Lenz
ELEKTRONIK GMBH

Command set XpressNet V3 with Interface LI100F

including XBus up to version 2.3

11/2001

1 Introduction

This documentation describes the commands used to control a model railrod layout with a Computer and the interface LI100F. Beside the new XpressNet commands the older commands used up to central unit version 2.3 are also included as well as future expansions which are partially implemented in some central units and some hand held controllers. Differences are made here between central units up to version 1.5, up to version 2.3 and version 3.0 or later.
XpressNet requires an exact timing of the attached devices and the data transmission is not like commonly used on PC´s. For these reasons, the LI100F was developed to realize a simple control mechanism over the layout with a PC. The LI100F is the specified XpressNet device which performs the bus timing and allows asynchronous communication with the PC. The LI100F passes the commands from the PC in its contents completely unmodified to the central unit (under consideration of the XpressNet timing) and vice versa the central units answers immediatley they are available. Because the central unit is the time conditioning system, the LI100F must be prepared to enable PC commands only if this is possible and on the other hand send informations from the central unit immediately to the PC so that no data is lost. This is guaranteed by using a serial port of the PC with following setting:
- baudrate: 9600 baud or 19200baud (19200baud only adjustable in LI100F),
- 8 databits, 1 startbit, 1 stopbit, no paritybit
- hardware handshake

The hardware handshake is done in a way that LI100F enables or stops transmission from the PC with the CTS signal. The RTS signal from the PC is not interpreted so that every information is send in the moment it occurs.
Although the PC has much more capabilities of data transmissions, it has to adapt its behaviour to the conditions needed for a model railroad layout.

1.1 XpressNet-Hardware

It is not needed, but useful to know some details of the XpressNet structure. Via XpressNet all controlling devices of the Digital plus system are connected to the central unit. The XpressNet is a serial bus conform to RS485 standard in half duplex with differential data transmission and following settings:

1 startbit (0), 9 databits, 1 stopbit (1), no paritybit
baudrate : 62,5 kbaud

A paritybit is only used in the adressbyte of the central unit to an attached device. This adressbyte is not visible to the PC and therefore not documented here (but neccessary for XpressNet bus communication). It is the byte which is preceding the headerbyte. Paritybit is Bit 7 of the adressbyte and is build in a way that in this adressbyte the number of 1-bits is even.
9 databits are used to differentiate between adressbyte (bit 9 is 1) and a databyte (bit 9 is 0) for a device. Pinout of the XpressNet:

L – Supply 12VDC
M – Ground
A – receive/transmit non inverting
B – receive/transmit inverting

Care must be taken not to swap any of these pins, especially A and B. No communication will be the result. The complete length of the bus can be up to 1000meters. It is also not required to have a dedicated bus structure. Branch lines, star or ring topologies are possible. Only if transmission problems arise in large wirings or a bad environment, it can be neccessary to use twisted pair cabling and termination of the bus with a 120 ohm resistor. The central unit LZ100 contains already one resistor, so that only one additional resistor must then be connected on the farest point of the bus between A and B.
Communication on the XpressNet is done in a way that the central unit as the only master device adresses all attached devices (slaves) periodically. The interface LI100F is one of these devices. After being adressd, the slave can send one command sequence to the central unit which sends back nothing or one answer sequence. After this is done, the central unit adresses the next slave. It is characteristic that the central unit completes a command which the slave has begun before the next slave can do something.
This sequence is also used if a slave sends a command which results in a message to all slaves (broadcast to all slaves). If e.g. an emergency stop is send, the central unit acknowledges this immediatly with a broadcast and only then adresss the next one. Depending on the desired action, the time until this next adressing can last very long if e.g. programming functions are activated by a slave.
Differing from this command-answer-scheme, the central unit sends informations which a device has not explicitly requested but must notice to react in any way (display etc.).
To complete a command sequence, a xor-byte is calculated and transferred as last byte. The receiver must check this byte.

1.2 Unrequested Informations

Unrequested informations for one or all devices (also for the PC) are send if a change on the layout must be given to the devices so that they can adapt their behaviour for this new situation. Unrequested informations are send as broadcast if all slaves should receive them or formatted as an answer if only one slave is involved. It is typical that a slave does not request these informations, but receives them in an unpredictable moment and must handle them correct. Unrequested informations are:

Broadcast „Everything On“ (to all slaves)
Broadcast „Everything Off“ (to all slaves)
Broadcast „All Locos Stoppped“ (to all slaves)
Broadcast „Programming mode“ (to all slaves)
Broadcast „Feedback information“ (to all slaves)
Information „Locomotive occupied“ (only to that slave which has controlled the loco)
Information „Double header occupied“ (only to that slave which has controlled the double header)

The informations „transmission error“, „central unit busy“, „double header error“ and „command unknown“ are not unrequested informations because they belong directly to the command which a slave has sent to the central unit.

1.3 Locomotive Adresses In The Digital Plus-System

When introducing the version 3.0 of the Lenz Digital plus central units, one new feature was the larger amount of locomotive adresss. This had consequences for the data transmission on the XpressNet. The previously used locomotive adresss from 0 to 99 can be transmitted using one byte which can represent 256 different values (8 bit). This is not enough for adresss 0 to 9999 so that two bytes must be used. This document uses the hexadecimal number representation if this leads to a clearer description.
NMRA standard differentiates between short and long locomotive adresss. Short adresss (one byte) go from 0 to 127dec., long adresss (two byte) from 0 to 16383dec. Short adresss are 7 bit in size, long adresss 14 bit. Trying not to confuse the user, we defined in our Digital plus system the following adress ranges: short adresses go from 0 to 99, long adresses from 100 to 9999. We do not overlap the adress areas.
The representation of the short adresses is easy: 0 to 99 is hexadecimal 0x00 to 0x63. For the long adresss, the upper two bits in the highbyte are set, so that this results in an offset of 0xC000. With our adress range of 100 to 9999, the locomotive adresses on XpressNet are 0xC064 up to 0xE70F.
For the new version 3 commands, we use in any case two bytes for a locomotive adress so that two bytes must be transmitted even if the adresss is short.
If a two-byte locomotive adress is neccessary in the following description (named AH and AL for adress highbyte and lowbyte) always the following must be send:
for locomotive adress 0 to 99: 0x0000 ... 0x0063
for locomotive adress 100 to 9999: 0xC064 ... 0xE707
(=0x0064 + 0xC000 ... 0x270F + 0xC000)

1.4 Programming Hints

The model railroad is digitally controlled with a central unit which produces a track signal conform to the NMRA definitions. This signal is received by locomotive decoders which controls the engine. In the case of our Digital plus system, this central unit is always present on the layout and not emulated with a PC software.
The different control units like hand held controllers, tower cab or the interface are connected to the central unit through a separat bus, the XpressNet. Over XpressNet, the central unit takes the information which locomotive should run with new speed, which turnout should be switched etc. The central unit converts these informations to the track signal format and sends them to the receivers on the track, but does not know if a receiver is present on the layout.
This means that the central unit receives one information through XpressNet (e.g. loco no. 10 to speed 4) and puts it periodically on the track to allow a decoder to receive the information.
For these purposes, the central unit holds a dataset for a locomotive which is periodically send and which can be modified or requested from a XpressNet device.
The communication scheme on the XpressNet requires therefor an exact timing which the devices have to follow. This is normally not so easy in a PC environment because of the neccessity of directly programming the uart chips to ensure the timing and operationsmode of XpressNet. To ease this communication, the LI100F is used. The LI100F takes over the time

critical part of the communication and allows an easy asynchronous communication with the PC through a serial port. But this does not mean that the model railroad layout can be controlled as fast as the high computing power of todays PC´s will allow. The division of time is done by the central unit and in many cases by the track signal itself which will be then the limiting factor for data transmission. Therefor every connected device (also the PC) must take care of this. The programmer of a PC software must always have in mind that it is about to control a real time system in which a given command must be checked to ensure that it was accepted.
It can be seen in some new commands e.g. request for locomotive information, that the answer of the central unit does not contain this locomotive adress. But in the given system it´s clear that this answer belongs to the just sended request. The central unit ensures that no other device can modify the data in the meantime.
The LI100F can be used with 9600 baud or 19200 baud. This seems to be not so much. But for comparison: XpressNet is running with 62500baud and the track output reaches in the middle a datarate of 4500baud.
Every command consists of a header byte, one or more databytes and a checksum byte.
The headerbyte is split into two parts, each with 4 bit (nibble). In the upper nibble a part of the command function is coded, in the lower nibble the number of databytes is given. The headerbyte and the xor-byte (checksum byte) do not count, so a maximum of 15 databytes can be send in one command. The databytes complete the command which has begun with the high nibble of the header.
The command ends with a checksum byte. This byte is calculated as xor of all previous bytes inclusive the headerbyte. On receipt, this xor byte must be checked to become 0. Otherwise this gives an indication that there were errors during transmission.

1.5 Messages Of LI100F

The commands from the PC to the central unit are split into two groups. The commands of the first group have a direct reaction (answer) of the central unit without adressing of the next slave, e.g. requests for locomotive infos and also commands from the PC which result in a broadcast. The commands of the second group do not have a central unit reaction, e.g. setting a function. To allow an assignment in the PC to the sended command, in this and only in this case the LI100F generates a message. So in any way the PC gets a reaction after sending a command.
The following messages can occur if no data from central unit is available. Please have in mind, that if the PC sends a command which results in a broadcast, this broadcast is seen by LI100F as answer from the central unit and therefor does not generate a message.

headerbyte	message	xor	description
01	01	00	error between interface and PC (timeout during data transmission PC to LI100F)
01	02	03	error between interface and central unit (timeout during data transmission LI to central unit)
01	03	02	unknown error (central unit adresses LI100F with acknowledge for error)
01	04	05	Command is sent to central unit or central unit restarts to adress LI100F (after timeout)
01	05	04	Central unit does not adress LI100F (timeout)
01	06	07	Buffer overrun in LI100F

Explanations:

01 / 01 / 00:

With the first byte of a command which the PC sends, the number of following bytes is calculated. If this number of bytes is not send in a defined time, this timeout message is generated.

01 / 02 / 03:

The reaction of the central unit for a given command must arrive in a defined time. Otherwise this central unit-timeout message is generated.

01 / 03 / 02:

If a transmission error occurs on XpressNet between LI100F and central unit, the LI100F is adressed again for repetition. This must be acknowledged. This is the corresponding message to the PC, but the PC must not answer to it. In most cases, this can be ignored in a PC program. If this message occurs often, the cabling should be checked. This message can also occur, if the xor-byte is calculated incorrect.

