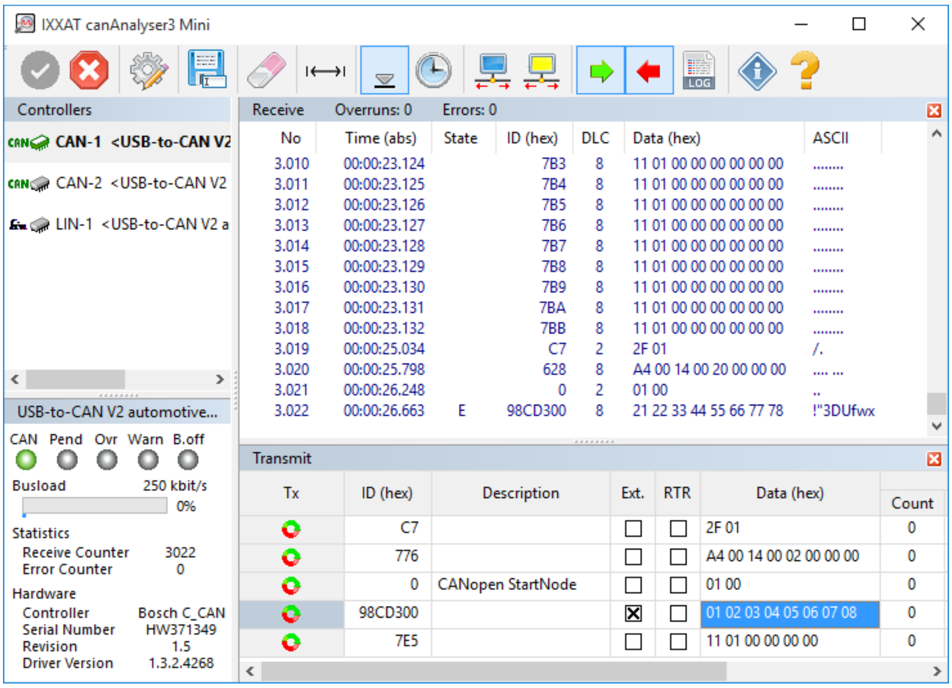
IXXAT canAnalyser3 Mini

Busmonitoringtool for Windows



IXXAT canAnalyser3 Mini

Controllers

- CAN-1 <USB-to-CAN V2
- CAN-2 <USB-to-CAN V2
- LIN-1 <USB-to-CAN V2 a

USB-to-CAN V2 automotive...

CAN Pend Ovr Warn B.off

Busload 250 kbit/s 0%

Statistics

Receive Counter	3022
Error Counter	0

Hardware

Controller	Bosch C_CAN
Serial Number	HW371349
Revision	1.5
Driver Version	1.3.2.4268

Receive Overruns: 0 Errors: 0

No	Time (abs)	State	ID (hex)	DLC	Data (hex)	ASCII
3.010	00:00:23.124		7B3	8	11 01 00 00 00 00 00 00	....
3.011	00:00:23.125		7B4	8	11 01 00 00 00 00 00 00	....
3.012	00:00:23.126		7B5	8	11 01 00 00 00 00 00 00	....
3.013	00:00:23.127		7B6	8	11 01 00 00 00 00 00 00	....
3.014	00:00:23.128		7B7	8	11 01 00 00 00 00 00 00	....
3.015	00:00:23.129		7B8	8	11 01 00 00 00 00 00 00	....
3.016	00:00:23.130		7B9	8	11 01 00 00 00 00 00 00	....
3.017	00:00:23.131		7BA	8	11 01 00 00 00 00 00 00	....
3.018	00:00:23.132		7BB	8	11 01 00 00 00 00 00 00	....
3.019	00:00:25.034		C7	2	2F 01	/.
3.020	00:00:25.798		628	8	A4 00 14 00 20 00 00 00	....
3.021	00:00:26.248		0	2	01 00	..
3.022	00:00:26.663	E	98CD300	8	21 22 33 44 55 66 77 78	!"3DUfwx

Transmit

Tx	ID (hex)	Description	Ext.	RTR	Data (hex)	Count
	C7		□	□	2F 01	0
	776		□	□	A4 00 14 00 02 00 00 00	0
	0	CANopen StartNode	□	□	01 00	0
	98CD300		☒	□	01 02 03 04 05 06 07 08	0
	7E5		□	□	11 01 00 00 00 00	0

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

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

Fax: +49 751 56146-29
E-Mail: support@ixxat.de

Further international support contacts can be found on our webpage www.hms-networks.de

Copyright

Registered trademarks

Document number: 4.02.0250.20013
Version: E-3.01

Contents

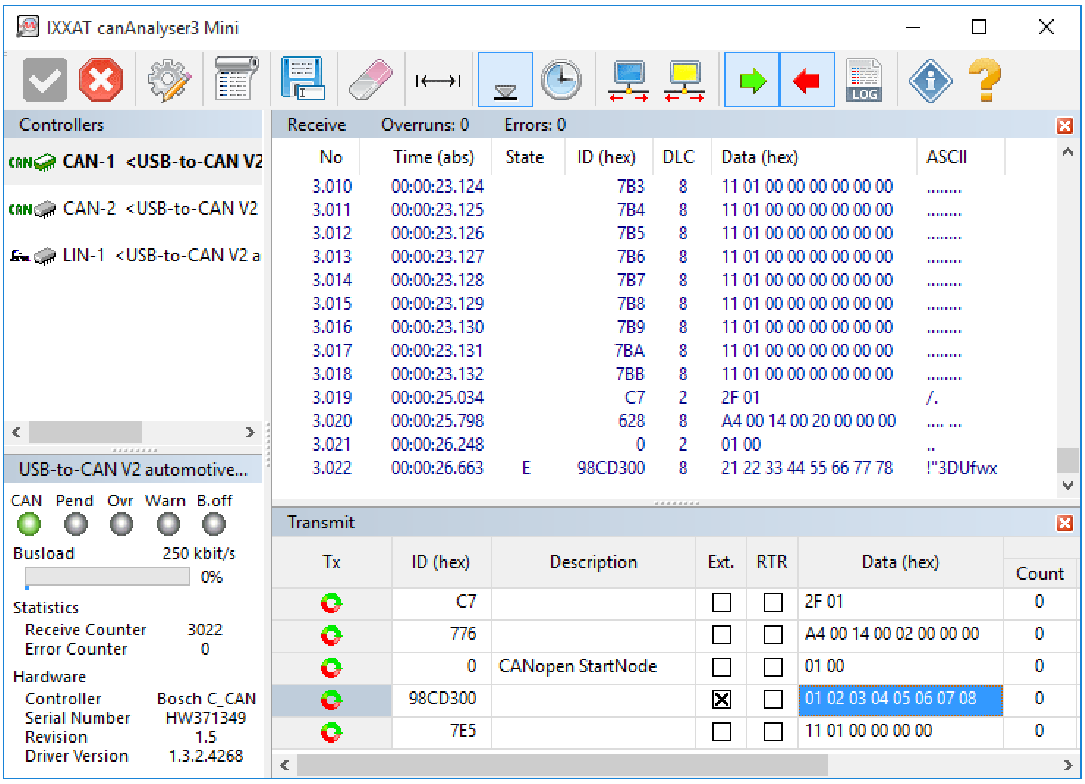
Chapter 1

Overview

1.1 Overview

IXXAT canAnalyser3 Mini is a bus monitor program which enables online monitoring of bus traffic on a CAN, LIN, and CAN-FD bus and the transmission of individual bus messages.
IXXAT canAnalyser3 Mini is contained in VCI4 Installation.
The main window (fig. 1.1) provides the following five areas:

- List of available Bus Controllers (top left)
- Current Status of selected Bus Controller (bottom left)
- Scroll View of received messages (top right)
- Transmit messages grid (bottom right)
- Toolbar



IXXAT canAnalyser3 Mini

Controllers

CAN-1 <USB-to-CAN V2

CAN-2 <USB-to-CAN V2

LIN-1 <USB-to-CAN V2 a

USB-to-CAN V2 automotive...

CAN Pend Ovr Warn B.off

Busload 250 kbit/s

0%

Statistics

Receive Counter	3022
Error Counter	0

Hardware

Controller	Bosch C_CAN
Serial Number	HW371349
Revision	1.5
Driver Version	1.3.2.4268

Receive Overruns: 0 Errors: 0

No	Time (abs)	State	ID (hex)	DLC	Data (hex)	ASCII
3.010	00:00:23.124		7B3	8	11 01 00 00 00 00 00 00	....
3.011	00:00:23.125		7B4	8	11 01 00 00 00 00 00 00	....
3.012	00:00:23.126		7B5	8	11 01 00 00 00 00 00 00	....
3.013	00:00:23.127		7B6	8	11 01 00 00 00 00 00 00	....
3.014	00:00:23.128		7B7	8	11 01 00 00 00 00 00 00	....
3.015	00:00:23.129		7B8	8	11 01 00 00 00 00 00 00	....
3.016	00:00:23.130		7B9	8	11 01 00 00 00 00 00 00	....
3.017	00:00:23.131		7BA	8	11 01 00 00 00 00 00 00	....
3.018	00:00:23.132		7BB	8	11 01 00 00 00 00 00 00	....
3.019	00:00:25.034		C7	2	2F 01	/.
3.020	00:00:25.798		628	8	A4 00 14 00 20 00 00 00	....
3.021	00:00:26.248		0	2	01 00	..
3.022	00:00:26.663	E	98CD300	8	21 22 33 44 55 66 77 78	!"3DUfwx

