Agilent N5479A FlexRay Protocol Analysis Software for the VPT1000 FlexRay Module

User's Guide

Agilent Technologies

Notices

Trademarks

Microsoft®, MS-DOS®, Windows®, Windows 2000®, and Windows XP® are U.S. registered trademarks of Microsoft Corporation.

Adobe®, Acrobat®, and the Acrobat Logo® are trademarks of Adobe Systems Incorporated.

Manual Part Number

N5479-97000

Edition

June 2009

Available in electronic format only

Agilent Technologies, Inc.
1900 Garden of the Gods Road
Colorado Springs, CO 80907 USA

Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Agilent Technologies' standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Safety Notices

CAUTION

A **CAUTION** notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a **CAUTION** notice until the indicated conditions are fully understood and met.

WARNING

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

The first comprehensive and integrated FlexRay protocol analyzer

Contents

1
Introduction

The VPT1000 PC-based software (N5479A) is a Microsoft Windows-based visualization and control tool for the VPT1000 FlexRay acquisition module.

The VPT1000 acquisition hardware module ships with Option FR2 (or N5432B) on an MSO6000 or MSO7000 series oscilloscope.

For stand-alone FlexRay measurements on the oscilloscope, the scope directly controls, receives, and processes data from the VPT1000 acquisition module. In this oscilloscope-only use-model, the VPT1000 PC software is not required.

With the addition of the VPT1000 PC Software (N5479A), you can also control and analyze data captured by the VPT1000 on a Windows-based PC; providing a higher abstraction level of FlexRay protocol analysis than is available on the scope alone.

Note that there are three different use-models of FlexRay protocol analysis measurements involving the VPT1000 acquisition module:

1. Scope + VPT1000 = FlexRay physical layer with basic protocol analysis measurements only
2. VPT1000 + PC + VPT1000 Software (N5479A) = Higher level FlexRay protocol layer measurements only
3. Scope + VPT1000 + PC + VPT1000 Software (N5479A) = Both

The Agilent VPT1000 PC software (N5479A) provides monitoring features for FlexRay and CAN as well as analog and digital signals allowing:

- Online monitoring with or without *Host Logging*, and
- Offline analysis.

During online monitoring the VPT1000 Hardware transmits monitored data to the VPT1000 PC Software. The VPT1000 PC Software visualizes the monitored data and, if Host Logging is enabled, also records the data on the PC's hard disk.

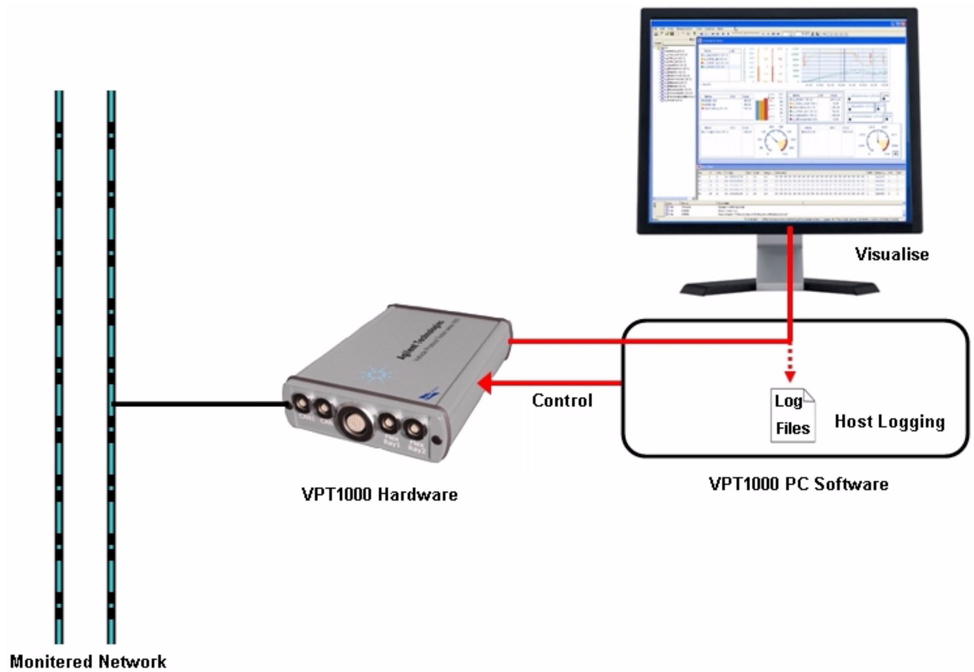


Figure 1 Online Monitoring

Monitored data recorded during *Host Logging* may be analyzed offline with the VPT1000 PC Software.

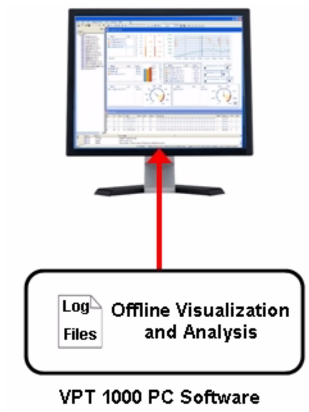


Figure 2 Offline Analysis

In addition to data visualization, the VPT1000 PC Software provides the means to control and to configure the VPT1000 Hardware. Filters can be configured via VPT1000 PC Software in order to record relevant data only.

With the addition of the N5479A-001 Advanced Software option, the VPT1000 Hardware also allows FlexRay frame transmission and fault injection into FlexRay clusters via COM interface.

Hardware/Software Features

- Online monitoring on PC via Ethernet: Continuous data transfer to PC via Ethernet or USB up to 2 x 10 Mbps FlexRay plus 2 x 1 Mbps CAN.
- Interfaces to FlexRay, CAN and analog/digital input: Monitoring and time-stamping of FlexRay, CAN and analog/digital input allow for analysis of temporal correlations.
- *Triggerout Pins*: Four configurable pins providing FlexRay-related triggers.
- Online/offline data analysis for FlexRay, CAN and analog/digital input.
- Import FIBEX, DBC and BOR files holding FlexRay or CAN protocol parameter configuration and signal information.
- Manual configuration of FlexRay protocol parameters.
- Manual FlexRay and CAN signal configuration.
- Multiple import of various signal databases.
- *Host Logging:* Monitored data is recorded to a PC's hard disk during online monitoring.
- Offline analysis: Analyzing data recorded during *Host Logging*.
- Integrated COM interface (optional, FlexRay Advanced Software option N5479A-001): Provides access to enhanced functionalities and allows development of extensions:
  - Fault injection module (see “Fault Injection" on page 106).
  - Startup buddy: VPT1000 Hardware can serve as leading coldstarter and as following coldstarter simultaneously and can, therefore, startup a FlexRay cluster autonomously.
  - Script-based transmit and receive functionality for FlexRay for VPT1000 Hardware.
- Agilent Technologies MSO6000/7000 Series Oscilloscope with option FR2 (N5432B): The VPT1000 Hardware decodes the FlexRay protocol information and hands it over to the scope where this information is displayed correlated with a FlexRay frame.

2
Measurement Hardware

Key Components

The key components of the VPT1000 Hardware are:

- Special FlexRay Measurement Controller interoperable with FlexRay controllers compliant to FlexRay Protocol Specification 2.1
- FlexRay fault injection as standard feature (see "Fault Injection" on page 106)
- 2 FlexRay Channel via TJA1080
- 2 FlexRay Channel via RS485
- 2 CAN Communication Controllers based on BOSCH CAN_CORE IP via TJA1040
- 4 analog inputs connected to 4 ADC
- 4 digital inputs
- 4 configurable *Triggerout Pins*
- Serial console
- 100 Mbps Ethernet
- USB 2.0 interface
- +6 to +30V power supply for VPT1000 Hardware

Optional Extensions

VPT1000 PC software can be optionally extended with N5479A-001 Advanced Software option, which adds two additional FlexRay communication controllers enabling the VPT1000 Hardware to act as a startup node and to insert frames. The VPT1000 can be programmed using COM interface

- to emulate two FlexRay coldstart nodes and to perform an autonomous FlexRay startup and
- to provide additional features for transmission and reception of FlexRay communication elements.

Note: Do not remove the cover from the VPT1000 Hardware. Doing so will result in loss of warranty coverage.

Installation

Physical Installation

To configure the VPT1000 Hardware and to perform monitoring, an Ethernet or an USB connection from the VPT1000 Hardware to a standard PC is required. It is recommended to use a dedicated Ethernet connection, as monitoring and *Host Logging* might require the full bandwidth of Ethernet. Furthermore it is possible to connect the VPT1000 Hardware to the PC via serial link (RS232). This connection is only used for configuration of Ethernet settings (see "Ethernet Configuration" on page 14).

For connections to FlexRay, CAN and analog/digital lines, see "VPT1000 Hardware Housing" on page 21.

Ethernet Configuration

To control the VPT1000 Hardware using VPT1000 PC Software via Ethernet the network setting on both the VPT1000 Hardware and the PC must be configured to be in the same subnet. The VPT1000 Hardware is shipped with a pre-configured IP address, which is:

192.168.80.80

The subnet mask is configured to 255.255.255.0. This means the IP address of the PC must be identical to the IP address of the VPT1000 Hardware at the first 24 Bits and must be different in the last 8 Bits.

For configuring the IP address of the PC see "Network setting of the measurement PC" on page 14. For configuring the IP address of the VPT1000 Hardware see "Network setting of VPT1000 Hardware" on page 14.

Network setting of the measurement PC

The network settings of the PC may be configured via dhcp server, if the VPT1000 Hardware is used in an existing network or may be configured statically if the VPT1000 Hardware is connected directly to the PC (which is recommended due to high network load, see "Physical Installation" on page 14).

For using the VPT1000 Hardware in an existing network no changes to the PC configuration are necessary.

Network setting of VPT1000 Hardware

To change the Network setting of the VPT1000 Hardware, start the *VPT1000 Firmware Configuration Tool*. Access it via Start->Programs->Agilent VPT1000->VPT1000 Firmware Configuration Tool.

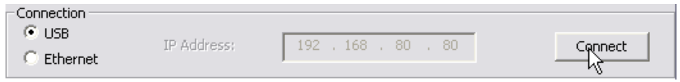


Figure 3 Connect to Hardware

Connect to the VPT1000 Hardware unit either via USB or via Ethernet (use the default IP address 192.168.80.80, if no other setting was applied previously).

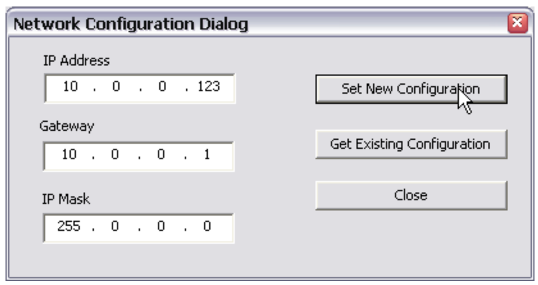


Figure 4 Network Configuration

If the connect is successful, you can access the network settings via the "Network Settings" button in the "Advanced Hardware controls" group (see Figure 4). To change the address, type the proper values in the dialog, and press "Set New Configuration". Upon success, you will see a confirmation dialog. Note that these changes do not take effect until after a reset of the VPT1000 hardware unit.

Afterwards the VPT1000 Hardware can be restarted with the new network settings interrupting the power supply. After reboot the Ethernet connection with the new settings is available. The IP address, configurable in VPT1000 PC Software (see "VPT1000 PC Software Basic Configuration" on page 42), must match with the IP address configured in the VPT1000 Hardware.

USB installation and configuration

The USB drivers for the VPT1000 Hardware are automatically installed with the VPT1000 PC Software (see "Installing VPT1000 PC Software" on page 32). Hence it is recommended to install the VPT1000 PC Software before you connect the VPT1000 Hardware the first time with the PC via USB.

USB Hardware setup

The USB cable, which is delivered with the VPT1000 Hardware has two USB plugs (USB and Power), which can be connected to a PC. At the opposite end of the cable there is an ODU plug, which is plugged into the USB connector of the VPT1000 Hardware.

It is only possible to operate one VPT1000 Hardware at a time, therefore it is recommended not to connect more than one VPT1000 Hardware to the PC via USB.

There are two ways to supply the VPT1000 Hardware with power:

- You can supply the VPT1000 Hardware with an external power supply via the power connector of the VPT1000 Hardware. In this case you must connect the USB plug with the PC (you don't need to connect the Power plug).
- You can supply the VPT1000 Hardware via USB. In this case it is recommended to connect the VPT1000 Hardware with the PC via USB and Power plug. The USB standard only guarantees 500mA per connector and the VPT1000 Hardware requires more than that for correct operation.

  Important: Connect both USB plugs to the PC first, before you connect the ODU plug to the VPT1000 Hardware. Thereby you make sure, that over 500mA are available from the very beginning.

Driver installation

Install the VPT1000 PC Software before you connect the VPT1000 Hardware as described in "USB Hardware setup" on page 15. Windows detects the new hardware, and prompts for driver installation as shown in Figure 5.

Figure 5 Found New Hardware dialog

Choose "Not this time" and continue with "Next".

The Dialog show in Figure 6 comes up.

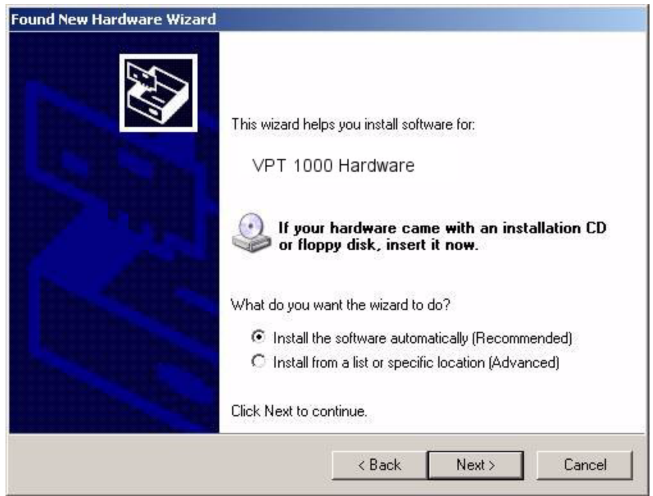


Figure 6 Install USB Driver

As the install shield for the VPT1000 PC Software has already copied the driver files to the Windows driver directories, you can choose "Install the software automatically", and Windows will thereby find and install the driver files automatically as you click "Next".

Figure 7 Completion

The dialog shown in Figure 7 confirms the completion of the driver installation. Close the Wizard with "Finish", the USB driver for the VPT1000 Hardware is now available and ready to use.

VPT1000 Hardware Firmware And Hardware License

VPT1000 Hardware is delivered with pre-installed firmware and pre-installed hardware license. Nevertheless, the firmware and hardware license update process for the VPT1000 Hardware is described below.

The Agilent Technologies *Firmware Update Tool* as shown in Figure 8 provides the following services:

- Firmware update
- Hardware license upgrade
- Advanced hardware services

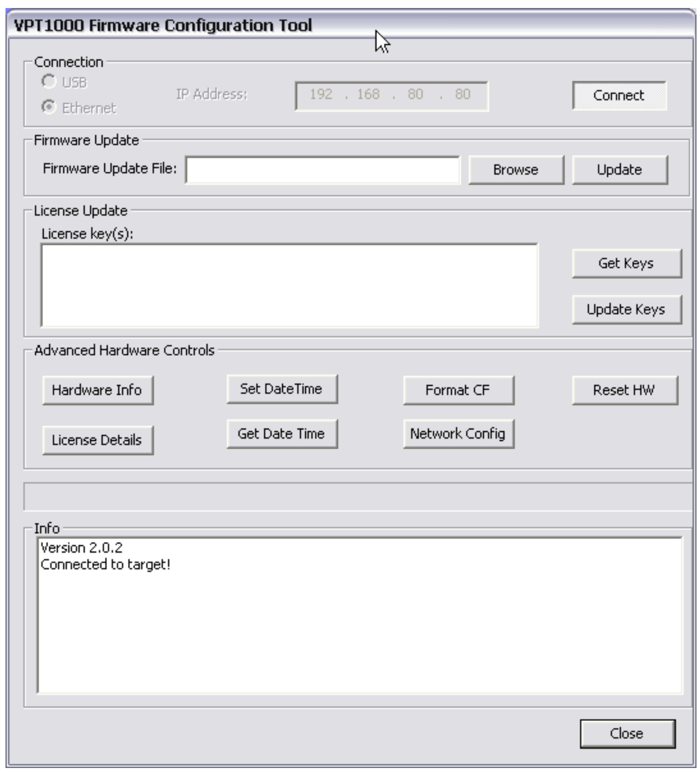


Figure 8 Firmware upgrade tool

Firmware update

The VPT1000 Hardware firmware images can be ordered at Agilent Technologies or downloaded from http://www.agilent.com/find/vpt1000. They are also included on the Software installation CD. These firmware images exhibit the file extension .fwu.

To update the firmware, follow the described procedure and consider additional hints in the README file delivered with the firmware image:

1 Start *Firmware Update Tool*
2 Select USB or enter IP address of the VPT1000 Hardware and press *Connect*
3 Select the firmware update file (firmware image) using the *Browse Button* in the *Firmware Update* area
4 Start firmware update by pressing the *Update Button* in the *Firmware Update* area
5 Wait until firmware update is completed and firmware and hardware information is printed in the text box *Info*

Attention: Do not disconnect the VPT1000 Hardware during the firmware update process!

Hardware license upgrade

The basic hardware license is already installed on the VPT1000 Hardware at delivery. To upgrade the supported feature set a new hardware license key may be installed on the VPT1000 Hardware.

If you have purchased N5479A-001 Advanced Software option with your N5479A FlexRay PC Software, you will have received a License Certificate with your shipment.

To install the license on your hardware, use the VPT1000 Firmware Configuration Tool. Access it via Start->Programs->Agilent VPT1000->VPT1000 Firmware Configuration Tool. Connect to your hardware using USB or LAN (note that the pre-configured IP address for the hardware is 192.168.80.80), and enter the key provided on the certificate To install, press "Update Keys". You will get a confirmation message in the info window that the update succeeded. You have to disconnect your VPT1000 unit and cycle power for the upgrade to become effective.

Follow the described procedure to install the license on your hardware:

1 Start the *VPT1000 Firmware Configuration Tool*. Access it via Start->Programs->Agilent VPT1000->VPT1000 Firmware Configuration Tool.
2 Choose USB or enter IP address of the VPT1000 Hardware and press *Connect (*note that the pre-configured IP address for the hardware is 192.168.80.80)
3 Enter the key provided on the certificate into the License Update area
4 Start license update by pressing the *Update Keys* in the *License Update* area
5 You will get a confirmation message in the info window that the update succeeded.

6 disconnect your VPT1000 Hardware unit and cycle power for the upgrade to become effective

Attention: A hardware license key only works for a specific hardware with dedicated serial number. Be sure to install the correct license on your VPT1000 Hardware. A mismatching license file can render your VPT1000 Hardware unusable.

Advanced hardware services

There is a set of service functionality available for VPT1000 Hardware. Note that these services are only available after connecting to the hardware by entering the IP address and pressing the *ConnectButton* in the *Firmware Update Tool*.

The following services are provided:

- Get HW Info: Print detailed hardware and firmware information to the *Info* area.
- Get License: Print installed and functional licenses to the *Info* area.
- Set DateTime: Set date and time of the VPT1000 Hardware to the actual system time of the measurement PC.
- Get DateTime: Print the actual date and time of the VPT1000 Hardware to the *Info* area.

Support

Please see page 41 for Agilent support contact information.

VPT1000 Hardware Housing

The VPT1000 Hardware housing provides the following new features:

- Robust connectors suited for rough conditions in the automotive environment
- Robust housing

The automotive housing offers several external interfaces. Figure 9 and Figure 10 give an overview of the available ODU connectors.

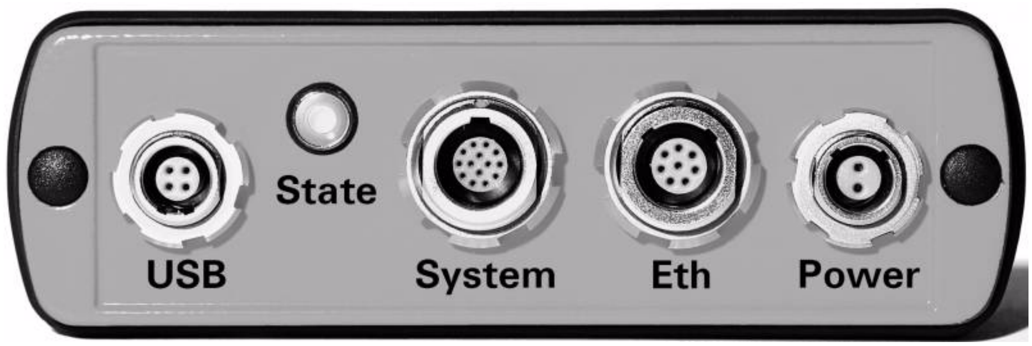


Figure 9 ODU connectors at VPT1000 Hardware housing (Front)

Figure 10 ODU connectors at VPT1000 Hardware housing (Back)

FlexRay 1 Connector (Channel A)

The FlexRay signals for Channel A from the TJA1080 transceiver and the RS485 transceiver are assigned to the 5-pin FlexRay 1 connector. The FlexRay 1 connector offers an additional ground reference pin. The pin assignment is shown in Figure 11 and Table 1.

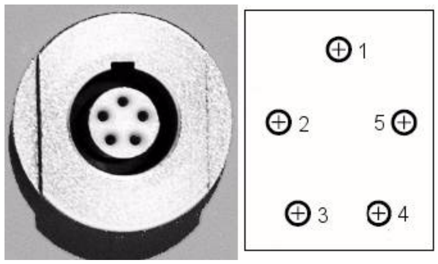


Figure 11 FlexRay 1 connector (Channel A)

Table 1 FlexRay 1 connector (Channel A)

Signal	Pin	Description
RS485_1_A	1	FlexRay Channel A (RS485, A)
RS485_1_B	2	FlexRay Channel A (RS485, B)
FL1_BP	3	FlexRay Channel A (TJA1080, BP)
FL1_BM	4	FlexRay Channel A (TJA1080, BM)
GND	5	Ground

FlexRay 2 Connector (Channel B)

The FlexRay signals for Channel B from the TJA1080 transceiver and the RS485 transceiver are assigned to the 5-pin FlexRay 2 connector. The FlexRay 2 connector offers an additional ground reference pin. The pin assignment is shown in Figure 12 and Table 2.

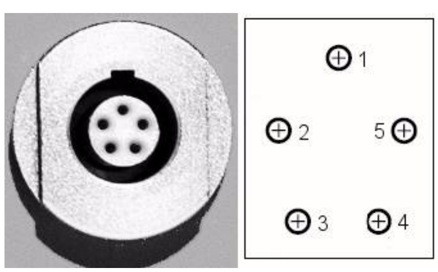


Figure 12 FlexRay 2 connector (Channel B)

Table 2 FlexRay 2 connector (Channel B)

Signal	Pin	Description
RS485_2_A	1	FlexRay Channel B (RS485, A)
RS485_2_B	2	FlexRay Channel B (RS485, B)
FL2_BP	3	FlexRay Channel B (TJA1080, BP)
FL2_BM	4	FlexRay Channel B (TJA1080, BM)
GND	5	Ground

CAN 1 Connector

One CAN channel is connectable via the 5-pin CAN 1 connector. The pin assignment is shown in Figure 13 and Table 3.

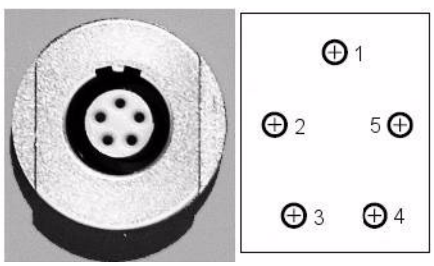


Figure 13 CAN 1 connector

Table 3 CAN 1 connector

Signal	Pin	Description
CANH1	1	CAN 1 (High)
CANL1	2	CAN 1 (Low)
GND	3	Ground
-	4	reserved
-	5	reserved

CAN 2 Connector

One CAN channel is connectable via the 5-pin CAN 1 connector. The pin assignment is shown in Figure 14 and Table 4.

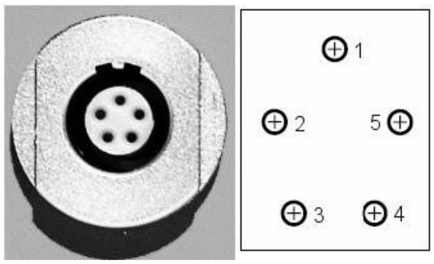


Figure 14 CAN 2 connector

Table 4 CAN 2 connector

Signal	Pin	Description
CANH2	1	CAN 2 (High)
CANL2	2	CAN 2 (Low)
GND	3	Ground
-	4	reserved
-	5	reserved

Digital/Analog Connector

The digital/analog connector comprises digital/analog input pins and *Triggerout Pins*. The digital input pins are TTL tolerant, which means 0V to 0.8V is interpreted as LOW and 2.0V to 5.5V as HIGH. The output voltage of the *Triggerout Pins* is 0V for trigger inactive and 3.3V for the trigger pulse. The analog input pins are connected to ADCs with a resolution of 12 Bit. The pin assignment is shown in Figure 15 and Table 5. The voltage ranges for analog inputs are listed in Table 5. The value range displayed in VPT1000 PC Software is 0x0 to 0x0FFF. I.e., for analog inputs 0x0FFF represents +10V.

