Operating Manual

English

PMX

Hottinger Baldwin Messtechnik GmbH
Im Tiefen See 45
D-64239 Darmstadt
Tel. +49 6151 803-0
Fax +49 6151 803-9100
info@hbm.com
www.hbm.com

Mat.: 7-2002.4354
DVS: A4354-2.2 HBM: public
11.2016

1 Safety instructions

Appropriate use

The PMX amplifier system is to be used exclusively for measurement tasks and directly related control tasks. Use for any purpose other than the above is deemed to be non-designated use. In the interests of safety, the device should only be operated as described in the Operating Manuals. It is also essential to comply with the legal and safety requirements for the application concerned during use. The same applies to the use of accessories.

Each time, before starting up the equipment, you must first run a project planning and risk analysis that takes into account all the safety aspects of automation technology. This particularly concerns personal and machine protection.

Additional safety precautions must be taken in plants where malfunctions could cause major damage, loss of data or even personal injury. In the event of a fault, these precautions establish safe operating conditions.

This can be done, for example, by mechanical interlocking, error signaling, limit value switches, etc.

Notice

The device must not be directly connected to the DC voltage supply system. Supply voltage 10 V … 30 V (DC).

Before commissioning, ensure that a suitable supply voltage is used and that the circuit used is sufficiently protected.

The device may only be supplied with separated extra-low voltage (safety isolating transformer according to DIN VDE 0551 / EN60742). Only operate built-in devices once they are installed in the housing provided. The device development is based on DIN EN 61010 Part 1 (VDE 0411 Part 1).

General dangers of failing to follow the safety instructions

The PMX system is a state of the art unit and as such is reliable. The module may give rise to dangers if it is inappropriately installed and operated by untrained personnel.

Any person instructed to carry out installation, starting up, maintenance or repair of the module must have read and understood the Operating Manual and in particular the technical safety instructions.

Conditions at the place of installation

- Protect the device from direct contact with water.
- Protect the PMX system from moisture and humidity or weather conditions such as rain, snow, etc.
- Do not expose the device to direct sunlight.
- Please observe the permissible maximum ambient temperatures stated in the specifications.
- The permissible relative humidity at 31 °C is 95 % (non condensing); linear reduction to 50 % at 40 °C.
- Install the device so that it can be disconnected from the supply voltage at any time without difficulty.
- It is safe to operate the PMX system up to a height of 2000 m.

Maintenance and cleaning

The PMX system is maintenance-free.

- Before cleaning, disconnect all connections.
- Clean the housing with a soft, slightly damp (not wet!) cloth. You should *never* use solvents, since these could damage the labeling.
- When cleaning, ensure that no liquid gets into the module or connections.

Residual dangers

The scope of supply and performance of the PMX system only covers a small area of measurement technology. In addition, equipment planners, installers and operators should plan, implement and respond to the safety engineering considerations of measurement technology in such a way as to minimize remaining dangers. On-site regulations must be complied with at all times. There must be reference to the remaining dangers connected with measurement technology.

Product liability

In the following cases, the protection provided for the device may be adversely affected. Liability for device functionality then passes to the operator:

- The device is not used in accordance with the operating manual.
- The device is used outside the field of application described in this Chapter.
- The operator makes unauthorized changes to the device.

Warning signs and danger symbols

Important instructions for your safety are specifically identified. It is essential to follow these instructions in order to prevent accidents and damage to property.

Safety instructions are structured as follows:

WARNING

Type of danger

Consequences of non-compliance

Averting the danger

- **Warning sign:**
  draws your attention to the danger
- **Signal word:**
  indicates the severity of the danger (see table below)
- **Type of danger:**
  mentions the type or source of the danger
- **Consequences:**
  describes the consequences of non-compliance
- **Defense:**
  indicates how the danger can be avoided/bypassed

Danger class according to ANSI

Symbol	Significance
WARNING	This marking warns of a *potentially* dangerous situation in which failure to comply with safety requirements *can* result in death or serious physical injury.
CAUTION	This marking warns of a *potentially* dangerous situation in which failure to comply with safety requirements *can* result in slight or moderate physical injury.
Notice	This marking draws your attention to a situation in which failure to comply with safety requirements *can* lead to damage to property.

Working safely

Notice

The device must not be directly connected to the DC voltage supply system. Supply voltage 10 V … 30 V (DC).

The supply connection, as well as the signal and sense leads, must be installed in such a way that electromagnetic interference does not adversely affect module functionality (HBM recommendation: "Greenline shielding design", downloadable from the Internet at http://www.hbm.com/Greenline).

Automation equipment and devices must be installed in such a way that adequate protection or locking against unintentional actuation is provided (e.g. access checks, password protection, etc.).

When devices are working in a network, these networks must be designed in such a way that malfunctions in individual nodes can be detected and shut down.

Safety precautions must be taken both in terms of hardware and software, so that a line break or other interruptions to signal transmission, such as via the bus interfaces, do not cause undefined states or loss of data in the automation device.

Conversions and modifications

The device must not be modified from the design or safety engineering point of view except with our express agreement. Any modification shall exclude all liability on our part for any damage resulting therefrom.

In particular, any repair or soldering work on motherboards is prohibited. When exchanging complete modules, use only original parts from HBM. The product is delivered from the factory with a fixed hardware and software configuration. Changes can only be made within the possibilities documented in the operating manual.

Qualified personnel

Qualified personnel means persons entrusted with siting, mounting, starting up and operating the product, who possess the appropriate qualifications for their function (qualified electrician, or by someone with electrical training under the supervision of a qualified electrician).

This device is only to be installed and used by qualified personnel strictly in accordance with the specifications and with the safety rules and regulations which follow.

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

- Knowledge of the safety concepts of automation technology is a requirement and, as project personnel, you must be familiar with these concepts.

- As automation plant operating personnel, you have been instructed how to handle the machinery. You are familiar with the operation of the equipment and technologies described in this documentation.
- As commissioning engineers or service engineers, you have successfully completed the training to qualify you to repair the automation systems. You are also authorized to activate, ground and label circuits and equipment in accordance with safety engineering standards.

It is also essential to comply with the legal and safety requirements for the application concerned during use. The same applies to the use of accessories.

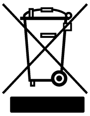
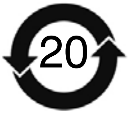
The PMX system must only be installed by qualified personnel, strictly in accordance with the specifications and with the safety requirements and regulations listed below.

Maintenance and repair work on an open device with the power on may only be carried out by trained personnel who are aware of the dangers involved.

Important

The safety instructions are also included in paper format with the product ("Documentation and Safety instructions PMX" A3260-2.0).

2 Symbols on the device

Symbol	Meaning
	**Statutory waste disposal mark** *See chapter 28, „Waste disposal and environmental protectionpage 539*
	Statutory mark of compliance with emission limits in electronic equipment supplied to China *See chapter 28, „Waste disposal and environmental protection“, page 539*
CE	**CE mark** The CE mark is used by the manufacturer to declare that the product complies with the requirements of the relevant EC directives (the Declaration of Conformity can be found at http://www.hbmdoc.com/en/industrial-amplifiers/pmx.html?geoip_cn=none
	Take details in the operating manual into account.
CODESYS	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 for your HBM products can be found at* http://www.hbm.com/en/menu/products/industrial-amplifiers/pmx/

3.1 Using this manual

- ► Read this operating manual thoroughly and in full before operating the device for the first time.
- ► Look upon this operating manual as part of the product and keep it so that it is accessible to all users, all of the time.
- ► 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 http://www.hbm.com/en/menu/products/industrial-amplifiers/pmx/

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

We can assume no liability for any damage arising from non-compliance with this operating manual.

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 a comprehensive 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 measuring amplifier system comprises

- This operating manual in PDF format
- A printed quick start guide for initial start-up
- A printed summary of the safety instructions
- The specifications (data sheet) in PDF format
- A PDF version of the web server Online help.
  The complete description of the functionalities and operation can be found in the Online help of the PMX web server.

Important

These documents can be found
• on the PMX system CD supplied with the device
• always up-to-date, on our website, at http://www.hbm.com/en/menu/products/industrial-amplifiers/pmx/

You can find additional information at http://www.hbm.com/en/menu/support/software-firmware-downloads/industrial-amplifiers/ regarding e.g. device description files for the realtime Ethernet cards (PROFINET / EtherCat) and configuration examples.

You can find additional information at http://www.hbm.-com/en/menu/products/industrial-amplifiers/pmx/ ,as well as a PMX video tutorial.

3.3 Symbols used in this manual

So that you can start working quickly and safely with your product, the symbols and terms used in this manual are standardized and are explained below (*see Chapter 28 Waste disposal and environmental protection*).

Symbol	Significance
•	List
-	List (second level)
► (blue)	Cross-reference to another point in this document or to other documents
►	This prompts you to take action (a single, independent action)
1. **2.** **3.**	Carry out this sequence of actions in the given order.
Important	**Important information** This draws your attention to important information about the product or about handling the product.
Tip	**Information/Application instructions** Practical tips or other useful information for you.

4 PMX product description

By buying the PMX measuring amplifier 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 / 38,400 measurements per second. The device thus achieves an overall processing rate of approx. 400,000 measured values per second.

Numerous different measurement, control and automation tasks can be resolved with this measurement system.

Connection to a PC (HOST)

The PMX measuring amplifier system is connected to a PC via the standard ETHERNET interface, and can be parameterized and operated via the internal web server.

The link to an automation system can be connected via digital and analog inputs/outputs, as well as over the fieldbus interfaces of the PMX, to a control (PLC) or a primary automation system.

Internal calculation channels

The PMX has 32 internal calculation channels as standard, which are freely available for evaluations and mathematical measurement signal calculations. Automation tasks from peak values to PID controllers can be implemented simply and elegantly.

The following types of plug-in card are available:

PX401

- The *PX401 measurement card* offers *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* is available for measurement with strain gages (SG) and also has *four channels* with 24-bit resolution and TEDS sensor detection.
- 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

- The frequency measurement card PX460 can be used to operate torque flanges (torque, speed, angle of rotation), angle/incremental encoders, SSI/PWM sensors and frequency measurements 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 flanges (T10, T12, T40) for torque or speed measurement (without direction of rotation detection)

- Or two measuring channels for simultaneous measurement of speed and angle (with direction of rotation detection)
- Or one measuring channel for simultaneous speed, angle and direction of rotation or reference pulse detection
- Or two angle/incremental encoders each, SSI/PWM sensors, magnetic transducers or pulse counters
- Or four measuring channels for frequency measurement up to 2 MHz, incl. 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 or be operated with a downstream controller (PLC). All real or calculated measurement signals can be freely assigned to the outputs.

PX01EC, PX01PN and PX01EP

- These interface cards can be optionally included in the PMX, enabling operation of the PMX in an automation system via the interface formats EtherCat®, PROFINET or EtherNet/IP. 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.

TEDS (Plug&Measure)

PMX amplifiers support TEDS (Transducer Electronic Data Sheet, IEEE1451.4). When transducers are connected, they are *automatically detected.* This efficiently minimizes setup times and user error.

PMX web server

An easy to operate web server that is specifically tailored to the PMX and is suitable for the measurement cards, is integrated into the device for configuration, data acquisition and visualization.

This gets you quickly to your measurement result, where you can visualize the measured data and view them at a later date.

PC software catman®EASY/AP

The HBM catman® software can be used to acquire, condition and analyze the PMX measurement data, as an option. This allows vast quantities of measurement data to be displayed (stripchart function), analyzed and exported in the most commonly used formats.

Software driver

The PMX command set and a dotNET-API are available to create customer-specific interfaces. This allows your own operating concepts and the inclusion in existing software solutions to be realized.

With the PMX-LabView driver, PMX can be integrated via Virtual Instruments (VI) in the software of National Instruments.

LabVIEW is a registered trademark 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, field-bus and analog outputs.

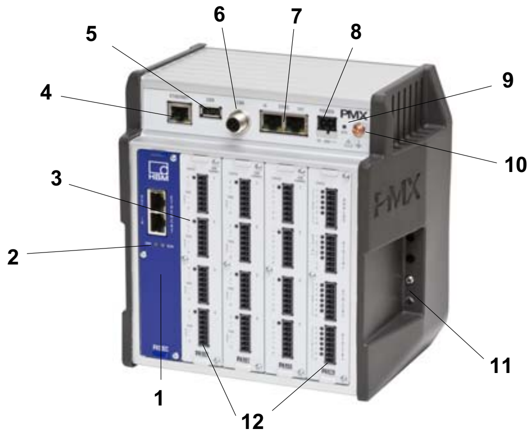
Calibration certificates

Documented quality: at the time of delivery, PDF documents of the HBM calibration certificates for the fitted measurement cards, and a certificate of compliance as per EN 10204 2.1 are already stored in the PMX device memory. The PMX browser can be used to download them.

5 Model overview, scope of supply, accessories

5.1 The PMX system

The PMX is a modular and universally applicable measuring amplifier system.



Nr	Bezeichnung
1	Communication card: EtherCAT® or PROFINET; EtherNet/IP
2	Fieldbus status LEDs
3	Measurement card status LED
4	RJ45-Ethernet socket to PC/Network
5	USB Host
6	CANbus (WGX001 only)
7	2x RJ45 socket for synchronization of up to 20 modules
8	10 … 30V voltage supply
9	System status LED
10	Ground
11	Positioning for the support rail
12	Max. 4 measurement card or input/output cards, e.g. : PX455, PX460, PX878, PX401

The PMX comprises

- Basic device
- Measurement cards
- I/O cards and
- Communication cards.

The measurement cards, input/output cards and communication cards can be individually combined and intelligently configured, in accordance with the measurement task.

The PMX system is a modular and universally applicable measuring amplifier system. The measurement cards, input/output cards and communication cards can be individually combined and intelligently configured in accordance with 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 CANBus nodes (WGX001 only)
SYNC	Synchronization of 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, always individually user-selectable between current and voltage input, TEDS (1-wire)
PX455	Strain gage amplifier	4 SG full or half bridges (CF). The bridge excitation voltage is 2.5 V Inductive full or half-bridges LVDT, potentiometric sensors, piezoresistive sensors, TEDS (0-wire)
PX460	Frequency/counter measuring amplifier	Up to four torque flanges (T10, T12, T40) for torque or speed measurement (without direction of rotation detection) Or two measuring channels for simultaneous measurement of speed and angle (with direction of rotation detection) Or one measuring channel for simultaneous speed, angle and direction of rotation or reference pulse detection Or two angle/incremental encoders each, SSI/PWM sensors, magnetic transducers or pulse counters Or four measuring channels for frequency measurement up to 2 MHz, incl. two shunt calibrations and two 1-Wire TEDS (sensor detection)

Input/output cards (I/O)

Basic device, model	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 module	EtherNET/IP communication adapter

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

Overview of measurement cards and input/output card

	base device		plugin board						
	WGX001	WGX002	PX401	PX455	PX460	PX878	PX01PN	PX01EC	PX01EP
Number of channels (total)	-	-	4	4	4	8/5/8	-	-	-
Data rate (Samples/s)	-	-	19200	19200	38400	19200	-	-	-
Bandwidth (Hz)	-	-	3000	2000	6000	3000	-	-	-
Full-bridge strain gages				•					
Half-bridge strain gages				•					
Inductive full-bridge				•					
Inductive half-bridge				•					
LVDT				•					
Potentiometric transducers				•					
Resistive 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 / rotary speed					•				
Incremental encoder					•				
Rot.angle w/ ref.impulse					•				
SSI encoder					•				
Inductive rotary encoder					•				
PWM					•				
EtherCAT								•	
PROFINET							•		
EtherNet/IP									•
CANopen	•								
	•								

1) For connecting current-fed piezoelectric transducers, a Smart Module (1-EICP-B-2) is required.
2) With 100 ohm completion resistor.

5.2 Scope of delivery

	Order No.
1 PMX basic device, with a wall mounting kit (1 wall bracket, 4 screws, 4 washers) and a set for DIN rail mounting with CAN connection and CODESYS-V3 Soft-PLC without CAN connection, without CODESYS	1-WGX001 1-WGX002
For each measurement card : A mating connector for each channel, all mating connectors with Push-In technology (4 connectors are enclosed for each measurement card, complete with coding pins)	1-CON-S1008 1-CON-S1012 for PX460
DIN rail mounting (2 units, packed in film cushion packaging with the mounting material in an Etimex bag) (4 M5x10 close tolerance screws, 4 spring washers)	1-RAILCLIP
PMX system CD with operating manual and data sheet, safety instructions and quick start guide	
For WGX001: Delivery with CODESYS-CD (CODESYS-V3 software, PMX package quick start guide and program examples)	
Mating connector M12x1 for CAN interface for WGX001	1-CON-S1002
Mating connector for PMX voltage supply (WGX001 / WGX002)	1-CON-S1010

5.3 Accessories

Accessories	Order No.
Ethernet crossover cable, for direction operation of devices on a PC or notebook, length 2 m, type CAT5+	1-KAB239-2
AC/DC power supply unit; Input: 90 V … 264 VAC, 1.5 m cable, output: 24 V DC, max. 1.25 A, 2 m cable with ODU plug	1-NTX001

Spare parts	Order No.
PX01, PMX blue blanking plate for plug-in card slot 0	1-PX01
PX02, PMX gray blanking plate for plug-in card slot 1-4	1-PX02
RAILCLIP, PMX DIN rail mounting set (2 pieces), 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 for PMX plug-in cards (4 x 7-pin, incl. coding plug and labeling sheets)	1-CON-S1009
Set of screw terminal for 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 x 13-pin and 2 x 2-pin, incl. coding plug and labeling sheets)	1-CON-S1012
Mating connector M12x1 for CAN interface for WGX001	1-CON-S1002

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

When ordering a PMX basic device, the delivery always includes DIN rail mounting and wall mounting elements.

Important

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

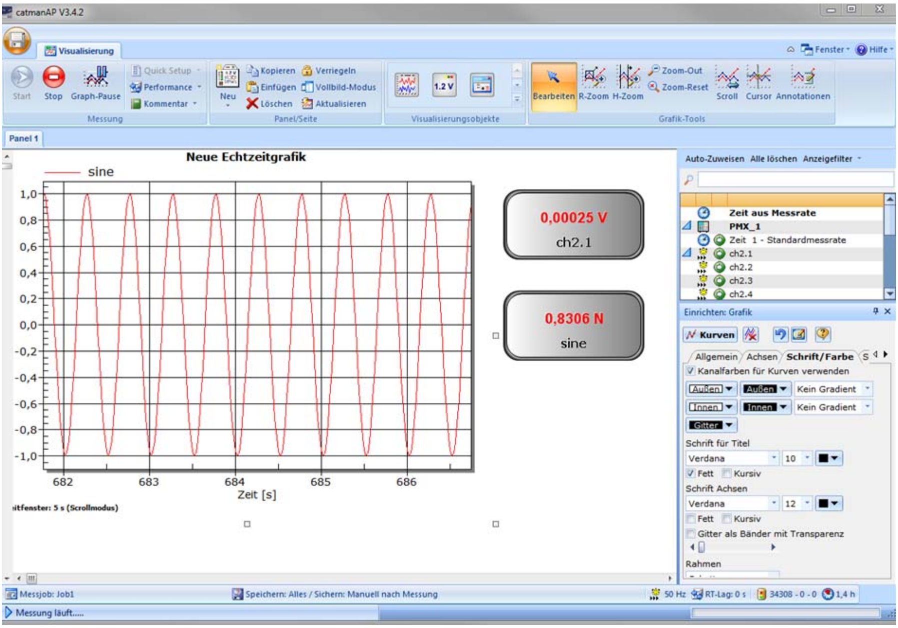
5.4 PMX web server/software

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

An online Help is integrated into the web server to support PMX operation and handling (click on the Help icon, top right in the overview menu).

PC software catman®EASY/AP

The HBM catman® software can be used to acquire, condition and analyze the PMX measurement data, as an option. This allows vast quantities of measurement data to be displayed (stripchart function) and analyzed (see following).



All real and calculated measurement channels, as well as digital inputs/outputs can be measured. Digital inputs/ outputs are displayed as binary coded values.

PMX supports up to three sampling rates that can be independently set. These sampling rates can then be assigned individual measurement signals.

Times of day, digital inputs/outputs of the PMX or triggering via limit values can be used to start/stop (trigger) a measurement.

The PMX can also be partially parameterized with catman®. This includes:

- Setting the sensor type via the sensor database or TEDS
- Writing the TEDS sensors via the integrated TEDS EDITOR in catman®
- Zero setting 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 saved in the active PMX parameter set by catman®. The sensor settings (sensor type, scaling and filter) are automatically changed by catman ® in PMX. This cannot be locked either in catman ® or in PMX.
Before starting catman® activate retaining of the PMX filter setting in the catman® option dialog (under DAQ channels: "Allow manual filter settings").
For firmware version 2.00 and later, catman®-version 4.0 or higher is mandatory.

Software driver

The PMX command set and a dotNET-API and a LabView driver are available to create customer-specific interfaces. This allows your own operating concepts and the inclusion in existing software solutions to be realized. With the PMX-LabView driver, PMX can be integrated via Virtual Instruments (VI) in the software of National Instruments.

The following functions are supported by the drivers in PMX:

Function	Description
Decice Scan	Scan Ethernet network
Measurement configuration	Set sample rate, filter, zero
Sensor configuration	Set scaling (2-point) or via TEDS
Analog In DAQ and calculated channels (streaming)	Read all measuring values and time stamps from sensors, channels
Status information (diagnostic)	Read each channel- and device status
Peak values	Read or clear peak values
Limit switches	Read or set limit switches
Analog Out (direct setting)	Read or set analog outputs (10 V)
Analog Out (configuration)	Set source, scaling
Digital In DAQ	Read digital outputs and set (high/low)
Digital Out DAQ (direct setting)	Read digital outputs and set (high/low)
CAN DAQ (via CODESYS / calculated channels)	Read calculated channels with CAN signals
Parameter sets	Read and select parameter sets

Notice

For firmware version 2.00 and later, catman®-version 4.0 or higher is mandatory.

Tip

All commands of the PMX command set can be used as low-level commands (see chapter 21) .
You can find extensive support and programming examples in the program help for the individual drivers:
All drivers and the catman® software as well can be downloaded from the HBM website as a free 30-day version: http://www.hbm.com/en/menu/support/software-firmware-downloads/daq-software/catman/

6 Degree of 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 present in the designation, are followed by two digits. These indicate which degree of protection a housing offers against contact or foreign bodies (first digit) and moisture (second digit).

All PMX modules and the basic device are designed with degree of protection IP20 (as per EN 60529).

IP 2 0

Code index	Degree of protection against contact and foreign bodies	Code index	Degree of protection against water
2	Protection against contact with fingers, protection against foreign bodies with ⌀▷ 12 mm	0	No water protection

New Greenline shielding design

To improve electromagnetic interference protection, HBM has developed an effective measure, the *Greenline* shielding design. The shield is connected to the connector housing. Appropriate routing of the cable shield means that the entire measuring chain is completely enclosed by a Faraday cage.

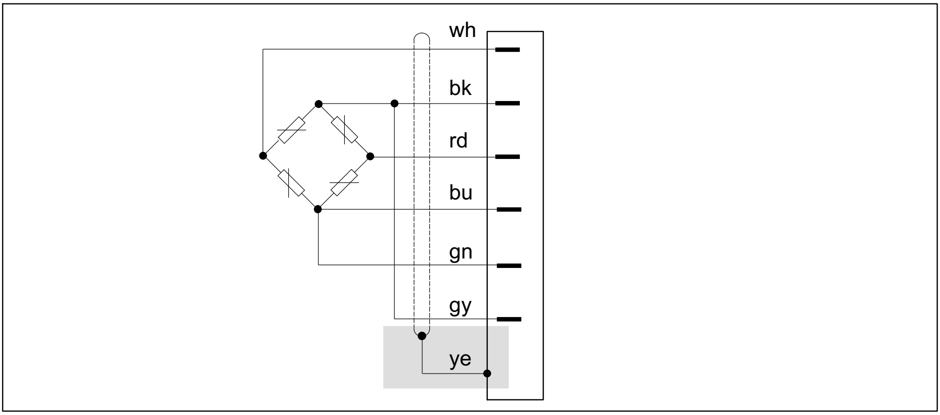
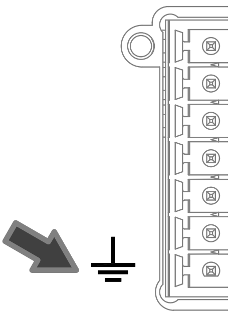


Fig. 6.1 Greenline shielding design



Important

Use standard HBM cables for connecting the transducers. When using other shielded, low-capacitance measurement cables, attach the shield of the transducer cable to the ground connection provided on the multipoint connector, in accordance with HBM Greenline information http://www.hbm.com/Greenline.
This results in greater EMC protection.

7 Mounting/Dismounting/Replacing

7.1 Mounting tools and tightening torques

Mounting	Required tool	Tightening torque
Fasten Rail-Clip to support rail M 2.5 hexagon socket screw	Hexagon socket screwdriver 2.5 a.f.	1.0 - 1.2 Nm
Mounting DIN rail on housing M5 hexagon socket screw	Hexagon socket screwdriver 3 a.f.	3 Nm
Mounting front panel M2.5 Torx screws	Torx screwdriver TX8	0.5 - 0.6 Nm
Mounting wall bracket M4 hexagon socket screw	Hexagon socket screwdriver 3 a.f.	1.5 … 2 Nm
Mounting side parts 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

7.2 Support rail mounting

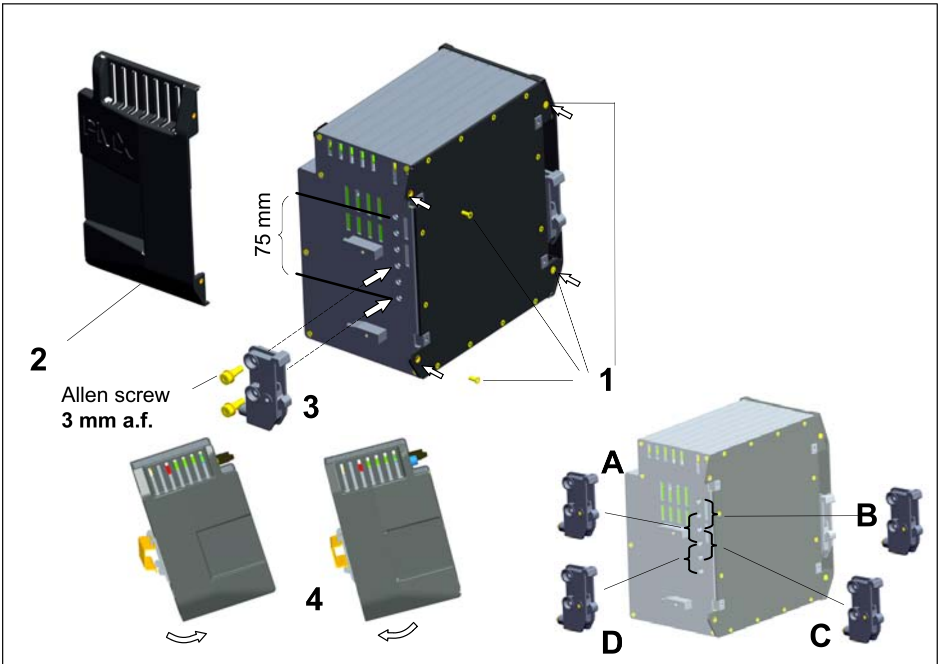


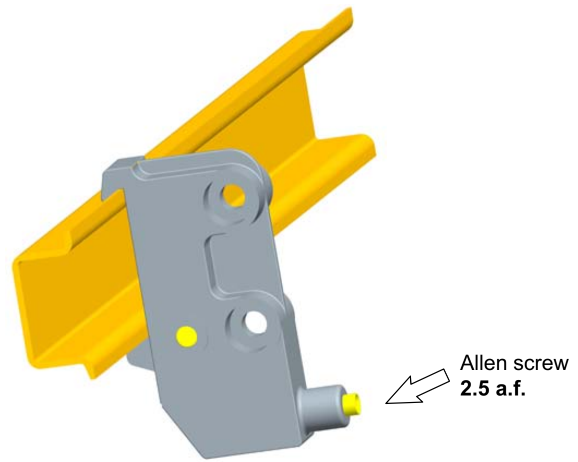
Fig. 7.1 Mounting on a support rail

- Undo the four back-panel screws (Torx T10) **(1)**
- Slide the side parts forward **(2)**
- Screw on the support rail mounting (Rail-Clip) **(3)** (approx. 5 Nm) in a choice of four positions (**A** to **D**) (two positions for the 7.5 mm support rail)
- Screw the side panels **(2)** back on
- Hook the PMX device into the support rail **(4)**

Important

Device damage caused by the PMX falling due to resistance in hooking/unhooking the PMX.
HBM recommends using a DIN support rail (DIN EN 60715), 15 mm in height. When using a smaller support rail (7.5 mm high), it should be shimmed to make it easy to hook/unhook the PMX device.
The 7.5 mm support rail can only be used in the top two positions (A and B).