Transmit

Tx	ID (hex)	Description	Ext.	RTR	Data (hex)	Count
	C7		□	□	2F 01	0
	776		□	□	A4 00 14 00 02 00 00 00	0
	0	CANopen StartNode	□	□	01 00	0
	98CD300		☒	□	01 02 03 04 05 06 07 08	0
	7E5		□	□	11 01 00 00 00 00	0

Figure 1.1: IXXAT canAnalyser3 Mini main window

Chapter 2

Functions and operation

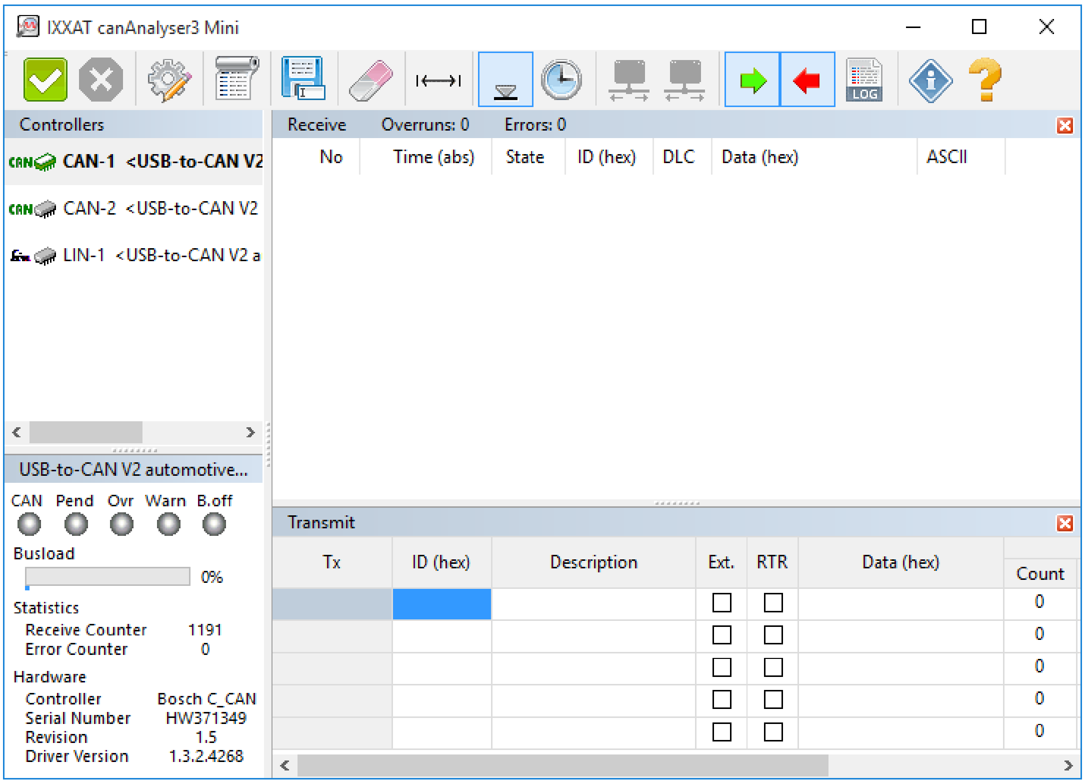
2.1 Starting the program

You start IXXAT canAnalyser3 Mini from the Start menu of VCI4 or by manually running the file `canAnaMini.exe`.

At first start (Fig. 2.1) the first available bus controller is selected, a default bit-rate is configured, and for safety reasons the communication is *de*activated.

Please check the default bit-rate prior to activating the communication by use of the first toolbar button.

Changing the bit-rate and advanced bus settings are described in chapter 2.6.

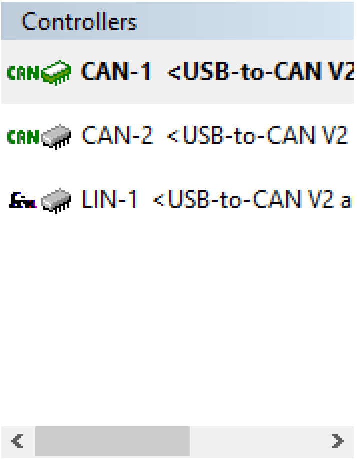


Statistics
Receive Counter 1191
Error Counter 0

Hardware
Controller Bosch C_CAN
Serial Number HW371349
Revision 1.5
Driver Version 1.3.2.4268

Tx	ID (hex)	Description	Ext.	RTR	Data (hex)	Count
			□	□		0
			□	□		0
			□	□		0
			□	□		0
			□	□		0

Figure 2.1: IXXAT canAnalyser3 Mini first start



Controllers

CAN-1 <USB-to-CAN V2

CAN-2 <USB-to-CAN V2

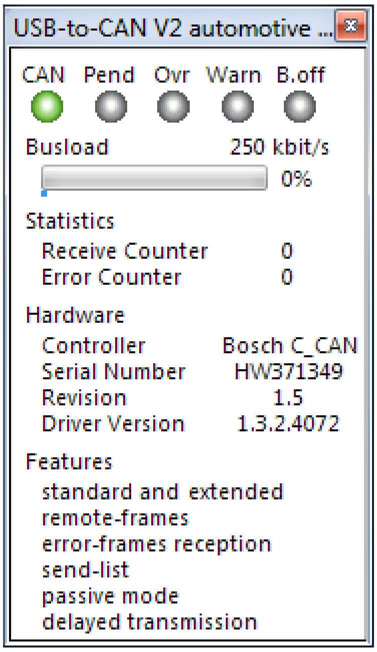
LIN-1 <USB-to-CAN V2 a

Figure 2.2: IXXAT canAnalyser3 Mini List of available Bus Controllers (top left)

2.2 Selection of the bus controller

IXXAT canAnalyser3 Mini utilises one bus controller at a time.
Double click an entry of the List of available Bus Controllers (Fig. 2.2) to activate it.
One can tell an activated bus controller by its color highlighting, and its bold denotation.
The icons in the List of available Bus Controllers indicate the state of the local bus controllers. See this table for the possible icons and their meaning:

Icon	Meaning	Means
CAN Lin can	Bus type: CAN, LIN or CAN-FD	
▨	Inactive controller with privileged access	By double clicking it becomes the active bus controller
🟩	Active CAN/LIN controller with privileged access	IXXAT canAnalyser3 Mini has privileged access to the Controller, all communication parameters can be set
🟦	Active CAN-FD controller with privileged access	IXXAT canAnalyser3 Mini has privileged access to the Controller, all communication parameters can be set
🟥	Active or inactive controller with regular access	IXXAT canAnalyser3 Mini cannot set communication parameters. Another application holds privileged access. Message reception and message transmission are possible without restriction.



Busload 250 kbit/s

0%

Statistics

Receive Counter	0
Error Counter	0

Hardware

Controller	Bosch C_CAN
Serial Number	HW371349
Revision	1.5
Driver Version	1.3.2.4072

Features

standard and extended
remote-frames
error-frames reception
send-list
passive mode
delayed transmission

Figure 2.3: CAN / CAN-FD status view

2.3 Current Status of selected Bus Controller

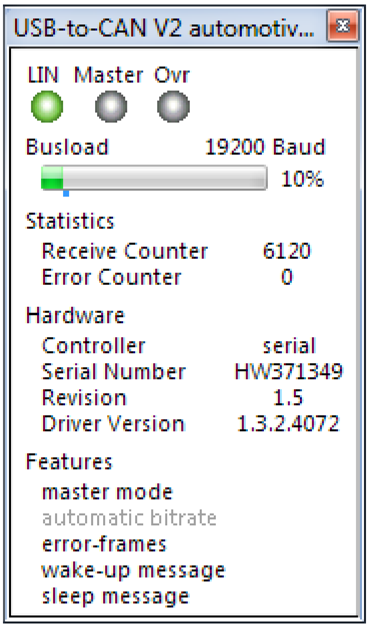
The status view (bottom left) shows the current bus controller state.
There is also hardware information regarding the corresponding IXXAT bus interface ("Hardware"), and the features flags of the bus controller ("Features").
The status view is bus type dependent.
Double clicking inside the status view makes the application's icon in Windows taskbar visualize the current busload as (green) progress indicator.
The CAN / CAN-FD status window (Fig. 2.3) comprises the following lights:

Meaning	Light off	Light on
CAN	CAN controller is stopped	CAN controller is started
Pend (Transmit pending)	All messages transmitted, transmit queue is empty	Messages not yet transmitted are in the hardware transmit queue
Ovr (Data overrun)	-	CAN controller overrun
Warn (Warning level)	-	CAN controller error counter in Error Warning Level
B.off (Bus off)	-	CAN-Controller in Bus off

The LIN status view (Fig. 2.4) comprises the following lights:

Meaning	Light off	Light on
LIN	LIN controller is stopped	LIN controller is started
Master	LIN operates in Slave mode	LIN operates in Master mode
Ovr (Data overrun)	-	LIN controller overrun

The hardware information section ("Hardware") contains these data:



