



Table of Contents

1	Important and general information	4
1.1	Important information	4
1.1.1	Safety and Warning instructions	4
1.2	Terms and conditions	5
1.2.1	Legend of used icons	5
1.2.2	Support	5
2	PlugIn Overview	6
2.1	PlugIn description	6
2.2	PlugIn installation	6
2.3	Overview of supported vendors	7
2.4	Detailed list of vendors and devices	8
3	PlugIn configuration	11
3.1	Functional architecture	11
3.2	Creating interface systems	12
4	Device specific configurations	14
4.1	IPEhub2 specific functions	14
4.2	ETHgateway-CLFD specific function	16
4.3	CAN FD Satellite	17
4.4	FlexRay Satellite	17
4.5	LIN Satellite	18
5	Measurements on CAN FD, LIN, ETH, FlexRay interfaces	19
5.1	Description file import format overview	20
5.2	CAN interface	22
5.3	LIN interface	27
5.4	ETH interface	29
5.4.1	ETH Status channels	31
5.4.2	ETH Traffic	31
5.4.3	IPETRONIK X	33
5.4.4	IP Cameras	33
5.4.5	Satellites for CAN FD, LIN, FlexRay	34
5.4.6	PLP Devices - Automotive Ethernet	35
5.4.7	WWH-OBD	36
5.4.8	DLT Diagnostic Log and Trace	37
5.4.9	IPETRONIK HV PMD-1	38
5.4.10	EtherCAT Master	38
5.4.11	SIEMENS S7 PLC	40
5.4.12	Modbus TCP	40
5.4.13	Aerospace ARINC 429	41
5.5	A2L import - DAQ list with graphical filling level indication	43
5.5.1	DAQ list filling process	45
5.5.2	DAQ list overflow – rejected signal export	47
5.6	Edit A2L dynamic DAQ list ODT values during the import	48
5.6.1	A2L Import for array signals	50
5.6.2	A2L import with additional FlexRay parameters import	53
5.6.3	A2L import from zip files (X-Modules)	57
5.6.4	XCP and CCP with Seed & Key with SKB licensing	58
5.6.5	XCP, CCP second tester	59
5.6.6	XCP transport layers	61
5.6.7	XCPonETH for IPv6	62
5.7	FIBEX import	63
5.7.1	Import CAN signals from FIBEX files	63
5.7.2	Import CAN FD signals from FIBEX files	65
5.7.3	Display of Sender name for FIBEX, DBC, AUTOSAR messages	66
5.8	Description file import with CSV file for channel reference	67
5.8.1	Multi column CSV selection for description file imports (DBC, A2L)	69

5.8.2 Check duplicate channel names during description file import	72
5.9 Description file import with INCA LAB file	76
5.10 Synchronization of description files	79
5.11 J1939 Diagnostics	83
5.12 UDS Diagnostics	89
5.13 OBD-2 diagnostics	92

1 Important and general information

1.1 Important information

Please follow these instructions before and during the use and application on any IPETRONIK product!

1.1.1 Safety and Warning instructions

Please follow the instructions **and** information as contained in the user manual!

1. The user can **influence an electronic system by applying the IPETRONIK product**. This might cause risk of personal injury or property damages.
2. The **use and application of the IPETRONIK product is permitted only to qualified professional staff**, as well as, only in appropriate manner and in the designated use.
3. **Before using an IPETRONIK measurement system** in the vehicle it **has to be verified that no function of the vehicle, which is relevant for secure operation, might be influenced**:
 - by the installation of the IPETRONIK measurement system in the vehicle,
 - by an potential malfunction of the IPETRONIK system during the test drive.

In order to avoid possible danger or personal injury and property damages, appropriate actions are to be taken; such actions have to bring the entire system into a secured condition (e.g. by using a system for emergency stop, an emergency operation, monitoring of critical values).

Please check the following points to avoid errors:

- Adaption of sensors to components of the electrical system / electronics, brake system, engine and transmission control, chassis, body.
- Tap of one or several bus systems (CAN, LIN, ETHERNET) including the required electrical connection(s) for data acquisition.
- Communication with the vehicle's control units (ECUs), especially with such of the brake system and/or of the engine and transmission control (power train control system).
- Installation of components for remote data transmission (mobiles, GSM/GPRS modems, WiFi and Bluetooth components).



The products can be operated in extended temperature ranges greater 70 °C and therefore the operator has to take safety measures to avoid any skin burnings on hot surfaces while touching the products.

4. **Before** directly or indirectly using **the data acquired by an IPETRONIK measurement system to calibrate control units, please review the data regarding to plausibility**.
5. With regard to the application of IPETRONIK products in vehicles during use on public roads the manufacturer and/or registered user of the vehicle **has to ensure that all changes/modifications have no influence concerning the license of the vehicle or its license of operation**.
6. **User does agree to the instructions and regulations as mentioned above**. In case the user does not agree with the instructions and regulations as mentioned above, he has to notify this expressly and immediately in writing to IPETRONIK before confirming the sales contract.

1.2 Terms and conditions

See IPETRONIK website for details: www.ipetronik.com

1.2.1 Legend of used icons

**Tip**

This icon indicates a useful tip that facilitates the application of the software.

**Information**

This icon indicates additional information for a better understanding.

**Attention!**

This icon indicates important information to avoid potential error messages.

1.2.2 Support

Headquarter:**IPETRONIK GmbH & Co. KG**

Im Rollfeld 28
76532 Baden-Baden, Germany
Phone +49 7221 9922 0
Fax +49 7221 9922 100
info@ipetronik.com
Website: www.ipetronik.com

Limited commercial partnership with its head office in Baden-Baden, registry court HRA No. 201313
IPETRONIK Verwaltungs-GmbH Baden-Baden is an individually liable society, registry court Mannheim HRB
No. 202089
CEOs: A. Wocke, C. Buchholz

Technical support and product information e-mail: support@ipetronik.com

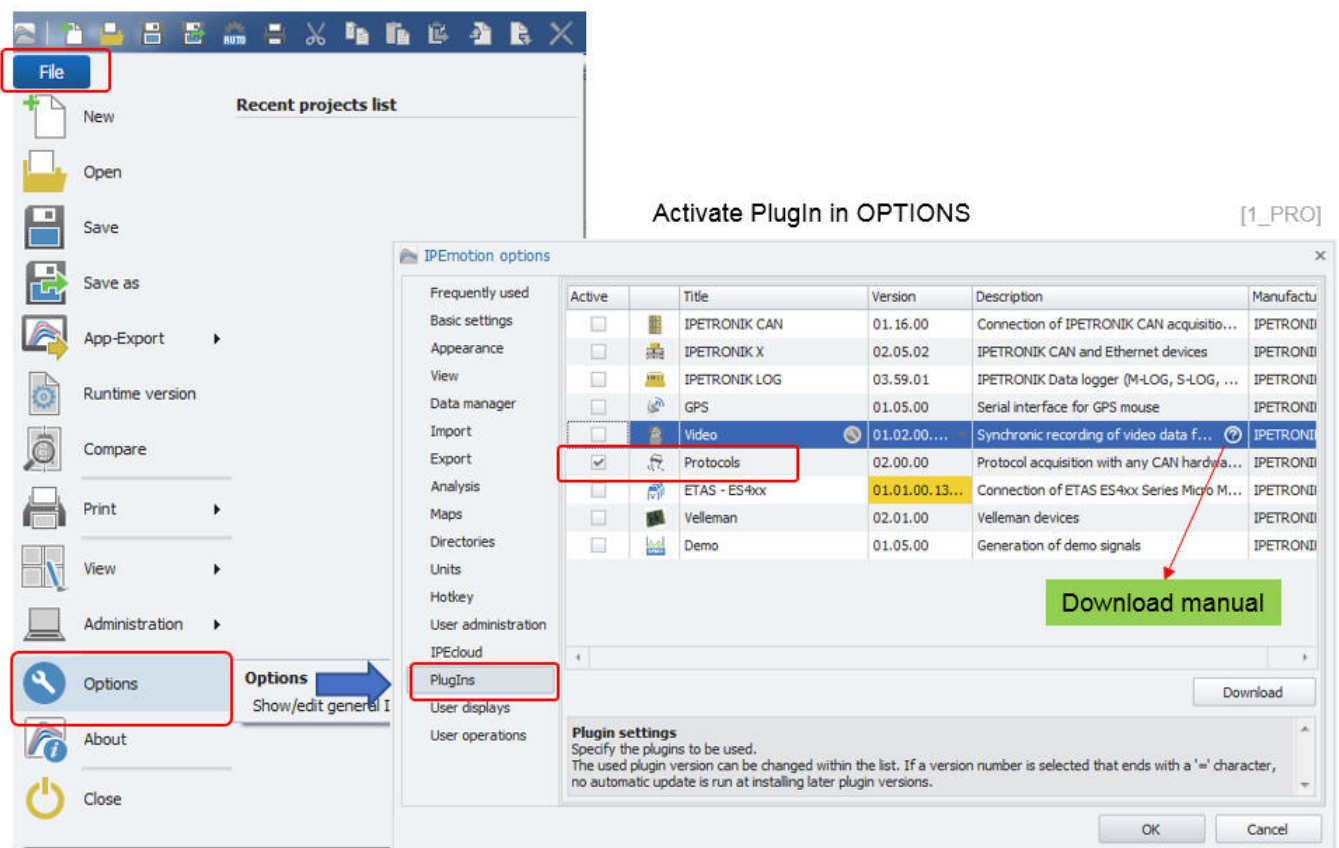
2 PlugIn Overview

2.1 PlugIn description

The Protocols PlugIn is supporting the measurement of traffic and bus networks protocols. A large range of different hardware interfaces from various vendors is supported. The PlugIn can measure bus network data, but you can also send traffic data to your network and ECUs. The IPEmotion software is providing instruments and function for traffic analysis and traffic generation and output.

2.2 PlugIn installation

In order to use the PlugIn together with IPEmotion you need to install it. The PlugIn is available for download from the IPETRONIK website: <https://www.ipetronik.com/> When you have installed the PlugIn, you need to launch the IPEmotion software. Then you need to access the application menu and open the OPTIONS. In the OPTIONS you can activate the PlugIn as indicated below.

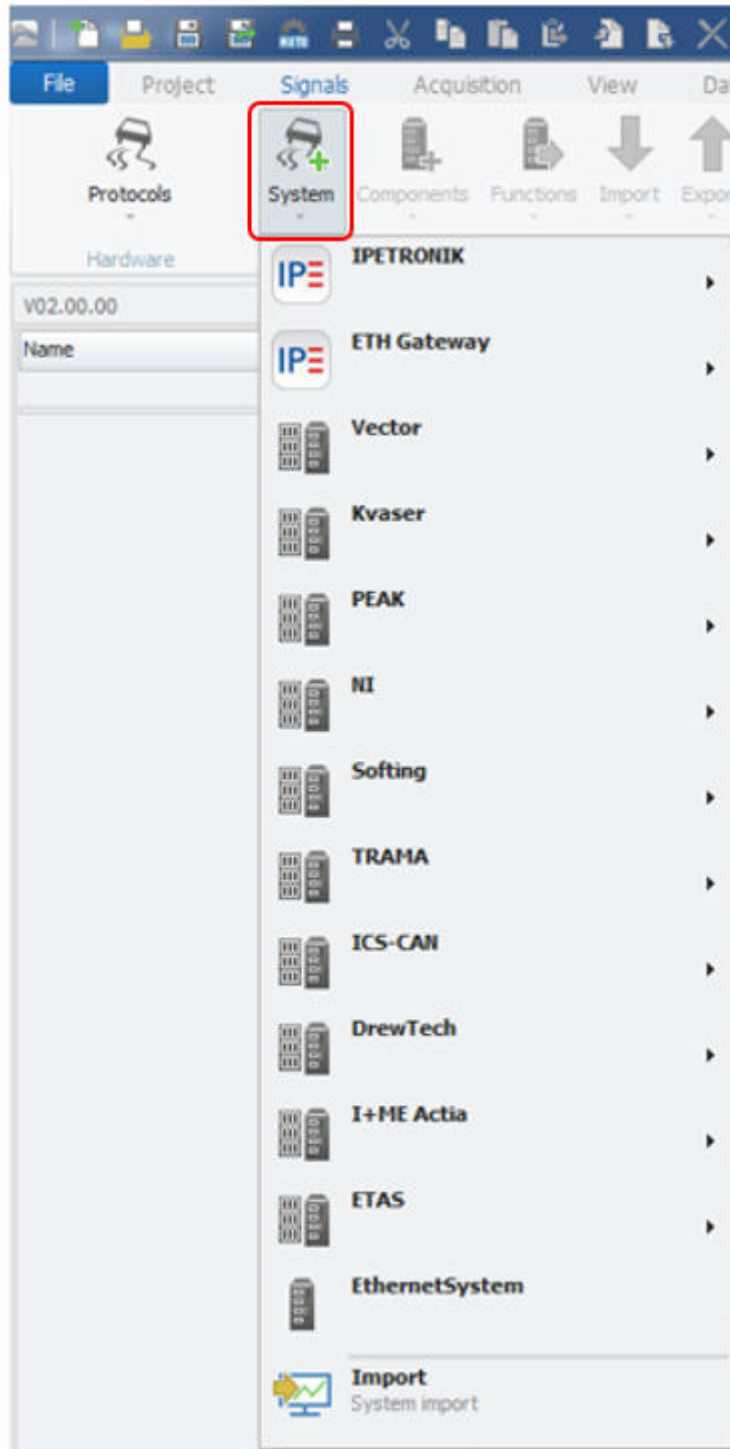


The PlugIn is supporting the following Windows operating systems:

- ▶ 32 bit
- ▶ 64 bit

2.3 Overview of supported vendors

Within the PlugIn a large range of hardware interfaces is supported from different vendors, to perform your network measurements. The list of vendors and devices is continuously growing. If your specific vendor or interface is missing, please contact our support to see if the implementation is possible in the oncoming releases.



[2_PRO]

2.4 Detailed list of vendors and devices

▶ IPETRONIK	IPEhub2
▶ IPETRONIK	IPEcan FD
▶ IPETRONIK	IPEcan FD PRO
▶ IPETRONIK	IPEcan
▶ IPETRONIK	IPEcan PRO
▶ IPETRONIK	M-WiFi
▶ IPETRONIK	ETHgateway CLFD V1.1
▶ IPETRONIK	ETHgateway CLFD V1.2
▶ IPETRONIK	CAN FD Satellite
▶ IPETRONIK	FlexRay Satellite
▶ VECTOR	CANcardXLe
▶ VECTOR	CANcardXL
▶ VECTOR	CANcaseXL
▶ VECTOR	CANboardXL
▶ VECTOR	CANboardXLcompact
▶ VECTOR	CANcardX
▶ VECTOR	VN1610
▶ VECTOR	VN1611
▶ VECTOR	VN1630
▶ VECTOR	VN1640
▶ VECTOR	VN5610
▶ VECTOR	VN5610A
▶ VECTOR	VN7570
▶ VECTOR	VN7572
▶ VECTOR	VN7600
▶ VECTOR	VN7610
▶ VECTOR	VN8900
▶ VECTOR	VN8950
▶ VECTOR	VN8970
▶ VECTOR	VN8972

▶ VECTOR	VX0312
▶ VECTOR	VX1121
▶ VECTOR	VX1131
▶ National Instruments	PCI-CAN
▶ National Instruments	PXI-CAN
▶ National Instruments	PCMCIA-CAN
▶ National Instruments	AT-CAN
▶ National Instruments	USB-CAN
▶ Kvaser	LAPcan
▶ Kvaser	PCIEcan
▶ Kvaser	PCcan
▶ Kvaser	PClcan
▶ Kvaser	PClcan II
▶ Kvaser	USBcan II
▶ Kvaser	Leaf II
▶ Kvaser	Leaf
▶ Kvaser	PClcanx II
▶ Kvaser	Memorator Professional II
▶ Kvaser	MemoratorPro
▶ Kvaser	Memorator Light
▶ Kvaser	USBcan Pro 5xHS
▶ Kvaser	USBcanPro
▶ Kvaser	USBcan Light
▶ Kvaser	BlackBird
▶ Kvaser	BlackBird V2
▶ Kvaser	Hybrid
▶ Softing	CANcard2
▶ Softing	EDICcardC
▶ Softing	EDICcard2
▶ Softing	CAN-Acx-PCI
▶ Softing	CAN-Acx-PCI/DN
▶ Softing	CANusb
▶ Softing	CAN-PROx-PCI

- ▶ Peak PCAN-USB X6
- ▶ Peak PCAN-PCI
- ▶ Peak PCAN-PCIe
- ▶ Peak PCAN-PCIe FD
- ▶ TRAMA CW-ISUB
- ▶ ICS ValueCAN
- ▶ ICS ValueCAN3
- ▶ ICS ValueCAN4
- ▶ ICS ValueCAN4-4
- ▶ DREWTECH Mongoose
- ▶ I+ME ACTIA Basic+24 XS
- ▶ ETAS ES581
- ▶ ETAS ES593
- ▶ ETAS ES595
- ▶ EthernetSystems

3 PlugIn configuration

3.1 Functional architecture

The following diagram shows the schematic system architecture when you would like to perform network traffic and protocol measurements. You need the IPEmotion software and the corresponding Protocols PlugIn. Alos you need to install the hardware drivers for your specific interfaces. For IPETRONIK interfaces the drivers are included in the setup and installed automatically.



PC with IPEmotion DAQ Software



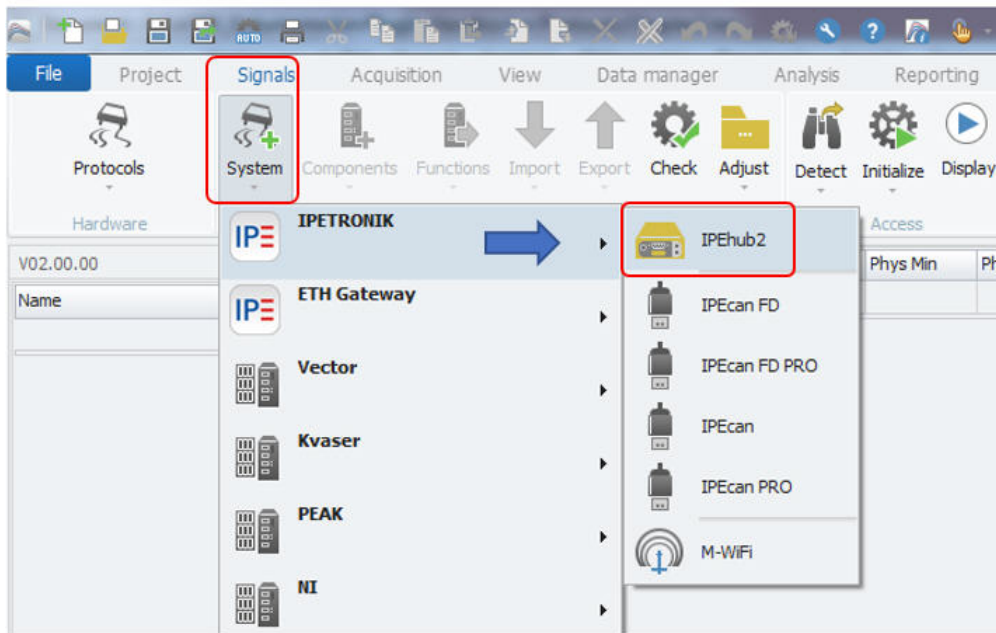
ETH
WiFi
USB



[3_PRO]

3.2 Creating interface systems

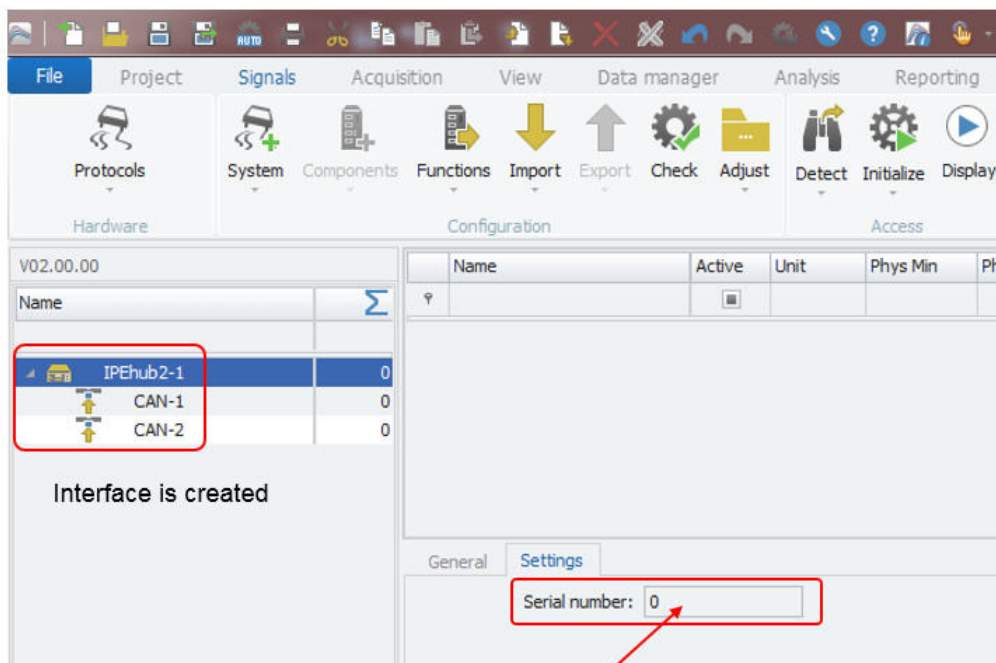
In order to start your measurement, you need to change to the SIGNALS work space and select the Protocols PlugIn from the hardware system drop down box. After that you need to create a hardware interface system from the list of devices.



Create new interface manually

[4_PRO]

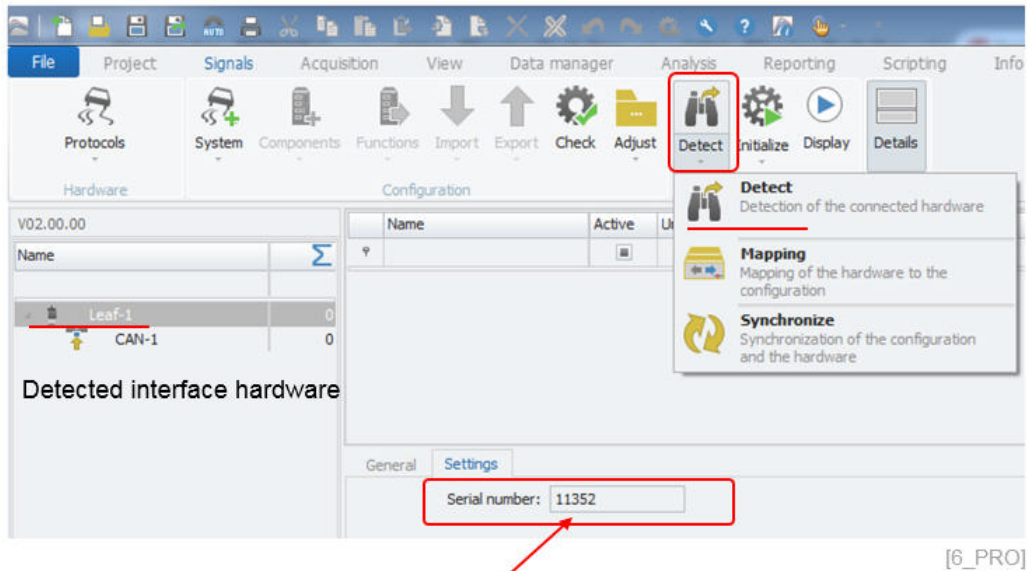
When the systems are manually created you have to enter the device serial number or IP-address to establish a connection for your measurement.



Enter serial number

[5_PRO]

Some devices support automatic hardware detection. In this case you do not need to create the systems manually. You can use the detect function from the ribbon.



Device serial number is detected



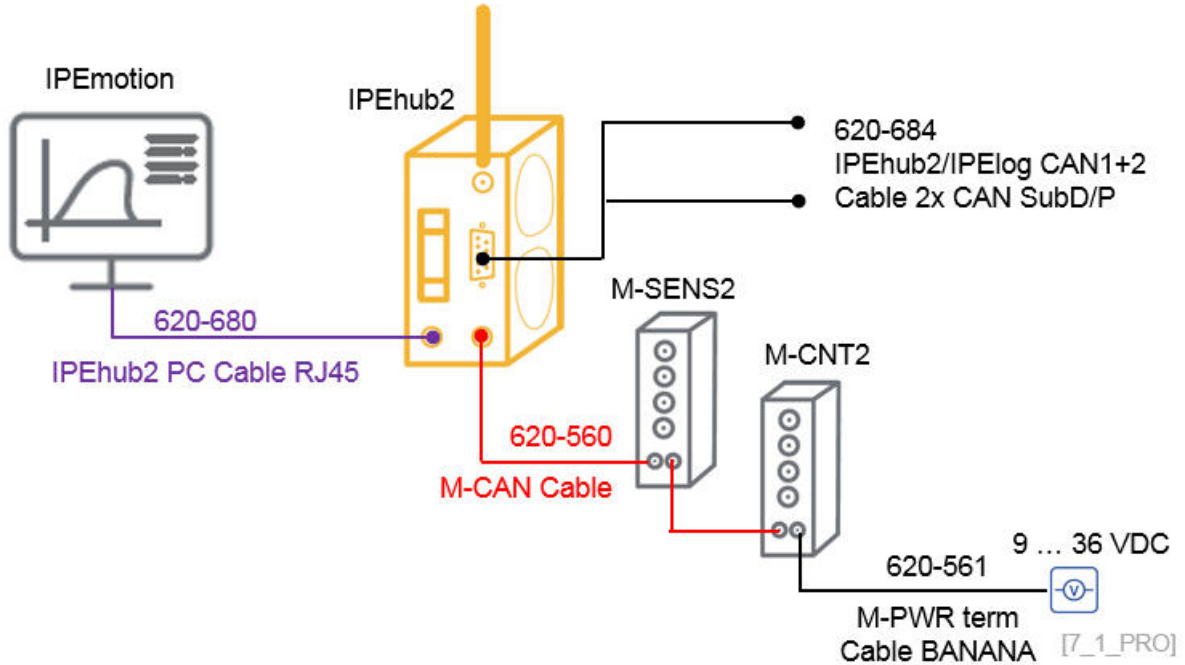
Information

Note that the Plugin might not support the same set of functions which are available through the vendors own software platforms. This can have several reasons, e.g. that the provided programming API from the vendor is not offering all functions to external developers. Another reason can be that IPETRONIK did not implement the function.

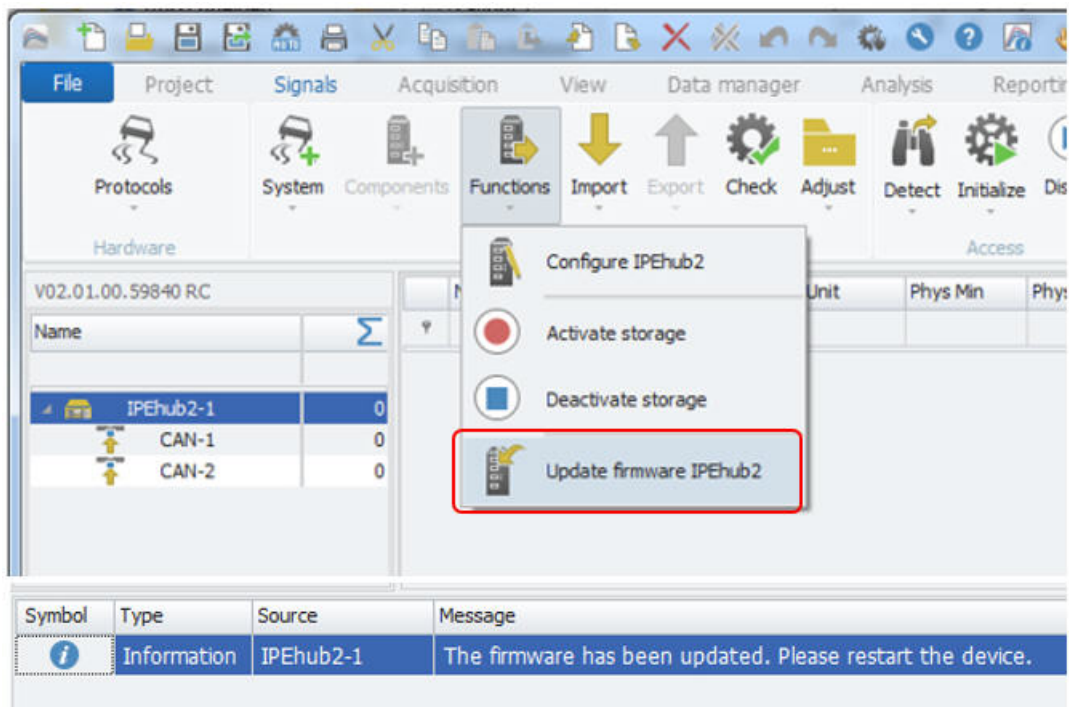
4 Device specific configurations

4.1 IPEhub2 specific functions

The cable sets to interface IPEhub2 to the PC and the M-CAN modules on CAN2 or to the CAN bus network via the SUB D9 interface are presented below:



The IPEhub2 device can be detected automatically over the LAN interface. The devs running a DHCP server and when your LAN network interface of the PC is configured for automatic IP-address you can directly detect it.



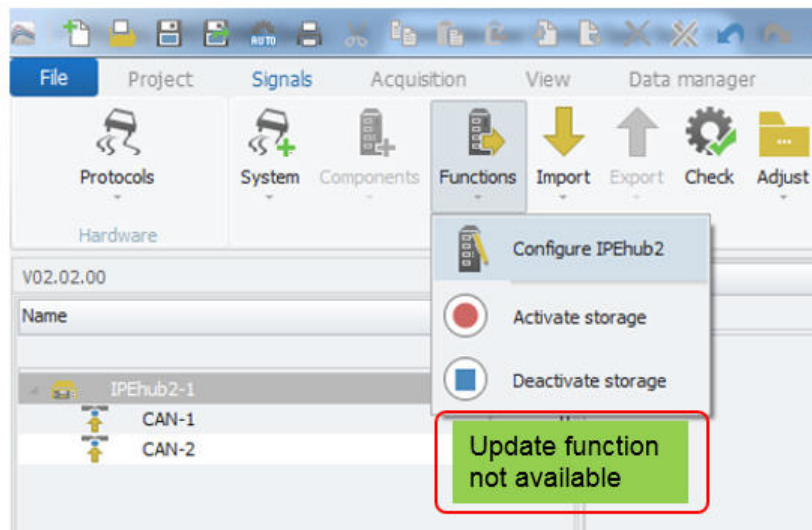
Update IPEhub2 device firmware over LAN only

[7_PRO]

The IPEhub2 device firmware is installed in the following directory of the Protocols PlugIn

- ▶ C:\Program Files (x86)\IPETRONIK\IPEmotion PlugIn Protocols V02.0x.0x\IPEhub2

The firmware update function is only available when you have detected the IPEhub2 device over the LAN cable. Over WiFi connection the firmware update is not supported because the update process is not stable.

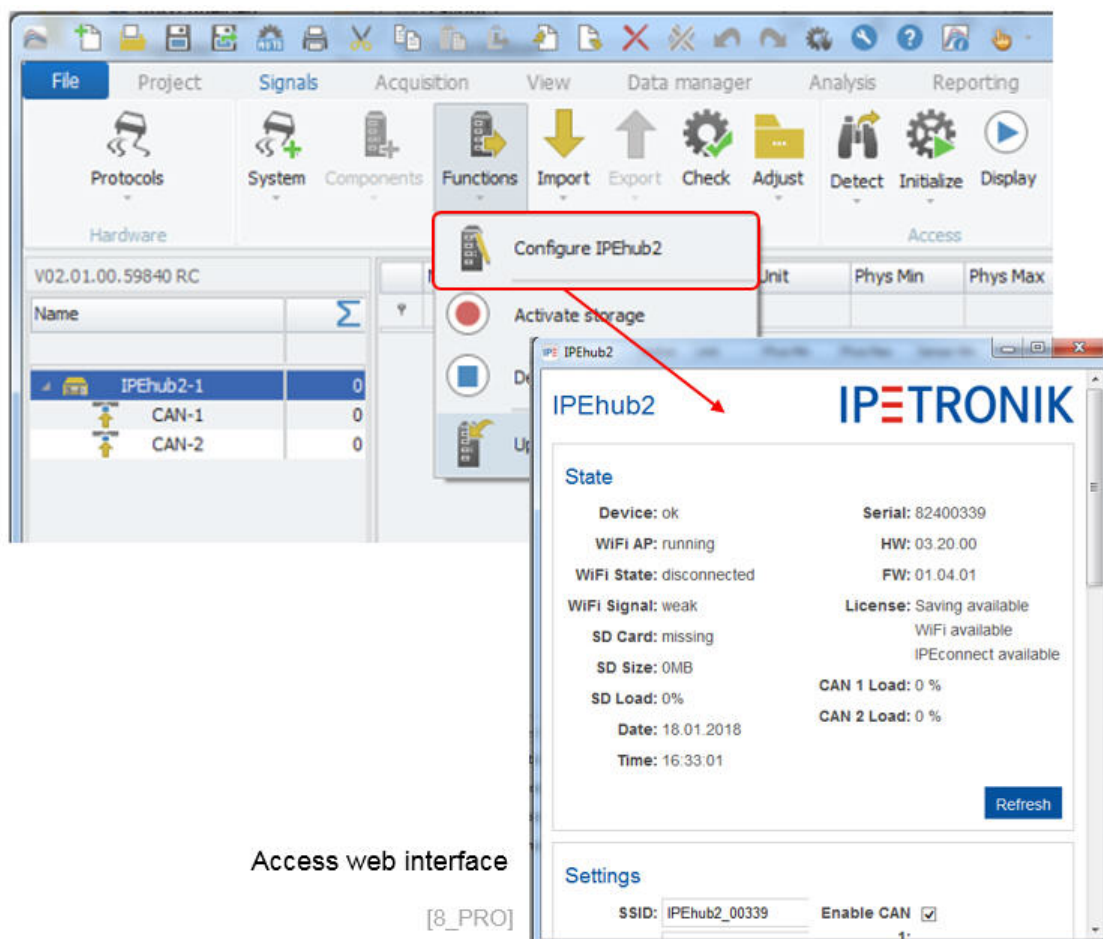


WiFi and dry config do not show update button

[7_2_PRO]

Besides the firmware update you can access the web interface to configure the unit and you can start and stop the data recording from the PlugIn. The web interface is also available from the following IP-address in your browser:

- ▶ IP-address to access web interface 192.168.232.1

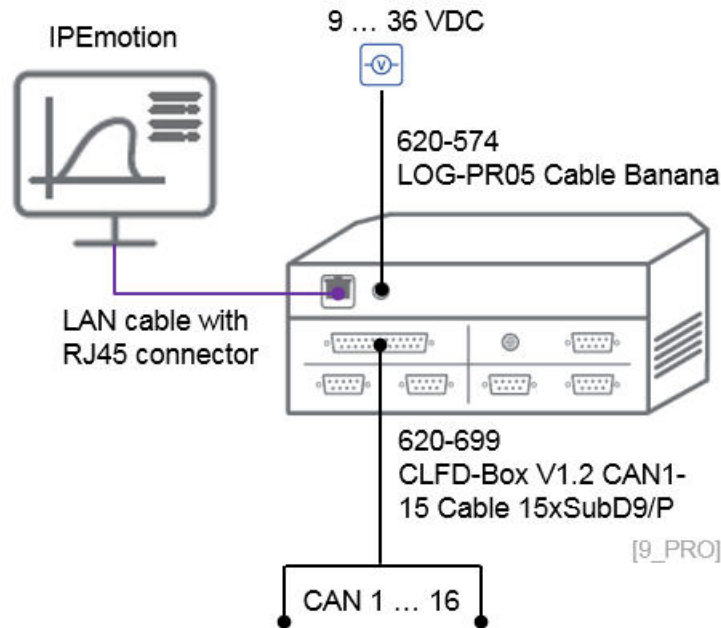


Access web interface

[8_PRO]

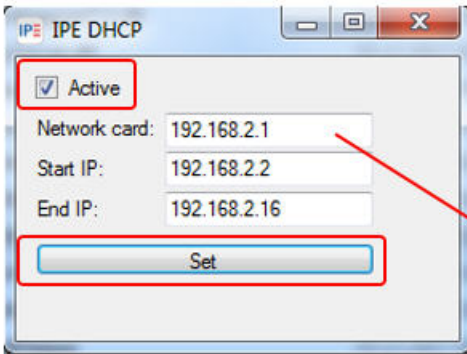
4.2 ETHgateway-CLFD specific function

The cable sets to interface ETHgateway-CLFD to the PC and bus network are presented below:



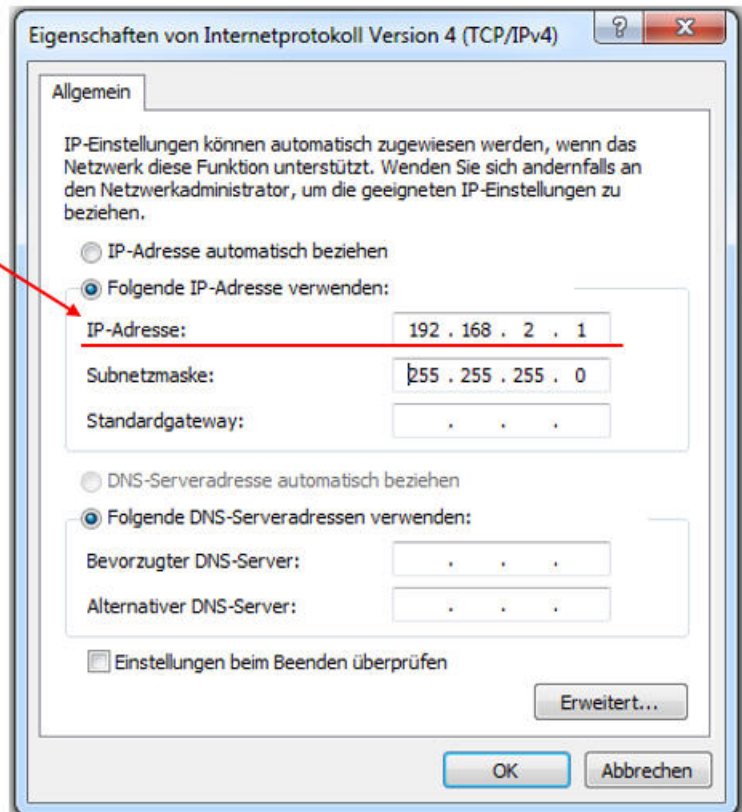
The ETHgateway CLFD is supporting an automatic hardware detect provided the IPETRONIK.IPEdhcpServerTool server tool is activated in the CAN-Server. The tool is available in the following directory.

DHCP server tool



DHCP server tool & PC network card settings

PC network card setting with fix IP-address



[10_PRO]

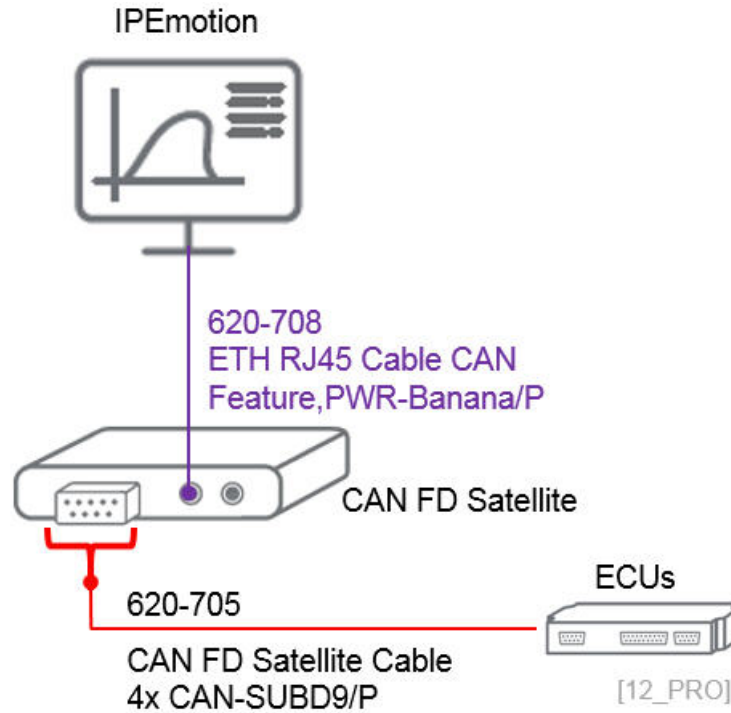
- ▶ C:\Program Files (x86)\IPETRONIK\IPETRONIK CAN-Server V01.17.xx\IPETRONIK.IPEdhcpServerTool

The ETHgateway CLFD is not supporting an internal DHCP server, therefore an external DHCP server tool is required to assign the device a fixed IP-address in order to be automatically detected by the PlugIn. You need

to activate the Server Tool by checking the Active check box and you need to use the SET operation to assign the IP-address to the ETHgateway. The SET function needs to be executed after you have defined a fixed IP-dress to your PC network card.

