Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

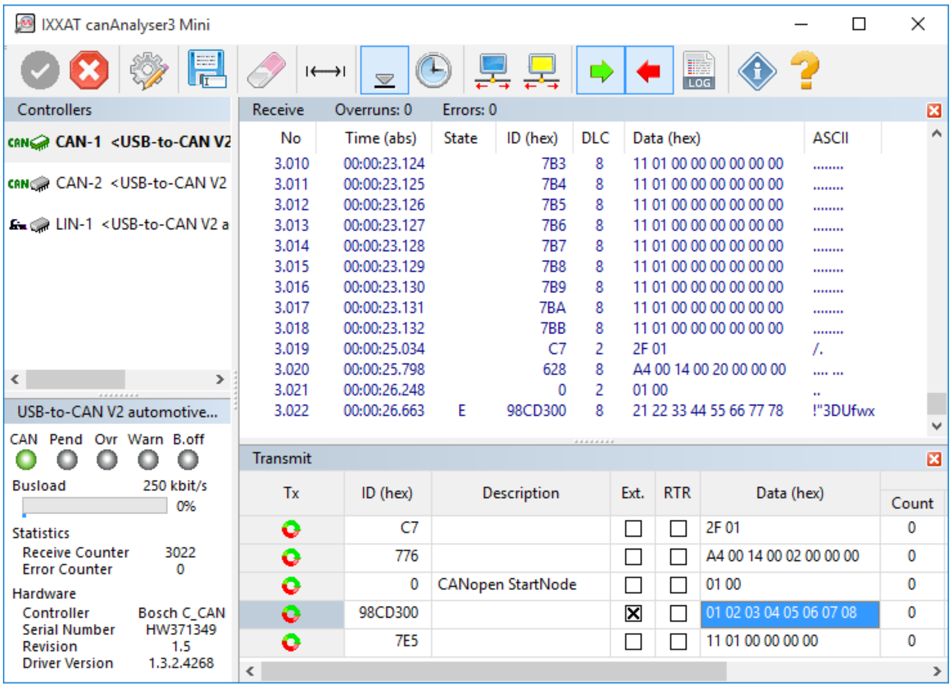
We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

