Manual

IPEmotion PlugIn CAETEC datalog

15.10 Version 1.0

Date of issue: May 16, 2018

CAETEC GmbH
Industriestrasse 1
82140 Olching
Germany

Pictures and sketches are for illustration purposes only and are not to be used as design drawings nor to serve as offer or assembly drawings.

All specifications are based on the technical status of May 16, 2018. We reserve the right to make any changes required to technically improve the equipment.

This manual has been produced with all due diligence.

CAETEC shall not be held liable for any damage resulting from the use of this manual, providing it is not due to gross negligence on our own part or the part of our legal representative or vicarious agent, and to the extent that the damage does not stem from personal injury, bodily harm or damage to health.

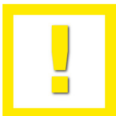
Contents

1 Foreword

2 Configuration

2.1 Symbols

Various paragraphs in this manual are marked with special symbols. These symbols have the following meanings:



This symbol highlights important information that, if ignored, may prevent successful use of the program.

This symbol refers to additional information supplementing this manual.

2.2 References

References to other sections of this manual are generally placed in brackets and are indicated by an arrow:
(→ 2.1) refers to Section 2.1.
When this manual is read in digital form, a mouse-click on such a reference accesses the particular section of the book.

3 Product description

3.1 Installation

3.1.1 System requirements

Minimum:
Screen resolution: 1080 x 800 pixel
Processor: 2 GHz
RAM: 2048 MB
DirectX 9

Recommended:
Screen resolution: 1920 x 1200 pixel
Processor: 3 GHz Multi-Core
RAM: 6144 MB
Storage medium type: SSD
DirectX 11

Supported platforms:
Microsoft Windows 10 (32 Bit and 64 Bit operating systems)
Microsoft Windows 8.1 (32 Bit and 64 Bit operating systems)
Microsoft Windows 8 (32 Bit and 64 Bit operating systems)
Microsoft Windows 7 (32 Bit and 64 Bit operating systems)

The following additional software is required:
Microsoft .NET 4.5.1 Framework

3.1.2 Where to get the installation file?

The installation file can be downloaded from https://myipe.ipetronik.com/ or here https://www.ipetronik.com/software/plugins. Once you have logged in with your username and password you will be able to locate the file in the section **UP- & DOWNLOADS**.

3.1.3 How to know the right version?

The format of the Plugin version is always Vxx.xx.xx (e.g. V15.10.00) and has to match the target system's dataLog firmware-version. To find out which firmware-version your target system is running you can either use the the data logger's web interface or open with a text editor the logger's logfile and look for the firmware version there. For further information please also refer to your data logger's documentation.

> **!** Multiple Plugin versions of the CAETEC Plugin for IPEmotion can be installed at the same time. This allows you to work with various data loggers that are running on different firmware versions. Once you have multiple Plugin versions installed, you will always have to ensure that your currently used Plugin version and target system's firmware version match. Please refer to **Switching between Plugin versions** (→ 3.2.2).

3.1.4 Installation on Windows

Navigate to your download directory and extract the downloaded zip-file. For 32-bit versions of Microsoft Windows choose the following file for installation.

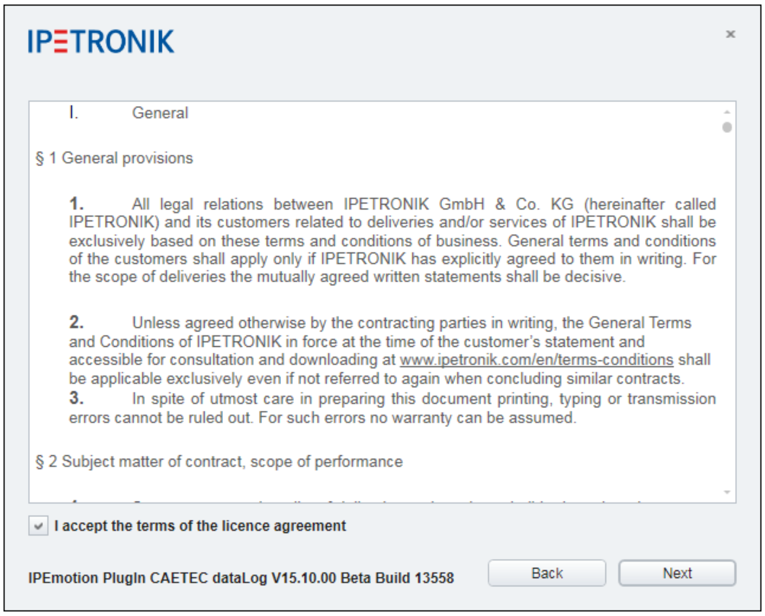
Setup IPEmotion PlugIn CAETEC dataLog.exe	30.10.2017 09:21	Anwendung	32.090 KB

For 64-bit versions of Microsoft Windows choose the following file for installation.

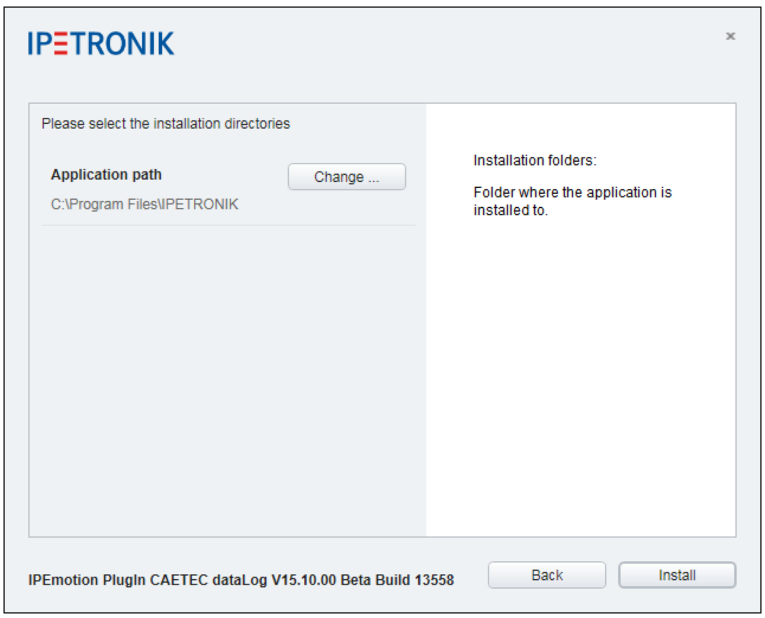
Setup IPEmotion PlugIn CAETEC dataLog (x64).exe	30.10.2017 09:21	Anwendung	25.966 KB

You may be asked to confirm execution of the program. If so, please confirm and type in your Windows user password if asked to do so.
In the next window you may choose the language in which you wish to install the plugin. The language can be changed later on (→ 3.2.2).

The next window presents the terms of the licence agreement. Check the box **I accept the terms of the licence agreement** and proceed to the next window by clicking next.



In the next window you select the target folder where the plugin is to be installed. The standard installation path is set as default. Normally you can simply accept it. Click Install to continue and start the installation process. Again you may be asked for confirmation and/or your Windows user or admin password. Please type it in and click OK to continue.



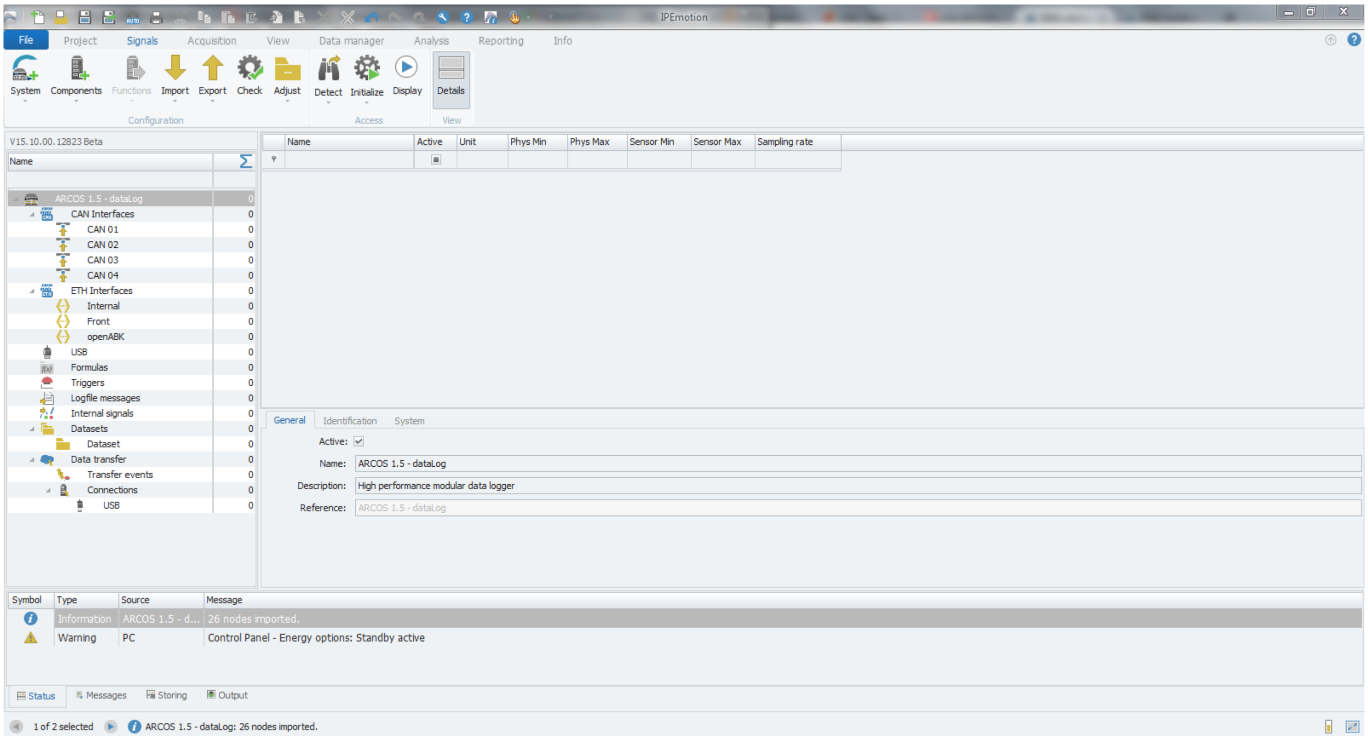
Once installation has been comleted successfully you will need to acitvate the Plugin in the Options dialogue in order start working with it. To do so please refer to the section **Activating the plugin** (→3.2.2).

> **!** After completion of the installation we strongly advice you to acitvate the expert mode and afterwards activate the extended tabs option and change the number of maximum polling lists in the expert settings. For detailed information refer to the sections **Expert mode, Extended tabs and Maximum polling lists** (→ 3.2.2).

> **!** The plugin allows for a customizable user guidance by giving you the option to choose which tree elements you would like to be available in the measurement task tree. For detailed information refer to the section **Customizing tree elements** (→ 3.2.2).

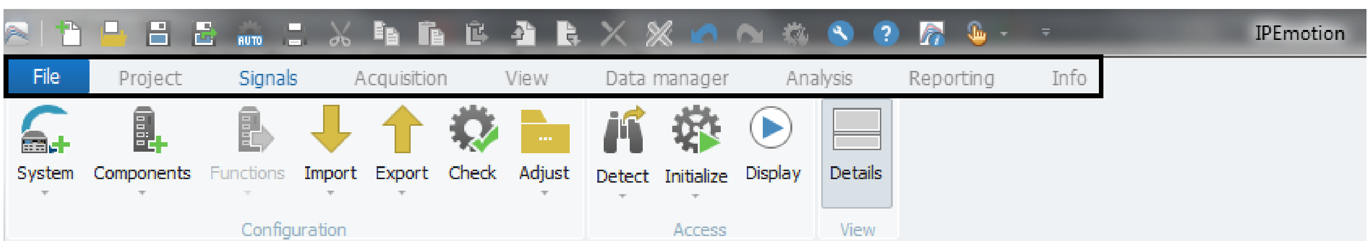
3.2 User interface

This section describes the general appearance and funcionality of the CAETEC Plugin for IPEmotion. However it will not describe specifics on how to create a measurement task or on the single elements available to configure your data logger. For information on these topics please refer to Chapter (→ 5).



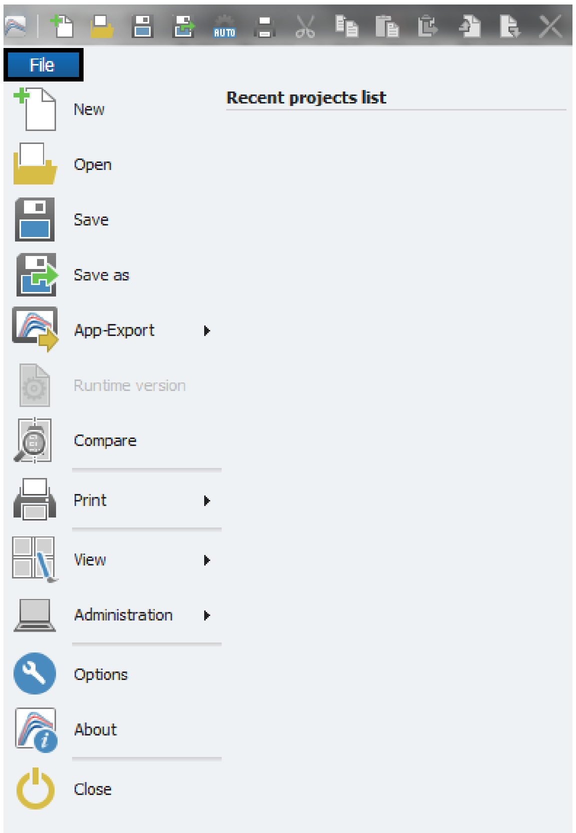
3.2.1 Menu bar

The menu bar provides the core funcionality of IPEmotion. However when working with the CAETEC Plugin you will only need to make use of the "**File**" menu and the "**Signals**" tab, as the Plugin offers a closed working environment which includes all funcionality needed in order to work with your data logger inside the "Signals"-tab.



3.2.2 "File" menu

A dropdown menu with basic IPEmotion funcionality. Only the item relevant for working with the CAETEC Plugin for IPEmotion will be explained here. For information on items not explained here, please refer to the IPEmotion documentation.



New
Creates a new configuration/project.

Open
Opens a previosly saved configuration/project. You can open three different file-types. The **IPEmotion configuration file (*.iwf)** is a container holding the entire IPEmotion-Project and its settings. The **IPEmotion acquisition configuration file (*.iac)** holds only the configuration contained in the "Signals" tab. The **System file (*.isf)** is currently not supported by this plugin.

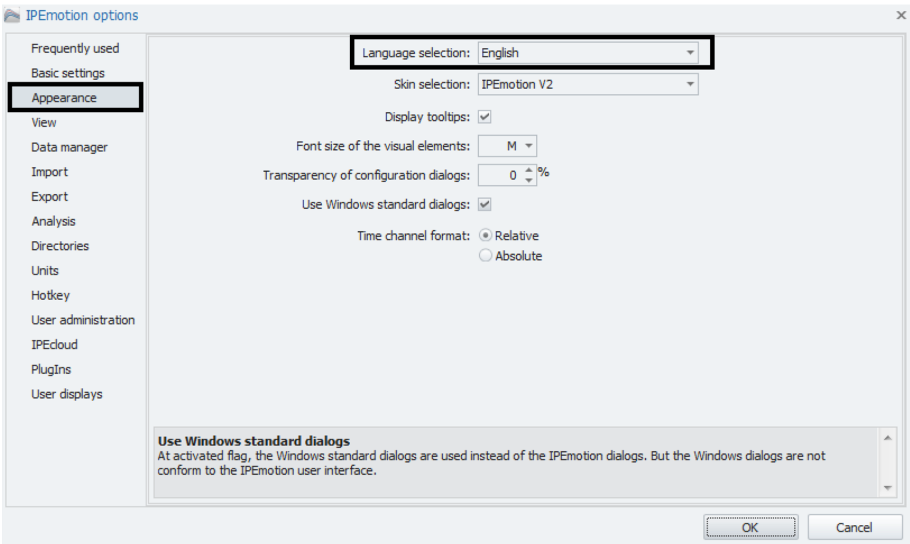
Save
Saves the current configuration/project.

Save as
Save the current configuration/project under a new name and/or in a new location. You can choose between two different file-types to save your projects, which will determine the information your saved file will hold. The **IPEmotion configuration file (*.iwf)** is a container holding the entire IPEmotion-Project and its settings. The **IPEmotion acquisition configuration file (*.iac)** holds only the configuration contained in the "Signals" tab.

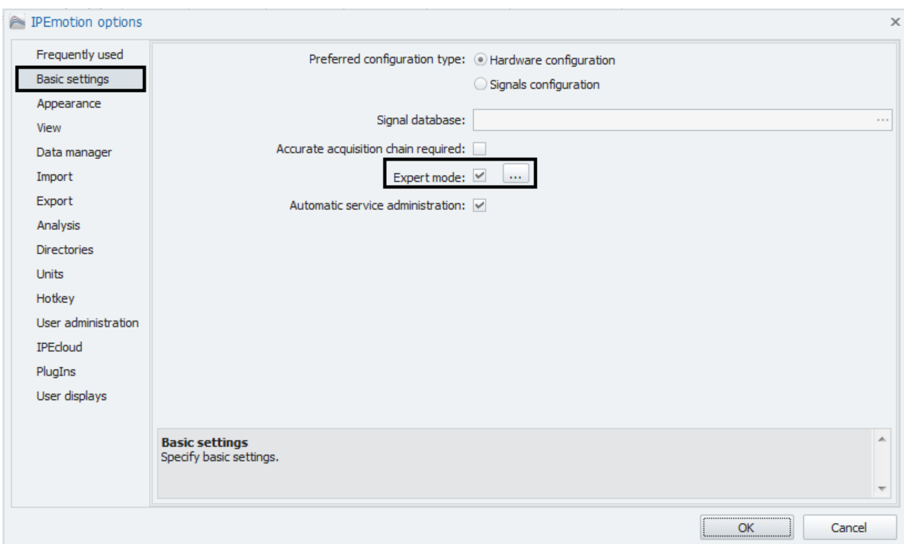
Options

Opens a window with options that affect the behaviour of IPEmotion. This manual will only address the options important for working with the CAETEC Plugin for IPEmotion.

- Change language
  The language can be changed in the options window by choosing the tab "Appearance" on the left and then choosing the desired language from the dropdown menu "Language selection" on the right.

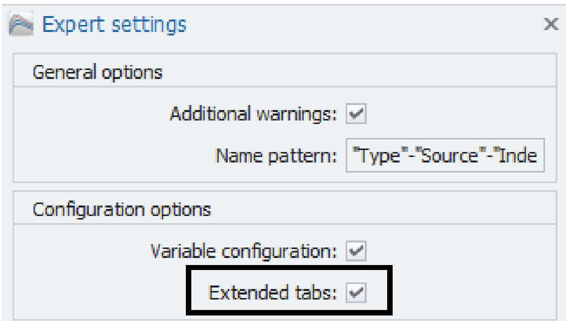


- Expert mode
  The Expert mode can be activated in the options window by choosing the tab "Basic settings" on the left and then checking the checkbox for Expert mode. Once you activated the Expert mode, Expert settings will be accessible by clicking on the button with the three dots next to the Expert mode checkbox.

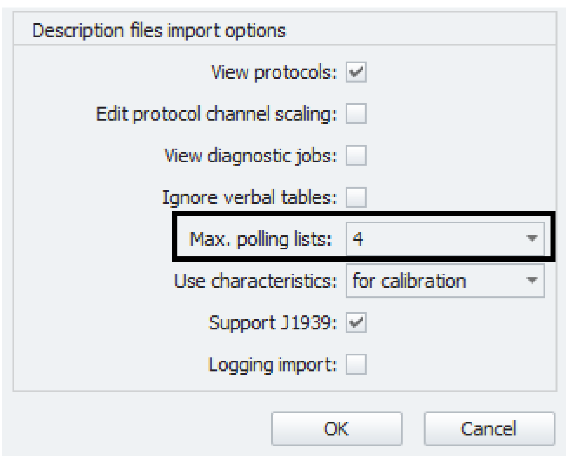


- Extended tabs

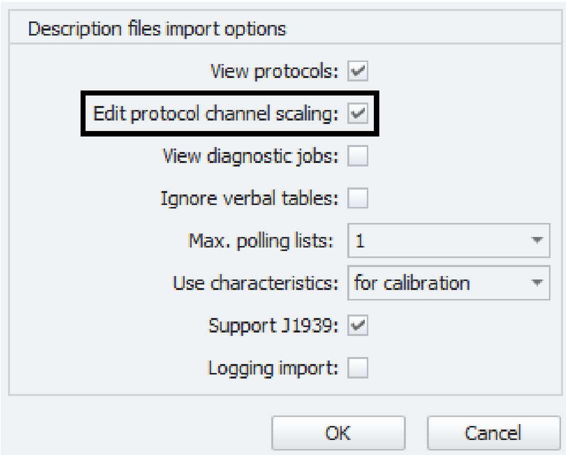
This Option should be activated. It will provide additional tabs in the Details area for some elements of the Measurement task tree. Extended tabs can be activated by checking the checkbox for **Extended tabs** in Expert settings.



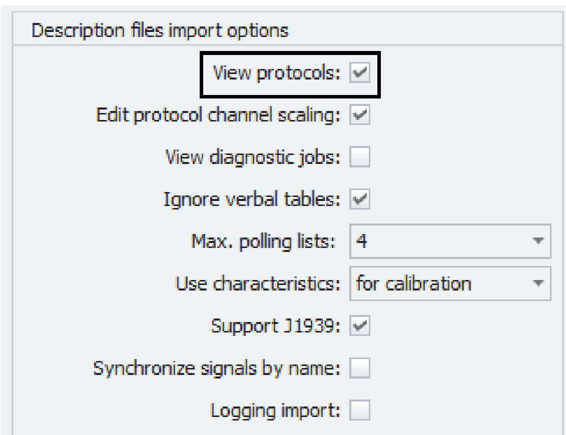
- **Maximum polling lists**
  Defines the maximum number of polling lists. Multiple polling lists must be supported by the connector. The value should be set to a maximum of 4 polling list. The maximum number of polling lists can be changed by choosing the desired number from the **Max. polling lists** dropdown menu in Expert settings.



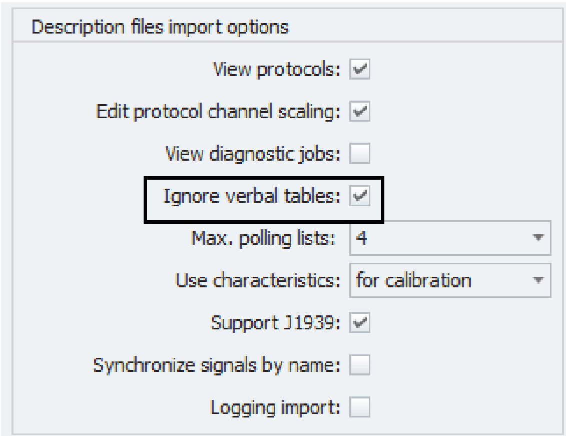
- Edit protocol channel scaling
  This option has to be activated in Expert settings in order for the scaling functionality to be available for protocol channels as well.



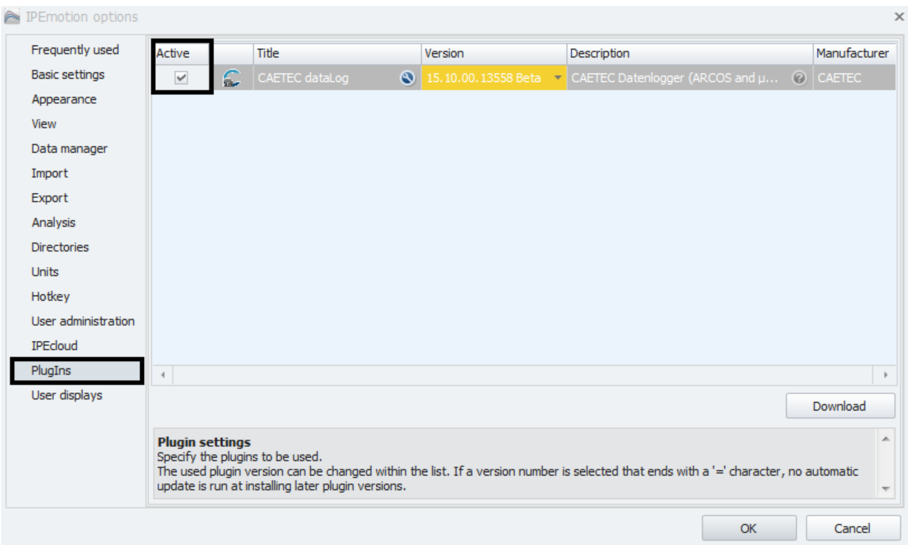
- View protocols
  Activating this expert setting will allow you to see not only the ECU in the "Measurement task tree", but also the associated daq lists and polling lists.



- Ignore verbal tables
  This option has to be activated in Expert settings as the plugin does not support verbal tables.

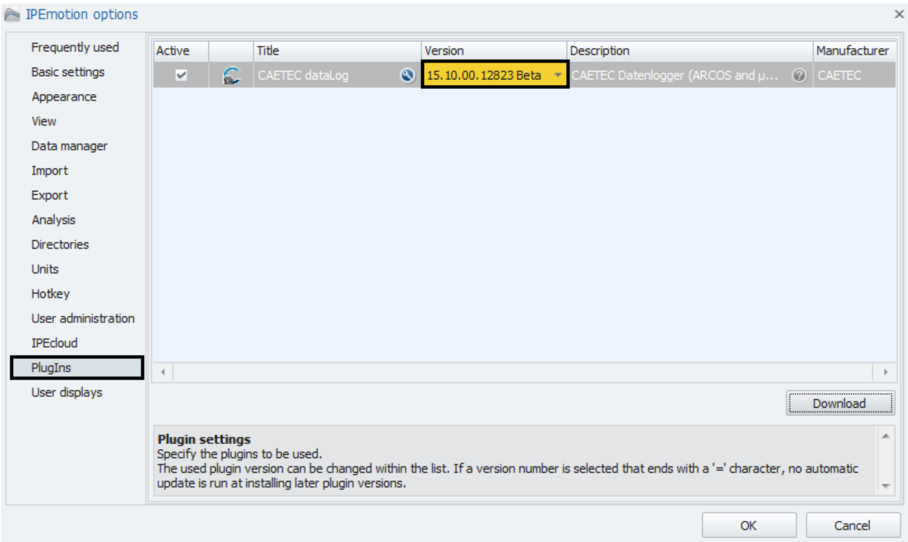


- Activating the Plugin
  In order to work with the CAETEC Plugin you need to activate it inside IPEmotion. To do so, choose **PluIns** on the left side of the Options window and then tick the checkbox saying *Active* for the CAETEC dataLog Plugin.



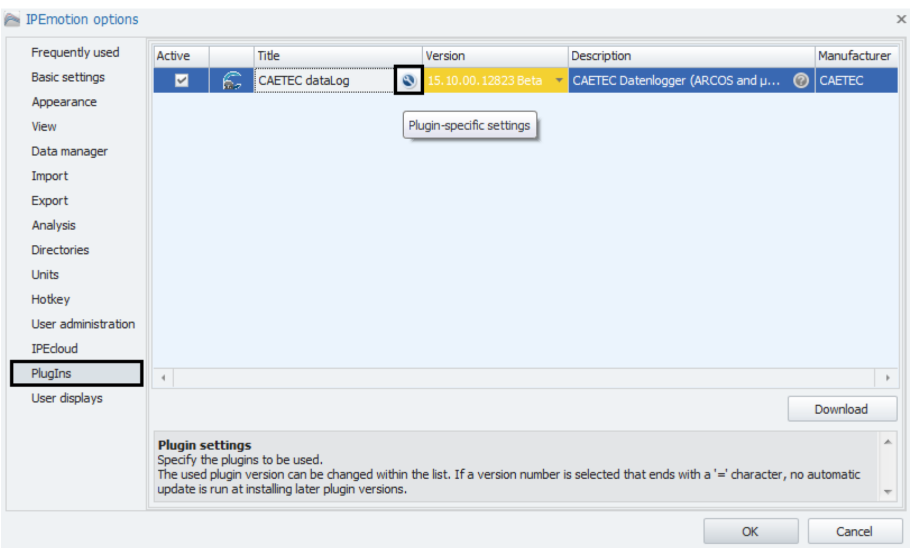
- Switching between Plugin versions

  The CAETEC Plugin allows for various versions of the plugin in to be installed at the same time. It offers you the option to switch between different Plugin versions according to the firmware version your dataLog system is running, and therefore allows the use of various dataLog systems with different firmware versions.
  In order to switch between Plugin versions choose **PluIns** on the left side of the Options window and then you can choose the desired version in the yellow marked dropdown menu on the right.

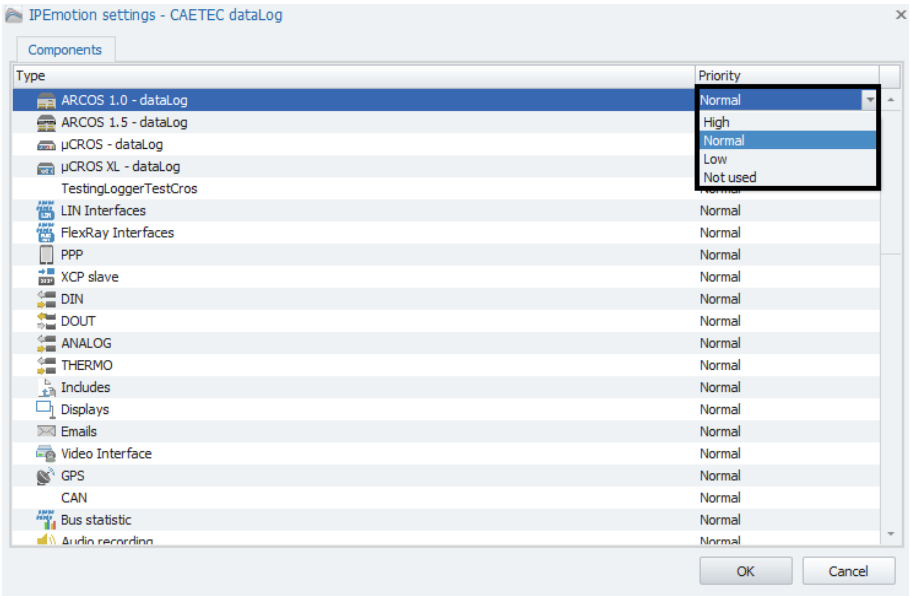


- Customizing tree elements

  The CAETEC Plugin for IPEmotion allows for a customizable user guidance by letting you specifically determine which elements of the Measurement task tree will be available or not. In order to do so, choose **PlugIns** on the left side of the Options window and then press the small button with the blue schrew wrench symbol to open the **Plugin-specific settings**.

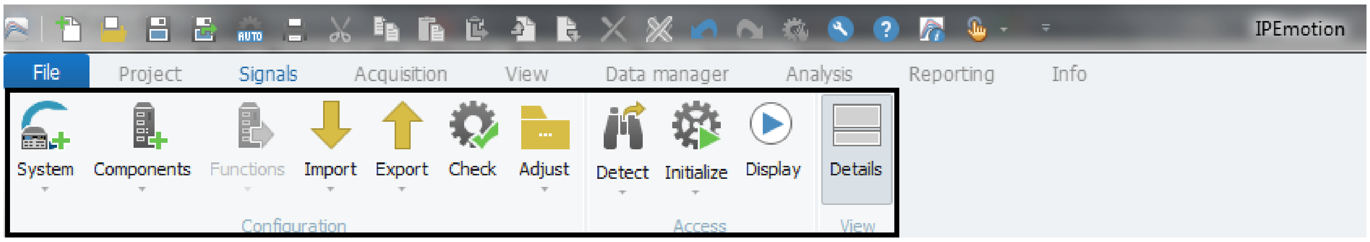


By choosing an element in the following window and setting its priority to “Not used”, as shown below, this element will not be available for the respective Plugin version anymore. If you would like to make available an element which you have previously set to “Not used” then you just have to set its priority back to “Normal”.



3.2.3 Working with the Ribbon

This area contains general functions regarding your configuration.



The "Ribbon's" functionalities "**Components**", "**Functions**", "**Import**" and "**Export**" are also accessible by righclicking on the tree element to which you would like to apply one of these options and then choosing the respective option from the context menu.

System
A dropdown menu that lets you choose which data logger system you want to configure. You can change the logger system into a different system at any given moment by right-clicking on your current logger system in the measurement task tree and selecting **Change into**.

Components
Offers a choice of additional components which are available for the element currently selected in the measurement task tree. You add the desired component by clicking on it. Each activated component will appear in the measurement task tree as a child element to your previously selected element. The same functionality is accessible through right-clicking an element in the measurement task tree and then choosing **Components**.

Functions
Offers a choice of additional functions which are available for the element currently selected in the measurement task tree. You activate the desired function by clicking on it. The same functionality is accessible through right-clicking an element in the measurement task tree and then choosing **Functions**.

Import
Offers a choice of import-options which are available for the element currently selected in the measurement task tree. You choose the desired import-option by clicking on it. The same functionality is accessible through right-clicking an element in the measurement task tree and then choosing **Import**.

Export

Offers a choice of export-options which are available for the element currently selected in the measurement task tree. You choose the desired export-option by clicking on it. The same functionality is accessible through right-clicking an element in the measurement task tree and then choosing Export.

Check

Perform a check on your current configuration's validity. The results will be automatically presented in a pop-up window once the check has finished. In the pop-up window you have the option of rerunning the check by clicking refresh and to export the results as a csv-file or html-file. The check function will be automatically performed each time the datalog.cfg is exported.

Adjust

This functionality is currently not supported by this plugin.

Detect

This functionality is currently not supported by this plugin.

Initialize

This functionality is currently not supported by this plugin.

Display

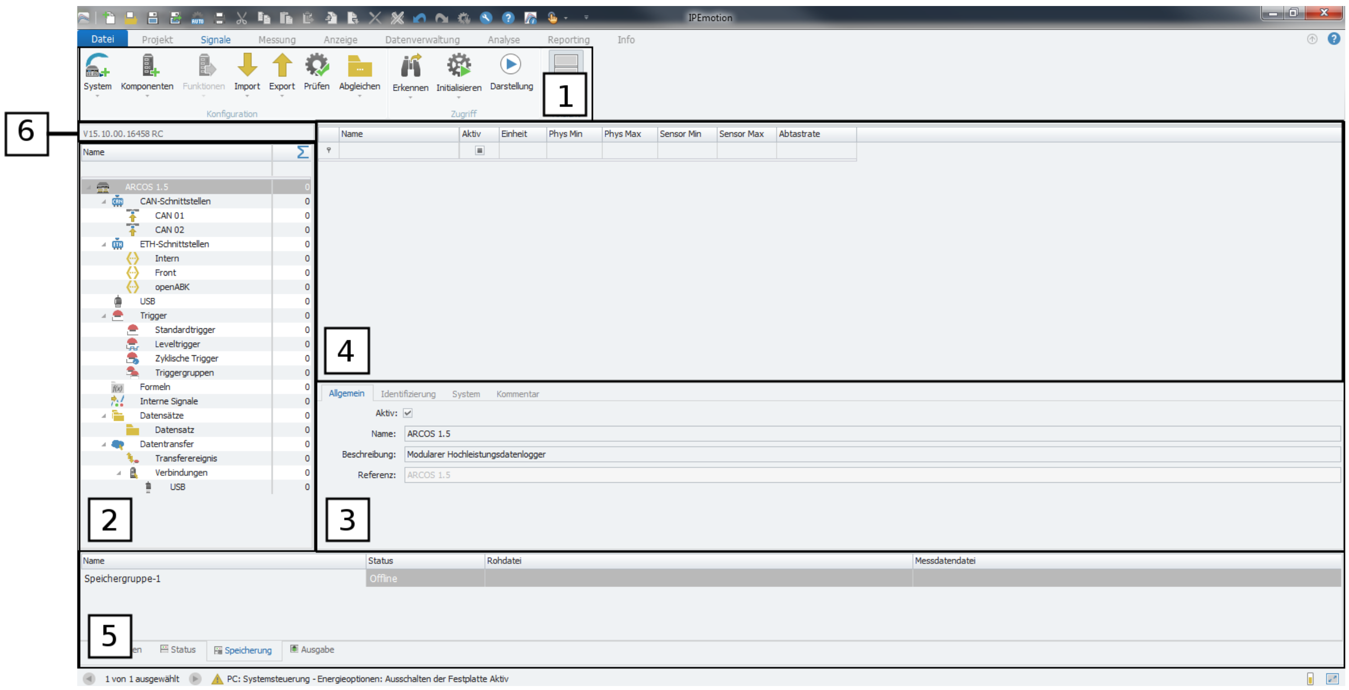
This functionality is currently not supported by this plugin.

Details

Allows you to hide/show the details area of your current configuration.

3.2.4 “Signals” tab

The “**Signals**” tab, which is located in the menu bar (→ 3.2.1), contains your main workspace when working with the CAETEC Plugin for IPEmotion. It is divided in various areas which allow you to create measurement tasks and configure your data logger.



1 Ribbon
A strip of icons that can be clicked for quick access to certain functions and tools. (→ 3.2.3)

2 Measurement task tree
The measurement task tree shows a hierarchical view of the individual configuration sections for the opened measurement task. Specific information on working with the measurement task tree will be given in chapter (→ 4.2.1).

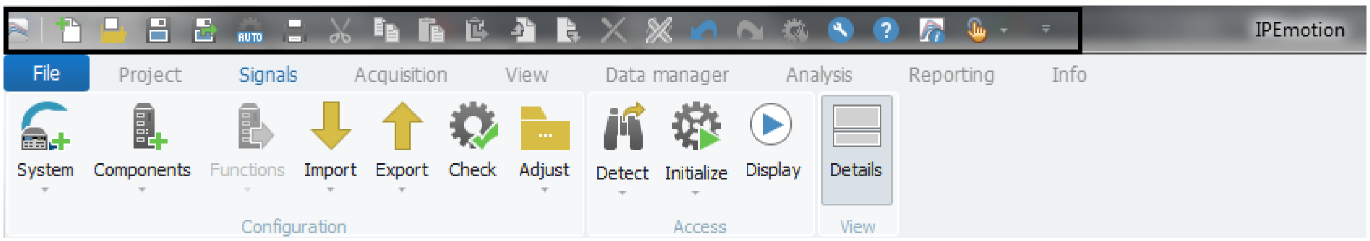
3 Details area
This area contains, for the selected section of the measurement task tree, a field and/or tabs that allows you to set the parameter settings for your selected section. Specific information on working with the Details area will be given in chapter (→ 4.2.2).

4 Grid area
This area contains, for the selected section of the measurement task tree and its child elements, a grid which shows an overview of available measurement channels. Specific information on working with the grid area will be given in chapter (→ 4.2.3).

5 Message area
This area contains messages about errors, warnings and information of the current configuration of the measurement task (→ 3.2.6).

6 Plugin version
The current Plugin version in use is shown in a field located above the measurement task tree. The format of the version is always Vxx.xx.xx and has to match the target system's dataLog firmware-version (→ 4.3).

3.2.5 Quick Access Toolbar



A customizable toolbar which allows you to quickly access your most frequently used tools. For further information please refer to the IPEmotion documentation.

3.2.6 Message area

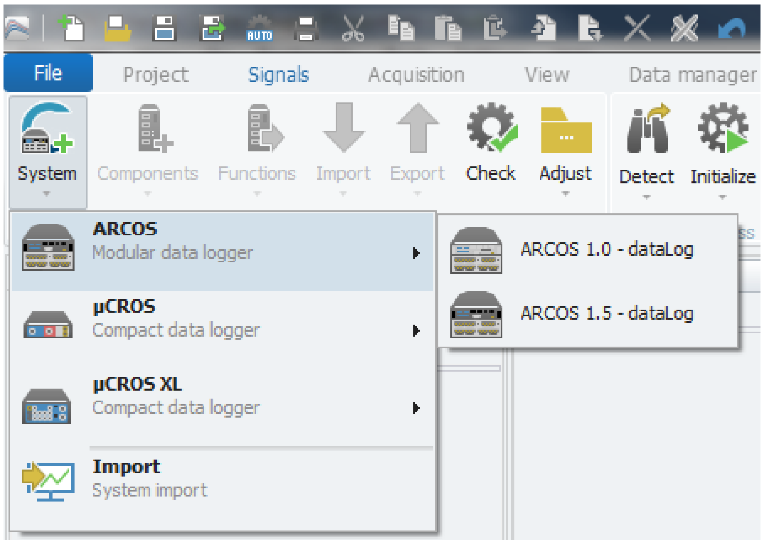
This area contains important messages, the status, potential conflicts and errors of the current configuration of the system.

4 Setting up a logger system

This section explains the workspace and the steps to set up the right logger system for your project/configuration. It will also explain options to customize the use of your logger system.

4.1 Choosing the right logger system

Once you have activated the Plugin (→3.2.2) choose the desired logger-system. Click the **System button** on the top left in the ribbon and then select your desired system.
The system you choose should match your target system (e.g. if your hardware is an ARCOS 1.5 then you should choose the ARCOS 1.5 as the system you wish to configure).



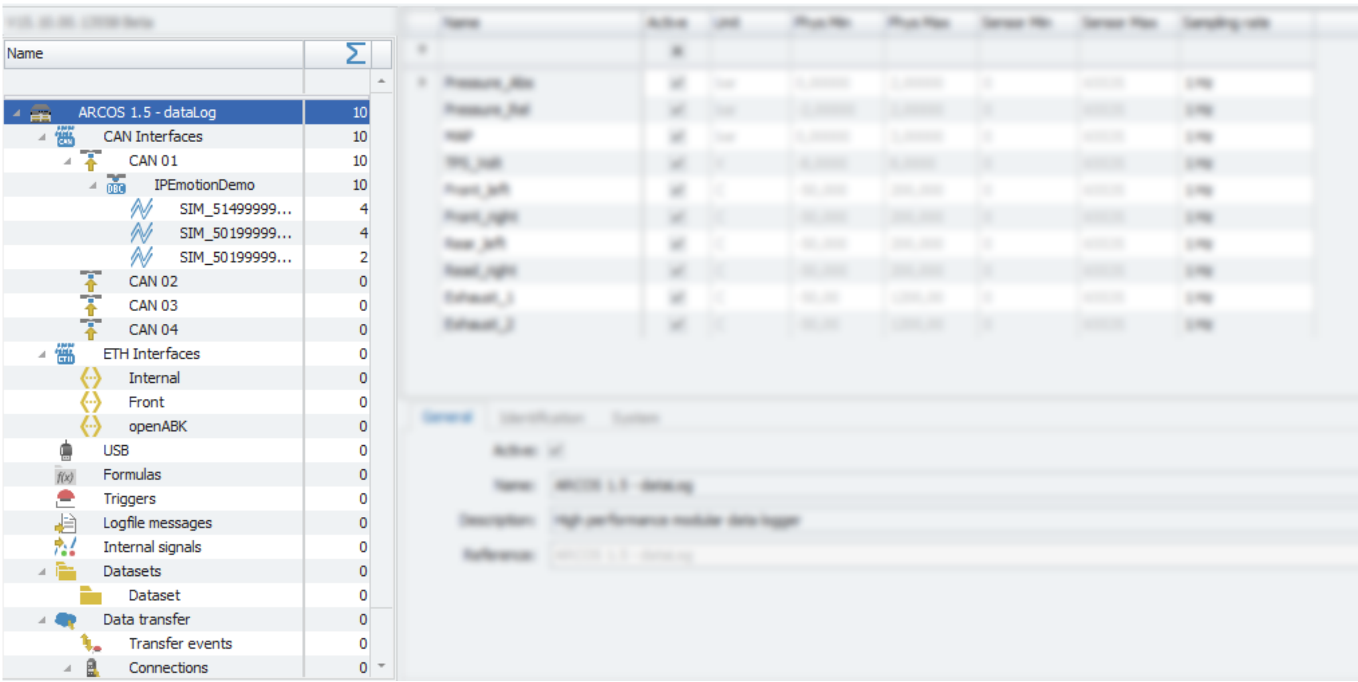
According to your hardware´s possible interfaces a preconfigured workspace will be opened inside the Signals tab. This workspace will be explained in more detail in the following.

4.2 The measurement task workspace

The main parts of the measurement task workspace are the **Measurement task tree** (→ 4.2.1), **the details area** (→ 4.2.2) and **the grid area** (→ 4.2.3). For information on other parts please refer to (→ 3.2.4).

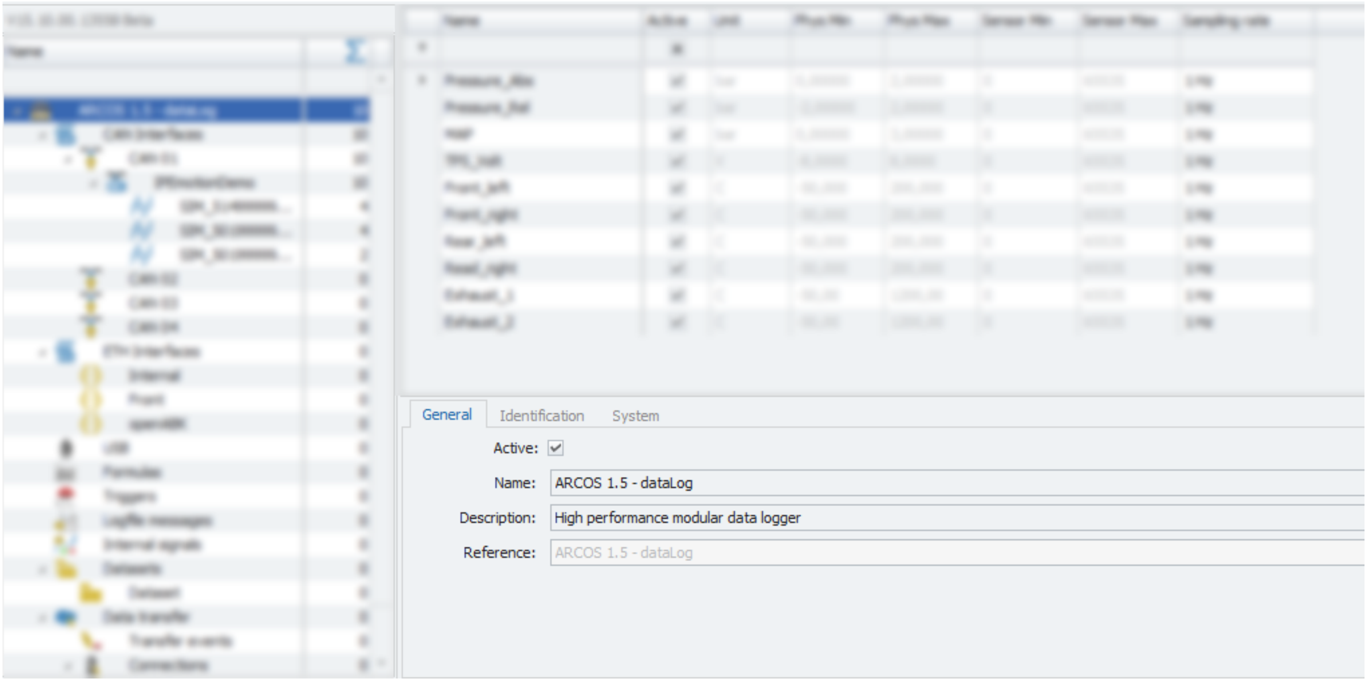
4.2.1 The measurement task tree

The measurement task tree shows a hierarchical view of the individual measurement task-configuration pages of the Plugin. Clicking an item in the tree opens the corresponding configuration pages in the details area and the grid area to the right of the tree, where you can perform the desired settings.
Right-clicking an element in the measurement task trees opens a context menu showing options for that element. Depending on the type of element, this context menu gives you the option of adding additional child elements below the clicked element, for example, or lets you remove or disable optional configuration elements.

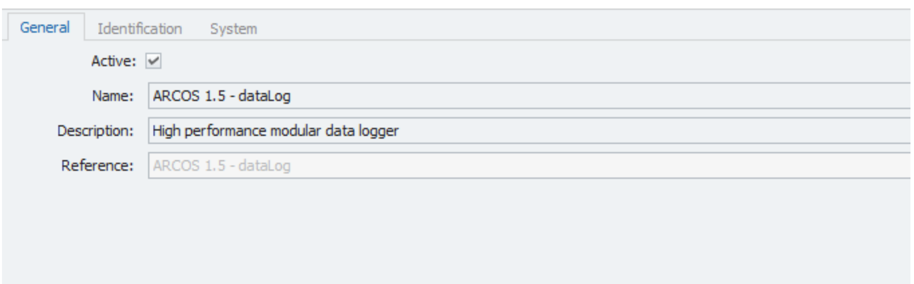


4.2.2 The details area

The details area contains tabs which allow for additional settings for the selected tree element. In this section we will quickly describe the Details settings available for your CAETEC dataLog system, which you will see, once you clicked on your system on top of the measurement task tree. More detailed settings for other tree elements will be handled in the respective sections of this manual.



- General
  This tab allows you to activate or deactivate the entire system by ticking/unticking the checkbox, give a user specific name to your system if wished and add an additional description. The Reference field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed.

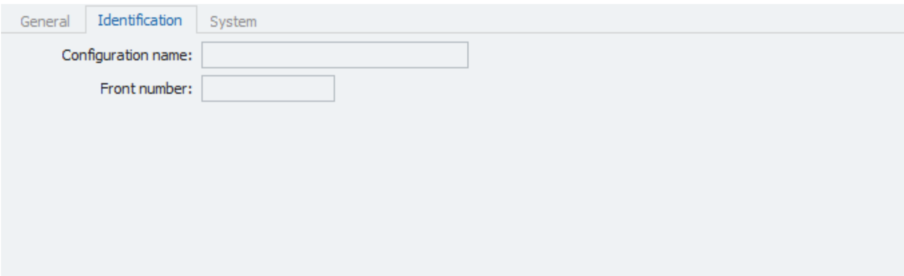


- Identification

  This tab allows for more specific Identification of your system. The **Configuration name** serves as an identifier of your configuration for the user. This is the name of your system that will be visible in the webinterface of your datalogger.

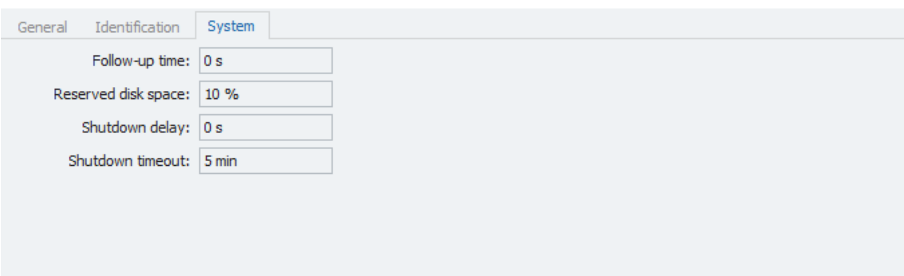
  The **Front number** field allows you to enter a target system's unique front number, which will have the effect, that the created datalog.cfg will only work with that specific target system.

  > ! If a front number is entered, the configuration will exclusively function for the logger with the corresponding front number. It will not work on any other logger.



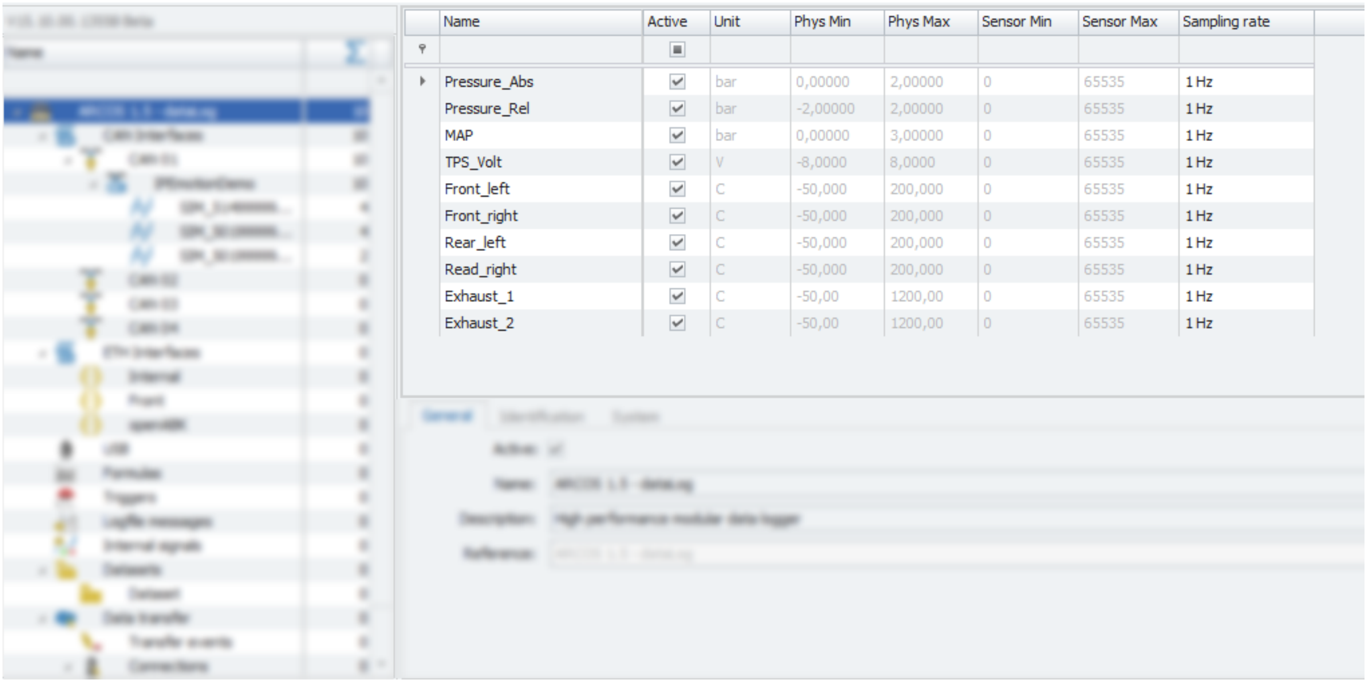
- System

  This tab allows for system specific settings. **Follow-up time** defines how long the logger keeps measuring after the last wake-up condition went away. **Reserved disk space** determines the amount of disk space that is reserved for internal processing (e.g. zip-compression). **Shutdown delay** sets how long the shutdown of the whole system will be delayed after having ended current measuring. During this time a new measurement can be started by a valid wake-up condition. **Shutdown timeout** determines the maximum allowed time for a shutdown of the system. If this time is exceeded, a hard shutdown will be forced and all processes, including datatransfers, will be cancelled.



4.2.3 The grid area

The grid area shows all available signals of the selected tree element. It also allows to activate certain signals, to rename them and sort by the different column's parameters. The "column chooser" allows you to customize which columns will be shown in the grid area. For more information on the "column chooser" please refer to (→4.3.1).



4.3 Plugin version

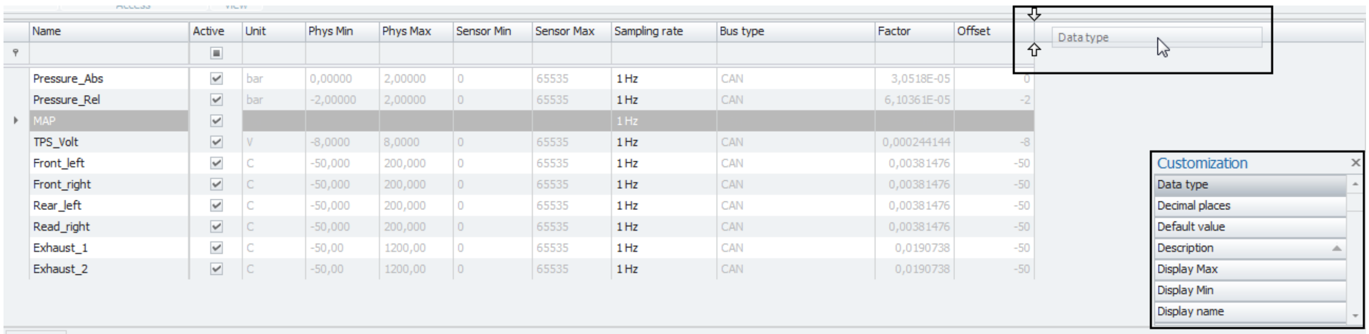
The current Plugin version in use is shown in a field located above the measurement task tree. The format of the plugin version is always Vxx.xx.xx (Vmajor.minor.hotfix). The plugin version has to match the target system's dataLog firmware-version.

> ! If you have multiple Plugin versions installed, you may switch between versions. For detailed information please refer to **Switching between Plugin versions** (→ 3.2.2).

4.3.1 Column chooser

The "column chooser" allows you to customize which columns will be visible in the grid area. You can customize your "grid area" in such a way, that every setting you can adjust in the "details area" for a given tree element can be made accessible through the "grid area" of this tree element.

In order to access the "Column chooser" rightclick on any point of the column title bar and choose "Column chooser" from the context menu.

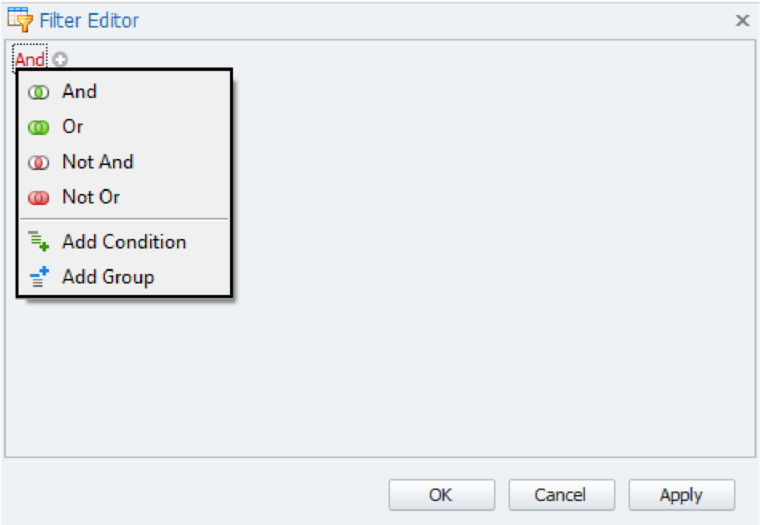


The resulting window on the right called "Customization" will present you with a selection of the available columns. In order to add an extra column, choose the one you would like to add from the "Customaization" window and drag id to the column bar as shown above.

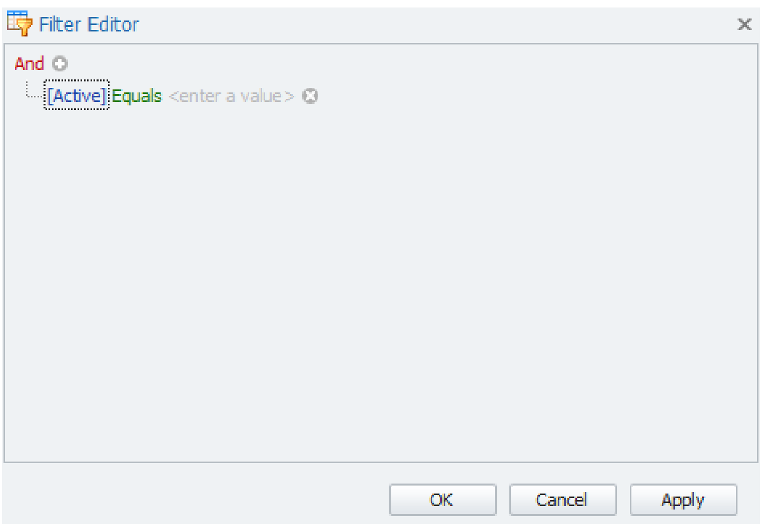
If you wish to remove a column just drag it to any point outside of the column title bar until you see a big black cross and then release it.

4.3.2 Filter editor

The “Filter editor” allows you to apply customized filter rules to your signals.
In order to access the “Filter editor” rightclick on any point of the column title bar and choose “Filter editor” from the context menu.

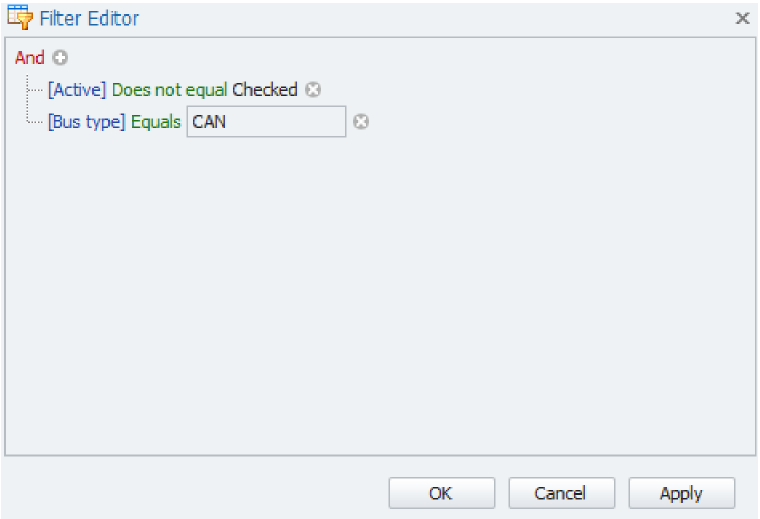


Once the “Filter editor” has opened, click on the red “And” in the top left corner. The resulting context menu allows you to choose the method of combining the different filter conditions (And, Or, Not And, Not). It also allows you to add a condition (which can alternatively be achieved by clicking the “+” sign right of the “And”) or add a new group of filter conditions.



Once you have added a filter condition, you will now need to specify this condition in order to funciton properly. There are three fields you will need to set.

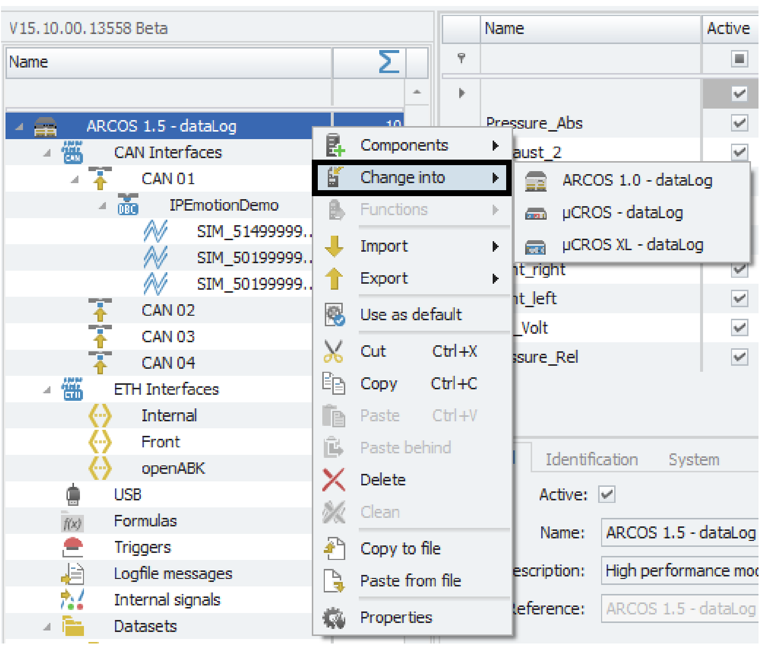
- The field on the left with **blue letters** defines to which column the filter will apply.
- The field in the middle with **green letters** sets a parameter for your filter condition.
- The field on the right with **grey letters** sets a value for the choosen parameter.



You can combine different filter conditions or groups of filter conditions to obtain the desired filter result. In the example shown above, the filter will select all CAN signals which are not checked "Active".

4.4 Changing the logger system with the *change into*-command

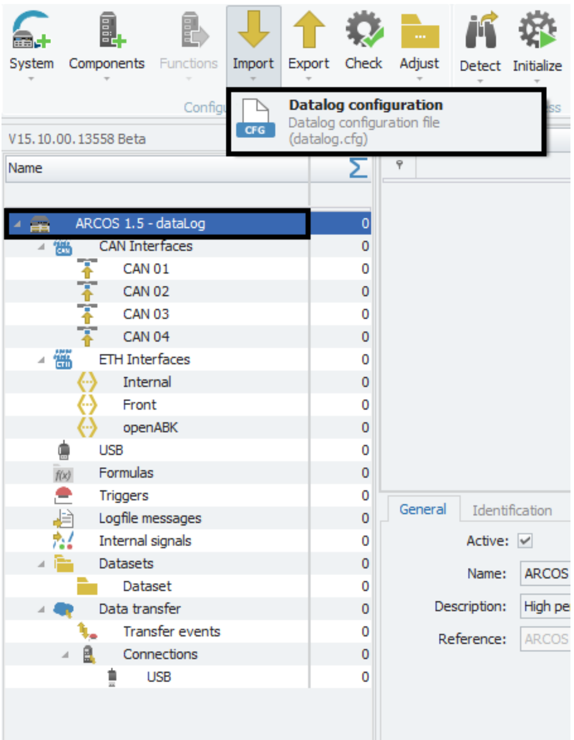
A previously set up logger system can still be changed into another one later. To do so, right-click on your system in the measurement task tree, navigate to **change into** and choose the system you would like to change it to.

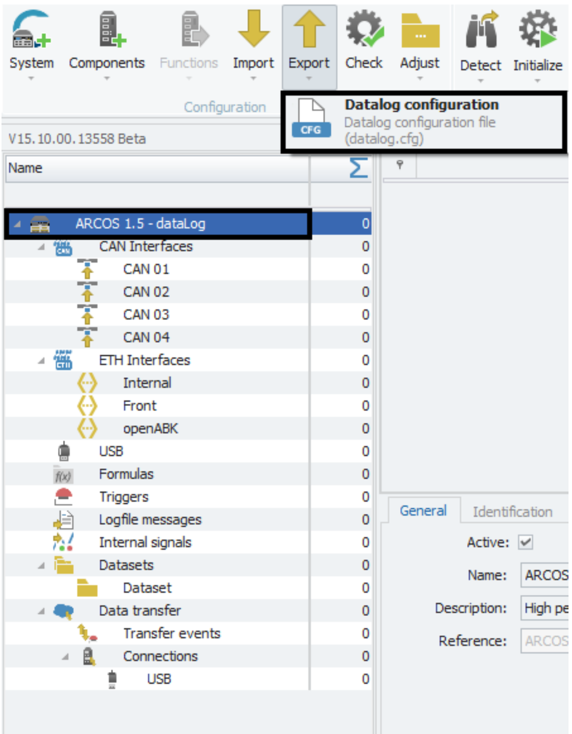


If you are trying to change into another logger system and one of your currently used tree elements is not available in the system you are changing into, you will see a **Warning-message** in the Messages area (→3.2.6) and this tree element will not be migrated.

4.5 Importing and exporting the datalog.ccmc

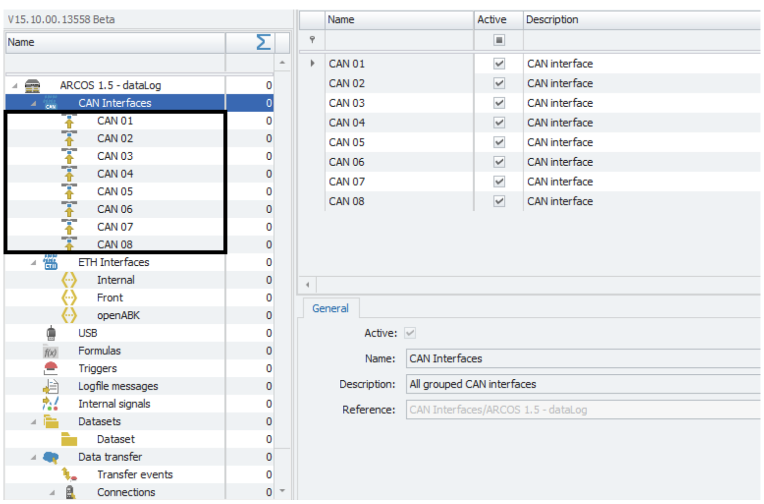
Once you have chosen the system you would like to configure, you will be able to import and/or export the datalog.ccmc. That way you can import a datalog.ccmc previously configured with the plugin and can keep the settings. The export function allows you to export your configuration and then copy it to one or more matching target systems. To do so, just select your system in the measurement task tree and then select either **Import** or **Export** from the Ribbon and choose datalog.ccmc. Alternatively you can right-click on your system in the measurement task tree and select Import or Export from the following context menu.



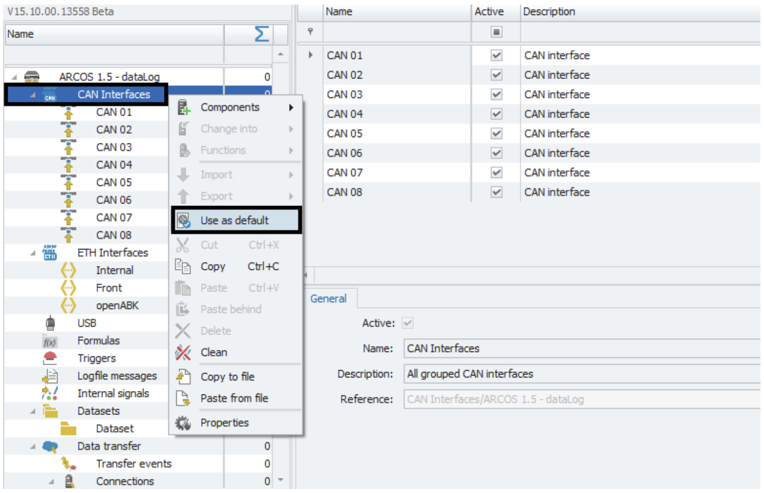


4.6 Changing a system´s default tree elements

The tree elements available by default for a given system are merely a preference, which can be adjusted via the measurement task tree. If you see, that you usually use 8 instead of 4 CAN busses, you can determine this setting as your default for the currently used system type and the next time you set up a new logger system it will automatically start with your new default settings. To do so, you will first need to add or remove the desired components. (In this case we added another 4 CAN busses in order to get a total of 8).



In the next step you will have to right-click on the category of tree elements for which you wish to define your new default settings as shown below. Note: it will not work, if you right-click the single component or the system itself. Make sure to right-click the category, in this case the **CAN Interfaces** and then choose **Use as default**.

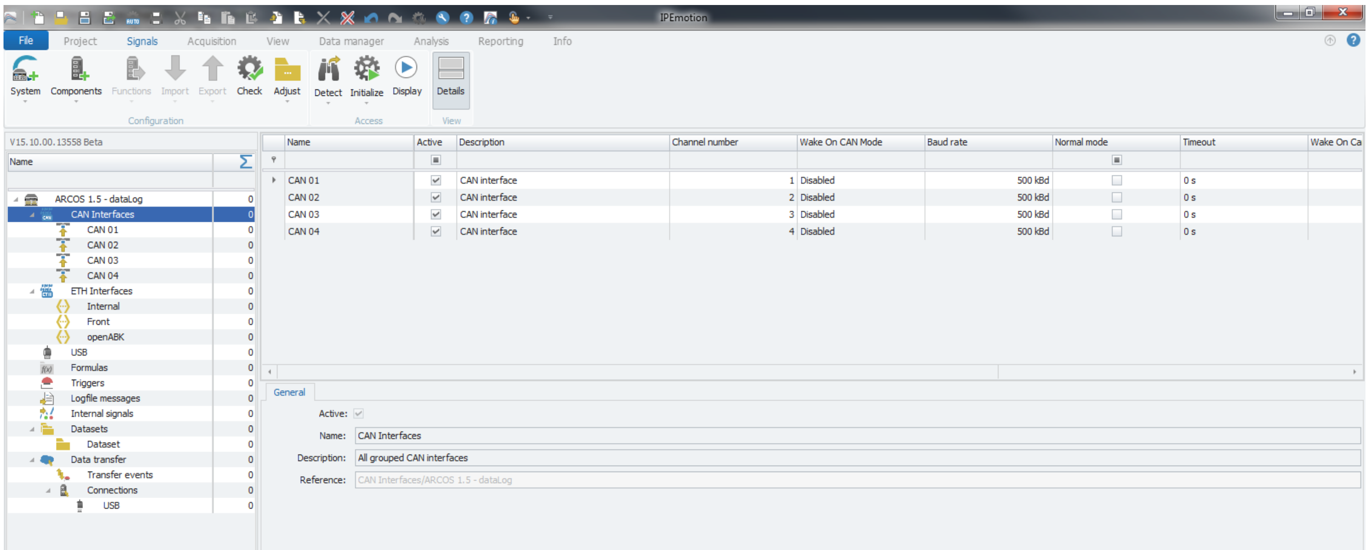


5 Signal Acquisition

5.1 CAN channels

All the CAN channels for your system are located in the tree element "CAN interfaces". There are two types of CAN channels, physical "**CAN**" channels and "**Virtual CAN**" channels.
According to the default settings, the tree element "CAN interfaces" will include a preset number of CAN channels. By clicking the tree element CAN Interfaces you will see all of its channels and signals in the grid area as well as a tab called **General** in the **Details area** which allows you to set a name and description. These settings apply to the entire element "CAN interfaces".

In the following will be described how to add CAN channels and adjust their settings (→5.1.3).

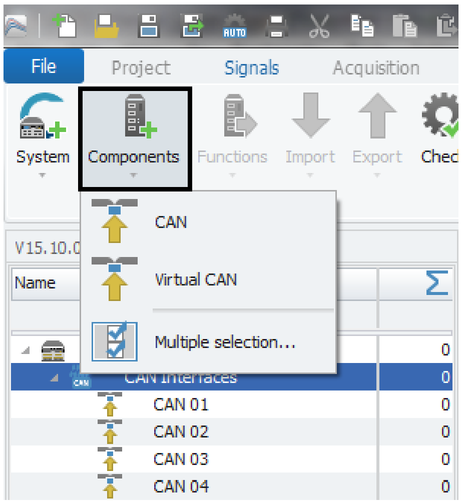


5.1.1 Storage method

In order to store all incoming traffic on a CAN channel use a bus tracing method for storage. Please refer to (→ 7.7).

5.1.2 Adding CAN channels

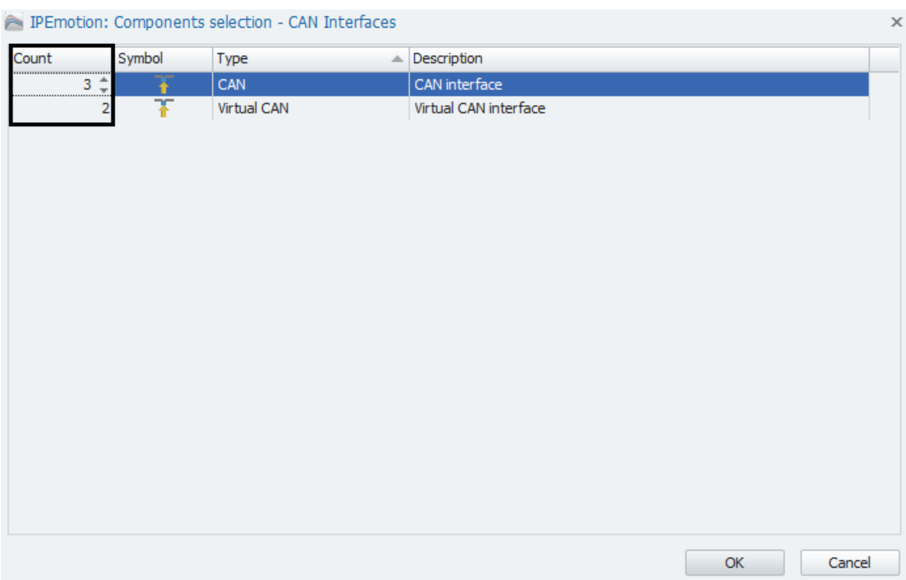
CAN channels can be added by selecting the tree element "**CAN interfaces**", then clicking the "**Components**" button and finally choosing the desired type of CAN channel you wish to add.



- CAN
  Adds a CAN channel that corresponds to a physical CAN channel of your logger. For instructions on CAN settings refer to (→5.1.3).

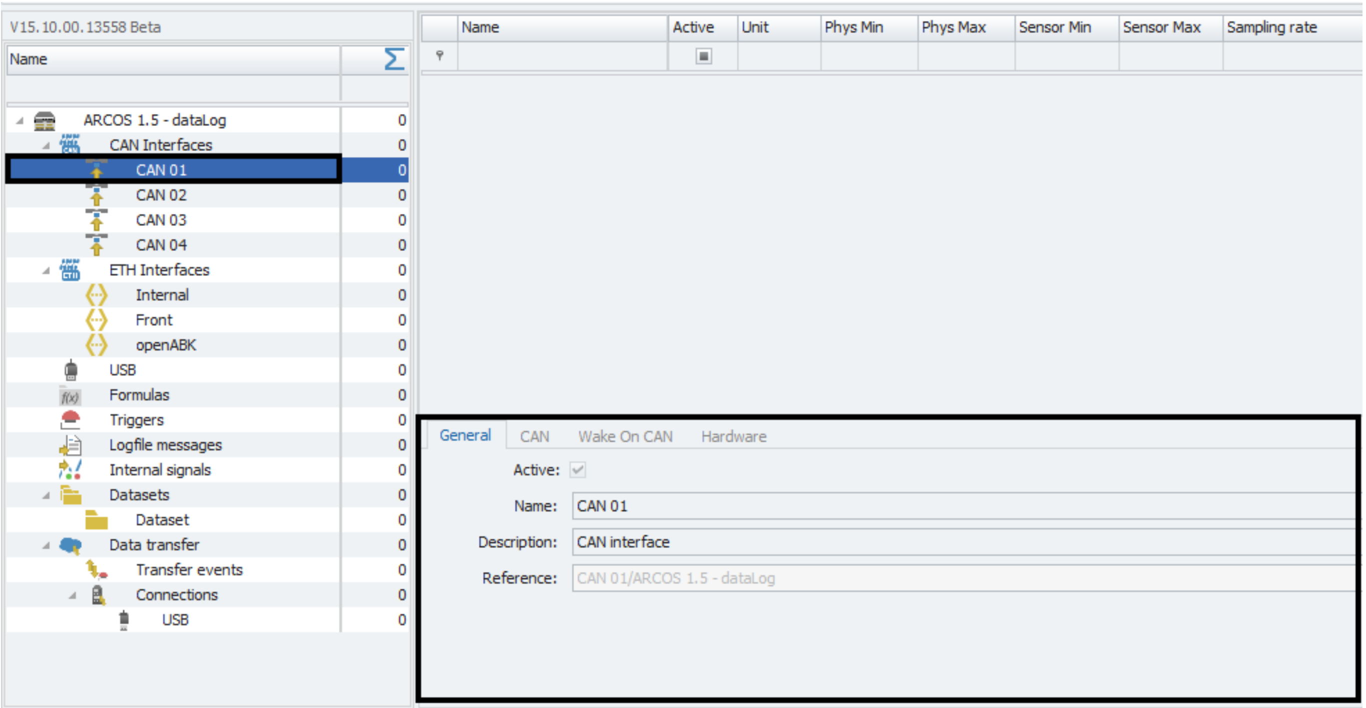
- Virtual CAN
  Adds a virtual CAN channel. For instructions on Virutal CAN settings refer to (→5.1.4).

Multiple selection Allows you to add multiple CAN channels of both types at the same time. To do so set the counter for each type to the desired number of channels that you wish to add as marked in the figure below.



5.1.3 CAN settings

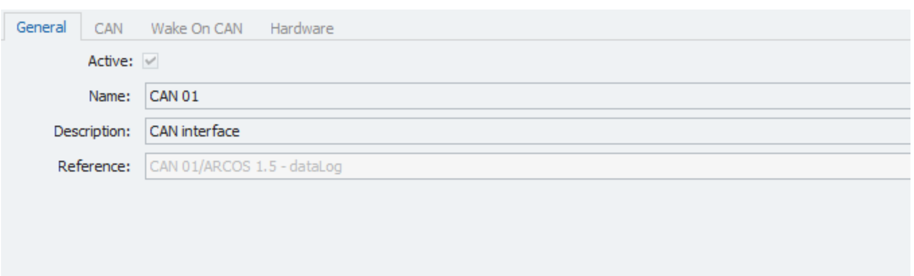
By selecting one of the CAN channels in the tree you will be able to define this channel's settings in the details area.



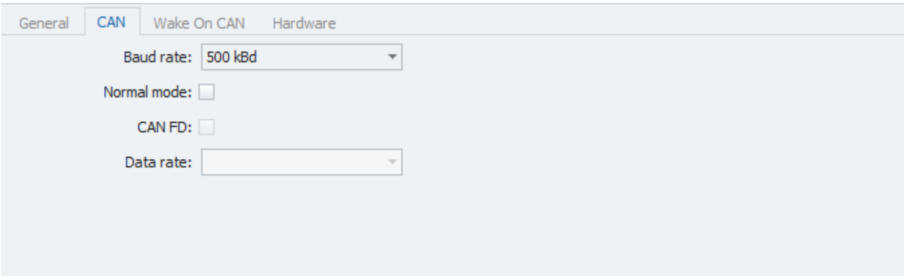
The same settings described in this section as part of the Details area can also be adjusted when selecting the tree element CAN interfaces and then directly changing the desired setting in the respective field of the Grid area.

5.1.3.1 General

This tab allows you to give a user specific name for the selected CAN channel if wished and add an additional description. The Reference field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed. The **Active** checkbox is by default marked as activated and cannot be changed.



5.1.3.2 CAN



Baud rate

The dropdown menu Baud rate allows you to set the Baud rate for the selected CAN channel. The baud rate defines the speed in bits/second at which data can be transmitted through the CAN bus. The speed muss be adapted to the source. Only if all the users on the bus are set at the same rate, is transfer possible.

Normal mode

The checkbox Normal mode allows you to determine whether the CAN channel may communicate or is in silent mode. If the checkbox is marked, the channel may communicate, if the checkbox is not marked, the channel is in silent mode.

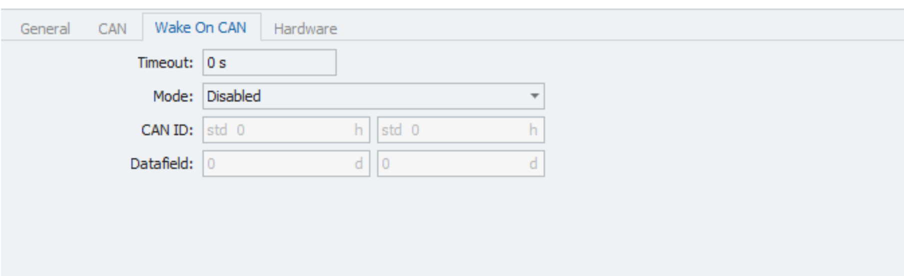
CAN FD

This option is not supported by this plugin

Data rate

This option is not supported by this plugin

5.1.3.3 Wake On CAN



Timeout

For Wake on CAN, timeout has a special significance. It defines how long a waking channel must be inactive to be recognized so and therefore allow for the logger to shut down. If timeout is recognized, an entry is made in the log file and an error message with an alert appears on the display, which has to be acknowledged.

Mode

This dropdwon menu allows you to set the wake-up function for your selected CAN channel.

Wake on CAN type	Characteristics
Disabled	No start on CAN messages, lowest energy consumption.
Enabled	Start on a CAN message, with first messages lost; low energy consumption.
Enabled (no message lost)	Start on CAN message, with no message lost; slightly higher idle current.
Keep awake	The logger starts with Clamp 15, but only shuts down if all the awakening conditions (Clamp 15 and WakeOnX) are no longer fullfilled.

CAN ID - Settings for starting on a specific CAN ID

The CAN ID consists of two fields in the "Wake on CAN" tab of the CAN channel's details area: The **"CAN identifier"** on the left and the **"CAN ID bitmask"** on the right.

The aim of this setting is to wake the logger with a message having a specific ID (or group of IDs), regardless of what the content of the message is. In order to do so, you can define a **"CAN identifier"** and a **"CAN ID bitmask"** to limit the identifier. Both parameters are used in their binary form. The "CAN ID bitmask" defines (or masks) the bit positions of a message ID that are to be applied. The "CAN identifier" specifies the contents that must exist at these bit positions in order for a start to take place.

Both fields can be defined as **"standard CAN ID"** or **"extended CAN ID"** by clicking the button in the left corner of the field. Both fields can process and show the entered number in its decimal or hexadecimal form. To switch between decimal or hexadecimal just click the button in the right corner of the field as marked in the figure above.

> The decimal number is processed in the logger in its binary form and if the number set for the "CAN ID bitmask" = 0, the logger will start on any message.

Example:
CAN identifier = 22 (binary = 10110)
CAN ID bitmask = 28 (binary = 11100)

In other words, the mask (CAN ID bitmask) specifies that, to start the logger, given values are expected at the positions Bit2, Bit3, Bit4. All other values are irrelevant ("x").
The CAN identifier calls for Bit2 to be "1", Bit3 to be "0" and Bit4 to be "1".

The following table lists the positions for an 11-bit CAN identifier (in red the masked positions, the numbers signifying the expected values).

Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
x	x	x	x	x	x	1	0	1	x	x

Example of values of a CAN ID that would start the logger in this configuration:

Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	CAN-ID (dec)
		1	0	0	0	1	0	1	0	1	277
				1	0	1	0	1	1	1	87
1	0	0	0	0	0	1	0	1	0	0	1044
			1	1	1	1	0	1	1	1	247
			1	0	0	1	0	1	1	0	150
						1	0	1	0	0	20
						1	0	1	0	1	21

*Values of a CAN ID that would **NOT** start the logger in this configuration (example):*

Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	CAN-ID (dec)
		1	0	0	0	1	1	0	0	1	281

Datafield - Settings for starting on a specific message value of a CAN ID

The Datafield consists of two fields in the "Wake on CAN" tab of the CAN channel's details area: The **"Datafield content"** field, which will contain the **"Datafield content value"** on the left and the **"Datafield bitmask"** on the right.

The aim of this setting is to wake the logger with a given content of a message with a particular ID (or a group of IDs).

After defining the ID that is supposed to start the logger, you can follow the above pattern to additionally specify which value within the message of the ID is to be an additional requirement for start. The entire 64-bit message is considered, single bits of which can be defined as start conditions. (In order to specify the values, it is necessary to decode the binary structure of the message. There is no DBC file assistance available here.)

A **"Datafield bitmask"** defines (masks) which bit positions of a message are to be applied. A "1" marks the bit positions to be used. Positions coded "0" are ignored. The "Datafield content value" specifies the contents checked in the mask by the logger. If "Datafield content value" writes "1" (or "0") in the positions of the mask (Datafield bitmask), then the mask positions of the ID must contain the identical values; i.e. "1" (or "0") too, otherwise the logger is not started.

Both fields can process and show the entered number in its binary, decimal or hexadecimal form. To switch between binary, decimal or hexadecimal just click the button in the right corner of the field as marked in the figure above.

The decimal number is processed in the logger in binary form and if the number set for the "Datafield bitmask" = 0, the logger will start on any message. When the number entered in the "Datafield bitmask" is considered in binary form,
"1" defines a bit position that is considered in filtering,
"0" means filtering ignored this bit position.

Example:
Wake on CAN trigger raw value = 22 (binary = 10110)
Wake on CAN raw value mask = 20 (binary = 10100)

The mask "Datafield bitmask" specifies that, the logger is only started if certain values are found at the positions Bit2 and Bit4. All other values are irrelevant ("x").
"Datafield content value" calls for Bit2 to be "0" and Bit4 to be "1".
The following table lists the positions for a 64-bit message (with the masked positions colored red and the numbers signifying the expected values).

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
x	x	x	1	x	0	x	x

Message values that would start the logger in this configuration (example):

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Message value (dec)
			1	0	0	0	1	17
	1	0	1	1	0	1	1	27
			1	0	0	0	0	16
1	1	1	1	1	0	1	1	251
1	0	0	1	0	0	1	0	146
			1	1	0	0	0	24

Negative example of message values that would NOT start the logger in this configuration:

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Message value (dec)
0	0	0	1	1	1	0	0	28

Settings for starting on a specific signal

The settings described in the previous paragraphs apply to entire messages. This section explains how to apply these settings to a specific signal contained in a message. It describes the procedure for deriving the required WoC parameters from the physical value of a signal. Since a simple formula such as "signal > 30" is not possible, it is necessary to define the start condition at the bit level. The numeric format is important, as well. The order in which the bit positions are counted depends on the numeric format (e.g., Intel or Motorola). Please bear in mind, when selecting the signal and a particular start value, that the start condition must be met during the entire measurement run. Remember: The start signal is also the stop signal. If the start signal is missing for a set period of time, the logger is stopped. This makes state bits good start signals. Signals such as temperature signals that generally fluctuate, are only suitable providing the definition of the start condition is sufficiently vague. Here vague means that not a specific bit combination switches on the logger, but that a range of bit combinations is possible. When defining filters, be sure to avoid gaps between the significant mask positions (marked), otherwise the covered range of values will also have gaps (see Filter 4), which would shut down the logger. Several filters serve as examples below. The table shows which values start the logger with which filter. In the column for each filter, these values are marked with an "X" .

