IPETRONIK

IPEMOTION Software

IPEmotion_PlugIn_X_V02_17_00

15. August 2022

Table of Contents

1 Important and general information . . . 4
1.1 Important information . . . 4
1.1.1 Safety and Warning instructions . . . 4
1.2 Terms and conditions . . . 5
1.2.1 Legend of used icons . . . 5
1.2.2 Support . . . 5

2 PlugIn overview . . . 6
2.1 PlugIn description . . . 6
2.2 PlugIn installation . . . 6
2.3 System overview . . . 7

3 CAN Modules hardware setups . . . 8
3.1 Example 1 . . . 9
3.2 Example 2 . . . 9
3.3 Example 3 . . . 10

4 X Modules hardware setups . . . 11
4.1 Example 1 . . . 11
4.2 Example 2 . . . 11
4.3 Example 3 - CAN Tunneling . . . 12

5 SIGNALS work space . . . 13
5.1 Ribbon main functions . . . 13
5.1.1 Hardware / PlugIn . . . 14
5.1.2 System . . . 14
5.1.3 Firmware Update . . . 15
5.1.4 Configuration check . . . 18
5.2 Adjust functions . . . 20
5.2.1 Database . . . 20
5.2.2 TEDS . . . 20
5.2.3 Offset adjust . . . 24
5.2.4 Shunt Check . . . 26
5.3 Detect . . . 29
5.3.1 Mapping . . . 30
5.3.2 Synchronize . . . 32
5.4 Initialize . . . 33
5.4.1 Reset . . . 34
5.4.2 Display and Quick Analyzer . . . 35
5.4.3 Details . . . 38
5.5 System Tree . . . 39
5.5.1 Column chooser . . . 39
5.5.2 Context menu for system, modules and channels . . . 41

6 Interface configuration . . . 45
6.1 Interface – General tab sheet . . . 45
6.2 Ethernet hardware tab sheet . . . 46
6.3 CAN hardware tab sheet . . . 47
6.4 Options tab sheet . . . 48
6.5 CAN- and X-module configuration . . . 49
6.5.1 General tab sheet . . . 50
6.5.2 Extended tab sheet – CAN modules . . . 51
6.5.3 Extended tab sheet – X modules . . . 52
6.5.4 Information tab sheet . . . 52
6.5.5 Module license update . . . 54
6.6 Channel configuration . . . 55
6.6.1 Column chooser in the channel grid . . . 55
6.6.2 General tab sheet . . . 57
6.6.3 Defining list box entries of channel names . . . 57
6.6.4 Format tab sheet . . . 58

6.7 Channel Scaling – defining ranges and engineering units . . . 65
6.7.1 Sensor mode . . . 65
6.7.2 Sensor range . . . 65
6.7.3 Physical range – Engineering units . . . 67
6.8 Scaling calculator – for advanced scaling functions . . . 67
6.8.1 2-point scaling . . . 69
6.8.2 Free 2-point scaling . . . 70
6.8.3 Factor/Offset scaling . . . 71
6.8.4 Multipoint scaling . . . 72
6.8.5 STG Strain gauge . . . 73
6.8.6 VTAB range . . . 74
6.8.7 VTAB . . . 75
6.8.8 Active Sensors . . . 76
6.8.9 Passive Sensors . . . 77
6.8.10 Snapshot – Test Measurement . . . 77
6.9 Sensor database in the scaling calculator . . . 78
6.9.1 Adding new Sensors - Sensor Database Editor . . . 78
6.9.2 The database format . . . 80
6.9.3 Multipoint linearization . . . 83
6.9.4 Adding new Sensors – through the Scaling Calculator . . . 86
6.10 Display tab sheet . . . 88
6.10.1 Define standard decimal templates on module level . . . 88
6.10.2 Output tab sheet for output channel . . . 91
6.10.3 Channel-specific tab sheets . . . 91

7 X-PlugIn OPTIONS . . . 92
7.1 Ethernet interface . . . 92
7.2 CAN interface . . . 94
7.3 Options . . . 95
7.4 Components . . . 98
7.5 CAN card hardware interfaces . . . 99

1 Important and general information

1.1 Important information

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

1.1.1 Safety and Warning instructions

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

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

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

   Please check the following points to avoid errors:

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

The products can be operated in extended temperature ranges greater $70\,^{\circ}\mathrm{C}$ and therefore the operator has to take safety measures to avoid any skin burnings on hot surfaces while touching the products.

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

1.2 Terms and conditions

See IPETRONIK website for details: www.ipetronik.com

1.2.1 Legend of used icons

	Tip	*This icon indicates a useful tip that facilitates the application of the software.*
	Information	*This icon indicates additional information for a better understanding.*
	Attention!	*This icon indicates important information to avoid potential error messages.*

1.2.2 Support

Headquarter:

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

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

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

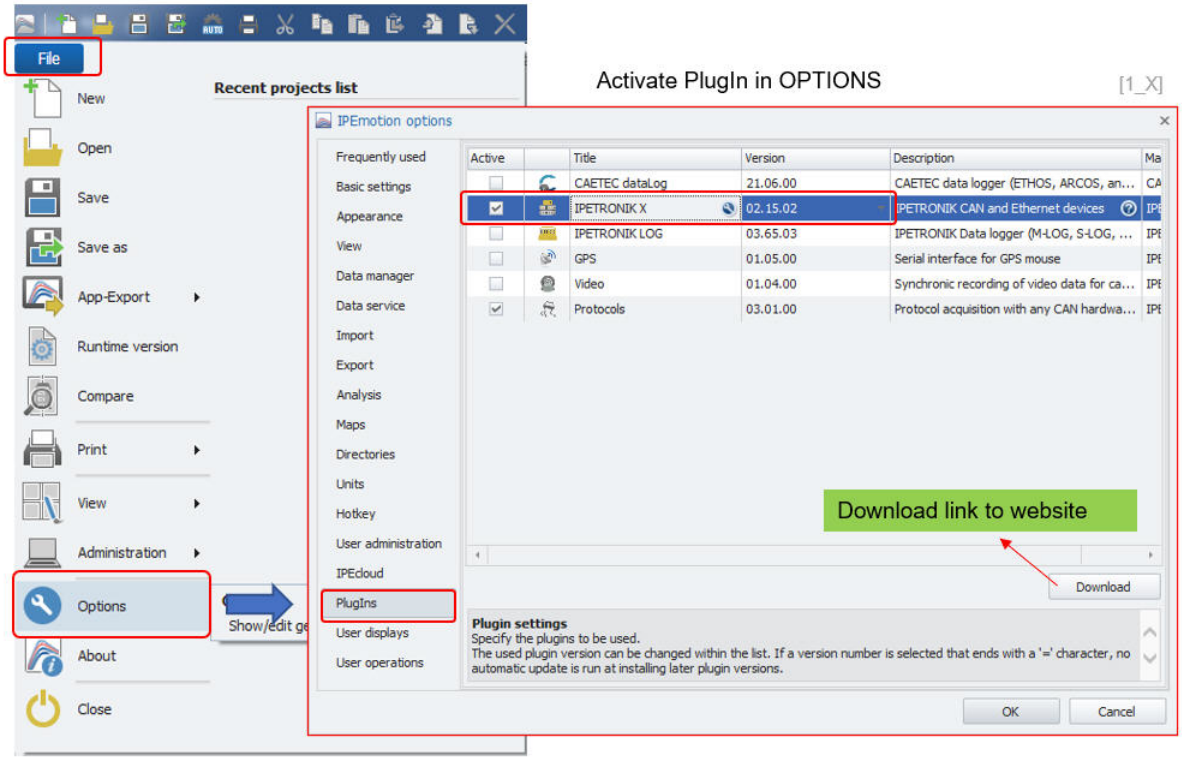
2 PlugIn overview

2.1 PlugIn description

With the IPETRONIK-X PlugIn you can configure analog and digital measurement modules from IPETRONIK. The two main module lines are CAN modules and Ethernet Modules. Therefore 2 configuration interfaces CAN and Ethernet are supported for the product lines. However, it is also possible to daisy chain CAN modules after the X-Modules. This setup is called X-LINK and will be explained too. The Modules can be operated with the IPEmotion PC software and on the IPEmotionRT data logger software.

2.2 PlugIn installation

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

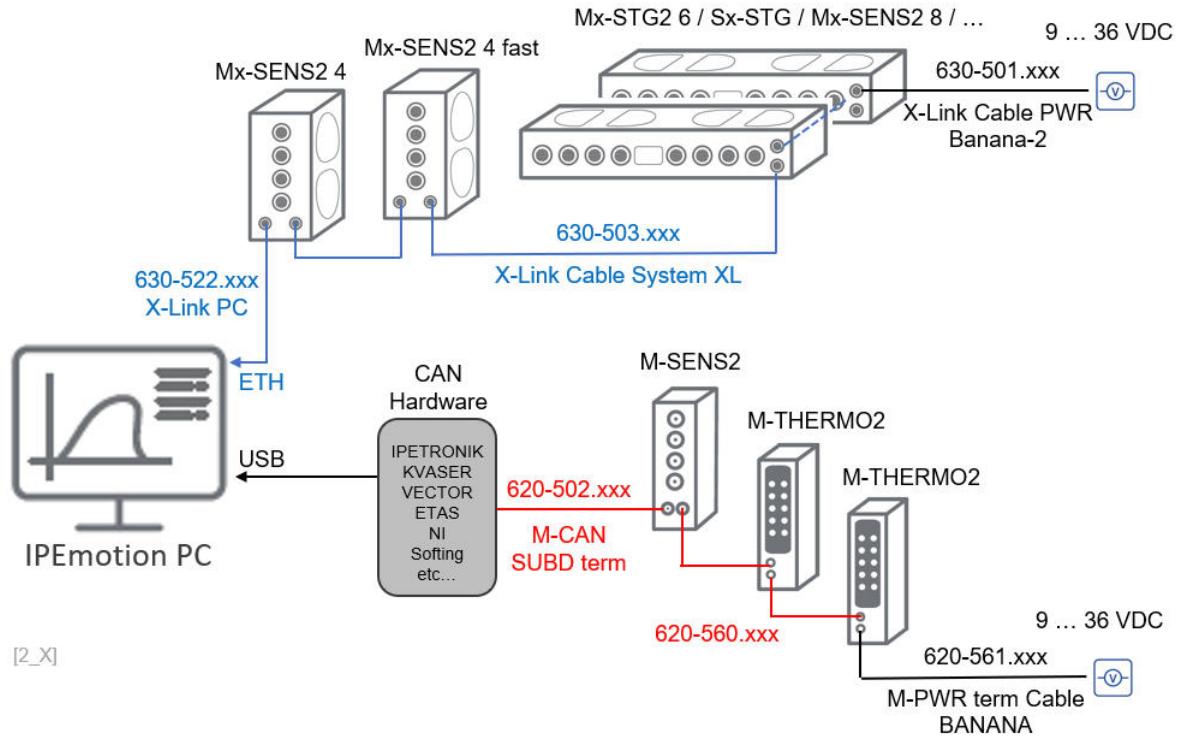


The PlugIn is supporting the following Windows operating systems:

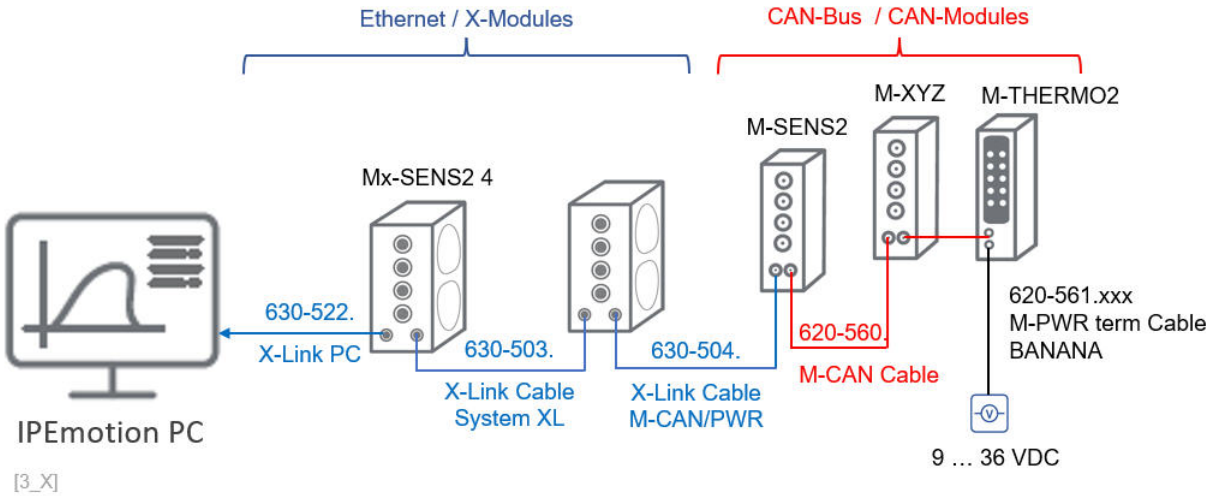
- 32 bit
- 64 bit

2.3 System overview

The following diagram is indicating the two main systema architectures. You can operate the X-Modules based on an Ethernet communication via your LAN port of your computer. The other setup requires a CAN card interface from IPETRONIK like IPEcanPro FD or IPEhub2 or other supported vendors for the CAN modules.



Another system setup is to combine Ethernet and CAN modules in one days chain. This setup requires that the Ethernet modules are connected first to the PC and the CAN modules are following the Ethernet modules.



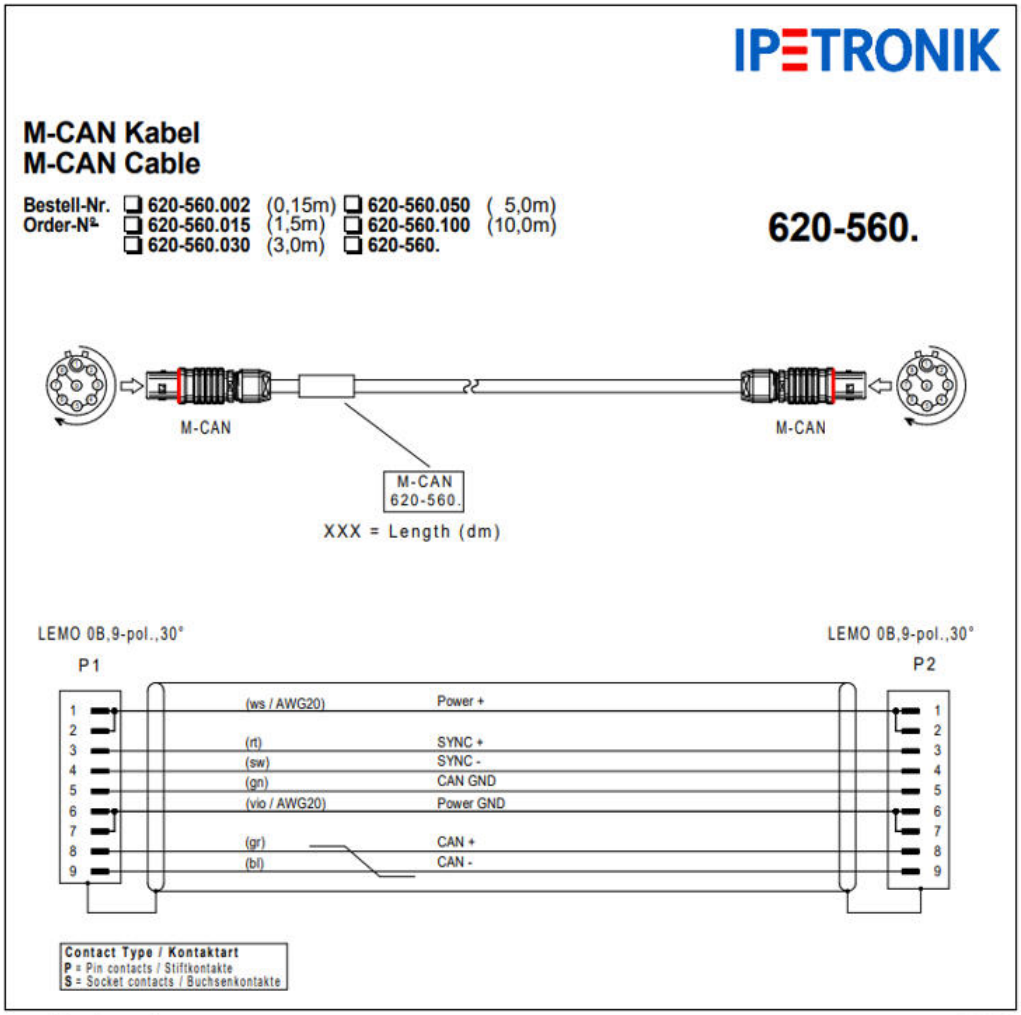
The most common and recommended hardware setups and the required cable sets will be explained in the following sections.

3 CAN Modules hardware setups

In the following the 3 main setups are explained. The required cable sets and lengths are depending on the physical installation environment. The cables are available in different lengths. The last 3 digits of the cable number are indicating the length. The required cable to interconnect the M-CAN modules is number: 620-560.xxx. The available lengths for the placeholder .xxx are for example:

- 002 = 15 centimeter
- 015 = 1.5 meter
- 030 = 3 meter
- 050 = 5 meter
- 100 = 10 meter

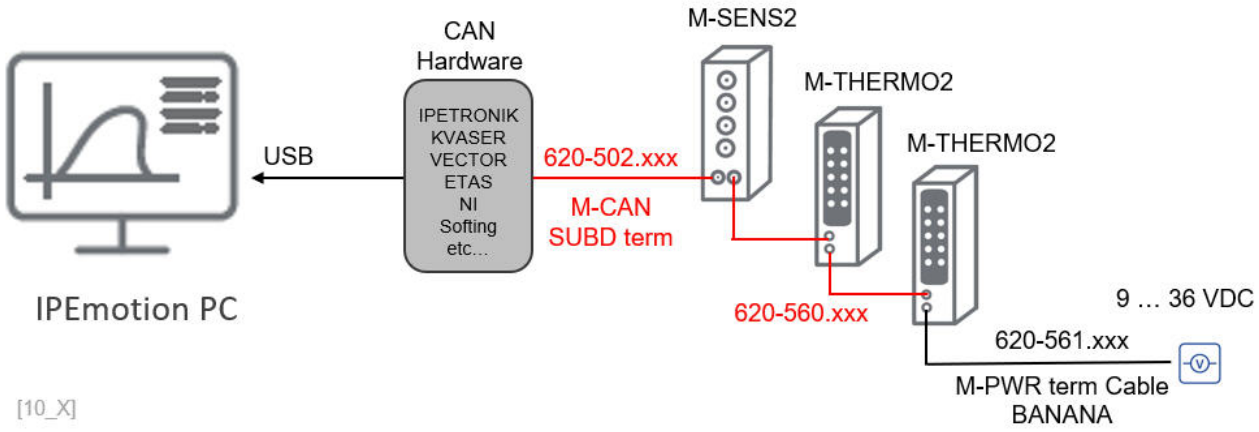
Every cable has a dedicated cable data sheet indicating the connectors and the cable pins and the color of the cable as indicated in the example below. .



Cable data sheet [9_X]

3.1 Example 1

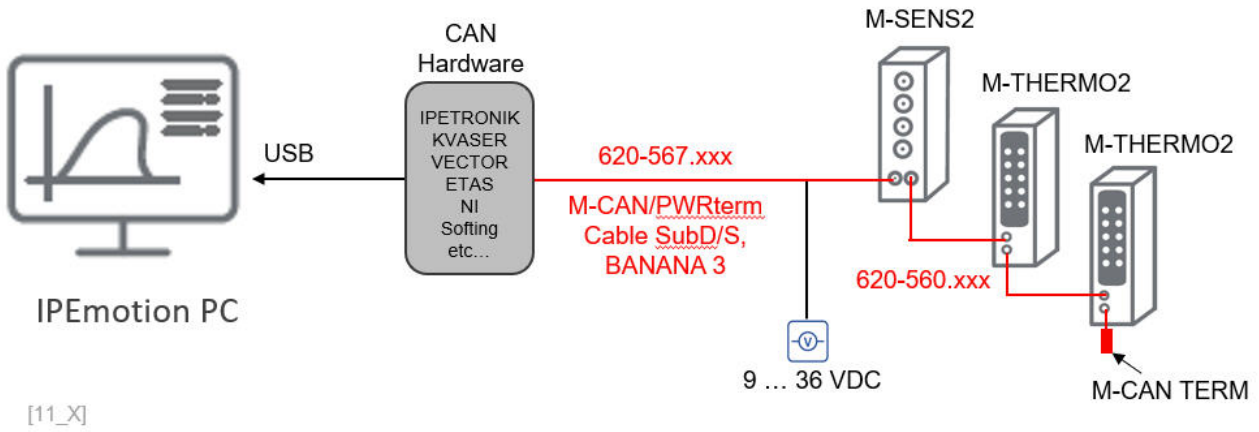
In this setup the power supply is provided from the very end of the module chain. This is applicable when only a few modules are in the measurement setup and one source of supply 9-36 VDC is sufficient.



[10_X]

3.2 Example 2

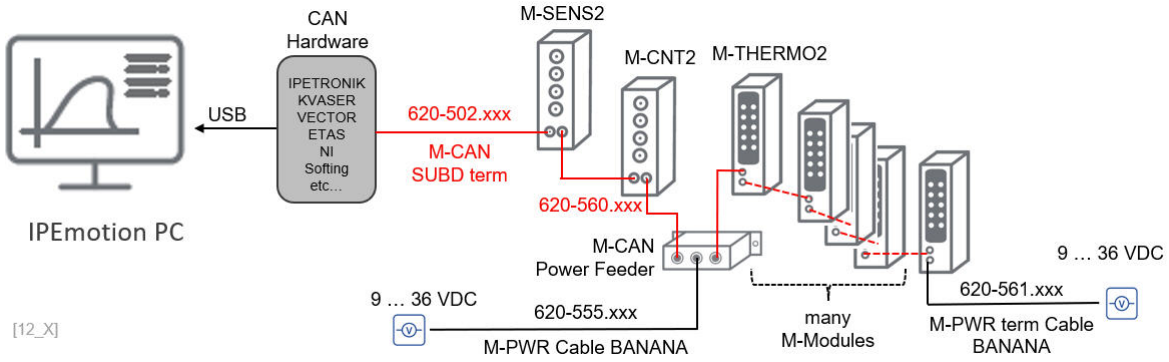
In this setup the supply is provided through a SUBD 9 and Y-splitter cable at the beginning of the module chain. This setup is also practical in the case the power supply and CAN interface are located at the same end. This system works well for smaller mule chains where one supply is sufficient. It is important to finish the measurement system on the last module with a CAN bus termination plug.



[11_X]

3.3 Example 3

In this case you operate many modules in our measurement setup and the interconnection between the different modules might be also large which causes voltage drops along the modules. It is recommended to add a power feeder T-Junction to the system. Within very large systems is many be required to have several power feeder and to use additional power supply via the last module or the first modules, as indicated in the two scenarios above. However, it is important to consider a separate power supply cable when using the power feeder. This cable has no internal CAN bus termination. As a rule of thumb every 15 modules a power feeder should be considered.

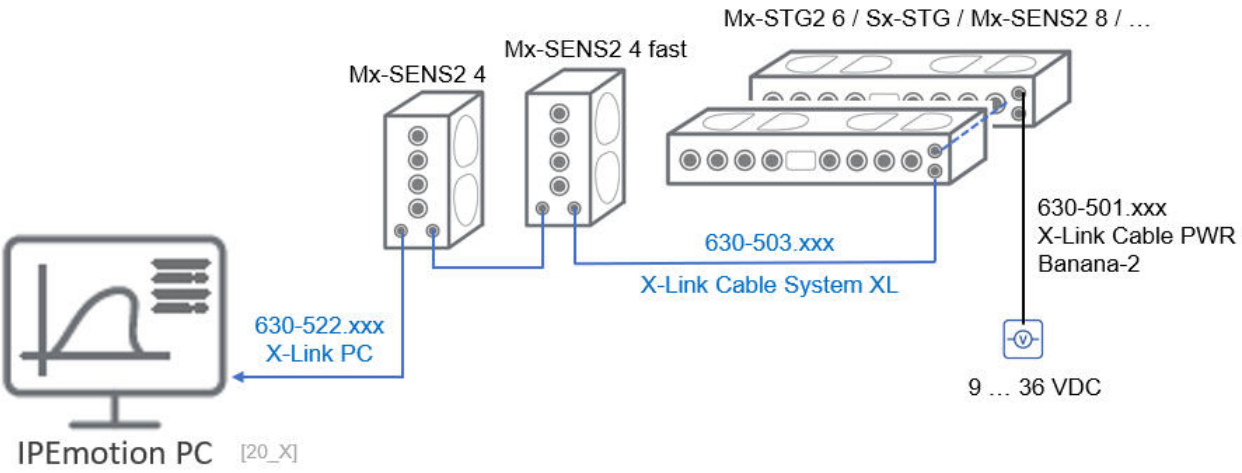


4 X Modules hardware setups

In the following 4 main hardware configurations and cable sets are presented.

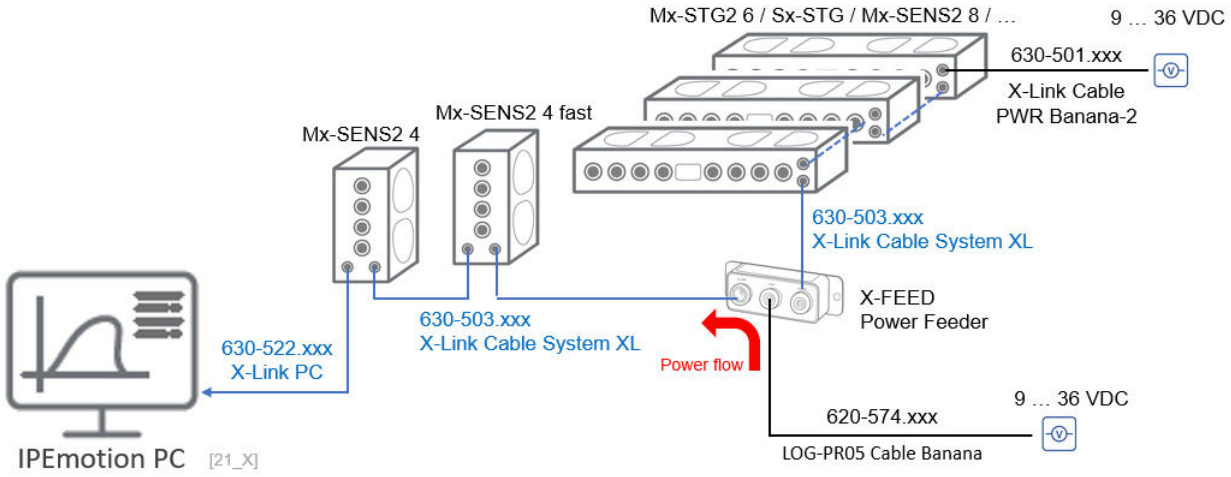
4.1 Example 1

In this setup the power supply is provided from the very end of the module chain. This is applicable when only a few modules are in the measurement setup and one source of supply 9-36 VDC is sufficient.



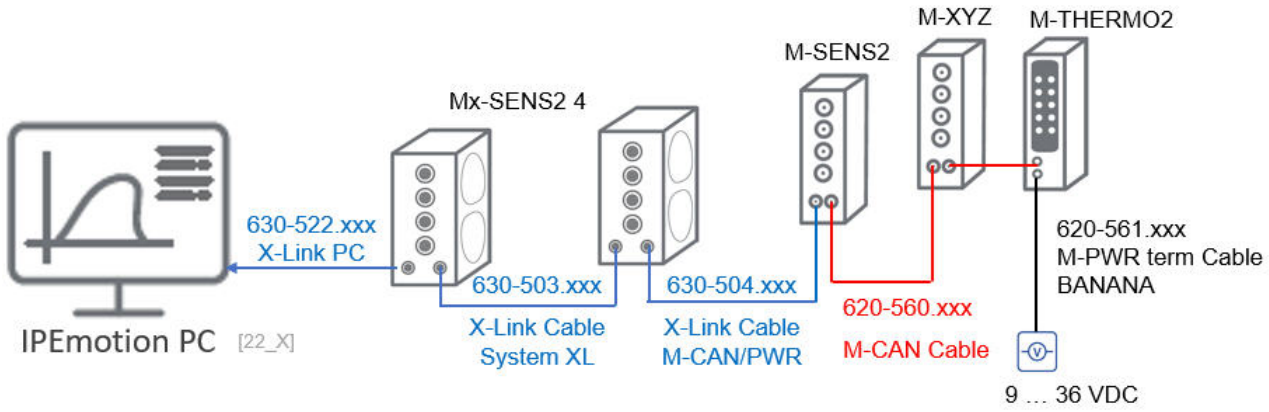
4.2 Example 2

In this setup many modules are involved and therefore intermediate power supply is needed. With the X-FEED power feeder the modules can get power feed into the middle of the measurement chain. As a rule of thumb 7 X-Modules can be supplied with one power feeder. If the system grows lager power supply from the very end or additional X-FEED modules can added to the system. As indicated the X-FEED provides power only to the X-Modles. The CAN-Modules require their own power supply either from the very end or using the M-CAN power feeder.



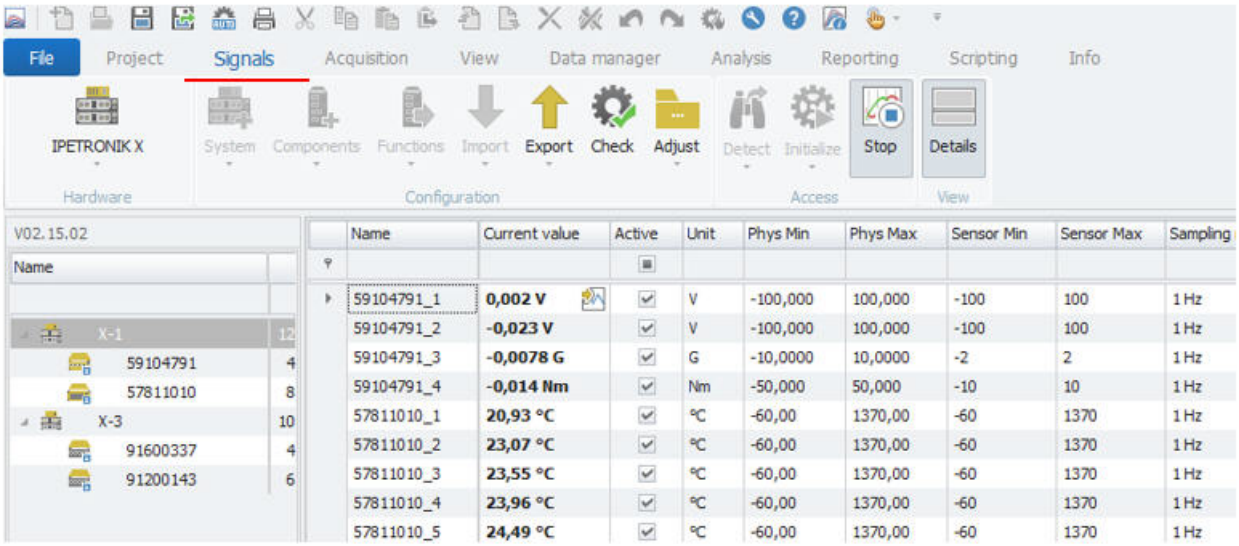
4.3 Example 3 - CAN Tunneling

Another system architecture can combine X- and CAN-Modules in one daisy chain. In this case a dedicated cable is required to link-up the Ethernet based X-Modules to the CAN based M-Modules. The architecture requires that the Ethernet modules come first and that the CAN modules are attached behind. It is not possible to add any Ethernet modules behind the CAN modules. In smaller setups one power feed from the very end can be sufficient. However, if the system grows lager, you can extend the power alimentation through adding X-FEED modules for the X-Modules and CAN POWER FEEDER to the CAN modules.



5 SIGNALS work space

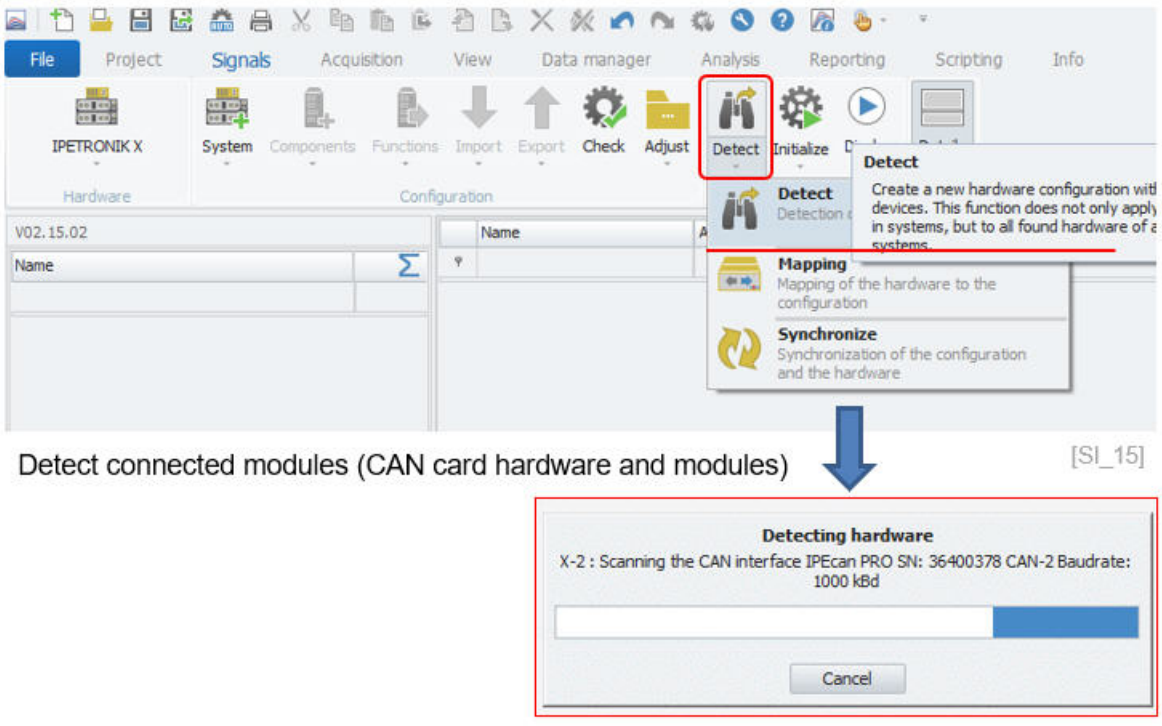
The SIGNALS work space is dedicated to configure your PlugIns and take measurements. All configuration functions are explained in reference to the IPETRONIK X PlugIn.



Example: IPETRONIK-X PlugIn [SI_1]

5.1 Ribbon main functions

When you start working with your analog measurement modules you need setup the hardware and cable sets as discussed above. A supported CAN card hardware and power supply is required. The easiest way to get started is to run the DETECT function as indicated below.



Detect connected modules (CAN card hardware and modules) [SI_15]

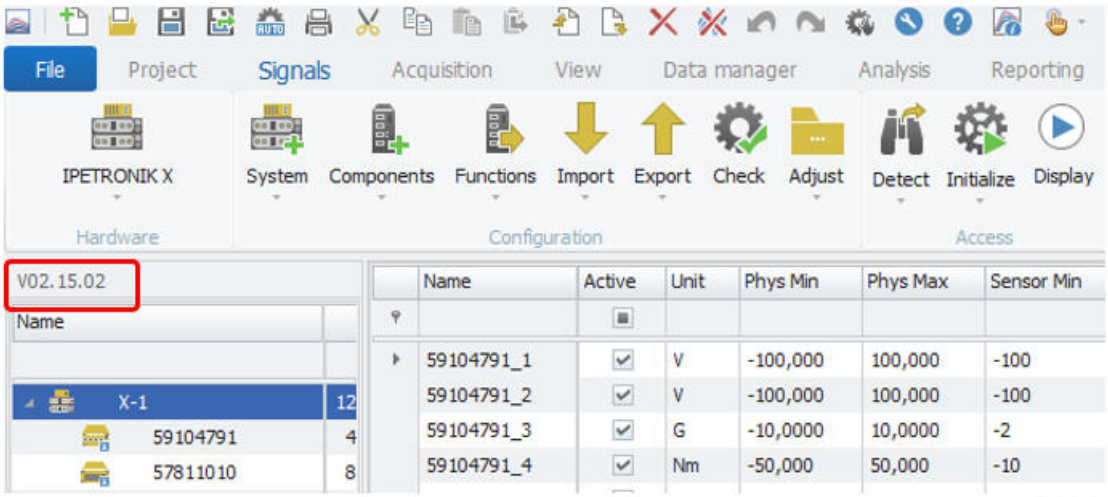
5.1.1 Hardware / PlugIn

Select the hardware / PlugIn you would like to use for your dry confoguration. The drop down list includes all PlugIns which were activated in OPTIONS >PlugIns. See chapter **??** for more details.

Tip

Sometimes users cannot access the list box and make manual configurations. In this case, in OPTIONS > Basic Settings the measurement configuration by MPC data base file was activated. See chapter ***??****.*

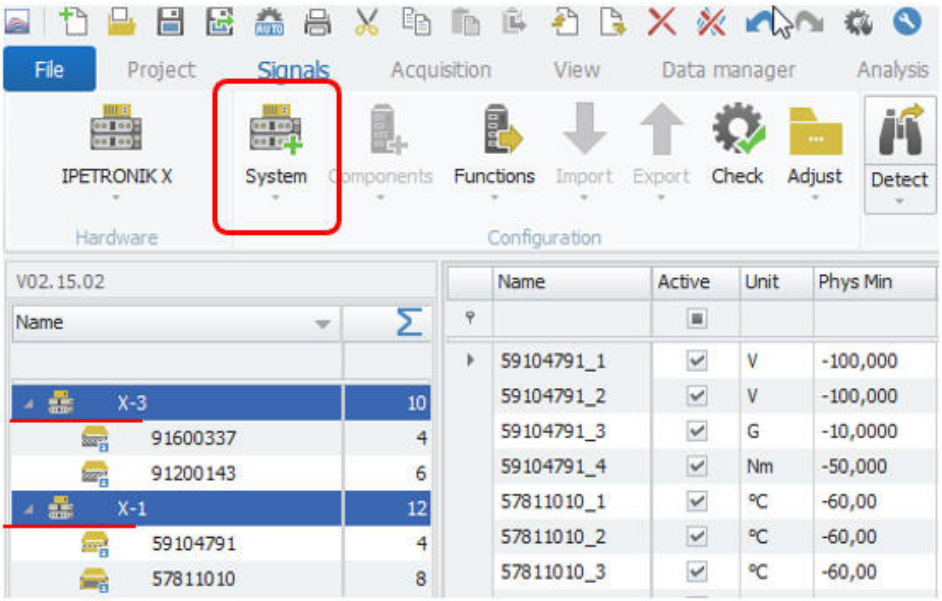
If you select a PlugIn from the active hardware list, you will see the currently loaded PlugIn version. For changing the PlugIn version you need to go back to OPTIONS >PlugIns. There you can switch to previous versions. An equal sign (=) behind the PlugIn version indicates that you will always use this version even if a more recent PlugIn version has been installed. For more details see OPTIONS >PlugIn **??**.



Indication of PlugIn version loaded [SI_2]

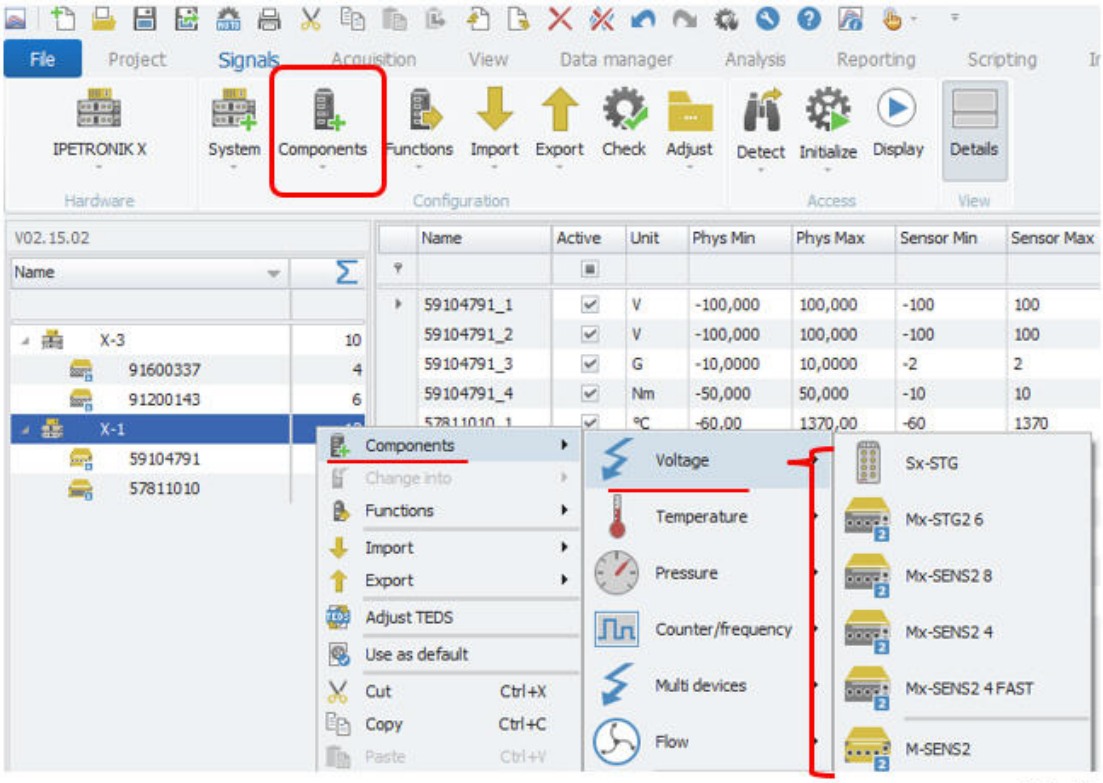
5.1.2 System

The system is the next level below the selected PlugIn. The system is basically the specific hardware or interface you are using to set up your data acquisition system. Each PlugIn consists at least of one system.