Attach the support rail mounting (Rail-Clip) to the DIN rail



At the time of delivery, the self-locking (2.5 mm a.f.) Allen screws are *unscrewed* as far as the stop.

► Clip on the support rail mounting (Rail-Clip)

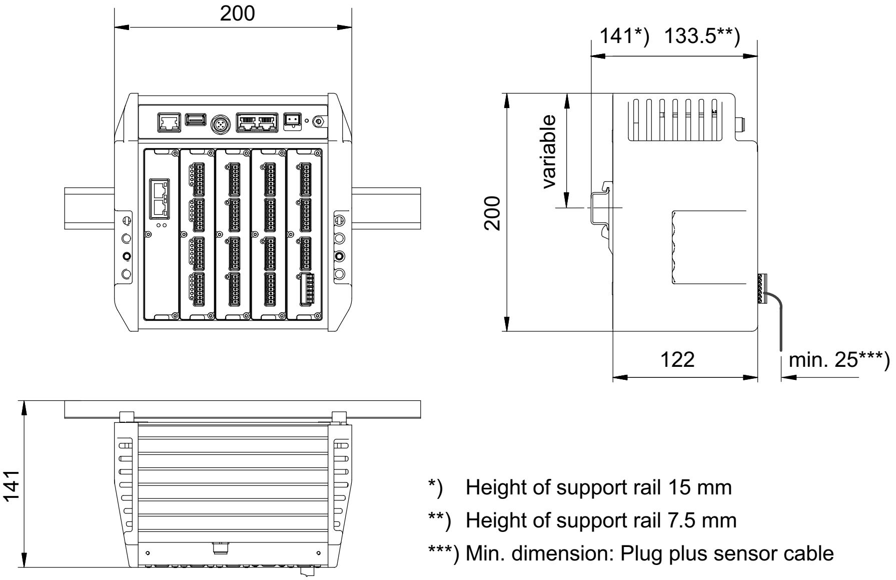
► Hand-tighten the self-locking Allen screw (1 - 1.2 Nm)

Important

Device damage caused by electromagnetic irradiation of external devices.
Faulty measurements due to electromagnetic irradiation from other devices.
To ensure adequate grounding for the PMX, the support rail must be connected to a functional earth ⏚ .
Both the DIN rail and the PMX must be free of paint, varnish and dirt at the point of installation.

► *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.

7.3 Wall-mounting

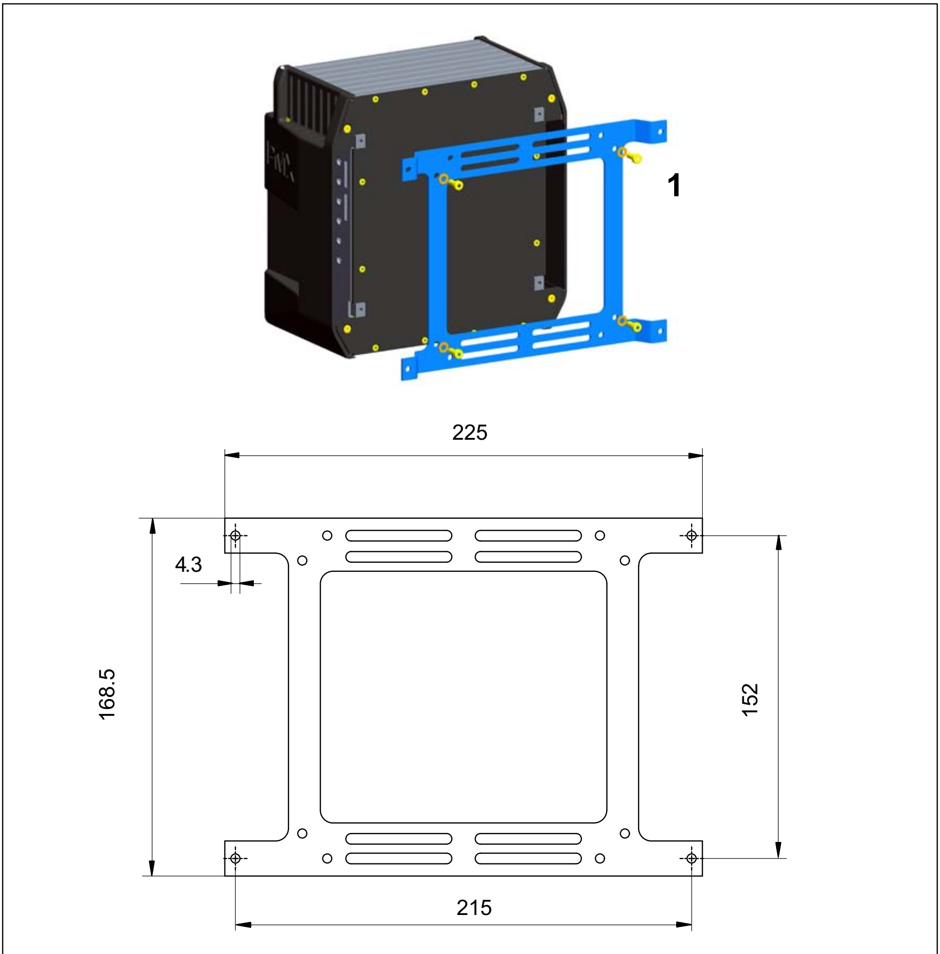
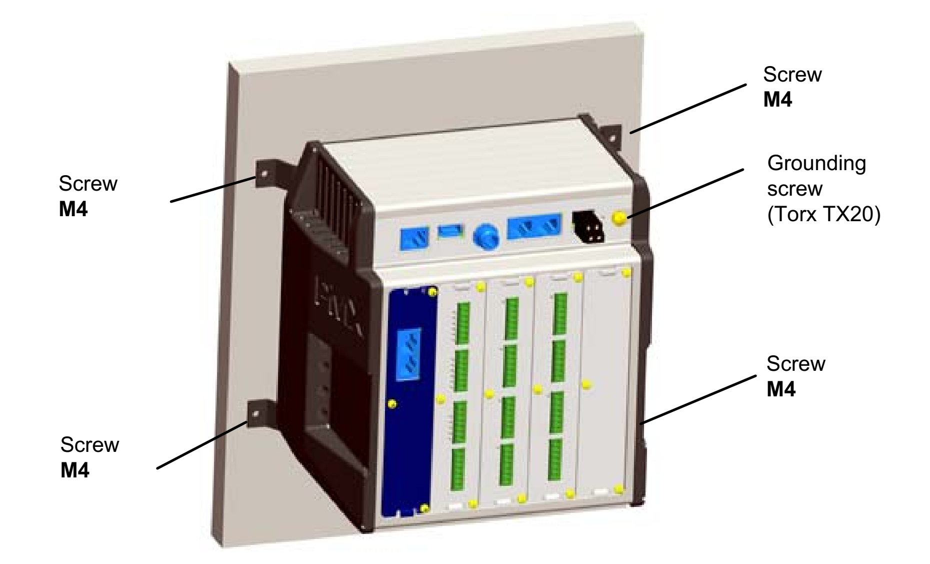


Fig. 7.2 Mounting on a wall

► Attach the wall bracket at the back of the PMX with the enclosed M4 screws **(1)** (3 Nm)



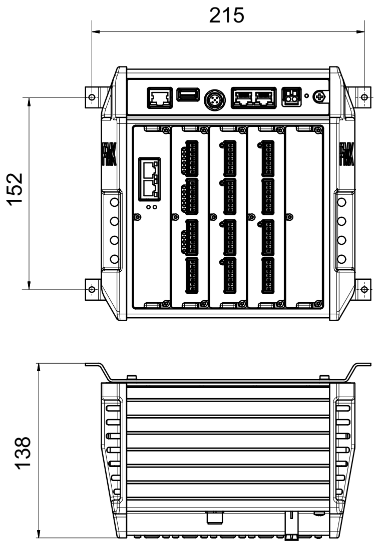
► Screw the complete unit to a wall; hole Ø 4 mm

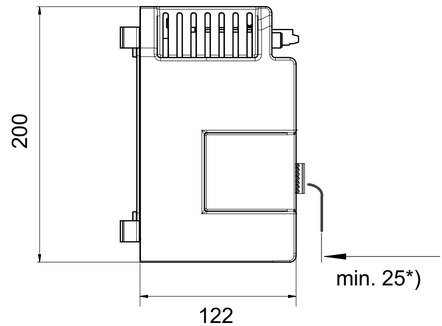
Notice

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 a functional earth ⏚ in a wall mounting.

► *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.

7.4 Replacing measurement and communication cards

Measurement and communication cards can be retrofitted or removed at a later date. Please note the combination options *(see chapter 8.2.1 on page 55)*.

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 reentered, also for the existing cards.

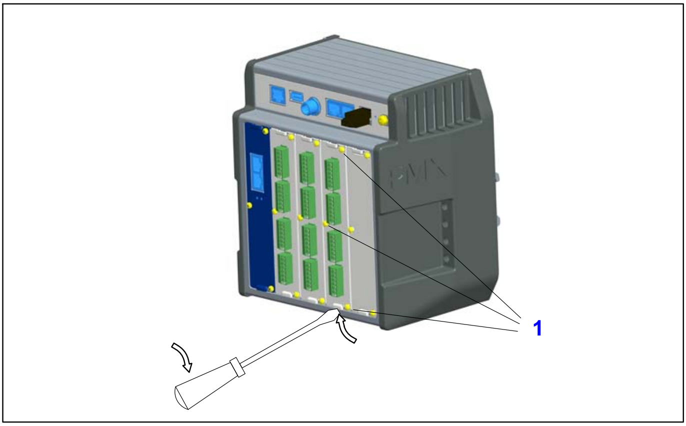
Notice

If the measurement or communication cards are incorrectly removed/replaced, they may be damaged or destroyed.
These cards must be ***de-energized*** *before they are removed or replaced*

► *Always disconnect the PMX from the power supply before removing a card.* Note that the device settings have to be re-parameterized if new cards are added.

The instructions below must also be followed:

Removal



1. Undo the three M2.5x8 Torx (Tx8) screws (**1**) of the card/blanking plate
2. Use a screwdriver to slightly lever up 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 strip at the back
3. Re-tighten 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 the open module slots with blanking plates (Accessories).*

8 Electrical connections PMX

8.1 Plug technology and clamping areas

All PMX plug-in cards (PX401, PX455, PX460, PX878) are delivered as standard with easy to fit plug terminals using Push-In technology. Plug terminals can also be delivered with a screw-on system as an option.

Push-In technology	

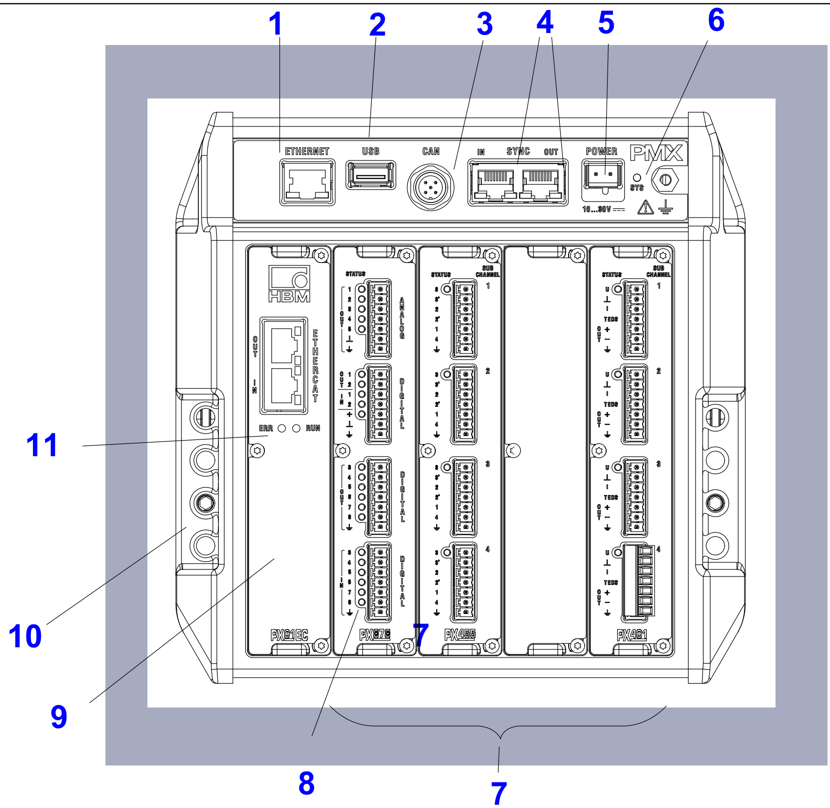
The clamping area is 0.2mm2 (AWG24) to 1.5mm2 (AWG16). If several conductors are to be connected to a terminal, the line cross-section must be adapted accordingly. End sleeves (without plastic collars, 10mm) should be used on the strands to connect the wires to the terminals.

The shield of the transducer cable must be connected to the ground connection provided on the PMX multipoint connector, in accordance with HBM Greenline information http://www.hbm.com/Greenline.

Important

The grounding terminal on the PMX is not a protective ground (optional connection).
The measurement system is fitted with an automatic current limiter for each device card, as well as for the PMX basic device.

8.2 Overview of PMX functions

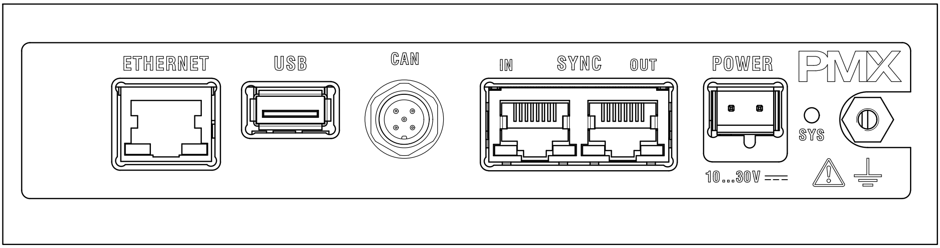


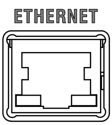
1	Ethernet plug for PC/network connection	**7**	Measurement cards (PX401, PX455, PX460) and/or PX878 I/O card or blanking plate
2	USB host, e.g. for a memory stick	**8**	LED Measurement card status
3	CAN for CAN driver, M12, option (WGX001)	**9**	Communication cards: PX01EC (EtherCAT®, PX01PN (PROFINET-IO, PC01EP EtherNet/IP) *or* blanking plate
4	2 x RJ45 for synchronization		
5	Excitation 10 . . . 30 V DC	**10**	DIN rail positioning
6	Device status - LED	**11**	Fieldbus LED

8.2.1 Combination options

	Slot 0	Slot 1	Slot 2	Slot 3	Slot 4	Number of plug-ins
Fieldbus or blanking 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

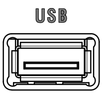
8.2.2 Relevance of the basic device connection sockets





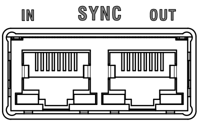
PC or network connection.

Cable: Ethernet cable CAT5, SFTP

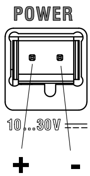


USB connection version 2.0 e.g. for mass storage device, scanner, USB stick

Cable: standard USB cable

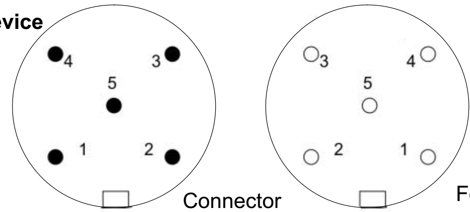


Synchronization of several (max. 20) PMX devices via two RJ45 sockets.
See chapter 8.1

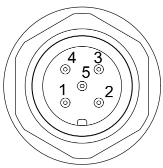


(view of device front)

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



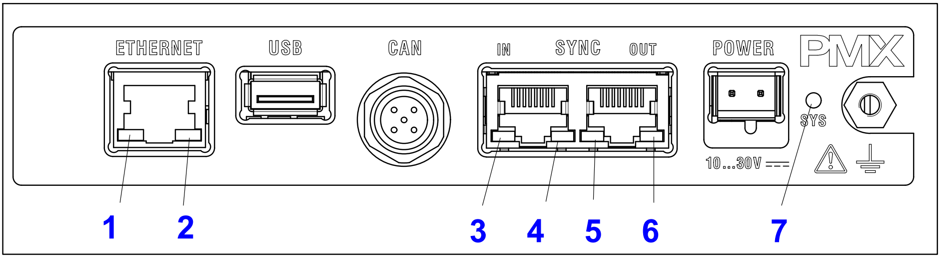
CAN connection (type WGX001 only)xx



Pin	Signal	Description
1	SHLD	CAN shielding
2		not connected
3	GND	Ground
4	CAN_H	CAN_H data lead (high)
5	CAN_L	CAN_L data lead (low)

8.2.3 LEDs for system monitoring (device LED)

Basic device (WGX001/002)



ETHERNET LED (1, 2)

LED	LED	Status	Significance
Ethernet Link (1)	green	Steady	Connection is present
Ethernet RX / TX (2)	yellow	Flashing	Data are being transmitted

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

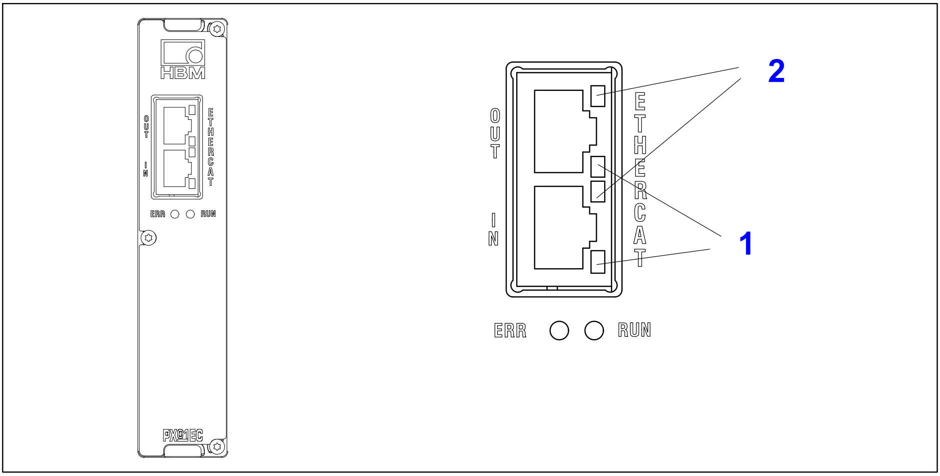
LED	LED	Status	Significance
IN (3)	green	On	Slave
IN (4)	yellow	On	Error
IN (3 + 4)		Off	Master
OUT (5)	green	On	always on
OUT (6)	yellow	On	Error (always identical to the right-hand LED of the IN socket)

SYS LED (7)

LED	Status	Significance
green	On Off	Voltage supply available Voltage supply missing
yellow	On	Device is booting
red	Flashing On	Serious internal error or Firmware update

8.2.4 Fieldbus LED

PX01EC



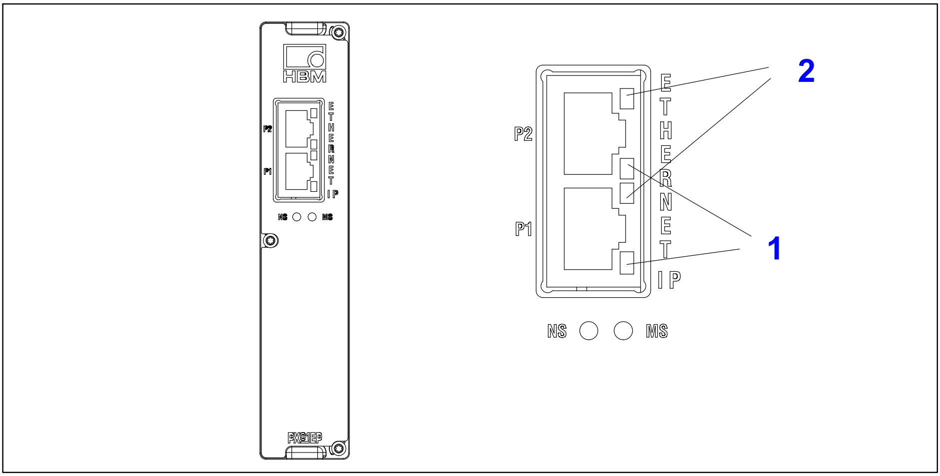
EtherCAT®

LED	LED	Status	Significance
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	Significance
RUN	green	Off	INIT status
	green	Flashing	PRE OPERATIONAL status
	green	Single flash	SAFE OPERATIONAL status
	green	On	OPERATIONAL

LED	LED	Status	Significance
1	green	Permanently on Flashing Off	Connection established Send / Receive No connection
2	-	-	No function

PX01EP



EtherNET/IP

LED	LED	Status	Significance
LV	green	On	**Connected**: If the device has at least one existing connection (even to the message router), the network status display will light up and remain green.
	green	Flashing	**No connections**: If the device has no existing connections, but has received an IP address, the network status display flashes green.
	red	On	**Double IP***: If the device detects that the IP address has already been used, the network status display lights up and remains red.

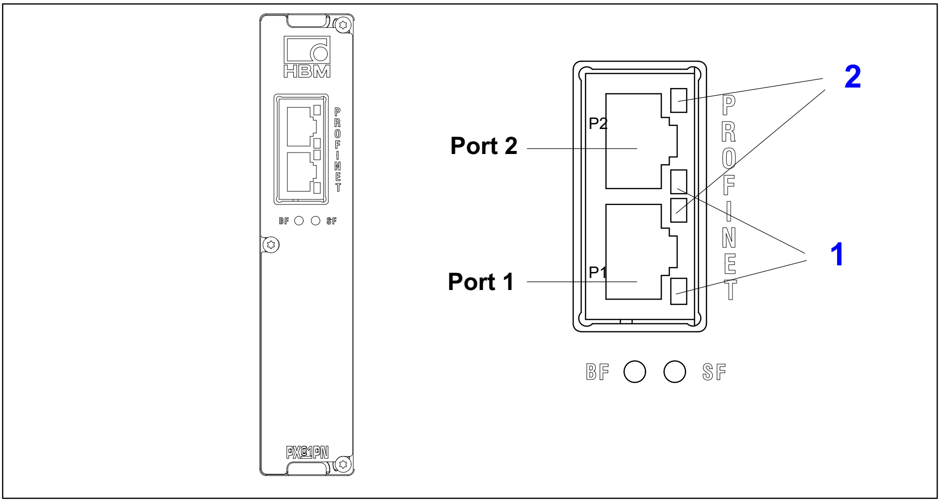
LED	LED	Status	Significance
	red	Flashing	**Connection time-out**: If one or more of the connections to this device are in time-out, the network status display flashes red. This status is only terminated when all connections in time-out have been restored or when the device is reset.
	red green	Flashing	**Self-test**: The network status display flashes green/red when the device is carrying out a self-test.
	-	Off	**Not switched on, no IP address**: If the device has no IP address (or is switched off), the network status display does not light up.

LED	LED	Status	Significance
MS	green	On	**Device operational**: If in operation and running correctly, the network status display will light up and remain green.
	green	Flashing	**Standby**: If the device has not been configured, the module status display flashes green.
	red	Flashing	**Serious error**: If the device detects a serious error that cannot be rectified, the module status display lights up and remains red.
	red	Flashing	**Simple error***: If the device detects a serious error that cannot be rectified, the module status display lights up and remains red. NOTE: A faulty or inconsistent configuration is classified, e.g. as a simple error.

LED	LED	Status	Significance
	red green	Flashing	**Self-test**: The module status display flashes green/red when the device is carrying out a self-test.
	-	Off	**Not switched on**: If the device is not switched on, the module status display does not light up.

LED	LED	Status	Significance
1	green	On	Connection to Ethernet established
	-	Off	The device has no connection to the Ethernet
2	yellow	Flashing	The device is sending/receiving Ethernet frames

PX01PN



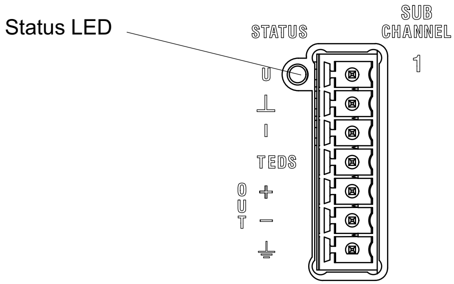
PROFINET

LED	LED	Status	Significance
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	Significance
1	green	Permanently on Flashing Off	Connection established Send / Receive No connection
2	-	-	No function

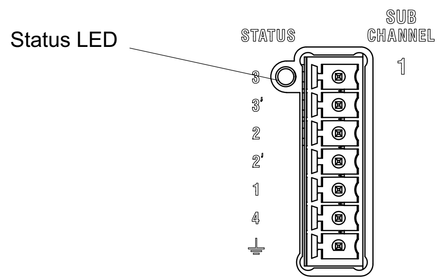
8.2.5 Measurement card LEDs

PX401, channel status



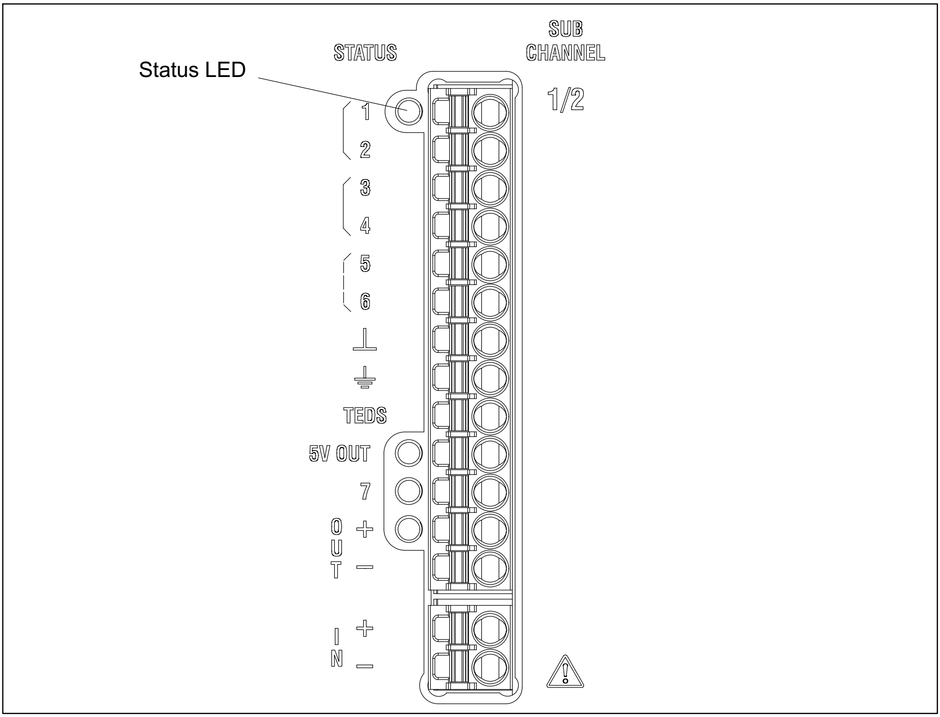
LED	Status	Significance
green	On	no errors
yellow	Flashing	Firmware update
red	On	Parameter not OK, overloaded

PX455, channel status



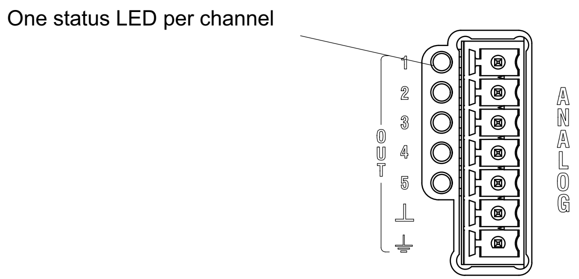
LED	Status	Significance
green	On	no errors
yellow	On Flashing	No transducer connected or wire break (self-calibration ongoing) Firmware update
red	On	Parameter not OK, transducer error, overloaded

PX460, channel status



LED	Status	Significance
green	On	no errors
yellow	On Flashing	No transducer connected or wire break or firmware update
red	On	Parameter not OK, transducer error, overloaded

PX878



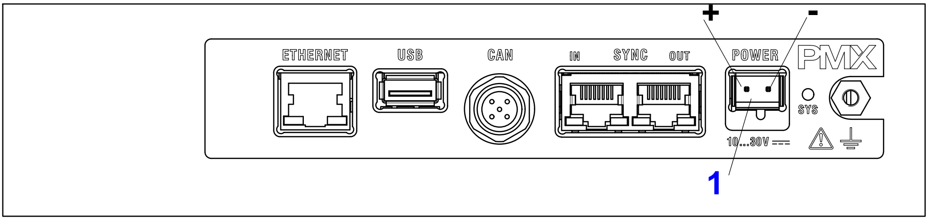
LED	Status	Significance
Digital		
green	On Off	Digital output: High Digital output: Low
green	On Off	Digital input: High Digital input: Low
Analog		
green	On Off	Analog output configured Analog output not configured
red	On	Analog output overloaded, signal invalid

8.3 Supply voltage

Notice

Device damage caused by high voltages.
If you are using power supply 1-NTX001 listed in the accessories, note the enclosed 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 Startup)*.