01 / 04 / 05:

This is the acknowledge to the PC if a command sent to the central unit does not have any answer. The LI100F recognizes this when the central unit adresses the next slave. It must be noticed that there is no possibility to check for the PC that a given command is put out from the central unit on the track. This message only means that the LI100F successfully sent the command to the central unit. This message is not send in the case the PC gives a command which results in a broadcast (e.g. „Everything On“). This resulting broadcast is seen as answer from the LI100F. The PC has to check the content of the broadcast to ensure that the desired action has happened. It is possible that the broadcast for „Everything On“ is „Everything Is Off“, because central unit could not switch on track voltage.

This message is also send, if the central unit restarts to adress the LI100F (after an central unit adressing timeout).

01 / 05 / 04:

The central unit adresses all attached slaves in a defined time interval. If this does not happen, the LI100F sends this message to the PC and disables CTS signal. If the central unit returns to ist normal adressing rate, the previously described message is send and normal operation goes on.

01 / 06 / 07:

To avoid loss of data from central unit to LI100F, a buffer is used in the LI100F. If this buffer can not be send to the PC and therefor overflows, this message is generated.

The LI100F messages described here differ a little between the ones used up to now in the LI100. If neccessary, existing PC software must be adapted to fulfill the demands described in this document.

1.6 Read Out Version Number Of LI100F

Read out LI100F version number is an action only between LI100F and PC. The command structure and the answer still correspond to the scheme described in chapter 2.

Command to read out version- and codenumber:

	headerbyte	xor-byte
binary:	1111 0000	1111 0000
Hex :	0xF0	0xF0
Dez :	240	240

Answer of LI100F:

	headerbyte	data 1	data 2	xor-byte
binary:	0000 0010	VVVV VVVV	CCCC CCCC	Xor-byte
Hex :	0x02	VV	CC	Xor-byte
Dez :	2	VV	CC	Xor-byte

Description:

VV is the version number hexadecimal in bcd-presentation. CC is the codenumber of LI100F hexadecimal in bcd-presentation. Example:
Answer = 0x02 0x30 0x01 0x33
Versionnumber 3.0, codenumber 01

Remarks:

This is the only command the LI100F interpretes. Everything which does not correspond to this command is send to the central unit and will not be checked in any way.

2 Data transmission Between Central Unit And PC

2.1 Central Unit To PC

Command structure:
The central unit sends data to the LI100F which immediately send them to the PC. Data can be requested from the PC or is unrequested due to changes on the layout.
A headerbyte is send, one up to max. 15 databytes and one xor-byte. In the lower nibble of the headerbyte is the number of databytes which follow.

Definitions for the following command descriptions:

N = number of following databytes
XA = XpressNet device adress

The command set is given binay, decimal and hexadecimal.
Not all central units support all commands. This must be noted to avoid endless loops. Refer also to chapter 2.1.6 , unknown command).

2.1.1 Broadcast

The group „broadcast“ allows the central unit to send informations to all slaves at the same time. A broadcast is send several times to ensure that every slave can receive it. Some commands trigger such a broadcast (e.g. „Everything Off“). A device which initiates a broadcast has to take care in its program when it receives this broadcast itself (e.g. broadcast „Programming mode“.

2.1.1.1 BC „Everything Is On“

Format:

	Headerbyte	Data 1	Xor-byte
Binary:	0110 0001	0000 0001	0110 0000
Hex :	0x61	0x01	0x60
Dez :	97	1	96

Description:

If the PC sends „Everything On“ (see section PC to central unit), a broadcast is send to all devices which represents the real layout status. If e.g. an emergency stop is to be serviced and a device sends „Everything On“, the resulting broadcast will be „Everything Is Off“.

Remarks:

This is an unrequested information.

2.1.1.2 BC „Everything Is OFF“ (Emergency Stop)

Format:

	Headerbyte	Data 1	Xor-byte
Binary:	0110 0001	0000 0000	0110 0001
Hex :	0x61	0x00	0x61
Dez :	97	0	97

Description:

This is the information that the track voltage is switched off and therefor no turnout or locomotive command can be send.

Remarks:

This is an unrequested information.

2.1.1.3 BC „Stopped All Locomotives“

Format:

	Headerbyte	Data 1	Xor-byte
Binary:	1000 0001	0000 0000	0110 0001
Hex :	0x81	0x00	0x81
Dez :	129	0	129

Description:

This is the information that the central unit sends the „Stop Locomotive“ command on the track to stop all locomotives. The track voltage is still present so that turnouts can be switched.

Remarks:

This is an unrequested information.

2.1.1.4 BC „Programming Mode“

Format:

	Headerbyte	Data 1	Xor-byte
Binary:	0110 0001	0000 0010	0110 0011
Hex :	0x61	0x02	0x63
Dez :	97	2	99

Description:

This information is send if the central unit enters programming mode. No XpressNet device will be adressd but only the one which initiates the programming mode by sending e.g. a programming read command. This means that if the PC sends a programming command, it can continue and if another XpressNet device sends it, no command can be send to LI100F.

Programming mode can only be resumed if the device which initiates it, sends „Everything On“.

Remarks:

This is an unrequested information.

2.1.1.5 BC „Feedback Information“

Format:

	Headerbyte	Data 1	Data 2	Data 3	Data 4	etc.	Xor-byte
Binary:	0100 NNNN	ADR_1	DAT_1	ADR_2	DAT2	etc.	Xor-byte
Hex :	0x40 + N						Xor-byte
Dez :	64 + N						Xor-byte

Description:

This is the information to all slaves that one or more feedback positions had changed. This broadcast is send only if changes occured. In the broadcast, one up to 7 changed adresses can be transmitted (one databyte per adress, up to 15 bytes without header and xor.)
Until now, this feedback information is used only in the format of adressbyte, databyte. The contents of adressbyte ADR_x and databyte DAT_x is described in chapter „Switching informations“.

Remarks:

This is an unrequested information.

2.1.2 Programming Informations

After sending a programming read command, the central unit is entering programming mode and performs the desired action. The result can be fetched by sending the request for result command. Then one of the following answers can occur.

2.1.2.1 Programming Information „Short Circuit“

Format:

	Headerbyte	Data 1	Xor-byte
Binary:	0110 0001	0001 0010	Xor-byte
Hex :	0x61	0x12	Xor-byte
Dez :	97	18	Xor-byte

Description:

During read out or write of a decoder the central unit has detected a current which was too high. The assumption is that the desired action has failed.

Remarks:

None.

2.1.2.2 Programming Information „No Data“

Format:

	Headerbyte	Data 1	Xor-byte
Binary:	0110 0001	0001 0011	Xor-byte
Hex :	0x61	0x13	Xor-byte
Dez :	97	19	Xor-byte

Description:

There is no decoder connected on the programming pins of the central unit or the decoder did not answer to the programming commands of the central unit.

Remarks:

None.

2.1.2.3 Programming Information „Central Unit Busy“

Format:

	Headerbyte	Data 1	Xor-byte
Binary:	0110 0001	0001 1111	Xor-byte
Hex :	0x61	0x1f	Xor-byte
Dez :	97	31	Xor-byte

Description:

This command is not used up to central unit version 3.x.

Remarks:

None.

2.1.2.4 Programming Information „Central Unit Ready“

Format:

	Headerbyte	Data 1	Xor-byte
Binary:	0110 0001	0001 0001	Xor-byte
Hex :	0x61	0x11	Xor-byte
Dez :	97	17	Xor-byte

Description:

This command is not used up to central unit version 3.x.

Remarks:

None.

2.1.2.5 Programming Information „Data In 3-Byte-Format“

Format:

	Headerbyte	Data 1	Data 2	Data 3	Xor-byte
Binary:	0110 0011	0001 0000	EEEE EEEE	DDDD DDDD	Xor-byte
Hex :	0x63	0x10	E	D	Xor-byte
Dez :	99	16	E	D	Xor-byte

Description:

This information is send only to that slave which has put the central unit into programming mode. Content is the Erprom adress (E) and the data (D) in this adress.

Remarks:

This information belongs only to programming actions in register- or pagemode. If a decoder was requested in cv-mode and this information is the result, the decoder cannot handle cv-mode and must be further programmed in register- or pagemode.

2.1.2.6 Programming Information „Data In 4-Byte-Format“

Format:

	Headerbyte	Data 1	Data 2	Data 3	Xor-byte
Binary:	0110 0011	0001 0100	CCCC CCCC	DDDD DDDD	Xor-byte
Hex :	0x63	0x14	C	D	Xor-byte
Dez :	99	20	C	D	Xor-byte

Description:

This information is send only to that slave which has put the central unit into programming mode. Content is the cv-adress (C) and the data (D) in this adress. It is send only if the decoder can handle cv programming. Possible cv´s on XpressNet are 1 to 255. CV256 is send as 0.

Remarks:

If a decoder was asked with cv read command and you receive this information, everything is ok. If the decoder cannot handle cv programming, the programming info can come in register- or pagemode. See 2.1.2.5.

2.1.3 Softwareversion Central Unit

Including central unit version 2.3, the answer for version command comes in one byte. Since central units version 3.0 and later, there is an additional byte containing the central unit ID. With this, a slave can decide which locomotive commands the central unit accepts, if multi units are possible etc. The request for version information is the same in both cases.

2.1.3.1 Softwareversion Central Unit up to LZ-version 2.3

Format:

	Headerbyte	Data 1	Data 2	Xor-byte
Binary:	0110 0010	0010 0001	OOOO UUUU	Xor-byte
Hex :	0x62	0x21	O + U	Xor-byte
Dez :	98	33	O + U	Xor-byte

Description:

The version number is divided into upper (O) and lower nibble (U) and bcd coded.
Example: Data 2 = 0010 0011 = 0x23 : Version 2.3

Remarks:

None.

2.1.3.2 Softwareversion Central Unit version 3.0 onward

Format:

	Headerbyte	Data 1	Data 2	Data 3	Xor-byte
Binary:	0110 0011	0010 0001	OOOO UUUU	IIII IIII	Xor-byte
Hex :	0x63	0x21	O + U	ID	Xor-byte
Dez :	99	33	O + U	ID	Xor-byte

Description:

The version number is divided into upper (O) and lower nibble (U) and bcd coded.
Example: Data 2 = 0011 0000 = 0x30 : Version 3.0.
Additionally the central unit ID is send where:
ID = 0x00: LZ 100 – central unit
ID = 0x01: LH 200 – central unit
ID = 0x02: DPC – central unit (Compact and Commander)

Remarks:

None.