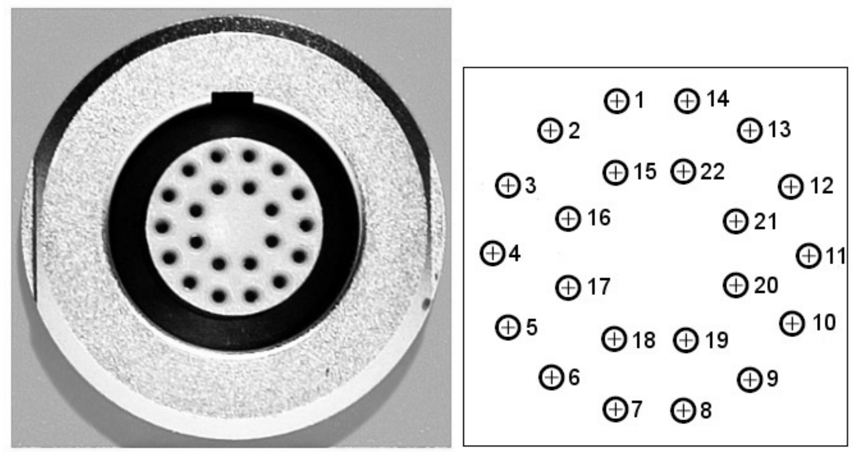


Figure 15 Digital/analog connector

Table 5 Digital/analog connector

Signal	Pin	Description
GND	1	Ground
-	2	Reserved
-	3	Reserved
-	4	Reserved
-	5	Reserved
-	6	Reserved
-	7	Reserved
-	8	Reserved
-	9	Reserved
FPGA_Dig_IO2[0]	10	Digital In, Bit 0
FPGA_Dig_IO2[1]	11	Digital In, Bit 1
FPGA_Dig_IO2[2]	12	Digital In, Bit 2
FPGA_Dig_IO2[3]	13	Digital In, Bit 3
FPGA_Dig_IO2[4]	14	Triggerout Pin 0
FPGA_Dig_IO2[5]	15	Triggerout Pin 1
FPGA_Dig_IO2[6]	16	Triggerout Pin 2
FPGA_Dig_IO2[7]	17	Triggerout Pin 3
CH0	18	Analog In 0 (0 – 10V)

Table 5 Digital/analog connector (continued)

Signal	Pin	Description
CH1	19	Analog In 1 (0 – 10V)
CH2	20	Analog In 2 (0 – 10V)
CH3	21	Analog In 3 (0 – 10V)
-	22	reserved

VPT1000 Hardware MSO connector

If the Oscilloscope mode is activated (see "MSO option settings" on page 99 for more information) this connector provides the FlexRay triggering, decoding, and error information for detailed protocol analysis with the Agilent MSO6000/7000 series oscilloscope.

Power Connector

The VPT1000 Hardware can be powered over the power connector or by an USB host device.

The pin assignment is shown in Figure 16 and Table 6.

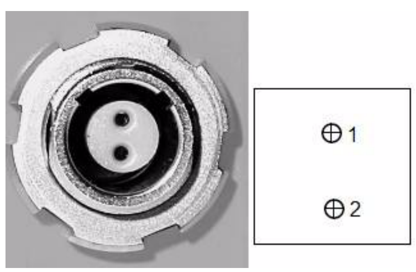


Figure 16 Power Connector

Table 6 Power connector

Signal	Pin	Description
Vin_Bat	1	+6...30V Power Supply
GND	2	Ground

System Connector

The system connector comprises the serial lines (RS232), reset and power supply for external purpose and lines for external buttons and LEDs. The pin assignment is given in Table 7 and Figure 17. Pin 12 is an output pin, which is also routed to the front LED. If active, the front LED will light up red.

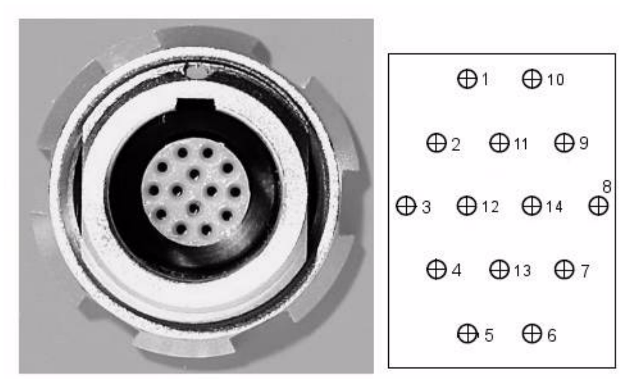


Figure 17 System connector

Table 7 System connector

Signal	Pin	Description
RS232_TXD	1	TXD line for RS232 interface
RS232_RXD	2	RXD line for RS232 interface
RS232_RTS	3	RTS line for RS232 interface
RS232_CTS	4	CTS line for RS232 interface
SYS_PORESET_N	11	Power-on reset line
PSC3_2	12	Front LED
GND	13	Ground
3V3	14	3.3V power line

Ethernet Connector

An 8-pin ODU connector is used for connecting the VPT1000 Hardware to a 10/100Mbps network.

The pin assignment is shown in Figure 18 and Table 8.

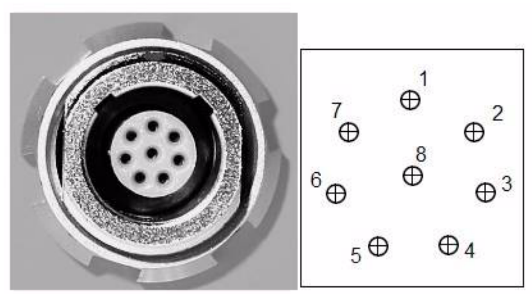


Figure 18 Ethernet connector

Table 8 Ethernet connector

Signal	Pin	Description
C_TPOP	1	Ethernet connection
C_TPON	2	
C_TPIP	3	
C_TPIN	6	

The cable shipped with the VPT1000 Hardware is a crossover cable, which works for a direct connection between VPT1000 Hardware and PC, but also with HUBs and switches providing crossover detection.

USB Connector

The *USB* connector comprises two pins for the 5V bus voltage and two pins for the differential data lines. A single USB port can provide 500mA at 5V, meaning 2.5 Watts, sometimes even more. Since the board requires more than 2.5 Watts, two *USB* ports connected with a special *USB* adapter cable have to be used.

The pin assignment is shown in Figure 19 and Table 9.

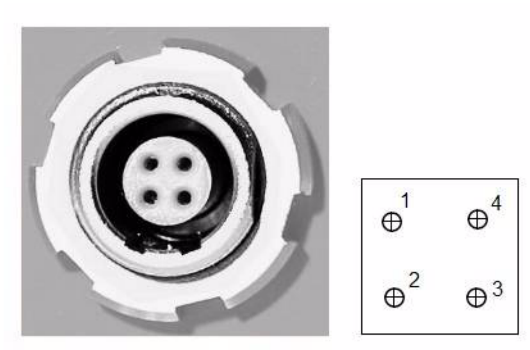


Figure 19 USB connector

Table 9 USB connector

Signal	Pin	Description
Vin_USB	1	USB connection
CUSB_DM	2	
CUSB_DP	3	
Ground	4	

Termination Resistors

The VPT1000 Hardware has no internal termination resistors for FlexRay TJA1080, FlexRay RS485 and CAN. Be sure to use external termination.

Warranty

The VPT1000 Hardware housing must not be opened by the user in order to maintain warranty.

3
VPT1000 PC Software (N5479A)

VPT1000 PC Software is a powerful analyzing tool for FlexRay, CAN and analog/digital data. It provides virtual instruments for analysis on raw data level and on a signal level. For online analyses the VPT1000 PC Software supports the VPT1000 Hardware. Online data can than be logged during an measurement (*Host Logging*) and analyzed in more detail afterwards (*Offline Analyses*).

Additionally, a small command line tool *BDLogConverter* is delivered with VPT1000 PC Software, which converts log files (binary format) recorded during *Host Logging* into ASCII text files or comma separated value files (csv files) (see Chapter 5, “Log File Conversion,” starting on page 107).

Installation

The installation files can be obtained from the Agilent Technologies download area at http://www.agilent.com/find/vpt1000. Install the VPT1000 PC Software by launching the setup file. Removal of the VPT1000 PC Software can be done by launching the setup file again or via *Add/Remove Programs* from the *Windows Control Panel*.

System Requirements

In order to run the VPT1000 PC Software smoothly the basic system requirements are:

- Standard PC compatible workstation with *Windows XP* operating system installed.
- CD ROM drive for installation
- Approximately 50 MB of free hard disk space
- 128MB RAM
- Network adapter card with 100MBit Ethernet or USB 2.0 or a free USB 2.0 port
- A serial interface (RS232) for configuration of the bootloader settings including the IP address of the VPT1000 Hardware.

Installing VPT1000 PC Software

Ensure that you are logged in as administrator or that you have administrator privileges. Launch the downloaded *setup* file. If the installation does not start automatically, open *Windows Explorer* and double-click on the *setup* file.

With *Next* or *Back* you can navigate through the installation dialogs.

1 First, a welcome dialog is displayed:

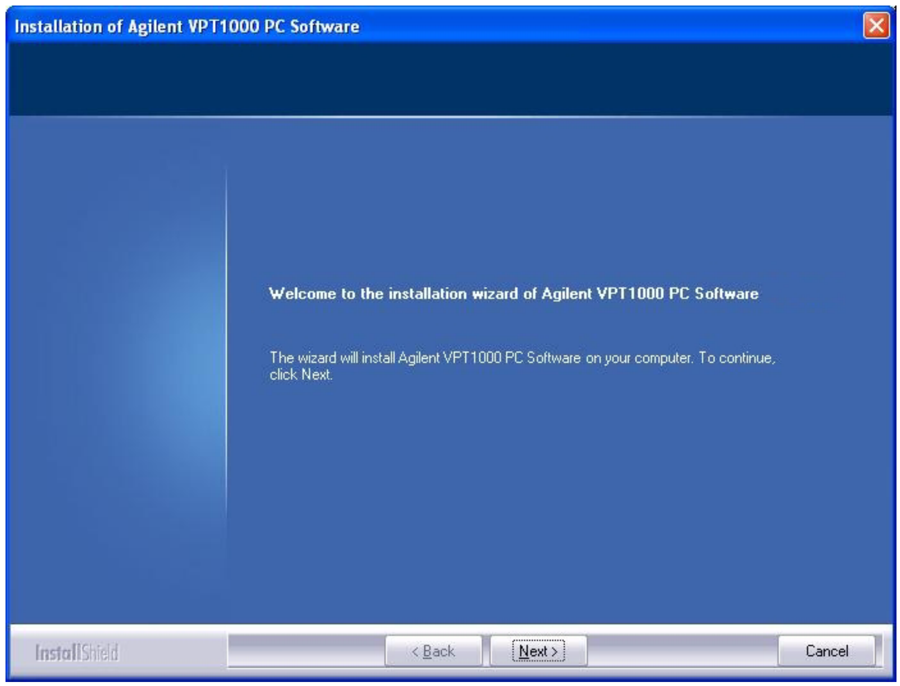


Figure 20 Welcome to VPT1000 PC Software

2 Read the Software License Agreement. The license agreement can also be found in the VPT1000 PC Software installation directory (File *License.pdf*).

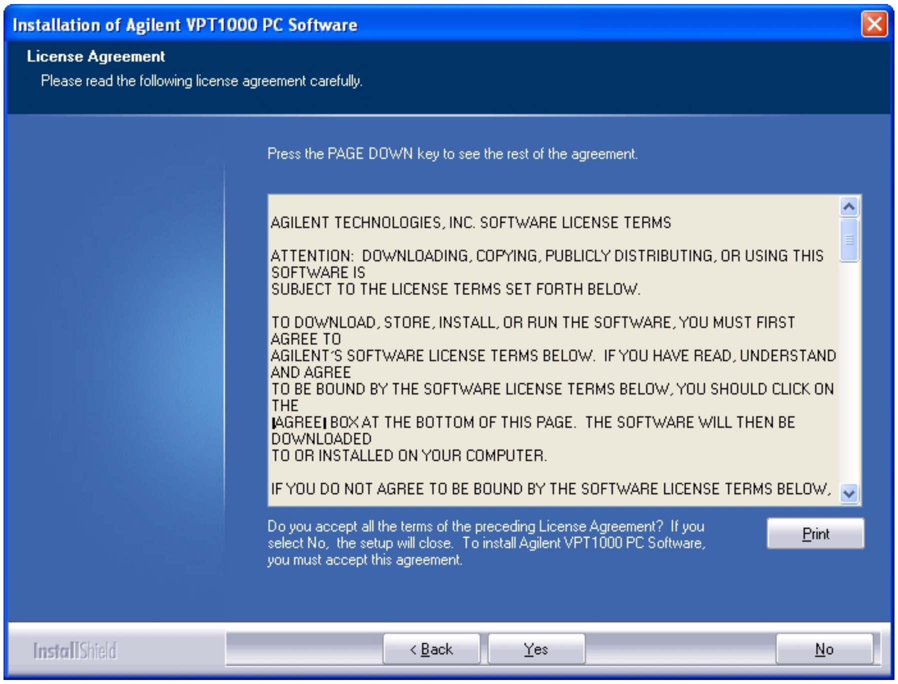


Figure 21 License agreement

3 Enter your user name and your company's name:

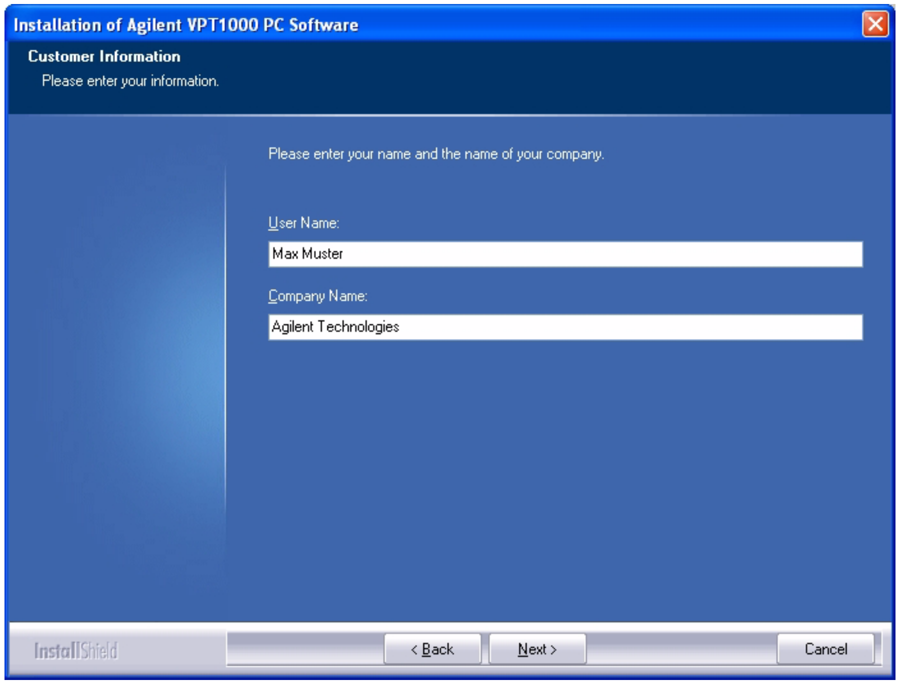


Figure 22 Customer information

4 Select the folder on your hard disk where you would like to have the VPT1000 PC Software files installed. To change the default directory, click *Browse*.

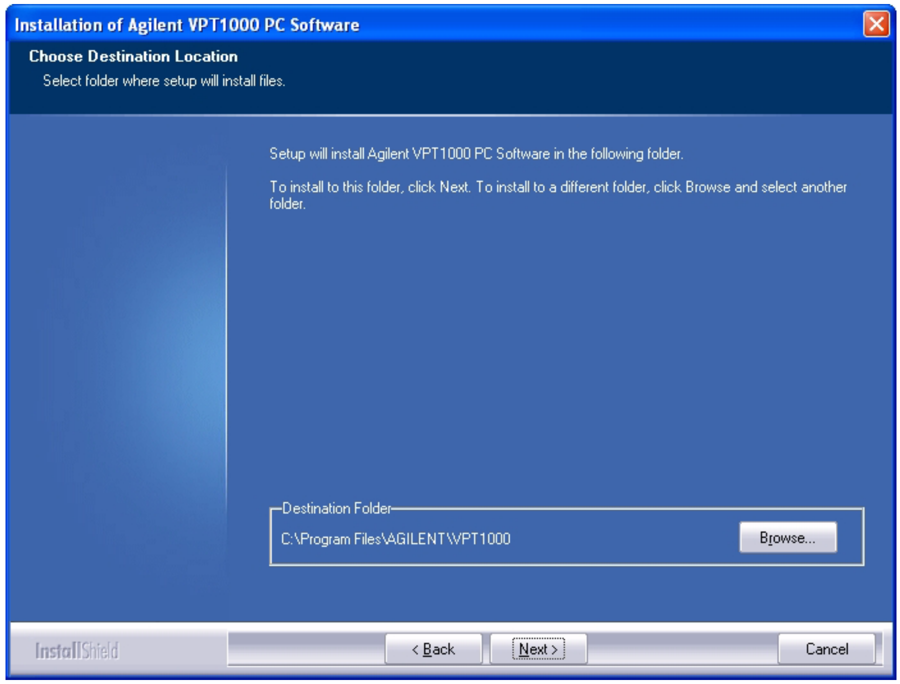


Figure 23 Destination location

5 Specify whether file type *vpf* shall be associated with VPT1000 PC Software:

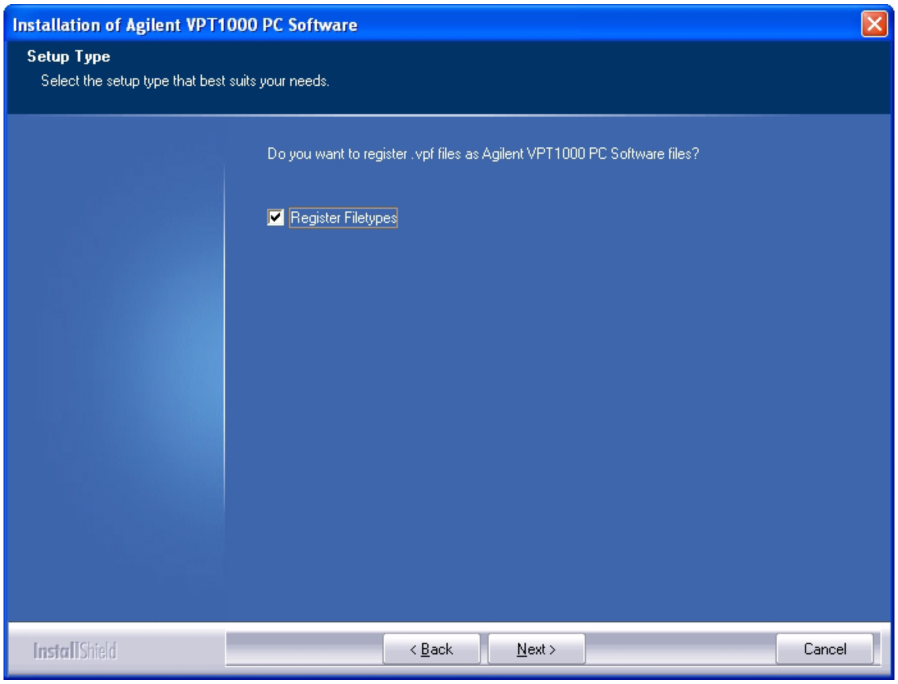


Figure 24 File type registration

6 Specify the license file path (see Figure 25):

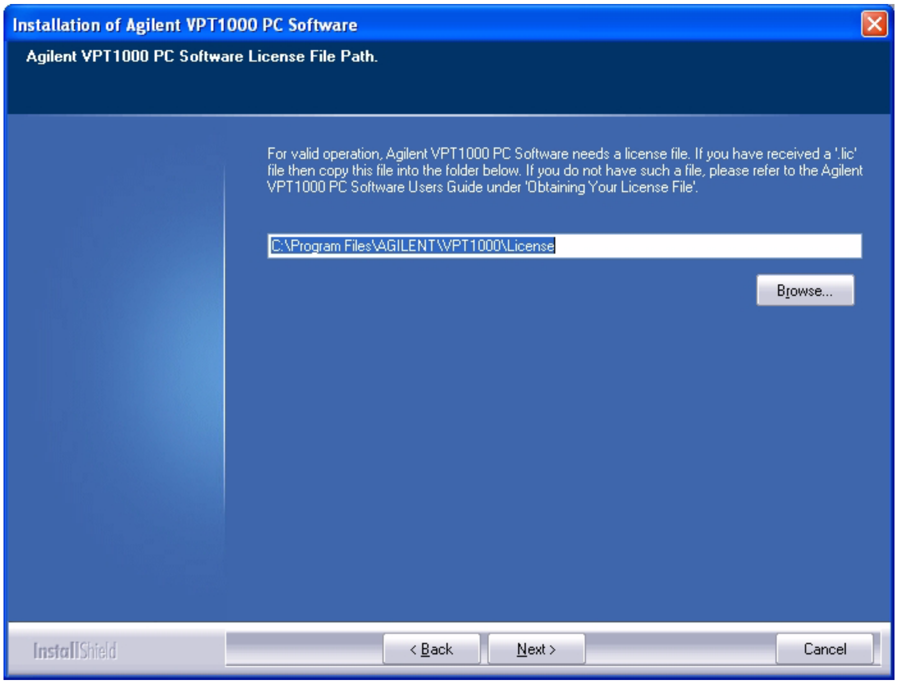


Figure 25 License file path specification

7 Before setup starts copying the files onto your hard disk, the selected options are displayed again. If you desire to change one of the install options, you can step back to the desired dialog by clicking *Back*. Clicking *Next* will start the copy process.

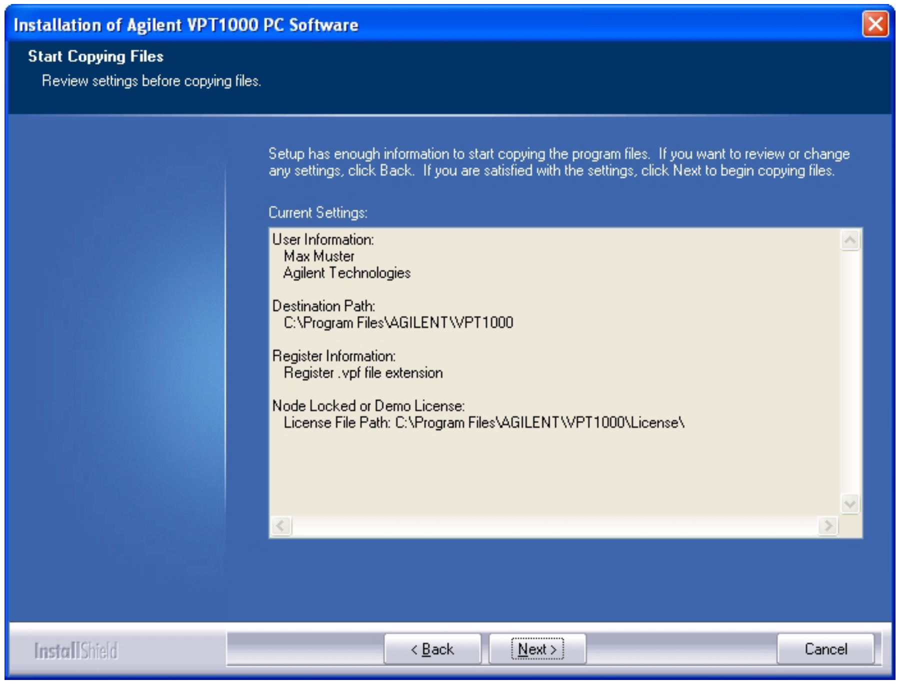


Figure 26 Installation options overview

8 After installation you can open the release notes. Reading this file is recommended as it contains the most recent information concerning VPT1000 PC Software.

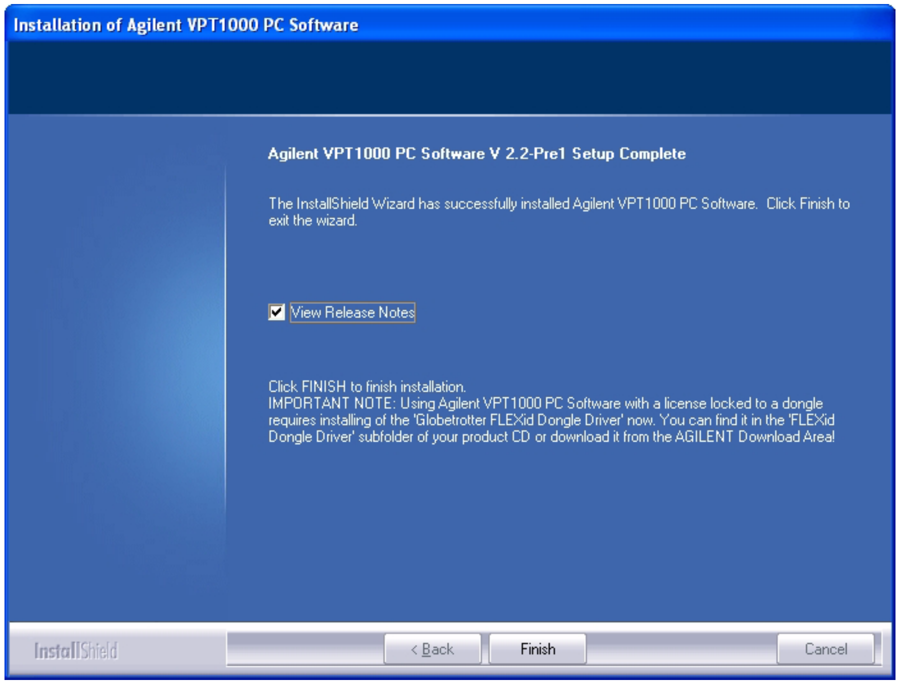


Figure 27 Readme file

Licensing

Each installed copy of the VPT1000 PC Software needs a valid license for operation. Licenses are organized in license files. A license file can consist of several entries, which determine the granted functionality. These entries are called *license features*. The license feature required for VPT1000 PC Software is:

```
FEATURE VPT1000_PC_SW agilent 2.0
```

If no license file with the appropriate feature is provided, the VPT1000 PC Software will refuse operation. See "Obtaining A License File" on page 41.

