CAN & LIN

Development Tools

Version 1.2

HW for Physical Bus Connection

PC SW Pre-Release Version for professional CAN & LIN Development

4 x HW (Trigger) Outputs to synchronize Bus Signals with external HW

Preliminary

CLDT1xx4 – series	:	CAN Bus
CLDTx1x4 – series	:	LIN Bus
CLDTxx14 – series	:	Single Wire CAN Bus (GM LAN)

Company Information :

JW Electronics
Heiligenbergerstr. 130
D-88682 Salem

1 Table of Contents

2 Revision

Version	Date	Comment
V1.0	28.09.2014	First Preliminary Release
V1.1	16.12.2014	Extended Value, Radio & Slide Controls available for restbus simulation
V1.2	10.03.2015	New Features : TX Sequence, Scope and RX-TX Automation

3 Warranty & Software Pre-Release - Version

Note that the current level of maturity for the CAN Box Software and the PC Software is in Pre-Release State (Test Version State).

All features described in this user manual are tested and are presumed to work correctly. However, not all features are tested under all thinkable test conditions and therefore some unexpected results may not be excluded completely.

JW Electronics would appreciate to get some customer feedback if any of the functions do not work as specified in this user manual or to get suggestions to improve CLDT- Box and / or software features.

All Pre-Release – Users have full upgrade privilege for future software releases.

A Flash Bootloader is integrated in the CAN Box Software, thereby CAN Box Software upgrades are feasible by the customer himself.
However, if the Flash Bootloader must be upgraded in order to use some new software features the CLDT Box must be sent back to JW Electronics. The new Flash Bootloader can only be programmed in the JWE labs.

Only shipment costs are at customer's own expense, the Flash Bootloader upgrade is for free.

JW Electronics takes no liability for correct user manual content or damages which follow from the use of this user manual or the use of the Pre-Release software versions.

JW Electronics reserves the right to change

- the contents of the user manual
- the CLDT Box software
- the PC software

without prior declaration at any time.

4 Overview CLDT Series – Main Features

4.1 Object & Main Features

As the CAN Bus and LIN Bus are more and more subject of development projects outside the automotive industries - and even become part of non-professional engineering - the need for CAN & LIN development tools will furthermore increase in the future.

The basic motivation for developping the CLDTxxxx – CAN & LIN Series was to create a low cost development tool, but, at the same time, not resign on the main functionalities which are essential for a professional HW and SW development.

Therefore features such as restbus simulation, dbc support, scope functionality, RX/TX Automation and easy synchronization between CAN Bus Data and external hardware were inherent parts of the design phase for these tools.

CLDT Box main features

- ✓ professional low cost HS CAN, LIN & Single Wire CAN Bus Development Tool
- ✓ CAN Bus Baudrates up to 1 MBit/s
- ✓ Single Wire CAN Support for Normal Operating Mode with Baudrates up to 40kBit/s
- ✓ self-explanatory & intuitive SW Tool and therefore very easy to handle
- ✓ Restbus Simulation Tool
- ✓ Scope Functionality to monitor up to 8 CAN signals in real time
- ✓ RX/TX Automation in dll Mode with synchronization possibility between CAN Messages and 4 physical digital HW Outputs
- ✓ TX Sequence Mode to apply pre-defined CAN Sequences
- ✓ Little Endian & Big Endian data format support
- ✓ dbc - data base support
- ✓ Bus Trace with different displaying modes
- ✓ save bus data
- ✓ RX bus data synchronization with 4 x HW Trigger Outputs
- ✓ TX bus data synchronization with 4 x HW Signal Outputs
- ✓ future PC SW upgrades included
- ✓ integrated Flash Bootloader for CAN Box upgrades

- ✓ future CAN Box SW upgrades included (exception see chapter 'Warranty')
- ✓ power supply by PC USB Port
- ✓ HS CAN interface compatible to other CAN Tools on the market

4.2 CLDT1xx4 – Series : High Speed CAN Bus Support

Version	HS CAN	LIN	SW CAN	HW Output	Availability
CLDT1004	1	0	0	4	11 / 2014
CLDT1104	1	1	0	4	likely Q2/2015

4.3 CLDTx1x4 – Series : LIN Bus Support

Version	HS CAN	LIN	SW CAN	HW Output	Availability
CLDT1104	1	1	0	4	likely Q2/2015
CLDT0114	0	1	1	4	likely Q2/2015

4.4 CLDTxx14 – Series : Single Wire CAN Bus Support

Version	HS CAN	LIN	SW CAN	HW Output	Availability
CLDT0014	0	0	1	4	02 / 2015
CLDT0114	0	1	1	4	likely Q2/2015

Important Note : **Single Wire CAN Bus is only supported for Normal Operating Mode with Baudrates up to 40 kBit/s.**

High Speed Programming Mode (e.g. 83 kBit/s/+12V) is not supported by the CLDT Box.

5 Installation Guide

5.1 Minimum Requirements

- CPU : Pentium 4 or higher
- RAM : 1GB or more

5.2 Supported Windows Operating Systems

- Windows XP SP3 : PC SW V1.0 is tested with Windows XP. For future SW updates no testing under Windows XP is planned

- Windows Vista SP1 : supported
- Windows 7 – 32bit : supported
- Windows 7 – 64 bit : supported
- Windows 8 – 32 bit : supported
- Windows 8 – 64 bit : supported

5.3 Timing Requirements under Windows Operating Systems

Note that most desktop PCs block CPU execution for specific time slots due to internal power management requirements. This has impact on the accuracy of timing generations in Windows operating systems. In applications with increased timing requirements these power management adjustments should be deactivated.

5.4 VCP Driver Installation

The USB Connection between PC and CLDT Box is realized via a virtual COM port device designed by Future Technology Devices International Ltd. (FTDI).
The setup executable to install the VCP driver is located on the setup disc in the directory :driver/DriverSetup/CDM v2.10.00 WHQL Certified.exe.
The latest driver version may also be downloaded directly from FTDI's homepage *http://www.ftdichip.com/Drivers/VCP.htm* .
The driver installation guides are available on the setup disc in the directory :driver/InstallationGuides/ .
The latest version of documentation is also available on FTDI's homepage *http://www.ftdichip.com/Support/Documents/InstallGuides.htm*

Use the setup executable and follow the instructions in the Installation Guides.

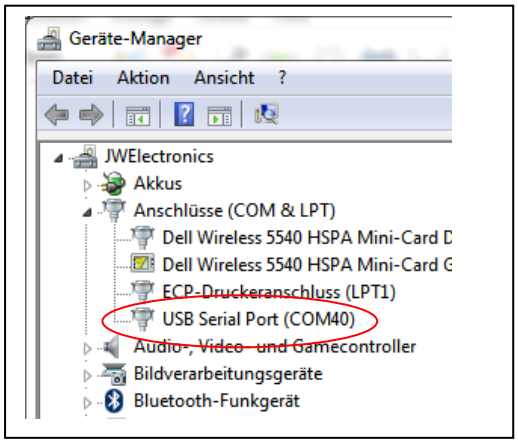
Note : For Windows 7 & Windows 8 operating systems the setup-exe must be executed with administrator rights. Therefore use a right mouse click on the setup-exe and choose 'execute as administrator'

5.4.1 Important : Correct Setting for VCP Latency Timer

The VCP latency timer is a form of time-out mechanism for the read buffer of FTDI devices. Its default value is 16ms.

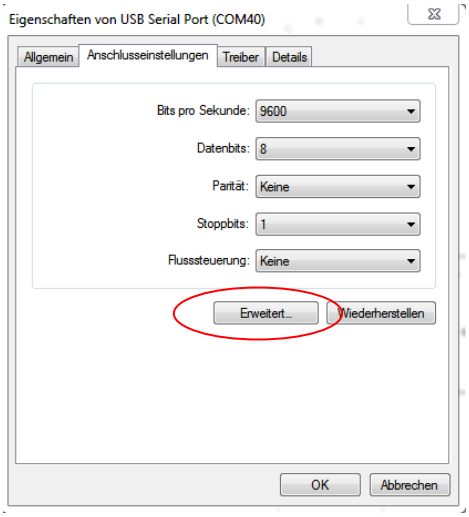
For correct CLDT SW functionality under different CAN bus charge scenarios this value should be changed to 5ms. In applications with increased timing requirements the time-out should even be set to a value of 2ms or 1ms.

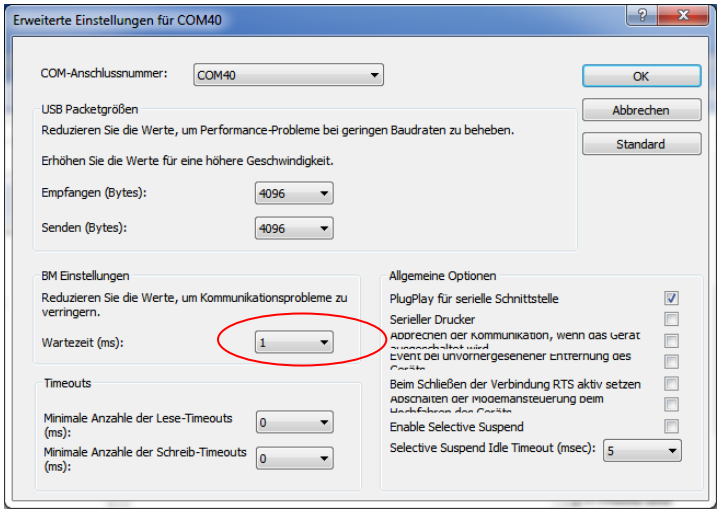
Change VCP Latency Timer in Windows Device Manager :



After correct VCP Driver Installation connect the CLDT Box to a USB Port and wait until the green LED appears on the box. Then open the Device Manager, right-click the dedicated COM Port and then click Properties.

On the Port Settings Tab click ‘Advanced Settings’ and change the Latency time-out to a value of 5ms or lower.





5.5 SW Installation

First install the VCP USB driver before starting the installation of the CLDT-Box PC Software (refer to the chapter *'VCP Driver Installation'*)

After the VCP USB driver has been successfully installed start the setup.exe in the main directory of the setup disc and follow the instructions. The installer will guide you through the installation.

Before installing the software from the CD check, if a more recent version of the installer is available at

http://www.jwe-electronics.com/mp1/downloads.php

5.5.1 Installation under Windows Vista, Windows 7 or Windows 8

Note : For Windows Vista, Windows 7 or Windows 8 operating systems the setup-exe must be executed with administrator rights. Therefore use a right mouse click on the setup-exe and choose 'execute as administrator'

Note : For Windows Vista, Windows 7 and Windows 8 operating systems the CLDT software **should not be installed** in the folders

- C:\Program Files\
- C:\Program Files (x86)\

The CLDT software must have write-access to some subfolders of its installation folder. Therefore, if the user is not logged in as administrator and the software is installed in the 'Program Files' folder, the software might not work correctly.

5.6 Quick Check

After successful VCP Driver and CLDT-Box Software Installation connect the CLDT Box to the PC and wait until the PC has automatically detected and installed the new Hardware. Depending on your operating system this may take up to 2 minutes.

When the green LED on the CLDT Box is lighting in continuous mode the new Hardware has been installed and is working correctly.

5.7 PC SW Upgrade

During Beta-Testing phase bugfixing, as well as the implementation of new software features, is done periodically.

Therefore new PC software releases can be downloaded at

http://www.jwe-electronics.com/mp1/downloads.php

5.7.1 Install a PC SW Upgrade

1) Download the new CLDT.exe - file

2) Copy the new CLDT.exe in the installation folder \JWE\

3) Delete the can_oszi.ini in the folder \JWE\ini\can_oszi.ini

4) Start the new CLDT.exe

5.8 CLDT Box SW Upgrade

During Beta-Testing phase it should be cyclically checked, if a new CLDT Box Software is available at

http://www.jwe-electronics.com/mp1/downloads.php

During Beta-Testing phase bugfixing, as well as the implementation of new software features, is done periodically.

1) Download the new CLDT-Box_Vx_y.cos - file

2) Copy the new .cos - file in the folder \JWE\download\

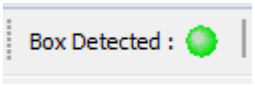
3) connect the CLDT-Box to the PC

4) start the CLDT.exe

5) follow the download procedure as described in the next chapters

5.8.1 CLDT Box SW Upgrade

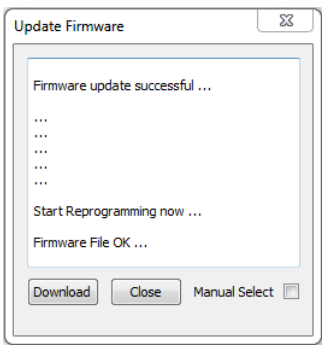
- Start the CLDT.exe



the green LED ‘Box detected’ should appear

- Choose → Firmware → Download in the main menu

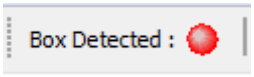
- Click the ‘download’ – button and choose the .cos-file



5.8.2 CLDT Box SW Upgrade : No Software installed on the CLDT Box

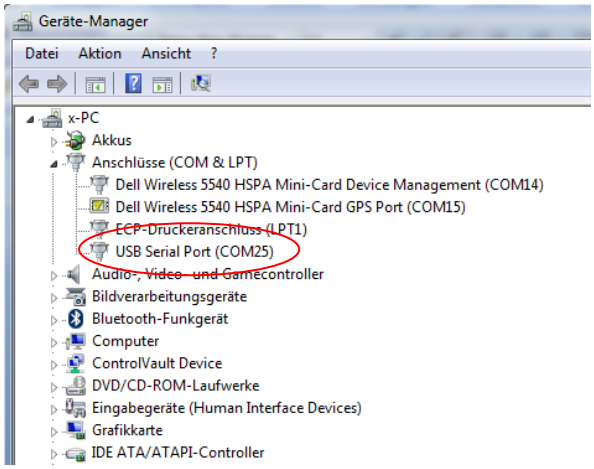
If the CLDT Box is connected to the PC and the **green LED on the CLDT-Box is blinking**, no valid software version is loaded on the Box.

- Start the CLDT.exe

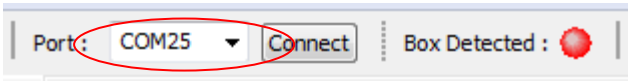


the red LED ‘Box not detected’ appears

- If the connection is not established, the correct virtual COM port, that is dedicated to the CLDT-Box must be chosen manually in the Windows Device Manager
  → open the device manager and chose the dedicated COM port as shown below :

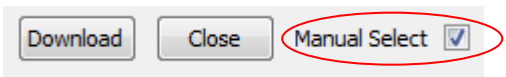


- Set the correct Com Port in the toolbar :



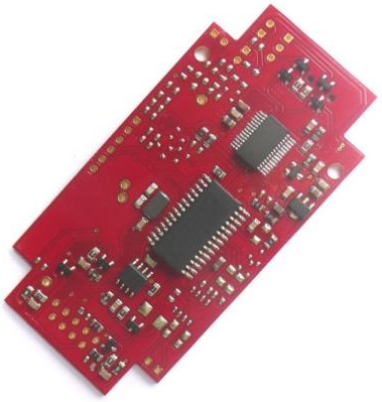
- Choose → Firmware → Download in the main menu

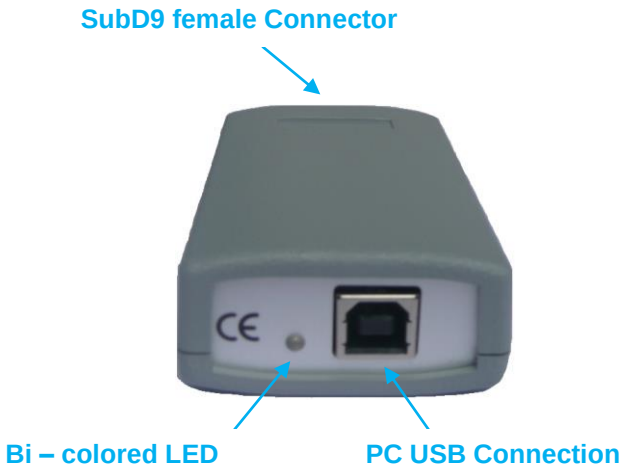
- Set the Manual Select check box



- Click the ‘download’ – button and choose the .cos-file

6 CLDT Box : Hardware Description





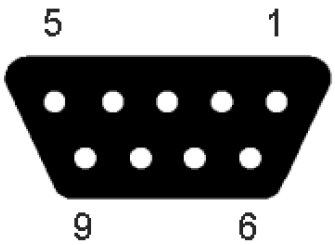
6.1 PC Connection

The CLDT Box is connected to the PC via a USB 2.0 type A to type B – Cable.

+5V supply voltage is provided from PC USB port with a maximum supply current of 200 mA.

6.2 SubD 9 Female Connector Pinout Description

HW (Trigger) Outputs and Bus Connections are realized via a SubD9 female connector with the following pinout :



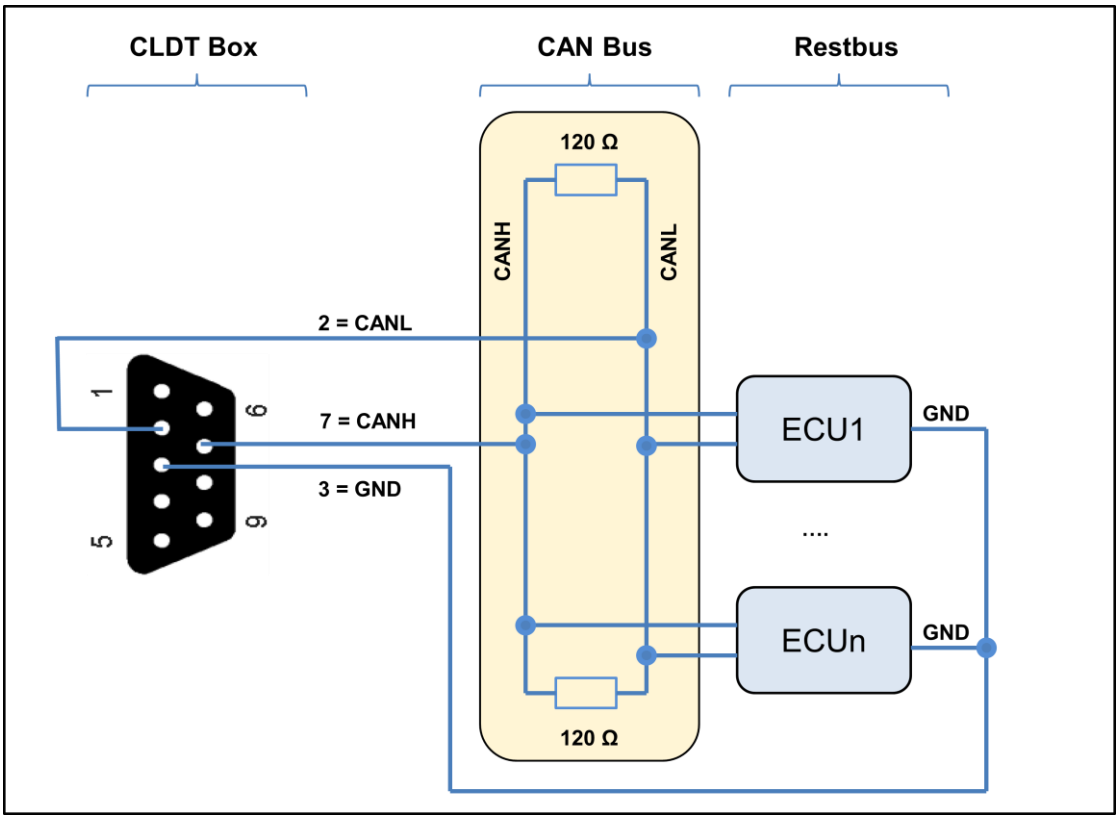
1 :	(Trigger) Output 2	
2 :	HS CANL	(CLDT1xxx series)
3 :	GND	
4 :	LIN	(CLDTx1xx series)
5 :	(Trigger) Output 3	
6 :	(Trigger) Output 1	
7 :	HS CANH	(CLDT1xxx series)
8 :	Single Wire CAN	(CLDTxx1x series)
9 :	(Trigger) Output 4	

6.3 Bi – colored LED

The following table describes the lighting conditions of the bi colored LED (orange / green) :

Color	Mode	Description
green	continuous	CLDT Box ready for use
green	fast flashing	No SW loaded : CLDT Box must be reprogrammed
green	slow flashing	SW Programming ongoing
orange	slow flashing	Measurement ongoing
orange	continuous	Measurement pause

6.4 CAN Bus Connection (CLDT1xx4 – Series)

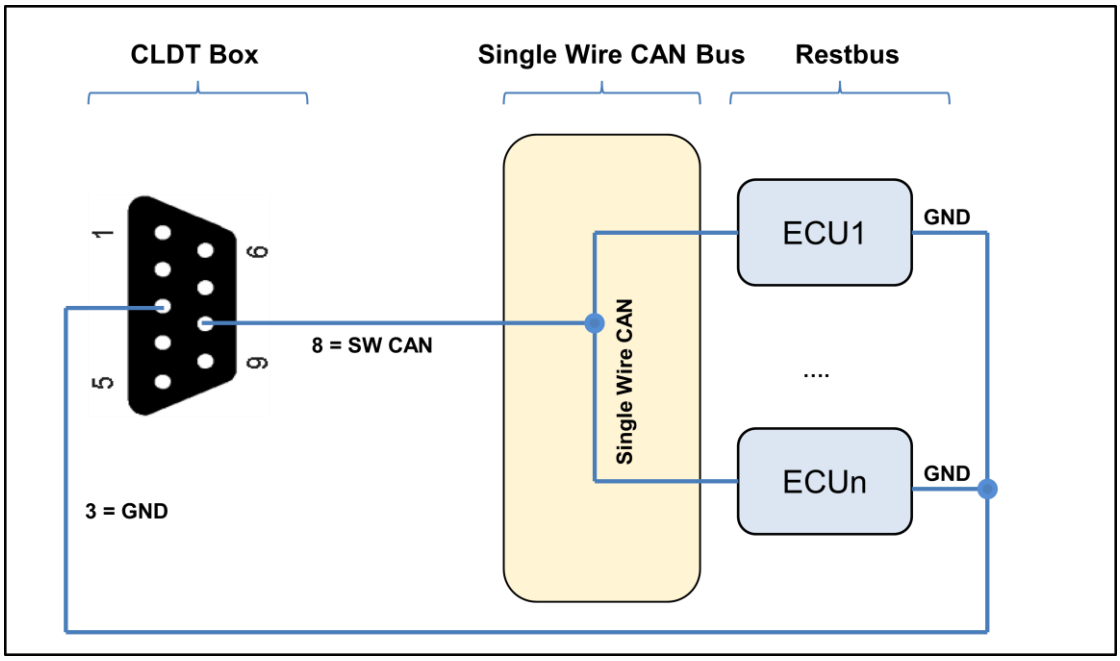


- Connection lines between CLDT Box and CAN Bus as short as possible
- 120 Ω terminating resistors to avoid reflections at cable ends

6.5 LIN Bus Connection (CLDTx1x4 – Series)

Chapter under Revision

6.6 Single Wire CAN Bus Connection (CLDTxx14 – Series)



- Connection line length less critical due to lower Baudrate (up to 40 kBit/s) for Single Wire CAN Bus

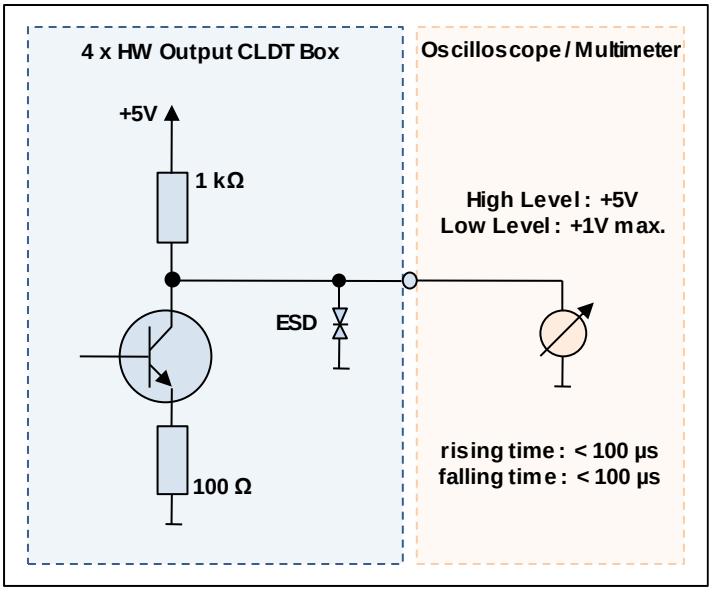
Important Note : **Single Wire CAN Bus is only supported for Normal Operating Mode with Baudrates up to 40 kBit/s.**

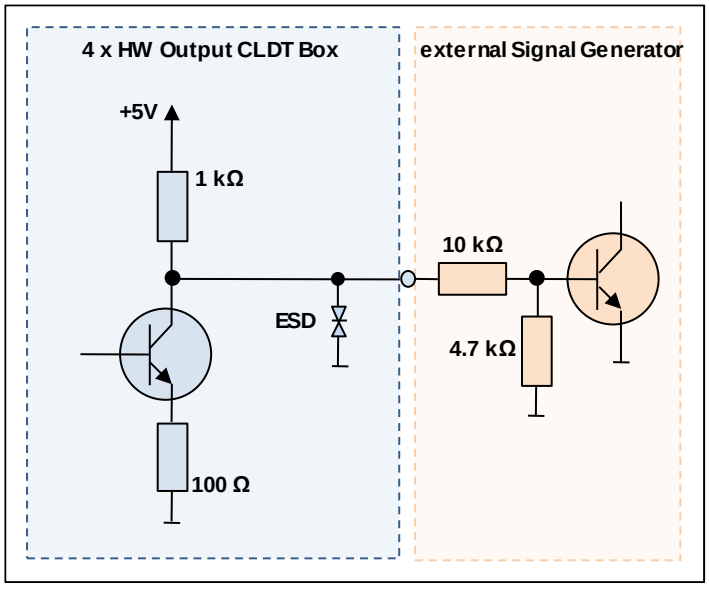
High Speed Programming Mode (e.g. 83 kBit/s/+12V) is not supported by the CLDT Box.

6.7 HW (Trigger) Output Interface

6.7.1 Output Signal Monitoring

Recommended connection method for HW Output Signal Monitoring by using voltage monitoring instruments (e.g. multimeter or oscilloscope) or external signal generators :





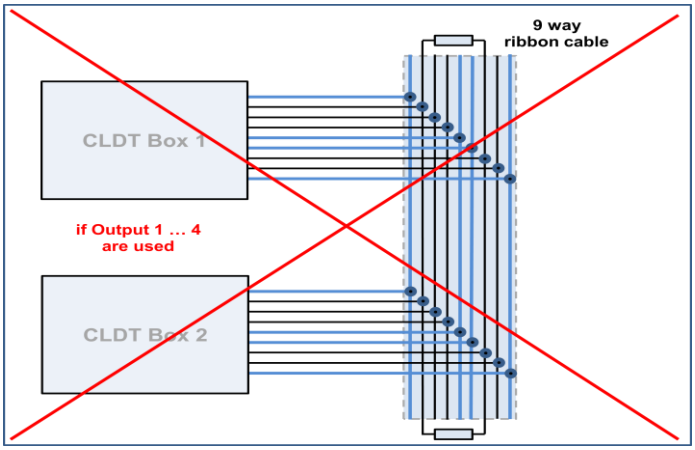
6.7.2 About CLDT Box HW Outputs and CAN Bus Connection

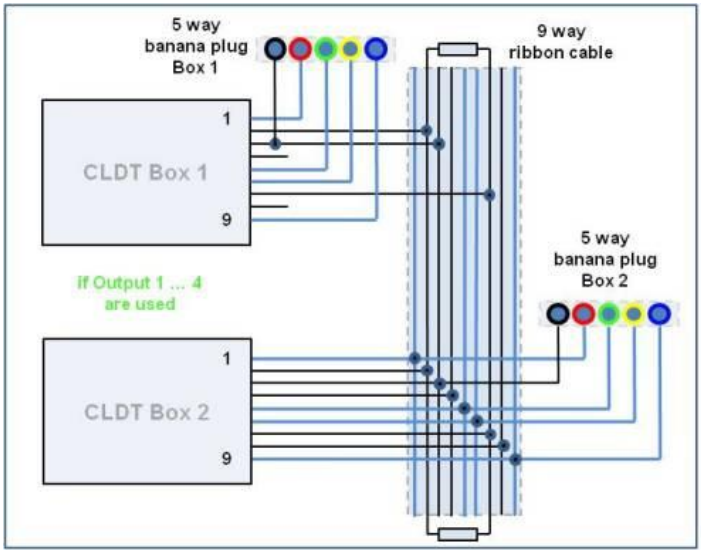
During development phases the real CAN Bus is usually replaced by using a 9-way ribbon cable as physical bus and by adding 2 terminating resistors at the end of lines (see chapter *'Hardware Accessory Kits'*).

If the CLDT Box is directly connected to all 9 pins of the 9-way ribbon cable via a 9 – way male/male adapter please consider, that the 4 HW Outputs may be connected to other signals.

If, for example, 2 CLDT Boxes are connected together at the same physical bus respectively with a 9 – way adapter, or if the ribbon cable lines 1, 5, 6 or 9 are connected to other signals from other ECU's, this could lead to unpredictable or casual results concerning the voltage levels on these lines.

