



01/2005

# **XpressNet Specification**

Supplement for using the  
LI-USB interface

## 1.6 LI-USB Interface Communication Specifications

The LI-USB provides two possibilities for communication with the interface. In the first method, the requests and responses can be as sent and received in the same way as used with the LI101F serial interface. The driver necessary for the installation is included with the LI-USB interface.

The included LI-USB server provides a second method that provides a simple extension to let more applications access the LI-USB interface at the same time.

### 1.6.1 Instruction structure when transmitting using the serial interface

Any request that is sent to the interface must be preceded by the sequence of **"0xFF 0xFE"**. This sequence is not included in the calculation of the check sum for the request.

As an example to send the "Resume operations request" to the interface, you must transmit the following data: **"0xFF 0xFE 0x21 0x81 0xA0"**

Every request to the interface is, followed up with a response with an answer that is either the general answer of **"0xFF 0xFE 0x01 0x04 0x05"** which indicates that the command was transmitted successfully to the command station or one of the error messages documented in section 1.5.2. Answers to a command that are generated by the LI-USB interface are always preceded by **"0xFF 0xFE"** as the header.

**Attention:** In service mode programming, the response can be delayed up to a minute before an answer is transmitted back.

As long as the interface has not transmitted back a receipt for a request, no further requests may be sent to the interface. A new request may be sent only after a timeout occurs (5 sec in the normal mode, 1.5 min in the service mode programming mode)

When a request is sent to the interface, the next answer from the interface is always the command answer. Broadcasts accumulating in the interface in between the request and response are stored and transmitted after the response to the request.

Broadcasts as well as unexpected reports, that is all unrequested information, that are sent by the LI-USB also contain the qualifier of **"0xFF 0xFD"** as a header so that the client can easily distinguish these command responses from normal request responses.

Please, take note that the **"0xFF 0xFD"** header is not indicated in the remainder of the XpressNet specification, however, you must send in before every request to the interface.

### 1.6.2 Instruction structure when using the LI-USB server

The substantial advantage for the use of the LI-USB server over the operation of the serial interface is that several applications "share" an interface and can use the interface with equal rights. Concern that the application which has sent a command to the interface also receives the answer is taken care of by the LI-USB server. Likewise communication with the USB interface as well as the recognition of the interface are accomplished by the LI-USB server, so that all time-critical operations are accomplished by the server instead of the application. The enclosed LI-USB server supports up to two applications at the same time both of which must be installed on the same computer as the LI-USB server.

The LI-USB server makes two TCP/IP ports available to the client applications for communication with the interface.

Port 5550: Requests from the application are transmitted to the interface and the responses concerning this request are transmitted back from the server over this port.

Port 5551: The LI-USB server transmits all unrequested information to the client application over this port. All data received on this port is ignored.

Both ports are controlled using a time out mechanism. If no characters are received at a port for longer than 60 seconds, the server terminates the connection.

Therefore, at a minimum, the client must send a request or a character that will be ignored to port 5550 at least once per minute. A character must also be sent to port 5551 at least once per minute. While port 5551 ignores all characters which are sent to it, sending transmissions serves to prevent a timeout.

The communication is carried out using plain text with hexadecimal representation. Every command line must be followed with the string <cr> <lf>.

An example illustrates this point:

To transmit a "Stop Operations Request" (emergency off) you must send the following string to port 5550: "**2180 A1 <cr> <lf>**".

As soon as the request can be completed successfully, the server answers on port 5550 with the string "**010405 <cr> <lf>**" which shows that that the command was transmitted successfully. Concurrently the following string that indicates that the track power is off is transmitted over the broadcast port 5551: "**610061 <cr> <lf>**".

A correctly built application should always combine the information that is transmitted over the two ports. However, only one individual port can be opened at a time.

The 5550 port ignores all characters apart from the numbers 0 -9 and the letters A – F or a - f. The 5551 port ignores all characters which it receives except for the purpose of keeping the port open.

## **Programming mode:**

If a client enters service mode programming using the programming track by the initiation of a programming command, then port 5550 is terminated for all other clients apart from the client which has initiated the programming request.

Connections on port 5551 remain unchanged so that all clients can learn when programming mode has ended. A connection structure to the port 5550 during the programming mode is closed. A connection structure to the port 5551 when programming mode is active has an immediate response of the broadcast "Service Mode entry" "**610263 <cr> <lf>**" which informs the registering client that a programming process is underway.

If the client that initiated service mode entry closes the connection on port 5550 during a programming process, the server cancels the programming mode automatically to allow a continuing to work.

During service mode programming the broadcast "Service Mode entry" can be repeatedly sent to the clients.