IXXAT canAnalyser3 Mini

Busmonitoringtool for Windows



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

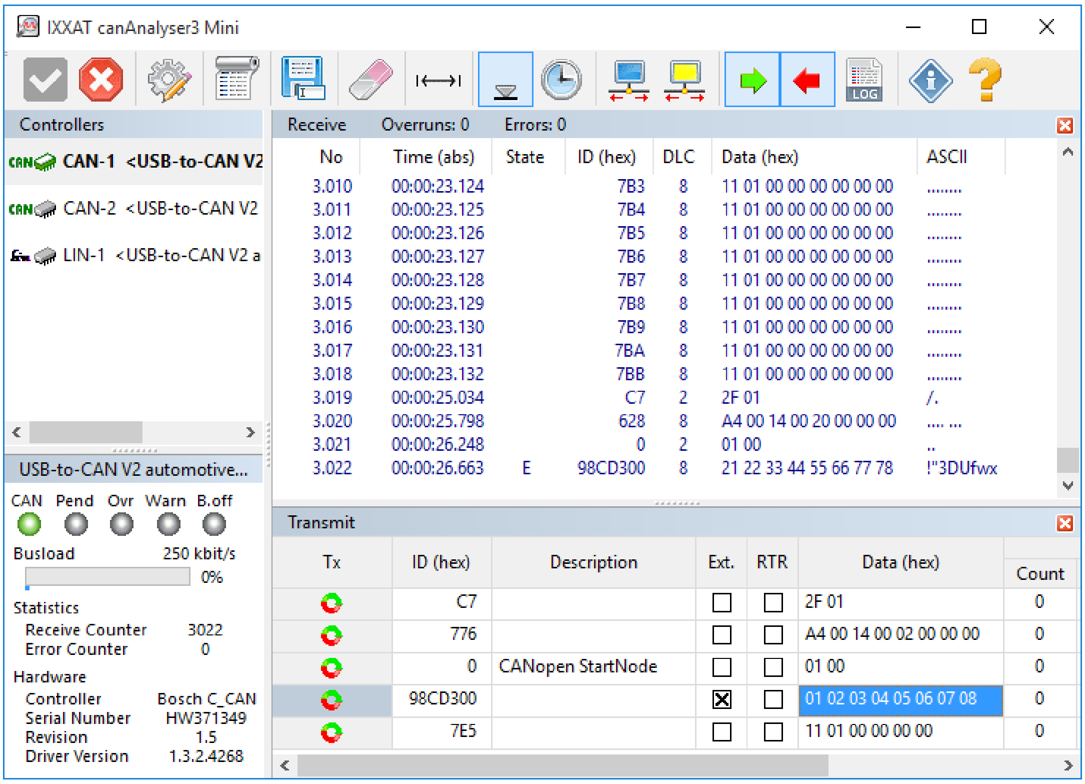


Figure 1.1: IXXAT canAnalyser3 Mini main window

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.

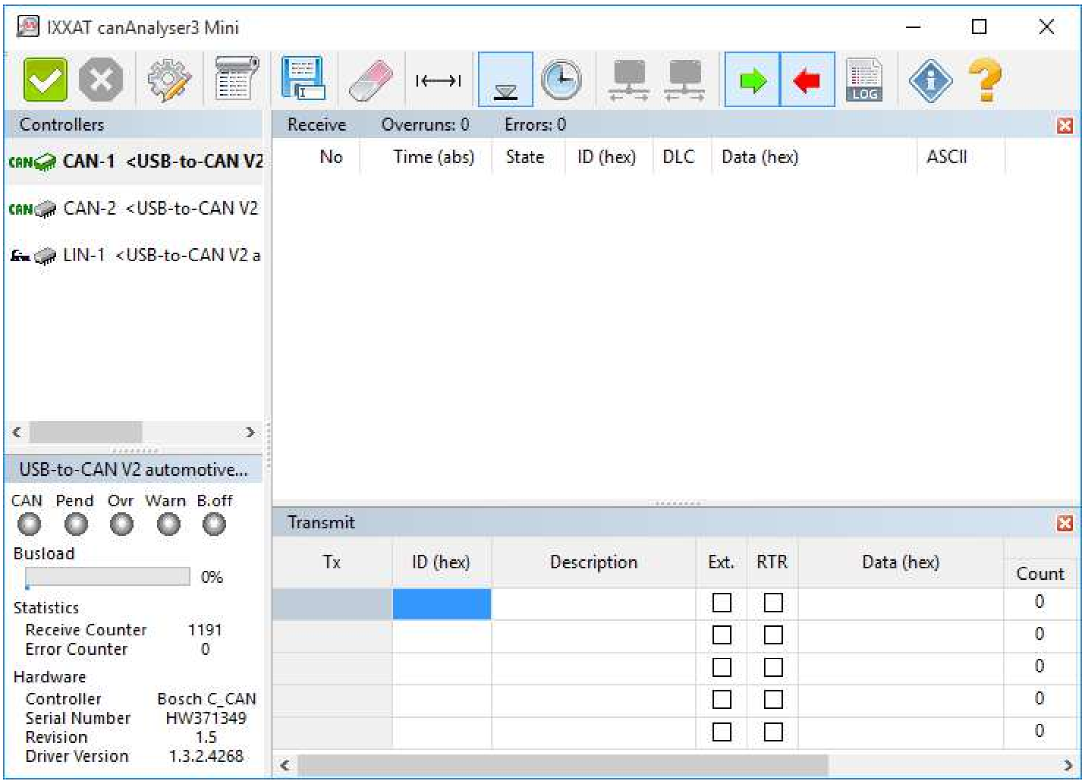


Figure 2.1: IXXAT canAnalyser3 Mini first start

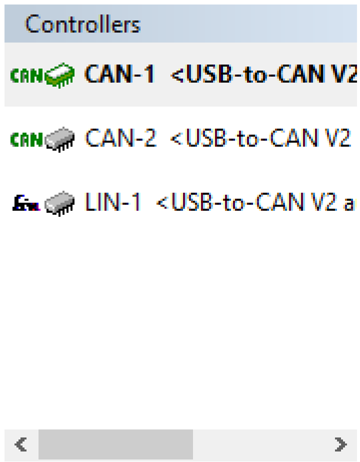


Figure 2.2: IXXAT canAnalyser3 Mini List of available Bus Controllers (top left)

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
	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.

