# 1. The new binary interface protocol

**This document is meant for ZIMO users who wish to use self programmed software….**

Please Note: Content marked with (Version x) describes functionality available since protocol version x.

Version 5 is introduced with MX1 software version 3.06.

# 1. Communication basics:

The basic communication format is N,8,1 (no parity, 8 data bits, 1 stop bit). Communication speed ranges from 1200 to 38400 bit/s and can be selected via CV12. The default value is 9600 bit/s.

# 2. Flow control

Flow control is done via hardware (RTS/CTS). It is necessary to use a serial cable with at least 5 wires which also connects the RTS and CTS lines between PC and command station. Hardware flow control can be disabled by setting CV13 to 0. Default value is 1 (hardware flow control enabled).

# 3. The new binary communication

The new binary communication consists of data frames with the following structure:

- long frames:

<SOH><SOH>headerinfo_and_data[checksum16]<EOT>

- short frames (for frames of maximum 15 bytes):

<SOH><SOH>headerinfo_and_data[checksum8]<EOT>

To establish a data transmission independent of content, all data (including headerinfo and checksum) characters identical to control characters have to be protected with the additional escape character prefix <DLE> and the character itself is XOR'ed with 0x20.

| control character | value | replacement within data | description |
|---|---|---|---|
| <SOH> | 0x01 | <DLE>(<SOH>^0x20) | start of a data frame |
| <EOT> | 0x17 | <DLE>(<EOT>^0x20) | end of data frame |
| <DLE> | 0x10 | <DLE>(<DLE>^0x20) | escape character |

Each data frame is immediately acknowledged by the receiver (level 1 reply). This reply may contain the appropriate data if it's immediately available in the command station. Otherwise the request is passed on via the CAN bus (to another station for example) and the subsequent incoming data is returned (level 2 reply).

Please note: Future expansions of single messages may involve additional bytes added to the frame.

## 3.1. Header info

The header info describes the meaning of the data content of a frame. The length of the header info is between 2 and 15 bytes. The first byte is the unique sequence-ID of the frame and must not be identical in two different consecutive frames. If a frame has to be repeated the sequence-ID remains unchanged. The second byte specifies the message type and the meaning of additional header bytes. Most messages are identified by the contents if the 2nd and 3rd header byte. The 2nd byte provides some kind of routing information the 3rd byte identifies a specific message. Together these two bytes might be considered as unique 16 bit message identifier.

**common information in header byte 2:**

| bit(s) | value | description |
|---|---|---|
| 7 | 0 | short frame protocol message |
| | 1 | long frame protocol message |
| 6 - 5 | 00 | "primary" message – identified by bits 4-0 and byte 3 |
| | 10 | acknowledgement / reply level 1 |
| | 01 | reply level 2 |
| | 11 | acknowledgement for level 2 reply |
| 4 | 0 | message sent by command station |
| | 1 | message sent by PC |
| 3 - 0 | 0 | message to/from command station (MX1) |
| | 1 | message to/from accessory module (MX8) |
| | 2 | message to/from track section module (MX9) |

Bits 4-0 may alternatively considered to be a 5 bit part of the unique message identifier. It should not be taken for granted that "useless" bit combinations involving bit 4 strictly interpreted as origin identifier that are not used at this stage might not be assigned some other meaning some day.

### 3.1.1. short frame protocol messages

**general messages**

**acknowledgement/reply sent by command station (level 1)**

| header byte | Value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0100xxxx | acknowledgement/reply sent by command station (level 1) |
| 3 | XXX | same as primary message |
| 4 | ID | sequence-ID of the message that is being acknowledged |
| 5..15 | data | optional: requested data or additional return code |

***Reply:*** n.a.

**acknowledgement/reply sent by PC (level 1)**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0101xxxx | acknowledgement/reply sent by PC (level 1) |
| 3 | XXX | same as primary message |
| 4 | ID | sequence-ID of the message that is being acknowledged |
| 5..15 | data | optional: requested data or additional return code |

***Reply:*** n.a.

**acknowledgement/reply sent by command station (level 2)**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0010xxxx | acknowledgement/reply sent by command station (level 2) |
| 3 | XXX | same as primary message |
| 4 | ID | Sequence-ID of the message that is being acknowledged |
| 5..15 | data | optional: requested data |

***Reply:*** Ack (level 1)

**acknowledgement/reply sent by PC (level 2)**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0011xxxx | acknowledgement/reply sent by PC (level 2) |
| 3 | XXX | same as primary message |
| 4 | ID | sequence-ID of the message that is being acknowledged |
| 5..15 | data | optional: requested data |

***Reply:*** Ack (level 1)

**command station messages**

These messages provide communication with the command station.

**reset message**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x00<br>0x10 | command station instruction (sent by command station)<br>(sent by PC) |
| 3 | 0 | reset message |

***Reply:*** Ack (level 1)

**negative acknowledgement sent by command station**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x00<br>0x10 | command station instruction (sent by command station)<br>(sent by PC) |
| 3 | 1 | negative acknowledgement |

***Reply:*** n.a.

**track control**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x10 | command station instruction |
| 3 | 2 | track control |

| | | |
|---|---|---|
| 4 | cAction | 0 stop broadcast (stops all locos)<br>1 switch track voltage OFF<br>2 switch track voltage ON and re-enable broadcast<br>3 query status |

***Reply:*** Ack (level 1)

| **return code** | **Value** | **Description** | |
|---|---|---|---|
| StatusBits | dm000uts | d ……. | DCC (1 on, 0 off) |
| | | m …… | Motorola (1 on, 0 off) |
| | | u ……. | UES (1 UES on, 0 UES off) |
| | | t …….. | track voltage (1 off, 0 on) |
| | | s ……. | broadcast stop (1 on, 0 off) |

**loco control**

(Version 5) With Version 5 an emergency stop is executed when the MSB of the speed step value is set. Additionally data bytes 7-9 are optional. If these values are omitted non of the bits will be altered.

