Manual

LIN2CAN & ConfigurationTool

IXXAT

Headquarter
IXXAT Automation GmbH
Leibnizstr. 15
D-88250 Weingarten

Tel.: +49 (0)7 51 / 5 61 46-0
Fax: +49 (0)7 51 / 5 61 46-29
Internet: www.ixxat.de
e-Mail: info@ixxat.de

Support

In case of unsolvable problems with this product or other IXXAT products please contact IXXAT in written form by:

Fax: +49 (0)7 51 / 5 61 46-29
e-Mail: support@ixxat.de

Copyright

Document number: 4.01.0130.20000
Version: 1.9

1 General

1.1 Definitions, acronyms, abbreviations

CAN	**C**ontroller **A**rea **N**etwork
HTML	**H**yper**t**ext **M**arkup **L**anguage
L2CAPI	LIN2CAN API (Programming interface)
LDF	**L**IN **D**escription **F**ile
LIN	**L**ocal **I**nterconnect **N**etwork
LIN- message	Data telegram of the LIN bus. A data telegram consists of an 8-bit identifier field and an up to 8-byte data field.
XML	E**x**tensible **M**arkup **L**anguage

1.2 Overview

This handbook is intended to familiarize you with your LIN2CAN Gateway. Please read this handbook before beginning the installation.

The LIN2CAN Configuration Tool (L2C_Cfg_Tool.EXE) is used for configuration and commissioning of the LIN2CAN- Gateway device. In addition a simple LIN-monitor is integrated in the configuration program that displays LIN-messages and errors on the bus. Once created, a configuration can be exported to an XML file. This file can then be changed and imported again in order to reload the configuration onto a device.

Sections 1 and 2 describe the LIN2CAN device. Section 3 explains the installation of the configuration software. In Section 4 the operating modes of the device are described, Section 5 and the following show the configuration of the operating modes and monitoring with the aid of the LIN2CAN Configuration Tool.

1.3 Support

For more information on our products, FAQ lists and installation tipps, please refer to the support area on our homepage (http://www.ixxat.com). There you will also find information on current product versions and available updates.

1.4 Returning hardware

If it is necessary to return hardware to us, please download the relevant RMA form from our homepage and follow the instructions on this form.

2 LIN2CAN device description

2.1 Features

- Power supply 7 - 16 V DC, Industrial 10 - 32V DC
- Standby power consumption < 1 mA
- LIN bus interface via the LIN Transceiver TJA1020
- CAN High-Speed bus interface according to ISO/IS 11898-2 with galvanic isolation
- CAN Low-Speed bus interface according to ISO 11898-3 (“Fault Tolerant“)
- Serial port (RS232) for Gateway configuration
- 16 Bit Microcontroller
- LIN Master and Slave operation possible

2.2 Block diagram

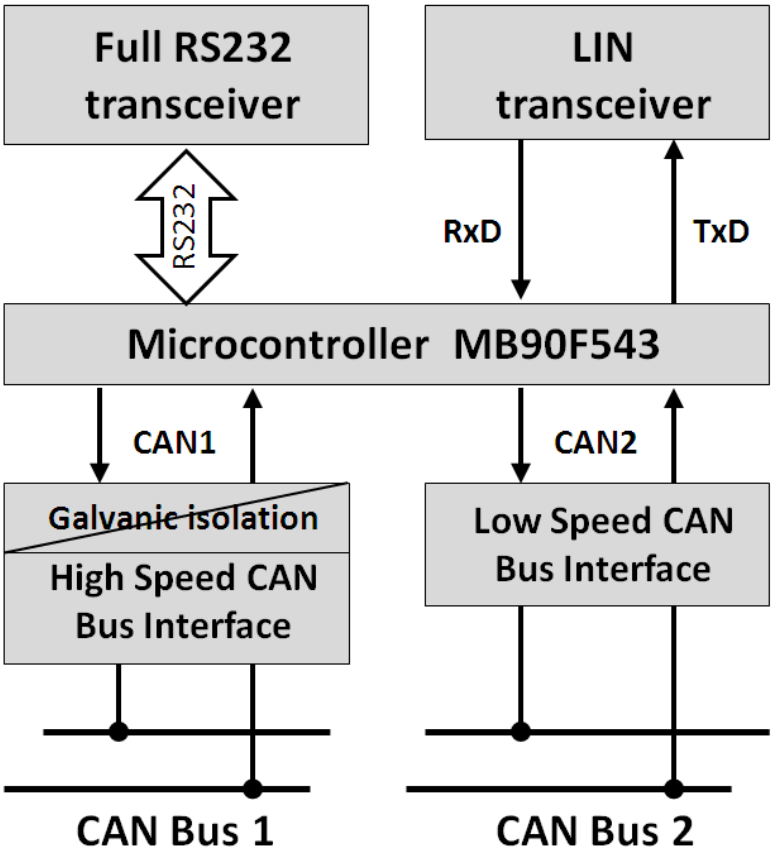


Fig. 2.2-1: Block circuit diagram LIN2CAN Gateway

2.3 Pin allocation

Fig. 2.3-2: Pin allocation

2.3.1 Supply voltage

The device is supplied at connector X1 with a DC voltage of 7 V – 16 V (Industrial 10V - 32V). A prefabricated cable for the power supply is included in the scope of delivery. The pin allocation is given in Table 2.3-1.

Pin No. X1	Signal	Lead color
1	+	white
2	-	brown
3	GROUND	shield

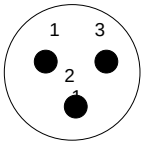


Table 2.3-1: Pin allocation Power

2.3.2 Serial port RS232

The signals of the serial port are connected to the 9-pin Sub-D connector X2 (see Table 2.3-2).

Pin No. X2	Signal
1	DCD
2	RX
3	TX
4	DTR
5	GND
6	DSR
7	RTS
8	CTS
9	RI

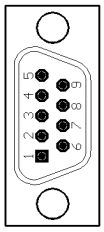


Table 2.3-2: Pin allocation RS232

2.3.3 CAN High Speed

The signals of the CAN bus interface according to ISO/IS 11898-2 are connected to the 9-pin Sub-D connector X3 (see Table 2.3-3).

Pin No. X3	Signal
1	-
2	CAN Low
3	GND
4	-
5	-
6	-
7	CAN High
8	-
9	-

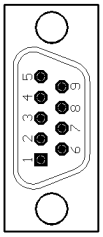


Table 2.3-3: Pin allocation CAN HS

2.3.4 CAN Low Speed / LIN

The bus interface according to ISO/IS 11898-3 (“Fault Tolerant“) and the LIN bus interface are connected to the 9-pin Sub-D connector X4 (see Table 2.3-4).

Pin No. X4	Signal
1	LIN
2	CAN Low
3	GND
4	-
5	-
6	-
7	CAN High
8	-
9	-

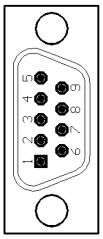


Table 2.3-4: Pin allocation CAN LS / LIN

2.4 CAN bus terminal

There is no bus terminal resistor for the CAN-bus assembled on the LIN2CAN Gateway.

2.5 LED- displays

The LIN2CAN Gateway has four 2-color LEDs. These are status LEDs allocated to the interfaces. The following table shows the LED statuses and their meaning.

LED	Status	Meaning
PWR (Power)	green flashing (1 Hz)	device active
	short red flashing	no configuration data are read out of or written onto the Flash.
	off	device in Power-Down mode
CAN	lit green	CAN message successfully received or transmitted
	lit red	CAN message could not be transmitted or an overrun occurred in the CAN Receive queue
	continually lit red	CAN Controller in Bus-off status
	off	no CAN message traffic
LIN	lit green	LIN message successfully received or transmitted
	lit red	a LIN-error occurred when receiving or transmitting a LIN-message
	off	no LIN-message traffic
SER (Serial)	lit green	data traffic between PC and LIN2CAN Gateway
	off	no data traffic

Table 2.5-5: Meaning of the LED-displays

3 Installation and Start

3.1 System requirements

The LIN2CAN Configuration Tool has the following PC-system requirements

- Pentium II or higher (500 MHz recommended)
- Windows 2000, Windows XP, Windows 7
- one free RS232 interface

3.2 Installation

To install the LIN2CAN Configuration Tools and the programming interface (L2CAPI), insert the program CD supplied in the CD-drive and start the installation program there by running the file "L2C_Configtool_Setup.exe ".

3.3 Connection of the LIN2CAN Gateway

To connect the device to the PC, a serial, fully wired, crossed cable (Null-modem-cable) and one free RS232 interface are required.

3.4 Starting the LIN2CAN Configuration Tool

In order to configure the LIN2CAN Gateway, the following steps must be carried out in order.