Filter 1:
Datafield content value = 16 (binary = 10000)
Datafield bitmask = 24 (binary = 11000)

Bit7	*Bit6*	*Bit5*	*Bit4*	*Bit3*	*Bit2*	*Bit1*	*Bit0*
x	*x*	*x*	*1*	*0*	*x*	*x*	*x*

Filter 2:
Datafield content value = 24 (binary = 11000)
Datafield bitmask = 24 (binary = 11000)

Bit7	*Bit6*	*Bit5*	*Bit4*	*Bit3*	*Bit2*	*Bit1*	*Bit0*
x	*x*	*x*	*1*	*1*	*x*	*x*	*x*

Filter 3:
Datafield content value = 16 (binary = 10000)
Datafield bitmask = 16 (binary = 10000)

Bit7	*Bit6*	*Bit5*	*Bit4*	*Bit3*	*Bit2*	*Bit1*	*Bit0*
x	*x*	*x*	*1*	*x*	*x*	*x*	*x*

Filter 4: (negative example)
Datafield content value = 18 (binary = 10010)
Datafield bitmask = 18 (binary = 10010)

Bit7	*Bit6*	*Bit5*	*Bit4*	*Bit3*	*Bit2*	*Bit1*	*Bit0*
x	*x*	*x*	*1*	*x*	*x*	*1*	*x*

Dec. value	Binary value	Filter 1	Filter 2	Filter 3	Filter 4
15	0000001111				
16	0000010000	x		x	
17	0000010001	x		x	
18	0000010010	x		x	x
19	0000010011	x		x	x
20	0000010100	x		x	
21	0000010101	x		x	
22	0000010110	x		x	x
23	0000010111	x		x	x
24	0000011000		x	x	
25	0000011001		x	x	
26	0000011010		x	x	x
27	0000011011		x	x	x
28	0000011100		x	x	
29	0000011101		x	x	
30	0000011110		x	x	x
31	0000011111		x	x	x
32	0000100000				

The following example of a temperature in Intel format illustrates how to derive the filter settings.

Example
With the logger operating in the temperature range of approx. 30 - 40 °C, the following signal definition is given:
Deriving Wake on CAN "Datafield bitmask" and Wake on CAN "Datafield content value":

First you use the limit values to derive the raw value and thus the binary value of the operating range.

$$Datafieldcontentvalue_{30degree} \frac{value - Offset}{Scale} = \frac{30-(-10)}{0.1} = 400 = Binary : 110010000$$

$$Datafieldcontentvalue_{40degree} \frac{value - Offset}{Scale} = \frac{40-(-10)}{0.1} = 500 = Binary : 111110100$$

The highest bit positions that are identical for both values, without a gap, are the two on the left, which have the value 1. This yields, as mask and filter binary: 1 1000 0000

So the lower bound for the Datafield content value is:

- *binary 1 1000 0000*
- *decimal 384*
- *physical 28,4°C*

So the upper bound for the Datafield content value is:

- *binary 1 1111 1111*
- *decimal 511*
- *physical 41,1°C*

Since the value has 9, but the signal 16 Bit, the 7 most significant Bits have to be 0. This is achieved by the Datafield bitmask.

Filter:
Wake on CAN Datafield content value = 384 (binary = 0000 0001 1000 0000)
Wake on CAN Datafield bitmask = 65408 (binary = 1111 1111 1000 0000)

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	0	1	1	x	x	x	x	x	x	x

Since, however, the signal lies in the middle of the 64 bit-wide message (bit offset=16), the bit positions to the right of the signal in the message must be filled with "0" (left is equal to Bit0 in the CAN-traffic).

	Bit-Offset				T_Aussen				Don´t care							
Binary	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
Intel:	LSB							MSB								
Datafield content value:	0000	0000	0000	0000	0000	1000	0001	0000	0000	0000	0000	0000	0000	0000	0000	0000
Hex:	0	0	0	0	0	8	1	0	0	0	0	0	0	0	0	0
Datafield bitmask:	0000	0000	0000	0000	0000	1000	1111	1111	0000	0000	0000	0000	0000	0000	0000	0000
Hex:	0	0	0	0	0	8	F	F	0	0	0	0	0	0	0	0

So the Datafield content value is equal to:
Datafield content value (hex) = 0xFF800000
Datafield content value (dec) = 4,286,578,688

So the mask is equal to:
Datafield bitmask (hex) = 0x1800000
Datafield bitmask (dec) = 25,165,824

Wake on CAN Datafield content value = Wake on CAN Datafield bitmask

Deriving Wake on CAN "CAN identifier" and "CAN ID bitmask":
The precise ID of the message is to be found – i.e. all the positions of the ID that are not leading zeros are significant. All the significant positions are to be assigned the value of the ID.
This means:

CAN identifier = Message D = 100

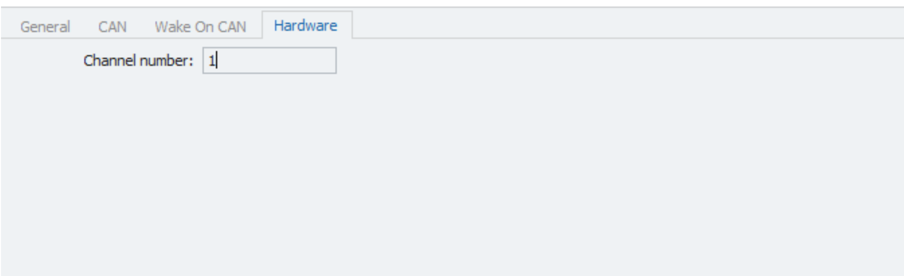
Deriving the significant bit positions of the ID:
ID (dec) = 100 => ID (bin) = 1100100
The ID also has 7 significant positions. So the mask must also be 7 bits long. All positions must be assigned "1".
"CAN ID bitmask" = 1111111 = 127 (dec)

Each node can thus be assigned individual start settings. Remember: During the booting phase, at the node that started the logger, the WoC LED on the front plate of the interface blinks quickly for approximately ten seconds.

> Even if WoC (Wake on CAN) is set as start condition the logger starts on clamp 15. So start on clamp 15 has priority over WoC. This is necessary to have a fallback solution in case a start message or start value can no longer be sent.
> If the logger was started with clamp 15, it can likewise only be shut down by "pulling" Clamp 15 – based on the principle, the source that starts, also stops – providing there is not simultaneously another wake condition (e.g. WoC) active.

5.1.3.4 Hardware (Channel number)

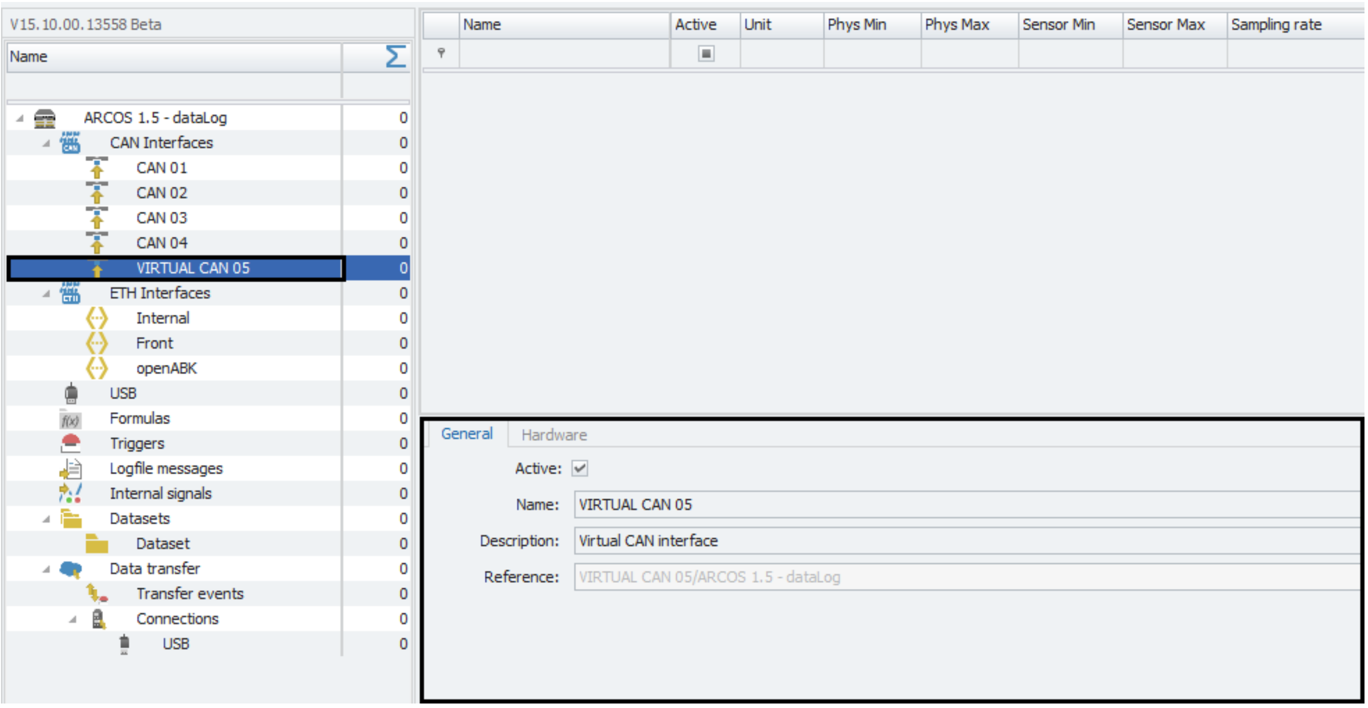
This tab allows you to set a **Channel number** for the selected CAN channel. This channel number has to be unique within the CAN interface.



> For better orientation and in order to avoid confusion regarding Channelnumbers and -names, a Channels physical number can be found in the logger's "**Web Interface**" and set accordingly.

5.1.4 Virtual CAN settings

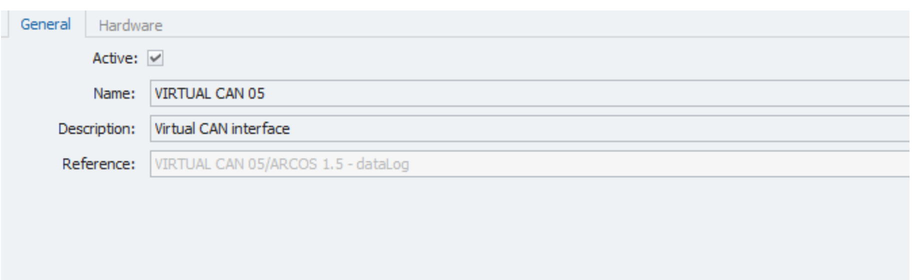
By selecting one of the Virtual CAN channels in the tree you will be able to define this channel's settings in the details area.



The same settings described in this section as part of the Details area can also be adjusted when selecting the tree element **CAN interfaces** and then directly changing the desired setting in the respective field of the **Grid area**.

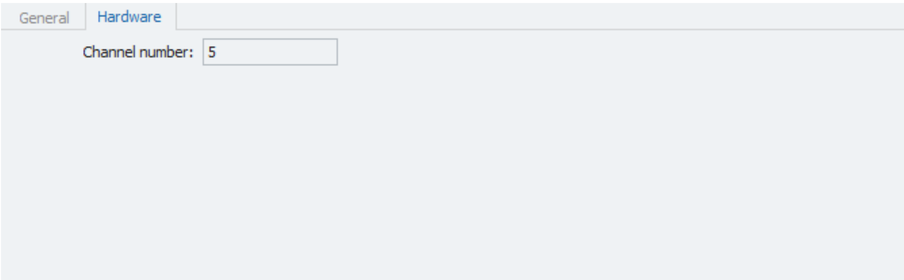
5.1.4.1 General

This tab allows you to give a user specific name to your for the selected Virtual CAN channel if wished and add an additional description. The Reference field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed. The **"Active"** checkbox allows you to activate or deactivate the selected Virtual CAN channel.



5.1.4.2 Hardware (Channel number)

This tab allows you to set a **Channel number** for the selected Virtual CAN channel. This channel number has to be uniqe within the CAN interface.



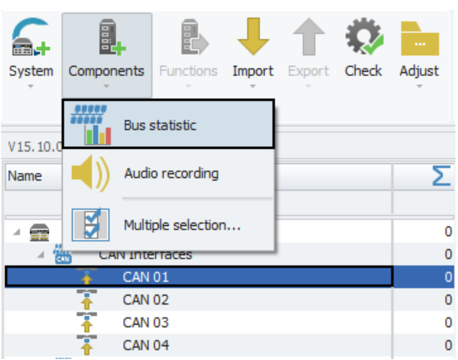
For better orientation and in order to avoid confusion regarding Channelnumbers and -names, a Channels physical number can be found in the logger's **"Web Interface"** and set accordingly.

5.1.5 CAN channel Bus statistic

The "Bus statistic" provides a range of statistics and status signals for the respective CAN channel. It contains information on the current state of the Bus, the Busload, as well as on the messages that have been received and errors that ocurred.

The "Bus statistic" only shows statistics for the channel to which it belongs. For each channel that you would like to see the statistic, you will have to add the component "Bus statistic".

5.1.5.1 Adding Bus statistics Select the channel in the tree for which you would like to add "Bus statistic", then click the "Components" button in the Ribbon and choose "Bus statistic".



5.1.5.2 Bus statistic signals Once the component "Bus statistic" has been added to your channel, it will appear in the measurement task tree as a child element of this channel and the grid area will give you an overview of the available signals.

The signals included in "Bus statistics" are of the type "Internal signal" and may be adjusted in the same way. For more information on "Internal signals" please refer to (→5.14).

Overview of signals

Subtype	Meaning	Unit
Controller state	nan= Channel not available 1= Bus on 2= Bus warning 3= Bus off	-
Busload (%)	Bus load of a CAN/LIN channel	(%)
Number of messages	Number of messages since beginning of measurement	-
Message rate total	Current bus load	(frames/s)
Number of messages with standard ID	Number of messages with standard ID	-
Number of messages with extended ID	Number of messages with extended ID	-
Number of messages with remote standard ID	Number of messages with remote standard ID	-
Number of messages with remote extended ID	Number of messages with remote extended ID	-
Number of error frames	Number of error frames	-
Message rate of standard IDs	Messages with standard ID	(frames/s)
Message rate of extended IDs	Messages with extended ID	(frames/s)
Message rate of standard IDs	Messages with remote standard ID	(frames/s)
Message rate of extended IDs	Messages with remote extended ID	(frames/s)
Error frame rate	Average of errors per second	(frames/s)

5.2 CAN signals

5.2.1 Storage method

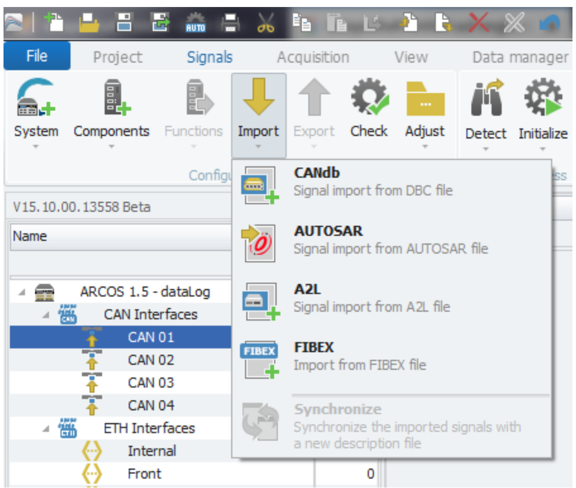
In order to store incoming signals on a CAN channel use one of the following signal storage methods.

- ATFX (→ 7.4)
- MDF 4.0 (→ 7.5)
- MDF 4.1 (→ 7.6)

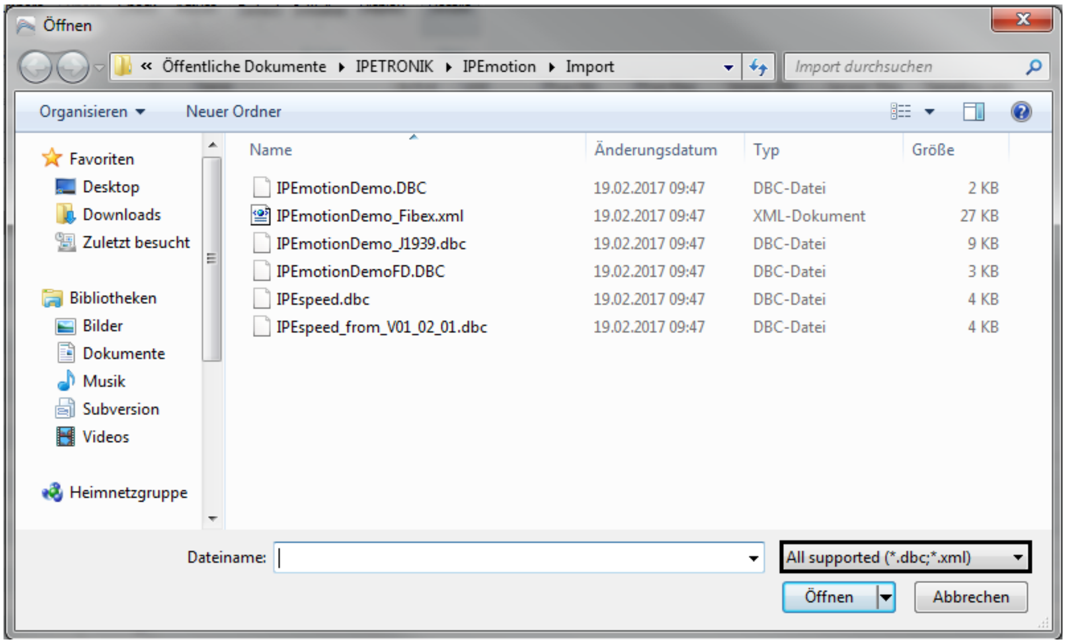
5.2.2 Importing CAN signals

This section explains how to import CAN signals. There are three different filetypes which can be used in order to import a single CAN signal or a group of CAN signals:
CANdb (DBC file), Autosar and Fibex.
The import procedure for all of these filetypes is the same and will be explained using the example of the CANdb import.

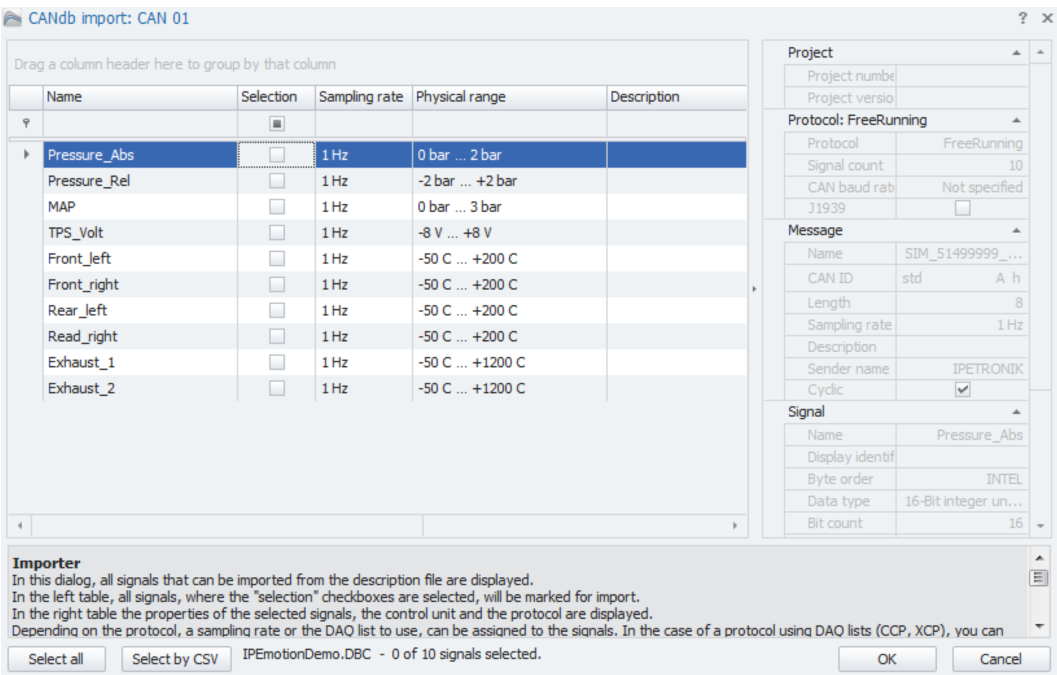
To import Signals, select the CAN channel to which you wish to import your signal in the tree, click the **"Import"** button in the ribbon and then choose which filetype the **"description file"**, you wish to use for the import, has. For more information on the "description file" refer to (→5.2.4.1).



The following window lets you choose which file you wish to import. According to the file-type you have chosen earlier for your import, you will now be able to choose files of the respective filetype. The dropdown menu on the bottom right of the window shows you, which filetypes are available. Choose the file you wish to import and click "**Open**".



Once you have opened your file, the "**Importer**" window will appear, that will present you with a range of importing options.



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. You can either choose manually, which signals to import, you can use the "**Select/Deselect all**" button on the bottom left, or you can use a CSV-file to determine which signals are to be imported, by clicking "**Select by CSV**" on the bottom left.

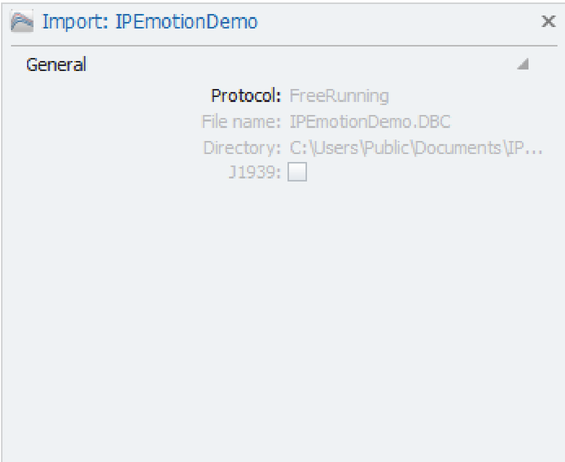
In the right table the metadata, properties of the selected signals, the control unit and the protocol are displayed.

Once you have choosen all the signals you wish to import, click "OK" to complete the import procedure.

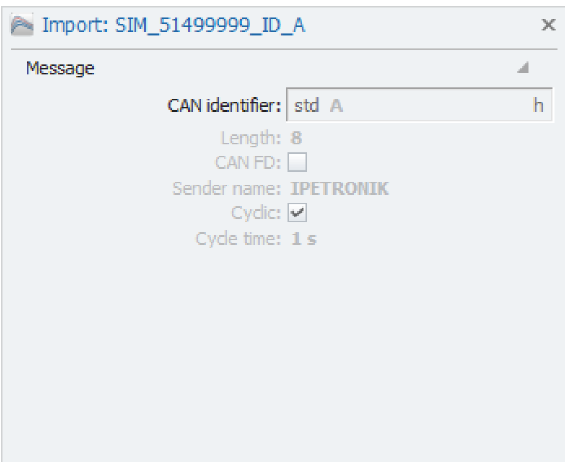
Multiple description files can be imported into the same CAN channel.

5.2.3 Import properties

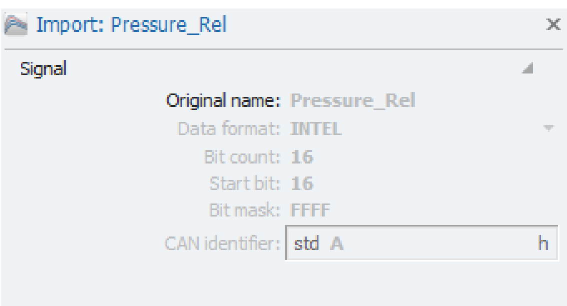
The "Import properties" of a description file, Message or signal allow you to see certain properties such as the Data format, The CAN identifier, the Bit mask, the start bit, bit count and more. It shows the signal's properties as described by the description file.



Example for "Import properties" of a CAN description file

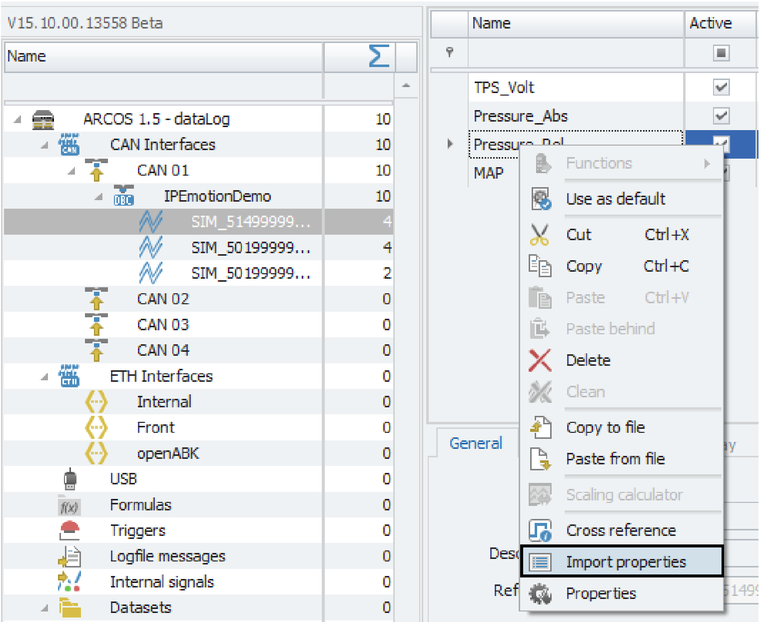


Example for "Import properties" of a CAN Message



Example for "Import properties" of a CAN signal

To access the "Import properties" rightclick on any desired description file, Message or signal and then choose "Import properties" from the resulting context menu.



5.2.4 Signal properties

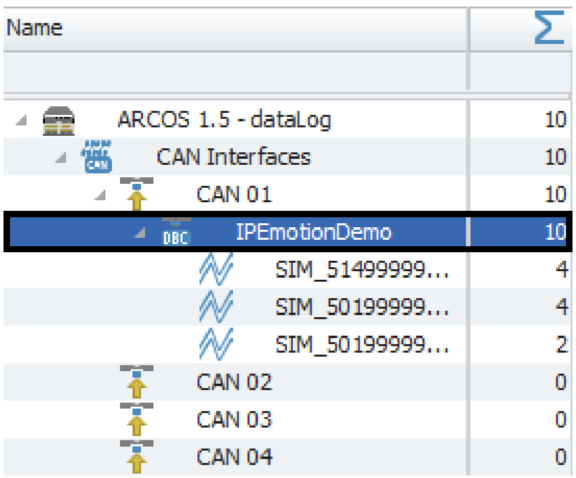
5.2.4.1 Tree elements for CAN signals

After having successfully imported the desired signals to your CAN channel, this channel will contain two new layers of child elements in the measurement task tree: The **"Description file"** and the **"Message"**.

Description file

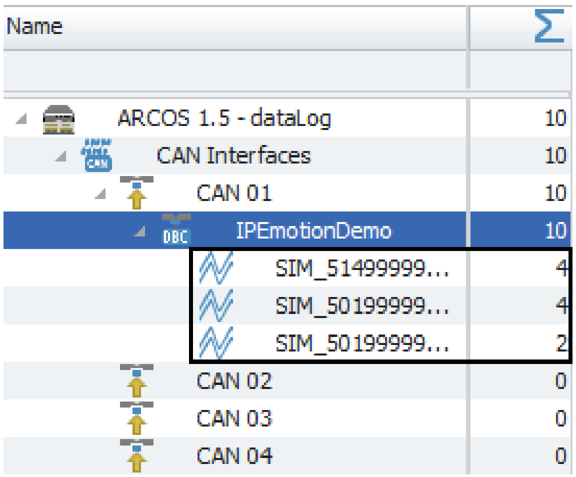
The "description file" is a database file which contains signal information and can be used to import those signals into a Signal channel in IPEmotion. The filetypes which are supported by the CAETEC Plugin for IPEmotion depend on the type of signal you wish to import.

The symbol in the left part of the tree element shows you the type of "description file" you imported (in this case a "DBC" file), then follows the name of the imported "description file" (in this case "IPEmotionDemo") and on the right is a number indicating how many signals the "description file" contains (in this case 10).



Message

Each “description file” can contain one or more “Messages” (in this case 3), which then contain the actual signals. A “Message” can be found in the “Measurement task tree” as a child element of the “description file”, it belongs to.
Each “Message” can, again, contain one or more signals (in this case the three “Messages” contain 4, 4 and 2 signals), which is indicated by the number on the right of the “Message’s” name.



5.2.4.2 Grid area for CAN signals

In the “grid area” you will be presented with an overview of your selected CAN channel’s signals. Also you can find here two important functions, which are the **“Column chooser”** (→4.3.1) and the **“Filter editor”** (→4.3.2).

5.2.4.3 Details area for CAN signals

The Details area shows settings either for the selected tree element (“description file” or “Message”) or the selected signal in the grid area. In case a tree element is selected, the details area will only show the “General” tab. Please refer to (→4.2.2).

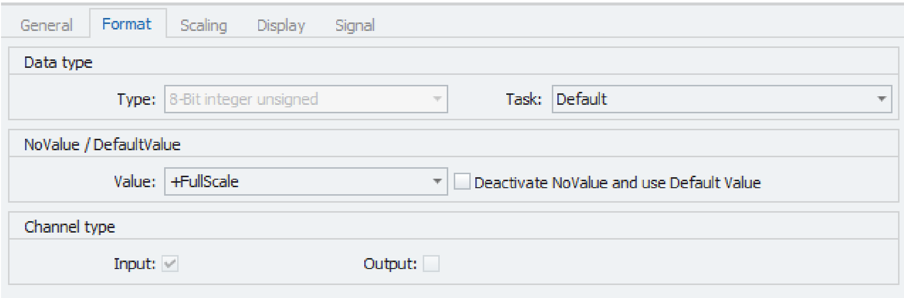
In case a signal is selected in the grid area, the details area will contain additional tabs which will be explained in the following.

General

Please refer to (→4.2.2).

Format

This tab contains information and options regarding file format, tasks and Channel type.



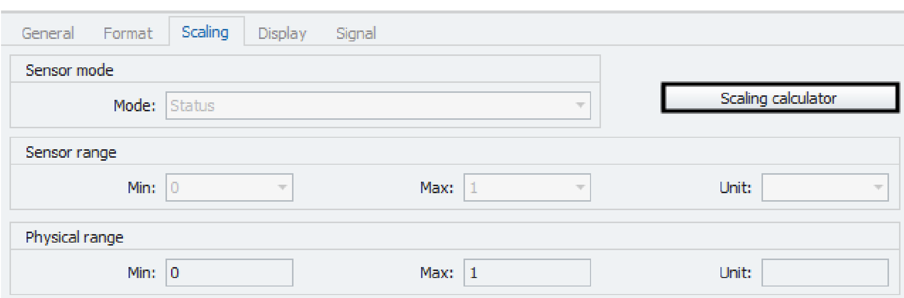
- Data type
  This field tells you the type of data (in this case **“8-Bit integer unsigned”**) and allows you to apply special tasks for this signal such as “GPS Longitude”, “GPS Latitude”, “UTC hour”, “Audio mono” and more.

- NoValue / DefaultValue
  This field allows you to define the value that will be shown if a signal value is read as invalid.

- Channel type
  This field tells you whether you are dealing with a **“Input”** channel or **“Output”** channel.

Scaling

The fields accessible directly through the tab allow for basic scaling operations to convert analog measurement in engineering units. The **“Scaling calculator”** allows for more refined scaling options with a large range of functions. For details on how to use the “Scaling calculator” please refer to the **IPEmotion Documentation - Section 3.4.5 “Channel configuration and scaling”**.



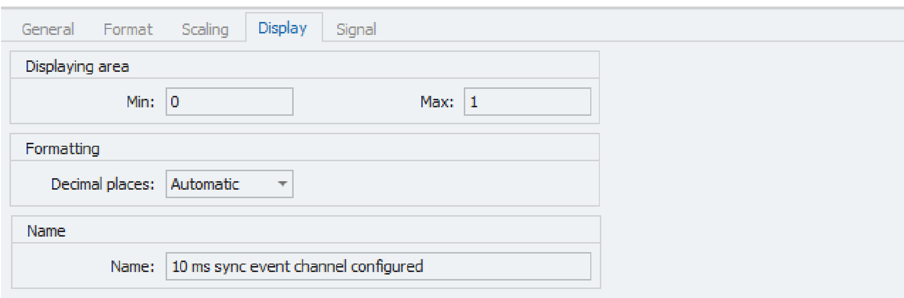
- Sensor Mode
  The sensor mode tells the type of signal. It can be of different types such as "Status", "Voltage", "Frequenzy" or others. It cannot be changed and serves for IPEmotion to know what kind of signal it is dealing with.

- Sensor Range
  Shows the raw value range of the signal.

- Physical Range
  Allows you to set a range to which you would like to "scale" your signal and also define the unit to use. For more refined scaling please use the "Scaling calculator" and refer to the **IPEmotion Documentation - Section 3.4.5 "Channel configuration and scaling"**.

Display

This tab allows you to define what information about the current signal will be shown on a display if one is connected.



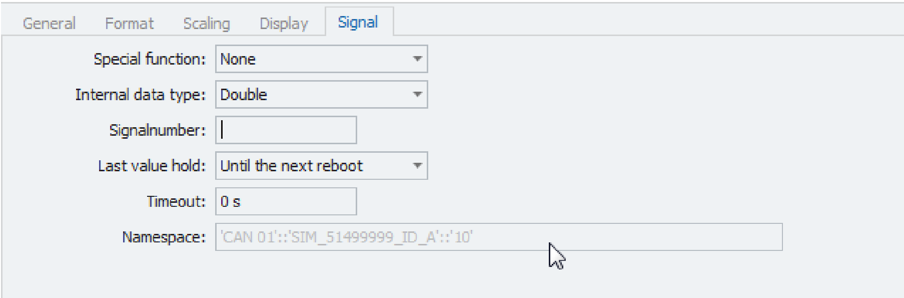
- Displaying area
  Shows the value range which will be shown on a display. It usually should match the "Physical range" from the "Scaling" tab.

- Formatting
  The dropdown menu **"Decimal places"** allows you to set how many decimal numbers of the value will be shown on a display.

- Name
  Allows you to set a Name to be shown on a display.

Signal

This tab allows you to define signal settings.



- Internal data type
  Assign an internal data type to the signal. Available data types are "Double" and "String".

- Signal number
  Assign a number to the current signal. This way you will later be able to sort the signals in the grid according to their "Signal numbers".

- Last value hold
  Specify, for how long the last value of the signal will be hold.

- Timeout
  Specify the timeout period for the current signal. If the data source doesn't send data for the specified time period, the value of the signal is set to "NaN (Not a Number)" and will be displayed as "–" in a display.

- Namespace
  The "Namespace" serves as unique identifier for the signal inside the logger.

5.3 CCP/XCP signals

5.3.1 Storage method

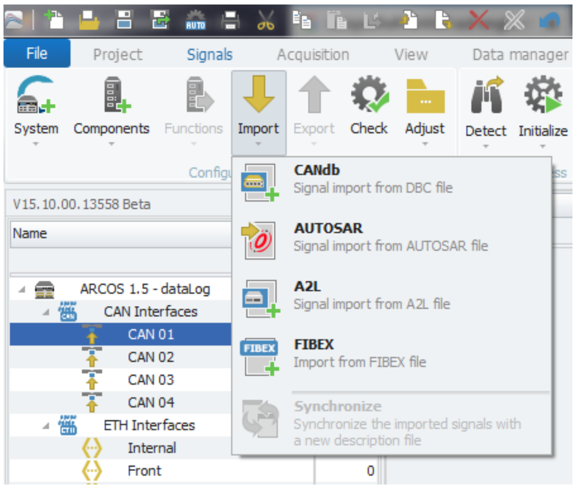
In order to store incoming signals on a CAN channel use one of the following signal storage methods.

- ATFX (→ 7.4)
- MDF 4.0 (→ 7.5)
- MDF 4.1 (→ 7.6)

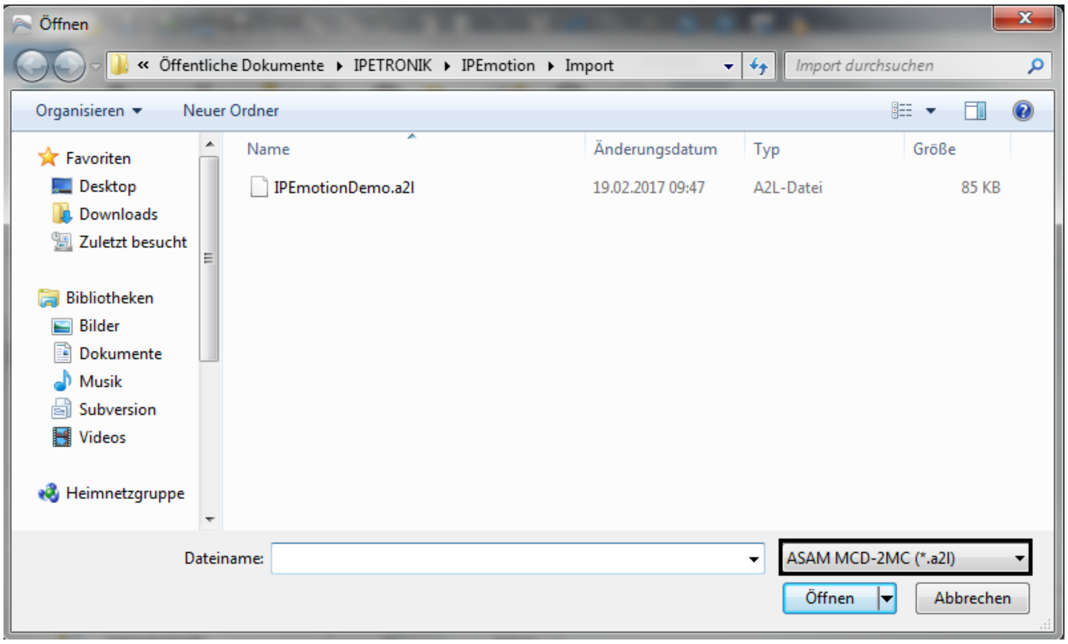
5.3.2 Importing CCP/XCP signals

This section explains how to import **CCP** or **XCP signals**. The filetype associated with these signals is a **"A2L file"**.

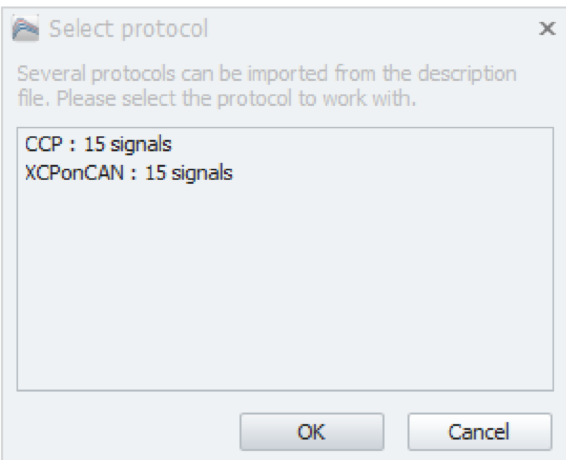
To import a CCP/XCP signal, select the CAN channel to which you wish to import your signal in the tree, click the **"Import"** button in the ribbon and then choose the "A2L" filetype for the import.



The following window lets you choose which file you wish to import. According to the filetype you have chosen earlier for your import, you will now only be able to choose files of the "A2L" filetype. Choose the file you wish to import and click "**Open**".

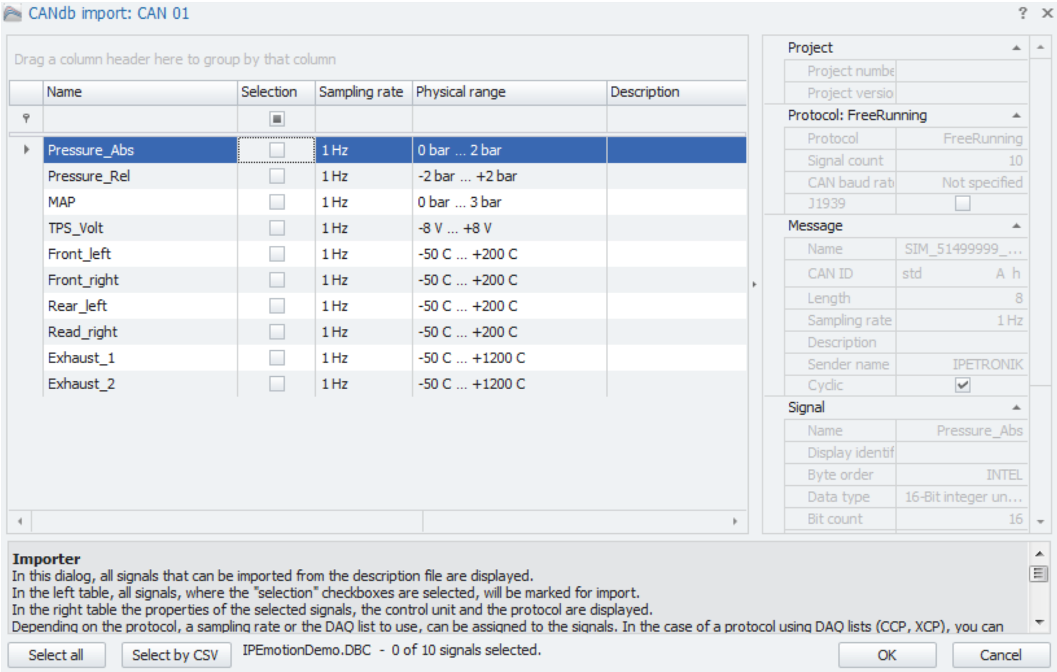


The following window lets you choose whether you want to import a "CCP" signal or a "XCP" signals. Choose the protocol you wish to import and click "**OK**".



Once you have chosen the protocoll and confirmed, the "**Importer**" window will appear, that will present you with a range of importing options.
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. You can either choose manually, which signals to import, you can use the "**Select/Deselect all**" button on the bottom left, or you can use a CSV-file to determine which signals are to be imported, by clicking "**Select by CSV**" on the bottom left.

In the right table the metadata, properties of the selected signals, the control unit and the protocol are displayed.



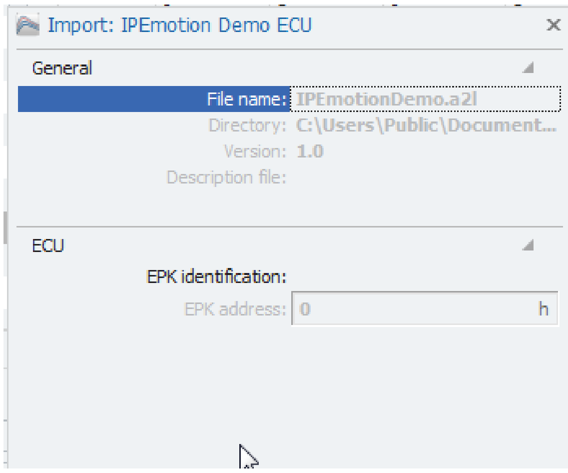
Once you have choosen all the signals you wish to import, click "OK" to complete the import procedure.

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 the sampling rate or a DAQ list. To achieve this you should open the column selection dialog, via the context menu of the table header, and then drag the desired column ("sampling" or "DAQ list") from the column selection dialog to the table header. The other column is removed automatically.
If in a protocol based on DAQ lists, sampling rates are used for the signal configuration, during import the signals are assigned to the available DAQ list with the most suitable sampling rate.
In case of protocols supporting array signals, you can specify via the "split array" column whether all the signals of the array or just the first to be imported. If this column does not appear it can be moved from the column selection dialog into the table.

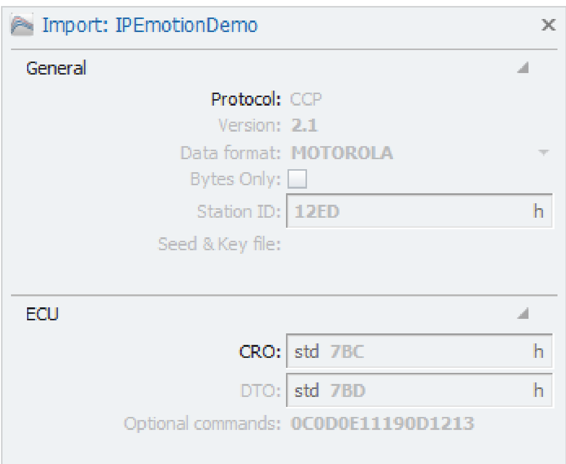
Multiple description files can be imported into the same CAN channel.

5.3.3 Import properties

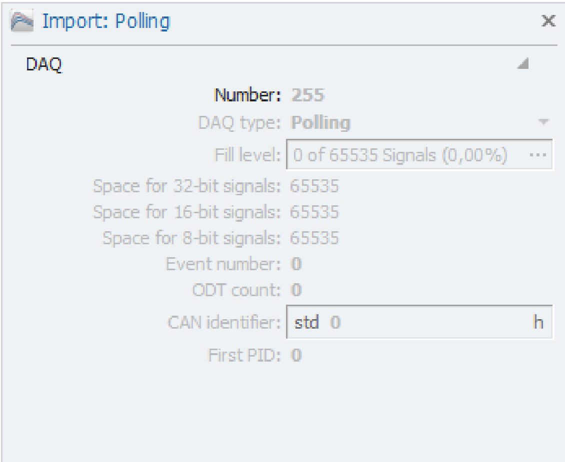
The “Import properties” of an “ECU”, “Description file”, “Polling list”, “DAQ list” or signal allow you to see certain properties such as the Data format, the Adress, the Bit mask, the start bit, bit count and more. It shows the properties as described by the description file.



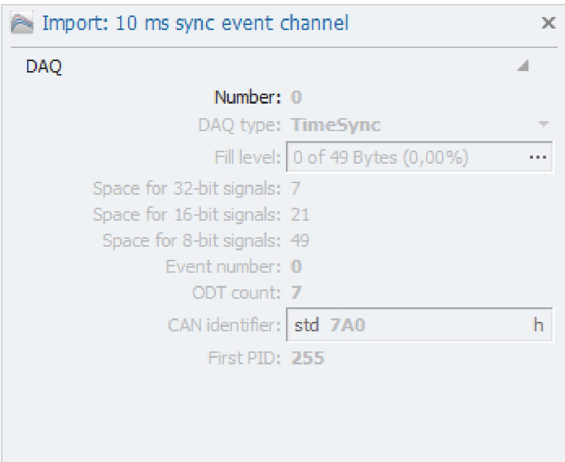
Example for Import properties of a CCP/XCP ECU



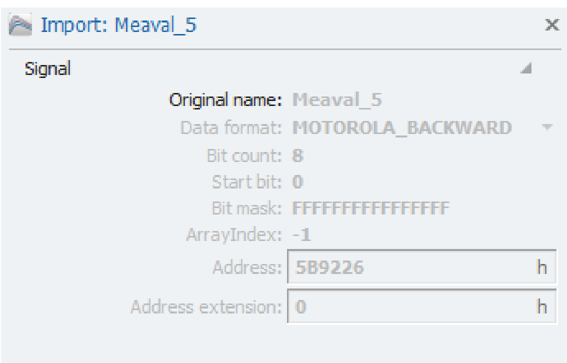
Example for Import properties of a CCP/XCP Description file



Example for Import properties of a CCP/XCP Polling list



Example for signal properties of a CCP/XCP DAQ list



Example for Import properties of a CCP/XCP signal

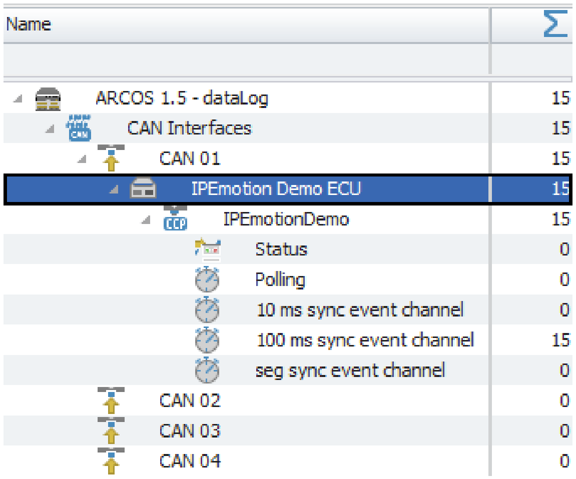
5.3.4 Signal properties

5.3.4.1 Tree elements for CCP/XCP signals

After having successfully imported the desired signals to your CAN channel, this channel will contain three new layers of child elements in the measurement task tree: The **"ECU"-layer**, the **"Description file"-layer** and the **"Message"-layer**.

ECU

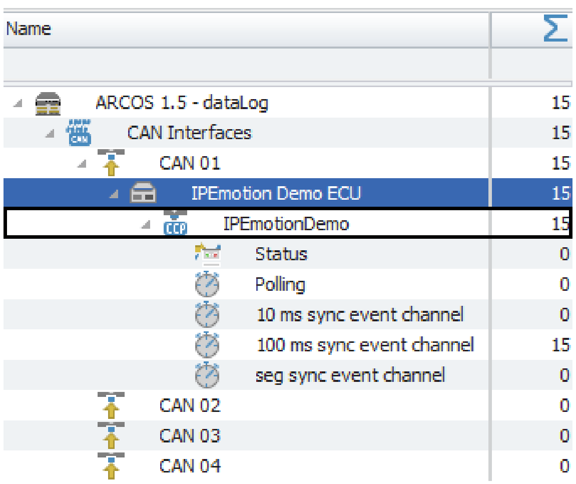
The "ECU" (Electronic control unit) represents the control unit inside the vehicle that the logger communicates with. It allows for an active communication between logger and vehicle, where the logger can not only receive messages but also send messages.



Description file

The "description file" is a database file which contains signal information and can be used to import those signals into a Signal channel in IPEmotion. The filetype associated with CCP/XCP signals is the "A2L file".

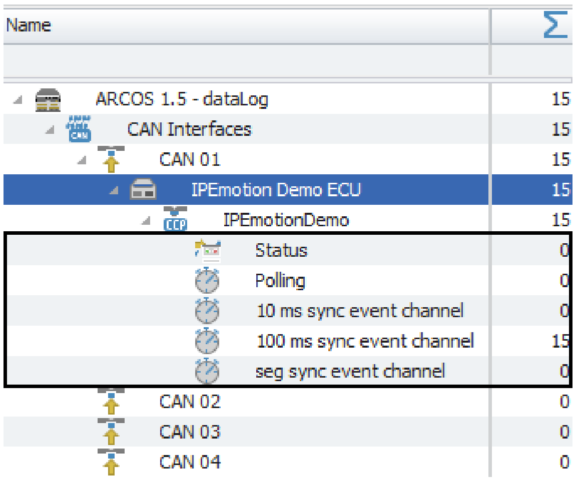
The symbol in the left part of the tree element shows you the type of "database" you imported (CCP or XCP), then follows the name of the imported "description file" (in this case "IPEmotionDemo") and on the right is a number indicating how many signals the "description file" contains (in this case 15).



Signal lists

Each "description file" can contain one or more Signal lists, which then contain the actual signals. A Signal list can be found in the "Measurement task tree" as a child element of the "description file", it belongs to.
Each Signal list can, again, contain one or more signals, which is indicated by the number on the right of the Signal list's name.

CCP/XCP Signal lists group the "description file's" signals in groups with different functionalities. The groups differ in the way that the logger obtains or requests a signal.



- **Status list**

  The status list contains signals on the ECU status and the status of DAQ lists included in the description file. Each signal list has two entries. "Configured" gives information whether the list has been configured, "started" gives information whether the respective list has been started and is running or not.
  "Station connected" tells you whether the respective ECU is connected and "EPK Versioncheck result" lets you know whether the "EPK check" has been passed successfully or not.

  Station connected

  Tells you the status of the connection with the respective ECU.

NaN = not started	The ECU has not been started.
0 = failed	The ECU has been started but the connection test has failed.
1 = successful	The ECU has been started and the connection test has been passed successfully.

  EPK Versioncheck result

  Tells you about the result of the "EPK check".

NaN = not started	The "EPK check" has not been performed.
0 = failed	The "EPK check" has been performed but not passed.
1 = successful	The "EPK check" has been performed and passed successfully.

xxx_ms_task_configured

Tells you whether the respective DAQ list has been configured.

0 = not yet configured	The DAQ list has not yet been configured.
1 = successful	The DAQ list has been configured.

xxx_ms_task_started

Tells you whether the respective DAQ list has been started.

0 = not yet started	The DAQ list has not been started. It is inactive.
1 = successful	The DAQ list has been started.

- Polling list
  Signals contained in a "Polling list", will be actively requested by the logger. That means, for each signal a sampling rate has to be defined, according to which the logger will request the signals which will then be sent to the logger by the ECU.

- DAQ list
  Signals contained in a "DAQ list" are assigned with a certain time intervall in which the ECU is requested to send these signals. I.e. any signal contained in the DAQ list **"100ms sync event channel"** will be sent to the logger in an intervall of 100ms. This guarantees that there is no unintended delay.

5.3.4.2 Grid area for CCP/XCP signals

In the "grid area" you will be presented with an overview of your selected CAN channel's signals. Also you can find here two important functions, which are the **"Column chooser"** (→4.3.1) and the **"Filter editor"** (→4.3.2).

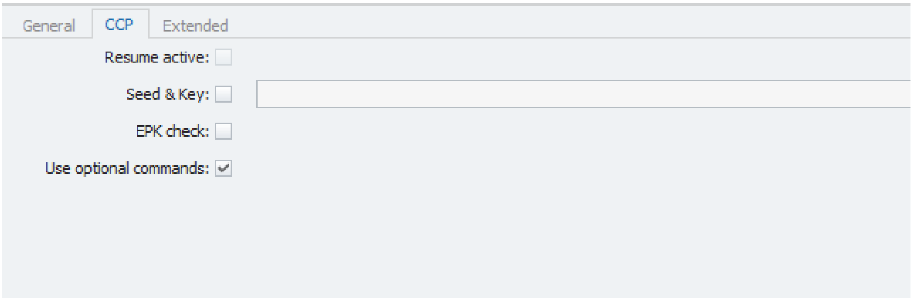
5.3.4.3 Details area for CCP/XCP signals

The Details area shows settings either for the selected tree element ("ECU","description file" or "signal list") or the selected signal in the grid area.

- **"ECU" selected**
  In this case the details area will only show the "General" tab. Please refer to (→4.2.2).

- "description file" selected
  In this case the details area will contain the "General" tab (→4.2.2) plus two additional tabs.

  CCP/XCP
  This tab contains CCP and XCP specific options.

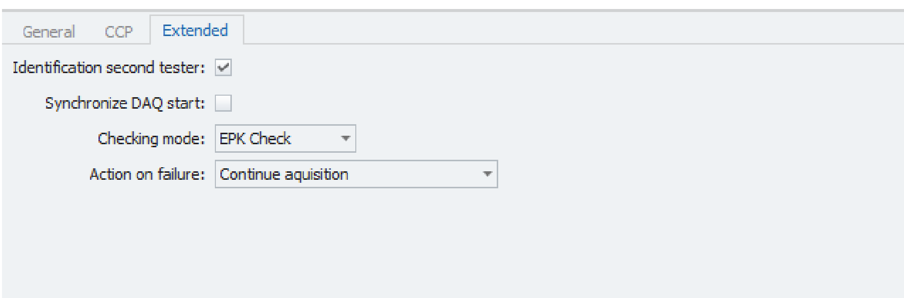
  – Resume active
    This option is not supported by the plugin at the moment

  – Seed & Key
    This field allows you to enter a Seed & Key binary file (*.skb) which contains the information required to unlock the ECU if necessary.

  – EPK check
    Checks for differences in the chekcsums of the current configuration and the ECU.

  – Use optional commands (CCP only)
    If marked active, the optional commands defined in the ECU file will be used.

  Extended
  This tab contains CCP and XCP specific extended options.

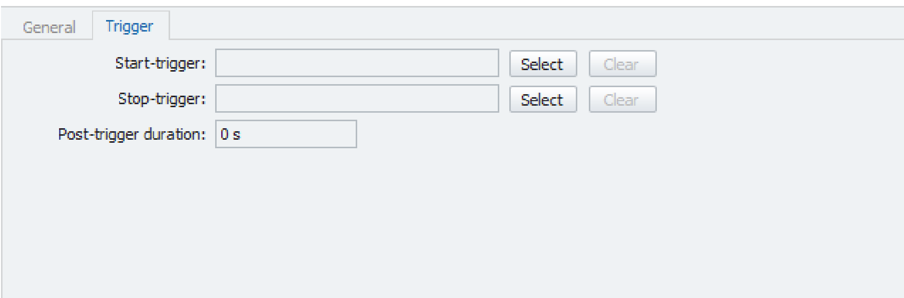
  – Identification second tester
    Allows you to activate or deactivate the detection of a second master at the bus.

- Synchronize DAQ start
  Allows you to activate or deactivate the synchronization of the start of all DAQ-lists.

- Checking mode
  The dropdown menu allows you to choose the checking mode. It can be either **"EPK"** or **"GETID"**. "GETID" is only available for XCP.

- Action on failure
  Allows you to define what action to take upon a EPK check failure. Regardless of the selected option the logger will always write an EPK check failure logging message.

• "Polling list" or "DAQ list" selected
In this case the details area will additionally contain the **"Trigger tab"**. The "Trigger tab" allows you to set a trigger upon whose activation the signals contained in the list will be requested.



- Start-trigger
  Allows you to choose a trigger upon whose activation the list's Signals will be requested. A trigger has first to be defined. Please refer to (→6).

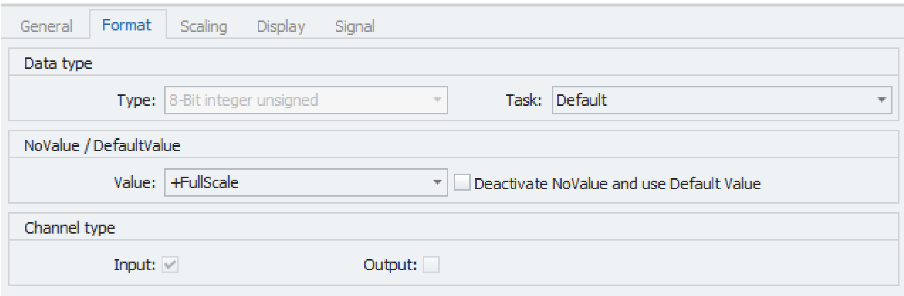
- Stop-trigger
  Allows you to choose a trigger upon whose activation the list's Signals will stop being requested. A trigger has first to be defined. Please refer to (→6).
  If there is no "Stop-trigger" defined the acquisition will be stopped by inverted start condition.

- Post-trigger duration
  You can determine here for how long after a stop request (either by explicit "Stop-trigger" or inverted start condition) the stop will be delayed and data acquisition will continue.

- Signal selcted
  In this case the details area will contain additional tabs which will be explained in the following.

  General
  Please refer to (→4.2.2).

  Format
  This tab contains information and options regarding file format, tasks and Channel type.



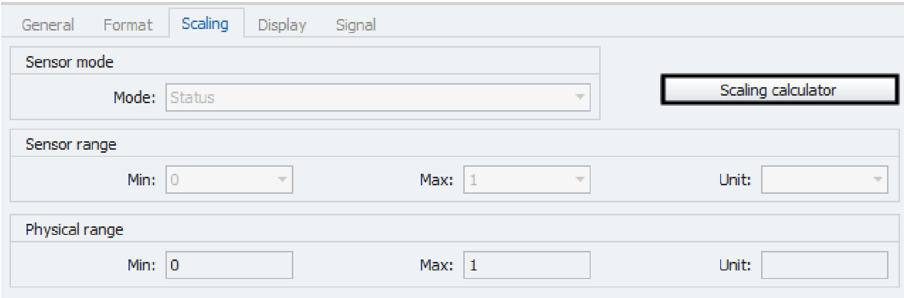
  - Data type
    This field tells you the type of data (in this case "8-Bit integer unsigned") and allows you to apply special tasks for this signal such as "GPS Longitude", "GPS Latitude", "UTC hour", "Audio mono" and more.

  - NoValue / DefaultValue
    This field allows you to define the value that will be shown if a signal value is read as invalid.

  - Channel type
    This field tells you whether you are dealing with a "Input" channel or "Output" channel.

Scaling

The fields accessible directly through the tab allow for basic scaling operations to convert analog measurement in engineering units. The **"Scaling calculator"** allows for more refined scaling options with a large range of functions. For details on how to use the "Scaling calculator" please refer to the **IPEmotion Documentation - Section 3.4.5 "Channel configuration and scaling"**.



- Sensor Mode

  The sensor mode tells the type of signal. It can be of different types such as "Status", "Voltage", "Frequenzy" or others. It cannot be changed and serves for IPEmotion to know what kind of signal it is dealing with.

- Sensor Range

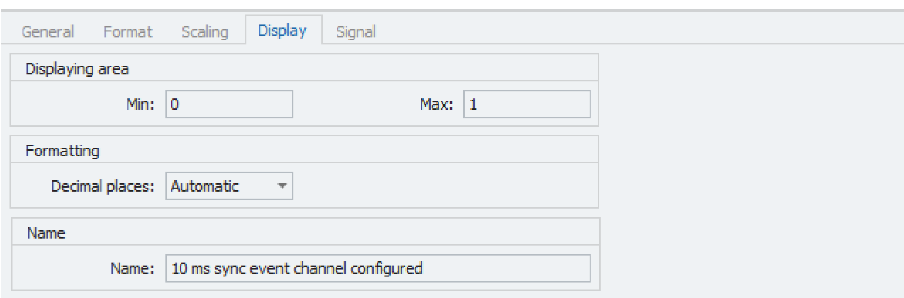
  Shows the raw value range of the signal.

- Physical Range

  Allows you to set a range to which you would like to "scale" your signal and also define the unit to use. For more refined scaling please use the "Scaling calculator" and refer to the **IPEmotion Documentation - Section 3.4.5 "Channel configuration and scaling"**.

Display

This tab allows you to define what information about the current signal will be shown on a display if one is connected.



- Displaying area

  Shows the value range which will be shown on a display. It usually should match the "Physical range" from the "Scaling" tab.

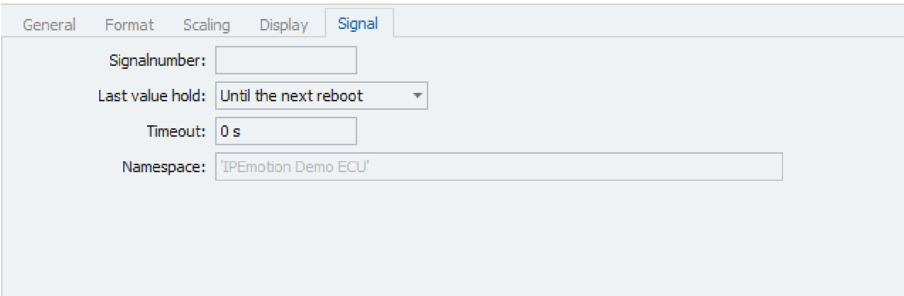
- Formatting

The dropdown menu **"Decimal places"** allows you to set how many decimal numbers of the value will be shown on a display.

– Name
  Allows you to set a Name to be shown on a display.

Signal

This tab allows you to define signal settings.



– Signal number
  Assign a number to the current signal. This way you will later be able to sort the signals in the grid according to their "Signal numbers".

– Last value hold
  Specify, for how long the last value of the signal will be hold.

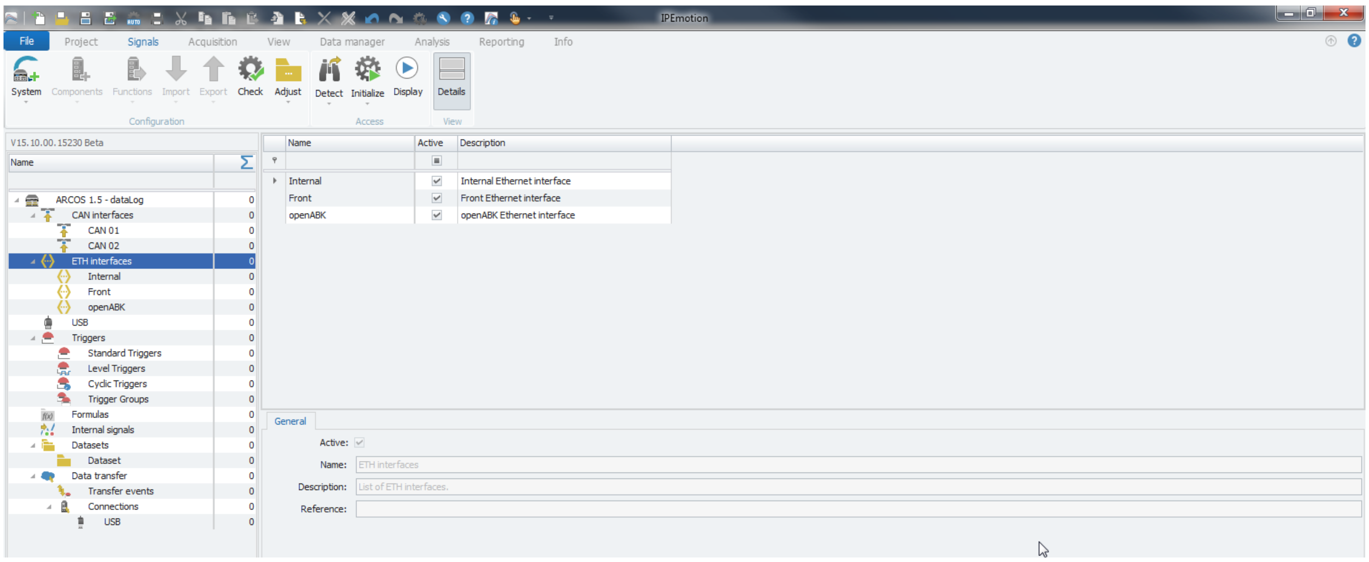
– Timeout
  Specify the timeout period for the current signal. If the data source doesn't send data for the specified time period, the value of the signal is set to "NaN (Not a Number)" and will be displayed as "–" in a display.

– Namespace
  The "Namespace" serves as unique identifier for the signal inside the logger.

5.4 ETH channels

All the ETH channels for your system are located in the tree element "ETH interfaces". According to the default settings, the tree element "ETH interfaces" will include a preset number of ETH channels. By clicking the tree element ETH Interfaces you will see all of its channels and signals in the grid area as well as a tab called **General** in the **Details area** which allows you to set a name and description. These settings apply to the entire element "ETH interfaces".

In the following will be described how to add ETH channels and adjust their settings (→5.4.3).

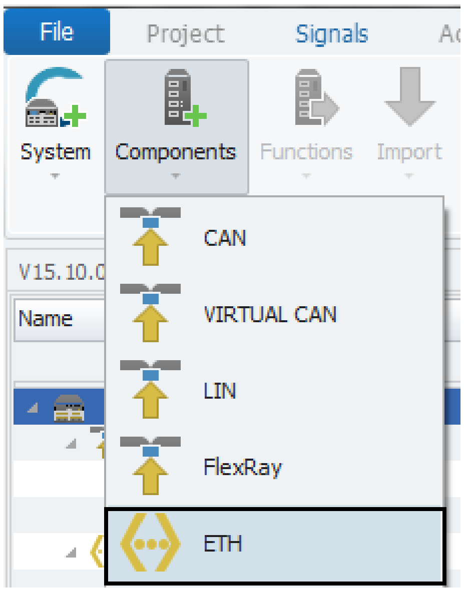


5.4.1 Storage method

In order to store all incoming traffic on an ETH channel use **"PCAP"** as a bus tracing method for storage. Please refer to (→ 7.8).

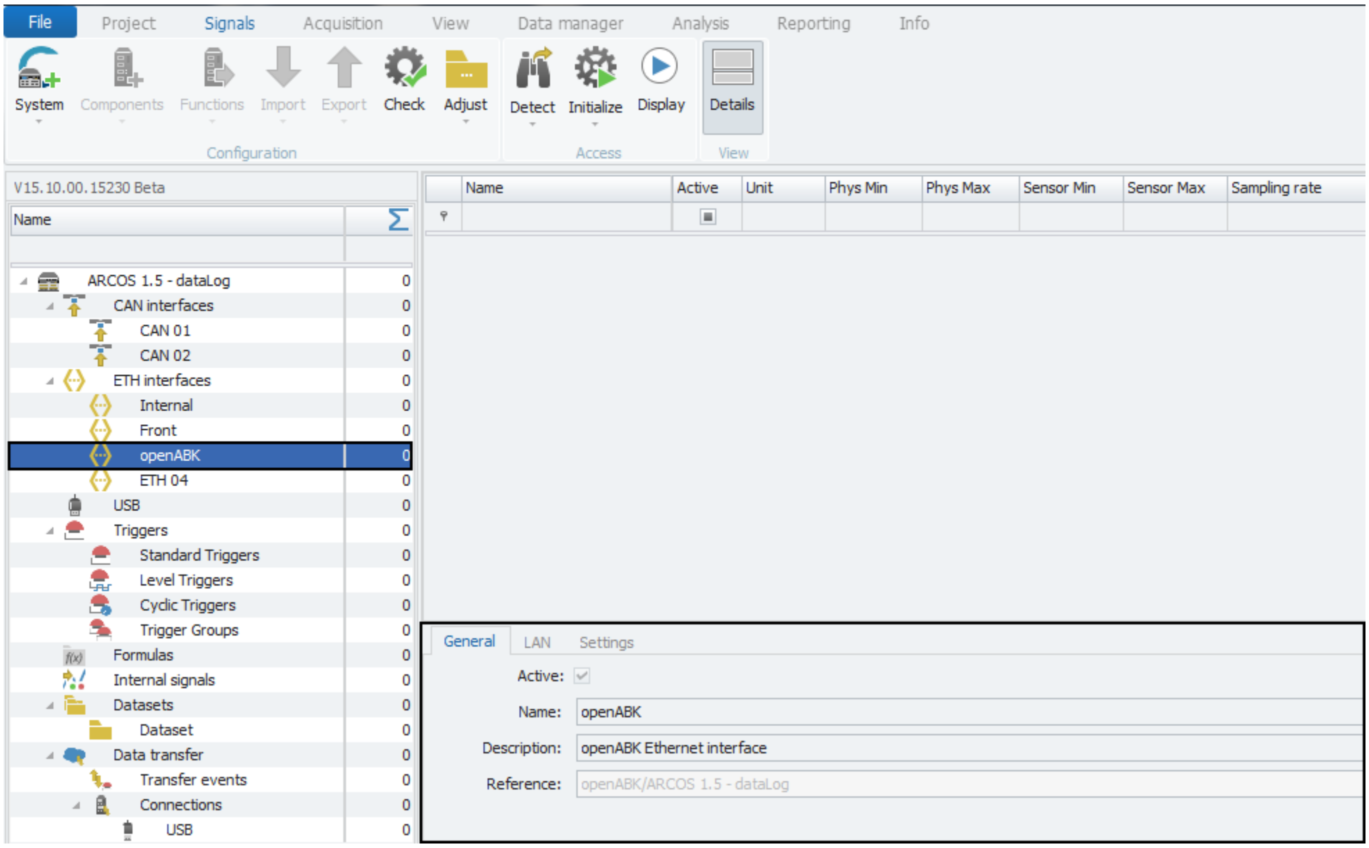
5.4.2 Adding ETH channels

ETH channels can be added by selecting the tree element "**ETH interfaces**", then clicking the "**Components**" button and finally choosing the desired type of ETH channel you wish to add.



5.4.3 ETH settings

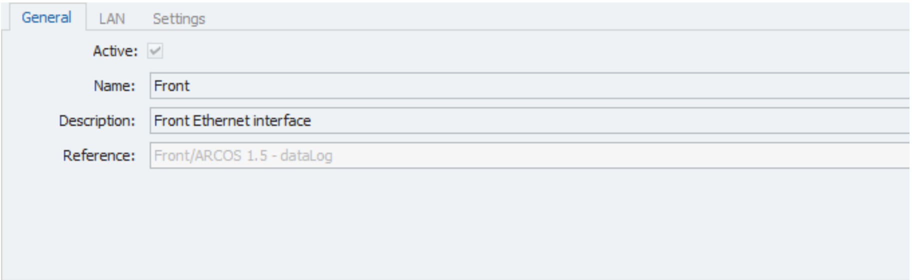
By selecting one of the ETH channels in the tree you will be able to define this channel's settings in the details area.



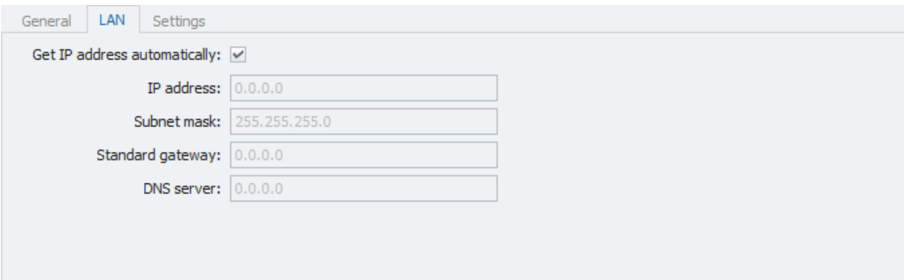
The same settings described in this section as part of the Details area can also be adjusted when selecting the tree element **ETH interfaces** and then directly changing the desired setting in the respective field of the **Grid area**.

5.4.3.1 General

This tab allows you to give a user specific name for the selected ETH channel if wished and add an additional description. The Reference field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed. The **Active** checkbox is by default marked as activated and cannot be changed.



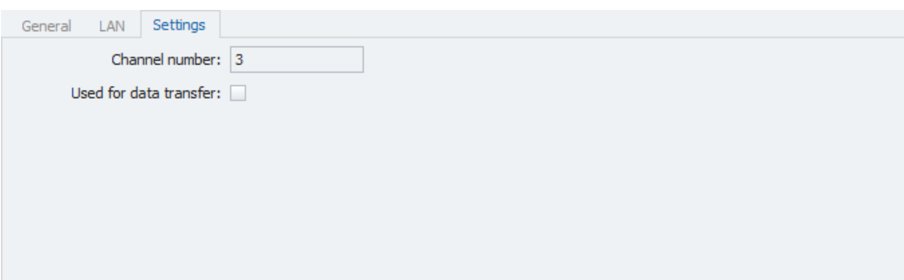
5.4.3.2 LAN



This tab allows you to set your current ETH channel's LAN settings. Whether it should receive an IP address automatically or not and consequently to manually set the channel's IP settings.

> For the tree elements **"Open ABK"** and **"Inernal"** the IP settings cannot be adjusted.

5.4.3.3 Settings



This tab allows you to set your current ETH channel´s physical channel number and whether it should be used for data transfer.

> The ETH channel “**Internal**” cannot be used for data transfer.

5.5 Signals on ETH

For data acqusition you can import **A2L** databases on the “Front” channel of your “ETH interface” or on any manually added ETH channel. The protocol for importing A2L databases is **XCPonUDP**.

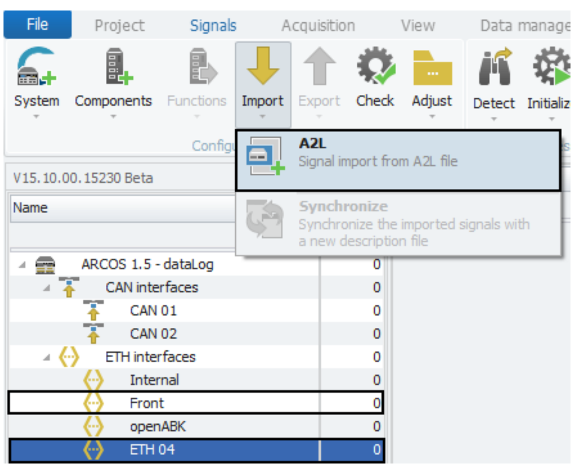
5.5.1 Storage method

In order to store incoming signals on an ETH channel use one of the following signal storage methods.

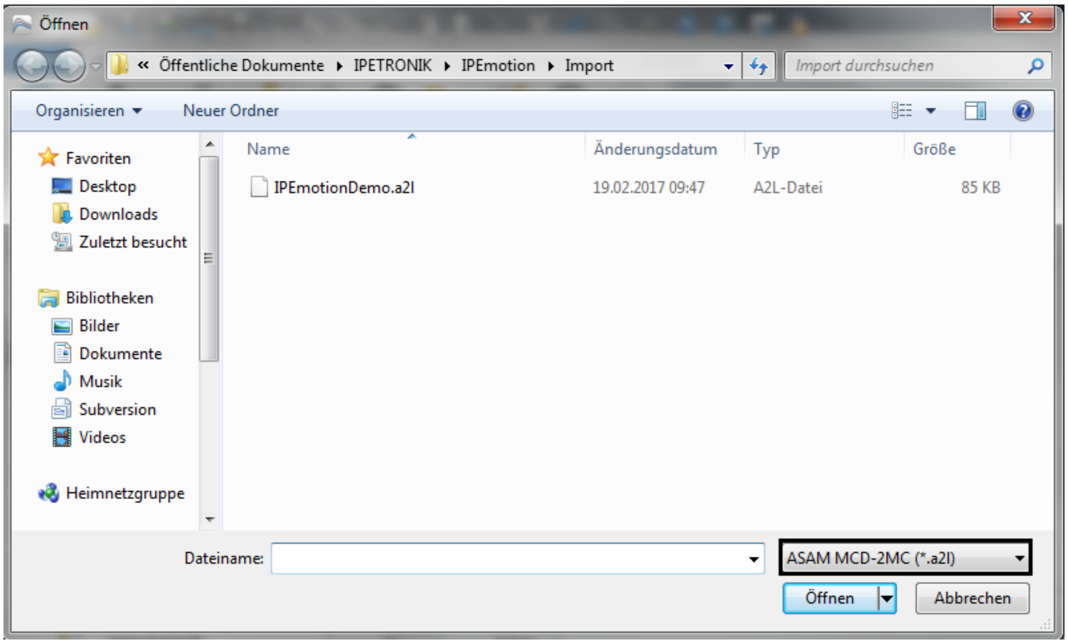
- ATFX (→ 7.4)
- MDF 4.0 (→ 7.5)
- MDF 4.1 (→ 7.6)

5.5.2 Importing A2L files (XCPonUDP)

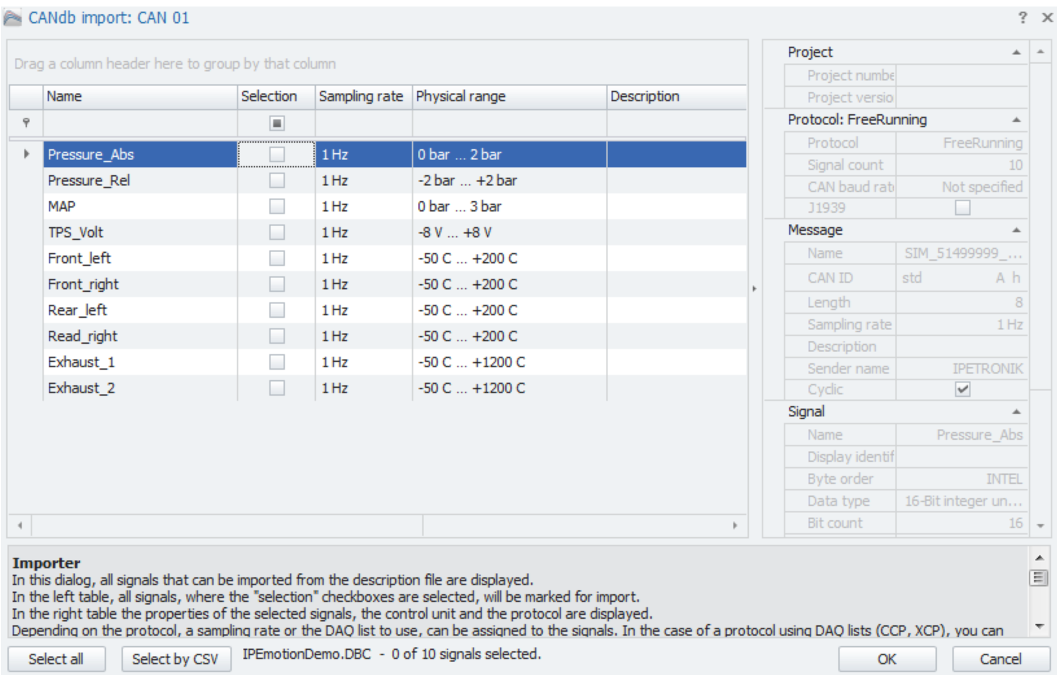
In order to import an A2L database, select either the “Front” channel or any manually added “ETH xx” channel of your “ETH interface”, click the “Import” button in the Ribbon and then choose “A2L”.



The following window lets you choose which file you wish to import. Choose the file you wish to import and click “**Open**”.



Next the "Importer" window will appear, that will present you with a range of importing options.



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. You can either choose manually, which signals to import, you can use the "Select/Deselect all" button on the bottom left, or you can use a CSV-file to determine which signals are to be imported, by clicking "Select by CSV" on the bottom left.

In the right table the metadata, properties of the selected signals, the control unit and the protocol are displayed.

Once you have choosen all the signals you wish to import, click "OK" to complete the import procedure.

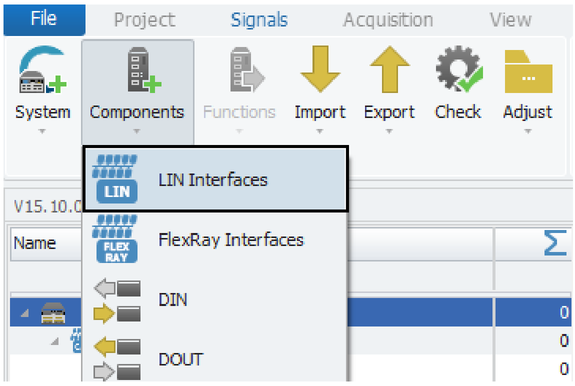
Depending on the protocol, a sampling rate or the DAQ list to use, can be assigned to the signals.
In the case of a protocol using DAQ lists (CCP, XCP), you can specify via the column selection dialog, if the signals are configured by the sampling rate or a DAQ list. To achieve this you should open the column selection dialog, via the context menu of the table header, and then drag the desired column ("sampling" or "DAQ list") from the column selection dialog to the table header. The other column is removed automatically.
If in a protocol based on DAQ lists, sampling rates are used for the signal configuration, during import the signals are assigned to the available DAQ list with the most suitable sampling rate.
In case of protocols supporting array signals, you can specify via the "split array" column whether all the signals of the array or just the first to be imported. If this column does not appear it can be moved from the column selection dialog into the table.

5.5.3 Signal properties

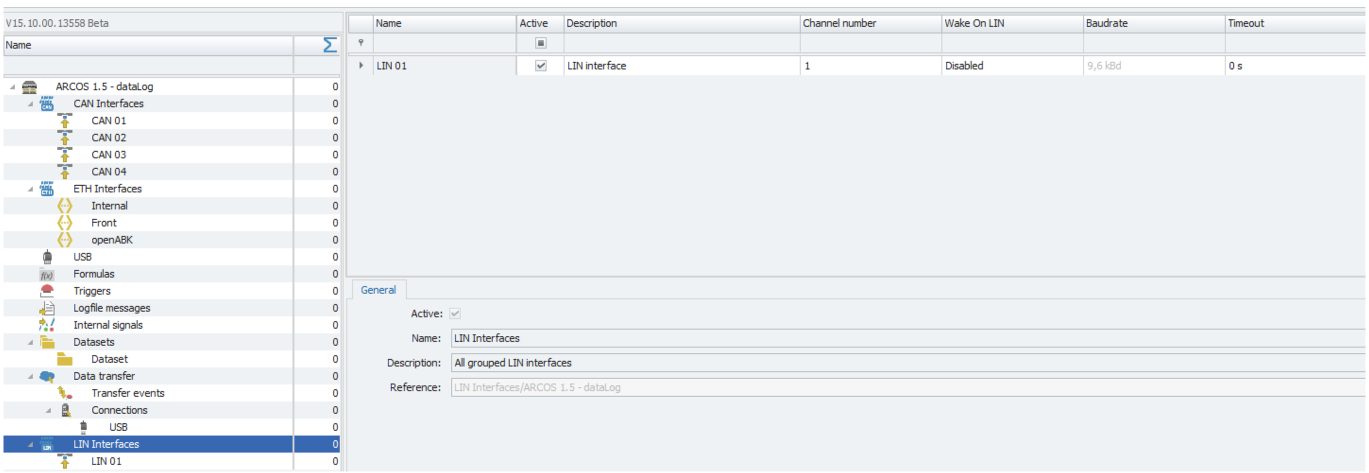
The signal properties for XCPonUDP signals are the same as for XCP signals on CAN channels. Please refer to (→ 5.3.4).

5.6 LIN channels

To work with “LIN channels”, you will first have to add the tree element “LIN interfaces” to your measurement task tree. To do so, select your system in the measurement task tree, click the “Components” button in the ribbon and choose “LIN interfaces” from the resulting dropdown menu.



Once the “LIN interfaces” component has been added it will appear in the measurement task tree as a tree element with one “LIN channel” as a child element.

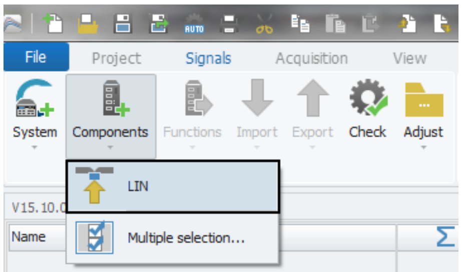


5.6.1 Storage method

In order to store all incoming traffic on a LIN channel use a bus tracing method for storage. Please refer to (→ 7.7).

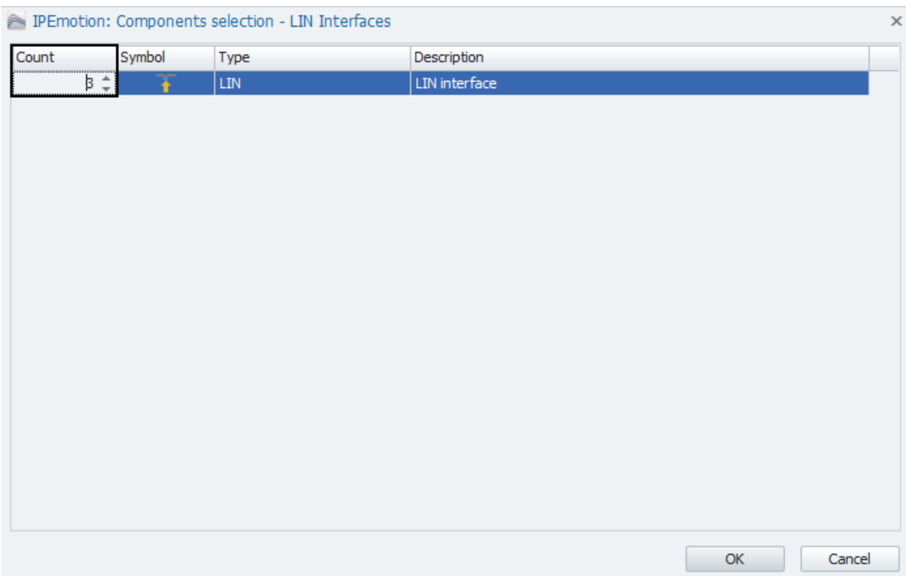
5.6.2 Adding LIN channels

LIN channels can be added by selecting the tree element "**LIN interfaces**", then clicking the "**Components**" button and then choosing "LIN channel".



LIN
Adds a LIN channel that corresponds to a physical LIN channel of your logger. For instructions on LIN settings refer to (→5.6.3).

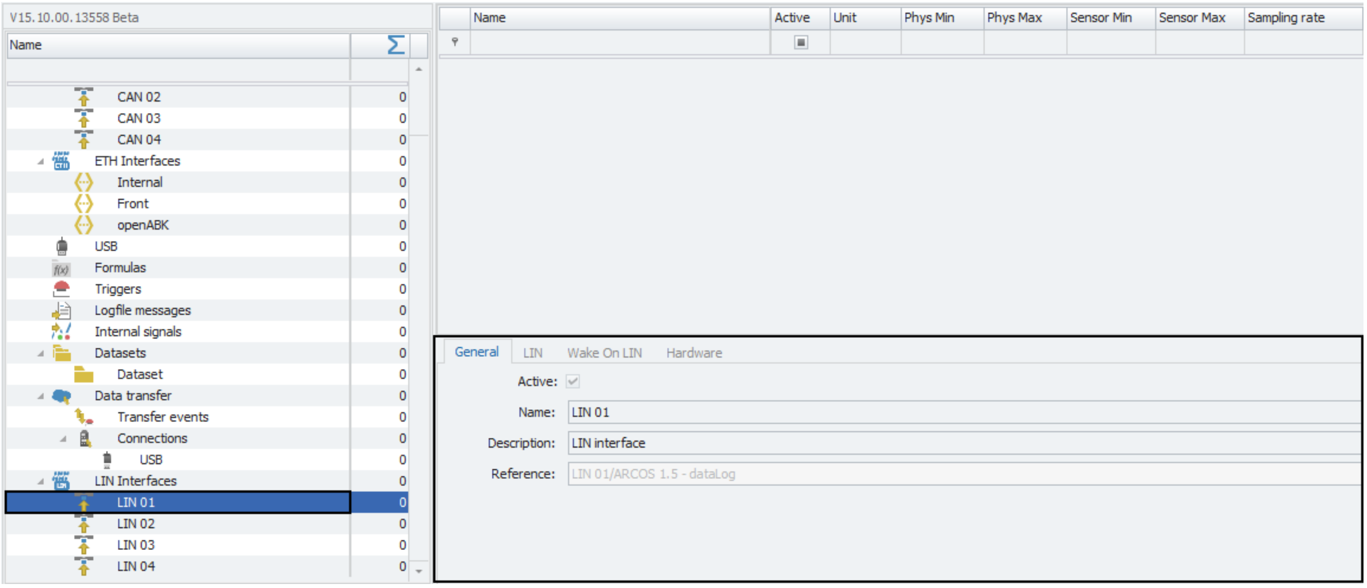
Multiple selection
Allows you to add multiple LIN channels of both types at the same time. To do so set the counter for each type to the desired number of channels that you wish to add as marked in the figure below.