Example: 2 X-System nodes are detected [SI_3]

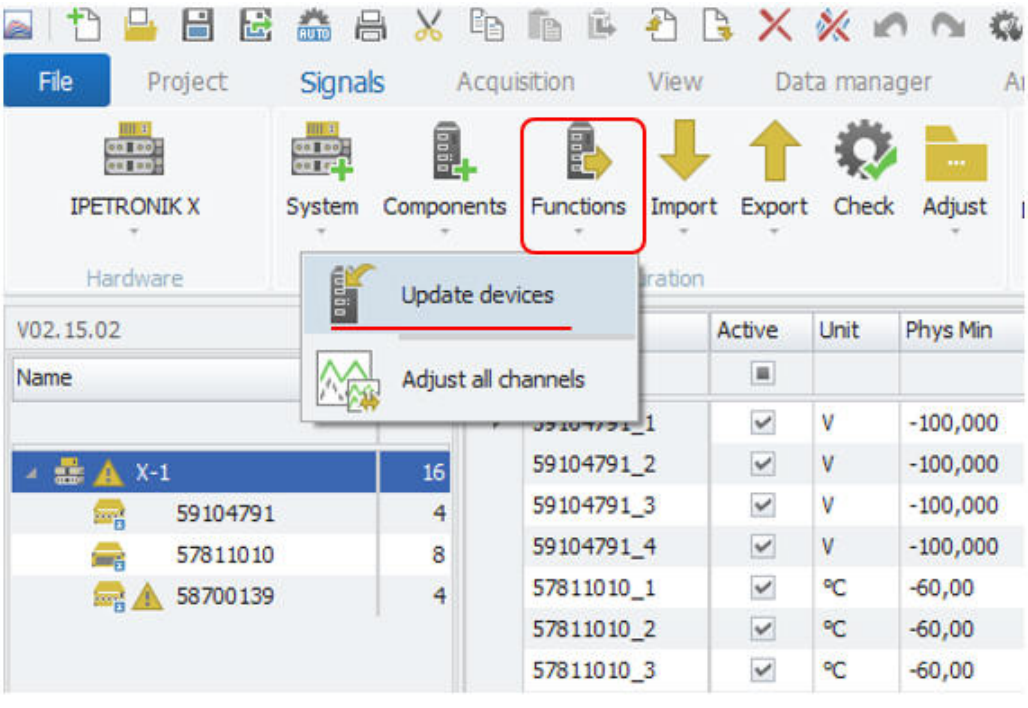
Behind the system node you can add components provided the hardware is modular. In the example the components are grouped in different categories like Voltage, Temperature, Pressure, etc. measurement modules.



Modules are grouped by components [SI_4]

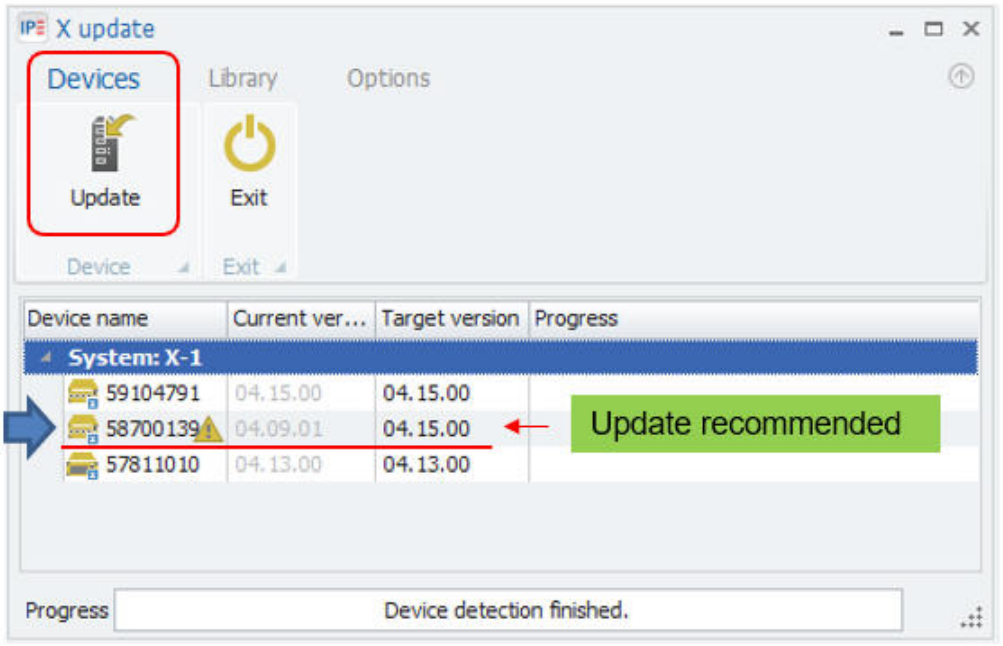
5.1.3 Firmware Update

With the firmware update function, you can update the module firmware directly from the PlugIn.



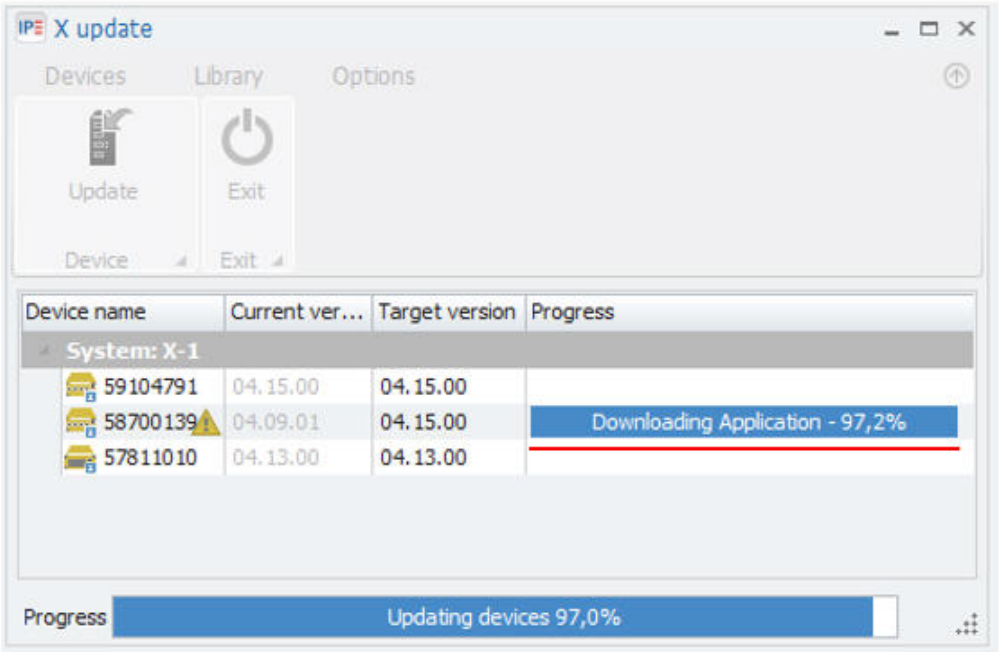
Module firmware update function [SI_4_1]

The update dialog will show you the current firmware the modules detected and will also indicate the latest firmware available on the computer. When the update process is started a progress bar will indicate the degree of completion.



Module firmware update interface [SI_4_2]

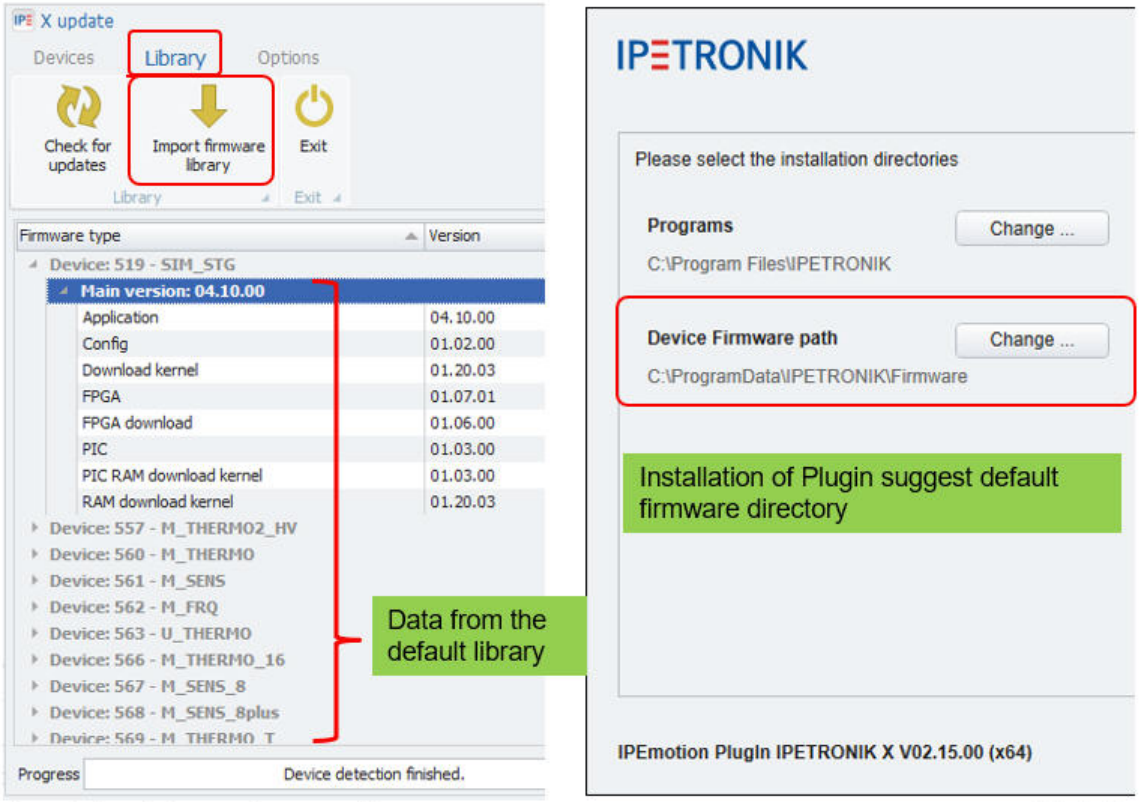
A progress bar will indicate the status of the update process.



Module firmware update progress information [SI_4_3]

In the library work sheet of X-UPDATE software you can see detailed information about the firmware latest firmware version available on the computer. The firmware data base will be installed together with the IPETRONIK-X PlugIn the following default directory:

- C: \ProgramData\IPETRONIK \Firmware



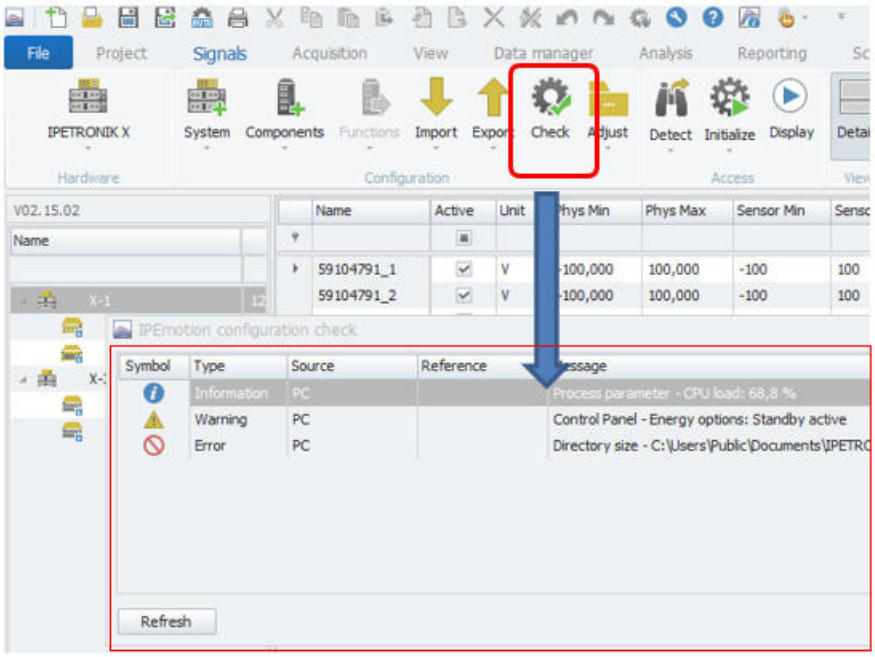
Import / update new firmware files [SI_4_4]

⚠ ***Attention!***

In the case you install an older IPETRONIK-X PlugIn version after a newer version was installed, the firmware folder will be overwritten by the latest installation. To prevent this from happening you may choose a different directory for the firmware folder of each PlugIn version. The you can import the firmware directories accordingly to your needs.

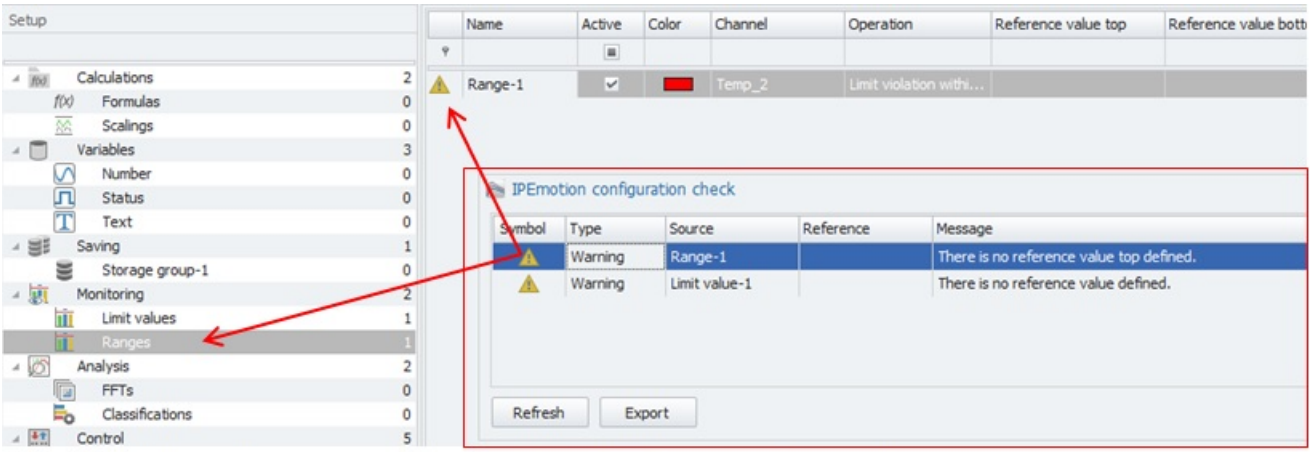
5.1.4 Configuration check

This function checks the configuration on consistency. However, this function does not work for all PlugIns. Messages are only returned if the PlugIn supports the check function. The configuration check function for example considering duplicate channel names across all active PlugIns across the SIGNALS and ACQUISITION work space. A comfortable function for message refresh and configuration error searching is implemented too.



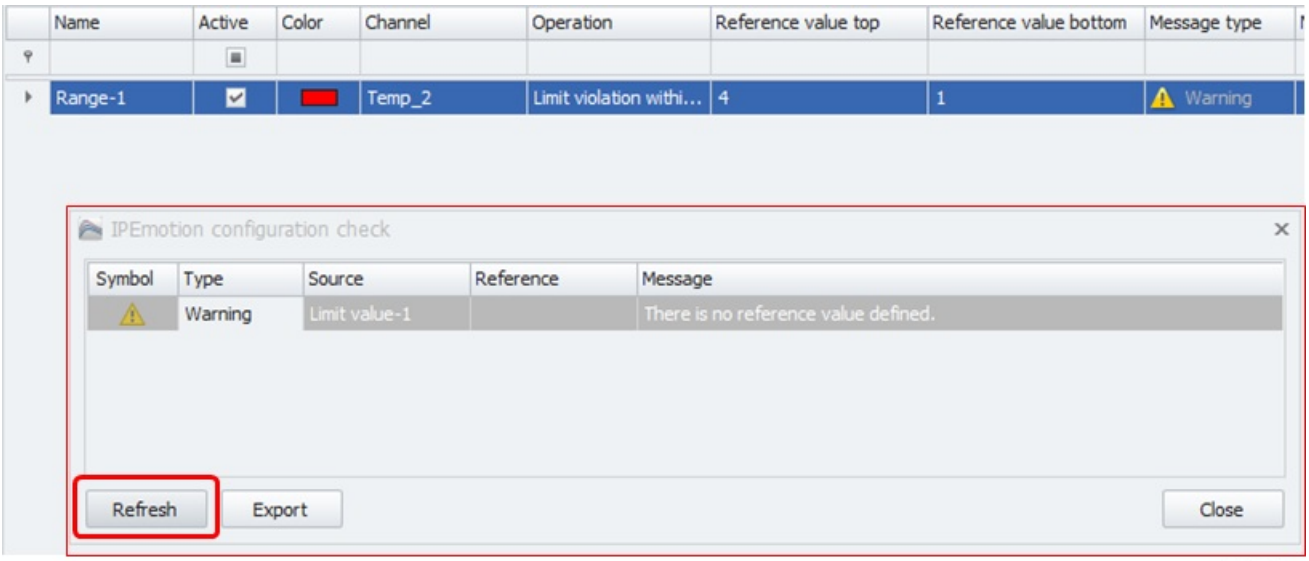
Information about possible configuration issues / problems. [SI_5]

You can directly jump to the channel in the check window for the messages as indicated in the example below of the ”Range-1” channel.



Double click on the message in the check window to reach the channel. [SI_6]

You can update the configuration and correct errors while the check window is open. With the refresh button you can update the message list.



Hit refresh button after the errors are corrected. > Message list is updated. [SI_7]

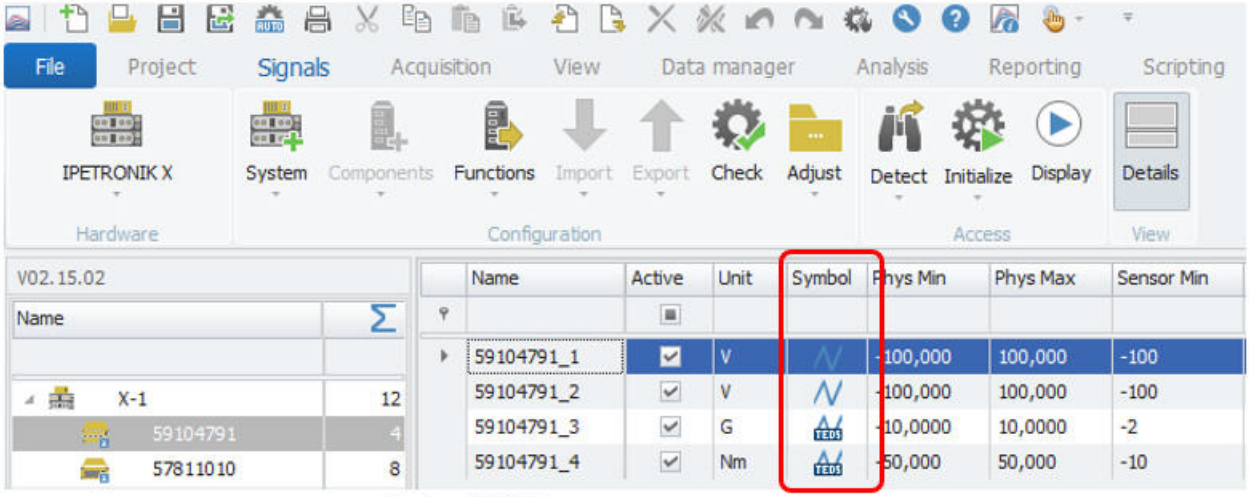
5.2 Adjust functions

5.2.1 Database

If youuse sensors from the sensor data base and run the **Adjust Database** function, the software automatically retrieves the latest sensor configuration from the data base. With this process you can automatically update all sensors with the latest calibration data from the data base in one click. Details on the sensor data base are discussed in chapter 6.9.

5.2.2 TEDS

When a hardware detection is executed and TEDS (Transducer Electronic Data Sheet)sensors are connected to the analog inputs the TEDS data stored in the TEDS chep in teh sensor are transferred to teh channel scaling. If you add the symbol clolum to the channel grid you will also see the TEDS icon.



Detect TEDS sensors. [SI_9]

The following M- and X-module modules support TEDS:

- M-SENS2
- M-SENS2 DSP
- M-SENS2 250Hz
- M-SENS2 250Hz DSP
- M-SENS 8
- M-SENS 8 DSP
- M-SENS 8plus
- M-SENS 8plus DSP
- Mx-SENS(2)8
- Mx-SENS2-4
- Mx-STG2 6
- Mx-SENS2-4 FAST

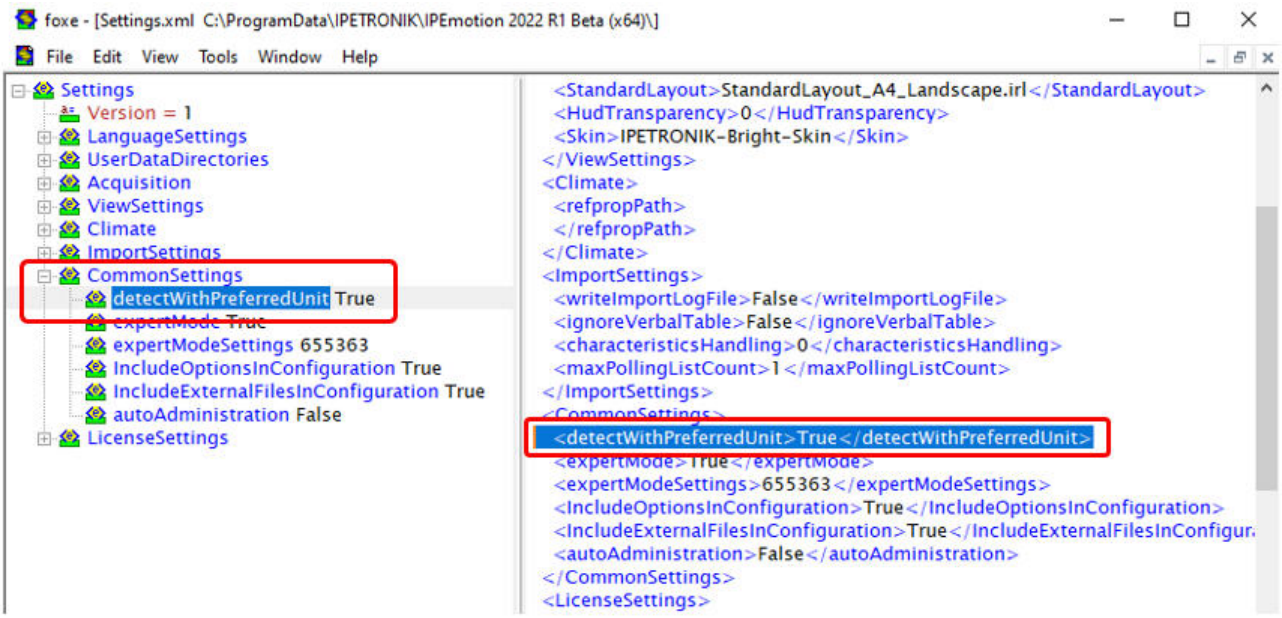
TEDS sensor detection with automatic unit transformation

When you detect TEDS sensors you can define an automatic unit conversion. This function is needed when the unit defined for the sensor does not meet the unit format required for the measurement application. In order to activate this feature you have to add an additional entry in the **Settings.XML** file. In the example above the standard sensor was detected with the unit [G].

When you add the following code into the settings.XML file the unit are automatically converted to the preferred unit defined in the OPTIONS >Unit settings.

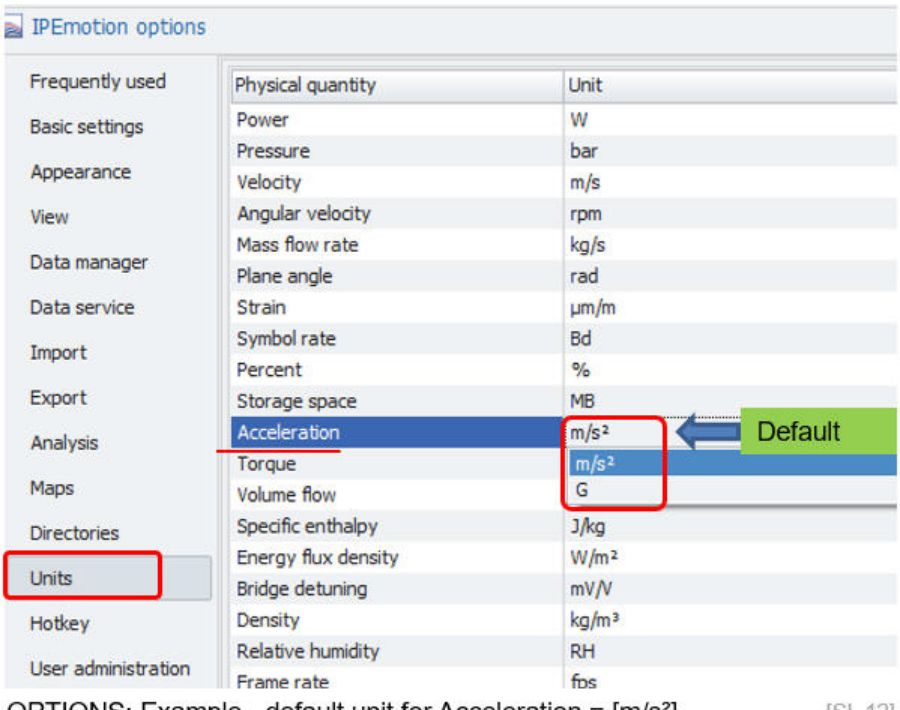
- C:\ProgramData\IPETRONIK**IPEmotion 2022 R2****Settings.XML**

The new entry in the XML file is defined as:
<detectWithPreferredUnit>**True** </detectWithPreferredUnit>



Add new function for TEDS unit conversion to Settings.xml. [SI_11]

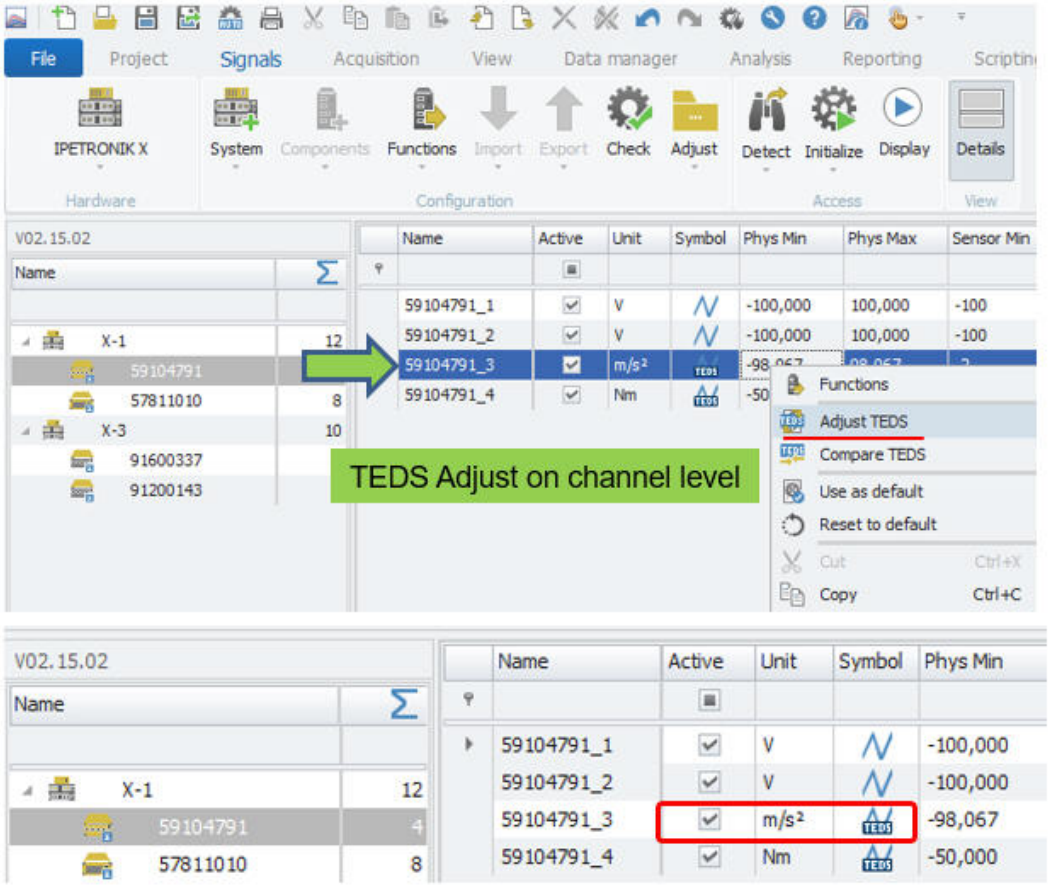
The default unit defined in the OPTIONS >Units is [m/s2].



OPTIONS: Example - default unit for Acceleration = [m/s²] [SI_12]

TEDS adjust on channel level

All modules supporting the TEDS (Transducer Electronic Data Sheets) function support in the IPETRONIK X-PlugIn a TEDS adjustment on channel level. Rather than synchronizing your whole configuration across all TEDS channels you can focus on a dedicated channel to integrate the TEDS data from a connected sensor.

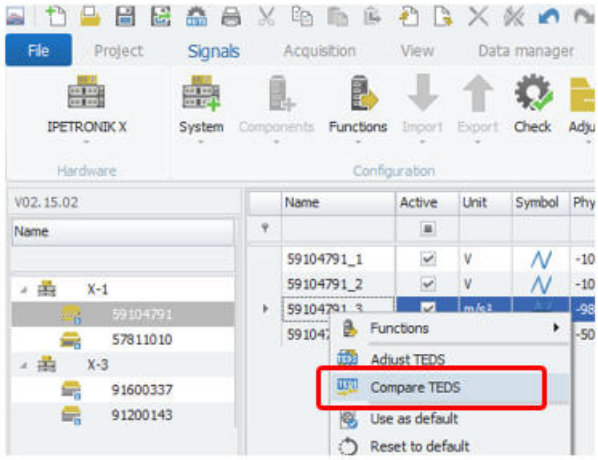


TEDS Adjust function changed unit from [G] to [m/s²] [SI_13]

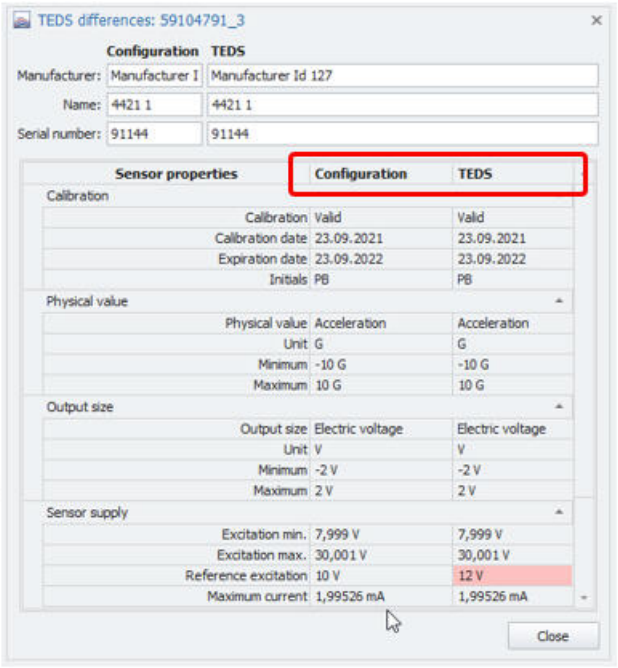
When the synchronization is finished the channel is updated with the TEDS data from the sensor including e.g. the preferred unit when defined inthe Settings.xml file.

Compare TEDS sensor with configuration

When a channel is already scaled based on TEDS parameters, you can use from the context menu the TEDS compare function to update the TEDS data e.g. when a new sensor was connected to the input.



Context menu: Compare TEDS on channel level



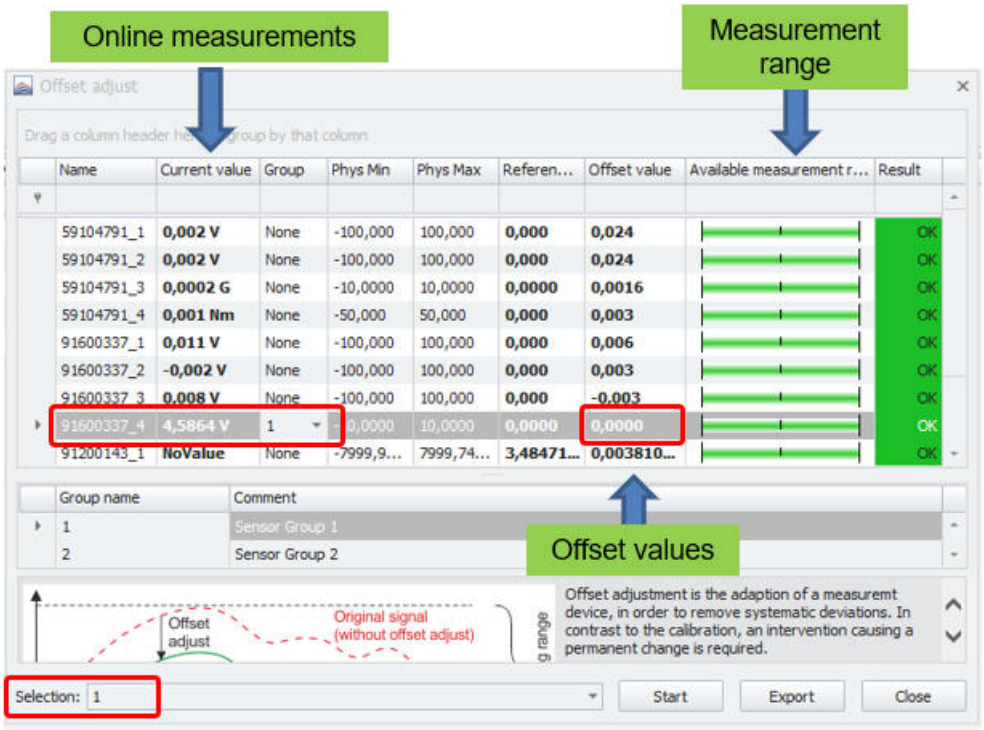
[SI_13_1]

When a TEDS sensor is detected the data is saved into a database file called **IPESensorDatabase.xmt**. In the section Scaling calulator is will be disccussed how to retrieve sensor data from the sensor data base.

- C:\Users\Public\Documents\IPETRONIK\IPEmotion\Database\IPESensorDatabase.xmt

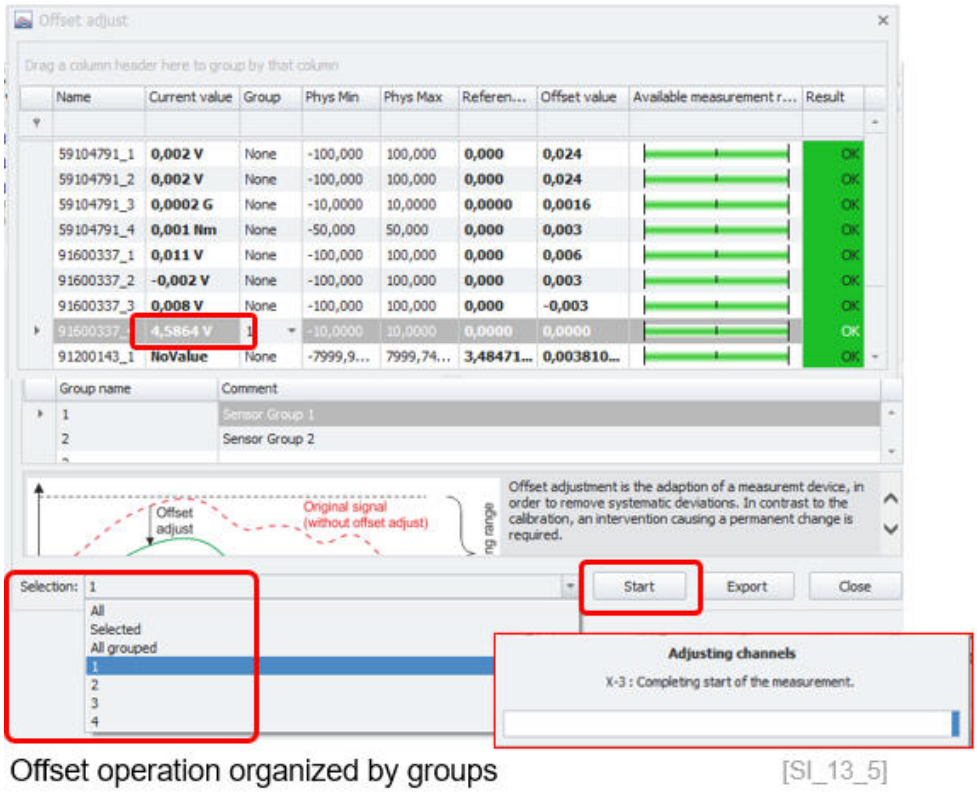
5.2.3 Offset adjust

The offset adjustment is a very useful function to check and update the physical measurements to the configured measurement range. With a offset operation you can shift the current sensor signals on the analog inputs as your new base reference. The offset function can be performed during the online measurements of the system.



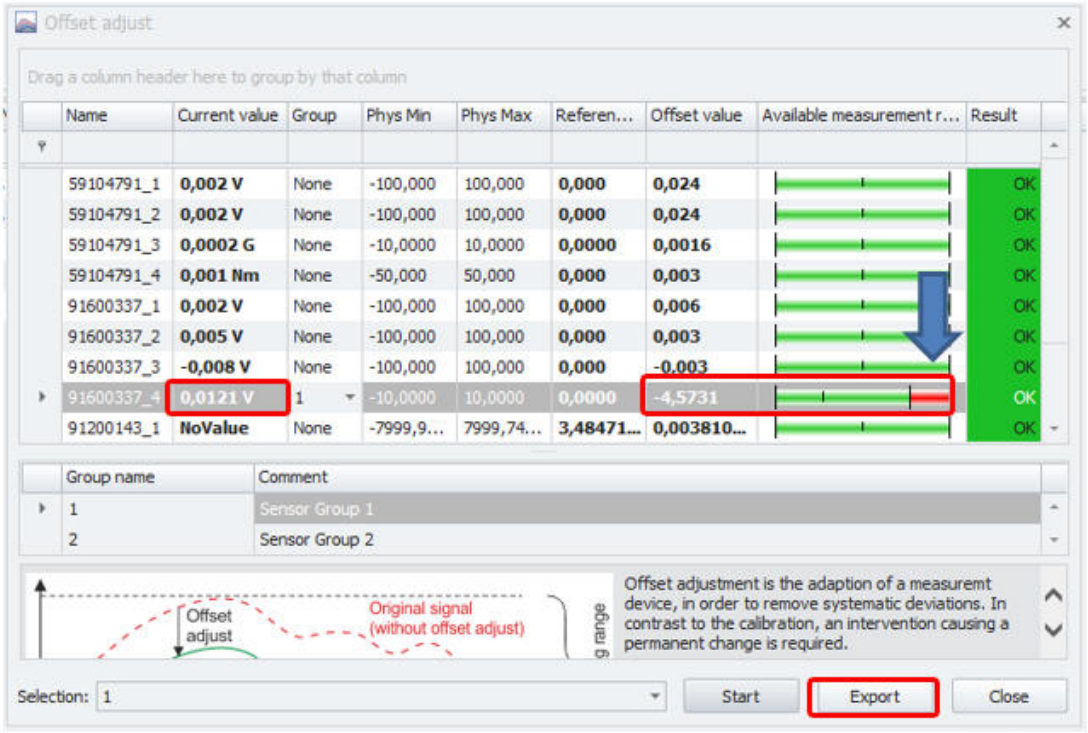
Offset adjust dialog [SI_13_4]

The offset operation can be performed for all channels or only for dedicated groups. In this example 2 sensor groups are defined. Each channel can be rated to a group. In the example below one analog input (assigned to group 1) has already an input voltage of 4.5 Volt.



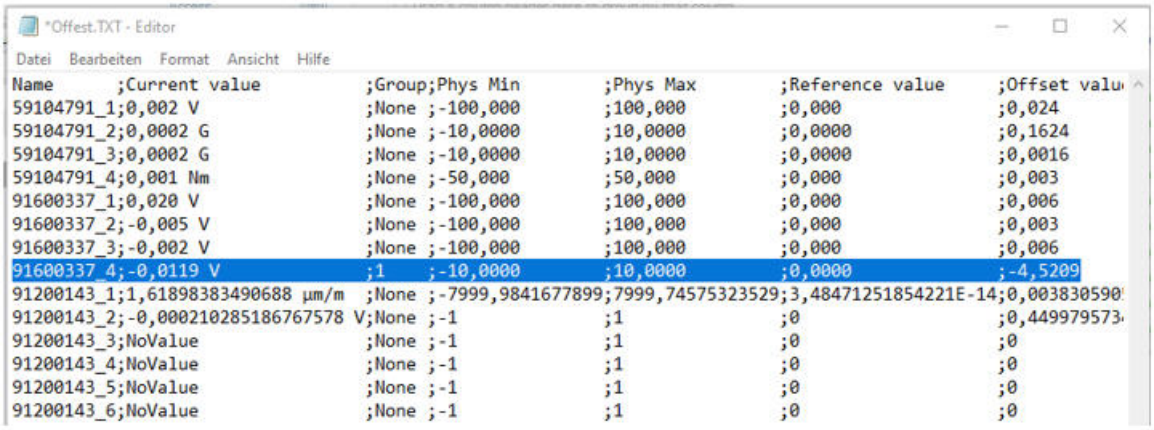
Offset operation organized by groups [SI_13_5]

After the offset operation the initial analog measurement of 4.5 Volt is considered as an offset value and the incoming signal of 4.5 Volt is reference as a new relative zero value. With the offset operation the zero line is shifted by 4.5 Volt. This is also graphically indicated by the available measurement range with is now reduced (red section).



Updated offset values [SI_13_6]

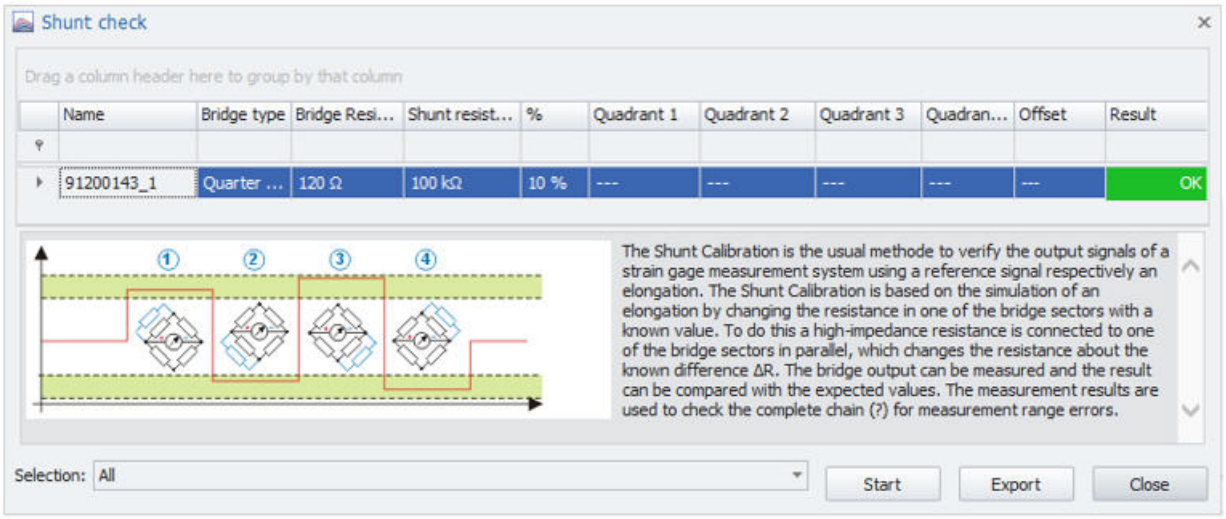
The data of the offset dialog can be exported as TEXT file too.



