Ixxat®
BY HMS NETWORKS

LIN2CAN & Configuration Tool

MANUAL
ENGLISH

HMS Connecting Devices™

HMS Technology Center Ravensburg GmbH
Helmut-Vetter-Straße 2
88213 Ravensburg
Germany

Tel.: +49 751 56146-0
Fax: +49 751 56146-29
Internet: www.hms-networks.de
E-Mail: info-ravensburg@hms-networks.de

Support

For problems or support with this product or other HMS products please request support at www.ixxat.com/support.

Further international support contacts can be found on our webpage www.ixxat.com

Copyright

Registered trademarks

Document number: 4.01.0130.20000
Version: 2.2

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. 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 be changed and imported again in order to reload the configuration onto a device.

1.3 Support

For more information on our products, FAQ lists and installation tips, please refer to the support area on our homepage (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

The LIN2CAN Gateway is a universal device for analyzing LIN networks via the CAN bus, and for emulating LIN Slave or LIN Master modules. The device can also be used as a LIN-PC-interface with PC configuration and monitoring software or for PC-compatible LIN device development (for a description of the operating modes see chapter 4, Operating Modes).

2.1 Features

- Power supply 7 to 16 V DC, Industrial 10 to 32 V 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 Pin Allocation

Fig. 2.2-1: Pin allocation

2.2.1 Supply Voltage

The device is supplied at connector X1 with a DC voltage of 7 V to 16 V (Industrial 10 V to 32 V). A prefabricated cable for the power supply is included in the scope of delivery.

Pin No. X1	Signal	Lead color
1	+	White
2	-	Brown
3	GROUND	Shield

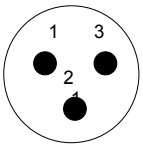


Table 2.2-1: Pin allocation Power

2.2.2 Serial Port RS232

The signals of the serial port are connected to the 9-pin Sub-D connector X2.

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

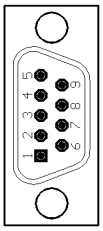


Table 2.2-2: Pin allocation RS232

1	7+8
2	3
3	2
4	6
5	5
6	4
7+8	1
Shield	Shield

Table 2.2-3: Pin Mapping Null Modem Cable

2.2.3 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	-

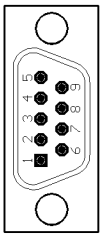


Table 2.2-4: Pin allocation CAN HS

2.2.4 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.

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

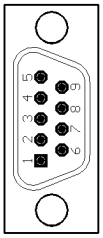


Table 2.2-5: Pin allocation CAN LS/LIN

2.3 CAN Bus Terminal

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

2.4 LIN Pull-Up Resistor

The LIN2CAN has a 1 kOhm pull-up resistor. The pull-up resistor is automatically activated in LIN Master Mode and automatically deactivated in LIN Slave Mode. External pull-up resistors are not necessary.

2.5 LED Displays

The LIN2CAN Gateway has four 2-color LEDs.

LED	Status	Meaning
PWR (Power)	Green flashing (1 Hz)	Device active
	Short red flashing	No configuration data read out of or written onto the Flash.
	Off	Device in Power Down mode
CAN	Green flash	CAN message successfully received or transmitted
	Red flash	CAN message not transmitted or overrun in CAN Receive queue
	Red	CAN Controller in bus off status
	Off	No CAN message traffic
LIN	Green flash	LIN message successfully received or transmitted
	Red flash	LIN error while receiving or transmitting a LIN message
	Off	No LIN message traffic
SER (Serial)	Green flash	Data traffic between PC and LIN2CAN Gateway
	Off	No data traffic

Table 2.4-6: 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

(1) To install the LIN2CAN Configuration Tool and the programming interface (L2CAPI), insert supplied program CD in CD drive.

(2) To start the installation program, run file *L2C_Configtool_Setup.exe*.

3.3 Connecting 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

(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 in use in menu **Device → Options**.

(6) Select baud rate according to HW version in menu **Device → Options**:
HW version 1.x ➔ baud rate = 57600
HW version 2.x ➔ baud rate = 115200

(7) To establish a connection to the device, click on LED in toolbar or select menu **Device → Connect**.

4 Operating Modes

The LIN2CAN Gateway supports various operating modes:

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

In every operating mode the LIN2CAN can be operated in Master/Slave or in Slave mode.

All operating modes are configured via the PC. LIN and LIN-CAN-Gateway operating modes are configured via the PC as startup mode. If the device is **not** connected to a PC during power-on, it starts in the configured startup mode and works in standalone mode. If the device is connected by a serial cable to a PC during power-on, it starts in LIN-PC-Interface Slave mode.

4.1 LIN-CAN-Gateway

The LIN-CAN-Gateway mode allows operating as LIN-CAN-Gateway Master/Slave or Slave and therefore the transparent transmission of LIN messages to a CAN network.

Received LIN messages are transmitted to CAN. LIN identifiers are translated into CAN identifiers and the data of the 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 a LIN Master (internal schedule table or external master) in the LIN network. The data of the CAN object are stored in a buffer and then transmitted as LIN message. 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. A CAN object can be specified, whose first data byte 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. For further information see chapter Tab Gateway, p. 20.

4.1.1 LIN-CAN-Gateway Master/Slave

The device acts as a LIN Master/Slave, which means as a LIN bus arbiter with LIN Slave function. Several configurable schedule lists are available, whereas only one is processed cyclically. For further information about the schedule table see chapter Tab Master, p. 22.

The active schedule table can be switched via CAN. For this, a CAN object can be specified, whose first data byte 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.