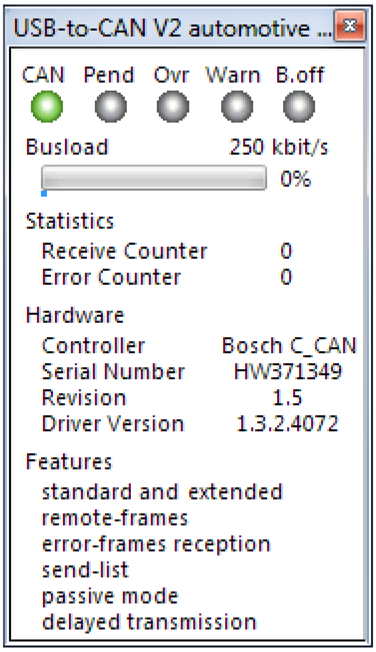


Figure 2.3: CAN / CAN-FD status view

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:

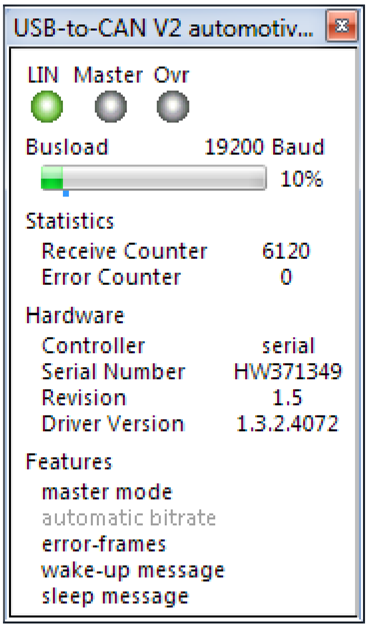
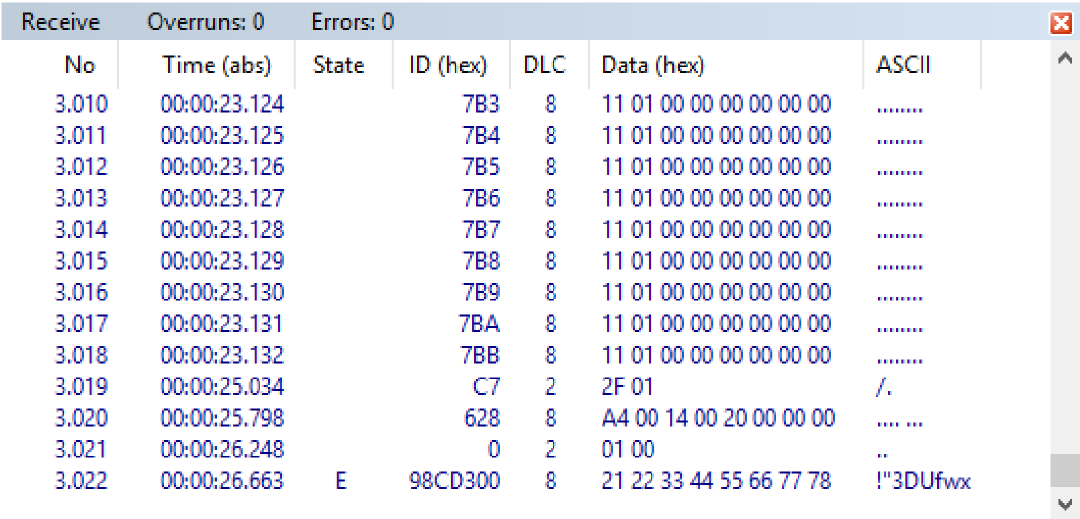


