erweiterungen

FREE GROUP [0X04]

Derzeit frei für Protokoll Erweiterungen

TRAIN CONTROL GROUP [0X05]

Diese Datagramm Gruppe dient der Verwaltung von Zügen (Verbund Objekten). So wie bei allen anderen Gruppen kann die Anzahl der im System gespeicherten Züge über den Group Count (Data Group Datagramm [0x07.00]) abgefragt werden. Ebenso können die NId's der vorhandenen Züge mit den Datagrammen Item List bei Index/NId abgefragt werden.

Züge bestehen aus zumindest 2 Decodern (typischerweise Fahrzeug Decoder, Funktionsdecoder sind jedoch ebenso möglich). Jeder Zug hat einen ‚Eigentümer‘, dies kann ein bestimmtes MX32 sein, ein PC oder eine andere Steuereinheit, welche Zugriff auf Fahrzeuge hat.

Das MX10 kann bis zu 256 solcher Züge verwalten und speichern. Jeder Zug darf im Grunde aus allen im MX10 vorhandenen Fahrzeug/Funktionsdecodern bestehen. Sinnvoll sind jedoch nur Züge mit bis zu maximal 16 FAHR Decodern, da sonst schon alleine die im Gleissignal (DCC) vorhandene (besser nicht vorhandene) Bandbreite zu einem sichtbaren ruckeln der Fahrzeuge führt.

TRAIN PART LIST [0X05.0X01]

Grp	Cmd	D	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x05	0x01	0b00		3	ZugNId		Idx					
0x05	0x01	0b11		8	ZugNId	DecoderNId	OwnerNId	Flag	Idx			

Mit diesem Datagramm können die ‚Teile‘ eines Zuges abgefragt werden. Die Antwort enthält die ‚DecoderNId‘ (Fahrzeug/Funktionsdecoder Adresse) und wer den jeweiligen Zug angelegt hat.

TRAIN PART FIND [0X05.0X02]

Grp	Cmd	D	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x05	0x02	0b00		2	DecoderNId							
0x05	0x02	0b11		8	DecoderNId	ZugNId	OwnerNId	State				

Mit diesen Datagrammen kann abgefragt werden, ob ein Fahrzeug zu einem Zug (Traktion) gehört. Wenn das Fahrzeug zu keinem Zug gehört, so ist im ACK die ZugNId = 0xFFFF. Unter Decoder sind sowohl Fahrzeug Decoder als auch reine Funktionsdecoder zu verstehen. Die jeweilige NId ist entsprechend der schon definierten Regeln anzugeben.

TRAIN CREATE [0X05.0X04]

Grp	Cmd	D	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x05	0x05	0b10		2	ZugNum							
0x05	0x05	0b11		6	ZugNum	OwnerNId	ZugNId					

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 30 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

Durch diese Datagramme kann ein ‚neuer‘ Zug angelegt werden.

Damit kann ein Bediengerät einen neuen Zug anlegen.

Die ZugNum ist dabei jene Nummer, welche das Endgerät nutzen will, die ZugNId ist Systemweit eindeutig.

Hinweis: Es sind nur ‚CMD‘ und ‚ACK‘ erlaubt.

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 31 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

TRACK CFG GROUP [0X06/0X16]

In dieser Gruppe sind alle Schienensignal-Funktionen zusammengefasst. Insbesondere sind dies die verschiedenen Methoden um Decoder zu programmieren, Firmwareupdates durchzuführen und Soundprojekte zu laden. Die Gruppen Kennung 0x06 wird CAN Bus intern genutzt, 0x16 hingegen ist für die Verwendung per Ethernet Schnittstelle vorgesehen. Der Hauptunterschied besteht schlicht darin, dass am Ethernet kein 8 Byte Limit gibt, wie es am CAN notwendig ist. Dadurch kann eine geeignete PC Software komplexere Befehle nutzen.

Diese Gruppe dient der (DCC) Programmierung von Decodern. Eine Anwendung sollte nur ca. alle 200mS einen POM Read/Write Befehle senden. Dies dient dazu, das auch andere Befehle ans Gleis gesendet werden können (z.B.: Fahrbefehle von MX32, RocoApp, ...). Alternativ kann eine Anwendung die Info Datagramme auswerten und sich dynamisch an die Schienen Bandbreite anpassen.

TSE TRACK MODE [0X06/0X16.0X00]

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x06	0x00	0b00		3	NID		Port					
0x06	0x00	0b01		4	NID		Port	Mode				
0x06	0x00	0b1x		8	NID		Port	Mode	Voltage		Current	

Durch M=0b00 kann der aktuelle Status der Track Signal Engine abgefragt werden. Der Port gibt an welcher MX10 Port in den gewünschten Mode geschaltet werden soll.

ACHTUNG:

Nicht jeder Port kann alle Betriebsmodi ausführen.
Durch Mode wird der jeweilige Betriebsmode festgelegt.

Im Acknowledgement ist die jeweilige Spannung (in mV) und der Maximal Strom (in mA) enthalten.

Hinweis:

Genaue Beschreibung der Betriebsmodi fehlt noch.

TSE INFO [0X06/0X16.0X02]

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5 .. DB8	DB9	DB10
0x06	0x02	0b1x		10	SysNId		FahrzeugNID		Cfg Num	State	Code

Durch diese Datagramme meldet die Zentrale (MX10) die diversen Programmierzustände. Während dem Lesen bzw. Schreiben enthält ‚State‘ den aktuellen Status und Count einen Zähler für den jeweiligen Status.

Wenn die jeweilige CV gelesen bzw. geschrieben ist, wird dies entweder durch das jeweilige ACK Datagramm gemeldet, bzw. bei einem Nichterfolg durch ein ‚Info ACK‘ mit Angabe des Fehlercodes beendet.

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 32 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

DEFINITION FÜR ‚CFG NUM‘:

Für die CfgNum wird ein 24Bit Wert verwendet, damit können theoretisch 16777215 unterschiedliche Konfigurationswerte adressiert werden.

Die Interpretation dieser Adresse ist vom jeweiligen Schienenformat abhängig.

Bei DCC werden die ersten 1024 Adressen gemäß NMRA in CV#1 (bzw. CV#17, CV#18 bei langen Adressen) verwendet.

Bei DCC Config Adressen > 1024 werden diese als indexierte CV interpretiert. Dazu werden die CV's #31, #32 auf den Wert der übergeben Adresse Modulo 256 gesetzt.