4.1.2 LIN-CAN-Gateway Slave

The LIN-CAN-Gateway Slave mode allows the transparent transmission of LIN messages to a CAN network. The device acts as a LIN-Slave. When a configured LIN header is received, the device executes the corresponding action defined in the action table (for further information see chapter Tab Slave, p. 23).

4.2 LIN

LIN operating is possible as LIN-Master/Slave or LIN Slave.

4.2.1 LIN Master/Slave

As LIN Master/Slave, the device works as a standalone LIN bus arbiter with LIN Slave function. A configurable schedule list is processed cyclically. The Slave function provides a transmit function via a configurable transmit buffer. For further information about the schedule list see chapter Tab Master, p.22.

4.2.2 LIN Slave

The LIN Slave mode provides a standalone 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 (for further information see chapter Tab Slave, p. 23).

4.3 LIN-PC-Interface

The LIN2CAN Gateway can be operated as LIN-PC-Interface via the serial port, either as a LIN Master/Slave or as a LIN Slave.

4.3.1 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 (duration 1...255 ms) to the bus.

With this mode it is possible with the aid of the L2CAPI to implement a PC-based LIN Master. THE L2CAPI has no access to CAN.

4.3.2 Slave Monitoring

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

4.4 Power Down mode

The LIN2CAN supports a Power Down mode for the standalone modes. (The device must not be connected to the PC via a 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.

5 Configuration of the Operating Modes

The following section describes the configuration of the device with the LIN2CAN Configuration Tool.

5.1 Overview

The Configuration Tool is divided into six tabs.

The most important settings, like CAN and LIN transceiver settings, start-up mode, device name and version information are displayed below the tab area.

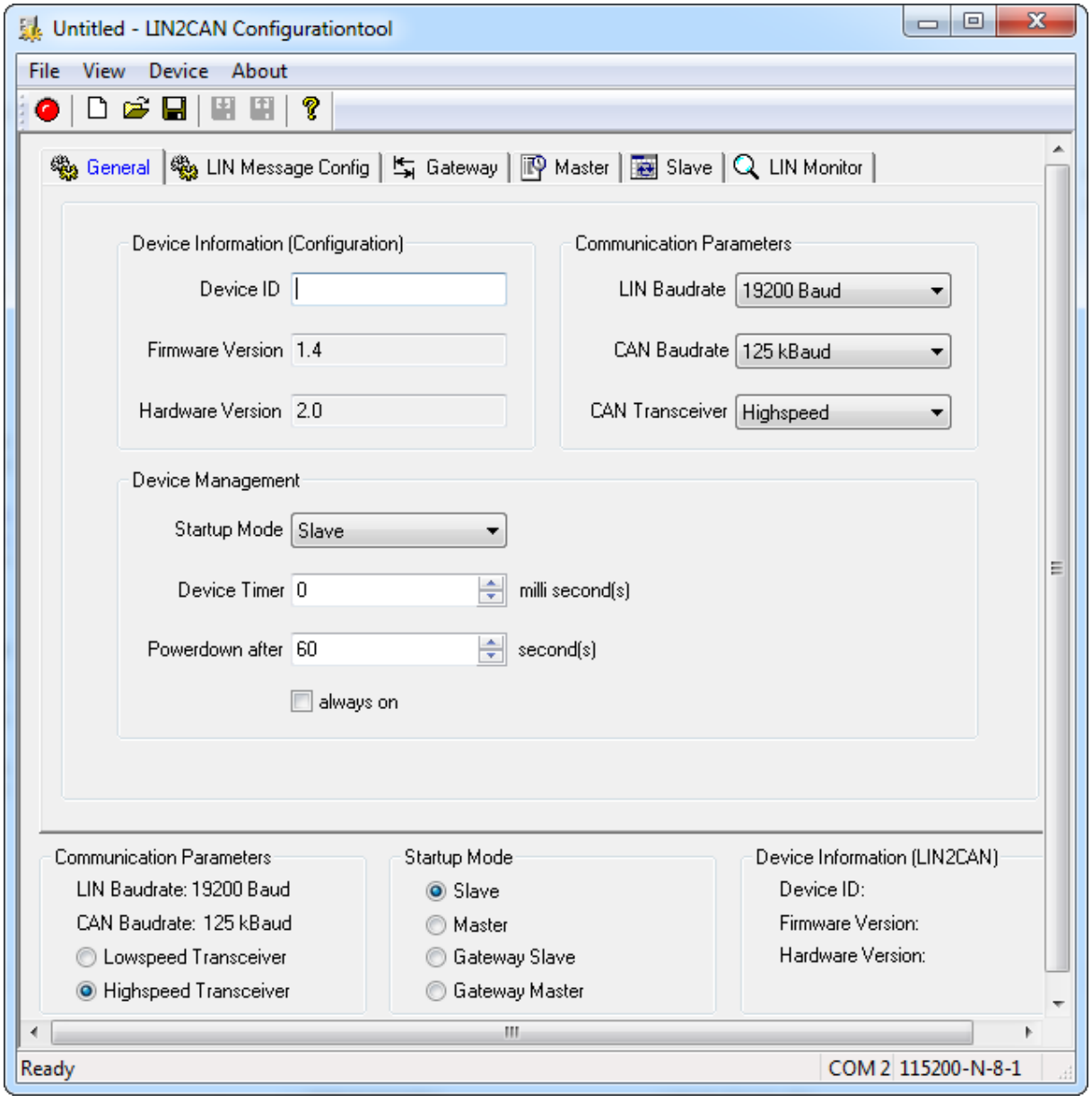


Fig. 5.1-1: Overview Configuration Tool

5.2 Configuring the Device

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

(2) Connect device to voltage supply.

(3) In the Configuration Tool in menu **Device → Options** define COM port to which the LIN2CAN Gateway is connected. Set baudrate for devices with hardware version 2.0 and newer to 115200 Baud, for older devices set to 57600 Baud.