Figure 2.4: LIN status view

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

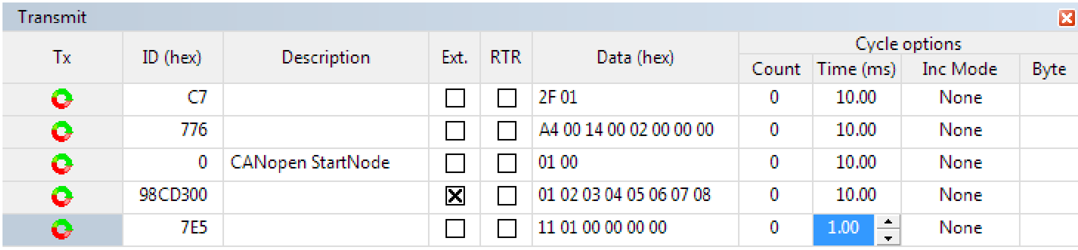
All messages are listed in the order of reception with the following information (Fig. 2.5):

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).

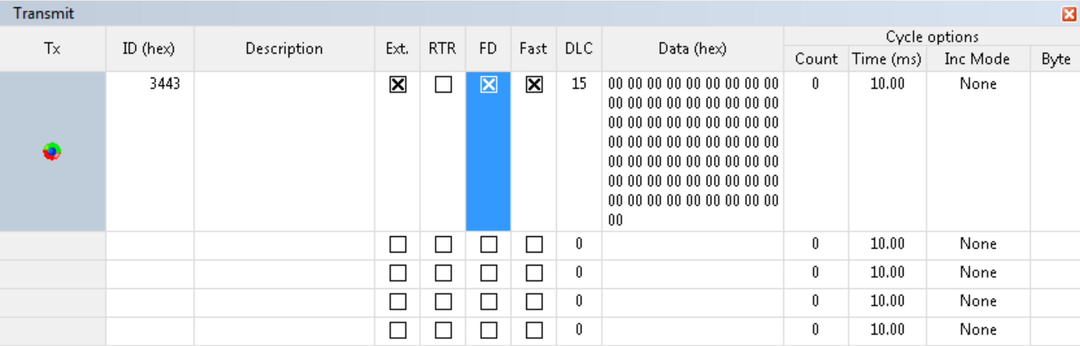


Figure 2.7: CAN-FD Transmit messages grid (bottom right) (instance)