State	Code	Beschreibung
0x00	0x00	Nicht verwendet
0x10	0x00	Programmier Mode ‚Init‘
0x10	0x01	Service Mode sendet ‚Idle‘, wartet auf Programmierbefehl
0x10	0x02	Service Mode sendet ‚Reset‘: Programmierung ‚startet‘
0x2n		Fortschrittmeldung für Gleis ‚n‘. Count gibt den jeweiligen Fortschritt an. Im Service Mode ist das der Bit Abfrage Counter
0x3n		Fortschrittmeldung für Gleis ‚n‘, wenn im ‚Byte Verify‘ Mode.
0x8n		‚Busy‘: Programmiermode aktiv. Zentrale arbeitet Programmierbefehle ab und kann derzeit keine weiteren mehr annehmen.
0x9n		Stromverbrauch auf Gleis ‚n‘ beträgt ‚Val‘ mA, wahrscheinlich kein Decoder angeschlossen
0xFn		Fehlermeldungen

Hinweis: Dokumentation der Fehlermeldungen (0xFn) fehlt

TSE PROG CLEAR [0X06/0X16.0X04]

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x06	0x04	0b01		4	NID		FahrzeugNID					

Mit diesem Befehl (M=0b01) werden die im MX10 gespeicherten CV Werte des entsprechenden Fahrzeuges (FahrzeugNID) gelöscht.

TSE PROG READ [0X6/0X16.0X08]

Mit den TSE Read Befehlen können CV's aus einem (DCC) Decoder gelesen werden.

Ob dies per POM (Default) oder Service Mode Befehlen geschehen soll entscheidet der gewählte TSE Mode (Cmd=0x00).

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5 ... 8	DB9
0x06	0x08	0b01		8	Sys-NID		FahrzeugNID		Cfg Num	
0x06	0x08	0b11		9	Sys-NID		FahrzeugNID		Cfg Num	Val

ZIMO CAN Protokoll 4.20 Public.doc	Erstellt von Mike F. Schwarzer 09.04.2020 20:16:45	Seite 33 von 62
------------------------------------	---	-----------------

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5 ... 8	DB9 ... 10
0x16	0x08	0b01		8	Sys-NID		FahrzeugNID		Cfg Num	
0x16	0x08	0b11		10	Sys-NID		FahrzeugNID		Cfg Num	Val

Das Kommando (0b01) veranlasst, dass die Zentrale (NID) einen ‚Config Read‘ Befehl an den Schienendecoder (FahrzeugNID) sendet.

Solange der Lesebefehl ‚aktiv‘ ist, wird durch ‚TSE Info’s der Fortschritt gemeldet.

Sobald die Zentrale den gewünschten Config Wert hat wird dies durch ein ‚TSE Read ACK‘ Telegramm gemeldet.

Solange ein Read (auch Write) für einen Decoder aktiv ist, werden weitere Read/Write Befehle mit dem Info Datagramm ‚Busy‘ beantwortet.

An andere Decoder können jedoch problemlos POM Befehle gesendet werden.

TSE PROG WRITE [0X06/0X16.0X09]

Mit den TSE Write Befehlen können CV’s in einen (DCC) Decoder geschrieben werden.

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5 ... 8	DB9
0x06	0x09	0b01		9	Sys-NID		FahrzeugNID		Cfg Num	Val
0x06	0x09	0b11		9	Sys-NID		FahrzeugNID		Cfg Num	Val

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5 ... 8	DB9 ... 10
0x16	0x09	0b01		10	Sys-NID		FahrzeugNID		Cfg Num	Val
0x16	0x09	0b11		10	Sys-NID		FahrzeugNID		Cfg Num	Val

Das Kommando (0b01) veranlasst, dass die Zentrale (NID) einen ‚Config Write‘ Befehl an den Schienendecoder (FahrzeugNID) sendet.

Solange der Schreibbefehl ‚aktiv‘ ist, wird durch ‚TSE Info’s der Fortschritt gemeldet.

Sobald die Zentrale den gewünschten Config Wert geschrieben wird dies durch ein ‚TSE Write ACK‘ Telegramm gemeldet.

ACHTUNG:

Nach Read/Write Befehlen sollte das jeweilige Steuergerät (MX32, PC, ...) eine Pause von ca. 200ms einhalten. Dies dient dazu, dass andere Schienenbefehle abgearbeitet werden können.

TSE SEARCH [0X06/0X16.0X0D]

Dieses Datagramm startet die Fahrzeug Suche.

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x06	0x0D	0b01		5	SysNid				Mode			
0x06	0x0D	0b11		6	SysNid			FahrzeugNid	0x5A	Sec		

Mode = 0x5A (‚Z‘) → ZIMO Abkippsuche

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 34 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

TSE BIDI RAW DATA, BROADCAST [0X06/0X16.0X1D]

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x06	0x1D	0b10		5	Loco/AccNId	Port	Data					

Mit diesem Datagramm meldet das MX10 BiDi Roh Daten, welche sich im Broadcast Channel befinden.

Die AbsendeNId im Header gibt an, welches Modul die Daten empfangen hat.

Bei Modulen mit mehreren Ports, ist im Feld Port anzugeben, welches der Ports die Daten empfangen hat.

TSE BIDI RAW DATA, DATA CHANNEL [0X06/0X16.0X1E]

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x06	0x1E	0b10		4 .. 8	Loco/AccNId	Port	Data					

Mit diesem Datagramm meldet das MX10 BiDi Roh Daten, welche sich im Data Channel befinden.

Die AbsendeNId im Header gibt an, welches Modul die Daten empfangen hat.

Bei Modulen mit mehreren Ports, ist im Feld Port anzugeben, welches der Ports die Daten empfangen hat.

Im oberen Nibbel von DB4 befindet sich die Anzahl GÜLTIGER Datennibbels,
Das untere Nibbel von DB4 enthält das erste RailCom Roh Daten Nibbel, in den weiteren Bytes folgen dann die restlichen RailCom Roh Daten Nibbels.

TSE BIDI RAW DATA, ACK/NACK [0X06/0X16.0X1F]

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x06	0x1F	0b10		4	Loco/AccNId	Port	[N]ACK					

Mit diesem Datagramm meldet das MX10 BiDi ACK's bzw. NACK's.

Allerdings sind reine ACK's bzw. NACK's gemäß der aktuellen RailCom Spezifikation praktisch irrelevant.

Wenn ein Decoder RailCom Datagramme sendet, so hat der damit schon implizit auch ein ‚ACK‘ gesendet.

Ein dezidiertes NACK wurde per 19.03.2015 aus der Spezifikation entfernt.

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

DATA GROUP [0X07]

GROUP COUNT [0X07.0X00]

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x07	0x00	0b00		4	SrcNID		Group					
0x07	0x00	0b11		4	Group		Count					

Durch M = 0b00 kann ein Gerät abfragen, ob das MX10 eine bestimmte Geräte Gruppe (ObjType) kennt. Das MX10 antwortet (M = 0b11) mit Gruppe und der ihm bekannten Anzahl an Geräten in der jeweiligen Gruppe.