VPT1000 PC Software uses Macrovision FLEXlm License Manager tool. For further information, please refer to the Macrovision website

"http://www.macrovision.com".

Node locked licenses are based on the MAC Address. VPT1000 PC Software will only run on the workstation containing the LAN card with this MAC Address.

To operate a node locked license, a license file is needed. The following section describes how this can be obtained.

Obtaining A License File

You can obtain a license file by contacting the Agilent Technologies support via:

- "http://www.agilent.com/find/softwarelicense"

In order to generate a valid license file, the following information is required:

1. Order number
2. MAC address of the workstation's primary network interface card. To obtain, go to Start/Run, and type "cmd.exe". On the command line that opens, type "getmac /v", and note the Physical Address reported for the Local Area Connection.

Detailed instructions for license redemption are available on the web site at the start of the redemption process. After registration, you will receive an e-mail with further instructions on how to install the license.

For Technical assistance contact us:

Americas
Canada (877) 894-4414
Latin America 305 269 7500
United States (800) 829-4444

Asia Pacific
Australia 1 800 629 485
China 800 810 0189
Hong Kong 800 938 693
India 1 800 112 929
Japan 81 426 56 7832
Korea 080 769 0800
Malaysia 1 800 888 848
Singapore 1 800 375 8100
Taiwan 0800 047 866
Thailand 1 800 226 008

Europe
Austria 0820 87 44 11
Belgium 32 (0) 2 404 93 40
Denmark 45 70 13 15 15
Finland 358 (0) 10 855 2100
France 0825 010 700
Germany 01805 24 6333*
*0.14€/minute
Ireland 1890 924 204
Italy 39 02 92 60 8484
Netherlands 31 (0) 20 547 2111
Spain 34 (91) 631 3300
Sweden 0200-88 22 55
Switzerland (French)
44 (21) 8113811 (Opt 2)
Switzerland (German)
0800 80 53 53 (Opt 1)
United Kingdom 44 (0) 7004 666666
Other European countries:
www.agilent.com/find/contactus

Getting Started

This section describes the configuration of VPT1000 PC Software.

A dedicated Ethernet connection is required between the VPT1000 Hardware and the PC on which VPT1000 PC Software is running. For configuration of the network settings in the VPT1000 Hardware see "Ethernet Configuration" on page 14.

The VPT1000 Hardware must be configured with appropriate FlexRay parameter values. VPT1000 PC Software allows importation of FlexRay configurations provided in the following file formats:

- FIBEX (*Field Bus Exchange Format*, a standardized, xml based exchange format defined by the ASAM consortium)
- BOR (*Binary Object Repository* from EB tresos Designer from Elektrobit Inc.)

Alternatively, the FlexRay configuration may be entered manually (see "FlexRay protocol parameters" on page 90).

VPT1000 PC Software Basic Configuration

Open the VPT1000 PC Software. In the *Log Window* at the bottom, the message "License checkout completed" indicates that the license checkout has been completed successful. If an Error Message pops up, ensure that the licensing is configured correctly. See "Licensing" on page 40.

A new project can be created by selecting *File->New Project* within the menu or the *New Project* icon in the toolbar.

The connection settings of the VPT1000 Hardware used may be configured in the *Settings Dialog* of the VPT1000 PC Software as shown in Figure 28 (menu item *Measurement->Settings...*). When using VPT1000 Hardware with USB the default setting is okay. Otherwise in the *General* Tab the correct hardware and connection type, and in case of using Ethernet the correct IP address of the VPT1000 Hardware hardware must be selected. Note that both the local IP address and the IP address of the VPT1000 Hardware must be set for using Ethernet connection (see "Ethernet Configuration" on page 14 for more details).

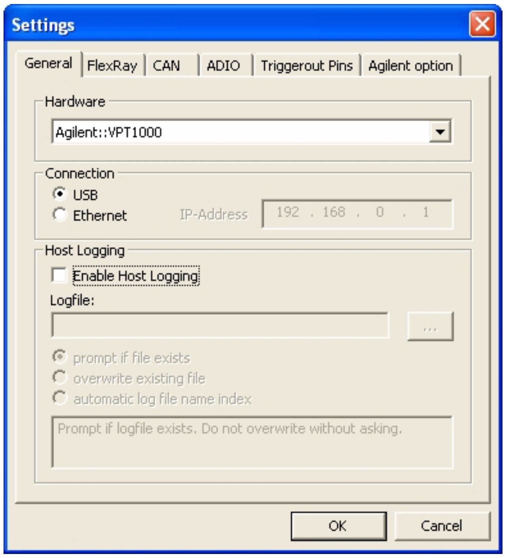


Figure 28 General settings

The VPT1000 Hardware together with VPT1000 PC Software allows recording FlexRay, CAN and analog/digital input data into a log file on the PC's hard disk. Data logging is enabled with the check box *Enable Host Logging*. For further information see Section "Host Logging" on page 103. For simple monitoring *Host Logging* may remain switched off.

To setup the connection to the VPT1000 Hardware select *Connect* in the *Measurement Menu*. In the *LogWindow*, the message "Connection established" should appear. If this is not the case, check the Ethernet connection (ping the IP address of the VPT1000 Hardware), check the physical connection, and check the IP address in the *SettingsDialog* of VPT1000 PC Software.

Connect the VPT1000 Hardware to the FlexRay network, startup the FlexRay network, so that frames are transmitted and choose *Play Forward* in the *MeasurementMenu* of VPT1000 PC Software or click the *Play Forward/Stop Button* on the *Control Panel* (Button 8 in Figure 69). In the *Raw DataWindow* (see "Raw Data Window" on page 45) the monitored FlexRay traffic should be displayed. If not (the *LogWindow* shows the message "No new frames received"), ensure that the FlexRay network is correctly connected to the VPT1000 Hardware. I.e., ensure that the channels are not interchanged, ensure the bus termination is correct etc. If the FlexRay network is OK, ensure that the FlexRay configuration used to setup the VPT1000 Hardware is consistent with the actual FlexRay cluster. If raw data is displayed in the *Raw DataWindow*, the VPT1000 Hardware and VPT1000 PC Software are operational. For detailed information about monitoring, logging and frame transmission see Chapter 4, “Working With VPT1000 PC Software,” starting on page 87.

Windows And Elements

Log Window

This window displays different log messages such as informative messages, warnings and errors. The window has four columns:

- *Class* is the classification of the event. Possible classifications are *Info*, *Warning* and *Error*.
- *Source* indicates the source of the log event causing the log message.
- *Description* provides additional information about the log event.
- *Time* gives the point in time the log event occurred.

The occurrence of an error or warning is additionally indicated by a red X-sign on the window's left (see Figure 30). This is helpful if an error or warning message is not visible in the *Log Window* due to numerous subsequent informative messages. The user may then scroll back in the Log Window and identify the reason for the error or warning. The X-sign may be cleared by clicking on it.

Figure 29 shows the *LogWindow* with an informative messages after the connection to the VPT1000 Hardware was successfully established.

The error messages as shown in Figure 30 are displayed when the connection to the VPT1000 Hardware fails.

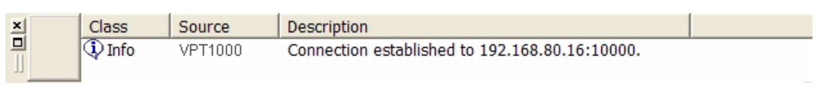


Figure 29 Log Window without error or warning

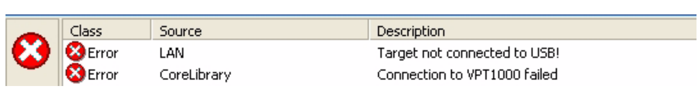


Figure 30 Log Window with errors

Status Bar

The *Status Bar* shows the current state of the VPT1000 PC Software and the current time of the measurement.

Figure 31 Status Bar when disconnected from VPT1000 Hardware

If the VPT1000 PC Software is not connected to the VPT1000 Hardware, the *Status Bar* shows the message "Disconnected".

Figure 32 Status Bar when connected to VPT1000 Hardware

If the VPT1000 PC Software is connected to the VPT1000 Hardware, the *Status Bar* shows the message "Connected", the address of the connection to the VPT1000 Hardware (IP address or USB), and the monitoring status for FlexRay (VPT1000 Hardware: *async only* for asynchronous only mode or *mixed* for mixed mode; *filtered* if any FlexRay filter is enabled), CAN A/B (*off/on* – CAN monitoring via CAN Controller A/B disabled/enabled; *filtered* if any CAN filter is enabled), and ADIO (*off/on* – monitoring of analog/digital input disabled/enabled).

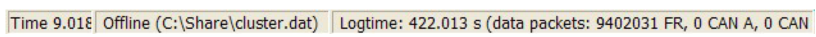


Figure 33 Status Bar during offline analysis

During offline analysis the *Status Bar* shows the message "Offline", the name of the log file, the total log time in seconds, the total number of data packets for FlexRay, CAN A/B and ADIO.

Raw Data Window

The data packets monitored by the VPT1000 Hardware are displayed in the *Raw Data* View. Therefore two display modes are provided by this window. In *fixed* viewing mode an overview of the measured data packets is displayed using one line per received packet ID. Regarding to the actual measurement time each line contains the most recent data of the corresponding ID. In *chronological* viewing mode the measured data packets are displayed in the same order as they were received by the VPT1000 Hardware.

By default the *Raw Data Window* is in fixed viewing mode while the chronological display mode can be activated by pressing the chronological button (see Figure 87, button 20) of the toolbar.

The content of the data packets is organized in columns. The following columns are displayed:

- *Bus:* Indicates the data source (FR – FlexRay, CAN, DIG – digital input, ANA – analog input, TRIGGER – start/stop trigger, marker, triggerout).
- Slot: Gives the FlexRay slot number the item was captured in. The slot number is only available for the VPT1000 Hardware if the hardware is synchronous to the cluster.

- *ID:* Gives the decoded FlexRay/CAN frame ID or indicates a FlexRay symbol or an unknown FlexRay communication element.
- *Channel:* Indicates the channel on which a FlexRay communication element or the CAN controller with which a CAN frame was received.
- *Time:* Shows the time stamp of the displayed item provided by the VPT1000 Hardware. The time stamp granularity is 25ns.
- *Sync:* Indicates whether the sync frame indicator of a FlexRay frame is set. "1" means that the FlexRay frame is a sync frame.
- *Cycle:* Shows the decoded cycle counter contained in a FlexRay frame.
- *Length:* Shows the decoded length of a valid FlexRay/CAN frame in Bytes.
- *Data:* Shows the payload contents of a FlexRay/CAN frame in hex format and provides additional information for FlexRay symbols, for analog/digital input, and for triggers events. The values for analog and digital signals are shown in hex format and in Volts.
- *Status:* Shows the status of a received FlexRay/CAN frame or a FlexRay symbol. For a detailed information on the FlexRay frame status see "FlexRay status bits" on page 112 and on the CAN frame status see "Status bits of the CAN frame data packet" on page 116.
- *Slot Status:* While the "Status" column shows status information about a received frame, the "Slot Status" column contains information about the FlexRay slot in which the frame has been received. The "Slot Status" is only available if the measurement hardware is synchronous to the FlexRay bus.

Following "Slot Status" values are available:

- VFR - Valid Frame Received
- SED - Syntax Error Detected
- CED - Content Error Detected
- ACI - Additional Communication Indicator
- SBV - Slot Boundary Violation
- *NFI:* Indicates whether the null frame indicator of a FlexRay frame is set. "0" means that the FlexRay frame is a null frame.
- *Startup:* Indicates whether the startup frame indicator of a FlexRay frame is set. "1" means that the FlexRay frame is a startup frame.

If design information was imported from FIBEX, BOR, or XCDEF following additional columns for FlexRay packets are available:

- Frame: Displays the name of the measured FlexRay frame.
- *Frame Triggering:* Displays the name of the frame triggering which has triggered the FlexRay frame.
- *BC:* Base cycle of the frame triggering.
- *CR:* Cycle repetition of the frame triggering.

- *Segment:* Indicates whether the FlexRay frame is sent in the *static* or *dynamic* part of the communication cycle.
- Tx ECU: Shows the name of the ECU which has sent the corresponding frame.
- *Service:* Shows the service type which is assigned to the FlexRay frame. The service type can be one of the following:
- APPLICATION
- NM
- TP
- DIAG_STATE
- DIAG_REQUEST
- DIAG_RESPONSE
- PRE CONFIGURED XCP
- RUNTIME CONFIGURED XCP
- SERVICE
- OTHER

A FlexRay frame with header CRC error, TSS violation or coding error within the header is displayed with an *unknown* ID because such a frame could contain an invalid frame ID.

A detailed description of all supported data packets and its content can be found in Chapter 6, “Measurement Data Representation,” starting on page 109.

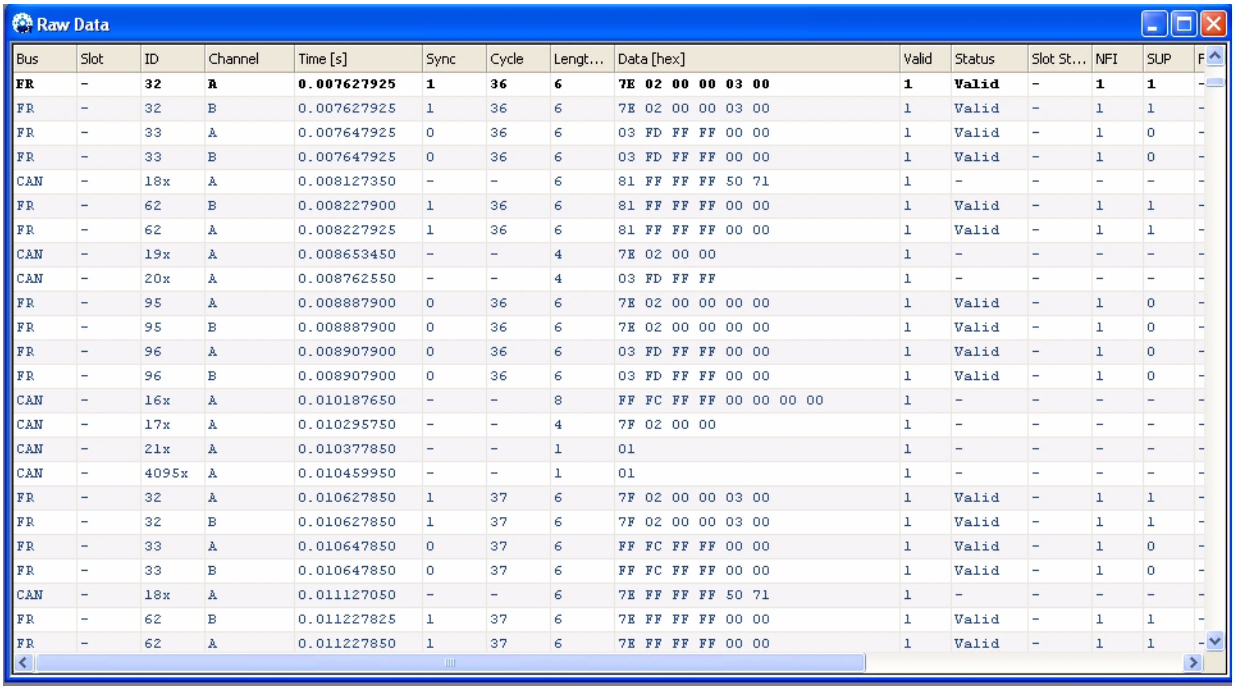
Chronological View

The chronological view is a listing of all received data packets in the exact temporal order as they are received by the VPT1000 Hardware.

The chronological view works with a data buffer of 2000 data packet entries. In online measurements, the view is actualized with the complete buffer content and after stopping measurement the view can be scrolled back to the beginning of the buffer size.

In offline mode the chronological view works with the same buffer size. When reaching the beginning or the end of the buffer the position in the offline file must be changed (see "Offline Analysis" on page 104).

Figure 34 shows a chronological view of log data containing FlexRay and CAN data.



Bus	Slot	ID	Channel	Time [s]	Sync	Cycle	Lengt...	Data [hex]	Valid	Status	Slot St...	NFI	SUP	F
FR	**-**	**32**	**A**	**0.007627925**	**1**	**36**	**6**	**7E 02 00 00 03 00**	**1**	**Valid**	**-**	**1**	**1**	-
FR	-	32	B	0.007627925	1	36	6	7E 02 00 00 03 00	1	Valid	-	1	1	-
FR	-	33	A	0.007647925	0	36	6	03 FD FF FF 00 00	1	Valid	-	1	0	-
FR	-	33	B	0.007647925	0	36	6	03 FD FF FF 00 00	1	Valid	-	1	0	-
CAN	-	18x	A	0.008127350	-	-	6	81 FF FF FF 50 71	1	-	-	-	-	-
FR	-	62	B	0.008227900	1	36	6	81 FF FF FF 00 00	1	Valid	-	1	1	-
FR	-	62	A	0.008227925	1	36	6	81 FF FF FF 00 00	1	Valid	-	1	1	-
CAN	-	19x	A	0.008653450	-	-	4	7E 02 00 00	1	-	-	-	-	-
CAN	-	20x	A	0.008762550	-	-	4	03 FD FF FF	1	-	-	-	-	-
FR	-	95	A	0.008887900	0	36	6	7E 02 00 00 00 00	1	Valid	-	1	0	-
FR	-	95	B	0.008887900	0	36	6	7E 02 00 00 00 00	1	Valid	-	1	0	-
FR	-	96	A	0.008907900	0	36	6	03 FD FF FF 00 00	1	Valid	-	1	0	-
FR	-	96	B	0.008907900	0	36	6	03 FD FF FF 00 00	1	Valid	-	1	0	-
CAN	-	16x	A	0.010187650	-	-	8	FF FC FF FF 00 00 00 00	1	-	-	-	-	-
CAN	-	17x	A	0.010295750	-	-	4	7F 02 00 00	1	-	-	-	-	-
CAN	-	21x	A	0.010377850	-	-	1	01	1	-	-	-	-	-
CAN	-	4095x	A	0.010459950	-	-	1	01	1	-	-	-	-	-
FR	-	32	A	0.010627850	1	37	6	7F 02 00 00 03 00	1	Valid	-	1	1	-
FR	-	32	B	0.010627850	1	37	6	7F 02 00 00 03 00	1	Valid	-	1	1	-
FR	-	33	A	0.010647850	0	37	6	FF FC FF FF 00 00	1	Valid	-	1	0	-
FR	-	33	B	0.010647850	0	37	6	FF FC FF FF 00 00	1	Valid	-	1	0	-
CAN	-	18x	A	0.011127050	-	-	6	7E FF FF FF 50 71	1	-	-	-	-	-
FR	-	62	B	0.011227825	1	37	6	7E FF FF FF 00 00	1	Valid	-	1	1	-
FR	-	62	A	0.011227850	1	37	6	7E FF FF FF 00 00	1	Valid	-	1	1	-

Figure 34 Raw Data Window (Chronological View)

Fixed View

Opposite to the chronological view, the fixed view does not add a new line for all received data packets to the view. Instead, there is one line available per slot, channel, and base cycle for FlexRay data packets respectively per ID and controller for CAN. If no FlexRay design data was imported from FIBEX, BOR, or XCDEF, a base cycle of 0 is assumed. For analog and digital data as well as for trigger data packets there is a separate line available as well.

Each of the available lines contain the most recent data for the actual time. For online monitoring this actual time equals to the current measuring time and for offline analysis the actual time complies to the global time of the navigation bar (see "Offline Analysis" on page 104).

Figure 35 shows a fixed view of log data containing FlexRay and CAN data.

Raw Data

Bus	Slot	ID	Channel	Time [s]	Sync	Cycle	Lengt...	Data [hex]	Valid	Status	Slot St...	NFI	SUP	Fram
FR	**-**	**32**	**A**	**0.007627925**	**1**	**36**	**6**	**7E 02 00 00 03 00**	**1**	**Valid**	**-**	**1**	**1**	**-**
FR	-	32	B	0.004628100	1	35	6	7D 02 00 00 03 00	1	Valid	-	1	1	-
FR	-	33	A	0.004648100	0	35	6	07 FD FF FF 00 00	1	Valid	-	1	0	-
FR	-	33	B	0.004648100	0	35	6	07 FD FF FF 00 00	1	Valid	-	1	0	-
FR	-	62	B	0.005228100	1	35	6	84 FF FF FF 00 00	1	Valid	-	1	1	-
FR	-	62	A	0.005228125	1	35	6	84 FF FF FF 00 00	1	Valid	-	1	1	-
FR	-	95	A	0.005888100	0	35	6	7D 02 00 00 00 00	1	Valid	-	1	0	-
FR	-	95	B	0.005888100	0	35	6	7D 02 00 00 00 00	1	Valid	-	1	0	-
FR	-	96	A	0.005908100	0	35	6	07 FD FF FF 00 00	1	Valid	-	1	0	-
FR	-	96	B	0.005908100	0	35	6	07 FD FF FF 00 00	1	Valid	-	1	0	-
CAN	-	17x	A	0.007297050	-	-	4	7E 02 00 00	1	-	-	-	-	-
CAN	-	21x	A	0.007379150	-	-	1	01	1	-	-	-	-	-
CAN	-	4095x	A	0.007461250	-	-	1	01	1	-	-	-	-	-
CAN	-	18x	A	0.005123650	-	-	6	84 FF FF FF 50 71	1	-	-	-	-	-
CAN	-	19x	A	0.005654750	-	-	4	7D 02 00 00	1	-	-	-	-	-
CAN	-	20x	A	0.005763850	-	-	4	07 FD FF FF	1	-	-	-	-	-
CAN	-	16x	A	0.007188950	-	-	8	03 FD FF FF 00 00 00 00	1	-	-	-	-	-
CAN	-	unk...	A	0.001188550	-	-	-	-	1	-	-	-	-	-

Figure 35 Raw Data Window (Fixed View)

Working with the Raw Data Window

To switch between chronological and fixed view click the *Fixed/Chronological Button* as shown in Figure 87 (Button 20). For detailed information on how to navigate through the raw data window refer to "Offline Analysis" on page 104.

For sorting the lines of the raw data window the user can click on the header of the column to sort. For sorting to more than one column hold the *Shift* key down and press on all columns to sort to. For restoring the original sort order right-click on the header area of the raw data window and choose *Reset Sort Order* of the context menu.

The display order of the columns can be exchanged by drag and drop of the column header to the desired position. Like the sort order, the column order can be restored using the context menu of the header area of the raw data window.

Signal View

The VPT1000 PC Software supports visualization of signals from various data sources. The VPT1000 Hardware hardware provides a FlexRay measurement controller, two independent CAN controllers, four digital, and four analog signal input pins. The analog and digital signals are automatically added to the corresponding data source as shown in Figure 36. Signals within the FlexRay and CAN data source can be either imported from a signal database file (see "Importing signals from a signal database file" on page 50) or entered manually (see "Manually adding signals to the signal tree" on page 52).

Importing signals from a signal database file

VPT1000 PC Software supports different import file types:

- FIBEX File (.xml)
- CanDB File (.dbc)
- DESIGNER PRO File (.bor)
- DESIGNER Legacy File (.xcdef)

FIBEX Files are supported by the FlexRay and the CAN data sources. DESIGNER legacy files can be used only within the FlexRay data source. CanDB Files are supported only within the data source CAN 1 and CAN 2. Depending on the size of the signal database file, the import may take some time.