2.1.4 Status Central Unit

Format:

	Headerbyte	Data 1	Data 2	Xor-byte
Binary:	0110 0010	0010 0010	SSSS SSSS	Xor-byte
Hex :	0x62	0x22	S	Xor-byte
Dez :	98	34	S	Xor-byte

Description:

The request for status brings this information. The status byte is coded in bits:

Bit 0: if 1, all locomotives are stopped
Bit 1: if 1, layout in emergency stop
Bit 2: central unit startmode (0 = manual start, 1 = automatic start)
Auto-start : All locomotives start with their last settings
Manueller Start : All locomotives set to speed 0 and functions off.
Bit 3: if 1, programming mode active
Bit 4: reserved
Bit 5: reserved
Bit 6: if 1, central unit has performed a cold start
Bit 7: if 1, RAM-check-error in central unit

Remarks:

Not all bits are used in all central units. If bits 2 and 6 are set, the central unit does not place data on the track but waits for one device selecting manual mode or automatic mode. If one device sends the startmode command, the central unit starts with track output and sends the broadcast „Everything On“. Not all central units support different startmodes. If a device detects coldstart and startmode auto, it should continue in it´s program only if it sends a startmode command or if it receives the broadcast „Everything On“. In the latter case, another XpressNet device has sent the startmode command. So good practice after a device reset is to check first for central unit status and then request locomotive infos etc.

2.1.5 Central Unit Busy

Format:

	Headerbyte	Data 1	Xor-byte
Binary:	0110 0001	1000 0001	Xor-byte
Hex :	0x61	0x81	Xor-byte
Dez :	97	129	Xor-byte

Description:

If a command cannot be serviced by the central unit, this busy message occurs. In most cases it means that a given command could not be placed on the track.

Remarks:

One situation for this information to come up is if a device (e.g. the PC) tries to switch many turnouts as fast as possible regarding CTS signal. This produces an amount of data which is higher than can be placed on track output.

2.1.6 Command Unknown In Central Unit

Format:

	Headerbyte	Data 1	Xor-byte
Binary:	0110 0001	1000 0010	Xor-byte
Hex :	0x61	0x82	Xor-byte
Dez :	97	130	Xor-byte

Description:

If a command is transferred correct but not implemented in the central unit, it responds with this information. „Command unknown“ also is send if a command cannot be performed in the current situation (e.g. request programming result without being in programming mode).

Remarks:

None.

2.1.7 Switching Informations

Format:

	Headerbyte	Data 1	Data 2	Xor-byte
Binary:	0100 0010	AAAA AAAA	ITTN ZZZZ	Xor-byte
Hex :	0x42	ADR	ITNZ	Xor-byte
Dez :	66	ADR	ITNZ	Xor-byte

Description:

As response on a request for switching informations, this information is send. The data contained can be information about the status of accessory decoders with feedback or without or the status of feedback modules. The coding in detail is:

Data 1: AAAA AAAA : For a turnout, data 1 is the turnout adress divided by 4. For central units less than V3.0, data 1 has a range of 0 to 63 = 6 bit. For central units V3.0 onward all 8 bits are used for this group adress so the range of turnouts is from 0..1023 = 256 * 4. Is for example ADR =0x00, the information is about turnouts 0, 1, 2 or 3 which means turnout group 0 (if a switching decoder is marked in the coding bits TT)
For a feedback module, the adress can be in the range of 0..127 (7 Bit) and is directly the adress of the module.

Data 2: I If this bit is 1, the given switching command is still not completed and the turnout has not reached end position.
Undefined for feedback modules because the inputs of these modules can become only 0 or 1.

Data 2: TT This is the coding of the type of requested adress:
TT = 0 0 : Adress is switching module without feedback
TT = 0 1 : Adress is switching module wih feedback

TT = 1 0 : Adress is a feedback module

TT = 1 1 : reserved for future use

Data 2: N

This bit describes in which nibble a turnout or feedback module adress resides. N=0 is the lower nibble, N=1 the upper nibble. For e.g. turnout group 0, the lower nibble contains the status of turnouts 0 and 1 in the status flags Z and the higher nibble the status of 3 and 4. For a feedback module the lower nibble contains status for the 4 lower feedback inputs, the upper nibble contains status of the 4 higher feedback inputs. To access all 8 inputs of a feedback module you need to request information twice, one for the lower nibble, one for the upper.

Please note, that the nibble bit is only correct, if a turnout was switched since central unit cold start.

Data 3: Z3 Z2 Z1 Z0

For the status flags is defined in the case of a turnout module:

Z1 and Z0 represent the status of the first turnout in the nibble, Z3 and Z2 the one of the second turnout.

Possible combinations:

Z1	Z0	(the first tunrout in the nibble)
0	0	turnout not switched since central unit start
0	1	last switching command was „0“, turnout „left“. This is only relativ.
1	0	last switching command was „1“
1	1	invalid combination if both endswitches of a turnout with feedback are pressed.

The same structure is used for Z3 and Z2, the second turnout in the nibble. For a feedback module, the 4 bits Z3..Z0 directly represent the value of the 4 inputs of the requested nibble.

Remarks:

None.

2.1.8 Locomotive Information up to LZ-version 1.5

This locomotive information is the result of the request described in 2.2.18.1. Up to version 1.5 of LZ100, only 14 speedsteps were used (no „ModSel“-byte). The locomotive info „Loco occupied“ is send unrequested to the device which currently had controlled the locomotive. So here maybe a routine should be used to inform the user that this locomotive was taken over from another device.

2.1.8.1 Locomotive Free up to LZ-version 1.5

If the locomotive is not used from another XpressNet device, this is the answer for request of locomotive info with the header meaning „Loco free“.

Format:

	Headerbyte	Data 1	Data 2	Data 3	Xor-byte
Binary:	1000 0011	Locadress	Locdata 1	Locdata 2	Xor-byte
Hex :	0x83	Locadress	Locdata 1	Locdata 2	Xor-byte
Dez :	131	Locadress	Locdata 1	Locdata 2	Xor-byte

Description:

Locadress: range is 0..99 dez. = 0..0x63. The adress 0 is used for conventional locomotive

Locdata 1: Bit 7: B7=1: planned: loco is occupied
B7=0: planned: Lok is free
Bit 7 is not used, so that the headerbyte gives the information if the loco is free.
Bit 6: B6=1: direction forward
B6=0: direction backward
Bit 5: B5=1: function 0 is on
B5=0: function 0 is off
Bit 4: not used, always 0
Bit3 to Bit0 represent the speed as follows:

Bit3	Bit2	Bit1	Bit0	
0	0	0	0	speed 0
0	0	0	1	Locomotive stop without decceleration.
0	0	1	0	speed1
.	.	.	.	.
1	1	1	1	speed 14

Locdata 2: Bit7 to Bit4 : not used
Bit3: status of function 4, „0“ = off, „1“ = on
Bit2: status of function 3
Bit1: status of function 2
Bit0: status of function 1

Remarks:

None.

2.1.8.2 Locomotive Occupied up to LZ-version 1.5

If the requested locomotive is used by another device, this information with the header „occupied“ is returned. This information is also send without request if another device takes over control over the locomotive.

Format:

	Headerbyte	Data 1	Data 2	Data 3	Xor-byte
Binary:	1010 0011	Locadress	Locdata 1	Locdata 2	Xor-byte
Hex :	0xA3	Locadress	Locdata 1	Locdata 2	Xor-byte
Dez :	163	Locadress	Locdata 1	Locdata 2	Xor-byte

Description:

Locadress, Locdata 1 and Locdata 2 are formatted as 2.1.8.1 describes.

Remarks:

This information can be received unrequested.

2.1.9 Locomotive Information up to LZ-version 2.3

This is the answer of a request for locomotive data as described in 2.2.18.2. From version 2.0 onward, the central unit uses beside 14 speedsteps also 27 and 28 speedsteps. Therefor the request for locomotive information was extended with a „ModSel“-byte (mode select). This byte contains only the information for maximum speedsteps. The „occupied“ information is send unrequested to the device which currently had controlled the locomotive.

2.1.9.1 Locomotive Free up to LZ-version 2.3

If the locomotive is not used from another XpressNet device, this is the answer for request of locomotive info with the header meaning „Loco free“.

Format:

	Headerbyte	Data 1	Data 2	Data 3	Data 4	Xor-byte
Binary:	1000 0100	Locadress	Locdata 1	Locdata 2	ModSel	Xor-byte
Hex :	0x84	Locadress	Locdata 1	Locdata 2	ModSel	Xor-byte
Dez :	132	Locadress	Locdata 1	Locdata 2	ModSel	Xor-byte

Description:

Locadress: range is 0..99 dez. = 0..0x63. The adress 0 is the conventional locomotive

ModSel : Contains the speedstep information for the requested locomotive. Depending on the content of ModSel, the meaning of the bytes Locdata 1 and Locdata 2 differs.

Bit7 .. Bit2: unused.

Bit1	Bit0	
0	0	14 speedsteps
0	1	27 speedsteps
1	0	28 speedsteps
1	1	reserved

ModSel = xxxxxx00 14-speedstep mode

Locdata 1 and Locdata 2 are coded as 2.1.8.1 describes.

ModSel = xxxxxx01 27 speedstep mode

Locdata 1: Bit 7: B7=1: planned: locomotive occupied

B7=0: planned: locomotive is free.

Bit 7 is not used, so that the headerbyte gives the information if the loco is free.

Bit 6: B6=1: direction vorward

B6=0: direction backward

Bit 5: B5=1: function 0 is on

B5=0: function 0 is off

Bit4 to Bit0 code the speedstep as follows: Please note that bit 4 is the LSB of the speed.

Bit3	Bit2	Bit1	Bit0	Bit4 (!)	
0	0	0	0	0	speed 0
0	0	0	0	1	not used
0	0	0	1	0	Loco stop without decceleration
0	0	0	1	1	not used
0	0	1	0	0	speed 1
0	0	1	0	1	speed 2
0	0	1	1	0	speed 3
.	.	.	.	.	
1	1	1	1	0	speed 27

Locdata 2: coded as 2.1.8.1 describes.

ModSel = xxxxxx10 28 speedstep mode

Locdata 1: Bit 7: B7=1: planned: locomotive occupied

B7=0: planned: locomotive is free.

Bit 7 is not used, so that the headerbyte gives the information if the loco is free.

Bit 6: B6=1: direction vorward

B6=0: direction backward

Bit 5: B5=1: function 0 is on

B5=0: function 0 is off

Bit4 to Bit0 code the speedstep as follows: Please note that bit 4 is the LSB of the speed.

Bit3	Bit2	Bit1	Bit0	Bit4 (!)	
0	0	0	0	0	speed 0
0	0	0	0	1	not used
0	0	0	1	0	Loco stop without decceleration
0	0	0	1	1	not used
0	0	1	0	0	speed 1
0	0	1	0	1	speed 2
0	0	1	1	0	speed 3
.	.	.	.	.	
1	1	1	1	0	speed 27
1	1	1	1	1	speed 28

Locdata 2: coded as 2.1.8.1 described.

Remarks:

None.

2.1.9.2 Locomotive Occupied up to LZ-version 2.3

If the requested locomotive is used by another device, this information with the header „occupied“ is returned. This information is also send without request if another device takes over control over the locomotive.