Anmerkung: SrcNID ist die Sourcequellen-NID (z.B. die NID vom MX10).

Group Count für MX8, MX9 Module liefert bei unbekannter Anzahl (z.B. weil die Anfrage zu früh, Autoscan off, Rückmeldefehler liegt vor,...) das Ergebnis 0xFFFF.

Ergebnis 0 auf eine Abfrage mit SrcNID und Group bedeutet das keine Objekte dieser Gruppe gibt.

GROUP CODES:

Group	
0x0000	Fahrzeuge
0x2F00	Züge
0x3000	Zubehör, DCC ‚simpel‘
0x3200	DCC ‚eXtended‘ Zubehördecoder
0x5040	MX8 Module (0x5040 bis 0x507F bzw. Nummer 0 bis 63 bzw. 800 bis 863)
0x5080	MX9 Module (0x5080 bis 0x50BF bzw. Nummer 0 bis 63 bzw. 900 bis 963)

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

ITEM LIST BY INDEX [0X07.0X01]

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x07	0x02	0b00		6	SrcNID		GroupNID		Index			
0x07	0x02	0b11		6	Index		NID		LastTick			

Durch M = 0b00 kann ein Gerät die Objekt Liste über den Objekt Index im MX10 abfragen. Das MX10 antwortet (M = 0b11) mit dem Objekt Index und der NID des Objektes. Dadurch kann ein Gerät eine Liste der dem MX10 bekannten Objekte aufbauen.

Fall 1: Gerät vorhanden

Wenn unter dem abgefragten Index ein Objekt im MX10 vorhanden ist, so liefert es in der Antwort den Index, die Nid des Gerätes und die Anzahl der mS seit der letzten Kommunikation mit dem Gerät.

Fall 2: Gerät nicht vorhanden/unbekannt

Wenn das MX10 unter dem angegeben Index kein Gerät kennt, so liefert es in der Antwort den abgefragten Index, als Nid=0xFFFF und ebenso als letzten Kommunikation-Tick 0xFFFF.

Ebenso, wenn der Index außerhalb der ‚Objektgruppe‘ liegt (z.B.: Bei MX8/MX9 sind nur Indexe von 0 ... 63) erlaubt), oder die ‚Objektgruppe‘ als solches unbekannt ist.

ITEM LIST BY NID [0X07.0X02]

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x07	0x02	0b00		4	SrcNID		Nid					
0x07	0x02	0b11		6	Nid		Index		LastTick			

Durch M = 0b00 kann ein Gerät jene Nid abfragen, welche nach der angegeben Nid gespeichert ist. Dieser Befehl ist insbesondere für Zubehör Module/Decoder hilfreich. Die Antwort (M = 0b11) enthält die ‚nächste‘ Nid, den jeweiligen Index und sofern vorhanden den letzten ‚Kommunikationstick‘.

Ähnlich wie bei ‚Item List by Index‘ [0x07.0x01] gibt es auch hier 2 Antwortmöglichkeiten:

1. Das MX10 findet ein ‚nächstes‘ Gerät nach der angegeben Nid in der gleichen Objektgruppe. In dem Falle liefert es die gefundene Nid, den Index und die letzten Kommunikationstick.
2. Dem MX10 sind keine weiteren Geräte in der Objektgruppe bekannt, die Nid verweist auf eine unbekannte Objektgruppe, ... In dem Falle antwortet das MX10 mit Nid=0xFFFF, Index=0xFFFF und LastTick=0xFFFF.

Beispiel:

Man sendet ans MX10 die Abfrage mit ScrNid=MX10 und Nid=0x5040.

Als Antwort vom MX10 erhält man das erste MX8, was das MX10 zu diesem Zeitpunkt kennt. Bei der nächsten Abfrage sendet man als Nid jene, welche man vom MX10 erhalten hat und bekommt als Antwort wiederum das nächste vorhandene, usw., usf. Bis eben 0xFFFF kommt, dann gibt es von diesem Typ einfach keine weiteren mehr.

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 37 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

HINWEIS ZU VERWENDUNG VON 0X07.0X01/0X07.0X02:

Beide Befehle erfüllen sehr ähnliche Aufgaben und liefern auch ähnliche Antworten. Mit ‚Item List by Index‘ wird jedoch ein ‚direkter‘ Speicher Zugriff ausgeführt. An der jeweiligen Speicherstelle können sich Daten (ein Objekt) befinden oder auch nicht. Wenn eine Abfrage auf Index z.B.: Index 10 ‚keine Daten‘ liefert, so können bei Index 11 durchaus noch welche vorhanden sein. Mit ‚Item List by NId‘, liefert das MX10 solange ‚positive‘ Antworten, wie es weitere Daten findet. ‚Leere‘ Speicherplätze werden dabei übersprungen.

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 38 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

DATA NAME [0X07.0X10]

Reserviert.

ITEM IMAGE CONFIG [0X07.0X12]

Reserviert.

PC ONLY: DATA NAME EXTENDED [0X07.0X21]

Dieser Befehl steht nur am PC Interface zur Verfügung (USB/LAN).

Damit kann eine App Texte mit bis zu 192 Zeichen in einem Befehl übertragen.

Etliche Einträge sind aber mit 32 Zeichen limitiert, bzw. gibt es in der GUI-Darstellung Limitierungen.

Hinweis: Namen und andere Zeichenketten sind 0x00 Terminiert zu senden!

Grp	Cmd	M	ID	DLC	DB 1 .. 2	DB 3 .. 4	DB 5 .. 8	DB 9 .. 12	DB 11 .. 14	DB 15 ... 204
0x07	0x21	0b00		14	SrcID	NID	SubID	Value 1	Value 2	
0x07	0x21	0b01		12	NID	Subl D	Value 1	Value 2	Z1 .. Z[x]	
0x07	0x21	0b11		12	NID	Subl D	Value 1	Value 2	Z1 .. Z[x]	

Die ‚NID‘ gibt an für welches Gerät der Text gilt.

Wenn ‚NID‘ z.B.: die NID eines Fahrzeuges ist, so werden die Texte mit diesem Fahrzeug verknüpft.

Auch alle anderen Texte können mit diesem Befehl übertragen werden.

NID	SubID	Value 1	Value 2	Verwendung	Max. Länge
Fahrzeug	0	0	0	Fahrzeugname	32 Zeichen

PC ONLY: LOCO GUI EXTENDED [0X07.0X27]

Reserviert.

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

INFO / CONFIG GROUP [0X08]

In der Info Group sind diverse Informationsabfragen und -meldungen zusammengefasst.