(4) To establish a connection to the device, click on red LED in toolbar or select menu **Device → Connect**.

(5) Select menu **Device → Read Config** to read configuration from the device (optional).

(6) It is now possible to change the configuration or to open an existing configuration.

(7) To store the configuration in the LIN2CAN Gateway, select menu **Device → Write Config**.

(8) The configuration can also be stored as XML.

(9) To terminate connection to the device, click on green LED in toolbar or select menu **Device → Disconnect**.

Note:

If standalone mode is set (after start-up), the operating mode is active only after removing the serial cable and a brief interruption of the voltage supply of the LIN2CAN Gateways.

5.3 Tab General

The tab **General** shows the general configuration independently of the mode.

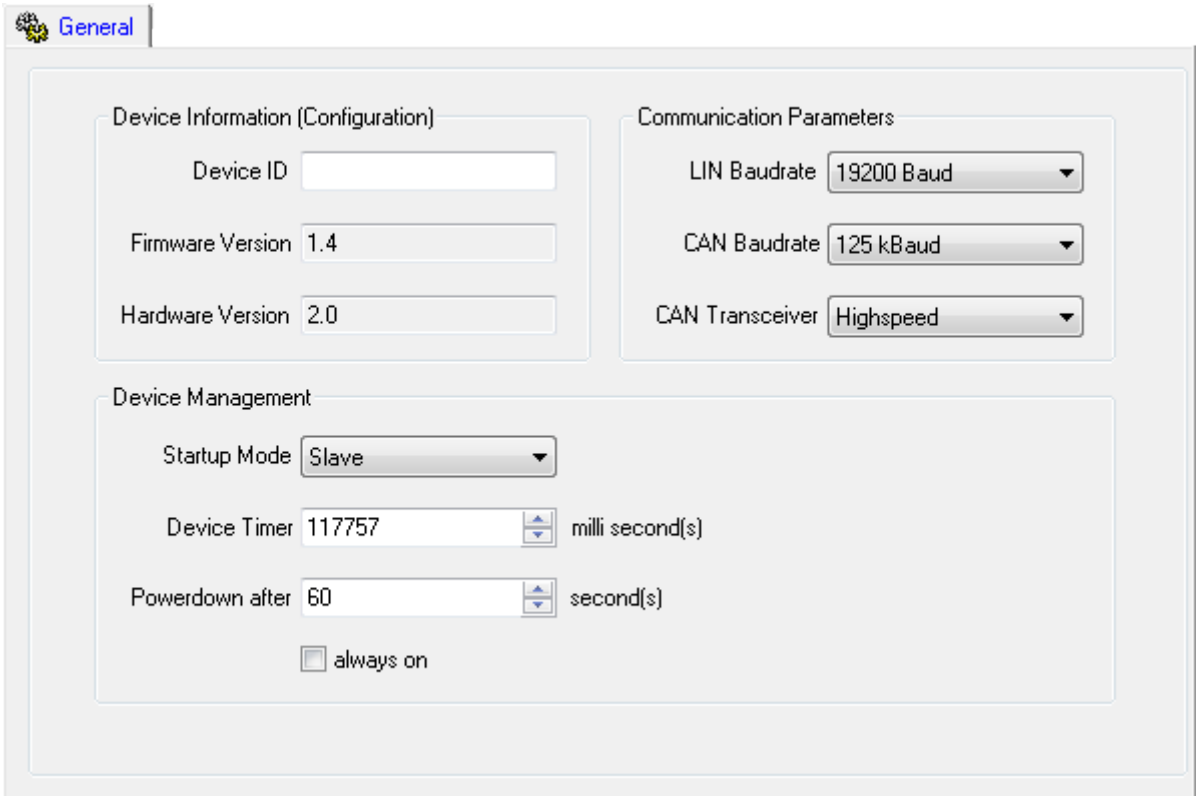


Fig. 5.3-2: Tab General

Device Information

The **Device Information** frame shows the device identification, a 15-figure alphanumerical 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 baudrate [kBaud]	CAN transceiver mode
10	Low-speed/high-speed
20	Low-speed/high-speed
50	Low-speed/high-speed
100	Low-speed/high-speed
125	Low-speed/high-speed
250	high-speed
500	high-speed
1000	high-speed

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 (*Power down 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.

With enabled checkbox *always on* the LIN2CAN stays active and does not go in energy saving mode (power down inactive).

5.4 Tab LIN Message Config

This tab is used for assignment of the LIN data length and CRC type to the LIN identifier. The length between 1-8 data bytes can be set for each LIN identifier. The LIN specification version 1.2 and higher regards the length information in the identifier as optional which can be replaced by LDF definitions. In this way LIN identifier independent data length are possible. The CRC type can be chosen between CRC type according to LIN specification 1.3 or 2.0.