Export Offset data to TXT file [SI_13_7]

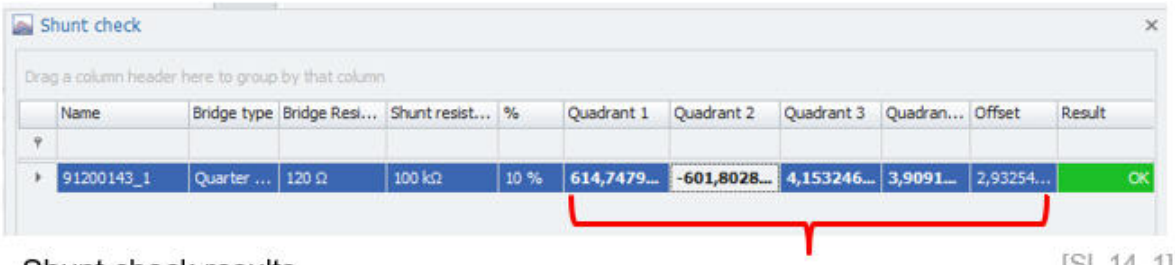
5.2.4 Shunt Check

A strain gauge is used to measure structural load e.g. a chassis frame. During installation and test sensors can be overstretched or damaged. This overload or damage of the sensor is not visible without applying a shunt check. Those damages can result in wrong measurements. Shunt check is used to verify the installed sensor. Shunt checks are performed before and after a measurement. The step response of the shunt check must be the same before and after the test.



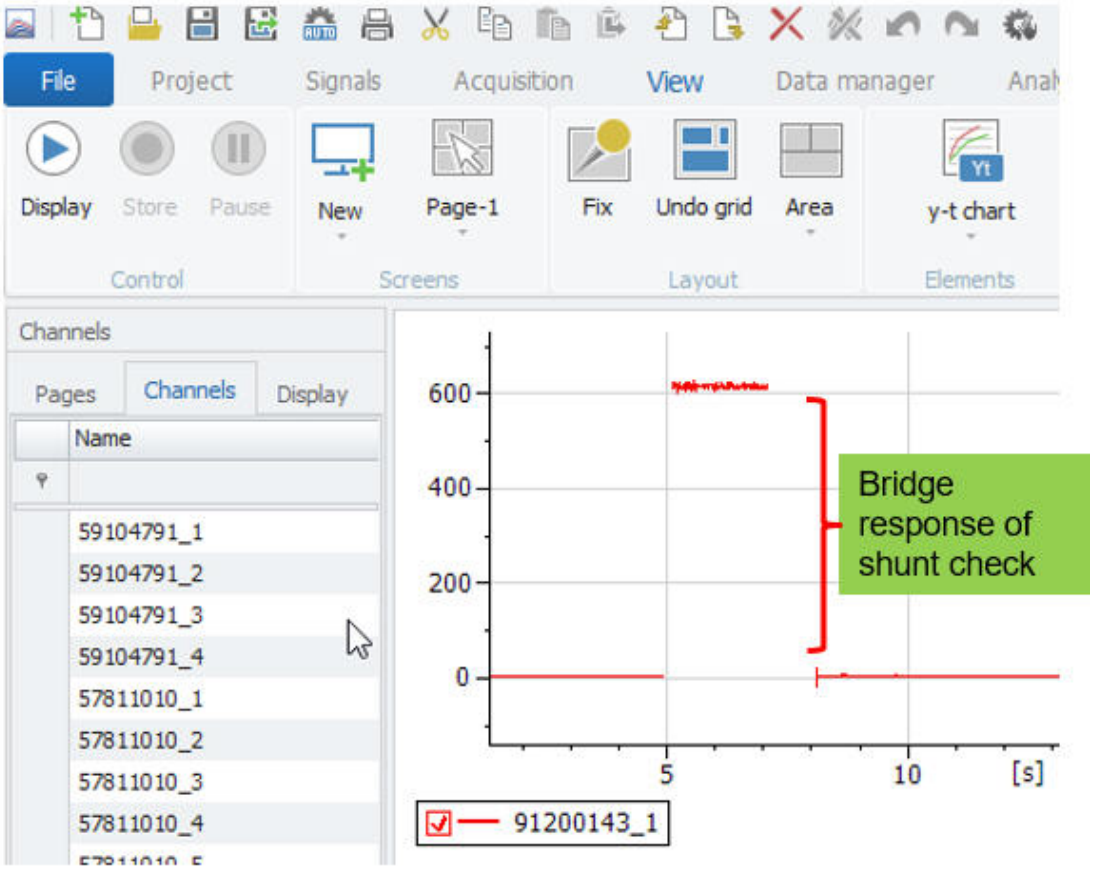
Shunt check dialog for Strain sensors [SI_14]

When the shunt check is raised the bridge resistance is measured and the results are displayed



Shunt check results [SI_14_1]

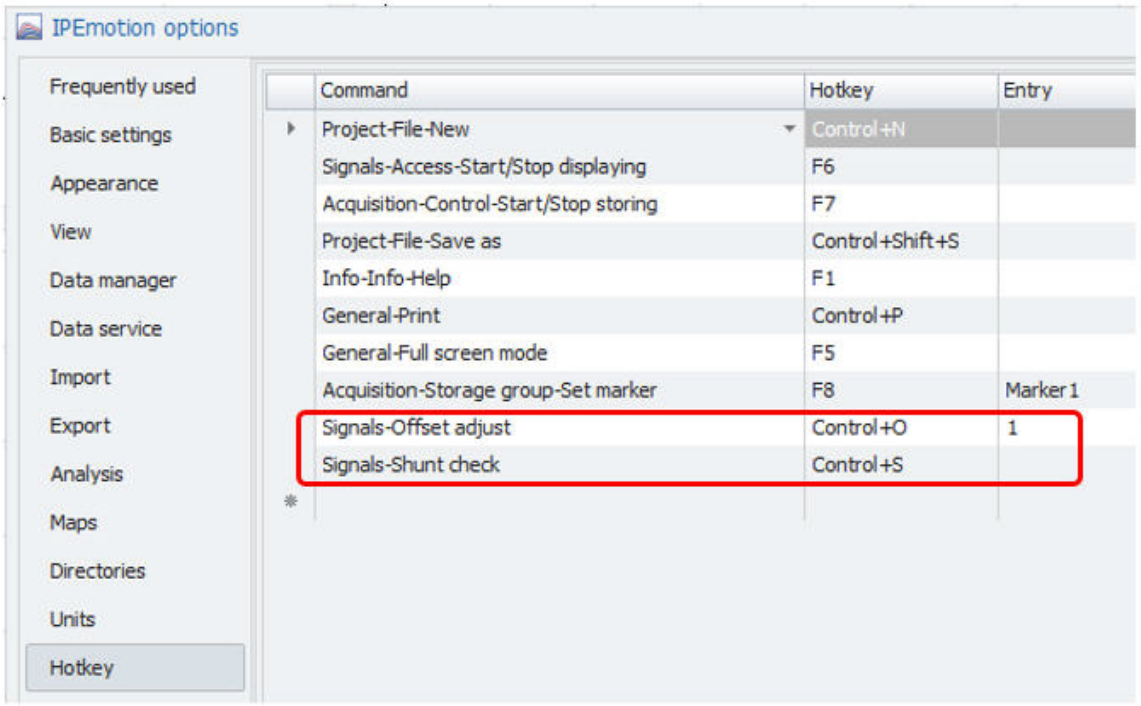
You can also execute the shunt check via a dedicated hot key command. The configuration of the hotkeys is explained. The shunt check can be also made visible in the online measurements like in the yt-chart as indicated below. During the shunt check operation the measurements are online updated in the Yt-chart. The shunt check data can be saved into the measurement file which makes it convenient to compare the shunt check before and after your test.



Shunt check results – bridge response [SI_14_2]

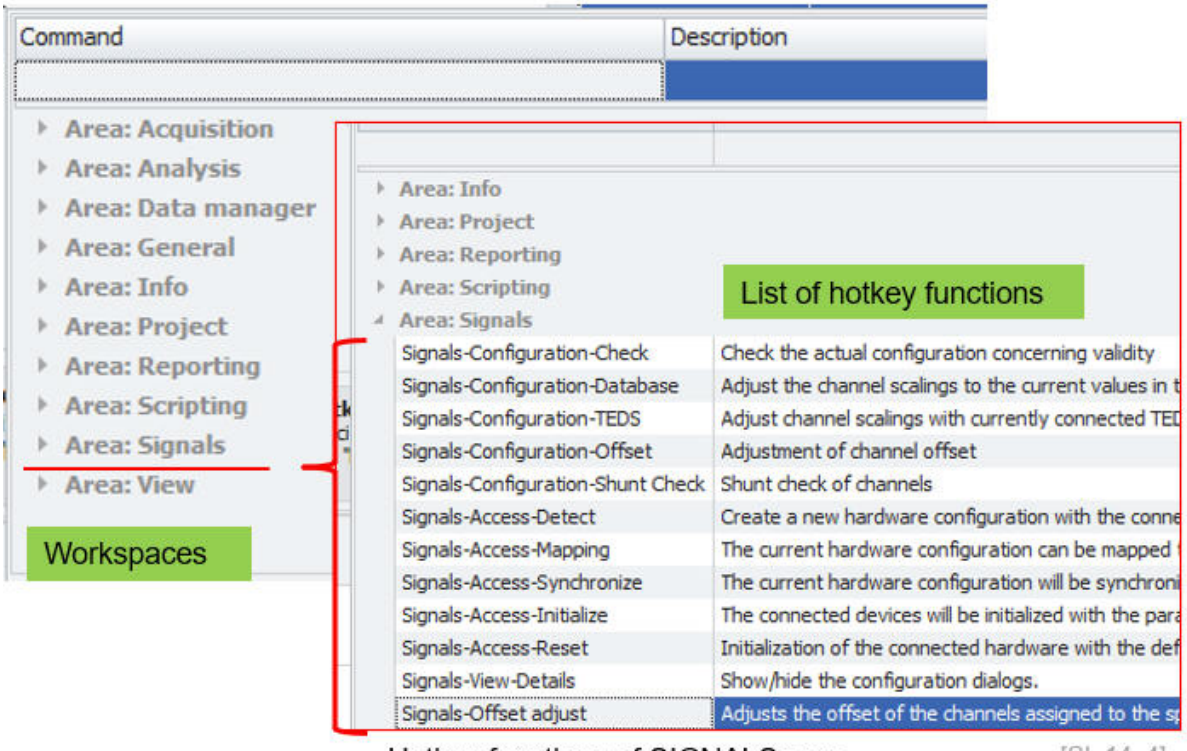
Hotkey operation: offset adjust and shunt check

With a hotkey you have access to functions without using the software user interface. For the offset adjust and shunt check operation and many other functions custom ho key can be configured in the options.



Options dialog: List of Hotkeys [SI_14_3]

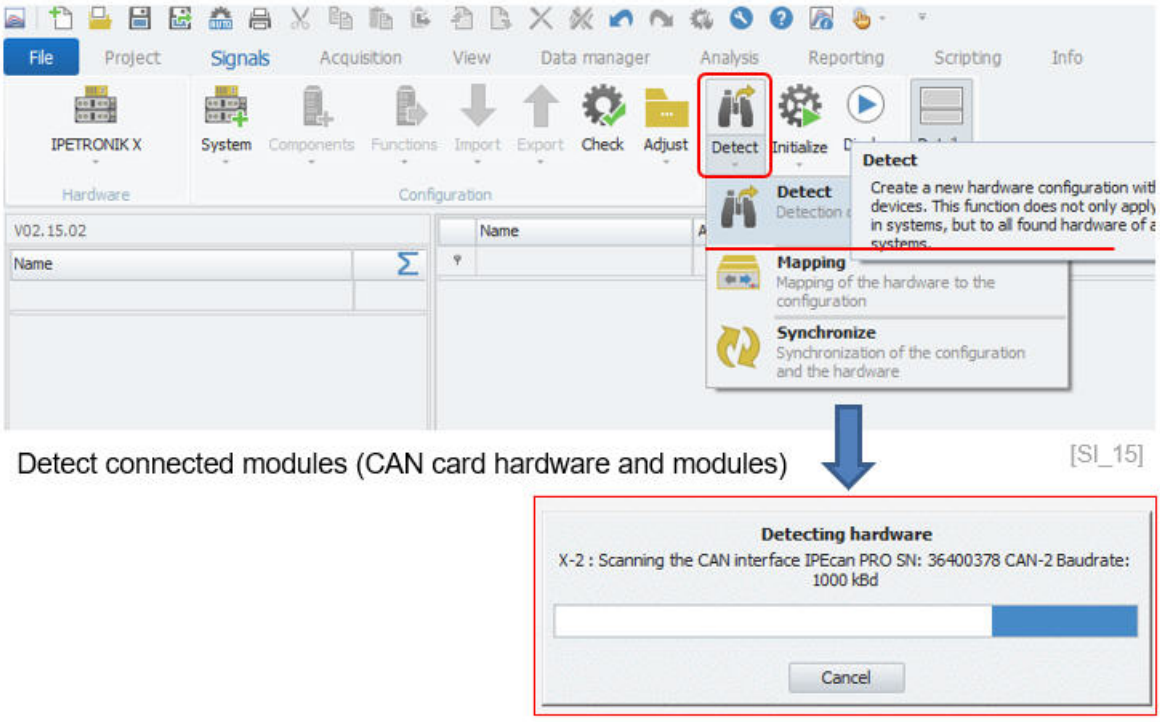
When you add a new hot key function you need to select the required area fist. Within each area dedicated functions are implemented. In the example of shunt check or offset adjust you need to select the SIGNALS area. When the function is selected you define afterwards via your keyboard the hotkey combination



Hotkey functions of SIGNALS area [SI_14_4]

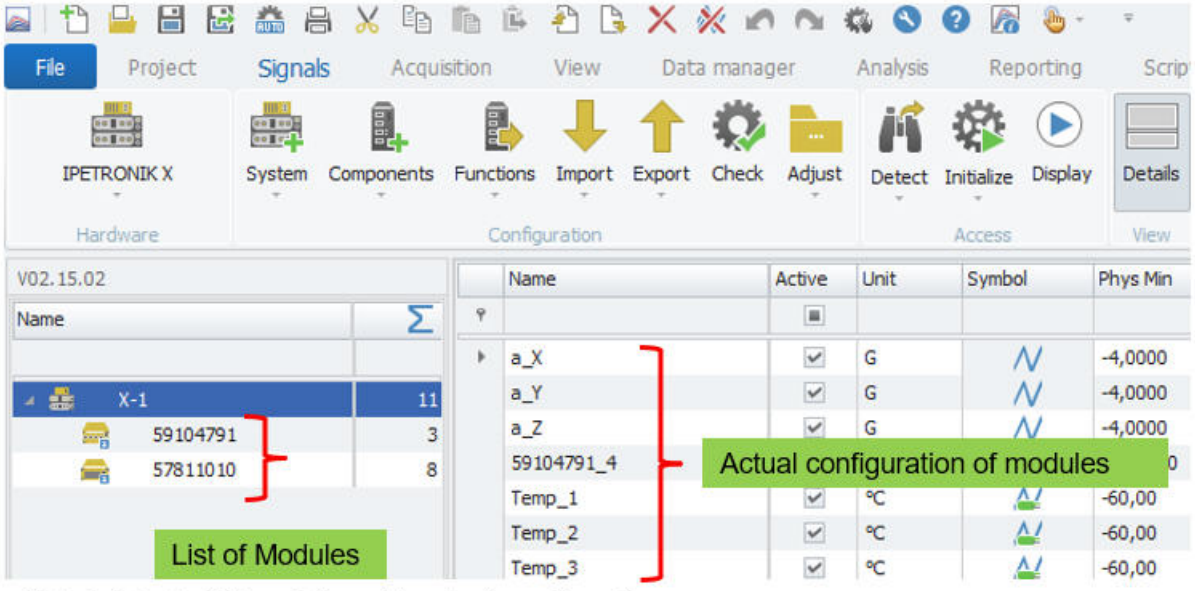
5.3 Detect

When you start working with your analog measurement modules you need setup the hardware and cable sets as discussed above. A supported CAN card hardware and power supply is required. The easiest way to get started is to run the DETECT function as indicated below.



Detect connected modules (CAN card hardware and modules) [SI_15]

The DETECT function is a very convenient function to identify any hardware connected to IPEmotion. Not every PlugIn supports automatic hardware detection. Usually, USB device interfaces support automatic hardware detection. The DETECT function is applied to all active PlugIns. It is recommended to use the DETECT function only for the very first time when you start to set up your measurement configuration. If the hardware configuration is changing by adding or removing modules, you need to execute the SYNCHRONIZE function to update the complete hardware configuration in the device tree. The SYNCHRONIZE function is explained in detail in section 5.3.2.

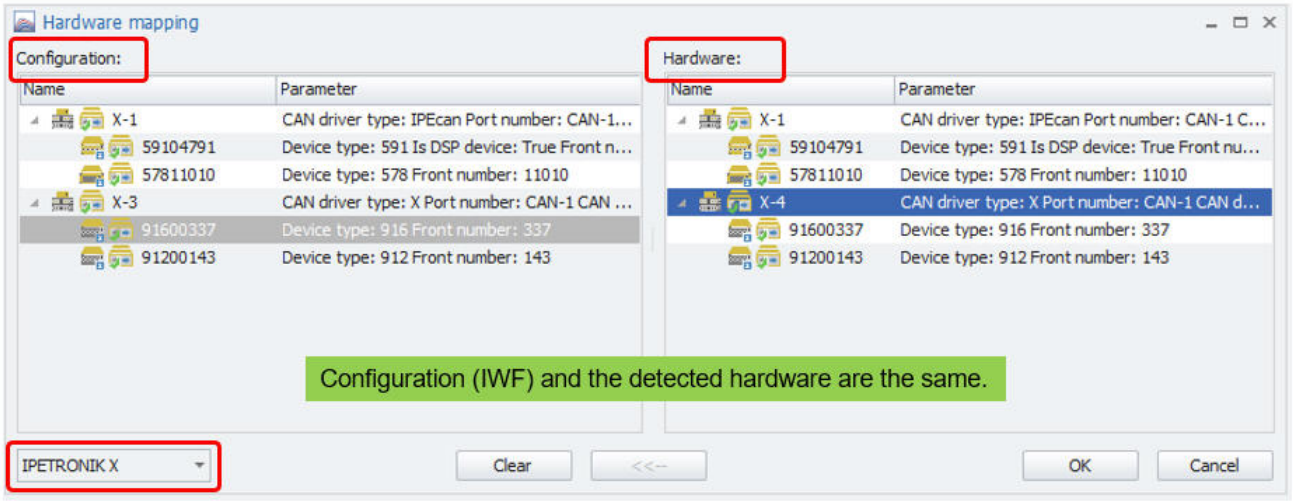


List of detected M-modules with actual configuration [SI_16]

Attention! *If you execute the DETECT function the complete configuration of SIGNALS of all connected devices is recreated. Additionally, all the configurations from the ACQUISITION work space are removed.*

5.3.1 Mapping

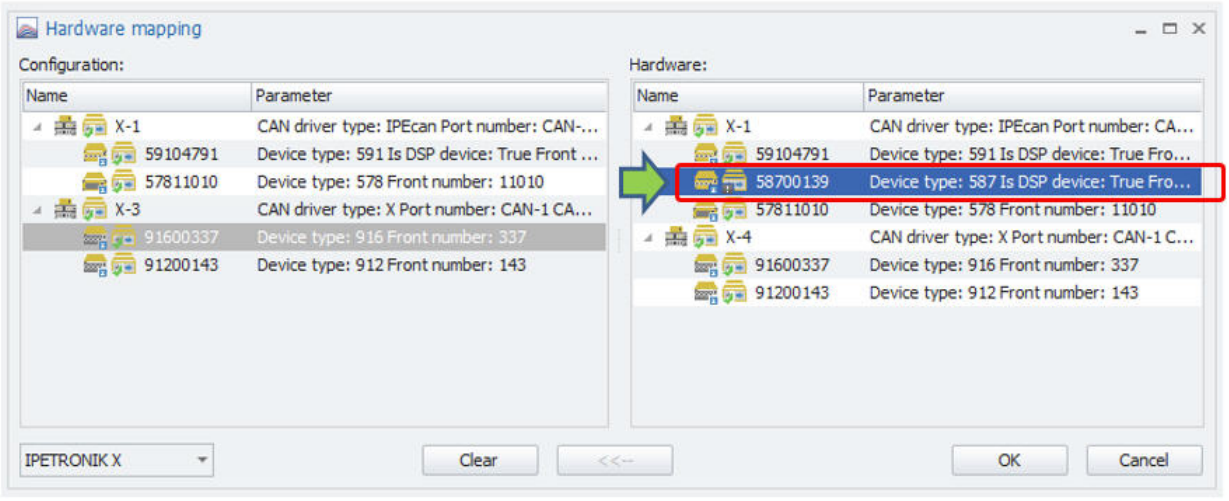
The hardware MAPPING is a very convenient function for merging configuration (IWF) files to the currently connected hardware. If you execute the MAPPING function, the current configuration is compared to the currently connected hardware. IPEmotion is starting the hardware detection to identify all currently connected modules.



The Mapping function compares the current configuration (IWF) to the currently detected hardware across all PlugIns. [SI_17]

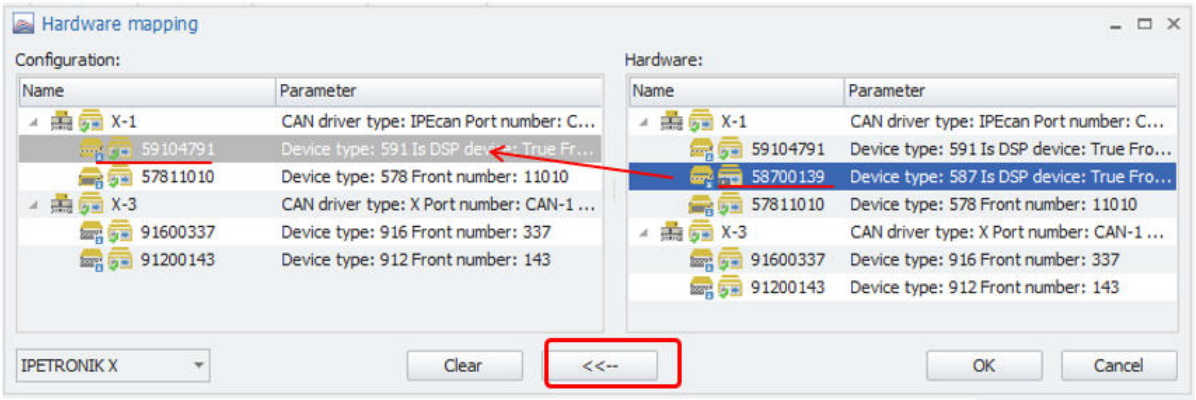
Information *The MAPPING function is only supported for those PlugIns which support automatic hardware detection.*

In the following you will see an example how to use the mapping function in practice. There are applications in which the same configuration is applied to different hardware setups. For example, each IPETRONIK module has an unique front number and using the mapping function, the actual hardware configuration can be matched to the configuration file. When the MAPPING ope process has detected one new module as indcated in the screenshot below.



Configuration (left) and the detected hardware (right) are different. [SI_18]

In oder to performe the mapping you need to select the modules you like to map from the detected hardware to the corresponding configuration. The mapping process works basically from right to left. You can define several mapping relations by linking one module from right side to one module of the left side. With the arrow button to save the mapping between modules.



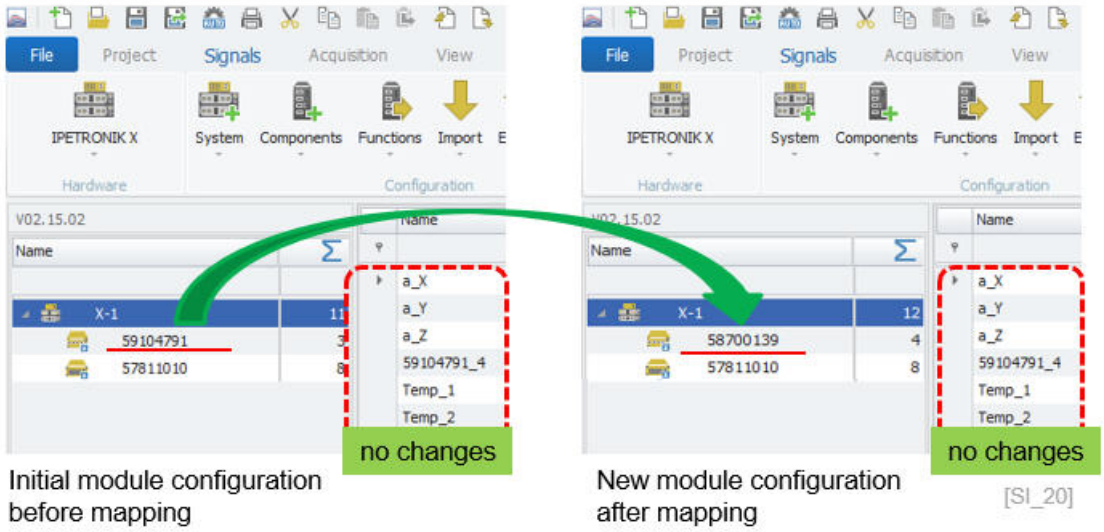
Step 1: Select the modules you like to map (one on each side only)
Step 2: Press the button [←] to execute the mapping process

[SI_19]

Information

The Mapping function can be applied across different modules types. The system is not preventing you from mapping for example a M-SENS module to a M-THERMO module.

When you conform the MAPPING process via the OK button the new module type including the device serial number is updated. However, the channel configuration remains untouched.



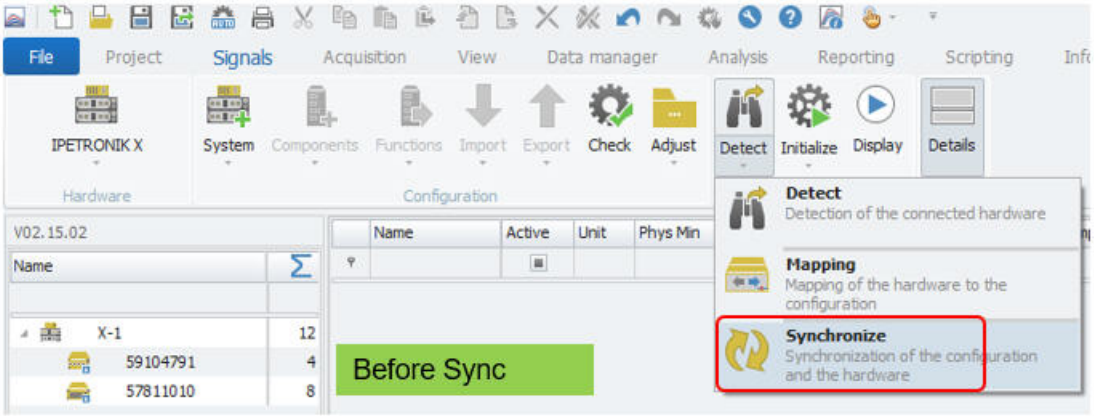
Initial module configuration before mapping

New module configuration after mapping

[SI_20]

5.3.2 Synchronize

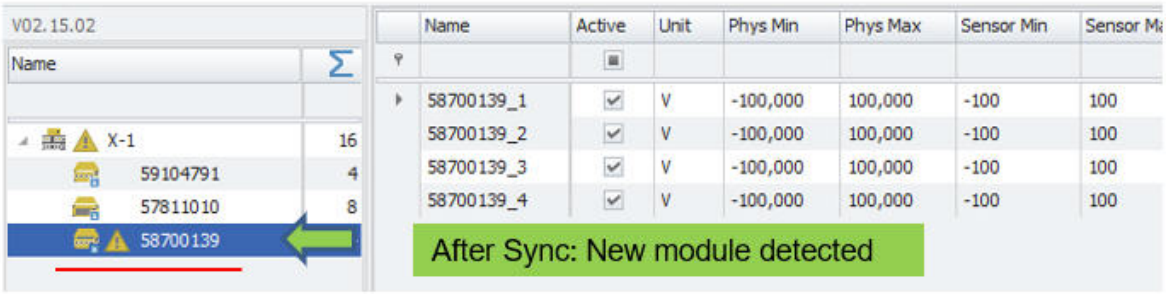
The SYNCHRONIZE function is designed to update an initial configuration (IWF) with an updated hardware setup. This function is the counterpart of the DETECT function. As discussed above the DETECT function is creating your initial module setup. In practice the module setup can change where new modules are added or removed to the configuration. With the SYNCHRONIZE function you update your modules easily to your configuration.



[SI_23]

Information *The SYNCHRONIZE function is not changing any configurations defined in the ACQUISITION work space.*

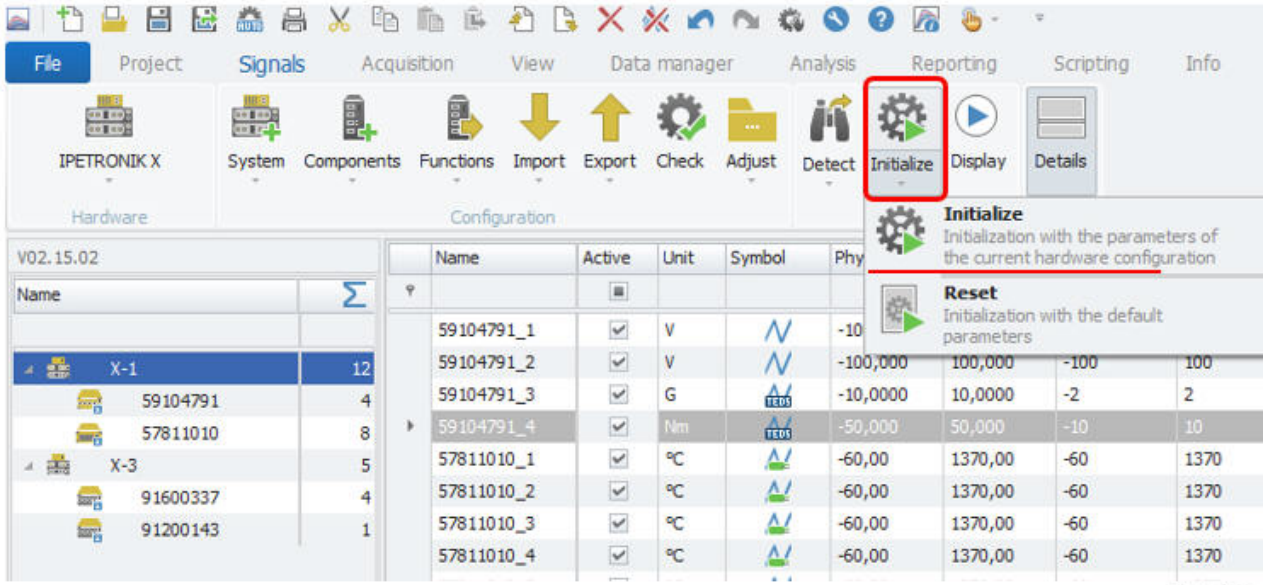
If you make changes to your measurement hardware by adding new modules or removing modules it is recommended to use the synchronize function to reflect the hardware changes in your module tree. New modules are added to the tree. In the case that modules are removed a waring icon is presented in front of the module serial number.



Synchronize function updates changes in the measurement hardware setup [SI_24]

5.4 Initialize

With the INITIALIZE function you can test the communication between your hardware and IPEmotion. If there are configuration errors or the hardware cannot be reached, messages are returned. Depending on the PlugIn version, error, info or warning icons are indicated.



Initialization of hardware [SI_25]

The INITIALIZE function is also updating the hardware with the latest configuration parameters defined in IPEmotion. The configuration is downloaded to the devices. So when you run a hardware detection the latest configuration settings like channel name, scaling etc. . . are automatically retrieved from the module and displayed in IPEmotion. However, in many cases the hardware cannot store a configuration. In this case, the configuration is only on the PC side but is not transferred and stored in the hardware. The IPETRONIK modules store the following configuration settings internally:

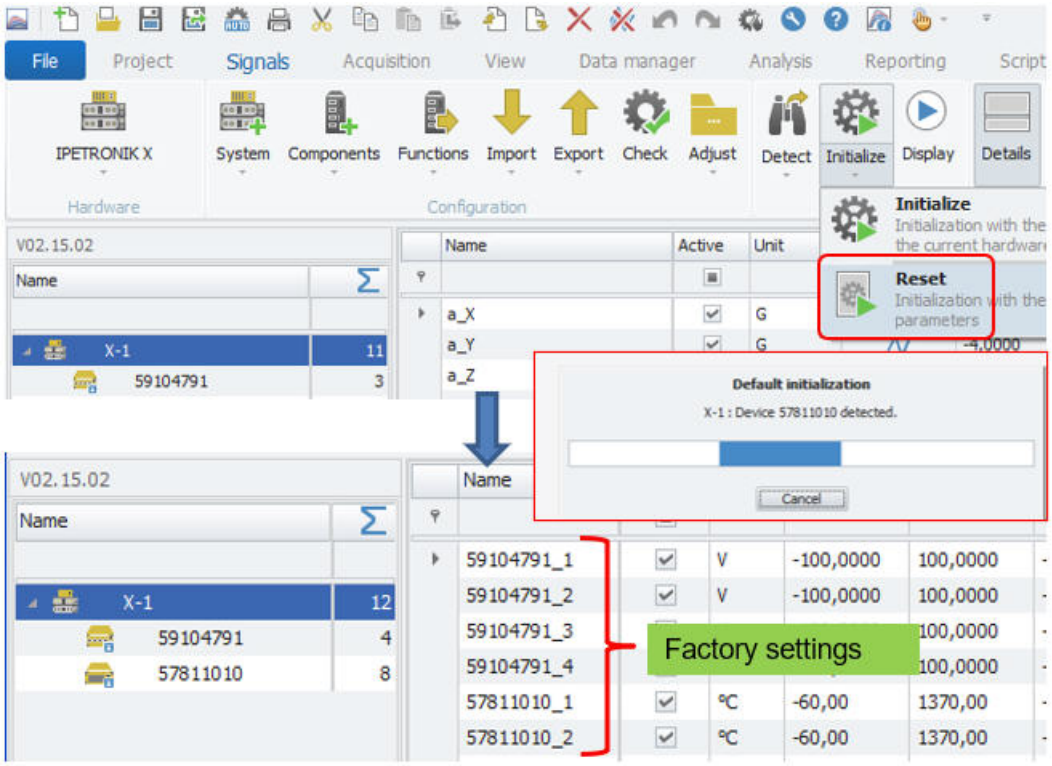
- Channel name
- Physical units
- 2-Point scaling
- Free 2 point scaling
- Factor offset scaling
- Sensor measurement range
- STG mode
- Data type (format)
- Characterisitc curves (for X-modules only)(see section XXX)

The following configuration settings **are not stored** in the IPETRONIK modules internally *:

- Channel description
- No value
- V-TAB (see section 6.8.7)
- V-TAB range (see section 6.8.6)
- Multi point scaling (6.8.4)

5.4.1 Reset

The reset function is relevant for instruments which can store a configuration in the device. After reset, all configurations stored inside the device are set back to factory default.

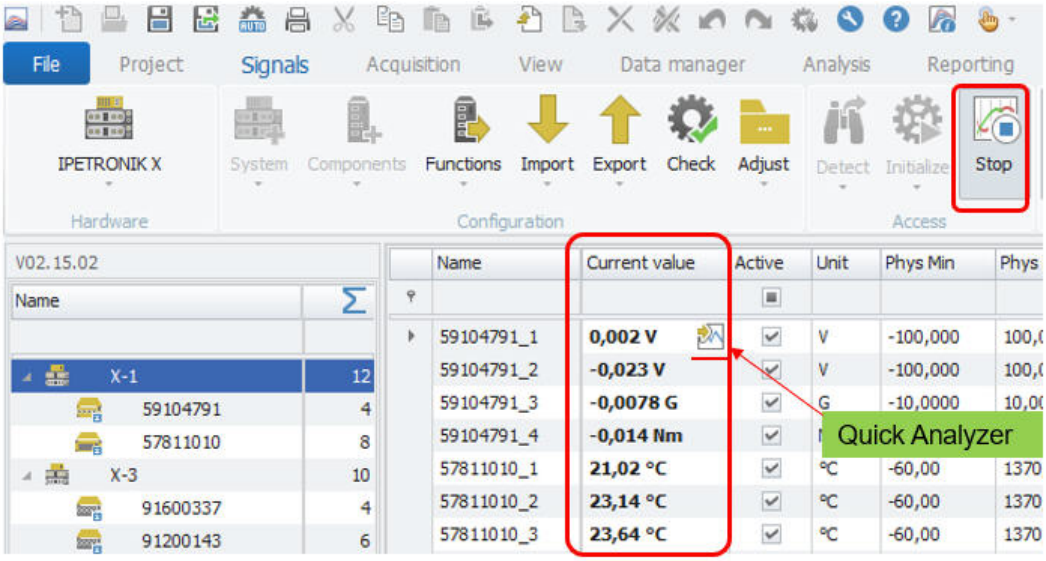


Reset to default / factory settings [SI_26]

Information *The RESET is applied to all PlugIns which support the RESET function. The function is implemented and used for IPETRONIK modules and data loggers as these instruments can store a configuration.*

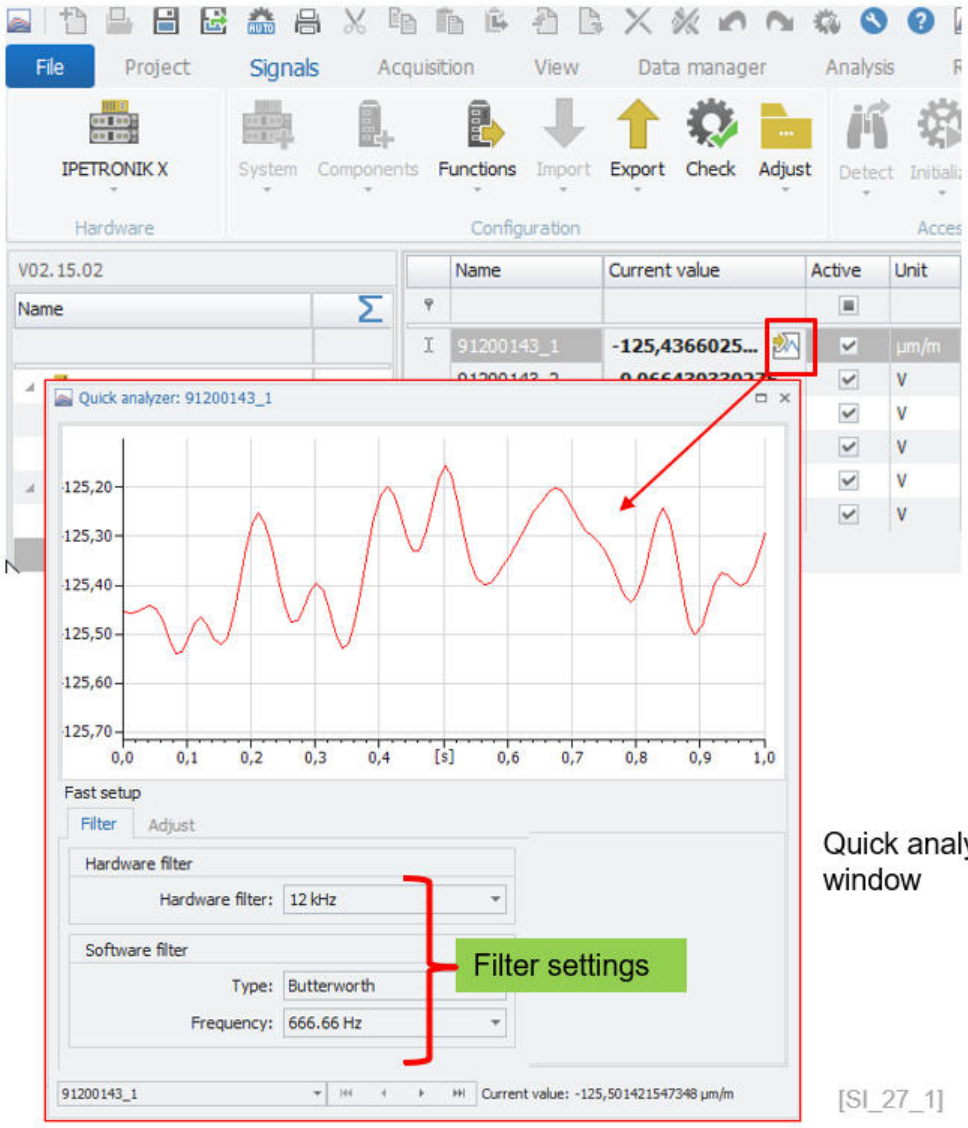
5.4.2 Display and Quick Analyzer

The Display button turns your configuration into measurement mode. Then you will see measurement values for all active channels.

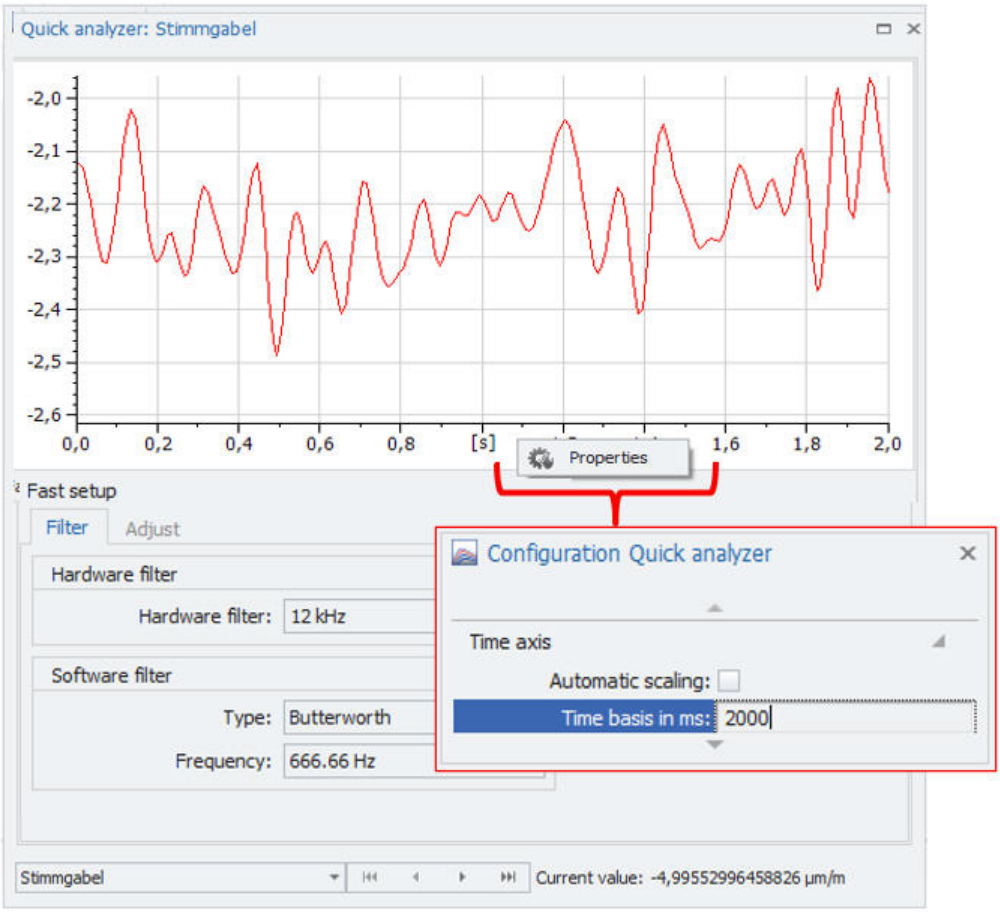


Display online measurements. [SI_27]

With the quick analyzer you get a direct preview to the channel signal. The analyzer performs auto scale and shows the current value. Within the instrument you can switch easily between channels of the same module.Some of the IPETRONIK X-Modules with the latest firmware support a fast setup functionality. With the fast setup, some channel properties can be changed without device initialization. These configuration elements are performed on the fly while measuring without interrupting the actual measurement. This fast setup functionalities is supported by the X-modules only and refers to software filter settings, offset adjust and shunt check.