Measurement card	Power consumption [W] for a 24 V supply voltage
Basic device	3
PX401	0,75
PX455	1,6
PX460	2
PX878	2
PX01EC (EtherCAT®)	2
PX01PN (PROFINET)	2,4
PX01EP (EtherNet/IP)	2,3

8.4 Measurement cards / transducer connection

8.4.1 Intrinsically safe measurement circuits - Operation with Zener barriers

To operate transducers (load cells, force transducers, etc.) in potentially explosive atmospheres, intrinsically safe measurement circuits (Ex II (1) GD, [EEx ia]IIC) must be set up on the PX455 by connecting safety barriers (Zener barriers) type SD01A. The safety barriers must also be mounted on the DIN rail like the PMX. An ATEX test certificate must be available for the transducers used. Transducers with 350 ohm bridge resistance can be used. One transducer per measurement channel of the PC455 can be operated (parallel connection is not possible).

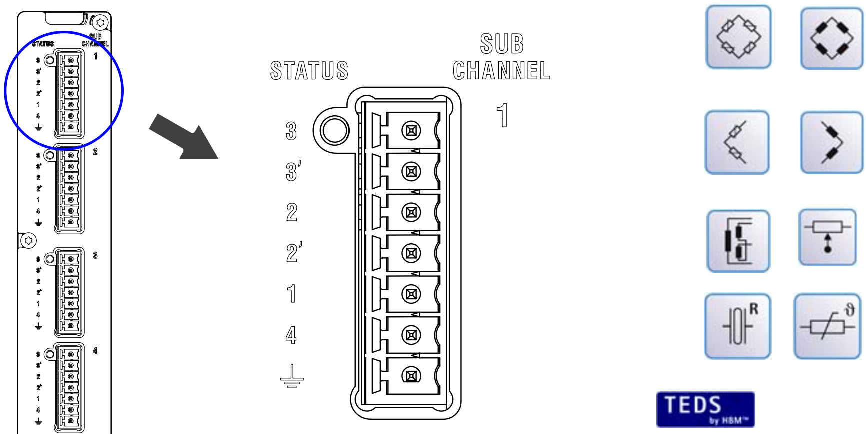
The PX455 offers 4 measurement channels with 4.8kHz carrier frequency measurement. If the measurement signals need to be added/subtracted or mean values formed, this can be set using the internal PMX calculation channels.

Important

In addition to the SD01A, the negative operating voltage of the PMX must be grounded!
Cable lengths of up to max. 100 m are permissible. TEDS cannot be used.
The measurement accuracy of the PX455 with operation of the SD01A is max. 0.5%.

8.4.2 PX455

Four individually configurable 4 SG full or half bridges (4.8kHz CF). Inductive full or half bridges, LVDT, potentiometric sensors, piezoresistive sensors, TEDS (0-wire), sensor detection



0-wire TEDS
see Chapter 8.7.3

The bridge excitation voltage is 2.5 V.

8.4.2.1 Strain gage and inductive full and half bridges (6-wire configuration)

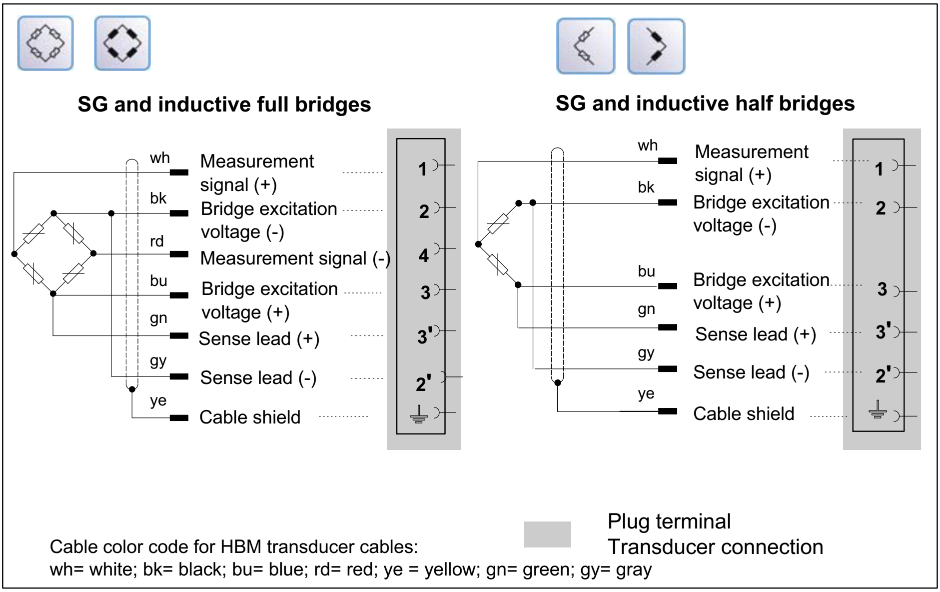


Fig. 8.1 PX455 plug terminal in a six-wire circuit

8.4.2.2 Strain gage and inductive full and half bridges (4-wire configuration)

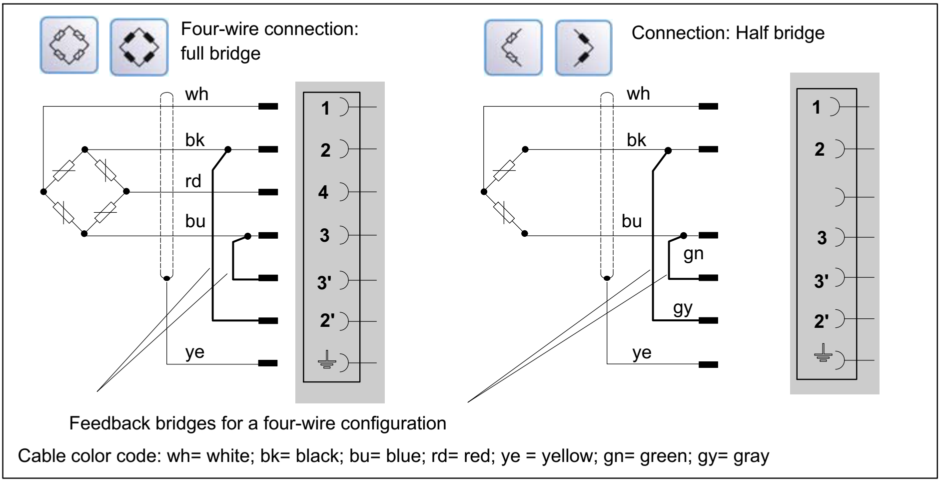


Fig. 8.2 PX455 plug terminal in four/three-wire configuration

8.4.2.3 LVDT

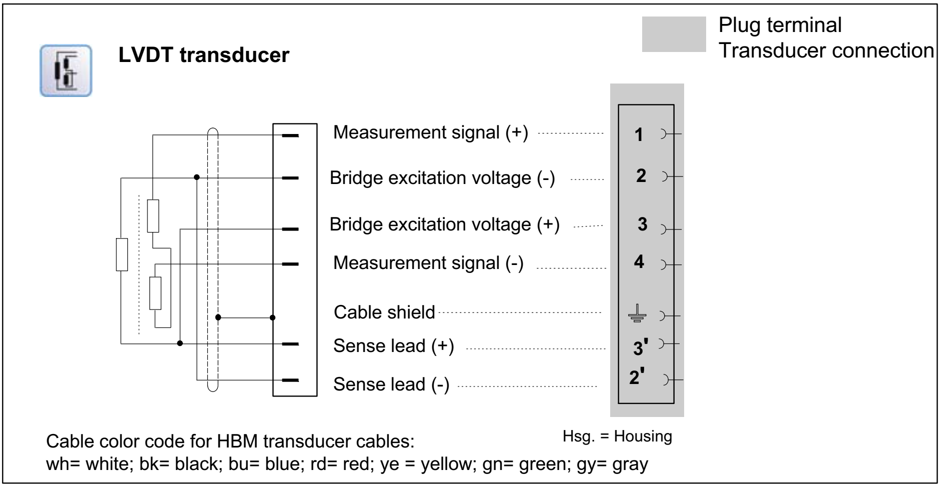


Fig. 8.3 PX455 plug terminal LVDT transducers

8.4.2.4 Potentiometric transducer

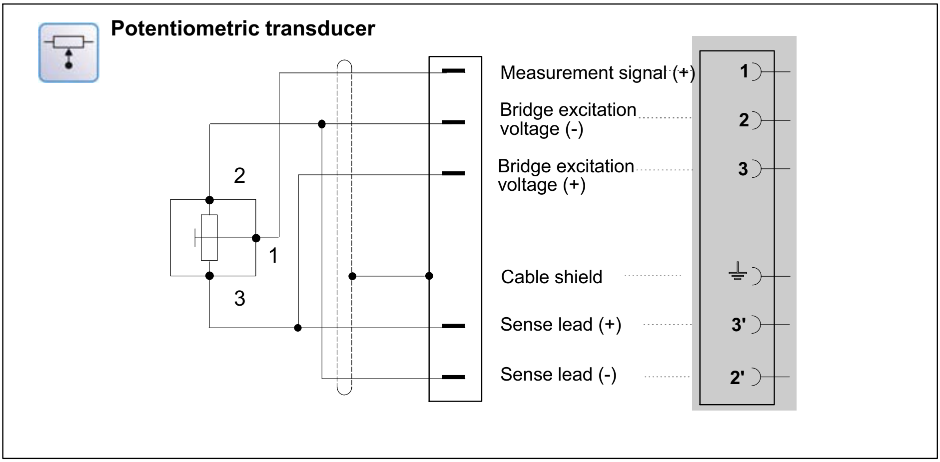


Fig. 8.4 PX455 plug terminal potentiometric transducer

Important

Transducer connection in a four/three wire configuration:

When connecting a transducer in a four/three wire configuration, the sense leads must be connected to the corresponding bridge excitation circuit (PIN 2'-2 and Pin 3'-3) by wire bridges, as otherwise a sensor error will be detected.

TEDS functionality is not available when connecting in a four-wire configuration.

8.4.2.5 PX455 with Pt100 temperature measurement

With the PX455 measurement module, temperatures can be measured with an uncertainty of +/-1°C without an external preamplifier. A 100 ohm precision resistor (R_comp) with a maximum tolerance of 0.1% is added to a Pt100 resistor to make a half bridge circuit. The PMX calculated channel "Pt100 on PX455" converts the measured bridge unbalance to degrees Celsius and performs a corrective calculation appropriate to the sensor cable being used (R_wire).

Important

Good heat dissipation is essential 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.

Resistance thermometer

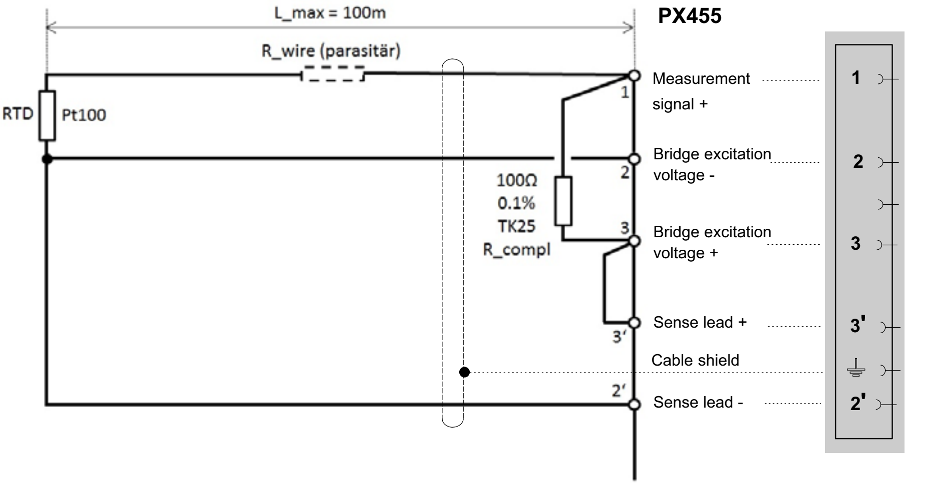
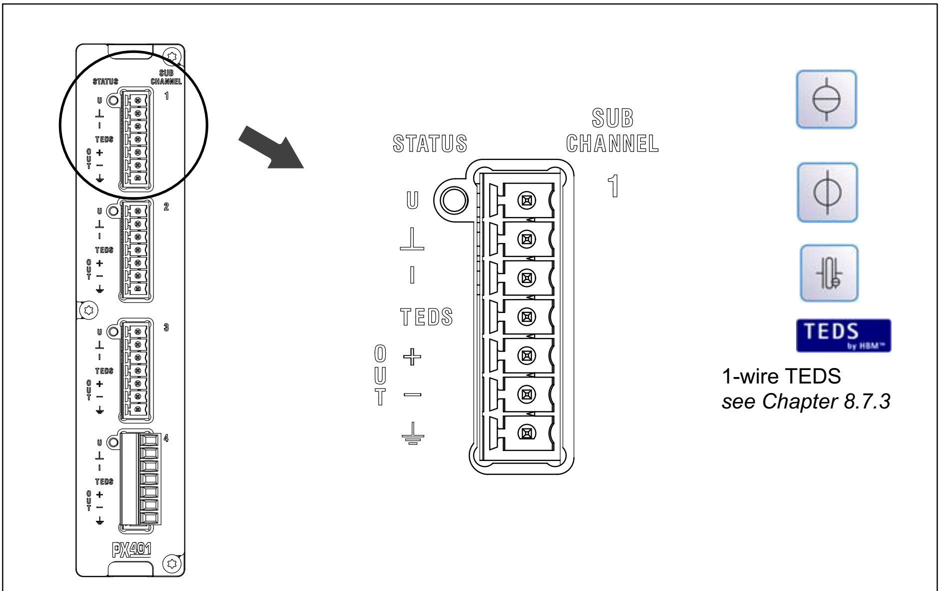


Fig. 8.5 PX455 with PT100 element for temperature measurement

8.4.3 PX401

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



IEPE sensors can also be operated in combination with the Smart module (1-EICP-B-2).

8.4.3.1 Voltage source ± 10 V

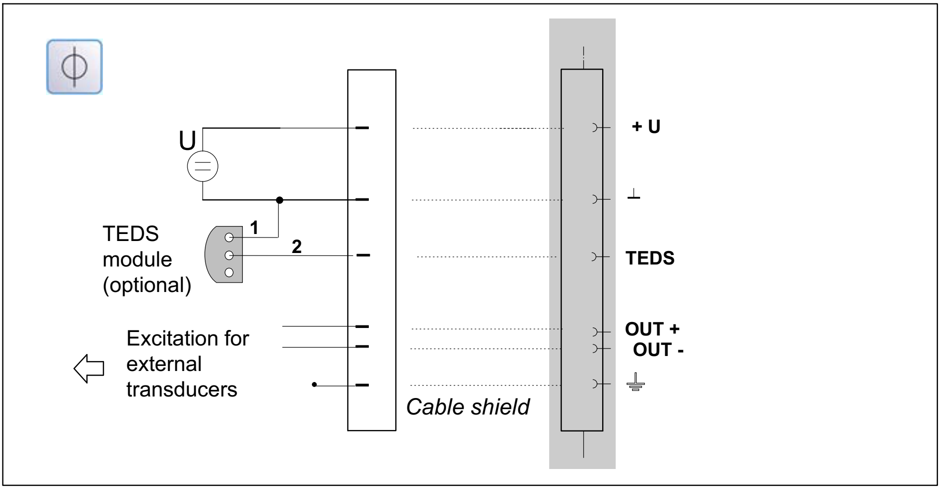


Fig. 8.6 PX401 plug terminal Voltage source ±10 V

8.4.3.2 Current source ± 20 mA

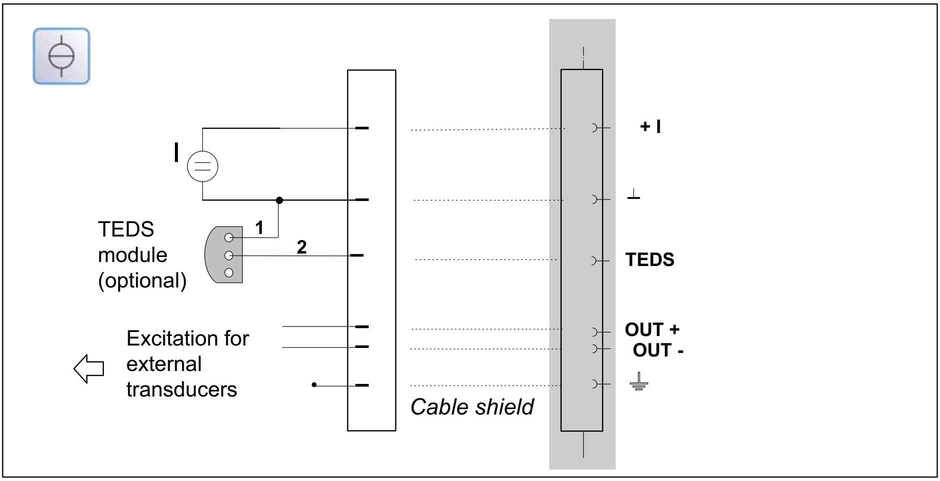


Fig. 8.7 PX401 plug terminal Current source ±20 mA (4-wire circuit)

8.4.3.3 Current source ± 20 mA

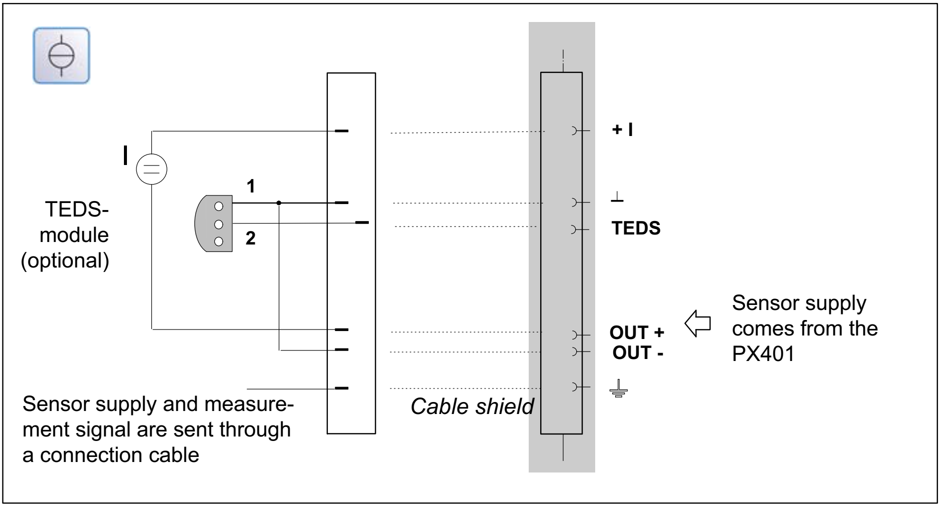


Fig. 8.8 PX401 plug terminal Current source ±20 mA (2-wire circuit)

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

8.4.3.4 IEPE transducers with external measuring amplifier

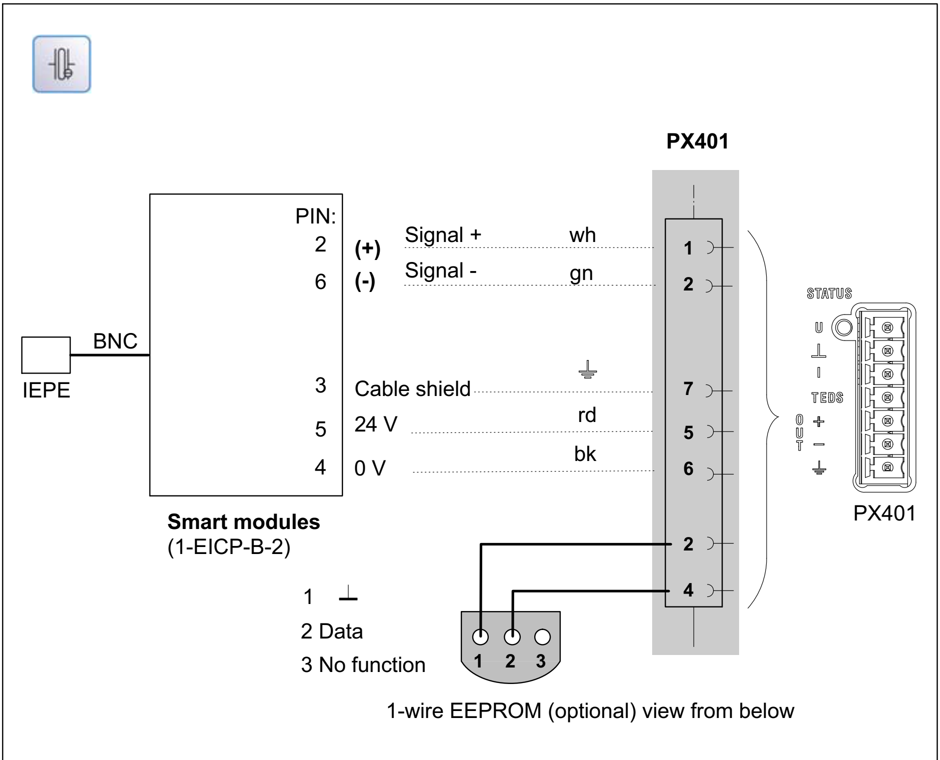


Fig. 8.9 PX401 plug terminal IEPE transducers

8.4.3.5 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 take place in the PMX from an external controller or via a PX878 digital output. The CMA charge amplifier can be supplied directly by the PX401 measurement module.

Important

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

Piezoelectric sensors with external charge amplifier

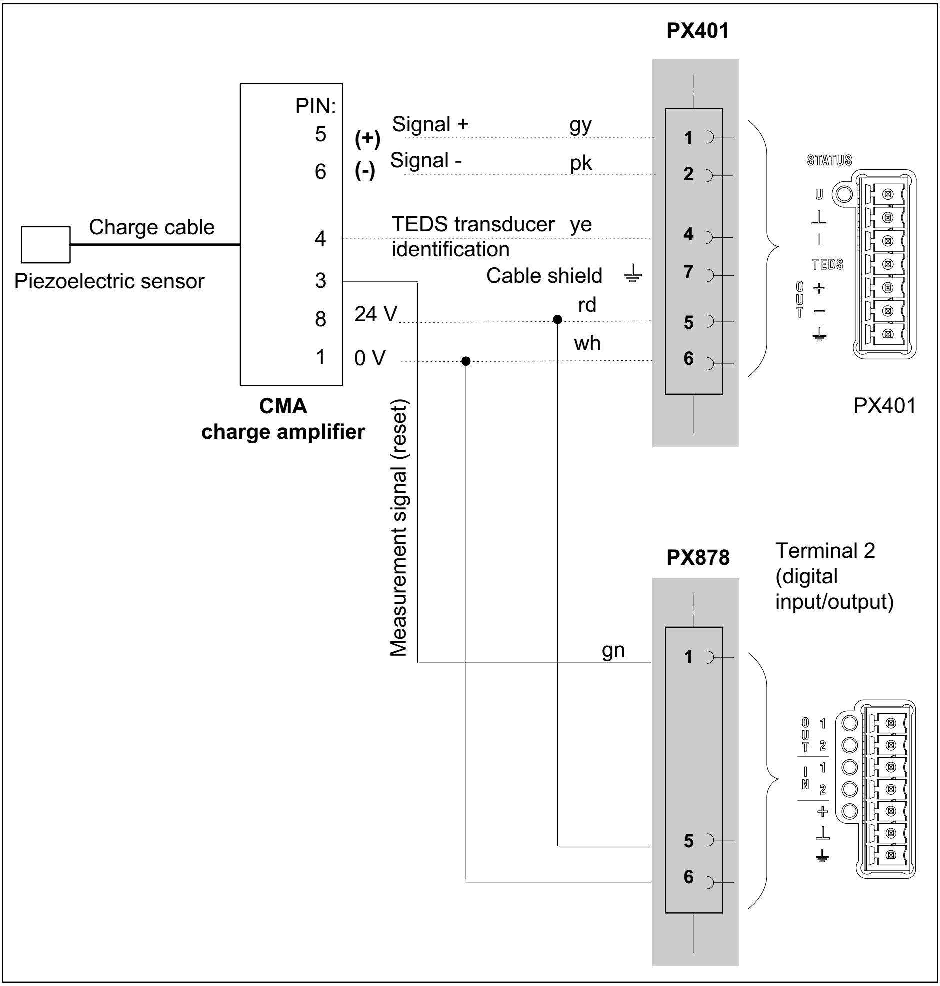


Fig. 8.10 PX401 pin assignment with external charge amplifier

External transducers are supplied with power via the PX401 measurement card (OUT + and OUT -). The supply voltage is identical to the device supply voltage. The maximum current is 400 mA per measurement card and is distributed across the used transducers.