(1) Disconnect voltage supply of the device
(2) Connect device to the PC with RS232 line
(3) Reconnect voltage supply of the device (device detects PC and starts in PC-interface mode)
(4) Start Configuration Tool (L2C_Cfg_Tool.exe)
(5) Select RS232 interface used under menu point Device → Options
(6) Select baud rate according to HW version under point Device → Options
HW version 1.x ➔ baud rate = 57600
HW version 2.x ➔ baud rate = 115200
(7) Make connection to the device by clicking on the LED in the toolbar or by clicking on the menu point Device → Connect

4 LIN2CAN Gateway Operating Modes

4.1 Overview

The LIN2CAN Gateway is a universal device for analyzing LIN-networks via the CAN-Bus, and for emulating LIN-Slave or LIN-Master modules. In addition, the device can be used as a LIN- PC-interface with PC configuration and monitoring software or for PC-compatible LIN-device development.

The LIN2CAN Gateway supports five operating modes

- LIN- CAN- Gateway Master/Slave
- LIN- CAN- Gateway Slave
- LIN-Master/Slave
- LIN-Slave
- LIN-PC-Interface

The LIN-PC-Interface mode can in turn be sub-divided into LIN-PC-Interface Master/Slave and LIN-PC-Interface Slave.

The first four modes can be configured via the PC as Startup-mode. If the device is **not** connected to a PC with Power-On, it starts in the configured Startup-mode and works in stand-alone mode. If the device is connected by serial cable to a PC with Power-On, it starts in LIN-PC- Interface Slave mode.

4.2 LIN-CAN-Gateway Master/Slave

The LIN-CAN-Gateway Master/Slave mode allows the transparent transmission of LIN-messages to a CAN-network, where the device acts as a LIN-Master/Slave in other words the device works as a stand-alone LIN bus arbiter with LIN- Slave function. Here several configurable schedule lists are available, whereas only one is processed cyclically. The layout of the Schedule table is described under LIN-Master/Slave.

Received LIN- messages in the mode LIN-CAN-Gateway are transmitted to the CAN, where the LIN-identifiers are translated into CAN-identifiers. The data of this message are passed on via the CAN-Object. The device also provides a transmit function on LIN via CAN. Here CAN-identifiers are translated into LIN-identifiers and transmitted once on request by the LIN-Master (internal Schedule table) in the LIN network. The data of the CAN- Object are stored in a buffer and then transmitted. A PC-configurable Gateway table is available for the translation of messages, which is stored on the device. In addition it is possible to map errors occurring on the LIN-Bus (e.g. timeout according to LIN-header) onto a CAN-message.

The Gateway functionality can be activated and deactivated via CAN. For this, a CAN-Object can be specified, where the first data byte of the CAN-objects is interpreted as a switch. If the content of this data byte is zero, the Gateway- func-

tionality is switched off. If a value other than zero is received, the Gateway-functionality is switched on again.

The active Schedule table can be switched via CAN. For this, a CAN-Object can be specified, where the first data byte of the CAN-objects is interpreted as the number of the Schedule table. If the requested Schedule table does exists it is activated after processing of the current one.

The Gateway table consists of a total of 67 entries, with LIN-identifier, LIN-Error, Gateway on/off or Schedule table switching, a translation regulation and CAN-identifier. The following options are available for the translation regulation:

- none
  There is no translation for this LIN-message
- LIN to CAN
  If a certain LIN-message is received, its data are sent with the specified CAN-identifier as a CAN-message on the CAN-Bus
- CAN to LIN
  If the CAN-object is received with the specified CAN-identifier, its data are held in a buffer and activated on the LIN-Bus on request.

The Gateway functionality is available for both 11-bit and 29-bit identifiers.

4.3 LIN-CAN-Gateway Slave

The LIN-CAN-Gateway Slave mode allows the transparent transmission of LIN-messages to a CAN-network, where the device acts as a LIN-Slave. Received LIN-messages are transmitted to the CAN, where the LIN-identifiers are translated into CAN-identifiers. The data of this message are passed on via the CAN-Object. The device also provides a transmit function on LIN via CAN. Here CAN-identifiers are translated into LIN-identifiers and transmitted once on request by the LIN-Master in the LIN network. The data of the CAN- Object are stored in a buffer and then transmitted. A PC-configurable Gateway table is available for the translation of messages, which is stored on the device. In addition it is possible to map errors occurring on the LIN-Bus (e.g. timeout according to LIN-header) onto a CAN-message.

The Gateway functionality can be activated and deactivated via CAN. For this, a CAN-Object can be specified, where the first data byte of the CAN-objects is interpreted as a switch. If the content of this data byte is zero, the Gateway- functionality is switched off. If a value other than zero is received, the Gateway-functionality is switched on again.

The Gateway table consists of a total of 67 entries, with LIN-identifier, LIN-Error, Gateway on/off or Schedule table switching, a translation regulation and CAN-identifier. The following options are available for the translation regulation:

- none
  There is no translation for this LIN-message
- LIN to CAN
  If a certain LIN-message is received, its data are sent with the specified CAN-identifier as a CAN-message on the CAN-Bus
- CAN to LIN
  If the CAN-object is received with the specified CAN-identifier, its data are held in a buffer and activated on the LIN-Bus on request.

The Gateway functionality is available for both 11-bit and 29-bit identifiers.

4.4 LIN-Master/Slave

As LIN Master/Slave, the device works as a stand-alone LIN bus arbiter with LIN-Slave function. Here a configurable schedule list is available, which is processed cyclically. The Slave function also provides a transmit function via a configurable transmit buffer.

Up to 64 entries can be held in the Schedule list. Each entry consists of a LIN-identifier and a time waited for after transmission of the LIN-message. The waiting time can be set between 0 and 255ms.

The structure of the Transmit table is described under LIN-Slave.

4.5 LIN-Slave

The LIN-Slave mode provides a stand-alone LIN-Slave-Emulation. The device has a configurable Transmit table. When a LIN-identifier of a LIN bus master is received, the stored data are transmitted.

The Transmit table consists of 64 entries, for every possible LIN-identifier.

4.6 LIN-PC-Interface

In addition to the stand-alone modes, the LIN 2 CAN-Gateway can be operated as a LIN-PC-interface via the serial port. Here the device can be used as a LIN-Master/ Slave or as a LIN-Slave. In this mode the configuration for the stand-alone modes is also made for which volatile storage in the Flash of the device is not possible.

4.6.1 Master/Slave

As a LIN-PC-interface Master/Slave, it is possible on the PC-side to initiate the LIN bus arbiter function or switch complete LIN-messages. The whole LIN-bus traffic can be read by the PC via an additional monitoring function. In addition it is possible to switch a Wakeup-signal with a duration of (1...255ms) to the bus.

With this mode it is possible with the aid of the L2CAPI to implement a PC-based LIN-Master.

4.6.2 Slave Monitoring

In this mode the device can work as a LIN-Slave with a dynamic transmit buffer. In addition it is possible to monitor and evaluate the LIN- message traffic with the monitoring function.

4.7 Other functionalities

The L2C device supports a Power-Down mode for the stand-alone modes. (The device must not be connected to the PC with the serial cable). After a configurable time without bus traffic on LIN or CAN, the Power-Down mode is activated. The device wakes up automatically when there is communication on LIN or CAN or when the serial lead is plugged in.

The moreover it is possible to set the data length between 1 and 8 bytes for each LIN Identifier.

A name can be allocated to the device in order to differentiate better between several devices for configuration. In addition the Firmware and Hardware version can also be read out.

5 Configuration of the Operating Modes

The following section describes the configuration of the device with the aid of the Configuration Tool.

5.1 Overview

The Configuration Tool is divided into six index cards, as shown in Fig. 5.1-1:

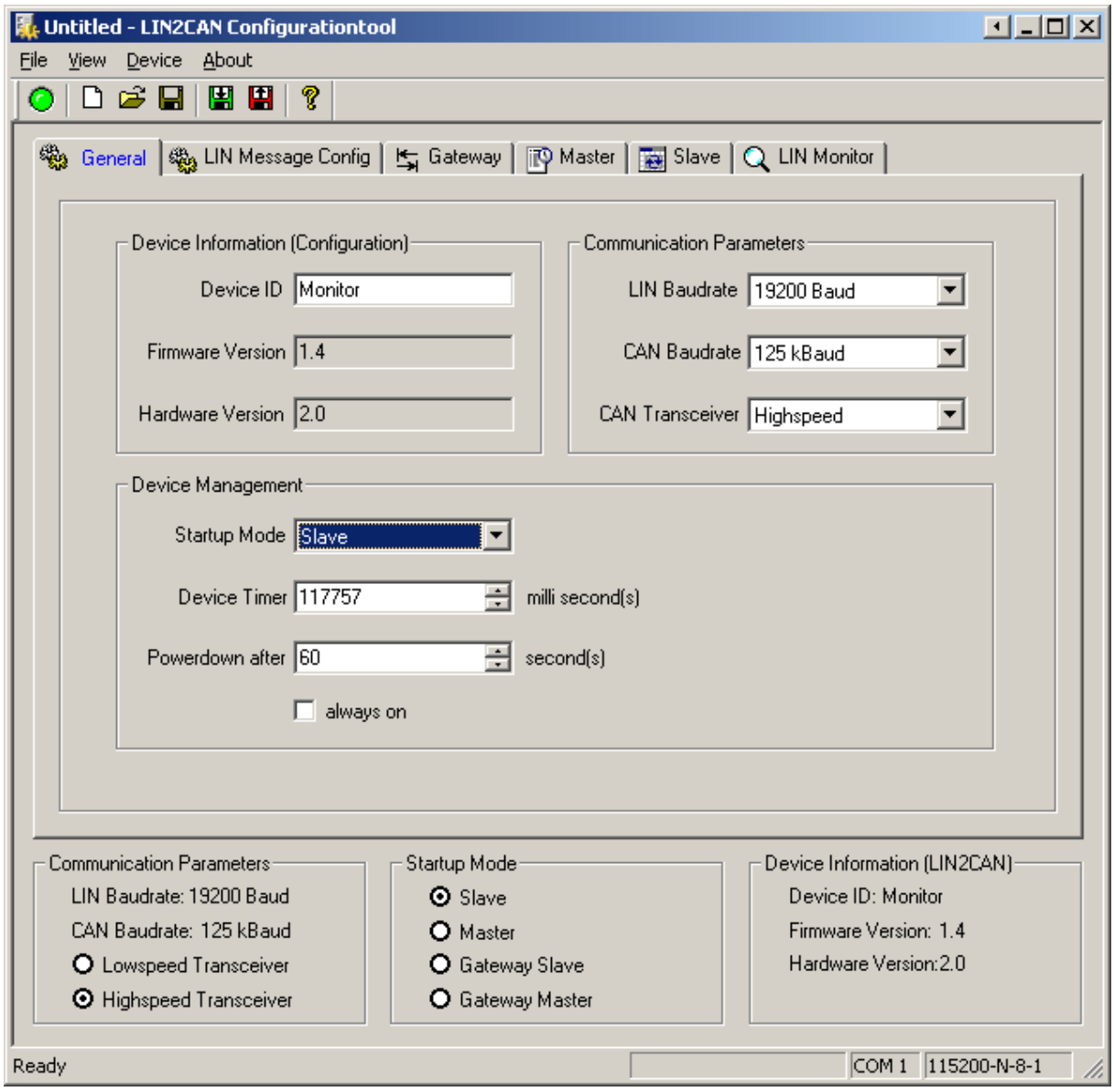


Fig. 5.1-1: Overview Configuration Tool

The General index card is used for the general configuration of the device, which refers to all operating modes. The second index card LIN Message Config is used to set the LIN data length and CRC type for each LIN Identifier. The third index card, Gateway, is used for configuration of the LIN-Gateway modes. This includes the translation table. The index card Master with the Schedule lists is used for

LIN- Master/Slave and LIN- Gateway Master/Slave modes. The Schedule lists contain the LIN-identifiers and a waiting time after transmission of the ID. The index card Slave contains the action table for the LIN-Slave, LIN-Master/Slave and LIN-PC-Interface Slave modes. These record which action is to be carried out, if the specified LIN-ID has been received. The sixth index card, Monitor, is used for monitoring the LIN-bus and manual transmission of LIN-identifiers or LIN- messages.

In order to always have the most important settings in view, the most important CAN- and LIN-Transceiver settings, the Start-up mode, the device name and the version information are displayed below the index card area (Fig. 5.1-2).

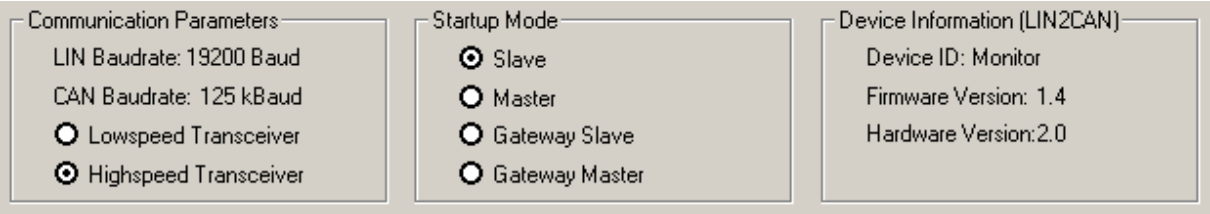


Fig. 5.1-2: Setting overview

5.2 Configuration sequence

To configure a device, the following steps must be observed:

(1) Connect LIN2CAN Gateway to the PC with serial Null modem cable.

(2) Connect device to the voltage supply.

(3) Then define COM port to which the LIN2CAN Gateway is connected. This is done via the Configuration Tool under menu point "Device → Options".

(4) Make the connection to the device in the Configuration Tool. This is done by clicking on menu point "Device → Connect" or on the red LED in the Toolbar. The Baudrate has to be set to 115200 Baud for device with hardware version 2.0 and newer, older devices communicate with 57600 Baud.

(5) Read out the configuration from the device by clicking on menu point "Device → Read Config" (optional).

(6) You can now make changes to the configuration or open an existing configuration.

(7) The configuration can be stored in the LIN2CAN Gateway again by clicking on menu point "Device -> Write Config".

(8) You can also store a configuration as an XML file in order to create a collection of configurations.

(9) The connection to the device is terminated by selecting the menu point "Device → Disconnect" or by clicking on the green LED in the Toolbar.

Note:

If a stand-alone mode was set (after Start-up), this is selected only after removing the serial cable and a brief interruption of the voltage supply of the LIN2CAN Gateways.

5.3 General - Index card

In this index card page, the general configuration is made independently of the mode. Fig. 5.3-3 shows this index card:

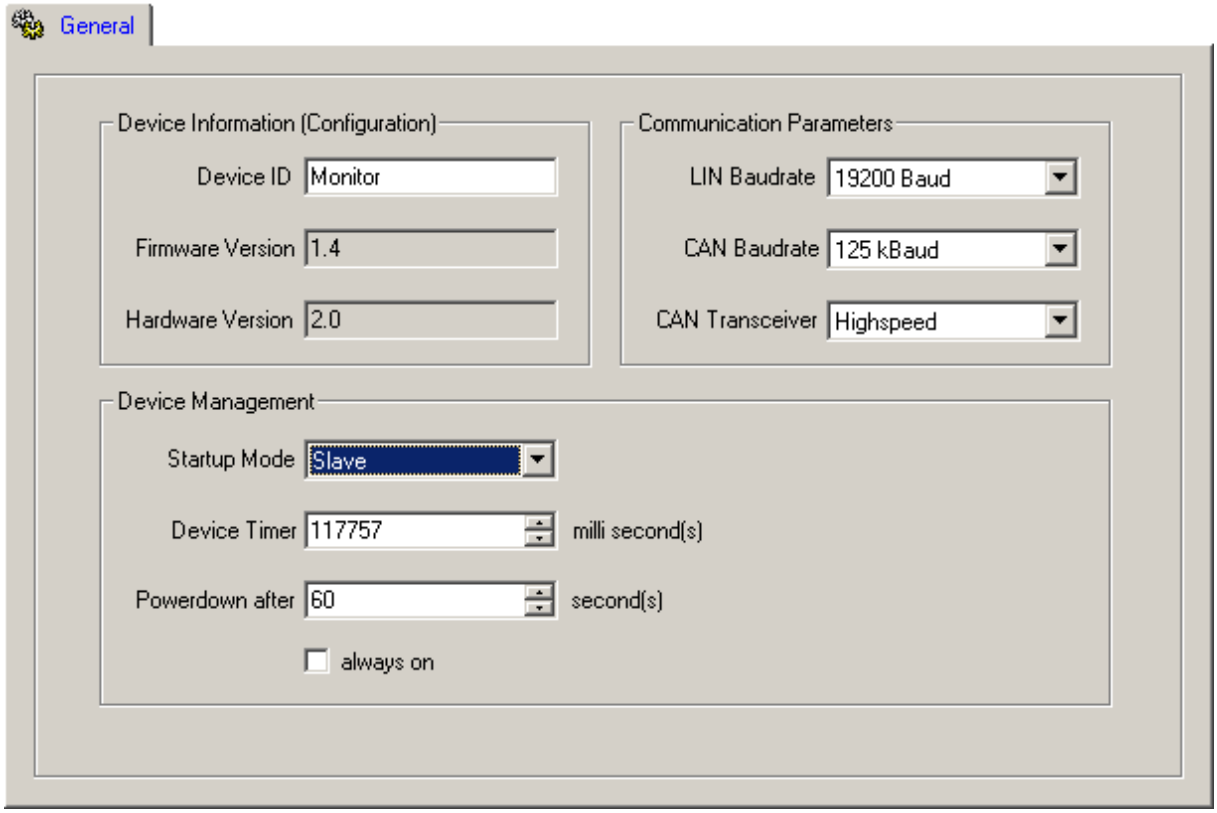


Fig. 5.3-3: Index card General

Device Information

The Device Information frame shows the device identification, a 15-figure alpha-numerical value and the Firmware and Hardware version of the connected device.

