ENGLISH

Operating Manual

PMX

Hottinger Brüel & Kjaer GmbH
Im Tiefen See 45
D-64293 Darmstadt
Tel. +49 6151 803-0
Fax +49 6151 803-9100
info@hbkworld.com
www.hbkworld.com

Mat.:
DVS: A04354 03 E00 03
07.2022

TABLE OF CONTENTS

1 SAFETY INSTRUCTIONS

Intended use

The PMX data acquisition system, subsequently also referred to as the device, is to be used exclusively for measurement tasks and directly related control tasks within the operating limits detailed in the specifications. Use for any purpose other than the above is deemed improper use.

Any person instructed to carry out installation, startup or operation of the device must have read and understood the operating manual and in particular the technical safety instructions.

In the interests of safety, the device should only be operated by qualified personnel and as described in the Operating Manual. During use, compliance with the legal and safety requirements for the relevant application is also essential. The same applies to the use of accessories.

Notice

The device must not be directly connected to a DC voltage supply system. The supply voltage may be 10 V ... 30 V (DC).

The device may only be powered by a safety extra low voltage (safety transformer to DIN VDE 0551 or EN 60742). Only operate built-in devices once they are installed in the housing provided. Device development is guided by DIN EN 61010 Part 1 (VDE 0411 Part 1).

Before starting up, make sure that you are using a suitable supply voltage, and that the circuit you are using is sufficiently protected.

Operating conditions

- Protect the device from direct contact with water.
- Do not expose the device to direct sunlight.
- Protect the device from moisture and weather such as rain or snow. The protection class of the device is IP20 (DIN EN 60529).
- The permissible relative humidity at 31 °C is 95 % (non condensing); linear reduction up to 50 % at 40 °C.
- It is safe to operate the PMX system up to a height of 2000 m.
- The design or safety engineering of the device must not be modified without our express consent. In particular, any repair or soldering work on motherboards (replacement of components) is prohibited. When exchanging complete modules, use only genuine parts from HBM.

- The device is supplied ex works with a fixed hardware and software configuration. Changes can only be made within the range of possibilities described in the corresponding documentation.
- The device is maintenance free.
- Please note the following when cleaning the housing:
  - Disconnect the device from all current and voltage supplies.
  - Clean the housing with a soft, slightly damp (not wet!) cloth. *Never* use solvent, as this could damage the label or the housing.
  - When cleaning, ensure that no liquid gets into the device or connections.
- Old equipment that can no longer be used must be disposed of separately from normal household garbage, in accordance with national and local regulations for environmental protection and material recovery and recycling, see also *section 26 on page 441*.

Qualified personnel

Qualified persons are individuals entrusted with the installation, fitting, startup and operation of the product and with the relevant qualifications for their work.

This includes people who meet at least one of the three following criteria:

- They have knowledge of the safety equipment and procedures of measurement and automation systems, and are familiar with them as project personnel.
- They are operating personnel of measurement or automation systems and have been instructed on how to handle the machinery. They are familiar with the operation of the equipment and technologies described in this document.
- As a commissioning or service engineer, they have successfully completed training on the repair of automation plants. Moreover, they are authorized to start up, ground and label circuits and equipment in accordance with safety engineering standards.

Working safely

- Error messages should only be acknowledged once the cause of the error has been eradicated and there is no further danger.
- Maintenance and repair work on an open device with the power on may only be performed by trained personnel who are aware of the dangers involved.
- Automation equipment and devices must be designed to ensure adequate protection or locking against inadvertent actuation (e.g. access control, password protection, etc.).
- For devices operating in networks, safety precautions must be taken in terms of both hardware and software, so that an open circuit or other interruptions to signal transmission do not result in undefined states or loss of data in the automation device.

- Following work on settings or password-protected activities, make sure that any controls that may be connected remain in a safe condition until the switching behavior of the device has been tested.

Additional safety precautions

Additional safety precautions to meet the requirements of the relevant national and local accident prevention regulations must be implemented in plants where malfunctions could cause major damage, loss of data or even personal injury.

The performance and scope of supply of the device cover only a small proportion of test and measuring equipment. Before starting up the device in a plant, first perform a project planning and risk analysis, taking into account all the safety aspects of measurement and automation engineering, to minimize residual risk. This particularly concerns the protection of personnel and equipment. In the event of a fault, appropriate precautions must produce safe operating conditions.

General dangers of failing to follow the safety instructions

This is a state-of-the-art device that is safe to operate. However, there may be residual risks if the device is installed or operated incorrectly.

Important

The safety instructions are also enclosed with the device in paper form ("Documentation and Safety instructions PMX", A03260).

2 MARKINGS USED

2.1 Markings used in this document

So that you can start working quickly and safely with your product, the symbols and markings used in this manual are standardized and are explained below.

Icon	Meaning
NOTICE	This marking draws your attention to a situation in which failure to comply with safety requirements *can* lead to damage to property.
Important	This marking draws your attention to *important* information about the product or about handling the product.
Tip	This marking indicates application tips or other information that is useful to you.
Information	This marking draws your attention to information about the product or about handling the product.
•	List
▶	This prompts you to take action (a single, independent action)
1. 2. ...	Carry out this sequence of actions in the given order.
Emphasis *See …*	Italics are used to emphasize and highlight text and identify references to sections of the manual, diagrams, or external documents and files.
Device -> New	Bold text indicates menu items, as well as dialog and window headings in the program environment. Arrows between menu items indicate the sequence in which the menus and sub-menus are called up
Sample rate	Bold text in italics indicates inputs and input fields in the user interfaces.

2.2 Symbols on the device

Pay attention to the supply voltage

The symbol indicates that the supply voltage must be between 10 and 30 V_{DC}, and that you should read and follow the instructions given in this operating manual.

CE mark

With the CE mark, the manufacturer guarantees that the product complies with the requirements of the relevant EC directives (the Declaration of Conformity can be found on the HBM website (www.hbm.com) under HBMdoc).

Statutory waste disposal marking

In accordance with national and local environmental protection and material recovery and recycling regulations, old devices that can no longer be used must be disposed of separately and not with normal household garbage. See also *section 26, page 441*.

Marking of pollutant emission limit values (for deliveries to China)

Statutory mark of compliance with emission limits in electronic equipment supplied to China.

Identifier when the CODESYS software is installed

CODESYS is a software platform for programmable logic controllers. The license for CODESYS is already implemented in WG001 basic housings.

3 USER INFORMATION

Important

Obsolete documentation!

If you use an obsolete version of this document, or an obsolete version of any of the documentation it mentions, this may result in the product being mounted and operated incorrectly.

- *Make sure that all the documents you possess and use are always the current version. The current documentation version can be found at https://www.hbm.com/de/2981/pmx-modular-measuring-amplifier-system-for-the-iot/.*

3.1 Using this manual

- Read this operating manual thoroughly and in full before operating the equipment for the first time.
- The operating manual forms part of the product. Keep it in a safe place so that it is permanently accessible to all users.
- If you pass the device on to a third party, always pass it on together with the requisite documentation.

Should you lose this manual, the current version can be found on our website, at https://www.hbm.com/de/2981/pmx-modular-measuring-amplifier-system-for-the-iot/.

Failure to comply with this manual can result in personal injury or damage to equipment.

To help you quickly find the information you require, there is a full list of contents right at the front of this operating manual.

There is also an index at the end of the manual, where you can look for individual keywords.

3.2 About the PMX documentation

The documentation of the PMX data acquisition system comprises:

- this operating manual in PDF format;
- a printed Quick Start Guide;
- a printed summary of the safety instructions;
- the specifications (data sheet) in PDF format;
- a description of the functionalities and operation in the online help of the PMX web server.

Important

These documents can be found on our website, always up-to-date.

You can find additional information at https://www.hbm.com/de/2981/pmx-modular-measuring-amplifier-system-for-the-iot/ , including:

- device description files for the real-time Ethernet cards (PROFINET® IO, EtherCAT®[1]) or EtherNet/IP™[2]),
- configuration examples;
- a video tutorial on PMX.

1) EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

2) EtherNet/IP™ is a trademark of ODVA Inc. For more information regarding ODVA, visit http://www.odva.org.

4 PMX PRODUCT DESCRIPTION

By buying the PMX data acquisition system, you have chosen a high-quality HBM measurement system that is compact, powerful and variable. The data rate for all the measurement and calculation channels is 19,200 or 38,400 measurements per second. The device thus achieves an overall calculating rate of approximately 400,000 measured values per second.

A vast number of different measurement, control and automation tasks can be resolved with this measurement system.

Connection to a PC (HOST)

The PMX data acquisition system is connected to a PC via the standard Ethernet port, and can be parameterized and operated via the internal web server.

You can connect to an automation system via the digital and analog inputs/outputs, as well as via the PMX's fieldbus interfaces. This means the PMX can be easily connected to a programmable logic controller (PLC) or a higher-level automation system.

Internal calculation channels

The PMX has 32 internal calculation channels as standard, which are freely available for analysis and mathematical calculation of measurement signals. This allows you to implement automation tasks from peak values to PID control simply and elegantly.

The following types of plug-in card are available:

PX401

- The PX401 *measurement card* provides *four* individually configurable current or voltage inputs with TEDS sensor detection.
- Extreme accuracy is guaranteed, as all the channels have their own A/D converter with 24-bit resolution. This also allows the acquisition of all the channels to be totally synchronized.

PX455

- The PX455 *measurement card*, also featuring *four channels* with 24-bit resolution and TEDS sensor recognition, is available for measurement with strain gages.
- The measurement card is suitable for strain gages in both half bridge and full bridge circuits, as well as for inductive transducers in half bridge and full bridge circuits, LVDTs, potentiometric sensors, and Pt100 resistance thermometers.

PX460

- With the PX460 *frequency measurement* you can operate torque transducers (torque, rotational speed, angle of rotation), angle/incremental encoders, SSI, PWM sensors, or perform a frequency measurement up to 2 MHz.

  Channels 1 and 3: Frequency measurement (fixed)
  Channels 2 and 4: Frequency (digital/inductive), counter, encoder, SSI, PWM (adjustable)

  The following measurement modes are available:
  - up to four torque transducers (T10, T12, T40) for torque or rotational speed measurement (without detection of direction of rotation);
  - or two measurement channels for simultaneous measurement of rotational speed and angle of rotation (with detection of direction of rotation);
  - or one measurement channel for simultaneous measurement of rotational speed and angle of rotation, and detection of direction of rotation and reference pulse;
  - or two angle/incremental encoders, SSI, PWM sensors, magnetic transducers or pulse counters each;
  - or four measurement channels for frequency measurement up to 2 MHz including two shunt calibrations and two 1-wire TEDS (sensor detection).

PX878

- The *PX878 input/output card* has a total of *eight digital inputs, eight digital outputs, and five analog voltage outputs*. The PMX can be controlled by this, as well as operated with a downstream control (PLC). All real or calculated measurement signals can be freely assigned to the outputs.

PX01EC, PX01PN and PX01EP

- These interface cards can be optionally configured, and enable operation of the PMX in an automation system via the PROFINET® IO, EtherCAT®[1)] or EtherNet/IP™[2)] interfaces. Only one variant can be used in each case.

Connection technique

Transducers are connected to the amplifiers via plug terminals.

Plug terminals using push-in technology are available as standard, with a screw-on system also available as an option. If required, both types can be coded with the enclosed coding pins to prevent mix-ups.

1) EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

2) EtherNet/IP™ is a trademark of ODVA Inc. For more information regarding ODVA, visit http://www.odva.org.

TEDS (Plug&Measure)

PMX amplifiers support TEDS (Transducer Electronic Data Sheet, IEEE1451.4). The transducers are *automatically detected* when connected, and the measurement channel settings are made automatically. This efficiently minimizes setup times and user errors.

PMX web server

An easy to operate web server, specifically tailored to the PMX and matched to the measurement cards, is integrated into the device for configuration, data acquisition and visualization. This gets you quickly to your measurement result. You can visualize the data and then view it subsequently.

catman®Easy/AP PC software

Optionally, you can use HBM's catman® software to acquire, process and analyze PMX measurement data. This allows you to display and analyze very large quantities of measurement data (chart recorder function), and export it to standard formats.

Software driver

The PMX command set, a .NET API and a LabVIEW driver are provided for you to create custom applications. This allows you to implement your own operating concepts and integrate them into existing software solutions.

With the HBM LabVIEW driver, the PMX can be integrated into National Instruments software via Virtual Instruments (VI). With the HBM DIAdem driver (version 6 or higher), the PMX can be integrated into the DIAdem data acquisition software from National Instruments. LabVIEW and DIAdem are registered trademarks of National Instruments.

Device implementation

The multi-client capability of the PMX makes it possible to access the device simultaneously and without loss of speed via all the interfaces - including web server, fieldbus and analog outputs.

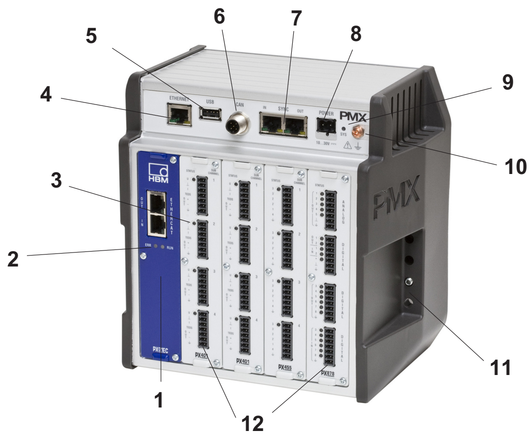
Calibration certificates

Documented quality: HBM calibration certificates according to ISO 10012 for the configured measurement cards and a declaration of compliance with the order 2.1 according to DIN EN 10204 are stored as PDF documents in the PMX's device memory on shipping. Use the PMX web server to download them from the device. You can also download the documents from the HBM website at https://www.hbm.com/de/6871/support-download-calibration-certificates/.

5 OVERVIEW, SCOPE OF SUPPLY & ACCESSORIES

5.1 The PMX system

The PMX system is a modular and universally applicable data acquisition system.



No.	Designation
1	Communication card: EtherCAT®, PROFINET® IO or EtherNet/IP™
2	Fieldbus status LEDs
3	Measurement card status LED
4	RJ45 Ethernet socket to PC/network
5	USB host
6	CAN bus (WGX001 only)
7	2x RJ45 sockets for synchronization of up to 20 modules
8	Voltage supply 10 … 30 V
9	LED system status
10	Ground connection
11	Support rail positioning
12	Max. 4 measurement cards or input/output cards, e.g.: PX455, PX460, PX878, PX401

The PMX comprises:

- Basic device
- Measurement cards
- Input/output cards
- Communication cards

The measurement cards, input/output cards and communication cards can be individually combined and configured according to the measurement task.

Basic device

Connections	Description
ETHERNET	Connection to an Ethernet network or a PC, 100 MBit/s; half and full duplex
USB	Device backup, data storage and special device functions
CAN	Local connection to CAN bus node (WGX001 only)
SYNC	Synchronizing up to 20 PMX devices
POWER	Voltage supply (10 … 30 V_{DC})

Measurement cards

Measurement card	Description	Transducers that can be connected
PX401	Current/voltage amplifier	4 current/voltage sources, individually user-selectable between current and voltage input, TEDS (1-wire)
PX455	Strain gage amplifier	4 strain gage full or half bridges (CF) The bridge excitation voltage is 2.5 V; Inductive full or half bridges, LVDT, potentiometric sensors, piezoresistive sensors, Pt100 resistance thermometers, TEDS (zero-wire)
PX460	Frequency/counter measuring amplifier	- up to four torque transducers (T10, T12, T40) for torque or rotational speed measurement (without detection of direction of rotation); - or two measurement channels for simultaneous measurement of rotational speed and angle of rotation (with detection of direction of rotation); - or one measurement channel for simultaneous measurement of rotational speed and angle of rotation, and detection of direction of rotation and reference pulse; - or two angle/incremental encoders, SSI, PWM sensors, magnetic transducers or pulse counters each; - or four measurement channels for frequency measurement up to 2 MHz including two shunt calibrations and two 1-wire TEDS (sensor detection).

Input/output cards (I/O)

Basic device, type	Interfaces	Transducers that can be connected
PX878	I/O card	8 digital inputs, 8 digital outputs, 5 analog voltage outputs, all individually configurable

Communication cards

Module	Interface	Description
PX01EC	EtherCAT®[1)] module	EtherCAT® slave
PX01PN	PROFINET® IO module	PROFINET® RT/IRT device
PX01EP	EtherNet/IP™[2)] module	EtherNet/IP™ communication adapter

1) EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

2) EtherNet/IP™ is a trademark of ODVA Inc. For more information regarding ODVA, visit http://www.odva.org.

Overview of measurement cards; input/output card

	Housing		Plug-in card						
	WGX001	WGX002	PX401	PX455	PX460	PX878	PX01PN	PX01EC	PX01EP
Number of channels (total)	-	-	4	4	4	8/5/8	-	-	-
Sample rate (samples/s)	-	-	19200	19200	38400	19200	-	-	-
Bandwidth (Hz)	-	-	3000	2000	6000	3000	-	-	-
SG full bridge				•					
SG half bridge				•					
Inductive full bridge				•					
Inductive half bridge				•					
LVDT				•					
Potentiometric transducer				•					
Resistance thermometer Pt100				•[2])					
Current-fed piezoelectric transducer (IEPE)			• [1])						
Piezoresistive full bridge				•					
Analog input, voltage			•						
Analog input, current			•						
5 analog outputs						•			
8 digital inputs						•			
8 digital outputs						•			
Frequency measurement, pulse counting					•				
Torque/speed					•				
Incremental encoder					•				
Angle of rotation with ref. pulse					•				
SSI encoder					•				
Inductive rotary encoders					•				
PWM					•				
EtherCAT								•	
PROFINET							•		
EtherNet/IP									•
CANopen	•								
	•								

1) A smart module (1-EICP-B-2) is required for the connection of IEPE transducers
2) In conjunction with 100 ohm completion resistor

5.2 Scope of supply

Description	Ordering number
1 PMX basic device, with wall mount kit (1 wall bracket, 4 screws, 4 washers) and support rail mounting material as well as 2 strips for cable mounting with screws and washers. With CAN connection and CODESYS-V3 Soft PLC Without CAN connection and without CODESYS	 1-WGX001 1-WGX002
For each measurement card: one mating plug per channel, all mating plugs push-in (4 plugs including coding pins supplied with each measurement card)	1-CON-S1008 1-CON-S1012 for PX460
Support rail mounting kit (2 kits, packed in film cushion packaging with the mounting material in an Etimex bag) (4 fitting screws M5x10, 4 spring washers)	1-RAILCLIP
PMX operating manual and data sheet, safety instructions and Quick Start Guide	
With WGX001: Supplied with CODESYS CD (CODESYS V3 software, PMX package Quick Start Guide and program examples)	
Mating plug M12x1 for CAN interface on WGX001	1-CON-S1002
Mating plug for PMX voltage supply (WGX001/WGX002)	1-CON-S1010

5.3 Accessories

Accessories	Ordering number
Ethernet crossover cable for direct operation of devices on a PC or laptop, length 2 m, type CAT5+	1-KAB239-2
AC/DC power supply unit; input: 90 V … 264 V_{AC}, 1.5 m cable, output: 24 V_{DC}, max. 1.25 A, 2 m cable with ODU plug	1-NTX001
Connection clamp ME-SAS MINI - 2200456 from PHOENIX for strain relief of the transducer cable.	1-CON-A1023

Replacement parts	Ordering number
PX01, PMX blank plate, blue, for plug-in card slot 0	1-PX01
PX02, PMX blank plate, gray, for plug-in card slots 1-4	1-PX02
RAILCLIP, PMX support rail mounting kit (x2), incl. screws	1-RAILCLIP
Phoenix plug terminals	
Set of plug terminals (push-in) for PMX plug-in cards (4 x 7-pin, incl. coding plug and labeling sheets)	1-CON-S1008
Set of screw terminals PMX voltage supply (1 x 2-pin, incl. coding plug and labeling sheets)	1-CON-S1010
Set of plug terminals (push-in) for PMX plug-in cards (2, 13 and 2-pin, incl. coding plug and labeling sheets)	1-CON-S1012
Mating plug M12x1 for CAN interface on WGX001	1-CON-S1002

In general, the mating plugs are always included for all plug-in cards (PX401, PX455, PX460 and PX878).

On ordering a PMX basic device, the scope of supply always includes support rail mounting material and a wall mount kit.

Important

You have the option to retrofit or subsequently remove all measurement cards, I/O cards and communication cards.

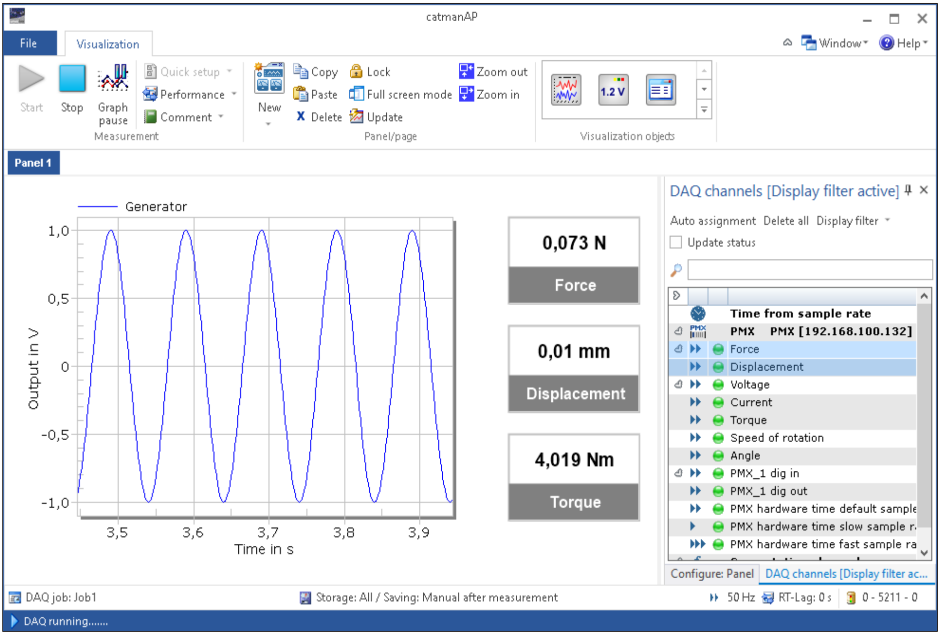
5.4 PMX web server and software

A PMX web server, including Help, is integrated into the device. The web server also has a function which can download new PMX firmware and web server versions to the PMX.

The web server includes a help for operating and managing the PMX (click on the Help icon in the top right corner of the overview menu).

catman®Easy/AP PC software

Optionally, you can use HBM's catman® software to acquire, process and analyze PMX measurement data. This allows you to display and analyze very large quantities of measurement data (chart recorder function), and export it to standard formats (see following illustration).



All real and calculated measurement channels are measured, as are the digital inputs and outputs. Digital inputs and outputs are represented as binary coded values.

The PMX supports up to three sample rates, which can be set independently of each other. The sample rates can then be assigned to individual measurement signals.

To start and stop a measurement (triggering), you can use times of day, PMX digital inputs or outputs, or trigger by way of limit values in catman®.

With catman® the PMX can be parameterized at least partially. This includes:

- setting the sensor type, via the sensor database or using TEDS;
- writing to the TEDS sensors using the TEDS EDITOR integrated into catman® ;
- zeroing the measurement signal and setting the filter frequency for each individual channel.

You can use the catman®Script programming language to program complete measurement sequences, including automated storage of measured data and creation of logs.

For more information, see the Online Help in catman®Easy/AP.

Important

The PMX device settings are permanently stored in the PMX's active parameter set after catman® is closed. catman® changes the sensor settings (sensor type, scaling, filter) automatically in the PMX.
Before starting catman®, enable retention of the PMX filter setting in the **Prepare new DAQ project** *dialog:* ***Do not change current sample rates and filter settings of devices on device scan***

Software driver

The PMX command set, a .NET API and a LabVIEW driver are provided for you to create custom applications. This allows you to implement your own operating concepts and integrate them into existing software solutions.

With the HBM LabVIEW driver, the PMX can be integrated into National Instruments software via Virtual Instruments (VI). With the HBM DIAdem driver (version 6 or higher), the PMX can be integrated into the DIAdem data acquisition software from National Instruments. LabVIEW and DIAdem are registered trademarks of National Instruments.

The following functions are supported by the drivers in PMX:

Function	Description
Device scan	Scan the Ethernet network
Measurement configuration	Set sample rate, filter, zero point
Sensor configuration	Set the scaling (2-point) or via TEDS
Analog input for data acquisition and calculated channels (streaming)	Read all measured values and time stamps of sensors and channels
Status information (diagnosis)	Read each channel and device status
Peak values	Read or delete peak values
Limit switches	Read or set limit value switches
Analog output (direct setting)	Read or set analog outputs (10 V)
Analog output (configuration)	Set source, scaling
Digital input for data acquisition	Read and set digital inputs (High/Low)
Digital input for data acquisition (direct setting)	Read and set digital outputs (High/Low)
CAN data acquisition (via CODESYS/ calculated channels)	Read calculated channels with CAN signals
Parameter sets	Read and select parameter sets

Information

Driver versions 2.0 or higher are required for firmware version 2.00 or higher.

Tip

*All commands in the PMX command set can be used as low-level commands (**see chapter 21, "PMX command set", page 346**).*
Detailed support and programming examples can be found in the program help of the individual drivers.
You can download all drivers, and also the catman® software, as a free 30-day trial version from the HBM website:
https://www.hbm.com/

6 PROTECTION / HOUSING / SHIELDING DESIGN

The degree of protection given in the specifications indicates the suitability of the device for various ambient conditions and also the protection it gives people against potential risks when they are using it. The letters *IP* (International Protection) which are always included in the designation are followed by two digits. These indicate what degree of protection a housing offers against touch contact and foreign objects (first digit) and moisture (second digit).

All PMX modules and the basic device have IP20 protection (as per EN 60529).

IP 2 0

Code index	Degree of protection against contact and foreign objects	Code index	Degree of protection against water
2	Protection against contact with fingers, protection against foreign objects with ∅> 12 mm	0	No water protection

Greenline shielding design

HBM has developed the *Greenline* shielding design to improve protection against electromagnetic interference. The layout of the cable shield means that the complete measuring chain is enclosed by a Faraday cage. With double-shielded sense leads, connect the shields by laying them together.

Notice

Observe the maximum cable lengths, and the completion resistors that may have to be mounted depending on the amplifier type and cable length.
See section 8.4, "Measurement cards/transducer connection", page 62.

6.1 Cabinet mounting

Lay the cable shields directly at the control cabinet input on a grounding bar and route the sense leads as short as possible to the PMX (*Fig. 6.1*). Also connect the PMX to the grounding rail via the grounding terminal on the PMX housing (*Fig. 6.2*), and ground the control cabinet and grounding rail.

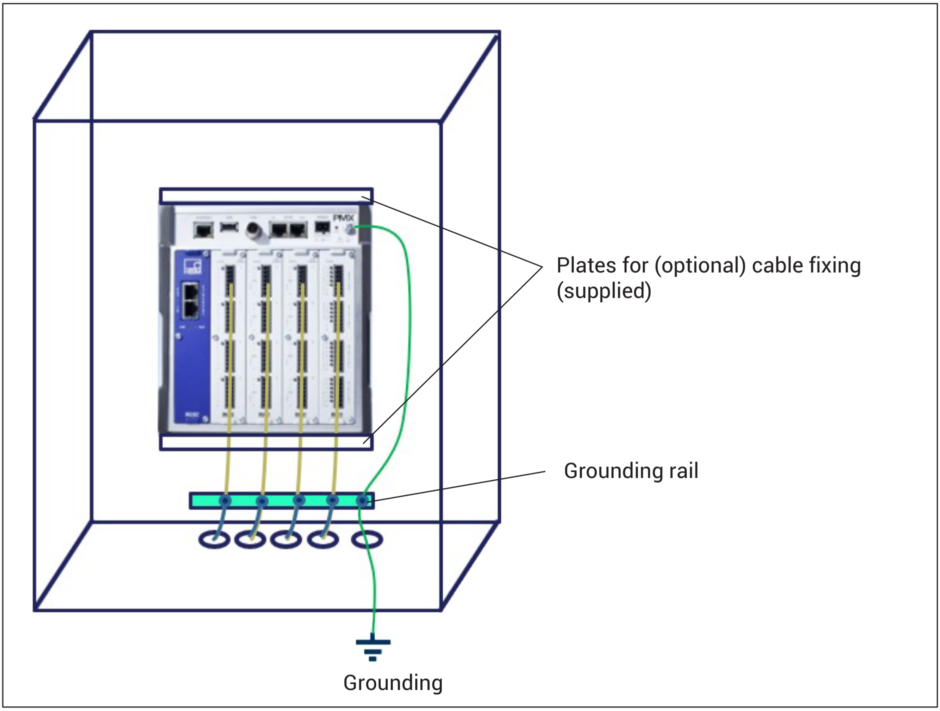


Fig. 6.1 Cabinet mounting with grounding rail

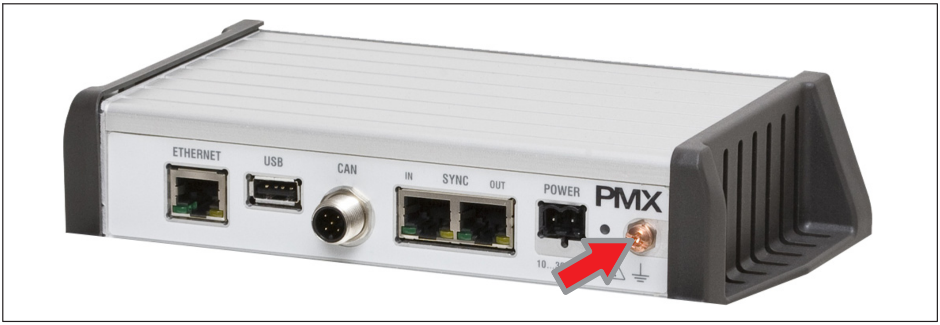


Fig. 6.2 Grounding terminal on PMX housing

6.2 Free mounting

Place the cable shields on the shield connection of the PMX terminals. If possible, use stranded wire and insulate the transition point from the shield to the connecting strand, such as with heat-shrink tubing (*Fig. 6.3*). Be sure to keep the sense leads as short as possible after the shield as far the plug.

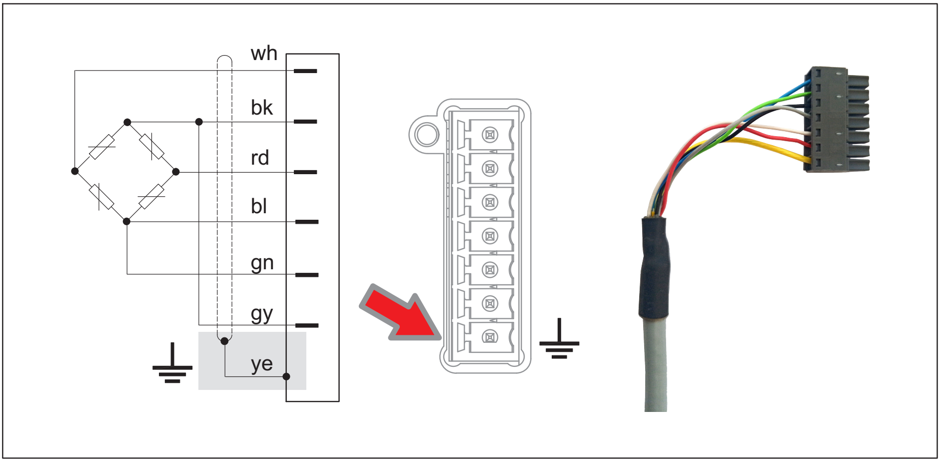


Fig. 6.3 Shield connection; right: Shield (yellow cable) on lower terminal

You can also, for example, use the ME-SAS MINI - 2200456 connection clamp from PHOENIX 1-CON-A1023, which simultaneously provides strain relief for the sensor cable (*Fig. 6.4*). Mount the connection clamp by the metal bracket into the lower connection on the plug.

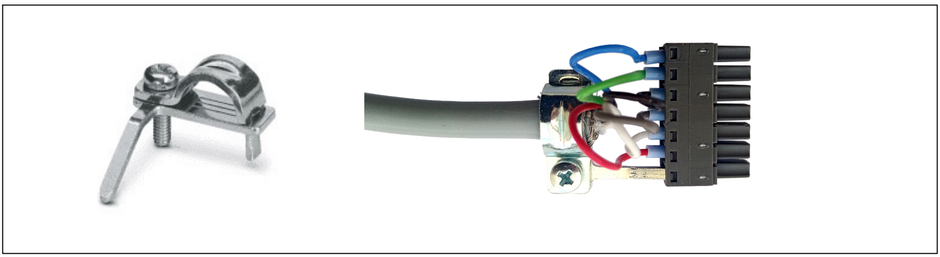


Fig. 6.4 Shield connection clamp; right, with mounted cable

You can also achieve strain relief using the supplied cable fixing plates at the top or bottom of the PMX (*Fig. 6.5*).

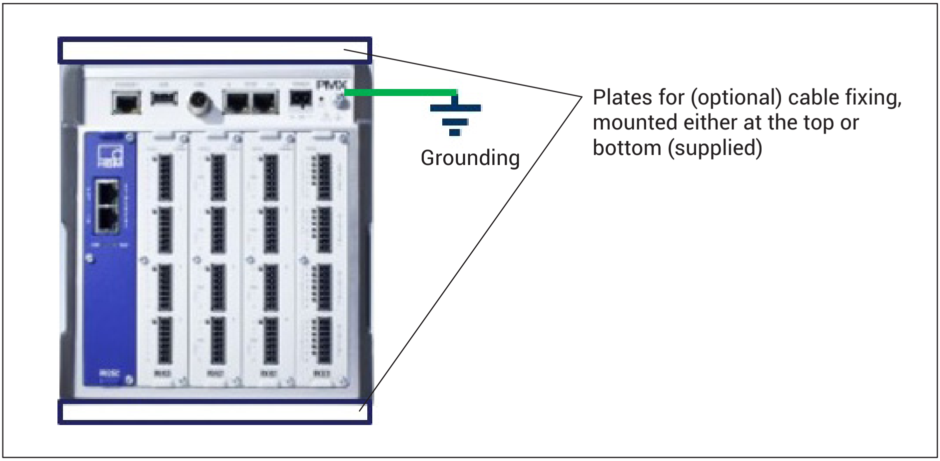


Fig. 6.5 Grounding and strain relief for cables

7 MOUNTING/DISMOUNTING/REPLACING

7.1 Assembly tools and tightening torques

Mounting	Required tool	Tightening torque
Fastening the rail clip to the support rail M5 hexagon socket screw	Hexagon socket wrench a.f. 2.5	1.0 … 1.2 Nm
Fastening the support rail mounting to the housing M5 hexagon socket screw	Hexagon socket wrench a.f. 3	3 Nm
Fastening the plug-in card M 2.5 Torx screws	Torx screwdriver TX8	0.5 … 0.6 Nm
Mounting wall bracket M4 hexagon socket screw	Hexagon socket wrench a.f. 3	1.5 … 2 Nm
Fastening the side panels M3 Torx screws	Torx screwdriver TX10	0.8 … 1 Nm
Grounding screw on the PMX M4 Torx screws	Torx screwdriver TX20	1.5 … 2 Nm
Cable fastening plates M4 hexagon socket screw	Hexagon socket wrench a.f. 3	1.5 … 2 Nm

7.2 Support rail mounting

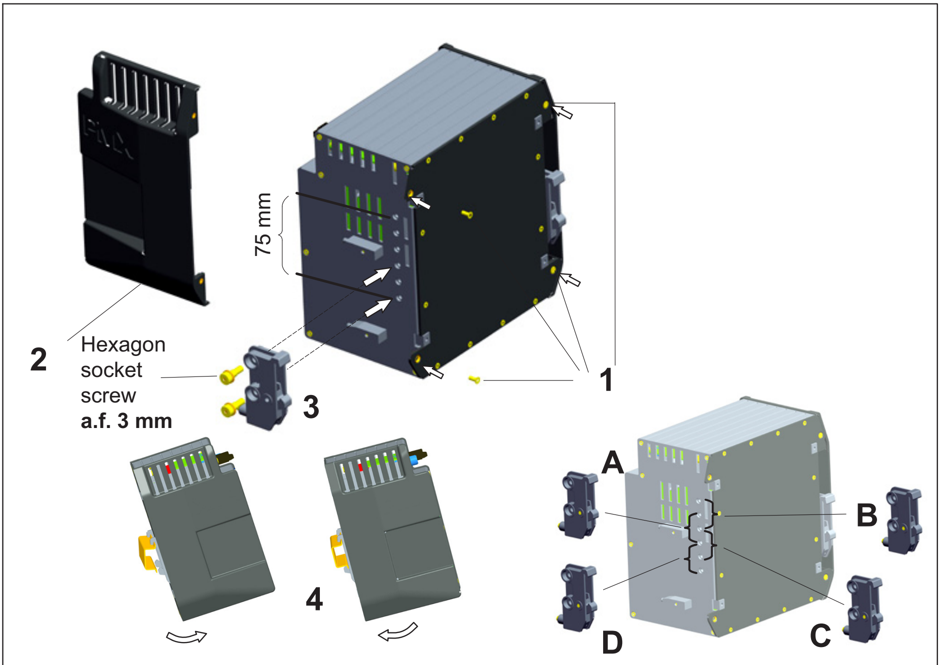


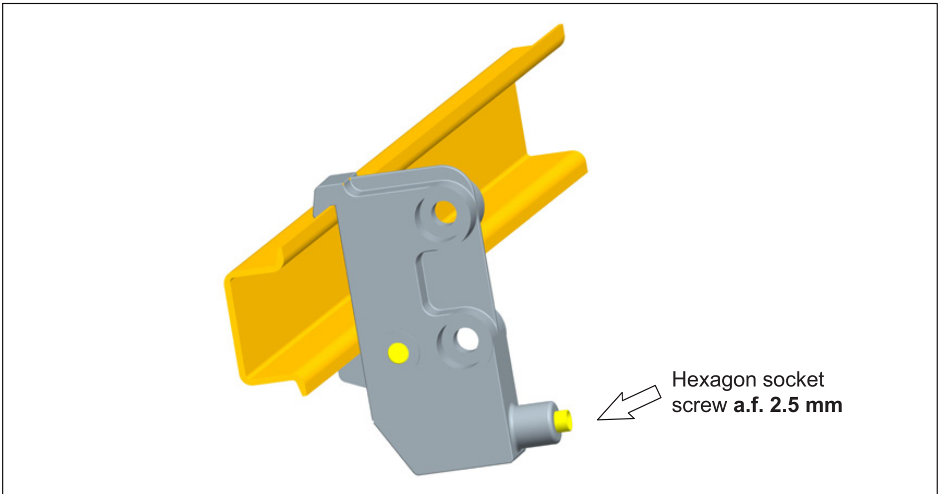
Fig. 7.1 Mounting on a support rail

1. Loosen the four rear panel screws (Torx Tx10) **(1)**.
2. Push the side panels forward **(2)**.
3. Screw on the support rail mounting **(3)** (about 5 Nm). Four positions (**A** to **D**) are optionally possible (two positions for 7.5 mm rail).
4. Screw the side panels **(2)** back on.
5. Attach the PMX to the support rail **(4)**.

Notice

Device damage by dropping the PMX due to difficulty of attaching/detaching the PMX. HBM recommends using a DIN support rail (DIN EN 60715) with a height of 15 mm. When using a smaller support rail (7.5 mm high), it should be packed, to make it easy to attach/detach the PMX.
The 7.5 mm support rail can only be used in the top two positions (A and B).

Fastening the support rail mounting (rail clip) to the support rail



On delivery, the self-locking (2.5 mm) hexagon socket screws are *unscrewed* as far as the stop.

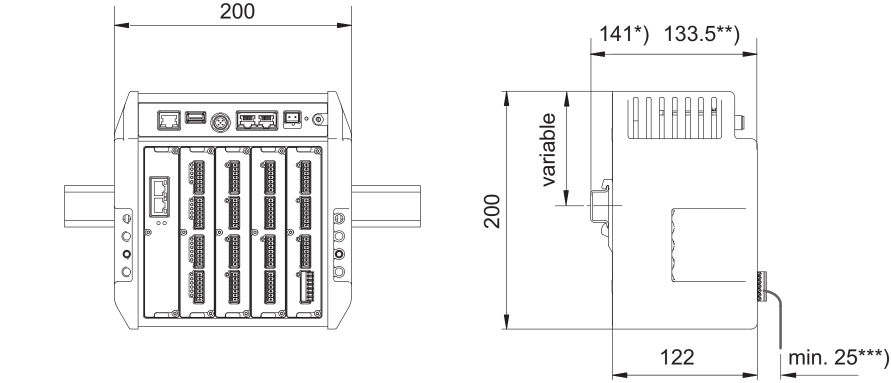
- Clamp on the support rail mounting (rail clip).
- Hand-tighten the self-locking hexagon socket screw.

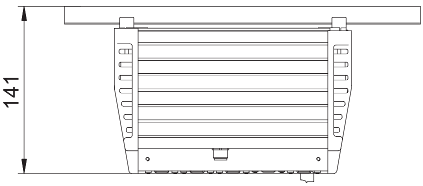
Notice

Device damage caused by electromagnetic irradiation of external devices. Faulty measurements due to electromagnetic irradiation from other devices.
To ensure sufficient grounding of the PMX, the support rail must be connected to functional ground ⏚.
Both the support rail and the PMX must be free of paint and dirt at the mounting location.

- *Connect the PMX housing to ground via the grounding screw.*

Dimensions and mounting instructions





*) Height of support rail 15 mm
**) Height of support rail 7.5 mm
***) Min. dimension: Plug plus sensor cable

NOTE:

To ensure sufficient ventilation/cooling, a 2 cm gap must be maintained above and below neighboring devices. Dimensions in mm.

7.3 Mounting the wall bracket

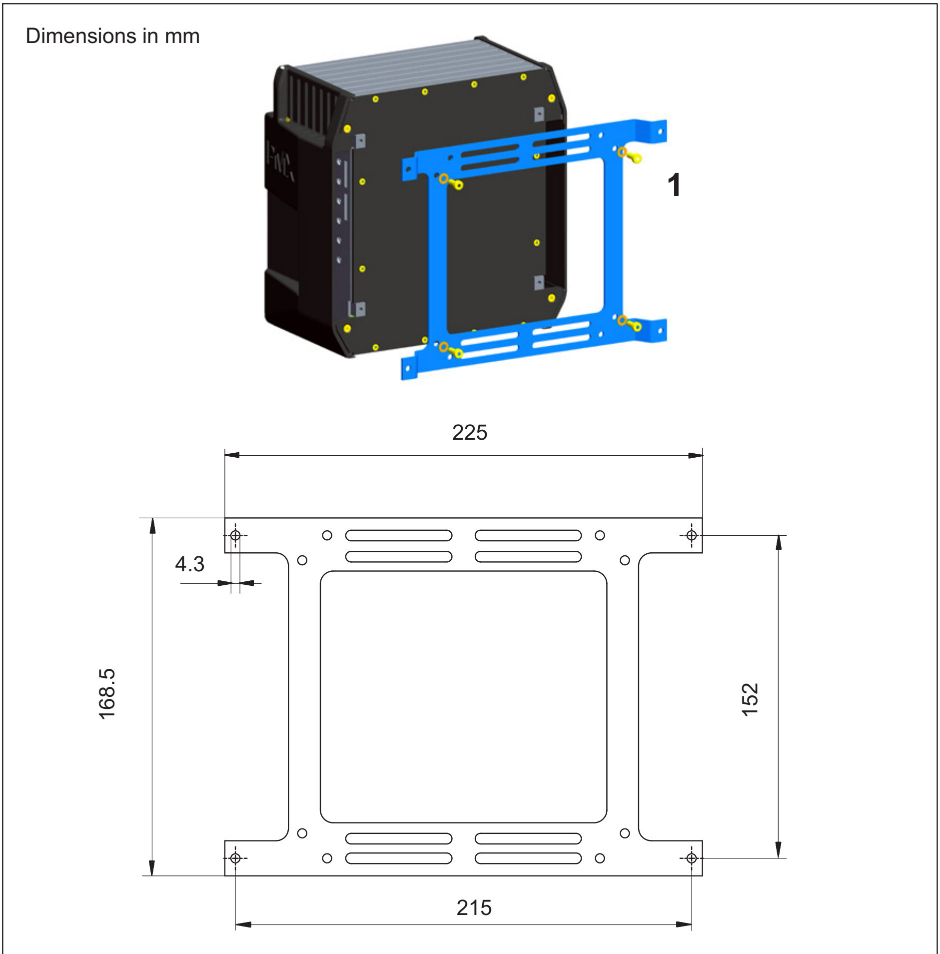
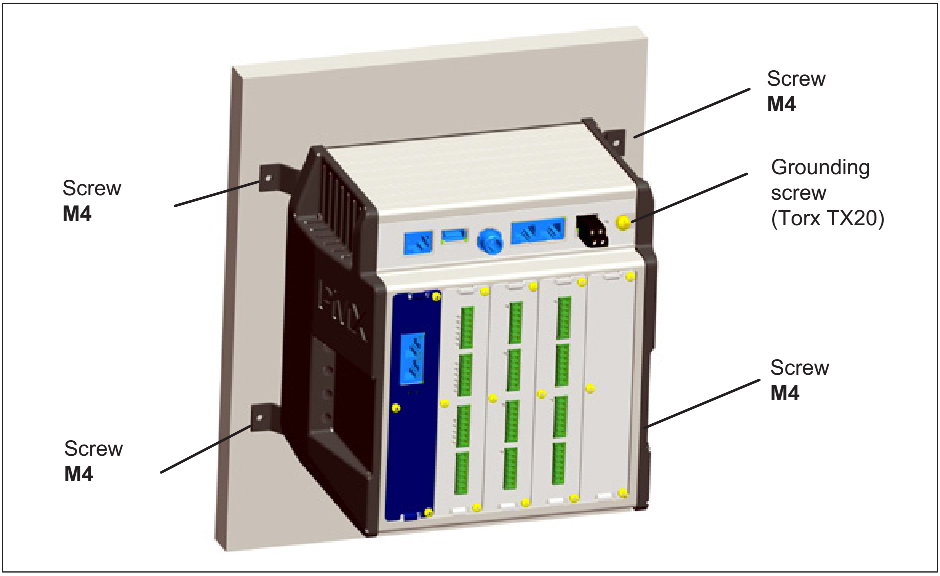


Fig. 7.2 Mounting on a wall

1. Attach the wall bracket to the back of the PMX by the supplied M4 screws **(1)**.



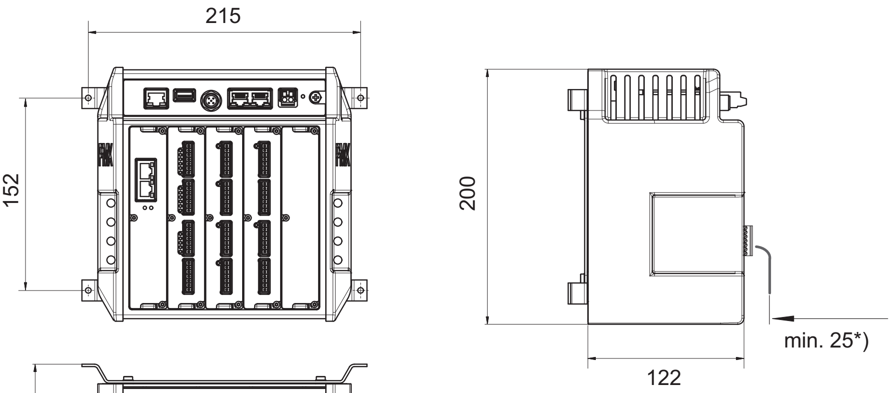
2. Screw the complete unit to the wall. The hole diameter is 4 mm.

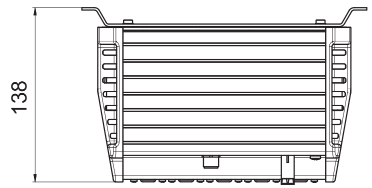
Notice

Device damage caused by electromagnetic irradiation of external devices. Faulty measurements due to electromagnetic irradiation from other devices.
The housing must also be connected to functional ground ⏚ when wall-mounted.

- *Connect the PMX housing to ground via the grounding screw.*

Dimensions and mounting instructions





*) Min. dimension: Plug plus sensor cable

NOTE:

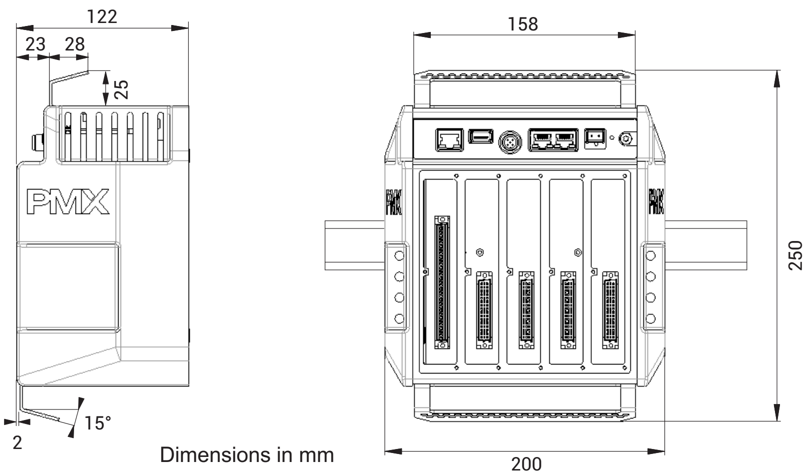
To ensure sufficient ventilation/cooling, a 2 cm gap must be maintained above and below neighboring devices. Dimensions in mm.

7.4 Installing cable fastening plates (optional)

Fig. 7.3 PMX with cable holder

To ensure that cables running from and to the PMX are fastened securely and reliably, an optional plate can be secured to the top and bottom of the PMX mainframe to fasten the cables using two M4 hexagon socket screws for each plate.

Holes in the plate can be used to fasten the cables using cable ties.



7.5 Replacing measurement and communication cards

Measurement and communication cards can be retrofitted or removed. Please note the combination options (see *page 49*).

After modification, and switching on the supply voltage, the PMX automatically detects and initializes the hardware configuration. The factory settings are loaded. All parameters must be re-entered, including for the existing cards.

Tip

So back up the parameter sets to your PC to be on the safe side. Use the free Parameter Set Reader for PMX (download from PMX website) to convert the device settings into a readable TXT file.

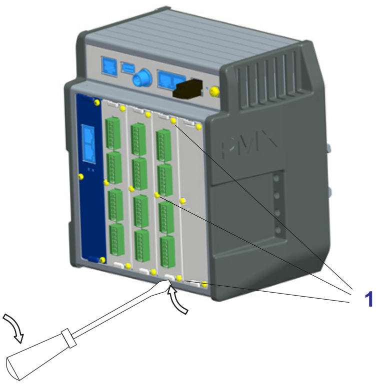
Notice

Removing/replacing measurement or communication cards incorrectly can damage or even destroy them.

The cards must only be removed or replaced ***with the power off****.*

- *Always disconnect the PMX from the power supply before removing a card. Note that device parameters must be reset when cards are added.*

Removal



1. Undo the three M2.5x8 Torx (Tx8) screws (**1**) of the card/blank plate.
2. Use a screwdriver to lever the card at the lug provided.
3. Carefully take out the board.

Installation

1. Carefully insert the board into the PMX slot (ribs prevent tilting).
2. The board centers itself in the VG connector strip at the back.
3. Retighten the three M2.5 screws.

Notice

Device damage caused by electromagnetic irradiation of external devices.
Faulty measurements due to electromagnetic irradiation from other devices.

▶ *Close off the open slots with blank plates (accessories).*

8 PMX ELECTRICAL CONNECTIONS

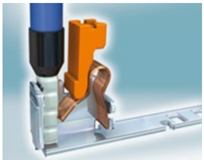
8.1 Plug connection technology and clamping areas

All PMX plug-in cards (PX401, PX455, PX460, PX878) are supplied as standard with easy-fit push-in plug terminals. But you can also obtain screw-type terminals from Phoenix Contact (www.phoenixcontact.com, BK = black variant), e.g.:

- MC 1.5/2-ST-3.5 BK for power supply to PX460;
- MC 1.5/7-ST-3.5 BK for connecting sensors to PX401 and PX455, and for the digital inputs and outputs on PX878;
- MC 1.5/13-ST-3.5 BK for connection to PX460.

Other variants, such as with locking clips, are also available from Phoenix Contact, e.g. MCVW 1.5/..., MCVR 1.5/..., FK-MCP 1.5/...

Push-in technology



The clamping area is 0.2 mm^2 (AWG24) to 1.5 mm^2 (AWG16). If you need to connect multiple wires to one terminal, adapt the wire cross-sections accordingly. Use 10 mm wire end ferrules (without plastic collars) to connect the wires to the terminals wher-ever possible.

Notice

The plug terminals are not interchangeable ex factory. Depending on the sensor type, plug connection errors can damage the plug-in card.
Use the supplied coding pins to prevent interchanging.

The plug terminals can be protected by coding pins against interchanging. To do this, insert a coding pin fully into one of the slots in the device sockets and snap it off from the holder - see *Fig. 8.1*. Use a different slot for each plug terminal and transducer type. You can also use more than one coding pin for one plug terminal.

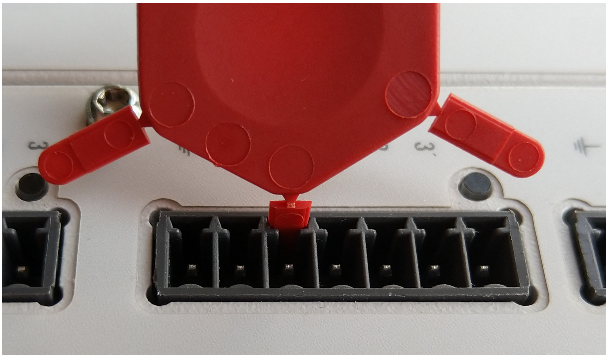


Fig. 8.1 Coding pin 90% inserted

Remove the lug on the corresponding plug terminal connector, using a knife for example (*Fig. 8.2*).

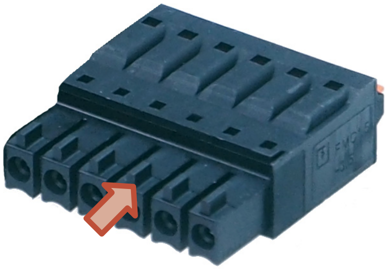


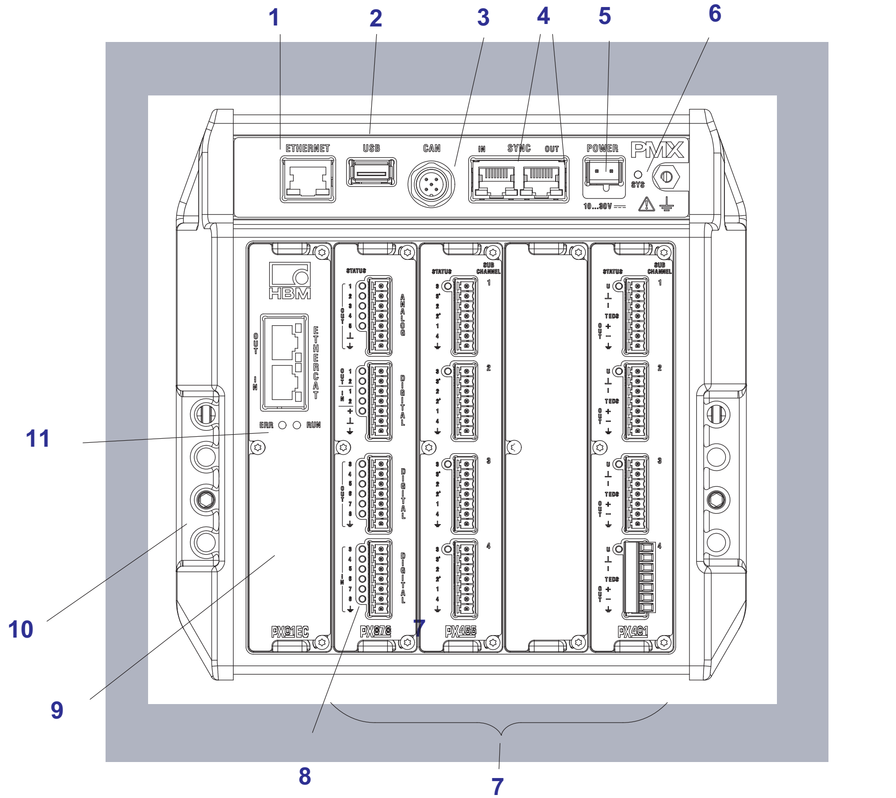
Fig. 8.2 Lug (arrow) on a plug terminal (zoomed view)

Attach the shield of the transducer cable to the ground connection provided on the PMX's multipoint connector, in accordance with HBM Greenline information https://www.hbm.com/Greenline.

Important

⏚ *The ground terminal on the PMX is not a protective ground (connection optional). The measurement system features automatic current limitation for each device card and for the PMX basic device.*

8.2 Overview of PMX functions



1 Ethernet plug for PC/network connection

2 USB host, e.g. for a memory stick

3 CAN for CAN driver, M12, option (WGX001)

4 2 x RJ45 for synchronization

5 Bridge excitation voltage 10 … 30 V_{DC}

6 Device status LED

7 Measurement cards (PX401, PX455, PX460) and/or PX878 I/O card *or* blank plate

8 Measurement card status LED

9 Communication cards: PX01EC (EtherCAT®), PX01PN (PROFINET® IO, PC01EP EtherNet/IP™) *or* blank plate

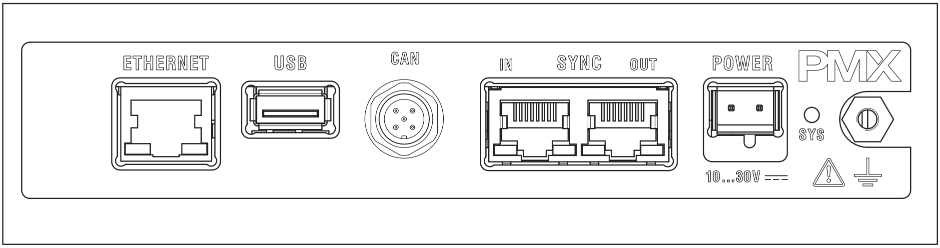
10 Support rail positioning

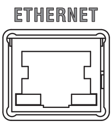
11 Fieldbus LED

8.2.1 Combination options for input cards

	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Number of plug-ins
Fieldbus or blank plate	x	-	-	-	-	0 - 1
PX401	-	x	x	x	x	0 - 4
PX455	-	x	x	x	x	0 - 4
PX460	-	x	x	x	x	0 - 4
PX878	-	x	x	-	-	0 - 2

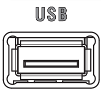
8.2.2 Meanings of the basic device connector sockets





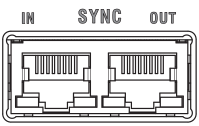
PC or network connection.

Cable: Ethernet cable Cat 5, SFTP

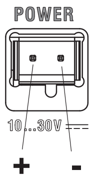


USB port version 2.0 e.g. for mass storage device, scanner, USB flash drive

Cable: standard USB cable

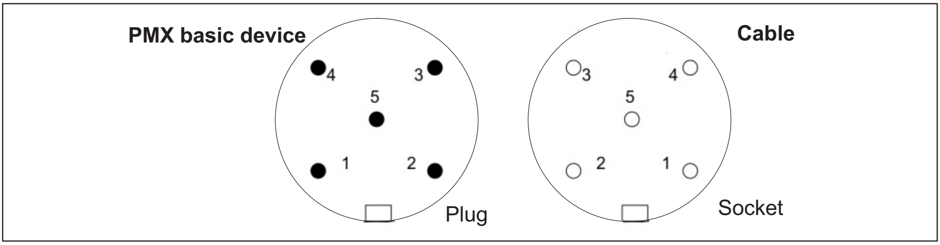


Synchronizing multiple (maximum 20) PMX devices via two RJ45 sockets, see *section 8.1, "Plug connection technology and clamping areas", page 46*.

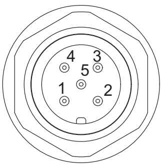


(view of device front)

Supplying voltage to the PMX by connecting a separate DC voltage supply.



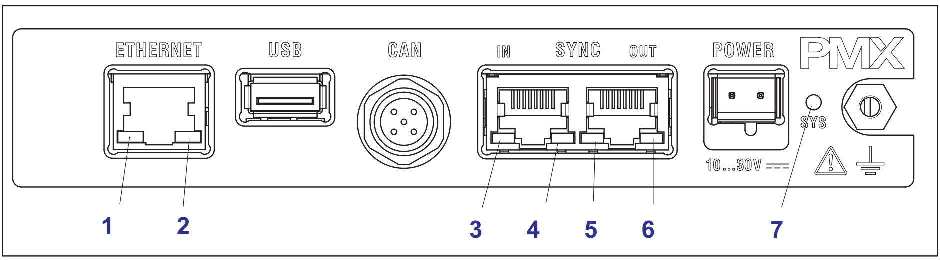
CAN connection (for type WGX001 only)



Pin	Signal	Description
1	SHLD	CAN shield
2		Not connected
3	GND	Ground
4	CAN_H	CAN_H data cable (high)
5	CAN_L	CAN_L data cable (low)

8.2.3 LEDs for system monitoring (device LED)

Basic device (WGX001/002)



ETHERNET LED (1, 2)

LED	LED	Status	Meaning
Ethernet Link (1)	Green	Steady	Connection present
Ethernet RX / TX (2)	Yellow	Flashing	Data are being transmitted

SYNC IN / OUT (3, 4 and 5, 6)

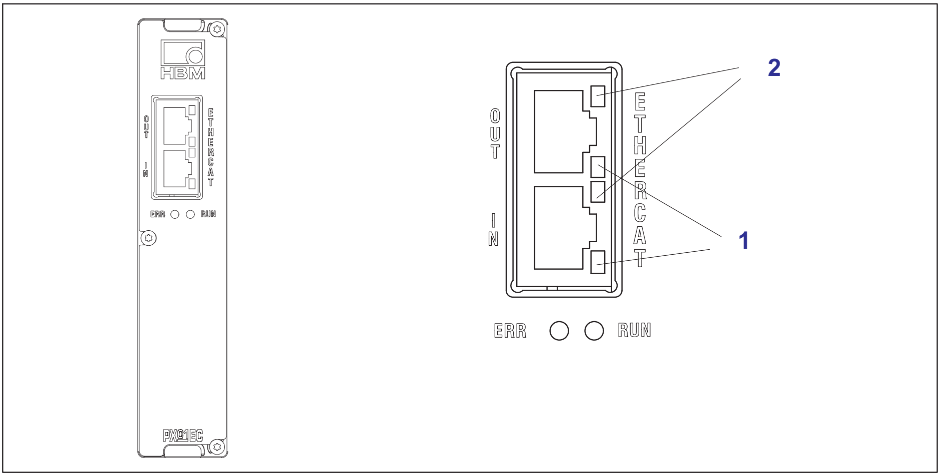
LED	LED	Status	Meaning
IN (3)	Green	On	Slave
IN (4)	Yellow	On	Error
IN (3 + 4)		Off	Master
OUT (5)	Green	On	Always on
OUT (6)	Yellow	On	Error (always identical to the right-hand LED of the IN socket)

SYS LED (7)

LED	Status	Meaning
Green	On Off	Voltage supply available Voltage supply missing
Yellow	On Flashing	Device is booting Factory settings not OK
Red	Flashing On	Serious internal error Firmware updates

8.2.4 Fieldbus LEDs

PX01EC52



EtherCAT®

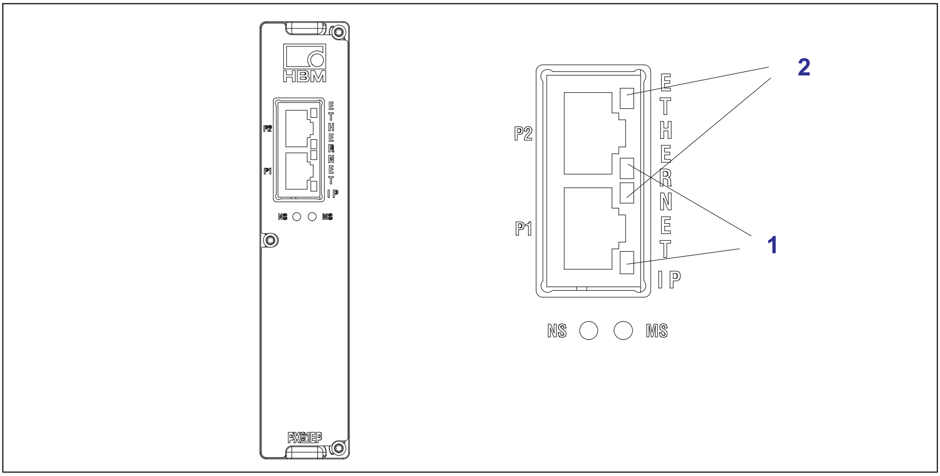
LED	LED	Status	Meaning
ERR	Red	Off	No error
	Red	Flashing	Configuration error
	Red	Single flash	Synchronization error
	Red	Double flash	Application timeout error
	Red	On	PDI timeout error

LED	LED	Status	Meaning
RUN	Green	Off	INIT status
	Green	Flashing	PRE OPERATIONAL status

LED	LED	Status	Meaning
	Green	Single flash	SAFE OPERATIONAL status
	Green	On	OPERATIONAL

LED	LED	Status	Meaning
1	Green	Steady Flashing Off	Connection established Send / Receive No connection
2	-	-	No function

PX01EP



EtherNet/IP™

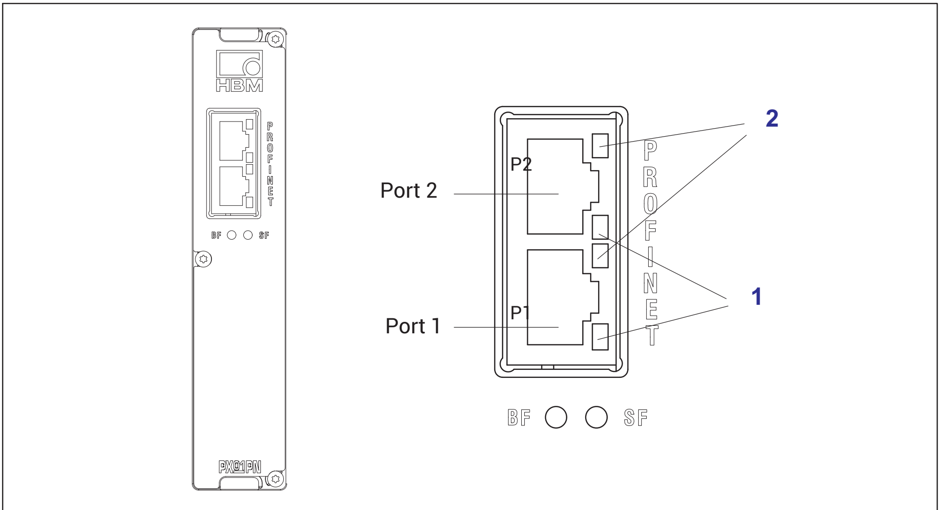
LED	LED	Status	Meaning
LV	Green	On	**Connected**: If the device has at least one active connection (even to a message router), the network status indicator is lit steadily green.
	Green	Flashing	**No connections**: If the device does not have any active connections, but has received an IP address, the network status indicator flashes green.
	Red	On	**Duplicate IP***: If the device determines that the IP address has already been used, the network status indicator is lit steadily red.
	Red	Flashing	**Connection timeout**: If one or more of the connections to this device is/are timed-out, the network status indicator flashes red. This status is only terminated when all timed-out connections have been restored, or when the device is reset.
	Red Green	Flashing	**Self-test**: While the device is performing the self-test, the network status indicator flashes green/red.
	-	Off	**Not switched on, no IP address**: If the device does not have an IP address (or is not switched on), the network status indicator is unlit.

LED	LED	Status	Meaning
MS	Green	On	**Device ready for operation**: When the device is operational and working correctly, the network status indicator is lit steadily green.
	Green	Flashing	**Standby**: If the device has not been configured, the module status indicator flashes green.
	Red	Flashing	**Serious error**: If the device detects an irreparable serious error, the module status indicator is lit steadily red.

LED	LED	Status	Meaning
	Red	Flashing	**Simple error***: If the device detects a reparable simple error, the module status indicator flashes red. NOTE: A misconfiguration, for example, is classified as a simple error.
	Red Green	Flashing	**Self-test**: While the device is performing the self-test, the module status indicator flashes green/red.
	-	Off	**Not switched on**: If the device is not switched on, the module status indicator is unlit.

LED	LED	Status	Meaning
1	Green	On	Connected to Ethernet
	-	Off	The device has no connection to Ethernet
2	Yellow	Flashing	The device is sending/receiving Ethernet frames

PX01PN



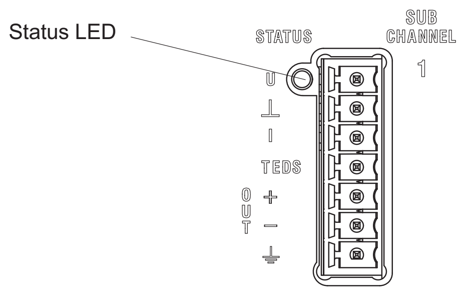
PROFINET® IO

LED	LED	Status	Meaning
SF	Red	On Flashing	System error, incorrect configuration Flashing for device detection is controlled by the IO controller
BF	Red	On Flashing	No connection or no configuration Bus error, incorrect configuration, not all IO devices are connected

LED	LED	Status	Meaning
1	Green	Steady Flashing Off	Connection established Send / Receive No connection
2	-	-	No function

8.2.5 Measurement card LEDs

PX401, channel status



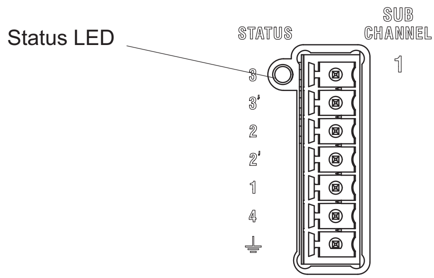
LED	Status	Meaning
Green	On	No errors
Yellow	Flashing	Firmware updates
Red	On	Parameter not OK, overloaded

Measuring range monitoring

By default, all inputs are checked for out-of-range (before a possibly set filter). The allowed measuring ranges are dictated by the specified sensor type. If the range is exceeded, the measured value becomes invalid.

Sensor type	Allowed measuring range
±10 V	±11.0 V
±20 mA	±21.0 mA
4 … 20 mA	3.9 … 21.0 mA

PX455, channel status



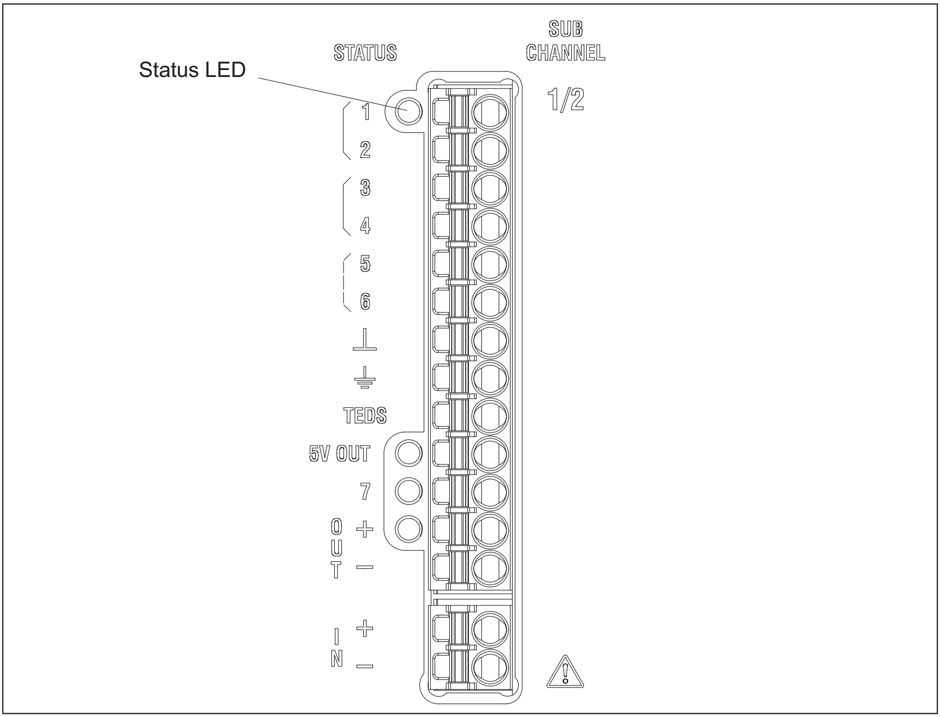
LED	Status	Meaning
Green	On	No errors
Yellow	On	No transducer connected or wire break (calibration in progress)
	Flashing	Firmware updates
Red	On	Parameter not OK, transducer error, overloaded

Measuring range monitoring

By default, all inputs are checked for out-of-range (before a possibly set filter). The allowed measuring ranges are dictated by the specified sensor type. If the range is exceeded, the measured value becomes invalid.

Sensor type	Allowed measuring range
Full bridge 1000 mV/V	±1100 mV/V
Half bridge 1000 mV/V	±550 mV/V
Full and half bridge 100 mV/V	±110 mV/V
Full and half bridge 4 mV/V	±4.5 mV/V
Potentiometer	±550 mV/V
LVDT	±550 mV/V

PX460, channel status



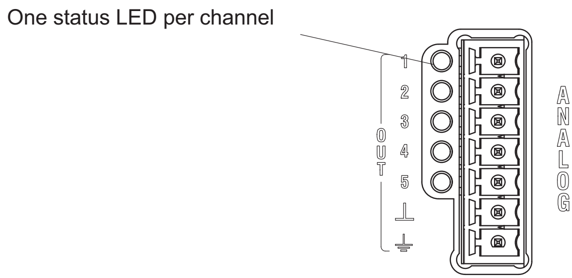
LED	Status	Meaning
Green	On	No errors
Yellow	On Flashing	No transducer connected or wire break (calibration in progress) Firmware updates
Red	On	Parameter not OK, transducer error, overloaded

Measuring range monitoring

By default, all inputs are checked for out-of-range (before a possibly set filter). The allowed measuring ranges are dictated by the specified sensor type. If the range is exceeded, the measured value becomes invalid.

Sensor type	Allowed measuring range
Frequency	±2.05 MHz
Counter	±8388607
SSI	-1073741824 … +1073741823
PWM	0 … 100.0

PX878



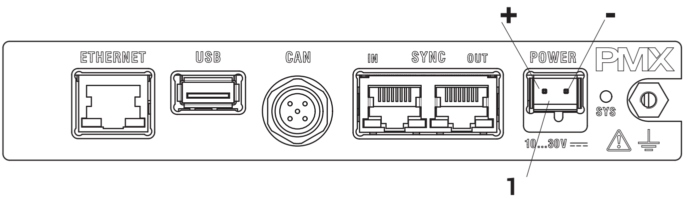
LED	Status	Meaning
Digital		
Green	On Off	Digital output: High Digital output: Low
Green	On Off	Digital input: High Digital input: Low
Analog		
Green	On Off	Analog output configured Analog output is not configured
Red	On	Analog output overloaded, signal invalid

8.3 Supply voltage

Notice

Device damage caused by too high voltages.
If you are using power supply 1-NTX001 listed in the accessories, note its included safety instructions.

With a separate DC voltage power supply (10 to 30 V_{DC}, nom. 24 V, power output at least 20 W), the PMX device is supplied with voltage via the POWER socket (**1**) (see *chapter 11, "Start-up", page 135*).



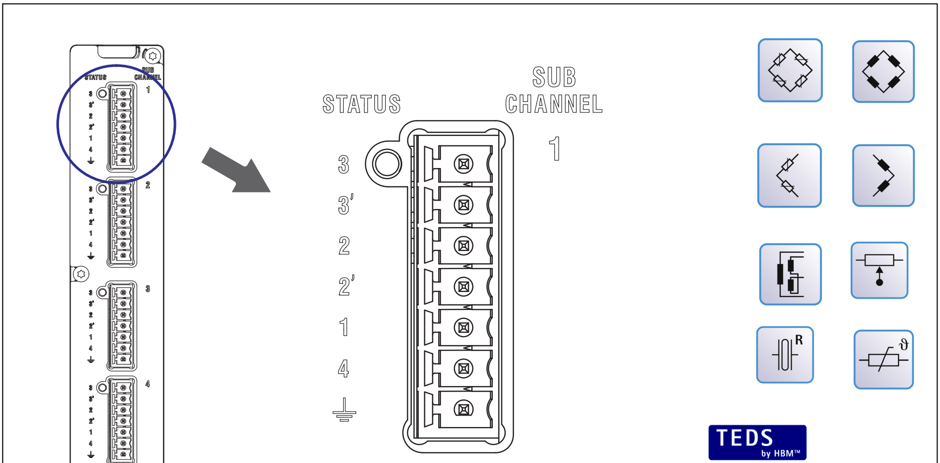
Measurement card	Power consumption [W] with 24 V supply voltage
Basic device	3
PX401	0.75
PX455	1.6
PX460	2
PX878	2
PX01EC (EtherCAT®)	2
PX01PN (PROFINET® IO)	2.4
PX01EP (EtherNet/IP™)	2.3

8.4 Measurement cards/transducer connection

See also *section 8.1, page 46*, for information on the (optional) coding of the plug terminals, and *chapter 6, page 32* , on the shielding design.

8.4.1 PX455

Four individually configurable strain gage full or half bridges (4.8 kHz CF). Inductive full or half bridges, LVDT, potentiometric sensors, piezoresistive sensors, 4 TEDS (zero-wire), sensor detection



Zero-wire TEDS
see *section 8.7.3*

The bridge excitation voltage is 2.5 V. By default, all inputs are checked for out-of-range (before a possibly set filter). The allowed measuring ranges are dictated by the specified sensor type. If the range is exceeded, the measured value becomes invalid. Click on the icon to turn measuring range monitoring off and on. When monitoring is off, the measured value is displayed, and remains valid but is limited by the maximum possible span.

Sensor type	Allowed measuring range
Full bridge 1000 mV/V	±1100 mV/V
Half bridge 1000 mV/V	±550 mV/V
Full and half bridge 100 mV/V	±110 mV/V
Full and half bridge 4 mV/V	±4.5 mV/V

Sensor type	Allowed measuring range
Potentiometer	±550 mV/V
LVDT	±550 mV/V

8.4.2 Strain gage and inductive full bridges (6-wire circuit)

Important

For connection cable lengths > 50 m, you must connect the sense leads to the PMX via one resistor each. This must have half the value of the bridge resistance ($R_B/2$), and be mounted at the sensor (e.g. at the end of a cable permanently connected to the sensor in the plug).

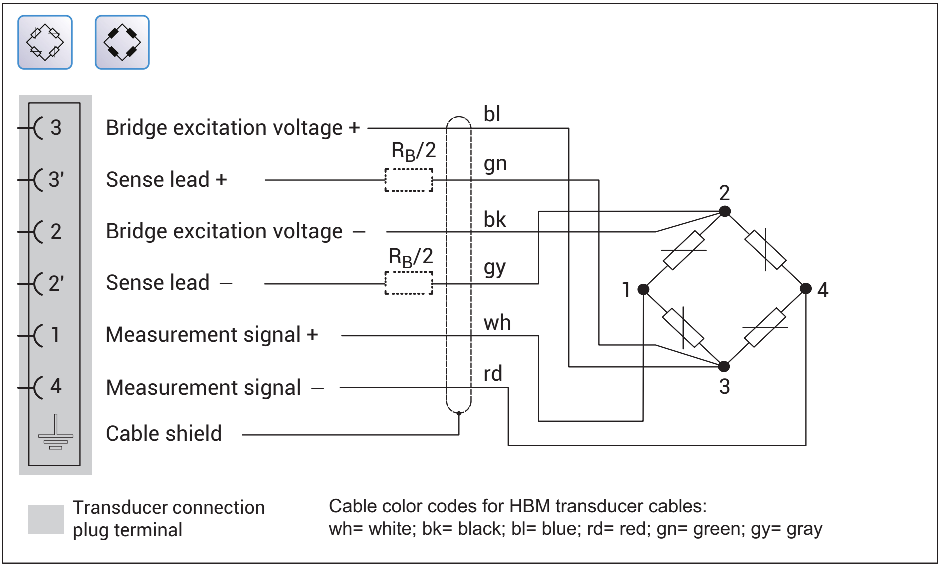


Fig. 8.3 PX455 pin assignment in 6-wire circuit

8.4.3 Strain gage and inductive half bridges (6-wire circuit)

Important

For connection cable lengths > 50 m, you must connect the sense leads to the PMX via one resistor each. This must have half the value of the bridge resistance ($R_B/2$), and be mounted at the sensor (e.g. at the end of a cable permanently connected to the sensor in the plug).

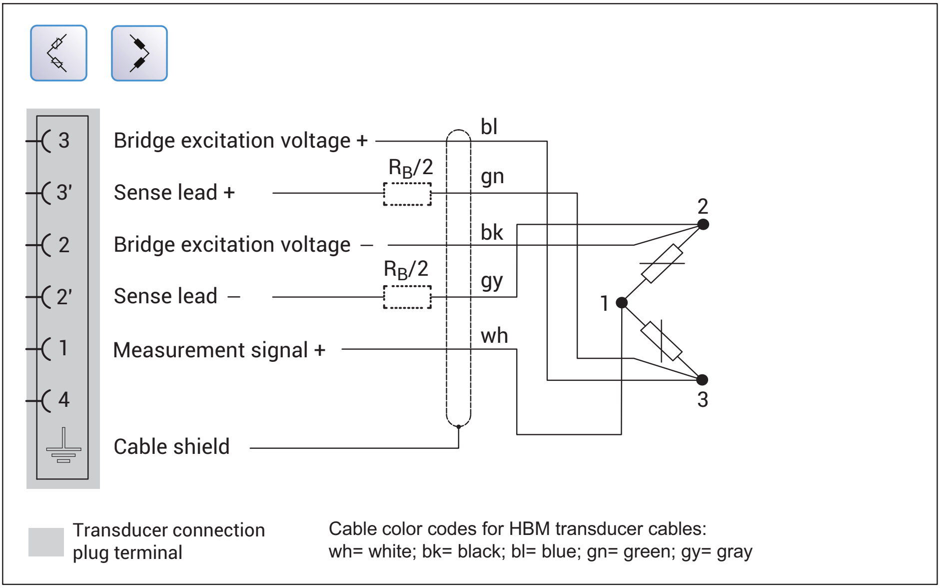
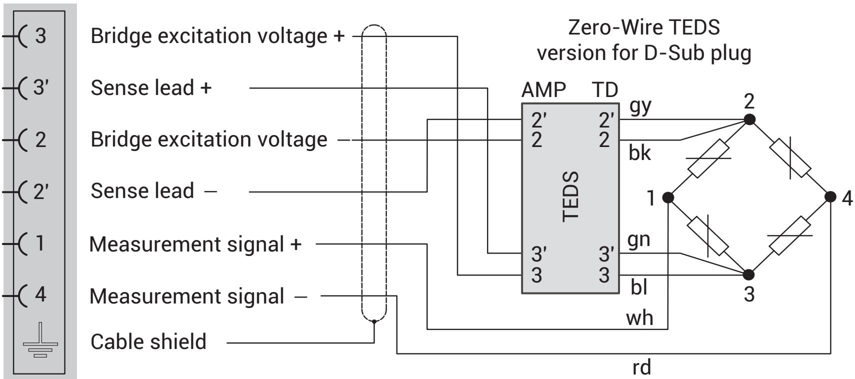


Fig. 8.4 PX455 pin assignment in 6-wire circuit

8.4.4 Strain gage and inductive full bridges in 6-wire circuit with zero-wire TEDS

Important

For connection cable lengths > 50 m, you must connect the sense leads to the PMX via one resistor each. The resistor must have half the value of the bridge resistance minus 100 Ω ($R_B/2$ - 100). At resistances greater than 300 Ω in a sensor cable, the TEDS module can no longer be read. The resistor must be mounted in the extension connector plug near the sensor, not between the sensor and the TEDS, and not on the PX455.
See also section 8.7.2, "Starting up the TEDS module", page 125.

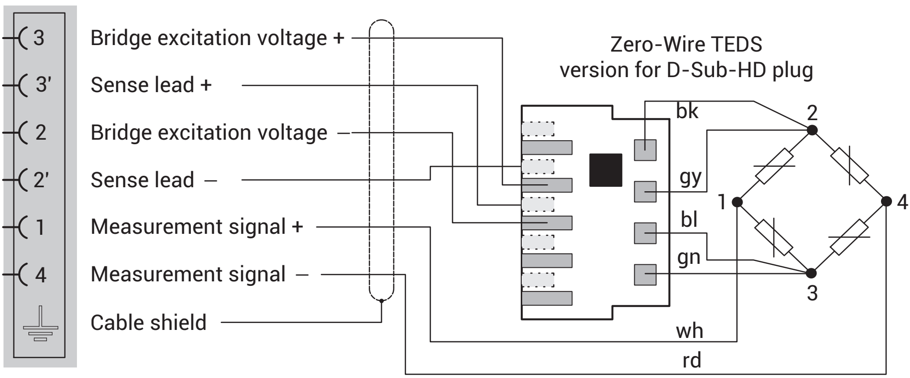


The solder terminals on the TEDS module are marked AMP (Amplifier) and TD (Transducer) together with the numbers. On the TD side, you have to connect contact 2 with 2' and 3 with 3' (short circuit or feedback bridges), otherwise the measured value is always invalid.

Transducer connection plug terminal

Cable color codes for HBM transducer cables:
wh= white; bk= black; bl= blue; rd= red; gn= green; gy= gray

Fig. 8.5 PX455 pin assignment in 6-wire circuit with zero-wire TEDS (D-Sub plug)



The solder terminals on this TEDS module are not labeled. You must connect the sensor leads on the side with the four contact surfaces (in the picture on the top of the module). The contacts represented with dashed lines are on the bottom of the TEDS module. Connect the + and - sense leads there.

Transducer connection plug terminal

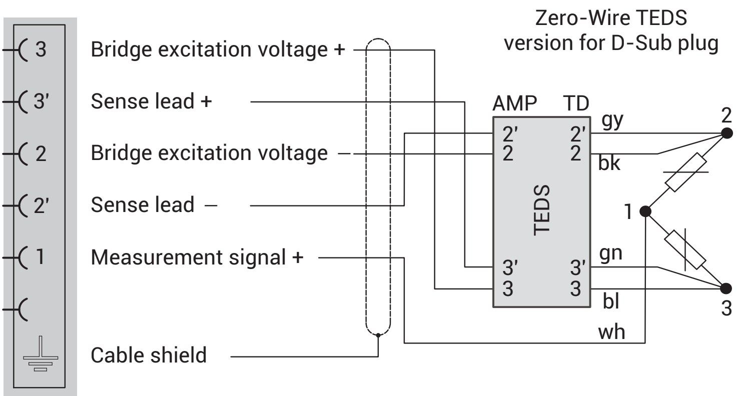
Cable color codes for HBM transducer cables:
wh= white; bk= black; bl= blue; rd= red; gn= green; gy= gray

Fig. 8.6 PX455 pin assignment in 6-wire circuit with zero-wire TEDS (D-Sub-HD plug)

8.4.5 Strain gage and inductive half bridges in 6-wire circuit with zero-wire TEDS

Important

For connection cable lengths > 50 m, you must connect the sense leads to the PMX via one resistor each. The resistor must have half the value of the bridge resistance minus 100 Ω (R_B/2 - 100). At resistances greater than 300 Ω in a sensor cable, the TEDS module can no longer be read. The resistor must be mounted in the extension connector plug near the sensor, not between the sensor and the TEDS, and not on the PX455.
See also section 8.7.2 "Starting up the TEDS module", page 125.



The solder terminals on the TEDS module are marked AMP (Amplifier) and TD (Transducer) together with the numbers.

Transducer connection plug terminal

Cable color codes for HBM transducer cables:
wh= white; bk= black; bl= blue; gn= green; gy= gray

Fig. 8.7 PX455 pin assignment in 6-wire circuit with zero-wire TEDS (D-Sub plug)

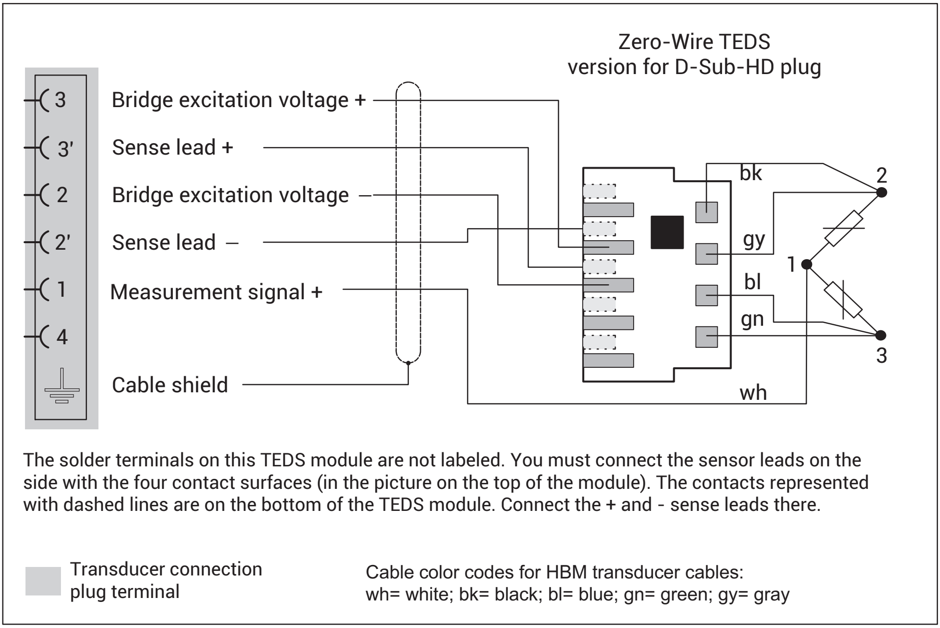


Fig. 8.8 PX455 pin assignment in 6-wire circuit with zero-wire TEDS (D-Sub-HD plug)

8.4.6 Strain gage and inductive full bridges (4-wire circuit)

Important

For connection cable lengths > 15 m, you must solder a resistor in place of the feedback bridges. This must have half the value of the bridge resistance (R_B/2), and must be mounted at the transition from 4-wire to 6-wire circuit (e.g. in the connector plug of the 6-wire cable). You must execute the extension in a 6-wire circuit; a 4-wire circuit is not permissible.

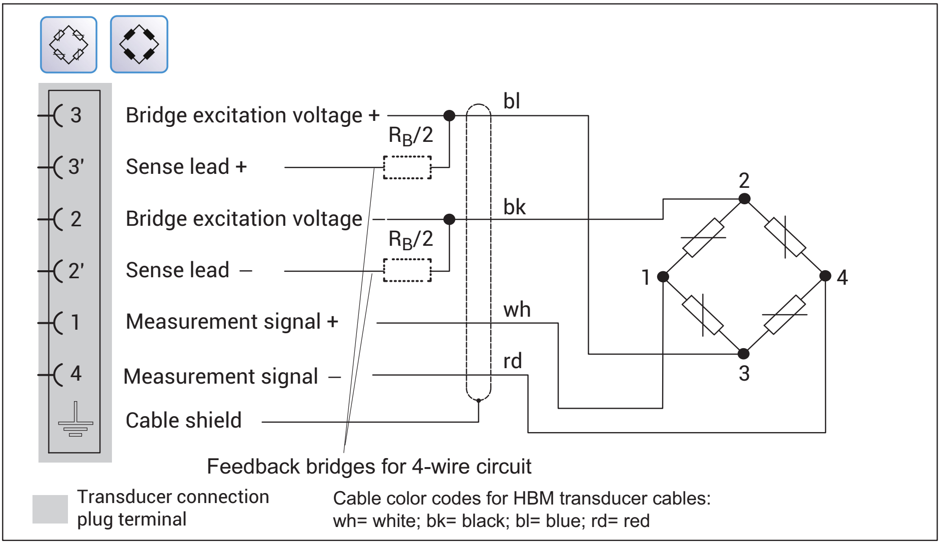


Fig. 8.9 PX455 pin assignment in 4-wire circuit

8.4.7 Strain gage and inductive half bridges (4-wire circuit)

Important

For connection cable lengths > 15 m, you must solder a resistor into each of the feed-back bridges. This must have half the value of the bridge resistance ($R_B/2$), and must be mounted at the transition from 4-wire to 6-wire circuit (e.g. in the connector plug of the 6-wire cable). You must execute the extension in a 6-wire circuit; a 4-wire circuit is not permissible.