The signals are imported by a right-mouse click on the data source that contain the signals and then selecting the 'Import Signals' entry in the shown context menu:

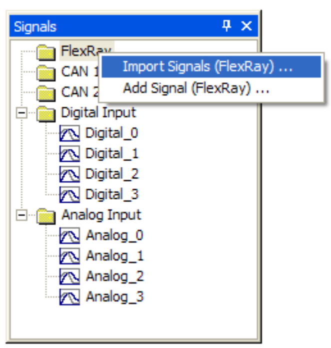


Figure 36 Importing signals for the FlexRay controller

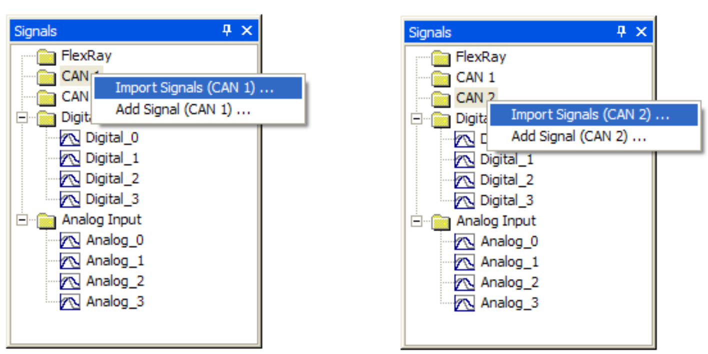


Figure 37 Importing signals for the CAN controllers

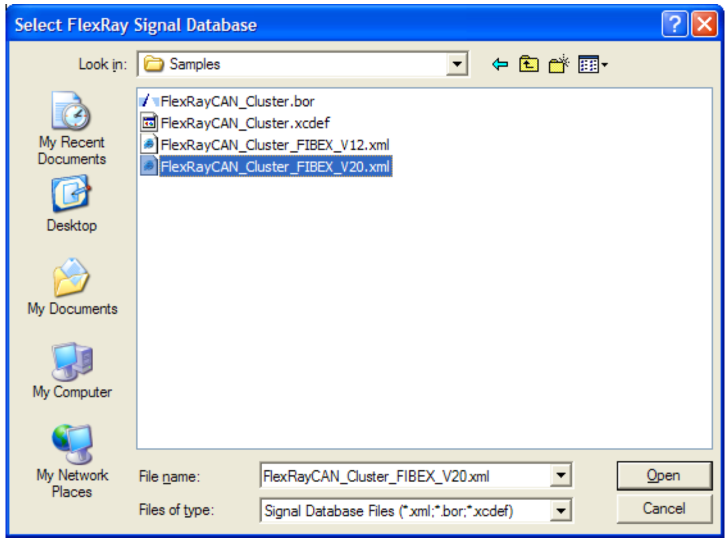


Figure 38 Selecting a FlexRay signal database file

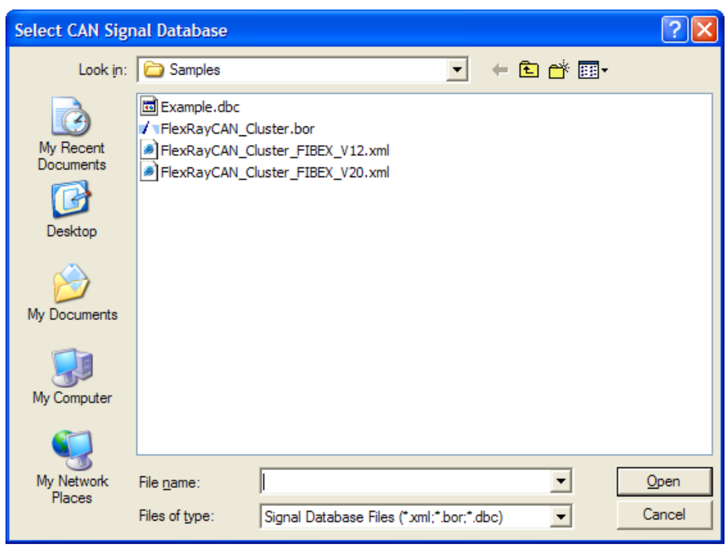


Figure 39 Selecting a CAN signal database file

After importing a signal database, the signals are shown in the signal view:

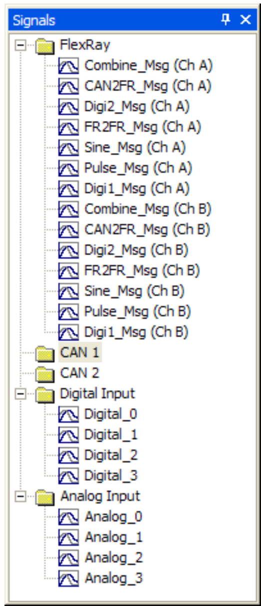


Figure 40 Signal view example containing FlexRay signals

It is possible to import several signal database files into one VPT1000 PC Software project file. Signals having the same name are overwritten during import!

Manually adding signals to the signal tree

Signals can be added manually to the FlexRay and the CAN data sources. A right-mouse click opens a context menu like shown in Figure 41. Selecting the 'Add Signal' entry creates a new signal in the corresponding data source.

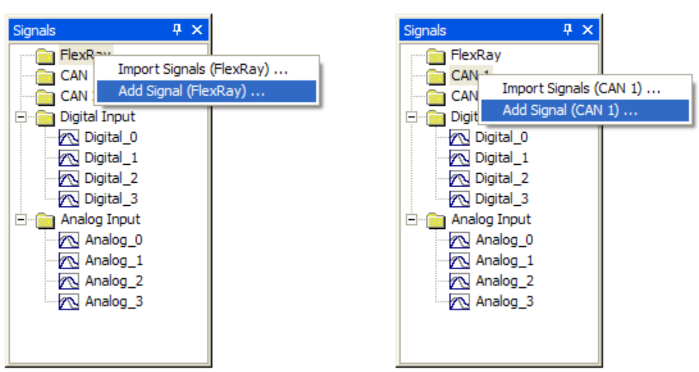


Figure 41 Create Signals

Adding signals to the Digital Input and Analog Input data sources is not possible because these signal inputs depends on the VPT1000 Hardware hardware.

Deleting signals

A right-mouse click on a signal shows a context menu where it is possible to delete the selected signal (see Figure 42).

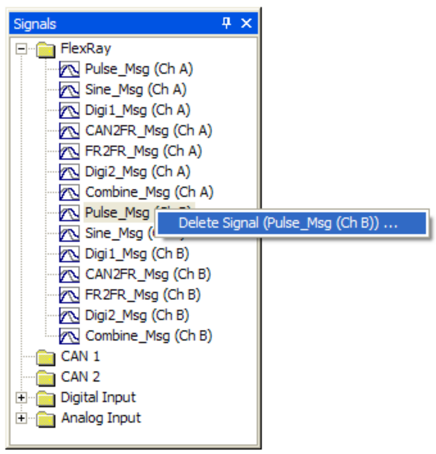


Figure 42 Deleting a signal

Deleting signals from the Digital Input and Analog Input data source is not possible.

Signal Property View

The properties of a selected signal are shown in the Signal Property view. FlexRay and CAN signal representations are rather similar (see Figure 43 and Figure 44). They only differ in the scheduling information. See "Scheduling of FlexRay signals" on page 56 for more information about the FlexRay scheduling information and to "Scheduling of CAN signals" on page 57 for more information about the CAN scheduling information. The analog signals do not have scheduling information. It is not possible to modify the parameters of the digital signals.

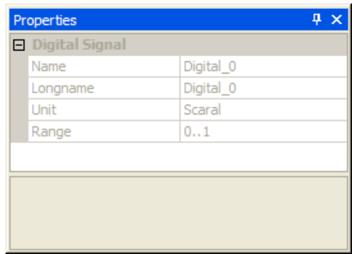


Figure 43 Properties of digital signals

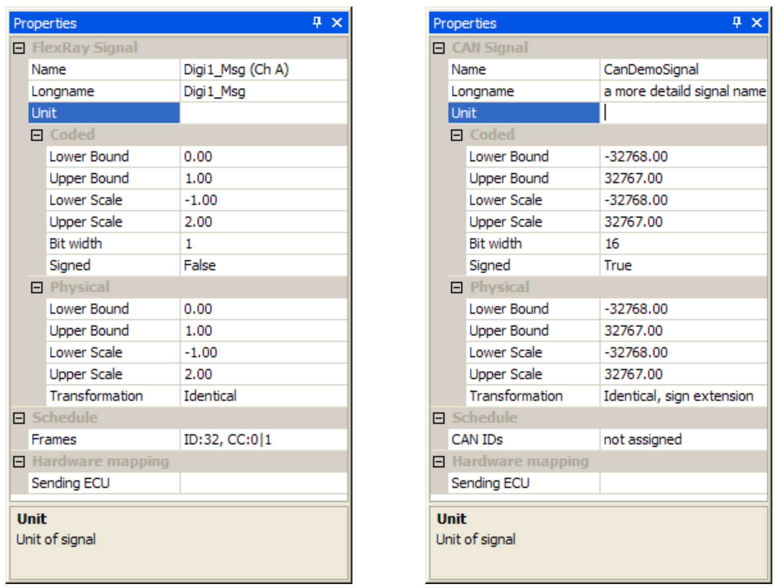


Figure 44 Properties of FlexRay and CAN signals

It is possible to enter and/or modify parameters of the signal using the signal property window. In FIBEX two different codings are defined (coded value, physical value). The conversion function between these representations can be shown or modified by selecting the Transformation property within the Physical section of the signal properties.

Signal Coding

By selecting the Transformation parameter within the Physical section of the signal properties, the coding editor of the VPT1000 PC Software is shown

In FIBEX several types of conversion functions are defined. The VPT1000 PC Software supports the following subset of coding types:

- Identical – no sign extension, useful for unsigned data types
- Identical – with sign extension, useful for signed data types

- Linear (using one interval with a linear function)
- Scale-Linear (using multiple intervals with liner functions)
- Texttable (using intervals with symbols)
- Scale-Linear & Texttable (using intervals with functions and symbols)

Figure 45 shows how to define an identical conversion function. Figure 46 shows how to define signal conversion intervals.

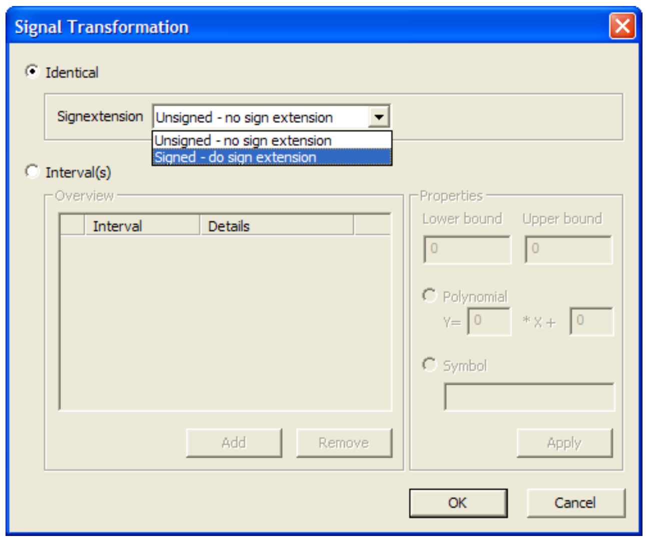


Figure 45 Identical Coding

Multiple intervals can be created by changing the selection from Identical to Interval(s). Using the Add and Remove button it is possible to create several linear functions and symbols. Each interval is defined by its lower bound and upper bound. The conversion of a coded value to a physical value is done only within these boundaries.

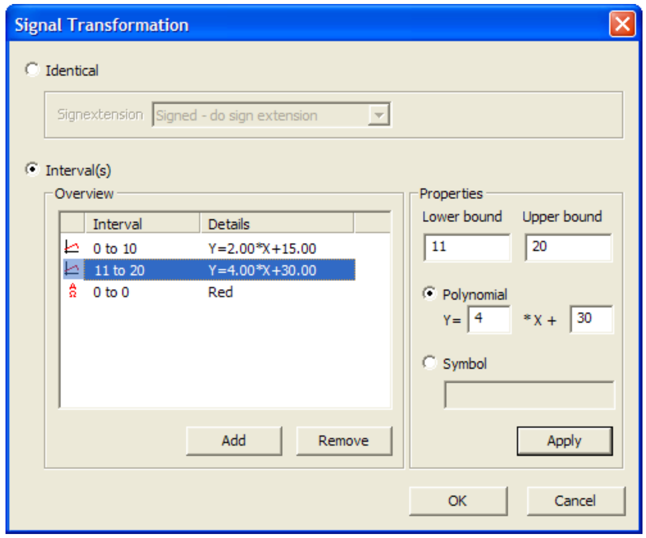


Figure 46 Coding Intervals

Scheduling of FlexRay signals

The position of a signal in the FlexRay schedule is defined by the channel, the slot, the base cycle (BC) the cycle repetition (CR), the bit offset within the payload and the endianness.

Each signal can be located within a PDU. This PDU can be controlled by a PDU update bit. A signal can be part of a signal group. The update of a signal group can be controlled by a signal group update flag.

Signals can be multiplexed. The selection of the signal is then controlled by a specific multiplexer value that is transmitted on the bus using several bits (MuxSize) on a specific bit position (MuxOffset).

Refer to the FIBEX specification for more information about PDUs, signal groups, multiplexer and update bits.

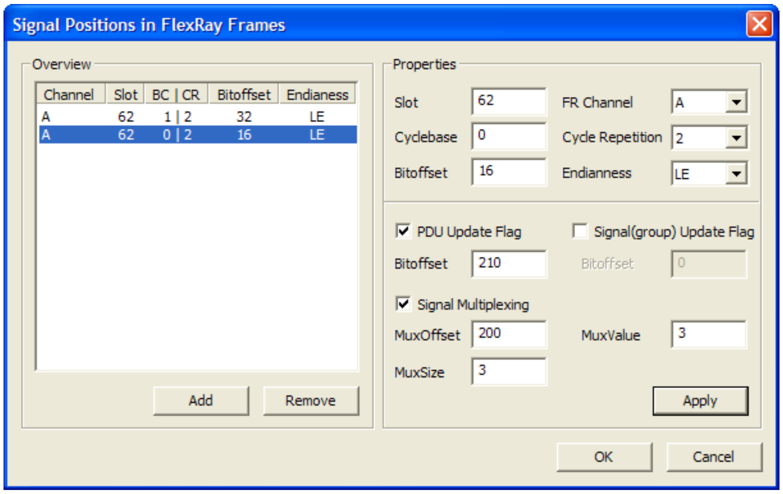


Figure 47 Scheduling of FlexRay signals

Scheduling of CAN signals

The position of a signal in a CAN message is defined by the message identifier (CAN ID), the type of message ID (standard or extended), the bit offset and the endianness. PDU update flags, signal groups, and multiplexer are identical to the specification described in "Scheduling of FlexRay signals" on page 56.

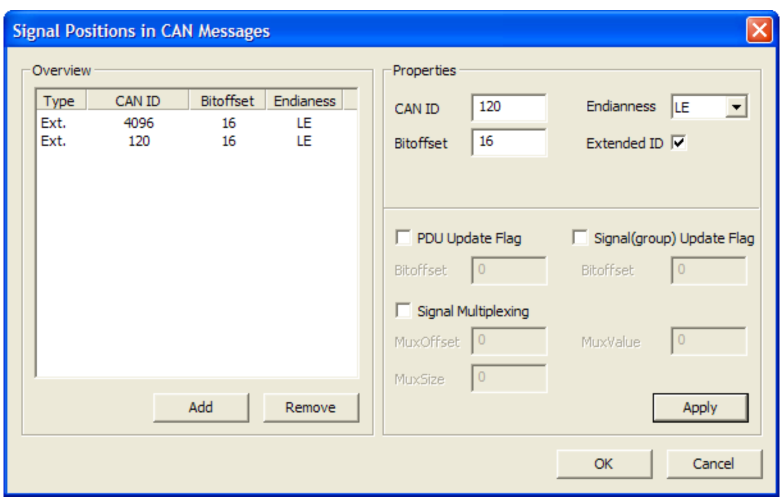


Figure 48 Scheduling of CAN signals

Signals can be located in several CAN messages at different bit positions as shown in Figure 48.

Instrument View

The *Instrument View* is a graphical container for the signal visualization controls and for additional static text elements. Signal visualization controls can be created by dragging a signal into the instrument view and static text elements can be created by the context menu of the instrument view itself. For detailed information on signal visualization controls see "Signal controls" on page 59.

It is possible to handle several instrument views in parallel. With File->New View or with <CTRL><I> a new instrument view is created. Each instrument view has the following properties:

- Caption of the instrument view
- Background color
- Background images

Figure 49 shows an instrument view with a static text element. Furthermore this figure shows the property dialog of the text element, which can be achieved via the context menu of the static text element (pressing right mouse key in the static text element area).

Figure 50 shows the property dialog of the instrument view settings. The instrument view property dialog can be started via the context menu of the instrument view as well (pressing right mouse key somewhere in the view area).

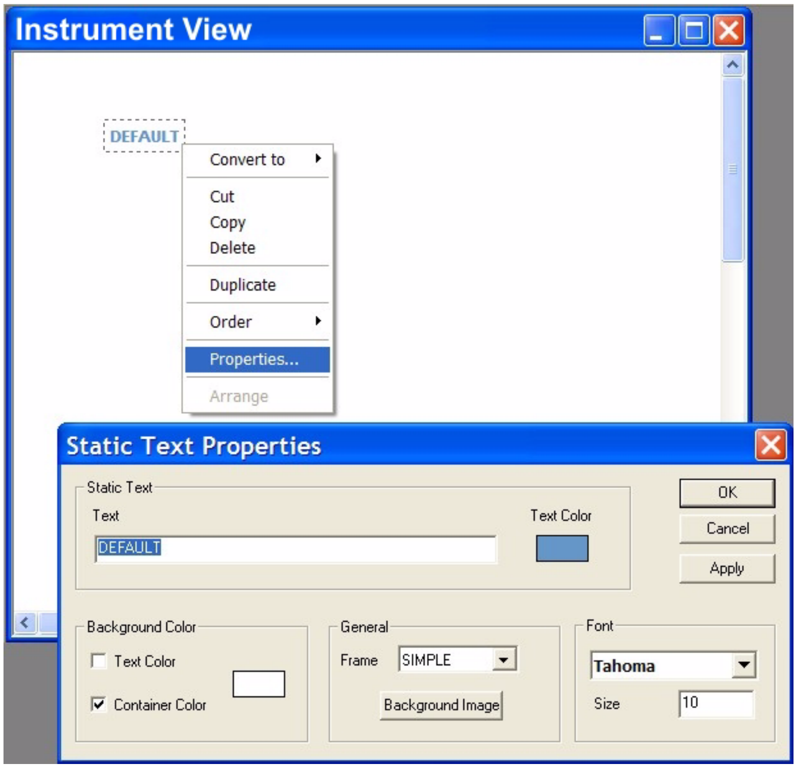


Figure 49 Static Text Properties Window

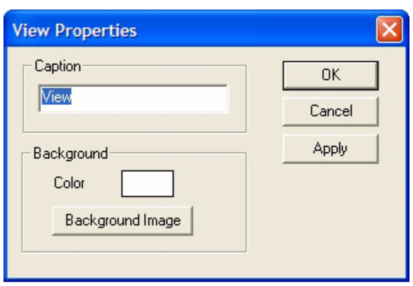


Figure 50 View Properties Dialog

Signal controls

For viewing and analyzing signal values there are 8 signal controls (see Figure 51) of two different categories available:

- Signal controls displaying multiple signals *(Trace, Gauge, Bar* and *Numeric)* and
- Signal controls displaying a single signal *(LED, Bit Set, Icon* and *Text)*.

Figure 51 Overview of available controls

Multi-signal controls include a signal table. The signal table lists the signal(s) contained within the signal control providing a column for the name, unit, and value. Individual signals within a multi-signal control may be enabled or disabled by a click onto the icon shown on the left of the signal name in the signal table.

Creating Controls

A new control is created by dragging and dropping a signal from the *signal view* into an *instrument view*. If the signal database does not contain any information about a signal's control, the default control is assigned, which is a *Trace*. Since current database files do not contain signal control information, a *Trace* is automatically assigned to every signal by default.

If a signal is dragged and dropped onto an existing control, the behavior is different depending on the type of that control:

- A single value control displays the new signal only.

- A multi-value control shows the new signal in addition to the previous signal(s).

A multi-value control can at most display six signals. If there are already six signals assigned and another signal is added, the error message is displayed as shown in Figure 52.

Figure 52 Multi-value control error message

Handling Controls

- Selecting Controls

  Controls can be selected with the left mouse button. The selected control can be copied or duplicated using the context menu (right mouse button).
- Moving Controls

  Any control in a *instrument view* can be moved with the mouse, by holding the left mouse button and moving the mouse. If the snap to grid option is enabled (menu item *Edit->Snap to grid* or *VÌew Window* context menu element *Snap to grid*) the control can only be placed on the grid.
- Changing Properties

  Choose *Properties...* in the context menu of the control to change its properties comprising signal caption, signal unit, signal source, signal color, control size, font, background image, background color, etc.

Figure 53 shows an example of the properties of the *Trace Control*.

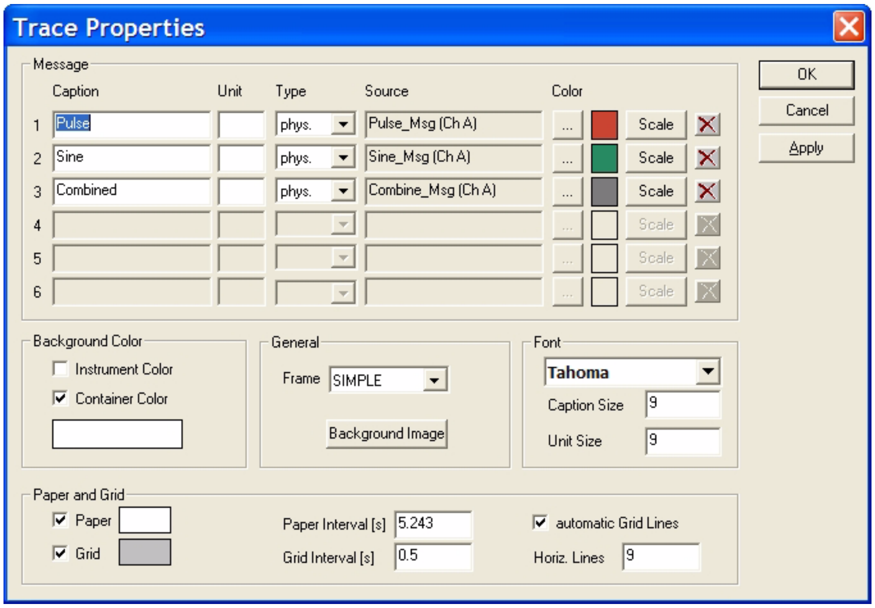


Figure 53 Trace Properties Dialog

The *Message* section allows modifying the signal name as it is displayed in the control, as well as the unit. The Type field specifies if the physical or the coded type of the signal shall be displayed. If a warning was not displayed when opening the .bor or FIBEX file, it is possible to show the signals either in the coded, or in the physical representation. If a warning was displayed for one or several signals, these signals will always be shown in coded representation independent from the type settings.

Clicking the button *Scale* opens an new dialog (see Figure 54) which allows you to modify the properties *Font*, *Layout* and *Range* of a signal.

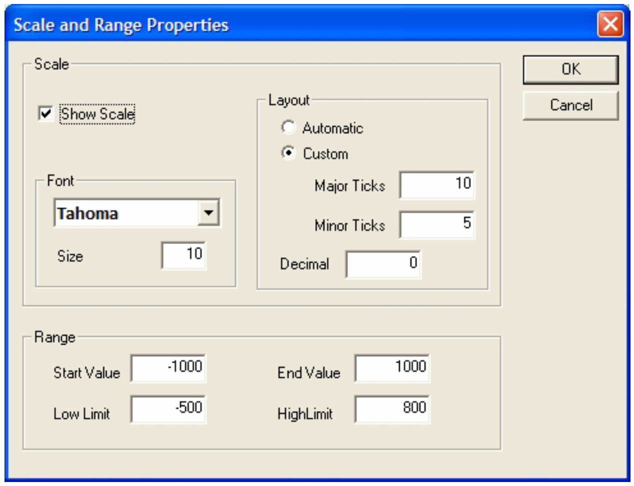


Figure 54 Scale and Range Properties Dialog

To ease the configuration of several controls it is possible to duplicate controls using its context menu and changing the signal source in the properties dialog.

Text Control

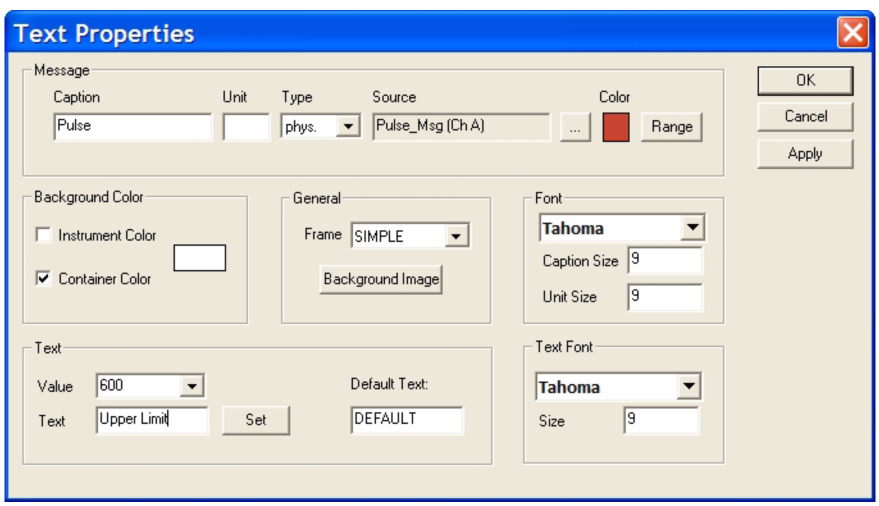


Figure 55 Text Properties Dialog

The *Text Control* shows the signal name and a configurable text label. Signal values may be attached to text labels as shown in Figure 55. Signal value 600 is assigned to the text label "Upper Limit". If the signal assumes a value attached to a text label, this text label is displayed in the *Text Control*. For all signal values not attached to a specific text label, the *Text Control* displays the default text which is "Default" in our example given in Figure 55.

Icon Control

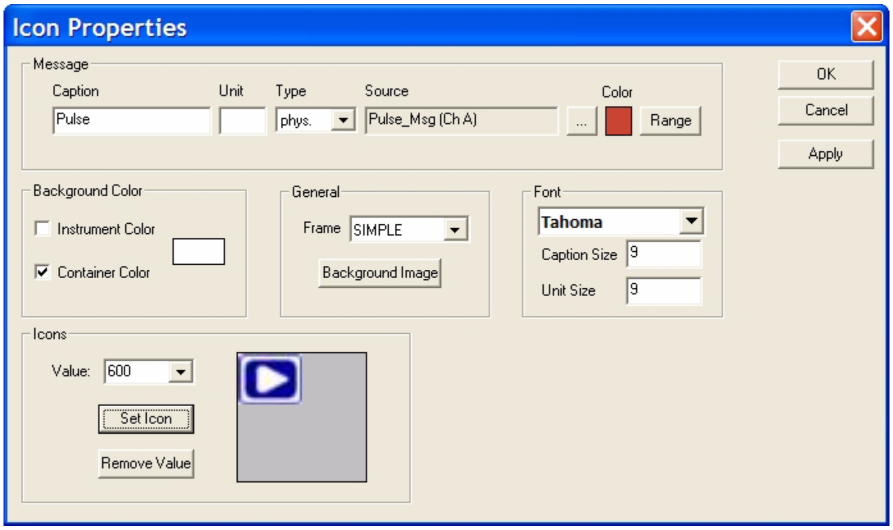


Figure 56 Icon Properties Dialog

The *Icon Control* shows the signal name and a configurable icon. Signal values may be attached to icons as shown in Figure 56. Signal value 600 is assigned to an icon showing an arrow. If the signal assumes a value attached to an icon, this icon is displayed in the *Icon Control*. For all signal values not attached to a specific icon, no icon is displayed within the *Icon Control*.

