Lenz
ELEKTRONIK GMBH

documentation
XpressNet
version 4.0

02/2022

Contents

1 General

In addition to the hardware description of the XpressNet, this description also contains the commands that were used up to and including version 3.0 of the control centers, as well as future extensions, some of which are already provided in control centers or bus devices, but will only be used in later versions. The control centers are differentiated according to versions up to and including 1.5, from version 1.5 up to and including version 2.3, and from version 3.0 or higher. For later versions planned, the aim is to no longer support the locomotive driving commands and double traction up to and including version 2.3, in order to avoid making the protocol actually used in a control center unnecessarily large.

1.1 XpressNet hardware

The XpressNet is a serial bus according to the RS485 standard in half-duplex operation with differential signal transmission. The following are transmitted:

1 start bit (0), 9 data bits, 1 stop bit (1), no parity bit

Baud rate: 62.5 kbaud

Currently 32 bus participants (incl. central)

A parity bit is only inserted in the call byte from the central unit to a device. This is bit 7. Bit 7 is added so that there is an even number of ones in the call byte (even parity). The 9th bit is added on the hardware side as an address bit (1) or data bit (0). For 8051 derivatives: 9-bit UART, Mode 2 with address/data identification of the serial interface. Data transfer from an XpressNet device to the central unit always takes place in such a way that the 9th data bit is set to "0" by the device. The only exception is when data is transferred from one device to another.

Pin assignment of the XpressNet:

L – Supply of the bus participants (12VDC) M –

Ground

A – Receive/Transmit non-inverting B –

Receive/Transmit inverting

It is important to ensure that the devices are correctly connected to the A and B lines. If they are mixed up, no communication will be established. Due to the low data transfer rate of 62.5 kBaud, the connection between the participants can be up to 1000m long. It is also not necessary to adhere to a specific bus structure. For example, spur lines, star and ring wiring are possible. If there are transmission problems with very extensive wiring or under unfavorable environmental conditions, it may be necessary to lay the bus in twisted pairs and terminate it with a resistance of 120-. The LZ100/ LZV100/LZV200 control center already contains a terminating resistor. The cross-section of the cables should not be less than 0.25 mm^2; 0.5 mm^2 or larger is recommended.

1.2 Data traffic on the XpressNet bus

Communication is always initiated by the central unit. The central unit sends a call byte that contains the address of the slave (or all slaves) and the identifier (the content of the following bytes) of the call. The call byte is provided by the central unit with a parity bit that only refers to the call byte. If more than one byte is sent to a slave, an X-Or byte is sent at the end to check for errors in the transmission. The X-Or byte is made up of all data bytes without the call byte. Likewise, each slave sends an X-Or byte when sending to the central unit.

A slave can receive information (e.g. about a locomotive) from the central unit if it has requested it, but also if, for example, the status of this locomotive has changed due to actions by another slave (unsolicited information).

Basically, the slave can only carry out an action (e.g. requesting locomotive data) after being called by the central unit. After this action has been completed, the central unit addresses the next device.

It is also possible that a device does not send a request or a command to the central station after addressing the central station, but that the central station does not process the command due to the command identifier and instead forwards it to another device. This means that XpressNet devices can also exchange data with each other, for example by a device that evaluates RailCom receivers sending their information as a "RailCom-Info" broadcast to all XpressNet devices.

1.3 Communication and timing

Communication on the XpressNet takes place in such a way that the central unit, as the only master participant, cyclically addresses all connected devices (slaves) (e.g. with a "normal request") and these can then exchange messages with the central unit. Once such an exchange is complete, the next slave is addressed. Before the next device is addressed, the current command sequence is processed. This also includes, for example, the central unit processing an emergency stop command and then sending a broadcast to all participants and only then addressing further. This can take a long time, depending on the commands required by the slave (e.g. programming locomotive receivers).

In contrast to this question-and-answer procedure, the control center also sends the participants information that they have not explicitly requested, but to which they may have to respond (e.g. notification, locomotive takeover, etc.).

For error control, an X-Or byte is added to each data packet, which the receiver must check. After the central unit has addressed a device, the device must begin its response no later than 100 usec, but not earlier than 40 usec after the last bit has arrived. The time interval between two addressings of the same XpressNet device is, as guaranteed by the central unit, between 400usec and 500msec. An exception is the situation when the central unit is in programming mode. XpressNet devices must therefore be able to handle these addressing rates. The time variance results from the fact that, for example, only one device is connected to the bus, which is addressed without responding, and the maximum number of 31 devices, all of which send the longest command to the central unit, as well as broadcasts in between.

1.4 Unsolicited information

Unsolicited information is always sent to one or all slaves when system statuses need to be made known to all devices so that they can react correctly as quickly as possible. Unsolicited information is either sent as a broadcast if all slaves are to receive it, or formatted as a response if it only concerns a specific slave. It is always characteristic that a slave does not request this information (i.e. does not actually expect it), but still receives it at any time and must react correctly to inputs (locomotive driving commands), for example. The slave must therefore decide whether to evaluate the response or not accept the information at all, depending on its program context. Unsolicited information is:

broadcast "Everything On"	(to all participants)
broadcast "Alles Aus"	(to all participants)
Broadcast “All locomotives off”	(to all participants)
Broadcast “Programming mode”	(to all participants)
Broadcast “Feedback” Broadcast	(to all participants)
“RailCom info” broadcast	(to all participants)
"Modellzeit"	(to all participants)
answer "locomotive occupied"	(to the participant who had access to the locomotive at the time)
response "Double traction occupied"	(to the participant who had access to the double traction at the time)

The responses "Transmission error", "Central station busy", "Double traction error" and "Command not present" are not unsolicited information, as these responses can generally come in response to commands from a slave to the central station. They are therefore linked in time to the command to the central station.

1.5 Commands no longer supported

For reasons of clarity and data consistency, not all XpressNet commands are supported from control center version 3.0 onwards. This is due, on the one hand, to the fact that there are various control centers that have the software identifier 3.0 but cannot support the entire command set (identified by a control center ID) due to their hardware structure or processor performance, and, on the other hand, to the fact that, due to technical progress (more speed levels, more functions), the existing commands cannot be expanded. The LZ100 V3.x control center still contains the complete command set, but later control centers will no longer support the "old" locomotive queries and driving commands. Only the more comprehensive "new" commands should then be used. This applies to all commands that contain a locomotive address, as this has basically been expanded from 1 byte to 2 bytes in length.

This affects the following commands, which are no longer used for central units version 4.0 and higher:

1.) XpressNet device to central unit:

Stop locomotive up to LZ version 2.3 (see 3.6)

Stop multiple locomotives (see 3.8)

Request locomotive information (see 3.41.1 and 3.41.2)

Control locomotive (see 3.42.1 and 3.42.2)

double traction (see 3.42.8)

2.) Central unit to XpressNet device Central unit software version (see

2.7.1) Locomotive information (see 2.17 and 2.18) Double

traction information (see 2.22.1, 2.23 and 2.23.3)

If an XpressNet device still uses an “old” command, the control center will respond with the answer “Command unknown”.

2 commands from control center to device

Command structure:

The control center sends a call byte with the slave address identifier, a header byte, one to a maximum of 16 data bytes and an X-Or byte.

Exception: The calls “Acknowledgement”, “TBD” and “Normal Request” consist only of the call byte.

The number of data bytes that still follow is entered in the lower nibble of the header byte.

Agreements for the following command descriptions:

P = parity bit (including P there is an even number of ones in the call byte A =

device address (5 bits), whereby slaves can have the addresses 1 to 31. N =

number of subsequent data bytes (binary and hexadecimal)

GA = device address

The command format is specified in binary, decimal and hexadecimal.

2.1 Normal request

Format:

call byte

Binary :	P10A AAAA
Hex :	P+0x40+GA
Dec :	P+64+GA

Description:

In response to this request, a device can send commands to the central unit to request information or issue commands. The slave then begins central unit communication.

Special features:

No.

2.2 Acknowledgement

Format:

call byte

Binary :	P00A AAAA
Hex :	P+0x00+GA
Dec :	P+GA

Description:

After an incorrect transmission from a device to the control center, the control center sends the call "Transmission error" and then the call for acknowledgment. Cause: The X-Or byte was incorrect.

Procedure: Normal call from control center to device

XpressNet slave sends command

-> Error on the transmission path Central

sends “Transmission error” Central sends

call for “acknowledgement” XpressNet

slave sends acknowledgement Normal call

from central

.

.

Special features:

The call for acknowledgement must be answered. Otherwise, the control center will address this device with acknowledgement commands until it has acknowledged.

2.3 TBD

Format:

	call byte
Binary :	P01A AAAA
Hex :	P+0x20+GA
Dec :	P+32+GA

Description:

Reserved for later expansion.

Special features:

No

2.4 Broadcast

The "Broadcast" call group gives the control center the option of sending information to all slaves at the same time. The call byte contains the device address in binary form b00000, which every XpressNet participant must listen to by agreement. A broadcast is sent several times in succession (three times up to control center V2.3) to ensure that every participant can receive it. Some commands from devices to the control center trigger such a broadcast (e.g. "emergency stop"). A device that triggers a broadcast must ensure that no internal discrepancies arise if it receives this broadcast again immediately afterwards (e.g. if the control center has been set to programming mode).

2.4.1 *BC "Alles An"*

Format:

	call byte	header byte data 1		X-Or-Byte
Binary :	0110 0000	0110 0001	0000 0001	0110 0000
Hex :	0x60	0x61	0x01	0x60
Dec :	96	97	1	96

Description:

If an XpressNet slave sends the command "All On" (see section Device to Central), the broadcast "All On" is sent to inform all participants. This broadcast then corresponds to the actual system status. If, for example, an emergency stop is pending that cannot be cancelled and a slave sends "All On", the broadcast "All Off" is sent!

Special features:

This call is sent without a request from an XpressNet slave. It is unsolicited information. A slave may only change its (display) status when the broadcast arrives.

2.4.2 *BC "Everything Off" (emergency stop)*

Format:

	call byte	header byte	data 1	X-Or-Byte
Binary :	0110 0000	0110 0001	0000 0000	0110 0001
Hex :	0x60	0x61	0x00	0x61
Dec :	96	97	0	97

Description:

The control center sends the information that the track voltage has been switched off and therefore no more switching or driving commands can be sent.

Special features:

This call is sent without a request from an XpressNet slave. It is unsolicited information. A slave may only change its (display) status when the broadcast arrives, not when the command is sent.

2.4.3 *BC "All locomotives off" (emergency stop)*

Format:

	call byte	header byte	data 1	X-Or-Byte
Binary :	0110 0000	1000 0001	0000 0000	0110 0001
Hex :	0x60	0x81	0x00	0x81
Dec :	96	129	0	129

Description:

The control center sends the information that all locomotives on the track have been stopped via a broadcast (on the track side). The track voltage is still present so that switching commands can be sent, but no locomotives will be addressed until everything has been switched on again.

Special features:

This call is sent without a request from an XpressNet slave. It is unsolicited information. A slave may only change its (display) status when the broadcast arrives, not when the command is sent.

2.4.4 *BC "programming mode"*

Format:

	call byte	header byte	data 1	X-Or-Byte
Binary :	0110 0000	0110 0001	0000 0010	0110 0011

Hex :	0x60	0x61	0x02	0x63
Dec :	96	97	2	99

Description:

If this call is sent to all slaves, this is information that programming actions are now running. After that, no XpressNet slave will be addressed except the one that triggered the programming action (by sending a programming read command to the central unit). All other slaves must ensure that their program works correctly if they are not addressed for a relatively long time (display, watchdog, etc.). However, the slave that triggered the programming action must not fall into this waiting mode, even though it also receives the broadcast! The programming mode can be canceled again by the triggering device sending the "All On" command.

Special features:

This call is sent without a request from an XpressNet slave. It is unsolicited information. A slave may only change its (display) status when the broadcast arrives.

2.4.5 *BC "Feedback"*

Format:

	call byte	header byte data 1	Data 2	Data 3	Data 4	etc.	X-Or-Byte
Binary :	P010 0000	0100 NNNN	ADR_1	DAT_1	ADR_2	DAT2	etc.	X-Or-Byte
Hex :	0xa0	0x40 + N						X-Or-Byte
Dec :	160	64 + N						X-Or-Byte

Description:

With this call, the central unit informs all slaves that one or more feedback states have changed. The call is only sent if there are changes. In a broadcast, at least one address state and a maximum of 8 states are transmitted (one data byte per address, a total of 16 bytes per call without header and X-OR byte).

ADR_x and DAT_x have the format as described in 2.15.

For example, to correctly display the status of a feedback module, a device must examine the entire content of the broadcast for the desired address.

Special features:

This call is sent without a request from an XpressNet slave. It is unsolicited information to which a slave can respond according to its program context (e.g. feedback display). A slave must ensure that its program does not become too slow if a high bus load is created by a large number of feedbacks. This broadcast is also sent when a participant switches a switch that does not have feedback capability, in order to inform all others of this change. This ensures that all input devices can show the same display.

2.4.6 *BC "Feedback from switch 1025"*

Format:

	call byte	header byte data 1	Data 2	Data 3	X-Or-Byte
Binary :	1010 0000	0100 0011	ADRH	ADRL	DAT	X-Or-Byte

Hex :	0xa0	0x43				X-Or-Byte
Dec :	160	67				X-Or-Byte

Description:

With this call, the central unit informs all slaves that a feedback status above switch 1024 has changed. The call is only sent if there are changes. In a broadcast, only an address status is transmitted (address & data byte, 3 bytes per broadcast without header and X-OR byte).

ADR and DAT have the format as described in 2.15.

For example, to correctly display the status of a feedback module, a device must examine the entire content of the broadcast for the desired address.

Special features:

This call is sent without a request from an XpressNet slave. It is unsolicited information to which a slave can respond according to its program context (e.g. feedback display). A slave must ensure that its program does not become too slow if a high bus load is created by a large number of feedbacks. This broadcast is also sent when a participant switches a switch that does not have feedback capability, in order to inform all others of this change. This ensures that all input devices can show the same display.

2.4.7 *broadcast "Modellzeit"*

Format:

	call byte	header byte Kenn/D1	D2	D3	X-Or-byte
Binary :	0110 0000	0110 0011	0000 0011	dddh hhhh	s0mm mmmm	
Hex :	0x60	0x63	0x03			
Dec :	96	99	3			

Description:

If the model time clock is active in the control center (SV50 of the LZV200 has a value between 1 and 31), this broadcast is sent once per model minute on the XpressNet. Each connected device can then display the current model time or use it for other purposes. This call can therefore be used to synchronize the model time of XpressNet devices.

Data 2:	**dddh hhhh**	d	Specify the day of the week: 0 – Monday to 6 – Sunday
		h	Indication of the current hour, 0 – 23
Data 3:	**s0mm mmmm**	m	Indication of the minutes, 0 – 59
		s	0 = model clock stops, 1 = model clock runs

2.5 Programming information

After a programming read command has been issued, the control unit is put into programming mode. With a subsequent result read command, the control unit responds with one of the responses described here. If the control unit is not in programming mode and a result read command has been sent by a slave, the control unit sends "command not present" as a response.

2.5.1 *programming info "short circuit"*

Format:

	call byte	header byte	data 1	X-Or-Byte
Binary :	P11A AAAA	0110 0001	0001 0010	0111 0011
Hex :	P+0x60+GA	0x61	0x12	0x73
Dec :	P+96+GA	97	18	115

Description:

A short circuit or too high a current occurred when reading or writing to a receiver on the programming connection of the control center. It can be assumed that a write command to a memory location on the receiver did not write to it or wrote to it incorrectly. A device should then abort programming with an error message and, if data from the receiver is used internally, reset it to its original values.

Special features:

No.

2.5.2 *programming info "Data not found"*

Format:

	call byte	header byte	data 1	X-Or-Byte
Binary :	P11A AAAA	0110 0001	0001 0011	0111 0010
Hex :	P+0x60+GA	0x61	0x13	0x72
Dec :	P+96+GA	97	19	114

Description:

There is no receiver connected to the programming port of the control panel or the receiver is not responding to the control panel's reading attempt. Programming this receiver should be aborted or attempted again.

Special features:

No

2.5.3 *Programming info "Central Busy"*

Format:

	call byte	header byte	data 1	X-Or-Byte
Binary :	P11A AAAA	0110 0001	0001 1111	0111 1110
Hex :	P+0x60+GA	0x61	0x1F	0x7E
Dec :	P+96+GA	97	31	126

Description:

This command is not used up to and including control panel version 3.0.

Special features:

No.

2.5.4 *Programming info "Central Ready"*

Format:

	call byte	header byte data 1		X-Or-Byte
Binary :	P11A AAAA	0110 0001	0001 0001	0111 0000
Hex :	P+0x60+GA	0x61	0x11	0x70
Dec :	P+96+GA	97	17	112

Description:

This command is not used up to and including control panel version 3.0.

Special features:

No.