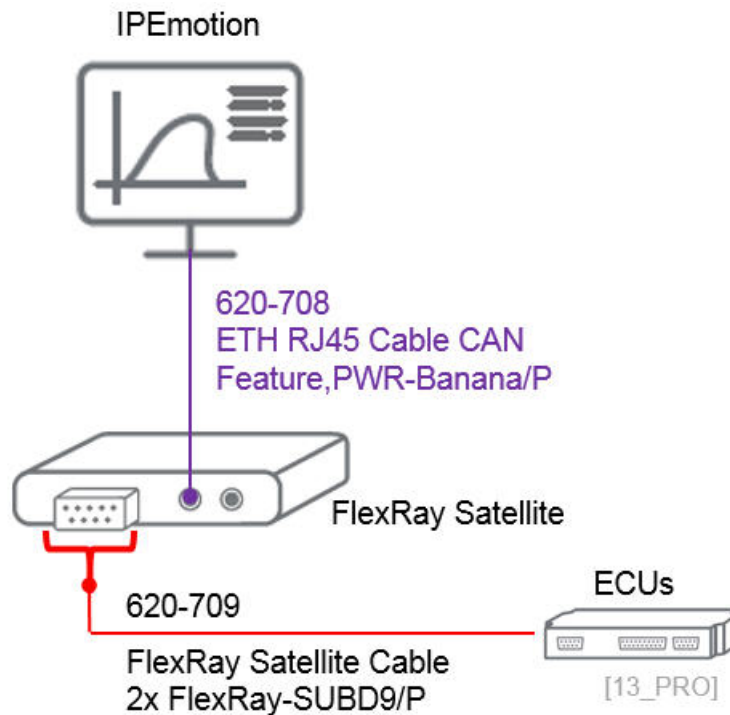
4.3 CAN FD Satellite

The cable sets to interface CAN FD Satellite to the PC and bus network are presented below:



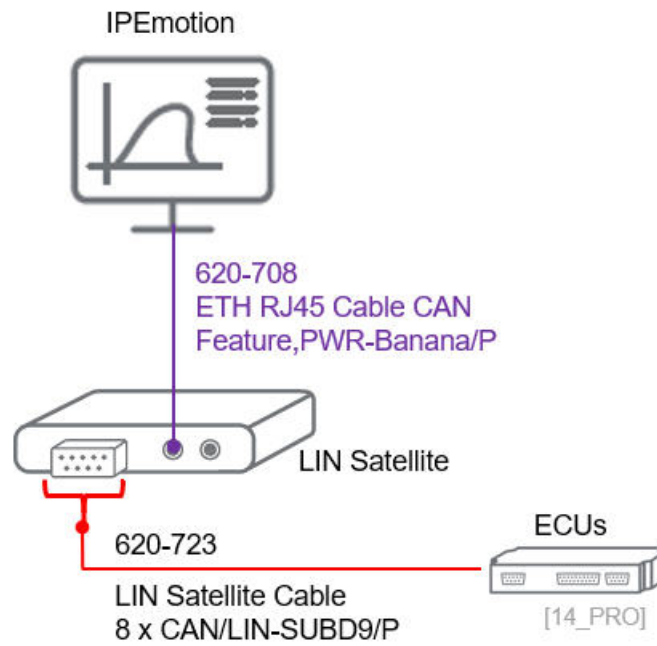
4.4 FlexRay Satellite

The cable sets to interface CAN FD Satellite to the PC and bus network are presented below:



4.5 LIN Satellite

The cable sets to interface LIN Satellite to the PC and bus network are presented below:



5 Measurements on CAN FD, LIN, ETH, FlexRay interfaces

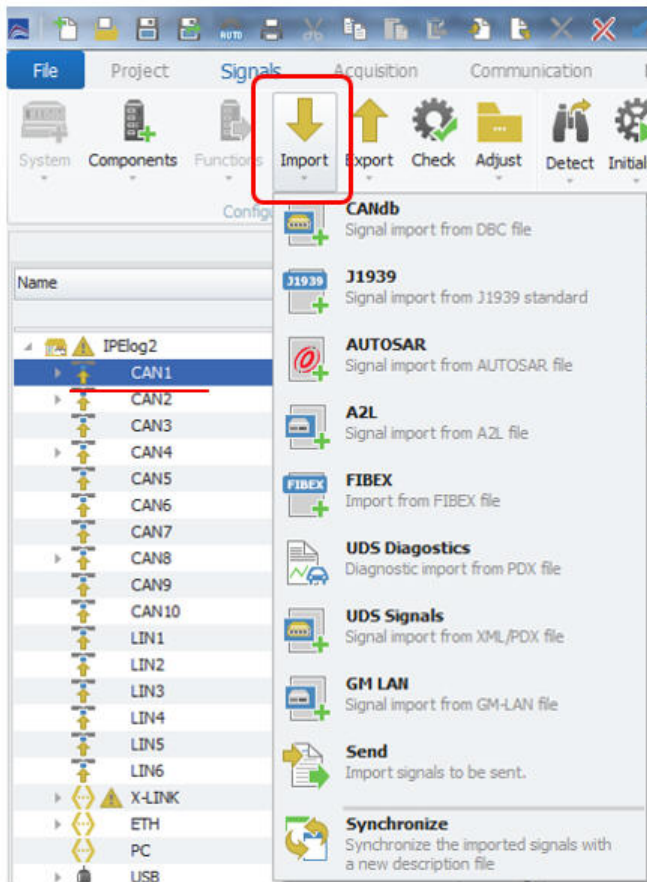
3rd Party Data Acquisition System Integration - DBC & A2L Export and Import: <https://youtu.be/dTE7ymj3ho8>

The interface hardware (logger, CAN/LIN interface or Network card) may supports or several of the following functions. The supported functions are also depending on if you are using the Protocols PlugIn of IPEmotion or a data logger. In general the data logger includes functions of Protocols PlugIn and additional features which are logger specific. Especially on the ETH connector you will see more function on the data logger

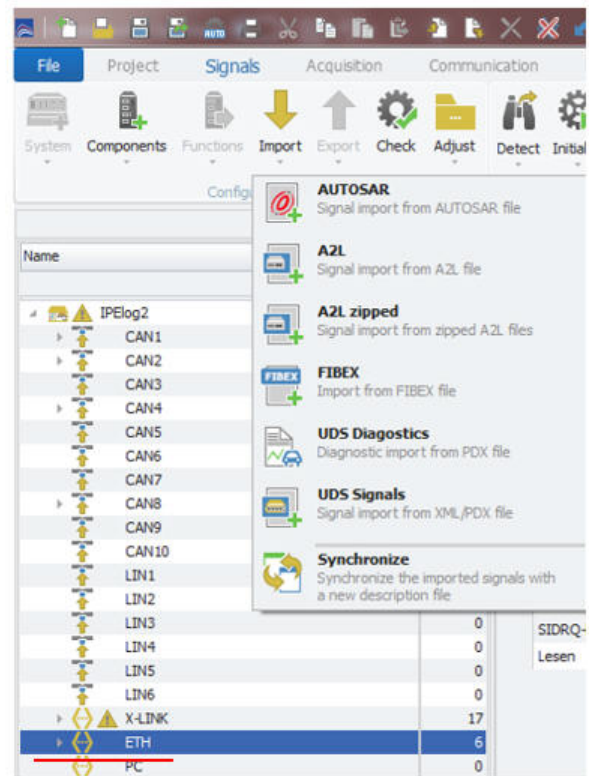
- ▶ CAN - Free running
- ▶ CAN - CCP and XCP (1.4 packed mode)
- ▶ CAN - Traffic
- ▶ CAN - J1939
- ▶ CAN FD - Free running
- ▶ CAN FD - XCP (1.4 packed mode)
- ▶ CAN FD - Traffic
- ▶ CAN FD - UDS Diagnostics
- ▶ LIN - Free running
- ▶ LIN - Traffic
- ▶ FlexRay - Free running
- ▶ FlexRay - Traffic
- ▶ FlexRay - XCP
- ▶ ETHERNET - Free running (TCP and UDP)
- ▶ ETHERNET - SOME/IP
- ▶ ETHERNET - Status and statistics
- ▶ ETHERNET - Traffic
- ▶ ETHERNET - IPETRONIK-X (Logger)
- ▶ ETHERNET - IP Cameras (Logger)
- ▶ ETHERNET - PLP devices (Probe Logger Protocol - Technica, Automotive Ethernet)
- ▶ ETHERNET - WWH-OBD
- ▶ ETHERNET - Active client - AK Protocol
- ▶ ETHERNET - IPETRONIK HV PMD-001 (high voltage and high current measurement device)
- ▶ ETHERNET - EtherCAT Master
- ▶ ETHERNET - DLT - Diagnostic Log and Trace
- ▶ ETHERNET - DoIP Diagnostics over IP
- ▶ ETHERNET - Siemens PLC TCP
- ▶ ETHERNET - Modbus TCP Master
- ▶ ETHERNET - Aerospace ARINC 429 (optional item selected during installation)

5.1 Description file import format overview

In order to measure data from your bus networks and ECUs you need to perform a description file import. The available import functions and file formats are deepening on the selected interface type. The screenshot below shows as an example import dialog and files for a CAN connector.



Description file imports on CAN



Description file imports on ETH

[IM_0]

On the CAN interface the following import formats are supported:

- ▶ CAN db (.dbc, .xml)
- ▶ Autosar (.arxml)
- ▶ A2L (.a2l) with Seed & Key .skb support
- ▶ Fibex (.xml)
- ▶ UDS (.pdx,.XML)
- ▶ GM-LAN (.odx)
- ▶ CAN-Send (switching formats into output direction)

On the LIN interface the following import formats are supported:

- ▶ CAN db (.dbc, .xml, .ldf)

On the FlexRay interface the following import formats are supported:

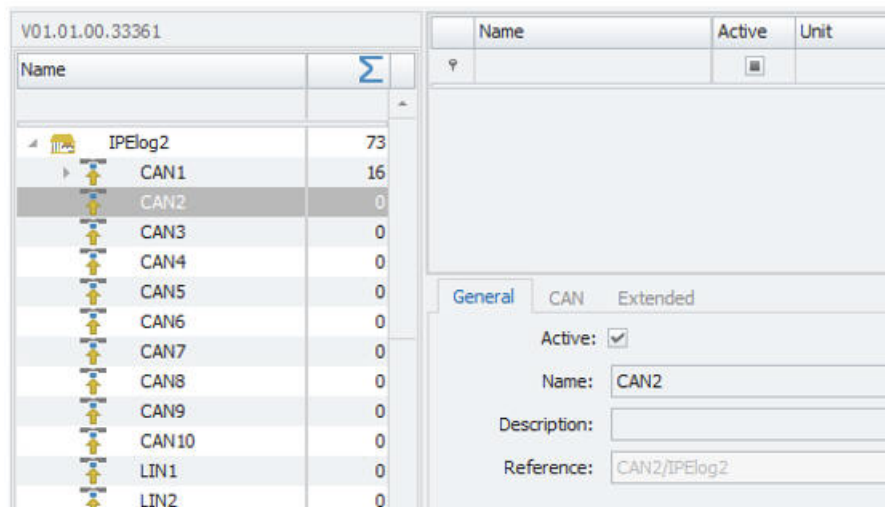
- ▶ Autosar (.arxml)
- ▶ FlexRay Parameter (.xml)
- ▶ Fibex (.xml)

On the ETHERNET interface the following imports are supported:

- ▶ Autosar (.arxml)
- ▶ A2L (.a2l)
- ▶ A2L zipped (.zip)
- ▶ Fibex (.xml)
- ▶ UDS Diagnostic (.pdx)
- ▶ UDS Signals (.pdx)

5.2 CAN interface

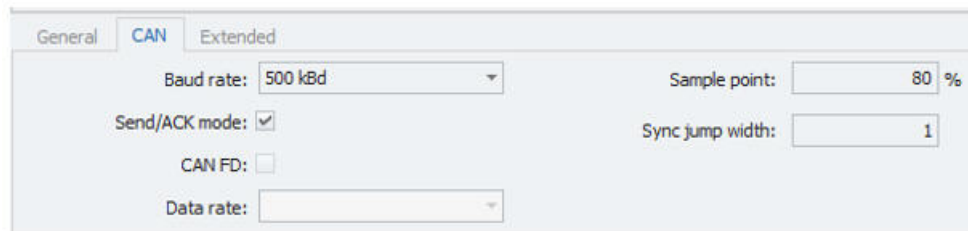
The CAN interface setting can vary in regard to the selected CAN hardware. The CAN hardware can be one of the supported vendors like VECTOR, KVASA, PEAK, etc.. or an IPETRONIK data logger. On the General interface tab sheet you have the following settings.



[IM_0_1]

- ▶ Active Activate the interface
- ▶ Name Default channel name
- ▶ Description Add an additional description to the interface
- ▶ Reference Is automatically generated by the system and stored in the data file to back reference the data source.

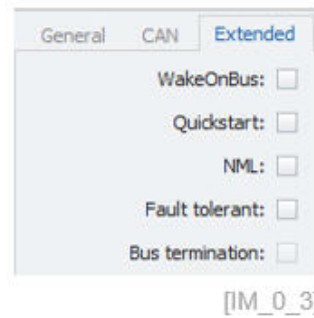
On the CAN tab sheets the following settings are supported:



[IM_0_2]

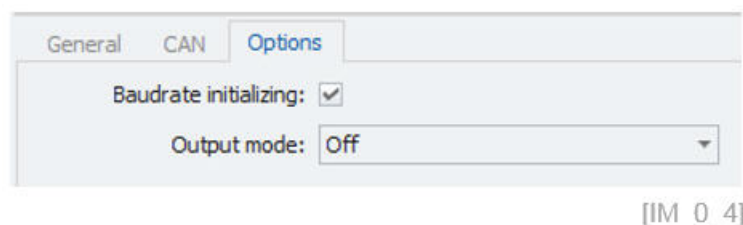
- ▶ Baud rate Here you can define the baud rate of the CAN bus
- ▶ Send ACK mode With an active check box for normal mode the CAN interface is operating in sciencence mode. The interface will not send any acknowledgement messages to the bus in order to avoid any disturbances on the bus network.
- ▶ CAN FD This check box can only be activated, when the CAN interface supports CAN FD messages which are based on 64 bit message size.
- ▶ Data rate This reference to the CAN FD data rate.
- ▶ Sample point Is relevant vor CAN FD measuremens and is usually taken from the description file during the import process.
- ▶ Sync jump with Is relevant vor CAN FD measuremens and is usually taken from the description file during the import process.

On the Extended tab sheet, the Wake on Bus (WoC) and the NML (No Message Lost) can be activated. This function is only supported by IPElog2, ETHOS, mCROS SL, CAN FD Satellite interfaces.



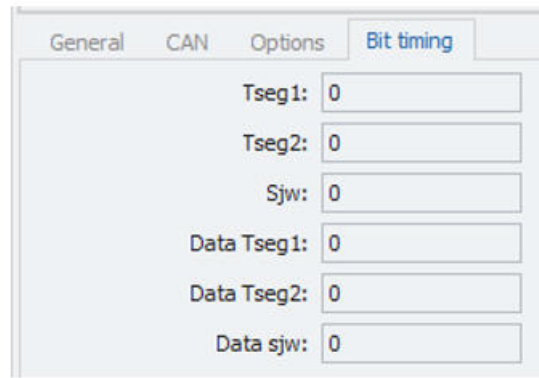
- ▶ **Wake on Bus** With an activated Wake setting the logger can be automatically started when CAN or LIN traffic is detected on the interface. In wake mode the logger power consumption is higher.
- ▶ **Quickstart** The Quickstart function records all early bus traffic until the logger is fully up and running. The configuration of Wake conditions to record CAN or LIN traffic can be separated from the DMA Quick-Start storage function, if needed. That gives the user now the ability to record quick start traffic without any mandatory Wake on Bus configurations. However, when using No Message Lost (NML) function the quick start and Wake on Bus functionality is by design included.
- ▶ **NML** No Message Lost. The Logger IPElog2 supports the No Message Lost (NML) functionality. When the logger is in NML mode the Power LED is blinking every 3 seconds. When bus traffic is received on the NML configured interface, the logger is booting up and stores all bus traffic data in the storage group. During the boot phase only CAN traffic is stored. The measurement of protocols or M-CAN / X-LINK measurement modules and other periphery devices like IP- and USB cameras, Satellite interfaces etc. is only stored in the data file, when the logger is completed booted and all systems are properly initialized.
- ▶ **Fault-tolerant** This is related CAN low speed supported e.g. by IPElog2
- ▶ **Bus termination** Activating 120 Ohm bus termination on CAN 1 / M-CAN of IPElog 2

The Option tab sheet provides the following settings.



- ▶ **Baud rate initializing** With this check box setting the baud rate of the CAN controller is updated to the setting defined on the previous CAN tab sheet.
- ▶ **Output mode** Off – no impact. Configuration – provides extended output messages in the message window about the ECU communication.

The Bit timing tab sheets is available on all CAN FD supporting interfaces.

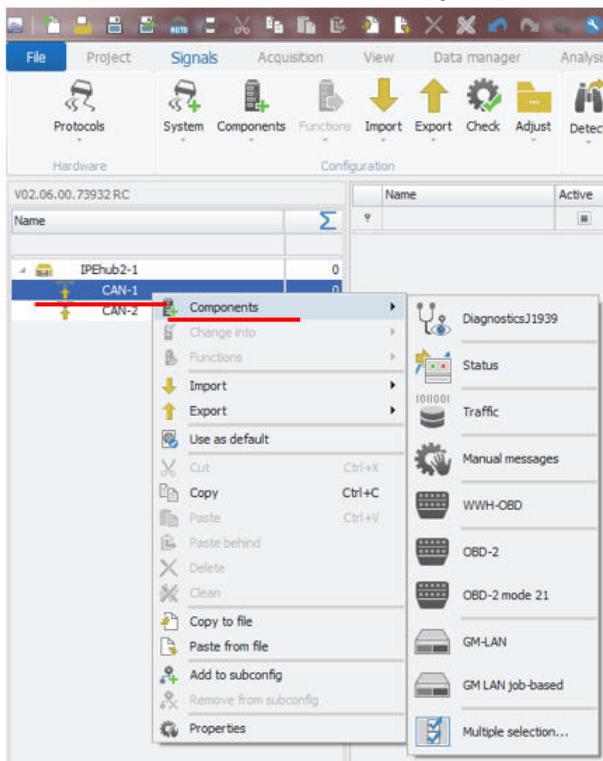


[IM_0_5]

CAN FD networks require a detailed timing setting for Tseg 1 and Tseg2, in order to capture the data from the right timing segments of the CAN FD message. For more details see the manuals of the CAN interface vendors or in the internet e.g. www.bittiming.can-wiki.info

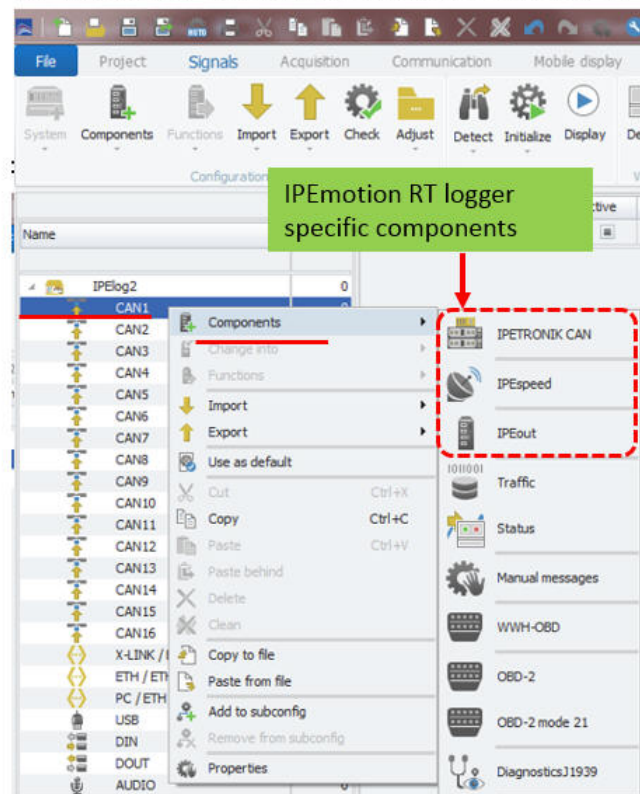
The CAN interface in an IPEmotion RT loggers support also a wider range of additional functions which will be discussed below.

IPEmotion PC: CAN interface components



IPEmotion PC: CAN interface components

IPEmotion RT: Logger CAN interface components



RT Logger: CAN interface components [IM_0_6]

- ▶ IPETRONIK CAN On this interface all M-CAN modules can be created. This function is only supported on IPEmotion RT data loggers.
- ▶ IPEspeed This interface will create all channels of the IPEspeed GPS receiver
- ▶ IPEout This interface will create all channels of the analog and digital in/out module from PEAK
- ▶ Traffic With this channel you can perform CAN bus traffic measurements.
- ▶ Status This refers to CAN interface status channels
- ▶ Manual messages Here you can create manual CAN messages.
- ▶ WWH-OBD This refers to world wide harmonized OBD measurements.
- ▶ OBD-2 With this interface a large range of on board diagnostic channel (OBD) are created which are accessible on almost all cars via the OBD connector.
- ▶ OBD-2 mode 21 This covers a special OBD mode.
- ▶ GM-LAN This covers a special GM diagnostic mode.
- ▶ GM-LAN job-based This covers a special GM diagnostic mode.
- ▶ J1939 Diagnostic messages

The overview of the CAN status channels is presented below.

V02.02.00		Name	Active	Unit
Name			<input type="checkbox"/>	
<ul style="list-style-type: none"> <ul style="list-style-type: none"> IPEhub2-1 15 <ul style="list-style-type: none"> CAN-1 15 <li style="background-color: #4a7ebb; color: white;">Status 15 CAN-2 0 				
		Process status	<input checked="" type="checkbox"/>	
		Bus load	<input checked="" type="checkbox"/>	%
		Standard messages per second	<input checked="" type="checkbox"/>	
		Standard messages	<input checked="" type="checkbox"/>	
		Extended messages per second	<input checked="" type="checkbox"/>	
		Extended messages	<input checked="" type="checkbox"/>	
		Error messages per second	<input checked="" type="checkbox"/>	
		Error messages	<input checked="" type="checkbox"/>	
		Chip status	<input checked="" type="checkbox"/>	
		Standard Remote messages p...	<input checked="" type="checkbox"/>	
		Standard Remote messages	<input checked="" type="checkbox"/>	
		Extended Remote messages p...	<input checked="" type="checkbox"/>	
		Extended Remote messages	<input checked="" type="checkbox"/>	
		Overload messages per second	<input checked="" type="checkbox"/>	
		Overload messages	<input checked="" type="checkbox"/>	

CAN interface – Status channels

[IM_0_7]

5.3 LIN interface

On the LIN tab sheets the following settings are supported:

General LIN Extended

Active:

Name: LIN1

Description: LIN port

Reference: LIN1/IPeLog2

[IM_0_10]

- ▶ Active Activate the interface
- ▶ Name Default channel name
- ▶ Description Add an additional description to the interface
- ▶ Reference Is automatically generated by the system and stored in the data file to back reference the data source.

On the LIN tab sheet the following settings are supported:

General LIN Extended

Baud rate: 9,6 kBd

LIN version: 2.0

[IM_0_11]

- ▶ Baud rate From the drop down list you can select from 3 different baud rates defines as: 2.4, 9.6, 10.417 kBaud
- ▶ LIN version The LIN version refers to the standards 1.3, 2.0 and 2.1.

In the Extended tab sheet the Wake on LIN (WoL) function can be activated. The WoL function is only supported by data loggers.

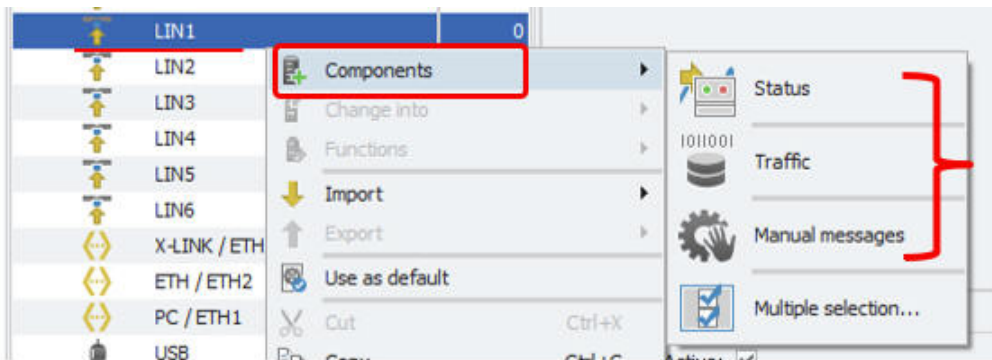
General LIN Mode

Mode Master Slave Listen

[IM_0_12]

- ▶ Mode Here you define Master, Salve or Listen mode.

The LIN interface of IPEmotion RT loggers supports also some additional functions which will be discussed below.



Additional LIN interface components

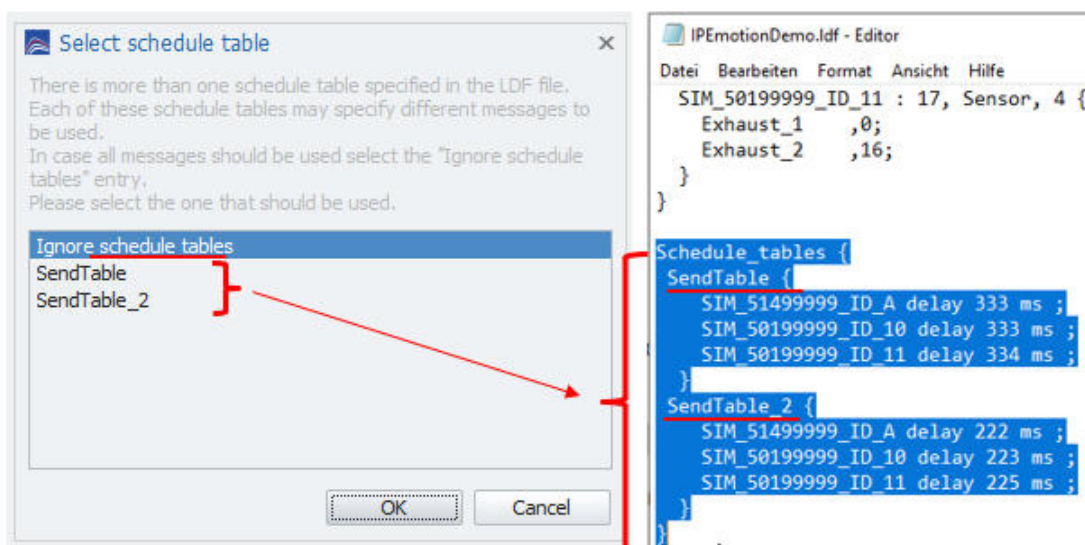
[IM_0_13]

Status channels

Name	Active	Unit
<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/> LIN1 Process status	<input type="checkbox"/>	
<input type="checkbox"/> LIN1 Bus load	<input type="checkbox"/>	%
<input type="checkbox"/> LIN1 Standard messages per s...	<input type="checkbox"/>	
<input type="checkbox"/> LIN1 Standard messages	<input type="checkbox"/>	
<input type="checkbox"/> LIN1 Error messages per second	<input type="checkbox"/>	
<input type="checkbox"/> LIN1 Error messages	<input type="checkbox"/>	

- ▶ Status Indication of the LIN interface status.
- ▶ Traffic With this channel you can perform LIN bus traffic measurements.
- ▶ Manual messages Here you can create manual LIN messages similar to CAN messages discussed above.

When the LDF import file includes multiple scheduled tables, you can select during the import process which scheduled table to use:



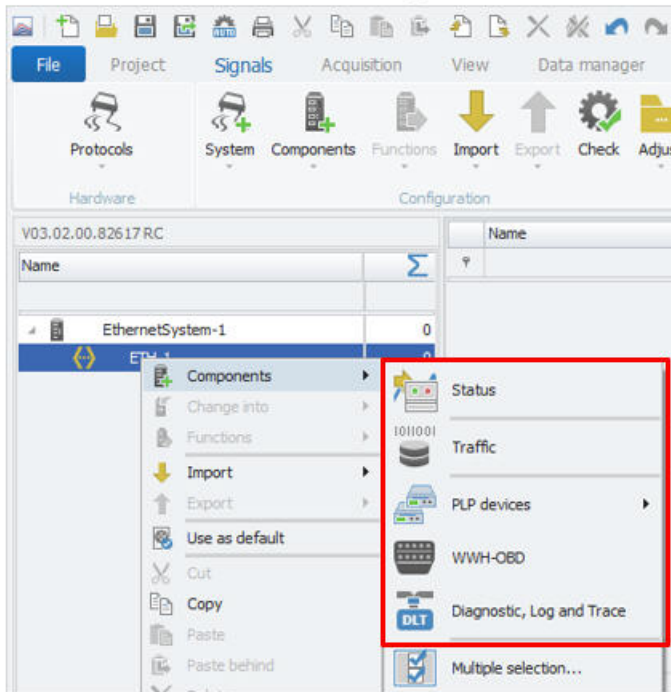
Select scheduled tables on LIN interface

[IM_0_14]

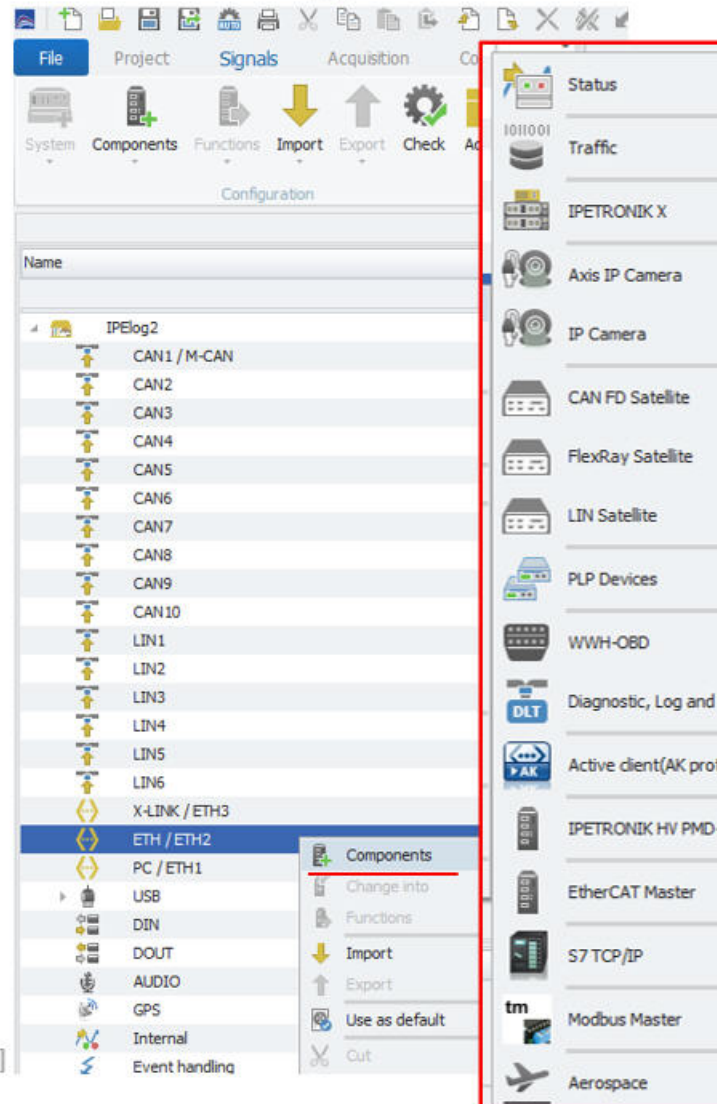
5.4 ETH interface

The ETHERNET interfaces on the Protocols PlugIn of IPEmotion PC and IPEmotion RT.UI data loggers software support different functions.

IPEmotion PC Protocols PlugIn



IPEmotion RT.UI Logger



Different functions of the ETH connector [IM_0_20]

For data loggers the default IP-addresses of the two ETH interfaces 1 and 2 are predefined. For M-LOG V3 the interfaces are named:

- ▶ ETH 1: IP. 192.168.232.1
- ▶ ETH 2: IP. 192.168.234.1

For IPElog2 the ETH interfaces are named as below:

- ▶ X-LINK/ETH3: IP. 192.168.232.1
- ▶ ETH/ETH2: IP. 192.168.234.1
- ▶ PC/ETH1: IP. 192.168.236.1

The ETH interfaces on an IPEmotion RT loggers support the following main functions:

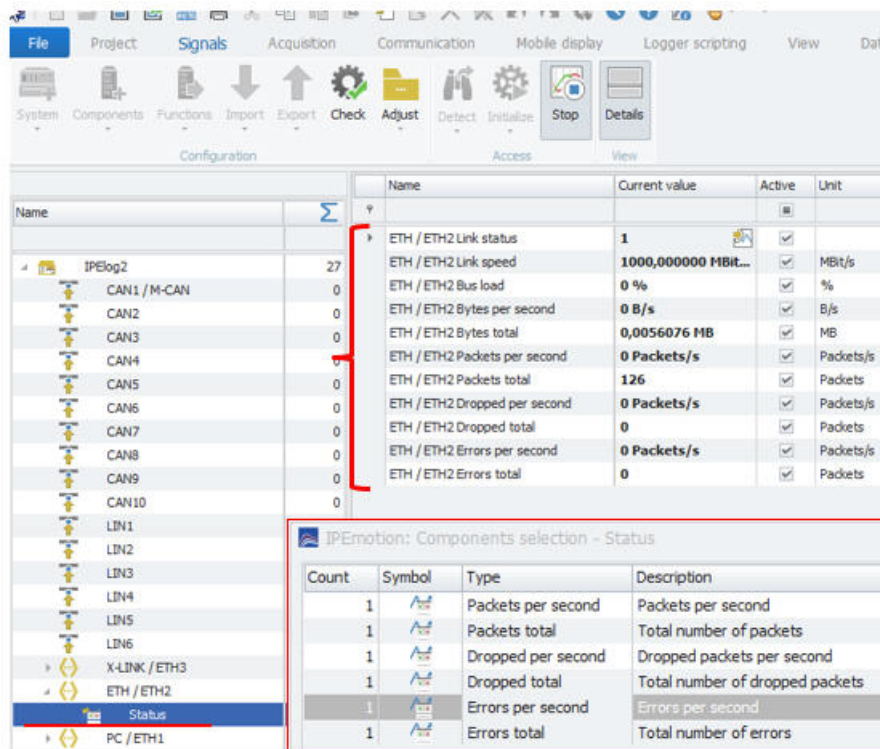
- ▶ Status Ethernet interface statistics.
- ▶ Traffic With the traffic channel you can record ETH traffic.
- ▶ IPETRONIK X On this interface all X-Modules and M-CAN in the tunneling mode can be configured. This function is only supported on IPEmotion RT data loggers only.
- ▶ IP camera AXIS Here you can create an IP-camera interface for pre-configured AXIS IP-camera system.
- ▶ IP camera Here you can create an general IP-camera interface system.
- ▶ Satellites You can add additional CAN FD, LIN, FlexRay interfaces to the logger.
- ▶ PLP Devices The Technica Automotive Ethernet devices of the latest generation CM 100 and 1000 HIGH (Capture Modules) are supported using the PLP (Logger Probe Protocol).
- ▶ WWH-OBD World Wide Harmonized-OBD.
- ▶ DLT Diagnostic Log and Trace.
- ▶ Active Client AK Protocol Interface to get data from MAHA roller benches.
- ▶ Siemens PLC TCP Here you can interface to Siemens PLC via the TCP communication based on S7 plc programs. The interface to the TIA portal is not yet supported.
- ▶ Modbus Master With Modbus PlugIn you can interface any type of external devices supporting Modbus protocol communication. The PlugIn support different function codes (01, 02, 03, 04) to read and write single and multiple registers and coils.
- ▶ Aerospace The ARINC 429 is a separate licensed protocol.

The ETH node has the following General configuration functions:

- ▶ Active Activate the interface
- ▶ Name Default channel name
- ▶ Description Add an additional description to the interface
- ▶ Reference Is automatically generated by the system and stored in the data file to back reference the data source.

5.4.1 ETH Status channels

On all ETH / LAN and PC interfaces status channel can added to monitor the interface status. This statistics about the ETH bus load, link speed, bytes, packets, error etc. . .

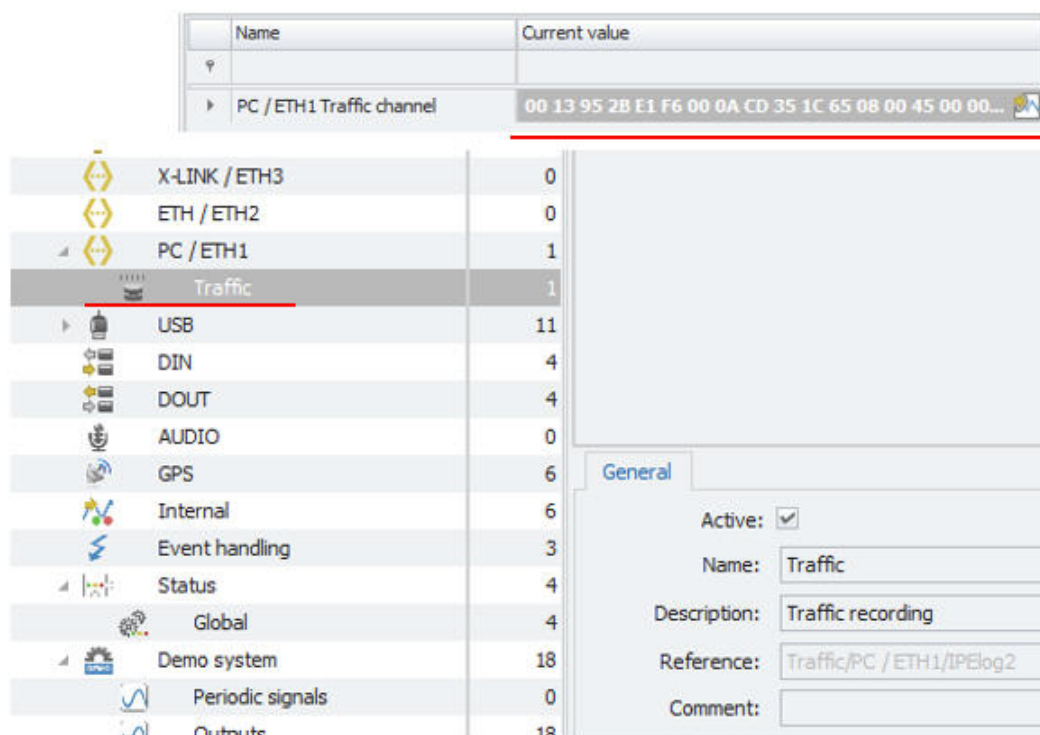


ETH Status channels

[IM_0_21]

5.4.2 ETH Traffic

For the ETHERNET Traffic measurement on the PC, you need to define the name of the LAN card of the PC which is PC dependent. This configuration is not required for data loggers where the ETH interface names are pre-defined by the hardware setup of the system.