| **header byte** | **value** | **description** |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x10 | command station instruction |
| 3 | 3 | loco control |
| 4 | cAdr_hi | loco address high byte<br>bit 6 + bit 7 format specifications (see 4.5) |
| 5 | cAdr_lo | loco address low byte |
| 6 | cSpeed | Speed step in the actual speed step system<br>If bit 7 is set: emergency stop<br>(DCC data signal: 0-14, 0-28 or 0-126; according to bit 3/2 in cData1;MOTOROLA protocol: 0-14) |
| 7 | cData1 | optional:<br>bit 7: MAN (override signal controlled speed limit)<br>bit 6: no function<br>bit 5: direction (0 = forward, 1 = backward)<br>bit 4 headlights on/off (= DCC function F0)<br>bit 3/2 DCC speed step system (see 4.6)<br>bit 1 decel. time “BZ” (definable by command “B”) on/off<br>bit 0 accel. time “AZ” (definable by command “B”) on/off |
| 8 | cData2 | bit 0-7: function outputs 1-8 on/off (DCC only) |
| 9 | cData3 | bit 0-3: function outputs 9-12 on/off (DCC only) |

***Reply:*** Ack (level 1) see also 4.7 and 4.9

**invert function bits**

| **header byte** | **value** | **description** |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x10 | command station instruction |
| 3 | 4 | invert function bits |
| 4 | cAdr_hi | loco address high byte<br>bit 6 + bit 7 format specifications (see 4.5) |
| 5 | cAdr_lo | loco address low byte |
| 6 | cData1 | the following bits are toggled when set:<br>bit 7: MAN (override signal controlled speed limit)<br>bit 6: no function<br>bit 5: direction (0 = forward, 1 = reverse)<br>bit 4 headlights on/off (= DCC function F0)<br>bit 1 decel. time “BZ” (definable by command “B”) on/off<br>bit 0 accel. time “AZ” (definable by command “B”) on/off |
| 7 | cData2 | bit 0-7: function outputs 1-8 on/off (DCC only) |
| 8 | cData3 | bit 0-3: function outputs 9-12 on/off (DCC only) |

***Reply:*** Ack (level 1) see also 4.7 and 4.9

**acceleration / deceleration (AZ / BZ) command**

| **header byte** | **value** | **description** |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x10 | command station instruction |
| 3 | 5 | acceleration / deceleration command |
| 4 | cAdr_hi | loco address high byte<br>bit 6 + bit 7 format specifications (see 4.5) |
| 5 | cAdr_lo | loco address low byte |
| 6 | cAzBz | bit 0-3: BZ (0 - 15)<br>bit 4-7: AZ (0 - 15) |

***Reply:*** Ack (level 1) see also 4.7 and 4.9

**shuttle train command**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x10 | command station instruction |
| 3 | 6 | shuttle train command |
| 4 | cAdr_hi | loco address high byte<br>bit 6 + bit 7 format specifications (see 4.5) |
| 5 | cAdr_lo | loco address low byte |
| 6 | cData | bit 0-3: contact rails 1-4 forward<br>bit 4-7: contact rails 1-4 reverse |

***Reply:*** Ack (level 1) see also 4.7 and 4.9

**accessory decoder command**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x10 | command station instruction |
| 3 | 7 | accessory decoder command |
| 4 | cAdr_hi | accessory address high byte<br>bit 6 + bit 7 format specifications (see 4.5) |
| 5 | cAdr_lo | accessory address low byte |
| 6 | cData | bit 3: switching output on / off (1=on, 0=off)<br>bit 2-0: number of output that should be activated |

***Reply:*** Ack (level 1) see also 4.7 and 4.10

**query command station's loco memory**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x10 | command station instruction |
| 3 | 8 | query command station's loco memory |
| 4 | cAdr_hi | address high byte<br>bit 6 + bit 7 format specifications (see 4.5) |
| 5 | cAdr_lo | address low byte |

***Reply:*** Ack (level 1) see also 4.7 and 4.11

**query command station's accessory decoder memory**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x10 | command station instruction |
| 3 | 9 | query command station's accessory decoder memory |
| 4 | cAdr_hi | address high byte<br>bit 6 + bit 7 format specifications (see 4.5) |
| 5 | cAdr_lo | address low byte |

***Reply:*** Ack (level 1) see also 4.7 and 4.12

**address control**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x10 | command station instruction |
| 3 | 10 | address control |
| 4 | cAdr_hi | address high byte<br>bit 6 + bit 7 format specifications (see 4.5) |
| 5 | cAdr_lo | address low byte |
| 6 | cControl | bits 7: 0 – address status query<br>1 – set address status according to bits 1-0<br>bit 5: 1: accessory address 0: loco address<br>bit 1: lock address when bit is set (no external changes)<br>bit 0: log external changes when bit is set |
| 7 | cOutputs | optional: for accessory decoder addresses: the outputs that shall be locked may be defined – keep in mind that there need to be 2 bits set for each paired output, if zero locking is deactivated |

***Reply:*** Ack (level 1) see also 4.7 and 4.13

**read command station I/O state**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x10 | command station instruction |

| | | |
|---|---|---|
| 3 | 11 | read command station |
| 4 | 0 | command station |

***Reply:*** Ack (level 1) see also 4.7 and 4.14

**read/set a command station CV**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x10 | command station instruction |
| 3 | 12 | read/set a command station CV |
| 4 | variable_hi | configuration variable number high byte that shall be read/set |
| 5 | variable_lo | configuration variable number low byte that shall be read/set |
| 6 | value | optional: if a value is present it will be set as new CV-value |

***Reply:*** Ack (level 1) ) see also 4.7 and 4.15

**command station equipment query**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x10 | command station instruction |
| 3 | 13 | command station equipment query |
| 4 | 0 | command station |

***Reply:*** Ack (level 1) see also 4.7 and 4.16

**serial info**

This command allows simple identification of the software connected to the command station. Furthermore it provides a way to inform the command station that the communication is about to stop and thus avoid a false broken connection error message.