Important

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

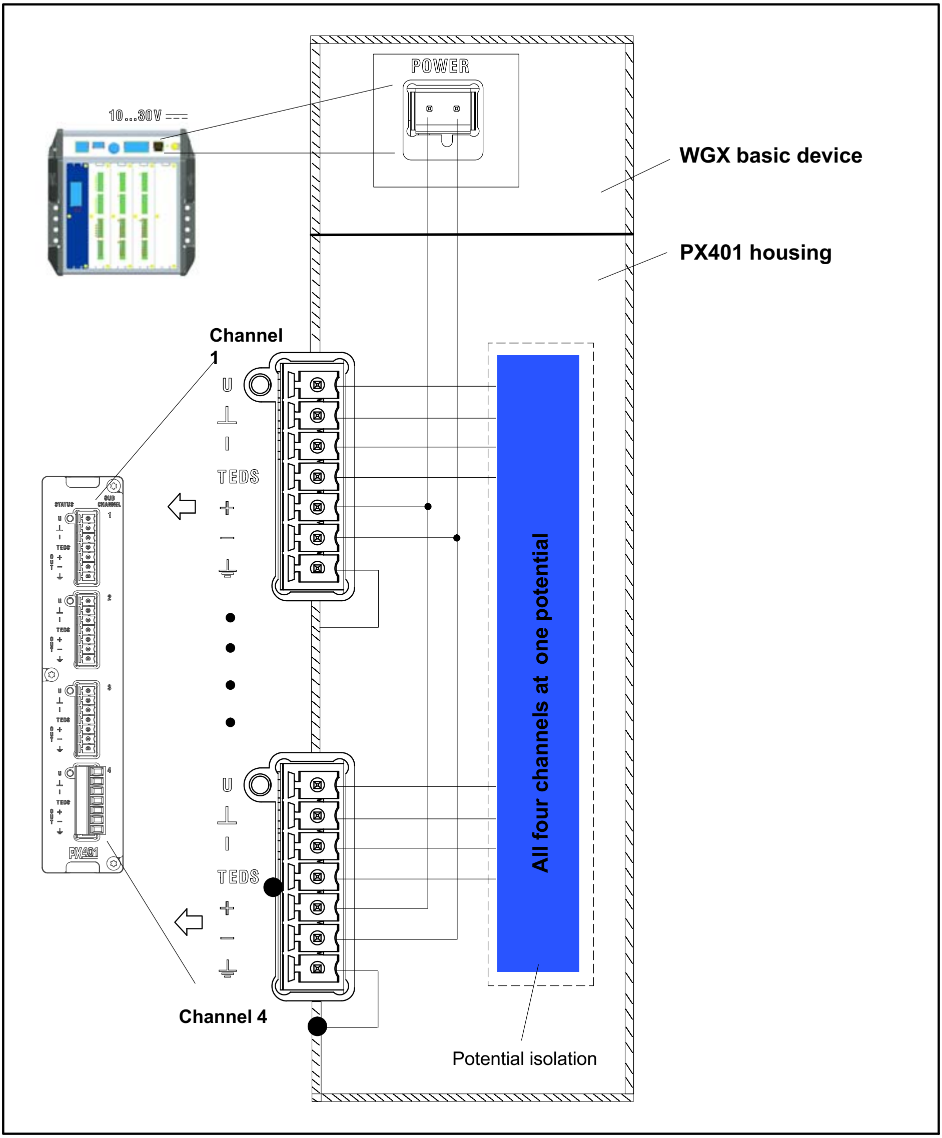


Fig. 8.11 Potential isolation PX401

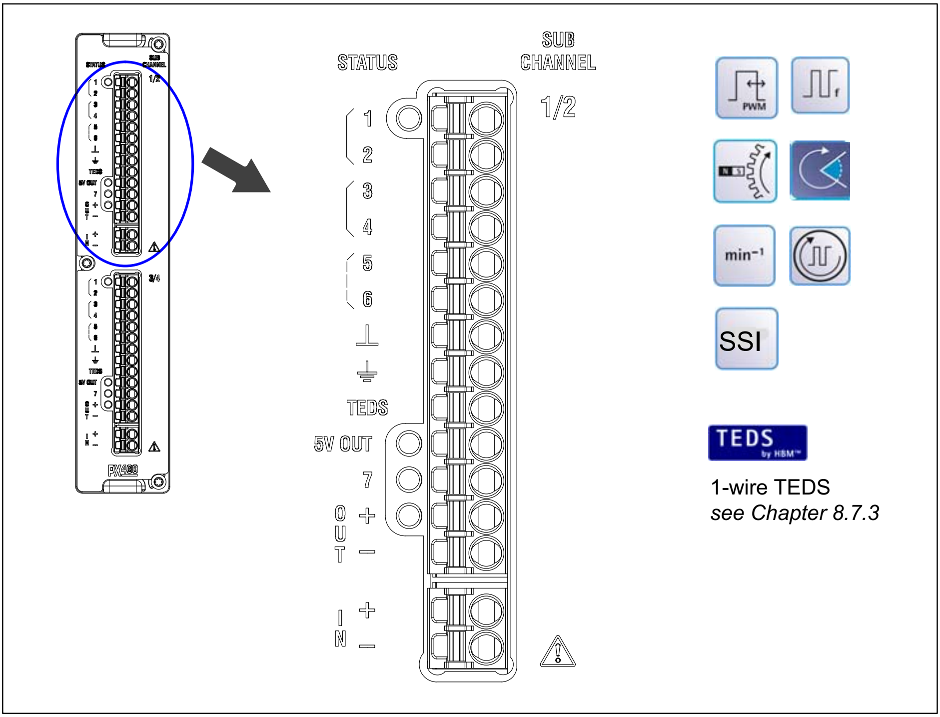
8.4.4 PX460

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



Notice

The sensors for the PX460 are supplied externally with voltage with the contacts (IN + -). The PX460 card then provides the supply for 24 V (OUT + -) and 5 V (5 V OUT).
The input signals fed from the sensor into the PX460 must not exceed ***max. ±12 V****. Otherwise the measurement inputs of the PX460 may be destroyed.*
A shunt can be operated via Pin7. It can be activated via the PMX web browser or the PMX command /dot-NET-API/Catman.

8.4.4.1 Voltage supply for signal transmitter and transducer up to 24 VDC nominal (rated) voltage.

NOTE

*The input signals fed from the sensor into the PX460 must not exceed **max. ±12 V**. Otherwise the measurement inputs of the PX460 may be destroyed.*

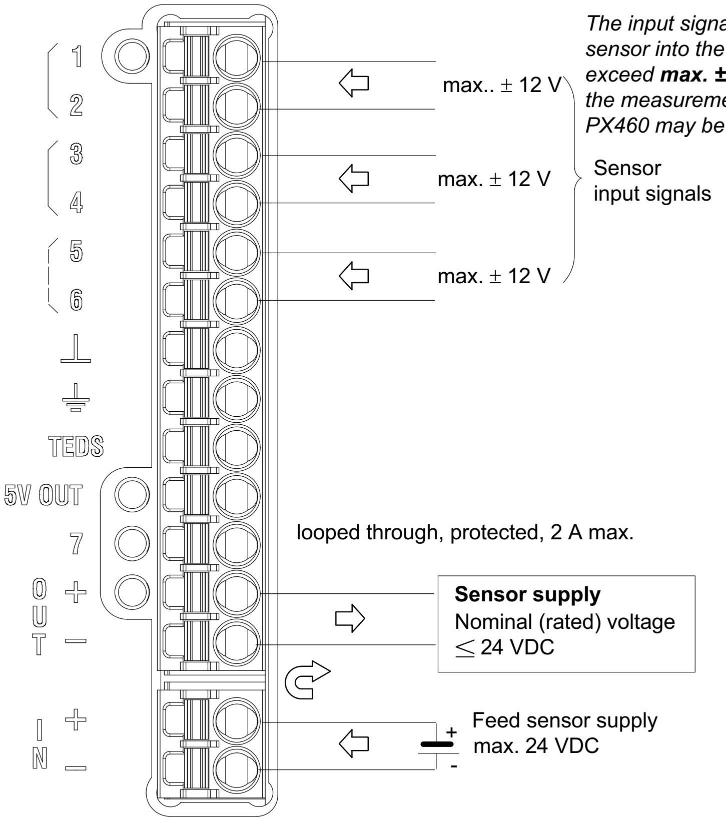


Fig. 8.12 PX460 Voltage supply options up to 24 VDC nominal (rated) voltage

8.4.4.2 Voltage supply for signal transmitter and transducer up to 5 VDC nominal (rated) voltage

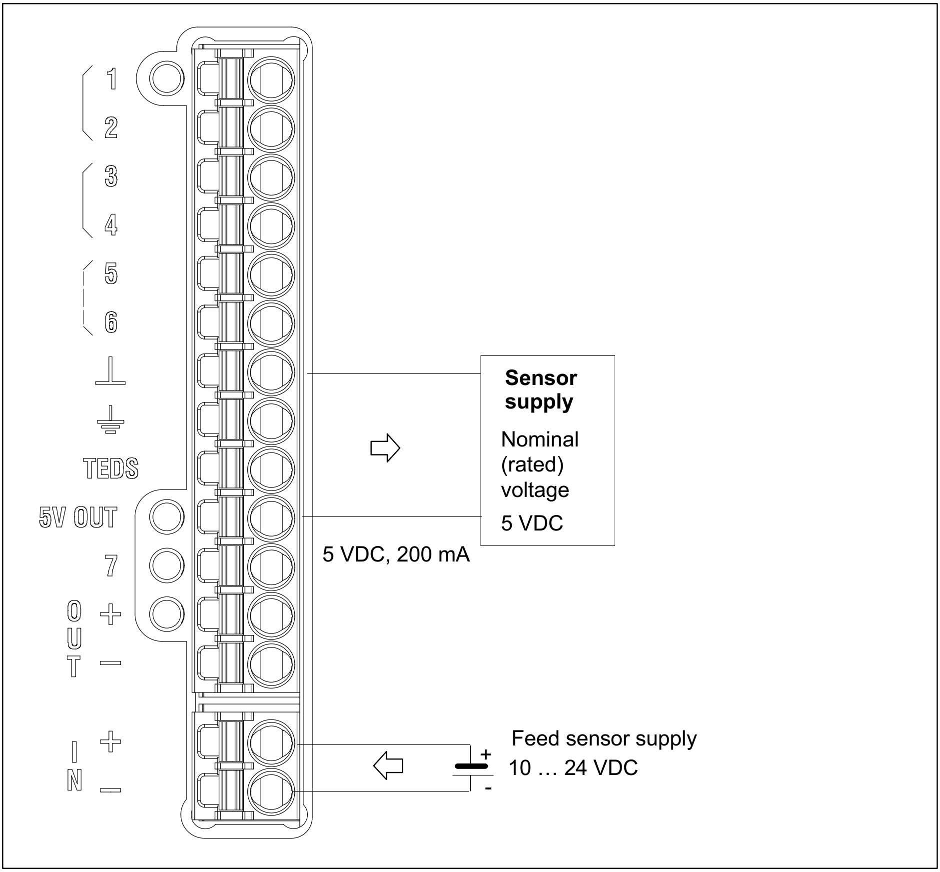


Fig. 8.13 PX460 Voltage supply options up to 5 VDC nominal (rated) voltage

8.4.4.3 Frequency measurement symmetrical (differential)

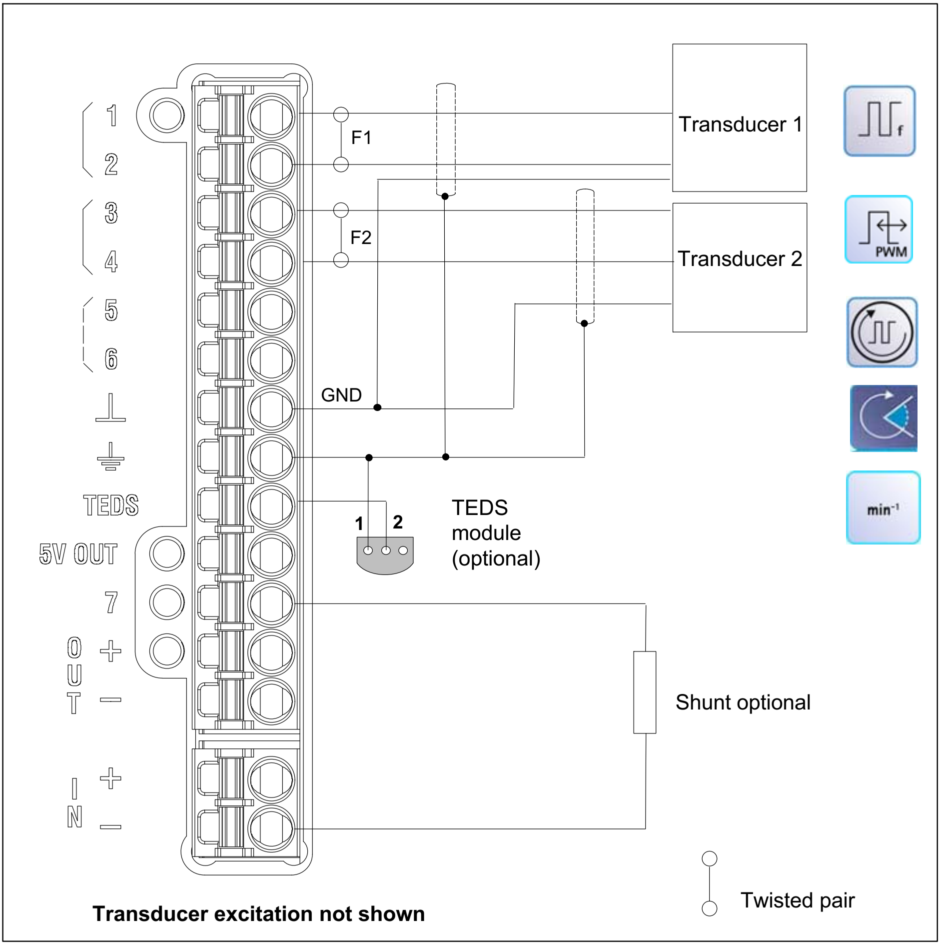


Fig. 8.14 PX460 pin assignment two frequencies, differential

Possible channel setting via web server:

Transducer 1 : Frequency (digital), permanent

Transducer 2 : Frequency (digital), counter, PWM

8.4.4.4 Frequency measurement symmetrical (single-pole)

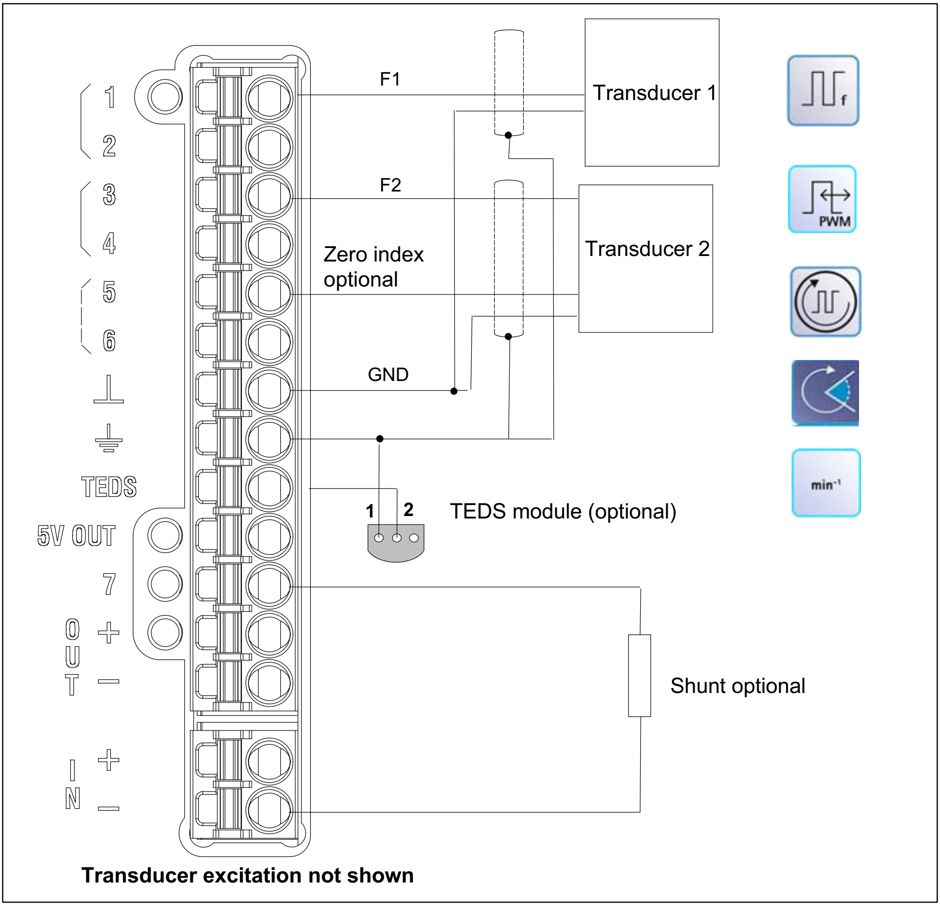


Fig. 8.15 PX460 pin assignment two frequencies, asymmetric

Possible channel setting via web server:

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

8.4.4.5 Encoder and incremental encoder, symmetrical (differential)

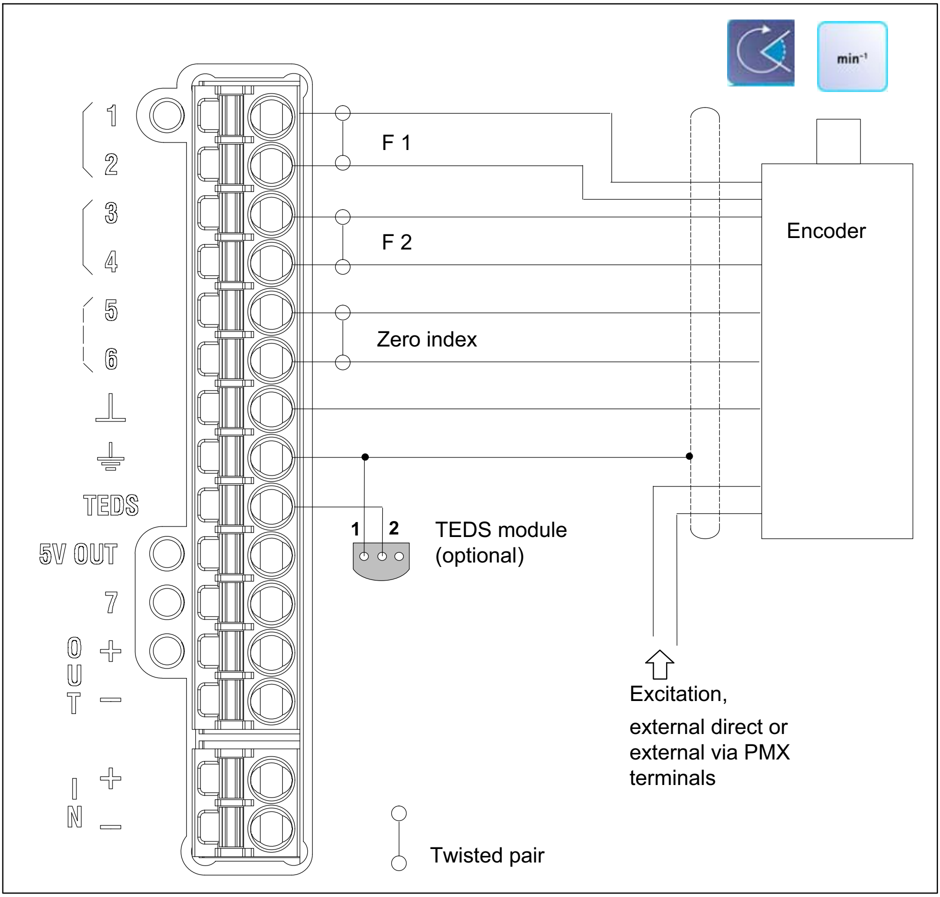


Fig. 8.16 PX460 pin assignment encoder, symmetrical

Possible channel setting via web server:

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

8.4.4.6 Encoder and incremental encoder, asymmetrical (single-pole)

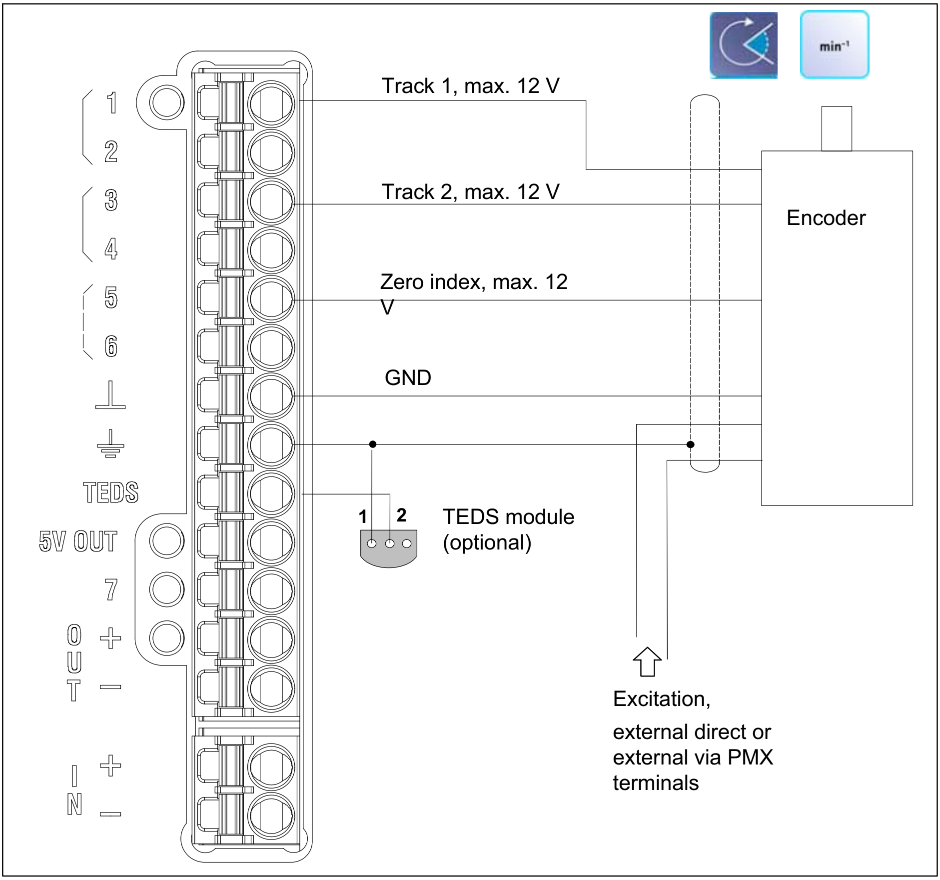


Fig. 8.17 PX460 pin assignment encoder, asymmetrical

Possible channel setting via web server:

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

8.4.4.7 SSI encoder (active only)

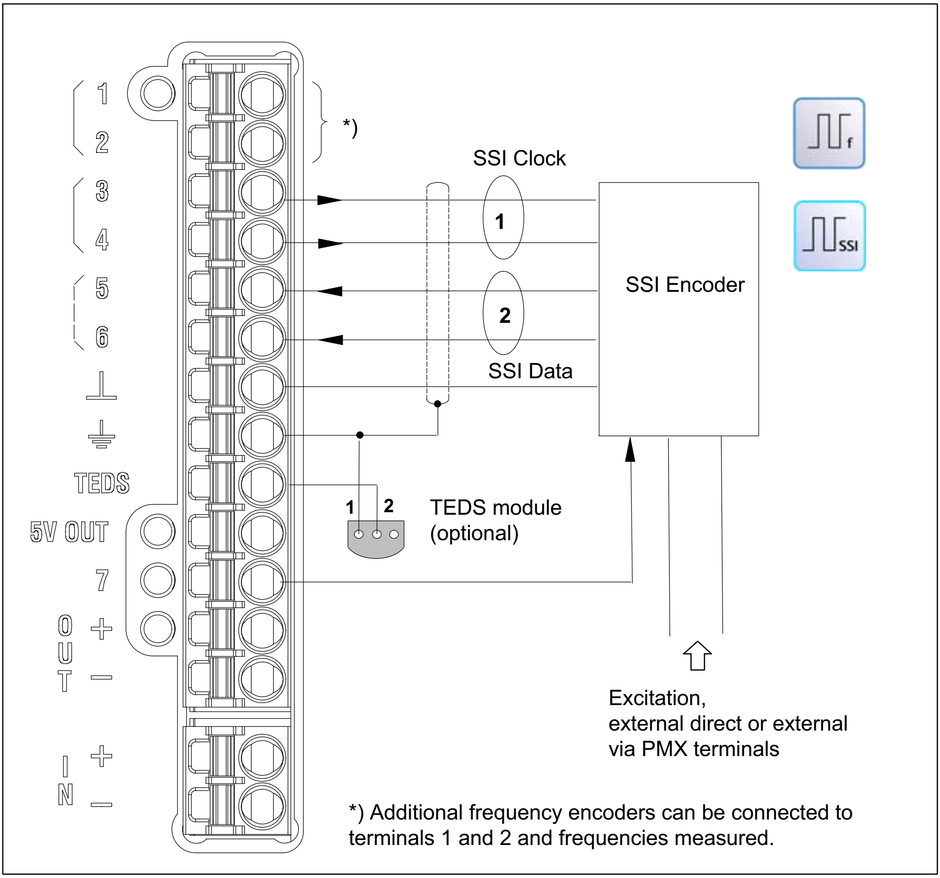


Fig. 8.18 PX460 pin assignment SSI encoder

Possible channel setting via web server:

Transducer: SSI

8.4.4.8 Inductive encoder or pulse encoder (passive only)

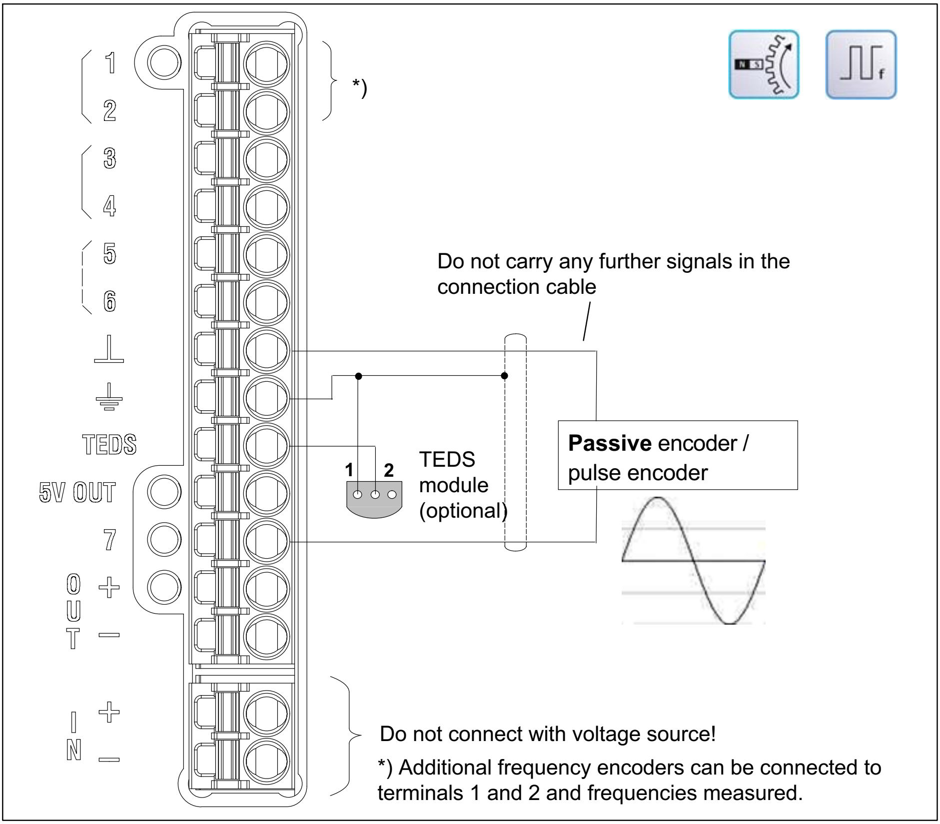


Fig. 8.19 PX460 pin assignment encoder and pulse encoder, differential

Possible channel setting via web server:
Transducer: Frequency (inductive)

Important

This signal input is only designed for passive pulse encoders.