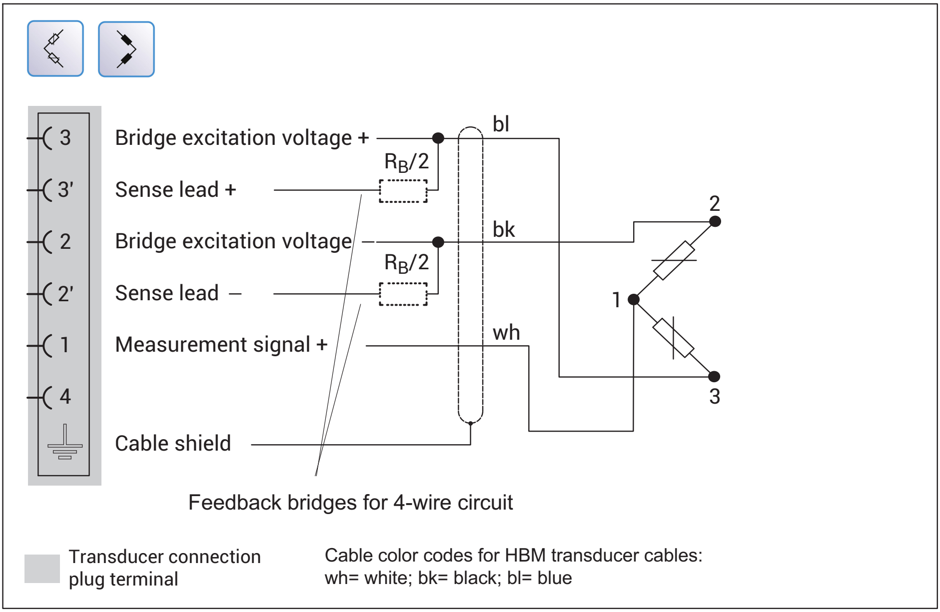
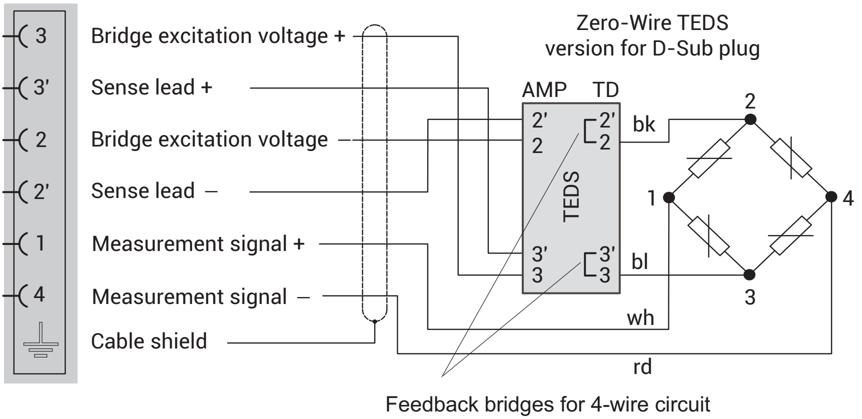


Fig. 8.10 PX455 pin assignment in 4-wire circuit

8.4.8 Strain gage and inductive full bridges (4-wire circuit) with zero-wire TEDS

Important

For connection cable lengths > 15 m, you must solder one resistor each into the sense leads on the PMX. The resistor must have half the value of the bridge resistance minus 100 Ω (R_B/2 - 100). At resistances greater than 300 Ω in a sensor cable, the TEDS module can no longer be read. The resistor must be mounted in the extension connector plug near the sensor, not between the sensor and the TEDS, and not on the PX455.
See also section 8.7.2 "Starting up the TEDS module", page 125.



The solder terminals on the TEDS module are marked AMP (Amplifier) and TD (Transducer) together with the numbers.

Transducer connection plug terminal

Cable color codes for HBM transducer cables:
wh= white; rd= red; bk= black; bl= blue

Fig. 8.11 PX455 pin assignment in 4-wire circuit with zero-wire TEDS (D-Sub plug)

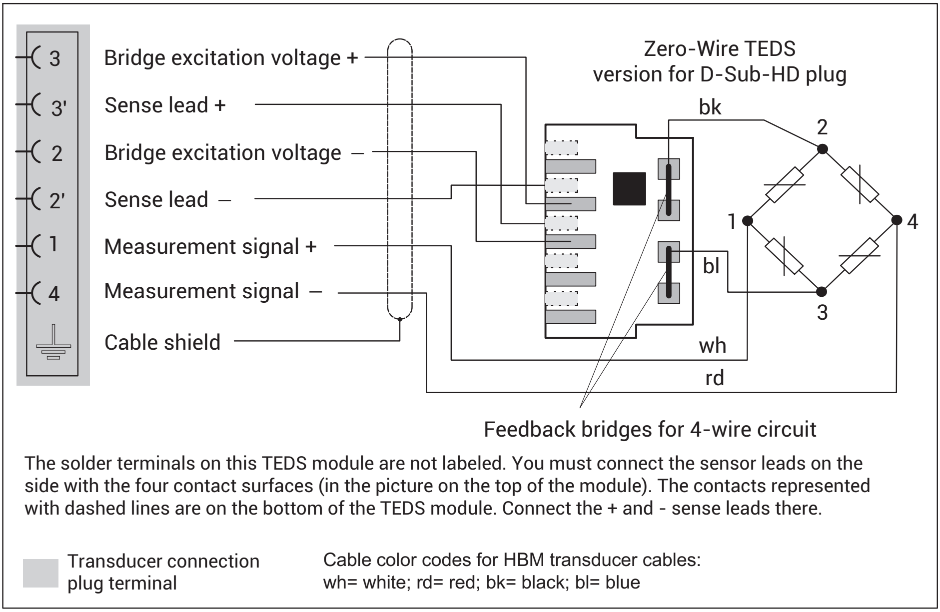


Fig. 8.12 PX455 pin assignment in 4-wire circuit with zero-wire TEDS (D-Sub-HD plug)

8.4.9 Strain gage and inductive half bridges (4-wire circuit) with zero-wire TEDS

Important

For connection cable lengths > 15 m, you must solder one resistor each into the sense leads on the PMX. The resistor must have half the value of the bridge resistance minus 100 Ω (R_B/2 - 100). At resistances greater than 300 Ω in a sensor cable, the TEDS module can no longer be read. The resistor must be mounted in the extension connector plug near the sensor, not between the sensor and the TEDS, and not on the PX455.
See also section 8.7.2 "Starting up the TEDS module", page 125.

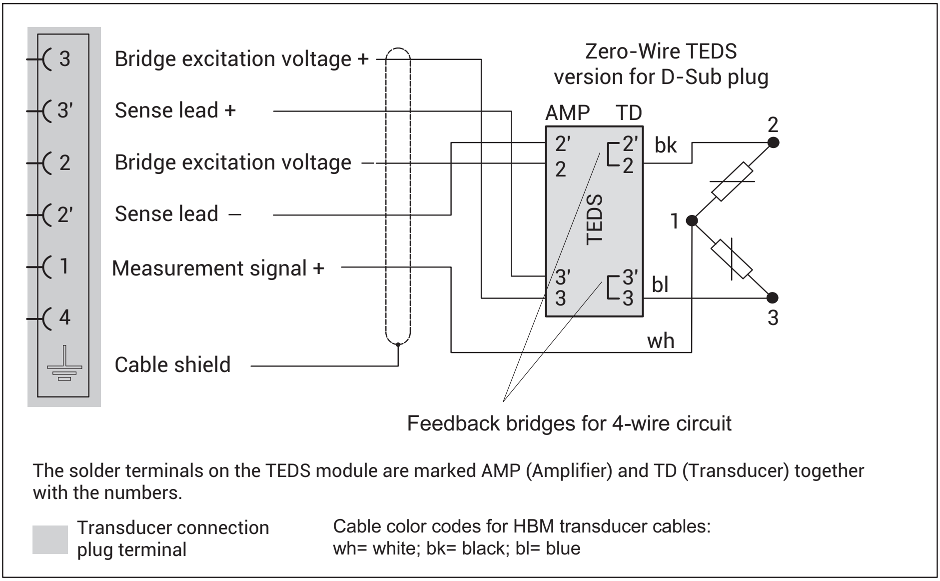


Fig. 8.13 PX455 pin assignment in 4-wire circuit with zero-wire TEDS (D-Sub plug)

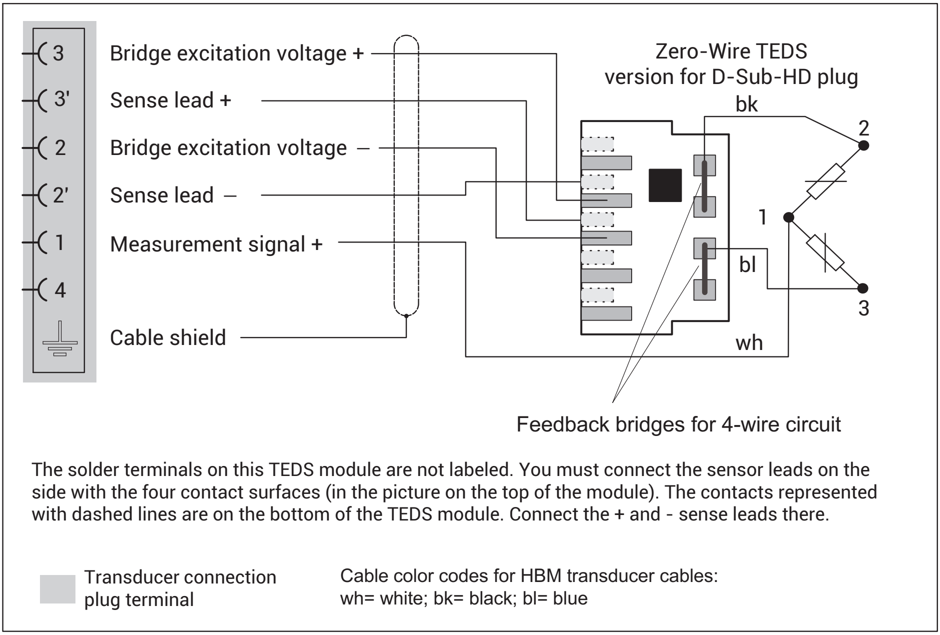


Fig. 8.14 PX455 pin assignment in 4-wire circuit with zero-wire TEDS (D-Sub-HD plug)

8.4.10 Intrinsically safe measuring circuits - operation with Zener barriers

To operate transducers (load cells, force transducers etc) in potentially explosive hazardous areas, intrinsically safe measurement circuits (Ex II (1) GD, [Ex ia]IIC) must be set up on the PX455 by connecting safety barriers (Zener barriers) of type SD01A. The safety barriers must also be mounted on the support rail like the PMX. An ATEX test certificate must be obtained for the transducers used. You can use transducers with a bridge resistance of 350 ohms. Only one transducer per PX455 measurement channel may be operated. Parallel connection is not possible. A TEDS module can also not be used.

Use the cable KAB7.5/00-2/2/2 from HBM, LF-ZYAECVY 3x2x0.14 mm^2, ordering number 1-CABE2/20 or 1-CABE2/100, or cables from Lappgroup (http://www.lapp-group.com): Li2YCYPMF 3x2x0.5 mm^2.

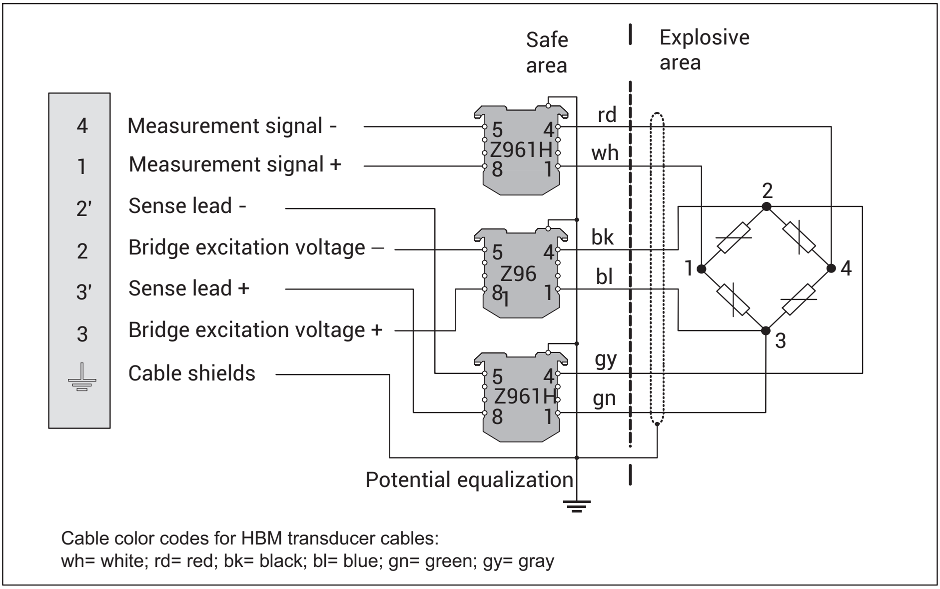


Fig. 8.15 PMX455 pin assignment with SD01A safety barriers

The PX455 offers four measurement channels with 4.8 kHz carrier frequency. Use the internal calculation channels of the PMX to add or subtract measurement signals or to calculate the arithmetic mean for example.

Important

In addition to the SD01A, the negative operating voltage of the PMX must also be grounded!
Cable lengths up to a maximum of 100 meters are allowed. A TEDS module cannot be used.
The accuracy class of the PX455 when operating with SD01A is 0.5%.

8.4.11 LVDT transducer

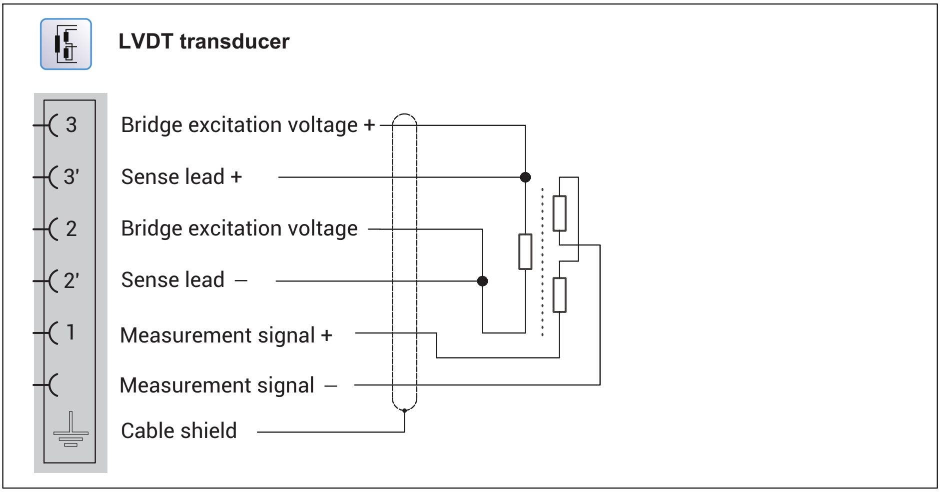


Fig. 8.16 Pin assignment for PX455 LVDT transducer

8.4.12 LVDT transducer with zero-wire TEDS

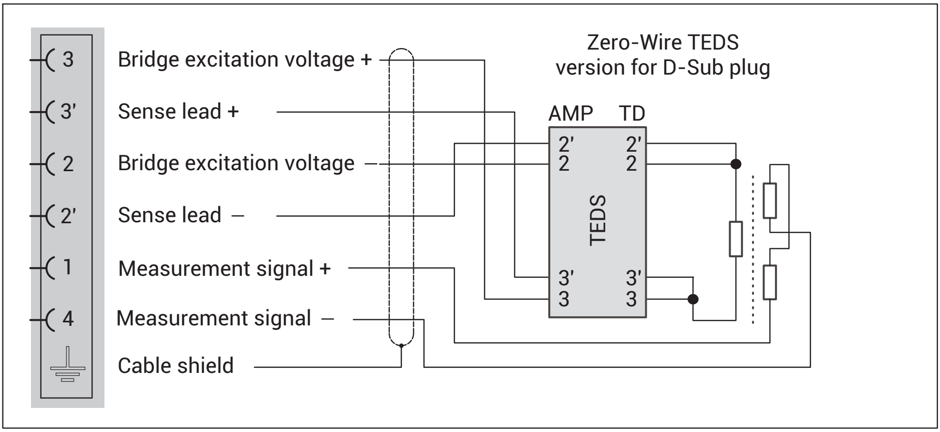


Fig. 8.17 PX455 pin assignment, LVDT transducers with zero-wire TEDS (D-Sub plug)

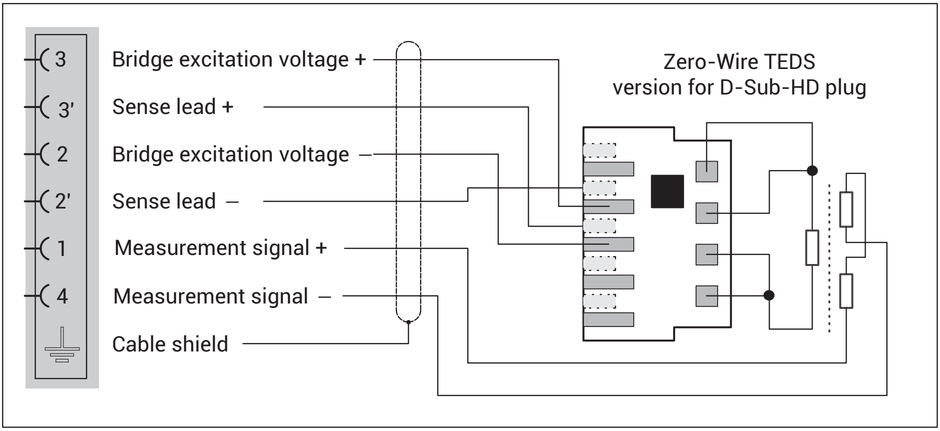


Fig. 8.18 PX455 pin assignment, LVDT transducers with zero-wire TEDS (D-Sub-HD plug)

8.4.13 Potentiometric transducer

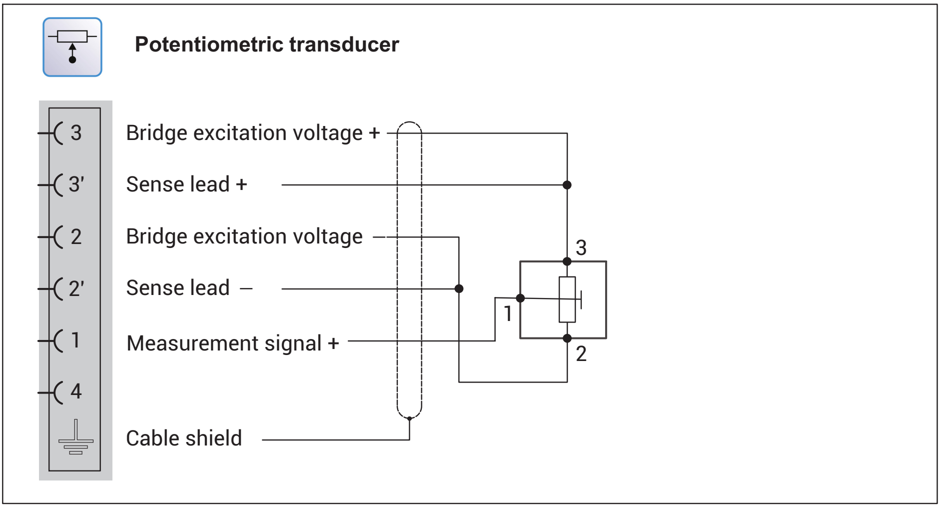


Fig. 8.19 PX455 pin assignment for potentiometric transducers

Important

Transducer connection in 4/3-wire circuit:

When connecting a transducer in a 4/3-wire circuit, you must connect the sense leads with the corresponding bridge excitation circuit (pin 2' to pin 2 and pin 3' to pin 3) by means of wire bridges, otherwise a sensor error will be signaled.

When connecting in a 4-wire circuit, the TEDS functionality is not available.

8.4.14 Potentiometric transducer with zero-wire TEDS

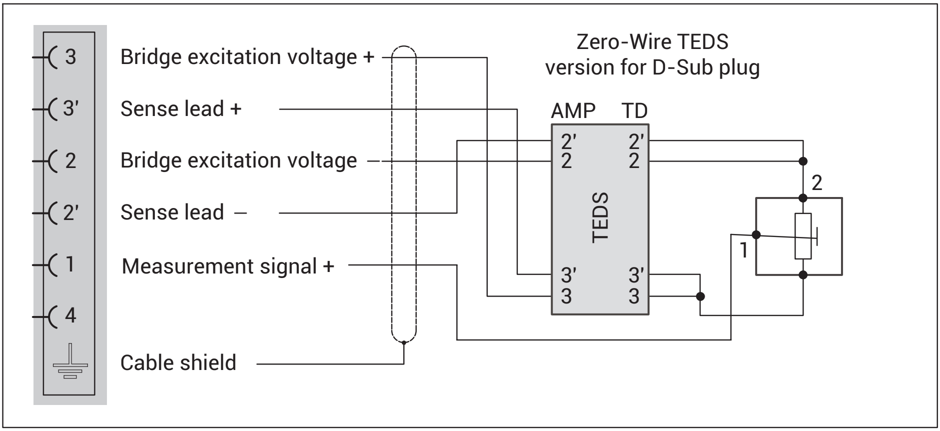


Fig. 8.20 PX455 pin assignment for potentiometric transducers with zero-wire TEDS (D-Sub plug)

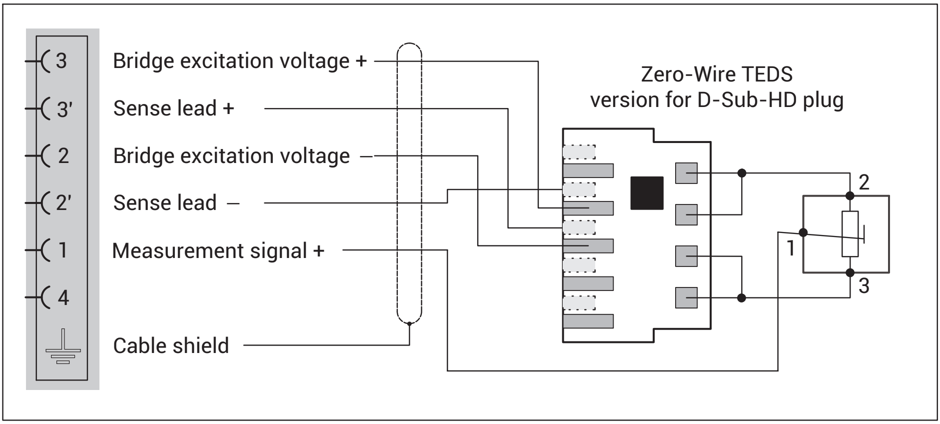


Fig. 8.21 PX455 pin assignment for potentiometric transducers with zero-wire TEDS (D-Sub-HD plug)

8.4.15 PX455 with Pt100 temperature measurement

With the PX455 measurement card, temperatures can be measured with an uncertainty of +/-1°C without an external preamplifier. To do this, you must add a 100 ohm precision resistor (R_compl) with a maximum tolerance of 0.1% to a Pt100 resistor to make a half bridge circuit, and connect it directly to the terminals of the PX455. The PMX calculation channel "Pt100 on PX455" then converts the measured bridge unbalance to degrees Celsius and performs a corrective calculation according to the sensor cable being used (R_wire).

Important

Ensure good heat dissipation so as to keep the measurement error caused by the self heating of the Pt100 as low as possible! This can be done, for example, by mounting it on a metallic body.

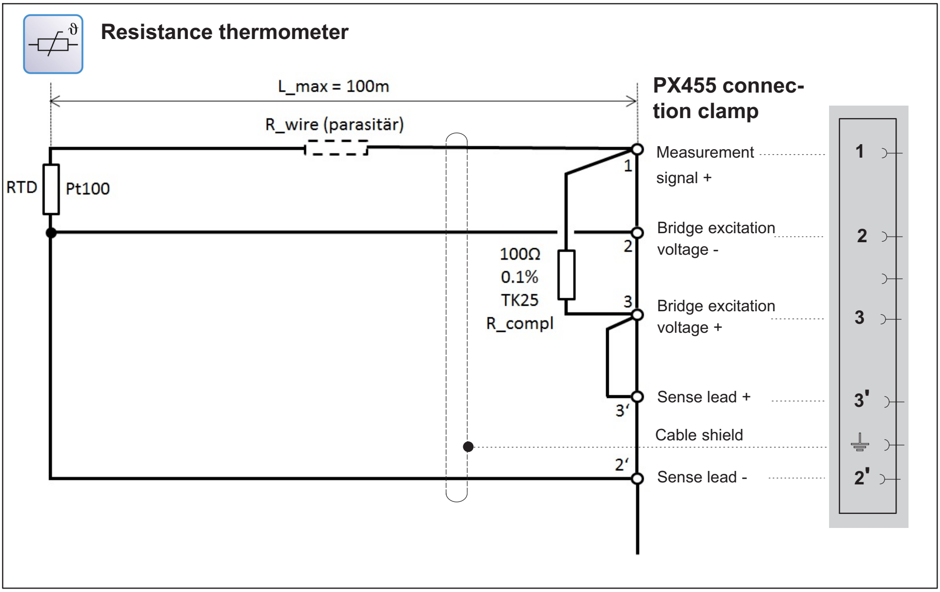
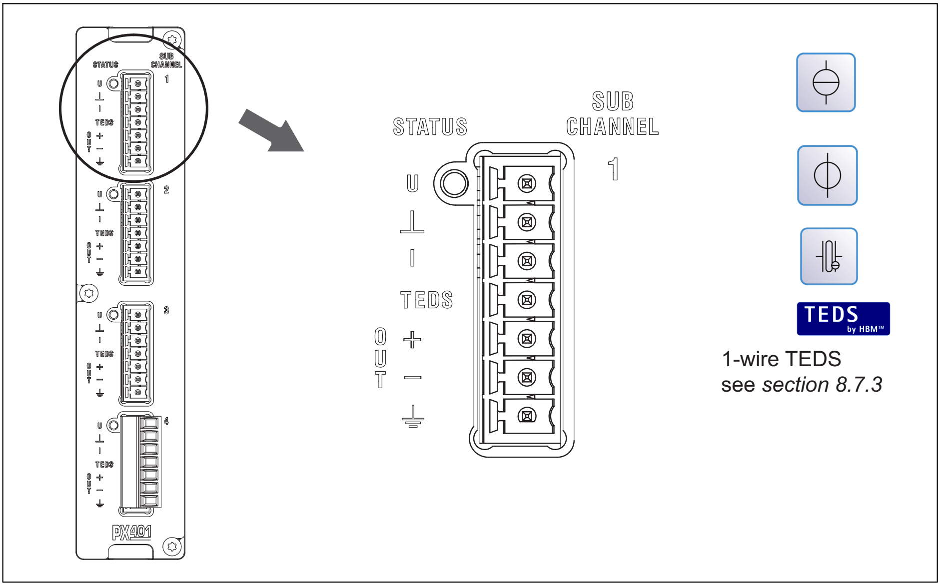


Fig. 8.22 PX455 with Pt100 element for temperature measurement

8.4.16 PX401

Four individually configurable current or voltage inputs with 4 TEDS (1-wire) sensor detection.



You can also use IEPE sensors in conjunction with the 1-EICP-B-2 smart module.

By default, all inputs are checked for out-of-range (before a possibly set filter). The allowed measuring ranges are dictated by the specified sensor type. If the range is exceeded, the measured value becomes invalid. Click on the icon to turn measuring range monitoring off and on. When monitoring is off, the measured value is displayed, and remains valid but is limited by the maximum possible span.

Sensor type	Allowed measuring range
±10 V	±11.0 V
±20 mA	±21.0 mA
4 … 20 mA	3.9 … 21.0 mA

8.4.17 Voltage source ± 10 V

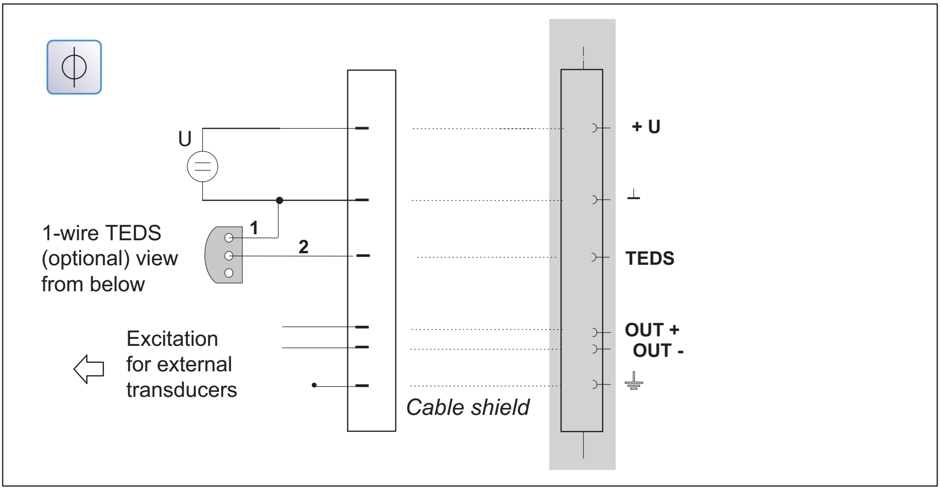


Fig. 8.23 Pin assignment PX401: Voltage source ±10 V

8.4.18 Current source ± 20 mA

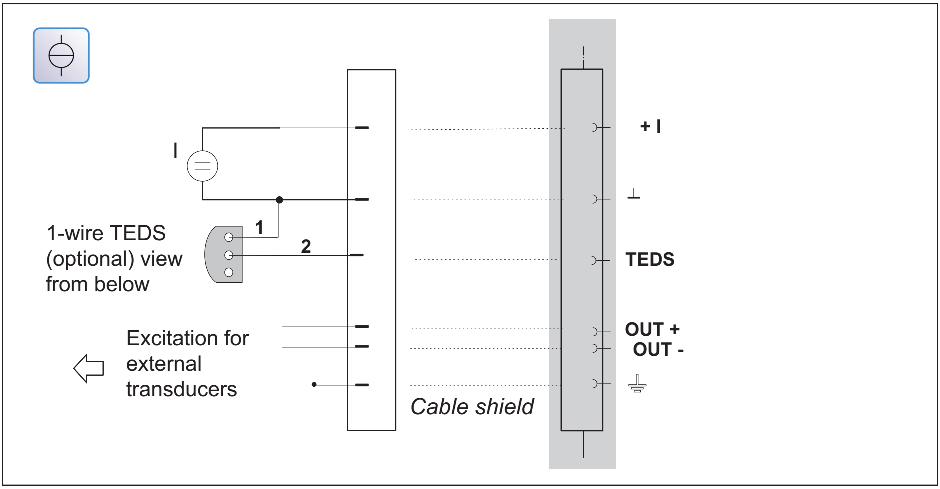


Fig. 8.24 Pin assignment PX401: Current source ±20 mA (4-wire circuit)

8.4.19 Current drain ± 20 mA

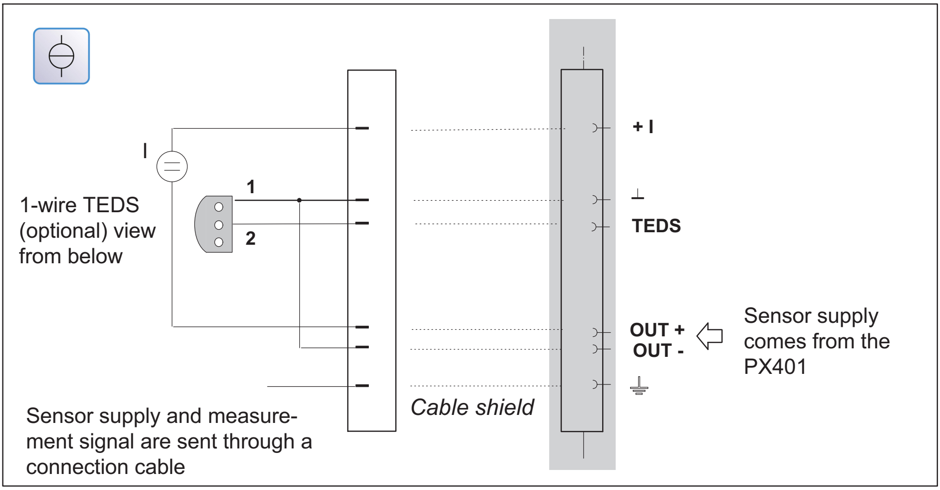


Fig. 8.25 Pin assignment PX401: Current drain ±20 mA (2-wire circuit)

Current-fed IEPE or IPC piezoelectric transducers are supplied with constant current, e.g. 4 mA, and provide a voltage signal that you can operate with the PX401 via an external module.

8.4.20 IEPE transducer with external amplifier

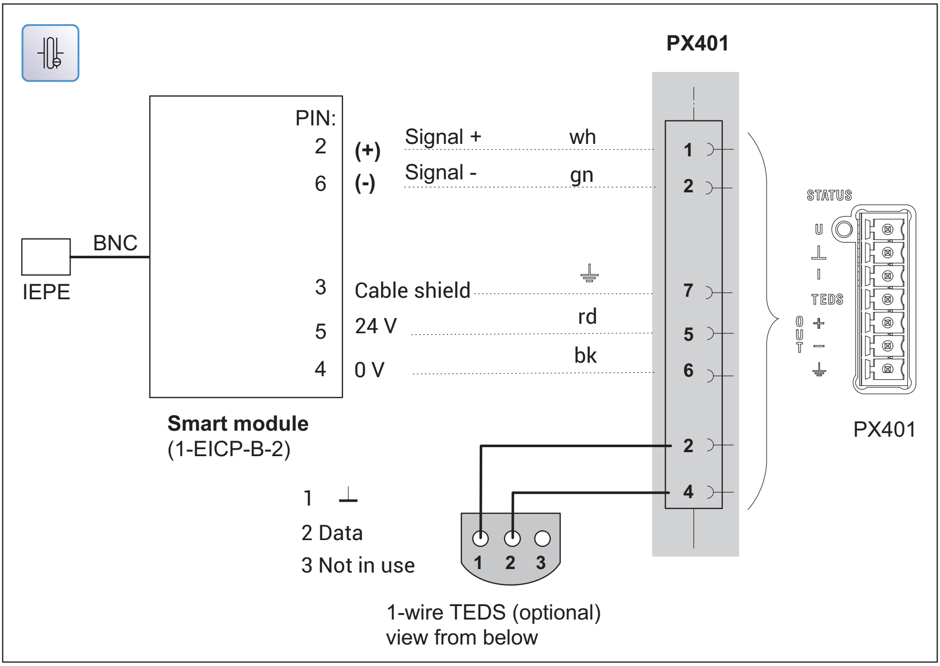


Fig. 8.26 PX401 pin assignment, IEPE transducers

8.4.21 PX401 with charge amplifier

Piezoelectric sensors can be operated with the PX401 via external CMA or CMD charge amplifiers. The charge amplifiers convert the sensor signal into a ±10 V voltage signal. The Reset/Operate charge amplifier signal can be implemented in the PMX from an external controller or via a PX878 digital output.

Important

*Due to the inrush current of the **CMD** charge amplifier, the bridge excitation of the CMD must be separate and not via the PX401 measurement card.*

Piezoelectric sensor with external charge amplifier

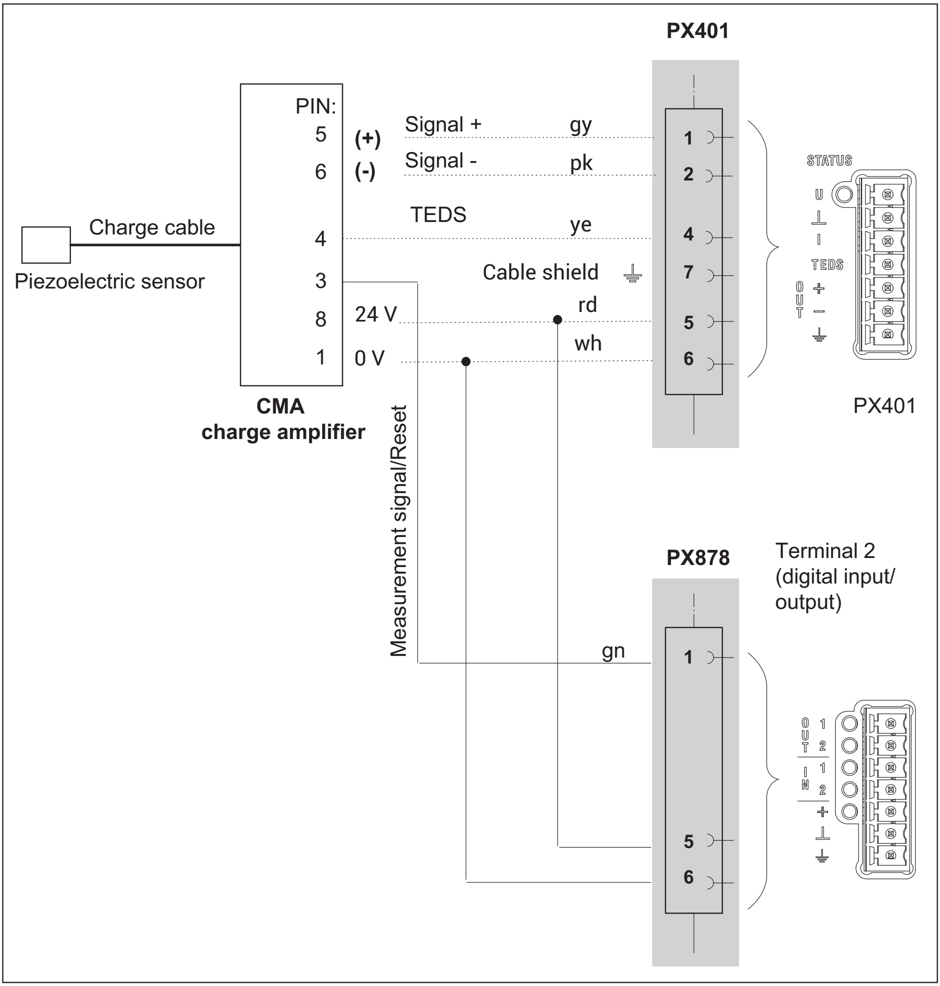


Fig. 8.27 PX401 pin assignment with external charge amplifier

External transducers are supplied with power via the PX401 measurement card (OUT + and OUT -). The supply voltage corresponds to the device supply voltage.

The maximum current is 400 mA per measurement card, and is divided among the transducers being used.

8.4.22 PX401 electrical isolation

Important

The individual measurement channels on the PX401 measurement card are not electrically isolated from each other. The PX401 measurement card has a common electrical isolation from the basic device.

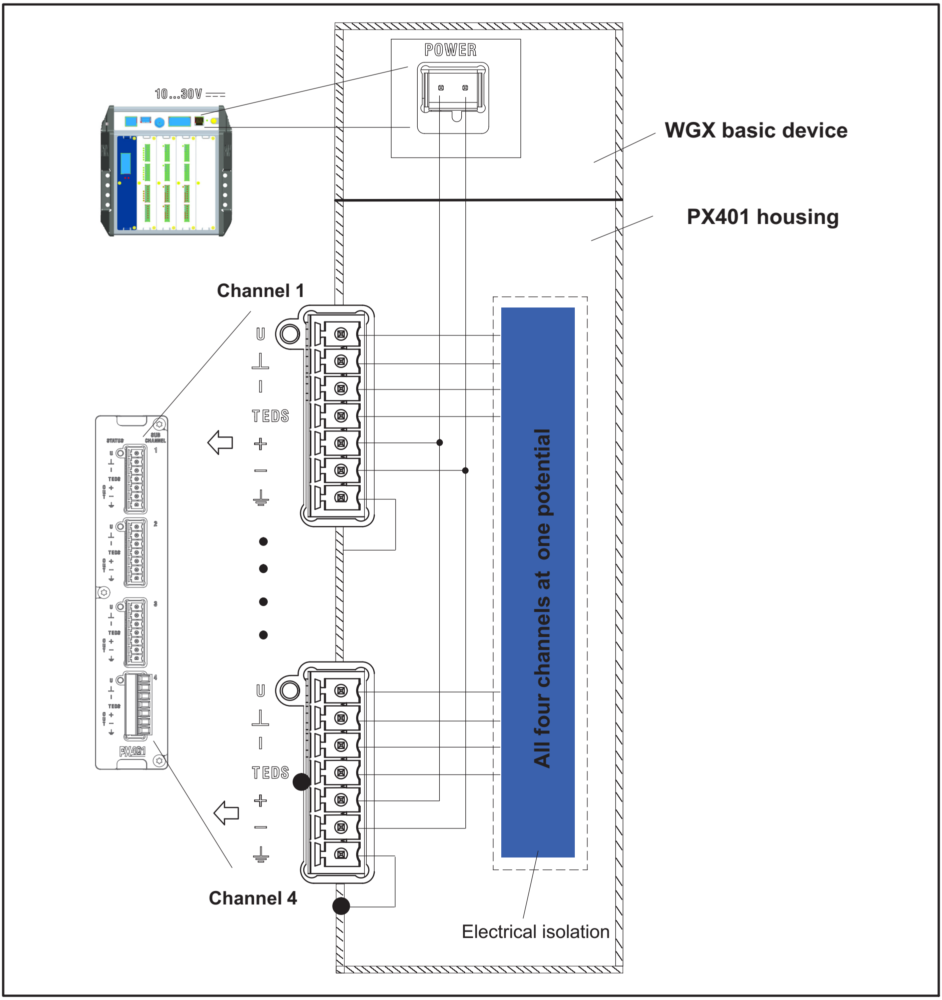


Fig. 8.28 PX401 electrical isolation

8.4.23 PX460

Torque transducers (torque, rotational speed, angle of rotation), angle/incremental encoders, SSI, PWM sensors, frequency measurement up to 2 MHz

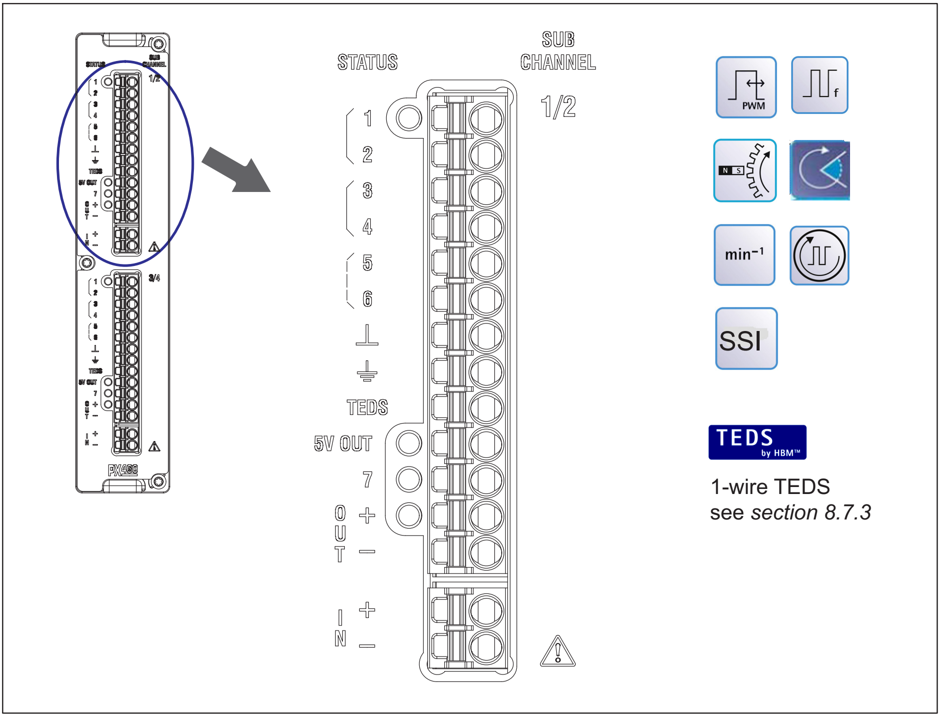
Channels 1 and 3: Frequency measurement (fixed)
Channels 2 and 4: Frequency (digital/inductive), counter, encoder, SSI, PWM (adjustable)

The following measurement modes are available:

- up to four torque transducers (T10, T12, T40) for torque or rotational speed measurement (without detection of direction of rotation);
- or two torque transducers for simultaneous measurement of torque transducer and rotational speed (without detection of angle of rotation/direction of rotation);
- or one torque flange for simultaneous measurement of torque, rotational speed, angle of rotation, and direction of rotation and reference signal detection;
- or two angle/incremental encoders, SSI, PWM sensors, magnetic transducers or pulse counters each;
- or four torque transducers for frequency measurement up to 2 MHz including two shunt calibrations and two 1-wire TEDS (sensor detection).

By default, all inputs are checked for out-of-range (before a possibly set filter). The allowed measuring ranges are dictated by the specified sensor type. If the range is exceeded, the measured value becomes invalid. Click on the icon to turn measuring range monitoring off and on. When monitoring is off, the measured value is displayed, and remains valid but is limited by the maximum possible span. Exception: In the event of a number overflow (counter, SSI), NaN (not a number) appears in the display and $\pm 3.4 * 10^{38}$ (invalid) is outputted.

Sensor type	Allowed measuring range
Frequency	±2.05 MHz
Counter	±8.388607
SSI	-1,073,741,824 … +1,073,741,823
PWM	0 … 100.0



Notice

The sensors for the PX460 are supplied with voltage externally via the contacts (IN + -). The PX460 card then provides the supply for 24 V (OUT + -) and 5 V (5 V OUT).
*The input signals fed into the PX460 from the sensor may be **max.** ±**15 V**, otherwise the measurement inputs of the PX460 may be destroyed.*
A shunt can be connected via pin 7. It can be activated via the PMX web browser, a PMX command, the .NET API, or catman®.

8.4.24 Voltage supply for signal transmitters and transducers up to 24 V_{DC} nominal

Notice

The input signals fed from the sensor into the PX460 must not exceed ***max. ±15 V****, otherwise the measurement inputs of the PX460 may be destroyed.*

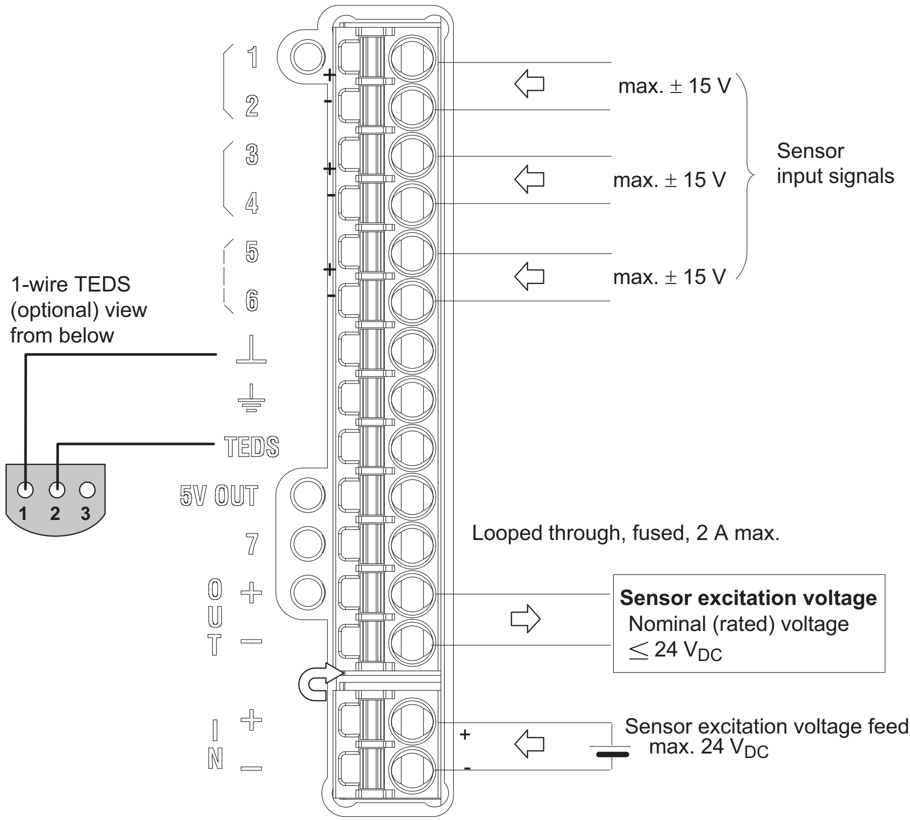


Fig. 8.29 Voltage supply, PX460 options up to 24 V_{DC} nominal

8.4.25 Voltage supply for signal transmitters and transducers up to 5 V_{DC} nominal

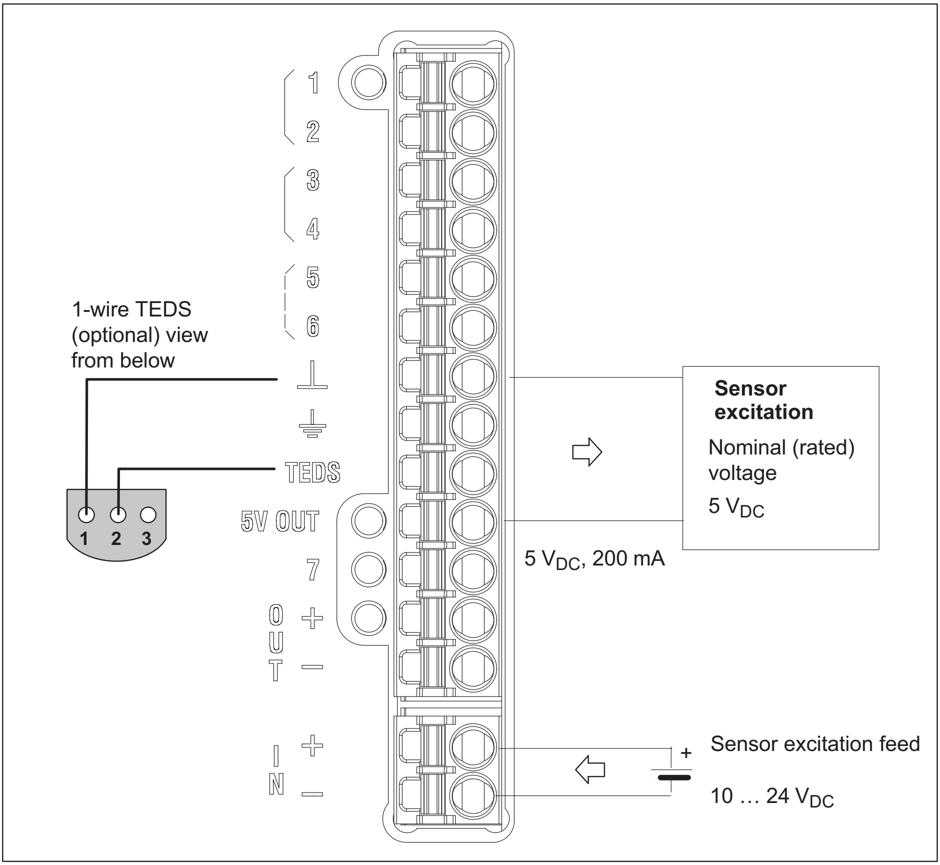


Fig. 8.30 Voltage supply, PX460 options up to 5 V_{DC} nominal

8.4.26 Frequency measurement (differential input)

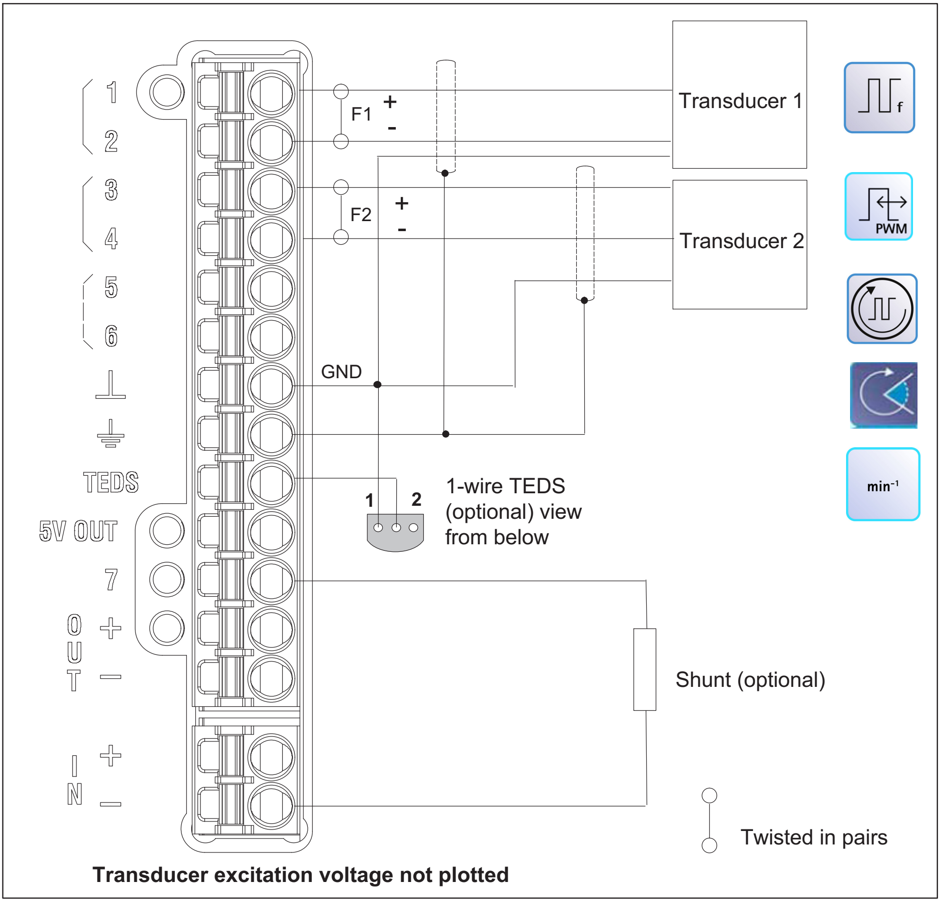


Fig. 8.31 PX460 pin assignment for two frequencies, differential

Possible channel setting via web server:

Transducer 1: Frequency (digital), fixed
Transducer 2: Frequency (digital), counter, PWM

8.4.27 Frequency measurement (single-ended input)

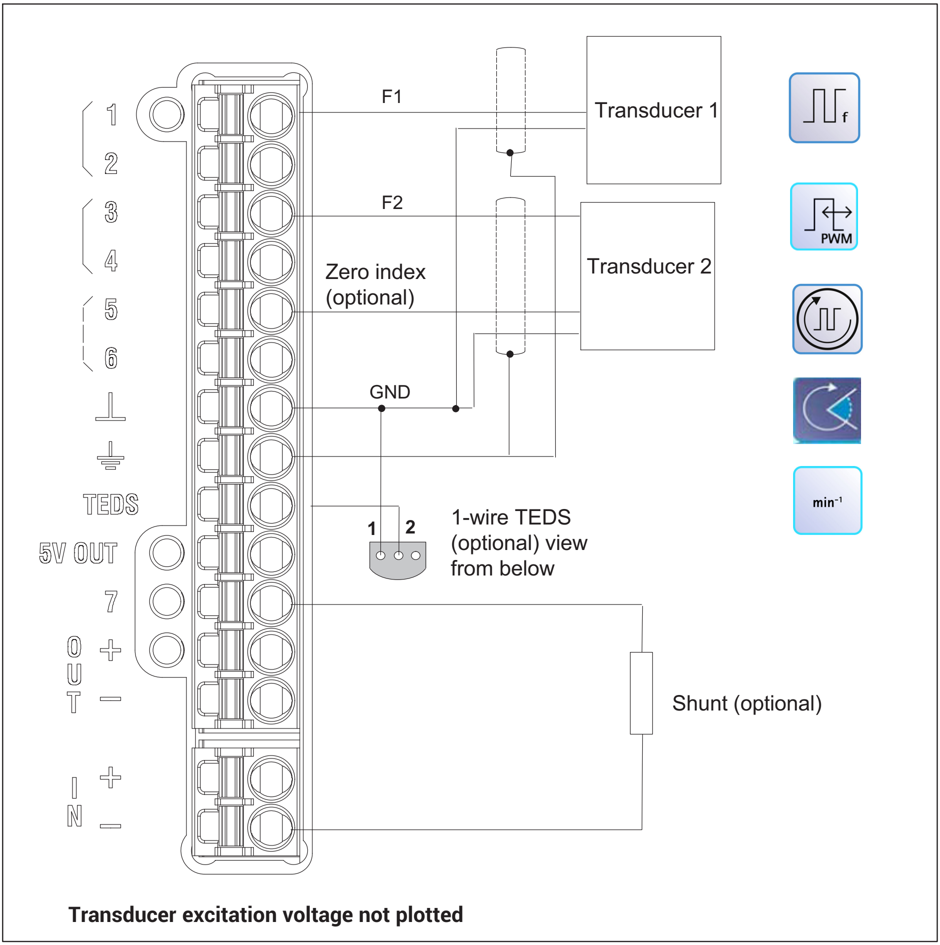


Fig. 8.32 PX460 pin assignment for two frequencies, unbalanced

Possible channel setting via web server:

Transducer 1: Frequency (digital), fixed
Transducer 2: Frequency (digital), fixed, counter, PWM

8.4.28 Rotary encoder and incremental encoder, balanced (differential)

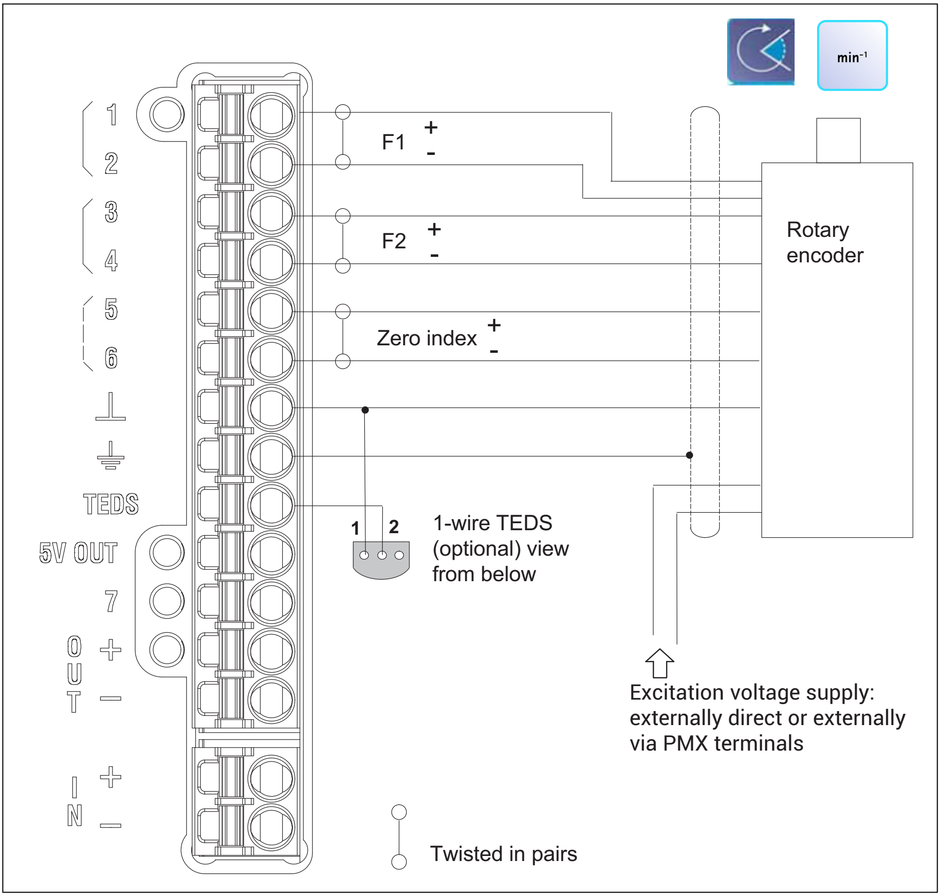


Fig. 8.33 PX460 pin assignment for rotary encoders, balanced

Possible channel setting via web server:

Transducer 1: Frequency (digital), fixed
Transducer 2: Frequency (digital), counter, PWM

8.4.29 Rotary encoder and incremental encoder with direction signal, balanced (differential)

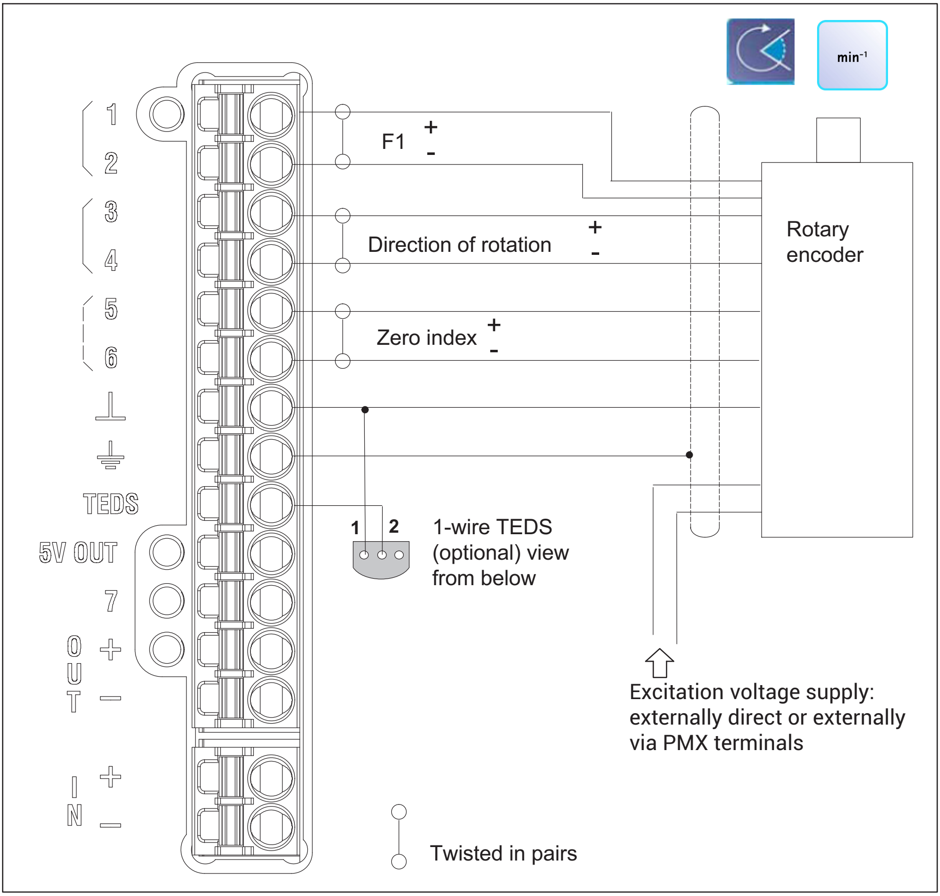


Fig. 8.34 PX460 pin assignment for rotary encoders, balanced

Possible channel setting via web server:

Transducer 1: Frequency (digital), fixed
Transducer 2: Type: "Direction bit", counter

8.4.30 Rotary encoder and incremental encoder, unbalanced (single-pole)

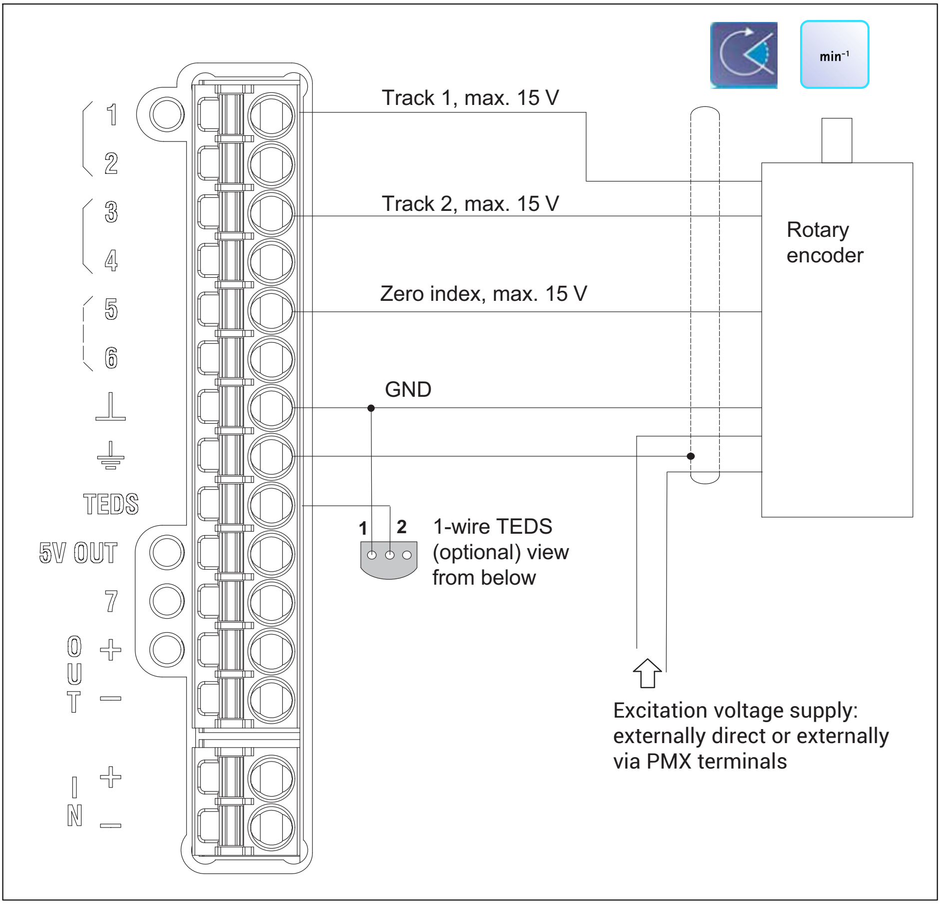


Fig. 8.35 PX460 pin assignment for rotary encoders, unbalanced

Possible channel setting via web server:

Transducer 1: Frequency (digital), fixed
Transducer 2: Frequency (digital), counter, PWM

8.4.31 Rotary encoder and incremental encoder with direction signal, unbalanced (single-pole)

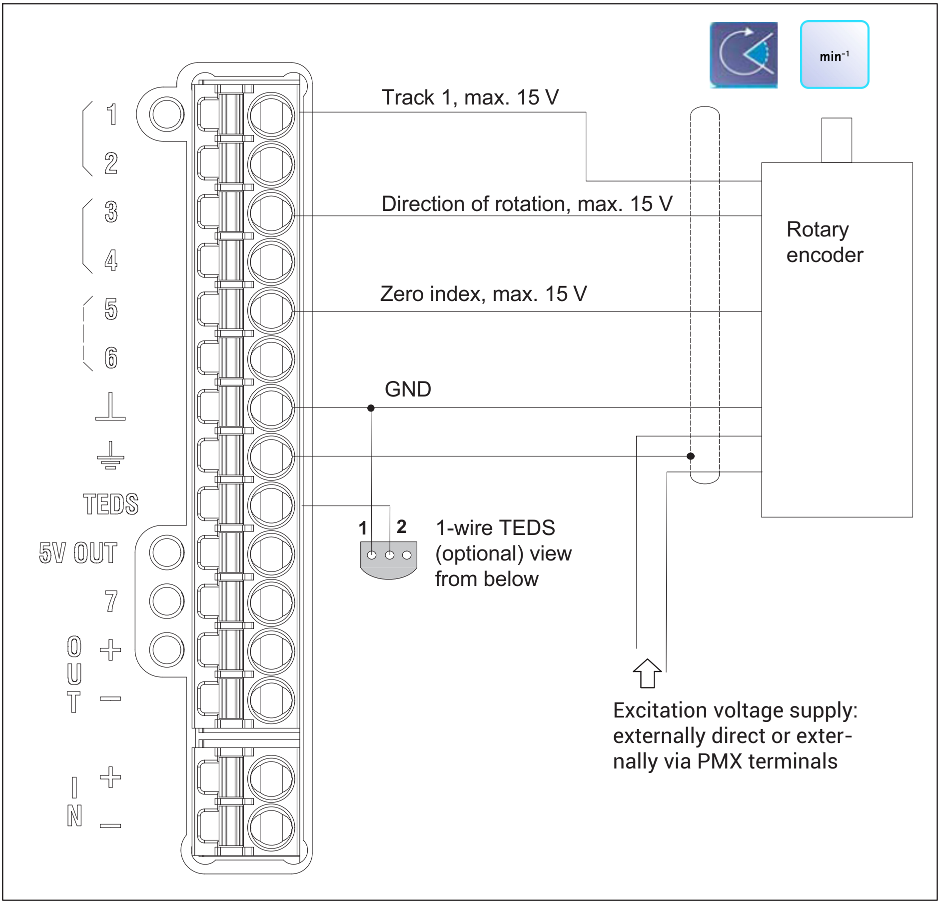


Fig. 8.36 PX460 pin assignment for rotary encoders, unbalanced

Possible channel setting via web server:

Transducer 1: Frequency (digital), fixed
Transducer 2: Type: "Direction bit", counter

8.4.32 SSI encoder (active only)

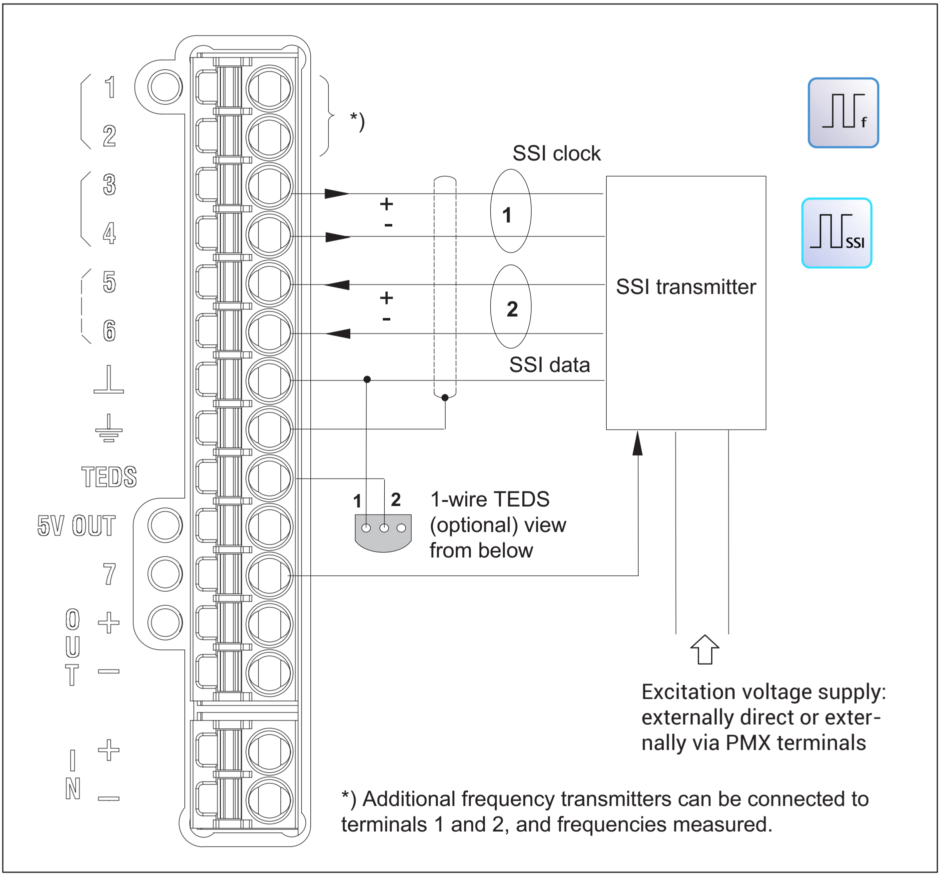


Fig. 8.37 PX460 pin assignment for SSI encoders

Possible channel setting via web server:

Transducer: SSI

8.4.33 Inductive rotary encoders or pulse generators (passive only)

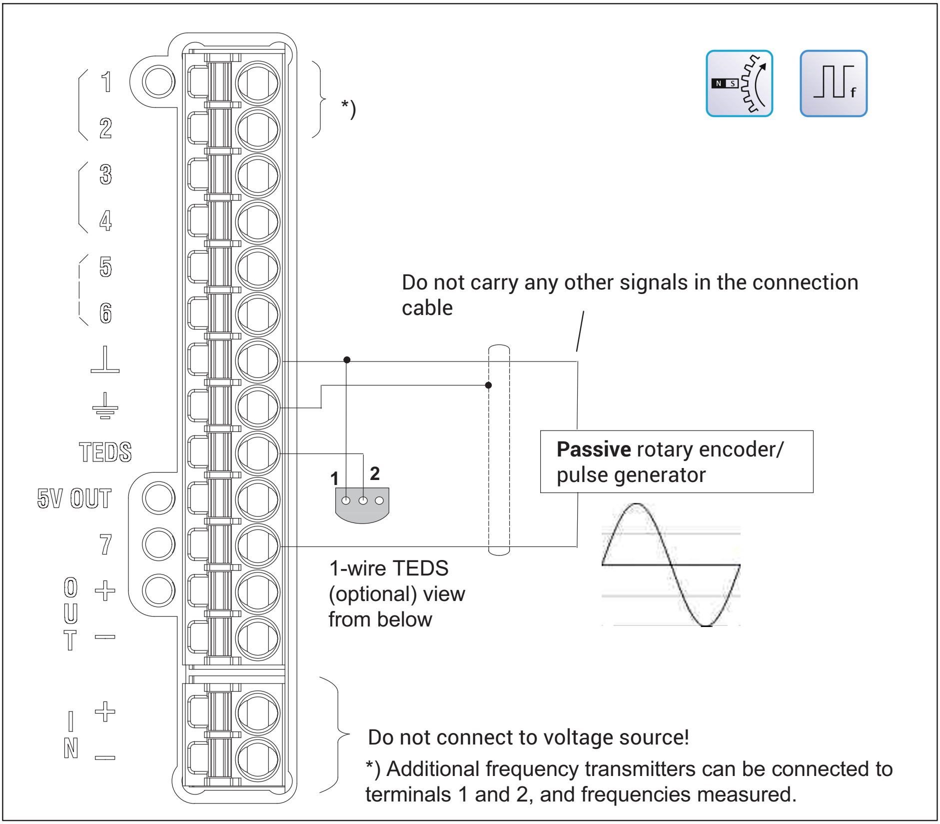


Fig. 8.38 PX460 pin assignment for rotary encoders and pulse generators

Possible channel setting via web server:

Transducer: Frequency (inductive)

Important

This signal input is designed for passive pulse generators only.

Pin 7 not connected or wired as output:

If pin 7 is connected as an input, the two potential ranges are interconnected (see *section 8.4.33*)

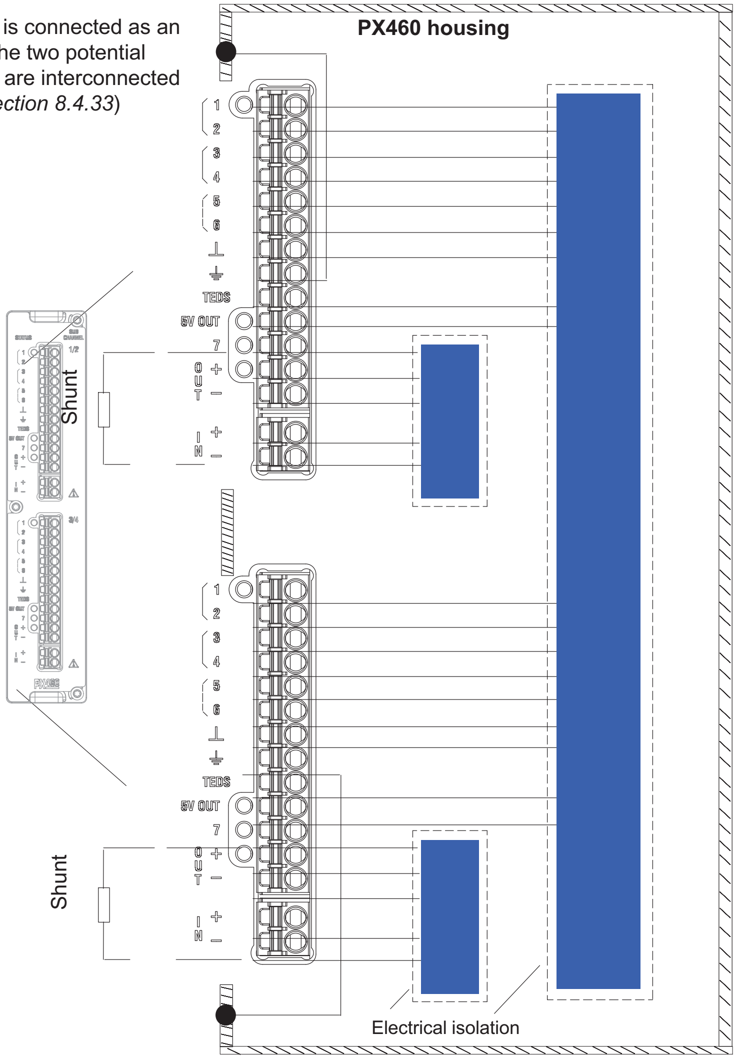
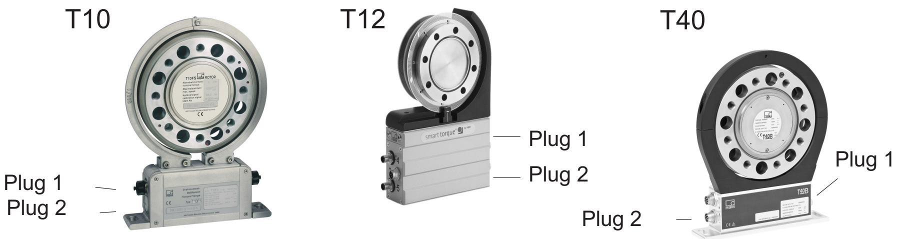


Fig. 8.39 PX460 electrical isolation

8.4.34 Connection and configuration of HBM torque transducers (T10, T12, T40)



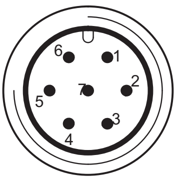
Plug 1: 1-KAB153-6 (torque)

Plug 2: 1-KAB154-6 (rotational speed); 1-KAB164-6 (rotational speed and angle of rotation)

Fig. 8.40 Plug positions for T10, T12, T40

Assignment for plug 1: T10, T12, T40

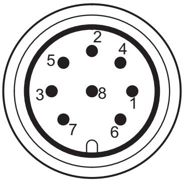
Supply voltage and frequency output signal.

Device plug	Plug pin	Assignment	Wire color
 Top view	1	Torque measurement signal (frequency output; 5 V[1),2)])	wh
	2	Supply voltage 0 V;	bk
	3	Supply voltage 18 V … 30 V	bl
	4	Torque measurement signal (frequency output; 5 V[1),2)])	rd
	5	Measurement signal 0V, balanced	gy
	6	Shunt signal resolution 5 V … 30 V	gn
	7	Shunt signal 0 V	gy
		Shielding connected to housing ground	

1) RS-422 complementary signals; with cable lengths exceeding 10 m, we recommend using a termination resistor R = 120 ohms between the (wh) and (rd) wires.
2) RS-422: Pin 1 corresponds to A, Pin 4 corresponds to B.

Assignment for connector 2: T10, T12, T40

Rotational speed output signal, reference pulse (optional)

<table>
<tr><th>Device plug</th><th>Plug pin</th><th>Assignment</th><th>Wire color</th></tr>
<tr><td rowspan="9">
Top view</td><td>1</td><td>Rotational speed measurement signal [1]
(pulse sequence, 5 V; 0°)</td><td>rd</td></tr>
<tr><td>2</td><td>Reference signal (1 pulse/revolution, 5 V)[1]</td><td>bl</td></tr>
<tr><td>3</td><td>Rotational speed measurement signal
(pulse sequence, 5 V; 90° phase shifted)</td><td>gy</td></tr>
<tr><td>4</td><td>Reference signal (1 pulse/revolution, 5 V)[1]</td><td>bk</td></tr>
<tr><td>5</td><td>Not in use</td><td>vt</td></tr>
<tr><td>6</td><td>Rotational speed measurement signal [1]
(pulse sequence, 5 V; 0°)</td><td>wh</td></tr>
<tr><td>7</td><td>Rotational speed measurement signal
(pulse sequence, 5 V; 90° phase shifted)</td><td>gn</td></tr>
<tr><td>8</td><td>Supply voltage zero</td><td>bn</td></tr>
<tr><td></td><td>Shielding connected to housing ground</td><td></td></tr>
</table>

1) RS-422 complementary signals; with cable lengths exceeding 10 m, we recommend using a termination resistor of R = 120 ohms.

Connection examples (torque transducers):

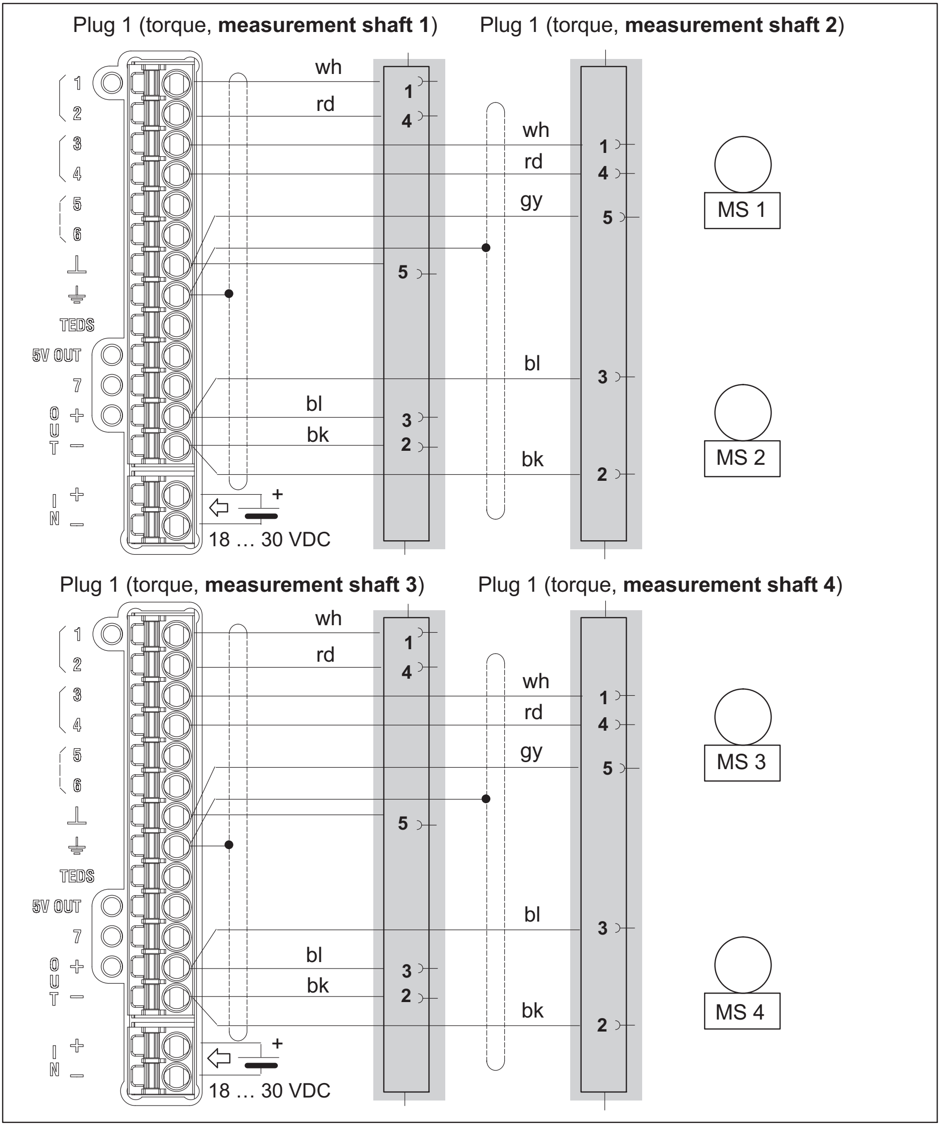


Fig. 8.41 PX460: Four torque transducers, torque only

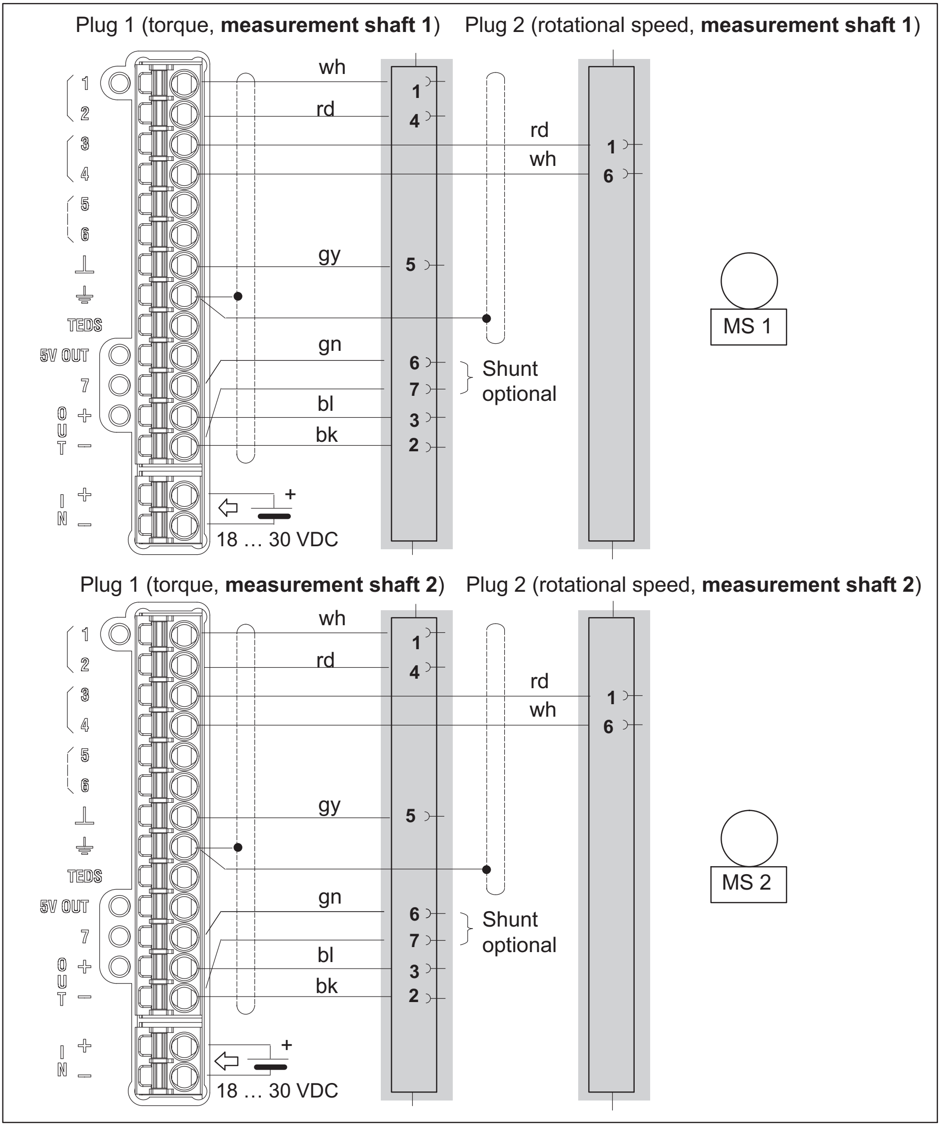


Fig. 8.42 PX460: Two torque transducers, torque and rotational speed without angle/direction of rotation

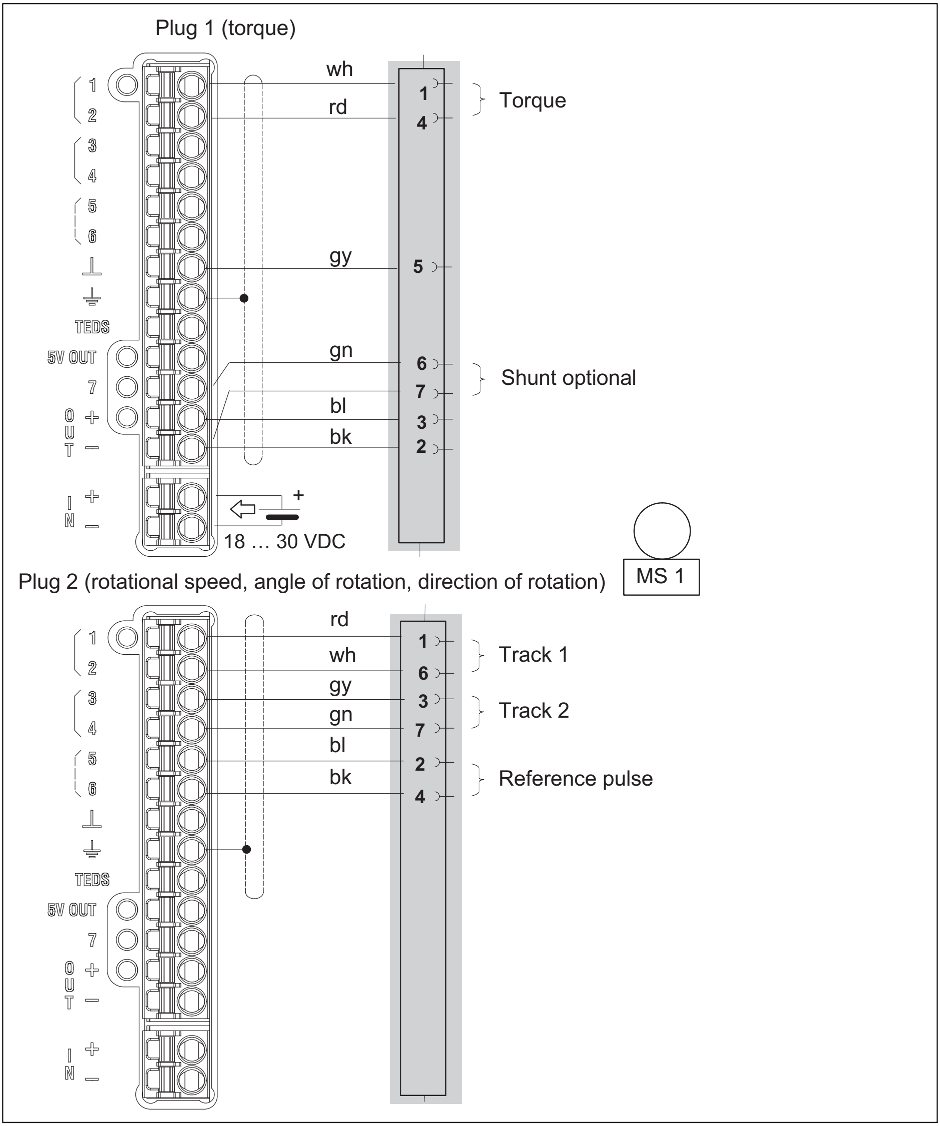


Fig. 8.43 PX460: One torque transducer, torque, rotational speed and angle/direction of rotation

Setup example (web browser): T40B on PMX

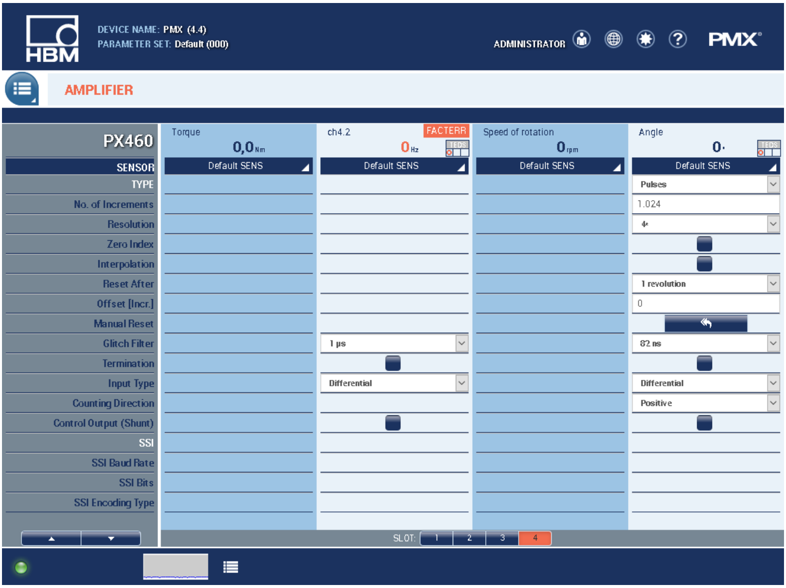
Channel 1 (PX460 top plug):
Torque, center frequency 10 kHz,
nominal torque 1 kNm

Channel 2 (PX460 top plug):
Not used

Channel 3 (PX460 bottom plug):
Rotational speed in rpm, 1024 pulses

Channel 4 (PX460 bottom plug):
Angle of rotation in degrees; 4x resolution,
so 4096 pulses = 360°

Based on the measurement principle (pulse counting), the measured values fluctuate around the true value. So test whether you need a filter, such as 500 Hz. The settings in the web browser are shown in the following screenshots.