In this case the CAN Bus Connection should be done (at least for one of the two CLDT Boxes) via an adapter type exemplary shown in the chapter *'Hardware Accessory Kits'*





7 Hardware Accessory Kits

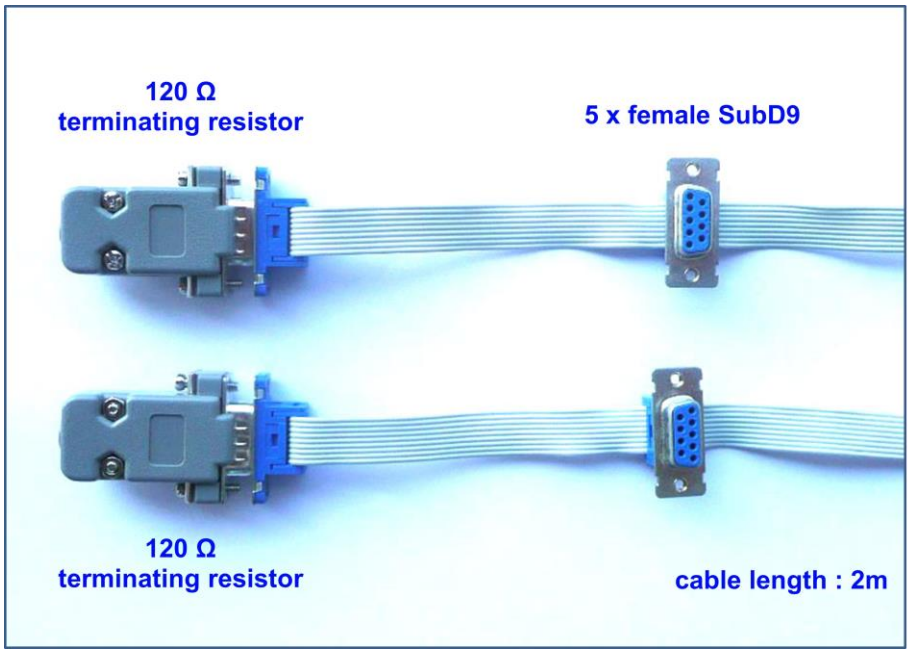
7.1 9 – way Ribbon Cable for CLDT Box to ECU's CAN Bus Connection

Including :

- 5 x SubD9 female connector mounted on 2 meter ribbon cable
- 2 x 120 Ω terminating resistor mounted in a SubD9 male & female plugs
- 1 x SubD9 female connector to plug on terminating resistor 1
- 1 x SubD9 male connector to plug on terminating resistor 2

- Cable extension possible due to SubD9 male / female combination at the end of lines

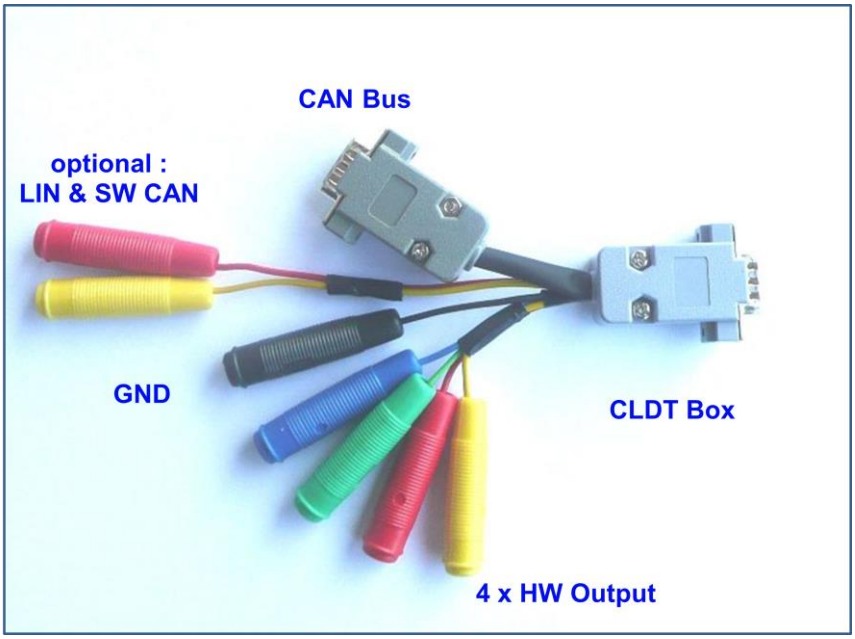
- Terminating resistors between lines 2 & 7



7.2 9 – way SubD9 CLDT to CAN Bus Connector

Including :

- 1 x SubD9 male connector CLDT Box Side
- 1 x SubD9 male connector Can Bus Side
- 5 x or 7 x banana female plugs CLDT Box Side

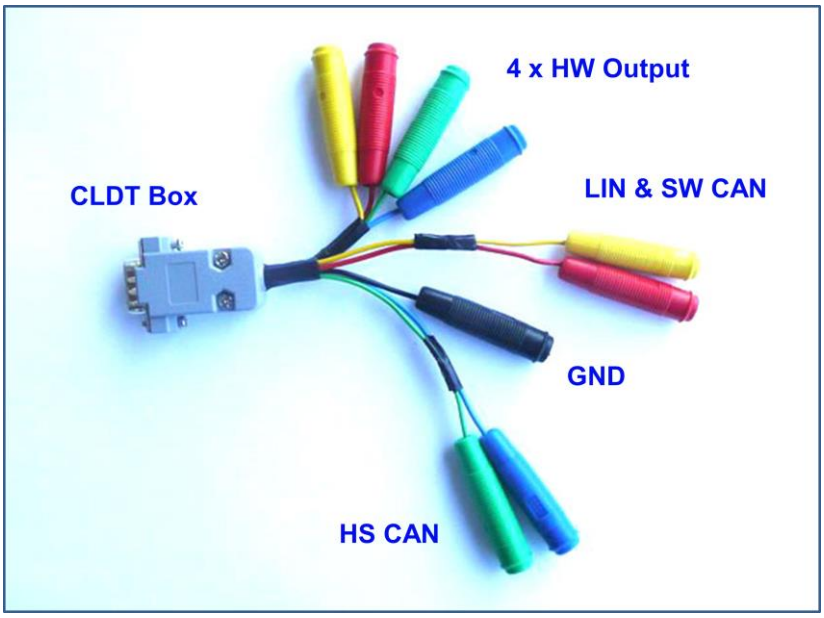


SubD9-1	Banana Plug Color	SubD9-2	Signal Description	Functional Group
3	black		GND	GND
3 2 7	-	3	GND	HS CAN
	-	2	HS CANL	
	-	7	HS CANH	
4 8	red (optional)		LIN	LIN / SW CAN
	yellow (optional)		Single Wire CAN	
1 5 6 9	red		(Trigger) Output 2	Output
	green		(Trigger) Output 3	
	yellow		(Trigger) Output 1	
	blue		(Trigger) Output 4	

7.3 9 – way SubD9 Ribbon Cable to Banana Plug Adapter

Including :

- 1 x SubD9 male connector Can Bus Side
- 9 x banana female plugs CLDT Box Side



SubD 9 Pin Nbr	Banana Plug Color	Signal Description	Functional Group
3	black	GND	GND
2 7	blue	HS CANL	HS CAN
	green	HS CANH	
4 8	red	LIN	LIN / SW CAN
	yellow	Single Wire CAN	
1 5 6 9	red	(Trigger) Output 2	Output
	green	(Trigger) Output 3	
	yellow	(Trigger) Output 1	
	blue	(Trigger) Output 4	

7.4 CLDT Box (or Ribbon Cable) to LIN & SW CAN Banana Plug Adapter

See picture in *chapter 7.3* : with only 3 banana female plugs.

Including :

- 1 x SubD9 male connector Can Bus Side
- 3 x banana female plugs (LIN, SW CAN, GND)

SubD 9 Pin Nbr	Banana Plug Color	Signal Description	Functional Group
3	black	GND	GND
4 8	red	LIN	LIN / SW CAN
	yellow	Single Wire CAN	

7.5 CLDT Box (or Ribbon Cable) to CAN Banana Plug Adapter

See picture in *chapter 7.3* : with only 3 banana female plugs

Including :

- 1 x SubD9 male connector Can Bus Side
- 3 x banana female plugs (CANH, CANL, GND)

SubD 9 Pin Nbr	Banana Plug Color	Signal Description	Functional Group
3	black	GND	GND
2 7	blue	HS CANL	HS CAN
	green	HS CANH	

7.6 CLDT Box (or Ribbon Cable) to 5 x Output Banana Plug Adapter

See picture in *chapter 7.3* : with only 5 banana female plugs

Including :

- 1 x SubD9 male connector Can Bus Side
- 5 x banana female plugs (4 x Output, 1 x GND)

SubD 9 Pin Nbr	Banana Plug Color	Signal Description	Functional Group
3	black	GND	GND
1 5 6 9	red	(Trigger) Output 2	Output
	green	(Trigger) Output 3	
	yellow	(Trigger) Output 1	
	blue	(Trigger) Output 4	

- 1 x SubD9 male connector Can Bus Side
- 5 x banana female plugs (4 x Output, 1 x GND)

7.7 OBDII male Connector to SubD9 male Connector

See picture in *chapter 7.8* : with only 1 SubD 9 male connector

Including :

- 1 x OBDII Diagnosis male Connector Car Side
- 1 x SubD9 male connector CLDT Box Side

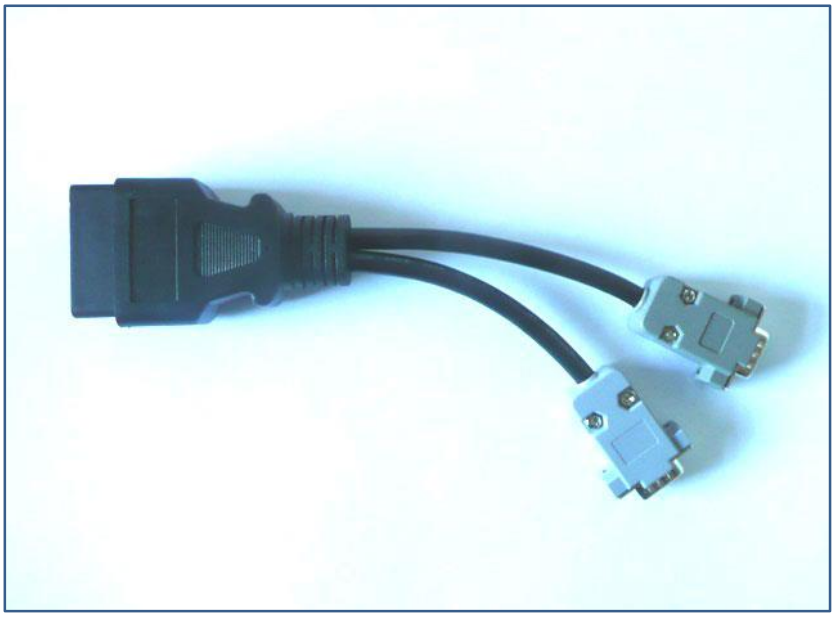
- **Pinning CLDT Box Side and OBDII Side on customers request**

7.8 OBDII male Connector to 2 x SubD9 male Connector

Including :

- 1 x OBDII Diagnosis male Connector Car Side
- 2 x SubD9 male connectors CLDT Box Side

- **Pinning CLDT Box Side and OBDII Side on customers request**



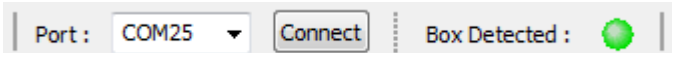
8 CLDT Box : Software Description CAN & Single Wire CAN

8.1 Overview – Main Features

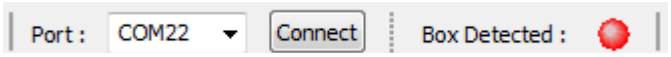
- CAN Bus Baudrates up to 1 MBit/s
- Single Wire CAN Bus Baudrates up to 40 kBit/s
- CAN Data RX Trace Window
- CAN Data RX Scroll Window
- Restbus Simulation Window with dedicated and configurable RX- and TX – Simulation Windows
- Scope Window to monitor up to 8 CAN Signals in real time
- RX / TX Automation in dll Mode with the synchronization possibility between CAN Bus Messages and 4 digital HW Outputs
- TX Sequence mode to load pre defined CAN Sequences
- Little Endian & Big Endian Support in Simulation Mode and in Trace Mode
- Save CAN RX & TX Trace Data
- CAN Data Base Support with RX Trace in Tree Data Structure
- In Trigger Mode : 4 x configurable HW Trigger Outputs to trigger on CAN Messages or CAN Message Contents
- In non-Trigger Mode : 4 x configurable HW Outputs to easily synchronize CAN TX messages with digital Signal Outputs
- Free Firmware upgrade via internal Flash Bootloader

8.2 CLDT Box Connection

If the CLDT Box is connected via the USB cable to the PC **and the green LED on the Box appears,** the connection between PC Software and CLDT Box is done automatically at the program start. The software detects the CLDT Box and indicates the virtual COM port that is dedicated to the Box.

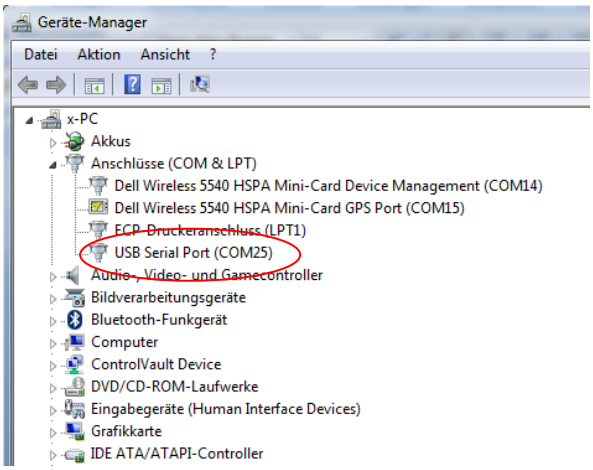


If the CLDT Box is not detected automatically or if the connection was lost due to any reason the PC LED state indicates 'not connected state' with a red color :



In this case, if the CLDT Box is connected and the green LED on the Box is lighting in continuous mode, a simple click on the 'Connect Button' re-initiates a new connection – process. The connection will be done automatically and the software indicates the new virtual COM port.

If the connection is not established, the correct virtual COM port must be chosen manually in the Windows Device Manager :

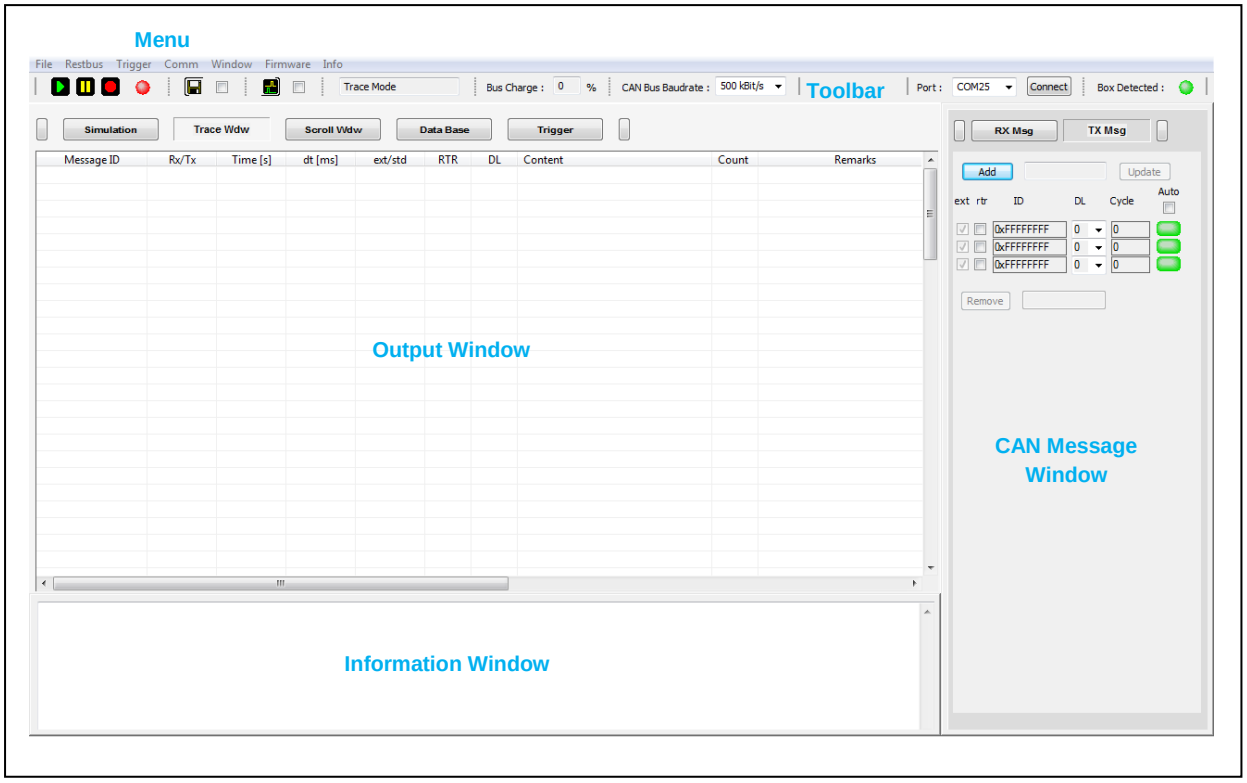


- ➔ disconnect the CLDT Box from the PC
- ➔ reconnect the CLDT Box to the PC
- ➔ wait until the green LED on the CLDT Box is lighting in continuous mode
- ➔ open Windows Device Manager and verify the dedicated COM port
- ➔ choose the right COM port in the Com-Port combo box and click the 'Connect Button' again
- ➔ the CLDT Box should be detected by the software now

If the Box is connected to the PC and no green LED appears on the Box, the drivers for the CLDT Box are not correctly installed. In this case please refer to the chapters *'VCP Driver Installation'* and *'Quick Check'*.

8.3 Main Window Overview

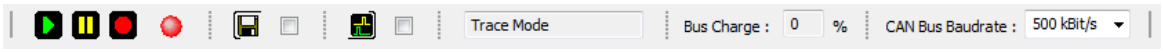
After starting the PC software the Applications Main Window will appear.



The Main Window is divided into five functional blocks. The detailed description of the functional blocks is given in the following chapters :

- Main Menu
- Toolbar
- Output Window
- Information Window
- CAN Message Window

8.4 Tool Bar Description



Start Trace or Start Simulation Button
Pause Button
Stop Trace or Stop Simulation Button
Trace Indicator : green = Simulation running
yellow = Pause
red = Simulation stopped

Save Trace Button

If the Save Trace Check Box is activated a file to save the CAN Trace data must be chosen at the beginning of a CAN Trace or Simulation.

Download Trigger Button and
Trigger Functionality Activation Check Box

Download Trigger Conditions to the CLDT Box with the Download Trigger Button.
If the Check Box is activated, the next CAN Trace or CAN Simulation is starting in Trigger Mode. Therefore the 4 HW Outputs of the CLDT Box are configured in Trigger Mode.

- 'Current Project' Text Field : Read only

  Indicates the currently loaded project

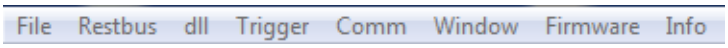
- Bus Charge Text Field : Read only

  If a Simulation or CAN Trace is running the current busload is monitored

- CAN Bus Baudrate List Field

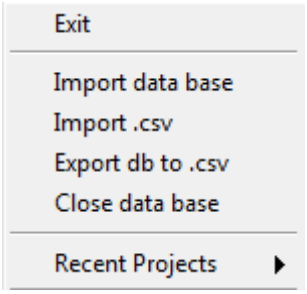
  Baudrate of the current project. Possible Baudrates : 1 MBaud / N
  N = 1, 2, 3 …

8.5 Menu Description



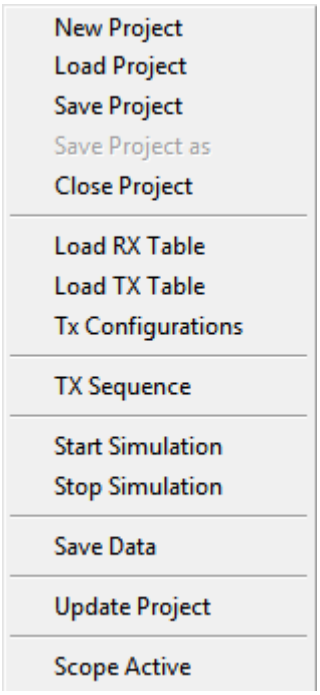
The Menu Items are described in detail in the following chapters :

8.5.1 File Menu



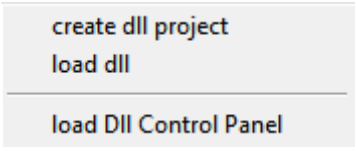
Menu Item	Description
Exit	Close Program
Import data base	Import dbc data base for RX Trace in Data Base Window
Import .csv	not implemented yet
Export db to .csv	not implemented yet
Close data base	Close current dbc data base
Recent Projects	Shortcut to the last five used projects

8.5.2 Restbus Menu



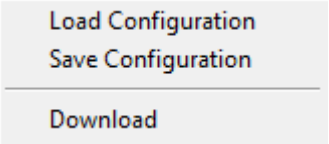
Menu Item	Description
New Project	opens a dialog to choose a new project
Load Project	opens a dialog to choose an existing project
Save Project	saves the current project configuration
Save Project as	not implemented yet
Close Project	closes the current project
Load RX Table	loads a new or an existing RX Simulation Window
Load TX Table	loads a new or an existing TX Simulation Window
TX Configurations	loads the TX Configurations Dialog
TX Sequence	loads a pre-defined TX Sequence
Start Simulation	same as Toolbar Button 'Start Simulation'
Stop Simulation	same as Toolbar Button 'Stop Simulation'
Save Data	not implemented yet
Update Project	saves all modifications to the current project
Scope Active	activates & shows the Scope Window

8.5.3 dll Menu



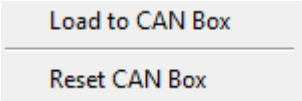
Menu Item	Description
create dll project	creates a dll project from the currently loaded project
load dll	loads a dll for RX/TX Automation Mode
load Dll Control Panel	adds a dll control panel to the current project

8.5.4 Trigger Menu



Menu Item	Description
Load Configuration	Load an Existing Trigger Configuration for the 8 Triggers
Save Configuration	Save the current Trigger Configuration
Download	Download the current Trigger Configuration to CLDT Box

8.5.5 Comm Menu

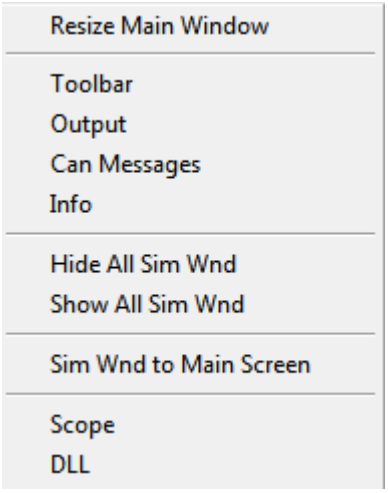


Menu Item	Description
Load to CAN Box	not implemented yet
Reset CAN Box	Resets the CLDT Box SW

8.5.6 Firmware Menu

Menu Item	Description
Download	opens the dialog to upgrade the CLDT Box Software Version

8.5.7 Window Menu



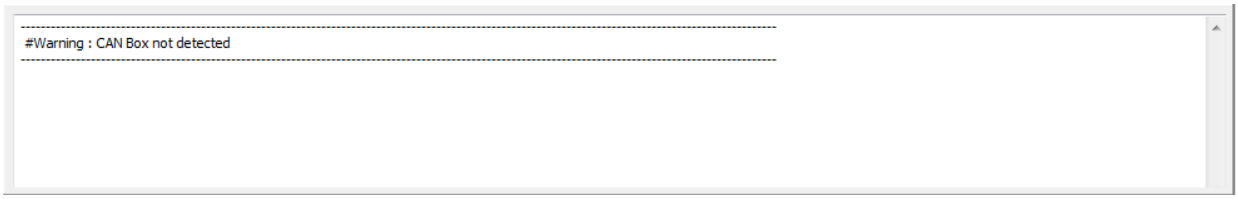
Menu Item	Description
Resize Main Window	Resets the Main Window Conmtent to show all Windows in the basic configuration
Toolbar	Hides / Shows the Toolbar
Output	Hides / Shows the Output Window
Can Messages	Hides / Shows the CAN Message Window
Info	Hides / Shows the Information Window
Hide All Sim Wnd	Hides all opened RX & TX Simulation Windows
Show All Sim Wnd	Shows all opened RX & TX Simulation Windows
Sim Wnd to Main Screen	Cascades all opened RX & TX Simulation Window on the main screen if more than one screen is used
Scope	Shows the Scope Window if scope function is active
DLL	Shows the DLL Window if dll function is active

8.5.8 Info Menu

Menu Item	Description
Version	opens a dialog which shows informations about PC and CLDT Box SW & HW Versions

8.6 Information Window

After starting the PC software the Information Window is situated at the bottom of the Application Window.



In the Information Window useful advices or details are indicated, especially if some kind of (logical) problems during the applications program flow appear.

Tip : To enlarge the screen size for other windows (e.g. Output Window) the Information Window can be hidden by the Menu Item
→ Window → Info

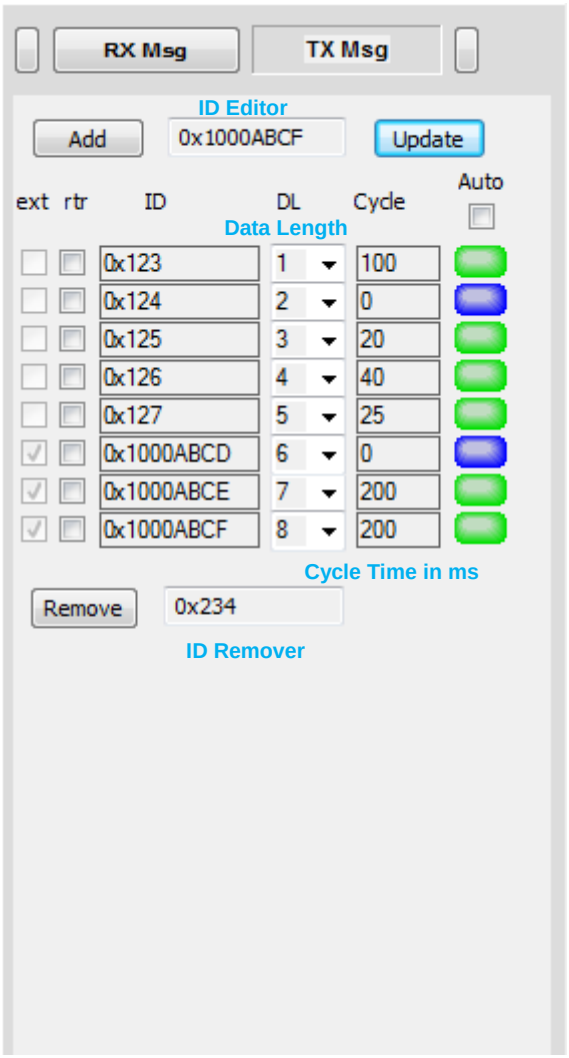
8.7 Can RX / TX Message Window

After starting the PC software the CAN Message Window is situated at the right side of the Application Window.

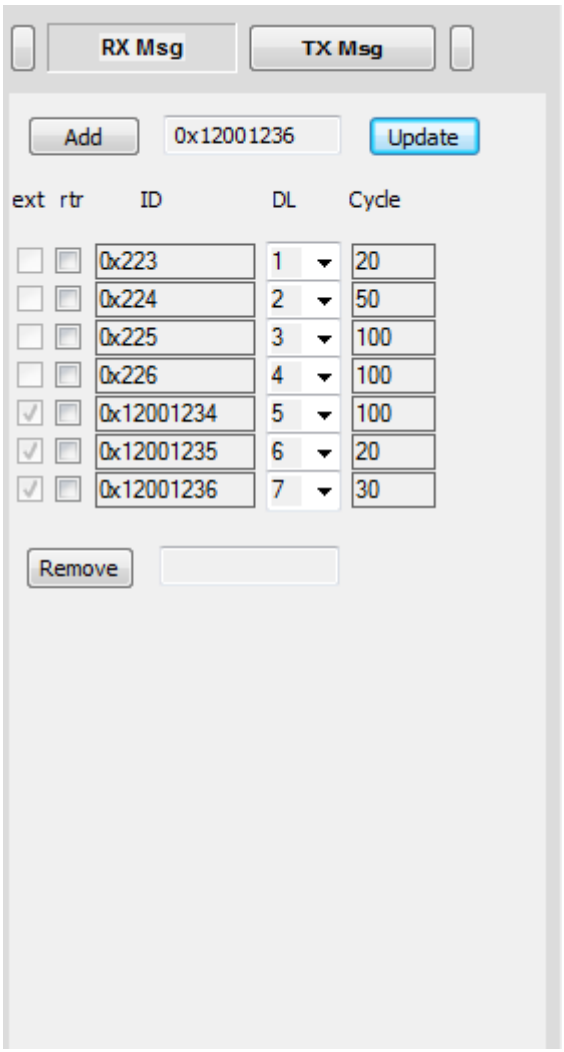
In the CAN RX & TX Message Windows the CAN IDs for the CAN Restbus Simulation are defined.

Up to 25 RX and 25 TX CAN IDs can be defined and be monitored in the Restbus Simulation Mode.

CAN TX Message Window



CAN RX Message Window



Explanation of the **'Auto'-Check Box** : see chapter *'Example : CAN Restbus Simulation'*

Procedure to define CAN RX & TX messages for Restbus Simulation

- Open a new or existing project (→Restbus – Menu)

- Choose the correct Baudrate for the project (→ Toolbar)

- edit CAN Messages in hex-format in the ID Editor with preceding 0x…

  - standard CAN messages in 5 character format (e.g. 0x012)
  - extended CAN Messages in 10 character format (e.g. 0x001234AB)

- remove CAN messages in hex-format in the ID Remover with preceding 0x…

- define Data Length & Cycle Time for each CAN Message

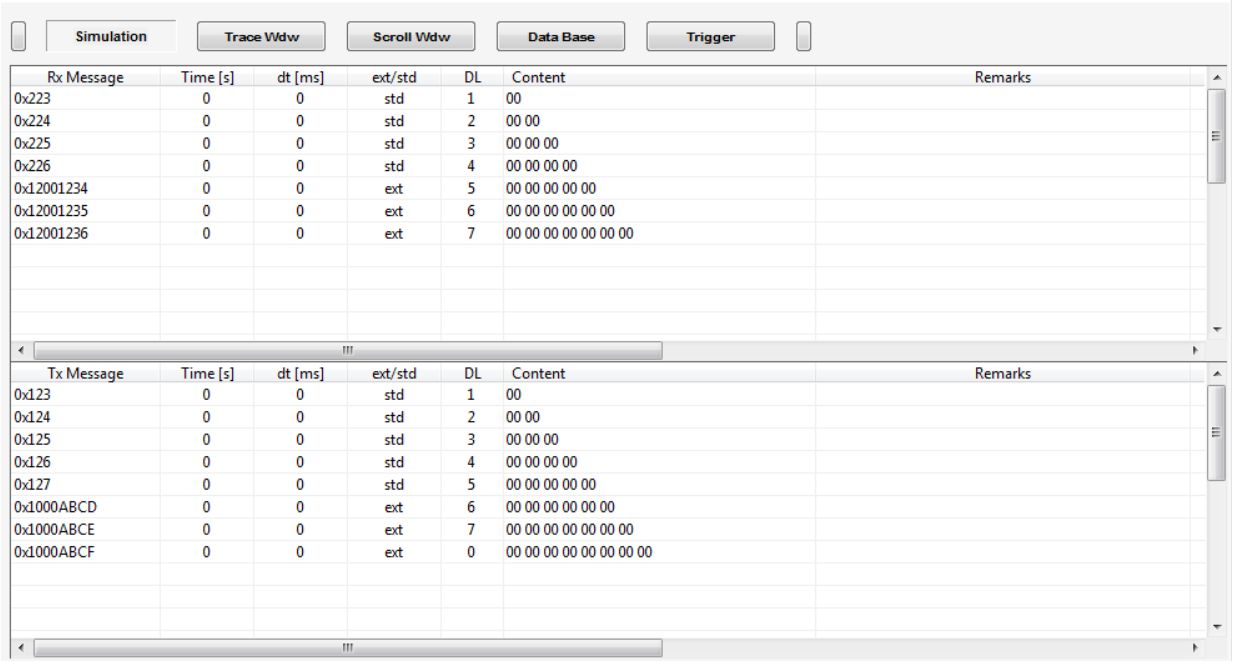
  - Cycle Time is mandatory for CAN TX Message
  - Cycle Time is optional for CAN RX Messages

Cycle Time	Description
0	manual triggered CAN Message
> 0	cyclic CAN Message in [ms]
Data Length	Description
0 … 8	possible Data Length : 0 byte … 8 byte

- finally click the ‘Update Button’ to save all changes to the project files

- Cyclic TX messages appear with green TX Buttons, triggered TX messages appear with blue TX Buttons

- the project is updated and the defined CAN RX & TX Messages appear now, as shown below, in the Output – Simulation Window



Rx Message	Time [s]	dt [ms]	ext/std	DL	Content	Remarks
0x223	0	0	std	1	00	
0x224	0	0	std	2	00 00	
0x225	0	0	std	3	00 00 00	
0x226	0	0	std	4	00 00 00 00	
0x12001234	0	0	ext	5	00 00 00 00 00	
0x12001235	0	0	ext	6	00 00 00 00 00 00	
0x12001236	0	0	ext	7	00 00 00 00 00 00 00	

Tx Message	Time [s]	dt [ms]	ext/std	DL	Content	Remarks
0x123	0	0	std	1	00	
0x124	0	0	std	2	00 00	
0x125	0	0	std	3	00 00 00	
0x126	0	0	std	4	00 00 00 00	
0x127	0	0	std	5	00 00 00 00 00	
0x1000ABCD	0	0	ext	6	00 00 00 00 00 00	
0x1000ABCE	0	0	ext	7	00 00 00 00 00 00 00	
0x1000ABCF	0	0	ext	0	00 00 00 00 00 00 00 00	

Output Simulation Window

> ***Tip :*** Each time the CAN Message IDs or the Baudrate Setting for a project are modified, the **project must be updated** via the ‘Update Button’ before starting a new restbus simulation

8.8 Output Window

The Output Window offers two main functional groups for user interaction :

- Display CAN Messages in different formats
- Define Trigger Conditions for the use in Trigger Mode

A) CAN Message Display

If a CAN Trace or CAN Simulation is running, the CAN Bus Data is shown in the Output Window. The CAN Data may be displayed in 4 different ways :

1) **Simulation Window**

   Only the CAN Messages that are defined in the CAN RX & CAN TX Message Windows are shown (in real time)

2) **Trace Window**

   All CAN Messages detected on the Bus are shown (in real time)

3) **Scroll Window**

   All CAN Bus Messages detected on the Bus are shown in a real-time scrolling mode.