Format:

	Headerbyte	Data 1	Data 2	Data 3	Data 4	Xor-byte
Binary:	1010 0100	Locadress	Locdata 1	Locdata 2	ModSel	Xor-byte
Hex :	0xA4	Locadress	Locdata 1	Locdata 2	ModSel	Xor-byte
Dez :	164	Locadress	Locdata 1	Locdata 2	ModSel	Xor-byte

Description:

Locadress, Locdata 1 and Locdata 2 are coded as 2.1.9.1 describes.

Remarks:

This information can be received unrequested.

2.1.10 Locomotive Information central unit version 3.0 onward

As response on a request for locomotive data in the format for version 3.0 or above (see 2.2.18.3), one of the following answers can be received. Unlike in previous central unit versions, the „occupied“-information is contained here. This means that these answers do not come unrequested. If a locomotive is taken over from another device, from now on the message therefor is send as unrequested information „locomotive occupied“ (see 2.1.2).
Also an additional data byte as marker byte is inserted after the headerbyte. This marker byte is used to distinguish between the various new commands in version 3. In the answers described

here, the locomotive adress is not included to avoid sending unneccessary data. But the structure of XpressNet allows the clear assignement of the answer because it is given directly after requesting it.

2.1.10.1 Locomotive Information „Normal Locomotive“

This information is send if the requested locomotive is not member of a double header, multi unit or base adress of a multi unit.

Format:

	Headerbyte	Marker	Data 1	Data 2	Data 3	Xor-byte
Binary:	1110 0100	0000 BFFF	RVVV VVVV	000F FFFF	FFFF FFFF	Xor-byte
Hex :	0xE4	Marker	Speed	F0	F1	Xor-byte
Dez :	228	Marker	Speed	F0	F1	Xor-byte

Description:

Marker: Bit3: B=0: Locomotive is free
B=1: Locomotive is used by another device (occupied)
Bit2 to Bit0: Coding of the speedsteps

Bit2	Bit1	Bit0	
0	0	0	14 speed steps
0	0	1	27 speed steps
0	1	0	28 speed steps
1	0	0	128 speed steps

Speed: Coding of speed and direction. R=1: forward, R=0: backward.
14 speedsteps: Coding of bits 3,2,1,0 for speed as described in 2.1.8.1
27 speedsteps: Coding of bits 4,3,2,1,0 for speed as described in 2.1.9.1, ModSel = xxxxxx01.
28 speedsteps: Coding of bits 4,3,2,1,0 for speed as described in 2.1.9.1, ModSel = xxxxxx10.
128 speedsteps:

Bit6	B5	B4	B3	B2	B1	B0	
0	0	0	0	0	0	0	speed 0
0	0	0	0	0	0	1	Loco stop
0	0	0	0	0	1	0	speed 1
0	0	0	0	0	1	1	speed 2
.	.	.	.	.	.	.	.
1	1	1	1	1	1	1	speed 126

F0: Status of functions 0 to 4. 0 0 0 F0 F4 F3 F2 F1
F1: Status of functions 5 to 12 F12 F11 F10 F9 F8 F7 F6 F5
Where a 1 means function is on.

Remarks:

None.

2.1.10.2 Locomotive Information „Locomotive Is Member Of A Multi Unit“

Format:

	Headerbyte	Marker	Data 1	Data 2	Data 3	Data 4	Xor-byte
Binary:	1110 0101	0001 BFFF	RVVV VVVV	000F FFFF	FFFF FFFF	MU	Xor-byte
Hex :	0xE5	Marker	Speed	F0	F1	MU	Xor-byte
Dez :	229	Marker	Speed	F0	F1	MU	Xor-byte

Description:

Marker: Bits 3 to 0 are coded as described in 2.1.10.1. The speedstep is the speedstep of the requested locomotive.This can be different to the speedstep of the multi unit.

Speed: The speed byte is coded as described in 2.1.10.1. The speed is the speed of the requested locomotive.

F0, F1: As described in 2.1.14.1.

MU: This is the multi unit base adress of the requested locomotive.

Remarks:

Speed commands must be send to the mu baseadress, because not all central units do this. Function commands must be send to the locomotive adress itself.

2.1.10.3 Locomotive Information „Locoadress Is Baseadress Of A Multi Unit“

Format:

	Headerbyte	Marker	Data 1	Xor-byte
Binary:	1110 0010	0010 BFFF	RVVV VVVV	Xor-byte
Hex :	0xE2	Marker	Speed	Xor-byte
Dez :	226	Marker	Speed	Xor-byte

Description:

Marker: Bits 3 to 0 are described in 2.1.10.1. The speedstep is the speedstep of the multi unit.

Speed: The speed byte is coded as described in 2.1.10.1. The speed is the speed of the multi unit.

Remarks:

No function commands should be send to a mu base adress.

2.1.10.4 Locomotive Information „Locomotive Is Member Of A Double Header“

Format:

	Headerbyte	Marker	Data 1	Data 2	Data 3	Data 4	Data 5	Xor-byte
Binary:	1110 0110	0110 BFFF	RVVV VVVV	000F FFFF	FFFF FFFF	Adr High	Adr Low	Xor-byte
Hex :	0xE6	Marker	Speed	F0	F1	AH	AL	Xor-byte
Dez :	230	Marker	Speed	F0	F1	AH	AL	Xor-byte

Description:

Marker: Bits 3 to 0 are described in 2.1.10.1.
Speed: The speed byte is described in 2.1.10.1.
F0, F1: The function bytes are described in 2.1.10.1.
AH: Highbyte of the second adress in the double header.
AL: Lowbyte of the second adress of the double header.
For locomotive adresss < 100 :
Highbyte of the adress is 0x00
Lowbyte of the adress is 0x00 to 0x63
For locomotive adresss 100 to 9999 :
Highbyte of the adress is: AH = (ADR&0xFF00)+0xC000
Lowbyte of the adress is: AL = (ADR&0x00FF)

Remarks:

This answer is send only if the locomotive request was using the „new“ command (see 2.2.21).

2.1.11 Locomotive Occupied LZ-version 3.0 onward

Format:

	Headerbyte	Marker	Data 1	Data 2	Xor-byte
Binary:	1110 0011	0100 0000	Adr High	Adr Low	Xor-byte
Hex :	0xE3	0x40	AH	AL	Xor-byte
Dez :	227	64	AH	AL	Xor-byte

Description:

This information is send to the device which had controlled the locomotive to flag that another device is using it now.
For locomotive adresss < 100 :
Highbyte of the adress is 0x00
Lowbyte of the adress is 0x00 to 0x63
For locomotive adresss 100 to 9999 :
Highbyte of the adress is: AH = (ADR&0xFF00)+0xC000
Lowbyte of the adress is: AL = (ADR&0x00FF)

Remarks:

This information is always unrequested.

2.1.12 Function Type central unit version 3.0 onward

From version 3.0 on, the LZ100 stores this additional information, if a function should be „continuous“ or „momentary“. The track output is not changed. This is only an information for a device to adapt it´s user interface (e.g. sound effect is momentary). This command is used in the central unit to allow the usage of a momentary function even if another device takes control over a locomotive. The corresponding locomotive adress is not send but is known due to the request for this information.

Format:

	Headerbyte	Marker	Data 1	Data 2	Xor-byte
Binary:	1110 0011	0101 0000	000S SSSS	SSSS SSSS	Xor-byte
Hex :	0xE3	0x50	S0	S1	Xor-byte
Dez :	227	80	S0	S1	Xor-byte

Description:

S0: Type of function 0 to 4.
S0 = 0 0 0 S0 S4 S3 S2 S1
S1: Type of function 5 to 12.
S1 = S12 S11 S10 S9 S8 S7 S6 S5
Sx=1 means function is momentary.

Remarks:

None.

2.1.13 Adress Search Results central unit version 3.0 onward

Format:

	Headerbyte	Marker	Data 1	Data 2	Xor-byte
Binary:	1110 0011	0011 KKKK	Adr High	Adr Low	Xor-byte
Hex :	0xE3	0x30 + K	AH	AL	Xor-byte
Dez :	227	48 + K	AH	AL	Xor-byte

Description:

This information is send as response to an adress search request as described in 2.2.24. It allows a device to show a selection list for comfortable adress selection.
The marker byte contains the type of adress which is placed in Adr High / Adr Low.

Marker: KKKK = 0: Normal locomotive in Data 1/2
KKKK = 1: Double header lcomotive in Data 1/2
KKKK = 2: Multi unit base adress in Data 1/2
KKKK = 3: Member of a multi unit in Data 1/2
KKKK = 4: no adress found. Data 1/2 = 0x00

AH/AL: Locomotive adress is calculated as described in 2.1.11.

Remarks:

None.

2.1.14 Double Header Information up to LZ-version 1.5

This is a response due to the request for locomotive information described in 2.2.18.1. Up to version 1.5 of the central unit, only 14 speedsteps were used (no „ModSel“-byte). The information „double header occupied“ is send unrequested to the device which had controlled this adress.

2.1.14.1 Double Header Free up to LZ-version 1.5

If the locomotive is not used by another device, this is the response for request of locomotive info with the headerbyte „locomotive free“.

Format:

	Headerbyte	Data 1	Data 2	Data 3	Data 4	Data 5	Xor-byte
Binary:	1100 0101	0000 0100	Locadress 1	Locdata 1	Locdata 2	Locadress 2	Xor-byte
Hex :	0xC5	0x04	Locadress 1	Locdata 1	Locdata 2	Locadress 2	Xor-byte
Dez :	197	4	Locadress 1	Locadress 2	Locdata 2	Locadress 2	Xor-byte

Description:

Locadress 1, Locdata 1, Locdata 2 and Locadress 2 are described in 2.1.8.1.

Remarks:

For central units vesion 3 or above, only commands with „ModSel“ are supported. Double header informations in this format are only send to device which request information in the old format. A request in the new format to a locomotive which is in an „old“ double header results in locomotive information „normal locomotive“ and can be used with the new speed commands.

2.1.14.2 Double Header Occupied up to LZ-version 1.5

Format:

	Headerbyte	Data 1	Data 2	Data 3	Data 4	Data 5	Xor-byte
Binary:	1100 0101	0000 0101	Locadress 1	Locdata 1	Locdata 2	Locadress 2	Xor-byte
Hex :	0xC5	0x05	Locadress 1	Locdata 1	Locdata 2	Locadress 2	Xor-byte
Dez :	197	5	Locadress 1	Locdata 1	Locdata 2	Locadress 2	Xor-byte

Description:

Locadress 1, Locdata 1, Locdata 2 and Locadress 2 are described in 2.1.8.1.

Remarks:

This information can be received unrequested.

2.1.15 Double Header Information up to LZ-version 2.3

This is the response due to the request described in 2.2.18.2. Because of the possibility of 14,27 and 28 speedsteps, the „ModSel“-byte is added. The information „double header occupied“ is send unrequested to the device which had controlled this adress.

2.1.15.1 Double Header Free up to LZ-version 2.3

If the locomotive is not used by another device, this is the response for request of locomotive info with the headerbyte „locomotive free“.