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x10 | command station instruction |
| 3 | 17 | info from service tool |
| 4 | ToolID | 0 = default value (no start command necessary)<br>1 = ZST (Zimo Service Tool)<br>2 = FTC (Freiwald Train Controller)<br>3 = KAM (KAM Train Server) (Version 5) |
| | | 4 = ADA (Sperrer ADaPT) (Version 5)<br>5 = STP (Sperrer STP) (Version 5)<br>6 = ZIR (ZIRC) (Version 5) |
| 5 | Action | 0 communication end<br>1 communication start<br>2 communication timeout-refresh |

***Reply:*** Ack (level 1)

**read/set a decoder CV** **(Version 5)**

At this time the command station supports only CV read/set commands for DCC decoder addresses. If the given decoder address is 0 register programming on the programming track is performed, otherwise on the main programming is used

| header byte | value | Description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x10 | command station instruction |
| 3 | 19 | read/set a decoder CV |
| 4 | cAdr_hi | address high byte<br>bit 6 + bit 7 format specifications (see 4.5) |
| 5 | cAdr_lo | address low byte |
| 6 | variable_hi | configuration variable number high byte that shall be read/set |
| 7 | variable_lo | configuration variable number low byte that shall be read/set |
| 8 | value | optional: if a value is present it will be set as new CV-value |

***Reply:*** Ack (level 1) see also 4.7 and 4.17

**current loco memory**

If logging is activated for a loco address, this message is sent spontaneously whenever loco memory's state changes due to external input.

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x00 | command station message |
| 3 | 255 | current loco memory |
| 4 | cAdr_hi | address high byte<br>bit 6 + bit 7 format specification (see 4.5) |
| 5 | cAdr_lo | address low byte |
| 6 | cSpeed | speed step of the current speed step system |

| | | |
|---|---|---|
| | | (DCC data signal: 0-14, 0-28 or 0-126; according to bit 3/2 in cData1;MOTOROLA protocol: 0-14) |
| 7 | cData1 | bit 7: MAN (override signal controlled speed limit)<br>bit 6: no function<br>bit 5: direction (0 = forward, 1 = reverse)<br>bit 4 headlights on/off (= DCC function F0)<br>bit 3/2 DCC speed step system (see 4.6)<br>bit 1 decel. time "BZ" (definable by command "B") on/off<br>bit 0 accel. time "AZ" (definable by command "B") on/off |
| 8 | cData2 | bit 0-7: function outputs 1-8 on/off (DCC only) |
| 9 | cData3 | bit 0-3: function outputs 9-12 on/off (DCC only) |
| 10 | cAzBz | bit 0-3: BZ (0 - 15)<br>bit 4-7: AZ (0 - 15) |
| 11 | cStatus | 1 … address is active |

***Reply:*** Ack (level 1)

**current accessory decoder memory**

If logging is activated for an accessory decoder address, this message is sent spontaneously whenever accessory decoder memory's state changes due to external input.

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x00 | command station message |
| 3 | 254 | current accessory decoder memory |
| 5 | cAdr_hi | address high byte<br>bit 6 + bit 7 format specification (see 4.5) |
| 6 | cAdr_lo | address low byte |
| 7 | cPair | 0 … paired output function<br>1 … single output function |
| 8 | cOutputs | state of the 8 outputs |

***Reply:*** Ack (level 1)

**accessory module messages (MX8)**

**activate/deactivate spontaneous messages of an accessory module**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x11 | MX8 control instruction |
| 3 | 0 | activate / deactivate spontaneous messages |
| 4 | MX8_ID | MX8 address (0-63 not 800-863) |
| 5 | Output0 | The bits 0-7 represent the module outputs 0-7 |
| 6 | Output1 | The bits 0-7 represent the module outputs 8-15. |
| 7 | Output2 | The bits 0-7 represent the module outputs 16-23 |
| 8 | Output3 | The bits 0-7 represent the module outputs 24-31. The message forwarding is activated for all main sections if the corresponding bit is set, otherwise it is deactivated. The odd bits of paired outputs are meaningless. |

***Reply:*** Ack (level 1)

**activate selected spontaneous messages of an accessory module**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x11 | MX8 control instruction |
| 3 | 1 | activate additional spontaneous messages |
| 4 | MX8_ID | MX8 address (0-63 not 800-863) |
| 5 | Output0 | The bits 0-7 represent the module outputs 0-7 |
| 6 | Output1 | The bits 0-7 represent the module outputs 8-15. |
| 7 | Output2 | The bits 0-7 represent the module outputs 16-23 |
| 8 | Output3 | The bits 0-7 represent the module outputs 24-31. The message forwarding is activated for all main sections with their corresponding bit set, for all others the forwarding remains unchanged. The odd bits of paired outputs are meaningless. |

***Reply:*** Ack (level 1)

**deactivate selected spontaneous messages of an accessory module**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x11 | MX8 control instruction |
| 3 | 2 | deactivate additional spontaneous messages |
| 4 | MX8_ID | MX8 address (0-63 not 800-863) |
| 5 | Output0 | The bits 0-7 represent the module outputs 0-7 |
| 6 | Output1 | The bits 0-7 represent the module outputs 8-15. |
| 7 | Output2 | The bits 0-7 represent the module outputs 16-23 |
| 8 | Output3 | The bits 0-7 represent the module outputs 24-31. The message forwarding is deactivated for all main sections with their corresponding bit set, for all others the forwarding remains unchanged. The odd bits of paired outputs are meaningless. |

***Reply:*** Ack (level 1)

**set output of accessory module**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x11 | MX8 control instruction |
| 3 | 3 | set output of accessory module |
| 4 | MX8_ID | MX8 address (0-63 not 800-863) |
| 5 | cData | bit 0-5: number of module output<br>bit 6: position (0 = left, 1 = right)<br>bit 7 is not used |

***Reply:*** Ack (level 1)

**query accessory module outputs**

| header byte | value | Description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x11 | MX8 control instruction |
| 3 | 4 | query accessory module outputs |
| 4 | MX8_ID | MX8 address (0-63 not 800-863) |

***Reply:*** Ack (level 1), Ack (level 2) see also 4.18

**read/set a MX8-CV**

| header byte | value | Description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x11 | MX8 control instruction |
| 3 | 5 | read/set a MX8-CV |
| 4 | MX8_ID | MX8 address (0-63 not 800-863) |
| 5 | variable_hi | configuration variable number high byte that shall be read/set |
| 6 | variable_lo | configuration variable number low byte that shall be read/set |
| 7 | value | optional: if a value is present it will be set as new CV-value |

***Reply:*** Ack (level 1), Ack (level 2) see also 4.19

**current accessory module output state**

If spontaneous messages are activated for an accessory module's output, this message is sent spontaneously whenever it's state changes.