5.6.3 LIN settings

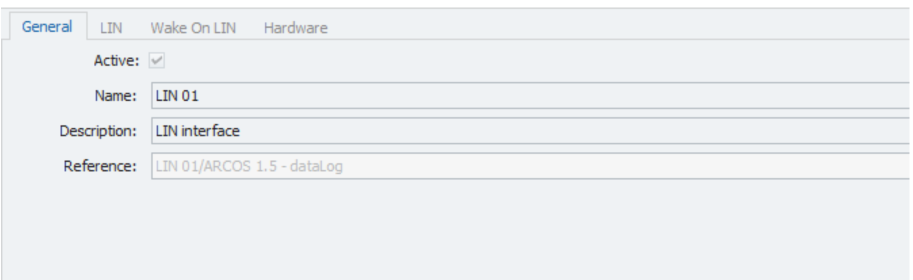
By selecting one of the LIN channels in the tree you will be able to define this channel's settings in the details area.

The same settings described in this section as part of the Details area can also be adjusted when selecting the tree element **LIN interfaces** and then directly changing the desired setting in the respective field of the **Grid area**.

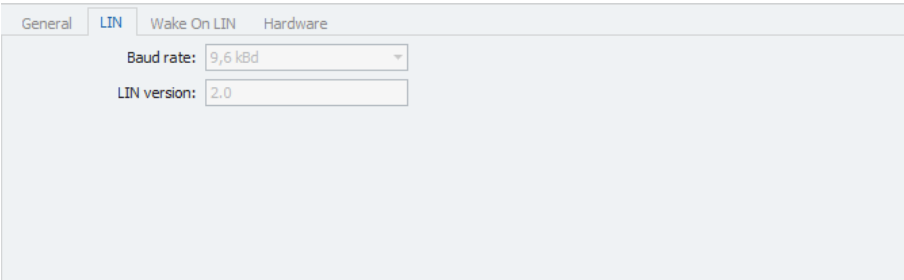


5.6.3.1 General

This tab allows you to give a user specific name for the selected LIN channel if wished and add an additional description. The Reference field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed. The **Active** checkbox is by default marked as activated and cannot be changed.



5.6.3.2 LIN



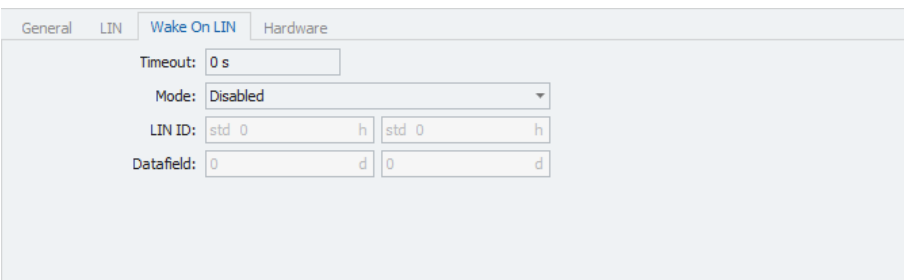
Baud rate

The Baud rate is determined automatically by the logger.

LIN version

This field shows you the version of the LIN protocol that is used as described in the "Description file". This field cannot be changed.

5.6.3.3 Wake On LIN



Timeout

For Wake On LIN, timeout has a special significance. It defines how long a waking channel must be inactive to be recognized so and therefore allow for the logger to shutdown. If timeout is recognized, an entry is made in the log file and an error message with an alert appears on the display, which has to be acknowledged.

Mode

This dropdown menu allows you to set the wake-up function for your selected LIN channel.

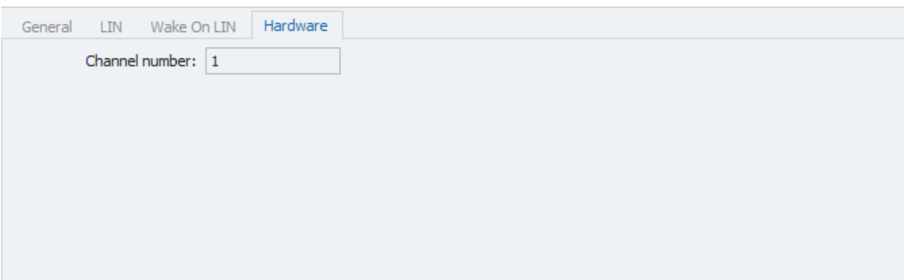
Wake on LIN type	Characteristics
Disabled	No start on LIN messages, lowest energy consumption.
Enabled	Start on a LIN message, with first messages lost; low energy consumption.
Enabled (no message lost)	Start on LIN message, with no message lost; slightly higher idle current.
Keep awake	The logger starts to other awakenings, but only shuts down if all the awakening-conditions are no longer fullfilled and if the keep awake condition is no longer fullfilled.

LIN ID - Settings for starting on a specific LIN ID

This funtionality is the same for CAN and LIN interfaces and has been described in great detail earlier. Please refer to (→5.1.3.3).

5.6.3.4 Hardware (Channel number)

This tab allows you to set a **Channel number** for the selected LIN channel. This channel number has to be unique within the LIN interface.



For better orientation and in order to avoid confusion regarding Channelnumbers and -names, a Channels physical number can be found in the logger's **"Web Interface"** and set accordingly.

5.6.4 LIN channel Bus statistic

The "Bus statistic" provides a range of statistics and status signals for the respective LIN channel. It contains information on the current state of the Bus, the Busload, as well as on the messages that have been received and errors that ocurred.

This functionality is the same for CAN and LIN interfaces and has been described in great detail earlier. Please refer to (→5.1.5).

5.7 LIN signals

5.7.1 Storage method

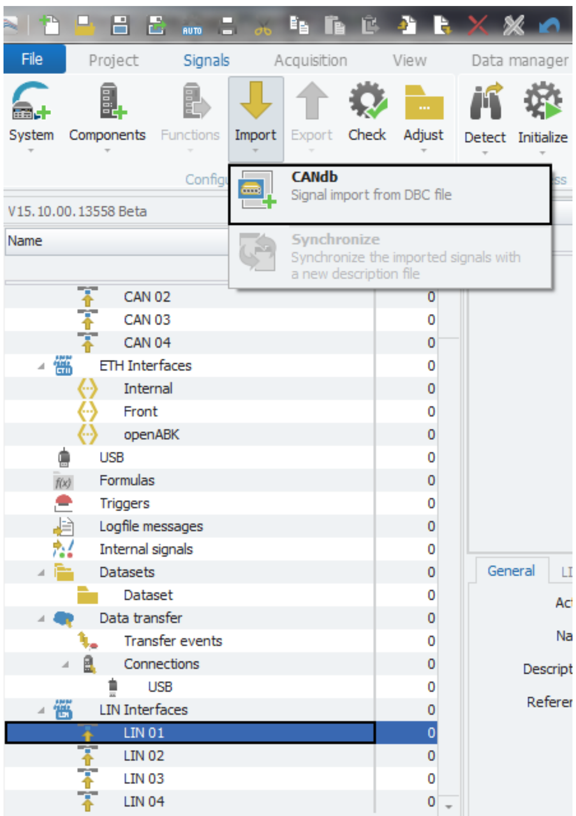
In order to store incoming signals on a LIN channel use one of the following signal storage methods.

- ATFX (→ 7.4)
- MDF 4.0 (→ 7.5)
- MDF 4.1 (→ 7.6)

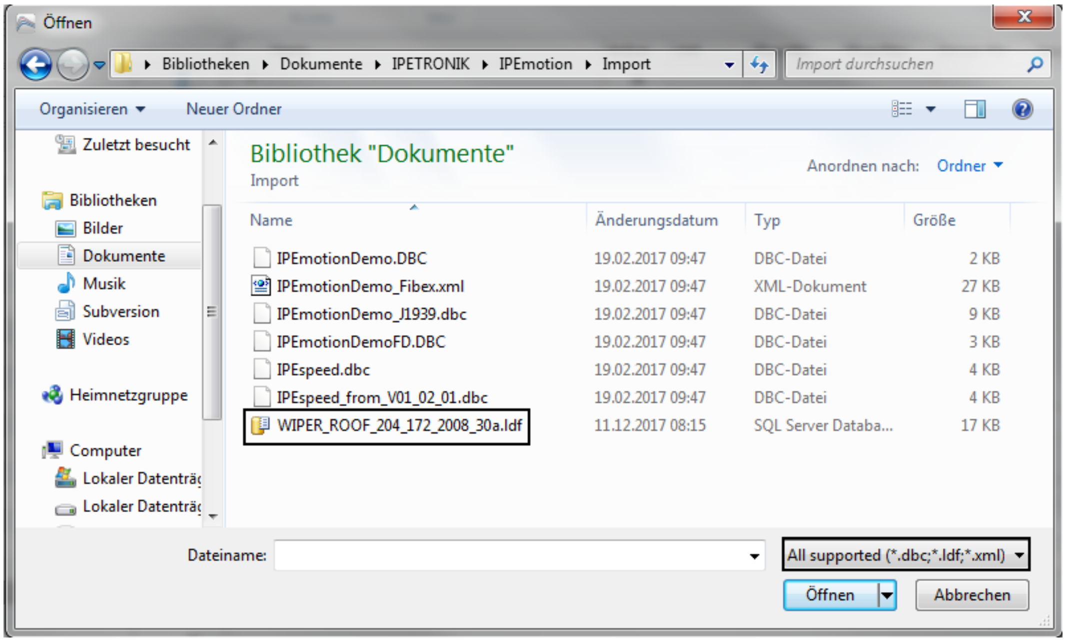
5.7.2 Importing LIN signals

This section explains how to import LIN signals. The filetype to be used for importing LIN signals is a **LINdb (LDF file).**

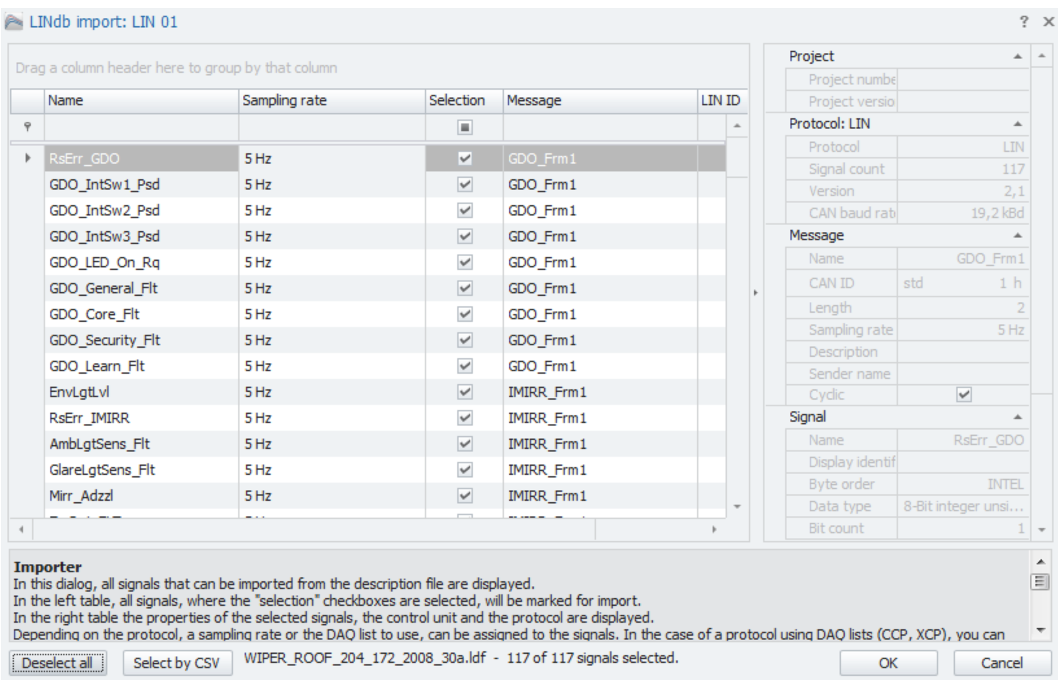
To import Signals, select the LIN channel to which you wish to import your signal in the tree, click the **"Import"** button in the ribbon and then choose **"CANdb"** as description file for the import. For more information on the "description file" refer to (→5.2.4.1).



The following window lets you choose which file you wish to import. The dropdown menu on the bottom right of the window shows you, which filetypes are available. Choose the file you wish to import and click "**Open**".



Once you have opened your file, the "**Importer**" window will appear, that will present you with a range of importing options.



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. You can either choose manually, which signals to import, you can use the "**Select/Deselect all**" button on the bottom left, or you can use a CSV-file to determine which signals are to be imported, by clicking "**Select by CSV**" on the bottom left.

In the right table the properties of the selected signals, the control unit and the protocol are displayed.

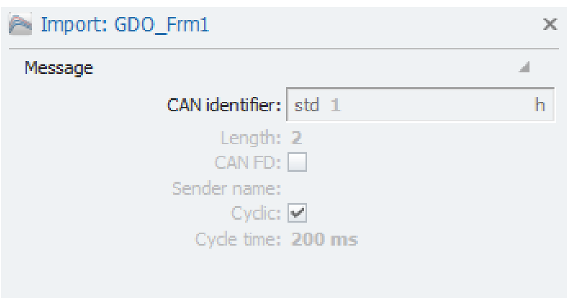
Once you have choosen all the signals you wish to import, click "OK" to complete the import procedure.

> Multiple description files can be imported into the same LIN channel.

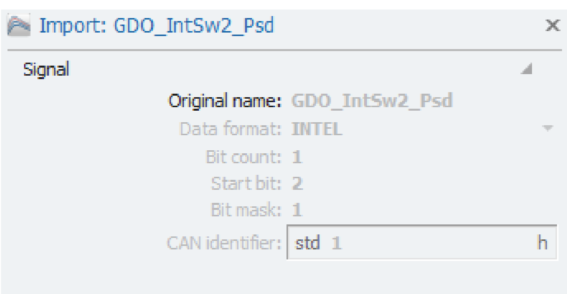
5.7.3 Import properties

The "Import properties" of a description file, Message or signal allow you to see certain properties such as the Data format, the LIN identifier, the Bit mask, the start bit, bit count and more. It shows the signal's properties as described by the description file.

Example for "Import properties" of a LIN description file

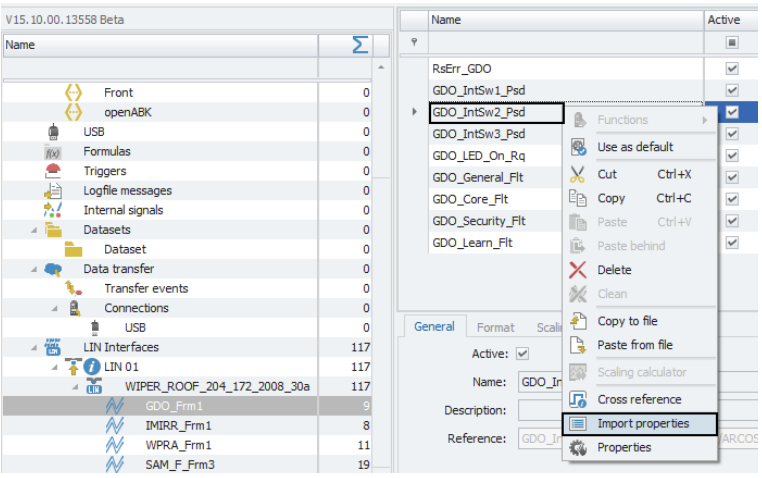


Example for "Import properties" of a LIN Message



Example for "Import properties" of a LIN signal

To access the "Import properties" rightclick on any desired description file, Message or signal and then choose "Import properties" from the resulting context menu.



5.7.4 Signal properties

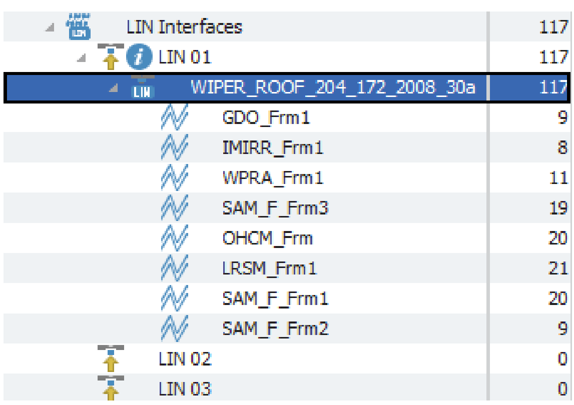
5.7.4.1 Tree elements for LIN signals

After having successfully imported the desired signals to your LIN channel, this channel will contain two new layers of child elements in the measurement task tree: The "Description file" and the "Message".

Description file

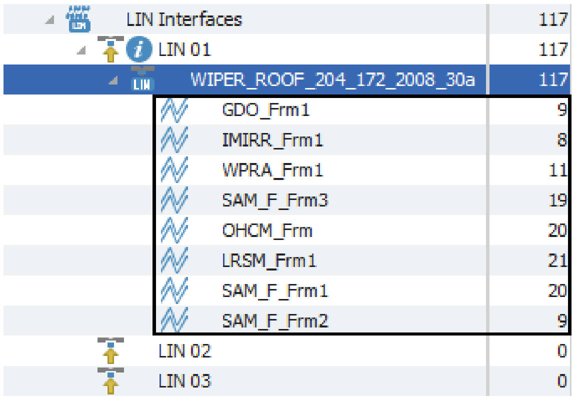
The "description file" is a database file which contains signal information and can be used to import those signals into a Signal channel in IPEmotion. The filetypes which are supported by the CAETEC Plugin for IPEmotion depend on the type of signal you wish to import.

The symbol in the left part of the tree element shows you the type of "description file" you imported (in this case a "LDF" file), then follows the name of the imported "description file" (in this case "WIPER_ROOF_...") and on the right is a number indicating how many signals the "description file" contains (in this case 117).



Message

Each "description file" can contain one or more "Messages", which then contain the actual signals. A "Message" can be found in the "Measurement task tree" as a child element of the "description file", it belongs to.
Each "Message" can, again, contain one or more signals, which is indicated by the number on the right of the "Message's" name.

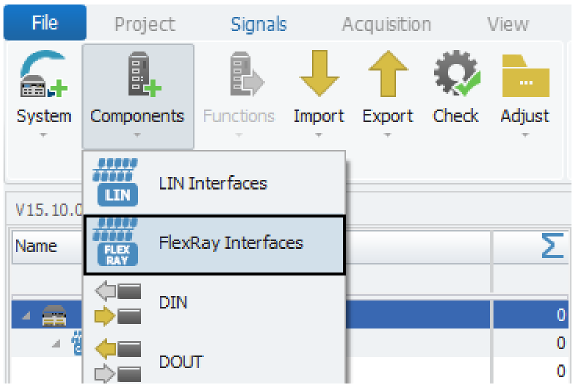


5.7.4.2 Grid area for LIN signals

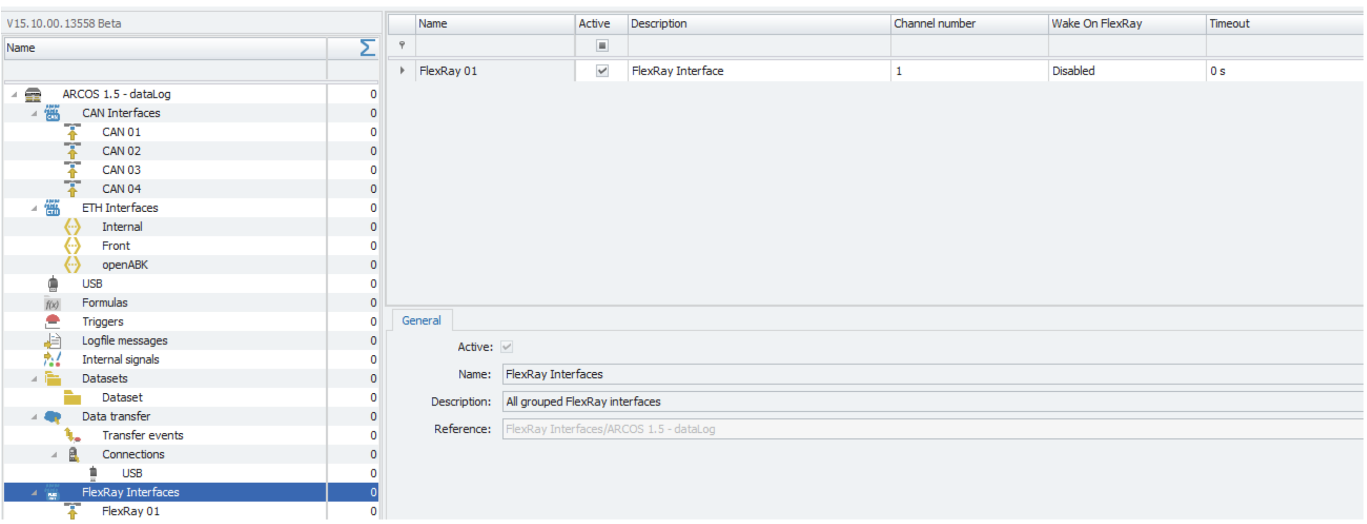
In the "grid area" you will be presented with an overview of your selected LIN channel's signals. Also you can find here two important functions, which are the **"Column chooser"** (→4.3.1) and the **"Filter editor"** (→4.3.2).

5.8 FlexRay channels

To work with "FlexRay channels", you will first have to add the tree element "FlexRay interfaces" to your measurement task tree. To do so, select your system in the measurement task tree, click the "Components" button in the ribbon and choose "FlexRay interfaces" from the resulting dropdown menu.



Once the "FlexRay interfaces" component has been added it will appear in the measurement task tree as a tree element with one "FlexRay channel" as a child element.



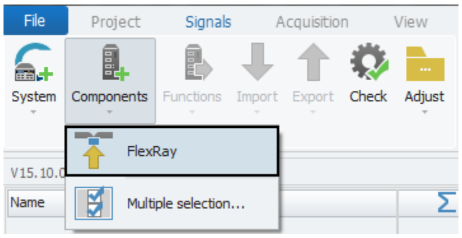
5.8.1 Storage method

In order to store all incoming traffic on a FlexRay channel use a bus tracing method for storage. Please refer to (→ 7.7).

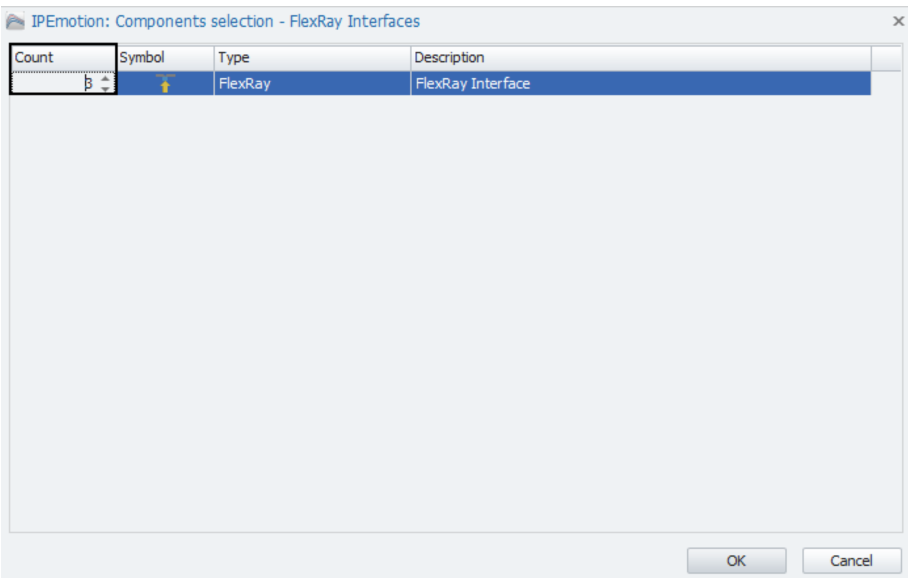
5.8.2 Adding FlexRay channels

FlexRay channels can be added by selecting the tree element **"FlexRay interfaces"**, then clicking the **"Components"** button and then choosing "FlexRay channel".

FlexRay
Adds a FlexRay channel that corresponds to a physical FlexRay channel of your logger. For instructions on FlexRay settings refer to (→5.8.3).



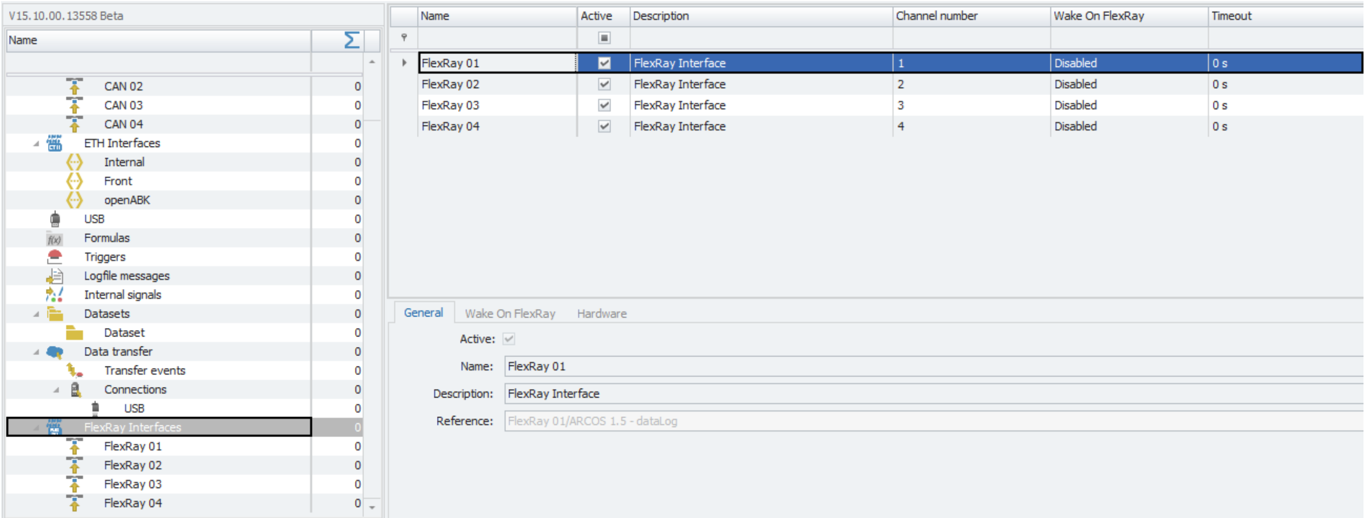
Multiple selection
Allows you to add multiple FlexRay channels of both types at the same time. To do so set the counter for each type to the desired number of channels that you wish to add as marked in the figure below.



5.8.3 FlexRay settings

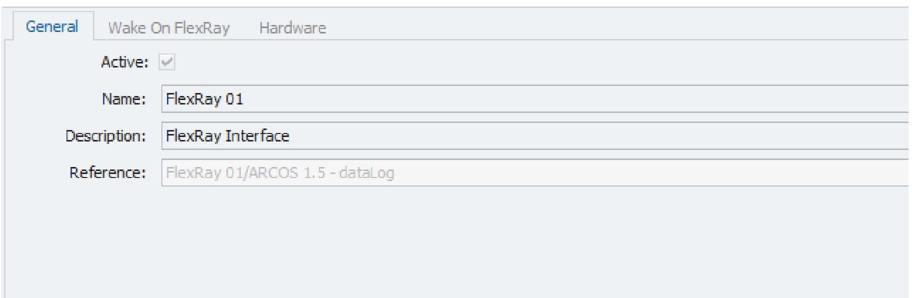
By selecting the "FlexRay Interfaces" in the tree and the choosing one of the FlexRay channels in the grid area you will be able to define this channel's settings in the details area.

> The same settings described in this section as part of the Details area can also be adjusted when selecting the desired channel in the grid area and then directly changing the desired setting in the respective field of the Grid area.

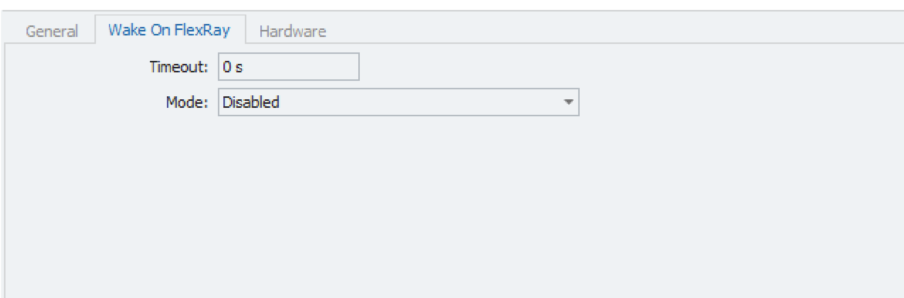


5.8.3.1 General

This tab allows you to give a user specific name for the selected FlexRay channel if wished and add an additional description. The Reference field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed. The Active checkbox is by default marked as activated and cannot be changed.



5.8.3.2 Wake On FlexRay



Timeout

Timeout defines how long a waking channel must be inactive to be recognized so and therefore allow for the logger to shutdown. If timeout is recognized, an entry is made in the log file and an error message with an alert appears on the display, which has to be acknowledged.

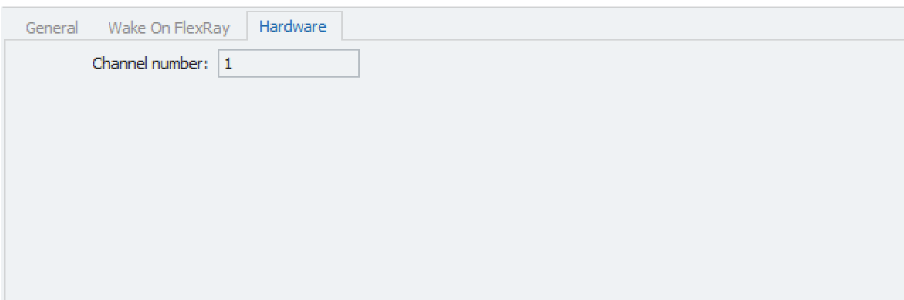
Mode

This dropdown menu allows you to set the wake-up function for your selected FlexRay channel.

Wake on FlexRay type	Characteristics
Disabled	No start on FlexRay messages, lowest energy consumption.
Enabled	Start on a FlexRay message, with first messages lost; low energy consumption.
Keep awake	The logger starts to other awakenings, but only shuts down if all the awakening-conditions are no longer fullfilled and if the keep awake condition is no longer fullfilled.

5.8.3.3 Hardware (Channel number)

This tab allows you to set a **Channel number** for the selected FlexRay channel. This channel number has to be unique within the FlexRay interface.



For better orientation and in order to avoid confusion regarding Channelnumbers and -names, a Channels physical number can be found in the logger's **"Web Interface"** and set accordingly.

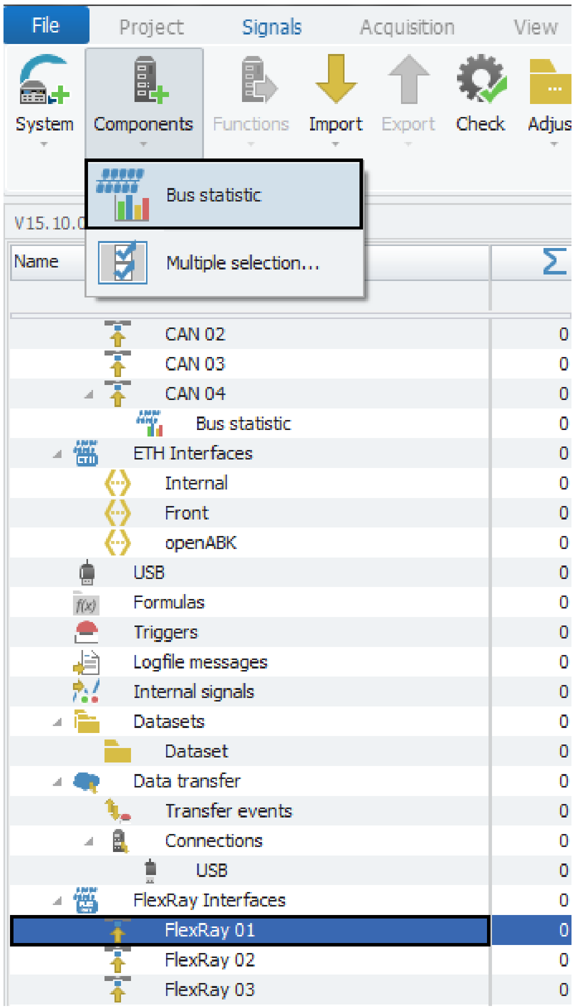
5.8.4 FlexRay channel Bus statistic

The "Bus statistic" for FlexRay provides only one entry on the current status of the FlexRay bus.

The "Bus statistic" only shows statistics for the channel to which it belongs. For each channel that you would like to see the statistic, you will have to add the component "Bus statistic".

5.8.4.1 Adding Bus statistics

Select the channel in the tree for which you would like to add "Bus statistic", then click the "Components" button in the Ribbon and choose "Bus statistic".



5.8.4.2 Bus statistic signals

Once the component "Bus statistic" has been added to your channel, it will appear in the measurement task tree as a child element of this channel and the grid area will give you an overview of the available signals.

The signals included in "Bus statistics" are of the type "Internal signal" and may be adjusted in the same way. For more information on "Internal signals" please refer to (→5.14).

Overview of signals

Subtype	Meaning	Unit
Controller state	nan= Channel not available 1= Bus on 2= Bus warning 3= Bus off	-

5.9 FlexRay signals

5.9.1 Storage method

In order to store incoming signals on a FlexRay channel use one of the following signal storage methods.

- ATFX (→ 7.4)
- MDF 4.0 (→ 7.5)
- MDF 4.1 (→ 7.6)

5.9.2 Importing FlexRay signals

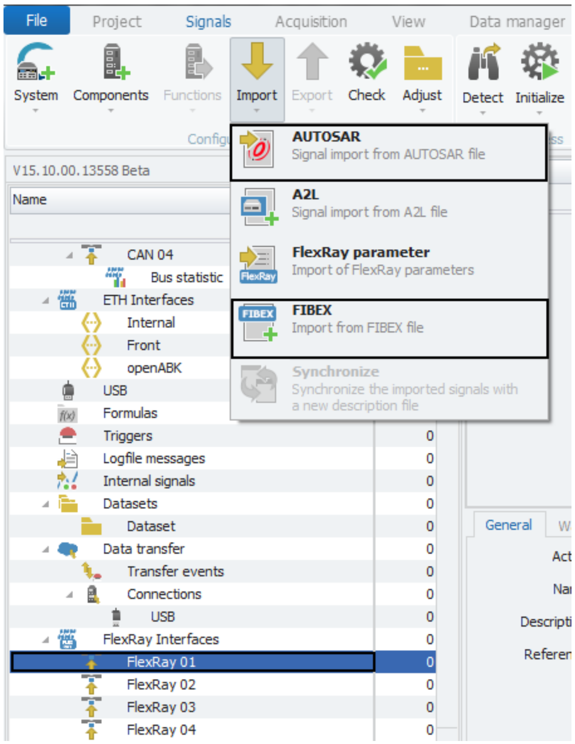
This section explains how to import FlexRay signals. There are three different filetypes which can be used in order to import a single FlexRay signal or a group of FlexRay signals:

- Autosar files
- A2L files
- Fibex files

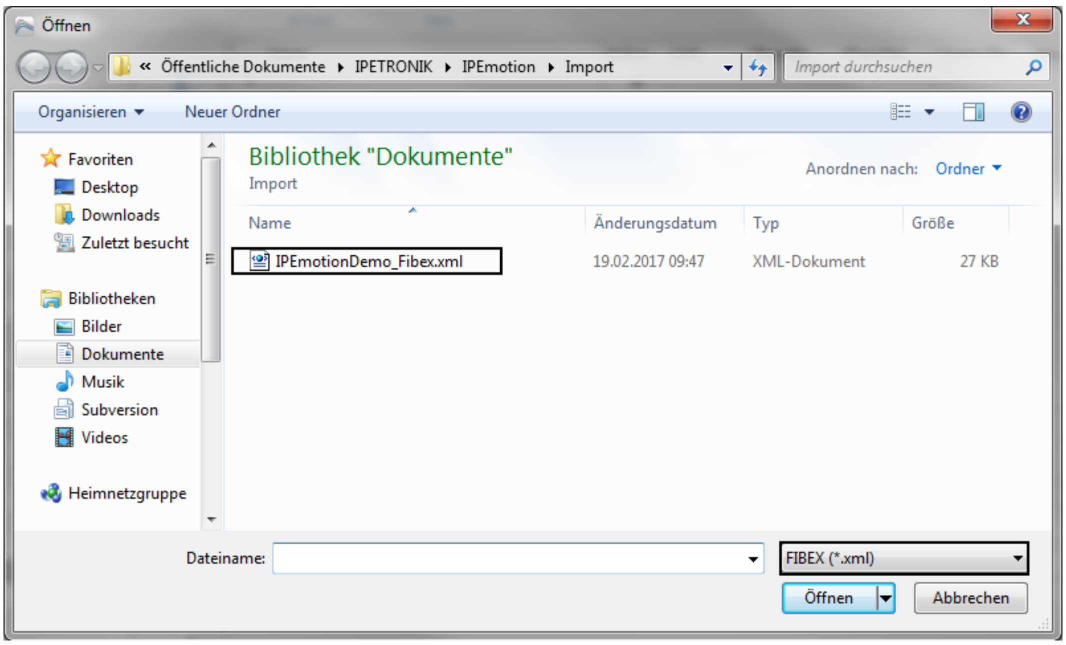
5.9.2.1 Importing Autosar and Fibex files

The procedure for both filtetypes is the same and will be exemplary explained in the following via the "Fibex" import.

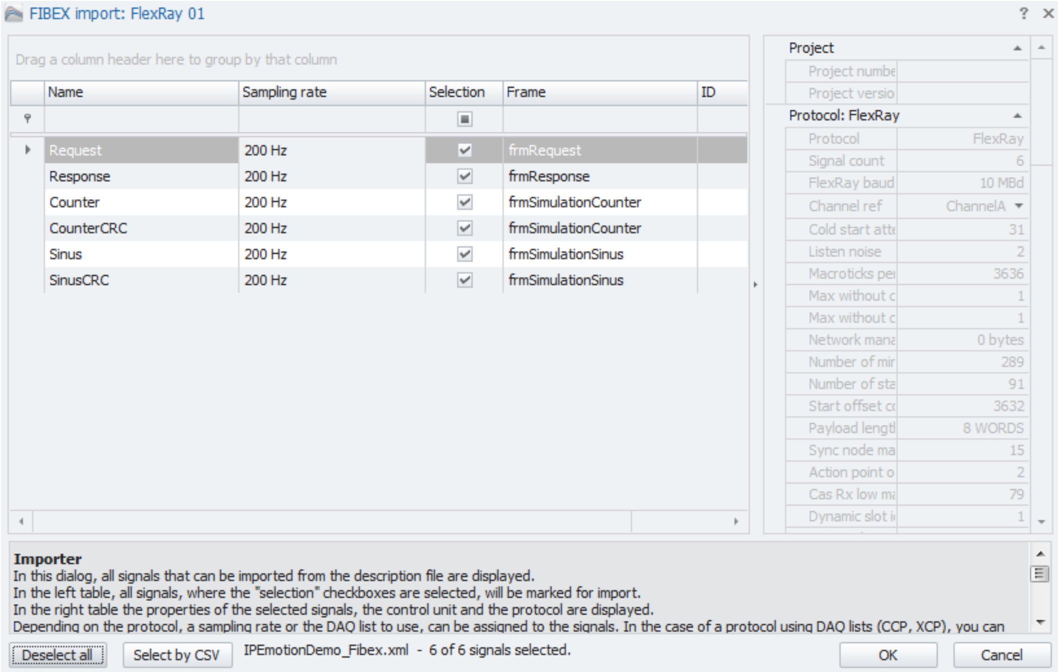
To import Signals, select the FlexRay channel to which you wish to import your signal in the tree, click the **"Import"** button in the ribbon and then choose **"Autosar"** or **"Fibex"** for the import. For more information on the "description file" refer to (→5.2.4.1).



The following window lets you choose which file you wish to import. The dropdown menu on the bottom right of the window shows you, which filetypes are available. Choose the file you wish to import and click "**Open**".



Once you have opened your file, the "**Importer**" window will appear, that will present you with a range of importing options.
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. You can either choose manually, which signals to import, you can use the "**Select/Deselect all**" button on the bottom left, or you can use a CSV-file to determine which signals are to be imported, by clicking "**Select by CSV**" on the bottom left.



In the right table the properties of the selected signals, the control unit and the protocol are displayed.

Once you have choosen all the signals you wish to import, click "OK" to complete the import procedure.

> Multiple description files can be imported into the same FlexRay channel.

5.9.2.2 Importing A2L files (XCP on FlexRay)

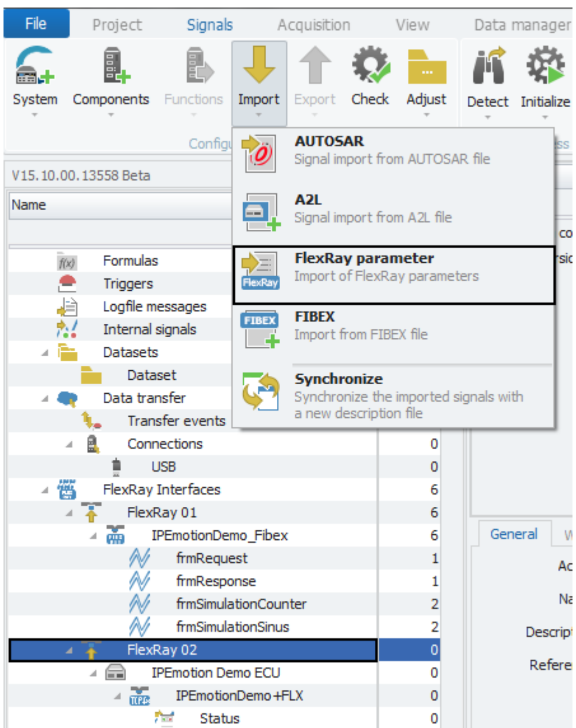
The Import of CCP/XCP databases via A2L files for FlexRay follows the same procedure as the CCP/XCP import via A2L file for CAN. Please refer to (→5.3.2).

> When importing signals from a CCP/XCP database via A2L file, you will need to additionally import the flexray parameters from a Fibex file as described below.

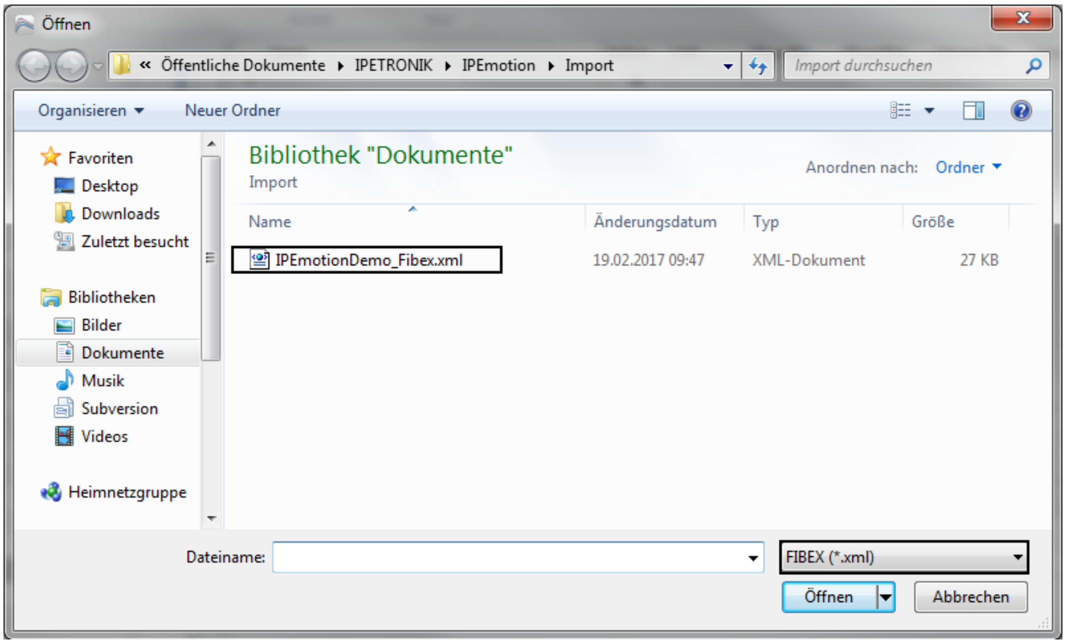
Importing FlexRay parameters

When creating a FlexRay XCP measurement task via an A2L file, it is necessary to import the FlexRay parameters via a Fibex file once the A2L file has been imported. Otherwise, the communication between ECU and FlexRay bus cannot be established.

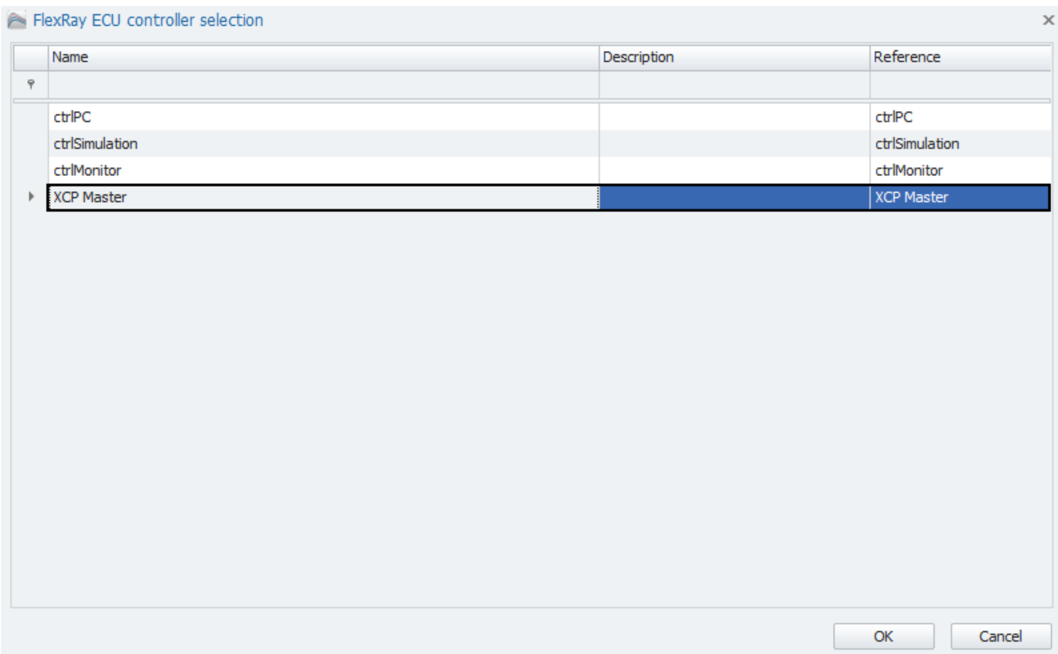
To import FlexRay parameters, select the FlexRay channel to which you wish to import your signal in the tree, click the "**Import**" button in the ribbon and then choose "**FlexRay parameters**" for the import.



The following window lets you choose which Fibex file you wish to import. Choose the file you wish to import and click "Open".

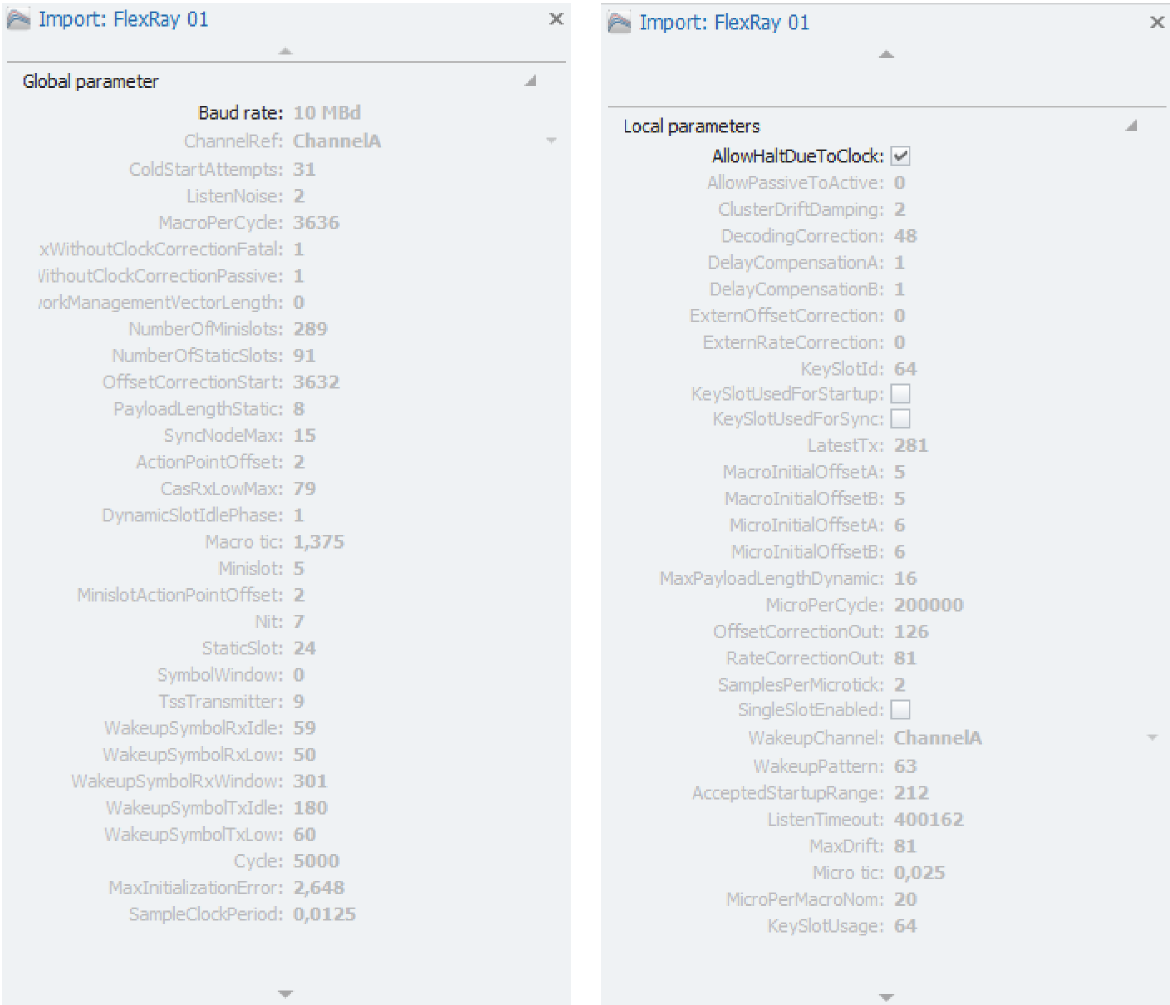


The following window lets you choose the ECU controller with which the FlexRay bus is supposed to communicate. Normally the right controller to choose is called "XCP Master". Select your desired controller and confirm with "OK".

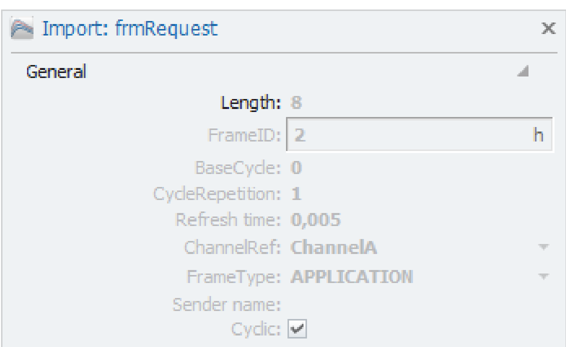


5.9.3 Import properties

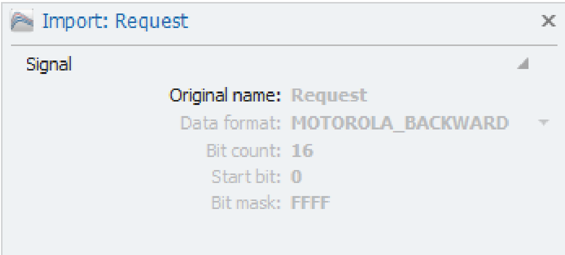
The "Import properties" of a description file, Message or signal allow you to see certain properties such as the Data format, the FlexRay identifier, the Bit mask, the start bit, bit count, optional commands and more. It shows the signal's properties as described by the description file.



Example for "Import properties" of a FlexRay description file

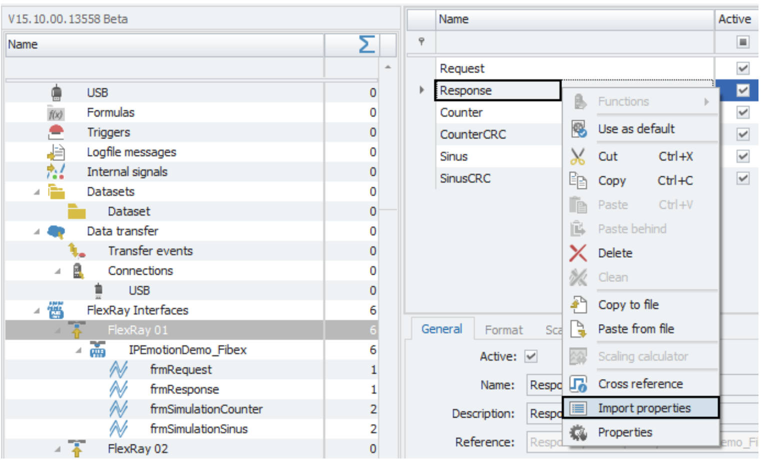


Example for "Import properties" of a FlexRay Message



Example for "Import properties" of a FlexRay signal

To access the "Import properties" rightclick on any desired description file, Message or signal and then choose "Import properties" from the resulting context menu.



5.9.4 Signal properties

5.9.4.1 Tree elements for FlexRay signals

The Tree elements available after import for your FlexRay channel depend on the Method you used for importing the signals.

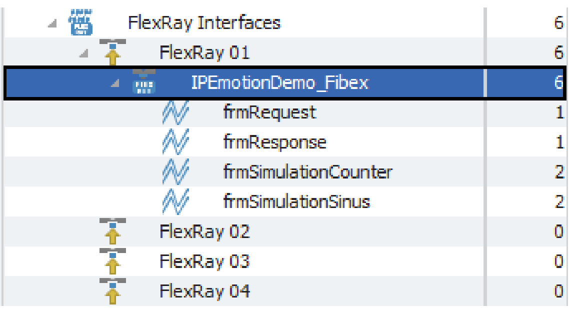
If you imported an A2L file, the resulting tree elements will be equal to a CCP/XCP import on a CAN channel. Please refer to (→5.3.4.1).

If you imported a Fibex file to your FlexRay channel, this channel will contain two new layers of child elements in the measurement task tree: The **"Description file"** and the **"Message"**.

Description file

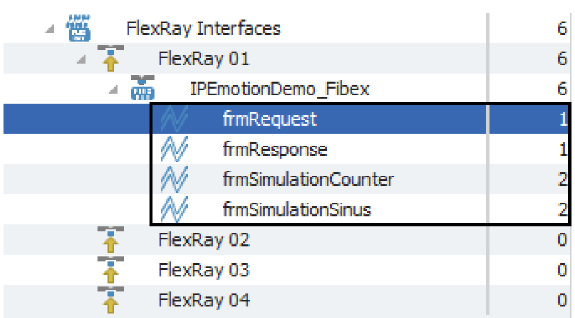
The "description file" is the database file which contains signal information and can be used to import those signals into a Signal channel in IPEmotion.

The tree element shows the "description file's" name (in this case "IPEmotionDemo_Fibex") and, on the right, the number of signals it contains (in this case 6).



Message

Each "description file" can contain one or more "Messages", which then contain the actual signals. A "Message" can be found in the "Measurement task tree" as a child element of the "description file", it belongs to.
Each "Message" can, again, contain one or more signals, which is indicated by the number on the right of the "Message's" name.



5.9.4.2 Grid area for FlexRay signals

In the "grid area" you will be presented with an overview of your selected FlexRay channel's signals. Also you can find here two important functions, which are the "**Column chooser**" (→4.3.1) and the "**Filter editor**" (→4.3.2).

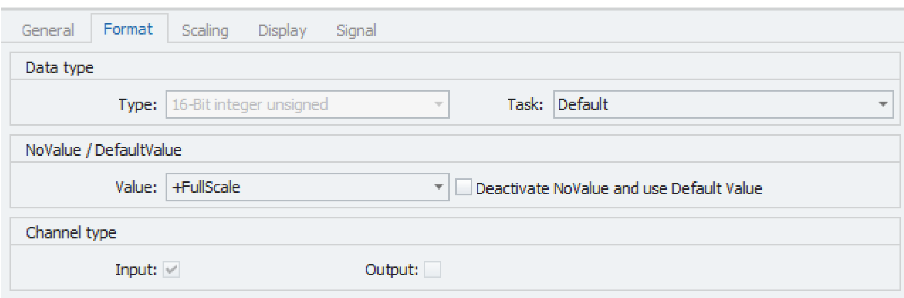
5.9.4.3 Details area for FlexRay signals

The Details area shows settings either for the selected tree element ("description file" or "Message") or the selected signal in the grid area. In case a tree element is selected, the details area will only show the "General" tab. Please refer to (→4.2.2).

In case a signal is selected in the grid area, the details area will contain additional tabs which will be explained in the following.

General
This tab allows you to activate or deactivate the entire signal by ticking/unticking the checkbox, give a user specific name to your signal if wished and add an additional description. The Reference field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed. The "Sampling rate" allows you to set, how frequently a signal should be requested.

Format
This tab contains information and options regarding file format, tasks and Channel type.



- **Data type**
  This field tells you the type of data (in this case "**16-Bit integer unsigned**") and allows you to apply special tasks for this signal such as "GPS Longitude", "GPS Latitude", "UTC hour", "Audio mono" and more.

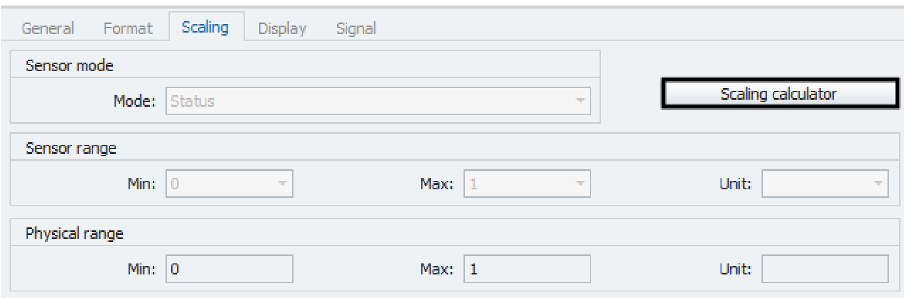
- **NoValue / DefaultValue**
  This field allows you to define the value that will be shown if a signal value is read as invalid.

- **Channel type**
  This field tells you whether you are dealing with a "**Input**" channel or "**Output**" channel.

Scaling

The fields accessible directly through the tab allow for basic scaling operations to convert analog measurement in engineering units. The **"Scaling calculator"** allows for more refined scaling options with a large range of functions. For details on how to use the "Scaling calculator" please refer to the **IPEmotion Documentation - Section 3.4.5 "Channel configuration and scaling"**.

> **!** In some cases it may be necessary to activate editing of protocol channel scaling in order to gain full acces to the scaling functionality. For instructions please refer to the point **"Edit protocol channel scaling"** of the Expert settings (→ 3.2.2).



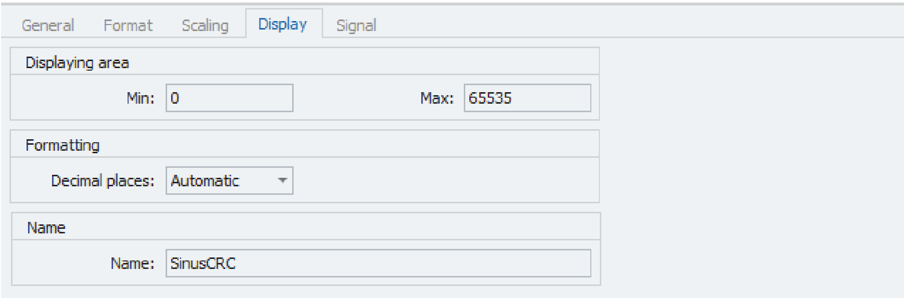
- Sensor Mode
  The sensor mode tells the type of signal. It can be of different types such as "Status", "Voltage", "Frequenzy" or others. It cannot be changed and serves for IPEmotion to know what kind of signal it is dealing with.

- Sensor Range
  Shows the raw value range of the signal.

- Physical Range
  Allows you to set a range to which you would like to "scale" your signal and also define the unit to use. For more refined scaling please use the "Scaling calculator" and refer to the **IPEmotion Documentation - Section 3.4.5 "Channel configuration and scaling"**.

Display

This tab allows you to define what information about the current signal will be shown on a display if one is connected.



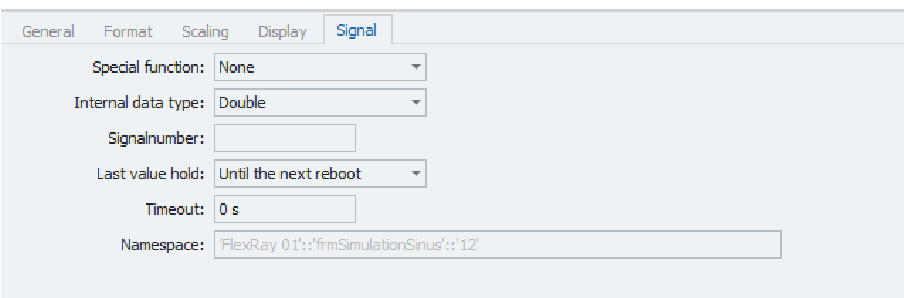
- Displaying area
  Shows the value range which will be shown on a display. It usually should match the "Physical range" from the "Scaling" tab.

- Formatting
  The dropdown menu **"Decimal places"** allows you to set how many decimal numbers of the value will be shown on a display.

- Name
  Allows you to set a Name to be shown on a display.

Signal

This tab allows you to define signal settings.



- Internal data type
  Assign an internal data type to the signal. Available data types are **"Double"** and **"String"**.

- Signal number
  Assign a number to the current signal. This way you will later be able to sort the signals in the grid according to their "Signal numbers".

- Last value hold
  Specify, for how long the last value of the signal will be hold.

- Timeout
  Specify the timeout period for the current signal. If the data source doesn't send data for the specified time period, the value of the signal is set to "NaN (Not a Number)" and will be displayed as "–" in a display.

- Namespace
  The "Namespace" serves as unique identifier for the signal inside the logger.

5.10 GPS Signals

The GPS module sends a constant stream of values to the logger. Its configuration defines which values from this data stream are to be evaluated and made available for further use.

5.10.1 Storage method

In order to store incoming GPS signals you can use the GPX storage method. These signals will then be stored in a separate file only containing your GPS signals.

- GPX (→ 7.11)

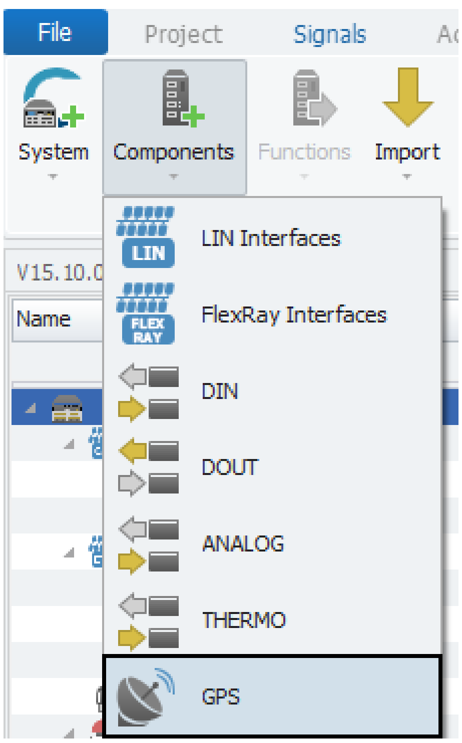
You can also store your GPS signals together with signals from other buses. To do so use one of the following signal storage methods.

- ATFX (→ 7.4)
- MDF 4.0 (→ 7.5)
- MDF 4.1 (→ 7.6)

5.10.2 Adding GPS Signals

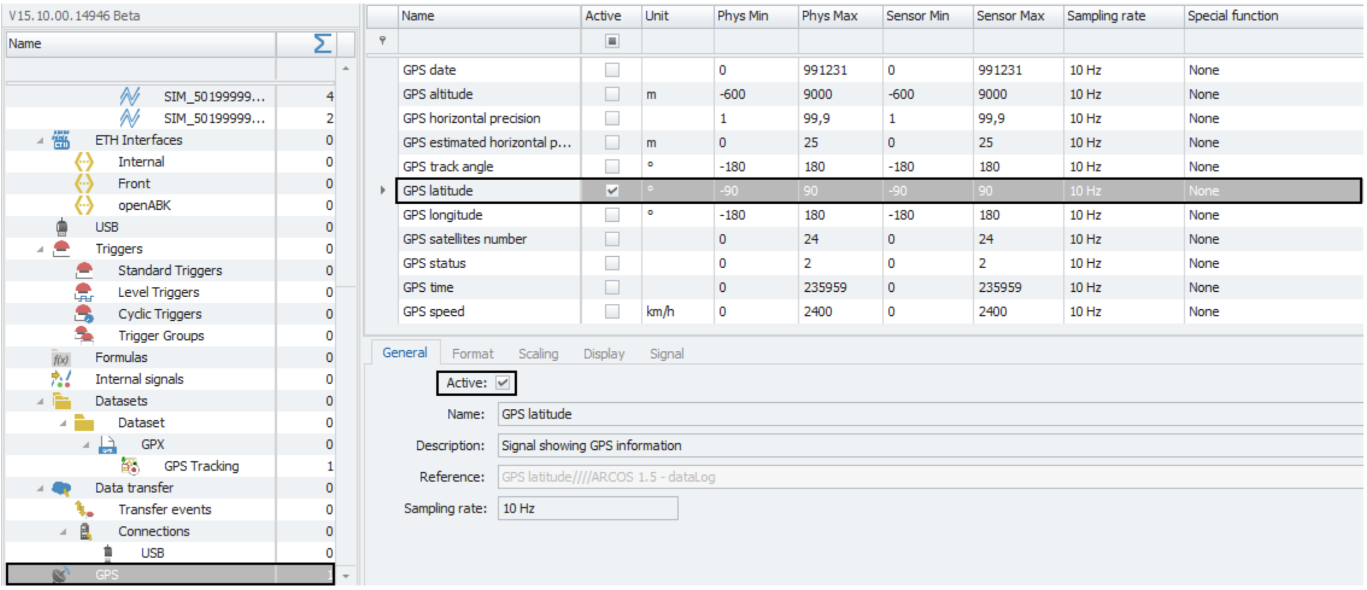
5.10.2.1 CAETEC GPS module

In order to use the CAETEC GPS module, select your system (Arcos 1.x, µcros) in the "Measurement task tree", click the "Components" button in the Ribbon and choose "GPS".

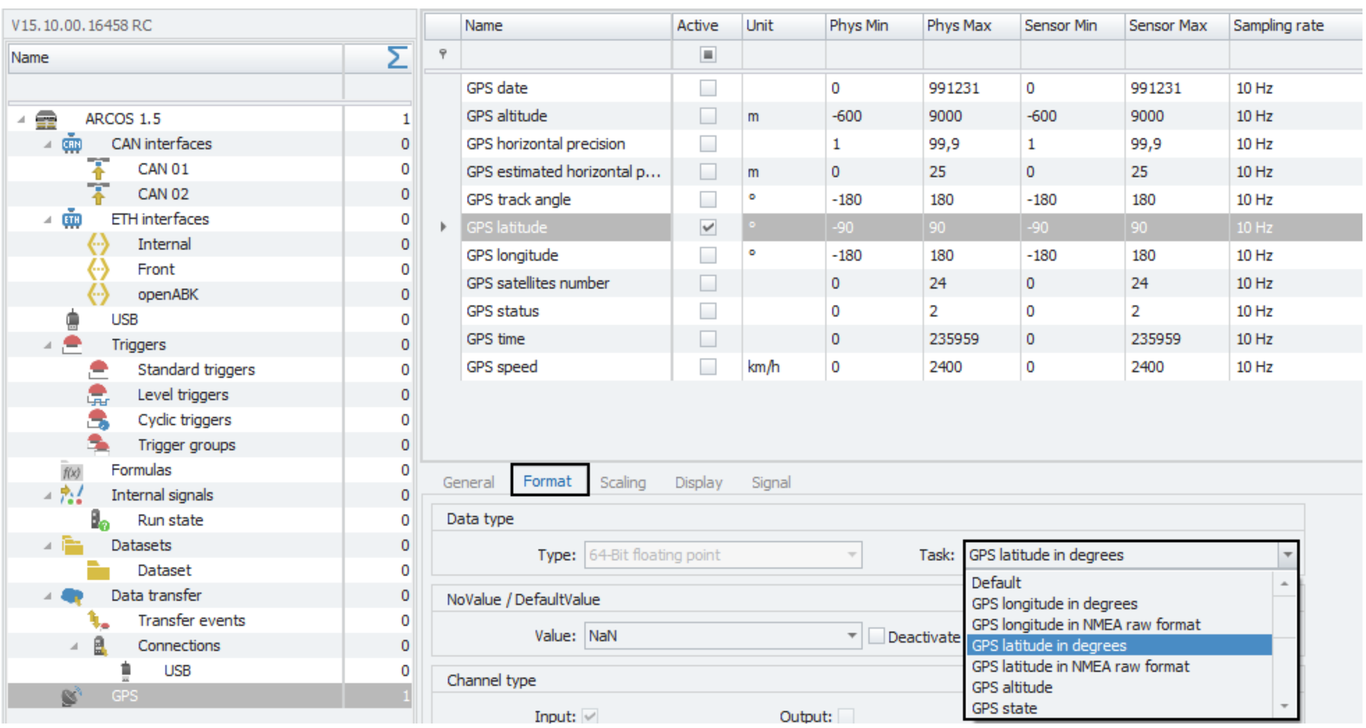


5.10.2.2 Other GPS signals (Assigning GPS signals)

If you are not using the CAETEC GPS module, but are receiving GPS signals on your logger GPS tasks have to be manually assigned to the respective signals in order to store those signals in a GPX file.
To assign a GPS task to a signal, select the desired signal in the "Grid area" and then activate it by ticking the "Active" box in the signal's "Details area".



Then navigate to the "Format" tab in the signal's "Details area" and use the dropdown menu "Tasks" to assign the desired GPS task.



Each GPS task can only be assigned to one signal. That means, if you reassign a previously already assigned GPS task to a new signal, make sure to unassign this task from its previously assigned signal.

Using the **"Check"** function will tell you, whether you have multiply assigned tasks.

5.10.3 Signal properties

5.10.3.1 Tree elements for GPS signals

There is one tree element for GPS signals, called "GPS". In the right table of the "Measurement task tree" a number will indicate how many active signals it contains (in this case 2).

5.10.3.2 Grid area for GPS signals

In the "Grid area" you will be presented with an overview of the availabe GPS signals. Also you can find here two important functions, which are the **"Column chooser"** (→4.3.1) and the **"Filter editor"** (→4.3.2).

Overview of GPS signals

Subtype	Meaning	Unit
GPS date	GPS date UTC yymmdd	-
GPS altitude	Elevation above sea level	(m)
GPS horizontal precision	Horizontal Dilution of Precision	-
GPS estimated horizontal precision	Estimation of horizontal dilution of precision (probability 95%)	(m)
GPS track angle	Inclination of the track	(°)
GPS latitude	Latitude	(°)
GPS longitude	Longitude	(°)
GPS satellites number	Number of received satellites	-
GPS status	0 = no connection 1 = connection 2 = Egnos active	-
GPS time	GPS time UTC hhmmss	-
GPS speed	Current speed	(km/h)

5.10.3.3 Details area for GPS signals

The Details area shows settings either for the tree element **"GPS"** or a selected signal in the grid area. In case the tree element "GPS" is selected, the details area will only show the "General" tab. Please refer to (→4.2.2).

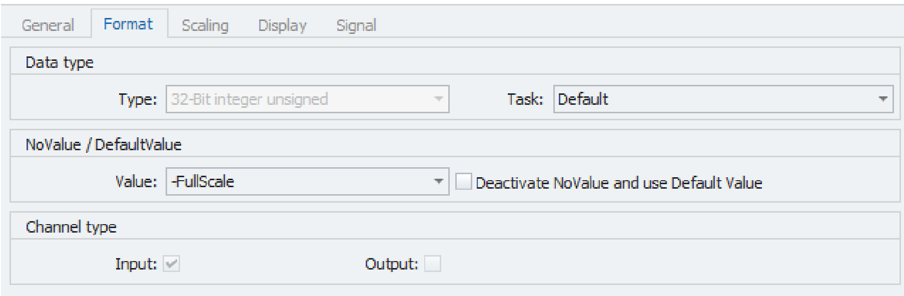
In case a signal is selected in the grid area, the details area will contain additional tabs which will be explained in the following.

General

This tab allows you to activate or deactivate the entire signal by ticking/unticking the checkbox, give a user specific name to your signal if wished and add an additional description. The Reference field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed. The "Sampling rate" allows you to set, how frequently a signal should be requested.

Format

This tab contains information and options regarding file format, tasks and Channel type.



- Data type
  This field tells you the type of data (in this case "**32-Bit integer unsigned**") and allows you to apply special tasks for this signal such as "GPS Longitude", "GPS Latitude", "UTC hour", "Audio mono" and more.

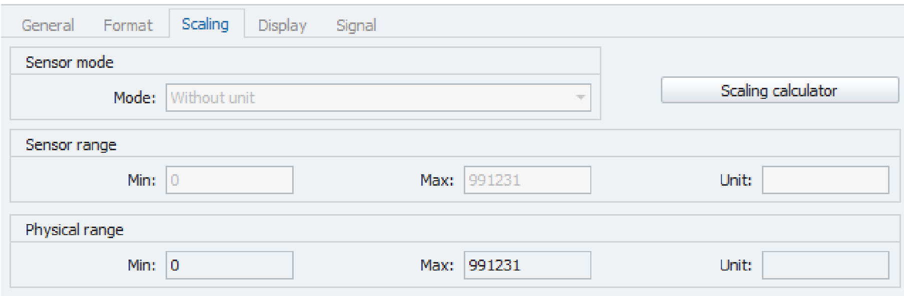
- NoValue / DefaultValue
  This field allows you to define the value that will be shown if a signal value is read as invalid.

- Channel type
  This field tells you whether you are dealing with a "**Input**" channel or "**Output**" channel.

Scaling

The fields accessible directly through the tab allow for basic scaling operations to convert analog measurement in engineering units. The **"Scaling calculator"** allows for more refined scaling options with a large range of functions. For details on how to use the "Scaling calculator" please refer to the **IPEmotion Documentation - Section 3.4.5 "Channel configuration and scaling"**.

> In some cases it may be necessary to activate editing of protocol channel scaling in order to gain full acces to the scaling functionality. For instructions please refer to the point **"Edit protocol channel scaling"** of the Expert settings (→ 3.2.2).



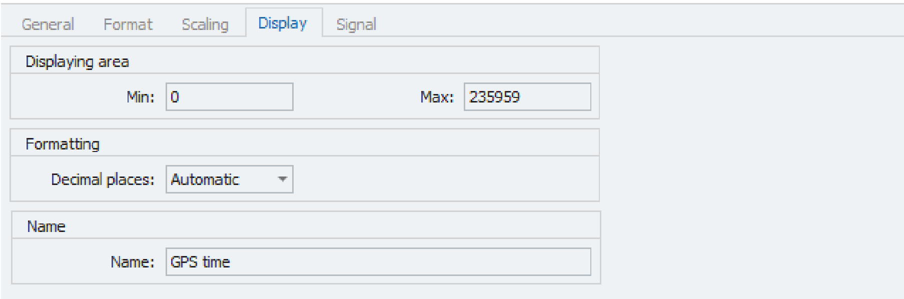
- Sensor Mode
  The sensor mode tells the type of signal. It can be of different types such as "Status", "Voltage", "Frequenzy" or others. It cannot be changed and serves for IPEmotion to know what kind of signal it is dealing with.

- Sensor Range
  Shows the raw value range of the signal.

- Physical Range
  Allows you to set a range to which you would like to "scale" your signal and also define the unit to use. For more refined scaling please use the "Scaling calculator" and refer to the **IPEmotion Documentation - Section 3.4.5 "Channel configuration and scaling"**.

Display

This tab allows you to define what information about the current signal will be shown on a display if one is connected.



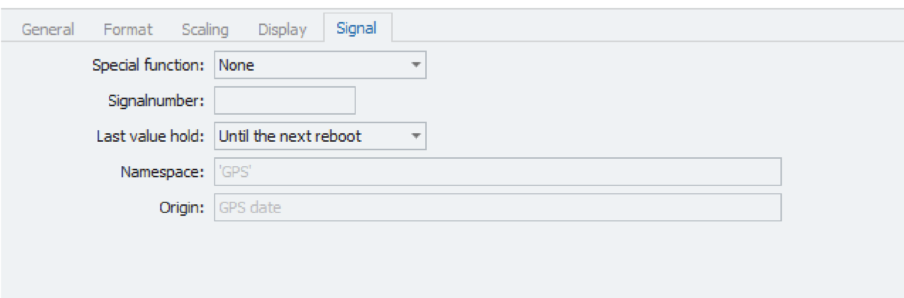
- **Displaying area**
  Shows the value range which will be shown on a display. It usually should match the "Physical range" from the "Scaling" tab.

- **Formatting**
  The dropdown menu **"Decimal places"** allows you to set how many decimal numbers of the value will be shown on a display.

- **Name**
  Allows you to set a Name to be shown on a display.

Signal

This tab allows you to define signal settings.



- **Signal number**
  Assign a number to the current signal. This way you will later be able to sort the signals in the grid according to their "Signal numbers".

- **Last value hold**
  Specify, for how long the last value of the signal will be hold.

- Namespace
  The “Namespace” serves as unique identifier for the signal inside the logger.

- Origin
  Tells you what is the original signal.

5.11 Video devices

In order to work with video signals, you will first need to add a video device to your system. There are three different video devices, that you can work with:

- Video Interface (→5.11.2)
- USB camera (→5.11.3)
- Ethernet camera (→5.11.4)

5.11.1 Storage method

In order to store an incoming signal on a video device use "AVI" as storage method. Please refer to (→ 7.9).

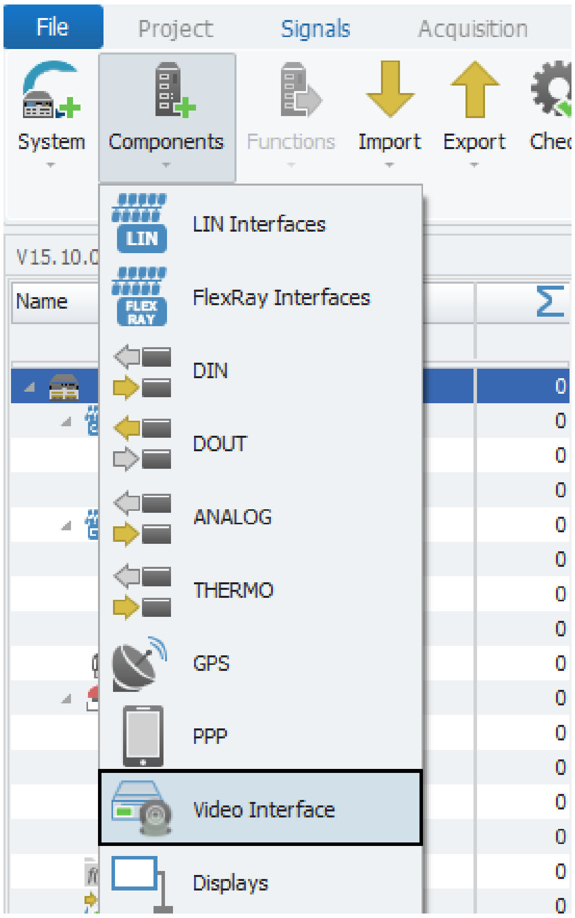
5.11.2 Video Interface

The "Video Interface" provides a set of four analog cameras, which are connected to your logger. It provides five signals, one for each camera and a combination of all four camera signals called "Quad camera". The "Quad camera" provides a single video signal which contains all four original signals at a quarter of their original resolution, and presents a 4 in 1 picture.
For communication between the cameras and the interface it needs to have its own subnet, in which only the "Video Interface" operates.

> The "Video Interface" is a depricated product and is not bein sold anymore.

5.11.2.1 Adding the Video Interface

The “Video Interface” can be added by selecting the system in the “Measurement task tree”, then clicking the **“Components”** button and finally choosing “Video Interface”



5.11.2.2 Tree elements for the Video Interface

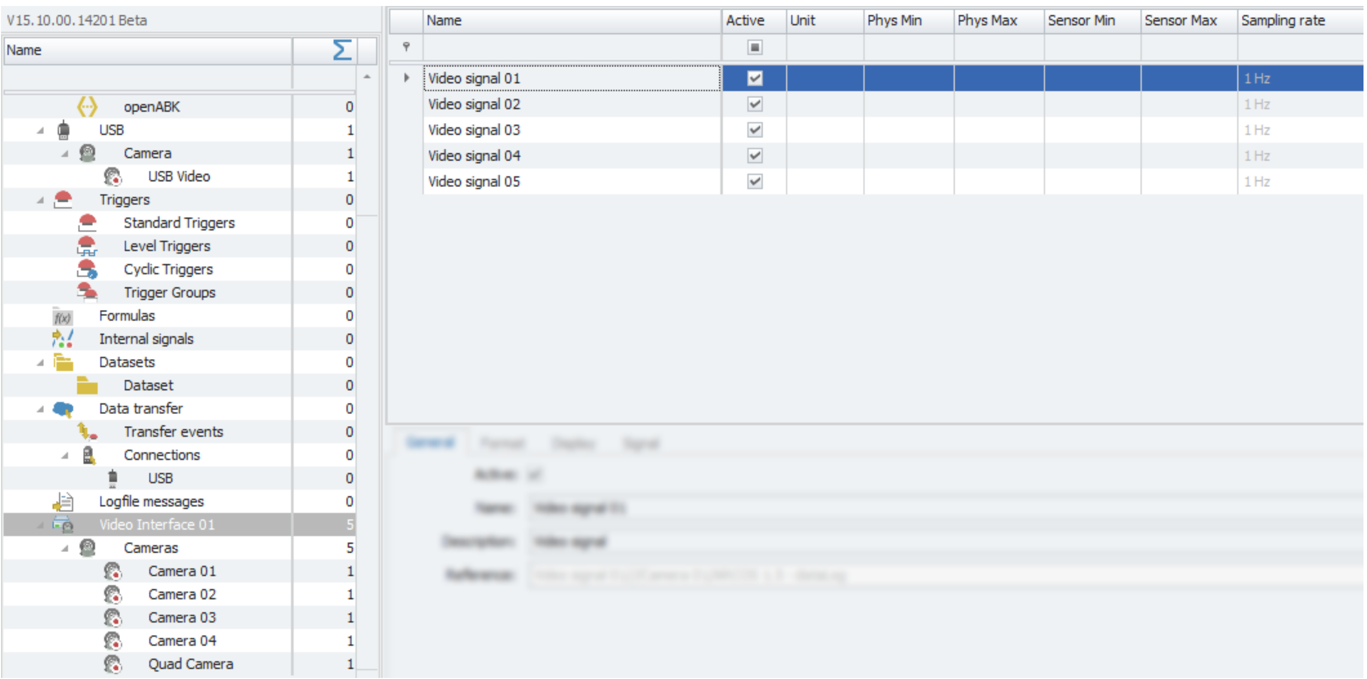
Adding the “Video Interface” to your system will add seven new elements to your “Measurement task tree”:

- Video Interface xx
  This item represents the entire “Video Interface” and all the included child elements.
- Cameras
  This item is a child element of the “Video Interfaces” element and represents all the included cameras.
- Camera xx
  There are four elements named “Camera xx”, they represent the four phyisical cameras connected to the logger.
- Quad Camera
  The “Quad Camera” is a virtual camera, that combines the four signals of the four

physical cameras in one single image. The image is divided into four quarters. Each quarter contains one of the physical cameras images.

5.11.2.3 Grid area for the Video Interface

In the "grid area" you will be presented with an overview of the "Video Interface's" signals. There will be five signals. **Signal five is the signal from the "Quad Camera".** Also you can find here two important functions, which are the **"Column chooser"** (→4.3.1) and the **"Filter editor"** (→4.3.2).



5.11.2.4 Details area for the Video Interface

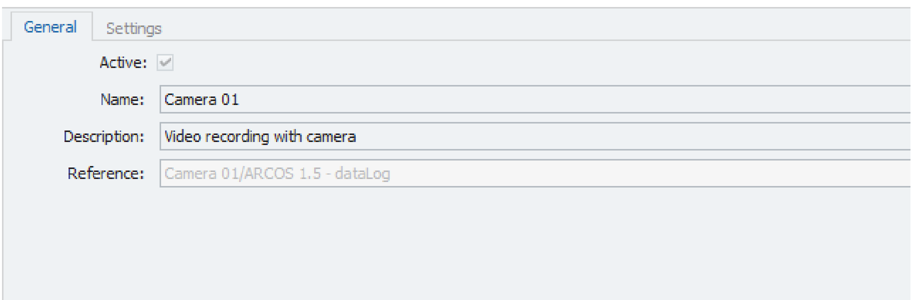
The Details area shows settings either for the tree element **"Video Interface xx"** or one of its child elements. In case the tree element "Cameras" is selected, the details area will only show the "General" tab. Please refer to (→4.2.2).

The "Details area" for video signal will be handled in a separate chapter. Please refer to the Chapter "Video signals" (→5.12).

In case the "Video Interface" or one of the "Camera" elements is selected in the tree, the "Details area" will contain additional tabs, which will be explained in the following.

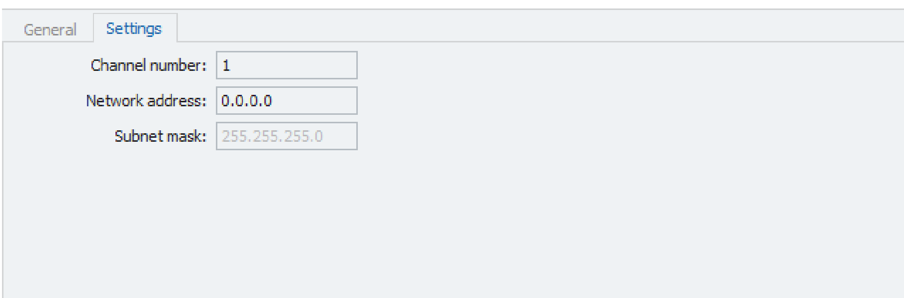
General

This tab allows you to activate or deactivate the entire tree element by ticking/unticking the checkbox, give a user specific name to tree element if wished and add an additional description. The Reference field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed.



Settings (for Video Interface)

This tab allows you to define the settings for the entire "Video Interface".



- Channel number
  Define the number of the "Video Interface" hardware channel.

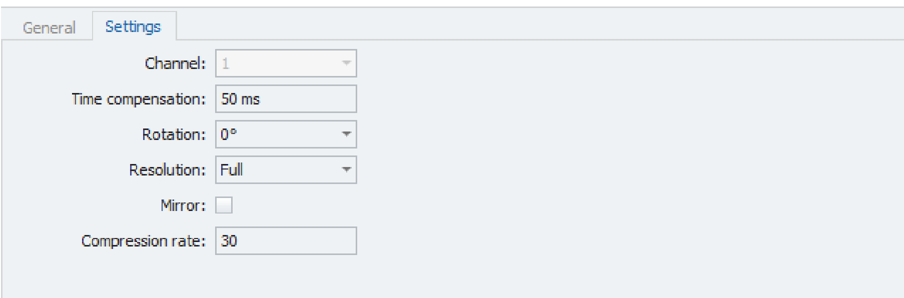
For better orientation and in order to avoid confusion regarding Channelnumbers and -names, a Channels physical number can be found in the logger's **"Web Interface"** and set accordingly.

- Network address
  Define the network adress of the subnet in which your "Video Interface" communicates.

- Subnet mask
  The subnet mask for your "Video Interface". It cannot be changed.

Settings (for Camera elements)

This tab allows you to define the settings for the entire "Camera" child elements of your "Video Interface".



- Channel
  The physical channel, on which the camera operates. It cannot be changed.

- Time compensation
  Define the compensation of the video capture latency. This value is experience based and can vary.

- Rotation
  Allows you to rotate the video in steps of 90°.

- Resolution
  Allows you to set the video's resolution to either full or one quarter.

- Mirror
  Checking the box will mirror the image of the video.

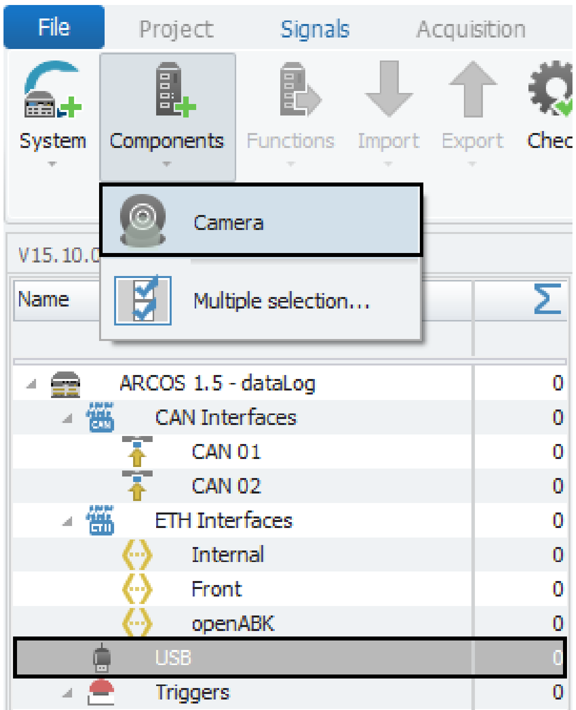
- Compression rate
  Allows you to define the compression rate of your video.