Format:

	Headerbyte	Data 1	Data 2	Data 3	Data 4	Data 5	Data 5	Xor-byte
Binary:	1100 0110	0000 0100	Locadr. 1	Locdat 1	Locdat 2	Locadr. 2	Modsel	Xor-byte
Hex :	0xC6	0x04	Locadr. 1	Locdat 1	Locdat 2	Locadr. 2	ModSel	Xor-byte
Dez :	198	4	Locadr. 1	Locdat 1	Locdat 2	Locadr 2	ModSel	Xor-byte

Description:

Locadress 1, Locdata 1, Locdata 2, Locadress 2 and ModSel are described in 2.1.9.1.

Remarks:

Double header informations in this format are only send to devices which request information in the old format with the „ModSel“ byte. A request in the new format to a locomotive which is in an „old“ double header results in lcomotive information „normal locomotive“ and can be used then with the new speed commands.

2.1.15.2 Double Header Occupied up to LZ-version 2.3

Format:

	Headerbyte	Data 1	Data 2	Data 3	Data 4	Data 5	Data 5	Xor-byte
Binary:	1100 0110	0000 0101	Locadr. 1	Locdat 1	Locdat 2	Locadr. 2	Modsel	Xor-byte
Hex :	0xC6	0x05	Locadr. 1	Locdat 1	Locdat 2	Locadr. 2	ModSel	Xor-byte
Dez :	198	5	Locadr. 1	Locdat 1	Locdat 2	Locadr. 2	ModSel	Xor-byte

Description:

Locadress 1, Locdata 1, Locdata 2, Locadress 2 and ModSel are described in 2.1.9.1.

Remarks:

This information can be received unrequested.

2.1.16 Double Header Error Messages up to LZ-version 2.3

A double header can be assembled or disassembled only if certain requirements are fulfilled. This can lead to errors which are described in the following

Format:

	Headerbyte	Data 1	Xor-byte
Binary:	0110 0001	1000 0FFF	Xor-byte
Hex :	0x61	0x80 + F	Xor-byte
Dez :	97	128 + F	Xor-byte

Description:

The coding of the errorbits is as follows:

FFF = 011: One of the locomotives is not controlled by the assembling XpressNet device or locomotive 0 is selected.

FFF = 100: One of the locomotives of the double header ist used by another XpressNet device.

FFF = 101: One locomotive is already in a double header

FFF = 110: The speed of one locomotive is not 0.

Remarks:

Normally, the command to assemble or disassemble a double header has no answer from the central unit if the action was succesful. The LI100F acknowledges this with request for next command. In the case of failure, the central unit sends one of the error messages.

2.1.17 Error Messages central unit version 3.0 onward

For central unit version 3 and above, certain errors are summarized in this error message. The exact meaning of the error depends on the action before which results in this message.

Format:

	Headerbyte	Marker	Xor-byte
Binary:	1110 0001	1000 FFFF	Xor-byte
Hex :	0xE1	0x80 + F	Xor-byte
Dez :	225	128 + F	Xor-byte

Description:

The 4 errorbits are coded as follows:

FFFF = 0001: One locomotive is not controlled by the assembling XpressNet device or locomotive 0 is selected

FFFF = 0010: One locomotive is used by another XpressNet device.

FFFF = 0011: One locomotive is already in a mu or double header.

FFFF = 0100: The speed of a locomotive is not 0.

FFFF = 0101: The locomotive is not member of a mu.

FFFF = 0110 The locomotive is not base adress of a mu.

FFFF = 0111: Deleting a locomotive is not possible.

FFFF = 1000: Central unit stack is full.

Remarks:

None.

2.2 PC To Central Unit

After the LI100F has set the CTS signal to enable PC transmission, the PC can begin a communication and e.g. request data, control locomotives etc. Depending of the desired action, the PC gets a result from the central unit or from LI100F. It is important that the PC program uses the hardware handshake of the serial port because only in that way communication becomes stable for this system.

2.2.1 Everything On

Format:

	Headerbyte	Data 1	Xor-byte
Binary:	0010 0001	1000 0001	1010 0000
Hex :	0x21	0x81	0xA0
Dez :	33	129	160

Description:

This command prompts the central unit to switch the track voltage on if it was off and to restart sending track commands. With this an emergency stop, stop all locomotives or programming mode is terminated. If this command is successful, the central unit broadcasts „Everything Is On“. See also 2.1.4.1.

Remarks:

None.

2.2.2 Everything OFF (Emergency Stop)

Format:

	Headerbyte	Data 1	Xor-byte
Binary:	0010 0001	1000 0000	1010 0001
Hex :	0x21	0x80	0xA1
Dez :	33	128	161

Description:

This command prompts the central unit to switch off the track voltage. After this, certain broadcasts „Everything Is Off“ are send to all XpressNet devices and also to the one which initiates the emergency stop.

Remarks:

None.

2.2.3 Stop All Locomotives

Format:

	Headerbyte	Xor-byte
Binary:	1000 0000	1000 0000
Hex :	0x80	0x80
Dez :	128	128

Description:

This command prompts the central unit to stop all locomotives without their programmed decceleration. This is done by sending a stop command on the track and no further speed comands. The track voltage remains on, so turnouts can be switched. A broadcast is send.

Remarks:

None.

2.2.4 Stop One Locomovtive

2.2.4.1 Stop One Locomotive up to LZ-version 2.3

Format:

	Headerbyte	Data 1	Xor-byte
Binary:	1001 0001	Locadress	Xor-byte
Hex :	0x91	Locadress	Xor-byte
Dez :	145	Locadress	Xor-byte

Description:

The central unit stops the desired locomotive wihout its programmed decceleration. Track voltage remains on and all other locomotives continue to run.
No broadcast is send.

Remarks:

The locomotive adress is in the range of 0 to 99.

2.2.4.2 Stop One Locomotive central unit version 3.0 onward

Format:

	Headerbyte	Data 1	Data 2	Xor-byte
Binary:	1001 0010	Adr High	Adr Low	Xor-byte
Hex :	0x92	AH	AL	Xor-byte
Dez :	146	AH	AL	Xor-byte

Description:

The central unit stops the desired locomotive wihout its programmed decceleration. Track voltage remains on and all other locomotives continue to run.
No broadcast is send.

Remarks:

The locomotives 0 to 9999 can be stopped.
AH/AL: Locomotive adress is calculated as described in 2.1.11.

2.2.5 Stop Locomotives up to LZ-version 2.3

Format:

	Headerbyte	Data 1	...	Daten N	Xor-byte
Binary:	1001 NNNN	Locadress 1	...	Locadress N	Xor-byte
Hex :	0x90 + N	Locadress 1	...	Locadress N	Xor-byte
Dez :	144 + N	Locadress 1	...	Locadress N	Xor-byte

Description:

The central unit stops the desired locomotives wihout their programmed deccelerations. Track voltage remains on and all other locomotives continue to run.
No broadcast is send.

Remarks:

The locomotive adress is in the range of 0 to 99.
This command is not supported for version 3 and above and should be replaced by a series of single stop commands.

2.2.6 Read Command Programming-Format 3-Byte (Registermode)

Format:

	Headerbyte	Data 1	Data 2	Xor-byte
Binary:	0010 0010	0001 0001	0000 RRRR	Xor-byte
Hex :	0x22	0x11	R	Xor-byte
Dez :	34	17	R	Xor-byte

Description:

This command prompts the central unit to switch to programming mode and try to read the decoder on the programming track in registermode. Central unit tries to read the value in register 0000 RRRR. Allowed are the values 1..8.

Remarks:

This read command has no response of the central unit. The answer must be explicitly requested with „request programming read result“. Only then can be decided if the read command was successful and the result has the desired format.
After sending the read command, the central unit responds with broadcast „programming mode“ and further adresss only the device which has send the command until this device sends „Everything On“ to terminate the programming mode.

2.2.7 Read Command Programming Format 4-Byte (CV-mode)

Format:

	Headerbyte	Data 1	Data 2	Xor-byte
Binary:	0010 0010	0001 0101	CCCC CCCC	Xor-byte
Hex :	0x22	0x15	CV	Xor-byte
Dez :	34	21	CV	Xor-byte

Description:

This command prompts the central unit to switch to programming mode and try to read the decoder on the programming track in cv-mode. Central unit tries to read the value in the CV selected in data 2. Allowed are the values 1..256, where CV256 has to be send as 0.

Remarks:

This read command has no response of the central unit. The answer must be explicitly requested with „request programming read result“. Only then can be decided if the read command was successful and the result has the desired format (cv-mode). If the decoder could not be read in cv-mode, the central unit tries to read in registermode. If this is successful, the information which can be requested is in the format for registermode.
After sending the read command, the central unit responds with broadcast „programming mode“ and further adresss only the device which has send the command until this device sends „Everything On“ to terminate the programming mode.

2.2.8 Read Command Programming Format 3-Byte (Pagemode)

Format:

	Headerbyte	Data 1	Data 2	Xor-byte
Binary:	0010 0010	0001 0100	CCCC CCCC	Xor-byte
Hex :	0x22	0x14	CV	Xor-byte
Dez :	34	20	CV	Xor-byte

Description:

This command prompts the central unit to switch to programming mode and try to read the decoder on the programming track in pagemode. Central units tries to read the value in the CV selected in data 2. Allowed are the values 1..256, where CV256 has to be send as 0.

Remarks:

This read command has no response of the central unit. The answer must be explicitly requested with „request programming read result“. Only then can be decided if the read command was successful and the result has the desired format (pagemode). After sending the read command, the central unit responds with broadcast „programming mode“ and further adresss only the device which has send the command until this device sends „Everything On“ to terminate the programming mode.

2.2.9 Request Programming Read Result

Format:

	Headerbyte	Data 1	Xor-byte
Binary:	0010 0001	0001 0000	0011 0001
Hex :	0x21	0x10	0x31
Dez :	33	16	49

Description:

To request the result of a previously given read command, this command is used. Possible answers are described in 2.1.2.

Remarks:

None.

2.2.10 Write Command Programming Format 3-Byte (Registermode)

Format:

	Headerbyte	Data 1	Data 2	Data 3	Xor-byte
Binary:	0010 0011	0001 0010	0000 RRRR	Data	Xor-byte
Hex :	0x23	0x12	R	Data	Xor-byte
Dez :	35	18	R	Data	Xor-byte

Description:

This command prompts the central unit to switch to programming mode and writes to the decoder on the programming track in registermode. Central units tries to write the value in data 3 to the register specified in data 2. Allowed range for the register is 1..8.

Remarks:

Before using a write command, the central unit should be placed in programming mode with a read command. There is no possibility to enshure for a XpressNet device the success of the programming action except reading the value back.