Pin 7 not connected or connected as output:

If Pin 7 is connected as input, both potential ranges are connected (see Chap. 8.4.4.8)

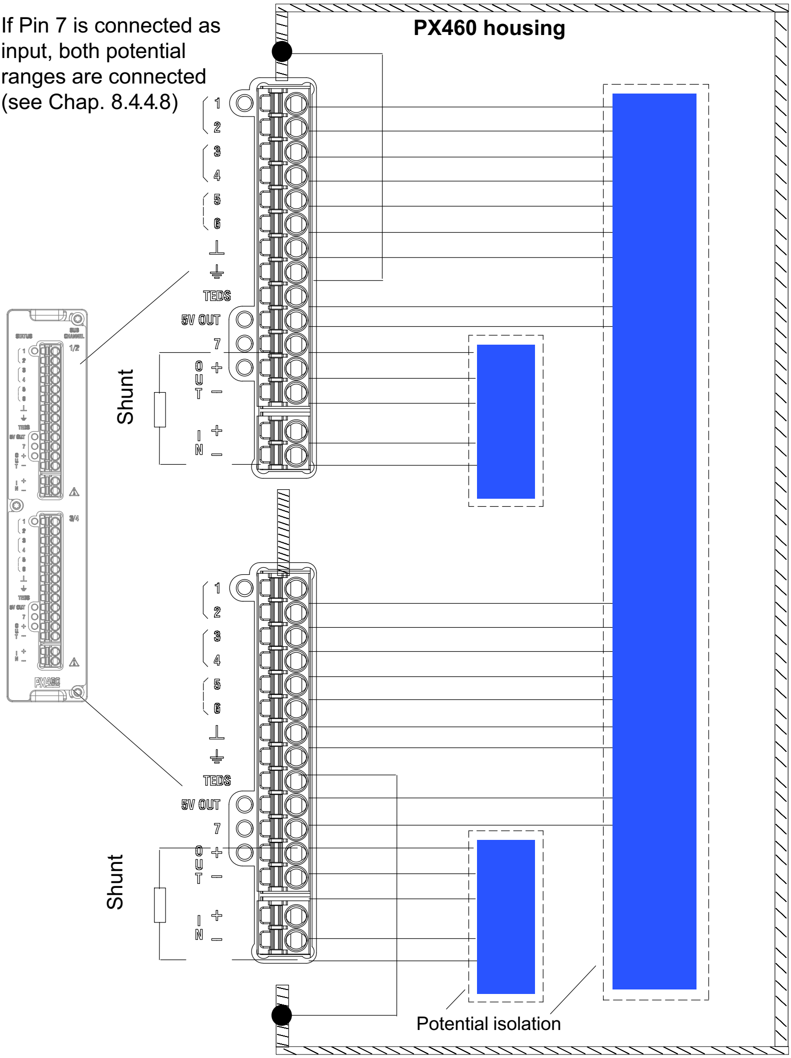


Fig. 8.20 Potential isolation PX460

8.4.4.9 Connection and configuration of the HBM torque flange (T10, T12, T40)

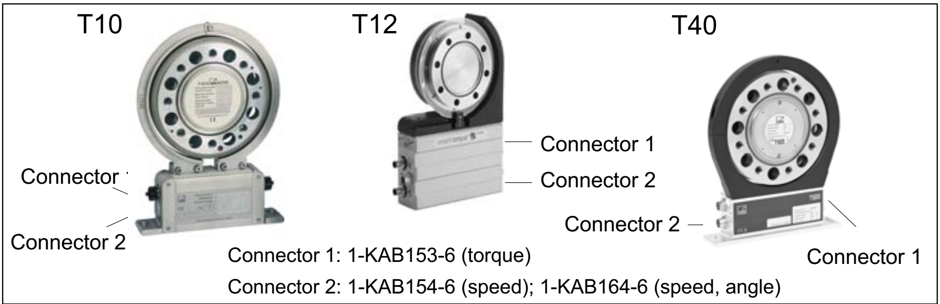
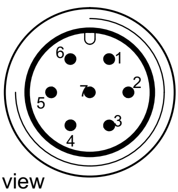


Fig. 8.21 Connector positions T10, T12, T40

Assignment for connector 1: T10, T12, T40

Supply voltage and frequency output signal.

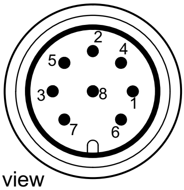
Device connector	**Connector pin**	**Assignment**0V	**Wire color**
 Top view	1	Torque measurement signal (frequency output; 5 V[1),2)])	wh
	2	Supply voltage 0 V;	bk
	3	Supply voltage 18 V … 30 V	bu
	4	Torque measurement signal (frequency output; 5 V[1),2)])	rd
	5	Measurement signal 0V, symmetric	gy
	6	Shunt signal trigger 5 V … 30 V	gn
	7	Shunt signal	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).

Device connector	Connector pin	Assignment	Wire color
 Top view	1	Speed measurement signal 1) (pulse string, 5 V; 0°)	rd
	2	Reference signal (1 pulse/revolution, 5 V1))	bu
	3	Measurement signal speed (pulse string, 5 V; phase shifted by 90°)	gy
	4	Reference signal (1 pulse/revolution, 5 V)1)	bk
	5	No function	vi
	6	Speed measurement signal 1) (pulse string, 5 V; 0°)	wh
	7	Measurement signal speed (pulse string, 5 V; phase shifted by 90°)	gn
	8	Supply voltage zero	bn
		Shielding connected to housing ground	

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

Connection examples (torque flange):

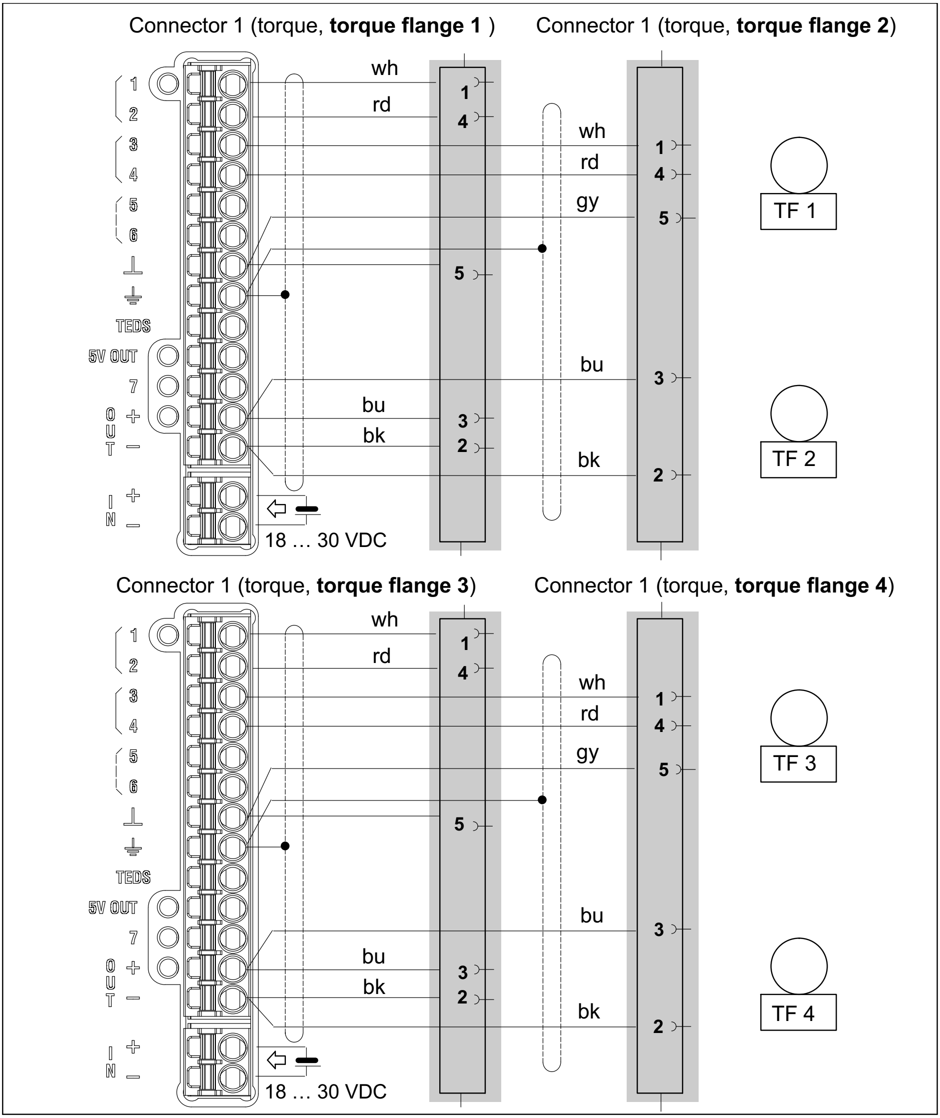


Fig. 8.22 PX460 four torque flanges, torque only

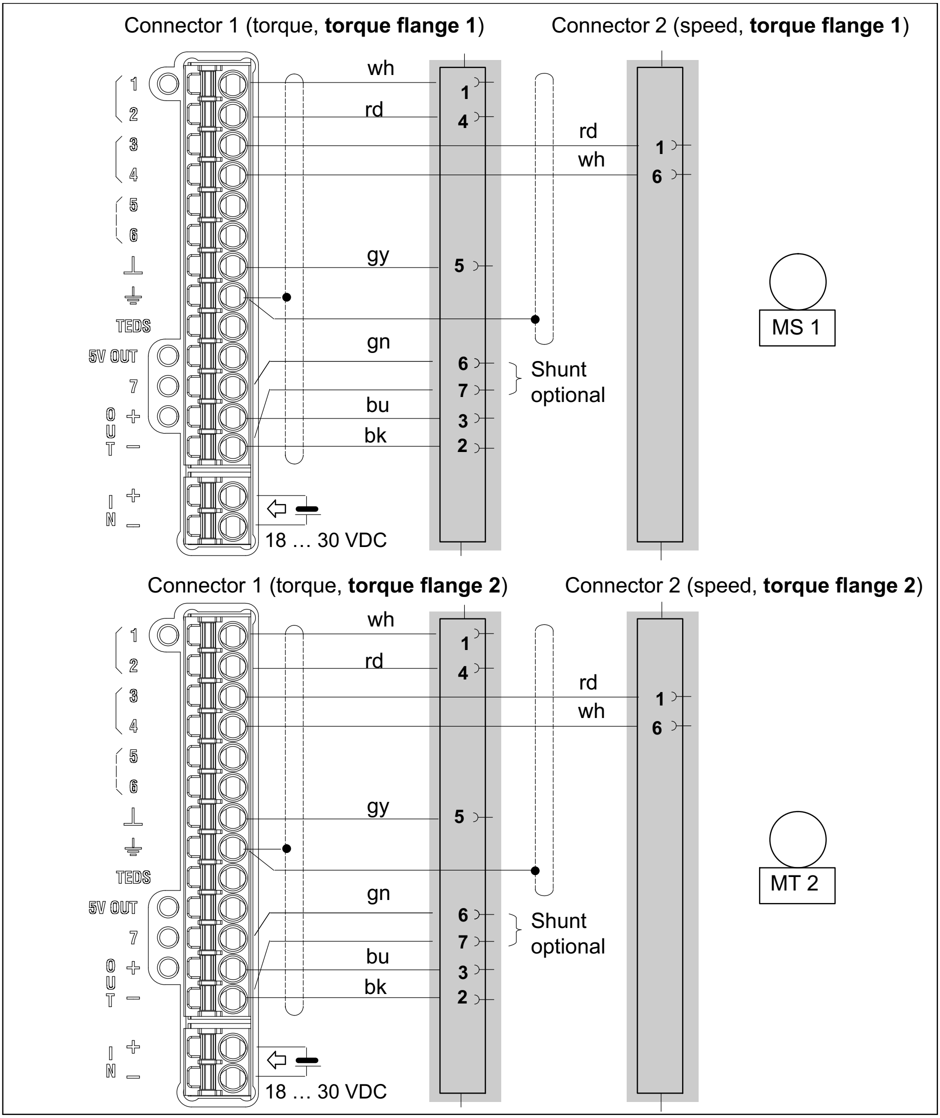


Fig. 8.23 PX460 two torque flanges, torque and speed without angle/direction of rotation

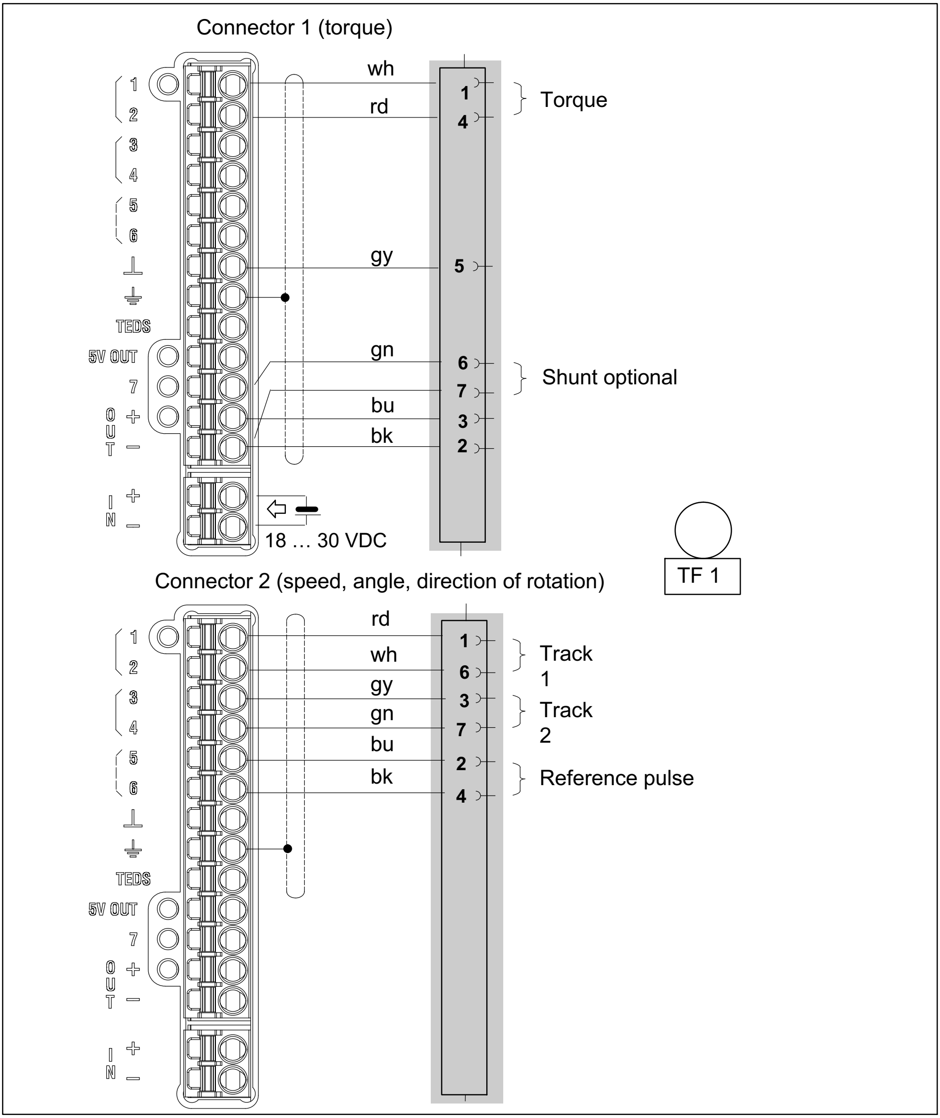


Fig. 8.24 PX460 one torque flange, torque, speed and angle/direction of rotation

Setting example (web browser): T40B on PMX

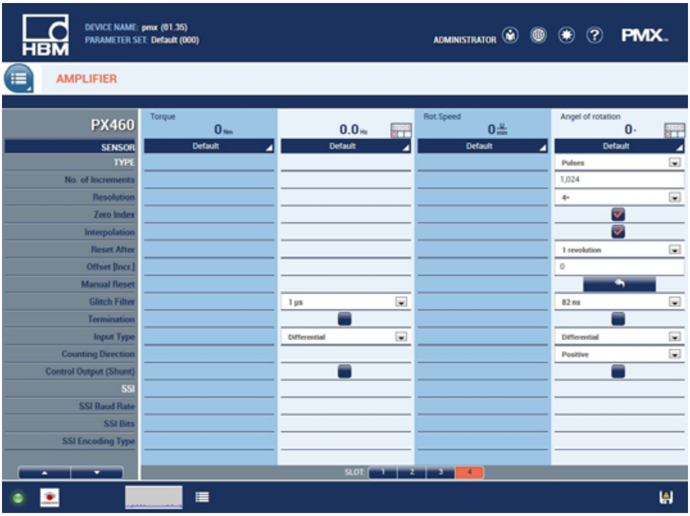
Channel 1 (PX460 upper connector):
Torque, center frequency 10kHz, nominal (rated) torque 1kNm

Channel 2 (PX460 upper connector):
not in use

Channel 3 (PX460 lower connector):
Speed in rpm, 1024 encoder

Channel 4 (PX460 lower connector):
Angle in degrees; 4x resolution,
therefore 4096 pulses = 360°

The measured values fluctuate around the real value dependent on the measurement principle (pulse counting). Therefore set filter provisionally to 500 Hz, then adapt to application.


DEVICE NAME: pmx (01.35)
PARAMETER SET: Default (000)
ADMINISTRATOR
PMX
AMPLIFIER
PX460
Torque
Rot. Speed
Angel of rotation
Default
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
Pulses
1,024
1 revolution
1 µs
82 ns
Differential
Positive
SLOT

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

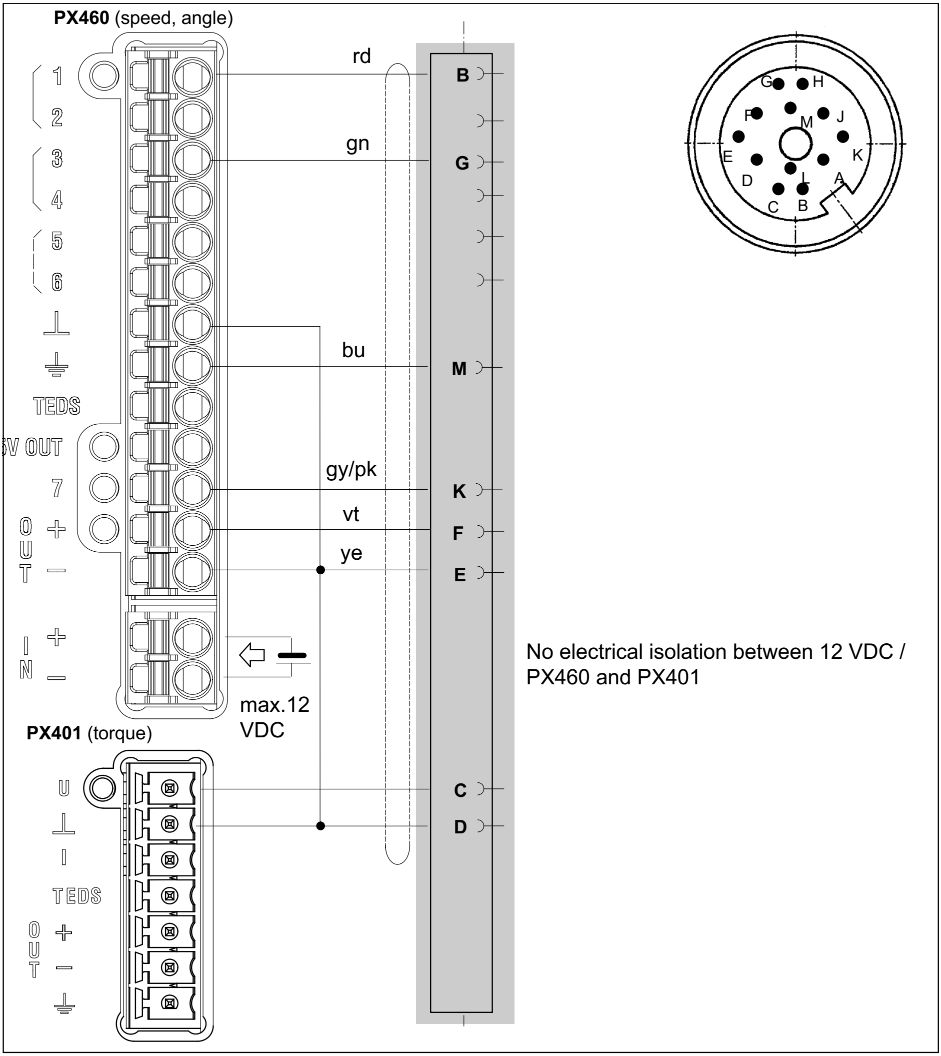
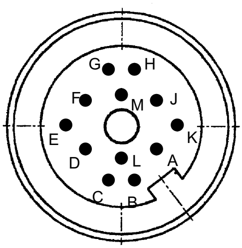


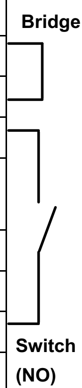
Fig. 8.25 Pin assignment T20WN without VK20A

> ## *Notice*
>
> *The voltage supply for the torque transducer T20WN must not exceed 12 V. The supply can be implemented via an external power supply unit (max. 12 V) or via the PMX device (with max. 12 V PMX device supply). Alternatively, the torque transducer can also be connected via the junction box VK20A to the PMX device. This can be supplied with max. 30 V (also via the PMX).*

Assignment T20WN connector



Pin	Assignment	Wire color
A	No function	bk
B	Speed/angle measurement signal 5 V	rd
C	Torque measurement signal ±10 V	br
D	Torque measurement signal 0 V	wh
E	Ground (supply + speed/angle)	ye
F	Supply voltage +12 V	vt
G	Speed/angle measurement signal 5 V; displaced by 90°	gn
H	No function	pk
J	No function	gy
K	Control signal trigger	gy/pk
L	No function	bu/rd
M	Cable shield	bu



Transducer connection cable:
3-3301.0158, 5 m
3-3301.0159, 10 m

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

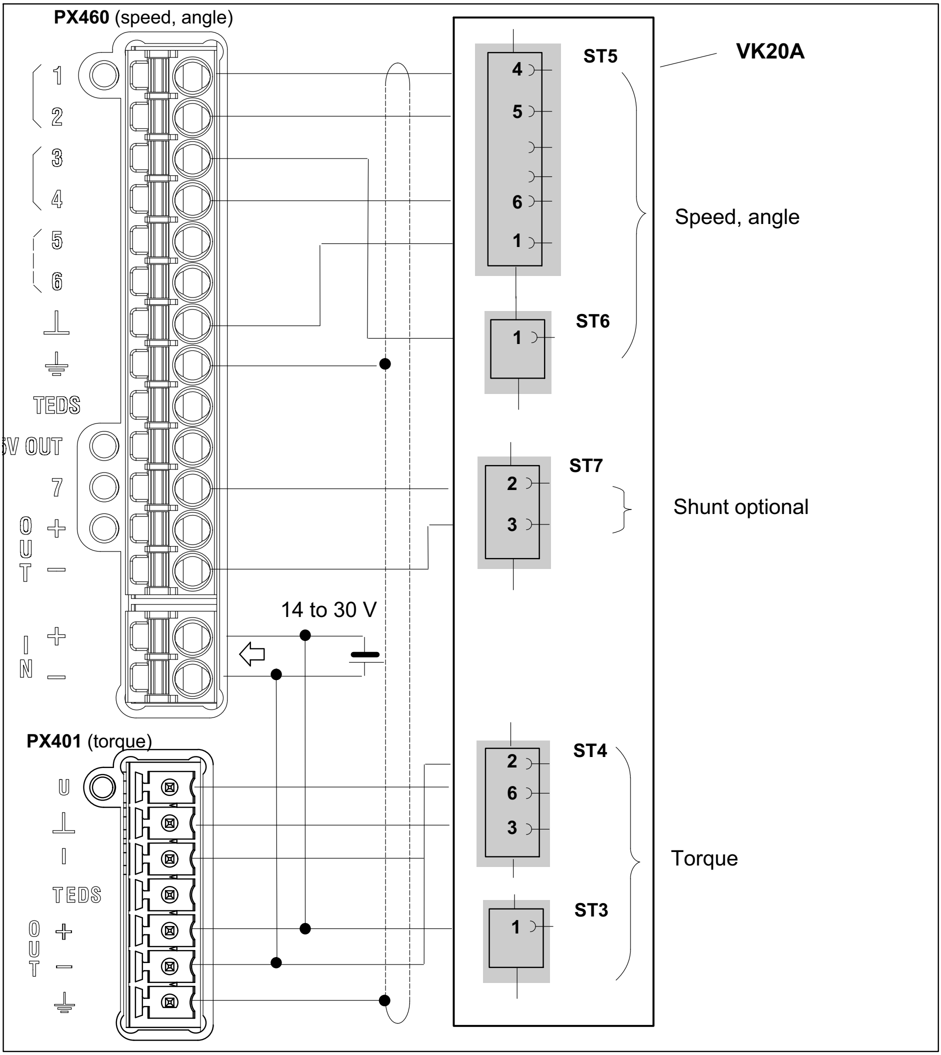
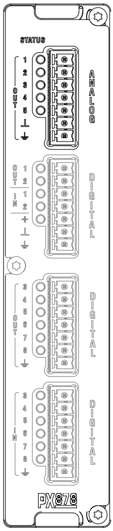


Fig. 8.26 Pin assignment T20WN with VK20A

8.5 Input/output cards

8.5.1 PX878

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



Terminal 1	Analog output
Terminal 2	Digital input/output
Terminal 3	Digital output
Terminal 4	Digital input

8.5.1.1 Analog output ± 10 V

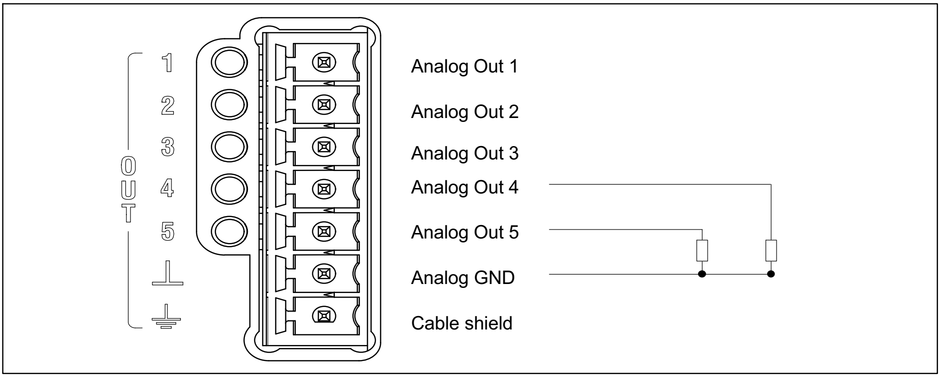


Fig. 8.27 Pin assignment analog output (terminal 1)

8.5.1.2 Digital inputs / digital outputs

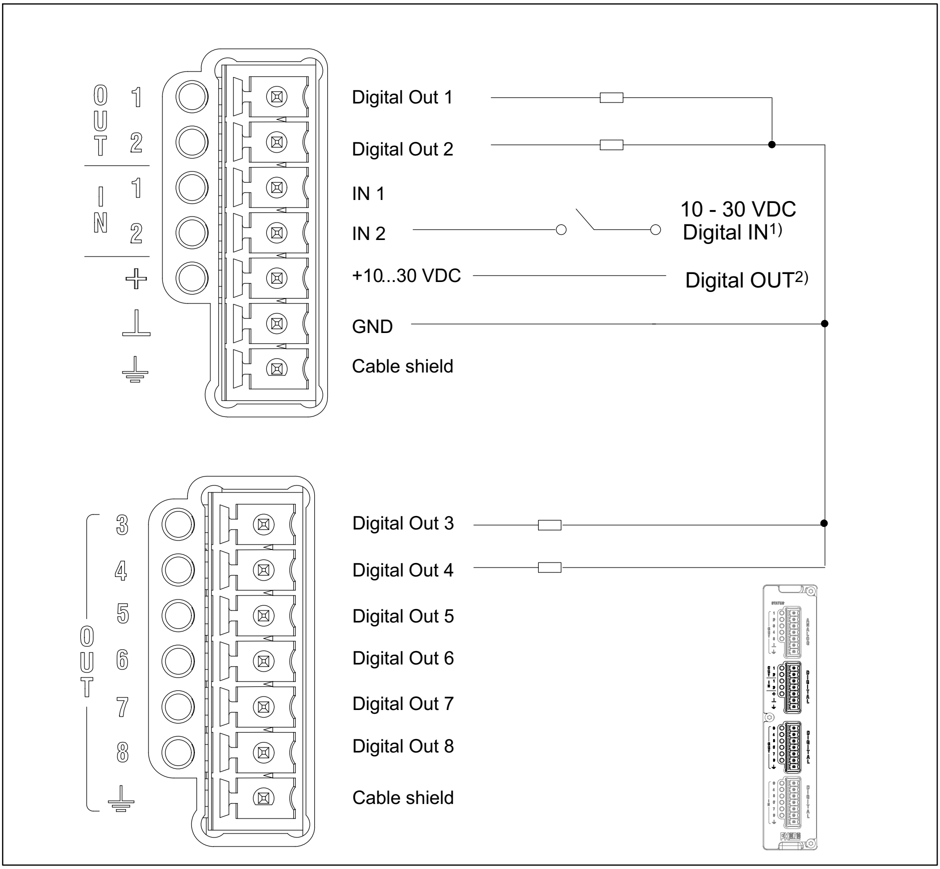


Fig. 8.28 PX878 pin assignment digital input/output (terminals 2 and 3)