5.11.3 USB camera

The"Camera" component for USB allows you to connect a digital video camera via USB to your logger and control it.

5.11.3.1 Adding a USB camera

The "Camera" component for USB can be added by selecting the tree element "USB" in the "Measurement task tree", then clicking the **"Components"** button and finally choosing "Camera"



5.11.3.2 Tree elements for USB camera

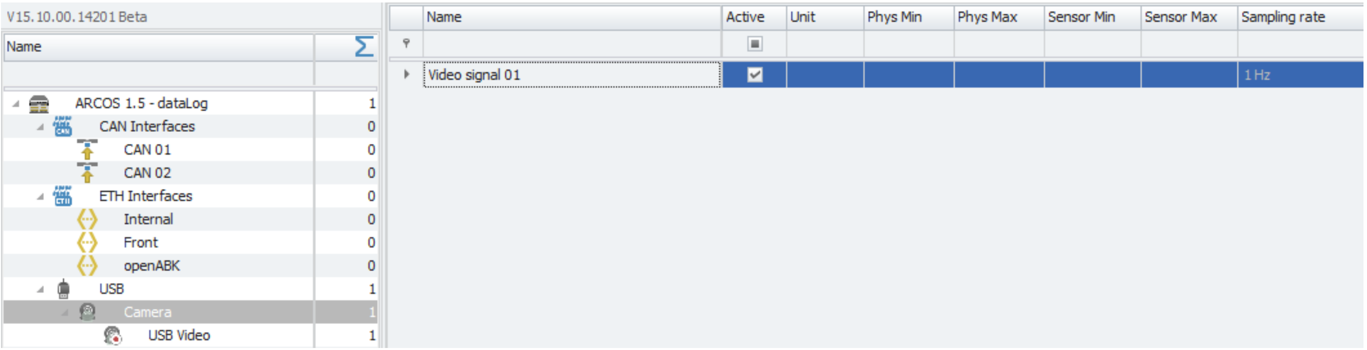
Adding the "Camera" component for USB to your system will add two new elements to your "Measurement task tree":

- Camera
  This item represents the camera itself.

- USB Video
  This item represents the video, that you will receive from the camera.

5.11.3.3 Grid area for USB camera

In the "grid area" you will see the video signal coming from your connected USB camera.



5.11.3.4 Details area for USB camera

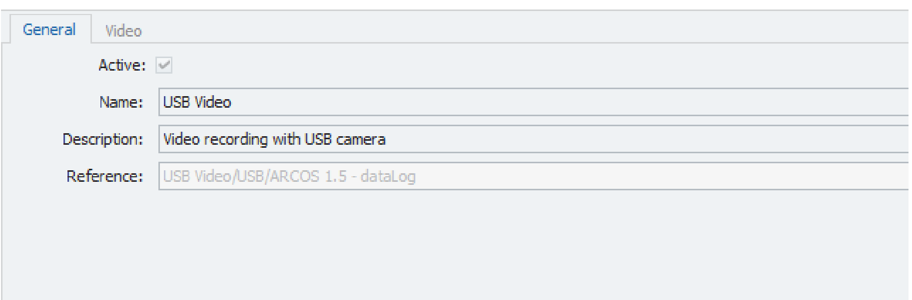
The Details area shows settings either for the tree element "Camera" or its child element "USB Video". In case the tree element "Camera" is selected, the details area will only show the "General" tab. Please refer to (→4.2.2).

The "Details area" for video signal will be handled in a separate chapter. Please refer to the Chapter "Video signals" (→5.12).

In case the "USB Video" is selected in the tree, the "Details area" will contain additional tabs, which will be explained in the following.

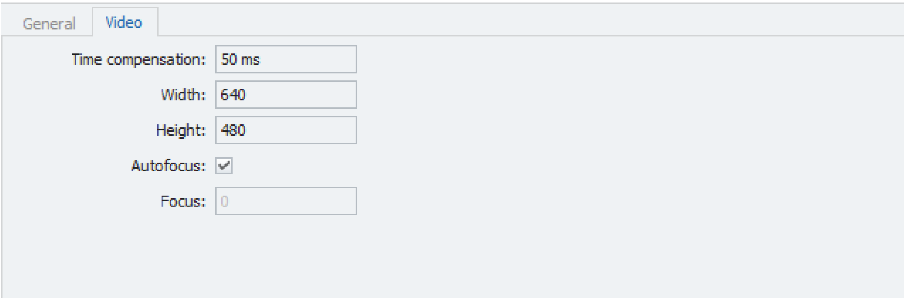
General

This tab allows you to activate or deactivate the entire tree element by ticking/unticking the checkbox, give a user specific name to tree element if wished and add an additional description. The Reference field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed.



Video

This tab allows you to define the settings for the incoming video.



- Time compensation
  Define the compensation of the video capture latency. This value is experience based and can vary.

- Width
  Define the video capture resolution width in pixel.

- Height
  Define the video capture resolution height in pixel.

- Autofocus
  Allows you to enable or disable the the camera's autofocus.

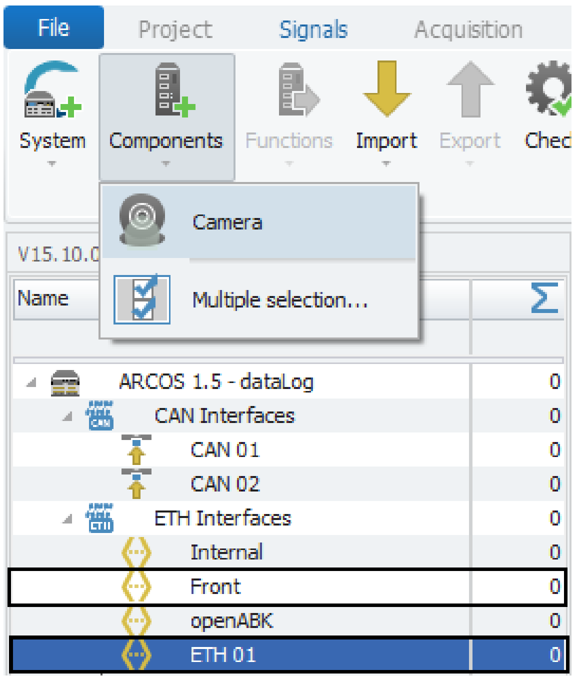
- Focus
  When "Autofocus" has been disabled, this field allow you to set the focus of the camera.
  APPROXIMATE and CALIBRATED devices report the focus metadata in units of diopters (1/meter), so 0.0f represents focusing at infinity, and increasing positive numbers represent focusing closer and closer to the camera device. The focus distance control also uses diopters on these devices.

5.11.4 Ethernet camera

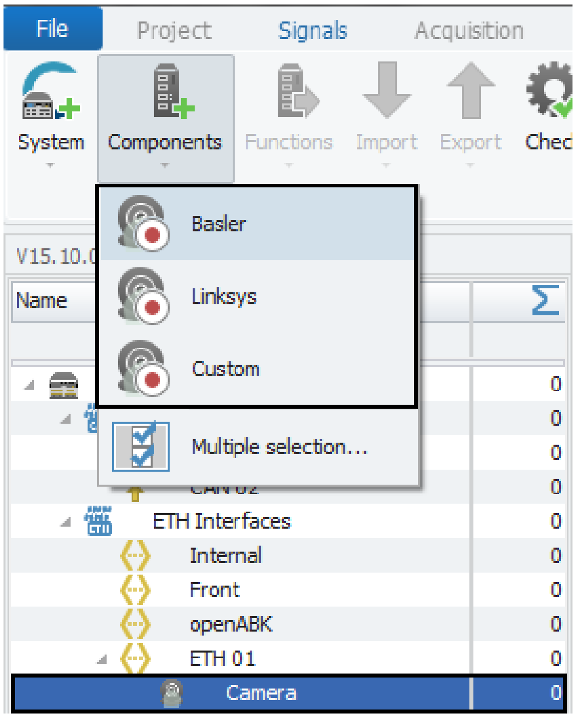
The "Camera" component for Ethernet allows you to connect a digital video camera via Ethernet to your logger and control it.

5.11.4.1 Adding an ETH camera

The "Camera" component for ETH can be added by selecting the tree element "Front", which is a childelement to the tree element "ETH Interfaces" in the "Measurement task tree" or by selecting any manually added "ETH" channel, then clicking the "Components" button and finally choosing "Camera".

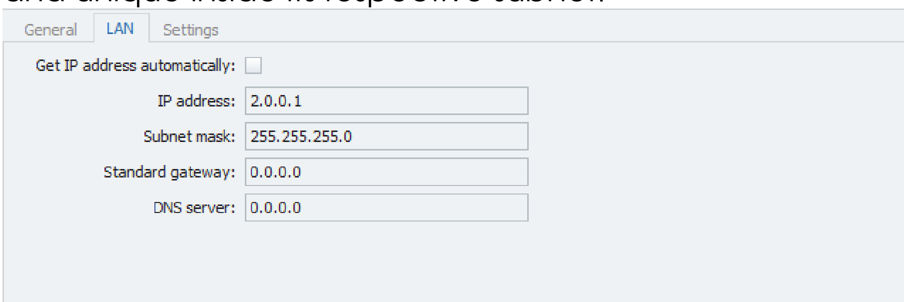


In the next step you select the newly created tree element "Camera", then click the "**Components**" button and finally choose the desired camera model.
At the moment the Plugin support two third party cameras, "Basler" and "Linksys". If your camera is not listed, you can simply choose "Custom" and then set the necessary setting manually.

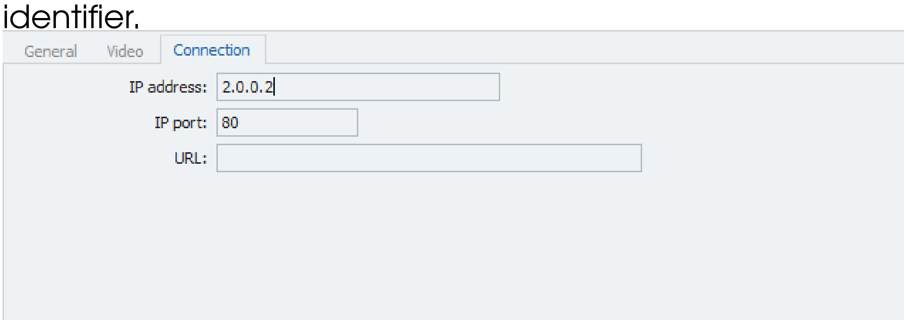


Once a new camera for Ethernet has been added, you may encounter problems regarding the cameras IP settings.
In order to resolve this problem, DHCP has to be disabled for the tree element "Camera" and IP adresses have to be set for both, the tree element "Camera" and its child element, the cameratype you have previously choosen. The IP addresses must belong to the same subnet, and this subnet must not be used by any other operator of the system.

To do so, select the tree element that contains your ETH camera navigate to the "LAN" tab in the details area, untick the checkbox "Get IP address automatically" and set a new IP address.
The first three numbers of your IP address mark the subnet, so they cannot be equal to any other operators IP address, that is not a childelement to the currently selected tree element. The last number marks the client inside the subnet. It has to be higher equal or higher than "1" and unique inside its respective subnet.



Then select your chosen cameratype in the tree (Basler, Linksys or Custom) and navigate to the "Connection" tab in the details area. Here you need to set an IP address that belongs to the same subnet as the one defined in the last step, but, again, with a unique client identifier.

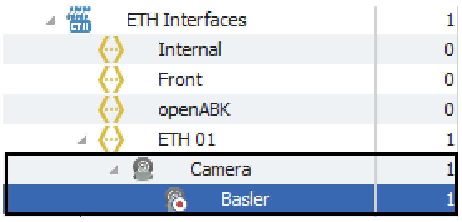


5.11.4.2 Tree elements for ETH camera

Adding the "Camera" component for ETH to your system will add two new elements to your "Measurement task tree".

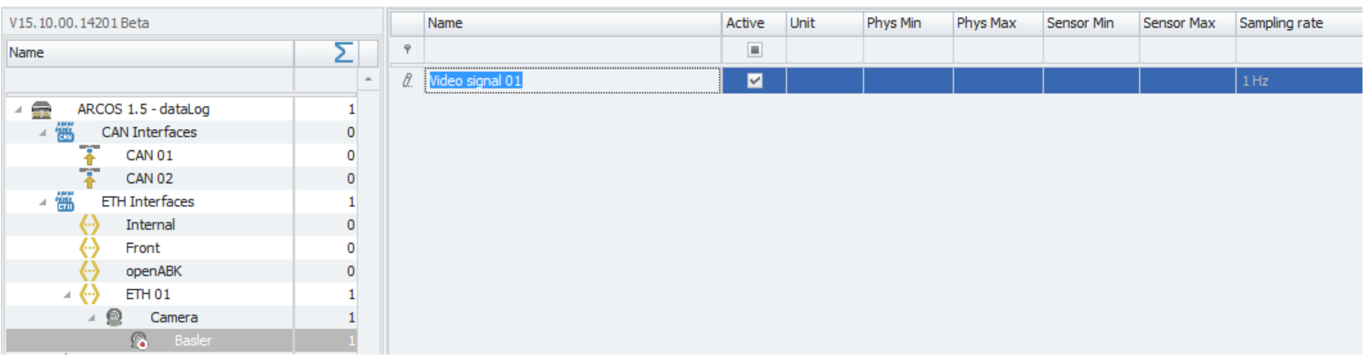
- Camera
  This item represents the camera interface.

- Basler/Linksys/Custom
  This item represents the specific type of camera, that you have connected are configuring.



5.11.4.3 Grid area for ETH camera

In the "grid area" you will see the video signal coming from your connected ETH camera.



5.11.4.4 Details area for ETH camera

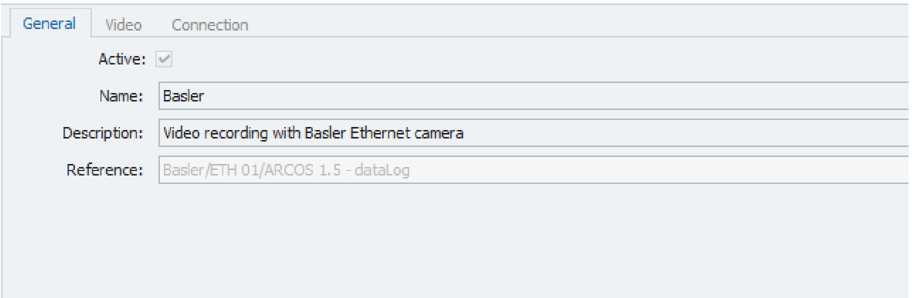
The Details area shows settings either for the tree element **"Camera"** or its child element "Basler/Linksys/Custom". In case the tree element "Camera" is selected, the details area will only show the "General" tab. Please refer to (→4.2.2).

The "Details area" for video signals will be handled in a separate chapter. Please refer to the Chapter "Video signals" (→5.12).

In case the childelement "Basler/Linksys/Custom" is selected in the tree, the "Details area" will contain additional tabs, which will be explained in the following.

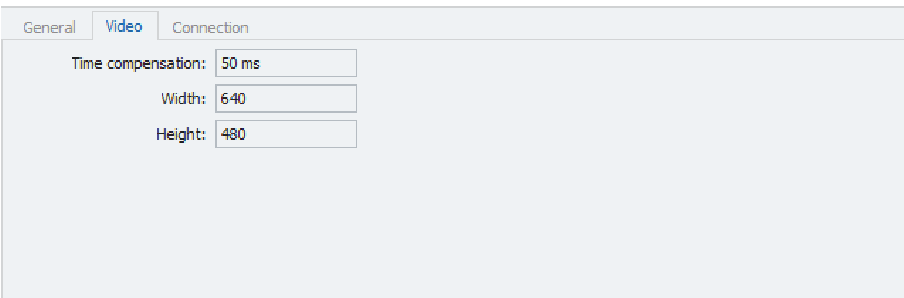
General

This tab allows you to activate or deactivate the entire tree element by ticking/unticking the checkbox, give a user specific name to tree element if wished and add an additional description. The Reference field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed.



Video

This tab allows you to define the settings for the incoming video.



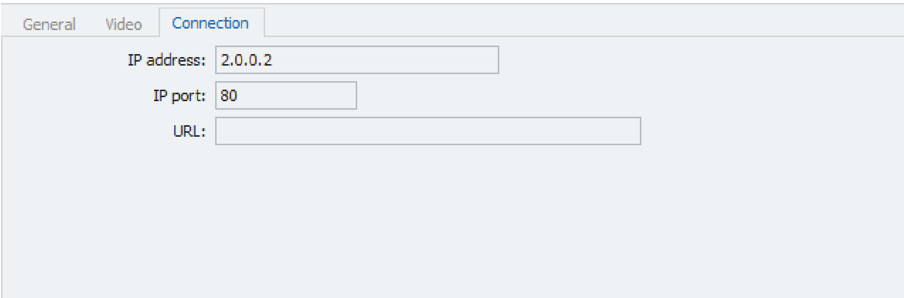
- Time compensation
  Define the compensation of the video capture latency. This value is experience based and can vary.

- Width
  Define the video capture resolution width in pixel.

- Height
  Define the video capture resolution height in pixel.

Connection

This tab contains settings according the connection of the camera with the ETH interface.



- IP address
  Define the IP address of your camera. It has to belong to the same unique subnet (The first three out of the four numbers define the subnet) as the ETH channel to which it belongs and it needs to have a unique client identifier (the last out of the four number defines the client identifier).

- IP port
  Define the IP port of the video stream.

- URL
  Set the URL on which your cameras stream is to be found. This information should be found in your camera's manual or on the manufacturer's website.

  > When working with one of the two already supported models (Basler or Linksys) this field will not be accessible, as the URL is already defined. When working with the Custom profile for other cameras you will need to fill in the URL of your camera stream, otherwise the logger will not now where to look for it and will not receive a video signal.

5.12 Video signals

The video signals, received from any of the three camera types is are equal in fuctionality. The setting for these signals can be set over the "Details area" for each respective signal and will be explained in the following.

5.12.1 Storage method

In order to store a video signal use "AVI" as storage method. Please refer to (→ 7.9).

5.12.2 Settings for video signals

General

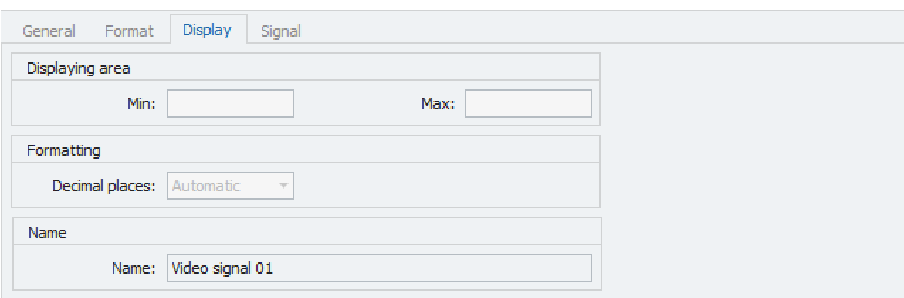
This tab allows you to activate or deactivate the entire signal by ticking/unticking the checkbox, give a user specific name to your signal if wished and add an additional description. The Reference field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed.

Format

Settings in the "Format"-tab do not have any effect on video signals and will therefore not be explained here.

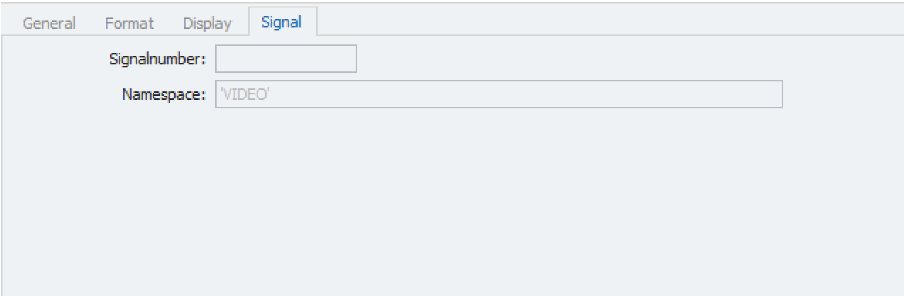
Display

The only setting in the "Display"-tab relevant for video signals is the "Name" setting. It allows you to set a Name to be shown on a display.



Signal

This tab allows you to define signal settings.



- Signal number
  Assign a number to the current signal. This way you will later be able to sort the signals in the grid according to their "Signal numbers".

- Namespace
  The "Namespace" serves as unique identifier for the signal inside the logger.

5.13 Audio recording

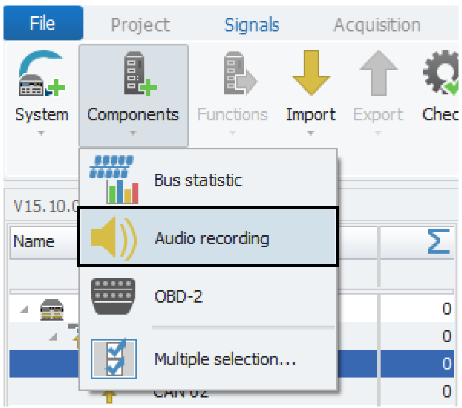
For audio recordings you can connect a microphone to the logger via CAN. The logger will receive the audio signals on one of the CAN channels.

5.13.1 Storage method

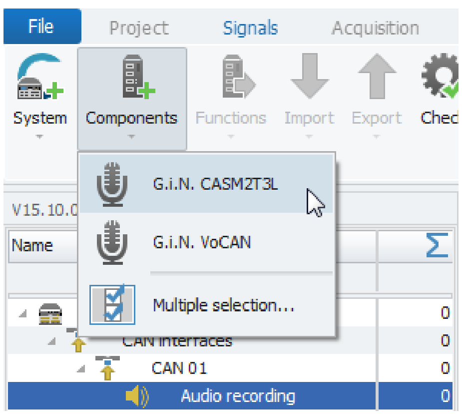
In order to store an audio recording use "WAV" as storage method. Please refer to (→ 7.10).

5.13.2 Adding an Audio recording

To add an "Audio recording" select the CAN channel to which you wish to add the recording, click the "Components" button in the Ribbon and then choose "Audio recording".



Once the "Audio recording" component has been added, select the newly created tree element "Audio recording" in the respective CAN channel, click the "Components" button in the Ribbon and then choose one of the two available microphones.



5.13.3 Tree elements for Audio recordings

Adding an "Audio recording" to your system will add five new elements to your "Measurement task tree":

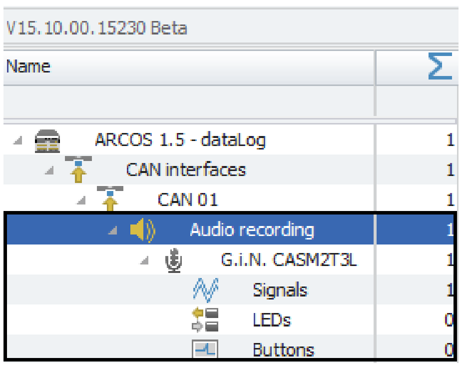
- Audio recording
  This item represents the entire "Audio recording" and all the included child elements.

- G.I.N. CASM2T3L / G.I.N. VoCAN
  This item represents the microphone which you have connected to the logger. At the moment these are the only supported microphone models.

- Signals
  This element contains the incoming audio signal.

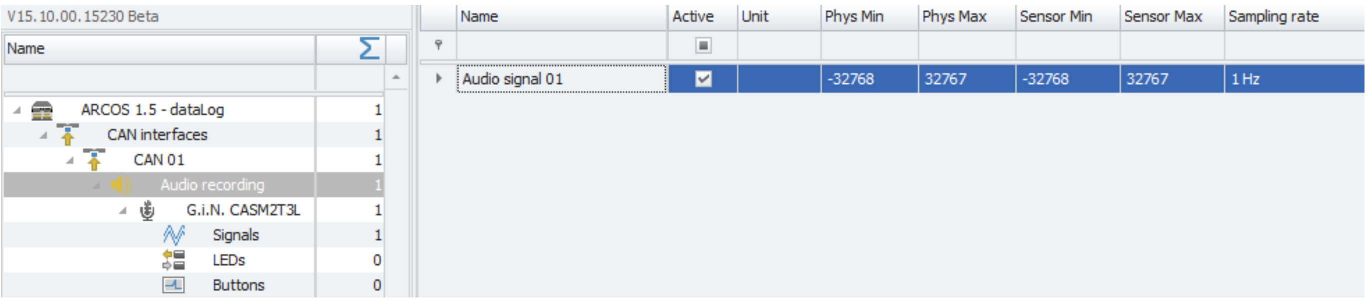
- LEDs
  This element represents the microphone's LEDs. By selecting this item you will be able to give user specific names to the single LEDs in the details area.

- Buttons
  This element represents the microphone's Buttons. By selecting this item you will be able to give user specific names to the single Buttons in the details area.



5.13.3.1 Grid area for Audio recordings

In the "grid area" you will see the incoming audio signal. Also you can find here two important functions, which are the **"Column chooser"** (→4.3.1) and the **"Filter editor"** (→4.3.2).



5.13.4 Details area for Audio recording

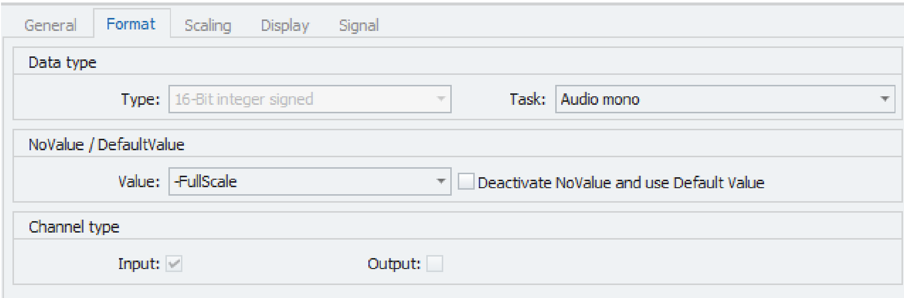
When selecting the audio signal in the grid area, you will be able to access the signal's settings in the details area.

General

This tab allows you to activate or deactivate the entire signal by ticking/unticking the checkbox, give a user specific name to your signal if wished and add an additional description. The Reference field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed.
The sampling rate allows you to set the frequency in which the logger will receive the signal.

Format

This tab contains information and options regarding file format, tasks and Channel type.



- Data type
  This field tells you the type of data (in this case "**16-Bit integer unsigned**") and allows you to special tasks for this signal such as "Audio mono".

- NoValue / DefaultValue
  This field allows you to define the value that will be shown if a signal value is read as invalid.

Scaling

The settings in the "Scaling"-tab are not relevant for working with audio recordings.

Display

The only setting in the "Display"-tab relevant for audio recordings is the "Name" setting. It allows you to set a name to be shown on a display.

- Displaying area
  Shows the value range which will be shown on a display. It usually should match the "Physical range" from the "Scaling" tab.

- Formatting

The dropdown menu "**Decimal places**" allows you to set how many decimal numbers of the value will be shown on a display.

- Name
  Allows you to set a Name to be shown on a display.

Signal
This tab allows you to define signal settings.

- Namespace
  The "Namespace" serves as unique identifier for the signal inside the logger.

5.14 Internal signals

Internal signals are values that are not fed into the logger from the outside (CAN, GPS...), but are generated within the logger and provide information about internal system states.

Internal signals are largely treated in the same manner as a CAN signal. They can be recorded over time, classed or processed; they can generate alarms or be displayed. Only they can't be directly stored in traces, since the values, with the exception of the bus statistics, are not, as required, in the form of bus messages.

The internal signals are divided into three groups and the details on each group can be found in the respective section:

- Run state (→5.14.4)
- System info (→5.14.5)
- Time (→5.14.6)

5.14.1 Storage method

In order to store internal signals use one of the following signal storage methods.

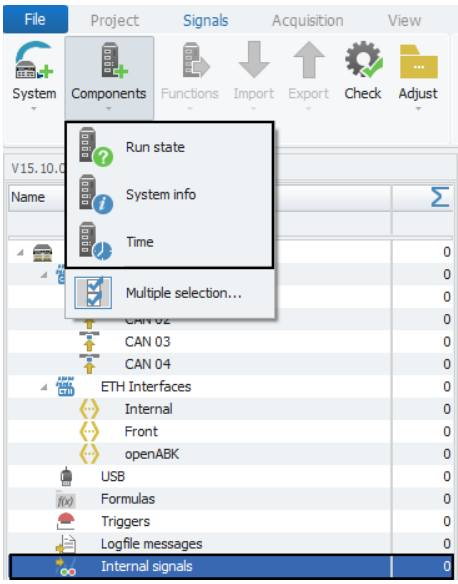
- ATFX (→ 7.4)
- MDF 4.0 (→ 7.5)
- MDF 4.1 (→ 7.6)

5.14.2 Accessing internal signals

"Internal signals" do not need to be imported, as they are continuosly produced by the logger itself. They only need to be made accessible and can then be activated for to be used liked regular signals in further processing, for example as triggers or in formulas.

To access "Internal signals" select the tree element "Internal signals", click the "Components" button in the Ribbon and then choose, which of the three categories of internal signals you wish to access.

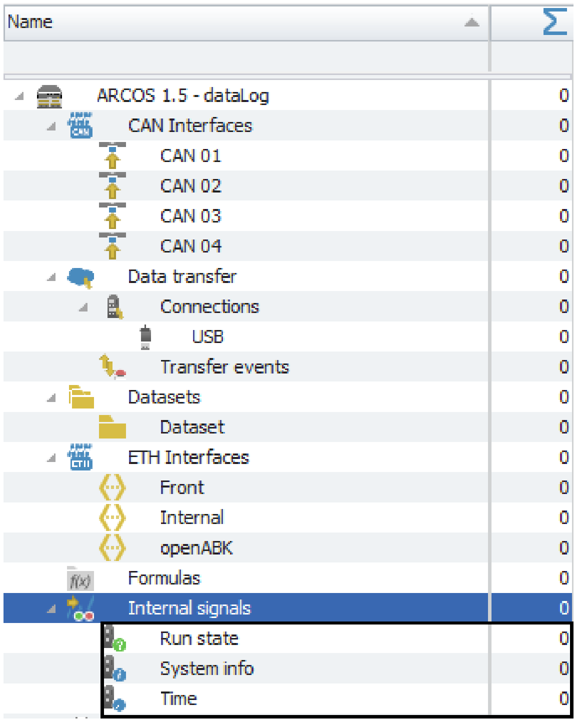
If you wish to acces two or all three categories, you can either access them one by one or through the button **"Multiple selection..."**.



5.14.3 Internal signals properties

5.14.3.1 Tree elements for Internal signals

Each category of "Internal signals" that has been accessed, will appear in the tree as a child element to the tree element "Internal signals" with its respective name. In the right part of the tree it will also show, how many signals have been activated for further processing.



5.14.3.2 Grid area for Internal signals

In the "Grid area" you will be presented with an overview of the available "Internal signals". Also you can find here two important functions, which are the **"Column chooser"** (→4.3.1) and the **"Filter editor"** (→4.3.2).

5.14.3.3 Details area for Internal signals

The Details area shows settings either for the selected tree element ("Internal signals", "Run state", "System info" or "Time") or the selected signal in the grid area. In case a tree element is selected, the details area will only show the "General" tab. Please refer to (→4.2.2).

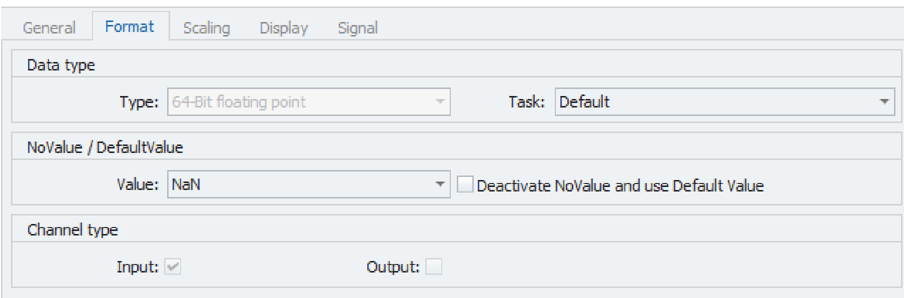
In case a signal is selected in the grid area, the details area will contain additional tabs which will be explained in the following.

General

This tab allows you to activate or deactivate the signal by ticking/unticking the checkbox and thus to make it available for internal Recording and further use (e.g. triggers, formulas, display,...)
It also allows you to give a user specific name to your signal if wished and add an additional description. The Reference field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed. The "Sampling rate" allows you to set, how frequently a signal should be requested.

Format

This tab contains information and options regarding file format, tasks and Channel type.



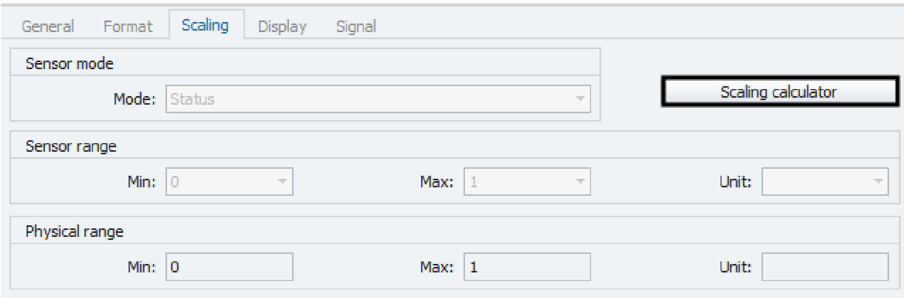
- Data type
  This field tells you the type of data (in this case "64-Bit floating point") and allows you to apply special tasks for this signal such as "GPS Longitude", "GPS Latitude", "UTC hour", "Audio mono" and more.

- NoValue / DefaultValue
  This field allows you to define the value that will be shown if a signal value is read as invalid.

- Channel type
  This field tells you whether you are dealing with a "Input" channel or "Output" channel.

Scaling

The fields accessible directly through the tab allow for basic scaling operations to convert analog measurement in engineering units. The **"Scaling calculator"** allows for more refined scaling options with a large range of functions. For details on how to use the "Scaling calculator" please refer to the **IPEmotion Documentation - Section 3.4.5 "Channel configuration and scaling"**.



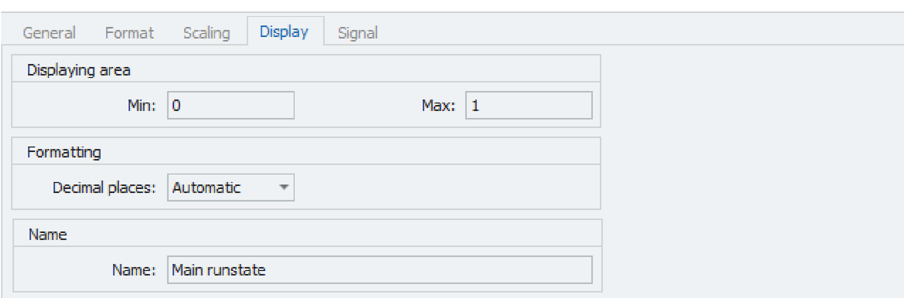
- Sensor Mode
  The sensor mode tells the type of signal. It can be of different types such as "Status", "Voltage", "Frequenzy" or others. It cannot be changed and serves for IPEmotion to know what kind of signal it is dealing with.

- Sensor Range
  Shows the raw value range of the signal.

- Physical Range
  Allows you to set a range to which you would like to "scale" your signal and also define the unit to use. For more refined scaling please use the "Scaling calculator" and refer to the **IPEmotion Documentation - Section 3.4.5 "Channel configuration and scaling"**.

Display

This tab allows you to define what information about the current signal will be shown on a display if one is connected.



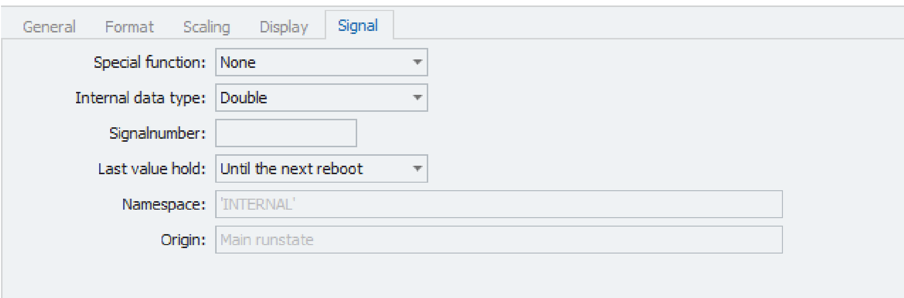
- Displaying area
  Shows the value range which will be shown on a display. It usually should match the "Physical range" from the "Scaling" tab.

- Formatting
  The dropdown menu "**Decimal places**" allows you to set how many decimal numbers of the value will be shown on a display.

- Name
  Allows you to set a Name to be shown on a display.

Signal

This tab allows you to define signal settings.



- Internal data type
  Assign an internal data type to the signal. Available data types are "**Double**" and "**String**".

- Signal number
  Assign a number to the current signal. This way you will later be able to sort the signals in the grid according to their "Signal numbers".

- Last value hold
  Specify, for how long the last value of the signal will be hold.

- Timeout
  Specify the timeout period for the current signal. If the data source doesn't send data for the specified time period, the value of the signal is set to "NaN (Not a Number)" and will be displayed as "–" in a display.

- Namespace
  The “Namespace” serves as unique identifier for the signal inside the logger.

5.14.4 Run state

This group of internal signals provides information about the states of external signals and switches, as well as about the wake conditions of the buses that started the logger.

These internal channels can be used as a supplement to the standard logger responses, for example, to start or stop defined methods on signal states. They can also set off warning messages and alarms.

If a bus wake condition is defined, it is output even if this condition was not used to start the current measurement. This makes it possible, during operation, to check when wake conditions are met and trigger methods/messages/alarms.

The following table gives an overview and explanation of “Runs state” signals.

Signal	Meaning	Possible Values
Main runstate	Main runstate	0 = off 1 = on
Active wake up condition	Type of wake condition that currently started the logger	1 = A-switch 2 = Cl. 15 3 = WoX
State system switch		0 = off 1 = on
State Clamp 15		0 = off 1 = on
State wake up condition		0 = not met 1 = met
Measuring time delay	Delay between the current logger time and the timestamp of the data being processed.	Value in (ms)
Error	Tells you whether an error has occurred in your present configuration.	0 = no error 1 = error
Error missing channel		0 = no error 1 = error
Error initialized interface		0 = no error 1 = error
Error missing interface		0 = no error 1 = error
Error persistencefile		0 = no error 1 = error
Dataset size	Tells you the current size of the dataset.	Value in MB
Measuring time delay	Tells you the current measurement delay.	Value in ms

5.14.5 System info

These signals provide information on CPU status and allocation of both working memory (mem) and hard drive (disk). Sizes are specified in absolute numbers (in MB or in °C) or as relative values (in %).
This category also contains the loggers frontnumber.

The following table gives an overview and explanation of "System info" signals.

Signal	Meaning	Unit
Total memory	total working memory (RAM)	(MByte)
Free memory	free working memory	(%)
Used memory	used working memory	(%)
Total disk space	total disk space	(MByte)
Free disk space	free disk space	(%)
Used disk space	used disk space	(%)
Total external disk space	total disk space	(MByte)
Free external disk space	free disk space	(%)
Used external disk space	used disk space	(%)
CPU load	processor load	(%)
CPU temperature	processor temperature	(°C)
Frontnumber	Tells you the loggers unique frontnumber	Value

5.14.6 Time

These signals provide information on time and date. They allow you to record a timeline in order to trace the occurrence of events in the logger.

The following table gives an overview and explanation of "Time" signals.

Subtype	Meaning	Unit
Time since beginning of day	Time since 00:00:00 h UTC	(s)
Time sincefirmware start	Time since the firmware was started (values <0 represent values during booting phase)	(s)
Time since START MEAS	Time since measurement has started	(s)
Current date	Current date in the format ddmmyy	Value
Current year	Current year in the format yyyy	Value
Current month	Current month in the format mm	Value
Current day	Current day in the format dd	Value
Current time	Current time in the format hhmmss	Value
Current hour	Current hour in the format hh	(h)
Current minute	Current minute in the format mm	(min)
Current second	Current second in the format ss	(s)
Current microseconds	Current microseconds	(µs)

5.15 Formulas

A formula is a calculated signal. Apart from the fact of being calculated it has largely the same properties as physical signals and can be modified or used for further processing in the same way. It allows you, to combine already existing signals into a new signal. Therefore the existence of physical signals is requirement in order for formulas to function properly.

All the signals/quantities, that have been individually defined can be further processed in formulas. This also applies to internal signals.

A formula is a one-line term made up of operators and functions that are applied to numbers and signals, which will have a calculated signal as a result. Calculated signals can in turn be used as normal signals in another formula. Not only numbers and signals can serve as function arguments, but also the name of any already defined formula.

The operator priorities used by the formula interpreter are listed in the Table In addition, the interpreter observes the "multiplication/division before addition/subtraction" rule. When uncertain about priorities, you should use brackets.
Bear in mind that signal names and operators, in particular, are case-sensitive. Throughout the signals, whether they be bus, internal or computed, no name may be used twice. The functions can have multiple applications within a formula – with the exception of integration (INT_STD), differential (DIFF) and moving average (MEAN). **So each formula may apply only once the function INT_STD, DIFF or MEAN.**

Operator	Meaning	Priority
AND	Logical and	1
OR	Logical or	1
XOR	Logical, exclusive or	1
<=	Less than or equal to	2
>=	Greater than or equal to	2
<>	Not equal to	2
=	Equal to (comparison)	2
>	Greater than	2
<	Less than	2
+	Addition	3
-	Subtraction	3
*	Multiplication	4
/	Division	4
^	Exponentiation (2^3 =>"raise 2 to the power of 3")	5

Special features of moving averages

Unlike the other formulas, the moving average (MEAN) uses not only the current value but also a certain number of previous values. This number is defined by the parameter "Delay depth (values)". The number is theoretically unlimited, but in practice it is limited by the working memory and processing speed.
Assuming the number is =100, then this computes the average over the last 100 samples. At the next sampling instance, the oldest of the 100 values is dropped and the current va-

lue is included. At start-up no samples are available for review, so the buffer is still empty. The buffer is filled up with the first valid value and then moves through the sample values. In the event of a signal timeout ("Not a Number", abbreviated NaN, or as a value, also called NoValue), this review is interrupted. As long as the value of the signal is NaN, the moving average is also equal to NoValue (processing a NoValue yields another NoValue). Once the signal goes back to a valid value, the buffer – just like at the start of measurement – is filled with the first valid value, thus resuming processing of a valid average.
In a triggered timelog, the moving average is based on the continuous, untriggered value stream. Once the start trigger is activated, the moving average buffer is filled, according to the depth of averaging, with the values gathered prior to the trigger time. This means the moving average at trigger time is computed from values obtained before the start trigger.

Special features of "logical" operators

Basic rules for applying the logical operators from are:

- Note upper/lower case: always lowercase logical operators
- Always bracket operands, if you use logical operators. Example: ('signal1') or ('signal2').

The results derived by an operator fundamentally depends on the type of data to which it is applied.
The formula "('signal1') and ('signal2')" applies the logical operator "and" to two channels with rational numbers. In this case, the integer parts of the respective channels are linked bit-wise. So if:

Signal1= 6 (dec) or 110 (bin)
Signal2= 3 (dec) or 011 (bin)

Then "('signal1') and ('signal2')" yields the result:

6 and 3 = 2 (dec) or 110 and 011 = 10 (bin)

The formula "('signal1'>5) and ('signal2'>2)" applies the logical operator "and" to the binary intermediate results of two channels with rational numbers. The intermediate results (signal1>5) and (signal2>2) yield binary "0" or "1", depending on the value of the channels. The link "and" merely links these values and can have only "0" or "1" as a result.

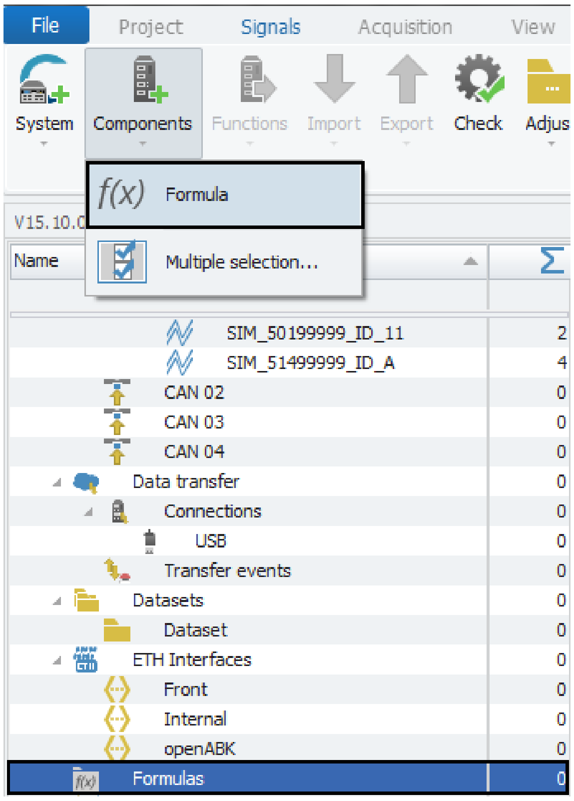
5.15.1 Storage method

In order to store calculated signals use one of the following signal storage methods.

- ATFX (→ 7.4)
- MDF 4.0 (→ 7.5)
- MDF 4.1 (→ 7.6)

5.15.2 Adding a formula

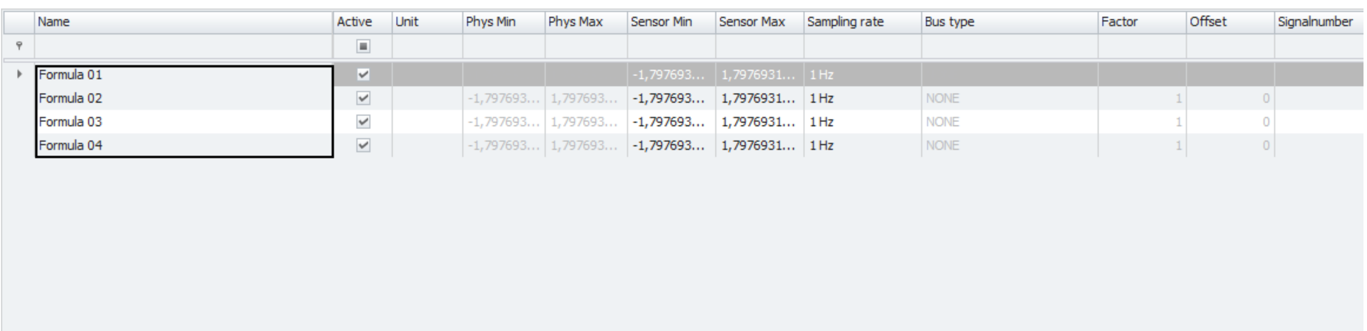
To add a new formula select the tree element "**Formulas**", click on the "**Components**" button in the Ribbon and then choose "**Formula**"



This will add a generic formula, which in the beginning will have the value "1". Instructions on how to work with that formula and modify it, will be explained in the section "**Calculation**" (→5.15.4).

5.15.3 Grid area for formulas

All the formulas, that have been added to your system so far, will be presented in an overview in the grid area. Also you can find here two important functions, which are the "**Column chooser**" (→4.3.1) and the "**Filter editor**" (→4.3.2).



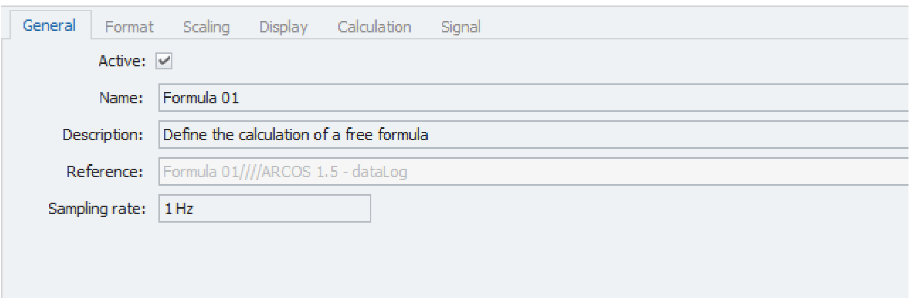
5.15.4 Details area for formulas

The Details area shows settings either for the tree element "Formulas" or for a single formula/signal which has been selected in the grid area. In case a tree element is selected, the details area will only show the "General" tab. Please refer to (→4.2.2).

In case a single formula/signal is selected in the grid area, the details area will contain additional tabs which will be explained in the following.

General

This tab provides general settings for the selected formula/signal.



- Active
  Allows you to activate or deactivate the selected formula/signal.

- Name
  Give a user-defined Name to the selected formula/signal.

- Description
  Give a user-defined description to the selected formula/signal.

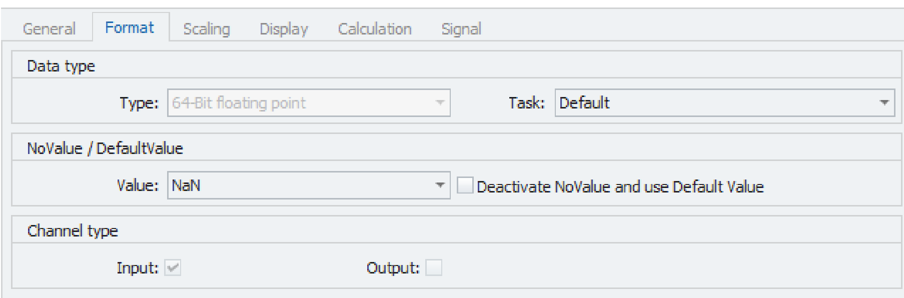
- Reference
  This field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed.

- **Sampling rate**
  This field allows you to set the formula/signal's processing rate.

> Formulas are executed in cycles. The processing frequency can be defined for each formula. This rate can be set independently of the recording rate. The computation process is applied to the values of the input quantities actually registered at the time of the processing phase. If the processing rate is higher than the signal rate of an input quantity, then the last value of the variable is used again. If the processing rate is lower, then some of the values of the quantity are omitted. (An analogous procedure applies when the recording rate is higher or lower than the processing rate.)
>
> Particularly with such channel processing as min, max, sum or avg, periodic processing means that results are derived for the current sampling instance and not over the entire channel. If, for example, the minimum over an entire channel is required, the minimum/maximum value recording must be used.
>
> Processing a timeout value (Not a Number , NaN) also yields a NaN value.

Format

This tab contains information and options regarding file format, tasks and Channel type.

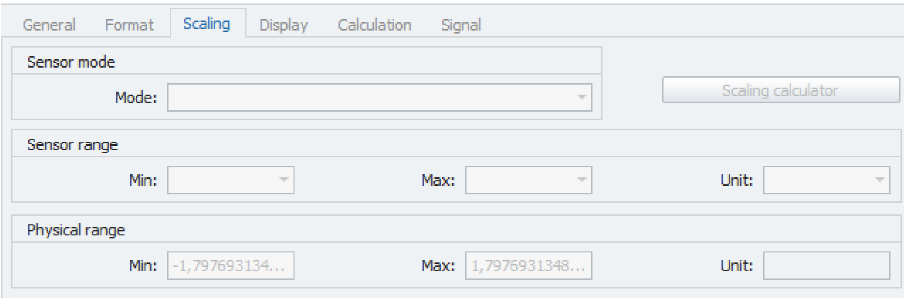


- **Data type**
  This field tells you the type of data (in this case "64-Bit floating point") and allows you to apply special tasks for this formula/signal such as "GPS Longitude", "GPS Latitude", "UTC hour", "Audio mono" and more.

- **NoValue / DefaultValue**
  This field allows you to define the value that will be shown if a formula/signal value is read as invalid.

Scaling

The fields accessible directly through the tab allow for basic scaling operations to convert analog measurement in engineering units. The **"Scaling calculator"** allows for more refined scaling options with a large range of functions. For details on how to use the "Scaling calculator" please refer to the **IPEmotion Documentation - Section 3.4.5 "Channel configuration and scaling"**.



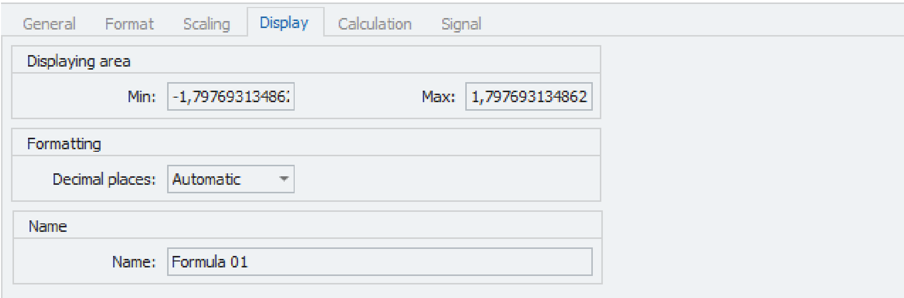
- **Sensor Mode**
  The sensor mode tells the type of formula/signal. It can be of different types such as "Status", "Voltage", "Frequenzy" or others. It cannot be changed and serves for IPEmotion to know what kind of formula/signal it is dealing with.

- **Sensor Range**
  Shows the raw value range of the formula/signal.

- **Physical Range**
  Allows you to set a range to which you would like to "scale" your formula/signal and also define the unit to use. For more refined scaling please use the "Scaling calculator" and refer to the **IPEmotion Documentation - Section 3.4.5 "Channel configuration and scaling"**.

Display

This tab allows you to define what information about the current formula/signal will be shown on a display if one is connected.



- Displaying area
  Shows the value range which will be shown on a display. It usually should match the "Physical range" from the "Scaling" tab.

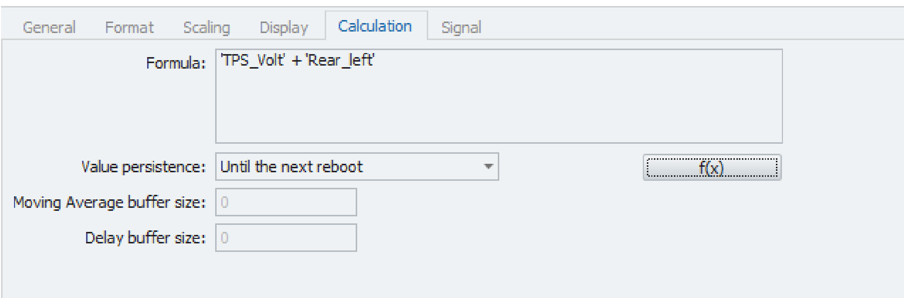
- Formatting
  The dropdown menu **"Decimal places"** allows you to set how many decimal numbers of the value will be shown on a display.

- Name
  Allows you to set a Name to be shown on a display.

Calculation

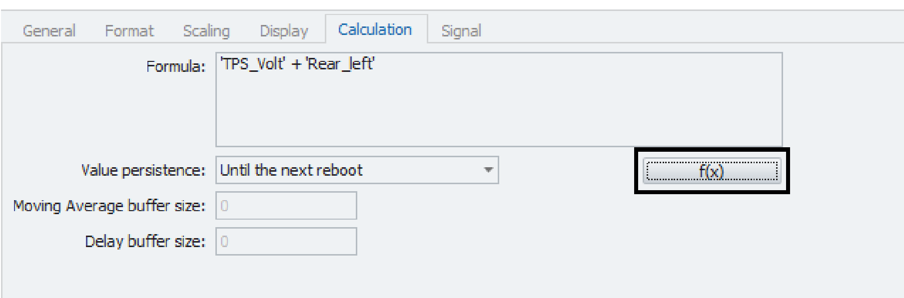
This tab provides the main functionality for working with formulas. Here you can define the functions and operators of your formula and what numbers and/or signals they should be applied to.
The operator priorities used by the formula interpreter are listed in the Table In addition, the interpreter observes the "multiplication/division before addition/subtraction" rule. When uncertain about priorities, you should use brackets.
Bear in mind that signal names and operators, in particular, are case-sensitive. Throughout the signals, whether they be bus, internal or computed, no name may be used twice. The functions can have multiple applications within a formula – with the exception of integration (INT_STD), differential (DIFF) and moving average (MEAN). **So each formula may apply only once the function INT_STD, DIFF or MEAN.**

Operator	Meaning	Priority
AND	Logical and	1
OR	Logical or	1
XOR	Logical, exclusive or	1
<=	Less than or equal to	2
>=	Greater than or equal to	2
<>	Not equal to	2
=	Equal to (comparison)	2
>	Greater than	2
<	Less than	2
+	Addition	3
-	Subtraction	3
*	Multiplication	4
/	Division	4
^	Exponentiation (2^3 =>"raise 2 to the power of 3")	5



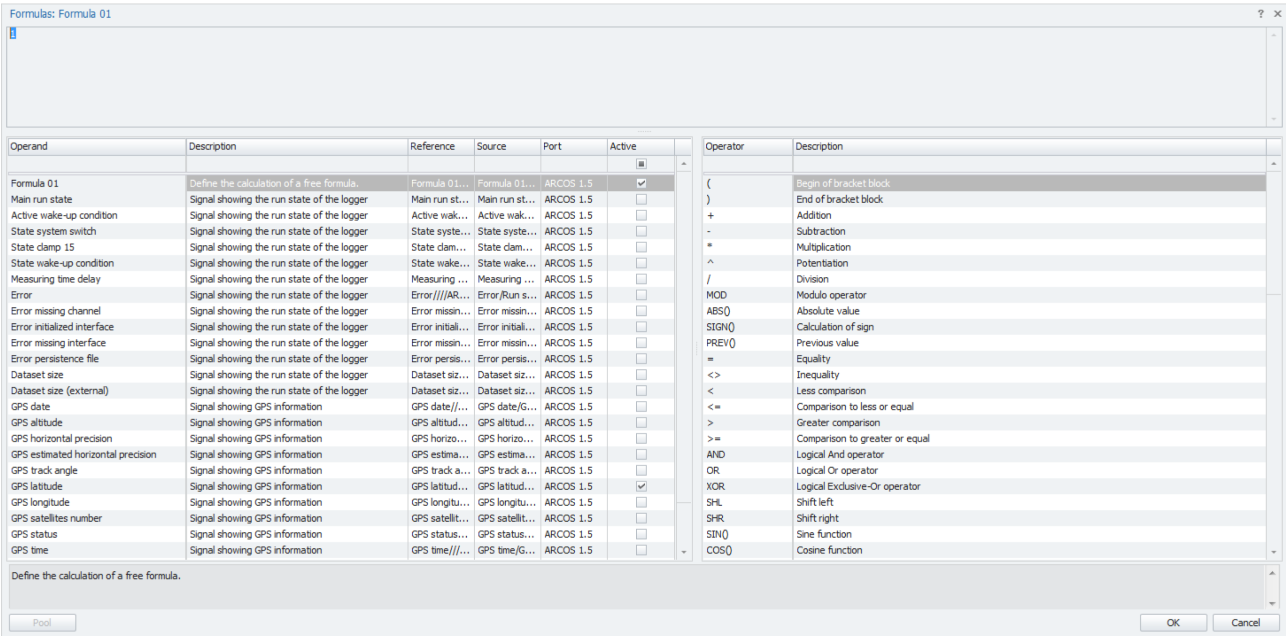
- Formula
  This field allows you to manually enter a formula.

- Formula editor
  Although there is the possibility to manually define formulas, most of the time it will be more convenient to define formulas using the "Formula editor".



The "Formula editor" provides an overview of your current formula (the upper table), as well as the possible "Operands" (left table) and "Operators" (right table) and a short description for each item in the lower table.

You can add items to your formula either by doubleclicking on them or per drag and drop.
If you wish to delete an item from the formula, you can simply mark it with the mouse and than delete it.
If you are uncertain about the operator priorities please refer to the table above.
When confirming a formula that has been defined in the "Formula editor" by clicking OK, the editor will automaitcally validate the formula's correctness. If the formula is not correct, that will result in an error message.



- Value persistence
  Defines the persistence of the signal, so that the value of the signal is sustained even beyond logger restart.

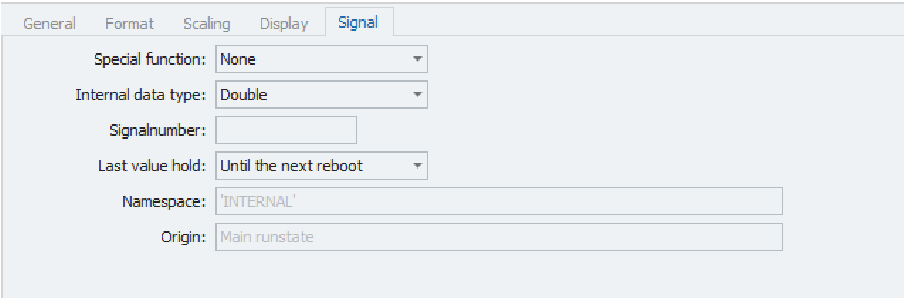
- Moving average buffer size
  Defines buffer size of the MEAN() filter.

- Delay buffer size
  Defines the size of the delay() buffer.

Signal

This tab allows you to define formula/signal settings.

- Signal number
  Assign a number to the current formula/signal. This way you will later be able to sort the formulas/signals in the grid according to their "Signal numbers".

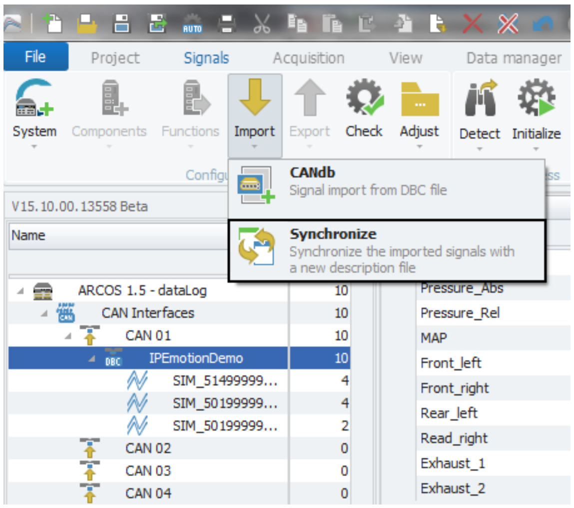


- Namespace
  The "Namespace" serves as unique identifier for the formula/signal inside the logger.

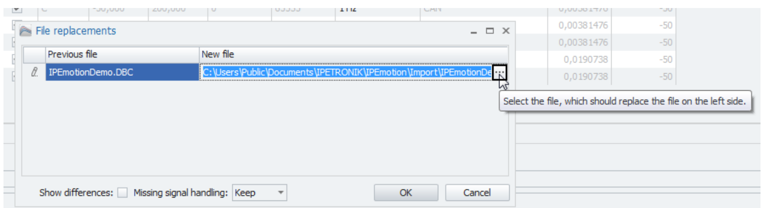
5.16 Synchronizing signals

This option allows you to synchronize the signals of a already imported description file with a newer or older version of the same description file.

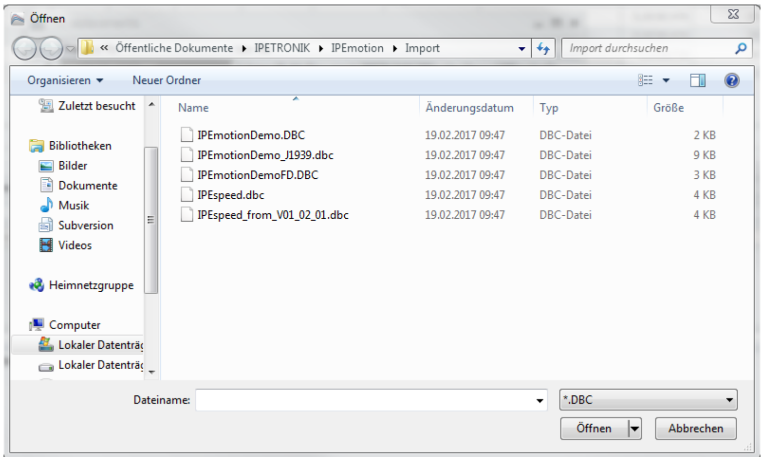
To do so, select the CAN channel or description file you wish to synchronize in the measurement task tree, click "Import" in the ribbon and then choose **"Synchronize"**.



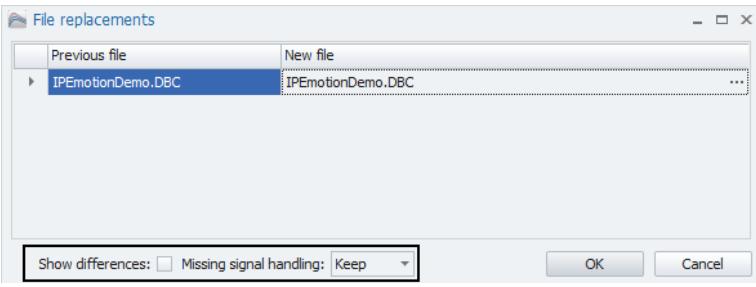
In the resulting window you will be shown which is the file preivously used under "Previous file" and you will be able to choose a new file with which you would like to synchronize. To do so, click in the field for the new file and then click the button with the three dots as shown in the following figure.



In the resulting window you will be able to choose the file with which you would like to synchronize and confirm by clicking "Open".



Before you complete the synchronization process, you may adjust two settings on how to handle differences between the two description files.



- Show differences
  If marked active and there are differences between the current configuration and the new description file, a dialog with all these differences will be displayed before the synchronization is be performed. Properties, that cannot be edited, will not be taken into account in this comparison.

- Missing signal handling
  This dropdown menu allows you to specify how signals, that are no longer available in the new description file, should be handled.

To complete the synchronization process click "OK".

5.17 Transferring measurement tasks to the logger

6 Triggers

Triggers define a certain event or condition in a way, that they can trigger an action. They possess two distinguishing features compared to formulas:

- They are not a signal but an **event**
- A trigger alway has to be a **truth condition** and therefore can only have to possible values: true or false

Other than that, they are similar to formulas and can mainly be treated in the same way:

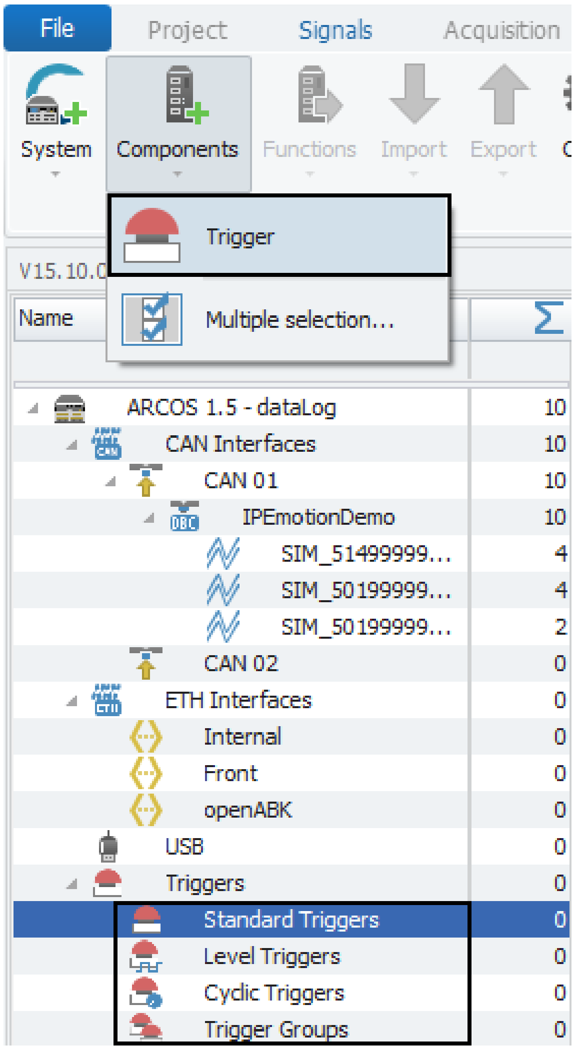
A trigger allows you, to combine already existing signals. Therefore the existence of physical signals is requirement in order for triggers to function properly.
All the signals/quantities, that have been individually defined can be further processed in triggers. This also applies to internal signals.
A trigger is a one-line term made up of operators and functions that are applied to numbers and signals, which will have a calculated condition as a result.

The triggers are divided into four groups and the details on each group can be found in the respective section:

- Standard Triggers (→6.5)
- Level Triggers (→6.6)
- Cyclic Triggers (→6.7)
- Trigger groups (→6.8)

6.1 Adding a trigger

To add a new trigger select the desired type of trigger in the tree, click on the "**Components**" button in the Ribbon and then again choose the desired type of trigger.

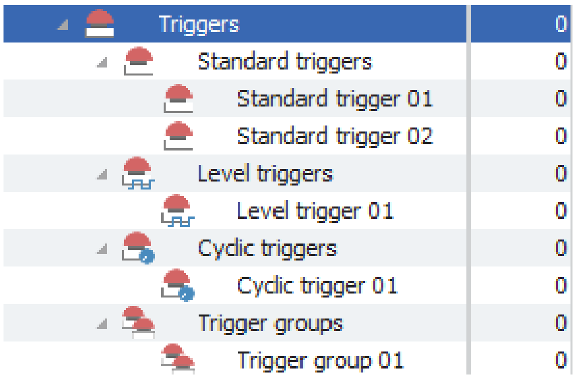


This will add a generic trigger condition, which in the beginning will have the value "1". Technically this trigger condition works the same way as a formula and instructions on how to modify it will be explained in the section "**Calculation**" (→5.15.4).

6.2 Tree elements for triggers

By default the "Measurement task tree" will contain all four categories of triggers.

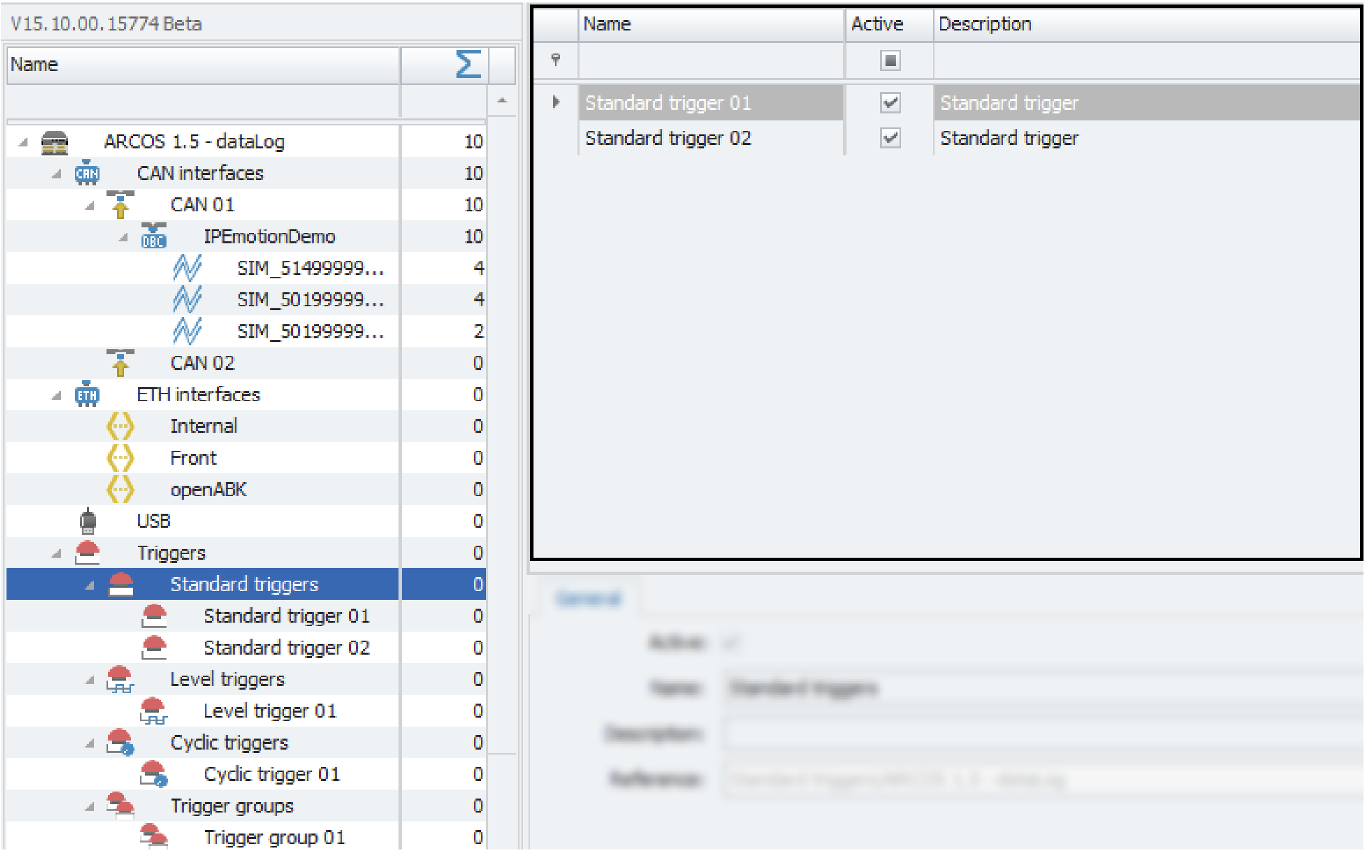
Each trigger that you add to your system will be shown and accessible in the measurement task tree in its respective trigger category.



6.3 Grid area for Triggers

The grid area for each Trigger category will present you with an overview of the triggers that have been added to your system so far.

Also you can find here two important functions, which are the "**Column chooser**" (→4.3.1) and the "**Filter editor**" (→4.3.2).



6.4 Details area for Triggers

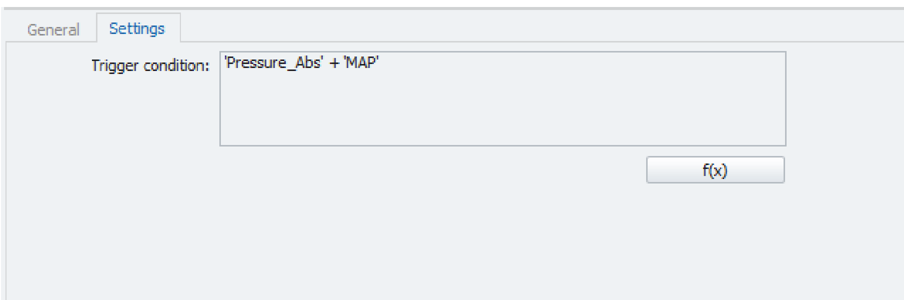
The Details area shows settings either for a selected tree element. In case the "Triggers" element or one of the four categories ("Standard Trigger", "Level Triggers", "Cyclic Triggers" or "Trigger groups") is selected, the details area will only show the "General" tab. Please refer to (→4.2.2).

Settings

In case a single trigger is selected in the tree or the grid area, the details area will additionally contain the **"Settings"** tab.

This tab provides the main functionality for working with Triggers. Here you can define the functions and operators for the formula of your trigger and what numbers and/or signals they should be applied to, as well as cycling rates, levels and groups.

As this tab is different for each trigger category, it will be explained in the respective section for each trigger category:

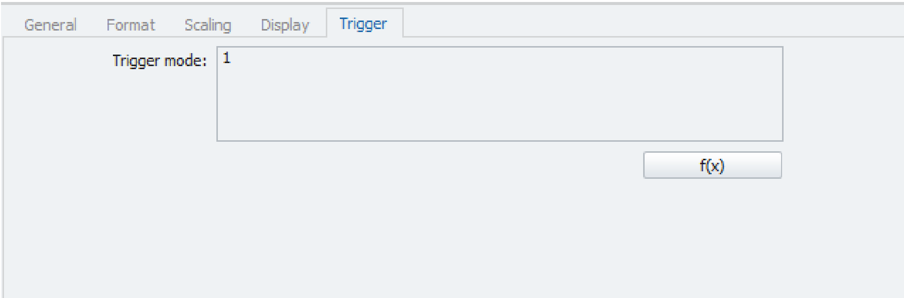


- Standard Triggers (→6.5)
- Level Triggers (→6.6)
- Cyclic Triggers (→6.7)
- Trigger groups (→6.8)

6.5 Standard Triggers

For standard triggers, only an activation condition is specified. If the condition is met, the trigger is set; once it is no longer met, the trigger is reset. This makes the standard trigger a simple and quick way to define a trigger. In the following will be explained how work with "Standard Triggers".

- **Trigger mode**
  This field allows you to manually enter a formula for your trigger condition.



- Formula editor

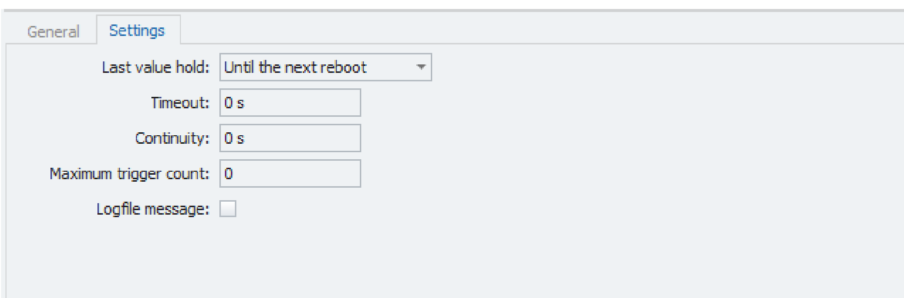
  Although there is the possibility to manually define formulas, most of the time it will be more convenient to define formulas using the **"Formula editor"**. To do so, please refer to the section **"Formula editor"** (→5.15.4

6.6 Level Triggers

A "Level Trigger" is an event with multiple levels of conditions, which have to be set consecutively, in a specified order. As trigger condition for one level, you can either define a single condition, which if met activates the level, and once it is no longer met, resets the trigger. Or else two conditions are defined, one to activate the level (set condition) and one to reset the trigger (reset condition). The program always checks the reset condition of the current level and the set condition of the next level, and reacts accordingly.
The trigger is set once the highest defined level is reached, and remains set until the reset condition of this level is met.
In the following will be explained how to work with "Level Triggers".

Settings

If a "Level trigger" is selected in the tree or the grid are, the settings tab allows for settings regarding the general behaviour of a level trigger.



- Last value hold

  Sustains the trigger counter, i.e. the number of activated triggers, even beyond logger restart, at which point their count continues.

- Timeout

Timeout (in ms), after which the trigger is reset, even if neither the highest level is reached nor another reset condition is met.

- Continuity
  Minimum duration of the trigger condition (in ms) before the trigger is activated.

- Maximum trigger count
  Maximum number of activations for this trigger (0 = unlimited)

- Logfile message
  If activated, an information on this trigger event will be written to the logfile.

Adding trigger levels

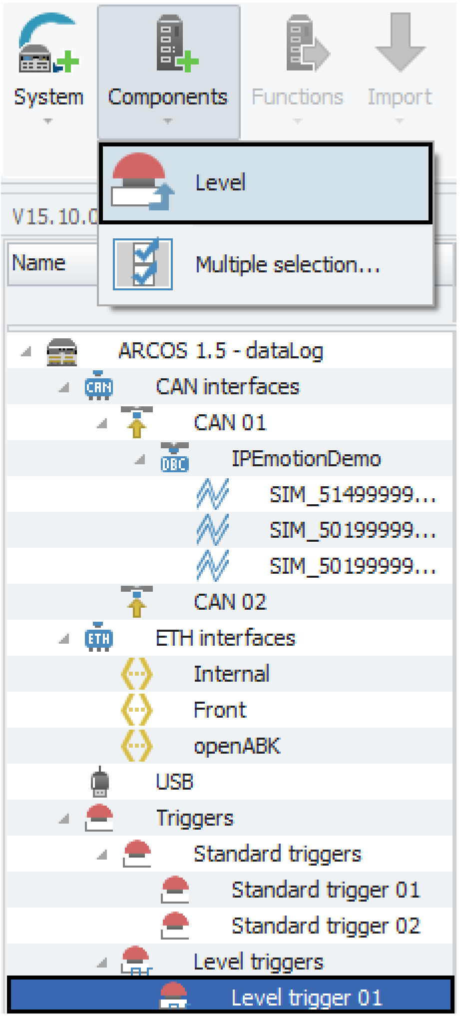
Working with a "Level trigger", allows you to add additional trigger levels, for which you can later define trigger conditions and settings. To add a trigger level, select the desired "Level trigger" in the tree, click the "Components" button of the Ribbon and choose "Level".

An overview of all Levels of a "Level trigger" will be presented in the grid area of the respective "Level trigger".

Level xx

The settings for each separate level of a "Level trigger"can be accessed by selecting the desired Level in the grid area and then navigating to the "Settings" tab in the details area.

- Priority
  The priority defines in which order the single levels of a "Level trigger" have to be met. A "Level trigger" will always ascend in priority starting from "Priority 1" to "Priority 2" and so on until the final defined priority is met or a reset condition is met.
- Mode
  Define the Mode of this trigger level.
  - **"Inactive"** means, this trigger level is deactivated.
  - **"Reset is inverted set"** meanst, this trigger level is activated and will be reset, as soon as the defined set condition is no longer met.
  - **"Set- and Reset-condition"** means that you can define a set condition as well, as a reset condition.
- Set condition
  Formula for the condition to activate this level.
  The result of the formula must always be 0 (not met) or 1(met), apart from that creating a formula is described in section (→5.15).

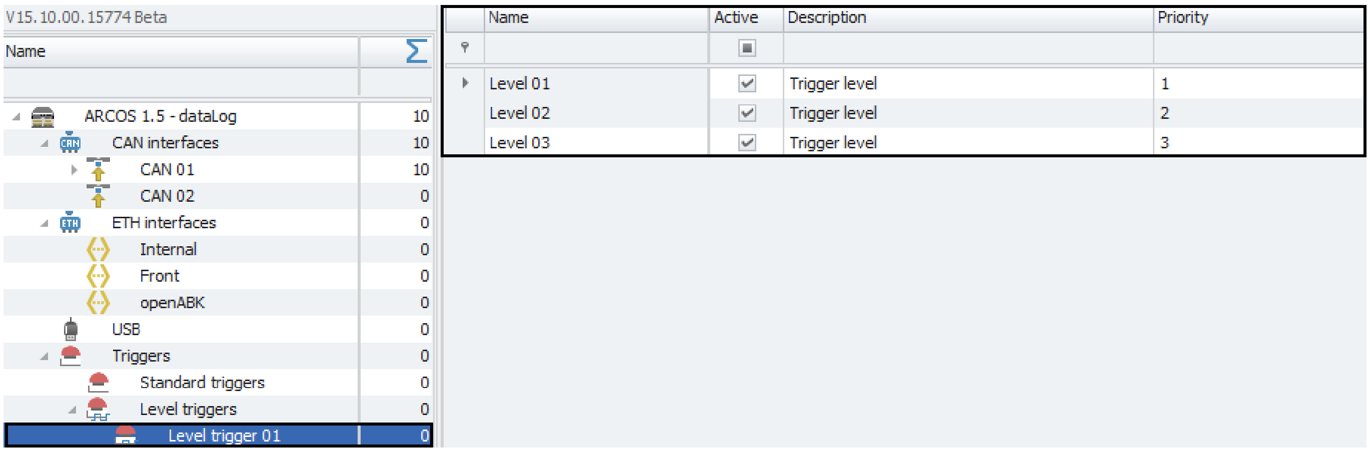


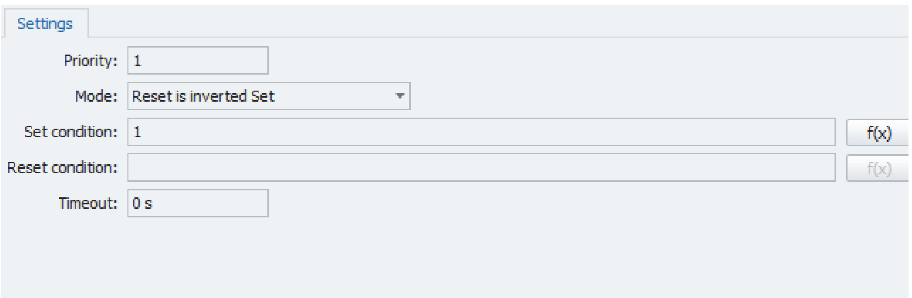
- Reset condition
  Formula for the condition to reset the trigger from this level. Once the reset condition is met, the entire trigger is reset and must therefore run through all levels again. If reset term =1 is set for the highest level, the trigger is immediately reset, i.e. only a single trigger impulse is generated. If no resetterm is specified, the end of the set condition is automatically used as the reset condition. This can be prevented by setting =0. The result of the formula must always be 0 (not met) or 1(met), apart from that creating a formula is described in section (→5.15).

- Timeout
  Define a level timeout after which the trigger is reset, regardless of whether the reset condition has been mat or the highest level has been reached.

6.7 Cyclic Triggers

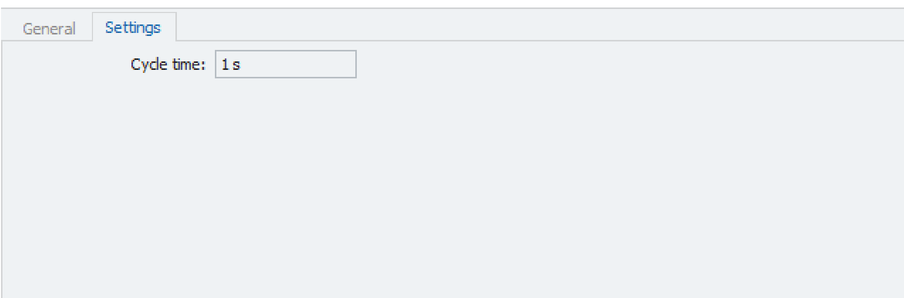
A "Cyclic Trigger" is an event, that is not defined by a formula. Its only condition is the cycle time to which it is set. According to this time, the trigger will be set periodically. In the following will be explained how to work with "Level Triggers".





Settings

The “Settings” tab in the details area of the Cyclic Trigger allows you to set the Cycle time according to which the trigger will be set periodically.

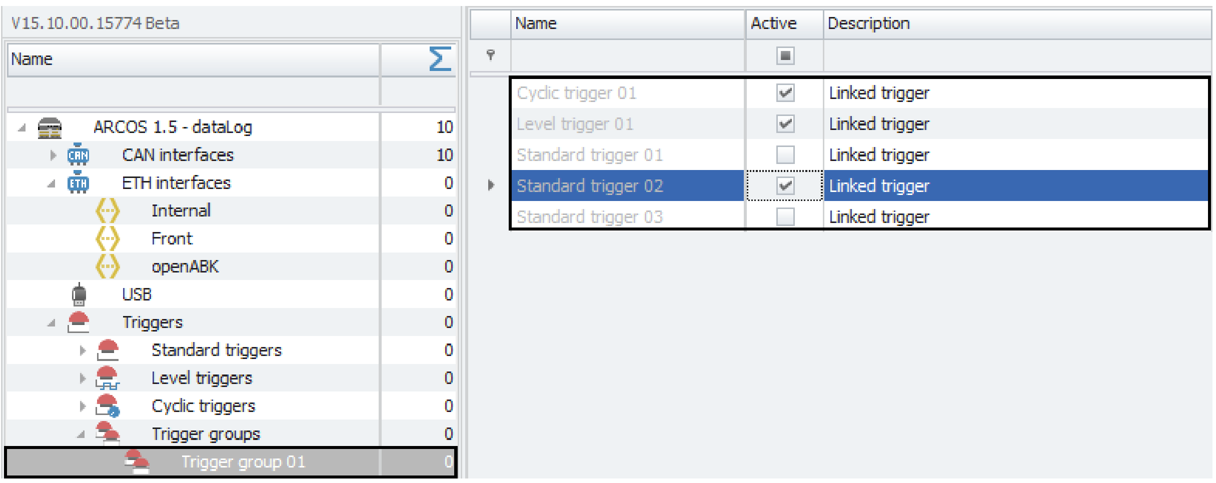


6.8 Trigger groups

"Trigger groups" allow you to combine two or more existing triggers and thus create a new trigger condition. Source trigger signals can be combined via "disjunction (or)" or "conjunction (and)".

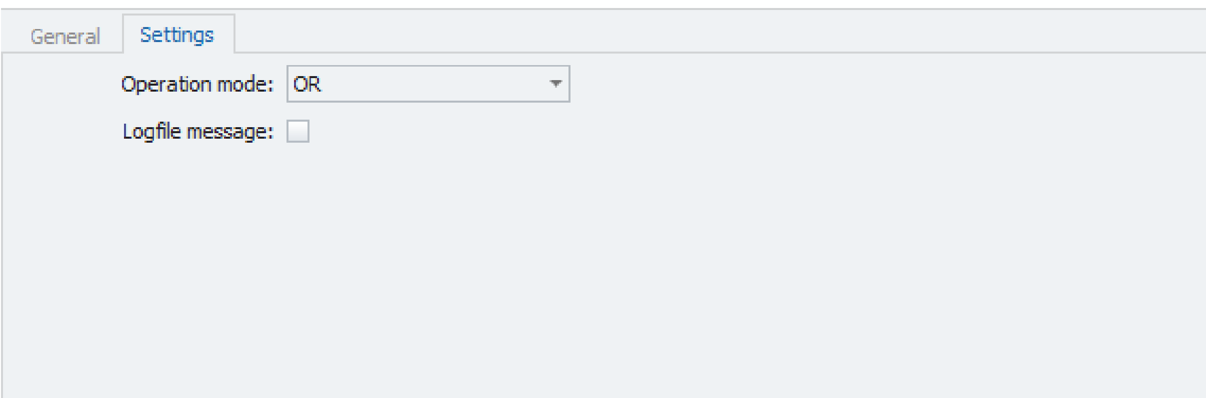
Selecting Triggers

In order for a "Trigger group" to function you will need to select at least two existing triggers to be part of the "Trigger group". To do so, select the "Trigger group" you wish to work with in the measurement task tree.
This will present you with an overview of all available triggers in the details area. Just mark active the ones you wish to add to the trigger group.