2.2.11 Write Command Programming Format 4-Byte (CV-mode)

Format:

	Headerbyte	Data 1	Data 2	Data 3	Xor-byte
Binary:	0010 0011	0001 0110	CCCC CCCC	Data	Xor-byte
Hex :	0x23	0x16	CV	Data	Xor-byte
Dez :	35	22	CV	Data	Xor-byte

Description:

This command prompts the central unit to switch to programming mode and writes to the decoder on the programming track in cv-mode. Central unit tries to write the value in data 3 to the cv specified in data 2. Allowed range for the cv is 1..256, where CV256 has to be send as 0.

Remarks:

Before using a write command, the central unit should be placed in programming mode with a read command and then checked if cv-mode is possible. There is no possibility to enshure for a XpressNet device the success of the programming action except reading the value back.

2.2.12 Write Command Programming Format 3-Byte (Pagemode)

Format:

	Headerbyte	Data 1	Data 2	Data 3	Xor-byte
Binary:	0010 0011	0001 0111	CCCC CCCC	Data	Xor-byte
Hex :	0x23	0x17	CV	Data	Xor-byte
Dez :	35	23	CV	Data	Xor-byte

Description:

This command prompts the central unit to switch to programming mode and writes to the decoder on the programming track in pagemode. Central unit tries to write the value in data 3 to the cv specified in data 2. Allowed range for the cv is 1..256, where CV256 has to be send as 0.

Remarks:

Before using a write command, the central unit should be placed in programming mode with a read command. There is no possibility to enshure for a XpressNet device the success of the programming action except reading the value back.

2.2.13 Request Central Unit Software Version

Format:

	Headerbyte	Data 1	Xor-byte
Binary:	0010 0001	0010 0001	0000 0000
Hex :	0x21	0x21	0x00
Dez :	33	33	0

Description:

The central unit responds for this request with one of the answers described in 2.1.3.

Remarks:

None.

2.2.14 Request Central Unit Status

Format:

	Headerbyte	Data 1	Xor-byte
Binary:	0010 0001	0010 0100	0000 0101
Hex :	0x21	0x24	0x05
Dez :	33	36	5

Description:

The answer for status request is described in 2.1.4.

Remarks:

None.

2.2.15 Set Cental Unit Startmode

Format:

	Headerbyte	Data 1	Data 2	Xor-byte
Binary:	0010 0010	0010 0010	0000 0M00	Xor-byte
Hex :	0x22	0x22	M	Xor-byte
Dez :	34	34	M	Xor-byte

Description:

Sets the startmode for the central unit after reset. M=0: manual start of all locomotives, M=1: automatic start of all locomotives with their last settings of speed and function.

Remarks:

Not all central units support this command.

2.2.16 Request Switching Information

Format:

	Headerbyte	Data 1	Data 2	Xor-byte
Binary:	0100 0010	AAAA AAAA	1000 000N	Xor-byte
Hex :	0x42	Adress	0x80 + N	Xor-byte
Dez :	66	Adress	128 + N	Xor-byte

Description:

The response for this command is described in 2.1.7.

Adress: For a switching decoder, this is the adress of the desired output divided by 4 (group adress). The range of the group adress is 0..63 = 6bit for central unit versions < 3.0.
For version 3.0 and above, all 8 bits are allowed for the group adress. So 1024 turnouts can be switched and turnouts 0..512 can be with feedback, turnouts 513..1023 not.
For a feedback module, the adress can be 0..127 and is the module adress.

N: Marks the desired nibble. N=0 is the lower nibble, N=1 the upper nibble.
For switching decoders, there are 4 turnouts in one group and the lower nibble marks turnouts 0 and 1 in the group and the upper nibble turnouts 2 and 3.
For a feedback module, in the lower nibble the 4 lower inputs of the module are included and in the upper nibble, the upper 4 inputs.

Remarks:

Example 1: Status of turnout 21 is desired. Adress 21 mod 4 = 5. So turnout 21 is in group 5. In group 5, the turnouts 20,21,22 and 23 are placed. So the nibble bit is 0 (lower nibble).

Example 2: Status of turnout 620 is desired. Adress 620 mod 4 = 155. So turnout 620 is in group 155. In group 155, the turnouts 620, 621, 622 and 623 are placed. So the nibble bit is 1 (higher nibble).

2.2.17 Switching Command

Format:

	Headerbyte	Data 1	Data 2	Xor-byte
Binary:	0101 0010	AAAA AAAA	1000 DBBD	Xor-byte
Hex :	0x52	Adress	0x80 + DBBD	Xor-byte
Dez :	82	Adress	128 + DBBD	Xor-byte

Description:

Switching commands can only be send to swicthing decoders. The adress is turnout / 4 (group). The offset in the group and which of the 2 outputs has to be activated or deactivated has to be defined. This is done with the bits D1 B1 B0 and D2 in data 2.

B1 and B0: These are the two LSB´s which are the rest of the division by 4.

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

D2: D2 = 0 means use output 1 of the selected turnout.
D2 = 1 means use output 2 of the selected turnout.

Remarks:

For central unit version less than 3 the range for the group is 0 to 63 (=6bit). From version 3 and above up to 256 groups (0 to 255) are allowed. See also 2.2.16.

2.2.18 Request Locomotive Information

2.2.18.1 Request Locomotive Information up to LZ-version 1.5

Format:

	Headerbyte	Data 1	Xor-byte
Binary:	1010 0001	Locadress	Xor-byte
Hex :	0xA1	Locadress	Xor-byte
Dez :	161	Locadress	Xor-byte

Description:

Up to version 1.5 of the central unit only 14 speedsteps were used. The request for locomotive information with this command results in an answer with also 14 speedsteps (format for version 1.5) because central unit assumes that the requesting device cannot handle other information formats.

Locadress is in the range of 0 to 99. The answer is described in 2.1.8.

Remarks:

None.

2.2.18.2 Request Locomotive Information up to LZ-version 2.3

Format:

	Headerbyte	Data 1	Data 2	Xor-byte
Binary:	1010 0010	Locadress	ModSel	Xor-byte
Hex :	0xA2	Locadress	ModSel	Xor-byte
Dez :	162	Locadress	ModSel	Xor-byte

Description:

Here the additional mode select byte is used to distinguish between 14, 27 and 28 speedsteps. The central unit will answer also with the mode select byte so the requesting XpressNet device knows the speedsteps of the locomotive. Locomotive adress is in the range of 0 to 99. The asnwer is described in 2.1.9.

Remarks:

None.

2.2.18.3 Request Locomotive Information central unit version 3.0 onward

Format:

	Headerbyte	Marker	Data 1	Data 2	Xor-byte
Binary:	1110 0011	0000 0000	Adr High	Adr Low	Xor-byte
Hex :	0xE3	0x00	AH	AL	Xor-byte
Dez :	227	0	AH	AL	Xor-byte

Description:

Information can be requested for locomotives 0 to 9999.
AH/AL: Locomotive adress is calculated as described in 2.1.11.
Possible answers are describend in 2.1.10.

Remarks:

None.

2.2.18.4 Request Type Of Function central unit version 3.0 onward

Format:

	Headerbyte	Marker	Data 1	Data 2	Xor-byte
Binary:	1110 0011	0000 0111	Adr High	Adr Low	Xor-byte
Hex :	0xE3	0x07	AH	AL	Xor-byte
Dez :	227	7	AH	AL	Xor-byte

Description:

The answer of this command is the type of functions f0 to f12 as „continuous“ or „momentary.
Range for locomotive adress is 0 to 9999.
AH/AL: Locomotive adress is calculated as described in 2.1.11.

The answers are described in 2.1.10.

Remarks:

None.

2.2.19 Locomotive Control

2.2.19.1 Locomotive Command up to LZ-version 1.5

Format:

	Headerbyte	Data 1	Data 2	Data 3	Xor-byte
Binary:	1011 0011	Locadress	Locdata 1	Locdata 2	Xor-byte
Hex :	0xB3	Locadress	Locdata 1	Locdata 2	Xor-byte
Dez :	179	Locadress	Locdata 1	Locdata 2	Xor-byte

Description:

Up to version 1.5 only 14 speedsteps were used. The rnage for locomotive adress is 0 .. 99.
Locdata 1 and Locdata 2 are described in 2.1.8.1.

Remarks:

None.

2.2.19.2 Locomotive Command up to LZ-version 2.3

Format:

	Headerbyte	Data 1	Data 2	Data 3	Data 4	Xor-byte
Binary:	1011 0100	Locadress	Locdata 1	Locdata 2	ModSel	Xor-byte
Hex :	0xB4	Locadress	Locdata 1	Locdata 2	ModSel	Xor-byte
Dez :	180	Locadress	Locdata 1	Locdata 2	ModSel	Xor-byte

Description:

Because of the possibility of having 14, 27 o 28 speedsteps, the mode select byte is used here so that the central unit can modulate the track signal correct for a decoder.
Range for locomotive adress is 0 to 99.
Locdata 1, Locdata 2 and ModSel are described in 2.1.9.1.

Remarks:

None.

2.2.19.3 Locomotive Speed Commands central unit version 3.0 onward

This new speed command is split into 4 variants according to 14, 27, 28 and 128 speedsteps. This split is done using different marker bytes. The speed coding itself is described for 14, 27 and 28 speedsteps in 2.1.9.1, the speed coding for 128 speedsteps is described in 2.1.10.1.

Speed command 14 speedsteps:

Format:

	Headerbyte	Marker	Data 1	Data 2	Data 3	Xor-byte
Binary:	1110 0100	0001 0000	Adr High	Adr Low	R000 VVVV	Xor-byte
Hex :	0xE4	0x10	AH	AL	RV	Xor-byte
Dez :	228	16	AH	AL	RV	Xor-byte

Speed command 27 speedsteps:

Format:

	Headerbyte	Marker	Data 1	Data 2	Data 3	Xor-byte
Binary:	1110 0100	0001 0001	Adr High	Adr Low	R00V VVVV	Xor-byte
Hex :	0xE4	0x11	AH	AL	RV	Xor-byte
Dez :	228	17	AH	AL	RV	Xor-byte

Speed command 28 speedsteps:

Format:

	Headerbyte	Marker	Data 1	Data 2	Data 3	Xor-byte
Binary:	1110 0100	0001 0010	Adr High	Adr Low	R00V VVVV	Xor-byte
Hex :	0xE4	0x12	AH	AL	RV	Xor-byte
Dez :	228	18	AH	AL	RV	Xor-byte

Speed command 128 speedsteps:

Format:

	Headerbyte	Marker	Data 1	Data 2	Data 3	Xor-byte
Binary:	1110 0100	0001 0011	Adr High	Adr Low	RVVV VVVV	Xor-byte
Hex :	0xE4	0x13	AH	AL	RV	Xor-byte
Dez :	228	19	AH	AL	RV	Xor-byte

Description:

The command only contains speed information and direction. Functions are switched separat.
AH/AL: Locomotive adress is calculated as described in 2.1.11.

Remarks:

None.