Communication Parameters

The LIN- and CAN-baud rates and the mode of the CAN-transceiver are set in the communication parameter frame via dropdown menus. For LIN baud rates of 2400 bauds, 9600 bauds and 19200 bauds are available. For CAN the following baud rates and transceiver modes can be selected:

CAN- baud rate [kBaud]	CAN- Transceiver mode
10	Lowspeed / Highspeed
20	Lowspeed / Highspeed
50	Lowspeed / Highspeed
100	Lowspeed / Highspeed
125	Lowspeed / Highspeed
250	Highspeed
500	Highspeed
1000	Highspeed

Table 5.3-1: Overview CAN baud rate and Transceiver mode

Only valid combinations of CAN-Transceiver mode and CAN- baud rate can be entered.

Device Management

In the Device Management frame you have the possibility to select the *Startup Mode* (see chapter 4), the time until the device goes into energy-saving mode (*Powerdown after*, see chapter 4.7) and the *Device Timer* (for timestamp generation of received LIN- messages) can be set.

The *Device Timer* - function is only useful in conjunction with the LIN-PC-Interface Mode (see chapter 4.6). The time set here is used as a start value for the timestamp generation of received LIN messages.

An enabled checkbox *always on* causes the LIN2CAN to stay active (Powerdown inactive).

5.4 LIN Message Config - Index card

This index card is used for assignment of the LIN data length and CRC type to the LIN Identifier. Therefore the length between 1 - 8 data bytes can be set for each LIN Identifier. The LIN Specification version 1.2 and higher say that the length information in the Identifier is optional and can be replaced through LDF definitions. In this way LIN Identifier independent data length are now possible. The CRC type can be chosen between CRC type according to LIN Specification 1.3 or 2.0. Fig. 5.4-4 shows this index card.

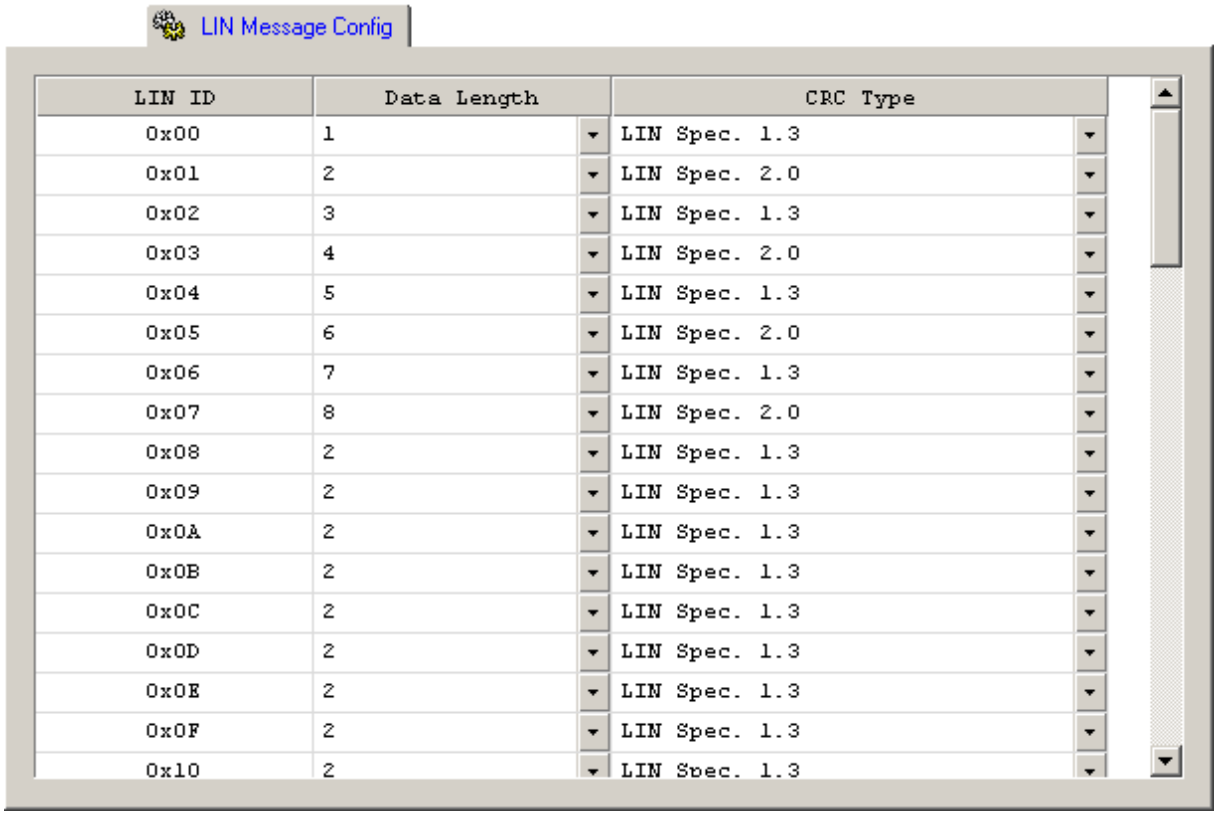


Fig. 5.4-4: Index card LIN Message Config

The standard LIN data length and the standard CRC type is set automatically if a new configuration is drawn up. Table 5.4-2 shows these standard LIN- data lengths and CRC types.

LIN Identifier	standard data lengths (Bytes)	standard CRC Type
0x00 - 0x1F	2	LIN Spec. 1.3
0x20 - 0x2F	4	LIN Spec. 1.3
0x30 - 0x3F	8	LIN Spec. 1.3

Table 5.4-2 Overview LIN Identifier – standard data length / CRC type

Note:
It has to be paid attention, that the respective LIN Identifier is configured to the same data length and CRC type for all LIN nodes in the whole LIN network. Otherwise the communication can not be assured.

5.5 Gateway - Index card

The index card Gateway is used for configuration of the Gateway mode of the device. This includes the translation table for the conversion of LIN-identifiers to CAN-identifiers and back. Fig. 5.5-5 shows this Gateway index card.

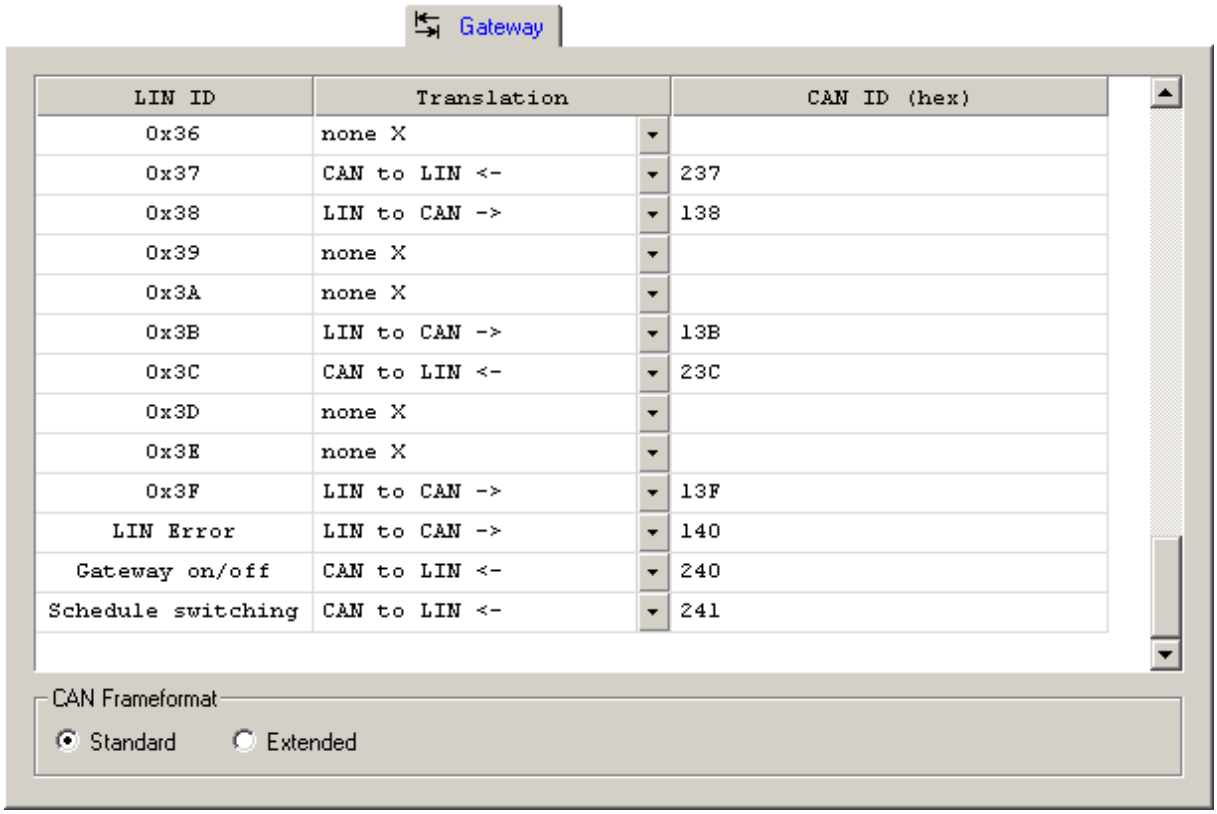


Fig. 5.5-5: Index card Gateway

A translation table consists of 64 entries for LIN-identifiers, an entry for LIN errors an entry for Gateway on/off and an entry for Schedule table switching. For the 64 LIN-identifiers the translation can be set to none, LIN to CAN or CAN to LIN. For the entry LIN errors, selection can be made between none and LIN to CAN. For the Gateway on/off and Schedule table switching entries, it is possible to choose between none and CAN to LIN.