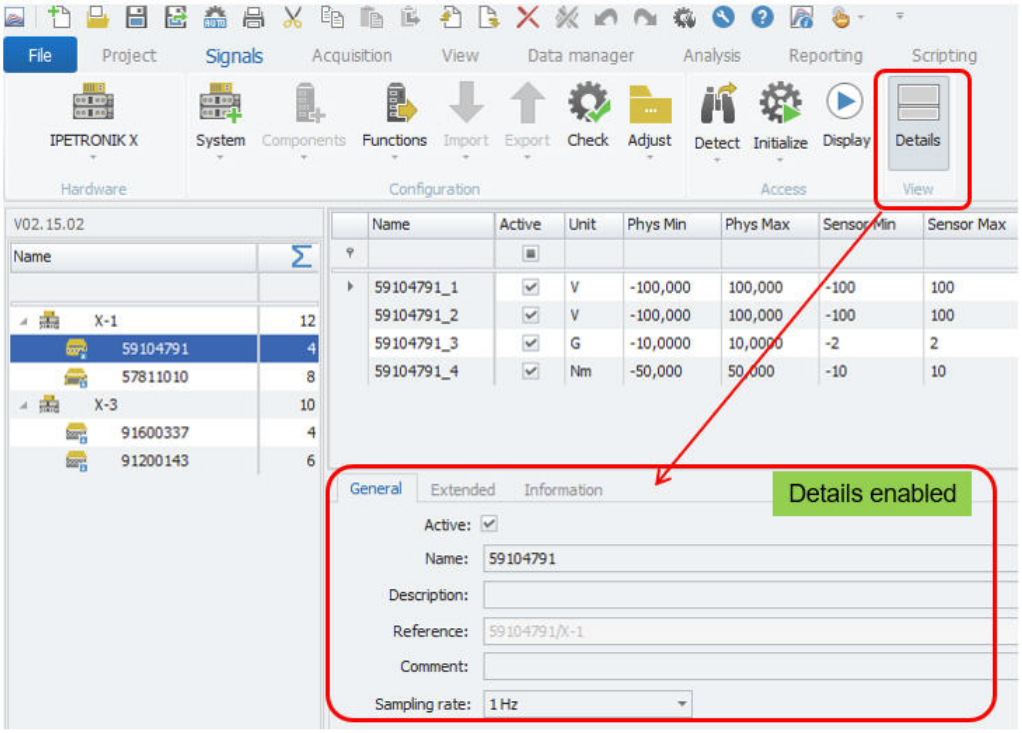
The default time window or update rate of the quick analyzer scope window is 1000 ms (1 second). However, if you access the properties via a right click on the x (time axis), you can change the display time range from 0,001 ms (1 micro second) up to 2000 ms (2 seconds).



Quick Analyzer – Time range / update setting [SI_27_2]

5.4.3 Details

With the DETAILS button in the ribbon you can display or hide all tab sheets for systems, modules and channels configuration.

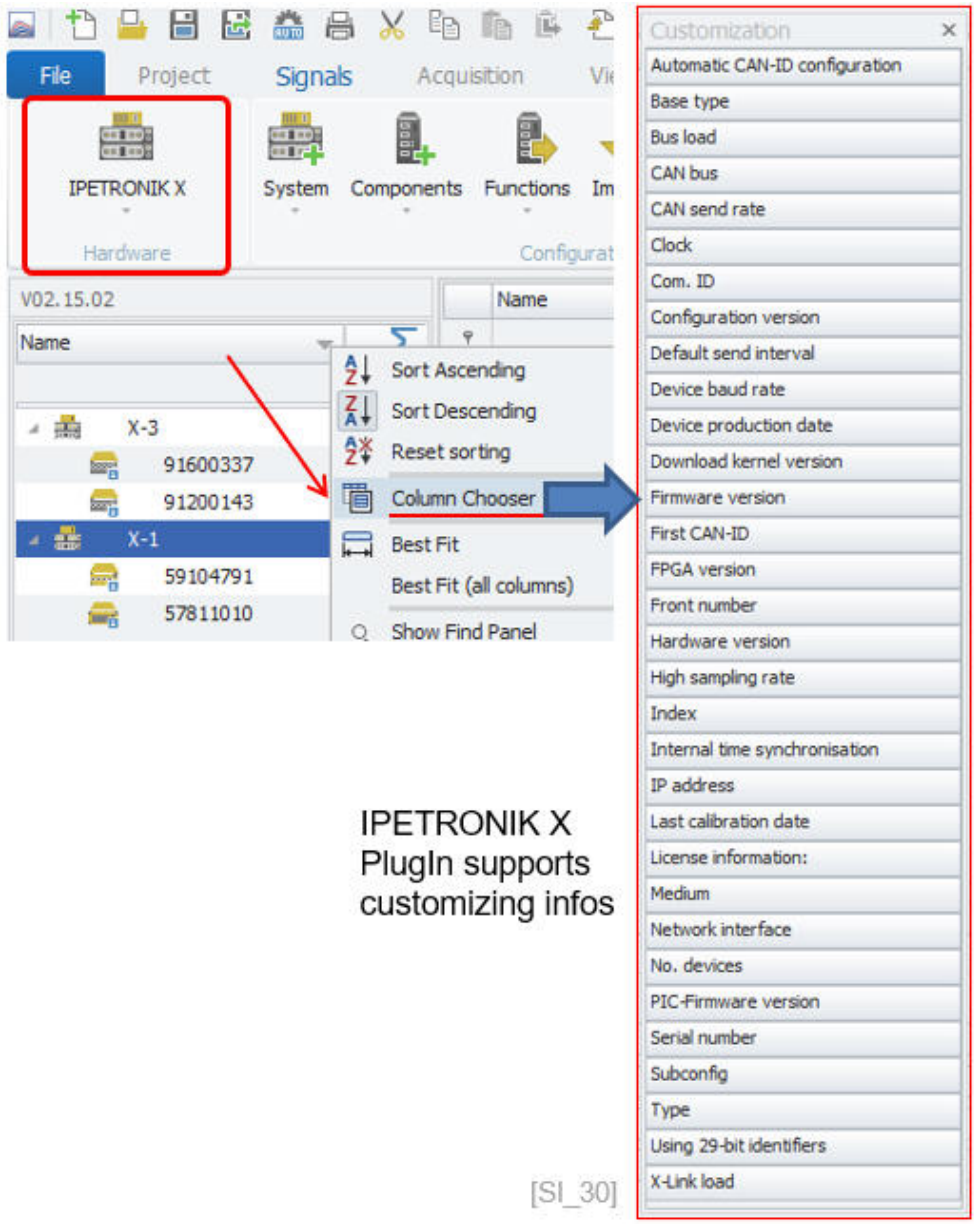


Tab sheets to access configuration details of the system, module or channel [SI_28]

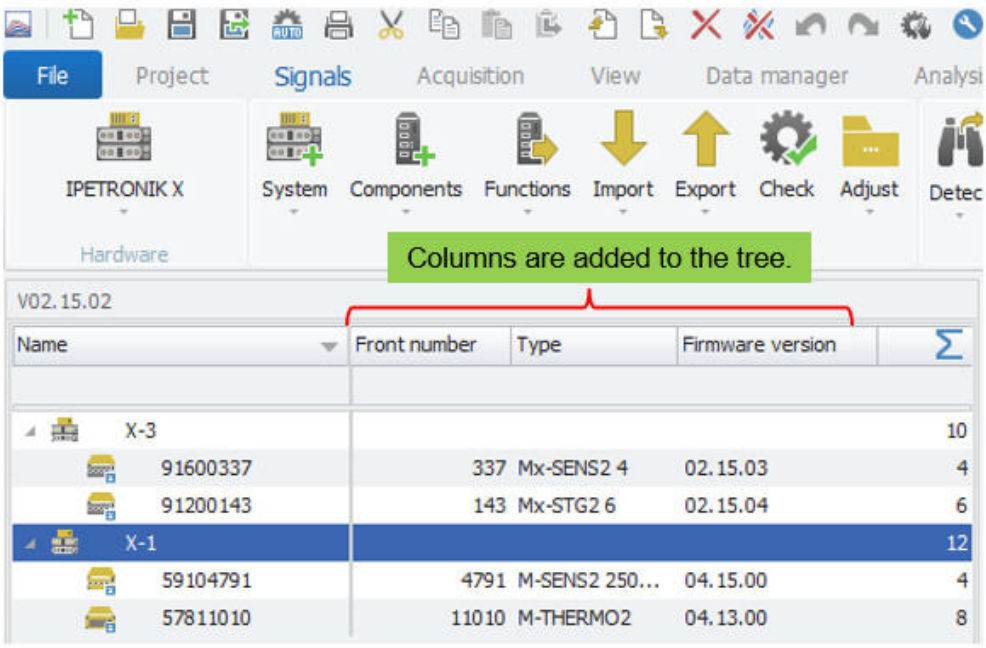
5.5 System Tree

5.5.1 Column chooser

In the system tree you can activate a column chooser. In the system tree you can activate a column chooser by right click on the column header. This is a very useful function to add additional properties to your devices and modules. The scope of functions in the column chooser depends on the scope of the implementation of the PlugIn.



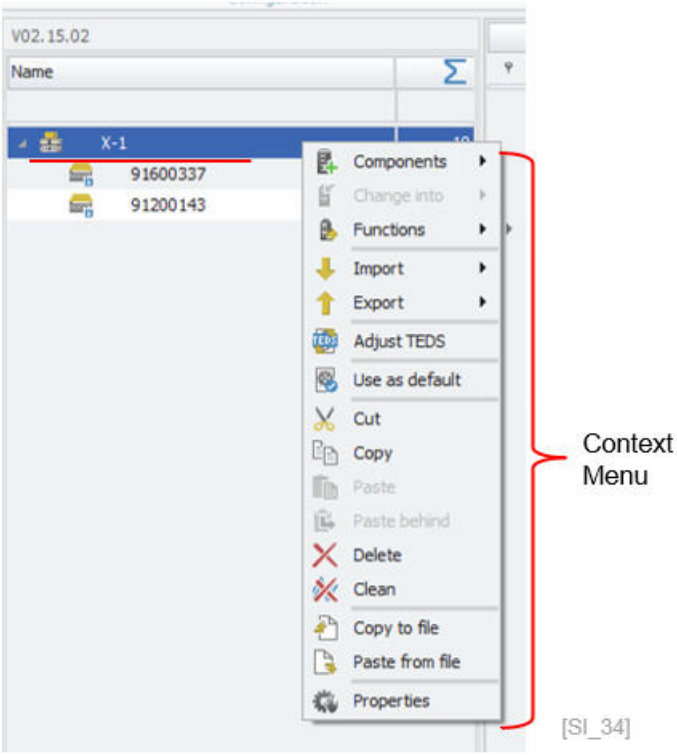
In the example below 3 additional columns are added on device try to indicate the front number, the device type and the firmware version. You can filter and sort across all additional columns if required.



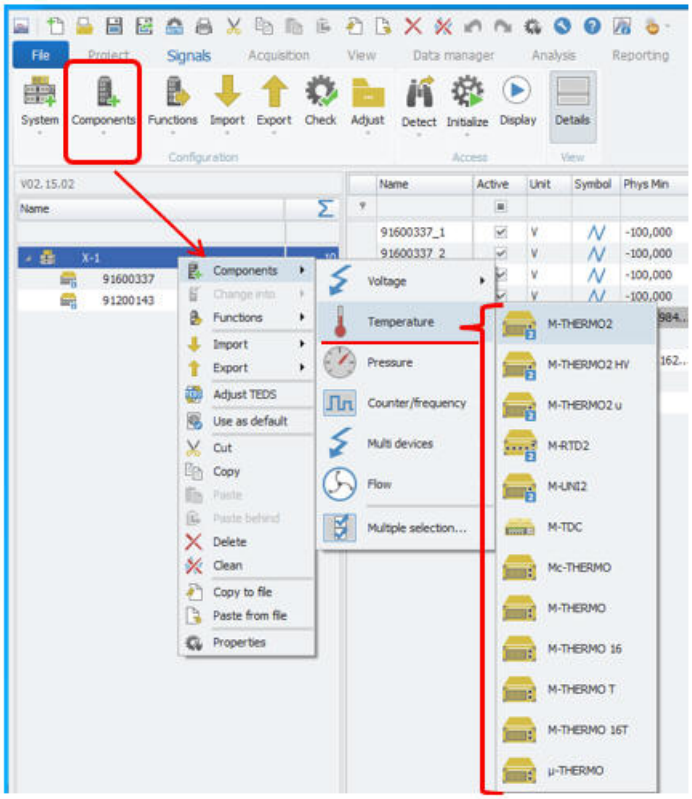
Column chooser on module / device level [SI_31]

5.5.2 Context menu for system, modules and channels

The context menu offers convenient functions for setting up your application. With right click to the system, module or channel you can access the context menu. The functions provided in the context menu depend on the PlugIn. Some PlugIns offer plenty of functions and other just provide some basic functions.



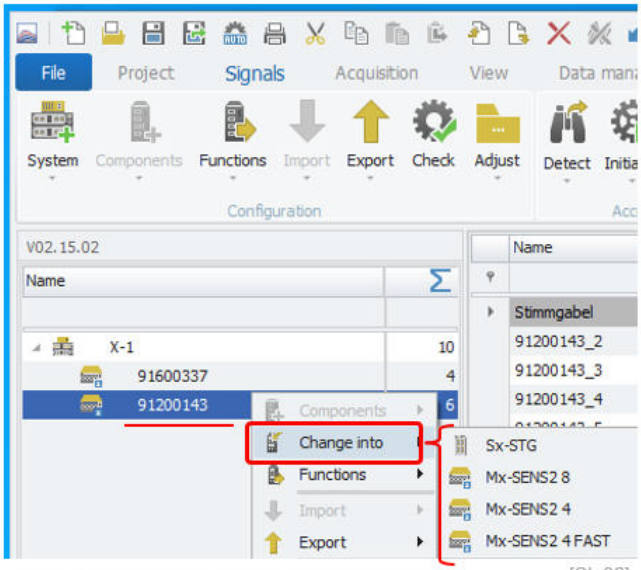
- Components — You can add components if the PlugIn supports a modular hardware structure.



Components of X-PlugIn for „Temperature“ modules [SI_35]

- Change into

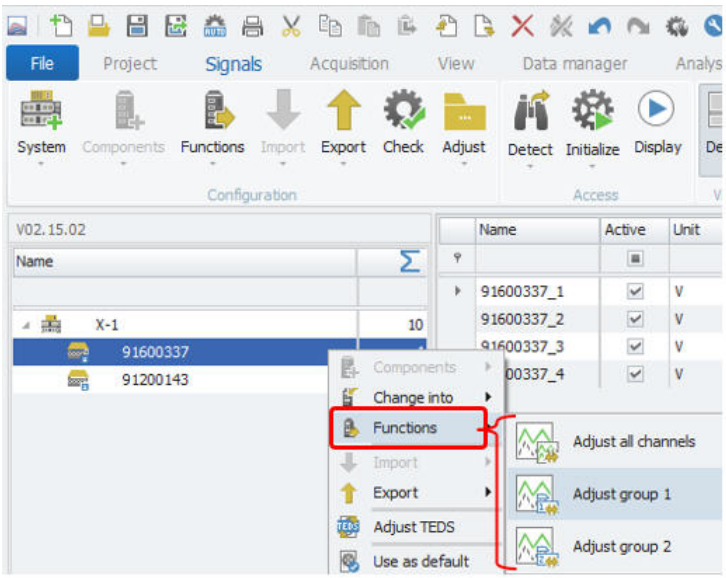
This function can convert a component/module to another type. Basically, if you build your configuration offline and you change the type of some modules without rebuilding the complete configuration, you switch modules with the ”Change into” function. The change into function will also try to shift the software configuration to the new module provided the function are supported. The configuration between SENS modules is most likely compatible. When the configuration is transferred between modules types the sensor excitation is set to zero as modules support different sensor excitation types (unipolar or bipolar) and sometimes different voltage levels.



Change a component to a different type. [SI_36]

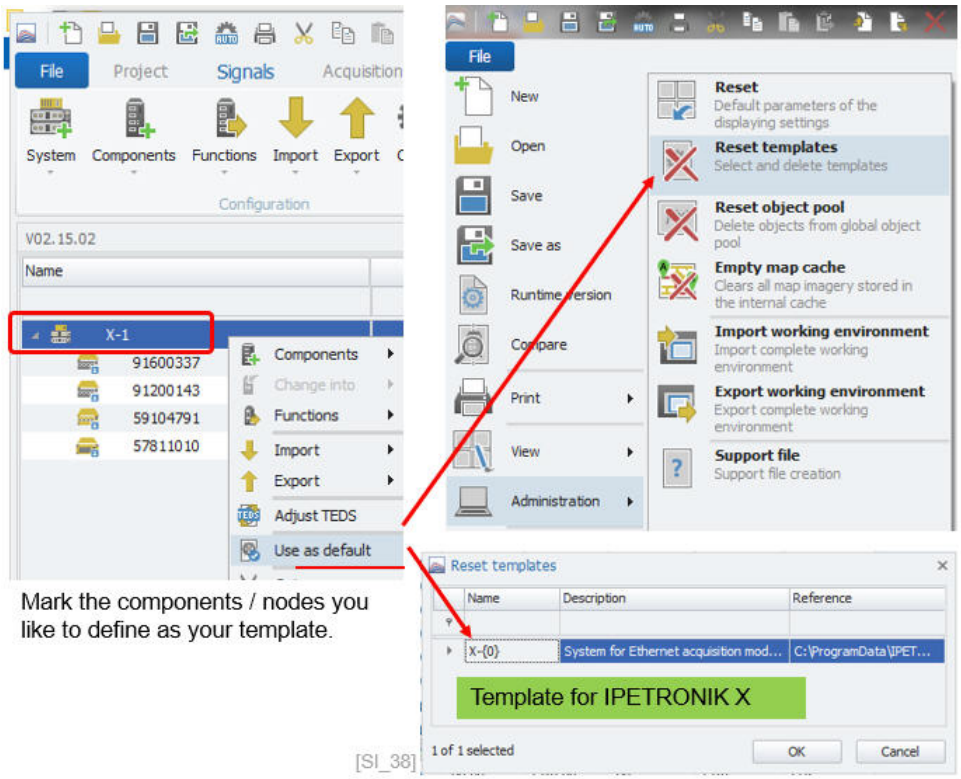
- Function

Behind the “Function” you can do on CAN or ETH interface level a firmware update. However, on module level you can run an offset-adjust of all channel or different channel groups when those groups are defined.



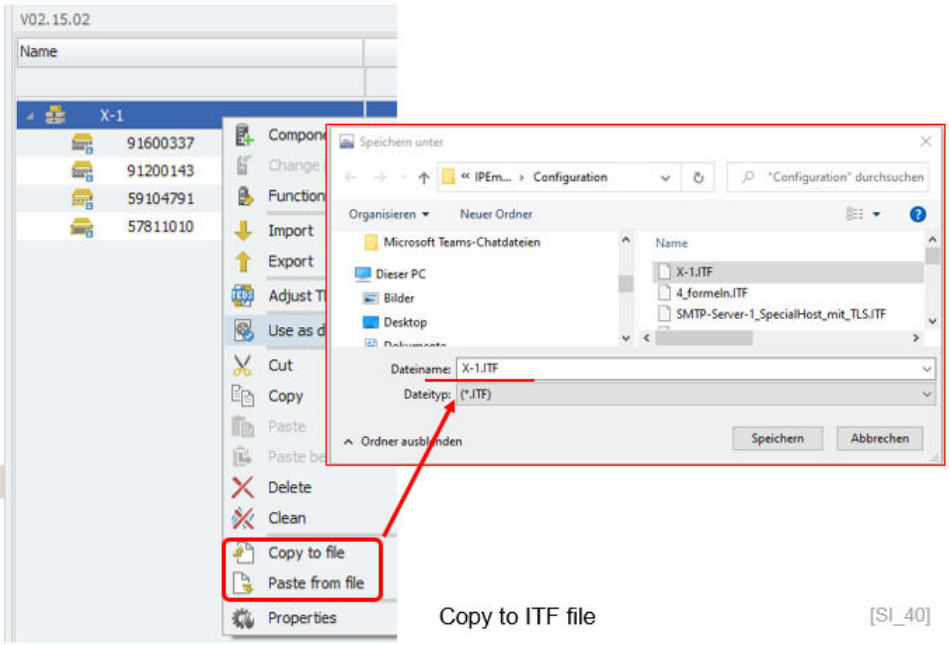
Function – Channel offset adjust on module level [SI_37]

- Import / Export — This function refers to the same function as implemented in the main ribbon. There are plenty of different import and export functions available. It is mainly related to configuration files like A2L, CANdb, Autosar etc. They are discussed in the previous chapter **??**
- Adjust TEDS — The TEDS detect and adjust function is explained in detail in section 5.2.2.
- Use as default — This function is useful for all users who need to create the same configuration several times. If you save your master configuration as DEFAULT, all systems are created with this order of modules, automatically. The default configuration is saved and can be deleted in the Application menu as discussed in detail in the OPTIONS. You can only define one template for one interface. E.g. you cannot have different module configurations for IPETRONIK X.



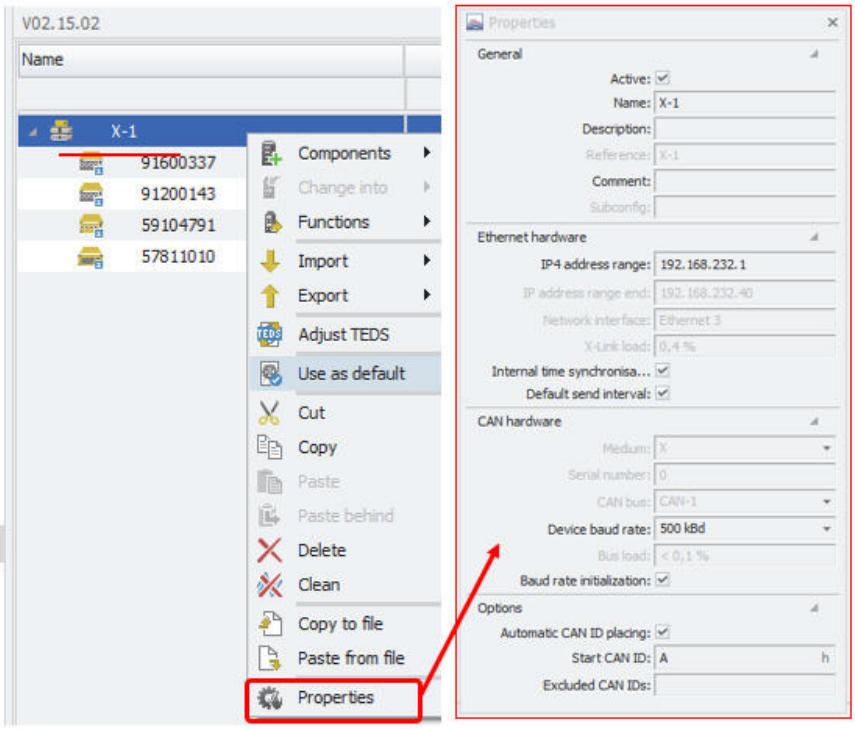
- Cut — With the cutting function you can cut out selected modules. After cutting components you can paste them in other sections of the system tree. There is a difference between ”Paste” and ”Paste behind”
- Paste — Insert one module
- Paste behind — Inserting all modules you have cut out and paste them behind a selected module
- Copy — With the copying function you can duplicate the one module or a list of selected modules.
- Delete — With ”Delete” you permanently remove the items from this configuration
- Clean — The ”Clean” function only works on an interface or system level. With this function you can remove all modules beneath the interface.

- **Copy to file** — With this function you can save module configurations in a separate file with the extension ITF. This ITF file can be imported, as well
- **Paste from file** — Import ITF files. They include all selected modules, channels and configuration elements



Copy to ITF file [SI_40]

- **Properties** — If you select ”Properties” from the context menu, another display opens up summarizing the tab sheets for configuration. The properties are context- sensitive. If you select a module you will get the context for module configuration. If you open the connext menu on channel level you will see all configuration tab sheets related to the channel.

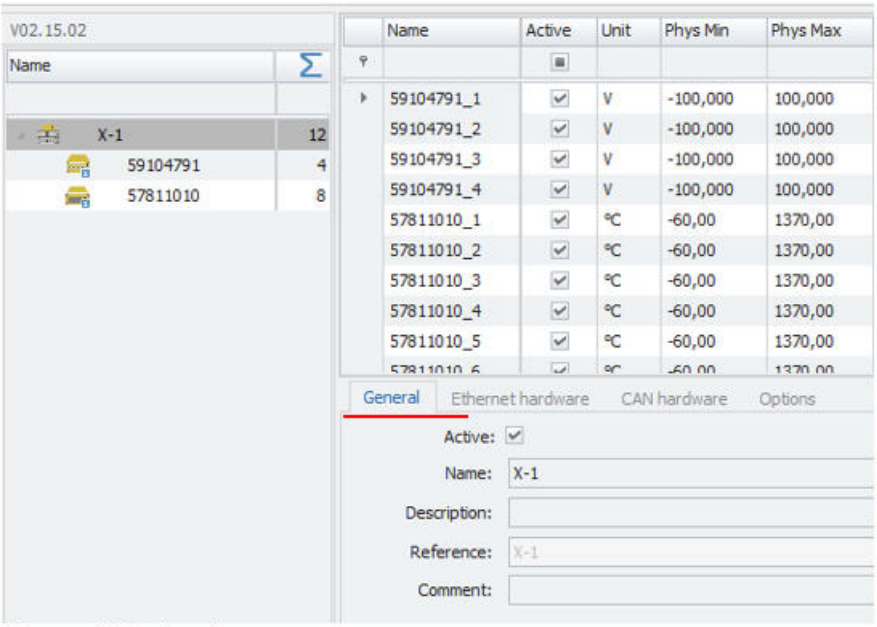


Interface properties [SI_41]

6 Interface configuration

As discussed above for the IPETRONIK X PlugIn are 2 main module product lines available based either on CAN bus or Ethernet communication. Both interface settings are discussed below. There are also plenty of device-specific tab sheets which are individual to each PlugIn. Detailed descriptions about different settings are part of the individual PlugIn manuals.

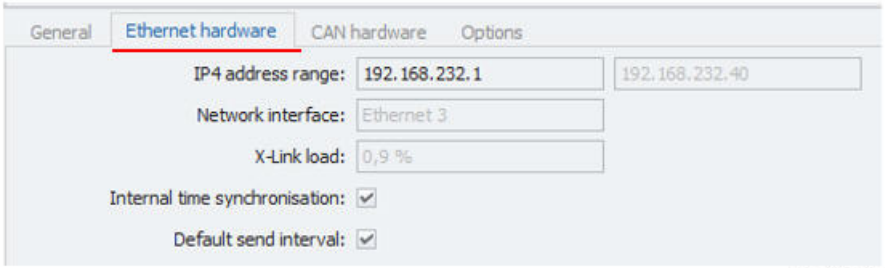
6.1 Interface – General tab sheet



General tab sheet [SI_42_1]

- **Active** – Here you can activate or deactivate the interface
- **Name** – Here you define the name of the interface.
- **Description** – Here you can define an individual description for the interface.
- **Reference** – The reference is automatically generated and defined by the software.
- **Comment** – Here you can define an individual comment.

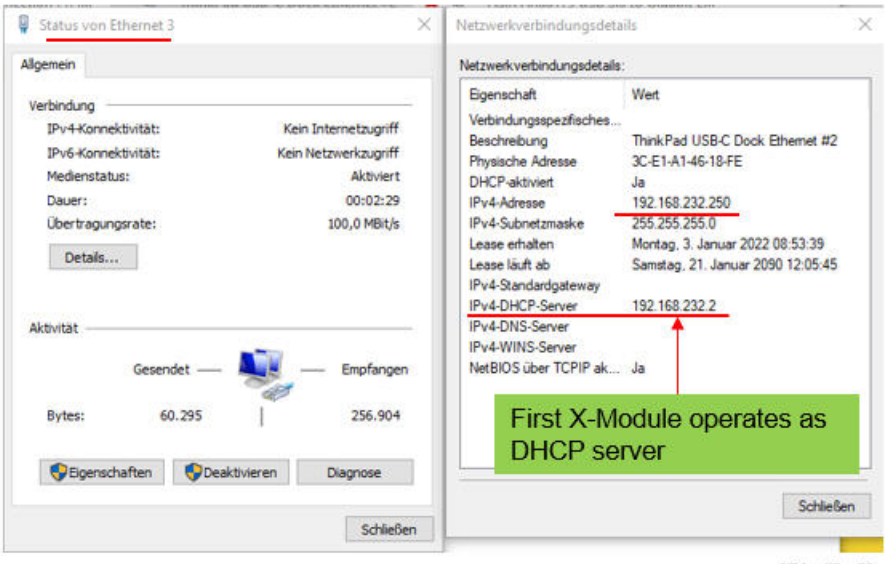
6.2 Ethernet hardware tab sheet



Ethernet hardware tab sheet [SI_42_2]

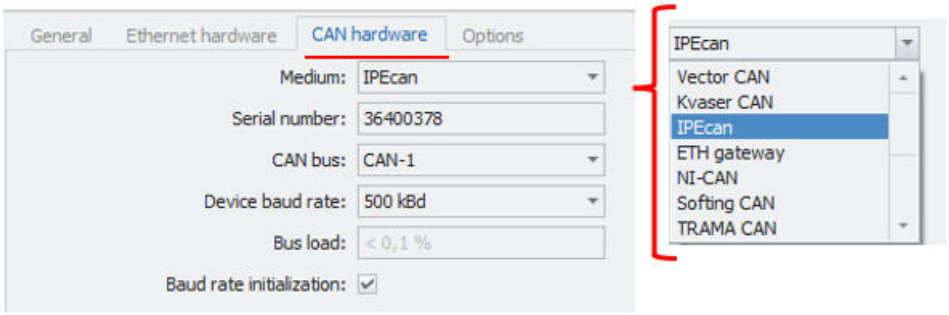
- IP address range — The default IP address range for the X-modules is defined as 192.168.232.1 to 192.168.232.40. However, in some cases the company Ethernet network settings can require a different IP address range which can be modified in the PlugIn OPTIONS discussed in section 7.1.
- Network interface — Here, the name of the network interface of the computer or data logger is indicated. This information is indicating on with Ethernet port the modules are connected. The fist X-Module is working as a DHCP server and assigns the right IP address matching to the module factory settings.
- X-LINK load — This is a calculated statistical value indication how much data is running over the Ethernet interface.
- Internal time synchronization — The PTP time synchronization master is installed together with the PlugIn to synchronize the time between modules. However, in some cases external time synchronization might be required. 0,2cm
- Default send interval — The Ethernet communication support block data transfer with is sending data every 10 ms which is equal to 100 Hz. However if higher block data transfer is required it can be deactivated. In this case the block data transfer is every 2 ms with is equal 500 Hz.

The X-module assign an IP-address to the LAN port of your computer. The X-module operate as a DHCP server. However, if your computer requires a different IP-address range because IT policies you can change the IP-address range of the modules in the X-PlugIn settings.



Computer Network settings [SI_42_3]

6.3 CAN hardware tab sheet

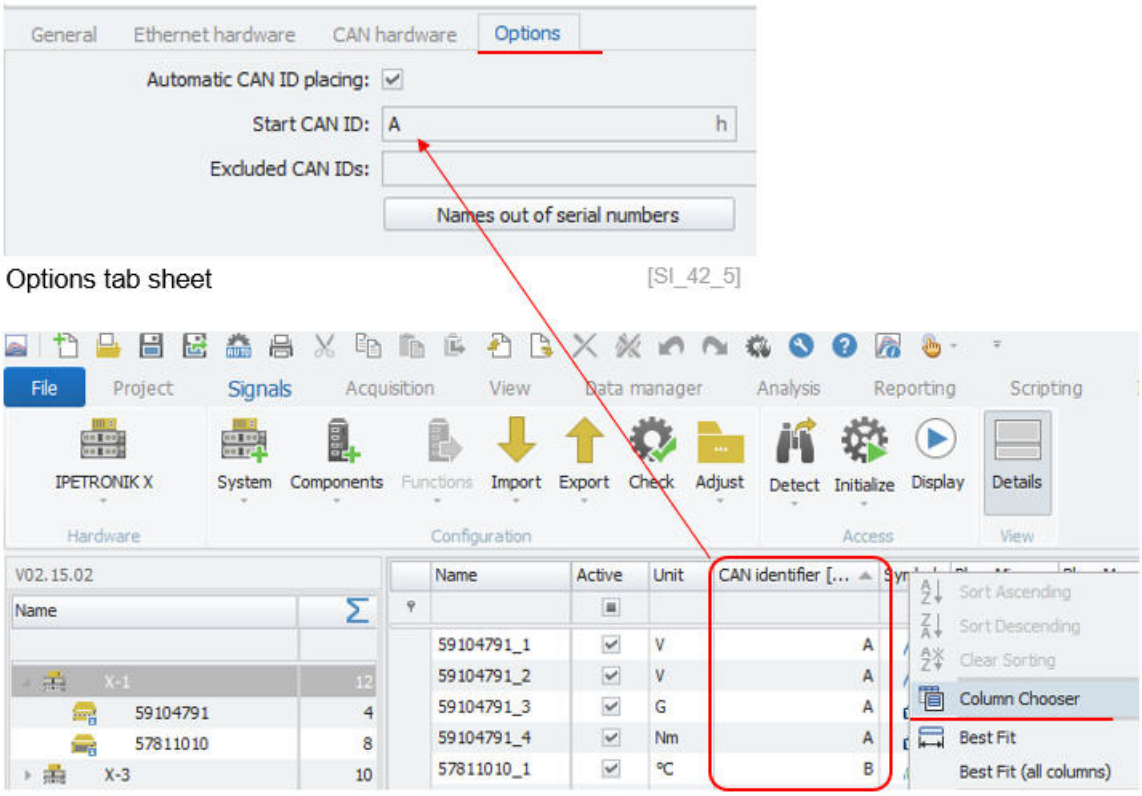


CAN hardware tab sheet [SI_42_4]

- **Medium** — The automatic hardware detect function will identify all CAN interfaces implemented in the CAN server. A list of all supported CAN interfaces id provided below.
- **Serial number** — The automatic CAN hardware detect process is also identifying the serial number of the CAN interface devices. In the cases you start a dry configuration without any hardware available you can type in the serial number manually of the hardware which will be used.
- **CAN bus** — When the device has more than one CAN interface the CAN server will identify on which CAN port number where the module are connected.
- **Device baud rate** — The factory settings for the M-Modules have a CAN bus baud rate of 500 K Baud. However depending on cable length, the baud rate can be set to lower values. 0,2cm
- **Bus load** — The bus load is a calculated statistical value by the CAN server, indication how much data is running over the CAN bus.
- **Baud rate initialization** — Here you can define the if the Baud rate configured above will be initialized to the modules.

Additional CAN interface settings like supported CAN card vendors, scanning baud rates, CAN ID placing etc. . . can be configured in the PlugIn OPTIONS discussed in section 7.5.

6.4 Options tab sheet



- **Automatic CAN ID placing** — With this check box you can define where the software will assign automatically the CAN IDs starting from the CAN ID defined in the box below.
- **Start CAN ID** — Here you define the first CAN ID to start the automatic placing. The start CAN ID can be displayed in a hex, decimal or in the binary format. The CAN ID range 640 . . . 767 is used internally by the modules and will be skipped in the CAN ID placing routine. With the column chooser function you can the CAN IDs information to the channel grid to display the software assigned IDs.
- **Excluded CAN IDs** — Here you can load a DBC file and exclude CAN IDs from the automatic generation process. This is particularly useful in the case the CAN measurement modules are integrated to another CAN bus data stream to ensure that there is no overlap of the CAN IDs from different CAN busses.
- **Name out of Serial numbers** — With this function all channel names are generated automatically considering the module serial number followed by an incrementing index for the channel number.

6.5 CAN- and X-module configuration

The following CAN modules are supported in the IPETRONIK X PlugIn.

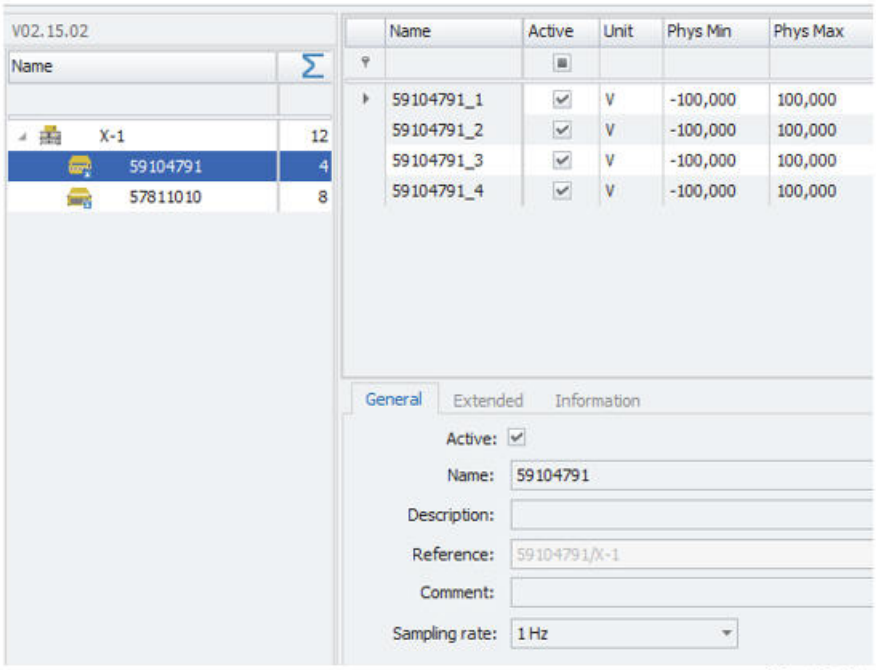
M2 and M-Module	Base Type Number	Firmware
M-SENS2	587	04.15.00
M-SENS2 DSP	587	04.15.00
M-SENS2 250Hz	591	04.15.00
M-SENS2 250Hz DSP	591	04.15.00
M-SENS	561	04.15.00
M-SENS DSP	561	04.15.00
M-SENS 8	567	04.15.00
M-SENS 8 DSP	567	04.15.00
M-SENS 8plus	568	04.15.00
M-SENS 8plus DSP	568	04.15.00
M-THERMO2	578	04.13.00
M-THERMO2 HV	557	04.13.00
M-THERMO2 u	579	04.13.00
M-RTD2	581	04.10.03
M-UNI2	584	04.13.00
Mc-THERMO	573	04.09.03
M-THERMO	560	04.09.51
M-THERMO 16	566	04.09.50
M-THERMO T	569	04.09.50
M-THERMO 16T	575	04.09.50
µ-THERMO	563	04.09.50
M-CNT2	586	04.13.01
M-FRQ	562	04.09.00
CANpressure	595	04.10.00
M-THERMO96 16	593	01.03.01
SIM-STG	519	04.10.00
M-Flow	519	04.10.00

The following X modules are supported in the IPETRONIK X PlugIn.

X module	Base Type Number	Firmware
Mx-SENS(2)8	911	02.15.03
Mx-SENS2-4	916	02.15.03
Sx-STG	920	02.15.06
Mx-STG2 6	912	02.15.04
Mx-SENS2-4 FAST	917	02.15.03

On CAN and X-module level the following configuration tab sheets are provided.

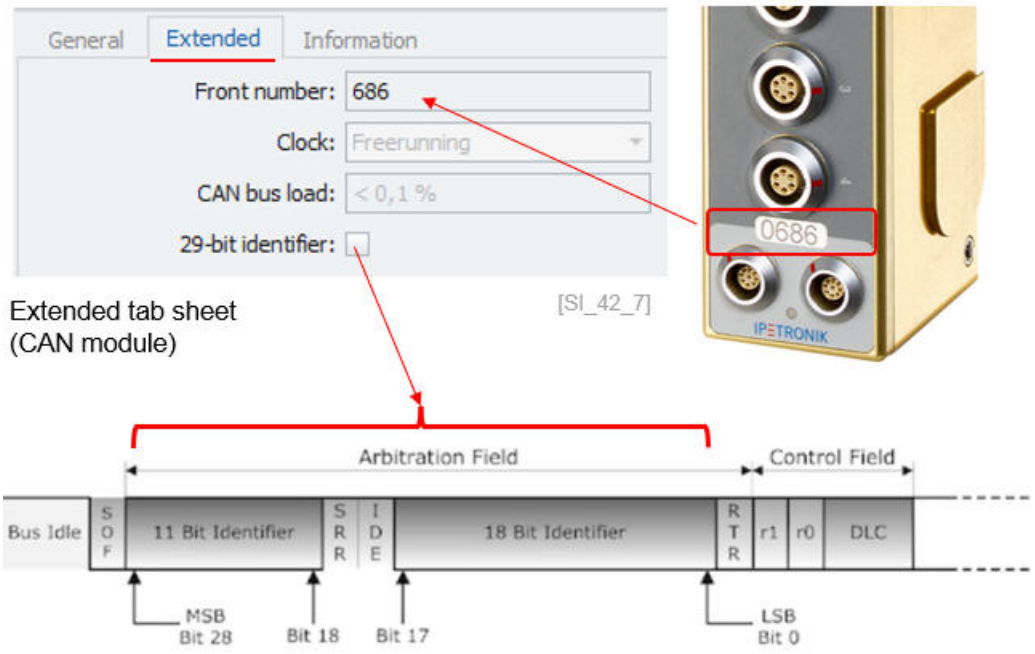
6.5.1 General tab sheet



General tab sheet [SI_42_6]

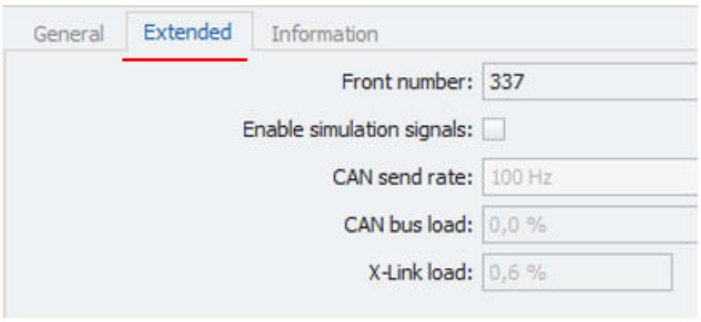
- **Active** — Here you can activate or deactivate the module.
- **Name** — Here you define the name of the module. The default name is based on the serial number.
- **Description** — Here you can define an individual description for the module
- **Reference** — The reference is automatically generated and defined by the software. 0,2cm
- **Comment** — Here you can define an individual comment.
- **Sampling rate** — In this drop down box, the sample rate for the module can be defined. The sample rate is set for the entire module. The lowest sample rate is 1 Hz and the fast sample rate is depending on the module type and can reach up to 5 kHz for the SIM STG module.