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

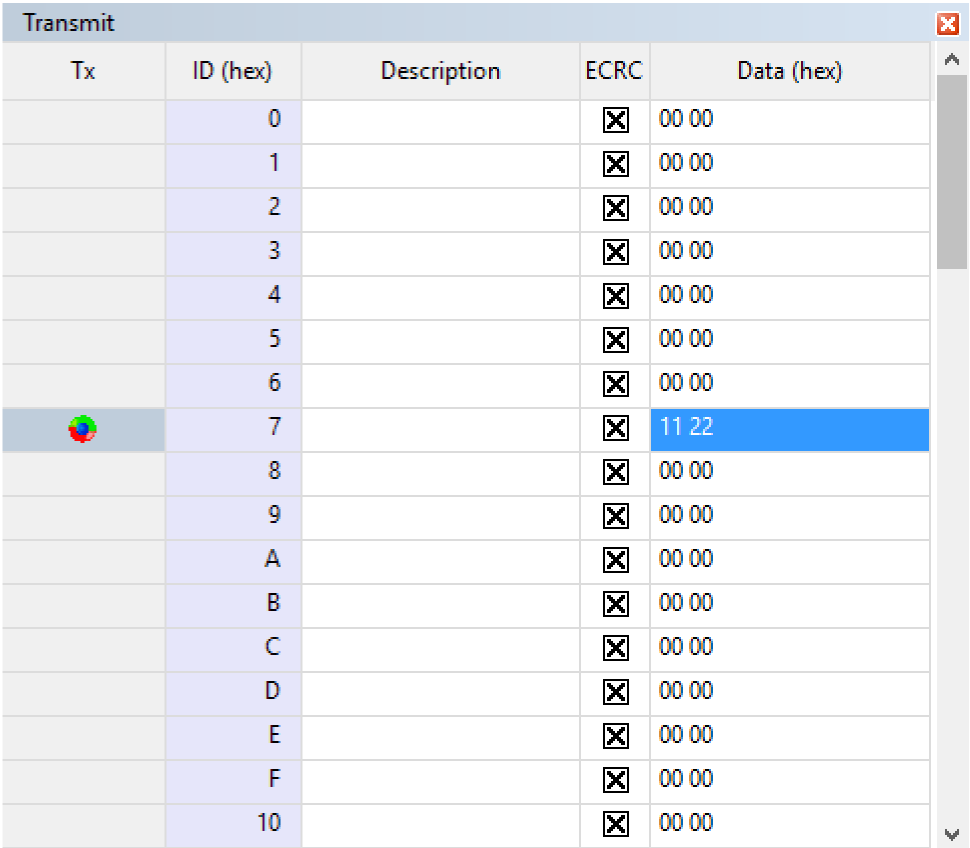


Figure 2.8: LIN Transmit messages grid (bottom right) - Slave mode (instance)

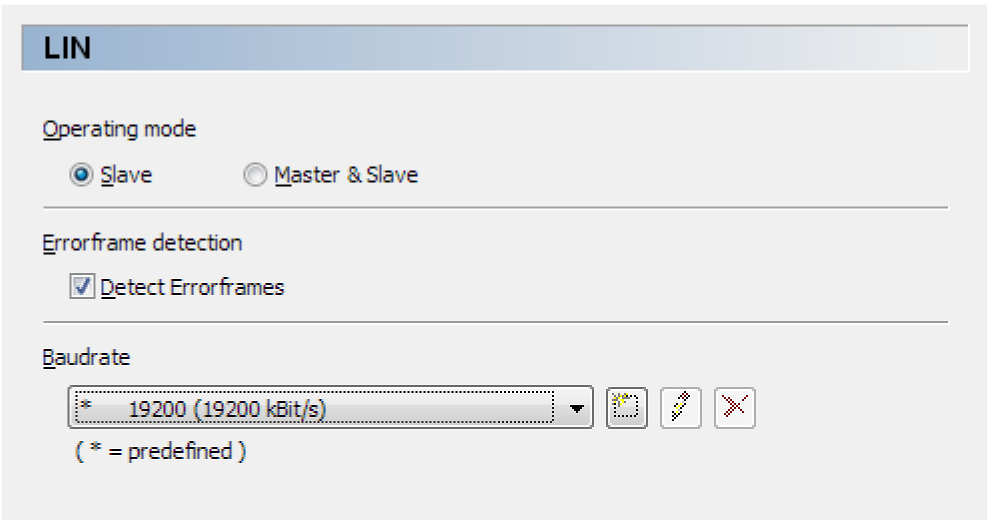


Figure 2.9: LIN settings

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.

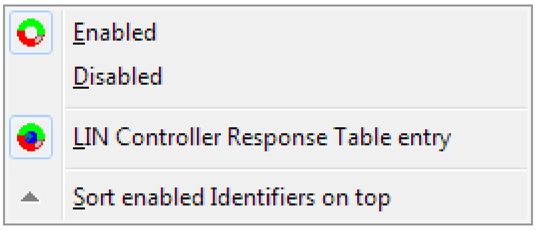


Figure 2.10: Context menu LIN (full)