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x01 | MX8 message |
| 3 | 255 | current accessory module output state |
| 4 | MX8_ID | MX8 address (0-63 not 800-863) |
| 5 | cData | bit 0-5: module output number<br>bit 6: position (0 = left, 1 = right)<br>bit 7 paired output |
| 6 | cCommand | 0 state detected by MX8 (feedback)<br>1 new state sent to MX8 |

***Reply:*** Ack (level 1)

**track section module messages (MX9)**

It is possible to choose individual MX9 track sections for forwarding spontaneous status messages (occupancy and loco numbers) as well as to query the current state of the module's track sections.

**activate/deactivate spontaneous messages of a track section module**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x12 | MX9 control instruction |
| 3 | 0 | activate / deactivate spontaneous messages |

| | | |
|---|---|---|
| 4 | MX9_ID | MX9 address (0-63 not 900-963) |
| 5 | sections | The bits 0-7 represent the main sections 0-7. The message forwarding is activated for all main sections if the corresponding bit is set, otherwise it is deactivated. |

***Reply:*** Ack (level 1)

**activate selected spontaneous messages of a track section module**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x12 | MX9 control instruction |
| 3 | 1 | activate additional spontaneous messages |
| 4 | MX9_ID | MX9 address (0-63 not 900-963) |
| 5 | sections | The bits 0-7 represent the main sections 0-7. The message forwarding is activated for all main sections with their corresponding bit set, for all others the forwarding remains unchanged. |

***Reply:*** Ack (level 1)

**deactivate selected spontaneous messages of a track section module**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x12 | MX9 control instruction |
| 3 | 2 | deactivate additional spontaneous messages |
| 4 | MX9_ID | MX9 address (0-63 not 900-963) |
| 5 | sections | The bits 0-7 represent the main sections 0-7. The message forwarding is deactivated for all main sections with their corresponding bit set, for all others the forwarding remains unchanged. |

***Reply:*** Ack (level 1)

**read the track section module's occupancy state**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x12 | MX9 control instruction |
| 3 | 3 | read the track section module's occupancy state |
| 4 | MX9_ID | MX9 address (0-63 not 900-963) |

***Reply:*** Ack (level 1) ) see also 4.20

**read the track occupancy state and loco numbers of a main section**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x12 | MX9 control instruction |
| 3 | 4 | read the occupancy state and loco numbers of the section |
| 4 | MX9_ID | MX9 address (0-63 not 900-963) |
| 5 | section | main section number |

***Reply:*** Ack (level 1) see also 4.21

**set new speed limits of a main track section**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x12 | MX9 control instruction |
| 3 | 5 | set new speed limits to the section |
| 4 | MX9_ID | MX9 address (0-63 not 900-963) |
| 5 | section | main section number |
| 6 | Limit | speed limit (H=0, U=1, L=2, F=3, A=4, HU=5, UL=6, LF=7) |
| 7 | Limit2 | optional: apply speed limit when reaching section A (bit 7=1) or section B (bit 7=0); there is no speed limit change when reaching the second section if this value is omitted |

***Reply:*** Ack (level 1)

**set LED-outputs of a track section module**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x12 | MX9 control instruction |
| 3 | 6 | set LED-outputs |
| 4 | MX9_ID | MX9 address (0-63 not 900-963) |
| 5 | cGreen | settings for „green-LED's“ on connector “signals” |
| 6 | cRed | settings for „red-LED's“ on connector “signals” |
| 7 | cLedsA | optional: settings for „LED's-A“ on connector “occupancy indicators” |

| | | |
|---|---|---|
| 8 | cLedsB | settings for „LED's-B“ on connector “occupancy indicators”<br>Attention: after sending a message including the optional settings for the leds on connector “occupancy indicators”, these outputs are no longer controlled automatically by the MX9 and do not represent the occupancy status! |

***Reply:*** Ack (level 1)

**read/set a MX9-CV**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x12 | MX9 control instruction |
| 3 | 7 | read/set a MX9-CV |
| 4 | MX9_ID | MX9 address (0-63 not 900-963) |
| 5 | variable | configuration variable that shall be read/set |
| 6 | value | optional: if a value is present it will be set as new CV-value |

***Reply:*** Ack (level 1), Ack (level 2) see also 4.22

**main track section message**

If spontaneous messages are activated for a track section module's section, this message is sent spontaneously whenever it's state changes.

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x02 | MX9 message |
| 3 | 255 | main section status |
| 4 | MX9_ID | MX9 address (0-63 not 900-963) |
| 5 | Section | main section number (0-7) |
| 6 | B-Status | occupancy state of main section 0<br>V C S B A T T T<br>V: data valid<br>C: changed (since last transmission)<br>S: short circuit (main section off)<br>B: section B occupied<br>A: section A occupied<br>T T T: number of recognized loco numbers (0-4) |
| 7 | F-Status | current speed limit (H=0, U=1, L=2, F=3, A=4, HU=5, UL=6, LF=7) |
| 8 | Zugnr1_H | optional: 1. loco number high byte |
| 9 | ZugNr1_L | optional: 1. loco number low byte |
| 10 | ZugNr2_H | optional: 2. loco number high byte |
| 11 | ZugNr2_L | optional: 2. loco number low byte |
| 12 | ZugNr3_H | optional: 3. loco number high byte |
| 13 | ZugNr3_L | optional: 3. loco number low byte |
| 14 | ZugNr4_H | optional: 4. loco number high byte |
| 15 | ZugNr4_L | optional: 4. loco number low byte |

***Reply:*** Ack (level 1)

## 3.1.2. long frame protocol messages

**general messages**

**acknowledge/reply sent by command station (level 1)**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-id |
| 2 | 1100xxxx | acknowledgement/reply sent by command station (level 1) |
| 3 | XXX | same as primary message |
| 4 | dd00nnnn | nnnn … length of header<br>dd … additional data flow information (see 4.8) |
| 5 | ID | sequence-id of the message that is being acknowledged |