Settings

en The "Settings" tab in the details area of your "Trigger group" allows you to set the operation mode of the group and whether a logfile should be created.



- Operation Mode
  Allows you to set the operation mode of the "Trigger group". You may choose between "AND" or "OR".

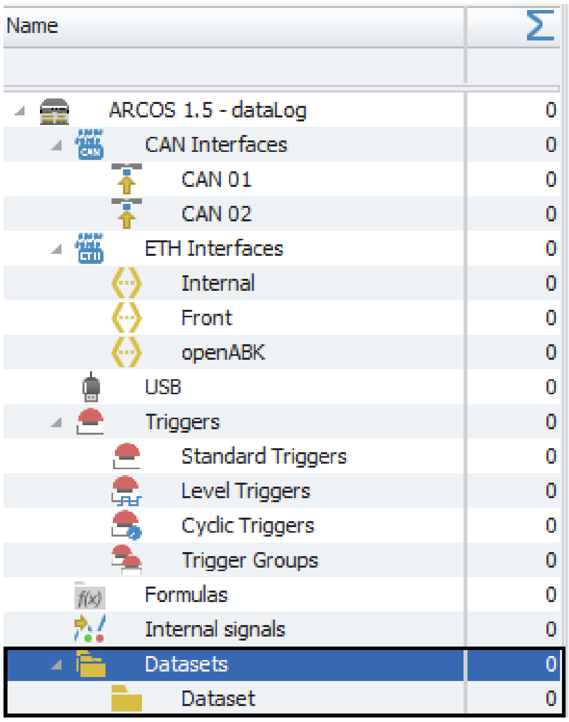
- Logfile message
  If activated, an information on this trigger event will be written to the logfile.

7 Datasets

Data acquired throughout a measurement task will be stored in a dataset by the logger. This chapter will explain how to configure the created dataset.

7.1 Tree elements for Datasets

The tree element "**Datasets**" will contain all the datasets you define for your system. The child element "**Dataset**" then contains the single components of your dataset.

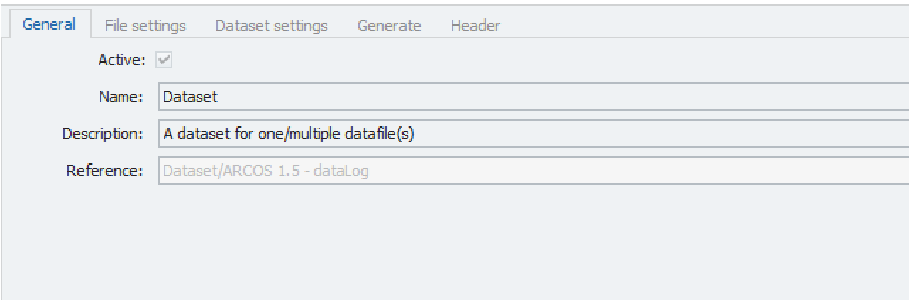


7.2 Details area for Datasets

This section contains settings regarding the overall behaviour of your dataset. These settings are global and will affect all components of your dataset.

In case the parent tree element **"Dataset"** selected, the details area will only show the "General" tab. Please refer to (→4.2.2).

In case the child element **"Dataset"** is selected, the details area will contain additional tabs which will be explained in the following.

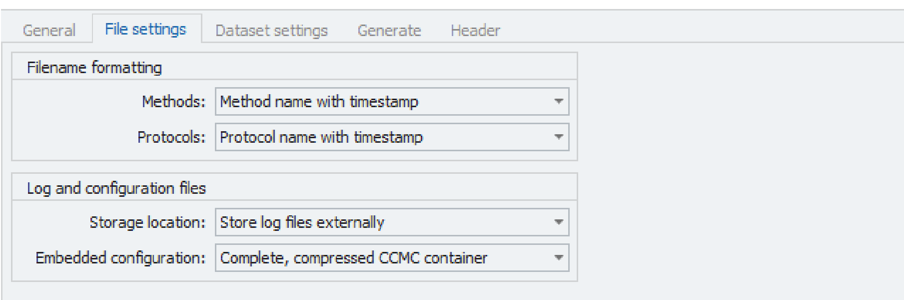


General

Please refer to (→4.2.2).

File settings

This tab contains settings regarding filename and location.

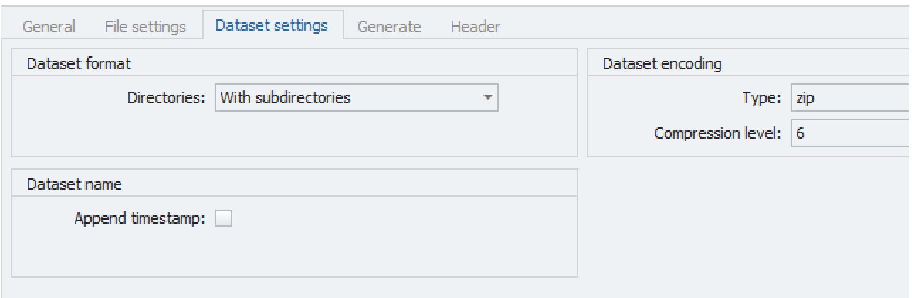


- Filename formatting
  This field allows you to define the filename formatting of the final dataset file. Both methods and protocols have to possible formatting options:
  - Method/Protocol name with timestamp: protocol name with timestamp (date and time)(default setting)
  - Timestamp, trigger name and counter: Appends the name of the activating trigger and its number/counter to the right of the timestamp in the file name.

- **Log and configuration files** This field allows you to set whether your dataset file wil be stored externally or internally and what type of configuration to embed. There are four types of configuration available:
  - Complete, compressed CCMC container: Entire container is stored as is.
  - Uncompressed contents of CCMC container: Content of the container is unpacked and stored.
  - Only CFG: Only the cfg is extracted.
  - Complete, compressed CCMC container and uncompressed contents: Entire container is stored as is and the unpacked content of the container is stored as well.

Dataset settings

This tab allows for settings regarding the dataset itself.



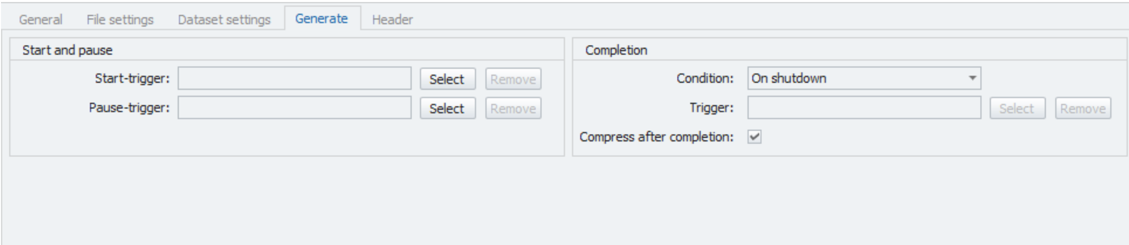
- **Dataset format**
  Define whether the directory structure of data within a dataset contains subdirectories or not.

- **Dataset encoding**
  Set the type of encoding and compression level for the dataset.

- **Dataset name**
  Select, whether a timestamp will be appended to the dataset name.

Generate

Define settings regarding the creation of the dataset.

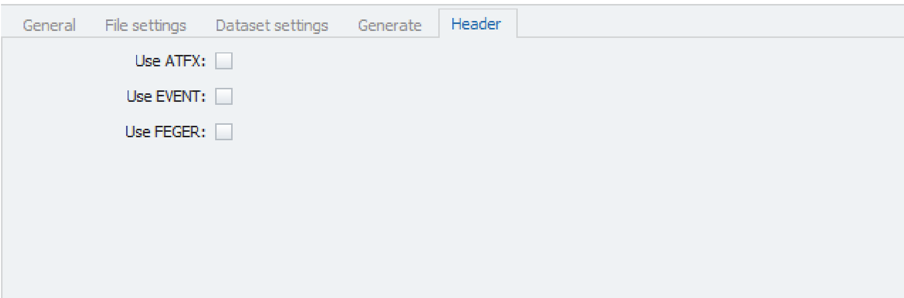


- Start and Pause
  This field allows you, to set triggers to start or pause dataset creation. If you do not set a start trigger, then by default dataset creation is always active during measurement.

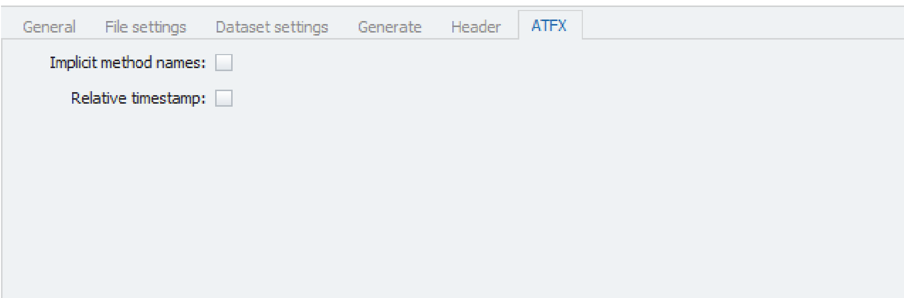
- Completion
  The dropdown menu **"Condition"** allows you to set a condition under which the dataset should be closed and completed. In case your condition for completion requires a trigger, you may choose the trigger in the below field.
  If you choose **"Compress after completion"**, the will be compressed directly after completion. If you do not choose this option, the dataset will be compressed before transfer.

Header

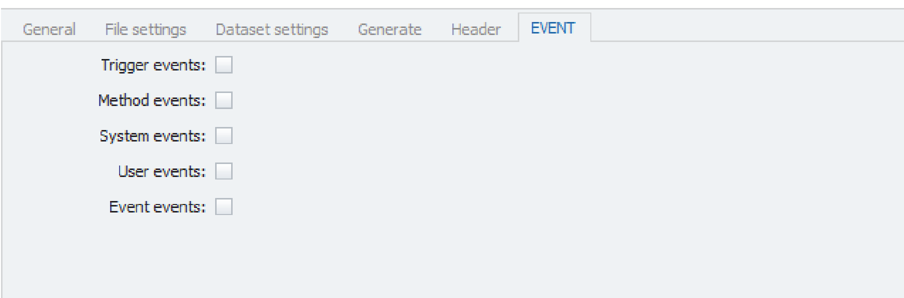
When storing the data, the logger saves general information about the measurement (identification, comments, start, stop) and information about the data files (name, structure) in a header file. For additional functionality a various headers with specific options can be attached. The options of each activated header-type will be accessible in a separate tab with the header-type's name.
Multiple header-types can be activated at the same time.



- ATFX



- EVENT

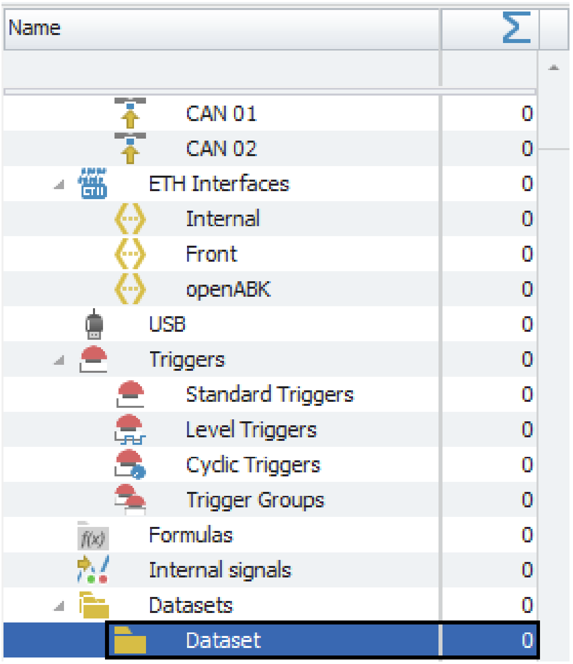


- FEGER

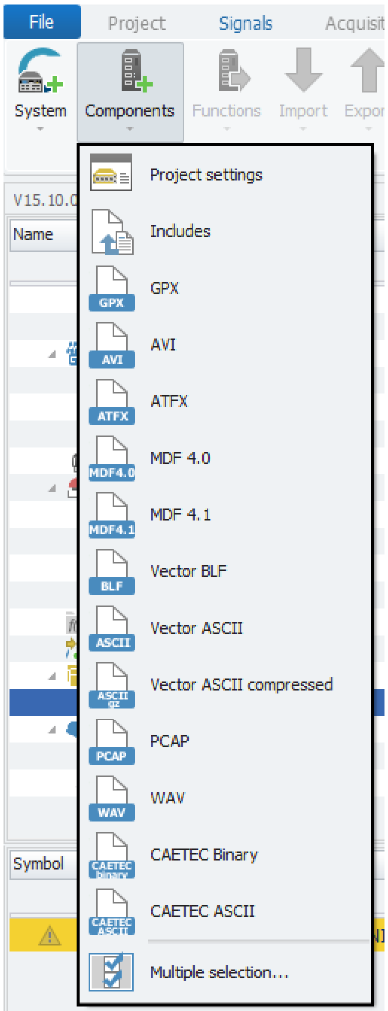
7.3 Setting up a dataset

A dataset can be set up using different filetypes, according to what information you want it to contain. Each filetype you include in your dataset has specific functionalities and for each included filetype will later be included a file in the exported dataset with its preivously configured name and the according filetype extension.

In order to include a filetype in your dataset, select the tree element "Dataset".



Then click the "Components" button in the Ribbon and choose the desired filetype from the resulting menu.



The following sections will explain in detail the different filetypes.

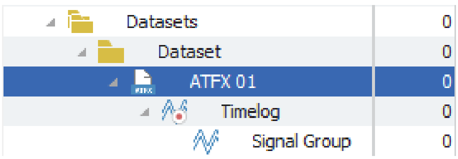
7.4 ATFX

The “ATFX” filetype is meant for recording of signals. It is compatible with all signals that produce values that can be represented on a 2-D graph. It is not compatible with video or audio files.
The recorded signal values will be represented in a timelog.

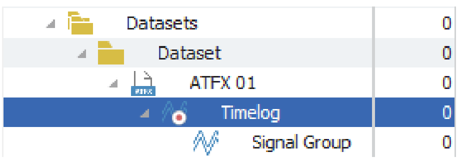
7.4.1 Tree elements for ATFX

Including the “ATFX” filetype in your dataset will add three new child elements to your tree element “Dataset”:

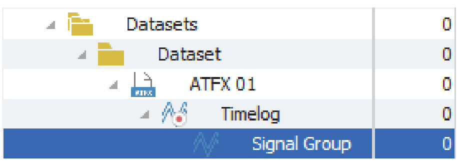
- ATFX xx
  This element represents the ATFX file, which will later be included in your exported dataset. You may add multiple files of the same filetype.



- Timelog
  This element represents the timelog for recording signal values.



- Signal Group
  The signal is a group of all the signals that the timelog will record and that will ultimately be included in the exported dataset.



7.4.2 Grid area for ATFX

If the “Signal Group” is selected in the Measurement task tree, the grid area will show an overview of the signals included in the “Signal Group”

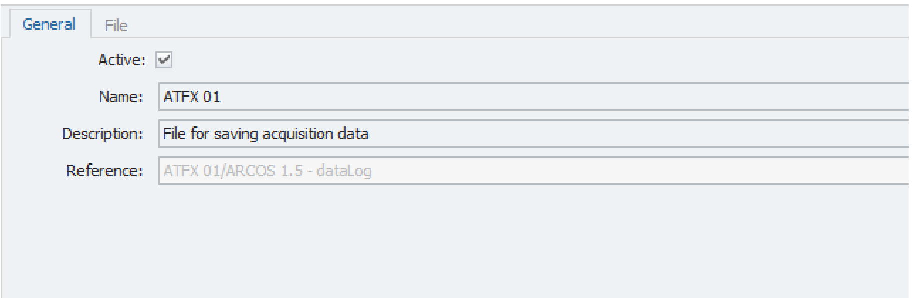
7.4.3 Details area for ATFX

The details area contains settings for the behaviour of your ATFX file, timelog or signal group.

7.4.3.1 ATFX file

General

This tab provides general settings for the selected ATFX file.



- Active
  Allows you to activate or deactivate the selected file.

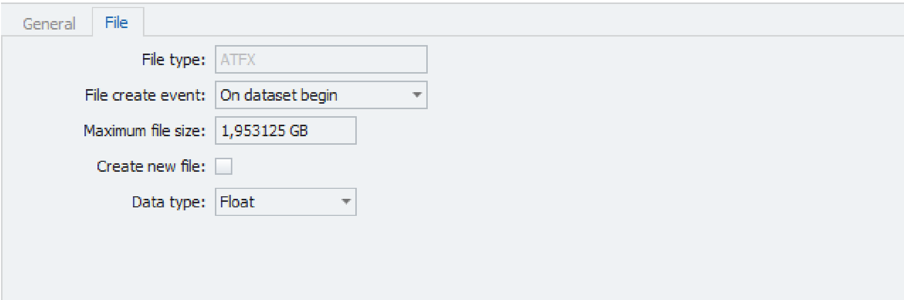
- Name
  Give a user-defined Name to the selected formula/signal.

- Description
  Give a user-defined description to the selected formula/signal.

- Reference
  This field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed.

File

This tab provides settings regarding the creation of the file.



- **File type**
  Tells you the type of the created file.
- **File create event**
  Define, when the ATFX file should be created. There are four possibilities:

File create event	Characteristics
On dataset begin	The file will be created everytime, the dataset is started or restarted after a pause. This may result in a splitting of the current ATFX file into multiple files, as a new file is created for each time a dataset ist started.
On recording start	The file will be created once at logger start.
On trigger	The file will be created on a trigger and record for a user defined duration. These settings can be defined in the timelog settings (→ 7.4.3.2). This will result in a splitting of the current ATFX file into multiple files, as a new file is created for each timethe trigger is set.
On first trigger	The file will be created once, when the defined trigger is set for the first time and record for a user defined duration. These settings can be defined in the timelog settings → 7.4.3.2). Each following thime the trigger is set, the data will be written in the same previously created file. Therefore there will only be one file.

- **Maximum file size**
  Define the maximum file size. It is recommended not to raise the maximum file size above 2GB, as some third party analysis tools cannot handle files, that are larger.
- **Create new file**
  If this box is marked active, a new file will be created, if the current file exceeds the maximum file size.
- **Data type**
  This dropdown menu allows you to switch the data type between float or double.

7.4.3.2 ATFX Timelog

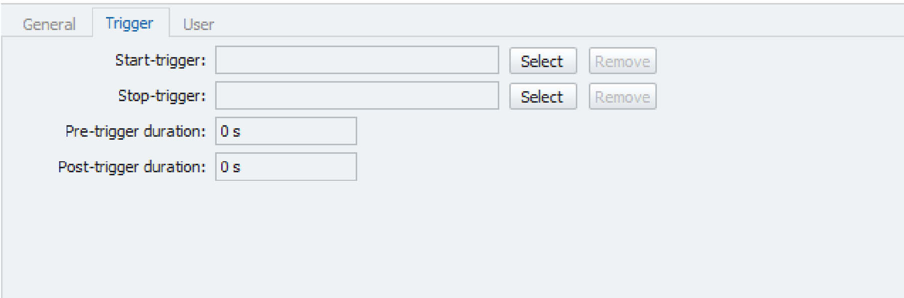
General

This tab provides general settings for the selected ATFX timelog.

- Active
  Allows you to activate or deactivate the selected file.
- Name
  Give a user-defined Name to the selected file.
- Description
  Give a user-defined description to the selected file.
- Reference
  This field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed.

Trigger

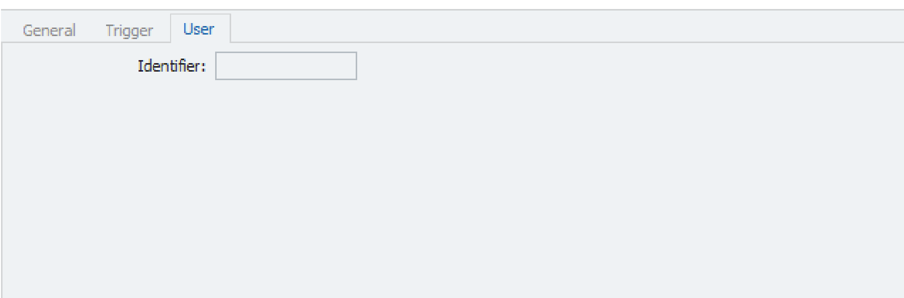
This tab provides settings regarding the trigger for the start and stop of the timelog. When the timelog is started, all signals contained in the signal group will be stored to the ATFX file according to their settings.
Furthermore will this trigger provoke the creation of the ATFX file, if you have choosen either "On trigger" or "On first trigger" as "File create event".

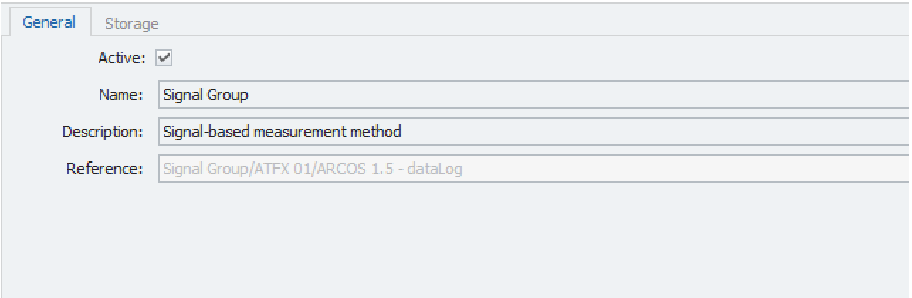


- Start-trigger
  Define a trigger, that will start the timelog.
- Stop-trigger
  Define a trigger, that will stop the timelog.
- Pre-trigger duration
  Pre-trigger duration allows you to define, how long before the start trigger was set, the timelog will start.
- Post-trigger duration
  Post-trigger duration allows you to define, how long after the start trigger was set, the timelog will stop.

User

The field "Identifier" allows you to give a user identifier to the timelog. It does not have any effect other than helping the user identifiy a specific timelog.





7.4.3.3 ATFX Signal Group

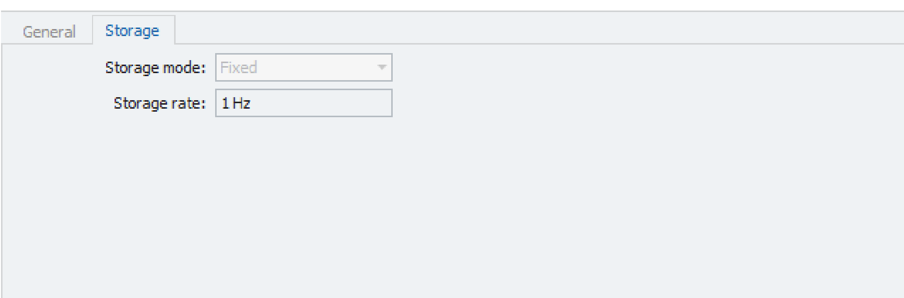
General

This tab provides general settings for the selected ATFX timelog.

- Active
  Allows you to activate or deactivate the selected file.
- Name
  Give a user-defined Name to the selected formula/signal.
- Description
  Give a user-defined description to the selected formula/signal.
- Reference
  This field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed.

Storage

This tab allows for setting regarding the storage of the contained signals inside the ATFX file.

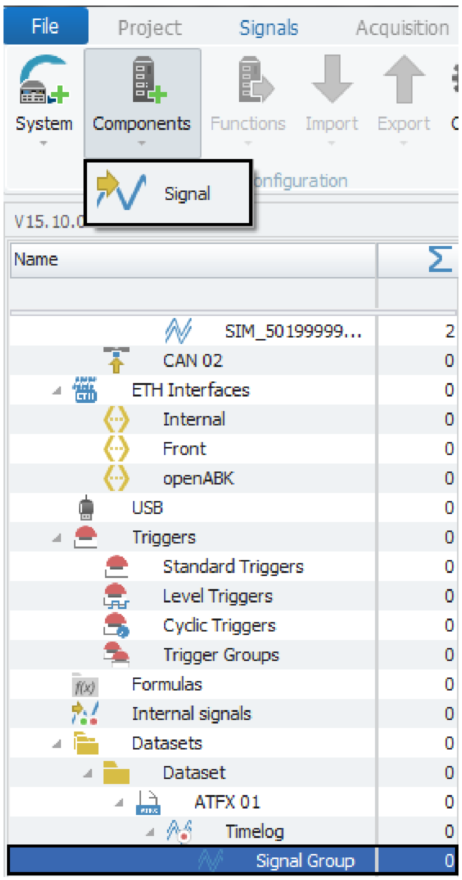


- Storage mode
  The storage mode in ATFX is fixed. It cannot be changed.

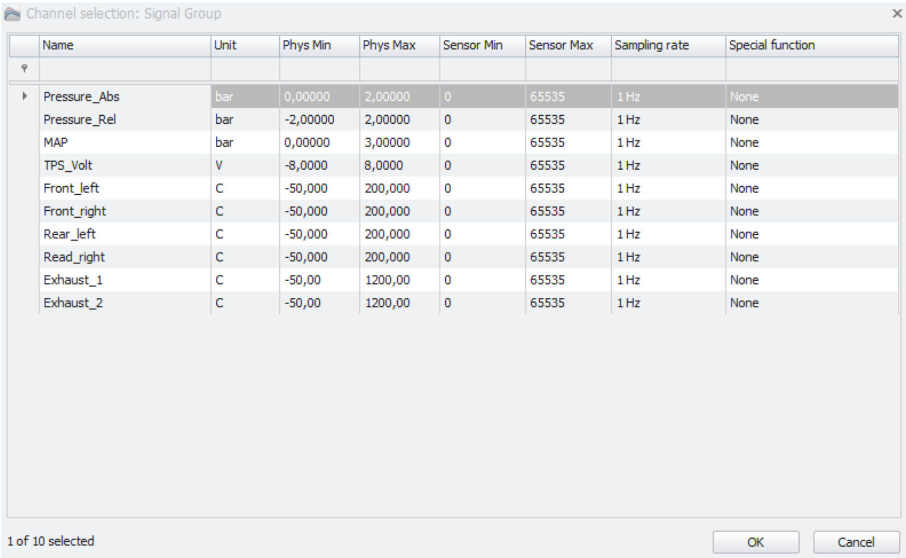
- Storage rate
  The storage rate defines how often the signals contained in this "Signal Group" will be stored.

7.4.4 Working with Signal Groups for ATFX

For filetypes intended for signal recording such as ATFX, MDF 4.0 and MDF 4.1, signals need to be included in a "Signal Group" belonging to the timelog, to which you would like the signal to be included. The signals, that are included in a "Signal Group" can then be stored.
To achieve this, select the "Signal group" in the "Measurement task tree", click the "Components" button in the Ribbon and choose "Signal".



In the following window you may choose all the available signals, that you wish to include within this group and confirm by clicking "OK".



ATFX can only contain one "Signal Group" and this "Signal Group" can only have one storage rate. That means, all signals, that are contained in the same "Signal Group", will be stored with the same storag rate.

7.5 MDF 4.0

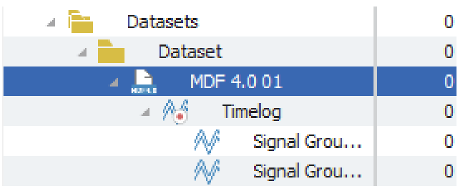
The “MDF 4.0” filetype is meant for recording of signals. It is compatible with all signals that produce values that can be represented on a 2-D graph. It is not compatible with video or audio files.
The recorded signal values will be represented in a timelog.

> MDF 4.0 and MDF 4.1 files can have (Other than ATFX) multiple “Signal Groups” and some of those groups allow for include signals to be stored in various rates.

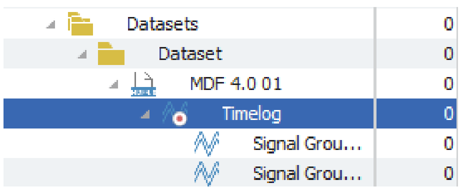
7.5.1 Tree elements for MDF 4.0

Including the “MDF 4.0” filetype in your dataset will add three new child elements to your tree element “Dataset”:

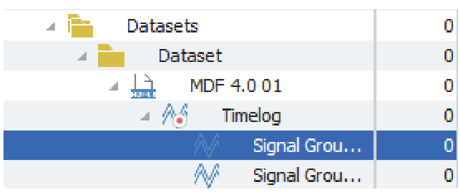
- MDF 4.0 xx
  This element represents the MDF 4.0 file, which will later be included in your exported dataset. You may add multiple files of the same filetype.



- Timelog
  This element represents the timelog for recording signal values.



- Signal Group
  The signal is a group of all the signals that the timelog will record and that will ultimately be included in the exported dataset.



7.5.2 Grid area for MDF 4.0

If the “Signal Group” is selected in the Measurement task tree, the grid area will show an overview of the signals included in the “Signal Group”

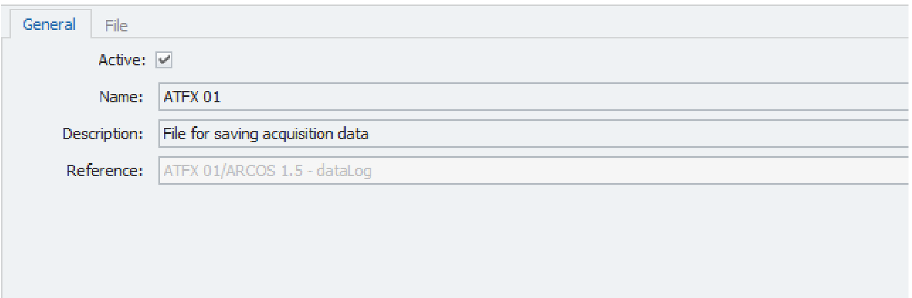
7.5.3 Details area for MDF 4.0

The details area contains settings for the behaviour of your MDF 4.0 file, timelog or signal group.

7.5.3.1 MDF 4.0 File

General

This tab provides general settings for the selected MDF 4.0 file.



- Active
  Allows you to activate or deactivate the selected file.

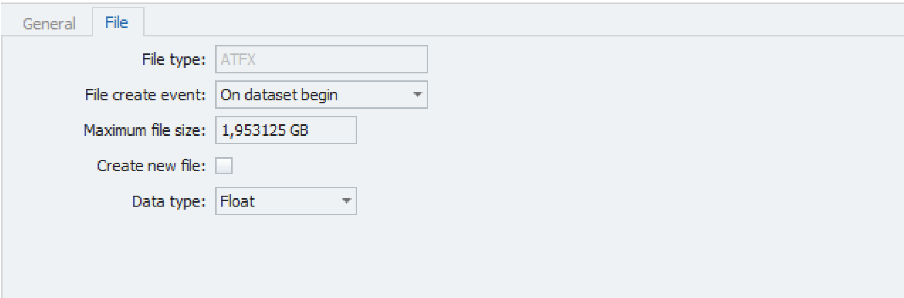
- Name
  Give a user-defined Name to the selected file.

- Description
  Give a user-defined description to the selected file.

- Reference
  This field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed.

File

This tab provides settings regarding the creation of the file.



- **File type**
  Tells you the type of the created file.
- **File create event**
  Define, when the MDF 4.0 file should be created. There are four possibilities:

File create event	Characteristics
On dataset begin	The file will be created everytime, the dataset is started or restarted after a pause. This may result in a splitting of the current MDF 4.0 file into multiple files, as a new file is created for each time a dataset ist started.
On recording start	The file will be created once at logger start.
On trigger	The file will be created on a trigger and record for a user defined duration. These settings can be defined in the timelog settings ($\rightarrow$ 7.5.3.2). This will result in a splitting of the current MDF 4.0 file into multiple files, as a new file is created for each timethe trigger is set.
On first trigger	The file will be created once, when the defined trigger is set for the first time and record for a user defined duration. These settings can be defined in the timelog settings $\rightarrow$ 7.5.3.2). Each following thime the trigger is set, the data will be written in the same previously created file. Therefore there will only be one file.

- **Maximum file size**
  Define the maximum file size. It is recommended not to raise the maximum file size above 2GB, as some third party evaluation tools cannot handle files, that are larger.
- **Create new file**
  If this box is marked active, a new file will be created, if the current file exceeds the maximum file size.
- **Data type**
  This dropdown menu allows you to switch the data type between float or double.

7.5.3.2 MDF 4.0 Timelog

General

This tab provides general settings for the selected MDF 4.0 timelog.

- Active
  Allows you to activate or deactivate the selected file.

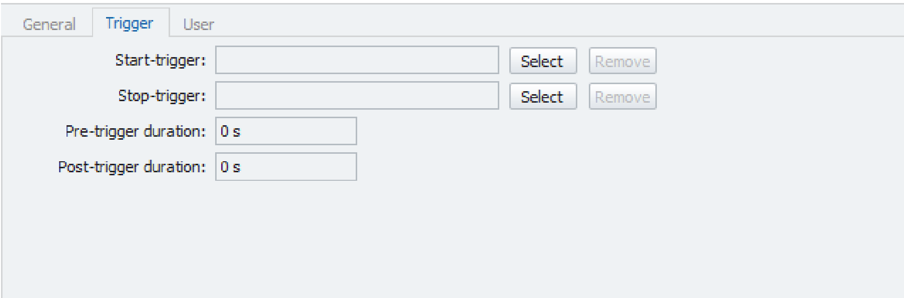
- Name
  Give a user-defined Name to the selected file.

- Description
  Give a user-defined description to the selected file.

- Reference
  This field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed.

Trigger

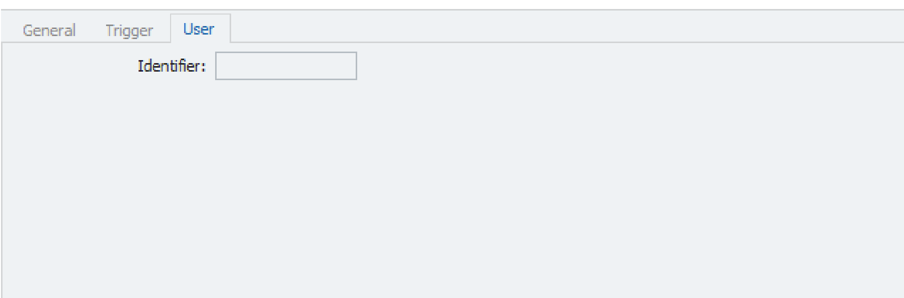
This tab provides settings regarding the trigger for the start and stop of the timelog. When the timelog is started, all signals contained in the signal group will be stored to the MDF 4.0 file according to their settings.
Furthermore will this trigger provoke the creation of the MDF 4.0 file, if you have choosen either "On trigger" or "On first trigger" as "File create event".

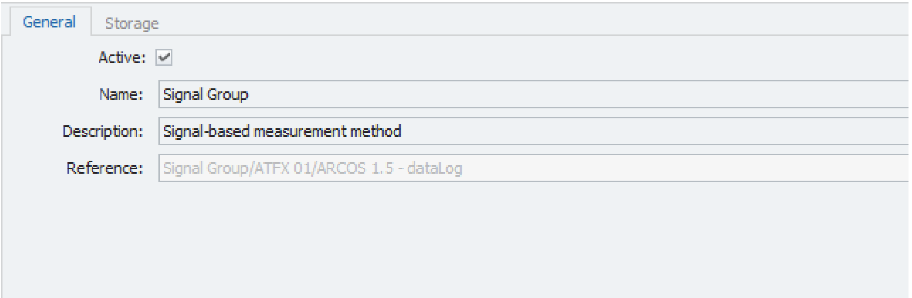


- Start-trigger
  Define a trigger, that will start the timelog.
- Stop-trigger
  Define a trigger, that will stop the timelog.
- Pre-trigger duration
  Pre-trigger duration allows you to define, how long before the start trigger was set, the timelog will start.
- Post-trigger duration
  Post-trigger duration allows you to define, how long after the start trigger was set, the timelog will stop.

User

The field "Identifier" allows you to give a user identifier to the timelog. It does not have any effect other than helping the user identifiy a specific timelog.





7.5.3.3 MDF 4.0 Signal Group

General

This tab provides general settings for the selected MDF 4.0 timelog.

- Active
  Allows you to activate or deactivate the selected file.

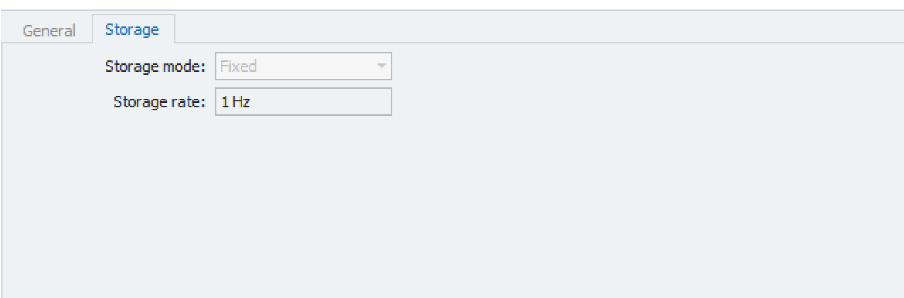
- Name
  Give a user-defined Name to the selected formula/signal.

- Description
  Give a user-defined description to the selected formula/signal.

- Reference
  This field serves as the tree element´s unique identifier inside the measurement task tree. It cannot be changed.

Storage

This tab allows for setting regarding the storage of the contained signals inside the MDF 4.0 file.



- Storage mode

Storage mode	Characteristics
Fixed	The rate, at which the signals included in the "Signal Group" will be stored, is the same for all included signals. You may set the rate below at "Storage Rate
From channel	The rate, at which the signals included in the "Signal Group" will be stored, is the same as each signal's source channel. This may result in a "Signal Gruop" with different storage rates for different signals, according to their sourche channel's sampling rate.
Individual	The rate, at which the signals included in the "Signal Group" will be stored, can be individually set for each signal. This may result in a "Signal Gruop" with different storage rates for different signals.

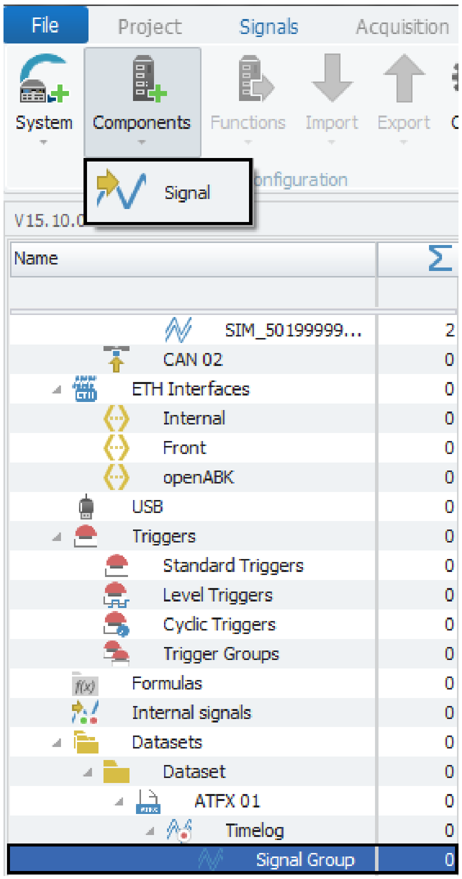
- Storage rate
  The storage rate defines how often the signals contained in this "Signal Group" will be stored.

7.5.4 Working with Signal Groups for MDF 4.0

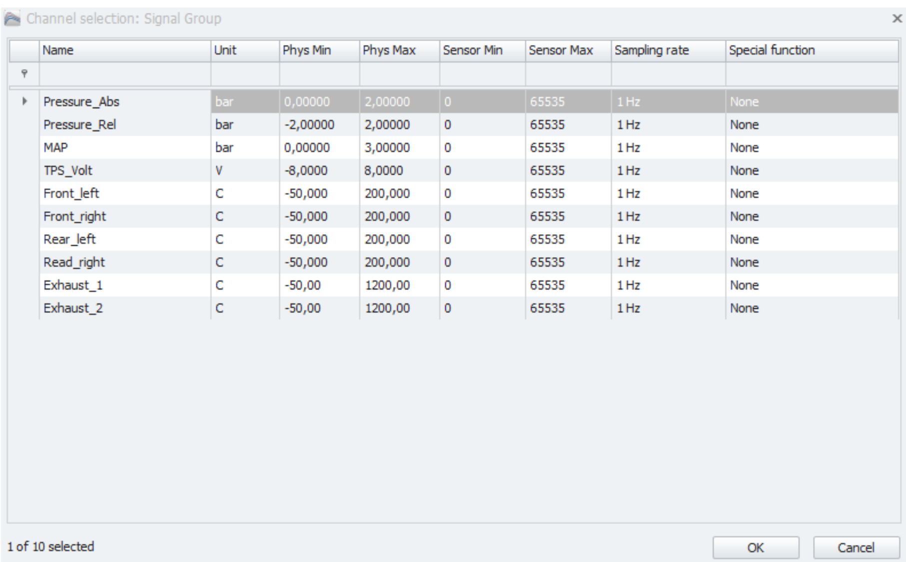
For filetypes intended for signal recording such as ATFX, MDF 4.0 and MDF 4.1, signals need to be included in a "Signal Group" belonging to the timelog, to which you would like the signal to be included. The signals, that are included in a "Signal Group" can then be stored.
Other than for ATFX files, MDF 4.0/4.1 files support multiple "Signal Groups" and storage modes with different characteristics and therefore allow for a more flexible storage of your data. For details ont the different storage modes please refer to (→ 7.5.3.3).

This allows for example to create one "Signal Group" with a fixed storage rate of 10Hz, one that stores signals according to their channel's sampling rate and one that has an individual storage rate for each signal.

To include signals within one group, select the "Signal group" in the "Measurement task tree", click the "Components" button in the Ribbon and choose "Signal".



In the following window you may choose from all the available signals the ones, that you wish to include within this group and confirm by clicking "OK".



Name	Unit	Phys Min	Phys Max	Sensor Min	Sensor Max	Sampling rate	Special function
Pressure_Abs	bar	0,00000	2,00000	0	65535	1 Hz	None
Pressure_Rel	bar	-2,00000	2,00000	0	65535	1 Hz	None
MAP	bar	0,00000	3,00000	0	65535	1 Hz	None
TPS_Volt	V	-8,0000	8,0000	0	65535	1 Hz	None
Front_left	C	-50,000	200,000	0	65535	1 Hz	None
Front_right	C	-50,000	200,000	0	65535	1 Hz	None
Rear_left	C	-50,000	200,000	0	65535	1 Hz	None
Read_right	C	-50,000	200,000	0	65535	1 Hz	None
Exhaust_1	C	-50,00	1200,00	0	65535	1 Hz	None
Exhaust_2	C	-50,00	1200,00	0	65535	1 Hz	None

Even though the Plugin allows for different storage rates within one "Signal Group" in the case of the storage modes "From channel" and "Individual", technically each rate will be exported as a separate group in the loggerconfig.
That means, if you have a "Signal Group" called **"examplegroup"** with three different storage rates (1Hz, 10Hz, 100Hz), at export will be created three groups named "examplegroup_1Hz", "examplegroup_10Hz" and "examplegroup_100Hz" (namemethod: signalgroup_storagerate).

7.6 MDF 4.1

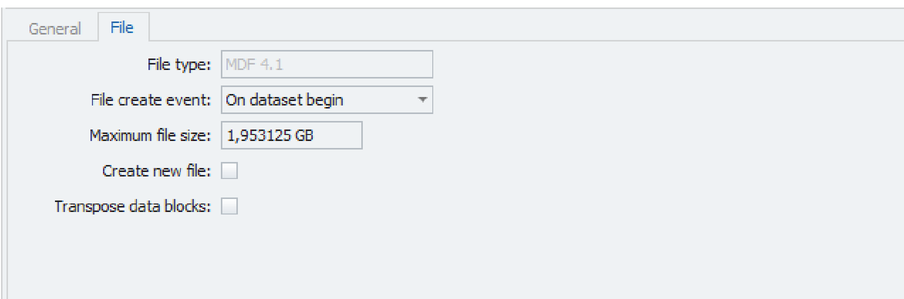
MDF 4.1 can largely be treated the same as MDF 4.0, with two main differences:

- MDF 4.1 supports file compression
- MDF 4.1 supports video attachments

7.6.1 File compression in MDF 4.1

MDF 4.1 files are by default compressed. The standard compression rate is 7-10.

For further compression MDF 4.1 offers the option "Transpose data blocks". To access this option, select the tree element "MDF 4.1 xx" and open the "File" tab in the details section.

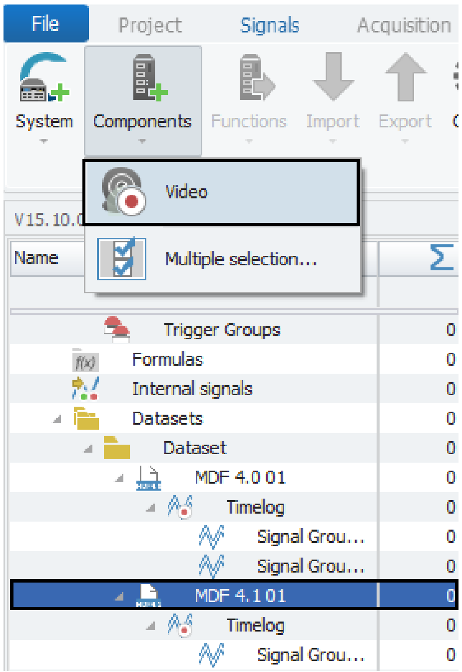


7.6.2 Video attachments in MDF 4.1

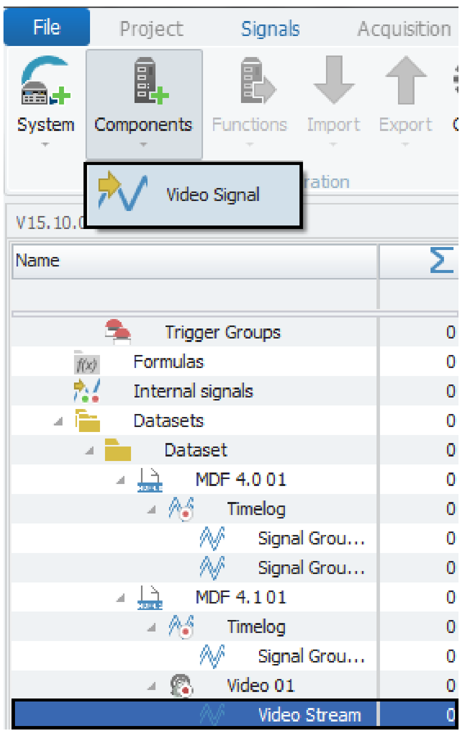
When working with an MDF 4.1 filesystem, you can attach "Video Streams" to your timelog. As the video is only an attachment, it can not have its own triggers nor duration settings. The video-file will be attached in the *.avi format and have the same filename as the MDF file it belongs to. The MDF file knows about the existence and automatically synchonizes the timelog and video.

7.6.2.1 Attaching a video

In order to attach a video, you will first need to add the "Video" component to your MDF 4.1 file.
Select the tree element "MDF 4.1 xx", click the "Components" button in the Ribbon and then choose "Video".



Once the video component has been added to your MDF file, you can choose a video signal, that you wish to attach.
Select the new tree element "Video stream", click the "Components" button in the Ribbon and then choose "Video signal".



In the resulting window you will be presented with an overview of all the available video signals. Choose the one, you wish to attach, and confirm with “OK”.

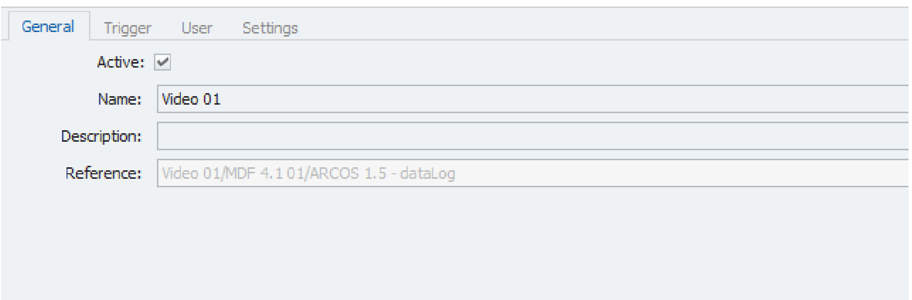
It is possible to attach multiple videos by adding an additinal “Video Stream”. To do so, select the “Video” tree element, click on the “Components” button in the ribbon and choose “Video Stream”.

7.6.2.2 Details area vor video in MDF 4.1

There are two relevant settings tabs for video attachments in the details area.

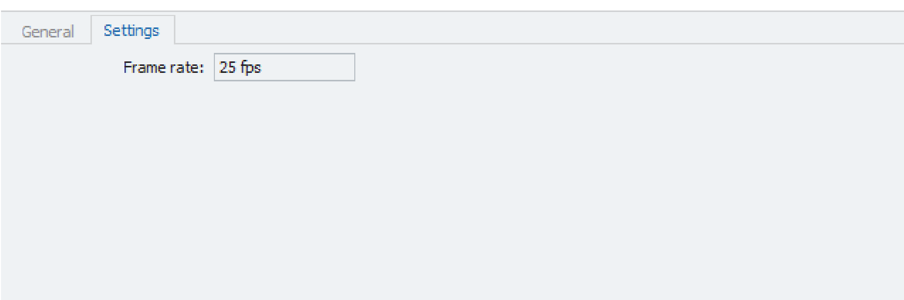
Video xx General

The tab “General” for the tree element “Video xx” allows you to activate or deactivate the video.



Video Stream Settings

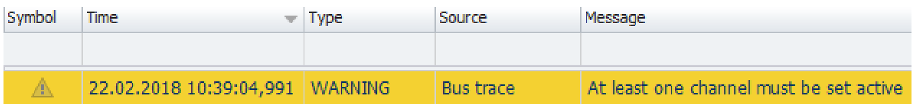
The tab “Settings” for the tree element “Video Stream” allows you to set the framerate for the video.



7.7 Vector BLF / Vector ASCII / Vector ASCII compressed

These three filetypes are equal in fuctionality and differ only in the final exported file. The trace method records all the messages that arrive on the input bus (CAN, LIN, FlexRay). Regardless of the signals defined, all the messages are recorded. Filter rules can be defined to reduce the data volume. A typical trace application is the acquisition of all raw data in order to later evaluate the total traffic on the channel. Unlike most of the other methods, traces are event-oriented. This means the messages are not retrieved from the channels according to a set time pattern, but are recorded as soon as they arrive on the channel. This method accordingly has no parameter for sampling rate.

Including a bus tracing filetype in your datasete will produce a "Warning" message, saying that "at least one channel must be set active".



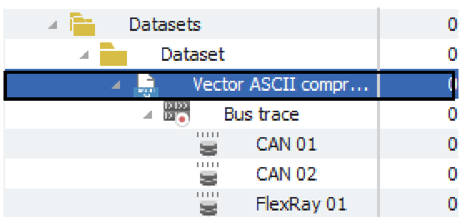
Symbol	Time	Type	Source	Message
⚠	22.02.2018 10:39:04,991	WARNING	Bus trace	At least one channel must be set active

Please refer to the section "Bus trace" (→ 7.7.3.2), in order to activate a channel for tracing.

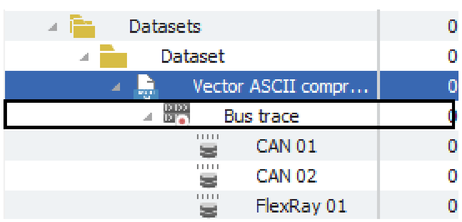
7.7.1 Tree elements for bus tracing

Including a bus tracing filetype in your dataset will add various new child elements to your tree element "Dataset":

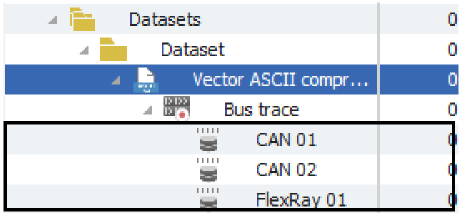
- Vector BLF / Vector ASCII / Vector ASCII compressed xx
  This element represents the bus tracing file, which will later be included in your exported dataset. You may add multiple files of the same filetype.



- Bus trace
  This element represents the "Bus trace" for recording all the traffic on a selected bus channel.

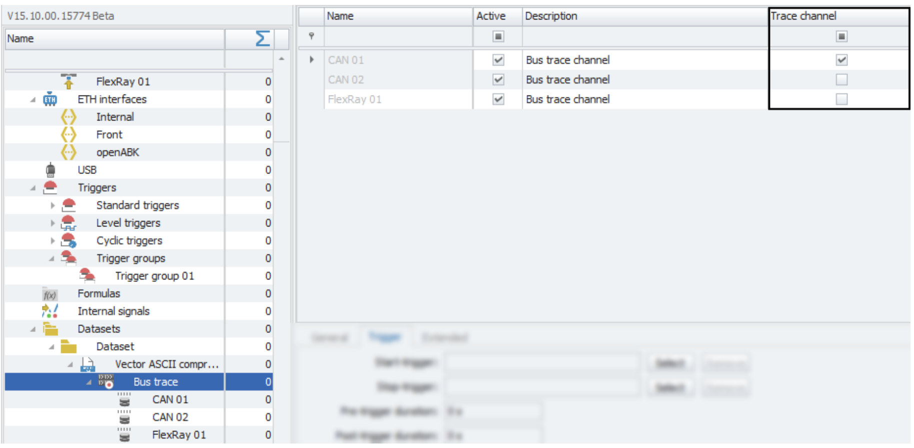


- Bus channels available for tracing
  As child elements to the tree element "Bus trace" will appear all the Bus channels which are currently available for tracing.



7.7.2 Grid area for bus tracing

If the "Bus trace" is selected in the Measurement task tree, the grid area will show an overview of the Bus channels available for tracing.
At least one channel must be marked active for tracing, by ticking the checkbox labeled "Trace channel".



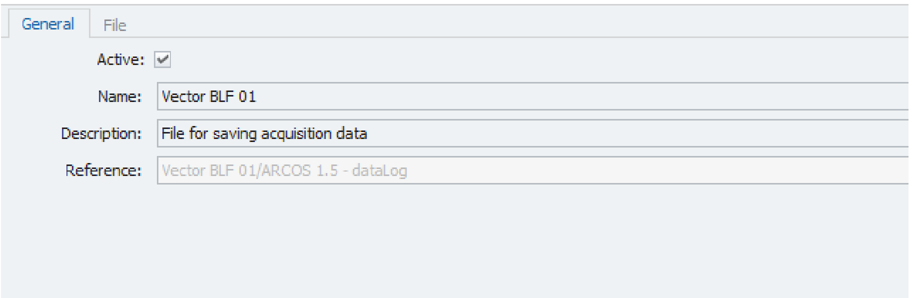
If an ID Filter has been added for a Bus trace channel, selecting this Bus trace channel in the tree will show an overview of the existing ID Filters in the grid area.
For information on working with ID Filters for Bus trace, please refer to the chapter **Bus trace ID Filter** (→7.7.4).

7.7.3 Details area for bus tracing

The details area contains settings for the behaviour of your bus tracing file (Vector BLF / Vector ASCII / Vector ASCII compressed), the "Bus trace" component or a tracable Bus channel.

7.7.3.1 Bus tracing file

General
This tab provides general settings for the selected bus tracing file.



- Active
  Allows you to activate or deactivate the selected file.

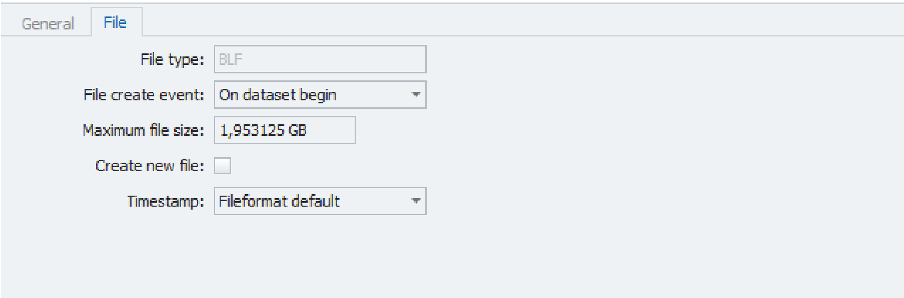
- Name
  Give a user-defined Name to the selected file.

- Description
  Give a user-defined description to the selected file.

- Reference
  This field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed.

File

This tab provides settings regarding the creation of the file.



- File type
  Tells you the type of the created file.
- File create event
  Define, when the bus tracing file should be created. There are four possibilities:

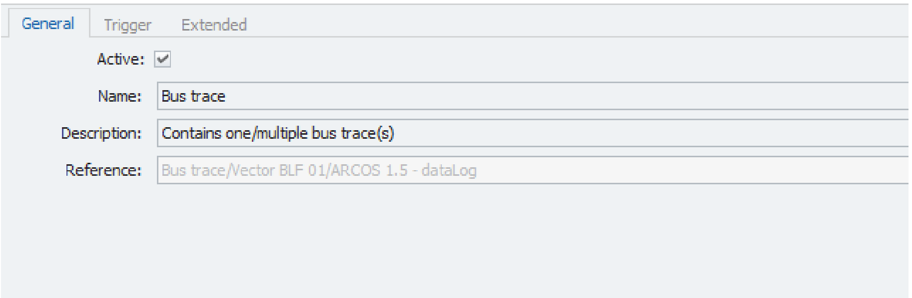
File create event	Characteristics
On dataset begin	The file will be created everytime, the dataset is started or restarted after a pause. This may result in a splitting of the current bus tracing file into multiple files, as a new file is created for each time a dataset ist started.
On recording start	The file will be created once at logger start.
On trigger	The file will be created on a trigger and record for a user defined duration. These settings can be defined in the "Bus trace" settings (→ 7.7.3.2). This will result in a splitting of the current bus tracing file into multiple files, as a new file is created for each timethe trigger is set.
On first trigger	The file will be created once, when the defined trigger is set for the first time and record for a user defined duration. These settings can be defined in the "Bus trace" settings → 7.7.3.2). Each following thime the trigger is set, the data will be written in the same previously created file. Therefore there will only be one file.

- Maximum file size
  Define the maximum file size. It is recommended not to raise the maximum file size above 2GB, as some third party evaluation tools cannot handle files, that are larger.
- Create new file
  If this box is marked active, a new file will be created, if the current file exceeds the maximum file size.
- Timestamp
  This dropdown menu allows you to set the format of the timestamp for the file.

7.7.3.2 Bus trace

General

This tab provides general settings for the selected "Bus trace".



- Active
  Allows you to activate or deactivate the selected file.

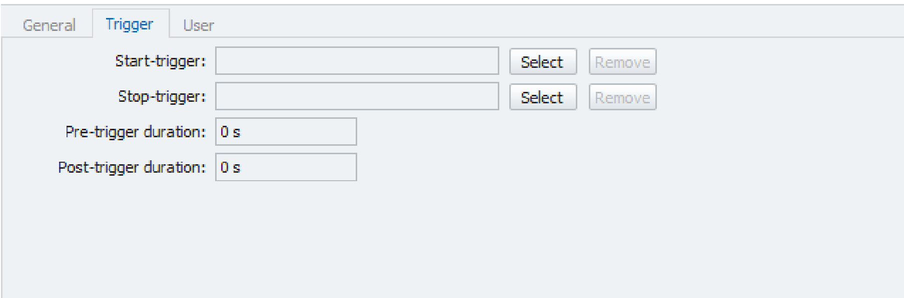
- Name
  Give a user-defined Name to the selected file.

- Description
  Give a user-defined description to the selected file.

- Reference
  This field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed.

Trigger

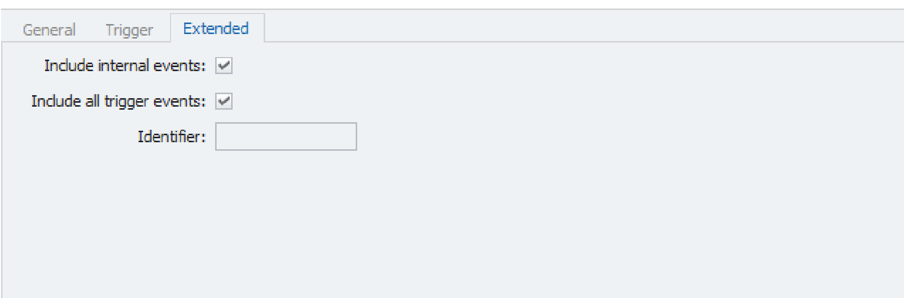
This tab provides settings regarding the trigger for the start and stop of the "Bus trace". When the "Bus trace" is started, all traffic on the channel will be stored to the bus tracing file.
Furthermore will this trigger provoke the creation of the bus tracing file, if you have choosen either "On trigger" or "On first trigger" as "File create event".



- Start-trigger
  Define a trigger, that will start the "Bus trace".
- Stop-trigger
  Define a trigger, that will stop the "Bus trace".
- Pre-trigger duration
  Pre-trigger duration allows you to define, how long before the start trigger was set, the "Bus trace" will start.
- Post-trigger duration
  Post-trigger duration allows you to define, how long after the start trigger was set, the "Bus trace" will stop.

Extended

This tab provides extended settings for "Bus trace".

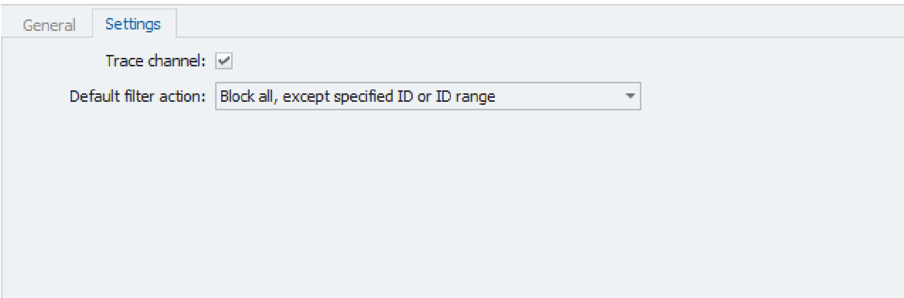


- Include internal events
  Define whether the occurrance of internal events should be stored in the trace data.
- Include all trigger events
  Define whether all trigger events should be stored in the trace data.

- Identifier
  The field Identifier allows you to give a user identifier to the Bus trace. It does not have any effect other than helping the user identifiy a specific Bus trace.

7.7.3.3 Tracable Bus channel

Settings

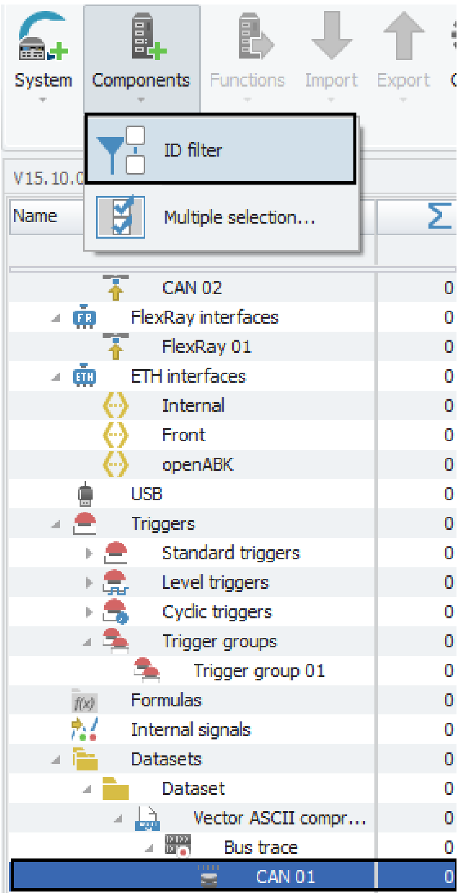


- Trace channel
  Mark this box activ in order to trace this channel.
- Default filter action
  Define the default action for an ID Filter if it has been defined. For information on working with ID Filters for Bus trace, please refer to the chapter **Bus trace ID Filter** (→7.7.4).

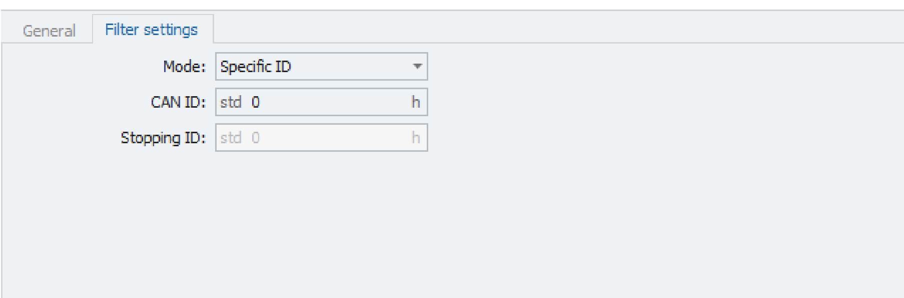
7.7.4 Bus trace ID Filter

For "Bus trace" it is possible to specify one or more "ID Filters". Such it is possible to specify a singe ID or an ID range for a Bus and filter incoming traffic on that bus accrodingly.
It is possible to either block all traffic except the specified ID/ID range or to pass all traffic except the specified ID/ID range.

In order to specify an "ID Filter", you will first need to add the "ID Filter" component to the desired Bus channel. To do so, select the desired Bus trace channel in the tree, click on the "Components" button in the Ribbon and then choose "ID Filter". The new "ID Filter"



will appear in the grid area of the channel it belongs to. Select the filter and navigate to the "Filter settings" tab in the details area. Here you will be able to specify the settings for the filter.

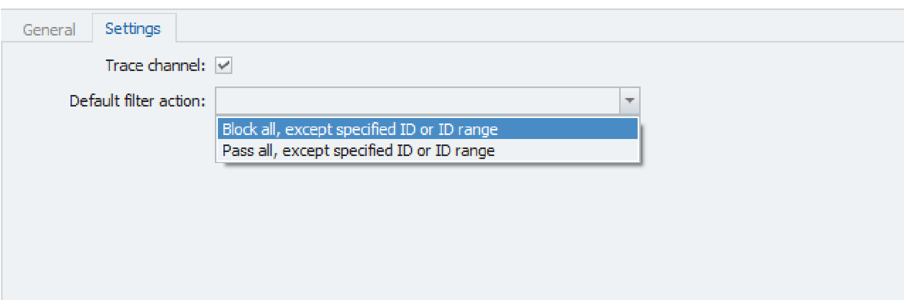


- Mode
  Specify whether a specific ID or a range of IDs should be used for the filter.

- CAN/LIN ID (For CAN and LIN channels only)
  Allows you to define the specific/starting ID of your filter.
- Frame ID (For FlexRay channels only)
  Allows you to define the specific/starting ID of your filter.
- Stopping ID
  If using a range of IDs, this field allows you to define the stopping ID.
- Cycle repetition (For FlexRay channels only)
  Allows you to define cycle repetition of the frame.
- Base cycle (For FlexRay channels only)
  Allows you to define Base cycle of the frame.

All numbers in this tab can be entered in binary, decimal or hexadecimal format. Furthermore can the ID mode for the CAN ID switched from standard (0-7FF) to extended (0-1FFFFFFF).

Now the "ID Filter" has been specified and you will be able to define what the default action for filters on the traced Bus channel should be. To do so, select the respective Bus trace channel in the tree and navigate to the "Settings" tab in the details area. Here you will be able to choose between to default actions for filters on this channel.



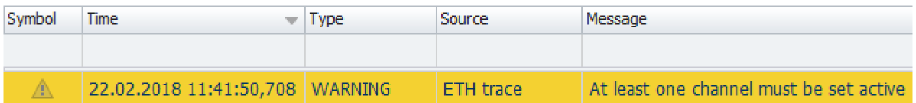
- **Block all, except specified ID or ID range** will block out and ignore all incoming traffic on the channel except the specified ID/ID range.
- **Pass all, except specified ID or ID range** will store all incoming traffic on this channel except the spdecified ID/ID range, which will be blocked and ignored.

It is possible to add and specify multiple ID Filters for one Bus trace channel.

7.8 PCAP

PCAP is a filetype for message orientated rawdate recordings on ethernet channels. The trace method records all the messages that arrive on the ethernet channel. Regardless of the signals defined, all the messages are recorded. Filter rules can be defined to reduce the data volume. A typical trace application is the acquisition of all raw data in order to later evaluate the total traffic on the channel. Unlike most of the other methods, traces are event-oriented. This means the messages are not retrieved from the channels according to a set time pattern, but are recorded as soon as they arrive on the channel. This method accordingly has no parameter for sampling rate.

Including PCAP in your dataset will produce a "Warning" message, saying that "at least one channel must be set active".



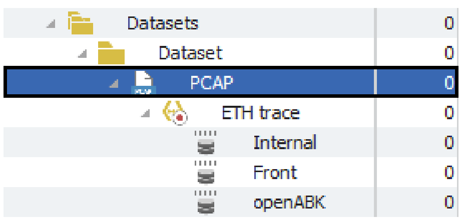
Symbol	Time	Type	Source	Message
⚠	22.02.2018 11:41:50,708	WARNING	ETH trace	At least one channel must be set active

Please refer to the section "ETH trace" (→ 7.8.3.2), in order to activate a channel for tracing.

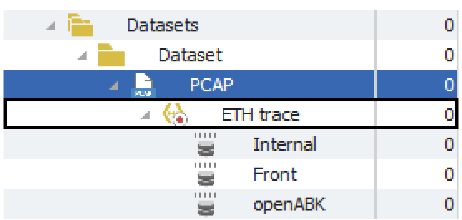
7.8.1 Tree elements for PCAP

Including PCAP in your dataset will add various new child elements to your tree element "Dataset":

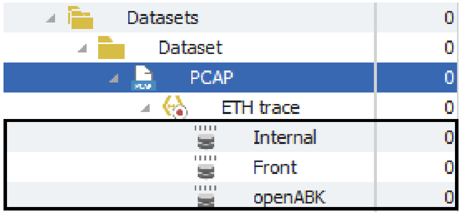
- PCAP
  This element represents the PCAP file, which will later be included in your exported dataset. You may add multiple files of the same filetype.



- ETH trace
  This element represents the "ETH trace" for recording all the traffic on a selected ethernet channel.

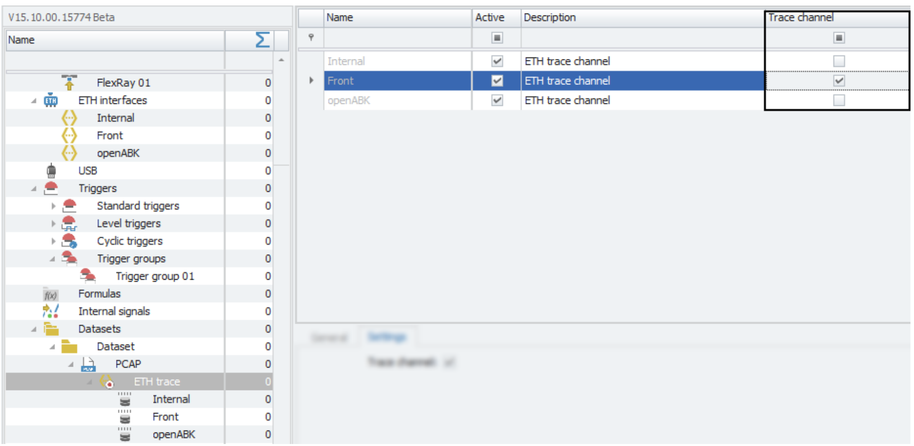


- ETH channels available for tracing
  As child elements to the tree element "ETH trace" will appear all the ETH channels which are currently available for tracing.



7.8.2 Grid area for PCAP

If the "ETH trace" is selected in the Measurement task tree, the grid area will show an overview of the ethernet channels available for tracing.
At least one channel must be marked active for tracing, by ticking the checkbox labeled "Trace channel".



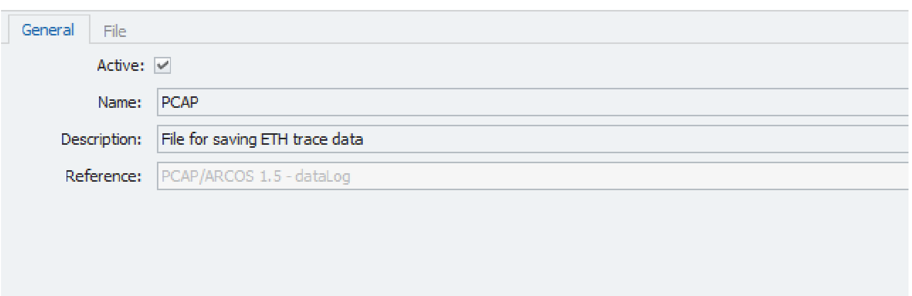
7.8.3 Details area for PCAP

The details area contains settings for the behaviour of your PCAP file or "ETH trace".

7.8.3.1 PCAP file

General

This tab provides general settings for the selected Eth tracing file.



- Active
  Allows you to activate or deactivate the selected file.

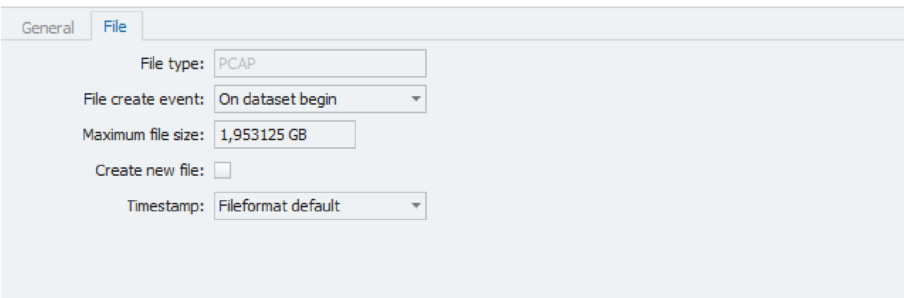
- Name
  Give a user-defined Name to the selected file.

- Description
  Give a user-defined description to the selected file.

- Reference
  This field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed.

File

This tab provides settings regarding the creation of the file.



- File type
  Tells you the type of the created file.
- File create event
  Define, when the PCAP file should be created. There are four possibilities:

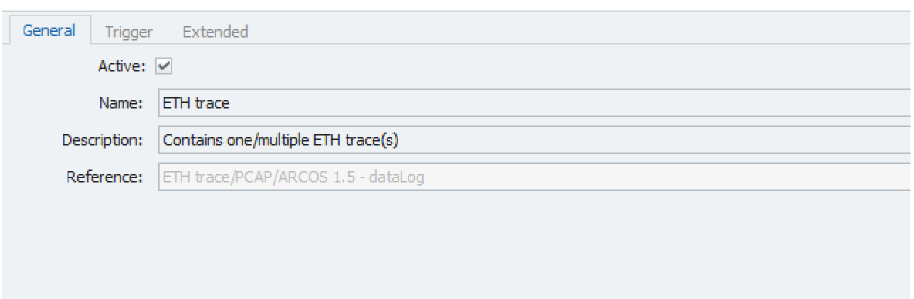
File create event	Characteristics
On dataset begin	The file will be created everytime, the dataset is started or restarted after a pause. This may result in a splitting of the current PCAP file into multiple files, as a new file is created for each time a dataset ist started.
On recording start	The file will be created once at logger start.
On trigger	The file will be created on a trigger and record for a user defined duration. These settings can be defined in the "ETH trace" settings (→ 7.8.3.2). This will result in a splitting of the current Eth tracing file into multiple files, as a new file is created for each timethe trigger is set.
On first trigger	The file will be created once, when the defined trigger is set for the first time and record for a user defined duration. These settings can be defined in the "ETH trace" settings → 7.8.3.2). Each following thime the trigger is set, the data will be written in the same previously created file. Therefore there will only be one file.

- Maximum file size
  Define the maximum file size. It is recommended not to raise the maximum file size above 2GB, as some third party analysis tools cannot handle files, that are larger.
- Create new file
  If this box is marked active, a new file will be created, if the current file exceeds the maximum file size.
- Timestamp
  This dropdown menu allows you to set the format of the timestamp for the file.

7.8.3.2 Eth trace

General

This tab provides general settings for the selected "ETH trace".



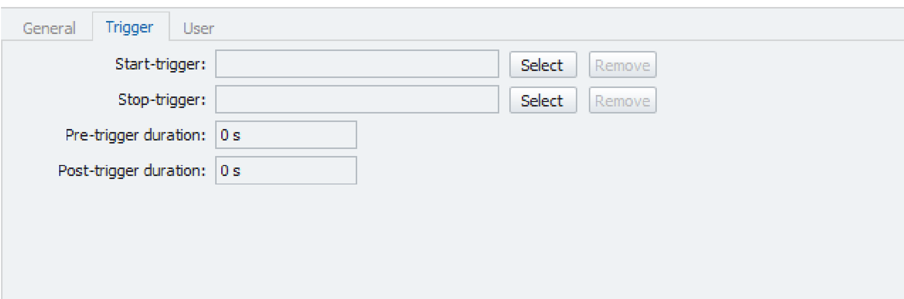
- Active
  Allows you to activate or deactivate the selected file.

- Name
  Give a user-defined Name to the selected file.

- Description
  Give a user-defined description to the selected file.

- Reference
  This field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed.

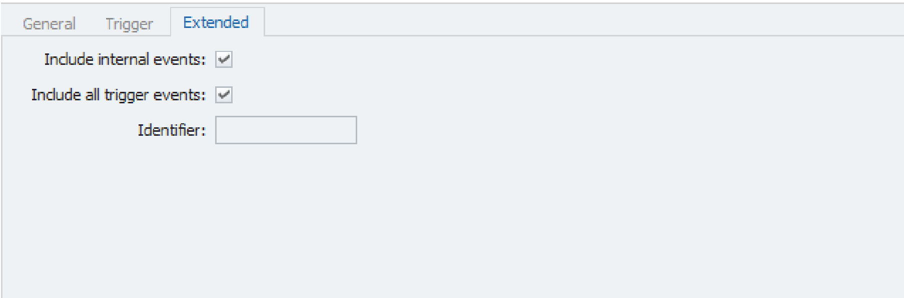
Trigger
This tab provides settings regarding the trigger for the start and stop of the "ETH trace". When the "ETH trace" is started, all traffic on the channel will be stored to the PCAP file. Furthermore will this trigger provoke the creation of the PCAP file, if you have choosen either "On trigger" or "On first trigger" as "File create event".



- Start-trigger
  Define a trigger, that will start the "ETH trace".
- Stop-trigger
  Define a trigger, that will stop the "ETH trace".
- Pre-trigger duration
  Pre-trigger duration allows you to define, how long before the start trigger was set, the "ETH trace" will start.
- Post-trigger duration
  Post-trigger duration allows you to define, how long after the start trigger was set, the "ETH trace" will stop.

Extended

This tab provides extended settings for "ETH trace".



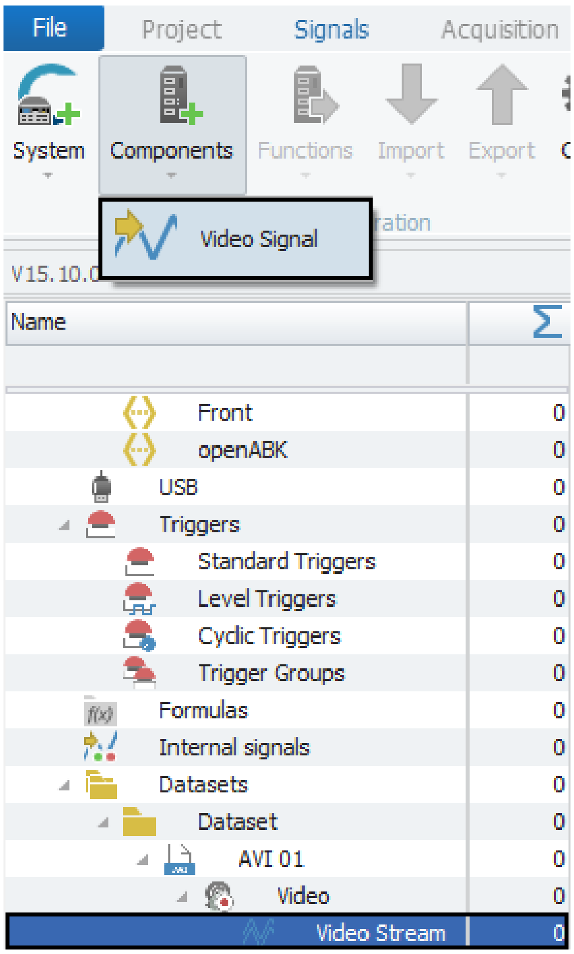
- **Include internal events**
  Define whether the occurrance of internal events should be stored in the trace data.
- **Include all trigger events**
  Define whether all trigger events should be stored in the trace data.
- **Identifier**
  The field Identifier allows you to give a user identifier to the ETH trace. It does not have any effect other than helping the user identifiy a specific ETH trace.

7.9 AVI

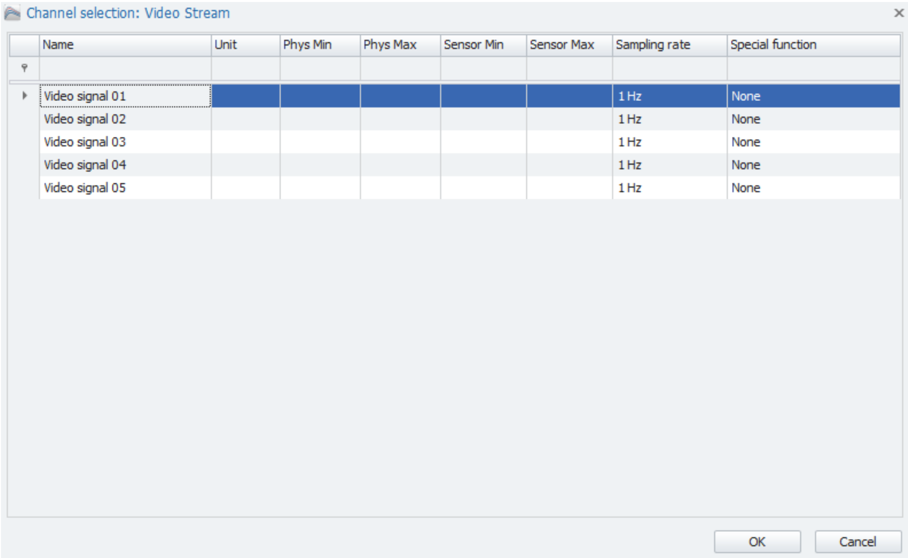
The “AVI” filetype is meant for recording of video streams. For each AVI file, that you add to your dataset you can only record one video stream.

7.9.1 Including a video signal in the Video Stream

In order to store a “Video Stream”, you will need to include a video signal in your “Video Stream”.
To do so, select the tree element “Video Stream”, click the “Compontents” button in the Ribbon and choose “Video Signal”.



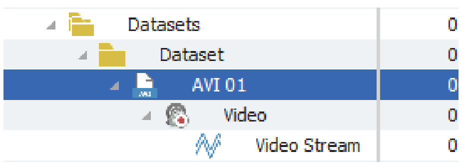
In the resulting window you will be presented with an overview of all the available video signals. You can chooe one signal and confirm by clicking "OK".



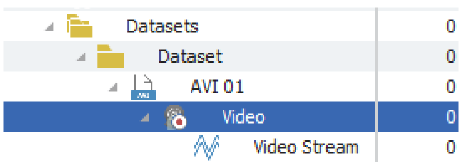
7.9.2 Tree elements for AVI

Including the "AVI" filetype in your dataset will add three new child elements to your tree element "Dataset":

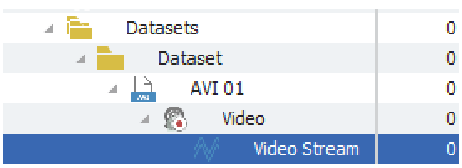
- AVI xx
  This element represents the AVI file, which will later be included in your exported dataset. You may add multiple files of the same filetype.



- Video
  Represents the video element in your AVI file. It tells you, that the type of signal, which can be stored, is a video signal.

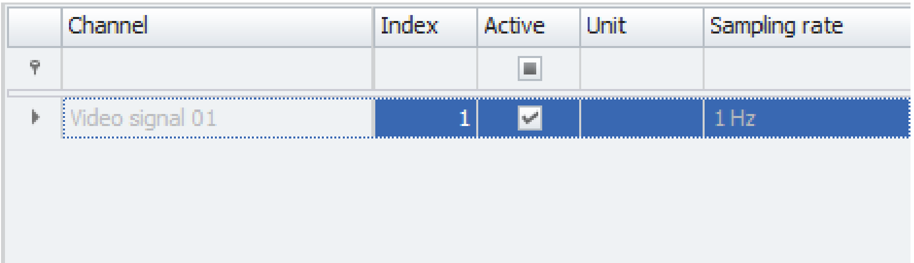


- Video Stream
  This element represents the video signal which will be stored.



7.9.3 Grid area for AVI

If the "Video Stream" is selected in the Measurement task tree, and a video signal has already been included in the "Video Stream", the grid area will show the video signal which has been included in the "Video Stream".



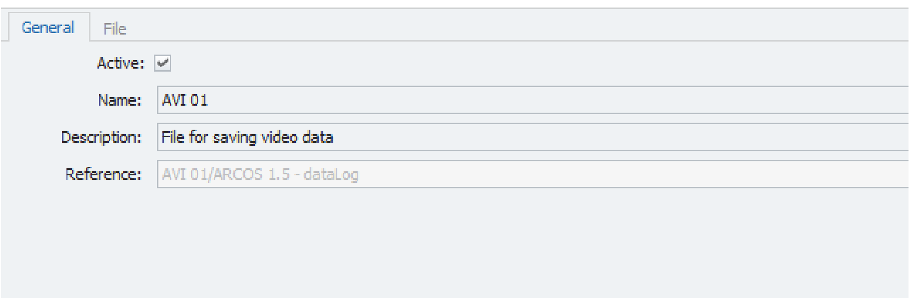
7.9.4 Details area for AVI

The details area contains settings for the behaviour of your AVI file, Video element or Video Stream.

7.9.4.1 AVI File

General

This tab provides general settings for the selected AVI file.



- Active
  Allows you to activate or deactivate the selected file.

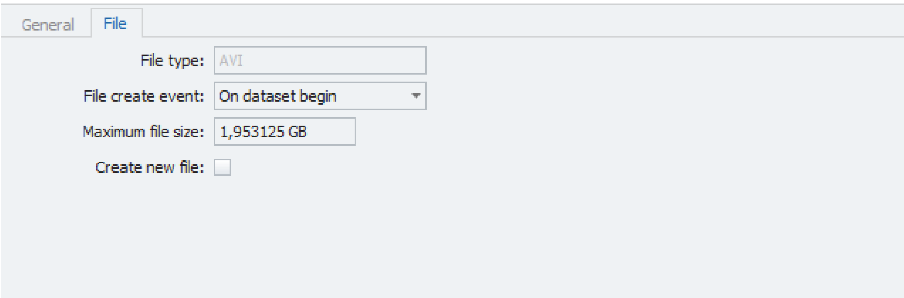
- Name
  Give a user-defined Name to the selected file.

- Description
  Give a user-defined description to the selected file.

- Reference
  This field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed.

File

This tab provides settings regarding the creation of the file.



- File type
  Tells you the type of the created file.
- File create event
  Define, when the AVI file should be created. There are four possibilities:

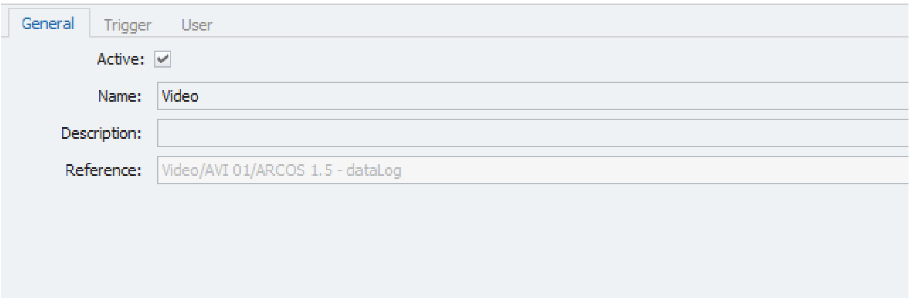
File create event	Characteristics
On dataset begin	The file will be created everytime, the dataset is started or restarted after a pause. This may result in a splitting of the current AVI file into multiple files, as a new file is created for each time a dataset ist started.
On recording start	The file will be created once at logger start.
On trigger	The file will be created on a trigger and record for a user defined duration. These settings can be defined in the timelog settings (→ 7.9.4.2). This will result in a splitting of the current AVI file into multiple files, as a new file is created for each timethe trigger is set.
On first trigger	The file will be created once, when the defined trigger is set for the first time and record for a user defined duration. These settings can be defined in the timelog settings → 7.9.4.2). Each following thime the trigger is set, the data will be written in the same previously created file. Therefore there will only be one file.

- Maximum file size
  Define the maximum file size. It is recommended not to raise the maximum file size above 2GB, as some third party evaluation tools cannot handle files, that are larger.
- Create new file
  If this box is marked active, a new file will be created, if the current file exceeds the maximum file size.

7.9.4.2 Video

General

This tab provides general settings for the selected Video element.



- Active
  Allows you to activate or deactivate the selected file.