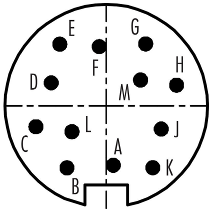


8.4.35 Connection and configuration of HBM torque transducer T210

Fig. 8.44 T210

T210 plug assignment

For the wiring of the supply voltage and the output signals on the plug refer to the following table. The wire colors correspond to the transducer connection cable, e.g. in the 5 m long version, ordering no. 3-3301.0158.



Pin	Assignment	Wire color	Control signal trigger (without VK20A)
A	Torque measurement signal (frequency output; 5 V)[1,2]	bk	
B	Rotational speed/angle of rotation measurement signal A; 5 V	rd	
C	Torque measurement signal ±10 V	br	
D	Torque measurement signal 0 V	wh	
E	Ground (supply + rotational speed/ angle of rotation)	ye	
F	Supply voltage 10 V ... 30 V	vt	
G	Rotational speed/angle of rotation measurement signal B; 5 V; lagging by 90°	gn	
H	Rotational speed reference signal Z; 5 V	pk	Switch (NO)
J	Measurement signal - ready for measurement	gy	
K	Control signal trigger (shunt), 50% signal	gy/ pk	
L	Torque measurement signal (frequency output; 5V)[1,2]	bl/rd	
M	Not in use	bl	

1) RS-422 complementary signals. If signal quality problems occur, a terminating resistor R=120 ohms between the wires (bk) and (bl/rd) can improve the situation.
2) RS-422: Pin A corresponds to A, pin L corresponds to B.

You can measure torque and both rotational speed and angle of rotation with a PX460 measurement card. This occupies three of the four inputs. The connection to a PX460 is shown by *Fig. 8.45*.

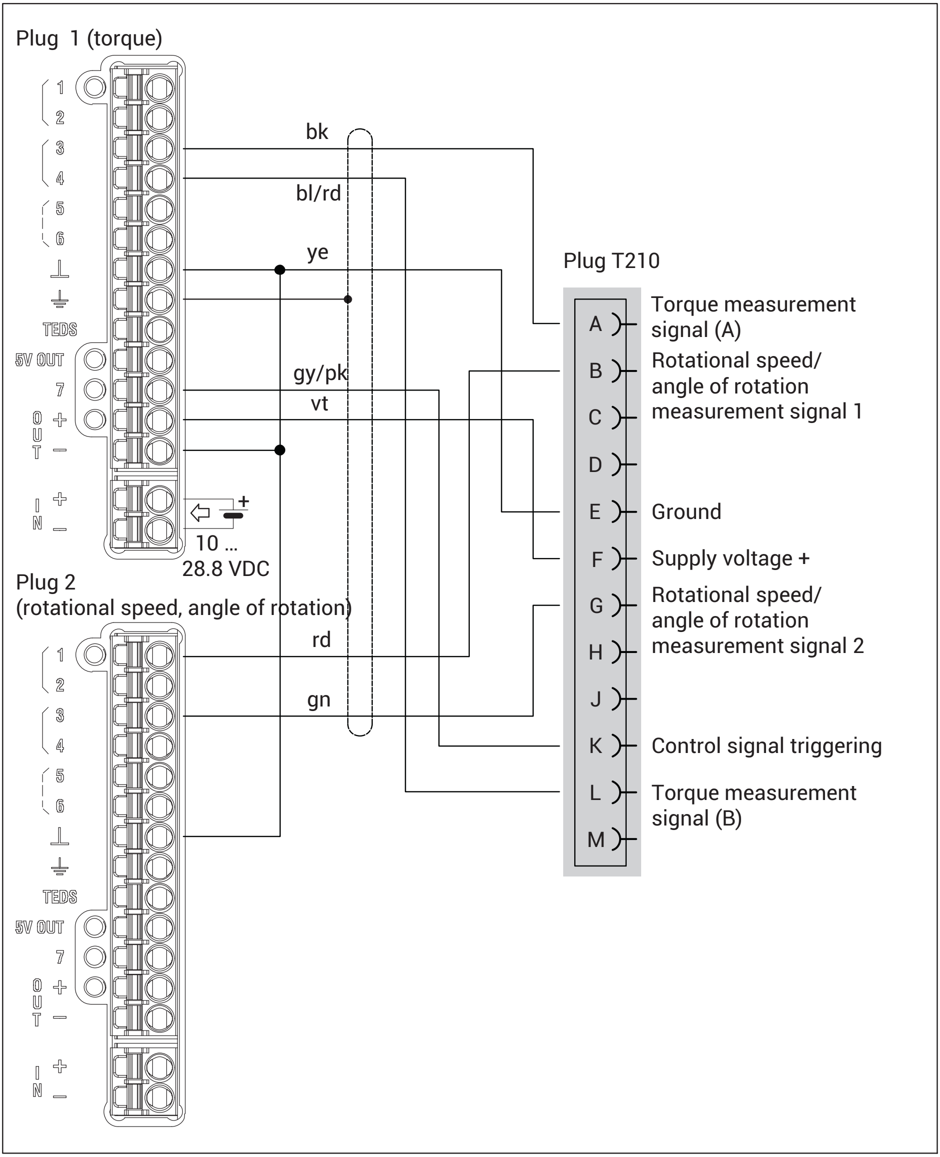


Fig. 8.45 PX460: T210 with torque, rotational speed and angle of rotation measurement

Setup example (web browser): T210 with 20 Nm on PMX

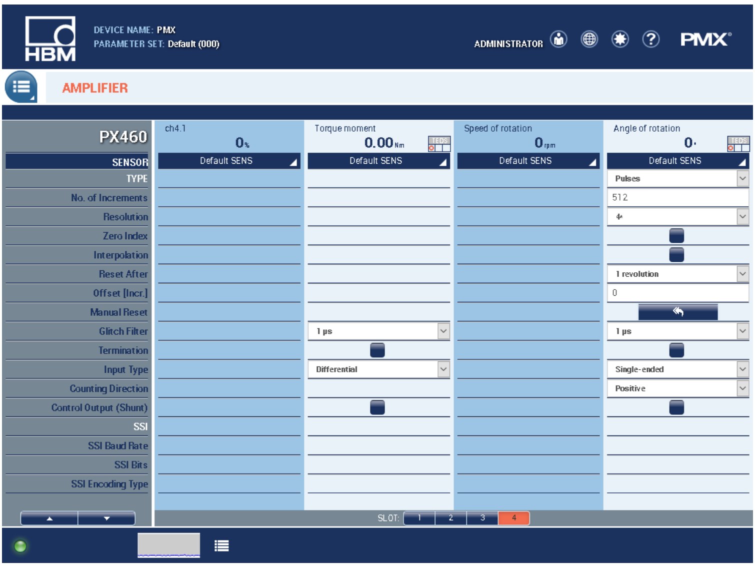
Channel 1:
Not used

Channel 2 (PX460 top plug), torque:
Sensor type: Frequency (digital)
Unit: Nm
1st point electrical: 10 kHz
1st point physical: 0 Nm
2nd point electrical: 15 kHz
2nd point physical: 20 Nm
Input type: Differential

Channel 3 (PX460 bottom plug), rotational speed:
Sensor type: Frequency (digital)
Unit: rpm
1st point electrical: 0 Hz
1st point physical: 0 rpm
2nd point electrical: 170.6667 kHz
2nd point physical: 20000 rpm

Channel 4 (PX460 bottom plug), angle of rotation:
Sensor type: Counter
Unit: °
1st point electrical: 0 pulses
1st point physical: 0 °
2nd point electrical: 2048 pulses
2nd point physical: 360 °
Type: Pulses
No. of Increments: 512
Resolution: 4*
Reset after: 1 revolution
Input type: Asymmetric
Count direction: Positive

Based on the measurement principle (pulse counting), the measured values fluctuate around the true value. So test whether you need a filter, such as 1000 Hz. The settings in the web browser are shown in the following screenshots.



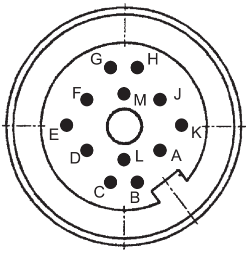
8.4.36 Connection and configuration of HBM torque transducer T21WN

Fig. 8.46 T21WN

T21WN plug assignment

For the wiring of the supply voltage and the output signals on the plug refer to the following table. The wire colors correspond to the transducer connection cable, e.g. in the 5 m long version, ordering no. 3-3301.0158.

You can measure torque and both rotational speed and angle of rotation with a PX460 measurement card. This occupies three of the four inputs. The connection to a PX460 is shown by *Fig. 8.47*.



Pin	Assignment	Wire color
A	Torque measurement signal (frequency output; 5 V)[1)][2)]	bk
B	Rotational speed/angle of rota-tion measurement signal 5 V	rd
C	Torque measurement signal ±10 V	br
D	Torque measurement signal 0 V	wh
E	Ground (supply+rotational speed/angle of rotation)	ye
F	Supply voltage +10 V ... 28.8 V	vt
G	Rotational speed/angle of rotation measurement signal 5 V, lagging 90°	gn
H	Not in use	pk
J	Measurement signal - ready for measurement	gy
K	Control signal triggering	gy/pk
L	Torque measurement signal (frequency output; 5 V)[1)][2)]	bl/rd
M	Voltage reference for rotational speed/angle[3)]	bl

Trigger control signal

Switch (NO)

1) RS-422 complementary signals; with cable lengths exceeding 10 m, we recommend using a R = 120 ohm termination resistor between the wires (bk) and (bl/rd), or pins A and L.

2) RS-422: Pin A corresponds to A, pin L corresponds to B.

3) Without an external voltage reference, the output for rotational speed, angle and ready for measurement returns a TTL level. If you require higher levels (such as for PLC inputs), preset a voltage reference 5 V < U < 24 V via pin M.

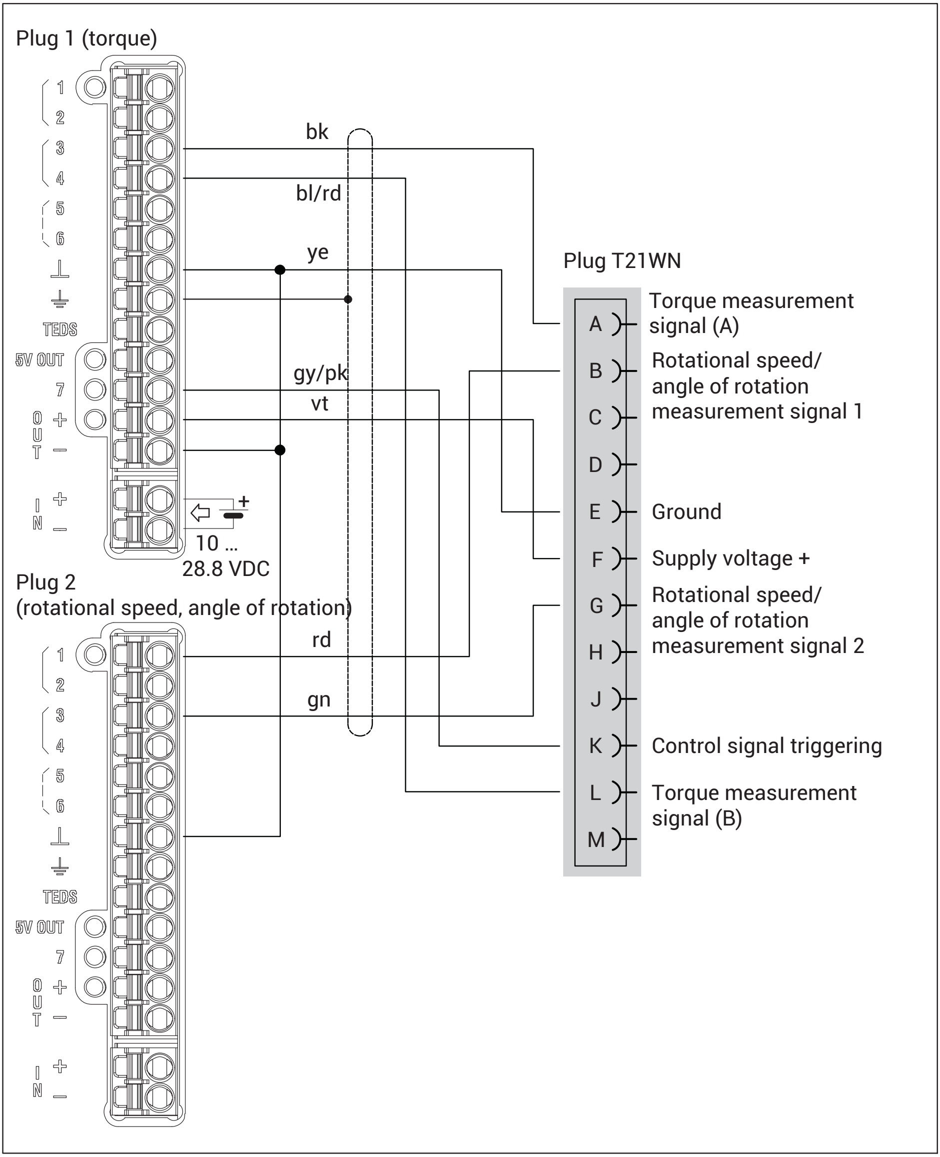


Fig. 8.47 PX460: T21WN with torque, rotational speed and angle of rotation measurement

Setup example (web browser): T21WN with 20 Nm on PMX

Channel 1:
Not used

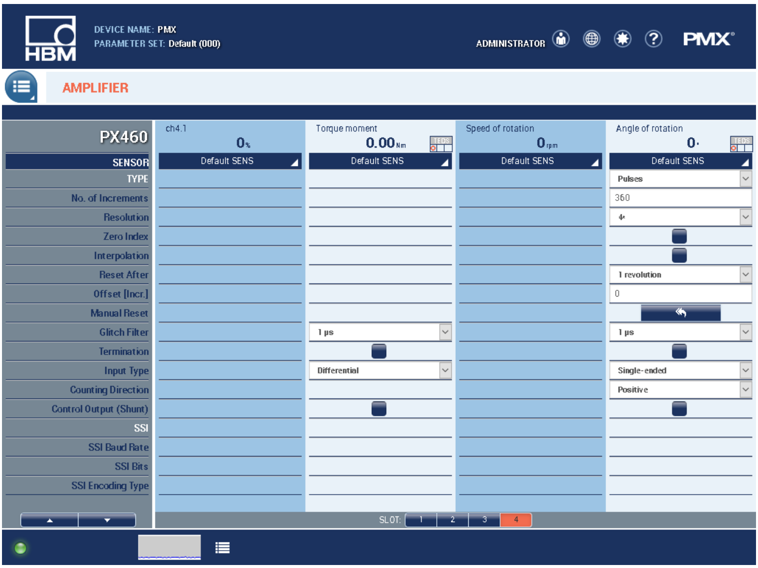
Channel 2 (PX460 top plug), torque:
Sensor type: Frequency (digital)
Unit: Nm
1st point electrical: 10 kHz
1st point physical: 0 Nm
2nd point electrical: 15 kHz
2nd point physical: 20 Nm
Input type: Differential

Channel 3 (PX460 bottom plug), rotational speed:
Sensor type: Frequency (digital)
Unit: rpm
1st point electrical: 0 Hz
1st point physical: 0 rpm
2nd point electrical: 360 Hz
2nd point physical: 60 rpm

Channel 4 (PX460 bottom plug), angle of rotation:
Sensor type: Counter
Unit: °
1st point electrical: 0 pulses
1st point physical: 0 °
2nd point electrical: 1440 pulses
2nd point physical: 360 °
Type: Pulses
No. of Increments: 360
Resolution: 4*
Reset after: 1 revolution
Input type: Asymmetric
Count direction: Positive

Based on the measurement principle (pulse counting), the measured values fluctuate around the true value. So test whether you need a filter, such as 1000 Hz. The settings in the web browser are shown in the following screenshots.


DEVICE NAME: PMX
PARAMETER SET: Default (000)
ADMINISTRATOR
PMX
AMPLIFIER
PX460
SENSOR
SENSOR TYPE
PHYSICAL UNIT
CHARACTERISTICS
1. Point Electrical
1. Point Physical
2. Point Electrical
2. Point Physical
SIGNAL CONDITIONING
Zero Value
Zero Target Value
CONTROL FUNCTIONS
Zero by
Clear Zero by
Test Signal
DATA ACQUISITION
Channel Name
Type
Cutoff Frequency (-3dB)
ch4.1
Torque moment
Speed of rotation
Angle of rotation
Default SENS
Frequency (digital)
Count
Default DAQ
Bessel
2000 Hz
1000 Hz
SLOT: 1 2 3 4


DEVICE NAME: PMX
PARAMETER SET: Default (000)
ADMINISTRATOR
PMX
AMPLIFIER
PX460
SENSOR
TYPE
No. of Increments
Resolution
Zero Index
Interpolation
Reset After
Offset [Incr.]
Manual Reset
Glitch Filter
Termination
Input Type
Counting Direction
Control Output (Shunt)
SSI
SSI Baud Rate
SSI Bits
SSI Encoding Type
ch4.1
Torque moment
Speed of rotation
Angle of rotation
Default SENS
Pulses
360
1 revolution
1 µs
Differential
Single-ended
Positive
SLOT: 1 2 3 4

8.4.37 Connection and configuration of HBM torque transducer T20WN (without VK20A)

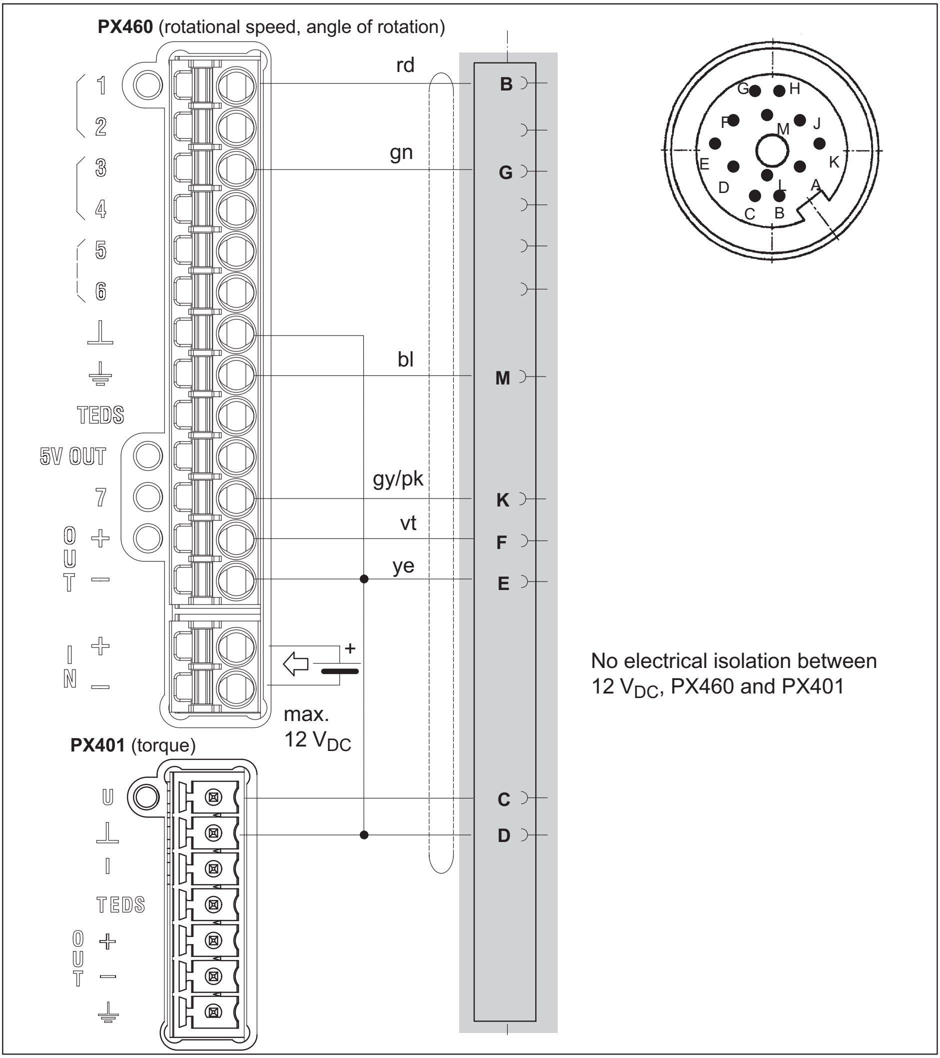
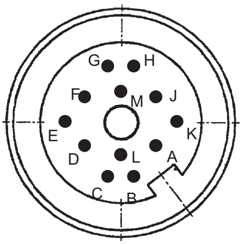


Fig. 8.48 Pin assignment for T20WN without VK20A

Notice

The voltage supply of torque transducer T20WN must not exceed 12 V. It can be provided by an external power supply unit (max. 12 V) or by the PMX device (with max. 12 V PMX device supply).
Alternatively, you can also connect the transducer to the PMX device via the VK20A terminal box. It can be supplied with max. 30 V (including via the PMX).

T20WN plug assignment



Pin	Assignment	Wire color
A	Not in use	bk
B	Rotational speed/angle of rotation measurement signal 5 V	rd
C	Torque measurement signal ±10 V	br
D	Torque measurement signal 0 V	wh
E	Ground (supply+rotational speed/ angle of rotation)	ye
F	Supply voltage +12 V	vt
G	Rotational speed/angle of rotation measurement signal 5 V, lagging 90°	gn
H	Not in use	pk
J	Not in use	gy
K	Control signal triggering	gy/ pk
L	Not in use	bl/rd
M	Cable shield	bl

Transducer connection cable:

3-3301.0158, 5 m
3-3301.0159, 10 m

8.4.38 Connection and configuration of HBM torque transducer T20WN (with VK20A)

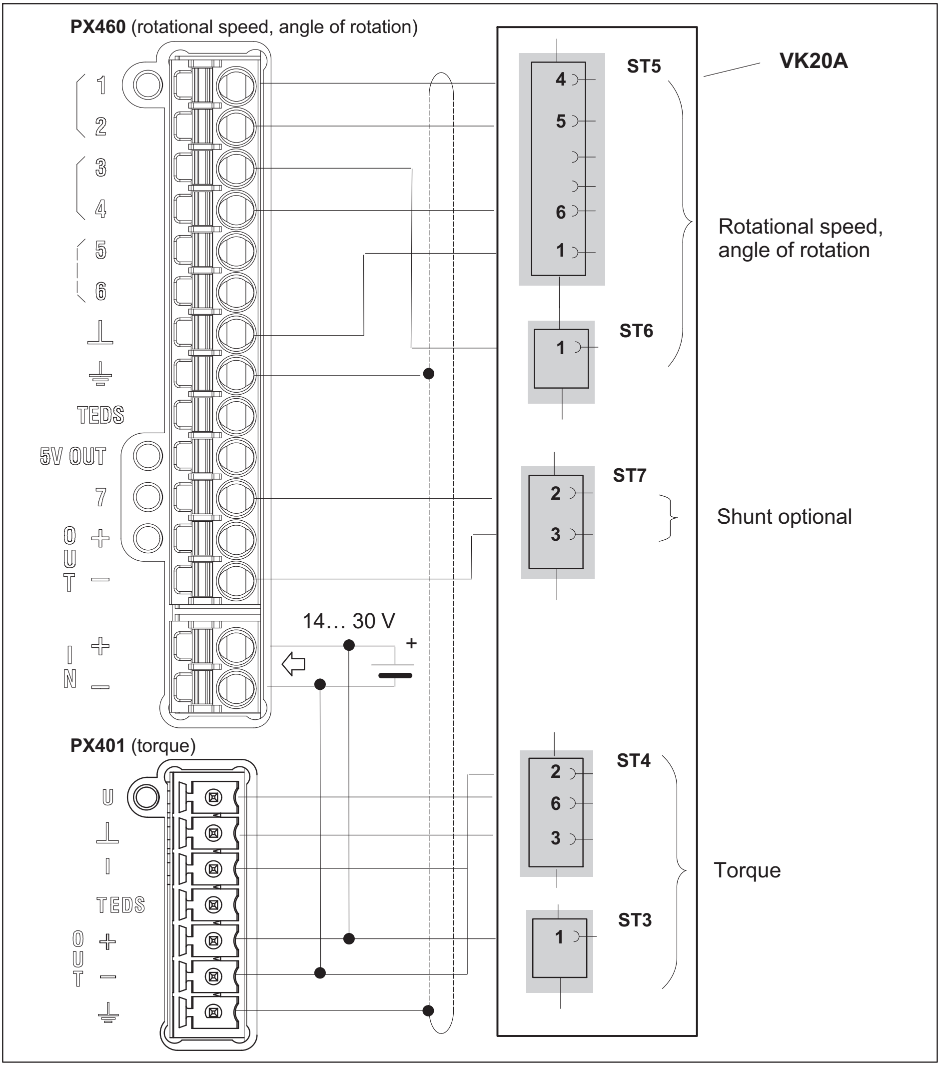
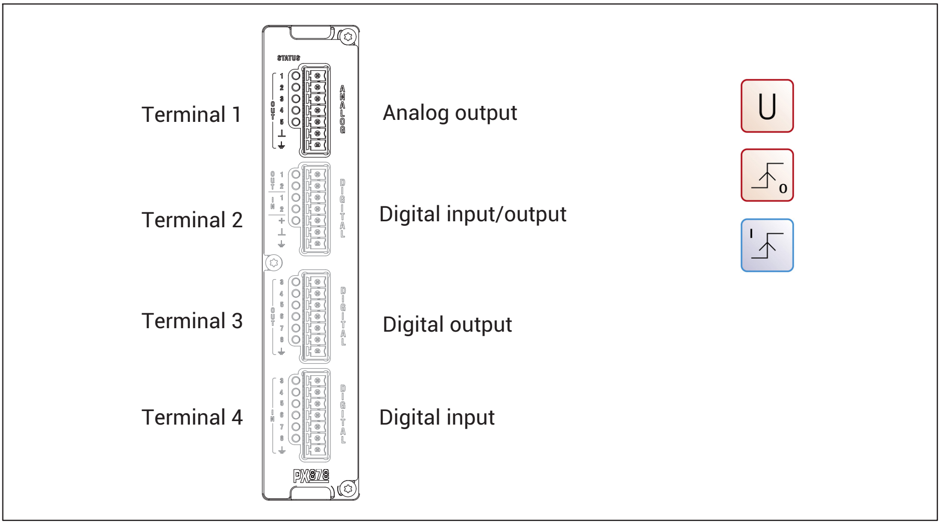


Fig. 8.49 Pin assignment for T20WN with VK20A

8.5 Input/output cards

8.5.1 PX878

Eight digital inputs, eight digital outputs and five analog voltage outputs



8.5.2 Analog output ± 10 V

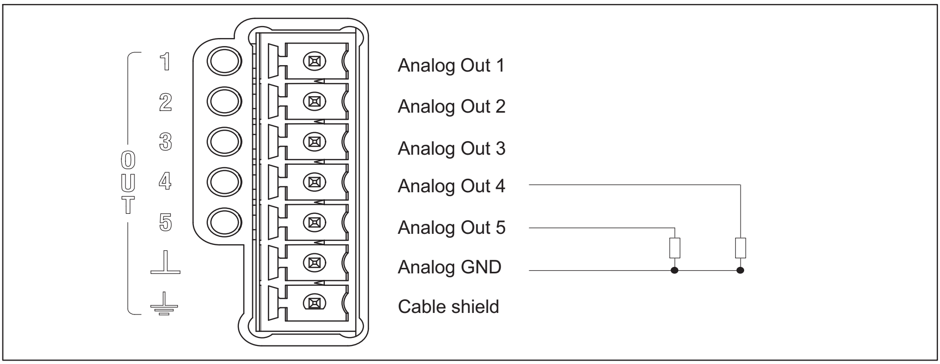


Fig. 8.50 Analog output pin assignment (terminal 1)

8.5.3 Digital inputs and outputs

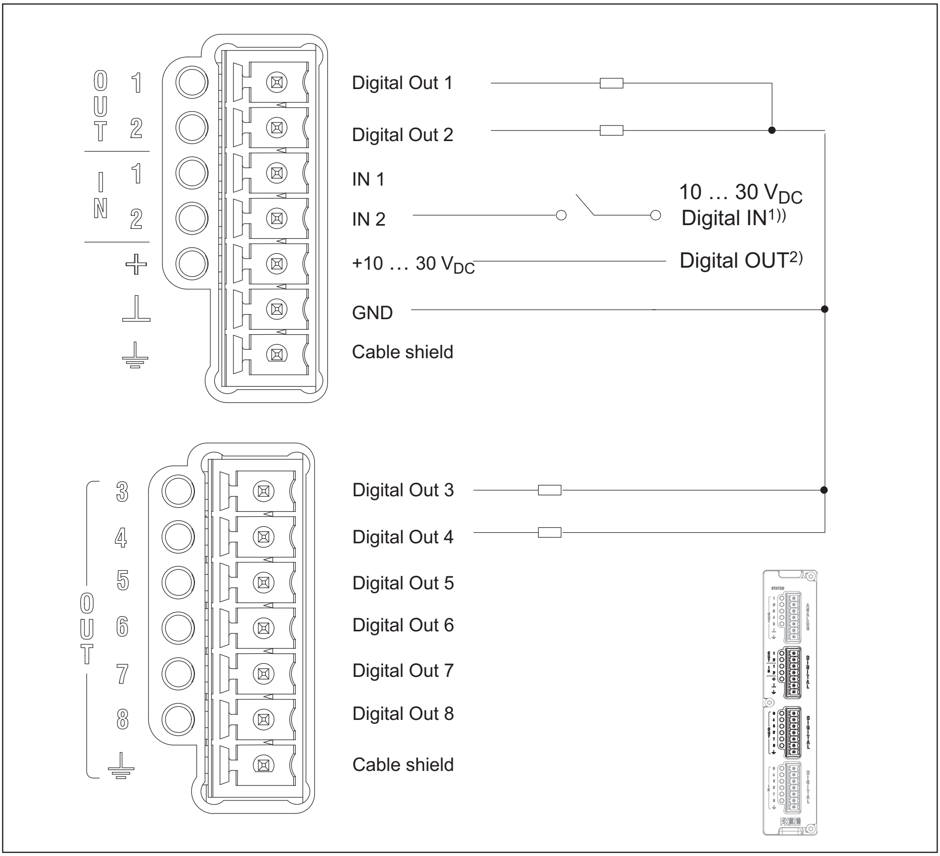


Fig. 8.51 PX878: Digital input/output pin assignment (terminals 2 and 3)

1) External voltage or signal from external source.
2) External voltage supply for Digital OUT, e.g. for power supply to the PMX (POWER).

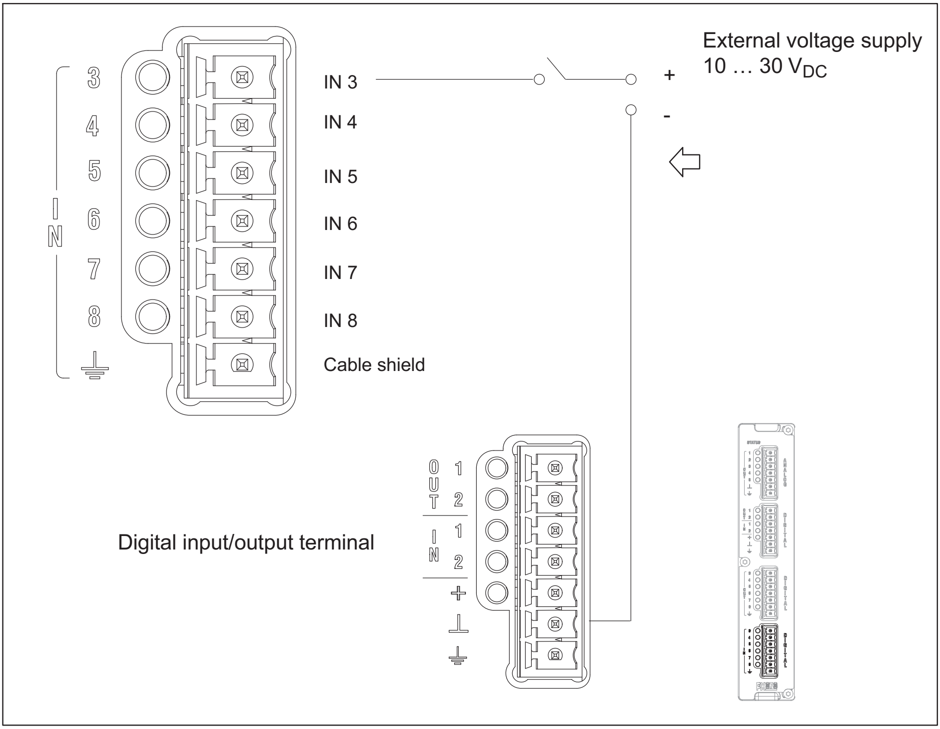


Fig. 8.52 PX878: Digital input pin assignment (terminal 4)

Important

Specify the functions of the control inputs and outputs and the analog outputs via the PMX web server. The digital input must be PNP (switching to plus). An open input will be detected as "low".

8.5.4 External supply voltages for the digital inputs and outputs (PX878)

Example: PLC connection (p-switched)

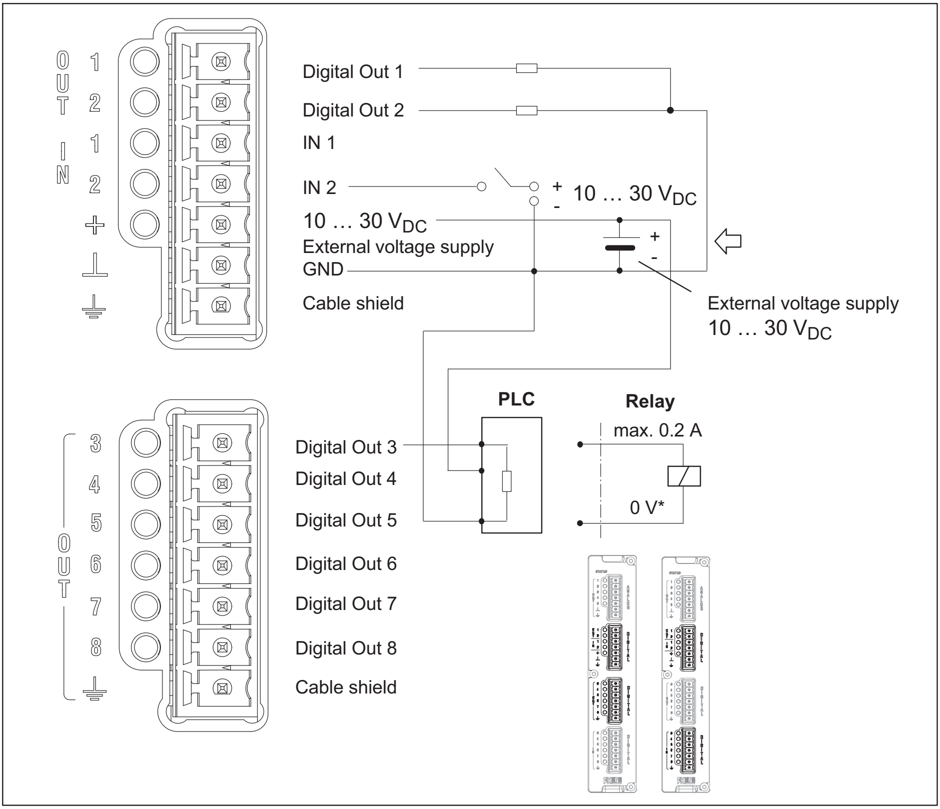


Fig. 8.53 PX878: Digital input and output pin assignment

The *control outputs* are available at the Digital OUT 1 and OUT 2 terminals, as well as at OUT 3 to OUT 8. They are executed as high-side switches, and are electrically isolated from the PMX housing, but not from each other (see *Fig. 8.54*).

The *control inputs* are available at the Digital IN 1 and IN 2 terminals, as well as at IN 3 to IN 8. They are electrically isolated from the PMX housing, but not from each other (see *Fig. 8.54*).

Important

Output behavior after switching on

- The digital outputs have a high resistance after switch-on, and retain this status until the state changes to active. The change to active status is dependent on the firmware and the preset actions.
- In the active state, the externally connected voltage source (see terminals + and ⊥) is internally switched through at low resistance by an electronic switch (high-side switch)
  Note: The electronic switch switches the + pole of the voltage source.
- In the active state, the electronic switch has a high resistance. If a defined state is expected for this situation (e.g. the electronic input of a controller), you must use a terminating resistor (pull-down) to terminate the high resistance state.
- For the control inputs, you must connect an external reference potential (⊥ IN) to which the control input signals refer.

Important

The PX878 I/O card has electrical isolation between the analog and digital sections and the basic device.

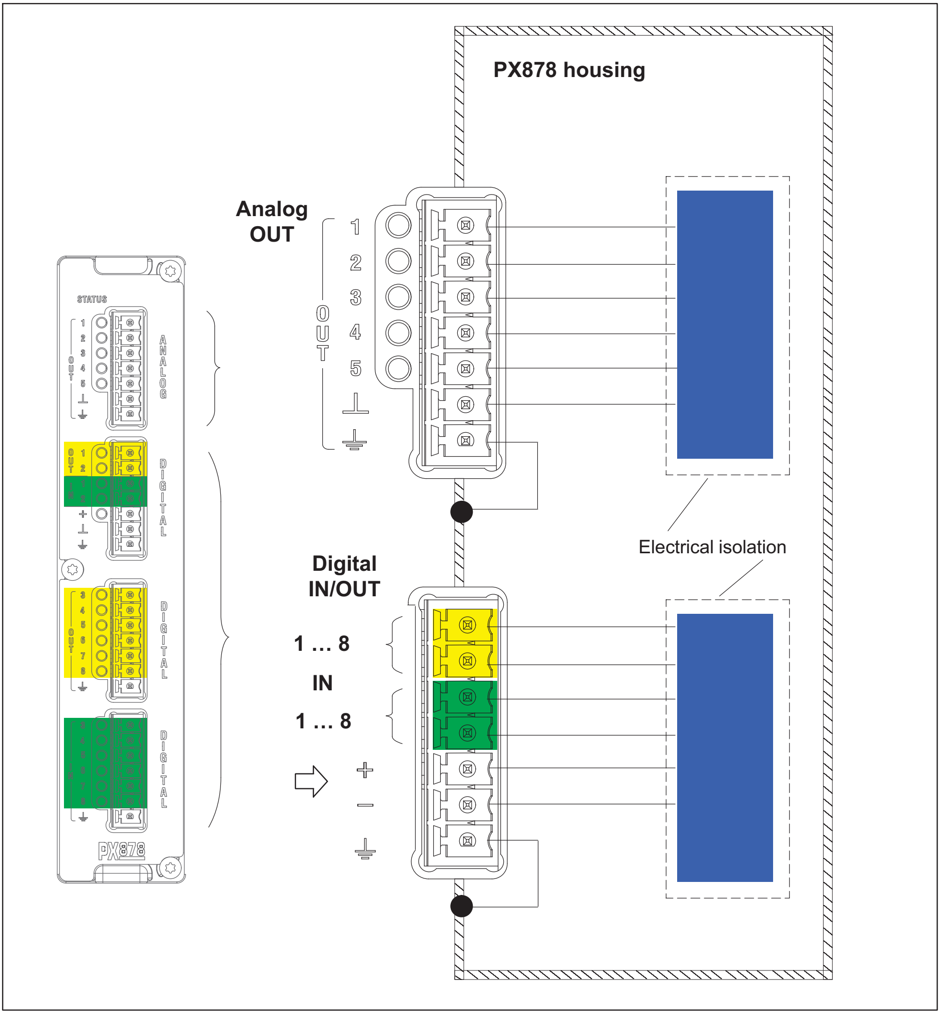


Fig. 8.54 PX878 electrical isolation

8.6 Communication cards

8.6.1 Pin assignment for the PX01EC EtherCAT® communication card

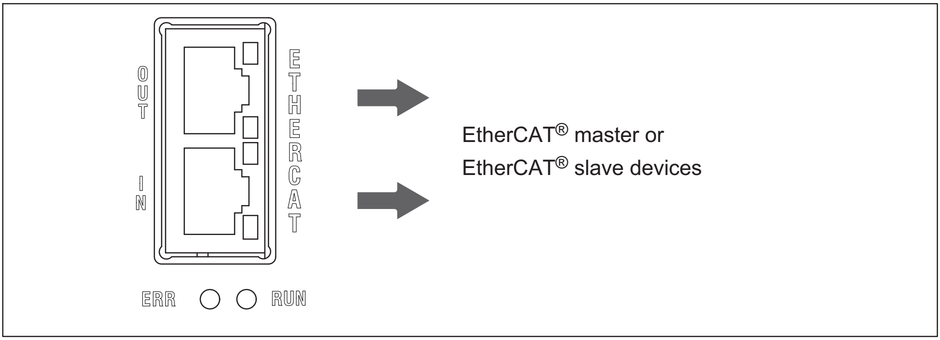


Fig. 8.55 EtherCAT® connection as per standard[1)]

8.6.2 Pin assignment for the PX01EP EtherNet/IP™ communication card

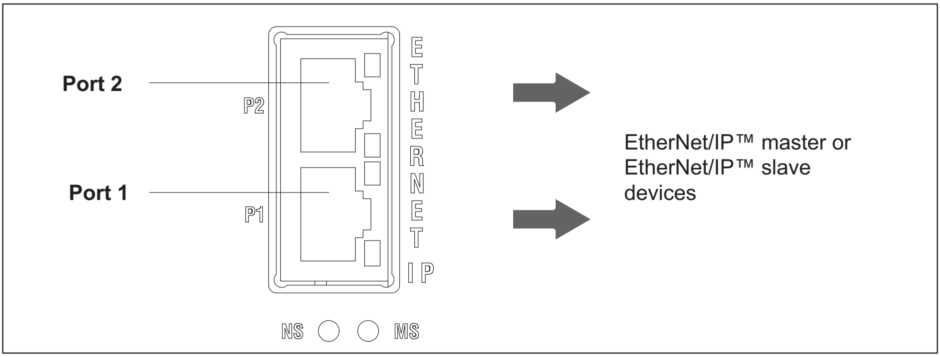


Fig. 8.56 EtherNet/IP™ connection as per standard[1)]

1) See user organization standards

8.6.3 Pin assignment for the PX01PN PROFINET® IO communication card

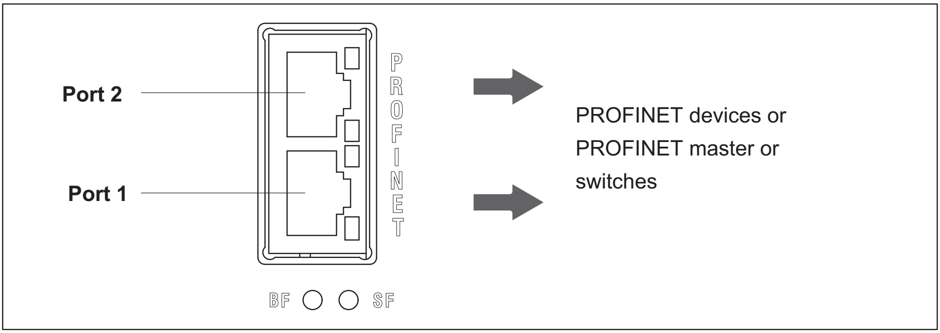


Fig. 8.57 PROFINET® IO connection as per standard[1)]

8.7 TEDS transducer

8.7.1 TEDS connection

TEDS stands for "Transducer Electronic Data Sheet". Transducers with an electronic data sheet as defined in the IEEE 1451.4 standard can be connected to the PMX system, making it possible for the amplifier to be set up automatically. A suitably equipped amplifier imports the transducer characteristics (electronic data sheet), translates them into its own settings, and you can start the measurement.[1)]

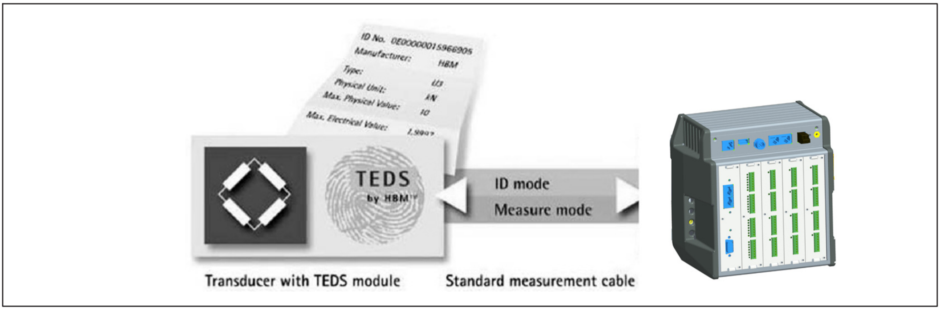


Fig. 8.58 PMX with TEDS technology

The IEEE standard 1451.4 defines a generally acknowledged process with which sensors can be identified. The sensor is identified by its respective data sheet, which is stored in electronic form in the sensor, in the cable, or in the plug on a TEDS module. The amplifier communicates with this TEDS module, reads the data sheet, and makes the corresponding measuring amplifier settings.

1) See user organization standards

Two types of TEDS chip connection are used in the PMX:

- *Zero-wire TEDS:* PX455

  Here the TEDS module is connected to the transducer sense leads. This means that no further wires are required; the measurement is briefly interrupted to read the TEDS information into the PMX.
- *1-wire TEDS*: PX401, PX460

  Here the TEDS module is connected to the PMX measurement channel by two additional wires (see *page 80*).

8.7.2 Starting up the TEDS module

Sensors can be fitted with TEDS at the factory and delivered in the written state.

The TEDS modules already mounted in the sensor and plug can also be configured and parameterized subsequently using a TEDS editor.

8.7.3 PMX parameterization with TEDS

If a transducer with TEDS is connected, and contains the parameterization data for a sensor, this can be used to automatically parameterize the PMX.

The PX455 measurement card features zero-wire TEDS. Here, the sense leads of the sensor cable are used to read or write to the TEDS module.

! **Important**

The PMX uses only the TEDS 2-point scaling. Scales that are stored as tables or polynomials cannot be loaded. You can, however, use one of the PMX's internal calculation channels for such linearizations (see chapter 13, "Internal calculation channels", page 174).

On the PX401 and PX460 measurement card, the TEDS module is addressed separately via an additional wire (1-wire TEDS).

In the PMX you can preset channel by channel how the PMX is to respond after being switched on or connecting TEDS sensors:

- Ignore existing TEDS.
- Load TEDS and configure measurement channel with it only if a TEDS sensor is present.
- Always load TEDS and configure measurement channel with it, otherwise report error.

When the PMX is switched on, it automatically detects whether a sensor with TEDS is connected. The data is read out, and the amplifier channel is parameterized accordingly. If you replace a sensor with TEDS while in the On state, the new TEDS module will also be detected automatically, but you must activate it manually.

Important

The PMX web server does not have a TEDS editor with read and edit functions. The catman®Easy/AP software includes a full TEDS editor. This allows TEDS information to be read from and written to TEDS sensors connected to the PMX.

9 SYNCHRONIZATION AND TIME RECORDING

If measurement signals need to be referenced over time with each other for processing and analysis, for measurement data acquisition tasks (DAQ), for example, they must be synchronously recorded. That is why all PMX modules can be synchronized with one another. This ensures simultaneous measurement on all channels. An internal counter in each PMX is used for the purpose.

Counter format used: 48-bit integer
Counter frequency: 153.6 kHz

- At a sampling rate of 19.2 kHz (factory setting) the counter is therefore incremented by 8 from one measured value to the next.
- At a sampling rate of 38.4 kHz the counter is incremented by 4 from one measured value to the next.

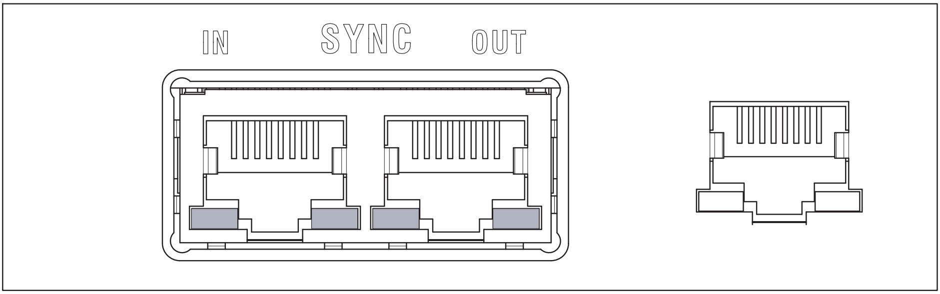
These counters are transmitted together with each measured value. The absolute measurement time must come from a different source. This can be the internal time of the PMX, or another external clock. The measurement data acquisition system then synchronously combines the measured values with the counter and time stamp.

To achieve a precise reference over time, you should parameterize the channels with the same filter settings. No automatic phase delay correction is carried out for different filters. The filter phase delays are specified in the data sheet.

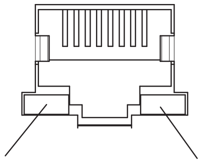
9.1 Synchronizing via internal PMX synchronization

All the modules are synchronized automatically, if they are interconnected in series by an Ethernet patch cable. This is the recommended method. It synchronizes the counters and carrier frequencies of all type PX455 measurement cards. This method only synchronizes PMX modules with one another however.

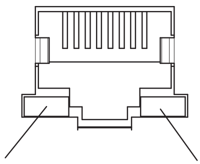
The synchronization status is indicated by the SYNC socket LEDs. The synchronization master/slave allocation is automatic, meaning one PMX device is automatically selected as the time master.



IN socket LEDs:

IN		Meaning
		
Green	Off	Slave
Off	Off	Master
Green	Yellow	Error

OUT socket LEDs:

OUT		Meaning
		
Green	Off	Voltage supply available
Green	Yellow	Error (always identical to the right-hand LED of the **IN** socket)

You can synchronize a maximum of 20 PMX devices. The first PMX device automatically becomes the master. The maximum cable length between adjacent devices is 30 meters. Recommended cable: Standard Ethernet Cat-5-SFTP.

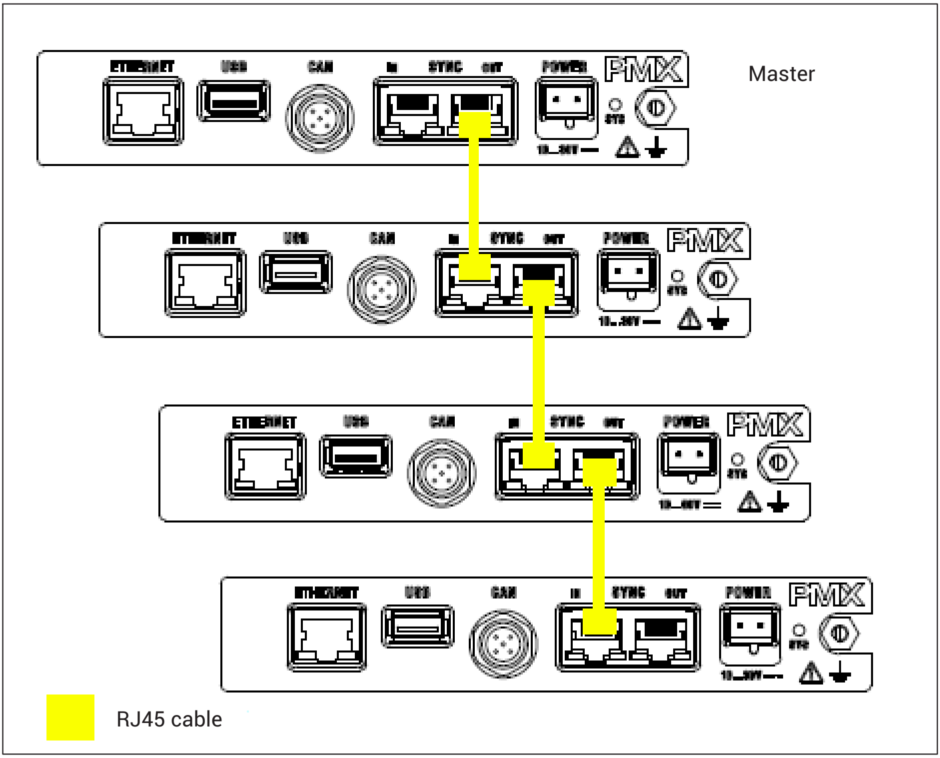


Fig. 9.1 Connecting multiple PMX modules

Important

The SYNC connection is not used to supply voltage to the devices.
SYNC sockets are not the same as Ethernet. Do not connect to Ethernet and do not configure as a ring.
In the event of a power failure, the time stamp is not buffered, but starts again from zero after restarting.

9.2 External synchronous data acquisition via an NTP server in the network

If synchronous measurements are to be performed by different measurement systems, an external master is required for synchronous acquisition.

Each PMX device has an NTP time that can be set by an external NTP server. The NTP time is distributed to all the devices over the Ethernet (TCP/IP) connection.

The PMX devices then transmit their measured values with the counter signal and NTP time. This means the DAQ systems can use this information to synchronously acquire the measured values of all the devices.

With NTP you can achieve accuracies of 1 ms and better when running PMX with other devices. But this depends on the utilization of the particular network, as well as on the quality of the NTP master. An NTP software program is included in the HBM catman® software suite.

Time format used

Base	1.1.1990
Time stamp:	64 bit 32 bit seconds 32 bit fractions of seconds, resolution (1/232)

Further information about NTP can be found at http://www.ntp.org.

9.3 Data acquisition via fieldbus: EtherCAT®, PROFINET® IO, EtherNet/IP™

The measured values cannot be synchronized over fieldbuses, and cannot be time-stamped, as the PX01EC EtherCAT® fieldbus card does not support EtherCAT® Distributed Clocks. The same is true for the PROFINET® IO and EtherNet/IP™ fieldbus cards. Consequently, the time cannot be distributed from a fieldbus master to a PMX working as a fieldbus slave.

In this type of master/slave system, however, the measured values and data, such as peak values and status information, are transmitted deterministically. The measured value counters are also transferred, so that the system master can process the measurement data synchronously.

9.4 Comparison of synchronization mechanisms

PMX feature	Intrinsic PMX synchronization	Ethernet (NTP)
Synchronization with other devices	PMX only	PMX, QuantumX, MGCplus, interrogators, other
Measured value time recording	Internal PMX counter (48-bit value) plus PMX time	NTP time signal from external NTP server
Synchronization accuracy	<1 µs	100 µs to approx. 10 ms
Number of modules to be synchronized	20	unlimited
Maximum distance between modules	30 m	100 m (Ethernet), 10 km with specific WLAN bridges
Synchronization settling time	Immediate	About 20 minutes on initial startup, about 2 minutes on restarting
Synchronization master	Automatic, the first PMX is the master	Recommended: separate NTP server/master

10 NETWORK, DATA SECURITY, POLICIES

10.1 Network access and remote maintenance

The web server in the PMX uses the hypertext transfer protocol (http) to transmit data over a computer network and load its user interface (hypertext documents) into a web browser. It is not possible to use hypertext transfer protocol secure (https) for encrypted transmission.

To allow access via one of the networks, various protocols (ports) must be enabled in the PC's or server's firewall so that the listed software components can be used.

- **PMX web server user interface**

Port type	Number	Explanation
TCP	80, 55000	Parameterize and measure
UDP	31 416, 31417	IPv4 multicast for HBM device scan
UDP	1900, 5353	Only for name resolution via Zeroconf (similar to Bonjour)
UDP TCP	1900 8200	Only for name resolution via UPnP
UDP	137	Only for name resolution via NetBIOS
UDP	123	NTP time synchronization
UDP	514	System log messages (log)

- **catman / .NET-API / LabVIEW driver / DIAdem driver**

Port type	Number	Explanation
TCP	80, 55000	Parameterize and measure
UDP	31 416, 31 417	IPv4 multicast for HBM device scan

Notice

As of firmware 3.0, only two simultaneous Ethernet connections are allowed on port 55000.
If you open a third connection, one of the other two will be terminated. For this purpose, the time of the last activity of the existing connections is recorded internally, and the older one is terminated.

- **CODESYS (only for WGX001)**

Port type	Number	Explanation
TCP UDP	1217 1217	Gateway
TCP	11740 … 11743	Communication
UDP	1740 … 1743	Broadcasts
TCP	8080	WebVisu

10.2 Data security

To minimize the risk of data security violations, we recommend taking the following organizational and technical measures for the system on which your applications are running:

- Avoid exposing the PMX and controller networks to open networks such as the Internet. Use additional security precautions to prevent this, such as a VPN for remote access, and install firewall mechanisms. In particular, the controller parameterization ports must not under any circumstances be accessible from the Internet without protection - see *section 10.1 on page 132*.
- Restrict access to authorized persons. Change any existing default passwords and access rights when first starting up. The procedure for changing PMX passwords and user right policies can be found in the PMX web server under **Change Password** and **Define Policies** - see also *section 10.3., "Policies definition and passwords", page 133*.

10.3 Policies definition and passwords

10.3.1 Policies definition

There are 3 levels to PMX policies definition: OPERATOR, MAINTENANCE and ADMINISTRATOR. The PMX settings can only be displayed or changed if you are at the MAINTENANCE or ADMINISTRATOR user level. At the OPERATOR user level, only the OVERVIEW, some SYSTEM SETTINGS and MONITORING (the LINE WRITER) are accessible. You can change the user level with  . The user level is reset to OPERATOR if there is no input for 10 minutes. The settings accessible at the MAINTENANCE user level can be configured in **Define Policies**, if you are at the Administrator level.

10.3.2 Passwords

The default setting is for no password to be set and you can switch directly to a different user level. But you can set a different password for both the MAINTENANCE and ADMINISTRATOR user levels, thus restricting access to the setting menus. At the OPERATOR user level, only the OVERVIEW, some SYSTEM SETTINGS and MONITORING (the LINE WRITER) are accessible. All the setting menus are accessible on the ADMINISTRATOR user level.

Click to select whether you would like to select a password for MAINTENANCE or ADMINISTRATOR. For reasons of security, you have to enter the password twice before you can exit the dialog with OK and the password is activated. The length of the password is limited to 10 characters. All characters are acceptable. Touching or clicking on  turns on the display of entered characters, and  turns it off again.

Notice

Keep the Administrator password somewhere safe. If you forget the password, you must contact HBM Technical Support. *See also section 25.3, "Resetting the PMX Administrator password", page 434.*

10.3.3 Presetting user level by controller

In firmware 3.02 and later with a connection via firmware or Ethernet interface you can activate the MAINTENANCE or ADMINISTRATOR user levels directly with a command. You must specify a time for which the deactivation will apply. The maximum possible is 24 hours. You can also disable the selected user level prematurely, or extend the time period.

The active user level is also available in the browser as from the time of switching and for the specified period of time, and is displayed accordingly. After a PMX restart (power-down) you have to re-release it.

11 START-UP

This chapter describes how to put the PMX into operation, how to configure it, and how to operate the user interface.

11.1 Hardware setup

11.1.1 Voltage supply/transducers

- Connect the power supply cable and the transducers to the module as described in *chapter 8, "PMX electrical connections", page 46*.
- Optionally connect the bus system (EtherCAT® or PROFINET® IO or EtherNet/IP™).
- Switch on the power supply.

11.1.2 Ethernet connection

In order to operate and parameterize the PMX via the integrated web server, you must connect it to a device that has a web browser, such as a PC.

In a point-to-point connection, use an Ethernet crossover cable or make sure that your PC's Ethernet interface has an autocrossing function available. This is the case with current devices.

Use only cables that are Category 5 (Cat 5) or higher for this purpose. This enables line lengths of up to 100 m to be achieved. You can also use a wireless (WLAN) connection.

Important

Make sure that HTTP port 80 is open in the firewall.

11.2 Integrated PMX web server

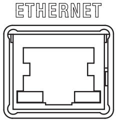
System requirements

To operate current versions of the PMX devices, you need a terminal (such as a PC or tablet with a mouse) running a current web browser (Internet Explorer version > 9.0, Firefox or Chrome) and with a screen resolution of 1024 x 768.

Windows XP must be installed on the PC as a minimum.

11.3 Connect the PMX to a PC (HOST) or via a network

▶ Connect the PMX to a PC/laptop or to a network via the ETHERNET socket.



Factory setting

- When the computer is powered-up, the PMX obtains the IP address via
  - DHCP (automatic address assignment as per RFC2131 and RFC2132) or
  - from the Apipa auto IP range (RFC5735) in the range 169.254.xxx.xxx
- The device name of the PMX is "pmx" (changeable).

Address assignment options

- Via a preset IP address (not factory-set).
- If no permanent IP address is assigned, the PMX waits for an address from the DHCP server. If there is no response from a DHCP server, an IP address is automatically chosen from the RFC5735 auto IP range.
- If the PMX is set to DHCP (factory setting), the PC must also use DHCP.

What are the options for finding the PMX in the network?

Option	Technology	Operating system
A	UPnP	Windows 7 and later
B	NetBIOS	Windows XP and later
C	Zeroconf (similar to Bonjour service)	Apple; Linux; Windows, if "Bonjour Print Services" is installed.
D	Broadcast ping	The address depends on the network settings. With a subnet mask of 255.255.0.0 and a network address of 192.168.169.123, for example, the resulting broadcast address is 192.168.255.255. However, this method does not work with all Windows configurations.

Tip

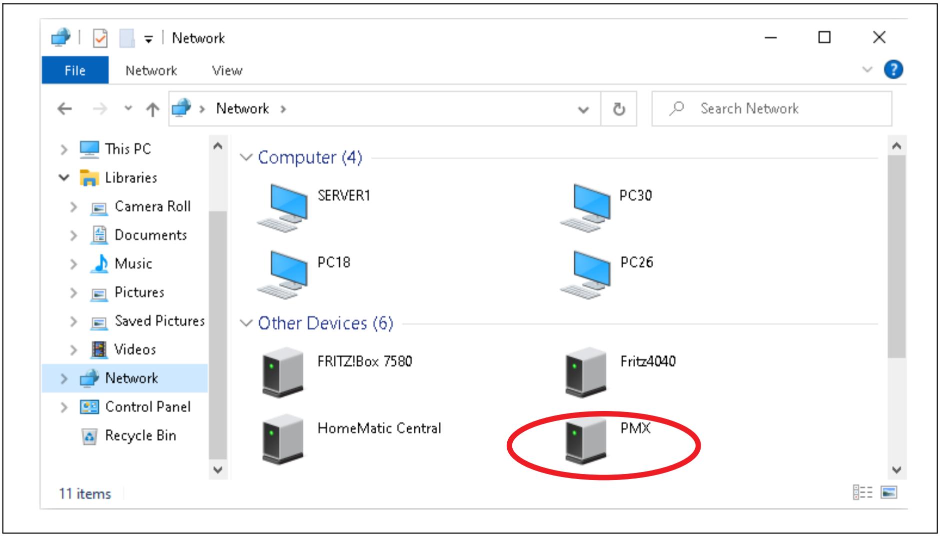
If a network connection does not materialize: plug the network cable in again! Then wait at least one minute, because assigning addresses via APIPA can take longer.

Option A:

Connection via UPnP (Universal Plug & Play) as from Windows 7

This connection depends on the network settings, and is also possible without DHCP and in the auto IP range[1]). It is not available with a PMX-PC connection (with no network) or in public networks.

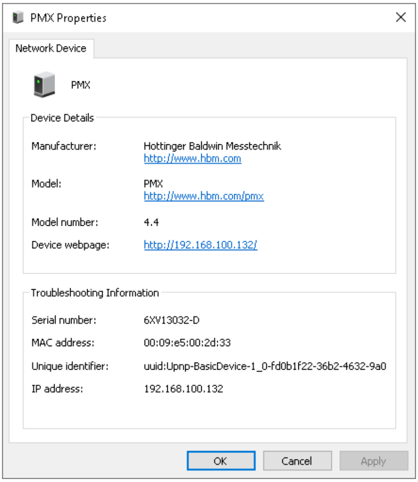
- Open your PC's network environment.
  - Under **Other devices** you will find one or more PMX devices.
- Double-click on ***pmx***.



Tip

Right-click PMX, then under **Properties** *you can find details such as the device website, PMX serial number, IP address, etc.*

1) Under Windows 7, **Media Streaming** must be turned on (**Control Panel Network and Internet Network and Sharing Center > MEDIA STREAMING**).



If there are multiple PMX devices in the network, this dialog will appear:

- Check the desired PMX.
- Click **Connect**.

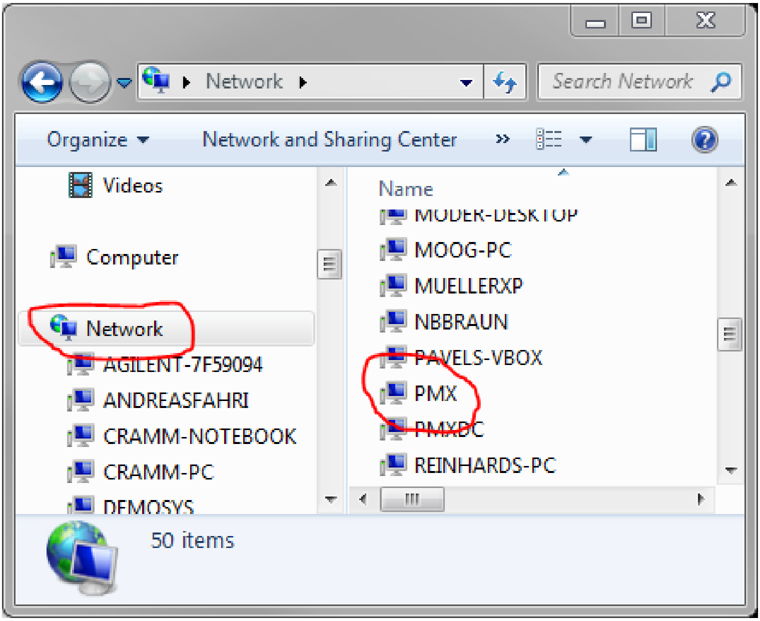
This takes you to the overview:

There you can see the current measured values and - at the appropriate user level - set up the PMX.

Option B:

Connection via the NetBIOS (name) under Windows

- Open your PC's network environment.
  - **PMX** appears in the network environment.



- Open your web browser.
- Type ***pmx/*** in the address bar.

Assigning names if there are several PMX devices in the network:

- First device: PMX
- Second device: PMX-2
- Third device: PMX-3 etc.

There you can see the current measured values and - at the appropriate user level - set up the PMX.

Important

If no DHCP server is found, and no fixed (static) IP address is set in the device, PMX automatically uses an IP address according to RFC5735 (APIPA, 169.254.xxx.xxx). If a static IP address has been set, two IP addresses are available: the set static address and an IO address from the automatic IP range.

Option C:

Connection via Zeroconf/Bonjour

- Download and install Apple's *"Bonjour Print Services"* software: http://support.apple.com/kb/DL999?viewlocale=de_DE&locale=de_DE

  If Apple software has already been installed, *Bonjour* will usually already be on the computer.
- Enter ***pmx.local.*** in your web browser's address bar.

You can then see the current measured values and - at the appropriate user level - set up the PMX.

Important

You can permanently change the device name (factory default "pmx") and the network settings (DHCP, IP address, subnet mask, gateway) in the **Network** *menu item.*

11.3.1 Making network settings via a USB flash drive

If you cannot find the PMX in the network, you can also make the network settings using a USB flash drive.

- Create a text file with the name "pmx.conf" in the flash drive's root directory.
- Enter text similar to that in the following examples. Change device names and network settings as appropriate.

  Example 1:
  *This pmx.conf file sets the device name to "pmx_new_name", and switches the PMX to DHCP mode.

```
<pmx type="set">
 <hostname>pmx_new_name</hostname>
 <network>
 <dhcp>true</dhcp>
 </network>
</pmx>
```

  Example 2:
  This pmx.conf file sets the device name to "pmx", and specifies a fixed IP address:

```
<pmx type="set">
 <hostname>pmx</hostname>
```

<network>
<ipaddress>192.168.1.2</ipaddress>
<broadcast>192.168.255.255</broadcast>
<netmask>255.255.0.0</netmask>
<gateway>192.168.169.254</gateway>
<dhcp>false</dhcp>
</network>
</pmx>

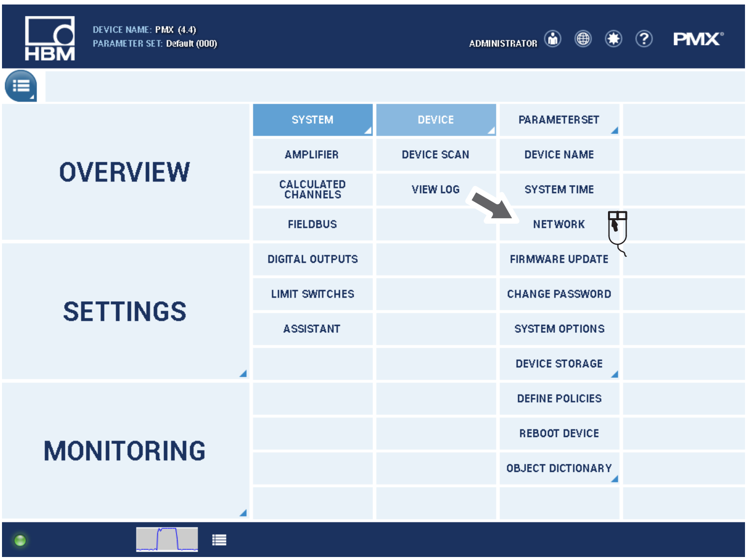
4. Plug the flash drive into the USB port on the PMX while the PMX is running normally.

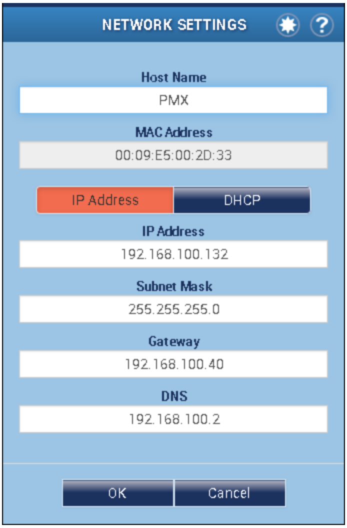
   The settings will change immediately, but will not be immediately apparent in the other network devices. We recommend restarting the PMX by disconnecting the power supply. The PMX can then be found in the network under the new settings.

Important

The flash drive converts each PMX device as soon as you plug it in! So you should delete the file, rename it, or move it to a subfolder.

11.3.2 Changing network settings in a web browser




NETWORK SETTINGS
Host Name
PMX
MAC Address
00:09:E5:00:2D:33
IP Address
DHCP
IP Address
192.168.100.132
Subnet Mask
255.255.255.0
Gateway
192.168.100.40
DNS
192.168.100.2
OK
Cancel

11.4 Display and control options

Important

For a detailed description of PMX operation see the PMX online help. Download the current firmware from the HBM PMX website: https://www.hbm.com/de/2981/pmx-modular-measuring-amplifier-system-for-the-iot/.

The overview shows the plug-in cards (amplifiers) in the PMX, with the current measured values, the status of the digital inputs/outputs and bus systems (if present), as well as the calculated channels.

- Touch or click on one of the icons, or one of the places where the cursor turns into a hand, to access the dialog to change the setting.

- Use the menu icon to open the Settings menu, from which you can access all dialogs by way of the menu table structure. All the menu items with a triangle in the bottom right corner have additional submenus. As soon as you have selected a menu item, the menu path is displayed next to the symbol for the setting menu.

Important

If you change a setting, the floppy disk icon is displayed at the bottom right to show that the setting is initially only stored in the RAM. Touch or click on this symbol to save the setting power failsafe in flash EPROM.

Symbols and displays used

The icon opens the user administration function.

The icon opens the language selection function, where you can switch to one of the displayed languages.

The icon opens the favorites list. You can add any view to the favorites list.

The icon opens the help.

The LED at the bottom left in the status line shows the status of the PMX:

The green LED indicates that everything is in order.

The yellow LED indicates that there is an error in one or more of the channels, but the PMX keeps on working.

The red LED indicates that there is a critical error. No more measured values can be acquired or processed.
It is also possible that the network connection to the PMX may have been lost. In this case the PMX can continue measuring.

The small graphic in the status bar at the bottom (A, see *Fig. 11.1*) shows the PMX's capacity utilization (0 … 100%). You can estimate in this way whether, at the selected sampling rate,

- the defined calculations can be performed;
- the number of calculations needs to be reduced;
- one of the sampling rates needs to be lowered;
- a CODESYS application has overloaded the capacity of the CPU.

Fig. 11.1 The overview page of the PMX web browser with the device and signal realizations of the connected PMX.

Measured value realization

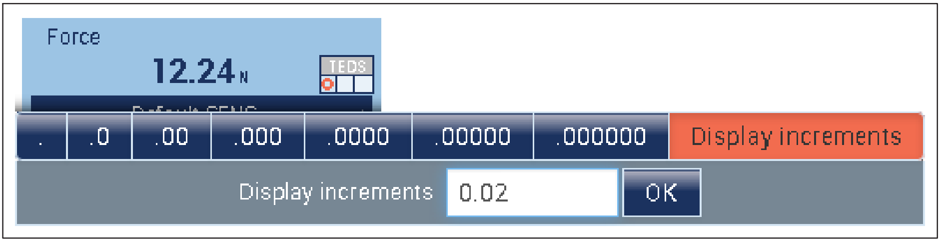
You can customize the display of the measured values and data from the calculation channels of the PMX individually for each measurement channel. This affects the number of decimal places and the increment of the digital scale. This allows you to adjust the display to the requirements of the application.

Measurement display: Click on the required measurement display in the amplifier settings.

Calculated channels: Option in the **Decimal Places** dropdown menu.

These settings only affect the display value in the PMX web serve, and do not relate to the values in the PMX or the interface output.

Example: Setting with 2 decimal places and an increment of 2 digits, i.e. 0.02, 0.04, 0.06, 0.08 ...

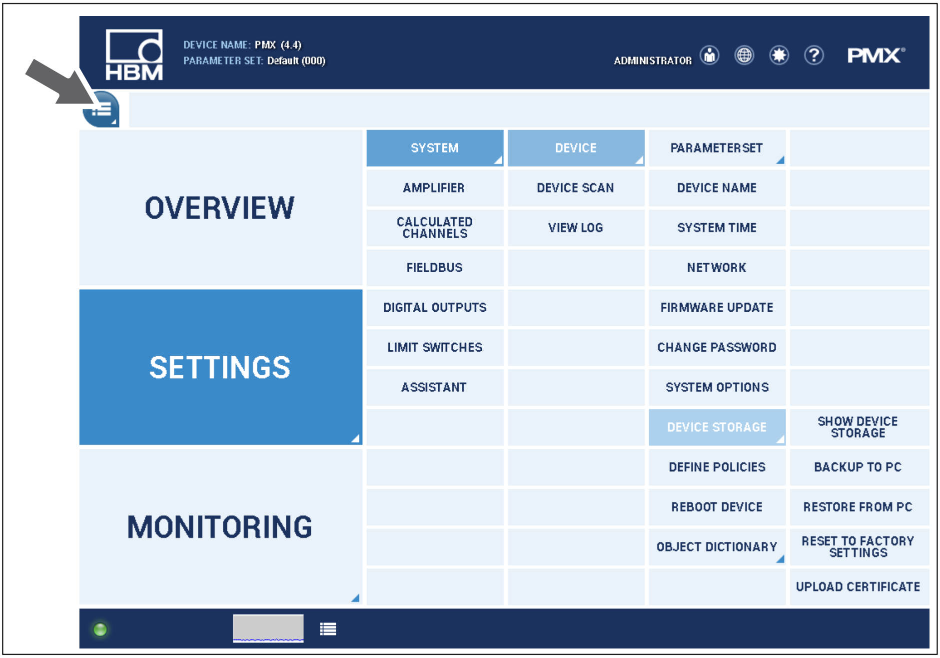

Force
12.24 N
TEDS
. | .0 | .00 | .000 | .0000 | .00000 | .000000
Display increments
Display increments
0.02
OK

11.5 PMX web server menu structure

11.5.1 Overview of all device settings

Each menu item has an online help that you can access by clicking on this icon.

Clicking on this icon opens the menu page. Choose **SETTINGS** to parameterize the PMX. Select the required menu item by clicking on it.



11.5.2 Restoring the factory settings

Loading the factory settings deletes the following settings:

- All channel and amplifier settings (measurement channels and calculated channels, e.g. min./max. values)
- All device settings (e.g. parameter sets).

The following are not deleted:

- The network settings
- The passwords for the different user levels (OPERATOR, MAINTENANCE, ADMINISTRATOR)
- CODESYS applications and CODESYS web visualizations (for updates installed on firmware 1.46 or higher).

11.6 PMX startup behavior

Important

PMX initialization takes a few seconds. A self-test of all the modules is run during this time. All the LEDs flash to indicate this status. After the self-test has run, read off the status of each component from the corresponding status LED - see sections 8.2.3 to 8.2.5 (starting on page 50).

- When the PMX is switched on, the digital and analog outputs are set to 0 V.
- When the system boots up, the analog outputs remain set to 0 V.
- After booting up, the configured and valid outputs are set to the specified value between -10 and + 10 V.
- Invalid (unconfigured) outputs go to 0 V (Safe Value).
  If an output becomes invalid during operation, it will also be set to 0 V.
- You can also set a different value for the Save Value. The factory setting is 0 V.

11.7 Operating behavior of the PMX

According to its application specifications, the PMX is suitable for test and measuring tasks with integrated monitoring and control tasks. However, it must not be used in areas where malfunctions could result in personal injury or damage to property.

A number of measures have been implemented in PMX to enhance the operational safety of the plants in which it is used.

Setup mode

A target value (test signal) can be simulated for every measurement signal without an actual measured value needing to be received from the plant. This allows you to test downstream functions and components. This also works for the analog outputs (±10 V).

Ongoing operation/measuring mode

- Digital outputs: Here you can signal in detail the device or measured value status via hardware outputs, fieldbus or Ethernet (PC).
- Thresholds: Normally the status of the measured values is taken into consideration in the analysis, meaning there is no switching process in the event of an invalid measured value. You can suppress this with the ***Ignore Meas. Status*** option. This means the limit conditions will then be analyzed even if the measured value is incorrect (invalid).

A measured value is invalid and identified accordingly if

- the input range of the measuring amplifier was exceeded;
- a calibration is running;
- the working standard calibration is incorrect;
- the setting saved in TEDS cannot be implemented in the ***Use TEDS if Available*** setting (e.g. wrong transducer type, no measuring range, invalid filter value, etc.);
- the TEDS content cannot be read correctly or there is no TEDS even though ***TEDS Required*** is set.

Digital inputs

These must be connected against +Ub (PLC logic). An open input is detected as Low by the internal pull-down resistor.

11.8 Signal delays

Typical signal delays of the individual PMX hardware and software components.

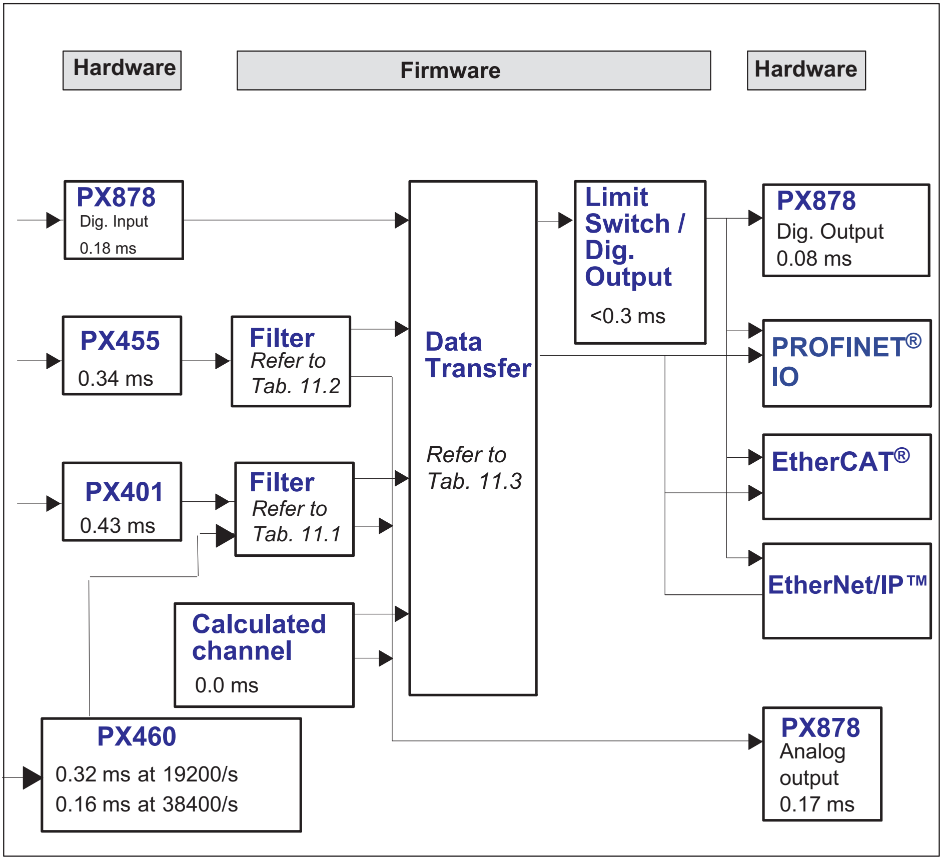


Fig. 11.2 Signal delays

Filter group delay

Cut-off frequency fc (in Hz, -3dB)	Phase delay (in ms)	
	Bessel	Butterworth
6000[1)]	0.07	0.94
5000[1)]	0.08	0.12
3000	0.10	0.14
2000	0.20	0.28
1000	0.42	0.61
500	0.86	1.23
200	2.00	3.10
100	4.15	6.17
50	8.45	12.5
20	21.4	30.7
10	39	47
5	74	91
2	174	216
1	340	430
0.5	680	840
0.2	1680	2090
0.1	3360	4200

[1)] *5000/6000 Hz cut-off frequency for PX460 only*

*Tab. 11.1 Phase delays for **PX401** and **PX460***

Cut-off frequency fc (in Hz, -3dB)	Phase delay (in ms)	
	Bessel	Butterworth
2000	0.16	0.23
1000	0.42	0.60
500	0.85	1.24
200	2.00	3.10
100	4.15	6.17
50	8.45	12.5
20	21.4	30.7
10	39	47
5	74	91
2	174	216
1	340	430
0.5	680	840
0.2	1680	2090
0.1	3360	4200

*Tab. 11.2 Phase delays for **PX455***

Data Transfer Rate (in Hz)	Phase delay in ms		
	Minimum	Typical	Maximum
1200	0.1	0.52	0.93
2400 (factory setting)	0.1	0.31	0.52
4800	0.1	0.21	0.31
9600	0.1	0.16	0.21

Tab. 11.3 Data phase delays

Example

Signal delay of a sensor signal through a Bessel filter to the analog output:

Signal path PX455 –> 2 kHz Bessel–> PX878
(0.34*)) + 0.16 (Tab. 11.3) + 0.17*)) ms = 0.67 ms

*) See diagram on *page 152*

Fieldbus

Delay time until signal appears in cyclic data frame.

Protocol	Data Copy Rate (in Hz)	Phase delay in ms	
		Typical	Maximum
PROFINET® IO	1200 (standard and max.)	1.8 + frame_cycle /2	2.4 + frame_cycle
EtherCAT®	2400 (standard) 4800 9600 (maximum)[1)]	1.0 + frame_cycle /2	1.5 + frame_cycle
EtherNet/IP™	1200 (standard and max.)	1.8 + frame_cycle /2	2.4 + frame_cycle

[1)] The EtherCAT® data copy rate has only minor effects on the signal delay. Between copy rates of 2.4 and 9.6 kHz, the advantage theoretically amounts to 0.16 ms, which is significantly smaller than the statistical spread.

Tab. 11.4 Fieldbus phase delays

"Data Copy Rate" is the time in which the data is copied to the fieldbus module in slot 0. frame_cycle is the rate of the cyclic data frame set in the bus configuration tool.

Example

Signal delay of a sensor signal via the EtherCAT® fieldbus:

Signal path: PX455 –> 2 kHz Bessel–> Data transfer @2.4 kHz –> EtherCAT @2.4 kHz PX01EC
(0.34[*)] + 0.16 (table 1.2) + 0.31 + 1.2) ms = 2.0 ms (average signal delay from input terminal to EtherCAT® fieldbus)

[*)] See diagram on page 152

11.9 Fieldbus integration

Important

Make sure to use the correct device description file (see also section 16.1, "Device description file", page 259).

11.9.1 PROFINET® IO connection

- Connect the PMX device(s) and PROFINET® master via Ethernet cable (Cat 5) (following the topology).
- When using the PROFINET® IRT protocol, observe the sequences of the PMX-RJ45 sockets: Port 1 (bottom), Port 2 (top).
- For PROFINET®-IRT, turn on the IRT protocol in the PLC configuration software, and specify the cable lengths and the ports for the wiring (see also *section 8.2.4, "Fieldbus LEDs", page 52*).

Notice

The bus does not need any termination resistors, as active nodes are involved.

Use the device description file (GSDML) to configure the PMX in the master. It is available as a download from the HBM PMX website, and you can generate it in the PMX itself and download it from there.

Procedure

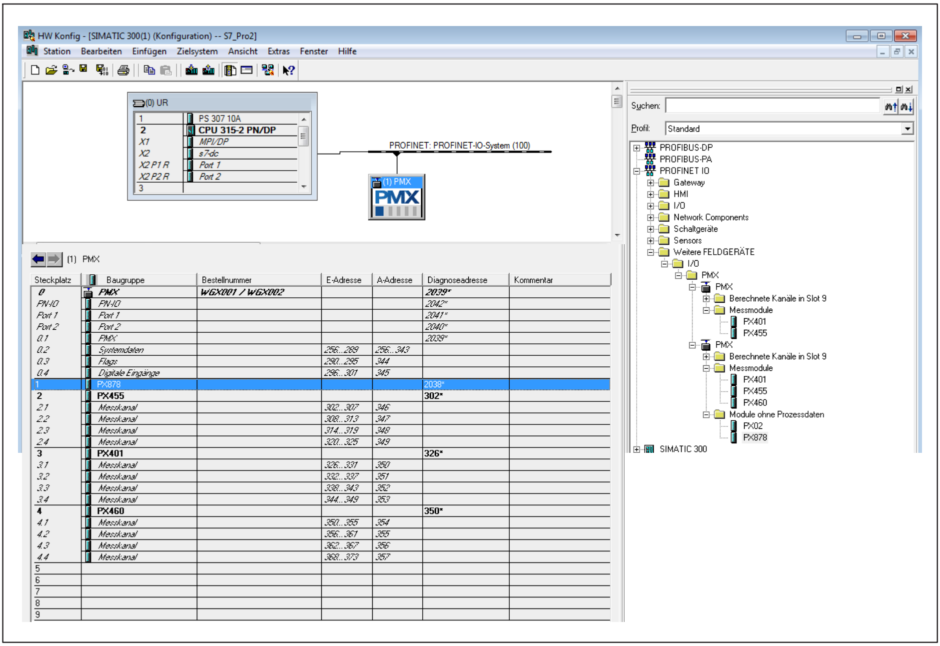
- Change the user level to ADMINISTRATOR.
- Go to **SETTINGS -> FIELDBUS**.
- For ***No. Transm. Calc. Channels*** specify how many channels will be transferred.
- Click on ***Create GSDML File***.

The file is saved in the "PROFINET" directory.

You can also access the directory with any browser. Enter the address ***http://<PMX-Name>/public/PROFINET/*** to open the "PROFINET" directory. For <PMX name> specify the network name or the IP address of the PMX.

The PROFINET® system is configured using your PROFINET® master supplier's engineering tool.

Example with a SIEMENS PLC under STEP7 with the SIMATIC Manager or TiA Portal



11.9.2 EtherCAT® connection

- Connect the PMX device(s) and EtherCAT® master via Ethernet cable (Cat 5). Follow the topology: IN (bottom) / OUT (top) of RJ45 sockets on PX01EC.

Notice

The bus does not need any termination resistors, as active nodes are involved.

Use the device description file (ESI) to configure the PMX in the master. It is available as a download from the HBM PMX website, and you can generate it in the PMX itself and download it from there.

Procedure

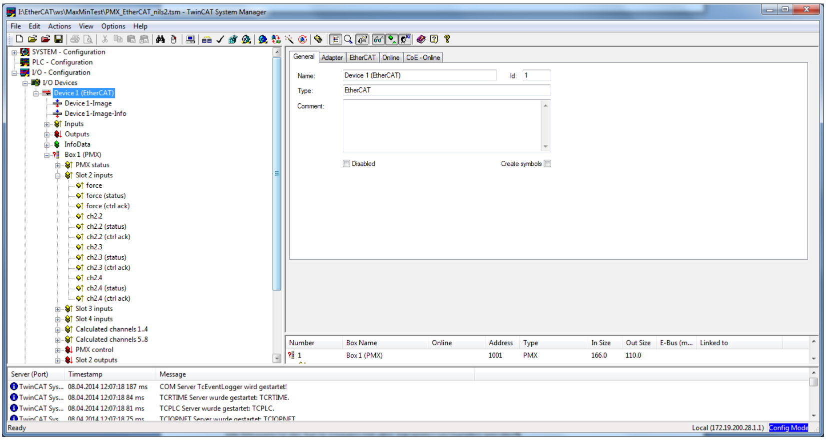
- Change the user level to ADMINISTRATOR.
- Go to **SETTINGS -> FIELDBUS**.
- For ***No. Transm. Calc. Channels*** specify how many channels will be transferred.
- Click on ***Create ESI File*** (EtherCAT®).

The file is saved in the "EtherCAT" directory.

You can also access the directory with any browser. Enter the address ***http://<PMX-Name>/public/EtherCAT/*** to open the "EtherCAT" directory. For <PMX name> specify the network name or the IP address of the PMX.

The EtherCAT® system is configured using your EtherCAT® master supplier's engineering tool.

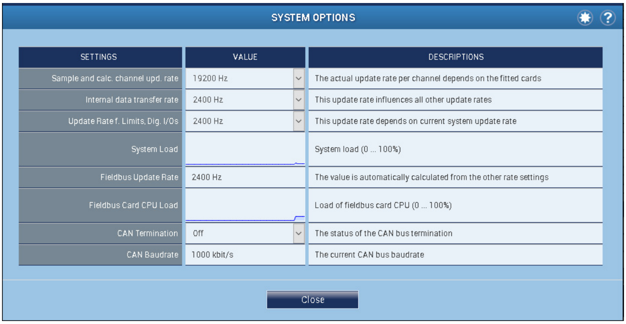
Example with a Beckhoff PLC with the TwinCAT System Manager



11.9.3 Setting the fieldbus update rate

- Change the user level to ADMINISTRATOR.
- Go to **SETTINGS -> SYSTEM -> DEVICE -> SYSTEM OPTIONS**.
- Select the **Internal data transfer rate**. The fieldbus update rate will follow this value up to the fieldbus-specific maximum.

The change is effective immediately. Click on the floppy disk icon at the bottom right to save permanently.



11.9.4 EtherNet/IP™ connection

- Connect the PMX device(s) and your EtherNet/IP™ scanner via Ethernet cable (Cat 5).

  Both ports, P1 and P2, have the same IP and MAC addresses.

Notice

The bus does not need any termination resistors, as active nodes are involved.

Use the device description file (EDS) to configure the PMX in the scanner. It is available as a download from the HBM PMX website, and you can generate it in the PMX itself and download it from there.

Procedure

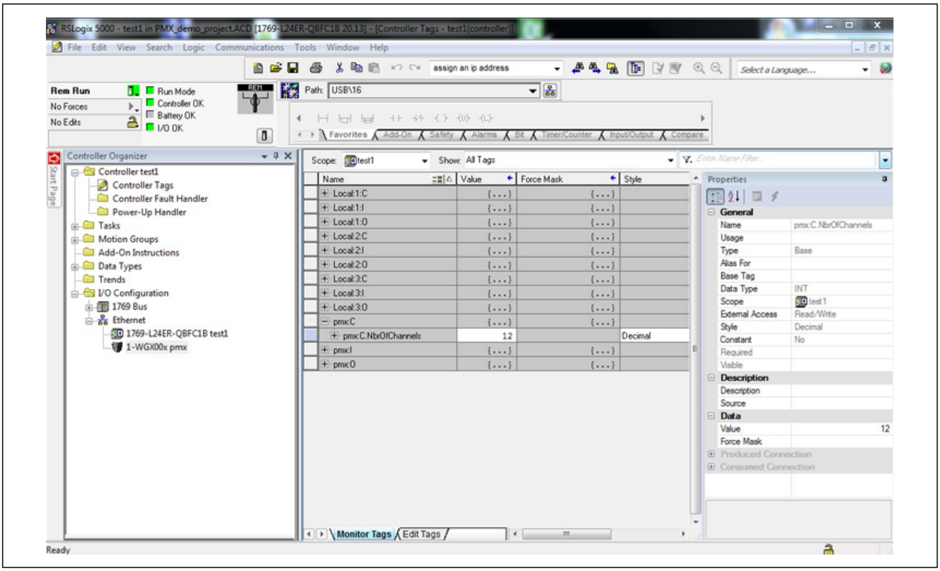
- Change the user level to ADMINISTRATOR.
- Go to **SETTINGS -> FIELDBUS**.
- For ***No. Transm. Calc. Channels*** specify how many channels will be transferred.
- Click on ***Create EDS File***.