Busload 19200 Baud
10%

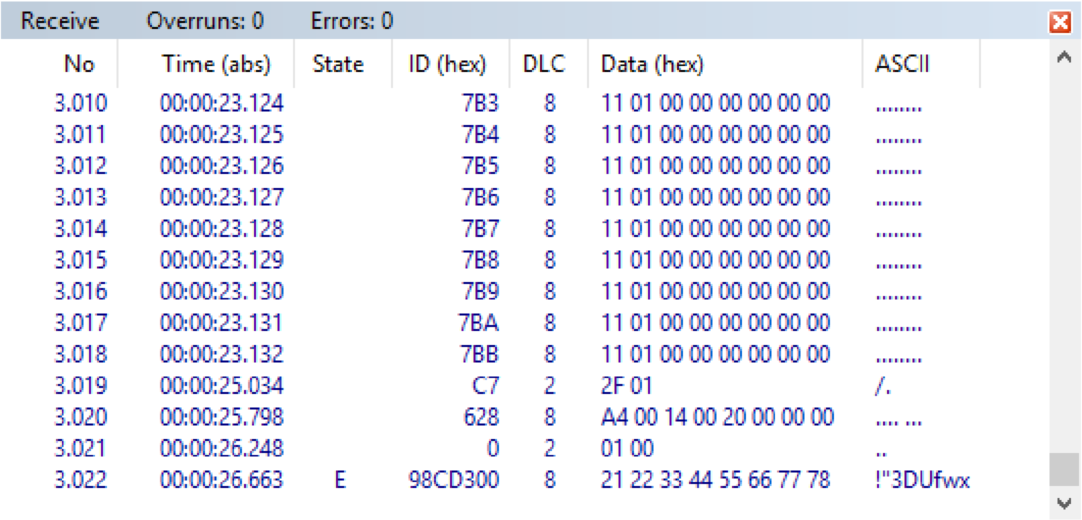
Statistics
Receive Counter 6120
Error Counter 0

Hardware
Controller serial
Serial Number HW371349
Revision 1.5
Driver Version 1.3.2.4072

Features
master mode
automatic bitrate
error-frames
wake-up message
sleep message

Figure 2.4: LIN status view

Data	Description
Controller	Bus controller name and manufacturer
Serial Number	IXXAT interface (board) serial number
Revision	IXXAT interface (board) revision number
Driver Version	VCI version number



Receive Overruns: 0 Errors: 0

No	Time (abs)	State	ID (hex)	DLC	Data (hex)	ASCII
3.010	00:00:23.124		7B3	8	11 01 00 00 00 00 00 00	....
3.011	00:00:23.125		7B4	8	11 01 00 00 00 00 00 00	....
3.012	00:00:23.126		7B5	8	11 01 00 00 00 00 00 00	....
3.013	00:00:23.127		7B6	8	11 01 00 00 00 00 00 00	....
3.014	00:00:23.128		7B7	8	11 01 00 00 00 00 00 00	....
3.015	00:00:23.129		7B8	8	11 01 00 00 00 00 00 00	....
3.016	00:00:23.130		7B9	8	11 01 00 00 00 00 00 00	....
3.017	00:00:23.131		7BA	8	11 01 00 00 00 00 00 00	....
3.018	00:00:23.132		7BB	8	11 01 00 00 00 00 00 00	....
3.019	00:00:25.034		C7	2	2F 01	/.
3.020	00:00:25.798		628	8	A4 00 14 00 20 00 00 00	....
3.021	00:00:26.248		0	2	01 00	..
3.022	00:00:26.663	E	98CD300	8	21 22 33 44 55 66 77 78	!"3DUfwx

Figure 2.5: IXXAT canAnalyser3 Mini Scroll View of received messages (top right)

2.4 Scroll View of received messages

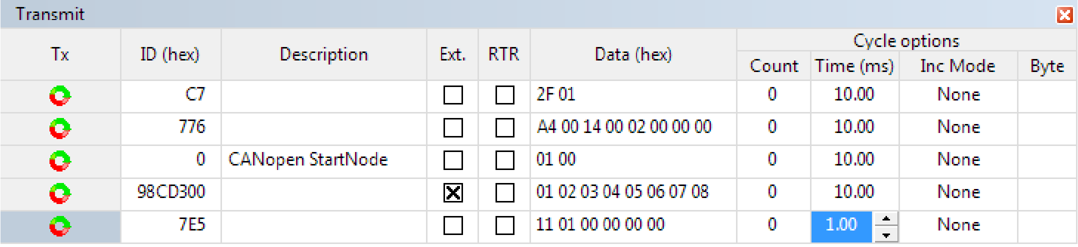
All messages are listed in the order of reception with the following information (Fig. 2.5):

Column	Meaning
No	Consecutive number of the received object
Time (abs/rel)	Time stamp of reception, optionally absolute in UTC time format or relative to the previously received message; by right-clicking on the column heading, the display of hours and minutes can be switched on or off
State	Display of the reception status flags
ID (hex)	Identifier of the received message
DLC	Data length code, codifies the number of data bytes
Data (hex)	Display of the received data in byte interpretation
ASCII	Display of the received data in ASCII interpretation

2.4.1 Display of the receive status flags

The receive status is displayed in the column **Status** with various letters. If the letter is visible, the status is set:

Status	Bustype	Meaning
C	-	Controller overrun: Messages were lost.
D	-	Driver queue overrun: The PC could not read out the driver queue fast enough. Messages were lost.
Q	-	Software queue overrun: The PC could not read out the internal software queue fast enough. Messages were lost.
S	-	Self-reception: Transmit and Receive view used the same controller.
E	CAN	Extended CAN frame: If E is not displayed, a standard CAN frame was received.
F	CAN-FD	A CAN-FD frame was received.
FF	CAN-FD	A CAN-FD frame having activated bitrate switching was received.
E	LIN	Enhanced CRC: A frame in enhanced CRC format acc.to LIN 2.0+ was received.
I	LIN	ID only: An ID only (i.e. a LIN Master request) message was received.



Transmit

Tx	ID (hex)	Description	Ext.	RTR	Data (hex)	Cycle options			
						Count	Time (ms)	Inc Mode	Byte
○	C7		□	□	2F 01	0	10.00	None	
○	776		□	□	A4 00 14 00 02 00 00 00	0	10.00	None	
○	0	CANopen StartNode	□	□	01 00	0	10.00	None	
○	98CD300		☒	□	01 02 03 04 05 06 07 08	0	10.00	None	
○	7E5		□	□	11 01 00 00 00 00	0	1.00	None	

Figure 2.6: CAN Transmit messages grid (bottom right) (instance)

2.5 Transmit messages grid

The following functionality is provided:

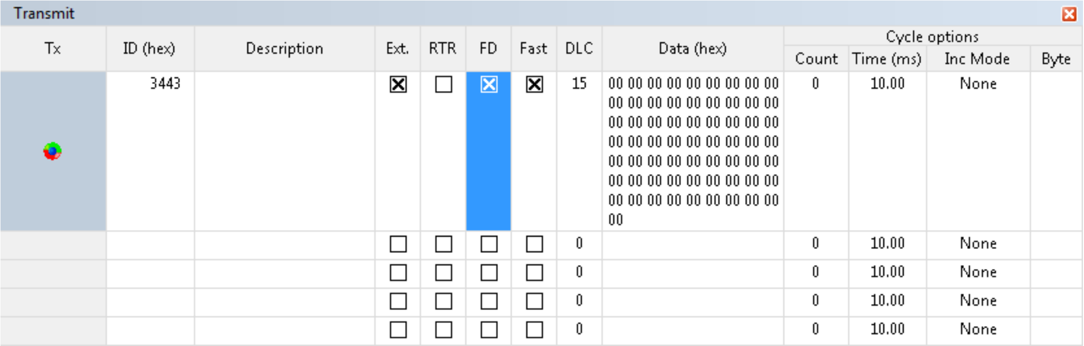
- Transmission of individual data and remote messages
- Transmission of any number of data or remote messages
  - with a certain cycle time
  - with incrementing of the identifier or of any data byte or word

2.5.1 CAN transmit grid

For CAN bus controllers the objects to be transmitted are entered in a fixed table consisting of five rows.

The CAN transmit grid (Fig. 2.6) has the following columns:

Column	Meaning
Tx	Icon ○ for transmission state visualization. It's rotating while the message's cyclic transmission is active. Icon ● shows that cyclic transmission is done directly by the hardware.
ID (hex)	Identifier of the transmit object
Description	Additional user-defined description of this transmit object. This description allows differentiation of the transmit objects with the same identifier.
Ext.	Defines whether a telegram is transmitted in extended frame format (29 bit identifier). This does NOT override the protocol setting in the CAN settings dialog.
RTR	Defines whether a data or a remote telegram is transmitted (only CAN)
Data (hex)	Data of the layer-2 message
Cycle options	The settings for cyclic transmit objects are specified in this column
Count	Number of transmit repeats; 0 stands for continual transmission
Time (ms)	Cycletime in milliseconds
Inc Mode	Operating mode of cyclic transmission (with/without increment). **None**: No incrementing. **Identifier**: Incrementing of identifier with each transmission. **Byte (Data)**: Incrementing of the databyte defined in the column **Byte** with each transmission. **Word (Data)**: Incrementing of a 16-bit value (compiled from 2 databytes), beginning with the databyte defined in the column **Byte** with each transmission
Byte	Start byte, with which incrementing of the data field is carried out when an increment mode is switched on (see Inc Mode column).



Transmit

Tx	ID (hex)	Description	Ext.	RTR	FD	Fast	DLC	Data (hex)	Cycle options			
									Count	Time (ms)	Inc Mode	Byte
●	3443		☒	□	☒	☒	15	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	0	10.00	None	
			□	□	□	□	0		0	10.00	None	
			□	□	□	□	0		0	10.00	None	
			□	□	□	□	0		0	10.00	None	
			□	□	□	□	0		0	10.00	None	

Figure 2.7: CAN-FD Transmit messages grid (bottom right) (instance)

2.5.2 CAN-FD transmit grid

For CAN-FD bus controllers the objects to be transmitted are entered in a fixed table consisting of five rows.

The CAN-FD transmit grid (Fig. 2.7) has the following columns in addition to the ones of the CAN transmit grid:

Column	Meaning
FD	Defines whether a CAN-FD frame is transmitted. This is only possible if the option **Enable FD Frames (FD)** in the CAN-FD settings dialog is enabled.
Fast	Defines whether a telegram is transmitted as CAN-FD in fast speed (FF).
DLC	Codifies the length of the data. The value range is 0 to 15. Values 0 to 8 correspond to the actual byte length, for the values 9 to 15 these increments apply: 12, 16, 20, 24, 32, 48, 64 bytes data length. The input is being quantised accordingly. This column and the column Data (hex) are mutually adjusting.