If the translation regulation is altered, a CAN-identifier is automatically allocated as follows:

none	→	no CAN- identifier
LIN to CAN	→	0x100 + LIN-identifier
CAN to LIN	→	0x200 + LIN-identifier

where 0x40 is added in each case for LIN errors and Gateway on/off and 0x41 is added for Schedule switching. The CAN- identifier can be altered. Here it must be ensured that the identifiers correspond to the Frame format set. An "empty box" for the CAN-identifiers and multiple use of a CAN-identifier are not permitted.

The setting for the CAN-Frame format influences the whole table. Switching from Extended to Standard is only possible if there are no CAN-identifiers in the Extended-Frame format in the table.

5.6 Master - Index card

The Master index card contains the Schedule list for the LIN-Master mode. Each of the 16 lists has space for 64 entries. See Fig. 5.6-6.

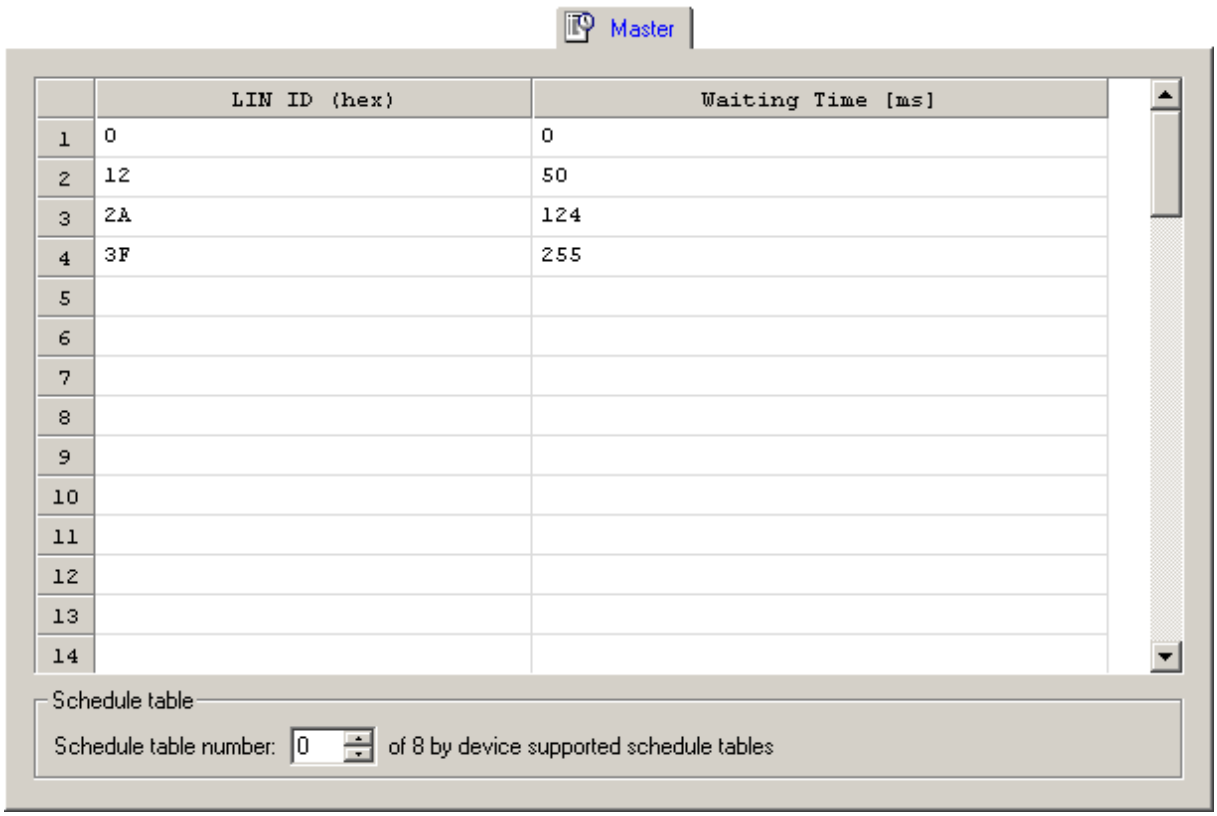


Fig. 5.6-6: Index card Master

An entry in the Schedule list consists of LIN-identifier and a time waited for after this LIN-message. This time can be between 0ms and 255ms.

An input of the waiting time is only possible after a LIN-identifier has been entered. If the LIN-identifier is cleared (by an "empty input"), the waiting time is also removed.

Up to 16 Schedule tables can be configured and saved in XML. But only the amount showed beside the Schedule table number can be stored in the device.

In the Master/Slave mode the Schedule table 0 is always active. Only in the Gateway Master/Slave mode the active Schedule table can be switched via CAN.

5.7 Slave - Index card

This index card is used for the configuration of the Action tables used in the LIN-Slave, LIN-Master/Slave and LIN-PC-Interface Slave modes. The table is only implemented once on the device. Fig. 5.7-7 shows the Slave index card with the Action table.

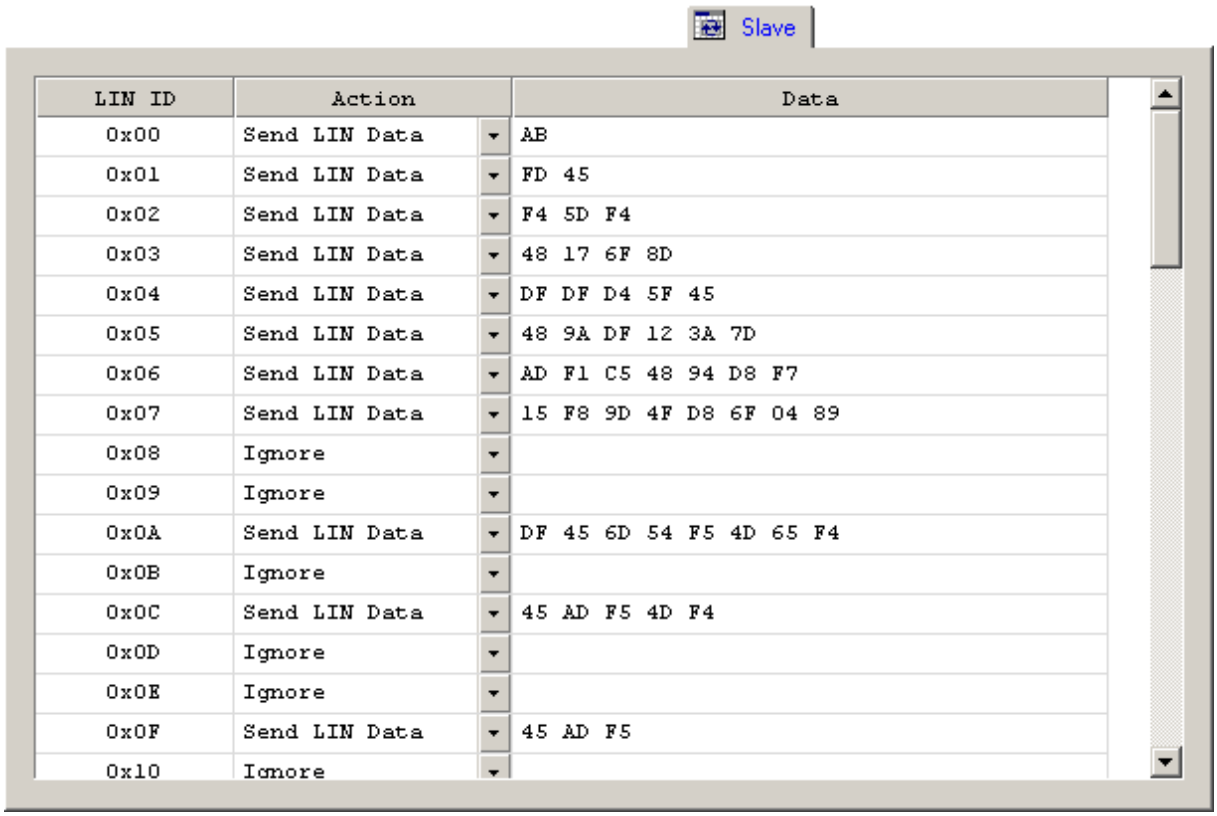


Fig. 5.7-7: Index card Slave

In the action table, one action for every LIN-identifier can be stored. It is possible to choose between the actions Ignore and "Send LIN Data".

If the action is set to Ignore, the LIN2CAN Gateway shows no reaction when the relevant identifier is received. In addition, no data can be entered in the data column.

If the action "Send LIN Data" is selected, the data field for the LIN-identifier is automatically filled up with NULL-data bytes. The data bytes are entered as a hexadecimal value. Each data byte is separated from the next by a space.