***Reply:*** n.a.

**acknowledge/reply sent by PC (level 1)**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-id |
| 2 | 1101xxxx | acknowledgement/reply sent by PC (level 1) |
| 3 | XXX | same as primary message |
| 4 | dd00nnnn | nnnn … length of header<br>dd … additional data flow information (see 4.8) |
| 5 | ID | sequence-id of the message that is being acknowledged |

***Reply:*** n.a.

**acknowledge/reply sent by MX1 (level 2)**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-id |

| | | |
|---|---|---|
| 2 | 1010xxxx | acknowledgement/reply sent by command station (level 2) |
| 3 | XXX | same as primary message |
| 4 | dd00nnnn | nnnn … length of header<br>dd … additional data flow information (see 4.8) |
| 5 | ID | sequence-id of the message that is being acknowledged |

***Reply:*** Ack (level 1)

**acknowledge/reply sent by PC (level 2)**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-id |
| 2 | 1011xxxx | acknowledgement/reply sent by PC (level 2) |
| 3 | XXX | same as primary message |
| 4 | dd00nnnn | nnnn … length of header<br>dd … additional data flow information (see 4.8) |
| 5 | ID | sequence-id of the message that is being acknowledged |

***Reply:*** Ack (level 1)

**display control of command station**

The characters that will be displayed are the data bytes following the header bytes and the command station stops to update the display. If there are no characters following the header data the display returns to normal.

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-id |
| 2 | 0x90 | command station instruction |
| 3 | 18 | display control of command station |
| 4 | 5 | length of header (5), no more frames |
| 5 | index | position of the 1. character (0-31) |

***Reply:*** Ack (level 1)

## 3.2. checksum

The checksum includes all bytes of the original raw data of the data frame (i.e. header info and data) but excluding the protecting escape characters and without XOR'ing with 0x20.

### 3.2.1. checksum8

Checksum8 is an 8 bit checksum as it is utilized by various chip manufacturers (with the polynomial $x^8 + x^5 + x^4 + 1$, bit reversion for data and remainder, initialized with 0xff)

### 3.2.2. checksum16

Checksum16 is a 16 bit checksum (CRC16/CITT: with the polynomial $x^{16} + x^{12} + x^5 + 1$, initialized with 0xffff). The high byte of the checksum is written first (big endian).

# 4. Protocol details

Each message has a unique sequence-ID. After sending the final EOT, an acknowledgment or reply is expected for a specific length of time. If no such acknowledgement or reply is received, the message will be sent again with the same sequence-ID. If the reply is a negative acknowledgement, the message is resent immediately. If a message with the same sequence-ID is received the appropriate reply will be re-sent, as the original reply was obviously lost – any additional actions will be omitted as they were fulfilled when the message was correctly received the first time. There are some rare cases of messages that are never to be acknowledged – for example the confirmation messages themselves (ACK,NAK).

In order to prevent the possibility of incorrect communication due to hardware related faults the confirmation messages of the command station and PC are different.

## 4.1. reset message

The reset message provides a synchronization of the sending and receiving station's (either one may be a PC or command station). After a reset, all buffers are cleared and the sequence ID's are reset. If the special acknowledgement for a reset message is received by the sender, everything is ok. If there is no reply or a negative acknowledgement the reset message has to be sent again just as any "normal" message would have to be too. In the rare case of a sequence ID conflict (the newly received reset message has the same sequence ID as the last (non reset) message) the answer would be a "normal" acknowledgement for the previous message. In this case the reset message has to be re-sent with the next sequence ID and thus resolve the conflict.

If the command station initiates a (spontaneous) reset of the serial communication, it will send a reset message.

## 4.2. negative acknowledgement

If a corrupt message is received, that is a message that fails the checksum test, a negative acknowledgement is sent. As it is impossible in these cases to determine whether the received sequence-ID of the incoming frame is correct, it has to be assumed as incorrect. Therefore a negative acknowledgement message cannot contain the sequence-ID of the message it's referring to – the sender must determine the erroneous message.

## 4.3. acknowledgement / reply - level 1

When a data frame is correctly received it is acknowledged with its unique sequence-ID. Additionally requested data may be included in the reply if it's immediately available in the command station –

data not available at this time will be sent separate via acknowledge/reply level 2. Alternatively, there may be a return code or data included to describe the result of the request.

## 4.4. acknowledgement / reply - level 2

A level 2 confirmation message is basically a normal message (including a reference to the primary message) that has to be acknowledged by the receiver.

## 4.5. address format specification

For messages:

| bit 7 | bit 6 | description |
|---|---|---|
| 0 | 0 | use protocol last used with this address |
| 1 | 0 | force DDC protocol |
| 0 | 1 | force Motorola protocol |
| 1 | 1 | reserved |

For replies:

| bit 7 | bit 6 | description |
|---|---|---|
| 0 | 0 | not used |
| 1 | 0 | address uses DDC protocol |
| 0 | 1 | address uses Motorola protocol |
| 1 | 1 | reserved |

For decoder CV access via register programming: (Version 5)

| bit 7 | bit 6 | description |
|---|---|---|
| 0 | 0 | not used |
| 1 | 0 | address uses DDC protocol |
| 0 | 1 | address uses Motorola protocol |
| 1 | 1 | Reserved |

For decoder CV access via on-the-main programming: (Version 5)

| bit 7 | bit 6 | description |
|---|---|---|
| 0 | 0 | Reserved |
| 0 | 1 | Reserved |
| 1 | 0 | decoder address uses DDC protocol |
| 1 | 1 | accessory decoder address uses DDC protocol |



## 4.6. speed step system

The speed system bits are only meaningful for loco addresses using the DCC protocol.