2.2.19.4 Function Commands central unit version 3.0 onward

The function command is split into 3 variants according to functions in group 1 (F0..F4), group 2 (F5..F8) and group 3 (F9..F12). The split is done using different marker byes.

Function command group 1:

Format:

	Headerbyte	Marker	Data 1	Data 2	Data 3	Xor-byte
Binary:	1110 0100	0010 0000	Adr High	Adr Low	000F FFFF	Xor-byte
Hex :	0xE4	0x20	AH	AL	Group 1	Xor-byte
Dez :	228	32	AH	AL	Group 1	Xor-byte

Function command group 2:

Format:

	Headerbyte	Marker	Data 1	Data 2	Data 3	Xor-byte
Binary:	1110 0100	0010 0001	Adr High	Adr Low	0000 FFFF	Xor-byte
Hex :	0xE4	0x21	AH	AL	Group 2	Xor-byte
Dez :	228	33	AH	AL	Group 2	Xor-byte

Function command group 3:

Format:

	Headerbyte	Marker	Data 1	Data 2	Data 3	Xor-byte
Binary:	1110 0100	0010 0010	Adr High	Adr Low	0000 FFFF	Xor-byte
Hex :	0xE4	0x22	AH	AL	Group 3	Xor-byte
Dez :	228	34	AH	AL	Group 3	Xor-byte

Description:

For the functions is defined:
Group 1: 0 0 0 F0 F4 F3 F2 F1
Group 2: 0 0 0 0 F8 F7 F6 F5
Group 3: 0 0 0 0 F12 F11 F10 F9
where Fx=1 means the function is on.
Range for locomotive adress is 0 to 9999.
AH/AL: Locomotive adress is calculated as described in 2.1.11.

Remarks:

None.

2.2.19.5 Set Function Type central unit version 3.0 onward

The central unit LZ100 stores for every locomotive adress the type of its functions as continuous or momentary. XpressNet devices can ask for this type to adapt their user interface. This functionality is very interesting for sound effects. As is done with the function status, the type of function is also divided into 3 groups distinguished using the marker byte.

Set function type in group 1:

Format:

	Headerbyte	Marker	Data 1	Data 2	Data 3	Xor-byte
Binary:	1110 0100	0010 0100	Adr High	Adr Low	000S SSSS	Xor-byte
Hex :	0xE4	0x24	AH	AL	Group 1	Xor-byte
Dez :	228	36	AH	AL	Group 1	Xor-byte

Set function type in group 2:

Format:

	Headerbyte	Marker	Data 1	Data 2	Data 3	Xor-byte
Binary:	1110 0100	0010 0101	Adr High	Adr Low	0000 SSSS	Xor-byte
Hex :	0xE4	0x25	AH	AL	Group 2	Xor-byte
Dez :	228	37	AH	AL	Group 2	Xor-byte

Set function type in group 3:

Format:

	Headerbyte	Marker	Data 1	Data 2	Data 3	Xor-byte
Binary:	1110 0100	0010 0110	Adr High	Adr Low	0000 SSSS	Xor-byte
Hex :	0xE4	0x26	AH	AL	Group 3	Xor-byte
Dez :	228	38	AH	AL	Group 3	Xor-byte

Description:

It is defined:
Group 1: 0 0 0 S0 S4 S3 S2 S1
Group 2: 0 0 0 0 S8 S7 S6 S5
Group 3: 0 0 0 0 S12 S11 S10 S9
where Sx = 1 means the function type is momentary.
The range for locomotive adress is 0 to 9999.
Locomotive adress is calculated as described in 2.1.11.

Remarks:

None.

2.2.20 Double Headers up to LZ-version 2.3

2.2.20.1 Assemble Double Header

Format:

	Headerbyte	Marker	Data 1	Data 2	Xor-byte
Binary:	1100 0011	0000 0101	Adress 1	Adress 2	Xor-byte
Hex :	0xC3	0x05	Adress 1	Adress 2	Xor-byte
Dez :	195	5	Adress 1	Adress 2	Xor-byte

Description:

The locomotives in data 1 and data 2 are assembled to a double header in the central unit. This means that a speed command to one locomotive is also send to the other.
Locomotives 1 to 99 can be assembled.
If the assembly is not successful, an error message described in 2.1.16 is send.

Remarks:

None.

2.2.20.2 Disassemble Double Header

Format:

	Headerbyte	Marker	Data 1	Data 2	Xor-byte
Binary:	1100 0011	0000 0100	Adress 1	Adress 2	Xor-byte
Hex :	0xC3	0x04	Adress 1	Adress 2	Xor-byte
Dez :	195	4	Adress 1	Adress 2	Xor-byte

Description:

The double header of locomotive in data 1 and data 2 is disassembled in the central unit.
If the disassembly is not successful, an error message described in 2.1.16 is send.

Remarks:

None.

2.2.21 Double Headers central unit version 3.0 onward

2.2.21.1 Assemble Double Header

Format:

	Headerbyte	Marker	Data 1	Data 2	Data 3	Data 4	Xor-byte
Binary:	1110 0101	0100 0011	Adr High 1	Adr Low 1	Adr. High 2	Adr Low 2	Xor-byte
Hex :	0xE5	0x43	AH1	AL1	AH 2	AL 2	Xor-byte
Dez :	229	67	AH1	AL1	AH 2	AL 2	Xor-byte

Description:

The locomotives in data 1/2 and data 3/4 are assembled to a double header in the central unit. This means that a speed command to one locomotive is also send to the other.
Locomotives 1 to 9999 can be assembled.
AH/AL: Locomotive adress is calculated as described in 2.1.11.
If the assembly is not successful, an error message described in 2.1.17 is send.

Remarks:

This command replaces the double header command used up to now.

2.2.21.2 Disassemble Double Header

Format:

	Headerbyte	Marker	Data 1	Data 2	Data 3	Data 4	Xor-byte
Binary:	1110 0101	0100 0011	Adr High 1	Adr Low 1	0000 0000	0000 0000	Xor-byte
Hex :	0xE5	0x43	AH1	AL1	0x00	0x00	Xor-byte
Dez :	229	67	AH1	AL1	0x00	0x00	Xor-byte

Description:

The locomotive in data 1/2 is removed from the double header and therefor the double header is removed in the central unit. Because data 3 and 4 are 0x00, the central unit can decide that it should disassemble a double header.
AH/AL: Locomotive adress is calculated as described in 2.1.11.
If the disassembly is not successful, an error message described in 2.1.17 is send.

Remarks:

This command replaces the double header command used up to now.

2.2.22 Programming On Main central unit version 3.0 onward

Programming on main means that cv´s of a decoder can be modified during the decoder is placed on the normal track. A programming track is not needed in this case. However the adress of a decoder cannot be modified because it is needed in the command.
Central units which does not support programming on main send „command unknown“ to the XpressNet device. In contrast to the XpressNet commands „programming on the programming track“, cv´s 0..1023 are possible here. XpressNet devices should not send cv´s containing adresses. This is per definition not allowed and decoders do not accept them.

2.2.22.1 Programming On Main Write Byte

Format:

	Headerbyte	Marker	Data 1	Data 2	Data 3	Data 4	Data 5	Xor-byte
Binary:	1110 0110	0011 0000	Adr High	Adr Low	1110 11CC	CCCC CCCC	DDDD DDDD	Xor-byte
Hex :	0xE6	0x30	AH	AL	0xEC + C	CV	D	Xor-byte
Dez :	230	48	AH	AL	236 + C	CV	D	Xor-byte

Description:

Data 1 and data 2 define the locomotive which the command belongs to in the range of 1..9999.
AH/AL: Locomotive adress is calculated as described in 2.1.11.
Because cv´s 0..1023 are possible (=10 bit), the upper 2 bits (MSB´s) are placed in data 3. The rest of the cv adress is in data 4. The value to be programmed is in data 5.
The cv adress is sended to the central unit as is needed on the track which means decremented by 1.

Remarks:

None.

2.2.22.2 Programming On Main Write Bit

Format:

	Headerbyte	Marker	Data 1	Data 2	Data 3	Data 4	Data 5	Xor-byte
Binary:	1110 0110	0011 0000	Adr High	Adr Low	0111 11CC	CCCC CCCC	1111 WBBB	Xor-byte
Hex :	0xE6	0x30	AH	AL	0x7C + C	CV	WB	Xor-byte
Dez :	230	48	AH	AL	124 + C	CV	WB	Xor-byte

Description:

Data 1 and data 2 define the locomotive which the command belongs to in the range of 1..9999.
AH/AL: Locomotive adress is calculated as described in 2.1.11.
Because cv´s 0..1023 are possible (=10 bit), the upper 2 bits (MSB´s) are placed in data 3. The rest of the cv adress is in data 4. The value to be programmed is in data 5 and is coded as follows:
W is the bit value 0 or 1.
Bits B2, B1, B0 define the position of the bit (bitposition 0 .. 7).

The cv adress is sended to the central unit as is needed on the track which means decremented by 1.

Remarks:

None.

2.2.23 Multi Units central unit version 3.0 onward

2.2.23.1 Add Locomotive To Multi Unit or Create Multi Unit

A locomotive can be added to a multi unit if it is not already a member of a mu. If the locomotive is the first locomotive in the mu, the mu is created by the central unit. In addition, the direction of the locomotive in the mu can be defined so that a locomotive can be in the „wrong“ direction in the mu but is running correct. This is done with an direction marker bit (R).

Format:

	Headerbyte	Marker	Data 1	Data 2	Data 3	Xor-byte
Binary:	1110 0100	0100 000R	Adr High	Adr Low	MU	Xor-byte
Hex :	0xE4	0x40 + R	AH	AL	MU	Xor-byte
Dez :	228	64 + R	AH	AL	MU	Xor-byte

Description:

R: R = 0 means that the locomotive direction is not inverted. If the mu runs forward, the locomotive runs forward too.
R =1 means the direction of the locomotive is inverted.

Data 1 and Data 2 define the locomotive adress in the range of 1..9999 which is to be added to the mu.

AH/AL: Locomotive adress is calculated as described in 2.1.11.

MU: This is the mu baseadress in the range of 1 .. 99.

Remarks:

Per definition a locomotive cannot be added in a multi unit with the same adress.

2.2.23.2 Remove Locomotive From Multi Unit or Delete Multi Unit

A locomotive can be removed from a multi unit, if it is a member of this mu. By removing the last locomotive, the mu itself is deleted.

Format:

	Headerbyte	Marker	Data 1	Data 2	Data 3	Xor-byte
Binary:	1110 0100	0100 0010	Adr High	Adr Low	MU	Xor-byte
Hex :	0xE4	0x42	AH	AL	MU	Xor-byte
Dez :	228	66	AH	AL	MU	Xor-byte

Description:

Data 1 and data 2 are the locomotive adress in the range of 1..9999 to be removed.

AH/AL: Locomotive adress is calculated as described in 2.1.11.

MU: This is the base adress of the mu in the range of 1 .. 99, under which the mu can be controlled.