Led Control

The *Led Control* shows the signal name and a configurable LED. The LED color may be configured as a function of the signal value. If the signal value is greater than 0, the LED assumes the high color which is green in our example as shown in Figure 57. If the signal value is equal to or less than 0, the LED assumes the low color.

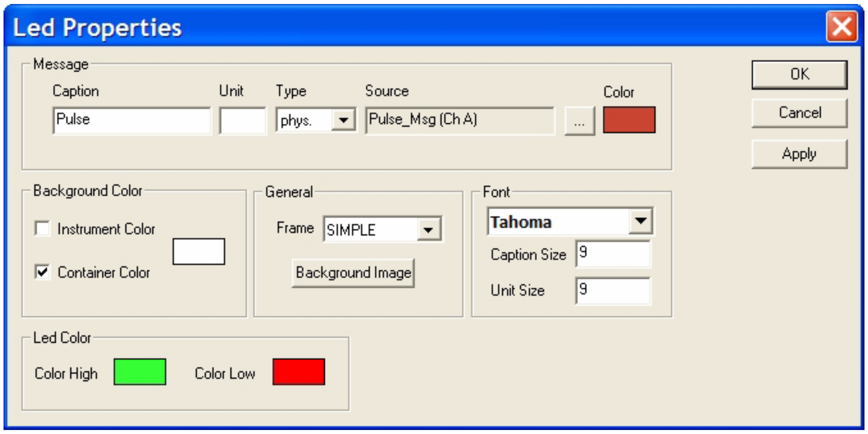


Figure 57 Led Properties Dialog

Bit Set Control

The *Bit Set Control* shows the signal name and a configurable bit set providing signal values in binary representation. Each bit of the bit set may be assigned a high color and a low color and a caption as shown in Figure 58. Bit 10 is configured to be red, if high, and dark blue, if low. The caption of Bit 10 is "10 (Sign)". Additionally, the orientation of the bit set may be set to vertical or horizontal and the bit ordering from right to left (MSB right-most bit, LSB left-most bit in horizontal orientation; MSB upmost bit, LSB lowest bit in vertical orientation) or from left to right (MSB left-most bit, LSB right-most bit in horizontal orientation; MSB lowest bit, LSB top-most bit in vertical orientation). Bit slices of a signal may be selected via configuration of start bit and end bit.

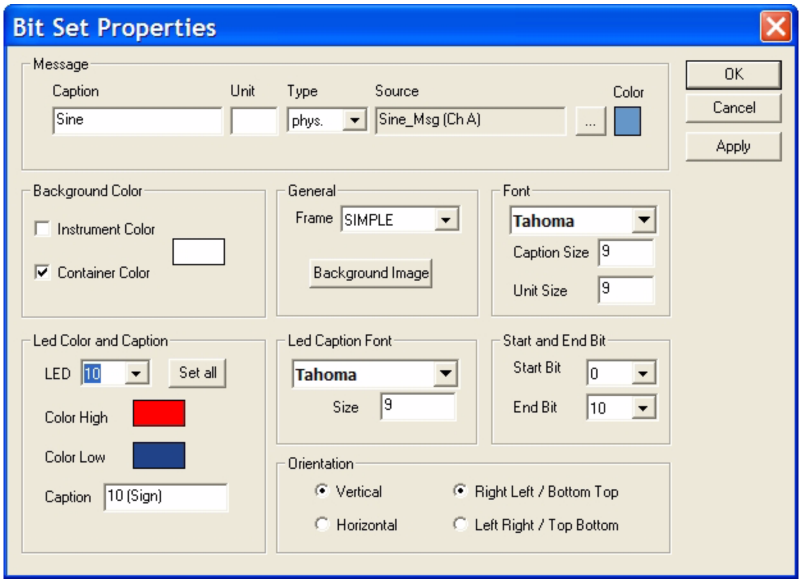


Figure 58 Bit Set Properties Dialog

Numeric Control

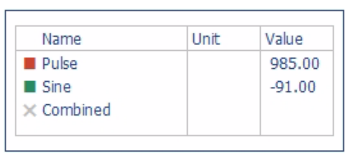


Figure 59 Numeric Control in instrument view

The *Numeric Control* shown in Figure 59 displays signals in a table representation providing three columns for the name, the unit and the value. The width of the columns may be changed via mouse interaction within the header of the signal table. At most six signals may be represented at once by a *Numeric Control*. The numeric representation of the signal values may be set to integer, float with configurable number of decimal places, hex, or binary as shown in Figure 60.

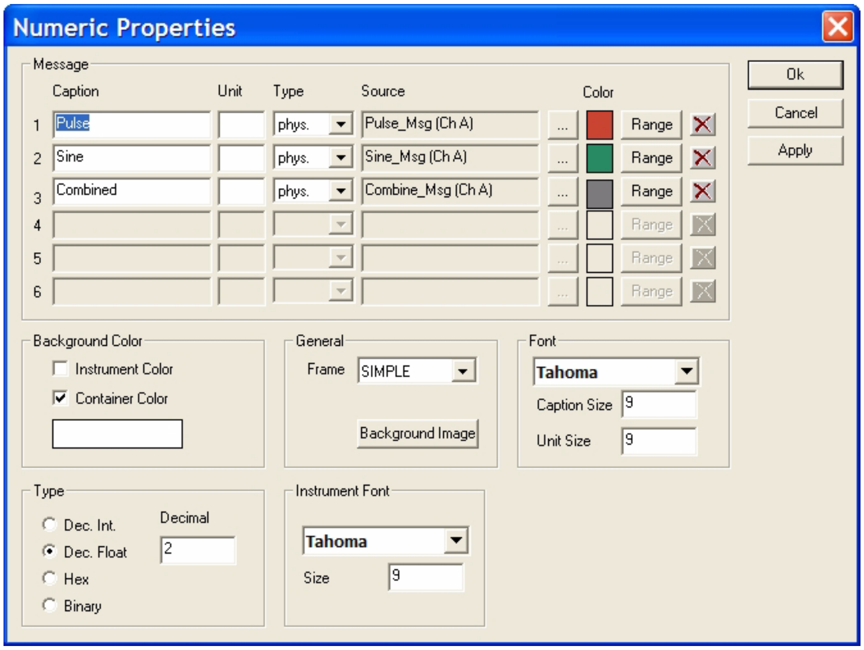


Figure 60 Numeric Properties Dialog

Gauge Control

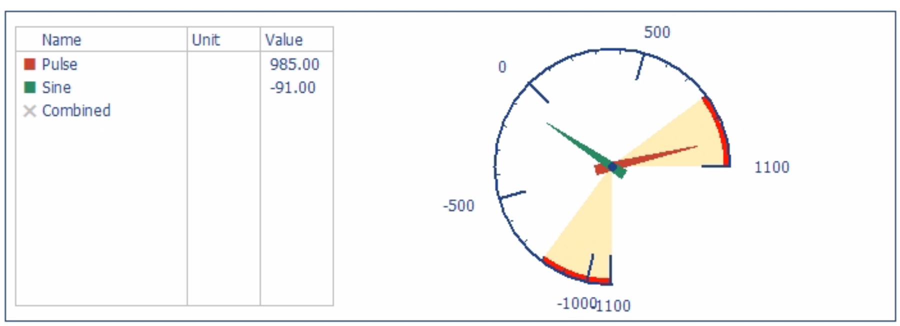


Figure 61 Gauge Control in instrument view

The *Gauge Control* shown in Figure 61 comprises a configurable gauge indicator in addition to the signal table contained in the *Numeric Control*. At most six signals may be represented at once by a *Gauge Control*. The start angle and the end angle of the gauge may be configured with the start angle applied clockwise and the end angle applied counter-clockwise.

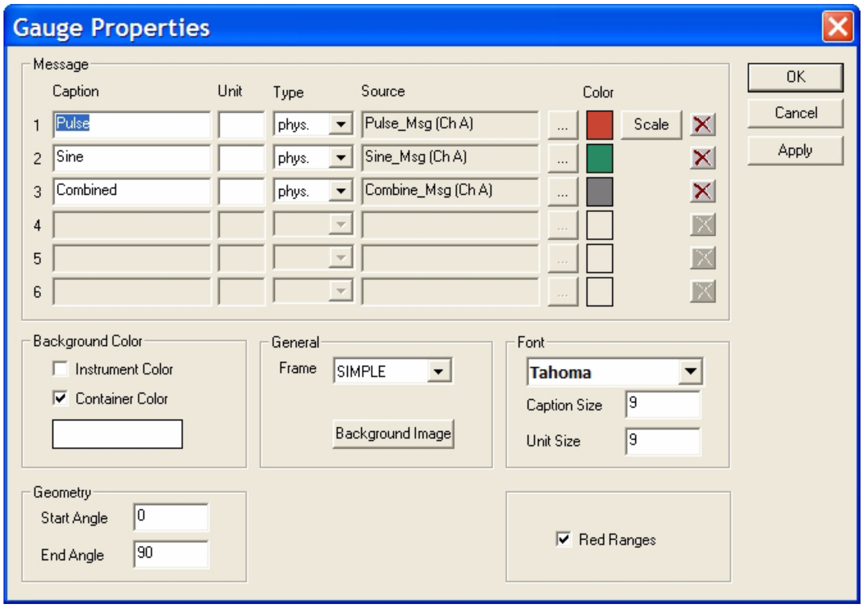


Figure 62 Gauge Properties Dialog

Bar Control

Figure 63 Bar Control in instrument view

The *Bar Control* shown in Figure 63 comprises a configurable bar indicator in addition to the signal table contained in the *Numeric Control*. At most six signals may be represented at once by a *Bar Control*. The orientation of the bar may be configured to be horizontal or vertical. The bar may be displayed as a solid bar indicator (called "Solid" in Figure 64) or as a needle (called "History" in Figure 64). Additionally, the ranges between start value and low limit and between high limit and end value are emphasized in the *Bar Control* if the check box *Red Ranges* is activated.

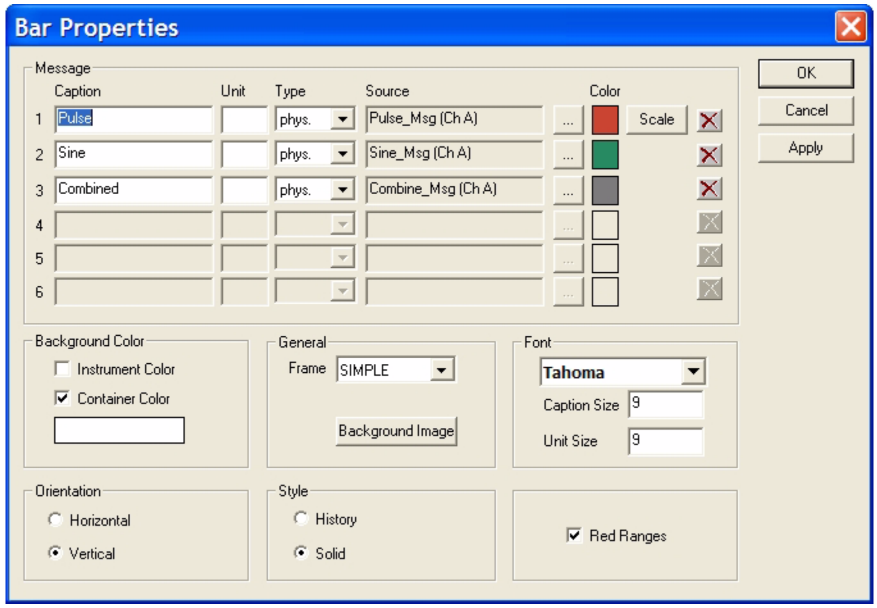


Figure 64 Bar Properties Dialog

Trace Control

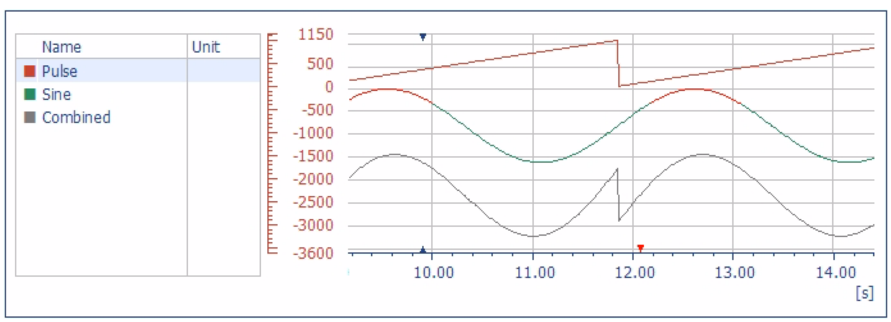


Figure 65 Trace Control in instrument view

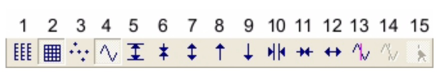


Figure 66 Trace Control Buttons

The *Trace Control* shown in Figure 65 comprises a trace indicator in addition to the signal table contained in the *Numeric Control*. At most six signals may be represented at once by a *Trace Control*. The selected signal is emphasized by a light-blue background in the signal table (see Figure 65). The current point in time is marked by two blue triangles in the trace indicator (one pointing up from the bottom line, one pointing down from the upper line) as shown in Figure 65. Trigger points are marked by red triangles pointing down on the bottom line. A series of control buttons as shown in Figure 66 are assigned to the *Trace Control*. Those buttons are:

- Toggle y axes (Button 1 in Figure 66)

  If pressed all y axes of the signals contained within the *Trace Control* are shown side by side next to the signal table with the y axis of the selected signal being the right-most. If unpressed only the y axis of the selected signal is shown next to the signal table.

- Toggle grid (Button 2 in Figure 66)

  If pressed a grid is displayed within the trace indicator. If unpressed no grid is displayed within the trace indicator.

- Toggle markers (Button 3 in Figure 66)

  If pressed all available signal values are emphasized by markers within the trace indicator. If unpressed no markers are displayed within the trace indicator.

- Toggle lines (Button 4 in Figure 66)

  If pressed available signal values are interpolated by lines on a per signal basis. If unpressed no interpolation is performed. Toggle lines and toggle markers may be pressed together but at least either toggle lines or toggle markers is active.

- Fit y scale (Button 5 in Figure 66)

  If pressed the range of values as displayed on the y axis (y range) for the selected signal is accommodated to the signal's range of values. The start value and the end value of the y range may be modified via *Scale* button in the *Trace Properties Dialog* as shown in Figure 67.

- Zoom in y scale (Button 6 in Figure 66)

  If clicked the selected signal is zoomed in on the y axis. The selected signal may also be zoomed by mouse interaction (move mouse pointer over the signal's y axis and use the scroll wheel to zoom in and out).

- Zoom out y scale (Button 7 in Figure 66)

  If clicked the selected signal is zoomed out on the y axis. The selected signal may also be zoomed by mouse interaction (move mouse pointer over the signal's y axis and use the scroll wheel to zoom in and out).

- Move up (Button 8 in Figure 66)

  If clicked the selected signal is moved up on the y axis. The selected signal may also be moved along the y axis by mouse interaction (move mouse pointer over the signal's y axis, press the left mouse button and move up and down).

- Move down (Button 9 in Figure 66)

  If clicked the selected signal is moved down on the y axis. The selected signal may also be moved along the y axis by mouse interaction (move mouse pointer over the signal's y axis, press the left mouse button and move up and down).

- Toggle zoom in mode (Button 10 in Figure 66)

  If pressed the mouse cursor may be used to select a time range to zoom into within the trace indicator via click and drag. If unpressed the mouse cursor may be used to click and drag the display in the trace indicator to the left or to the right on the x axis. The zoom in mode may only be active if no time measurement cursor is active (see description of Button 13 and Button 14 below).

- Zoom in time scale (Button 11 in Figure 66)

  If clicked the trace indicator zooms in on the x axis.

- Zoom out time scale (Button 12 in Figure 66)

  If clicked the trace indicator zooms out on the x axis.

- Toggle measure cursor (Button 13 in Figure 66)

  If pressed the time marker becomes active. The *Trace Control* shows the position of the time marker in seconds beneath the signal table (t = ...) and the corresponding signal value(s) in an additional column within the signal table (y column). The mouse may be used to modify the time marker by a click into the trace indicator. The time marker can also be moved around within the trace indicator when the left mouse button is pressed. If the toggle measure cursor is unpressed the time marker becomes inactive. The indication of the time marker position and its corresponding signal value(s) disappear. The time marker is shown as a solid vertical line in the trace indicator.

- Toggle difference cursor (Button 14 in Figure 66)

  If pressed the time difference marker becomes active. The *Trace Control* shows the position of the time difference marker in seconds beneath the signal table (dt = ...) and the corresponding difference of the signal value(s) between time marker and time difference marker in an additional column within the signal table (dy column). The mouse may be used to modify the time difference marker by a click into the trace indicator. The time difference marker can also be moved around within the trace indicator when the left mouse button is pressed. If the toggle difference cursor is unpressed the time difference marker becomes inactive. The indication of the time difference marker position and its corresponding signal value(s) disappear. The time difference marker may only be activated together with the time marker. The time difference marker is shown as dashed vertical line in the trace indicator.

- Toggle cursor set by mouse (Button 15 in Figure 66)

If both the time marker and the time difference marker are active, this button determines the marker to be altered by mouse interaction. If pressed the time difference marker may be modified. If unpressed the timer marker may be manipulated. Independent of the state of the toggle cursor button, a marker may be selected and moved by a direct click onto the marker in the trace indicator.

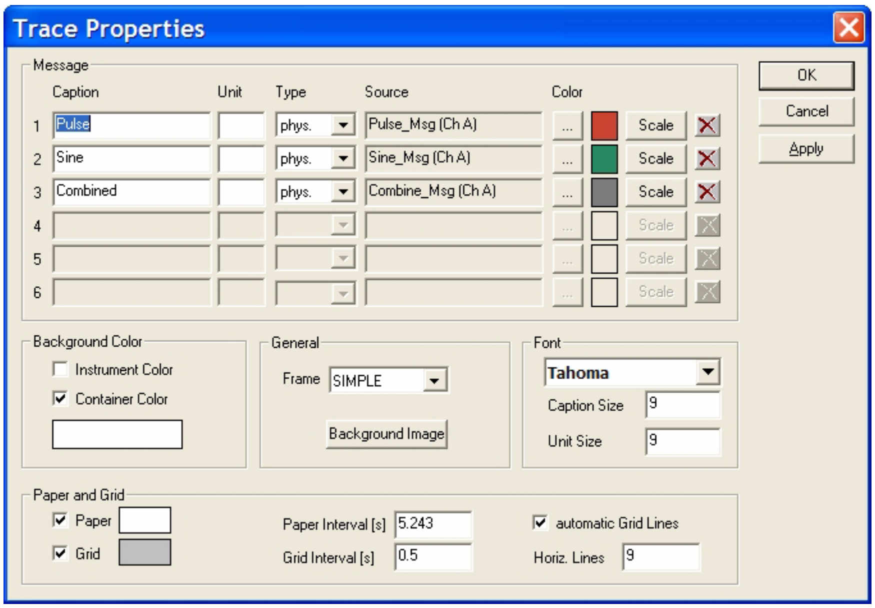


Figure 67 Trace Properties Dialog

COM Interface

The VPT1000 Hardware provides a COM interface (Microsoft Component Object Model) for following functionalities:

- Data transmission and reception with the *N5479A-001 Advanced Software option*
- Data transmission with the *Measurement Controller*
- Frame destruction

Those functions are provided by the control library of the VPT1000 Hardware on the PC (BDDRIVER). This library is used by the VPT1000 PC Software to control the VPT1000 Hardware. Additionally, this library acts as COM Server, which allows remote access to several functions (not to all functions) and to the E-Ray controllers included in the optional *N5479A-001 Advanced Software* option of the VPT1000. You can setup client software accessing the COM interface. Your client application may be written in Microsoft Visual Basic, Microsoft Visual C++ or in any other programing language that is supported by COM. Examples in Visual Basic and Visual C++ are delivered with the VPT1000 PC Software in the directory:

`<VPT1000_prog_dir>\COMSamples.`

COM Server Registration

To have the COM functions available it is necessary to register the COM server at *Microsoft Windows*. This procedure is handled automatically by the setup program of the VPT1000 PC Software.

If you are using visualization tools other than the VPT1000 PC Software you have to register the COM server manually to make the COM functions available. Use the following command to register BDDRIVER as COM server at Microsoft Windows:

1 Open a *CommandWindow* with *Start->Run*

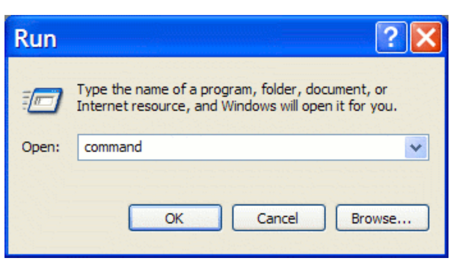


Figure 68 Open a Command Window

2 Use the following *Windows* command to register the COM server

regsvr32 <VPT1000_prog_dir>\bddriver.dll

COM API Description

The following functions are available via COM interface.

The COM interface is defined in BDDRIVER.idl, which is also part of the VPT1000 PC Software delivery:

```
<VPT1000_prog_dir>\COMFiles\BDDRIVER.idl
```

Extracts are taken from this file to describe the API in this section.

All the functions below return HRESULT, which can be one of the following:

- S_OK ... function successfully executed
- S_FALSE ... there was a minor problem when executing the function (equivalent to warning)
- E_FAIL ... execution of function failed – an exception is raised, which has to be handled by the calling function (see C++ example for details)

General Functions

The following functions are required to initialize and release the COM server, open and close COM sessions, to check the version and to connect to the VPT1000 Hardware.

COM_BD_InitCOM This function initializes the COM server. The function has to be called before using any other functions.

```
// COM_BD_InitCOM

[id(1), helpstring("method COM_BD_InitCOM")] HRESULT COM_BD_InitCOM();
```

COM_BD_ReleaseCOM This function releases the COM server. The function has to be called at the end of the COM server usage.

```
// COM_BD_ReleaseCOM

[id(2), helpstring("method COM_BD_ReleaseCOM")] HRESULT COM_BD_ReleaseCO
M();
```

COM_BD_InitSession This function initiates a COM session and provides a session handle which is required by other functions.

```
// COM_BD_InitSession

[id(3), helpstring("method COM_BD_InitSession")] HRESULT COM_BD_InitSess
ion(

        [out] int* pSessionHandle

        );
```

COM_BD_UninitSession This function closes a COM session.

```
// COM_BD_UninitSession

[id(4), helpstring("method COM_BD_UninitSession")] HRESULT COM_BD_Uninit
Session(

        [in] int nSessionHandle

        );
```

COM_BD_GetMainVersion This function provides the version of the driver running on the PC and is used to communicate with the VPT1000 Hardware.

```
// COM_BD_GetMainVersion

[id(5), helpstring("method COM_BD_GetMainVersion")] HRESULT COM_BD_GetMa
inVersion(

        [out]       int         *driverMajorVersion,

        [out]       int         *driverMinorVersion,

        [out]       int         *driverPatchVersion,

        [out]       BSTR         *driverVersionVerbose,

        [out]       BSTR         *driverPath

        );
```

COM_BD_Connect This function establishes a connection to the VPT1000 Hardware.

```
// COM_BD_Connect

[id(6), helpstring("method COM_BD_Connect")] HRESULT

COM_BD_Connect(

        [in]        int       nSessionHandle,

        [in]        BSTR      Address //String e.g.: "192.168.83.111"

        );
```

Transmission With Measurement Controller

In addition to the ability to send frames, the *Measurement Controller* has fault injection abilities, to destroy frames. This section describes the frame sending and frame destruction COM functions for the *Measurement Controller*. For information about frame transmission and reception via E-Ray communication controllers, see "Controlling E-Ray Controller(s)" on page 78.