For messages:

| bit 3 | bit 2 | description |
|---|---|---|
| 0 | 0 | use speed system last used with this address |
| 0 | 1 | force 14 speed steps (0-14) |
| 1 | 0 | force 28 speed steps (0-28) |
| 1 | 1 | force 126 speed steps (0-126) |

For replies:

| bit 3 | bit 2 | Description |
|---|---|---|
| 0 | 0 | not used |
| 0 | 1 | address uses 14 speed steps (0-14) |
| 1 | 0 | address uses 28 speed steps (0-28) |
| 1 | 1 | address uses 126 speed steps (0-126) |

## 4.7. additional return code to a message

The reply to message 0x0A inherits an additional return code. This return code reflects the logical error-status of the sent message. If the message is transmitted correctly but there is a logical error within the message that prevents the command station from executing it, then the codes below indicate the reason for it.

| Return code | value | description |
|---|---|---|
| NO_ERROR | 0x00 | no error within the request |
| ERR_ADRESSE | 0x01 | not a valid address |
| ERR_INDEX | 0x02 | error with index of extended address |
| ERR_FORWARD | 0x03 | request couldn't be forwarded |
| ERR_BUSY | 0x04 | the command station is busy with an other accessory decoder command |
| ERR_NO_MOT | 0x05 | Motorola jumper off |
| ERR_NO_DCC | 0x06 | DCC jumper off |
| ERR_CV_ADRESSE | 0x07 | not a valid CV address |
| ERR_SECTION | 0x08 | not a valid section |
| ERR_NO_MODUL | 0x09 | module with given address doesn't exit |
| ERR_MESSAGE | 0x0a | error within message |
| ERR_SPEED | 0x0b | given speed not valid (Version 5) |

## 4.8. additional data flow information

The fourth header byte of long data frames contains, besides the length of the header (original length without counting escape character prefixes), additional data flow information. This information describes whether there will be any consecutive frames with data.

| flow information in dd | description |
|---|---|
| 00 | no more data to come |
| 01 | more data will follow |
| 10 | more data will possibly follow |

## 4.9. reply to message 0x10, codes 3, 4, 5, 6

In case there is no error within a request, the level 1 acknowledgement contains additional information. If the track state is not in normal operational mode bit 0 is set. In this case the actual track state can be determined via the track control message.

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x40 | reply level 1 – short frame |
| 3 | 3,4,5,6 | code of message being replied to |
| 4 | ID | sequence-ID of the message being replied to |
| 5 | 0 | no error within the request |
| 6 | ff00ss0t | ff used format specification (see 4.5)<br>ss used speed step system (see 4.6)<br>t track state (0: normal operation, 1: fault) |

## 4.10. reply to message 0x10, code 7

In case there is no error within the request, the level 1 acknowledgement contains additional information. If the track state is not in normal operational mode, bit 0 is set. In this case the actual track state can be determined via the track control message.

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x40 | reply level 1 – short frame |
| 3 | 7 | code of message being replied to |
| 4 | ID | sequence-ID of the message being replied to |
| 5 | 0 | no error within the request |
| 6 | ff00000t | ff used format specification (see 4.5)<br>t track state (0: normal mode, 1: fault) |

## 4.11. reply to message 0x10, code 8 (query command station's loco memory)

In case there is no error within the request, the level 1 acknowledgement contains the requested data.

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x40 | reply level 1 – short frame |
| 3 | 8 | code of message being replied to |
| 4 | ID | sequence-ID of the message being replied to |
| 5 | 0 | no error within the request |
| 6 | cAdr_hi | address high byte<br>bit 6 + bit 7 format specification (see 4.5) |
| 7 | cAdr_lo | address low byte |
| 8 | cSpeed | speed step of the current speed step system<br>(DCC data signal: 0-14, 0-28 or 0-126; according to bit 3/2 in cData1;MOTOROLA protocol: 0-14) |
| 9 | cData1 | bit 7: MAN (override signal controlled speed limit)<br>bit 6: no function<br>bit 5: direction (0 = forward, 1 = reverse)<br>bit 4 headlights on/off (= DCC function F0)<br>bit 3/2 DCC speed step system (see 4.6)<br>bit 1 decel. time "BZ" (definable by command "B") on/off<br>bit 0 accel. time "AZ" (definable by command "B") on/off |
| 10 | cData2 | bit 0-7: function outputs 1-8 on/off (DCC only) |
| 11 | cData3 | bit 0-3: function outputs 9-12 on/off (DCC only) |
| 12 | cAzBz | bit 0-3: BZ (0 - 15)<br>bit 4-7: AZ (0 - 15) |
| 13 | cStatus | 1 … address is active |

## 4.12. reply to message 0x10, code 9
**(query command station's accessory decoder memory)**

In case there is no error within the request, the level 1 acknowledgement contains the requested data.

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x40 | reply level 1 – short frame |
| 3 | 9 | code of message being replied to |
| 4 | ID | sequence-ID of the message being replied to |
| 5 | 0 | no error within the request |
| 6 | cAdr_hi | address high byte<br>bit 6 + bit 7 format specification (see 4.5) |
| 7 | cAdr_lo | address low byte |
| 8 | cPair | 0 … paired output function<br>1 … single output function |
| 9 | cOutputs | state of the 8 outputs |

## 4.13. reply to message 0x10, code 10 (address control)

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x40 | reply level 1 – short frame |
| 3 | 10 | code of message being replied to |
| 4 | ID | sequence-ID of the message being replied to |
| 5 | 0 | no error within the request |
| 6 | ffa0sskl | ff used format specification (see 4.5)<br>a 1: accessory address 0: loco address<br>ss used speed step system (see 4.6)<br>k lock address if bit is set (no external changes)<br>l log external changes if bit is set |
| 7 | cOutputs | bit mask of locked outputs of accessory address (only included if applicable) |

## 4.14. reply to message 0x10, code 11 (read command station I/O state)

Voltage values are given in 0.1 V. Current values are given in 0.01 A.

| header byte | value | Description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x40 | reply level 1 – short frame |
| 3 | 11 | code of message being replied to |
| 4 | ID | sequence-ID of the message being replied to |
| 5 | 0 | values for command station |
| 6 | cCurrent1_h | current output 1 - high byte |
| 7 | cCurrent1_l | current output 1 – low byte [0.01A] |
| 8 | cVoltage1 | voltage output 1 – [0.1V] |
| 9 | cCurrent2_h | current output 2 – high byte (0 for MX1_EC) |
| 10 | cCurrent2_l | current output 2 - low byte [0.01A] (0 for MX1_EC) |
| 11 | cVoltage2 | voltage output 2 - [0.1V] (0 for MX1_EC) |
| 12 | cAux | auxiliary inputs |