6.5.2 Extended tab sheet – CAN modules



- **Front number** — In this field the device front number is displayed. When you run a detect function the front number is automatically detected and extracted from the serial number. The serial number is composed of the front number and the device type number.
- **Clock** — The default configuration is the Freerunning mode. However, a synchronized mode is supported too, where the first module operate as a Master and all the other modules as Slaves. The clock can only be changed in the PlugIn settings discussed in the section 7.3.
- **CAN bus load** — This is a statical value calculated by the PlugIn. Higher sample rates will increase the bus load.
- **29-bit identifier** — With this check box you can activate the extended CAN identifier. The standard CAN identifier is 11 bit. 0,2cm

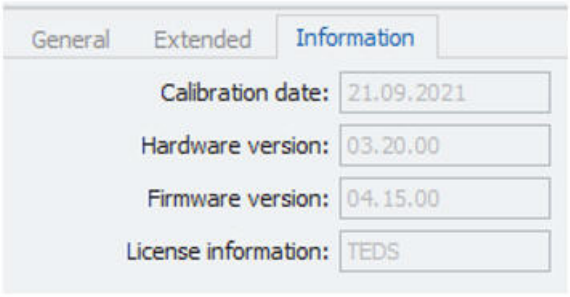
6.5.3 Extended tab sheet – X modules



Extended tab sheet (X-module) [SI_42_8]

- **Front number** — In this field the device front number is displayed. When you run a detect function the front number is automatically detected and extracted from the serial number. The serial number is composed of the front number and the device type number.
- **Enable simulation signals** — This check box will generate a sawtooth simulation signal, in the case on real sensor signal is received on the analog input.
- **CAN send rate** — This function is only available when on channel level a CAN output is configured. This function will be discussed in section XXX.
- **CAN bus load** — This is a statical value calculated by the PlugIn. Higher sample rates will increase the CAN bus load.
- **X-Link load** — This is a statical value calculated by the PlugIn. Higher sample rates will increase the Ethernet bus load.

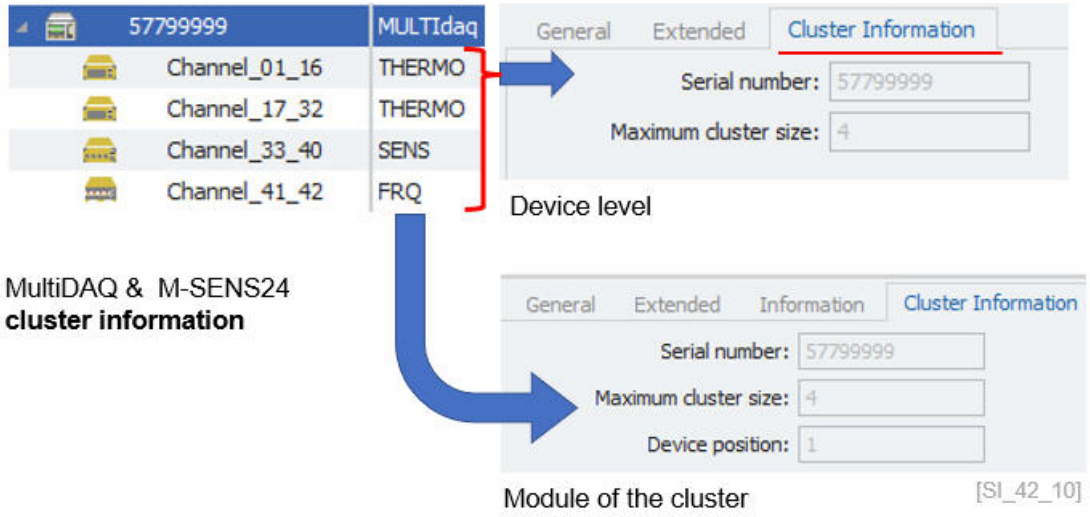
6.5.4 Information tab sheet



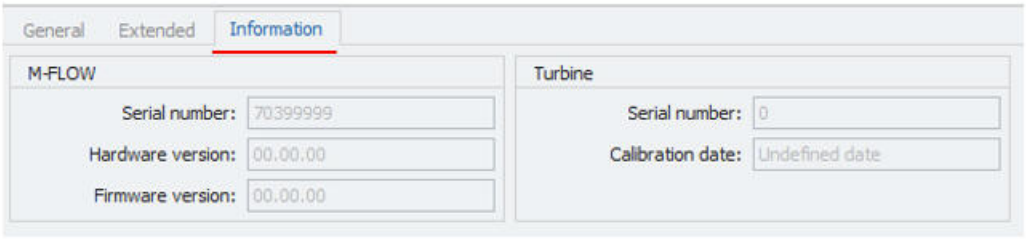
Information tab sheet [SI_42_9]

- **Calibration date** — In this field the last calibration date is indicated.
- **Hardware version** — In this field the hardware version of is indicated.
- **Firmware version** — In this field the current firmware version is indicated. The firmware can be updated as discussed above in section 5.1.3.
- **License information** — Some modules support additional licensing functions like the TEDS functionality, additional DSP filters and the FAST sample rates. These licenses are delivered from the factory side as part of the order. However. it is also possible to update modules after purchase with new license. 0,2cm

Some modules have additional information. The SIM STG and all Ethernet X-modules provide information about the FPGA version. The MultiDAQ and M-SENS24 indicate a Cluster information with includes the serial number and the size of the cluster. On module level inside the cluster additional information about the cluster position and the sub-serial number of the individual device is indicated.



The M-FLOW device provides information about the M-FLOW signal conditioning unit and separate information about the flow turbine. A firmware updated is not supported via X-UPDATE function of the PlugIn.



6.5.5 Module license update

In order to perform a license update, you need to detect the module in the first place. After that you add the module specific license key into the IPEmotion license dialog. With the assign function in the license dialog the new license key is activated on the module. After a new hardware detection the new licenses information is displayed.

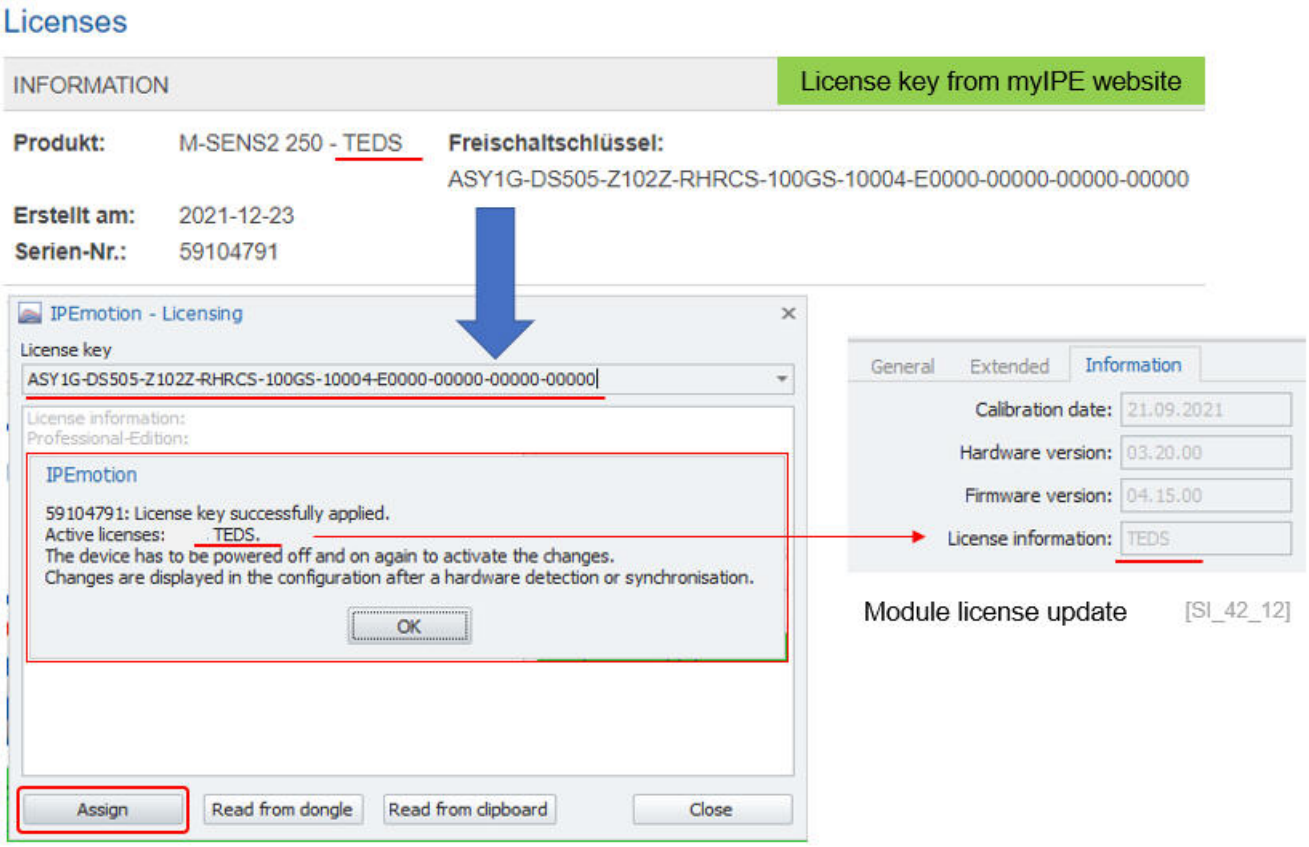
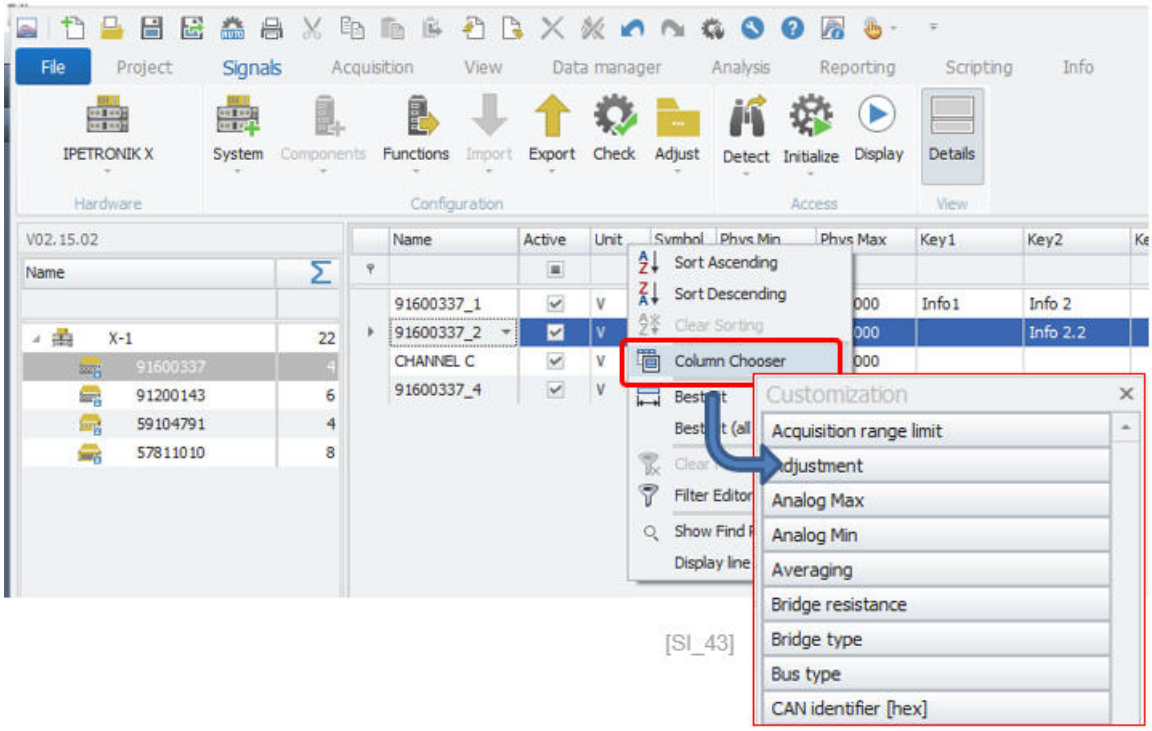


TABELLE Module mit Lizenzen

6.6 Channel configuration

6.6.1 Column chooser in the channel grid

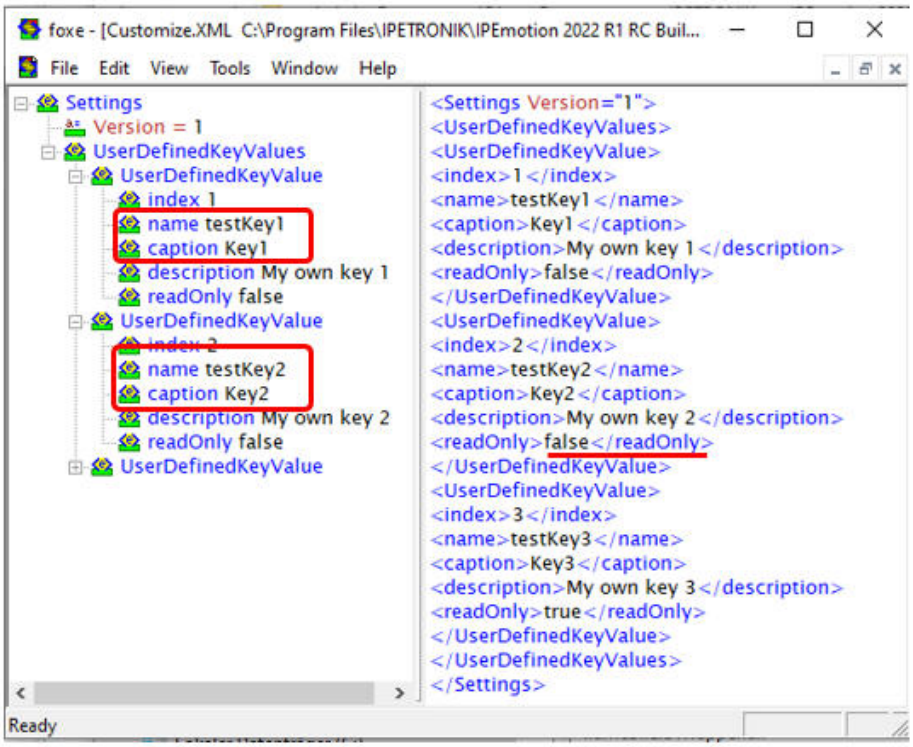
In the channel grid head line you can access a context menu to add additional columns to your channel grid. The available columns are depending on the PlugIn.



[SI_43]

You can add your own columns into your channel grid. In order to add individual columns you need to create in the installation directory a new xml file called: **Customize.XML.**

- C:\Program Files\IPETRONIK**IPEmotion 2022 R2****Customize.xml**

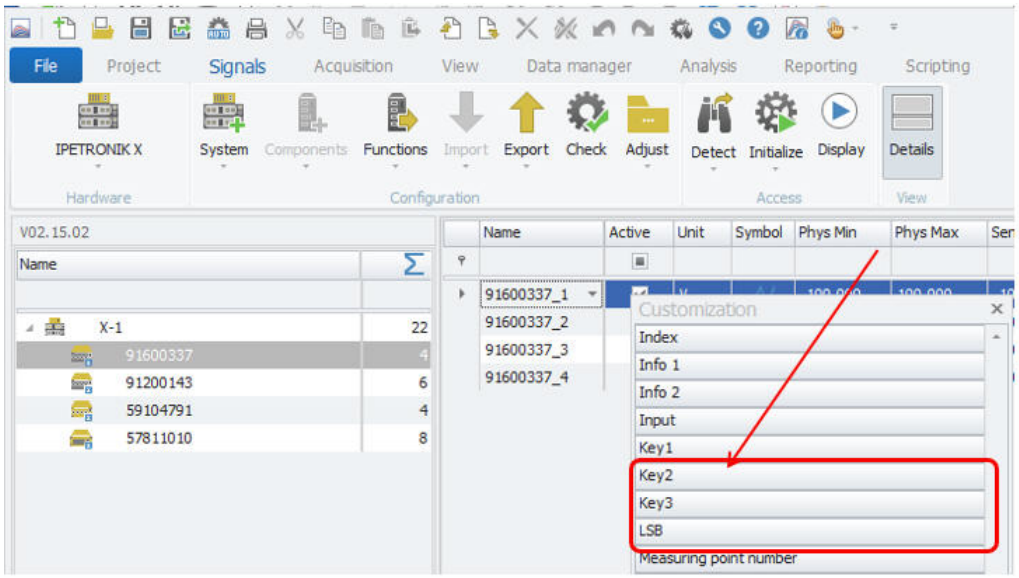


Customize.XML – User defined key fields. [SI_45]

With the ”readOnly” status (true/false) you define if the field can be edited though the channel grid. XML Code to be included in the **customize.xml** file:

```
<Settings Version="1">
<UserDefinedKeyValues>
<UserDefinedKeyValue>
<index>1</index>
<name>testKey1</name>
<caption>Key1</caption>
<description>My own key 1</description>
<readOnly>false</readOnly>
</UserDefinedKeyValue>
<UserDefinedKeyValue>
<index>2</index>
<name>testKey2</name>
<caption>Key2</caption>
<description>My own key 2</description>
<readOnly>false</readOnly>
</UserDefinedKeyValue>
<UserDefinedKeyValue>
<index>3</index>
<name>testKey3</name>
<caption>Key3</caption>
<description>My own key 3</description>
<readOnly>true</readOnly>
</UserDefinedKeyValue>
</UserDefinedKeyValues>
</Settings>
```

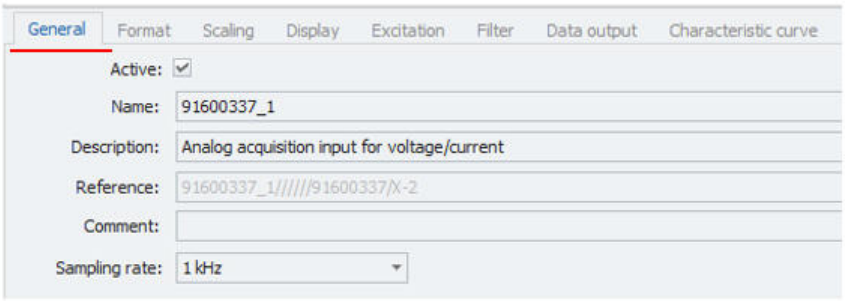
The following screenshot shows a channel grid which includes 3 individually defined ”KEY fields”.



Drag & drop key fields into the column grid. [SI_46]

6.6.2 General tab sheet

This tab sheet covers general channel settings

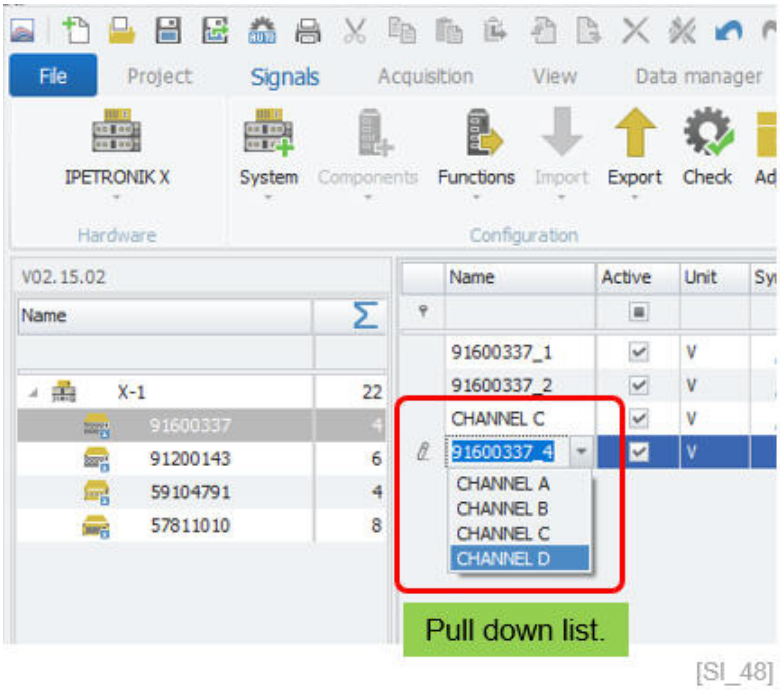


General channel tan sheet [SI_47]

- Active — Checkbox to activate or deactivate a channel
- Name — Default name - can be changed to individual names
- Description — Default description - can be changed to any individual description
- Reference — Is automatically generated and very useful to check where the channel is linked to.
- Comment — Enter a comment.
- Sampling rate — Select from drop down list the module sample rate.

6.6.3 Defining list box entries of channel names

For the channel name you can also define a pull down menu.



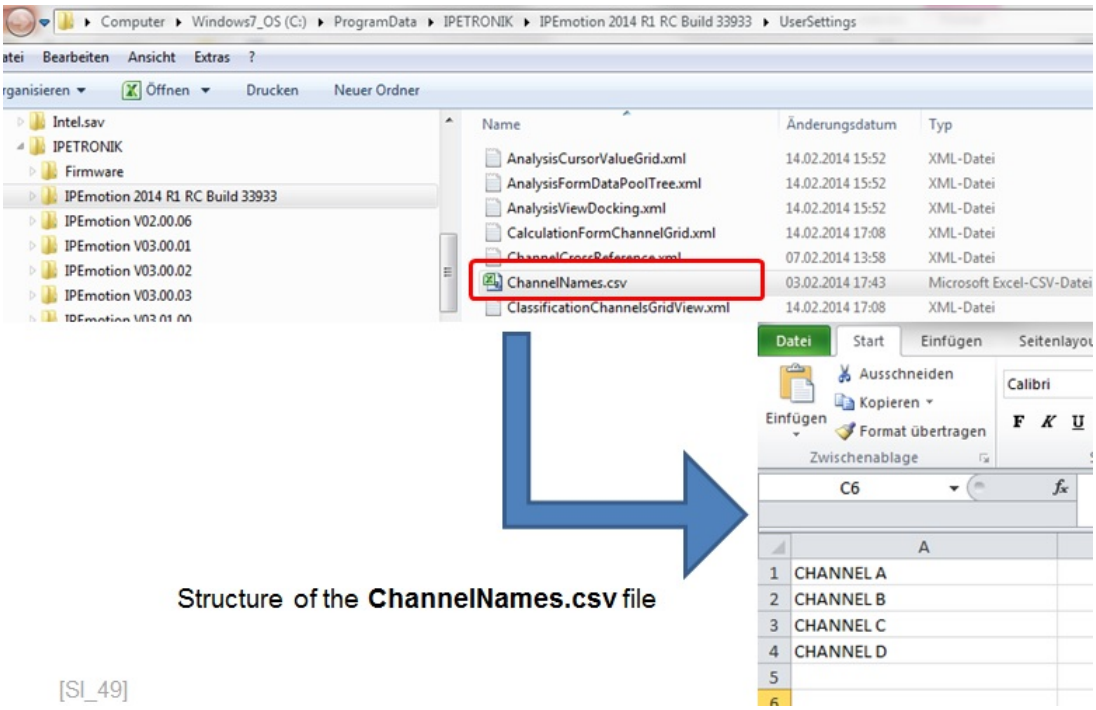
[SI_48]

The entries of the pull down menu are stored in a CSV file with the name (ChannelNames.csv) in the following user settings directory.

- C:\ProgramData\IPETRONIK**IPEmotion 2022 R2**\UserSettings

6.6.4 Format tab sheet

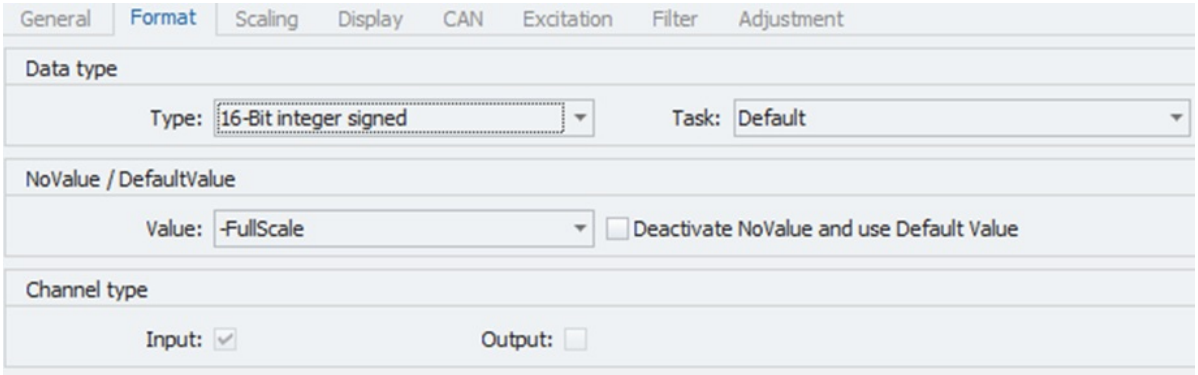
The FORMAT tab sheet is only visible for users who activate this function in OPTIONS >Expert mode >Extended tabs in chapter **??**. In the Format tab sheet we can configure a couple of functions which are usually only relevant for expert users. The different configuration functions are explained below.



[SI_49]

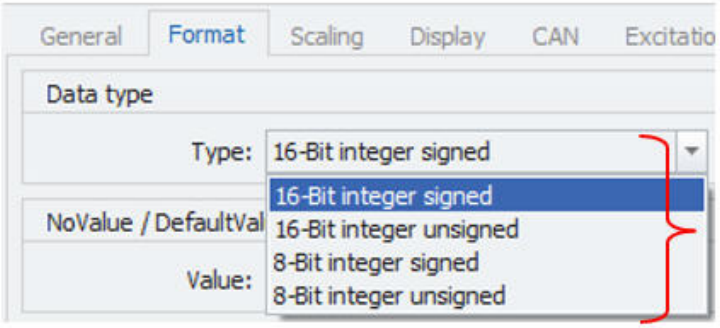
- Data type

  This refers to the data format (resolution) of the measurements. Depending on the module / instrument, sometimes different formats are supported. On most of the instruments, it is not possible to change the configuration of the data type. They always transmit data in the same format. For IPETRONIK modules the signed or unsigned format is important. The 8 bit format is still included because of historic reasons.



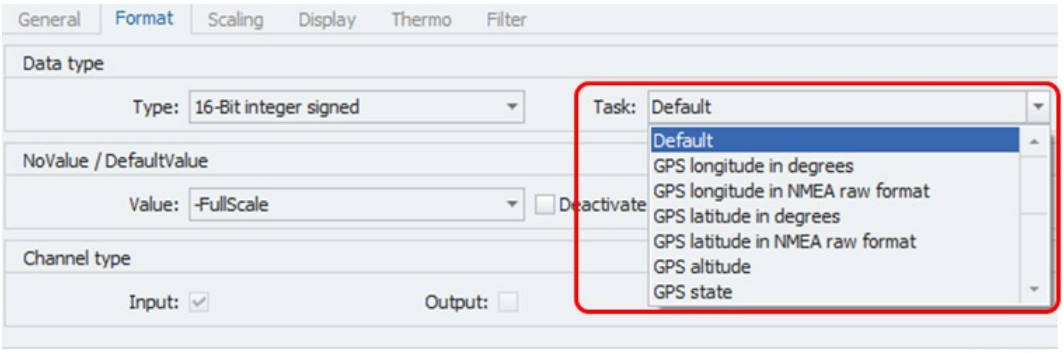
[SI_50]

Some modules support a change in the data type format from a drop downlist as indicated below.



Change data type [SI_51]

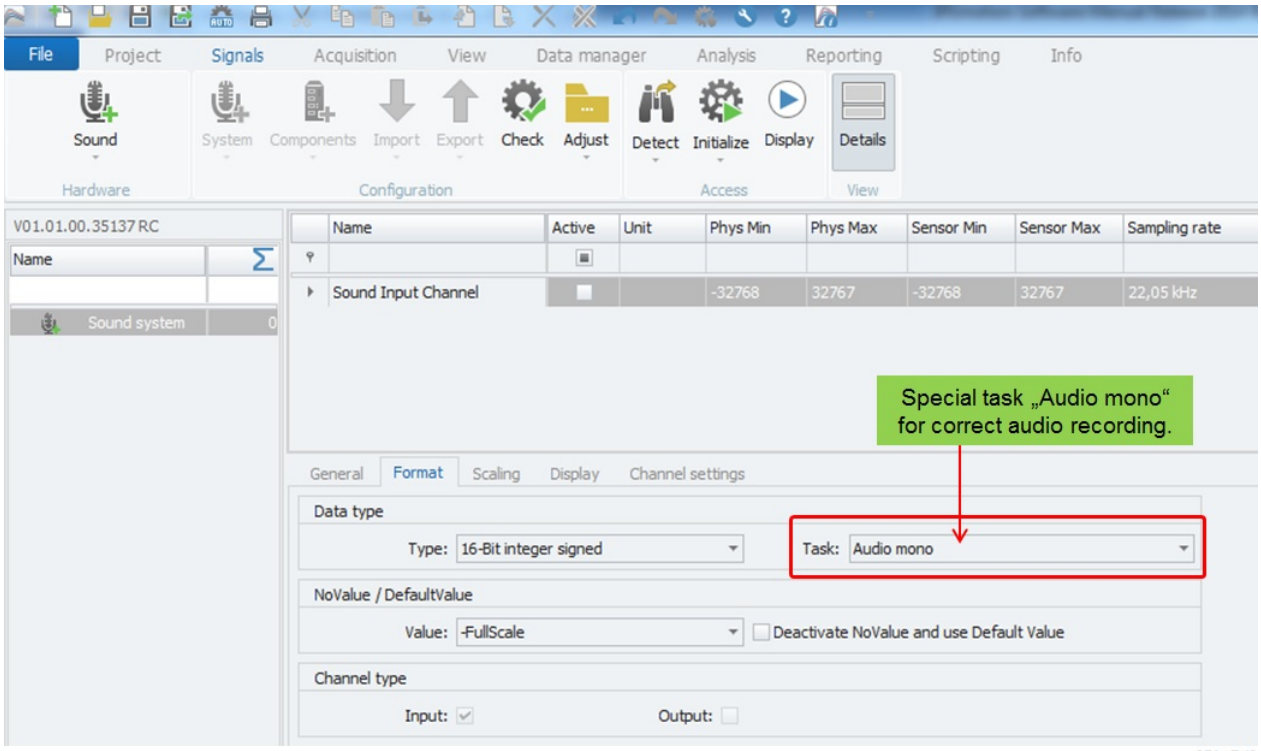
- **Task** — The task is a very special setting developed for some specific PlugIns
- **Task: GPS Recording** — The settings for a special task are needed for the GPS signals. This sensor sends the NMEA protocol in a special format and in order to convert this signal to a standard format which can be used by IPEmotion, the measurement channels need a task configuration for longitude, latitude etc... to get a correct data dispaly in the map instrument in the ANALYSIS works pace. A correct configuration of the task is also required when you would like to save or export data in the GPX format. The coordinats longitude, latitude and altitude are only correctly interpreted in the GPX export when the corresponding task is defined. See also GPX export in chapter DATA MANAGER **??**.



[SI_52]

- Task: Audio Recording

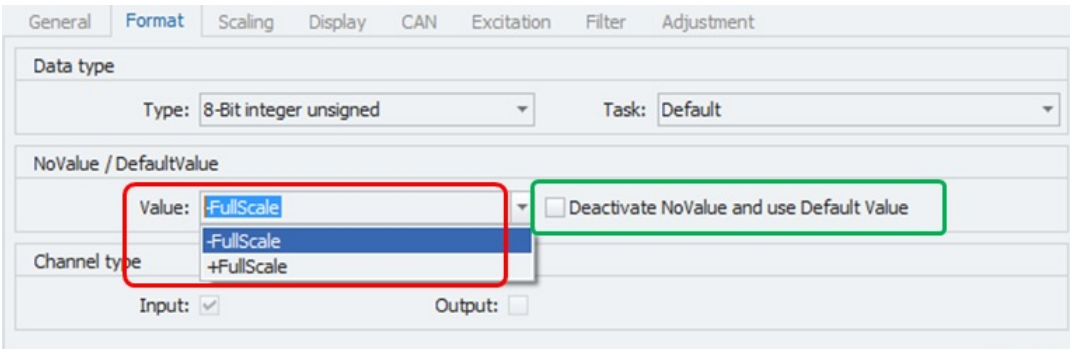
When you like to record audio e.g. via an MX-SENS2 4 fast module or over the PC-Sound PlugIn you should check the setting of the Task which should configured to ”Audio mono”.



[SI_54]

- NoValue

This configuration is important for all users who would like to see a certain behavior when NO measurements received in IPEmotion. The default configuration is that No Values are recorded in the data file. They are indicated as NoValue in the DATA MANAGER. In the Yt- chart in the ANALYSIS work space you will see missing data points in the graph. The software will always store No VALUE in the data file irrespectively what you select from the drop down box. In the data file NoValue is stored and in the diagrams you will see missing data points.



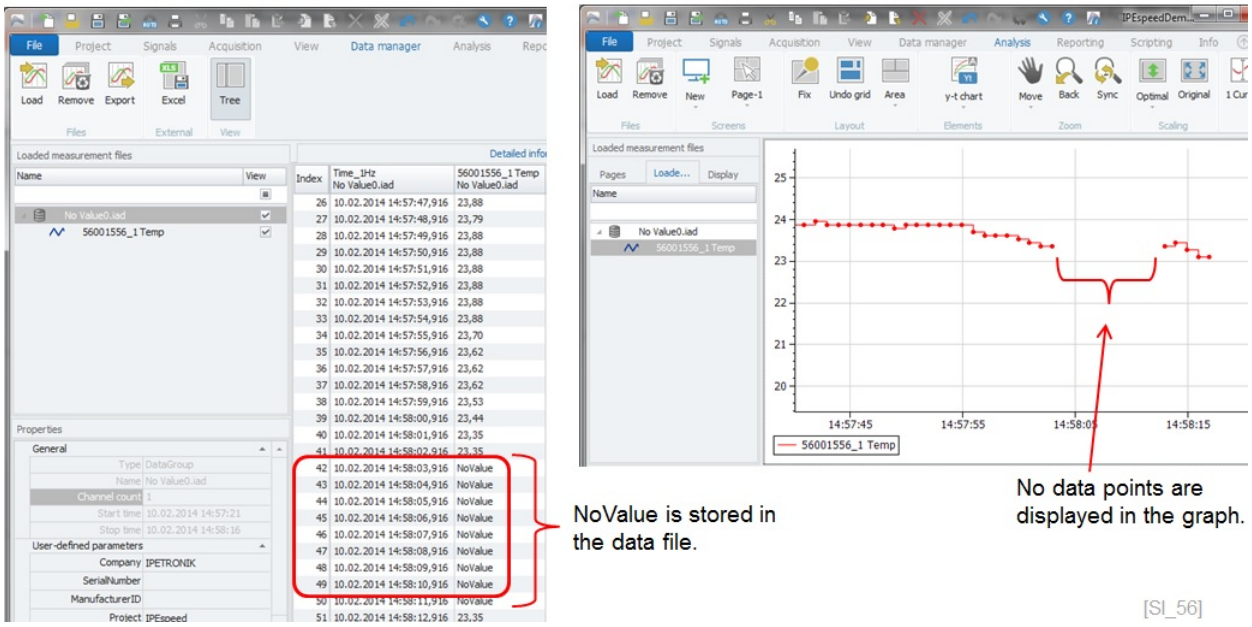
Drop down selection has no impact when check box „Deactivate Novalue…“ is deactivated. [SI_55]

- DefaultValue

Another configuration option is a check box to enables the DefaultValue. With this check box you change the storage and display behavior when no measurements are received. With the check box you can show and store + FullScale, − FullScale or NULL as a numerical value. You can only select NULL if you have a signed +− measurement range) data format. An unsigned measurement is only covering positive measurements.

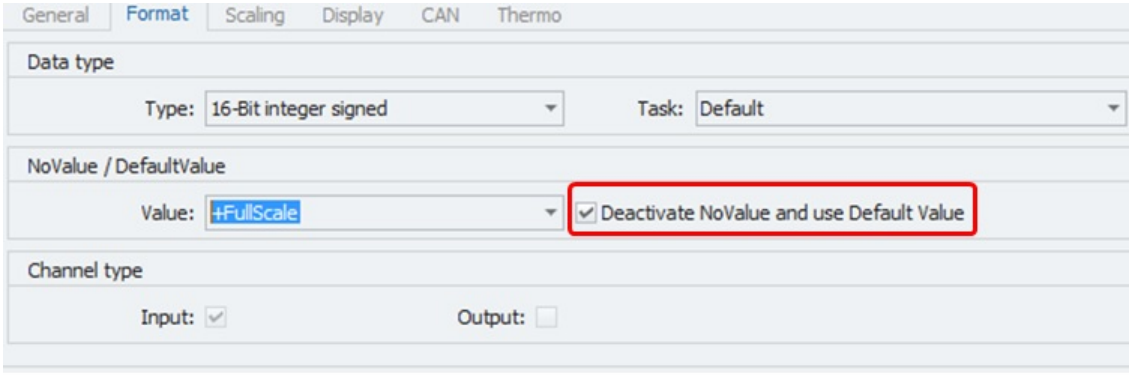
DATA MANAGER

ANALYSIS



[SI_56]

The NoValue configuration also has an impact on the data display in the VIEW work area. As the screen shot below indicates. When the check box ”Deactivate NoValue and use Default Value” is not activated the instrument will show always Novalue.

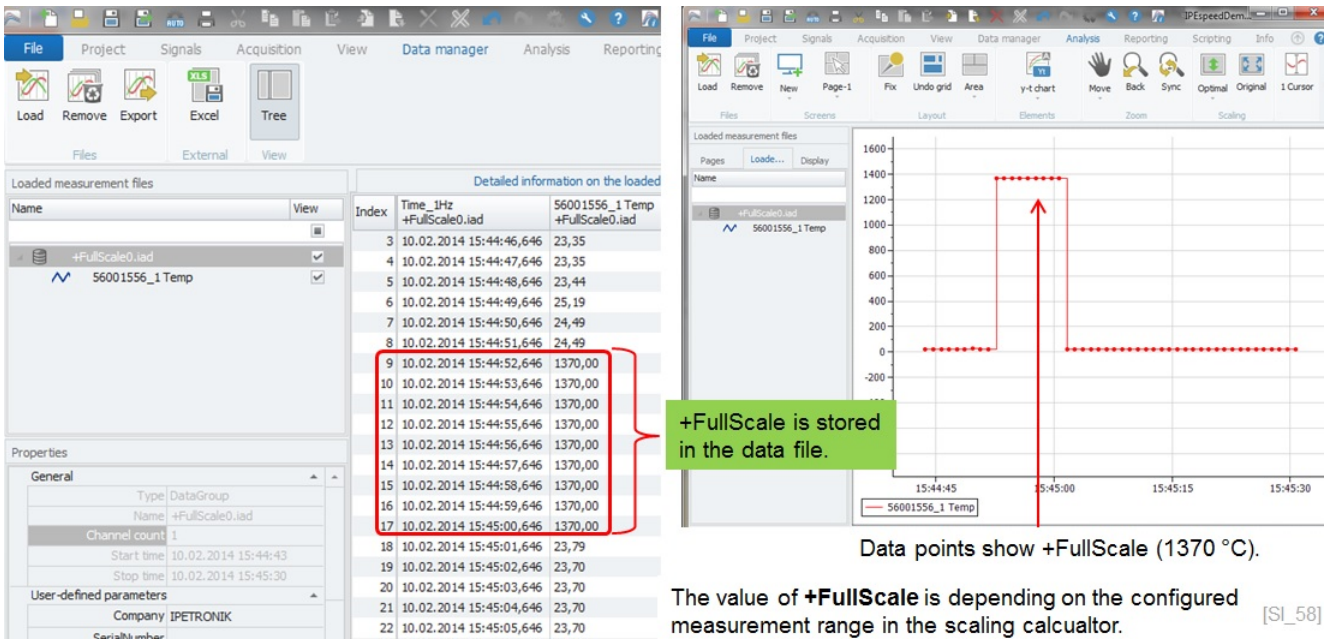


Filter has an impact when check box „Deactivate NoValue…“ is activated.
In this example +FullScale of the measurement range will be stored.

[SI_57]

However if the check box ”Deactivate NoValue and use Default Value” is activated you will enable the the list box entries and the instrument will show the selected values for:

- + Full Scale
- - Full Scale
- Null

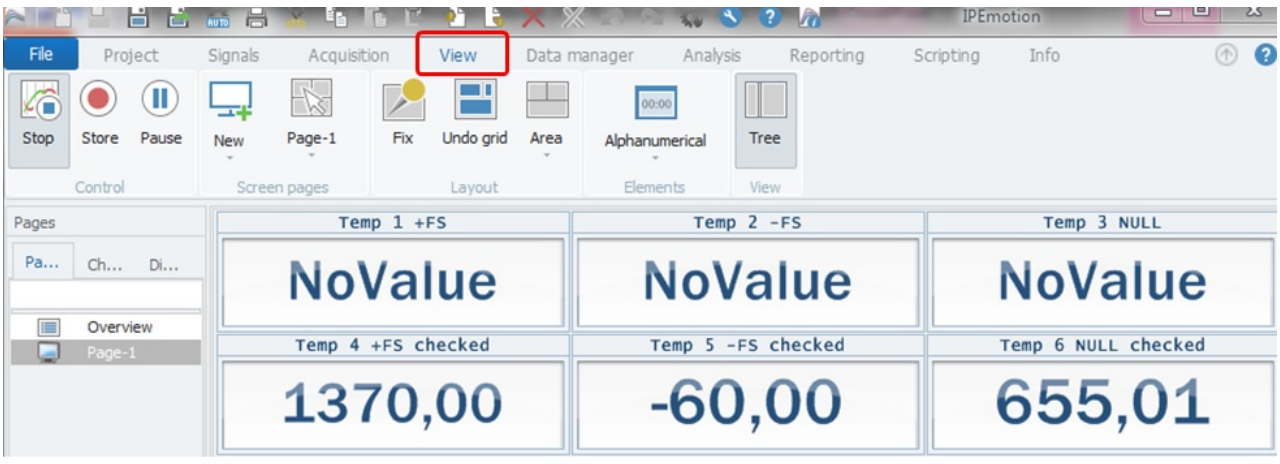


Data points show +FullScale (1370 °C).