Please note that for correct execution of the following functions the following prerequisites must be fulfilled:

- VPT1000 PC Software has to be connected to the VPT1000 Hardware
- Online monitoring has to be started in *Synchronous Mode*

COM_BD_SendFlexRayMessage, COM_BD_SendFlexRayMessageVT This function sends a FlexRay Frame via the *Measurement Controller*. Contrary to the send function for the E-Ray, the send function for the *Measurement Controller* also allows sending of frames with an invalid header CRC.

```
// COM_BD_SendFlexRayMessage

[id(7), helpstring("method COM_BD_SendFlexRayMessage")] HRESULT COM_BD_S
endFlexRayMessage(

        [in]    int    nSessionHandle,

        [in]    int    Reserved,

        [in]    int    PLPI,

        [in]    int    Null,

        [in]    int    Sync,

        [in]    int    SUP,

        [in]    int    FrameID,

        [in]    int    Len,

        [in]    int    CycleOffset,

        [in]    int    CycleDistance,

        [in]    int    bCalculateCorrectHCRC,

        [in]    int    HeaderCRC,

        [in]    unsigned char *pData,

        [in]    int    Channel,

        [in]    int    SendCounter

        );


// COM_BD_SendFlexRayMessageVT

[id(7), helpstring("method COM_BD_SendFlexRayMessageVT")] HRESULT COM_BD
_SendFlexRayMessageVT(

        [in]    int    nSessionHandle,

        [in]    int    Reserved,

        [in]    int    PLPI,

        [in]    int    Null,

        [in]    int    Sync,

        [in]    int    SUP,

        [in]    int    FrameID,

        [in]    int    Len,
```

```
[in]    int    CycleOffset,
[in]    int    CycleDistance,
[in]    int    bCalculateCorrectHCRC,
[in]    int    HeaderCRC,
[in]    VARIANT Data,
[in]    int    Channel,
[in]    int    SendCounter
);
```

The following input parameters must be provided for this function:

- nSessionHandle – Handle provided by the function COM_BD_InitSession
- Reserved – Reserved bit (0 or 1)
- PLPI – Payload preamble indicator (0 or 1, 1 = frame with message ID in dynamic segment or network management vector in static segment)
- Null – Null frame indicator (0 or 1, 0 = null frame)
- Sync – Sync frame indicator (0 or 1, 1 = sync frame)
- SUP – Startup frame indicator (0 or 1, 1 = startup frame)
- FrameID – FlexRay Frame Identifier (1 – 2047)
- Len – Payload length in two-byte words
- CycleOffset – Base cycle (0 – 63)
- CycleDistance – Cycle repetition (1 – 64)
- bCalculateCorrectHCRC – 1 means, that the parameter HeaderCRC is ignored and the header CRC is correctly calculated by the send function.
- HeaderCRC – Header CRC
- pData – Pointer to payload data (COM_BD_SendFlexRayMessage only)
- Data – Pointer to a VARIANT object of the payload data (COM_BD_SendFlexRayMessageVT only)
- Channel – FlexRay channel (1 = A, 2 = B, 3 = A and B)
- SendCounter – Integer number determining how often the frame transmission is repeated

COM_BD_ConfigureFrameDestruction This function allows configuring the disturbance of a FlexRay Channel, starting at a configurable offset, measured from the slot boundary of a configurable slot. Furthermore the length of the disturbance is configurable.

```
// COM_BD_ConfigureFrameDestruction

[id(8), helpstring("method COM_BD_ConfigureFrameDestruction")] HRESULT C
OM_BD_ConfigureFrameDestruction(
```

```
        [in]    int    nSessionHandle,
        [in]    int    chChannel,
        [in]    int    wdOffset,
        [in]    int    wdLength,
        [in]    int    wdSlotID,
        [in]    int    chBase,
        [in]    int    chRepetition
        );
```

The following input parameters must be provided for this function:

- nSessionHandle – Handle provided by the function COM_BD_InitSession
- chChannel – FlexRay Channel (1 = A, 2 = B, 3 = A and B)
- wdOffset – Offset from the slot boundary in microticks
- wdLength – Length of the disturbance in microticks
- wdSlotID – Slot to be disturbed
- chBase – Base cycle
- chRepetition – Cycle repetition

COM_BD_StartFrameDestruction This function starts the slot disturbance and repeats the specified disturbance for a configurable number of times.

```
// COM_BD_StartFrameDestruction

[id(9), helpstring("method COM_BD_StartFrameDestruction")] HRESULT COM_B
D_StartFrameDestruction(

        [in]    int    nSessionHandle,
        [in]    int    chChannel,
        [in]    int    chTimes
        );
```

The following input parameters must be provided for this function:

- nSessionHandle – Handle provided by the function COM_BD_InitSession
- chChannel – FlexRay Channel (1 = A, 2 = B, 3 = A and B)
- chTimes – Number of repetitions (0 = Stop, 1 – 254 = number of repetitions, 255 = unlimited number of repetitions)

Controlling E-Ray Controller(s)

The following API functions are only available, if the VPT1000 Hardware is equipped with the *N5479A-001* option providing E-Ray communication controllers. The API functions for controlling the E-Ray controllers run independent from VPT1000 PC Software. Hence contrary to the send and fault injection functions for the *Measurement Controller*, they don't need VPT1000 PC Software running in parallel.

With the *N5479A-001 Advanced Software option* the VPT1000 Hardware can be configured as coldstart node(s) and can serve as a startup buddy.

COM_BD_ConfigureController This function configures the E-Ray controller specified by the controller index. The first controller index is 0. In addition to the session handle, which is provided by the function COM_BD_InitSession, the function COM_BD_ConfigureController requires several FlexRay protocol parameters which must be configured according to the FlexRay Protocol Specification.

```
// COM_BD_ConfigureController

[id(10), helpstring("method COM_BD_ConfigureController")] HRESULT COM_BD
_ConfigureController(

         [in]     int     SessionHandle,
         [in]     int     nCtrlIDX,
         [in]     float gdBit,
         [in]     int     gdTSSTransmitter,
         [in]     int     pDelayCompensationA,
         [in]     int     pDelayCompensationB,
         [in]     int     gdCycle,
         [in]     float gdMacrotick,
         [in]     int     gdStaticSlot,
         [in]     int     gNumberOfStaticSlots,
         [in]     int     gdActionPointOffset,
         [in]     int     gPayloadLengthStatic,
         [in]     int     gdDynamicSlotIdlePhase,
         [in]     int     gdMinislot,
         [in]     int     gdMinislotActionPointOffset,
         [in]     int     pLatestTx,
         [in]     int     gdNIT,
         [in]     int     gdSymbolWindow,
         [in]     int     gOffsetCorrectionStart,
         [in]     float gOffsetCorrectionMax,
         [in]     int     gPayloadLengthDynMax,
```

```
        [in]     int     pKeySlotID,
        [in]     int     KeySlotMode
        );
```

COM_BD_RunController This function sets the respective E-Ray controller into the FlexRay startup state. The E-Ray controller behaves according to the configuration, i.e. performs a coldstart or integrates in an already running cluster.

```
// COM_BD_RunController
[id(16), helpstring("method COM_BD_RunController")] HRESULT COM_BD_RunCo
ntroller(
        [in]     int    SessionHandle,
        [in]     int    nCtrlIDX
        );
```

The following input parameters must be provided for this function:

- SessionHandle – Handle provided by the function COM_BD_InitSession
- nCtrlIDX – Controller index of the E-Ray communication controller

COM_BD_HaltController This function halts the respective E-Ray communication controller.

```
// COM_BD_HaltController
[id(17), helpstring("method COM_BD_HaltController")] HRESULT COM_BD_Halt
Controller(
        [in]     int    SessionHandle,
        [in]     int    nCtrlIDX
        );
```

The following input parameters must be provided for this function:

- SessionHandle – Handle provided by the function COM_BD_InitSession
- nCtrlIDX – Controller index of the E-Ray communication controller

COM_BD_GetPOCState This function returns the POC state of the respective E-Ray communication controller.

```
// COM_BD_GetPOCState
[id(19), helpstring("method COM_BD_GetPOCState")] HRESULT COM_BD_GetPOCS
tate(
        [in]     int    SessionHandle,
        [in]     int    nCtrlIDX,
        [out]    int    *pState
```

```
        );
```

The following input parameters must be provided for this function:

- SessionHandle – Handle provided by the function COM_BD_InitSession
- nCtrlIDX – Controller index of the E-Ray communication controller

The following output parameter is returned by this function:

- *pState – Communication controller POC state
  - **a** 0 ... POCStateConfig
  - **b** 1 ... POCStateDefaultConfig
  - **c** 2 ... POCStateHalt
  - **d** 3 ... POCStateNormalActive
  - **e** 4 ... POCStateNormalPassive
  - **f** 5 ... POCStateReady
  - **g** 6 ... POCStateStartup
  - **h** 7 ... POCStateWakeup

COM_BD_RequestBuffer The request buffer function requests a buffer for reception or transmission and provides a handle to this buffer, if successful.

```
// COM_BD_RequestBuffer

[id(12), helpstring("method COM_BD_RequestBuffer")] HRESULT COM_BD_Reque
stBuffer(

        [in]      int     SessionHandle,

        [in]      int     nCtrlIDX,

        [in]      int     nBufferType,

        [out]     int     *nBufferHandle

        );
```

The following input parameters must be provided for this function:

- SessionHandle – Handle provided by the function COM_BD_InitSession
- nCtrlIDX – Controller index of the E-Ray communication controller
- nBufferType - Buffer of the following type
  - **a** 0 ... KeyBuffer – 1 key buffer available
  - **b** 1 ... StaticBuffer – 14 static buffers available
  - **c** 2 ... DynamicBuffer – The number of available dynamic buffers is 15 for a payload length of 254 byte and can be calculated using the following formula:

    number of dynamic buffers =

min(113; floor(1028 / (4 + ceil(pPayloadLengthDynMax[Byte] / 4)))
)

The following output parameter is returned by this function:

- *nBufferHandle – Handle of the requested buffer

COM_BD_ReleaseBuffer The release buffer function releases a receive or transmit buffer.

```
// COM_BD_ReleaseBuffer

[id(13), helpstring("method COM_BD_ReleaseBuffer")] HRESULT COM_BD_Relea
seBuffer(

         [in]     int    SessionHandle,

         [in]     int    nCtrlIDX,

         [in]     int    nBufferHandle

         );
```

The following input parameters must be provided for this function:

- SessionHandle – Handle provided by the function COM_BD_InitSession
- nCtrlIDX – Controller index of the E-Ray communication controller
- nBufferHandle – Handle of the buffer to be released

COM_BD_Configure_Buffer This function configures a buffer previously requested with the function COM_BD_RequestBuffer.

```
// COM_BD_Configure_Buffer

[id(11), helpstring("method COM_BD_ConfigureBuffer")] HRESULT COM_BD_Con
figureBuffer(

         [in]     int    SessionHandle,

         [in]     int    nCtrlIDX,

         [in]     int    nBufferHandle,

         [in]     int    nTransferDirection,

         [in]     int    nFrameID,

         [in]     int    nChannel,

         [in]     int    nRepCycle,

         [in]     int    nBaseCycle,

         [in]     int    nTransmissionMode

         );
```

The following input parameters must be provided for this function:

- SessionHandle – Handle provided by the function COM_BD_InitSession

- nCtrlIDX – Controller index of the E-Ray communication controller
- nBufferHandle – Handle of the buffer to be configured
- nTransferDirection – Transfer direction
  - **a** 0 ... Reception
  - **b** 1 ... Transmission
- nFrameID - FlexRay frame identifier (1 – 2047)
- nChannel – FlexRay channel (1 = A, 2 = B, 3 = A and B)
- nRepCycle – Cycle repetition
- nBaseCycle – Base cycle
- nTransmissionMode – Transmission mode
  - **a** 0 ... If a buffer configured for a static slot is not updated after transmission, a null frame is transmitted in that static slot during subsequent cycles. If a buffer configured for a dynamic slot is not updated after transmission, nothing is transmitted in that dynamic slot during subsequent cycles.
  - **b** 1 ... If a buffer configured for a certain slot (static or dynamic) is not updated after transmission, the frame is transmitted with previous data.

COM_BD_PutBuffer, COM_BD_PutBufferVT This functions allows filling a buffer with data and transmition of buffer content. For script languages character array cannot be handled, therefore the function is available in two versions, one with data as a character array and one with data as a VARIANT object.

```
// COM_BD_PutBuffer

[id(14), helpstring("method COM_BD_PutBuffer")] HRESULT

COM_BD_PutBuffer(

         [in]     int    SessionHandle,

         [in]     int    nCtrlIDX,

         [in]     int    nBufferHandle,

         [in]     unsigned char *pData,

         [in]     int    nLen

         );

// COM_BD_PutBufferVT

[id(20), helpstring("method COM_BD_PutBufferVT")] HRESULT

COM_BD_PutBufferVT(

         [in]     int    SessionHandle,

         [in]     int    nCtrlIDX,
```

```
        [in]    int    nBufferHandle,
        [in]    VARIANT Data,
        [in]    int    nLen
        );
```

The following input parameters must be provided for this function:

- SessionHandle – Handle, which is provided by the init session function
- nCtrlIDX – Controller index of the E-Ray communication controller
- nBufferHandle – Handle of the transmit buffer
- *pData – Pointer to a character array of the transmit data (COM_BD_GetBuffer only)
- Data – Pointer to a VARIANT object of the transmit data (COM_BD_GetBufferVT only)
- nLen – Number of bytes to be transmitted

COM_BD_GetBuffer, COM_BD_GetBufferVT This functions allows reception of data from one of the previously configured receive buffers. For script languages a character array cannot be handled, therefore the function is available in two versions, one with data as a character array and one with data as a VARIANT object.

```
// COM_BD_GetBuffer
[id(15), helpstring("method COM_BD_GetBuffer")] HRESULT
COM_BD_GetBuffer(
        [in]    int    SessionHandle,
        [in]    int    nCtrlIDX,
        [in]    int    nBufferHandle,
        [out]   unsigned char *pData,
        [in]    int    nLen,
        [out]   int    *pnReceivedLength
        );


// COM_BD_GetBufferVT
[id(21), helpstring("method COM_BD_GetBufferVT")] HRESULT
COM_BD_GetBufferVT(
        [in]    int    SessionHandle,
        [in]    int    nCtrlIDX,
        [in]    int    nBufferHandle,
```

```
        [out]    VARIANT *Data,
        [out]    int     *pnReceivedLength
        );
```

The following input parameters must be provided for this function:

- SessionHandle – Handle provided by the function COM_BD_InitSession
- nCtrlIDX – Controller index of the E-Ray communication controller
- nBufferHandle – Handle of the receive buffer
- *pData – Pointer to a character array, where the receive data is to be transferred (COM_BD_GetBuffer only)
- *Data – Pointer to a VARIANT object, where the receive data is to be transferred (COM_BD_GetBufferVT only)
- nLen – Number of bytes to be transferred (COM_BD_GetBuffer only. This value is included in the argument *Data in the function COM_BD_GetBufferVT).
- *pnReceivedLength - Received payload data length in bytes

COM_BD_CleanupController This function resets the E-Ray communication controller to the default configuration. Furthermore it releases all buffers.

```
// COM_BD_CleanupController

[id(18), helpstring("method COM_BD_CleanupController")] HRESULT COM_BD_C
leanupController(

        [in]          int     SessionHandle,

        [in]          int     nCtrlIDX

        );
```

The following input parameters must be provided for this function:

- SessionHandle – Handle provided by the function COM_BD_InitSession
- nCtrlIDX – Controller index of the E-Ray communication controller

4
Working With VPT1000 PC Software

Online Monitoring

This section describes online monitoring, i.e., the monitoring of FlexRay, CAN and ADIO with the VPT1000 Hardware connected to the VPT1000 PC Software.

Before starting online monitoring perform the following tasks:

- Launch the VPT1000 PC Software
- Set up a project as described in "Getting Started" on page 42
- For observation of signals in addition to raw data, set up a *instrument view* with controls as described in "Instrument View" on page 58
- Verify network settings and connect to the VPT1000 Hardware.

Online monitoring can be controlled with the *Control Panel* shown in Figure 69.

Figure 69 Control Panel

Online Mode can be activated by pressing the *Connect/Disconnect Button* (Button 1 in Figure 69). After connecting to the VPT1000 Hardware online monitoring can be started and stopped with the *Play Forward/Stop Button* (Button 8 in Figure 69) or the keys *F9* and *ESC*. The monitored data packet representation may be switched from fixed view to chronological view and vice versa with the *Fixed/Chronological Button* (Button 20 in Figure 69).

Online Mode can be deactivated by pressing the *Connect/Disconnect Button* (Button 1 in Figure 69).

Alternatively, activation/deactivation of *Online Mode* and start/stop of online monitoring may also be controlled via the *Measurement Menu.*

There are several settings that can be modified to configure the online monitoring.

Monitoring Settings

The settings for monitoring and logging can be entered in the menu item *Measurement->Settings*. In addition to the general settings already described in "Getting Started" on page 42, there are settings concerning FlexRay, CAN, analog/digital input, Triggerout Pins, and Agilent option.

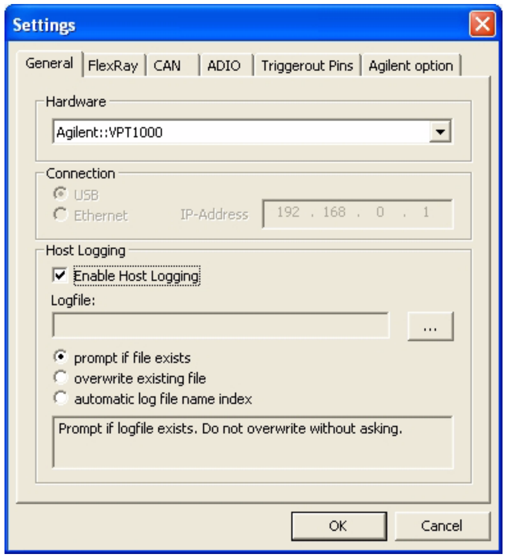


Figure 70 General Settings Dialog

For simple online monitoring keep the *Host Logging* check box unchecked. For measurement data recording check the check box (see "Host Logging" on page 103).

FlexRay Settings

To modify the FlexRay settings click on the *FlexRay Tab* in the *SettingsDialog* (Figure 71).

VPT1000 Hardware The VPT1000 Hardware supports monitoring FlexRay data in *asynchronous only mode* or in *mixed mode*.

- Asynchronous Only Mode: *Asynchronous Only Mode* monitors FlexRay data without trying to synchronize its timebase to the cluster time. Therefore no slot information is available in this mode. No timing checks are performed on the received data. See "FlexRay data packets" on page 110 for more detailed information on slot information and timing checks for FlexRay data. Asynchronous only mode only needs a subset of FlexRay parameters for monitoring data (baud rate and connected physical layer).Please note that frame transmission, fault injection and several features of *Triggerout Pins* are not available in this mode.

- *Mixed mode: Mixed Mode* starts monitoring FlexRay data immediately without slot information and timing checks and tries to synchronize its timebase to the cluster. After synchronization is established, slot information and timing check data are added to the received data. Furthermore, frame transmission (see "Frame Transmission/Reception With E-Ray Controller(s)" on page 106), fault injection (see "Fault Injection" on page 106) and several features of *Triggering Pins* (see "Triggerout Pin Settings" on page 96) are available after synchronization. If synchronization is lost, monitoring is continued loss-free in asynchronous mode and the VPT1000 Hardware tries to re-synchronize. *Mixed Mode* only works if the full FlexRay parameters are configured correctly.

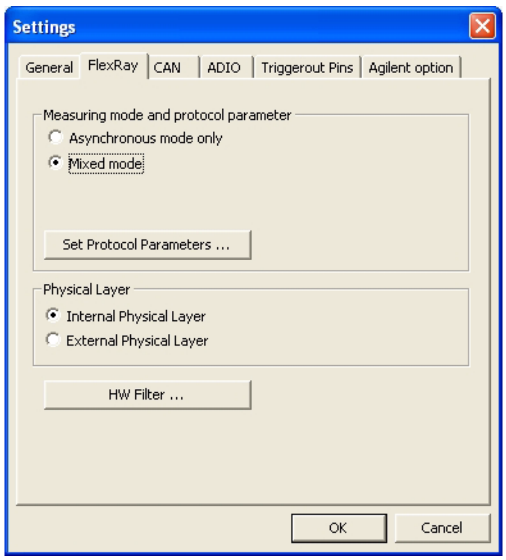


Figure 71 FlexRay Settings Dialog

FlexRay protocol parameters The FlexRay protocol parameters can be modified in the dialog that pops up when clicking on *Set Protocol Parameters...* in the *FlexRay Settings Dialog*. The initial parameter values are taken from the database file selected during project creation.

A different database file may be opened with the *Import Button* to import protocol parameters from.

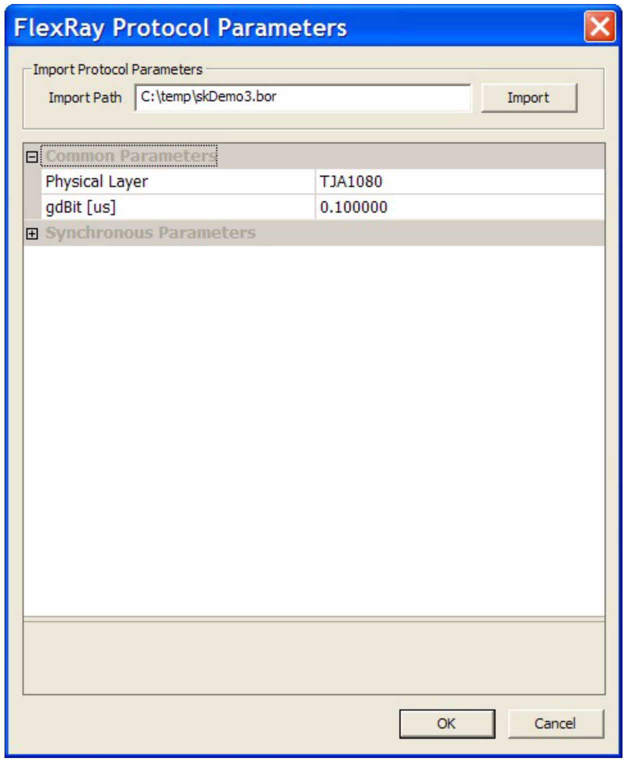


Figure 72 FlexRay Protocol Parameters Dialog - Asynchronous (Only) Mode

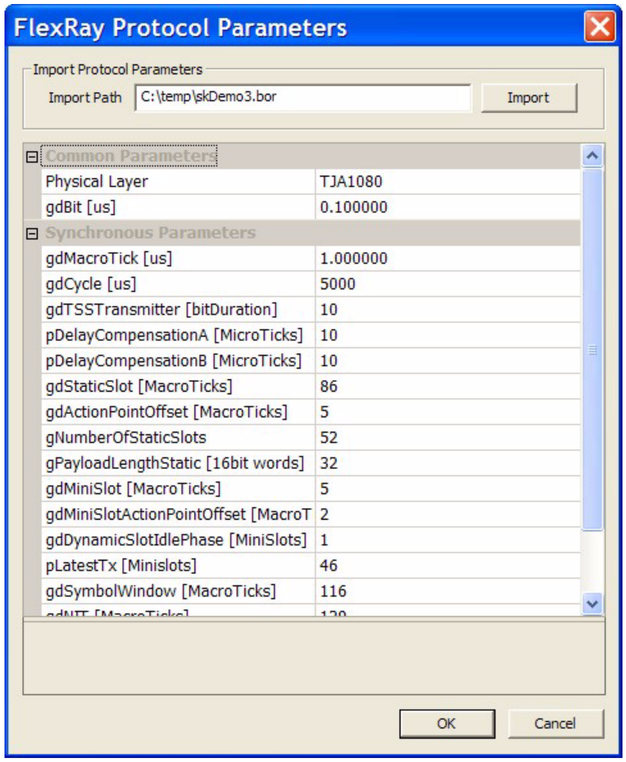


Figure 73 FlexRay Protocol Parameters Dialog – Mixed / Synchronous Mode

The parameters dialog depends on the selected measuring mode (compare Figure 72 and Figure 73). The parameter values have to be in accordance with the FlexRay Protocol Specification.

Incorrect cluster parameters can cause incorrect measurements!

Filter settings The VPT1000 PC Software allows configuration filters for FlexRay. The button *HW Filter...* in the *FlexRay Settings Dialog* opens the *FlexRay Filters Dialog* as shown in Figure 74. In this dialog the FlexRay filters may be set up. Since filtering is performed in the VPT1000 Hardware, the filter reduces the amount of data transferred from the VPT1000 Hardware to the VPT1000 PC Software. There are two alternatives filters available

- Acceptance filter: Frames meeting the filter criteria are accepted and transferred to the VPT1000 PC Software. All other frames are rejected by the VPT1000 Hardware.
- Rejection filter: Frames meeting filter criteria are rejected by the VPT1000 Hardware. All other frames are transferred to the VPT1000 PC Software.

It is not possible to combine the acceptance filter and the rejection filter.

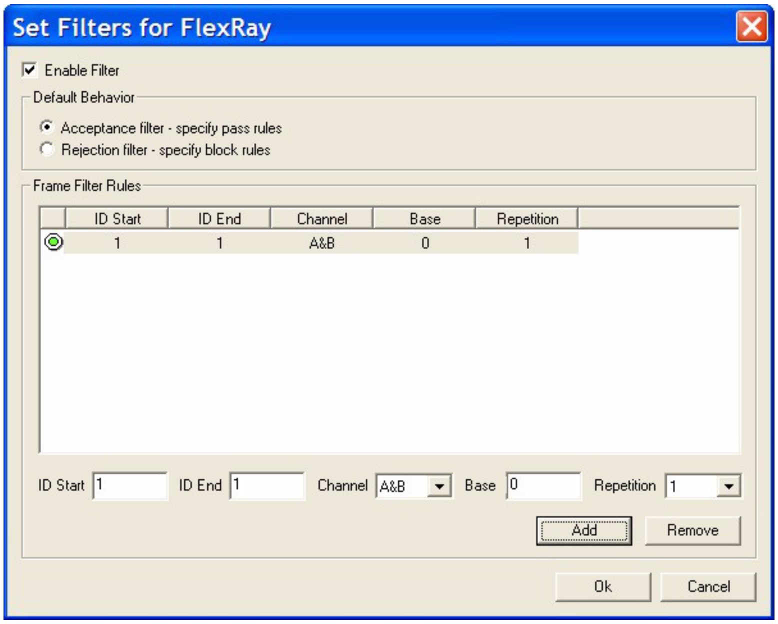


Figure 74 FlexRay Filters Dialog

A FlexRay filter rule comprises an ID range, the receive channel, a cycle base and a cycle repetition for acceptance filtering and an ID range and the receive channel for rejection filtering.

Only the last filter rule relevant for a certain frame applies. The following example will explain this concept.