Current values with bit 15 set are special values:

| special value | decription |
|---|---|
| 0x8000 | no 2nd value (MX1EC) |
| 0x8001 | "OFF" |
| 0x8002 | "UEP" |
| 0x8003 | "UES" |
| 0x8004 | "AUS" |
| 0x8005 | "SSP" |
| 0x8006 | "No Si" |
| 0x8007 | "SL UES" |

## 4.15. reply to message 0x10, code 12 (read/set a CV of the command station)

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x40 | reply level 1 – short frame |
| 3 | 12 | code of message being replied to |
| 4 | ID | sequence-ID of the message being replied to |
| 5 | 0 | no error within the request |
| 6 | cValue | value of the CV |

### 4.16. reply to message 0x10, code 13 (command station equipment query)

This is a long frame level 1 reply. The returned data (following the header bytes) includes the following information:

| byte number | value | description |
|---|---|---|
| 1 | cAddress_h | unique (CAN-)address high byte |
| 2 | cAddress_l | unique (CAN-)address low byte |
| 3 | cDevice | device – ID |
| 4 | cRom_size | ROM size of command station [32k –pages] |
| 5 | cRam_size | RAM size of command station [32k –pages] |
| 6 | cPrintver_h | print version main number |
| 7 | cPrintver_l | print version sub-number |
| 8 | cVerison_h | sw-Version main number |
| 9 | cVersion_l | sw-Version sub-number |
| 10 | cDate_day | sw-date day |
| 11 | cDate_month | sw-date month |
| 12 | cDate_century | sw-date century |
| 13 | cDate_year | sw-date year |
| 14 | cSwitches | switches |
| 15 | cDevelopVersion | 00 Release or ASCII-pre-release version („“a“,“b“,..) |
| 16 | cBootRom_h | boot rom version main number |
| 17 | cBootRom_l | boot rom version sub-number |
| 18 | cBootRom_develop | 00 boot rom release or ASCII-pre-release version („“a“,“b“,..) |
| 19 | 0 | values for command station |

meaning of the device-ID:

| device – ID | device |
|---|---|
| 1 | MX1 2000 / HS |
| 2 | MX1 2000 EC |
| 3 | MX31 ZL |

### 4.17. reply to message 0x10, code 19 (read/set a decoder CV) (Version 5)

In a first step, the usual level 1 acknowledgment follows. The received data is forwarded as a level 2 reply with short frames:

| header byte | value | Description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x20 | reply level 2 – short frame |
| 3 | 19 | code of message being replied to |
| 4 | ID | sequence-ID of the message being replied to |
| 5 | cAdr_hi | address high byte<br>bit 6 + bit 7 format specification (see 4.5) |
| 6 | cAdr_lo | address low byte |
| 7 | variable_hi | configuration variable number high byte that shall be read/set |
| 8 | variable_lo | configuration variable number low byte that shall be read/set |
| 9 | cValue | value of the CV |

***Reply:*** Ack (level 1)

In cases where the command station receives no data from the specified decoder, the reply doesn't contain any data or there occurred some error

| header byte | value | Description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x20 | reply level 2 – short frame |
| 3 | 19 | code of message being replied to |
| 4 | ID | sequence-ID of the message being replied to |
| 5 | cAdr_hi | address high byte<br>bit 6 + bit 7 format specification (see 4.5) |
| 6 | cAdr_lo | address low byte |
| 7 | cError | error code |

***Reply:*** Ack (level 1)

A new read/set command of a decoder CV cannot be started until the previous query is finished. In case of an unfinished query, a level 1 reply is shown. If the source of the active query is a read/set command of a decoder CV additional information referring to the active query is included.

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x40 | reply level 1 – short frame |
| 3 | 19 | code of message being replied to |
| 4 | ID | sequence-ID of the message being replied to |
| 5 | 0x04 | busy – there is already an active request |
| 6 | cAdr_hi | address high byte (that couldn't be handled)<br>bit 6 + bit 7 format specification (see 4.5) |
| 7 | cAdr_lo | address low byte |

| | | |
|---|---|---|
| 8 | variable_hi | configuration variable number high byte that can't be read/set |
| 9 | variable_lo | configuration variable number low byte that can't be read/set |
| 10 | ID | optional: unique sequence-ID of the message that requested the query |
| 11 | cAdr_hi | optional: address high byte (that is currently being serviced)<br>bit 6 + bit 7 format specification (see 4.5) |
| 12 | cAdr_lo | optional: address low byte |
| 13 | variable_hi | optional: configuration variable number high byte that is read/set |
| 14 | variable_lo | optional: configuration variable number low byte that is read/set |

***Reply:*** n.a.

## 4.18. reply to message 0x11, code 4 (query accessory module outputs)

In a first step, the usual level 1 acknowledgment follows. The received data is forwarded as a level 2 reply with short frames. The data of the frame includes the following information:

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x21 | reply level 2 – short frame |
| 3 | 4 | code of message being replied to |
| 4 | ID | sequence-ID of the message that is being acknowledged |
| 5 | MX8_ID | MX8 address (0-63 not 800-863) |
| 6 | cTime0 | pulse-time 0 (CV 515) |
| 7 | cTime1 | pulse-time 1 (CV 516) |
| 8 | cTime2 | pulse-time 2 (CV 517) |
| 9 | cTime3 | pulse-time 3 (CV 518) |
| 10 | cPaired | number of paired outputs (0,8,16) |
| 11 | cOutput0 | the bits 0-7 represent the module outputs 0-7 |
| 12 | cOutput1 | the bits 0-7 represent the module outputs 8-15. |
| 13 | cOutput2 | the bits 0-7 represent the module outputs 16-23 |
| 14 | cOutput3 | the bits 0-7 represent the module outputs 24-31. For paired outputs (starting at output 0) the odd bits are meaningless. |

***Reply:*** Ack (level 1)

In cases where the command station receives no data from the specified accessory module or the reply doesn't contain any data

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x21 | reply level 2 – short frame |
| 3 | 4 | code of message being replied to |
| 4 | ID | sequence-ID of the message that is being acknowledged |
| 5 | MX8_ID | MX8 address (0-63 not 800-863) |

***Reply:*** Ack (level 1)

A new query of accessory module outputs cannot be started until the previous query is finished. In case there is an unfinished query, a level 1 reply is shown with additional information according to the active query.