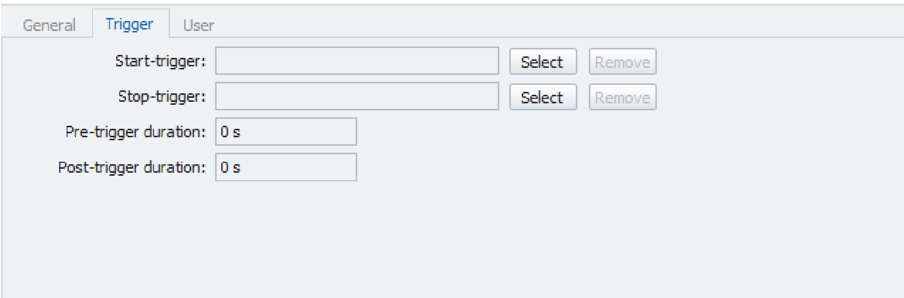
- Name
  Give a user-defined Name to the selected file.

- Description
  Give a user-defined description to the selected file.

- Reference
  This field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed.

Trigger

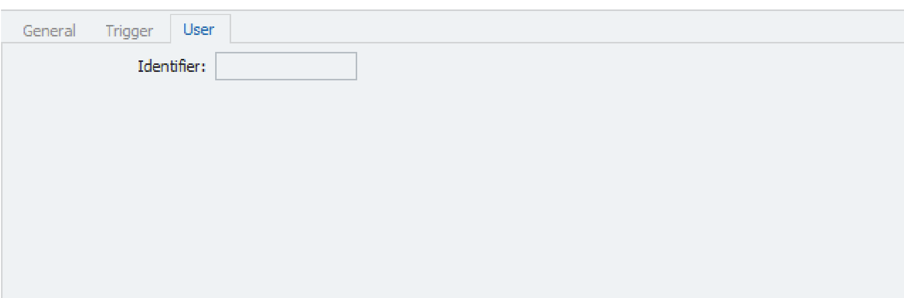
This tab provides settings regarding the trigger for the start and stop of the Video element. When the Video element is started, the "Video Stream" will be stored to the AVI file. Furthermore will this trigger provoke the creation of the AVI file, if you have choosen either "On trigger" or "On first trigger" as "File create event".



- Start-trigger
  Define a trigger, that will start the Video element.
- Stop-trigger
  Define a trigger, that will stop the Video element.
- Pre-trigger duration
  Pre-trigger duration allows you to define, how long before the start trigger was set, the Video element will start.
- Post-trigger duration
  Post-trigger duration allows you to define, how long after the start trigger was set, the Video element will stop.

User

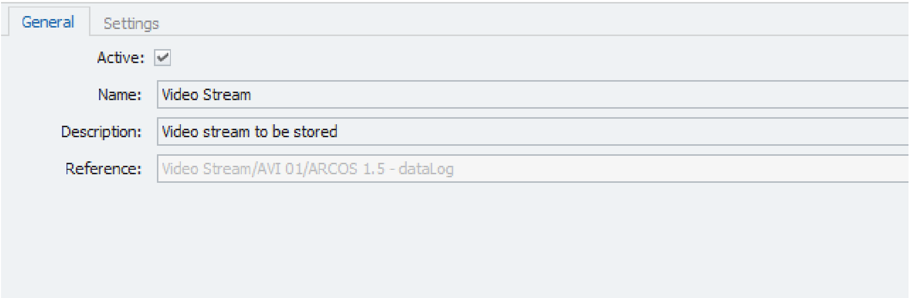
The field "Identifier" allows you to give a user identifier to the timelog. It does not have any effect other than helping the user identifiy a specific timelog.



7.9.4.3 Video Stream

General

This tab provides general settings for the selected "Video Stream".



- Active
  Allows you to activate or deactivate the "Video Stream".

- Name
  Give a user-defined Name to the selected "Video Stream".

- Description
  Give a user-defined description to the selected "Video Stream".

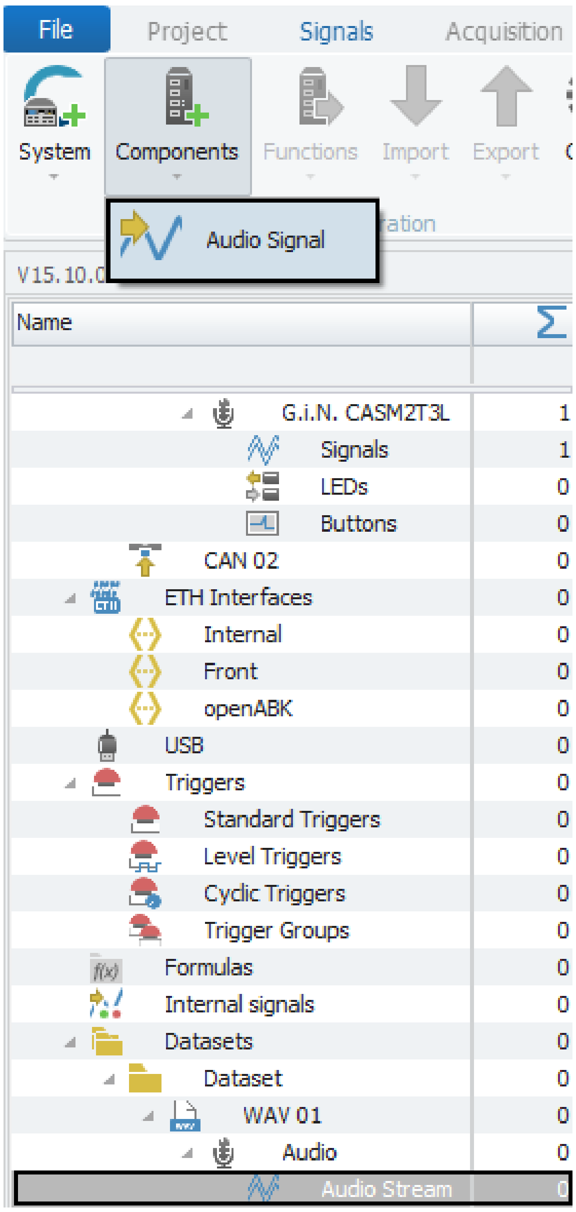
- Reference
  This field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed.

7.10 WAV

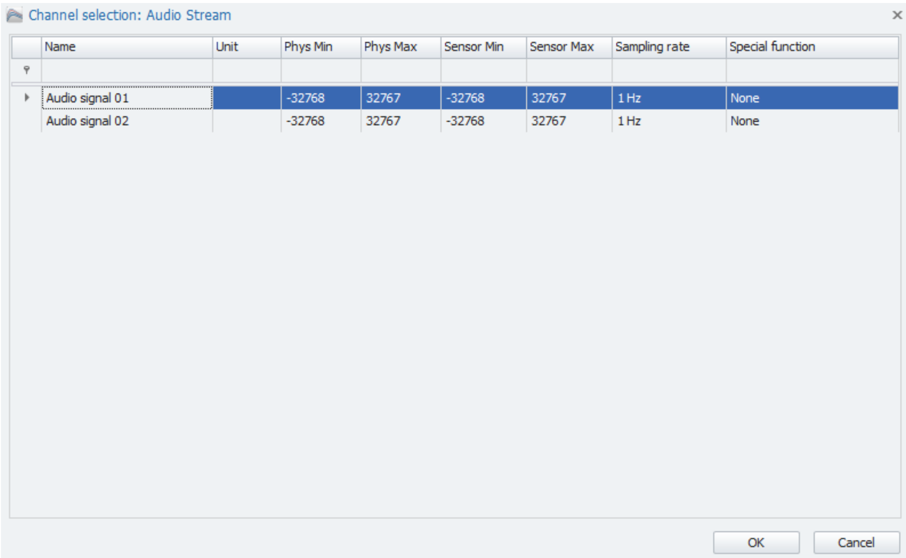
The “WAV” filetype is meant for recording of audio streams. For each WAV file, that you add to your dataset you can only record one audio stream.

7.10.1 Including an audio signal in the audio Stream

In order to store an “Audio Stream”, you will need to include a audio signal in your “Audio Stream”.
To do so, select the tree element “Audio Stream”, click the “Compontents” button in the Ribbon and choose “Audio Signal”.



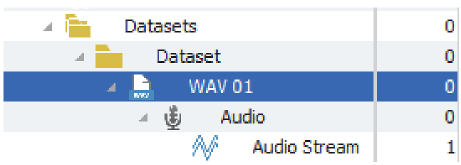
In the resulting window you will be presented with an overview of all the available audio signals. You can chooe one signal and confirm by clicking "OK".



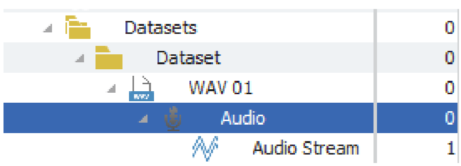
7.10.2 Tree elements for WAV

Including the "WAV" filetype in your dataset will add three new child elements to your tree element "Dataset":

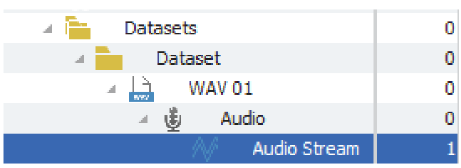
- WAV xx
  This element represents the WAV file, which will later be included in your exported dataset. You may add multiple files of the same filetype.



- Audio
  Represents the audio element in your WAV file. It tells you, that the type of signal, which can be stored, is an audio signal.

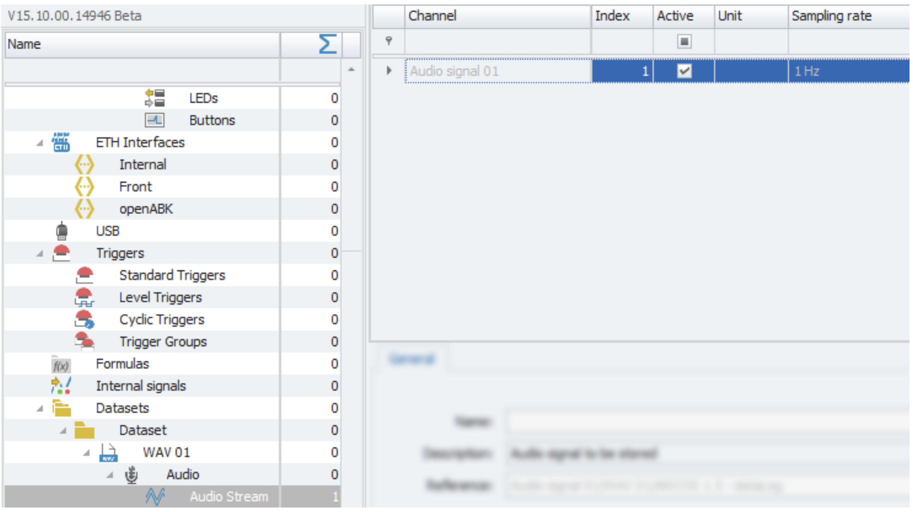


- Audio Stream
  This element represents the audio signal which will be stored.



7.10.3 Grid area for WAV

If the "audio Stream" is selected in the Measurement task tree, and a audio signal has already been included in the "audio Stream", the grid area will show the audio signal which has been included in the "audio Stream".

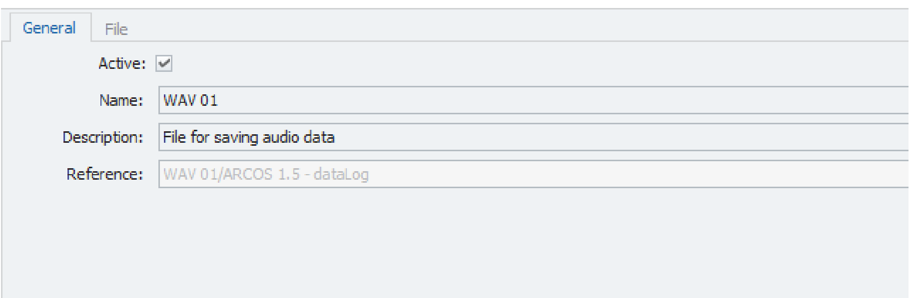


7.10.4 Details area for WAV

The details area contains settings for the behaviour of your WAV file.

General

This tab provides general settings for the selected WAV file.



- **Active**
  Allows you to activate or deactivate the selected file.

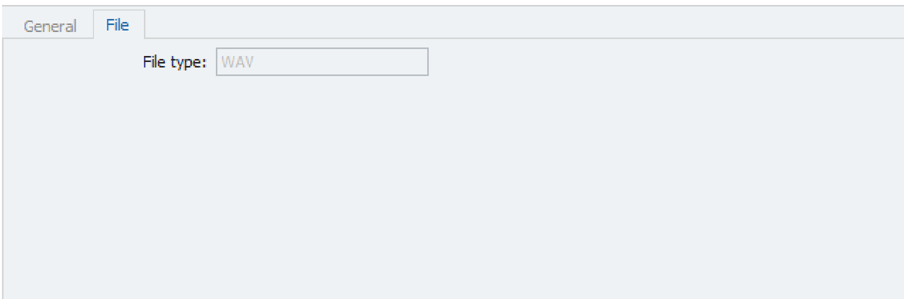
- **Name**
  Give a user-defined Name to the selected file.

- **Description**
  Give a user-defined description to the selected file.

- Reference
  This field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed.

File

This tab tells you the filetype of your file.



7.11 GPX

The “GPX” filetype is meant for GPS Tracking.

7.11.1 Assigning GPS signals

Other than for other filetypes, when working with GPX, you will not be able to choose the signals you wish to store from a list of available signals, but you will need to assign a GPS task (latitude, longitude or altitude) to a signal. That means, that technically any signal can be used as GPS signal.

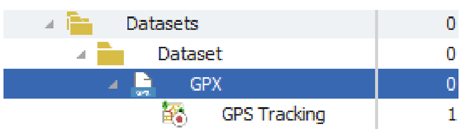
For instructions on assignig GPS tasks please refer to (→ 5.10.2.1).

> One GPS task may only be assigned to one signal. Use the **“Check”** button in the Ribbon to verify, that GPS tasks are uniquely assigned. If a GPS task is multiply assigned, navigate to the “Format” tab of the wrongly assigned signal and deassign the GPS task.
>
> Exporting a configuration will also check your system for validity and inform you, if you have multiply assigned GPS tasks.

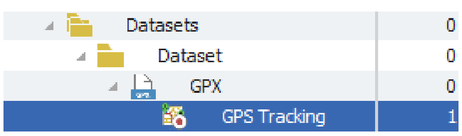
7.11.2 Tree elements for GPX

Including the “GPX” filetype in your dataset will add two new child elements to your tree element “Dataset”:

- GPX
  This element represents the GPX file, which will later be included in your exported dataset.



- GPS Tracking
  Represents the GPS signals you are tracking



7.11.3 Grid area for GPX

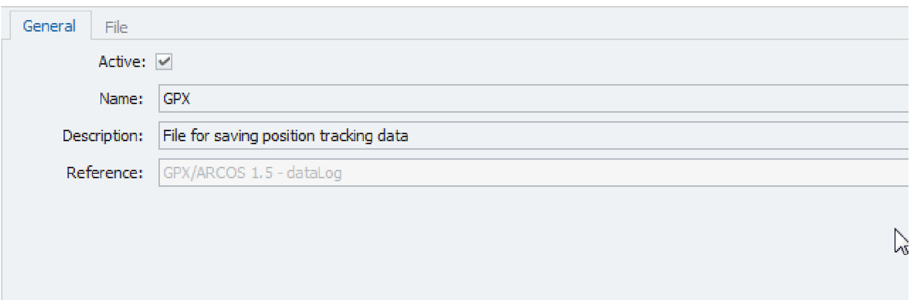
The grid area is not used for configuration of GPX. Instead signal selection in this case works via assignation of GPS tasks. Please refer to (→ 5.10.2.1).

7.11.4 Details area for GPX

The details area contains settings for the behaviour of your GPX file and "GPS Tracking".

7.11.4.1 GPX File

General
This tab provides general settings for the selected GPX file.



- Active
  Allows you to activate or deactivate the selected file.

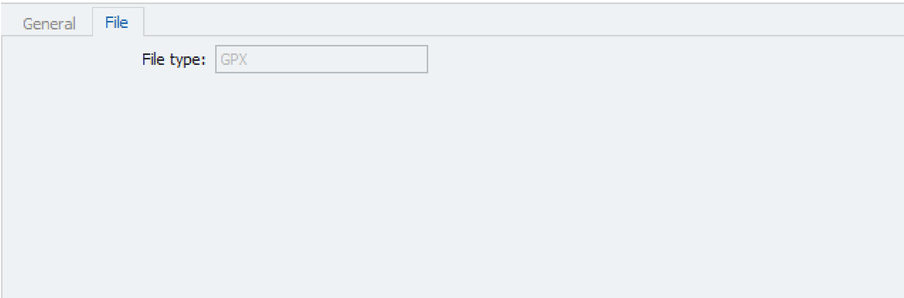
- Name
  Give a user-defined Name to the selected file.

- Description
  Give a user-defined description to the selected file.

- Reference
  This field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed.

File

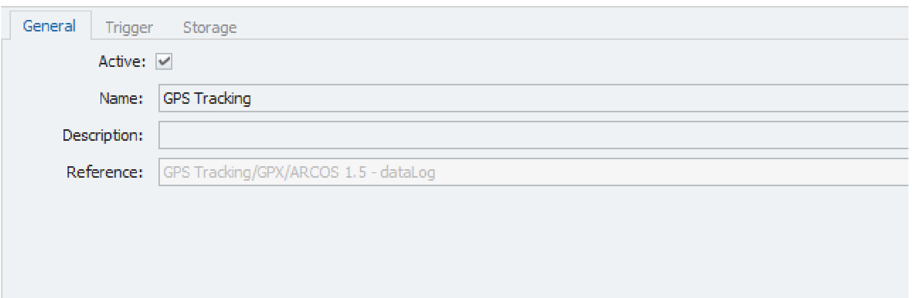
This tab tells you the filetype of your file.



7.11.4.2 GPS Tracking

General

This tab provides general settings for GPS Tracking.



- Active
  Allows you to activate or deactivate "GPS Tracking".

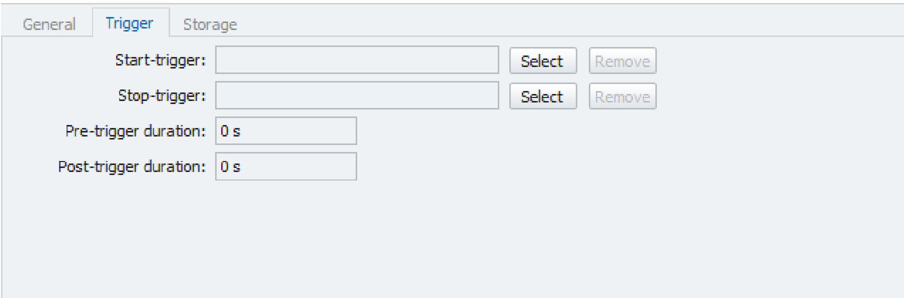
- Name
  Give a user-defined Name to your "GPS Tracking".

- Description
  Give a user-defined description to your "GPS Tracking".

- Reference
  This field serves as the tree element's unique identifier inside the measurement task tree. It cannot be changed.

Trigger

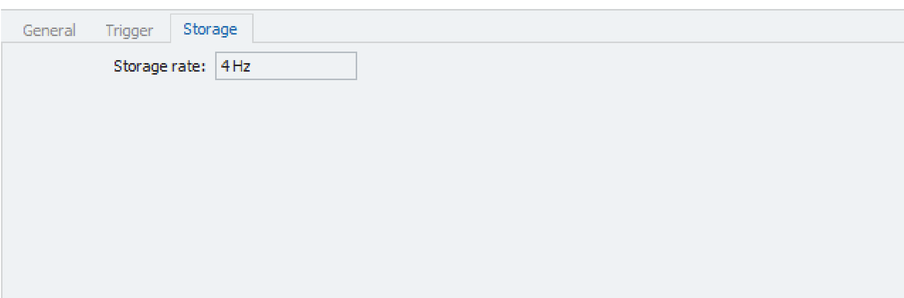
This tab provides settings regarding the trigger for the start and stop of the "GPS Tracking". When the "GPS Tracking" is started, all activated signals with GPS task will be stored to the GPX file.



- Start-trigger
  Define a trigger, that will start "GPS Tracking".
- Stop-trigger
  Define a trigger, that will stop "GPS Tracking".
- Pre-trigger duration
  Pre-trigger duration allows you to define, how long before the start trigger was set, "GPS Tracking" will start.
- Post-trigger duration
  Post-trigger duration allows you to define, how long after the start trigger was set, "GPS Tracking" will stop.

Storage

This tab allows you to set the storage rate for "GPS Tracking"



8 Datatransfer

Stored measurement data can later be transferred from the logger and thus made available for further analysis and processing. To do so, it is necessary, to configure one or more "Transfer events", that wilt trigger the transfer, and the desired connection method used for the transfer. The first chapter of this section will explain "Transfer events" and in the following chapter will be explained configuration of the transfer connection method.

Furthermore the logger will check whenever a data transfer is happening, if a newer version of the current logger-configuration (datalog.ccmc) and firmware are available. Instructions on how to set define the correct path will be explained for each type of transfer connection in the respective chapter.

8.1 Transfer events

In order for data transfer to take place, it must be triggered by an event. These events can be system events, such as starting or shutting down the logger; trigger events, i.e. reactions to defined triggers; or time events, such as the arrival of a particular point in time. Multiple events can also be defined, even of the same type.

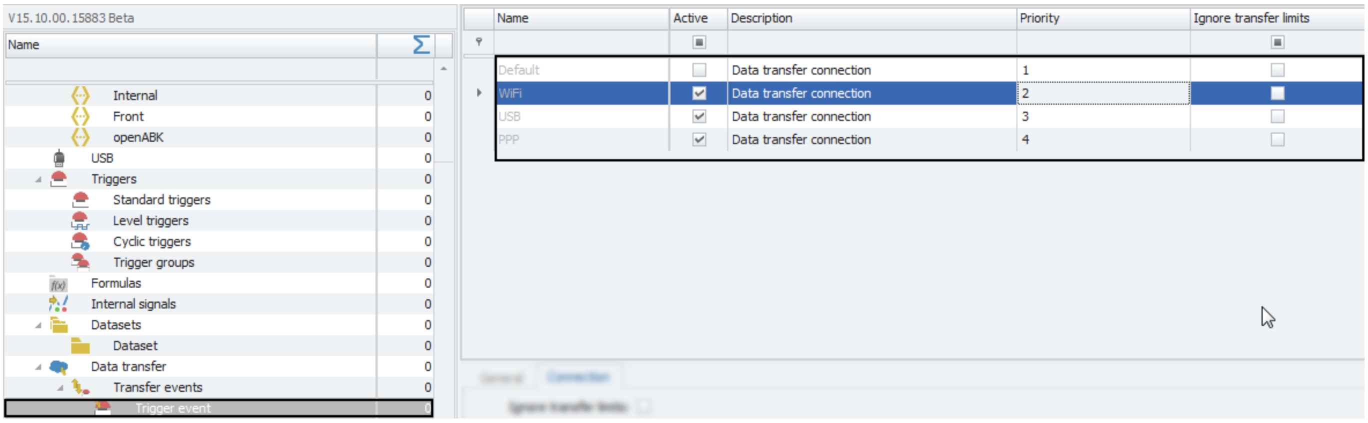
In the event that data transfer limits have been defined, these can, if the need arises, be over-ridden by specific transfer events.

8.1.1 General Information about transfer events

Multiple targets for transfer events

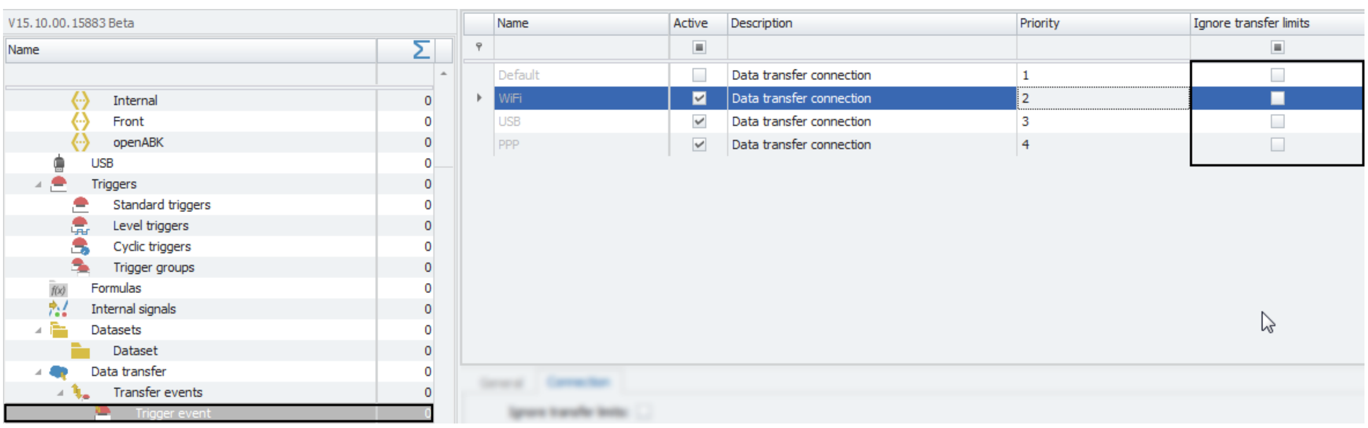
Each transfer event can use multiple connections to different targets, in order to ensure successfull data transfer. In order to do so, activate every connection you want to use for this transfer event in the transfer events grid area. For instructions on transfer connections please refer to (→8.2).

By setting the priority you define, which connection will be used first, starting with priority 1. As soon as data transfer via one connection has been successfully completed, the other connections will be skipped and the dataset will be erased from the logger.



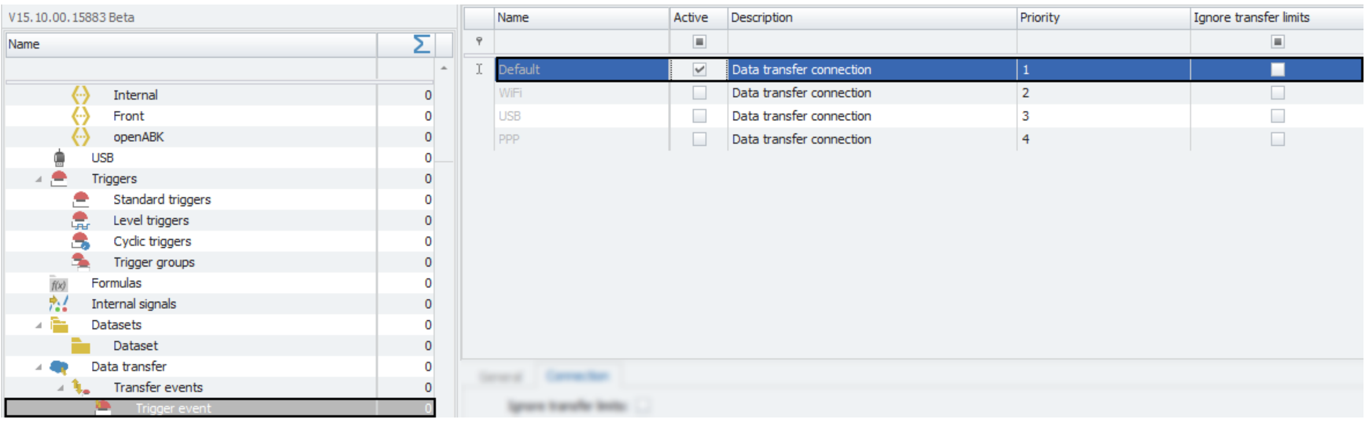
Ignore transfer limits

In the grid area of every transfer event you can choose to "Ignore transfer limits" for every transfer connection. This means, it will ignore any defined restrictions for that transfer connection. These restrictions can be set in the "Basic" settings tab in the details area of each transfer connection. Please refer to (→8.2) and then to the respective connection type's section.



Default data transfer connection

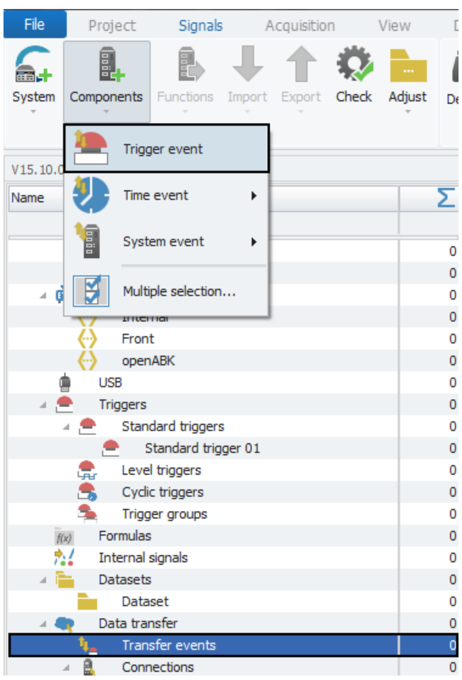
In the grid area of every transfer event you can choose choose the default data transfer connection for data transfer. This means, the connection, that has been defined as default data transfer connection will be used. This setting can be set in the "Basic" settings tab in the details area of each transfer connection. Please refer to (→8.2) and then to the respective connection type's section.



8.1.2 Trigger events

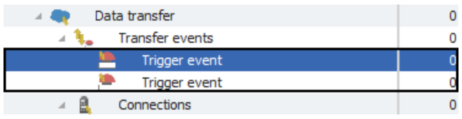
A "Trigger event" will trigger data transfer whenever the selected trigger is set. Any previously configured trigger of your system can be used to trigger data transfer.

In order to configure a "Trigger event", you will first need to add the "Trigger event" as a "Transfer event". To do so, select the tree element "Transfer events", click on the "Components" button in the Ribbon and choose "Trigger event".



8.1.2.1 Tree elements for Trigger events

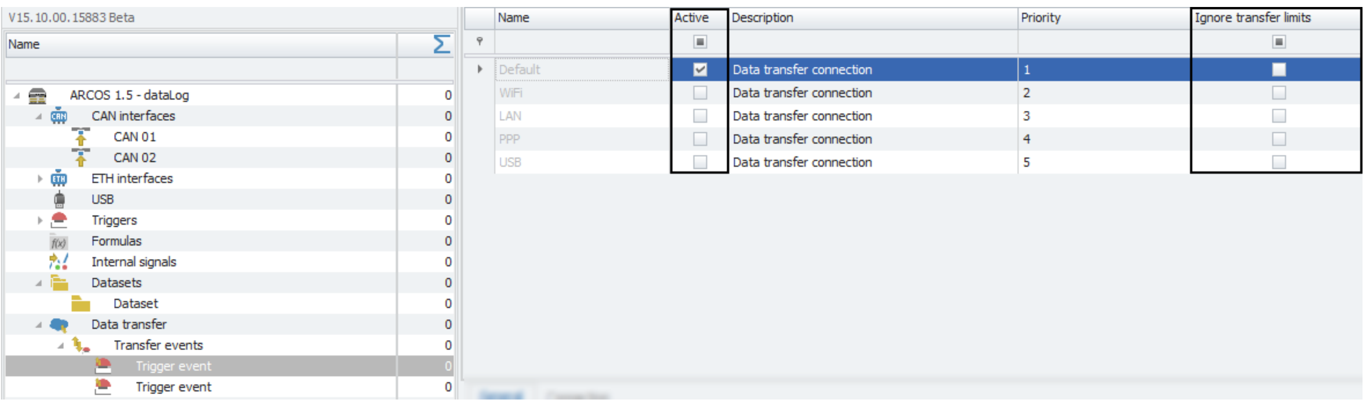
Adding a “Trigger event” for data transfer will add one new child element to the tree element “Transfer events”, called “Trigger event”. Multiple “Trigger events” may be added, which will each result in an extra tree element. The name of these elements can be changed in the details area of each tree element.



8.1.2.2 Grid area for Trigger events

The grid area for a “Trigger event” for data transfer will present you with an overview of the available connection for data transfer.
You can choose the desired connection for data transfer by ticking the “Active” tickbox, and you can override transfer limits by ticking the tickbox labeled “Ignore transfer limits”.

Also you can find here two important functions, which are the **“Column chooser”** (→4.3.1) and the **“Filter editor”** (→4.3.2).



8.1.2.3 Details area for Trigger events

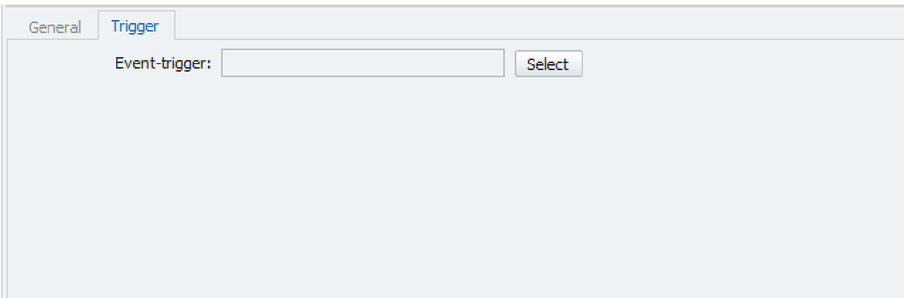
The Details area shows settings for the "Triggger event", that has been selected in the tree. The different tabs of the details area will be explained in the following.

General

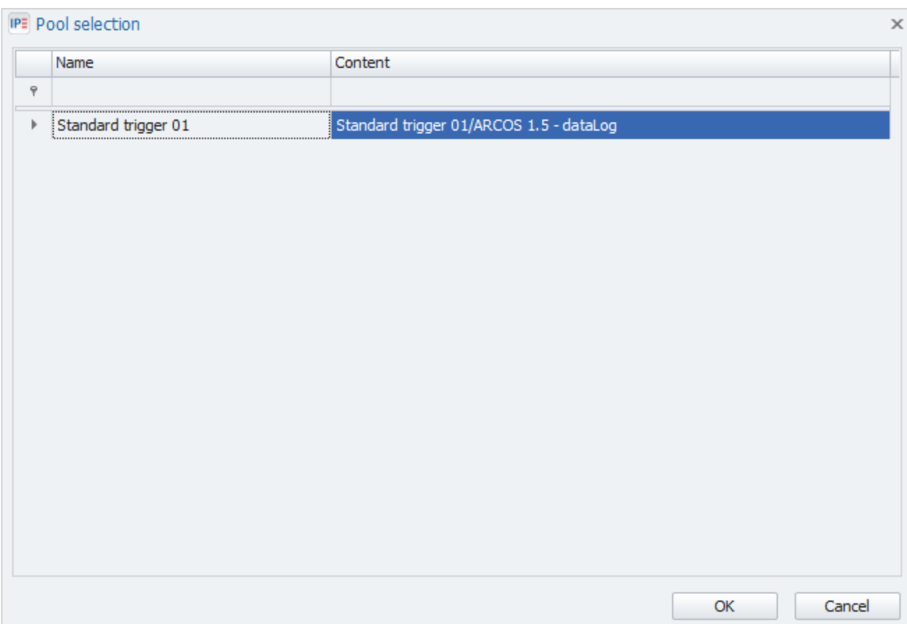
Please refer to (→4.2.2).

Trigger

This tab contains only the setting **"Event trigger"**, which allows you to select which trigger should be used to start data transfer.



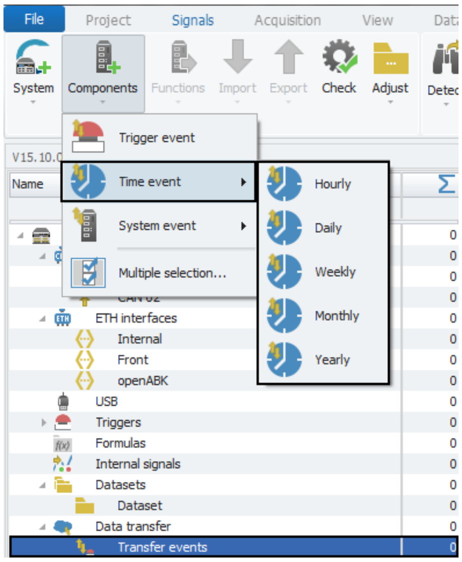
Clicking the "Select" button will open a window, which allows you, to choose the desired trigger. Confirm with "OK".
Once a trigger has been chosen, the "Trigger event" will automatically named after this event.



8.1.3 Time events

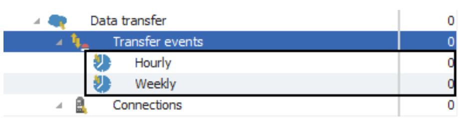
A "Time event" will trigger data transfer according to a set time intervall.

In order to configure a "Time event", you will first need to add the "Time event" as a "Transfer event". To do so, select the tree element "Transfer events", click on the "Components" button in the Ribbon and choose the desired time intervall from the menu "Time event".



8.1.3.1 Tree elements for Time events

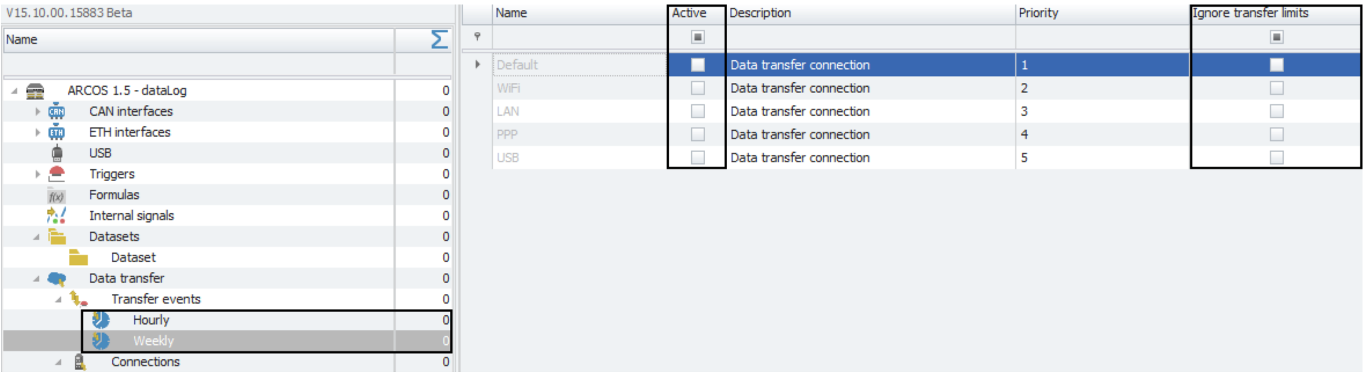
Adding a "Time event" for data transfer will add one new child element to the tree element "Transfer events", called "Time event". Multiple "Time events" may be added, which will each result in an extra tree element. The name of these elements can be changed in the details area of each tree element.



8.1.3.2 Grid area for Time events

The grid area for a "Time event" for data transfer will present you with an overview of the available connections for data transfer.
You can choose the desired connection for data transfer by ticking the "Active" tickbox, and you can override transfer limits by ticking the tickbox labeled "Ignore transfer limits".

Also you can find here two important functions, which are the "Column chooser" (→4.3.1) and the "Filter editor" (→4.3.2).

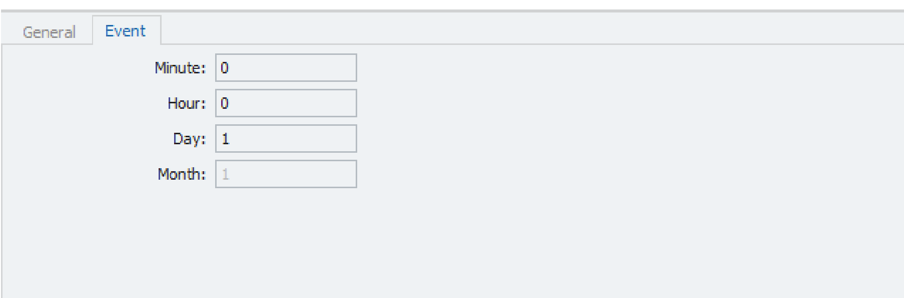


8.1.3.3 Details area for Time events

The Details area shows settings for the "Time event", that has been selected in the tree. The different tabs of the details area will be explained in the following.

General
Please refer to (→4.2.2).

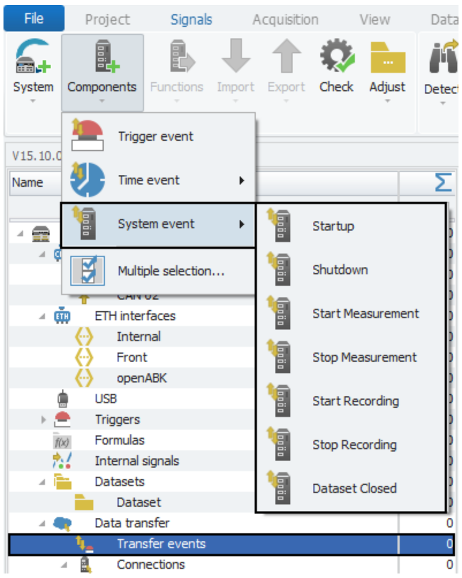
Time
This tab allows you to specify the time intervall for the transfer event.



8.1.4 System events

A “System event” will trigger according to a set “System event” such as “Startup”, “Shutdown”, “Dataset closed” and others.

In order to configure a “System event”, you will first need to add the “System event” as a “Transfer event”. To do so, select the tree element “Transfer events”, click on the “Components” button in the Ribbon and choose the desired event from the menu “System event”.

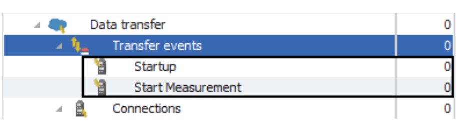


Available System events

System event	Characteristics
Startup	Data transfer will start as soon as startup is finished.
Shutdown	Data transfer will start when shutdown is prompted. Shutdown will not occur until data transfer has finished.
Start Measurement	Data transfer will start as soon as measurement has started.
Stop Measurement	Data transfer will start as soon as measurement has stoped.
Start Recording	Data transfer will start as soon as recording of measurement data in a dataset has started.
Stop Recording	Data transfer will start as soon as recording of measurement data in a dataset has stoped.
Dataset closed	Data transfer will start as soon as the current dataset is closed.

8.1.4.1 Tree elements for System events

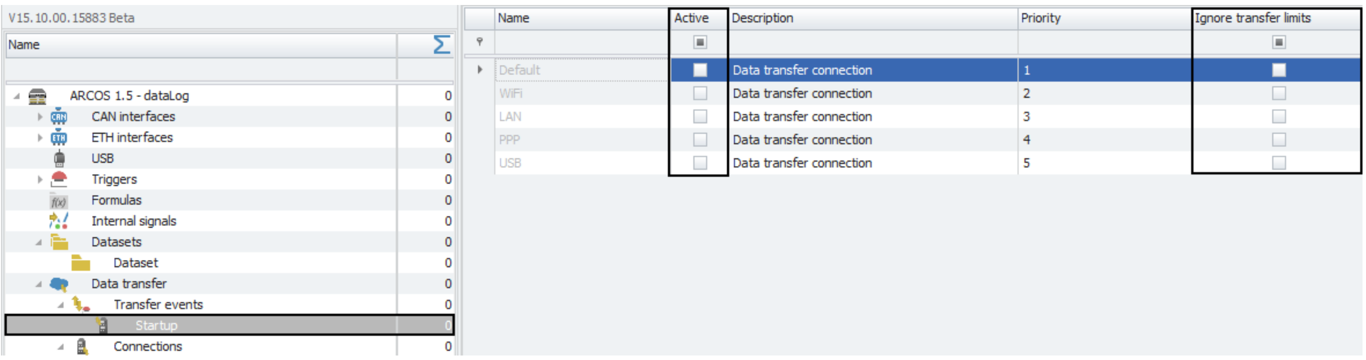
Adding a “System event” for data transfer will add one new child element to the tree element “Transfer events”, called according to the type of “System event” you have chosen. Multiple “System events” may be added, which will each result in an extra tree element.



8.1.4.2 Grid area for System events

The grid area for a "System event" for data transfer will present you with an overview of the available connections for data transfer.
You can choose the desired connection for data transfer by ticking the "Active" tickbox, and you can override transfer limits by ticking the tickbox labeled "Ignore transfer limits".

Also you can find here two important functions, which are the **"Column chooser"** (→4.3.1) and the **"Filter editor"** (→4.3.2).



8.1.4.3 Details area for System events

As "System events" are definite predefined events, there are no further settings for these events available in the details area. It contains only the "General" tab, which will allow you to give a user specific description to the event. Please refer to (→4.2.2).

8.2 Transfer connections

In order for data to be transferred, a transfer connection has to be set up. There are four different types of connections, that can be set up:

- USB (→8.2.1)
- WIFI (→8.2.2)
- LAN (→8.2.3)
- PPP/UMTS (→8.2.4)

8.2.1 Data transfer via USB

USB is the predefined default connection for data transfer and therefore it is included in the measurement task tree by default. Via USB it is possible to transfer data to and from an external storage device.
Configuration of a USB connection happens exclusively inside the details area of the tree element "USB".

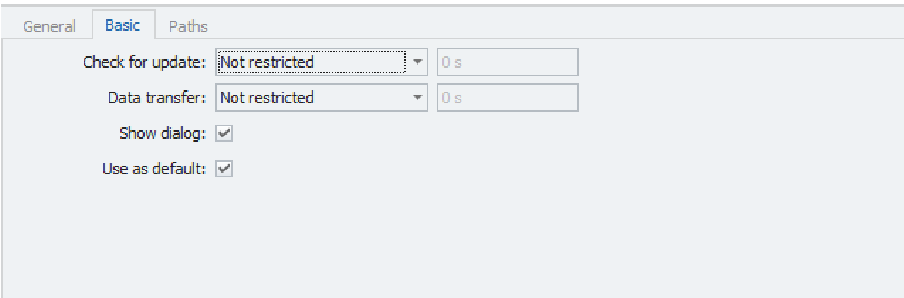
8.2.1.1 Details area for USB

General

Please refer to (→4.2.2).

Basic

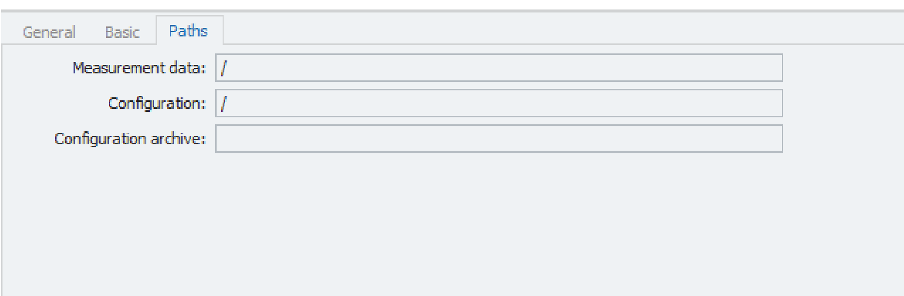
This tab contains basic settings for the connection.



- Check for update
  Allows you to define, how often and when the logger should check for updated configuration or firmware.
  If the option "Interval" is selected, the time interval may be set in the field on the right of the dropdown menu.
- Data transfer
  Allows you to restrict, how often data transfer from the logger may occur.
  If the option "Interval" is selected, the time interval may be set in the field on the right of the dropdown menu.
- Show dialog
  Tick or untick this box in order to show or hide the transfer dialog during measurement.
- Use as default
  Tick or untick this box in order to make this your default connection for data transfer.

Paths

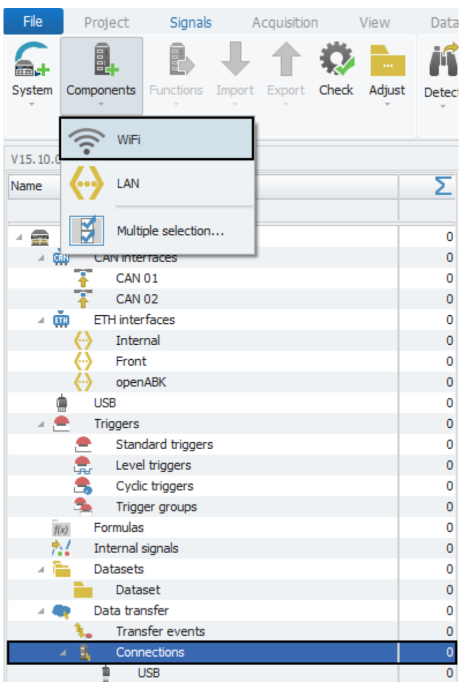
This tab allows you to set the paths for data transfer.



- Measurement data
  Define the path, where measurement data should be stored.
- Configuration
  Define the path, where the logger will check for a newer version of the current configuration (datalog.ccmc) and firmware.
  If there is a newer version, the logger will download it, append the current timestamp in the filename, and apply/install it at the next possible moment.
- Configuration archive (optional)
  Define the subpath for previous logger configurations and firmwares. If this subpath has been defined, the logger will copy the previously used configuration/firmware here, when he receives a newer version. If this subpath is not defined, the logger will leave the file in the "Configuration" path, that has been defined before.

8.2.2 Data transfer via WIFI

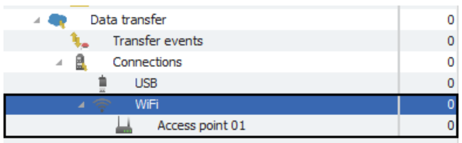
In order to transfer data via WIFI, you will first need to add a WIFI connection to your system. To do so, select the tree element "Connections", click on the "Components" button in the Ribbon and then choose "WIFI".
After having set up the WIFI connection, you will need to set up a fileserver to which to transfer data. For instructions on how to set up a fileserver please refer to the section "Fileserver" (→8.3).



8.2.2.1 Tree elements for WIFI connections

Adding a WIFI connection for data transfer will add two new child elements to the tree element "Connections":

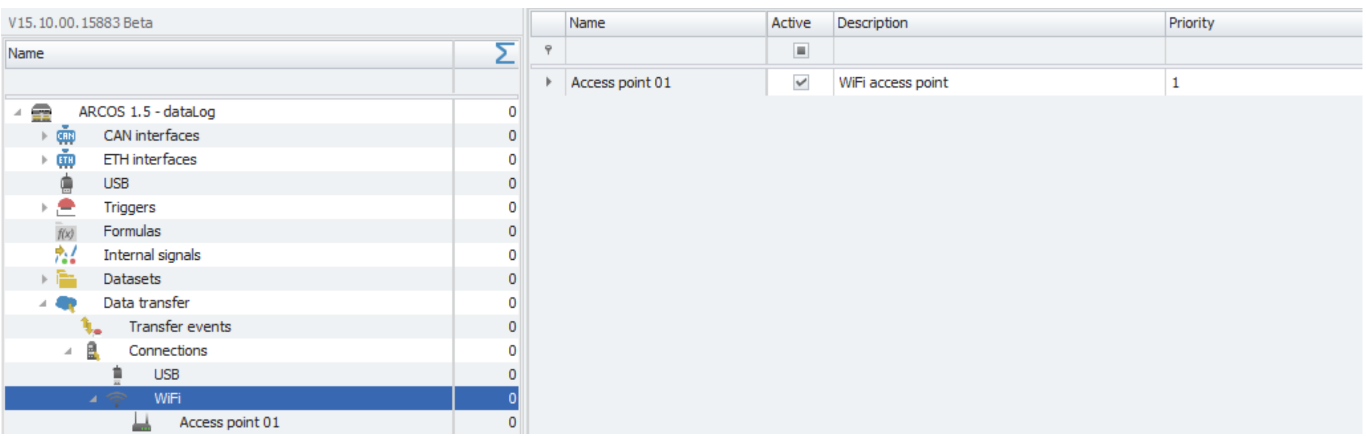
- WIFI
  Represents the WIFI connection itself.
- Access point xx
  Represents the WIFI access point. A WIFI connection can have various access points, in order to connect to different networks. To add a new access point, select the tree element WIFI, then click the "Components" button in the Ribbon and select "Access point".



8.2.2.2 Grid area WIFI connections

The grid area for a "WIFI connection" for data transfer will present you with an overview of all the currently defined access points for WIFI connections.

Also you can find here two important functions, which are the "**Column chooser**" (→4.3.1) and the "**Filter editor**" (→4.3.2).



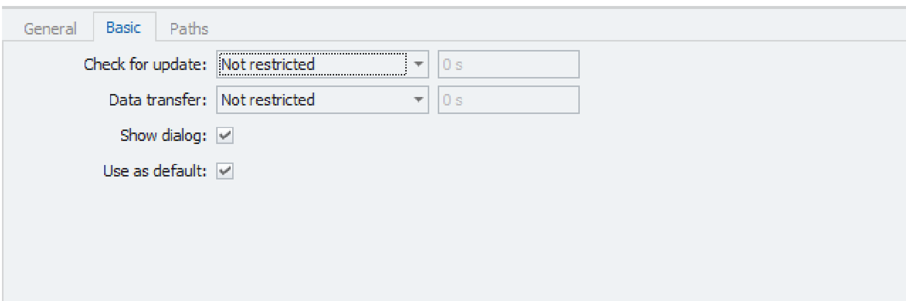
8.2.2.3 Details area for WIFI

The details area provides settings either for the WIFI connection in general or for a specific access point, depending on which element has been selected in the tree.

WIFI Settings

General
Please refer to (→4.2.2).

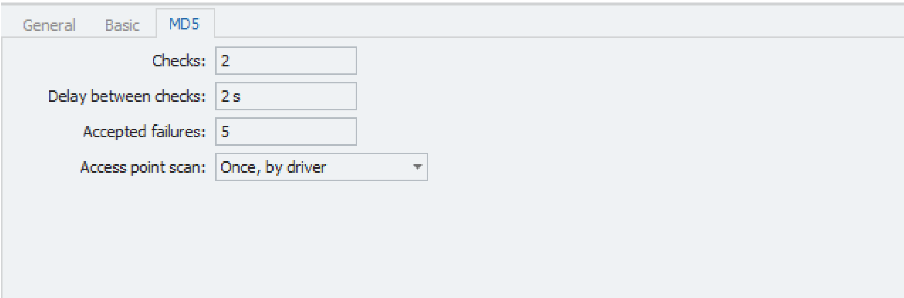
Basic
This tab contains basic settings for the connection.



- Check for update
  Allows you to define, how often and when the logger should check for updated configuration or firmware.
  If the option "Interval" is selected, the time interval may be set in the field on the right of the dropdown menu.
- Data transfer
  Allows you to restrict, how often data transfer from the logger may occur.
  If the option "Interval" is selected, the time interval may be set in the field on the right of the dropdown menu.
- Show dialog
  Tick or untick this box in order to show or hide the transfer dialog during measurement.
- Use as default
  Tick or untick this box in order to make this your default connection for data transfer.

MD5

This tab provides settings regarding the MD5 check of transferred files.



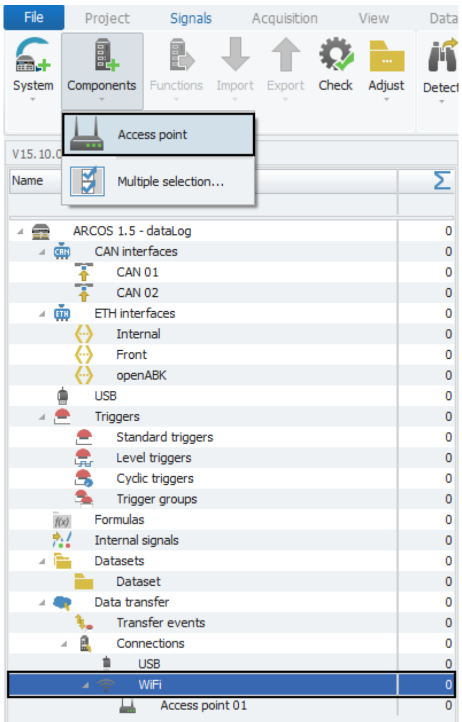
- Checks
  Define how often the check should be executed.
- Delay between checks
  Define the delay between two checks.
- Accepted failures
  Define the number of accepted failures before aborting the data transfer.
- Access point scan
  Specify the method how to search for WIFI access points.

Access point Settings

Each WIFI connection can have multiple access points, in order to be able to connect to different wireless networks. By setting the priority of the access points in the grid area, you define, which access point will be used first, starting with priority 1. As soon as data transfer via one access point has been successfully completed, the other access points will be skipped and the dataset will be erased from the logger.

In order to add an extra access point, select the tree element "WIFI", click on the "Components" button in the Ribbon and the choose "Access point".

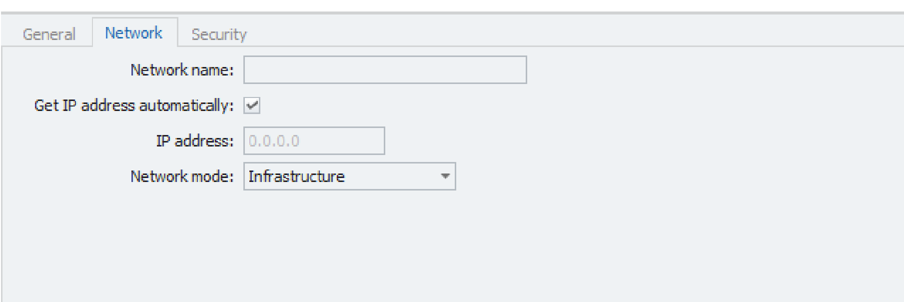
General



Please refer to (→4.2.2).

Network

This tab contains settings regarding the network to which the logger will connect.

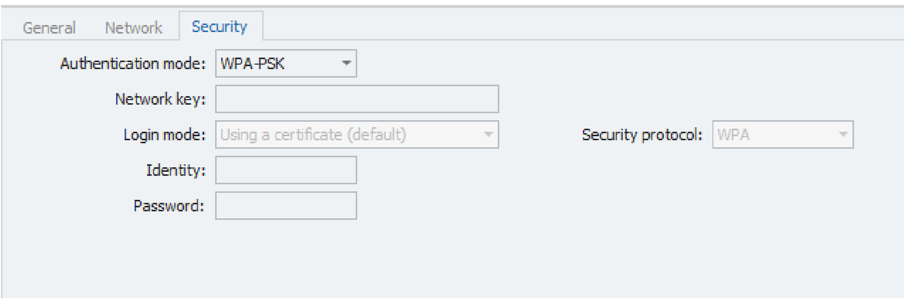


- **Network name**
  Define the name of the WIFI network (SSID).

- Get IP address automatically
  Define whether the logger will expect the allocation of a valid IP address automatically by a DHCP server, or whether you want to manually set an IP address.
- IP address
  IF DHCP is disabled, this field allows you to manually enter an IP address.
- Network mode
  Allows you to choose the mode of the network you wish to connect to.

Security

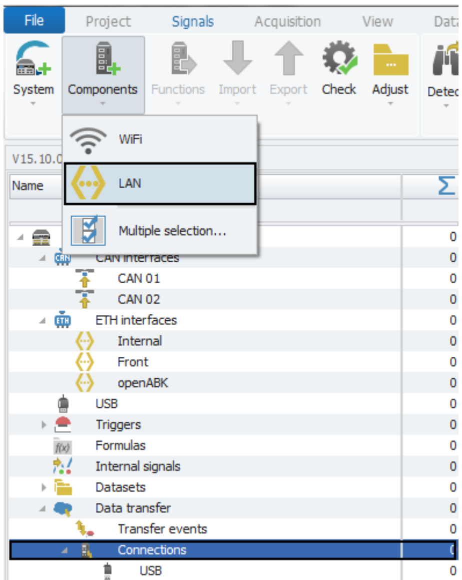
This tab provides security settings regarding the access point and the network you wish to connect to.



- Authentication mode
  Select the authentication mode of the network.
- Network key
  Type in the authentication key of the network.
- Login mode
  If the authentication mode of the network is WPA-EAP, then you may here set whether to login with a certificate or not, and if so, set the security protocol.
- Identity
  If the authentication mode of the network is WPA-EAP, then you may here enter your identity/username.
- Password
  If the authentication mode of the network is WPA-EAP, then you may here enter your password.

8.2.3 Data transfer via LAN

In order to transfer data via LAN, you will first need to add a LAN connection to your system. To do so, select the tree element "Connections", click on the "Components" button in the Ribbon and then choose "LAN".
After you have set up the LAN connection, you will need to additionally set up a fileserver, where the transferred data will be stored. Please refer to the section "Fileserver" (→8.3).



8.2.3.1 Tree elements for LAN connections

Adding a LAN connection for data transfer will add one new child element called "LAN" to the tree element "Connections"

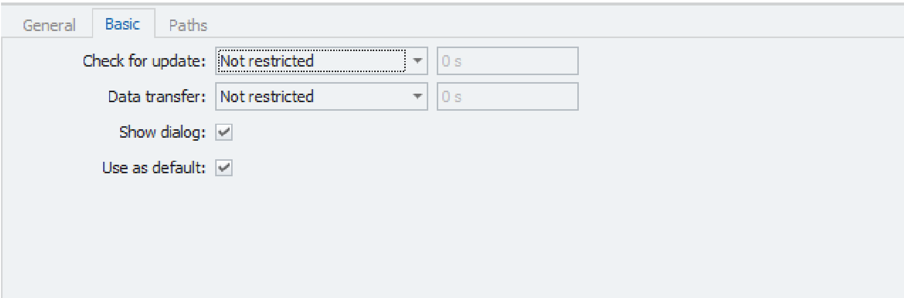
8.2.3.2 Details area for LAN

The details area provides settings for the LAN connection.

General
Please refer to (→4.2.2).

Basic

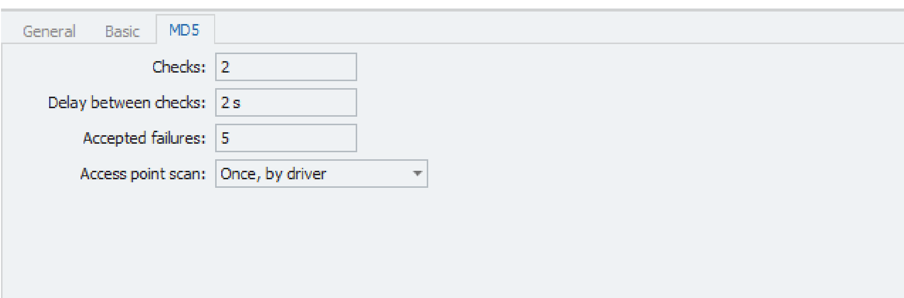
This tab contains basic settings for the connection.



- Check for update
  Allows you to define, how often and when the logger should check for updated configuration or firmware.
  If the option "Interval" is selected, the time interval may be set in the field on the right of the dropdown menu.
- Data transfer
  Allows you to restrict, how often data transfer from the logger may occur.
  If the option "Interval" is selected, the time interval may be set in the field on the right of the dropdown menu.
- Show dialog
  Tick or untick this box in order to show or hide the transfer dialog during measurement.
- Use as default
  Tick or untick this box in order to make this your default connection for data transfer.

MD5

This tab provides settings regarding the MD5 check of transferred files.



- Checks
  Define how often the check should be executed.
- Delay between checks
  Define the delay between two checks.

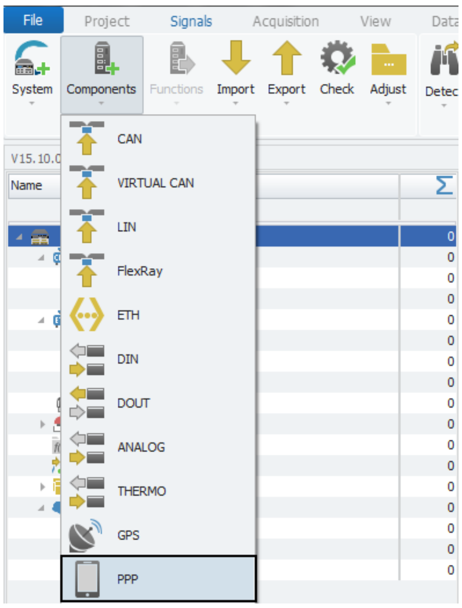
- Accepted failures
  Define the number of accepted failures before aborting the data transfer.

8.2.4 Data transfer via PPP/UMTS

In order to transfer data via PPP/UMTS, you will first need to configure a PPP/UMTS connection. To do so, please refer to (→8.2.4.1). Once a PPP/UMTS connection has been set up, you will need so set the basic settings for PPP data transfer and you will also need to set up a fileserver. This chapter treats the basic settings for data transfer via PPP.
For instructions on how to set up a fileserver please refer to the section "Fileserver" (→8.3).

8.2.4.1 Setting up a PPP/UMTS connection

To set up a PPP/UMTS connection, select your system (Arcos, μCros, μCros XL) in the tree, click the "Components" button in the Ribbon and choose "PPP".

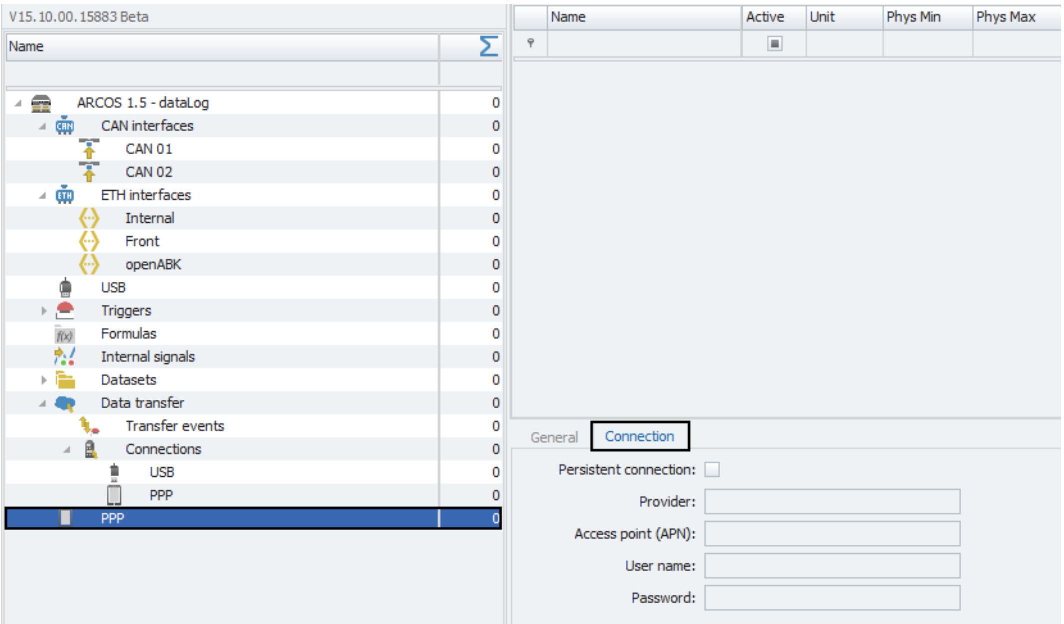


This will add two elements called "PPP" to the measurement task tree. One as a childelement to the main system (Arcos, μCros, μCros XL) and one as a childelement to the tree element "Connections". Select the first of the two, navigate to the tab "Connection" in the details area and fill in the access data. This data can be obtained from your simcard provider.
The option "Persistent connection" allows you to maintain a connection not only during data transfer but also during measurement.

8.2.4.2 Details area for PPP/UMTS

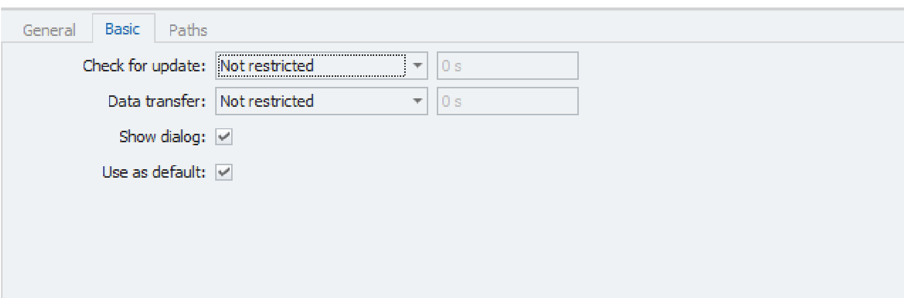
Configuration of a USB connection happens exclusively inside the details area of the tree element "USB".

General



Please refer to (→4.2.2).

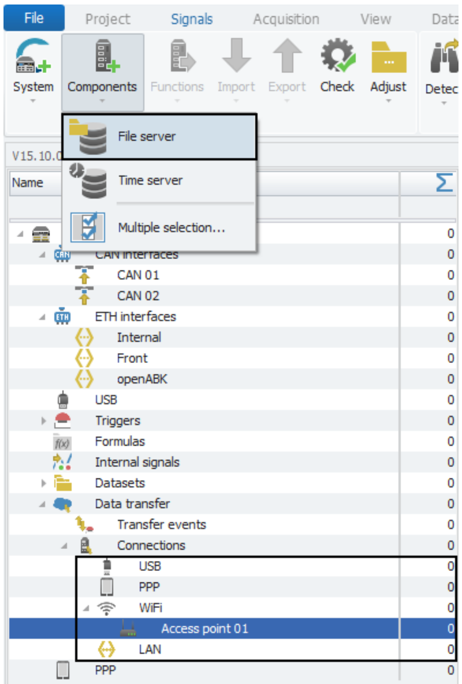
Basic
This tab contains basic settings for the connection.



- Check for update
  Allows you to define, how often and when the logger should check for updated configuration or firmware.
  If the option "Interval" is selected, the time interval may be set in the field on the right of the dropdown menu.
- Data transfer
  Allows you to restrict, how often data transfer from the logger may occur.
  If the option "Interval" is selected, the time interval may be set in the field on the right of the dropdown menu.
- Show dialog
  Tick or untick this box in order to show or hide the transfer dialog during measurement.
- Use as default
  Tick or untick this box in order to make this your default connection for data transfer.

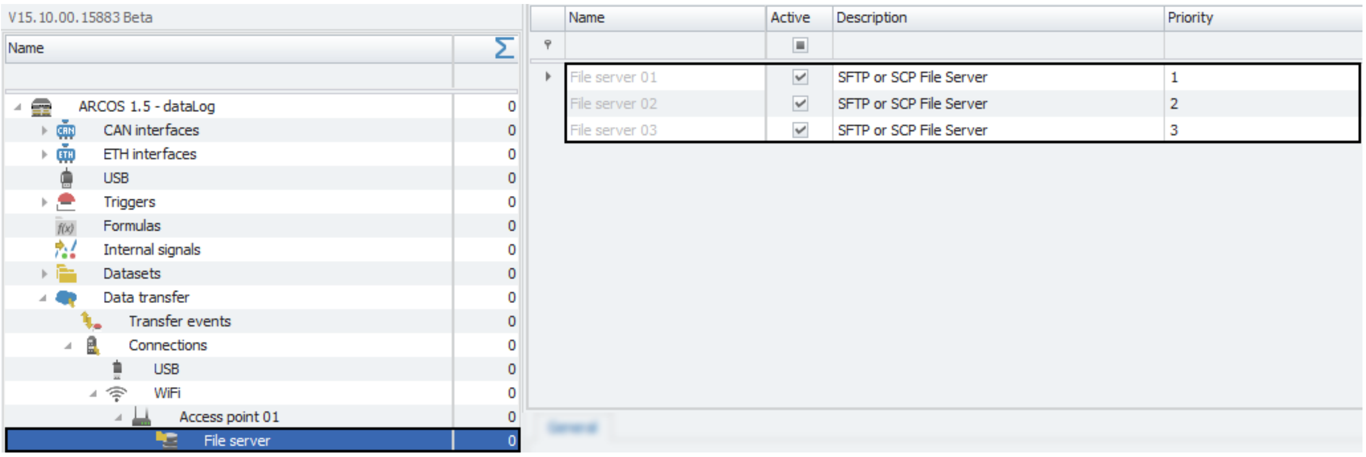
8.3 Setting up a Fileserver

Transferring data via WIFI/LAN/PPP/UMTS requires the configuration of a fileserver to which the data will be transferred. To do so, select the tree element for the connection you wish to configure: "PPP", "LAN", "Access point xx" (for WIFI connections the acccess point has to be selected instead of the WIFI connection), click the "Components" button in the Ribbon and choose "Fileserver".



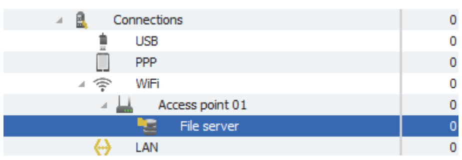
8.3.0.1 Multiple File servers

Each data transfer connection can have multiple file servers, in order to ensure successfull data transfer. By setting the priority of the file servers in the grid area, you define, which file server will be used first, starting with priority 1. As soon as data transfer via one file server has been successfully completed, the other file servers will be skipped and the dataset will be erased from the logger.



8.3.0.2 Tree elements for File servers

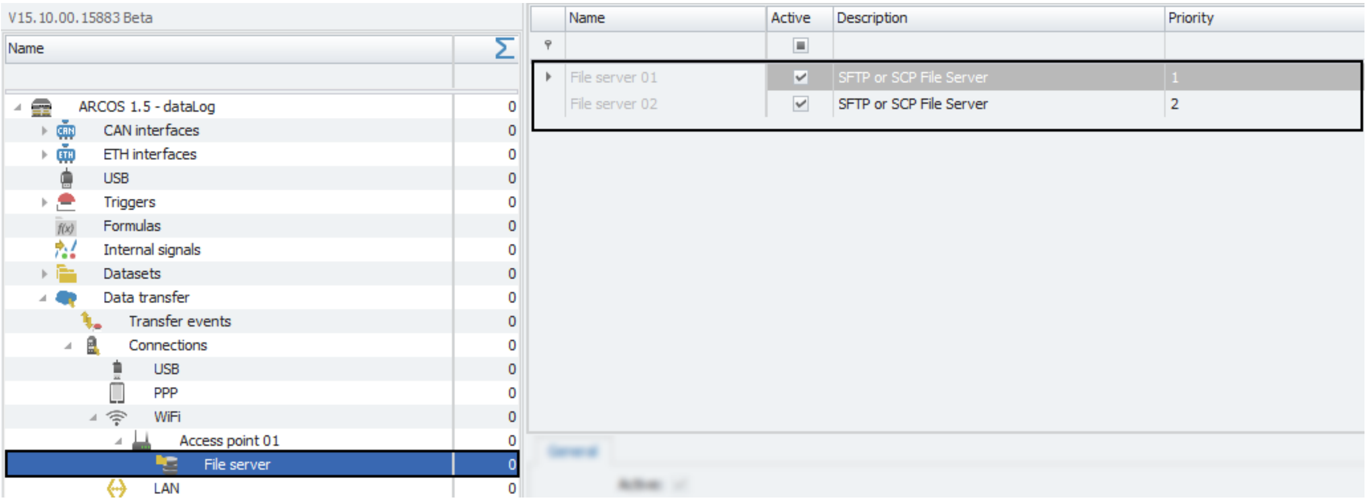
Adding a Fileserver will add one new element called “Fileserver” to the tree. It is possible to add multiple fileservers, to one connection, but they will all be found under the same tree element. An overview of the fileservers can be seen in the grid area.



8.3.0.3 Grid area for File servers

The grid area for a "Filserver" will present you with an overview of all the currently defined added fileservers. It also allows you to prioritize the various fileservers.

Also you can find here two important functions, which are the "**Column chooser**" (→4.3.1) and the "**Filter editor**" (→4.3.2).

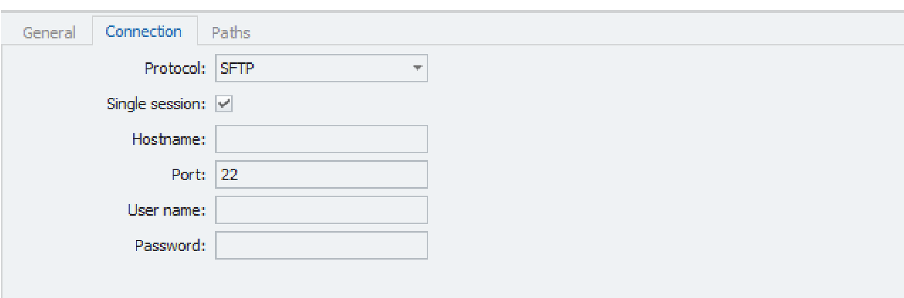


8.3.0.4 Details area for File servers

The details area provides settings for a file server. Choose the fileserver you wish to configure in the grid area of the tree element "File server" and then navigate to the details area.

General
Please refer to (→4.2.2).

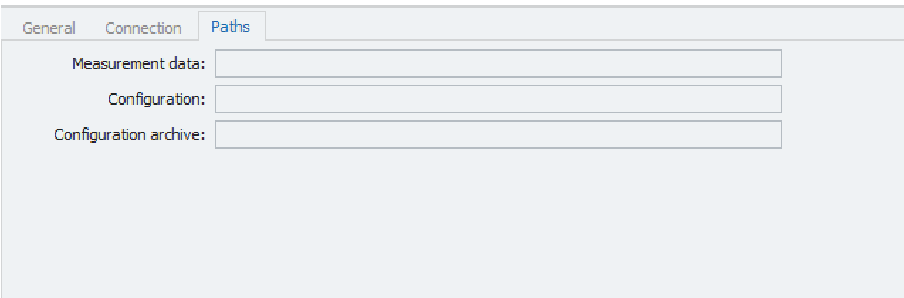
Connection
This tab contains settings for the connection.



- **Protocol**
  Select the protocol used for data transfer to the file server.
- **Single session**
  If set, the SFTP single session mode will be used.

- Hostname
  Define the IP address or hostname of the SSH file server.
- Port
  Define the port or hostname of the SSH file server.
- User name
  Fill in the user name to access the SSH file server.
- Password
  Fill in the password to access the SSH file server.

Paths
This tab allows you to set the paths for data transfer.

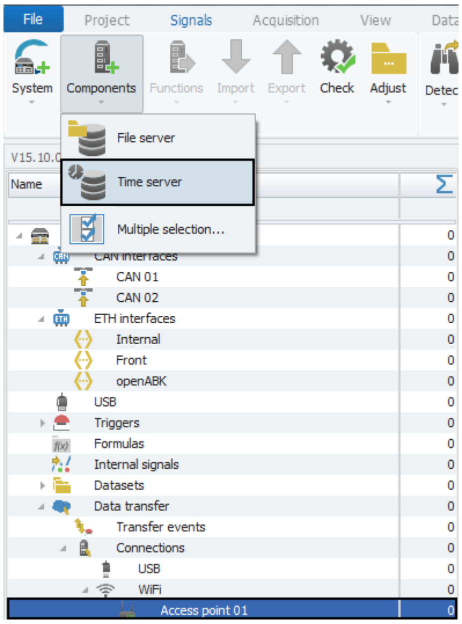


- Measurement data
  Define the path, where measurement data should be stored.
- Configuration
  Define the path, where the logger will check for a newer version of the current configuration (datalog.ccmc) and firmware.
  If there is a newer version, the logger will download it, append the current timestamp in the filename, and apply/install it at the next possible moment.
- Configuration archive (optional)
  Define the subpath for previous logger configurations and firmwares. If this subpath has been defined, the logger will copy the previously used configuration/firmware here, when he receives a newer version. If this subpath is not defined, the logger will leave the file in the "Configuration" path, that has been defined before.

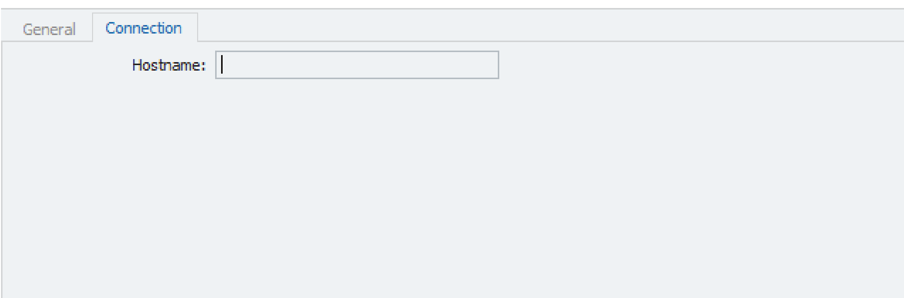
9 Setting up a time server

In order for your logger to always have the correct time, you will need to set up a time server to which the logger will connect and then synchronize its time with.

A time server can be configured for every LAN, WIFI or PPP/UMTS connection. To do so, select the tree element for the connection you wish to configure: "PPP", "LAN", "Access point xx" (for WIFI connections the acccess point has to be selected instead of the WIFI connection), click the "Components" button in the Ribbon and choose "Time server".



Then select the "Time server xx" in the grid area **of the tree element** "Time server xx" and navigate to the "Connections" tab in the details area and set the hostname/IP address in the field "Hostname".



10 Obtaining extended support

support@caetec.de
+49 8142 501365

Manual

ARCOS
µCROS

Script Environment Guide

Date of issue: May 16, 2018

CAETEC GmbH
Industriestrasse 1
82140 Olching
Germany

Pictures and sketches are for illustration purposes only and are not to be used as design drawings nor to serve as offer or assembly drawings.

All specifications are based on the technical status of May 16, 2018. We reserve the right to make any changes required to technically improve the equipment.

This manual has been produced with all due diligence.

CAETEC shall not be held liable for any damage resulting from the use of this manual, providing it is not due to gross negligence on our own part or the part of our legal representative or vicarious agent, and to the extent that the damage does not stem from personal injury, bodily harm or damage to health.

Contents

1 Using this Manual

1.1 Content

This guide explains how scripts can be applied to control the measurement procedures of the ARCOS and µCROS dataLog, and how they are integrated into existing configuration files. It also provides a compact reference list of the required commands.

1.2 Scope

This document does not explain the general structure of the dataLog configuration file (see the separate document).

1.3 Syntax Definition

In the various syntax examples, angle brackets, ‘< >’, enclose placeholders, i.e. words that are to be replaced by the user. If the placeholder must be a specific data type, this specification is enclosed in square brackets, ‘[ ]’, within the angle brackets.
Optional specifications are enclosed in double square brackets, ‘[[ ]]’.

2 General Script Statements

It is possible to create any number of script statements, with user-defined functions and variables.

2.1 Variables

Variable names must begin with a letter, but otherwise may be composed of numbers and letters. The maximum length is 40 characters.

2.1.1 Integers

Integer variables are always internally stored in signed 64-bit form. So the possible values range from -9223372036854775808 to 9223372036854775807. As variable types, 'int' and 'long' can be used equivalently. Hexadecimal values are specified with a preceding '0x', binary values with a preceding '0b'.

Syntax:	int <variable name > ((= <script expression>)); long <variable name > ((= <script expression>));
Examples:	int threehundred = 0x12C; long twohundred = 0b11001000;

2.1.2 Floating-point numbers

Floating-point numbers are always internally stored in signed 64-bit form. Possible values thus range from $\pm 4.941 * 10^{-324}$ to $\pm 1.798 * 10308$; a point '.' is used as a decimal separator. As variable types, float' and 'double' can be used equivalently. Variables can be reset by using 'NaN'.

Syntax:	float <variable name > ((= <expression>)); double <variable name > ((= <expression>));
Example:	float pi = 3.1416; double pi = 3.1416;

2.1.3 Strings

Strings are specified by means of the variable type 'string', enclosed in double quotation marks. The length of a string is theoretically infinite. The addition operator, '+', can be used to combine several strings. Length of a string can be returned by using its attribute

'length'. An interval (x:y) as a suffix for a string variable will return a sub-string from character x to character y (zero-based).
Escape sequences can be used within strings:

\n Newline
\t Tab
\\ Backslash
\' Single quote
\" Double quote

Syntax:	string <variable name> ((= <expression>)) ; <variable name>.length <variable name>(<Pos 1 (int)>:<Pos 2 (int)>)

Example:	string version = "v0.3.1-beta";
	int stringlength = version.length; // = 11
	string substring = version(1:5); // = „0.3.1"
	string str = "line1 \n line2";

2.1.4 Truth Values

A Boolean variable requires the variable type 'bool' and can have the values 'true' or 'false'.

Syntax:	bool <variable name > ((= <expression>));
Example:	bool isValid = true;

2.1.5 Signal references

The variable type 'Signal' contains a reference to a signal. All properties of the signal can be requested from the reference. The reference variable can be passed to a user-defined function as an argument.

Syntax:	Signal <variable name> ((= <signal name>));
Example:	Signal sig = 'cpuload'

2.2 Persistence

In the script, variables can be marked as persistent. When the logger is ended, their values are stored and then imported during the next start-up. There are four optional modes for the duration of the persistence:

2.2.1 P_DATASET

Stored data is deleted at the end of the dataset (max. FW 1.5).

Syntax:	P_DATASET <variable type> <variable name>;
Example:	P_DATASET int persistCounter;

2.2.2 P_CONFIG

Stored data is kept until a new configuration file is imported.

Syntax:	P_CONFIG <variable type> <variable name>;
Example:	P_CONFIG int persistCounter;

2.2.3 P_EVER

Stored data is never deleted.

Syntax:	P_EVER <variable type> <variable name>;
Example:	P_EVER int persistCounter;

2.3 Operators

2.3.1 Arithmetic Operators

Addition, '+', is defined for all data types. Two integer values yield an integer result. If at least one of the two summands is a floating-point number, then the sum is, as well. The Boolean values 'true' and 'false' are interpreted numerically, as 1 and 0, respectively. The addition operator can also be used to combine strings. If the right summand is numeric or Boolean, then its character representation is used.

The subtraction operator, '-', is defined for numeric and Boolean data types. If minuend and subtrahend are both integers, then this is also true of the result of the subtraction. In any other case, the result is a floating-point number. The Boolean values 'true' and 'false' are interpreted numerically, as 1 and 0, respectively.

The multiplication operator, '*', is defined for numeric and Boolean data types. If both factors are integers, then this is also true of the result of the multiplication. In any other case, the result is a floating-point number. The Boolean values 'true' and 'false' are interpreted numerically, as 1 and 0, respectively.

The division operator, '/', is defined for numeric and Boolean data types. If dividend and divisor are both integers, then this is also true of the result of the division. In any other case, the result is a floating-point number. The Boolean values 'true' and 'false' are interpreted numerically, as 1 and 0, respectively. Division by zero is not defined.

The modulo operator, '%', finds the remainder of an integer division and is only defined for integer operands.

2.3.2 Relational Operators

The operators equal to, '=='; and not equal to, '!=', are defined for two Boolean values, two strings and two numeric values. These three types cannot be combined.

The operators, less than, '<'; less than or equal to, '<='; greater than, '>'; and greater than or equal to, '>=', are defined for numeric values.

2.3.3 Logical Operators

The AND relation, 'AND' / '&&'; the OR relation, 'OR' / '||'; the exclusive OR relation, 'XOR'; and the negation, 'NOT' / '!', are defined for Boolean values.

2.3.4 Binary Operators

The bit-wise AND relation, '&'; the bit-wise OR relation, '|'; the bit-wise exclusive OR relation, '^'; and the bit-wise negation, '~', are defined for integer values.

2.4 Conditions and Loops

The script statement used in the following section consists of either a single expression with a trailing semicolon, or a number of expressions in curly brackets.

2.4.1 if

Specification of 'else' statements is optional.

Syntax:	if (<condition (bool)>) <statement>; ((else <statement>;))
Example:	if (isValid == true) num = 4; else num = 3;

2.4.2 while / do-while

'While' means the condition is evaluated before the body of the loop is executed; 'do-while' means it is evaluated afterward.

Syntax:	while (<condition (bool)>) <statement> do <statement> while (<condition (bool)>);
Example:	do x++; y = x * 2; while (y <= 10); while (y <= 10) {y++;}

2.4.3 for

For a 'for' loop, three parameters are specified. The first, an expression executed once at the beginning – usually – initializes a counter variable. The second, a condition checked before each loop iteration, stops the execution of the loop as soon as its result is 'untrue'. The third parameter, again an expression, is executed after each loop iteration and – generally – increments the counter.

Syntax:	for (<expression>; <condition (bool)>; <expression>)<statement>;
Example:	for (int i = 0; i < 10; i++) x = x * i;

2.5 User-defined Functions

For function names, the same conventions apply as for variables.

2.5.1 Declaration

Before each function can be used, it must first be declared in the script. Zero or any number of parameters can be passed in a list. The data type must be defined for the return value. If the function is not meant to return a value, the specification should be 'void'.

Syntax:	<return data type> <function name> (<parameter1>, <parameter2>, ..., <parameterN>) <statement>
Example:	void AddtoValue (double d) { value = value + d; }

2.5.2 Return Value

Each function can have a return value, whose type need not be declared. The key word 'return' may occur exactly once in the body of the function. Execution is stopped at this point.

Syntax:	return <expression>;
Example:	int multi (int x, int y) { return x * y; }

2.5.3 Calling Functions

Once declared, a function can be used at any subsequent place in the script. All parameters must be passed as previously defined. If the given function has a return value, it can be processed.

Example:	if (multi(5,3) != 15) error = true;

2.6 System Functions

2.6.1 Writing to the Log File or File in Dataset

The command 'printf' prints to the log file. Furthermore it is possible to write into a datafile of type 'raw' (since v2017.02) or a filewriter method (only v2016.10).
The first parameter is the target of the operation and can be either LOGFILE or the name of the target file.

Syntax:	return <expression>;
Example:	printf (LOGFILE, "Variable x has the value " + x + "."); printf('myFile', "string1\n string2");

2.6.2 System time

The command 'system time' returns the current system time and stores it in an integer variable. Defined methods can then extract from this time stamp such discrete information as milliseconds, minute, day or year. Microsecond (usec), millisecond (msec), second (sec), minute (min), hour (hour), day (day), month (month) and year (year) can be accessed.
The command 'systemtimeString' returns the system time as string.
The current measurement time (measdelay corrected system time) can be requested with the commands 'meastime' and 'meastimeString'.

Syntax:	int <variable name> = systemtime (); <variable name>.<unit of time>
Example:	int time = systemtime(); int currentHour = time.hour;

2.7 Mathematical Functions

The following functions are available for mathematical computations. The data type of the return value of each function is specified in square brackets.