2.5.5 *programming info "Data 3-byte format"*

Format:

	call byte	header byte data 1		Data 2	Data 3	X-Or-Byte
Binary :	P11A AAAA	0110 0011	0001 0000	EEEE EEEE	DDDD DDDD	X-Or-Byte
Hex :	P+0x60+GA	0x63	0x10	E	D	X-Or-Byte
Dec :	P+96+GA	99	16	E	D	X-Or-Byte

Description:

This response is only given on request from the slave that put the central unit into programming mode. The EEPROM address (E) and the data read from it (D) are returned. This response is only given in register and page mode!

Special features:

The response refers to programming actions in register and page mode. However, if a receiver was requested to read CVs and this response was received, then the receiver cannot handle CV programming (old receiver). For further programming actions, the slave must now use write and read commands for register and page mode.

2.5.6 *Programming info “Data 4-byte format” CV 1-255 and CV1024 (new from version 3.6; replaces the same command up to version 3)*

Format:

	call byte	header byte data 1		Data 2	Data 3	X-Or-Byte
Binary :	P11A AAAA	0110 0011	0001 0100	CCCC CCCC	DDDD DDDD	X-Or-Byte
Hex :	P+0x60+GA	0x63	0x14	C	DAT	X-Or-Byte
Dec :	P+96+GA	99	20	C	DAT	X-Or-Byte

Description:

This response is only given on request from the slave that put the central unit into programming mode. The CV address (C) and the data read from it (D) are returned. This only applies to CV programming of receivers that support this mode.

Assignment value in C <=> CV – addresses:

C	CV
0	1024
1 ... 255	1 ... 255

Special features:

If a receiver is requested to read CVs and this response is received, everything is OK because the receiver can handle it. However, this response does not necessarily have to be returned when a CV request is made. A device must take this into account and then program the receiver in register or page mode. See 2.5.5.

2.5.7 *Programming info "Data 4-byte format" CV256 to CV511 (new from version 3.6)*

Format:

	call byte	header byte data 1		Data 2	Data 3	X-Or-Byte
Binary :	P11A AAAA	0110 0011	0001 0101	CCCC CCCC	DDDD DDDD	X-Or-Byte
Hex :	P+0x60+GA	0x63	0x15	C	DAT	X-Or-Byte
Dec :	P+96+GA	99	21	C	DAT	X-Or-Byte

Description:

This response is only given on request from the slave that put the central unit into programming mode. The CV address (C) and the data read from it (D) are returned. This only applies to CV programming of receivers that support this mode.

Assignment value in C <=> CV – addresses:

C	CV
0 ... 255	256 ... 511

Special features:

See 2.5.6

2.5.8 *Programming info "Data 4-byte format" CV512 to CV767 (new from version 3.6)*

Format:

	call byte	header byte data 1		Data 2	Data 3	X-Or-Byte
Binary :	P11A AAAA	0110 0011	0001 0110	CCCC CCCC	DDDD DDDD	X-Or-Byte

Hex :	P+0x60+GA	0x63	0x16	C	DAT	X-Or-Byte
Dec :	P+96+GA	99	22	C	DAT	X-Or-Byte

Description:

This response is only given on request from the slave that put the central unit into programming mode. The CV address (C) and the data read from it (D) are returned. This only applies to CV programming of receivers that support this mode.

Assignment value in C <=> CV – addresses:

C	CV
0 ... 255	512 ... 767

Special features:

See 2.5.6.

2.5.9 *Programming info "Data 4-byte format" CV768 to CV1023 (new from version 3.6)*

Format:

	call byte	header byte	data 1	Data 2	Data 3	X-Or-Byte
Binary :	P11A AAAA	0110 0011	0001 0111	CCCC CCCC	DDDD DDDD	X-Or-Byte
Hex :	P+0x60+GA	0x63	0x17	C	DAT	X-Or-Byte
Dec :	P+96+GA	99	23	C	DAT	X-Or-Byte

Description:

This response is only given on request from the slave that put the central unit into programming mode. The CV address (C) and the data read from it (D) are returned. This only applies to CV programming of receivers that support this mode.

Assignment value in C <=> CV – addresses:

C	CV
0 ... 255	768 ... 1023

Special features:

See 2.5.6.

2.6 Report Service Variable

Format:

	call byte	header byte	Kenn/D1	D2	D3	X-Or-byte
Binary :	P11A AAAA	0110 0011	0010 0000	SSSS SSSS	VVVV VVVV	

Hex :	P+0x60+GA	0x63	0x20			
Dec :	P+96+GA	99	32			

Description:

This is the response to the "Service Variable Read Command" (=>p. 51) command of an XpressNet slave. It is also sent when a service variable has been written.

Data 2: **SSSS SSSS** S Address of the service variable (SV)
(0 = SV256)

Data 3: **VVVV VVVV** V Content of the addressed SV

2.7 Software version central

Up to and including central unit version 2.3, a byte is sent back in response to the slave's request for the central unit software version. From version 3.0, a second byte is also sent, which contains the central unit ID. This allows a slave to decide, for example, which driving commands the central unit understands, whether multiple traction is possible, etc. The slave's request for the central unit software is identical to the response in both cases.

2.7.1 *software version up to LZ version 2.3*

Format:

	call byte	header byte data 1	Data 2	X-Or-Byte
Binary :	P11A AAAA	0110 0010	0010 0001	OOOO UUUU	X-Or-Byte
Hex :	P+0x60+GA	0x62	0x21	O + U	X-Or-Byte
Dec :	P+96+GA	98	33	O + U	X-Or-Byte

Description:

When you request the central software version, you receive this answer. The version number is hexadecimal coded in upper (OOOO) and lower (UUUU) nibbles. Example: Data 2 = 0010 0011 = 0x23 : Version 2.3

Special features:

No.

2.7.2 *Software version from LZ version 3.0*

Format:

	call byte	header byte data 1	Data 2	Data 3	X-Or-Byte
Binary :	P11A AAAA	0110 0011	0010 0001	OOOO UUUU	IIII IIII	X-Or-Byte
Hex :	P+0x60+GA	0x63	0x21	O + U	ID	X-Or-Byte
Dec :	P+96+GA	99	33	O + U	ID	X-Or-Byte

Description:

When you request the central software version, you will receive this answer. The version number is hexadecimal coded in upper (OOOO) and lower (UUUU) nibbles. Example: Data 2 = 0011 0000 = 0x30 : Version 3.0.

In addition, the central unit ID is sent, which has the following meaning: ID =

0x00: LZ 100 – Central

ID = 0x01: LH 200 – Central

ID = 0x02: DPC – Central (Compact and Commander)

Special features:

No.

2.8 Status Central

Format:

	call byte	header byte data 1	Data 2	X-Or-Byte
Binary :	P11A AAAA	0110 0010	0010 0010	SSSS SSSS	X-Or-Byte
Hex :	P+0x60+GA	0x62	0x22	S	X-Or-Byte
Dec :	P+96+GA	98	34	S	X-Or-Byte

Description:

When you request the central status, you get the status byte back. This byte is coded bit by bit as follows:

Bit 0: if 1, system in emergency stop if

Bit 1: 1, system in emergency off

Bit 2: Central start mode (0 = manual start, 1 = automatic start) Auto-start: All locomotives start immediately with their settings Manual start: All locomotives have speed 0 and functions off.

Bit 3: if 1, then programming mode active

Bit 4: reserved

Bit 5: reserved

Bit 6: if 1, then cold start in the control center

Bit 7: if 1, then RAM check error in the control center

Special features:

Not all bits are present in all control centers. If bit 6 and bit 2 are set, the control center does not yet put any data onto the track. The track output only begins when a device sets the start mode to manual or automatic. Before this, the control center sends the "All on" broadcast. So if a device determines "cold start" and "start mode auto", it should only continue in its context (e.g. for controlling locomotives) when it has sent the desired start mode to the control center or an "All on" broadcast has been received. In this case, another XpressNet device has set the start mode. It is therefore sensible to first query the control center status after a device reset and only then locomotive data, etc.

2.9 Report extended version information

Format:

	call byte	header byte Kenn/D1	D2	D3	D4	D5	D6	D7	X-Or-byte
Binary :	P11A AAAA	0110 0111	0010 0011							

Hex :	P+0x60+GA	0x67	0x23	ZBldH	ZBldL	RMVer	RMBldH	RMBldL	BIvers	
Dec :	P+96+GA	103	35							

Description:

This is the response to the command "Query extended central version information" (=>p. 50) of an XpressNet slave.

Data 2: ZBldH Build number of the central unit, high byte in hex

Data 3: ZBldL Build number of the central unit, low byte in hex

Data 4: RMVer Version number of the feedback controller in dec; one decimal place Build

Data 5 RMBldH number of the feedback controller, high byte in hex

Data 6 RMBldL Build number of the feedback controller, low byte in hex

Data 7 BIvers Version of the bootloader in dec; one decimal place

2.10 Report PoM Result

Format:

	call byte	header byte Kenn/D1	D2	D3	D4	X-Or-byte
Binary :	P11A AAAA	0110 0100	0010 0100				
Hex :	P+0x60+GA	0x64	0x24	ADRH	ADRL	PoMVal	XOR
Dec :	P+96+GA	100	36				

Description:

This is the response to the command "Get PoM result", section 3.30, page 52 **Data**

2: ADRH Locomotive address, high byte in hex

Data 3: ADRL Locomotive address, low byte in hex

Data 4: PoMVal Contents of the CV (0-255)

If the address 0 is reported, this means “read error” or “CV not present”

2.11 Report model time

Format:

	call byte	header byte Kenn/D1	D2	D3	D4	X-Or-byte
Binary :	P11A AAAA	0110 0100	0010 0101	DDDh hhhh	00mm mmmm	000f ffff	
Hex :	P+0x60+GA	0x64	0x25	DOW+h	min	factor	XOR
Dec :	P+96+GA	100	37				

Description:

This response provides the model time to the XpressNet participant that sent the "Request model time" command (section 3.33, page 53) to the central station.

Data 2:	**DDDh hhhh**	DDD: Day of the week (DOW):	0 = Monday 1 = Tuesday ... 6 = Sunday 7 not allowed!
		h hhhh: Hour:	0 - 23
Data 3	**min**	minutes:	0 - 59
Data 4	**factor**	time factor model time	0 – 31, 0=Stop (clock stopped)

2.12 Transmission errors

Format:

	call byte	header byte data 1		X-Or-Byte
Binary :	P11A AAAA	0110 0001	1000 0000	1110 0001
Hex :	P+0x60+GA	0x61	0x80	0xE1
Dec :	P+96+GA	97	128	225

Description:

The central unit sends a "transmission error" if the command received from the slave was not received correctly. The central unit immediately addresses the slave in question again with a request for acknowledgment, which it must answer. A new "normal request" is then sent to this slave so that the action can be repeated. Only then is the next slave addressed. If the slave does not respond to the request for acknowledgment with an acknowledgment, this slave is repeatedly addressed by the central unit with a request for acknowledgment. If a slave does not respond correctly to the request for acknowledgment, the next slave is addressed.

Special features:

No.

2.13 Central Busy

Format:

	call byte	header byte data 1		X-Or-Byte
Binary :	P11A AAAA	0110 0001	1000 0001	1110 0000
Hex :	P+0x60+GA	0x61	0x81	0xE0
Dec :	P+96+GA	97	129	224

Description:

The central unit sends a "Busy" command to a slave in response to this command if it cannot currently execute this command. The slave in question is then only addressed again later. The slave must decide whether or not to repeat the command that encountered "central unit busy" the next time it is addressed.

Special features:

No.

2.14 Command not available in control center

Format:

	call byte	header byte data 1		X-Or-Byte
Binary :	P11A AAAA	0110 0001	1000 0010	1110 0011
Hex :	P+0x60+GA	0x61	0x82	0xE3
Dec :	P+96+GA	97	130	227

Description:

If a command was transmitted correctly but is not included in the command set of the control center, the control center sends this response back. The same applies if commands are not possible from the current context (reading the programming result without the control center being in programming mode).

Special features:

No.

2.15 Switching information

Format:

	call byte	header byte data 1		Data 2	X-Or-Byte
Binary :	P11A AAAA	0100 0010	AAAA AAAA	ITTN ZZZZ	X-Or-Byte
Hex :	P+0x60+GA	0x42	ADR	ITNZ	X-Or-Byte
Dec :	P+96+GA	66	ADR	ITNZ	X-Or-Byte

Description:

The control center sends this information in response to a request for switching/feedback information. The information may include the status of switches that can or cannot provide feedback, or the status of a feedback receiver. In detail, it means:

Data 1: **AAAA AAAA** For a switch, data 1 is the address 1 divided by 4 of a switch from the value range 0..255. Switch 1 is therefore address 0

For central units lower than version 3.0, data 1 has the value 0 to 63 = 6 bits. If ADR = 0x00, for example, you have information about the switches 0, 1, 2 or 3, ie about the switch group 0 (if the identification bits TT identify a switching receiver).

For control units from version 3.0 onwards, all 8 bits of data 1 are permitted as a group address. This means that 256*4=1024 switches can be queried and switched.

For a feedback block, the address can be in the range 0..127 (7 bit address). This is the direct address of the block.
The address information is returned in this info as it was sent in the request for switching information from the control center.

Data 2: **I** If the bit = 1, this means that the switching command is still being executed and the switch has not yet reached its end position. Not defined for feedback modules, since their inputs are always 0 or 1 and cannot assume an intermediate state.

Data 2: **TT** These two bits represent the identifier of the requested address. The following applies:

TT = 0 0 Address is switching receiver without feedback

	TT = 0 1	Address is switching receiver with feedback
	TT = 1 0	Address is a feedback module
	TT = 1 1	Reserved for future use
Data 2:	**N**	This is the identifier of which nibble of a switch or feedback module it is.
	N = 0	Corresponds to the lower nibble.
	N = 1	Corresponds to the upper nibble.
		For example, for switch group 0, the lower nibble represents the state of switches 0 and 1 in the 4 state bits Z.
		The upper nibble represents the state of switches 2 and 3 in the 4 state bits Z.
		For a feedback receiver, the lower nibble represents the state of the lower 4 inputs in the 4 status bits Z, the upper nibble represents the state of the upper 4 inputs in the 4 status bits Z. In order to capture all 8 inputs of a feedback module, a request must be sent to the lower nibble and a second request to the upper nibble of the feedback address.
		Attention: The nibble bit is only correct if the switch has already been switched once!
Data 3:	**Z3 Z2 Z1 Z0**	The following applies to the state of a switching receiver:
	Z1 and Z0	represent the state of the first switch (e.g. switch no. 0 in switch group 0, nibble = 0) in the nibble.
	Z3 and Z2	represent the state of the second switch in the nibble (e.g. switch no. 3 in switch group 0, nibble = 1).
		Possible combinations:

Z1	**Z0**	first switch in the nibble
0	**0**	Switch has not been switched since the control center started or, in the case of switches with feedback capability, no input for the end position is connected.
0	**1**	The last switching command was “0”, the switch is on the left (this is of course only relative).
1	**0**	The last switching command was “1”, the switch is in the other end position (e.g. right, relative).
1	**1**	Invalid combination if both limit switches of a feedback-capable switch are active.->Wiring error?
Z3	**Z2**	second switch in the nibble. Validity of the bits as for the first switch in the nibble

For a feedback receiver, the 4 bits Z3..Z0 represent the state of the 4 inputs of the requested nibble.

Special features:

No.

2.16 Switching information from switch 1025

Format:

	call byte	header byte data 1	Data 2	Data 3	X-Or-Byte
Binary :	P11A AAAA	0100 0011	0000 000A	AAAA AAAA	ITTNZZZZ	X-Or-Byte
Hex :	P+0x60+GA	0x43	ADRH	ADRL	ITNZ	X-Or-Byte
Dec :	P+96+GA	67	ADRH	ADRL	ITNZ	X-Or-Byte

Description:

This response from the control center is functionally equivalent to the response "switching information", a second address byte is added. Here too, address = (switch address-1)/4; switch 1 is therefore address 0

Data 3 corresponds to the byte "Data 2" from 2.15, "switching information"

2.17 Locomotive information up to LZ version 1.5

This locomotive information is obtained by sending a locomotive request to the central unit as described in 3.41.1. Up to and including version 1.5 of the LZ100 central unit, only 14 speed steps were used (ie no "ModSel" byte). The information "locomotive occupied" is sent without being asked to the device that was just driving this locomotive, so that this device should have a corresponding routine to alert the user that this locomotive has just been taken over by another device.

2.17.1 *Locomotive free until LZ version 1.5*

If the locomotive is not yet in use by any other XpressNet device, you will receive this locomotive information with the header byte “Locomotive free” upon request.

Format:

	call byte	header byte data 1	Data 2	Data 3	X-Or-Byte
Binary :	P11A AAAA	1000 0011	locomotive address	locomotive data 1	locomotive data 2	X-Or-Byte
Hex :	P+0x60+GA	0x83	locomotive address	locomotive data 1	locomotive data 2	X-Or-Byte
Dec :	P+96+GA	131	locomotive address	locomotive data 1	locomotive data 2	X-Or-Byte

Description:

Locomotive address: has the value range 0..99 dec. = 0..0x63. The address 0 designates the conventional locomotive.