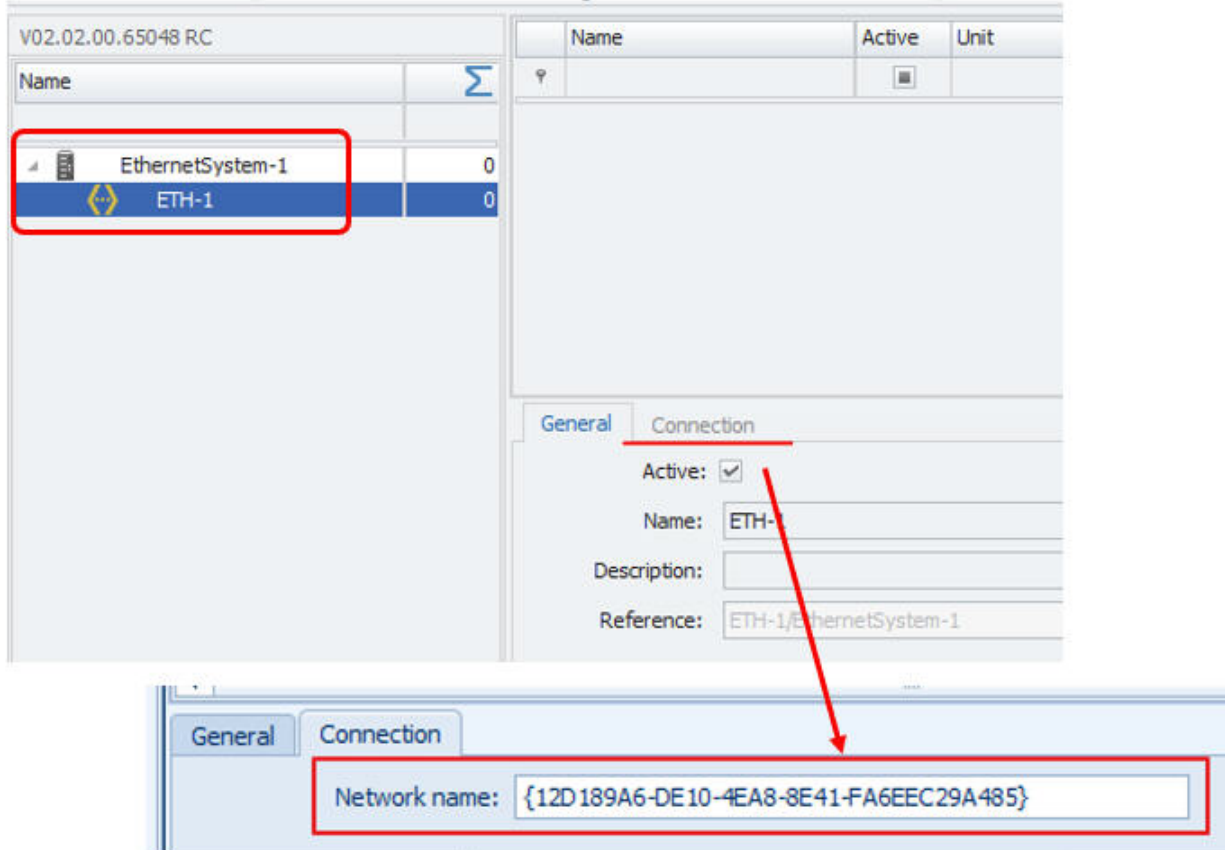
ETH Traffic

[IM_0_22]

If you like to perform Ethernet traffic measurements on the Ethernet interface of your computer, you must identify the Ethernet port name. For this you need the help of IPETRONIK support. To identify the ETH name you need to open windows PowerShell. Within the programm you run the command:

- ▶ `wmic nicconfig get ServiceName,description,macaddress,settingid`

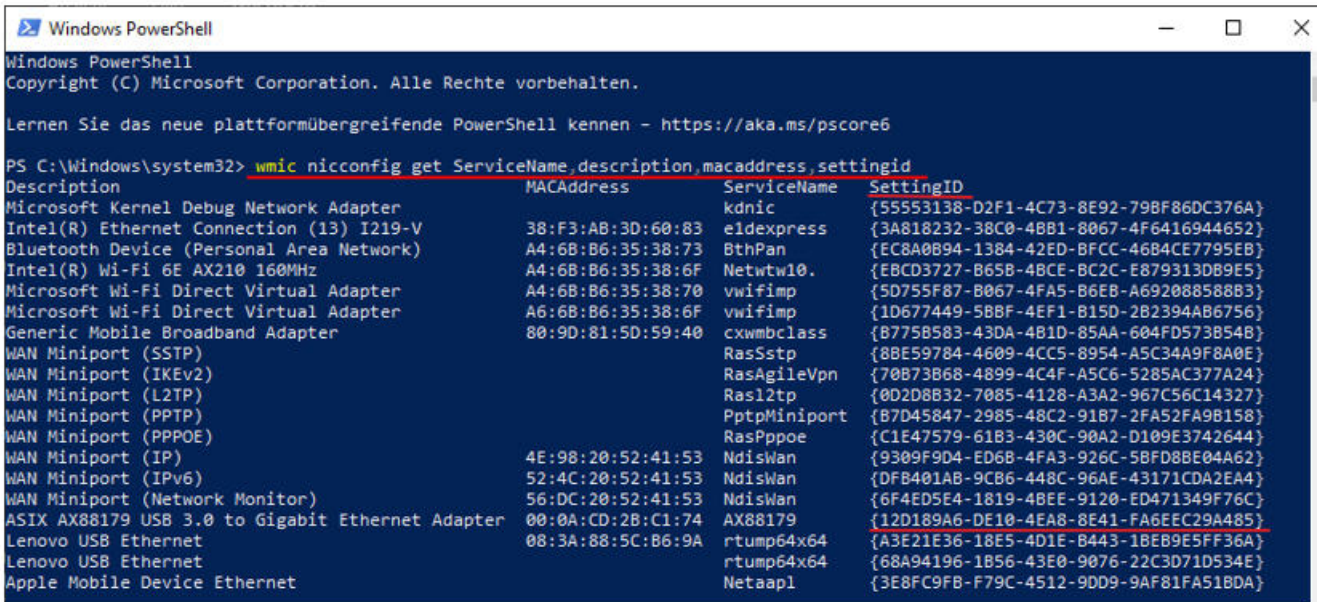
IPEmotion PC – ETH Settings of the Protocols PlugIn



PC ETHERNET network name (extracted via PowerShell) [IM_0_22_1]

The following screen shot indicates the windows PowerShell software to extract the detailed names of all Ethernet network interfaces.

Command: `wmic nicconfig get ServiceName,description,macaddress,settingid`



```

Windows PowerShell
Copyright (C) Microsoft Corporation. Alle Rechte vorbehalten.

Lernen Sie das neue plattformübergreifende PowerShell kennen - https://aka.ms/pscore6

PS C:\Windows\system32> wmic nicconfig get ServiceName,description,macaddress,settingid
Description                               MACAddress                               ServiceName                               SettingID
-----
Microsoft Kernel Debug Network Adapter   38:F3:AB:3D:60:83                       kdnic                                     {55553138-D2F1-4C73-8E92-79BF86DC376A}
Intel(R) Ethernet Connection (13) I219-V  A4:6B:B6:35:38:73                       e1dexpress                               {3A818232-38C0-48B1-8067-4F6416944652}
Bluetooth Device (Personal Area Network) A4:6B:B6:35:38:73                       BthPan                                   {EC8A0B94-1384-42ED-BFCC-4684CE7795EB}
Intel(R) Wi-Fi 6E AX210 160MHz           A4:6B:B6:35:38:6F                       Netwtw10.                                {EBCD3727-B65B-4BCE-BC2C-E8793130B9E5}
Microsoft Wi-Fi Direct Virtual Adapter   A4:6B:B6:35:38:70                       vwifimp                                  {5D755F87-B067-4FA5-B6EB-A692088588B3}
Microsoft Wi-Fi Direct Virtual Adapter   A6:6B:B6:35:38:6F                       vwifimp                                  {1D677449-5BBF-4EF1-B15D-2B2394AB6756}
Generic Mobile Broadband Adapter         80:9D:81:5D:59:40                       cxwmbclass                               {B775B583-43DA-4B1D-85AA-604FD573B548}
WAN Miniport (SSTP)                      4E:98:20:52:41:53                       RasSstp                                  {8BE59784-4609-4CC5-8954-A5C34A9F8A0E}
WAN Miniport (IKEv2)                     52:4C:20:52:41:53                       RasAgileVpn                              {70B73B68-4899-4C4F-A5C6-5285AC377A24}
WAN Miniport (L2TP)                       0D:2D:8B:32-7085-4128-A3A2-967C56C14327 Rasl2tp                                  {0D2D8B32-7085-4128-A3A2-967C56C14327}
WAN Miniport (PPTP)                       B7D45847-2985-48C2-91B7-2FA52FA9B158} PptpMiniport                            {B7D45847-2985-48C2-91B7-2FA52FA9B158}
WAN Miniport (PPPOE)                      C1E47579-61B3-430C-90A2-D109E3742644} RasPppoe                                  {C1E47579-61B3-430C-90A2-D109E3742644}
WAN Miniport (IP)                         4E:98:20:52:41:53                       NdisWan                                  {9309F9D4-ED6B-4FA3-926C-5BFD88E04A62}
WAN Miniport (IPv6)                       52:4C:20:52:41:53                       NdisWan                                  {DFB401AB-9C86-448C-96AE-43171CDA2EA4}
WAN Miniport (Network Monitor)            56:DC:20:52:41:53                       NdisWan                                  {6F4ED5E4-1819-48EE-9120-ED471349F76C}
ASIX AX88179 USB 3.0 to Gigabit Ethernet  00:0A:CD:2B:C1:74                       AX88179                                  {12D189A6-DE10-4EAS-8E41-FA6EEC29A485}
Lenovo USB Ethernet                       08:3A:88:5C:86:9A                       rtump64x64                              {A3E21E36-18E5-4D1E-8443-18EB9E5FF36A}
Lenovo USB Ethernet                       08:3A:88:5C:86:9A                       rtump64x64                              {68A94196-1B56-43E0-9076-22C3D71D534E}
Apple Mobile Device Ethernet              3E8FC9FB-F79C-4512-9DD9-9AF81FA51BDA} Netaapl                                  {3E8FC9FB-F79C-4512-9DD9-9AF81FA51BDA}

```

PowerShell software

[IM_0_22_2]

5.4.3 IPETRONIK X

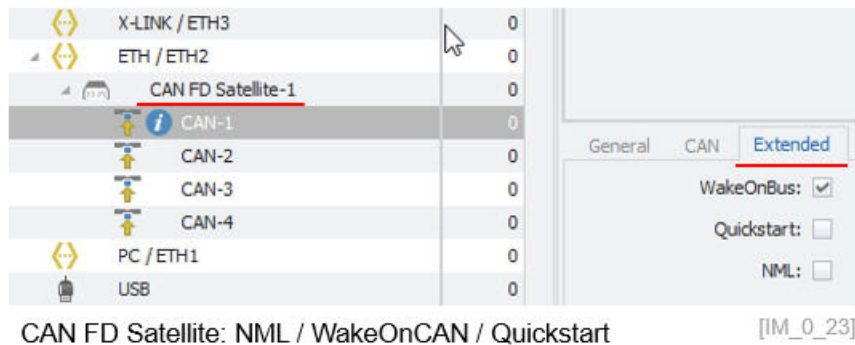
The configuration of the IPETRONIK-X Module is discussed in section ??.

5.4.4 IP Cameras

The configuration of Ip cameras is discussed in section ??.

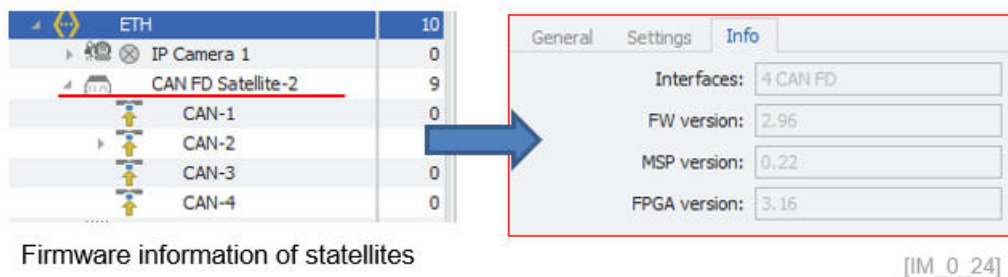
5.4.5 Satellites for CAN FD, LIN, FlexRay

The CAN FD satellite interface supports extended wake functions.

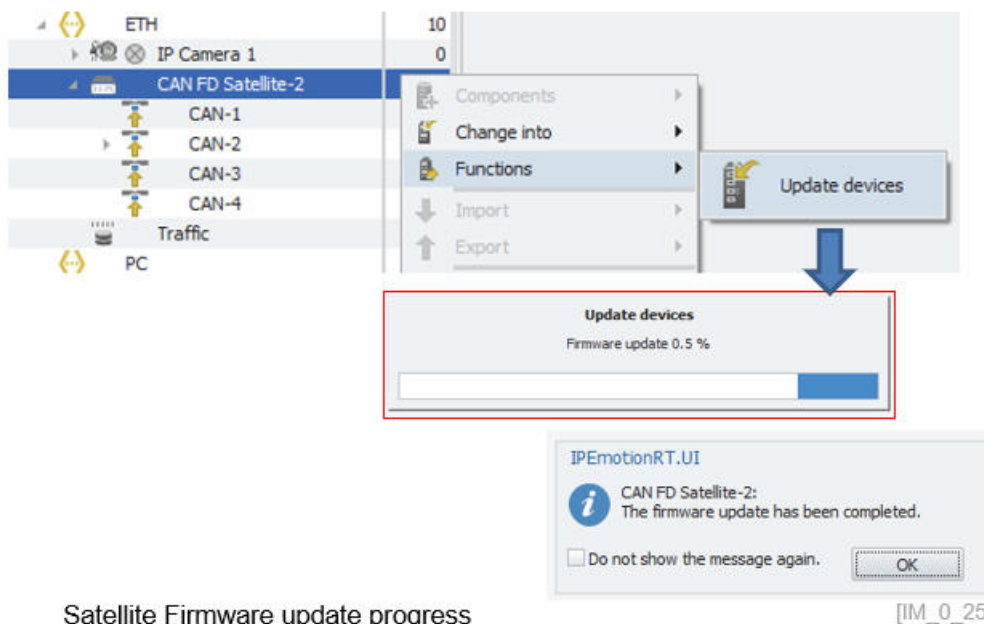


- ▶ Wake on Bus (WoC) CAN LEDs of the satellite are flashing to indicate the wake configuration.
- ▶ Quickstart Will record CAN traffic of the early measurements and buffer it inside the FPGA up to the logger is fully boote
- ▶ No Message Lost (NML) To indicate the NML configuration the CAN LED of the configured input is permanently on. NML will record all CAN traffic from the very first message.

On the ETH node also the satellite interfaces for CAN FD, LIN, FlexRay are detected. Device information about the firmware is available.



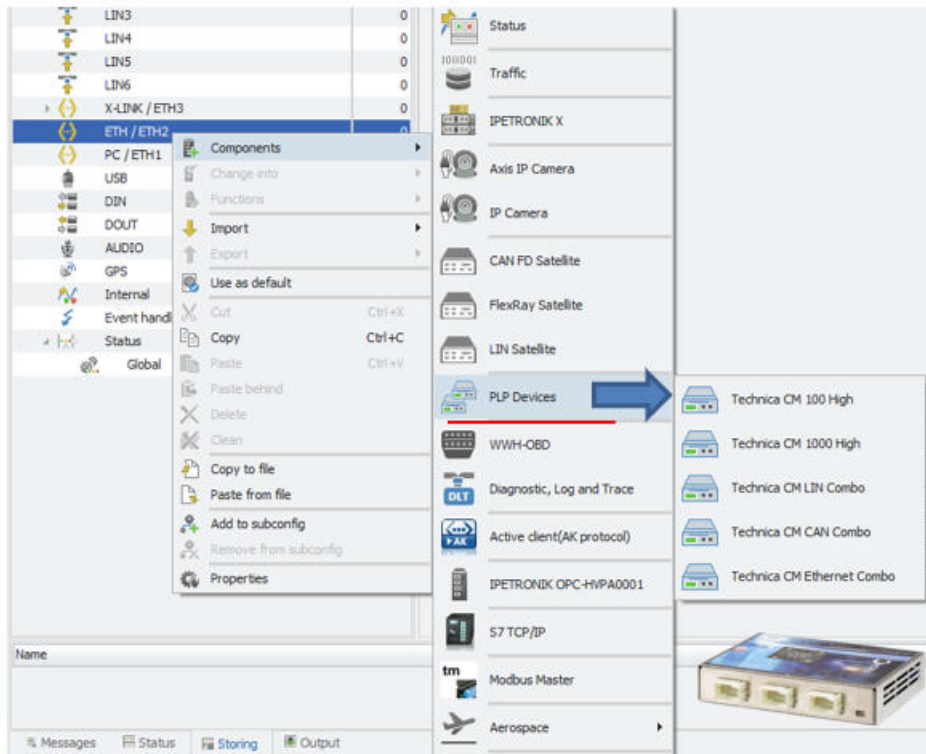
With the update function the device firmware can be updated with progress bar information.



5.4.6 PLP Devices - Automotive Ethernet

To measure automotive Ethernet networks the following interfaces from Technica Engineering are supported:

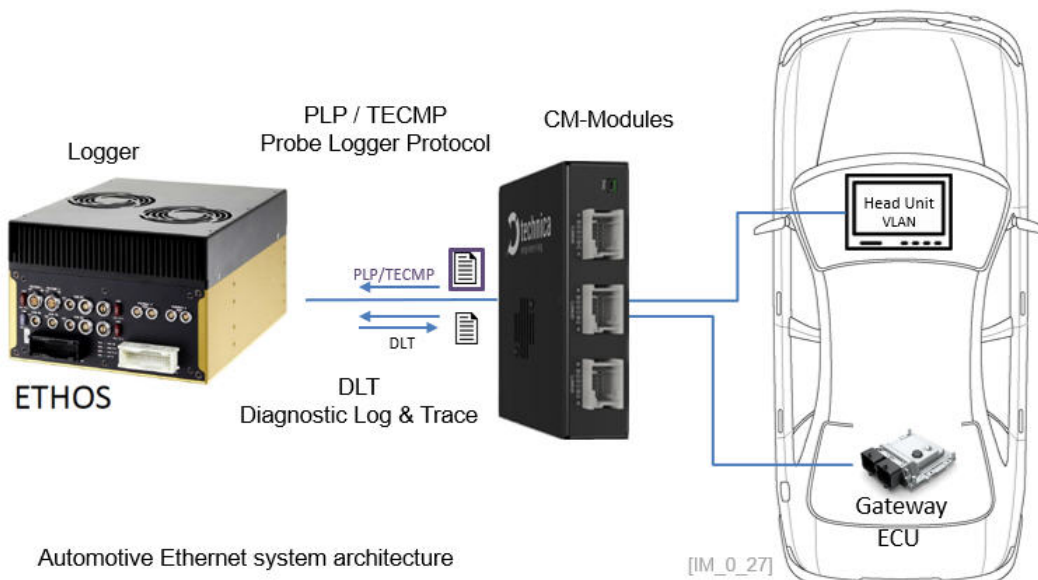
- ▶ Technica CM 100 High
- ▶ Technica CM 1000 High
- ▶ Technica CM LIN Combo
- ▶ Technica CM CAN Combo
- ▶ Technica CM Ethernet Combo



PLP Devices – Capture modules

[IM_0_26]

The system architecture below provides an overview of the ETHOS and other loggers with Ethernet ports can be used to store the automotive Ethernet data.

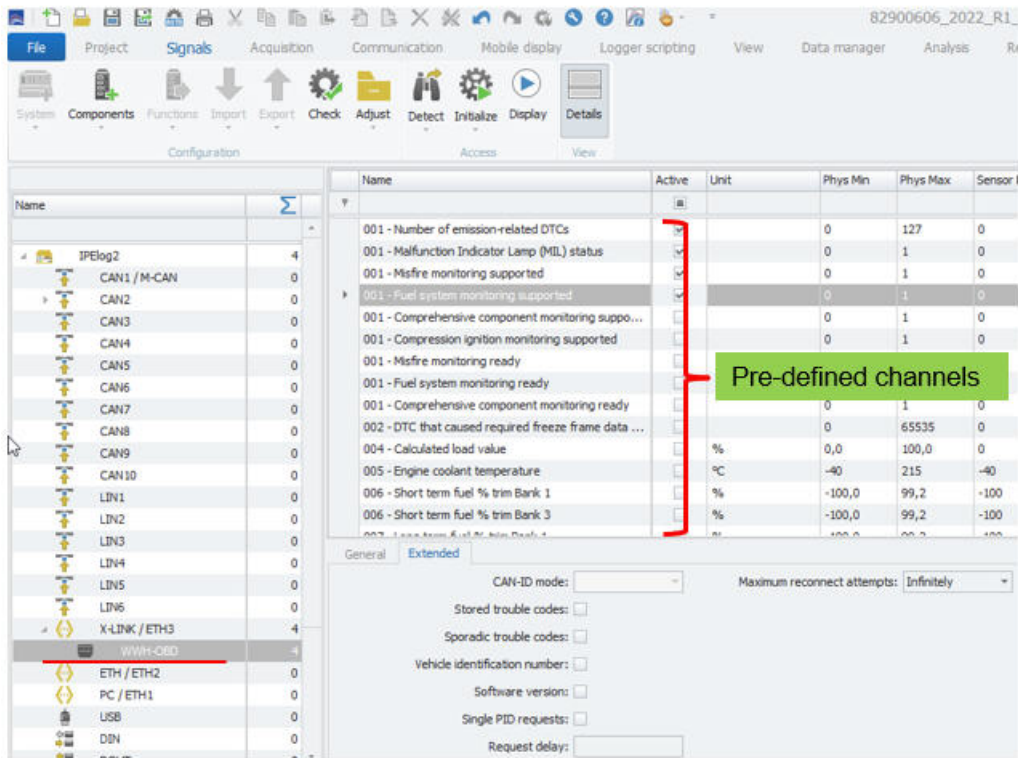


Automotive Ethernet system architecture

[IM_0_27]

5.4.7 WWH-OBD

The WWH-OBD (World Wide Harmonized on Board Diagnostics) protocol is now supported on the ETH connector. The protocol is also supported on CAN.



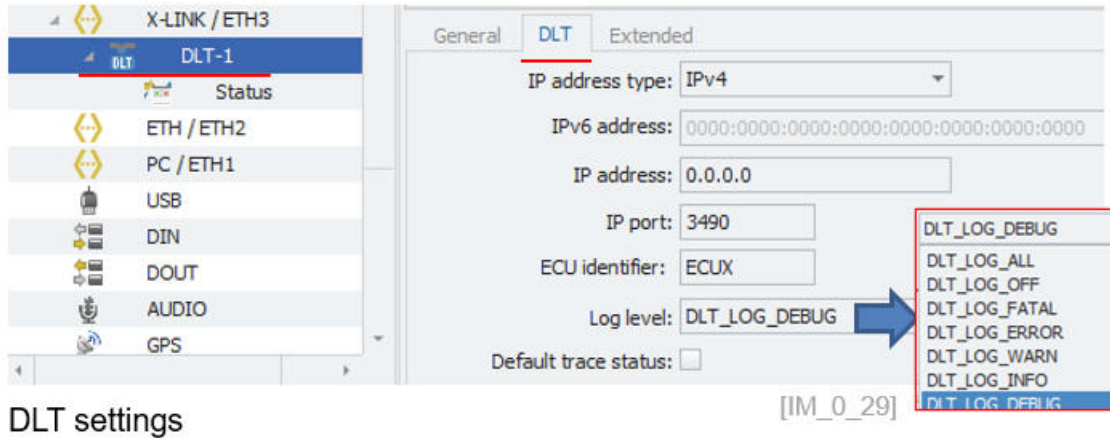
WWH-OBD

[IM_0_28]

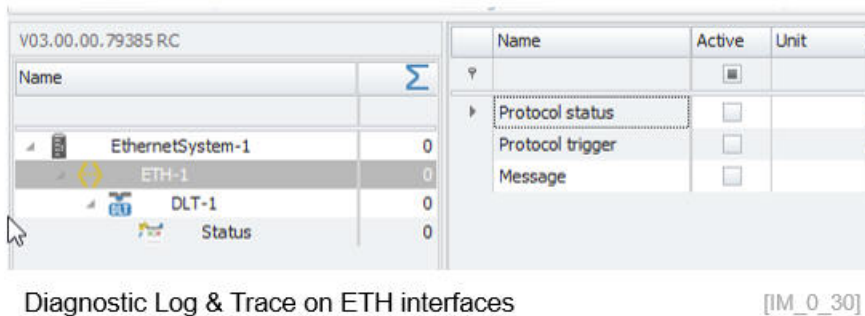
5.4.8 DLT Diagnostic Log and Trace

On the Ethernet ports of the logger and the PC using the PROTOCOLS PlugIn the diagnostic log and trace protocol from AUTOSAR is supported. The DLT protocol is used in the development phase of an ECU. It is assumed to use an external logging- and tracing tool to store the debug information generated by the ECU.

On the DLP interface node the ECU connection IP-address settings and the log level is defined. For the IP-address setting you can chose between IPv4 and the IPv6 standard.

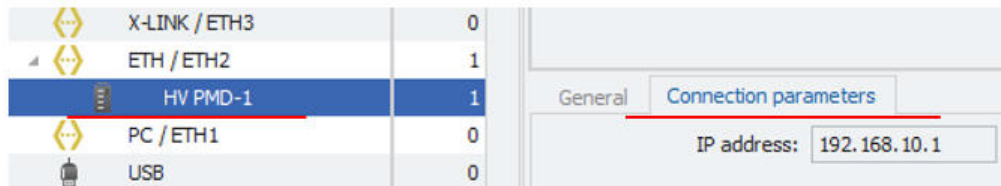


For the measurement, 2 status channels for the communication status and the start trigger are provided. The message channel will record the log information.



5.4.9 IPETRONIK HV PMD-1

On the ETH interface of the data logger a high voltage power analyzer is supported. To establish a communication to the device the IP-address has to be defined. The default IP-address of 192.168.10.1. If you like to change the default IP you need to use the DEWESoft X software at this stage of implementation.



High Voltage Power Measurement Device

Name	Active	Unit	Phys Min	Phys Max	Sensor Min	Sensor Max	Sampling rate
Voltage	<input checked="" type="checkbox"/>	V	-2000,00	2000,00	-2000	2000	1 MHz
Current	<input checked="" type="checkbox"/>	A	-2000,00	2000,00	-2000	2000	1 MHz

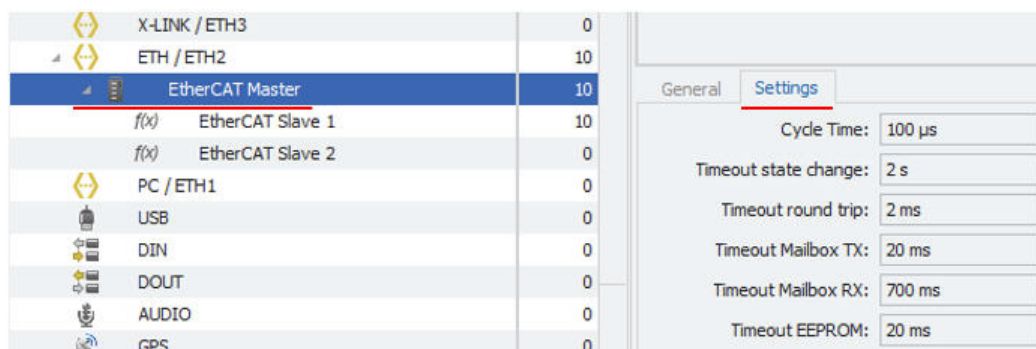
Measurement channels (Volt & Current)

[IM_0_31]

The measurement range is up to 2000 V and 2000 A for AC and DC signals.

5.4.10 EtherCAT Master

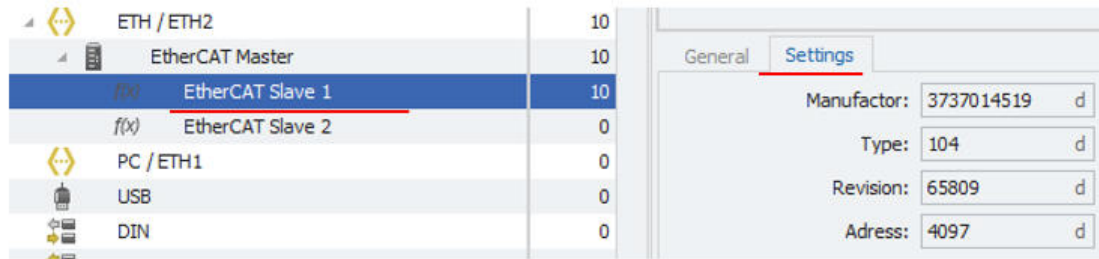
On the Ethernet Interfaces of IPElog 2, Arcos 1.5 and ETHOS data logger a EtherCAT Master is supported. You can only create one EtherCAT Master for the entire logger. Multiple Masters are at this stage of the implementation are not supported.



EtherCAT Master settings

[IM_0_31_1]

When the master interface is created on master node level the communication parameters must be defined. The communication settings are default values and might be changed in reference to the actual interface module, coupler or gateway which is acting as the inroad to the actual IO modules. After the master is created you add the actual IO modules you like to read and write data from. The actual slave address settings are obtained from the EtherCAT configuration software of the hardware vendor. Important parameters are the manufacturer ID, and the device address.

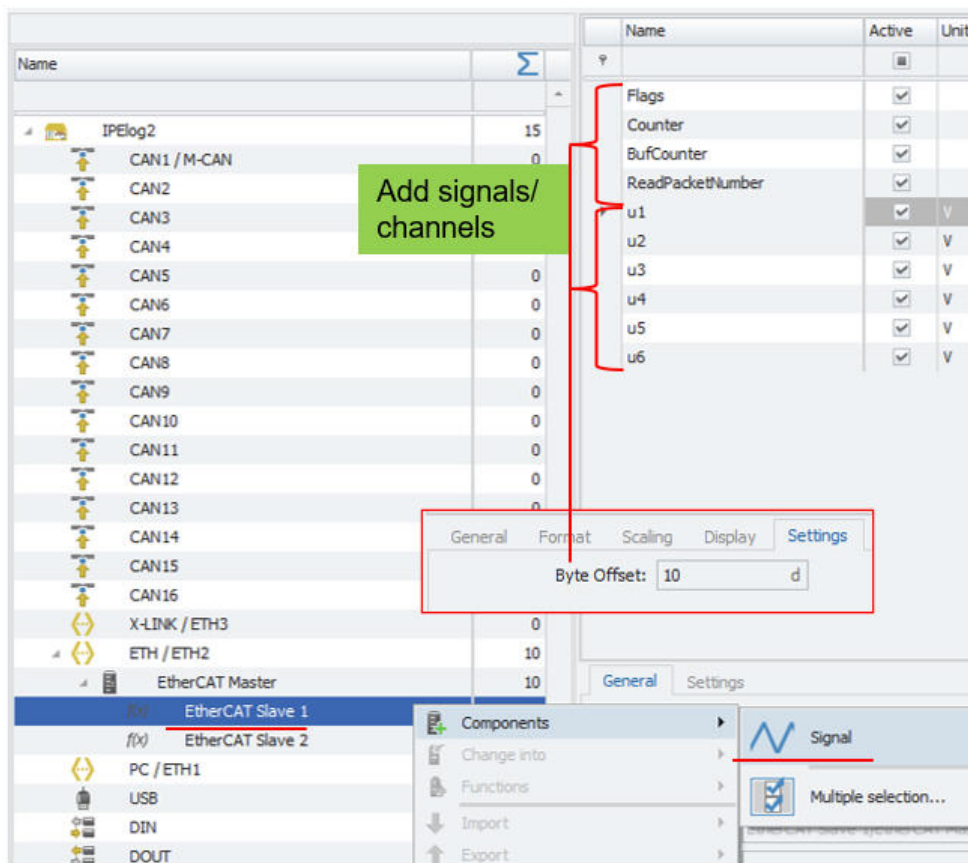


EtherCAT Slave device settings

[IM_0_31_2]

Information about the byte offset can be found in the vendor specific configuration software.

After having created the slave nodes, the corresponding measurement channels are created. You create as many channels as you like to read data from the IO module. In the example below the IO module supports 6 physical analog inputs and 4 additional internal software channels. For each channel the byte offset is defined. Information about the byte offset can be found in the vendor specific configuration software.

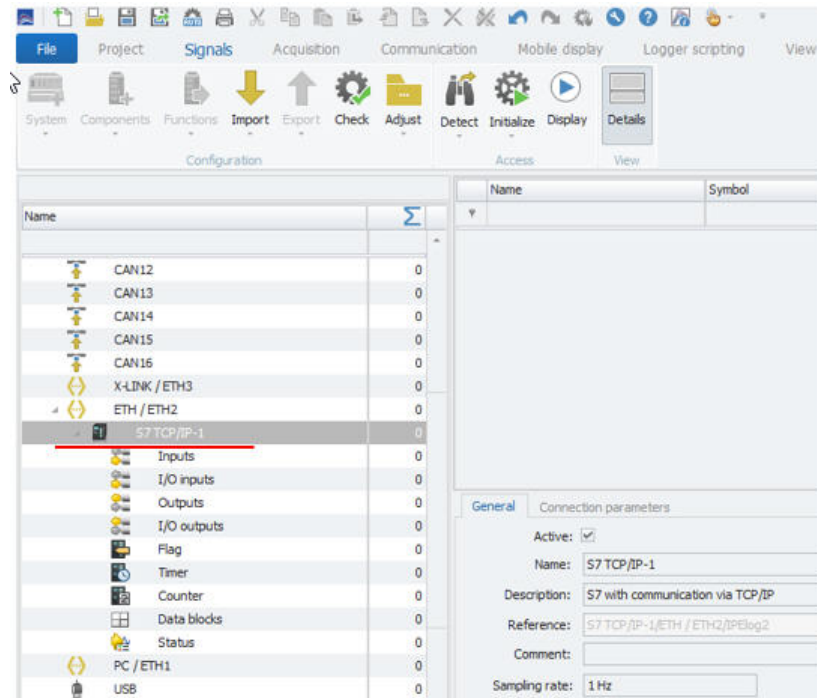


EtherCAT Signal / Channel – define byte offset

[IM_0_31_3]

5.4.11 SIEMENS S7 PLC

The Siemens S7 PLC PlugIn from the IPEmotion PC software was implemented for the data logger ETH interfaces too. For the communication to the PLC via TCP/IP protocol is available. You create a PLC interfaces node and can then start the communication to PLC. The interface configuration is more detailed specified in the Siemens PLC PlugIn manual.



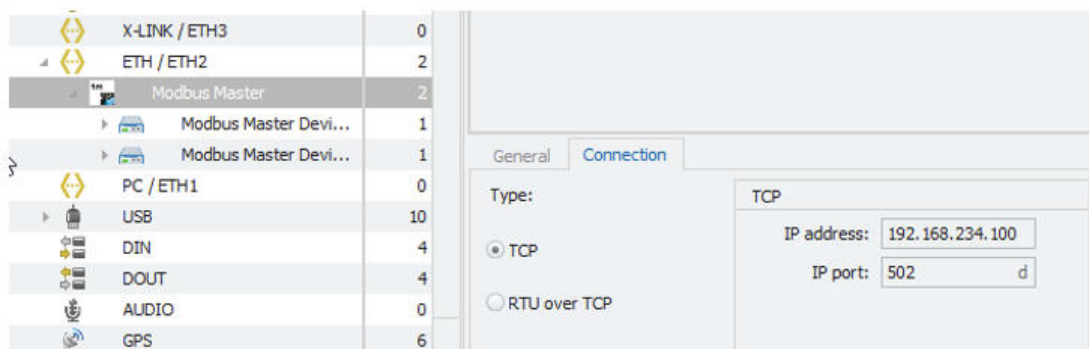
ETH: S7 PLC Interface

[IM_0_33]

5.4.12 Modbus TCP

The Modbus TCP protocol on Ethernet and Modbus RTU on the USB port of the logger is supporting the following function codes:

- ▶ Function Code 1 - Read Coil Status (Add. 0xxxx)
- ▶ Function Code 2 - Read Input Status (Add. 1xxxx)
- ▶ Function Code 3 – Read Holding Registers (Add. 4xxxx)
- ▶ Function Code 4 - Read Input Registers (Add. 3xxxx)

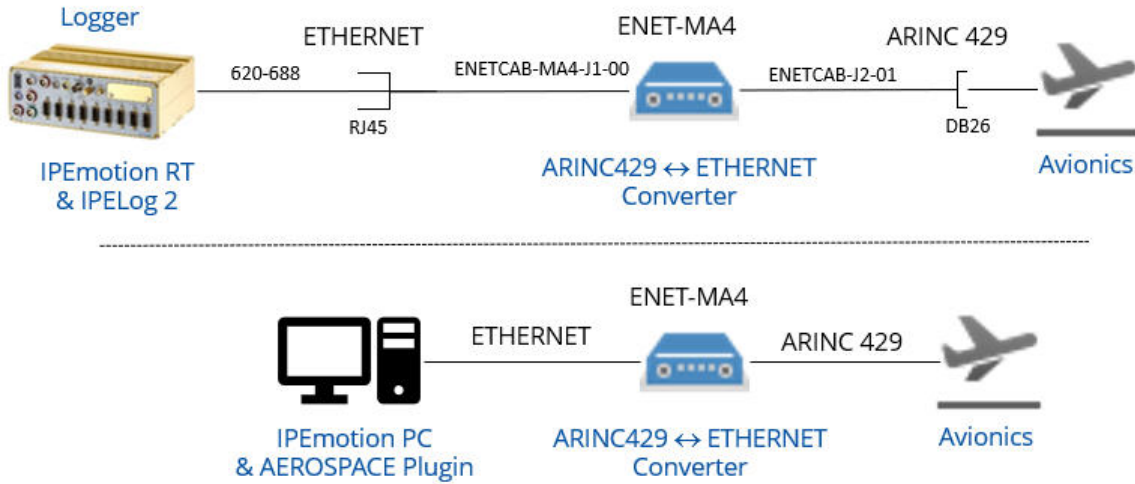


Modbus TCP

[IM_0_34]

5.4.13 Aerospace ARINC 429

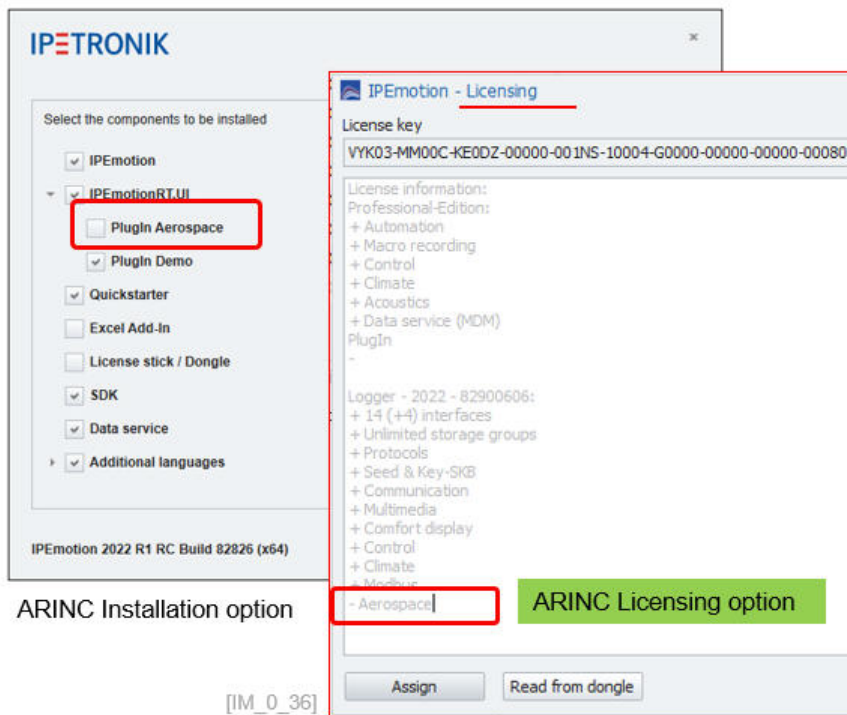
The ARINC 429 protocol is implemented on all Ethernet interfaces of the data loggers and also available in the Aerospace PlugIn for IPEmotion PC measurement software. To measure data an ARINC to Ethernet converter is required. At this stage of implementation, the eNet-MA4 converter is implemented. The following system architectures using a data logger or the measurement software are supported.



System architecture

[IM_0_35]

For data loggers the ARINC protocol is a sperate PlugIn installation options which must be activated in the installation process. However also a separate license is required too.