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

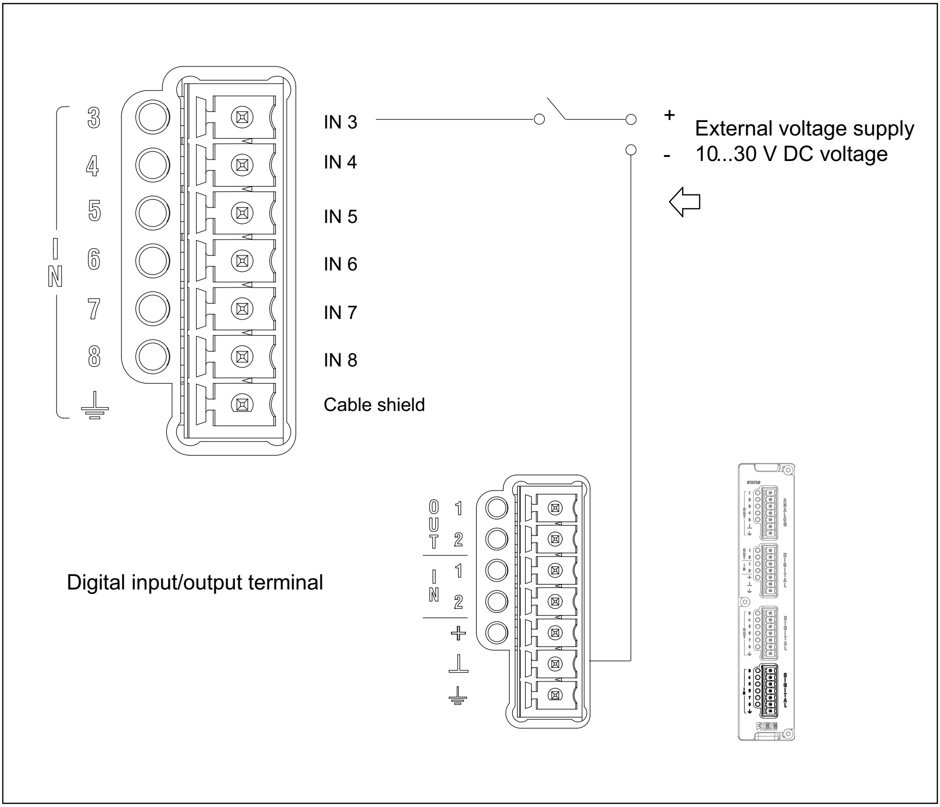


Fig. 8.29 PX878 pin assignment digital input (terminal 4)

Important

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

8.5.2 External supply voltage for digital inputs and outputs (PX878)

Example: PLC connection (p-switched)

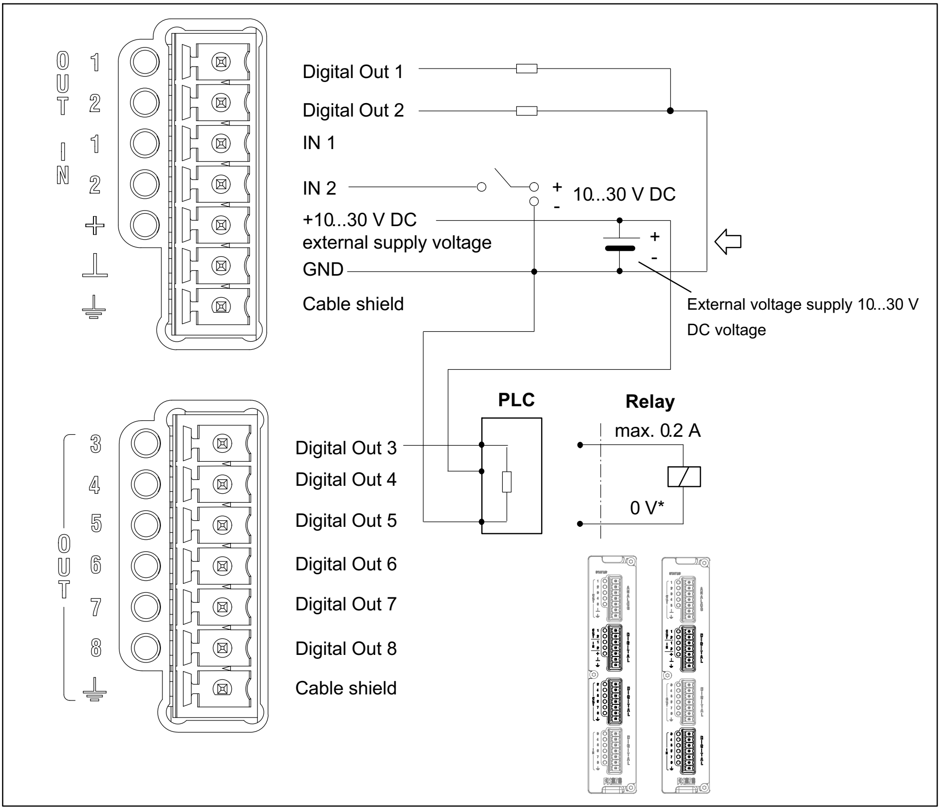


Fig. 8.30 PX878 pin assignment digital input/output and digital output

The *control outputs* are available at the Digital OUT 1 and OUT 2 terminals, as well as at the OUT 3 to OUT 8 terminals. They are designed as High-Side switches and are electrically isolated from the PMX housing, but not from each other *(see Fig. 8.31)*.

The *control inputs* are available at the Digital IN 1 and IN 2 terminals, as well as at the IN 3 to IN 8 terminals. They are electrically isolated from the PMX housing, but not from each other *(see Fig. 8.31)*.

Important

Output behavior after switch-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 its set actions.*
- *In the active state, the externally connected voltage source (see + and ⊥ terminals) is switched through internally with low resistance, with the aid of 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 control), a (pull-down) termination resistor must be used to terminate the high resistance state.*
- *An external reference potential (⊥ IN), to which the control input signals relate, must be connected for the control inputs.*

Important

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

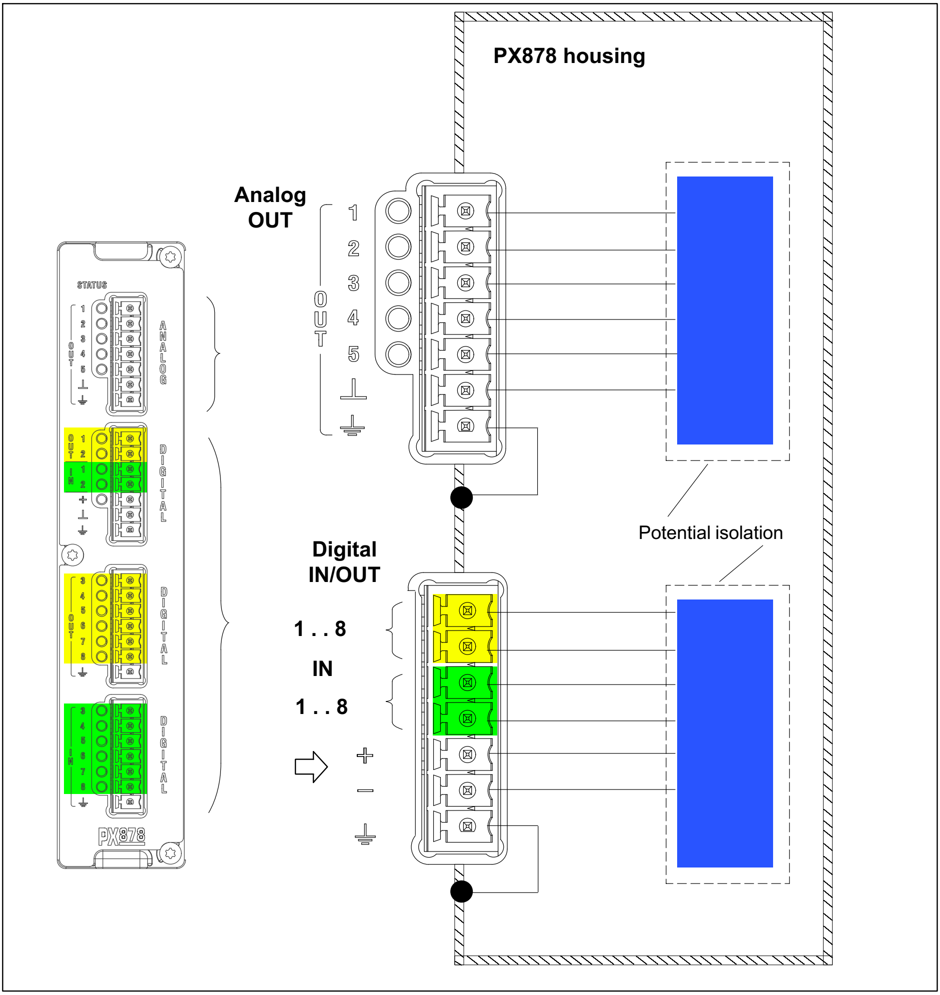


Fig. 8.31 Electrical isolation PX878

8.6 Communication cards

8.6.1 Pin assignment PX01EC EtherCAT® fieldbus module

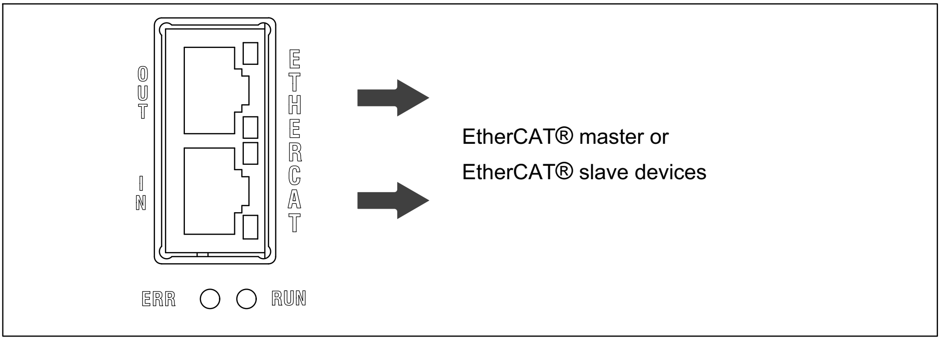


Fig. 8.32 EtherCAT® connection as per standard[1)]

8.6.2 Pin assignment PX01EP EtherNet/IP fieldbus module

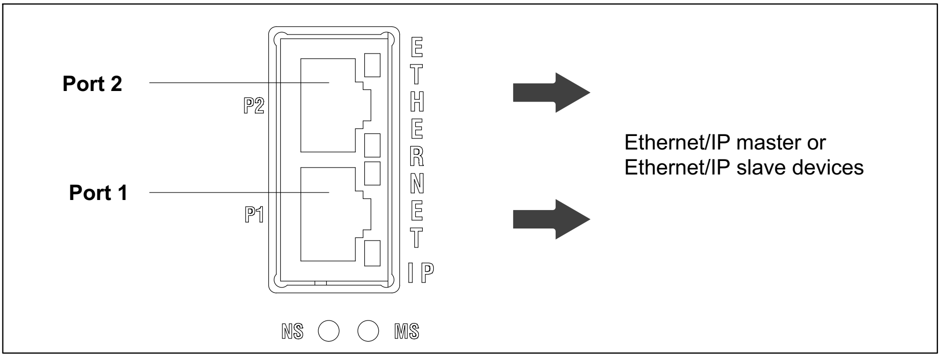


Fig. 8.33 EtherNet/IP connection as per standard[1)]

1) See user organization standards

8.6.3 Pin assignment PX01PN PROFINET-IO fieldbus module

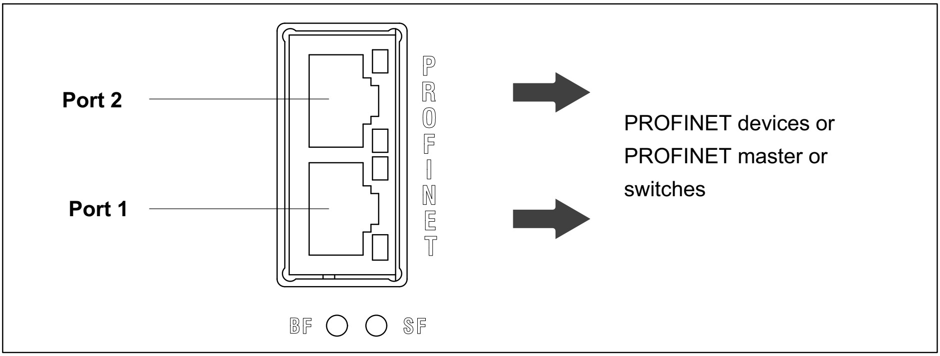


Fig. 8.34 PROFINET connection as per standard[1)]

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 measurement can then start.[1]

1) See user organization standards

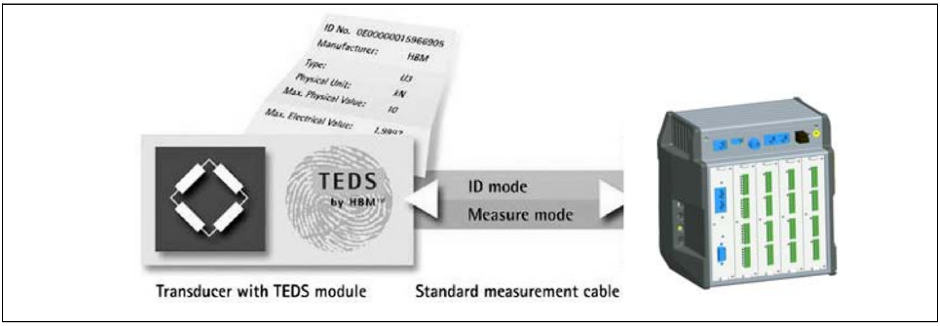


Fig. 8.35 PMX with TEDS technology

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 an EEPROM. The amplifier communicates with this EEPROM via the serial interface, reads the data sheet and makes the corresponding amplifier settings.

Two connection types of TEDS chips are used in PMX:

0-wire TEDS: PX455

The TEDS chip is connected here in the sensor leads of the transducer. No further leads are necessary, the measurement is briefly interrupted to read in the TEDS information in the PMX.

1-wire TEDS: PX401, PX460

The TEDS chip is connected here to the PMX measurement channel with 2 additional leads (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.

TEDS modules that are already mounted in the connector can be subsequently configured and parameterized with a TEDS dongle (1-TEDS-DONGLE) and a TEDS Editor.

The TEDS Editor “scans“ the TEDS module and indicates its readiness for programming (also see the HBM brochure: TEDS data memory in the transducer - content and data memory editing).

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 has a 0-wire TEDS. The sense leads of the sensor cable are used to address the TEDS chip.

Important

PMX only works with the 2-point scaling of the TEDS. Scalings that have been stored as tables or polynomials cannot be read in. But an internal calculation channel of the PMX can be used for this purpose (see Chap. 13).

In the PX401 and PX460 measurement cards, the TEDS chip is addressed separately via an additional connection (1-wire TEDS).

The TEDS functionality can be set by channel in PMX to determine how PMX should behave after switching on or plugging in TEDS sensors:

1. Ignore existing TEDS
2. Read in TEDS and configure measurement channel only when a TEDS sensor is present
3. Always read in TEDS and configure the measurement channel

When the PMX is switched on, it automatically detects whether a sensor with TEDS is connected. The data are read out and the amplifier channel is parameterized accordingly. When the transducer is replaced in the activated state, the new TEDS is also detected automatically but has to be activated manually.

Important

The PMX web server does not have a TEDS editor with read and edit functions. The catman®EASY/AP software includes a complete TEDS editor. This allows TEDS information to be read and written from TEDS sensors that are 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 is used for this, and every PMX has one.

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 included in the transfer of every measured value. The absolute measurement time must come from a different source. This can be the internal time from 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, the applicable channels must be parameterized with the same filter settings. No automatic runtime correction is carried out. The filter runtimes are shown in the data sheet.

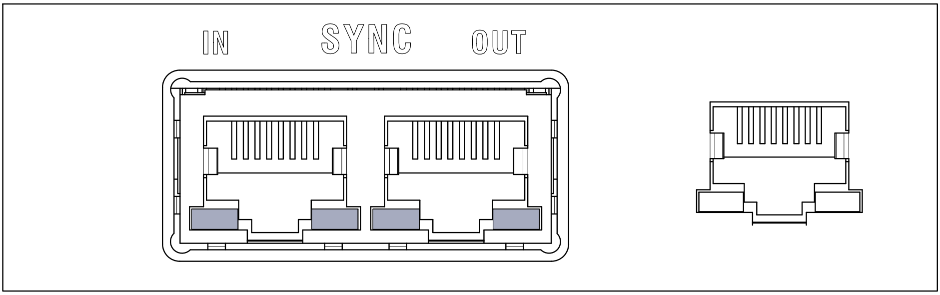
When PMX devices are connected, the carrier frequency and the time stamp can be synchronized via the SYNC socket. The status can be read from the LED.

The master/slave allocation is automatic.

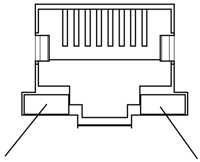
9.1 Synchronizing via internal PMX synchronization

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.

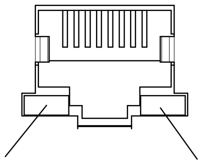
The synchronization status can be read on the SYNC socket LEDs. The synchronization master/slave allocation is automatic, i.e. one PMX device is automatically declared the time master.



IN socket LEDs:

IN		Significance
		
green	Off	Slave
Off	Off	Master
green	yellow	Error

OUT socket LEDs:

OUT		Significance
		
green	Off	Voltage supply available
green	yellow	Error (always identical to the right-hand LED of the **IN** socket)

20 PMX devices can be synchronized with one another in this way. The cables between neighboring devices must be no more than max. 30 m long. Recommended cable: CAT5-SFTP standard Ethernet.

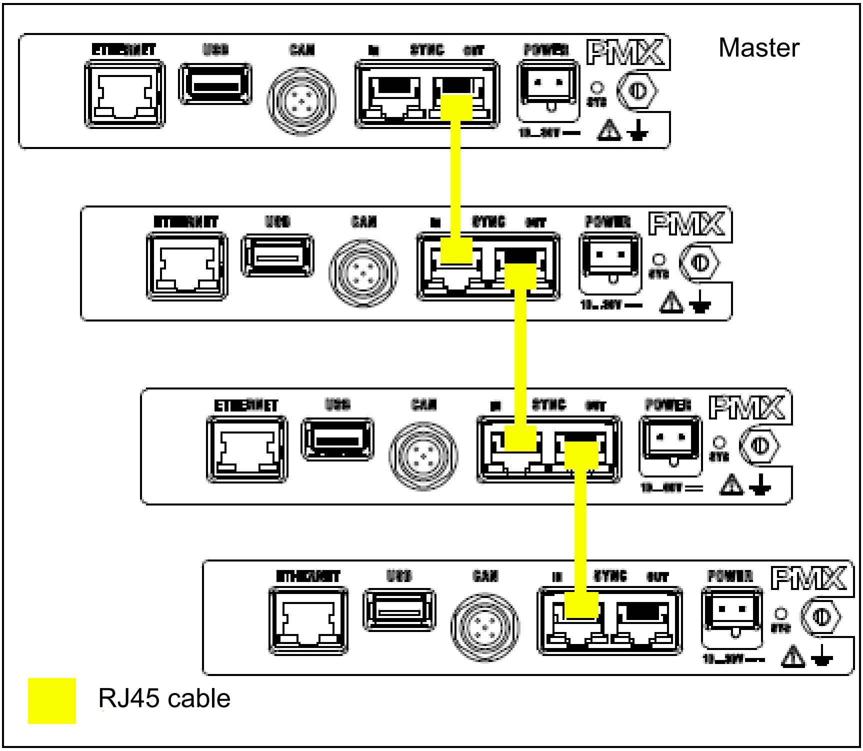


Fig. 9.1 Connecting several PMX modules

Important

The SYNC connection is not used to supply voltage to the devices.
SYNC sockets are not the same as Ethernet. Do not connect with Ethernet
Do not interconnect SYNC cables to make a ring.
In the event of a power failure, the time stamp is not buffered, but begins again at zero after the restart.

9.2 External synchronous measurement 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 module has an NTP time that can be set by an external NTP server. The NTP time is distributed to all the modules over the Ethernet (TCP/IP) connection.

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

When operating the PMX and other devices, it is then possible to achieve accuracies of 1 ms and better. But this depends on the utilization of the particular network, as well as on the quality of the NTP master. An NTP software package is included with the HBM software catman®EASY.

Time format used

Based on	1.1.1990
Time stamp:	64-bit 32-bit seconds 32-bit fractions of a second, Resolution (1/232)

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

9.3 Measurement acquisition via EtherCAT fieldbus, ProfiNET, Ethernet/IP

The measured values cannot be synchronized over fieldbuses and cannot be time stamped, as the PX01EC EtherCAT fieldbus card does not support the EtherCAT® enhancement “Distributed Clocks”. The same is true for the PROFINET and Ethernet/IP fieldbus cards. The time from a fieldbus master cannot be distributed to the PMXs working as fieldbus slaves.

In this type of master/slave system, however, the measured values and data, e.g. peak values and status information, are deterministically transferred. 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 10 ms
Number of modules to be synchronized	20	Unlimited
Maximum distance between modules	30 m	100 m (Ethernet), 10 km specific WLAN bridges
Synchronization settling time	Immediate	Approx. 20 min. during initial startup, approx. 2 min. during restart
Synchronization master	Automatic, the first PMX is the master	Recommended : separate NTP server/master

10 Network, data security, policies, passwords

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 device web pages (hypertext documents) to 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, the following protocols (ports) must be enabled in the PC´s or server's firewall, so that the listed software components can be used:

- **Web interface (PMX web server):**

TCP 80	Parameterization and measurements

  Communication via IPv4 multicast group address: 239.255.77.76 and 239.255.77.77 ports 31416 and 31417

	HBM device scan
UDP 1900	Name resolution (Bonjour ZeroConf)
UDP 1900, TCP 8200	Name resolution (UPnP)
UDP 137	Name resolution (netBIOS)
UDP 123	NTP time synchronization
UDP 514	System log messages

- **Catman / dotNET-API / LabView driver / DIAdem driver**

  TCP 55000 Parameterization and measurement

  IPv4 multicast 31416 and 31417
  HBM device scan

Notice

As from firmware version 3.0, the PMX will only allow 2 Ethernet connections at port 55000.
If a third connection is opened, the PMX closes one of the other two. The time of the last activity of the existing connections is measured internally and whichever connection has been inactive for longer (less data transmission) is selected for closure.

- **CODESYS (optional with WGX001)**

Gateway	TCP/UDP1217
	TCP ports 11740-11743
UDP broadcasts	
	1740
	1741
	1742
	1743
WebVisu	TCP8080

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 and the Internet. Adopt additional safety precautions to prevent this, such as a VPN for remote access, and install firewall mechanisms.

The controller programming ports in particular must never be vulnerable to unprotected access from the Internet.

The PMX ports used can be found in section 10.1.

Limit access to authorized persons, change any default passwords and access rights that may exist when starting up the system for the first time, and at regular intervals thereafter. The procedures for changing PMX passwords and policies can be found on the PMX web server in the dialogs "Change password" and "Define policies", also see section 10.3.

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. Select 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 the "Define Policies" menu, 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 at the ADMINISTRATOR level. Click to select whether you want to set the password for MAINTENANCE or for 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. Click or tap  to turn on the display of the entered characters,  switches it off again.

Notice

Keep the Administrator password somewhere safe. If you forget the password, you must contact HBM Technical Support, also see section 24.3.

11 Startup

This chapter describes starting up the PMX system, how to configure it and how the user interface can be realized and operated.

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 on page 53*.

► You have the option to connect the bus system (EtherCAT or PROFINET or Ethernet/IP).

► Switch on the power supply.

11.1.2 Ethernet connection

The PMX must be connected with a device that has a web browser so that it can be operated and parameterized via the PMX web server.

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.

Category 5 (Cat5) or higher cables must be used for this purpose. This enables line lengths of 100 m to be implemented. A W-LAN wireless connection can also be used.

Important

Ensure in all cases that the HTTP Port 80 is open.

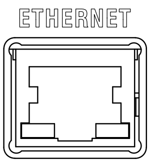
11.2 Integrated PMX web server

11.2.1 System requirements

To operate current versions of the PMX devices, you need a terminal (e.g. PC / tablet with a mouse) with a current Internet browser (Internet-Explorer (Version > 9.0), Firefox or Chrome) and a screen resolution of 1024 x 768.

Windows 7/8, or at least Windows XP, 2000 or Vista, should be installed on the PC.

11.3 Connect the PMX with 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 set at the factory for the PMX is "pmx", but this can be changed.

Address assignment strategy

- Via an IP address that has already been set (not from the factory)
- 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 via the RFC5735 auto IP range.
- If the PMX is set to DHCP, the PC should also be set to DHCP

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

Option	Technology	Operating system
A	netBIOS	WIN XP or later
B	UPnP	Windows Vista or later
C	Bonjour	Apple; Linux; Windows, if "Bonjour Print Services“ is installed
D	Ping to multicast address[1]	

[1] *see Chapter 24 Diagnosis and maintenance*

Tip

If a network connection does not materialize: Plug the network cable in again!

Option A :

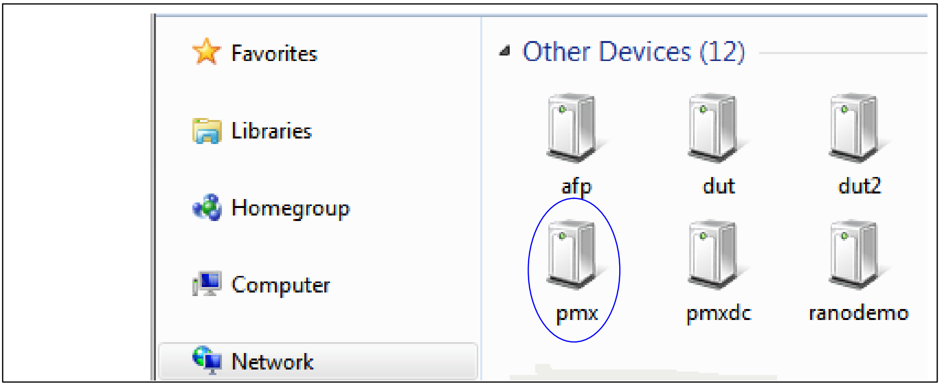
Connection via Universal Plug & Play, from the Windows 7 version

This connection depends on the network settings and is also possible without DHCP and in the auto IP range[1]).

Not available in a PMX - PC connection (without a network) and not in public networks.

► Open the network

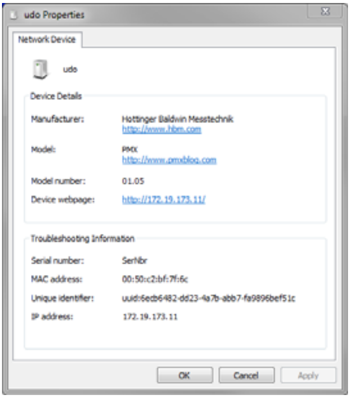
- One or more PMX devices will be found under "Other devices"



Tip

Right-click on PMX and look under "Properties" to find device details such as device webpage, PMX serial number, IP address, etc.

1*) Under WIN7, **"Media Streaming"** must be switched on (Control Panel > Network and Internet > Network and Sharing Center > *MEDIA STREAMING*



If several PMX devices are available in the network, this selection box will also appear:

► Check the box for the required PMX

► Click Connect

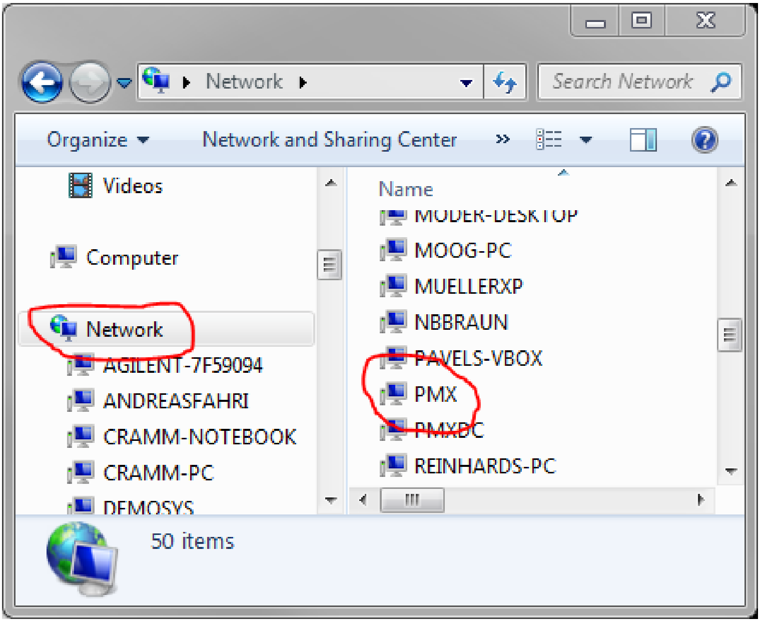
This takes you to the device overview:

Now you can measure, make settings and observe.