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x41 | reply level 1 – short frame |
| 3 | 4 | code of message being replied to |
| 4 | ID | sequence-ID of the message being replied to |
| 5 | 0x04 | busy – there is already an active request |
| 6 | MX8_ID | MX8 address (0-63 not 800-863) that is blocked |
| 7 | ID | unique sequence-ID of the message that requested the query |
| 8 | MX8_ID | MX8 address (0-63 not 800-863) that is actually searched for |

***Reply:*** n.a.

### 4.19. reply to message 0x11, code 5 (read/set a MX8-CV)

In a first step, the usual level 1 acknowledgment follows. The received data is forwarded as a level 2 reply with short frames:

| header byte | value | Description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x21 | reply level 2 – short frame |
| 3 | 5 | code of message being replied to |
| 4 | ID | sequence-ID of the message being replied to |
| 5 | MX8_ID | MX8 address (0-63 not 800-863) |
| 6 | cValue | value of the CV |

***Reply:*** Ack (level 1)

In cases where the command station receives no data from the specified accessory module or the reply doesn't contain any data

| header byte | value | Description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x21 | reply level 2 – short frame |
| 3 | 5 | code of message being replied to |
| 4 | ID | sequence-ID of the message being replied to |
| 5 | MX8_ID | MX8 address (0-63 not 800-863) |

***Reply:*** Ack (level 1)

A new read/set a MX8-CV cannot be started until the previous query is finished. In case there is an unfinished query, a level 1 reply is shown with additional information according to the active query.

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x41 | reply level 1 – short frame |
| 3 | 5 | code of message being replied to |
| 4 | ID | sequence-ID of the message being replied to |
| 5 | 0x04 | busy – there is already an active request |
| 6 | MX8_ID | MX8 address (0-63 not 800-863) |
| 7 | ID | unique sequence-ID of the message that requested the query |
| 8 | MX8_ID | MX8 address (0-63 not 800-863) that is actually searched for |

***Reply:*** n.a.

### 4.20. reply to message 0x12, code 3

**(read the track section module's occupancy state)**

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x42 | reply level 1 – short frame |
| 3 | 3 | code of message being replied to |
| 4 | ID | sequence-ID of the message being replied to |
| 5 | 0 | No error within the request |
| 6 | MX9_ID | MX9 address (0-63 not 900-963) |
| 7 | B-Status0 | occupancy state of main section 0<br>V C S B A T T T<br>V: data valid<br>C: changed (since last transmission)<br>S: short circuit (main section off)<br>B: section B occupied<br>A: section A occupied<br>T T T: number of recognized loco numbers (0-4) |
| 8 | B-Status1 | occupancy state of main section 1 |
| 9 | B-Status2 | occupancy state of main section 2 |
| 10 | B-Status3 | occupancy state of main section 3 |
| 11 | B-Status4 | occupancy state of main section 4 |
| 12 | B-Status5 | occupancy state of main section 5 |
| 13 | B-Status6 | occupancy state of main section 6 |
| 14 | B-Status7 | occupancy state of main section 7 |

### 4.21. reply to message 0x12, code 4

**(read the track occupancy state and loco numbers of a main section)**

In case there was an error within the request, the error code will be sent within a short frame. If there was no error the requested data will be sent by a long frame as follows:

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0xC2 | reply level 1 – long frame |
| 3 | 4 | code of message being replied to |
| 4 | 5 | length of header |
| 5 | ID | sequence-ID of the message being replied to |
| 6 | MX9_ID | MX9 address (0-63 not 900-963) |
| 7 | Section | main section number (0-7) |
| 8 | B-Status | occupancy state of main section 0<br>V C S B A T T T<br>V: data valid |

| | | |
|---|---|---|
| | | C: changed (since last transmission)<br>S: short circuit (main section off)<br>B: section B occupied<br>A: section A occupied<br>T T T: number of recognized loco numbers (0-4) |
| 9 | F-Status | current speed limit (H=0, U=1, L=2, F=3, A=4, HU=5, UL=6, LF=7) |
| 10 | Zugnr1_H | optional: 1. loco number high byte |
| 11 | ZugNr1_L | optional: 1. loco number low byte |
| 12 | ZugNr2_H | optional: 2. loco number high byte |
| 13 | ZugNr2_L | optional: 2. loco number low byte |
| 14 | ZugNr3_H | optional: 3. loco number high byte |
| 15 | ZugNr3_L | optional: 3. loco number low byte |
| 16 | ZugNr4_H | optional: 4. loco number high byte |
| 17 | ZugNr4_L | optional: 4. loco number low byte |

***Reply:*** n.a.

### 4.22. reply to message 0x12, code 7 (read/set a MX9-CV)

In a first step, the usual level 1 acknowledgment follows. The received data is forwarded as a level 2 reply with short frames:

| header byte | value | Description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x22 | reply level 2 – short frame |
| 3 | 7 | code of message being replied to |
| 4 | ID | sequence-ID of the message being replied to |
| 5 | MX9_ID | MX9 address (0-63 not 900-963) |
| 6 | cValue | value of the CV |

***Reply:*** Ack (level 1)

In cases where the command station receives no data from the specified accessory module or the reply doesn't contain any data

| header byte | value | Description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x22 | reply level 2 – short frame |
| 3 | 7 | code of message being replied to |
| 4 | ID | sequence-ID of the message being replied to |
| 5 | MX9_ID | MX9 address (0-63 not 900-963) |

***Reply:*** Ack (level 1)

A new read/set a MX9-CV cannot be started until the previous query is finished. In case there is an unfinished query, a level 1 reply is shown with additional information according to the active query.

| header byte | value | description |
|---|---|---|
| 1 | 0-255 | unique sequence-ID |
| 2 | 0x42 | reply level 1 – short frame |
| 3 | 7 | code of message being replied to |
| 4 | ID | sequence-ID of the message being replied to |
| 5 | 0x04 | busy – there is already an active request |
| 6 | MX9_ID | MX9 address (0-63 not 900-963) |
| 7 | ID | unique sequence-ID of the message that requested the query |
| 8 | MX9_ID | MX9 address (0-63 not 800-863) that is actually searched for |