4) **Data Base Window**

   If a CAN Data Base is loaded, the CAN Messages that are defined in the dbc-file are shown in a Message – Signal – Tree Structure (in real time)

B) Define Trigger Conditions

Trigger Window

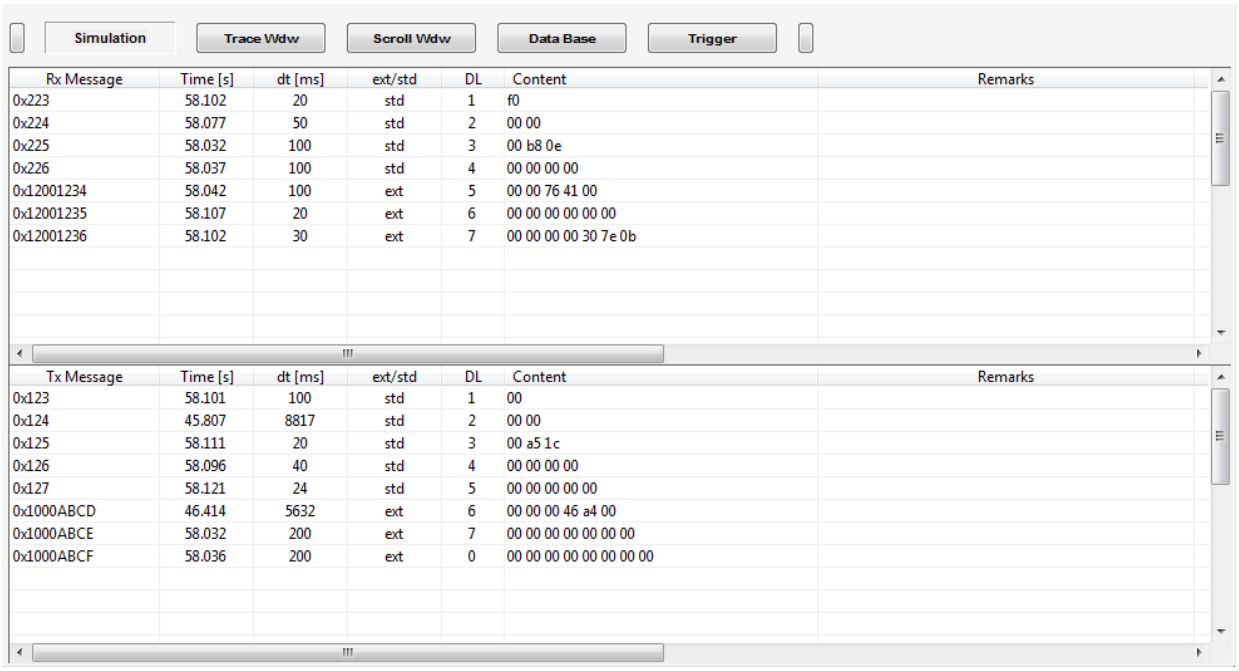
The user has the possibility to define up to 8 different trigger conditions that can be used to trigger the 4 HW Outputs when certain CAN messages or CAN message contents appear on the bus

- 4 x HW Trigger Output on CAN Messages or CAN Message Contents
- 4 x Virtual Triggers to create up to 8 different trigger conditions
- combine trigger conditions with logical combinations (AND, OR …)
- typical Trigger Delay < 100 µs

8.8.1 Simulation Window

If a Restbus Simulation is ongoing the CAN Messages that are defined in the CAN Message RX & TX Windows are displayed in real time in the Simulation Window with the following message content :

- CAN Message ID
- absolute time in [s] of last message appearance on the bus
- relative time in [ms] between the last two message appearances on the bus
- information if message is in standard or extended format
- Message Data Length
- Message content in hex-format
- remarks edit field for additional user information



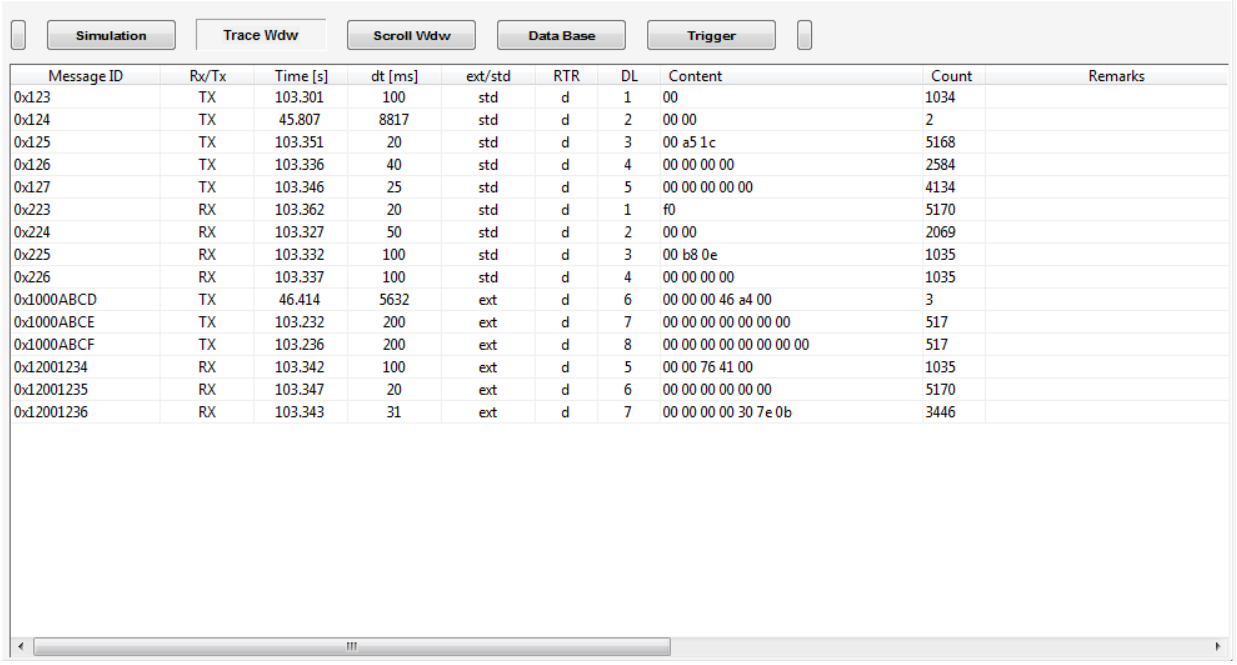
Rx Message	Time [s]	dt [ms]	ext/std	DL	Content	Remarks
0x223	58.102	20	std	1	f0	
0x224	58.077	50	std	2	00 00	
0x225	58.032	100	std	3	00 b8 0e	
0x226	58.037	100	std	4	00 00 00 00	
0x12001234	58.042	100	ext	5	00 00 76 41 00	
0x12001235	58.107	20	ext	6	00 00 00 00 00 00	
0x12001236	58.102	30	ext	7	00 00 00 00 30 7e 0b	

Tx Message	Time [s]	dt [ms]	ext/std	DL	Content	Remarks
0x123	58.101	100	std	1	00	
0x124	45.807	8817	std	2	00 00	
0x125	58.111	20	std	3	00 a5 1c	
0x126	58.096	40	std	4	00 00 00 00	
0x127	58.121	24	std	5	00 00 00 00 00	
0x1000ABCD	46.414	5632	ext	6	00 00 00 46 a4 00	
0x1000ABCE	58.032	200	ext	7	00 00 00 00 00 00 00	
0x1000ABCF	58.036	200	ext	0	00 00 00 00 00 00 00 00	

8.8.2 Trace Window

If a Restbus Simulation or a CAN Trace is ongoing all CAN Messages received or transmitted by the CLDT Box are displayed in real time in the Trace Window with the following message content :

- CAN Message ID
- Information if message is received or transmitted by the CLDT Box
- absolute time in [s] of last message appearance on the bus
- relative time in [ms] between the last two message appearances on the bus
- information if message is in standard or extended format
- information if message is a RTR (Remote Transmission Request) or Data
- Message Data Length
- Message content in hex-format
- Number of total messages received with this CAN ID
- remarks edit field for additional user information

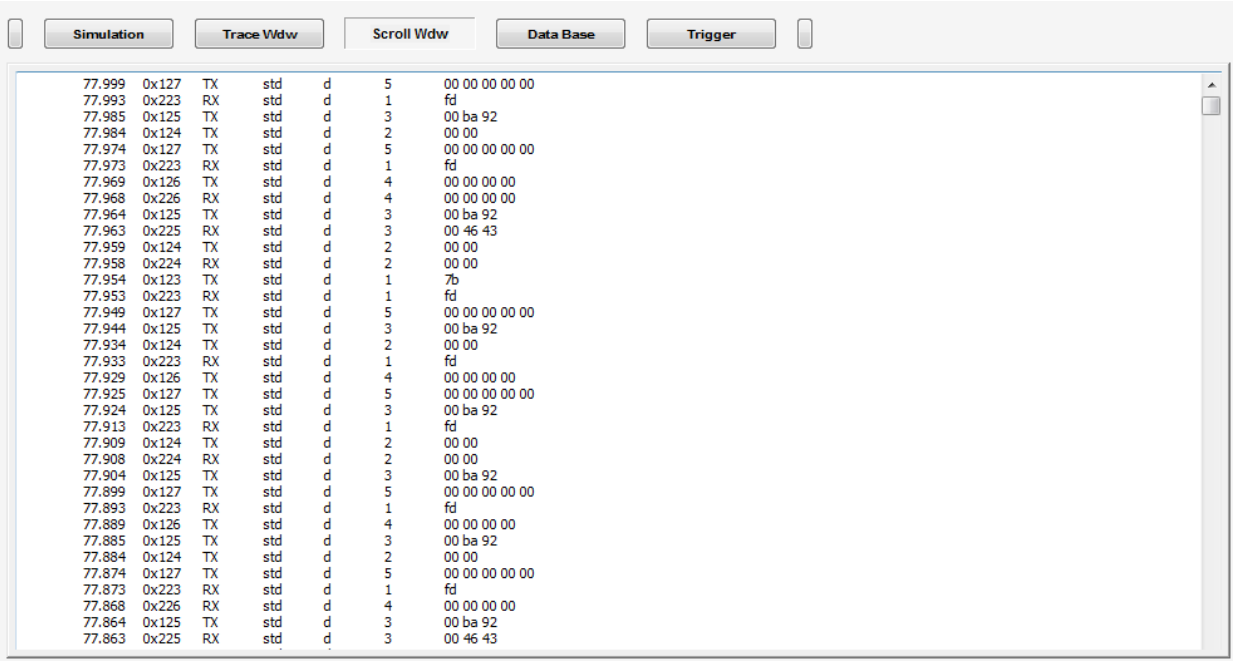


Message ID	Rx/Tx	Time [s]	dt [ms]	ext/std	RTR	DL	Content	Count	Remarks
0x123	TX	103.301	100	std	d	1	00	1034	
0x124	TX	45.807	8817	std	d	2	00 00	2	
0x125	TX	103.351	20	std	d	3	00 a5 1c	5168	
0x126	TX	103.336	40	std	d	4	00 00 00 00	2584	
0x127	TX	103.346	25	std	d	5	00 00 00 00 00	4134	
0x223	RX	103.362	20	std	d	1	f0	5170	
0x224	RX	103.327	50	std	d	2	00 00	2069	
0x225	RX	103.332	100	std	d	3	00 b8 0e	1035	
0x226	RX	103.337	100	std	d	4	00 00 00 00	1035	
0x1000ABCD	TX	46.414	5632	ext	d	6	00 00 00 46 a4 00	3	
0x1000ABCE	TX	103.232	200	ext	d	7	00 00 00 00 00 00 00	517	
0x1000ABCF	TX	103.236	200	ext	d	8	00 00 00 00 00 00 00 00	517	
0x12001234	RX	103.342	100	ext	d	5	00 00 76 41 00	1035	
0x12001235	RX	103.347	20	ext	d	6	00 00 00 00 00 00	5170	
0x12001236	RX	103.343	31	ext	d	7	00 00 00 00 30 7e 0b	3446	

8.8.3 Scroll Window

If a Restbus Simulation or a CAN Trace is ongoing all CAN Messages received or transmitted by the CLDT Box are displayed in real time in the Scroll Window with the following content :

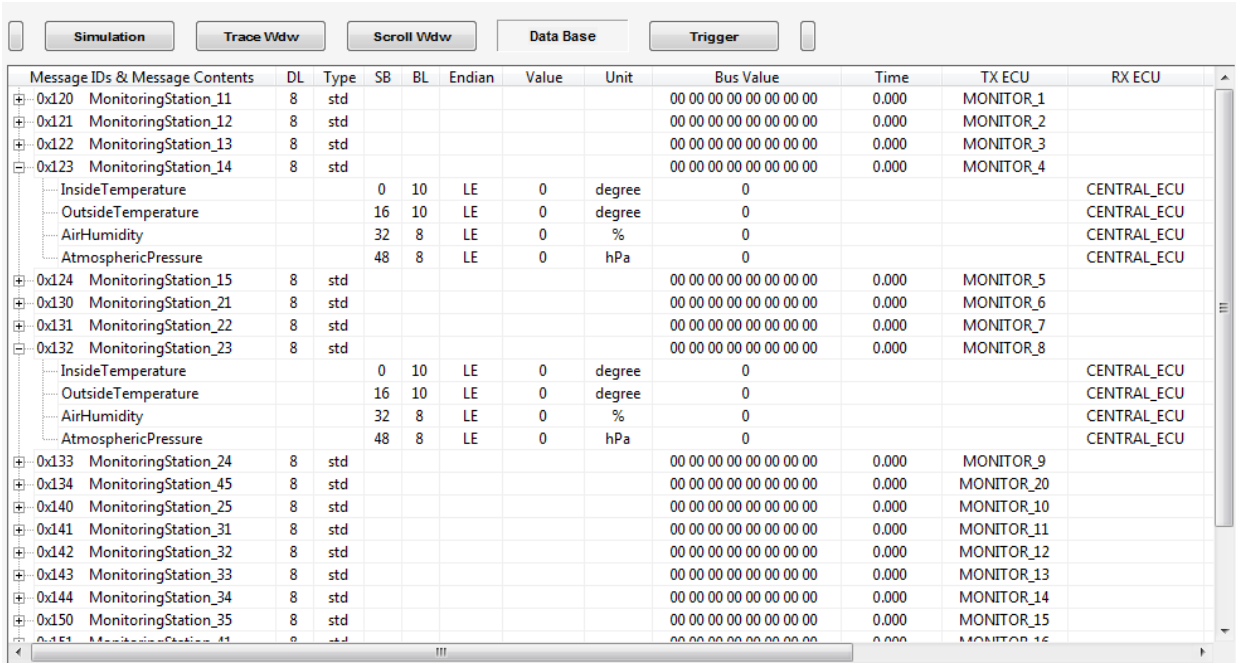
- absolute time in [s] of last message appearance on the bus
- CAN Message ID
- Information if message is received or transmitted by the CLDT Box
- information if message is in standard or extended format
- information if message is a RTR (Remote Transmission Request) or Data
- Message Data Length
- Message content in hex-format



8.8.4 Data Base Window

If a CAN data base (.dbc or .dbcj-file) has been loaded and a CAN Trace is ongoing, all received CAN Messages are displayed in real time in the Data Base Window in a Message-Signal-Tree-Structure with the following contents :

- CAN Message ID & Message or Signal Name
- Message Data Length
- information if message is in standard or extended format
- Signal Startbit information absolute position (0 … 63 max.)
- Signal Bitlength information
- Signal format information : Little Endian or Big Endian
- Calculated signal value
- Signal unit
- Message content in hex-format
- absolute time in [s] of last message appearance on the bus
- information about the Transmitter ECU
- information about the Reveiver ECU



Message IDs & Message Contents	DL	Type	SB	BL	Endian	Value	Unit	Bus Value	Time	TX ECU	RX ECU
0x120 MonitoringStation_11	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_1	
0x121 MonitoringStation_12	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_2	
0x122 MonitoringStation_13	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_3	
0x123 MonitoringStation_14	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_4	
InsideTemperature			0	10	LE	0	degree	0			CENTRAL_ECU
OutsideTemperature			16	10	LE	0	degree	0			CENTRAL_ECU
AirHumidity			32	8	LE	0	%	0			CENTRAL_ECU
AtmosphericPressure			48	8	LE	0	hPa	0			CENTRAL_ECU
0x124 MonitoringStation_15	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_5	
0x130 MonitoringStation_21	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_6	
0x131 MonitoringStation_22	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_7	
0x132 MonitoringStation_23	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_8	
InsideTemperature			0	10	LE	0	degree	0			CENTRAL_ECU
OutsideTemperature			16	10	LE	0	degree	0			CENTRAL_ECU
AirHumidity			32	8	LE	0	%	0			CENTRAL_ECU
AtmosphericPressure			48	8	LE	0	hPa	0			CENTRAL_ECU
0x133 MonitoringStation_24	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_9	
0x134 MonitoringStation_45	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_20	
0x140 MonitoringStation_25	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_10	
0x141 MonitoringStation_31	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_11	
0x142 MonitoringStation_32	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_12	
0x143 MonitoringStation_33	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_13	
0x144 MonitoringStation_34	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_14	
0x150 MonitoringStation_35	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_15	

8.8.4.1 dbcj - Data Base Editor – (not implemented yet)

A Data Base Editor is planned to be integrated in a future SW release to offer the possibility to create user specific data tree structures.

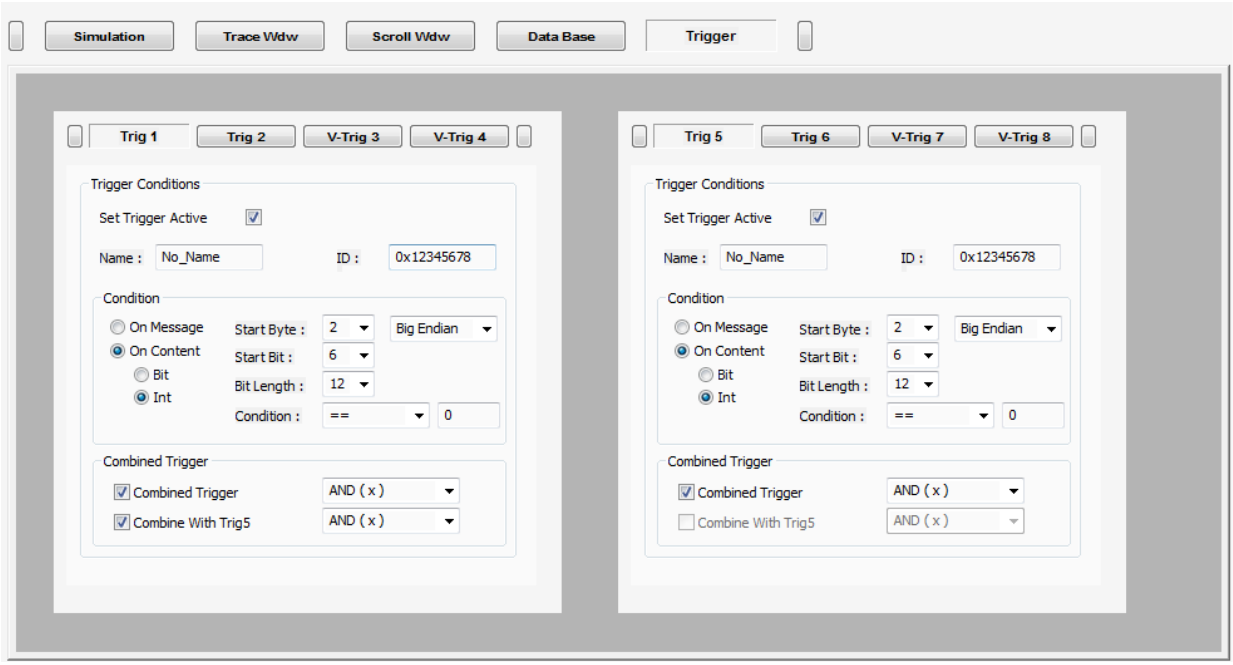
8.8.5 Trigger Window

The CLDT Box provides 4 digital HW Outputs that can either be used

➔ as Digital Signal Outputs which can be synchronized with a Restbus Simulation TX Message or manually set / reset by the user

or

➔ as Digital Trigger Signal Outputs triggering when predefined CAN Messages or predefined CAN Message Contents appear on the Bus



Trigger Window with 4 x HW Trigger & 4 x Virtual Trigger

8.8.5.1 Define Trigger Events

In the Trigger Window up to 8 different trigger conditions causing trigger pulses or trigger signals on the 4 HW Trigger Outputs can be defined.

The trigger conditions can be combined among themselves by respecting certain rules in order to offer a wide range of message and message content combinations for triggering.

The following table is showing the trigger conditions with their dedicated HW Outputs :

SW Trigger Condition	triggered HW Output
Trig1	HW Output 1
Trig2	HW Output 2
Trig5	HW Output 3
Trig6	HW Output 4
V-Trig3	Virtual Trigger : only useful for combined triggering
V-Trig4	Virtual Trigger : only useful for combined triggering
V-Trig7	Virtual Trigger : only useful for combined triggering
V-Trig8	Virtual Trigger : only useful for combined triggering

Trigger Conditions for Simple Trigger Events

Condition 1	Condition 2	Nb of bits	Condition 3	Value	Trigger Output
On Message					positive pulse
On Content	Bit	1	== 1		true : High false : Low
			== 0		true : High false : Low
			On Change		positive pulse
	Int	2 … 16	==	16 bit value	true : High false : Low
			<	16 bit value	true : High false : Low
			>	16 bit value	true : High false : Low
			On Change		positive pulse

Message Info	Description
Set Trigger Active	the trigger is activated only if this check box is set
Name	user definable trigger name
ID	CAN ID for the message which shall be triggered
Start Byte	Startbyte if trigger on message content
Start Bit	Startbit if trigger on message content
Bit Length	Bitlength if trigger on message content
Endian	Little Endian or Big Endian Format

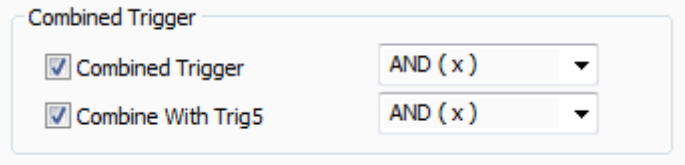
Explanation of ‘Startbyte’, ‘Startbit’, ‘Bitlength’ and ‘Endian’ :

Chapter *Little Endian / Big Endian Support*

Trigger Usage Example : Chapter *Example : Trigger Output Usage*

Trigger Conditions for Combined Trigger Events : Use of Virtual Triggers

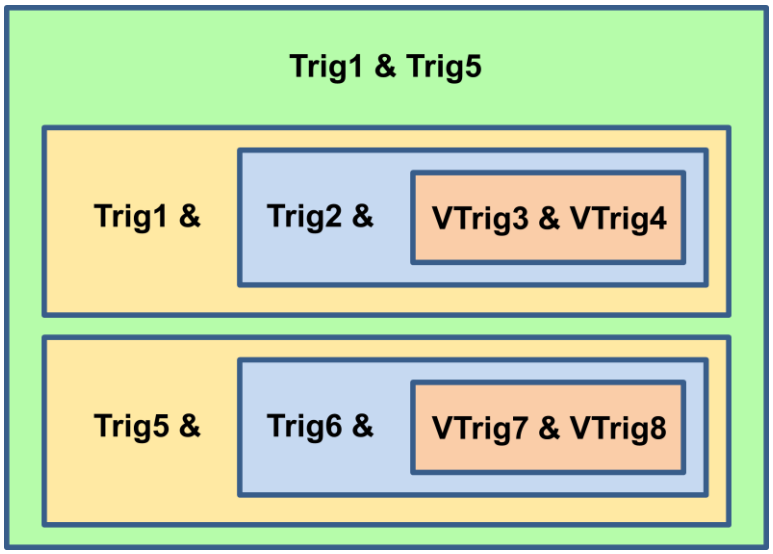
Each trigger can be combined with its trigger superior in hierarchy by setting the corresponding check box in the Trigger Window. Additionally the trigger 1 can be combined with trigger 5.



The following logical concatenations for combined triggers can be used :

➔	trigger(x)	AND	trigger(x+1)
➔	trigger(x)	OR	trigger(x+1)
➔	trigger(x)	AND NOT	trigger(x+1)
➔	trigger(x)	OR NOT	trigger(x+1)

The following trigger combinations are possible :

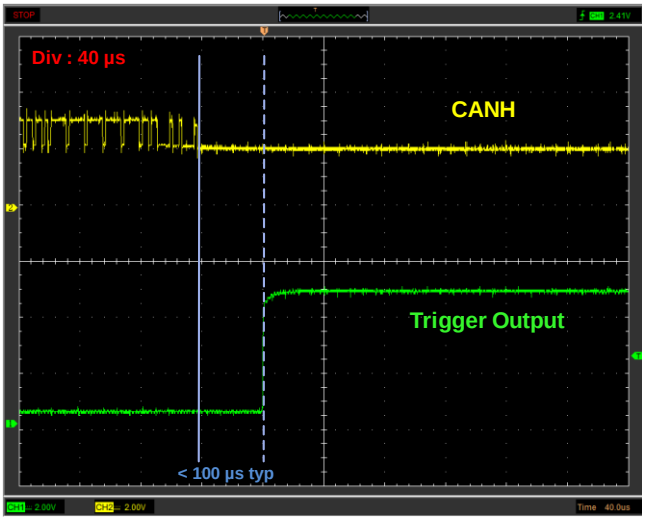


Trigger Usage Example : Chapter *Example : Trigger Output Usage* .

8.8.5.2 *Typical Trigger Response Time*

The trigger response time depends slightly on the number of trigger combinations which are defined for a chosen trigger. The greater the number of trigger combinations, the greater the trigger delay.

The following figure shows the typical delay and typical rising time for the HW trigger outputs :



typical trigger response time :	**< 100 µs**
typical rising time :	**< 50 µs**

8.9 Restbus Simulation RX & TX Windows

For the restbus simulation the user interaction is realized via the RX & TX Windows.

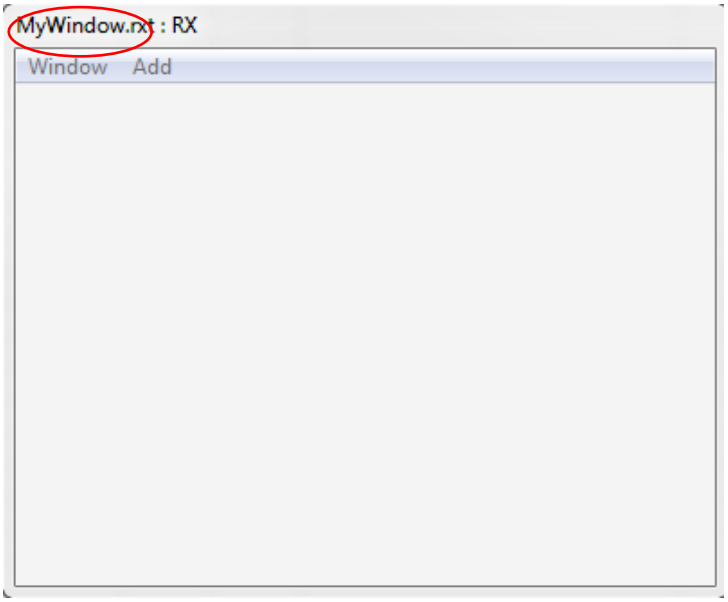
8.9.1 RX Windows

RX Windows allow the user to choose variables within the pre-defined CAN RX messages from the *CAN RX Message Window* and display their values in the RX Windows in real time.

8.9.1.1 Add RX Windows to an existing project

Main Menu	Item	Description
Restbus	Load Rx Table	Load a new or an existing RX Simulation Window

Choose the RX Windows Name and click the ‘Open-Button’. An empty RX-Window will be opened. The Window Name appears at the top of the window



8.9.1.2 Remove RX Windows from an existing project

Sim Menu	Item	Description
Window	Close	Close RX Windows or remove RX Windows from the project

→ Choose if the window just shall be closed (window will be re-opened on next project load)

or

if the window shall be completely removed from the project.

Remark : The window will never be deleted. If a window shall be deleted definitively this must be done by the Windows Explorer.

8.9.1.3 Open existing RX Windows in a project

Main Menu	Item	Description
Restbus	Load Rx Table	Load a new or an existing RX Simulation Window

Choose the existing RX-Table and click on the ‘Open-Button’.

8.9.1.4 RX / TX Window Menu Items

Menu	Item	Description
Window	Hide	window is hidden (not closed !)
	Redraw	window is just redrawn in case of displaying problems
	Close	Close window or remove window from project
Add	Flag	a new boolean control is added
	Value	a new integer value control is added
	Ext Value	a new extended integer value control is added
	Radio Control	a new radio control is added
	Slide Control	a new slide control is added

Remark : Extended Values, radio controls & slide controls are only available from PC SW Version VA003 on.

8.9.2 TX Windows

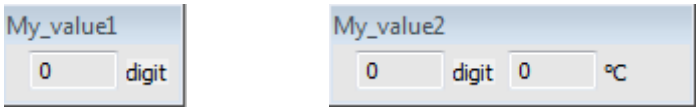
TX Windows allow the user to set variables within the pre-defined CAN TX messages from the *CAN TX Message Window* and put their values on the bus in real time.

TX Window handling (add, remove etc.) is the same as *RX Window handling* .

8.9.3 User Controls for RX & TX Windows

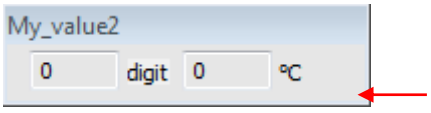
8.9.3.1 Value Control

- Up to 20 integer variable controls (2 … 32 bit) can be added to each RX or TX Window
- For each of the values the Value Properties must be set to define the allocation between the bus value and the RX / TX Window Values
- Depending on the properties 'Show digit only' (see chapter Value Properties), the value is shown in small or enlarged format



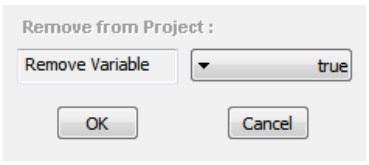
Remove Value Controls from RX or TX Windows

- A double mouse click on the right bottom corner of the control opens its properties dialog

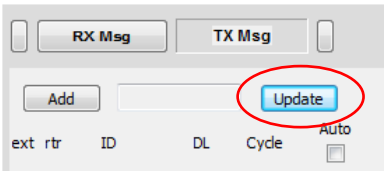


double click

- Set the 'Remove' properties at the bottom of the dialog to 'true', press 'Enter' or type 'OK'



- update the project with the 'update command button' in the CAN message window



8.9.3.2 Value Control Properties

The value control properties define the allocation between the bus value and the simulation values that are displayed in the RX / TX Windows.

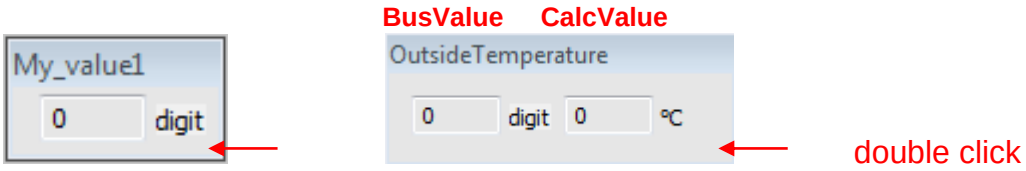
Define Value Control Properties

In the Value Control Window the integer bus value is displayed on the left side. On the right side a float value which is calculated with the linear formula

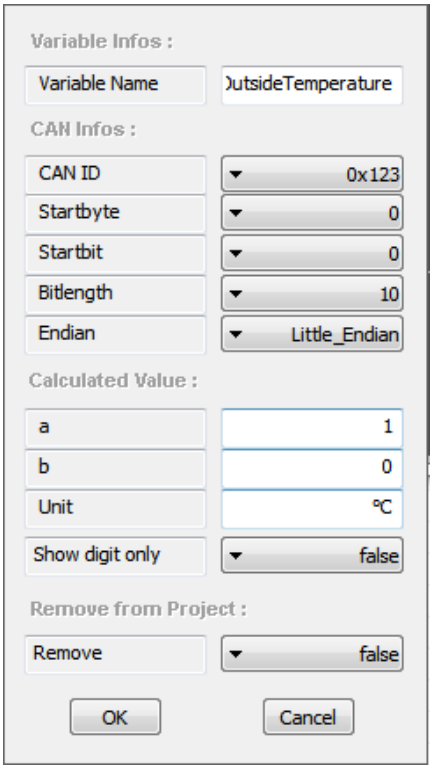
*CalcValue = a * BusValue + b*

is displayed. a & b are defined in the control properties.