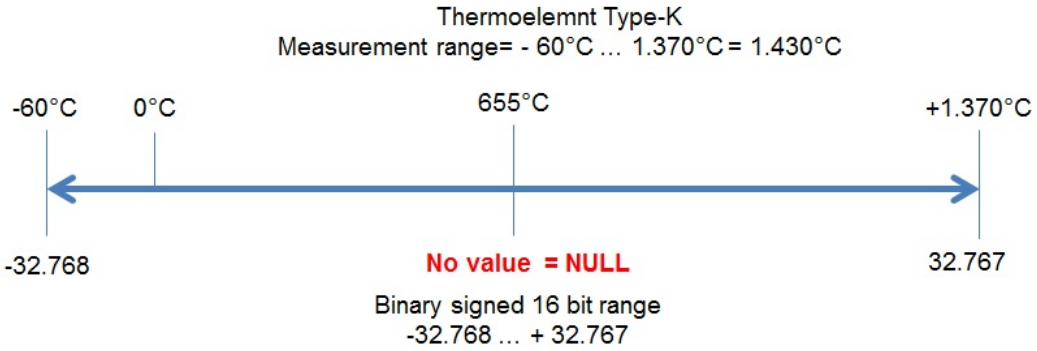
The value of **+FullScale** is depending on the configured measurement range in the scaling calcualtor. [SI_58]

- Default Value Null

The DefaultValue (NULL) is related to the Null value of the binary measurement range. If you select a signed 16bit (2^16 = 65536) measurement range, the temperature signal for the IPETRONIK thermo module is split up between the values -65.536 and +65.536 as the graphic demonstrates below



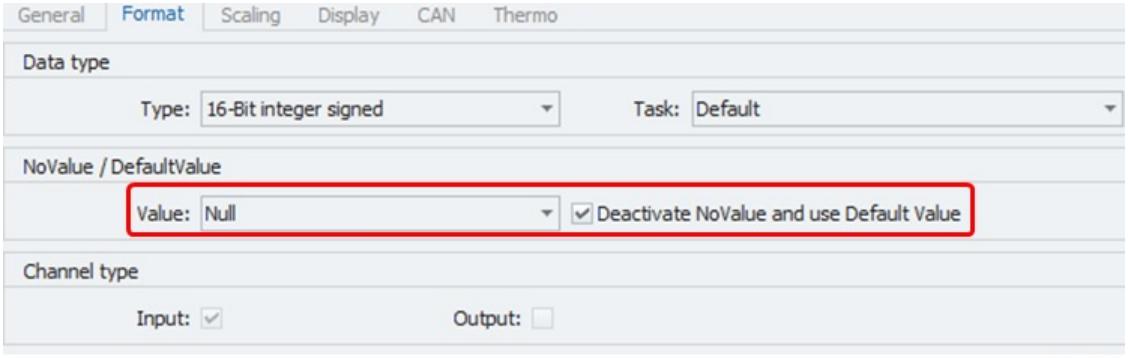
[SI_59]



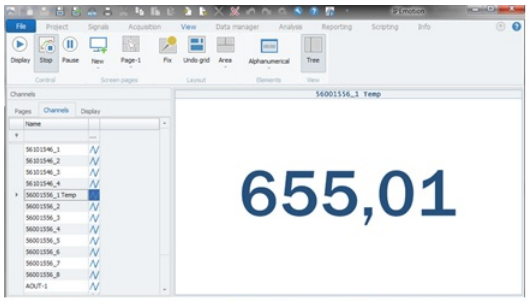
Temp 4 +FS checked	**1370,00 °C**	☑	°C
Temp 5 -FS checked	**-60,00 °C**	☑	°C
Temp 6 NULL checked	**655,01 °C**	☑	°C

[SI_60]

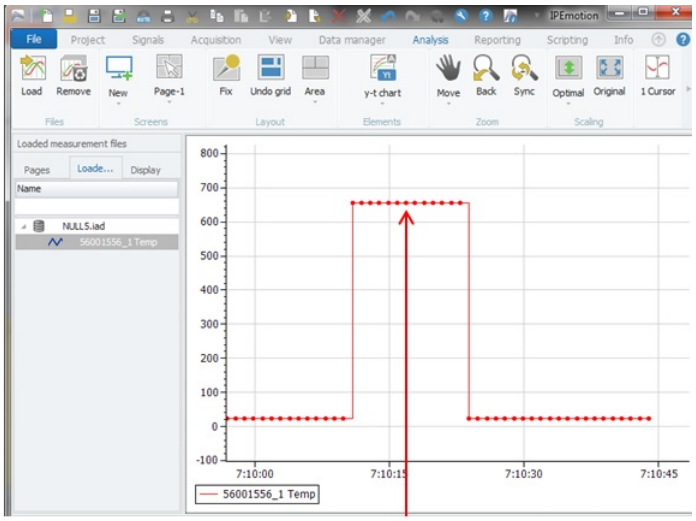
The binary NULL value of this measurement range is $655\,^{\circ}\mathrm{C}$. This value is then indicated to the online instruments and stored in the data file.



[SI_61]



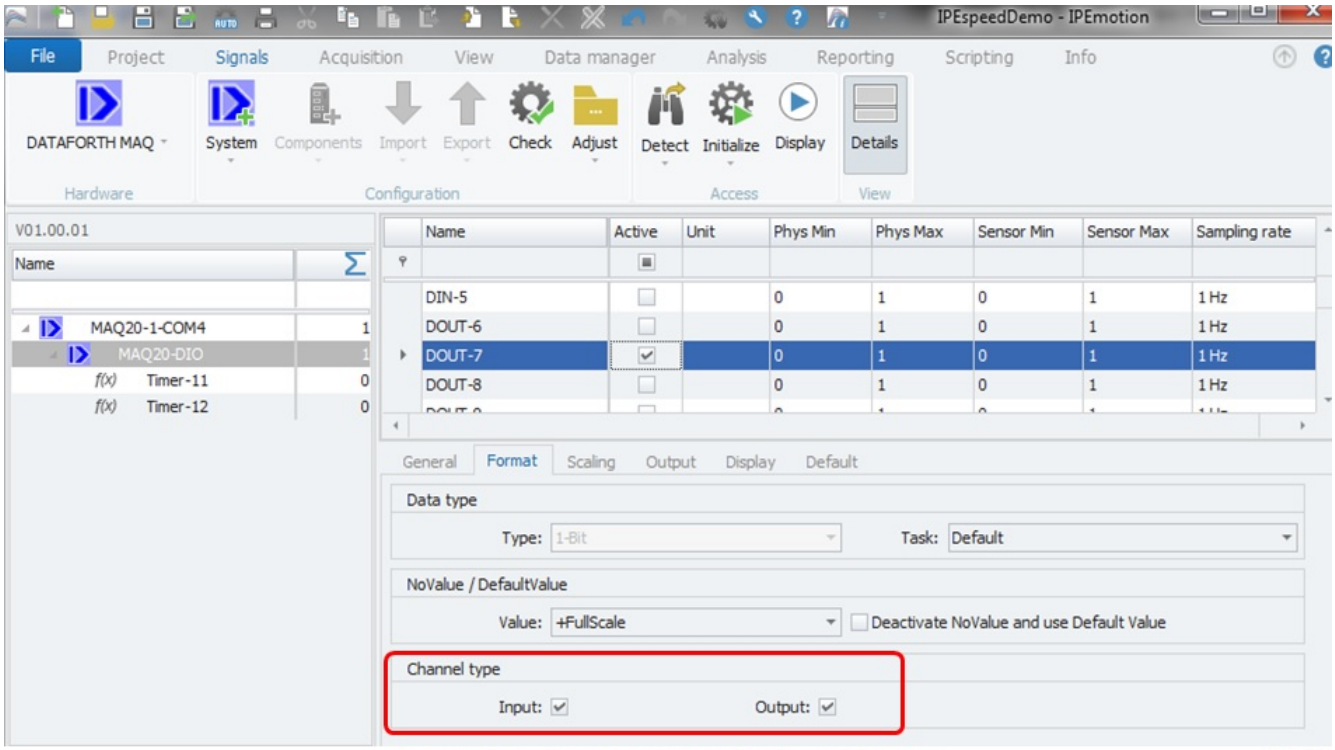
When you select NULL as a NoValue the Thermoelement Type-K will show 655°C.



Data points for NULL show 655 °C. [SI_62]

- **Channel type**

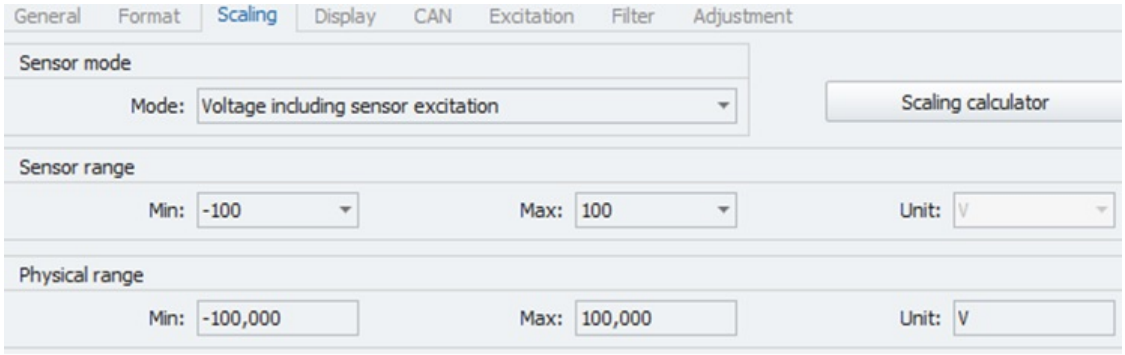
  The channel type indicates the data direction INPUT or OUTPUT. Output channels can be updated through manual entries, through slide controllers or alphanumerical displays in the VIEW work area. Some PlugIns support channels which can be operated as input and output. In digital IOs you will also quite often find the option to change the channel direction input to output or vice a versa through this checkbox



Operated as Input and Output at the same time. [SI_63]

6.7 Channel Scaling – defining ranges and engineering units

IPEmotion Sensor Scaling - How do Analog Sensors Work: `https://youtu.be/7uWNIrpTOAM`

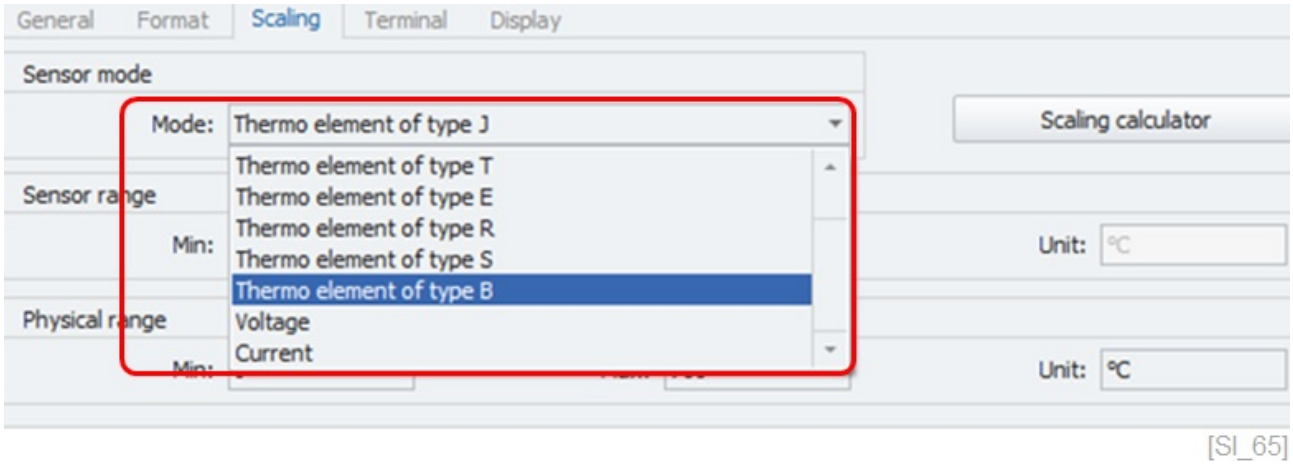


[SI_64]

The basic scaling operations can be defined directly in the scaling tab sheet. The scope of functions depends on PlugIn and IO module type. Some inputs, especially analog inputs, support many different functions and ranges and provide more scaling options.

6.7.1 Sensor mode

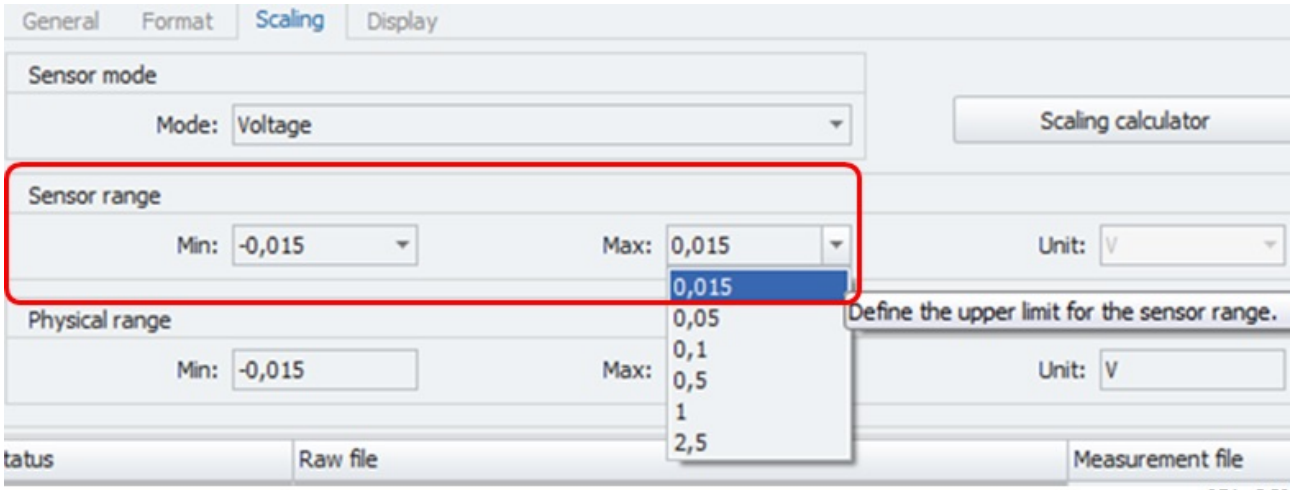
The sensor mode covers the main measurement type, for example Volt or Current, accelerometers (ICP). You select the sensor mode from your drop-down list. In this example, the analog input module supports many different measurements of thermo element, voltage or current. The supported sensor modes are defined by the PlugIn and you can only select the mode which is supported. Many modules only support one static sensor mode.



[SI_65]

6.7.2 Sensor range

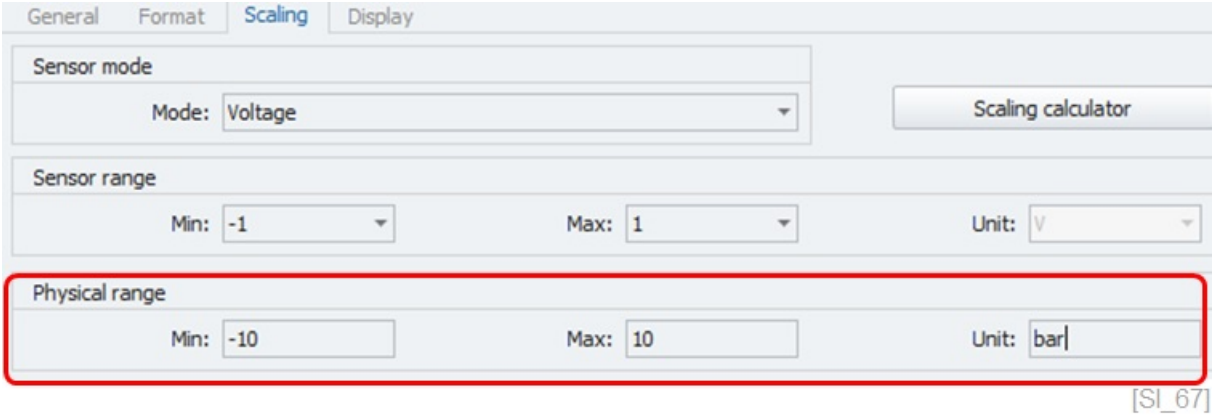
The next configuration option is the sensor measurement range. The range is related to the measurement mode. For thermo elements, the measurement range is redefined and cannot be changed. The available voltage and current measurement ranges depend on the functionality of the analog input. In the example below you can select ranges from 15 mV (0,015V) up to 2,5 Volt. The Unit is automatically linked to the selected measurement mode Voltage >V or current >A or temperature >C and cannot be changed manually. It is defined by the PlugIn developer.


General
Format
Scaling
Display
Sensor mode
Mode:
Voltage
Scaling calculator
Sensor range
Min:
-0,015
Max:
0,015
Unit:
V
0,015
0,05
0,1
0,5
1
2,5
Define the upper limit for the sensor range.
Physical range
Min:
-0,015
Max:
Unit:
V
tatus
Raw file
Measurement file

[SI_66]

6.7.3 Physical range – Engineering units

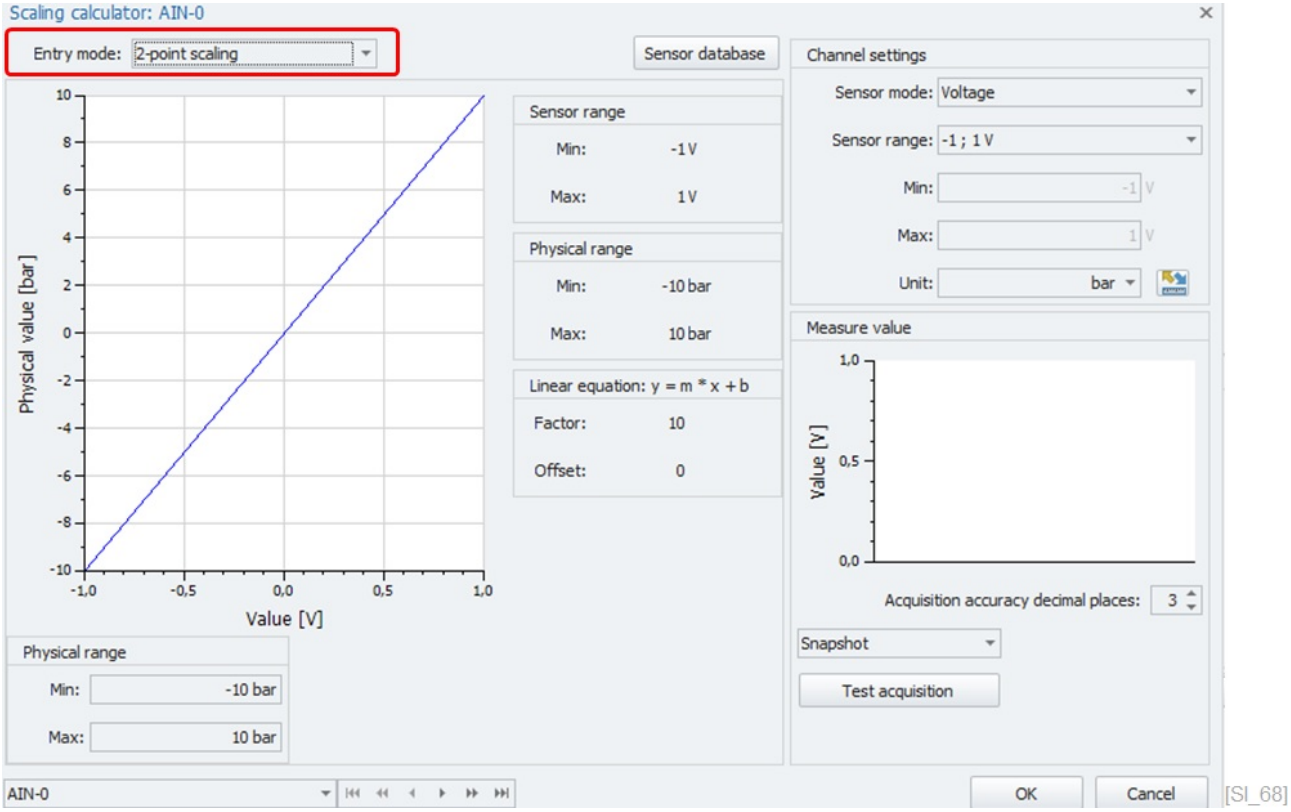
The physical range is related to your engineering units. Here you define into which unit (mm, bar, etc.) the electrical signal is converted.



[SI_67]

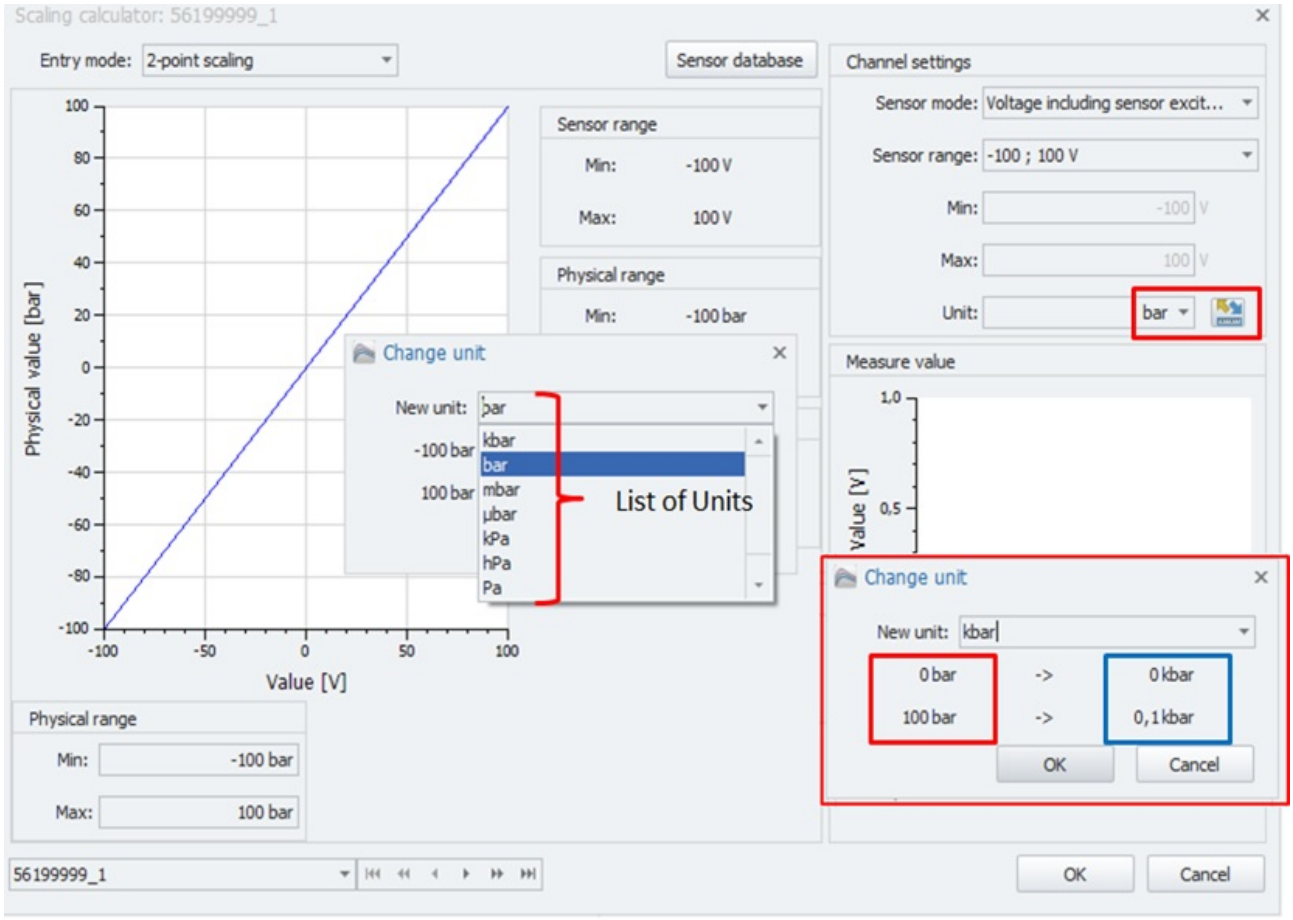
6.8 Scaling calculator – for advanced scaling functions

For advanced scaling functions you can use the scaling calculator. This interface provides many different scaling functions which will be discussed later on.



[SI_68]

- ► Sensor mode — is related to the type of measurement mode as discussed above
- ► Sensor range — is related to the measurement range as discussed above
- ► Unit — To simplify the conversion between engineering units you can use the change unit editor. Switching between units only works within the same engineering unit family like temperatures, pressures, weight, energy, etc.



[SI_69]

Example: Changing **bar** into **kbar**.

The main advantage is that the new engineering unit automatically converts the physical measurement range. As shown in the screenshot, **100 bar** are automatically converted to **0,1 kbar**. This conversion also works across different metric standards.

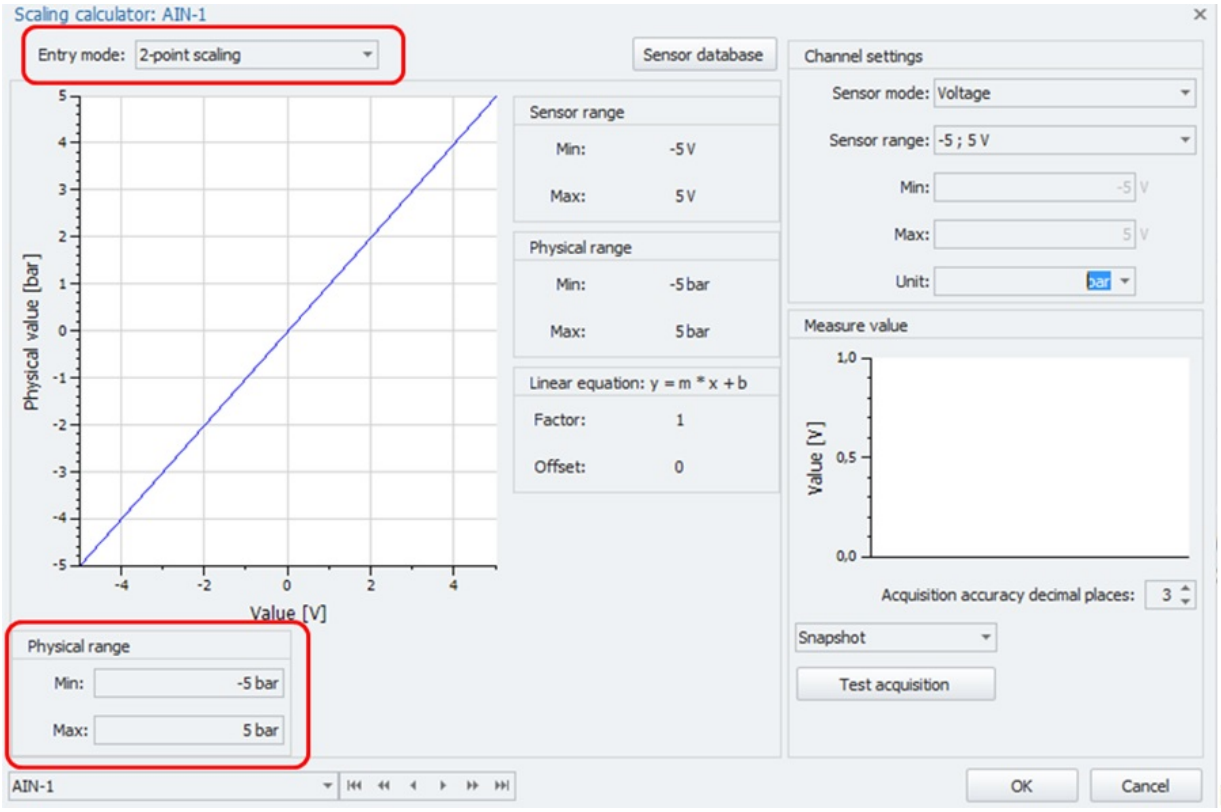
Changing for example:

- Pressure Bar >kbar >mbar >psi >etc.
- Temperature C >K >F

An overview of all supported engineering units can be found in the OPTIONS chapter **??**.

6.8.1 2-point scaling

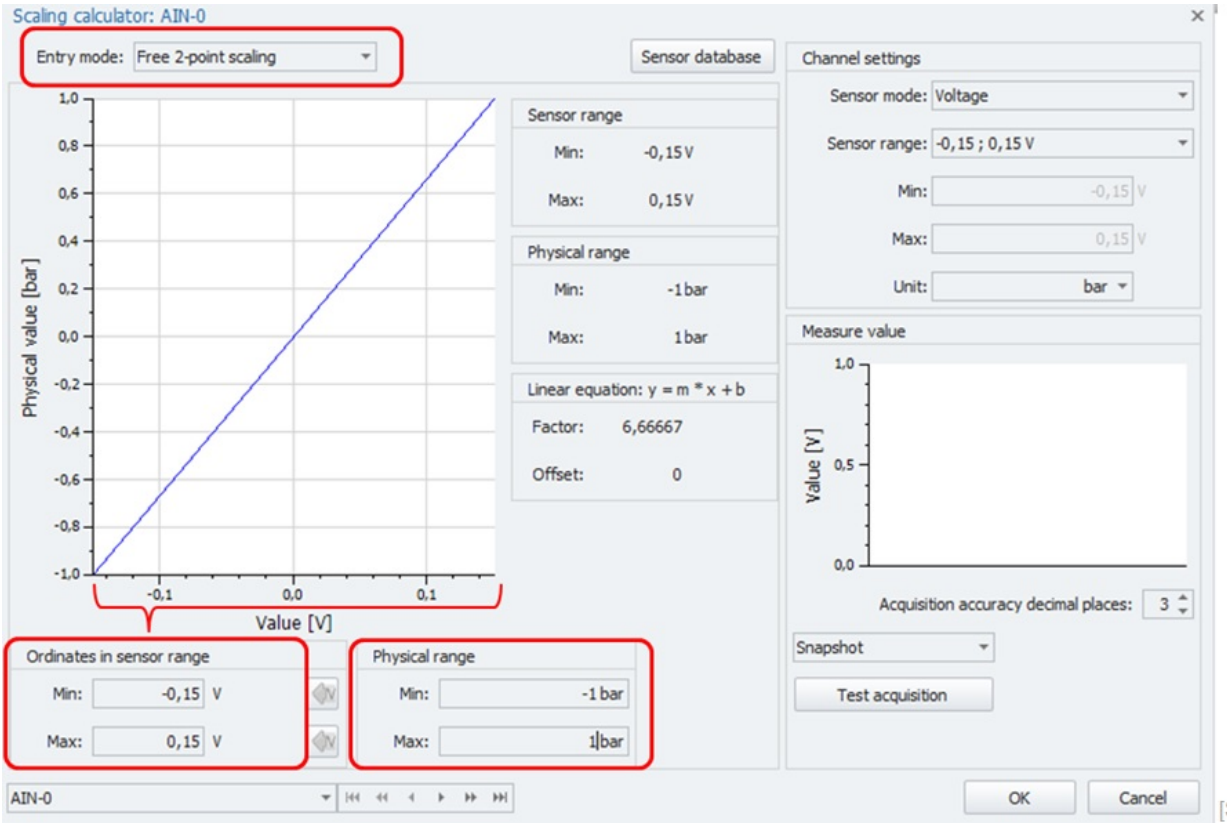
This is a classical scaling configuration using two points, usually the MIN and MAX value of the physical range of the sensor. The scaling information is included in the data sheet / calibration sheet of the sensor.



[SI_70]

6.8.2 Free 2-point scaling

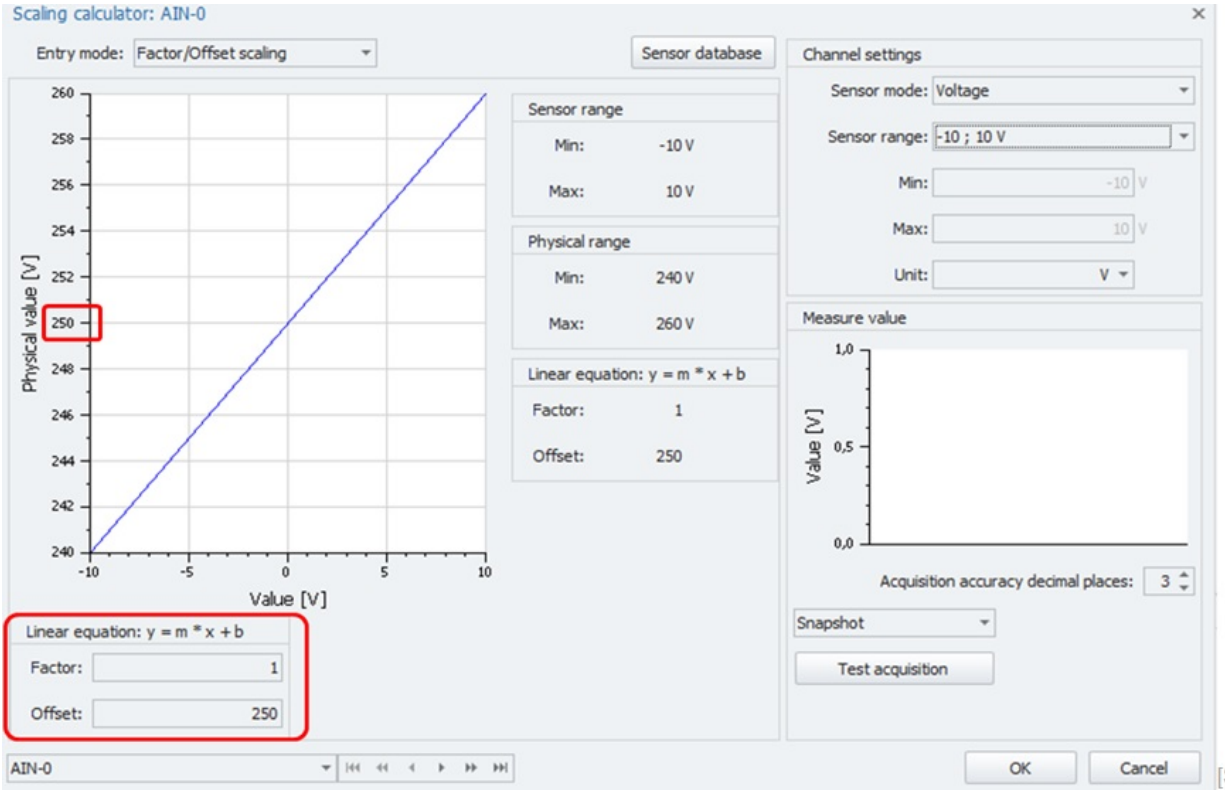
This scaling mode offers the possibility to scale the sensor range and the physical range (engineering units) at the same time.



[SI_71]

6.8.3 Factor/Offset scaling

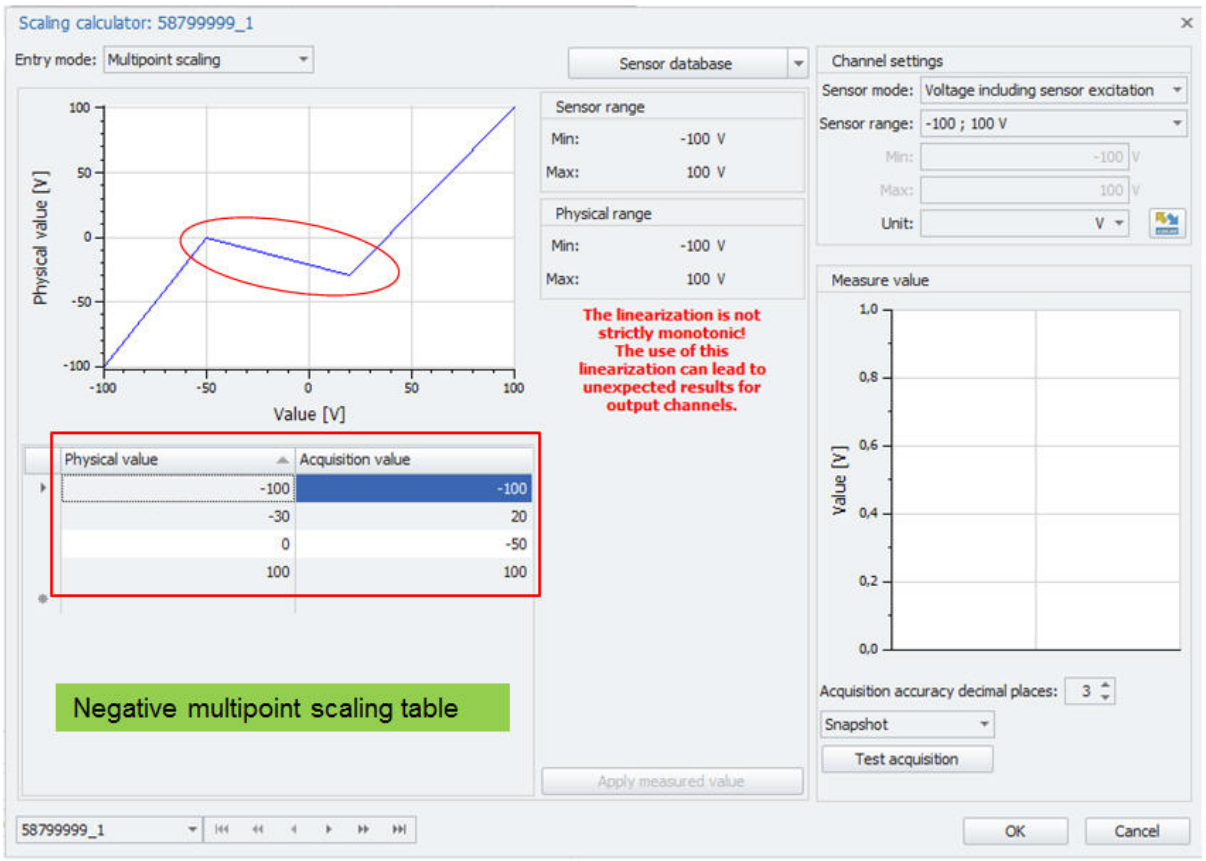
This scaling method uses the equation Physical value (y) = m *x + b (b= offset) with (m = slope factor). The m-factor influences the slope >1 steeper slope / <1 flatter slope. The offset-b shifts the physical value by a constant value.



[SI_72]

6.8.4 Multipoint scaling

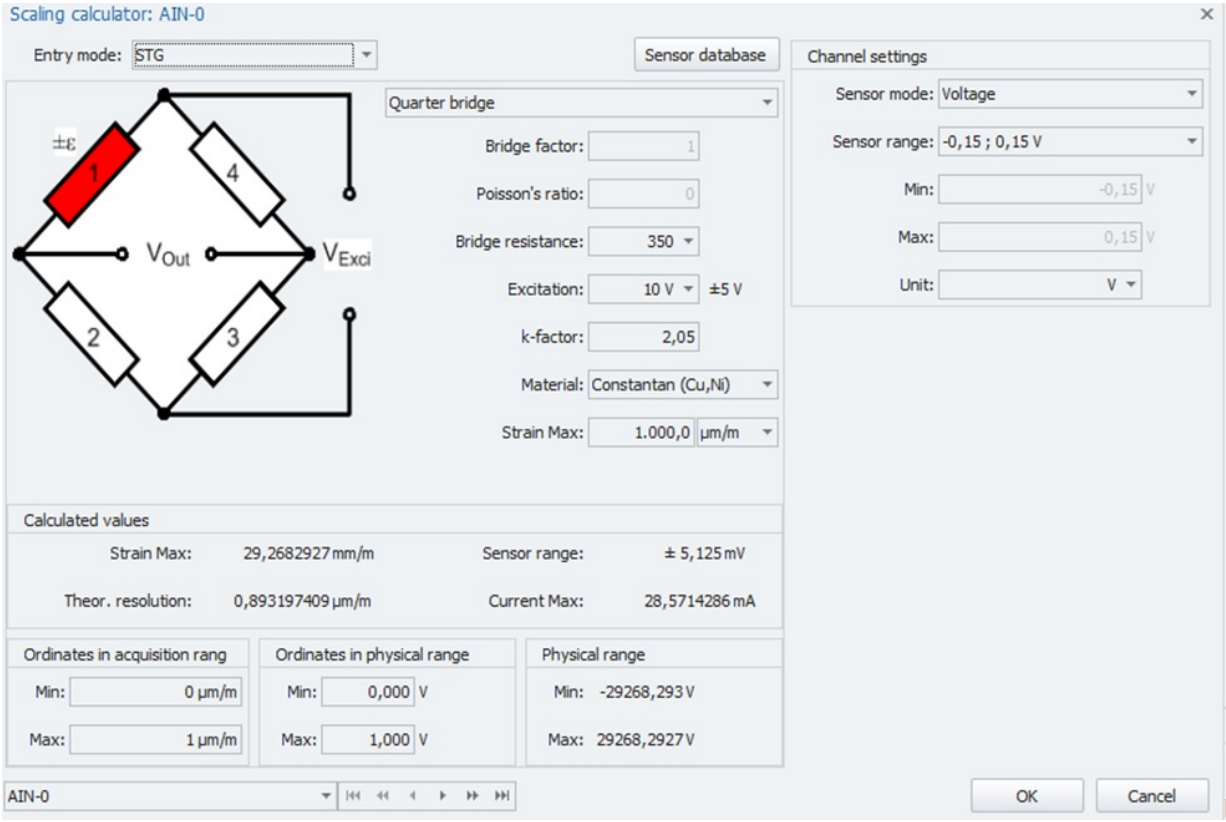
The multipoint scaling is a scaling method that allows to define a nonlinear scaling with as many data points as possible.



⚠ ***Attention!*** *The multipoint scaling parameters are only stored in IPEmotion. They are not transferred to the instrument unless the instrument is supporting this function. See chapter 5.4.*

6.8.5 STG Strain gauge

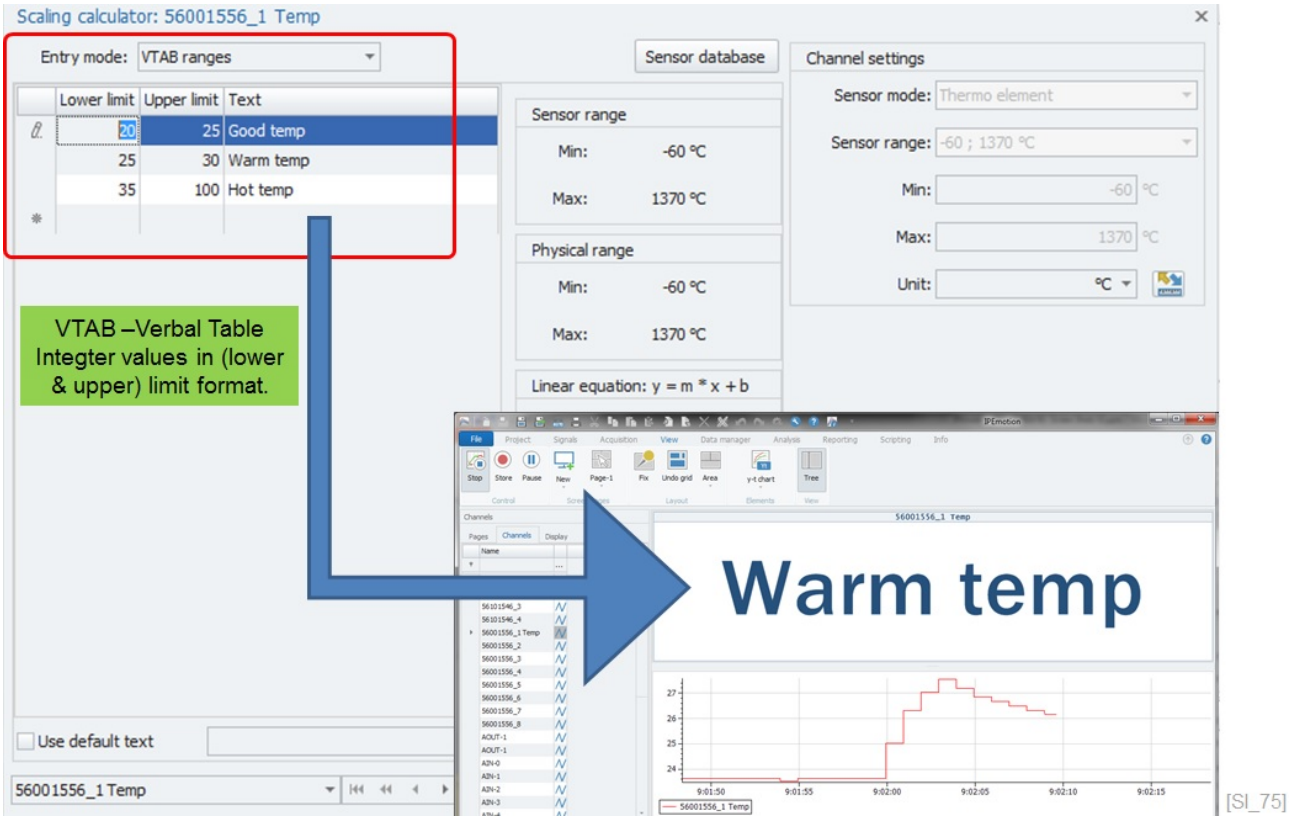
In this interface, strain gauge bridge types like 1/4; 1/2 or full, etc. can be configured.