Option B:

Connection via the NetBIOS name under Windows

"PMX" appears in the network environment



► Enter "pmx/" (without "" but with /) in the address bar of an Internet browser

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

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

Now you can measure, make settings and observe.

Important

If a DHCP server cannot be found, the PMX device (as per RFC5735 Apipa) automatically assigns its own IP address (169.254.xxx.xxx/16).
Requirement: There is no static IP address entered in the PMX device!
If a static IP address has been set, there will be 2 IP addresses available: the set static address and an IO address from the automatic IP range.

Option C :

Connection with Bonjour Apple software

► Download and install Apple's *"Bonjour Print Services"* (https://support.apple.com/kb/DL999?viewlocale=en_US&locale=de_DE)

If Apple software has already been installed, *Bonjour* will usually already be on the computer.

► Enter ***pmx.local.*** in the address bar of an Internet browser

Now you can measure, make settings and observe.

Important

The device name ("pmx" from the factory) and the network settings (DHCP, IP address, netmask, gateway) can be permanently changed by the user (Network menu item).

11.3.1 Restoring lost network settings

If you cannot find the PMX in the network, you can use a USB memory stick to provide the network settings you require.

1. On a USB memory stick, create a text file called pmx.conf in the root directory
2. Example 1:

   This pmx.conf file sets the device name to "pmx_new_name" and switches PMX into the DHCP mode
   <pmx type="set">
   <hostname>pmx_new_name</hostname>
   <network>
   <dhcp>true</dhcp>
   </network>
   </pmx>
3. Example 2:

   Sets the name to "pmx", as well as a permanent 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>

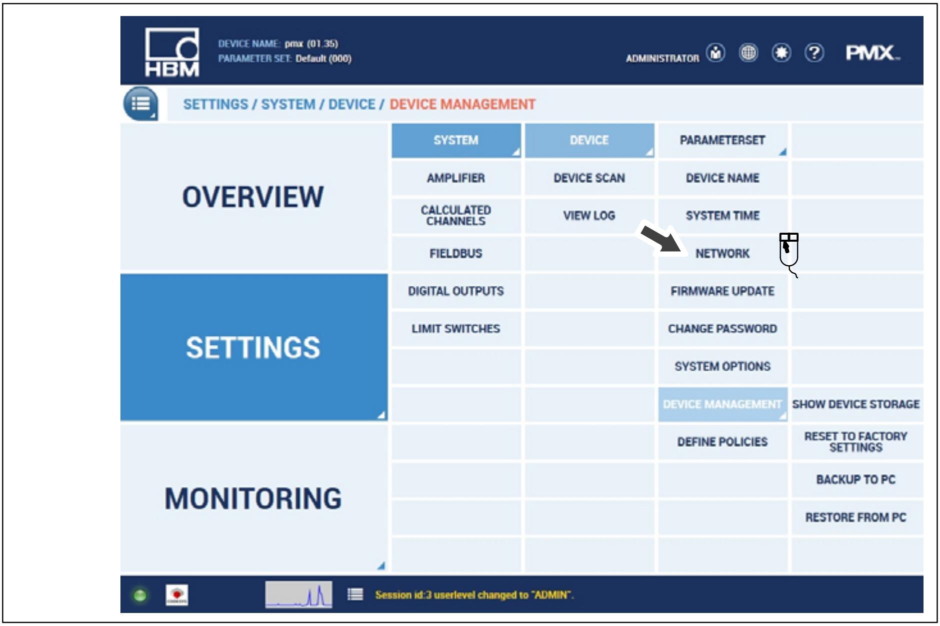
4. Plug the USB stick into the PMX device *while operation is ongoing*.

   The settings will change immediately, but will not be immediately apparent in the other network devices. So it is a good idea to to restart the PMX by interrupting the power supply.

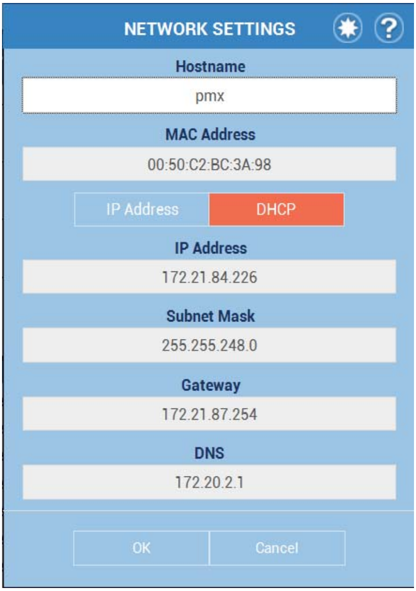
5. The PMX can be found in the network under the new settings.

6. Caution: This memory stick converts each PMX device as soon as you plug it in!
   You should therefore delete the file, rename it or move it to a subdirectory.

Change the network settings



Network settings



11.4 Display and control options

Important

For a detailed description of PMX operation see the PMX online help. To download the current firmware go to http://www.hbm.com/en/menu/support/software-firmware-downloads/industrial-amplifiers/

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

Touch or click on one of the targets, or one of the places where the cursor turns into a hand, to change the relevant setting or to get to the dialog to change the setting.

Calls the setting menu, from which you can reach all the dialogs by using the tabular menu structure. There are additional submenus in all the menu items for which a triangle shows in the bottom-right corner. 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, is displayed at the bottom right, as the setting is initially only stored in the RAM. Touch or click on this symbol, to save the setting, power failsafe in the flash EPROM.

Symbols and displays used

Calls user administration.

The globe icon opens language selection. You can switch to one of the displayed languages.

Calls the list of favorites. You can add any view to the list of favorites

Calls this Help.

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

Everything OK.

There is an error in one or more of the channels, but the PMX keeps on working.

The red LED indicates a critical error has occurred. There is a critical error, measured values can no longer be recorded or processed.
The network connection to the PMX may also possibly be lost. In this case the PMX can continue measuring.

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

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

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

Measured value realization

The realization of measured values and data from the calculated channels of the PMX can be individually adjusted for each measurement channel. This affects firstly the number of decimal places and secondly the increment of the digital scale. The reading can be adjusted to meet the requirements of the application

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

Calculated channels: Option in the Decimal Places drop-down menu

These settings only affect the display value in the PMX web server and do not relate to the values in the PMX.

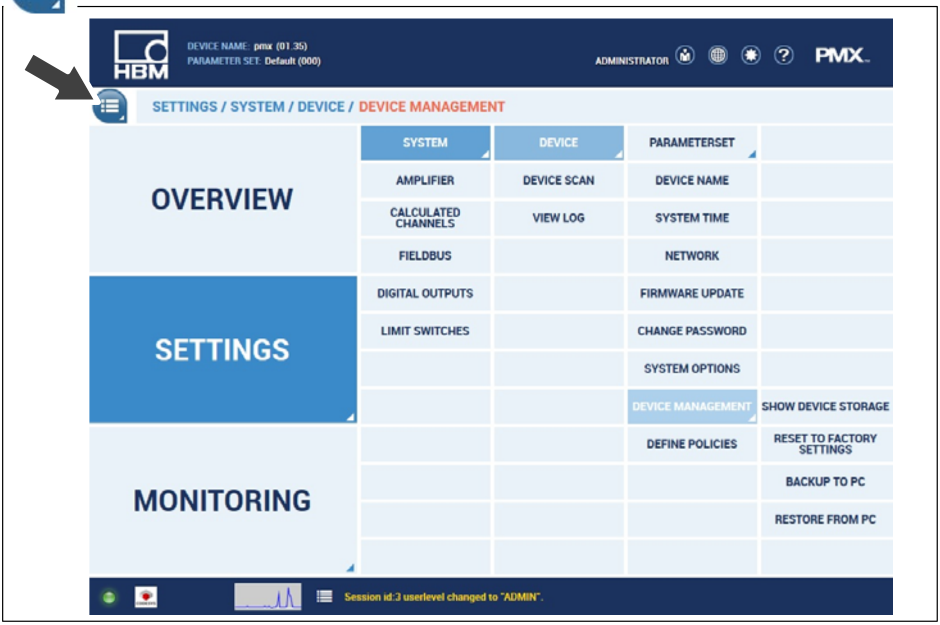
Example: Setting with 2 decimal places and an increment of 2 digits ..0.08..0.06..0.04..

11.5 Menu structure PMX web server

11.5.1 Overview -> SETTINGS

The PMX can be parameterized via SETTINGS. Menus can be selected individually. Each menu item has an online Help, that is called by clicking on the symbol.

► Click on to open the menu page



11.5.2 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 various user levels (Worker, Service, Administrator)
- CODESYS applications and CODESYS web visualizations (for updates that are loaded on a running 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. Once the self-test has been completed, you can read the status of each component from the relevant status LED (see Chapter 8.2.3 to 8.2.5).

- When the PMX is switched on, the digital and analog outputs are set to 0 V.
- When the system powers up, the analog outputs are set to 0 V.
- After power-up, the configured and valid outputs are set to -10 … + 10 V.
- Invalid (unconfigured) outputs jump to 0 V (safe value).
  If an output becomes invalid during operation, it will also jump to 0 V.
- The safe value can be set to any other value, but its default setting is 0 V.

11.7 PMX operating behavior

PMX is suitable according to the intended use for measurement technology tasks with integrated control and regulating tasks. However, it may not be used in areas where malfunctions could lead to personal injuries or damage to property.

However, to improve the operating safety of the system in which PMX is used, several measures have been implemented in PMX.

Setup mode

A target value (test signal) can be simulated for every measurement signal without an actual measured value needing to be present from the system. This allows downstream functions and components to be tested. This can also be simulated for the analog outputs (+/- 10V).

Ongoing operation/measuring mode

- Digital outputs: The device or measured value status can be signaled in detail here via hardware outputs, fieldbus or Ethernet (PC).
- Limit values: The measured value status is normally taken into account during evaluation, i.e. a switching process does not occur if the measured value is invalid. This can be prevented by Ignore measurement status, i.e. the limit value conditions continue to be evaluated even if the measured value is faulty (invalid).

A measured value is invalid and is correspondingly identified as such when:

- The amplifier input range is exceeded
- A calibration is ongoing
- The factory calibration is faulty
- TEDS is used in the setting, if present, and the setting stored in TEDS cannot be implemented (e.g. incorrect transducer model, measuring range not present, invalid filter value, etc.)
- The TEDS content cannot be read correctly or no TEDS is present even though TEDS needs to be set

Digital inputs

Must be switched to +Ub (PLC logic). An open input will be detected as LOW by the internal Pull-Down resistance.

11.8 Signal runtimes

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

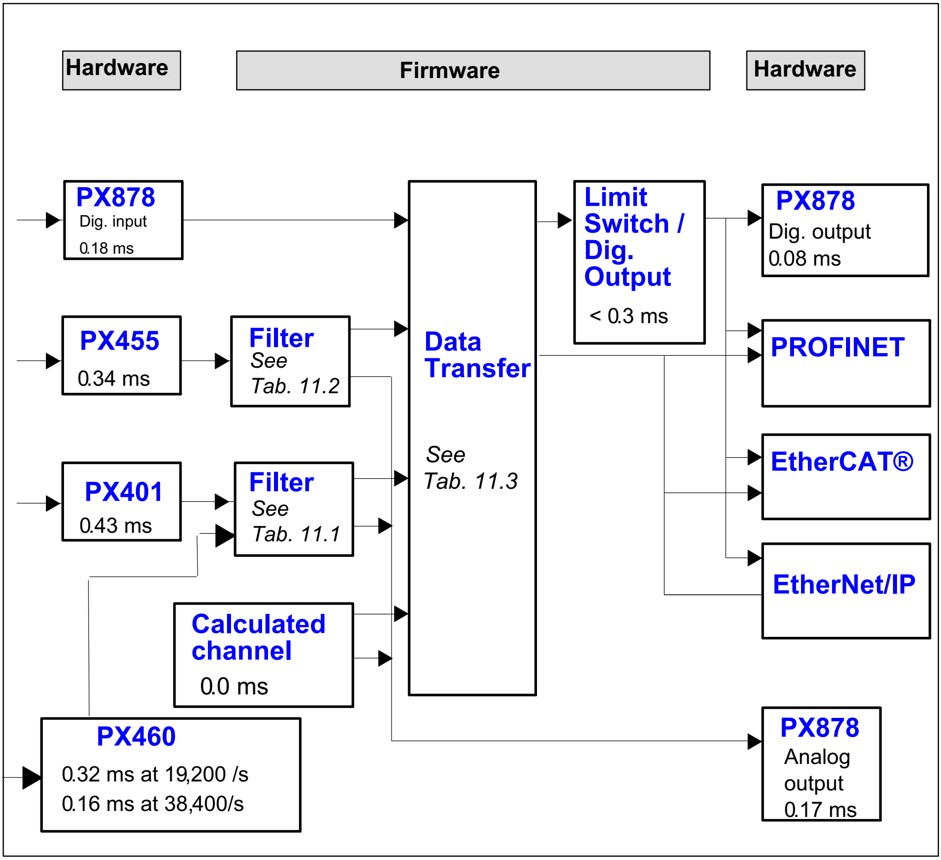


Fig. 11.1 Signal runtimes

Filter group runtime (ms)

Cut-off frequency fc [Hz] (-3dB)	Runtime [ms]	
	Bessel	Butterworth
off	0	0
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

*Tab. 11.1 Runtimes for **PX401** and **PX460***

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

Cut-off frequency fc [Hz] (-3dB)	Runtime [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 Runtimes for ***PX455***

Data transfer rate [Hz]	minimum [ms]	typical [ms]	maximum [ms]
1200	0.1	0.52	0.93
2400 (factory default)	0.1	0.31	0.52
4800	0.1	0.21	0.31
9600	0.1	0.16	0.21

Tab. 11.3 Data runtimes

Example

Signal runtime of a sensor signal via the analog output with filter:

Signal path:

PX455 → 2 kHz Bessel→ PX878
0.34*) + 0.16 (Tab. 11.3) + 0.17*) ms = 0.67 ms

*) See diagram on Page 151

Fieldbus

Delay until signal appears in cyclical data frame.

Protocol	Data Copy Rate [Hz]	Typical [ms]	Maximum [ms]
PROFINET	1200 (default and max.)	1.8 + frame_cycle /2	2.4 + frame_cycle
EtherCAT	2400 (default) 4800 9600 (max.)[1)]	1.0 + frame_cycle /2	1.5 + frame_cycle
EtherNet/IP	1200 (default and max.)	1.8 + frame_cycle /2	2.4 + frame_cycle

Tab. 11.4 Fieldbus runtimes

1) The EtherCAT data copy rate only has minor effects on the signal runtime. The advantage between copy rates of 2.4 and 9.6 kHz is theoretically 0.16 ms, which is significantly smaller than the statistical spread.

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

Example

Signal runtime of a sensor signal via the EtherCAT fieldbus:

Signal path:

PX455 → 2 kHz Bessel→ Data transfer @2.4 Hz→ EtherCAT@2.4 kHz PX01EC

0.34*) + 0.16 (Table 1.2) + 0.31 ms + 1.2 ms = 2.0 ms (mean signal runtime from input terminal to EtherCAT fieldbus

*) See diagram on Page 151

11.9 Fieldbus integration

Important

Make certain you are using the correct device description file (see chapter 16.1)

11.9.1 PROFINET connection

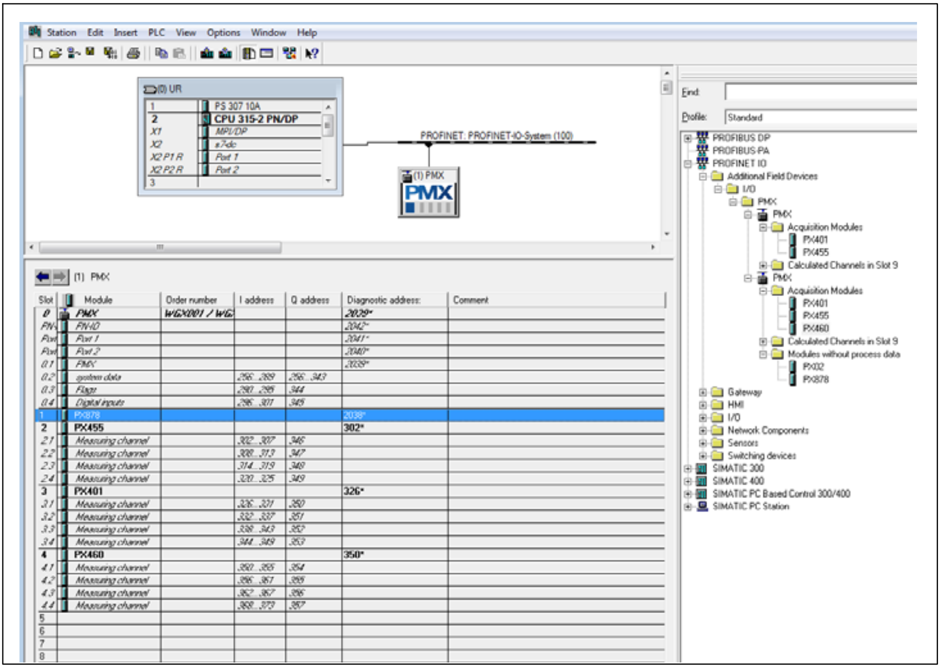
Connect to the PROFINET network with an Ethernet cable.

- Connect the PMX device(s) and the PROFINET master via (Cat5) Ethernet cables (follow the topology).
- When using the PROFINET-IRT protocol keep in mind the correct sequence of the PMX RJ45 sockets:
  Port 1 (bottom), Port 2 (top)

  For this purpose, it is essential to activate the IRT protocol in the PLC configuration software and to specify the cable lengths and the cable ports.
  See also page 58, Kapitel 8.2.4.
- When using the PROFINET-IRT protocol, follow the PMX sequence (Port1 (bottom) / Port2 (top) of the RJ45 sockets on the PX01PN), *also see Chapter 8.2.4*.
- The bus does not need any termination resistors, as active nodes are involved. The device description file (GSDXML) is available for configuring the PMX in the master. This is on the PMX System CD or is available as a download from www.hbm.com/en/menu/support/software-firmware-downloads/industrial-amplifiers .

- The PROFINET system is configured using the engineering tool of the PROFINET Master supplier.

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



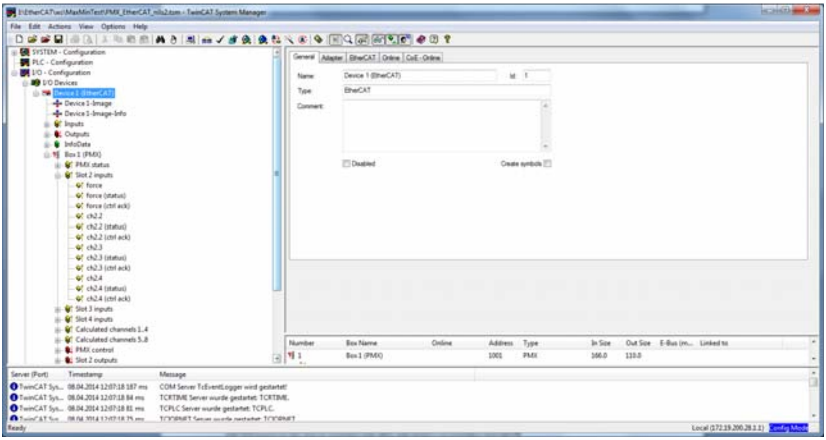
11.9.2 EtherCAT® connection

Connect to the EtherCAT® network with an Ethernet cable.

► Use (Cat5) Ethernet cables to connect the PMX device(s) and the EtherCAT® master – follow the topology (IN (bottom) / OUT (top) of the RJ45 sockets on the PX01EC).

- The bus does not need any termination resistors, as active nodes are involved. The device description file (HBM PMX XML) is available for configuring the PMX in the master. This is on the PMX System CD or is available as a download from www.hbm.com/en/menu/support/software-firmware-downloads/industrial-amplifiers
- The EtherCAT® system is configured using the engineering tool of the EtherCAT® master supplier.

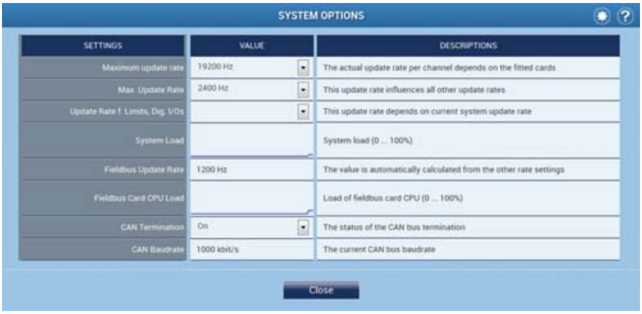
Example with a Beckhoff PLC with the TwinCAT System Manager



11.9.3 Adjusting the fieldbus update rate

► Set the user level to Administrator in the overview top right.

► Select the menu SETTINGS -> SYSTEM -> DEVICE -> SYSTEM OPTIONS: "Internal data transfer rate." . The fieldbus update rate will follow this value up to the fieldbus-specific maximum.



► The change is effective immediately. To save permanently, click on the floppy disk symbol.

11.9.4 EtherNet/IP connection

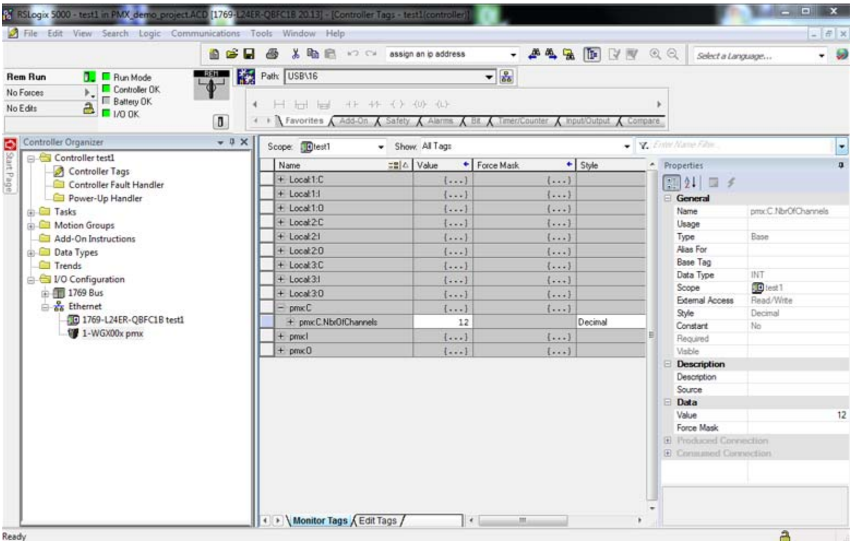
Connect to the EtherNet/IP network with an Ethernet cable.

► Connect the PMX device(s) and the EtherNet/IP scanner via (Cat5) Ethernet cables.

- The bus does not need any termination resistors, as active nodes are involved. The EDS file is available for configuring the PMX in the scanner. This is on the PMX System CD or is available as a download from www.hbm.com/en/menu/support/software-firmware-downloads/industrial-amplifiers

- The EtherNet/IP system is configured using the engineering tool of the EtherNet/IP scanner supplier.
- Both Ports P1 and P2 have the same IP address and MAC address.

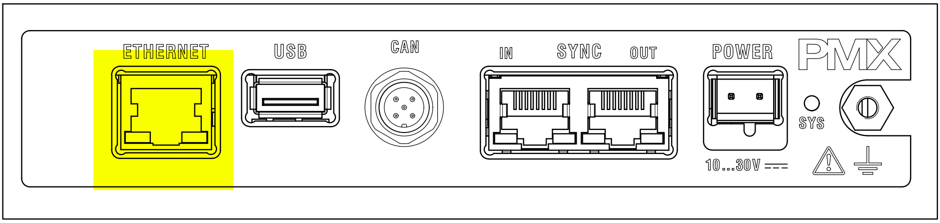
Example with Allan-Bradley SPS ControlLogix and LogixStudio



12 Quick start

12.1 Preparing the measurement system

1. Connect the PMX to a PC via the Ethernet socket



Cable: Standard Ethernet cable (CAT5)

2. Connect your transducers to the measurement cards (plug terminals) *see Chapter 8.4 to 8.7*

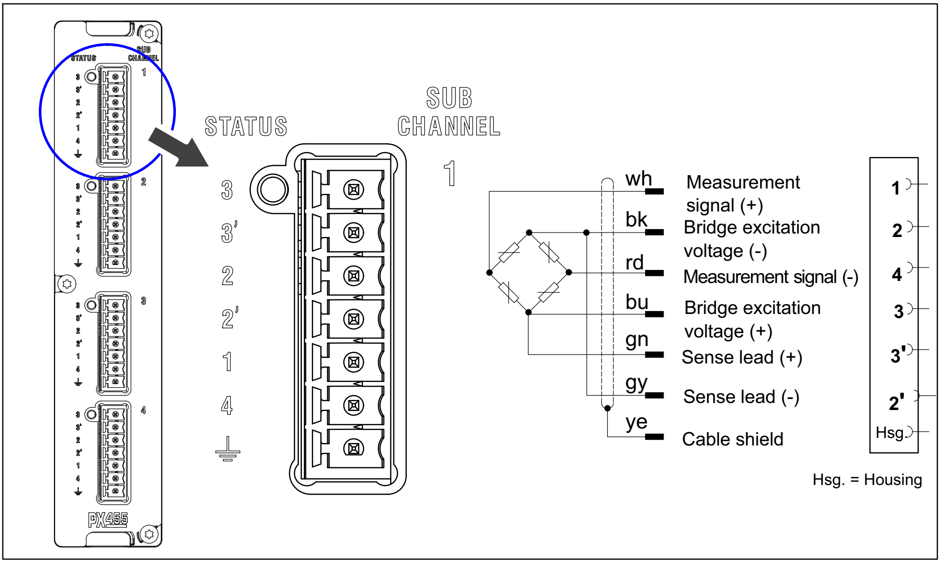


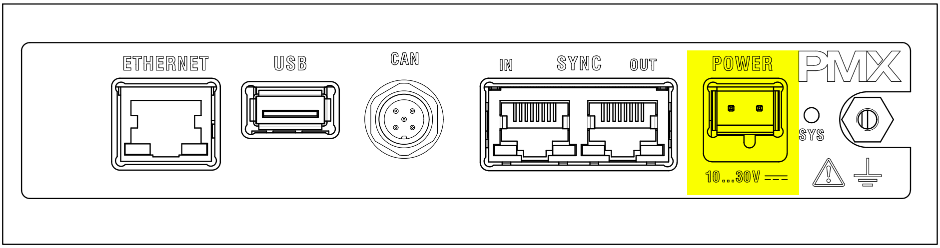
Fig. 12.1 Example: Force transducer / load cell on PX455

Notice

The transducers can also be connected if you have previously connected the voltage supply.

3. Connect the voltage supply (10 … 30 V DC) *see Chapter 8.3*

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



At least 15 W of power must be supplied.

4. Connect the PMX to a PC (HOST)
see Chapter 11.3

The PMX is set to DHCP (automatic address assignment) at the factory. Set the PC (HOST) to DHCP as well. Automatic adjustment and setting of the IP addresses will then follow. This process takes a few seconds.

Call the PMX web server by entering "PMX" in the browser bar and press RETURN.

The PMX announces itself with a 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 language of the PMX web server.

If several PMX devices are available in the network, this selection box will also appear:

► Check the box for the required PMX

► Click **Connect**

The Flash function allows the device to be identified by flashing all the device LED´s.

5. Configure the system with the web server

► Click on the user icon to go to the service or administrator level. Depending on authorization, you can make the following settings

- Assign sensors
- Assign units
- Set filter
- Monitor maximum and minimum values
- Monitor limit values
- Set up virtual (calculated) channels
- Configure digital and analog inputs / outputs
- Create and administer parameter sets

Notice

Clicking on the floppy disk symbol saves the settings/ changes power failsafe.

Security prompt

To get additional help, click the Help symbol .

This opens the web server Help.