To display the control properties window double click the right bottom corner of the control



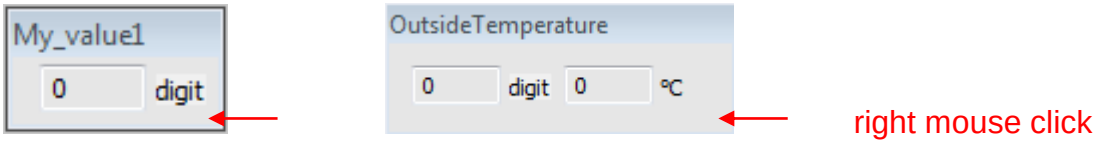
Define Properties



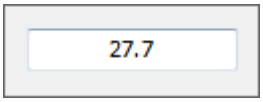
Properties	Description
Variable Name	name that appears on top of control
CAN ID	CAN ID that contains the variable
Startbyte	see ‘CAN Message Format’
Startbit	see ‘CAN Message Format’
Bitlength	see ‘CAN Message Format’
Endian	see ‘CAN Message Format’
a	coefficient for value calculation
b	offset for value calculation
Unit	unit for the calculated value
Show digit only	display digit value only
Remove	remove variable from project

8.9.3.3 Change the controls TX Value during Restbus Simulation

To display the controls value window right mouse click the right bottom corner of the control



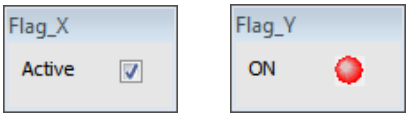
Insert the new value into the value dialog



Confirm with ‘ENTER’ or exit with ‘ESC’

8.9.3.4 Flag Control - Boolean Value

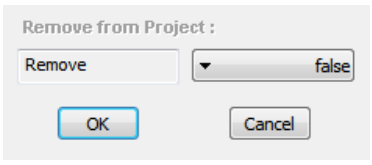
- Up to 20 flag controls (1 bit values) can be added to each RX or TX Window
- For each of the flags the Flag Properties must be set to define the allocation between the bus value and the RX / TX Window Values
- Depending on the properties 'LED symbol' (see chapter Flag Properties), Boolean values can be displayed in 'Check Box' or 'LED' format



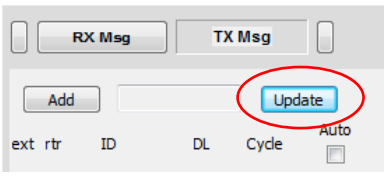
Remove Flag Controls from RX or TX Windows

- A double mouse click on the right bottom corner of the control opens its properties dialog

- Set the 'Remove' properties at the bottom of the dialog to 'true', press 'Enter' or type 'OK'



- update the project with the 'update command button' in the CAN message window



8.9.3.5 Flag Control Properties

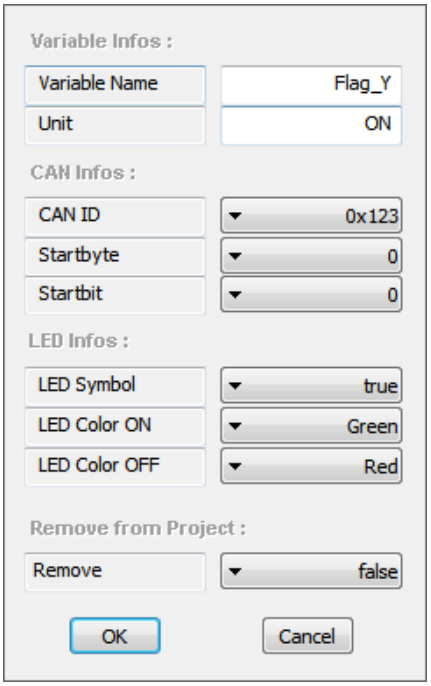
The flag control properties define the allocation between the bus value and the simulation flags that are displayed in the RX / TX Windows.

Define Flag Control Properties

To display the control properties window double click the right bottom corner of the control

double click

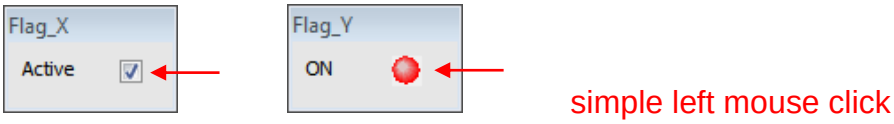
Define Properties



Properties	Description
Variable Name	name that appears on top of control
Unit	unit for the calculated value
CAN ID	CAN ID that contains the variable
Startbyte	see ‘CAN Message Format’
Startbit	see ‘CAN Message Format’
LED Symbol	display LED symbol or check box
LED Color ON	‘true‘ color
LED Color OFF	‘false‘ color
Remove	remove flag from project

8.9.3.6 Change the controls TX Value during Restbus Simulation

To change the Boolean TX value of a flag control during restbus simulation just simple click the check box or the LED symbol.

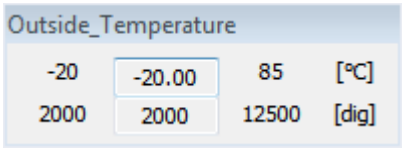


If a LED symbol is activated the color will switch between the two colors defined in the properties dialog.

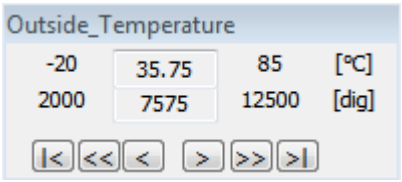
8.9.3.7 Extended Value Control - (from PC SW VA003 on)

- Up to 10 extended integer variable controls (2 … 32 bit) can be added to each RX or TX Window
- For each of the values the Value Properties must be set to define the allocation between the bus value and the RX / TX Window Values
- Depending on RX or TX windows the extended value is shown with or without positioning buttons

RX Window

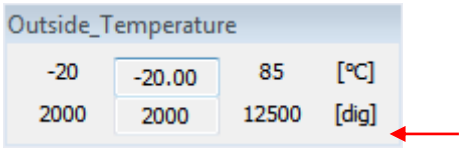


TX Window

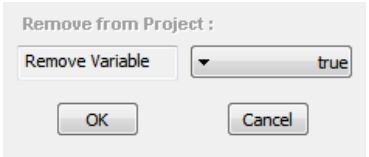


Remove Extended Value Controls from RX or TX Windows

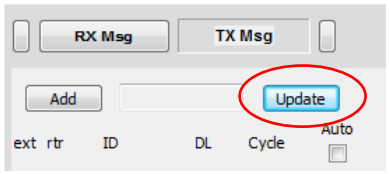
- A double mouse click on the right bottom corner of the control opens its properties dialog



- Set the ‘Remove’ properties at the bottom of the dialog to ‘true’, press ‘Enter’ or type ‘OK’



- update the project with the ‘update command button’ in the CAN message window

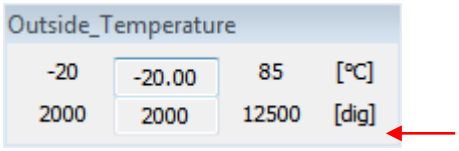


8.9.3.8 Extended Value Control Properties

The extended value control properties define the allocation between the bus value and the simulation flags that are displayed in the RX / TX Windows.

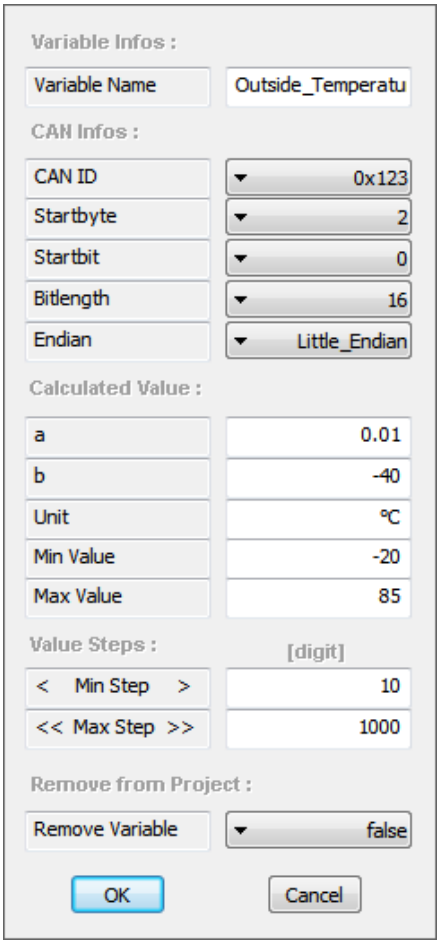
Define Extended Value Control Properties

To display the control properties window double click the right bottom corner of the control



double click

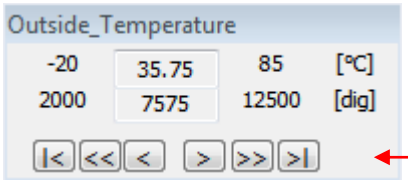
Define Properties



Properties	Description
Variable Name	name that appears on top of control
CAN ID	CAN ID that contains the variable
Startbyte	see ‘CAN Message Format’
Startbit	see ‘CAN Message Format’
Bitlength	see ‘CAN Message Format’
Endian	see ‘CAN Message Format’
a	coefficient for value calculation
b	offset for value calculation
Unit	unit for the calculated value
Min Value	min. limit value for TX values
Max Value	max. limit value for TX values
< Min Step >	TX : step size in digit : < or > button
<< Max Step >>	TX : step size in digit : << or >> button
Remove	remove variable from project

8.9.3.9 Change the controls TX Value during Restbus Simulation

1) Use controls positioning buttons

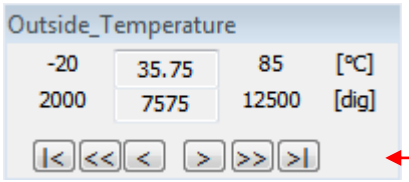


simple mouse click

With each simple mouse click the TX value will increase or decrease with the step size [digit] defined in the controls properties.

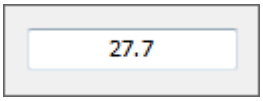
2) Use controls Value Window

To display the controls value window right mouse click the right bottom corner of the control



right mouse click

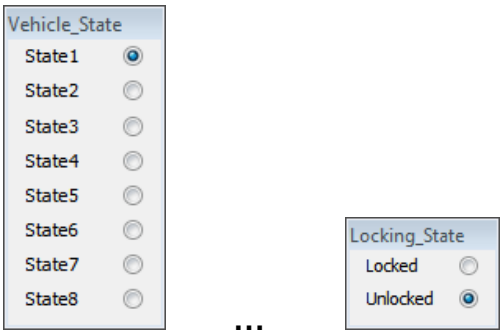
Insert the new value into the value dialog



Confirm with ‘ENTER’ or exit with ‘ESC’

8.9.3.10 Radio Control - (from PC SW VA003 on)

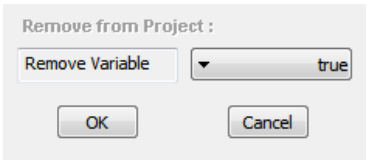
- Up to 10 radio controls can be added to each RX or TX Window
- For each of the values the Value Properties must be set to define the allocation between the bus value and the RX / TX Window Values
- Each radio control can display between 2 and 8 different states



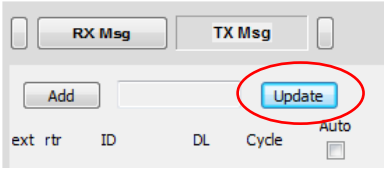
Remove Radio Controls from RX or TX Windows

- A double mouse click on the right bottom corner of the control opens its properties dialog

- Set the ‘Remove’ properties at the bottom of the dialog to ‘true’, press ‘Enter’ or type ‘OK’



- update the project with the ‘update command button’ in the CAN message window

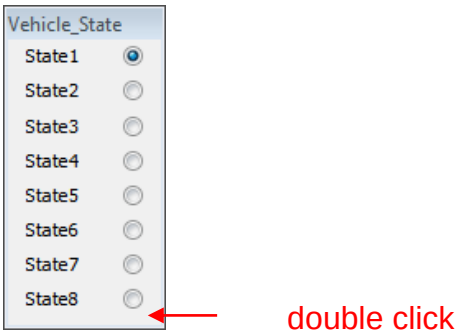


8.9.3.11 Radio Control Properties

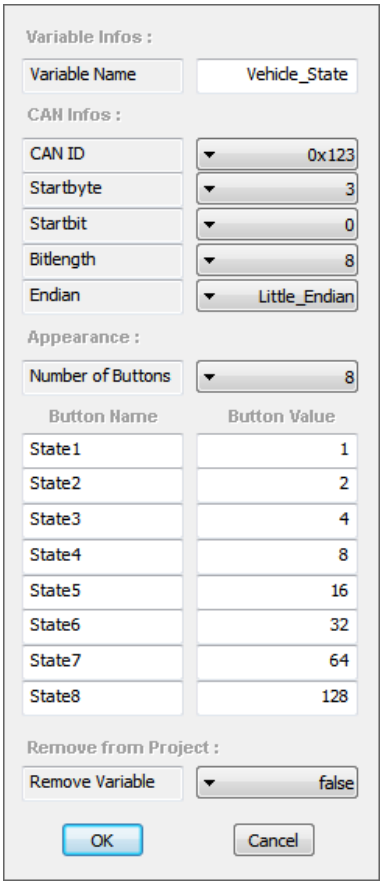
The radio control properties define the allocation between the bus value and the simulation flags that are displayed in the RX / TX Windows.

Define Radio Control Properties

To display the control properties window double click the right bottom corner of the control



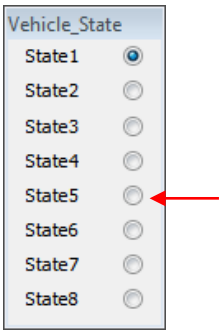
Define Properties



Properties	Description
Variable Name	name that appears on top of control
CAN ID	CAN ID that contains the variable
Startbyte	see ‘CAN Message Format’
Startbit	see ‘CAN Message Format’
Bitlength	see ‘CAN Message Format’
Endian	see ‘CAN Message Format’
Nb of Buttons	Number of displayed buttons
Button Name	name of the corresponding button
Button Value	digit value for 'true' condition
Remove	remove variable from project

8.9.3.12 Change the controls TX Value during Restbus Simulation

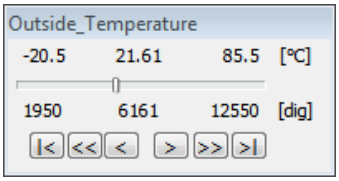
To change the TX value of a radio control during restbus simulation just simple click the new button within the control.



The value, concatenated to this button in the controls properties, will be put on the bus.

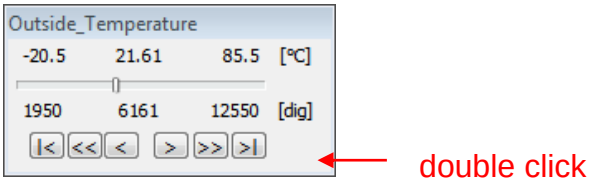
8.9.3.13 Slide Control - (from PC SW VA003 on)

- Up to 10 slide controls can be added to each RX or TX Window
- For each of the values the Value Properties must be set to define the allocation between the bus value and the RX / TX Window Values

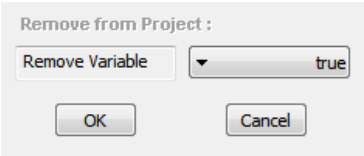


Remove Slide Controls from RX or TX Windows

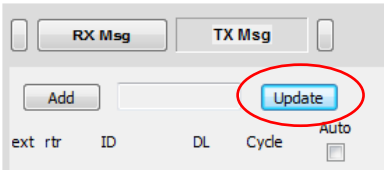
- A double mouse click on the right bottom corner of the control opens its properties dialog



- Set the ‘Remove’ properties at the bottom of the dialog to ‘true’, press ‘Enter’ or type ‘OK’



- update the project with the ‘update command button’ in the CAN message window

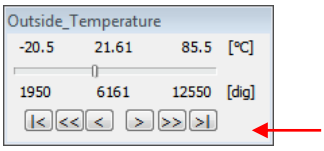


8.9.3.14 Slide Control Properties

The slide control properties define the allocation between the bus value and the simulation flags that are displayed in the RX / TX Windows.

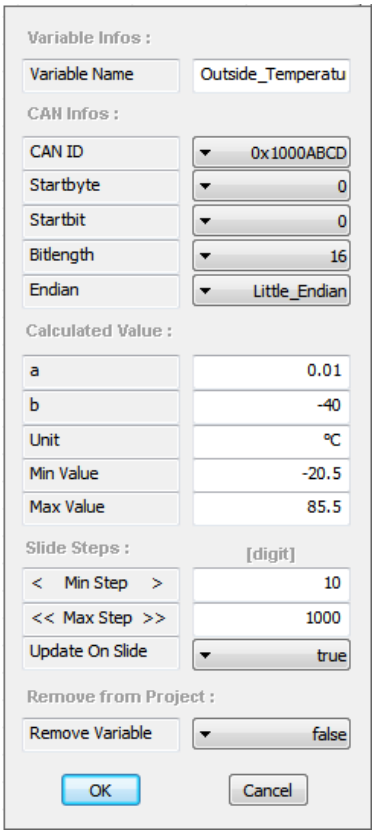
Define Slide Control Properties

To display the control properties window double click the right bottom corner of the control

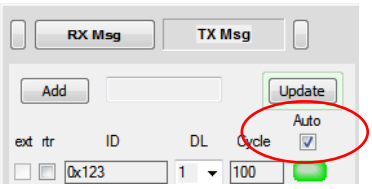


double click

Define Properties

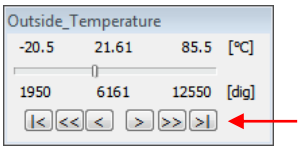


Properties	Description
Variable Name	name that appears on top of control
CAN ID	CAN ID that contains the variable
Startbyte	see ‘CAN Message Format’
Startbit	see ‘CAN Message Format’
Bitlength	see ‘CAN Message Format’
Endian	see ‘CAN Message Format’
a	coefficient for value calculation
b	offset for value calculation
Unit	unit for the calculated value
Min Value	min. limit value
Max Value	max. limit value
< Min Step >	step size in digit : < or > button
<< Max Step >>	step size in digit : << or >> button
Update on slide	auto update slide position during sliding (only if auto - update is enabled)
Remove	remove variable from project



8.9.3.15 Change the controls TX Value during Restbus Simulation

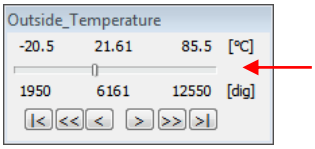
1) Use controls positioning buttons



simple mouse click

With each simple mouse click the TX value will increase or decrease with the step size [digit] defined in the controls properties.

2) Use Slider

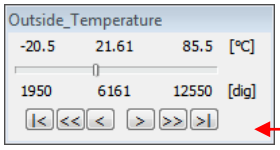


left mouse down & draw the slider

Remark : If the control property ‘Update On Slide’ is set to ‘true’ and the auto-update check box in the CAN TX Messages window is set, the TX value, put on the bus, is updated cyclically during sliding !

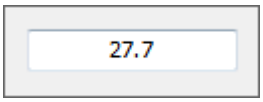
3) Use controls Value Window

To display the controls value window right mouse click the right bottom corner of the control



right mouse click

Insert the new value into the value dialog



Confirm with ‘ENTER’ or exit with ‘ESC’

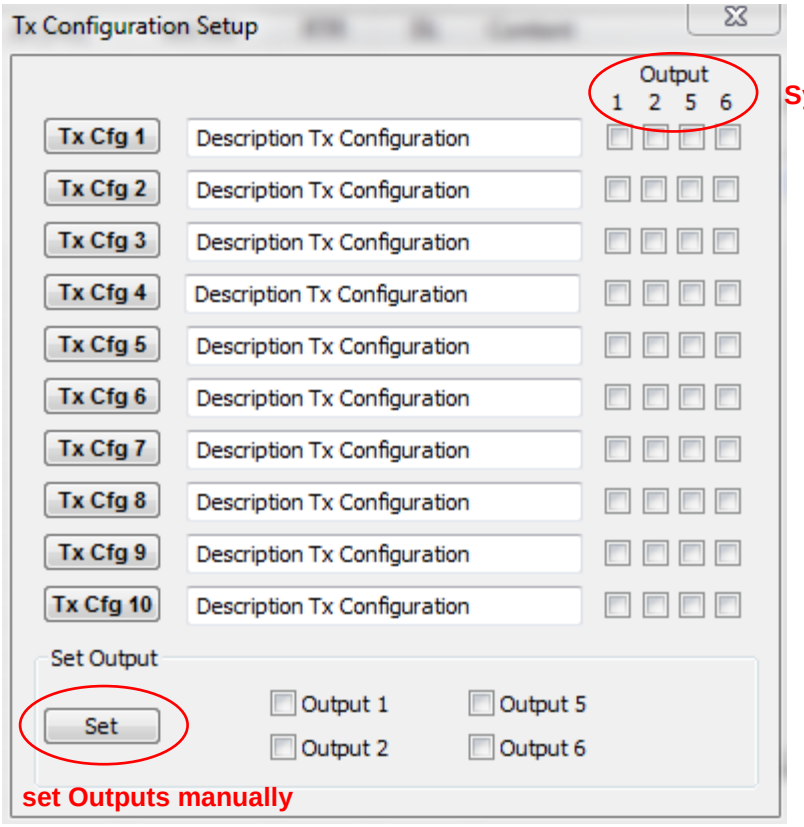
8.9.4 TX Configuration Window

Main Menu	Item	Description
Restbus	TX Configurations	Open TX Configuration Window

TX Configurations serve as shortcuts to pre-defined TX window variable values in the restbus simulation mode. If a stored TX Configuration is applied all TX variables in all TX windows are updated with the values which were applied at the moment when the TX configuration was stored.

When the Trigger Mode is not activated (Activate Osci Trigger check box in the Toolbar is not set)

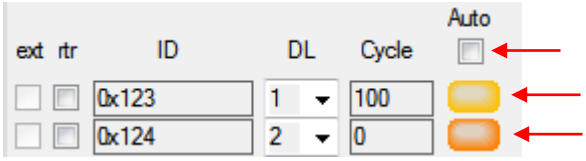
the 4 x Outputs can additionally be used as digital signal outputs. In this mode they either can be set manually by the user or they can be synchronized with the TX Configurations.



8.9.4.1 Save TX Configurations

- start the restbus simulation
- set all control values in all TX simulation windows to the desired values
- update the simulation bus values : only blue & green TX LEDs must appear

- if there are still yellow or orange LEDs click on the corresponding LED to update the bus values or click the auto-update check box



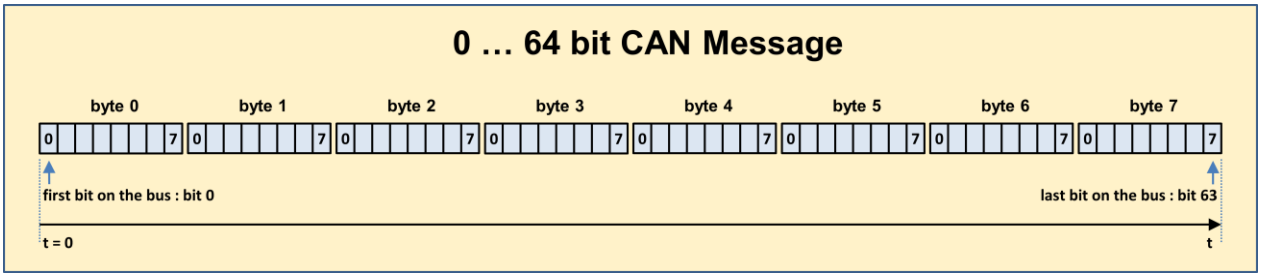
- set the description text for the new TX configuration
- if not in Trigger Mode set the 4 HW Output states if Outputs shall be synchronized with the TX configuration
- right mouse click the ‘Tx Cfg X’ Button to store the current configuration

8.9.4.2 Apply TX Configurations

- start the restbus simulation
- open the TX Configuration window
- left mouse click the ‘Tx Cfg X’ Button to apply the stored TX configuration

8.10 CAN Message Format : Little Endian / Big Endian Support

The user data within a CAN message is encapsulated in an up to 64 bit wide data field in which the least significant bit (bit 0) is sent first and the most significant bit (bit 63) is sent last :

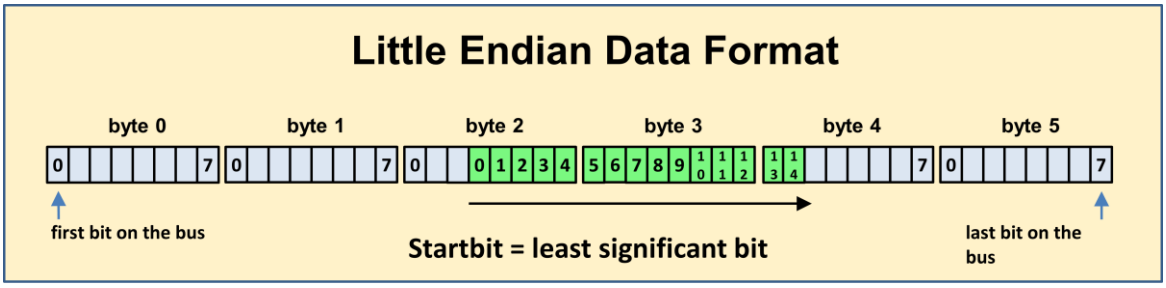


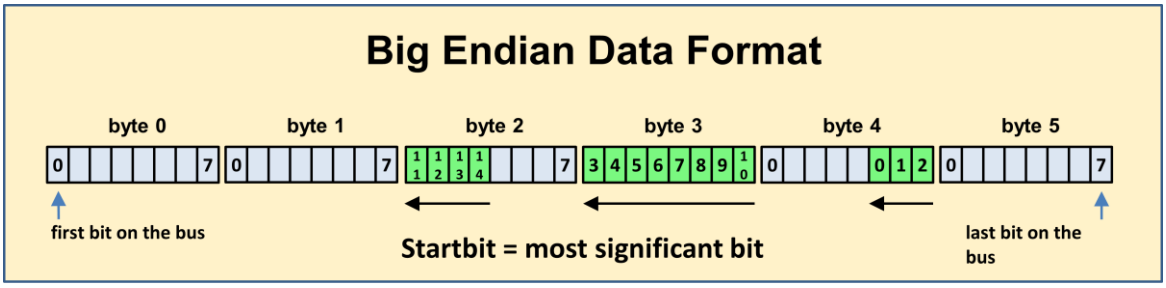
Within the 64 bit field the user variables can be coded in two different formats that are both supported by the CLDT Tool :

- Little Endian Data Format : least significant bit is sent first
- Big Endian Data Format : most significant bit is sent first

To explain the differences between the two formats the following example illustrates how a variable with a Bitlenth of 15 bits is coded within a 6 byte CAN message with

- Variables Startbyte : byte 2
- Variables Startbit : bit position 3 within the Startbyte byte 2

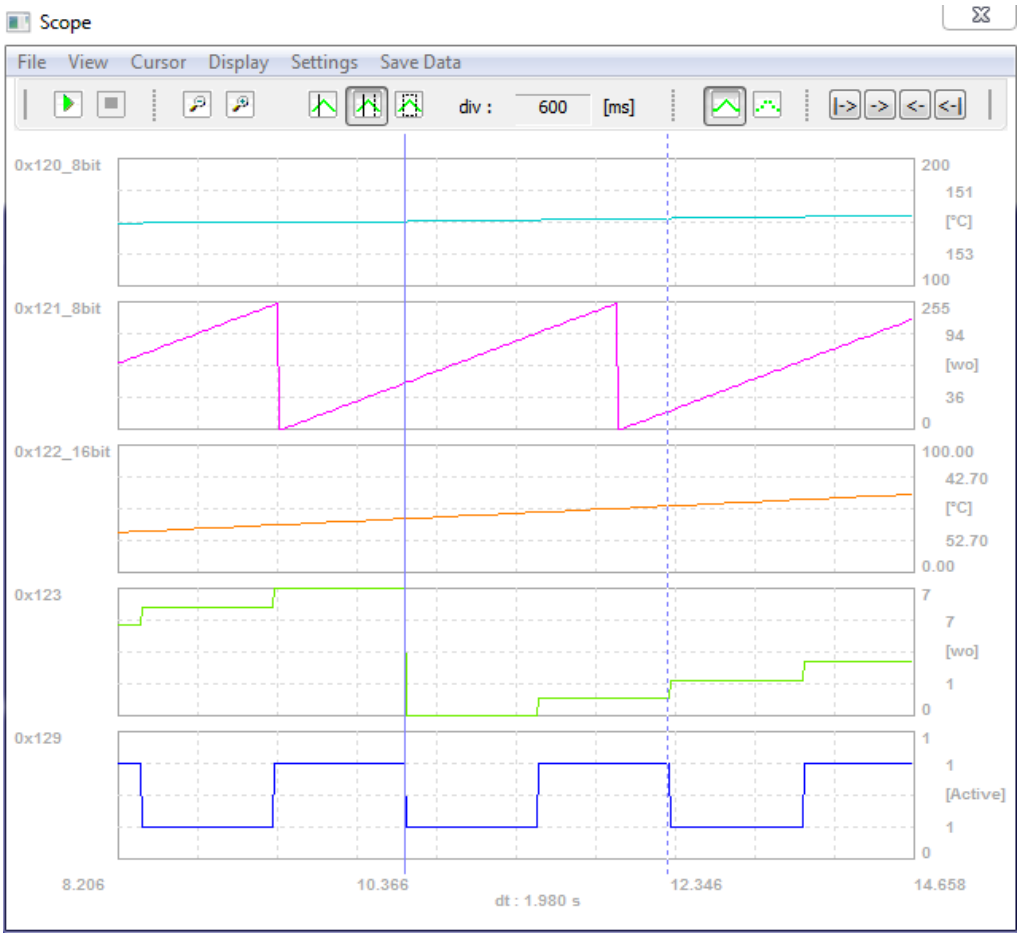




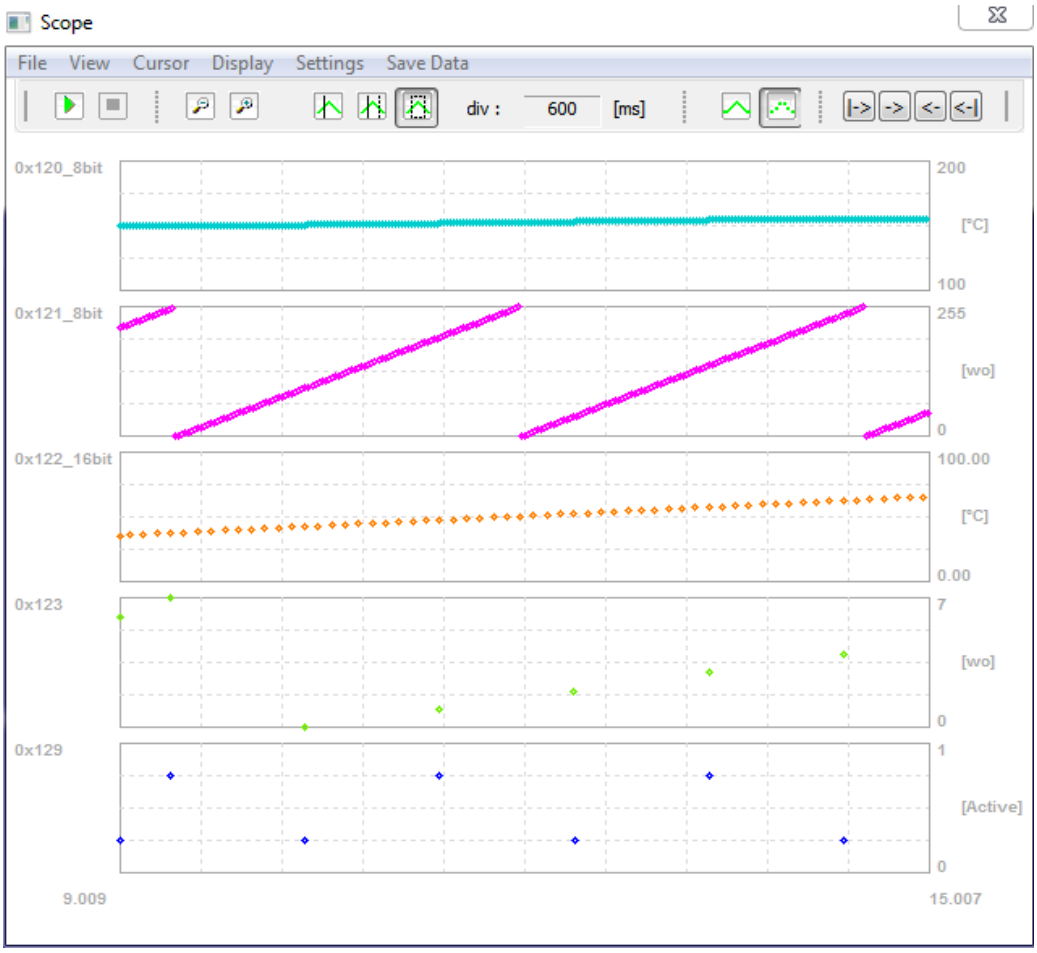
9 Scope Functionality

A Scope Functionality is integrated in the SW since Version A004

- monitor up to 8 bus signals in real time with a resolution of 1ms
- up to 50s monitoring memory
- display signals in Line Mode or in Dot Mode
- various zooming, measuring and storing possibilities