Locomotive data 1:

Bit 7 = 1: planned: Locomotive is called by another device

Bit 7 = 0: planned: Locomotive is not called by any other device.

Bit 7 is currently not used, so only the header byte can be used to determine whether the locomotive is being driven freely or by another device!

Bit 6 = 1 forward direction of travel

Bit 6 = 0: Reverse direction Function

Bit 5 = 1: 0 is switched on Function 0

Bit 5 = 0: is switched off

Bit 4: not used, is always returned as 0.

Bit 3 to Bit 0 indicate the gear level coded as follows:

3	2	1	0	
0	0	0	0	gear 0
0	0	0	1	Locomotive-specific emergency stop. The locomotive stops immediately without the set delay.
0	0	1	0	gear level 1
.	.	.	.	etc. until
1	1	1	1	gear level 14

Locomotive data 2: Bit7 to Bit4 : undefined state

Bit3: State of function 4, “0” = Off, “1” = On State

Bit2: of function 3, “0” = Off, “1” = On State of

Bit1: function 2, “0” = Off, “1” = On State of

Bit0: function 1, “0” = Off, “1” = On

Special features:

No.

2.17.2 *Locomotive occupied up to LZ version 1.5*

If the requested locomotive is in use by another XpressNet device, you will receive the locomotive information with the header byte "Locomotive occupied". However, you will also receive this information without being asked if another XpressNet device takes over this locomotive.

Format:

	call byte	header byte	data 1	Data 2	Data 3	X-Or-Byte
Binary :	P11A AAAA	1010 0011	locomotive address	locomotive data 1	locomotive data 2	X-Or-Byte
Hex :	P+0x60+GA	0xA3	locomotive address	locomotive data 1	locomotive data 2	X-Or-Byte
Dec :	P+96+GA	163	locomotive address	locomotive data 1	locomotive data 2	X-Or-Byte

Description:

Locomotive address, locomotive data 1 and locomotive data 2 have the format as described in 2.17.1.

Special features:

An XpressNet device can also receive the information “locomotive occupied” without being asked.

2.18 Locomotive information up to LZ version 2.3

This locomotive information is obtained by sending a locomotive request to the central unit as described in 3.41.2. As of version 2.0 of the central unit, 27 and 28 speed steps for a locomotive were supported in addition to 14. For this purpose, the previous locomotive request command was expanded to include a byte "ModSel" (mode select). This byte only contains the coding of the number of speed steps for the requested locomotive.

The information “locomotive occupied” is sent without being asked to the device that was just driving this locomotive, so that this device should have a corresponding routine to alert the user that this locomotive has just been taken over by another device.

2.18.1 *Locomotive free until LZ version 2.3*

If the locomotive is not yet in use by any other XpressNet device, the following locomotive information with the header byte “Locomotive free” is received upon request.

Format:

	call byte	header byte data 1	Data 2	Data 3	Data 4	X-Or-Byte	
Binary :	P11A AAAA	1000 0100	locomotive address	locomotive data 1	locomotive data 2	ModSel	X-Or-Byte
Hex :	P+0x60+GA	0x84	locomotive address	locomotive data 1	locomotive data 2	ModSel	X-Or-Byte
Dec :	P+96+GA	132	locomotive address	locomotive data 1	locomotive data 2	ModSel	X-Or-Byte

Description:

Locomotive address: has the value range 0..99 dec. = 0..0x63. The address 0 designates the conventional locomotive.

ModSel: Contains the coding of the number of speed steps of the locomotive: Depending on the content of ModSel, the bytes Locomotive Data 1 and Locomotive Data 2 have a different meaning.

Bit7 to Bit2: not used.

Bit1	Bit0	
0	0	14 gears
0	1	27 gears
1	0	28 gears
1	1	reserved combination

ModSel = xxxxxx00 **14-speed mode**

Locomotive data 1 and locomotive data 2 are coded as described in 2.17.1.

ModSel = xxxxxx01 **27 gear mode**

Locomotive data 1: is coded in 27-speed-step mode as follows: planned:

Bit 7 = 1: locomotive is called by another device planned:

Bit 7 = 0: locomotive is not called by any other device.

Bit 7 is currently not used, so only the header byte can be used to determine whether the locomotive is being driven freely or by another device!

Bit 6 = 1 forward direction of travel

Bit 6 = 0: Reverse direction Function

Bit 5 = 1: 0 is switched on Function 0

Bit 5 = 0: is switched off

Bit4 to Bit0 indicate the gear level coded as follows: Note that theBit4 that **LSB**the driving level.

3	2	1	0	**4!**	
0	0	0	0	0	gear 0
0	0	0	0	1	not used!
0	0	0	1	0	Locomotive-specific emergency stop. The locomotive stops immediately without the set delay.

0	0	0	1	1	not used!
0	0	1	0	0	gear level 1
0	0	1	0	1	gear 2
0	0	1	1	0	gear 3
.	.	.	.	.	
1	1	1	1	0	gear level 27

Locomotive data 2: Is coded as described in 2.17.1.

ModSel = **xxxxxx10** **28 gear mode**

Locomotive data 1: is coded in 28-speed step mode as follows: planned:

Bit 7 = 1: locomotive is called by another device planned:

Bit 7 = 0: locomotive is not called by any other device.

Bit 7 is currently not used, so only the header byte can be used to determine whether the locomotive is being driven freely or by another device!

Bit 6 = 1 forward direction of travel

Bit 6 = 0: Reverse direction Function

Bit 5 = 1: 0 is switched on Function 0

Bit 5 = 0: is switched off

Bit4 to Bit0 indicate the gear level coded as follows: Note that theBit4 that **LSB**the driving level.

3	2	1	0	**4!**	
0	0	0	0	0	gear 0
0	0	0	0	1	not used!
0	0	0	1	0	Locomotive-specific emergency stop. The locomotive stops immediately without the set delay.
0	0	0	1	1	not used!
0	0	1	0	0	gear level 1
0	0	1	0	1	gear 2
0	0	1	1	0	gear 3
.	.	.	.	.	
1	1	1	1	0	gear level 27
1	1	1	1	1	gear level 28

Locomotive data 2: Is coded as described in 2.17.1.

Special features:

No.

2.18.2 *Locomotive occupied up to LZ version 2.3*

If the requested locomotive is in use by another XpressNet device, you will receive the locomotive information with the header byte "Locomotive occupied". However, you will also receive this information without being asked if another XpressNet device takes over this locomotive.

Format:

	call byte	header byte data 1	Data 2	Data 3	Data 4	Data 5	X-Or-Byte
Binary :	P11A AAAA	1010 0100	locomotive address	locomotive data 1	locomotive data 2	ModSel	X-Or-Byte
Hex :	P+0x60+GA	0xA4	locomotive address	locomotive data 1	locomotive data 2	ModSel	X-Or-Byte
Dec :	P+96+GA	164	locomotive address	locomotive data 1	Lokdatne 2	ModSel	X-Or-Byte

Description:

Locomotive address, locomotive data 1 and locomotive data 2 have the format as described in 2.18.1.

Special features:

An XpressNet device can also receive the information “locomotive occupied” without being asked.

2.19 Locomotive information from central control unit version 3.0

In response to a general locomotive request in the format for version 3.0 or higher (see 3.41.3), one of the four responses described below can be received. In contrast to earlier versions, the "busy" information is included here. This means that the responses described here do not come as unsolicited information. If a locomotive is taken over by another XpressNet device, this is now communicated via the unsolicited information "locomotive busy" (see 2.5). An additional identification byte is also inserted after the header byte, which is used to distinguish between the various commands from central control unit version 3.0.

2.19.1 *Locomotive information normal locomotive*

This response is always sent if the requested locomotive is not in a multiple/double traction and is not the base address of a multiple traction.

Format:

	call byte	header byte identifier	Data 1	Data 2	Data 3	Data 4	X-Or-Byte
Binary :	P11A AAAA	1110 0100	0000 BFFF	RVVV VVVV	000F FFFF	FFFF FFFF	X-Or-Byte
Hex :	P+0x60+GA	0xE4	identifier	speed	F0	F1	X-Or-Byte
Dec :	P+96+GA	228	identifier	speed	F0	F1	X-Or-Byte

Description:

Identifier: Bit 3: B=0: Lok is free

Bit 3: B=1: Locomotive is called on another device (occupied)

Bit2 to Bit0: Identification of the speed step number

Bit2	Bit1	Bit0	
0	0	0	14 gears
0	0	1	27 gears
0	1	0	28 gears
1	0	0	128 gears

Data 1 **Speed:** Coding of speed and direction. forward

R=1:

R=0: backward

VVVVVVV

With 14 gears: Coding of bits 3,2,1,0 for the speed as described in 2.17.1.

With 27 gears: Coding of bits 4,3,2,1,0 for the speed as described under 2.18.1, ModSel = xxxxxx01

With 28 gears: Coding of bits 4,3,2,1,0 for the speed as described in 2.18.1, ModSel = xxxxxx10.

With 128 gears:

B6	B5	B4	B3	B2	B1	B0	
0	0	0	0	0	0	0	gear 0
0	0	0	0	0	0	1	emergency stop
0	0	0	0	0	1	0	gear level 1
0	0	0	0	0	1	1	gear 2
.	.	.	.	.	.	.	
1	1	1	1	1	1	1	gear level 126

Data 2: F0: State of functions 0 to 4.

7	6	5	4	3	2	1	0	bit
0	0	0	F0	F4	F3	F2	F1	

A 1 means "function is on". F1: Status of

Data 3: functions 5 to 12.

7	6	5	4	3	2	1	0	bit
F12	F11	F10	F9	F8	F7	F6	F5	

A 1 means "function is on".

Special features:

No.

2.19.2 *Functional status F13 to F28 of the requested locomotive (from version 3.6)*

	call byte	header byte identifier		Data 1	Data 2	X-Or-Byte
Binary :	P11A AAAA	1110 0011	0101 0010	FFFF FFFF	FFFF FFFF	X-Or-Byte
Hex :	P+0x60+GA	0xE3	0x52			X-Or-Byte
Dec :	P+96+GA	227	82			X-Or-Byte

Description:

Data 1: F: State of functions 13 to 20. Fx=1 means "function is on".

7	6	5	4	3	2	1	0	bit
F20	F19	F18	F17	F16	F15	F14	F13	

Data 2: F: State of functions 21 to 28. Fx=1 means "function is on".

7	6	5	4	3	2	1	0	bit
F28	F27	F26	F25	F24	F23	F22	F20	

Special features:

No.

2.19.3 *Functional status F29 to F68 of the requested locomotive (from version 4.0)*

	call byte	header byte identifier		Data 1	Data 2	Data 3	Data 4	Data 5	X-Or-Byte
Binary :	P11A AAAA	1110 0110	0101 0011	ffff ffff	ffff ffff	ffff ffff	ffff ffff	ffff ffff	X-Or-Byte
Hex :	P+0x60+GA	0xE6	0x53						X-Or-Byte
Dec :	P+96+GA	230	83						X-Or-Byte

Description:

Data 1: F: State of functions 29 to 36. FX=1 means "function is on".

7	6	5	4	3	2	1	0	bit
F36	F35	F34	F33	F32	F31	F30	F29	

Data 2: F: State of functions 37 to 44. FX=1 means "function is on".

7	6	5	4	3	2	1	0	bit
S44	S43	S42	S41	S40	S39	S38	S37	

Data 3: F: State of functions 45 to 52. FX=1 means "function is on".

7	6	5	4	3	2	1	0	bit
S52	S51	S50	S49	S48	S47	S46	S45	

Data 4: F: State of functions 53 to 60. FX=1 means "function is on".

7	6	5	4	3	2	1	0	bit
S60	S59	S58	S57	S56	S55	S54	S53	

Data 5: F: State of functions 61 to 68. FX=1 means "function is on".

7	6	5	4	3	2	1	0	bit
S68	S67	S66	S65	S64	S63	S62	S63	

Special features:

No.

2.19.4 *Locomotive information Locomotive is in a multiple traction*

Format:

	call byte	header byte identifier		Data 1	Data 2	Data 3	Data 4	X-Or-Byte
Binary :	P11A AAAA	1110 0101	0001 BFFF	RVVV VVVV	000F FFFF	FFFF FFFF	MTR	X-Or-Byte
Hex :	P+0x60+GA	0xE5	identifier	speed	F0	F1	MTR	X-Or-Byte
Dec :	P+96+GA	229	identifier	speed	F0	F1	MTR	X-Or-Byte

Description:

Identifier: bits 3 to 0 are coded as described in 2.19.1.

The speed step number indicates the speed step number of the requested locomotive! This can be different from the speed step number of the multiple traction (MTR) in which the locomotive is located.

Data 1: **Speed:** The speed byte is encoded as described in 2.19.1.

The speed indicates the speed of the requested locomotive!

Data 2: F0 Coded as described in 2.1.14.1. Coded

Data 3: **F1** as described in 2.1.14.1.

Data 4 **MTR** This is the multiple traction address of the requested locomotive.

Special features:

No.

2.19.5 *Locomotive information Locomotive address is the base address of a multiple traction*

Format:

	call byte	header byte identifier		Data 1	X-Or-Byte
Binary :	P11A AAAA	1110 0010	0010 BFFF	RVVV VVVV	X-Or-Byte
Hex :	P+0x60+GA	0xE2	identifier	speed	X-Or-Byte
Dec :	P+96+GA	226	identifier	speed	X-Or-Byte

Description:

Identifier: bits 3 to 0 are coded as described in 2.19.1.

The number of speed steps indicates the number of speed steps of the multiple traction.

Data 1: **Speed:** The speed byte is encoded as described in 2.19.1.

The speed indicates the speed of the MTR!

Special features:

No.

2.19.6 *Locomotive information Locomotive is in a double traction*

Format:

	call byte	Header Byte	Identifier	Data 1	Data 2	Data 3	Data 4	Data 5	X-Or-Byte
Binary :	P11A AAAA	1110 0110	0110 BFFF	RVVV VVVV	000F FFFF	FFFF FFFF	Adr High	Adr Low	X-Or-Byte
Hex :	P+0x60+GA	0xE6	identifier	speed	F0	F1	UH	AL	X-Or-Byte
Dec :	P+96+GA	230	identifier	speed	F0	F1	UH	AL	X-Or-Byte

Description:

Identifier: bits 3 to 0 are coded as described in 2.19.1.

Data 1: **Speed:** The speed byte is encoded as described in 2.19.1.

Data 2 **000F FFFF** F0: Coded as described in 2.19.1. Coded

Data 3 **FFFF FFFF** F1: as described in 2.19.1.

Data 4 **UH** High byte of the second locomotive address of the double traction. Low

Data 5 **AL** byte of the second locomotive address of the double traction

For locomotive addresses < 100 the following applies:

High byte of the locomotive address is 0x00 Low byte of

the locomotive address is 0x00 to 0x63

For locomotive addresses from 100 to 9999 the following applies:

High byte of the locomotive address is: AH = (ADR&0xFF00)+0xC000

Low byte of the locomotive address is: AL = (ADR&0x00FF)

Special features:

This response only comes if the locomotive was requested in the DTR with the “new” locomotive request command (see 3.42.9).

2.19.7 *Locomotive occupied from central version 3.0*

Format:

	call byte	header byte	identifier	Data 1	Data 2	X-Or-Byte
Binary :	P11A AAAA	1110 0011	0100 0000	Address High	Address Low	X-Or-Byte
Hex :	P+0x60+GA	0xE3	0x40	UH	AL	X-Or-Byte
Dec :	P+96+GA	227	64	UH	AL	X-Or-Byte

Description:

Data 1 **UH** High byte of the second locomotive address of the double traction. Low

Data 2 **AL** byte of the second locomotive address of the double traction

For locomotive addresses < 100 the following applies:

High byte of the locomotive address is 0x00 Low byte of

the locomotive address is 0x00 to 0x63

For locomotive addresses from 100 to 9999 the following applies:

High byte of the locomotive address is: AH = (ADR&0xFF00)+0xC000

Low byte of the locomotive address is: AL = (ADR&0x00FF)

Special features:

This information always comes unsolicited when another XpressNet device has taken over this locomotive.

2.19.8 *Function status F0 to F12 of the requested locomotive (from version 3.0)*

As of version 3.0 of the LZ100 central unit, this stores additional information about a locomotive, whether its functions should be touch-sensitive or non-touch-sensitive. The track output for this does not change, however. XpressNet devices can, however, extend their functionality in the user interface so that, for example, an assigned function for sounds is only carried out as long as a button is pressed. The command is provided in the central unit so that this feature can also be used when a locomotive is taken over by another XpressNet device.

Format:

	call byte	header byte identifier		Data 1	Data 2	X-Or-Byte
Binary :	P11A AAAA	1110 0011	0101 0000	000S SSSS	SSSS SSSS	X-Or-Byte
Hex :	P+0x60+GA	0xE3	0x50	S0	S1	X-Or-Byte
Dec :	P+96+GA	227	80	S0	S1	X-Or-Byte

Description:

Data 1: S0: Contains the status of functions 0 to 4.