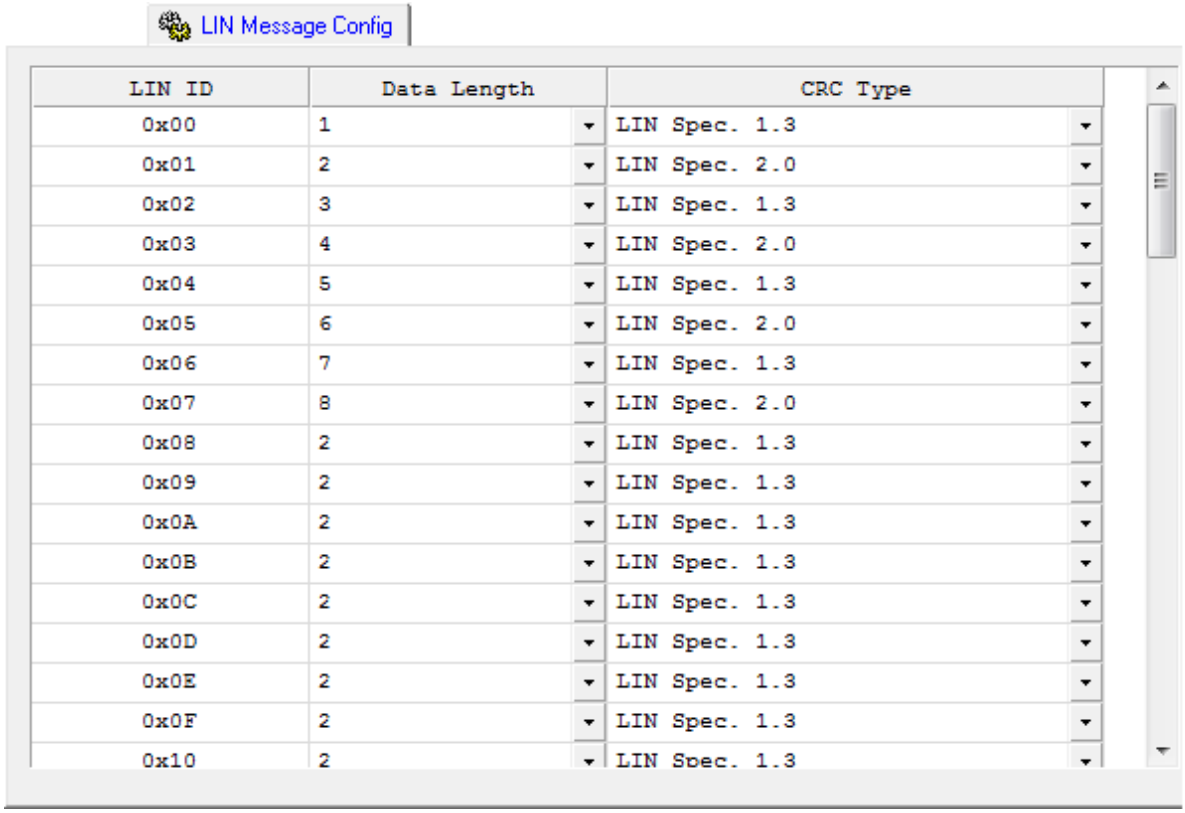


Fig. 5.4-3: Tab LIN Message Config

The standard LIN data length and the standard CRC type is set automatically while creating a new configuration.

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 and CRC types

Note:
Make sure, 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 cannot be assured.

5.5 Tab Gateway

The tab **Gateway** is used for configuration of the Gateway mode of the device and includes the translation table for the conversion of LIN identifiers to CAN identifiers and back.

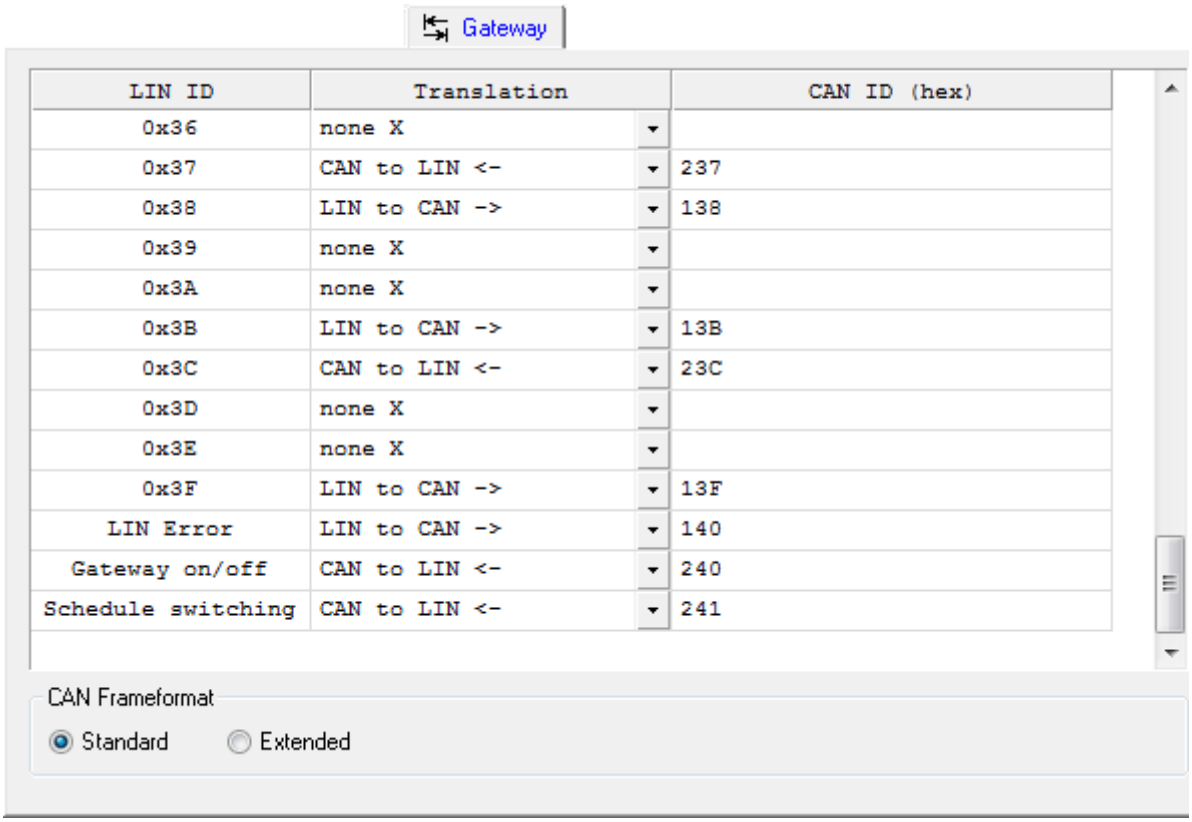


Fig. 5.5-4: Tab Gateway

A translation table consists of the following entries and translation possibilities:

- 64 entries for LIN identifiers
  Translation: **none**, **LIN to CAN** or **CAN to LIN**
- 1 entry for LIN errors
  Translation: **none** or **LIN to CAN**
- 1 entry for gateway on/off
  Translation: **none** or **CAN to LIN**
- 1 entry for schedule table switching
  Translation: **none** or **CAN to LIN**

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

none → no CAN identifier

LIN to CAN → 0x100 + LIN identifier

CAN to LIN → 0x200 + LIN identifier

For LIN errors and gateway on/off 0x40 is added in each case, for schedule switching 0x41. The CAN identifier can be altered. Make sure, that the identifiers correspond to the frame format set. An empty box for the CAN identifiers or 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 Tab Master

The tab **Master** contains the schedule list for the LIN Master/Slave and the LIN Gateway Master/Slave mode. Each of the 16 lists has space for 64 entries.

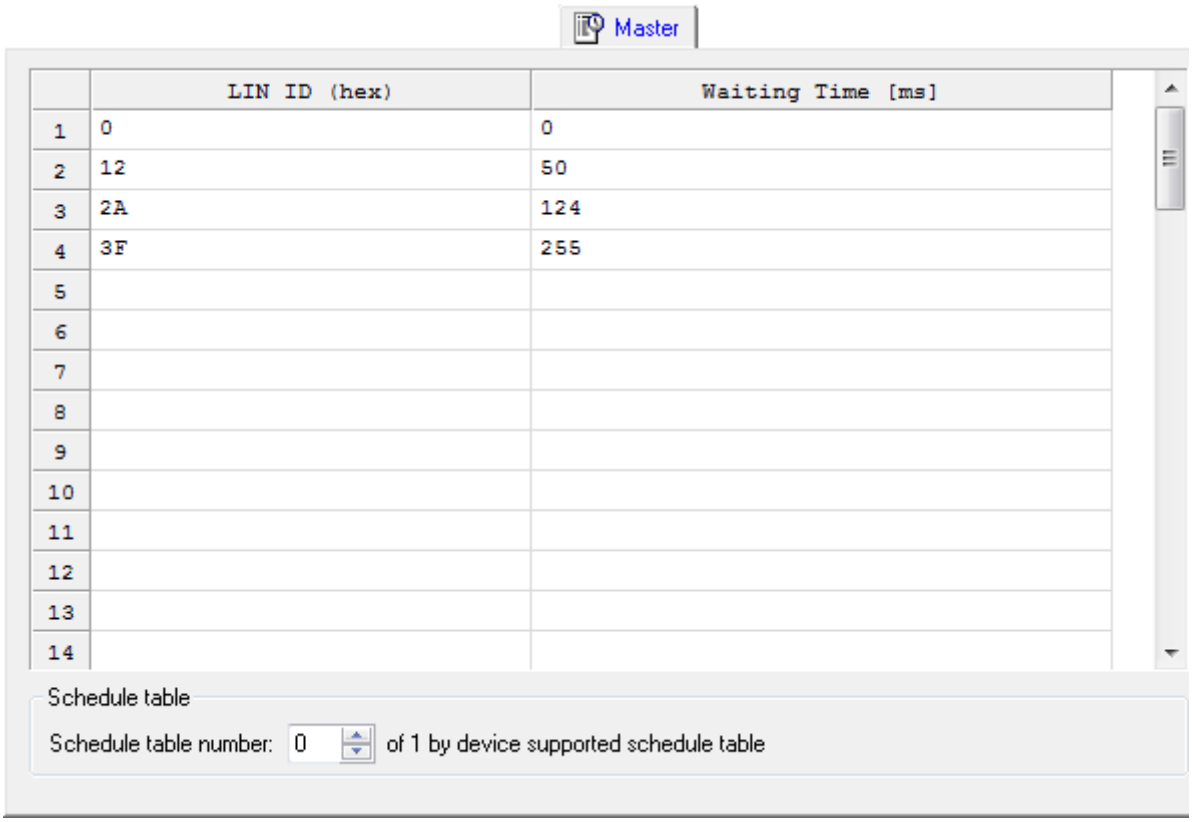


Fig. 5.6-5: Tab Master

In the schedule list LIN identifier and the waiting time after transmission of this LIN message (0 to 255 ms) is entered. The transmit table consists of 64 entries, one for each possible LIN identifier.

Input of the waiting time is only possible if a LIN identifier is entered. If the LIN identifier is removed (via an empty input), the waiting time is also removed.

Up to 16 schedule tables can be configured and saved in XML. Only the number 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 Tab Slave

The tab **Slave** is used for the configuration of the action tables in the LIN Slave, LIN Master/Slave and LIN PC Interface Slave modes. The table is only implemented once on the device and determines which action is executed when the configured LIN header is received.