Scope Functionality in Line Mode



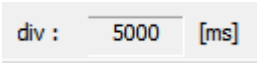
Scope Functionality in Dot Mode

9.1 Scope Toolbar Description

start / stop restbus simulation

zoom in / zoom out

switch between simple cursor, time cursor & rectangle zoom

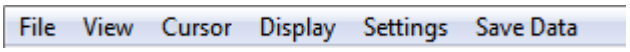


shows the current resolution in ms

switch between line mode & dot mode

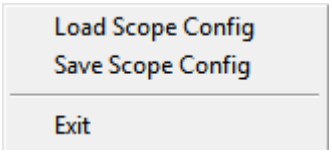
move signals to the right / left

9.2 Scope Menu Description



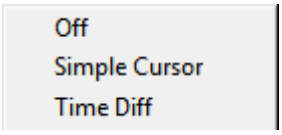
The Menu Items are described in detail in the following chapters :

9.2.1 File Menu



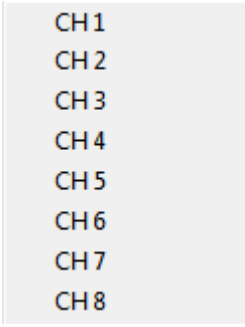
Menu Item	Description
Load Scope Config	Loads a scope configuration that has been saved before
Save Scope Configuration	Saves the current scope configuration
Exit	Closes the scope mode

9.2.2 Cursor Menu



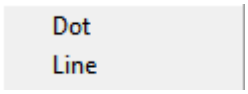
Menu Item	Description
Off	activates the rectangle zooming mode
Simple Cursor	activates the line cursor only
Time Diff	activates the line cursor and the dot cursor

9.2.3 View Menu



Menu Item	Description
CH1	Shows / Hides Scope Channel 1
CH2	Shows / Hides Scope Channel 2
CH3	Shows / Hides Scope Channel 3
CH4	Shows / Hides Scope Channel 4
CH5	Shows / Hides Scope Channel 5
CH6	Shows / Hides Scope Channel 6
CH7	Shows / Hides Scope Channel 7
CH8	Shows / Hides Scope Channel 8

9.2.4 Display Menu

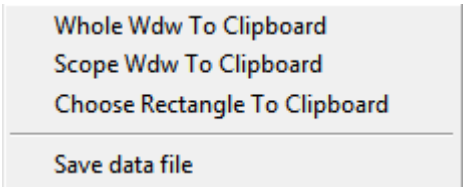


Menu Item	Description
Dot	activates dot displaying mode
Line	activates line displaying mode

9.2.5 Settings Menu

Menu Item	Description
Channels	activates the Channel Settings dialog to define signal parameters

9.2.6 Save Data Menu

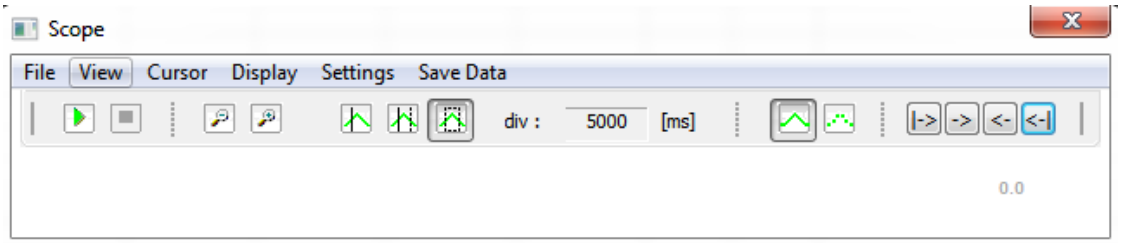


Menu Item	Description
Whole Wdw To Clipboard	Copies the whole scope window to the clipboard
Scope Wdw To Clipboard	Copies the signals only to the clipboard
Choose Rectangle To Clipboard	User to choose the rectangle content that is copied to the clipboard
Save data file	The signal data in [digit] is exported to a text file

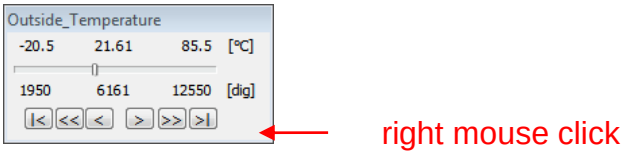
9.3 Use of Scope Mode

The Scope Functionality can only be used if a project is loaded and if RX or TX signals are defined for restbus simulation mode.

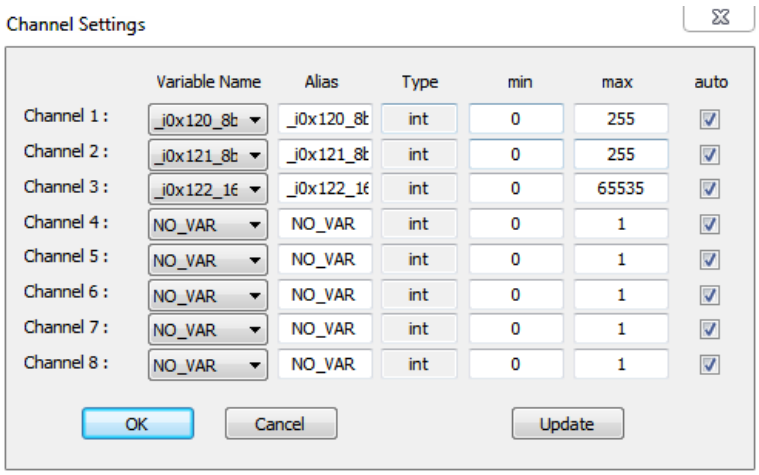
Then the main window menu →Restbus→Scope Active activates the Scope Mode.



Before starting the restbus simulation each signal that is defined in a RX or TX simulation window can be added to the scope.
A right click on the right bottom corner of the control and choosing 'Add To Scope' adds up to 8 signals to the scope.



Alternatively the signals can be manually chosen by the scope menu item →Settings→Channels and manually activated via the menu item →View



9.4 Start / Stop Restbus Simulation in Scope Mode

Use the toolbars Start & Stop Button to control the restbus simulation flow.

Use the mouse wheel to zoom in / out during restbus simulation

The mouse wheel can be used to switch between displaying modes

5s / div
2,5s / div
1s / div

during simulation

9.5 Analyzing Bus Signals when Restbus Simulation is stopped

9.5.1 Zooming Possibilities

Use the mouse wheel to zoom in / out

A resolution of 1ms/div can be reached if the mouse wheel is used for zooming

Use the time cursor mode to zoom in

mouse click to activate time cursor mode

position the line cursor with the left mouse button, position the dotted cursor with the right mouse button

mouse click the zoom button to zoom the signal content between the 2 cursors

Use the rectangle zoom to zoom in

mouse click to activate rectangle zoom mode

Left click the left top corner of the desired zooming rectangle, draw the mouse to the right bottom corner and then release the mouse button.

Use the zoom out button to zoom out

zoom out button

9.5.2 Measuring Possibilities

Simple Cursor Mode

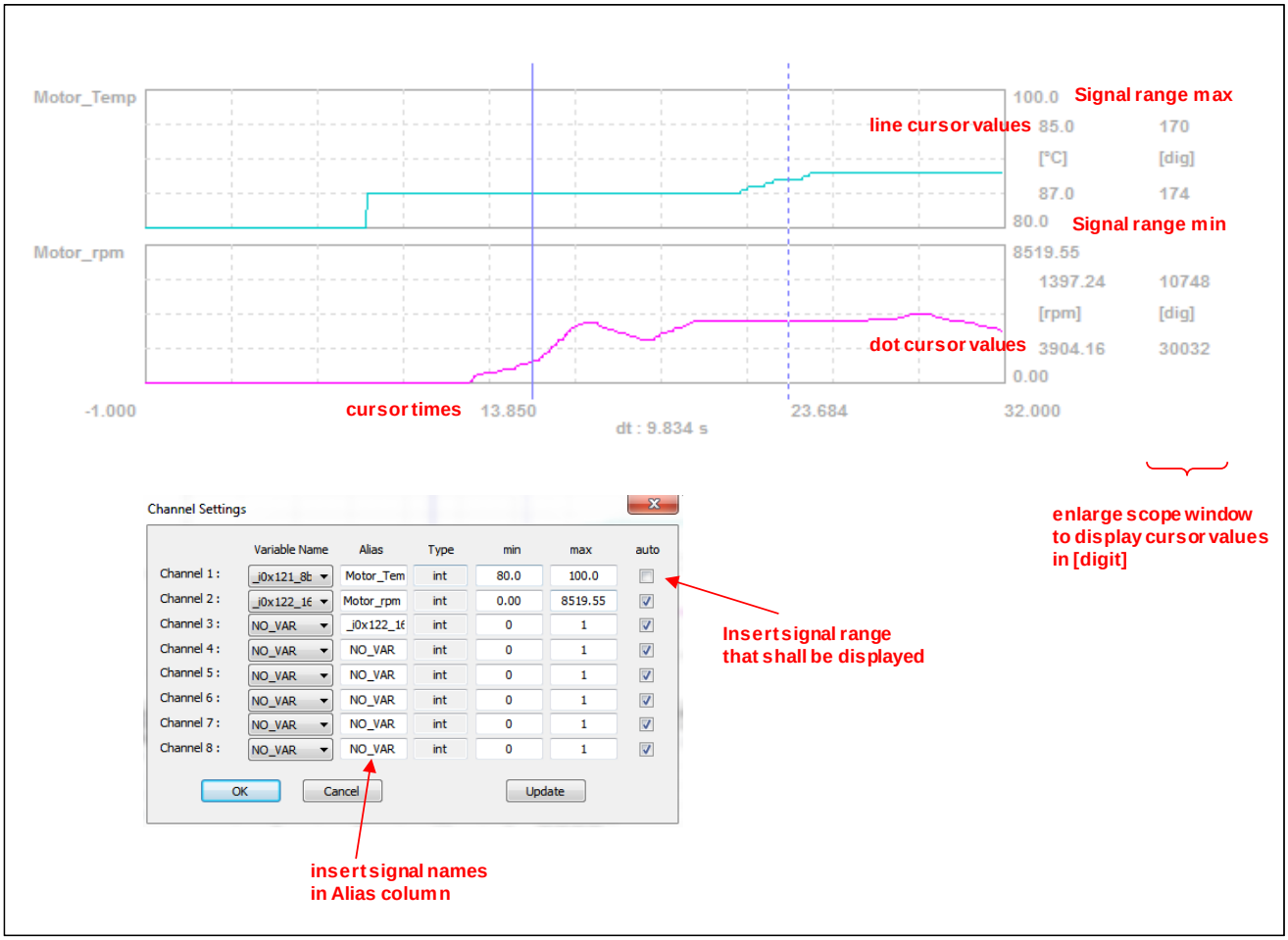
mouse click to activate simple cursor mode

position the line cursor with the left mouse button

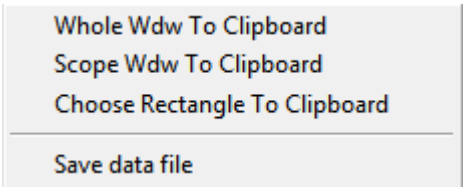
Time Cursor Mode

mouse click to activate time cursor mode

position the line cursor with the left mouse button
position the dotted cursor with the right mouse button



9.5.3 Storing Possibilities



Copy Scope Window to Clipboard

For documentation purpose the whole content or parts of the the scope window can be copied to the clipboard and afterwards inserted in another application like Word, Excel etc.

Use the scope menu items →Save Data

→Save Data→Whole Wdw To Clipboard

→Save Data→Scope Wdw To Clipboard

→Save Data→Choose Rectangle To Clipboard

Save Signal Data in Text File

The values in [digit] for each signal can be exported to a text file, which can be used by other applications like Excel to generate user specific diagrams or to superpose different signals.

Use the scope menu item

→Save Data→Save data file

10 RX – TX Automation with synchronization possibility between physical HW Outputs and CAN Bus Messages

RX-TX Automation Functionality is integrated in the SW since Version A004.

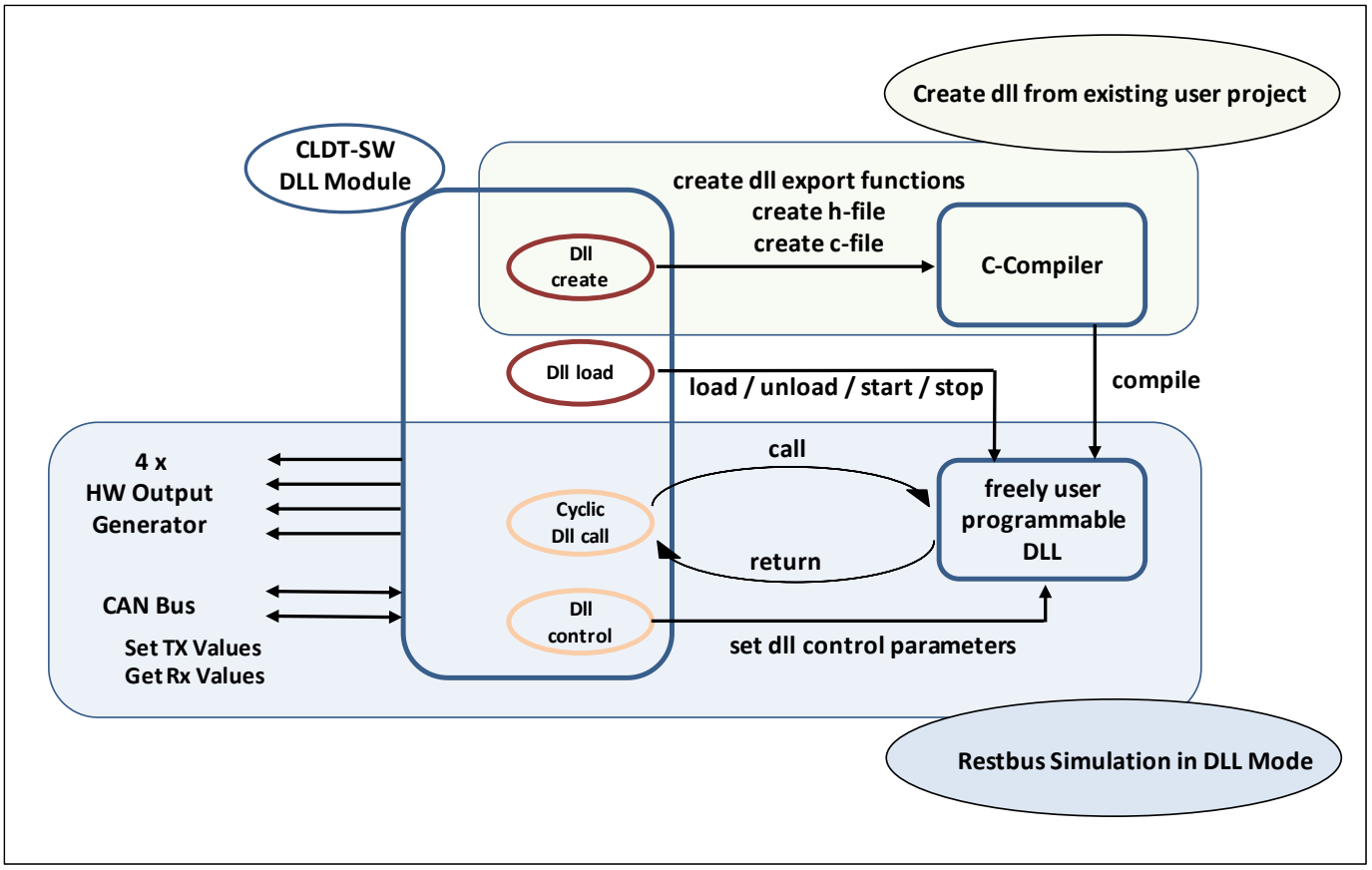
IMPORTANT : **If applications with increased timing requirements shall be simulated please refer to the chapter *'Timing Requirements under Windows Operating Systems'*.**

In any case ensure that the VCP Latency Timer for the USB device is set to a value < 3ms as described in the chapter *'Correct Setting for VCP Latency Timer'*

10.1 RX-TX Automation : Functional principle

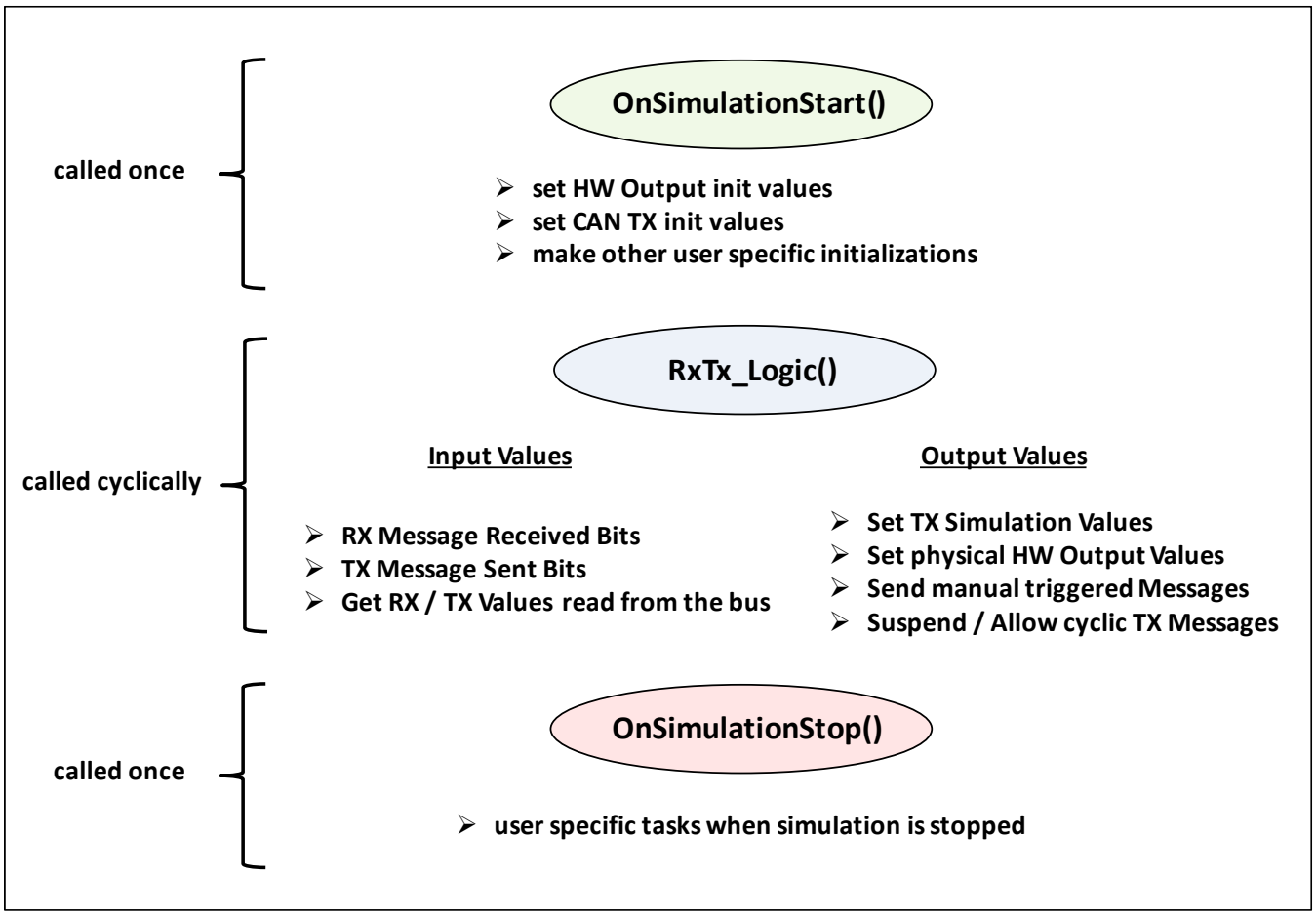
The user has the possibility to automatize the whole CAN Bus RX/TX transmission and thereby synchronize the RX/TX messages with 4 freely programmable HW Output generators. The following figures describe the functional principle of creating a user specific RX/TX logic.

- create a restbus simulation project with CLDT SW
- add a 'dll control panel' to the restbus simulation project
- create a dll project by using the 'dll-create panel'
- edit 3 central dll export functions with C-Compilers IDE
- Compile the dll project with C-Compiler
- load the compiled dll by using the 'dll-load panel'
- start the restbus simulation by using the 'dll-load panel'



10.2 Central DLL export functions to create user specific RX/TX logic

Three dll export functions are pre-defined and can be edited with any C-Compiler able to create regular 32bit dll projects :



Currently the LCC Compiler and the Microsoft VC++ Compiler are directly supported to automatically generate the skeletal structure with its integrated dll export functions for a dll project (see chapter *'Supported C-Compilers'*). In the future direct support for Borlands C++ Builder is planned.

But, theoretically, each C-Compiler able to create regular 32 bit DLLs can be used to create user specific applications.

10.3 DLL export functions for Data Exchange with CLDT SW

An overview of all dll export functions which must be provided by the dll is listed below. If the CLDT SW does not find all of the listed functions, the dll cannot be loaded.

dll export Functions Data Exchange	Description
void vSetOutputPointer(int *)	sets a pointer to an array in the CLDT SW
void vSetRxValuePointer(int*)	sets a pointer to an array in the CLDT SW
void vSetTxValueBusPointer(int*)	sets a pointer to an array in the CLDT SW
void vSetTxValueUserPointer(int*)	sets a pointer to an array in the CLDT SW
void vSetDllControlDataPointer(int*)	sets a pointer to an array in the CLDT SW
void vSetRxMsgPointer(int*)	sets a pointer to an array in the CLDT SW
void vSetTxMsgPointer(int*)	sets a pointer to an array in the CLDT SW
void vSetManTxMsgPointer(int*)	sets a pointer to an array in the CLDT SW
void vSetSuspendTxMsgPointer(int*)	sets a pointer to an array in the CLDT SW
void vSetRxMsg(int)	sets the BOOL value RX Message Received
void vSetTxMsg(int)	sets the BOOL value TX Message Sent

dll export Functions User Logic	Description
void vSimulationStart(void)	user specific initialization tasks before simulation start
int* iRxTx_Logic(INT64)	user specific RX/TX logic
void vSimulationStop(void)	user specific tasks after simulation stop

Remark :

All function parameters and function return values are explained in the chapter *'Example : RX/TX Automation with HW Output Synchronization'*

10.4 Supported C-Compilers for automated code generation

In principle each C-Compiler with the ability to create regular 32 bit DLLs can be used to create user specific applications.

However some C-Compilers like the LCC Compiler and the Microsoft VC++ Compiler are supported directly in order to create the skeletal structure of the dll project with its integrated dll export functions.

10.4.1 Free LCC-Win32 Compiler

Important Note : **The LCC Compiler is available for free only for non-commercial use !**
(see http://www.cs.virginia.edu/~lcc-win32/)

An example for a dll restbus simulation project, compiled with the LCC Compiler is shown in the chapter *'Example : RX/TX Automation with HW Output Synchronization'*

10.4.2 Microsoft VC++ Compiler

Both, the free VC++ Express edition and the commercial VC++ edition are supported for automatic code generation.

Examples for dll restbus simulation projects, compiled with the MSVC++ Compiler are shown in the chapter *'Example : RX/TX Automation with HW Output Synchronization'*

10.4.3 Borland C++ Builder

The Borland C++ Builder is not supported yet, but support is planned in the near future.

10.5 DLL Control Panel

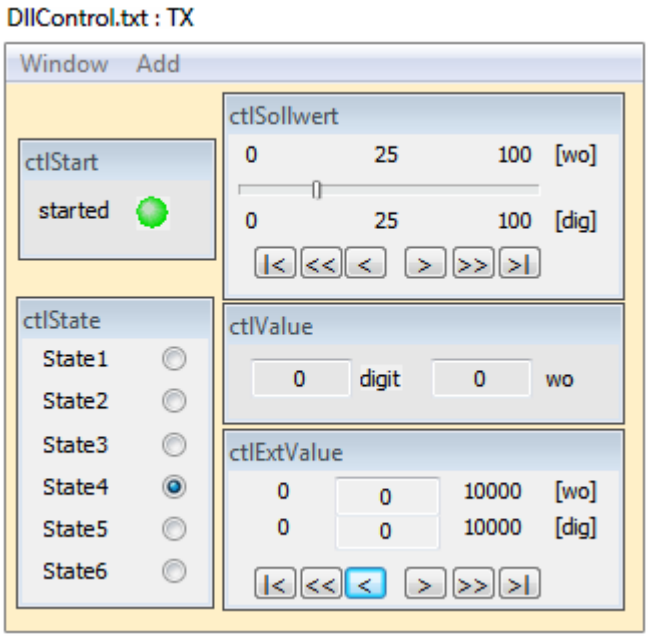
Purpose : control dll user logic program flow

All TX Controls which are available for TX simulation windows, can be added to the dll control panel.
Thus, during restbus simulation the user can interact and change the control panels TX values. The modified values are transmitted in real time to the dll.

Add a dll control panel to a restbus simulation project

Main menu item →dll→load dll Control Panel

Each time, just before the dll function iRxTx_Logic() is called, the dll control panel values are updated.
The dll function iRxTx_Logic() can access these values.



Example : dll control panel with 5 control variables

An example forfor the use of dll control panels is shown in the chapter *'Example : RX/TX Automation with HW Output Synchronization'*

10.6 DLL Create Panel

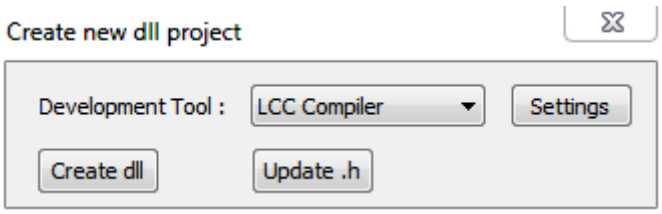
Purpose : ***create the skeletal structure of a dll project***

With the dll create panel the h-file and the c-file for a dll project can be generated automatically.

- All dll export functions are generated in the h-file
- All pointers for data access to the restbus simulation variables are defined and set correctly in the h-file
- The function bodies for the 3 central dll functions are generated in the c-file

Access the dll create panel

Main menu item →dll→create dll project

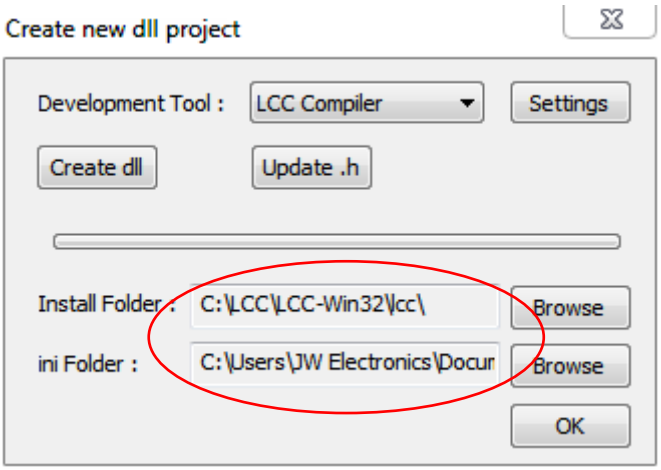


Create dll a new dll project is created.
h-file and c-file, as well as the project file (.prj) if the LCC Compiler is chosen.

Update.h just updates the h-file of an existing dll project. This will be necessary if the projects TX / RX simulation windows or dll control panel have been changed / modified after the creation of the dll project.

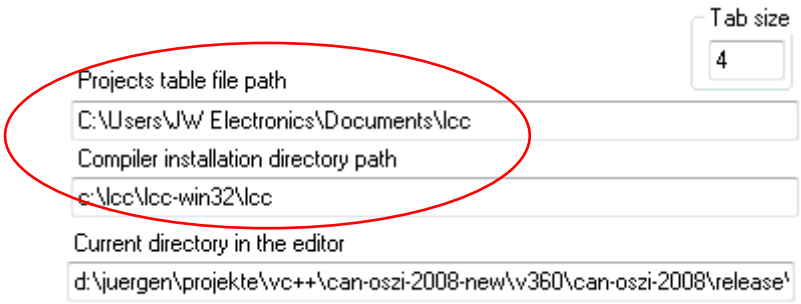
10.6.1 Settings for LCC Compiler

If the LCC Compiler is used, the complete dll-project is automatically generated by the CLDT SW.
Therefore some additional information about the LCC install folder and the LCC ini path are necessary :



After the successful installation of the LCC-Win32 Compiler check the LCC main menu item

→project→configuration

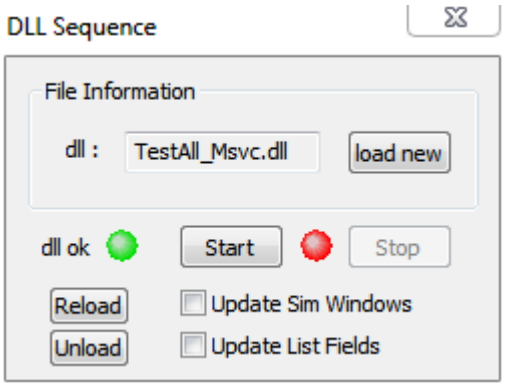


Insert ‘project table file path’ & ‘compiler installation directory path’ information in the corresponding edit fields in the CLDT SW and confirm with OK.

10.7 DLL Load Panel

Purpose : load and unload dll. Start and stop restbus simulation in dll mode

Access the dll load panel : Main menu item →dll→load dll



load new loads an existing dll. If all dll export functions are found the green 'dll ok LED' appears.

Start Start the restbus simulation in dll mode. RX/TX automation is generated by the dll.

Stop Stop the restbus simulation in dll mode.

Update Sim Wdw If the check box is not set, the simulation RX/TX windows are not updated. If applications with increased timing requirements are running this check box should not be set

Update List Fields If the check box is not set, the RX/TX simulation list fields are not updated. If applications with increased timing requirements are running this check box should not be set

Unload / Reload If a dll shall be modified by the C-Compiler the re-compilation will fail, if the dll is still loaded by the CLDT SW.
→ just unload the dll, re-compile the dll with the C-Compiler and then reload the dll.

11 Pre-defined TX Sequences

TX Sequence Functionality is integrated in the SW since Version A004.