2.5.3 LIN transmit grid

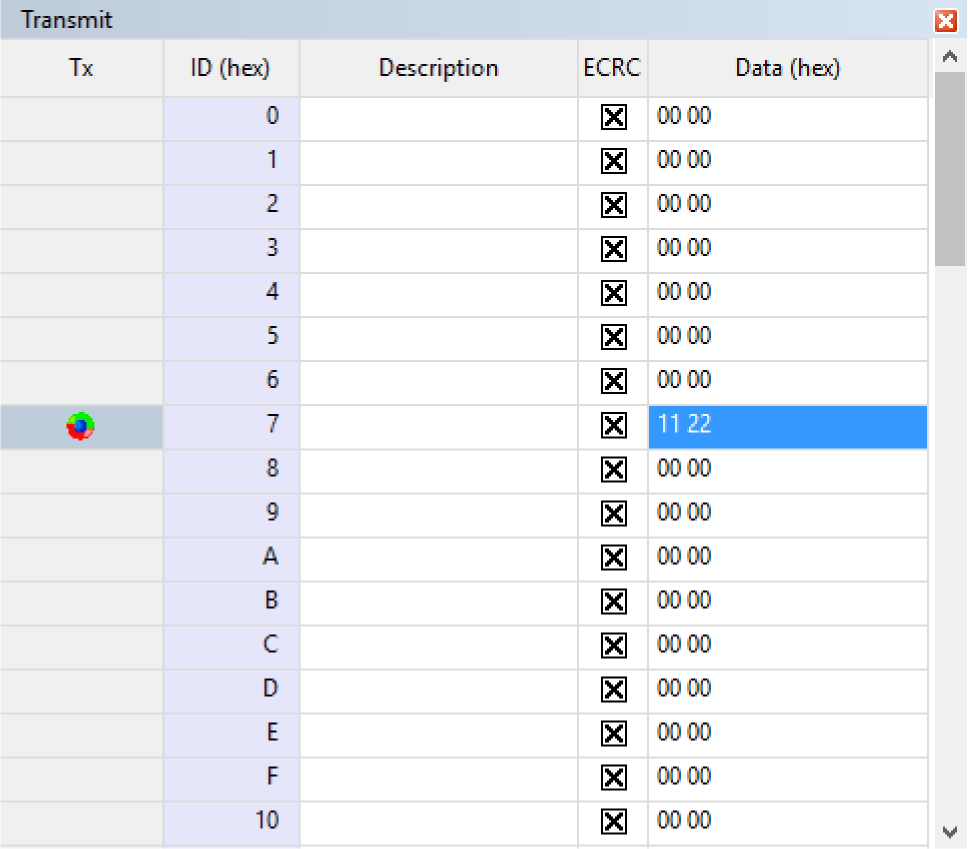
For LIN bus controllers (Fig. 2.8) it shows a static table with all 64 possible LIN identifiers sorted ascendingly. Special messages fall into line with them.
Depending on the LIN operating mode both the layout and the behaviour are slightly different. The LIN operating mode is set in the hardware configuration dialog of the LIN Controller in the IXXAT canAnalyser3 Mini bus settings dialog. It can be switched at any time (Fig. 2.9).

There is a separate configuration set for LIN Master mode and for LIN Slave mode.

Contrary to CAN and LIN Master mode, messages can not spontaneously be sent in LIN Slave mode. A LIN Slave responds to an external LIN Master request (IDO), which is handled by the hardware controller itself. The latter uses a so-called Response Table, that is visualised by the Transmit grid in Slave mode (Fig. 2.8). This hardware based processing is also called *auto response* or *auto transmit* in the following.

Even in LIN Master mode, slave behaviour is implemented in firmware by means of an *implicit Response Table*. This can make for the curious situation where the Master responds to its own requests. Hence, operation and presentation of the Response Table in LIN Master mode shall be addressed particularly here. See also the popup menu description below.

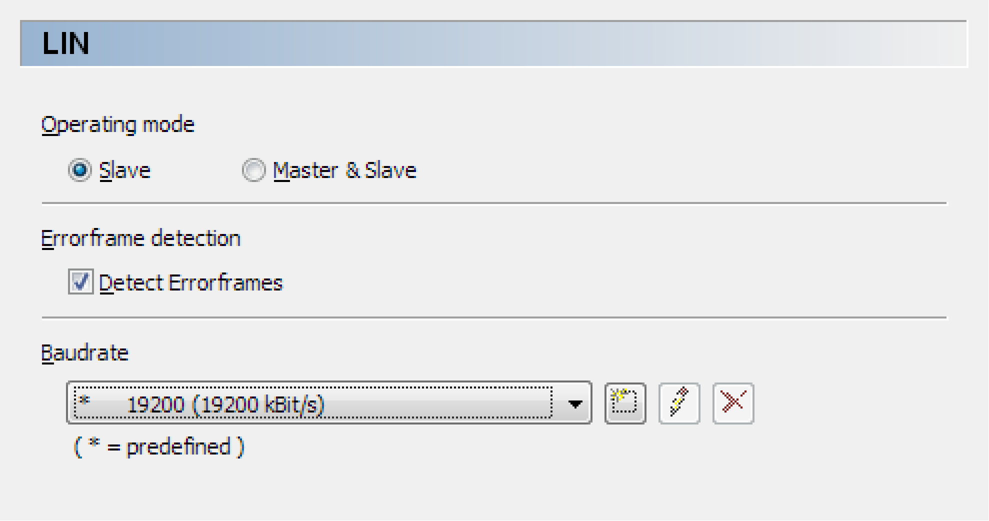
By default, all LIN identifiers of the Response Table are disabled. This is illustrated by an empty **Tx** column. A LIN identifier needs to be enabled explicitly both in Slave Mode and in Master



Transmit

Tx	ID (hex)	Description	ECRC	Data (hex)
	0		☒	00 00
	1		☒	00 00
	2		☒	00 00
	3		☒	00 00
	4		☒	00 00
	5		☒	00 00
	6		☒	00 00
●	7		☒	11 22
	8		☒	00 00
	9		☒	00 00
	A		☒	00 00
	B		☒	00 00
	C		☒	00 00
	D		☒	00 00
	E		☒	00 00
	F		☒	00 00
	10		☒	00 00

Figure 2.8: LIN Transmit messages grid (bottom right) - Slave mode (instance)



LIN

Operating mode

● Slave ○ Master & Slave

Errorframe detection

☑ Detect Errorframes

Baudrate

* 19200 (19200 kBit/s)

(* = predefined)

Figure 2.9: LIN settings

mode to allow for transmitting it automatically it. An enabled identifier is one with a ○ resp ● icon in the **Tx** column. In LIN Slave mode, simply click it, or use the popup menu to enable it.

In LIN Master mode, when manual and cyclic transmission as with CAN is possible, not the *Response Table* of the LIN Controller, but a *transmit table* is displayed. Handling of the *implicit Response Table* is woven into it. A Response Table entry clearly has less parameters than a transmit table entry, only the **data** field (bytes and length). More on that later.
The entries are transmitted by selecting the row and then clicking their toolbar matches **Transmit Single Message** resp **Transmit Cyclic Message**.

When a Response Table entry in LIN Master mode is activated, its presentation alters: The **Data** cell turns to royal blue, the **IDO** box gets checked, and the send icon becomes ●. So, the contents of the auto response is entered in the data cell, which is the trick of weaving the Response Table entry into the transmit table, since the data cell is unoccupied for a checked **IDO** cell, and is available for entering the auto response around it.
Once again, the **IDO** checkbox allows for switching the presentation of the response table entry and the transmit table entry of a LIN identifier in LIN Master mode. Physically both are existing independently and simultaneously, and can be configured differently, of course. Even if the cells depicting the cycle options (**Count**, **Cycle Time** etc) are shown with such a Response Table entry, they refer to the corresponding transmit table entry (otherwise they would be colored in royal blue). Alas, the data field of an auto response cannot be configured to cyclic changes !

The LIN (Master mode) transmit grid has the following columns:

Column	Meaning
Tx	Icon ○ signals an enabled identifier. It is rotating while the message's cyclic transmission is active. Icon ● shows that a LIN Response Table entry is enabled which is handled directly by the hardware. It is permanently rotating.
Identifier	Identifier of the transmit object
Description	Additional user-defined description of this transmit object. This description allows differentiation of the transmit objects with the same identifier.
ECRC	Defines whether a message is transmitted in enhanded CRC format (LIN 2.0+)
IDO	Defines whether an Identifier only frame is transmitted (Master mode required)
Data	Data of the layer-2 message
Count	Number of transmit repeats; 0 stands for continual transmission
Cycle Time	Cycletime in milliseconds
Inc Mode	Operating mode of cyclic transmission (with/without increment). **None**: No incrementing. **Identifier**: Incrementing of identifier with each transmission. **Byte (Data)**: Incrementing of the databyte defined in the column **Byte** with each transmission. **Word (Data)**: Incrementing of a 16-bit value (compiled from 2 databytes), beginning with the databyte defined in the column **Byte** with each transmission
Byte	Start byte, with which incrementing of the data field is carried out when an increment mode is switched on (see Mode column).

There are different background colors used to illustrate the input rules of a cell:
Light lavender colored cells are for informational purposes only. They are readonly and cannot be selected.
The data column is usually highlighted in green, to indicate a fixed data length.