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

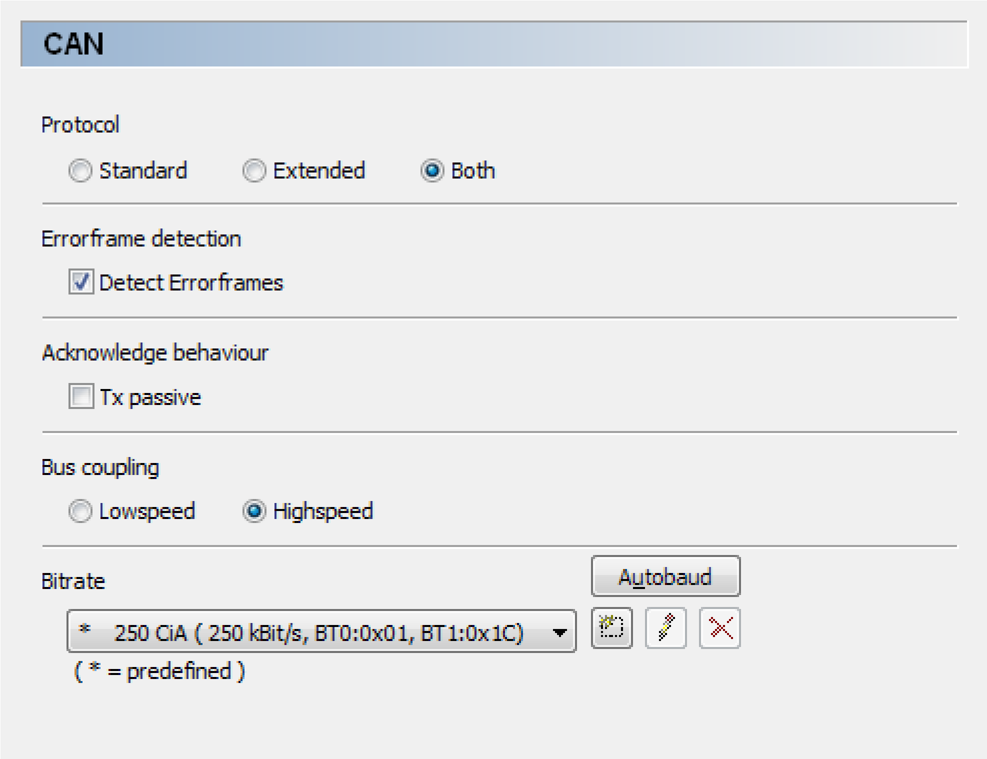


Figure 2.11: CAN Settings

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:

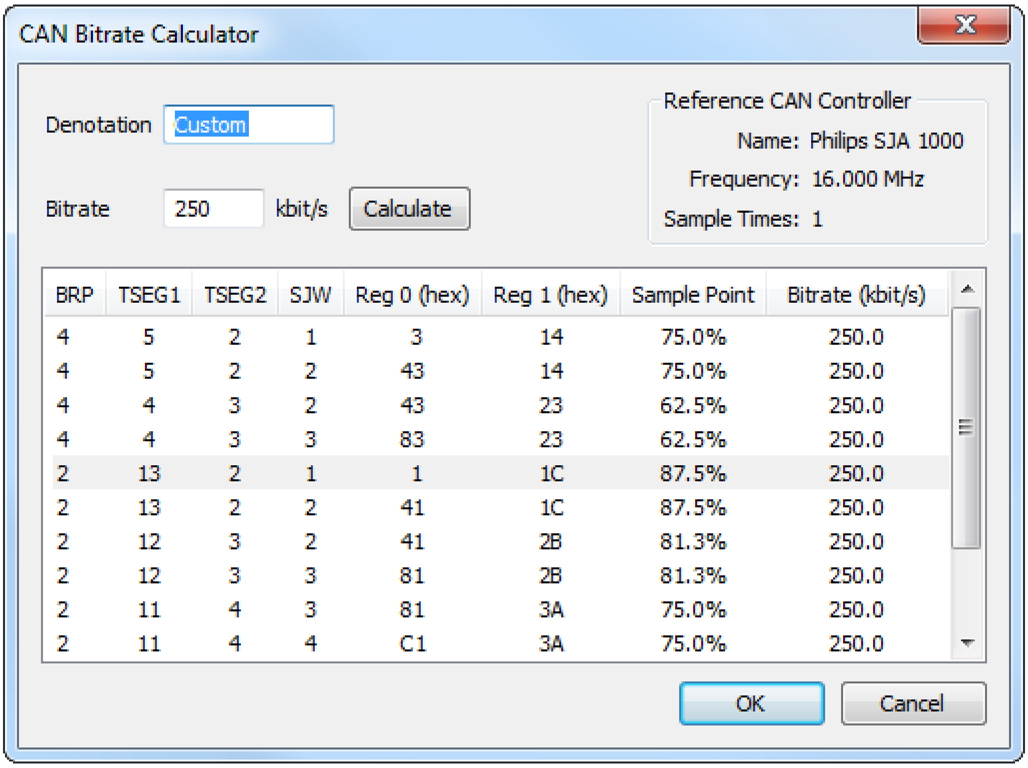


Figure 2.13: The CAN bitrate calculator

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

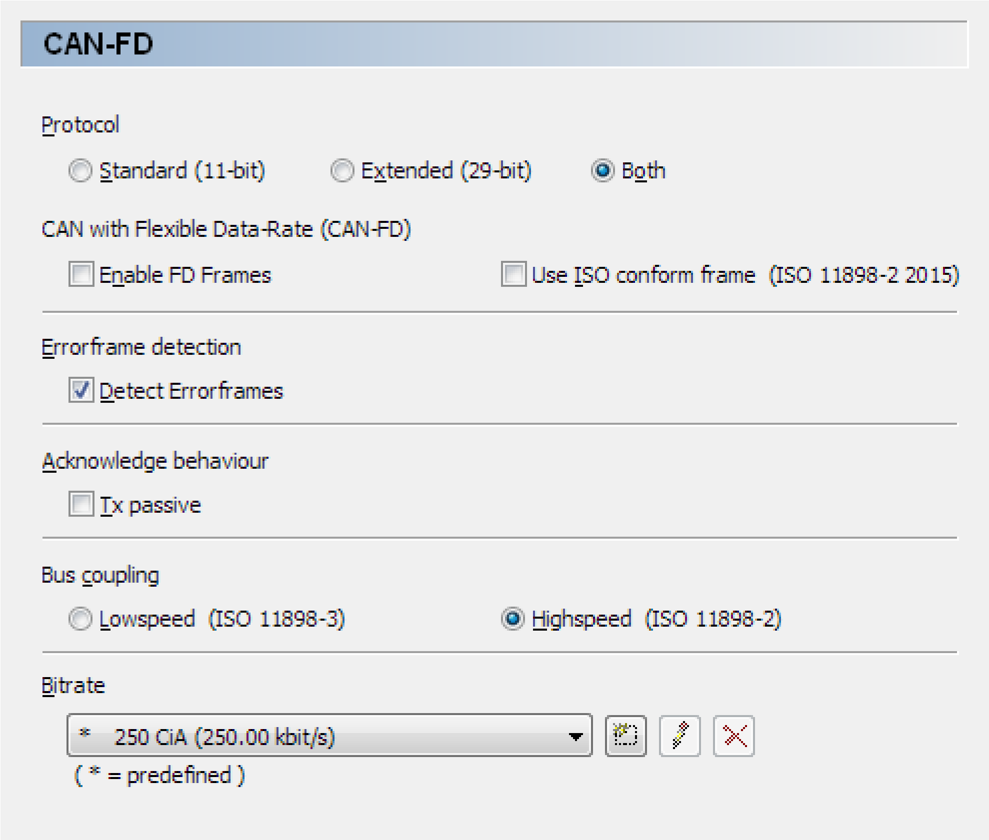


Figure 2.14: CAN-FD Settings

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).