With the TX Sequence Feature the user has the possibility to

- pre - define up to 2048 TX cycles with an **accuracy of 1ms / cycle**
- combine up to 25 TX IDs within one TX Sequence
- pre-define the CAN message contents for each TX cycle
- **synchronize 4 physical HW Outputs** to each TX cycle

- to guarantee the accuracy of 1ms for the synchronization between the HW Outputs and the CAN Bus Messages, the
- TX Bus cycles and the HW output settings are pre-defined in a .seq-file and
- downloaded to the CLDT Flash memory before sequence start

- An Excel Template and Excel Macros can be used to automatically generate the correct .seq-file format
- Examples for the .seq-file generation via Excel are delivered with the installer for the CLDT Box and are available for download at *http://www.jwe-electronics.com/mp1/downloads.php*

11.1 .seq-file format description

Line 1 … Line 37 :	**ID Header**
Line 38 … max. Line 2091	**TX Sequence data**

11.1.1 ID Header

The ID Header contains

- the Project name
- the Can Bus Baudrate
- Up to 25 different CAN Bus TX IDs that can be used for the TX cycles
- the initialization values for the cyclic CAN messages
- the initialization values for the 4 physical HW Outputs

In the following example 2 standard CAN messages are defined :

- one cyclic message with the ID 0x200, data length 4 and the initialization values 01 02 03 04 for byte0 … byte3
- one non cyclic message with the ID 0x100, data length 8
- the physical Outputs 1 & 5 are initialized to ‘0’ at sequence start
- the physical Outputs 2 & 6 are initialized to ‘1’ at sequence start
- the Baudrate is defined to 1 Mbit/s
- the project name is defined as TestSequence_2msg.seq

```
1  ------------------------------------------------------------------------------------------
2  Project_Name    TestSequence_2msg.seq
3  ------------------------------------------------------------------------------------------
4  Baudrate        0
5  ------------------------------------------------------------------------------------------
6  NbMsg   ext rtr ID  DL  cyc cyc_time       b0  b1  b2  b3  b4  b5  b6  b7
7  ------------------------------------------------------------------------------------------
8  1   std data    0x100   8   no  0       0   0   0   0   0   0   0   0
9  2   std data    0x200   4   yes 200     1   2   3   4   0   0   0   0
10 3   ext data    0xFFFFFFFF  0   no  0       0   0   0   0   0   0   0   0
11 4   ext data    0xFFFFFFFF  0   no  0       0   0   0   0   0   0   0   0
12 5   ext data    0xFFFFFFFF  0   no  0       0   0   0   0   0   0   0   0
13 6   ext data    0xFFFFFFFF  0   no  0       0   0   0   0   0   0   0   0
14 7   ext data    0xFFFFFFFF  0   no  0       0   0   0   0   0   0   0   0
15 8   ext data    0xFFFFFFFF  0   no  0       0   0   0   0   0   0   0   0
16 9   ext data    0xFFFFFFFF  0   no  0       0   0   0   0   0   0   0   0
17 10  ext data    0xFFFFFFFF  0   no  0       0   0   0   0   0   0   0   0
18 11  ext data    0xFFFFFFFF  0   no  0       0   0   0   0   0   0   0   0
19 12  ext data    0xFFFFFFFF  0   no  0       0   0   0   0   0   0   0   0
20 13  ext data    0xFFFFFFFF  0   no  0       0   0   0   0   0   0   0   0
21 14  ext data    0xFFFFFFFF  0   no  0       0   0   0   0   0   0   0   0
22 15  ext data    0xFFFFFFFF  0   no  0       0   0   0   0   0   0   0   0
23 16  ext data    0xFFFFFFFF  0   no  0       0   0   0   0   0   0   0   0
24 17  ext data    0xFFFFFFFF  0   no  0       0   0   0   0   0   0   0   0
25 18  ext data    0xFFFFFFFF  0   no  0       0   0   0   0   0   0   0   0
26 19  ext data    0xFFFFFFFF  0   no  0       0   0   0   0   0   0   0   0
27 20  ext data    0xFFFFFFFF  0   no  0       0   0   0   0   0   0   0   0
28 21  ext data    0xFFFFFFFF  0   no  0       0   0   0   0   0   0   0   0
29 22  ext data    0xFFFFFFFF  0   no  0       0   0   0   0   0   0   0   0
30 23  ext data    0xFFFFFFFF  0   no  0       0   0   0   0   0   0   0   0
31 24  ext data    0xFFFFFFFF  0   no  0       0   0   0   0   0   0   0   0
32 25  ext data    0xFFFFFFFF  0   no  0       0   0   0   0   0   0   0   0
33 ------------------------------------------------------------------------------------------
34 Op1 Op2 Op5 Op6
35 ------------------------------------------------------------------------------------------
36 0   1   0   1
37 ------------------------------------------------------------------------------------------
```

IDs in ascending order

Requested Lines :

Line 2 : Project Name

Line 4 : Baudrate (1Mbit/s / (n+1)) 0 : 1Mbit/s 1 : 500 kBit/s …

Line 8 … Line 32 **CAN TX ID's in ascending order (ID1 < ID2 < ID3 …)**

NbMsg	number of message
ext, std	message format
rtr / data	Request or Data
ID	Message ID in hex format with prefix 0x
DL	Data Length in byte
yes / no	cyclic message → yes, non cyclic message → no
0 … 65535 (ms)	cycle time if cyclic message
0 … 255	b0 … b7 : initialization values for cyclic messages

Line 36 0, 1 HW Output 1, 2, 5, 6 initialization values

11.1.2 TX Sequence Data

The TX Sequence Data contains up to 2048 TX cycles

- number of the CAN ID whose message content shall be changed
- 16 bit Δt value in ms : dt from cycle n to cycle n+1
- 8 byte message content
- 1 byte physical HW Output Setting

- If only n (n < 2048) TX cycles are used, the values for the not used cycles (n+1 … 2048) must be filled with 12 x 255 (see exemplary .seq-files)

In the following example the 2 CAN IDs from the ID Header above are used

- the non-cyclic CAN message shall be sent every 150 ms with the 8 byte content that is defined in the corresponding TX cycle
- the content of the cyclic CAN message shall be changed every 900ms by applying the 4 byte content defined in the corresponding TX cycle
- the output states shall be changed from 1001 → 1000 → 1100

	NbMsg	dt_h	dt_l	b0	b1	b2	b3	b4	b5	b6	b7	Op
44	1	0	150	1	222	4	5	6	4	5	6	5
45	1	0	150	0	222	4	5	6	4	5	6	5
46	1	0	150	2	222	4	5	6	4	5	6	5
47	1	0	150	0	222	4	5	6	4	5	6	5
48	1	0	150	3	222	4	5	6	4	5	6	5
49	1	0	150	0	222	4	5	6	4	5	6	5
50	2	0	10	1	2	3	4	0	0	0	0	5
51	1	0	140	0	222	2	5	3	4	5	8	5
52	1	0	150	4	222	2	5	3	4	5	8	5
53	1	0	150	0	222	2	5	3	4	5	8	5
54	1	0	150	5	222	2	5	3	4	5	8	5
55	1	0	150	0	222	2	5	3	4	5	8	5
56	1	0	150	6	222	2	5	3	4	5	8	5
57	2	0	10	2	3	4	5	0	0	0	0	5
58	1	0	140	7	222	4	5	6	4	5	6	1
59	1	0	150	0	222	4	5	6	4	5	6	3
60	1	0	150	8	222	4	5	6	4	5	6	3
61	1	0	150	0	222	4	5	6	4	5	6	3
62	1	0	150	9	222	4	5	6	4	5	6	3
63	1	0	150	0	222	4	5	6	4	5	6	3
64	2	0	10	3	4	5	6	0	0	0	0	3
65	1	0	140	0	222	2	5	3	4	5	8	3
66	1	0	150	4	222	2	5	3	4	5	8	3
67	1	0	150	0	222	2	5	3	4	5	8	3
68	1	0	150	5	222	2	5	3	4	5	8	3

Line 44 … Line n **Maximum 2048 lines (44 … 2091) : 1 line = 1 TX Sequence Cycle**

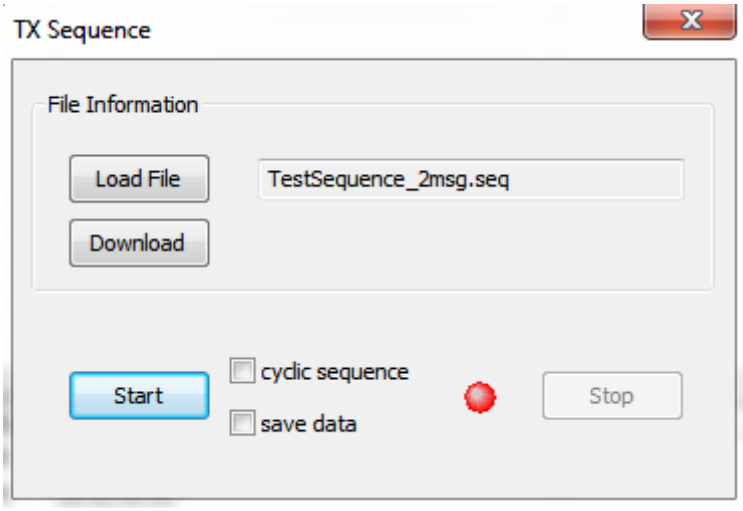
Line n+1 … Line 2091 255 255 255 255 255 255 255 255 255 255 255 255

NbMsg		number of message sent in next TX cycle
dt_h	0 … 255	Δt_h : High byte
dt_l	0 … 255	Δt_L : Low byte
		$\Delta t_{ges} = (\Delta t_h * 255) + \Delta t_L$: time difference to next TX cycle
b0	0 … 255	TX value for byte 0 for next TX cycle
b1	0 … 255	TX value for byte 1 for next TX cycle
b2	0 … 255	TX value for byte 2 for next TX cycle
b3	0 … 255	TX value for byte 3 for next TX cycle
b4	0 … 255	TX value for byte 4 for next TX cycle
b5	0 … 255	TX value for byte 5 for next TX cycle
b6	0 … 255	TX value for byte 6 for next TX cycle
b7	0 … 255	TX value for byte 7 for next TX cycle
Op	0 … 15	HW Output values for next TX cycle
		bit0 : Output1, bit1 : Output2, bit2 : Output5, bit3 : Output6

11.2 Use of TX Sequences

Once the .seq-file is created, the TX Sequence can be applied by following the procedure below :

- Choose the TX Sequence Control Panel in the main menu
  →Restbus→Tx Sequence



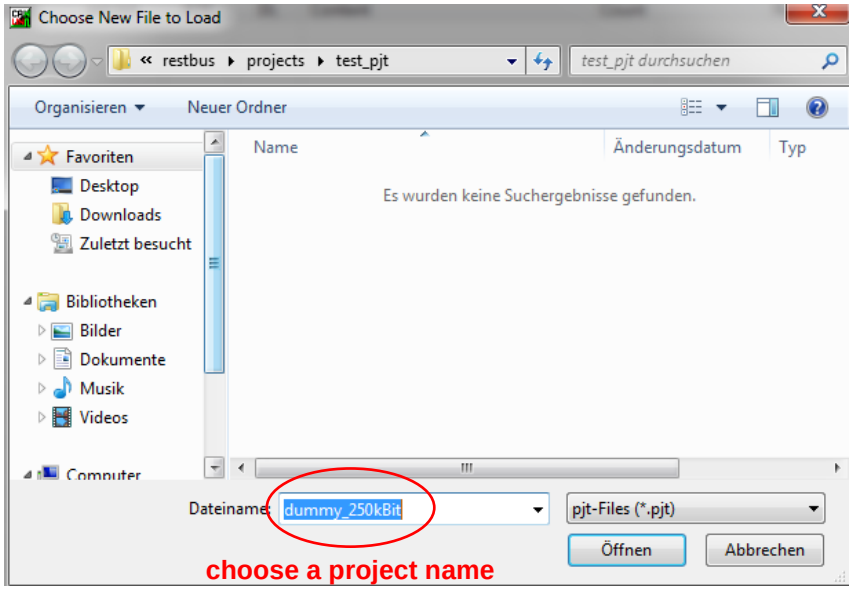
- Load the .seq-file
- choose if the TX sequence shall be applied cyclically or just once
- choose if the CAN data shall be saved in a file
- start the TX Sequence

12 Example : Can Trace

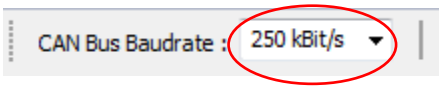
A new CAN Trace (without restbus simulation) shall be started. The trace shall monitor CAN bus data with a transfer rate of 250 kBit/s.

To start the new trace the following 3 steps must be performed :

1) Load a new project → Restbus → New Project

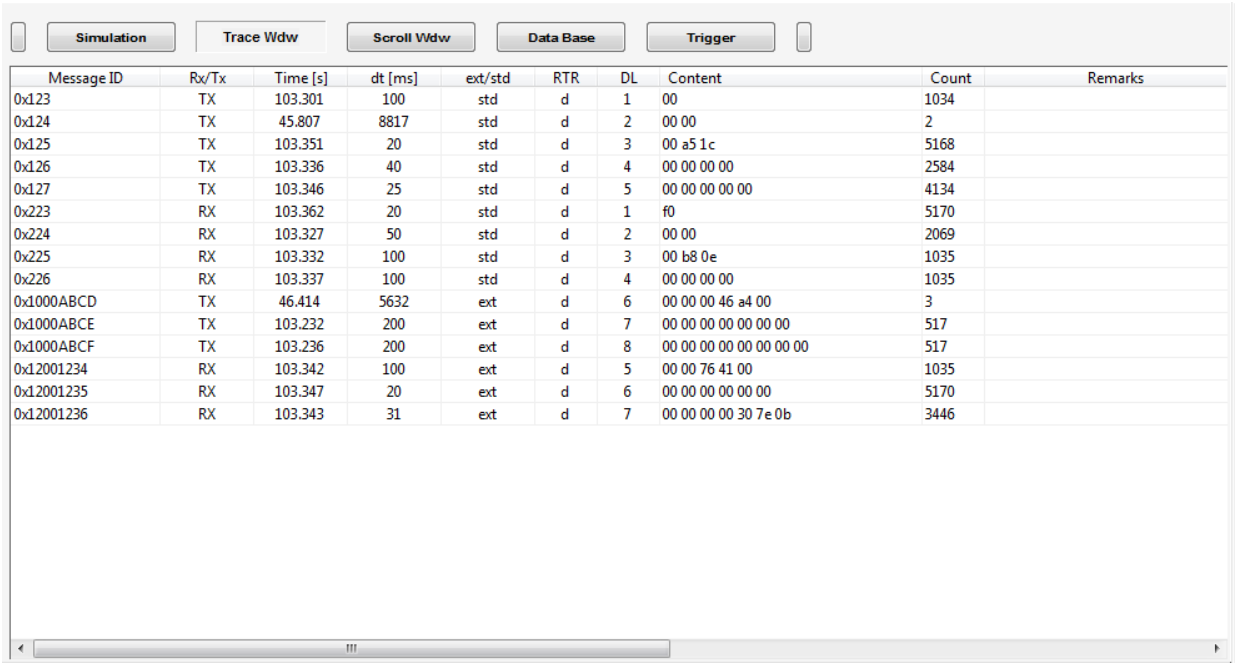


2) Set the Can Bus Baudrate in the Toolbar to 250 kBit/s



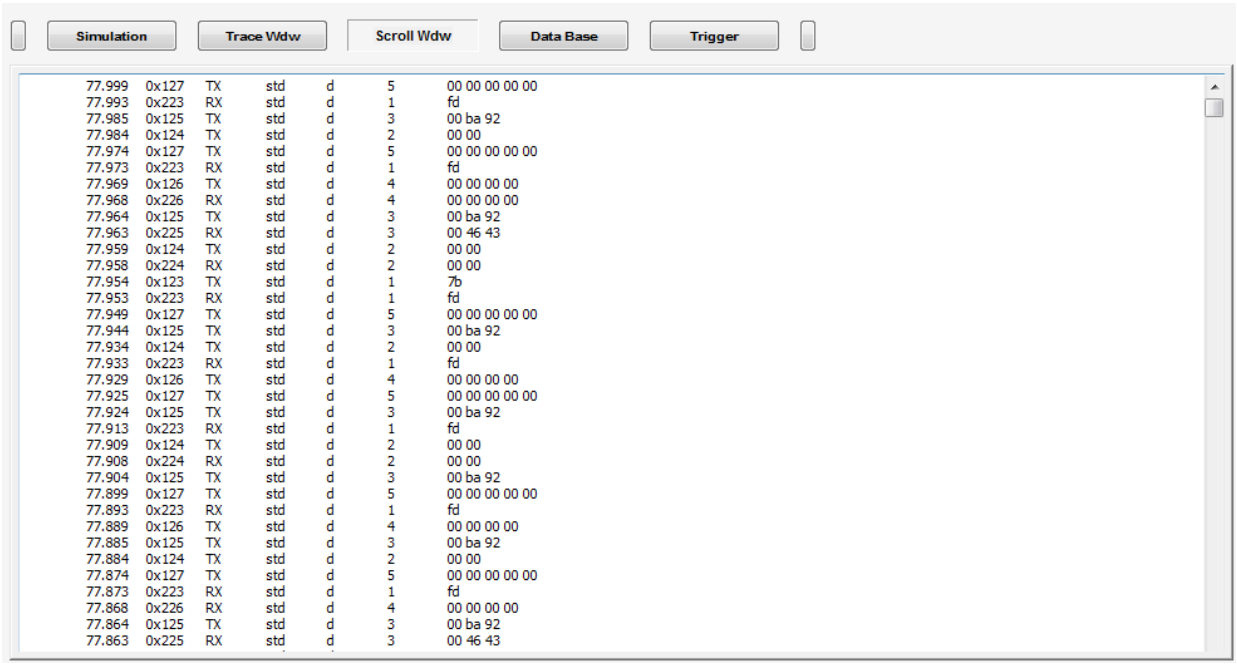
3) Start the CAN Trace with the Toolbar shortcut

If other ECUs or other CLDT Boxes are connected to the bus and are sending data with the correct baudrate of 250 kBit/s the CAN Bus data is displayed in the Trace Window and in the Scroll Window as shown below :



Message ID	Rx/Tx	Time [s]	dt [ms]	ext/std	RTR	DL	Content	Count	Remarks
0x123	TX	103.301	100	std	d	1	00	1034	
0x124	TX	45.807	8817	std	d	2	00 00	2	
0x125	TX	103.351	20	std	d	3	00 a5 1c	5168	
0x126	TX	103.336	40	std	d	4	00 00 00 00	2584	
0x127	TX	103.346	25	std	d	5	00 00 00 00 00	4134	
0x223	RX	103.362	20	std	d	1	f0	5170	
0x224	RX	103.327	50	std	d	2	00 00	2069	
0x225	RX	103.332	100	std	d	3	00 b8 0e	1035	
0x226	RX	103.337	100	std	d	4	00 00 00 00	1035	
0x1000ABCD	TX	46.414	5632	ext	d	6	00 00 00 46 a4 00	3	
0x1000ABCE	TX	103.232	200	ext	d	7	00 00 00 00 00 00 00	517	
0x1000ABCF	TX	103.236	200	ext	d	8	00 00 00 00 00 00 00 00	517	
0x12001234	RX	103.342	100	ext	d	5	00 00 76 41 00	1035	
0x12001235	RX	103.347	20	ext	d	6	00 00 00 00 00 00	5170	
0x12001236	RX	103.343	31	ext	d	7	00 00 00 00 30 7e 0b	3446	

CAN Bus data Output in the Trace Window




```
77.999  0x127  TX  std  d  5  00 00 00 00 00
77.993  0x223  RX  std  d  1  fd
77.985  0x125  TX  std  d  3  00 ba 92
77.984  0x124  TX  std  d  2  00 00
77.974  0x127  TX  std  d  5  00 00 00 00 00
77.973  0x223  RX  std  d  1  fd
77.969  0x126  TX  std  d  4  00 00 00 00
77.968  0x226  RX  std  d  4  00 00 00 00
77.964  0x125  TX  std  d  3  00 ba 92
77.963  0x225  RX  std  d  3  00 46 43
77.959  0x124  TX  std  d  2  00 00
77.958  0x224  RX  std  d  2  00 00
77.954  0x123  TX  std  d  1  7b
77.953  0x223  RX  std  d  1  fd
77.949  0x127  TX  std  d  5  00 00 00 00 00
77.944  0x125  TX  std  d  3  00 ba 92
77.934  0x124  TX  std  d  2  00 00
77.933  0x223  RX  std  d  1  fd
77.929  0x126  TX  std  d  4  00 00 00 00
77.925  0x127  TX  std  d  5  00 00 00 00 00
77.924  0x125  TX  std  d  3  00 ba 92
77.913  0x223  RX  std  d  1  fd
77.909  0x124  TX  std  d  2  00 00
77.908  0x224  RX  std  d  2  00 00
77.904  0x125  TX  std  d  3  00 ba 92
77.899  0x127  TX  std  d  5  00 00 00 00 00
77.893  0x223  RX  std  d  1  fd
77.889  0x126  TX  std  d  4  00 00 00 00
77.885  0x125  TX  std  d  3  00 ba 92
77.884  0x124  TX  std  d  2  00 00
77.874  0x127  TX  std  d  5  00 00 00 00 00
77.873  0x223  RX  std  d  1  fd
77.868  0x226  RX  std  d  4  00 00 00 00
77.864  0x125  TX  std  d  3  00 ba 92
77.863  0x225  RX  std  d  3  00 46 43
```

CAN Bus data Output in the Scroll Window

13 Example : Can Restbus Simulation

A new Restbus Simulation Project with the following design parameters shall be developed :

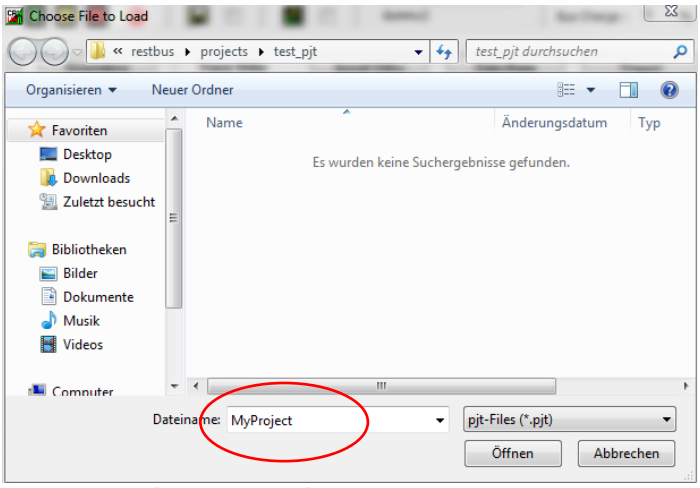
		DL	Cycle [ms]
Project Name	MyProject		
CAN Bus Baudrate	250 kBit/s		
CAN RX Messages	0x223	1	20
	0x224	2	50
	0x225	3	100
	0x226	4	100
	0x12001234	5	100
	0x12001235	6	20
	0x12001236	7	30
CAN TX Messages	0x123	1	100
	0x124	2	0
	0x125	3	20
	0x126	4	40
	0x127	5	25
	0x1000ABCD	6	0
	0x1000ABCE	7	200
	0x1000ABCF	8	200
Nbr Rx Windows	1		
Nbr Tx Windows	2		

In each of the RX & TX windows shall be placed 3 controls :

- ➔ One 16 bit int value
- ➔ One 1 bit Flag value with check box symbol
- ➔ One 1 bit Flag value with LED symbol

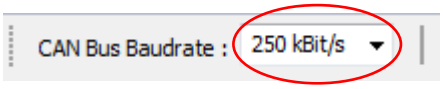
13.1 Step 1 : Load a new Project and choose the Baudrate

In the Main Menu choose → Restbus → New Project

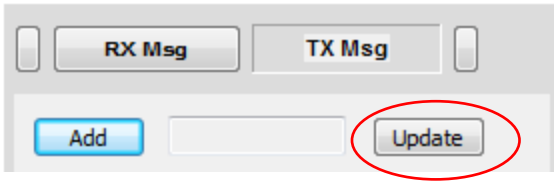


choose a project name

Set the Can Bus Baudrate in the Toolbar to 250 kBit/s

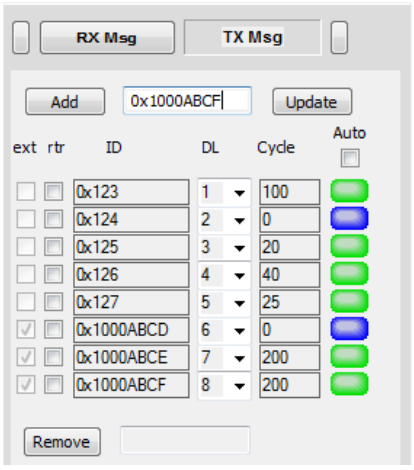


Save the basic project parameters : click the 'update-Button' in the CAN RX / TX Message Window



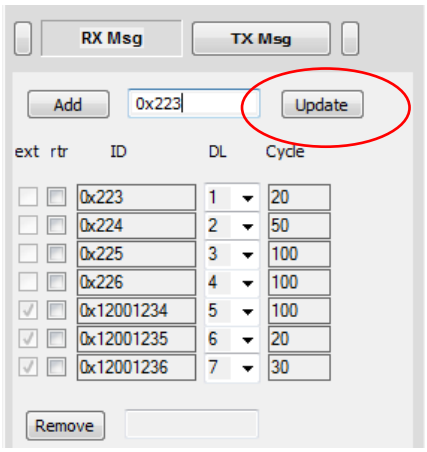
13.2 Step 2 : Edit the CAN RX and CAN TX IDs

First click on the TX Msg – register field and add all TX messages with their correct data length and cycle time settings



➔ Green color indicates a cyclic CAN message
➔ Blue color indicates a manually triggered CAN message

Then click on the RX Msg – register field and add all RX messages with their correct data length and cycle time settings



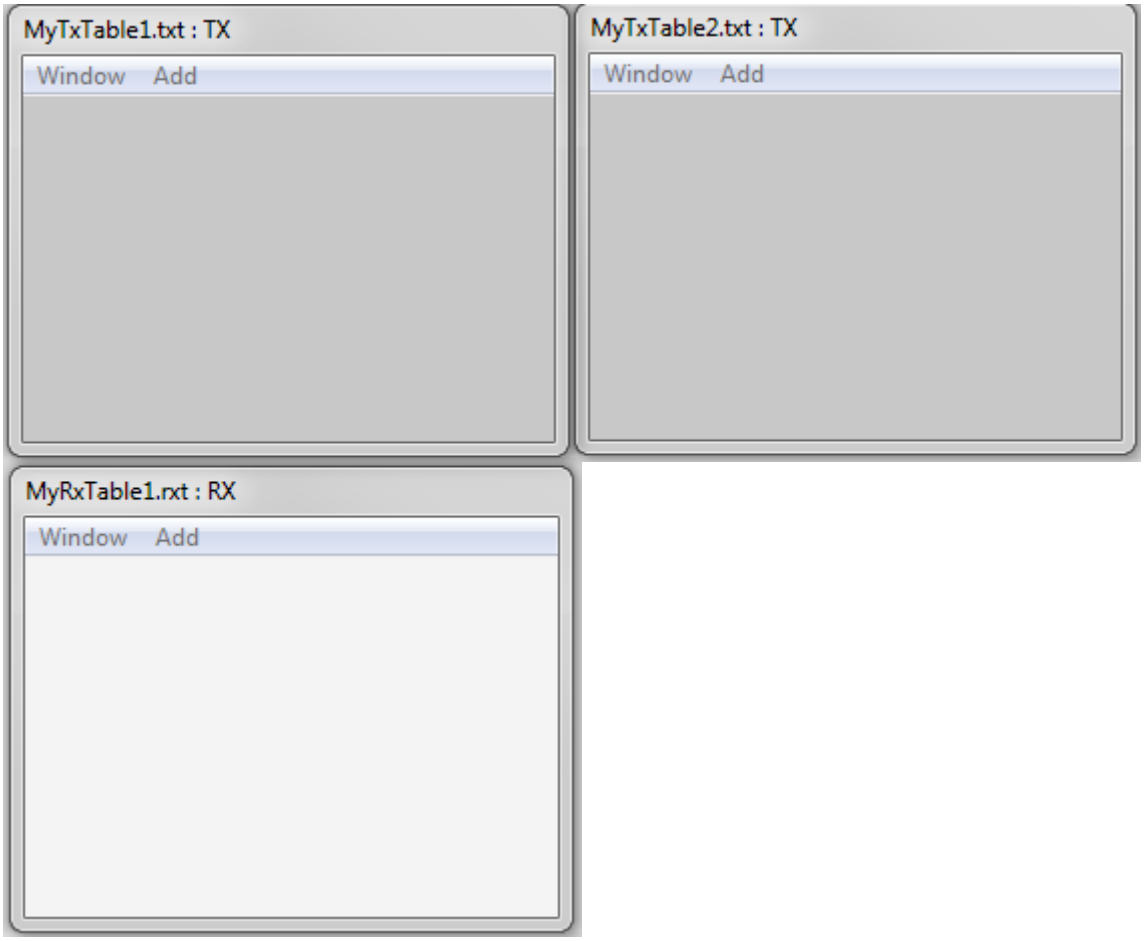
Finally save the CAN Message parameters to the project data with the 'update-Button'.

A 'red update-Button' indicates that the project must be updated before next simulation start.

13.3 Step 3 : Add one RX and two TX Windows each with 3 variables

In the Main Menu choose → Restbus → Load RX Table
In the Main Menu choose → Restbus → Load TX Table

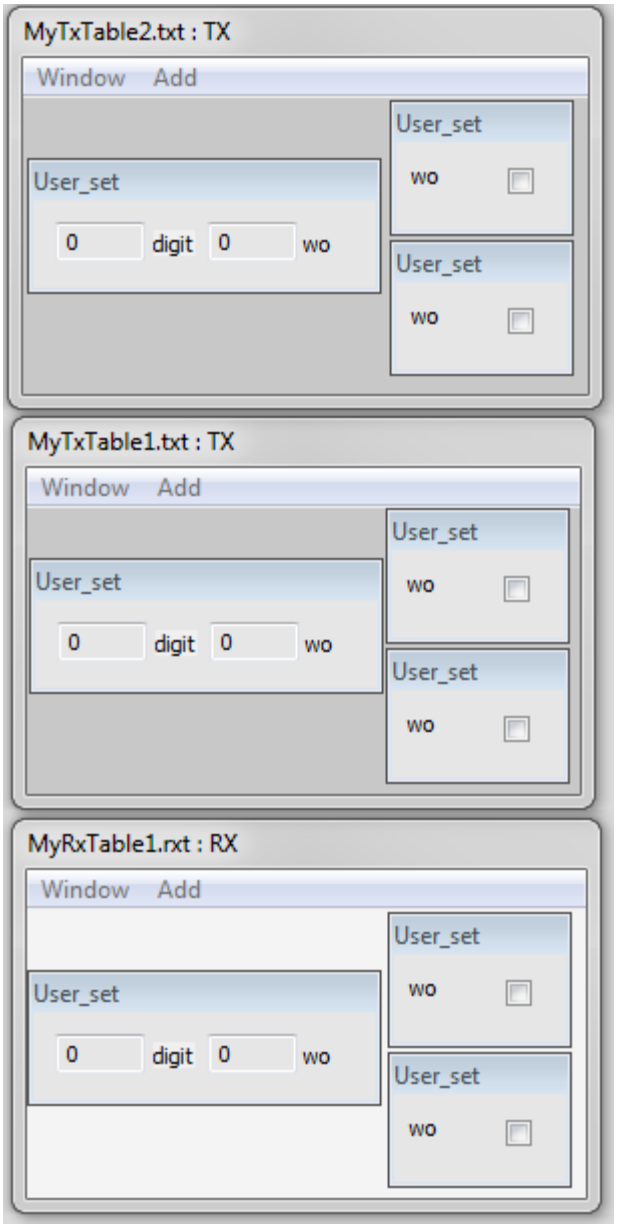
Choose a unique name for each of the RX / TX Windows. The project disposes now of 3 empty simulation windows.



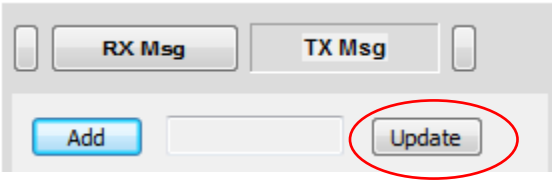
➔ RX Windows are white colored
➔ TX Windows are gray-colored

For each of the 3 windows add now 1 value control and 2 flag controls

→ Add → Value
→ Add → Flag

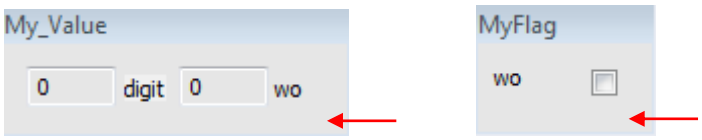


Save the new project configuration with the ‘update-Button’.