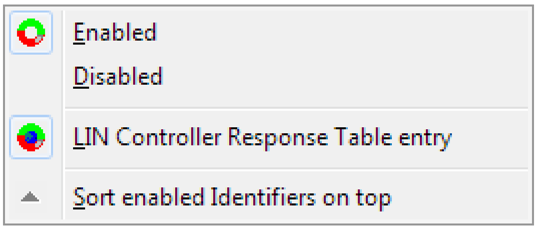

Enabled
Disabled
LIN Controller Response Table entry
Sort enabled Identifiers on top

Figure 2.10: Context menu LIN (full)

A royal blue colored cell signals that LIN Controller Response Table data is shown in Master mode.

The popup menu (Fig. 2.10) of the LIN transmit table has the following entries:

Menu item	Function
Enabled	Indicates an enabled Response Table entry. Only enabled entries will be auto transmitted by the LIN Controller. For LIN Slave mode only !
Disabled	Indicates a disabled Response Table entry. For LIN Slave mode only !
LIN Controller Response Table entry	Enable Response Table entry. In addition to the manual and cyclic transmission, this LIN identifier will be transmitted automatically by the LIN Controller upon Master request (IDO). For LIN Master mode only !
Sort enabled Identifiers on top	Brings all enabled rows to the top of the transmit table

2.5.4 Editing the fields

The editable fields change automatically to edit mode as soon as a numerical or alphanumerical key resp the F2 or the SPACE key is pressed. There is a difference between non-destructive and destructive editing. By pressing F2 or SPACE the cursor will be placed at the end of the field keeping the present values, whilst simply starting to type at an editable field will overwrite the current contents. In either case, the editing can be aborted pressing the ESC key.
Editing is finished by pressing the ENTER key, or by clicking on another cell of the transmit table. Readonly fields are identified by a different background color (lavender).

2.5.5 Manual transmission

Individual messages from the table are transmitted by selecting the message and triggering the transmit command.
A message is selected by:

- Clicking on the message with the mouse
- Moving the marking bar with the cursor keys ↑ or ↓ on the keyboard.

Once a message is selected, it can be transmitted by:

- Pressing the key F5
- Clicking the **Transmit single message** button in the toolbar
- Clicking with the left mouse button on the transmit icon ○ in the first column

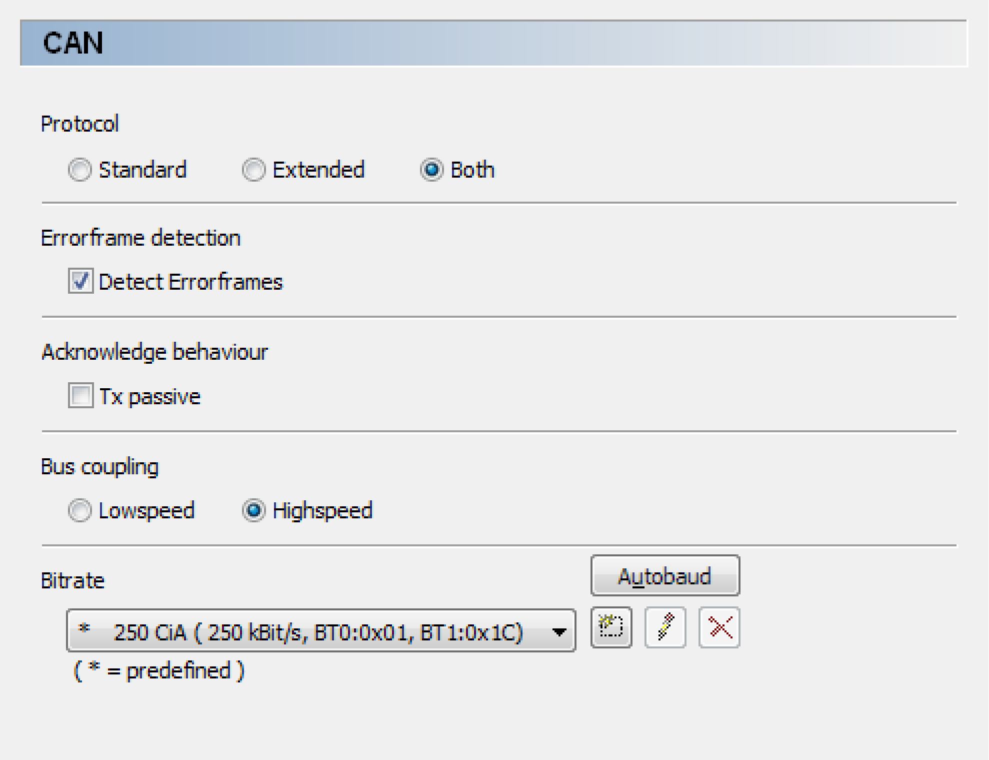
2.5.6 Cyclic transmission

To be able to transmit messages cyclically, values must be entered in the fields **Count** and **Time** of the column **Cycle options**. A cyclic message can be transmitted both cyclically (automatically) and individually (manually).
Cyclic transmission is carried out by:

- Pressing the key F6
- Clicking the **Transmit cyclic message** button in the toolbar
- Holding the Ctrl-key and at the same time clicking with the left mouse button on the transmit icon in the first column
- Holding the Ctrl-key and at the same time clicking with the left mouse button on the **Transmit cyclic message** button in the toolbar to begin cyclic transmission of all messages

While the selected message is transmitted cyclically, its icon rotates in the transmit table .
When the number of messages specified under **Count** has been transmitted, no further messages of this transmit object are transmitted and the icon stops rotating.
The cyclic transmission of a selected message can be stopped manually by:

- Clicking again on the **Transmit cyclic message** button in the tool bar
- Pressing again the F6 key



CAN

Protocol

○ Standard ○ Extended ● Both

Errorframe detection

☑ Detect Errorframes

Acknowledge behaviour

□ Tx passive

Bus coupling

○ Lowspeed ● Highspeed

Bitrate

* 250 CiA (250 kBit/s, BT0:0x01, BT1:0x1C)

(* = predefined)

Figure 2.11: CAN Settings

2.6 Bus settings - Basic and Advanced

The third toolbar icon opens up the Bus Settings dialogue.

2.6.1 CAN Settings

The settings of the CAN controller are:

- Message format
- Error frame detection
- Acknowledge behavior
- Bus coupling
- Timing parameters

Fig. 2.11 shows the dialog to set the CAN controller parameters. In order to identify timing parameters (**Bitrate**) more easily, they are managed via symbolic names. Using the button symbols next to the name, the parameters which are configured for this name can be altered, new entries can be added and old ones can be deleted.

The meaning of the parameters:

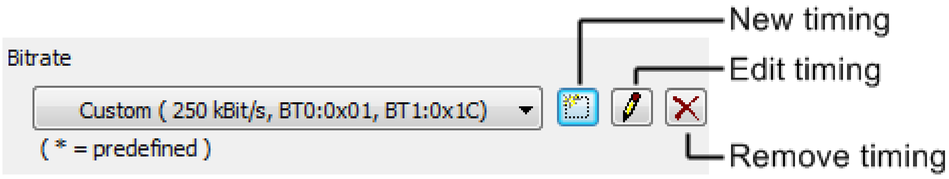


Figure 2.12: Create new entry in the Timings dialog or delete entry

Setting	Function
Protocol	Defines the message format with which the CAN controller works (standard 11-bit identifier and/or extended 29-bit identifier)
Detect Errorframes	If this checkbox is set, error frames are passed on to the associated analysis View
Tx passive	If this checkbox is set, the CAN controller is initialized in Tx-passive mode, i.e. it listens on the bus but behaves passively and therefore does not transmit any acknowledgements or error frames.
Bus coupling	Selects the physical bus coupling of the CAN controller (Highspeed by default, Lowspeed if available). Lowspeed is a fault-tolerant 2-wire standard with max 125 kBit/sec bitrate acc.to ISO 11898-3.

Setting a bitrate

The bitrate is selected via the symbolic name of the timing. The timing parameters assigned to the name can be altered, new parameter sets can be added and old ones can be deleted. For this, the buttons next to the symbolic name (Fig. 2.12) are pressed.

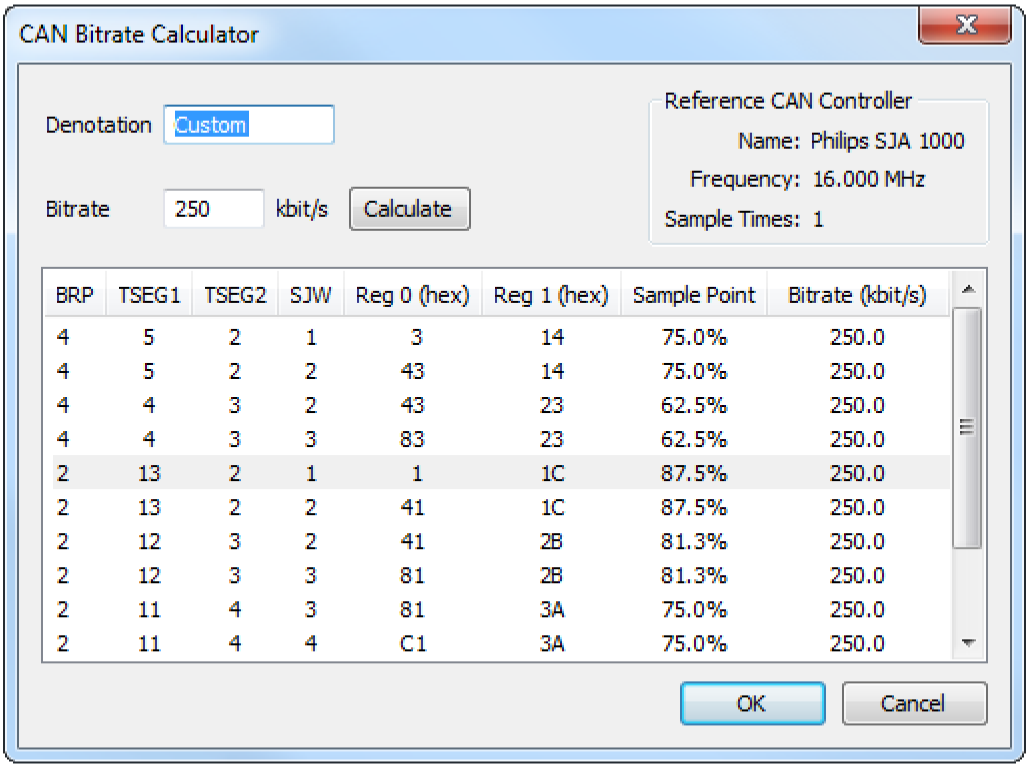
CAN Bitrate Calculator

The CAN bitrate calculator (Fig. 2.13) can be opened via the **New** or **Edit** button in the CAN Settings dialog. Here you can choose the timing parameters fitting a desired bitrate.
Once you enter the desired bitrate and press the **Calculate** button, the table displays all suitable combinations of the CAN controller's registers. Choose one by moving the highlighted line up and down, and press **OK** to accept these timing parameters.