MODUL POWER INFO [0X08.0X00]

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x08	0x00	0b00		3	Ziel-NID		Port					
0x08	0x00	0b10		8	Port	0	Status		Track U		Track I	
0x08	0x00	0b11		8	Port	0	Status		Track U		Track I	

Port:

Bit	
0 ... 3	Port Nummer: 0=Schiene 1, 1=Schiene 2, 2=Booster
4 ... 7	Frei

Status:

Bit	
0 ... 3	,0' = Run, ,1' = SSP, ,2' = Service Mode ,3' = frei ,4' = Decoder Update ,5' = Sound Laden
4 ... 7	,0' = Run, ,1' = Unterspannung ,2' = Überstrom ,4' = Netzteilspannung
8 ... 9	Frei
10	Zugnummern Impulse
11	RailCom®
12	mfx®

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

MODUL INFO [0X08.0X08]												
Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x08	0x08	0b00		4	NID		Type					
0x08	0x08	0b01		8	NID		Type		Info			
0x08	0x08	0b11		6	Type		Info					

Über die Modul Info Datagramme können diverse Informationen abgefragt werden.

ACHTUNG:

Die meisten Informationen sind ‚Read Only‘ Informationen, können also NICHT per Command geändert werden. In der nachfolgenden Tabelle ist angegeben, welche Informationen Read Only/Write sind und das jeweilige Format.

INFO TYPES

Type	Verwendung	R/W	Format
1	Hardware Version	RO	
2	Software Version	RO	
3	Software Build Date	RO	Info Byte 1 = Tag Info Byte 2 = Monat Info Byte 3/4 = Jahr
4	Software Build Time	RO	Info Byte 1 = ‚0‘ Info Byte 2 = Sekunde Info Byte 3 = Minute Info Byte 4 = Stunde
5	Realtime Clock Date	RW	Info Byte 1 = Tag Info Byte 2 = Monat Info Byte 3/4 = Jahr
6	Realtime Clock Time	RW	Info Byte 1 = ‚0‘ Info Byte 2 = Sekunde Info Byte 3 = Minute Info Byte 4 = Stunde
7	Frei		
8	MiWi Hardware Version	RO	
9	MiWi Software Version	RO	
10	MiWi Channel Zentrale	RO	Aktueller MiWi Kanal der Zentrale Kommt auch ungefragt als ‚ACK‘, wenn an der Zentrale geändert
20	Modul Nummer	RO	‚Logische‘ Nummer des Moduls
100	Modul Art	RW	Info Byte 1/2: 0x2105 → Booster 10806 Info Byte 1/2: 0x2106 → Booster 10807 Info Byte 1/2: 0x9001 → Stein, Version 1 Info Byte 1/2: 0x9002 → Stein, Version 2 Info Byte 1/2: 0x9201 → Roco Detector

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

MODUL OBJECT/PROPERTY CONFIG [0X08.0X0A]

Daten Zugriffsdatagramm:

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x08	0x0A	0b00		6	Nid		Tag					
0x08	0x0A	0b01		8	Nid		Tag			Val		
0x08	0x0A	0b11		8	Nid		Tag			Val		

Mit diesen Datagrammen können Parameter des Gerätes 'NID' abgefragt (M=0b00) und geändert (M=0b01) werden. Die Rückantwort bzw. Bestätigung erfolgt durch M=0b11.

Bei Abfrage bzw. Änderung ist unter 'Nid' (z.B. ID von StEin, Objekt,...) das jeweils angesprochen Gerät anzugeben. Das Tag setzt sich dann z.B. zusammen aus 'Aggregat' und 'Item' ggf. mit einem Wert 'Val' und ist vom jeweiligen Gerät abhängig und ist aus der Beschreibung des Gerätes zu entnehmen.

Dabei bedeutet:

'Nid': Angesprochenes Gerät (z.B.: MX10, StEin, MX32,)

'Aggregat': oder Objekt, damit ist die jeweilige Teilfunktion der Gerätes gemeint, Teil von Tag

'Item': wäre ein Parameter, Teil von Tag

'Val': Wert

Beispiel: MX10 Schienen Spannung

Nid → MX10 Nid (Oder Booster)

Aggregat → Schienen Netzteil [0x0410]

Item → Spannung [0x0010]

Wert → Val [mV]

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 42 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

MODUL TAG LIST [0X08.0X0C]

Abfrage Datagramm:

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x08	0x0C	0b00		8	NId		Cmd	Dev	Tag			
0x08	0x0C	0b11		8	NId		Unit	Mul	Tag			

PC ABFRAGE/ANTWORT:

Grp	Cmd	M	ID	DLC	DB1/2	DB3	DB4	5/8	6/7	8/9	10/11	12/13	14...
0x08	0x0C	0b00		8	NId	Cmd	Dev	Tag	0x89				
0x08	0x0C	0b11		14...	NId	Cmd	Dev	Tag	0x89	Content	Min.	Max	Unit/Text

Durch dieses Datagramm kann ein Gerät die verfügbaren Objects und Properties eines Gerätes abfragen. Somit können ‚Fremd‘ Geräte ohne konkretes Wissen feststellen, was in einem ‚anderen‘ Gerät überhaupt einstellbar ist.

Kurzbeschreibung der notwendigen Vorgangsweise:

Gerät ‚A‘ möchte die möglichen Einstellwerte von Gerät ‚B‘ wissen:

- Somit sendet Gerät ‚A‘ einen Request an Gerät ‚B‘ (Per NId angegeben) mit Aggregat ‚0x0000‘ und Item ‚0x0000‘, Next = ‚1‘.
Gerät ‚B‘ antwortet daraufhin mit dem seinem ersten Aggregat und der ersten Eigenschaft dieses Aggregates. Ebenso welche Einheit diese Eigenschaft hat.
- Gerät ‚A‘ wiederholt nun seine Abfragen mit dem zuletzt empfangenen Aggregat/Item, und erhält so die nächste Geräteeigenschaft. Dies ist solange zu wiederholen, bis vom Gerät ‚B‘ die Antwort: Aggregat ‚0xFFFF‘ und Item ‚0xFFFF‘ kommt.

Command (Cmd):

Wenn ‚0‘ so wird die Definition vom angegebenen Tag zurückgeliefert. Ist der Tag unbekannt, so wird in den Datenbytes 3/4 0xFFFF zurückgegeben.

Wenn ‚1‘ so wird die Information vom nächsten Tag geliefert. Sollte es keine weiteren Tags mehr geben, so sind die Datenbyte 3 ... 8 0xFF.

Device (Dev):

Wird für MX8 (=8) bzw. MX9 (=9) verwendet, da für diese Geräte das MX10 diese Funktionalität simuliert.

HINWEIS PC:

Der PC bekommt eine deutlich umfangreichere Antwort, sofern ‚lange‘ Datagramme ‚eingeschalten‘ sind. Zu den 8 ‚Basis‘ Bytes kommt noch:

- Content
Bit ‚0‘: ni (Nicht implementiert)
Bit ‚1‘: RO, Wert kann nur gelesen werden (read-only)
- Bezeichnung für Einheit (4 Byte, fix)
- Bezeichnung des Wertes, wie am MX32 angezeigt (Bis 32 Bytes variabel).