Trigonometric functions (float):	cos (a), sin (a), tan (a)
Trigonometric inverse functions (float):	acos (a), asin (a), atan (a)
Hyperbola functions (float):	cosh (a), sinh (a), tanh (a)
Exponential functions and logarithms (float):	exp (a), log (a), ln (a)
Powers and square roots (float):	pow (a, b), sqrt (a)
Rounding up and down (int):	ceil (a), floor (a)
Absolute values (float):	abs (a)

2.8 Special Functions

2.8.1 Sending Bus Messages

The function ’send’ sends messages on a particular bus (see 3.2).

2.8.2 Checking Busses for Lapsed Time

The function ’check_bus’ checks whether, within a specific past time interval, messages have arrived on a particular bus. Instead of a single bus, it also possible to list several busses, separated by commas. The return is a ’bool’ type value and specifies whether traffic occurred on the bus within the specified time (in ms).

Syntax:	check_bus (<time (int)>, <bus type>, <channel number>); check_bus (<time (int)>, <bus alias>);
Example:	bool alive = check_bus (1000, ’PT-CAN’, CAN, 2, ’FR1’);

2.8.3 User-defined Events

Specified anywhere in the script, the function ’userevent’ stores user-defined pairs, consisting of a name and a pertinent value, in an internal data structure. These entries can be processed, as required, by custom dataLog extensions and written to separate output files. The first parameter must be a ’string’ type, the second can be any data type. Sources of the latter can be variables, constants or even signal values.

Syntax:	userevent (<name (string)>, <value>);
Example:	userevent(”Level of TestTrigger”, TestTrigger.level);

2.8.4 ASCII Conversion

The function ’toChar’ converts the ASCII values of printable characters into the equivalent characters. The first parameter must be a type ’int’ and represent the ASCII value of a printable character. The return type is ’string’.

Syntax:	toChar(<ASCII value (int)>);
Example:	string A = toChar(0x41);

2.8.5 Char Conversion (decimal)

With the 'toASCII' function numeric variables can be converted into a formatted string. For integers, the number of digits is specified in floating point numbers, the total width and the amount of decimal places contained therein.

Syntax:	toASCII (<number/variable (int)>, <output width (int)>); toASCII (< number/variable (float)>, < output width (int)>, <decimal places (int)>);
Example:	string A = toASCII(intVar, 4); string B = toASCII(floatVar, 8, 2);

2.8.6 Char Conversion (hexadecimal) (from scriptEngine 1.6.3)

With the 'toHexFull' function integer variables can be converted into a hexadecimal formatted string. The optional secend parameters can contain the minimum width.

Syntax:	toHexFull(<number/variable (int)>, [(<output width (int)>)]);
Example:	string A = toHexFull(intVar, 8);

2.8.7 Conversion from string to integer

With the 'toInteger' function a string can be converted into an integer containing the ASCII values of the string characters, e.g. 'toInteger("ABCD")' returns 0x44434241. At most the first eight characters of the string are converted.

Syntax:	toInteger (<text/variable (string)>);
Example:	int A = toInteger(„ABCD");

2.8.8 Shutting Down the System

The function 'shutdown_system' shuts down the data logger. Any normally performed data transmissions are omitted.

Syntax:	shutdown_system ();

2.8.9 Switching Power Out (supported only by the CAN PM interface)

The function 'set_bus_power' switches the separate power supply of a particular CAN channel. This function is only supported by the ARCOS CAN PM (Power Management) slide-in module.

Syntax:	set_bus_power (<bus type>, <channel number>, <status (bool)>); set_bus_power (<bus alias>, <status (bool)>);
Example:	set_bus_power (CAN, 12, true); set_bus_power ('PT_CAN', true);

2.8.10 Checking Signal Names

The function 'isValidSignal' checks whether a signal with a given name exists. This indicates whether an external configuration is loaded, for example. The return type is 'bool'.

Syntax:	isValidSignal<signal name (string)>);
Example:	bool extSignalPresent = isValidSignal('extSignal');

2.8.11 Checking Method Names

The function 'isValidMethod' checks whether a method with a given name exists. This indicates whether an external configuration is loaded, for example. The return type is 'bool'.

Syntax:	isValidMethod(<method name (string)>);
Example:	bool extMethodPresent = isValidMethod('extMethod');

2.8.12 Checking Datafile Names

The function 'isValidDatafile' checks whether a datafile with a given name exists. This indicates whether an external configuration is loaded, for example. The return type is 'bool'.

Syntax:	isValidDatafile(<datafile name (string)>);
Example:	bool extDatafilePresent = isValidDatafile('extDatafile');

2.8.13 Regulate Backlight

With the 'hid_SetBacklight' function, the backlight of an attached ABK screen can be controlled.

Syntax:	void hid_SetBacklight (<brightness (in %) (float)>)
Example:	hid_SetBacklight (50);

2.8.14 Show Display Limitwarning

With the 'hid_limitwarning' function, a warning can be displayed on an attached ABK screen. The parameters "triggering signal" and "limit value" are not evaluated, but simply passed to the screen.

Syntax:	void hid_limitwarning (<priority (0-255) (int)>, <class name (string)>, <triggering signal>, <limit value (float)>, <additional parameters (arbitrary number)>)
Syntax additional parameters: (<key (string)>, <value (string)>)	
Example:	hid_limitwarning(1, "", 'cpu_temp', 80, "instruction_1", "WARNING", "instruction_2", "CPU temperature is too high");

2.8.15 Special Tags

With the 'specialtagvalue' function it is possible to access tags that obtain special values at runtime. The following tags are available:

- fn: Frontnumber of the logger
- vin: Value of the signal with special role "vin" (special=vin)
- odo: Value of the signal with special role "odo" (speical=odo)

The return value of all tags is of type string.

Syntax:	string specialtagvalue(<tag (string)>)
Example:	string frontnumber = specialtagvalue("fn");

2.9 Comments

A comment in the script begins with the character combination slash - asterisk, '/*', and ends with asterisk - slash, '*/'. Alternatively, a single-line comment begins with a pair of slashes, '//', and ends automatically at the next line break.

3 Sending on Busses

3.1 Bus Messages

Bus messages are first declared, then subsequently sent. The exact 'value' of a bus message is calculated at the time it is sent.

Sources can be Boolean or numeric variables, constants or the raw values of signals. If not all the bits of a value are to be used, an interval can be specified, enclosed in square brackets, (<highest bit>:<lowest bit>). The bits are always counted from 0 (the least significant bit). Boolean variables are sent as a single bit. Floating-point values are sent as a 32-bit float – followed by the specification 'float' enclosed in square brackets – or, without this specification, as a 64-bit double.
If no data length is specified, it is computed from the used sources.
Warning: Numeric constants and variables have an internal bit length of 64. This means, if no interval is specified, they fill a complete CAN message!

Syntax:	MESSAGE <message name> (((<data length in bytes (int)>))) { <sources > }
Example:	MESSAGE testMsg (8) { longY(23:0), doubleX(float), 1(7:0) }

3.2 Sending on a CAN Bus

The system function 'send' can be used to send data (bus message or integer variable) on the CAN bus. The byte data length defined in the bus message is converted to the appropriate DLC, which differs from the length for CAN-FD messages with more than eight bytes. If the message is too short it is filled with zeros. The parameter 'ext. ID' can have the values 0 and 1 and specifies whether the following ID is from the extended range.

Syntax:	send (<bus message >, <CAN bus alias>, <ext. ID>, <ID>); send (<data (int)>, CAN, <channel number>, <ext. ID>, <ID>);
Example:	send (testMsg, CAN, 8, 0, 264); send (intVar, 'FA-CAN', 1, 4301);

3.3 Sending on FlexRay/LIN Bus

The system function 'send' can be used to send data (bus message or integer variable) on the FlexRay and (since v2017.10) LIN bus. The DLC contained in the bus message is applied when sending. If the parameter 'oneshot' is left out or set to true, the message is sent only once. Otherwise it is repeated until another message is sent on the same ID.

Syntax:	send (send (<bus message>, <FlexRay/LIN bus alias >, <ID> ((, <oneshot (bool)>))); send (<data (int)>, FLEXRAY/LIN, <channel number>, <ID> ((, <oneshot (bool)>)));
Example:	send (testMsg, LIN,1, 0x100); send (intVar, 'FR1', 1392, false);

3.4 Sending of ASCII-Texts or Numeric Values (CAN-Bus only)

The system functions 'send_text_to_can' and 'send_number_to_can' can be used to send data (ASCII-texts, integer variables or bus messages) on a CAN bus.

Unique to these functions is that it is possible to specify a delay parameter (in ms). With this the system ensures to respect a send pause between two consecutive transmissions of the specified delay value. This is to make sure that i.E. receiver with limited processing power won't miss any message.

ASCII-Texts get Sent with multiple messages of up to 8 byte. Once the remaining rest of the text to be sent is less than 8 bytes, the last messege will be transmitted with a DLC exactly matching the remaining number of bytes to be sent.
Numeric variables always get sent with a DLC of 8 as all interger script variables have a 64-bit representation internally.
If 'send_number_to_can' is used with a previously defined message insted of an integer variable, it will respect the DLC defined with the definition of the bus message.

The parameter 'ext. ID' can have the values 0 and 1 and specifies whether the following ID is from the extended range.

Syntax:	send_text_to_can (<string>, <CAN bus alias>, <ext. ID>, <ID>, <delay>); send_text_to_can (<string>, CAN, <channel number>, <ext. ID>, <ID>, <delay>); send_number_to_can (<bus message>, <CAN bus alias>, <ext. ID>, <ID>, <delay>); send_number_to_can (<data (int)>, CAN, <channel number>, <ext. ID>, <ID>, <delay>);
Example:	send_text_to_can ("Hello World", CAN, 8, 0, 264, 10); send_number_to_can (intVar, 'FA-CAN', 1, 4301, 10);

4 Signals

4.1 Accessing Signal Properties

Signal proerties can be accessed using the syntax <Signal>.<Property>. <Signal> is either the name of the signal or a signal reference variable.

<Signal>.value	value of the signal (float/string)
<Signal>.stringvalue	string representation of the signal (e.g. used with vtabs) (string)
<Signal>.unit	unit (string)
<Signal>.raw	raw value of the signal (int)
<Signal>.name	signal name (string)
<Signal>.scale	factor of the comput. method (float)
<Signal>.offset	offset of the comput. method (float)
Example:	int sigraw = 'Signal1'.raw;

4.2 Script Signals

Aside from 'real' signals from bus sources, computed signals can be generated from parts of the script. Using the specified sample rate (in Hz), the script expression is computed and made available as a signal value. This signal can then be recorded by a measurement method or presented in the display. Additionally an optional argument can be used to mark the signal as having a special meaning for the logger. Currently the roles "vin", "odo", "gps_lat", "gps_lon" and "gps_ele" are supported.

Syntax:	SIGNAL <signal name> (<sample rate (int)>, <expression>, [(<special_role (string)>)]); SIGNAL <signal name> (<unit (string)>, <sample rate (int)>, <expression>, [(<special_role (string)>)]);
Examples:	SIGNAL varX (100, intVarX); SIGNAL speedMph ("mph", 1, speed); SIGNAL odosig ("odosignal", 1, intVarX, "odo");

4.3 Digital Output Signals

The script can also be used to control the digital output channels of the logger. The script expression is processed at the specified sample rate (in Hz). If the expression returns 'true', the output channel specified by 'channel number' is switched.

Syntax:	DIGITAL_OUT <signal name> (<channel number (int)>, <sample rate (int)>, <expression>);
Example:	DIGITAL_OUT DigOut2 (2, 10, digOut == true);

5 Triggers

Triggers can be defined as single-level triggers, with an optional continuity specification, or as multi-level, finite state machines. Triggers have a duration – they remain active for as long as the triggering condition is met.
All times must be specified in ms. Passive triggers aren't processed. Timeouts, pre or post-times, as well as trigger group specifications are optional. Any trigger groups that have not been previously declared are automatically created. The individual elements must be specified in the prescribed order.

5.1 Single-Level

In the case of single-level triggers, the continuity option specifies that the trigger is only activated if the trigger condition remains met for the specified duration.

Syntax:

```
TRIGGER <trigger name> (<active (bool)>) {
TERM = (<expression (bool)>);
[( CONTINUITY = <time (int)>; )]
[( TIMEOUT = <time (int)>; )]
[( PRETIME = <time (int)>; )]
[( POSTTIME = <time (int)>; )]
[( GROUPS (<trigger groups>); )]
}
```

Example:

```
TRIGGER test1 (true) {
TERM = (x > 10);
CONTINUITY = 5000;
PRETIME = 10000;
POSTTIME = 500;
GROUPS (DEFAULT, 'SpecialGroup_A');
}
```

5.2 Multi-Level

Multi-level triggers can theoretically comprise an infinite list of levels. The starting level is always 0; only one level can be active at any given time. Both the set condition of the next higher and the reset condition of the current level are checked. If necessary, a level change takes place.

```
Syntax:     TRIGGER <trigger name> (<active (bool)>) {
            [[ TIMEOUT = <time (int)>; ]]
            [[ PRETIME = <time (int)>; ]]
            [[ POSTTIME = <time (int)>; ]]
            [[ GROUPS (<trigger groups>); ]]
            <level list>
            }

Example:    TRIGGER test2 (true) {
            PRETIME = 10000;
            POSTTIME = 500;
            GROUPS (DEFAULT)
            LEVEL (1)  SET (x == 2); RESET (x <= 1);
            LEVEL (2)  (x > 2); TIMEOUT = 500;
            LEVEL (3)  SET (x > 10); RESET (true);
            }
```

5.3 Levels

The specifications timeout, name and reset condition are optional. Levels needn't be consecutively numbered and are – if necessary – sorted in ascending order. If the set condition is defined without specifying 'SET', the negation of the expression acts as the reset condition. If 'true' is specified as reset condition, the level resets as soon as it becomes active.

```
Syntax:     LEVEL (<level number (int)> [[, <level name>]] ){
            SET (<expression (bool)>);
            [[ RESET (<expression (bool)>); ]]
            [[ TIMEOUT = <time (int)>; ]]
            }
            LEVEL (<level number (int)> [[, <level name>]] )
            {
            <expression (bool)>;
            [[ TIMEOUT = <time (int)>; ]]
            }
Example:    See Section 5.2.
```

5.4 Groups

Trigger groups are logical relations between several individual triggers. Default is an OR relation. To specify an AND relation, the group must be explicitly declared. Permitted values for the mode are 'AND' and 'OR'.

Syntax:	TRIGGER_GROUP (<group name>, <mode>);
Example:	TRIGGER_GROUP ('TEST_GRP3', and);

5.5 Accessing States

From the outside, it is possible to access the trigger and its states in the following ways:

<trigger name>	trigger active (bool)
<trigger name>.<level number>	level active (bool)
<trigger name>.level	active level (int)
<trigger name>.pre	pre-trigger time (int)
<trigger name>.post	post-trigger time (int)
<trigger name>.count	trigger event counter (int)

It is possible to modify the trigger event counter from the script environment:

Syntax:	<trigger name>.count = <trigger event counter (int)>;
Example:	'Trig_Speed'.count = 5;

5.6 Persistence

The event counter for a trigger can be assigned a multi-level persistence – if not otherwise specified, it persists for the duration of a data set.
The desired persistence level P_XXX (cf. Section 2.2) precedes the key word TRIGGER.

Example:	P_CONFIG TRIGGER trigger name (true) { TERM = (isValid);}

6 Methods and datafiles

6.1 Accessing States

From the script it is possible to access the states of a method or datafile in the following ways:

<method name>	recording active (bool)
<method name>.pre	pre-trigger time (int)
<method name>.post	post-trigger time (int)
<method/datafile name>.datafiles	number of files in the current dataset (int)
<method/datafile name>.datasize	size of the current file in bytes (int)

For methods or datafiles using snapshot mode (Snapshot – see the dataLog manual), the following additional parameters are available:

<method/datafile name>.snapshots	number of markers (int)
<method/datafile name>.snapshotfiles	number of marked files (int)
<method/datafile name>.snapshotsize	total size of marked files (int)

7 Signal engines

7.1 Accessing States

From the script it is possible to access the states of signal engine stations and their DAQ lists in the following ways:

<station name>.connected	connection state (0: disconnected, 1: connected, NULL: uninitialized) (double)

For CCP and XCP stations, the following additional parameters are available:

<station name>.versioncheck_result	result of the versioncheck (0: failed, 1: successful, NULL: not yet checked) (double)
<station name>.versioncheck_version	current version string (only set if versioncheck was successful) (string)
<station name>.<DAQ list name>.configured	true when the DAQ list was successfully configured (bool)
<station name>.<DAQ list name>.started	true when the DAQ list was successfully started (bool)

8 Storage Targets

8.1 Accessing States

It is possible to access size information about connected storage media from within a script. In order to be able to do so, the requested partition must have been given a unique name or *label* beforehand. The following information is then provided:

<label>.total	total size of partition in MB (double)
<label>.used_mb	currently used size of partition in MB (double)
<label>.used	currently used size of partition in % (double)

Note that this information is only available while the partition is actually used.

9 Gateway

A gateway relays all the messages from one or more source busses to one or more target busses. By using an ID filter and the appropriate number of filter rules, it is possible to include or exclude individual IDs or entire ranges of IDs.
Warning: Only CAN busses are currently supported as sources or targets.

Syntax:	GATEWAY <name> { <source(s)> <target(s)> }

9.1 Source

One or more busses can be specified as a source for a gateway. Busses are addressed either by using the combination of bus type and channel number, or by means of the respective alias. If no ID filter is specified, all messages are relayed.

Syntax:	SOURCE (<bus>); SOURCE (<bus>, <channelnum>);

9.2 Source with ID Filter

A source bus can be combined with a filter to restrict the IDs that are sent on. Permitted values for the filter action are 'include' and 'exclude'. The standard action is performed on all IDs to which none of the specified rules apply.

9.2.1 General Format

Syntax:	SOURCE (<bus>) {
	<filter rules>
	DEFAULT = <action>;}
	
	SOURCE (<bus>, <channelnum>) {
	<filter rules>
	DEFAULT = <action>;}

9.2.2 Filter rules

Filter rules are applied in the order they are defined in the script. The first pertinent rule is applied . 'Ext. ID' can have the values '0' or '1', and specifies whether the following IDs are from the extended range.

Syntax:	FILTERRULE (<action>, <ext. ID>, <ID>); FILTERRULE (<action>, <ext. ID>, <StartID>, <En- dID>);

9.3 Target

One or more busses can be specified as a target for a gateway. The busses can be addressed using either the combination bus type and channel number, or by means of the respective alias. An ID filter is not supported.

Syntax:	TARGET (<bus>); TARGET (<bus>, <channelnum>);

9.4 Example

The following example generates a gateway for relaying messages from CAN 1 to the FA-CAN. The IDs 0x64 bis 0xc7 – except for 105 – are to be filtered out.

Code:

```
GATEWAY testGW_CAN1_FACAN {
SOURCE (CAN, 1) {
FILTERRULE (include, 0, 105);
FILTERRULE (exclude, 0, 0x64, 0xc7);
DEFAULT = include;
}
TARGET ('FA-CAN');
}
```

10 Events

All times are specified in ms. It is always possible to use a bus type / channel number tuple, instead of a bus alias. Unless otherwise specified, events always occur synchronized to the rate of measurement (1kHz).

10.1 Cyclically Occurring Events

Syntax:	EVENT <event name> CYCLE (<time (int)>);
Example:	EVENT cycle_3s CYCLE (3000);

10.2 Free Script Event

A free script event occurs if the Boolean expression yields 'true'.

Syntax:	EVENT <event name> FREE (<expression (bool)>);
Example:	EVENT free_isValid FREE (isValid);

10.3 State Change

The events SET and RESET are comparable in their behavior to free events, except that they only activate once, when the expression yields true (for SET) or untrue (for RESET).

Syntax:	EVENT <event name> SET (<expression (bool)>); EVENT <event name> RESET (<expression (bool)>);
Example:	EVENT free_becomeValid SET (isValid); EVENT free_becomeInvalid RESET (isValid);

10.4 Received Events

Asynchronous events occur when a signal is received – either an unspecified message on a specific bus type, an unspecified message on a specific bus, or a specific message on a specific bus. Aside from specifying a concrete ID, it is also possible to specify the upper and lower limits of a range of IDs.
In all frame-based receive events (from CAN or LIN bus) the message content can be

accessed via the integer variable "_MESSAGE" and the current frame id via "_ID". The latter is useful when receiving events on ID ranges or all buses. How to get the signal value in an signal-based receive event see subsection 4.1 Accessing Signal Properties.

Syntax:	EVENT <event name> RECEIVE (<signal name>); EVENT <event name> RECEIVE (<bus type>); EVENT <event name> RECEIVE (<bus alias>); EVENT <event name> RECEIVE (<CAN bus alias>, <ext. ID>, <ID>); EVENT <event name> RECEIVE (<CAN bus alias>, <ext. ID>, <ID1>, <ID2>); EVENT <event name> RECEIVE (<LIN bus alias>, <ID>); EVENT <event name> RECEIVE (<LIN bus alias>, <ID1>, <ID2>); EVENT <event name> RECEIVE (<FlexRay bus alias>, <ID>, <CycleDiv>, <CycleMod>); EVENT <event name> RECEIVE (<FlexRay bus alias>, <ID1>, <ID2>, <CycleDiv>, <Cycle-Mod>);
Example:	EVENT recvTemp1 RECEIVE ('temp1sig'); EVENT recvCAN RECEIVE (CAN); EVENT recvFR1 RECEIVE (FlexRay, 1); EVENT recvTempMsg RECEIVE ('PT-CAN', 0, 101); EVENT recvCAN3_100_199 RECEIVE (CAN, 3, 0, 0x64, 0xc7);

10.5 Timeout

This event occurs when, for a specified time period, messages are not received – either any messages on a specific bus type, any messages on a specific bus, or a specific message on a specific bus.
The event also occurs if the specified message has never been received. Once it has occurred, the event remains inactive until the message is received again.

Syntax:	EVENT <event name> TIMEOUT (<time (int)>, <bus type>); EVENT <event name> TIMEOUT (<time (int)>, <bus alias>); EVENT <event name> TIMEOUT (<time (int)>, <CAN bus alias>, <ext. ID>, <ID>); EVENT <event name> TIMEOUT (<time (int)>, <LIN bus alias>, <ID>); EVENT <event name> TIMEOUT (<time (int)>, <FlexRay bus alias>, <ID>, <CycleDiv>, <Cycle-Mod>);
Example:	EVENT toCAN TIMEOUT (1000, CAN); EVENT toFR1 TIMEOUT (1000, FlexRay, 1); EVENT toTempMsg TIMEOUT (1000, 'PT-CAN', 0, 101);

10.6 Automatic Timer

The timer is automatically activated once (!) when the specified condition remains met for the specified period of time. The name of the timer can be used to query its state; for example, in order to use it in a trigger condition.

Syntax:	EVENT <event name> TIMER (<expression (bool)>, <time (int)>); TIMER <event name> (<expression (bool)>, <time (int)>);
Example:	EVENT timer_xOddFor2sec TIMER ((x % 2 != 0), 2000); TIMER timer_xEvenFor2sec ((x % 2 == 0), 2000);

10.7 Manual Timer

The manual timer must be started; once the period of time expires, it activates once (!). Stopping it resets the time to 0.

Syntax:	EVENT <event name> TIMER (<time (int)>); <event name>.start <event name>.stop
Example:	EVENT timer_2sec TIMER (2000); timer_2sec.start; timer_2sec.stop;

10.8 Trigger Event

The event occurs once (!) when the corresponding trigger is activated.

Syntax:	EVENT <event name> TRIGGER (<trigger name>);
Example:	EVENT trig1_started TRIGGER ('trig1');

10.9 Value Change

This asynchronous event occurs when the value of a specific signal or of a specific message on a specific bus changes.

Syntax:	EVENT <event name> UPDATE (<signal name>); EVENT <event name> UPDATE (<CAN bus alias>, <ext. ID>, <ID>); EVENT <event name> UPDATE (<LIN bus alias>, <ID>); EVENT <event name> UPDATE (<FlexRay bus alias>, <ID>, <CycleDiv>, <CycleMod>);
Example:	EVENT updTemp1 UPDATE ('temp1sig'); EVENT updTempMsg UPDATE ('PT-CAN', 0, 101);

10.10 System Events

System events are used for starting up and shutting down the system, as well as for starting and ending a measurement.

Syntax:	EVENT <event name> SYSTEM (STARTUP); EVENT <event name> SYSTEM (SHUTDOWN); EVENT <event name> SYSTEM (MEASBEGIN); EVENT <event name> SYSTEM (MEASEND); EVENT <event name> SYSTEM (RECORDINGBEGIN); EVENT <event name> SYSTEM (RECORDINGEND);

10.11 OpenABK Key Events

For the combination of keys on the openABK input device with actions following events are available. If no device ID is specified, the assignment is valid for each connected input device.

Syntax:	EVENT <event name> HID_KEY (<device-ID (int)>, <key code>, <key event>, <time (int)>); EVENT <event name> HID_KEY (<key code>, <key event>, <time (int)>);
Key Codes:	HID_KEY_ROTATE_LEFT HID_KEY_ROTATE_RIGHT HID_KEY_ROTATE_PUSH HID_KEY_PAGE_LEFT HID_KEY_PAGE_RIGHT HID_KEY_PAGE_UP HID_KEY_PAGE_DOWN HID_KEY_MENU HID_KEY_BACK HID_KEY_F1 HID_KEY_F2 HID_KEY_TRIGGER
Key Events:	HID_KEY_EVENT_PRESS HID_KEY_EVENT_HOLD HID_KEY_EVENT_RELEASE

11 Actions

For any event, the script can specify any number of associated actions that are executed when the event occurs.
The following script statement consists of either a single expression, with a trailing semicolon; or a series of expressions, in curly brackets.
If event and action are directly combined, then there is no need to specify an event name.

Syntax:	ON <event name> <statement> ON EVENT <event name> <event type> (<event parameter>) <statement> ON EVENT <event type> (<event parameter>) <statement>

	ON outside_hot warnDriver();
	ON upd_indoor_temp {
	doSomething();
	printf (LOGFILE, "Temp = " + indoor_temp.value);}
	ON EVENT trig1_started TRIGGER ('trig1') x = 3.124325;
	ON EVENT CYCLE (2000) { doSomethingElse(); }

12 Alive Counters

Alive counters sent by control units can be monitored and their states checked whenever required.

12.1 Declaration

In addition to the source message specification, the start and end values of the expected range, an error value, and a timeout (in milliseconds) are specified.
Instead of "bus type, channel number", a bus alias can be used.
Instead of defining a new signal source, an existing signal can also be specified. If the signal has its own timeout, a new specification of a timeout is optional.
However, if a timeout different from signal-specific one is to be applied, or if the signal is to have no timeout, specifying a different value creates a copy of the signal with the pertinent timeout.

Syntax:	CHECKALIVE <name> (CAN, <channel number>, <ext. ID>, <ID (int)>, <offset (float)>, <scale (float)>, <bit offset (int)>, <bit length (int)>, <data format>, <CAN format>, <start value (int)>, <end value (int)>, <error value (int)>, <timeout (int)>); CHECKALIVE <name> (LIN, <channel number>, <ID (int)>, <offset (float)>, <scale (float)>, <bit offset (int)>, <bit length (int)>, <data format>, <LIN format>, <start value (int)>, <end value (int)>, <error value (int)>, <timeout (int)>); CHECKALIVE <name> (FlexRay, <channel num>, <ID (int)>, <CycleDiv (int)>, <CycleMod (int)>, <offset (float)>, <scale (float)>, <bit offset (int)>, <bit length (int)>, <data format>, <FlexRay format>, <start value (int)>, <end value (int)>, <error value (int)>, <timeout (int)>); CHECKALIVE <name> (<signal name>, <start value (int)>, <end value (int)>, <error value (int)> [(, <timeout (int)>)]);
Example:	CHECKALIVE chk_alv_test1 (CAN, 1, 0, 100, -65.0, 0.2, 32, 16, unsigned, intel, 1, 254, 255, 1000); CHECKALIVE chk_alv_test2 ('alv_counter_sig', 1, 14, 15, 500);

12.2 CheckAlive

Whenever required, the name of the CheckAlive object can be used to check the validity of the last value. The name can be used analogously to a Boolean variable.

Example:	CHECKALIVE chk_alv_test (...); if (chk_alv_test == false) anErrorFunction();

13 CAN display protocol

This protocol was developed in cooperation with BMW EG-7 to transmit signal values to a CAN-based display.

13.1 Frame composition

Composition of a CAN frame:

Byte	0	1	2	3	4	5	6	7
Content	VarNum	Command	Data					

Composition of command byte:

Byte	0	1	2	3	4	5	6	7
Content	Attributes				Reserved		Deleted	Commit

Available attributes are:

- 0x0 -> current value
- 0x1 -> minimum value
- 0x2 -> maximum value
- 0x8 -> signal name
- 0x9 -> display name
- 0xA -> unit
- 0xB -> description

13.2 Script Commands

IDs have to be unique and >0.

13.2.1 Create CAN display

Creates CAN display with DisplayID.

Syntax:	createCanDisplay(<DisplayID>, <DisplayName>, <BusAlias>, <ExtID>, <CANID>) createCanDisplay(<DisplayID>, <DisplayName>, CAN, <BusChannelNum>, <ExtID>, <CANID>)
Example:	createCanDisplay(0x01, "MKT View II", 'RemoteCAN', 0, 0x14); createCanDisplay(0x01, "MKT View II", CAN, 8, 0, 0x14);

13.2.2 Configure CAN display

Sets up parameter for minimum send cycle, decimal symbol and fill characters for NaN values.

Syntax:	setCanDisplayProperty(<DisplayID>, <PropertyName>, <PropertyValue>) setCanDisplayProperty(<DisplayName>, <PropertyName>, <PropertyValue>)
PropertyName = "min_send_cycle" | "decimal_symbol" | "no_value_fill_character"	
Example:	setCanDisplayProperty(0x01, "min_send_cycle", "1"); setCanDisplayProperty("MKT View II", "min_send_cycle", "1");

Note: Parameter 'PropertyValue' will always be stated as string in double exclamation marks.

13.2.3 Signal groups

Creates a group for signals with unified update rate.

Syntax:	createDisplaySignalGroup(<DisplayID>, <GroupID>, <GroupName>) createDisplaySignalGroup(<DisplayName>, <GroupID>, <GroupName>)
Example:	createDisplaySignalGroup(0x01, 0x01, "1 Hz"); createDisplaySignalGroup("MKT View II", 0x01, "1 Hz");

13.2.4 Display signals

Creates a display signal and sets up display signal name, value width, decimal places, display unit and display factor.

Syntax:	createDisplaySignal(<GroupID>, <SignalID>, <SignalName>, <DisplayedSignalName>, <ValueWidth>, <DecimalPlaces>) createDisplaySignal(<GroupName>, <SignalID>, <SignalName>, <DisplayedSignalName>, <ValueWidth>, <DecimalPlaces>) createDisplaySignal(<GroupID>, <SignalID>, <SignalName>, <DisplayedSignalName>, <DisplayedUnit>, <Factor>, <ValueWidth>, <DecimalPlaces>) createDisplaySignal(<GroupName>, <SignalID>, <SignalName>, <DisplayedSignalName>, <DisplayedUnit>, <Factor>, <ValueWidth>, <DecimalPlaces>)
Example:	createDisplaySignal("1 Hz", 0x01, 'measdelay', "Measurement delay", 8, 2); createDisplaySignal(0x01, 0x01, 'measdelay', "Measurement delay", "s", 0.001, 8, 5);

13.2.5 Send information for a signal

Sends chosen attribute for a single signal. Available attributes are current value, min/max values, signal name, display name, unit and description.

Syntax:	sendDisplaySignal(<DisplayID>, <SignalID>, <AttributeName>) sendDisplaySignal(<DisplayName>, <SignalID>, <AttributeName>)
AttributeName = "current" | "min" | "max" | "signalname" | "displayname" | "unit" | "description"	
Example:	sendDisplaySignal(0x01, 0x04, "current"); sendDisplaySignal("MKT View II", 0x04, "current");

13.2.6 Send information for a signalgroup

Sends chosen attribute for all signals in a group. Available attributes are current value, min/max values, signal name, display name, unit and description.

Syntax:	sendDisplaySignalGroup(<DisplayID>, <GroupID>, <AttributeName>) sendDisplaySignalGroup(<DisplayName>, <GroupName>, <AttributeName>)

AttributeName = "current" I "min" I "max" I "signalname" I "displayname" I "unit" I "description"

Example:	ssendDisplaySignalGroup(0x01, "1 Hz", "displayname"); sendDisplaySignalGroup("MKT View II", 0x01, "displayname"

14 Application Example

The following section demonstrates the application of a simple CAN gateway, used to manipulate the data. There are three options for relaying a message: upon receipt of the source data; only when the signal value has changed; or at fixed, variant frequency.

This example specifies relaying the message immediately upon receipt:

```
ON EVENT gw1 RECEIVE (CAN, 3, 0, 100) {
send (_MESSAGE, CAN, 5, 0, 273);}
```

In the following example, the message is relayed upon receipt only if the value has changed:

```
ON EVENT gw2 UPDATE (CAN, 3, 0, 101) {
send (_MESSAGE, CAN, 5, 0, 272); }
```

And here, the currently valid value is sent cyclically:

```
long gw3_tmp_message;

ON EVENT gw3_recv RECEIVE (CAN, 1, 0, 752) {
gw3_tmp_message = _MESSAGE; }

ON EVENT gw3_cycl CYCLE (250) {
send (gw2_tmp_message, CAN, 3, 0, 43); }
```

15 Appendix

15.1 Embedding in the Configuration File

Scripts are embedded in the configuration file 'datalog.cfg' in the following manner. Theoretically, any number of scripts can be thus embedded, one after another.

```
<datalog_cfg>
        <settings>
           ...
        </settings>
        <signals>
           ...
        </signals>
        <triggers>
           ...
        </triggers>
        <scripts>
           <script>
              name=scriptname
                 <scriptbody>
                          scriptcode
                 </scriptbody>
           </script>
              ...
        </scripts>
        <methods>
           ...
        </methods>

</datalog_cfg>
```

15.2 Reserved Words

The following terms (in alphabetical order) are reserved and must never be used, either as identifiers for variables or functions, or as any other type of name:

abs, acos, AND, asin, atan, bool, BUS, can, Can, CAN, ceil, CHECKALIVE, cos, cosh, CYCLE, DEFAULT, do, double, else, EVENT, exp, false, FILTERRULE, flexray, Flexray, FlexRay, float, floor, for, FREE, GATEWAY, GROUPS, HID_KEY, if, int, isValidMethod, isValidSignal, LEVEL, lin, Lin, LIN, ln, log, LOGFILE, long, MESSAGE, ON, OR, POSTTIME, pow, PRETIME, print, printf, P_CONFIG, P_DATASET, P_EVER, P_TRANSFER, RECEIVE, RESET, return, send, SET, set_bus_power, shutdown_system, Signal, SIGNAL, sin, sinh, SOURCE, sqrt, string, SYSTEM, TARGET, TIMEOUT, TIMER, toAscii, toChar, toInteger, TRACE, TRIGGER, TRIGGER_GROUP, true, UPDATE, VOID, while, XOR

Manual

ARCOS

µCROS

Hardware and Configuration

Version 1.4

Stand: 11/18/2016

CAETEC GmbH
Industriestrasse 1
82140 Olching
Germany

Pictures and sketches are for illustration purposes only and are not to be used as design drawings nor to serve as offer or assembly drawings.

All specifications are based on the technical status of 21.03.2011. We reserve the right to make any changes required to technically improve the equipment.

This manual has been produced with all due diligence.
CAETEC shall not be held liable for any damage resulting from the use of this manual, providing it is not due to gross negligence on our own part or the part of our legal representative or vicarious agent, and to the extent that the damage does not stem from personal injury, bodily harm or damage to health.

Contents

1 Introduction

This manual explains the ARCOS and µCROS logger systems. It provides technical descriptions of the components and their technical data, along with a description of the configuration and the resulting application options.

The functionality of the systems depends on the particular firmware version in use. Therefore, all functions and parameters that are only available with a specific firmware version (or higher) have been marked with the respective version number in superscript (e.g., with $^{1.3}$). Which particular firmware version is installed can be determined by checking the display, the Web interface or the log file.

In addition, you will find brief instructions for a variety of useful procedures and methods for working with the logger. This information is provided with no claim to completeness, and revisions shall be made as development progresses.

Disclaimer Using the logger and display while operating a vehicle can distract the driver. Concentration on traffic should be the driver's top priority at all times.

Disclaimer Operating a data logger in transmission mode can interfere with the proper functioning of safety and control components, resulting in the cancellation of the respective vehicle operating permit. It is the sole responsibility of the installing party to supervise this. Moreover, it is that same party's obligation to instruct vehicle drivers on this matter, as well as to advise them on proper procedures.

2 Hardware

2.1 General Information

The "ARCOS" hardware system comprises a number of modules connected to a basic platform unit. The functioning of the particular system (ARCOS dL, ARCOS PC or ARCOS X-Over) varies according to the types of components used.

ed

Illus. 2.1-2 The complete logger unit, comprising the basic platform and the chassis units (without slide-in modules)

The basic platform consists of a housing containing the central processor assembly. The particular front employed determines the functionality of the system. The number of modules in the assembly is variable and depends on the interfaces required for a particular application.

A basic distinction is made between those modules that are designed to be mounted on top of the basic platform unit and those for assembly underneath. These two types (upper/lower units) are not interchangeable, if only due to their different connector channels (the so-called "backplane"). Separately, however, the chassis units for the lower (or upper) modules are interchangeable, irrespective of their order. There is no set limit to the number of chassis units assembled. The number is limited by the maximum size of the complete housing (installation space), the maximum permitted current draw, and similar peripheral parameters.

All modules are centrally connected to the supply voltage via the jack in the basic platform. Each chassis unit has its own power supply conditioning (DC/DC conversion), its own "power supply", so to speak.

The connecting sockets in the top or bottom chassis units can be covered with a protective plate. It is screwed onto the housing and is fitted with a rubber seal.

A display is auxiliary to the ARCOS dL and µCROS. Basically, the logger can be used without a display, as well, with no impairment of measurement. The significant advantage of the display is that it serves as the user interface for transferring the measured data at the end of a measurement (cf. Chapter "Display") or for documenting the driven track (track selection).

2.2 ARCOS Basic Platform

The basic platform unit comprises an x86 processor platform, to which all the information and signals are fed. The chassis units are connected to it via bus systems. All the connections between the basic platform unit and the external periphery (power supply, display,…) plug into the front plate.

The operating system with the data acquisition software, as well as the measured data, is stored on a compact flash memory card (CF card). Another option for data storage is a separate S-ATA hard drive (rotating or solid state). The removable front plate provides access to the storage media.

A so-called "heat spreader" connects the processor system to its housing. The heat produced by the processor is conveyed through the housing to the cooling fins. When required, the rear fans increase convection in the fins, but do not ventilate within the housing.

In the ARCOS dL, the USB connector is for transferring data (importing configurations or transferring measured data) and operating an external configuration provided. This port, therefore, is only used for inserting the same type of USB memory sticks or external USB hard drives as are used in PC technology (max. 500 mA).
Expanding the periphery by using USB-CAN interfaces or comparable components via the USB connection is only supported for ARCOs as PC.

Along with the data lines, the ARCOS dL and X-Over display connection socket includes the power supply lines, i.e. the display receives its operating power from the basic unit. The connector is configured as an Ethernet port, connecting the basic platform unit via adapter cable with the Ethernet jack for diagnostic/webinterface with a PC.
The ARCOS PC and X-Over are equipped with a standard VGA connector, while the ARCOS PC also has a standard LAN connector.

The power supply connector contains not only the pins for positive and ground (or, in automotive engineering terms, the "Clamps 30 and 31"), but also a pin for switch input (what automotive engineers often call a "Clamp 15"). A voltage level above approx. 3V on this pin starts the logger. This start capability exists in addition to start via bus messages (e.g., "Wake-on-CAN" = WoC). Regardless of whether WoX is activated, sufficient voltage at the control pin (Cl 15) will start the logger.
This behavior also enables an emergency start-up, ensuring there's always a way to safely start the logger (following a configuration error, e.g., such as Start on a CAN message that never occurs).

To minimize standby current consumption, the Clamp 15 has a very high impedance (> 10 kilohm). In surroundings with high disturbance levels, one should make sure the potential is properly connected to the clamp connector and not leave the pin open – in order to avoid unexpected system start-ups.

The external fuse is a vehicle fuse, for a maximum current of 7.5 A.

As an indication of the total current draw of a typical system, we measured a current draw of: approx. 3A with 12 V,

for the following assembly:

- basic platform, with
- four-channel CAN modules (2 units), plus
- WLAN module

Note

The upper limit for supply voltage is 50 V. Loggers manufactured after February 2008 are equipped with an integrated over-voltage trip in the supply line. This protects the device against voltage > 50 V DC by tripping the fuse in a controlled manner. For this purpose, the supply source must be able to supply current at least twice as high as the nominal strength of the fuse. Otherwise the over-voltage device won't trip safely, in which case, over-voltage may cause serious damage to the logger. (In a vehicle, this condition is generally met.)

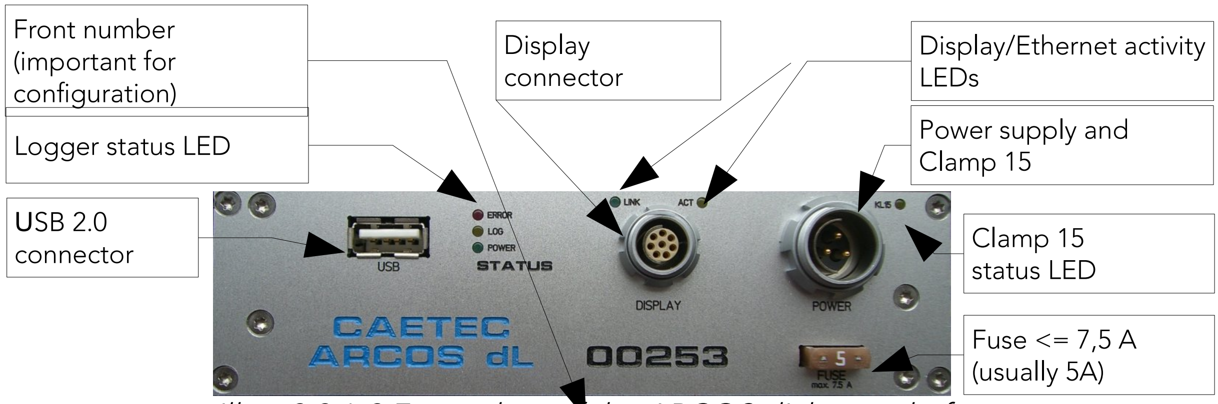
Warning

Loggers manufactured before February 2008 have no such over-voltage trip. Under no circumstances are the supply lines of these devices to be connected to voltages > 50 V DC.

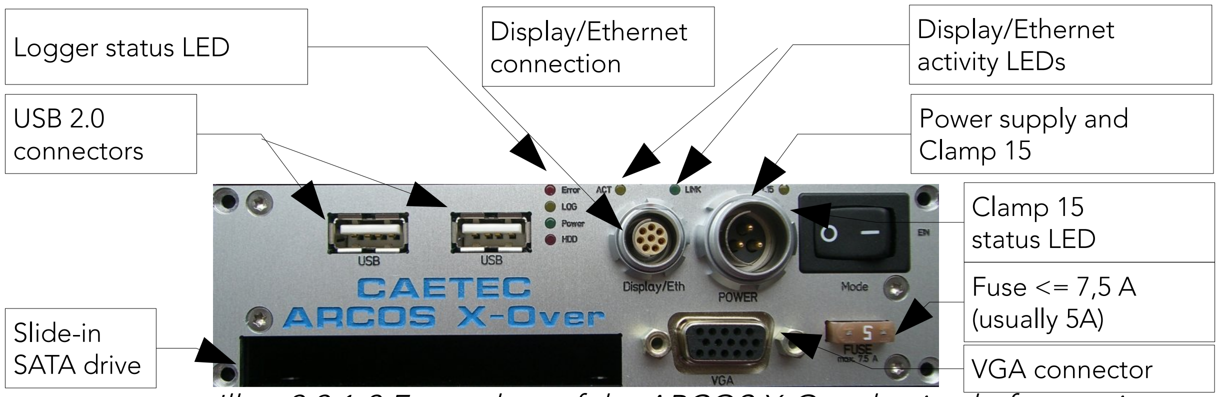
The fuse may only be changed when the logger is disconnected from the power supply. (Pull the plug in the front plate.)

2.2.1 About the Connections

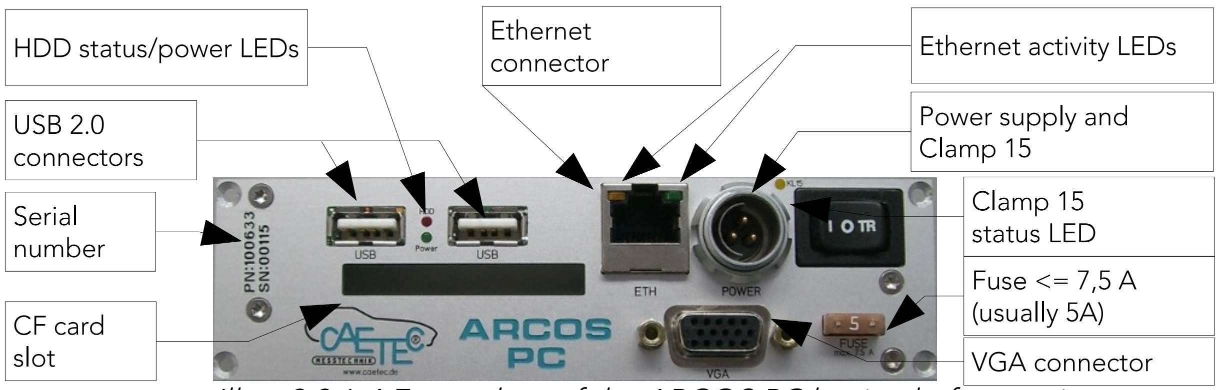
Illus. 2.2.1-1 Front view of the basic platform unit.



Illus. 2.2.1-2 Front plate of the ARCOS dL basic platform unit.



Illus. 2.2.1-3 Front plate of the ARCOS X-Over basic platform unit



Illus. 2.2.1-4 Front plate of the ARCOS PC basic platform unit.

2.2.2 Connection and LED Assignment

USB	
Standard assignment of PC technology.	Transfers configurations and measurement data (ARCOS dL), and (in the ARCOS PC) is available for other USB hardware.

Logger Operating Status (ARCOS dL and X-Over)	
Error (red): Off	No error.
Briefly flashes	Firmware malfunction has occurred, fallback will start shortly.
Blinks (lit/dark:50/50%)	Firmware running with fallback configuration.
Lit (steady)	Firmware malfunction.
Log (yellow): Off	No measurement.
Blinks[1,3] (lit/dark:50/50%)	Measurement running, but there is no method storing data.
On	Measurement running, data are being stored.
Power (green): Off	Measurement system is off – no power supply connected.
Short blinks (lit/dark:25/75%)	Measurement system is off – power supply is connected (approx. 900µA current draw).
Blinks (lit/dark:50/50%)	System booting, due to met start condition (e.g. Clamp 15 or WoX).
Lit (steady)	Boot was successful, logger is operational. (Check the Log LED for the measurement mode.)
Long blinks (lit/dark:75/25%)	System shutting down (takes approx. 10 s), ends with the mode "short blinks".

Display Pin Assignment (ARCOS dL and X-Over)	
1	Supply -
2	Supply -
3	Supply +
4	Supply +
5	Data 1 (RX+, orange and white).
6	Data 2 (RX–, orange).
7	Data 3 (TX+, green and white).
8	Data 4 (TX–, green).

Display Activity LEDs (ARCOS dL and X-Over)	
Link	Lit when the cables are connected and a link has been established.
Act	Blinks when data packets are being transported.

Power Pin Assignment	
1	Supply +
2	Ground
3	Clamp 15

Clamp 15 Status Display	
LED on/off.	Voltage level for logger start is present/not present (voltage above 3V).

VGA (ARCOS X-Over and PC)	
Standard assignment of PC technology.	Connection to a standard VGA monitor.

LAN (RJ45) (ARCOS PC)	
Standard assignment of PC technology.	LAN connector.

2.2.3 Technical Data

- Technology: Industrial x86 platform, with a wide selection of available processors (currently 1.66 GHz CoreDuo)
- Operating system: ARCOS dL: GNU/Linux
  ARCOS PC: e.g., Windows XP (customized)
- Working memory: 2 GB (DDR-RAM)
- Hard drive: 1 x CF card with 4 to 32 GB of memory (of which 1 GB is reserved for the operating system)
  Optional: 1 x SATA 2.5" SSD with 16 to 512 GB of memory.
- Interfaces: 1-3 x USB 2.0 host
  1 x display (incl. power supply) (only ARCOS dL)
  1 x upper module interface
  1 x lower module interface
  1 x VGA (ARCOS PC/X-Over only)
  1 x Ethernet (ARCOS PC/X-Over only)
- Connector: Display: Fischer DGP 103 A058-130
  female connector: Fischer S 103 A058-130+
  Power: Fischer DGP 104 Z040-80
  female connector: Fischer S 104 Z040-80+
- Input voltage: 6 - 50 V DC
- Power consumption: approx. 12 W
- Operating temperature: -40 °C to +85 °C
- Dimensions (W x H x D): 184 mm x 42 mm x 144 mm
- Weight: Approx. 1336 g

2.2.4 Standard Configuration

- dL/Logger: Recommended for time data/classing results from CAN and LIN.
  Intel CoreDuo 1.66 Ghz
  2 GB RAM
  4 GB CF card

- dL/Logger: Recommended for Full Traffic Traces and FlexRay
  Intel CoreDuo 1.66 Ghz
  2 GB RAM
  16 GB CF card
- PC: Recommended for Windows XP
  Intel CoreDuo 1.66 Ghz
  2 GB RAM
  16 GB CF card

2.3 µCROS

Unlike the ARCOS, the µCROS has a closed housing to which no expansion modules can be connected. Due to its hardware architecture, it runs the same µCROS software as an ARCOS dL; the configurations are also totally interchangeable – except that the µCROS has a lower computing power and supports fewer interfaces and bus interfaces.

In the µCROS LE version, the CAN channels are purely passive, no CAN Acknowledge is set and sending is not supported (which means there's no CCP/XCP!). However, an upgrade via software license is subsequently possible.

The µCROS comprises an x86 processor platform connected via internal Ethernet connector to the integrated two or four-channel CAN interface. The platform also provides two slots for any combination of GPS, WLAN or UMTS modules. Each module has an SMA jack on the front plate as antenna connector. All the µCROS connectors to external periphery (power supply, display, CAN, USB, the antenna connections of the custom options) are on the front plate.

The operating system with the data acquisition software and the measurement data are stored on a separate mSATA storage medium installed in the µCROS.

A so-called "heat spreader" connects the processor system to its housing. The heat produced by the processor is conveyed through the housing to the cooling fins. Since it produces less heat than the ARCOS, the housing is totally without fans. So the µCROS contains no moving parts.

The µCROS USB connector is for transferring data (importing configurations or transferring measured data) and operating an external configuration provided. This means this port is only used for inserting the same type of USB memory sticks or external USB hard drives (max. 500 mA) as are used in PC technology.

No provision has been made for expanding the periphery by means of USB-CAN interfaces or comparable components.

The µCROS display jack includes the data lines, as well as the power supply lines, i.e. the display receives its operating power from the logger.
The µCROS connector is configured as an Ethernet port. This port is connected via adapter cable with the Ethernet jack for diagnostic/Webinterface with an PC.

The power supply jack contains not only the pins for positive and ground, but also a pin for switch input (a so-called "Clamp 15"). At this pin, a voltage level above approx. 3 V starts the logger. This start capability is provided in addition to start via bus messages ("Wake-on-CAN" = WoC). Regardless of whether WoX is activated, sufficient voltage at the control pin (Cl 15) will start the logger.

To minimize standby current consumption, the Clamp 15 has a very high impedance (> 10 kilohm). In surroundings with high disturbance levels, one should make sure the potential is properly connected to the clamp connector and not leave the pin open – in order to avoid unexpected system start-ups.

The fuse of the complete system is an SMD fuse for a maximum current of 3 A. The display power supply has been equipped with a self-healing 500 mA fuse.

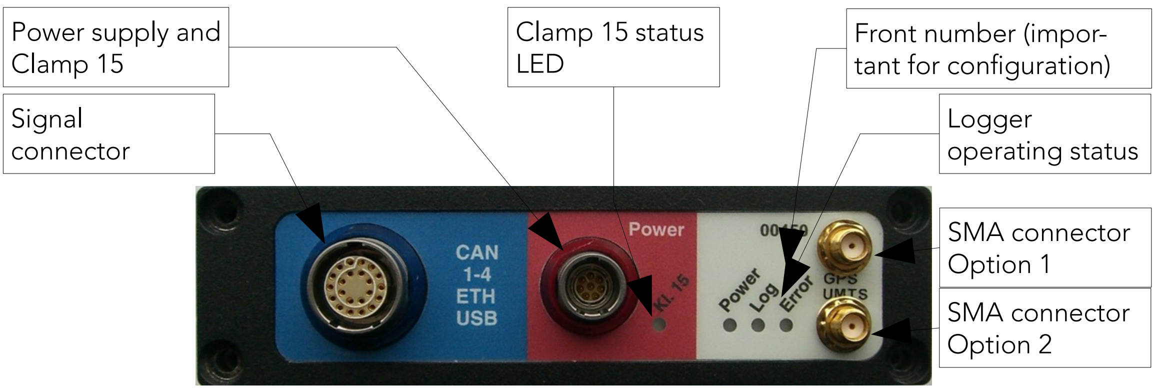
With the µCROS, typical total current draw is virtually independent of the options selected (WLAN, UMTS, GPS).

Current draw: approx. 0.5 A with 12 V.

Note

The upper limit for the supply voltage is 36 V, without an integrated over-voltage trip in the supply line. The fuse for the total system can only be changed by CAETEC.

2.3.1 About the Connections



Illus. 2.3.1-1 Front plate of the µCROS.

2.3.2 Connection and LED Assignment

Power Pin Assignment	
1	Clamp 15
2	Supply +
3	Supply +
4	Supply +
5	Ground
6	Ground
7	Ground
8	n.c.

Pin Assignment of Signal Connectors	
1	CAN 1 High
2	CAN 1 Low
3	CAN 2 High
4	CAN 2 Low
5	CAN 3 High
6	CAN 3 Low
7	CAN 4 High
8	CAN 4 Low
9	Display Vcc
10	Display GND
11	n.c.
12	n.c.
13	n.c.
14	n.c.
15	Ethernet RX+
16	Ethernet RX–
17	Ethernet TX+
18	Ethernet TX–
19	USB Vcc
20	USB D–
21	USB D+
22	USB GND

Logger Operating Status	
Error (red): Off	No error.
Briefly flashes	Firmware malfunction has occurred, fallback will start shortly.
Blinks (lit/dark:50/50%)	Firmware running with fallback configuration.
Lit (steady)	Firmware malfunction
Log (yellow): Off	No measurement.
Blinks[1.3] (lit/dark:50/50%)	Measurement running, but no method is storing data.
On	Measurement running, data are being stored.
Power (green) Off	Measurement system is off – no power supply connected.
Short blinks lit/dark:25/75%)	Measurement system is off – power supply is connected (approx. 900µA current draw).
Blinks. (lit/dark:50/50%)	System booting, due to met start condition (e.g., Clamp 15 or WoC).
Lit (steady)	Boot was successful, logger is operational. (Check the Log LED for the measurement mode.)
Long blinks (lit/dark:75/25%)	System shutting down (takes approx. 10 s), ends with the mode "short blinks".

CL15 Status Display	
LED on/off.	Voltage for logger start is present/not present (voltage above 3V).

The signal breakout box cable, which is available as an accessory and is connected to the signal connection socket, provides the following connections:

USB	
Standard assignment of PC technology.	Transfer of configurations and measured data.

Display/Ethernet Pin Assignment (identical to ARCOS dL and X-Over)	
1	Supply -
2	Supply -
3	Supply +
4	Supply +
5	Data 1 (RX+, orange and white)
6	Data 2 (RX–, orange)
7	Data 3 (TX+, green and white)
8	Data 4 (TX–, green)

CAN Pin Assignment	
1	CAN 1 Low
2	CAN 2 Low
3	CAN 3 Low
4	CAN 4 Low
5	n.c.
6	CAN 1 High
7	CAN 2 High
8	CAN 3 High
9	CAN 4 High

A CAN bus with "standard" assignment (D-Sub socket with CAN Low Pin 2, CAN High Pin 7 and all other pins unassigned!) can also be directly connected to this plug, without an adapter/breakout box and received on the second CAN channel of the µCROS.

2.3.3 Technical Data

- Technology: Industrial x86 platform, with Intel Atom 1.1 GHz
- Operating system: GNU/Linux
- Working memory: 1 GB (DDR-RAM)
- Hard drive: 1 x mSATA storage with 4 to 32 GB of memory.
  (approx. 1GB are reserved for the operating system)
- Interfaces: 1 x USB 2.0 host
  1 x display/Ethernet (incl. display power supply)
  4 x CAN HighSpeed, without terminating resistor
- Connector: Power: Lemo HES.1F.308.XLDP
  female connector: Lemo FGS.1F.308.YLM
  Signal connection: Lemo HEN.3F.322.XLNP
  female connector: Lemo FGN.3F.322.XLC

  Signal breakout box cable (accessory):
  Display: Fischer K 103 A058-130+
  female connector: Fischer S 103 A058-130+
  USB: Standard USB-A socket
  CAN: 9-pin D-Sub male
- Input voltage: 6 - 36 V DC
- Power consumption: approx. 6 W
- Current draw: in standby 0.2 mA (12 V)
  in standby, with WoC 2 mA (12 V)
  in standby, with WoCnml 50 mA/channel (12 V)
- Operating temperature: -20 °C to +75 °C
- Dimensions (W x H x D): 132 mm x 36 mm x 108 mm

- Weight: approx. 680 g

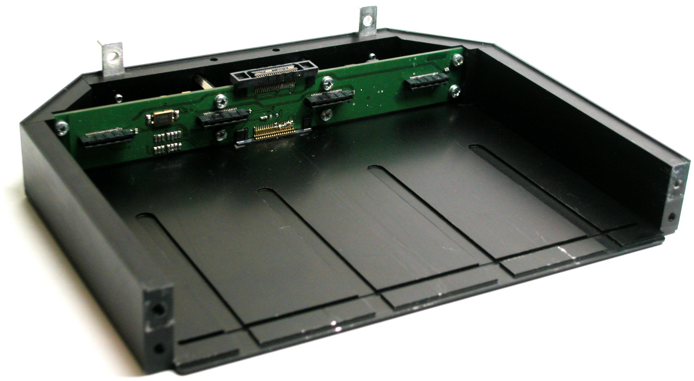
2.3.4 Standard Configuration

- µCROS: Intel ATOM 1.1 Ghz
  1 GB RAM
  4 GB mSTA storage (internal)
  Four CAN channels
  Options GPS and UMTS

2.4 Upper Expansion Chassis Unit (USB)

This expansion chassis unit is mounted on top of the basic platform unit and houses slide-in modules with USB interfaces.

To avoid accidental use as a "lower chassis unit", its connector has been keyed, with differently positioned alignment pins. In addition, the grooves cut in the lid of the chassis unit (cf. 2.4-1), which serve as assembly guides for the modules, have a different design. The lid serves as the bottom for the next chassis unit. This means that modules for the "upper chassis unit" cannot be inserted in the "lower chassis unit", and vice versa.



Illus. 2.4-1 Lower view of the upper expansion chassis unit

The chassis units have been designed to house up to four slide-in modules per unit. The available space can be divided up between 1/4, 2/4 and 4/4 slide-in modules. The form of the slide-in modules depends on the interface type. The modules currently available for the upper chassis unit are:

- WLAN
- GPS
- UMTS/GPRS
- Power switch
- USB/Powerswitch
- USB hub

- ETH 1G
- USB HDD
- Video
- CANCaseXL (ARCOS PC and ARCOS X-Over only)
- VN1630 (ARCOS PC and ARCOS X-Over only)

A so-called "backplane" connects the interfaces to the basic unit. The backplane loops both the data signals and the supply through the chassis unit, which makes the chassis units cascadable. Each chassis unit has its own power supply conditioning, its own power supply unit (DC/DC conversion), so to speak. The slide-in modules are identified by their individual IDs, so they function independently of their slot position or the assembly order of the chassis units.

The chassis unit stacks are screwed to the basic platform unit by means of four connecting tabs (two at the front and two at the rear).

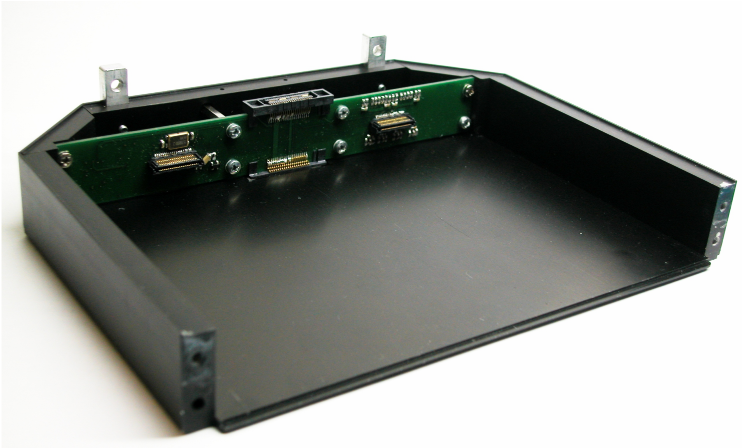
In the final expansion chassis unit, the opening above the contacts can be covered with a screw-down lid.

2.4.1 Technical Data

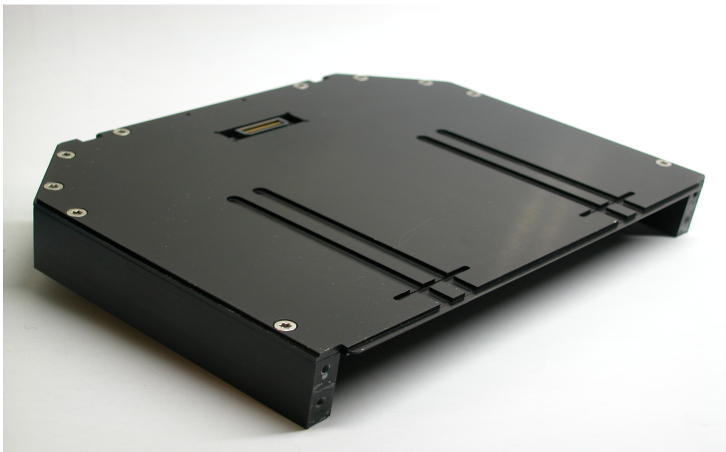
- Dimensions (W x H x D): 184 mm x 27 mm x 144 mm
- Weight: Approx. 430 g

2.5 Lower Expansion Chassis Unit (ETH)

The lower expansion chassis unit is mounted under the basic platform unit and houses slide-in modules with Ethernet interfaces (cf. 2.5-1).
To avoid accidentally substituting it for an "upper chassis unit", its connector has been keyed, with differently positioned alignment pins. In addition, the grooves cut in the lid of the chassis unit (i.e. on the bottom of the adjoining unit, cf. 2.5-2), which serve as assembly guides for the interfaces, have a different design. The bottom serves as the lid for the next chassis unit. This means that modules for the "lower chassis unit" cannot be used in the "upper chassis unit", and vice versa.



Illus. 2.5-1 Upper view of a lower expansion chassis unit.



Illus. 2.5-2 Lower view of a lower expansion chassis unit.

The chassis unit has been designed to house up to two modules per unit. The available space can be divided up between 1/2 and 2/2 slide-in modules. The currently available interfaces are CAN, LIN, FlexRay, MOST and Analog/Digital I/O.

A modified Ethernet interface in the so-called "backplane" connects the interfaces to the basic unit. The backplane loops both the interface and the supply through the chassis unit, which makes the chassis units cascadable. Each chassis unit has its own power supply conditioning, its own power supply unit (DC/DC conversion), so to speak.
In the final expansion chassis unit, the opening above the contacts can be covered with a screw-down lid.

The slide-in modules are each identified by an individual ID (product number/serial number), so they function independently of their slot position or the assembly order of the chassis units. When setting up the system, one should thus bear in mind that the node number for the bus channel is linked to the interface module and not the slot position. The assignment of each interface module to a node number is stored in the logger and configurable via the webinterface.

Important

Any time an interface is replaced, please check the configuration of its node assignment (see 2.8.2 Initializing the Modules)!

The total number of chassis units that can be mounted under the basic platform is limited only by such peripheral parameters as the maximum current of the complete assembly, the overall data throughput or the assembly size.

2.5.1 Technical Data

- Dimensions (W x H x D): 184 mm x 27 mm x 144 mm
- Weight: Approx. 430 g

2.6 Interface Module for the Lower Expansion Chassis Unit

The lower expansion chassis unit houses the modules with field bus channels. Currently available are slide-in modules for LIN, CAN, FlexRay, MOST and Analog/Digital I/O.
Except for Analog/Digital I/O, all module types have quick-start functionality, i.e. signals on one or more channels start and stop the logger.

2.6.1 CAN

The models currently available are galvanically separate CAN slide-in modules – with two CAN channels (half-width), and four CAN channels (full-width) – and a not galvanically separate CAN slide-in with four CAN channels (half-width). This means that per expansion chassis unit up to eight CAN channels are available. These modules are equipped with Wake-on-CAN (WoC) and quick-start functionality.

In this context, Wake-on-CAN means that any CAN signal on one or more of the CAN channels can start the logger. You can even specify for the content of a specific message bit to be used for start-up. For details about starting and stopping with WoC, see Error: Reference source not found, Error: Reference source not found and the following chapter.

Idle current draw needs to be minimized, but message loss should also be avoided. To optimize idle current draw with respect to these competing demands, two modes were introduced:

- WoC: very little idle current, but the first messages are lost (max. 30 ms).
- WoC with "No message lost": higher idle current, but no messages are lost.

2.6.1.1 Input Buffer

Logger booting time is currently approx. 20 seconds. During this time, any data in the CAN interface is buffered. Once booting procedure is completed, it is transferred to the logger.

The following table shows the buffer size, with the maximum number of buffered messages – for the complete slide-in module, regardless of whether it is a two-channel or four-channel model. A CAN message requires exactly 20 Bytes.

Buffer sizes		
CAN 2/4 Version 1	16 MB	838,860 messages
CAN 4 Version 2	16 MB	838,860 messages

Table 2.6.1-1 CAN interface buffer sizes

The buffer time for a particular buffer size depends on the bus load and CAN bit rate. It is easy to compute using the above information (a 500 kbit CAN bus handles about 3600 messages per second).

If, for example, we have

- *50% bus load, 500 kbit rate => 1800 messages/s and*
- *4 CAN channels,*

we can calculate the time t until the buffer is full:

$$t = \frac{838860\ messages}{4 * 1800\ messages/s} = 116.5\ s$$

(Note: In a two-channel expansion module, the buffer is distributed on only two nodes, so the buffer available per node is twice as large and, as a result, the maximum buffer time is twice as long.)

The current implementation uses FIFO memory. In the event of buffer overflow, you lose any data arriving while the memory is full.

Overflow can occur if booting time is excessively long. This can occur if, at an unfavorable moment, the power supply drops, so the logger shuts off without shutdown. In certain cases, the operating system will then perform a disk consistency check, which can take a minute or longer.
The logger will always boot, as the operating system is robust against sudden voltage drops. You must take this possible circumstance into consideration, though, when calculating the length of the booting phase.

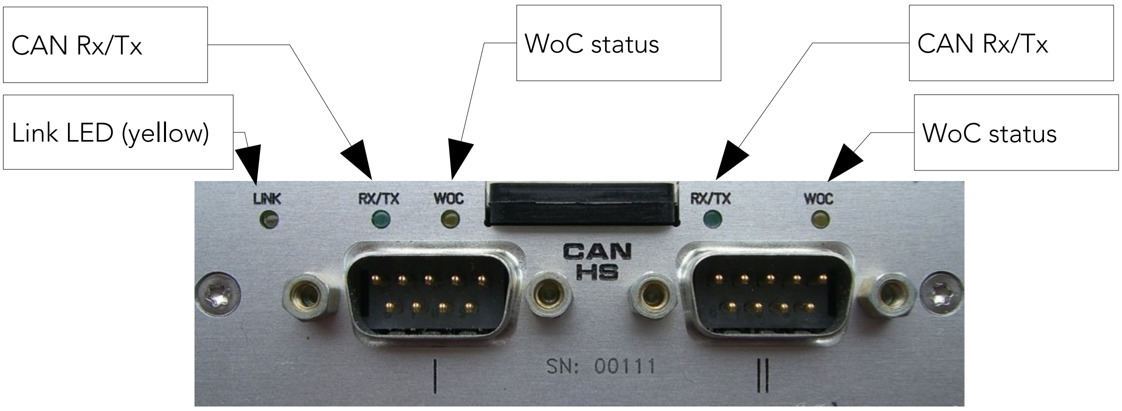
2.6.1.2 Switching Output for External Hardware (not CAN interface v2)

On the one hand, an external voltage level can start the logger, as well as any external hardware such as CAN measurement modules (thermo, analog, ...). Conversely, CAN signals can start the logger (WoC). In that case, the logger must produce a control signal to initiate the power-up process in the external measurement hardware.

The switching voltage is provided by the logger to the nine-way D-Sub sockets of the CAN channels.

The signal consists of a galvanically separate 5 V TTL voltage on pin 1 of the 9-pin D-Sub. The corresponding ground (galvanically separate from the system power supply) is on pin 6.

2.6.1.3 Assigning the Connections



Illus. 2.6.1-1 CAN interface – front plate of two-channel module (four-channel module is comparable).

CAN Interface LEDs	Comment
Link LED (yellow)	Data transfer from the interface to the logger
CAN Rx/Tx (green)	Data transfer on the CAN channel
WoC (yellow)	"Wake on CAN" status (for more on blink codes, see Table 2.6.1-3)

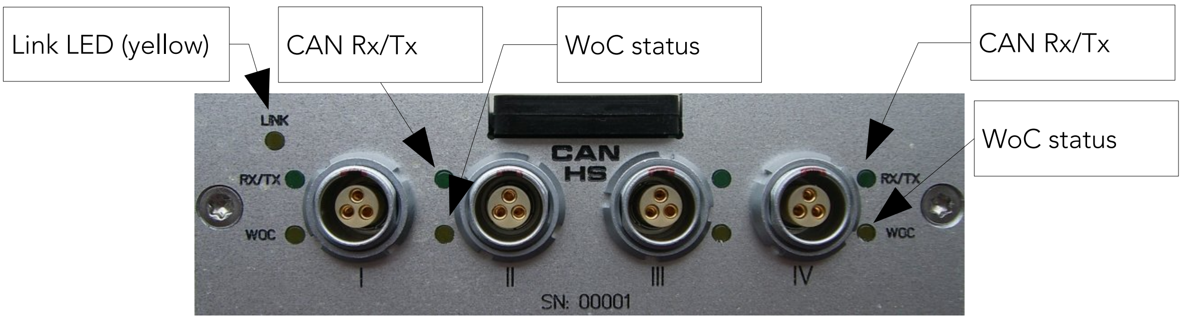
Table 2.6.1-2 CAN channel LEDs

WoC Status LED (blink codes)	Comment
Off	WoC not activated.
Short, double	WoC activated and waiting for CAN signal (to wake the logger).
Lit (steady) (basic platform Log LED is off)	WoC with "No Message Loss" activated and waiting for CAN signal (to wake the logger).
Fast blinks	This CAN channel has woken the logger (logger is starting).
Lit (steady)	CAN interface is in operation and measuring.

Table 2.6.1-3 WoC LED Blink codes for the CAN channel.

Pin	CAN D-Sub Assignment
1	TTL signal for the switch module (galvanically isolated from supply).
2	CAN Low (galvanically isolated)
3	GND
6	GND for TTL switch voltage (galvanically isolated from power supply).
7	CAN High (galvanically decoupled)

Table 2.6.1-4 CAN channel D-Sub plug assignment.



Illus. 2.6.1-2 CAN interface - four-channel module front plate version 2

Pin	CAN LEMO Assignment
1	CAN High (not galvanically decoupled!)
2	CAN Low (not galvanically decoupled!)
3	n.c.

Table 2.6.1-5 CAN channel LEMO jack assignment.

The functions of the LEDs and blink codes are identical to those of the D-Sub version (see Table 2.6.1-2 and Table 2.6.1-3.

2.6.1.4 Technical Data

Four-channel module:

- Dimensions (W x H x D): 168 mm x 25 mm x 119 mm
- Weight: approx. 260 g
- Input voltage: 6 - 50 V DC
- Operating temperature: -40 °C to +85 °C
- Connectors: 4 x 9-pin D-Sub male
- Number of channels: four channels
- Current draw: in standby, 0.145 mA (12 V)
  in standby with WoC, 19.5 mA (12 V)
  in standby with WoCnml, 106.5 mA (12 V)
  in operation, 150 mA (12 V)

Two-channel module:

- Dimensions (W x H x D): 84 mm x 25 mm x 119 mm

- Weight: approx. 146 g
- Input voltage: 6 - 50 V DC
- Operating temperature: -40 °C to +85 °C
- Connectors: 2 x 9-pin D-Sub male
- Number of channels: two channels
- Current draw: in standby, 0.155 mA (12 V)
  in standby with WoC, 10.5 mA (12 V)
  in standby with WoCnml, 70.5 mA (12 V)
  in operation, 110 mA (12 V)

Four-channel module version 2[1.3]:

- Dimensions (W x H x D): 84 mm x 25 mm x 119 mm
- Weight: approx. 150 g
- Input voltage: 6 - 50 V DC
- Operating temperature: -40 °C to +85 °C
- Connector: LEMO EGG.0B.303.CLL
  mating plug: LEMO FGG.0B.303.CLAD…
- Number of channels: four channels
- Current draw: in standby, 0.130 mA (12 V)
  in standby with WoC, 3.1 mA (12 V)
  in standby with WoCnml, 39.6 mA (12 V)
  in operation, 80 mA (12 V)

2.6.2 LIN

The models currently available are a galvanically separate LIN slide-in module with two LIN channels (half-width), and a not galvanically separate LIN slide-in with four LIN channels (half-width).
This means that per expansion chassis unit up to eight LIN channels are available. These modules are equipped with Wake on LIN (WoL) functionality.

In this context, Wake on LIN means that any LIN signal on one or more of the LIN channels can start the logger. You can even specify that the content of a specific message bit be used to trigger start. (For details about starting and stopping with WoL, see Error: Reference source not found, Error: Reference source not found and the following chapter.)

The LIN modules operate in the logger only in “slave mode”.

2.6.2.1 Input Buffer

Logger booting time is currently approx. 20 seconds. During this time, any data in the LIN interface is buffered. Once booting procedure is completed, the data is transferred to the logger.
Buffer size is 16 MB per slide-in module.

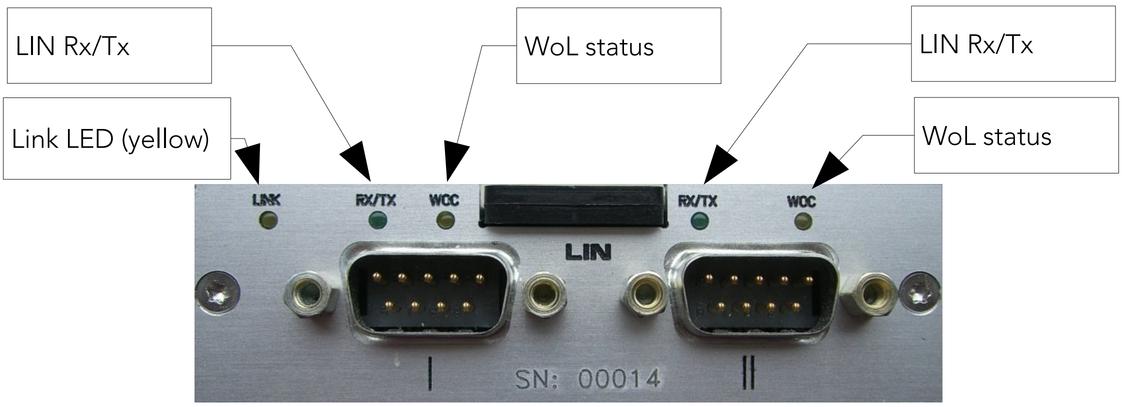
2.6.2.2 Power Supply to the LIN Channel

Supply voltage is typically provided through the LIN wiring set. If this voltage is missing or is lower than 10V, there is an auxiliary power supply to the LIN interface.
In other words, a LIN connector supplying only signal GND and LIN data can operate at the interface.

2.6.2.3 Setting the Baud Rate

The LIN interface has automatic baud rate recognition. This means the baud rate need not be set in the configuration.

2.6.2.4 Connection Assignment



Illus. 2.6.2-1 LIN interface - two-channel module front plate

LIN Interface LED		Comment
Link LED	(yellow)	Data transfer from the interface to the logger
LIN Rx/Tx	(green)	Data transfer on the LIN channel
WoL	(yellow)	“Wake on LIN” status (blink codes, see Table 2.6.2-2)

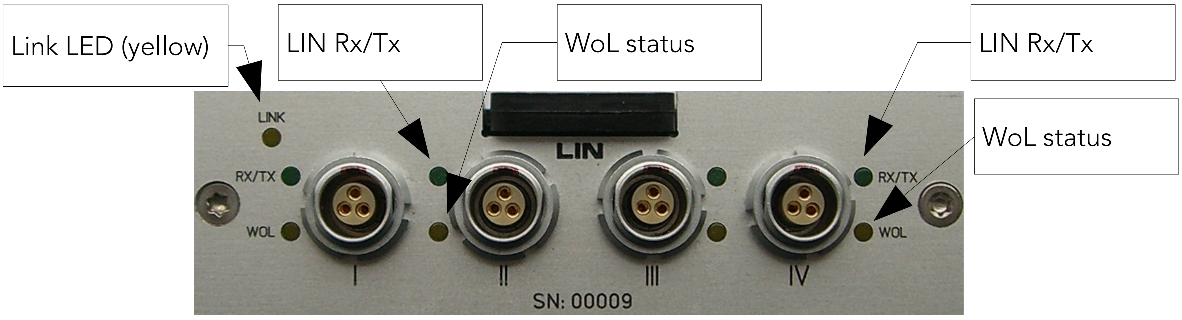
Table 2.6.2-1 LIN channel LED functions.

WoL Status LED (blink codes)	Comment
Off	No WoL activated
Short, double	WoL activated and waiting for LIN signal (to wake the logger)
Fast blinks	This LIN channel has woken the logger (logger is starting)
Lit (steady)	LIN interface is in operation and measuring

Table 2.6.2-2 LIN module WoL LED blink codes.

Pin	LIN D-Sub Assignment
1	TTL switch voltage for switch module (galvanically isolated from the power supply).
3	GND
6	GND for TTL switch voltage (galvanically isolated from power supply).
7	DATA-LIN (galvanically decoupled)
9	VCC-LIN

Table 2.6.2-3 LIN module D-Sub plug assignment.



Illus. 2.6.2-2 LIN interface - four-channel module front plate version 2

Pin	LIN LEMO Assignment
1	DATA-LIN (not galvanically decoupled!)
2	n.c.
3	n.c.

Table 2.6.2-4 LEMO jack assignment for the LIN channel

The functions of the LEDs and blink codes are identical to those of the D-Sub version (see Table 2.6.2-1 and Table 2.6.2-2).

2.6.2.5 Technical Data

Two-channel module:

- Dimensions (W x H x D): 84 mm x 25 mm x 119 mm
- Weight: approx. 145 g
- Input voltage: 6 - 50 V DC
- Operating temperature: -40 °C to +85 °C
- Connectors: 2 x 9-pin D-Sub male

- Number of channels: two channels
- Current draw: in standby, 0.16 mA (12 V)
  in operation, 120 mA (12 V)

Four-channel module version 2[1.3]:

- Dimensions (W x H x D): 84 mm x 25 mm x 119 mm
- Weight: approx. 156 g
- Input voltage: 6 - 50 V DC
- Operating temperature: -40 °C to +85 °C
- Connectors: LEMO EGG.0B.303.CLL
  mating plug: LEMO FGG.0B.303.CLAD…
- Number of channels: four channels
- Current draw: in standby, 0.20 mA (12 V)
  in standby with 4 x WoL, 50 mA (12 V)
  in operation, 115 mA (12 V)

2.6.3 FlexRay [1.3]

Currently available are half-width FlexRay slide-in modules with two FlexRay channels, each configurable as an A or B channel.
This means that per expansion chassis unit up to four FlexRay channels are available. These modules are equipped with Wake-on-FlexRay functionality (WoFR).

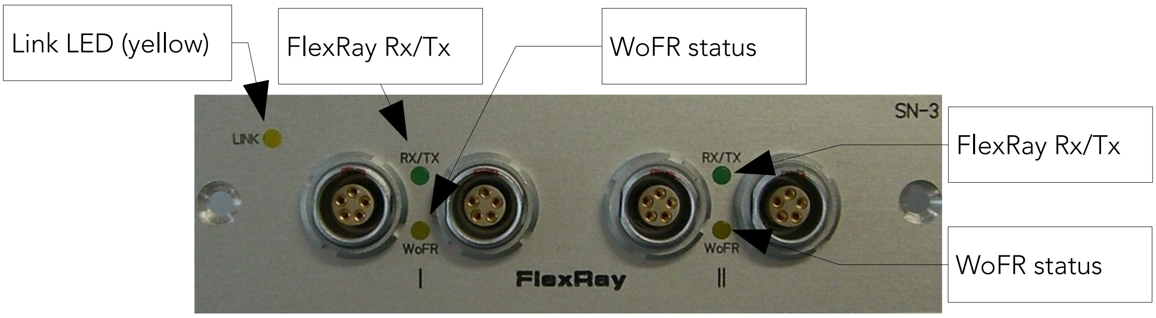
2.6.3.1 Input Buffer

Logger booting time is currently approx. 20 seconds. During this time, any data in the FlexRay interface is buffered. Once booting procedure is completed, the data is transferred to the logger.
Buffer size per slide-in module is 128 MB.

2.6.3.2 Timing Parameters

Due to the method used for signal acquisition on the bus, the FlexRay interface needs no additional parameters to define the exact timing on the bus. Therefore, they require no specification in the configuration, providing messages or signals need only be received and not sent (see also Chapter Error: Reference source not found Error: Reference source not found).

2.6.3.3 Connection Assignment



Illus. 2.6.3-1 FlexRay interface - two-channel module front plate

LED FlexRay Interface	Comment
Link LED (yellow)	Data transfer from the interface to the logger
FlexRay Rx/Tx (green)	Data transfer on the FlexRay channel
WoFR (yellow)	"Wake on FlexRay" Status (for more on blink codes, see Table 2.6.3-2)

Table 2.6.3-1 FlexRay module LED functions.

WoFR Status LED (blink codes)	Comment
Off	No WoFR activated
Short, double	WoFR activated and waiting for FlexRay signal (to wake the logger)
Fast blinks	This FlexRay channel has woken the logger (logger is starting)
Lit (steady)	FlexRay interface is in operation and measuring

Table 2.6.3-2 FlexRay module WoFR LED blink codes.

Pin	FlexRay LEMO Assignment
1	Reserve (shield)
2	BP in
3	BM in
4	BP out
5	BM out

Table 2.6.3-3 FlexRay module LEMO jack assignment.

Per FlexRay channel there are two five-pin Lemo sockets, whose PINs are connected straight through. This offers a variety of connection possibilities, from a simple branch line to looping right into the bus.

2.6.3.4 Technical Data

- Size: 84 mm x 25 mm x 119 mm
- Weight: approx. 165 g

- Input voltage: 6 - 50 V DC
- Operating temperature: -40 °C to +85 °C
- Connector: LEMO EGG.0B.305.CLN
  mating plug: LEMO FGG.0B.305.CLAD…
- Number of channels: two (either A or B channel possible)
- Transceivers: 2 x TJA1080
- Controllers: 2 x Fujitsu MB88121B
- Memory: 128 MB
- Current draw: in standby, 0.26 mA (12 V)
  in standby with WoFR, 5 mA (12 V)
  in operation, 340 mA (12 V)

2.6.4 Analog/Digital I/O [1.3]

The currently available unit is a (half-width) Analog/Digital I/O (ADIO) slide-in module with eight analog channels, eight digital bit inputs and eight digital bit outputs.

2.6.4.1 Input Buffer

Logger booting time is currently approx. 20 seconds. During this time, any data in the ADIO interface is buffered. Once booting procedure is completed, the data is transferred to the logger. During booting, digital outputs are not addressed and remain in a defined start state (no bit set). Once the logger firmware is fully started and backlog execution has been started, the digital outputs are also activated.
Buffer size is 8 MB per slide-in module.

2.6.4.2 Sampling Rate

Each channel of the ADIO interface can be independently parameterized with a sampling rate in the range of 0.1…1000.0 Hz. The digital outputs are activated when needed, i.e. upon value change, with a maximum rate of 100 Hz (for more information, see Chapters Error: Reference source not found, Error: Reference source not found and Error: Reference source not found).

2.6.4.3 Connection Assignment

Link LED (yellow)

Illus. 2.6.4-1 ADIO interface - front plate

LED ADIO Interface	Comment
Link LED (yellow)	Data transfer from the interface to the logger.

Table 2.6.4-1 ADIO interface LED functions.

Pin	ADIO LEMO Analog Input Assignment
1	Analog input channel 1 + or 5 +
2	Analog input channel 1 – or 5 –
3	Analog input channel 2 + or 6 +
4	Analog input channel 2 – or 6 –
5	Analog input channel 3 + or 7 +
6	Analog input channel 3 – or 7 –
7	Analog input channel 4 + or 8 +
8	Analog input channel 4 – or 8 –
9	GND
10	GND

Table 2.6.4-2 LEMO jack assignment at the analog input.

For analog input, the ADIO provides two jacks, for the channels 1 - 4 and channels 5 - 8 respectively. Though these are differential inputs, it may be useful to connect the ADIO grounds (GND) with the ground of the voltage source to be measured, so as to create a common reference potential.

Pin	ADIO LEMO Digital Input Assignment
1	Digital input bit 1
2	Digital input bit 2
3	Digital input bit 3
4	Digital input bit 4
5	Digital input bit 5
6	Digital input bit 6
7	Digital input bit 7
8	Digital input bit 8
9	Digital GND
10	Digital GND

Table 2.6.4-3 LEMO jack assignment for digital input.

Always connect the digital inputs to a fixed reference ground. The switching threshold between high (=1) and low (=0) level is approx. 1.5 V DC.

Pin	ADIO LEMO Digital Output Assignment
1	Digital output bit 1
2	Digital output bit 2
3	Digital output bit 3
4	Digital output bit 4
5	Digital output bit 5
6	Digital output bit 6
7	Digital output bit 7
8	Digital output bit 8
9	Digital GND
10	Digital GND

Table 2.6.4-4 LEMO jack assignment for digital output.

The digital outputs function as switches, for connecting the particular connection pin to ground. If a logical 0 is applied, they are open; they are closed if a logical 1 is applied. In combination with the appropriate external switches, it is thus possible to control external devices (e.g., relays, LEDs).

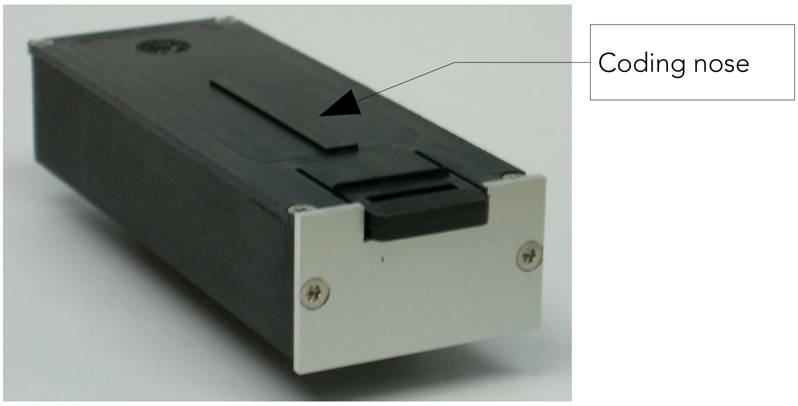
2.6.4.4 Technical Data

- Dimensions (W x H x D): 84 mm x 25 mm x 119 mm
- Weight: approx. 180 g
- Input voltage: 6 - 50 V DC
- Operating temperature: -40 °C to +85 °C
- Connectors: Analog input: LEMO EEB.2B.310.CLN
  mating plug: LEMO FGB.2B.310.CLAD…
  Digital input: LEMO EEG.2B.310.CLN
  mating plug: LEMO FGG.2B.310.CLAD…
  Digital output: LEMO EEA.2B.310.CLN
  mating plug: LEMO FGA.2B.310.CLAD…
- Number of channels: 2 x four analog channels
  8 digital bit inputs
  8 digital bit outputs
- Measuring range: Analog input: ±60 V, 16-bit resolution (1.8 mV)
  Digital input: 0…60 V,
  switching threshold at approx. 1.5 V
  input resistance: 1 MOhm
  Digital output switching voltage: 0…60 V
  Max. switching current: 250 mA
- Memory: 8 MB

- Current draw: in standby, 0.13 mA (12 V)
  in operation, 68 mA (12 V)

2.7 Interface Modules for the Upper Expansion Chassis Unit

All the modules for the upper expansion chassis unit have a coding nose. This ensures that they are operated only in an upper expansion unit. 2.7-1 shows a simple dummy module with the coding nose.