7	6	5	4	3	2	1	0	bit
0	0	0	S0	S4	S3	S2	S1	SX=1 means function is tentative.

Data 2: S1: Contains the status of functions 5 to 12. SX=1 means function is momentary.

7	6	5	4	3	2	1	0	bit
S12	S11	S10	S9	S8	S7	S6	S5	SX=1 means function is tentative.

Special features:

No.

2.20 Function status F13 to F28 of the requested locomotive (from version 3.6)

	call byte	header byte identifier		Data 1	Data 2	Data 3	X-Or-Byte
Binary :	P11A AAAA	1110 0100	0101 0001	SSSS SSSS	SSSS SSSS	RRRR RRRR	X-Or-Byte
Hex :	P+0x60+GA	0xE4	0x51	S0	S1	R	X-Or-Byte

Dec :	P+96+GA	228	81	S0	S1	R	X-Or-Byte

Description:

Data 1: **S0:** Contains the status of functions 13 to 20.

7	6	5	4	3	2	1	0	bit
S20	S19	S18	S17	S16	S15	S14	S13	SX=1 means function is tentative.

Data 2: **S1:** Contains the status of functions 21 to 28.

7	6	5	4	3	2	1	0	bit
S2	S19	S18	S17	S16	S15	S14	S13	SX=1 means function is tentative.

Data 3: **R** refresh mode

Value	Refresh for
0	F0 ... F4
1	F0 ... F8
3	F0 ... F12
7	F0 ... F20
0xF	F0 ... F28

Special features:

No.

2.21 Function status F29 to F68 of the requested locomotive (from version 4.0)

	call byte	header byte identifier		Data 1	Data 2	Data 3	Data 4	Data 5	X-Or-Byte
Binary :	P11A AAAA	1110 0110	0101 0100	ssss ssss	ssss ssss	ssss ssss	ssss ssss	ssss ssss	X-Or-Byte
Hex :	P+0x60+GA	0xE6	0x54	S0	S1	S2	S2	S2	X-Or-Byte
Dec :	P+96+GA	230	84	S0	S1	S2	S2	S2	X-Or-Byte

Description:

Data 1: **S0:** Contains the status of functions 29 to 36.

7	6	5	4	3	2	1	0	bit
S36	S35	S34	S33	S32	S31	S30	S29	SX=1 means function is tentative.

Data 2: **S1:** Contains the status of functions 37 to 44.

7	6	5	4	3	2	1	0	bit
S44	S43	S42	S41	S40	S39	S38	S37	SX=1 means function is tentative.

Data 3: **S2:** Contains the status of functions 45 to 52.

7	6	5	4	3	2	1	0	bit
S52	S51	S50	S49	S48	S47	S46	S45	SX=1 means function is tentative.

Data 4: **S3:** Contains the status of functions 53 to 60.

7	6	5	4	3	2	1	0	bit
S60	S59	S58	S57	S56	S55	S54	S53	SX=1 means function is tentative.

Data 5: **S4:** Contains the status of functions 61 to 68.

7	6	5	4	3	2	1	0	bit
S68	S67	S66	S65	S64	S63	S62	S63	SX=1 means function is tentative.

Special features:

No.

2.22 Locomotive information for address search queries from central control unit version 3.0

Format:

	call byte	header byte identifier		Data 1	Data 2	X-Or-Byte
Binary :	P11A AAAA	1110 0011	0011 KKKK	Address High	address Low	X-Or-Byte
Hex :	P+0x60+GA	0xE3	0x30 + K	UH	AL	X-Or-Byte
Dec :	P+96+GA	227	48 + K	UH	AL	X-Or-Byte

Description:

This response is sent when the XpressNet device has used one of the search queries from 3.44. This can be used, for example, to show a selection list of the desired locomotives in the device (next locomotive in a multiple traction, etc.).

identifier The identifier contains the type of locomotive address, which is in Address High / Address Low.

KKKK = 0 Normal locomotive in data 1/2 Double traction

KKKK = 1 locomotive in data 1/2 Multiple traction base

KKKK = 2 address in data 1/2 Member of a multiple traction

KKKK = 3 in data 1/2

KKKK = 4 No more addresses found for the search query. Data 1/2 = 0x00 high byte of

Data 1 **UH** the second locomotive address of the double traction.

Data 2 **AL** Low byte of the second locomotive address of the double traction

AH/AL: The locomotive address is calculated as described in 2.19.7.

Special features:

No.

2.22.1 *Double traction information up to LZ version 1.5*

This locomotive information is obtained by sending a locomotive request to the central unit as described in 3.41.1. Up to and including version 1.5 of the LZ100 central unit, only 14 speed steps were used (ie no “ModSel” byte). The information “Double traction occupied” isunasked sent to the device that was just driving this locomotive, so that this device should have a corresponding routine to alert the user that this locomotive has just been taken over by another device.

2.22.1.1 Double traction free up to LZ version 1.5

If the locomotive is not yet in use by any other XpressNet device, you will receive this locomotive information with the header byte “Locomotive free” upon request.

Format:

	call byte	header byte	Data 1	Data 2	Data 3	Data 4	Data 5	X-Or-Byte
Binary :	P11A AAAA	1100 0101	0000 0100	locomotive address 1	locomotive data 1	locomotive data 2	locomotive address 2	X-Or-Byte
Hex :	P+0x60+GA	0xC5	0x04	locomotive address 1	locomotive data 1	locomotive data 2	locomotive address 2	X-Or-Byte
Dec :	P+96+GA	197	4	locomotive address 1	locomotive address 2	locomotive data 2	locomotive address 2	X-Or-Byte

Description:

Locomotive address 1, locomotive data 1, locomotive data 2 and locomotive address 2 are coded as described in 2.17.1.

Special features:

From central control unit version 3.0 or higher, only the driving commands with the "ModSel" byte are supported. Double traction information is only sent to an XpressNet device that has requested this information in the locomotive request with "ModSel". A locomotive request in the new format means that this locomotive, which is in an "old" double traction, is reported to the XpressNet device as a normal locomotive and can be used there with the new driving commands.

2.22.2 *Double traction occupied up to LZ version 1.5*

Format:

	call byte	header byte data 1		Data 2	Data 3	Data 4	Data 5	X-Or-Byte
Binary :	P11A AAAA	1100 0101	0000 0101	locomotive address 1	locomotive data 1	locomotive data 2	locomotive address 2	X-Or-Byte
Hex :	P+0x60+GA	0xC5	0x05	locomotive address 1	locomotive data 1	locomotive data 2	locomotive address 2	X-Or-Byte
Dec :	P+96+GA	197	5	locomotive address 1	locomotive data 1	locomotive data 2	locomotive address 2	X-Or-Byte

Description:

Locomotive address 1, locomotive data 1, locomotive data 2 and locomotive address 2 are coded as described in 2.17.1.

Special features:

An XpressNet device can also receive the information “double traction occupied” without being asked.

2.23 Double traction information up to LZ version 2.3

This locomotive information is obtained by sending a locomotive request to the central station as described in 3.41.2. Because of the possibility of using 14, 27 and 28 speed steps, the “ModSel” byte is added here. The information “Double traction occupied” isunasked sent to the device which

this locomotive was just running, so this device should have a corresponding routine to alert the user that this locomotive has just been taken over by another device.

2.23.1 *Double traction free up to LZ version 2.3*

If the locomotive is not yet in use by any other XpressNet device, you will receive this locomotive information with the header byte “Locomotive free” upon request.

Format:

	call byte	header byte	Data 1	Data 2	Data 3	Data 4	Data 5	Data 5	X-Or-Byte
Binary :	P11A AAAA	1100 0110	0000 0100	local doctor 1	Lokdat 1	Lokdat 2	local doctor 2	Modsel	X-Or-Byte
Hex :	P+0x60+GA	0xC6	0x04	address 1	Lokdat 1	Lokdat 2	address 2	ModSel	X-Or-Byte
Dec :	P+96+GA	198	4	address 1	Lokdat 1	Lokdat 2	address 2	ModSel	X-Or-Byte

Description:

Locomotive address 1, locomotive data 1, locomotive data 2, locomotive address 2 and ModSel are coded as described in 2.18.1.

Special features:

From central unit version 3.0 or higher, only the driving commands with the "ModSel" byte are supported. Double traction information is only sent to an XpressNet device that has requested this information in the locomotive request with "ModSel". A locomotive request in the new format means that this locomotive, which is in an "old" double traction, is reported to the XpressNet device as a normal locomotive and can be used there with the new driving commands. For central units from version 4.0 onwards, "old" driving commands are no longer supported.

2.23.2 *Double traction occupied up to LZ version 2.3*

Format:

	call byte	header byte	Data 1	Data 2	Data 3	Data 4	Data 5	Data 5	X-Or-Byte
Binary :	P11A AAAA	1100 0110	0000 0101	local doctor 1	Lokdat 1	Lokdat 2	local doctor 2	Modsel	X-Or-Byte
Hex :	P+0x60+GA	0xC6	0x05	address 1	Lokdat 1	Lokdat 2	address 2	ModSel	X-Or-Byte
Dec :	P+96+GA	198	5	address 1	Lokdat 1	Lokdat 2	address 2	ModSel	X-Or-Byte

Description:

Locomotive address 1, locomotive data 1, locomotive data 2, locomotive address 2 and ModSel are as under2.18.1coded as described.

Special features:

An XpressNet device can also receive the information “double traction occupied” without being asked.

2.23.3 *Double traction error up to LZ version 2.3*

A double traction can only be assembled or disassembled if certain boundary conditions are met. This can therefore also lead to errors, which are described below.

Format:

	call byte	header byte data 1		X-Or-Byte
Binary :	P11A AAAA	0110 0001	1000 0FFF	X-Or-Byte
Hex :	P+0x60+GA	0x61	0x80 + F	X-Or-Byte
Dec :	P+96+GA	97	128 + F	X-Or-Byte

Description:

The 3 error bits are coded as follows:

FFF = 011: One of the locomotives is not called up by the assembling device when assembling the double traction or locomotive 0 is selected.

FFF = 100: One of the locomotives in the double traction is called up on another device. One

FFF = 101: of the locomotives is already integrated into another double traction.

FFF = 110: The speed of one of the locomotives is not zero.

Special features:

Normally, the command "assemble double traction" or "remove double traction" does not result in a response from the control center if the action was successful. If it fails, however, the control center sends the error messages described. An XpressNet device must therefore be set up to intercept these error messages before continuing with its program. To determine whether the action was successful, it can, for example, wait for the next normal addressing by the control center. If an error occurs, the response is received immediately before the control center addresses the next device.

2.24 Error messages from central version 3.0

Starting with control panel version 3.0, various errors are summarized in an error response. The connection is derived from the previous command given to the control panel.

Format:

	call byte	header byte identifier		X-Or-Byte
Binary :	P11A AAAA	1110 0001	1000 FFFF	X-Or-Byte
Hex :	P+0x60+GA	0xE1	0x80 + F	X-Or-Byte
Dec :	P+96+GA	225	128 + F	X-Or-Byte

Description:

The 4 error bits are encoded as follows:

FFFF = 0001: When assembling a multiple traction, a locomotive is not called up by the assembling device or locomotive 0 is selected.

FFFF = 0010: One of the locomotives of the multiple traction is called up on another device.

FFFF = 0011: One of the locomotives is already involved in another multiple traction or double traction.

FFFF = 0100: The speed of one of the locomotives in the multiple traction is not zero. The

FFFF = 0101: locomotive is not in a multiple traction.

FFFF = 0110 The locomotive address is not a multiple traction base address. Deleting
FFFF = 0111: the locomotive is not possible
FFFF = 1000: The central stack is full
FFFF = 1001: MTR address not found during deletion attempt The
FFFF = 1010: requested MTR address is a used locomotive address

Special features:

No.

3 commands from device to control center

After the central unit has addressed an XpressNet device with the call "Normal device request", this device can begin communication with the central unit and, for example, request data, control locomotives or carry out programming actions. Depending on the desired action, the central unit then gives the device the appropriate response. After the central unit call, only one action is possible from the slave. This means that an XpressNet device cannot send a drive command to the central unit and then immediately request a locomotive. In this case, the central unit first addresses other devices. This ensures that a device cannot block the bus.

3.1 Acknowledgement

Format:

	header byte	X-Or-Byte
Binary :	0010 0000	0010 0000
Hex :	0x20	0x20
Dec :	32	32

Description:

If a device is addressed with the request for acknowledgment, it must respond with this command. Otherwise it will be addressed again and again with the acknowledgment.

Special features:

No.

3.2 Everything On

Format:

	header byte	data 1	X-Or-Byte
Binary :	0010 0001	1000 0001	1010 0000
Hex :	0x21	0x81	0xA0
Dec :	33	129	160

Description:

The command causes the control center to switch the voltage on the track back on if it was switched off and to start sending track commands again. This ends an emergency stop, an emergency off or programming operation on the programming track. After switching on successfully, the control center sends the broadcast "All On". See 2.1.4.1.

Special features:

No.

3.3 All Off (Emergency Stop)

Format:

	header byte	data 1	X-Or-Byte
Binary :	0010 0001	1000 0000	1010 0001
Hex :	0x21	0x80	0xA1
Dec :	33	128	161

Description:

The command causes the control center to switch off the voltage on the track. The control center then sends the broadcast “All Off” to all bus participants, including the person who gave this command.

Special features:

No.

3.4 Stop all locomotives (emergency stop)

Format:

	header byte	X-Or-Byte
Binary :	1000 0000	1000 0000
Hex :	0x80	0x80
Dec :	128	128

Description:

The command causes the central unit to immediately stop all locomotives on the track without their set delay. However, the voltage on the track remains switched on so that, for example, switches can still be switched.

Special features:

No.

3.5 Stopping a locomotive (emergency stop for a locomotive)

3.6 Stop locomotive up to LZ version 2.3

Format:

	header byte	data 1	X-Or-Byte
Binary :	1001 0001	locomotive address	X-Or-Byte
Hex :	0x91	locomotive address	X-Or-Byte
Dec :	145	locomotive address	X-Or-Byte

Description:

The command causes the central unit to immediately stop only the desired locomotive on the track without its set delay. The voltage on the track remains switched on so that, for example, switches can still be switched and all other locomotives continue to run normally.

Special features:

The range from 0 to 99 is permitted as a locomotive address.

3.7 Stop locomotive from central control unit version 3.0

Format:

	header byte	data 1	Data 2	X-Or-Byte
Binary :	1001 0010	address High	address Low	X-Or-Byte
Hex :	0x92	UH	AL	X-Or-Byte
Dec :	146	UH	AL	X-Or-Byte

Description:

The command causes the central unit to immediately stop only the desired locomotive on the track without its set delay. The voltage on the track remains switched on so that, for example, switches can still be switched and all other locomotives can continue to run normally.

Special features:

Locomotives 0 to 9999 can be stopped.

The locomotive address AH/AL is calculated as specified in 2.19.7.

3.8 Stopping multiple locomotives up to LZ version 2.3

Format:

	header byte	data 1	...	Data N	X-Or-Byte
Binary :	1001 NNNN	locomotive address 1	...	locomotive address N	X-Or-Byte
Hex :	0x90 + N	locomotive address 1	...	locomotive address N	X-Or-Byte
Dec :	144 + N	locomotive address 1	...	locomotive address N	X-Or-Byte

Description:

The command causes the central unit to immediately stop only the desired locomotives on the track without their set delay. The voltage on the track remains switched on so that, for example, switches can still be switched and all other locomotives can continue to run normally.

Special features:

Only locomotives 0 to 99 are allowed. The command is no longer supported in version 3.0 and should be replaced by a sequence of commands "Stop a locomotive".

3.9 Read request programming 3-byte format (register mode)

Format:

	header byte	data 1	Data 2	X-Or-Byte
Binary :	0010 0010	0001 0001	0000 RRRR	X-Or-Byte
Hex :	0x22	0x11	R	X-Or-Byte

Dec :	34	17	R	X-Or-Byte

Description:

The command causes the central unit to switch to programming mode and to read the receiver on the programming track in register mode. An attempt is made to read the register specified as 0000 RRRR. Registers 1..8 are permitted.

Special features:

The read request does not result in a response from the control center! This must be explicitly requested using the "Request programming result" command. Only then can it be determined whether the read command was successful or not and whether the result is in the desired form (register mode).

After issuing a read command, the control unit sends the broadcast “programming mode” to all bus participants and from then on only addresses the device that triggered the programming mode.

3.10 Read request programming 4-byte format (CV mode, CV 1-256)

Format:

	header byte	Data 1	Data 2	X-Or-Byte
Binary :	0010 0010	0001 0101	CCCC CCCC	X-Or-Byte
Hex :	0x22	0x15	CV	X-Or-Byte
Dec :	34	21	CV	X-Or-Byte

Description:

The command causes the command station to switch to programming mode and read the receiver on the programming track in CV mode. An attempt is made to read the CV specified with CCCC CCCC.

The range is from 1 to 256, where CV256 is to be sent as 00.