Somit kann die Gesamtlänge bis zu 46 Bytes betragen.

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 43 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

PC ONLY: MODUL POWER INFO [0X08.0X20]

Reserviert.

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 44 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

NETWORK GROUP [0X0A]

In der Network Group sind all jene Telegramme zusammengefasst, welche sich mit dem Networkmanagement befassen.

PING [0X0A.0X00]

Grp	Cmd	M	NID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x0A	0x00	0b00		2	Nid/Grpid							
0x0A	0x00	0b10		8	Master-UID				Type	Session		

Durch einen Ping Request können Geräte andere Geräte am CAN Bus auffordern mit einem Ping ACK zu antworten. Dadurch kann jedes Gerät prüfen ob das / die andere(n) Gerät(e) vorhanden ist/sind. Beim Request kann die Nid eines bestimmten Gerätes angegeben werden. In dem Falle antwortet das Gerät unmittelbar, der Sender kann daher nach ca. 100mS davon ausgehen, dass das Gerät antwortet hat oder eben zur Zeit nicht verfügbar ist.

Wenn statt einer konkreten Nid jedoch eine Gruppen Nid (z.B.: 0xD000 für neue Module) abgefragt wird, so antworten alle Geräte dieser Gruppe mit einer zufälligen Verzögerung von bis zu 1000mS.

Alle Geräte (daher z.B. auch Rückmelde- und Schaltmodule) sollen regelmäßig einen Ping senden!

Dabei sollten **folgende Kriterien** eingehalten werden:

- Nach dem Start des Gerätes zumindest ein Ping innerhalb der ersten Sekunde.
- Im laufenden Betrieb zumindest einen Ping alle 10 Sekunden.
- Beide Zeitintervalle sollten durch einen Zufallswert einen bewussten Jitter aufweisen.

Um den CAN-Bus (125kbaud -> ca. 1000 Datagramme/Sekunde möglich) bei sehr vielen angeschlossenen Geräten nicht übermäßig mit Pings zu belasten (Stichwort CAN-Bus-Auslastung), gilt folgendes:

- Wenn ein Gerät ‚echte‘ Daten (z.B. Event, ACK,...) sendet, gilt dies automatisch als Ping. Daher jedes Gerät/Modul hat sozusagen eine ‚freie‘ Nachricht/Sekunde, ohne dass die CAN-Buslast steigen würde.
- MX10/MX32 verlängern das Ping Timeout automatisch auf 2 Sekunden, wenn die CAN-Bus-Traffic 50% erreicht. Bei Erreichen von 75% CAN-Bus-Auslastung wird das Ping Timeout auf 5 Sekunden erhöht, bei 90% auf 10 Sekunden.
- Ebenso verlängert sich das Ping Timeout anhand der Nodeanzahl (Die Anzahl der vorhanden aktiven Nodes erkennen MX10/MX32 am Ping selbst): Bei >200 Nodes auf 2 Sekunden, bei >500 Nodes auf 5 Sekunden, bei >1000 Nodes auf 10 Sekunden.

Durch diese Regeln sollte die CAN-Bus-Belastung durch Pings bei ca. 10% bis 20% bleiben, meist aber eher darunter, da ja jede ‚echte‘ Meldung einen Ping von dieses Gerätes ersetzt.

Die primäre Zentrale versendet diesen Befehl etwa alle 500ms, zumindest jedoch jede Sekunde.

Master-UID: UID der Zentrale
 Type: Art der Zentrale, siehe Tabelle
 Session: Session Nummer

Anhand dieses Befehls sollen die angeschlossenen Module erkennen, dass sie immer noch mit der Ihnen bekannten Zentrale verbunden sind. Dabei muss auch die Session Nummer geprüft werden. Diese Session Nummer wird von der Zentrale bei jeder UID Änderung inkrementiert. Dies erfolgt z.B.: wenn die Zentrale ein neues Objekt in Ihre Objektliste aufnimmt oder wenn ein vorhandenes aus

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 45 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

dieser Liste gelöscht wird.

Erkennt ein Modul, dass es mit einer ‚unbekannten‘ Zentrale verbunden ist, so muss es einen Anmeldevorgang initiieren.

Für Ping und Modul Info gilt:

Kennung	Geräte
0x1nnn	MX10
0x02nn	Z21
0x0205	Roco Booster 10806
0x0206	Roco Booster 10807
0x90nn	StEin
0x92nn	Roco Besetztmelder 10808

Anhand dieses Befehls sollen die angeschlossenen Module erkennen, dass sie immer noch mit der Ihnen bekannten Zentrale verbunden sind. Dabei muss auch die Session Nummer geprüft werden. Diese Session Nummer wird von der Zentrale bei jeder UID Änderung inkrementiert. Dies erfolgt z.B.: wenn die Zentrale ein neues Objekt in Ihre Objektliste aufnimmt oder wenn ein vorhandenes aus dieser Liste gelöscht wird.

Erkennt ein Modul, dass es mit einer ‚unbekannten‘ Zentrale verbunden ist, so muss es einen Anmeldevorgang initiieren.

PORT OPEN [0X0A.0X06]												
Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x0A	0x06	0b01		0								

Mit diesem Datagramm kann ein Gerät die Ethernet Schnittstelle des MX10 ‚öffnen‘. Als Antwort bekommt das Gerät ein ‚Ping‘ vom MX10.

LOGOFF / PORT CLOSE [0X0A.0X07]												
Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x0A	0x07	0b01		2	NID							

Durch dieses Telegramm kann sich ein Gerät von einer Zentrale abmelden. Sofern dies eine PC Software sendet, wird dadurch auch automatisch das jeweilige Kommunikationssport (USB oder Ethernet) geschlossen. Das jeweilige Gerät muss zur Wiederaufnahme der Verbindung wieder die jeweiligen Initialschritte abarbeiten. Als (Ziel-) NID ist dabei die NID jenes Gerätes anzugeben, von welchem sich der Schnittstellen Benutzer (PC Software) abmelden will.

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

INTERFACE OPTION [0X0A.0X0A]

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x0A	0x0A	0b00		4	NID		Type					
0x0A	0x0A	0b01		8	NID		Type		Value			
0x0A	0x0A	0b11		8	NID		Type		Value			

Durch diese Datagramme kann eine PC Software diverse Kommunikationsoptionen abfragen bzw. einstellen.