The file is saved in the "EtherNet_IP" directory.

You can also access the directory with any browser. Enter the address ***http://<PMX-Name>/public/EtherNet_IP/*** to open the "EtherNet_IP" directory. For <PMX name> specify the network name or the IP address of the PMX.

The **EtherNet/IP™** system is configured using your EtherNet/IP™ scanner supplier's engineering tool.

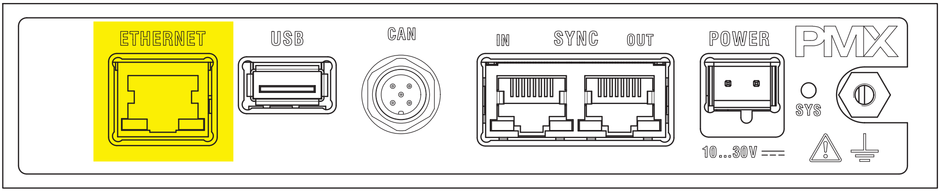
Example with Allan-Bradley PLC ControlLogix and LogixStudio



12 QUICK START

12.1 Preparing the measurement system

1. Connect the PMX to your PC via the Ethernet socket - see *section 11.3, page 136*.



Cable: Standard Ethernet cable (Cat 5)

2. Connect your transducers to the measurement cards (plug terminals) - see *sections 8.4 to 8.7 (starting on page 62)*.

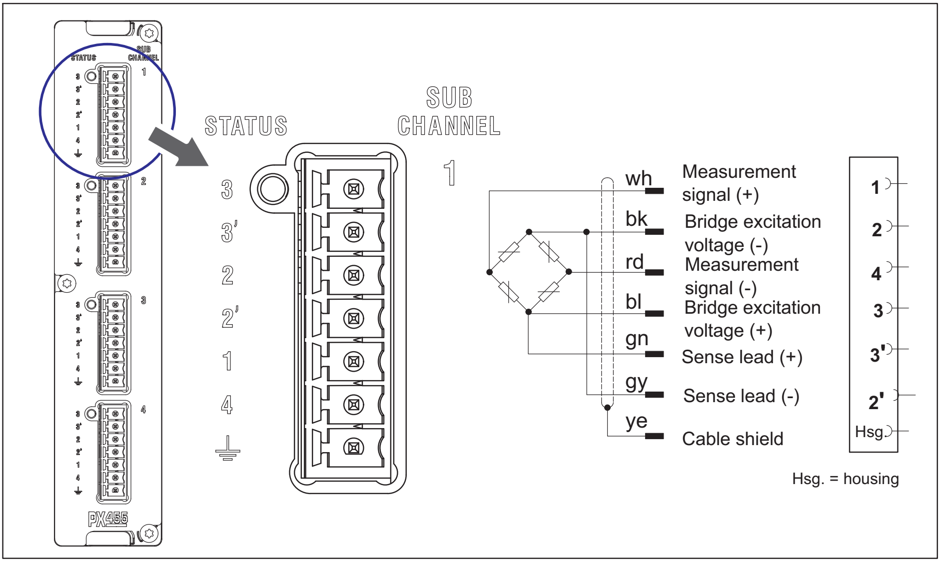


Fig. 12.1 Example: Force transducer / load cell on PX455

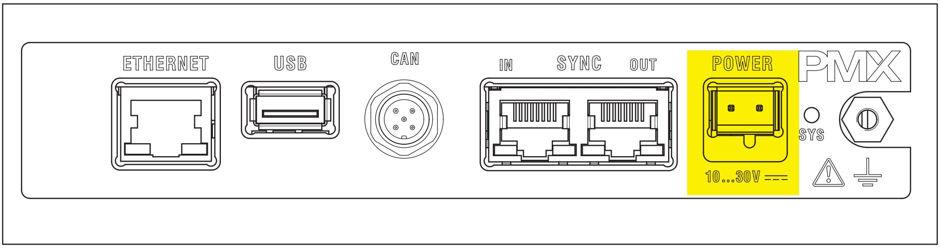
Notice

The transducers can also be connected if you have already connected the power supply.

3. Connect the power supply (10 … 30 V_{DC}) - see *section 8.3, "Supply voltage", page 61*.

The power supply must be at least 15 W.

The PMX boots, and then displays its system status (see *section 8.2.3*). The system LED must light up green. This process takes a few seconds.



The PMX is set to DHCP (automatic address assignment) at the factory. Set your PC to DHCP as well. The IP addresses will then be set automatically. This process takes several tens of seconds.

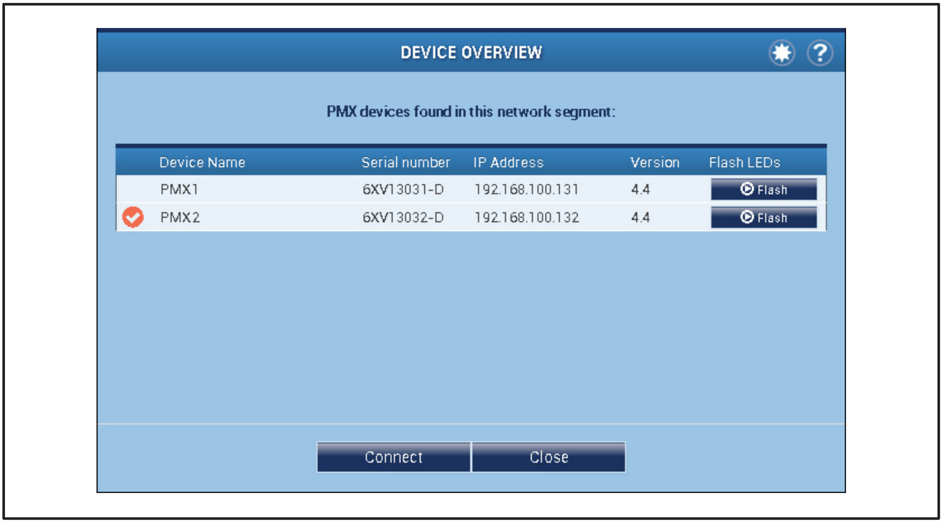
Call the PMX web server by entering ***PMX/*** in the address bar of your browser.

The PMX web server appears, showing the start screen (overview).

The PMX system is now ready for measurement, and you can see live measured values.

- Click on the globe icon  to switch to another PMX web server language.

If there are multiple PMX devices in the network, this selection box will appear before the overview:



- Check the desired PMX.
- Confirm with **Connect**.

The flash function allows the device to be identified by flashing all the device LEDs.

4. Configuring the PMX with a web browser

- Click on the user icon to go to the Maintenance or Administrator level. Depending on authorization, you can make the following settings:
  - Assign sensors
  - Assign units
  - Set filters
  - Monitor maximum and minimum values
  - Monitor limit values (Thresholds)
  - Set up virtual (calculated) channels
  - Configure digital and analog inputs/outputs
  - Create and administer parameter sets

Notice

Click on the floppy disk icon to save changed settings power failsafe to the device.

Confirmation prompt

To get additional help, click the Help icon .

The web server help opens with information relevant to the displayed page.

12.2 Typical operating sequence (measurement example)

The easiest way to configure the PMX measuring amplifier and its measurement channels is via the PMX web browser. The sensors, Ethernet cables and power supply must be connected correctly (see *sections 8.3 and 8.4 starting on page 61 as well as section 11.3, page 136*).

The overview shows the PMX with all its measurement cards and signals, as well as all device information.

 Switch to the **ADMINISTRATOR** user level (might be password-protected), then click on the **Settings -> Amplifier** menu icon. 

Alternatively, you can also switch directly to the relevant settings menu by clicking on the desired channel or function (e.g. a limit value). This always requires authorization for the respective user level.

Make the appropriate sensor and signal settings here for each slot and channel.

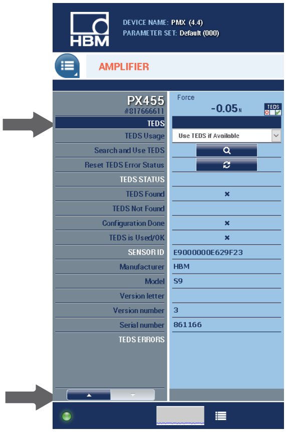
Click on the slot number to select a different measurement card:

Orange = selected measurement card; blue = measurement cards in the PMX; gray = empty module slot.

Example: Configuring with force transducer

In the example above, a PX455 is in slot 2 and a force transducer S2M 1000N (strain gage full bridge) is connected to the first channel.

- The PMX amplifier channel is set to the ***full bridge*** sensor type, with a measuring range of ***4 mV/V***.
- The scaling (**Characteristic**) is set to ***1000 N*** with a sensor sensitivity of ***2.010270 mV/V***. If the sensor has TEDS, the channel is automatically parameterized by default. Otherwise, activate the TEDS settings on the second amplifier settings page.



- The filter type here is set to ***Bessel***, with a cut-off frequency of ***5 Hz***.
- The data is now changed in the PMX, and the floppy disk icon appears in the status bar.
- To save the setting power failsafe in the PMX, click on the icon and confirm the save prompt.

Example: Configuring the PMX with a strain transducer

In the example above, a PX455 is in slot 2 and a strain transducer SLB700A is connected to the second channel.

- The PMX amplifier channel is set to the ***full bridge*** sensor type, with a measuring range of ***4 mV/V***.
- The scaling (**Characteristic**) is set to ***500 µm/m*** with a sensor sensitivity of ***1.5 mV/V***.
- The filter type here is set to ***Bessel***, with a cut-off frequency of ***5 Hz***.
- The data is now changed in the PMX, and the floppy disk icon appears in the status bar.
- To save the setting power failsafe in the PMX, click on the icon and confirm the save prompt.

Example: Configuring the PMX with a displacement transducer

In the example above, a PX455 is in slot 2 and a displacement transducer WI 10 mm is connected to the third channel.

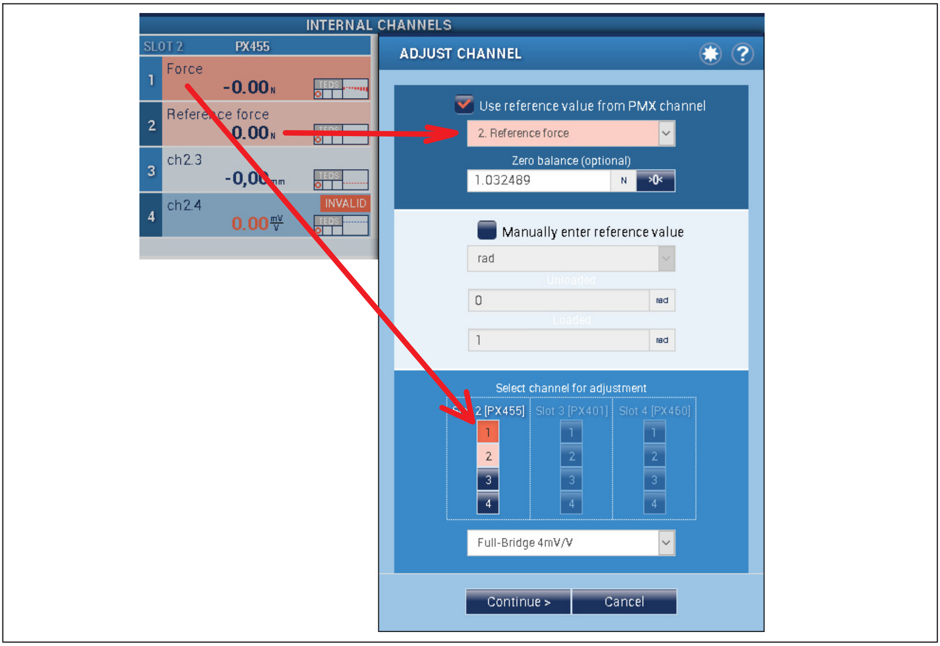
- The PMX amplifier channel is set to the ***inductive half bridge*** sensor type, with a measuring range of ***100 mV/V***.
- The scaling (**Characteristic**) is set to ***10 mm*** with a sensor sensitivity of ***80 mV/V***.
- The filter type here is set to ***Bessel***, with a cut-off frequency of ***20 Hz***.
- The data is now changed in the PMX, and the floppy disk icon appears in the status bar.
- To save the setting power failsafe in the PMX, click on the icon and confirm the save prompt.


DEVICE NAME: PMX (4.4)
PARAMETER SET: Default (000)
ADMINISTRATOR
PMX
AMPLIFIER
PX455
#817666611
SENSOR
SENSOR TYPE
PHYSICAL UNIT
CHARACTERISTICS
1. Point Electrical
1. Point Physical
2. Point Electrical
2. Point Physical
SIGNAL CONDITIONING
Zero Value
Zero Target Value
CONTROL FUNCTIONS
Zero by
Clear Zero by
Test Signal
DATA ACQUISITION
Channel Name
Type
Cutoff Frequency (-3dB)
Force
Strain
Displacement
ch2.4
INVALID
Default SENS
Default DAQ
Full-Bridge 4mV/V
Inductive Half-Bridge 100m
Bessel
5 Hz
20 Hz
2000 Hz
SLOT:

12.3 Adjustment Assistant

Firmware version 2.04 and later provides an Adjustment Assistant to help adjust non-calibrated sensors. It provides a simple, menu-guided way of calibrating up to four sensors. This is useful, for example, where the sensors are not installed in a machine until on site and are operated in force shunt mode. You can use a measurement channel in the PMX or an external reference measurement chain as a reference sensor.

1. Selecting the adjustment channel and the reference channel



2. Adjusting the measuring chain in the unloaded and loaded state

Tip

To improve accuracy, you should repeat the adjustment process multiple times.

3. After the adjustment processes, the channel settings are applied by clicking ***Apply***. Clicking ***Cancel*** restores the initial state.

12.4 Updating firmware (PMX web server)

To operate current versions of the PMX devices, you need a terminal (such as a PC or tablet with a mouse) running a current web browser (Internet Explorer version > 9.0, Firefox or Chrome) and with a screen resolution of 1024 x 768.

Windows XP must be installed on the PC as a minimum.

A new version of the web server is a component part of the PMX firmware, and is installed together with it in a firmware update (see *chapter 24, "Firmware update", page 424*).

Run the firmware update in the PMX web browser using the **Settings -> System -> Device -> Firmware Update** menu.

For more support, see the web browser online Help.

Tip

You can download the latest firmware from: https://www.hbm.com/de/2981/pmx-modular-measuring-amplifier-system-for-the-iot/

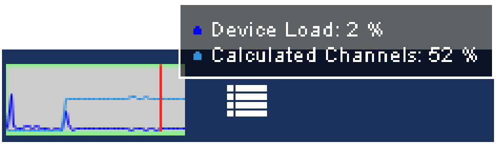
13 INTERNAL CALCULATION CHANNELS

The PMX provides a total of 32 internal calculation channels for free use. You can distribute the more than 30 available calculations made in the PMX's DSP (Digital Signal Processor) to those channels as you like, and also use calculations multiple times as well as in other calculations. Calculation channels are calculated at the same speed as measurement channels. Like them, they can be processed further via analog outputs, fieldbus or Ethernet (PC software/ API).

Calculations are initially performed internally only. You can use the PX878 plug-in card, for example, to send the result to external devices as an analog or digital signal. Calculations can be used in other calculations if they are arranged above the calculation in which they will be used in the list. Some constants that are helpful for calculations, such as invalid, 0 (constant 0) or π (pi), are always available. Create additional constants yourself.

Important

Make sure that the CPU load caused by the calculation channels is below 100%, otherwise individual values will be lost. Reduce the output rate of the calculation channels as necessary to do so.
The CPU load of the calculation channels is displayed in the status bar of the PMX web browser and in the **Settings -> System -> Device -> System Options** *menu.*
The information is additionally available in the PMX system status, and you can output it via fieldbus, Ethernet and digital output.



When using the object dictionary, adding, changing or deleting calculation channels dynamically changes the object list. Then recreate the header files and adapt the programming via fieldbus or PC control.

13.1 Calculation rate

The following applies to all modules , unless otherwise specified:

Calculation rate	Equivalent to update rate (default setting 19200/s)
Value range of floating point values	Simple floating point resolution according to IEEE 754 range approx. ±3.4*10^{38}

Important

If one of the source signals is invalid, the output signal will be invalid as well. If the result is outside the output range, NaN (not a number) appears in the display and $\pm 3.4 * 10^{38}$ *(invalid) is outputted. This is displayed in the PMX browser and in the measurement status.*

13.2 Descriptions of calculations

13.2.1 Scaling

13.2.2 Two-point scaling

Function	Linear scaling of a signal Output = m * In + b where m = (y2 − y1) / (x2 - x1) and b = y2 − m * x2
Source	Input
Output	Result channel
Parameter(s)	Two interpolation points (x1 | y1) and (x2 | y2) where x = input values and y = output values
Default setting	x1 = y1 = 0; x2 = y2 = 1 (corresponds to m = 1 and b = 0)
Exception handling	In case of out-of-range, e.g. if x1 = x2, NaN is outputted.

13.2.3 Characteristic table (21 interpolation points)

Function	Linearize non-linear characteristic
Source	Input
Output	Result channel
Parameter(s)	Number of interpolation points used (2 … 21), (x0 | y0)…(x20 | y20)
Default setting	Number of points: 2 x0 | y0: -1000 | -1000 x1 | y1: 1000 | 1000
Exception handling	If the source signal is invalid, the output signal will also be invalid.

The first y value is outputted below the first x value, the last y value above the last x value. You can also enter jumps (two equal x values for two different y values). Then the first y value will be outputted under that x value, above the second y value.

The function is suitable for limiting values or forming amounts for example.

13.2.4 Polynomial 4th order

The input signal is converted by the specified polynomial. You can specify up to five coefficients. To use different polynomials for positive and negative input values, enter 2 for the ***number of coefficient sets***. If you use two coefficient sets, the coefficient set with the b coefficients is used for negative input values.

Function	Polynomial 4th order Result channel = $a0 + a1*x + a2*x^2 + a3*x^3 + a4*x^4$ (if x >= 0 or only 1 set of coefficients) Result channel = $b0 + b1*x + b2*x^2 + b3*x^3 + b4*x^4$ (if x < 0)
Source	Input
Output	Result channel
Parameter(s)	Number of coefficient sets Set A: a0, a1, a2, a3, a4 Set B: b0, b1, b2, b3, b4 (only with 2 sets of coefficients)
Default setting	Input: 0 Number of coefficient sets: 1 a0 … a4, b0 … b4: 0
Exception handling	If the source signal is invalid, the output signal will also be invalid. If the output signal is outside the display range ($\pm 3.4*10^{38}$), NaN is outputted and the signal becomes invalid.

13.2.5 Tare

Function	Quick zeroing or taring, controlled via digital input.
Sources	Input Tare target value: Value to be outputted at the output after zeroing/taring. Tare with (level-controlled): At High level the output is set to the tare target value. Reset (level-controlled): At High level the zeroing/taring is reversed. Reset overrides tare trigger.
Output	Input - tare value + tare target value

Parameter(s)	–
Default setting	Input: 0 Tare target value: 0 Tare with: None Reset: None
Exception handling	If the source signal is invalid, the output signal will also be invalid. If the source signal is invalid, no taring is performed. The tare target value is not checked for validity (it is usually a constant).

Quick tare is an alternative to zeroing in the **Amplifier** menu:

	Zero (menu)	Tare (calculation)
Availability	Always available in Amplifier settings	Only as calculated channel
Offset (difference input - output)	Known and selectable by user	Unknown, not selectable
Effect	Acts directly on sensor on original measurement signal	Original measuring signal is not influenced
Execution time	Fieldbus approx. 30 ms[1)] Digital input approx. 12 ms[1)]	approx. 0.2 ms[1)] at an update rate of 19200/s

1) Recommended values for default settings

13.2.6 6x6 matrix

Function	out1 = a11*in1 + a12*in2 + a13*in3 + a14*in4 + a15*in5 + a16*in6 out2 = a21*in1 + a22*in2 + a23*in3 + a24*in4 + a25*in5 + a26*in6 out3 = a31*in1 + a32*in2 + a33*in3 + a34*in4 + a35*in5 + a36*in6 out4 = a41*in1 + a42*in2 + a43*in3 + a44*in4 + a45*in5 + a46*in6 out5 = a51*in1 + a52*in2 + a53*in3 + a54*in4 + a55*in5 + a56*in6 out6 = a61*in1 + a62*in2 + a63*in3 + a64*in4 + a65*in5 + a66*in6
Sources	6 inputs
Outputs	6 outputs

Parameter(s)	a11, a12, a13, a14, a15, a16 a21, a22, a23, a24, a25, a26 a31, a32, a33, a34, a35, a36 a41, a42, a43, a44, a45, a46 a51, a52, a53, a54, a55, a56 a61, a62, a53, a64, a65, a66
Default setting	Coefficients as a matrix: **1** 0 0 0 0 0 0 **1** 0 0 0 0 0 0 **1** 0 0 0 0 0 0 **1** 0 0 0 0 0 0 **1** 0 0 0 0 0 0 **1**
Exception handling	If one of the source signals is invalid, all the output signals will become invalid as well.

You can use the calculation to compensate for the crosstalk of a multi-component force transducer in the output signal. Enter only the coefficients (axx) in the dialog table. Use 0 for unused coefficients. For example, if you do not need Input 6, set every ax6 (a16 to a66) to 0.

13.2.7 Strain gage stress analysis

The function uses the measured values from the measuring grids of a 0°/45°/90° or 0°/60°/120° strain gage rosette to calculate the principal normal stress 1 (σ 1), the principal normal stress 2 (σ 2), the angle (φ), the von Mises reference stress (σ V) and the shear stress (τ max), which lies at 45° to the direction of the principal normal stress.

Compensation for temperature effects

There are three ways to compensate for the temperature effects of your material:

1. You can use a second (passive) strain gage rosette that is connected as a half bridge circuit with the active strain gage and only acquires the thermal expansion of the material.
2. You can use a single measuring grid (Temperature compensation SG) that is measured in another measurement channel.
3. You measure the temperature at the measuring point of the rosette with a temperature sensor, and calculate the apparent strain using the temperature response coefficients specified on the strain gage package.

   To do this, generate a polynomial calculation with the polynomial coefficients specified on the package and the temperature value as the input signal; see also *section 13.2.4, "Polynomial 4th order", page 176*. Then you can use this channel as the input signal for the Apparent strain.

Enter ***Constant 0*** for ***Temperature compensation SG*** or ***Apparent strain*** if you are not using the method in question.

Function	Stress analysis in biaxial stress state with strain gage rosette with three measuring grids (0° / 45° / 90° or 0° / 60° / 120°). Zero The channels with the measuring grids a, b and c do not have to be zeroed. The function block provides a zeroing input that clears all outputs. The zeroing result is saved together with the other settings and restored after restarting the device.
Sources	Measuring grid a (0°, angular reference) in μm/m Measuring grid b (45° / 60°) in μm/m Measuring grid c (90° / 120°) in μm/m Temperature compensation SG in μm/m Apparent strain (from polynomial channel) in μm/m Reset by (level-controlled): Zeros all outputs except angle φ.
Outputs	σ 1: Principal normal stress 1 (unit as unit of modulus of elasticity) σ 2: Principal normal stress 2 (unit as unit of modulus of elasticity) φ: Angle of σ 1 based on the axis of the measuring grid a in degrees; range 0 … 180° τ max: Maximum shear stress (unit as unit of modulus of elasticity) σ V: von Mises stress (unit as unit of modulus of elasticity)
Parameter(s)	SG type: 0/45/90° or 0/60/120° Modulus of elasticity; the unit (e.g. N/mm²) determines the unit of the output signals Poisson's ratio
Default setting	Measuring grids a, b, c and temperature compensation strain gage: 0 SG type: 0/45/90° Modulus of elasticity: 200000 Poisson's ratio: 0.3
Exception handling	If one of the source signals is invalid, the output signal will be invalid as well.

Information

The measuring grids of the strain gage rosette must be connected as half bridge circuits. It is not possible to make the connection as a quarter bridge circuit with PMX. So if you cannot connect the active strain gage rosette to a second rosette for temperature compensation to form a half bridge circuit, you will have to add fixed resistors (TK0) to the SG measuring grids of the rosette to form half bridge circuits.

13.2.8 Evaluation functions

13.2.9 Filter (IIR, high pass or low pass)

Function	Filters on signal
Source	Input
Output	Result channel
Parameter(s)	Type: Low pass or High pass Characteristic: Bessel or Butterworth Cut-off frequency (-3 dB)
Default setting	Type: Low pass Characteristic: Bessel Cut-off frequency: Off (filter off)
Exception handling	If the source signal is invalid, the output signal will also be invalid. If the output signal is outside the display range ($\pm 3.4 * 10^{38}$), NaN is outputted.

Filter frequency and phase delay of the low-pass filters

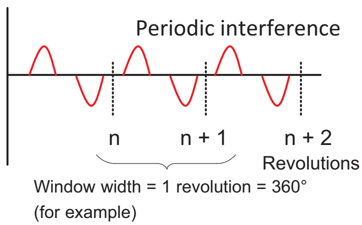
fc in Hz	Bessel Phase delay in ms	Butterworth Phase delay in ms
off	0	0
3000	0.13	0.19
2000	0.21	0.30
1000	0.43	0.61
500	0.86	1.23
200	2.00	3.10
100	4.15	6.17
50	8.45	12.5
20	21.4	30.7

fc in Hz	Bessel Phase delay in ms	Butterworth Phase delay in ms
10	39	47
5	74	91
2	174	216
1	340	430
0.5	680	840
0.2	1680	2090
0.1	3360	4200

The phase delay of the high-pass filters is 0 ms.

Filters with Bessel characteristic (default setting) do not cause any signal distortion, but they have a relatively flat frequency response. Filters with Butterworth characteristic are steeper, but they cause overshooting (about 10%), which stands out especially in the event of rapid signal changes. The most suitable use for filters with Bessel characteristic is for pulse-shaped signals. Filters with Butterworth characteristic are best used with signals for which the bandwidth will be restricted.

13.2.10 Rotation synchronous filter (CASMA)

Function	This calculation suppresses periodic interference that occurs when measuring rotating parts. The filter works synchronously to the angle of rotation, not relative to time or with a fixed cut-off frequency. Because of this the quality of interference suppression does not depend on the rotational speed (CASMA = Crank Angle Synchronous Moving Average). Based on the angle set as the Resolution, the calculation determines a new (moving) average of the number of values recorded in the Window Width. As soon as the rotary motion stops, no new values are calculated. To perform the calculation, in addition to the signal being filtered (a torque, for example) you will need the signal of an angle of rotation sensor. 
Sources	Filter input Angle (0° ... 360°)
Output	Moving average of filter input
Parameter(s)	Window Width: 30° ... 720° Minimum rotational speed in rpm Resolution: 1° ... 8° The window width/resolution ratio must not be greater than 180.
Default setting	Window Width: 180° Minimum rotational speed: 0 Resolution: 1°
Exception handling	—

The resolution determines how frequently (every how many degrees) a new arithmetic mean will be calculated. Note that the maximum permissible rotational speed depends on this value, because the speed of calculation is determined by the total update rate (see **System Options**) menu. The theoretical value is derived by:

Maximum rotational speed = resolution * total update rate / 6 (see also following table).

For practical purposes you should use values that amount to only 10 to 20% of this theoretically possible maximum rotational speed.

Resolution	The theoretical maximum rotational speed at a total update rate of 19200 Hz	The theoretical maximum rotational speed at a total update rate of 38400 Hz
1°	3200 rpm	6400 rpm
2°	6400 rpm	12800 rpm
4°	12800 rpm	25600 rpm
6°	19200 rpm	38400 rpm
8°	25600 rpm	51200 rpm

The following multiples of the rotational speed are suppressed depending on the window width:

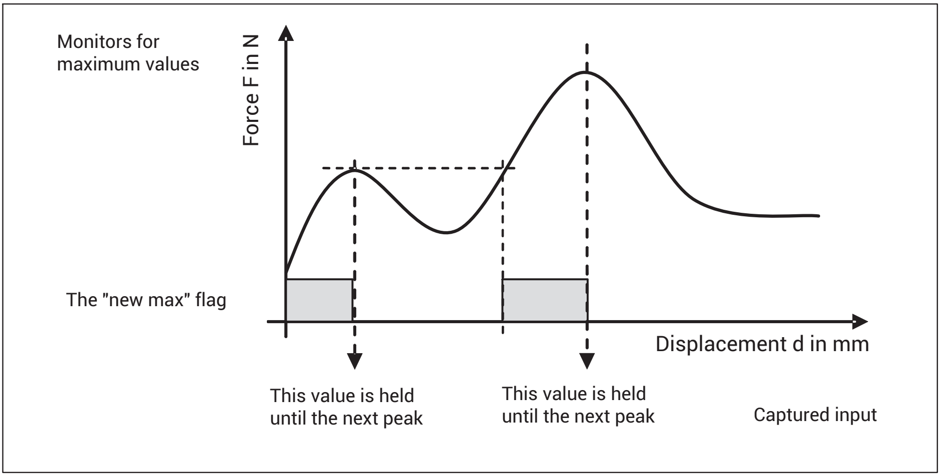
Window width	Multiples
90°	4, 8, 12, ...
120°	3, 6, 9, ...
180°	2, 4, 6, ...
360°	1, 2, 3, ...
720°	0.5, 1.0, 1.5, ...

13.2.11 Peak value

Function	This calculation determines the minimum, maximum or oscillation width (peak-to-peak) value of a signal. You can also determine the value of another channel (Input 2) when an extreme value is reached. If you specify a discharge rate, you will get an envelope function. The maximum or peak-to-peak value decreases with the discharge rate; the minimum value increases with the discharge rate. A new peak value is additionally indicated by a flag. The flag is only at High in the cycle in which a peak value was detected. When the input signal rises, the flag is continuously at High. Reset by (virtual) digital input, flag (level-controlled) or fieldbus (event-controlled) Hold via digital input (level-controlled) This function block may return false values after being first created or after switching parameter sets. Reset the extreme value first in such cases. Low discharge rates can cause problems due to the numerical accuracy applied. Cross-check the discharge rate against the expected peak values. Discharge rate 1 % ... 5 % of peak value: Check whether the value decreases as expected. Discharge rate < 1 % of peak value: The value may not decrease as expected.
Sources	Input 1 Input 2 Hold by Reset by
Outputs	Extreme value input 1 Input 2 at extreme value 1 New extreme value (flag)
Parameter(s)	Function: Maximum, Minimum, Peak-to-peak Invert holding channel Discharge rate (1/second)

Default setting	Inputs 1, 2: 0 Hold by: None Reset by: None Function: Maximum Invert holding channel: No Discharge rate: 0
Exception handling	If one of the source signals is invalid, the output signal will be invalid as well.

Example: Measure displacement when F = F_{max}



13.2.12 Tolerance window

Function	The tolerance window combines several functions: Acquisition of minimum, maximum, peak-to-peak and arithmetic mean and monitoring of exceeding/undershooting a limit for a definable period of time. You can also have the value of another channel calculated when an extreme value is reached. The process is started and stopped by edge-sensitive digital signals. A second input signal is acquired when a new peak (min. or max.) is detected. After the start, all values and the flags are first reset. Then the Input 1 signal is checked for minimum and maximum, and the oscillation bandwidth and arithmetic mean are calculated. The arithmetic mean is calculated over a maximum of 100,000 cycles (about 5.2 s at an update rate of 19.2 kHz). The amount of time between a start and the corresponding subsequent stop is measured as Duration 1 in milliseconds. Signal 1 is also monitored for exceeding or undershooting the threshold values. If the respective threshold is exceeded or undershot, the specified flag is set. If you specify a channel for Input 2, its value when a minimum or maximum of the channel specified for Input 1 occurs will be retained until the next occurrence (Input 2 at Min.1 or Input 2 at Max.1).
Sources	Input 1 Input 2 Start: Starts the measurement Stop: Ends the measurement Upper threshold: After x has exceeded this value, the "Limit Hi" output becomes active until the next start. Lower threshold: After x has undershot this value, the "Limit Lo" output becomes active until the next start.

Outputs	Max Input 1 (since start) Min Input 2 (since start) At Max recorded value of second input signal At Min recorded value of second input signal Peak-to-peak 1 (Input 1 since start) Arithmetic mean (Input 1 since start) Duration 1: the time in ms since start. After the process, the time from start to stop. Flag Limit Hi indicates exceeding of "Upper Threshold" Flag Limit Lo indicates exceeding of "Lower Threshold"
Parameter(s)	Start: on rising/falling edge Stop: on rising/falling edge
Default setting	Input 1, 2: 0 Input 2 Start, stop: None Stop: Ends the measurement. Upper threshold, lower threshold: 0 Start, stop: on rising edge
Exception handling	If the signal at Input 1 is invalid, the peak and average value outputs are not updated, and are invalid until the stop condition is met. On the next start signal the values become valid again. The time output (***Duration***) is not affected by an invalid input signal. The signal at Input 2 is not checked for validity. If the time exceeds 100,000 updates, the arithmetic mean value is no longer updated and becomes invalid.

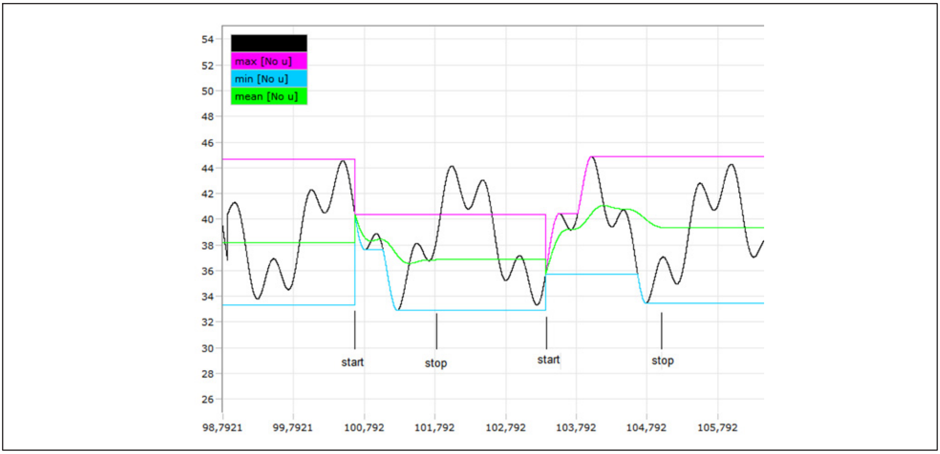


Fig. 13.1 Example of max., min. and mean value. Reset at start, hold from stop to start.

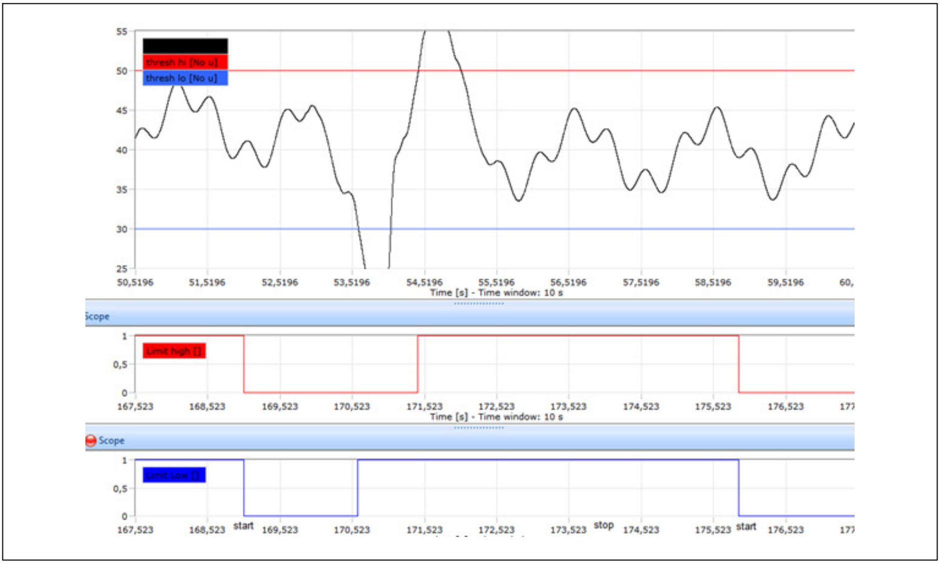


Fig. 13.2 Example of limit values and limit value flags. Reset with start event.

13.2.13 Hold (analog triggering)

Function	Retains the current value of the input signal (instantaneous value). Control is based on the level of an analog signal or a calculation. Hold condition: If the input is inside or outside the upper/lower threshold interval. If the hold condition is met, the input is outputted to the output. If the hold condition is not met, the output is held at the last held value and marked as invalid if necessary. Holding the instantaneous value can be delayed by up to 60000 ms. You can use **Invalid Outside**: ***Yes*** (click on **2**) to additionally set the output to invalid, provided no instantaneous value has been yet acquired.
Sources	Input Control input Init by: Digital input, sets the output to the start value
Output	Captured value
Parameter(s)	Upper limit (of the hold range) Lower limit (of the hold range) Capture Outside: Yes/No On entry only (hold into interval): Yes/No Start Value: Outputted on a positive edge at ***Init by*** input. Invalid Outside: Yes/No Delay: 0 … 60000 ms
Default setting	Upper limit, lower limit: 0 Capture Outside: No On entry only: No Start Value: 0 Invalid Outside: No Delay: 0
Exception handling	If the source signal is invalid, the output signal will also be invalid. However, it will only be marked once the next specified holding event occurs. If the control input is invalid, no instantaneous value is recorded and the output remains on the last value.

13.2.14 Hold (digital triggering)

Function	Retains the current value of the input signal (instantaneous value). Control is based on the signal edge of a digital input (edge-controlled).
Sources	Input Hold by (edge-controlled) Reset by (only effective if no retrigger is used),
Output	Captured value
Parameter(s)	Input Hold by Reset by Invert holding channel Retriggerable: Means the output is updated on each edge of the **Hold by** input. Delay: 0 … 60000 ms (= 1 min); Resolution 0,1 ms The value is only held when the **Hold by** input has been active continuously for this time.
Default setting	Invert holding channel: No Retriggerable: Yes Delay: 0 ms
Exception handling	If the source signal is invalid, the output signal will also be invalid. The Hold function will nevertheless be executed and exported.

13.2.15 Mean (arithmetic, RMS)

Function	The calculation determines the arithmetic mean or root mean square (RMS) value of a signal. There are three ways to define the duration over which a calculation takes place: - By a specific number of values. - When the actual value exceeds or falls below a specific value in the **Trigger** channel. You can define that the actual value must exceed or fall below the limit multiple times (**Number of triggers**). - If a rising or falling edge occurs in a channel. You can define that the edges must occur multiple times. You have to specify different sources depending on which method you want to use. Note that after the device is initialized or the calculation is restarted, the output does not contain a meaningful value until after the second trigger/measured value. A maximum of 100,000 values can be calculated.
Sources	Input Trigger (analog channel): Input that is compared with the threshold input. Only relevant for ***Threshold*** trigger events. Trigger (digital channel): Input for starting the calculation. Only relevant for ***Edge*** trigger events. Threshold: Only relevant for ***Threshold*** trigger events.
Outputs	Mean: either the arithmetic mean or the root mean square. Number: Number of measured values averaged.
Parameter(s)	Trigger: Number of measured values, above/below threshold, rising/falling edge Number of measured values: Number of measured values averaged. Only relevant for ***Number of measured values*** trigger event. Number of triggers: The arithmetic mean (average) value is calculated and updated after this number of trigger events. Only relevant for ***Threshold*** and ***Edge*** trigger events. Mean: arithmetic or RMS

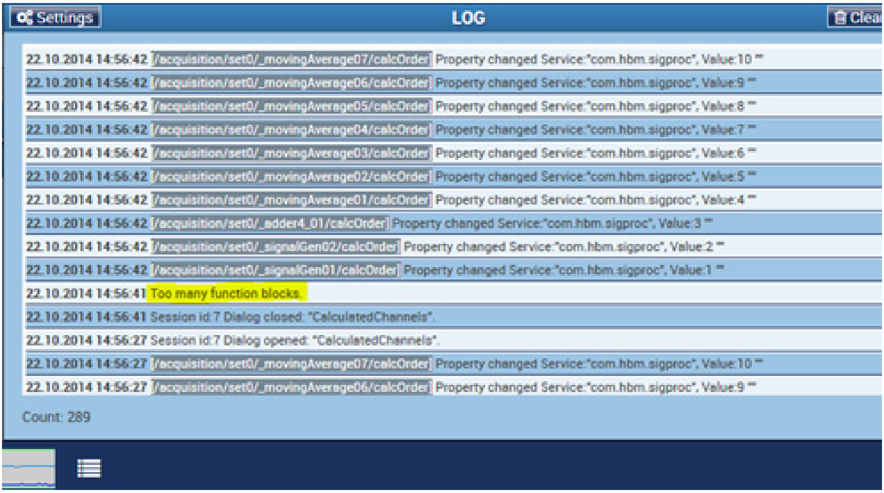
Default setting	Input: 0 Trigger: 0 or none Threshold: 0 Trigger: Above threshold Number of triggers: 1 Mean: arithmetic mean Number of measured values: 2
Exception handling	A maximum of 100,000 values can be calculated. If more values are encountered, the output signal becomes invalid, but the average of the first 100,000 values is outputted. Then the calculation starts from the beginning with the next trigger signal (restart calculation). The inputs are not checked for invalid signals.

13.2.16 Moving average

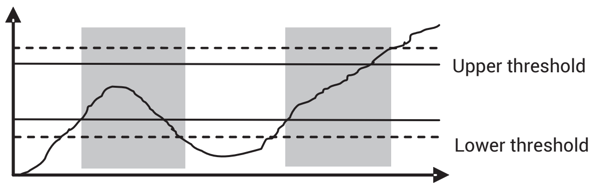
Function	This calculation determines the arithmetic mean of a signal for a certain number of values. You can calculate the arithmetic mean for a maximum of 385 values. The settling time is derived from the number of values divided by the update rate for calculations (default 1s/19200). The calculation is equivalent to a FIR filter.
Source	Input
Output	Moving average
Parameter(s)	Number of measured values (filter width): 1 … 385
Default setting	Input: 0 Number of measured values: 385
Exception handling	If the source signal is invalid, the last valid value is displayed and the output signal becomes invalid. The calculated is stopped during this time. As soon as the source signal becomes valid again, the calculation resumes. However, the specified number of values must be present in the buffer (settling time) before the values are correct.

This calculation is very suitable among other things for suppressing interference frequencies of 50 or 60 Hz and their harmonics. At an update rate of 19200 Hz, you will need 384 values to suppress a frequency of 50 Hz and the harmonics 100 Hz, 150 Hz, 200 Hz, etc. (19200/50 = 384). Use 320 values for 60 Hz and the corresponding harmonics (19200/60 = 320).

The calculation requires a relatively high amount of internal memory. So usually no more than six or seven calculations of this type can be performed. If not enough memory is available, the error message "Too many function blocks" is generated.

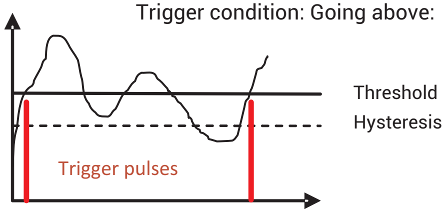
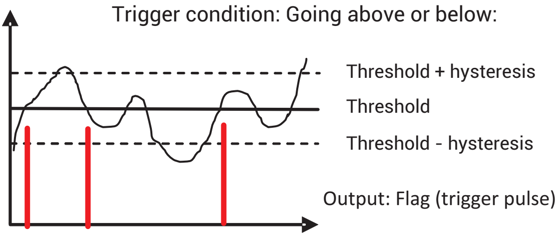


13.2.17 Trigger (range)

Function	This calculation monitors whether an analog signal is within or outside a window. Enter the high and low Threshold (the window). The Hysteresis is outside the window, meaning above the high threshold and below the low. 
Sources	Input Upper threshold Lower threshold
Output	Flag: Active when the condition is met

Parameter(s)	Hysteresis: Effective when the input moves out of the window. Delay: The measurement signal must fulfill the trigger condition for at least as long as specified here before a switching operation takes place. The time applies only to going over the thresholds, not to the hysteresis. Active: high or low. Sets the level of the output when the input value is within the window.
Default setting	Input: 0 Upper threshold: 1 Lower threshold: 0 Hysteresis: 0 Delay in ms: 0 Active: high
Exception handling	If the source signal is invalid, the value for *not* **Active** is outputted. The values for the thresholds and the hysteresis are not checked.

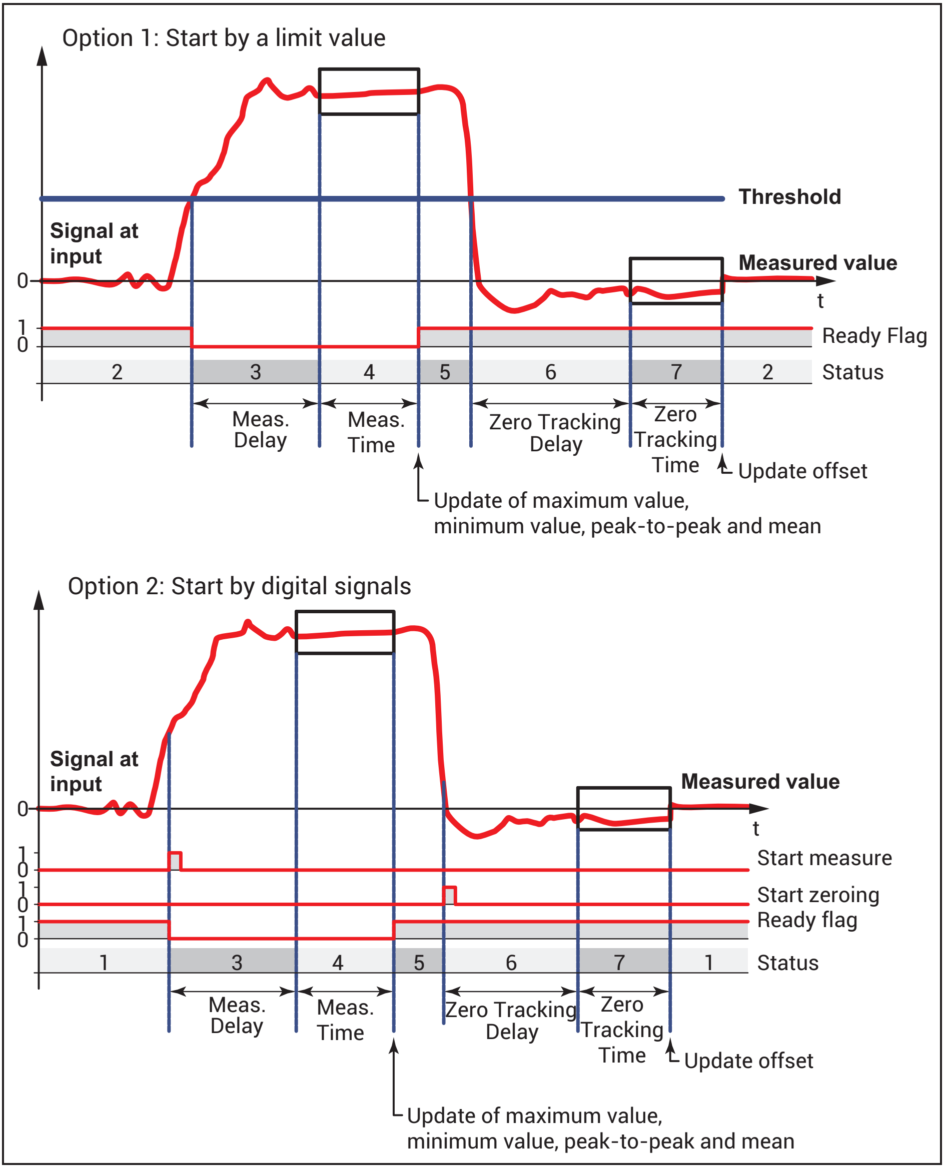
13.2.18 Trigger function (pulse)

Function	This calculation monitors whether an analog signal goes above and/or below a specific value, and generates a pulse if the measured value was previously outside the hysteresis at least once. The length of the pulse is 1/total update rate (default 1/19200 = 52 µs). Trigger condition: Going above:  Trigger condition: Going above or below: 
Sources	Input Threshold

Output	Flag
Parameter(s)	Hysteresis: Effective when the input moves beyond the thresholds. Condition: On going above or below threshold, or both. Active: High or low if the condition is met.
Default setting	Input: 0 Threshold: 0 Hysteresis: 0 Condition: Underrange Active: high
Exception handling	If the source signal is invalid, the value for *not* **Active** is outputted. If the source signal becomes valid again, the hysteresis will not be considered until after the first trigger pulse. The threshold and hysteresis values are not checked.

13.2.19 Checkweigher

Function	A checkweigher weighs while the product is being moved, such as on a conveyor belt. The aim of this dynamic weighing is a high throughput rate (weighing operations per minute) with no loss of accuracy (low standard deviation). This calculation filters an easily usable signal from a severely fluctuating, or noisy, signal by using a specific segment of the signal curve to compute an arithmetic mean value. This enables you to reduce the impact of overlaid interference. You can also define a range in which the signal is averaged and used as the zero value for further measurements. You have two options in each case for starting measurement and zeroing: - By a limit value. - By a digital signal. You can also combine the methods - that is. start measuring an arithmetic mean value, min/max etc. by way of a limit value and start zero measurement by a digital signal. If you do not want to monitor starting and stopping of the calculation via a digital signal, set Enable by to 1 (Constant signal) and Start on to High level for example. The calculation is then performed whenever the conditions (Threshold or Start Measure/Start Zeroing with) are met. Maximum, Minimum, Peak-Peak and Mean are the values determined during the measuring time via the signal at the input. The Offset is the arithmetic mean over the Zero Tracking Time. The measured value is continually updated throughout the time (current measurement value), and corresponds to the input signal minus the offset. After a restart (Device menu), a zero measurement is first performed with the duration of Zero Tracking Time.



Option 1: Start by a limit value
Threshold
Signal at input
Measured value
0
t
1
0
Ready Flag
2
3
4
5
6
7
2
Status
Meas. Delay
Meas. Time
Zero Tracking Delay
Zero Tracking Time
Update offset
Update of maximum value, minimum value, peak-to-peak and mean
Option 2: Start by digital signals
Signal at input
Measured value
0
t
1
0
Start measure
1
0
Start zeroing
1
0
Ready flag
1
3
4
5
6
7
1
Status
Meas. Delay
Meas. Time
Zero Tracking Delay
Zero Tracking Time
Update offset
Update of maximum value, minimum value, peak-to-peak and mean

	Option 1: The specified limit value (Threshold) is used in the example for both start conditions. Use a Constant signal for Threshold. Also set the **Start Measure with** and **Start Zeroing with** conditions to 0 (Constant signal). Option 2: Specify a digital signal (digital input, limit switch or flag) for the **Start Measure with** and **Start Zeroing with** conditions. As soon as a channel is entered here, any specified limit value is ignored for the condition in question. The inputs are triggered by an edge from Low to High. A constant value deactivates the input.
Sources	Input Threshold: Does not have to be set when triggered by flags. Enable by: Toggling this flag resets everything. Start Measure with: As soon as a digital signal is set, the limit value is no longer taken into account. Start Zeroing with: The zero measurement starts after the set delay time.
Outputs	Meas. value: is continuously updated. Mean: from the signal measuring time; updated when the Ready flag becomes active. Max: Maximum value from the signal measuring time; updated when the Ready flag becomes active. Min: Minimum value from the signal measuring time; updated when the Ready flag becomes active. Peak-to-peak Peak-to-peak value from the signal measuring time; updated when the Ready flag becomes active. Offset: the mean value of the input during zero measurement. It is updated at the end of the zero measurement. Status: shows the section in which the measurement is located (see graphs above and table below). Ready flag: indicates the end of the signal measuring time.

Parameter(s)	Measurement Delay: Delay for start of measurement, 0 … 30000 ms. Measurement Time: Width of measurement window 0 … 10000 ms. If you enter 0 here, the measurement is deactivated, but the zero measurement continues to work. Zero Tracking Delay: Time from the drop below the threshold or from the flag to starting zero measurement until start of measurement, 0 ... 30000 ms. Measuring time for zero value: Time duration of zero measurement, 0 ... 10000 ms. If you enter 0 here, the zeroing is deactivated, but the measurement continues to work. Start on: High or Low level.
Default setting	Input = 0 Threshold = 0 Enable by: 1 Start Measure with: 0 Start Zeroing with: 0 Measurement Delay: 1 ms Measurement Time: 1 ms Zero Tracking Delay: 1 ms Measuring time for zero value: 1 ms Start on: High level
Status = 1, 2	Wait for start of measurement or value exceeding the limit value.
Status = 3	Wait for end of measurement delay.
Status = 4	Status during measuring time. When the measurement is complete, the maximum, minimum, peak-to-peak and mean values are updated.
Status = 5	Wait for start of zero measurement or value undercutting the limit value.
Status = 6	Wait for end of zero measurement delay.
Status = 7	Status during zero measurement. When the zero measurement is complete, the offset is updated.

Exception handling	If the input value is invalid, no operation is performed, and the measured value at the output becomes invalid. All other outputs keep their respective value and status. These exceptions occur only in threshold trigger mode: - If the net value falls below the threshold before starting the measurement, the measurement execution is delayed until the threshold is exceeded again. - If the net value within the measurement window falls below the threshold, the measurement is aborted. The next step is to delay the zero measurement. - If the measured value exceeds the threshold before the start of the zero delay, this and the zero measurement are skipped. - If the measured value exceeds the threshold within the zero measurement, the zero measurement is aborted and the zero value is not changed.

13.2.20 Mathematical functions

13.2.21 Adder/subtractor

Function	Adds four summands, each weighted by one factor Output = factor1 * input1 + factor2 * input2 + factor3 * input3 + factor4 * input4
Sources	Summand 1 … 4
Output	Total
Parameter(s)	Factor 1 … 4
Default setting	Summand 1 … 4: 0 Factor 1 … 4: 0
Exception handling	If one of the source signals is invalid, the output signal will be invalid as well. The calculation will nevertheless be performed and exported. In case of out-of-range , NaN ($\pm 3.4 * 10^{38}$) is outputted.

13.2.22 Multiplier

Function	Multiplies *four* signals together
Sources	Input 1 … 4
Output	Product
Parameter(s)	—

Default setting	Input 1 … 4: 1
Exception handling	If one of the source signals is invalid, the output signal will be invalid as well. The calculation will nevertheless be performed and exported. In case of out-of-range , NaN (±3.4*10^{38}) is outputted.

13.2.23 Divider

Function	Quotient = dividend/ divisor
Sources	Dividend Divisor
Output	Quotient
Parameter(s)	—
Default setting	Dividend: 1 Divisor: 1
Exception handling	If one of the source signals is invalid, the output signal will be invalid as well. The calculation will nevertheless be performed and exported. In case of out-of-range , NaN (±3.4*10^{38}) is outputted.

13.2.24 Counter

Function	Counts the transitions of a digital signal. Maximum counter reading 2^{32} -1. The counter mode is either positive edge, negative edge, or both edges. A digital input activates or deactivates the counting process. If the counter exceeds a defined value, a flag can be set. The counter can be cleared after a defined period without countable transitions.
Sources	Input: The digital signal to be counted Gate: A digital signal that activates the counter Reset by: A digital signal that clears the counter (level-controlled)
Outputs	Output: The current counter value Flag: Set if counter value ≥ threshold value for flag
Parameter(s)	Mode: Rising or falling edge, both edges Timeout after: Resets the counter if no countable edge was detected in this period. 0 ms means that the timeout is deactivated. Threshold value for flag: Specifies the counter value at which the flag is to be set

Default setting	Input: 0 Gate: 1 Reset by: 0 Mode: Rising edge Timeout after: 0 ms (no timeout) Threshold value for flag: 1
Exception handling	—

13.2.25 Integrator

Function	Integrates a signal over the Integration Time. You can reset the integration via a digital signal or a flag, and you can assign an Initial Value that will be used after the reset. The resolution is 1/update rate. To ensure that the signal does not become infinitely large or small, you can limit the value range for the result with Ymax and Ymin. Integration will then stop when one of these values is reached.
Sources	Input Reset: Resets the output value to the value at the input (Init value) Start Value: Initial value of integration
Output	Result channel
Parameter(s)	Integration Time: Duration for integration Ymax, Ymin: Limits the value range of the output. If the integrator is limited, no wind-up can occur.
Default setting	Input: 0 Reset: 0 Start Value: 0 Integration Time: 1 s Ymax: $10.^{6}$ Ymin: -10^{6}
Exception handling	If the source signal is invalid, the last valid value is displayed and the output signal becomes invalid. The calculated is stopped during this time.

13.2.26 Differentiator

Function	Calculates the derivative of a signal via Δt = 4/update rate. To ensure that the signal does not become infinitely large or small, you can limit the value range for the result with Ymax and Ymin. We recommend additionally limiting the bandwidth of the input signal with a low-pass filter.
Source	Input
Output	Result channel
Parameter(s)	Ymax, Ymin: Limits the value range of the output
Default setting	Input: 0 Ymax: $10.^{6}$ Ymin: -10^{6}
Exception handling	If the source signal is invalid, the last valid value is displayed and the output signal becomes invalid. The calculated is stopped during this time.

The bandwidth of the calculated signal is limited and depends on the update rate. The signal becomes zero with the frequency specified under Max. bandwidth.

Update rate	-3 dB at	Max. bandwidth
19200/s (default)	3.6 kHz	4.8 kHz
38400/s	7.2 kHz	9.6 kHz

13.2.27 Cartesian to polar coordinates

Function	This calculation converts two input channels representing the position (x, y) of a point in the Cartesian coordinate system into the corresponding polar coordinate values. Two output channels, one channel with the angular values (Θ, theta) and one channel with the radius values (r) are available. The value range for the angle extends from -179.99° to +180°. Multiply the value by π/180 as necessary to get the radian (rad).
Sources	X: X coordinates Y: Y coordinates
Outputs	Radius r, same unit as x, y Angle Θ (theta) in degrees
Parameter(s)	None

Default setting	X: 1 Y: 0
Exception handling	If one of the source signals is invalid, the output signal will be invalid as well. The calculation will nevertheless be performed and exported.

13.2.28 Polar coordinates to Cartesian coordinates

Function	This calculation converts two input channels representing the position (radius r, angle Θ = theta) of a point in polar coordinates into the corresponding Cartesian coordinates. Two output channels are available, one channel with the x values and one channel with the y values. The angle value must be in degrees (-360° to +360°).
Sources	Angle: Angle in degrees Radius: Radius r
Outputs	X: X coordinates, same unit as r Y: Y coordinates, same unit as r
Parameter(s)	None
Default setting	Angle: 0 Radius: 0
Exception handling	If one of the source signals is invalid, the output signal will be invalid as well. The calculation will nevertheless be performed and exported.

13.2.29 Modulo function

Function	Calculates the remainder from a division. In addition to whole numbers, you can also use the calculation for real numbers. For negative numbers, the same sign is used for the result as for the signal source, and an absolute value less than the absolute value of the divisor is calculated.
Source	Input
Output	Result channel
Parameter(s)	Divisor
Default setting	Input: 0 Divisor: 1
Exception handling	If the source signal is invalid, the last valid value is displayed and the output signal becomes invalid. The calculated is stopped during this time.

13.2.30 Constant signal

This calculation allows you to define constant values. The constants invalid, 0 (constant 0), 1 (constant 1) and -1 (constant -1) as well as π (pi), π/2 (pi/2) and 2*π (2*pi) are always available, and do not need to be created.

13.2.31 Technology functions

13.2.32 Two-state controller

Function	The calculation implements a two-state controller with delayed/ yielding feedback. Tr2 must be greater than Tr1. Tr1 is the dominant section constant.
Sources	Setpoint Actual value
Output	Flag, outputted in Calculated Channel Flags.
Parameter(s)	Hysteresis Kr: Feedback amplification, common amplification of the two parallel feedback branches. Tr1: Feedback time constant of the negative feedback branch (PT1 element), Tr1 < Tr2. Tr2: Feedback time constant of the positive feedback branch (PT1 element), Tr1 < Tr2.
Default setting	Setpoint: 0 Actual value: 0 Hysteresis: 1 Kr: 0 Tr1, Tr2: 10^{38}; this effectively turns off the feedback branches.
Exception handling	If either the setpoint or the actual value is invalid, the computed values from feedback are frozen and the output is deactivated.

The regulating variable of the control affects the PT1 element. The output signal of the PT1 element acts in turn subtractively on the control deviation. An additional PT1 element with a longer time constant affects the control deviation additively. The step-function response of feedback accordingly becomes zero after a sufficiently long time: In steady state of the control loop, DC voltage components of the two returned sawtooth signals of the PT1 elements are subtracted. The relatively small difference between the two ripples remains as an AC current superimposed around the control deviation. The amplitudes of this ripple are determined by the magnitude of the hysteresis or by the switching frequency that forms. As the time constant of the subtractive feedback increases, the switching frequency of the control loop drops. As the amplification Kr increases, the switching frequency rises, the control deviation increases and the overshooting response of the control quantity decreases.

13.2.33 PID controller

Function	This calculation implements a PID controller in a parallel structure. The PID (proportional-integral-derivative) controller consists of three elements: the P element, I element and D element. The parallel structure of the control makes it possible to prevent the wind-up effect. You can limit the output signal with Ymax and Ymin. Tp is the parasitic time constant at 1/update rate (52 µs at 19200 Hz). $$Kp\left(1+\frac{1}{Ti*s}+\frac{Td*s}{Tp*s+1}\right)$$
Sources	Setpoint Actual value Turn on with: Allows deactivating of the controller and output of a defined signal.
Output	Result channel: Controller output
Parameter(s)	Kp: Amplification, P component Ti: Reset time, I component Td: Derivative time, D component Ymax: Upper limit of controller output Ymin: Lower limit of the controller output Invalid Signal Result: The value is outputted when **Switch on with** goes Low.

Default setting	Kp = 0 Ti = 10^{38} Td = 0 Ymax = 10^{20} Ymin = -10^{20} Invalid Signal Result: 0
Exception handling	If either the setpoint or the actual value is invalid, the controller stops and the value becomes invalid, but the output retains the last value. If the controller is deactivated (Turn on with = 0), the value of **Invalid Signal Result** is outputted.

You cannot define the PID controller as a simple P controller. At least one I element must be present in addition.

13.2.34 RTD Pt100 on PX455

Function	The PX455 measurement card can be used to connect a Pt100 resistance thermometer in a half bridge circuit (100 ohm completion resistor for the half bridge). You can use this function to convert the measured signal into the temperature value in °C. The input signal of the calculation must be available in mV/V. The display range goes from -100 °C to +500 °C.
Source	Input: Signal from PX455
Output	Result channel: Temperature in °C
Parameter(s)	Cable resistance: dependent on the Pt100 connection - The completion resistor is connected directly to the PX455 In this case, you must specify the resistance of the cable to the resistance thermometer (simple distance) here. - The completion resistor is connected with the same cable length as the resistance thermometer In this case, enter 0 here, as the cable resistances balance each other out due to the bridge layout.
Default setting	Cable resistance: 0
Exception handling	If the source signal is invalid, the last valid value is displayed and the output signal becomes invalid. The calculated is stopped during this time. If the temperature value is outside the display range, -333.3 °C is displayed and the output also becomes invalid.

13.2.35 Signal generators (square, triangle, sine, ...)

Function	This calculation generates a periodic signal, for example a sine wave. Determine the desired frequency, amplitude and, if applicable, an offset. Signal forms available are sine, square wave, triangle, counter, constant and (white) noise. The signal on **Switch on with** determines whether the selected function will be outputted. In case of a Low signal, output is stopped and the value specified for Offset is outputted. When the signal switches back to High, a new output period begins. You can define how often periodic signals (sine periods, square wave periods, etc) are outputted with **Number of periods**. Entering 0 means continuous output.
Source	Switch on with: Starts and stops the sequence
Output	Result channel
Parameter(s)	Function: Sine, square, noise, counter, constant, triangle. Frequency: 0 … Update rate/4 (default setting 4800 Hz); only effective for sine, square, triangle. Number of periods: Number of periods to be executed. Only effective with periodic waveform. Zero means that the operation will be executed endlessly. Maximum value 2^{31}-1. Amplitude: Signal amplitude Offset: Outputted after the specified number of periods
Default setting	Switch on with: 1 Function: Sine Frequency: 100 Hz Number of periods: 0 (infinite) Amplitude: 1 Offset: 0
Exception handling	If the output signal is outside the display range, NaN ($\pm 3.4*10^{38}$) is outputted.

Use 1/10 the update rate as the maximum frequency. Otherwise the output signal will not be formed from enough points. This results in a distorted curve shape (stairway effect) for a sine wave vibration.

Counter function: At an offset of 0, the counter generates a number between -Amplitude and +Amplitude that is incremented by one each time by the update rate (default setting 19200 Hz).

White noise function: The noise signal is generated by a pseudo-random sequence with a period of 2^{31}.

13.2.36 Logic modules (AND, OR …)

Function	This calculation makes various logical (Boolean) functions available: AND, NAND, OR, NOR, XOR, XNOR and NOT. Depending on the function, one or more function modules are available per calculation, for example 1 AND function, 2 XOR functions or 4 NOT functions.
Sources	Input A … D: Digital signals
Outputs	Y1 Y2: Only used with XOR, XNOR, NOT Y3: Only used with NOT Y4: Only used with NOT
Parameter(s)	Function: AND, NAND, OR, NOR, XOR, XNOR or NOT
Default setting	Inputs A … D: 0 Function: AND
Exception handling	—

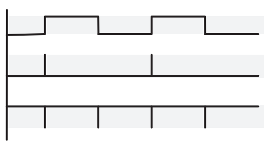
13.2.37 4:1 multiplexer

Function	Depending on the control bits, input 1, input 2, input 3 or input 4 is outputted. Control bit 0 — 0 — 1 — 0 — 1 Control bit 1 — 0 — 1 — 0 — 1 Output — Input 1 — Input 2 — Input 3 — Input 4
Sources	Inputs 0 … 3
Output	Result channel
Parameter(s)	Control bit 0, 1
Default setting	Inputs 0 … 3: 0 Control bit 0: 0 Control bit 1: None
Exception handling	If the source signal is invalid, the output signal will also be invalid. If the output signal is outside the display range, NaN (±3.4*10^{38}) is outputted.

13.2.38 Deadband

Function	Holds the output signal constant as long as the input signal is less than the output signal by less than the value specified for Delta. If that value is exceeded, the current input signal is outputted and used as a new value for calculating the deviation. The function supports you in holding an (output) value constant as long as possible, even if the (input) signal changes slightly. The output does not change until there are larger changes in the input signal. There is no change in the output signal within the range defined by Delta (deadband).
Source	Input
Output	Result channel
Parameter(s)	Delta: The maximum signal change before the output signal is changed.
Default setting	Input: 0 Delta: 1
Exception handling	If the source signal is invalid, the output signal will also be invalid. The calculated is stopped during this time.

13.2.39 Edge detector

Function	This calculation monitors whether a digital signal changes level, and generates a pulse if it does. You can monitor rising edges only, falling edges only or both edges. The length of the pulse is 1/total update rate (default 51 µs). The calculation consists of two function blocks. This means you can monitor different digital signals (A and B). Input Rising edge, High active Both edges, Low active 
Sources	Input A and B
Outputs	Flag A and B
Parameter(s)	Function A, B: On rising/falling edges, or both High active A, B: If yes, the output is High when the condition is met, otherwise Low.

Default setting	Input A, B: 0 Function A, B: On rising edge High active A, B: Yes
Exception handling	—

13.2.40 Pulse-width measurement

Function	This calculation measures the time between two edges. You can use one or two (digital) channels as the input. The result can be outputted as a time (seconds or milliseconds) or as frequency (1/s). Maximum resolution (smallest measurement duration): 1/update rate; at an update rate of 19200/s this is equivalent to about 52 µs. Maximum measurement duration: 1/update rate * 8,388,608; at an update rate of 19200/s this is equivalent to about 437 s.
Sources	Start input: Starts time measurement Stop input: Stops time measurement
Output	Result channel: Contains the pulse duration or the frequency
Parameter(s)	Start: on rising/falling edge Stop: on rising/falling edge If the Start and Stop conditions (input, edge) are identical, the period duration is measured. Otherwise the pulse length is measured. Result type: Frequency, time (in seconds or milliseconds)
Default setting	Start input: None Stop input: None Start: on rising edge Stop: on rising edge Result type: Time
Exception handling	The output becomes invalid and the calculation is stopped if the maximum measurement duration (about 437 seconds) is exceeded. This status is reset by the next start signal.

Pulse duration, corresponding frequency and achievable measurement uncertainty at an update rate of 19200/s.

Pulse duration/period in ms	Frequency in Hz	Measurement uncertainty as a %
1	1000	5.21
2	500	2.60
5	200	1.04
10	100	0.52
20	50	0.26
50	20	0.10
100	10	0.05
200	5	0.03
500	2	0.01
1000	1	0.01

13.2.41 Timer

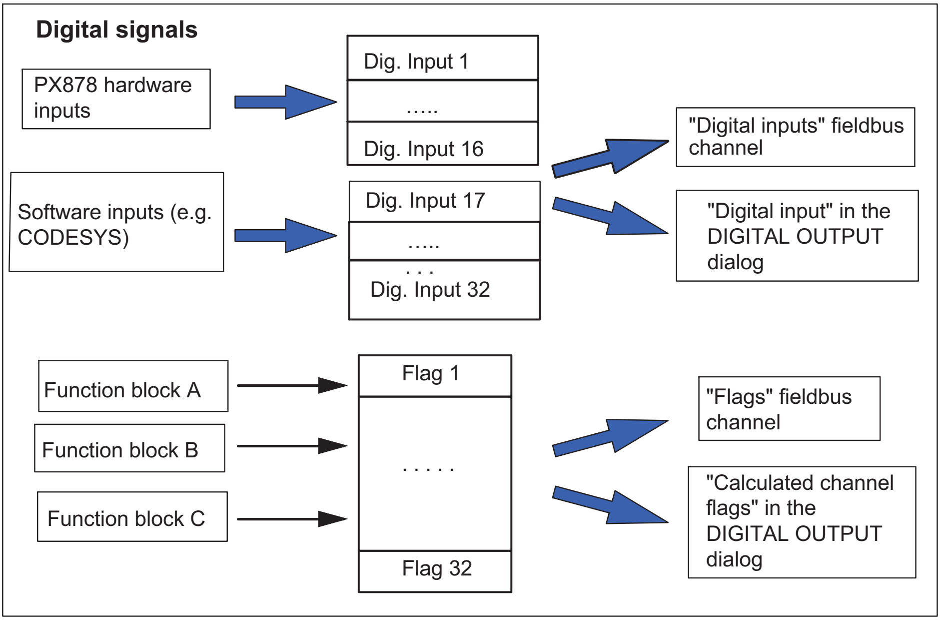
Function	After an adjustable time (Interval), sets a flag to High and holds the level for the Flag Pulse Length before the level is reset to Low. If you enter 0 for the Flag Pulse Length, the flag will be set to High for at least one update interval (1/update rate). The maximum interval is 100,000 seconds. The Flag Pulse Length should not be less than the Interval, otherwise the level will always be High as long as the timer is running. The sequence can only occur once for a positive edge (Type: Single Shot) or repeatedly (Type: Continuous) as long as a positive source signal is present. With Single Shot, a positive edge starts the timer and the next positive edge does not start the timer again until after the timer elapses. In continuous mode the timer starts as soon as a positive source signal is present and starts again immediately after the interval has elapsed. As soon as there is no more positive source signal present, the timer stops and does not start again until a positive source signal is present.
Sources	Switch on/start with: The timer starts with a High level
Outputs	Result channel: Current time value; 0 when the timer starts, the **Interval** value at the end of the interval. When the timer is stopped, the interval time is outputted constantly. Flag: Set to High when the timer expires.

Parameter(s)	Interval: Duration of timer interval Type: Continuous or single shot Flag Pulse Length: Duration of output pulse count
Default setting	Switch on/start with: 1 Interval: 1 s Type: Continuous Flag Pulse Length: 0.1 s
Exception handling	–

13.2.42 Connection channel with (optional) delay (CODESYS)

Function	The connecting channel forwards the input signal on the output, optionally with a delay. You can duplicate the input signal with this function, for example to forward it to other functions such as filters. For the PMX version with CODESYS (WGX001), the function is also suitable for forwarding values or signals that have been determined to other channels or outputs. Delay: Enter the number of cycles for the delay. The cycle time depends on the update rate; 1 cycle = 1/update rate. With an update rate of 19200 Hz (default setting), a cycle is equivalent to 52.08 µs (enter 1). An entry of 0 means "No delay".
Source	Input
Output	Result channel
Parameter(s)	Delay: Indicated in cycles
Default setting	Input: 0 Delay: 0
Exception handling	If the source signal is invalid, the output signal will also be invalid. The buffer with the delayed values is completely deleted and the output does not become valid again until the buffer is completely filled in with valid values again.

13.2.43 Processing of digital signals



13.3 Calculation examples

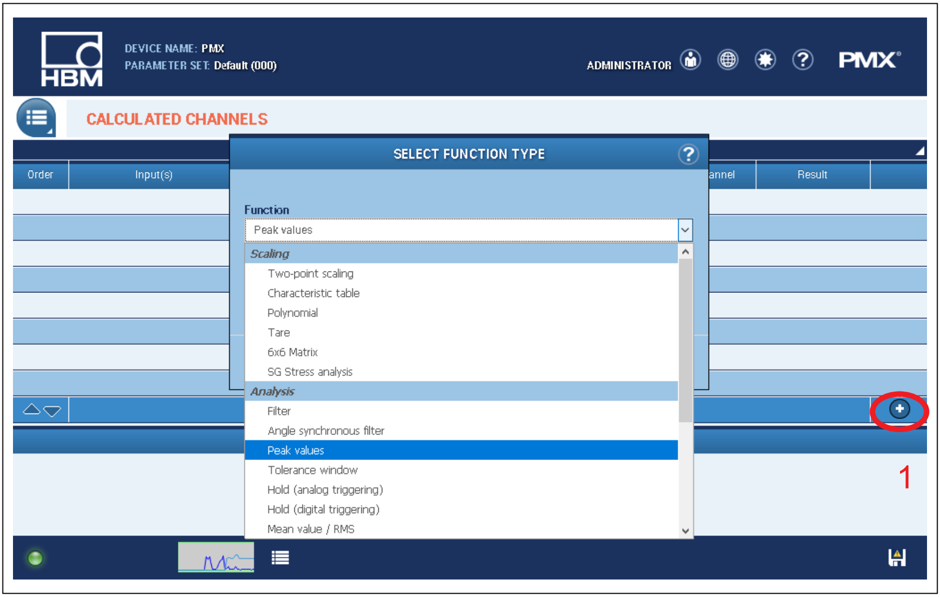
Tip

Examples of the calculation channels can be found in the TechNotes at https://www.hb-m.com/de/2981/pmx-modular-measuring-amplifier-system-for-the-iot/.

13.3.1 Peak value generation

Before you start the configuration, please make sure that you have the necessary rights to make changes:

- Click the person icon in the top right corner and choose **Maintenance** or **Administrator**.
- Then click on **Settings >** ***Calculated channels***.
- Click on **Add function**.
- You can then select one of many functions for each individual channel. In this example we are using "Peak value (Maximum)".

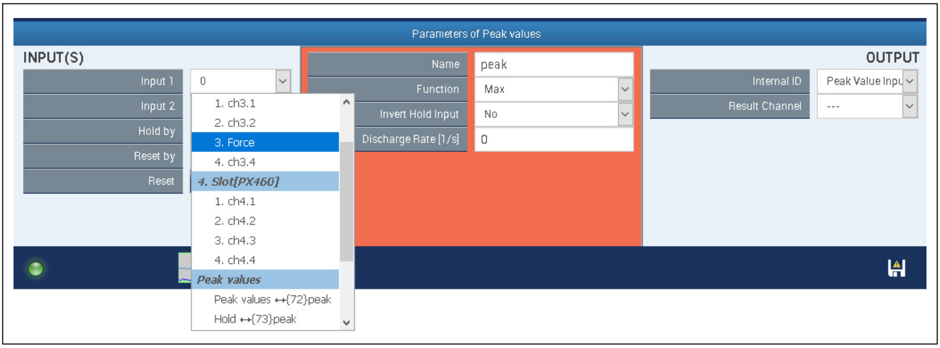


In the next section you make the settings for your function.

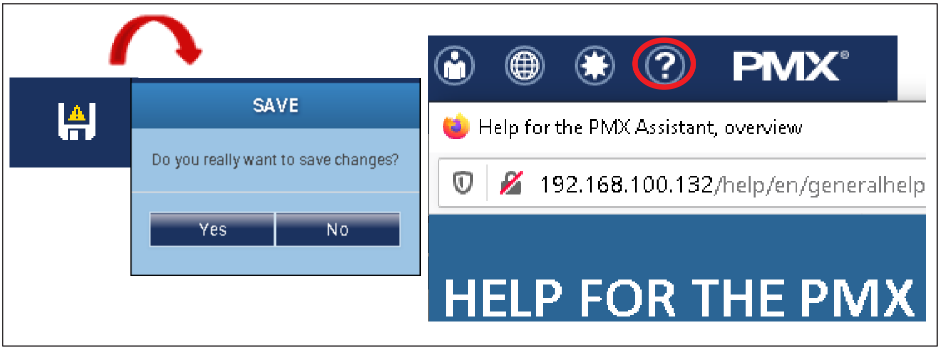
- Choose ***Peak values***.
- Select an input channel for **Input channel 1**.
- Choose ***Max*** under **Function**.
- For **Output** select the ***Input 1 extreme value*** signal.

- Specify the **Result channel** for output of the maximum.
- Specify a **Name** for the channel.
- Also specify the **Decimal places**, the **Physical unit** and the **Update rate**.

  You can change most of the settings at any time later, though the "Peak values" main function cannot be changed.



- Don't forget to save your changes. Click on the floppy disk icon at the bottom right to do so.



Tip

You can get assistance from the Help function at the top of the PMX web browser (colored red).

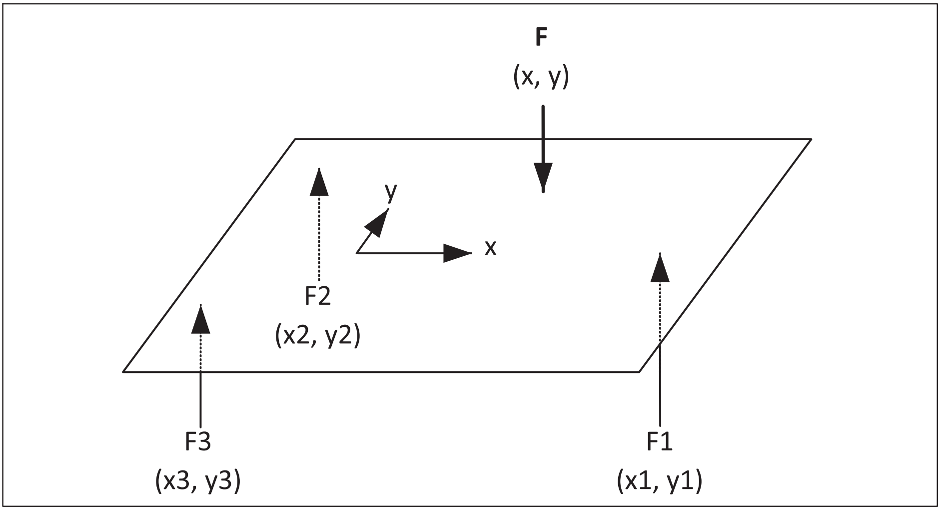
13.3.2 Calculating the force introduction point

Short description

Determination of the force introduction point on a plate with three force transducers.

Introduction

The coordinates of a force F being measured can be easily determined using three force transducers F1, F2, F3.



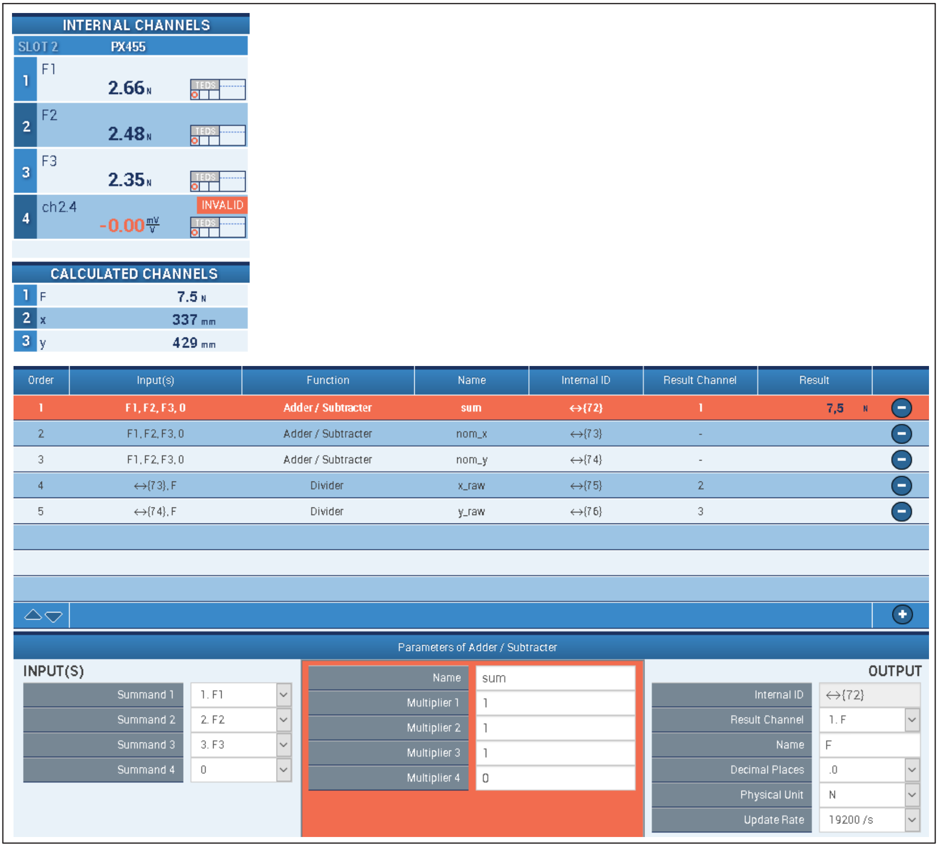
A plate is supported on three transducers; the force F acts orthogonally. From the equilibrium of moments relating to the source, the force introduction point results as

$$x = \frac{F1 \cdot x1 + F2 \cdot x2 + F3 \cdot x3}{F}$$

$$y = \frac{F1 \cdot y1 + F2 \cdot y2 + F3 \cdot y3}{F}$$

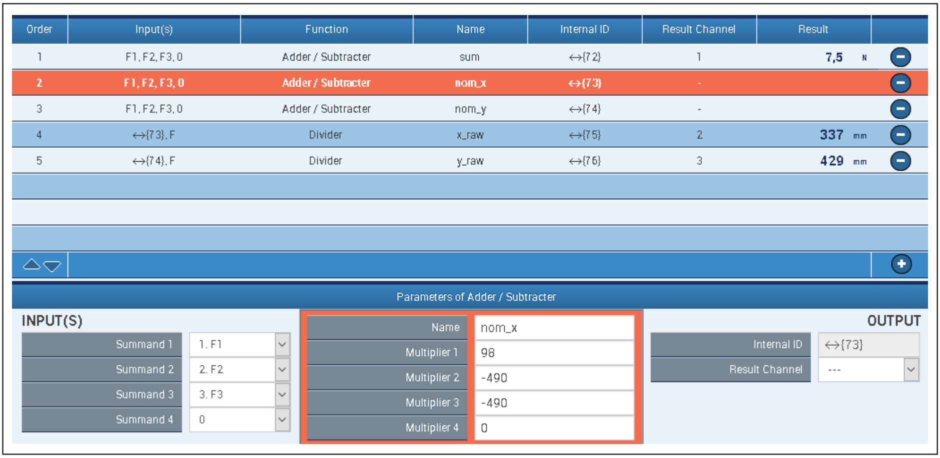
Procedure

The force F is the sum of the three single forces:



The counters for the x and y calculation are determined in an intermediate step. The coordinates x1, y1, x2,.... are included in the factors of the summands.

For x:

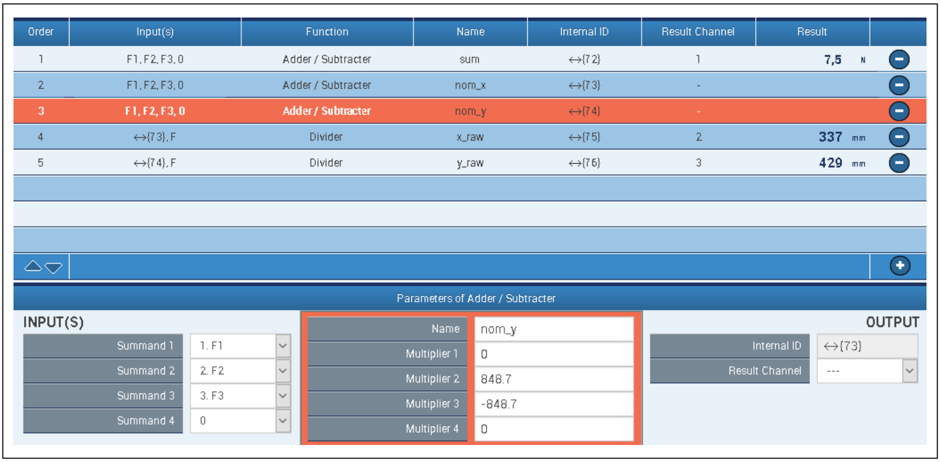


Order	Input(s)	Function	Name	Internal ID	Result Channel	Result
1	F1, F2, F3, 0	Adder / Subtracter	sum	↔{72}	1	7,5 N
2	F1, F2, F3, 0	Adder / Subtracter	nom_x	↔{73}	-	
3	F1, F2, F3, 0	Adder / Subtracter	nom_y	↔{74}	-	
4	↔{73}, F	Divider	x_raw	↔{75}	2	337 mm
5	↔{74}, F	Divider	y_raw	↔{76}	3	429 mm

Parameters of Adder / Subtracter

INPUT(S)				OUTPUT	
Summand 1	1. F1	Name	nom_x	Internal ID	↔{73}
Summand 2	2. F2	Multiplier 1	98	Result Channel	---
Summand 3	3. F3	Multiplier 2	-490		
Summand 4	0	Multiplier 3	-490		
		Multiplier 4	0		

For y:

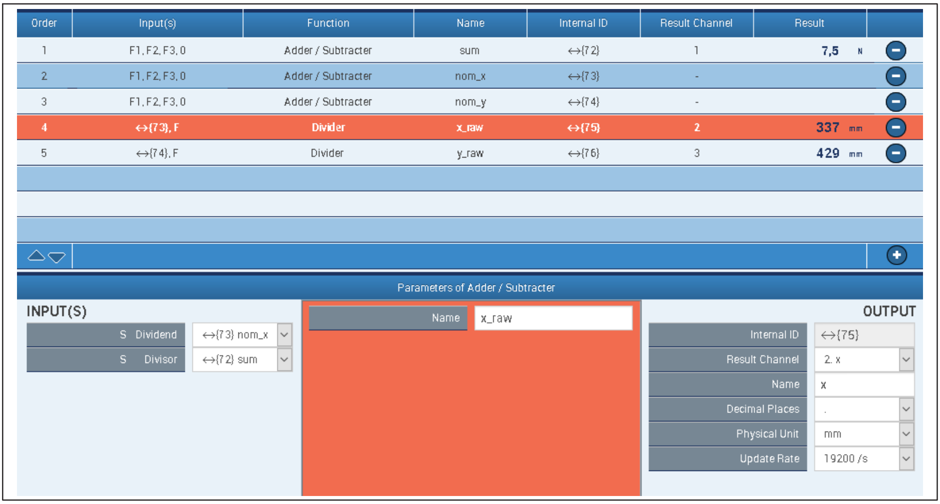


Order	Input(s)	Function	Name	Internal ID	Result Channel	Result
1	F1, F2, F3, 0	Adder / Subtracter	sum	↔{72}	1	7,5 N
2	F1, F2, F3, 0	Adder / Subtracter	nom_x	↔{73}	-	
3	F1, F2, F3, 0	Adder / Subtracter	nom_y	↔{74}	-	
4	↔{73}, F	Divider	x_raw	↔{75}	2	337 mm
5	↔{74}, F	Divider	y_raw	↔{76}	3	429 mm

Parameters of Adder / Subtracter

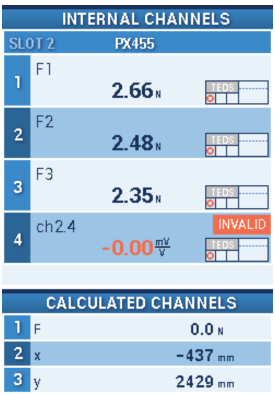
INPUT(S)				OUTPUT	
Summand 1	1. F1	Name	nom_y	Internal ID	↔{73}
Summand 2	2. F2	Multiplier 1	0	Result Channel	---
Summand 3	3. F3	Multiplier 2	848.7		
Summand 4	0	Multiplier 3	-848.7		
		Multiplier 4	0		

x and y are finally calculated with two divisions. Here for x (y in the same way):



Implausible values in the no-load state

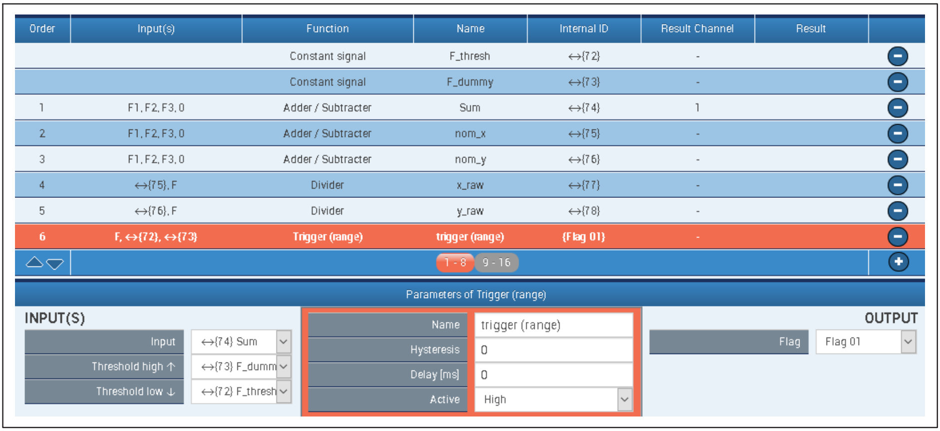
Noise predominates at F near zero. Implausible values are returns for x and y:



Remedy:

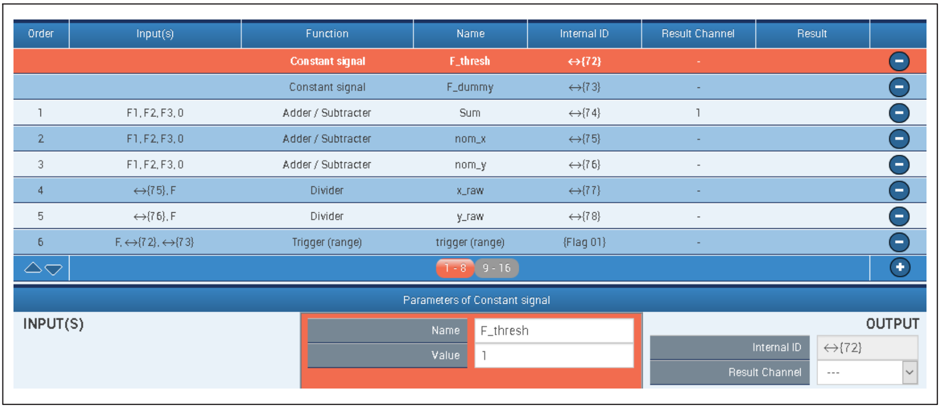
Only when F is greater than 1 N, for example, are x and y outputted regularly. Otherwise zero is returned each time.