Remarks:

None.

2.2.24 Adress Search Commands central unit version 3.0 onward

With the introduction of multi units and an expanded stack management in the central units it was neccessary that XpressNet devices can search for certain locomotive adresses to realize a comfortable user interface.

2.2.24.1 Search Member Of A Multi Unit

The decision forward or backward search is done with a bit in the marker byte.

Format:

	Headerbyte	Marker	Data 1	Data 2	Data 3	Xor-byte
Binary:	1110 0100	0000 00RR	MU	Adr High	Adr Low	Xor-byte
Hex :	0xE4	0x01 + R	MU	AH	AL	Xor-byte
Dez :	228	1 + R	MU	AH	AL	Xor-byte

Description:

To have fast access to the locomotives in a mu, e.g. to switch functions, the central unit responds on this command with the next (forward search) or previous (backward search) adress following or preceding the locomotive adress in data 2 and data 3.

For central units version 3.x only the forward search is implemented.

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

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

Data 1 is the multi unit base adress in the range of 1..99 the search belongs to.

Data 2 and data 3 are the locomotive adress from 1..9999, to which the next or previous adress should be searched.

AH/AL: Locomotive adress is calculated as described in 2.1.11.

MU: This is the base adress of the mu in the range of 1 .. 99, under which the mu can be controlled.

The search result is send to the XpressNet device in a format described in 2.1.13.

Remarks:

None.

2.2.24.2 Search Multi Unit

The decision forward or backward search is done with a bit in the marker byte.

Format:

	Headerbyte	Marker	Data 1	Xor-byte
Binary:	1110 0010	0000 0RRR	MU	Xor-byte
Hex :	0xE2	0x03 + R	MU	Xor-byte
Dez :	226	3 + R	MU	Xor-byte

Description:

This command prompts the central unit to search the next or previous base adress which follows or precedes the requested adress.

For central units version 3.x only the forward search is implemented.

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

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

MU: This is the base adress of the mu in the range of 1 .. 99, under which the mu can be controlled.

The serarch result is send to the XpressNet device in a format described in 2.1.13.

Remarks:

None.

2.2.24.3 Search Locomotive In Central Unit Stack

The decision forward or backward search is done with a bit in the marker byte.

Format:

	Headerbyte	Marker	Data 1	Data 2	Xor-byte
Binary:	1110 0011	0000 01RR	Adr High	Adr Low	Xor-byte
Hex :	0xE3	0x05 + R	AH	AL	Xor-byte
Dez :	227	5 + R	AH	AL	Xor-byte

Description:

The command prompts the central unit to search the next or previous locomotive adress which follows or precedes the requested adress int the central unit stack.

For central units version 3.x only the forward search is implemented.

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

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

Data 1 and data 2 are the locomotive adress in the range of 0..9999 to which the following or preceding adress should be searched.
AH/AL: Locomotive adress is calculated as described in 2.1.11.
The serarch result is send to the XpressNet device in a format described in 2.1.13.

Remarks:

None.

2.2.25 Delete Locomotive In Central Unit Stack

Format:

	Headerbyte	Marker	Data 1	Data 2	Xor-byte
Binary:	1110 0011	0100 0100	Adr High	Adr Low	Xor-byte
Hex :	0xE3	0x44	AH	AL	Xor-byte
Dez :	227	68	AH	AL	Xor-byte

Description:

Data 1 and data 2 are the locomotive adress in the range of 1..9999, which should be deleted from central unit stack.
AH/AL: Locomotive adress is calculated as described in 2.1.11.

Remarks:

The problem of stack overflow is present in central units with not much hardware capabilities. These central units are not able to hold every used locomotive with all the data required. The stack is used for track signal output for the locomotives.
The XpressNet device which has deleted a locomotive from the stack should ensure that it has a valid locomotive adress to continue. If this is the same adress as before, the locomotive is deleted from stack and immediatly placed in again.
Not all central units support this command.

3 Commands Central Unit To PC

The exact description of the bytes is found in their corresponding chapters. N in the header byte means the number of following databytes. Vx.y means central unit version x.y.

V3	Command	Header	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
x	BC Everything On	0x61	0x01	0x60					
x	BC Everything Off	0x61	0x00	0x61					
x	BC Stop All Locos	0x81	0x00	0x81					
x	BC Programmingmode	0x61	0x02	0x63					
x	BC Feedback	0x40 + N	ADR_1	DAT_1	ADR_2	DAT2	etc.	etc.	X-Or
x	P-Info Short Circuit	0x61	0x12	X-Or					
x	P-Info No Data	0x61	0x13	X-Or					
x	P-Info Busy	0x61	0x1f	X-Or					
x	P-Info Ready	0x61	0x11	X-Or					
x	P-Info Data 3 Byte	0x63	0x10	EE	DAT	X-Or			
x	P-Info Data 4 Byte	0x63	0x14	CV	DAT	X-Or			
	Software LZ up to V2.3	0x62	0x21	DAT	X-Or				
x	Software LZ >= V3.0	0x63	0x21	DAT1	DAT2	X-Or			
x	Status Central Unit	0x62	0x22	DAT	X-Or				
x	Central Unit Busy	0x61	0x81	X-Or					
x	Command Unknown	0x61	0x82	X-Or					
x	Switching Information	0x42	ADR	DAT	X-Or				
	Loco Free up to V1.5	0x83	Locadr	Locdat 1	Locdat 2	X-Or			
	Loco Occupied up to V1.5	0xA3	Locadr	Locdat 1	Locdat 2	X-Or			
	Loco Free up to V2.3	0x84	Locadr	Locdat 1	Locdat 2	ModSel	X-Or		
	Loco Occupied up to V2.3	0xA4	Locadr	Locdat 1	Locdat 2	ModSel	X-Or		
x	Loco Info >= V3.0	0xE4	Marker	Speed	FKT0	FKT1	X-Or		
x	MU-Member >= V3.0	0xE5	Marker	Speed	FKT0	FKT1	MU	X-Or	
x	MU-Baseadress>= V3.0	0xE2	Marker	Speed	X-Or				
x	Loco In DH >= V3.0	0xE6	Marker	Speed	FKT0	FKT1	ADR High	ADR Low	X-Or
x	Loco Occupied >= V3.0	0xE3	0x40	ADR High	ADR Low	X-Or			
x	Function Type >= V3.0	0xE3	0x50	STAT 0	STAT 1	X-Or			
x	Searchresult >= V3.0	0xE3	0x30 + K	ADR High	ADR Low	X-Or			
	DH Free up to V1.5	0xC5	0x04	Locadr 1	Locdat 1	Locdat 2	Locadr 2	X-Or	
	DH Occupied up to V1.5	0xC5	0x05	Locadr 1	Locdat 1	Locdat 2	Locadr 2	X-Or	
	DH Free up to V2.3	0xC6	0x04	Locadr 1	Locdat 1	Locdat 2	Locadr 2	ModSel	X-Or
	DH Occupied up to V2.3	0xC6	0x05	Locadr 1	Locdat 1	Locdat 2	Locadr 2	ModSel	X-Or
x	DH Error	0x61	0x80 + F	X-Or					
x	Errormessage >= V3.0	0xE1	0x80 + F	X-Or					

4 Commands PC To Central Unit

The exact description of the bytes is found in their corresponding chapters. N in the header byte means the number of following databytes. Vx.y means central unit version x.y.

V3	Command	Header	Marker	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6
x	Everything On	0x21	0x81	0xA0					
x	Everything Off	0x21	0x80	0xA1					
x	Stop All Locos	0x80	0x80						
	Stop One Loco up to V2.3	0x91	Lokadr	X-Or					
x	Stop One Loco >= V3.0	0x92	ADR High	ADR Low	X-Or				
	Stop Locos up to V2.3	0x90 + N	Locadr 1	Locadr 2	etc.	Locadr N	X-Or		
x	Prog.-Read Register	0x22	0x11	REG	X-Or				
x	Prog.-Read CV	0x22	0x15	CV	X-Or				
x	Prog.-Read Paging	0x22	0x14	CV	X-Or				
x	Request Prog.-Result	0x21	0x10	0x31					
x	Prog.-Write Register	0x23	0x12	REG	DAT	X-Or			
x	Prog.-Write CV	0x23	0x16	CV	DAT	X-Or			
x	Prog.-Write Paging	0x23	0x17	CV	DAT	X-Or			
x	Request Softwareversion	0x21	0x21	0x00					
x	Request Status	0x21	0x24	0x05					
	Set Startmode	0x22	0x22	00000M00	X-Or				
x	Request Switching Information	0x42	ADR	Nibble	X-Or				
x	Switching Command	0x52	ADR	DAT	X-Or				
	Request Locinfo up to V1.5	0xA1	Locadr	X-Or					
	Request Locinfo up to V2.3	0xA2	Locadr	ModSel	X-Or				
x	Request Locinfo >= V3.0	0xE3	0x00	ADR High	ADR Low	X-Or			
x	Request Function Type >= V3.0	0xE3	0x07	ADR High	ADR Low	X-Or			
	Loco Control up to V1.5	0xB3	Locadr	Locdat 1	Locdat 2	X-Or			
	Loco Control up to V2.3	0xB4	Locadr	Locdat 1	Locdat 2	ModSel	X-Or		
x	Loco Speed Command >= V3.0	0xE4	Marker	ADR High	ADR Low	Speed	X-Or		
x	Loco Function Cmd >=V3.0	0xE4	Marker	ADR High	ADR Low	Group	X-Or		
x	Set Function Type >= V3.0	0xE4	Marker	ADR High	ADR Low	Group	X-Or		
	DH Assemble up to V2.3	0xC3	0x05	Locadr 1	Locadr 2	X-Or			
	DH Disaasemble up to V2.3	0xC3	0x04	Locadr 1	Locadr 2	X-Or			
x	DH Commands >= V3.0	0xE5	0x43	ADR1 H	ADR1 L	ADR2 H	ADR2 L	X-Or	
x	Prog. On Main Byte >= V3.0	0xE6	0x30	ADR High	ADR Low	0xEC + C	CV	DAT	X-Or
x	Prog. On Main Bit >= V3.0	0xE6	0x30	ADR High	ADR Low	0x7C + C	CV	DAT	X-Or
x	Add Loco To MU >= V3.0	0xE4	0x40 + R	ADR High	ADR Low	MU	X-Or		
x	Remove Loco From MU	0xE4	0x42	ADR High	ADR Low	MU	X-Or		
x	Search Loco In Mu >= V3.0	0xE4	0x01 + R	MU	ADR High	ADR Low	X-Or		
x	Search MU >= V3.0	0xE2	0x03 + R	MU	X-Or				
x	Search Stack >= V3.0	0xE3	0x05 + R	ADR High	ADR Low	X-Or			
x	Delete From Stack >= V3.0	0xE3	0x44	ADR High	ADR Low	X-Or			