Example: An acceptance filter is defined by the following three filter rules:

- ID Start = 2, ID End = 2, Channel = B, Base = 12, Repetition = 64
- ID Start = 4, ID End = 5, Channel = A&B, Base = 4, Repetition = 32
- ID Start = 5, ID End = 6, Channel = A, Base = 0, Repetition = 1

The frame passing this acceptance filter are:

- frame with ID 2 on Channel B in cycle 12
- frame with ID 4 on Channels A&B in cycles 4 and 36
- frame with ID 5 on Channel A in all cycles
- frame with ID 6 on Channel A in all cycles

Interchanging the first and the last filter rule causes the following frames to pass the acceptance filter:

- frame with ID 2 on Channel B in cycle 12
- frame with ID 4 on Channels A&B in cycles 4 and 36
- frame with ID 5 on Channels A&B in cycles 4 and 36

- frame with ID 6 on Channel A in all cycles

Clicking *Add* adds a filter rule to the list of filter rules. Clicking *Remove* deletes a filter rule from the filter rule list.

CAN Settings

The *CAN Tab* allows modifying the CAN settings (Figure 75).

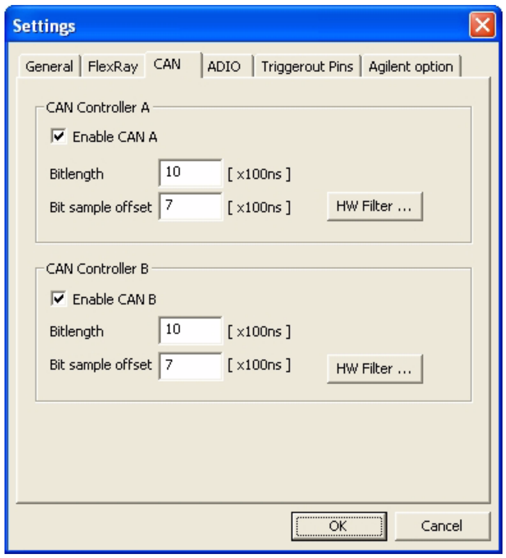


Figure 75 CAN Settings Dialog

It is possible to enable the two CAN controllers separately. Furthermore the settings for the bit timing (Bit length and Bit sample offset) can be modified in this dialog. The bit length minimum is 1000 ns resulting a baud rate of 1 Mbps.

CAN acceptance and rejection filtering may be configured in the *CAN Filters Dialog* as shown in Figure 76. A CAN filter rule comprises an ID range for acceptance and rejection filtering.

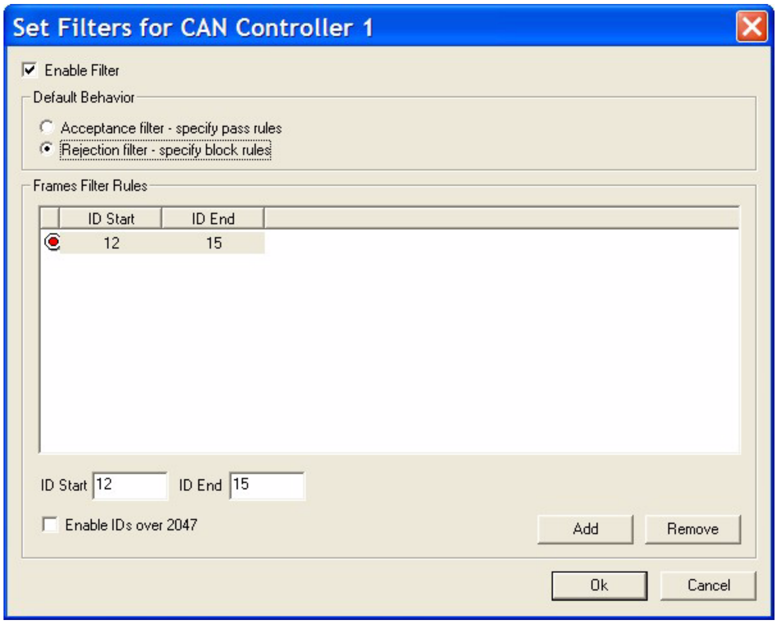


Figure 76 CAN Filters Dialog

Analog/Digital IO Settings

The *ADIO Tab* allows modifying the analog/digital IO settings (see Figure 77).

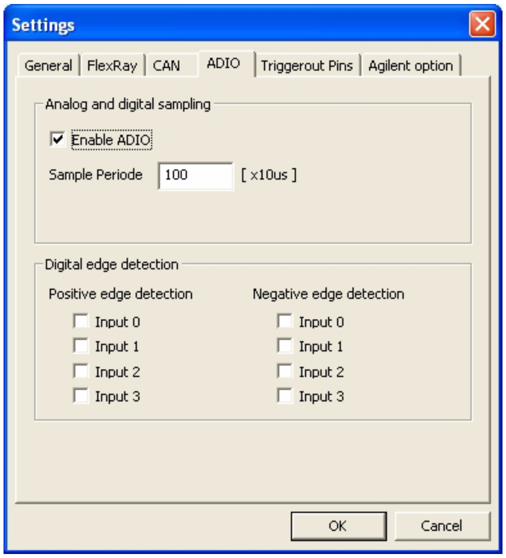


Figure 77 ADIO Settings Dialog

The analog and digital sampling is enabled and disabled and the sampling period is configured in multiples of 10 µs in this dialog. The minimum sampling period is 100 µs. The sample period is applied to all available analog and digital inputs.

Triggerout Pin Settings

When the VPT1000 FlexRay module has been configured with an Agilent MSO6000/7000 scope, FlexRay triggering and decoding are available. However, the VPT1000 Hardware also provides four *Triggerout Pins*, which can be configured individually also, to provide a FlexRay-related trigger source for other oscilloscopes or measurement hardware. These *Triggerout Pins* are available at the *Digital/Analog Connector* of the VPT1000 Hardware housing.

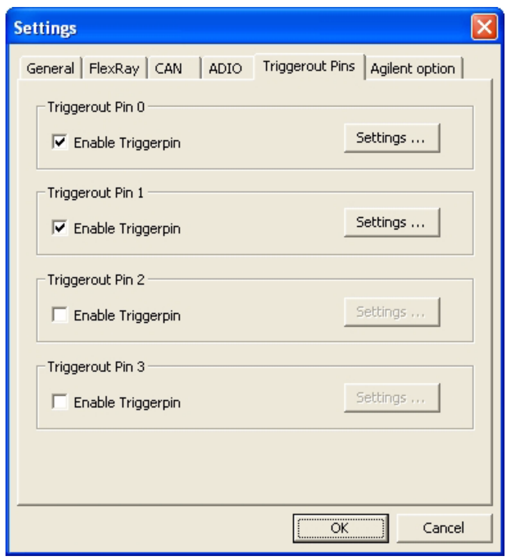


Figure 78 Triggerout Pins Settings Dialog

Use the check boxes *Enable Triggerpin* to enable the corresponding *Triggerout Pins*. Use the appropriate button *Settings...* to open the *Triggerout Pin Properties Dialog* shown in Figure 79.

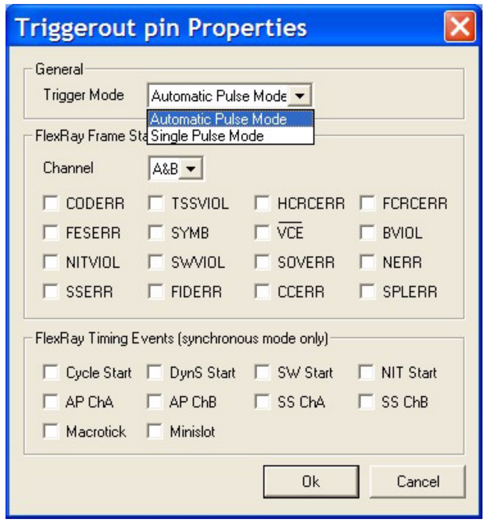


Figure 79 Triggerout Pin Properties Dialog

In this dialog the general behavior of *Triggerout Pins* may be configured.

- *Automatic Pulse Mode* generates a 100 ns wide high pulse on the *Triggerout Pin* each time the trigger event occurs as shown in Figure 81.
- *Single Pulse Mode* generates a positive edge on the *Triggerout Pin* when the trigger event occurs. Subsequent trigger events are not visible on the *Triggerout Pin* until the user reactivates the trigger. You can reactivate a trigger within the VPT1000 PC Software in the *ToolsMenu* by choosing *Reactivate Trigger Pin n*, or with the *Reactivate Trigger n Buttons* on the *Control Panel* (Buttons 16 – 19 in Figure 80). After reactivation the *Triggerout Pin* changes from high to low (see Figure 82).

Figure 80 Triggerout Pin Properties Dialog

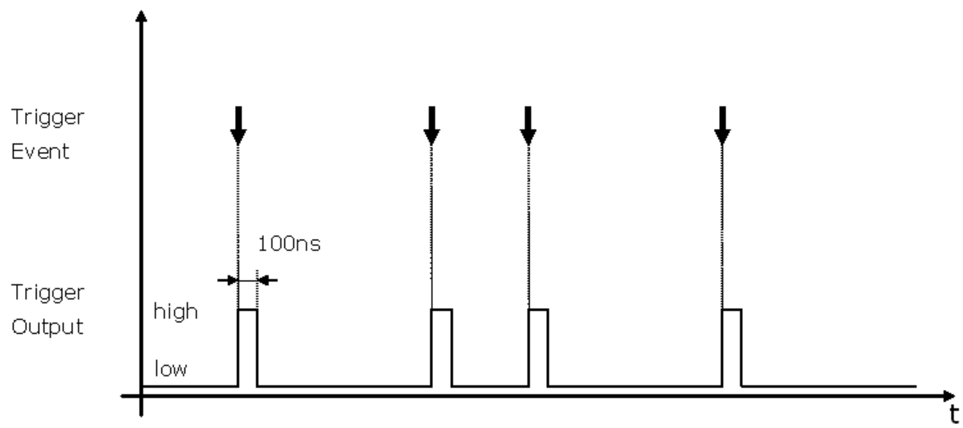


Figure 81 Automatic Pulse Mode

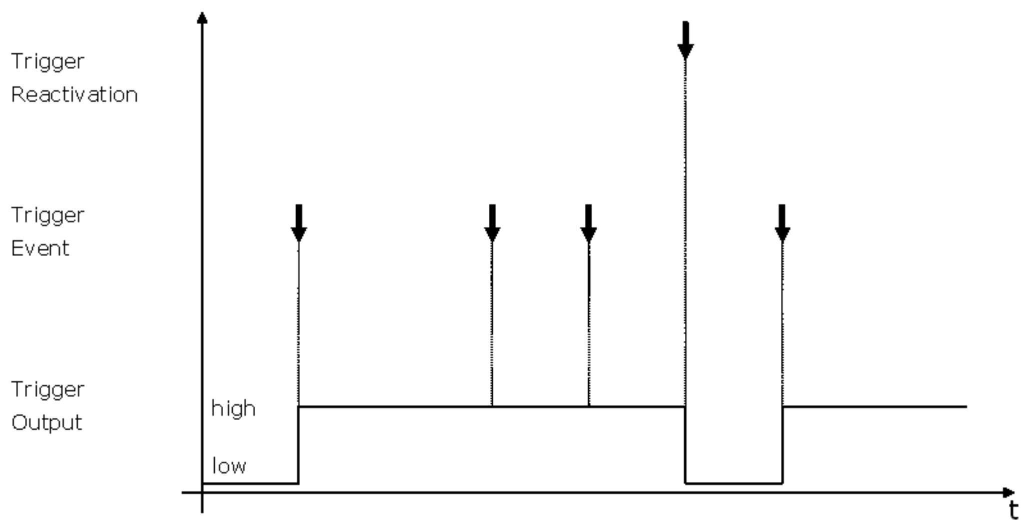


Figure 82 Single Pulse Mode

The group *FlexRay Status Events* allows the configuration of a FlexRay status-related trigger. For details about the FlexRay status see "FlexRay status bits" on page 112.

The group *FlexRay Timing Events* allows the configuration of a FlexRay timing-related trigger. The following FlexRay timing-related events are available:

- Cycle Start: Start of FlexRay cycle
- DynS Start: Start of Dynamic Segment
- SW Start: Start of Symbol Window

- NIT Start: Start of Network Idle Time
- AP ChA: Action point on FlexRay Channel A
- AP ChB: Action point on FlexRay Channel B
- SS ChA: Slot start on FlexRay Channel A
- SS ChB: Slot start on FlexRay Channel B
- Macrotick: Start of Macrotick
- Minislot: Start of Minislot

Providing this detailed FlexRay status and FlexRay timing-related events is a unique feature of the VPT1000 Hardware. The VPT1000 Hardware uses a dedicated FlexRay communication controller for that purpose enabling the VPT1000 Hardware to provide more detailed FlexRay status and FlexRay timing information than standard FlexRay controllers do.

MSO option settings

The VPT1000 Hardware can optionally be used together with an Agilent MSO6000/7000 series oscilloscope. In this case the VPT1000 Hardware decodes the FlexRay protocol information and hands it over to the Agilent MSO6000/7000 scope where the information is displayed time-correlated with the FlexRay analog/physical layer waveform. Figure 83 shows an measurement with an Agilent MSO6000 series oscilloscope.

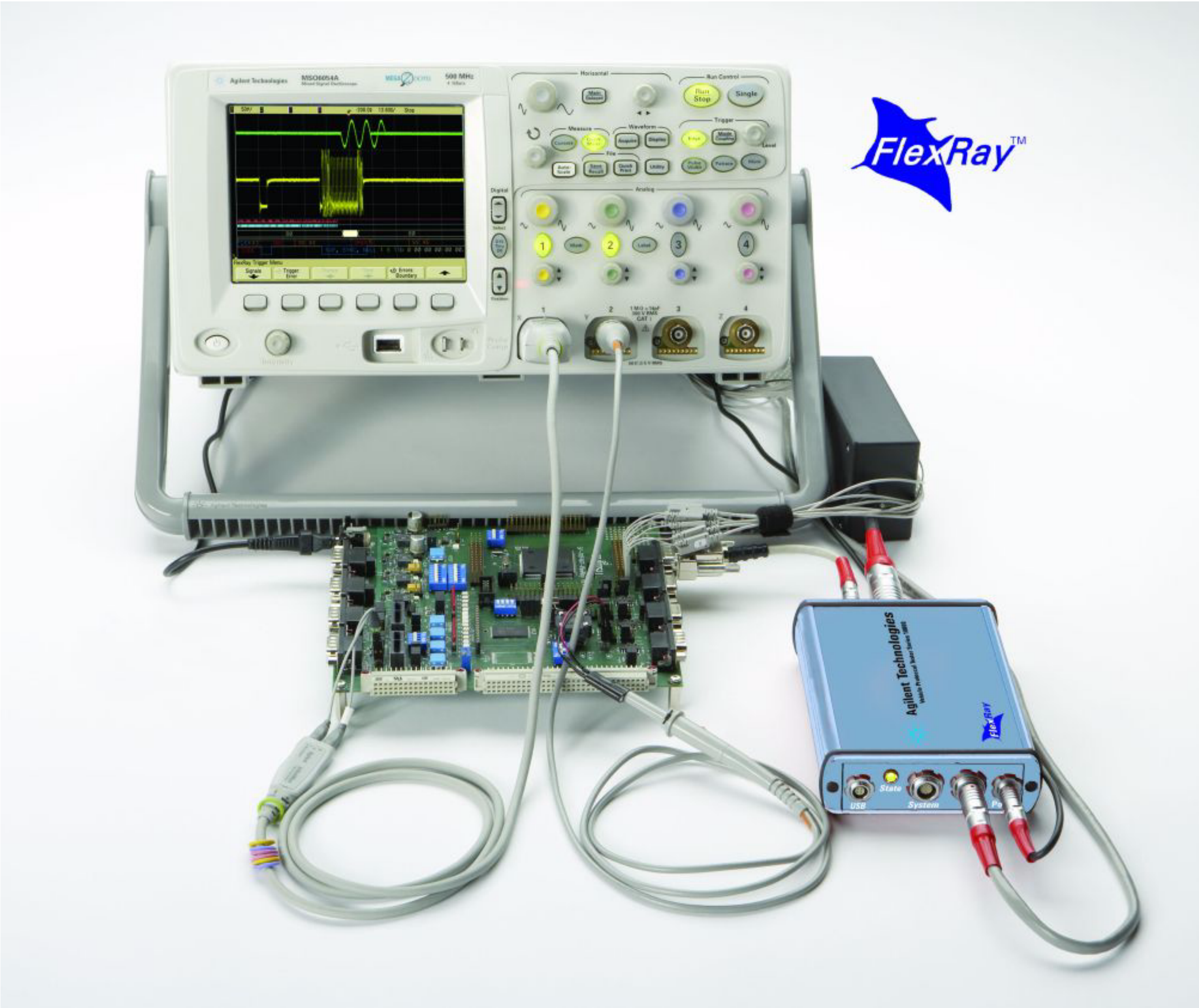


Figure 83 VPT1000 Hardware with Agilent MSO6000 Series oscilloscope

There are two modes available to control the VPT1000 Hardware together with the Agilent MSO6000 series oscilloscope:

- The VPT1000 Hardware is controlled by the scope. In this case a simultaneously measurement with the VPT1000 PC Software is not possible.
- The VPT1000 Hardware is controlled by the VPT1000 PC Software. This mode allows simultaneous measurement of the scope and the VPT1000 PC Software with only one VPT1000 Hardware module.

For configuring the VPT1000 Hardware for simultaneous measurement of the VPT1000 PC Software and the scope, use the *Agilent option Tab* shown in Figure 84.

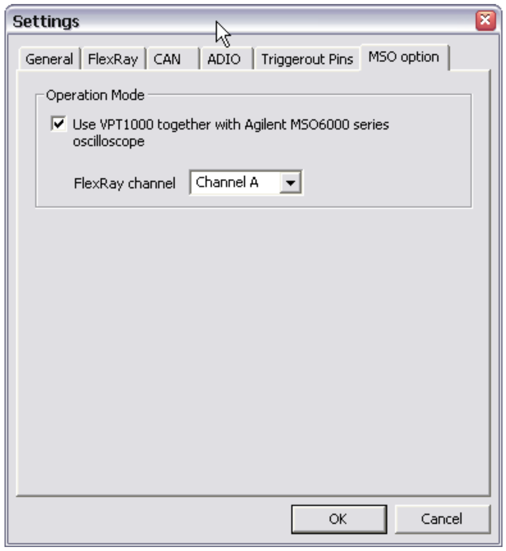


Figure 84 Agilent option Settings Dialog

The check box *Use VPT1000 PC Software together with Agilent MSO6000 series oscilloscope* enables the VPT1000 Hardware to generate the decoding information to the scope while a measurement is running. In this mode the FlexRay channel, which provides the decoding information on the bus, has to be selected as well.

NOTE If oscilloscope mode is selected, *ADIO* and *Triggerout Pins* are not available.

Attention: Do not connect digital signals to the digital input lines of the VPT1000 Hardware while the oscilloscope mode is active. This can damage the VPT1000 Hardware, therefore the MSO settings will not be saved in the VPF file!

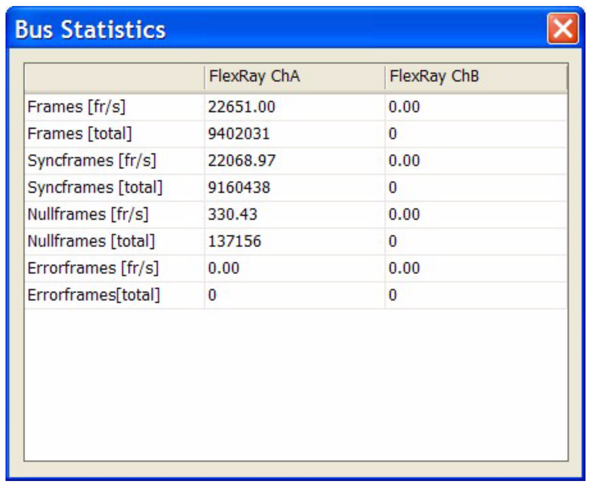
For more detailed information on using the VPT1000 Hardware together with the Agilent MSO6000/7000 series oscilloscope please refer to the InfiniiVision oscilloscope user's guide.

Bus Statistics

Bus Statistics provides accumulated information about the bus traffic on FlexRay channels:

NOTE Valid frames, null frames, sync frames, and startup frames exhibit a FlexRay status with VCE = 1 (valid communication element). Erroneous frames exhibit a FlexRay status with VCE = 0.

- Total number of valid frames (including null frames, sync frames and startup frames),
- Total number of sync frames (including startup frames),
- Total number of null frames,
- Total number of erroneous frames,
- Frame rates for the above categories of frames in frames per second.



	FlexRay ChA	FlexRay ChB
Frames [fr/s]	22651.00	0.00
Frames [total]	9402031	0
Syncframes [fr/s]	22068.97	0.00
Syncframes [total]	9160438	0
Nullframes [fr/s]	330.43	0.00
Nullframes [total]	137156	0
Errorframes [fr/s]	0.00	0.00
Errorframes[total]	0	0

Figure 85 Bus Statistics Window

The information is given for each FlexRay channel separately.

To open the *Bus Statistics Window* select *Bus Statistics* from the *ViewMenu*. *Bus Statistics* are available both in *Online* and *Offline Mode*. In *Online Mode*, the *Bus Statistics* are updated constantly; in *Offline Mode*, after loading a log file, the *Bus Statistics* are presented for the whole log data.

To reset *Bus Statistics* (i.e., to set all values back to 0), select *Clear Bus Statistics* from the context menu of the *Bus StatisticsWindow*. Note that this only makes sense in *Online Mode*. e.g., if you want to restart the statistics data collection after changing some parameters or before starting fault injection.

Host Logging

The VPT1000 Hardware together with the VPT1000 PC Software allows recording of FlexRay, CAN and ADIO packets into a log file on the PC's hard disk. Data logging is enabled with the check box *Enable Host Logging* in the *General Tab* of the *Settings Dialog*.

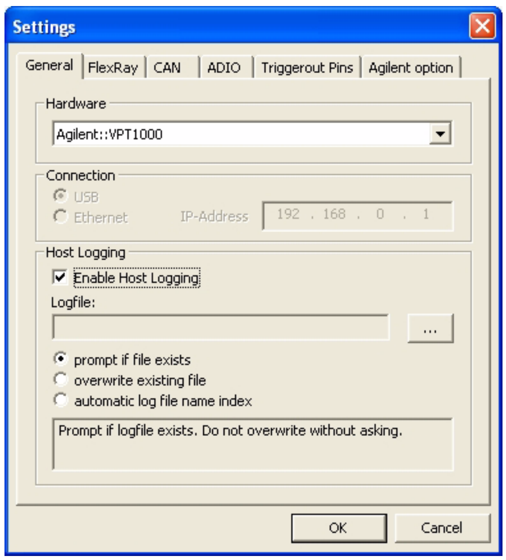


Figure 86 General Settings Dialog (Enable Host Logging)

Three alternative methods are available for the handling of existing log files:

- Prompt if file exists: If a log file exists the user is asked whether to overwrite that log file or to disable logging for the current measurement.
- Overwrite existing file: An existing log file is automatically overwritten without user interaction.
- Automatic log file name index: An existing log file remains unmodified. Every time a measurement is started an additional log file is created. The original file name is concatenated with an ascending index.

Host Logging is performed simultaneously to online monitoring and uses the online monitoring settings.

Offline Analysis

VPT1000 PC Software allows the analysis of log files recorded during *Host Logging* (see "Host Logging" on page 103). This function is called offline analysis.

The offline analysis is controlled with the control panel as shown in Figure 87.

Figure 87 Control Panel

After clicking the *Open/Close Offline File Button* (Button 2 in Figure 87), the user is asked to provide a log file (*.dat). After loading the log file, the user can navigate through the log data using backward buttons *Play Backward/Stop*, *Backward Data Packet*, *Backward Round*, *Seek to Begin* (Buttons 3 – 6 in Figure 87), forward buttons *Play Forward/Stop*, *Forward Data Packet*, *Forward Round*, *Seek to End* (Buttons 8 – 11 in Figure 87), or the slider (Item 7 in Figure 87). Keys *F9* and *ESC* start and stop forward playback (hot keys for *Play Forward/Stop Button*). Key *ESC* also stops backward playback.

The *Round Step IntervalControl* (Item 12 in Figure 87) allows to specify the step width in rounds for *Forward Round* and *Backward Round* navigation.

NOTE Round in this context is determined by the duration of the FlexRay cycle.

The *Forward to Trigger Button* (Button 14 in Figure 87) and the *Backward to Trigger Button* (Button 15 in Figure 87) may be used to jump to the next and to the previous trigger packet (*Trigger Packet* or *Triggerout Packet*) within the log data.

The *Fixed/Chronological Button* (Button 20 in Figure 87) allows to switch between the fixed view and the chronological view.

The chronological view displays all logged data packets in chronological order. The chronological order may be switch from decreasing to increasing temporal order by a click onto the header of the column *Time*.

The fixed view displays one data packet per packet container.

Data packets may be ordered by all available columns (*Bus*, *ID*, *Channel*, etc.). Additionally, data packets may be ordered by multiple columns by shift-clicking (Shift key + right mouse click) the respective column headers. E.g., to view all FlexRay data packets in chronological order click onto the header of column *Bus*. To view all FlexRay data packets ordered by frame IDs in chronological order click onto the header of column *Bus* and then onto the header of column *ID* while holding the Shift key. To deactivate multi-column ordering, click onto the header of a column which is not used for the active multi-column ordering.

In the context menu of the *Raw Data Window* as shown in Figure 88, the *Raw Data Window* may be cleared (context menu entry *Clear Raw Data Window*) or the displayed contents may be restored such that the active data packet is visible (context menu entry *Goto Active Data Packet*). The active data packet is indicated in bold characters in the *Raw Data Window*.