Special features:

This command exists in addition to the command described in 3.11. If a control unit from version 3.6 is used, the command returns the value of CV1024 instead of CV256. We therefore recommend using the command described in 3.11.

The read request does not result in a response from the central unit! This must be explicitly requested using the "Request programming result" command. Only then can it be determined whether the read command was successful or not and whether the result is in the desired form (CV mode). If the receiver could not be read in CV mode, the central unit tries in register mode. If this reading action is successful, a result is available for collection in the central unit and the XpressNet device must check this result to see whether it is a CV or register result and adjust its display accordingly so that subsequent write commands to this receiver are only executed in the mode that the receiver understands.

After issuing a read command, the control unit sends the broadcast “programming mode” to all bus participants and from then on only addresses the device that triggered the programming mode.

3.11 Read request programming 4-byte format (CV 1-255 and CV1024) (new from V3.6)

Format:

	header byte	Data 1	Data 2	X-Or-Byte

Binary :	0010 0010	0001 1000	CCCC CCCC	X-Or-Byte
Hex :	0x22	0x18	CV	X-Or-Byte
Dec :	34	24	CV	X-Or-Byte

Description:

The command causes the command station to switch to programming mode and read the receiver on the programming track in CV mode. An attempt is made to read the CV specified with CCCC CCCC.

Assignment value in "Data 2" <=> CV – addresses:

Data 2	CV
0	1024
1 ... 255	1 ... 255

Special features:

This command should always be used on a control panel from version 3.6 onwards; further special features apply as described under 3.10.

3.12 Read request programming 4-byte format (CV 256-511) (new from V3.6)

Format:

	header byte	Data 1	Data 2	X-Or-Byte
Binary :	0010 0010	0001 1001	CCCC CCCC	X-Or-Byte
Hex :	0x22	0x19	CV	X-Or-Byte
Dec :	34	25	CV	X-Or-Byte

Description:

The command causes the command station to switch to programming mode and read the receiver on the programming track in CV mode. An attempt is made to read the CV specified with CCCC CCCC.

Assignment value in "Data 2" <=> CV – addresses:

Data 2	CV
0 ... 255	256 ... 511

Special features:

As described in 3.10.

3.13 Read request programming 4-byte format (CV 512-767) (new from V3.6)

Format:

	header byte	Data 1	Data 2	X-Or-Byte
Binary :	0010 0010	0001 1010	CCCC CCCC	X-Or-Byte
Hex :	0x22	0x1A	CV	X-Or-Byte
Dec :	34	26	CV	X-Or-Byte

Description:

The command causes the command station to switch to programming mode and read the receiver on the programming track in CV mode. An attempt is made to read the CV specified with CCCC CCCC.

Assignment value in "Data 2" <=> CV – addresses:

Data 2	CV
0 ... 255	512 ... 767

Special features:

As described in 3.10.

3.14 Read request programming 4-byte format (CV 768-1023) (new from V3.6)

Format:

	header byte	Data 1	Data 2	X-Or-Byte
Binary :	0010 0010	0001 1011	CCCC CCCC	X-Or-Byte
Hex :	0x22	0x1B	CV	X-Or-Byte
Dec :	34	27	CV	X-Or-Byte

Description:

The command causes the command station to switch to programming mode and read the receiver on the programming track in CV mode. An attempt is made to read the CV specified with CCCC CCCC.

Assignment value in "Data 2" <=> CV – addresses:

Data 2	CV
0 ... 255	768 ... 1023

Special features:

As described in 3.10.

3.15 Read request programming 3-byte format (page mode)

Format:

	header byte	data 1	Data 2	X-Or-Byte
Binary :	0010 0010	0001 0100	CCCC CCCC	X-Or-Byte
Hex :	0x22	0x14	CV	X-Or-Byte
Dec :	34	20	CV	X-Or-Byte

Description:

The command causes the central unit to switch to programming mode and to read the receiver on the programming track in page mode. An attempt is made to read the CV that begins with CCCC

CCCC is specified. The central unit converts the page information to registers (on the track side) and attempts to read the receiver in register mode.

The range is from 1 to 256, where CV256 is to be sent as 00.

Special features:

The read request does not result in a response from the central unit! This must be explicitly requested using the "Request programming result" command. Only then can it be determined whether the read command was successful or not and whether the result is in the desired form (page mode). If a read action is successful, a result is available for collection in the central unit and the XpressNet device must examine this result for its content.

After issuing a read command, the control unit sends the broadcast “programming mode” to all bus participants and from then on only addresses the device that triggered the programming mode.

3.16 Request programming result

Format:

	header byte	data 1	X-Or-Byte
Binary :	0010 0001	0001 0000	0011 0001
Hex :	0x21	0x10	0x31
Dec :	33	16	49

Description:

The command causes the control center to send the result of a previous reading action to the XpressNet device. The response is one of the options described in 2.5.

Special features:

No.

3.17 Write command programming 3-byte format (register mode)

Format:

	header byte	data 1	Data 2	Data 3	X-Or-Byte
Binary :	0010 0011	0001 0010	0000 RRRR	Data	X-Or-Byte
Hex :	0x23	0x12	R	Data	X-Or-Byte
Dec :	35	18	R	Data	X-Or-Byte

Description:

The command causes the central unit to switch to programming mode and write the receiver on the programming track in register mode. An attempt is made to write the value in data 3 to the register address in data 2.

The range is registers 1 to 8.

Special features:

Before using a write command, the central unit should be put into programming mode by a read command. There is no control on the part of the XpressNet device as to whether the receiver has understood the programming sequence, except by reading it out again.

3.18 Write command programming 4-byte format (CV mode, CV 1-256)

Format:

	header byte	data 1	Data 2	Data 3	X-Or-Byte
Binary :	0010 0011	0001 0110	CCCC CCCC	Data	X-Or-Byte
Hex :	0x23	0x16	CV	Data	X-Or-Byte
Dec :	35	22	CV	Data	X-Or-Byte

Description:

The command causes the central unit to switch to programming mode and write the receiver on the programming track in CV mode. An attempt is made to write the value in data 3 to the CV address in data 2.

The range is CV 1 to 256, where CV256 must be sent as 0x00.

Special features:

This command exists in addition to the command described in 3.19. If a control unit from version 3.6 is used, CV1024 is written instead of CV256. We therefore recommend using the command described in 3.19.

Before using a write command, the central unit should be put into programming mode using a read command and checked to see whether the receiver can be programmed in CV mode. There is no way for the XpressNet device to check whether the receiver has understood the programming sequence, except by reading it out again.

3.19 Write command programming 4-byte format (CV mode, CV 1-255 and CV1024) (new from version 3.6)

Format:

	header byte	data 1	Data 2	Data 3	X-Or-Byte
Binary :	0010 0011	0001 1100	CCCC CCCC	Data	X-Or-Byte
Hex :	0x23	0x1C	CV	Data	X-Or-Byte
Dec :	35	28	CV	Data	X-Or-Byte

Description:

The command causes the central unit to switch to programming mode and write the receiver on the programming track in CV mode. An attempt is made to write the value in data 3 to the CV address in data 2.

Assignment value in "Data 2" <=> CV – addresses:

Data 2	CV
0	1024
1 ... 255	1 ... 255

Special features:

Before using a write command, the central unit should be put into programming mode using a read command and checked to see whether the receiver can be programmed in CV mode. There is no way for the XpressNet device to check whether the receiver has understood the programming sequence, except by reading it out again.

3.20 Write command programming 4-byte format (CV mode, CV 256-511) (new from version 3.6)

Format:

	header byte	data 1	Data 2	Data 3	X-Or-Byte
Binary :	0010 0011	0001 1101	CCCC CCCC	Data	X-Or-Byte
Hex :	0x23	0x1D	CV	Data	X-Or-Byte
Dec :	35	29	CV	Data	X-Or-Byte

Description:

The command causes the central unit to switch to programming mode and write the receiver on the programming track in CV mode. An attempt is made to write the value in data 3 to the CV address in data 2.

Assignment value in "Data 2" <=> CV – addresses:

Data 2	CV
0 ... 255	256 ... 511

Special features:

like 3.19.

3.21 Write command programming 4-byte format (CV mode, CV 512-767) (new from version 3.6)

Format:

	header byte	data 1	Data 2	Data 3	X-Or-Byte
Binary :	0010 0011	0001 1110	CCCC CCCC	Data	X-Or-Byte
Hex :	0x23	0x1E	CV	Data	X-Or-Byte
Dec :	35	30	CV	Data	X-Or-Byte

Description:

The command causes the central unit to switch to programming mode and write the receiver on the programming track in CV mode. An attempt is made to write the value in data 3 to the CV address in data 2.

Assignment value in "Data 2" <=> CV – addresses:

Data 2	CV
0 ... 255	512 ... 767

Special features:

like 3.19.

3.22 Write command programming 4-byte format (CV mode, CV 768-1023) (new from version 3.6)

Format:

	header byte	data 1	Data 2	Data 3	X-Or-Byte
Binary :	0010 0011	0001 1111	CCCC CCCC	Data	X-Or-Byte
Hex :	0x23	0x1F	CV	Data	X-Or-Byte
Dec :	35	31	CV	Data	X-Or-Byte

Description:

The command causes the central unit to switch to programming mode and write the receiver on the programming track in CV mode. An attempt is made to write the value in data 3 to the CV address in data 2.

Assignment value in "Data 2" <=> CV – addresses:

Data 2	CV
0 ... 255	768 ... 1023

Special features:

like 3.19.

3.23 Write command programming 3-byte format (page mode)

Format:

	header byte	data 1	Data 2	Data 3	X-Or-Byte
Binary :	0010 0011	0001 0111	CCCC CCCC	Data	X-Or-Byte
Hex :	0x23	0x17	CV	Data	X-Or-Byte
Dec :	35	23	CV	Data	X-Or-Byte

Description:

The command causes the central unit to switch to programming mode and write the receiver on the programming track in page mode. An attempt is made to write the value in data 3 to the CV address in data 2, whereby the central unit converts the CV to the page to be used and programs the receiver in register mode.

The range is CV 1 to 256, where CV256 must be sent as 0x00.

Special features:

Before using a write command, the control unit should be put into programming mode by a read command and checked whether the receiver can be programmed in page mode.

There is no control on the part of the XpressNet device as to whether the receiver has understood the programming sequence, except by reading it out again.

3.24 Request software version of the control unit

Format:

	header byte	data 1	X-Or-Byte
Binary :	0010 0001	0010 0001	0000 0000
Hex :	0x21	0x21	0x00
Dec :	33	33	0

Description:

This command causes the control panel to communicate its software version to the XpressNet device. Depending on the control panel version, the responses are possible as described in 2.7.

Special features:

No.

3.25 Query advanced central version information (from version 3.8)

Format:

	header byte	Kenn/D1	X-Or-Byte
Binary :	0010 0001	0010 0011	0000 0010
Hex :	0x21	0x23	0x02
Dec :	33	35	2

Description:

This command causes the control panel to respond with the extended version information, see "Report extended version information" (=>p. 19).

3.26 Request status of the control center

Format:

	header byte	data 1	X-Or-Byte
Binary :	0010 0001	0010 0100	0000 0101
Hex :	0x21	0x24	0x05
Dec :	33	36	5

Description:

The request for the central status results in the answer described in 2.8.

Special features:

No.

3.27 Setting the central start mode

Format:

	header byte	data 1	Data 2	X-Or-Byte
Binary :	0010 0010	0010 0010	0000 0M00	X-Or-Byte
Hex :	0x22	0x22	M	X-Or-Byte
Dec :	34	34	M	X-Or-Byte

Description:

Sets the start mode of the central unit after reset. M=0: Manual start of all locomotives, M=1: Automatic start of all locomotives with the last speed and function settings.

Special features:

Not all control panels support the “automatic start” operating mode.

3.28 Service Variable Read Command (from version 3.8)

Format:

	header byte	Kenn/D1	D2	X-Or-byte
Binary :	0010 0010	0010 0101	AAAA AAAA	
Hex :	0x22	0x25		
Dec :	34	37		

Description:

This command causes the control panel to respond with the content of the addressed service variable, see "Reporting service variable" (=>p. 19).

Data 2: AAAA AAAAS Address of the service variable (SV); 0=SV256

3.29 Service Variable Write Command (from version 3.8)

Format:

	header byte	Kenn/D1	D2	D3	X-Or-byte
Binary :	0010 0011	0010 0110	AAAA AAAA	VVVV VVVV	
Hex :	0x23	0x26			
Dec :	35	38			

Description:

This command causes the control panel to write the addressed service variable from D2 with the value in D3.

This command is automatically followed by the response "Report service variable" (p. 17), so that the execution of the command can be checked.

SVs 1 to 256 can be addressed, address of SV256 is "0"

Data 2:	**AAAA AAAA**	S	Address of the service variable (SV), 0=SV256
Data 3:	**VVVV VVVV**	V	value to be written

3.30 Get PoM result (from version 3.8)

Format:

	header byte	Kenn/D1	X-Or-Byte
Binary :	0010 0001	0010 0111	0000 0110
Hex :	0x21	0x27	0x06
Dec :	33	39	6

Description:

This command is used to collect the result of a previously sent PoM read command from the central unit. The central unit responds with "Report PoM result", section 2.10, page 20. You should give the central unit some time before collecting the result because reading from the decoders also takes time; about 0.5 to 1 second seems OK.

3.31 Trigger central reset

Format:

	header byte	Kenn/D1	X-Or-Byte
Binary :	0010 0001	0010 1000	0000 1001
Hex :	0x21	0x28	0x09
Dec :	33	40	9

Description:

This command resets the control panel. During this reset, all data and settings are reset to factory values:

- all function statuses are set to "permanent"
- Multiple and double traction are deleted
- The locomotive memory is deleted (and with it all stored speed levels and functional states)
- all feedback information will be deleted
- all switch information is deleted

Attention; there is no inquiry ;-)

3.32 Send device info

Format:

	header byte	Kenn/D1	D2	D3	D4	X-Or-byte
Binary :						
Hex :	0x24	0x29	SW-version number	level	type	XOR
Dec :	36	41				

Description:

With this command the device sends device-describing information to the control center.

SW version number: in the format xx.y as in the control panel software coding: The version number is hexadecimal coded in upper (OOOO) and lower (UUUU) nibbles. Example: Data 2 = 0011 0000 = 0x30 : Version 3.0.

Level: Functional scope of the device; currently for LH101 = 1 Type:

Hardware information of the device: currently LH101 = 101

3.33 Request model time

Format:

	header byte	Kenn/D1	X-Or-Byte
Binary :	0010 0001	0010 1010	0000 1011
Hex :	0x21	0x2A	0x0B
Dec :	33	42	11

Description:

This command requests the current model time from the control center.

The response to this command is described in Section 2.11, "Reporting Model Time", page 20.

3.34 Setting the model time

Format:

	header byte	Kenn/D1	D2	D3	D4	X-Or-byte
Binary :	0010 0100	0010 1100	DDDh hhhh			
Hex :	0x24	0x2B	DOW+h	min	factor	XOR
Dec :	36	43				

Description:

This command is used to set and start the model time in the control center. The clock is always started when a factor greater than 0 is sent.

Data 2:	**DDDh hhh**	DDD: Days of the week:	0 = Monday 1 = Tuesday ... 6 = Sunday 7 not allowed!
		h hhhh: Hour:	0 - 23
Data 3	**min**	minutes:	0 - 59
Data 4	**factor**	time factor model time	0 – 31, 0=Stop (clock stopped)

Special features:

If the model time is active in the central unit (factor 1-31), the broadcast "Model time", section 2.4.7, page 13, is sent every time the model minute changes. In addition, the track command defined in RCN-212 is sent to the track for the model time, also every time the model minute changes.

3.35 Stop model time (from version 4.0)

Format:

	header byte	Kenn/D1	X-Or-Byte
Binary :	0010 0001	0010 1101	0000 1100
Hex :	0x21	0x2D	0x0C
Dec :	33	45	12

Description:

This command stops the current model time without losing its factor.

3.36 Start model time (from version 4.0)

Format:

	header byte	Kenn/D1	X-Or-Byte
Binary :	0010 0001	0010 1100	0000 1101
Hex :	0x21	0x2C	0x0D
Dec :	33	44	13

Description:

This command restarts the current model time with the preset factor.

3.37 Request switching information, address 1 to 1024 (up to version 3.6)

This command is used up to and including version 3.6.

For reasons of downward compatibility, control panels of a higher version should also be able to process this command. This means that "older" devices can also be used on "newer" control panels.