A Trigger block sets the flag_01 if F is greater than the minimum value:



Order	Input(s)	Function	Name	Internal ID	Result Channel	Result
		Constant signal	F_thresh	↔{72}	-	
		Constant signal	F_dummy	↔{73}	-	
1	F1, F2, F3, 0	Adder / Subtracter	Sum	↔{74}	1	
2	F1, F2, F3, 0	Adder / Subtracter	nom_x	↔{75}	-	
3	F1, F2, F3, 0	Adder / Subtracter	nom_y	↔{76}	-	
4	↔{75}, F	Divider	x_raw	↔{77}	-	
5	↔{76}, F	Divider	y_raw	↔{78}	-	
6	F, ↔{72}, ↔{73}	Trigger (range)	trigger (range)	{Flag 01}	-	

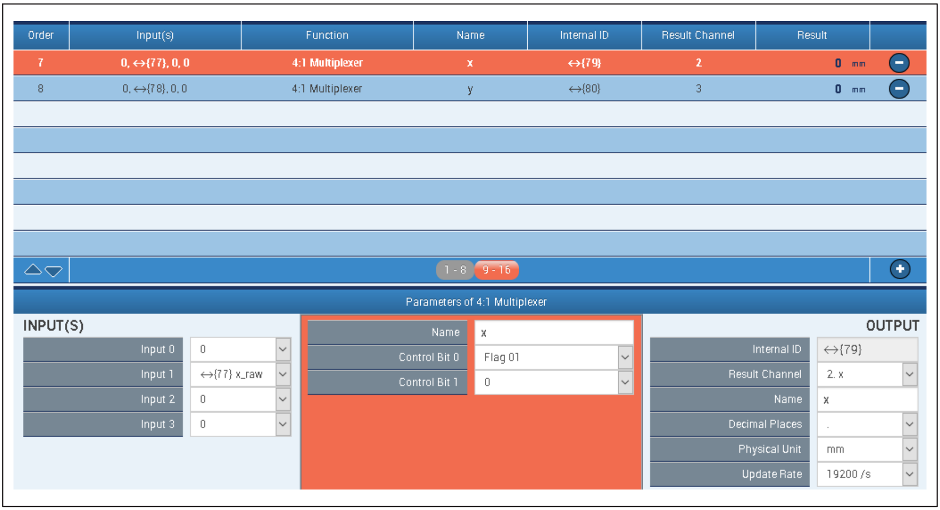
The two threshold values for the trigger. Only the lower switching threshold at 1 N is required. For the upper threshold, a value is selected that is well above the measuring range:



Order	Input(s)	Function	Name	Internal ID	Result Channel	Result
		Constant signal	F_thresh	↔{72}	-	
		Constant signal	F_dummy	↔{73}	-	
1	F1, F2, F3, 0	Adder / Subtracter	Sum	↔{74}	1	
2	F1, F2, F3, 0	Adder / Subtracter	nom_x	↔{75}	-	
3	F1, F2, F3, 0	Adder / Subtracter	nom_y	↔{76}	-	
4	↔{75}, F	Divider	x_raw	↔{77}	-	
5	↔{76}, F	Divider	y_raw	↔{78}	-	
6	F, ↔{72}, ↔{73}	Trigger (range)	trigger (range)	{Flag 01}	-	

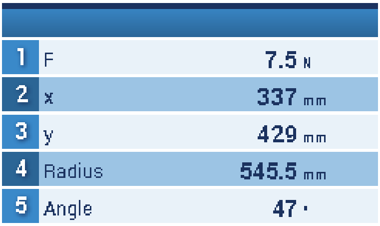
Two multiplexer blocks finally switch between zero and the calculated values.

Here for x:

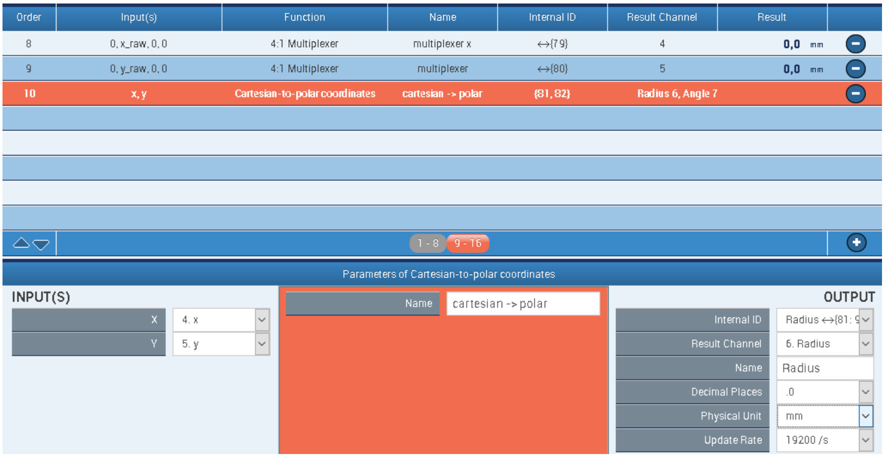


Tips

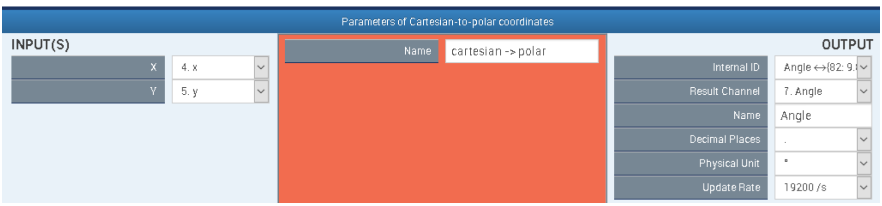
1. In the case of division by zero, a divisor returns Not a Number (NaN).
2. Polar coordinates can also be supplied if required:



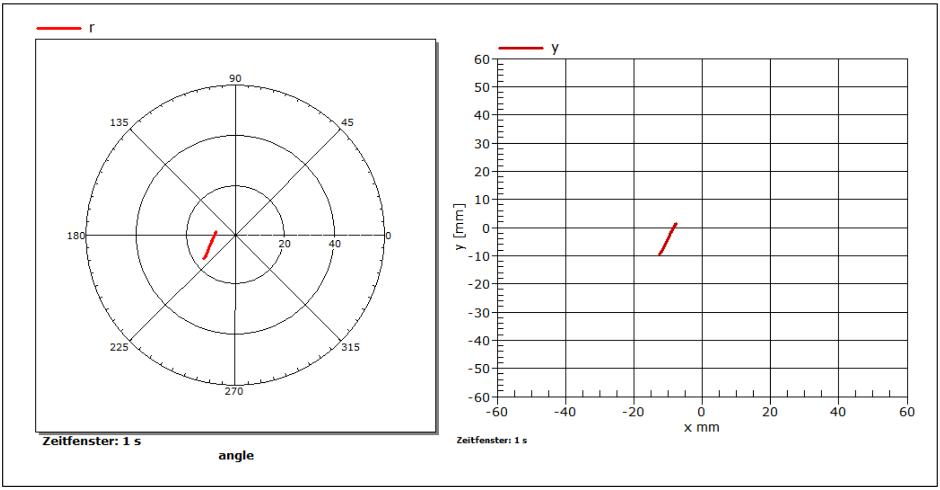
The settings for Radius ...:



... and angles:



Representation of a moving force with catman in polar and Cartesian coordinates:



13.3.3 Mechanical work via force-displacement integration

Short description

Force-displacement integration with PMX for measuring mechanical work

Introduction

The mechanical work done W is to be measured by integrating the force F over the displacement.

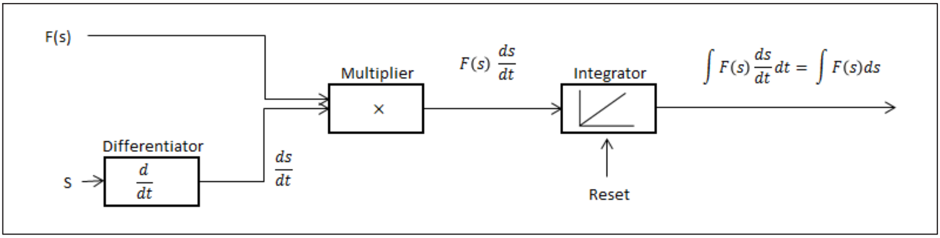
$$W = \int F(s) \cdot ds$$

(Because of the discrete-time processing, this is actually a summation, but the term integration is still used here.)

The start and end of the integration are determined by measurable events, e.g. fixed displacement or force values, or edges at a digital input.

Procedure

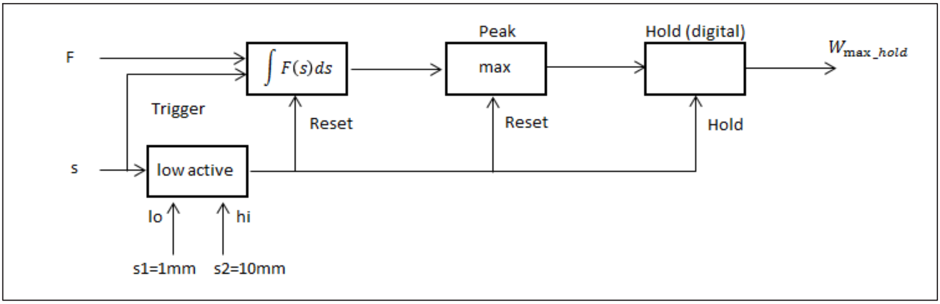
Integration via displacement is achieved by first deriving the displacement by time, multiplying it by F, and then integrating it over time again:



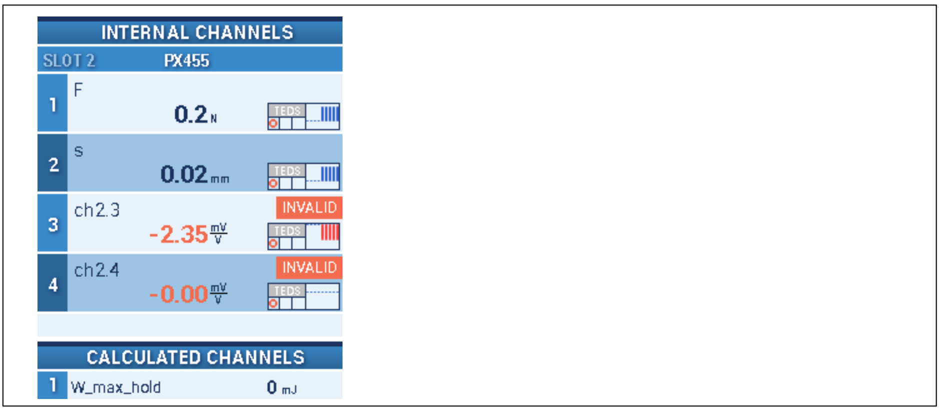
Example A

Integration via the **displacement from s1 to s2**: $W = \int_{s1}^{s2} F(s) \cdot ds$

Here the events s=s1 and s=2 determine the start and end of the integration. These way-points must be reached in any case during the process, otherwise the start or end will not be detected.



The overview with the measured quantities F and s as well as the calculated work



Overview of function blocks

Order	Input(s)	Function	Name	Internal ID	Result Channel	Result	
		Constant signal	s_low	↔{72}	-		
		Constant signal	s_high	↔{73}	-		
1	s	Differentiator	d_s/d_t	↔{74}	-		
2	s, ↔{72}, ↔{73}	Trigger (range)	gate	{Flag 01}	-		
3	F, ↔{74}, 1, 1	Multiplier	d_W	↔{75}	-		
4	↔{75}, Flag 01, 0	Integrator	W	↔{76}	-		
5	↔{78}	Hold (digital triggering)	W_max_hold	↔{77}	1	0 mJ	
6	↔{76}, 0, 0, Flag 01	Peak values	W_max	{78, 79, Flag 02}	-, -, -		
		1 - 8 9 - 16					

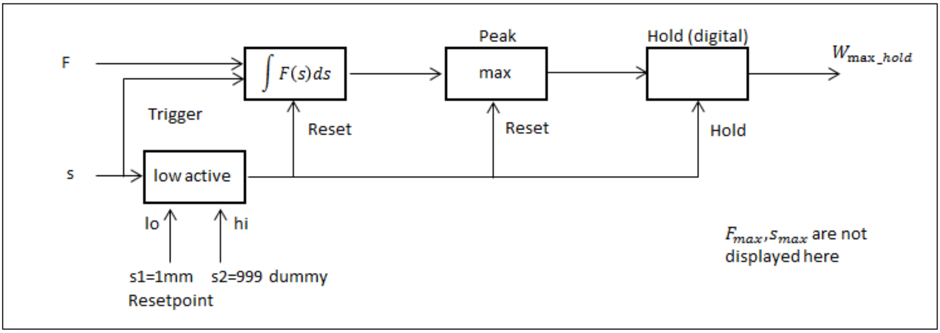
Example B

Measurement of the **maximum value** of the work.

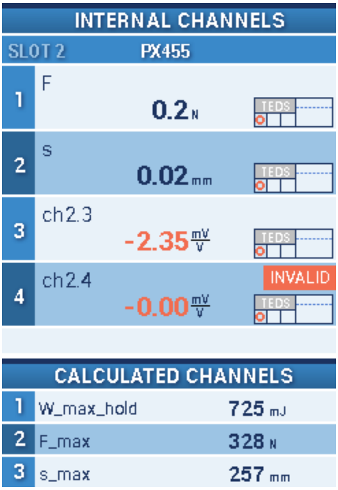
This is useful, for example, if the final value of the displacement is indeterminate, or cannot be reliably reached.

The integration starts and ends at the same way-point s=s1: $W = \oint_{s1} F(s) \cdot ds.$

The work can decrease again when the displacement is reversed, such as by relaxing a spring. The maximum value is recorded with a Peak block.



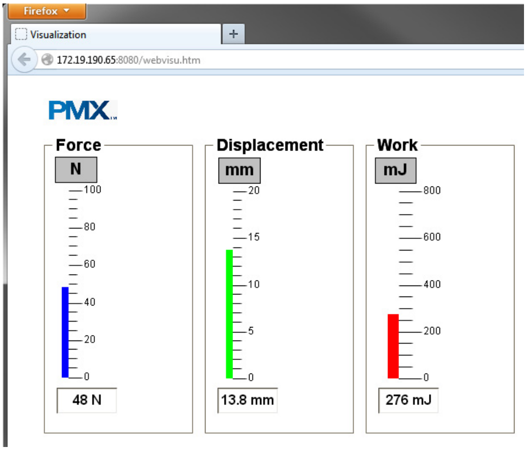
In addition to example A, the maximums of F and s are outputted here:



Overview of function blocks:

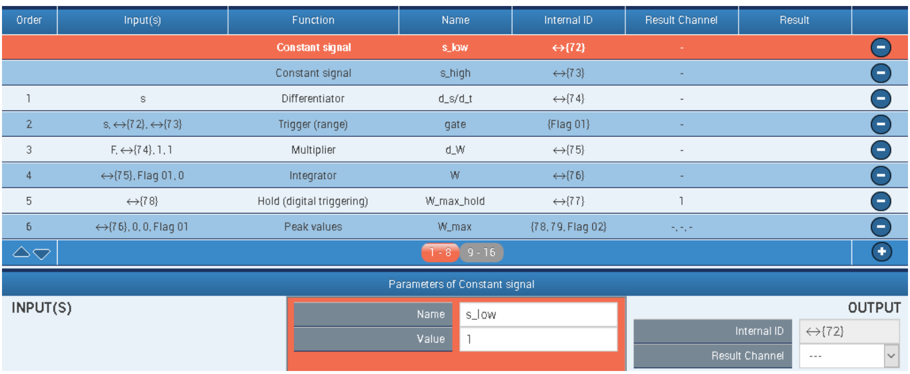
Order	Input(s)	Function	Name	Internal ID	Result Channel	Result
		Constant signal	s_low	↔{72}	-	
		Constant signal	s_high	↔{73}	-	
1	s	Differentiator	d_s/d_t	↔{74}	-	
2	s, ↔{72}, ↔{73}	Trigger (range)	gate	{Flag 01}	-	
3	F, ↔{74}, 1, 1	Multiplier	d_W	↔{75}	-	
4	↔{75}, Flag 01, 0	Integrator	W	↔{76}	-	
5	↔{78}	Hold (digital triggering)	W_max_hold	↔{77}	1	725 mJ
6	↔{76}, 0, 0, Flag 01	Peak values	W_max	{78, 79, Flag 02}	-, -, -	
		1 - 8 9 - 16				

Example of visualization with CODESYS WebVisu



Appendix

The function blocks of example A in detail:



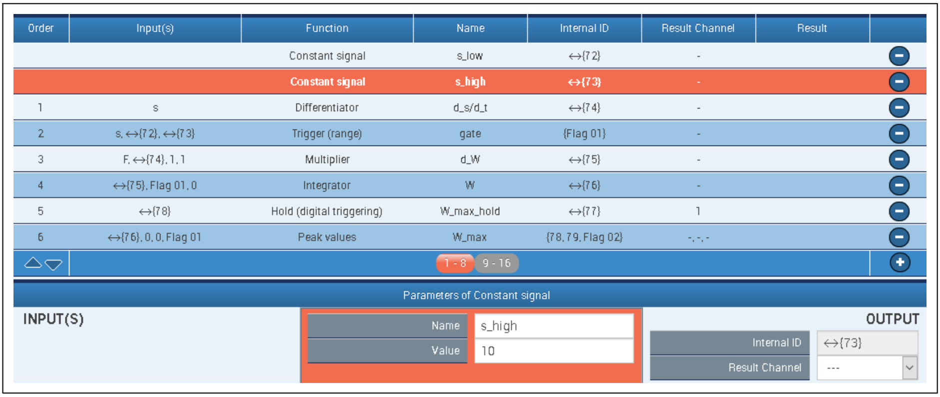
Order	Input(s)	Function	Name	Internal ID	Result Channel	Result
		Constant signal	s_low	↔{72}	-	
		Constant signal	s_high	↔{73}	-	
1	s	Differentiator	d_s/d_t	↔{74}	-	
2	s, ↔{72}, ↔{73}	Trigger (range)	gate	{Flag 01}	-	
3	F, ↔{74}, 1, 1	Multiplier	d_W	↔{75}	-	
4	↔{75}, Flag 01, 0	Integrator	W	↔{76}	-	
5	↔{78}	Hold (digital triggering)	W_max_hold	↔{77}	1	
6	↔{76}, 0, 0, Flag 01	Peak values	W_max	{78, 79, Flag 02}	-, -, -	

1 - 8 9 - 16

Parameters of Constant signal

INPUT(S)		
Name	s_low	
Value	1	

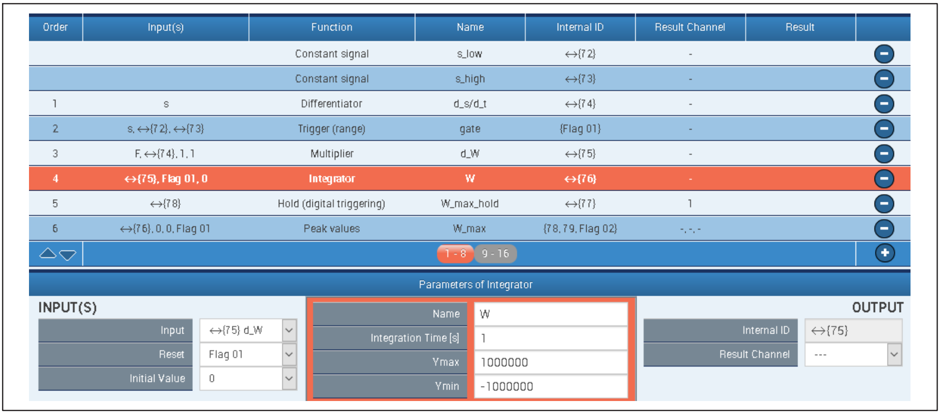
OUTPUT	
Internal ID	↔{72}
Result Channel	---


Order | Input(s) | Function | Name | Internal ID | Result Channel | Result
Constant signal | s_low | ↔{72} | -
Constant signal | s_high | ↔{73} | -
1 | s | Differentiator | d_s/d_t | ↔{74} | -
2 | s, ↔{72}, ↔{73} | Trigger (range) | gate | {Flag 01} | -
3 | F, ↔{74}, 1, 1 | Multiplier | d_W | ↔{75} | -
4 | ↔{75}, Flag 01, 0 | Integrator | W | ↔{76} | -
5 | ↔{78} | Hold (digital triggering) | W_max_hold | ↔{77} | 1
6 | ↔{76}, 0, 0, Flag 01 | Peak values | W_max | {78, 79, Flag 02} | -, -, -
1 - 8
9 - 16
Parameters of Constant signal
INPUT(S)
Name
s_high
Value
10
OUTPUT
Internal ID
↔{73}
Result Channel


Order | Input(s) | Function | Name | Internal ID | Result Channel | Result
Constant signal | s_low | ↔{72} | -
Constant signal | s_high | ↔{73} | -
1 | s | Differentiator | d_s/d_t | ↔{74} | -
2 | s, ↔{72}, ↔{73} | Trigger (range) | gate | {Flag 01} | -
3 | F, ↔{74}, 1, 1 | Multiplier | d_W | ↔{75} | -
4 | ↔{75}, Flag 01, 0 | Integrator | W | ↔{76} | -
5 | ↔{78} | Hold (digital triggering) | W_max_hold | ↔{77} | 1
6 | ↔{76}, 0, 0, Flag 01 | Peak values | W_max | {78, 79, Flag 02} | -, -, -
1 - 8
9 - 16
Parameters of Differentiator
INPUT(S)
Input
2. s
Name
d_s/d_t
Ymax
1000000
Ymin
-1000000
OUTPUT
Internal ID
↔{74}
Result Channel


Order | Input(s) | Function | Name | Internal ID | Result Channel | Result
Constant signal | s_low | ↔{72} | -
Constant signal | s_high | ↔{73} | -
1 | s | Differentiator | d_s/d_t | ↔{74} | -
2 | s, ↔{72}, ↔{73} | Trigger (range) | gate | {Flag 01} | -
3 | F, ↔{74}, 1, 1 | Multiplier | d_W | ↔{75} | -
4 | ↔{75}, Flag 01, 0 | Integrator | W | ↔{76} | -
5 | ↔{78} | Hold (digital triggering) | W_max_hold | ↔{77} | 1
6 | ↔{76}, 0, 0, Flag 01 | Peak values | W_max | {78, 79, Flag 02} | -, -, -
1 - 8
9 - 16
Parameters of Trigger (range)
INPUT(S)
Input
2. s
Threshold high ↑
↔{73} s_high
Threshold low ↓
↔{72} s_low
Name
gate
Hysteresis
0.1
Delay [ms]
0
Active
Low
OUTPUT
Flag
Flag 01

Order	Input(s)	Function	Name	Internal ID	Result Channel	Result
		Constant signal	s_low	↔{72}	-	
		Constant signal	s_high	↔{73}	-	
1	s	Differentiator	d_s/d_t	↔{74}	-	
2	s, ↔{72}, ↔{73}	Trigger (range)	gate	{Flag 01}	-	
3	F, ↔{74}, 1, 1	Multiplier	d_W	↔{75}	-	
4	↔{75}, Flag 01, 0	Integrator	W	↔{76}	-	
5	↔{78}	Hold (digital triggering)	W_max_hold	↔{77}	1	
6	↔{76}, 0, 0, Flag 01	Peak values	W_max	{78, 79, Flag 02}	-, -, -	



Order	Input(s)	Function	Name	Internal ID	Result Channel	Result
		Constant signal	s_low	↔{72}	-	
		Constant signal	s_high	↔{73}	-	
1	s	Differentiator	d_s/d_t	↔{74}	-	
2	s, ↔{72}, ↔{73}	Trigger (range)	gate	{Flag 01}	-	
3	F, ↔{74}, 1, 1	Multiplier	d_W	↔{75}	-	
4	↔{75}, Flag 01, 0	Integrator	W	↔{76}	-	
5	↔{78}	Hold (digital triggering)	W_max_hold	↔{77}	1	
6	↔{76}, 0, 0, Flag 01	Peak values	W_max	{78, 79, Flag 02}	-, -, -	

Note that the Hold block comes *before* the Peak block in the calculation sequence. So the edge of Flag_01 first triggers the Hold function and then resetting of the Peak block.

Tip

In example A, the Hold block is already reset when the displacement is reversed. If the value is to be held longer, the block could be reset such as via a digital signal from an external source or by another Trigger block.

13.3.4 Checking the force at specific points on the displacement axis

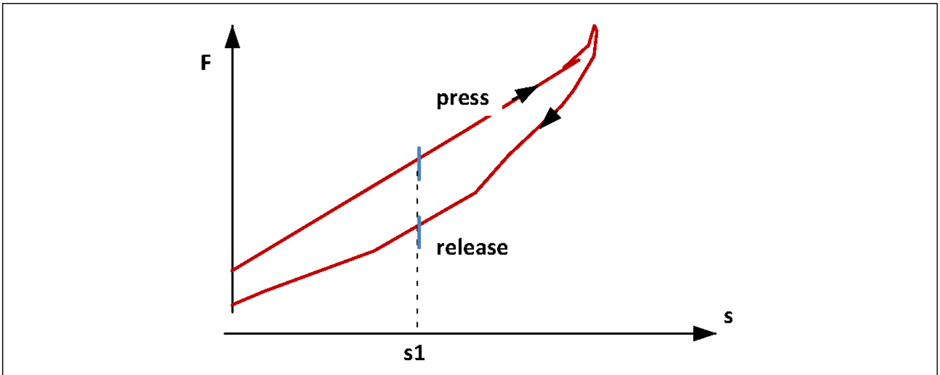
Short description

In a stroke movement with force and displacement measurement, the force is measured at a specific point on the displacement axis. The measured force should be within an acceptance band; a good/bad decision is made.

The force is checked once on the way forward and once on the way back.

Introduction

In this example, the force behaves roughly as follows:



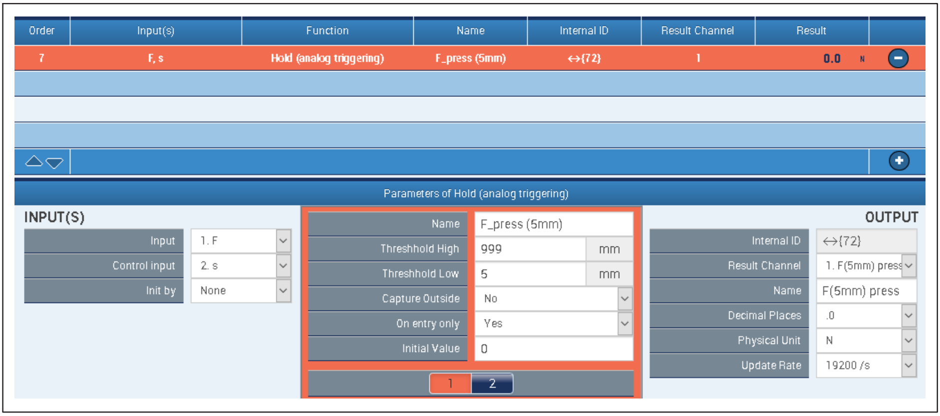
Two "Hold (analog)" function blocks hold the measured force value at point s1. One block on the way forward, the other on the way back. Two limit switches check whether the values are within acceptance bands. In the following the rising force is referred to as "press", and the falling force as "release".

Two digital outputs indicate the results of the limit switches.

Procedure

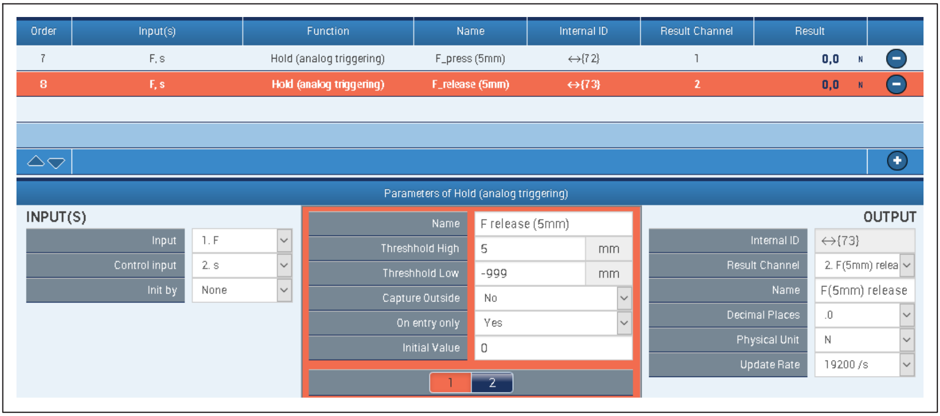
With rising "press" displacement:

- The measured force value F is held at point s1 = 5 mm with a "Hold (analog triggering)" function block. The displacement s controls the hold function, so it is applied to the "control input".
- The function block reads the force F at s = 5 mm ("Lower limit") and holds it at the output.
- The upper value of the holding range ("Upper limit") is not required here, and assumes a dummy value outside the measuring range.
- The "On entry only" setting means that the force is read and held exactly once each time the force enters the holding range (5 mm ... 999 mm).



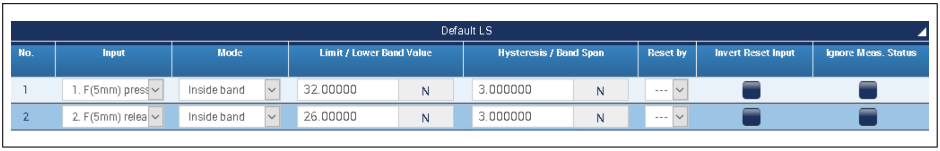
With falling "release" displacement:

- The holding range now covers the range (-999 mm ... 5 mm); -999 is again a dummy value here. The block holds the measured force value when the displacement s enters the holding range at 5 mm *from above*.

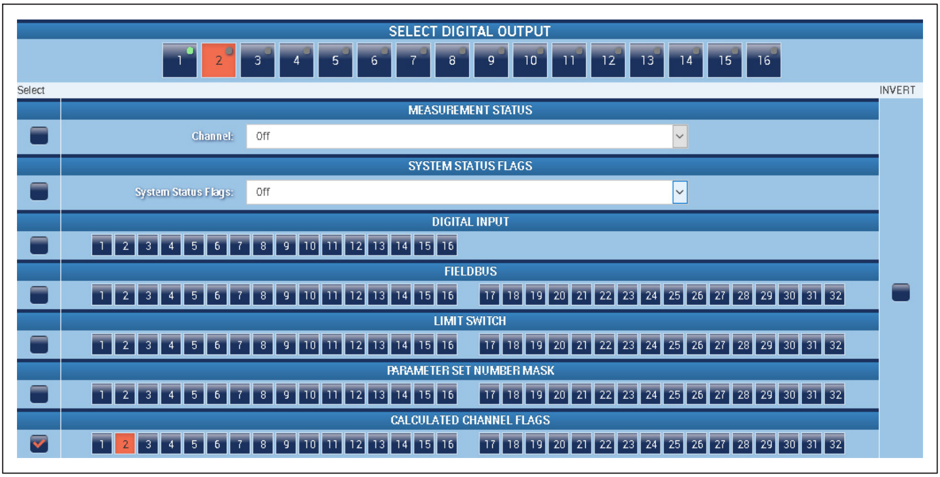


Evaluation by limit switches:

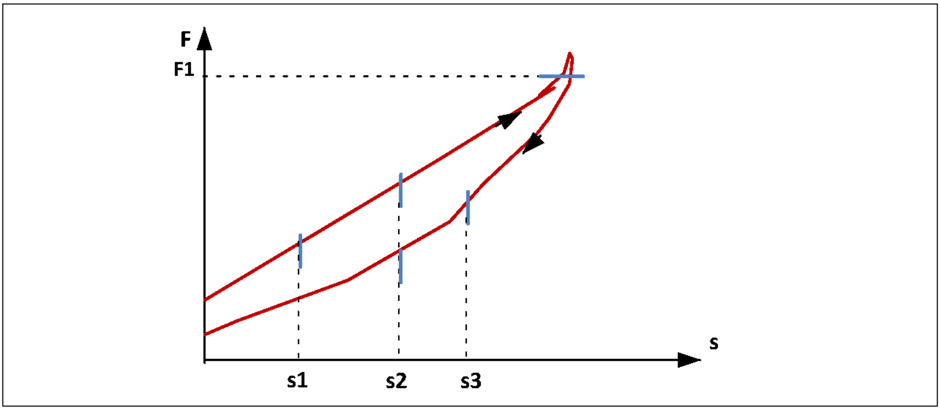
- The outputs of the Hold blocks are the inputs of two limit switches.
- Switch #1 signals the correct force value as the displacement increases. It is active when the signal "F (5 mm) press" is in the band 32 N ... 35 N.
- The settings for switch #2 apply in the same way; the acceptance band is 26 N ... 29 N.



Finally, digital outputs #1 and #2 output the states of limit switches #1 and #2:



Tips



- A violation of the acceptance band (instead of adherence to it) can be indicated by reversing the logic, for example at the limit switches.
- Instead of checking the force at points on the displacement axis, the displacement at specific force values can of course also be checked.
- With a corresponding number of Hold blocks, even more direction-dependent check points can be set.
- Parameter sets can be used to switch the values of the measuring points and/or the acceptance bands to different values. To do this, a copy of the "Data Acquisition" sub-parameter set is created in which the numerical values are changed. The sub-parameter sets are then assigned to different parameter sets, which are switched via fieldbus or digital inputs for example.
  The structure of the calculated channels can thus also be switched instead of individual numerical values.
  The limit switches are switched via the "Limit Switches" sub-parameter set.
- The task can also be performed with "Trigger (pulse)" and "Hold (digital)" blocks.

13.3.5 Force-displacement measurement with relative zero

Short description

The force F and the displacement s are measured in a stroke movement. To compensate for random offsets in the measured quantities, new offset-free signals F_tared and s_tared are generated.

Case A

The force is zeroed at the point on the displacement axis s = s0.

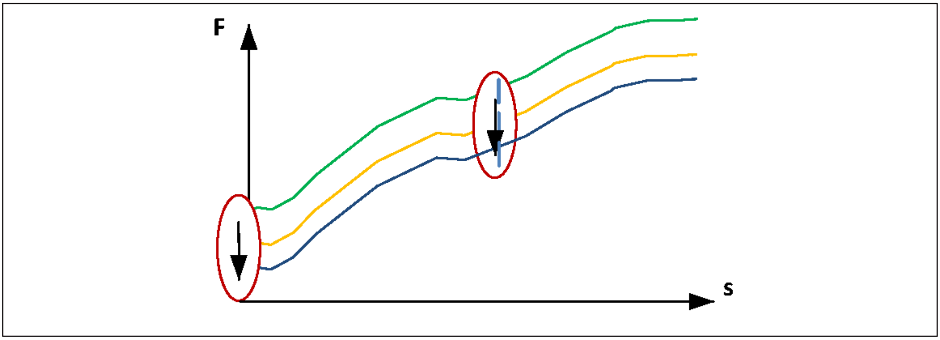
Case B

The force and displacement are zeroed at force F = F0.

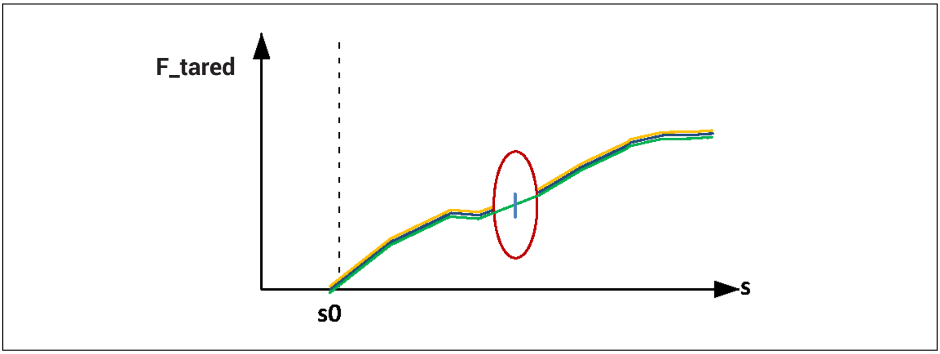
Introduction

Case A

When a displacement-dependent force F is checked for adherence to limit values, a force offset from stroke to stroke is a disturbance. The force limits would have to be adjusted for each stroke movement.

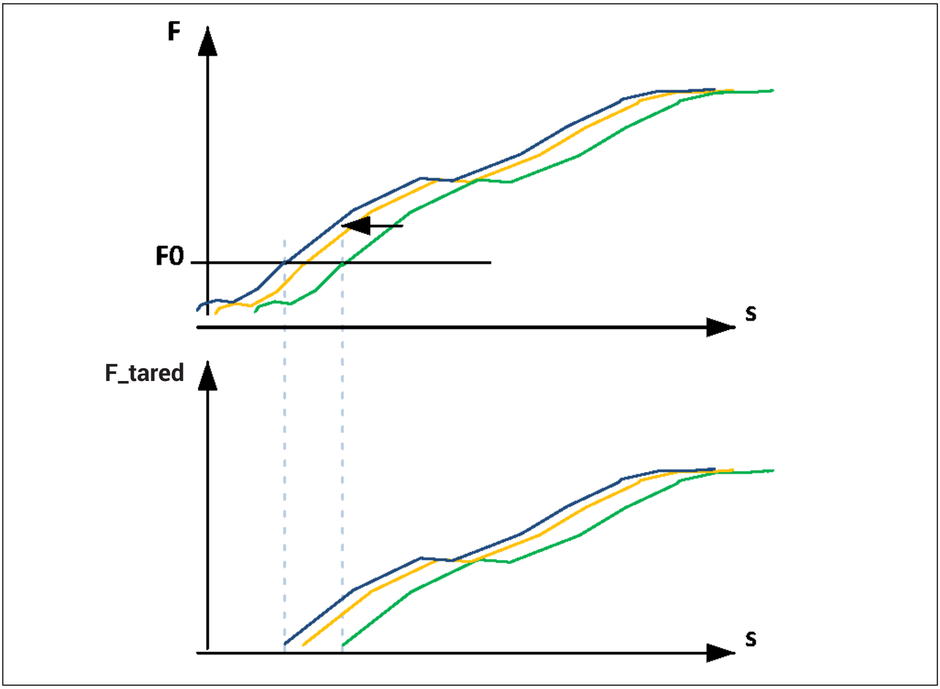


Two function blocks "Trigger (range)" and "Tare" are used to zero the force below the point s0. The force characteristic below s0 is irrelevant for zeroing.

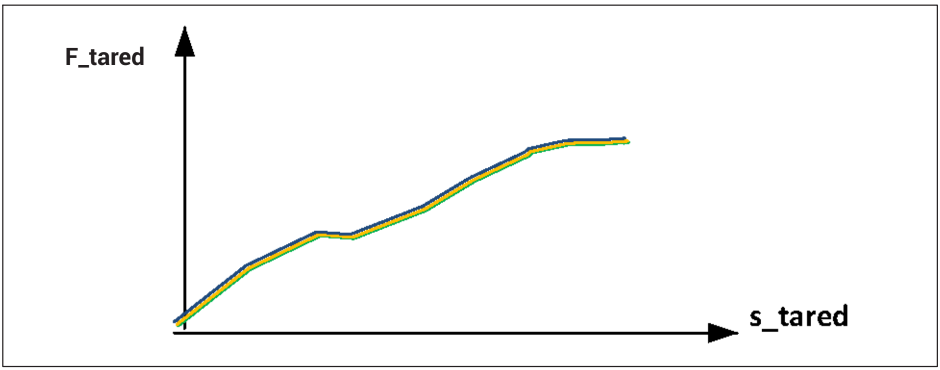


Case B

To compensate for a displacement offset, the new force zero point is to be determined not by the displacement, but by the force F0.



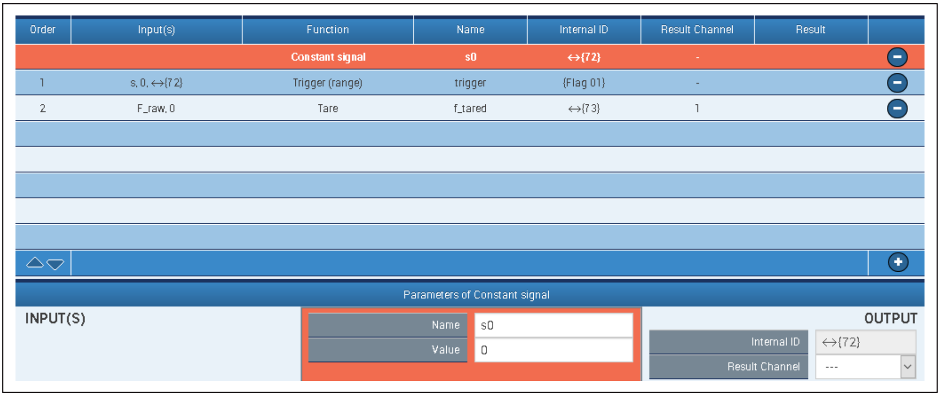
According to case A, the displacement zero point can also be immediately shifted. The same condition (F = F0) is used for the purpose. Both quantities - force and displacement - now have a new zero point.



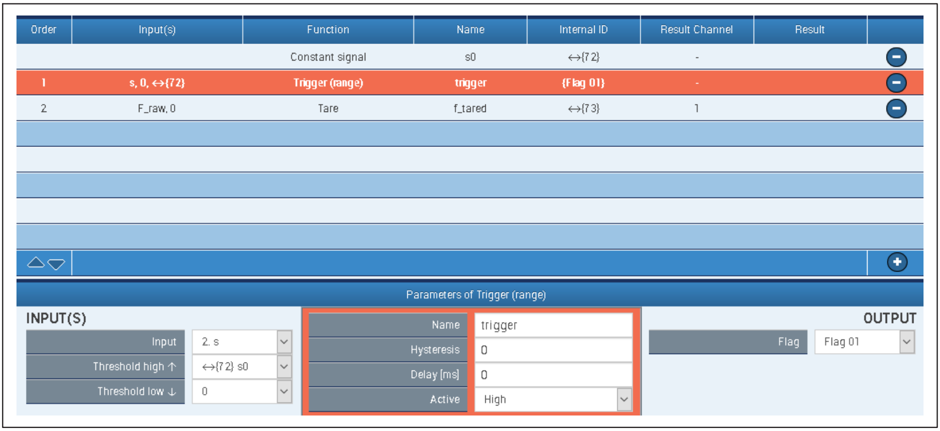
Procedure

Case A

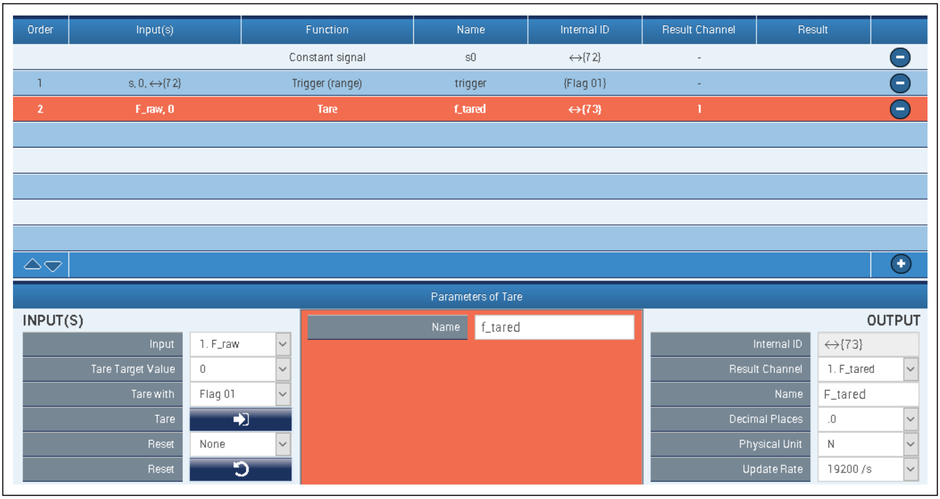
s0 delivers a constant signal, here 5 mm:



A Trigger block sets the "Flag01" output if the displacement s is in the range 0 ... 5 mm:

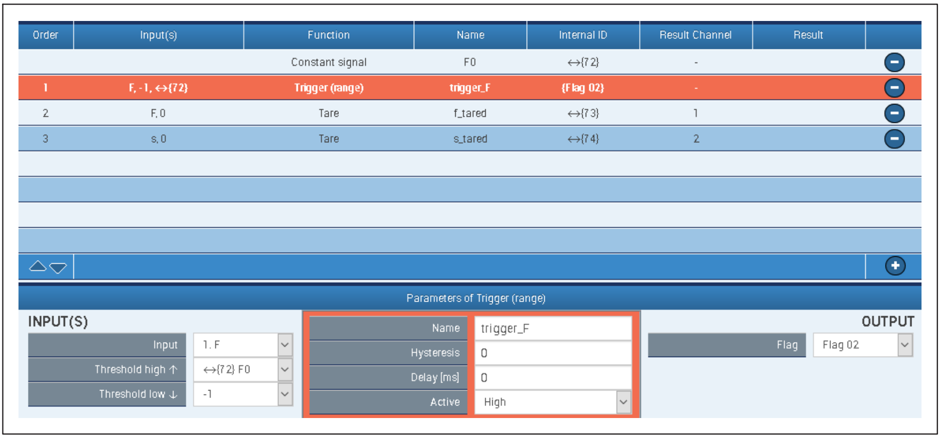


Finally, the Trigger block zeros the raw value of the force F_raw below 5 mm. F_tared is the offset-adjusted force value:

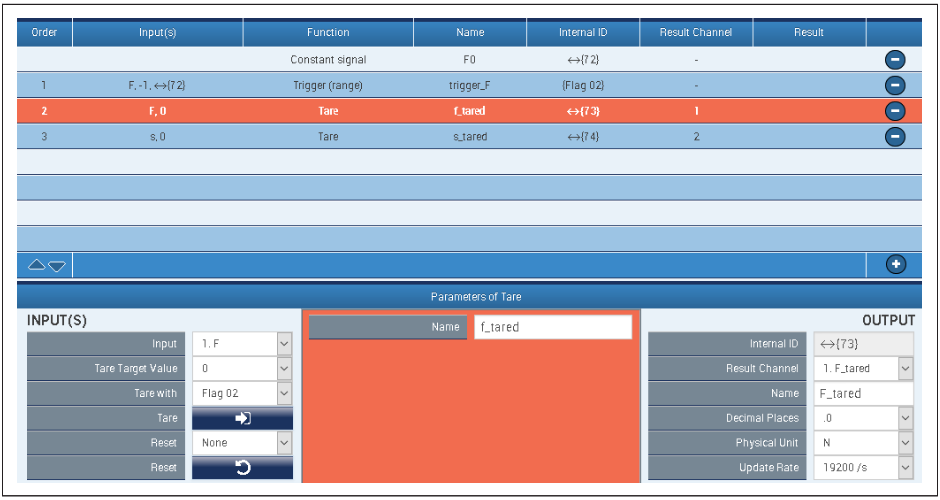


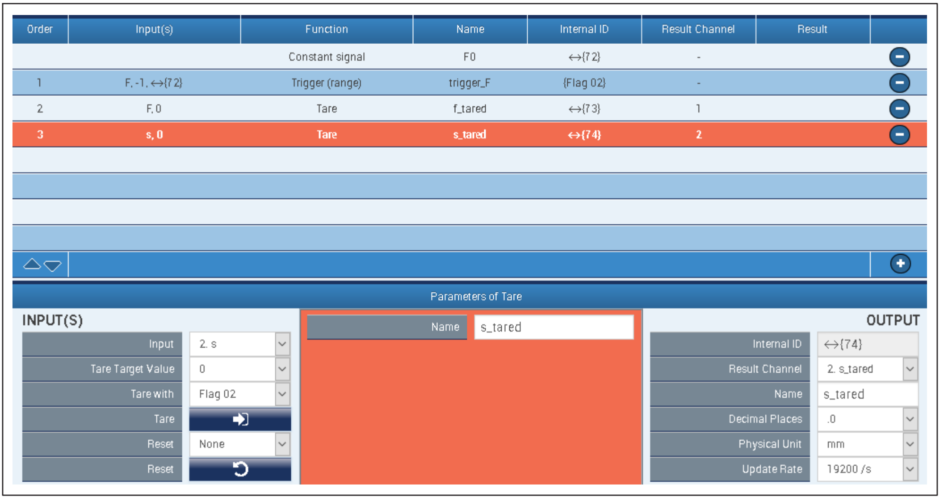
Case B

The Trigger block now checks the force F against F0.



The force and displacement are each zeroed with the "Flag02" signal:





13.3.6 Checking the force with a tolerance band

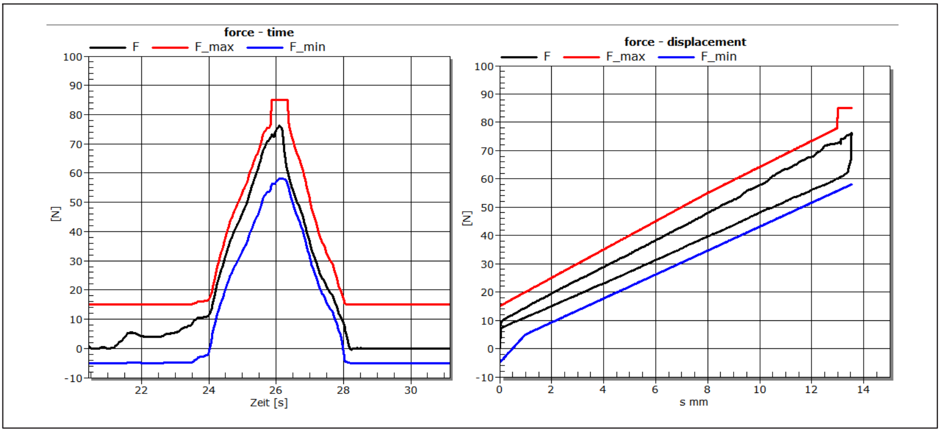
Short description

In a stroke movement with force and displacement measurement, the force is continuously checked against a tolerance band. The tolerance limits are defined as displacement-dependent. A counter counts the tolerance violations, and a digital output signals when the counter reading is greater than zero. The counter reading is automatically deleted at the beginning of each stroke movement.

(The required function blocks are available as from firmware version 1.34).

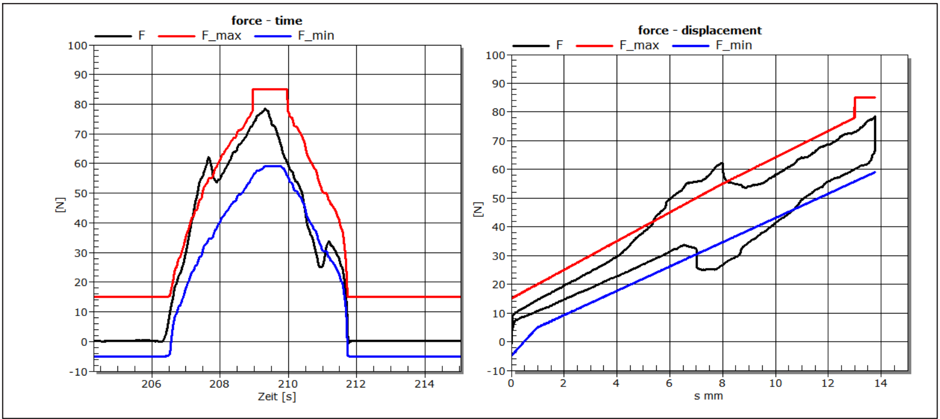
Introduction

The measured force F is checked against an upper and lower tolerance (F_max and F_min). Two value tables provide the tolerance values depending on the measured displacement s. The figure shows the force/time and force/displacement curves of a stroke movement with no violation of the tolerance values.



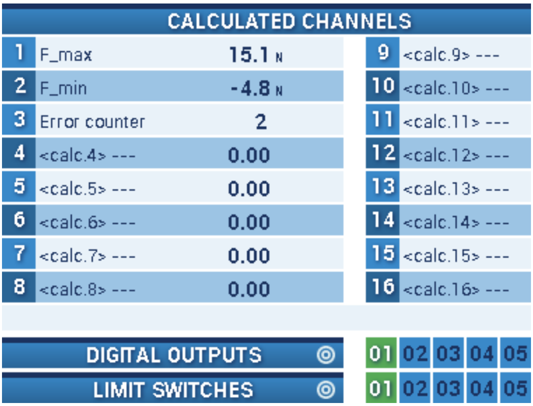
A stroke movement with no tolerance violation

A "Trigger (range)" function block compares the force F with the tolerances. A tolerance violation is indicated by Flag_01 and counted by a Counter block.



Double tolerance violation

Overview after double tolerance violation. The digital output 01 signals the error.

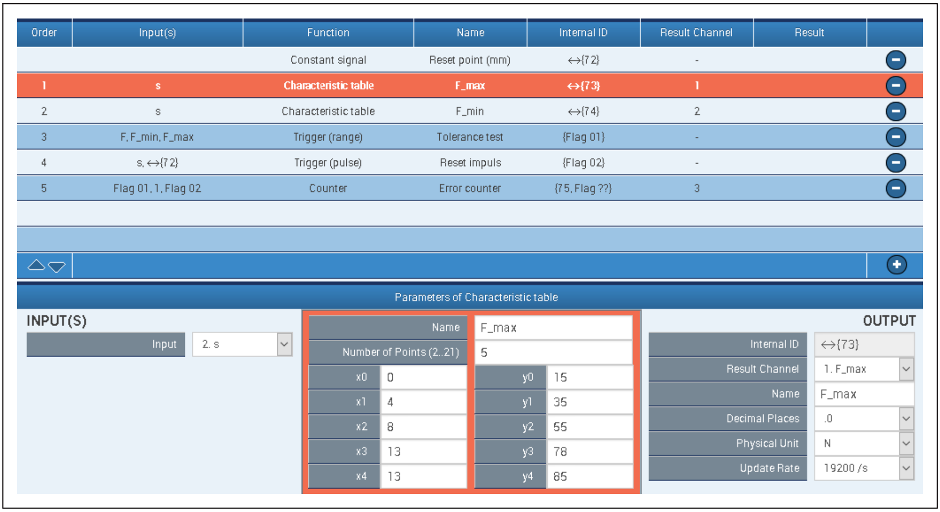


Procedure

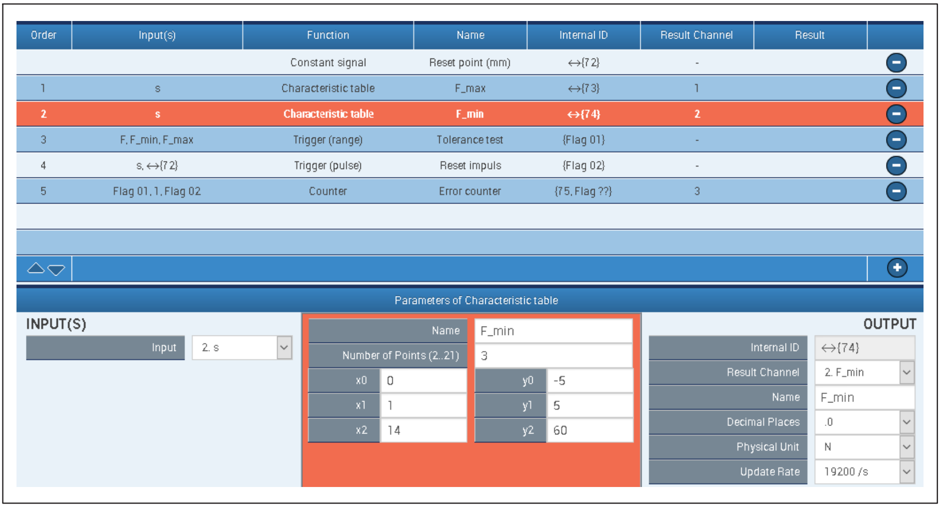
The table for the upper tolerance line has five interpolation points. The x values are the displacement in mm; the y values are the tolerance limit of the force in N.

Tip

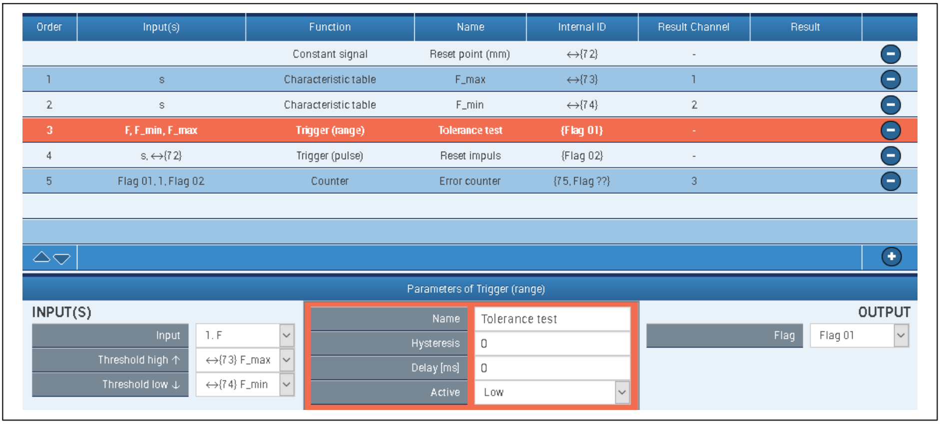
Interpolation points with the same x values cause a jump in the characteristic curve, here at x3 = x4 = 13.



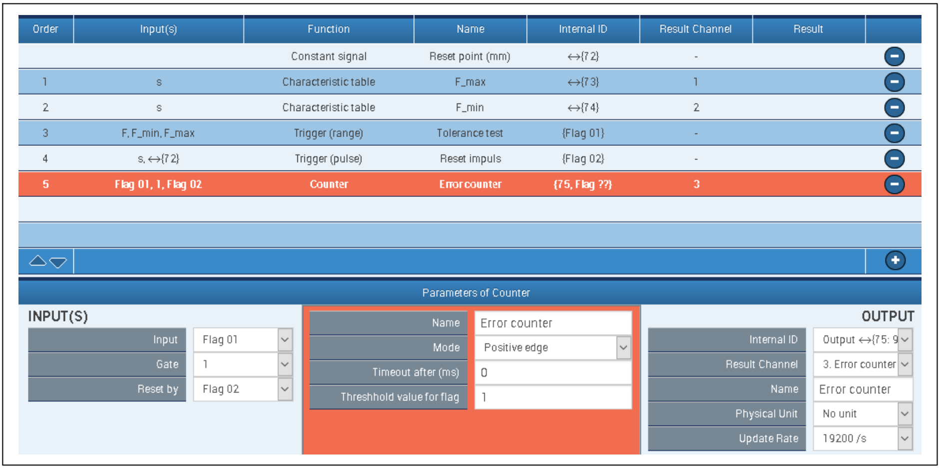
The table for the lower tolerance line:



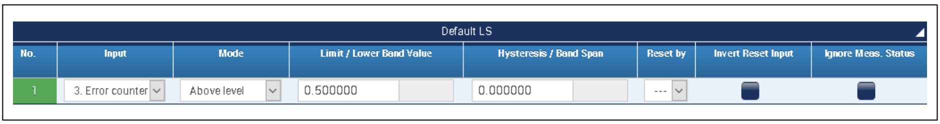
The Trigger block compares F with the tolerance limits. Within the tolerance the "Flag_01" output is Low.



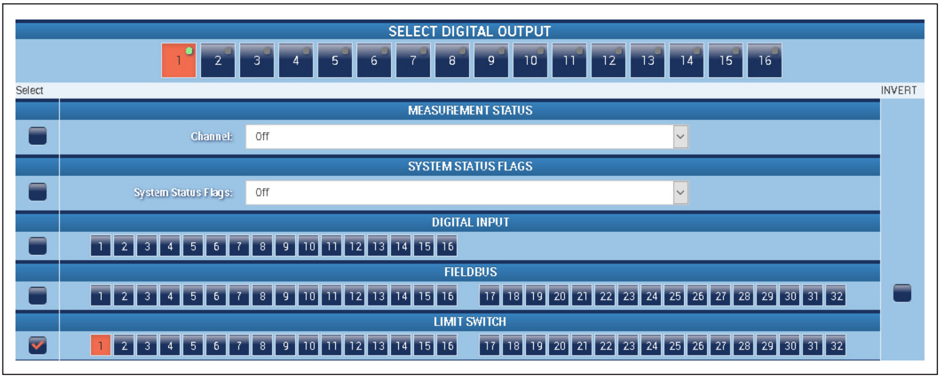
The "Counter" block counts the tolerance violations. It is cleared with Flag_02:



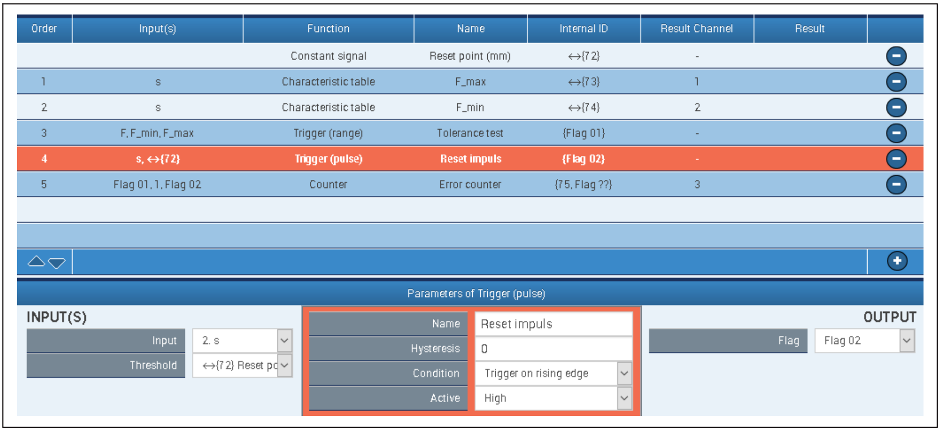
Limit switch #1 detects counter readings ≥ 1:

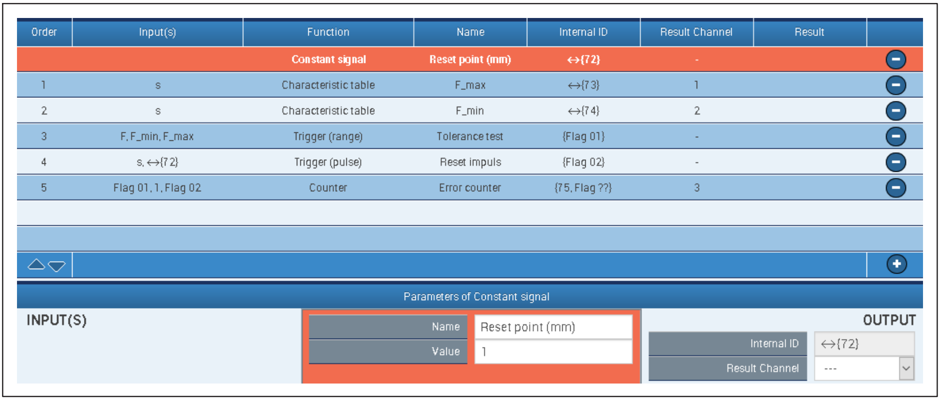


Finally, digital output #1 outputs the state of limit switch #1:



The "Trigger (pulse)" block provides the pulse to clear the counter when the displacement exceeds 1 mm.



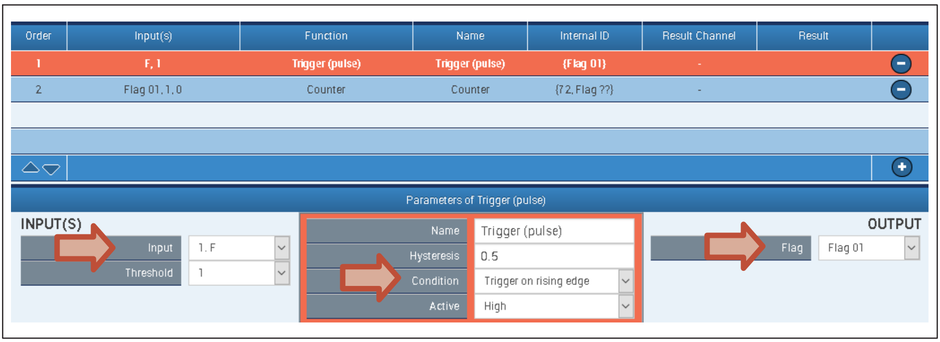


13.3.7 Event counter

The following shows how to program an event counter in PMX. Two calculation channels are required.

Create "Trigger (pulse)" calculation channel

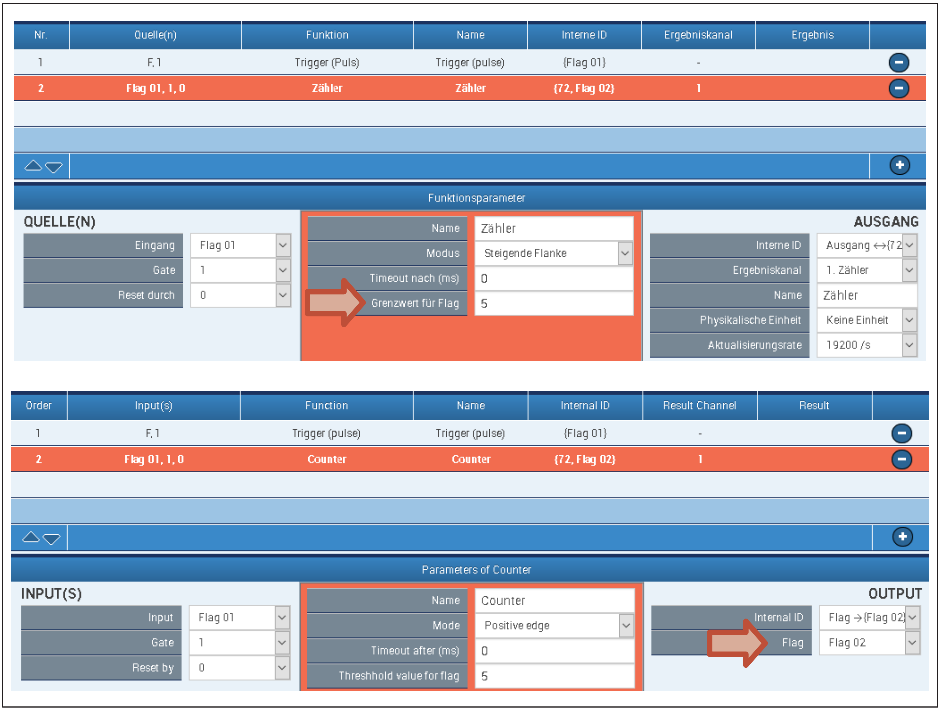
- The input here is the measured force value.
- A predefined value can be selected as the threshold.
- Condition here when the threshold value is "exceeded".
- The result is stored here in "Flag01".



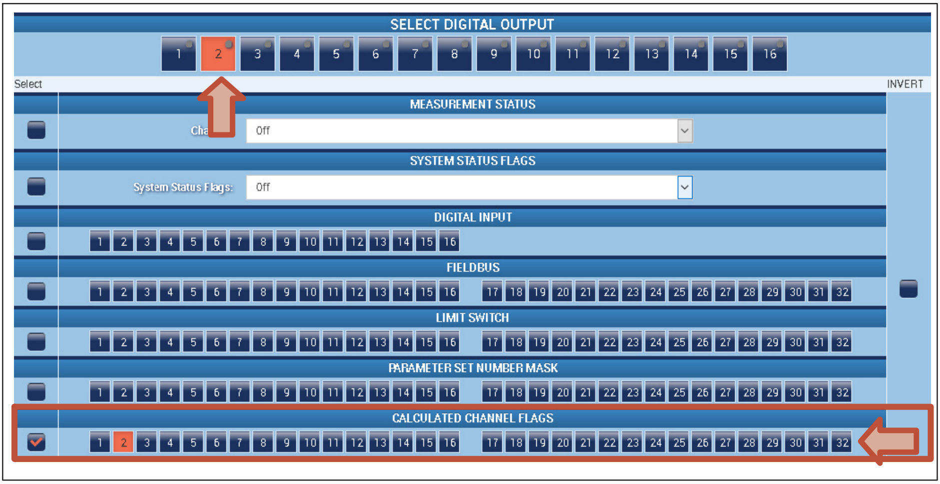
Create "Counter" calculation channel

- The input here is now Flag01.
- The count is performed on a rising edge.
- By way of "Threshold value for flag" a signal can be generated that becomes active when this counter value is reached.

- Here Flag02 is activated when the counter reading reaches 5 and above.
- These flags can be interrogated directly by the PLC over the fieldbus.

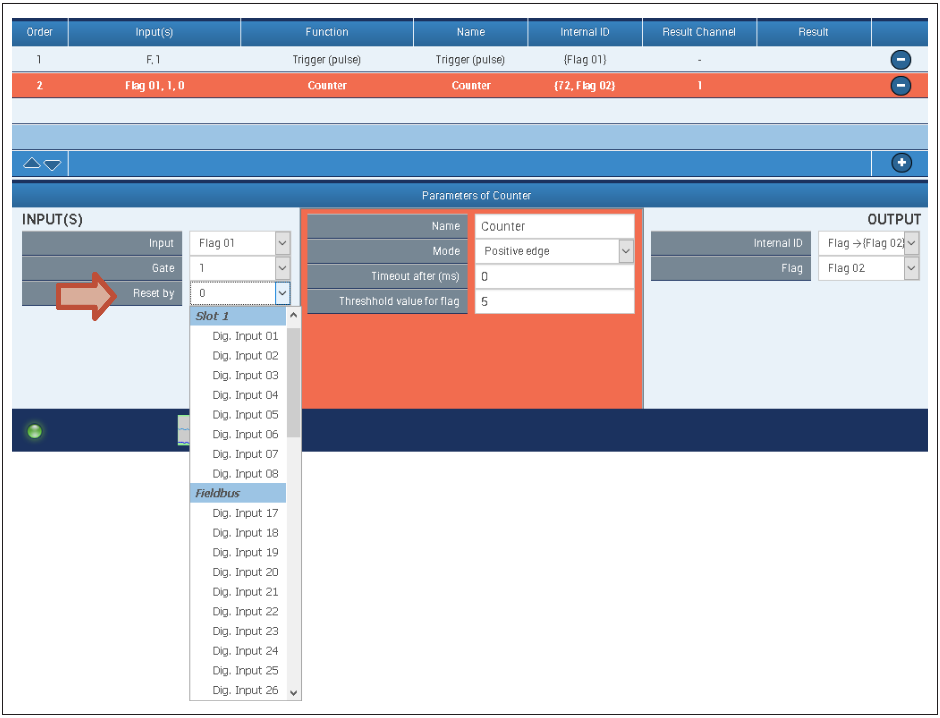


- If the result (the counter reading) is to be signaled at a digital output (PX878), this flag is selected as the input signal for a digital output. Here digital output #2 is activated by Flag02.



The event counter can be reset in the following ways:

- Digital input (PX878).
- By fieldbus control word, mapped by digital inputs 17 ... 32 in PMX.
- By other flags from other PMX calculation channels.


Order
Input(s)
Function
Name
Internal ID
Result Channel
Result
1
F, 1
Trigger (pulse)
Trigger (pulse)
{Flag 01}
-
2
Flag 01, 1, 0
Counter
Counter
{72, Flag 02}
1
Parameters of Counter
INPUT(S)
Input
Flag 01
Gate
1
Reset by
0
Slot 1
Dig. Input 01
Dig. Input 02
Dig. Input 03
Dig. Input 04
Dig. Input 05
Dig. Input 06
Dig. Input 07
Dig. Input 08
Fieldbus
Dig. Input 17
Dig. Input 18
Dig. Input 19
Dig. Input 20
Dig. Input 21
Dig. Input 22
Dig. Input 23
Dig. Input 24
Dig. Input 25
Dig. Input 26
Name
Counter
Mode
Positive edge
Timeout after (ms)
0
Threshhold value for flag
5
OUTPUT
Internal ID
Flag → {Flag 02}
Flag
Flag 02

14 TEST SIGNALS AND SIGNAL GENERATORS

Test signals

With the PMX you can generate and output various signals. This can be a test signal during the starting-up phase, to simulate measured values and so already test the functioning of the components. As long as a test signal is active, this is displayed in the PMX browser and also transmitted on the fieldbus as a status.

Signal generators

The PMX has internal signal generators that can be set up using the "Signal generators" calculation channel. The following functions are available in the function parameters:

sine, rectangle, white noise, counter, constant and triangle

Other parameters include:

frequency, amplitude and offset

You can activate the signal generator with an On switch. You define the number of repetitions by specifying periods.

As well as conventional waveforms, the PMX also has the option to define a separate function by specifying up to 21 points. These points are interconnected along a line. You control this "test profile" by a ramp (timer) for example.

To create such a ramp, create the new calculation channel "Timer" in the "Technology" category. In its function parameters you define the period duration (the interval of the ramp), and whether it is to be outputted continuously or once.

You can generate additional signal shapes using a CODESYS or .NET API application.

Signal output

The generated signals can also be outputted via the existing interfaces in the PMX. Please note the maximum output rates for each medium.

Signal form	PMX signal generators	CODESYS	LabVIEW .NET API
Rectangle, triangle, sine, noise	x	x	x
21-point characteristic	x	x	x
Free signal form	-	x	x

	PMX signal generators	CODESYS	LabVIEW .NET API
Output rate (max.):			
PX878 (±10 V)	19.2 kHz	2.4 kHz	10 Hz
Fieldbus	1 … 9.6 kHz	1 … 9.6 kHz	1 … 9.6 kHz
Ethernet	19.2 kHz	2.4 kHz	19.2 kHz

Tip

Practical examples of data storage can be found in the PMX TechNotes at https://www.hbm.com/de/2981/pmx-modular-measuring-amplifier-system-for-the-iot/.

15 PARAMETER SETS (FORMULAS)

The PMX has over 100 independent parameter sets (measurement programs) which are stored power failsafe in the device's internal flash memory. This allows you to define measuring tasks or test sections that can be used later during operation without additional resetting times. You can configure the active or a non-active parameter set (EDIT MODE). The switch can be made via digital inputs, fieldbus, or also via Ethernet, i.e. PMX command set, .NET API, LabVIEW driver or CODESYS application.

Important

The active parameter set should not be configured during operation. To avoid malfunctions, parameter sets may only be switched after the measuring or testing task in question has been completed. In all cases, status bits, or items of status information, are available to signal error-free switching (digital outputs, system status, cyclic device data in fieldbus operation).

A parameter set always consists of the four sub-parameter sets:

- Sensor data
- Data acquisition (also includes the calculated channels)
- Thresholds
- Digital outputs

A parameter set is composed of the individual sub-parameter sets. The settings are in the sub-parameter sets. If a sub-parameter set is not to be changed, define "ignore" for it. That component is then not changed when switching.

Depending on which components are switched in a parameter set, the following switching times result:

Sensor data	Data acquisition	Thresholds	Digital outputs	Average switching time in ms
1200	-	-	-	1200
-	950	-	-	950
1200	950	-	-	2150
-	-	100	-	100
1200	950	100	-	2250
	-	-	80	80
1200	950	100	80	2330

Important

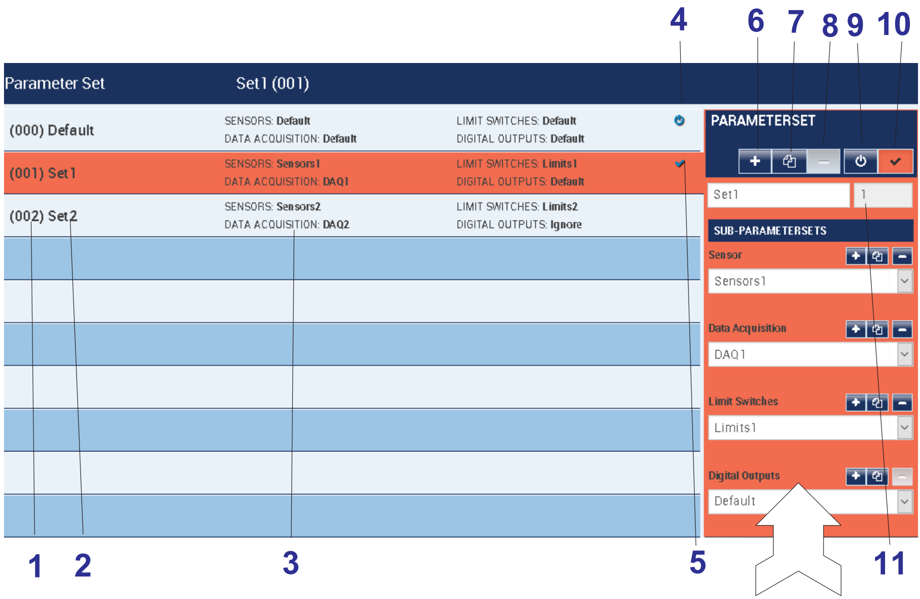
If the power supply fails while a parameter set is being saved, the parameter set will be destroyed, and the PMX will restart with its factory settings after being switched on. To avoid this, we recommend backing up the device settings to PC.

15.1 Configuring parameter sets

Follow these steps:

- Create a master parameter set.
- Open one of the sub-parameter sets and change the desired parameters.
- Finally, save the parameter set(s) in the PMX power failsafe by clicking on the floppy disk icon in the status bar.

Administration of parameter sets



Create, copy, assign, delete sub-parameter sets

1 Unique parameter set index

2 Parameter set name, freely selectable

3 The parameter set is comprised of these sub-parameter sets

4 Active now

5 Used when booting

6 Create parameter set with factory settings

7 Clone selected parameter set

8 Delete selected parameter set

9 Make boot parameter set

10 Make active parameter set

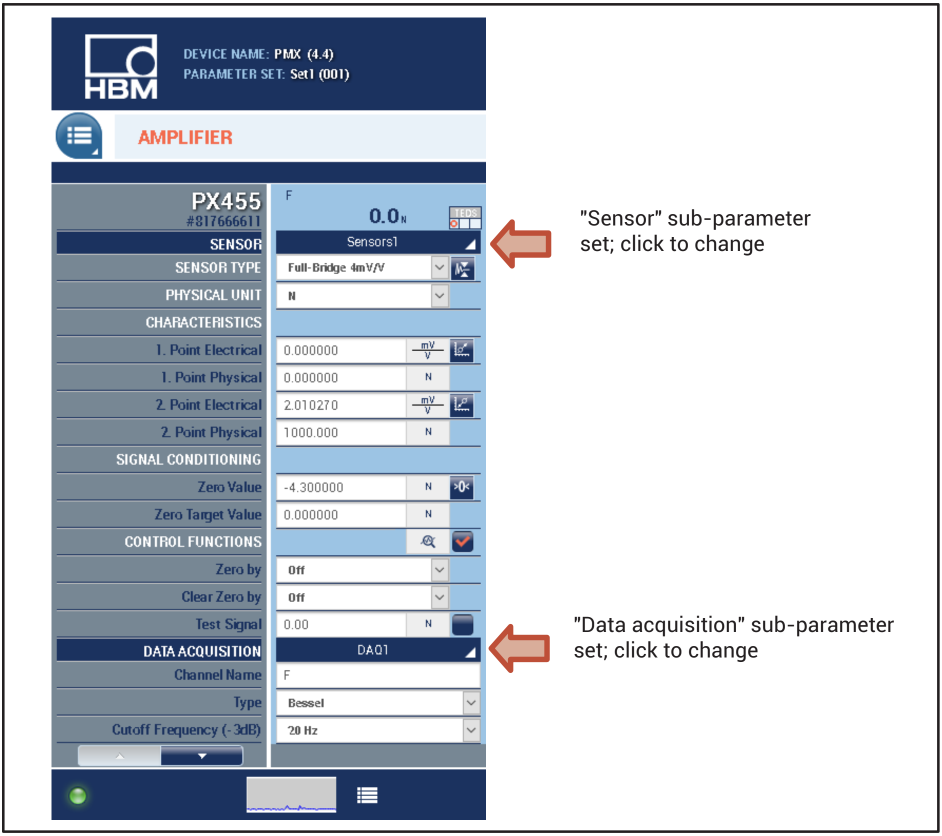
11 Parameter set index

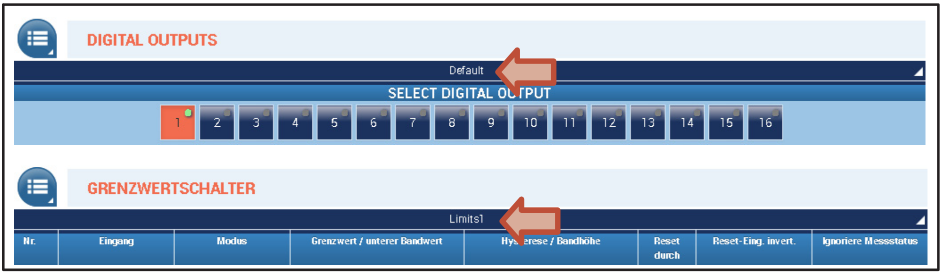
Important

Parameter sets can be copied, deleted and combined in the "Parameter set" dialog. The specific settings for sub-parameter sets are made in the dialogs for the measurement cards.

15.2 Changing from parameters to parameter sets

To change sub-parameter sets, click on the relevant buttons and select the desired sub-parameter set. Then make the changes.





One of the sub-parameter sets is now not active.
Settings can be edited, but they do not take effect immediately. They take effect when a parameter set is activated that includes this sub-parameter set.

15.3 Saving and loading measurement programs (parameter sets)

Save to PMX

All settings you make on the device take effect immediately, even without being saved. However, saving your settings protects you against data loss if the device is turned off. To save, click on the disk icon at the bottom right in the web browser.

Save to and load from PC

The **Backup to PC** menu item is used to create an XML data record that you can use as a backup or transfer to other devices fitted with the same measurement and I/O cards.

The corresponding upload command in the PMX browser is **Restore from PC** to load the parameter set file back to PMX.

Important

Passwords and network settings are not changed with this method (see also section 25.6, page 437).
Parameter sets are not saved as individual files. They are mapped for the entire device in the XML data set. Offline configuration of the device is not possible. It is theoretically possible to edit parameter sets in the XML backup file, but we advise against it.

Example:

The table shows the assignment of parameter sets to sub-parameter sets:

Parameter set	Sensor	Data acquisition	Thresholds	Digital outputs
000 Factory setting	Default	Default	Default	Default
001 Workpiece A	Default	Fast filter	Thresholds - workpiece A	Ignore
002 Workpiece B	Default	Slow filter	Thresholds - workpiece B	Ignore

Switching from parameter sets 000 to 001 has the following effect:

Sensor Default -> Default	No change, but after switching the "Sensor Default" settings are guaranteed to be used.
Data acquisition Default -> "Fast filter"	The settings in "Fast filter" are activated.
Thresholds Default -> "Thresholds - workpiece A"	The settings in "Thresholds - workpiece A" are activated.
Digital outputs Default -> ignore	No change; the "Digital outputs" settings remain as they are; they depend on the prior history.

15.4 Device storage (Clone device)

In the **Device storage** menu, you can back up all of the device settings to your PC or restore them from it. You can also transfer all of the settings to another device (clone device). A precondition is that the PMX devices are all identically configured. The network settings, passwords for the various user levels (Operator, Maintenance, Administrator) and the CODESYS applications and CODESYS web visualizations are not transferred.

Important

To prevent any malfunctions, no measurement or control operations may be carried out with the PMX while backing up or loading the device settings. Also do not switch off the operating voltage during backup as the settings will otherwise be lost.

The device storage also contains the working standard calibration certificates for the measurement cards, the manufacturer's certificate, device description files for the fieldbuses (PROFINET® IO, EtherCAT® and EtherNet/IP™), and the log files. The measured value files that were recorded and saved via CODESYS are also saved here. All of the files can be downloaded on to a PC from here.

You can also upload your own files to this area, or delete files from PMX.

16 COMMUNICATION WITH A CONTROL SYSTEM

Digital inputs and outputs, as well as digital interfaces (EtherCAT®, PROFINET® IO or EtherNet/IP™) are available for linking the PMX to machine or plant control systems.

The same device function is accessed in all cases. The input and output signals are also available via the interface. These can be found in the tables in *sections 16.4 and 16.5 starting on page 262*.

16.1 Device description file

Physical properties (e.g. transmitted/received bytes) are described in the device master data file. This is needed to parameterize the master and create the automation program.

Use the following combinations of PMX firmware and device description files. They can be found:

- in the PMX's internal device memory;
- on the website at https://www.hbm.com/de/2981/pmx-modular-measuring-amplifier-system-for-the-iot/.

Important

These device description files are structured modularly like the PMX. The PMX entry must be adapted exactly to the PMX that is used and the number of computing channels transferred in the configuration software of the relevant control unit.

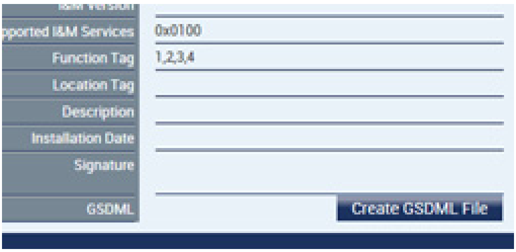
PMX device description files	PMX firmware up to and including 1.46	PMX firmware as of 2.0
PROFINET® IO	GSDML-V2.25-HBM-PMX-20121025.xml GSDML-V2.25-HBM-PMX_IRT-20130404.xml	GSDML-V2.3-HBM-PMX_IRT-PLC-20141215.xml
EtherCAT®	HBM_PMX.xml	HBM_PMX_rev2.xml
EtherNet/IP™	HBM_PMX_023.eds	HBM_PMX_024.eds

When using the PROFINET® IO interface card (PX01PN) note the version (software stack) of the card. It appears in the **Fieldbus** menu in the web browser.

Use the appropriate GSDML file as listed in the table below depending on the software stack and firmware in the PMX.

PMX firmware	PROFINET® IO Stack 3.4.15	PROFINET® IO Stack 3.5.49
2.0 with PLC channels PROFINET application V2.2.0	GSDML-V2.25-HBM-PMX-PLC-V3.4-20141216.xml Note: Use the "Create GSDML File" button	GSDML-V2.3-HBM-PMX_IRT-PLC-20141215.xml Note: Use the "Create GSDML File" button
1.46 without PLC channels PROFINET® IO application V2.1.0	GSDML-V2.25-HBM-PMX-20121025.xml (RT only)	GSDML-V2.25-HBM-PMX-20121025.xml (RT only)
	GSDML-V2.25-HBM-PMX_IRT-20130404.xml (RT and IRT)	GSDML-V2.3-HBM-PMX-IRT-noPLC -20141216.xml (RT and IRT)
1.44 without PLC channels PROFINET® IO application V1.0.0	GSDML-V2.25-HBM-PMX-20121025.xml (RT only)	Not supported
	GSDML-V2.25-HBM-PMX_IRT-20130404.xml (RT and IRT)	

Generating a fixed device description file (firmware 2.00 or higher)



Click on the **Create … File** button to create a device description file which exactly matches the PMX that is used with its plug-in cards and calculation channels to be transferred. This eliminates the need for a manual adjustment in the configuration software of the control unit.

The file can be generated any number of times. The filename contains "… generated …". If there is already a file with the same name, it will be overwritten.

Card types PX878 and PX02 (empty slot) do not appear in the file because they do not return any data relevant to the fieldbus.

Important

Before generating, you must select the number of calculation channels to be transferred.

Storing data in the device

The file is stored in public device storage. It can be downloaded or deleted there. To access it, from the menu choose **Settings -> System -> Device -> Device storage -> Show device storage**.

Folder: *public/PROFINET* or *public/EtherCAT or public/EtherNet_IP.*

The folder can also be displayed in the web browser. To do this, enter in the address bar:

http://<PMX-Name>/public/PROFINET/ or
http://<PMX-Name>/public/EtherCAT/ or
http://<PMX-Name>/public/EtherNet_IP/

where "PMX-Name" is the PMX network name. Input is case-sensitive!

You can use the IP address as an alternative notation, e.g.:

http://172.19.201.184/public/PROFINET/

Exceptions

The file is created from a template file located in public/PROFINET or public/EtherCAT or public/EtherNet_IP. If that file is not found, an error message appears "Cannot open source file."

A firmware update (including one with the same version number as the installed firmware) will restore the template file.

If a measurement card is not installed, and no calculation channels have been transferred yet, the generated file is not a valid file and will not be accepted by PROFINET® IO configurators, nor by EtherCAT® or EtherNet/IP™.

16.2 Setting the transfer speed of the fieldbus

You can set the transfer speed of the fieldbus.

- Set the user level to Administrator in the overview at the top right.

- In the **Settings -> System -> Device -> System options** menu, set **Internal data rate** to the desired value.

The fieldbus update rate will follow this value up to the fieldbus-specific maximum. The change is effective immediately.

- Click on the floppy disk icon  at the bottom right to save permanently.

16.3 Data transfer via fieldbus:

Six bytes are required per measured value for the fieldbus (EtherCAT®, PROFINET® or EtherNet/IP™). This is 4 bytes of data + 1 byte control word + 1 byte status.

The formula for the amount of data is:
46 bytes base load + 6 bytes * number of measurement and calculation channels.

16.4 Input data, PMX -> PLC

16.4.1 Device data (cyclic)

Function		EtherCAT® Index	PROFINET® Slot.Subslot	Data type
System status	See "System status" table in *section 16.4.2*	6000.1	0.2 bytes 0 … 3	uint32
Parameter set	Currently active parameter set	6000.2	0.2 bytes 4 … 7	int32
GUI status	Object dictionary response	6000.3	0.2 bytes 8 … 15	uint64
Limit value switch status	Bit x = 1: limit value switch x is set	6000.4	0.2 bytes 16 … 19	uint32

Function		EtherCAT® Index	PROFINET® Slot.Subslot	Data type
Limit value reset acknowledge-ment	Acknowledgment of "Limit value switch reset request"; acknowledgment same as request means: Reset has been performed	6000.5	0.2 bytes 20 … 21	uint16
Time stamp	PMX time stamp, counts at 153.6 kHz	6000.6	0.2 bytes 22 … 29	uint64
Digital outputs	Current status	6000.7	0.2 bytes 30 … 33	uint32

16.4.2 System status

Bit	Function	
0	Error in factory settings	
1	Device is sync master	Also set in the single-user device
2	Sync error	No connection or disrupted connection
3	Sync error	Synchronization not possible
4	Heartbeat	Bit switches with approx. 1 Hz
5	Excitation overload	Excess current caused by exter-nal consumers (transducer excitation)
6	catman® interface buffer overrun	Data transmission error, loss of data
7	Device not ready	Device working and not deliver-ing valid measured values
8	Calculated Channels Overrun	Calculation time overrun in the calculation channels

16.4.3 Measured values (cyclic)

Function		EtherCAT® Index	PROFINET® Slot.Subslot	Data type
Flags	Status flags from calculated channels	6001.1	0.3 bytes 0 ... 3	uint32
Status flags (reserved for future use)	always 0	6001.2	0.3 byte 4	uint8
Control word acknowledgment flags (reserved for future use)	'Flags control word' return 7001.1	6001.3	0.3 byte 5	uint8
Digital inputs	Level of the digital inputs	6002.1	0.4 bytes 0 ... 3	uint32
Digital inputs for status (reserved for future use)	always 0	6002.2	0.4 byte 4	uint8
Digital inputs for control word acknowledgment (reserved for future use)	'Digital inputs control word' return	6002.3	0.4 byte 5	uint8
Measured value slot x.y		60xy.1	x.y bytes 0 ... 3	float32
Measurement status	See "Measurement status" table in *section 16.4.4*	60xy.2	x.y byte 4	uint8
Measured value for control word acknowledgment (confirms processing of the control word)	Control word return 70xy.2	60xy.3	x.y byte 5	uint8
– – –	Number dependent on the connected measurement cards			

Function		EtherCAT® Index	PROFINET® Slot.Subslot	Data type
Calculated channel in slot 9.z		60xy.1	9.z bytes 0 ... 3	float32
Status	See "Measure-ment status" table, chapter 16.4.4'	60xy.2	9.z byte 4	uint8
Control word acknowledgment (confirms processing of the control word)	Control word return	60xy.3	9. byte 5	uint8
–	Number dependent on the number of calculated chan-nels set on the fieldbus			

Note on calculated channels

In the PMX, calculated channels are assigned to virtual slot 9. For technical reasons, 9 cannot be the third digit in the EtherCAT® indexes. So the calculated channels appear in indexes 6051 to 60b4.

16.4.4 Measurement status

Bit	Function	
0	Working standard calibration invalid	-
1	Measured value invalid	Overflow, underflow, defective sensor, calibration in progress
2	Autocalibration running	Measurement channel with auto-matic calibration (measuring bridges)
3	TEDS error	-
4	Test signal	The measured value is overloaded by a test signal (amplifier dialog). This is not an error state, it provides information.

Tip

If the channel status is 0 for all bits, the measured value is OK.