[SI_74]

6.8.6 VTAB range

This scaling method converts measurements of a specific range into a text message. If the measurement value is in a defined range you can see the corresponding text information on an alphanumerical instrument in the VIEW work area.

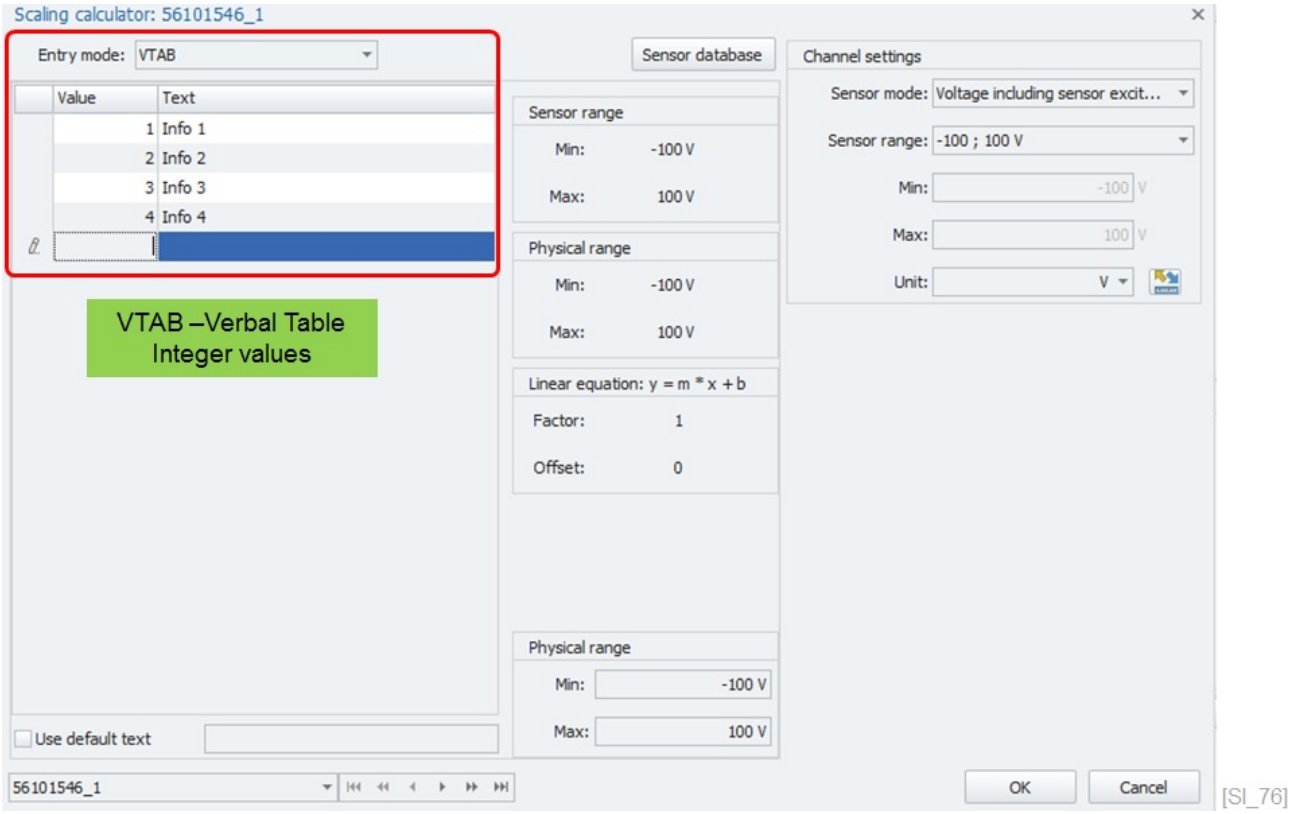


Attention!

The multipoint scaling parameters are only stored in IPEmotion. They are not transferred to the instrument unless the instrument is supporting this function. See chapter 5.4.

6.8.7 VTAB

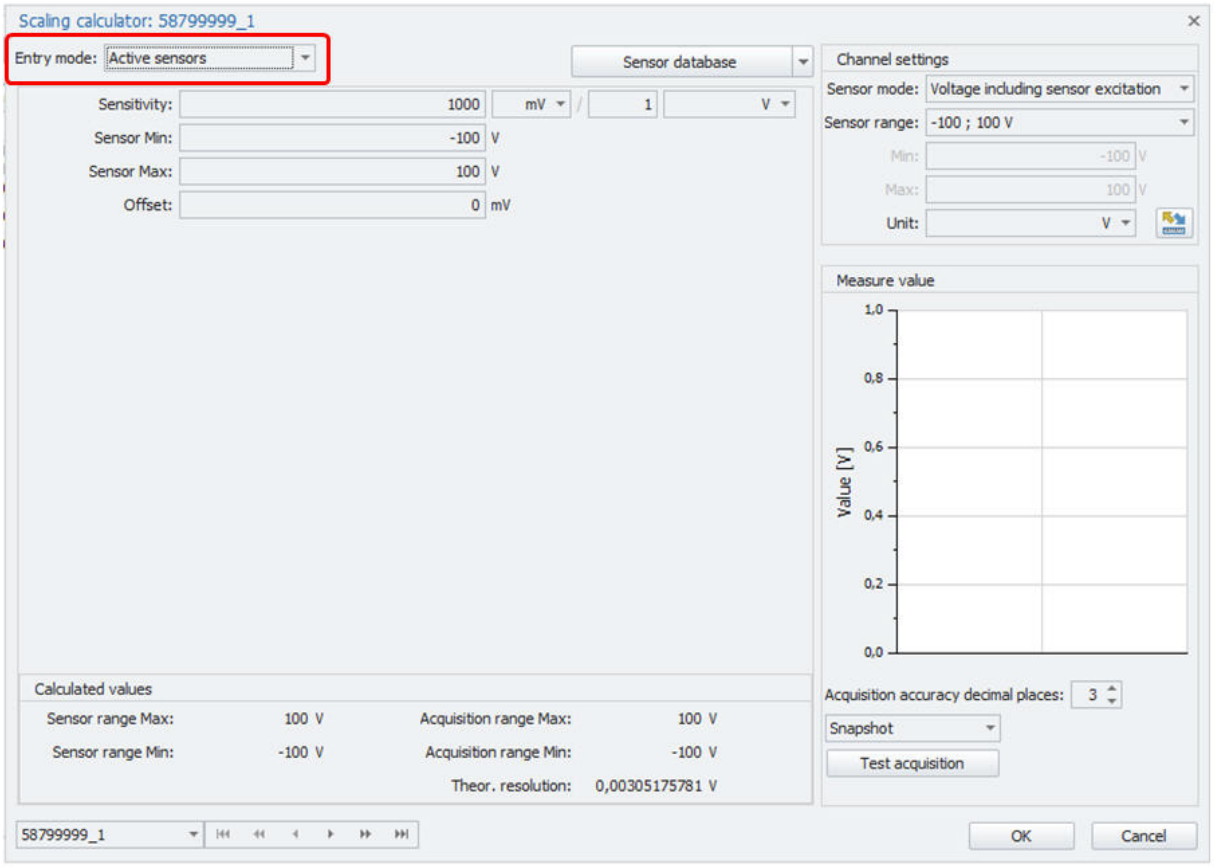
In this mode you can relate a specific integer (1, 2, 3, 4, ..) value to a specific text display. You can display this text on the VIEW work area for example in an alphanumerical instrument.



[SI_76]

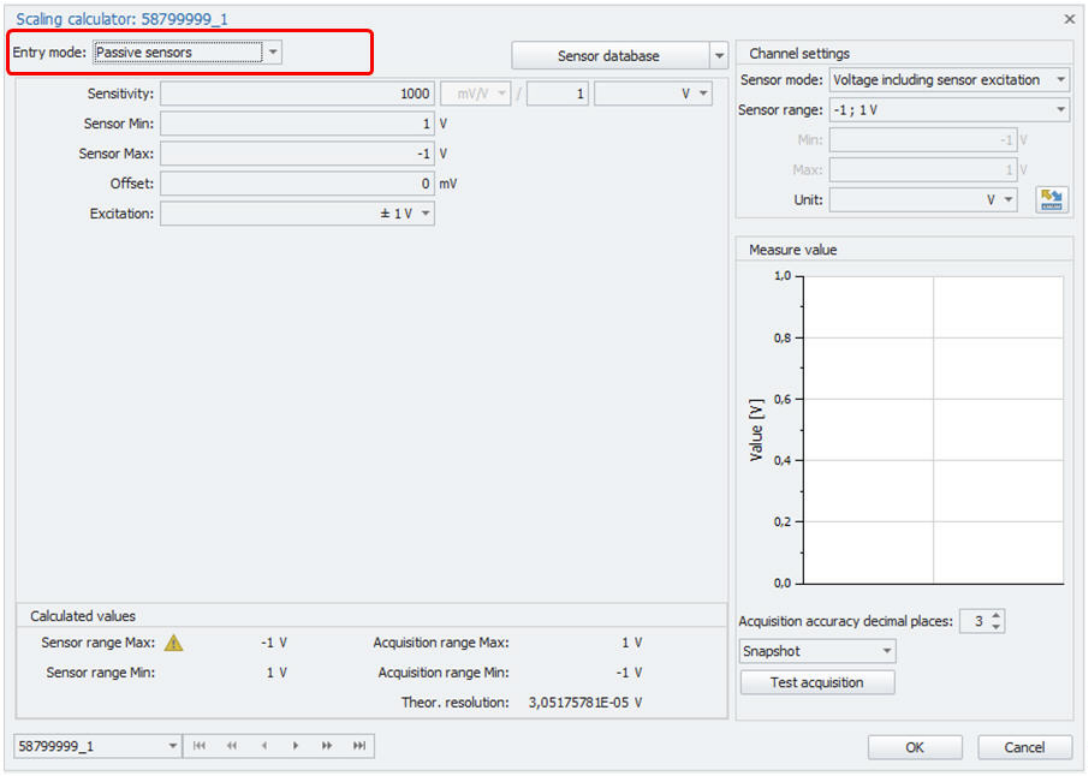
Attention! *The multipoint scaling parameters are only stored in IPEmotion. They are not transferred to the instrument unless the instrument is supporting this function. See chapter 5.4.*

6.8.8 Active Sensors



[SI_77]

6.8.9 Passive Sensors

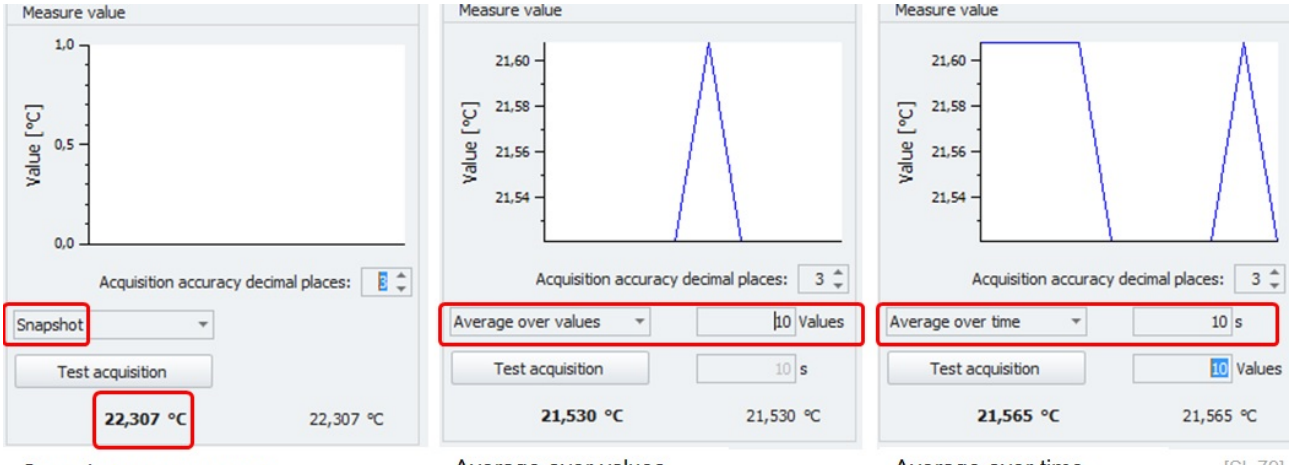


[SI_78]

6.8.10 Snapshot – Test Measurement

You can perform a test measurement within the scaling calculator to check your scaling and to see the actual measurements. Three different test measurements are supported:

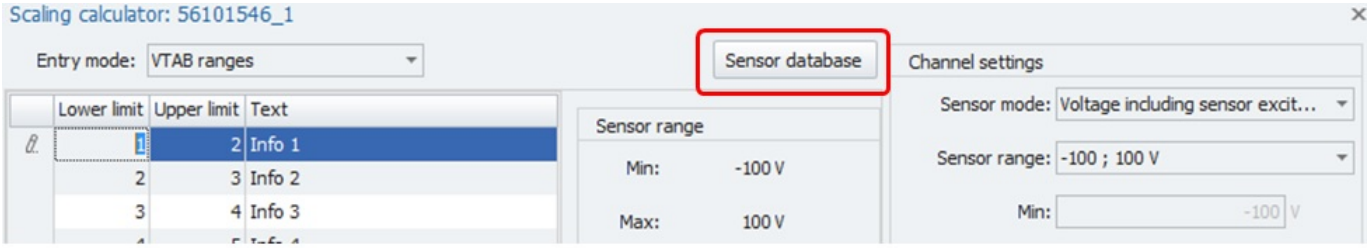
- Snapshot
- Average over values
- Average over time



Snapshot measurement | Average over values | Average over time [SI_79]

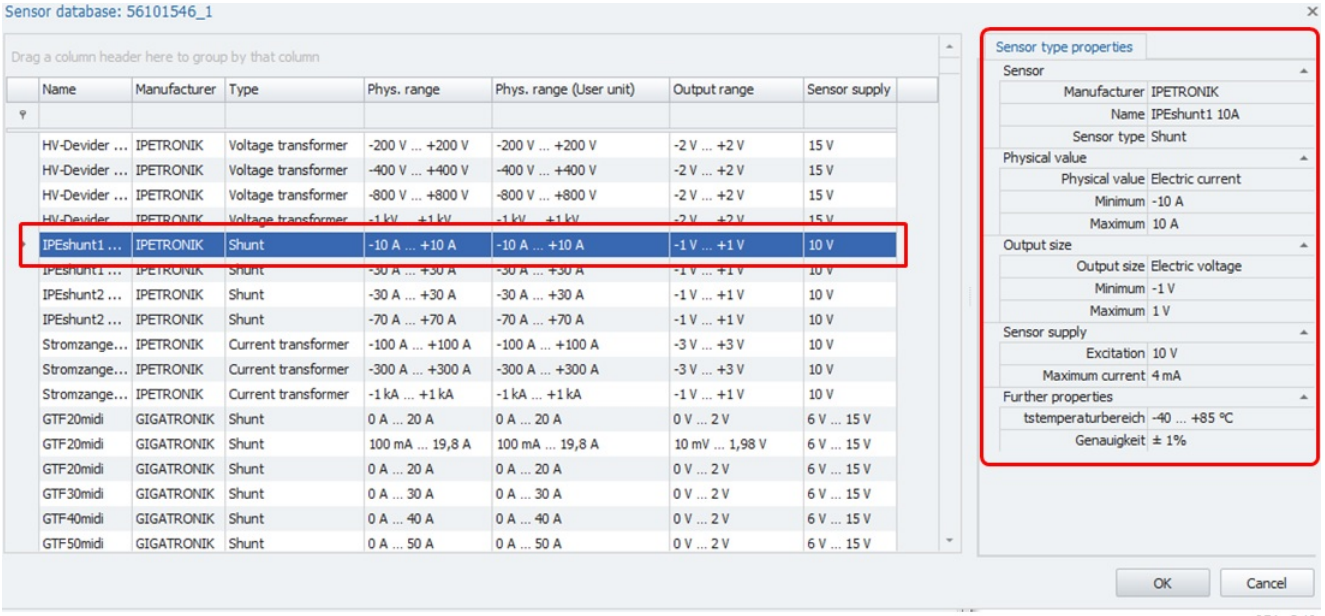
6.9 Sensor database in the scaling calculator

The scaling calculator supports a sensor database. In this database, the scaling parameters of many different sensors are included. If you select a sensor from the database, you have directly defined the measurement range and the physical range and, if needed, a sensor excitation.



[SI_80]

In this example, you see a shunt for high current measurements. This shunt can measure +-10 Amperes and the output of the shunt is +-1 Volt. The sensor requires a 10 Volt sensor excitation.



[SI_81]

6.9.1 Adding new Sensors - Sensor Database Editor

The sensor database (SDB.exe) is installed with each IPEmotion installation in the following directory:

-
- C:\Program Files (x86)\IPETRONIK**IPEmotion 2022 R2**\Tools

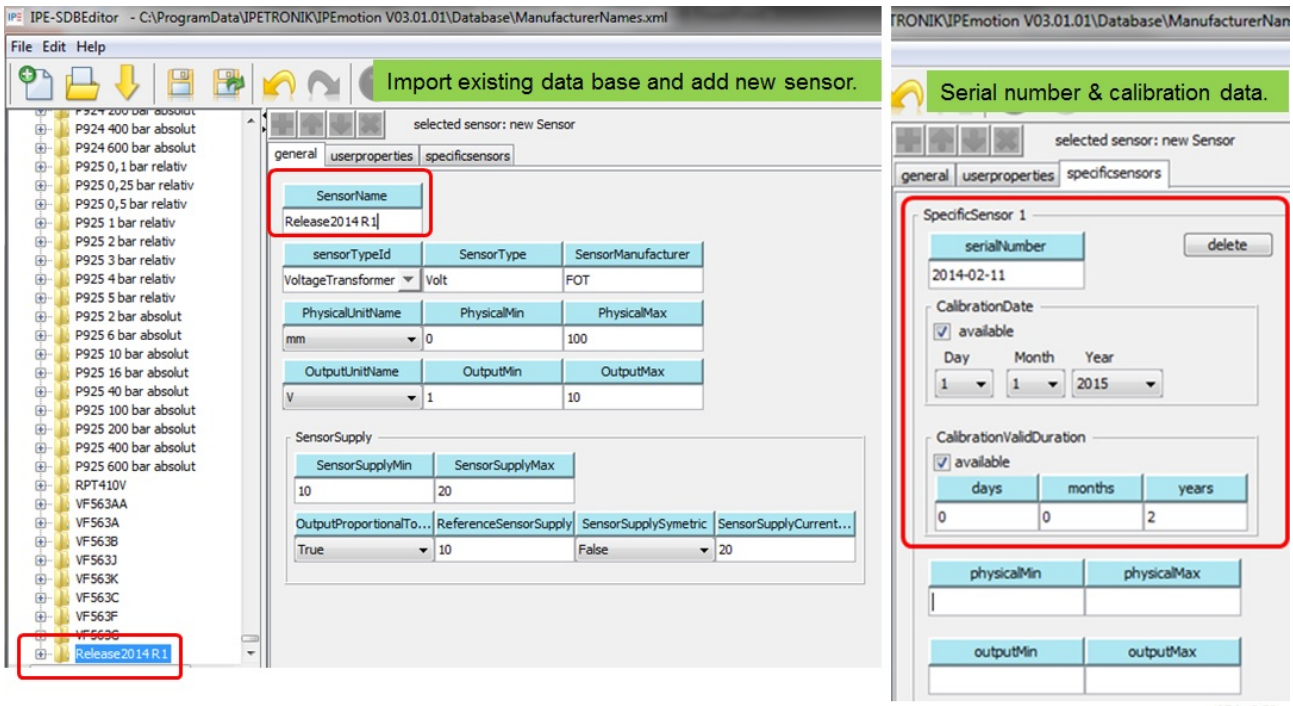
If you like to add your sensor to the existing standard database, it is recommended to import the standard sensor database. The database is installed in the following directory.

-
- C:\ProgramData\IPETRONIK**IPEmotion 2022 R2**\Database

You can also create your own sensor database XML file from scratch. If you like to use your own database file you have to store it in the right directory and give the file the correct name: **IPESensorDatabase.xml**

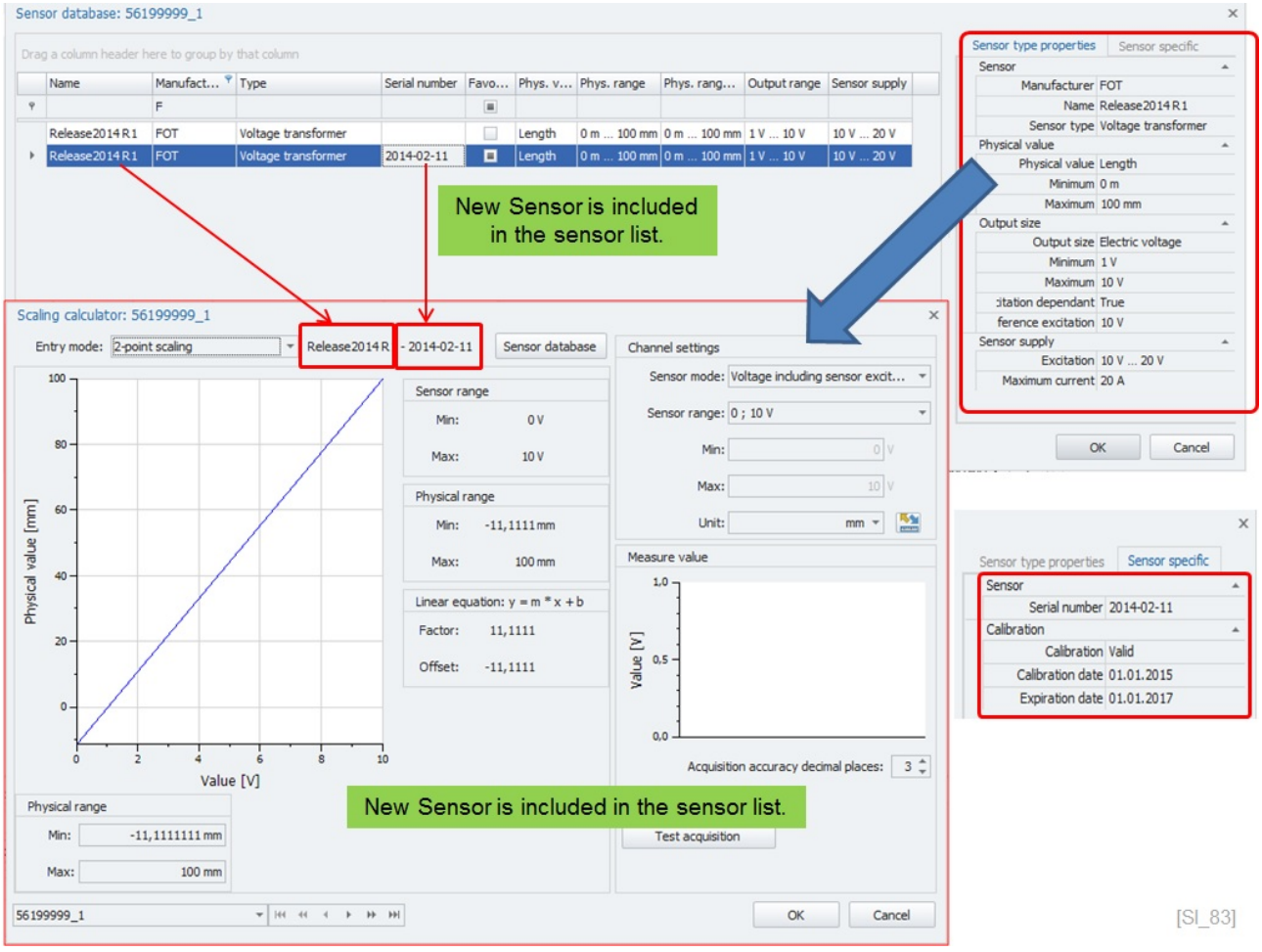
IPEmotion can only work with one database XML file.

You can add new sensor by means of the SensorDB editor. This tool is installed along with IPEmotion and entries can be made through the GUI.



[SI_82]

If you save the new sensor and restart IPEmotion, the new sensor will be included in the database and can be selected for channel scaling. Serial numbers and calibration dates can be defined, as well.



[SI_83]

However, you can import your own database from Excel using the import function of the SensorDB Editor. The import function is explained in the help manual of the SensorDB Editor.

6.9.2 The database format

The standard Excel template for importing sensors has the following structure:

- sensorName
- sensorTypeId (see next page for details)
- sensorType
- sensorManufacturer
- physicalUnitName
- physicalMin
- physicalMax
- outputUnitName
- outputMin
- outputMax
- sensorSupplyMin
- sensorSupplyMax
- outputProportionalToSupply
- referenceSensorSupply
- sensorSupplySymetric
- sensorSupplyCurrentMax
- propertyName1
- propertyValue1
- propertyName2
- propertyValue2
- propertyName3
- propertyValue3
- serialNumber
- calibrationDate
- calibrationValidMonths
- calibrationValidDays
- physicalMin
- physicalMax
- outputMin
- outputMax

The Sensor type ID

- 0 = UNKNOWN // User-defined sensor
- 1 = DisplacementTransducer // Displacement transducer
- 2 = LoadCell // Load cell
- 3 = Shunt // Shunt
- 4 = CurrentTransformer // Current transformer
- 5 = VoltageTransformer // Voltage transformer
- 6 = ForceTransducer // Force transducer
- 7 = PressureTransmitter // Pressure transmitter
- 8 = AbsolutePressureTransmitter // Absolute pressure transmitter
- 9 = GaugePressureTransmitter // Gauge pressure transmitter
- 10 = DifferentialPressureTransmitter // Differential pressure transmitter
- 11 = FlowRateTurbine // Flow rate turbine
- 12 = PistonFlowmeter // Piston flow meter
- 13 = ScrewFlowmeter // Screw flow meter
- 14 = VortexSheddingDevice // Vortex shedding device
- 15 = Accelerometer // Accelerometer
- 16 = TriAxialAccelerometer // Triaxial accelerometer
- 17 = TorqueMeter // Torque meter
- 18 = Counter // Counter
- 19 = StrainGauge // STG
- 20 = LVDT // LVDT
- 21 = StrainGaugeBridge // STG bridges (Strain)
- 22 = TemperatureSenso // Temperature senso

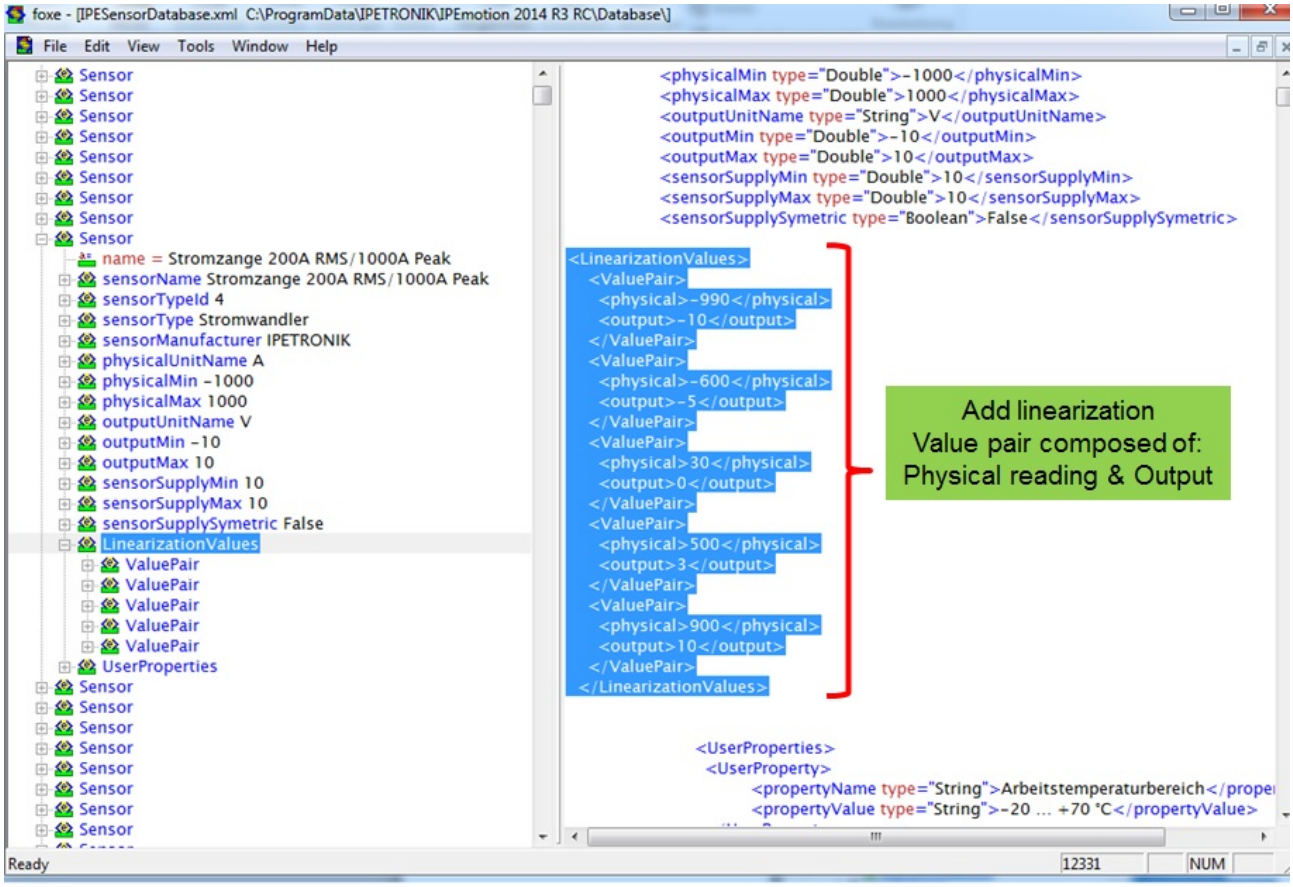
If none of the predefined types meets your requirements, you can add user-defined types to the sensor database. The ”sensorTypeId” value must be set to 0. A short description text should classify the corresponding sensor.
If you want to use sensors of the same type within the sensor database, use the ”SpecificSensors” entry. Each one of the sensors must get a unique serial number (”serialNumber”). In addition, each one of these sensors can get a calibration date (”calibrationDate”), as well as a period of validity of the calibration (”CalibrationValidDuration”) including the data ”calibrationnValidYears”, ”calibrationValidMonths”, and ”calibrationValidDays”. Furthermore, the values for ”physicalMin”, ”physicalMax”, ”outputMin”, and ”outputMax”, which can be found in the data sheet, can be overwritten by values, which are read at the calibration.
You can add non-relevant information for the functionality of the sensor data base like the working temperature range under the ”UserProperties” entry. These are Key/Value pairs, which are used for displaying the information. Please note that these data are not used in any calculation.
All the sensor data is stored in an XML file with the following structure.

The sensor names (”sensorName”) must be unique!

```
<Sensor name=SSensor2">
<sensorName type=SString">Sensor2</sensorName>
<sensorTypeId type=Ïnt32">7</sensorTypeId>
<sensorManufacturer type=SString">IPETRONIK</sensorManufacturer>
<physicalUnitName type=SString">bar</physicalUnitName>
<physicalMin type="Double">1</physicalMin>
<physicalMax type="Double">50</physicalMax>
<outputUnitName type=SString">V</outputUnitName>
<outputMin type="Double">-4</outputMin>
<outputMax type="Double">4</outputMax>
<sensorSupplyMin type="Double">-5</sensorSupplyMin>
<sensorSupplyMax type="Double">5</sensorSupplyMax>
<sensorSupplyCurrentMax type="Double">0.01</sensorSupplyCurrentMax>
<PreferedSensorModes>
<sensorMode />
</PreferedSensorModes>
<UserProperties>
<UserProperty>
<propertyName type=SString">Genauigkeit</propertyName>
<propertyValue type=SString">+- 4,7 %</propertyValue>
</UserProperty>
</UserProperties>
<SpecificSensors>
<SpecificSensor>
<serialNumber type=SString/>
</SpecificSensor>
<SpecificSensor>
<serialNumber type=SString">SN01277</serialNumber>
<calibrationDate type="Date">2012-04-04</calibrationDate>
<CalibrationValidDuration>
<calibrationValidYears type=Ïnt32">1</calibrationValidYears>
<calibrationValidMonths type=Ïnt32">6</calibrationValidMonths>
<calibrationValidDays type=Ïnt32">0</calibrationValidDays>
</CalibrationValidDuration>
<outputMin type="Double">-3.895</outputMin>
<outputMax type="Double">4</outputMax>
</SpecificSensor>
</SpecificSensors>
</Sensor>
```

6.9.3 Multipoint linearization

The sensor data base is supporting sensor linearization functions. You can add for sensors multipoint linearization into sensor data base XML file. In the XML file you can add value pairs of ”Physical reading / Sensor Output”.

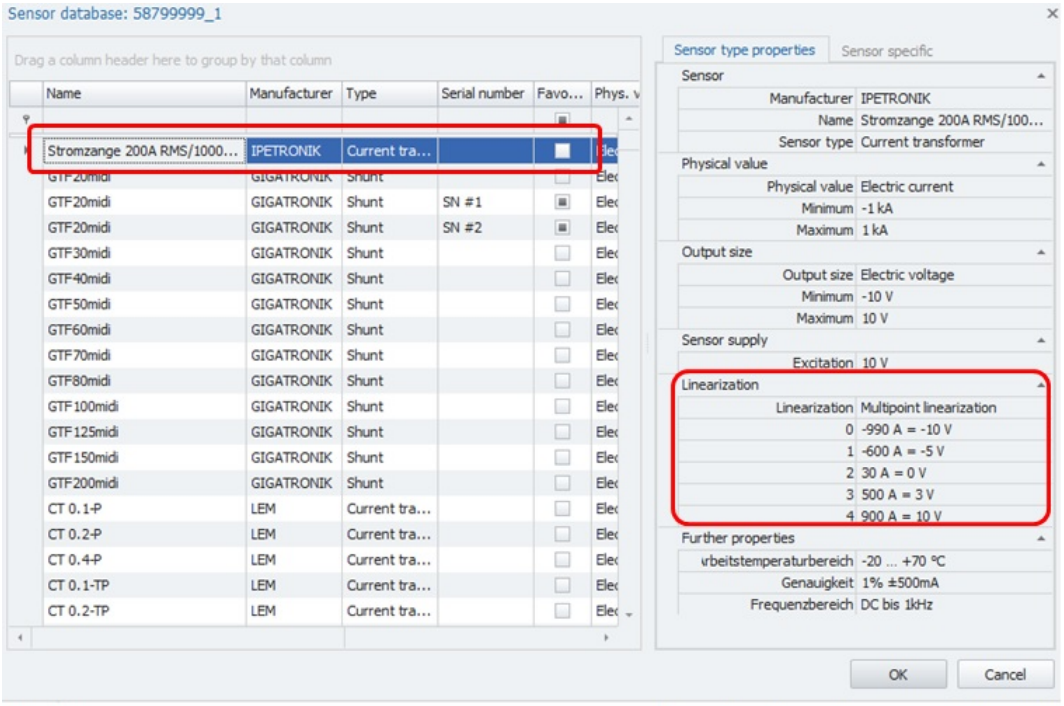


[SI_84]

Information

The sensor specific linearization information can only be added through the XML file directly. The Sensor Database Editor and the corresponding CSV/Excel import function is ***currently not supporting*** *this function.*

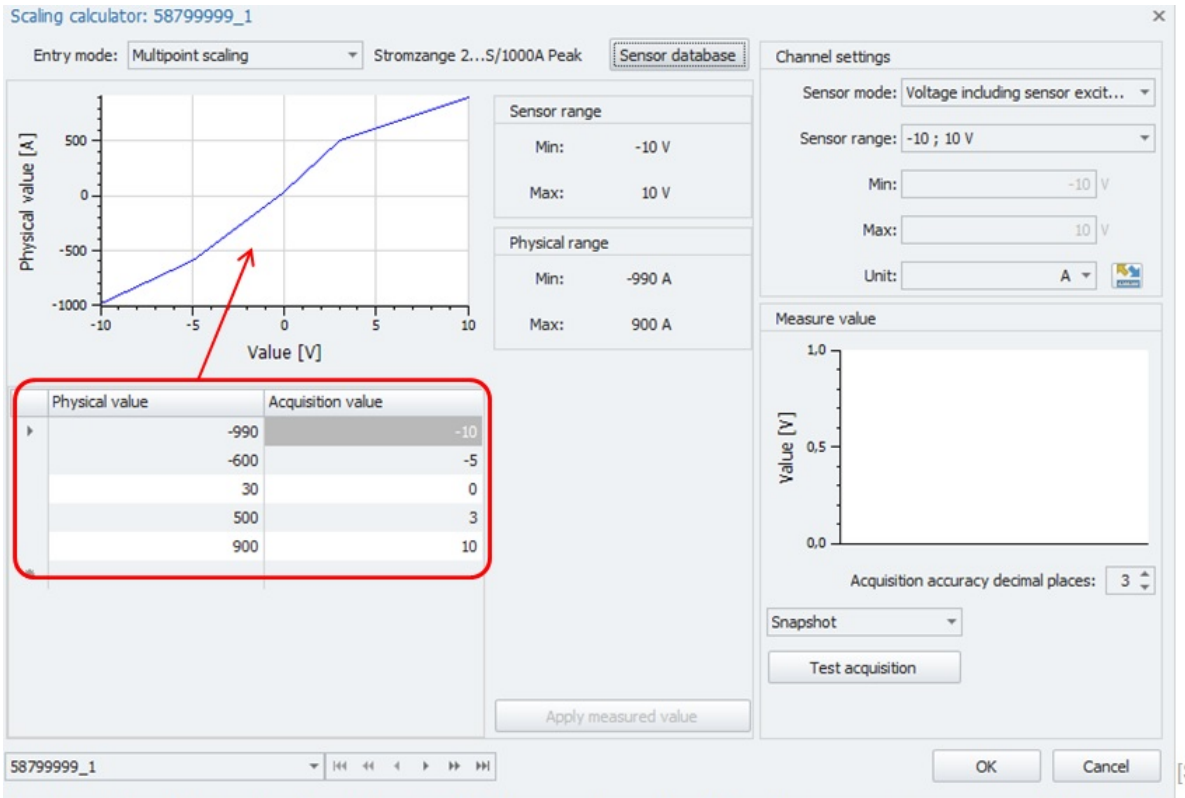
When you select a sensor with linearization values they are directly indicated in the sensor parameter overview. In this example the scaling is integrated to the ”Sensor type properties”.



[SI_85]

Linearization values are indicated.

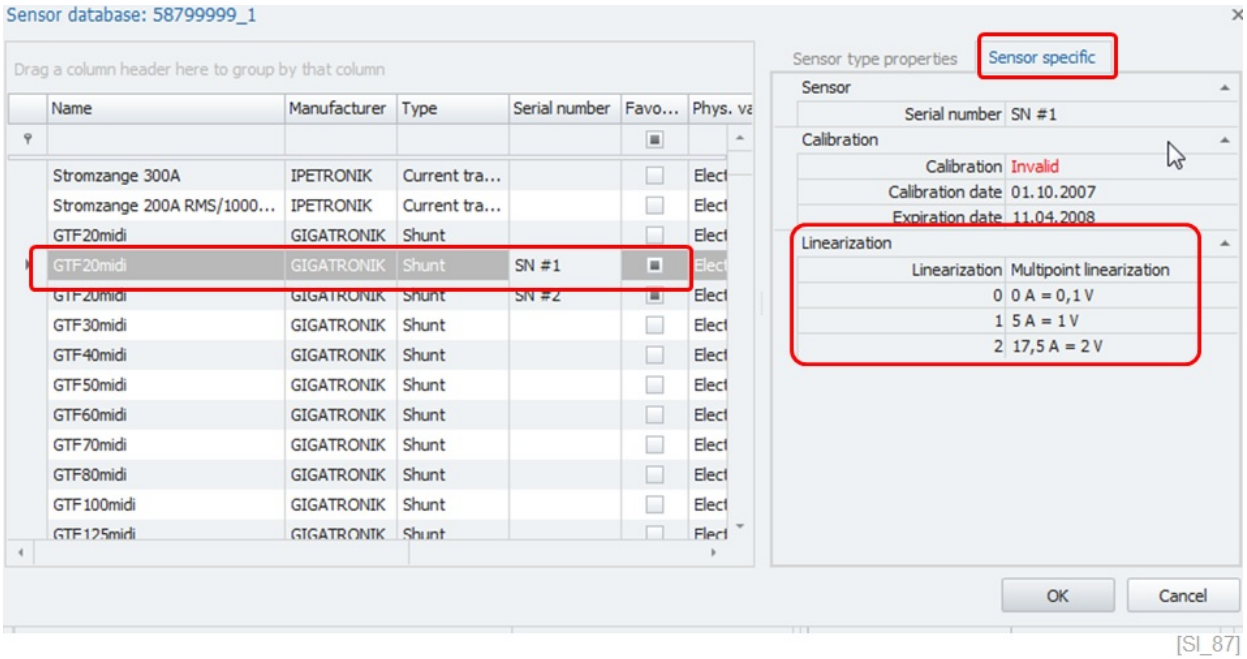
The linearization values are imported from the sensor database into the multipoint scaling mode with a graphical presentation of the calibration curve.



[SI_86]

Multi point scaling is directly integrated to the scaling calculator with a graphical presentation of the calibration curve.