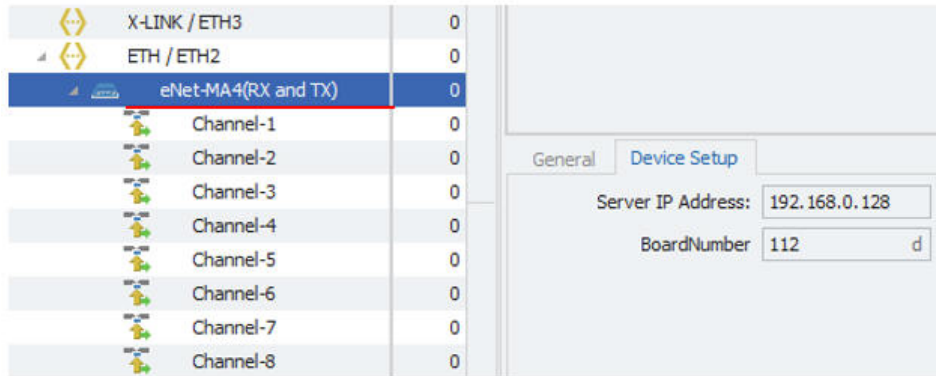


ARINC Installation option

ARINC Licensing option

[IM_0_36]

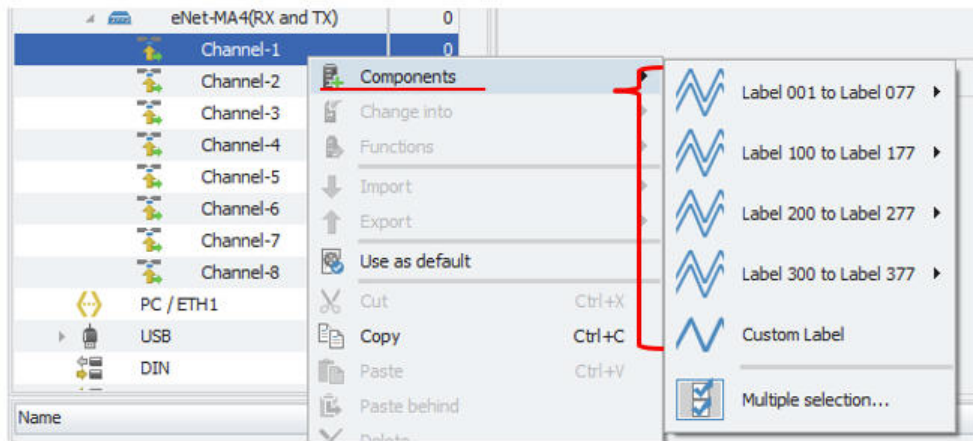
When the PlugIn is installed and activated on the data logger, you can create a standard interfaces template for the NET-MA4 interface converter.



ARINC 429

[IM_0_37]

On channel level you can then create the labels you require for measurement.

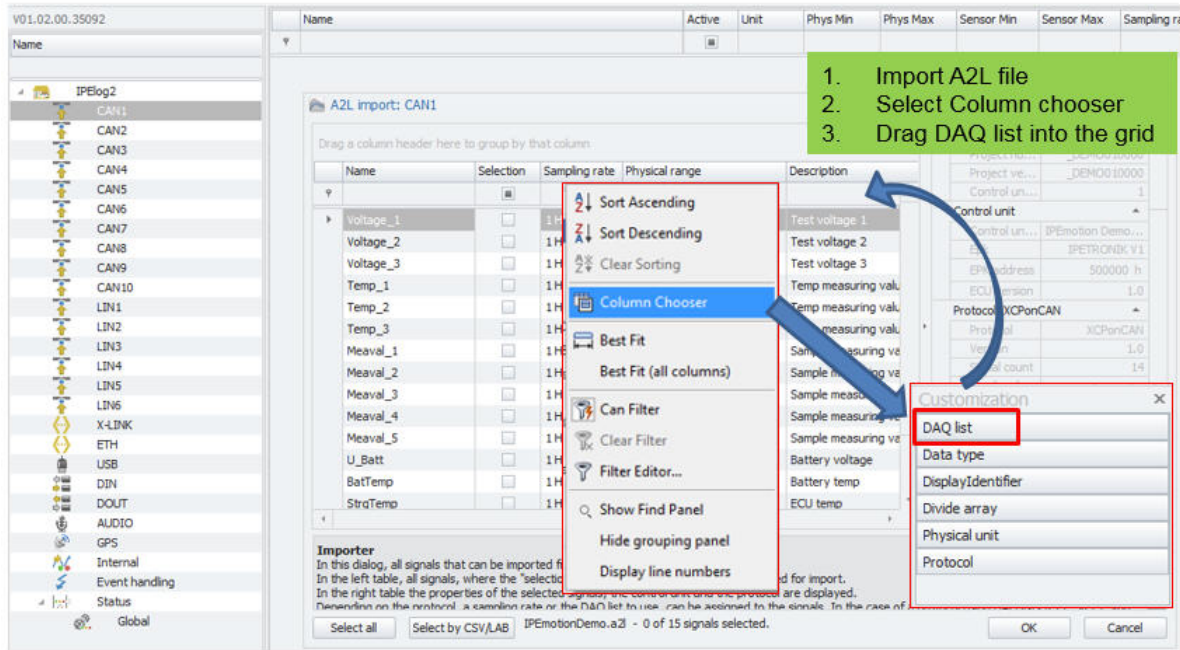


ARINC 429 Labels

[IM_0_38]

5.5 A2L import - DAQ list with graphical filling level indication

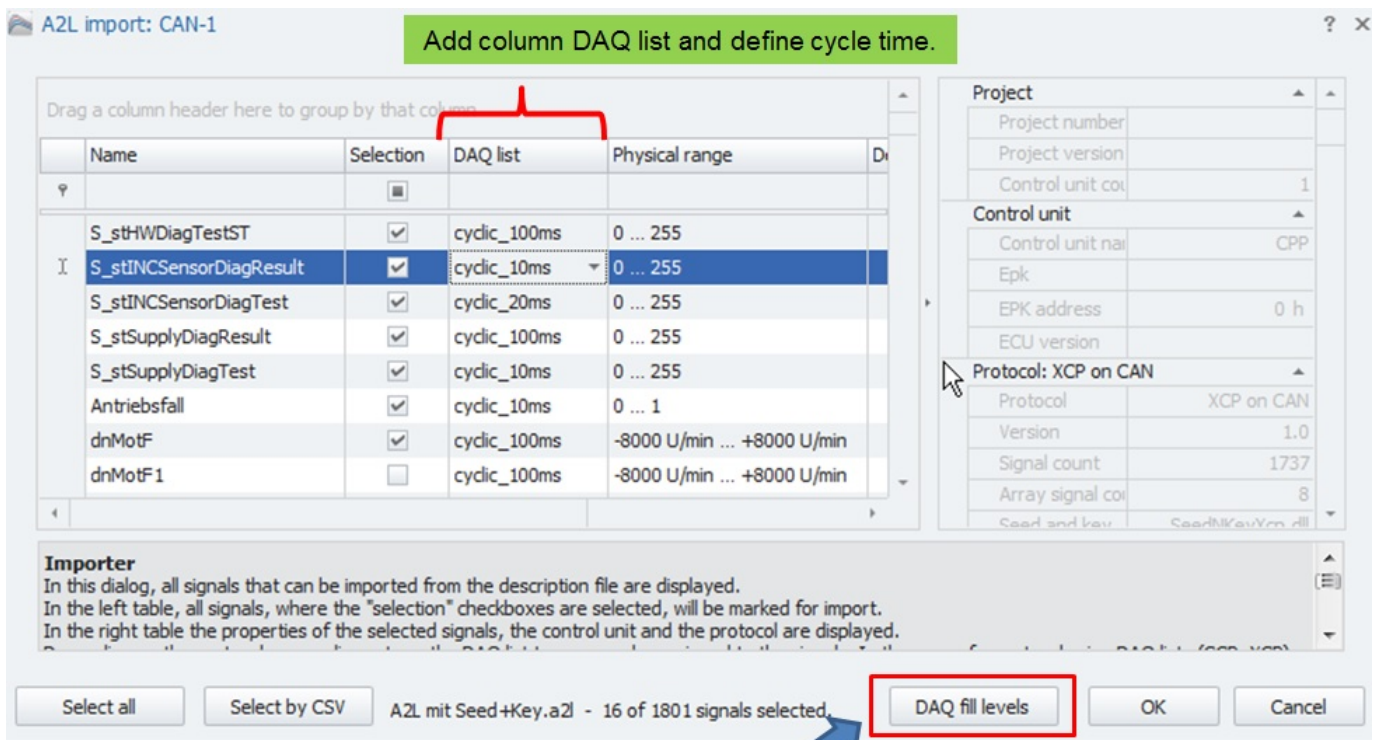
Measurements on ECUs can easily reach the performance limits if many measurements are required. With this graphical import and filling level indication overview you can now clearly identify which signals are measured and which signals are rejected. To activate the DAQ list filling level indication, you have to add the DAQ list from the column chooser to your channel grid as shown below.



Import A2L – add DAQ List

[IM_1]

Then you will see a new button to open the graphical DAQ filling level indication.



New button is visible to show fill level.

[IM_2]

When you open the graphical filling level indication you will see how the signals are allocated to the Data Transfer Objects (DTO). The number of supported DTO's is defined by the A2L file. In one DTO row you can have several signals. The color is randomly selected and is a visual aid showing how many byte a signal is utilizing from a DTO.

Drag a column to group by that column

Name	Selection	DAQ list	Physical range
S_stFRMessageDiagTest	<input checked="" type="checkbox"/>	cydic_100ms	0 ... 4294967295
S_stFRSignalDiagResult	<input checked="" type="checkbox"/>	cydic_100ms	0 ... 4294967295
S_stFRSignalDiagTest	<input checked="" type="checkbox"/>	cydic_100ms	0 ... 4294967295
S_stFRTimeoutDiagResult	<input checked="" type="checkbox"/>	cydic_100ms	0 ... 4294967295
S_stFRTimeoutDiagTest	<input checked="" type="checkbox"/>	cydic_100ms	0 ... 4294967295
S_stHWDiagResultST	<input checked="" type="checkbox"/>	cydic_100ms	0 ... 255
S_stHWDiagTestST	<input checked="" type="checkbox"/>	cydic_100ms	0 ... 255
S_stSupplyDiagResult	<input checked="" type="checkbox"/>	cydic_100ms	0 ... 255
dnMotF	<input checked="" type="checkbox"/>	cydic_100ms	-8000 U/min ... +8

9 signals

100 ms DAQ list

DAQ list fill level visualization

Automatically allocated to 4 DTO's

Fill process
>Left to right
>Row by row
Depending on signal size.

DTO (Data Transfer Objects rows 1 ... 251)

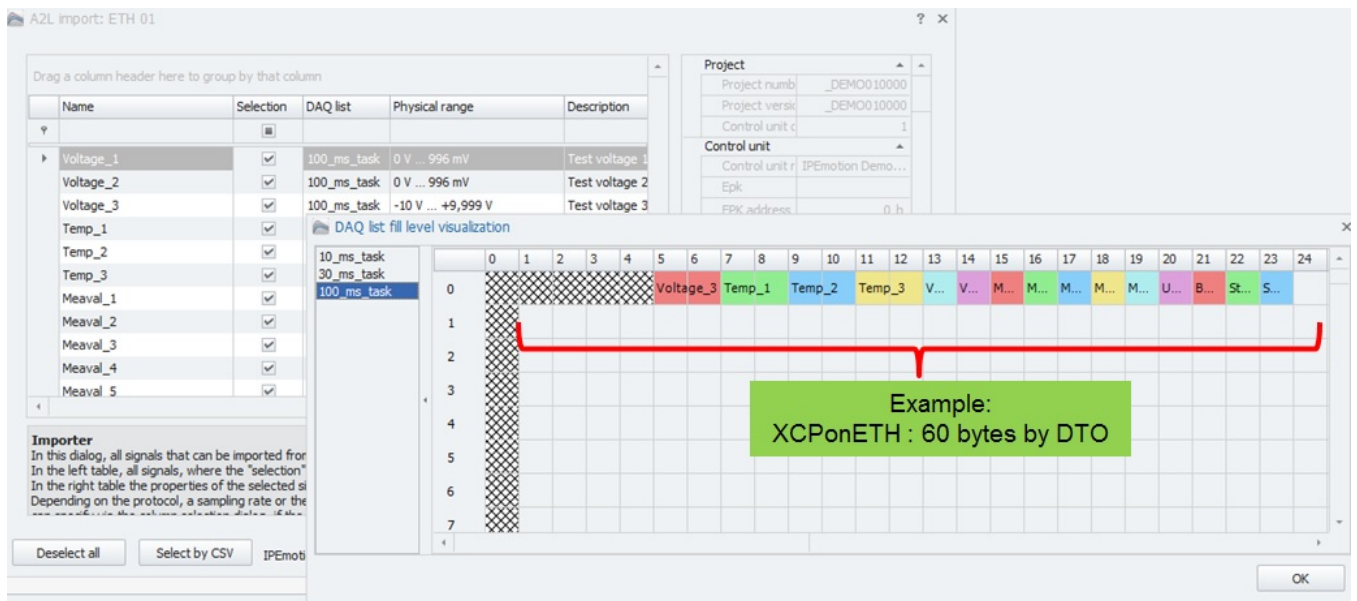
[IM_3]

5.5.1 DAQ list filling process

The import dialog fills the DAQ list in an optimized way in order to use as much of the available capacity as possible. The maximum capacity for signal measurement of a DTO is 7 byte for CCP measurements and XCPonCAN. The first byte is used for the address header.

Example 1:

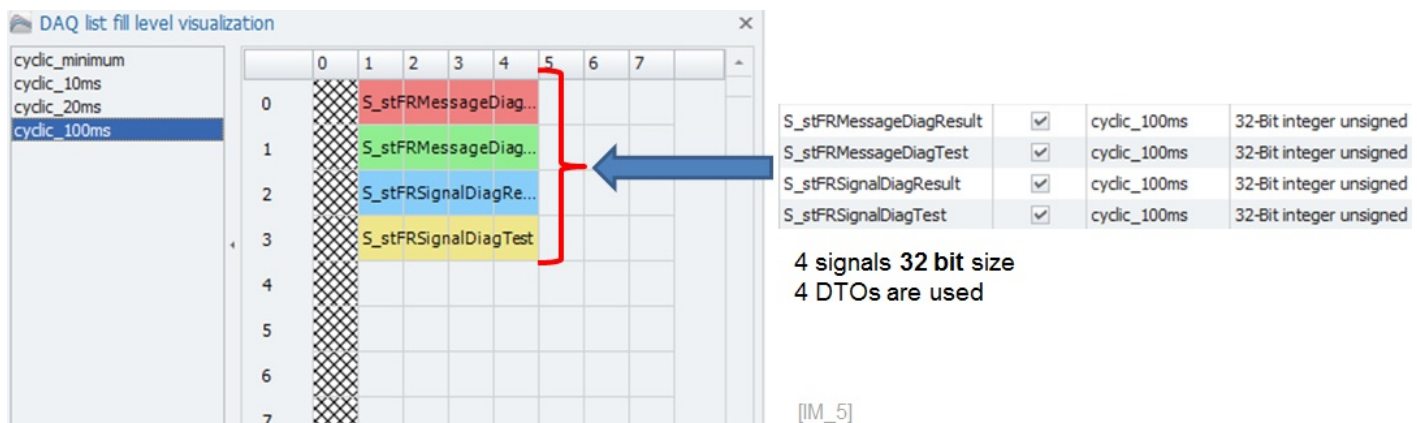
In comparison of you take an A2L file and measure XCPonETH on the Ethernet interface of IPETRONIK data loggers like M-LOG have a lot more byte capacity on the DTO. The number of available bytes is defined in the A2L file. In the screenshot below you can see the import dialog and the DTO fill level for an XCPonETH measurement with 60 byte capacity on each DTO.



[[IM_4]]

Example2:

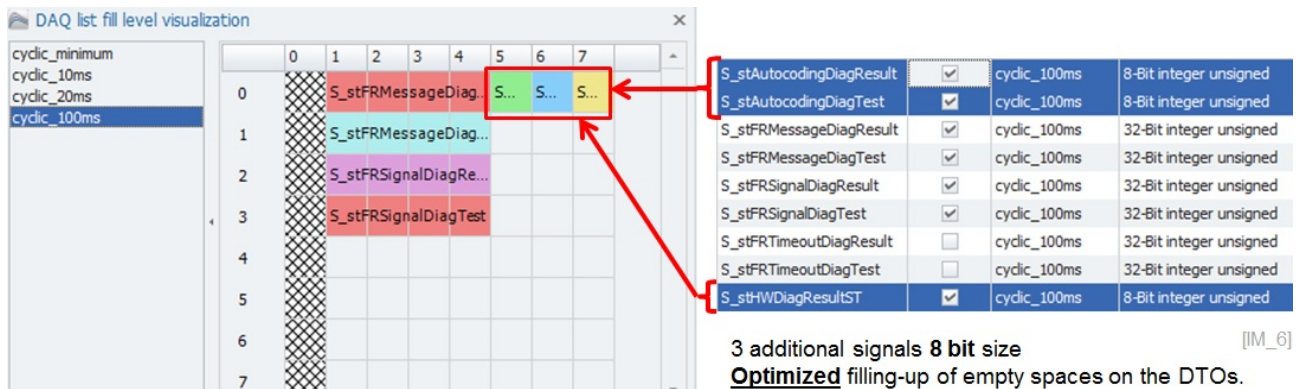
In the screenshot below you can see that 4 DTOs are used for four 32-bit signals. It is not possible to fill 2 signals of 32-bit signals in one 7 byte DTO.



[[IM_5]]

Example 3:

In the following example, 3 additional 8-bit signals are activated. Now you can see that the IPEmotion software automatically fills up the 3 empty bytes on the first DTO (Data Transfer Object).



Information

The allocation of signals to DTOs is optimized by IPEmotion, internally. It cannot be influenced by users. Some A2L files support the reading of multiple signals from one common DTO address. In this case several signals are allocated to the same DTO address and the mouse over tip text is indicating all channels grouped together in this DTO.

5.5.2 DAQ list overflow – rejected signal export

If you activate more channels than the DAQ list is able to support, you can create a list of rejected channels which can be exported to CSV. With the mouse over function you can read the channel names which are included in the DAQ list.

The screenshot shows the 'DAQ list fill level visualization' window. On the left, a list of channels is shown, with 'cyclic_100ms' selected. The main area is a grid with columns 0-7 and rows 0-11. A red hatched pattern indicates filled channels. A tooltip for 'S_stHWDiagResultST' is shown over a cell in row 2, column 7. Below the grid is a 'Rejected signals:' list containing various signal names and their data types (e.g., 'W_Output_idxDiagErrMT (2 byte)'). A green box labeled 'Rejected Signals list' has an arrow pointing to a CSV export window. The CSV window shows a list of rejected signals in column A, starting with 'W_Output_idxDiagErrMT' and ending with 'W_phiSMSetPointClutchOp'.

Mouseover indicates channel name.

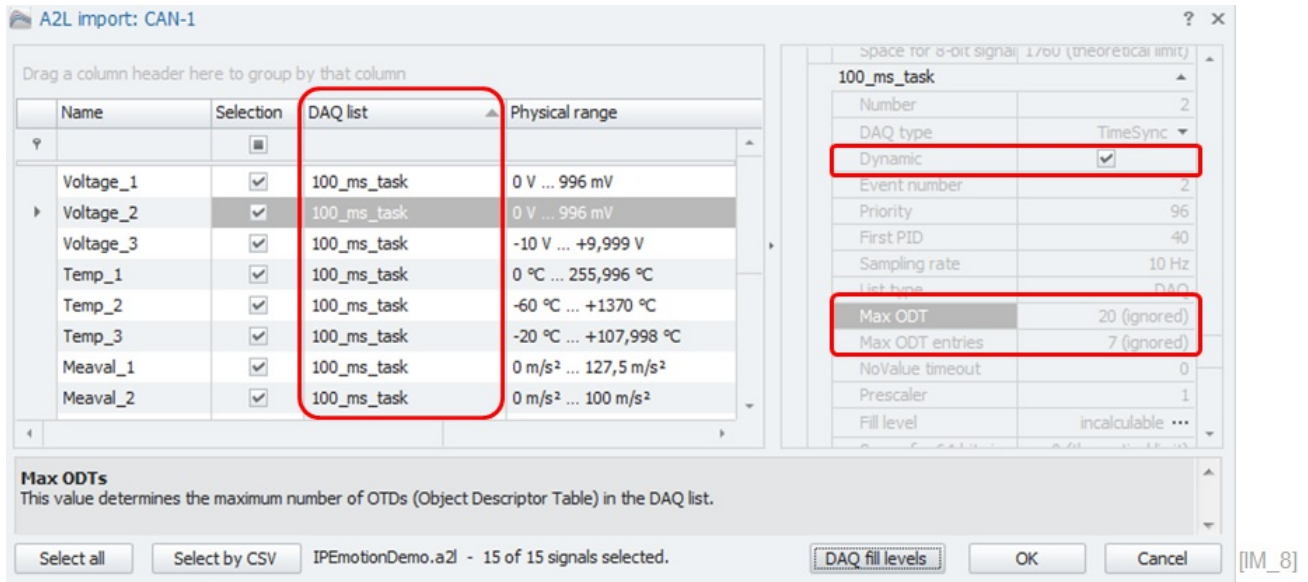
Rejected Signals list

CSV export – rejected signals

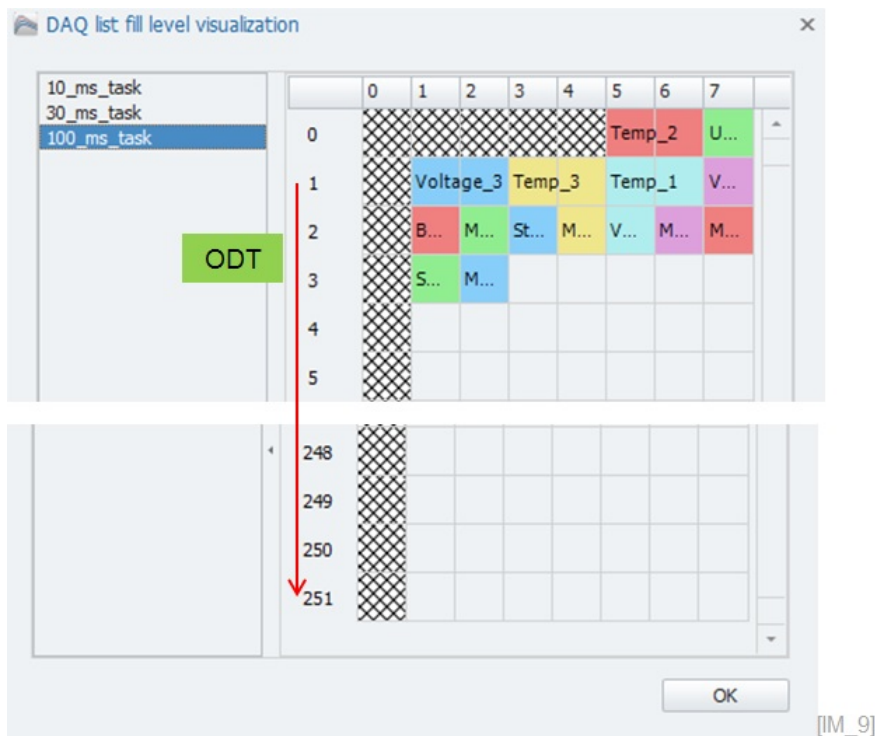
5.6 Edit A2L dynamic DAQ list ODT values during the import

When you import an A2L file for dynamic DAQ list measurements e.g. for XCPonCAN on your ECU the import dialog considers by default the maximum ODT count = 252. (ODT = Object Descriptor Table). The DAQ list fill level calculation is based on the default assumption that the 252 ODT can be serviced by the ECU.

In practice the user tend to overload the dynamic DAQ list why adding to many ODT. This problem can be solved in the way that the user can define in the import dialog the appropriate values for the MAX ODT and MAX ODT entries to calculate the fill level correctly.



Dynamic DAQ list setting: Where the MAX ODT values are not considered (ignored) for fill level calculation.



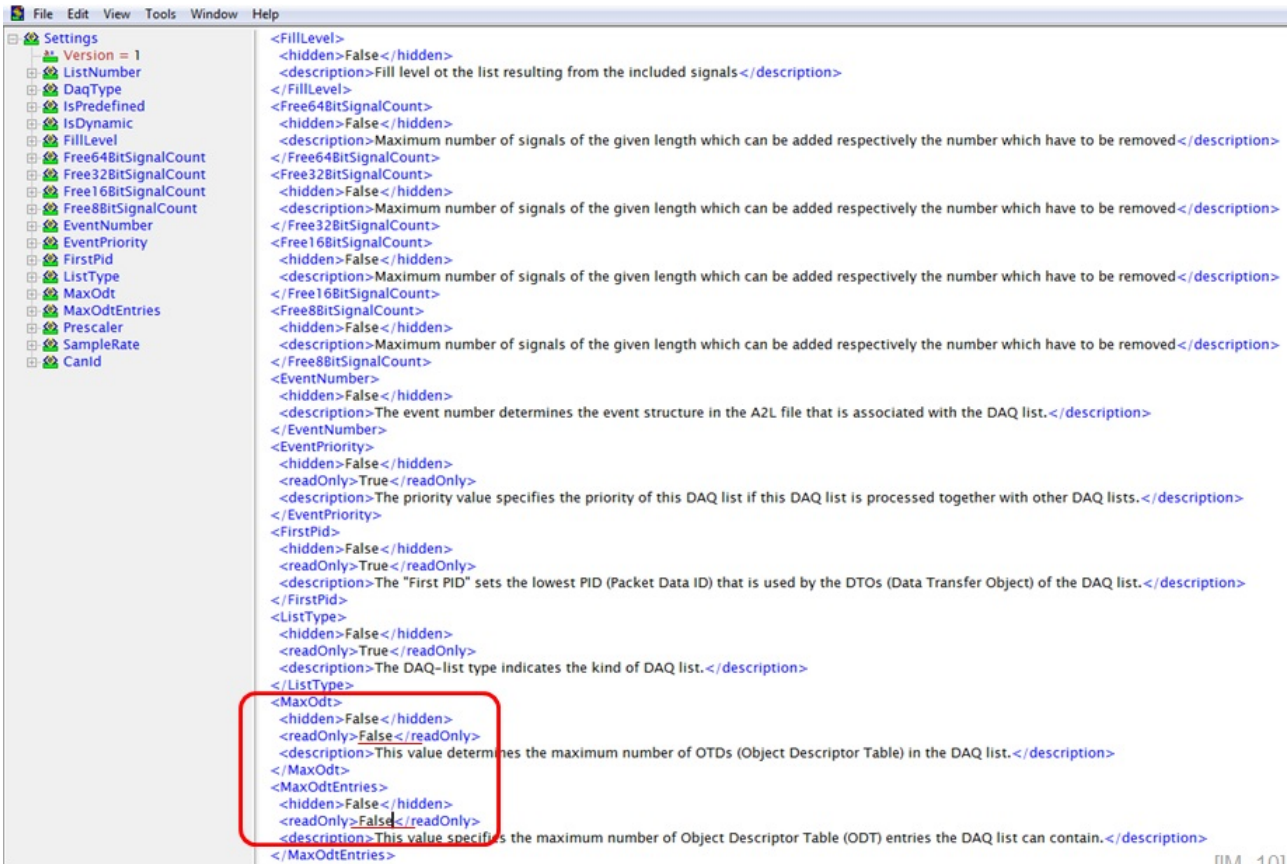
252 ODT's are considered as available capacity for fill level calculation.

If you now like to calculate the DAQ list fill level correctly you must correct the values for **MAX ODT** and **MAX ODT Entries**. In order to do this you can take a template file from the following directory:

- ▶ C:\ProgramData\IPETRONIK\IPEmotion 2022 R1\Template\DaqListXcpCan.xml

This file has to be transferred into the following directory:

- ▶ C:\ProgramData\IPETRONIK\IPEmotion 2016 R1 RC\UserSettings \DaqListXcpCan.xml

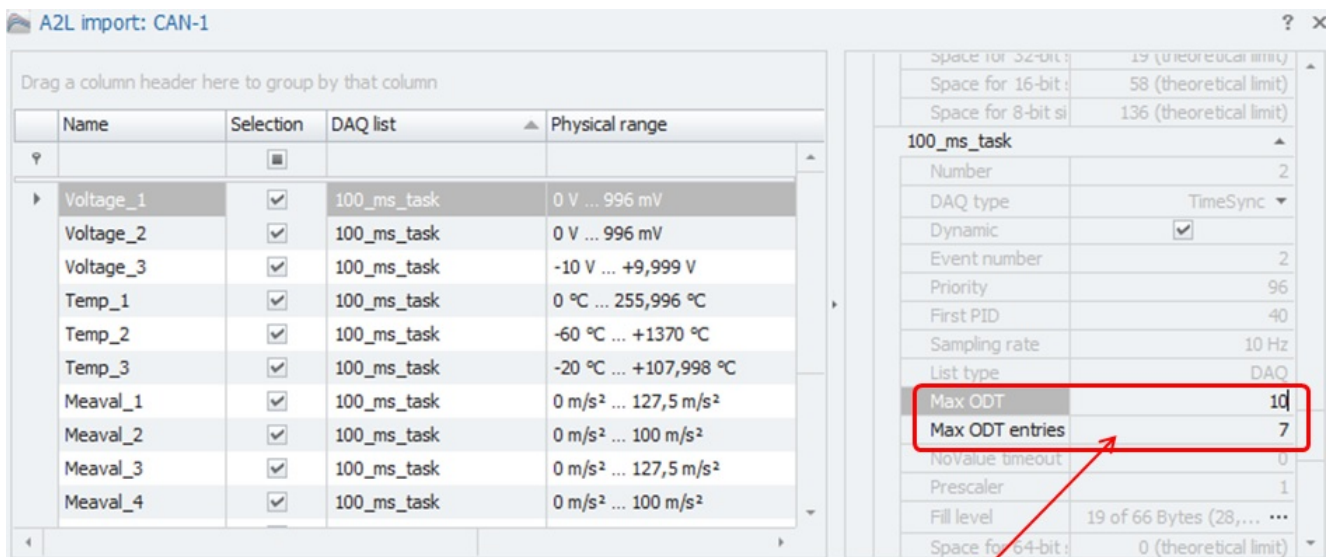


[IM_10]

Change the value from „True“ into „False“ to make the field editable.

You need to close IPEmotion and edit the XML file and set the values for Max ODT and max ODT Entries for read only into "FALSE".

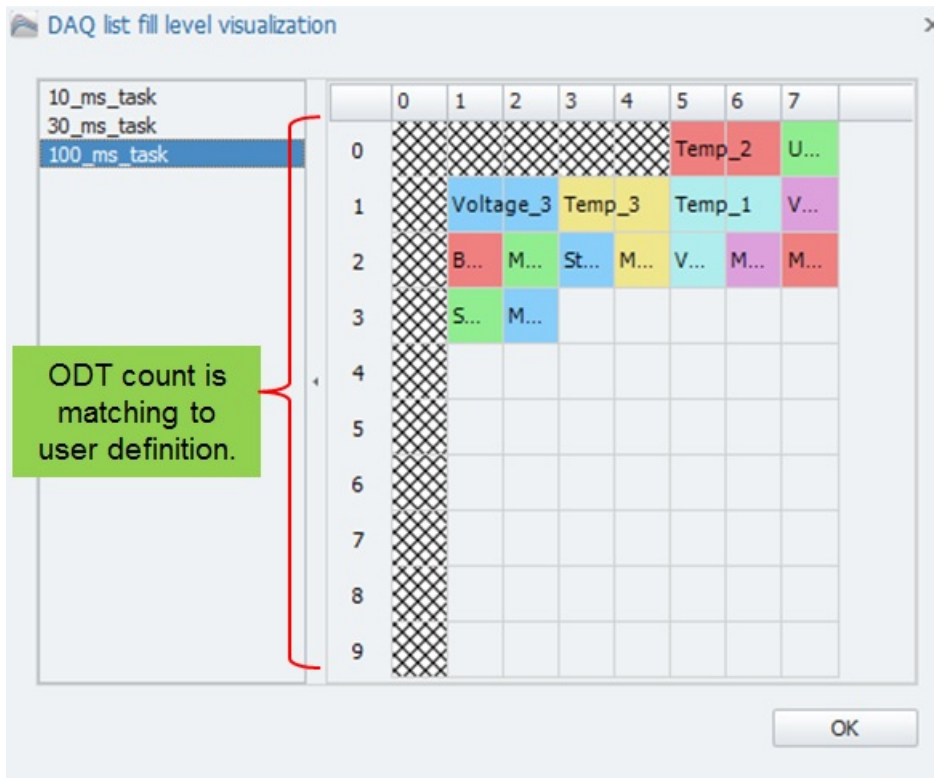
With this setting in the XML file you are able to modify the ODT setting in the import dialog.



Editable fields for user defined ODT count.

[IM_11]

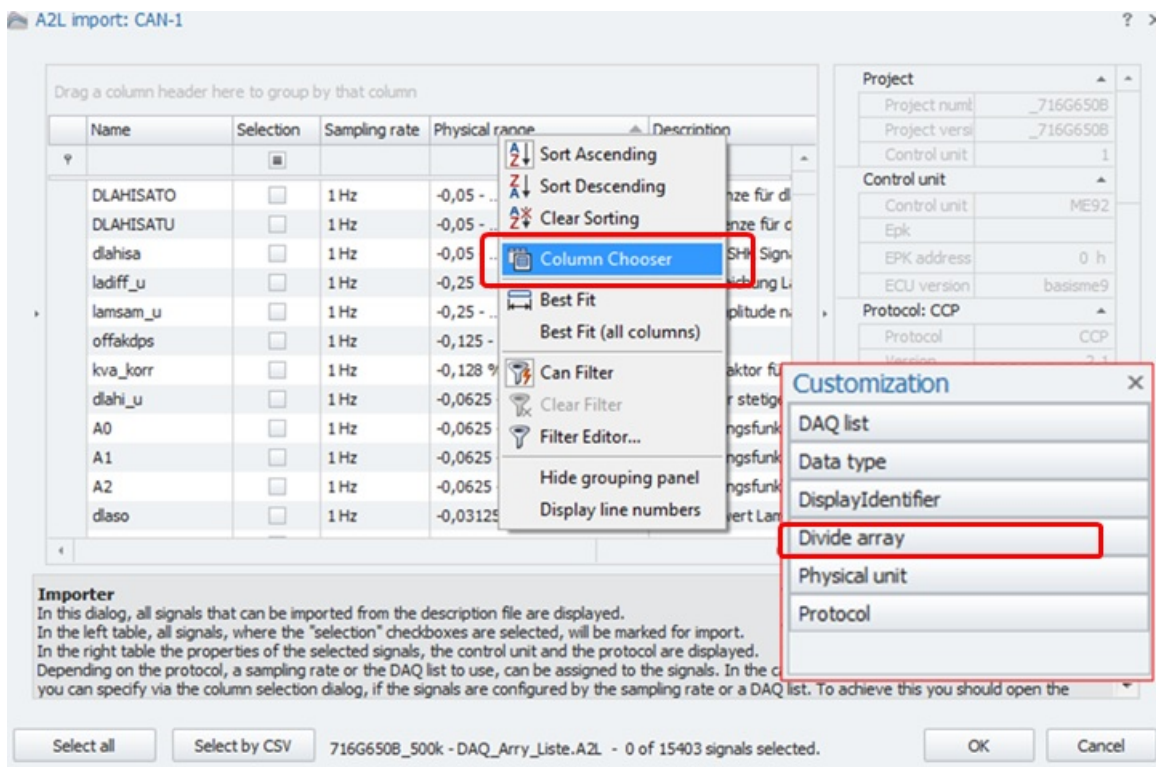
In the example above the MAX ODT count was set to 10 and the fill level calculation is now considering this value.



[IM_12]

5.6.1 A2L Import for array signals

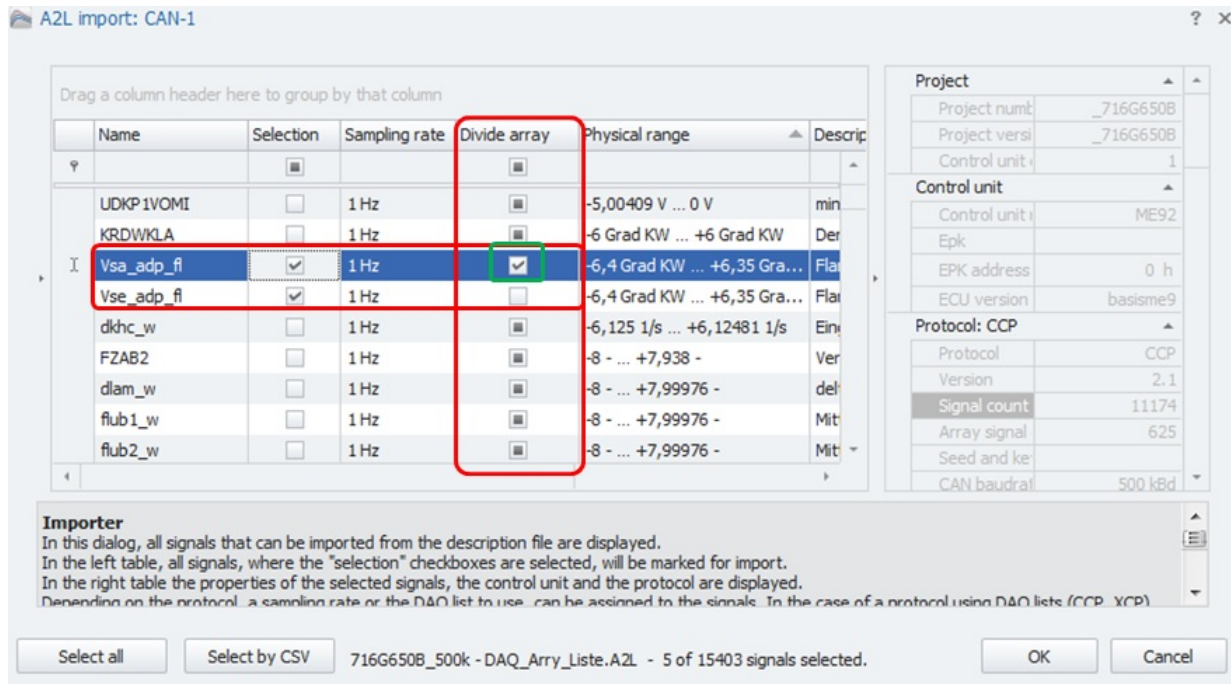
The A2L import is supporting array signals. Array signals are basically grouping several measurements together to one channel. In order to import the array signals you need to add in the import dialog from the column chooser the field "Divide array" to the channel grid.



Select "Divide Array" column.

[IM_13]

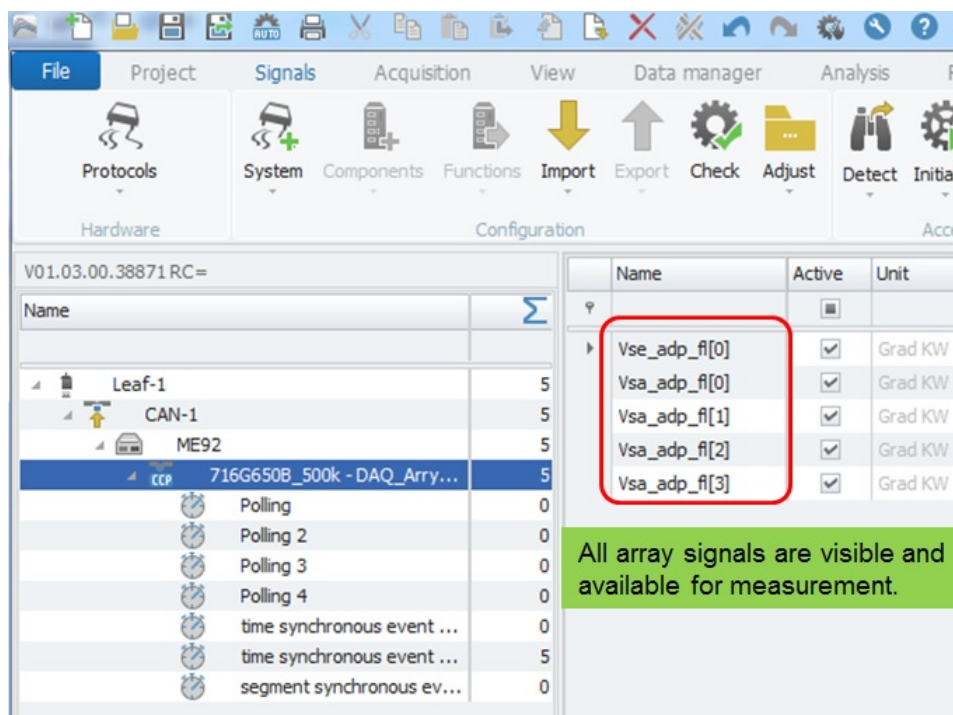
When the array column is included in the channel grid, you can activate the check boxes which include array signals.