INFO VALUES FÜR TYPE 0X0001

Type	Value	Verwendung
0x0001	0x0000	ZIMO Intern
	0x0010	Kennung für ESTWGJ
	0x0018	RocRail
	0x0020	Kennung für STP
	0x0021	Kennung für Pfusch
	0x0030	Kennung für TrainController
	0x0031	Kennung für TrainProgrammer
	0x0040	RailManager
	0x0050	Win-Digipet
	0x0070	iTrain

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

INTERFACE ERROR [0X0A.0X0F]

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x0A	0x0F	0b11		2	Option							

Dieses Datagramm wird vom MX10 gesendet, wenn ein Befehl fehlerhafte Parameter enthält oder aus anderen Gründen nicht ausführbar ist.

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

NETWORK GROUP [0X1A] ÜBER LAN

PORT OPEN [0X1A.0X06]

Grp	Cmd	M	ID	DLC	DB 1 ... 4	DB 5 .. 8	DB 9 ...
0x1A	0x06	0b01		0, 8...	Options	App. Code	App Name

Mit diesem Datagramm kann ein Gerät die Ethernet Schnittstelle des MX10 ‚öffnen‘.
Als Antwort bekommt das Gerät ein ‚Ping‘ vom MX10.

Mit Länge ‚0‘ erfolgt ein simples ‚Open‘ der Schnittstelle, weitere Angaben sind nicht nötig.

Wenn Länge grösser 8, dann:

Enthalten die ersten 4 Byte Optionen

In den Bytes 5 ... 8 folgt eine Anwendungs-Kennung (Eindeutige Programm Kennung)

Danach können bis zu 24 Zeichen als Anwendungs-Name gesendet werden.

Dieser Name wird in diversen Anzeigen am MX10 und MX32 verwendet.

Wird KEIN Name gesendet, so wird die 32 Bit Kennung angezeigt, wird auch diese nicht gesendet,
so wird schlicht ‚PC-Steuerung‘ angezeigt. Dies ist insbesondere bei größeren Anlagen mit mehreren
Stellwerken extrem Anwenderunfreundlich.

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

RAILWAY CONTROL SYSTEM (RCS) [0X0B]

Die Datagramme dieser Gruppe dienen der Kommunikation mit Stellwerken (Railway Control System, in weiterer Folge mit RCS abgekürzt). Sie dienen in erster Linie einem verbesserten Zusammenspiel zwischen dem ZIMO System (MX10, MX32/FU) und einer PC Stellwerks Software.

Diese Befehle sind erst nach einer primären Identifikation der Stellwerkssoftware verwendbar. Diese Identifikation läuft über die Interface Options (Grp=0x0A, Cmd=0x0A, Type=0x0000, 0x0001) ab.

Ebenso ist es sinnvoll (aber nicht zwingend erforderlich), wenn ein PC Software Ihren Namen dem System mitteilt (Data Group 0x07, CMD=0x21, NID=0xC199 ... 0xC1FF).

Das System kann bis zu 64 Stelltische mit maximal 8192 Tischfeldern verwalten. Die Aufteilung der Tischfelder zu den Stelltischen erfolgt dynamisch, es kann also z.B.: ein Monster Tisch mit 8192 Felder definiert werden, oder eben bis zu 64 Tische mit entsprechend weniger Felder.

Im Folgenden wird ein Stelltisch meist als ‚Tab‘ bezeichnet. Viele Stellwerks- bzw. PC Steuerprogramme verwenden Tab's um Stelltische übersichtlich darzustellen bzw. um Module je nach Bedarf zusammenzustellen und/oder zu organisieren. In einigen Fällen wird dies auch als ‚Z‘ Ebene bezeichnet.

Diese Befehle stehen NICHT zur allgemeinen Verwendung zur Verfügung.

Erlaubt ist die Nutzung von Group 0x0B für Stellwerkprogrammen, um z.B. bei falscher Farbe der Weichenstellung (grün und rot vertauscht) im MX32 in der Zubehörlistenansicht zu ändern.

RCS OPTIONS [0X0B.0X01]

Grp	Cmd	M	ID	DLC	DB1	DB2	DB3	DB4	DB5	DB6	DB7	DB8
0x0B	0x01	0b00		2	Option							
0x0B	0x01	0b01		3...8	Option							
0x0B	0x01	0b1x		3...8	Option							

OPTION FÜR RCS OPTION

Value	Verwendung
0x01nn	RGB Wert für Ausleuchtung, nn=Stellung
	0x0100: Ausleuchtung ‚Gerade‘
	0x0101: Ausleuchtung ‚Abzweig‘
	DB3 Helligkeit für Rot, DB4 Helligkeit von Grün, DB5 Helligkeit für Blau

FILE CONTROL [0X0E]

Reserviert.

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 50 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

FILE TRANSFER [0X0F]

Reserviert.

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 51 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

FUNKTIONELLE EIGENSCHAFTEN

ABLAUF FAHRZEUG ‚AKTIVIEREN‘

Mit dem im Folgenden beschriebenen Ablauf aktiviert ein Fahrpult bzw. sonstiges Steuergerät (z.B.: Computer) ein Fahrzeug, um dieses zu steuern.

Schritt	
1	Abfrage des Fahrzeug Status
2	Antwort abwarten, max. 500ms. Kommt in dieser Zeit keine Antwort, so kann das MX10 das gewünschte Fahrzeug nicht aktivieren. Ein Steuern des Fahrzeuges ist somit unmöglich.
3	Abfrage des Fahrzeug Modes bzw. Setzen des Fahrzeug Modes, insbesondere für neue Fahrzeuge
4	Antwort abwarten, max. 500ms. Normalerweise kommt die Antwort in weniger als 10ms. Sollte die Antwort nicht innerhalb von 500ms kommen, so liegt ein Fehler vor.
5a	Ab hier kann das Fahrzeug in vollem Umfang gesteuert werden. Sämtliche Fahr, Schalt und POM Befehle können genutzt werden.

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

ABLAUF MX8, MX9

Das MX10 verwaltet für diese Module eigene Objekte. Grundsätzlich werden Abfragen von diesem Objekt-speicher beantwortet bzw. Befehle in diesen Objektspeicher eingetragen.

Jedes dieser Objekte bildet gleichzeitig eine autonom laufende Task-Engine. Diese sendet bei Daten- bzw. Zustandsänderungen (durch Befehle) die passenden Befehle an die Module (MX8/MX9).

Umgekehrt werden alle Informationen von diesen Modulen ebenfalls im jeweiligen Objektspeicher eingetragen und danach an den PC weitergeleitet.

Diese Logik hat sowohl Vorteile als auch Nachteile:

Vorteile:

- Im laufenden Betrieb kann die volle Bandbreite des PC Interfaces genutzt werden.
- Die fortlaufende Überwachung der Module wird vom MX10 übernommen.
- Einheitliche Kommandologik, egal ob es sich um ein MX8, MX9 oder später StEin handelt.