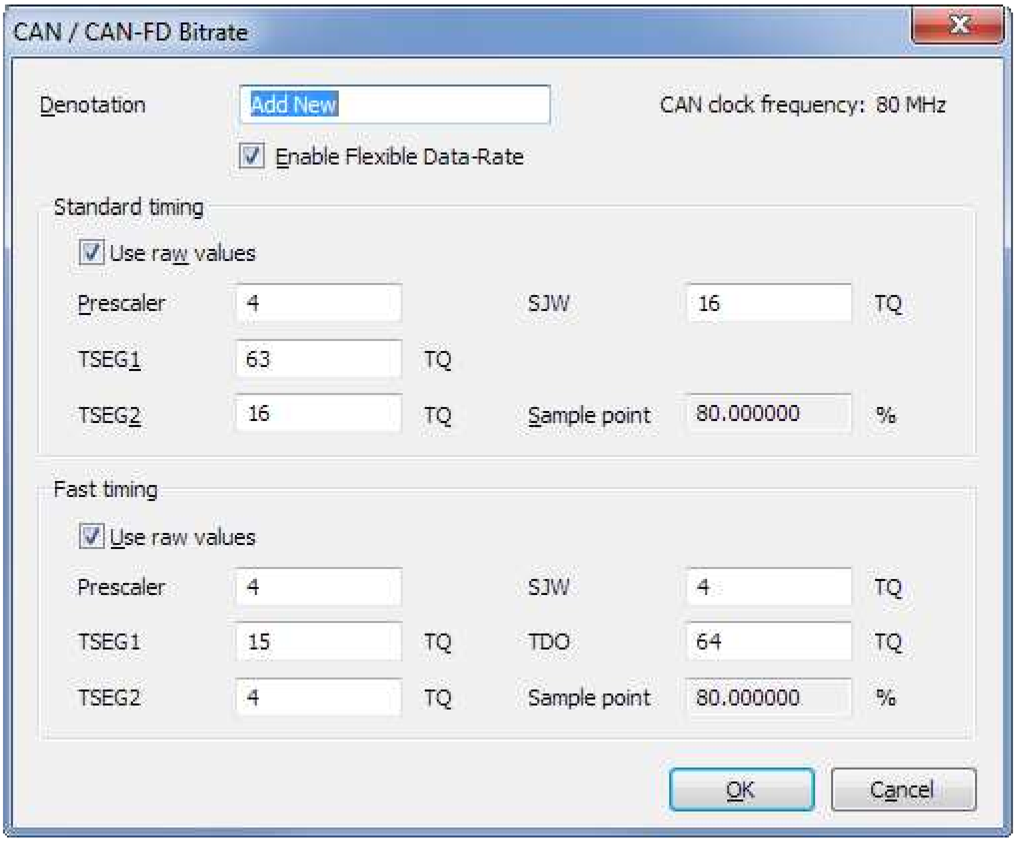


Figure 2.15: The CAN-FD bitrate dialog

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:

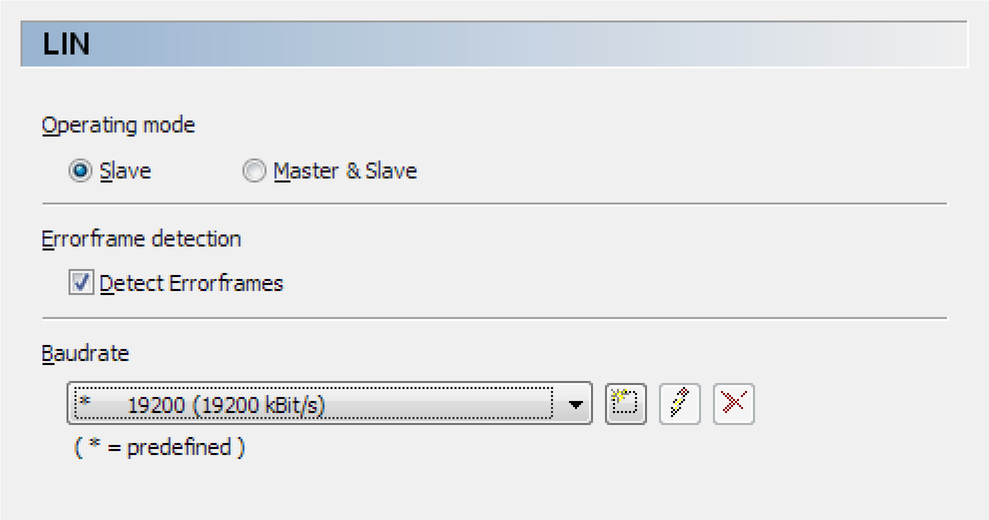


Figure 2.16: LIN Settings

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.