Format:

	header byte	data 1	Data 2	X-Or-Byte
Binary :	0100 0010	AAAA AAAA	1000 000N	X-Or-Byte

Hex :	0x42	address	0x80 + N	X-Or-Byte
Dec :	66	address	128 + N	X-Or-Byte

Description:

Based on this command, the control center sends the response described in 2.15. **Data**

1:	**address**	For a switching receiver, this is the address of the desired (switching output -1) (=switch group) divided by 4. This gives the address a range of 0 to 63 = 6 bits for all versions below 3.0. As of version 3.0, all 8 bits are permitted for the switch group. This results in a range of 256 (0..255) switch groups. This means that 1024 switches can be switched, whereby switches no. 1..512 are capable of feedback, but switches no. 513 to 1024 are not. For a feedback block, the address is in the range 0 to 127 (=7 bits) and directly indicates the desired block.
Data 2:	**N**	The lower nibble is the marking for the
	N = 0	desired nibble
	N = 1	is the upper nibble
		For switching receivers, a switch group contains 4 switches and the lower nibble designates switches 0 and 1 of the switch group and the upper nibble designates switches no. 2 and 3 of the switch group. For a feedback block, the lower nibble indicates the state of the first 4 inputs of the feedback block and the upper nibble indicates the state of the upper 4 inputs.

Special features:

Example 1:	switch area	the condition of switch No. 21 is desired.
	0..255	Address: (21-1) mod 4 = 5, ie switch 21 is in switch group 5. Switches 20, 21, 22, 23 are in switch group 5. This means that the nibble bit is 0 (lower nibble).
Example 2:	switch area	the condition of switch no. 623 is desired.
	0..1023	Address: (623-1) mod 4 = 155, ie switch 623 is in switch group 155. Switches 620, 621, 622, 623 are in switch group 155. This means that the nibble bit is 1 (upper nibble).

3.38 Request switching information, address 1 to 2048 (from version >3.6)

XpressNet devices that are operated on control panels from version 3.8 onwards use this command for the complete switch address range from 1 to 2048.

Format:

	header byte	data 1	Data 2	Data 3	X-Or-Byte
Binary :	0100 0011	0000 000A	AAAA AAAA	1000 000N	X-Or-Byte

Hex :	0x43	address H/4	address L/4	0x80 + N	X-Or-Byte
Dec :	67	address		128 + N	X-Or-Byte

Description:

Data 1: **address, high byte** For a switching receiver, this is the address of the desired (switch output – 1) (= switch group) divided by 4. Switch 1 is therefore address 0

Data 2: **address, low byte**

Data 3: **N** The lower nibble is the marking for the

N = 0 desired nibble

N = 1 is the upper nibble

For switching receivers, a switch group contains 4 switches and the lower nibble designates switches 0 and 1 of the switch group and the upper nibble designates switches no. 2 and 3 of the switch group.

For a feedback block, the lower nibble indicates the state of the first 4 inputs of the feedback block and the upper nibble indicates the state of the upper 4 inputs.

3.39 switching command up to version 3.6 incl.

This command is used up to and including version 3.6.

For reasons of downward compatibility, control panels of a higher version should also be able to process this command. This means that "older" devices can also be used on "newer" control panels.

Format:

	header byte	data 1	Data 2	X-Or-Byte
Binary :	0101 0010	AAAA AAAA	1000 DBBD	X-Or-Byte
Hex :	0x52	address	0x80 + DBBD	X-Or-Byte
Dec :	82	address	128 + DBBD	X-Or-Byte

Description:

Switching commands can only be given to switching receivers. The address is therefore (switch number-1) / 4 (=switch group). The offset in the switch group still needs to be defined in order to precisely define the desired switch, as well as selecting which of the two outputs of this switch is required and whether this output is to be activated or deactivated. This is done using the 4 bits D1 B1 B0 D2 in data 2.

Data 2: **1000 D1 B1 B0 D2**

B1 B0: These are the two LSBs of the switch address that were omitted when dividing by 4.

D1: D1 = 0 means deactivate output. D1 = 1 means activate output.

D2: D2 = 0 means output 1 of the switch is selected. D2 = 1 means output 2 of the switch is selected.

Special features:

An activation and a deactivation command are always required to operate the switching receiver. After an activation command, the switching command is issued on the track until the control center receives the deactivation command. This means that the switching receiver can be controlled for any length of time, e.g. by driving a turntable until the desired track exit is reached.
For example, when operating a switch button, a controlling bus participant must send an activation command when it is pressed (once) and a deactivation command when it is released (once).

For control units with versions lower than 3.0, a range of 0..63 is defined for the switch group. This means that the group address is 6 bits long. From version 3.0 onwards, groups up to 255 can be used. See also 3.28.

3.40 Switching command from version 3.8

XpressNet devices that are operated on control panels from version 3.8 onwards use this command for the complete switch address range from 1 to 2048.

Format:

	header byte data 1	Data 2	Data 3	X-Or-Byte
Binary :	0101 0011	0000 000A	AAAA AAAA	1000 DBBD	X-Or-Byte
Hex :	0x53	Address H	Address L	0x80 + DBBD	X-Or-Byte
Dec :	83	Address H	Address L	128 + DBBD	X-Or-Byte

Description:

Data 1: **address, high byte** For a switching receiver, this is the address of the desired (switch output -1) (=switch group) divided by 4. Switch 1 is therefore address 0

Data 2: **address, low byte**

Data 3: Definition as 3.39, "Switching command", page 56

Special features:

For the operation of activation and deactivation commands see 3.39

3.41 Request locomotive information

3.41.1 *Request locomotive information up to LZ version 1.5*

Format:

	header byte data 1		X-Or-Byte
Binary :	1010 0001	locomotive address	X-Or-Byte
Hex :	0xA1	locomotive address	X-Or-Byte
Dec :	161	locomotive address	X-Or-Byte

Description:

Up to and including version 1.5 of the central unit, only 14 speed steps were used, so that no additional differentiation byte (ModSel) was necessary. A locomotive request with this command means that the central unit only responds with locomotive information up to LZ version 1.5, since it assumes that the XpressNet device only understands this type of communication.

Locomotive address is in the range 0 to 99. The answer is described under 2.17.

Special features:

No.

3.41.2 *Request locomotive information up to LZ version 2.3*

Format:

	header byte	data 1	Data 2	X-Or-Byte
Binary :	1010 0010	locomotive address	ModSel	X-Or-Byte
Hex :	0xA2	locomotive address	ModSel	X-Or-Byte
Dec :	162	locomotive address	ModSel	X-Or-Byte

Description:

Here, the ModSel byte is also sent so that the central unit is prompted to respond with the response including the ModSel byte so that the XpressNet device can determine the number of speed steps of the requested locomotive.

Locomotive address is in the range 0 to 99. The answer is described under 2.18.

Special features:

No.

3.41.3 *Request locomotive information from central control unit version 3.0*

Format:

	header byte	identifier	Data 1	Data 2	X-Or-Byte
Binary :	1110 0011	0000 0000	Address High	address Low	X-Or-Byte
Hex :	0xE3	0x00	UH	AL	X-Or-Byte
Dec :	227	0	UH	AL	X-Or-Byte

Description:

Locomotives 0 to 9999 can be requested.

The locomotive address AH/AL is calculated as specified in 2.19.7.

The possible answers are described in 2.19.

Special features:

No.

3.41.4 *Request function status from central version 3.0*

Format:

	header byte	identifier	Data 1	Data 2	X-Or-Byte

Binary :	1110 0011	0000 0111	Address High	address Low	X-Or-Byte
Hex :	0xE3	0x07	UH	AL	X-Or-Byte
Dec :	227	7	UH	AL	X-Or-Byte

Description:

Gets the functional status F0 to F12 as momentary or non-momentary.

The locomotives 0 to 9999 can be requested.

The locomotive address AH/AL is calculated as specified in 2.19.7.

The possible answers are described in 2.19.

Special features:

No.

3.41.5 *Request function status F13 – F28 (new from central version 3.6)*

Format:

	header byte identifier		Data 1	Data 2	X-Or-Byte
Binary :	1110 0011	0000 1000	Address High	address Low	X-Or-Byte
Hex :	0xE3	0x08	UH	AL	X-Or-Byte
Dec :	227	8	UH	AL	X-Or-Byte

Description:

Gets the function status F13 to F28 as momentary or non-momentary.

The locomotives 0 to 9999 can be requested.

The locomotive address AH/AL is calculated as specified in 2.19.7.

The possible answers are described in 2.20.

Special features:

No.

3.41.6 *Request function status F29 – F68 (from control panel version 4.0)*

Format:

	header byte identifier		Data 1	Data 2	X-Or-Byte
Binary :	1110 0011	0000 1010	Address High	address Low	X-Or-Byte
Hex :	0xE3	0x0A	UH	AL	X-Or-Byte
Dec :	227	12	UH	AL	X-Or-Byte

Description:

Gets the function status F29 to F68 as momentary or non-momentary.

The locomotives 0 to 9999 can be requested.

The locomotive address AH/AL is calculated as specified in 2.19.7.

The possible answers are described under 2.20.

Special features:

No.

3.41.7 *Request functional status F13 – F28 (new from central control unit version 3.6)*

Format:

	header byte identifier		Data 1	Data 2	X-Or-Byte
Binary :	1110 0011	0000 1001	Address High	address Low	X-Or-Byte
Hex :	0xE3	0x09	UH	AL	X-Or-Byte
Dec :	227	9	UH	AL	X-Or-Byte

Description:

Gets the functional status of functions F13 to F28.

Locomotives 0 to 9999 can be queried.

The locomotive address AH/AL is calculated as specified in 2.19.7.

The possible answers are described in 2.19.2.

Special features:

No.

3.41.8 *Request functional status F29 – F68 (from central control unit version 4.0)*

Format:

	header byte identifier		Data 1	Data 2	X-Or-Byte
Binary :	1110 0011	0000 1011	Address High	address Low	X-Or-Byte
Hex :	0xE3	0x0B	UH	AL	X-Or-Byte
Dec :	227	11	UH	AL	X-Or-Byte

Description:

Gets the functional status of functions F29 to F68.

Locomotives 0 to 9999 can be queried.

The locomotive address AH/AL is calculated as specified in 2.19.7.

The possible answers are described in 2.19.2.

Special features:

No.

3.42 Controlling the locomotive

3.42.1 *drive command up to LZ version 1.5*

Format:

	header byte	data 1	Data 2	Data 3	X-Or-Byte
Binary :	1011 0011	locomotive address	locomotive data 1	locomotive data 2	X-Or-Byte
Hex :	0xB3	locomotive address	locomotive data 1	locomotive data 2	X-Or-Byte
Dec :	179	locomotive address	locomotive data 1	locomotive data 2	X-Or-Byte

Description:

Up to and including version 1.5 of the central unit, only 14 speed steps were used, so that no additional differentiation byte (ModSel) was necessary.

Locomotive address is in the range 0 to 99.

Locomotive data 1 and locomotive data 2 are coded as described in 2.17.1.

Special features:

No.

3.42.2 *drive command up to LZ version 2.3*

Format:

	header byte	data 1	Data 2	Data 3	Data 4	X-Or-Byte
Binary :	1011 0100	locomotive address	locomotive data 1	locomotive data 2	ModSel	X-Or-Byte
Hex :	0xB4	locomotive address	locomotive data 1	locomotive data 2	ModSel	X-Or-Byte
Dec :	180	locomotive address	locomotive data 1	locomotive data 2	ModSel	X-Or-Byte

Description:

Due to the possible number of speed steps of 14, 27 or 28, the ModSel byte is sent here so that the central unit can modulate a corresponding track signal.

Locomotive address is in the range 0 to 99.

Locomotive data 1, locomotive data 2 and ModSel are coded as described in 2.18.1.

Special features:

No.

3.42.3 *Drive command from central version 3.0*

The new driving command for a locomotive is divided into 4 different options, which are assigned to the speed levels 14, 27, 28 and 128. This is achieved by the different identifiers. The speed itself is coded for 14, 27 and 28 speed levels as described in 2.18.1. The speed for 128 speed levels as described in 2.19.1.

Driving command 14 speed levels:

Format:

	header byte	identifier	Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0001 0000	address High	address Low	R000 VVVV	X-Or-Byte
Hex :	0xE4	0x10	UH	AL	RV	X-Or-Byte
Dec :	228	16	UH	AL	RV	X-Or-Byte

Driving command 27 speed levels:

Format:

	header byte	identifier	Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0001 0001	address High	address Low	R00V VVVV	X-Or-Byte
Hex :	0xE4	0x11	UH	AL	RV	X-Or-Byte
Dec :	228	17	UH	AL	RV	X-Or-Byte

Driving command 28 speed levels:

Format:

	header byte	identifier	Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0001 0010	address High	address Low	R00V VVVV	X-Or-Byte
Hex :	0xE4	0x12	UH	AL	RV	X-Or-Byte
Dec :	228	18	UH	AL	RV	X-Or-Byte

Driving command 128 speed steps:

Format:

	header byte	identifier	Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0001 0011	address High	address Low	RVVV VVVV	X-Or-Byte
Hex :	0xE4	0x13	UH	AL	RV	X-Or-Byte
Dec :	228	19	UH	AL	RV	X-Or-Byte

Description:

The travel command for version 3 control units only contains the speed and direction information. The functions are set separately.

Locomotives 0 to 9999 can be controlled. The locomotive address is calculated as follows: The

locomotive address AH/AL is calculated as specified in 2.19.7.

Special features:

No.

3.42.4 *Function commands from central control unit version 3.0 / version 3.6 / version 4.0*

The function commands for a locomotive are divided into 3 different options, which are assigned to the functions of group 1 (F0..F4), group 2 (F5..F8), group 3 (F9..F12), group 4 (F13...F20) (from V3.6), group 5 (F21...F28) (from V3.6) and from version 4.0 group 6 (F29-F36) to group 10 (F61 to F68). This is achieved by the different identifiers.

Function command group 1:

Format:

	header byte	identifier	Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0010 0000	address	address	000F FFFF	X-Or-Byte

			High	Low		
Hex :	0xE4	0x20	UH	AL	Group 1	X-Or-Byte
Dec :	228	32	UH	AL	Group 1	X-Or-Byte

The following applies to the

functions: **Data 3:** Group 1:

7	6	5	4	3	2	1	0	bit
0	0	0	F0	F4	F3	F2	F1	FX=1 means "function is on".

Function command group 2:

Format:

	header byte identifier		Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0010 0001	address High	address Low	0000 FFFF	X-Or-Byte
Hex :	0xE4	0x21	UH	AL	Group 2	X-Or-Byte
Dec :	228	33	UH	AL	Group 2	X-Or-Byte

The following applies to the

functions: **Data 3:** Group 2:

7	6	5	4	3	2	1	0	bit
0	0	0	0	F8	F7	F6	F5	FX=1 means "function is on".

Function command group 3:

Format:

	header byte identifier		Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0010 0010	address High	address Low	0000 FFFF	X-Or-Byte
Hex :	0xE4	0x22	UH	AL	Group 3	X-Or-Byte
Dec :	228	34	UH	AL	Group 3	X-Or-Byte

The following applies to the

functions: **Data 3:** Group 2:

7	6	5	4	3	2	1	0	bit
0	0	0	0	F12	F11	F10	F9	FX =1 means "function is on".

Function command group 4 (new from version 3.6):

Format:

	header byte identifier		Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0010 0011	address	address	FFFF FFFF	X-Or-Byte

			High	Low		
Hex :	0xE4	0x23	UH	AL	Group 4	X-Or-Byte
Dec :	228	35	UH	AL	Group 4	X-Or-Byte

The following applies to the

functions: **Data 3:** Group 4:

7	6	5	4	3	2	1	0	bit
F20	F19	F18	F17	F16	F15	F14	F13	FX =1 means "function is on".

Function command group 5 (new from version 3.6):

Format:

	header byte identifier		Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0010 1000	address High	address Low	FFFF FFFF	X-Or-Byte
Hex :	0xE4	0x28	UH	AL	Group 5	X-Or-Byte
Dec :	228	40	UH	AL	Group 5	X-Or-Byte

The following applies to the

functions: **Data 3:** Group 5:

7	6	5	4	3	2	1	0	bit
F28	F27	F26	F25	F24	F23	F22	F21	FX =1 means "function is on".

Function command group 6: F29-F36 (from version 4.0)

Format:

	header byte identifier		Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0010 1001	address High	address Low	0000 FFFF	X-Or-Byte
Hex :	0xE4	0x29	UH	AL	Group 6	X-Or-Byte
Dec :	228	41	UH	AL	Group 6	X-Or-Byte

The following applies to the

functions: **Data 3:** Group 6:

7	6	5	4	3	2	1	0	bit
F36	F35	F34	F33	F32	F31	F30	F29	FX =1 bedindicates "function is on".

Function command group 7: F37-F44 (from version 4.0)

Format:

	header byte identifier		Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0010 1010	address	address	0000 FFFF	X-Or-Byte

			High	Low		
Hex :	0xE4	0x2A	UH	AL	Group 7	X-Or-Byte
Dec :	228	42	UH	AL	Group 7	X-Or-Byte

The following applies to the functions: **Data 3:** Group 7:

7	6	5	4	3	2	1	0	bit
F44	F43	F42	F41	F40	F39	F38	F37	FX=1 means "function is on".

Function command group 8: F45-F52 (from version 4.0)

Format:

	header byte identifier		Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0010 1011	address High	address Low	0000 FFFF	X-Or-Byte
Hex :	0xE4	0x2B	UH	AL	Group 8	X-Or-Byte
Dec :	228	43	UH	AL	Group 8	X-Or-Byte

The following applies to the functions: **Data 3:** Group 8:

7	6	5	4	3	2	1	0	bit
F52	F51	F50	F49	F48	F47	F46	F45	FX=1 means "function is on".