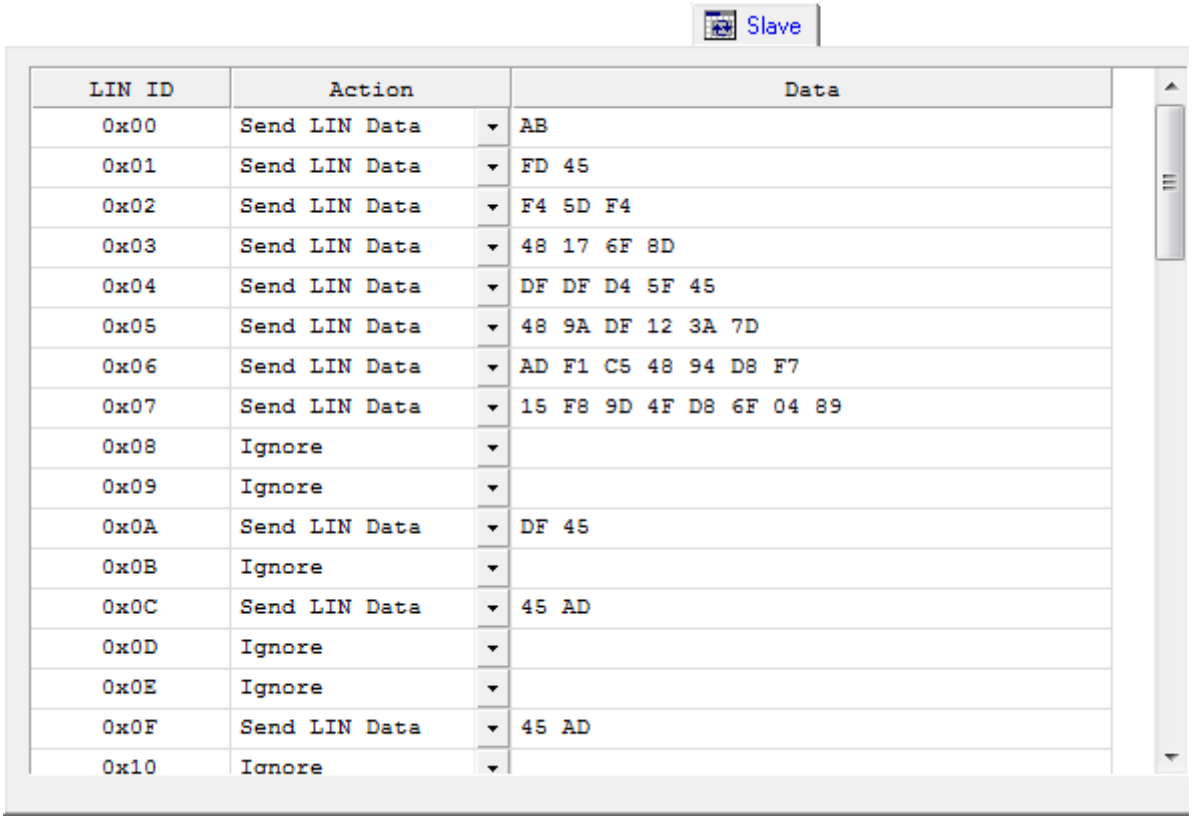


Fig. 5.7-6: Tab Slave with action table

In the action table for every LIN identifier either the action **Ignore** or **Send LIN Data** can be stored.

With **Ignore**, the LIN2CAN Gateway shows no reaction when receiving the relevant identifier. In column **Data** no data can be entered.

With **Send LIN Data** the data field for the LIN identifier is automatically filled with NULL data bytes. The data bytes are entered as hexadecimal value. The data bytes are separated by spaces.

The number of data bytes depends on the setting in tab **LIN Message Config**. If the length is modified in this tab, the data length in tab **Slave** is adapted, too. In case of decreasing the data length the last data bytes are removed, in case of increasing the data length zeros are added at the end.

5.8 Tab LIN Monitor

The tab **LIN Monitor** is used for monitoring the data traffic on the LIN bus in the LIN PC Interface mode. The action table (see in tab **Slave**) is valid, i.e. if a LIN ID with an assigned action is received, this action is executed.

LIN identifiers, complete LIN messages or a wakeup signal can be transmitted. The mode must be set to LIN PC Interface Master/Slave using the Radio button. 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.