Nachteile:

- Unmittelbar nach dem Hochfahren des MX10 sind alle Daten ‚invalid‘

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 53 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

TABELLEN:

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 54 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

ANHANG:

FAHRZEUG STATUS FLAGS

SPEED & FLAGS

In den obersten 6 Bit des 16 Bit Speed Wertes sind folgende Flags kodiert:

Bit	Beschreibung	Gültige Werte
0 ... 09	Fahrzeug Geschwindigkeit Hinweis: Für alle Schienenformaten ist der ZIMO Wert 1008 die Max. Fahrstufe, Werte > 1008 werden automatisch auf 1008 skaliert. In diesen Bits ist die Geschwindigkeit auf 1008 skaliert zu senden, bzw. empfangen	0 ... 1023
10	Richtungsbit ans System (Client an MX10) ,0' → Vorwärts, ,1' → Rückwärts	0/1
11	Richtungsbit vom System (MX10 an Client) ,0' → Vorwärts, ,1' → Rückwärts	0/1
12 ... 14	Derzeit frei Müssen ,ignoriert' werden	0
15	Fahrzeug Emergency Stopp	0/1

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

EINGETRAGENE MARKENZEICHEN

mfx®	Gebr. Märklin & Cie. GmbH
Motorola®	Motorola Inc., Tempe-Phoenix, USA
ZIMO	ZIMO Elektronik GmbH
HLU	ZIMO Elektronik GmbH
DCC	NMRA
RailCom®	Lenz Elektronik GmbH
LocoNet	Digitrax Inc.
Android	Google Inc.
iPad, iPhone	Apple Inc.
iOS	Apple Inc.
App Store	Apple Inc.
Google Play	Google Inc.

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 56 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

HAFTUNGSAUSSCHLUSS

Information von der Firma ZIMO Elektronik GmbH:

Die Firma ZIMO Elektronik GmbH erklärt ausdrücklich, in keinem Fall für den Inhalt in diesem Dokument oder für in diesem Dokument angegebene weiterführende Informationen rechtlich haftbar zu sein.

Die Rechtsverantwortung liegt ausschließlich beim Verwender der angegebenen Daten oder beim Herausgeber der jeweiligen weiterführenden Information.

Für sämtliche Schäden die durch die Verwendung der angegebenen Informationen oder durch die Nicht-Verwendung der angegebenen Informationen entstehen übernimmt ZIMO Elektronik GmbH ausdrücklich keinerlei Haftung.

Die Firma ZIMO Elektronik GmbH übernimmt keinerlei Gewähr für die Aktualität, Korrektheit, Vollständigkeit oder Qualität der bereitgestellten Informationen. Haftungsansprüche, welche sich auf Schäden materieller, immaterieller oder ideeller Art beziehen, die durch die Nutzung oder Nichtnutzung der dargebotenen Informationen verursacht wurden, sind grundsätzlich ausgeschlossen.

Die Firma ZIMO Elektronik GmbH behält es sich vor, die bereit gestellten Informationen ohne gesonderte Ankündigung zu verändern, zu ergänzen oder zu löschen. Alle innerhalb des Dokuments genannten und gegebenenfalls durch Dritte geschützten Marken- und Warenzeichen unterliegen uneingeschränkt den Bestimmungen des jeweils gültigen Kennzeichenrechts und den Besitzrechten der jeweiligen eingetragenen Eigentümer.

Sollten Teile oder einzelne Formulierungen des Haftungsausschlusses der geltenden Rechtslage nicht, nicht mehr oder nicht vollständig entsprechen, bleiben die übrigen Teile des Haftungsausschlusses in ihrem Inhalt und ihrer Gültigkeit davon unberührt.

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 57 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

GLOSSAR

Begriff	Erklärung
UID	Weltweit eindeutig 32 Bit Nummer (Unique Identifier). Diese wird typischerweise während des Anmeldeprozesses verwendet.
NID	Network ID, 16 Bit Nummer welche im laufenden Betrieb zur Adressierung der Module, Fahrzeuge, Decoder,... verwendet wird.
TSE	Track Signal Engine. Jener Programmteil, welcher die logischen Befehle in die jeweiligen Schienen Befehle (DCC, MM2, mfx, ...) umsetzt. Ebenso ist dieser Programmteil für die Synchronisierung des Schienen Empfangs (RailCom, ZACK, mfx) zuständig.
OBJECT	Der Begriff Objekt bezeichnet eine allgemeine Datenstruktur, welche Daten unterschiedlicher Module, Fahrzeuge, Decoder, Enkoder, etc. enthält. Diese Struktur kann dabei in abstrakter Form oder als konkreter Eintrag in einer Datenbank verwendet werden.
OBJDB	Die Objekt Datenbank beruht auf OBJECT's (siehe oben). Die konkreten Werte der Objekte werden in der OBJDB koordiniert verwaltet und je nach Bedarf permanent gespeichert.

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

REFERENZ CODE IN C# FÜR PC ANBINDUNG

Im Folgenden befinden sich einige Beispiele und Hilfsfunktionen für die PC Kommunikation:

UMWANDLUNG VON 16BIT ZAHLEN:

Da das interne Protokoll im Little Endian Format arbeitet, PC's jedoch das Big Endian Format verwenden ist eine Umwandlung zwischen diesen Formaten erforderlich.

Um eine Zahl aus einem Byte Stream (typischerweise Empfangsdaten vom System) in eine 16 Bit Zahl für den PC umzuwandeln ist folgende Funktion sinnvoll:

Die Funktion geht davon aus, dass die Empfangsdaten in einem Byte Buffer mit Namen `iData[.]` vorliegen. Durch `_iByte` wird angegeben ab welchem Byte die Zahl in diesem Bytearray liegt. Die jeweils 'umgewandelte' Zahl wird dem Aufrufer zurückgegeben.

```
public UInt16 DataI16Get(int _iByte)
{
    UInt16 iTemp;
    iTemp = (UInt16)((iData[_iByte + 0] >> 0) & 0x00FF);
    iTemp |= (UInt16)((iData[_iByte + 1] << 8) & 0xFF00);
    return (iTemp);
}
```

Natürlich ist auch eine Umkehrung zum Senden von 16Bit Zahlen erforderlich, dies kann mit folgender Funktion geschehen:

Als Parameter sind `_iByte` und `_iData` zu übergeben, dabei bestimmt `_iByte` ab welcher Position die Zahl in den Byte Stream (Buffer) einzutragen sind. `_iData` ist dabei die Zahl, welche entsprechend umzuwandeln ist.

```
public void DataI16Set(int _iByte, UInt16 _iData)
{
    iData[_iByte + 0] = (byte)((_iData >> 0) & 0xFF);
    iData[_iByte + 1] = (byte)((_iData >> 8) & 0xFF);
}
```