Description of the CAN bitrate calculator input fields:

Field	Description
Denotation	Symbolic name of the timing
Bitrate (kbit/s)	Bitrate to be calculated in kBit per second

Description of the columns in the list of calculated values:



CAN Bitrate Calculator

Denotation: Custom

Bitrate: 250 kbit/s

Reference CAN Controller

Name: Philips SJA 1000

Frequency: 16.000 MHz

Sample Times: 1

BRP	TSEG1	TSEG2	SJW	Reg 0 (hex)	Reg 1 (hex)	Sample Point	Bitrate (kbit/s)
4	5	2	1	3	14	75.0%	250.0
4	5	2	2	43	14	75.0%	250.0
4	4	3	2	43	23	62.5%	250.0
4	4	3	3	83	23	62.5%	250.0
2	13	2	1	1	1C	87.5%	250.0
2	13	2	2	41	1C	87.5%	250.0
2	12	3	2	41	2B	81.3%	250.0
2	12	3	3	81	2B	81.3%	250.0
2	11	4	3	81	3A	75.0%	250.0
2	11	4	4	C1	3A	75.0%	250.0

Figure 2.13: The CAN bitrate calculator

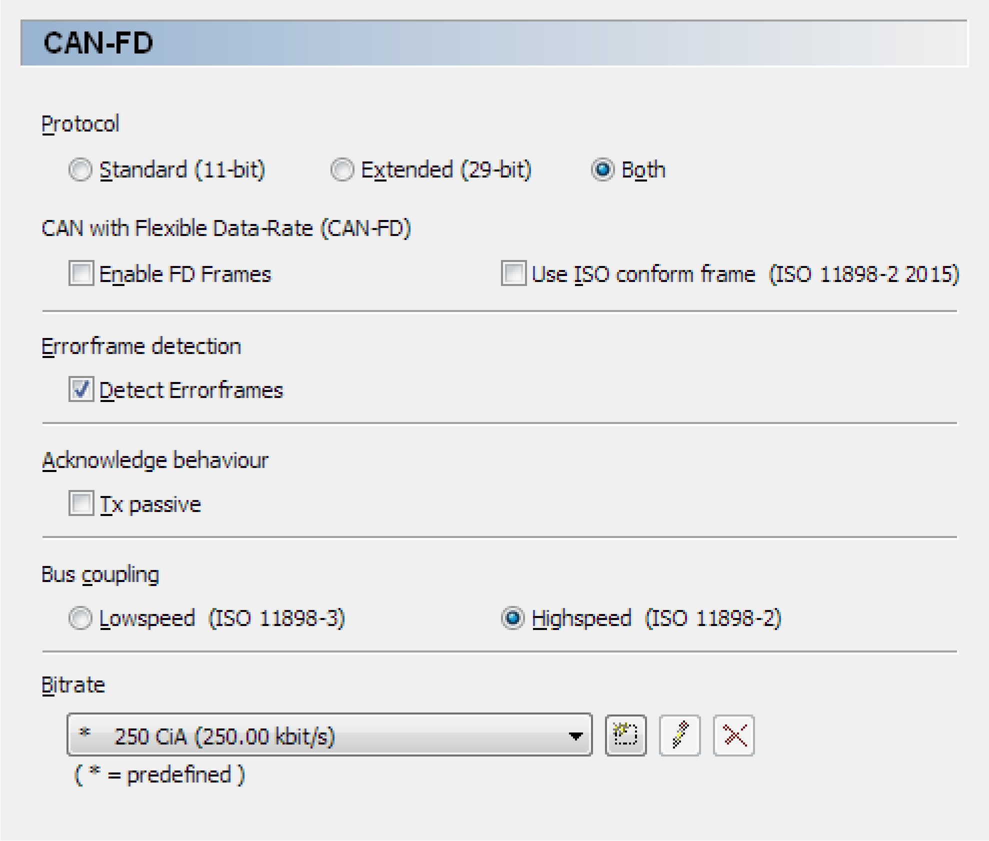
Column	Description
BRP	Baudrate Prescaler
TSEG1	Timing Segment 1
TSEG2	Timing Segment 2
SJW	Synchronisation Jump Width
Reg 0 (hex)	Bus timing register 0 (hexadecimal format)
Reg 1 (hex)	Bus timing register 1 (hexadecimal format)
Sample Point	Sample location
Bitrate (kbit/s)	Calculated bitrate with the values of the marked line

Please note: Columns *Reg 0* and *Reg 1* summarize the values of the following five columns: BRP, TSEG1, TSEG2, SJW, and Sample Point, bitcoded in hexadecimal format. Also, column *Bitrate* displays the resulting actual bitrate, which is expected to be equal to the entered desired bitrate.

2.6.2 CAN-FD Settings

The settings of the CAN-FD controller (which include the CAN settings also) are:

- Message format
- Error frame detection
- Acknowledge behavior
- Buscoupling
- Timing parameters



CAN-FD

Protocol

○ Standard (11-bit) ○ Extended (29-bit) ● Both

CAN with Flexible Data-Rate (CAN-FD)

□ Enable FD Frames □ Use ISO conform frame (ISO 11898-2 2015)

Errorframe detection

☑ Detect Errorframes

Acknowledge behaviour

□ Tx passive

Bus coupling

○ Lowspeed (ISO 11898-3) ● Highspeed (ISO 11898-2)

Bitrate

* 250 CiA (250.00 kbit/s)

(* = predefined)

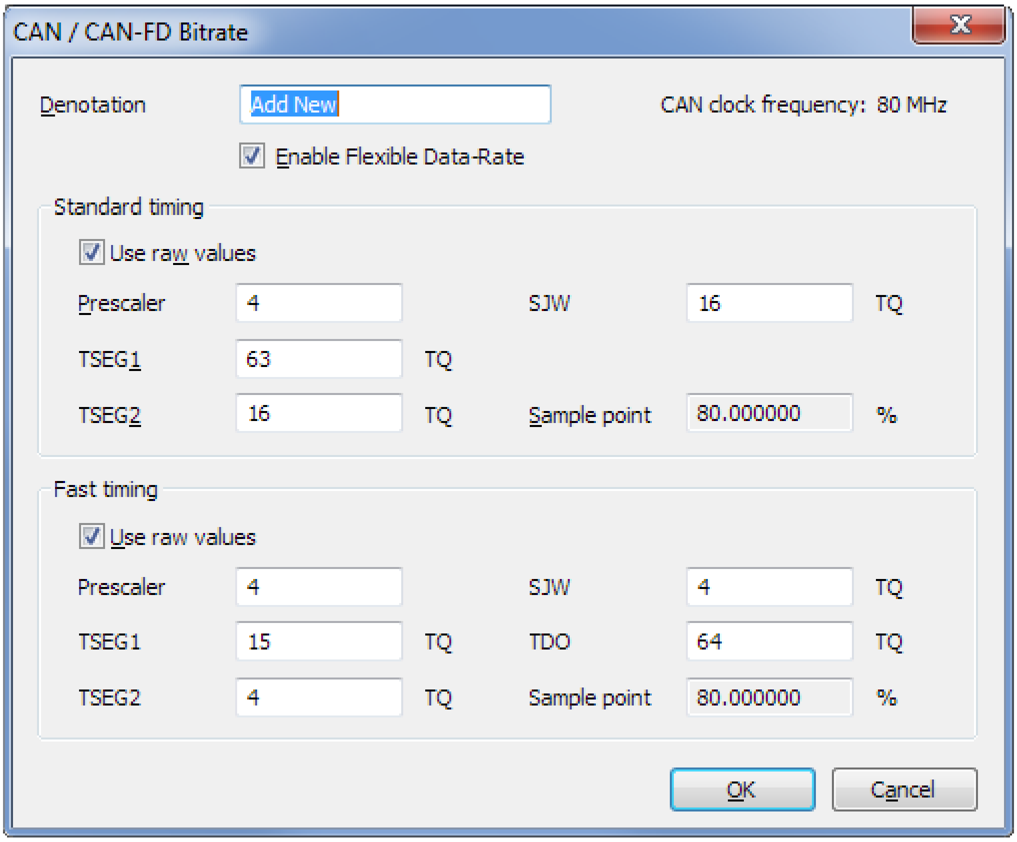
Figure 2.14: CAN-FD Settings

Fig. 2.14 shows the dialog to set the CAN-FD controller parameters. In order to identify timing parameters (**Bitrate**) more easily, they are managed via symbolic names. Using the button symbols next to the name, the parameters which are configured for this name can be altered, new entries can be added and old ones can be deleted.