In the context menu of the header of the *Raw Data Window* the sort order and the column order within the *Raw Data Window* may be reset (context menu entries *Reset Sort Order* and *Reset Column Order*), respectively.

As an alternative to offline analysis using the VPT1000 PC Software, the log file can be converted to .csv or text format and analyzed with the user's tool of choice. For details about log file conversion see Chapter 5, “Log File Conversion,” starting on page 107.

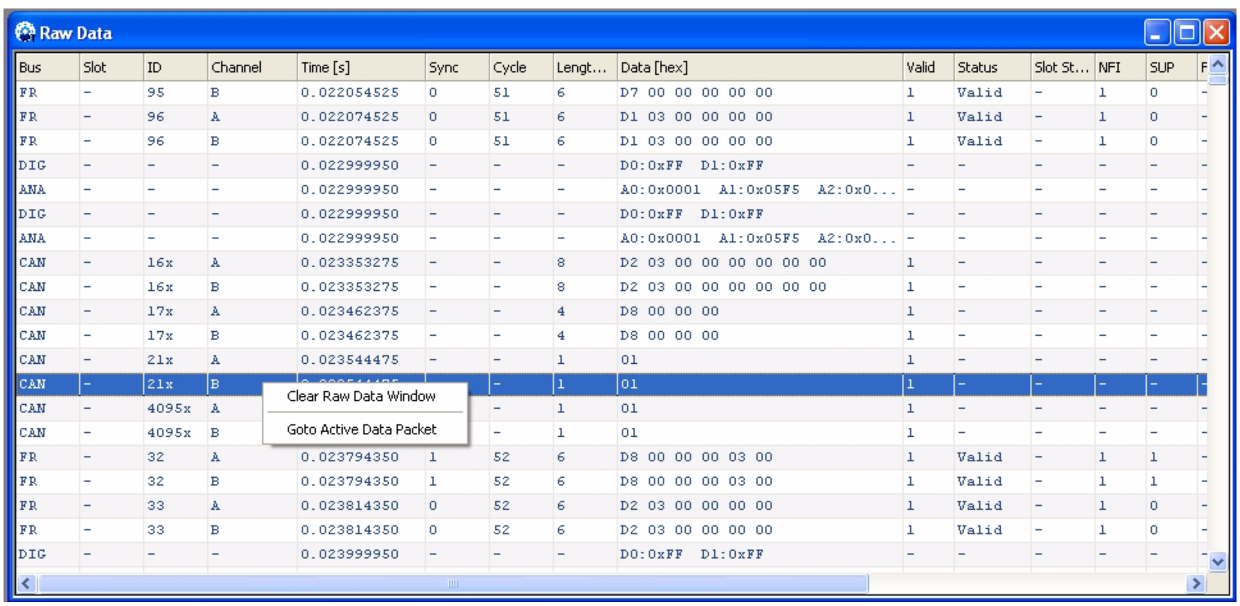


Figure 88 Context Menu of Raw Data Window

FlexRay Frame Transmission

Fault Injection

The fault injection feature allows bus traffic disturbance with low pulses generated by the VPT1000 Hardware using the N5479A-001 Advanced Software option. These disturbance pulses are configured via the COM interface (see "COM_BD_ConfigureFrameDestruction" on page 77). Configuration parameters for disturbance pulses are the channel, the slot ID, the cycle set (cycle base and cycle repetition), the offset of the pulse start relative to the slot start, the pulse duration, and the number of recurrences of the disturbance pulse. After configuration the disturbance function can be started (see "COM_BD_StartFrameDestruction" on page 78). The fault injection function is available only in *Synchronous Mode*.

Startup Buddy

The N5479A-001 Advanced Software option allows configuration of the VPT1000 Hardware to generate startup functionality.

The N5479A-001 Advanced Software option for VPT1000 Hardware comprises two additional FlexRay controllers (E-Rays) and enables the VPT1000 Hardware to startup a FlexRay cluster autonomously.

No additional coldstart node is required within the FlexRay cluster.

For detailed information about the configuration and the control of additional FlexRay controller(s) see "Controlling E-Ray Controller(s)" on page 78.

FlexRay frames transmitted by the additional FlexRay controller(s) can be monitored and recorded along with other frames in the cluster.

Frame Transmission/Reception With E-Ray Controller(s)

In addition to the startup buddy feature, additional FlexRay controller(s) can be used for FlexRay frame transmission and reception via the COM interface (refer to "Controlling E-Ray Controller(s)" on page 78). Applications may be implemented using that feature.

5
Log File Conversion

The VPT1000 PC Software stores the log files in a proprietary binary format readable only by the VPT1000 PC Software. The *BDLogConverter* is a command line tool to convert the binary log files into human readable text files. The converted files can then be analyzed manually.

The path of the *BDLogConverter* is registered in the system path environment variable during installation of the VPT1000 PC Software. After navigating into the directory where the log file is located, the *BDLogConverter* can be called in the *Command Window* with the following command:

```
BDLogConverter <BDFileIn> <Output_File_Format> [<BDFileOut>]
```

Calling the *BDLogConverter* without parameters provides usage information as shown in Figure 89.

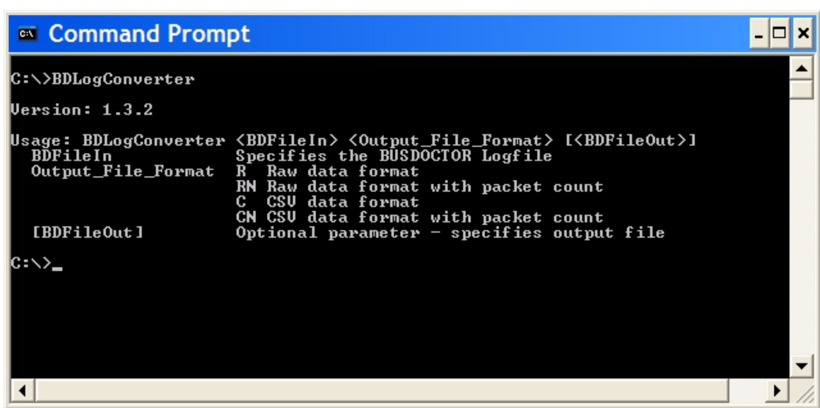


Figure 89 BDLogConverter help

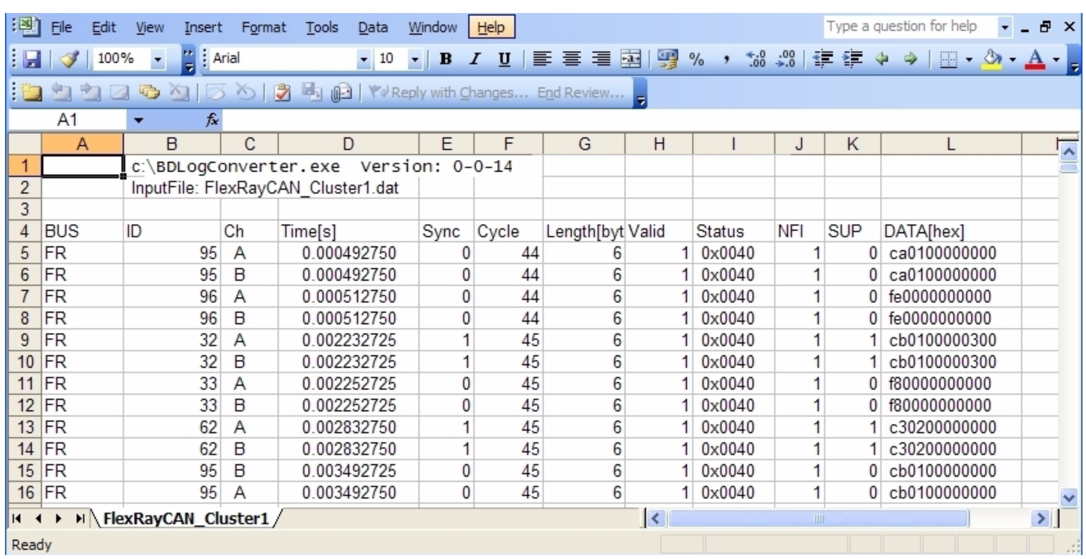
The *BDLogConverter* supports four different output formats: raw data format with and without packet count (output file format option R and RN) similar to the output in the *Raw DataWindow* of the VPT1000 PC Software and a comma separated format with and without packet count (output file format option C and CN). Raw data format and comma separated format are shown in Figure 90 and Figure 91. Converted log files in comma separated format may be easily imported into spreadsheet tools, e.g. Windows Excel (see Figure 92).

```
; c:\BDLogConverter.exe  Version: 0-0-14
; InputFile: FlexRayCAN_Cluster1.dat
;
;BUS |ID        |Ch |Time          |Sync |Cycle |Length  |Valid |Status |NFI |SUP |DATA
;    |          |   |  [s]         |     |      | [byte] |      |       |    |    | [hex]
FR   |        95|  A|   0.000492750|    0|    44|       6|     1| 0x0040|   1|   0| ca0100000000
FR   |        95|  B|   0.000492750|    0|    44|       6|     1| 0x0040|   1|   0| ca0100000000
FR   |        96|  A|   0.000512750|    0|    44|       6|     1| 0x0040|   1|   0| fe0000000000
FR   |        96|  B|   0.000512750|    0|    44|       6|     1| 0x0040|   1|   0| fe0000000000
FR   |        32|  A|   0.002232725|    1|    45|       6|     1| 0x0040|   1|   1| cb0100000300
FR   |        32|  B|   0.002232725|    1|    45|       6|     1| 0x0040|   1|   1| cb0100000300
FR   |        33|  A|   0.002252725|    0|    45|       6|     1| 0x0040|   1|   0| f80000000000
FR   |        33|  B|   0.002252725|    0|    45|       6|     1| 0x0040|   1|   0| f80000000000
FR   |        62|  A|   0.002832750|    1|    45|       6|     1| 0x0040|   1|   1| c30200000000
FR   |        62|  B|   0.002832750|    1|    45|       6|     1| 0x0040|   1|   1| c30200000000
FR   |        95|  B|   0.003492725|    0|    45|       6|     1| 0x0040|   1|   0| cb0100000000
FR   |        95|  A|   0.003492750|    0|    45|       6|     1| 0x0040|   1|   0| cb0100000000
```

Figure 90 Converted log file in raw data format

```
; c:\BDLogConverter.exe  Version: 0-0-14
; InputFile: FlexRayCAN_Cluster1.dat
;
BUS;ID;Ch;Time[s];Sync;Cycle;Length[byte];Valid;Status;NFI;SUP;DATA[hex]
FR   ;        95;  A;   0.000492750;    0;    44;       6;     1; 0x0040;   1;   0; ca0100000000
FR   ;        95;  B;   0.000492750;    0;    44;       6;     1; 0x0040;   1;   0; ca0100000000
FR   ;        96;  A;   0.000512750;    0;    44;       6;     1; 0x0040;   1;   0; fe0000000000
FR   ;        96;  B;   0.000512750;    0;    44;       6;     1; 0x0040;   1;   0; fe0000000000
FR   ;        32;  A;   0.002232725;    1;    45;       6;     1; 0x0040;   1;   1; cb0100000300
FR   ;        32;  B;   0.002232725;    1;    45;       6;     1; 0x0040;   1;   1; cb0100000300
FR   ;        33;  A;   0.002252725;    0;    45;       6;     1; 0x0040;   1;   0; f80000000000
FR   ;        33;  B;   0.002252725;    0;    45;       6;     1; 0x0040;   1;   0; f80000000000
FR   ;        62;  A;   0.002832750;    1;    45;       6;     1; 0x0040;   1;   1; c30200000000
FR   ;        62;  B;   0.002832750;    1;    45;       6;     1; 0x0040;   1;   1; c30200000000
FR   ;        95;  B;   0.003492725;    0;    45;       6;     1; 0x0040;   1;   0; cb0100000000
FR   ;        95;  A;   0.003492750;    0;    45;       6;     1; 0x0040;   1;   0; cb0100000000
FR   ;        96;  B;   0.003512725;    0;    45;       6;     1; 0x0040;   1;   0; f80000000000
```

Figure 91 Converted log file in csv format



	A	B	C	D	E	F	G	H	I	J	K	L
1		c:\BDLogConverter.exe Version: 0-0-14										
2		InputFile: FlexRayCAN_Cluster1.dat										
3												
4	BUS	ID	Ch	Time[s]	Sync	Cycle	Length[byt	Valid	Status	NFI	SUP	DATA[hex]
5	FR	95	A	0.000492750	0	44	6	1	0x0040	1	0	ca0100000000
6	FR	95	B	0.000492750	0	44	6	1	0x0040	1	0	ca0100000000
7	FR	96	A	0.000512750	0	44	6	1	0x0040	1	0	fe0000000000
8	FR	96	B	0.000512750	0	44	6	1	0x0040	1	0	fe0000000000
9	FR	32	A	0.002232725	1	45	6	1	0x0040	1	1	cb0100000300
10	FR	32	B	0.002232725	1	45	6	1	0x0040	1	1	cb0100000300
11	FR	33	A	0.002252725	0	45	6	1	0x0040	1	0	f80000000000
12	FR	33	B	0.002252725	0	45	6	1	0x0040	1	0	f80000000000
13	FR	62	A	0.002832750	1	45	6	1	0x0040	1	1	c30200000000
14	FR	62	B	0.002832750	1	45	6	1	0x0040	1	1	c30200000000
15	FR	95	B	0.003492725	0	45	6	1	0x0040	1	0	cb0100000000
16	FR	95	A	0.003492750	0	45	6	1	0x0040	1	0	cb0100000000

Figure 92 Converted log file in csv format imported into Windows Excel

6
Measurement Data Representation

Measurement data is captured by the VPT1000 Hardware in data packet units. These elements are transmitted to the host PC, stored in logfiles and displayed in the raw data window (see "Raw Data Window" on page 45) respectively extracted by the BDLogConverter (see Chapter 5, “Log File Conversion,” starting on page 107).

FlexRay data packets

For gathering the exact FlexRay traffic the following data packets are supported for FlexRay:

- FlexRay frame data packet
- FlexRay slot status data packet
- FlexRay symbol data packet

Content and semantic of FlexRay data packets

FlexRay frame data packets

This data packet type is generated by the VPT1000 Hardware when monitoring error free FlexRay traffic without FlexRay symbols. In this case one data packet is generated per received FlexRay frame containing the following information:

- Exact timestamp of the frame reception (25ns solution, captured at the beginning of the TSS)
- Decoded header and payload data
- Frame status of the received FlexRay frame (timing status only available if VPT1000 Hardware is synchronous to cluster)
- Slot number the FlexRay frame was received in (only available if synchronous to cluster)
- Slot status of the slot the FlexRay frame was received in (only available if VPT1000 Hardware is synchronous to cluster)

FlexRay slot status data packets

This data packet is generated in case of erroneous traffic if the FlexRay slot status cannot be appended to a FlexRay frame data packet. This is for example the case if a FlexRay slot is overbooked. This kind of data packets is only generated if the VPT1000 Hardware is synchronous to the cluster. The data packet contains the following information:

- Exact timestamp of the slot boundary the slot belongs to
- Slot number the slot status belongs to
- Slot status of the corresponding FlexRay slot

FlexRay symbol data packets

Communication elements with a low phase longer then 30 times of *gdBit* are interpreted as FlexRay symbols. FlexRay symbol data packets contain the following information:

- Exact timestamp of the FlexRay symbol (beginning of the low phase)

- Length of the received low phase (in bit cells)
- Status of the decoded symbol

FlexRay data packet chronology

There are different normal and fault scenarios available when monitoring FlexRay traffic. This section shall list the possible scenarios and specify the sequence of data packets generated by the VPT1000 Hardware. Figure Figure 93 shows the possible scenarios.

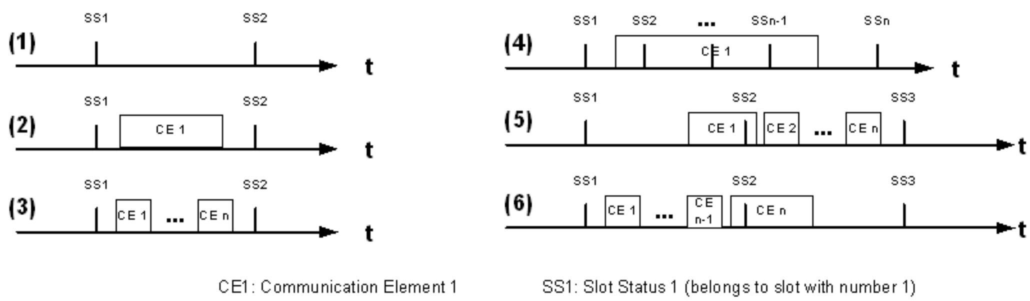


Figure 93 Different scenarios in a FlexRay cluster

Empty slot scenario (Figure 93 - 1)

An empty slot does not generate any FlexRay data packet.

One communication element in one slot (Figure 93 - 2)

One communication element starts and ends in one slot. In this case one *FlexRay frame data packet* is generated containing all available information including the slot status of the slot the communication element is decoded in. The slot number in this case is *2* and the appended slot status is *SS2*.

Multiple communication elements in one slot (Figure 93 - 3)

This scenario shows multiple communication elements in one slot. All the communication elements completely start and end in the same slot.

In this case each communication element generates a *FlexRay frame data packet* containing all available information except the FlexRay slot status. The slot status is only appended to the last communication element completely finished in the slot.

One communication element booking multiple slots (Figure 93 - 4)

Scenario 4 shows one communication element booking more than one slot. In this case one *FlexRay frame data packet* is generated containing all the available information of the communication element except the slot status information. The slot number assigned to the data packet is from the slot the communication elements ends in.

Additionally to the *FlexRay frame data packet* one *FlexRay slot status data packet* is generated for all slots the communication element booked.

Note that even the last slot status is not appended to the *FlexRay frame data packet*. A slot status is only appended to the last fully decoded communication element in one slot!

The following FlexRay data packets are generated in the order they are listed:

- FlexRay frame data packet of communication element *CE1* with slot number *n* and without slot status information
- FlexRay slot status data packet for the slots *SS2* to *SSn*

Special Scenario A (Figure 93 - 5)

In this special scenario already characterized scenarios are mixed. The following FlexRay data packets are generated in the order they are listed:

- FlexRay frame data packet of communication element *CE1* with slot number *3* and without slot status information
- FlexRay slot status data packet of slot status *SS2*
- FlexRay frame data packets for *CE2* to *CEn-1* without slot status information and slot number *3*
- FlexRay frame data packet for *CEn* with the slot number *3* and the slot status information *SS3*

Special Scenario B (Figure 93 - 6)

In this special scenario already characterized scenarios are mixed, too. The following data packets are generated in the order they are listed:

- FlexRay frame data packets for *CE1* to *CEn-1* without slot status information and slot number *2*
- FlexRay frame data packet for *CEn* with the slot number *3* and without slot status information
- FlexRay slot status data packets for *SS2* and *SS3*

FlexRay status bits

This section describes in detail the FlexRay status provided with the FlexRay data packets.

Asynchronous vs. synchronous measurement

A measurement of a FlexRay cluster can be performed asynchronous or synchronous to the FlexRay cluster (see "FlexRay Settings" on page 89). While measuring in asynchronous mode only the syntax status information is available. No timing status or slot status is available in this mode. Even no *FlexRay slot status data packets* are generated in asynchronous mode.

In the synchronous mode, timing status, slot status, and the slot number are provided to the measured FlexRay data.

Syntax status bits

The following syntax status bits are supported for *FlexRay frame data packets*:

- Transmit start sequence violation (TSSVIOL): This violation is indicated if the transmit start sequence of the received communication element is longer than the configured parameter *gdTSSTransmitter*. See also FlexRay specification 2.1A, figure 3-26. Note that this parameter is configured to the maximum in asynchronous mode and therefore never will appear.
- Frame start sequence error (FSSERR): Frame start sequence respectively first byte start sequence detection produced a decoding error. See also FlexRay specification 2.1A, figure 3-28.
- Byte start sequence error (BSSERR): Byte start sequence detection produced a decoding error. See also FlexRay specification 2.1A, figure 3-31.
- Frame end sequence error (FESERR): Frame end sequence detection produced a decoding error. See also FlexRay specification 2.1A, figure 3-32.
- Coding error (CODERR): This error is a combination of at least one of the above listed errors (TSSVIOL, FSSERR, BSSERR, FESERR)
- Header CRC error (HCRCERR): This flag indicates a header CRC error. See also FlexRay specification 2.1A, figure 3-34.
- Frame CRC error (FCRCERR): This flag indicates a frame CRC error. See also FlexRay specification 2.1A, figure 3-34.

Additionally syntax status bits for *FlexRay symbol data packet*:

- Symbol length error (SYERR): This flag indicates a low phase longer than the allowed maximal symbol length *gdCASrxLowMax*.

Timing status bits

For validating correct timing of communication elements, the start and end points of a communication element are validated as follows:

- FlexRay frames (beginning low phase smaller than 30*gdBit)
  - The start point is given by the falling edge of the transmit start sequence (TSS)
  - The end point is given by

    - The end of the channel idle recognition point (CHIRP) for the slot boundary check
    - The strobe point of the last bit of the frame CRC for other checks (identifier mismatch, cycle counter mismatch, belonging to the communication part and valid check)
- FlexRay symbols (low phase greater than or equal to 30*gdBit)
    - The start point is given by the falling edge of the low phase
    - The end point is given by the raising edge of the end of the low phase

The following timing information bits are provided:

- Sync frame in dynamic segment error (SYNCERR): This flag indicates a sync frame in the dynamic segment.
- Startup frame in dynamic segment error (SUERR): This flag indicates a startup frame in the dynamic segment.
- Sync or startup frame in dynamic segment error (SSERR): This flag is a combination of one of the above listed errors SYNCERR or SUERR.
- Nullframe in dynamic segment error (NERR): Indicates a nullframe in the dynamic segment.
- Startup frame without Sync frame error (SUWSYERR): Indicates a startup frame which is no sync frame.
- Frame identifier error (FIDERR): This flag is set if the decoded identifier does not match to the slot the communication element is received in.Note that the symbol window and the NIT are interpreted as slot number zero.
- Cycle counter error (CCERR): This flag is set if the decoded cycle counter does not match the cycle the communication element was received in.
- Static payload length error (SPLERR): Indicates a communication element in the static segment with a decoded payload length different to the configured payload length *gdPayloadLengthStatic*.
- Slot overbooked error (SOVERR): This flag is set if the corresponding communication element is not the first communication element in the slot where the corresponding communication element ends in.
- Boundary violation (BVIOL): Indicates at least one (static) slot boundary during decoding of the communication element has occurred.
- Symbol window violation (SWVIOL): Indicates either a communication element crossing the symbol window start- or endpoint or a communication element which is not a symbol window.
- Network idle time violation (NITVIOL): Indicates a communication element crossing the NIT boundaries or a communication element in the NIT.

Slot status bits

The following slot status information bits are supported:

- Syntax error detected (SED): Indicates one or more observed syntax errors in the corresponding slot.
- Content error detected (CED): Indicates one or more observed content errors in the corresponding slot.
- Additional Communication Indicator (ACI): Indicates one or more observed valid frames and additional activity in the corresponding slot.
- Slot boundary violation (SBV): Indicates one or more observed boundary violations. Note that even a boundary violation between symbol window and NIT is indicated with this bit.

CAN data packets

For gathering the CAN traffic with the VPT1000 Hardware a special CAN data packet is supported.

Content of the CAN frame data packet

A CAN frame data packet is generated by the VPT1000 Hardware when a communication element is received and decoded on the selected CAN channel. The following information is packed into the CAN frame data packet:

- Exact time stamp of the frame reception (25ns solution, captured at the edge of the first dominant bit in the arbitration field)
- Decoded header and payload data
- Frame status of the received CAN frame

Status bits of the CAN frame data packet

Like the FlexRay data also the received CAN data is checked for correct syntax by the VPT1000 Hardware. The following status bits are supported for monitoring CAN data:

- Stuff error (StuffError): Sequence of more than 5 equal-valued bits received in a part of a frame where this is not allowed
- Form error (FormError): Fixed-format part of a received frame exhibits a wrong format
- CRC error (CRCError): CRC checksum received does not match the CRC value calculated
- Error warning limit reached (ERRW): Receive error counter reached warning limit
- Error passive limit reached (ERRP): Receive error counter reached error limit
- Bus off limit reached (BOFF): Receive error counter reached bus off limit

ADIO data packets

For capturing analog and digital inputs together with FlexRay and CAN data (synchronized time stamp) the hardware supports a special ADIO data packet.

Capturing of ADIO data is driven by a configurable sample rate (see "Analog/Digital IO Settings" on page 95). The following data information is packed into the ADIO data packet:

- Exact capturing time stamp of the sample point.
- Captured value of digital input lines
- Captured value of analog input lines

Trigger data packets

There are different trigger data packets available for marking trigger events in the data stream. The trigger events and the data before and after can than be analyzed offline in more detail.

The following trigger data packets are available:

- Trigger mark data packets
- General marker data packets
- Triggerout pins data packets

Triggerout pins data packet

The marker *Triggerout pins data packet* is only inserted in *Single Pulse Mode* of the triggerout pins (see "Triggerout Pin Settings" on page 96). The content of the data packet shell mark the trigger condition(s) which caused the trigger pin(s) to rise.

The data packet contains the following information:

- Exact time stamp of the trigger condition
- FlexRay status information for both FlexRay channels. Indicates which of the (configured) FlexRay status bits caused the triggerout pins. For more detailed information on the FlexRay status bits see "FlexRay status bits" on page 112.
- Timing events, which caused the trigger condition for triggerout pins.
- Trigger lines, which where driven by the marked trigger conditions.

Index