13.4 Step 4 : Add the Properties for each of the controls

Open the Properties Dialog for each of the variables by a double mouse click on the right bottom corner of the flag control or value control



and define for each of them

- ➔ the CAN message ID in which the variable is coded
- ➔ the Variable Name
- ➔ Startbyte, Startbit & Datalength
- ➔ the data format (Little Endian or Big Endian)
- ➔ the unit that shall be shown
- ➔ for flag controls : if LED symbol is used or not
- ➔ for value controls : if a calculated value (a, b) are used or not

Please refer to the chapters *CAN Message Format* , *Flag Properties* or *Value Properties* for a more detailed description of the above mentioned parameters.

Note : **Currently no check is done if the data for bit positioning within a CAN message, inserted by the user, is coherent. The user is responsible by himself for correct message formats, correct data length and that no variable interferences appear.**

In a future SW upgrade the implementation of an auto-check for correct CAN format is planned

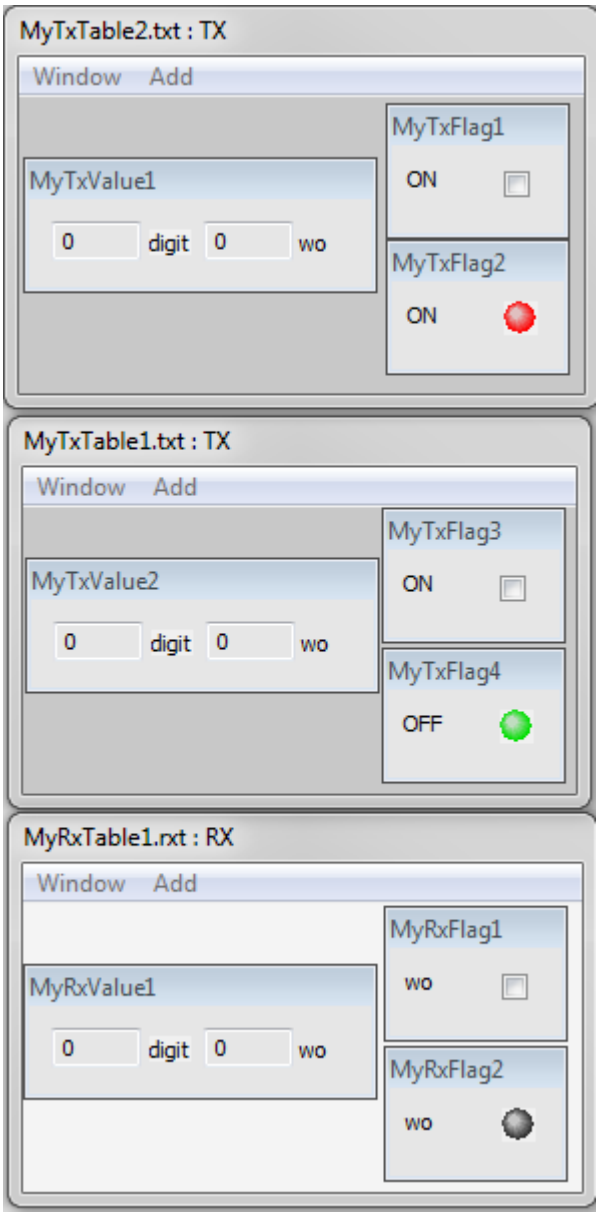
Tip : Each time the CAN Message RX or TX IDs are modified, the **project must be updated** via the 'Update Button'. Only after an project update new CAN IDs are displayed in the Properties Windows.

In the given example some fictive data has been inserted in the properties windows of each control.

The result is one RX window and two TX windows, each with 3 controls :

one value

two flags, one check box and one LED.

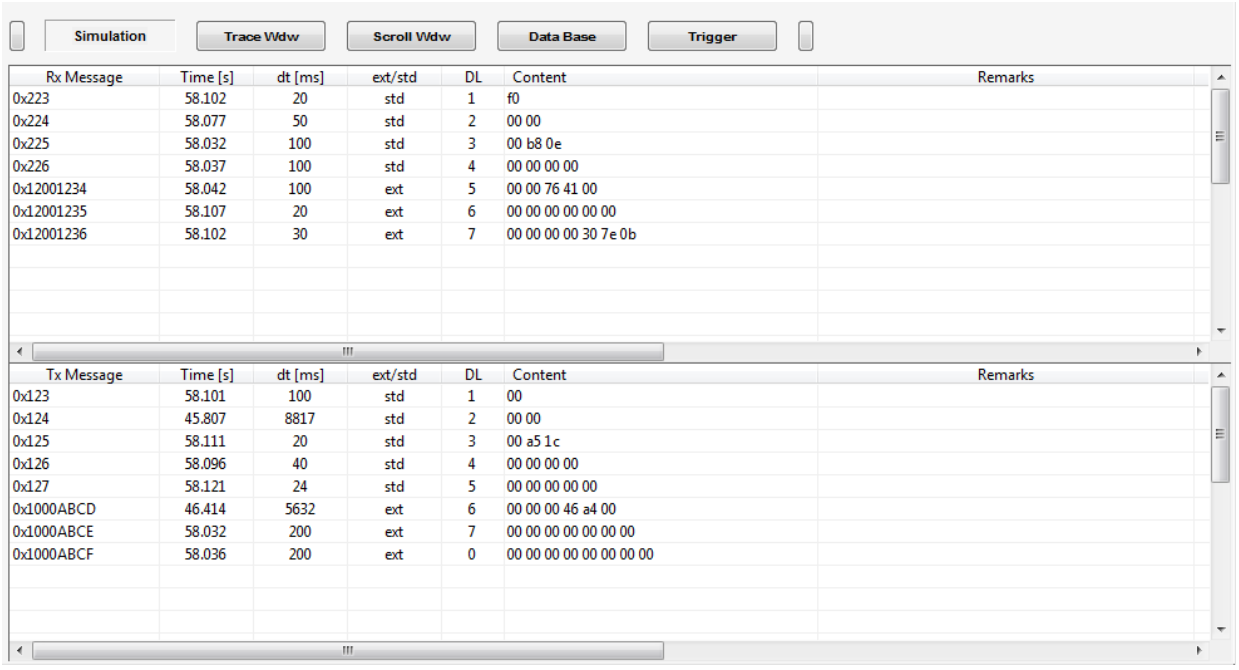


13.5 Step 5 : Start the Restbus Simulation

- Start the CAN Restbus Simulation with the Toolbar shortcut

If other ECUs or other CLDT Boxes are connected to the bus and are sending data with the correct baudrate of 250 kBit/s the CAN Bus data is now displayed in the Trace Window and in the Scroll Window as already described in the chapter *Example : CAN Trace* .

Additionally the RX Simulation and TX Simulation data is displayed in the Simulation Window as shown below :



The Simulation Window is updated approximately every 200ms, slightly depending on the real bus charge

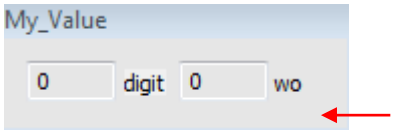
13.6 Step 6 : Update RX Simulation Variables

The RX Simulation Windows with the defined RX Variables are updated approximately every 200ms with the data that is read in real time from the bus.

13.7 Step 7 : Update TX Simulation Variables

To change the values of a TX flag control a simple click on the check box symbol or on the LED symbol switches between its 2 states (true & false).

To change the value of a TX value control, open the Value - dialog by a **right mouse click** on the right bottom corner of the value, insert the new value and confirm with 'ENTER' or escape with 'ESC'.

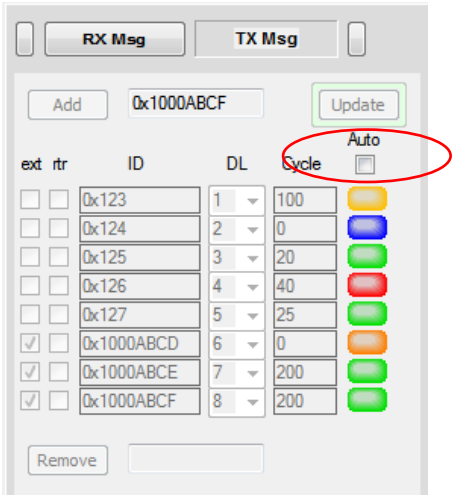


right mouse click

Immediate update or update with mouse click on the rectangular LEDs

Depending on the 'Auto check box' in the CAN RX TX Messages window an update of the real bus TX values is done

- ➔ Immediately if the auto check box is set
- ➔ user triggered if the auto check box is not set



Meaning of LED Colors :

blue :	**non-cyclic message :**	**message is updated**
green :	**cyclic message :**	**message is updated**
orange :	**non-cyclic message :**	**message not updated**
yellow :	**cyclic message :**	**message not updated**
red :	**cyclic message :**	**message suspended**

➔ suspend cyclic messages : click the right mouse button on the LED
➔ re-allow cyclic messages : click the right mouse button on the LED again

If the auto check box is not set and TX variable values are modified, the color of the cyclic messages changes from green to yellow, respectively from blue to orange for the non-cyclic messages

Yellow and orange colors indicate, that the CAN messages with the corresponding IDs have not been updated with their new values until now.

- ➔ a simple mouse click on the yellow or orange button updates the corresponding TX message contents.

13.8 Step 8 : Hardware Output Usage

If the 4 HW signal Outputs shall be used please verify, that signal collisions as described in the chapter *'HW Outputs & CAN Bus Connection'* cannot appear.

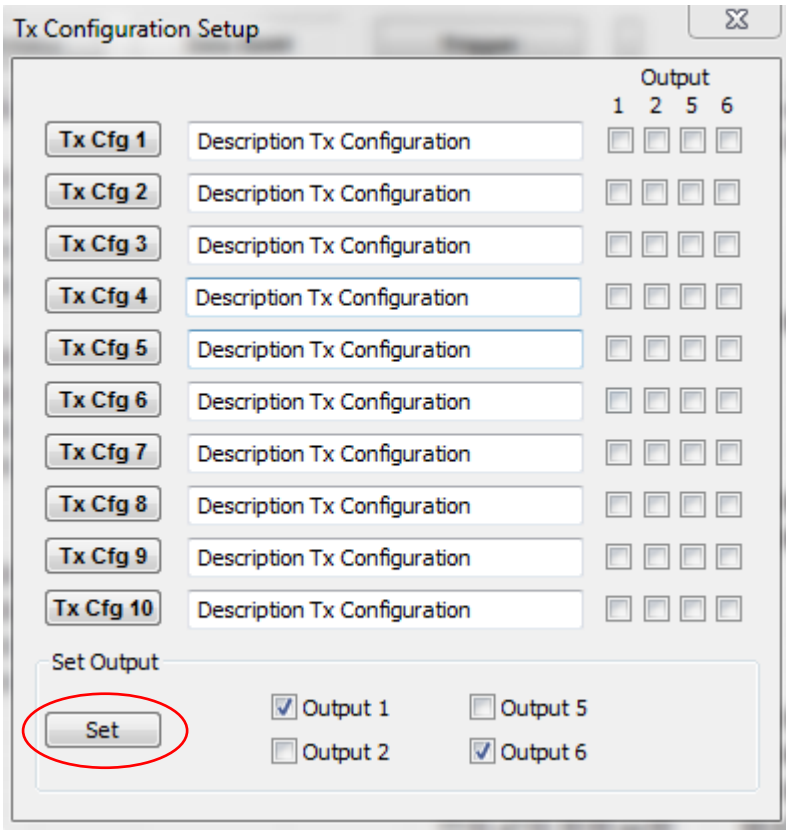
To start a Restbus Simulation in HW Output Mode the 'Trigger Mode Active' check box must not be set when starting the simulation.

Otherwise the HW Outputs will be considered as HW Trigger Outputs (see chapter *'Trigger Mode'*)

To set the HW Outputs manually the TX Configuration Window must be loaded

→ Restbus → TX Configurations

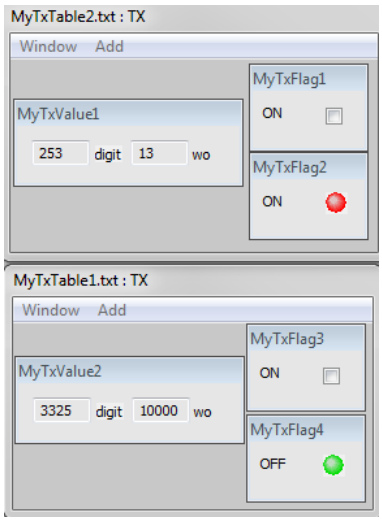
At the bottom of the TX Configuration Window the 4 HW Outputs can be set manually at any time by setting or resetting the desired outputs and then applying a simple mouse click on the 'Set Button'.



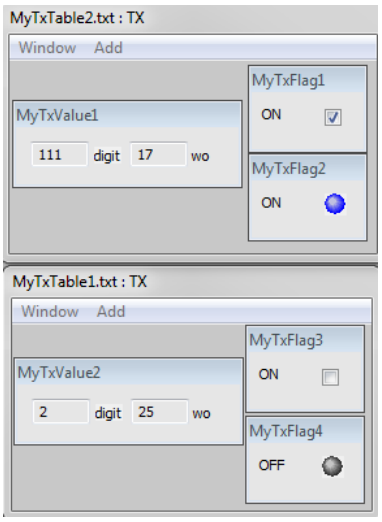
13.9 Step 9 : Use of TX Configurations when the Simulation is running

The following two TX Configurations shall be stored as Tx Cfg 1 and Tx Cfg 2. Additionally in

Tx Cfg 1 the HW Outputs 1 & 3 shall be set
Tx Cfg 2 the HW Outputs 2 & 4 shall be set

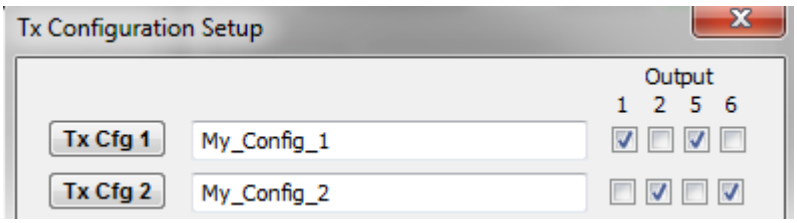


Config 1

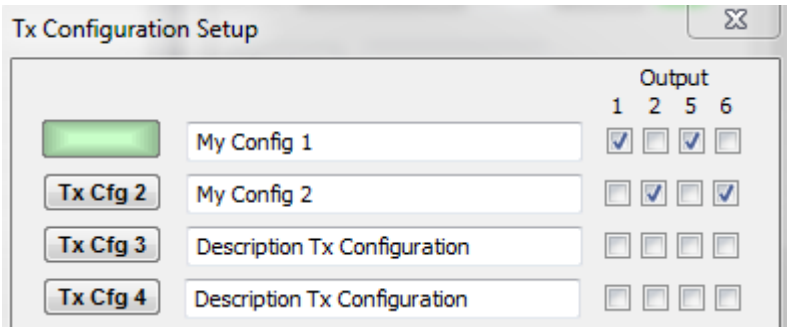


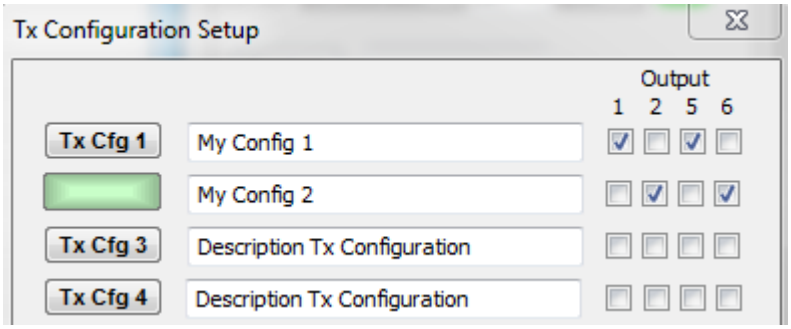
Config 2

- set all values and flags as desired for configuration 1 **and update** the TX messages manually, if the Auto-update check box is not set.
- set the HW Outputs 1 & 3
- right mouse click the ‘Tx Cfg 1’ Button
- set all values and flags as desired for configuration 2 **and update** the TX messages manually, if the Auto-update check box is not set.
- set the HW Outputs 2 & 4
- right mouse click the ‘Tx Cfg 2’ Button



At any time, when the restbus simulation is running, the TX Configurations can be chosen by a simple mouse click on the corresponding Tx Cfg X Button :





Tip : TX Configurations can only be stored correctly if the **Restbus Simulation is running** and if all TX bus messages are **already updated** with the new values !

14 Example : Can Data Base Trace

An example for a dbcj - Data Base is distributed with the Installation CD
(see folder \dbc\ 00-MonitoringStations.dbcj)

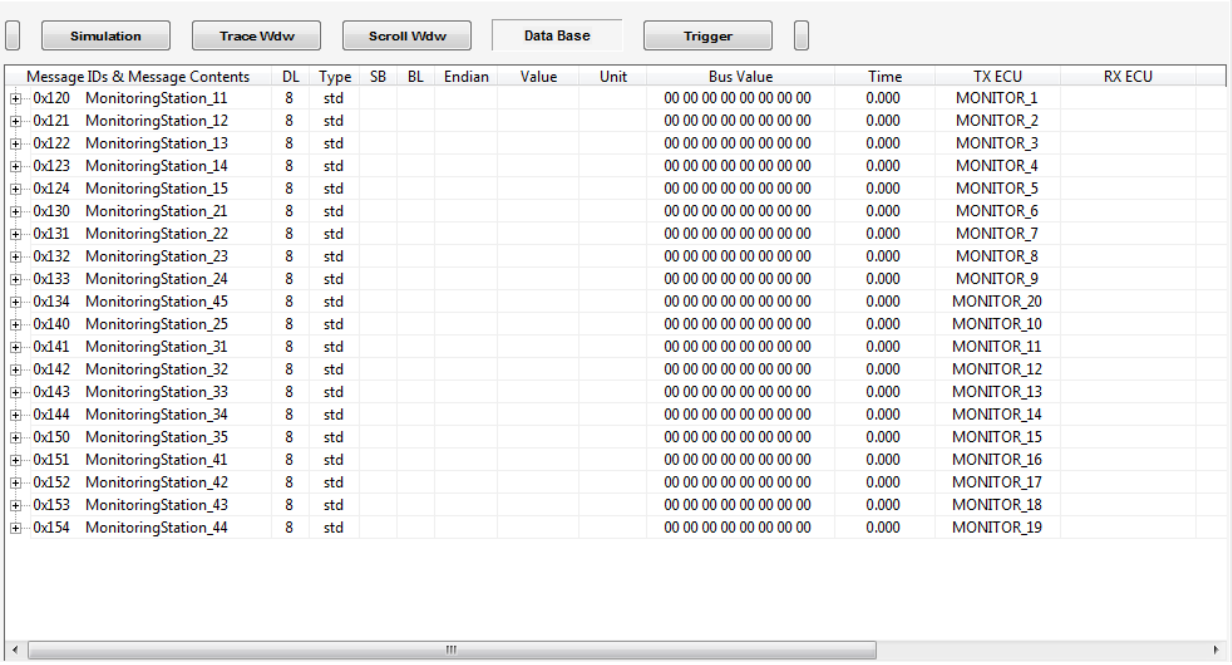
- CAN Bus Baudrate : 500 kBit/s
- 20 different CAN messages, each with 4 signals

(about editing user specific dbcj-files see chapter *Data Base Editor*)

Start a CAN Data Base Trace

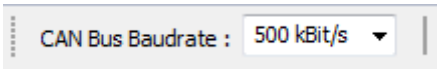
Main Menu	Item	Description
File	Import Data Base	Opens a new data base in the Data Base Window

The data base is displayed in a Message-Signal Tree-Structure :



Message IDs & Message Contents	DL	Type	SB	BL	Endian	Value	Unit	Bus Value	Time	TX ECU	RX ECU
0x120 MonitoringStation_11	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_1	
0x121 MonitoringStation_12	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_2	
0x122 MonitoringStation_13	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_3	
0x123 MonitoringStation_14	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_4	
0x124 MonitoringStation_15	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_5	
0x130 MonitoringStation_21	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_6	
0x131 MonitoringStation_22	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_7	
0x132 MonitoringStation_23	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_8	
0x133 MonitoringStation_24	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_9	
0x134 MonitoringStation_45	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_20	
0x140 MonitoringStation_25	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_10	
0x141 MonitoringStation_31	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_11	
0x142 MonitoringStation_32	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_12	
0x143 MonitoringStation_33	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_13	
0x144 MonitoringStation_34	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_14	
0x150 MonitoringStation_35	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_15	
0x151 MonitoringStation_41	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_16	
0x152 MonitoringStation_42	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_17	
0x153 MonitoringStation_43	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_18	
0x154 MonitoringStation_44	8	std						00 00 00 00 00 00 00 00	0.000	MONITOR_19	

→ Verify the correct Can Bus Baudrate in the Toolbar



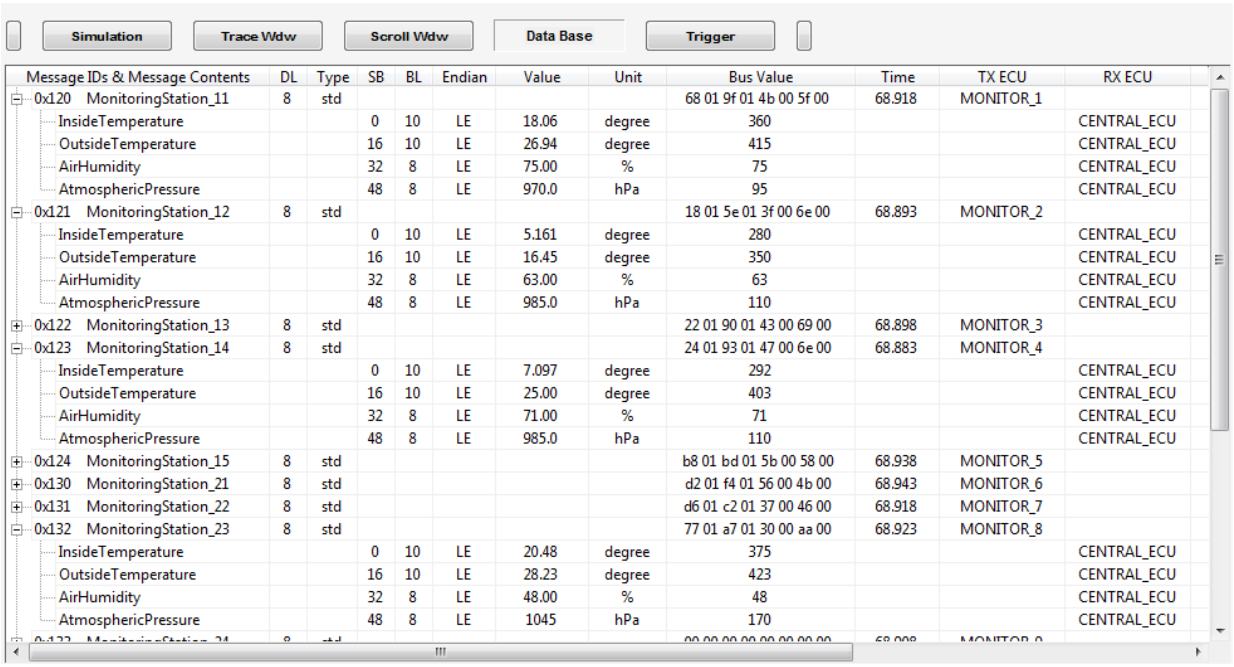
→ Store CAN Data (optional)

If the trace data shall be stored set the ‘Save Data check box’ in the Toolbar

→ Start a new CAN Trace

If the CAN IDs defined in the dbc-file appear on the bus, the messages & signal contents are displayed in the Data Base Window.

Content update is done every 200ms approximately.



Message IDs & Message Contents	DL	Type	SB	BL	Endian	Value	Unit	Bus Value	Time	TX ECU	RX ECU
0x120 MonitoringStation_11	8	std						68 01 9f 01 4b 00 5f 00	68.918	MONITOR_1	
InsideTemperature			0	10	LE	18.06	degree	360			CENTRAL_ECU
OutsideTemperature			16	10	LE	26.94	degree	415			CENTRAL_ECU
AirHumidity			32	8	LE	75.00	%	75			CENTRAL_ECU
AtmosphericPressure			48	8	LE	970.0	hPa	95			CENTRAL_ECU
0x121 MonitoringStation_12	8	std						18 01 5e 01 3f 00 6e 00	68.893	MONITOR_2	
InsideTemperature			0	10	LE	5.161	degree	280			CENTRAL_ECU
OutsideTemperature			16	10	LE	16.45	degree	350			CENTRAL_ECU
AirHumidity			32	8	LE	63.00	%	63			CENTRAL_ECU
AtmosphericPressure			48	8	LE	985.0	hPa	110			CENTRAL_ECU
0x122 MonitoringStation_13	8	std						22 01 90 01 43 00 69 00	68.898	MONITOR_3	
0x123 MonitoringStation_14	8	std						24 01 93 01 47 00 6e 00	68.883	MONITOR_4	
InsideTemperature			0	10	LE	7.097	degree	292			CENTRAL_ECU
OutsideTemperature			16	10	LE	25.00	degree	403			CENTRAL_ECU
AirHumidity			32	8	LE	71.00	%	71			CENTRAL_ECU
AtmosphericPressure			48	8	LE	985.0	hPa	110			CENTRAL_ECU
0x124 MonitoringStation_15	8	std						b8 01 bd 01 5b 00 58 00	68.938	MONITOR_5	
0x130 MonitoringStation_21	8	std						d2 01 f4 01 56 00 4b 00	68.943	MONITOR_6	
0x131 MonitoringStation_22	8	std						d6 01 c2 01 37 00 46 00	68.918	MONITOR_7	
0x132 MonitoringStation_23	8	std						77 01 a7 01 30 00 aa 00	68.923	MONITOR_8	
InsideTemperature			0	10	LE	20.48	degree	375			CENTRAL_ECU
OutsideTemperature			16	10	LE	28.23	degree	423			CENTRAL_ECU
AirHumidity			32	8	LE	48.00	%	48			CENTRAL_ECU
AtmosphericPressure			48	8	LE	1045	hPa	170			CENTRAL_ECU

15 Example : Trigger Output Usage

If the HW Trigger Outputs are used, signal collisions as described in the chapter *'HW Outputs & CAN Bus Connection'* must not appear !

In the following example the message mapping from the dbcj-file in the chapter *'Example : CAN Data Base Trace'* is used.

Message IDs & Message Contents	DL	Type	SB	BL	Endian	Value	Unit	Bus Value
0x120 MonitoringStation_11	8	std						00 00 00 00 00 00 00 00
InsideTemperature			0	10	LE	0	degree	0
OutsideTemperature			16	10	LE	0	degree	0
AirHumidity			32	8	LE	0	%	0
AtmosphericPressure			48	8	LE	0	hPa	0
0x121 MonitoringStation_12	8	std						00 00 00 00 00 00 00 00
InsideTemperature			0	10	LE	0	degree	0
OutsideTemperature			16	10	LE	0	degree	0
AirHumidity			32	8	LE	0	%	0
AtmosphericPressure			48	8	LE	0	hPa	0
0x122 MonitoringStation_13	8	std						00 00 00 00 00 00 00 00
0x123 MonitoringStation_14	8	std						00 00 00 00 00 00 00 00
0x124 MonitoringStation_15	8	std						00 00 00 00 00 00 00 00
0x130 MonitoringStation_21	8	std						00 00 00 00 00 00 00 00
0x131 MonitoringStation_22	8	std						00 00 00 00 00 00 00 00

Each monitoring station message consists of 4 signals

Signal	Startbyte	Startbit	Bitlength	Format
Inside Temperature	0	0	10	LE
Outside Temperature	2	0	10	LE
Air Humidity	4	0	8	LE
Atmospheric Pressure	6	0	8	LE

Correlation between the bus value and the 'real world' value :

Signal	Bus Value [digit]	a	b	Calc Value	unit
Inside Temperature	x	0.16129	-40	0.16129 * x - 40	degree
Outside Temperature	x	0.16129	-40	0.16129 * x - 40	degree
Air Humidity	x	1	0	x	%
Atmospheric Pressure	x	1	875	x + 875	hPa

Trigger Request

In the following example the trigger TRIG1 shall be a **logical AND combination** of all other triggers V-Trig4, V-Trig3, Trig5, Trig2 and Trig1.

- Virtual Trigger V-Trig4 is used to create a combined trigger condition for V-Trig3

- Virtual Trigger V-Trig3 is a logical combination of the trigger conditions V-Trig4 and V-Trig3 and is used to create a combined trigger condition for Trig2

- Trig2 is a logical combination of the trigger conditions V-Trig4, V-Trig3 and Trig2

  Trig2 can be monitored on HW Output 2 and is furthermore used to create a combined trigger condition for Trig1

- Trig5 can be monitored on HW Output 3 and is furthermore used to create a combined trigger condition for Trig1

- Trig1 is a logical combination of the trigger conditions V-Trig4, V-Trig3, Trig2, Trig5 & Trig1.

  Trig1 can be monitored on HW Output 1.