Function command group 9: F53-F60 (from version 4.0)

Format:

	header byte identifier		Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0101 0000	address High	address Low	0000 FFFF	X-Or-Byte
Hex :	0xE4	0x50	UH	AL	Group 9	X-Or-Byte
Dec :	228	80	UH	AL	Group 9	X-Or-Byte

The following applies to the functions: **Data 3:** Group 9:

7	6	5	4	3	2	1	0	bit
F60	F59	F58	F57	F56	F55	F54	F53	FX=1 means "function is on".

Function command group 10: F61-F68 (from version 4.0)

Format:

	header byte identifier		Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0101 0001	address	address	0000 FFFF	X-Or-Byte

			High	Low		
Hex :	0xE4	0x51	UH	AL	Group 10	X-Or-Byte
Dec :	228	81	UH	AL	Group 10	X-Or-Byte

The following applies to the functions: **Data 3:** Group 10:

7	6	5	4	3	2	1	0	bit
F68	F67	F66	F65	F64	F63	F62	F61	FX=1 means "function is on".

Description:

The locomotives 0 to 9999 can be addressed.

The locomotive address AH/AL is calculated as specified in 2.19.7.

Special features:

No.

3.42.5 *Function commands "binary states" from central version 4.0*

The function commands in the form of "binary states" for a locomotive are an extension in the RCN-212 to switching commands per locomotive address with up to 32738 functions. However, these are not "refreshed"

Each function is addressed individually, not in groups like the other function commands. Additionally, there is the option to set or delete all "short" or all binary states together.

Binary states 0 to 28 are not allowed. Binary states less than 128 must be sent in short form.

Binary states are used by some decoders in parallel or instead of functions F29 to F68.

3.42.5.1 Function command binary state short form:

Format:

	header byte identifier		Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0101 1110	address High	address Low	cbbb bbbb	X-Or-Byte
Hex :	0xE4	0x5E	UH	AL	bbbbbbb= binstate-number, c=0: delete c=1: set	X-Or-Byte
Dec :	228	94	UH	AL		X-Or-Byte

With bbbbbbb=0 all binstates are addressed short; data 3 = 00000000 deletes the binstates 29-127, 10000000 sets the binstates 29-127

3.42.5.2 Function command binary state long form:

Format:

	header byte identifier		Data 1	Data 2	Data 3	Data 4	X-Or-Byte
Binary :	1110 0100	0101 1111	address High	address Low	cbbb bbbb	bbbb bbbb	X-Or-Byte
Hex :	0xE4	0x5F	UH	AL	bbbbbbb= binstate-number L, c=0: delete c=1: set	Binstate-number H	X-Or-Byte
Dec :	228	95	UH	AL			X-Or-Byte

With bbbbbbbb bbbbbbb=0 all binstates (long & short) are addressed; data 3 = 00000000 with data 4=0 deletes all binstates 29-32767, data 3 = 10000000 and data 4=0 sets all binstates 29-32767; binstates H may only be 0 if all binstates are to be addressed; otherwise the short form is to be used.

3.42.6 *Set function status from central version 3.0 / version 3.6 / version 4.0*

The LZ100 central unit from version 3.0 saves the status of its functions as touch-sensitive or non-touch-sensitive for each locomotive address. XpressNet devices can query this status and design their user interface accordingly. This functionality is primarily intended for sounds.

As with the functions, groups 1 to 10 are distinguished by the identifier. New to version 3.6 are

group 4 (F13...20) and group 5 (F21...F28).

In version 4 also group 6 (F29-F36) to group 10 (F61-F68).

Set function status group 1:

Format:

	header byte identifier		Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0010 0100	address High	address Low	000S SSSS	X-Or-Byte
Hex :	0xE4	0x24	UH	AL	Group 1	X-Or-Byte
Dec :	228	36	UH	AL	Group 1	X-Or-Byte

The following applies to the

functions: **Data 3:** Group 1:

7	6	5	4	3	2	1	0	bit
0	0	0	S0	S4	S3	S2	S1	SX =1 means function is momentary

Set function status group 2:

Format:

	header byte identifier		Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0010 0101	address High	address Low	0000 SSSS	X-Or-Byte
Hex :	0xE4	0x25	UH	AL	Group 2	X-Or-Byte

Dec :	228	37	UH	AL	Group 2	X-Or-Byte

The following applies to the

functions: **Data 3:** Group 2:

7	6	5	4	3	2	1	0	bit
0	0	0	0	S8	S7	S6	S5	SX =1 means function is momentary

Set function status group 3:

Format:

	header byte identifier		Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0010 0110	address High	address Low	0000 SSSS	X-Or-Byte
Hex :	0xE4	0x26	UH	AL	Group 3	X-Or-Byte
Dec :	228	38	UH	AL	Group 3	X-Or-Byte

The following applies to the

functions: **Data 3:** Group 3:

7	6	5	4	3	2	1	0	bit
0	0	0	0	S12	S11	S10	S9	SX=1 means function is momentary

Set function status group 4 (from version 3.6):

Format:

	header byte identifier		Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0010 0111	address High	address Low	SSSS SSSS	X-Or-Byte
Hex :	0xE4	0x27	UH	AL	Group 4	X-Or-Byte
Dec :	228	39	UH	AL	Group 4	X-Or-Byte

The following applies to the

functions: **Data 3:** Group 4:

7	6	5	4	3	2	1	0	bit
S20	S19	S18	S17	S16	S15	S14	S13	SX=1 means function is momentary

Set function status group 5 (from version 3.6):

Format:

	header byte identifier		Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0010 1100	address High	address Low	SSSS SSSS	X-Or-Byte
Hex :	0xE4	0x2C	UH	AL	Group 5	X-Or-Byte

Dec :	228	44	UH	AL	Group 5	X-Or-Byte

The following applies to the

functions: **Data 3:** Group 5:

7	6	5	4	3	2	1	0	bit
S28	S27	S26	S25	S24	S23	S22	S21	SX=1 means function is momentary

Set function status group 6; F29-F36 (from version 4.0):

Format:

	header byte identifier		Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0010 1101	address High	address Low	SSSS SSSS	X-Or-Byte
Hex :	0xE4	0x2D	UH	AL	Group 6	X-Or-Byte
Dec :	228	45	UH	AL	Group 6	X-Or-Byte

The following applies to the

functions: **Data 3:** Group 6:

7	6	5	4	3	2	1	0	bit
S36	S35	S34	S33	S32	S31	S30	S29	SX=1 means function is momentary

Set function status Group 7: F37-F44 (from version 4.0):

Format:

	header byte identifier		Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0010 1110	address High	address Low	SSSS SSSS	X-Or-Byte
Hex :	0xE4	0x2E	UH	AL	Group 7	X-Or-Byte
Dec :	228	46	UH	AL	Group 7	X-Or-Byte

The following applies to the

functions: **Data 3:** Group 7:

7	6	5	4	3	2	1	0	bit
S44	S43	S42	S41	S40	S39	S38	S37	SX=1 means function is momentary

Set function status group 8; F45-F52 (from version 4.0):

Format:

	header byte identifier		Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0101 0010	address High	address Low	SSSS SSSS	X-Or-Byte
Hex :	0xE4	0x52	UH	AL	Group 8	X-Or-Byte

Dec :	228	82	UH	AL	Group 8	X-Or-Byte

The following applies to the functions: **Data 3:** Group 8:

7	6	5	4	3	2	1	0	bit
S52	S51	S50	S49	S48	S47	S46	S45	SX=1 means function is momentary

Set function status Group 9: F53-F60 (from version 4.0):

Format:

	header byte identifier		Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0101 0011	address High	address Low	SSSS SSSS	X-Or-Byte
Hex :	0xE4	0x53	UH	AL	Group 9	X-Or-Byte
Dec :	228	83	UH	AL	Group 9	X-Or-Byte

The following applies to the functions: **Data 3:** Group 9:

7	6	5	4	3	2	1	0	bit
S60	S59	S58	S57	S56	S55	S54	S53	SX=1 means function is momentary

Set function status Group 10: F61-F68 (from version 4.0):

Format:

	header byte identifier		Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0101 0100	address High	address Low	SSSS SSSS	X-Or-Byte
Hex :	0xE4	0x54	UH	AL	Group 10	X-Or-Byte
Dec :	228	84	UH	AL	Group 10	X-Or-Byte

The following applies to the functions: **Data 3:** Group 10:

7	6	5	4	3	2	1	0	bit
S68	S67	S66	S65	S64	S63	S62	S61	SX=1 means function is momentary

Description:

The locomotives 0 to 9999 can be addressed.

The locomotive address AH/AL is calculated as specified in 2.19.7.

Special features:

No.

3.42.7 *Set function refresh mode from central version 3.6*

Function refresh means the cyclical repetition of function data on the track. From central control unit version 3.6 onwards, it is possible to set which function data is refreshed. The factory setting is to refresh functions 0 to 8.
Functions from F29 onwards will not be refreshed.

Format:

	header byte	identifier	Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0101	0010 1111	address High	address Low	refresh mode	X-Or-Byte
Hex :	0xE4	0x2F	UH	AL	RF	X-Or-Byte
Dec :	228	47	UH	AL	RF	X-Or-Byte

Data 3: **R** refresh mode

Value	Refresh for
0	F0 ... F4
1	F0 ... F8
3	F0 ... F12
7	F0 ... F20
0xF	F0 ... F28

3.42.8 *double traction up to LZ version 2.3*

3.42.8.1 Install double traction

Format:

	header byte	identifier	Data 1	Data 2	X-Or-Byte
Binary :	1100 0011	0000 0101	Address 1	Address 2	X-Or-Byte
Hex :	0xC3	0x05	Address 1	Address 2	X-Or-Byte
Dec :	195	5	Address 1	Address 2	X-Or-Byte

Description:

The locomotives in data 1 and data 2 are combined in the control center to form a double traction, which means that a drive command to one of the locomotives is also sent to the other by the control center.

Locomotives 1 to 99 can be assembled together.

If the installation fails, the control unit sends one of the error messages described under 2.23.3.

Special features:

No.

3.42.8.2 dissolving double traction

Format:

	header byte	identifier	Data 1	Data 2	X-Or-Byte
Binary :	1100 0011	0000 0100	Address 1	Address 2	X-Or-Byte
Hex :	0xC3	0x04	Address 1	Address 2	X-Or-Byte
Dec :	195	4	Address 1	Address 2	X-Or-Byte

Description:

The double traction of the locomotives in data 1 and data 2 is resolved in the control center.

If the resolution fails, the control center sends one of the error messages described in 2.23.3.

Special features:

No.

3.42.9 *Double traction from central version 3.0*

3.42.9.1 Install double traction

Format:

	header byte identifier		Data 1	Data 2	Data 3	Data 4	X-Or-Byte
Binary :	1110 0101	0100 0011	Adr High 1	Adr Low 1	Adr. High 2	Adr Low 2	X-Or-Byte
Hex :	0xE5	0x43	AH1	AL1	AH 2	AL 2	X-Or-Byte
Dec :	229	67	AH1	AL1	AH 2	AL 2	X-Or-Byte

Description:

The locomotives in data 1/2 and data 3/4 are combined in the control center to form a double traction, which means that a drive command to one of the locomotives is also sent to the other by the control center.

The locomotive addresses AH/AL are calculated as specified in 2.19.7.

If the installation fails, the control unit sends one of the error messages described under 2.24.

Special features:

The command replaces the old double traction commands, which are no longer supported in later central unit versions.

3.42.9.2 dissolving double traction

Format:

	header byte identifier		Data 1	Data 2	Data 3	Data 4	X-Or-Byte
Binary :	1110 0101	0100 0011	Adr High 1	Adr Low 1	0000 0000	0000 0000	X-Or-Byte
Hex :	0xE5	0x43	AH1	AL1	0x00	0x00	X-Or-Byte
Dec :	229	67	AH1	AL1	0x00	0x00	X-Or-Byte

Description:

The locomotive in data 1/2 is removed from the double traction in which it is integrated. This also removes the double traction in the control center.

The central unit can recognize that this is a DTR resolution by the second locomotive address, which is 0 in this case.

The locomotive address AH/AL is calculated as specified under 2.1.15.

If the installation fails, the control unit sends one of the error messages described under 2.24.

Special features:

The command replaces the old double traction commands, which are no longer supported in later central unit versions.

3.42.10 *Multiple traction from central version 3.0*

3.42.10.1 Add a locomotive to a multiple traction or create an MTR

A locomotive can be added to a multiple traction train (MTR) if it is not already included in any other MTR. If this locomotive is the first locomotive, an MTR is automatically created.

In addition, the insertion direction can be defined so that a locomotive can sit the "wrong" way around in an MTR but still travel in the correct direction. This is decided by a bit in the identifier (R).

Format:

	header byte	identifier	Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0100 000R	address High	Address Low	MTR	X-Or-Byte
Hex :	0xE4	0x40 + R	UH	AL	MTR	X-Or-Byte
Dec :	228	64 + R	UH	AL	MTR	X-Or-Byte

Description:

R: R = 0 means that the locomotive is not inserted into the MTR inverted. This means that if the MTR moves forward, the locomotive also moves forward.

R =1 means that the direction of travel of the locomotive is inverted.

Data 1 and data 2 specify the locomotive address from 1..9999 that is to be inserted into the MTR. The locomotive address AH/AL is calculated as specified in 2.19.7.

MTR: This is the MTR base address in the range 1 to 99.

Special features:

By definition, a locomotive cannot be inserted into a multiple traction that has the same address.

3.42.10.2 Remove locomotive from a multiple traction or delete MTR

A locomotive can be removed from an MTR if it is a member of this MTR. When the last locomotive from an MTR is removed, the MTR is also deleted from the central station.

Format:

	header byte	identifier	Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0100 0010	address High	Address Low	MTR	X-Or-Byte
Hex :	0xE4	0x42	UH	AL	MTR	X-Or-Byte
Dec :	228	66	UH	AL	MTR	X-Or-Byte

Description:

Data 1 and data 2 specify the locomotive address from 1..9999 that is to be removed from the MTR.

The locomotive address AH/AL is calculated as specified in 2.19.7.

MTR: This is the base address or MTR address in the range 1 to 99 under which the multiple traction can be driven.

Special features:

No.

3.43 Programming on Main from central version 3.0

Programming on Main means that CVs of a receiver can be changed while the locomotive is on the normal track. A programming track is not necessary in this case. However, the address of a receiver cannot be changed with this, as this must be used in the programming command.

Control panels that do not support Programming on Main send “Command not present” to the XpressNet device.

In contrast to programming on the programming track, CVs 1..1024 are possible here, but XpressNet devices should not allow CVs that result in an address change, because then a receiver would no longer be able to listen to data packets sent later if it evaluates the transmission (which is not allowed).

3.43.1 *Writing Programming on Main Byte*

Format:

	header	identifier	Data 1	Data 2	Data 3	Data 4	Data 5	X-Or-Byte
Binary :	1110 0110	0011 0000	Address High	address Low	1110 11CC	CCCC CCCC	DDDD DDDD	X-Or-Byte
Hex :	0xE6	0x30	UH	AL	0xEC + C	CV	D	X-Or-Byte
Dec :	230	48	UH	AL	236 + C	CV	D	X-Or-Byte

Description:

Data 1 and Data 2 specify the locomotive address from 1..9999 to which the byte programming refers.

The locomotive address AH/AL is calculated as specified in 2.19.7.

Since CVs from 0..1023 are possible (=10 bits), the upper 2 bits (MSBs) are written to data 3.
The rest of the CV address (the 8 LSBs) are in data 4.

The value of this CV to be programmed is in data 5.

The CV address is sent as it appears on the track, i.e. decremented by one.

Special features:

CVs that refer to recipient addresses should not be used.

3.43.2 *Programming on Main Byte reading (from version 3.6)*

Format:

	header	identifier	Data 1	Data 2	Data 3	Data 4	Data 5	X-Or-Byte
Binary :	1110 0110	0011 0000	Address High	Address Low	1110 01CC	CCCC CCCC	0000 0000	X-Or-Byte
Hex :	0xE6	0x30	UH	AL	0xE4 + C	CV	0x00	X-Or-Byte
Dec :	230	48	UH	AL	228 + C	CV	0	X-Or-Byte

Description:

Data 1 and data 2 specify the locomotive address from 1..9999 to which the byte reading refers. The

locomotive address AH/AL is calculated as specified in 2.19.7.

Since CVs from 0..1023 are possible (=10 bits), the upper 2 bits (MSBs) are written to data 3.
The rest of the CV address (the 8 LSBs) are in data 4.

Historically, this is actually the byte verify command. That is why there is still the byte 'Data 5', but this no longer has any meaning. That is why the value 0 is entered here.

The CV address is sent as it appears on the track, i.e. decremented by one.

Special features:

No.