Illus. 2.7-1 USB dummy module for the upper expansion chassis unit.

2.7.1 WLAN

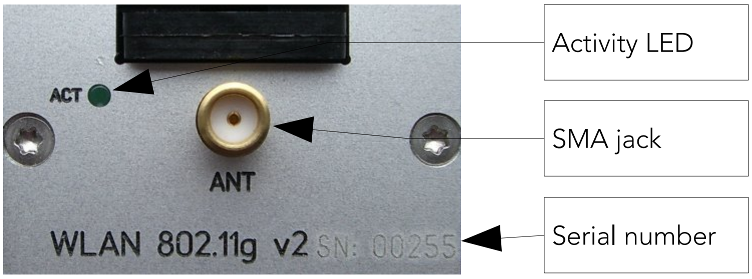
2.7.1.1 Applications

The WLAN module is another channel (like a memory stick, for example) for transmitting recorded data from the logger to an external computer, such as a server station, or for importing new configurations. Data transfer that is started when the logger is shut down, for example, can be directed to the WLAN module.

The module is only activated when data transfer is started, and is inactive during measurement operation. This increases data security – a WLAN that is inactive cannot be misused for unauthorized data transmission. Moreover, this reduces total current draw of the measurement system.

Conversely, it isn't possible to use the logger's WLAN port for continuous measurement-data transfer or for data display during measurement.

2.7.1.2 Module Placement

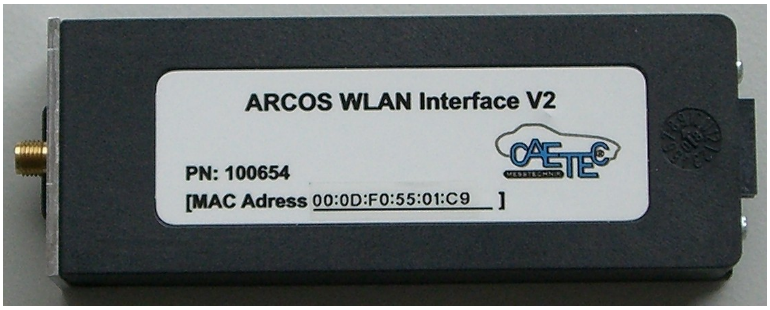


Illus. 2.7.1-1 V2 WLAN module

The module is a quarter-unit wide and can be operated in any slot within a given chassis unit. Neither its position within a chassis unit nor the chassis level it occupies requires specification during configuration.

2.7.1.3 Configuration Parameters

Each module has a fixed MAC address. During module activation, the IP address is either, depending on its configuration, dynamically assigned (DHCP server) or set from the logger.



Illus. 2.7.1-2 Bottom view of WLAN module

The MAC address is very often used by system administrators to protect networks against unauthorized use. When replacing modules, make sure the used module has authorized access to the required networks. For this purpose, the sticker on the bottom of the module has the MAC address printed under the product number (2.7.1-2).

In the case of the µCROS, you can obtain the MAC address from CAETEC (please specify the µCROS serial number). Due to lack of space, there is no sticker, so the address must be read by software from the module.

In the case of built-in modules, you can list MAC addresses according to the respective serial numbers printed on the front of the module.

For details on the application of parameters in configuration, see Chapter Error: Reference source not found.

Note With firmware Version 1.3 and higher, the option of transferring data via WLAN, "DT_WLAN", must be enabled in the logger. Without activation no data transfer is possible.

2.7.1.4 Technical Data

- Dimensions (W x H x D): ARCOS: 42 mm x 25 mm x 114 mm
  µCROS: 25 mm x 8 mm x 80 mm
- Weight: ARCOS: approx. 63 g
  µCROS: approx. 14 g
- Operating temperature: -40 °C to +85 °C
- Input voltage: ARCOS: 6 – 50 V DC
  µCROS: 6 – 36 V DC
- WLAN: standard 802.11g, 2.4 Ghz
  maximum rate: 54 Mbit/s
  security settings: WPA-PSK and WPA-EAP
  Access only through infrastructure (access point).
  No "ad-hoc" connection possible.
  Available for ARCOS and µCROS.
- Current draw: approx. 40 mA

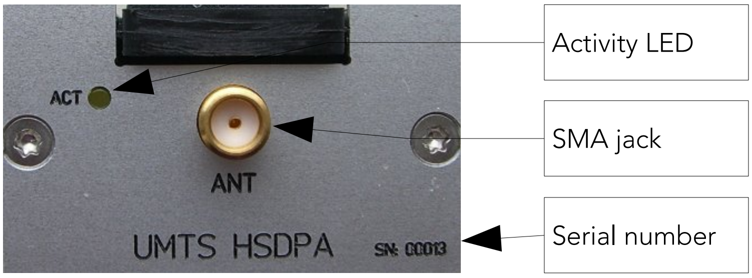
2.7.2 UMTS

2.7.2.1 Applications

The UMTS module is (like a memory stick or WLAN) another channel for transmitting recorded data from the logger to an external computer, such as a server station, or for importing new configurations. Data transfer that is started when the logger is shut down, for example, can be directed to the UMTS module.

The module is activated only once data transfer is started, and is inactive during measurement operation. This increases data security – a mobile telephone connection that is inactive cannot be misused for unauthorized data transmission. Moreover, this reduces total current draw of the measurement system.

2.7.2.2 Module Placement



Illus. 2.7.2-1 UMTS module

The module is a quarter-unit wide, and can be operated in any slot within the given chassis unit. Neither its position within a chassis unit nor the chassis level it occupies requires specification during configuration.

UMTS Interface ACT LED (yellow)	Comment
Fast blinks	No connection to a network carrier.
Slow blinks	Connected to a network carrier.

Table 2.7.2-1 UMTS interface LED functions.

2.7.2.3 Configuration Parameters

Each module has a fixed IMEI number, with which the module accesses the particular mobile telephone network. During module activation, the IP address is dynamically assigned upon registration with the service provider.

Illus. 2.7.2-2 Bottom view of UMTS module.

The IMEI number uniquely identifies a mobile telephone subscriber and can be used by system administrators to protect networks against unauthorized use. When replacing modules, make sure the used module has authorized access to the required networks. For this purpose, the sticker on the bottom of the module has the IMEI number printed under the product number (2.7.2-2).

In the case of the µCROS, you can obtain the IMEI number from CAETEC (please specify the µCROS serial number). Due to lack of space, there is no sticker, so the address must be read from the module by software.

In the case of built-in modules, you can list IMEI numbers according to the respective serial numbers printed on the front of the module.

For details on the application of parameters in configuration, see Chapter Error: Reference source not found.

Note

The option for transferring data via UMTS, "DT_PPP", must be enabled in the logger. Without activation no data transfer is possible.

2.7.2.4 SIM Card

Operation of the UMTS module requires a SIM card.

Note

The SIM card must configured without PIN number request (the logger does not support PIN number input!) and be authorized for data connection.

To insert the SIM card, remove the ARCOS module from the logger (after unplugging the power supply from the logger!) and the lid by removing the four screws. Once you have inserted the SIM card into the card holder (with the card oriented correctly), close the module. Slide it into the chassis unit, plug in an UMTS antenna and the module is ready for operation.

Illus. 2.7.2-3 SIM card holder – in ARCOS and µCROS.

To insert the SIM card in the µCROS, first open the module by removing the four screws on the front. On the bottom you will find the UMTS module with card holder. Once you have inserted the SIM card, carefully slide the µCROS back into the housing. As you slide it back, you may have to push the UMTS module down by pressing on the card holder to avoid damaging it. Plug in a UMTS antenna and the module is ready for operation.

2.7.2.5 Technical Data

- Dimensions (W x H x D): ARCOS: 42 mm x 25 mm x 114 mm
  µCROS: 35 mm x 10 mm x 80 mm
- Weight: ARCOS: approx. 63 g
  µCROS: approx. 31 g
- Operating temperature: -40 °C to +85 °C

- Input voltage: ARCOS: 6 – 50 V DC
  µCROS: 6 – 36 V DC
- UMTS: HSDPA (downlink 7.2 Mbps, uplink 384 kbps)
  UMTS/HSDPA (WCDMA/FDD) 2100 MHz
  Quad-Band EGSM 850/900/1800/1900
  GPRS multi-slot class 12
  EDGE multi-slot class 12
- Antenna: SMA jack in the module, for connecting a standard GSM/UMTS antenna with SMA plug. This is required for operation.
- Current draw: approx. 40 mA

2.7.3 GPS/Galileo

2.7.3.1 Applications

With an update rate of 4 Hz, the GPS/Galileo module provides global position data, which can be recorded as additional signals, parallel to the CAN data.

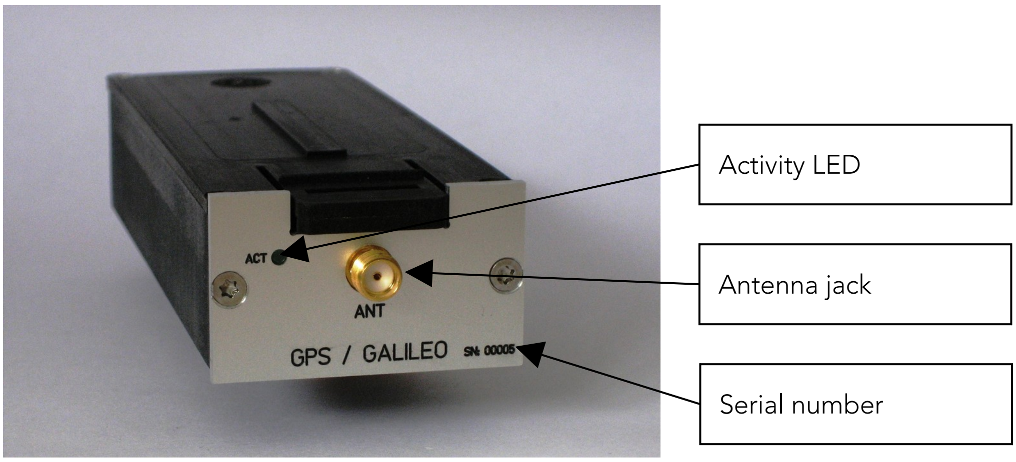
The receiver can process signals from the US GPS system or, alternatively, once the appropriate firmware has been downloaded to the module, the signals from the future European system "Galileo".

The last received satellite data is stored in the module in persistent form, which saves start-up time, providing the satellites are still available.

Currently, the time information in the GPS signal is not used for time control.

Note: This module is only supported by logger firmware version 1.0 and higher.

2.7.3.2 Module Placement



Illus. 2.7.3-1 GPS/Galileo module

The module is a quarter-unit wide and can be operated in any slot within the given chassis unit. Neither its position within a chassis unit nor the chassis level it occupies requires specification during configuration.

GPS Interface ACT LED (yellow)	Comment
Blinks	Blinks upon GPS fix, i.e. once a valid position has been determined. Simultaneously data output begins.

Table 2.7.3-1 GPS interface LED functions.

2.7.3.3 Configuration Parameters

The following values can currently be recorded as signals:

- number of received satellites,
- longitude [°],
- latitude [°],
- current velocity [km/h],
- compass (heading) [°], N=> 0°,
- elevation above sea level [m],
- status (no connection (=0), connection (=1), Egnos (=2))
- GPS date,
- GPS time and
- quality of precision (the raw value, directly acquired from the GPS signal, is converted to measurement uncertainty [in m], with a probability of 95%).

The values s*tatus* and *number of received satellites* are transmitted by the module as soon as it is recognized by the logger. All other values are sent after the first GPS fix, i.e. once a valid position has been determined. Until then they have the value NaN (Not a Number).

For details on the application of parameters in configuration, see Chapter Error: Reference source not found.

2.7.3.4 Technical Data

- Dimensions (W x H x D): ARCOS: 42 mm x 25 mm x 114 mm
  µCROS: 25 mm x 8 mm x 80 mm
- Weight: ARCOS: approx. 65 g
  µCROS: approx. 18 g
- Operating temperature: -40 °C to +85 °C
- Input voltage: ARCOS: 6 to 50 V DC
  µCROS: 6 to 36 V DC

- GPS: Requires an active antenna, with phantom power supply (approx. 3 to 3.3 V).
  SMA jack
  maximum rate, 4 Hz
  sensitivity, -148 dBm
  Start-up time, approx. 34 seconds (with good satellite visibility).
  precision, 2.5 CEP
- Current draw: approx. 45 mA

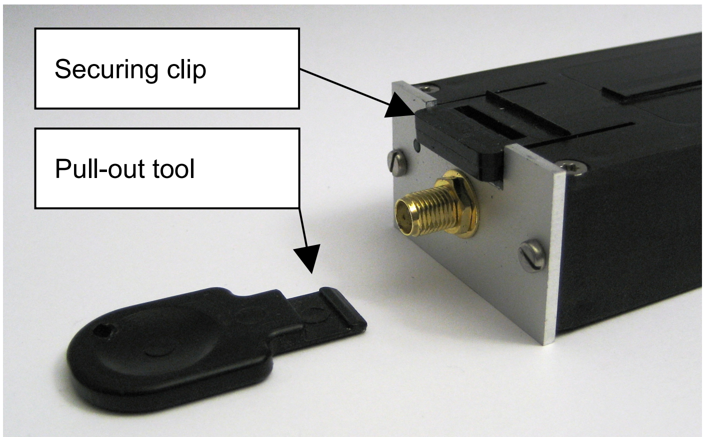
2.8 Mounting the Slide-In Modules

2.8.1 Mechanical Installation

The slide-in modules for both chassis units (upper, as well as lower) are not screwed, but only clipped into place. A plastic clip at the front secures the slide-in module. This clip can be released either with the pull-out tool, as shown in 2.8.1-1, or manually, by pressing it down. The slide-in module is then simply pulled out.

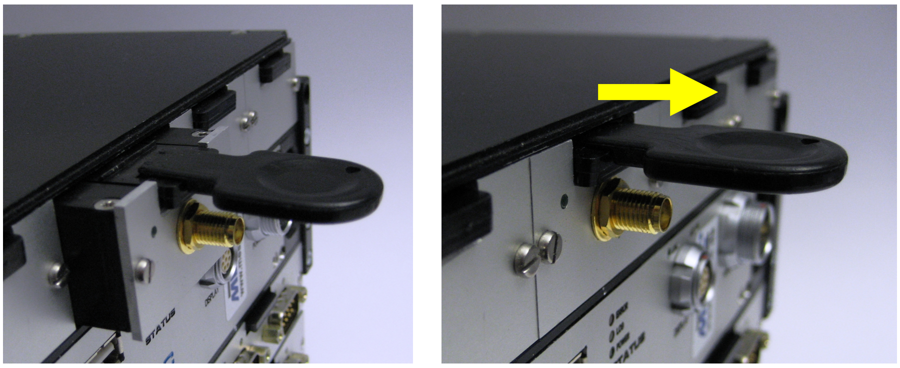
Important

The modules should only be removed or inserted after disconnecting the power. This is necessary, on the one hand, to protect the modules electrically, but also because of the initialization procedures performed once the logger is started.



Illus. 2.8.1-1 Securing clip on slide-in module, with pull-out tool.

Insert the pull-out tool, with the hook pointed downward, into the gap between the clip and the chassis unit housing. Once it clicks into place, pull it out, along with the slide-in module (see 2.8.1-2).



Illus. 2.8.1-2 With the tool inserted, the module can be slid out.

2.8.2 Initializing the Modules

2.8.2.1 Upper Expansion Chassis Unit

The slide-in modules of the upper expansion chassis unit require no manual connection establishment. These modules (GPS, WLAN, etc.) are automatically initialized when the logger is started. However, they do not appear on the Web interface.

2.8.2.2 Lower Expansion Chassis Unit

A variety of different LIN, CAN, FlexRay and Analog/Digital I/O interfaces can be combined here. The modules of a given type must be assigned channel numbers so that the logger can unambiguously assign a channel jack to the signal configurations. The numbering of each module type must be unambiguous; e.g., the number of each CAN channel is unique, and each LIN channel has a unique number with respect to other LIN channels. It is permitted though to have a LIN module numbered one, while a CAN module has the number one, as well. Numbering need not be consecutive (though it is recommended!).

This number must also be assigned after each replacement or addition of an interface, using the individual serial number of the specific interface.

The assignment of a node number to an interface is done through the Web interface included in the logger firmware, and which is used for all interface types (see Chapter 2.9 Web Interface, page 43).

2.8.2.3 Adjusting for Changes in Installed Interfaces

Once the Web interface is started, all the interfaces are displayed that either were already recorded in the logger or are found as new. The "Actions/Delete" function can be used to delete all the interfaces reported to be "offline".

Upon system start, the logger searches for all the interfaces that have been assigned an offset value. If any module is no longer installed in the logger, this will extend booting time, since the system continues searching for the particular module until a timeout marks it as "offline". It is therefore recommended that you delete non-installed interfaces from the list.

In order for the nodes to be consecutively numbered, the offset must now be set in each interface. Within an interface, the nodes are numbered from the left to the right. If, for example, in a four-channel interface node, the numbers 5 to 8 are to be created, an offset of 4 must be set. After specifying the offset, it must be written to each separate module by clicking "Send".

To identify a specific interface within the chassis unit stack, you can use the respective line in the list to activate the "blink" function. This causes all the activity LEDs of the specific interface to blink for approx. five seconds.

All settings are initially temporary and can be reversed using "Restore logger settings". Only by clicking "Write settings to logger" are the settings permanently stored in the logger.

A node number is assigned to each interface as a component – not to its position within a chassis unit! If the interface is re-plugged within the chassis units (e.g., from level 1 to level 3), it takes its node number with it.

Important

If interfaces are replaced, the logger must be rebooted. This guarantees that the configuration settings are also set in the interface (WoX, No-Message-Loss, etc). Otherwise old settings might be used.

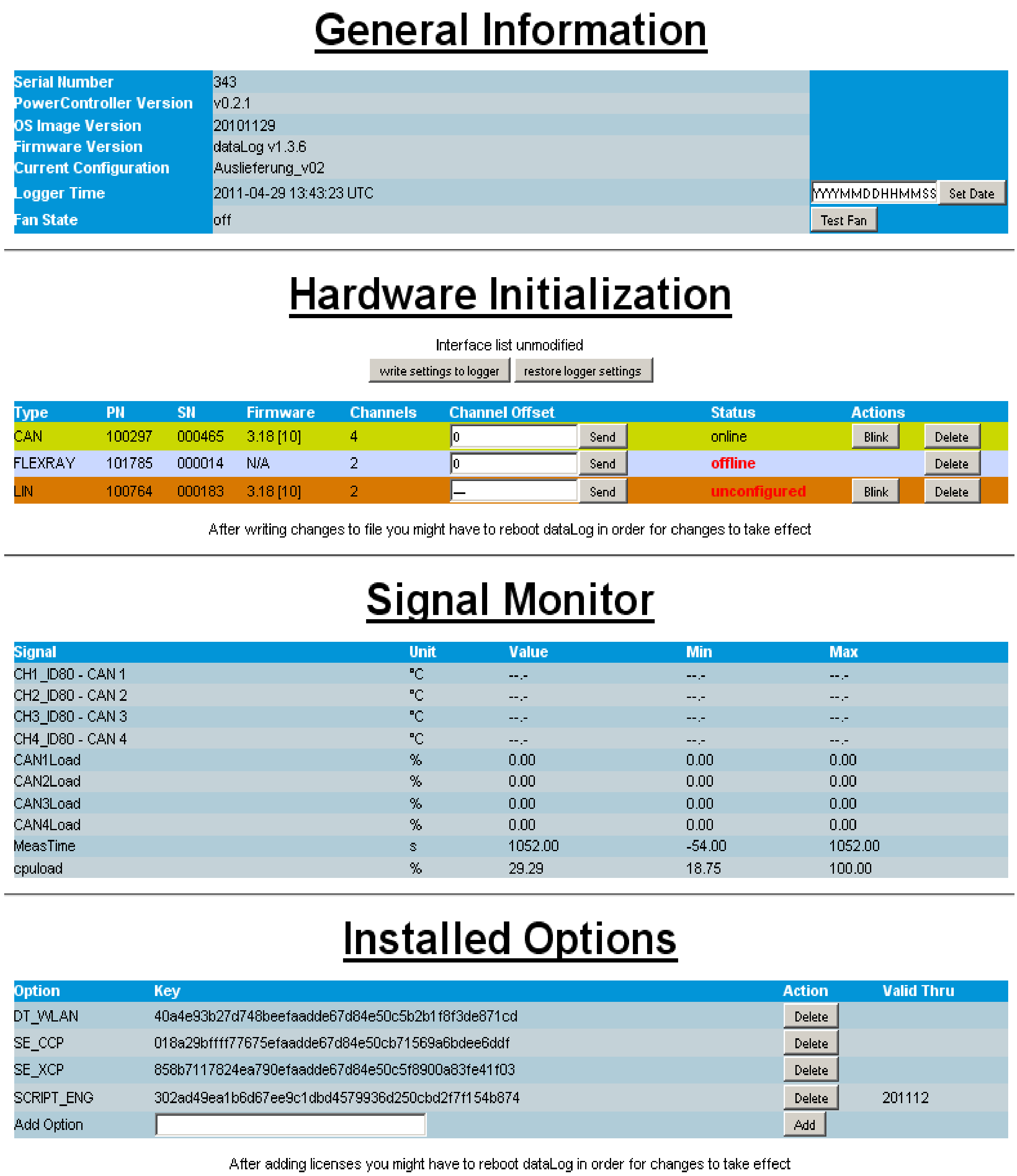
2.9 Web Interface

For quick information about the logger, we have integrated a Web interface in the logger firmware. It offers the user not only information about the component versions, but also a handy tool for setting the time for the logger, initializing the hardware, specifying logger options and checking the current values of all the signals defined in the configuration.

To access the configuration tools:

1) Start the logger and run it in normal measurement operation.
2) Unplug the display from the basic platform and instead connect the logger to a PC (Ethernet adapter cable; set the PC network settings, making sure the computer is in the same subnetwork as the IP 192.168.1.88).
3) Start an Internet browser (optimized for Firefox).
4) In the address bar enter: "http://192.168.1.88:8080" (without quotes).

A screen appears with approximately the following content:



General Information

Serial Number	343	
PowerController Version	v0.2.1	
OS Image Version	20101129	
Firmware Version	dataLog v1.3.6	
Current Configuration	Auslieferung_v02	
Logger Time	2011-04-29 13:43:23 UTC	YYYMMDDHHMMSS Set Date
Fan State	off	Test Fan

Hardware Initialization

Interface list unmodified

write settings to logger | restore logger settings

Type	PN	SN	Firmware	Channels	Channel Offset	Status	Actions	
CAN	100297	000465	3.18 [10]	4	0 Send	online	Blink	Delete
FLEXRAY	101785	000014	N/A	2	0 Send	offline		Delete
LIN	100764	000183	3.18 [10]	2	— Send	unconfigured	Blink	Delete

After writing changes to file you might have to reboot dataLog in order for changes to take effect

Signal Monitor

Signal	Unit	Value	Min	Max
CH1_ID80 - CAN 1	°C	--.-	--.-	--.-
CH2_ID80 - CAN 2	°C	--.-	--.-	--.-
CH3_ID80 - CAN 3	°C	--.-	--.-	--.-
CH4_ID80 - CAN 4	°C	--.-	--.-	--.-
CAN1Load	%	0.00	0.00	0.00
CAN2Load	%	0.00	0.00	0.00
CAN3Load	%	0.00	0.00	0.00
CAN4Load	%	0.00	0.00	0.00
MeasTime	s	1052.00	-54.00	1052.00
cpuload	%	29.29	18.75	100.00

Installed Options

Option	Key	Action	Valid Thru
DT_WLAN	40a4e93b27d748beefaadde67d84e50c5b2b1f8f3de871cd	Delete	
SE_CCP	018a29bffff77675efaadde67d84e50cb71569a6bdee6ddf	Delete	
SE_XCP	858b7117824ea790efaadde67d84e50c5f8900a83fe41f03	Delete	
SCRIPT_ENG	302ad49ea1b6d67ee9c1dbd4579936d250cbd2f7f154b874	Delete	201112
Add Option		Add	

After adding licenses you might have to reboot dataLog in order for changes to take effect

Illus. 2.9-1 Web interface (ARCOS dL and µCROS)

The screen page is divided into three main parts:

a) General information = basic module status information

Element	Meaning	Function
Serial number	Unique front plate number (unchangeable)	Display
Power controller version	Version of the power controller in the logger front	Display
OS Image version	Version of the logger operating system	Display
Firmware version	Version of the measurement program (logger firmware)	Display
Current configuration	Path prefix/measurement task name of the current configuration	Display
Logger time	Current time of the logger (in UTC! – i.e. Central European standard time -1 h, or summer time -2 h) To manually change the logger time, enter the current date with the time, in the format YYYYMMDDhhmmss (without any separators, such as : or .) and click on "Set time". The new time is set the next time the logger is started.	Display , input field and button
Fan state since 1.2.4	Status of the fan. To check the proper functioning of the fan, click "Test fan".	Display and button

Table 2.9-1 General Information – screen elements

b) Hardware Initialization = executing/checking the assignment.

Various interface types have different colors.

Element	Meaning	Function
PN	Product number of the found interface.	Display
SN	Serial number of the found interface.	Display
Firmware	Firmware of the interface hardware LPC [MPC].	Display
Channels	Number of nodes of the individual interfaces,	Display
Channel offset	Offset for the number of the first node of the interface (end input by clicking "Send"). The channel number in the logger is the sum of the offset + node number of the module. E.g., if the offset is 4, the first node of the module has channel number = 5 in the logger.	Input field with button
Status	Activity of the interface (on/offline)	Display
Actions/blink	Identifies the interface – at the respective interface all CAN activity LEDs start blinking at the CAN node, for 5s.	Button
Actions/delete	Deletes the entries from the list.	Button
Write settings…	Writes the settings to the logger.	Button
Restore logger settings	Restores the previous logger settings.	Button

Table 2.9-2 Hardware Initialization – screen elements

c) Signal Monitor = all defined measurement signals

Element	Meaning	Function
Signal	The name of the signal – corresponds to that in the display.	Display
Unit	Unit of the signal – corresponds to the display.	Display
Value	Current value of the signal.	Display
Min	Smallest signal value measured.	Display
Max	Largest signal value measured.	Display

Table 2.9-3 Signal Monitor – screen elements

d) Installed Options = Activates or displays additional logger functions.

Note

The screen page is current only at that instant when it is created by the Web browser. The page is not continuously refreshed. Particularly when you wish to monitor values during measurement, the display must be manually refreshed (by frequently pressing, depending on the browser used, "F5" or the "refresh" button).

2.10Display

The display is the input station for entering values in the logger, as well as the output station for displaying logger values and messages. Technically, measurement operation is always possible without a display. The display is hot-pluggable, meaning it can be connected or unplugged at any time during operation without loss of measured data.

If the display is unplugged and the configuration requires driver input, the system uses the configured default settings.

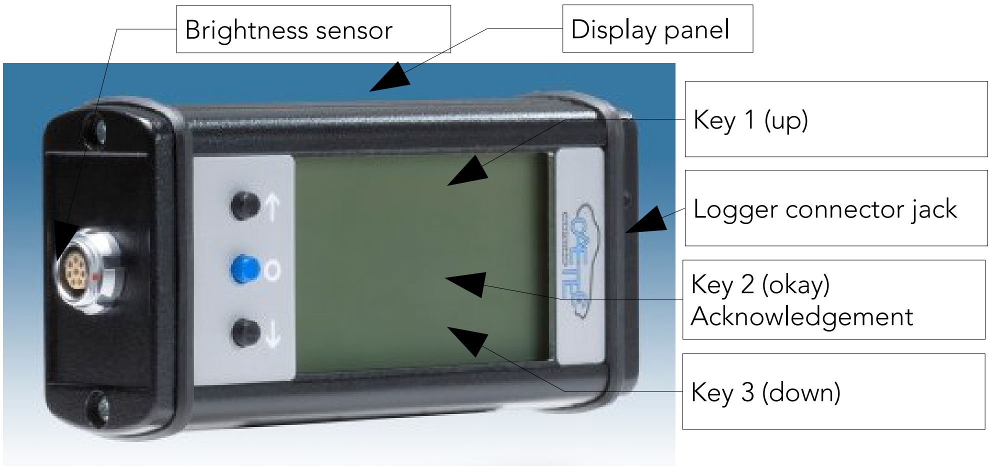
The display enables the logger operator to monitor measurement values and logger status, to select a "Track" and mark its end, as well as to set event markers in the log file ("Trigger from display").

In addition, alarms and messages are output to the display, warning the driver about critical conditions of the test object.

These messages are displayed on a 4-line screen, with 20 characters per line, while the three keys control display content and acknowledge messages and alarms.
The USB interface integrated in the display is at present nonfunctional.

To enable flexible display installation (choice of connections on the left or the right), display content can be rotated 180°.
An integrated sensor adapts display brightness to ambient light, contrast is adjusted manually (the set value is automatically adjusted for temperature).
The required parameters are set using an additional display dialog.

2.10.1 About the Connections and Keys



Illus. 2.10.1-1 Display design

2.10.2 General Connection and Key Assignment

Operating the device involves no input of numbers or values. The display instead offers a selection (which, depending on the configuration, can comprise only a single entry). Keys 1 and 3 are for changing the selection. Once you have finished selection, you simply acknowledge by pressing "okay", i.e. the middle (blue) Key 2.

Standard pressing duration for the "okay" key is approx. one second. This avoids hasty, unintentional acknowledgment. After a one-second delay, a brief confirmation tone sounds.

Key Assignment	
1 (< 1 s)	Up
(> 1 s)	Fast up (accelerates the longer it is held).
2 (< 1 s)	Starts/Stops the list scan.
(> 1 s)	Switches the screen page, acknowledges input.
(> 3 s)	Sets a trigger mark in the log file.
3 (< 1 s)	Down
(> 1 s)	Fast down (accelerates the longer it is held).
1 & 2 & 3 (> 1 s)	Switches to the setting dialog.

Table 2.10.2-1 Display key functions

Display Plug Pin Assignment	
1	Supply -
2	Supply -
3	Supply +
4	Supply +
5	Data 1 (RX+, orange and white)
6	Data 2 (RX–, orange)
7	Data 3 (TX+, green and white)
8	Data 4 (TX–, green)

Table 2.10.2-2 Assignment of the Fischer display connection plug

2.10.3Display During Start

As soon as the logger is started with a connected display, or when the display is plugged in during logger operation, the display and the logger start "synchronization". For its duration, the start screen appears on the display, presenting the CAETEC logo above the text "Waiting for logger".

Illus. 2.10.3-1 Display with start screen

Aside from the text, three dots appear in sequence, providing information about the status of the connection being established. The following table explains the meaning of the dots.

Number	Current Action
0 dots	Basic functions are initialized. The display is in operation but without external connections.
1 dot	The display has been assigned its network address by the logger; i.e. it is physically connected to the logger.
2 dots	The display has found the logger and begun communication.
3 dots	The display has imported the statistical data from the logger needed for operation (the number of channels, channel names, etc.)

Table 2.10.3-1 Meaning of the dots on the start screen.

Finally the screen switches to dynamic measurement-data display mode, if no track has been previously defined (cf. Chapter 2.10.4.2). Once the logger is ready for measurement (with booting completed, yellow LOG LED lit or blinking), synchronization is generally completed within seconds.

Note

If the display stops at three-dot status, this can indicate that no signals have been configured for output – in which case, you must check and modify the configuration.

2.10.4Operation

The following description of logger operation assumes a totally started logger, with connection established to the display (start screen has vanished).

2.10.4.1System Start without Track Selection

In this logger mode, the display switches directly to "View measured data" (cf. Chapter 2.10.4.4). The user need not make any entry or selection. (Measurement is performed without track assignment, and file names are appended with a time stamp.)
In all other respects, the display behaves the same as when the system is started with track selection, once that selection has been completed.

2.10.4.2System Start with Track Selection

When the system is started with track selection, you have two options:

1.) to select a new track, because either a new configuration with track selection is being applied for the first time or the last measurement was completed with "End Track", or
2.) to continue measurement with a track that was not yet completed.

Case 1) Selecting a new track:

Illus. 2.10.4-1 Track selection screen (example)

2.10.4-1 shows an example of the display screen for track selection. To start with, the first lines display the vehicle number (the "path prefix" taken from the configuration file), the version number of the configuration ("cfg_identity"), together with the date and time.
Next, the small black triangles prompt the driver to use the up/down keys to select the track from a list (stored in the logger). If there is no response, a steady tone sounds until the driver selects the track.

The next screen page (cf. 2.10.4-2) shows the selected track. If a mistake has been made, the arrow keys can be used to change the selection. (Change from “Have a good trip” to “Change”, then acknowledge with “okay” – thus returning to Selection.) If the selection is fine, acknowledge with “okay” (blue key). Brief warning tones, two seconds apart, prompt the operator to acknowledge. If the user fails to acknowledge the selection within 10 seconds, it is automatically acknowledged.

Illus. 2.10.4-2 Dialog screen for confirming track selection.

In summary:
- Press the “up”/”down” key to select the track, then end with “okay” (Key 2, duration > 1s).
- Confirm the selection with “okay” (Key 2, duration > 1s) (”Drive safely”). If necessary, use the “up”/”down” key to open “Change” and correct your selection.

If the display is plugged in after the system was started and no track has been assigned, the track is now selected. In other words, after plug-in, the same cycle takes place as occurs at system start with a plugged-in display. (On the other hand, if the display was unplugged during operation and then plugged back in – i.e. after a track had already been assigned – track selection is naturally not repeated.)

Case 2) Continuing a track

2.10.4-3 shows the screen for continuing a track or measurement. This dialog appears whenever the logger was shut down without ending a track. This can be useful, for example, when resuming track measurement after a break.

Illus. 2.10.4-3 Dialog screen for continuing a track.

Confirming this dialog with “okay” starts measured-value display (normal mode). If no entry is made within ten seconds, the logger automatically continues, with “yes” .

Should you at this point decide to change the track after all, simply switch to "modify" by pressing either of the arrow keys. Once the function "modify" is confirmed with "okay", the display returns to the track selection dialog.
The selection of a new track ends the current measurement and closes its data set, to begin a new one (standard procedure does not include data transfer. This is only started, if depending on the configuration, the measurement was ended by stopping the vehicle, for example, by means of a clamp 15 voltage drop).

Note

During track selection, the logger continues to record measured data. Neither track selection itself nor the involved dialog interferes with data recording.

2.10.4.3Shutting Down the System

System shutdown is initiated through Start/Stop signals at the logger. (The conditions depend on the configuration – cf. Chapter "Configuration"). In other words, one cannot directly stop the logger from the display. However, certain shutdown circumstances require input from the operator, which is described here.

It may be necessary, depending on the configuration, to enter End Track or specify the data transfer destination. 2.10.4-4 shows the dialog that may display at shutdown.

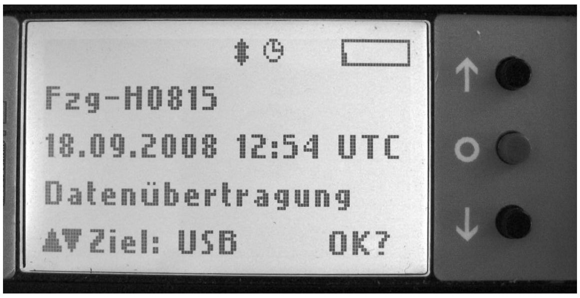
Illus. 2.10.4-4 "End Track" input dialog at shutdown.

The default value for the "End Track" dialogue is "no". If the operator fails to respond within approx. ten seconds, the logger shuts down without ending the track.

By using the arrow keys, the operator can switch from the default value "no" to "yes" and end with "okay". If the operator selects "no", the logger is shut down without further query. A "yes" reply closes the current data set. If the configuration is set accordingly, data transfer is then executed.

There are two alternatives for saving the data:
- transfer without operator input via a preset path; for example, always via WLAN, or
- transfer with operator input, i.e. selection of the transfer method.

If selection is the configured option, the user is first prompted to specify the target. 2.10.4-5 shows the involved dialog.



Illus. 2.10.4-5 Selecting the data transfer target.

In the example shown in 2.10.4-5, the display proposes "USB" as the target. Depending on what was activated in configuration, the operator can use the arrow keys to select from the following targets:

- "WLAN"
- "USB"
- "PPP"
- "---" (No target, therefore no transfer.)

Once WLAN, PPP or USB has been selected (and acknowledged with "okay"), transfer begins. The logger displays, during the transfer process, each step as it is executed – subsequent screen pages indicate that the data set is being closed, packed ("zipping") and transmitted. Depending on the quality of the connection, transfer via WLAN can take a while. The progress bar shows the transfer progress (in percentage), allowing you to estimate the total duration of the transfer.

If malfunction occurs during transfer, a screen page displays an error message (see Appendix). Then, after a ten-second timeout, the logger shuts down and switches off.
As part of the data transfer process, the configuration and logger firmware are also updated. This involves the messages "GetUpdate", "Checking" and "Processing" – only briefly mentioned here, but described in full detail later.

Note

For operation in track mode, please bear in mind that without an End of Track no data transfer takes place.
During data transfer, it is not possible to begin a new measurement.
[1.3]Depending on the configuration, a measurement can continue in parallel with data transfer, or a new one can be started.

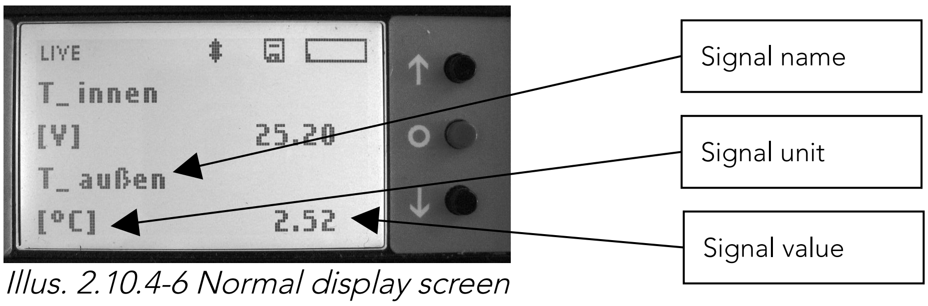
2.10.4.4 Viewing Measured Data (Normal Mode)

After system start, the data is automatically displayed in normal mode. Normal mode means that the data channels are displayed consecutively, at five second intervals, in the order defined in the measurement configuration (cf. 2.10.4-6 – identified in the display by the word "Live", along with a double arrow in the upper left corner).

A single screen page displays up to two channels, with two lines per channel. The first line of each channel displays the channel name, followed by the unit and the current value in the second line.
In the event of a (configurable) signal timeout, the display shows horizontal lines (--.-) instead of a value. In the event of signal value overrun (e.g., division by 0 in computation, or an incorrectly scaled CAN signal), the text "Infinity" is displayed instead of a signal value.

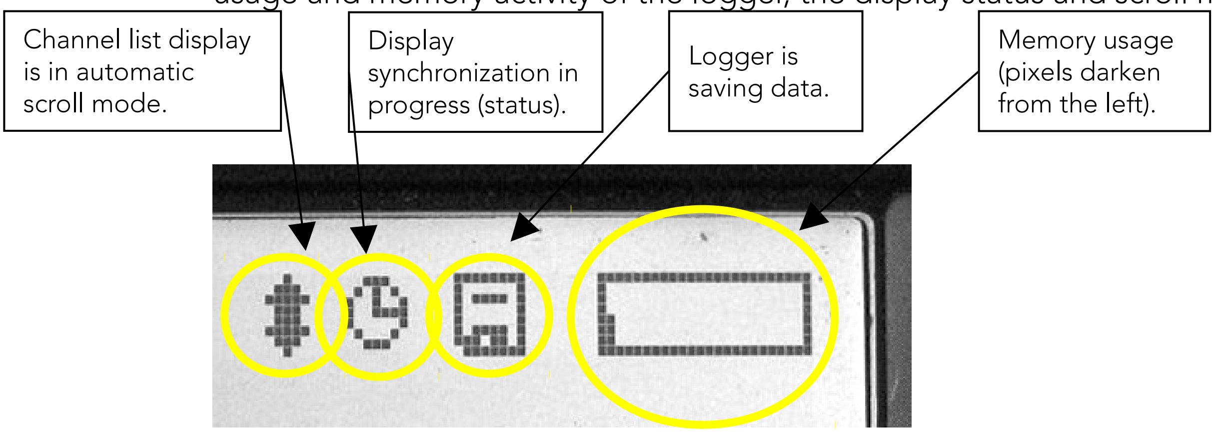
By briefly pressing (arrow) Key 1 or 3, you can stop automatic scroll and change the screen manually. A longer press of this key starts a quick scroll through the measured data, keeping it pressed accelerates the stroll. The manually selected reading remains on display for a configurable duration (default value = 2 h), after which the system reverts to automatic scroll (for more information, see Error: Reference source not found Error: Reference source not found).

To activate automatic scroll sooner, all it takes is a brief press (<1 s, no tone) on Key 2 (blue).
When automatic scroll is active, the symbol "↕" appears at the upper edge of the display.



Illus. 2.10.4-6 Normal display screen

Four symbols in the upper right corner of the screen indicate memory card usage and memory activity of the logger, the display status and scroll mode.

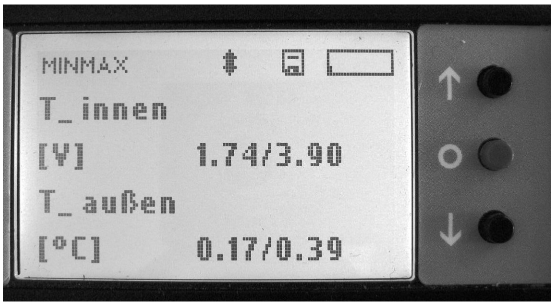


Illus. 2.10.4-7 Displayed logger status messages.

The display status symbol (clock) indicates that, due to specific circumstances (e.g., following message display or a booting phase), the displayed measurement values are not totally synchronous with the current measured data and are being refreshed. Once this is completed, usually within seconds, the clock vanishes.

2.10.4.5Displaying the Minimum/Maximum Values

Aside from displaying the current measurement values (normal display mode), you can also switch to viewing minimum/maximum values. In normal mode, you simply press the "okay" key (blue) until you hear a tone (approx. 1 second). This method is also used to switch from Min/Max display back to normal display mode.
2.10.4-8 shows the Min/Max display screen, identified by the text "MINMAX" in the upper left corner of the display).



Illus. 2.10.4-8 Min/Max display screen

In two lines for each channel, the screen displays its name, unit and a pair of values. The value left of the separator "/" denotes the minimum, the value to the right is the maximum of the current track. For very long numeric representations, such as 12345.6/ 78901.2, the display is automatically shifted horizontally ("horizontal scroll"), so all the digits are visible.

As in normal mode, in Min/Max display mode it is possible to scroll either automatically or manually.

Note

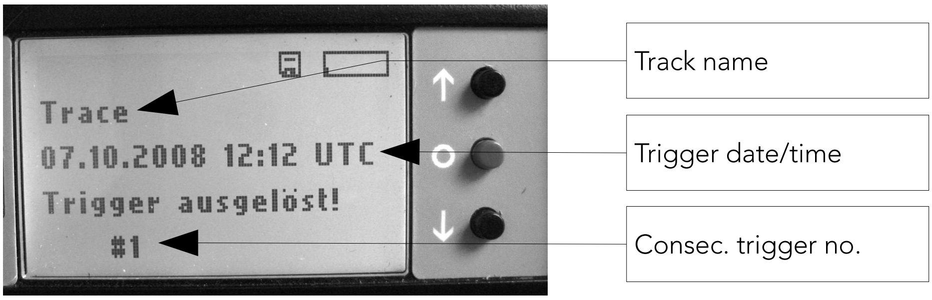
The minimums and maximums are valid for the duration of a track. If a single track spans multiple shifts/driver changes and pauses, the Min/Max value is valid across multiple logger starts.

2.10.4.6Setting Display Triggers

If anything unusual occurs during measurement, if events occur that need to be logged, you can mark the event via the display, by setting a "trigger".

It is activated, during normal display or Min/Max display, by continuing to press the "okay" key beyond the first signal tone, until the screen shown in 2.10.4-9 appears and a double tone sounds (after approx. 3 s).
In addition to the track name, the screen displays the date, time (UTC-standard!) and the track-specific consecutive number under which the trigger has been set. After approx. five seconds, the display automatically switches from the trigger screen back to the previous screen.

The time, date and number uniquely identify the trigger. Other information specifying what happened at the time of the trigger can be added later (e.g., in a driver's report).



Illus. 2.10.4-9 Display screen after activating a trigger.

Through its name 'trig_display', this trigger can be configured, for example, to start methods or data transfer. You should bear in mind though, that the display trigger is only a trigger pulse with a duration of two seconds, as opposed to the parameterized triggers (for more information, see Chapter Error: Reference source not found Error: Reference source not found, page Error: Reference source not found).

The trigger is recorded in the log file of the measurement.

Example of an entry in the log file:

...
2008-10-01 13:54:25; TRIGGER; display; 1
...

2.10.4.7 Displaying Setting Dialogs

The setting dialogs set parameters in the display or show status information (e.g., software versions, contrast settings, etc.):

Display Setting Dialog	Function
Language[1.20]	Sets the dialog language.
Illumination opt.	Sets the max. brightness of background illumination (control range)
Contrast	Sets the contrast.
CPU load	Displays the usage of the CPU.
Memory usage	Displays the used space on the memory card.
Rotate display	Rotates the display screen 180°.
Restart logger	Ends the current measurement and restarts the firmware without booting the logger.
Softw. versions	Displays the software versions of the display, dL (logger firmware) and pC (power controller/front).
Expanded menu[1.20]	Sets the signal tone (on/off).
Use default settings	Resets display brightness, contrast and rotation to the factory defaults.
Close menu	Ends the setting dialog.

Table 2.10.4-1 Setting dialogs for configuring via display.

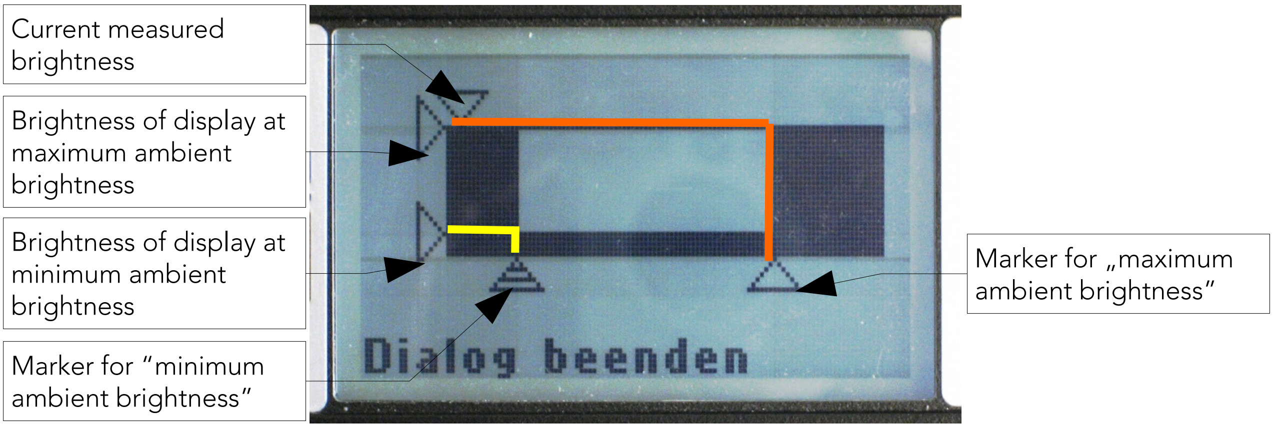
The dialogs marked [1.20] are only available with display firmware version 1.20 and higher.

To start the display mode for the setting dialogs, simultaneously press all three keys. To end, select the dialog item “Close menu”. Keys 1 and 3 shift input from one field to another (the first character is inverted as a cursor). Longer pressing of Key 2 starts and ends the setting dialog and switches the screen back to the previous menu item.

Setting the Illumination Options

The display is equipped with a brightness sensor that responds to ambient brightness. You can use the dialog “Illumination opt.” to set the brightness of the background illumination with regard to the surroundings.

This dialog offers the setting options shown in 2.10.4-10:



Illus. 2.10.4-10 Setting dialog for the illumination options

The unshaded surface within the markers defines the active range, within which the brightness of the background illumination is regulated according to the sensor value.

The triangle at the upper edge shows the current value measured by the sensor. In darkness, it moves toward the left, with increasing brightness toward the right.

The colored lines highlight the pairs of triangles defining the brightness of the display with respect to the ambient brightness (2-point representation). The "yellow" pair defines the point of least brightness, the "orange" pair defines the point of highest brightness of the background illumination.

Use the left triangle under the graph to set the level of ambient brightness below which the display should not get darker. The corresponding triangle at the left edge defines how bright the display should then be.

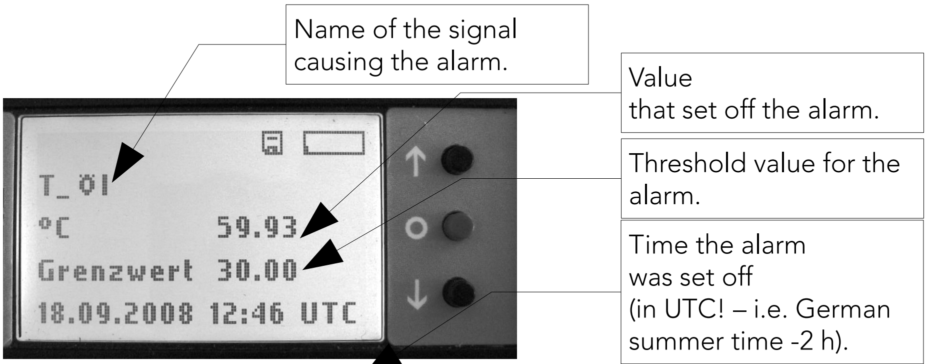
Use the right triangle under the graph to set the level of ambient brightness above which the display should not get brighter. The corresponding triangle at the left edge defines how bright the display should then be.

The setting markers can be shifted, using the arrow keys. The markers set the border ranges for illumination control.

- With the display arrow keys, select the particular setting marker (this is then displayed as hatched). Acknowledge selection by pressing "Okay" for one second (marker is then displayed as solid).
- Use the arrow keys to shift the activated marker.
- The marker shift must also be acknowledged by pressing "Okay" for one second. Then proceed to the next marker.
- Exit the dialog by marking the text "Exit dialog" and then pressing "Okay" for one second.

2.10.4.8 Alarm Management for Limit Violation

If a signal deviates from a valid range into an alarm range, a warning tone sounds, while the screen in 2.10.4-11 is displayed.

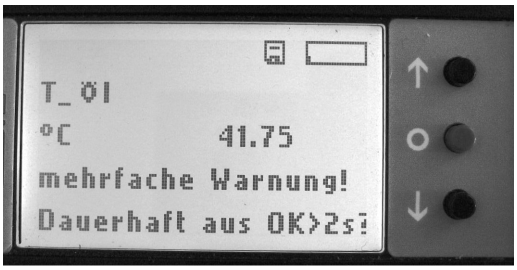


Illus. 2.10.4-11 Display screen for a signal alarm (example).

The first item displayed is the name of the signal that set off the alarm. Next you see the value of the signal that set off the alarm, and which threshold (= limit) was set for the alarm, followed by the time when the alarm occurred.

The warning tone sounds until the message is acknowledged by briefly pressing the "Okay" key. Another press on "Okay" > 1 second returns the display to normal mode, unless another alarm has occurred, in which case the next alarm is displayed. Alarms are collected and then, one after the other, are output to the display.

In the event of repeated alarms set off by a signal, after the fifth acknowledgment (counting all violations of upper and lower thresholds) you are given the option of permanent acknowledgment – indicated by the displayed prompt "Alarm off" (cf. 2.10.4-12).



Illus. 2.10.4-12 "Alarm off" prompt, displayed after the fifth alarm (example).

Once this dialog is acknowledged by pressing "Okay" (Key 2) for longer than three seconds, there is a single signal tone, followed after approx. three seconds by a double tone. Alarms for this signal are subsequently suppressed for the duration of the current measurement (track). No further limit violations are reported for this signal until the measurement is ended or a new track is selected.

Whenever a signal is outside the valid range from the beginning, from logger start-up, instead of producing a warning for the signal, the system writes a log file entry. Whenever a signal remains in the alarm range, warnings are set off at ten-minute intervals.

Violations of alarm limits are recorded in the log file.
Here are a few examples of pertinent log file entries:

Entry for a signal running out of the valid range:

2007-12-13 14:30:18; OUT OF LIMIT; T_OIL; 30.000000; 30.146844, 1, 2007-12-13 14:30:18

Entry for a signal entering the valid range:

2007-12-13 14:30:20; INSIDE LIMIT; T_OIL; 30.000000; 29.956109; 2007-12-13 14:30:20

Timestamp of the alarm
Value of the alarm signal
Defined limit
Triggering signal
Identifier of the entry
Timestamp of the entry

Entry, with "Alarm off":

2008-06-06 07:40:11; LIMIT WARNING; OFF; T_OIL

2.10.4.9Handling Messages

Messages are texts that can be freely defined within the configuration. They are displayed once a "signal trigger" occurs. This means that messages can be used in any situation where a trigger can be defined for a signal.

The example in 2.10.4-13 shows a message that appears when a temperature exceeds the value of 50°C.

Illus. 2.10.4-13 Message with the free text "Temperature above 50°C".

The screen displays (line by line, from the top) the name of the configuration (pathprefix), the message text itself, and the date and time when the message has occurred.

The message remains on the screen until it is acknowledged by pressing "Okay" (= Key 2, with signal tone confirmation).
Like alarms, messages are collected in the background. In the event that additional alarms or messages occur during message display, they are displayed one after the other. Once a displayed message is acknowledged, the next one appears.

Just like with alarm messages, it is possible to permanently acknowledge free-text messages.

If so configured, each message can also be recorded in the log file.

Log file entries (excerpt example):

…

(1) 2008-08-25 08:30:48; START METHOD; Message 2; 2008-08-25 08:30:48
(2) 2008-08-25 08:30:48; NOTIFICATION; Memory 90% full
(3) 2008-08-25 08:30:49; STOP METHOD; Message 2; 2008-08-25 08:30:49
(4) 2008-08-25 08:30:55; NOTIFICATION WARNING; OFF; Message 2

…

Meaning:

(1) Message 2 was triggered.
(2) The text for Message 2 was displayed.
(3) Message was acknowledged.
(4) Message was switched off for the duration of the track.

2.10.5 Technical Data

- Dimensions (W x H x D): 115 mm x 60 mm x 33 mm
- Weight: approx. 210 g
- Operating temperature: -20 °C to +70 °C
- Input voltage: 6 - 50 V DC
- Maximum line length: approx. 10 m
- Maximum pressure on key: 30 N (3 kg) for 3 seconds.
- Current draw: approx. 100 mA 12 V

3 Configuration

3.1 General Information

3.1.1 Basic Principles

The logger is configured through an ASCII file. This file contains all the parameters for the logger. If LAPI-capable measurement modules are used for the CAN channels, the configuration file can also include the LAPI. The settings enable the logger to configure any LAPI measurement modules connected to the CAN bus.

It is equally possible, though to conventionally configure the measurement modules, without LAPI, thus treating them as conventional CAN users. In this case, only the CAN parameters for the module signals are declared in the logger configuration, and the logger performs no further configuration of the measurement modules.
This means it is possible to configure the modules in advance, separately from the logger, and then to give the logger a configuration file dealing only with data recording.

The only section of the ASCII file where conventional configuration differs from "LAPI" configuration is that containing the module configuration. The remainder is identical.

The logger can only configure the CAN measurement modules via LAPI if a driver exists for each module type, and the measurement module must be "LAPI-capable" (cf. the LAPI specifications).

The configuration file, regardless of whether it includes a LAPI section, is imported by the logger during the next data transfer. The file is then analyzed and stored in the logger. Any newly imported configuration is used upon the next system restart, and is already active after shutdown.

The configurations are not usually created manually, but by a so-called configurator, such as the one provided with the logger, "PHÖNIX". (This tool includes no LAPI configuration though.)

Various logger configurators are currently in development, so the configuration description presented here is based on the current logger functionality. The configuration involved with CEUS is described in a separate document.

Note

All names and designations, such as units, tracks, comments, etc. should always be set in in single quotation marks, to avoid problems with special characters and blanks.
This manual distinguishes configuration parameters typographically by setting them in the typeface courier.
Do not include empty entries! If an entry is not required, you should delete the entire line, or comment it out by inserting the character "#" at the beginning of the line.

3.1.2 Configuration Blocks

The logger configuration comprises groups of functions, each dealing with a coherent topic.

Group	Content
General options	global parameters comments track lists
General settings	CAN/LIN/FlexRay node settings data source definitions (CCP/XCP, KWP) Data transfer definitions (network and server), data set generation, file header and system state
Data sources	CAN/LIN/FlexRay/XCP/CCP/ADIO and GPS signals Computed signals (formulas) Internal signals (logger state)
Protocol	Special protocol services: KWP
Trigger	Starting/stopping methods
Scripts	Enhanced programming with scripts
Methods	time logs classing display output limit output min./max. values traces (raw data recording) notifications gateway

Table 3.1.2-1 The sections of a configuration

Within the configuration file, each group has its own section. The sections are marked by key terms. The sections are structured with a hierarchy similar to that of XML. It is not possible to randomly shift the blocks around within this structure.

For a detailed explanation of this structure, see the separate documentation[1].

The entries are combined in blocks beginning with a key word, set in angular brackets. Preceded by a slash, the given word also marks the end of the block, which must always be explicitly stated. All words must be lowercase (!!).

Example of how such a block is written:

```
<key word>
    parameter1=…
    parameter2=…
    parameter3=…
    …
</key word>
```

[1] Description of the Configuration File for CAETEC datalogger *ARCOS dL*, *µCROS*, CAETEC GmbH

Important

Blanks between parameters, equals signs and the assignment must be avoided, otherwise the line will not be completely read.
Any specification of names and designations (signal names, names of methods, paths, comments, etc) should be set in single quotes (' ') to avoid problems with such special characters as umlauts, hyphens and the like (a blank is considered a special character).

Example: name='Oil temperature'

3.1.3 Standard Logger Run Behavior

The following paragraphs describe the procedure of a typical, simple logger run, to illustrate the standard behavior of the logger. The logger run extends from start to stop of the logger, including data transfer. In side notes, we shall point out possible variations in this standard behavior.

The logger is permanently connected to the supply voltage (V+ and GND) and is started with Clamp 15 (Error: Reference source not found Error: Reference source not found). During the logger booting phase, the configured bus interface modules temporarily buffer any incoming bus messages. These are time-stamped the moment they arrive at the interface.

Once booting is completed and a track has been selected on the display screen (Error: Reference source not found Error: Reference source not found and Error: Reference source not found Error: Reference source not found), the running measurement values are displayed. However, the logger processes all the buffered data first (Backlog – indicated by a clock symbol at the top of the display), before proceeding, without interruption, to live display (Error: Reference source not found Error: Reference source not found).

As soon as data processing begins, the logger automatically starts a permanent recording in .atfx format (Error: Reference source not found Error: Reference source not found). The record contains, at equal time intervals, the values of certain bus signals defined in the configuration (Error: Reference source not found Error: Reference source not found, Error: Reference source not found Error: Reference source not found, Error: Reference source not found Error: Reference source not found, Error: Reference source not foundff. Analog/Digital I/O), GPS values (Error: Reference source not found Error: Reference source not found), as well as such internal values as logger runtime, bus load and error frame rate (Error: Reference source not found Error: Reference source not found).

Pressing the display trigger key sets a display trigger (2.10.4.6 Setting Display Triggers), displaying it with its timestamp and trigger number, Simultaneously the associated internal trigger event 'trig_display' would be initiated.

This can e.g. be used to trigger a data storage of a trace (raw bus data, Error: Reference source not found Error: Reference source not found). The trace data is stored with pre and post triggers in e.g. Vector ASCII format, the data file name is automatically appended with the date and time of the trigger (Error: Reference source not found Error: Reference source not found). The trigger event is recorded in the log file and in the trace file.

Switching off Clamp 15 triggers the end of the logger run. Once Clamp 15 is switched off (Error: Reference source not found Error: Reference source not found), the display immediately prompts whether the track is to be ended. If this is confirmed with "yes", the current data set containing all the measurement files is closed. This is followed by a dialog prompt to specify the target medium for the data (Error: Reference source not found Error: Reference source not found). After ten seconds, the default transfer path "USB" is selected. To establish connection, the logger searches for a USB stick prepared for data transfer, while the closed data set is compressed and finally transmitted (4 Data Transfer). Successful transfer is verified by computing a check sum on the storage medium and directly comparing it with the check sum made prior to transfer. Only then is the transmitted data deleted from the logger and memory space released for new measurements.

While the measured data is being transferred, the logger checks the USB stick for a new configuration file (4.4 Importing the Configuration to the Logger) or a firmware update file (5.3 Firmware Updates). If found, the new configuration is immediately activated, the update performed. Only then is the logger shut down and switched off.

The logger is now ready, possibly with a new configuration and/or updated firmware, for a new measurement.

3.1.4 Starting/Stopping the Logger

Once the logger is connected to the power supply, it can be started either by:

- hardware - signal voltage (Clamp 15), or
- message on a bus, WoX, e.g., Wake on CAN (= WoC).

As soon as Clamp 15 occurs, the logger immediately (delay < 0.5 s) begins recording. During the booting phase, the logger firmware is not yet able to process values, so the messages are first buffered in the interfaces and then processed once booting is completed.

The option of starting with Clamp 15 is always available, regardless of whether bus channels are configured as WoX.
This option exists parallel to start by bus message (e.g., "Wake on CAN" = WoC). This behavior also supports emergency start-up – ensuring there's always a way to safely start the logger (e.g., due to a configuration error, such as Start on a CAN message that never occurs).

If WoX is configured, the logger starts (just like with Clamp 15), as soon as a message (or the message/data defined in the configuration) is received on the activated channel.

Once the firmware is totally loaded, the buffered data (backlog) is processed. This is executed with full system performance (quicker than real time) and is usually finished within seconds. From this point on, the data is processed in "real time" – under these system conditions the delay is at least three to approx. 20 minutes, depending on the data rate in the interfaces.

In the event that *all* the wake conditions are eliminated during a measurement, the system initiates a logger stop. In this context, it is irrelevant whether a wake condition was active at logger start-up or was activated later. In the case of channels with WoX, the requirement is that there be no received messages (or the defined message/data) on the particular channel for the bus timeout period (see Error: Reference source not found Error: Reference source not found), i.e. for at least 10 seconds. This condition must be met on all the channels for which WoX is configured.

Once the configured post-runtime (Error: Reference source not found Error: Reference source not found) has expired, the current data set is closed and data transfer is started ([1.3]providing this has been configured). Once data transfer has been completed, the logger shuts down.

[1.3]If the involved display dialog is still on the screen or data transfer is still in progress, and a wake condition is reactivated, this immediately starts a new measurement, initially in the background. Once all the dialogs/the data transfer is finished, the screen returns to displaying a current measurement.

If all the start conditions are eliminated while the logger is still booting, then once the firmware is started, the data buffered up to this point (including any defined post-runtime) is processed and the logger is stopped (just as in a regular measurement). Since the state of Clamp 15 is not monitored until the firmware is running, the earliest point at which the logger can be stopped by an inactive Clamp 15 is when the firmware starts.

3.1.5 Persistence

How long the value of defined bus signals and internal variables remains stored in the logger can vary. In a number of cases, standard behavior can be influenced by parameter persistence. The value is accordingly available for a longer or shorter period even without an update or re-computation.

The following table lists the standard behavior. The value is preserved until the given event occurs. For the various signals the start value of all values/variables is NaN (Not a Number); for trigger counters and classing methods it is 0.

Quantity/Variable	Persistence
Bus signals	none
Computed signals	none
Internal signals	none, unchangeable
GPS signals	none
Analog/Digital signals	none, unchangeable
Trigger (trigger counter)	end of data set
Display trigger (trigger counter)	logger start
Classing	end of data set

Table 3.1.5-1 Standard persistence behavior

The persistence options are:

none — The value is preserved for the logger run. Even upon data transfer, the value is not reset, as long as the logger is not switched off or rebooted. Once the logger is switched off or restarted, or a new data set is begun, the values are initialized with the start value.

end of data set — The value is preserved until the end of the data set. Once the data set is closed (e.g., by End Track, a trigger or shutting down the logger), the value is reset and the new data set is begun with the reset value.

data transfer — The value is preserved until a data transfer is successfully completed. If data transfer is being performed in parallel with a running measurement, then the value is reset as soon as data transfer is successfully completed. To avoid parallel access to the variable, the current measurement is briefly paused and then continued with the reset values (there is no data loss, only a brief increase in measurement delay).

config — The value is preserved until a new configuration file is installed. Once the logger restarts due to activation of a new configuration, the value is reset and the new data set is begun with the reset value.

ever — The value is preserved forever; it is only updated but not reset. To reset it, you must import a special update packet, which then resets all the persistences.

3.1.6 Bus Channels/Nodes

Due to the modular design of the logger, it can be run with a variable number of interface modules.
To make sure the various signals are unambiguously assigned to their respective channels (also known as nodes) in the interface module, you must assign each channel a number.

This step is a prerequisite for further configuration. All other node settings can then be made through the configuration file.

The assignment of a node number to the connection number in the interface – so-called initialization – must be performed upon installation of each interface module, with the aid of the (logger-integrated) Web interface. The procedure is described in Chapter 2.8.2 Initializing the Modules.

The following text describes the various node parameters. Within the <settings>…</settings> section of the configuration file, the node settings are first grouped according to the key word of the connection type <can>, <lin> or <flexray>. Then each node is bracketed by <canch> (stands for "can channel"), <linch> or <flexraych> (do not forget to close the bracket with the corresponding </...>!). This section is to specified individually for each node/channel being configured.

3.1.7 CAN Channels

Due to its modular design, the logger can be equipped with a variable number of CAN interface modules.
Channel numbers are assigned to the CAN modules as described above, cf. 3.1.6 Bus Channels/Nodes and 2.8.2 Initializing the Modules.

The following parameters for CAN channels are grouped within the configuration file, bracketed by the key words <canch> and </canch> (stands for "CAN channel").

3.1.8 LIN Channels

Due to its modular design, the logger can be equipped with a variable number of LIN interface modules.
Channel numbers are assigned to the LIN modules as already described in Chapter 3.1.6 Bus Channels/Nodes and 2.8.2 Initializing the Modules.

The following parameters for LIN channels are grouped within the configuration file, bracketed by the key words <linch> and </linch> (stands for "LIN channel").

3.1.9 FlexRay Channels [1.3]

Due to its modular design, the logger can be equipped with a variable number of FlexRay interface modules.
Channel numbers are assigned to the FlexRay modules as described in Chapters 3.1.6 Bus Channels/Nodes and 2.8.2 Initializing the Modules.

The following parameters for FlexRay channels are grouped within the configuration file, bracketed by the key words <flexraych> and </flexraych> (stands for "FlexRay channel").

4 Data Transfer

4.1 Basic Information

Data transfer includes not only the transfer of measurement results from the logger to a storage medium, but also the transfer of a new configuration to the logger.

Since both data types are file-based, the same transfer channels are used for both types, even though they involve slightly different mechanisms and procedures.

One principle is that all data can be transmitted either over the WLAN or UMTS module in the upper chassis unit or via any USB interface (e.g., in the basic platform).

[1.3] The specific time for transfer via these interfaces is defined in the configuration (see Chapter Error: Reference source not found Error: Reference source not found). If not further specified, it is always performed at the end of a measurement, when the logger is shut down. If configured accordingly, transfer can also be performed in parallel with a running measurement. Whenever, during the transfer process, a new configuration file is loaded, any running measurement is automatically ended, the logger is restarted and the new configuration activated.

[1.3] If the frequency of data transfer needs to be restricted, then this restriction can be defined for each individual transfer path – either generally, by setting the parameter max_occurrence, or only for measured data transfer, by means of the parameter max_datatransfer. These restrictions can be overridden by individual transfer events. For details, see the description of the relevant parameters in Chapter Error: Reference source not found Error: Reference source not found.

When data is transferred, only already closed data sets are transmitted. Depending on the configuration, this can also include the current data set, providing it is closed at this point for transfer (for more information, see Error: Reference source not found Error: Reference source not found.

There is also the option of accessing the logger via Ethernet, by plugging an adapter cable into the display jack of the basic platform. However, this access requires detailed knowledge of the file structure, as well as login to the logger. This path should rather be reserved as a fallback solution for extraordinary circumstances.

For more information about operation, along with photographs of the screen output, see Chapter 2.10 Display.

4.1.1 General Transfer Procedure

The transfer steps are nearly identical for all transfer paths, with only small differences due to the requirements of the different transfer mediums.

When transfer is started (e.g., upon logger shutdown), if so configured, the display prompts selection of a transfer path.

Once the destination, i.e. a transfer path, has been selected, the logger attempts to establish connection. If this is successful, the logger searches the target for a new logger configuration. If found, it is first downloaded.

Next, the logger compresses the data sets – located within the logger's internal main directory (<root>\home\datalog), in a directory whose name was formed out of the configured "PathPrefix", the date of the measurement start, plus the track name (if there is no track name selected, "xxx" is appended as track name, if no track names are defined, then this part is omitted). Finally, a check sum is computed for the created file. Depending on the quantity of data, this can take several minutes. To prevent this process from exceeding the default timeouts, the parameters zipping and md5sum can be adjusted accordingly (see Error: Reference source not found Error: Reference source not found). Once packing is finished, the original measurement files are deleted.

At this point, the zip and log files are transmitted. If there is either insufficient memory or a loss of connection, the logger deletes only those zipped measurement files that were completely transmitted. The next time connection is established, the logger retries to transmit the data.

Once all the data sets have been completely transferred, the logger checks whether a firmware update packet is available for download and, if one is found, downloads it.

The transfer and its result (successful, or abort with error code) is recorded in the log file of the currently active data set, or else in the next one (the transmitted data sets being closed and thus unable to record any further log data).

In the case of transfers started at logger shutdown, once transfer is completed, the logger shuts down and switches off. In the process, any new configuration is already activated, and any new update is performed. So the next time the logger is started, it operates with the settings of the new configuration and/or updated firmware.

[1.3] When transfer is started during active measurement and the logger finds a new configuration, once transfer is completed, the logger automatically restarts, activating the new configuration.

This three-step order – new configuration first, then data sets, then firmware update – ideally ensures three things: that the logger always operates with the latest configuration, and measured data is transferred without delay; and, finally, that firmware is updated only when the transfer path is sufficiently stable.

4.2 Data Transfer via USB

For data transfer from the logger you can use USB storage media, either so-called "memory sticks" or external USB hard drives.
However, target media are restricted to those containing a key that enables the logger to identify them. This is designed to prevent unauthorized read-out from the logger by random media. Moreover, the medium must be formatted with the file system FAT or FAT32.

At the USB port, memory sticks are the most common medium, so for simplicity sake the following text refers to a USB stick.

4.2.1 Preparation

To structure data storage, the logger configuration allows the specification of separate directories for configurations (cfgdir) and measured data (datadir). It is also necessary to ensure that a particular logger receive only the configuration designed for that logger. Therefore the logger looks for the configuration in a directory bearing the name of the front number of the logger, created as a subdirectory of the configuration directory.

The USB stick for data exchange with the logger must therefore be prepared as follows:

- The file datalog.key is copied into the root directory of the USB stick.
- Once the directory <cfgdir>\<front number of the logger>\ is created in the root directory of the stick,
- if a new configuration is to also be transmitted, the file "datalog.cfg" is copied into the above directory.
- Finally, the directory <datadir> is created as configured.

Example:

As specified in the logger configuration:
The data directory is datadir=/test data.
The configuration directory is cfgdir=/testkonfig,

Front number of the logger = 42,
The stick is to be recognized at the PC under the drive letter "E:".

Directories to be created:
E:\test data
E:\testkonfig\42

Copy the file "datalog.cfg" (created for logger "42"!!!) into the directory "E:\testkonfig\42\".

4.2.2 Transfer Procedure

Thus prepared, the stick is inserted into any logger USB interface (e.g., in the basic platform). This can take place at any time during operation – as long as the logger is not booting or shutting down – or even before the logger is switched on.

Important

Before proceeding, be sure to allow the logger enough time to recognize the USB stick (just as is required under Windows). This generally takes only a few seconds (< 10 s) (also indicated by a brief flicker of the activity LED at the stick).

If so configured, once transfer is started, "USB" is selected as the transfer destination. If a valid stick is recognized, the closed data sets are compressed by the logger, and a check sum is computed.

Finally the zip and log files are moved(!) to the stick – i.e. once copying is finished, the logger directory for the measured data is empty.

If there is insufficient memory on the stick, transfer is interrupted, with an error message. Any files not completely transmitted remain on the logger.

If transfer was started upon logger shutdown, once the logger has switched off, the stick can be removed without risk.

[1.3] If transfer was started during an active measurement, the stick can be removed once the latter is completed, which is clearly indicated, either by the display, which returns to the normal screen; or by the LED at the stick, which goes from blinking to steady. If during the transfer process, a new configuration is found, the stick should not be removed until the logger has completely restarted and the new measurement is running.

4.3 Data Transfer via WLAN/UMTS

For transfer via WLAN or UMTS, the basic mechanisms or steps are the same as for transfer via USB. Only the transfer medium is different.

4.3.1 Preparation

The data server where data is to be stored must have an operating SSH server, as described in Chapter Error: Reference source not found. This server must also be able to compute md5 check sums.

Analog to the configuration of a USB stick, the directories for configurations (cfgdir) and measured data (datadir) specified in the logger configuration must be created on the server, as well as the subdirectory with the name of the front number.
The "datalog.cfg" is then copied into the latter.

Since secure access is already provided by the used access log (SSH/SCP), there is no need for a datalog.key file on the server.

4.3.2 Transfer Procedure

At the configured time (standard is upon logger shutdown), the logger tries to establish connection to the server.

Once connection has been successfully established and the logger has checked whether a new configuration is available, the closed data sets are packed and then transmitted. Once a data set is transferred, a check sum for the transmitted file (md5) is generated on the server and compared with the one on the logger. Only if these check sums match, confirming successful transfer, are the data on the logger deleted.

If the server runs out of memory or the connection is interrupted, data transfer ends and the check sums do no match. In this case, the data on the logger are not deleted.

Once the data sets are transferred, the logger checks to see if there is a new firmware update, and if it is found it is downloaded.

The transfer and its result (successful, or abort with an error code) is recorded in the log file.

Depending on the circumstances triggering data transfer, the logger then either shuts down and switches off, [1.3] continues measurement, or restarts in order to activate a new configuration.

4.4 Importing the Configuration to the Logger

A configuration is imported as part of measured data transfer, either via USB, WLAN or UMTS. At the beginning of data transfer the logger checks whether there is a new configuration available. If a new one is found, it is transmitted and imported by the logger. At the next start, the logger operates with this new configuration.

The location where the logger searches for a configuration is defined in the configuration settings of the current configuration. If required, a selection dialog can be displayed prior to transfer of the configuration.
In special cases, a special display adapter cable can be plugged in instead of the display, thus creating a direct Ethernet connection for manually transmitting a configuration.

Note

Whether the logger has read a configuration file is indicated by the file name "datalog.cfg". If the file name found in the configuration directory is unchanged following transfer, then the new file has not been imported (wrong directory or wrong name). On the other hand, if the file is accepted, then the extension .cfg is appended with the import date.

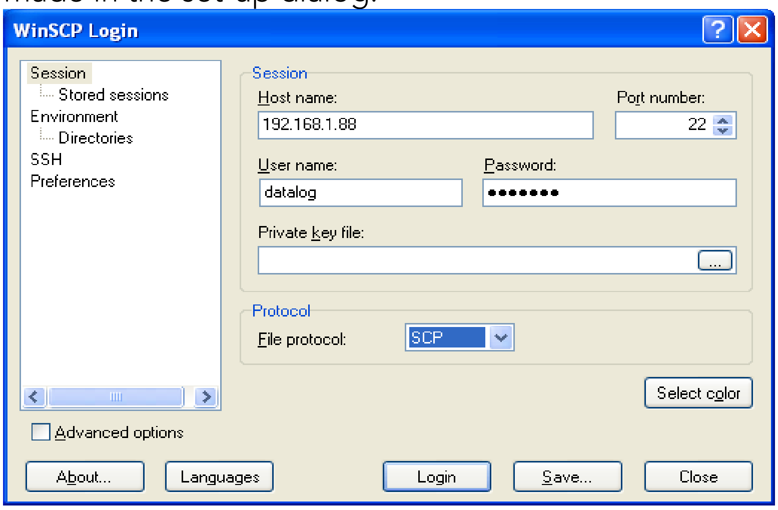
Example:
datalog.cfg_20080114_181327 – read on 14.01.2008 at 18:13:27h

This not only prevents accidentally re-importing an old configuration, but also records in the logger the exact time of transfer.

4.5 Data Transfer via LAN

4.5.1 Establishing LAN Connection to the Logger (via Display Connector)

It is possible to go through the display jack to access the logger via LAN. The required cable is sold separately. In order to establish connection to the logger, one needs a tool supported by the SCP log. We recommend the OpenSource tools "WinSCP" (to download the current version, go to http://winscp.net/eng/download.php).
Once you have Ethernet connection to the computer – whose permanently set IP must be located in the same segment as the logger (192.168.1.xxx) – and the tool has been installed and started, the following specifications are to be made in the set-up dialog:



Illus. 4.5.1-1 Parameter dialog screen for establishing logger connection via WINSCP

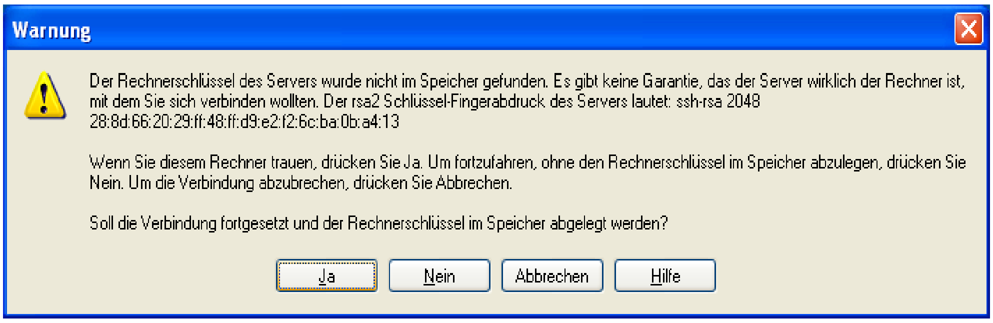
(The password is identical to the user name = "datalog").

The most important settings of the dialog are:

- host name = 192.168.1.88
- user name = datalog
- password = datalog
- file protocol = SCP

The button “save” allows you to save the settings for later use.

After the first start, the following warning is displayed:



This dialog can be ended with “yes” (in which case it will not appear again).

4.5.2 Transferring Configurations via Ethernet

Unlike data transfer via USB, WLAN or UMTS, with transfer via Ethernet, data is always transmitted during operation.
The file “datalog.cfg” must be copied into the directory “<root>\home\datalog”. The measured data is stored relative to this directory. For details on the structure of the measured data, see Chapter 4.6 File Structure – Storing Measurement Data on the Logger.

When measurement is started, the configuration file is copied into a buffer directory (cache) and deleted from the “datalog” directory.

Note The new configuration is not used until measurement is started. Importing a configuration during running measurement has no influence on this measurement. Upon logger shutdown, the new settings are accepted, and the interfaces reconfigured accordingly. The new configuration is applied upon the next logger start.

Note This path cannot be used for firmware updates. These must always be performed via regular transfer path (USB, WLAN or UMTS).

4.5.3 Transferring Data via LAN

Once connection to the logger has been established, in one of the windows the standard data directory of the logger is selected: “<root>\home\datalog”.

This directory contains subdirectories, each in turn containing a completed measurement. (For details on the data structure, see Chapter 4.6 File Structure – Storing Measurement Data on the Logger). To transfer a measurement, you must copy the entire subdirectory. If so configured, you will also find in the directory the respective log file (with a file name identical to that of the data subdirectory), which also belongs to this measurement.

Note

Access to the logger is only possible once the logger is started. This means, though measurement is running, the current measurement is kept in the cache and cannot be seen at the time of measurement in the directory "datalog".

4.5.4 Releasing Connection

There are several ways to release an established connection:

- through the key combination "Shift + Ctrl + D",
- via the menu item "Session" -> "Release connection", or
- by directly closing the program window.

4.6 File Structure – Storing Measurement Data on the Logger

The base directory for the file structure is the logger directory where the datalog.cfg file is stored: "<root>\home\datalog".

Once a measurement is completed, the data is moved from the data buffer (cache) into a new data directory relative to the base directory "datalog". The name of this data directory is made up of the prefix (see: dataLog configuration) followed by the date, time and track name.

This directory is where the logger stores the CFG file used for the measurement, as well as the ATFX file containing the description (header) of the stored data files, and the start/stop list (*.SSL) and the minimum/maximum list (*.MML). The actual measured data is located in the "data" subdirectory. A separate file is created for each method. The name is the name of the method from the configuration. The corresponding log file is stored parallel to the directory.

[1.3] This standard structure can be altered with the appropriate settings in the configuration. If no subdirectory "data" is used, all the files from a measurement are located directly in the data directory.

5 Service

5.1 Power Outage

In the event of a power interruption during a running measurement, the logger-operating system is designed to ensure that the logger always properly restarts.

Since the data is buffered prior to being permanently stored in the file system, buffer loss due to power outage has the following consequences:

- Classing results are stored at 60-second intervals, i.e. up to 60 seconds of classing results can be lost.
- Time logs are directly stored. The operating system uses only a small writing buffer for this purpose. Any recording losses are within the seconds range.
- The ATFX header file is not written, since the measurement was not completed. This is done later, once the logger is restarted and the measurement properly ended.

In the event that a logger becomes inoperable, due to an accident during operation, for example, CAETEC can attempt to reconstruct the ATFX header, providing the card is still usable.

5.2 Fallback-behavior

For emergencies the logger has a fallback configuration. This is fixed on the manufacturer side and does not include bus signals or logging methods. It is activated if the logger has not been correctly shutdown at the first 2 minutes of the last run. This can be caused by a faulty configuration that crashes the firmware, by a hardware defect or by the user who wants to put the system in a secure state (by disconnecting from the power supply), e.g. To enter a new configuration.

A modified configuration can be imported via USB. No user input is necessary. However, the changed configuration must be located on the USB stick in the subdirectory / fallbackcfg / <frontnumber>. This transmission path is independent of the configured data transmission. No measurement data are transmitted to the stick.

5.3 Firmware Updates

The basic firmware update procedure is the same as for importing a new CFG file into the logger. The update can be performed via WLAN, UMTS or USB. The import of an update packet can coincide with that of a new configuration (.cfg).

With logger firmware version 1.0.0 and higher, the higher version can generally be imported without dealing with intermediate versions. To update lower versions, please contact CAETEC Support.
The name of the file containing the firmware update must be "dlupdate.dlua"; with firmware version 1.3.0 and above, only the attachment .dlua is mandatory, while the preceding name can be freely selected (e.g., "dL_1.3.1.dlua").
With this mechanism, downgrading from a high version number to a lower one is generally not possible.

If an update fails, the logger continues with the last valid firmware status. An update may also include updates for the display and the various bus interfaces. If these are not connected at the time of logger update, their update is performed at the next shutdown with connected display/interfaces.

The following preparations must be made:

- The current configuration must permit an "End Track" (either by display selection or by default).
- The datalog.key must be located in the root directory of the USB medium.
- A new firmware version must be copied into the directory where the logger also accepts CFG files.
- The valid configuration should be running (updating from an error situation should be avoided).

Update procedure:

- Switch on the logger and let it boot until it is in measurement operation.
- For update via USB: Connect the USB medium and wait until it is accepted (until the activity LED first lights up takes approx. 10 seconds.)
- Start data transfer (e.g., by shutting down the logger, depending on the configuration).
- During the data transfer process, the logger checks for a firmware update file. If a valid one is found, update begins. This is indicated in the display by the words "in progress". If the display is also updated, then the background illumination is extinguished for the duration of the procedure. The procedure can take up to one to two minutes; the typical duration is < 30 s. Once the update process is completed, the logger restarts.

Check

- After the update flash, the firmware version can be checked either via the display or the Web interface.
- After the flash, the extension of the firmware file is appended with the timestamp of the update.

5.4 Error Codes, Problems and Suggested Remedies

Error codes, signified by the combination "letter - number", help identify the error:

Letter:	i = initialization t = transfer
Number:	Consecutive number – with a specific meaning described in the following tables. (The numbers do not always have the same meaning for WLAN, USB and PPP).

5.4.1 Error Codes - WLAN

Error number	Description
i-1	Internal error
i-2	WLAN hardware not recognized
i-3	Internal error
i-4	No WLAN connection possible
i-5	Internal error
i-6	Internal error
i-7	Internal error
i-8	Error in dynamic allocation of IP address
i-9	Error in setting the configured static IP address

Table 5.4.1-1 WLAN initialization errors.

Error number	Description
i-11	Internal error
i-12	Error in TCP Connect (Note: Do not use subnet 192.168.1.xxx)
i-13	Error in initializing an SSH session
i-14	Error in opening an SSH session
i-15	SSH server provides no usable authentication method
i-16	SSH server authentication attempt failed (e.g., incorrect password)

Table 5.4.1-2 WLAN SSH connection errors.

Error	Description
t-11	Error in opening the internal source file
t-12	Error in creating the target file on the SSH server (no SCP log support on server, path does not exist, storage medium full, ...)
t-21	Error during transfer of target file to SSH server (storage medium full full, ...)

Table 5.4.1-3 WLAN data transfer errors

5.4.2 Error Codes - USB

Error number	Description
i-1	USB storage medium (stick/disk/partition) not found
i-2	USB storage medium not authorized

Table 5.4.2-1 USB initialization errors

Error number	Description
t-11	Error in opening the internal source file
t-12	Error creating the target file on the USB stick (path does not exist, data storage medium full, …)
t-21	Error transferring the target file to the USB stick (data storage medium full, …)

Table 5.4.2-2 USB data transfer errors

5.4.3 Error Codes - PPP

Error number	Description
i-3	Error in section <ppp> <peer> of the cfg
i-4	Modem not found
i-9	Connection failed

Table 5.4.3-1 PPP initialization errors

Error number	Description
i-11	Internal error
i-12	Error during TCP Connect (Note: Do not use sub-net 192.168.1.xxx)
i-13	Error in initialization of an SSH session
i-14	Error in opening an SSH session
i-15	SSH server provides no usable authentication method
i-16	SSH server authentication attempt failed (e.g., incorrect password)

Table 5.4.3-2 PPP SSH connection errors

Error number	Description
t-11	Error in opening the internal source file
t-12	Error in creating the target file on SSH server (no SCP log support on server, path does not exist, data storage medium full, ...)
t-21	Error in transferring target file to SSH server (data storage medium full, ...)

Table 5.4.3-3 PPP data transfer errors

5.4.4 Error Codes XCP LAPI

Error number	Description
Errors (hexadecimal) upon initialization have the (right-most) bit with index 0 as the LSB, and the indexes linearly ascending (to the left) to the MSB. Error code is written to the log file, behind the string "Parameterization failed with error code".	
Bit 0 is set.	SET_MTA or DOWNLOAD of data failed
Bit 1 is set.	USER_CMD (active) failed
Bit 2 is set.	USER_CMD (deactivated) failed
Bit 3 is set.	CONNECT failed
Bit 4 is set.	DISCONNECT failed

Table 5.4.4-1 XCP LAPI parameterization errors.

5.4.5 Problems and Suggested Remedies

Problem	Possible Remedy
Logger starts unexpectedly.	The input wire for start on preset voltage (Clamp 15) has a very high impedance. Ambient disturbance can start the logger. Proper wiring of the potential on the pin is recommended.
Signals on the CAN channel, but green LED is not lit.	Wrong baud rate of CAN signals. Check setting.
No signals in the display, only the start screen, with three dots.	No signals configured for output to display. Check configuration for errors (upper/lower-casing or omissions) Old display firmware (lower than 1.05) used with new logger firmware (1.3.x and higher). Unplug and re-plug display, update firmware.

Table 5.4.5-1 Problems with suggested remedies

6 List of Changes

1.3, 21.03.2011, Robin Fink

- Reformatting and revision.
- Added: Hardware CAN V2, LIN V2.
- Update, to firmware 1.3 (added/revised chapters: Hardware: FlexRay, Analog/Digital I/O, UMTS; configuration: FlexRay, Analog/Digital signal, virtual CAN, transfer events, data transfer via PPP, data set generation, event files, system state output, scripts, multi-level triggers, data formats (blf, mdf4), persistence, data transfer, classing methods (open border classes), Gateway, language, comments)
- Added: µCROS.

1.3, 15.06.2011, Robin Fink

- Sections restructured; added chapter: Web interface.
- Added chapter: Starting / stopping the logger.
- Added: Display trigger persistence.
- Revised: Illustrations.
- New list of changes.
- Added: Blink codes GPS, UMTS interface.
- Reformatted: parameters.

1.3, 22.09.2011, Robin Fink, Lonnie Legg

First release of English version

1.4, 09.09.2013, Stefan Prousa

- Changed description of display trigger in 3.1.3

1.4, 09.04.2014, Stefan Prousa

- PPP error code i-4 added

1.4, 23.04.2014, Stefan Prousa

- Table 3.3.5-1 (The bus statistics subtype parameters) added missing subtype "state"

1.4, 27.08.2014, Philipp Schmid

- Transfered Configuration Part to Phönix-Manual