16.5 Output data PLC ⇒ PMX

16.5.1 Device data (cyclic)

Function		EtherCAT® Index	PROFINET® Slot.Subslot	Data type
Device control word	Bit0: LEDs flash for 30 s Bit1 (value 0x02): Enable object dictionary server Bit2 (value 0x04): Save parameter (same function as floppy disk icon on web interface), edge-triggered from 0 -> 1	7000.1	0.2 bytes 0..3	uint32
Parameter set request	Range 0 … 999	7000.2	0.2 bytes 4 … 7	uint32
GUI signaling	Object dictionary command	7000.3	0.2 bytes 8 … 15	uint64
Limit value switch reset request	Bit x = 1: Output of limit value switch x is reset (x = 0 … 15)	7000.4	0.2 bytes 16 … 17	uint16
Limit value switch enable (one bit must be "1", so that the corresponding limit value can be changed via the fieldbus)	Bit x = 1: Limit value switch x is defined via the fieldbus (x = 0 … 15)	7000.5	0.2 bytes 18 … 19	uint16
Limit switch 0	Limit value no. 0	7000.6	0.2 bytes 20 … 23	float32

<table>
<tr><th>Function</th><th></th><th>EtherCAT® Index</th><th>PROFINET® Slot.Subslot</th><th>Data type</th></tr>
<tr><td>...</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Limit switch 15</td><td>Limit value no. 15</td><td>7000.21</td><td>0.2
bytes 80 ... 83</td><td>float32</td></tr>
<tr><td>Digital outputs</td><td>Set digital outputs: Digital output x = bit x
(this set bit is assigned to a digital output on a PX878 via the Digital output menu)</td><td rowspan="2">7000.22</td><td rowspan="2">0.2
bytes 84 ... 87</td><td rowspan="2">uint32</td></tr>
<tr><td>Digital inputs</td><td>Bits 16 ... 31 of the digital outputs are also transferred to the calculated channels as "Digital inputs 17 ... 32". This allows you to control function blocks.</td></tr>
</table>

Function		EtherCAT® Index	PROFINET® Slot.Subslot	Data type
PLC channel 1 (from firmware 2.00)	Freely usable	7002.1	0.2 bytes 88 ... 91	float32
PLC channel 2 (from firmware 2.00)	Freely usable	7002.2	0.2 bytes 92 ... 95	float32
PLC channel 3 (from firmware 2.00)	Freely usable	7002.3	0.2 bytes 96 ... 99	float32
PLC channel 4 (from firmware 2.00)	Freely usable	7002.4	0.2 bytes 100 ... 103	float32
PLC channel 5 (from firmware 2.00)	Freely usable	7002.5	0.2 bytes 104 ... 107	float32
PLC channel 6 (from firmware 2.00)	Freely usable	7002.6	0.2 bytes 108 ... 111	float32
PLC channel 7 (from firmware 2.00)	Freely usable	7002.7	0.2 bytes 112 ... 115	float32
PLC channel 8 (from firmware 2.00)	Freely usable	7002.8	0.2 bytes 116 ... 119	float32

16.5.2 Measured value control words (cyclic)

Function		EtherCAT® Index	PROFINET® Slot.Subslot	Data type
Flags control word	Reserved for future processing	7006.1	0.3	uint8
Digital inputs control word	Reserved for future processing	7006.2	0.4	uint8
Control word for measured value slot x.y	Function see *section 16.5.3*	70xy.1	x.y	uint8

Function		EtherCAT® Index	PROFINET® Slot.Subslot	Data type
---	Number of control words depending on the connected measurement cards. One control word per measurement channel.			
Control word for calculated channel slot 9.z	Function see *section 16.5.3*	70xy.1	9.z	uint8
---	Number of control words, dependent on the connected measurement cards. One control word per calculated channel.			

Important

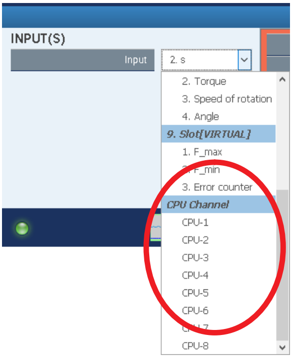
Note on calculated channels:
In the PMX device, the calculated channels are assigned to virtual slot 9.
For technical reasons, 9 cannot be the third digit in the EtherCAT® indexes.
Calculated channels currently appear in indexes 7051 to 70b4.

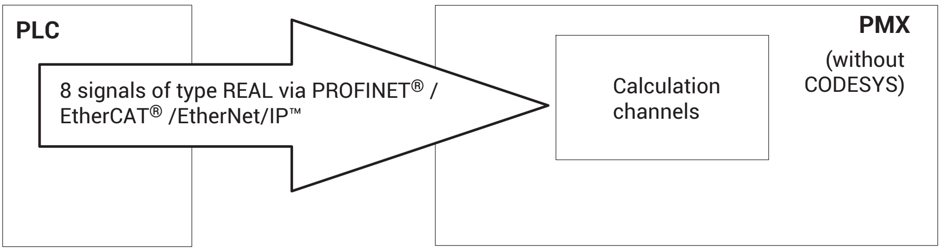
16.5.3 Measured value control words

Bit	Function	Responds to	Applicable to
0	Set to zero	Edge 0 -> 1	Measurement channel
1	Offset = 0	Edge 0 -> 1	Measurement channel
2	Reset of max, min or peak-to-peak values	Edge 0 -> 1	Extreme value channel (calculated channel in slot 9)
3	Hold	Level = 1	Extreme value channel (calculated channel in slot 9)
4	Recalibrate	Edge 0 -> 1	Measurement channel with automatic calibration (measuring bridges), relates only to PX455
5	Shunt	Edge	PX460 channels 2 and 4

16.5.4 Fieldbus channels (CPU channels)

A controller (PLC) can feed up to 8 signals into the PMX as CPU channels via fieldbus (Industrial Ethernet). Then they are available in the calculation channels for further processing. This function is available in the PMX as from firmware version 2.00 for PROFINET®, EtherCAT® and EtherNet/IP™.



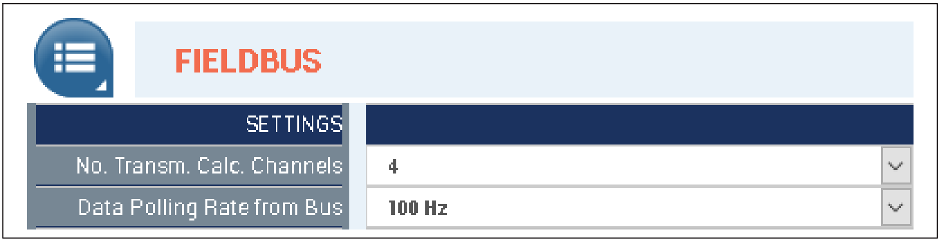


Important

Does not work in devices with CODESYS (basic device WGX001), only in devices with WGX002.

You set the processing speed for the signal transmission of the fieldbus channels from a PLC into the PMX with the "Data Polling Rate from Bus".

When the signals are in the PMX, they are processed at the default processing rate of 19200/s or 38460/s respectively.

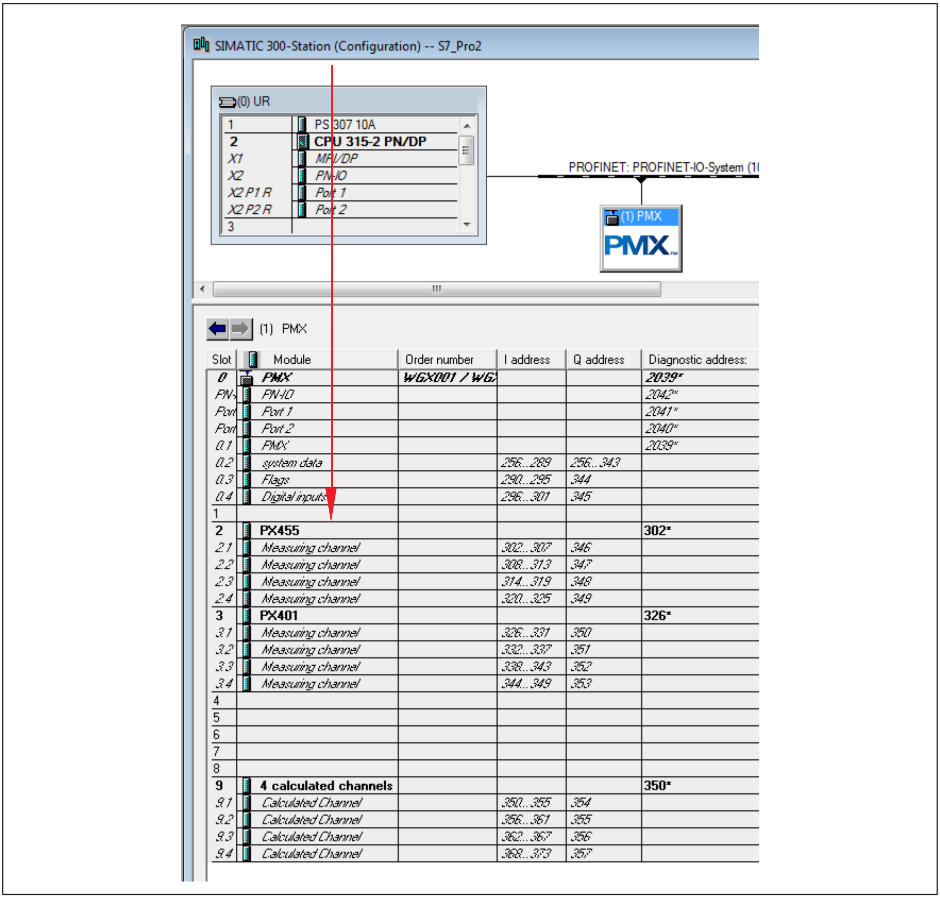


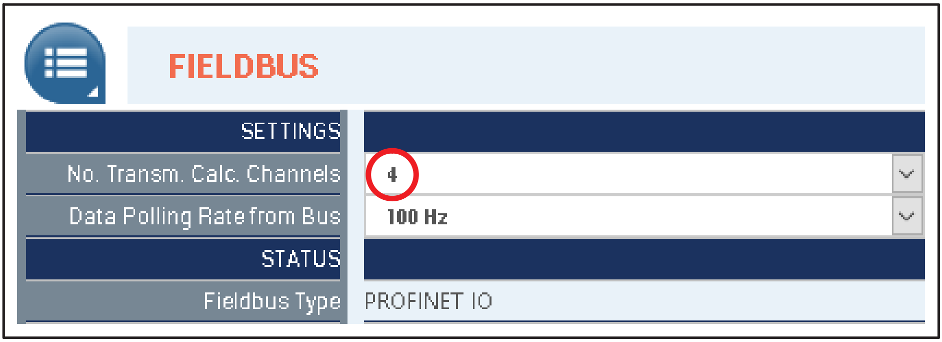
16.6 PROFINET® IO

- Network settings
  The PROFINET®-specific network settings (IP address, device name, etc.) are selected using the PROFINET® configuration tool and set via the PROFINET® cable. This data can be read for monitoring purposes, and as from PMX firmware version 3.0 also set, in the **Fieldbus** dialog of the PMX user interface.
- The PROFINET® configuration must match the installed PMX cards.
- The cable length must be entered in the master configuration for IRT operation, otherwise transfer errors may occur when using long cables.

Example

	Slot 1	Slot 2	Slot 3	Slot 4	Slot 9 (virtual)
Installed in PMX	PX878	PX455	PX401	empt-y	Calculated channels
PROFINET® configuration	No data for PROFINET®. Leave this slot empty, see below.	PX455	PX401	empt-y	Number of calculated channels must match the PMX setting (**Fieldbus** menu).


SIMATIC 300-Station (Configuration) -- S7_Pro2
(0) UR
1 PS 307 10A
2 CPU 315-2 PN/DP
X1 MPI/DP
X2 PN-IO
X2 P1 R Port 1
X2 P2 R Port 2
3
PROFINET: PROFINET-IO-System (1(
(1) PMX
PMX
(1) PMX
Slot Module Order number I address Q address Diagnostic address:
0 PMX WGX001 / WGX 2039*
PN-IO 2042*
Port 1 2041*
Port 2 2040*
0.1 PMX 2039*
0.2 system data 256...289 256...343
0.3 Flags 290...295 344
0.4 Digital inputs 296...301 345
1
2 PX455 302*
2.1 Measuring channel 302...307 346
2.2 Measuring channel 308...313 347
2.3 Measuring channel 314...319 348
2.4 Measuring channel 320...325 349
3 PX401 326*
3.1 Measuring channel 326...331 350
3.2 Measuring channel 332...337 351
3.3 Measuring channel 338...343 352
3.4 Measuring channel 344...349 353
4
5
6
7
8
9 4 calculated channels 350*
9.1 Calculated Channel 350...355 354
9.2 Calculated Channel 356...361 355
9.3 Calculated Channel 362...367 356
9.4 Calculated Channel 368...373 357


FIELDBUS
SETTINGS
No. Transm. Calc. Channels
4
Data Polling Rate from Bus
100 Hz
STATUS
Fieldbus Type
PROFINET IO

Important

Examples of the configuration and operation of the PMX via fieldbuses can be found at https://www.hbm.com/de/2981/pmx-modular-measuring-amplifier-system-for-the-iot/.

16.7 EtherCAT®

The EtherCAT® master configuration must match the installed cards.

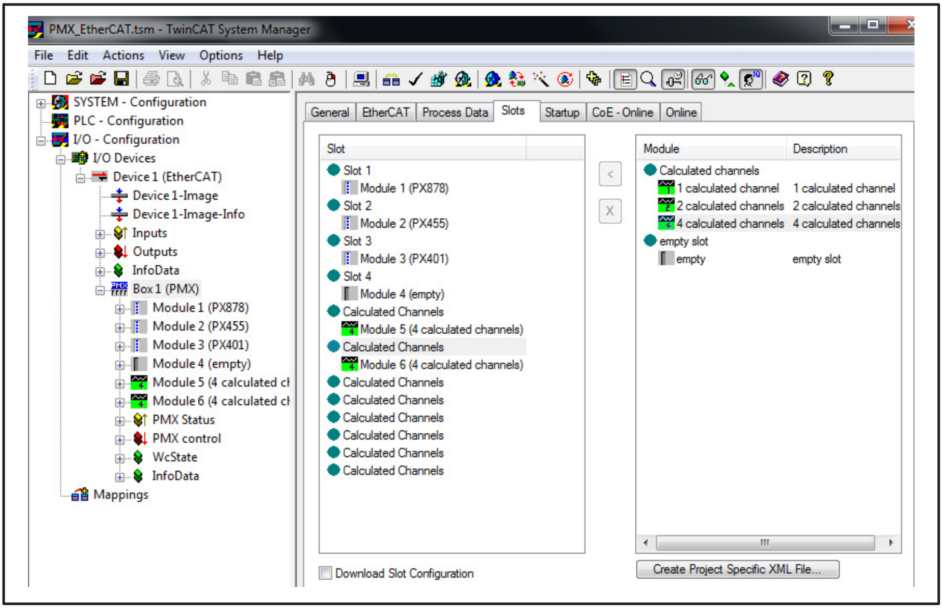
Calculated channels

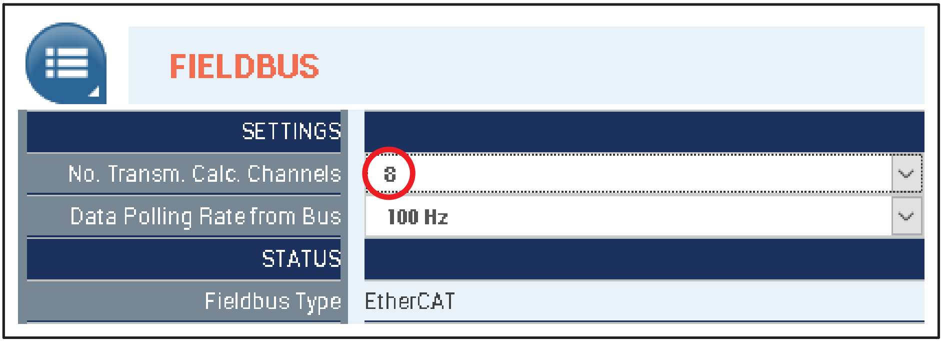
The number must match the PMX setting (**Fieldbus** menu).

The calculated channels are distributed in the EtherCAT® master to virtual "Calculated Channels" slots.

The distribution across the slots is irrelevant, but the total number of the channels must match.

Example with eight calculated channels:





16.8 Using the PMX CoE Object Dictionary

CoE stands for CAN over EtherCAT®.

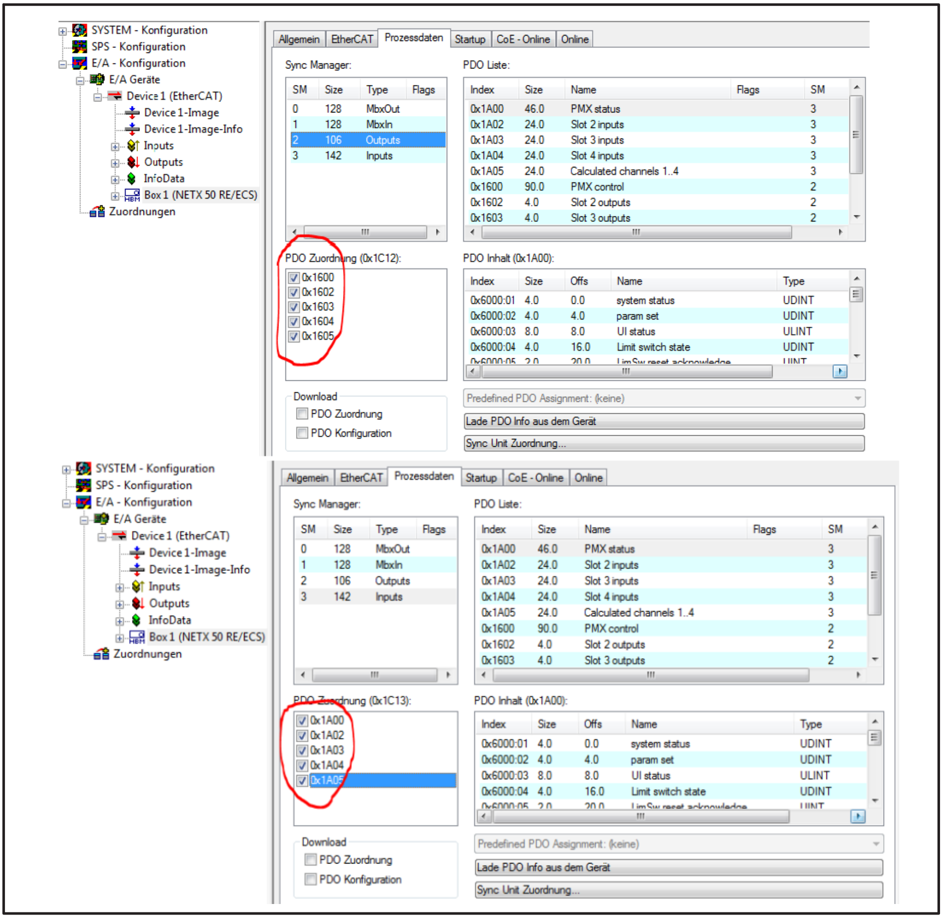
This means there is a great available variety of CANopen™ devices and application profiles for device classes and applications: Starting from the I/O modules via drives (e.g. drive profile CiA 402 standardized as IEC 61800-7-201/301), encoders (CiA 406), proportional valves and hydraulic regulators (CiA 408) right up to application profiles.

Method with TwinCAT

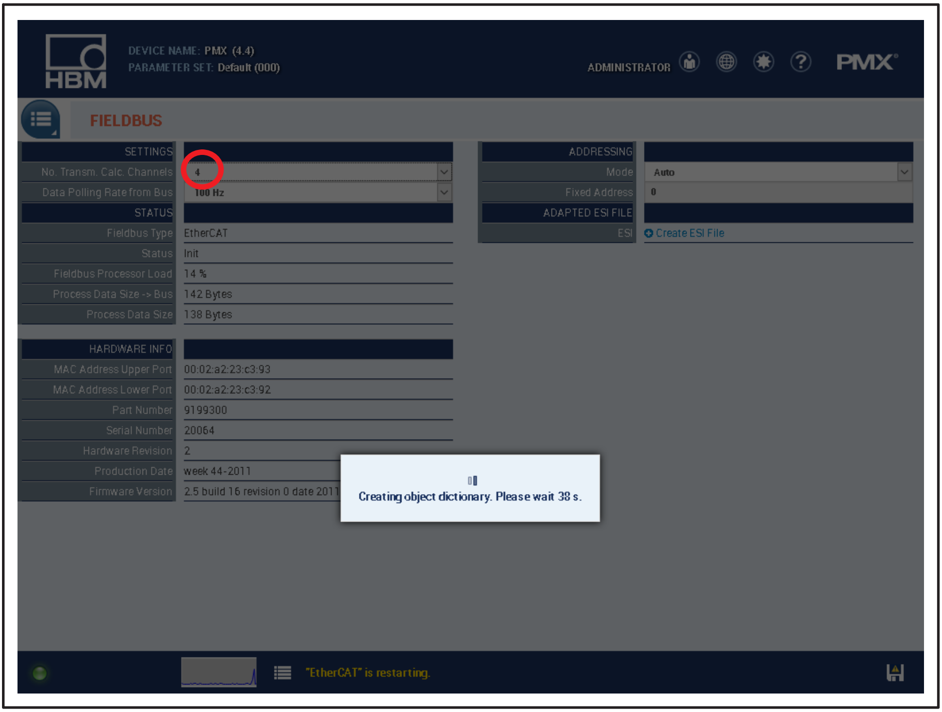
1. Delete the PMX ESI file from the TwinCAT folder (default C:\TwinCAT\Io\EtherCAT) before starting TwinCAT.
   Alternatively, you can change the .xml extension, such as to "HBM_PMX .xml.doNotUseYet".
2. TwinCAT finds the PMX by the device scan.
   The PMX does not support partial PDO selection.

Important

You have to select all PDOs manually, otherwise the Sync Manager sizes will not be correct (unfortunately it is not possible to parameterize the PDOs as Fixed and Mandatory).


SYSTEM - Konfiguration
SPS - Konfiguration
E/A - Konfiguration
E/A Geräte
Device 1 (EtherCAT)
Device 1-Image
Device 1-Image-Info
Inputs
Outputs
InfoData
Box 1 (NETX 50 RE/ECS)
Zuordnungen
Allgemein
EtherCAT
Prozessdaten
Startup
CoE - Online
Online
Sync Manager:
SM Size Type Flags
0 128 MbxOut
1 128 MbxIn
2 106 Outputs
3 142 Inputs
PDO Liste:
Index Size Name Flags SM
0x1A00 46.0 PMX status 3
0x1A02 24.0 Slot 2 inputs 3
0x1A03 24.0 Slot 3 inputs 3
0x1A04 24.0 Slot 4 inputs 3
0x1A05 24.0 Calculated channels 1..4 3
0x1600 90.0 PMX control 2
0x1602 4.0 Slot 2 outputs 2
0x1603 4.0 Slot 3 outputs 2
PDO Zuordnung (0x1C12):
0x1600
0x1602
0x1603
0x1604
0x1605
PDO Inhalt (0x1A00):
Index Size Offs Name Type
0x6000:01 4.0 0.0 system status UDINT
0x6000:02 4.0 4.0 param set UDINT
0x6000:03 8.0 8.0 UI status ULINT
0x6000:04 4.0 16.0 Limit switch state UDINT
Download
PDO Zuordnung
PDO Konfiguration
Predefined PDO Assignment: (keine)
Lade PDO Info aus dem Gerät
Sync Unit Zuordnung...
PDO Zuordnung (0x1C13):
0x1A00
0x1A02
0x1A03
0x1A04
0x1A05

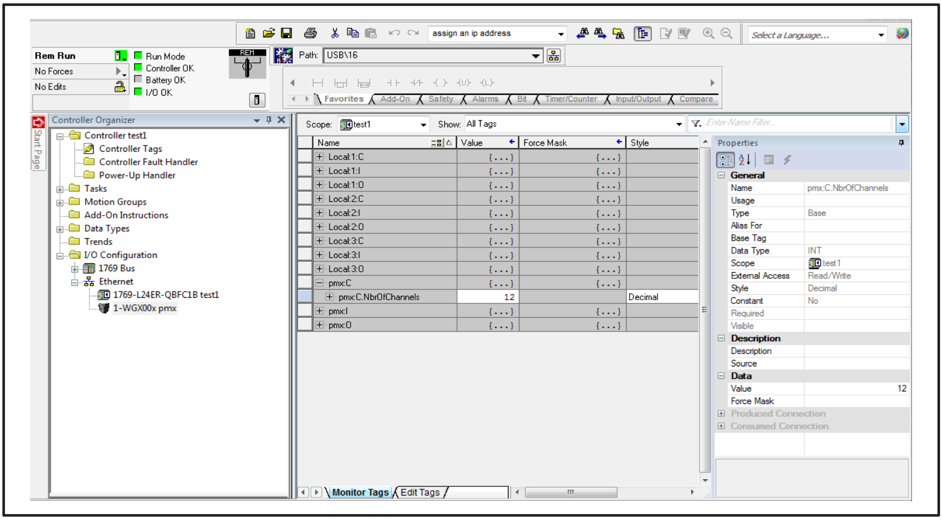
3. The further procedure is identical to that with an ESI file.
4. If calculated channels are to be sent via EtherCAT®, set the desired number in the **Settings -> Fieldbus** dialog.



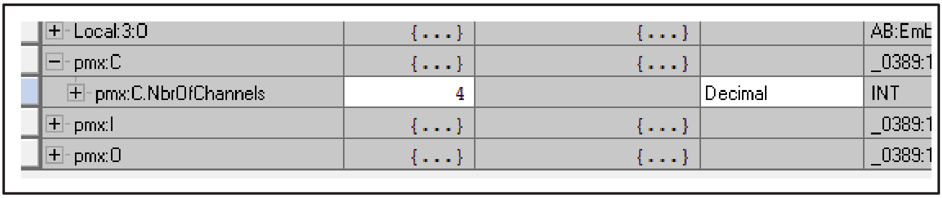
16.9 EtherNet/IP™

16.9.1 Configuration

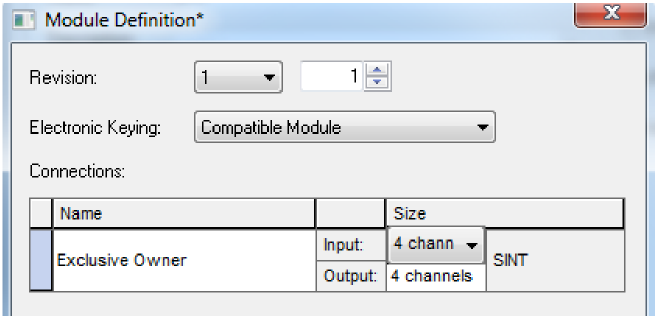
How to set the number of transferred measurement channels:



1. Set configuration object 199 "**NbrOfChannels**" (class 4, instance 199).
   This determines the number of channels that are copied into the data frame (range 0 … 48).
   In RSLogix 5000, for example, it looks like this:



2. Select the quantities of the two module instances 100 and 101. This number should match "**NbrOfChannels**", range 0 … 48 in increments of 4.
   In RSLogix 5000, for example, it looks like this:



Important

Examples of the configuration and operation of the PMX via fieldbuses can be found at https://www.hbm.com/de/2981/pmx-modular-measuring-amplifier-system-for-the-iot/.

16.9.2 Channel settings

The installed measurement cards provide the signals of the channels in the order of their installation in the PMX rack. The card in slot 1 delivers data starting with channel 1. An empty slot does not supply any channels; only the EtherNet/IP™ channels are all occupied without gaps.
The calculated channels deliver their data according to the measurement card channels.

Card Type	Channels Use
PX02 (empty slot)	0
PX401	4
PX455	4
PX460	4
PX878	0, like an empty slot
Calculated channel	1

Example

	Slot 1	Slot 2	Slot 3	Slot 4	Calculated Channels
Card type	PX878	PX401	PX455	empty	none
Channels used	0	4	4	0	3
Channels in EtherNet/IP™	none	Channels 1, 2, 3, 4	Channels 5, 6, 7, 8	none	Channels 9, 10, 11

16.9.3 Data structure

Assembly 100

From PMX (adapter) to scanner

Index	Size in octets	Type	Tag	
0 ... 3	4	UDINT	System status	SystemData (transmitted always)
4 ... 7	4	DINT	ParameterSet	
8 ... 15	8	ULINT	UiStatus	
16 ... 19	4	UDINT	LimitSwitchState	
20 ... 21	2	UINT	LimitResetAckn	
22 ... 29	8	ULINT	TimeStamp	
30 ... 33	4	UDINT	DigitalOutputState	
34 ... 37	4	UDINT	Flags	
38	1	USINT	FlagsStatus	
39	1	USINT	FlagsAcknowledge	
40 ... 43	4	UDINT	DigitalInputsState	
44	1	USINT	DigitalInputsStatus	
45	1	USINT	DigitalInputsAcknowledge	

The number of items of channel data transferred depends on the configuration - see below.

Index	Size in octets	Type	Tag	
46 ... 49	4	REAL	MeasValue	Channel 1
50	1	USINT	MeasStatus	
51	1	USINT	MeasAcknowledge	
52 ... 55	4	REAL	MeasValue	Channel 2
56	1	USINT	MeasStatus	
57	1	USINT	MeasAcknowledge	
58 ... 61	4	REAL	MeasValue	Channel 3
62	1	USINT	MeasStatus	
63	1	USINT	MeasAcknowledge	
64 ... 67	4	REAL	MeasValue	Channel 4
68	1	USINT	MeasStatus	
69	1	USINT	MeasAcknowledge	
70 ... 73	4	REAL	MeasValue	Channel 5
74	1	USINT	MeasStatus	
75	1	USINT	MeasAcknowledge	
76 ... 79	4	REAL	MeasValue	Channel 6
80	1	USINT	MeasStatus	
81	1	USINT	MeasAcknowledge	
82 ... 85	4	REAL	MeasValue	Channel 7
86	1	USINT	MeasStatus	
87	1	USINT	MeasAcknowledge	
88 ... 91	4	REAL	MeasValue	Channel 8
92	1	USINT	MeasStatus	
93	1	USINT	MeasAcknowledge	
94 ... 97	4	REAL	MeasValue	Channel 9
98	1	USINT	MeasStatus	
99	1	USINT	MeasAcknowledge	
100 ... 103	4	REAL	MeasValue	Channel 10
104	1	USINT	MeasStatus	
105	1	USINT	MeasAcknowledge	

Index	Size in octets	Type	Tag	
106 … 109	4	REAL	MeasValue	Channel 11
110	1	USINT	MeasStatus	
111	1	USINT	MeasAcknowledge	
112 … 115	4	REAL	MeasValue	Channel 12
116	1	USINT	MeasStatus	
117	1	USINT	MeasAcknowledge	
118 … 121	4	REAL	MeasValue	Channel 13
122	1	USINT	MeasStatus	
123	1	USINT	MeasAcknowledge	
124 … 127	4	REAL	MeasValue	Channel 14
128	1	USINT	MeasStatus	
129	1	USINT	MeasAcknowledge	
130 … 13	4	REAL	MeasValue	Channel 15
134	1	USINT	MeasStatus	
135	1	USINT	MeasAcknowledge	
136 … 139	4	REAL	MeasValue	Channel 16
140	1	USINT	MeasStatus	
141	1	USINT	MeasAcknowledge	
142 … 145	4	REAL	MeasValue	Channel 17
146	1	USINT	MeasStatus	
147	1	USINT	MeasAcknowledge	
148 … 151	4	REAL	MeasValue	Channel 18
152	1	USINT	MeasStatus	
153	1	USINT	MeasAcknowledge	
154 … 157	4	REAL	MeasValue	Channel 19
158	1	USINT	MeasStatus	
159	1	USINT	MeasAcknowledge	
160 … 163	4	REAL	MeasValue	Channel 20
164	1	USINT	MeasStatus	
165	1	USINT	MeasAcknowledge	

Index	Size in octets	Type	Tag	
166 ... 169	4	REAL	MeasValue	Channel 21
170	1	USINT	MeasStatus	
171	1	USINT	MeasAcknowledge	
172 ... 175	4	REAL	MeasValue	Channel 22
176	1	USINT	MeasStatus	
177	1	USINT	MeasAcknowledge	
178 ... 181	4	REAL	MeasValue	Channel 23
182	1	USINT	MeasStatus	
183	1	USINT	MeasAcknowledge	
184 ... 187	4	REAL	MeasValue	Channel 24
188	1	USINT	MeasStatus	
189	1	USINT	MeasAcknowledge	
190 ... 193	4	REAL	MeasValue	Channel 25
194	1	USINT	MeasStatus	
195	1	USINT	MeasAcknowledge	
196 ... 199	4	REAL	MeasValue	Channel 26
200	1	USINT	MeasStatus	
201	1	USINT	MeasAcknowledge	
202 ... 205	4	REAL	MeasValue	Channel 27
206	1	USINT	MeasStatus	
207	1	USINT	MeasAcknowledge	
208 ... 211	4	REAL	MeasValue	Channel 28
212	1	USINT	MeasStatus	
213	1	USINT	MeasAcknowledge	
214 ... 217	4	REAL	MeasValue	Channel 29
218	1	USINT	MeasStatus	
219	1	USINT	MeasAcknowledge	
220 ... 223	4	REAL	MeasValue	Channel 30
224	1	USINT	MeasStatus	
225	1	USINT	MeasAcknowledge	

Index	Size in octets	Type	Tag	
226 ... 229	4	REAL	MeasValue	Channel 31
230	1	USINT	MeasStatus	
231	1	USINT	MeasAcknowledge	
232 ... 235	4	REAL	MeasValue	Channel 32
236	1	USINT	MeasStatus	
237	1	USINT	MeasAcknowledge	
238 ... 241	4	REAL	MeasValue	Channel 33
242	1	USINT	MeasStatus	
243	1	USINT	MeasAcknowledge	
244 ... 247	4	REAL	MeasValue	Channel 34
248	1	USINT	MeasStatus	
249	1	USINT	MeasAcknowledge	
250 ... 253	4	REAL	MeasValue	Channel 35
254	1	USINT	MeasStatus	
255	1	USINT	MeasAcknowledge	
256 ... 259	4	REAL	MeasValue	Channel 36
260	1	USINT	MeasStatus	
261	1	USINT	MeasAcknowledge	
262 ... 265	4	REAL	MeasValue	Channel 37
266	1	USINT	MeasStatus	
267	1	USINT	MeasAcknowledge	
268 ... 271	4	REAL	MeasValue	Channel 38
272	1	USINT	MeasStatus	
273	1	USINT	MeasAcknowledge	
274 ... 277	4	REAL	MeasValue	Channel 39
278	1	USINT	MeasStatus	
279	1	USINT	MeasAcknowledge	
280 ... 283	4	REAL	MeasValue	Channel 40
284	1	USINT	MeasStatus	
285	1	USINT	MeasAcknowledge	

Index	Size in octets	Type	Tag	
286 … 289	4	REAL	MeasValue	Channel 41
290	1	USINT	MeasStatus	
291	1	USINT	MeasAcknowledge	
292 … 295	4	REAL	MeasValue	Channel 42
296	1	USINT	MeasStatus	
297	1	USINT	MeasAcknowledge	
298 … 301	4	REAL	MeasValue	Channel 43
302	1	USINT	MeasStatus	
303	1	USINT	MeasAcknowledge	
304 … 307	4	REAL	MeasValue	Channel 44
308	1	USINT	MeasStatus	
309	1	USINT	MeasAcknowledge	
310 … 313	4	REAL	MeasValue	Channel 45
314	1	USINT	MeasStatus	
315	1	USINT	MeasAcknowledge	
316 … 319	4	REAL	MeasValue	Channel 46
320	1	USINT	MeasStatus	
321	1	USINT	MeasAcknowledge	
322 … 325	4	REAL	MeasValue	Channel 47
326	1	USINT	MeasStatus	
327	1	USINT	MeasAcknowledge	
328 … 331	4	REAL	MeasValue	Channel 48
332	1	USINT	MeasStatus	
333	1	USINT	MeasAcknowledge	

Assembly 101

From scanner to PMX (adapter)

Index	Size in octets	Type	Tag	
0..3	4	UDINT	PMX Control	SystemData (transmitted always)
4..7	4	DINT	ParamSetRequest	
8..15	8	ULINT	UiControl	
16..17	2	UINT	LimitSwitchReset	
18..19	2	UINT	LimitSwitchEnable	
20..23	4	REAL	LimitThresh0	
24..27	4	REAL	LimitThresh1	
28..31	4	REAL	LimitThresh2	
32..35	4	REAL	LimitThresh3	
36..39	4	REAL	LimitThresh4	
40..43	4	REAL	LimitThresh5	
44..47	4	REAL	LimitThresh6	
48..51	4	REAL	LimitThresh7	
52..55	4	REAL	LimitThresh8	
56..59	4	REAL	LimitThresh9	
60..63	4	REAL	LimitThresh10	
64..67	4	REAL	LimitThresh11	
68..71	4	REAL	LimitThresh12	
72..75	4	REAL	LimitThresh13	
76..79	4	REAL	LimitThresh14	
80..83	4	REAL	LimitThresh15	
84..87	4	UDINT	DigitalOutputSetting (Note *)	
88..91	4	REAL	PLC channel 0	
92..95	4	REAL	PLC channel 0	
96..99	4	REAL	PLC channel 0	
100..103	4	REAL	PLC channel 0	
104..107	4	REAL	PLC channel 0	
108..111	4	REAL	PLC channel 0	

Index	Size in octets	Type	Tag	
112..115	4	REAL	PLC channel 0	
116..119	4	REAL	PLC channel 0	
120	1	USINT	FlagsControl	
121	1	USINT	DigInputControl	

The number of items of channel data transferred depends on the configuration - see below.

*) Note

Index	DigitalOutputSetting bits …	… are mapped to Digital Inputs (in the calculated channels)
86 bits 0 … 7	16..23	17..24
87 bits 0 … 7	24..31	25..32

Index	Size in octets	Type	Tag	
122	1	USINT	MeasControl	Channel 1
123	1	USINT	MeasControl	Channel 2
124	1	USINT	MeasControl	Channel 3
125	1	USINT	MeasControl	Channel 4
126	1	USINT	MeasControl	Channel 5
127	1	USINT	MeasControl	Channel 6
128	1	USINT	MeasControl	Channel 7
129	1	USINT	MeasControl	Channel 8
130	1	USINT	MeasControl	Channel 9
131	1	USINT	MeasControl	Channel 10
132	1	USINT	MeasControl	Channel 11
133	1	USINT	MeasControl	Channel 12
134	1	USINT	MeasControl	Channel 13
135	1	USINT	MeasControl	Channel 14
136	1	USINT	MeasControl	Channel 15
137	1	USINT	MeasControl	Channel 16
138	1	USINT	MeasControl	Channel 17
139	1	USINT	MeasControl	Channel 18

Index	Size in octets	Type	Tag	
140	1	USINT	MeasControl	Channel 19
141	1	USINT	MeasControl	Channel 20
142	1	USINT	MeasControl	Channel 21
143	1	USINT	MeasControl	Channel 22
144	1	USINT	MeasControl	Channel 23
145	1	USINT	MeasControl	Channel 24
146	1	USINT	MeasControl	Channel 25
147	1	USINT	MeasControl	Channel 26
148	1	USINT	MeasControl	Channel 27
149	1	USINT	MeasControl	Channel 28
150	1	USINT	MeasControl	Channel 29
151	1	USINT	MeasControl	Channel 30
152	1	USINT	MeasControl	Channel 31
153	1	USINT	MeasControl	Channel 32
154	1	USINT	MeasControl	Channel 33
155	1	USINT	MeasControl	Channel 34
156	1	USINT	MeasControl	Channel 35
157	1	USINT	MeasControl	Channel 36
158	1	USINT	MeasControl	Channel 37
159	1	USINT	MeasControl	Channel 38
160	1	USINT	MeasControl	Channel 39
161	1	USINT	MeasControl	Channel 40
162	1	USINT	MeasControl	Channel 41
163	1	USINT	MeasControl	Channel 42
164	1	USINT	MeasControl	Channel 43
165	1	USINT	MeasControl	Channel 44
166	1	USINT	MeasControl	Channel 45
167	1	USINT	MeasControl	Channel 46
168	1	USINT	MeasControl	Channel 47
169	1	USINT	MeasControl	Channel 48

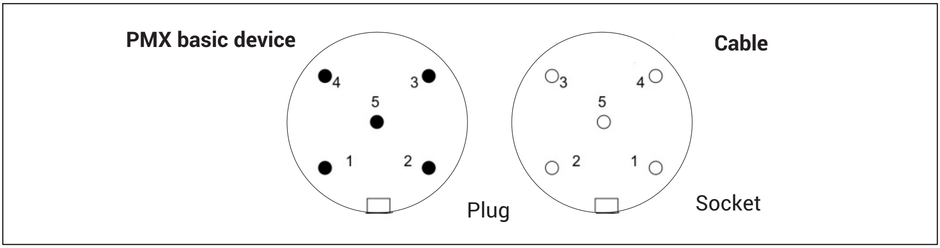
17 CAN INTERFACE (WGX001 ONLY)

17.1 General

The WGX001 basic device has a CAN interface according to ISO11898. In conjunction with the CODESYS Soft PLC, you can operate the PMX as a CANopen slave or CANopen master. To do this, add a CAN component and a CANopen stack in CODESYS. The corresponding PMX package and a collection of sample programs for code generation, web visualization and integration of CANopen modules are also included.

The software is freely available from HBM at:
https://www.hbm.com/de/2981/pmx-modular-measuring-amplifier-system-for-the-iot/.

17.2 CAN pin assignment



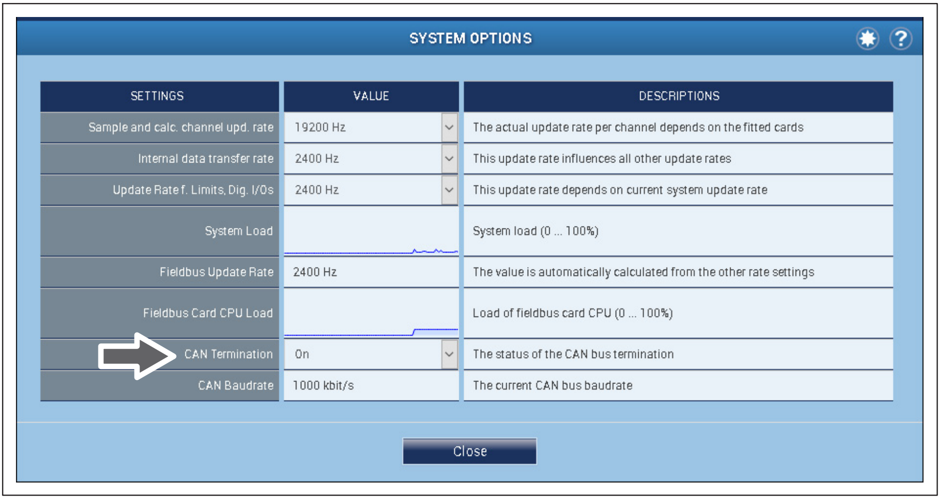
Pin	Signal	Description
1	SHLD	CAN shield
2		Not in use
3	GND	Ground
4	CAN_H	CAN_H data cable (high)
5	CAN_L	CAN_L data cable (low)

Important

The nodes in the CAN network must be supplied separately, and not via the CAN connection (M12 socket) of the PMX.

The bus is terminated by a 120 ohm termination resistor at each end.

The termination resistor is integrated into the PMX. To activate it, from the menu choose **Settings -> System -> Device -> System options -> CAN termination**.


SYSTEM OPTIONS
SETTINGS
VALUE
DESCRIPTIONS
Sample and calc. channel upd. rate
19200 Hz
The actual update rate per channel depends on the fitted cards
Internal data transfer rate
2400 Hz
This update rate influences all other update rates
Update Rate f. Limits, Dig. I/Os
2400 Hz
This update rate depends on current system update rate
System Load
System load (0 ... 100%)
Fieldbus Update Rate
2400 Hz
The value is automatically calculated from the other rate settings
Fieldbus Card CPU Load
Load of fieldbus card CPU (0 ... 100%)
CAN Termination
On
The status of the CAN bus termination
CAN Baudrate
1000 kbit/s
The current CAN bus baudrate
Close

17.3 CANopen master/slave mode

Master mode

No nodes are addressed directly during data transfer over the CAN bus. A unique identifier denotes the content of a message (e.g. press force or stroke).

The identifier also signifies the priority of the message. Message = identifier + signal + additional information from bus node = node

In master mode, you can integrate CAN modules such as digiCLIP, PME, SomatXR or third-party devices via the CODESYS programming environment.
They are integrated by way of the device description files (EDS or DCF) of the CAN modules.

The bus speed of all CAN modules (CAN baud rate) must be identical, and is limited by the length of the bus. The transfer rate can be set between 100 kBit and 1 MBit in the CODESYS programming environment, and is displayed in the PMX web browser's **System options** menu.

Contact the CAN module suppliers as necessary with regard to bus speed setting.

Slave mode

In slave mode the PMX can send SDOs and PDOs of all measurement and calculation channels. A maximum of 128 PDO streams with a total maximum of 128 bytes data size and maximum 199 SDO*255 subIDs are available. The PDO streams can be sent triggered by timer control up to min. 300 Hz or by measured value control up to 1.2 kHz or via a SYNC message.

You create the SDOs and PDOs in the CODESYS programming environment. The transfer rate can be set between 100 kBit and 1 MBit in the CODESYS programming environment, and is displayed in the PMX web browser's **System options** menu.

This provides you with several SDO channels and module-dependent PDO mapping, as well as CAN low-level libraries.

A maximum of 30 CAN messages can in turn be made available as measured values via the PMX calculation channel "Connection with (CODESYS)" in the PMX, where they are immediately "time-stamped".

This enables directly measured quantities and CAN messages to be acquired and analyzed in parallel and synchronously throughout the entire system.

Important

The CAN bus must be terminated at both ends, and the appropriate baud rate must be set for all bus nodes.
Errors in CAN bus operation are not indicated or saved.

18 CODESYS V3 SOFT PLC (WGX001 ONLY)

18.1 General

The WGX001 basic housing enables solutions for many tasks in industrial automation technology with the PMX's CODESYS V3 software platform. It includes everything you need for programming, fieldbus and I/O configuration, visualization, motion control, and other tasks. The CODESYS V3 software platform is based on the IEC 61131-3 programming system. All IEC 61131-3 programming languages are supported.

With a PMX running CODESYS V3, applications can be automated and simultaneously displayed and operated in real time. You create the appropriate web visualization in the CODESYS software. It runs together with the application in the PMX. Via the amplifier's EthernetTCP/IP port you can use the visualization on any browser-based devices or in the PC browser.

A CODESYS runtime license is included in the PMX with the WGX001 basic device. CODESYS V3 and the related PMX package are included on the supplied PMX CODESYS CD. A collection of sample programs for code generation, web visualization and integration of CANopen modules is also included.

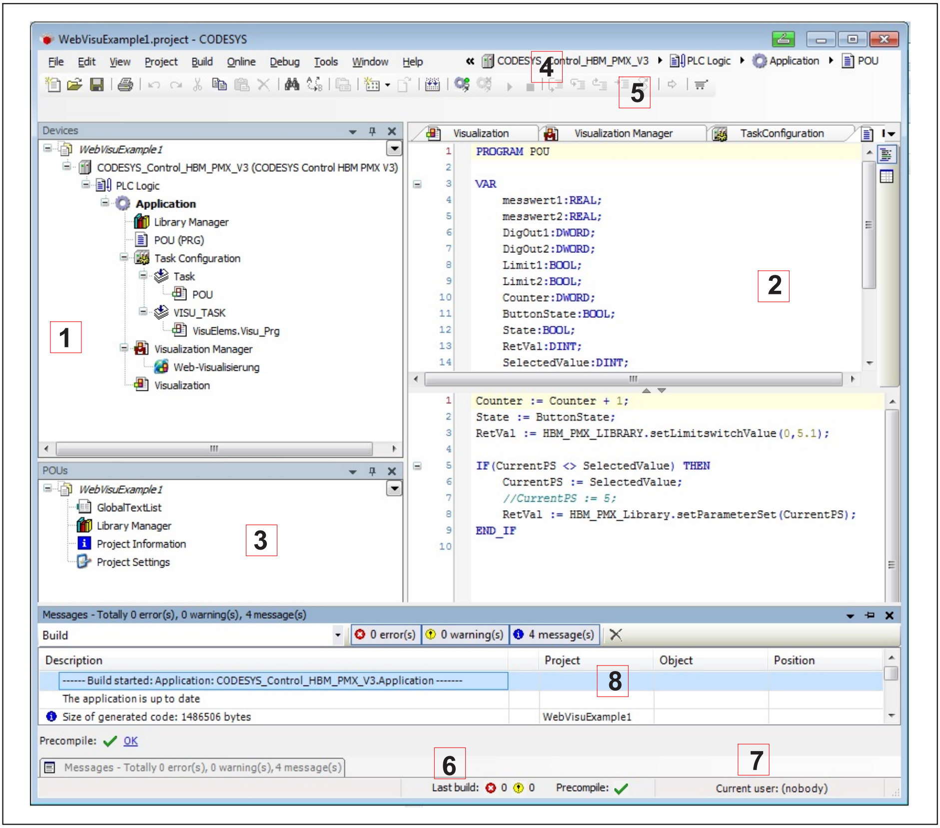
The software package is freely available from HBM at: https://www.hbm.com/de/2981/pmx-modular-measuring-amplifier-system-for-the-iot/.

Important

When a CODESYS application or CODESYS web visualization is running on the PMX, they are retained even after a card is swapped out or the firmware is updated (as from V2.00). Please note that all signals are permanently assigned for CODESYS, and must be checked and corrected as necessary if the measurement cards are moved.
A running application can be stopped directly in the CODESYS development environment. As of firmware V3.00, you can start, stop, reset and also delete CODESYS applications and visualizations individually in the ***CODESYS*** *menu.*
You can also load CODESYS projects that you have transferred to a PMX via the CODESYS development environment to a PC and save them via this menu.

18.2 CODESYS development environment

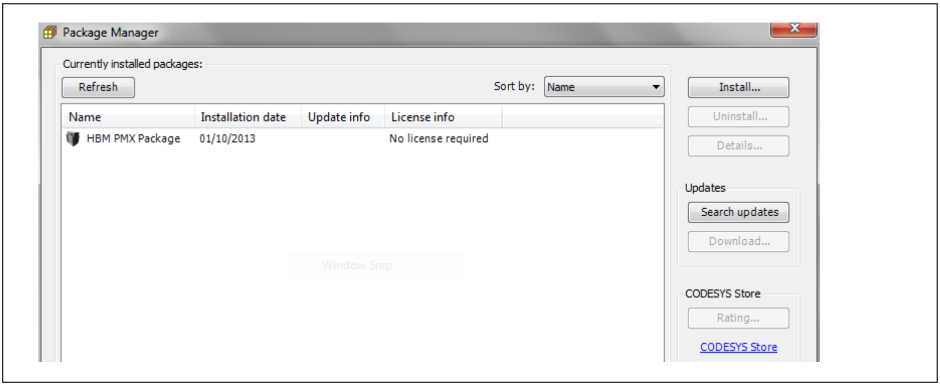
The user interface provides menus and toolbars, windows for editor views, object organization, monitoring and message output, as well as an information and status line.



No.	Meaning
1	Devices window
2	Editor window
3	POU (Programmable Organizational Unit) window
4	Menu bar
5	Toolbar
6	Info on editor position
7	Info on current user
8	Message window

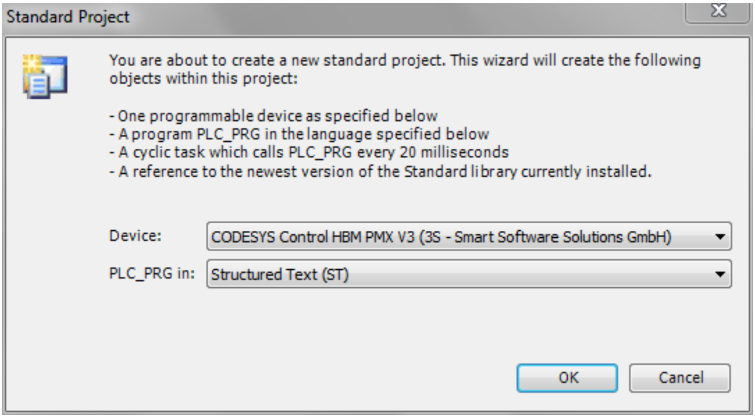
18.3 Preparation

- Install "Setup_CODESYS...exe".
- Start CODESYS with administrator rights. One way to do this, for example, is to hold down the Shift key, right-click on the icon, and choose **Run as administrator**.
- Install the PMX package:
  In the **Tools -> Package Manager -> Install** menu, locate and select the "hbmpmx.-package" file.
- Choose **Typical Installation**.
  The Package Manager now contains the PMX package:



18.4 Creating a project

- Choose **File -> New project -> Standard project**.
  As the device type select "CODESYS Control HBM PMX V3":

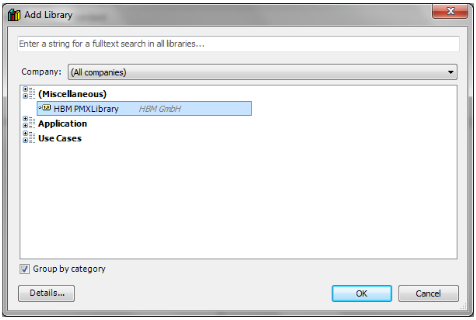


- After creating the project, double-click on **CODESYS Control HBM PMX V3**.

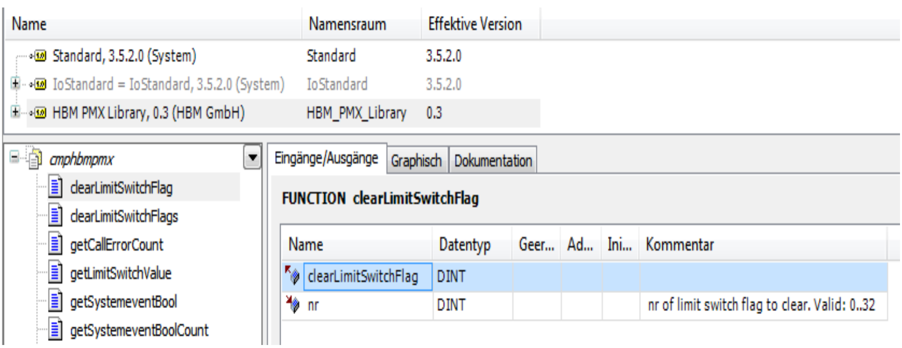
  Add a gateway in the communication settings as necessary. (The gateway type is usually "TCP/IP" if the PMX is connected to the PC via Ethernet. The IP address is "localhost", or use a fixed device address or the PMX device name).
- Select the gateway entry and click on **Browse network**. The target device should now be displayed. Double-click on it to make it the active device.
- Under **PLC_PRG (PRG)** you can now create the program.

18.5 Adding a PMX library

- Double-click on **Library Manager**, then **Add library**, and select **HBM PMXLibrary** under **Other**.



The library functions are explained in the online help, e.g.



18.6 PMX library

Description of functions of the PMX reference library, version 0.94.

Function: clearLimitSwitchFlag

Clears a limit switch flag

Name	Data type	Comments
clearLimitSwitchFlag	DINT	
nr	DINT	No of limit switch flag to clear. Valid: 0 … 32

Function: clearLimitSwitchFlags

Clears multiple limit switches

Name	Data type	Comments
clearLimitSwitchFlags	DINT	
mask	DWORD	Bitmask: every limit switch flag is cleared where corresponding bit is set

Function: diskfree

Returns the available memory space.

Name	Data type	Comments
diskfree	UDINT	Worst case estimate in bytes
disk	DINT	Disk number 0: user storage, 1 … 9: partition on usb-stick, 10 system partition

Function: GetCallErrorCount

Returns the number of errors that occurred in function calls that return a call handle. This function should always return zero in normal operation.

Name	Data type	Comments
getCallErrorCount	DINT	

Function: GetLimitSwitchValue

Returns the level for the limit switch. This is the value to which the limit switch flag is set.

Name	Data type	Comments
getLimitSwitchValue	REAL	Value of the limit switch
nr	DINT	Nr of the limit switch starting with 0

Function: GetShuntState

Returns the value of the shunt query started with startGetShuntState.

Name	Data type	Comments
getShuntState	DINT	0: shunt off, 1: shunt on, -1: error, -2: result not available, retry later
callHandle	DINT	The handle returned by startGetShuntState

Function: GetSystemeventBool

This function is only valid if Task -> External event -> System event has been selected. The number of arguments depends on the signal type determined by the "getSystemeventNr" function. This function returns the nth argument of type "BOOL" for the corresponding system event. Each PMX device has an xml file describing the valid system events that you can retrieve from the device via http://pmx/data/systemevent.xml.

Name	Data type	Comments
getSystemeventBool	BOOL	Value of the corresponding systemevent argument, false if invalid.
idx	DINT	Index value of the n-th bool of the systemevent. Possible values: 0 ≤ idx < 5 and idx<getSystemeventBoolCount()>.

Function: GetSystemeventBoolCount

This function is only valid if Task -> External event -> System event has been selected. The number of arguments depends on the signal type determined by the "getSystemeventNr" function. This function returns the available number of arguments of type "BOOL" for the current system event. Each PMX device has an xml file describing the valid system events that you can retrieve from the device via http://pmx/data/systemevent.xml.

Name	Data type	Comments
Get SystemeventBoolCount	BYTE	

Function: GetSystemeventByte

This function is only valid if Task -> External event -> System event has been selected. The number of arguments depends on the signal type determined by the "getSystemeventNr" function. This function returns the nth argument of type "BYTE" for the corresponding system event. Each PMX device has an xml file describing the valid system events that you can retrieve from the device via http://pmx/data/systemevent.xml.

Name	Data type	Comments
Get SystemeventByte	BYTE	
idx	DINT	Index value of the n-th Byte of the systemevent. Possible values: 0 ≤ idx < 5 and idx<getSystemeventByteCount()>

Function: GetSystemeventByteCount

This function is only valid if Task -> External event -> System event has been selected. The number of arguments depends on the signal type determined by the "getSystemeventNr" function. This function returns the available number of arguments of type "BYTE" for the current system event. Each PMX device has an xml file describing the valid system events that you can retrieve from the device via http://pmx/data/systemevent.xml.

Name	Data type	Comments
getSystemeventByteCount	DINT	

Function: GetSystemeventDInt

This function is only valid if Task -> External event -> System event has been selected. The number of arguments depends on the signal type determined by the "getSystemeventNr" function. This function returns the nth argument of type "DINT" for the corresponding system event. Each PMX device has an xml file describing the valid system events that you can retrieve from the device via http://pmx/data/systemevent.xml.

Name	Data type	Comments
getSystemeventDint	DINT	
	DINT	Index value of the n-th DINT of the systemevent. Possible values: 0 ≤ idx < 5 and idx<getSystemeventDIntCount()>.

Function: GetSystemeventDIntCount

This function is only valid if Task -> External event -> System event has been selected. The number of arguments depends on the signal type determined by the "getSystemeventNr" function. This function returns the available number of arguments of type "DINT" for the current system event. Each PMX device has an xml file describing the valid system events that you can retrieve from the device via http://pmx/data/systemevent.xml.

Name	Data type	Comments
clearLimitSwitchFlag		

Function getSystemeventInt

This function is only valid if Task -> External event -> System event has been selected. The number of arguments depends on the signal type determined by the "getSystemeventNr" function. This function returns the nth argument of type "INT" for the corresponding system event. Each PMX device has an xml file describing the valid system events that you can retrieve from the device via http://pmx/data/systemevent.xml.

Name	Data type	Comments
getSystemeventInt	INT	
idx	DINT	Index value of the n-th Int of the systemevent. Possible values: 0 ≤ idx < 5 and idx<getSystemeventIntCount()>.

Function getSystemeventIntCount

This function is only valid if Task -> External event -> System event has been selected. The number of arguments depends on the signal type determined by the "getSystemeventNr" function. This function returns the available number of arguments of type "INT" for the current system event. Each PMX device has an xml file describing the valid system events that you can retrieve from the device via http://pmx/data/systemevent.xml.

Name	Data type	Comments
getSystemeventIntCount	DINT	

Function: GetSystemeventLInt

This function is only valid if Task -> External event -> System event has been selected. The number of arguments depends on the signal type determined by the "getSystemeventNr" function. This function returns the nth argument of type "LINT" for the corresponding system event. Each PMX device has an xml file describing the valid system events that you can retrieve from the device via http://pmx/data/systemevent.xml.

Name	Data type	Comments
getSystemeventInt	INT	
idx	DINT	Index value of the n-th Int of the systemevent. Possible values: 0 ≤ idx < 5 and idx<getSystemeventIntCount()>.

Function: GetSystemeventLIntCount

This function is only valid if Task -> External event -> System event has been selected. The number of arguments depends on the signal type determined by the "getSystemeventNr" function. This function returns the available number of arguments of type "LINT" for the current system event. Each PMX device has an xml file describing the valid system events that you can retrieve from the device via http://pmx/data/systemevent.xml.

Name	Data type	Comments
getSystemeventInt-Count	DINT	

Function: GetSystemeventLReal

This function is only valid if Task -> External event -> System event has been selected. The number of arguments depends on the signal type determined by the "getSystemeventNr" function. This function returns the nth argument of type "REAL" for the corresponding system event. Each PMX device has an xml file describing the valid system events that you can retrieve from the device via http://pmx/data/systemevent.xml.

Name	Data type	Comments
getSystemeventLReal	LREAL	Value of the corresponding byte, 0 if invalid
idx	DINT	Index value of the n-th Real of the systemevent. Possible values: 0 ≤ idx < 5 and idx<getSystemeventRealCount()>.

Function: GetSystemeventLRealCount

This function is only valid if Task -> External event -> System event has been selected. The number of arguments depends on the signal type determined by the "getSystemeventNr" function. This function returns the available number of arguments of type "REAL" for the current system event. Each PMX device has an xml file describing the valid system events that you can retrieve from the device via http://pmx/data/systemevent.xml.

Name	Data type	Comments
getSystemeventLRealCount	DINT	

Function: GetSystemeventNr

This function is only valid if Task -> External event -> System event has been selected. This function returns the system event number for the system event in question. Each PMX device has an xml file describing the valid system events that you can retrieve from the device via http://pmx/data/systemevent.xml.

Name	Data type	Comments
getSystemeventNr	DINT	

Function: GetSystemeventString

This function is only valid if Task -> External event -> System event has been selected. The number of arguments depends on the signal type determined by the "getSystemeventNr" function. This function returns the nth argument of type "STRING" for the corresponding system event. Each PMX device has an xml file describing the valid system events that you can retrieve from the device via http://pmx/data/systemevent.xml.

Name	Data type	Comments
getSystemeventString	STRING	
idx	DINT	Index value of the n-th String of the systemevent. Possible values: 0 ≤ idx < 5 and idx<getSystemeventStringCount()>.

Function: GetSystemeventStringCount

This function is only valid if Task -> External event -> System event has been selected. The number of arguments depends on the signal type determined by the "getSystemeventNr" function. This function returns the available number of arguments of type "STRING" for the current system event. Each PMX device has an xml file describing the valid system events that you can retrieve from the device via http://pmx/data/systemevent.xml.

Name	Data type	Comments
getSystemeventString-Count	DINT	

Function: GetSystemeventUDInt

This function is only valid if Task -> External event -> System event has been selected. The number of arguments depends on the signal type determined by the "getSystemeventNr" function. This function returns the nth argument of type "UDINT" for the corresponding system event. Each PMX device has an xml file describing the valid system events that you can retrieve from the device via http://pmx/data/systemevent.xml.

Name	Data type	Comments
getSystemeventUDint	UDINT	
idx	DINT	Index value of the n-th UDint of the systemevent. Possible values: 0 ≤ idx < 5 and idx<getSystemeventUDIntCount()>.

Function: GetSystemeventUDIntCount

This function is only valid if Task -> External event -> System event has been selected. The number of arguments depends on the signal type determined by the "getSystemeventNr" function. This function returns the available number of arguments of type "UDINT" for the current system event. Each PMX device has an xml file describing the valid system events that you can retrieve from the device via http://pmx/data/systemevent.xml.

Name	Data type	Comments
getSystemeventUDint-Count	DINT	

Function: GetSystemeventUInt

This function is only valid if Task -> External event -> System event has been selected. The number of arguments depends on the signal type determined by the "getSystemeventNr" function. This function returns the nth argument of type "UINT" for the corresponding system event. Each PMX device has an xml file describing the valid system events that you can retrieve from the device via http://pmx/data/systemevent.xml.

Name	Data type	Comments
getSystemeventUInt	UINT	
idx	DINT	Index value of the n-th UIint of the systemevent. Possible values: 0 ≤ idx < 5 and idx<getSystemeventUIntCount()>.

Function: GetSystemeventUIntCount

This function is only valid if Task -> External event -> System event has been selected. The number of arguments depends on the signal type determined by the "getSystemeventNr" function. This function returns the available number of arguments of type "UINT" for the current system event. Each PMX device has an xml file describing the valid system events that you can retrieve from the device via http://pmx/data/systemevent.xml.

Name	Data type	Comments
getSystemeventUInt-Count	DINT	

Function: GetSystemeventULInt

This function is only valid if Task -> External event -> System event has been selected. The number of arguments depends on the signal type determined by the "getSystemeventNr" function. This function returns the nth argument of type "ULINT" for the corresponding system event. Each PMX device has an xml file describing the valid system events that you can retrieve from the device via http://pmx/data/systemevent.xml.

Name	Data type	Comments
getSystemeventULint	ULINT	
idx	DINT	Index value of the n-th ULint of the systemevent. Possible values: 0 ≤ idx < 5 and idx<getSystemeventUIntCount()>.

Function: GetSystemeventULIntCount

This function is only valid if Task -> External event -> System event has been selected. The number of arguments depends on the signal type determined by the "getSystemeventNr" function.
This function returns the available number of arguments of type "ULINT" for the current system event. Each PMX device has an xml file describing the valid system events that you can retrieve from the device via http://pmx/data/systemevent.xml.

Name	Data type	Comments
getSystemeventULint-Count	DINT	

Function: IsFinished

The status of the call can be queried for all functions that return a handle. The function returns TRUE when the corresponding function call to the handle is completed.

Name	Data type	Comments
callHandle	DINT	Handle of the corresponding function call e.g. recalibrate.

Function: Recalibrate

Recalibrates the hardware channel concerned. This function is only effective if the channel is occupied and has calibration hardware.

Name	Data type	Comments
recalibrate	DINT	
slot	DINT	Slot of the channel to calibrate (valid 1 … 4 depending on hardware).
signal	DINT	Signal of the channel to calibrate. Count starts with 1.

Function: set2PointCharacteristic

Sets a two-point characteristic for a signal.

Name	Data type	Comments
Set2Pointcharacteristic	DINT	
slot	DINT	
signal	DINT	(* slot of the hardware (valid 1 … 4 depending on hardware)*)

Name	Data type	Comments
Point1electrical	REAL	(* signal nr of the hw slot (valid 1 … 4 depending on hardware)*)
Point1physikal	REAL	(* 1. point electrical value*)
Point2electrical	REAL	(* 1. point physical value*)
Point2physical	REAL	(* 1. point electrical value*)

Function: setHoldPeak

This function holds or releases a peak value.

Name	Data type	Comments
setHoldPeak	DINT	
slot	DINT	Slot of peak value (valid 1 … 4 depending on hardware)
signal	DINT	Signal of peak value (valid 1 … 4 depending on hardware)
hold	BOOL	hold = true; run = false

Function: setLimitswitchValue

Sets the limit switch value. The limit switch value is the value to which the corresponding limit switch flag is set.

Name	Data type	Comments
setLimitswitchValue	DWORD	
nr	DINT	Nr of the limit switch starting with 0
value	REAL	New value of the limit switch

Function: SetParameterSet

Sets the current parameter set. The current parameter set is available via the HBM PMX CODESYS I/O. The parameter sets must be configured in advance via the web interface. This function returns a handle that can be queried via the "isFinished" function. It is nevertheless possible that the parameter switching might not yet be finished when this call is completed, as this function only triggers its start. Use system event with event number = 2000 to check via a trigger whether the parameter set switch was successful.

Name	Data type	Comments
setParameterSet	DINT	
paremeternr	DINT	The parameter of the desired parameterset

Function: SetResetPeak

Resets the peak value. This function should be called twice to perform a complete reset cycle.

Name	Data type	Comments
setResetPeak	DINT	
slot	DINT	Slot of peak value (valid 1 … 4 depending on hardware).
signal	DINT	Signal of peak value (valid 1 … 4 depending on hardware).
reset	BOOL	True: peak is held in reset, false: peak block operates

Function: SetShuntState

Sets the shunt state of a signal.

Name	Data type	Comments
setShuntState	DINT	Handle: check with isFinished(handle)
slot	DINT	The slot to modify, valid 1 … 4 and Cardtype PX460 only
signal	DINT	The signal to modify, valid 2, 4
shunt	DINT	The new shunt value off=0, on=1

Function: SetToZero

Sets the offset so that zero applies to the measured value. Note that this function impacts on the current parameter set. It is reversed via "setUserOffset(...,0.0)".

Name	Data type	Comments
setToZero	DINT	
slot	DINT	Slot of corresponding measval (valid 1 ... 4)
signal	DINT	Signal of corresponding measval (valid 1 ... 4 depending on hardware)

Function: setUserOffset

Set a user-defined measurement offset. Note that this function impacts on the current parameter set.

Name	Data type	Comments
setUserOffset	DINT	
slot	DINT	Slot of corresponding measval, use 9 for computed channels
signal	DINT	Signal of corresponding measval, starting with 1
offset	REAL	The new offset value

Function: setZeroTargetValue

By setting a target value for zero, a constant can be added to a currently measured value for a specified signal.

Name	Data type	Comments
startZeroTargetValue	DINT	
slot	DINT	
signal	DINT	(* slot of the hardware (should be 9 for calculated channels) *)
value	DINT	(* nr of the calculated channel *)

Function: startGetShuntState

Initiates a change in the shunt state of a PX460.

Name	Data type	Comments
startGetShuntState	DINT	Handle: query with getShuntState(handle)
slot	DINT	Sthe slot to modify, valid 1 … 4 and Card PX460 only
signal	DINT	The signal to modify valid 2,4

Function: startLedEffect

Various LED effects, e.g. for locating the PMX device or for feedback to the user in front of the device.

Name	Data type	Comments
startLedEffect	DINT	Handle which can be queried by isFinished
tinInSeconds	DINT	The duration in seconds of the effect
effect	DINT	Effect type: 0 green running, 1 yellow running, 2 red running, 3 green blink, 4 yellow blink, 5 red blink

Function: setTedsSetup

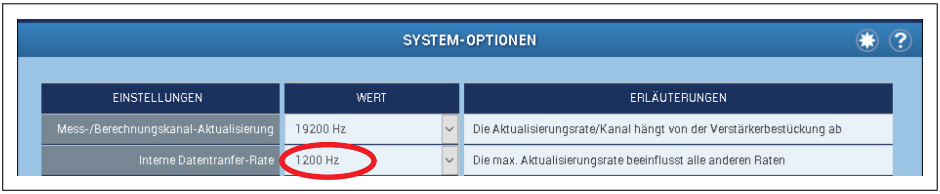
Reinitializes the TEDS setup for the specified channel.

Name	Data type	Comments
startTedsSetup	DINT	
slot	DINT	Slot of the hardware (valid 1 … 4 depending on hardware)
signal	DINT	Signal nr of the hw slot (valid 1 … 4 depending on hardware)

18.7 Task configuration

The following task types can be selected under MainTask:

- **Cyclic**
  The task is started asynchronously to the measured values. *The interval should be at least 4 ms*, which is the shortest possible resolution.
- **External – Measval Event**
  The task is started synchronously with the acquired measured values. You set the call frequency in the **System Options** dialog. Default setting: 1200 Hz, i.e. at a sample rate of 19200 Hz the task is started after every 16th measured value.



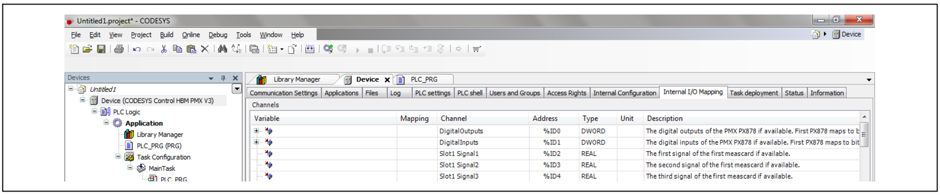
- **External – SystemEvent**
  The task is started when a PMXSystemEvent occurs (these events are also displayed in the system log of the device). The number of the event is supplied in the task by the library function getSystemeventNr (see section *section 18.10, "System events for PMX", page 313*).
  You call the system events in the browser from the device path http://<pmx device-name>/data/systemevent.xml.

Select only these task types if possible.

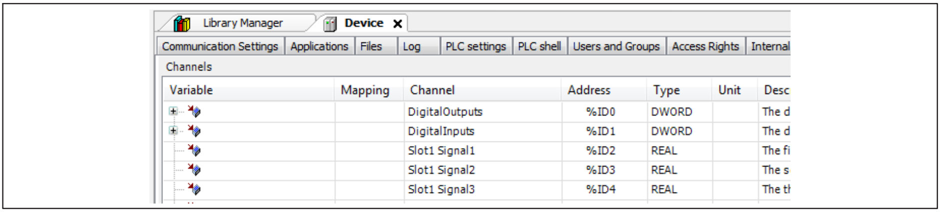
18.8 Cyclic data

The cyclic data exchanged with the PMX firmware is displayed like this:

▶ Double-click on the links in the **CODESYS Control HBM PMX V3** project tree. Select the **Internal I/O map** tab.



▶ To connect to an existing program variable of the same type, double-click in the desired cell in the **Variable** column.

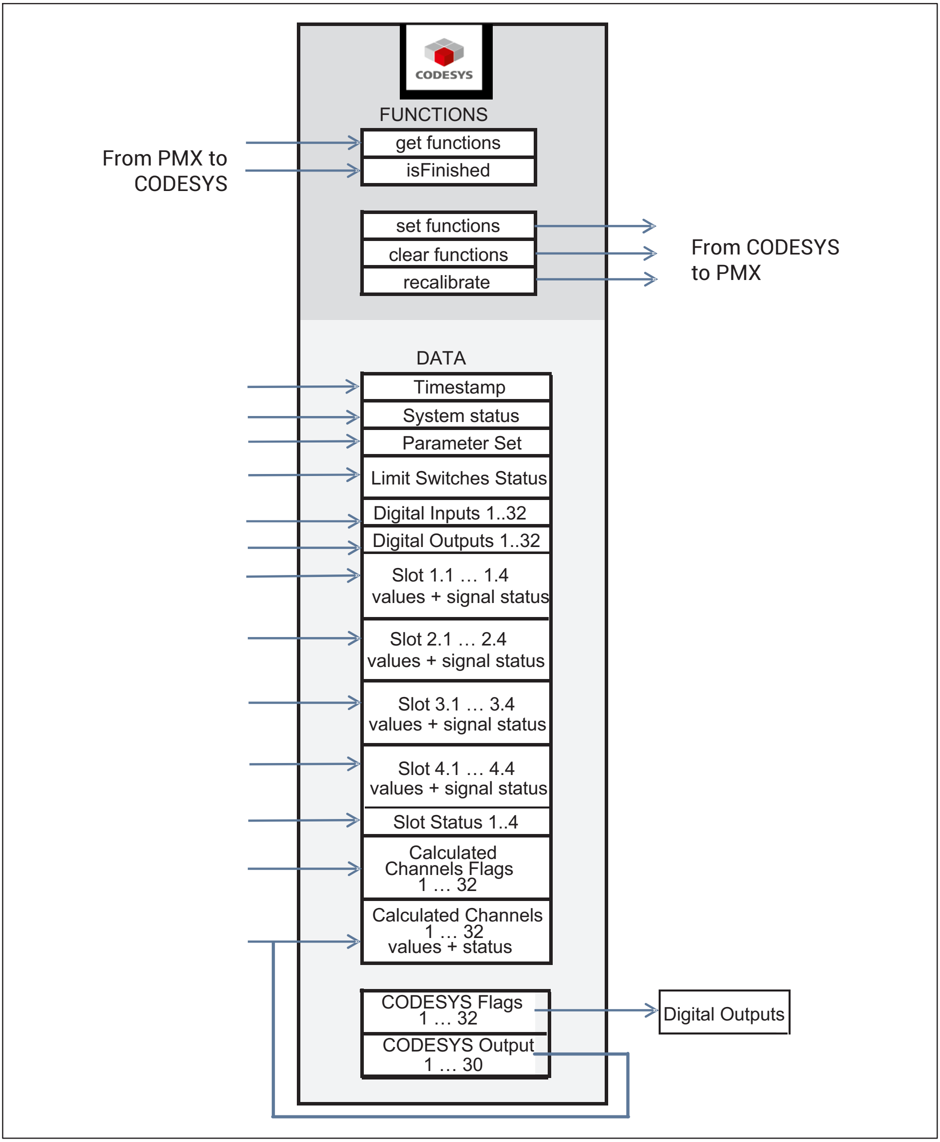


18.9 Signal diagram (I/O Mapping)

In I/O Mapping, all the incoming signals from the PMX can be mapped into the CODESYS application and back out of the application into the PMX.

Notice

The PMX Web Server allocates additional functions (e.g. use in the analog output or in calculation channels) to the incoming signals from the CODESYS application.


CODESYS
FUNCTIONS
From PMX to CODESYS
get functions
isFinished
set functions
clear functions
recalibrate
From CODESYS to PMX
DATA
Timestamp
System status
Parameter Set
Limit Switches Status
Digital Inputs 1..32
Digital Outputs 1..32
Slot 1.1 … 1.4 values + signal status
Slot 2.1 … 2.4 values + signal status
Slot 3.1 … 3.4 values + signal status
Slot 4.1 … 4.4 values + signal status
Slot Status 1..4
Calculated Channels Flags 1 … 32
Calculated Channels 1 … 32 values + status
CODESYS Flags 1 … 32
Digital Outputs
CODESYS Output 1 … 30

18.10 System events for PMX

Important

PMX system events can only be called in tasks that were started as an external event/ system event. To display the events list, enter the following address in the browser bar: http://<pmx>/data/systemevent.xml/; <pmx> stands for the URL of the PMX device.

18.10.1 All

ID: **1** Name:propertyChanged
Argument:dbusInterface Type:string
Argument:value Type:variant
Argument:serviceName Type:string
Description: Property changed Service:%3, Value:%2 %1

ID: **2** Name:valueCorrected
Argument:dbusInterface Type:string
Argument:newValue Type:string
Argument:serviceName Type:string
Description: Value changed to %2, Interface:%1, Service:%3

ID: **42** Name:serviceAdded
Argument:serviceName Type:string
Description: Service started:%1

ID: **43** Name:serviceRemoved
Argument:serviceName Type:string
Description: Service stopped:%1

18.10.2 com.hbm.fwconfig

ID: **1000** Name:Firmwareaktualisierung
Argument:state Type:enum
Description: firmware update in progress: %1

ID: **1001** Name:testmessage
Argument:integer Type:int32
Argument:string Type:string
Description: test message: integer:%1 string:%2

ID: **1002** Name:firmwareDeleted
Description: Firmware has been deleted.

ID: **1010** Name:networkAddressChange
Argument:address Type:string
Description: Network address change to %1

ID: **1011** Name:deviceNameChange
Argument:name Type:string
Description: Device name changed to %1

ID: **1012** Name:hostnameInvalid
Argument:invalidHostname Type:string
Argument:validHostname Type:string
Description: Given Hostname %1 is invalid. Keeping %2 as Hostname.

ID: **1013** Name:pwResetVerifyFailed
Description: Administrator password reset failed: invalid signature!

ID: **1014** Name:pwResetFileError
Description: Administrator password reset failed: file operation failed!

ID: **1015** Name:pwResetFormatError
Description: Administrator password reset failed: file format invalid!

ID: **1016** Name:pwResetHostnameError
Description: Administrator password reset failed: hostname does not match!

ID: **1017** Name:pwResetMacError
Description: Administrator password reset failed: mac does not match!

ID: **1018** Name:pwResetSuccessful
Description: Administrator password reset successful!

ID: **1019** Name:codesysFileRemoved
Argument:deletedCODESYSFile Type:string
Description: The codesys application file:%1 has been deleted!

ID: **1111** Name:reboot
Description: PMX is rebooting

18.10.3 com.hbm.parameter

ID: **2000** Name:parameterChanged
Argument:oldParameterNr Type:int32
Argument:parameterNr Type:int32
Argument:jsonCurrentDomains Type:string
Description: parameter set changed from %1 to %2. Subdomains(%3)

ID: **2001** Name:parameterInconsistent
Argument:index Type:int32
Argument:correctedDomainindex Type:int32
Description: Inconsistent parameterset #%1 loaded. Setting to %2

ID: **2002** Name:parameterErrorCantDeleteLastParameter
Description: The last parameter must not be deleted!

ID: **2003** Name:parameterErrorCantDeleteLastDomain
Argument:domainName Type:string
Description: The last domain:%1 must not be deleted!

ID: **2004** Name:parameterErrorParameterSwitchIsLocked
Description: Parameter switching is locked! Could not switch parameters.

ID: **2005** Name:parameterSwitchFailed
Argument:failedServices Type:string
Description: Parameter switching failed. Failed services:%1

ID: **2006** Name:parameterDeleteDomainNotFound
Argument:domain Type:string
Argument:domainnr Type:int32
Description: %1: deleting domain %2 failed: Not found!

ID: **2007** Name:parameterDeleteDomainInUse
Argument:domain Type:string
Argument:domainnr Type:int32
Description: %1: deleting domain %2 failed: In use!

ID: **2008** Name:parameterInvalidName
Argument:name Type:string
Description: Invalid name "%1": slashes not allowed.

ID: **2009** Name:parameterInvalidNameExists
Argument:name Type:string
Description: Invalid name "%1": Name exists.

ID: **2010** Name:parameterDomainNotExists
Description: Domain does not exists.

ID: **2011** Name:parameterCantDeleteCurrent
Description: Can't delete current parameter.

ID: **2012** Name:parameterCantDeleteBootup
Description: Can't delete bootup parameter.

ID: **2013** Name:parameterListChanged
Description: Parameter list has changed.

ID: **2014** Name:subparameterListChanged
Description: Parameter list has changed.

18.10.4 com.hbm.fpgasrv

ID: **3000** Name:powerOverload
Argument:status Type:string
Argument:cardNr Type:int32
Description: %1Card %2: Power Overload

ID: **3001** Name:adcPhaseError
Argument:status Type:string
Argument:cardNr Type:int32
Description: %1Card %2: ADC Phase Error. This may break measurement values. Electrostatic discharge? Damaged card?

ID: **3002** Name:stuckInOverflow
Argument:status Type:string
Argument:cardNr Type:int32
Argument:channel Type:int32
Description: %1Card %2, channel %3: Stuck in overflow

ID: **3003** Name:forcedSyncModeSet
Argument:type Type:string
Description: The user forces the device to be %1.

ID: **3004** Name:forcedSyncModeReleased
Description: User's forced sync mode disabled. Back to automatic sync mode.

ID: **3005** Name:syncUnlocked
Description: Not locked to incoming sync signal.

ID: **3006** Name:syncLocked
Description: Locked to incoming sync signal.

ID: **3007** Name:syncCannotLock
Argument:type Type:string
Description: %1Cannot lock to incoming sync signal.

ID: **3008** Name:syncAvailableSlaveMode
Description: Sync available. Switching to slave mode.

ID: **3009** Name:crcErrorsMasterMode
Argument:type Type:string
Description: %1Too many CRC errors on sync input. Temporarily switching to master mode.

ID: **3010** Name:noSyncSlaveMode
Description: The user forced this device to be slave, but it has no valid sync input.

ID: **3011** Name:noSyncMasterMode
Description: No sync input. Switching to master mode.

ID: **3012** Name:PX460FPGAfailure
Description: The PX460 FPGA chip stopped and will be reconfigured. ESD event? Power problem?.

18.10.5 com.hbm.SysCfgMgr

ID: **4000** Name:wrongSensorType
Argument:slot Type:int32
Argument:signal Type:int32
Argument:sensortype Type:int32
Description: Wrong or unsupported sensortype. Slot:%1, Signal:%2, Sensortype:%3

ID: **4020** Name:measvalStatus
Argument:slot Type:int32
Argument:signal Type:int32
Argument:statusText Type:string
// "valid" or "invalid" Description: Measval-status changed. New status: '%3'. Slot:%1, Signal:%2

ID: **4040** Name:sensorSupplyOverloadStatus
Argument:statusText Type:string
// "activated" or "deactivated" Description: System status Sensor-Power-Output-Overload has been %1'

ID: **4042** Name:bufferOverflowStatus
Argument:statusText Type:string
// "activated" or "deactivated" Description: System status Command-interface-buffer-overflow has been %1'

ID: **4044** Name:factorySettingsStatus
Argument:statusText Type:string
// "activated" or "deactivated" Description: System status Factory-Settings-Error has been %1'

ID: **4046** Name:datalogActiveStatus
Argument:statusText Type:string
// "activated" or "deactivated" Description: System status Datalogger-Ready has been %1'

ID: **4048** Name:datalogErrorStatus
Argument:statusText Type:string
// "activated" or "deactivated" Description: System status Datalogger-Ready has been %1'

ID: **4050** Name:datalogBufOvrStatus
Argument:statusText Type:string
// "activated" or "deactivated" Description: System status Datalogger-Buffer-Overrun has been %1'

ID: **4052** Name:datalogBuf50Percent
Argument:statusText Type:string
// "activated" or "deactivated" Description: System status Datalogger-Buffer-50% has been %1'

ID: **4100** Name:tedsBitlenErr
Argument:slot Type:int32
Argument:signal Type:int32
Argument:currentBitpos Type:int32
Argument:totalBitlen Type:int32
Description: TedsParser: Current TEDS bitposition is too big. Slot:%1, Signal:%2, current bitpos.:%3, total bitlen.:%4

ID: **4102** Name:tedsUnsupportedManufacturerID
Argument:slot Type:int32
Argument:signal Type:int32
Argument:manufacturerID Type:int32
Description: TedsParser: Unsupported manufacturer ID. Slot:%1, Signal:%2, manufacturer ID:%3

ID: **4104** Name:tedsUnsupportedTemplateIDorSelector
Argument:slot Type:int32
Argument:signal Type:int32
Argument:templateID Type:int32
Argument:selector Type:int32
Description: TedsParser: Unsupported template ID. Slot:%1, Signal:%2, template ID:%3, selector ID:%4

ID: **4106** Name:tedsUnknownIEEETemplate
Argument:slot Type:int32
Argument:signal Type:int32
Argument:templateID Type:int32
Description: TedsParser: Unknown IEEE template. Slot:%1, Signal:%2, template ID:%3

ID: **4108** Name:tedsUnknownHbmTemplate
Argument:slot Type:int32
Argument:signal Type:int32
Argument:templateID Type:int32
Description: TedsParser: Unknown HBM template. Slot:%1, Signal:%2, template ID:%3

ID: **4110** Name:tedsEmbeddedTemplateNotSupported
Argument:slot Type:int32
Argument:signal Type:int32
Description: TedsParser: Embedded template not supported. Slot:%1, Signal:%2

ID: **4112** Name:tedsTemplateError
Argument:slot Type:int32
Argument:signal Type:int32
Description: TedsParser: Template error. Slot:%1, Signal:%2

ID: **4114** Name:tedsUnknownSelector
Argument:slot Type:int32
Argument:signal Type:int32
Description: TedsParser: unknown TEDS selector. Slot:%1, Signal:%2

ID: **4120** Name:tedsNoValidData
Argument:slot Type:int32
Argument:signal Type:int32
Description: TedsParser: No valid TEDS data. Slot:%1, Signal:%2

ID: **4122** Name:tedsNotFound
Argument:slot Type:int32
Argument:signal Type:int32
Description: TedsParser: No TEDS available or not found. Slot:%1, Signal:%2

ID: **4124** Name:tedsNoDataToWrite
Argument:slot Type:int32
Argument:signal Type:int32
Description: Teds: No TEDS data available. Slot:%1, Signal:%2

ID: **4130** Name:tedsSaveUsageFailed
Argument:slot Type:int32
Argument:signal Type:int32
Description: TEDS: Save param 'usage' failed. Slot:%1, Signal:%2

ID: **4132** Name:tedsSaveConvertUnitFailed
Argument:slot Type:int32
Argument:signal Type:int32
Description: TEDS: Save param 'convert unit to device unit' failed. Slot:%1, Signal:%2

ID: **4134** Name:tedsSaveParamsFailed
Argument:slot Type:int32
Argument:signal Type:int32
Description: TEDS: Save parameters failed. Slot:%1, Signal:%2

ID: **4140** Name:tedsDestUnitUnknown
Argument:destUnit Type:int32
Argument:slot Type:int32
Argument:signal Type:int32
Description: TEDS: Destination Unit code %1 not found. Slot:%2, Signal:%3

ID: **4142** Name:tedsUnitConversionFailed
Argument:slot Type:int32
Argument:signal Type:int32
Description: TEDS: Unit conversion failed. Slot:%1, Signal:%2

ID: **4144** Name:tedsCantGetUnitcode
Argument:slot Type:int32
Argument:signal Type:int32
Description: TEDS: Can't get current unitcode. Slot:%1, Signal:%2

ID: **4150** Name:tedsConfigurationOK
Argument:slot Type:int32
Argument:signal Type:int32
Description: TEDS: channel configuration OK. Slot:%1, Signal:%2

ID: **4152** Name:tedsConfigurationFailed
Argument:slot Type:int32
Argument:signal Type:int32
Description: TEDS: channel configuration failed. Slot:%1, Signal:%2

ID: **4160** Name:tedsConfigHbmPulseFailed
Argument:slot Type:int32
Argument:signal Type:int32
Description: TEDS: sensor configuration for 'HBM pulse' failed. Slot:%1, Signal:%2

ID: **4162** Name:tedsConfigIeeeLvdtExcFreqFailed
Argument:slot Type:int32
Argument:signal Type:int32
Description: TEDS: sensor configuration for 'IeeeLvdt' failed (exc.frequ. or ampl.). Slot:%1, Signal:%2

ID: **4164** Name:tedsConfigWrongCardtype
Argument:slot Type:int32
Argument:signal Type:int32
Description: TEDS: sensor configuration failed, sensortype not supported from this measurement card. Slot:%1, Signal:%2

ID: **4166** Name:tedsConfigIeeeBridgeFailed
Argument:slot Type:int32
Argument:signal Type:int32
Description: TEDS: sensor configuration for 'HBM pulse' failed. Sensitivity, excitation voltage or bridge resistors not suitable for Slot:%1, Signal:%2

ID: **4168** Name:tedsConfigSensorFailed
Argument:slot Type:int32
Argument:signal Type:int32
Description: TEDS: sensor configuration failed. Slot:%1, Signal:%2

ID: **4170** Name:tedsConfigHbmDisplExcFreqFailed
Argument:slot Type:int32
Argument:signal Type:int32
Description: TEDS: sensor configuration failed because of exc-frequency or amplitude. Slot:%1, Signal:%2

ID: **4180** Name:tedsConfigScalingOK
Argument:slot Type:int32
Argument:signal Type:int32
Argument:physSignalVal_x1 Type:double
Argument:usrVal_y1 Type:double
Argument:physSignalVal_x2 Type:double
Argument:usrVal_y2 Type:double
Description: TEDS: scaling configuration OK. Slot:%1, Signal:%2, Scaling: physSignal-Val_x1:%3, usrVal_y1:%4 ; physSignalVal_x2:%5, usrVal_y2:%6

ID: **4182** Name:tedsConfigScalingFailed
Argument:slot Type:int32
Argument:signal Type:int32
Description: TEDS: scaling configuration failed. Slot:%1, Signal:%2

ID: **4190** Name:tedsConfigHpFilterNotSupported
Argument:slot Type:int32
Argument:signal Type:int32
Description: TEDS: highpass filter configuration not supported. Slot:%1, Signal:%2

ID: **4192** Name:tedsConfigFilterCharactFailed
Argument:slot Type:int32

Argument:signal Type:int32
Description: TEDS: setting filter characteristic failed. Slot:%1, Signal:%2

ID: **4194** Name:tedsConfigFilterCutOffAdapted
Argument:slot Type:int32
Argument:signal Type:int32
Description: TEDS: cut off frequency adapted. Slot:%1, Signal:%2

ID: **4196** Name:tedsConfigTaraNotSupported
Argument:slot Type:int32
Argument:signal Type:int32
Description: TEDS: tare configuration not supported. Slot:%1, Signal:%2

ID: **4198** Name:tedsConfigUCCfailed
Argument:slot Type:int32
Argument:signal Type:int32
Description: TEDS: user channel comment configuration failed. Slot:%1, Signal:%2

ID: **4200** Name:tedsSkipCalCurve
Argument:slot Type:int32
Argument:signal Type:int32
Description: TEDS: cal curve ignored. Slot:%1, Signal:%2

ID: **4202** Name:tedsSkipCalTable
Argument:slot Type:int32
Argument:signal Type:int32
Description: TEDS: cal table ignored. Slot:%1, Signal:%2

ID: **4300** Name:changeShuntStat
Argument:slot Type:int32
Argument:signal Type:int32
Argument:shuntStat Type:int32
Description: Slot:%1, Signal:%2 shunt state changed:%3

18.10.6 com.hbm.storagemanager

ID: **5000** Name:saveStarted
Argument:filename Type:string
Description: Started saving to file %1

ID: **5001** Name:saveFinished
Argument:filename Type:string
Description: Saved systemstate to file %1

ID: **5002** Name:restoreStarted
Argument:filename Type:string
Argument:systemrestore Type:bool
Description: Started restore from file %1

ID: **5003** Name:restoreFinished
Argument:filename Type:string

Argument:systemrestore Type:bool
Description: Restored systemstate from file %1

ID: **5004** Name:systemdefaultsUploaded
Argument:filename Type:string
Description: Systemdefaults uploaded %1

ID: **5005** Name:hashFailed
Argument:filename Type:string
Description: md5 hash failed for %1

18.10.7 com.hbm.sigproc

ID: **6002** Name:noMoreDspSignalsAvail
Description: No more internal signals available."