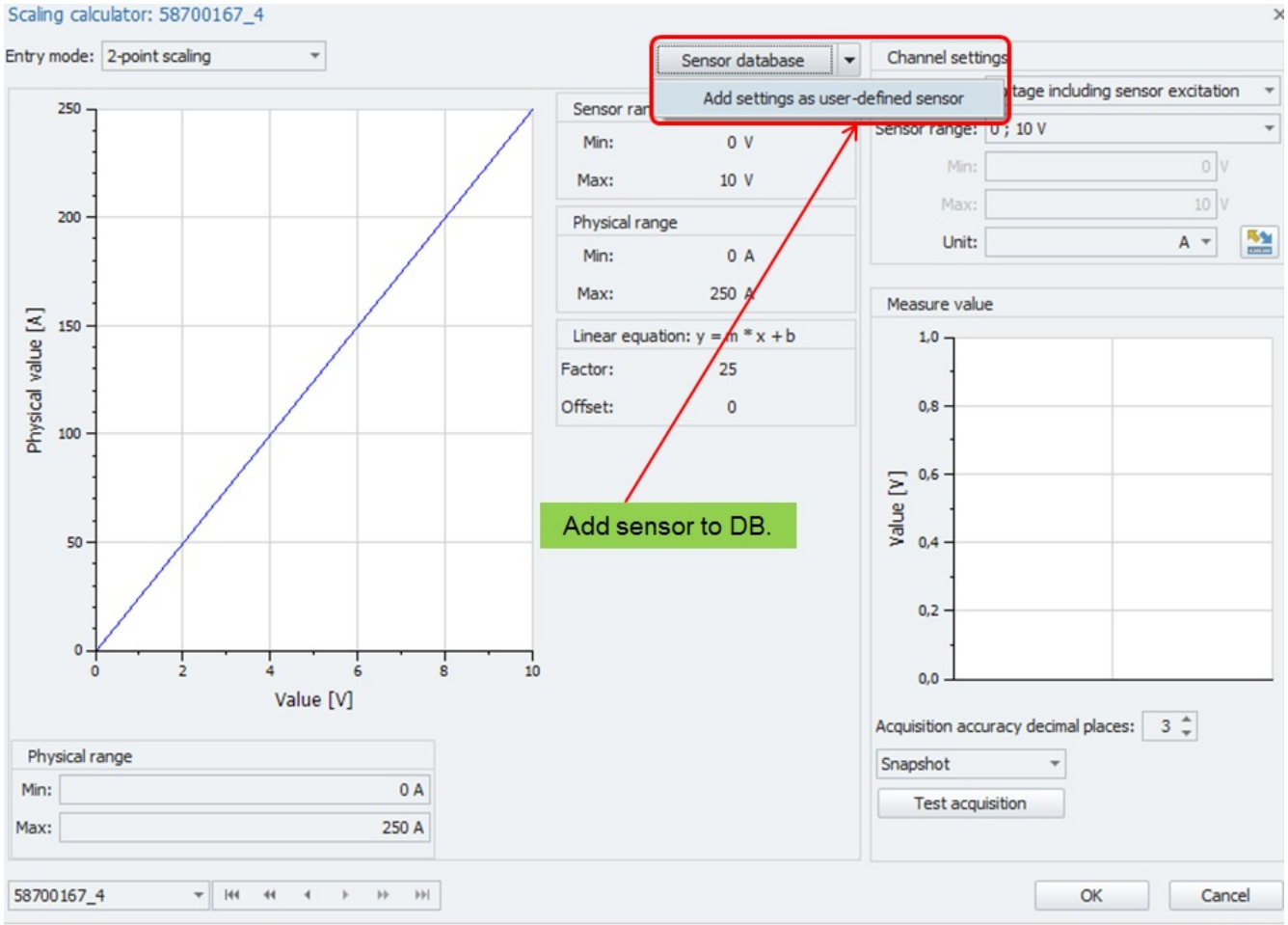
You can integrate multi point scaling also to the ”Sensor specific” properties.



[SI_87]

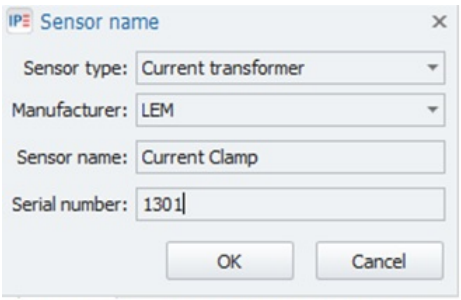
6.9.4 Adding new Sensors – through the Scaling Calculator

The sensor data base is a powerful tool to simplify the channels scaling and reduce scaling error. You can now add your own sensor to the data base. All the settings defined in the sailing interface are saved to the data base. All scaling entry modes are supported to add individual sensors 6.8.



Sensor definition from the scaling calculator is the basis for the sensor. [SI_88]

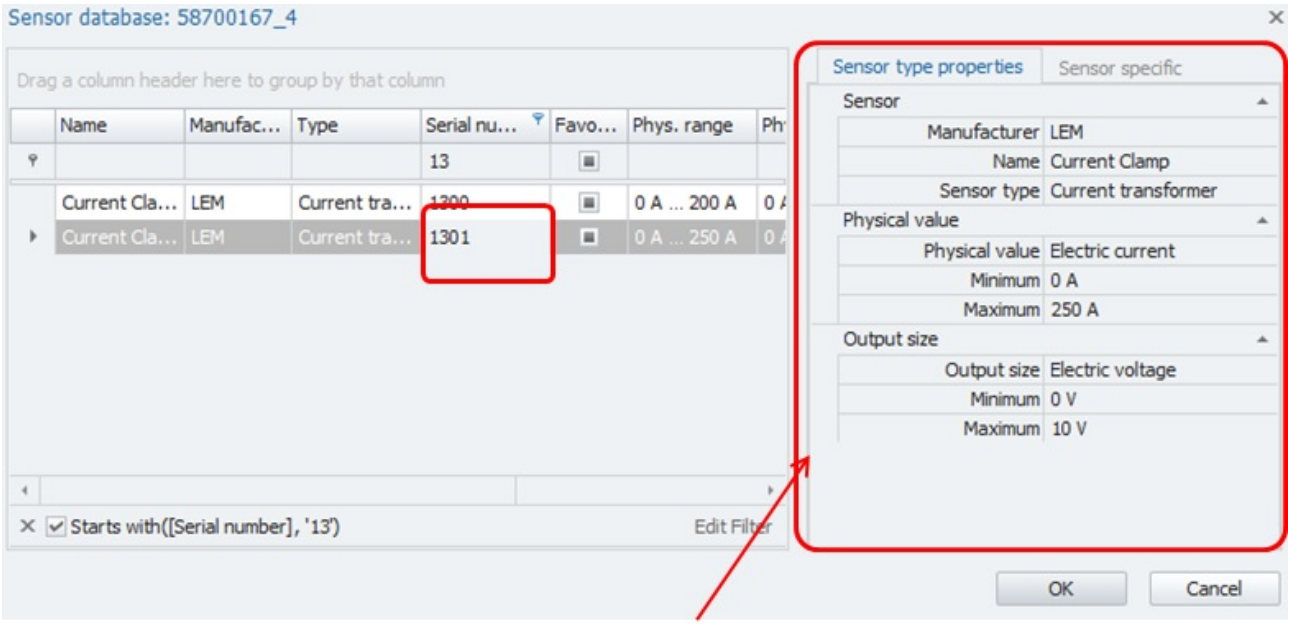
When the sensor parameters are defined you add the sensor header information by accessing the add button.



[SI_89]

Define sensor header information.

After you have created the sensor in the data base you can search for your sensor. The example below shows the parameters as defined in the scaling calculator.



Sensor properties defined in the scaling calculator are displayed. [SI_90]

When a sensor was added to a user define sensor data base file it is saved in:

- C:\Users\Public\Documents\IPETRONIK\IPEmotion\Database\ IPESensorDatabase.xmu
- Extension u = user defined sensor data base.

If you like to modify a manually created sensor you need select the sensor from the sensor data base and you can modify settings in the scaling interface. With the function save sensor to data base the modifications are overwritten.

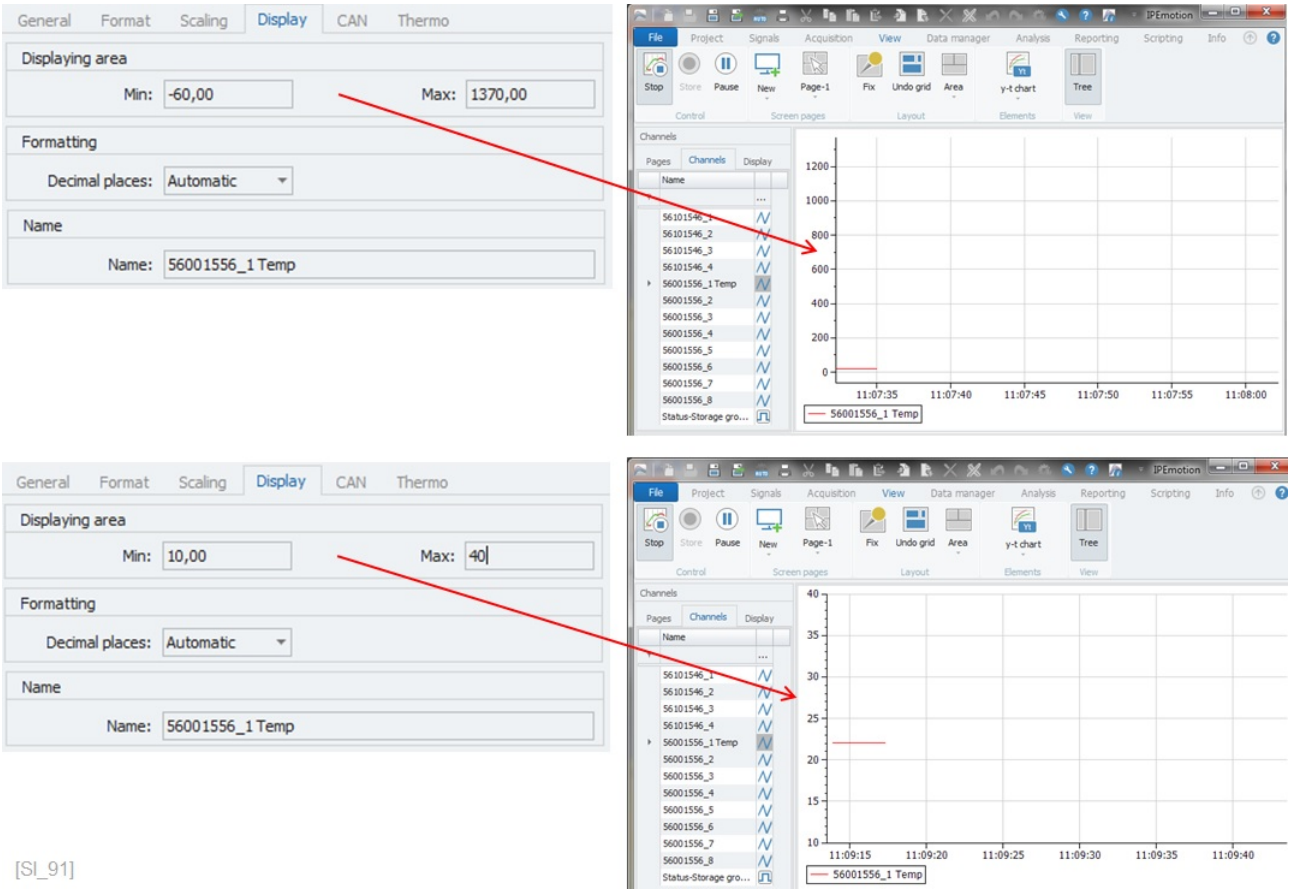
Information

Note: There is no possibility to delete a sensor from the sensor data base. If you need to remove a sensor permanently you need to delete it from the XML files.

6.10 Display tab sheet

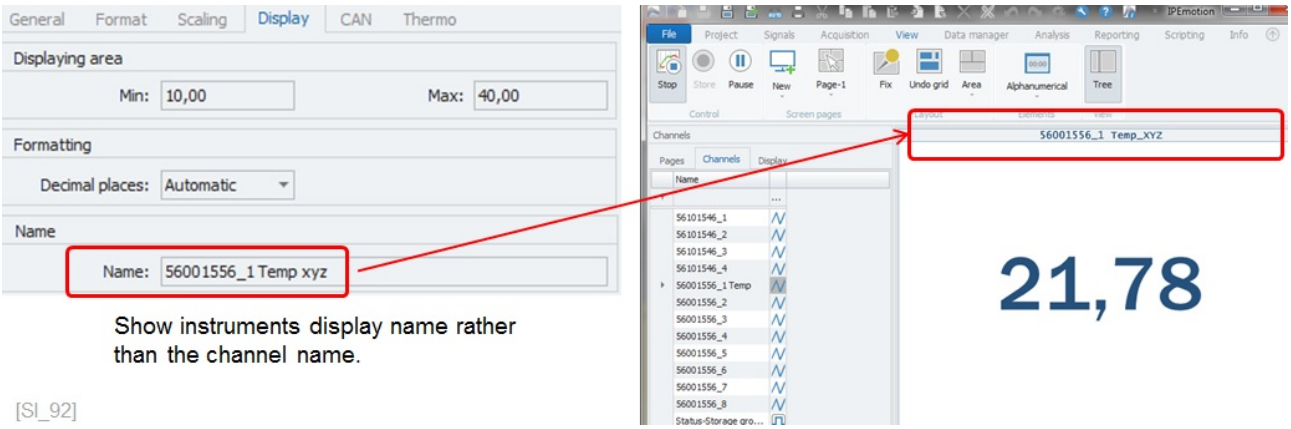
This tab sheet covers display settings for the online VIEW work area. The Display tab sheet is also relevant for formula channels and scaling channels **??**. The main configuration elements are:

- Display Area — Covers the initial Y-axis scaling of the Yt-chart.



[SI_91]

- Formatting — Covers the decimal places. The default setting is Automatic which will show as many decimal places as provided by the PlugIn
- (Display) Name — Covers the display name which can differ from the channel name. The display name is only relevant for the VIEW work area. The display name will not be used for formulas and other functions like limit or range monitoring. If you like to see the display name on the instruments, you will have to activate this function in OPTIONS >VIEW **??**



[SI_92]

6.10.1 Define standard decimal templates on module level

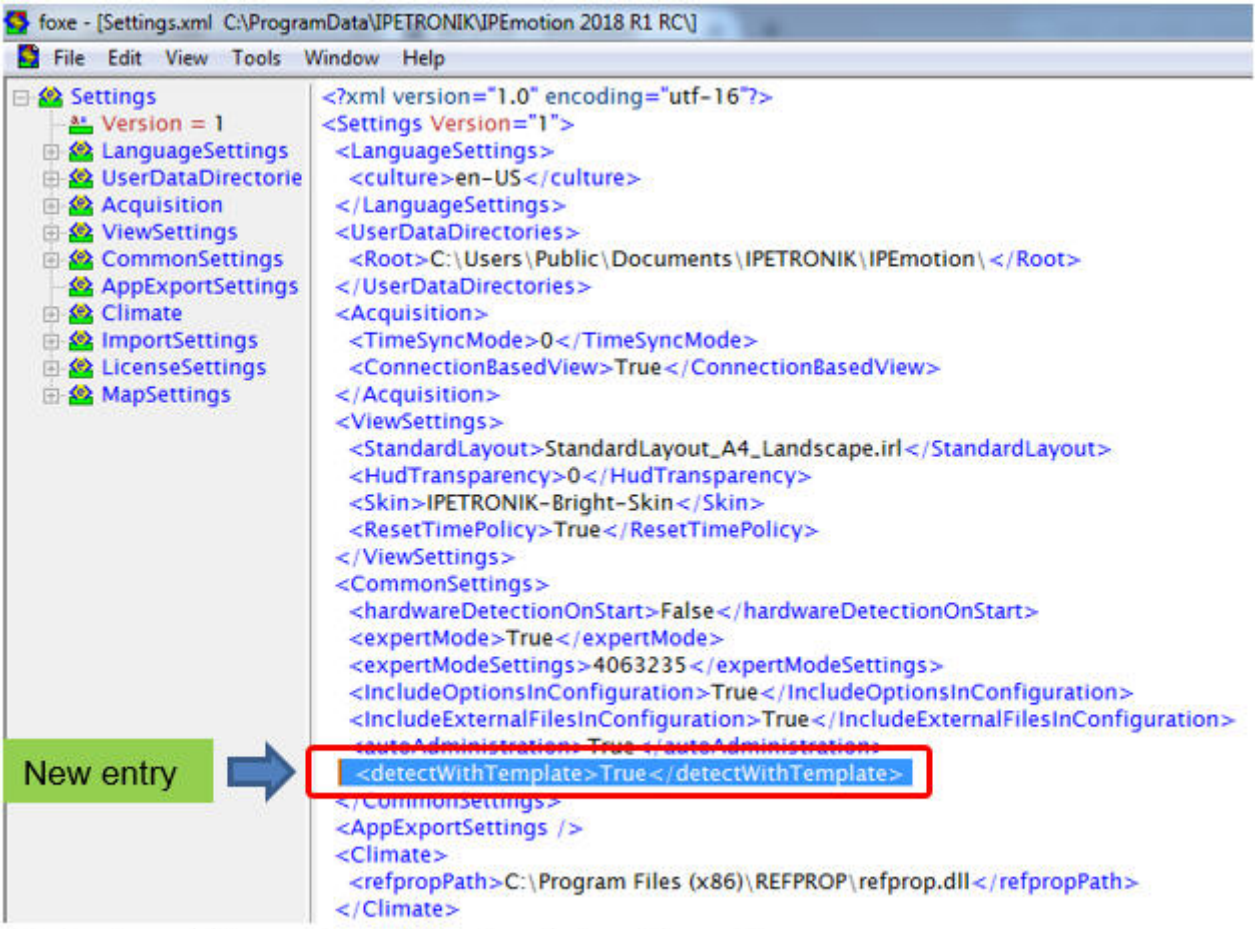
When detecting modules, the default setting of the decimal paces is defined as Automatic. However, if you like to define a default setting for the number of decimal places you like to use you can add to the Settings.XML a

new command line in order to use the template as default. The settings XML file is stored on the following directory:

- C:\ProgramData\IPETRONIK**IPEmotion 2022 R2**\Settings.xml

In the settings XML you have to add in the section ”Common Settings” the following command line:

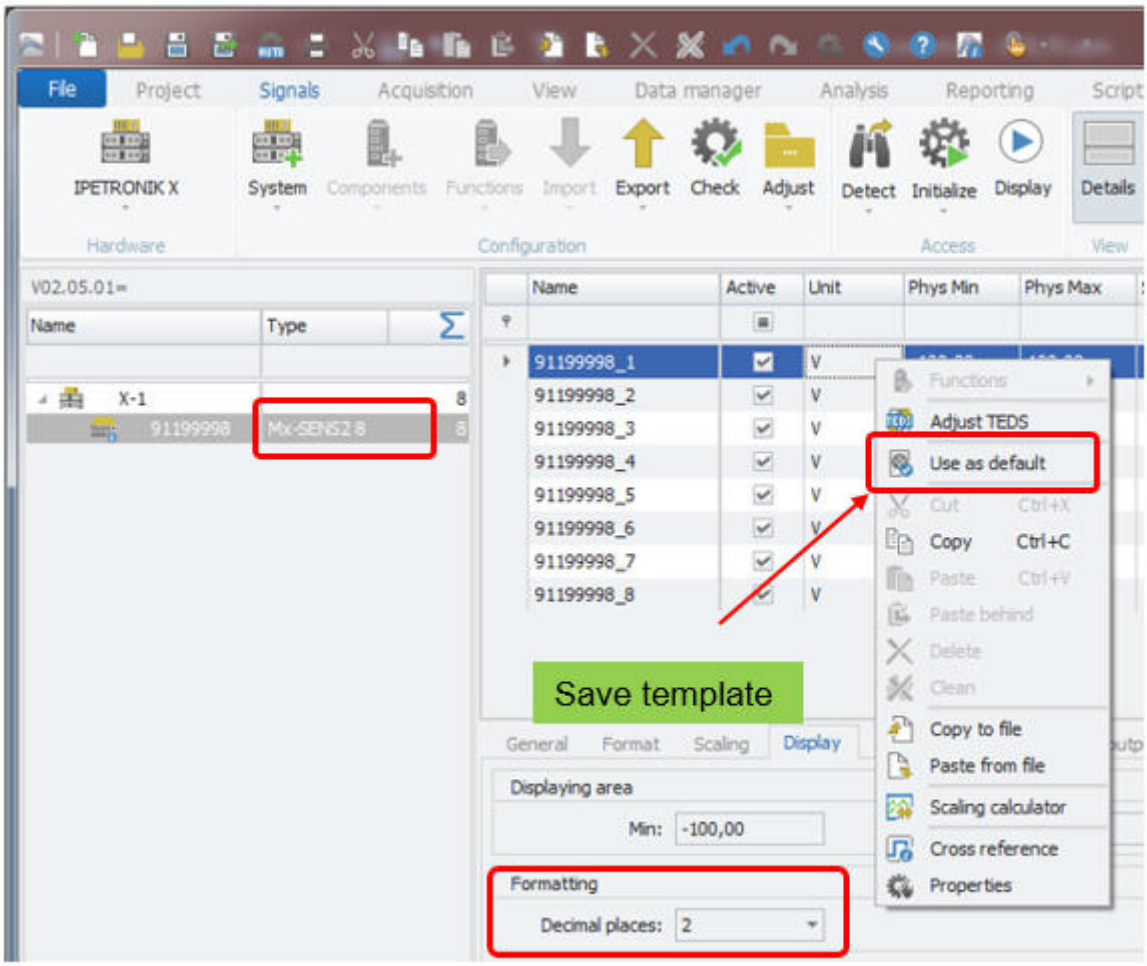
<detectWithTemplate>True <\detectWithTemplate>



Detect modules and consider the defined template. [SI_95]

You can disable the function also by setting the command line to "False". <detectWithTemplate>False <\detectWithTemplate>

With this command line you can save a lot of time because all modules will be detected with the number of decimal places as defined in the template. The template is applied to all channels of the module.

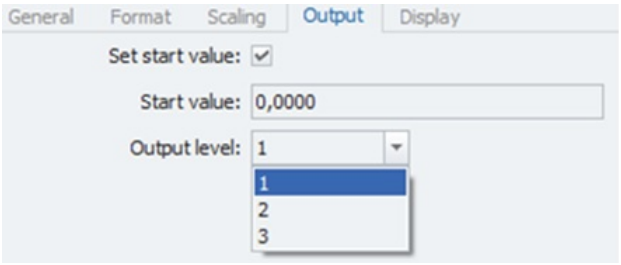


Example: Template with 2 decimal places for the Mx-SENS2 8 is created.

[SI_96]

6.10.2 Output tab sheet for output channel

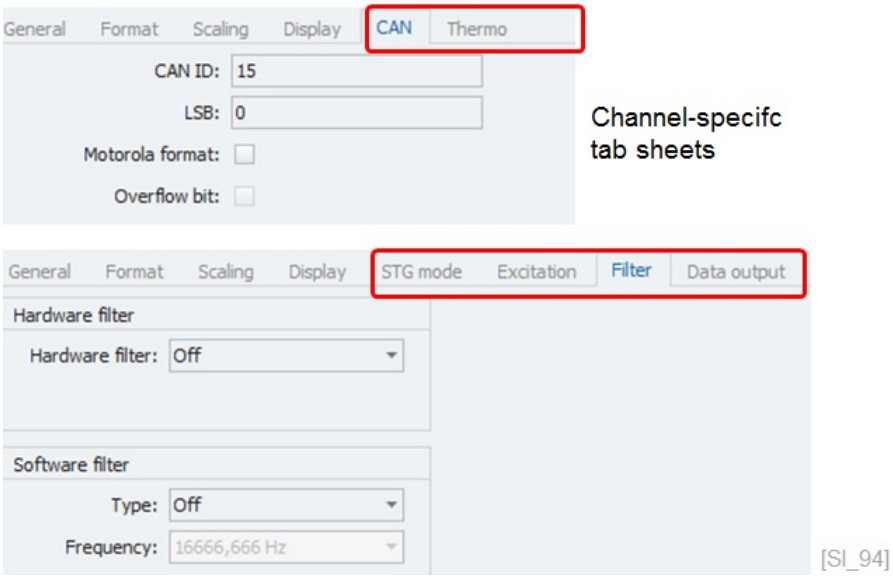
The output tab sheet is only visible for analog and digital output channels. Its main function is to define a start value. This value will be set to the output when you start the measurement. You can also define an output level. The output level is related to the user administration which is discussed in detail in OPTIONS >User administration **??**.



[SI_93]

6.10.3 Channel-specific tab sheets

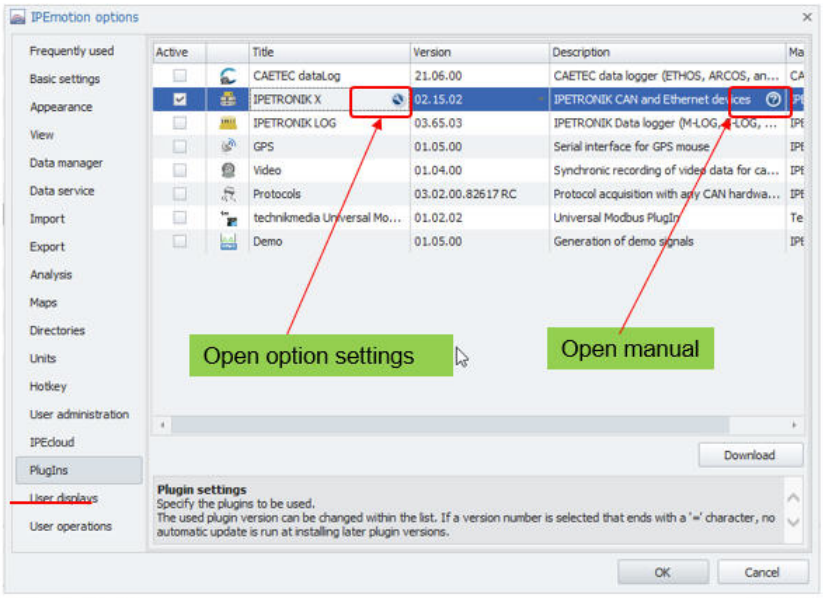
Each PlugIn can have channel-specific tab sheets to cover additional configuration functions. There are many individual functions which are discussed in the PlugIn manual in more detail. Some examples are shown below.



[SI_94]

7 X-PlugIn OPTIONS

When you access the OPTIONS dialog of IPEmotion you have access to the advanced PlugIn settings and the manual. In this example the options settings of the IPETRONIK X PlugIn are discussed. However other PlugIns will have other settings which are explained in the dedicated PlugIn manuals

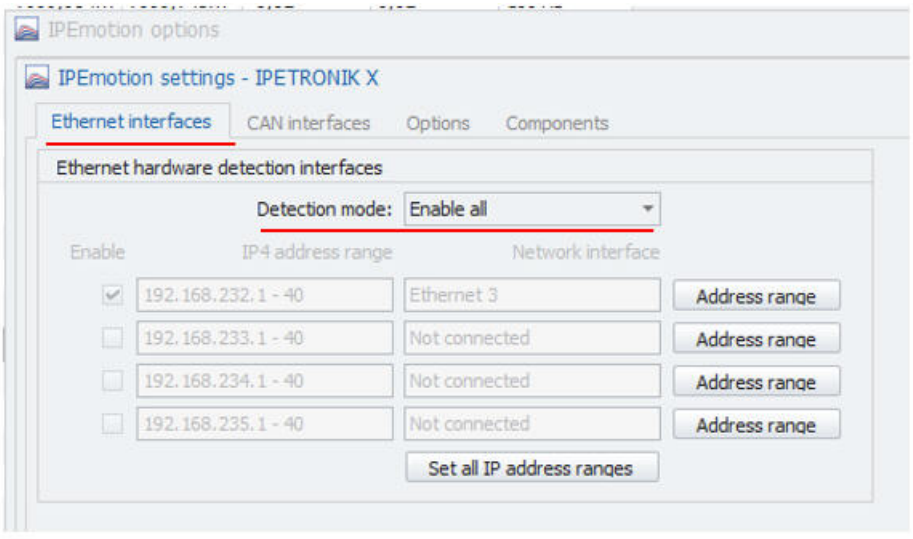


PlugIn Options [SI_150]

7.1 Ethernet interface

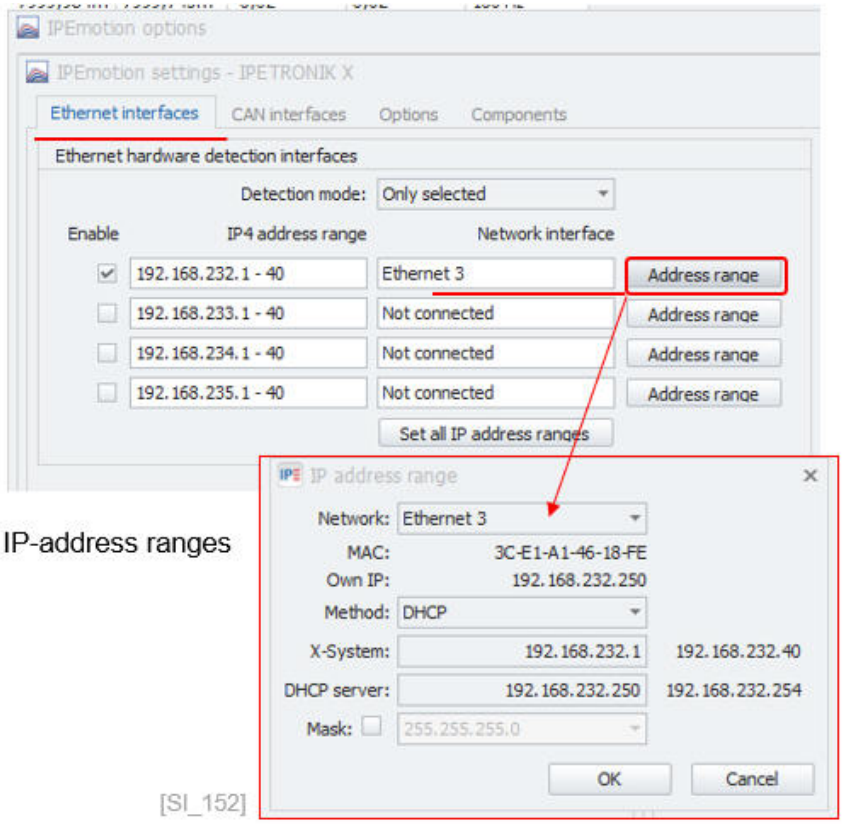
On the Ethernet interface tabs het you have configuration functions for the IP-address and the detection mode. The ethernet and IP-adress settings are relevant for the X-Modules only.

- Enable all — Will perform the scan for module across all Ethernet interfaces of the PC. This can take more time.
- Disable all — Will not allow any detection of the modules on an Ethernet port.
- Selected — Here you will perform the scan only on a dedicated Ethernet port of the computer.



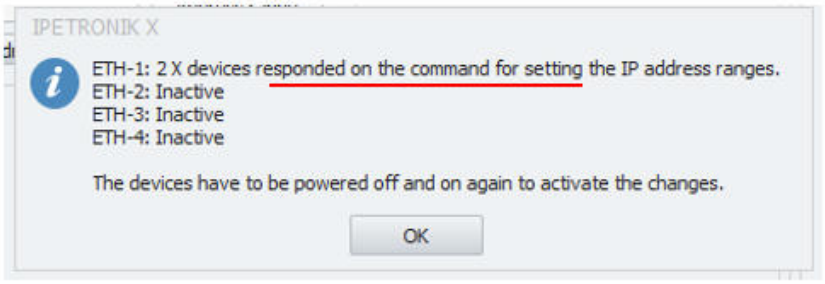
Ethernet interface – detection mode [SI_151]

If you define a selected Ethernet ports you have also access to an advanced configuration dialog to change the IP-address ranges of the modules. The setting in the advanced dialog should be handled with care a as it is has impact on the Ethernet interface of the computer and the address of the modules. If the address ranges are changed to match corporate IT network requirements is might be possible that the modules cannot be detected any more on another computer with different network settings.



IP-address ranges [SI_152]

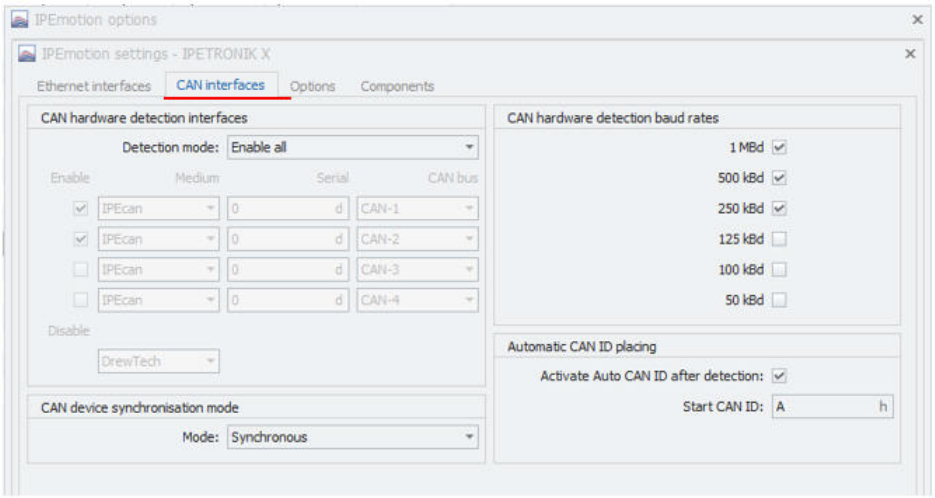
When you make any changes you need to conform those and a message window is indicating the update of the modules.



Update info [SI_153]

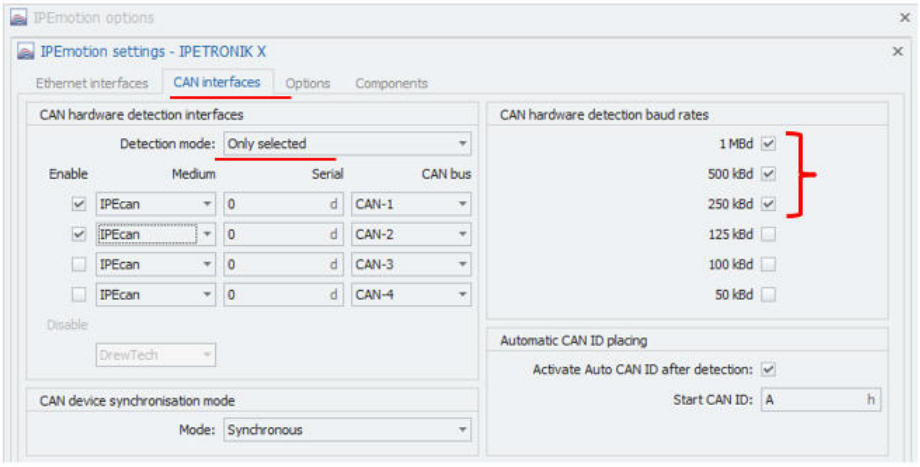
7.2 CAN interface

The CAN interface settings are relevant when you work with the CAN modules. With the setting in this dialog you have an impact on the detection process.



CAN interface settings [SI_160]

- **Enable all** — Will perform the scan for module across all CAN interfaces detected by the PC The supported CAN interfaces are managed in the CAN server. A list of the vendors and devices is provided in section 7.5. The full scan will take more time.
- **Disable all** — Will not allow any detection of the modules on a CAN card.
- **Only Selected** — Here you will perform the module scan on a dedicated on the dedicated CAN card including the serial number and CAN port. This focused scan will speed up the detection process and avoid any detection of modules connected to other CAN interfaces.
- **Exclude** — Here you define which CAN hardware should be ignored durong the scan process.

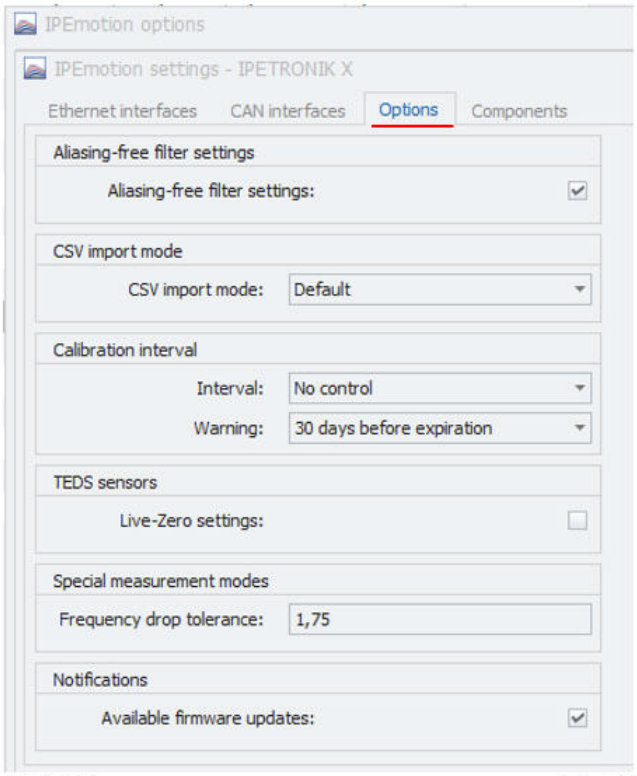


Define CAN medium priority for detection [SI_161]

With the check box for the baud rate settings you can control which baud rates should be considered for the scan process. If your modules run always on the same baud rate can focus the scan process on this dedicated rate to speed up the process.

7.3 Options

In the option settings different functions are grouped together.



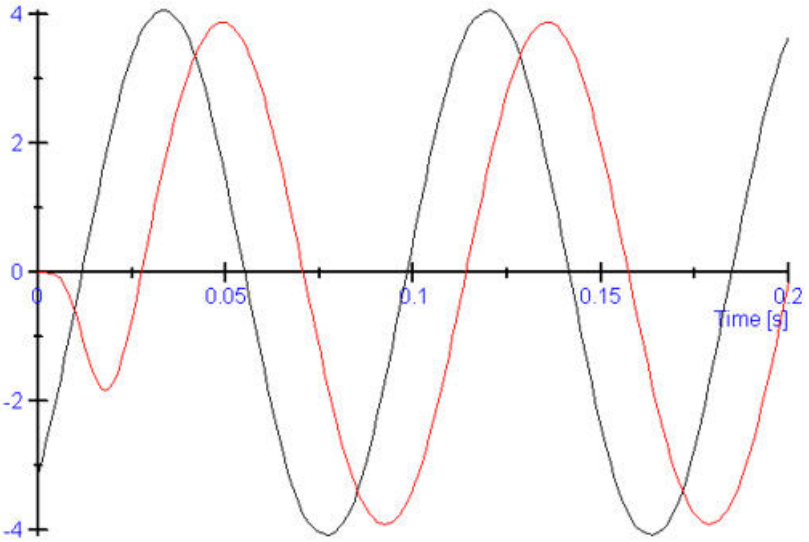
Options [SI_170]

- Alias free measurement — This checkbox has an impact on the DSP and hardware software filter settings. The function is only supported for modules which have adjustable DSP and / or hardware filter. If aliasing-free filtering is active, the software filter frequency is automatically adjusted when the sampling rate changes. The frequency is changed so that the new value is always the maximum possible frequency, where aliasing free measurement is guaranteed. If the filter frequency previously had been changed to a lower value intentionally, the filter frequency has to be changed manually by the user after the sample rate has been changed. This also applies, if the sample rate is decreased. The automatic adaptation of the software filter frequency is not applied, in case that the aliasing free measurement is disabled.

Signal filtering

Filters of analog measuring amplifiers are used for avoiding interrupting frequencies (frequency spectra, which do not contribute to the signal and/or which cannot be processed by the system). A low pass filter, which reduces the amplitudes of the frequencies above a specific cut-off frequency, is usually used for avoiding negative effects to the useful signal. The threshold in the range of the cut-off frequency (the barrier between the useful and the unrequested signal) is continuous. Useful signals below the cut-off frequency are also damped. A damping of 3 dB at the cut-off frequency means a reduction of the initial signal of 30

The image above shows the result of two inputs with the same input signal of 4 V amplitude and 12.5 Hz frequency.



Signal frequency: 12.5 Hz black: Ch2 without any filter red: Ch3 Filter 30 Hz

Impact of filter (signal shift) [SI_171]

- Channel 2: black curve without filter
- Channel 3: red curve with 30 Hz hardware filter (Bessel type). Channel 3 shows themain behaviours of filters like the damping, the phase shifting, and the start oscillation of the filtered signal.

Hardware filters

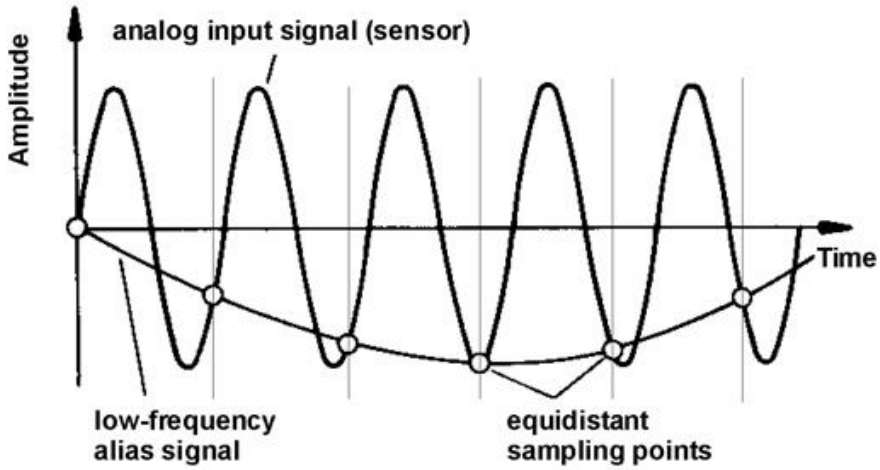
Although today's microprocessors provide a high processing power, the use of hardware filters is still essential. Especially when users cannot exclude that (periodic) signals can pass the AD converter and software filter, which cannot process the signals. Every sampling system follows Shannon's sampling theorem whereby one must at least sample with twice the signal frequency. Otherwise, aliasing effects can occur, whereas the acquired frequency is considerably lower than the actual signal (see image below).

DSP software filters

The hardware filter at the input excludes a distortion by frequency spectra above the system limit with the maximum sampling rate. Depending on the application, it can be required to lower the cut-off frequency. Example: M-SENS devices provide a switchable hardware filter with 150 Hz cut-off frequency. If the cut-off frequency is e.g. 50 Hz, interrupting frequency spectra (of devices with additional software filter) in the range between 50 Hz and the hardware filter frequency can be filtered with DSP. The filter frequency can be configured in defined steps up to the hardware filter frequency.

Alising effect

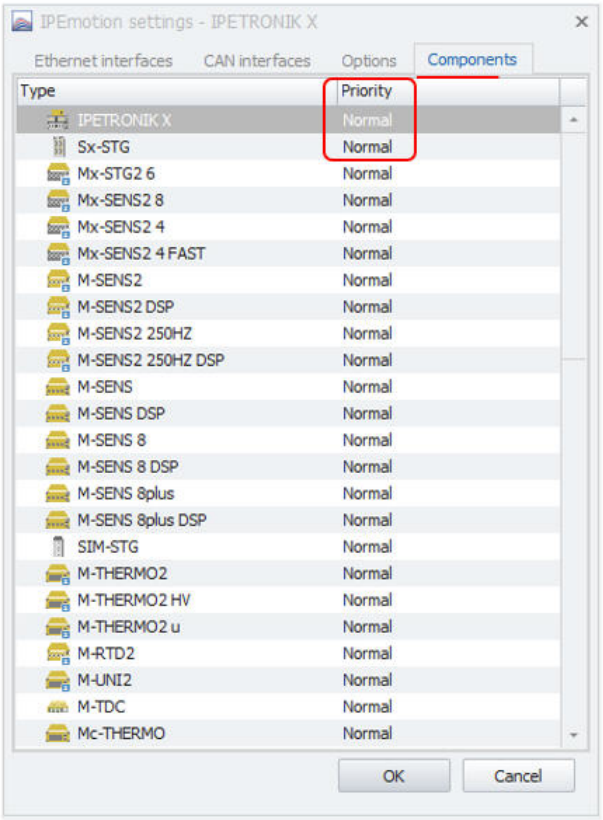
Despite sophisticated measurement engineering, errors can occur due to wrong settings. If, for example, a 100 Hz signal is acquired with a sampling rate of 100 Hz system can independently acquire the correct signal, but the result is wrong because the sampling rate was set too low I



Aliasing effect [SI_172]

7.4 Components

In the components overview you can see all supported modules. With the priority setting you have an impact on the visibility. When a module is put into the status not used it will be made invisible in the module tree for selection during the try configuration.



Components [SI_180]

7.5 CAN card hardware interfaces

List of supported CAN card interfaces

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

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

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