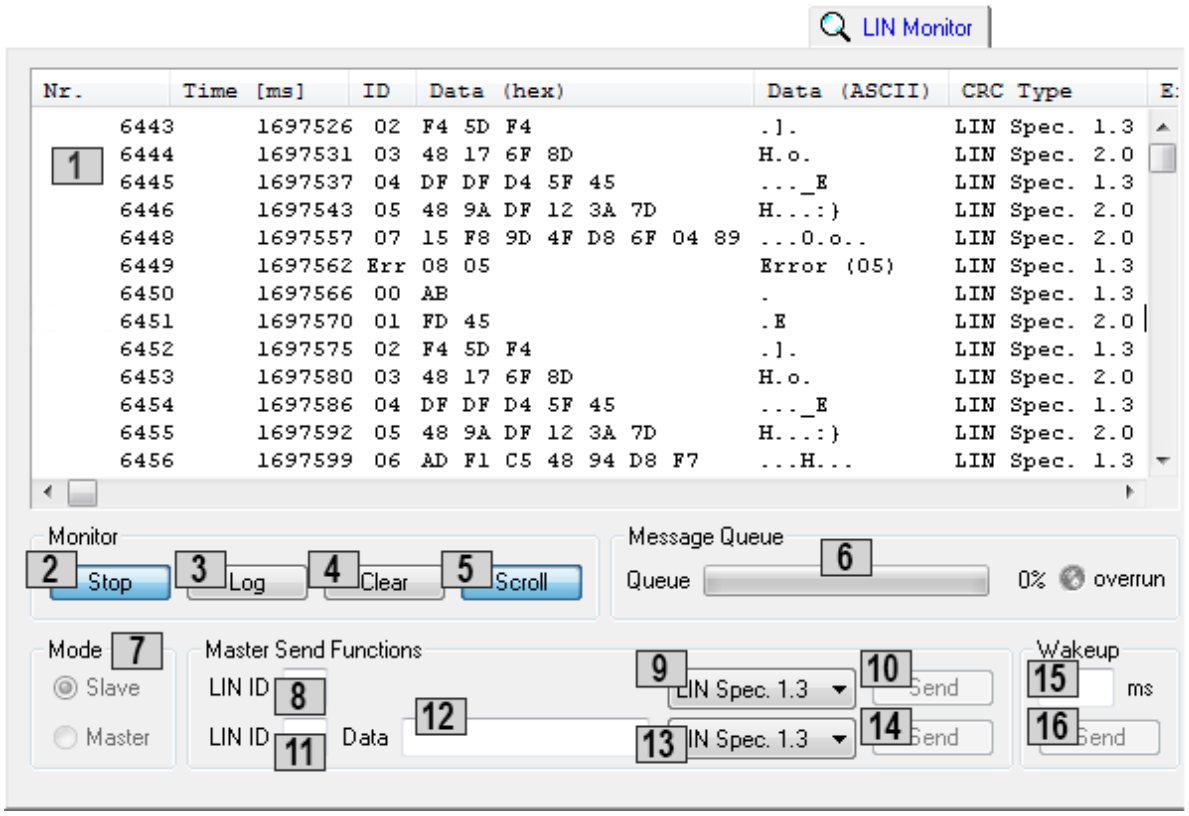


Fig. 5.8-7: Tab LIN Monitor

The monitor provides the following operating and display elements:

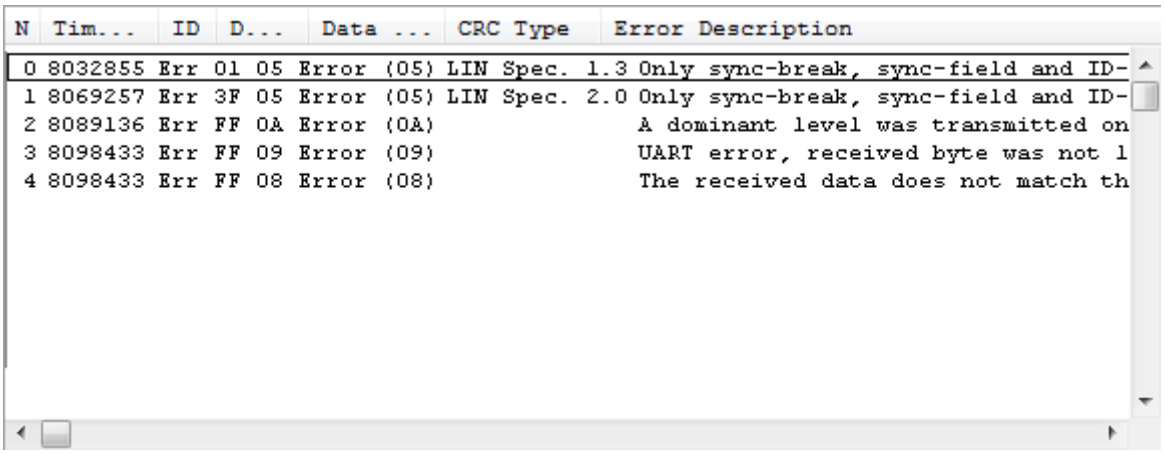
1) **Monitor window:** Table holds up to 3000 LIN messages.13 messages can be displayed simultaneously in the window.
   - Column **Nr.**: sequence number of the LIN message
   - Column **Time (ms)**: receiving timestamp (start of device in milliseconds)
   - Column **ID:** LIN identifier or remark **Err** for error
   - Columns **Data Hex/Data ASCII:** LIN data of LIN message or error code for LIN error message
   - Column **Error Description**: plain text of an error message

2) **Start/Stop Button:** Controls entry of the LIN messages in the monitor window.
3) **Log Button:** Opens the window for selection of the Log file, in which the received LIN messages are written. Clicking again ends the logging. 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. Logging is only active if the tab is active.
4) **Clear Button:** Clears content of monitor window.
5) **Scroll Button:** Controls scrolling of the monitor window. If button is activated, the monitor window scrolls automatically to the latest entry.
6) **LIN Queue Status display:** bar and number show level of receive queue. LED indicates queue state: green → OK / red → overrun.
7) **Slave/Master Radio Buttons:** Switching between LIN PC Interface Slave and LIN PC Interface Master/Slave mode.
8) **LIN ID Edit Box:** LIN identifier to be transmitted (termination of the LIN Headers, only possible in Master mode)
9) **CRC Type Choice Box:** Select 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. Data length can vary between 1 and 8 data bytes and is independent from setting in tab **LIN Message Config**.
13) **CRC Type Choice field:** Select 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:** Duration of wakeup signal (dominant level) on the LIN bus, between 1 ms and 255 ms.
16) **Send Button:** Switches a dominant level with the duration specified in box 13 to the LIN bus.

Elements 10, 14 and 16 are only active in operating mode LIN PC Interface Master/Slave.

5.8.1 Error Overview LIN Messages

An error is indicated by **Err** in the ID column. The error is encoded in 2 data bytes: byte 1 contains the associated LIN ID, byte two the error code. An error on the LIN bus can also generate several error messages. For each error an error description is shown in the **error description** column as plain text.