The number of data bytes depends on the setting in the LIN Message Config - Index card. If the length is modified in this index card, the data length in des Slave index card is adapted, too. By decreasing the data length data bytes will be removed beginning at the end, by increasing the data length zeros will be appended at the end.

5.8 LIN Monitor - Index card

This index card is used for monitoring the data traffic on the LIN Bus. The device is operated here in LIN-PC-Interface mode. The action table (see under Slave index card) is valid, i.e. if a LIN-ID is received behind which an action is stored, this is executed.

Furthermore LIN-identifiers, complete LIN-messages or a Wakeup signal can be transmitted. For this the mode must be set to LIN- PC- Interface Master/Slave using the Radio button. However, there must not be a second LIN-Master in the LIN-network, as this leads to collisions of the messages. The transmitted messages and errors on the LIN bus are displayed in the monitor. Fig. 5.8-8 shows this index card.

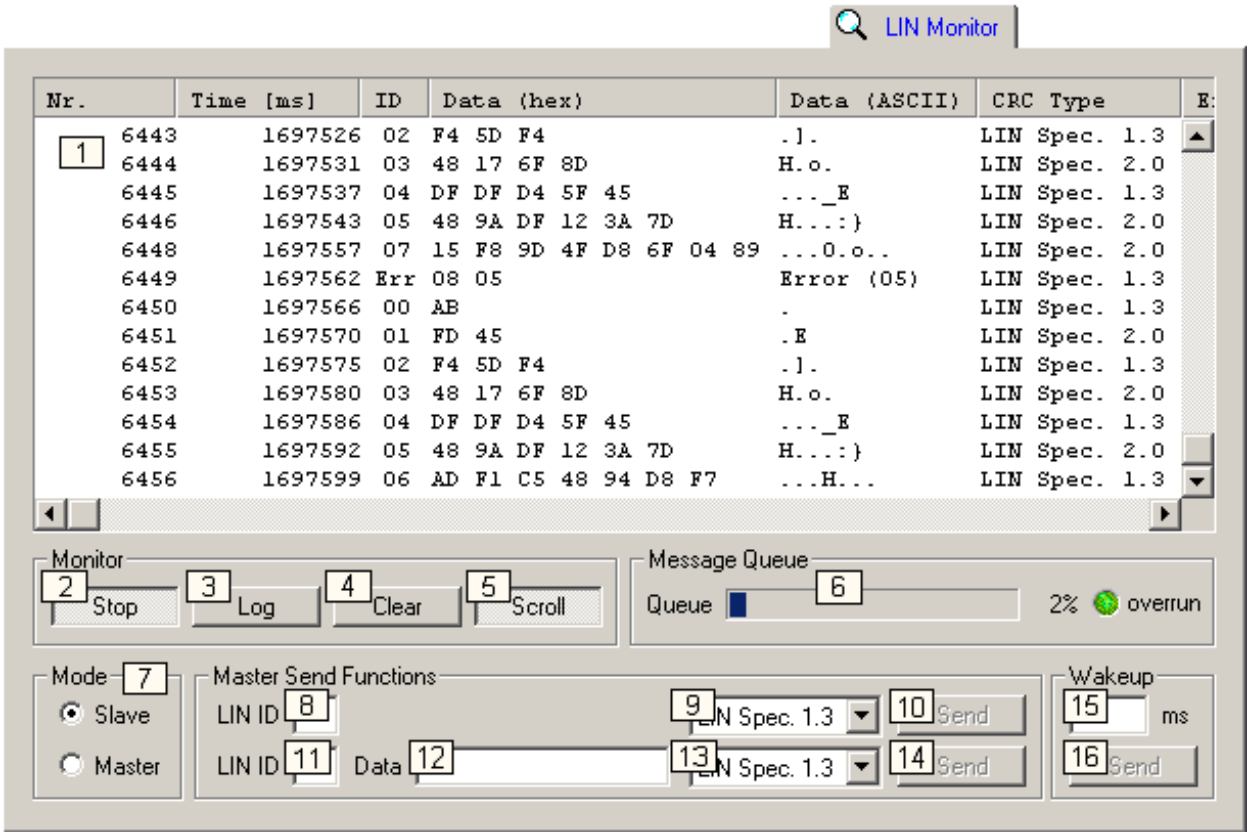


Fig. 5.8-8: Index card LIN Monitor

The monitor provides the following operating and display elements.

1) **Monitor window:** This table holds up to 3000 LIN messages, where 13 messages can be viewed simultaneously in the window. Column one contains the sequence number of the LIN message. Column two contains the timestamp (start of the device in milliseconds) at which the message was received by the device. The ID column contains the LIN-identifier or the remark "Err" for error. The Data Hex and Data ASCII columns contain the LIN data for a LIN message or the error code for a LIN-error message. Additionally, the Error Description column shows the plain text error code of an error message.
2) **Start / Stop Button:** Controls entry of the LIN-messages in the Monitor window.
3) **Log Button:** When the button is selected, a window for selection of the Log-file opens, in which the received LIN-messages are written. If the button is clicked again, logging is ended. The logging is independent of the Start / Stop button of the Monitor window, i.e. data can also be logged if updating of the Monitor window is deactivated. However, the index card must not be changed when logging as otherwise the messages are no longer logged.
4) **Clear Button:** Clears the contents of the Monitor window.
5) **Scroll Button:** Controls scrolling of the Monitor window. If this button is activated, the Monitor window scrolls automatically to the latest entry.
6) **LIN Queue Status display:** The bar and the number show the level of the Receive queue. The LED indicates an Overrun of the Queue. Green means → OK and red → Overrun.
7) **Slave / Master Radio Buttons:** The Radio buttons are used to switch between the LIN-PC-Interface Slave and LIN-PC-Interface Master/Slave mode.
8) **LIN ID Edit Box:** Used for transmitting a LIN-Identifier. (Termination of the LIN Headers). (Only possible in Master mode!)
9) **CRC Type Choice Box:** Choice box of the CRC type for the LIN-Identifier to be sent from box 8.
10) **Send Button:** Transmits the LIN-Identifier (LIN Header) from box 8 on the LIN-Bus and expects the LIN-Data with chosen CRC type from box 9.
11) **LIN ID Edit Box:** and
12) **LIN Data Edit Box:** LIN-ID and LIN Data for transmitting a LIN message. The data length can vary between 1 and 8 data bytes and is independent from the setting in the LIN Message Config - Index card.
13) **CRC Type Choice field:** Choice box of the CRC type for the LIN-Message to be sent from box 11 and 12.

14) **Send Button:** Transmits the LIN-Message from boxes 11 to 13 on the LIN bus.
15) **Wakeup Edit Box:** Used for transmitting a Wakeup signal (dominant level) on the LIN bus. The duration of the signal is entered in [ms] between 1 ms and 255 ms.
16) **Send Button:** Switches a dominant level with the duration specified in box 13 to the LIN Bus.

The elements 10, 14 and 16 are only active if the mode is switched to LIN PC-Interface Master/Slave.

5.8.1 Error overview LIN-Messages

An error is indicated by the entry "Err" in the ID column. The error is encoded in 2 data bytes, where byte one contains the associated LIN-ID and byte two the error code. An error on the LIN-bus can also generate several error messages. For each error a error description is shown in the Error Description column as plain text. See Fig. 5.8-9.

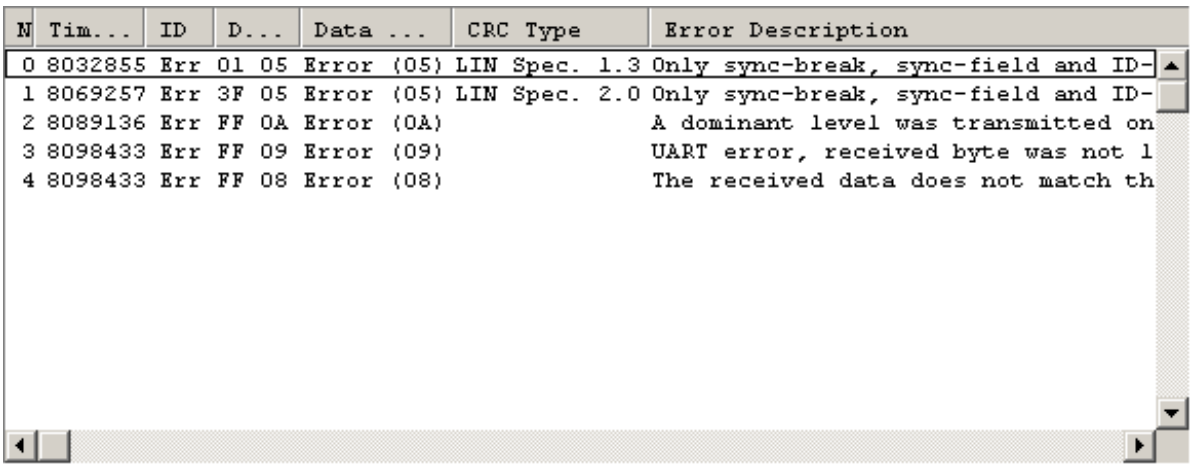


Fig. 5.8-9: LIN Monitor with error codes

Table 5.8-3 gives the error numbers and their meaning.