The meaning of the parameters:

Setting	Function
Protocol	Defines the message format with which the CAN-FD controller works (standard 11-bit identifier and/or extended 29-bit identifier)
Enable FD Frames (FD)	Allows for the usage of CAN-FD on the bus
Use ISO conform frame	Force ISO conform CAN-FD frames according to ISO 11898-2 2015
Detect Errorframes	If this checkbox is set, error frames are passed on to the associated analysis View
Tx passive	If this checkbox is set, the CAN-FD controller is initialized in Tx-passive mode, i.e. it listens on the bus but behaves passively and therefore does not transmit any acknowledgements or error frames
Bus coupling	Selects the physical bus coupling of the CAN-FD controller (Highspeed by default, Lowspeed if available). Lowspeed is a fault-tolerant 2-wire standard with max 125 kBit/sec bitrate acc.to ISO 11898-3

Please note: Running CAN-FD e.g. on a low speed line makes no sense of course, but the CAN-FD controller can be configured to behave like a plain CAN controller if the following conditions are met: Enabling neither FD nor ISO frames, and abstaining from fast bit timings (as shown in figure 2.14).



Denotation: Add New

CAN clock frequency: 80 MHz

☑ Enable Flexible Data-Rate

Standard timing

☑ Use raw values

Prescaler 4 SJW 16 TQ

TSEG1 63 TQ

TSEG2 16 TQ Sample point 80.000000 %

Fast timing

☑ Use raw values

Prescaler 4 SJW 4 TQ

TSEG1 15 TQ TDO 64 TQ

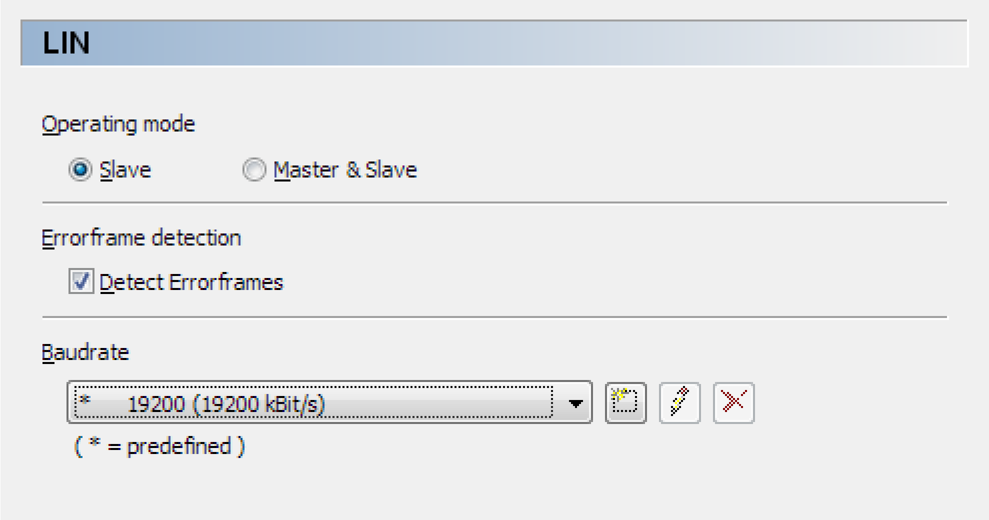
TSEG2 4 TQ Sample point 80.000000 %

Figure 2.15: The CAN-FD bitrate dialog

CAN-FD Bitrate Dialog

The CAN-FD bitrate dialog (Fig. 2.15) can be opened via the **New** or **Edit** button in the CAN-FD Settings dialog.
Firstly, there are two timing sets: **Standard Timing**, and **Fast Timing**. This matches the concept of CAN-FD. As the name says, CAN-FD transmits only the data field of a message in fast speed. The rest of the message, like e.g. the identifier, in normal speed. The speed switch happens in transmission, during every single message. Accordingly, there are two timings, one for normal speed (Standard Timing), and one for fast speed (Fast Timing). **Fast Timing** is accessible if **Enable Flexible Data-Rate (Fast)** is checked.
By the checkboxes **Use raw values** the controller dependent native mode (Raw Mode) can be selected. In this mode the CAN-FD controllers' register values are set straightly, rather than being calculated by VCI as intermediary based on the bit rate entered.

Description of the CAN-FD bitrate dialog input fields:



LIN

Operating mode

● Slave ○ Master & Slave

Errorframe detection

☑ Detect Errorframes

Baudrate

* 19200 (19200 kBit/s)

(* = predefined)

Figure 2.16: LIN Settings

Field	Description
Prescaler	Preceding prescaler in the CAN-FD controller. Only visible if Use raw values is checked.
Bitrate	Desired Bitrate. Only visible if Use raw values is UNchecked.
TSEG1	Length of Time Segment 1 in time quantas. If Use raw values is UNchecked, it comprises the bit timing segments PROP und PHASE1. If Use raw values is checked, it comprises the bit timing segments SYNC, PROP und PHASE1.
TSEG2	Length of Time Segment 2 in time quantas.
SJW	Sync Jump Width for (re-)synchronisation in time quantas.
TDO	Transceiver Delay Offset in time quantas.

Please note: The displayed *Sample point* are calculated from the ratio of *TSEG1* and *TSEG2*. Please find further explanations in the VCI programming manual (PDF).

2.6.3 LIN Settings

The settings of the LIN controller are:

- Operating mode
- Errorframe detection
- Baudrate

The meaning of the parameters in the **LIN** section:

Setting	Function
Operating mode	Switches between Slave mode and Master mode. Since the LIN controller Response Table is active in Master mode too, it is denoted as Master & Slave here.
Detect errorframes	If this checkbox is set, error frames are passed on to the associated analysis View.
Baudrate	Selects the physical serial baudrate of the LIN controller.

Setting a baudrate

The baudrate is selected from the combobox. New baudrates can be defined and old ones can be deleted. For this, the buttons next to the symbolic name are pressed. In order to identify user baudrates more easily, they are managed via symbolic names.

2.7 Event Log

The control panel has its own logging facility that records internal events and errors. It can be made visible by menu command **View | Event Log** and contains the following information:

Column	Meaning
Icon	Kind of event: Success, Information, Warning, Error, or subsequent message line
Timestamp	Date and Time of the event
Sequence	Message number based on the IXXAT canAnalyser3 Mini session
Code	Hexadecimal errorcode
Thread	Hexadecimal thread identifier
Module	Name of IXXAT canAnalyser3 Mini module that reported the event
Message	Message text

The eventlog is a comma separated text file which is located in the user folder (e.g. in C:\Users\John\AppData\Local\IXXAT\canAnalyserMini\3.1\Log\*\canAnalyser.log)
Use **View** main menu to configure which event kinds should be shown in the Event Log window. Menu command **View | Clear Eventlog** empties the Event Log.

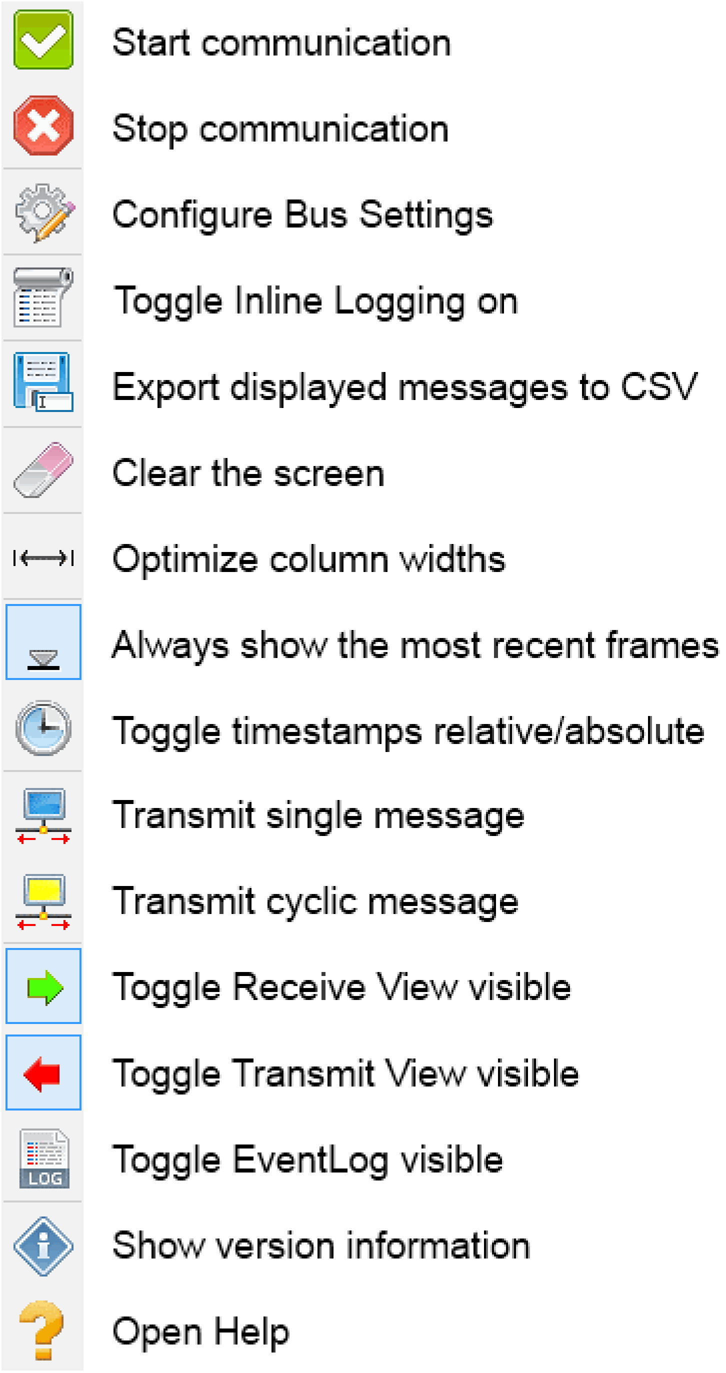


Figure 2.17: IXXAT canAnalyser3 Mini Toolbar

2.8 Toolbar

IXXAT canAnalyser3 Mini can be operated all by its toolbar (fig. 2.17).