ID: **6003** Name:noMoreCalcedChannelAvail
Description: No more calculated channels available."

ID: **6050** Name:blockNotSupported
Argument:blockNbr Type:int32
Description: Block type %1 is not supported.

ID: **6051** Name:blockCreated
Argument:type Type:string
Argument:calcOrder Type:int32
Description: Function block '%1' at calculation rank %2 created.

ID: **6052** Name:blockDeleted
Argument:type Type:string
Description: Function block '%1' deleted.

ID: **6053** Name:calcChanCreated
Argument:channelNbr Type:int32
Description: Calculated channel %1 #%2 created.

ID: **6054** Name:calcChanDeleted
Argument:channelNbr Type:int32
Description: Calculated channel %1 #%2 deleted.

ID: **6055** Name:tooManyFunctionBlocks
Description: Too many function blocks.

ID: **6100** Name:calcChanRuntimeOverrun
Description: Calculated channels runtime overrun.

ID: **6200** Name:setToZero
Argument:slot Type:int32
Argument: signal Type:int32
Argument: newOffset Type:double
Description: slot %1.%2 zero value=%3

18.10.8 com.hbm.fieldbus

ID: **7001** Name:fieldbusRestart
Argument:bustype Type:string
Description: %1 is restarting.

ID: **7002** Name:fieldbusFatalFault
Description: Fieldbus fatal fault. Device restart required.

ID: **7050** Name:txedCalculatedChans
Argument:chanCount Type:int32
Description: %1 calculated channels transmitted on fieldbus.

18.10.9 com.hbm.CatmanServer

ID: **8001** Name:test
Argument:cat_is_goil Type:int32
Description: %1 is here.

ID: **8002** Name:oldConnectionTerminated
Argument:conCount Type:int32
Argument:timeInSeconds Type:int32
Description: More than %1 Eth. connections requested. Oldest terminated. Last activity %2s ago.

ID: **8003** Name:newConnectionEstablished
Description: New Eth. Connection on port 55000 established.

ID: **8004** Name:connectionClosed
Description: Eth. connection closed.

18.10.10 com.hbm.meassrv

ID: **9001** Name:bufferOverrun
Description: Buffer overrun occurred.

18.10.11 com.hbm.httpdata

ID: **10001** Name:newSession
Argument:session Type:int32
Argument:address Type:string
Description: New session id:%1 address:%2.

ID: **10002** Name:closedSession
Argument:session Type:int32
Argument:address Type:string
Description: Closed session id:%1 address:%2.

18.10.12 GUI

ID: **11001** Name:dialogOpened
Argument:session Type:int32
Argument:dialogname Type:string
Description: Session id:%1 Dialog opened: %2.

ID: **11002** Name:dialogClosed
Argument:session Type:int32
Argument:dialogname Type:string
Description: Session id:%1 Dialog closed: %2.

ID: **11003** Name:viewOpened
Argument:session Type:int32
Argument:viewname Type:string
Description: Session id:%1 View opened: %2.

ID: **11004** Name:viewClosed
Argument:session Type:int32
Argument:viewname Type:string
Description: Session id:%1 View closed: %2.

ID: **11005** Name:UserLevelChanged
Argument:session Type:int32
Argument:userlevel Type:string
Description: Session id:%1 userlevel changed to %2.

ID: **11100** Name:calibrationAssist
Argument:slot Type:int32
Argument:signal Type:int32
Argument:msg Type:string
Description: CalibrationAssist: Slot:%1 signal:%2%3

18.10.13 com.hbm.DataLogger

ID: **12001** Name:testLogger
Argument:log_baby_log Type:int32
Argument:type Type:string
Description: Log it!

ID: **12002** Name:createServiceFailed
Description: Creating data logger measservice failed.

ID: **12005** Name:maxFilecountReached
Argument:fileCount Type:int32
Description: Max filecount in directory reached (%1). Logging stopped.

ID: **12006** Name:storageMediaFull
Description: Data logger storage media is full. Logging stopped.

ID: **12007** Name:loggingStarted
Description: Data logging started.

ID: **12008** Name:logging
Description: Logging data.

ID: **12009** Name:openingFileFailed
Argument:errcode Type:int32
Argument:errstr Type:string
Description: Opening datalogger file failed. Code %1:%2. Try again.

ID: **12010** Name:erasingOldestFileNoPar
Argument:filename Type:string
Description: Erasing oldest file %1.

ID: **12011** Name:erasingOldestFileNoPar
Description: Erasing oldest file.

ID: **12012** Name:erasingOldestFileFailed
Argument:filename Type:string
Argument:errcode Type:int32
Argument:errstr Type:string
Description: Erasing oldest file %1 failed. ErrCode %2:%3.

ID: **12014** Name:closeFile
Description: Close datalogger file.

ID: **12015** Name:writeError
Argument:errcode Type:int32
Argument:errstr Type:string
Description: Writing to datalogger file failed. ErrCode %1:%2.

ID: **12016** Name:fileRenamed
Argument:filename Type:string
Description: Current datalogger file renamed to %1.

ID: **12017** Name:createTmpLogfile
Argument:filename Type:string
Description: Creating temporary datalogger file %1.

ID: **12018** Name:dataloggerHardRestart
Description: Datalogger restarted.

ID: **12019** Name:dataloggerStartRequested
Description: Datalogger stop requested.

ID: **12020** Name:dataloggerStartRequested
Description: Datalogger start requested.

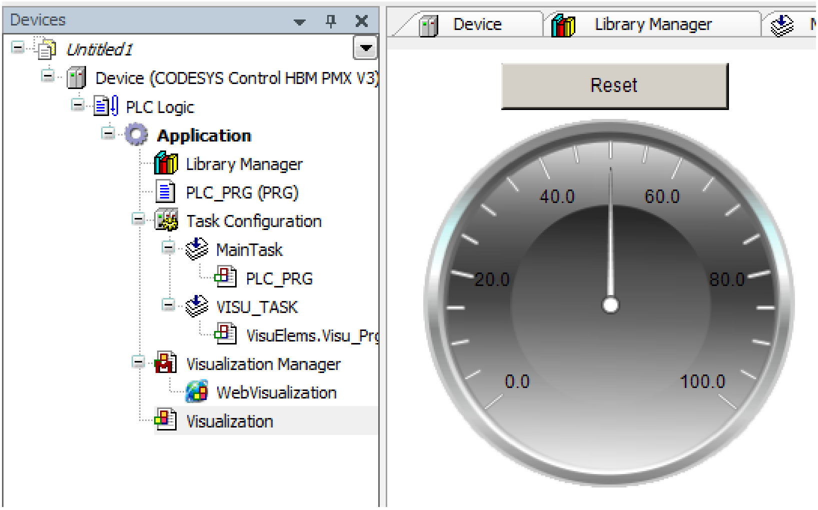
ID: **12021** Name:dataloggerDirectoryRemoved
Argument: filename Type:string
Description: Datalogger start requested %1.

18.11 Web visualization

CODESYS in PMX contains the WebVisu. This allows you to visualize and control the process via a freely configurable web page. The web server runs in CODESYS in the PMX.

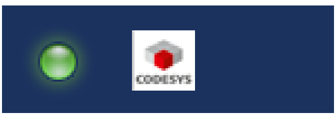
▶ Right-click on **Application** in the project tree. Then **Add object -> Visualization**.

You can now add graphical elements and link them to program variables. Example:



After starting the WebVisu in the PMX, the web page can be accessed from a web browser at **<pmx>:8080/webvisu.htm**. <pmx> is the device name here. You might need to replace it with the actual device name or an IP address. "webvisu.htm" is the CODESYS default name. It can be changed in the Visualization Manager.

There is a link to the WebVisu from the PMX user interface via the CODESYS icon in the footer. The default name "webvisu.htm" is required.



Tip

Examples of how to use WebVisu can be found in the TechNotes at https://www.hbm.com/de/2981/pmx-modular-measuring-amplifier-system-for-the-iot/.

18.12 CAN interface

The device can be operated with CODESYS as a CANopen slave or master. To do this, add a CAN component, and then a CANopen stack. You will find several examples in the package provided.

Tip

Examples of how to use CODESYS can be found in the TechNotes at https://www.hbm.com/de/2981/pmx-modular-measuring-amplifier-system-for-the-iot/.

18.13 CAN master and slave mode

Short description

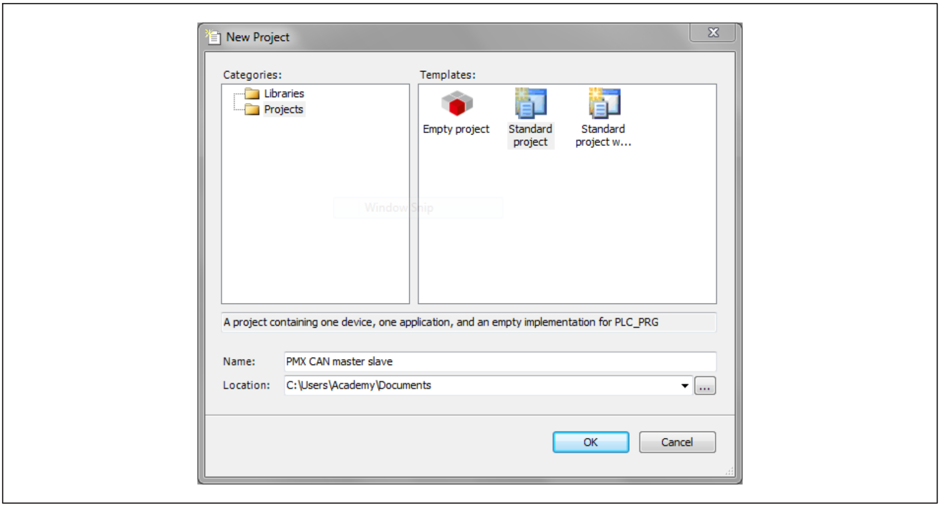
This is a guide to creating CODESYS applications on the PMX. Basic experience with CODESYS is required. Experienced users can feel free to take a different approach. Further help is available with the examples that are installed on the desktop by default when importing the PMX package, and from the online help on the package.

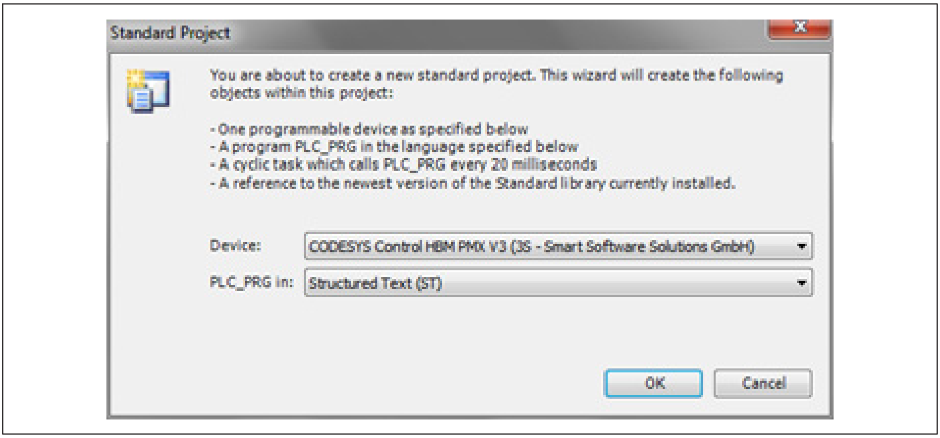
The required files are located on the "PMX CODESYS" CD, which is supplied with each PMX containing CODESYS or can be downloaded from the hbm.com Support page.

In this example, two PMX devices are interconnected via the CANopen interface. One PMX works as the master, the second as a slave in the network. A PDO with four measured values is generated in the second PMX, which then transmits the measured values to the first PMX (master) and displays them there on four calculation channels.

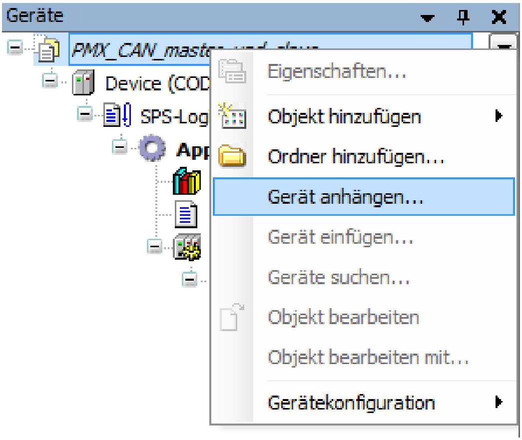
Start

- ▶ Call up the CODESYS environment. Create a standard project and select PMX as the device.

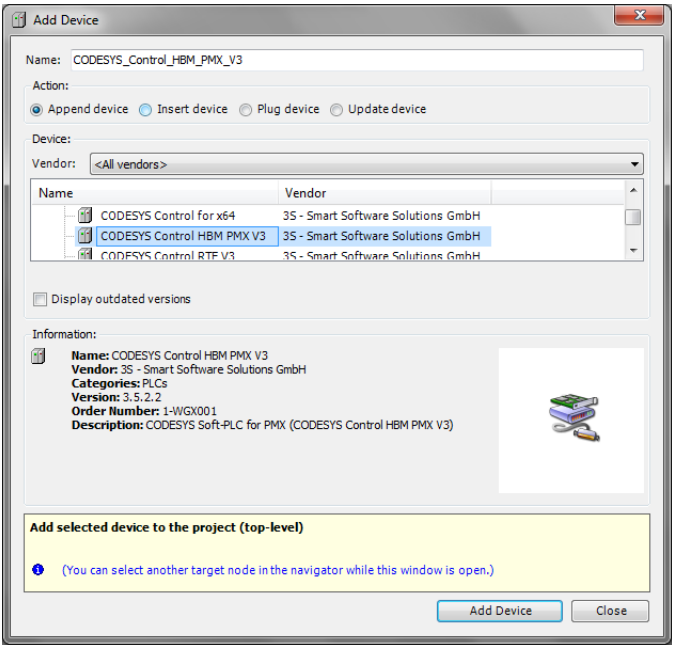




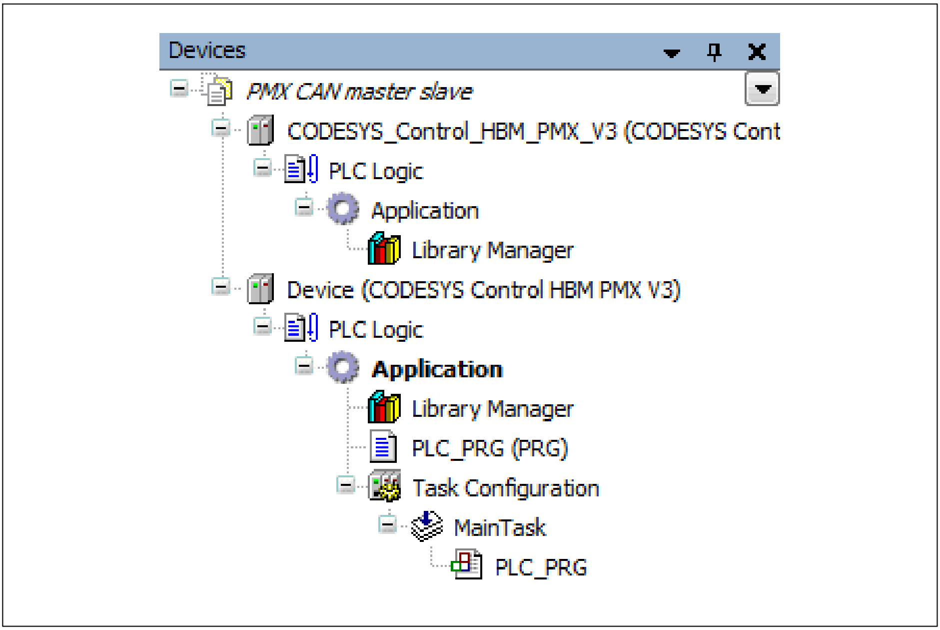
▶ In the open project device structure on the left, right-click on the file name and select **Append device…**



▶ ... and select another PMX.

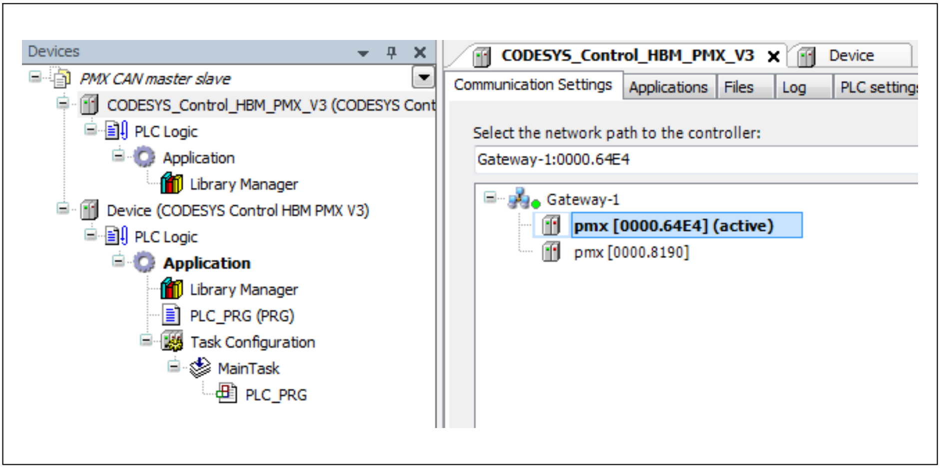


This results in the following structure with two PMX devices:

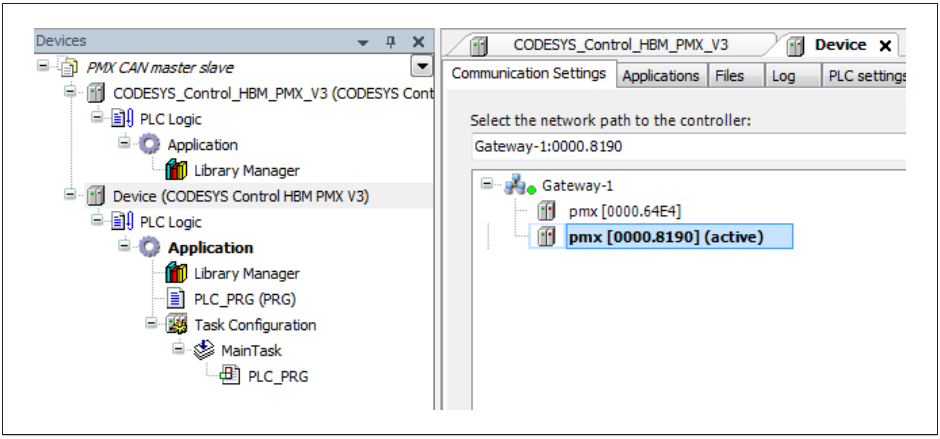


Activating the gateways

- Double-click on the first PMX (CODESYS_Control).
- Double-click on the gateway in the window that opens on the right to update the status.
- Then double-click on one of the PMX devices (here: [0000.64E4]) to activate it.

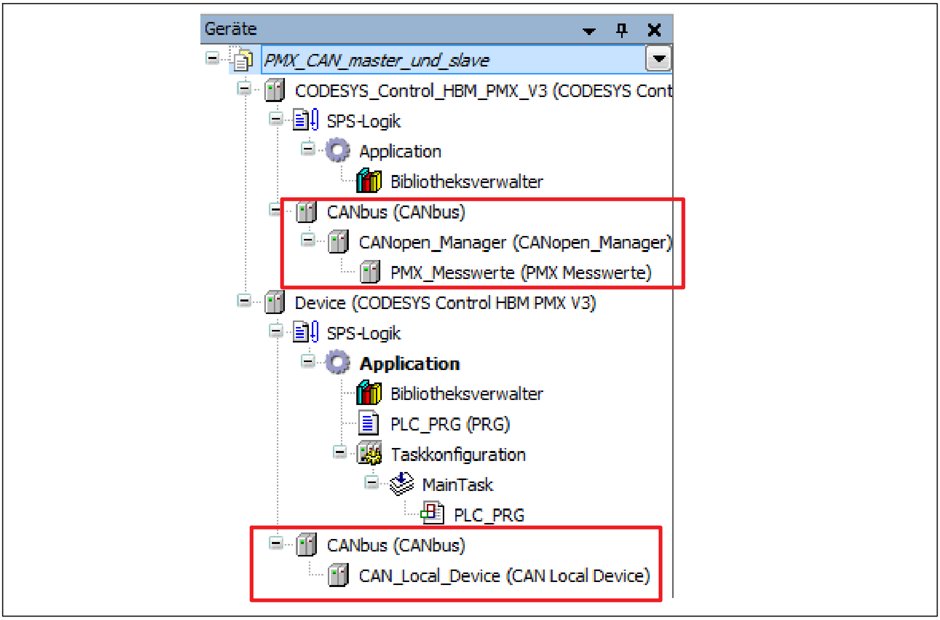


▶ Repeat the process for the second PMX (device) in the structure, and link it to the other PMX accordingly (here: [0000.8190]).



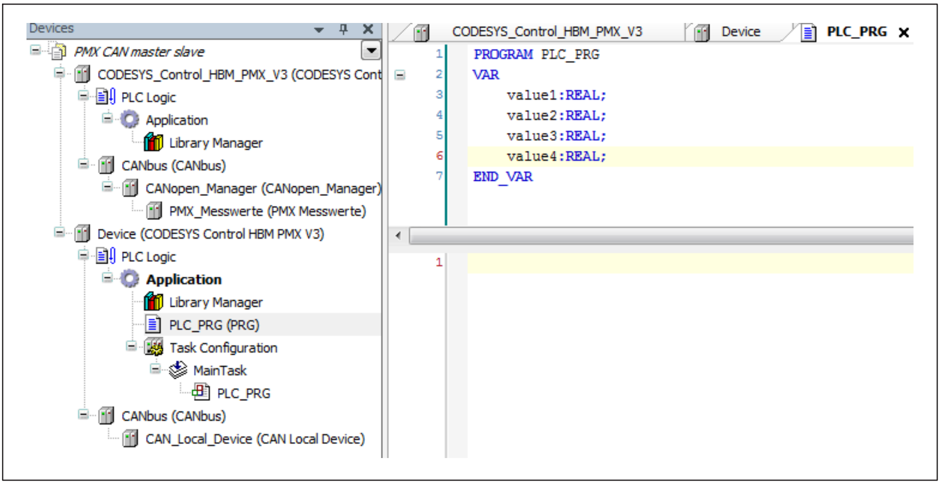
Appending CAN devices

- ▶ *For master mode*: For CODESYS_Control, add a CAN-Bus, a CANopen Manager and PMX_Measured values via the **Append device** selection option.
- ▶ *For slave mode*: For Device, append a CAN-Bus and a CAN_Local_Device in the same way.
- ▶ Set the baud rate for the CAN buses (here: 100000 bits/s).



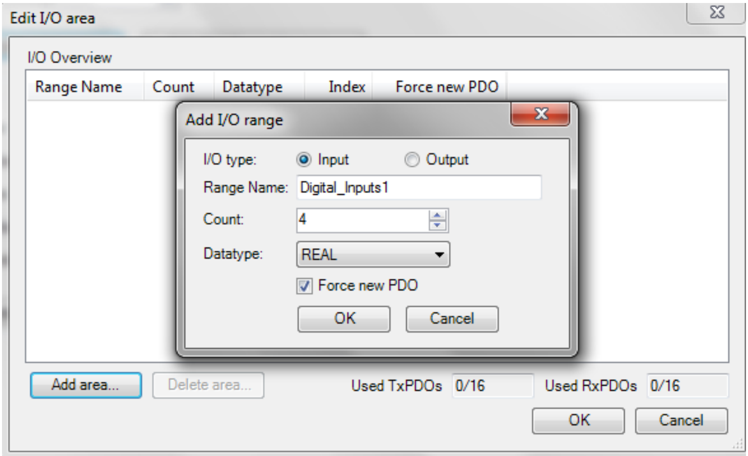
Declaring and mapping variables on the device

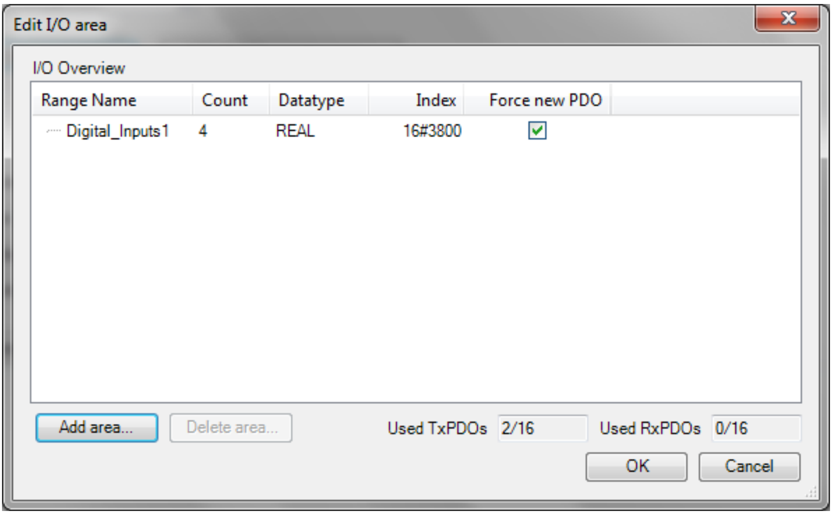
- Open the PLCPRG tab via Device, and declare variables as shown in the screenshot below.



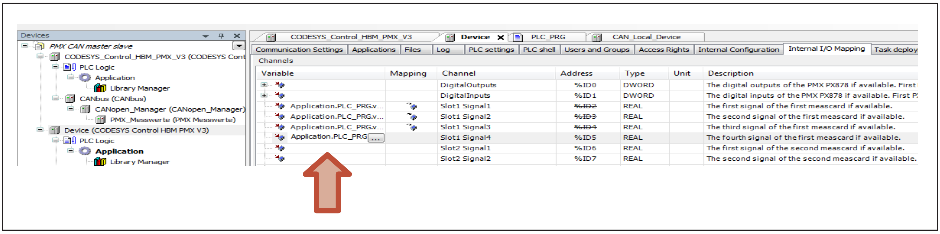
- Double-click on CAN_Local_Device.

- In the window that opens up, click on the **Edit I/O range** button.
- In the **Edit I/O range** window, click on **Add range**, and add a range as shown below.

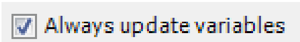




- Double-click on Device or Open tab. Select the **Internal I/O map** tab.
- In the **Variable** column, double-click in the cell to map a variable to a slot.

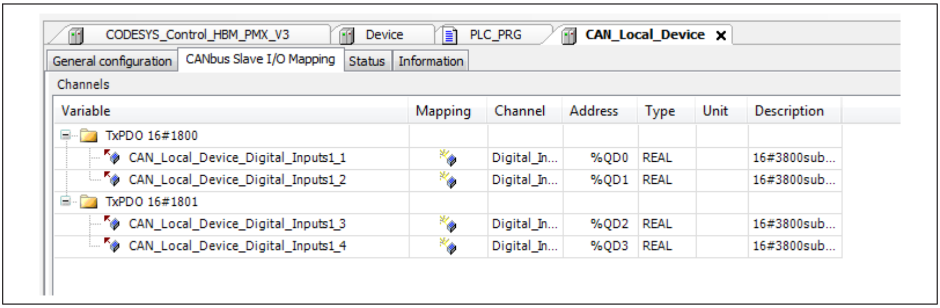


- Please note: Always check the box in the bottom right corner:

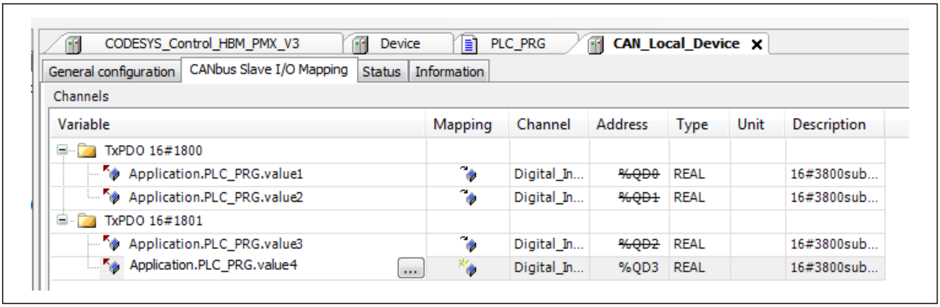


or

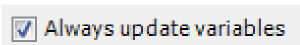
- Reopen the CAN_Local_Device and select the **CAN bus slave I/O map** tab ...



- ... map the variables here as well.



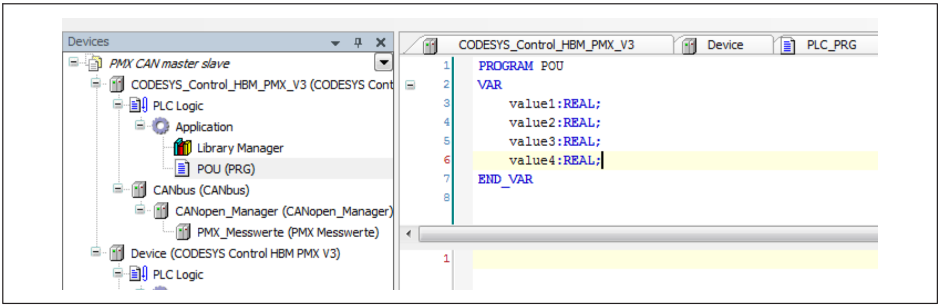
- Please note: Always check the box in the bottom right corner:



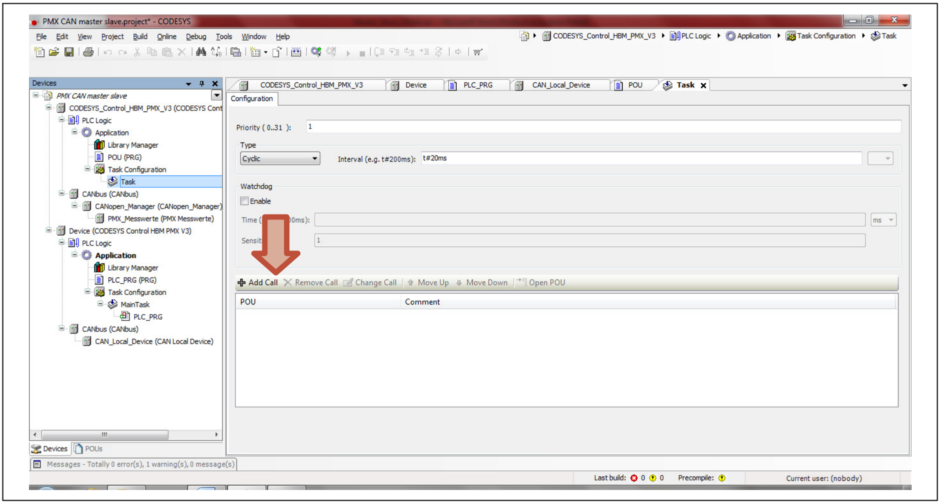
or

Declaring and mapping variables on the CODESYS_Control

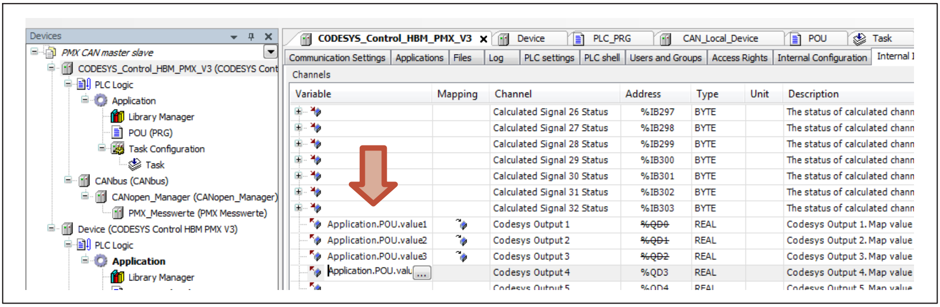
- Right-click on **Application** under the PMX CODESYS_Control, scroll to ***Add an object*** and select **POU**. Declare the variables in the same way here as on the device.



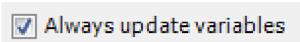
- Right-click on **Application** and choose **Add object**, and select a task configuration.



- In the window on the right, choose **Add call**, and select **POU**.
- Double-click on **CODESYS Control** or click on the tab.
- Select the **Internal I/O map**. In the **Variable** column, double-click to open the cell to map a variable to the corresponding CODESYS output channel.

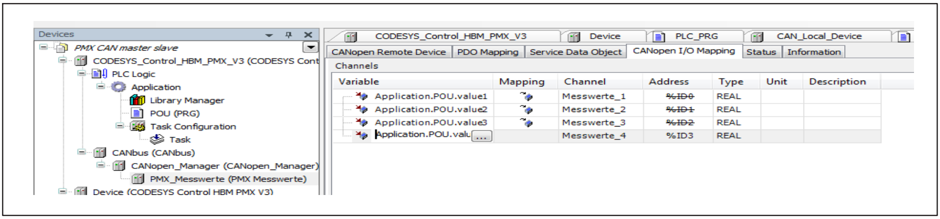


- Please note: Always check the box in the bottom right corner:



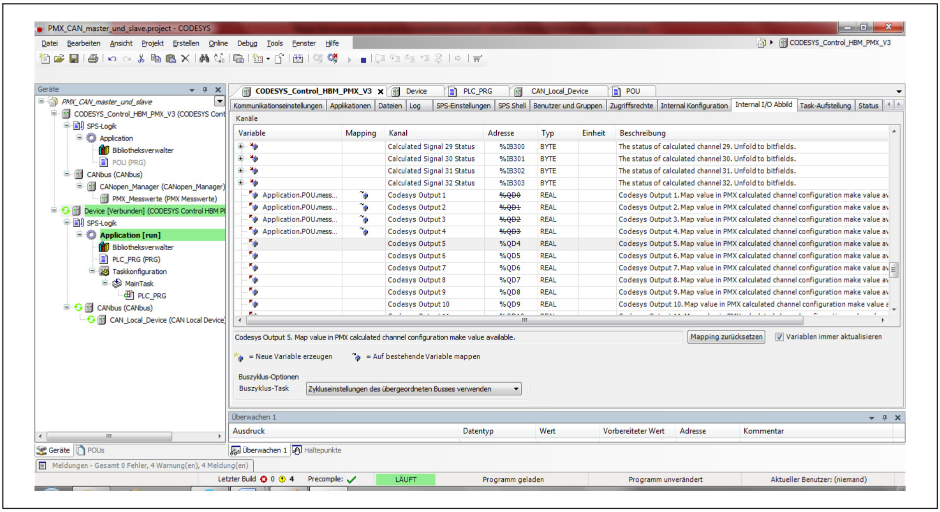
or

- Under **PMX_Measured values** map the variables.

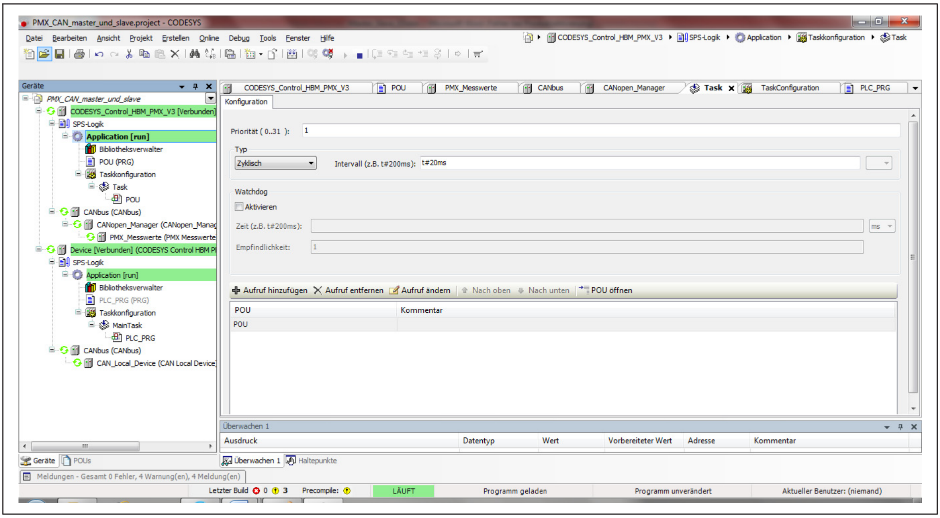


Running programs

- Click on **Device** and log in, then start the application with **F5**. The following screen is displayed.



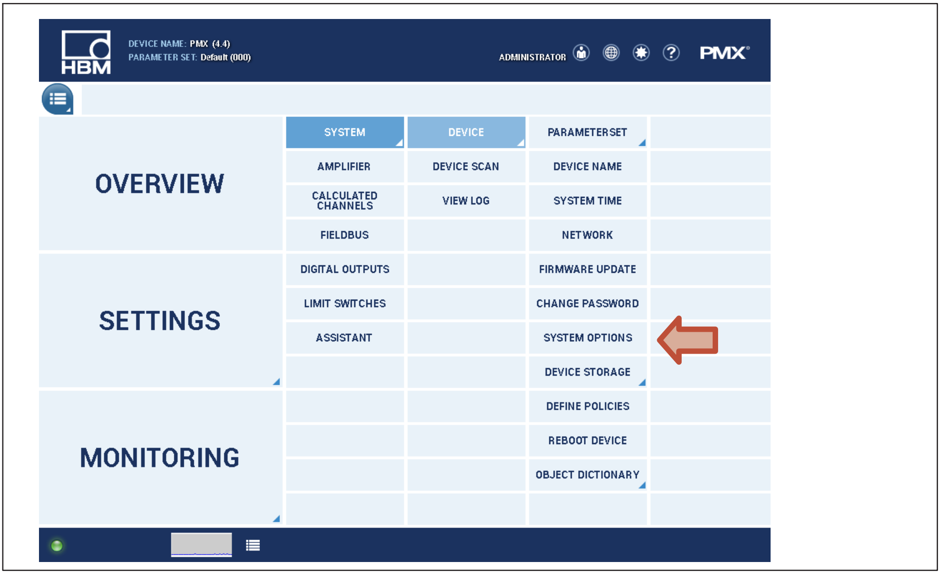
- Log out afterwards (Important: without stopping the application).
- Right-click on **Application** under CODESYS_Control and choose **Set active application**, and log back in.



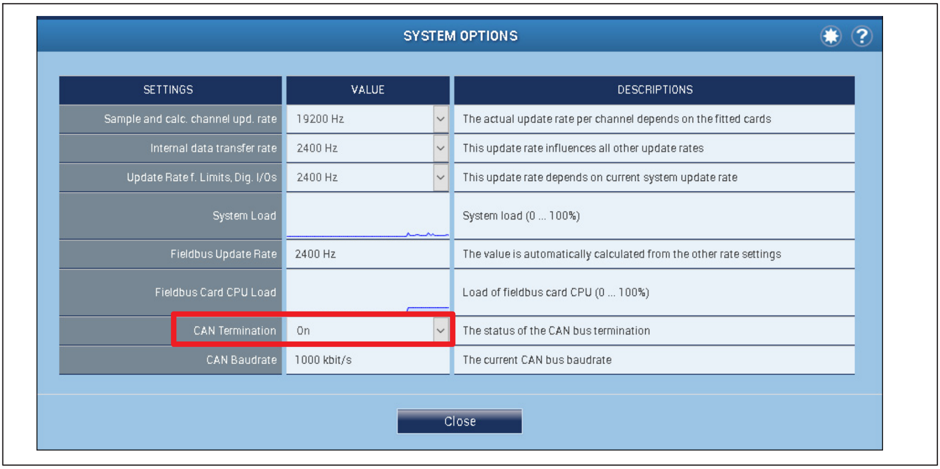
Configuring web browser

- Open the PMX in the browser. As two PMX devices are connected, the following overview appears. Copy the IP address of the second PMX. Open the two PMX devices in separate tabs.

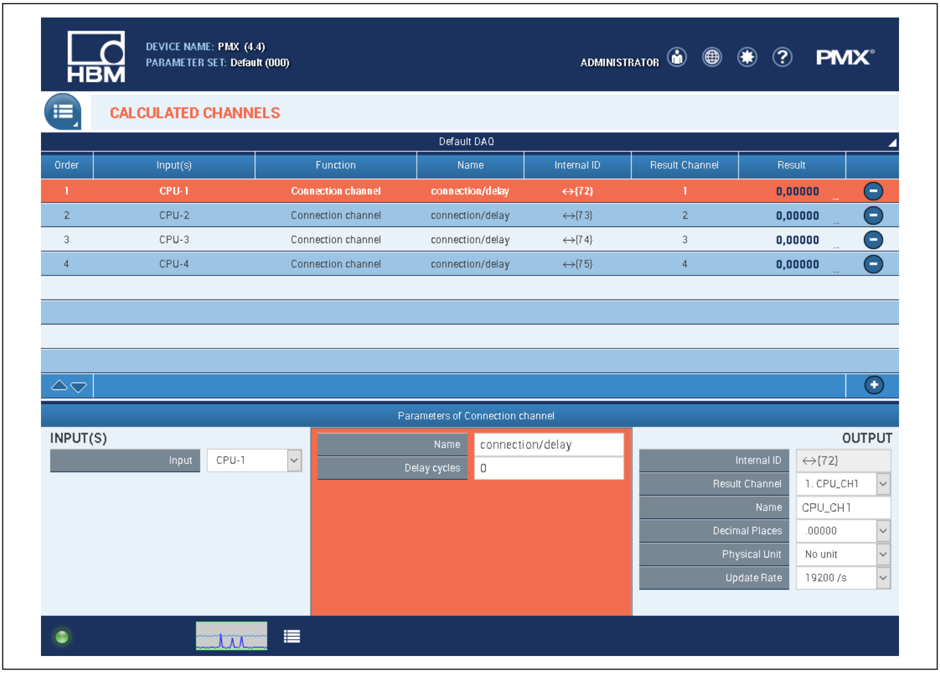
- Check the termination. To do this, in the Administrator in each device choose **Settings -> System -> Device -> System Options**.



- Check there that CAN termination is turned on for both devices.



- The values on the calculated channels can be displayed in the browser. To do this, choose **Connecting channel**, select the relevant CPU channel as the input, and define a calculated channel as the output. Select sufficient decimal places when doing this.



Tip

You can get more information and help on CODESYS programming from the CODESYS online help at
http://www.codesys.com/ or the CODESYS Chat at
http://forum-de.codesys.com/
Utilize the available knowledge and experience in the CODESYS Store. It offers lots of examples of programs and solutions for a wide variety of tasks:
http://store.codesys.com/?___store=default&___from_store=en

18.14 PMX package

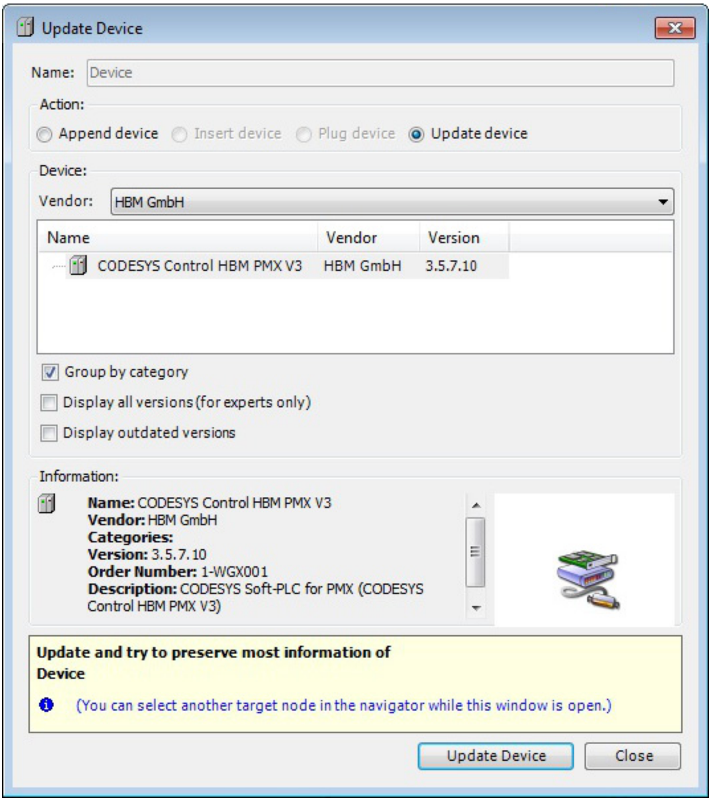
PMX package 0.94 includes some new functions for the PMX (see release notes). Proceed as follows when updating the PMX package from version 0.6 to 0.94:

1. Install the new PMX package. Use the CODESYS package manager to do this.
2. Update the (PMX) device. The PMX library, I/O Mapping and system events are updated here.

Information

Package version 0.94 requires PMX firmware 3.0. Run a firmware update as necessary. The current PMX firmware can be found at hbm.com: https://www.hbm.com/de/2981/pmx-modular-measuring-amplifier-system-for-the-iot/.
CODESYS applications created with an older PMX package (0.4 or 0.6) and PMX firmware < 3.0 might not run anymore, and their code might need to be updated.

Dialog: Updating the (PMX) device



19 DATA STORAGE

Measured values and data from the calculated channels and data present on the fieldbus or in CODESYS can be stored with the PMX in different ways. The amount of data, the storage speed, and the storage destination are key. The speed of data storage does not affect the measurement and data rates of the PMX.

DAQ (Data acquisition):

The Ethernet port of the PMX with a connected PC and DAQ software is suitable for storing large data volumes. The HBM catmanEASY/AP software is available for this, or you use custom software created using the PMX drivers in .NET, LabVIEW or DIAdem. You can store values from up to 20 PMX devices synchronously over an Ethernet network.

Monitoring

For stand-alone monitoring applications, measured values can also be stored in the PMX device memory (1 GB) or to a USB flash drive (max. size 32 GB) plugged into the PMX. Only values that have been acquired by the particular PMX can be saved. For this type of data storage, you need the basic device WGX001 with a free CODESYS program ("Measure and Save1.2.projectarchive") that is included in the collection of examples at https://www.hbm.com/de/2981/pmx-modular-measuring-amplifier-system-for-the-iot/.

The following table gives an overview of the possible signals and maximum storage rates.

Signals	catman	API/LabVIEW/ DIAdem	CODESYS
Medium	Store on PC or server		Store internally on PMX (1 GB) or USB flash drive (32 GB)
Measuring inputs (PX455, PX401, PX460)	x	x	x
Calculated channel	x	x	x
Digital inputs (PX878)	x	x	x
Digital outputs (PX878)	x	x	x
Analog outputs (PX878)	-	-	-
Signals from fieldbus (PROFINET®, EtherCAT®, EtherNet/IP™)	Max. 8 signals	Max. 8 signals	-

	read	write/store
Green	19.2 kHz	19.2 kHz
Yellow	2.4 kHz	10 Hz
Orange	250 Hz	19.2 kHz

Tip

Practical examples of data storage can be found in the PMX TechNotes at
https://www.hbm.com/de/2981/pmx-modular-measuring-amplifier-system-for-the-iot/

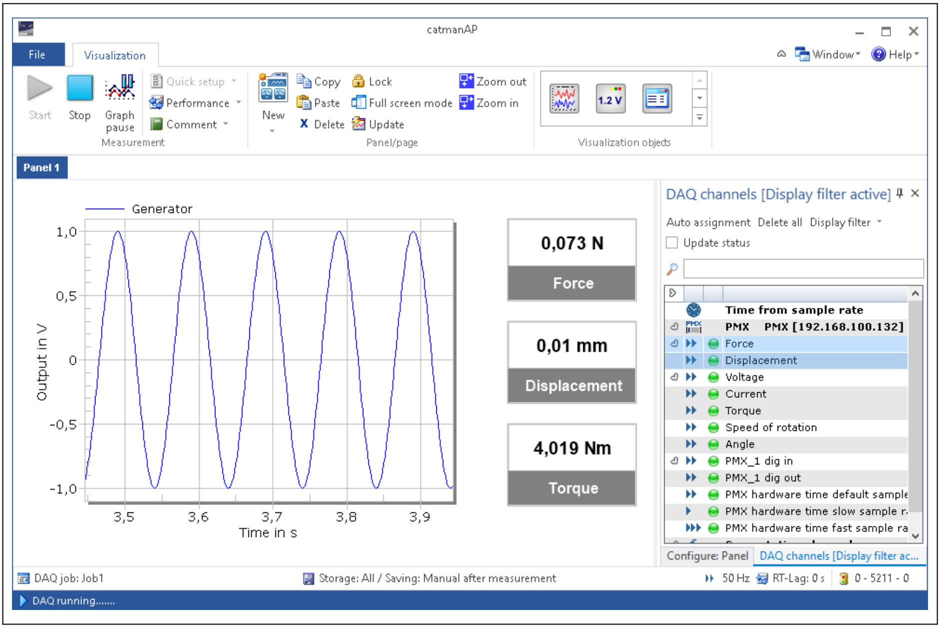
20 DATA ACQUISITION SOFTWARE (DAQ) CATMAN

You can use HBM's PC-based data acquisition software catmanEASY/AP for professional data acquisition and analysis. You have many professional data acquisition (DAQ) functions available that are also helpful for starting up and for acquiring quality data:

- Visualization, storage and analysis of PMX measurement data, internal calculation channels, digital inputs/outputs and fieldbus data with up to 38,400 measured values per second and channel.
- Simple PMX system and channel configuration (sensor database, TEDS editor, sample rate, filters, etc.).
- Measured value trigger by PMX digital inputs and outputs, external PLC (trigger including pre- and post-trigger, cyclic storage, long-term measurement, etc.).
- Powerful data analysis (signal-to-signal, zoom, magnifier, ruler, min/max, FFT, cutting to size, eliminating outliers, etc.).
- Automatically generate reports or display and export measurement data (MS Word, Excel, etc.).
- Create your own test and measurement routines with the catman Script language.

You can use up to 20 PMX simultaneously in one measurement with catman. Synchronization (<1 µs) runs via the Sync ports in the PMX (see *chapter 9, "Synchronization and time recording", page 127*).

If the PMX is to be used in a measurement together with other measuring devices (e.g. MGCplus or QuantumX), the devices are synchronized via NTP (±1 to ±10 ms).



Important

No signals may be added or cleared in the PMX while a catman measurement is running, otherwise the catman measurement will be aborted.

Tip

You can download a fully functional demo version of catman for free from https://www.hbm.com/index.php?id=1254&L=1.

21 PMX COMMAND SET

You can use the command set (API) to integrate the PMX into your own software applications, including Microsoft Windows and Linux-based software systems, such as VisualStudio, LabVIEW or Delphi. This allows you to implement custom solutions.

21.1 Requirements and notation

- The PMX TCPIP port is 55000.
- All the commands are summarized in the command list (*section 21.2, starting on page 348*).
- Virtual analog channels (calculated channels) use channel 9 (slot 9).
- Virtual digital channels (calculated channels, 1 = dig in, 2 = dig out) use channel 10 (slot 10)
  Due to the binary representation of a floating point value, only the 24 lowest bits are used. The 8 most significant bits are always zero.
- Character strings must be entered with a quotation mark at the beginning and end of the text. A quotation mark inside a character string is not permitted!

The following sections use the notations set out below for greater clarity.

- (x) Terminating character of a command:
  Line feed (LF) or
  carriage return/line feed (CRLF)
- (y) Closing sequence of a response:
  Carriage return/line feed (CRLF)
- Carriage return = decimal 13
  Line feed = decimal 10
- A positive response normally comprises a "0", followed by (y). A negative response is usually a "?", followed by (y).

Example: TELNET connection

The TELNET protocol under Windows offers a convenient way to use PMX commands.

The IP addresses of the PMX and the PC (HOST) must be compatible, and the nodes must be connected via Ethernet (assign a suitable IP address to the PMX as necessary, because DHCP is used in the factory setting).

PMX command list example in a Telnet session under Microsoft Windows

Identify the PMX IP address either by directly assigning the address, or in the **Network** dialog of the PMX web browser.

Open the command line input window:

- Under Windows 7: **Start -> All programs -> Accessories -> Run.**

Under Windows 7 it may be necessary to enable a Telnet client. Click on **Start -> Control Panel -> Programs and Features -> Turn Windows features on or off**. Scroll to **Telnet Client** and activate it.

Click ***OK*** and wait for the feature to be installed.

- In Windows 10, you usually have to install the Telnet client first: In the Windows search bar, type ***Turn Windows features on or off*** and click on the entry with the same name. Scroll to **Telnet Client** and activate it.

  Click ***OK*** and wait for the features to be installed.

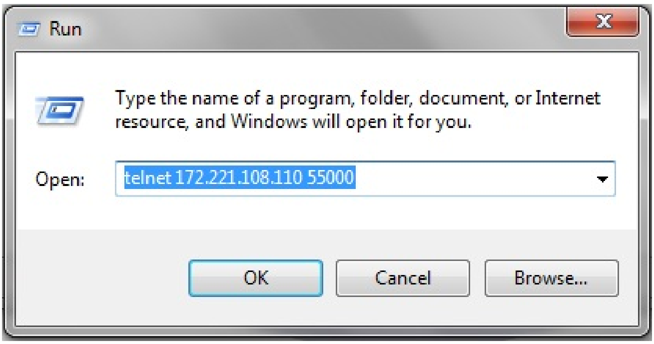
Starting a Telnet session and connecting to PMX

Type ***Run*** in the Windows search bar. Enter:

"Telnet <IP address of PMX> 55000"

Replace <IP address of PMX> with the IP address you are using.

Example:



Retrieve measured values:

PCS3,4(x) 'select channels 3 and 4

0(y)

SPS1(x) 'select subchannel 1 (of channels 3 and 4)

0(y)

RMV?214(x) 'Retrieve measured values.

9.998,8.888(y)

21.2 Command list

IDN?

Identification Query

Device identification output

Syntax: IDN?(x)

Parameters: None

Response: *String(y):* possibly more than 16 characters

Example: IDN?(x)
HBM,PMX,1234-5678,1.12, 6415M,*0.20,myPMX (y)*

Organization, device name, serial number, firmware version number, firmware build number, hardware version, host name

AMT?

Amplifier Type Query

Output amplifier type

Syntax: AMT?(x)

Parameters: None

Response: *q1(y)*

q1	Amplifier type
5125	PX401
5126	PX455
5128	PX460
5127	PX878, not for measurement
999 Subject to modificat-ions	PX999 for proprietary commands only
5130	Channel (slot) 9, calculated channels
5131	Channel (slot) 10, digital I/O

PCS

Programming Channel Select
Channel selection for the setting commands

This command is used to select channels for the subsequent setting commands.

Syntax:	PCS p1,.., pxx (x)
Parameter:	p1,.., pxx channels PCS 0 (x) selects all existing channels

Initially all the channels (cards) are selected.

If a channel (card or slot) is not available, this channel is ignored and not added to the list.

PCS(x) clears all the selected channels (cards/slots). The PCS?1 only returns (x).

Note: The response depends on the SRB command.

PCS?

Programming Channel Select Query
Output channel selection for the setting commands

Syntax:	PCS? p1(x)
Parameters:	p1:Output mode 0 Existing channels 1 Selected channels
Response:	q1,.., q16 List of existing or active channels

PCS?0 corresponds to PCS?

All the available subchannels (signals) are selected initially.

SPS

Subchannel Programming Select
Select channels for the setting commands

This command sets the subchannel selection mask for the settings. The modules to set should already have been selected with PCS.

Syntax:	SPS p1,.., pxx(x)
Parameters:	p1,.., pxx 1,.., xx subchannel selection SPS 0 (x) selects all the subchannels of a module
Syntax:	SPS? p1(x)

Returns 1,2,3:3,4:1,2,3,4,5 for example. Channels (slots/cards) are separated by ":".

Note: The response depends on the SRB command.

SPS?

Subchannel Programming Select Query
Output channel selection for the setting commands

Syntax: SPS? p1(x)

Parameters:	p1:Output mode 0 Existing subchannels 1 Selected subchannels
Response:	q1,.., qxx List of existing or active channels

UCC

User Channel Comment

Enter comment

Syntax:	UCCp1(x)
Parameter:	p1: Any string "_____", max. 45 characters
Note:	If you want to enter a channel name and comment, both of which will be stored in the UCC string, you are recommended to separate them by ";".
Example:	To store the channel name "ChannelName_1" and the comment Comment "ChannelComment_1" in the amplifier, send the command: UCC"ChannelName_1;ChannelComment_1"

All the selected subchannels (PCS and SPS) are given this name!

Note:	The response depends on the SRB command.

UCC?

User Channel Comment Query

Output comment

Syntax:	UCC?(x)
Parameters:	None
Response:	"*__(String)__"(y):* stored in a string with quotes at start and end
Note:	All the comments of all the selected subchannels of all the selected channels (PCS and SPS) are returned**! All names (and comments) are separated by ":", not by ","!!!.**

All the comments of all the selected subchannels of all the selected channels (PCS and SPS) are returned! *All names (and comments) are separated by ":", not by "," !!!*

EUN

Engineering Unit

Enter physical unit:

Syntax 1: EUNp1(x)

Parameter: p1: "UnitString"

Syntax 2: EUNp1(x)

Parameter: p1: Unit code

Note: The response depends on the SRB command.

EUN?

Engineering Unit Query

Output physical unit

Syntax 1: EUN?(x)

Parameters: None

Response: *q1(y):* "UnitString"

Syntax 2: EUN??(x)

Parameters: None

Response: *q1(y):* Unit code

Supported units

Code	Name	ASCII name
// angle (radian)		
100	"rad"	""
101	"radian"	""
102	"°"	"deg"
103	"%degrees"	""
// length		
300	"m"	""
301	"µm"	"um"
302	"mm"	""
303	"cm"	""
304	"dm"	""
305	"km"	""
306	"inch"	"in"
307	"feet"	""
308	"yard"	""

Code	Name	ASCII name
309	"mile"	""
// mass		
400	"kg"	""
401	"g"	""
402	"t"	""
403	"kt"	""
404	"ons"	""
405	"bs"	""
// time		
500	"s"	""
501	"ms"	""
502	"µs"	"us"
503	"min"	""
504	"h"	""
505	"days"	""
// current		
600	"A"	""
601	"A rms"	""
602	"mA"	""
603	"µA"	"uA"
604	"mA rms"	""
605	"µA rms"	"uA rms"
// temperature		
700	"K"	""
701	"ºC"	"degC"
702	"ºF"	"degF"
703	"ºRank"	"degRank"
704	"ºR"	"degR"
// voltage/sensitivity		
1000	"V/V"	""
1001	"mV/V"	""
1002	"µV/V"	"uV/V"

Code	Name	ASCII name
	// voltage	
1100	"V"	""
1101	"mV"	""
1102	"µV"	"uV"
1103	"V rms"	""
1104	"mV rms"	""
1105	"µV rms"	"uV rms"
	// resistance	
1200	"ohm"	""
1201	"kOhm"	""
1202	"MOhm"	""
1203	"mOhm"	""
	// inductance	
1300	"H"	""
1301	"mH"	""
1302	"µH"	"uH"
1303	"nH"	""
	// capacitance	
1400	"F"	""
1401	"mF"	""
1402	"µF"	"uF"
1403	"nF"	""
1404	"pF"	""
	// charge r m kg s A K mol cd	
1500	"C"	""
1501	"nC"	""
1502	"pC"	""
	// frequency	
1600	"Hz"	""

Code	Name	ASCII name
1601	"kHz"	""
1602	"MHz"	""
1603	"1/s"	""
1604	"mHz"	""
// rotational speed		
1700	"radian/s"	""
1701	"rev/min"	""
1702	"rpm"	""
1703	"1/min"	""
// power r m kg s A K mol cd		
1800	"W"	""
1801	"mW"	""
1802	"kW"	""
1803	"MW"	""
1804	"GW"	""
// force		
1900	"N"	""
1901	"kN"	""
1902	"MN"	""
1903	"kp"	""
1904	"kgf"	""
1905	"lb"	""
1906	"GN"	""
// pressure		
2000	"Pa"	""
2001	"bar"	""
2002	"mbar"	""
2003	"kbar"	""
2004	"pas"	""
2005	"hPa"	""
2006	"kPa"	""
2007	"psi"	""

Code	Name	ASCII name
2008	"N/mm^2"	"N/mm2"
2009	"N/m^2"	"N/m2"
2010	"N/cm^2"	"N/cm2"
// energy		
	// torque	
2101	"Nm"	""
2100	"J"	""
2102	"kNm"	""
2103	"MNm"	""
2104	"ftlb"	""
2105	"inlb"	""
2106	"GNm"	""
	// torsion	
2200	"Nm"	""
2201	"Nm/radian"	""
2202	"oz-in"	""
	// strain	
2300	"m/m"	""
2301	"µm/m"	"um/m"
2302	"strain"	""
2303	"mm/m"	""
	// speed	
2400	"m/s"	""
2401	"km/h"	""
2402	"mph"	""
2403	"fps"	""
2404	"m/h"	""
	// acceleration	
2500	"m/s^2"	"m/s2"
2501	"ga"	""

Code	Name	ASCII name
2502	"mm/s^2"	""
	// density	
2700	"kg/m^3"	"kg/m3"
2701	"g/l"	""
	// flow rate	
2800	"m^3/s"	"m3/s"
2801	"l/min"	"l/mn"
2802	"m^3/h"	"m3/h"
2803	"gpm"	""
2804	"cfm"	""
2805	"l/h"	""
2806	"l/s"	""
	// quotas	
2900	"%"	""
2901	"%o"	""
2902	"ppm"	""
	// temperature drift	
3000	"%/°C"	"%/degC"
3001	"%o/°C"	"%o/degC"
3002	"ppm/°C"	"ppm/degC"}
	// numerical values	
3100	"Imp"	""
3101	"kImp"	""
	// general physical units	
	// r m kg s A K mol cd	
5001	"%/decade"	""
5002	"dB"	""
5003	"l/l"	""

Code	Name	ASCII name
5004	"m^3/m^3"	"m3/m3"
5005	"m^3"	"m3"
5006	"mm^2"	"mm2"
5007	"kg/s"	""
5008	"mole/l"	""
5009	"mole/m^3"	"mole/m3"
5010	"N/m"	""
5011	"RH"	""
5012	"V/(m/s^2)"	"V(m/s2)"
5013	"V/C"	""
5014	"V/N"	""
5015	"V/Pa"	""
5016	"W/°C"	"W/degC"
100000	"UserDefined"	"usr"

ESR?

Read status register

Read default status register

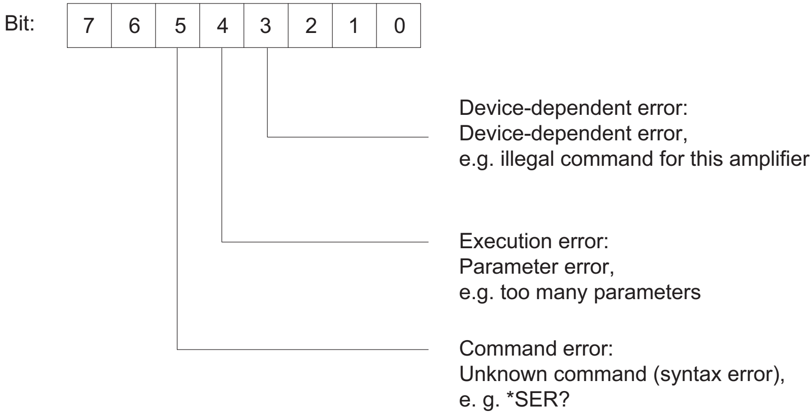
Output default event status register error status register

Syntax: ESR?p1 (x)

Parameters: None

Effect: The content of the default event status register (ESR) is outputted in its decimal equivalent. The default event status register (ESR) is set if there are communication errors. The different error causes set different bits, so that the errors can be accurately identified.

Response: *q1(y)*
q1: 8, 16 or 32 (or sum total)



None of the other bits are assigned.

Execution error: e.g.: command not valid for the selected channel (card).

ESR is deleted after reading.

Response: *p1: 1*
q1(y)
q1: Sum of the binary representation of individual status bits of status 1; see following table

Status 1	Binary value	Comments
FACTORYSETTINGS_ERROR	1	Factory calibration for PMX housing damaged (not measurement cards!)
SYNC_MASTER	2	Bit set: Bit of the synchronization master cleared: Synchronization slave
SYNCMESSAGE_ERR	4	Missing or invalid synchronization messages
SYNC_UNLOCKEDSLAVE_ERR	8	Stable connection not possible, controller cannot synchronize with master
ALIVE	16	Switch takes place at approx. 1 Hz
POWEROVERLOAD	32	Power supply is overloaded at unspecified location
CAT_BUF_OVERRUN	64	Overrun of "Catman" interface buffer (TSV..) ==> Measured value current interrupted

Status 1	Binary value	Comments
SYSTEM_NOT_READY	128	e.g. change of parameter set in progress or failed
DSP_OVERRUN	256	e.g. too many calculated channels

MCS

Measuring Channel Select

Choose channel selection for channels to be recorded

The channels to be recorded are selected with this command. MCS cannot be used for selection during data acquisition. In this case, the command is acknowledged with a "?". The query command is also possible during recording.

Syntax: MCS p1,.., pxx (x)

Parameters: p1,.., pxx 1,.., xx Channel selection
MCS 0 (x) selects all the existing channels
MCS (x) cancels selection of all channels

Note: Channels 17, 18 and 19 are the internal time stamps. The time stamp is a 6-byte counter with a rate of approx. 153640 Hz. Time channels 17, 18 and 19 do not appear in commands PCS?/SPS?. Times have no subchannel representations (command SMS).
The binary ("measurement") data has a length of 8 bytes, with the two most significant bytes being zero.
Time data is always placed at the end of a measured value line. There is an implicit connection between the measurement rate group and the three time channels:
Channel 17: Measurement rate group 0
Channel 18: Measurement rate group 1
Channel 19: Measurement rate group 2

Note: The response depends on the SRB command. Also take a look at the STF and NTP commands. The multi-I/O card PX878 cannot be selected here. It does not generate any measured values.

MCS?

Measuring Channel Select Query

Output channel selection for channels to be recorded

Syntax: MCS?p1(x)

Parameters: p1:Output mode:
0 Existing channels, e.g. 1,2,3,4(,17,18,19)
1 Active channels

Response: q1,.., q19 List of existing or active channels

SMS

Subchannel Measurement Select

Choose subchannel mask for recording

This command sets the subchannel selection mask for recording. The channels to set (= PMX slots) should already have been selected with PCS.

Syntax: SMS p1,.., pxx (x)

Parameter: p1,.., pxx 1,.., subchannel selection
SMS 0 (x) selects all existing subchannels

Note: The response depends on the SRB command.

SMS?

Subchannel Measurement Select Query

Query the subchannel mask for recording

Syntax: SMS?p1(x)

Parameters: p1:Output mode
0 Existing subchannels
1 Selected subchannels

Response: q1,.., qxx List of existing or active channels

MSS

Subchannel Measurement Select

Choose measurement signal selection for channels to be recorded

This command chooses the signals to be recorded of the channels selected with PCS/SPS. Various signal combinations can be selected for the different channels. Above all, it is possible to record more than one signal for each subchannel.

Syntax: MSS p1, p2, p3, p4 (x)

p2…p4 are optional.

If the call has no parameters, the selected subchannels are set to gross.

Parameters:

p1...p4	Signal to be recorded
214	Gross, dynamic
204	Min. value, virtual channel
205	Max. value, virtual channel
217	Max. - Min. (peak-to-peak), virtual channel

Note: This command can drastically increase the amount of data to be calculated and transferred. Not everything is possible.
As the PMX has an internal multi-client software architecture and catman® is "only" one of those clients, the available sig-

nals (apart from gross) must be generated at the upper level. Otherwise these signals will not be available.

Note: The response depends on the SRB command.

MSS?

Measuring Signal Select Query

Output measurement signal selection for channels to be recorded

Syntax: MSS?(x)

Parameters: None

Response: list[i]: list[j]:...: list [k]
list [x]

Example: 214,204:214,205:217….

MRG

Measurement Rate Group

Measurement signal selection for channels to be recorded

This signal assigns the measurement rate group to a selected channel or subchannel (PCS/SPS). Up to 3 synchronous measurement rate groups are supported. The measured values of the different groups are stored in separate FIFO buffers, and must be read out separately over the interface.

Syntax: MRG p1,p2,p3 (x)

Parameter: p1: 0..2 Measurement rate group
p2: 0..2 Measurement rate group (optional)
p3: 0..2 Measurement rate group (optional)

Note: The response depends on the SRB command.

MRG?

Measurement Rate Group Query

Output measurement signal selection for channels to be recorded

Syntax: MRG? (x)

Response: q1(y)

q1: Measurement rate group

Example: MrgOfSubSignal11: MrgOfSubSignal12: MrgOfSubSignal21: MrgOfSubSignal22 …

NTP

Network Time Parameter

Sets the IP address on the NTP server in dot notation or its name (as a string).

Syntax: NTP p1 (x)

Parameter: p1: IP address or name of the NTP server: e.g. 172.19.178.12 or "172.19.178.12" or "ntp.devel.hbm.com"

NTP?

Network Time Parameter Query

Sets the IP address on the NTP server in dot notation or its name (as a string).

Syntax: NTP? (x)

Parameters: p1: is optional
p1 missing or 0: Call NTP server
p1 =1; accuracy information as a character string

Response: for p1= 0 (or p1 missing p1):
q1 (,q2,q3..)(y)
q1 ...: NTP servers used as character strings, separated by ","

Note: This query responds to the NTP server(s) currently in use. This may be one or more different servers than the one previously defined with the NTP command. This case can occur when a DHCP server defines another NTP server.

ICR

Internal Channel Recordingrate

Internal channel sample rate

This command is used to set (only) one data rate per group. A second data rate in one group is not supported.

Syntax: ICR p1, p2(x)

Parameters: p1:Data rate 1, see rate list below
p2: 0, 1, 2 ; Measurement rate group

If parameter p2 is omitted, the command affects measurement rate group 0.

Status	Value	Comments
1 Hz	6300	
2	6301	
5	6302	
10	6303	
20	6326	
25	6304	
50	6305	

Status	Value	Comments
75	6307	
100	6308	
150	6309	
200	6310	
300	6311	
600	6313	
1200	6315	
2400	6317	standard
4800	6319	
9600	6320	
19200	6345	
38400	6346	

Note: The response depends on the SRB command.

ICR?

Internal Channel Recordingrate Query

Output internal channel sample rate

Syntax: ICR? p1(x)

Parameter: p1: Measurement rate group 0, 1, 2

Response: q1 (y) sample rate

Note: If parameter p1 is omitted, the command affects measurement rate group 0.

TSV

Transient Setup Values

This command defines and starts data acquisition.

Syntax: TSVp1 (x)

Parameter: p1: 0, 1,..,N Number of value lines to be measured in a single measurement:

1…N -> Max. FIFO size 15 MB per measurement rate group, 0 means infinite. -> Standard FIFO size 5MB per measurement rate group

-1 means infinite in FIFO size of 1 line.

This allows the user to be get *ONE* line (RMB?1,…) with the *latest* measurement data, without constantly starting a new measurement. Not yet fully tested as to whether the values of the various subchannels can be recorded at exactly the same time.

Note: The response depends on the SRB command. Deletes "Overrun" status bit, see TSV? query

TSV?

Transient Setup Values Query

This command defines and starts data acquisition.

Syntax: TSV? p1(x)

Parameter: p1: Measurement rate group; 0, 1, 2

Response: q1, q2, q3 (y)

q1: Number of measurement lines in FIFO buffer that were not sent.

q2: Trigger status of current measurement
2: Measurement active (waiting for end)
3: Measurement ended (default)

q3: Status bits
Bit 0, (value = 1): FIFO buffer overrun, will be cleared by next TSV command
Bit 1, (value=2): not yet used
Bit 2, (value=4): not yet used
Bit 3, (value=8): not yet used
Bit 4, (value=16): not yet used

STP

Stop

End measurement output and data acquisition

Syntax: STP(x)

Parameters: None

Response: None

Note: The response depends on the SRB command.

OMP

Output Measuring Pointer

Measured value buffer output pointer

This command is used to position the read pointer in system memory (FIFO memory in which the measured values are recorded). The user must know what is to be done. There is no error management!

Syntax: OMP p1, p2 (x)

Parameter: p1: -N,.,N, offset for moving the read pointer: -(max- FIFO lines -1)…max. FIFO lines -1 max. FIFO lines as from TSV command

P2: 0,..,2 Measurement rate group 3 asynchronous FIFO

If the measurement rate group (p2) is not specified, measurement rate group 0 is affected.

Note: The response depends on the SRB command.

OMP?

Output Measuring Pointer Query

Query measured value buffer output pointer

Syntax: OMP? p1(x)

Parameter: p1: Measurement rate group 0,..,2

Response: q1, q2 (y)
q1: Available (readable) lines, current read pointer to current write pointer
q2: Data logging status
0 Data acquisition is not executed
1 Data acquisition is executed

If the measurement rate group is not specified, measurement rate group 0 is affected.

MBF

Measuring Buffer Format

Output format

This command defines the RMB output format. The query command returns the format currently set.

Syntax: MBFp1,p2(x)

Parameters: p1:
1257 4 bytes binary (float) INTEL (physical quantity), no other formats are supported

In floating point formats, an error (overflow/calibration error) is encoded by 2e20.

p2: Measurement rate group 0,..,2

If parameter p2 is omitted, the setting affects all measurement rate groups.

Note: The response depends on the SRB command.

MBF?

Measuring Buffer Format Query

Output the output format

Syntax: MBF? p1(x)

Parameter: p1: Measurement rate group 0,..,2

Response: q1(y)
q1: Output format

If parameter p1 is omitted, you are given the output format of measurement rate group 0.

RMB?

Read Measuring Buffer Query

This command is used to output the measured values recorded in system memory.

The character string "#0" (2 bytes) is placed before the measured values for output (only in the first line); this can then be followed by as many values as are available, or as have been requested. Each value is 4 bytes in size, and the value format is "Float".

If more measured values were requested than are currently available, the output routine remains on standby, until more measured values come in. CR LF is outputted once, as the terminating character after the last line. The output format is defined with the aid of the MBF command. As this is always followed by an output, even if less measured values than requested are present, you should always use the OMP? command before using the RMB? command, to discover how many lines of measured values are present.

Syntax: RMB? p1,p2,p3(x)

Parameter: p1: Number of lines of measured values to output

p2: Output mode
6400 from the start of the entire measurement memory (transmission of read pointer), not supported
6406 from the current read pointer, not supported
6407 from the current read pointer; release of all previous, not supported
6408 read the latest values (read pointer is unchanged), not supported
6409 from the current read pointer; move read pointer forward by p1 lines

p3: Measurement rate group 0,..,2

If the measurement rate group is not specified, measurement rate group 0 is affected.

RMV?

Read Current Measurement Value

Output measurement data.

Syntax: RMV? p1 (x)

Parameter: p1 signal

p1	Signal
214	Gross
204	Min
205	Max
217	Peak-to-peak

Effect: When possible, the RMV? command outputs the required signal of the channels selected with PCS and SPS. Not every channel type supports every signal type. If a channel is selected that does not support the signal type conveyed in parameter p1, 2.0e20 is displayed.

The min, max and peak-to-peak signals must be "defined/created" at the upper level (see "MSS" command) if they are to be displayed!

Otherwise 2.0e20 is returned.

If p1 is omitted, gross values are returned.

Example: PCS3,4(x) 'Select channels 3 and 4
0(y)

SPS1(x) 'Select subchannel 1 (of channels 3 and 4)
0(y)

RMV?214(x) 'Retrieve measured values.
9.998,8.888(y)

Virtual subchannels (slot 9/channel 9) do not support the subsignals min, max and peak-to-peak.

Digital values (slot 10) and analog output values (PX878) do not have any min/max/peak-to-peak values themselves. Analog output values (PX878) are updated about every 0.5 seconds (2 Hz). Therefore if this command is called more often, the same value will be generated. If the signal source of the analog output has an invalid status, 2e20 is generated (regardless of the "invalid signal value" entered via the web GUI). The voltage levels for PX878 correspond to the calculated levels. They are NOT (re)measured at the outputs!

SFC

Signal Filtering Characteristic

Cut-off frequency and filter characteristics

Defines the cut-off frequency and filter characteristics for all channels/subchannels selected with PCS and SPS.

Syntax: SFCp1,p2(x)

Parameters:

p1	Filter characteristics as per table 1
p2	Cut-off frequency as per table 2

Filter characteristics	Value	Comments
No filter	140	Virtual slot 9 only
Butterworth	141	6th order filter
Bessel	142	6th order filter

Tab. 21.1 Filter characteristics

The table below shows the available cut-off frequencies with Bessel or Butterworth characteristics, dependent on the measurement card.

p1=141 / 142	Cut-off frequency (Hz)			
914	0.1	X	X	X
917	0.2	X	X	X
921	0.5	X	X	X
927	1	X	X	X
931	2	X	X	X
935	5	X	X	X
941	10	X	X	X
945	20	X	X	X
949	50	X	X	X
955	100	X	X	X
958	200	X	X	X
962	500	X	X	X
969	1000	X	X	X
973	2000	X	X	X
976	3000	X	-	X
978	5000	-	-	X[1)]
979	6000	-	-	X[1)]
1150	100000[2)]	X	-	X

Tab. 21.2 Filter frequencies

1) These cut-off frequencies are only available if the parameter "Channel update rate for sampling and calculation" (browser user interface: Settings -> System -> Device -> System Options) is set to 38.4k.

2) This value indicates that the digital filter is working with "neutral" coefficients, and that only the analog anti-aliasing filter is active. The physical cut-off frequency may depend on the card.

Virtual subchannels (slot 9/channel 9), digital subchannels (slot 10/channel 10) and PX878 multi-IO card do not support filters. Parameters p1, p2 (and p3) can be defined, but this is ignored!

Note: The response depends on the SRB command.

SFC?

Signal Filtering Characteristic query

Output cut-off frequency and filter characteristics

Syntax 1: SFC?(x)

Parameters: None

Response: q1,q2(y)
q1 Filter characteristics
q2 Cut-off frequency

e.g. 142,969:142, 969

Syntax 2: SFC??(x)

Parameters: None

Response: *q1,q2(y):* Possible filter characteristics
e. g. 141,142:141,142

Syntax 3: SFC?142,?(x)

Response: *q1,...,qn(y):* Available Bessel frequencies
e. g.
914,917,921,927,931,935,941,945,949,955,958,962,969,973,1150:914,917…

Virtual subchannels (slot 9/channel 9), digital subchannels (slot 10/channel 10) and PX878 multi-IO card do not support filters. Query returns 140, 1150

CAP

Calibration Point

Enter the characteristic points of the transducer (input)

Affects all the selected channels (PCS/SPS)

Syntax: CAPp1,p2,p3(x)

Parameter: p1: Number of points (1 or 2)

p2: Measurement signal (amplifier-dependent unit), if there is no input value present, the current measured value is applied

p3: Display value

Effect: The input characteristic is defined by two points. The input signal and the associated display value must be entered for each point. This command also defines the scaling of the PX878. That includes the physical values (p2, from the signal source) and electrical values (p3, output in volts) of the voltage output.

Virtual subchannels (channel 9) do not support calibration points. It is possible to define parameters p1, p2 and p3, but this is ignored!

Note: The response depends on the SRB command.

CAP?

Calibration Point Query

Output the characteristic points of the transducer (input)

Output the input characteristic points (transducer) of all the selected channels (PCS/SPS).

Syntax: CAP?<p1>(x)

Parameter: p1:Number of points (1 or 2)

Response: *q1,q2,q3(y)*
q1: Number of points (1 or 2)
q2: Measurement signal (amplifier-dependent unit)
q3: Value in the displayed units

Virtual subchannels (channel 9) do not support calibration points.

Response for p1=1: q2=0, q3=0.
Response for p1=2: q2=100, q3=100.

CAL

Calibration

Calibrate the amplifier

Calibrate the amplifier, all selected channels (PCS/SPS). Activates the Cal option implicitly. The ACL setting does NOT change! Only supported for PX455! Other (measuring) cards generate "OK" (0).

Syntax: CAL(x)

Parameters: None

Note: In all CF bridge amplifiers, this command triggers a calibration. Measured values "flicker" for several seconds.

The response depends on the SRB command.

CAL?

Status of Calibration procedure

Output the status of the calibration procedure

The status of the calibration procedure, all selected channels (PCS/SPS).
Only supported for PX455! Other (measuring) cards generate "OK" (0).

Syntax: CAL?(x)

Parameters: None

Response: *q1(y)*

0	Autocalibration is not executed
1	Autocalibration is executed

e.g. two cards with 4 channels each: 0,0,0,0,:1,0,1,1

ACL

Enable / Disable Autocal

Switching autocalibration on and off

Activate (default) or deactivate the automatic start of calibration of all the selected channels (PCS/SPS). Calibration is then executed if the sensor is blocked, or the measurement signal overflows for a few seconds. Only supported for PX455! Other (measuring) cards generate "OK" (0).

Syntax: ACLp1(x)

Parameters:

p1	Autocalibration
0	Off
1	On

Note: When switched on (activated), automatic calibration is activated for bridges or bridge-like sensors (potentiometers/ LVDT). An ACL 0 command does not abort a running calibration sequence. It only suppresses any others from starting.

The response depends on the SRB command.

ACL?

Enable / Disable Autocal Query

Output of autocalibration status

All selected channels (PCS/SPS). Only supported for PX455! Other (measuring) cards generate "OK" (0).

Syntax: ACL?(x)

Response: *q1(y):*

q1	Autocalibration
0	Deactivated
1	Activated

e.g. 0,0,0,0:1,1,1,1:0,0

AIS

Amplifier Input Signal

Select the amplifier input signal

Select amplifier input signal. Supported for PX455 only. The command is ignored for other (measuring) cards. They generate "OK" (0).

Syntax: AISp1(x)

Parameters:

p1	Input signal	Supported
40	Internal zero signal	PX455
41	Internal calibration signal	PX455
42	Measurement signal	All measurement cards, virtual and digital channels
43	Reference point, not supported	---
46	Measurement signal without excitation point, not supported	---

AIS?

Amplifier Input Signal Query

Output the amplifier input signal

Requirement of the amplifier input signal. Supported for PX455 only. Other (measurement) cards generate 42.

Syntax: AIS?(x)

Parameters: None

Response: *q1(y): Input signal*

CPV

Clear Peak Value

Clear peak-value memory

Virtual subchannels (slot 9/channel 9), digital subchannels (slot 10/channel 10) and PX878 multi-IO card do not support peak values.

Affects all the selected channels (PCS/SPS)

Clear peak-value memory

Syntax: CPVp1(x)

Parameters:

p1	Clears
none	Peak-value memory 1, Max
1	Peak-value memory 1, Max
2	Peak-value memory 2, Min
3	Peak-value memory 3, peak-to-peak

Note: When cleared, the peak-value memories (Min or Max) are set to the current measured value. Peak-to-peak is set to 0,0.
Peak-to-peak has its own min/max memory!
The peak value signals must be parameterized on the upper level beforehand. Otherwise, they are not available.
Virtual subchannels (channel 9) do not support peak values.

Note: The response depends on the SRB command.

HPV

Hold Peak Value

Peak-value memory update status

Affects all selected channels (PCS / SPS). Deactivate/activate peak-value memory storage.

Interrupt/activate peak-value memory update

Syntax: HPVp1,p2(x)

Parameter: p1: Peak-value memory 1 (Max), 2 (Min) or 3 (Peak-to-peak)

p2=1: Interrupt update

p2=0: Activate update (default)

Every time you switch on, the status is set to "Activate update".

Virtual subchannels (slot 9/channel 9), digital subchannels (slot 10/channel 10) and PX878 multi-IO card do not support peak values.

Note: The response depends on the SRB command.

HPV?

Hold Peak Value Query

Read out the peak-value memory update status of all selected channels (PCS/SPS)

Syntax1: HPV?p1(x)

Parameter: p1: Peak-value memory 1 (Max), 2 (Min) or 3 (Peak-to-peak)

Response: *q1,q2(y):*

q1: Requested peak-value memory
q2: 1: Update interrupted

0: Update activated: e.g. for Max (p1=1) 1,1⁚1,0⸬1,1

⸬ means that this subchannel (between the two ⸬) has no max, min or peak-to-peak values

Syntax 2: HPV??(x)

(y): Available peak memory: e.g. 1,2,3⁚1,2⸬1⸬ means that this subchannel (between the two ⸬) has no peak values

The command returns the status of the peak-value memory, which can be set by the HPV command.

Virtual subchannels (slot 9/channel 9), digital subchannels (slot 10/channel 10) and PX878 multi-IO card do not support peak values.

SAD

Sensor Adaption

Select transducer adaptation for all selected channels (PCS/SPS)

Syntax: SAD p1,p2,p3(x)

Parameters:

p1	Bridge excitation voltage (or current), see *Tab. 21.3 and Tab. 21.6* for PX460
p2	Transducer type, see *Tab. 21.4* and *Tab. 21.7* for PX460
p3	Sensitivity (optional), see *Tab. 21.5* (not usable for PX460)

Status	Value	Comments	Command
No excitation	10	PX401	
1 V	11	Not supported	
1.25 V	12	Not supported	
2.5 V	13	PX455	
5 V	14	Not supported	

Tab. 21.3 Bridge excitation voltage (p1)

Status	Value	Comments
Full bridge	350	PX455
Half bridge	351	PX455
Quarter bridge	352	
SG full bridge	353	
SG half bridge	354	
SG quarter bridge	355	
Inductive full bridge	356	PX455 (= FB 100 mV/V)
Inductive half bridge	357	PX455 (= HB 100 mV/V)
LOW level full bridge	358	
LOW level half bridge	359	
HIGH level full bridge	360	
HIGH level half bridge	361	
SG full bridge, 120 ohm	362	
SG full bridge, 350 ohm	363	

Status	Value	Comments
SG full bridge, 700 ohm	364	
SG half bridge, 120 ohm	365	
SG half bridge, 350 ohm	366	
SG half bridge, 700 ohm	367	
LVDT	380	PX455 (= HB 1000 mV/V)
Potentiometer	385	PX455 (= HB 1000 mV/V)
75 mV DC	425	
10 V DC	426	PX401
DC 20 mA	427	PX401
60 V DC	433	
DC 4 .. 20 mA	435	PX401
Charge 0.1 nC	571	
Charge 1 nC	572	
Charge 10 nC	573	
Charge 100 nC	574	
Virtual sensor	575	PMX

Tab. 21.4 Transducer type (p2)

Status	Value	Comments
4 mV/V	778	PX455
100 mV/V	774	PX455
1000 mV/V	776	PX455

Tab. 21.5 Transducer sensitivity (p3)

SAD parameters for PX460

p1:

Value	Input type
23	Direct (digital connection, differential or single-pole)
24	Indirect (only for frequency measurement)

Tab. 21.6 Input type for PX460

p2:

Status	Transducer type
520	Frequency (direct or inductive)
525	Pulse counter (direct only)
580	SSI (direct only)
527	PWM (direct only)

Tab. 21.7 Transducer types for PX460

Virtual subchannels (channel 9) use values p1=10, p2=575. This is ignored if other values are set (response OK).

Note: The response depends on the SRB command.

SAD Parameters for PX878

P2:

Status	Value	Comments
± 10 V	290	PX878
-20 ... 20 mA	291	not supported
4 ... 20 mA	291	not supported

Tab. 21.8 Output type for PX878 (p2)

Virtual subchannels (slot 9/channel 9) and digital subchannels (slot 10/channel 10) use values p1=10, p2=575. This is ignored if other values are set (response OK).

PX878 uses values p1=10, p2=290. The setting of other values is ignored and "?" is generated as an error.

Note: The response depends on the SRB command.

SAD?

Sensor Adaption Query

Output the set transducer adaptation for all selected channels (PCS/SPS)

Syntax 1: SAD?(x)

Parameters: None

Response: *q1,q2(y)*

q1	Excitation voltage (or current), see *SAD command tables Tab. 21.3 to Tab. 21.6*
q2	Transducer type see *SAD command table Tab. 21.7*
p3	Sensitivity (1 if not supported/needed), see *Tab. 21.4*

e.g. PX401: 10,426,-1:10,427,-1:….

Syntax 2: SAD??(x)

Parameters: None

Response: *q1,..,qn(y):* Possible bridge excitation voltage or similar according to table 1 to 4 (SAD command)
e.g. PX401: 10,10,10:10,10,10:0

Syntax 3: SAD?,?(x)

Response: *q1..qn(y):* Possible transducer type as per *Tab. 21.4* (SAD command)
e.g. PX401: 426,427,435: 426,427,435:…

Virtual subchannels (channel 9) return the values q1=10, q2=575.

SAF

Set Additional Function, only available for PX460

Parameters for counter

Only the 2nd and 4th subchannels can be configured (SPS2 / SPS4). The 1st and 3rd subchannels are permanently assigned for frequency measurement with direct (digital) input.

The 1st and 2nd subchannels use the same settings for glitch filter, digital input type, and termination. The 3rd and 4th subchannels are linked in the same way.

Syntax: SAF p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11(x)

p2…p11 are optional

Parameters:

p1	Glitch filter	0: Off / 0,082, 1, 10, 100: On	Removes pulse widths < (p1) µs
p2	Digital input type	0: Differential 1: Single-pole	Differential or single-pole digital input (negative input set to "medium" voltage), default value is 0
p3	Termination	0: None	Electrical termination for operation with differential input

		Termination 1: Termination resistors active	Default value is 0
p4	Sensor type is Counter	525	Defines meaning and number of the following parameters
p5	F1+F2	0: Off / 1 (F2=90deg), 2 (F2=dir): On	Signal F1+F2 is evaluated
p6	Resolution quadrupling	0: Off / 1: On	Evaluates only one or all edges
p7	Physical zero index input	0: Off / 1 On	Activates hardware (zero) input. When ON, the counter result is set to p10 at (each) zero index pulse.
p8	Factor	0: Off (deactivated) / 1,2,3,4: On	Activates automatic reset of the counter after p8 revolutions (p8 * p9 pulses)
p9	Pulses per revolution	0: Off / 1...16000: On	When On, the counter is set to zero after a count result of p9 is reached. A physical zero pulse (p7) can also reset the counter result. These options are usually combined with each other.
p10	Offset in pulses	0...16000	Reset value of the counter, normally not greater than p9
p11	Reverse direction	0: Default 1: Reversed	Reverses the counting direction
p12	Interpolation	0: OFF 1: ON	Works similarly to a filter. Useful for signals with slow pulses.

Note: This command automatically resets the current counter value to zero (NOT p10!).
Only p1 is needed to reset the counter without changing the current settings (parameterization was done before).

Explanations:

Glitch filter, p1

Input signals with pulse widths < x µs are not evaluated. Default value is 1 (1 µs).

F2 evaluation, p5

Activates detection and evaluation of the direction of rotation using the second hardware input. Default value is 0 (Off).

Resolution quadrupling, p6

All adjacent edges of F1 and F2 are evaluated. If F2 is not connected or F2 evaluation is off (p6=), the resolution is doubled. Default value is (Off).

Zero index input, p7

Hardware input for the zero index signal. Relevant for incremental transducers. In count mode, an active zero index signal (electrical HIGH level) sets the counter value to p10. Default value is 0.

Factor, p8

Factor for automatic reset of counter value to p10 after p8*p9 pulses (p8 revolutions). Default value is 0.

Pulses per revolution, p9

Used for rotary encoders. Pulse count for a single revolution. If resolution quadrupling is enabled, this value must be multiplied by 2 or 4. Default value is 0.

The "physical zero index" provides the zero position. The "Pulses per revolution" setting knows the maximum value for the transition.

Zero->maximum. The "pulses per revolution" in combination with the "physical zero index" can also compensate for a difference of a quarter pulse width between these two competing possibilities for resetting the counter value.

Angle offset in pulses, p10

Value the counter receives when zero index is active. Default value is 0.

Reversal of counting direction, p11

With this parameter it is possible to change the counting direction. Default value is 0.

Inversion of interpolation, p12

Works similarly to a filter. Useful for signals with slow pulses for smoothing the measured value.
Default value is 0.

Configuration examples

	Physical zero index input, p7	Factor, p8	Pulses per revolution, p9	
Linear configuration 1	0	0	0	Count up (and down if p5<>0), no reset to zero or p10
Linear configuration 2	1	0	0	Count up (and down if p5<>0), reset to zero or p10 if physical zero index is active
Rotation configuration 1	0	0	1...16000	Count up (and down if p5<>0), no reset to zero or p10, no wrap-around
Rotation configuration 2	0	1...4	1...16000 / p8	Reset counter every p8 revolutions, no use of "physical zero index", but automatic wrap-around after p8 revolutions
Rotation configuration 3	1	1...4	1...16000 / p8 *	Reset counter every p8 revolutions, use of "physical zero index" and "pulses per revolution" (p9); counter reset is synchronized with F1/F2 pulses
Rotation configuration 4	1	0	1...16000, but value is ignored	Hard reset with "physical zero index" to offset, "pulses per revolution" (p9) is NOT used, no error detection; better to use rotation configuration 3

*Subject to modifications.