N	Tim...	ID	D...	Data ...	CRC Type	Error Description
0	8032855	Err	01	05 Error (05)	LIN Spec. 1.3	Only sync-break, sync-field and ID-
1	8069257	Err	3F	05 Error (05)	LIN Spec. 2.0	Only sync-break, sync-field and ID-
2	8089136	Err	FF	0A Error (0A)		A dominant level was transmitted on
3	8098433	Err	FF	09 Error (09)		UART error, received byte was not l
4	8098433	Err	FF	08 Error (08)		The received data does not match th

Fig. 5.8-8: LIN monitor with error codes

Error Numbers and Meaning

Data byte 1	Data byte 2	Description
FF	01	Received data are no valid sync field.
FF	02	Received data are no 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	Calculated checksum does not match received checksum. If the LIN2CAN GW is operating in LIN Spec. 2.0 mode, the error may occur due to **data collision or invalid CRC mode setting**.
LIN ID	08	Received data does not match sent data. Error may occur due to data collision in LIN Spec. 2.0 mode.
FF	09	UART error, received byte was not like UART standard (e.g. StopBit missing).
FF	0A	Dominant level transmitted on the LIN BUS (**Wakeup signal**)
FF	0B	Dominant level received on the LIN BUS, but it is too short to be a sync-break. (Note: Error code is only available in slave operation mode after at least one valid sync-break was received. If another error code was received before, this error code has no relevance.)

Table 5.8-3: LIN error codes

5.9 Menu Reference

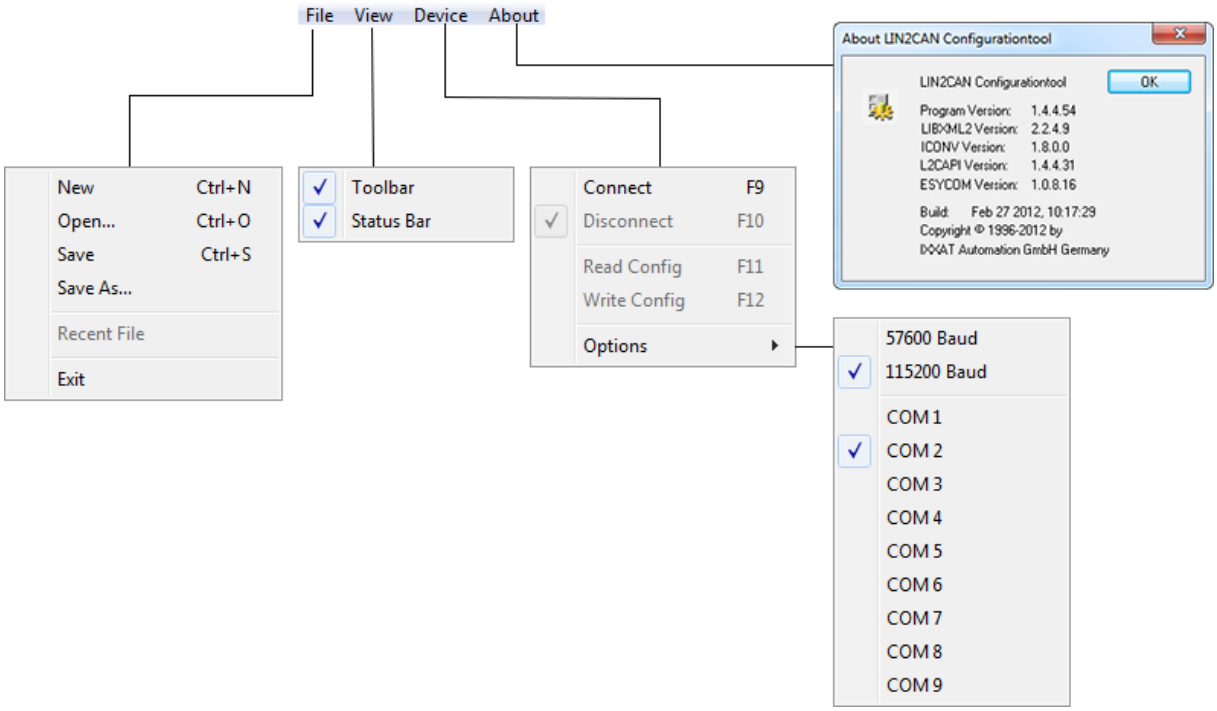


Fig. 5.9-9: Overview of menu structure Configuration Tool

Functions of the Menu Commands

Menu point	Function
File	Opens menu **File**
New	Resets current configuration to 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 so far opened/saved XML configuration files
Exit	Exits the program
View	Opens menu **View**
Toolbar	Shows and hides the toolbar
Status Bar	Shows and hides the status bar

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

Table 5.9-4: Menu commands

5.10 Status Bar

Fig. 5.10-10: Status bar LIN2CAN Configuration Tool

The status bar consists of the following display elements:

1) Menu information: displays information on the currently selected menu.
2) Action information: displays the currently executed action (e.g. Reading configuration...)
3) COM port: displays the selected COM port.
4) COM setting: displays baud rate and protocol of 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. But the tags must be defined. The following structure is 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 reasons of space!

The XML file, which is generated with the Configuration Tool, can be edited or viewed with any (XML) editor (e.g. 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 division into tabs of the Configuration Tool. There are General, Gateway, Master and Slave sections. Exceptions are the CAN transceiver setting and the LIN message configuration table. The CAN transceiver setting is in the Gateway section and the LIN message configuration table is in the General section. The parameters from the individual tabs 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 can vary. To avoid that entries have to be counted when this table is altered, the missing entries are filled with empty <Entry/> tags. No tags need be added or removed. It is sufficient to adapt the attributes of the tags or in 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 semicolon. 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, 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. A high comma must be doubled to be treated as a character.

8 Technical Data

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

LIN messages according to specification V1.3, V2.0 and V2.1 are supported.

8.1 Notes on EMC

The CAN bus connected to the interface must have a shielded lead, the braiding must 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

The product is in compliance with the Electromagnetic Compatibility Directive. More information and the Declaration of Conformity is found at www.ixxat.com.