Check box activated will provide the complete array list for measurement.

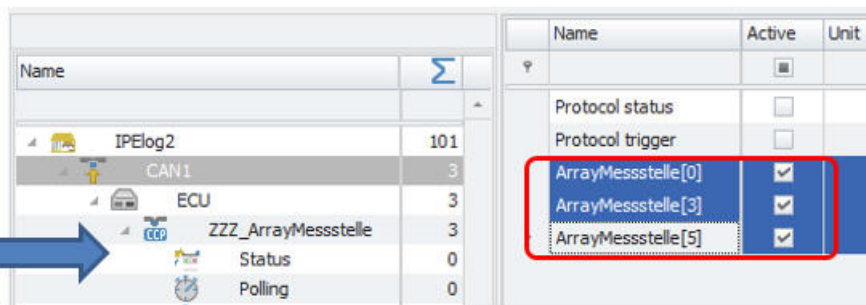
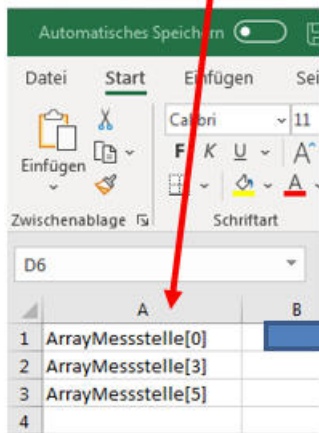
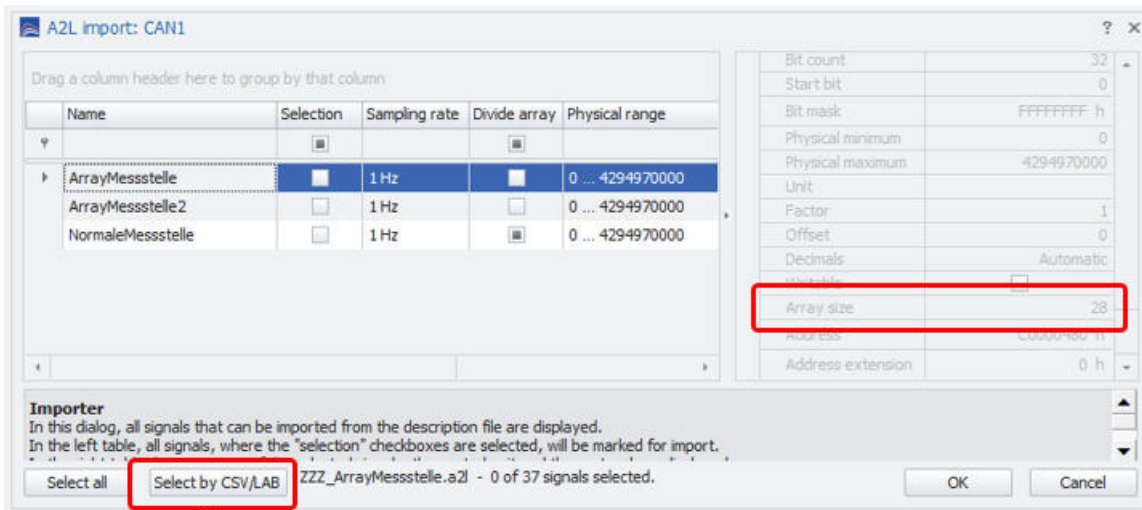
[IM_14]

After the import the array signals are all listed and available for measurement.



[IM_15]

When the A2L file includes array signals you can import specific signals with a CSV file selection. The standard process is that the complete array is imported. However, if you like to import only specific signals from the array the CSV file selection is only possibility. In the CSV file you define array signal name and the ID in brackets.



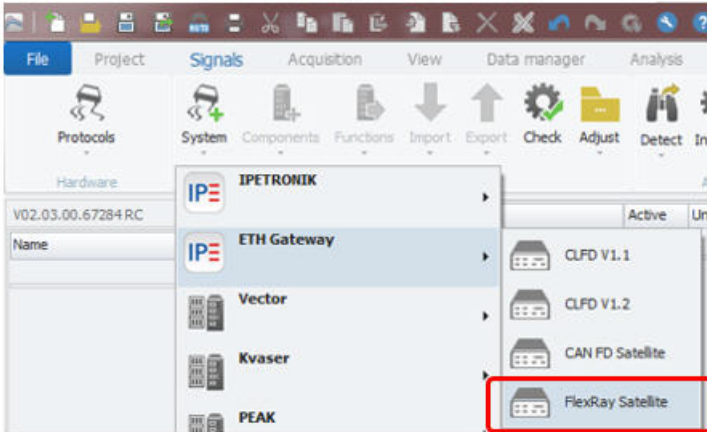
Array signal import via CSV selection

[IM_15_1]

5.6.2 A2L import with additional FlexRay parameters import

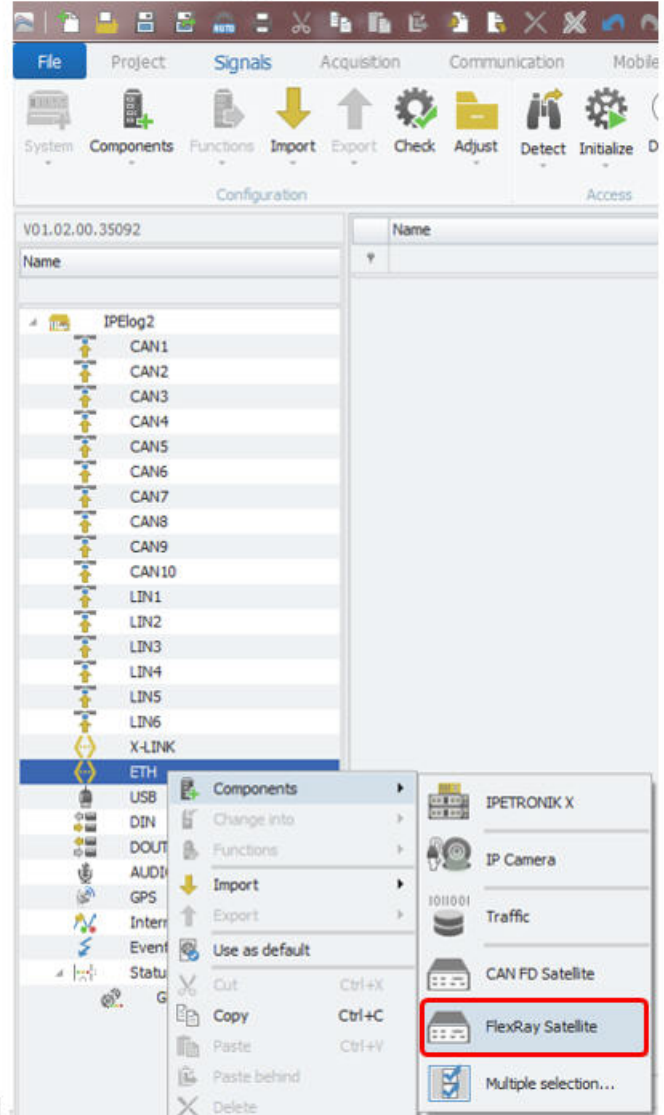
For FlexRay measurements you need to create a FlexRay interface. For IPEmotion RT and PC based FlexRay measurements a FlexRay Satellite Interface is available.

IPEmotion Protocols PlugIn

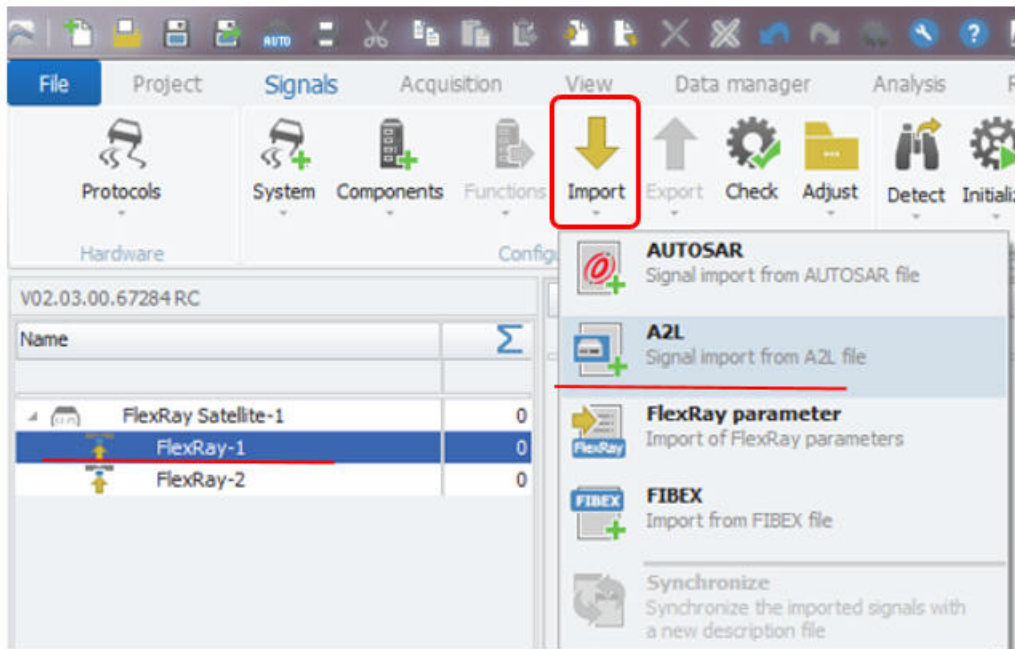


Add FlexRay Satellite interface

IPEmotion RT logger

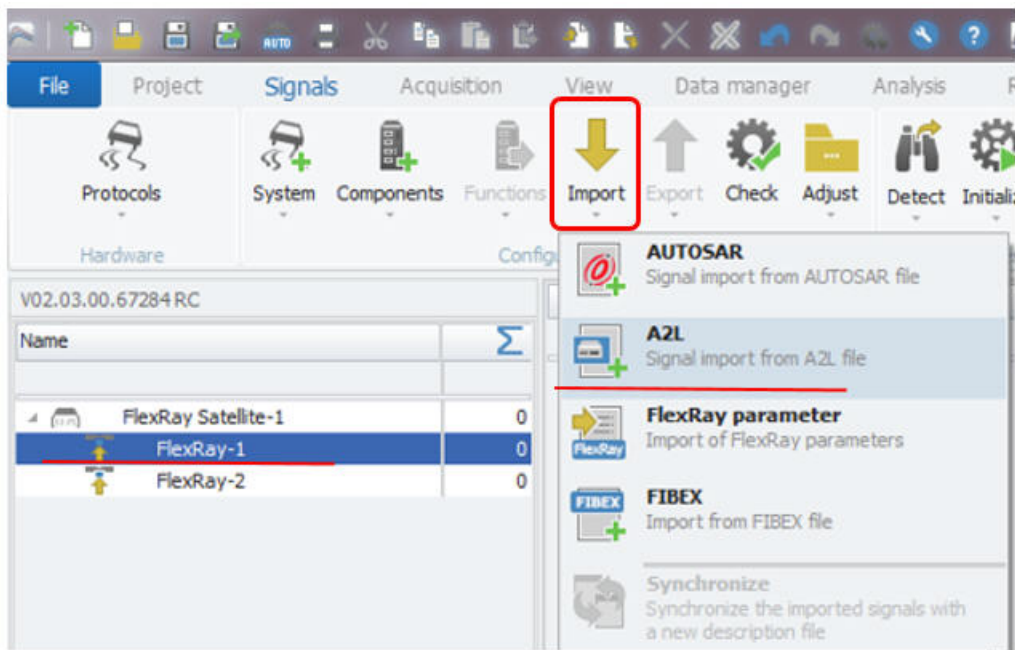


[IM_15_2]



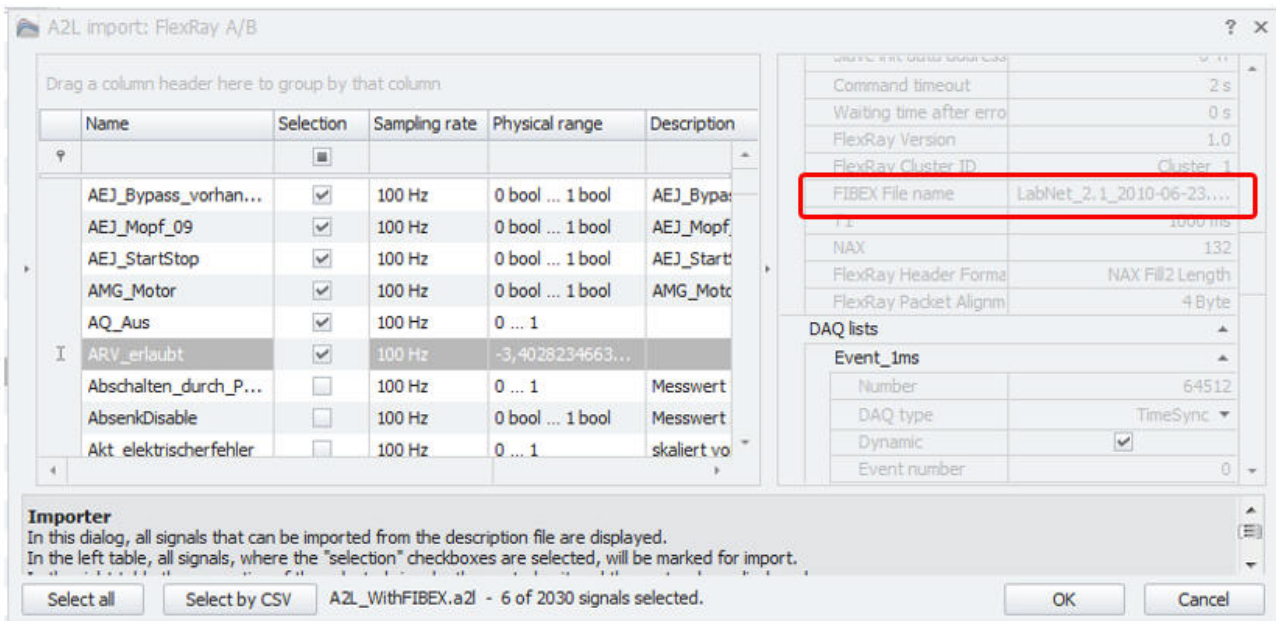
FlexRay Satellite for FlexRay measurement on PC and IPEmotion RT loggers. [IM_16]

When you have an A2L description file for a FlexRay protocol measurement and the A2L is not including any FlexRay parameter the import process will open automatically a file open dialog to load the parameter file.



FlexRay Satellite for FlexRay measurement on PC and IPEmotion RT loggers. [IM_16]

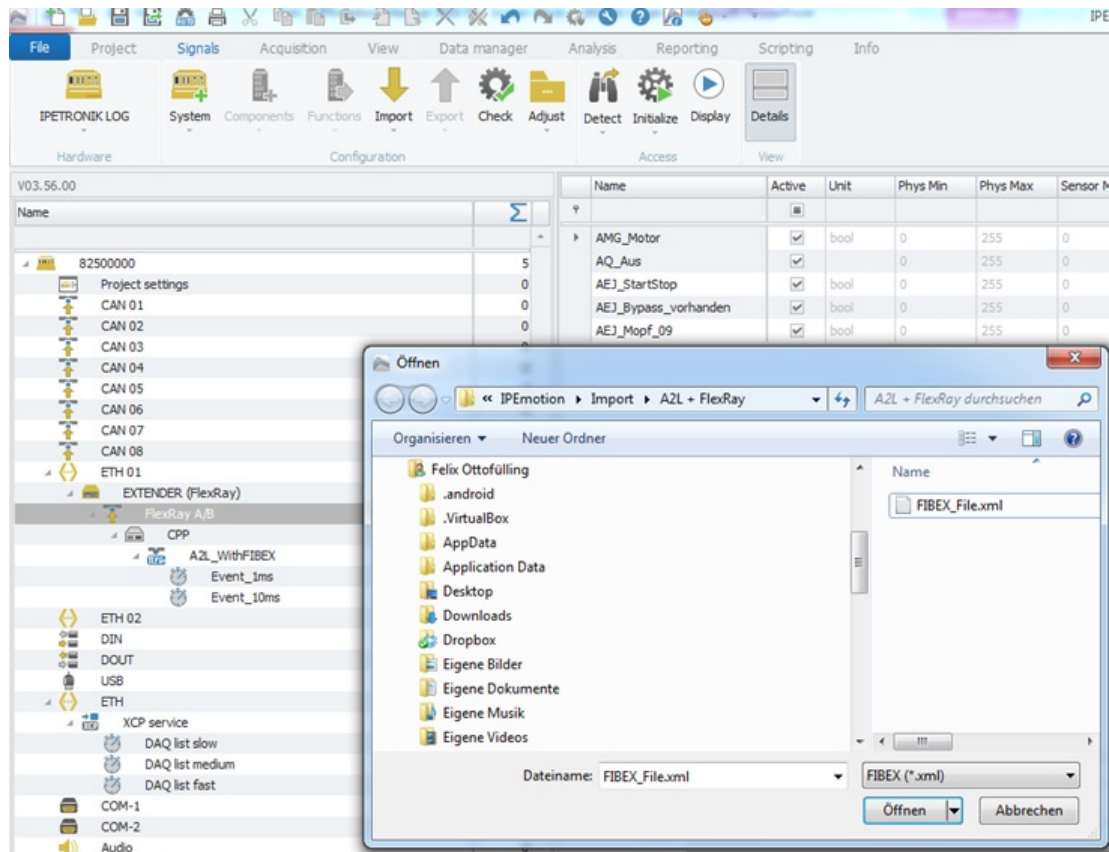
Import dialog to select the FlexRay signals.



Import dialog to select the XCPonFlexRay signals. Link to the FlexRay parameter XML file for Ecu communication.

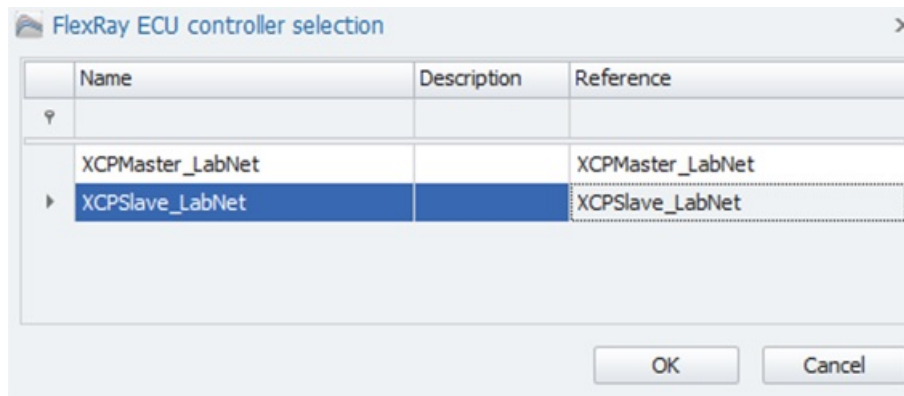
[IM_17]

When the FlexRay parameter file as defined in the A2L is not available a second import dialog will be opened automatically.



Automatically new import dialog to select FlexRay parameter file. [IM_18]

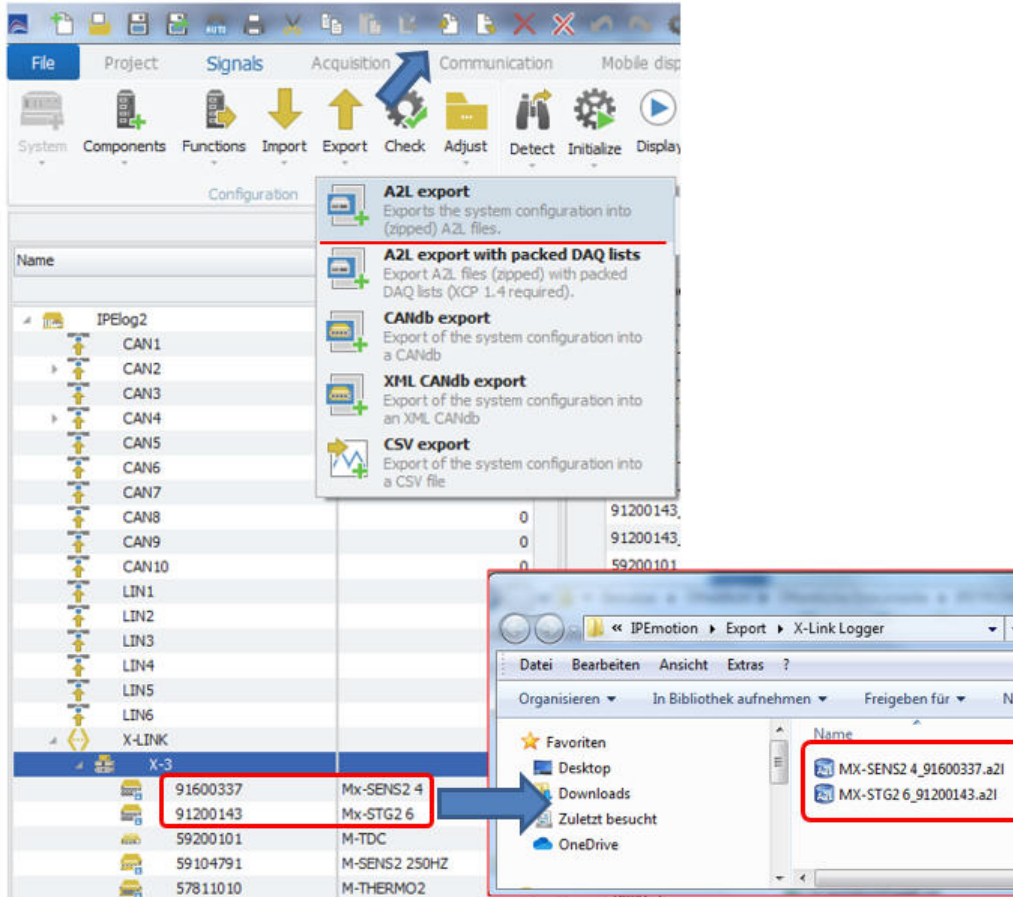
New dialog, to select FlexRay parameters.



Select FlexRay parameters for ECU communication. [IM_19]

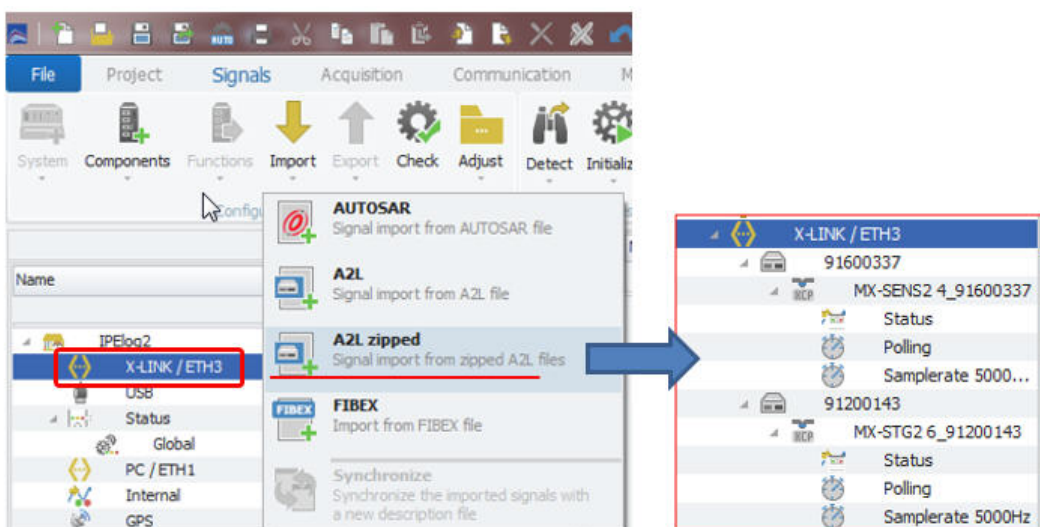
5.6.3 A2L import from zip files (X-Modules)

When X-Modules are configured on the ETH interface in the of the logger or on the PC with the IPETRONIK PlugIn X the export function is creating one ZIP file which includes all A2L files of each X-Module. The dedicated import of A2L ZIP format is then automatically importing all A2L files to create the configuration. On the activated channels are included in the A2L file.



Export of multiple X-Modules in one ZIP file

[IM_19_1]

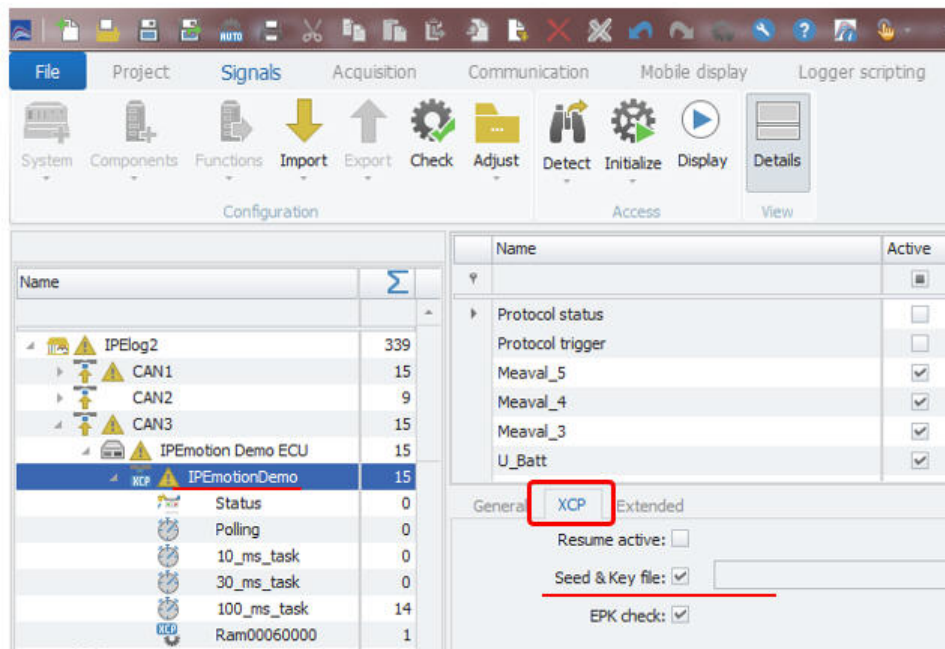


Import one zip file which contains multiple A2L

[IM_19_2]

5.6.4 XCP and CCP with Seed & Key with SKB licensing

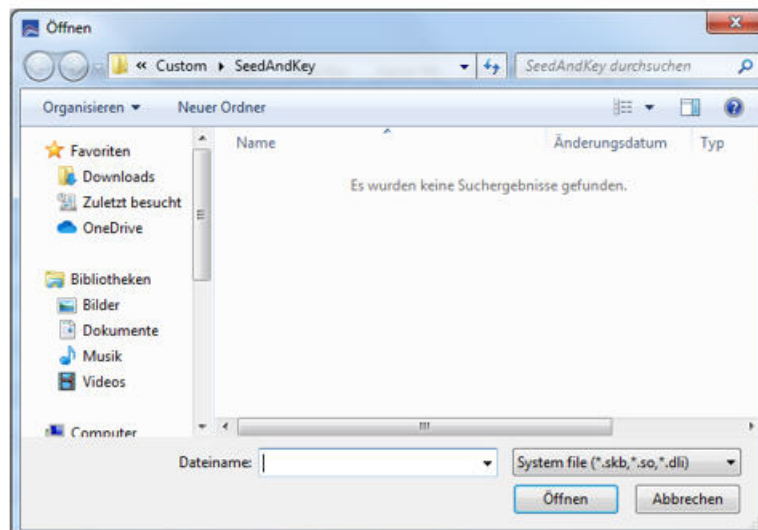
For the ECU measurements over the XCP and CCP protocol some ECU require a Seed & Key .SKB file in order to establish an ECU communication. The SKB file can be linked on the XCP and CCP CAN node in order to get access to the ECU.



XCP & CCP protocol measurement with Seed & Key file

[IM_19_3]

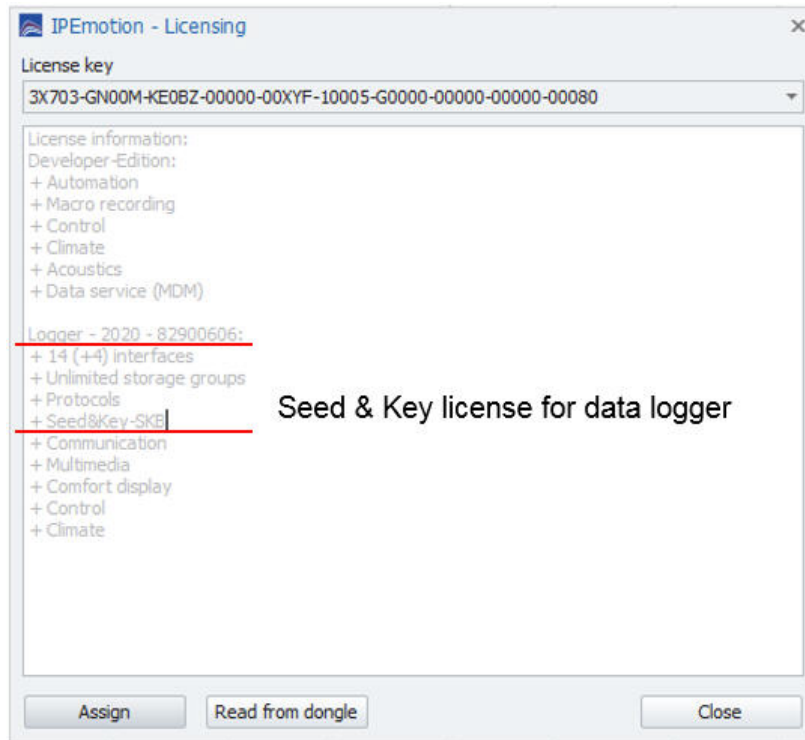
The SKB license file is based on a Windows .DLL. This file cannot be directly used on the LINUX based IPEmotion RT logger. The conversion support to a LINUX supported SKB file in .SO or .DLI file is provided by IPETRONIK.



Upload S&K file from the file pick dialog

[IM_19_4]

The Seed & Key functionality requires a license on the IPEmotion RT Data logger. The availability of the license will be checked when the Seek & Key functionality is initialized on the logger. The availability of the license can be checked in the license dialog when the logger is detected.

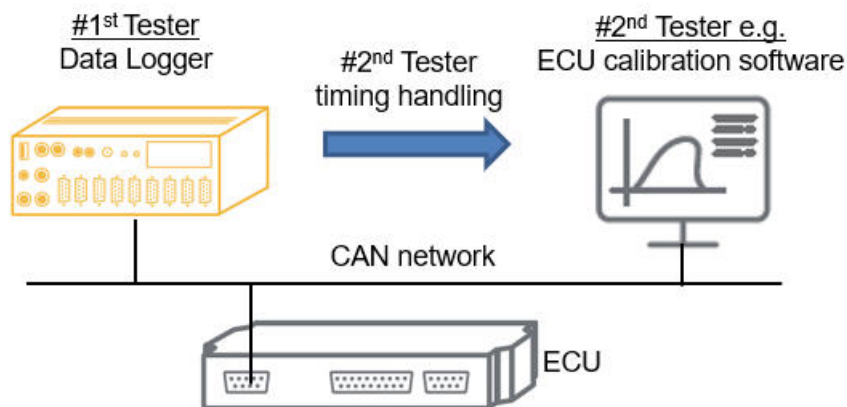


Seed & Key licensing on the logger enabled

[IM_19_5]

5.6.5 XCP, CCP second tester

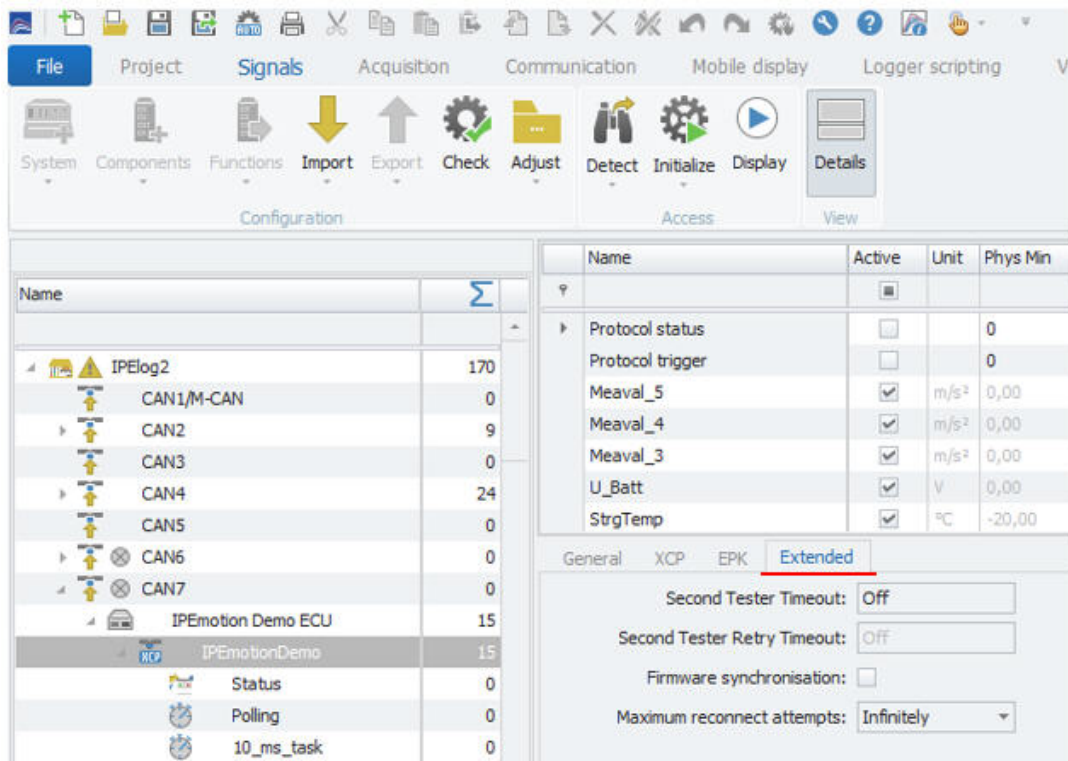
In the extended tab sheet, the second tester configuration for the CCP and XCP protocols on the different layers (CAN, CAN FD, FlexRay, UDP, TCP) can be configured. A second tester is defined as a second software program accessing an ECU while it is communicating with the data logger.



Second tester application

[IM_19_6]

To avoid communication conflicts between data logger and the second tester Ecu calibration software, the logger provides specific functions to handle the parallel communication to the ECU. The second tester software has a higher priority and established a connection and causes a disconnect of the data logger. However, with timeout settings and reconnect settings the logger can re-establish a connection after the second tester is disconnected or inactive. If the connection between logger and ECU is interrupted due to a detected second tester, the next connection attempt by the logger within this session is not done before this timeout expired.



ECU extended second tester settings for CCP and XCP

[IM_19_6_1]

- ▶ **Second Tester Timeout**

If the connection between logger and ECU is interrupted due to a detected second tester, the next connection attempt by the logger within this session is not done before this timeout expired. The time out value is defined in seconds. The timeout is deactivated when the value is set to 0 s. This is displayed as OFF. If the SSecond Tester Retry Timeout is deactivated, the time defined for SSecond Tester Timeout will cyclically check, if a second tester can be detected.

Example: "Second Tester Timeout": 5s "Second Tester Retry Timeout": Off If a second tester is detected within the first 5 seconds, the system will immediately wait another 5 seconds while searching for a second tester again. This continues as long as a second tester is detected.
- ▶ **Sec. Tester Retry Timeout**

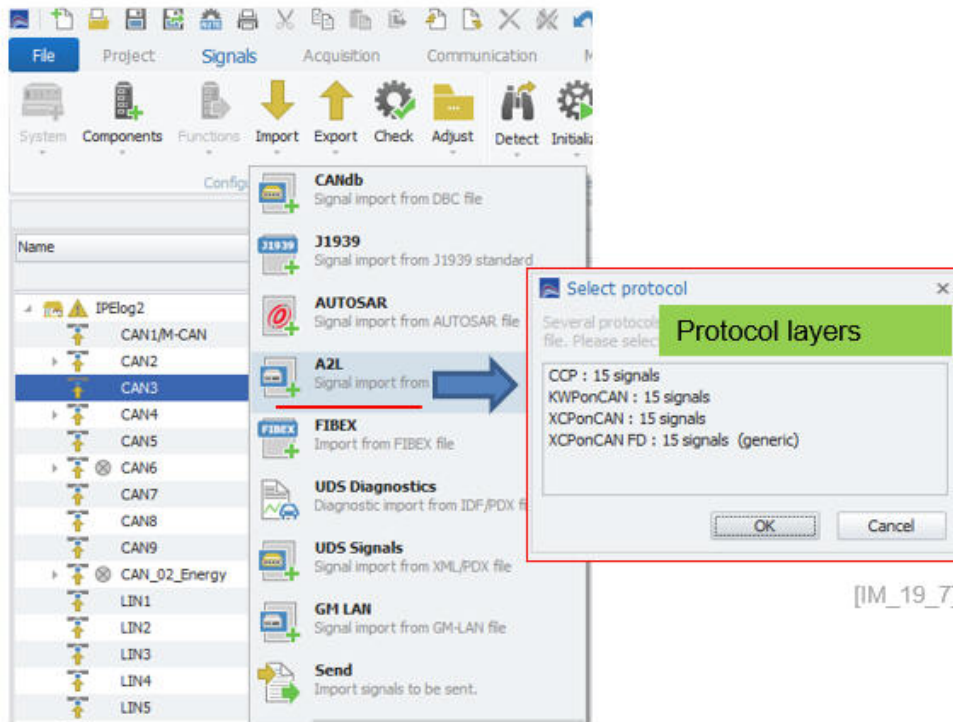
The input field "Second Tester Retry Timeout" is inactive when SSecond Tester Timeout from above is set to OFF". The retry timeout is deactivated when the value is set to 0 s. This is displayed as OFF. The retry timeout can be defined in the range of 0 ... 3600 s. The step width is 30 s. Valid input data is e.g. 0 s, 30 s, 60 s, 90 s etc. If the SSecond Tester Retry Timeout is activated, the time defined for "Second Tester Retry Timeout" is cyclically checked, if a second tester can be detected. An activation of the "Protocol Trigger" resets the waiting time of "Second Tester Retry Timeout"
- ▶ **Firmware synchronization**

Here you can enable that an ECU firmware synchronization. With this check box enabled, a new ECU firmware can be synchronized from the data logger to the ECU via the IEV firmware container. The IEV file can be created from an export function in the SIGNALS workspace.
- ▶ **Max. reconnect attempts**

Here you define how often an ECU reconnect will be attempted.

5.6.6 XCP transport layers

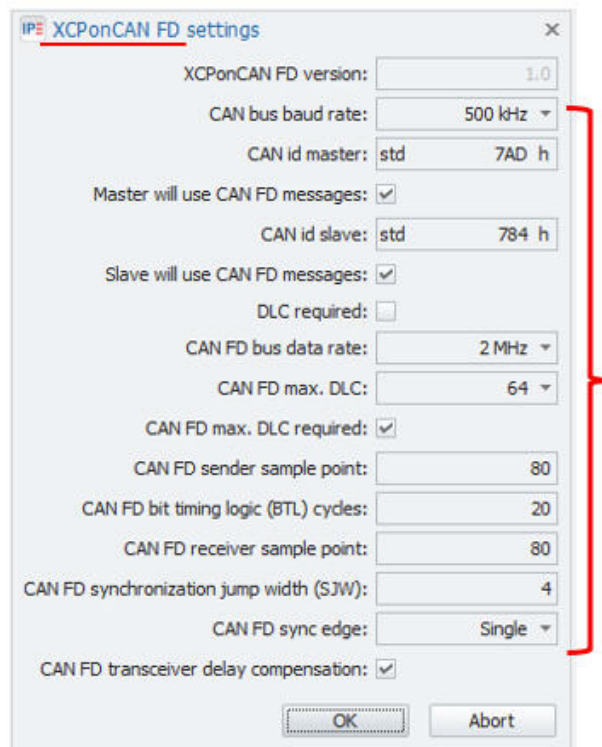
The A2L import for the XCP protocol supports different transport layers. The latest layer implemented is based on CAN FD. CAN FD can only be selected when it is defined in the A2L file.



[IM_19_7]

XCP protocol – Select transport layers

When CAN FD is selected in the import dialog above the following dialog appears where additional protocol settings from the A2L file can be modified, when needed.