Parameters for SSI transducers

Syntax: SAF p1,p2,p3,p4,p5,p6,p7(x)
p2…p7 are optional

Parameters:

p1	Glitch filter	0: Off / 0,082, 1, 10, 100: On	Removes pulse widths < (p1) µs
p2	Digital input type	0: Differential 1: Single-pole	Differential or single-pole digital input (negative input set to "medium" voltage), default value is 0
p3	Termination	0: No termination 1: Termination resistors active	Electrical termination for operation with differential input, default value is 0
p4	Sensor type is SSI	580	Defines meaning and number of the following parameters
p5	Coding	0, 1	0: Gray code (default), 1: Binary
p6	Bit length	6...32	Bit length of the transducer (resolution), normally 12,13,24,25 (default)
p7	Baud rate	1...5	Baud rate for transducer clock generator: 1:10 kBit, 2:100 kBit (default), 3:200 kBit, 4:500 kBit, 5:1000 kBit

SAF?

Set Additional Function Query, only available for PX460

Syntax: SAF? (x)

Response: *q1,q2,q3,q4,q5,q6,q7,q8,q9,q10,q11*(y)
q4…q11 are dependent on the currently activated sensor

q1	Glitch filter	0: Off / 0,082, 1, 10, 100: On	All sensor types
q2	Digital input type	0: Differential 1: Single-pole	All sensor types
q3	Termination	0: No termination 1: Termination resistors active	All sensor types

q4	Sensor type		520 Frequency (direct or inductive) 525 Pulse counter 580 SSI 527 PWM
q5	F1+F2 or coding		525 Pulse counter or 580 SSI
q6	Resolution quadrupling or bit length		525 Pulse counter or 580 SSI
q7	Physical zero index input or baud rate		525 Pulse counter or 580 SSI
q8	Factor		525 Pulse counter
q9	Pulses per revolution		525 Pulse counter
q10	Offset in pulses		525 Pulse counter
q11	Reverse direction		525 Pulse counter
q12	Use interpolation		525 Pulse counter

SCL

Shunt Calibration Output (PX460 only)

Only the shunt output of the 2nd and the 4th subchannel can be configured (SPS2 / SPS4, PX460 has only 2 shunt outputs).

Set shunt output to On/Off

Syntax: SCL p1(x)

Parameters:

p1	Set shunt output
0	Off
1	On

SCL?

Shunt Calibration Output Query (PX460 only)

Syntax: SCL?(x)

Parameters: None

Response: *q1(y):* Currently set status of the shunt output

TDD

Transmit Device Data

Activate different amplifier settings (parameter sets)

Syntax: TDD p1,p2(x)

p2 is optional

Parameters:

p1	Parameter set
-2	Saves all current parameters and parameter sets of the device, and defines the currently active parameter set as the boot parameter set. This command can take a long time (e.g. 10...60 s).
-1	Loads factory setting into currently active parameter set and activates it. p2 is not used. Command returns immediately. If a parameter set is set to the factory setting (p1 = -1), this also resets all assigned sub-parameter sets to their factory settings. Note that these sub-parameter sets could also be used in other parameter sets!!
0,1,2,...	Loads parameter set p1 and activates it. If an error occurs when changing the parameter set, "OK" could still be outputted as the response. With "TDD?" the parameter set currently in use can be checked.
p2	**Response behavior / Timeout**
Not present or 0	Does not wait, no timeout detection, command responds immediately.
>0.1,...	Timeout in seconds. Waits until switching or saving of the parameter set is completed (p1=-2) or a timeout has occurred. If the parameter set is set to default values (factory setting, p1 = -1), the timeout option p2 is not available!

Note: The parameter set system of the PMX consists of sub-parameter sets which are linked to a main parameter set that can be activated here (p1 >= 0).

TDD?

Transmit Device Data Query

Query where the amplifier setup comes from

Syntax: TDD? p1(x)

p1 is optional

Parameters:

p1	Command
Not present or 0	Gets active parameter set
1	Gets status bit SYSTEM_NOT_READY , -> if preset parameter switch is in progress.

Response: *q1(y):* in case of p1

p1	Query parameter
Not present or 0	Currently active parameter set
1	0: System ready, parameter switch completed; 1: System still running

CDT

Calibration Dead Load Target

Zero offset target value

Enter the target value for the zero point offset of the input characteristic (for command CDV) for all selected channels (PCS/SPS).

Syntax: CDTp1(x)

Parameters: p1:The target value should be set to the current measured value

Effect: The value in displayed units to which the amplifier should be set by the CDV command (no parameters). The factory setting is 0.

Note: The response depends on the SRB command.

CDT?

Calibration Dead Load Target Query

Output the zero point offset target value

Output the target value for the zero point offset of the input characteristic (for command CDV) for all selected channels (PCS/SPS).

Syntax: CDT?(x)

Parameters: None

Response: *q1(y):*Target value to which the current measured value is set e.g. 0.01,0,0.5,0.502

Virtual subchannels (channel 9) do not support "dead load targets". q1=0.

CDV

Calibration Dead Load Value

Zero point offset

Enter zero point offset of the input characteristics (transducer) for all the selected channels (PCS/SPS).

Syntax: CDVp1 (x)

Parameters: p1:Zero point value (offset) in displayed units

No parameters: The current measured value is set to the target value entered with the CDT command: (default: 0,0)
So the current measured value is needed. If the status of one of the selected subchannels is not valid, a ? is returned!

Effect: Additional zero point value (offset) which offsets the entire characteristic curve.

Explanation: The displayed measured value = gross (real measured value without offset) — p1

Virtual subchannels (slot 9/channel 9), digital subchannels (slot 10/channel 10) and PX878 multi-IO card do not support dead load values. The command is ignored with the response OK.

Note: The response depends on the SRB command.

CDV?

Calibration Dead Load Value Query

Output the zero point offset

Output the zero point offset of the input characteristics for all the selected channels (PCS/SPS).

Syntax: CDV?(x)

Parameters: None

Response: *q1(y):* Current zero point value in displayed units
e. g. 0.01**,**0**,**10.5**,**10.502

Virtual subchannels (slot 9/channel 9), digital subchannels (slot 10/channel 10) and PX878 multi-IO card do not support dead load values. q1=0..

ATB

Application To Bus

Application to bus

Writes a 64-bit integer value that can be read by the fieldbus master.

Syntax: ATBp1 (x)

Parameter: p1: 64-bit integer value

The format of p1 can be a decimal value, e.g. 87612398745, or a hexadecimal value, e.g. "0xaa12bb34cc56dd78", which must be entered as a string with the prefix "0x".

Note: The response depends on the SRB command.

ATB?

Application To Bus Query

Application to bus query

Outputs the current 64-bit integer value as a hexadecimal number previously written with the ATB command.

Syntax: ATB?(x)

Parameters: None

Response: *q1(y):* The current value written with the ATB command as a hexadecimal number
e. g. 0xab12

BTA?

Bus To Application Query

Bus to application query

Reads a 64-bit integer value that can be written by the fieldbus master.

Syntax: BTA?(x)

Parameters: None

Response: *q1(y):* The current value written by the fieldbus master as a hexadecimal number
e. g. 0xab12

STF

Set Time Format

Set time format

Defines the content and format of the time channels (MCS 17,18,19)

Syntax: STFp1 (x)

Parameters: p1 = 0:Factory setting, ticks (incrementing counter) as a 64-bit integer value

p1 = 1: System time as two 32-bit integer values, nanoseconds (the first 4 bytes) and seconds (the second 4 bytes)
p1 = 2:System time as two 32-bit integer values, microseconds and seconds
p1 = 3:System time as two 32-bit integer values, 2^32 seconds and seconds

The sum of the seconds and their fractions is equal to the time that has elapsed since 01.01.1970.

The system time can be derived from the NTP time. The accuracy is not 100% predictable.

Note: The response depends on the SRB command.

STF?

Set Time Format Query

Query set time format

Reads the time format currently in use

Syntax: STF?(x)

Parameters: None

Response: *q1(y):* Current setting for the time format

BLK

Activates or deactivates the LED signal functions of the entire device, or the signal function of the selected (sub-)channels (PCS / SPS)

Syntax: BLK p1,p2,p3(x)

	Selection p1 = 0: whole device p1 =1: subchannel
	Blink mode see tables below for both selections of p1
	P1 = 0: time of LED signaling in seconds (1…60) p1 = 1: ignored, no timeout possible

Parameters: none

Response: *q1(y):* current setting of the timeformat

TED

Transducer electronic datasheet

TEDS data sheet

Syntax: TED p1,p2,p3(x)

Parameters:

p1	p2	P3	Effect
0	-	-	Reads in the TEDS data *from the transducer to the amplifier*. If the data is damaged, or if TEDS is not available, the response is q1 = "?". In this case, no TEDS data is transmitted to the amplifier (length = 0) If more than one TEDS transducer identification is read out (PCS/SPS), the error response is also outputted if only one TEDS transducer identification has a read error. TEDS data is arranged in 32-byte pages. The first byte is the checksum, the subsequent 31 bytes are data bytes. The data are read and checked until the first invalid page is found, or all the pages have been read. Valid data pages are stored in the amplifier. The checksum bytes are removed. The command is executed synchronously. This means that the response is outputted when the TEDS transducer identification readout is concluded. Virtual subchannels (channel 9) do not support TEDS transducer identification (not physically present). The readout is ignored, and OK is returned.
1	Data length	Data (ASCII hex string)	Transmits and writes data to the transducer memory. p2 = data length: Number of bytes. p3: Data in ASCIIHex format. e.g. "AB75e2...". If p2 (data length) is equal to 0, and p3 is an empty string, (but has to be outputted), the data written to the transducer is taken from the amplifier memory. This is of course only possible if it has previously been read out by "Ted 0" without error. Internally, only entire pages with 32 bytes are written to the transducer (1 checksum byte + 31 data bytes). For example: If the user wants to write 32 data bytes to the transducer, two pages are written. The second page contains the checksum byte, a data byte from the user, and 30 filler bytes (zero values). The checksum is calculated internally and added. Virtual subchannels (channel 9) do not support TEDS transducer identification (not physically present). The write is ignored, and OK is returned.

p1	p2	P3	Effect
100	-	-	Reads and interprets the TEDS data. If the data is damaged, or settings for the amplifier are not possible, the response is q1 = "0" but EST?1 returns the code 15023: "TEDS ERROR" or the code 20031: "TEDS WARNING". TED?100 and TED?101 provide you with more detailed information about these errors and warnings. In the case of competing templates, the settings of the last template are accepted. *Not yet supported!*
101	-	-	Clears any error bit set for the TEDS measured value. The error bit might be set if a TED100 command finds a valid TEDS content but the device configuration failed. This might result in an invalid or only partially performed configuration, and so possibly produce invalid measured values. Virtual subchannels (channel 9) and the PX878 do not support TEDS (not physically present). Command is ignored, OK is outputted.

TED?

Transducer electronic datasheet Query

Output TEDS

Syntax: TED? p1(x)

Parameters:

P1	Effect
0	Reads out the TEDS header (8 byte binary) on the TEDS transducer q1: Binary with "#" and block length (16-bit binary). There is no CR/LF at the end of the binary TEDS data. If more than one subchannel is selected (PCS/SPS), the data is separated by a "**;**". Virtual subchannels (channel 9) do not support TEDS transducer identification (not physically present).
1	Reads the TEDS data *from* the amplifier *memory*. q1: Binary with "#" and block length (16-bit binary). There is no CR/LF at the end of the binary TEDS data. The block length depends on the TEDS chip (one-wire). e.g. 512 bytes. If more than one subchannel is selected (PCS/SPS), the data is separated by a "**;**". The minimum number of bytes should be 31 (1 checksum byte is deducted from the 32-byte page).

P1	Effect
100	Gets the warning/error status of the TEDS template handling (caused by TED100). **Not like** q1 = "0": OK q1 = "?": Error or no information available.
101	Gets error bit of TEDS measured value. q1 = "0": OK q1 = "?": Error bit set
102	Gets status of TEDS setting. q1 = "0": Parameters defined in TEDS were changed later. q1 = "1": All parameters defined in the TEDS are set in the amplifier; **not yet supported.**

TED?100

Response: q1,q2,q3

q1:0: No error,
otherwise template ID with error

q2: Error bit position in template

q3: Error type:17000 Template and amplifier not compatible
17002 Value above limit
17003 Value below limit
17004 Value outside limits
TED?102

Response q1: Checks the status of the TEDS setting, does not read TEDS transducer identification from the transducer

q1 = 0 Amplifier setting is *not* current

q1 = 1 All parameters defined by TEDS are set in the amplifier

TID?

Transducer Identification Query

Read chip identification

Syntax: TID?p1(x)

Parameters:

p1	Effect
1	Reads the 8 ident bytes of the TEDS chip

Response:

Response	Meaning
?	No TEDS chip available
e.g. "0A0000008A3D4C23"	Chip identification as hexadecimal string

Note: Virtual subchannels (channel 9) and the PX878 do not support TEDS. The query generates an error for each selected subchannel.

SRB

Select Response Behavior

Select the response behavior of the current interface

Syntax: SRB p1(x)

Parameters:

p1	Switch response output on/off
0	Switch response output off
1	Switch response output on

Effect: There are two types of command:
a.) Query commands (e.g. RMV?) are characterized by a question mark, and generate output data regardless of the response behavior selected for the interface. It is not possible to prevent this data being outputted with this type of command.

b.) The setting commands (e.g. SRB) generate feedback data (0 or ?). You can specify whether this data should be outputted with this type of command by switching the option on or off.

Response:

Response	Meaning
0	The command has been executed (if SRB 1(x) was executed previously)
?	Error (if SRB 1(x) was executed previously)
none	The command has been executed, or error if SRB 0(x) was executed previously

SRB?

Select Response Behavior Query

Output the response behavior of the current interface

Syntax: SRB?(x)

Parameters: None

Response: *q1(y)*

q1	Switch response output on/off
0	Off
1	on

EST?

Error Status Query

Output the response behavior of the current interface

Syntax: EST?p1(x)

Parameters: p1: Ignored, optional

Outputs existing errors and warnings in the form of a list for each selected subchannel. The individual errors/warnings of a subchannel are separated by a comma (a subchannel can have more than one error/warning). The individual subchannels are separated by a colon (:) for the error status values; see the table below.

Status	Value	Comments
No error	0	
FACTORY CAL ERR	15001	Factory calibration corrupted
CALIBRATION ERR	15020	Calibration lines have not produced a valid setting or calibration in progress
TEDS Error	15023	Error interpret TEDS
Hardware underflow	15030	Error in six wire circuit or value out of range
Hardware overflow	15031	Error in six wire circuit or value out of range
TEDS warning	20031	Warning interpret TEDS

IDS?

Identifier Settings Query

Read currently used time format

Syntax: IDS?p1(x)

Parameters: p1: Numeric value of the text access number

Response: q1: ID character string in English for p1

Example: IDS?15030(x)
"Hardware underflow"(y)

Supported text access numbers
15001, 15020, 15023, 15030, 15031, 20031

LSS?

Limit Switch Status Query

Output limit value status

Syntax:	LSS?(x)
Parameters:	p1: LIV1-Status OFF or ON: 0 or 1; p2: LIV2-Status OFF or ON: 0 or 1
	….
	p32: LIV32-Status OFF or ON: 0 or 1

LVL

Limit Value Level

Enter level of limit value

Syntax:	LVL p1,p2(x)
Parameters:	p1: Numbers of the limit switches (1...32)
	p2: Level of the limit value in displayed units (floating point); response depends on the SRB command.

LVL?

Limit Value Level Query

Output level of limit value

Syntax 1:	LVL?p1(x)
Parameters:	p1: Number of the limit switch (1...32)
Response:	*q1,q2(y)*
	q1: Number of the limit switch (1...32)
	q2: Level of the value in displayed units (floating point)
Syntax 2:	LVL??(x)
Parameters:	None
Response:	*q1,q2(y):* Available limit switches (range): 1, 32
Syntax 3:	LVL?,?(x)
Parameters:	None
Response:	*q1,q2(y):* Possible input range for level of value (floating point)

LVS

Limit Value Switch

Parameterizes limit value switches.

Input value is used as from the first signal defined with the command PCS and SPS.

Syntax: LVS p1,p2,p3,p4,p5,p6,p7,p8(x)

Parameters: p1: Number of the limit switch - (1...32)

p2: OPERATION (ON=1 or OFF=0)

P3	Direction
130	Above limit value
131	Below limit value
132	In band
133	Outside band

p4: Level of the value in displayed units (lower value in band mode) (floating point)

p5: Hysteresis (or width of the band in band mode) in displayed units (floating point)

p6: Reset behavior (can be used for hysteresis management): Binary mask for which an AND link is created with all digital inputs. If the result is >0 (true), this limit switch is cleared. Value range: 0,1,2,4,8,16,....32768. The reset behavior can be inverted with p7. Default value is 0. The parameter is optional.

p7: 0 or 1. Inverts the reset behavior.

0: Reset behavior works as defined with p6.

1: Reset behavior is inverted.

Default value is 0. The parameter is optional.

p8: 0 or 1.

0: Only if the measurement status is OK is the limit switch evaluated; freezes the status of the limit switch if the measurement status has one or more errors.

1: Measurement status is ignored.

Default value is 0. The parameter is optional.

LVS?

Limit Value Switch Query

Output limit switch parameter assignment

Syntax1: LVS?p1(x)

Parameters: p1: Number of the limit switch (1...32)

Response: *q1...q10(y)*

q1: Number of the limit switch (1...32); q2: Activated status (ON=1, OFF=0); q3: Input channel (slot)

q4: Input subchannel (signal)

q5: Operating direction (1(Off),130,131,132,133; see LVS command)

q6: Level of the value or lower value of the band in displayed units (floating point); q7: Hysteresis or width of the band in displayed units (floating point); p8: Reset behavior (binary mask, see LVS command)

q9: Inverts the reset behavior (0, 1, see LVS command)

q10: Ignores the measurement status (0, 1, see LVS command)

Syntax 2: LVS??(x)

Parameters: None

Response: *q1,q2(y):* Available limit switches (1..32)

SOP

Setup Output

Parameterizes digital outputs.

Measurement status is used as from the first signal defined with

the PCS and SPS command.

Syntax: SOP p1,p2,p3,p4,...,p17(x); p4 ... p17 are optional. Default value is 0.

Parameters: p1: Number of the digital output (1...16)

p2: Activation for limit switches (ON=1, OFF=0). If activated, the status of the 32 limit switches is used together with the mask for the limit switches (p3) to define the status of a digital output.

p3: Binary mask for limit switches for which an AND link is created with all limit switches.

Bit 0 of this mask is used for limit switch 1, bit 1 for limit switch 2, and so on. If the result is >0 (true), this digital output is set.

Value range: 0,1,2,4,8,16,....32768,65536,....2^30,2^31. A mask with more than one bit set is also possible. It could be used to induce "window behavior" of a digital output.

p4: Activation for measurement status (ON=1, OFF=0). If activated, the status value of a measurement status is used to define the status of a digital output. If this measurement status has an error, the digital output is set to 1 / On. (PCS / SPS)

p5: Activation for fieldbus bits (ON=1, OFF=0). If activated, the status value of a 32-bit word that can be written by a fieldbus master is used together with the fieldbus bits mask (p6) to define the status of a digital output.

p6: Binary mask for the fieldbus bitsfor which an AND link is created with the 32 fieldbus bits. If the result is >0 (true), this digital output is set. A mask with more than one bit set is also possible. Value range: 0,1,2,3...(2^32-1)

p7: Activation for digital inputs (ON=1, OFF=0). If activated, the status of the 16 limit digital inputs is used together with the mask for the digital inputs (p8) to define the status of a digital output.

p8: Binary mask for digital inputs for which an AND link is created with the 16 digital inputs. If the result is >0 (true), this digital output is set. A mask with more than one bit set is also possible. Value range: 0,1,2,3...(2^16-1)

p9: Activation for parameter set number (ON=1, OFF=0). If activated, the parameter set number is used together with the parameter set number mask (p10) to define the status of a digital output.

p10: Binary mask for the parameter set number for which an AND link is created with the currently active parameter set number. If the result is >0 (true), this digital output is set. A mask with more than one bit set is also possible. Value range: 0,1,2,3...(2^32-1)

p11: Activation for bits of calculated channels (ON=1, OFF=0). If activated, the status of the 32-bit word that can be defined by one or more calculated channels is used together with the calculated channels mask (p12) to define the status of a digital output.

p12: Binary mask for the calculated channels for which an AND link is created with the 32 bits of the calculated channels. If the result is >0 (true), this digital output is set. A mask with more than one bit set is also possible. Value range: 0,1,2,3...(2^32-1)

p13: Activation for CodeSys bits (ON=1, OFF=0). If activated, the status of the 32-bit word that can be defined by the CodeSys application is used together with the mask for the CodeSys bits (p14) to define the status of a digital output. If CodeSys is not available, the value is 0.

p14: Binary mask for the CodeSys bits for which an AND link is created with the 32 bits of the CodeSys application. If the result is >0 (true), this digital output is set. A mask with more than one bit set is also possible. Value range: 0,1,2,3...(2^32-1).

If CodeSys is not available, the value is 0.

p15: Activation for system status bits (ON=1, OFF=0). If activated, the status of the 32 bits of the system status is used together with the system status mask (p16) to define the status of a digital output.

p16: Binary mask for the system status bits for which an AND link is created with the 32 system status bits. If the result is >0 (true), this digital output is set. A mask with more than one bit set is also possible. Value range: 0,1,2,3...(2^32-1)

p17: Inverts the digital output (0: as described before; 1: Inverted).

SOP?

Setup Output Query

Output parameter assignment of a specific digital output

Syntax1: SOP?p1(x)

Parameters: p1: Number of the digital output (1...16)

Response: *q1...q19(y)*

q1: Number of the digital output (1...16)

q2: Activation for limit switches (ON=1, OFF=0). If activated, the status of the 32 limit switches is used together with the mask for the limit switches (q3) to define the status of a digital output.

q3: Binary mask for limit switches for which an AND link is created with all limit switches.
Bit 0 of this mask is used for limit switch 1, bit 1 for limit switch 2, and so on. If the result is >0 (true), this digital output is set. Value range: 0,1,2,4,8,16,....32768,65536,....2^30,2^31. A mask with more than one bit set is also possible. It could be used to induce "window behavior" of a digital output.

q4: Activation for measurement status (ON=1, OFF=0). If activated, the status value of a measurement status is used to define the status of a digital output. If this measurement status has an error, the digital output is set to 1 / On.

q5: Input channel (slot)

q6: Input subchannel (signal)

q7: Activation for fieldbus bits (ON=1, OFF=0). If activated, the status value of a 32-bit word that can be written by a fieldbus master is used together with the fieldbus bits mask (q8) to define the status of a digital output.

q8: Binary mask for the fieldbus bitsfor which an AND link is created with the 32 fieldbus bits. If the result is >0 (true), this digital output is set. A mask with more than one bit set is also possible. Value range: 0,1,2,3...(2^32-1)

q9: Activation for digital inputs (ON=1, OFF=0). If activated, the status of the 16 limit digital inputs is used together with the mask for the digital inputs (q10) to define the status of a digital output.

q10: Binary mask for input channels for which an AND link is created with the 16 input channels. If the result is >0 (true), this digital output is set. A mask with more than one bit set is also possible. Value range: 0,1,2,3...(2^16-1)

q11: Activation for parameter set number (ON=1, OFF=0). If activated, the parameter set number is used together with the parameter set number mask (q12) to define the status of a digital output.

q12: Binary mask for the parameter set number for which an AND link is created with the currently active parameter set number. If the result is >0 (true), this digital output is set. A mask with more than one bit set is also possible. Value range: 0,1,2,3...(2^32-1)

q13: Activation for bits of calculated channels (ON=1, OFF=0). If activated, the status of the 32-bit word that can be defined by one or more calculated channels is used together with the calculated channels mask (q14) to define the status of a digital output.

q14: Binary mask for the calculated channels for which an AND link is created with the 32 bits of the calculated channels. If the result is >0 (true), this digital output is set. A mask with more than one bit set is also possible. Value range: 0,1,2,3...(2^32-1)

p15: Activation for CodeSys bits (ON=1, OFF=0). If activated, the status of the 32-bit word that can be defined by the CodeSys application is used together with the mask for the CodeSys bits (p16) to define the status of a digital output. If CodeSys is not available, the value is 0.

p16: Binary mask for the CodeSys bits for which an AND link is created with the 32 bits of the CodeSys application. If the result is >0 (true), this digital output is set. A mask with more than one bit set is also possible. Value range: 0,1,2,3...(2^32-1).

If CodeSys is not available, the value is 0.

p17: Activation for system status bits (ON=1, OFF=0). If activated, the status of the 32 bits of the system status is used together with the system status mask (p18) to define the status of a digital output.

p18: Binary mask for the system status bits for which an AND link is created with the 32 system status bits. If the result is >0 (true), this digital output is set. A mask with more than one bit set is also possible. Value range: 0,1,2,3...(2^32-1)

q19: Inverts the digital output (0: as described before; 1: Inverted).

Syntax 2: SOP??(x)

Parameters: None

Response: *q1,q2(y):* Available digital outputs (1..16)

RIP?

Read Digital Input query

Read and output the digital inputs

Syntax: RIP?(x)

Parameters: None

Effect: Reads the 16 possible digital inputs of the PMX device and outputs the binary status of each input as an integer value between 0 and 65535. The least significant 8 bits realize the 8 inputs of the first PX878. The most significant 8 bits realize the 8 inputs of the second PX878.

Bits 16 ... 31 are always virtual.

Example: RIP?(x)
1025(y)

Input 3 (bit 10 (8+2)) of the 2nd PX878 and input 1 (bit 0) of the 1st PX878 are set (the input count starts at 1).

ROP

Set Digital Outputs

Set the digital outputs

Syntax: ROP p1, p2(x)

Parameters: p1: Binary representation of all outputs, 0...65535
p2: Optional, binary representation of the selected outputs. If no value is specified, all 16 outputs are set to the status

defined by p1.
Value for p2: 0...65535, default value: 65535

Effect: Sets the 16 possible digital outputs of the PMX device. The least significant 8 bits realize the 8 outputs of the first PX878. The most significant 8 bits realize the 8 outputs of the second PX878.
The outputs are available, even if there is no PX878 connected. In this case, they are purely virtual. They can be set or reset, but are not present electrically.
p2 defines the selected bits whose corresponding output is set or cleared with p1. Outputs, the corresponding bits of which are 0 in p2, are not affected.

Bits 16 ... 31 are always virtual.

Example: ROP2, 32770(x)
0(y)

Output 8 (bit 15) of the 2nd PX878 is cleared and output 2 (bit 1) of the 1st PX878 is set (the output count starts at 1).
Only these two bits are selected with p2. All the other outputs are unchanged.

Notice

The outputs of the PMX device are defined by the settings stored in the (sub) parameter sets that can also be enabled. This command changes the settings of the selected outputs of the sub-parameter sets being used so that the output switches to the desired status. If a parameter set is connected, an earlier ROP command is overwritten.

ROP?

Read Digital Output Query

Digital outputs

Syntax: ROP? (x)

Parameters: None

Effect: Reads the 16 possible digital outputs of the PMX device and outputs the binary status of each input as an integer value between 0 and 65535. The least significant 8 bits realize the 8 outputs of the first PX878. The most significant 8 bits realize the 8 outputs of the second PX878.
The outputs are available, even if there is no PX878 connected. In this case, they are purely virtual. They can be set or reset, but are not present electrically.

Example: ROP?(x)
32770(y)

Output 8 (bit 15) of the 2nd PX878 and output 2 (bit 1) of the 1st PX878 are set (the output count starts at 1).

OSP?

Output Signal Path Query (only PX878)

Analog outputs

Responds to the source channel and source subchannel of the analog output(s) of the PX878 previously selected with PCS and SPS.

Syntax:	OSP? (x)
Parameters:	none
Effect:	Source channel**,** source-sub channel **:** source-channel**,** source-subchannel … (y)
Example:	OSP?(x) *1,4: 9, 1 … (y)*

OSP

Output Signal Path (only PX878)

Analog outputs

Sets the source channel and source subchannel of the analog output(s) of the PX878 previously selected with PCS and SPS, and deactivates any active test mode that may have been previously activated with SAO.

Syntax:	OSP p1, p2 (x)
Parameters:	p1: source-channel (slot) p2: source-subchannel
Example:	OSP 1,4 (x)

SAO?

Set analog Output Query (only PX878)

Analog outputs

Responds to the voltage(s) of the test signal(s) of the analog output(s) of the PX878 previously selected with PCS and SPS. This does not mean that the test signal is active and that the output voltage(s) is/are currently being passed to the output(s).

Syntax:	SAO? (x)
Parameters:	none
Response:	voltage**,** voltage**,** … (y)
Example:	SAO?(x) *1.1, -4.2, … (y)*

! **Important**

This command is implemented in PMX firmware 2.00 and higher.

SAO

Set analog Output (only PX878)

Analog outputs

Sets the voltage of all selected analog outputs from one or more PX878 multi-I/O cards to a given level (10 V ... +10 V). The command activates a test mode and disconnects the analog output from the path of its previously connected signal source. The OSP command is used to deactivate the test mode.

Syntax:	SAO p1 (x)
Parameters:	p1: voltage
Example:	SAO 1.25(x)
Note:	This command causes a heavy load on the CPU. A setting of 10 values per second for a single analog output increases the CPU load by about 15%.

Important

This command is implemented in PMX firmware 2.00 and higher.

21.3 Examples

Simple case of a measured value configuration

Terminology:

Example of a PMX command list in a Telnet session under Microsoft Windows

PMX names	Catman interface names
Occupied card slots	Channels
Physical channels on a card	Subchannels
Types of internal channels: Original, Min, Max, PP	Signals: Gross, Min, Max, Max-Min

Example:

Select a filter globally, and capture subchannels that have been divided into measurement rate groups.

Please do not change the example in this document, because a sample code refers to it!

pcs 0 (x) sps 0 (x) sfc 141,969 (x)
pcs 1 (x) sps 3,4 (x) mrg 0 (x)
pcs 2 (x) sps 1,2 (x) mrg 1 (x)
icr 6320,0 (x)
icr 6319,1 (x)
pcs 1 (x) sms 3,4 (x) sps 3,4 (x) mss 214 (x)
pcs 2 (x) sms 1,2 (x) sps 1,2 (x) mss 214 (x)
mcs 1,2 (x)
tsv 10 (x)

omp? 0 (x) omp? 1 (x)
rmb? 10,6409,0 (x) rmb? 10,6409,1 (x)

(x)
Command terminator: [CR][LF]
?
The question mark is itself part of query commands that output values other than a confirmation.

This means:

Set a Butterworth filter with 1000 Hz global.

Select all cards	**Select all their subchannels**	Select Butterworth filter characteristics with cut-off frequency **1000 Hz**
pcs **0**	sps **0**	Sfc 141, **969**

Sort cards and their subchannels into measurement rate groups.

Select card **1**	Select its subchannels **3** and **4**	Assign the selection to measurement rate group **0**
pcs **1**	sps **3.4**	mrg **0**
Select card **2**	Select its subchannels **1** and **2**	Assign the selection to measurement rate group **1**
pcs **2**	sps **1.2**	mrg **1**

Set sample rates for measurement rate groups.

Select 9600 Hz for measurement rate group **0**
icr 6320, **0**

Select 4800 Hz for measurement rate group **1**
icr 6319, **1**

Set recording mask (in this example the same subchannels are selected that were assigned to the measurement rate groups above).

Select card **1**	Set its subchannels **3** and **4** for recording	Select subchannels **3,4** of the previously selected card 1	Set "**gross**" signal (= dynamic actual measured values) for selected subchannels
pcs 1	sms 3,4	sps 3,4	mss214
Select card **2**	Set its subchannels **1** and **2** for recording	Select subchannels **1,2** of the previously selected card 2	Set "gross" signal (= dynamic actual measured values) for selected subchannels
pcs **2**	sms **1,2**	sps **1.2**	mss **214**

Set cards **1,2** for recording
mcs **1,2**

Record a set of values

tsv **10**

Check whether the line of values has arrived in the FIFO buffer

Get available lines for measurement rate group **0**	Get available lines for measurement rate group **1**
omp? **0**	omp? **1**
	Response: 2 bytes header "#0", 80 bytes values, 2 bytes response end sequence CRLF.

Retrieve the values recorded in the FIFO buffer

Retrieve 10 value lines from measurement rate group **0** and set the internal FIFO read pointer to it (constant ADJUST_READ_POINTER = 6409)	**Retrieve 10** value lines from measurement rate group **1** and set the internal FIFO read pointer to it
rmb? **10,** 6409**, 0**	rmb? **10,** 6409**, 1**
Response: 2 bytes header "#0", 80 bytes values, 2 bytes response end sequence CRLF.	Response: 2 bytes header "#0", 80 bytes values, 2 bytes response end sequence CRLF.
These are 20 values, each consisting of 4 bytes in floating point format. (10 lines (sets) with two subchannels, each with an active signal).	

Notes

- Spaces within or between commands are ignored, and are optional. The commas between the parameters are very important.
- Channels and subchannels are counted 1, 2..., and 0 means "all", whereas measurement rate groups are counted 0, 1, 2, as there is no "all" of them.
- Some commands may be optional, as they only select the default setting. If the default setting has not been changed before, they can be omitted: pcs 0 sps 0 mss 214
- The default setting of the recording mask (if not preset) is: all occupied channels (except time channel), all their subchannels, and for all of them the "gross" signal.
- A *line* value is also referred to as a *page*. It is a set of values defined by the configuration of the measurement rate group. For the above example, this means that the lines have the size of two floating point values for each measurement group, as two subchannels (each with an active signal) have been assigned to one

group. A floating point value consists of 4 bytes. So the measured values arrive as multiples of 2 floating point values (size(line)=2) or 8 bytes.

- In the above example, the selections are made two times via pcs and sps. This can be packaged:

  pcs 1 (x) sps 3,4 (x) mrg 0 (x)
  pcs 2 (x) sps 1,2 (x) mrg 1 (x)
  icr 6320,0 (x)
  icr 6319,1 (x)
  pcs 1 (x) sms 3,4 (x) sps 3,4 (x) mss 214 (x)
  pcs 2 (x) sms 1,2 (x) sps 1,2 (x) mss 214 (x)

  This is identical to:

  pcs 1 (x) sms 3,4 (x) sps 3,4 (x) mss 214 (x) mrg 0 (x)
  pcs 2 (x) sms 1,2 (x) sps 1,2 (x) mss 214 (x) mrg 1 (x)
  icr 6320,0 (x)
  icr 6319,1 (x)

22 OBJECT DICTIONARY

The object dictionary is a collection of PMX setup and status parameters. So as from firmware version 3.02 PMX parameters can be changed via a PC or PLC program.

The PMX object dictionary is *not* the EtherCAT® dictionary of cyclic data objects. The EtherCAT® objects are only visible for the EtherCAT® master.

22.1 Accessible data objects

The object dictionary contains all relevant parameters of the PMX plug-in cards (except the fieldbus card) and the created calculation channels in the PMX. It also includes objects for switching operator level.

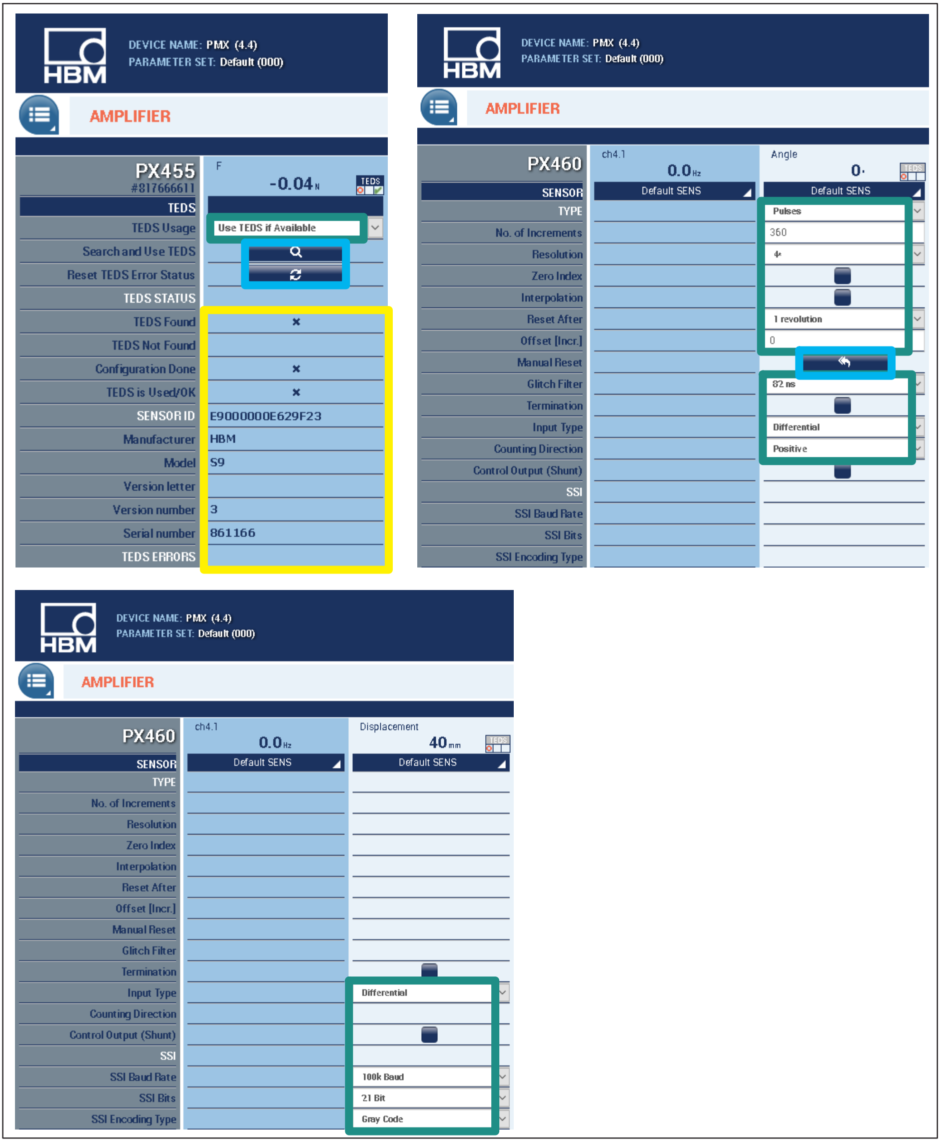
All other available device parameters can be found in *chapter 21, page 346*, of the .NET API or in the device description files of the fieldbuses (*chapter 16, page 259*).

The marked parameters in the screenshots give an impression of the accessible data objects. Data objects that are transmitted periodically via the Ethernet interface or the fieldbus anyway are not visible in the object dictionary.

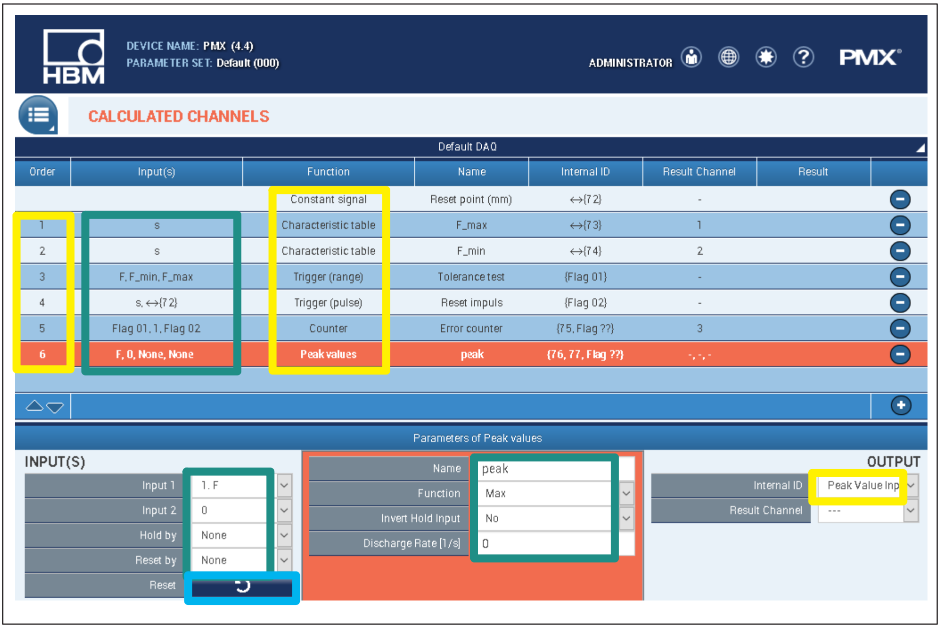
Read only	Read/write	Write only

22.1.1 Measurement channels

The object dictionary contains practically all the parameters from the Amplifiers dialog box.


HBM
DEVICE NAME: PMX (4.4)
PARAMETER SET: Default (000)
AMPLIFIER
PX455
#817666611
F
-0.04 N
TEDS
TEDS Usage
Use TEDS if Available
Search and Use TEDS
Reset TEDS Error Status
TEDS STATUS
TEDS Found
TEDS Not Found
Configuration Done
TEDS is Used/OK
SENSOR ID
E9000000E629F23
Manufacturer
HBM
Model
S9
Version letter
Version number
3
Serial number
861166
TEDS ERRORS
HBM
DEVICE NAME: PMX (4.4)
PARAMETER SET: Default (000)
AMPLIFIER
PX460
ch4.1
0.0 Hz
Angle
0·
SENSOR
Default SENS
Default SENS
TYPE
Pulses
No. of Increments
360
Resolution
Zero Index
Interpolation
Reset After
1 revolution
Offset [Incr.]
0
Manual Reset
Glitch Filter
82 ns
Termination
Input Type
Differential
Counting Direction
Positive
Control Output (Shunt)
SSI
SSI Baud Rate
SSI Bits
SSI Encoding Type
HBM
DEVICE NAME: PMX (4.4)
PARAMETER SET: Default (000)
AMPLIFIER
PX460
ch4.1
0.0 Hz
Displacement
40 mm
SENSOR
Default SENS
Default SENS
TYPE
No. of Increments
Resolution
Zero Index
Interpolation
Reset After
Offset [Incr.]
Manual Reset
Glitch Filter
Termination
Input Type
Differential
Counting Direction
Control Output (Shunt)
SSI
SSI Baud Rate
100k Baud
SSI Bits
21 Bit
SSI Encoding Type
Gray Code

22.1.2 Calculated channels



22.2 Numbering plan

A data object is addressed by:

- The index 0x4000 ... 0x41ff, which is normally shown in hexadecimal notation.
- The subindex 0 ... 255, which is normally shown in decimal notation.

For example, 0x4123.45 denotes the data object with the index 0x4123 and the subindex 45.

22.2.1 General objects

Index	Name	
0x4001.1	Apply	By writing "1" to this object, the previously changed parameters are applied.
0x4002.1	Save all parameters	To save all settings in non-volatile memory. This parameter calls the same function as the Save icon  on the web user interface. Note: The command returns immediately, although the save operation takes a few seconds.

22.2.2 Measurement channels

Index	Data objects from...	Subindex
0x4010	Slot 1	The subindexes are defined by the actually installed module type. They are listed in the generated files.
0x4011	Slot 1, signal 1	
0x401x	Slot 1, signal x	
0x4020	Slot 2	
0x4021	Slot 2, signal 1	
0x4030	Slot 3	
0x4040	Slot 4	

22.2.3 Calculated channels

Index of the function block = 0x40A0 + calculation position

Index	Data objects from...	Subindex
0x40A1	The function block at calculation position 1 	The subindexes depend on the function block type. They are listed in the generated files.
0x40A2	The function block at calculation position 2	
...		
0x40D0	The function block at calculation position 48	

22.2.4 Constant signals

0x40E1	User defined constant signal #1	8: Output signal identifier 22: Name 30: Value
0x40E2	User defined constant signal #2	
...		

22.2.5 Passwords

The passwords in the PMX browser for the MAINTENANCE and ADMINISTRATOR user levels can be temporarily disabled, such as via a PLC by service access. They are disabled via the data object 0x4003 subindex 1 in the object dictionary with access via fieldbus, command interface (Ethernet), Common API, or CODESYS.

Data format of 0x4003 (uint32):

Bit 17 = 1 (0xnnn2 nnnn) unlocks the ADMIN level.

Bit 16 = 1 (0xnnn1 nnnn) unlocks the MAINTENANCE level. Bits 16 and 17 can also be set simultaneously.

Bits 0 ... 15 (0xnnnn TTTT) contain the timeout in minutes. Range 1 ... 1440 minutes. Larger values are limited to 1440.

Examples:

0x4003.1 = 0x0001 000A: MAINTENANCE level enabled for 10 minutes

0x4003.1 = 0x0002 05A0: ADMIN level enabled for 1440 minutes = 24 h

0x4003.1 = 0x0000 0000: OPERATOR level, GUI locked by passwords.

22.3 Data types

The object dictionary supports the following data types from IEC 61131.

BOOL	1 bit	
USINT	8 bit unsigned	
SINT	8 bit signed	
UINT	16 bit unsigned	
INT	16 bit signed	
UDINT	32 bit unsigned	
DINT	32 bit signed	
ULINT	64 bit unsigned	Not accessible via fieldbus
LINT	64 bit signed	Not accessible via fieldbus
REAL	32 bit with floating point	
LREAL	64 bit with floating point	Not accessible via fieldbus
STRING		Not accessible via fieldbus

22.4 Access via Ethernet command interface

For general information about the command interface refer to the *PMX operating manual, chapter 21, page 346*.

The **oda** (Object Dictionary Access) command is used to write or read individual data objects via Ethernet port 55 000.

Query	oda? *index,subindex*	Queries the value of the data object index: The index of the data object in decimal or hexadecimal notation subindex: The subindex of the data object in decimal or hexadecimal notation
Res-ponse	index,subindex, value, error_code	Response from PMX index: The index of the query in decimal notation subindex: The subindex of the query in decimal notation subindex: The value of the data object error_code: 0: No error, the returned value is valid 1: Access error (e.g. attempt to read a write-only object) 2: Format error (e.g. data type is not supported) 4: Not found, the data object does not exist

Example:

oda? 0x4011,13
Read output voltage from object 0x4011.13, slot 1.1 (analog output PX878)

16401,13,0.125,0
(Successful, the voltage is 0.125 V)

oda? 0x4fff,1 (Read object 0x4fff.1)

20479,1,0,4 (Failed, the data object does not exist)

Setting	oda *index,subindex, value*	Set value index: The index of the data object in decimal or hexadecimal notation subindex: The subindex of the data object in decimal or hexadecimal notation value: Value to be set. The programmer is responsible for ensuring that the value can be converted to the data object type.
Res-ponse	index,subindex error_code	Response from PMX index: The index of the query in decimal notation subindex: The subindex of the query in decimal notation error_code: 0: No error, the parameter was updated 1: Access error (e.g. attempt to write a write-only object) 2: Format error (e.g. wrong data type) 4: Not found, the data object does not exist

Example:

oda 0x4011,13,1.2
Set test signal of slot 1.1 (=output voltage) to 1.2 V)

16401,13,0
(successful)

oda 0x4011,14,1
(Activate test signal of slot 1.1)

16401,14,0
(successful)

oda 0x4011,19,"my channel" (set channel name of slot 1.1)

16401,19,0
(successful)

22.5 Access via fieldbus

For general information about fieldbus communication see *chapter 16, page 259*.

Note: Only data objects whose value can be expressed in 32 bits are transferred via the fieldbus. They are BOOL, SINT, INT, DINT, USINT, UINT, UDINT and REAL.

LREAL values are transferred as REAL values, and lose a little accuracy in the process.

Before sending a request over the fieldbus, the service must be activated with bit 1 in the device control word.

Output data PLC ⇒ PMX

Device data (cyclic)

PROFINET®/EtherCAT®

Function		EtherCAT® Index	PROFINET® Slot.Subslot	Data type
Device control word	Bit1 (value 0x02): Enable object dictionary server	7000.1	0.2 bytes 0..3	uint32

EtherNet/IP™

Function		EtherCAT® Index	PROFINET® Slot.Subslot
Index	**Size in octets**	**Type**	**Day**
0..3	4	UDINT	PMX Control

22.5.1 Send a request

Read and write requests to the object dictionary are transmitted via the (previously unused) "GUI signaling" data word. For bit assignment see *section 16.5.1, page 266, and section 16.9.3, page 280*.

Output data PLC ⇒ PMX

Device data (cyclic)

PROFINET®/EtherCAT®

Function		EtherCAT® Index	PROFINET® Slot.Subslot	Data type
GUI signaling	Object dictionary command	7000.3	0.2 bytes 8..15	uint64

EtherNet/IP™

Index	Size in octets	Type	Day	
8..15	8	ULINT	UiControl	SystemData (transmitted always)

22.5.2 Bit assignment

Bit assignment for request and response

<table>
<tr><th>Bit
63 … 56</th><th>Bit
55 … 48</th><th>Bit
47 … 40</th><th>Bit
39 … 32</th><th>Bit
31 … 24</th><th>Bit
23 … 16</th><th>Bit
15 … 8</th><th>Bit
7 … 0</th></tr>
<tr><td>Control and status flags</td><td>Subindex</td><td colspan="2">Index</td><td colspan="4">Value</td></tr>
<tr><td></td><td></td><td colspan="2"></td><td colspan="4">DINT, UDINT, REAL</td></tr>
<tr><td></td><td></td><td colspan="2"></td><td colspan="2">Not used, set to zero!</td><td colspan="2">INT, UINT</td></tr>
<tr><td></td><td></td><td colspan="2"></td><td colspan="3">Not used, set to zero!</td><td>SINT, USINT</td></tr>
<tr><td></td><td></td><td colspan="2"></td><td colspan="4">BOOL (true if > 0)</td></tr>
</table>

The control/status flags

Bit number in 64-bit word	Bit number in octet	PLC sets control bits PMX sets status bits	
63	7	Control system	Read request. Set this bit to read a data object.
62	6	Control system	Write request. Set this bit to write to a data object.
61	5	Control system	Reread (not available with a write request) 0: PMX responds once 1: PMX continually updates the response until the following request is sent

Bit number in 64-bit word	Bit number in octet	PLC sets control bits PMX sets status bits	
60	4		Not in use
59	3		Not in use
58	2	Status	Not found, the data object does not exist
57	1	Status	Format error (e.g. wrong data type)
56	0	Status	Access error (e.g. attempt to write a write-only object)

Bit assignment of REAL values (32 bits with floating point)

Sign	Exponent	Fraction
Bit 31	Bits 30..23	Bits 22..0

22.5.3 Response from PMX

PMX responds in the (previously unused) "GUI status" data word. For bit assignment see *section 16.4, page 262, and section 16.9.3, page 280*.

Input data PMX ⇒ PLC

Device data (cyclic)

PROFINET®/ EtherCAT®

Function		EtherCAT® Index	PROFINET® Slot.Subslot	Data type
GUI status	Object dictionary response	6000.3	0.2 bytes 8..15	uint64

EtherNet/IP™

Function		EtherCAT® Index	PROFINET® Slot.Subslot	Data type
Index	Size in octets	Type	Day	
8..15	8	ULINT	UiStatus	SystemData (transmitted always)

22.5.4 Response to a read request

PMX copies the index, the subindex, the control flags and the requested value into the response.

The request has been successfully processed if bits 32..63 of the response match bits 32..63 of the request. Bits 0..31 contain the requested value.

Do not use the value if one of the error flags is set.

22.5.5 Response to a write request

PMX copies all data from the request to the response.

The request has been successfully processed if all 64 bits of the response match the request.

22.5.6 Retry

To resend a failed request, at least one bit of the request must be changed.
Normally PMX waits for changes in the request.

Example:

To resend a read request, toggle the read control bit (bit 63).

1. Set the read control bit = 0
2. Wait for read control bit = 0 in the response
3. Set the read control bit = 1
4. Check the response

22.6 Applying the new value

Most values must be explicitly applied after they have been changed.

Write "1" to 0x4001.01 to apply the changed values (command oda 0x4001,1,1).

It is a best practice to change all parameters first, and then set 0x4001.01 to apply all values simultaneously. (Note: In contrast, any parameter change on the web interface takes effect immediately.)

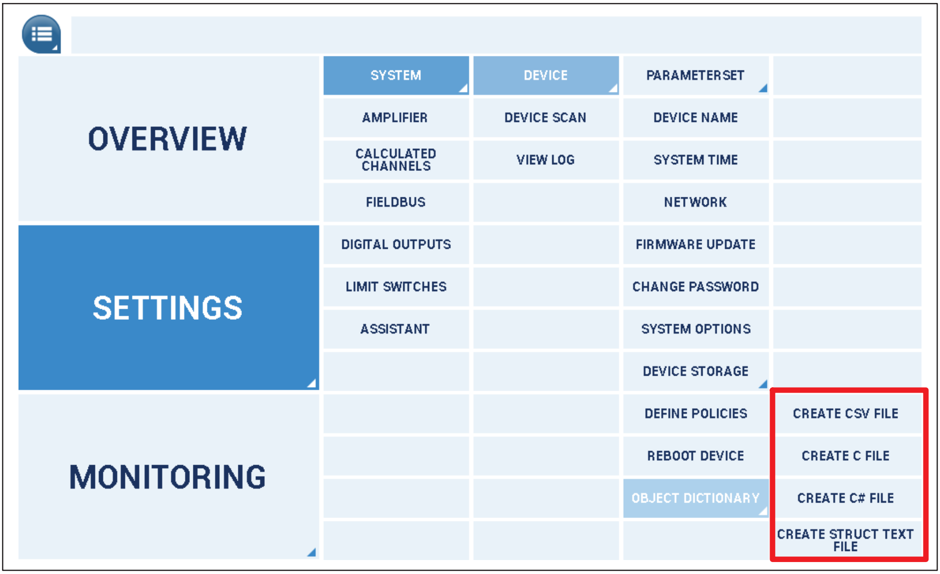
These values do not have to be applied explicitly. They become effective immediately after writing:

Index	Subindex	Name
0x40yz where y = 1..4	10 11 15 16	set_zero zero_value test_signal test_signal_enable
(function blocks) Peak Hold digital	33	reset_now

22.7 Generated header files

The PMX generates header files to assist your programming.

First, set up the calculated channels via the web user interface. Then have the PMX generate a CSV, C, C# or ST (Structured Text) file.



Note that two *C* files are created. The browser download window is displayed twice.

The *Structured Text* (ST, SCL) file for PLC can only be imported into certain PLC configuration tools. If this is not possible, copy and paste the content into your source code.

The files contain a list of all data object, type and constant definitions.

Important

Note that adding, moving or deleting calculated channels will change the object dictionary. The files must be generated again.

22.7.1 Value ranges of the objects

Most data objects are just numbers. The allowed data range is specified in the list for the object.

Example from the C header files

Object 0x4021.19 is the filter type of slot 2.1.

The string "FILTER_CHARACTERISTIC" defines the value range

```
{ 0x4021, 19, 2, 1, odDINT, 1, ACCESS_RW, "[slot2.1] filter_type","FILTER_CHARACTERISTIC"}
```

The allowed values can be found in the .h file.

/* FILTER_CHARACTERISTIC */ enum FILTER_CHARACTERISTIC{ fltBessel = 0, fltButterworth = 1 };

Important

Note that most data objects are NOT CHECKED for overrange or underrange by the firmware. You are responsible for that

RANGE_AS_DATA_TYPE specifies an unrestricted range. The value can be virtually any number of the data type, but is usually constrained by the context behind it.

```
{ 0x4021, 15, 2, 1, odREAL, 1, ACCESS_RW, "[slot2.1] test_signal","RANGE_AS_DATA_TYPE"}
```

22.8 Tips on using the object dictionary

Best sequence for using calculated channel objects:

1. Set up the calculated channels via the web user interface.
2. Have the PMX create the files with definitions and data objects.
3. Edit the data objects using your PC-based or PLC program.

Important

When the calculation sequence of calculated channels is changed, the indexes of the corresponding data objects also change.
When function blocks are created/deleted, the corresponding data objects are also created/deleted.
Most data objects are NOT CHECKED for overrange or underrange by the firmware. The user is responsible for entering permissible data.
The performance when accessing data objects over the fieldbus is normally 25 ... 35 ms per request.

23 QUALITY & CALIBRATION CERTIFICATES

Documented quality:

HBM calibration certificates for the configured measurement cards and a declaration of compliance with the order 2.1 according to DIN EN 10204 are stored as PDF documents in the PMX's device memory (public -> certificates) on shipping.

Download it from there using the PMX browser and the **Device storage** menu.

If the device is recalibrated at HBM, the new calibration certificates will again be stored in the device memory. This ensures full and complete documentation.

A calibration certificate is not compiled for fieldbus cards. Correct operation is documented by the certificate of compliance.

If the PDF documents have been deleted from the device memory, you can obtain a replacement certificate from HBM Technical Support at: support@hbkworld.com.

24 FIRMWARE UPDATE

24.1 Preparation

You can update one or more PMX devices at the same time. The devices must be connected to the PC (HOST) in order to do so.

A firmware update takes about 15 minutes. The device is *not* ready for measurement while the firmware is being updated.

Download the latest firmware file from HBM at https://www.hbm.com/de/2981/pmx-modular-measuring-amplifier-system-for-the-iot/ .

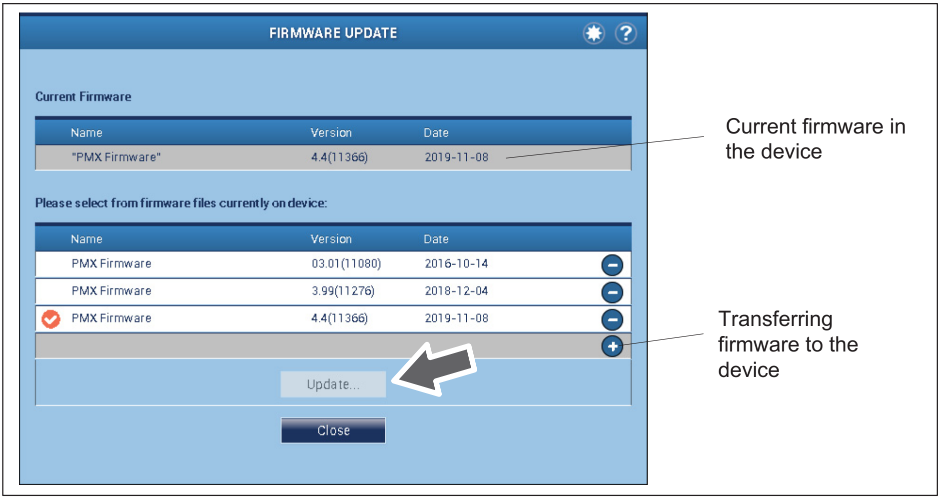
To identify a specific device:

- Select a device using the checkbox and click **Flash**.
  All controllable LEDs (system LED, measurement card LED) on the selected PMX flash by way of identification.

- Click **Connect**
- Copy the firmware file, e.g. "PMX_01.107412M.tgz", to the local PC (HOST).

24.2 Installing firmware

1. From the menu choose **Settings -> System -> Device -> FIRMWARE UPDATE**.



2. Transfer the new firmware to the device by clicking the plus sign (**+**) and specifying the location. You can choose whether you only want to load the firmware into the device, or also apply it immediately.
3. If the firmware is in the device, select the firmware you want by clicking on the corresponding entry. A tick will indicate your selection.
4. Start the update by clicking the **Update** button.

The browser will reconnect to the device after the update.

Important

Two outcomes are possible if the power supply fails during the firmware update:
1. The PMX will restart with its factory settings after being switched on, or
2. The device loads and initializes the new firmware, and will be ready to run after 10 to 15 minutes.

Tip

The device setup and parameter sets are retained after a firmware update. We nevertheless recommend creating a backup on a PC before updating the firmware. As of firmware version 2.0, CODESYS applications and CODESYS WebVisu are also retained after updating to a higher firmware version.

25 DIAGNOSIS & MAINTENANCE (HEALTH MONITORING)

Before actually starting to measure, you should check your system.

25.1 Error messages/operating state (LED display)

For the system to be ready for measurement, the LEDs on the basic device and plug-in cards must indicate the states described in *sections 8.2.3 to 8.2.5 and section 8.1, starting on page 46*.

If this is not the case, follow the instructions under "Remedy" in the tables below.

SYS LED

LED	Status	Meaning	Remedy
Green	On Off	Voltage supply available Voltage supply off	- Check the voltage supply
Yellow	On Flashing	Device is booting Factory settings not OK	. Send in the device
Red	Flashing On	Serious internal error Firmware update in progress	Check the mounting of the plug-in card and replace if necessary.

PX01EC, EtherCAT®

LED	LED	Status	Meaning
ERR Error state	-	Off	No error
	Red	Flashing	Configuration error: The configuration on the PLC side (master) must *exactly match* the configuration of the PMX (slave), e.g. card types in slots 1 to 4 and number of calculated channels (see PMX browser, **Fieldbus** menu).
		Single flash	Synchronization error
		Double flash	Application timeout error
		On	PDI timeout error

PX01PN, PROFINET®

LED	LED	Status	Meaning
SF System error	Red	On	No connection or no valid license
		Flashing	Configuration error: The configuration on the PROFINET® master side (PLC) must *exactly match* the configuration of the PMX, e.g. card types in slots 1 to 4 and number of calculated channels (see PMX browser, **Fieldbus** menu).
BF Bus error	Red	On	No connection or no valid license
		Flashing	Incorrect configuration, not all IO devices are connected.

EtherNet/IP™

<table>
<tr><th>LED</th><th>LED</th><th>Status</th><th>Meaning</th></tr>
<tr><td rowspan="7">MS
Mode status</td><td colspan="3">Duo LED red/green</td></tr>
<tr><td>-</td><td>Off</td><td>Not switched on: The device is not switched on.</td></tr>
<tr><td rowspan="2">Green</td><td>On</td><td>Device ready for operation: The device is in operation and running correctly.</td></tr>
<tr><td>Flashing</td><td>Standby: The device has not been configured.</td></tr>
<tr><td rowspan="2">Red</td><td>On</td><td>Serious error: The device has detected an irreparable serious error.</td></tr>
<tr><td>Flashing</td><td>Simple error: The device has detected a reparable simple error.
The configuration on the PLC side (master) must exactly match the configuration of the PMX (slave); card types in slots 1 to 4 and number of calculated channels (see PMX browser, Fieldbus) menu.
Note: A faulty or inconsistent configuration, for example, is classified as a simple error.</td></tr>
<tr><td>Red/Green</td><td>Flashing</td><td>Self-test: The device runs through its self-test.</td></tr>
</table>

LED	LED	Status	Meaning
NS Network status	**Duo LED red/green**		
	-	Off	**Not switched on, no IP address:** The device has no IP address (or is not switched on).
	Green	On	**Connected:** The device has at least one existing connection to another device (also to the message router).
		Flashing	**No connections:** The device has no existing connections to another device, but has received an IP address.
	Red	On	**Duplicate IP:** The device has detected that its IP address is already in use.
		Flashing	**Connection timeout:** One or more of the connections to this device has timed out. This status is only terminated when all timed-out connections have been restored, or when the device is reset.
	Red/Green	Flashing	**Self-test:** The device runs through its self-test.

PX401, channel status

LED	Status	Meaning	Remedy
Green	On	No errors	-
Yellow	Flashing	Firmware update in progress	-
Red	On	Parameter not OK, overloaded	Check: Sensor, sensor leads, TEDS module, send in the card if necessary

PX455, channel status

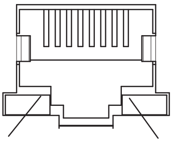
LED	Status	Meaning	Remedy
Green	On	No errors	-
Yellow	On Flashing	No transducer connected or wire break (calibration in progress) Firmware update in progress	Connecting transducers
Red	On	Parameter not OK, transducer error, overloaded	Check: Sensor, sensor leads, TEDS module, send in the card if necessary

PX878, channel status

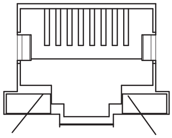
Analog			Remedy
Green	On	Analog output configured	-
Yellow	Flashing	Firmware update in progress	-
Red	On	Analog output overloaded, signal invalid or no signal assigned	Check the sensor signal, check the settings for the analog output channel

Synchronization SYNC

IN socket LEDs:

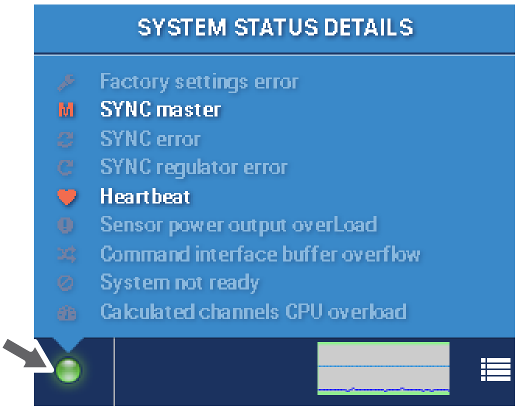
IN		Meaning	Remedy
			
Green	Off	Slave	-
Off	Off	Master	-
Off	Yellow	Error	Check the cable connection to the master/slave

OUT socket LEDs:

OUT		Meaning	Remedy
			
Green	Off	Power on	-
Off	Yellow	Error (always identical to the right-hand LED of the IN **socket)**	Check the cable connection to the master/slave

25.2 Device status error messages

The device status is indicated directly on the PMX by the device LED (green=OK / red=error) In the event of an error message, detailed information about the device status can be retrieved via the web browser and double clicking on the system LED, the PMX command set or the fieldbus.



25.2.1 Error in factory settings

No production data (serial no., prod.date 0). The device has not been tested at the HBM final inspection station. The system LED flashes yellow.
The device is nevertheless fully operational.

25.2.2 SYNC master

Status bit, no error. When set, the device is the sync master, i.e. no sync signal was detected at the Sync-IN socket.

25.2.3 SYNC error

Faulty or missing sync telegrams. Indicates connection problems at Sync-In socket.

25.2.4 SYNC controller error

The device cannot follow the master as a slave. The controller is saturated. The time stamps and the CF (carrier frequency amplifier) are not synchronous.

25.2.5 Heartbeat

System bit, no error. Flashing at approx. 1 Hz. If stopped there is a CPU error.

25.2.6 Sensor excitation voltage overloaded

The sensor supply was switched off due to overcurrent on at least one measurement card.

25.2.7 Buffer overflow in command interface

Measured values have been lost in catman® or the command interface due to buffer overflow.

25.2.8 System not ready

The device is damaged and is not delivering valid measured values.
Temporarily set when parameter set is switched (ok).
Statically set when the parameter set does not match the card configuration. Occurs when cards have been removed/added/swapped, or when an incompatible parameter set has been imported.

25.2.9 CPU overload during calculations

Calculation timeout in the calculated channels This can result in gaps in the measurement data stream.
Temporarily harmless when switching parameter sets, editing calculated channels or bridge inputs self-calibrating

When set during normal operation: Reduce number of function blocks and/or reduce global sample rate (38.4 kHz -> 19.2 kHz).

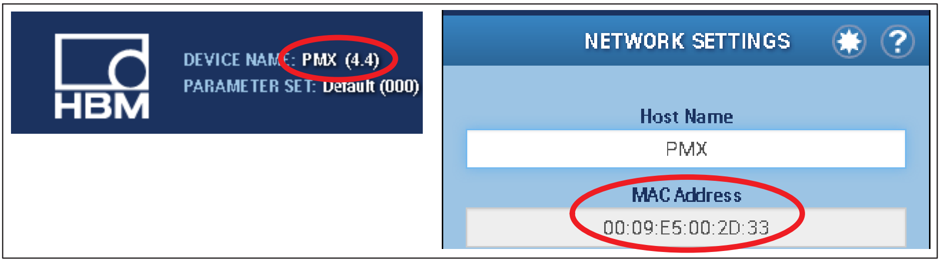
25.3 Resetting the PMX Administrator password

This procedure does not affect the Maintenance level password.

1. User

Send the PMX host name and MAC address to HBM's Technical Support Center (TSC) at: support@hbkworld.com.

You can find the host name on the "Overview" screen in the "Network" dialog. You can find the MAC address in the "Network" dialog and on the sticker on the bottom.



2. HBM Technical Support Center

HBM's Technical Support Center creates a file with a signature.

The signature is saved in a file named "pmxpasswordreset" and sent back to you.

3. User

Copy the "pmxpasswordreset" file to the root directory of a USB flash drive. Plug the flash drive into the USB port on the PMX while the PMX is running normally.

The Administrator password is immediately removed and the file is deleted from the USB flash drive.

If you want to keep a copy of the Reset file to be able to reset the password in the future, save the file in a secure location. You can reuse the file as long as you do not change the device's host name.

25.4 Resetting the PMX to factory settings

All device settings are reset in the **Settings -> System -> Device -> Device storage -> Restore factory settings** menu.

This function is not accessible at user level 1 (Operator).

Loading the factory settings deletes the following settings:

- All channel and amplifier settings (measurement channels and calculated channels, e.g. min./max. values)
- All device settings (e.g. parameter sets).

The following are not deleted:

- The network settings
- The passwords for the different user levels (Operator, Maintenance, Administrator)
- CODESYS applications and CODESYS web visualizations (with updates as from firmware 1.46)

25.5 Restoring lost PMX network settings and device names

If you cannot find the PMX in the network, you can use a USB flash drive to make the network settings you require.

1. Create a text file called "pmx.conf" in the root directory of a USB flash drive.

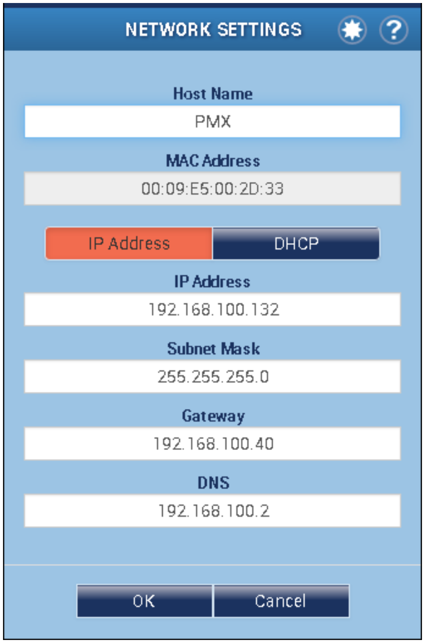
   Example 1:
   This pmx.conf file sets the device name to "pmx_new_name", and switches the PMX to DHCP mode
   ```
   <pmx type="set">
    <hostname>pmx_new_name</hostname>
    <network>
    <dhcp>true</dhcp>
    </network>
   </pmx>
   ```
   Example 2:
   Sets the name to "pmx", sets a fixed IP address:
   ```
   <pmx type="set">
    <hostname>pmx</hostname>
    <network>
    <ipaddress>192.168.1.2</ipaddress>
    <broadcast>192.168.255.255</broadcast>
    <netmask>255.255.0.0</netmask>
    <gateway>192.168.169.254</gateway>
    <dhcp>false</dhcp>
    </network>
   </pmx>
   ```
2. Insert the USB flash drive into the PMX *while it is in operation*.

   The settings will change immediately, but will not be immediately apparent in the other network devices. So it's best to restart the PMX by cutting the power.
   The PMX can then be found under the new settings in the network.

   Attention: The flash drive converts each PMX device as soon as you plug it in! So once you have used it you should delete the file, rename it, or move it to a subfolder.

Changing network settings

Network settings



25.6 Saving and restoring PMX device settings and CODESYS applications

If you want to transfer all device settings, parameter sets, network settings and also CODESYS applications from one PMX to another, you can use a USB flash drive. Passwords cannot be transferred in this way, and must be modified manually via the PMX browser (see also *section 10.3.2, page 134*).

Notice

Make sure that the configuration of both devices is the same, as otherwise the settings cannot be transferred and malfunctions may occur.

---

1. Create a text file in the root directory of a USB flash drive called: "pmx.conf". Depending on the content of this file, inserting a USB flash drive into the device can start a number of different actions, as described below:

   Example 1:
   Save all the PMX device settings with the network settings to the USB flash drive:
   <pmx type="save"> path="defaults.pmx" />

You can also create the parameter set file "defaults.pmx" by saving via the PMX browser with **Settings -> System -> Device -> Device storage -> Backup to PC**.

Example 2:
Load all the PMX device settings <u>without</u> the network settings from the USB flash drive:
<pmx type="load" path="defaults.pmx" />

Example 3:
Restore all the PMX device settings <u>with</u> the network settings from the USB flash drive:
<pmx type="restore">path="defaults.pmx" codesys="codesys.tgz"/>

2. The "codesys" attribute is an optional entry. It is only possible to create a "codesys.tgz" file with a WGX001 basic device with a CODESYS license. To do this, one or more applications must first be loaded into the device with the CODESYS IDE (development environment). Then you can save the "codesys.tgz" file under **Settings -> System -> Device -> CODESYS -> Backup to PC**. Copy it to the USB flash drive.
3. Insert the USB flash drive into the PMX while it is in operation. The settings will be transferred immediately.

Notice

Once inserted, this memory stick performs the function in each device! Once you have used it, you should therefore rename the file, delete it or move it to a different directory.

25.7 Replacing measurement and communication cards

Measurement and communication cards can be retrofitted or removed at a later date. Note the possible combinations (see *section 8.2.1, page 49*).

After modification, and switching on the supply voltage, the PMX automatically detects and initializes the hardware configuration.

Important

If measurement cards have been added, removed or mounted in other slots, the factory settings are loaded. You must then re-enter all parameters, including for the existing cards.
When communication cards (EtherCAT®, PROFINET® or EtherNet/IP™) are added, removed or swapped, the parameter settings are retained. This requires an adaptation to the new fieldbus using the control system's configuration tool.
If a CODESYS application or a CODESYS web visualization is running on the PMX, they are also retained after swapping cards. Please note that signal mapping in CODESYS is fixed, and if the measurement cards are moved the signal mapping must be checked and corrected as necessary.

When using the object dictionary, the object list also changes, and you must recreate the header files and adapt the programming via fieldbus or PC control.

25.8 Log file

To improve operational reliability, the PMX is equipped with an automatic log function. User inputs at all three user levels as well as all PMX (error) messages are logged and saved internally in the device.

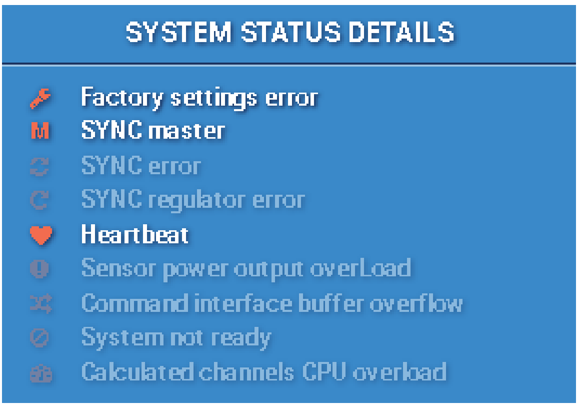
The device and channel states are also logged and saved.

This enables easy, unambiguous analysis in the event of errors. User level 1 (Operator) has no rights to delete the log file.

The file size can be set between 500 kB and 20 MB by user level 3 (Administrator). In parallel with the log entry, you also have the option to transfer the messages via network profile RCF5424 to a network server/PC, where they can be read with a standard text editor.

The log file can be retrieved via the icon (see below) or the user menu (**Settings -> System -> View log**).

25.8.1 System log entries for system status



- Sync master or slave:
  No sync input. Switch to master mode.
  Sync available. Switch to slave mode.

- Sync signal error
  Too many CRC errors at Sync input.
  Temporary switch to master mode.
- Sync controller error
  Coupling with incoming Sync signal not possible.
- Sensor supply overloaded:
  System status for sensor supply output overload "activated".
  System status for sensor supply output overload "deactivated".
- Buffer overflow in command interface:
  System status for command interface buffer overflow "activated".
  System status for command interface buffer overflow "deactivated".
- CPU overload on calculated channels
  Runtime overrun on calculated channels.

Information

The states for:

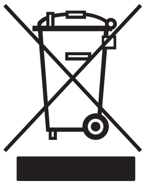
"Error in factory settings"
"Heartbeat"
"System not ready"

are not logged or saved.

25.8.2 System log entries for channel status/measurement status

- If the channel status changes from 0 to not 0, meaning at least 1 error is reset and there was no error before, the message: "Measval-status changed. New status: "invalid". Slot:X, Signal:Y" is generated.
  If there was already an error and a new one is added, no message is generated.
- If the measurement status contains no (more) error entries, i.e. changes to valid: "Measval-status changed. New status: "valid". Slot:X, Signal:Y"

All electrical and electronic products must be disposed of as hazardous waste. The correct disposal of old equipment prevents ecological damage and health hazards.



On the module

Statutory waste disposal marking

The electrical and electronic devices that bear this symbol are subject to the European waste electrical and electronic equipment directive 2002/96/EC.

The symbol indicates that the device must not be disposed of as household garbage.

In accordance with national and local environmental protection and material recovery and recycling regulations, old modules that can no longer be used must be disposed of separately and not with normal household garbage.

If you need more information about waste disposal, please contact your local authorities or the dealer from whom you purchased the product.

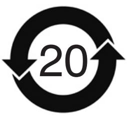
As waste disposal regulations within the EU may differ from country to country, we ask that you contact your supplier as necessary.

Packaging

The original packaging of HBM devices is made from recyclable material and can be sent for recycling. For ecological reasons, empty packaging should not be returned to us.

Environmental protection

The product will comply with general hazardous substances limits for at least 20 years, and will be ecologically safe to use during this period, as well as recyclable. This is documented by the following symbol.



On the module

Statutory marking of compliance with emission limits in electronic equipment supplied to China

27 FAQS

- Does the PMX have any fuses that need changing?
  No. The PMX has an internal current limiter to automatically regulate the power consumption in the event of a fault.
- Are there any moving parts that need to be maintained?
  No. The PMX has no fans or the like, and is maintenance free.
- Are the plugs protected to prevent connection errors?
  Not when shipped. But you can use the supplied coding pins to code the plugs and so protect them against connection errors.
- What plug options are available?
  The multipoint connectors are supplied as standard in the form of push-in terminals. You can also order them as screw-in terminals.
- What are the options for adjusting the measuring amplifiers?
  3 options:
  1. Enter sensor values (zero point/span) as numerical values
  2. Calibrate sensor values
  3. TEDS (Transducer Electronic Datasheet): Read sensor values from TEDS module into PMX amplifier and adjust automatically.
- What are the options for connecting the PMX to a web server?
  1. A direct 1:1 connection via Ethernet.
  2. An Ethernet connection via a network.
- Do I have to install operating software?
  No. The PMX has an internal web server for parameterization. All you need is a web browser, such as Windows Internet Explorer (min. version 9), Firefox or Google Chrome. Optionally, you can also use HBM's catman®Easy/AP software for recording and data analysis.
- What do I need to consider when connecting the PMX to a PC?
  The Ethernet cable must be plugged in. Both nodes (PMX [factory setting DHCP] and PC) must be set to DHCP. Connect by entering "PMX/" in your browser's address bar.
- Can problems occur if the RJ45 connectors for Ethernet, fieldbus and synchronization are inadvertently swapped?
  No. All connections are short-circuit proof. You can identify errors by the status LEDs on the device or in the PMX web browser.
- What do I need to consider when swapping plug-in cards?
  The PMX's power must be off! When you switch on, all the cards are automatically detected. The factory settings are loaded. All parameters must be re-entered, including for the existing cards. This does not affect the swapping of communication or fieldbus cards.

- How can I synchronize multiple PMXs?
  By connecting the SYNC sockets using standard Ethernet cables. The first PMX is automatically configured as the master, all the others are automatically slaves.
  A maximum of 20 PMX devices can be interconnected.
- How many measurement channels are available?
  A PMX can be fitted with a fieldbus card and max. 4 measurement cards.
  4 measurement channels are possible for each measurement card, i.e. a total of 16 measurement channels.
- How many calculation channels are available?
  There are always 32 calculation channels available in the basic device for each PMX. This allows you to implement a wide variety of control tasks in the PMX, from peak value calculations to PID controllers. The load on downstream systems and PLCs is relieved.
- How high are sample and processing rates in the PMX?
  All channels, measurement and calculation channels, are sampled and processed at 19200 Hz, or 38400 Hz in the case of the PX460. This means that extremely fast measurement data processing and automation is possible. For the measuring bandwidths refer to the technical data of the individual measurement cards.
- Are the measurement time stamps retained after a power failure?
  No, the time stamps reset to zero after restarting.
- How high is the resolution and accuracy of the PMX?
  The measurement channels have 24-bit resolution. This allows even very small signals in the partial-load range to be measured accurately and reliably. The accuracy class is a maximum of 0.05%.
- Can the channels of adjacent PMX devices be offset?
  No. Only the measurement and calculation channels of the individual PMX can be processed, not those of the other connected devices.
- How many parameter sets/measurement programs are there in the PMX, and how long are the switching times?
  A maximum of 100 parameter sets can be used in the PMX. These are divided into 4 sub-parameter sets, which can be switched separately.
  The switching time is between 0.1 and 2.5 seconds, depending on how many sub-parameter sets are being switched.
- Can the PMX also be used as a fieldbus master?
  On Ethernet-based fieldbuses (EtherCAT®, PROFINET® and EtherNet/IP™) the PMX can only be used as a slave.
  When using the CANopen interface under CODESYS Soft-PLC, you can run the PMX either as a CAN master or CAN slave.
- What happens if the power supply fails while a parameter set is being saved?
  The parameter set will be destroyed, and the PMX will restart with its factory settings after being switched on. To avoid this, we recommend backing up the device settings to a PC.

- What happens if the power supply fails during a firmware update?
  Either the device restarts with its old firmware after switching on, or it loads and initializes the new firmware, and is then ready for operation after 10 to 15 minutes.
- What happens with a CODESYS application after a power failure?
  If the CODESYS project was saved as a "boot project", the application will automatically start running again after the PMX is switched on.
- Can the source code of a CODESYS project be reloaded from the PMX?
  No, because the compilation generates machine code, which is loaded into the PMX This ensures the protection of know-how.
  However, you can also transfer the original source code to the PMX when writing the program. It can then be reloaded into the CODESYS development environment subsequently.
- What happens with a CODESYS application after a firmware update or when a measurement card is changed in the PMX?
  A CODESYS application or web visualization running on the PMX will also be retained when a card is changed or after a firmware update (as of firmware 1.46). Please note that signal mapping in CODESYS is fixed and if the measurement cards are moved or the PMX computing channels are changed, the signal mapping must be checked and corrected if necessary.
- Where can I find the latest firmware and device description files?
  You can download the current version of the firmware, including the PMX web server, and the device description files from https://www.hbm.com/de/2981/pmx-modular-measuring-amplifier-system-for-the-iot/.
- Is there an electrical design tool for PMX?
  Yes. Ready-made ePLAN macros, which you can use without a license, are available for PMX at https://www.hbm.com/de/2981/pmx-modular-measuring-amplifier-system-for-the-iot/.
- Are there 3D (STEP) files for mechanical design (CAE) for PMX?
  Yes. STEP files are available free of charge for PMX at https://www.hbm.com/de/2981/pmx-modular-measuring-amplifier-system-for-the-iot/.
- How do I get support if I have any questions?
  For technical questions, the HBM TSC (Technical Support Center) is available at support@hbkworld.com. If you have questions about technical project planning and design, our colleagues from Application Engineering at application-engineering@hbkworld.com will be happy to answer your questions or come to your site.

28 TECHNICAL SUPPORT

If you have any questions when working with the PMX data acquisition system, you can contact HBM's Technical Support:

Email support

support@hbkworld.com

Telephone support

Telephone support is available on all working days from 9 am to 5 pm (CET):
+49 6151 803-0

Extended support can be obtained through a maintenance contract.

The following options are also available:

Download software updates from HBM
https://www.hbm.com/de/2981/pmx-modular-measuring-amplifier-system-for-the-iot/

HBM on the Internet
https://www.hbm.com/contact/worldwide/

Headquarters worldwide

Europe
Hottinger Brüel & Kjaer GmbH
Im Tiefen See 45, 64293 Darmstadt, Germany
Tel. +49 6151 803-0
E-mail: info@hbm.com
www@hbm.com

North and South America
HBM, Inc., 19 Bartlett Street, Marlborough, MA 01752, USA
Tel. +1-800-578-4260 / +1-508-624-4500,
Fax +1-508-485-7480
E-mail: info@usa.com

Asia
Hottinger Baldwin Measurement (Suzhou) Co., Ltd.
106 Heng Shan Road, Suzhou 215009, Jiangsu, PR China
Tel. (+86) 512 68247776, Fax (+86) 512 68259343
E-mail: hbmchina@hbm.com.cn

APIPA RFC	*Automatic Private IP Addressing*, APIPA is designed to enable operation of a TCP/IP network without having to deal with IP addressing and parameters. Automatic IP addressing has been implemented in Microsoft Windows since Windows 98. It does not correspond fully to the IETF RFC however. Microsoft calls this process Automatic Private IP Addressing or APIPA.
Bonjour	*Bonjour* is a technology that provides automatic detection of network services in IP networks. It is an implementation of the Zeroconf system from Apple.
catman	Software package for data acquisition and processing: Measurement, analysis and evaluation of large measurement data sets, including mathematical and graphical functions (statistics, signal analysis, digital filters).
CAN bus	The **CAN** (**C**ontroller **A**rea **N**etwork) bus is a serial bus system, and is one of the fieldbus systems.
Cat-5-SFTP	Cat-5 screening. The S/FTP (Screened Foiled Twisted Pair) design is similar to FTP, but has an additional full screen (copper mesh screen) around the wire bundle.
CODESYS	CODESYS is a development environment for programmable logic controllers (PLCs) according to the IEC61131-3 standard for application development in industrial automation.
Communication card	The PMX basic device (WGX001/ WGX002) can be optionally equipped with a fieldbus communication card in slot 0. This allows you to implement the connection to a fieldbus master (PLC) via PROFINET®, EtherCAT® or EtherNet/IP™. This form of automation allows deterministic data transfer, meaning data is transferred at predefined time intervals.
Crossover cable	Al *crossover cable* in computer network (LAN) technology is an eight-core twisted-pair cable in which specific *cores* are swapped over in <u>one</u> of the two RJ45 plugs. While an uncrossed (*straight-through*) network cable connects a computer to switches, a crossover cable can directly interconnect two computers (or two switches). Most PCs toggle between direct and crossed automatically.

DHCP	The *Dynamic Host Configuration Protocol (DHCP)* enables the assignment of network configurations to clients via a server.
EtherCAT®	EtherCAT® ("Ethernet for Controller and Automation Technology") is real-time Ethernet initiated by the Beckhoff corporation. The protocol, disclosed in IEC standard IEC 61158, is suitable for both hard and soft real-time requirements in automation technology. The development of EtherCAT® focused on short cycle times (≤100 µs), low jitter for exact synchronization (≤1 µs), and low hardware cost.
Fieldbus	A *fieldbus* connects field devices in a plant, such as sensors and actuators, for the purpose of communicating with a control system. If several communication participants need to send their messages via the same line, it must be specified who (ID) says what (measured value, command) when (initiative). There are standardized protocols for this, such as CAN bus.
Greenline	HBM shielding design that ensures that HBM products can function safely without interference, and that no interference is emitted into the environment and supply networks are not overloaded.
GSDXML	The functionality of a PROFINET® IO device is always described in a GSD file. This file contains all relevant data essential for engineering and for data exchange with the IO device. The XML based PROFINET® makes IO devices writable. Following the international standards, the description language of the GSD file is GSDML (**G**eneric **S**tation **D**escription **M**arkup **L**anguage). As the name indicates, this is an XML (e**X**tensible **M**arkup **L**anguage) file that is language-neutral.
GUI status	Control word for transfer of data via the PMX web browser to a connected PLC (this function is currently not activated).
Host	The host name (also termed site name) is the unique identifier of a computer in a network. It is primarily used in electronic data exchange (e.g. e-mail, Usenet, FTP) to provide communication partners with a human-readable and comprehensible format.

Industrial Ethernet	Ethernet-based fieldbus protocols (for example PROFINET).® are referred to as Industrial Ethernet.
Input/output card	The PMX basic device (WGX001/WGX002) can be freely configured in slots 1-4 with measurement cards for acquiring measurement signals and output cards for outputting analog or digital signals.

Measurement card	The PMX basic device (WGX001/ WGX002) can be freely equipped with measurement cards for the acquisition of measurement signals in slots 1-4.

NETBIOS	*NetBIOS* (***Net****work* ***B****asic* ***I****nput* ***O****utput* ***S****ystem*) is a programming interface for communication between two programs over a network.

PLC	PLC is the abbreviation for Programmable Logic Controller. The PLC controls the functions of a machine and serves as an interface to the PMX.
PROFINET®	*PROFINET®* (***Pro****cess* ***Fi****eld* ***Net****work*)is the open Industrial Ethernet standard from *Profibus & Profinet International* (PI) for automation. PROFINET® uses TCP/IP and IT standards, is real-time Ethernet-capable, and enables the integration of fieldbus systems. PROFINET® is modular in design, so users can choose their own functionality. This differs mainly in the type of data exchange, in order to meet varying speed requirements.
PROFINET® IRT protocol	Isochronous data exchange with PROFINET® is defined in the Isochronous Real Time (IRT) concept. Data exchange cycles usually lie within the range of a few hundred microseconds up to a millisecond. The difference to real-time communication is essentially the high degree of determinism, so that the start of a bus cycle is maintained with high precision. The start of a bus cycle can deviate by up to a maximum of 1 µs. IRT is required, for example, in motion control applications (positioning processes).
Push-in technology	Easy wiring with no need for tools. This technology enables easy and direct connection of rigid and flexible wires with end ferrules

	of 0.34 mm^2 in size and more. A contact spring opens automatically, providing the necessary contact force.
Rail clip	Various items of electrical equipment (such as relays) can be pushed laterally onto a support rail with a U-shaped profile or pushed on from the front and locked in place. The support rail is also referred to as a DIN rail.
RFC2131	The Dynamic Host Configuration Protocol (DHCP) enables the assignment of network configurations to clients via a server. The Dynamic Host Configuration Protocol was defined in RFC 2131, and was assigned UDP ports 67 and 68 by the Internet Assigned Numbers Authority.
RJ45	RJ plug connectors are the plug connectors standardized by the US Federal Communications Commission (FCC) for telecommunication cabling. The connectors and sockets are available in various designs, shapes and with various numbers of contacts. The designation follows a format for categorization: The designation begins with the letter sequence *RJ*, followed by two numbers specifying the actual plug type. In networking, any fully wired eight-pin (8P8C) modular plug connector is often called "RJ45".
SIMATIC Manager	The SIMATIC Manager administers all data belonging to an automation project, irrespective of the target system (e.g. SIMATIC S7).
TCP-IP	*Transmission Control Protocol / Internet Protocol (TCP/IP)* is a family of network protocols, and is also referred to as the Internet protocol family because of its great importance for the Internet. The computers in the network are identified by IP addresses. A computer, or any device with an IP address, is referred to in TCP/IP terminology as a *host*. Originally, TCP was developed as a monolithic network protocol, but was subsequently split into the IP and TCP protocols. The core group of this protocol family is supplemented by the User Datagram Protocol (UDP) as an additional transport protocol. There are also numerous auxiliary and application protocols, such as DHCP and ARP.
TEDS	TEDS stands for "Transducer Electronic Datasheet", and indicates the electronic data sheet of a transducer or sensor which is stored in an electronic module and is inseparably linked

to the transducer.
It also records valuable metadata, such as calibration data, which is key information in terms of measurement and testing traceability. The electronic data sheet can be located in the transducer housing, or in the inseparable cable or connector plug.

Telnet	Telnet (*Telecommunication Network*) is the name of a widely used Internet network protocol. This long-established and well-known client/server protocol is based on character-orientated data exchange via a TCP connection. Programs that implement the function of the end device are often also called Telnet. Telnet comprises two services: Telnet Client and Telnet Server.
Time stamp	A time stamp is used to assign an unambiguous time to an event.
TwinCat	The Beckhoff *TwinCAT* software system converts virtually any compatible PC into a real-time controller with a multi-PLC system, NC axis control, programming environment and operator station. TwinCAT substitutes conventional PLC and NC/CNC control systems as well as operating units.
VG connector	The DIN standard **DIN 41612**, referred to as the *VG connector* standard, defines the design of plug connectors which are primarily used for multi-pin electrical connection of circuit boards in the low voltage sector. The connector pin number ranges from 20 to 160 pins.
Web server	A *web server* is a server that transfers documents to clients, such as web browsers. A web server refers either to a computer with web server software or merely the web server software itself. Web servers are used locally in company networks and primarily as a WWW service on the Internet. Documents can therefore be made available locally, within the company or worldwide according to the required purpose. The main task of a web server is the delivery of files, e.g. non-modifiable HTML or image files, or dynamically generated files, e.g. pages with content that is generated individually according to the profile of a logged in user.

INDEX

HBK - Hottinger Brüel & Kjaer
www.hbkworld.com
info@hbkworld.com

A04354 03 E00 03