3.43.3 *Writing Programming on Main Bit*

Format:

	header	identifier	Data 1	Data 2	Data 3	Data 4	Data 5	X-Or-Byte
Binary :	1110 0110	0011 0000	Address High	Address Low	0111 11CC	CCCC CCCC	1111 WBBB	X-Or-Byte
Hex :	0xE6	0x30	UH	AL	0x7C + C	CV	0xF0 + WB	X-Or-Byte
Dec :	230	48	UH	AL	124 + C	CV	240 + WB	X-Or-Byte

Description:

Data 1 and Data 2 specify the locomotive address from 1..9999 to which the bit programming refers.

The locomotive address AH/AL is calculated as specified in 2.19.7.

Since CVs from 0..1023 are possible (=10 bits), the upper 2 bits (MSBs) are written to data 3.
The rest of the CV address (the 8 LSBs) are in data 4.

The bit value to be programmed is in data 5 and is calculated as follows: W is

the bit value 0 or 1.

The bits B2, B1, B0 indicate the position of the bit in the CV (bit position 0 to bit position 7). The

CV address is sent as it appears on the track, ie decremented by one.

Special features:

CVs that refer to recipient addresses should not be used.

3.44 Address search commands from central unit version 3.0

Due to the introduction of multiple traction and extended stack handling in the control centers, it has become necessary for XpressNet devices to also search for locomotive addresses in order to achieve a comfortable user interface.

3.44.1 *Address request member of a multiple traction*

The distinction between forward and backward search is made via the identifier.

Format:

	header byte	identifier	Data 1	Data 2	Data 3	X-Or-Byte
Binary :	1110 0100	0000 00RR	MTR	address High	Address Low	X-Or-Byte

Hex :	0xE4	0x01 + R	MTR	UH	AL	X-Or-Byte
Dec :	228	1 + R	MTR	UH	AL	X-Or-Byte

Description:

In order to have quick access to the locomotives in an MTR, e.g. to be able to switch functions, the central unit supplies the next address that follows the requested one (forward search) or precedes it (backward search) based on this request.

For control panels including version 3.8, only the forward search is defined.

Identifier = 0x01: (RR=01) means forward search

Identifier = 0x02: (RR=10) means backward search

Data 1 specifies the MTR base address in the range 1..99 to which the search refers.

Data 2 and data 3 specify the locomotive address from 1..9999 for which the following or previous address is to be searched.

The locomotive address AH/AL is calculated as specified in 2.19.7. MTR:

This is the base address or MTR address in the range 1 to 99 under which the multiple traction can be driven.

The result of the search is sent to the device in the response as described in 2.20.

Special features:

No.

3.44.2 *Address request multiple traction*

The distinction between forward and backward search is made via the identifier.

Format:

	header byte	identifier	Data 1	X-Or-Byte
Binary :	1110 0010	0000 0RRR	MTR	X-Or-Byte
Hex :	0xE2	0x03 + R	MTR	X-Or-Byte
Dec :	226	3 + R	MTR	X-Or-Byte

Description:

This command causes the central unit to send the XpressNet device the next base address of an MTR that follows (forward search) or precedes (backward search) the requested MTR.

For control panels including version 3.8, only the forward search is defined.

Identifier = 0x03: (RRR=011) means forward search

Identifier = 0x04: (RRR=100) means backward search

MTR: This is the base address or MTR address in the range 1 to 99 under which the multiple traction can be driven.

The result of the search is sent to the device in the response as described in 2.22.

Special features:

No.

3.44.3 *Address request locomotive in central stack*

The distinction between forward and backward search is made via the identifier.

Format:

	header byte	identifier	Data 1	Data 2	X-Or-Byte
Binary :	1110 0011	0000 01RR	address High	Address Low	X-Or-Byte
Hex :	0xE3	0x05 + R	UH	AL	X-Or-Byte
Dec :	227	5 + R	UH	AL	X-Or-Byte

Description:

The locomotive address that is stored in the central stack behind (forward search) or in front of the locomotive address (backward search) in data 1/2 is sent to the XpressNet device.

For control panels including version 3.8, only forward search is defined.
From LZV200 version 4, backward search is also possible.

Identifier = 0x05: (RR=01) means forward search

Identifier = 0x06: (RR=10) means backward search

Data 1 and data 2 specify the locomotive address from 1..9999 for which the following or previous address is to be searched. The locomotive address AH/AL is calculated as specified in 2.19.7.

Address 0 reports the first locomotive entered in the stack.

The result of the search is sent to the device in the response as described in 2.20.

Special features:

No.

3.44.4 *Delete locomotive from central stack*

Format:

	header byte	identifier	Data 1	Data 2	X-Or-Byte
Binary :	1110 0011	0100 0100	address High	Address Low	X-Or-Byte
Hex :	0xE3	0x44	UH	AL	X-Or-Byte
Dec :	227	68	UH	AL	X-Or-Byte

Description:

Data 1 and data 2 specify the locomotive address from 1..9999, which is to be deleted in the central stack.

The locomotive address AH/AL is calculated as specified in 2.19.7.

Special features:

The problem of the central stack being full occurs primarily with centrals with limited hardware, which cannot then save all the locomotives that have ever been called up with all the data. The central stack is used to send the data from these locomotives to the track.

The XpressNet device that deleted the locomotive in the stack should ensure that it can continue working with another locomotive so that the deletion is successful. Otherwise, the locomotive that was just deleted would be added again immediately.

4 Central units and their properties

Since not all control panels support all features, an XpressNet device should use the version number and, if applicable, the control panel ID to determine which control panel it is working with, which commands it understands and, if a menu structure exists, respond to these.

The table is shown from the perspective of an XpressNet device.

	locomotives	F-status	DTR	MTR	switching	Rückm.	PoM	Prog.
LZ100V2.3	**0..99**	**no**	**Yes**	**no**	**Yes**	**Yes**	**no**	**Yes**
LZ100V3.0	**0..9999**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
LH200	**0..9999**	**no**	**no**	**no**	**Yes**	**no**	**Yes**	**no**
Compact	**0.99**	**no**	**Yes**	**no**	**Yes**	**no**	**no**	**Yes**
LZV200	**0..9999**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

If you specify MTR for the LH200 central unit, for example, no means that multiple traction cannot be set up from the XpressNet device. However, the LH200 as the central unit can do this. If a locomotive request from an XpressNet device encounters such a multiple traction, it can be driven but not resolved. Likewise, no further locomotives can be added from the XpressNet device.

5 Command overview central to device

They mean: P: parity bit, GA: XpressNet device address

The meaning of the respective data bytes is described in the corresponding chapters.

Chapter	V3	command	Call	header	Data1	data2	data3	data4	data5	data6	data7
2.1	V3	Normal request	P+0x40+								
2.2	V3	acknowledgement	P+0x00+								
2.3	V3	TBD	P+0x20+								
2.4.1	V3	BC Alles An	0x60	0x61	0x01	0x60					
2.4.2	V3	BC Alles Aus	0x60	0x61	0x00	0x61					
2.4.3	V3	BC All Locomotives Off	0x60	0x81	0x00	0x81					
2.4.4	V3	BC programming mode	0x60	0x61	0x02	0x63					
2.4.5	V3	BC feedback	0xA0	0x40 + N	ADR_1	DAT_1	ADR_2	DAT2	etc.	etc.	X-Or
2.4.6	3.8	BC feedback from W1025	0xA0	0x43	ADRH	ADRL	DAT	X-Or			
2.4.7	3.8	BC model time	0x60	0x63	0x03	dddhhhhh	Min&stop	X-Or			
2.5.1	V3	P-Info short circuit	P+0x60+	0x61	0x12	X-Or					
2.5.2	V3	P-Info No data	P+0x60+	0x61	0x13	X-Or					
2.5.3	V3	P-Info Busy	P+0x60+	0x61	0x1f	X-Or					
2.5.4	V3	P-Info ready	P+0x60+	0x61	0x11	X-Or					
2.5.5	V3	P-Info data 3 bytes	P+0x60+	0x63	0x10	EE	DAT	X-Or			
2.5.6	V3.6	P-Info CV1-255 and 1024	P+0x60+	0x63	0x14	CV	DAT	X-Or			
2.5.7	V3.6	P-Info CV256 - 511	P+0x60+	0x63	0x15	CV	DAT	X-Or			
2.5.8	V3.6	P-Info CV512 - 767	P+0x60+	0x63	0x16	CV	DAT	X-Or			
2.5.9	V3.6	P-Info CV768 - 1023	P+0x60+	0x63	0x17	CV	DAT	X-Or			
2.6	3.8	Report Service Variable	P+0x60+	0x63	0x20	SV#	SVval	X-Or			
2.7.2	V3	Software LZ from 3.0	P+0x60+	0x63	0x21	DAT1	DAT2	X-Or			
2.8	V3	Status Center	P+0x60+	0x62	0x22	DAT	X-Or				
2.9	3.8	Ext. Version Info	P+0x60+	0x67	0x23	ZBldH	ZBldL	RMver	RMBldH	RMBldL	RMver
2.10	3.8	Report PoM Result	P+0x60+	0x64	0x24	ADRH	ADRL	PoMval	X-Or		
2.11	3.8	Report model time	P+0x60+	0x64	0x25	H&DOW	min	factor	X-Or		

2.12	V3	transmission error	P+0x60+	0x61	0x80	X-Or					
2.13	V3	Busy Central	P+0x60+	0x61	0x81	X-Or					
2.14	V3	Command not available	P+0x60+	0x61	0x82	X-Or					
2.15	V3	switching information	P+0x60+	0x42	ADR	DAT	X-Or				
2.16	V3.8	Switching information from W1025	P+0x60+	0x4	ADRH	ADRL	DAT	X-Or			
2.19.1	V3	Normal locomotive info from V3	P+0x60+	0xE4	identifier	speed	FKT0	FKT1	X-Or		
2.19.2	V3.6	F-state F13 ... F28	P+0x60+	0xE3	0x52	F 13-20	F 21-28	X-Or			
2.19.3	V4.0	F-state F29 ... F68	P+0x60+	0xE6	0x53	F 29-36	F 37-44	F 45-52	F 56-63	F 64-68	X-Or
2.19.4	V3	MTR member from V3	P+0x60+	0xE5	identifier	speed	FKT0	FKT1	MTR	X-Or	
2.19.5	V3	MTR base address from V3	P+0x60+	0xE2	identifier	speed	X-Or				
2.19.6	V3	Locomotive is in DTR from V3	P+0x60+	0xE6	identifier	speed	FKT0	FKT1	ADR High	ADR Low	X-Or
2.19.7	V3	Locomotive occupied from V3	P+0x60+	0xE3	0x40	ADR High	ADR Low	X-Or			
2.19.8	V3	functional status from V3	P+0x60+	0xE3	0x50	STAT 0	STAT 1	X-Or			
2.20	V3.6	F-status F13 to F28	P+0x60+	0xE4	0x51	STAT 2	STAT 3	RSTAT	X-Or		
2.21	V4	F-status F29 to F68	P+0x60+	0xE6	0x54	STAT6	STAT7	STAT8	STAT9	STAT10	X-Or
2.22	V3	Locomotive search results from V3	P+0x60+	0xE3	0x30 + K	ADR High	ADR Low	X-Or			
2.24	V3	error message from V3	P+0x60+	0xE1	0x80 + F	X-Or					

6 Command overview device to control center

The meaning of the respective data bytes is described in the corresponding chapters.

Chapter	V3	command	header	identifier	Data1	data2	data3	data4	data5	data6
3.1	V3	acknowledgement	0x20	0x20						
3.2	V3	Everything On	0x21	0x81	0xA0					
3.3	V3	Everything's Over	0x21	0x80	0xA1					
3.4	V3	Stop all locomotives	0x80	0x80						
3.7	V3	Stopping a locomotive from V3	0x92	ADR High	ADR Low	X-Or				
3.9	V3	Prog.-Reading Register	0x22	0x11	REG	X-Or				
3.11	V3.6	Prog.-Read CV1-255; 1024	0x22	0x18	CV low	X-Or				
3.12	V3.6	Prog.-Read CV256-511	0x22	0x19	CV low	X-Or				
3.13	V3.6	Prog.-Reading CV512-767	0x22	0x1A	CV low	X-Or				
3.14	V3.6	Prog.-Reading CV768-1023	0x22	0x1B	CV low	X-Or				
3.15	V3	Prog.-Reading Paging	0x22	0x14	CV	X-Or				
3.16	V3	Request program result	0x21	0x10	0x31					
3.17	V3	Prog.-Schreiben Register	0x23	0x12	REG	DAT	X-Or			
3.19	V3.6	Prog.-Schr. CV1-255; 1024	0x23	0x1C	CV low	DAT	X-Or			
3.20	V3.6	program script CV256-511	0x23	0x1D	CV low	DAT	X-Or			
3.21	V3.6	program code CV512-767	0x23	0x1E	CV low	DAT	X-Or			
3.22	V3.6	program script CV768-1023	0x23	0x1F	CV low	DAT	X-Or			
3.23	V3	Programming Paging	0x23	0x17	CV	DAT	X-Or			
3.24	V3	Request software version	0x21	0x21	0x00					
3.25	V3.8	Extended central version	0x21	0x23	0x02					
3.26	V3	Request Status Headquarters	0x21	0x24	0x05					
3.27		set start mode	0x22	0x22	00000M0	X-Or				
3.28	V3.8	SV reading	0x22	0x25	SV#	X-Or				
3.29	V3.8	SV write	0x23	0x26	SV#	SV value	X-Or			
3:30	V3.8	Request PoM result	0x21	0x27	0x06					
3.31	V3.8	trigger a control panel reset	0x21	0x28	0x09					

3.32	V4.0	Send device info	0x24	0x29	SW	lever	type	X-Or		
3.33	V3.8	Request model time	0x21	0x2A	0x0B					
3.34	V3.8	set model time	0x24	0x2B	DOW&d	min	factor	X-Or		
3.35	V4.0	Pause model time	0x21	0x2D	0x0C					
3.36	V4.0	start model time	0x21	0x2C	0x0D					
3.37	V3	Request switching information	0x42	ADR	Nibble	X-Or				
3.38	V3.8	Request switching information up to W2048	0x43	ADRH	ADRL	DAT	X-Or			
3.39	V3	switching command up to W1024	0x52	ADR	DAT	X-Or				
3.40	V3.8	switching command up to W2048	0x53	ADRH	ADRL	DAT	X-Or			
3.41.3	V3	Request locomotive data from V3	0xE3	0x00	ADR High	ADR Low	X-Or			
3.41.4	V3	Request Fkt status from V3	0xE3	0x07	ADR High	ADR Low	X-Or			
3.41.5	V3.6	Fkt-Status anf. F13-F28	0xE3	0x08	ADR High	ADR Low	X-Or			
3.41.6	V4.0	Fkt-Status anf. F29-F68	0xE3	0x0A	ADR High	ADR Low	X-Or			
3.41.7	V3.6	Fkt-Status anf. F13-F28	0xE3	0x09	ADR High	ADR Low	X-Or			
3.41.8	V4.0	Fkt-Status anf. F29-F68	0xE3	0x09	ADR High	ADR Low	X-Or			
3.42.3	V3	Locomotive driving command from V3	0xE4	identifier	ADR High	ADR Low	speed	X-Or		
3.42.4	V3-V4	Lok function command	0xE4	identifier	ADR High	ADR Low	group	X-Or		
3.42.5.1	V4	Binary states short setting	0xE4	0x5E	ADR	ADR	D&FKT	X-Or		
3.42.5.2	V4	Binary states long setting	0xE5	0x5F	ADR	ADR	D&FKTL	FKTH	X-Or	
3.42.6	V3-4	set function status	0xE4	identifier	ADR High	ADR Low	group	X-Or		
3.42.7	V3.6	Set Func.refresh mode	0xE4	0x2F	ADR High	ADR Low	mode	X-Or		
3.42.9	V3	DTR commands	0xE5	0x43	ADR1 H	ADR1 L	ADR2 H	ADR2 L	X-Or	
3.42.10.1	V3	Add locomotive to MTR from V3	0xE4	0x40 + R	ADR High	ADR Low	MTR	X-Or		
3.42.10.2	V3	Remove locomotive from MTR from V3	0xE4	0x42	ADR High	ADR Low	MTR	X-Or		
3.43.1	V3	Write Prog. on Main Byte	0xE6	0x30	ADR High	ADR Low	0xEC + C	CV	DAT	X-Or
3.43.2	V3.6	Prog. on Main Byte read	0xE6	0x30	ADR High	ADR Low	0xEA + C	CV	DAT	X-Or
3.43.3	V3	Prog. on Main Bit from V3	0xE6	0x30	ADR High	ADR Low	0x7C + C	CV	DAT	X-Or
3.44.1	V3	Address search locomotive in Mtr from V3	0xE4	0x01 + R	MTR	ADR High	ADR Low	X-Or		
3.44.2	V3	Address search MTR from V3	0xE2	0x03 + R	MTR	X-Or				
3.44.3	V3	Stack search locomotive from V3	0xE3	0x05 + R	ADR High	ADR Low	X-Or			
3.44.4	V3	Delete locomotive from stack from V3	0xE3	0x44	ADR High	ADR Low	X-Or			