Settings of CAN FD transport layer [IM_19_8]

5.6.7 XCPonETH for IPv6

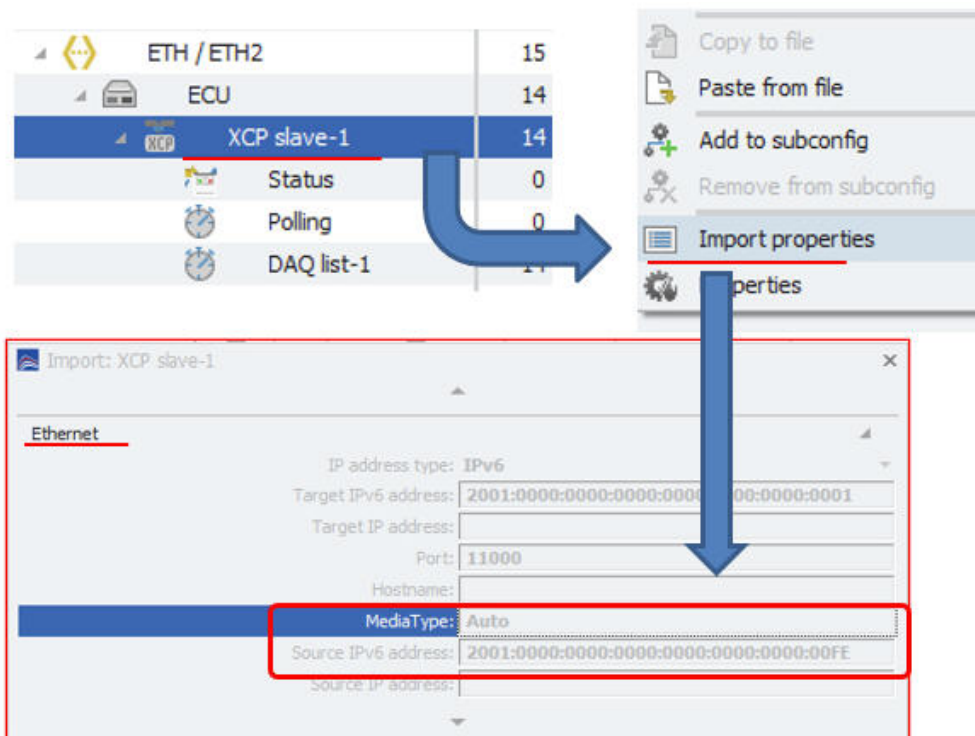
The IPv6 IP-address standard was implemented on the XCPonEthernet measurement operation. The IPv6 IP-address range is much larger as the IPv4 address range and becomes more important to handle the growing number of network devices.



Structure of IPv6 IP-address

[IM_19_9]

When importing A2L or ARXML files including IPv6 IP-address they are supported, and the IP-address is visible in the import property dialog.



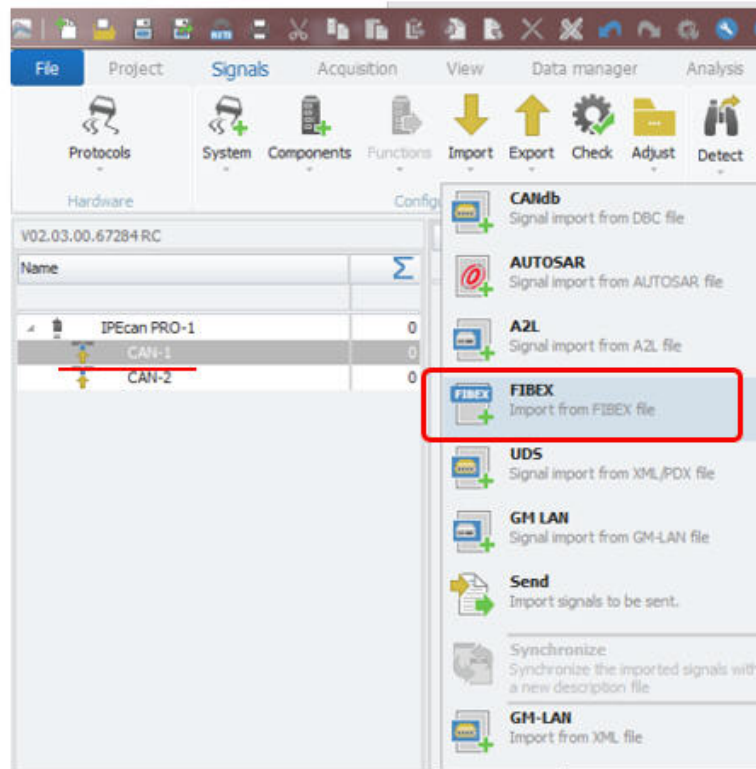
IPv6 IP-address for XCPonETH protocol

[IM_19_10]

5.7 FIBEX import

5.7.1 Import CAN signals from FIBEX files

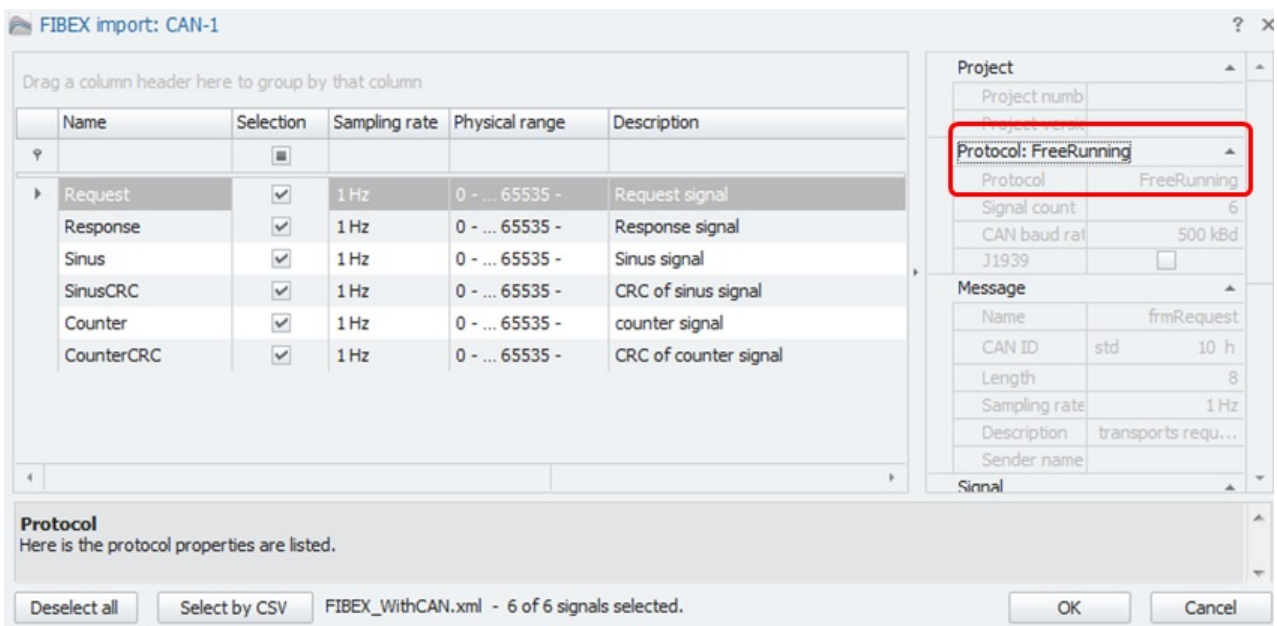
You can import on all PlugIn supporting CAN interfaces for CANdb measurements a FIBEX file. When the FIBEX file includes CAN messages it is an adequate replacement for the DBC file.



FIBEX import on all CAN interfaces.

[IM_20]

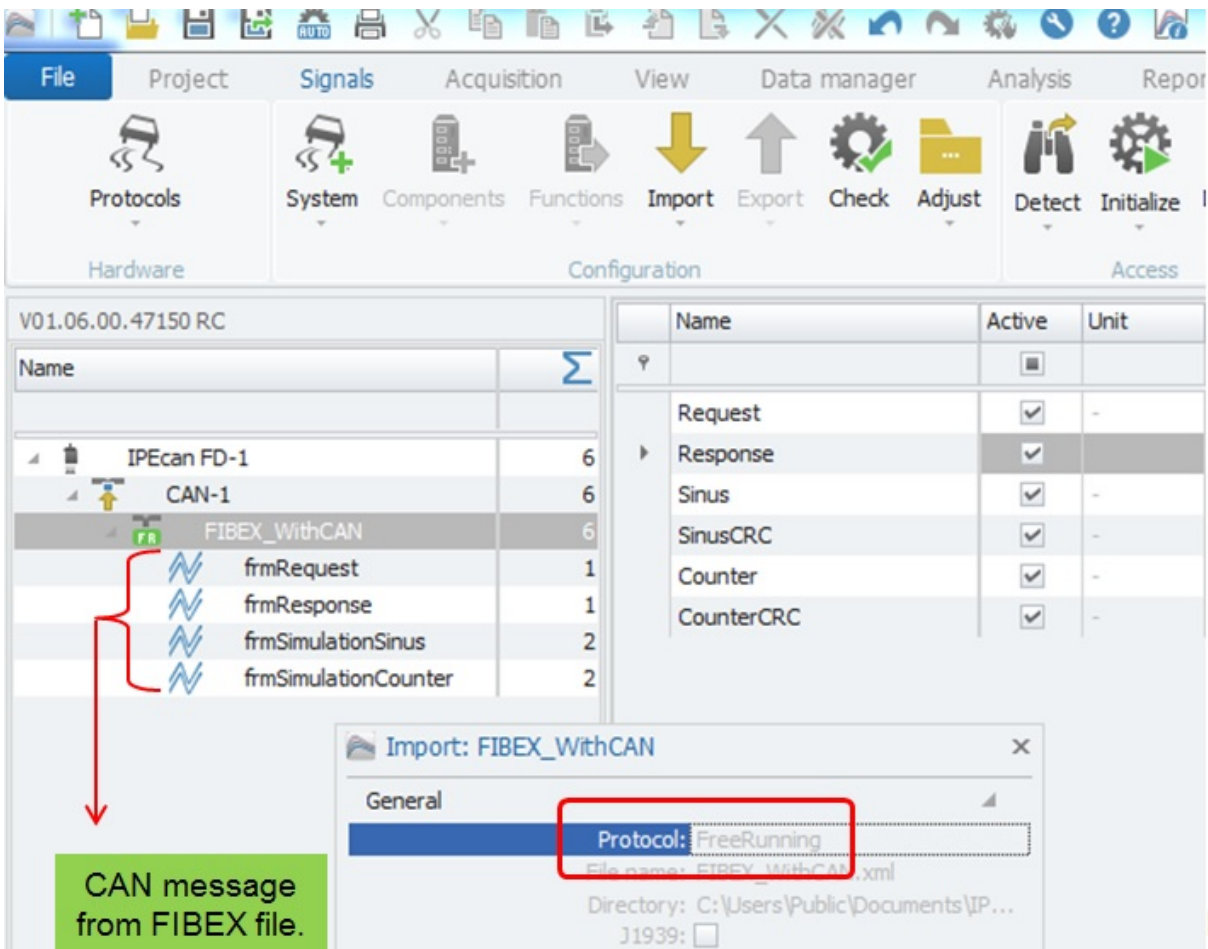
The import dialog is indicating in the protocol header "Free running".



Import dialog of a FIBEX file including free running CAN signals.

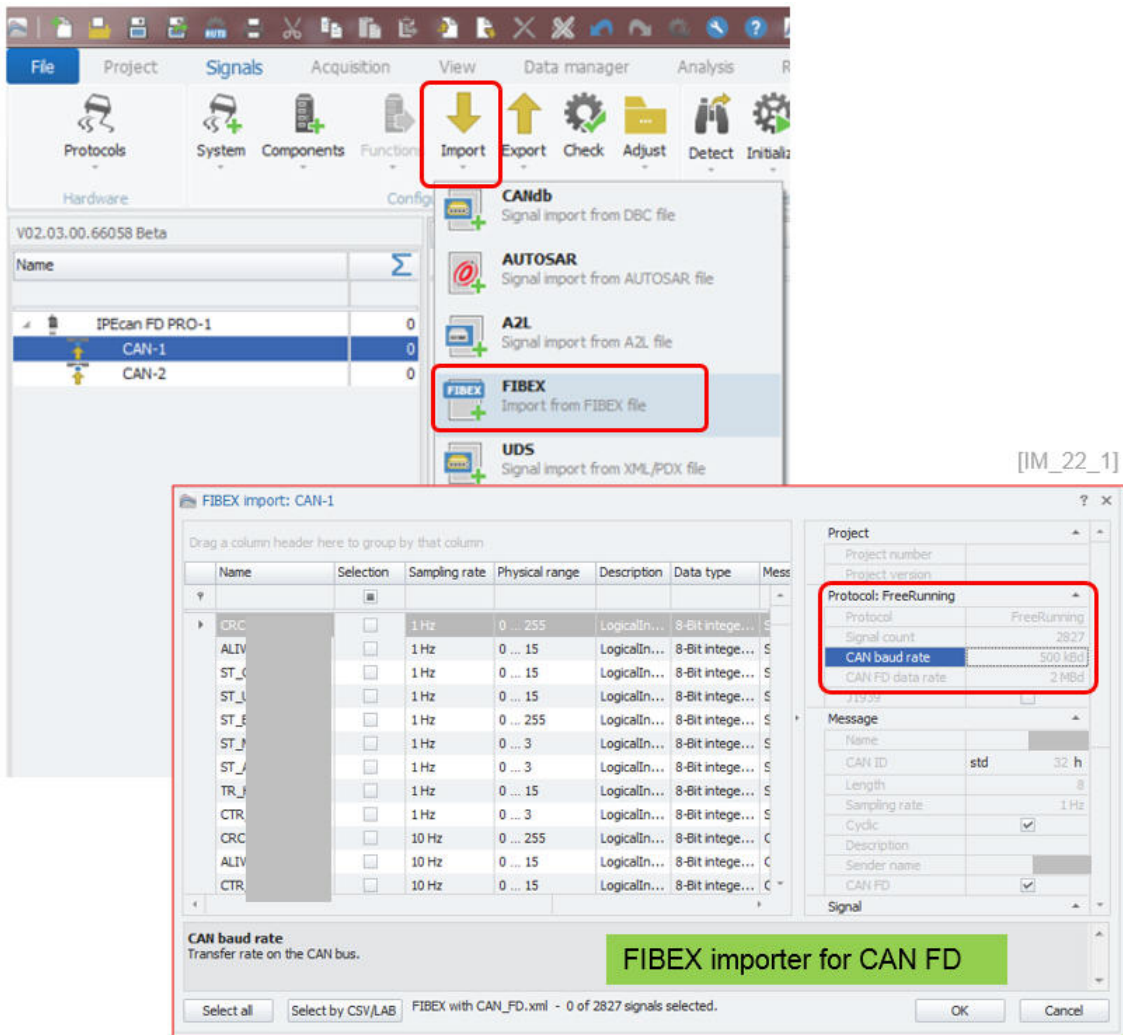
[IM_21]

After the import you see you CAN messages and the channel.



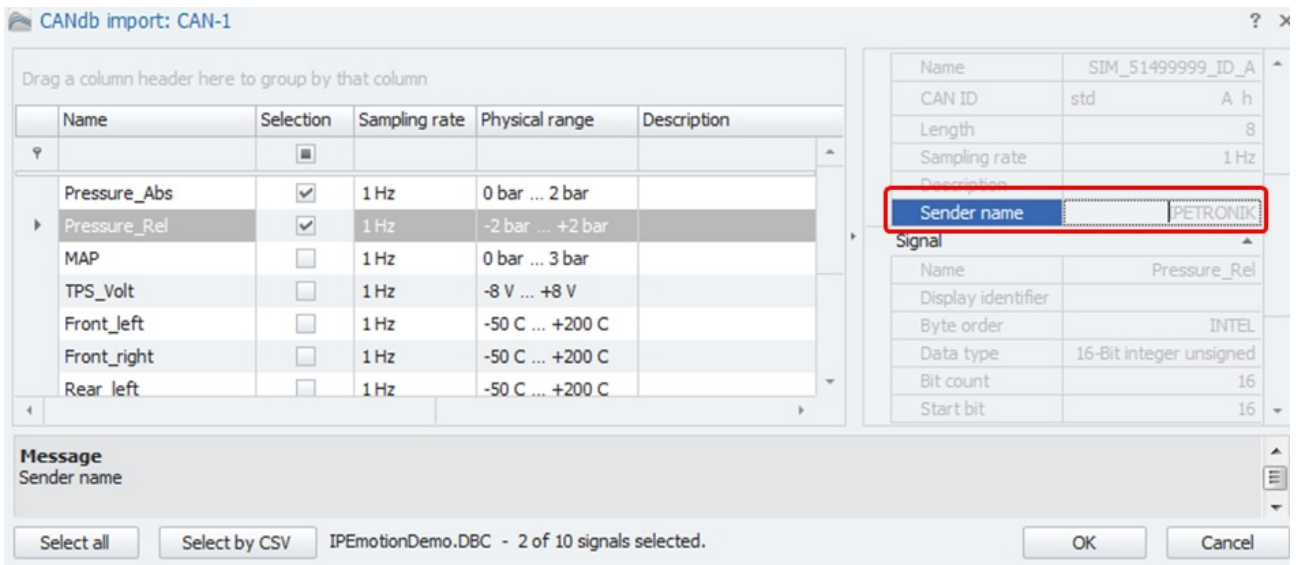
5.7.2 Import CAN FD signals from FIBEX files

The FIBEX XML file is supports CAN FD protocol imports.



5.7.3 Display of Sender name for FIBEX, DBC, AUTOSAR messages

The import properties of the CAN, FIBEX and AUTOSAR messages include now the sender name when defined in the description file.



Example: DBC import dialog including the „Sender name“.

[IM_23]

5.8 Description file import with CSV file for channel reference

The CSV reference file significantly improves the description file import and channel activation. Especially when you are working with large description files with many channels, sometimes you are uncertain if all required channels are included in the description file. It is also time-consuming to search and activate only the relevant channels for your specific measurement manually.

With the CSV reference file you can compare your description file to a CSV channel list. This comparing process covers two functions:

- ▶ All matching channel names from the CSV reference list are automatically activated. This saves a lot of time compared to activating channel by channel. The channel selection is not cases sensitive. Channels are selected even when the lower and upper cases do not match.
- ▶ All the channels which are included in the CSV file but not in the description file are listed in a separate "missing channels" list. Missing channels can be saved in a separate CSV file for later analysis purposes.

CANdb import: CAN-1

Drag a column header here to group by that column

Name	Sele...	Sampln...	Physical range	Description
C_56199999_1	<input type="checkbox"/>	1 Hz	-100 V ... +100 V	Analog acquis
C_56199999_2	<input type="checkbox"/>	1 Hz	-100 V ... +100 V	Analog acquis
C_56199999_3	<input type="checkbox"/>	1 Hz	-100 V ... +100 V	Analog acquis
C_56199999_4	<input type="checkbox"/>	1 Hz	-100 V ... +100 V	Analog acquis
C_57399999_1	<input type="checkbox"/>	1 Hz	-60 °C ... +1370 °C	Analog thermx
C_57399999_2	<input type="checkbox"/>	1 Hz	-60 °C ... +1370 °C	Analog thermx
C_57399999_3	<input type="checkbox"/>	1 Hz	-60 °C ... +1370 °C	Analog thermx
C_57399999_4	<input type="checkbox"/>	1 Hz	-60 °C ... +1370 °C	Analog thermx
C_57399999_5	<input type="checkbox"/>	1 Hz	-60 °C ... +1370 °C	Analog thermx
C_57399999_6	<input type="checkbox"/>	1 Hz	-60 °C ... +1370 °C	Analog thermx
C_57399999_7	<input type="checkbox"/>	1 Hz	-60 °C ... +1370 °C	Analog thermx
C_57399999_8	<input type="checkbox"/>	1 Hz	-60 °C ... +1370 °C	Analog thermx

The DBC file contains 12 signals.

Importer
In this dialog, all signals that can be imported from the description file are displayed. In the left table, all signals, where the "selection" checkboxes are selected, will be marked for import. In the right table the properties of the selected signals, the control unit and the protocol. Depending on the protocol, a sampling rate or the DAQ list to use, can be assigned to the protocol using DAQ lists (CCP, XCP), you can specify via the column selection dialog, if

Select all **Select by CSV**

CANdb import: CAN-1

Drag a column header here to group by that column

Name	Sele...	Sampln...	Physical range	Description
C_57399999_1	<input checked="" type="checkbox"/>	1 Hz	-60 °C ... +1370 °C	Analog thermx
C_57399999_2	<input checked="" type="checkbox"/>	1 Hz	-60 °C ... +1370 °C	Analog thermx
C_57399999_3	<input checked="" type="checkbox"/>	1 Hz	-60 °C ... +1370 °C	Analog thermx
C_57399999_4	<input checked="" type="checkbox"/>	1 Hz	-60 °C ... +1370 °C	Analog thermx
C_57399999_5	<input checked="" type="checkbox"/>	1 Hz	-60 °C ... +1370 °C	Analog thermx
C_57399999_6	<input checked="" type="checkbox"/>	1 Hz	-60 °C ... +1370 °C	Analog thermx
C_57399999_7	<input checked="" type="checkbox"/>	1 Hz	-60 °C ... +1370 °C	Analog thermx
C_57399999_8	<input checked="" type="checkbox"/>	1 Hz	-60 °C ... +1370 °C	Analog thermx

The CSV file contains only 8 signals which are in common to the DBC. > Automatic activation

Missing signals
CSV file: Demo.csv
C_Reference Channel_1
C_Reference Channel_2
C_Reference Channel_3

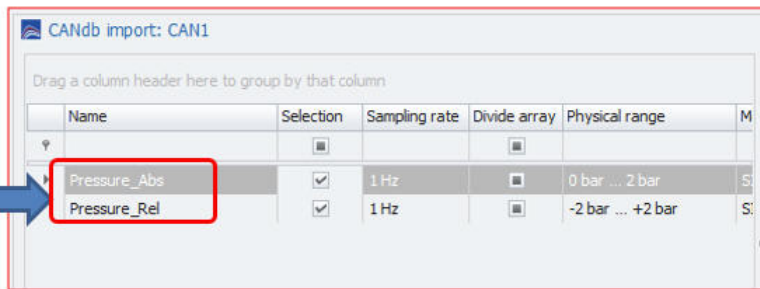
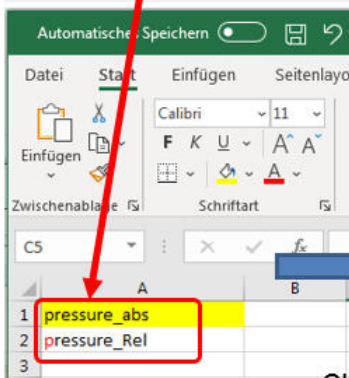
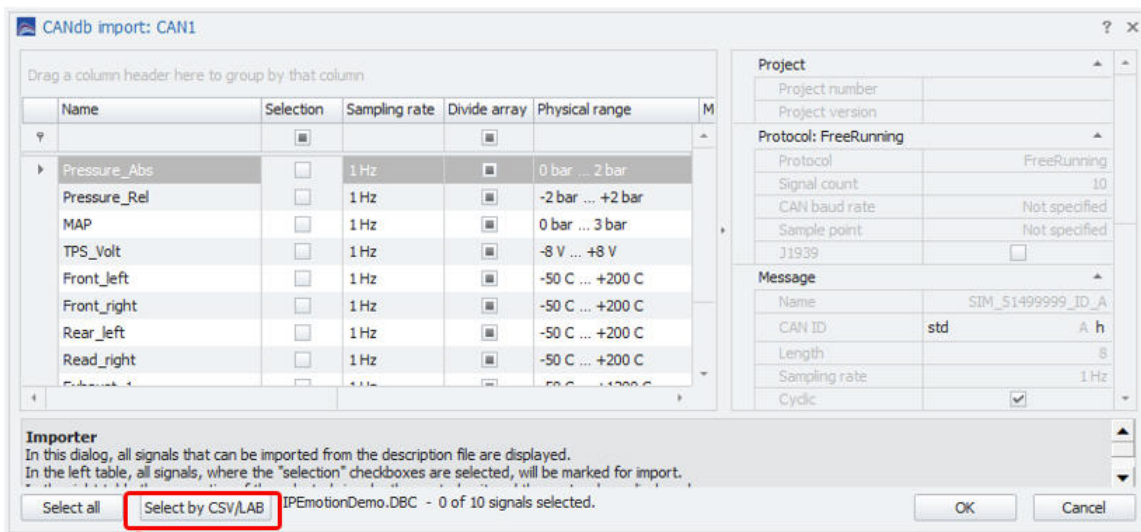
Save as CSV

Importer
In this dialog, all signals that can be imported from the description file are displayed. In the left table, all signals, where the "selection" checkboxes are selected, will be marked for import. In the right table the properties of the selected signals, the control unit and the protocol are displayed. Depending on the protocol, a sampling rate or the DAQ list to use, can be assigned to the signals. In the case of a protocol using DAQ lists (CCP, XCP), you can specify via the column selection dialog, if the signals are configured by

Select all Show all signals IPE_CANdb-CSV.dbc - 0 of 12 signals selected. OK Cancel

[IM_24]

Channels are selected even when the lower and upper cases do not match.



Channels are selected without checking the cases sensitivity [IM_24_1]

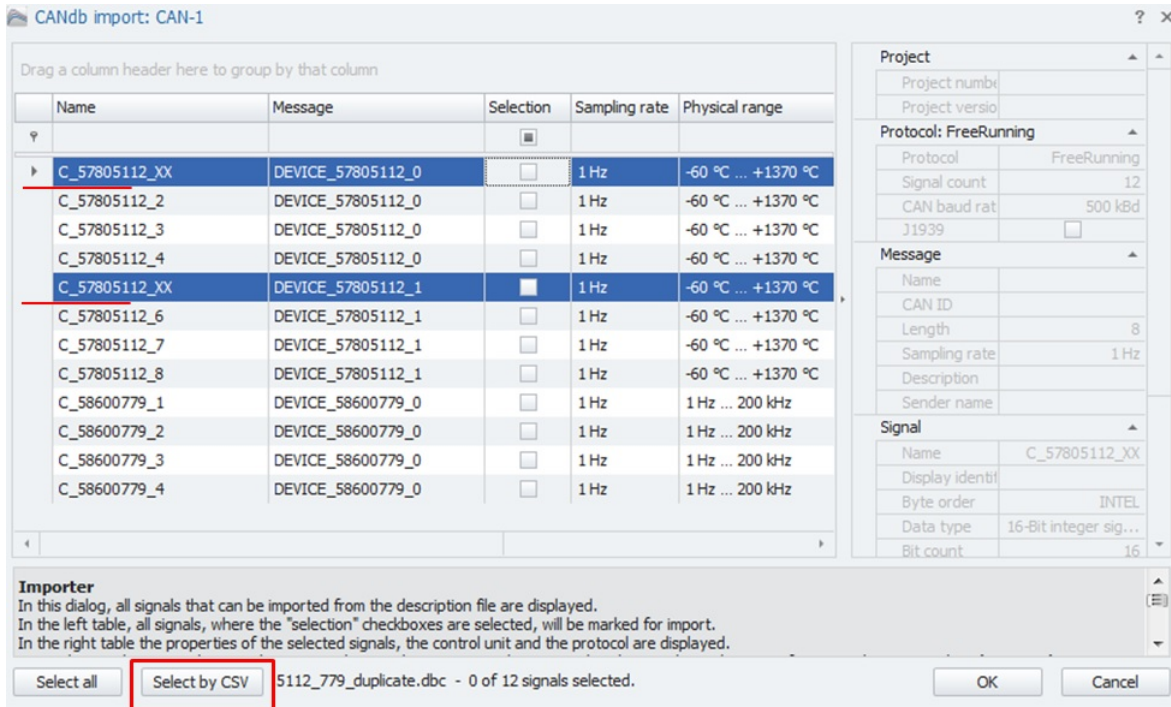


Information

A second filter criteria can be added to the CSV file to optimize the import, e.g. to specify sample rate or DAQ list settings in the import process.

5.8.1 Multi column CSV selection for description file imports (DBC, A2L)

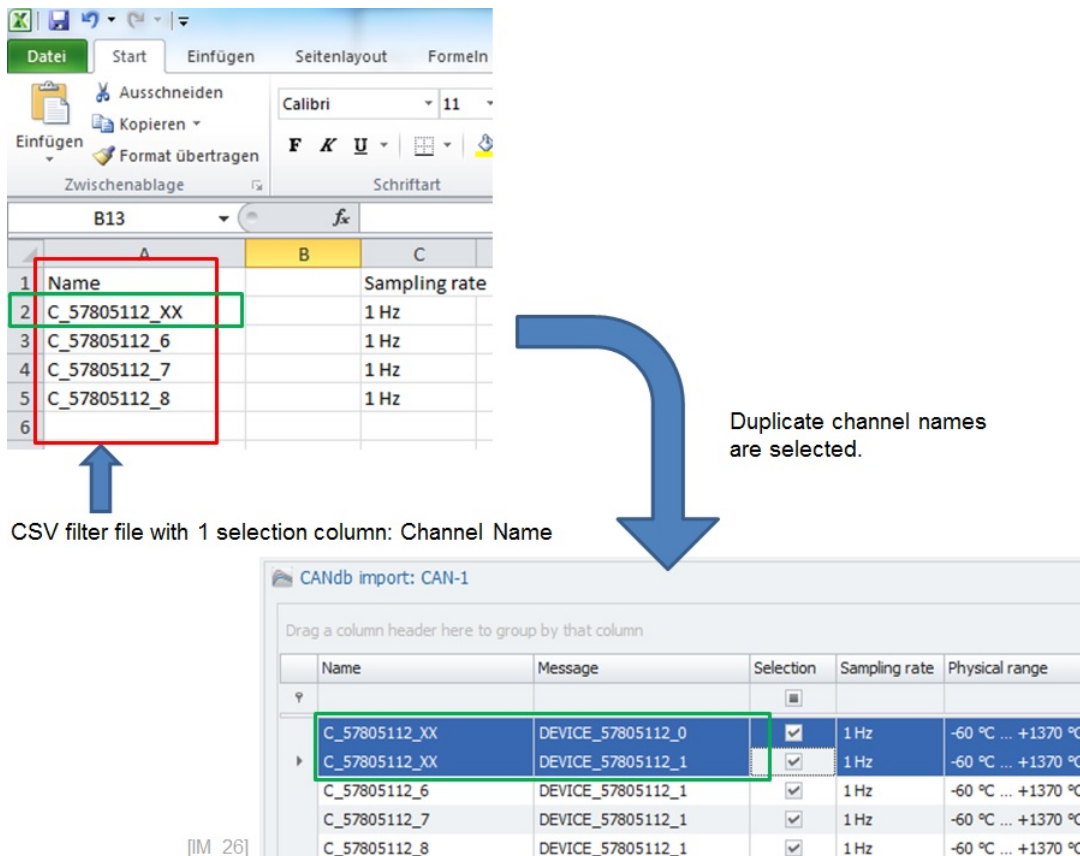
The CSV filter can support additional columns apart from the channel name to select dedicated channels in your description file import. In the example below you see how a DBC import can be improved by adding the message name as a second selection criteria.



DBC import with duplicate channel names.

[IM_25]

In this example we define only the channel names as selection criteria. In this case duplicate channels across the whole DBC file get selected during the import process.



[IM_26]

If you add the message ID to the selection criteria in column 2, you can pick the specific channels of the messages you are interested in.

CSV filter file with 2 columns: Channel names and Message ID

	A	B	C
1	Name	Reference 1	Sampling rate
2	C_57805112_XX	DEVICE_57805112_0	1 Hz
3	C_57805112_6	DEVICE_57805112_1	1 Hz
4	C_57805112_7	DEVICE_57805112_1	1 Hz
5	C_57805112_8	DEVICE_57805112_1	1 Hz

Channels of specific messages are selected.

CANdb import: CAN-1

Drag a column header here to group by that column

	Name	Message	Selection	Sampling rate	Physical range
☐			<input type="checkbox"/>		
▶	C_57805112_XX	DEVICE_57805112_0	<input checked="" type="checkbox"/>	1 Hz	-60 °C ... +1370 °C
	C_57805112_6	DEVICE_57805112_1	<input checked="" type="checkbox"/>	1 Hz	-60 °C ... +1370 °C
	C_57805112_7	DEVICE_57805112_1	<input checked="" type="checkbox"/>	1 Hz	-60 °C ... +1370 °C
	C_57805112_8	DEVICE_57805112_1	<input checked="" type="checkbox"/>	1 Hz	-60 °C ... +1370 °C

If you like you can set during the DBC import the sample rate by message ID too.

CSV filter file with 3 filter columns:
Channel name / Message ID / Sample rate

	A	B	C
1	Name	Reference 1	Sampling rate
2	C_57805112_XX	DEVICE_57805112_0	5
3	C_57805112_2	DEVICE_57805112_0	5
4	C_57805112_4	DEVICE_57805112_0	5
5	C_57805112_6	DEVICE_57805112_1	20
6	C_57805112_7	DEVICE_57805112_1	20
7	C_58600779_3	DEVICE_58600779_0	100
8	C_58600779_4	DEVICE_58600779_0	100

Select channels and set sample rate during import too.

[IM_28]

Name	Message	Selection	Sampling rate	Physical range
▾ C_57805112_XX	DEVICE_57805112_0	<input checked="" type="checkbox"/>	5 Hz	-60 °C ... +1370 °C
C_57805112_2	DEVICE_57805112_0	<input checked="" type="checkbox"/>	5 Hz	-60 °C ... +1370 °C
C_57805112_4	DEVICE_57805112_0	<input checked="" type="checkbox"/>	5 Hz	-60 °C ... +1370 °C
C_57805112_6	DEVICE_57805112_1	<input checked="" type="checkbox"/>	20 Hz	-60 °C ... +1370 °C
C_57805112_7	DEVICE_57805112_1	<input checked="" type="checkbox"/>	20 Hz	-60 °C ... +1370 °C
C_58600779_3	DEVICE_58600779_0	<input checked="" type="checkbox"/>	100 Hz	1 Hz ... 200 kHz
C_58600779_4	DEVICE_58600779_0	<input checked="" type="checkbox"/>	100 Hz	1 Hz ... 200 kHz

The sample definition can be located on column 2 also. It is not required that the sample rate definition must be located on column 3 at any time.

When you are using CSV filter for A2L import you can select channels by channel name in the first column and associated to the signal a DAQ list or sample rate during the import too.

5.8.2 Check duplicate channel names during description file import

When run an import of any description file (DBC, FIBEX, A2L, ...) the import process can check for any duplicate channel name and provide a dialog to resolve duplicate channel name conflicts.

In order to activate this feature you have to make an additional entry in the **Settings.XML** file.

- ▶ C:\ProgramData\IPETRONIK\IPEmotion 2022 R1\Settings.XML

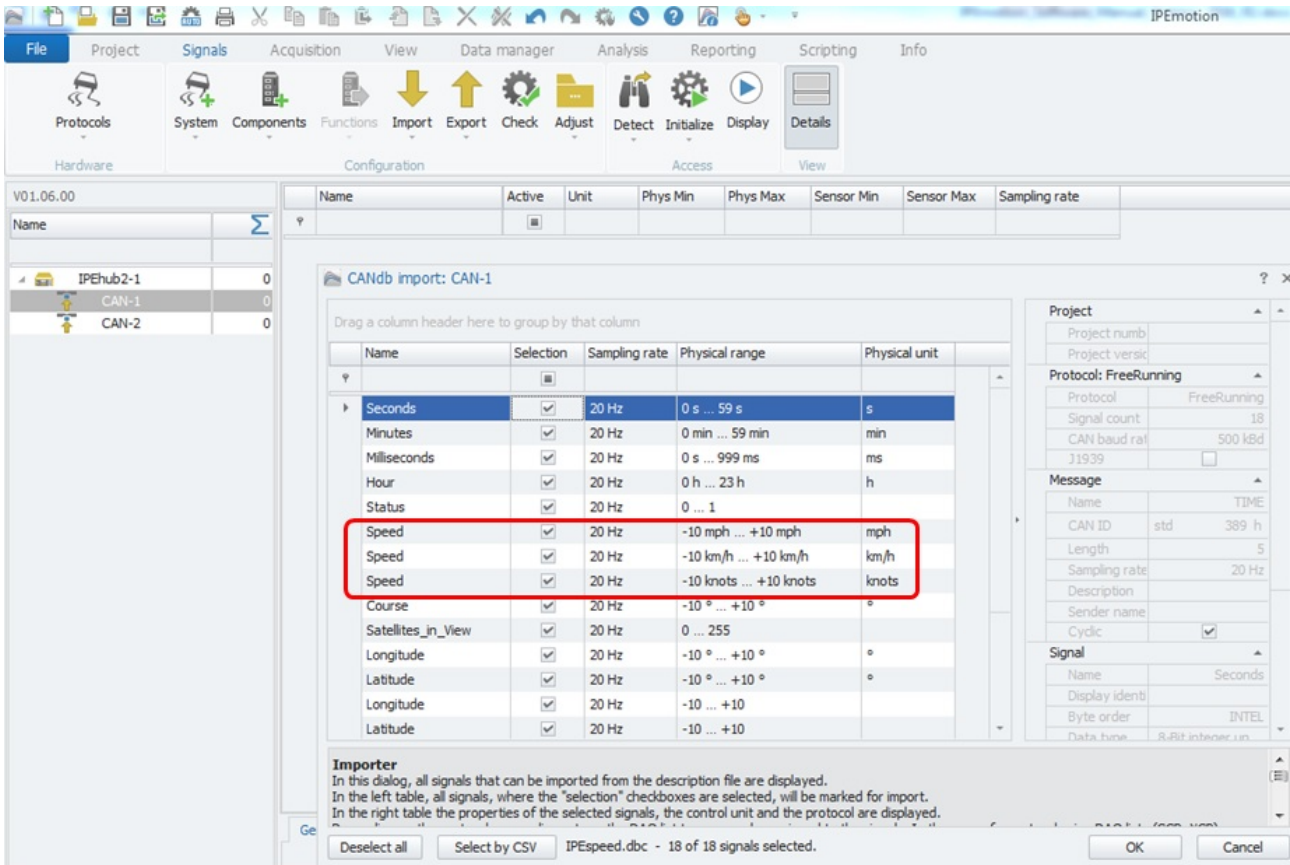
The new entry in the XML file should be:

```
<ImportSettings>
<checkForDuplicateSignalNames>True</checkForDuplicateSignalNames>
</ImportSettings>
```

The screenshot shows the Settings.XML file in a text editor. The left pane displays a tree view of settings categories, with 'ImportSettings' expanded and 'checkForDuplicateSignalNames' set to 'True'. The right pane shows the XML code, with the corresponding entry highlighted in a red box.

```
<Root>C:\Users\Public\Documents\IPETRONIK\IPEmotion\</Root>
<Database>C:\Users\Public\Documents\IPETRONIK\IPEmotion\Database</Database>
</UserDataDirectories>
<Acquisition>
<TimeSyncMode>0</TimeSyncMode>
</Acquisition>
<ViewSettings>
<StandardLayout>C:\Users\Public\Documents\IPETRONIK\IPEmotion\Reports\
<Skin>IPETRONIK-Bright-Skin</Skin>
<HudTransparency>0</HudTransparency>
<GlobalFontSize>9</GlobalFontSize>
</ViewSettings>
<CommonSettings>
<hardwareDetectionOnStart>False</hardwareDetectionOnStart>
<IncludeOptionsInConfiguration>True</IncludeOptionsInConfiguration>
<expertMode>True</expertMode>
<expertModeSettings>1966083</expertModeSettings>
<IncludeExternalFilesInConfiguration>True</IncludeExternalFilesInConfiguration>
<NodeNamePatternConfiguration>"Source" xx "Type" yy "Index" zz "Description"
</CommonSettings>
<LicenseSettings>
<serverKey>ZEP13-NP00D-KE06Z-00000-00KHW-00005-40040-00000-00000
<serverKey>DYS13-NP00D-KE06Z-00000-00FHW-00005-8W040-00000-00000
<serverKey>FHG13-NP00D-KE06Z-00000-00QKQ-00005-G0000-00000-00000
<serverKey>MXG13-NP00D-KE06Z-00000-00CKQ-00004-G0000-00000-00000
<serverKey>1HG13-NP00D-KE06Z-00000-00PKQ-00005-D0000-00000-00000
<serverKey>ZU211-20200-Z106Z-00000-00EKQ-00008-40000-00000-00000
<serverKey>KP111-20200-Z106Z-00000-00CKQ-00004-G0000-00000-00000
<serverKey>H7G13-NP00D-KE06Z-00000-00SAQ-00005-70000-00000-00000
<serverKey>8D803-6H40W-Y10ZZ-2E000-00Z7S-CZ202-00000-00000-00000
<serverKey>2R513-NP00D-KE05Z-00000-00FTD-00005-70000-00000-00000
<serverKey>8BF03-1000X-Y10ZZ-6H4Y0-00000-00002-10000-00000-00000
<serverKey>XEG17-NP00D-KE06Z-00000-003TP-00004-L2040-00000-00000
<serverKey>N2G13-NP00D-KE05Z-00000-00RRP-00004-50020-00000-00000
<serverKey>EMU03-20007-LP05Z-00000-00XTX-X7004-10000-00000-00000
</LicenseSettings>
<Climate>
<refpropPath>C:\Program Files (x86)\REFPROP\refprop.dll</refpropPath>
</Climate>
<ImportSettings>
<checkForDuplicateSignalNames>True</checkForDuplicateSignalNames>
<writeImportLogFile>False</writeImportLogFile>
```

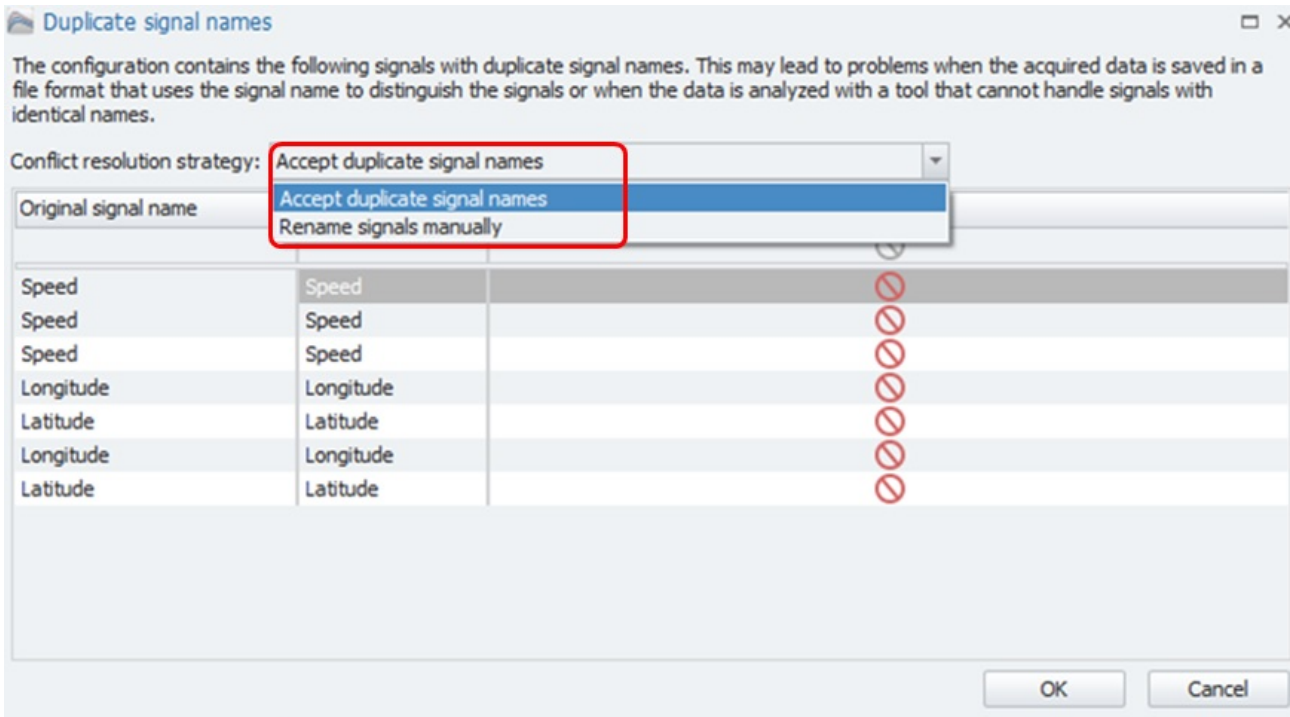
[IM_29]



Duplicate channel names are selected during the import.