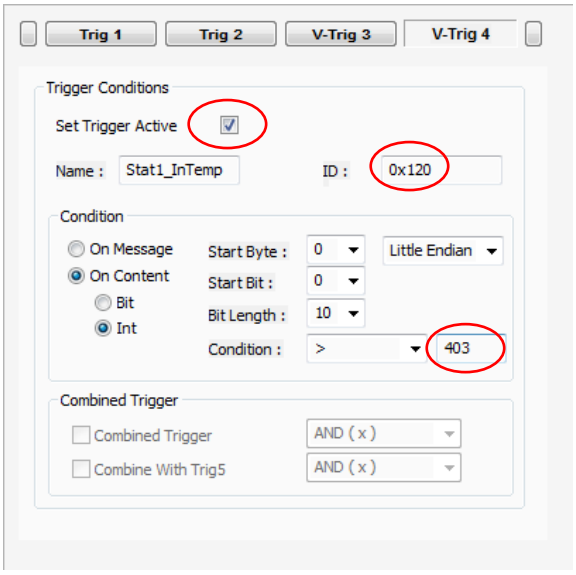
The following Trigger Table shall be realized :

Trigger	Message	Condition1	Condition2
V-Trig4	0x120	On Content	InsideTemp > 25°C
V-Trig3	0x120	On Content	OutsideTemp > 35°C
Trig2	0x120	On Content	Humidity > 80%
Trig5	0x121	On Content	OutsideTemp < 32°
Trig1	0x121	On Content	Pressure < 1000 hPa

15.1 Step 1 : Define the different Triggers in the Trigger Window

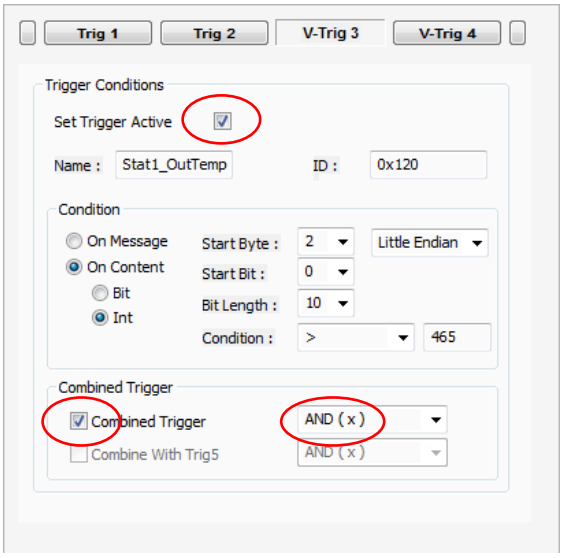
V-Trig4 condition : Inside Temperature at Monitoring Station 11 > 25°C

25°C → (25 + 40) / 0.16129 = 403 digit



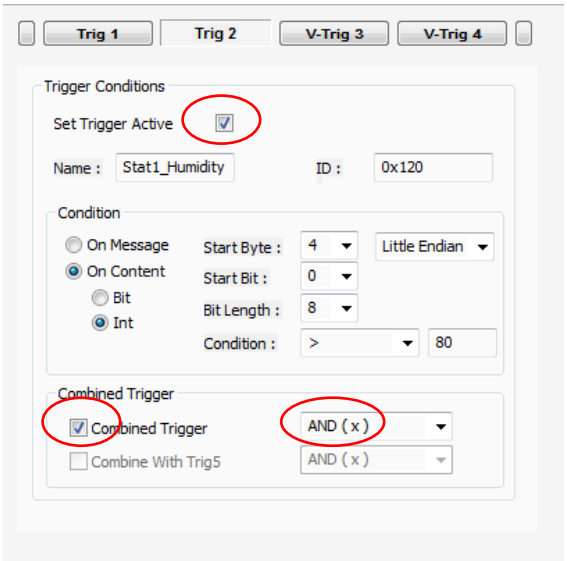
V-Trig3 condition : Outside Temperature at Monitoring Station 11 > 35°C

35°C → 465 digit



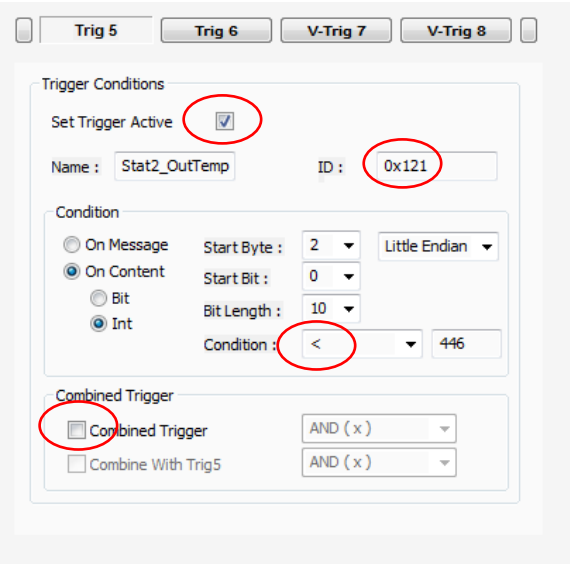
Trig2 condition : Air Humidity at Monitoring Station 11 > 80%

80% → 80 digit



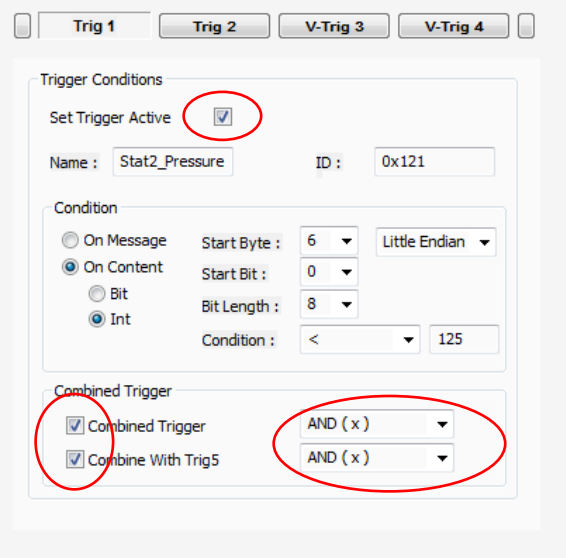
Trig5 condition : Outside Temperature at Monitoring Station 12 > 32°C

32°C → 446 digit



Trig1 condition : Atm. Pressure at Monitoring Station 12 < 1000 hPa

1000 hPa → 125 digit



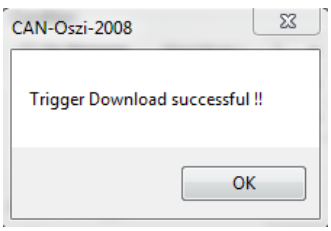
- set the triggers **'Active'**
- **Save the Trigger Configuration → Trigger → Save Configuration**

15.2 Step 2 : Download the trigger configuration to the CLDT Box

- Trigger download is done via the Menu Item → Trigger → Download

  or via the Toolbar shortcut 

- The 'Trigger Download successful' Message Box should be displayed. If an error or error code is displayed please save the configuration again and re-start the trigger download.



15.3 Step 3 : Activate the Trigger Mode and start the CAN Trace

To start a CAN Trace or a CAN Restbus Simulation in Trigger Mode the 'Trigger Mode Active' check box must be set **before** starting the trace or the simulation.

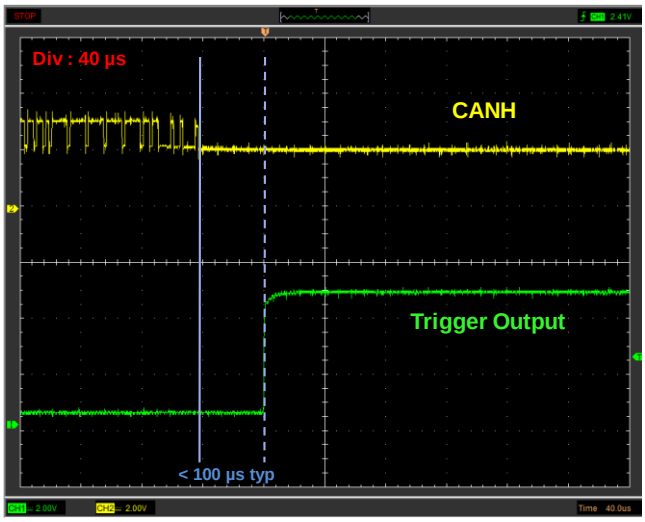
Otherwise the HW Outputs will be considered as HW Signal Outputs (see chapter *'HW Output Usage'*)

15.4 Step 4 : Monitor the Trigger Outputs with a monitoring tool

Several monitoring or signal generating tools can be used to supervise the trigger output signals. A block diagram of the physical HW Outputs is shown in the chapter *'Output Signal Monitoring'* .

Trigger Condition	triggered HW Output
Trig1 = true	Output 1 : logic high level or positive pulse
Trig2 = true	Output 2 : logic high level or positive pulse
Trig5 = true	Output 3 : logic high level or positive pulse
Trig6 = true	Output 4 : logic high level or positive pulse

Depending on *the particular trigger conditions* a logic high level or a pulse is generated at the corresponding trigger outputs.



oscilloscope as monitoring tool for trigger Output 1

16 Example : RX / TX Automation with HW Output Synchronization

All CLDT project files and all dll projects (including source codes) for the following examples are available in the CLDT Box installation folder under

…\examples\RxTxAutomation\...

Use of C Compilers

If any C-Compiler other than the LCC-Win32 Compiler or the MSVC++ Compiler shall be used, please refer to the source code of the delivered h-files and c-files (see example_16_x.zip) in order to implement correctly all dll export functions.

Note that the correct pointer setting for the shared Rx & Tx variables is done by the CLDT SW when creating or updating a dll project.

- ➔ create a dummy project for the MSVC++ or LCC Compiler
- ➔ derive your own h-file and c-file by using the pointer settings defined in the h-file of the dummy project

16.1 Use LCC-Win32 Compiler for dll restbus simulation

The CLDT project file for this example, as well as an example for a user specific restbus application are available in the CLDT Box installation folder under

…\examples\RxTxAutomation\example_16_1.zip

16.1.1 Create a new dll project, compile it and load it with the CLDT SW

1) Ensure that the LCC-Win32 Compiler is correctly installed and that the *LCC Compiler Settings for the CLDT SW* are correct

2) Close the LCC Compiler (if it was open)

3) Start the CLDT SW and open the project *example_16_1.pjt*

4) Open the dll create panel (→dll→create dll project)

5) Check the LCC Compiler Settings (see 1))

6) Click ‘Create dll’ and choose a project name

7) Click ‘OK’ on the success MessageBox

8) The LCC Compiler should be opened automatically and the created dll project should appear. If the LCC Compiler is not opened automatically or if the new dll project is not displayed, the project must be chosen manually (see below).

9) Compile the dll (LCC menu item →Compiler→Rebuild All)

10) Confirm with ‘OK’ or ‘yes’

11) The dll will be created in the dll projects subfolder \lcc\...

12) Close the dll create panel in the CLDT SW

13) Open the dll load panel (→dll→load dll)

14) Choose ‘load new’ and select the created dll in the dll projects subfolder \lcc\...

15) The green ‘dll ok LED’ should appear

The new dll is now loaded and the restbus simulation could be started.
However the user logic should be programmed by the user before …

Choose the dll project manually in the LCC Compiler

1) Start the LCC Compiler
2) Close the current project, if one is open (→Project→close)
3) Import the desired project (→Project→import)
4) Set the files drop down list on the bottom right corner to ‘Project Files’
5) Select the desired dll project .prj

16.1.2 Program the user logic for the restbus simulation

The three central functions **OnSimulationStart()**

RxTxLogic(i64DeltaT)

OnSimulationStop()

may now be programmed by the user.

The integer function parameter *i64DeltaT* contains the elapsed time in [ns] since the last dll call. Therefore exact timings (as far as possible under Windows operating systems) can be created.

Have a closer look at the h-file and c-file which are delivered with this example. They show, how the RX & TX simulation values are used inside the dll, how the 4 physical HW Outputs can be controlled and how a simple sub-operating system can be implemented with the dll.

16.1.3 Testing … \example_16_1\ ...

1) Create the folder C:\temp\debug\ to store debug informations

2) Start the CLDT SW and load the project

   …\example_16_1\CLDT_pjt\example_16_1.pjt

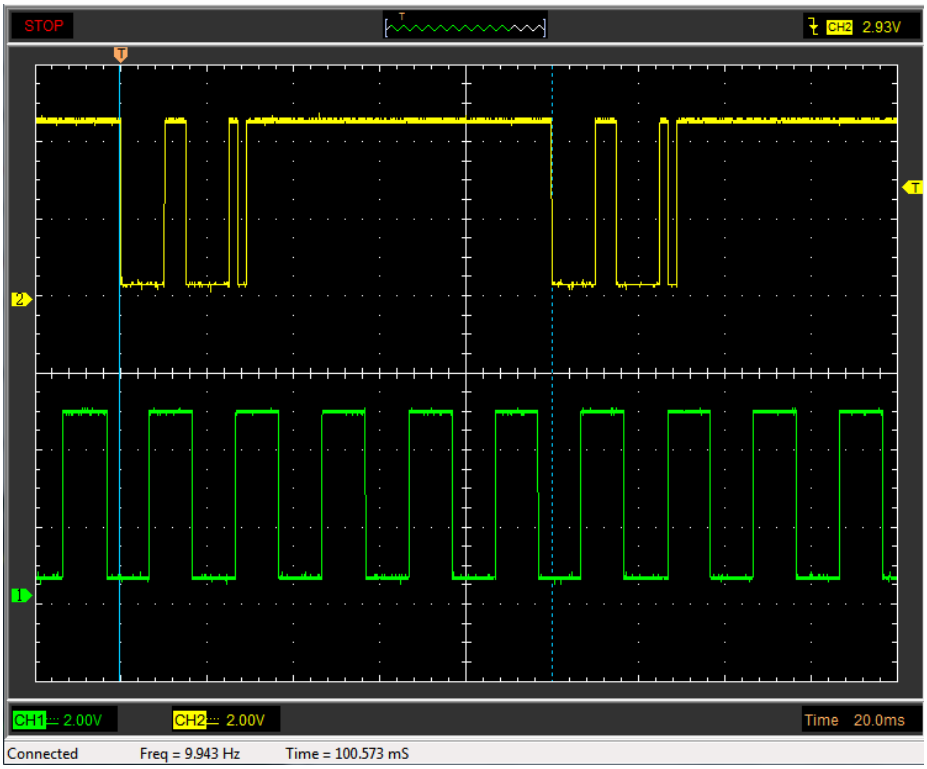
3) Open the dll load panel and load the dll

   …\example_16_1\LCC_pjt\example_16_1\Lcc\example_16_1.dll

4) Connect the CLDT Box and a 2nd ECU (or 2nd CLDT Box) with 500 kBit/s to the CAN Bus

5) Set the check boxes ‘Update Sim Windows’ and ‘Update List Fields’ on the dll load panel

6) Start the Restbus Simulation and use the dll control panel to start / stop CAN Bus communication



HW Output Curve Generation : Output 2 & Output 6 for example_16_1

16.2 Use Microchip VC++ Compiler for dll restbus simulation

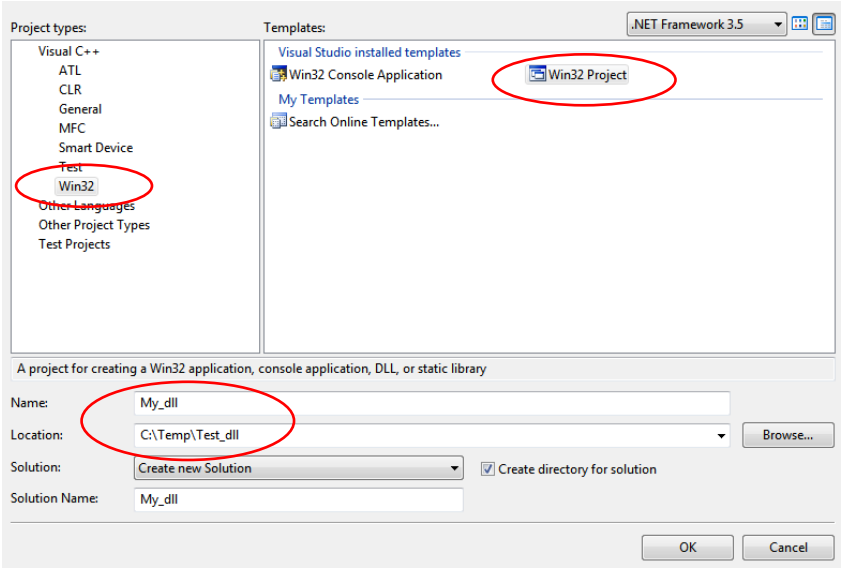
The CLDT project file for this example, as well as an example for a user specific restbus application are available in the CLDT Box installation folder under

…\examples\RxTxAutomation\example_16_2.zip

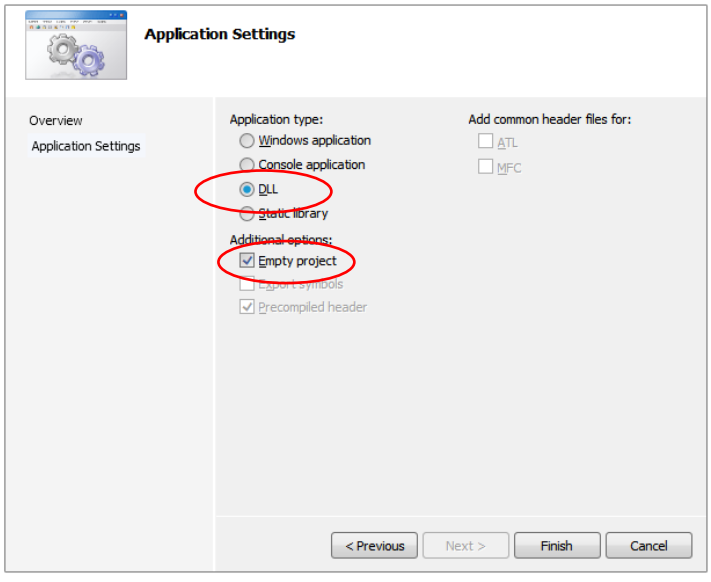
16.2.1 Create an empty dll project with VC++ Compiler

1) Open Microchip VC++ Compiler, choose →File→New→Project

2) Select ‘Win32 Project’ choose the project folder & project name & click ‘OK’



3) Click ‘Next’, choose DLL and ‘empty project’ & click ‘Finish’

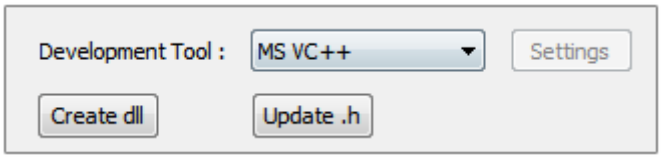


16.2.2 Create h-file & cpp-file with CLDT SW

4) Start the CLDT SW and open the project *example_16_2.pjt*

5) Open the dll create panel (→dll→create dll project)

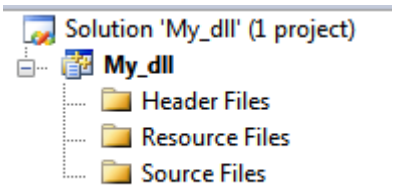
6) Choose the MSVC++ Compiler, click 'Create dll'



7) Choose the VC++ project folder (see 2)) and choose a name for the dll's h-file and cpp-file

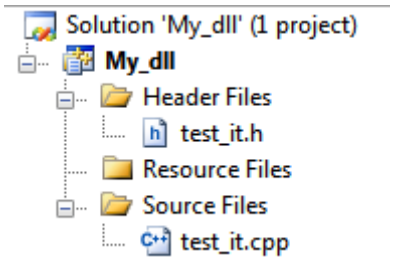
The cpp-file & h-file are created in the VC++ project folder

16.2.3 Add h-file and cpp-file to VC++ dll Project



8) In the VC++ Solution Explorer right click *Header Files* → *Add* → *Existing Item* and choose the h-file

9) In the solution Explorer right click *Source Files* → *Add* → *Existing Item* and choose the cpp-file



16.2.4 Compile and load dll with CLDT SW

10) Set VC++ build option to 'Release Built' and choose *→Build→Build Solution*

11) The dll is compiled successfully and stored in the project folder …\Release\

12) Close the dll create panel in the CLDT SW

13) Open the dll load panel (→dll→load dll)

14) Choose 'load new' and select the created dll in the project folder …\Release\

15) The green 'dll ok LED' should appear

The new dll is now loaded and the restbus simulation could be started.
However the user logic should be programmed by the user before …

16.2.5 Program the user logic for the restbus simulation

The three central functions **OnSimulationStart()**

RxTxLogic(i64DeltaT)

OnSimulationStop()

may now be programmed by the user.

The integer function parameter *i64DeltaT* contains the elapsed time in [ns] since the last dll call. Therefore exact timings (as far as possible under Windows operating systems) can be created.

Have a closer look at the h-file and cpp-file which are delivered with this example. They show, how the RX & TX simulation values are used inside the dll, how the 4 physical HW Outputs can be controlled and how a simple sub-operating system can be implemented with the dll.

16.2.6 Testing … \example_16_2\ ...

1) Create the folder C:\temp\debug\ to store debug informations

2) Start the CLDT SW and load the project

   …\example_16_2\CLDT_pjt\example_16_2.pjt

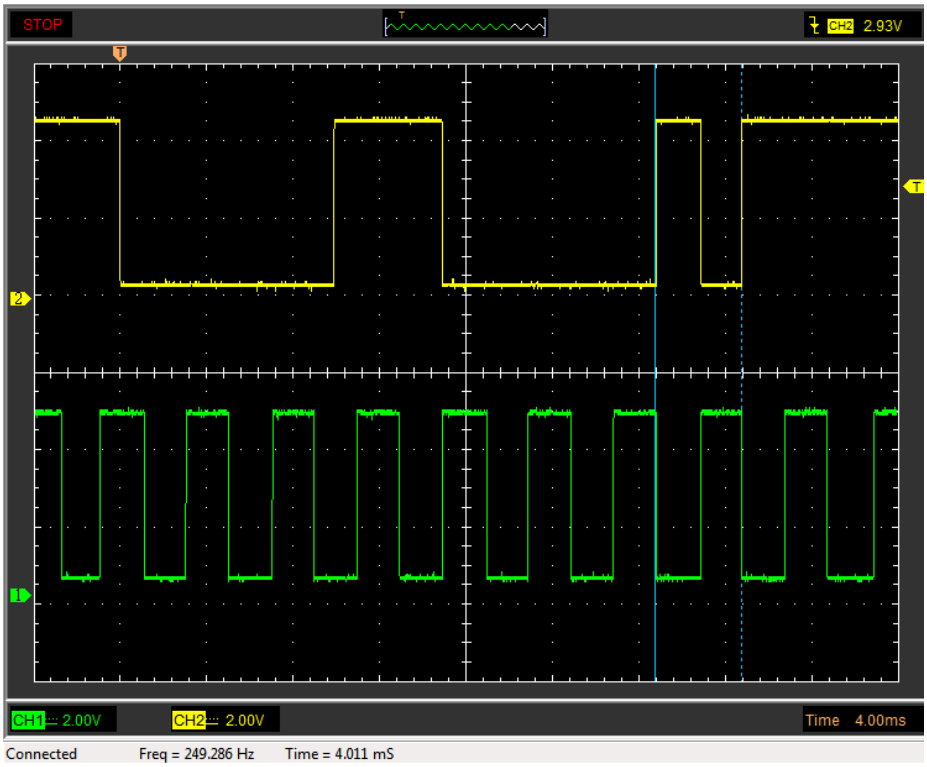
3) Open the dll load panel and load the dll

   …\example_16_2\MSVCdll_pjt\example_16_2\Release\example_16_2.dll

4) Connect the CLDT Box and a 2nd ECU (or 2nd CLDT Box) with 500 kBit/s to the CAN Bus

5) Set the check boxes ‘Update Sim Windows’ and ‘Update List Fields’ on the dll load panel

6) Start the Restbus Simulation and use the dll control panel to start / stop CAN Bus communication



HW Output Curve Generation : Output 1 & Output 6 for example_16_2

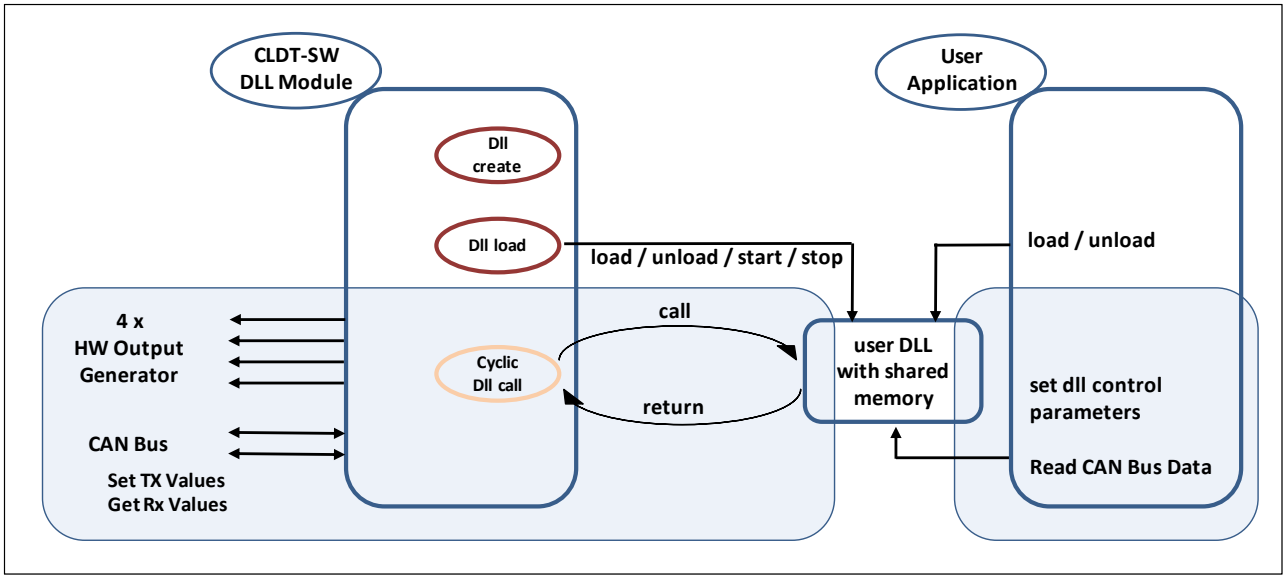
16.3 Use Microchip VC++ Compiler to create an external dll Control Panel

The CLDT project file for this example, as well as the VC++ dll project and the VC++ User Application are available in the CLDT Box installation folder under

…\examples\RxTxAutomation\example_16_3.zip

16.3.1 Simple way of data exchange with external application

In this example the dll control shall not be realized by the dll control panel (which is integrated in the CLDT SW), but by an external application. Additionally the external application should monitor some RX or TX data from the CLDT application.



- the dll must be created in the same way as described in the last chapter
- the export functions for the communication with the user application must be added to the dll
- data space that is shared between all applications must be declared in the dll
- the user application must be realized

Have a closer look at the dll project and the user application project. Both projects were generated with the VC++ Compiler. These two projects demonstrate how an external application can control the dll program flow and how it can access data from the CLDT Application.

16.3.2 Testing … \example_16_3\ ...

1) Create the folder C:\temp\debug\ to store debug informations

2) Start the CLDT SW and load the project

   …\example_16_3\CLDT_pjt\example_16_3.pjt

3) Open the dll load panel and load the dll

   …\example_16_3\MSVCdll_pjt\example_16_3\Release\example_16_3.dll

4) Connect the CLDT Box and a 2nd ECU (or 2nd CLDT Box) with 500 kBit/s to the CAN Bus

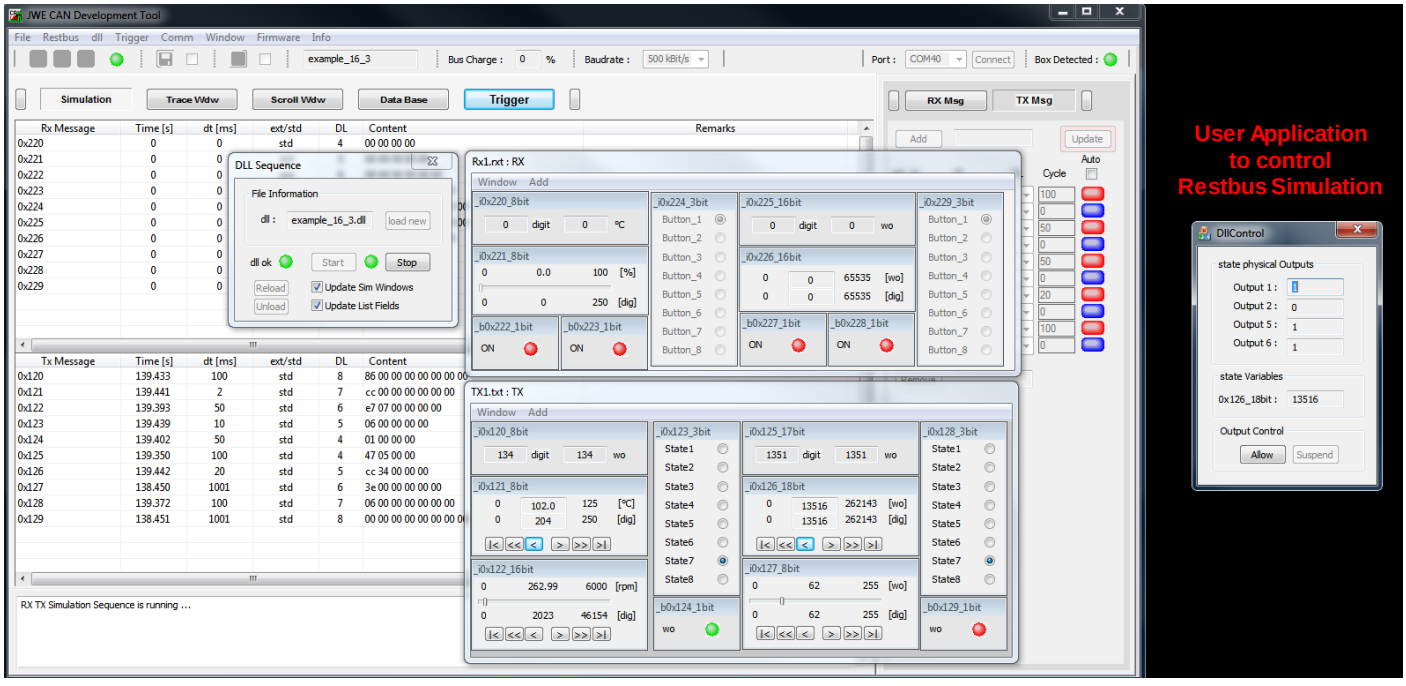
5) Set the check boxes ‘Update Sim Windows’ and ‘Update List Fields’ on the dll load panel

6) Start the user application

   …\example_16_3\DllControl_userApp\Release\DllControl_userApp.exe

7) Load the same dll as under 3)

8) Start the Restbus Simulation and use the user application to start / stop CAN Bus communication



17 Example : TX Sequence Generation with Excel Makro

The Excel file ‘TxSequenceMacros_V01.xls’ with the 2 Makros ‘CreateTxSequence1’ & ‘CreateTxSequence2’, as well as the generated .seq-files are available in the CLDT Box installation folder under

…\examples\TxSequence\example_17.zip

17.1 TX Sequence Header ID and TX Sequence Data

The Header ID data (see chapter *‘.seq-file format description’*) for the 2 TX sequences is defined in the Excel sheets ‘TX_IDs1’ & ‘TX_IDs2’.

The Sequence Data for the 2 TX sequences is defined in the Excel sheets ‘TX_Sequence1’ & ‘TX_Sequence2’.

A click on the ‘Create TxSequence X’ command button creates the corresponding .seq-file in the folder c:\\temp\

The Macros should demonstrate how Excel can be used to easily create a correctly formatted .seq-file.

18 CLDT Box : Software Description LIN Bus Support

Planned LIN Support : **Q2 / 2015**

- Master Mode
- Slave Mode
- Trace Mode

Chapter under Revision

19 Future Software Scope

Chapter under Revision

19.1 CAN – LIN Gateway Functionality

Chapter under Revision

20 Technical Specification

20.1 ABSOLUTE MAXIMUM RATINGS

SubD 9 female Connector		min	max
Temperature range		- 20°C	+ 85°C
Voltage range	CANH	- 27V	+ 40V
	CANL	- 27V	+ 40V
	LIN	- 18V	+ 40V
	Single Wire CAN	- 20V	+ 40V
	HW Output 1 … 4 (external Voltage !)	**0V**	**+ 5V**
Output Current	CAN Receiver	-	20 mA
	LIN : Ibus Short Circuit Current Limit	-	200 mA
	Single Wire CAN : Ibus Short Circuit Current Limit	-	350 mA
	HW Output 1 … 4	-	5 mA

20.2 NORMAL OPERATION RATINGS

SubD 9 female Connector & USB Connector		min	typ	max
Supply Voltage	by PC USB Port		+5V	
Supply current				200 mA
Output 1 … 4	Input Capacitor		10 nF	
	Output High Level		5 V	
	Output High Resistance		1 kΩ	
	High Level Output Current			5 mA
	Output Low Level	0.3V	0.6V	1 V
	Input Low Level Current			4 mA
GND line fuse				200 mA

21 Change History

Version	Date	Changes
V1.0	28.09.2014	First Preliminary Release
V1.1	15.12.2014	Chapter 8.9 : New RX & TX Controls added Chapter 10.2 : Meaning of green & red update button Chapter 10.7 : Meaning of LED colors New function : Suspend CAN message : Red LED
V1.2	10.03.2015	Chapter 3 : change from BETA state to Pre-Release state Chapter 5.4.1 : correct setting of latency time-out added Chapter 9 : Scope Functionality added Chapter 10 : RX / TX Automation added Chapter 11 : TX Sequence added Chapter 16 : Example RX/TX Automation added Chapter 17 : Example TX Sequence added

22 Environmental Aspects

The following symbol indicates that this product complies with the European Union requirements according to Directives 2002/96/EC on Waste Electrical and Electronic Equipment (WEEE).

22.1 Product End Of Life

The equipment may contain materials or substances that could cause damage to the environment or human health. In order to avoid release of such substances into the environment this product should be recycled in an appropriate system to ensure that parts of the materials are reused or recycled in an appropriate way.

23 EC Declaration Of Conformity

CE

EG - Konformitätserklärung / EC – Declaration of Conformity

Firma / Company	JW Electronics Hauptstrasse 43 D – 78601 Mahlstetten
Hiermit erklären wir, dass die Produkte We herewith declare, that the products	CLDT1004, CLDT1014, CLDT0114, CLDT0014 CLDT0104
Bezeichnung der Produkte : Product Description :	CAN & LIN Development Tool

mit den folgenden EG-Rats-Richtlinien übereinstimmen :
are in conformity with the following European Directives :

2004/108/EC (EMC)

Angewandte harmonisierte Normen : Applicable Harmonized Standards :	**EN 61326-1:2006** **EN 61000-6-1:2007** **EN 61000-6-3:2007 +A1:2011**

Mahlstetten, 26.Juli 2014

Jürgen Walter
Geschäftsinhaber