data byte 1	data byte 2	Description
FF	01	Received data is not a valid sync-field.
FF	02	Received data is not a valid ID-field.
FF	03	Only sync-break detected before timeout.
FF	04	Only sync-break and sync-field detected before timeout.
LIN-ID	05	Only sync-break, sync-field and ID-field detected before timeout. (**Slave does not respond!**)
LIN-ID	06	Only sync-break, sync-field, ID-field and at least one data byte detected before timeout.
LIN-ID	07	The calculated checksum does not match the received checksum. If the LIN2CAN GW is operating in LIN Spec. 2.0 mode, this error may occur due **data collision or invalid CRC mode setting**.
LIN-ID	08	The received data does not match the sent data. this error may occur due data collision in LIN Spec. 2.0 mode.
FF	09	UART error, received byte was not like UART standard ("StopBit missing" for instance).
FF	0A	A dominant level was transmitted on the LIN BUS. (**Wakeup signal**)
FF	0B	A dominant level was received on the LIN BUS, but it was to short to be a sync-break. (Note: This error code is only available in slave operation mode after at least one valid sync-break was received. If an other error code was received before, this error code has no relevance.)

Table 5.8-3: LIN error codes

5.9 Menu reference

Fig. 5.9-10 shows an overview of the menu structure of the LIN2CAN Configuration Tool.

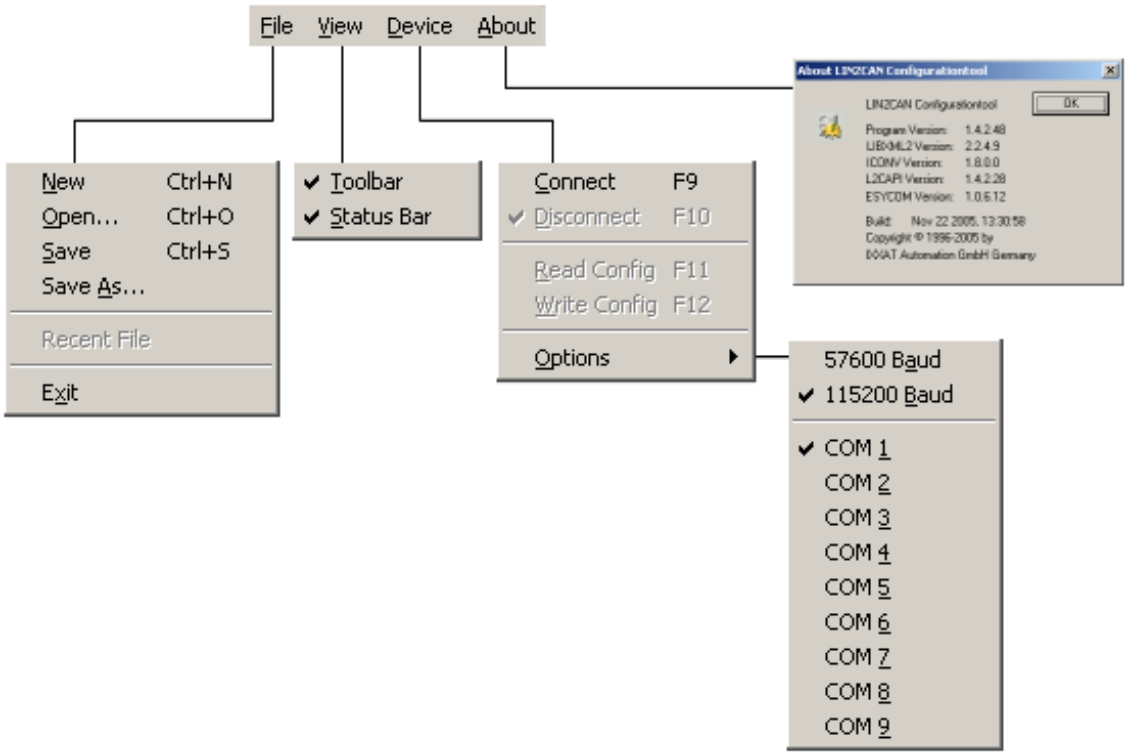


Fig. 5.9-10: Menu structure Configuration Tool

Table 5.9-4 describes the individual functions of the menu commands.

Menu point	Function
File	Opens the "File" menu
New	Resets the current configuration to the default values
Open...	Opens a dialog window in order to load an existing XML-configuration file
Save	Saves the current configuration in an XML-file
Save As	Saves the current configuration under a new name in an XML-file
"Recent File"	List of the XML configuration files opened/saved so far
Exit	Exits the program

View	Opens the "View" menu
Toolbar	Shows and hides the Toolbar
Status Bar	Shows and hides the Status Bar
Device	Opens the "Device" menu
Connect	Connects the device
Disconnect	Disconnects the device
Read Config	Reads the configuration from the connected device and displays it
Write Config	Writes the current configuration into the connected device
Options	Opens the "Options" sub-menu
57600 Baud	Defines the COM baud rate to 57600 baud
115200 Baud	Defines the COM baud rate to 115200 baud
COM 1	Defines the port to be used as COM 1
COM 2	Defines the port to be used as COM 2
…	…
COM 8	Defines the port to be used as COM 8
COM 9	Defines the port to be used as COM 9
About	Opens the "Info" dialog window

Table 5.9-4: Menu commands

5.10 Status bar

Fig. 5.10-11 shows the status bar of the LIN2CAN Configuration Tool.

Fig. 5.10-11: Status bar Configuration Tool

The status bar consists of the following display elements:

1) Menu point information. Displays information on the currently selected menu point.
2) Action information. Displays the action currently being executed (e.g. "Reading configuration...")
3) COM Port. Displays the selected COM- port.
4) COM setting. Displays the baud rate and the protocol of the COM port.

6 XML configuration file

The general structure of the XML-file corresponds to a tree structure with a Root Tag. The format is similar to the HTML-format. However, the Tags themselves must be defined. The following structure was defined for the configuration file:

```
<?xml version="1.0" standalone="yes"?>
<LIN2CAN-Gateway>
   <Config ProgramVersion="1.4.2.48">

      <!-- General config section -->
      <General>
         <Device ID="Demo XML"/>

         <VersionInformation Firmware="1.4" Hardware="2.0 "/>

         <!-- Possible Modes are: Slave, Master, Gateway Slave and Gateway Master-->
         <LIN Baudrate="19200"/>

         <!-- Possible Modes are: Gateway, Slave and Master-->
         <StartupOperation Mode="Slave"/>

         <!-- Possible Times are: 0-65535, 65535 means never Standby-->
         <Standby Time="60"/>

         <!-- LIN Message configuration table -->
         <!-- Possible LINIDs are 00 - 3F -->
         <!-- Possible Datalength are 1 - 8 -->
         <!-- Possible CRCTypes are: "LIN Spec. 1.3", "LIN Spec. 2.0"-->
         <LinMessageConfigTable>
            <Entry LINID="00" DataLength="2" CRCType="LIN Spec. 1.3"/>
            <Entry LINID="01" DataLength="2" CRCType="LIN Spec. 1.3"/>
            ...
            <Entry LINID="3e" DataLength="8" CRCType="LIN Spec. 1.3"/>
            <Entry LINID="3f" DataLength="8" CRCType="LIN Spec. 1.3"/>
         </LinMessageConfigTable>
      </General>
```

```
<!-- LIN-CAN-Gateway config section  -->
      <Gateway>
         <!-- Possible Baudrate are: 10, 20, 50, 100, 125, 250, 500, 1000-->
         <!-- Possible TransceiveModes are: Highspeed and Lowspeed-->
         <CAN Baudrate="125" TransceiverMode="Highspeed"/>

         <!-- Gateway translation table -->
         <!-- Possible LINIDs are: 00 - 3f-->
         <!-- Valid: true or false-->
         <!-- Possilbe Directions are: LIN2CAN and CAN2LIN-->
         <!-- ID29bit: true or false-->
         <!-- Possible CANIDs are: (11bit) 0 - 7ff or (29bit) 0 - 1fffffff-->

         <TranslationTable>
            <Entry LINID="00" Valid="false" Direction="LIN2CAN" ID29bit="false" CANID="00"/>
            <Entry LINID="01" Valid="false" Direction="LIN2CAN" ID29bit="false" CANID="00"/>
            ...
            <Entry LINID="3e" Valid="false" Direction="LIN2CAN" ID29bit="false" CANID="00"/>
            <Entry LINID="3f" Valid="false" Direction="LIN2CAN" ID29bit="false" CANID="00"/>

            <!-- Special  LIN-Error mapping -->
            <LINError Valid="false" ID29bit="false" CANID="00"/>

            <!-- Special  Gateway Activation mapping -->
            <ActivationViaCAN Valid="false" ID29bit="false" CANID="00"/>

            <!-- Special  Schedule table switching mapping -->
            <ScheduleSwitchingViaCAN Valid=" false " ID29bit="false" CANID="00"/>
         </TranslationTable>
      </Gateway>

      <!-- LIN-Master-Slave Emulation config section -->
      <Master>
         <!-- Schedule Table -->
         <!-- Possible Numbers are: 0 - 15-->
         <!-- Possible LINIDs are: 00 - 3f-->
         <!-- Possible WaitingTimes are: 0 - 255 [ms]-->
         <ScheduleTable Number="0">
            <Entry LINID="00" WaitingTime="2"/>
            <Entry LINID="01" WaitingTime="2"/>
            ...
            <Entry LINID="3e" WaitingTime="2"/>
            <Entry LINID="3f" WaitingTime="2"/>
         </ScheduleTable>
         ...
         <ScheduleTable Number="15">
            <Entry LINID="00" WaitingTime="2"/>
            <Entry LINID="01" WaitingTime="2"/>
            ...
            <Entry LINID="3e" WaitingTime="2"/>
            <Entry LINID="3f" WaitingTime="2"/>
         </ScheduleTable>
      </Master>
```

```
      <!-- LIN-Slave Emulation config section -->
      <Slave>
         <!-- Action Table -->
         <!-- Possible LINIDs are: 00 - 3f-->
         <!-- Possible Actions are: Ignore and SendData-->
         <!-- AdditionalInfo according to Action: Ignore: "" and SendData: LIN Data with
              matching length-->
         <ActionTable>
            <Entry LINID="00" Action="SendData" AdditionalInfo="00 00"/>
            <Entry LINID="01" Action="SendData" AdditionalInfo="00 00"/>
            ...
            <Entry LINID="3e" Action="SendData" AdditionalInfo="00 00 00 00 00 00 00 00"/>
            <Entry LINID="3f" Action="SendData" AdditionalInfo="00 00 00 00 00 00 00 00"/>
         </ActionTable>
      </Slave>
   </Config>
</LIN2CAN-Gateway>
```