2.9 Hotkeys

Ctrl+E	Export all available received messages to a file
Ctrl+C	Copy marked lines CSV formatted from Receive View to clipboard
Ctrl+F2	Toggle Marker in Receive View
Shift+F2	Go to Previous Marker in Receive View
F1	Online-Help
F2	Go to Next Marker in Receive View
Ctrl+0	Jump to Time Reference message in Receive View
F2	Start editing in Transmit Grid
Space	Start editing RESP (Un)Check a checkbox in Transmit Grid
F5	Send message
F6	Send cyclic message
F8	Clear Receive View
PageUp	Scroll one page backward in Receive View
PageDown	Scroll one page ahead in Receive View
Ctrl+PageUp	Scroll 1000 messages backward in Receive View
Ctrl+PageDown	Scroll 1000 messages ahead in Receive View
Ctrl+1..9	Jump to 10%..90% of Receive View
Ctrl+Up	Increase Cycle Time by 1 in Transmit Grid
Ctrl+Down	Decrease Cycle Time by 1 RESP Display drop-down list in Transmit Grid

2.10 Drag-and-Drop

Received messages might be dragged from Scroll View of received messages to a row of Transmit messages grid. Upon dropping, the existing transmit message will be replaced, transcribing all receive message attributes (Identifier, DLC, Data, Format) to the transmit message.

Chapter 3

Limitations of IXXAT canAnalyser3 Mini

3.1 IXXAT canAnalyser3

IXXAT canAnalyser3 is a modern, powerful tool for the development, operation, maintenance and testing of CAN/LIN/CAN-FD networks.

The IXXAT canAnalyser3 is based on a modular concept: communication with the driver and the hardware is handled by a central server application, the control panel, to which several client applications, so-called analysis modules, can be connected. These analysis modules are managed by the control panel and they are supplied with the messages received by the hardware. Time-critical pre-processing, such as buffering and stamping of the telegrams with the time of reception is carried out on the hardware.

The analysis modules provide the actual analysis functionality with pre-processing and editing of the telegrams supplied by the control panel. The network is also stimulated via analysis modules, which transfer the messages to be transmitted to the server, which handles further communication with the hardware.

The advantage of this structure lies in the modularity and easy extendibility. In addition, the same analysis modules can be started more than once. With the aid of different module settings (e.g. filters), a better overview can be obtained.

The following basic functions are provided by the analysis modules::

- Online display of layer-2 messages (Receive module)
- Individual and cyclic transmission of layer-2 messages (Transmit module)
- Tracing and offline analysis of layer-2 messages (Trace module)
- Text and graphic display of interpreted messages (signals) along with statistic signals (Signal module)
- Sending of signals (SignalTransmit module)
- Time-synchronous analysis of several buses
- Display of bus load
- Emulation of nodes and protocol sequences by processing command-controlled message sequences (Sequencer module)
- Data modification and cycle time monitoring

Extended functionality could be added by creating user defined modules in a .NET compatible language. Examples in C# and VB.NET for typical scenarios are installed during setup.
Further specialized analysis modules for CANopen, DeviceNet and J1939 are available.

3.2 Differences to IXXAT canAnalyser3

The free IXXAT canAnalyser3 Mini is completely derivated from IXXAT canAnalyser3.
Apart from the single document user interface (one single window) the feature set of Scroll View of received messages and Transmit messages grid is impaired as follows:

- Timestamp precision merely in milliseconds
- No reception filters nor display filters
- No hexadecimal/decimal representation toggling
- No views customization (guides / word wrap / fonts etc.)
- Column "Message name" is absent
- Only five transmit messages
- Transmit grid row heights are slightly increased

Appendix A

Export

A.1 Export of CSV files

Many export opportunities within IXXAT canAnalyser3 Mini create CSV files (comma separated value). This text based format is suitable to export tabular data and could be read by most spreadsheet applications. Nevertheless there are some differences which are subject of this chapter.

A.1.1 CSV format used by IXXAT canAnalyser3 Mini

The list separator character, which is language dependant and could be altered in the Windows ® control panel (via language settings), is used in all exports to separate columns. Lines are delimited by carriage return/line feed. Cell data is surrounded by quotation marks ("). Quotation marks within cell data are replaced by an escape sequence ("").

A.1.2 Import in Microsoft ® Excel

CSV files could be imported into excel by selecting the file type "Text files" within the "File open" dialog. Depending on the file extension (.csv or .txt) of the selected file Excel uses different import filters.
Files with the extension ".csv" will be imported by Excel without further interaction with the user. Excel is trying to determine the format of the cell data automatically. This behaviour could lead to undesirable results. One small example:
Enter "3e0" in a Excel table and export it as CSV file. After you reimport the CSV the cell contains the value "3,00E+00". This is because Excel interprets "3e0" as a floating point number on import.
The Excel CSV import uses the language dependant list separator character, from the system settings to determine column boundaries.
While importing files with extension ".txt" Excel opens the Text import dialog. Within this dialog you can fine tune the import settings. You could use other column separator or field separator characters or set the data type per column manually. The following parameters could be used to import files exported by IXXAT canAnalyser3 Mini:

- Separated - characters separate fields
- Separator - semicolon (;), comma (,) or other, depends on the system language setting during export

- If columns contains hexadecimal numbers you should set the column type to "Text" or else specific hexadecimal numbers will be interpreted as floating point numbers.

Another characteristic with Excel is the Drag&Drop behaviour: If you Drag a CSV file onto an Excel instance, files with ".csv" extension are treated as if opened via file open. But if the file has the extension ".txt" the content of the file is copied line by line into the first column of the Excel sheet without opening the text import dialog.

A.1.3 Import in OpenOffice/LibreOffice

When importing files with extension ".csv" into OpenOffice the text import dialog is displayed automatically. Within this dialog you could set all necessary parameters:

- Separated - characters separate fields
- Separator - semicolon (;), comma (,) or other, depends on the system language setting during export
- If columns contains hexadecimal numbers you should set the column type to "Text" or else specific hexadecimal numbers will be interpreted as floating point numbers.

Files with extension ".txt" will be treated as text files and opened via OpenOffice Writer, if you have not selected the CSV import filter explicitely. Because of this Drag&Drop works only for files with extension ".csv".

Appendix B

Definitions

B.1 Definitions, acronyms, abbreviations

Bitrate Transmission rate in bits/sec. with which a bus is operated.

CAN Controller Area Network

CAN status In order not to block a CAN network with a defective node, CAN controllers have internal error counters. If these error counters exceed a certain limit, the status of the CAN controller changes to the warning level. If a further level is exceeded, the node is switched off by the bus (Bus off).

Data Frame Standard data telegram of the CAN bus. A data frame consists of an 11 or 29 bit wide identifier (COBID), a data field of between 0 and 8 bytes and protocol information such as RTR flag and DLC (data length code).

Database editor Application to create and alter databases on which the interpretation of layer-2 messages is based.

Error frame Special telegram for error signalling on the CAN bus

FIBEX Field Bus Exchange Format - Fibex is an XML exchange format proposed for data exchange between tools that deal with message-oriented bus communication systems.The FIBEX specification document is downloadable from the web page of ASAM e.V. (Association for Standardisation of Automation- and Measuring Systems) on `http://www.asam.net`.

Filter Module to select or exclude messages according to certain criteria for display or trace.

FlexRay FlexRay is a fast, deterministic and fault-tolerant bus system, developed for automotive use.

FlexRay CCM IXXAT PC-Interface for FlexRay and CAN

Online mode Recording or display of messages immediately after reception without further processing.

Remote frame CAN request telegram. Special telegram format without data field to request a data telegram

RTR RemoteTransmitRequest: The RTR-bit within a CAN message distinguishes between data telegrams and data request telegrams

Standard/Extended The CAN bus supports two message formats, which differ in the length of the identifier. Standard with 11-bit identifier and extended with 29-bit identifier.

Trace Recording of messages in a file

Trace file A recording carried out of layer-2 messages, which can be saved as a binary or text file, and which can then be evaluated

Trigger Event used to start/stop a recording (Trace).

TX-echo Mode in which the IXXAT canAnalyser3 Mini also receives messages which it has transmitted itself.

TX-passive Mode in which active access to the bus is prevented by hardware. Neither acknowledge nor errors can be terminated. The IXXAT canAnalyser3 Mini is only a listener.

VCI Universal CAN driver for all PC/CAN boards of IXXAT

Appendix C

Copyrights

C.1 Urheberrecht

C.2 Zusätzliche Urheberrechte

C.2.1 Dundas software

This software contains material that is © 1994-2000 DUNDAS SOFTWARE LTD., all rights reserved.

C.2.2 FatCow Web Hosting Free Icons

`http://www.fatcow.com/free-icons`
These icon sets are licensed under a Creative Commons Attribution 3.0 License (`http://creativecommons.org/licenses/by/4.0/legalcode`)
HMS Networks kindly thanks WebCow Web Hosting for providing such a neat, versatile, and comprehensive collection of icons for free.

C.2.3 Lua.org, PUC-Rio

License for Lua 5.0 and later versions
Copyright © 1994-2010 Lua.org, PUC-Rio.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,

ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.