[IM_30]

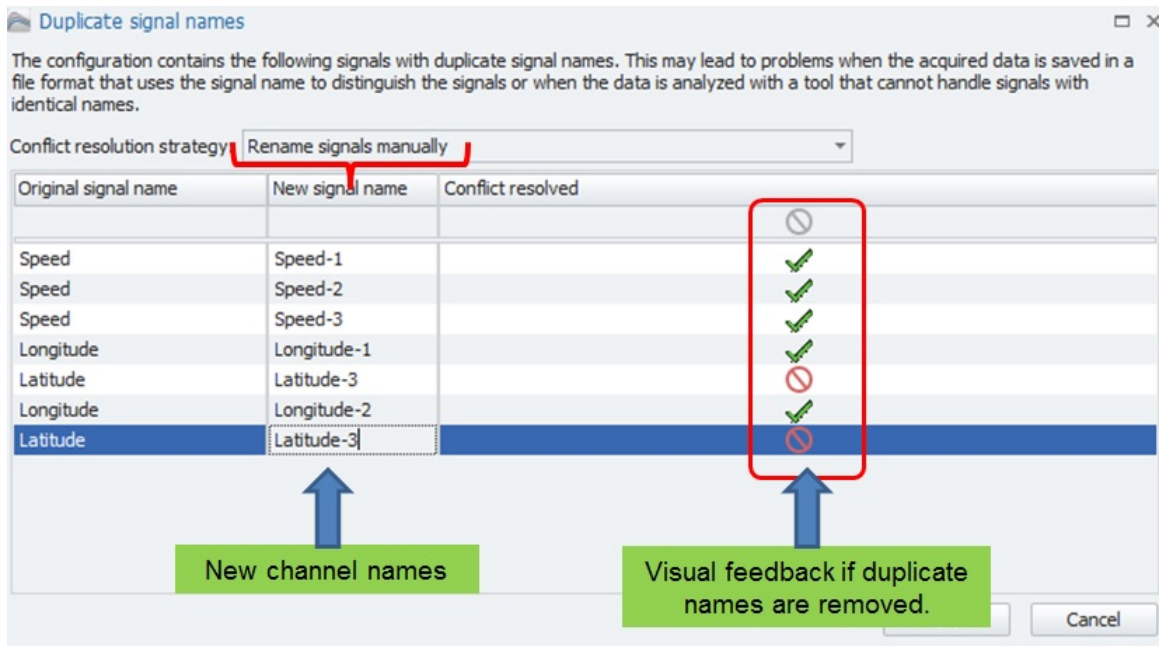
With the new entry in the Settings.XML file the import dialog will guide you to a new dialog highlighting all duplicate channels out of the selected channel list. There are two functions available in the drop down list:



Dialog to resolve duplicate channel entries.

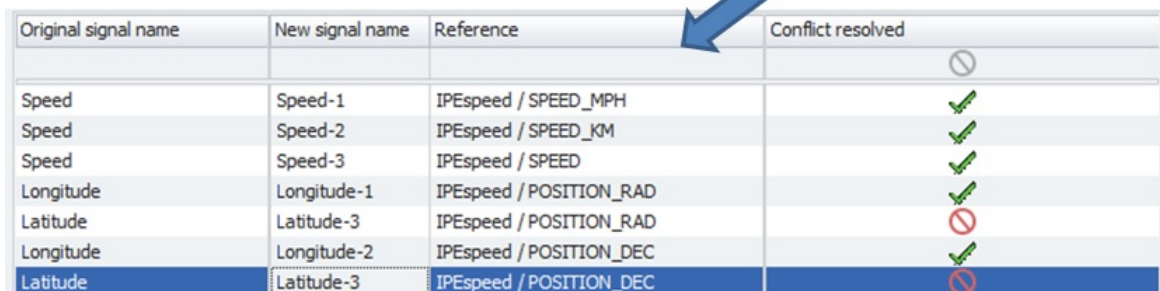
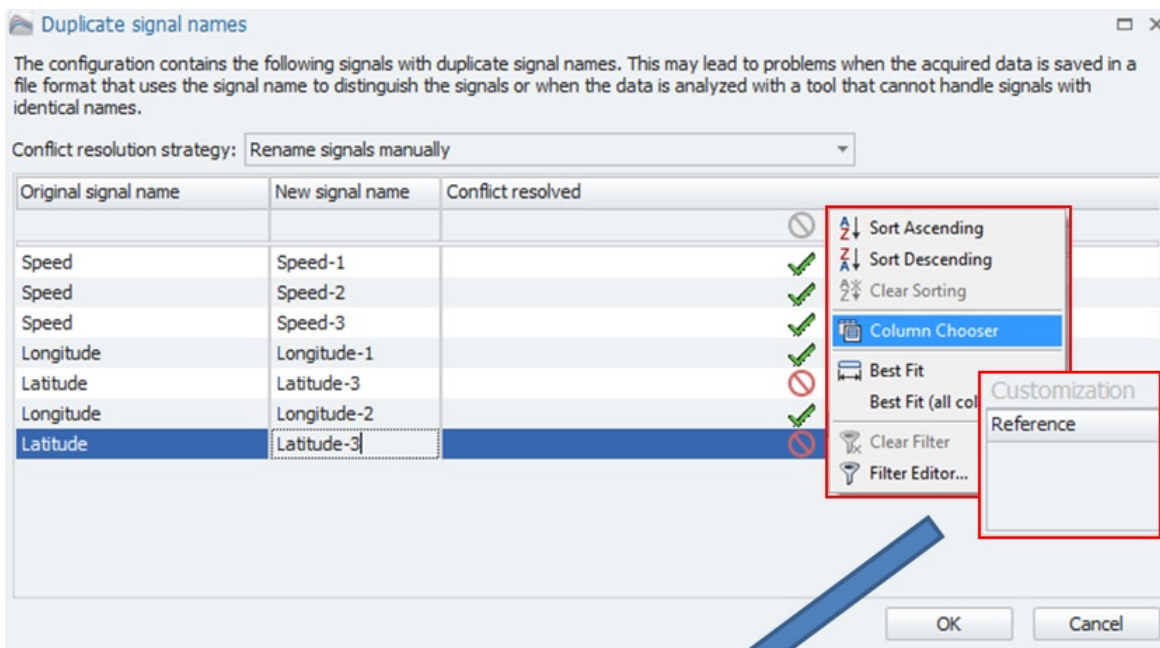
[IM_31]

- ▶ Accept duplicates With this function you accept the duplicates and confirm them
- ▶ Rename signals When you select the rename function you have to edit the channel names in the grid



[IM_32]

Apply column chooser to get information about the channel reference.



[IM_33]

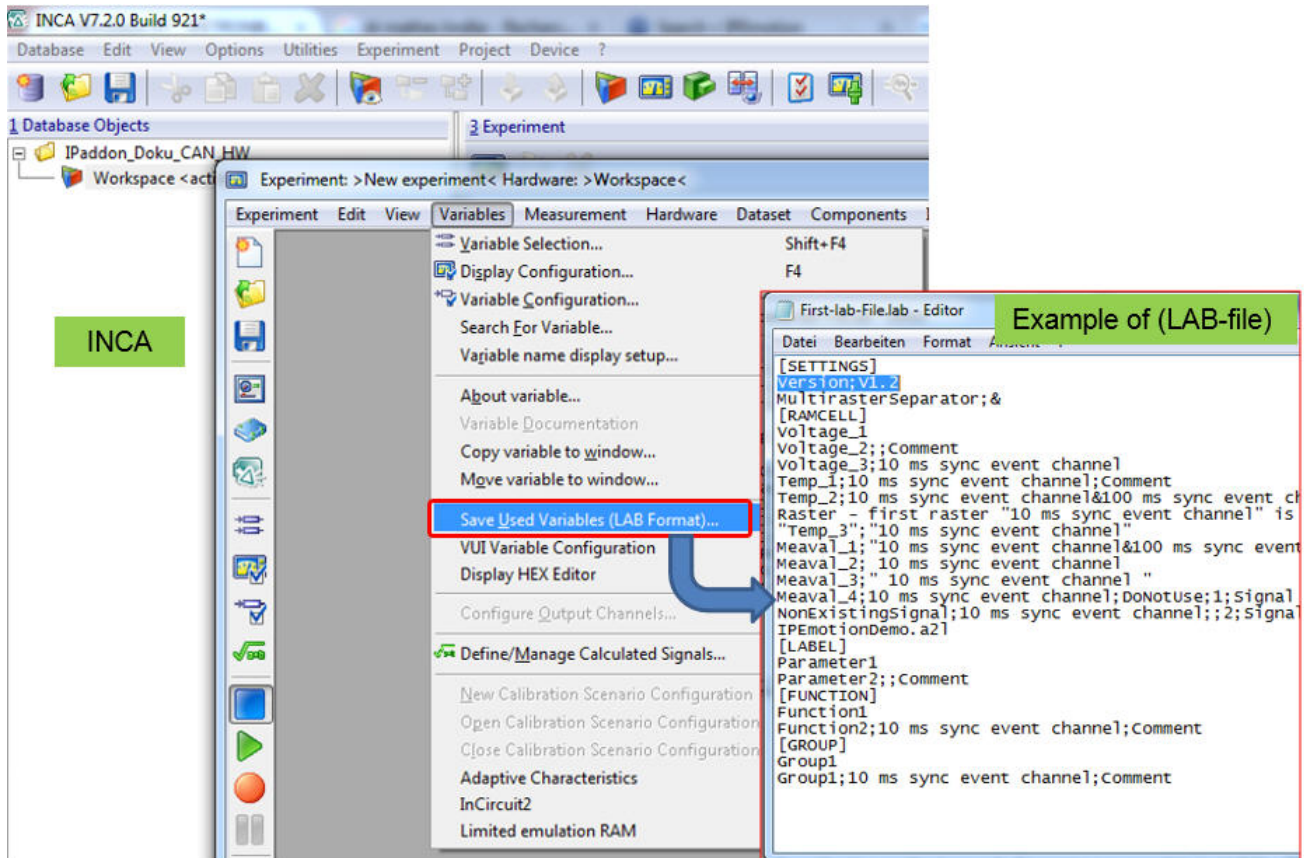
When all duplicate conflicts are resolved the import can be finalized and all channels are renamed.

V01.06.00		Name	Active	Unit
Name			<input type="checkbox"/>	
IPEhub2-1 18				
CAN-1 18				
IPEspeed 18				
	TIME 4		<input checked="" type="checkbox"/>	s
	STATUS 1		<input checked="" type="checkbox"/>	min
	SPEED... 1		<input checked="" type="checkbox"/>	ms
	SPEED... 1		<input checked="" type="checkbox"/>	h
	SPEED 2		<input checked="" type="checkbox"/>	mph
	SATELL... 1		<input checked="" type="checkbox"/>	km/h
	POSITI... 2		<input checked="" type="checkbox"/>	knots
	POSITI... 2		<input checked="" type="checkbox"/>	°
	DATE 3		<input checked="" type="checkbox"/>	
	ALTTU... 1		<input checked="" type="checkbox"/>	
CAN-2 0				
		Seconds	<input checked="" type="checkbox"/>	s
		Minutes	<input checked="" type="checkbox"/>	min
		Milliseconds	<input checked="" type="checkbox"/>	ms
		Hour	<input checked="" type="checkbox"/>	h
		Status	<input checked="" type="checkbox"/>	
		Speed-1	<input checked="" type="checkbox"/>	mph
		Speed-2	<input checked="" type="checkbox"/>	km/h
		Speed-3	<input checked="" type="checkbox"/>	knots
		Course	<input checked="" type="checkbox"/>	°
		Satellites_in_View	<input checked="" type="checkbox"/>	
		Longitude-1	<input checked="" type="checkbox"/>	°
		Latitude-3	<input checked="" type="checkbox"/>	°
		Longitude-2	<input checked="" type="checkbox"/>	
		Latitude-4	<input checked="" type="checkbox"/>	
		Year	<input checked="" type="checkbox"/>	
		Month	<input checked="" type="checkbox"/>	
		Day	<input checked="" type="checkbox"/>	
		Altitude	<input checked="" type="checkbox"/>	m

Channels are all renamed. [IM_34]

5.9 Description file import with INCA LAB file

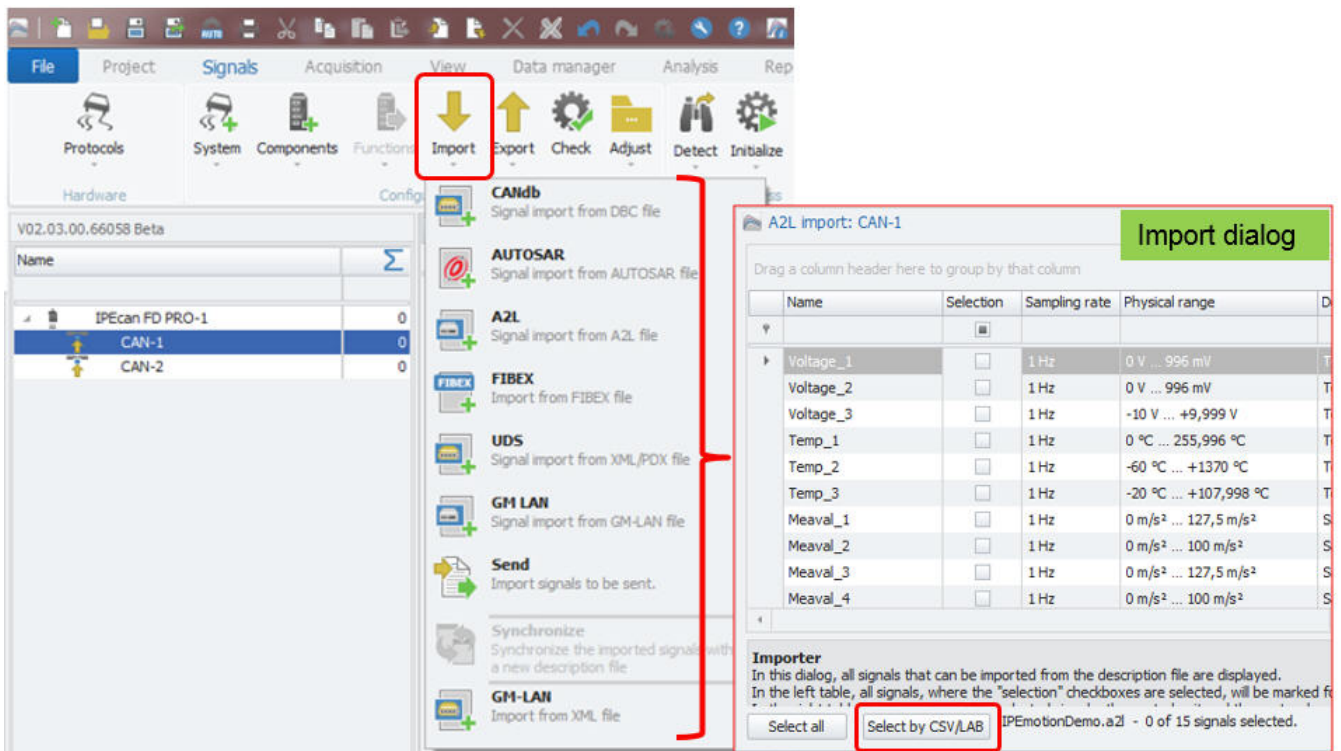
To support the workflow between INCA and IPEmotion measurements an INCA experiment (LAB-file) import is supported besides the CSV import. You can save part of your INCA experiment in the LAB file format. This LAB file can be used to create easily the same measurement configuration for IPEmotion.



INCA > Export (LAB-file) of INCA experiment

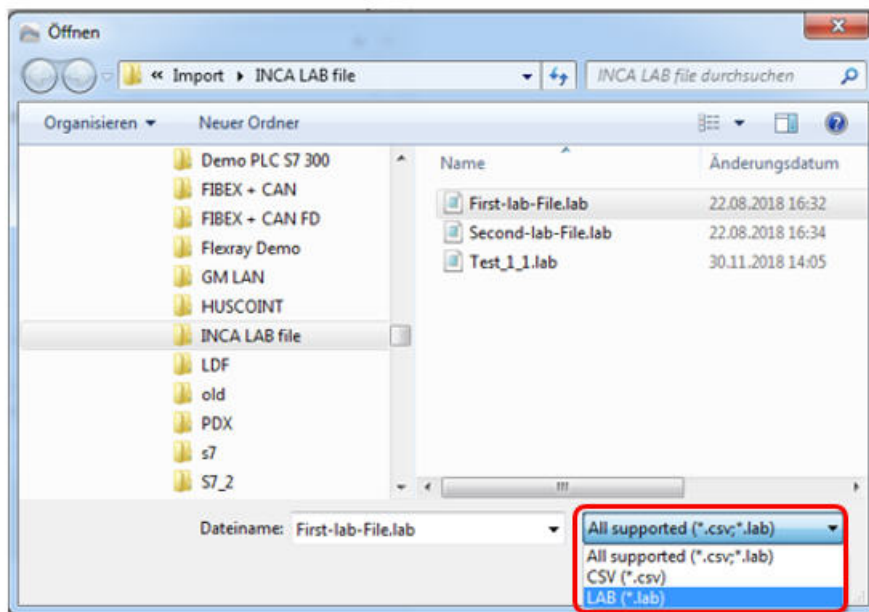
[IM_35]

All description file imports on all PlugIns support the CSV and LAB file filter.



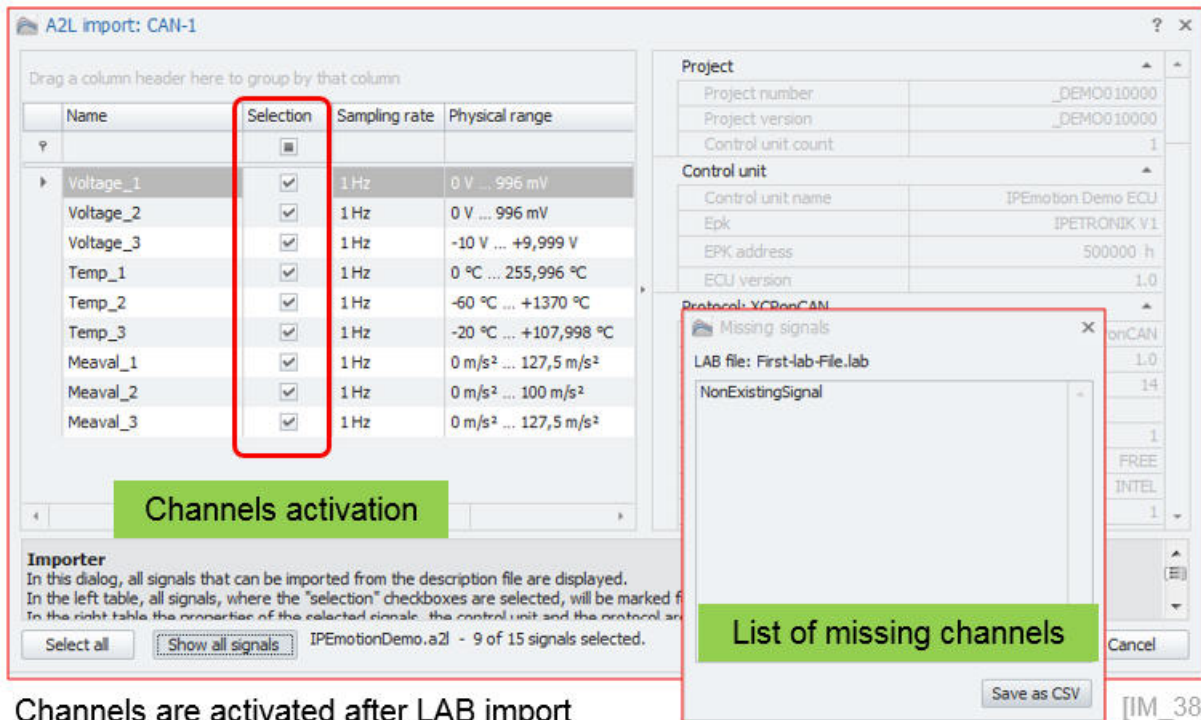
All IPEmotion description file imports support CSV / LAB filters for channel selection [IM_36]

After you have imported your complete description file you can use the CSV and LAB reference to select only those channels which are relevant for your measurement application.



Example: Select LAB-file for channel activation [IM_37]

When the LAB file was selected the associated channels from the description file are automatically activated in the import dialog. A message box is also returning all channels which are included in the LAB-file and missing in the description file.



Channels are activated after LAB import

[IM_38]

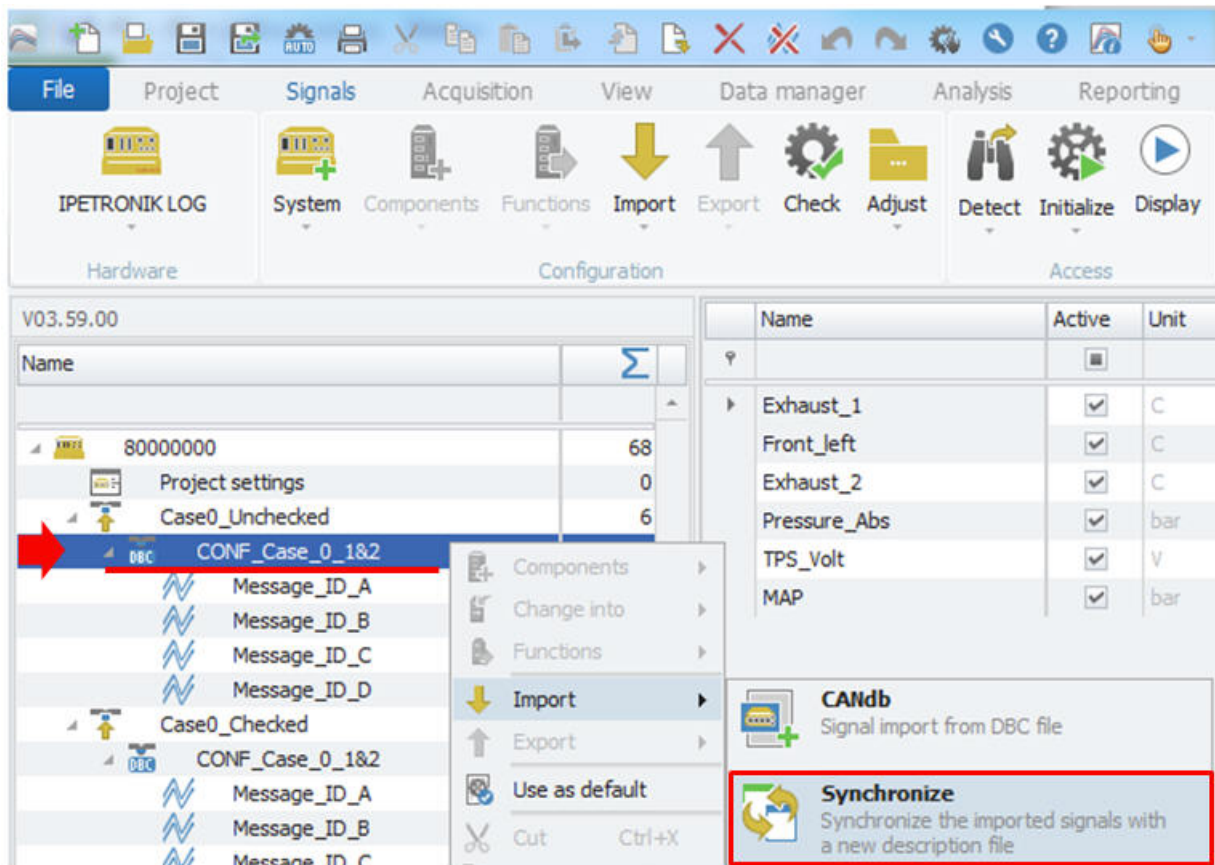
5.10 Synchronization of description files

You can use the description file synchronization function to compare your existing measurement configuration against a new description file. The description file synchronization is useful to get a direct view about the differences between a new ECU description file and your current measurement configuration. The following file formats are supported for synchronization:

File Format	Object type	Protocol	Properties
A2L	Signal	CCP, XCP, KWPOncAN	Name
A2L	DAQ-Liste	CCP, XCP	DaqType, ListNumber, EventNumber
A2L	Characteristic	CCP, XCP, KWPOncAN	Name
AutoSar (ARXML)	Botschaft	FreeRunning	Name
AutoSar (ARXML)	Frame	FlexRay	Name
AutoSar (ARXML)	Signal	FreeRunning	Name, Message
AutoSar (ARXML)	Signal	FlexRay	Name, Frame
DBC	Botschaft	FreeRunning	Name
DBC	Signal	FreeRunning	Name, Message
DBC	Signal	J1939	Name, Message-Sender
DBC	Signal	GMLan	PID
FIBEX (XML)	Frame	FlexRay	Name
FIBEX (XML)	Signal	FlexRay	Name, Frame
IDF	Job	KWP-, UDS-Diagnose	Name
LDF	Botschaft	LIN	Name
LDF	Signal	LIN	Name, Message
PDX	Job	KWP-, UDS-Diagnose	Name
UDS-Messen (XML)	Signal	UDS	Name

Supported files for synchronization [IM_41]

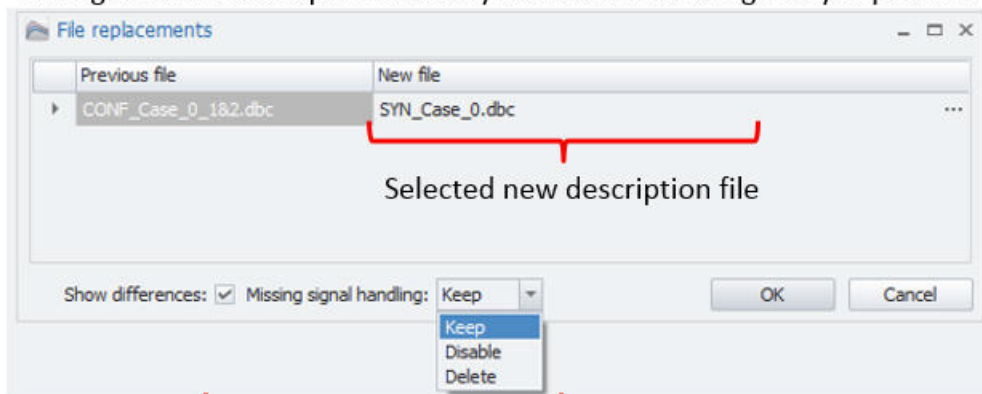
The default setting of the file synchronization is based on message ID and message name. In the standard setup the description file is compared on Message ID and channel name. When one of two parameters is different, the synchronization will not take place and you will get a notification in the message window. The standard synchronization behavior can cause an additional workload when e.g. a channel is switched to a different message ID because then the synchronization does not take place and the related links to the VIEW instruments and formulas or storage in the ACQUISITION work space etc. is removed as well. In the following example the default synchronization behaviour is presented by selecting the synchronization function on description file level.



Synchronize function on decription file level [IM_42]

In the next step you select the description file you like to compare against and the processing treatment operation. Here you can enable a message window to get an overview of the differences detected and how to treat the differences. In this example we will get an message window to see the differences. The treatmetn selcted is: Deactivate the channels which cannot be synchronized

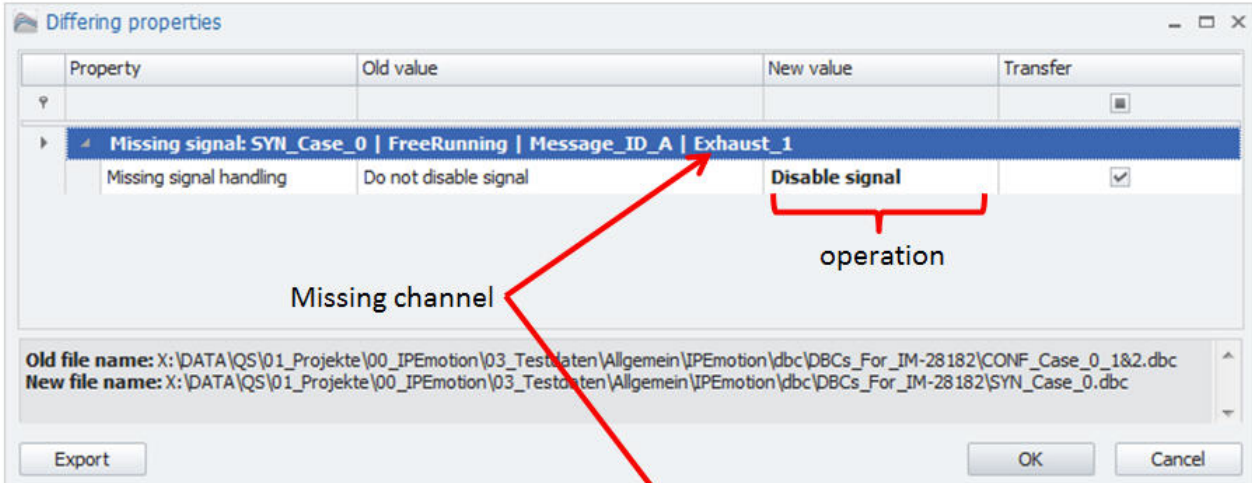
Dialog to select description file to sync and define configure sync process.



Defined actions after synchronization [IM_43]

The message popup window is now indicating that the channel Exhaust1 could no tbe synchronized and was therefore accordingly to the treatment operation was deactivated.

Message window indicating the differences and the selected treatment



Message dialog indicating the differences too.

[IM_44]

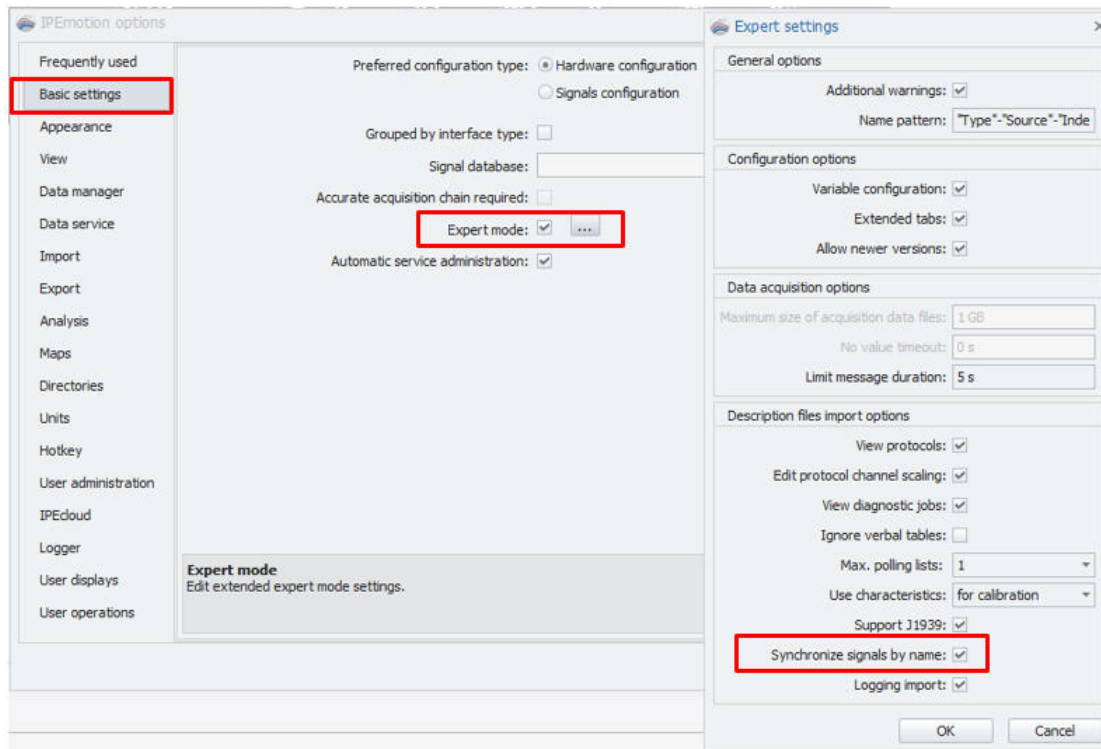
Symbol	Time	Type	Source	Message
	15.11.2017 09:42:02,016	INFORMATION	Exhaust_1	The signal "Exhaust_1" of the message "Message_ID_A" is not included in the description file "SYN_C
	15.11.2017 09:42:02,016	INFORMATION	Import	Following signals are not defined in the new description file: Exhaust_1

The configuration is now indicating the deactivated channel Exhaust 1.

Name	Active	Unit
Exhaust_1	<input type="checkbox"/>	C
Front_left	<input checked="" type="checkbox"/>	C
Exhaust_2	<input checked="" type="checkbox"/>	C
Pressure_Abs	<input checked="" type="checkbox"/>	bar
TPS_Volt	<input checked="" type="checkbox"/>	V
MAP	<input checked="" type="checkbox"/>	bar

Result: missing channel is deactivated [IM_45]

With the EXPERT setting "Synchronize signals by name" the synchronization mechanism is focusing only on the channel name and not any more on the message ID and channel name together. The benefit for the user is that the synchronization will take place and the links to the VIEW instruments etc. remain intact to display online data.



[IM_46]

The screenshot below is indicating on the right hand side, that the channel Exhaust 1 was successfully synchronized even so that it is included on a different CAN message ID.

Synchronization: Message ID & Channel Name

V02.03.00.67284 RC			Name	Active
Name			Exhaust_1	<input type="checkbox"/>
IPEcan PRO-1			Front_left	<input checked="" type="checkbox"/>
CAN-1			Exhaust_2	<input checked="" type="checkbox"/>
SYN_Case_0			Pressure_Abs	<input checked="" type="checkbox"/>
Message_ID_A			TPS_Volt	<input checked="" type="checkbox"/>
Message_ID_B			MAP	<input checked="" type="checkbox"/>
Message_ID_C				
Message_ID_D				
CAN-2				

V02.03.00.67284 RC			Name
Name			Exhaust_1
IPEcan PRO-1			Front_left

Synchronization: Only by Channel Name

V02.03.00.67284 RC			Name
Name			Exhaust_1
IPEcan PRO-1			Front_left
CAN-1			Exhaust_2
SYN_Case_0			Pressure_Abs
Message_ID_A			TPS_Volt
Message_ID_B			MAP
Message_ID_C			
Message_ID_D			
CAN-2			

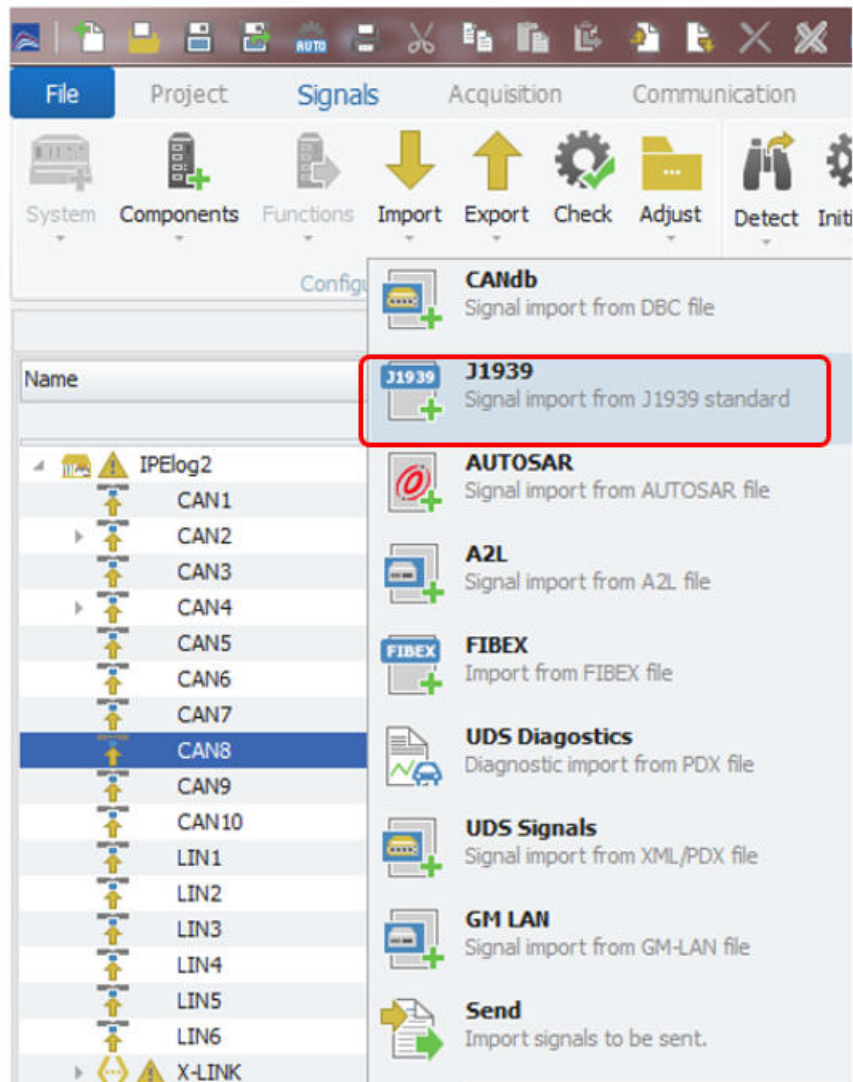
V02.03.00.67284 RC			Name
Name			Exhaust_2
IPEcan PRO-1			Exhaust_1

[IM_47]

5.11 J1939 Diagnostics

IPEmotion 2020 R1 - Newly implemented J1939 Features: <https://youtu.be/dyqk3jh2V5c>

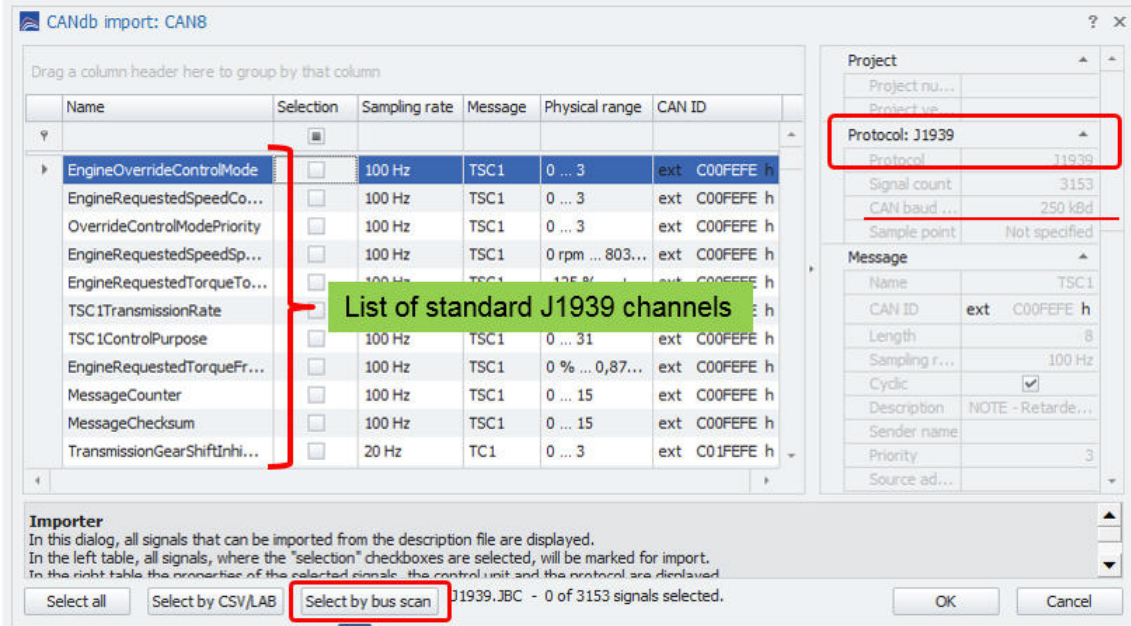
You need to select on your CAN interface of your CAN card or data logger the J1939 diagnostic function. In order to use the J1939 protocols function on a logger a license is required. For more details see RT licensing table. ??