ZIMO CAN Protokoll 4.20 Public.doc	Erstellt von Mike F. Schwarzer 09.04.2020 20:16:45	Seite 59 von 62
------------------------------------	---	-----------------

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

CHANGELOG

Datum	Änderung
2015.03.28	Ergänzung Fehlermeldungen, Insbesondere MX8 Korrektur RCS Command Group Fehlerbehandlung für MX8 Module
2015.03.30	Nachtrag für ‚alte‘ Befehle.: Accessory Port hat nun wie schon lange vorgesehen zusätzlich einen ‚Type‘ und 16 Bit Values
2015.03.31	Beschreibung für ‚Field Actuator‘ ergänzt
2015.04.01	Data Name eXtended [0x21]: Fehler in der Byte Zählung behoben
2015.04.02	Grp 0x06 Com 0x0B: Cfg Val korrigiert (5 -> 4 Byte)
2015.04.07	Accessory Port Befehl in PIN4 und Pin6 getrennt. Die PIN4 Befehle sind 3 bzw. 4 Byte lang verhalten sich so wie in den ‚alten‘ Dokumentationen beschrieben. Die erweiterten Port Befehle haben ab sofort eine eigene Command Kennung [0x06]. Haftungsausschluß Roco wurde ergänzt
2015.04.14	Accessory Port Befehle für StEin Module erg.
2015.04.14	Config Befehle (Grp 0x05) für StEin Module erg.
2015.04.15	Grp 0x01 Com 0x06: DLC korrigiert (3 -> 4 Byte), Com korrigiert (0x04 -> 0x06); Grp 0x06 Com 0x0B, M=0b11: Val in BD7 ergänzt Grp 0x07 Com 0x00, M=0b11: Com korrigiert (0x01 -> 0x00) Diverse eindeutige falsche DLC korrigiert
2015.04.16	Ergänzung Beschreibung für MX8 Fehlmeldungen, insbesondere für möglicherweise falsche PORT Meldungen.
2015.04.23	Änderung Reihenfolge bei Abfrage von Namen (Grp=0x07, Req=0x10)
2015.04.28	Korrektur für Loco Info
2015.04.29	Ergänzung Info Type 8, MiWi Channel der Zentrale
2015.04.29	Modul Info in Gruppe Info verschoben (War fälschlicherweise in Gruppe NetWork)
2015.05.04	Grp 0x01 Com 0x04 wieder auf alte Stand zurückgesetzt
2015.05.11	Korrektur für Grp=0x00, Cmd=0x00 Ergänzung der Modi für SSP, Fahrstufe ‚0‘ bzw. Emergency SSP
2015.05.13	Grp=0x02, Cmd=0x04: Ergänzung Port Nummer um ‚Valid‘ Bit.
2015.05.13	Grp=0x01, Cmd=0x04 erneut korrigiert; Grp=0x01, Cmd=0x06 ergänzt;
2015.06.16	Modul Config, Insbesondere MX8/MX9 Group 0x05 ab sofort frei; Modul Config nun in Info Gruppe [0x08]
2015.06.17	Grp=0x02, Cmd=0x02, Accessory Mode Beschreibung um MX8/MX9 Modi ergänzt
2015.07.20	Das TSE Mode Ack enthält nun die Serv. Prog. Spannung und Strom
2015.08.24	Grp=0x04, Cmd=0x05 um Signalbilder und Beschreibung ergänzt Grp=0x04, Cmd=0x0C Feld Folge richtiggestellt, Feld Action Werte festgelegt
2015.08.26	Grp Angaben in den Datagramm Tabellen der Info Group richtiggestellt
2015.09.04	Grp=0x04, Cmd=0x05 RCS Look Ahead Signal Definition überarbeitet
2015.11.10	MAN/Rg Funktion aus Speed entfernt und als eigene Funktion definiert
2016.01.11	Fehler Code Tabelle MX10, StEin überarbeitet
2018.11.28	Ergänzung: Modul Eigenschaftsabfrage [0x08.0x0C]
2019.04.19	Ergänzung: Accessory Track Multilimit [0x01.0x09] mit Software 01.26.0001

ZIMO CAN Protokoll 4.20 Public.doc	Erstellt von Mike F. Schwarzer 09.04.2020 20:16:45	Seite 60 von 62
------------------------------------	---	-----------------

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

Datum	Änderung
2019.06.03	Grp 0x08 Cmd 0x0A: Beschreibung ergänzt; Grp 0x08 Cmd 0x0C: Beschreibung ergänzt; Grp 0x01 Cmd 0x09: O/W Code ergänzt, Beschreibung ergänzt;
2020.04.09	Ergänzung: Unterkapitel Zusatzbefehle für PC Anbindung über LAN, Network-Grp über LAN erg.

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 61 von 62

ZCAN20	PUBLIC	Vers. : 4.20
ZIMO CAN Protokoll 2.00, Geräteserie ZS		

FAHRZEUG STEUERN, ‚SIMPLE DRIVE‘:

Die ‚Simple Drive‘ Methode ist wie der Name schon sagt, die einfachste Art Fahrzeuge per PC zu steuern. Bei dieser Methode sendet der PC schlicht Fahr- und Funktionsbefehle ohne weitere Rücksichtnahme auf das System.

In diesem Falle entscheidet das MX10 anhand der Befehlsdaten in welcher Priorität die Befehle an die Schiene zu senden sind und ‚exekutiert‘ den Befehl.

System intern (also MX32) werden diese Befehle gespiegelt und kurzfristig angezeigt. Da die PC Software auf diese Art und Weise jedoch das Fahrzeug NICHT übernimmt, gibt es auch keinen Übernahme Dialog oder sonstige Zusammenarbeit zwischen System (MX32 + MX10) und der PC Software.

FAHRZEUG STEUERN, ‚FAHRPULT STYLE‘:

Bei dieser Methode verhält sich eine PC Software wie ein ‚reales‘ Fahrpult.

In diesem Falle muss sich durch den ‚Aktiv‘ Befehl Fahrzeuge übernehmen, die Übergabe Logik implementieren und auch etwa alle 500mS die Fahrzeug aktiv Meldung senden.

Natürlich kann eine PC Software dies für mehrere Fahrzeug gleichzeitig machen, bzw. einen Teil der Fahrzeuge im ‚Fahrpult Style‘ steuern und andere Fahrzeuge mit einer der beiden anderen Methoden.

ACHTUNG:

Wenn eine PC Software mehrere Steuermethoden anwendet, so muss sie selber für einen sauberen Übergang zwischen diesen Methoden sorgen.

FAHRZEUG STEUERN, ‚VOLLWERTIGES STELLWERK‘:

	Erstellt von Mike F. Schwarzer	
ZIMO CAN Protokoll 4.20 Public.doc	09.04.2020 20:16:45	Seite 62 von 62