Note: In the tables of the XML-configuration, only the first two and last two entries are shown in each case for space reasons!

The XML file, which is generated with the Configuration Tool, can be edited or viewed with any (XML) Editor, e.g. also with MS Internet Explorer. For better understanding of the parameters, XML configuration files are provided with inline commentaries.

The structure of the XML file is similar to the sub-division into index cards of the ConfigurationTool. There are “General”, “Gateway”, “Master” and “Slave” sections. An exception here is the CAN-Transceiver setting and the LIN message configuration table. The CAN-Transceiver setting is found in the “Gateway” section and the LIN message configuration table is found in the “General” section. The parameters from the individual index cards are contained in the XML configuration by Tag and Attribute.

The LIN message configuration table, the translation table and the Action table always have a fixed length of 64 entries (LIN ID 0x00 - 0x3f). The number of Schedule tables is fixed to 16. The number of entries in the Schedule table, on the other hand, can vary. In order that the entries do not have to be counted when this table is altered, the missing entries are filled up with empty <Entry/> Tags. This no Tags need be added or cleared. It is sufficient to adapt the Attributes of the Tags or in the case of the Schedule table to clear or add additional Attributes.

7 LIN messages Log- file

The LIN-messages are logged in an ASCII file, in which the columns are separated by a semi-colon. This file can be edited, for example, in Microsoft Excel.

The file has the following structure:

- Timestamp in milliseconds
- Timestamp in days, hours, minutes, seconds and milliseconds
- LIN identifier as hexadecimal number
- LIN identifier as decimal number
- Length of the LIN message
- LIN data as hexadecimal byte string
- LIN data as ASCII character string, where unprintable characters are replaced by a dot.
- Error description in plain text, if it is an error message

7.1 Extract of a log in tabular form

Timestamp (ms)	TimeStamp (d:h:m:s:ms)	LIN ID (Hex)	LIN ID (Dec)	Length	Data (Hex)	Data (ASCII)	Error Description
2147785100	24:20:36:25:100	20	32	4	00 00 00 00	....	
2147785108	24:20:36:25:108	30	48	8	61 62 63 64 65 22 3B 22	abcde";"	
2147785112	24:20:36:25:112	0	0	2	00 00	..	
2147785117	24:20:36:25:117	20	32	4	00 00 00 00	....	
2147785124	24:20:36:25:124	30	48	8	61 62 63 64 65 22 3B 22	abcde";"	
2147785134	24:20:36:25:134	20	32	4	00 00 00 00	....	
2147785151	24:20:36:25:151	20	32	4	00 00 00 00	....	
2147785158	24:20:36:25:158	30	48	8	61 62 63 64 65 22 3B 22	abcde";"	
2147785162	24:20:36:25:162	0	0	2	00 00	..	
2147785167	24:20:36:25:167	20	32	4	00 00 00 00	....	
2147785175	24:20:36:25:175	30	48	8	61 62 63 64 65 22 3B 22	abcde";"	
2147785179	24:20:36:25:179	0	0	2	00 00	..	
2147785184	24:20:36:25:184	20	32	4	00 00 00 00	....	
2147785192	24:20:36:25:192	30	48	8	61 62 63 64 65 22 3B 22	abcde";"	
2147785196	24:20:36:25:196	0	0	2	00 00	..	
2147785201	24:20:36:25:201	20	32	4	00 00 00 00	....	
2147785208	24:20:36:25:208	30	48	8	61 62 63 64 65 22 3B 22	abcde";"	
2147785213	24:20:36:25:213	0	0	2	00 00	..	
2147785218	24:20:36:25:218	20	32	4	00 00 00 00	....	

Fig. 7.1-1: Extract of a log in tabular form

7.2 Extract of a log as an ASCII file

```
Timestamp (ms);TimeStamp (d:h:m:s:ms);LIN ID (Hex);LIN ID (Dec);Length;Data (Hex);Data (ASCII)
;Error Description
2147785100;24:20:36:25:100;20;32;4;00 00 00 00;"....";
2147785108;24:20:36:25:108;30;48;8;61 62 63 64 65 22 3B 22;"abcde"";""";
2147785112;24:20:36:25:112;00;0;2;00 00;"..";
2147785117;24:20:36:25:117;20;32;4;00 00 00 00;"....";
2147785124;24:20:36:25:124;30;48;8;61 62 63 64 65 22 3B 22;"abcde"";""";
2147785134;24:20:36:25:134;20;32;4;00 00 00 00;"....";
2147785151;24:20:36:25:151;20;32;4;00 00 00 00;"....";
2147785158;24:20:36:25:158;30;48;8;61 62 63 64 65 22 3B 22;"abcde"";""";
2147785162;24:20:36:25:162;00;0;2;00 00;".." ;
2147785167;24:20:36:25:167;20;32;4;00 00 00 00;"...." ;
2147785175;24:20:36:25:175;30;48;8;61 62 63 64 65 22 3B 22;"abcde"";""";
2147785179;24:20:36:25:179;00;0;2;00 00;".." ;
2147785184;24:20:36:25:184;20;32;4;00 00 00 00;"....";
2147785192;24:20:36:25:192;30;48;8;61 62 63 64 65 22 3B 22;"abcde"";""";
2147785196;24:20:36:25:196;00;0;2;00 00;"..";
2147785201;24:20:36:25:201;20;32;4;00 00 00 00;"....";
2147785208;24:20:36:25:208;30;48;8;61 62 63 64 65 22 3B 22;"abcde"";""";
2147785213;24:20:36:25:213;00;0;2;00 00;"..";
2147785218;24:20:36:25:218;20;32;4;00 00 00 00;"....";
```

Note:

The data (ASCII) column represents a special feature. The data are written in high commas. Thus a ";" may appear in the character string. However, a high comma must be doubled for it to be treated as a character.

8 Technical data

Power supply:	7 V – 16 V DC, Industrial 10V - 32V DC
Power consumption:	approx. 1.5 W
Operating temperature range:	-20 °C ..+70 °C
Protection type:	IP40
Power consumption in standby mode:	max. 1 mA
Dimensions:	150 X 82 X 32 mm
Weight:	approx. 400 g

8.1 Notes on EMC

The CAN-bus connected to the interface must have a shielded lead, the braiding is to be connected with low impedance to the connector housing.

8.2 FCC Compliance

Declaration of conformity

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

1. This device may not cause harmful interference, and
2. This device must accept any interference received, including interference that may cause undesired operation

Class A digital device – instructions:

Note: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

8.3 EC Declaration of Conformity

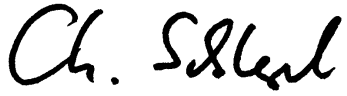
IXXAT Automation hereby declares that the product: LIN2CAN

with article numbers: 1.01.0130.00000
1.01.0130.91000

complies with the EC directives 2004/108/EC.

Applied harmonized standards in particular:
EN 55022:2006 + A1:2007
EN 61000-6-2:2005

23.08.2011, Dipl.-Ing. Christian Schlegel , Managing Director



IXXAT Automation GmbH
Leibnizstrasse 15
88250 Weingarten