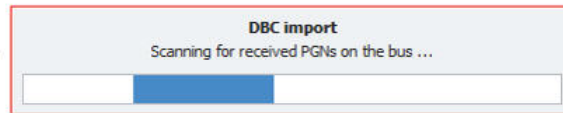
J1939 diagnostic measurement

[IM_50]

The J1939 import dialog provides a list of over 3000 default channels defined in the standard. With the bus scan functionality, you can identify which channels are available on the connected ECU. The default CAN bus baud rate is set during the bus scan to 250 kB.

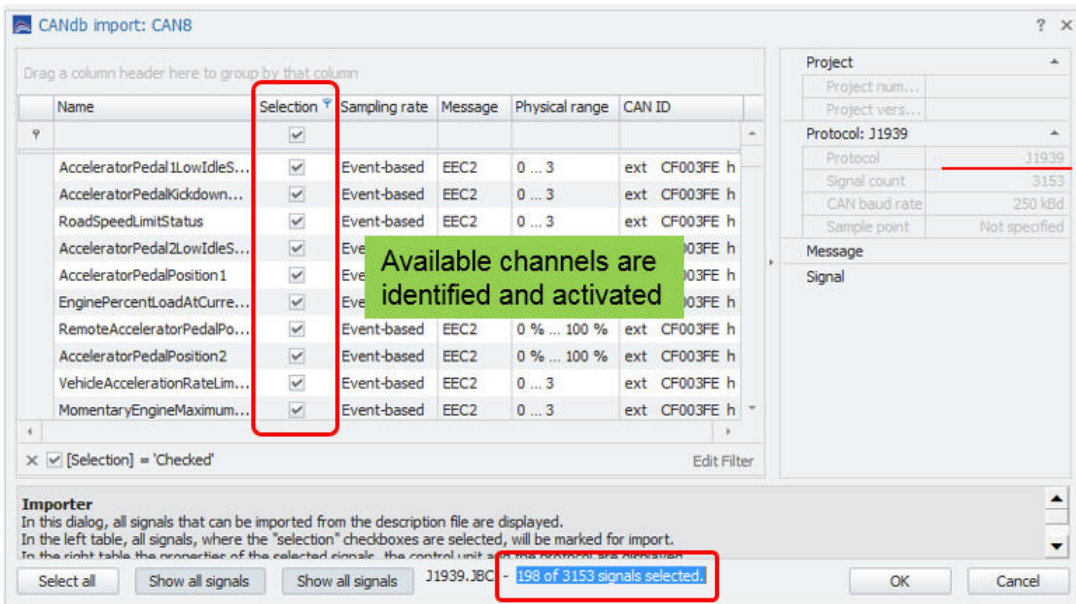


J1939 bus scan function



[IM_51]

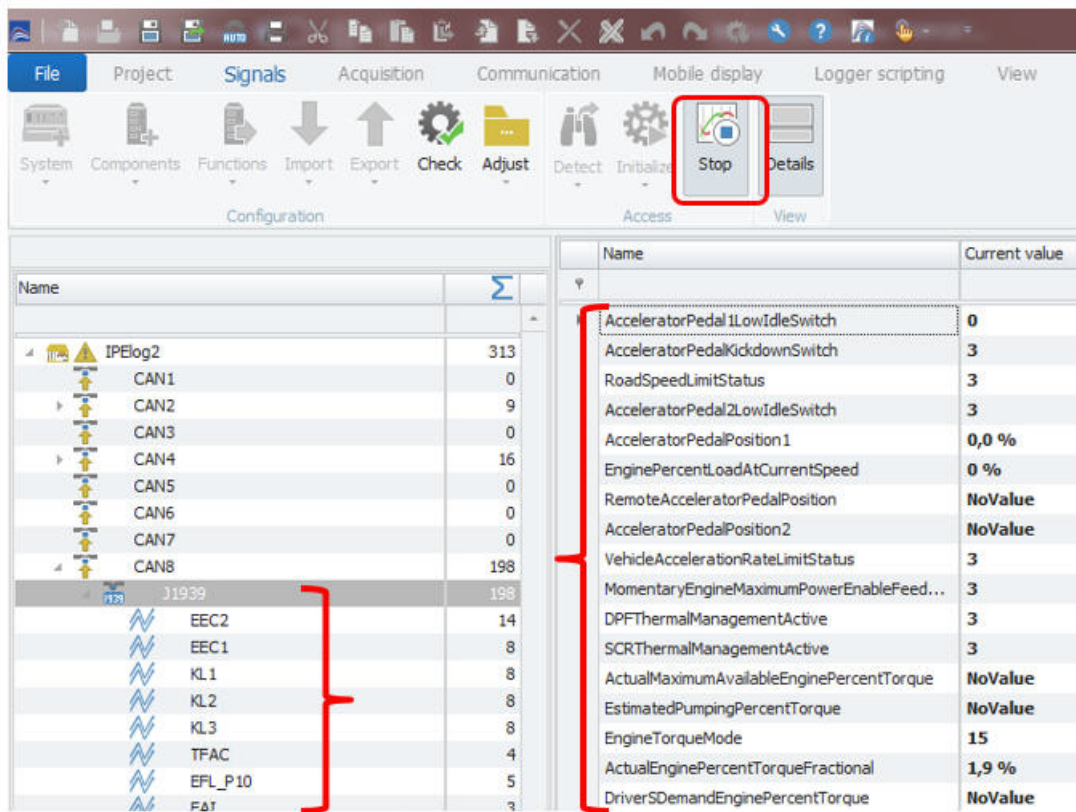
When the bus scan is performed the available channels are activated.



J1939 bus scan result

[IM_52]

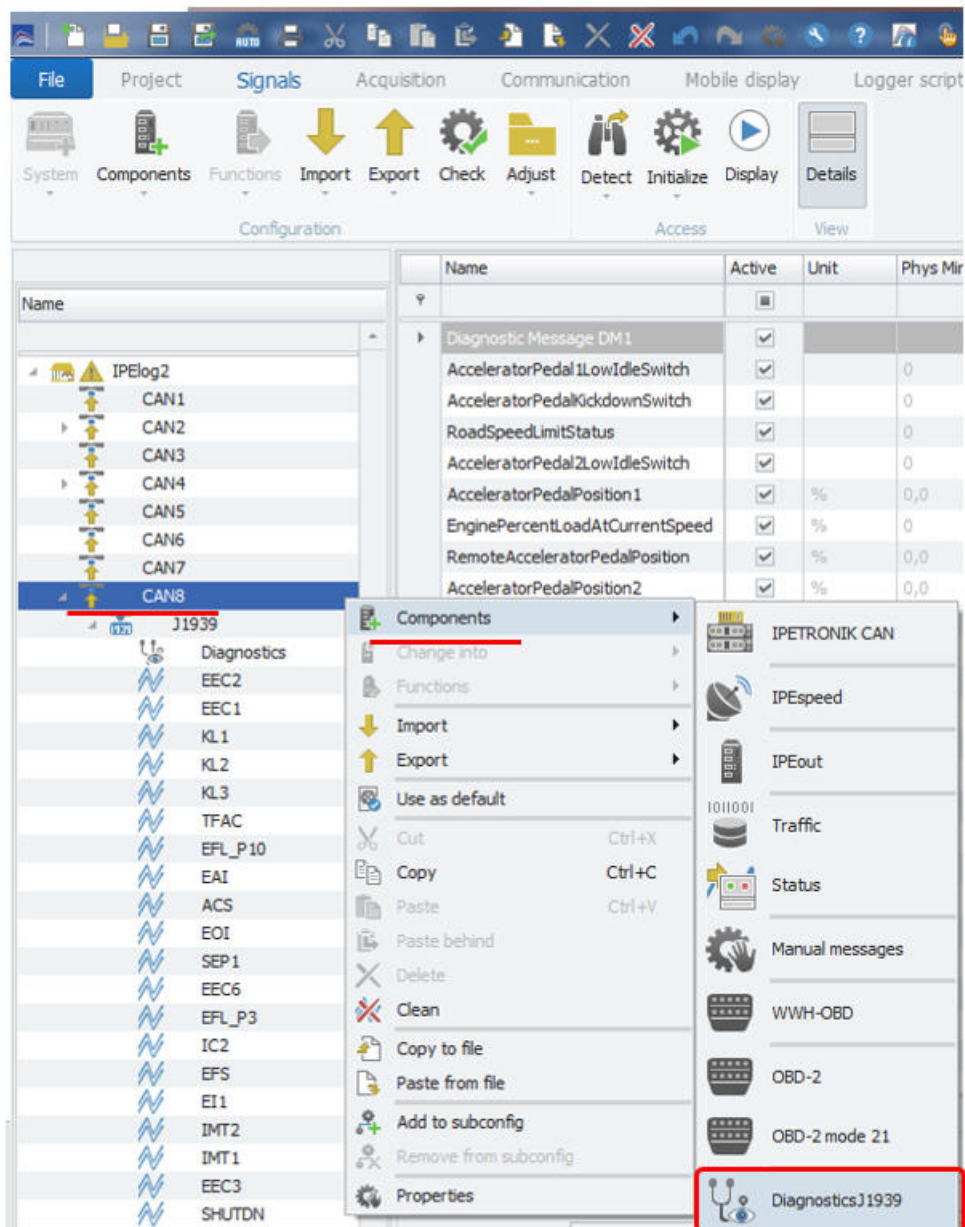
When you start the measurement the actual messages and channel readings are displayed.



J1939 online data of available messages and channel readings

[IM_53]

On the J1939 interface you can select diagnostic messages too.

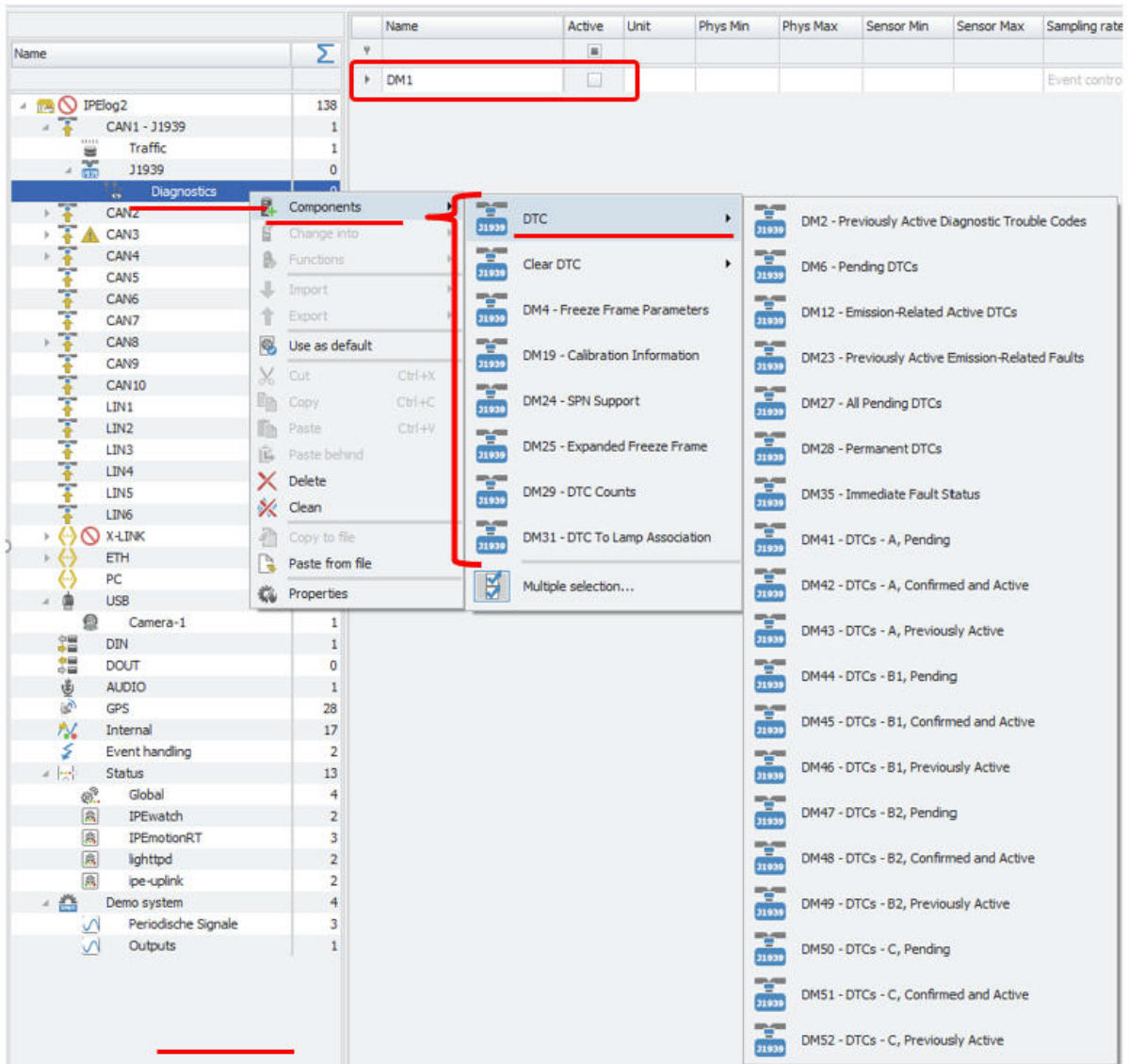


J1939 Diagnostics

[IM_54]

When the channel is activated you select from a wide range of components. The DM1 channel is created by default.

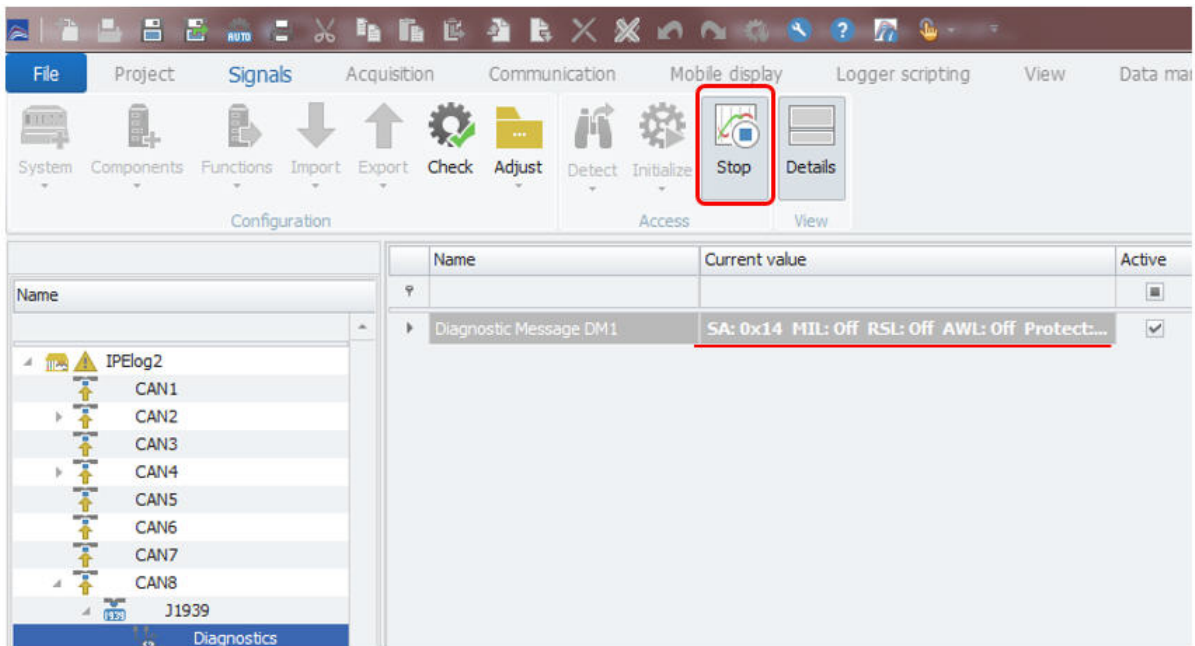
- ▶ DTC: 1, 2, 6, 12, 23, 27, 28, 35, 41, 42, 43, 44, 45, 46, 47,48, 49, 50, 51, 52
- ▶ Clear DTC: 3, 11
- ▶ DM 4, 19, 24, 25, 29, 31
- ▶ Multiplexed signals are supported too



DM1 is available by design – additional messages are listed

[IM_55]

The readability of the data in the channel is very limited. Therefore a dedicated diagnostic instrument is available. This instrument is explained in chapter ??

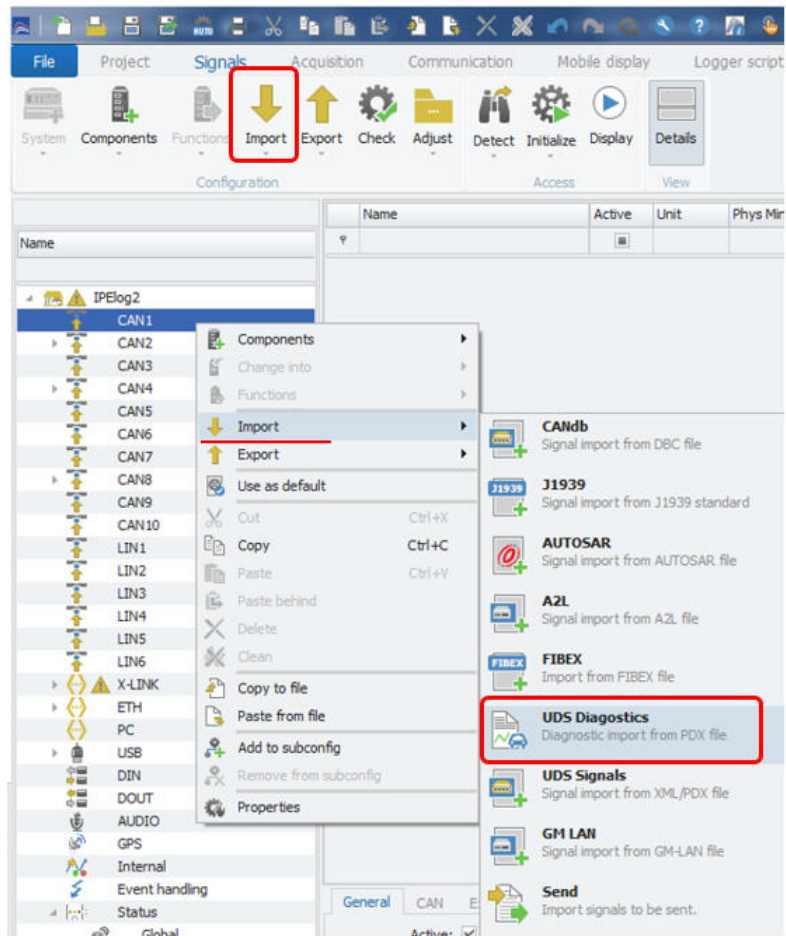


DM01 Diagnostic channel readings

[IM_56]

5.12 UDS Diagnostics

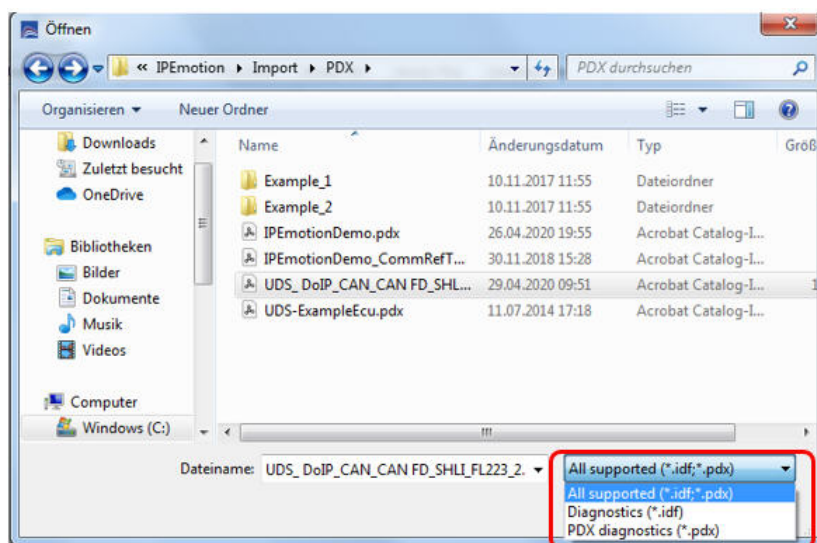
On CAN and ETH interfaces you can run UDS diagnostic measurements.



UDS Diagnostic import

[IM_60]

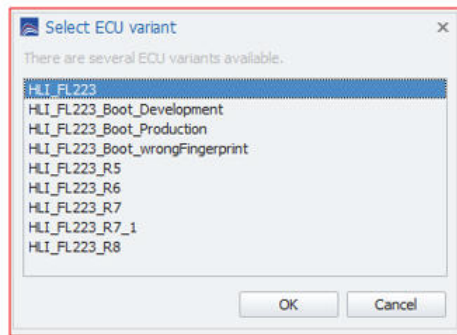
The .PDX and .IDF import file format is currently supported.



UDS Diagnostic import from .IDF and .PDX file

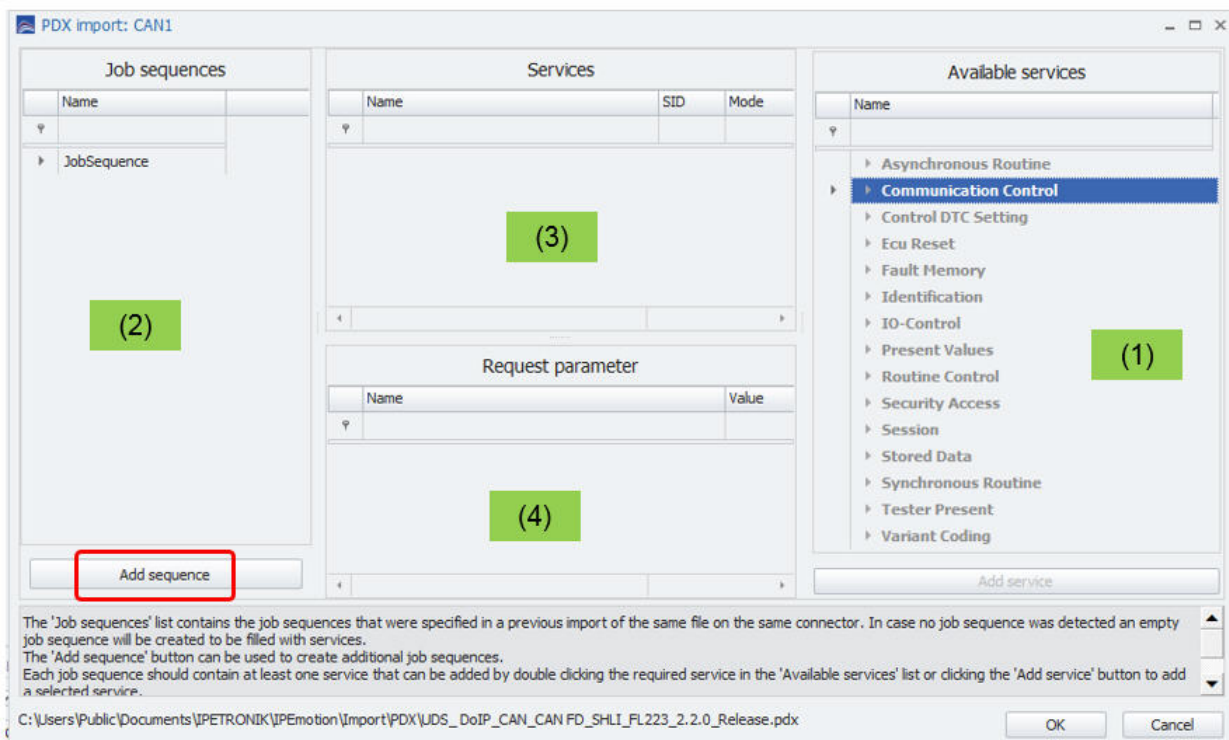
[IM_61]

The PDX file can contain several software versions. You need to select the one relevant for the ECU in place.



Select your ECU variant [IM_62]

In the last stage you reach the jobs and service configuration dialog. This dialog consists of 4 distinctive areas with different functions.



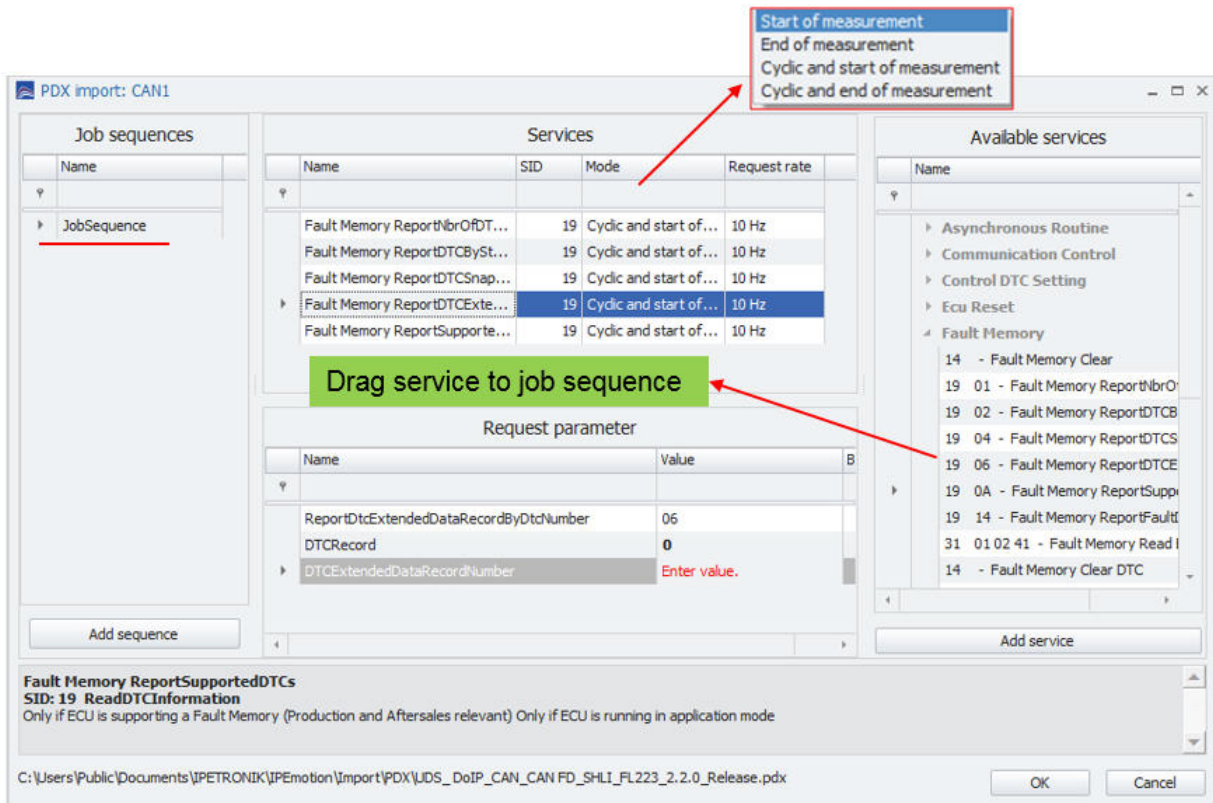
Jobs and services configuration dialog

[IM_63]

- ▶ Area 1 This includes all diagnostic services provided by the PDX file.
- ▶ Area 2 Here you define the job sequences. You can have several sequences and trigger them individually.
- ▶ Area 3 Here you drag and drop the service you like to execute.
- ▶ Area 4 Here you can define for some specific jobs which require a output values the corresponding parameter to get data back from the ECU.

In this example a job sequence with several service is configured. Within a job sequence the service can operate in different modes.

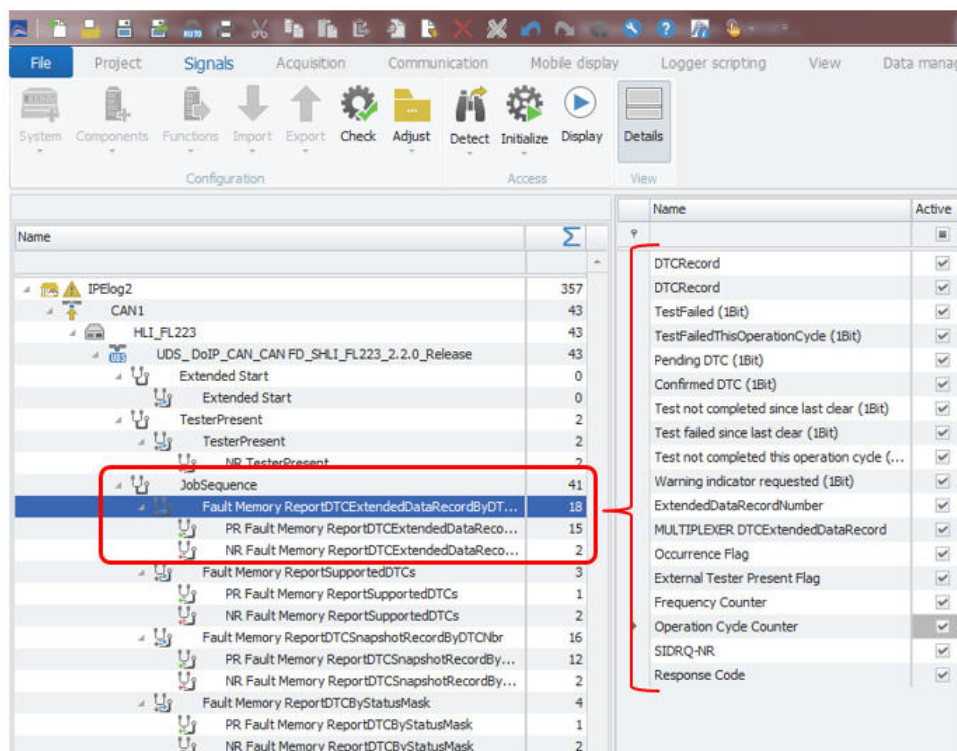
- ▶ Start of measurement
- ▶ End of measurement
- ▶ Cyclic and Start of measurement
- ▶ Cyclic and End of measurement



Jobs configuration dialog

[IM_64]

When you have activated in OPTIONS > EXPERT MODE > View Diagnostics Jobs (see section ??) you can see all configured job sequences and the selected services. For each service a list of channels for PR = positive responses and NR = negative responses is listed.

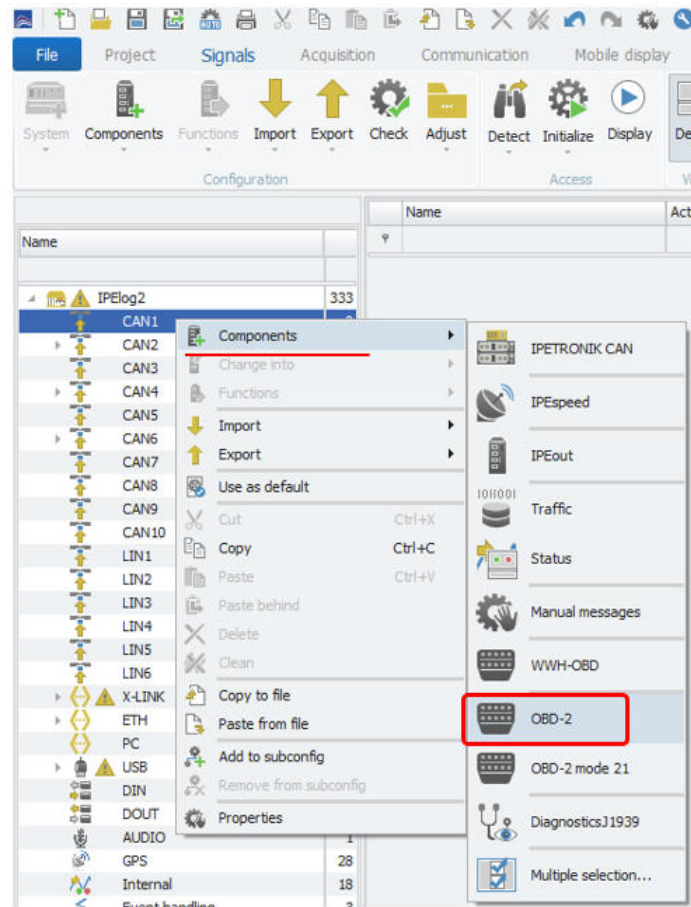


Channels with responses of each service are listed

[IM_65]

5.13 OBD-2 diagnostics

On all CAN interfaces you can measure data from the standard OBD-2 protocol to get standard vehicle diagnostic information.



OBD-2

[IM_70]

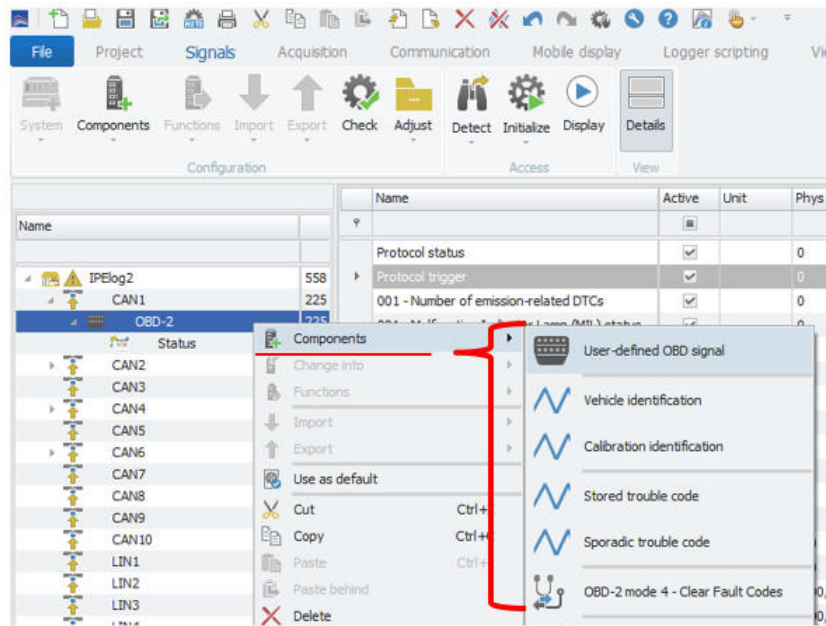
When you activate the OBD-2 measurement over 200 standard OBD channels are created. You can then activate the channels you require. Not all channels will provide data as this is vehicle manufacturer specific.

Name	Active	Unit
Protocol status	<input checked="" type="checkbox"/>	
Protocol trigger	<input checked="" type="checkbox"/>	
001 - Number of emission-related DTCs	<input checked="" type="checkbox"/>	
001 - Malfunction Indicator Lamp (MIL) status	<input checked="" type="checkbox"/>	
001 - Misfire monitoring supported	<input checked="" type="checkbox"/>	
001 - Fuel system monitoring supported	<input checked="" type="checkbox"/>	
001 - Comprehensive component monitoring ...	<input checked="" type="checkbox"/>	
001 - Compression ignition monitoring suppor...	<input checked="" type="checkbox"/>	
001 - Misfire monitoring ready	<input checked="" type="checkbox"/>	
001 - Fuel system monitoring ready	<input checked="" type="checkbox"/>	
001 - Comprehensive component monitoring ...	<input checked="" type="checkbox"/>	
002 - DTC that caused required freeze frame...	<input checked="" type="checkbox"/>	
004 - Calculated load value	<input checked="" type="checkbox"/>	%
005 - Engine coolant temperature	<input checked="" type="checkbox"/>	°C

OBD-2 Standard channels

[IM_71]

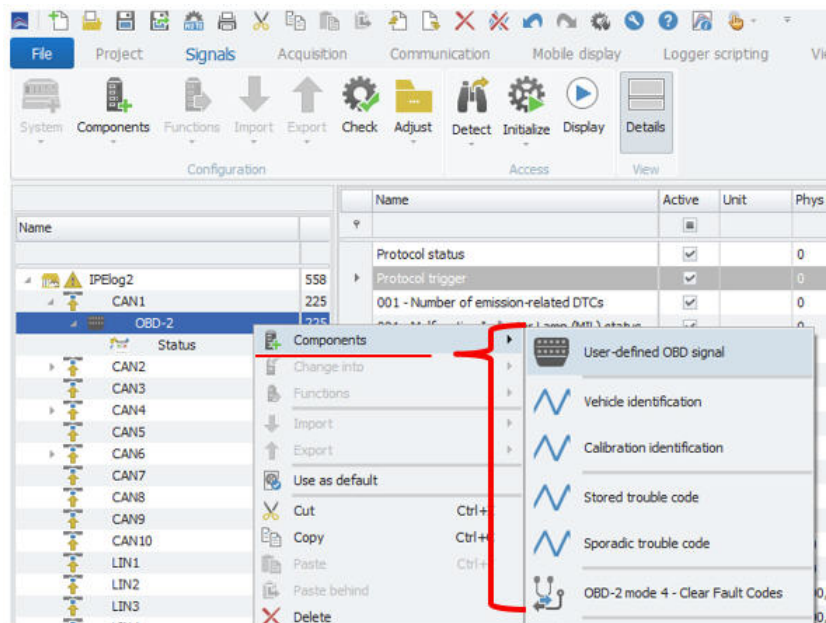
Besides the standard OBD-2 channel you can activate additional components for measurement as indicated below.



OBD-2 additional components

[IM_72]

With the user defined signals, you can request PID which are outside of the standard and not available to the public. The identification and trouble code channels you can get additional information too. With mode 4 you can use a trigger channel to clear fault codes.



OBD-2 additional components

[IM_72]

Author: FOT