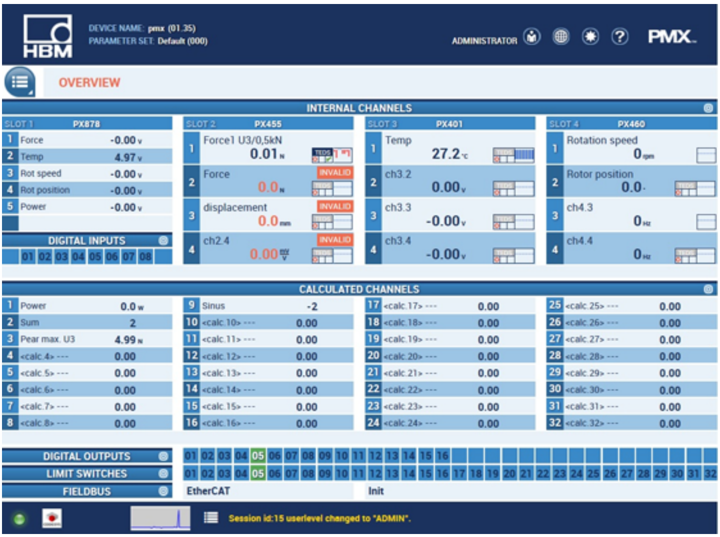
12.2 Typical operating procedure

12.2.1 Measurement example

The easiest way to configure the PMX measuring amplifier and its measurement channels is via the PMX web browser. The sensors, Ethernet cable and voltage supply must be properly connected (*see Chapter 8.4 and 8.3*).

Connect the PMX to a PC (HOST) (*see Chapter 11.3*) to see the device overview.

The entire device, with all measurement cards and signals, as well as all device information, is displayed here.



► Switch to the **ADMINISTRATOR** level (may be password-protected), then via the menu button to **Settings/ Amplifier**.

Alternatively, clicking on the required channel or function (e.g. Limit values) directly calls up the corresponding settings menu. This requires authorization for the respective user levels.

Suitable sensor and channel settings can be made here for each module (slot) and each channel.

Click the slot numbers to select the measurement cards:

orange = selected measurement card, blue = measurement cards present in the PMX, gray = empty module slot (slot).

Example of force sensor at slot 3.1

In the example above, module 3 is fitted with a PX455, and the first channel is assigned to a force sensor (SG full bridge).

- The PMX amplifier channel is set to the full bridge sensor type, with the measuring range of 4 mV/V.
- Amplification is set to 1,000 N, for a sensor sensitivity of 2 mV/V. If the sensor has TEDS sensor detection, the amplifier channel can be parameterized automatically. Activation takes place on the 2nd page of the applicable amplifier settings.

- The filter type is Bessel, the filter frequency is set to 5 Hz (average to high attenuation).
- The data are now changed in the PMX and are displayed by a floppy disk symbol in the status line .
- Press this symbol to store the settings power failsafe in the PMX (security prompt).

Example: Configuring the PMX with strain transducer

Example: SL*B700A*

Slot 2 is fitted with a PX455, its third channel is assigned to the strain transducer.

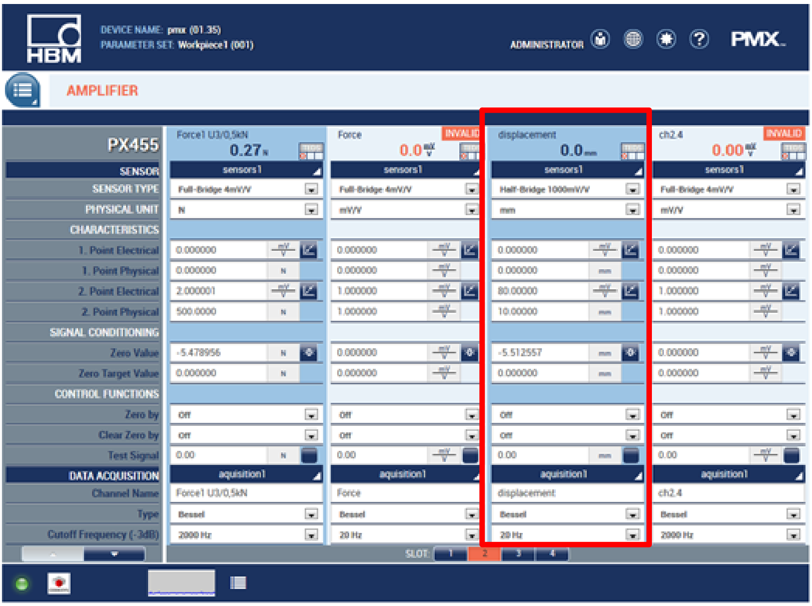
- The PMX amplifier channel is set to the full bridge sensor type, with the measuring range of 4 mV/V.
- Amplification is set to 500 µm/m, with a sensor sensitivity of 1.5 mV/V.
- The filter type is Bessel, the filter frequency is set to 5 Hz (average to high attenuation).

Example: Configuring the PMX with displacement transducer

Example: *WI/10mm*

Slot 2 is fitted with a PX455, its third channel is assigned to the displacement transducer.

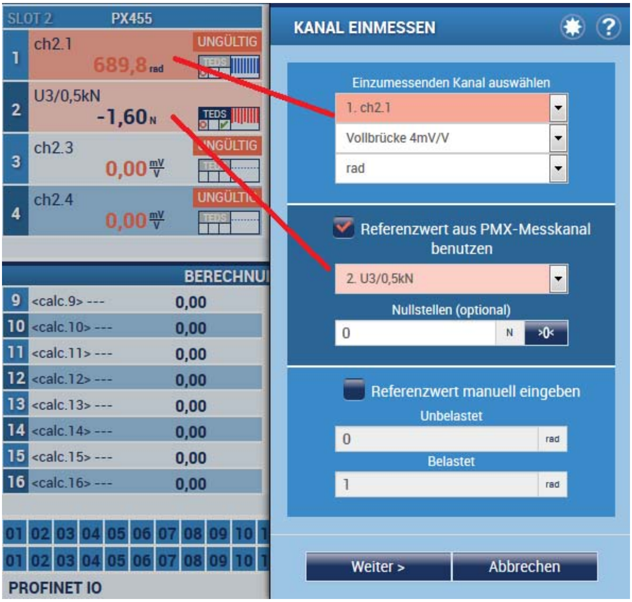
- The PMX amplifier channel is set to the inductive half bridge sensor type, with the measuring range of 100 mV/V.
- Amplification is set to 10mm, with a sensor sensitivity of 80 mV/V.
- The filter type is Bessel, the filter frequency is set to 20 Hz.



12.3 Adjustment assistant

Firmware Version 2.04 and later provides an adjustment assistant, to make it easier to adjust non-calibrated sensors. With PMX as the measuring chain, this uses simple, menu-driven operation to help with adjusting up to 4 sensors with the PMX as a measuring chain, that are only installed in the machine on site and are operated in a force shunt. A measurement channel in the PMX or an external reference measuring chain can be used as the reference sensor.

Step 1: Selecting the adjustment channel and the reference channel



Step 2: Adjusting the measuring chain in the unloaded and loaded state

Tip

The adjustment process should be repeated several times to increase accuracy.

Step 3: After the adjustment processes, the channel settings can be applied in the PMX adjustment channel by clicking the "Apply" button. Clicking "Cancel" restores the initial state.

12.4 Update software (PMX web server)

The configuration software (PMX web server) is implemented in the PMX, and does not have to be installed (*see Chapter 11* for more information).

To operate current versions of the PMX devices, you need a terminal (e.g. PC / tablet with a mouse) with a common Internet browser (Internet-Explorer (Version > 9.0), Firefox or Chrome) and a screen resolution of at least 1024 x 768.

A new version of the web server is a component part of the PMX firmware, and they are installed together, in a firmware update (*see Chapter 22*).

The firmware update is carried out in the PMX web browser in the **Firmware update** menu.

For more support, see the web browser online Help.

Tip

You can download the latest firmware at www.hbm.com/en/menu/support/software-firmware-downloads/industrial-amplifiers

13 Internal calculation channels

A total of 32 calculated channels is available in the PMX for use as required. They can be used for any purpose and several times. The calculations can be internally linked to other calculated channels.
Once the calculation is complete, the signal can be exported to one of the 32 free computing channels at the end of the calculation.
It is then available for display and further use on the analog output, fieldbus and Ethernet interface.
The calculated channels are processed in the sequence in which they are listed. This sequence can be changed any time.

Important

Please make sure that the CPU load resulting from the calculated channels does not exceed 100%; otherwise, individual values will be lost. For this purpose, if necessary, reduce the calculated channels' output rates.
The CPU load generated by the calculated channels is displayed in the PMX web browser's status bar and in the "System options" menu.
In addition, the information is available in the PMX system status and can be signaled via fieldbus, Ethernet and digital output.

When the object directory is being used, the object list changes dynamically as computing channels are added, changed or deleted. Then the header files have to be recreated and the programming has to be adapted via fieldbus or PC control.

13.1 Calculation rate

The following applies for all modules, unless otherwise specified:

Calculation rate	same as measuring rate (default 19,200/s)
Value range of floating point values	Simple floating point resolution as per IEEE 754 Range approx. +/-$3.4*10^{38}$

13.2 Description of calculations

13.2.1 Scaling

13.2.1.1 Two-point scaling

Function	Linear scaling of a signal *0ut0 = m * in= + b* where *m = y2 − y1 / x2 - x1* *b = y2 − m * x2*
Inputs	*in0*
Outputs	Scaled signal *out0*
Parameter	- Two **interpolation points** (x1|y1) and (x2|y2) x: Input values y: Output values
Default	*x0=y0=0; x1=y1=1*; therefore m=1, b=0
Exception handling	In the case of a range overrun, Not-a-Number is output. E.g. where x1 = x2:

13.2.1.2 Characteristic curve table (21 interpolation points)

Function	Non-linear characteristics
Inputs	Characteristic curves-Input *in0*
Outputs	Characteristic curves-Output *out0*
Parameter	- **Number** of interpolation points used (2 .. 21) *nbrPoints* - up to 21 **interpolation points** (x0|y0) … (x20|y20)

Default	nbrPoints: 2 points 0 and 1: (-1000|-1000) (1000|1000)
Exception handling	If the input is marked as invalid (Invalid Bit), then the output is also marked as invalid. The output value is still determined and output.

Suitable for e.g. Limitation of values or e.g. Value formation

Transitions in the characteristic curve are permissible, i.e. x1=x2 is possible for example. The output jumps at this point from y1 to y2.

If input < *x0* , then output *out0* = *y0.*

If output is > the largest used point x_n, then output *out0* = y_n

13.2.1.3 Polynomial 4th order

Function	Polynomial 4th order y = a0 + a1*x + a2*x^2 + a3*x^3 + a4*x^4 (when x >= 0 OR OneSet = true) y = b0 + b1*x + b2*x2 + b3*x3 + b4*x4 (when x < 0) With the OneSet option, coefficients A are applied for negative values of x, too.
Inputs	Input in0 (x)
Outputs	Output out0 (y)
Parameter	- Set A: a0, a1, a2, a3, a4 - Set B: b0, b1, b2, b3, b4 - One Set (boolean): true: Coefficients set A is applied for all values of x. Set B is ineffective. false: Set A is applied for x >= 0, else set B.

Default	in0: Constant 0.0 a0, a1, a2, a3, a4: 0.0 b0, b1, b2, b3, b4:0.0 OneSet = true
Exception handling	If the input is marked as invalid (Invalid Bit), then the output is also marked as invalid. The output value is still determined and output. In the case of a range overrun, Not-a-Number is output and the output is marked as not valid.

13.2.1.4 Taring

Function	Rapid zero setting or taring, controlled via digital input.
Inputs	- Input *in0*, input with untared original measured value. - Tare value (tareValue), input with value, applied during taring to the output. Tare value = 0 equivalent to zero setting. - Trigger (level controlled): at level = high the output is set to the tare value. - Reset (level controlled): at level = high the zero setting/taring is cancelled. Reset overloaded trigger.
Outputs	Tared value *out0* = *in0* - internal offset
Parameters	none

Default	in0: Constant 0.0 *tareValue*: Constant 0.0 Trigger: Invalid Reset: Invalid
Exception handling	If the input is marked as invalid (Invalid Bit), then the output is also marked as invalid. If the input is invalid, the trigger is ineffective. The tared value is not checked for validity (it is normally a constant signal that is always valid).

Fast taring is an alternative to permanently installed sensor taring (zero setting):

	Sensor taring	Fast taring
Availability	Always available in the amplifier settings	Only as calculated channel
Offset (Difference input-output)	Known and selectable by user	Unknown, not selectable
Effectiveness	Acts directly at sensor on the original measurement signal	Original measurement signal is not influenced
Trigger	Fieldbus (runtime >= 40 ms*) or digital input (< 14 ms*) or user interface	Digital input (0.17 ms*)

* Typical values with default settings

13.2.1.5 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
Inputs	6 Inputs *in1 .. in6*
Outputs	6 outputs *out1..out6*
Parameters	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	*in0, in1, in2, in4, in5, in6:* Constant 0.0 Matrix = identity 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 or a number of inputs is marked as invalid (Invalid bit), all outputs are also marked as invalid. The computation is nonetheless done and output.

13.2.1.6 SG stress analysis

Function	Biaxial stress analysis with strain gage rosette A, B, C (0° / 45° / 90° or 0° / 60°/ 120°). The temperature can be compensated by either of two options: **Option A:** By a separate strain from a gauge that shall have the same temperature as A, B, C, but shall be free from strain. **Option B:** By the apparent strain that comes from a polynomial, provided that the temperature reading is available. ε_{actual}: strain used for stress calculation ε_{read}: strain read from gauges A, B, or C in µm/m ε_T: strain read from gauge T in µm/m ε_p: apparent strain from polynomial in µm/m Do not use options A and B at a time. If the strains A, B, C come already compensated, use neither of the options. Unused inputs must be connected to constant 0.0. **Zero setting capability** The strains A, B, C don't need to be set to zero. This function block provide a set-to-zero trigger input, that clears all outputs. Setting to zero may happen before each measurement or once in a lifetime. The result is stored with the other settings and will restore after starting up the device.
Inputs	- Strain gauge A (0°, reference of angle) **in µm/m** - Strain gauge B (45° / 60°) **in µm/m** - Strain gauge C (90° / 120°) **in µm/m** - Strain gauge T (optional), temperature compensating strain gauge **in µm/m** - Apparent strain ε_p from polynomial **in µm/m** - Reset, a digital input that clears the outputs except the angle phi

Outputs	- sigma1: Principal stress 1 [unit follows unit of elastic modulus] - sigma2: Principal stress 2 [unit follows unit of elastic modulus] - phi: Angle of sigma1 based on axis of strain gage A [degrees] range 0..180° - tau: Maximum shear stress [unit follows unit of elastic modulus] - sigma_v: von Mises stress (= equivalent tensile stress) [the unit follows unit of elastic modulus] $\sigma_v=\sqrt{(\sigma_1^2+\sigma_2^2-\sigma_1 x \sigma_2)}$
Parameters	- Elastic modulus (E, Young's modulus), the unit (e.g. N/mm²) determines the unit of the output signals - Poisson's ratio (v, greek nu) - Strain gauge type. Either 0/45/90° or 0/60/120°
Default	strain gages A, B, C, T: Constant 0.0 E = 200,000 N/mm² Poisson's ratio v = 0.3 - Strain gauge 0/45/90°
Exception handling	If one or a number of inputs is marked as invalid (Invalid bit), the outputs are also marked as invalid. The calculation is nonetheless done and output.

13.2.2 Evaluation functions

13.2.2.1 Filter (high pass, low pass, CAMSA)

Function	Filters a signal
Inputs	Signal to be filtered *in0*
Outputs	Filtered signal *out0*
Parameters	- **Cut-off frequency** fc (damping -3 dB) - **Characteristics**: Bessel or Butterworth - **Type**: Low pass or high pass
Default	*fc:* off (filter ineffective) Bessel, low pass
Exception handling	If the input is marked as invalid (Invalid Bit), then the output is also marked as invalid. The filtering is still implemented and output. If the range is overrun, Not-a-Number is output.

Frequencies and group runtimes

Low pass

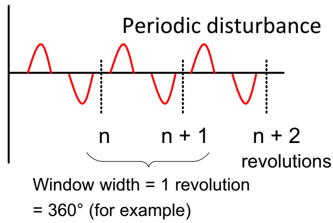
fc [Hz]	**Bessel** Runtime [ms]	**Butterworth** Runtime [ms]
off	0	0
3000	0.13	0.19
2000	0.21	0.30
1000	0.43	0.61
500	0.86	1.23
200	2.00	3.10
100	4.15	6.17
50	8.45	12.5
20	21.4	30.7

10	39	47
5	74	91
2	174	216
1	340	430
0.5	680	840
0.2	1680	2090
0.1	3360	4200

High pass

fc [Hz]	**Bessel** Runtime [ms]	**Butterworth** Runtime [ms]
off	0	0
100	0	0
10	0	0

13.2.2.2 Angular synchronous filter

Function	The filter eliminates periodic disturbances that are synchronous to the rotation of a shaft. It performs a moving average. The filter does not work time synchronous as usual. It rather works synchronously to the shaft rotation.Thus the filter effect does not depend on the rotation speed. The filter's calculation period is proportional to the rotation speed. When the rotation stops, the filter stops operating as well.  The filter width can cover 30° to 720°.
Inputs	X: the signal to be filtered The shaft angle in degrees [0°, 360°]
Outputs	Y: the moving average of input X
Parameters	Detection mode - The width 30°…720° - The resolution 1°…8° The smaller the resolution, the smaller the maximum rotation speed (see below). Note: The ratio width/resolution must not be greater than 180
Default	width = 180° resolution = 1°
Exception handling	-

Filter width	Disturbances of order are eliminated (the rotation speed is of order 1)
90 º	4,8,12, …
120 º	3, 6, 9, …
180 º	2, 4, 6, …
360 º	1, 2, 3, …
720 º	0.5, 1.0, 1.5, …

The filter’s calculation period cannot be faster than the calculation cycle (see chapter1).

The theoretical maximum rotation speed depends on the filter resolution and the calculation cycle:

Resolution	N_max at 19,200/s = resolution * 19,200/6	N_max at 38,400/s = resolution * 38,400/6
1º	3,200	6,400
2º	6,400	12,800
4º	12,800	25,600
6º	19,200	38,400
8º	25,600	51,200

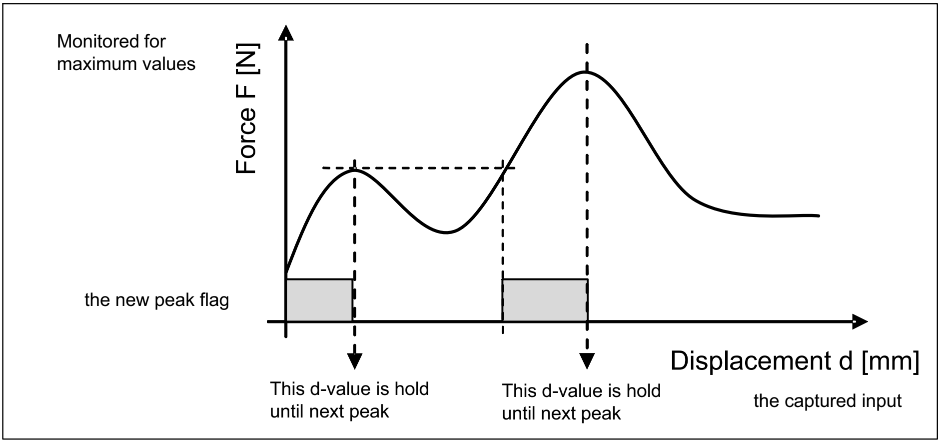
It is recommended that the actual maximum speed is much smaller (factor 5 - 10) than the theoretical maximum speed.

13.2.2.3 Extreme Value (Peak)

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 time constant, you will obtain an envelope function. Additionally, a new peak is signalled by an output flag. The flag only is high in the cycle, in which a new peak was detected. When the input signal increases constantly, the output flag of a maximum peak block is permanently high. Reset via (virtual) digital input or Flag (level-controlled) Reset via fieldbus (event-controlled) Hold via digital input (level-controlled) This function block may deliver wrong values after its creation or after switching of the parameter set. It is strongly recommended to reset the extreme value before starting the process. The maximum or peak-to-peak value decreases with the discharge rate. The minimum value increases with the discharge rate. Small discharge rates can cause numerical issues. Please compare the discharge rate with the expected peak values. Discharge rate > 5% of peak value: No issue expected. Discharge rate 1% ... 5% of peak value: Please check whether the value discharges as expected. Discharge rate < 1% of peak: Critical. The value may not discharge as expected.
Inputs	- Input 1 - Input 2 - Hold with - Reset by

Outputs	Extreme Input 1 Input 2 at Extr.1 New extreme value
Parameters	- **Type** [Maximum, Minimum, Peak-to-Peak] - **Inversion of** the hold input - Discharge rate [unit/second]
Default	Type: Maximum Digital inputs: Not in use Inversion: off Discharge rate : 0.0
Exception handling	If the input is marked as invalid (Invalid Bit), then the output is also marked as invalid. The extreme value is still determined and output. When the Input 2 is marked as invalid (invalid bit), the value is captured nonetheless and also marked as invalid.

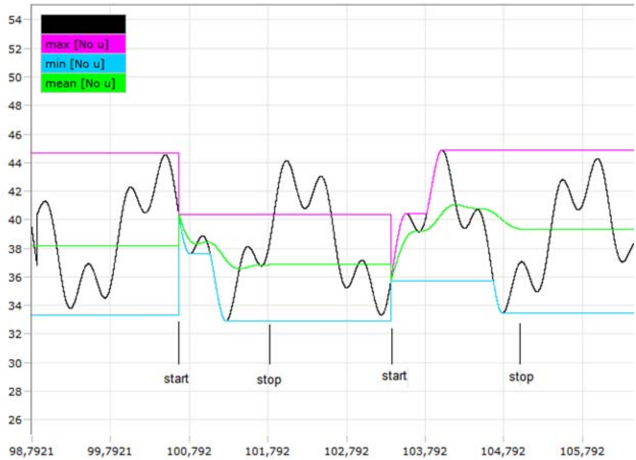
Example: Capturing displacement when $F = F_{max}$:



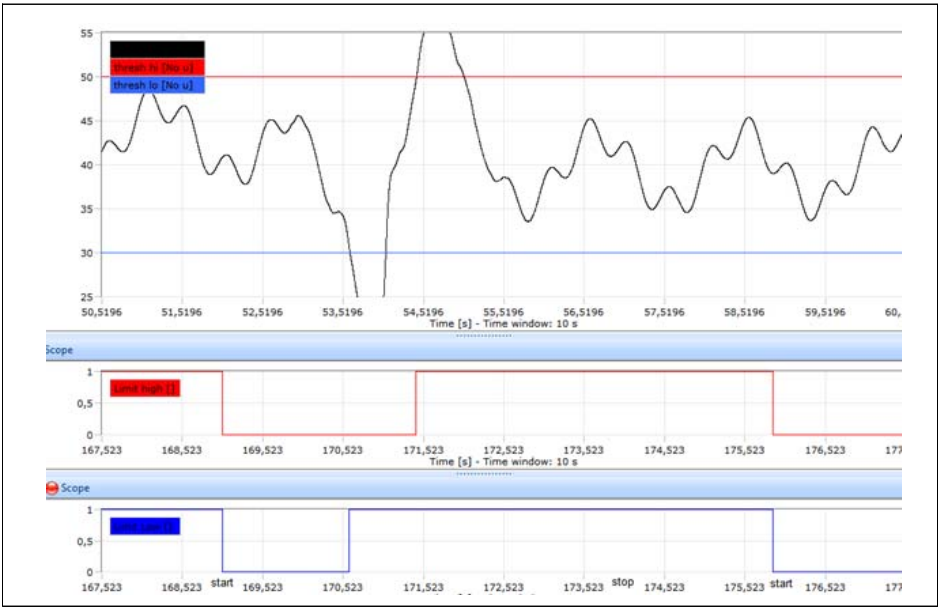
13.2.2.4 Tolerance window

Function	A check is performed to ensure that a tolerance band defined by the "Threshold Hi" and "Threshold Lo" inputs is present. Two flags indicate that user-defined limit values have been exceeded. The maximum, minimum, peak-to-peak and mean value are also generated. The process is started and stopped by edge-sensitive digital signals. A second input signal is captured when new peak (min or max) was detected. All outputs are reset at the start. All outputs are held after stop until the next start. The outputs are continuously updated during operation. The mean value output can accumulate a maximum of 100,000 measurements. This is equivalent to a recording time of 5.2 s at 19,200 measurements/s. (All these functions can also be performed with the other function blocks.)
Inputs	Input 1 Input 2 Start : Starts the measurement Stop : Ends the measurement Upper threshold: After x has exceeded this value, the "Limit Switch Hi" output is set until the next start. Lower threshold: After x has fallen below this value, the "Limit Switch Lo" output is set until the next start.

Outputs	**Max(x)** since start. **Min(x)** since start. Bei Max(x) erfasster Wert des zweiten Eingangssignals Bei Min(x) erfasster Wert des zweiten Eingangssignals The capture function is ilustrated in section 13.2.2.3.
Parameters	Start falling edge (startFallEdge): - true: Start monitoring on falling edge - false: Start monitoring on rising edge Stop falling edge (stopFallEdge): - true: Stop monitoring on falling edge - false: Stop monitoring on rising edge
Default	Start digital input: invalid Stop digital input: invalid Start falling edge: false Stop falling edge: false
Exceptions handling	If the input is marked as invalid (Invalid bit), the peak value output and mean value output will not be updated and their Invalid bits will be set until the stop condition is fulfilled. The values become valid again with the next start signal. The time output is not adversely affected by an invalid input signal. The capture input (Inputsignal 2) is not checked for cvalidity. If the time exceeds 100,000 samplings, the mean value is no longer updated and is marked as invalid.



Example of max., min. and mean value. Reset when starting Hold from stop to start.



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

13.2.2.5 Hold function (analog)

Function	Holds an input value controlled by a measured value (or a calculated channel) Hold condition: When the control input is inside (or optionally outside) the *[threshLow, threshHigh]* interval. When the hold condition is met, the input in0 is output to the output out0 (including a status bit). When the hold condition is not met, the output is held at the last hold value and may be made invalid.
Inputs	- Input *in0* - Control input *control* - Digital Init-input (controls initialization, high-active)
Outputs	Hold value *out0*
Parameters	- **Upper threshold** of hold range *threshHigh* - **Lower threshold** of hold range *threshLow* - **Hold outside** *holdOutside* (yes/no), hold outside the interval *[threshLow, threshHigh]* - **Only on entry in the interval** *onEntryOnly* (yes/no) **yes:** The value is hold on entry in the interval (if applicable, after expiry of the delay) and then held up to the next entry. **no**: The output is constantly updated (if applicable, after the delay has expired), as long as the hold condition has been met. - **Delay** in ms. Range 0.0 … 60000.0 ms (= 1 min); Resolution 0.1 ms The value is only held when the hold condition remained active without interruption for this period - **Invalid outside** *invalidOutside* (yes/no) The output value is marked as invalid as long as the hold condition is not met - **Initial value** *initValue* Is output when the edge is positive at the digital Init input

Default	Upper threshold: 0.0 Lower threshold: 0.0 Outside hold: no Only on entry in interval: no Delay: 0.0 ms Outside invalid: no Initial value: 0.0
Exception handling	If the input *in0* is marked as invalid, the hold function is still implemented. The output is then also marked as invalid during hold. If the control input is invalid, the hold function is not implemented. The current output value is held.

13.2.2.6 Hold (digital)

Function	Holds the input value Controlled by a digital input
Inputs	- Input *in0* - Digital hold input (edge-controlled) - Digital reset input (only effective if it cannot be re-triggered)
Outputs	Held value *out0*
Parameters	- **Inversion** of the hold input (yes/no) yes: Hold at high -> low no: Hold at low -> high - **Re-triggerable** (yes/no) yes: The output is updated with each edge of the hold input no: The output is only updated with the first edge after reset - **Hold delay** in ms. Range 0.0 … 60000.0 ms (= 1 min); Resolution 0.1 ms The value is only held when the Hold input remained active without interruption for this period

Default	Inversion hold: no Re-triggerable: yes Hold delay: 0.0 ms
Exception handling	If the input is marked as invalid (Invalid Bit), then the output is also marked as invalid. The hold function is still implemented and output.

13.2.2.7 Mean values (arithmetic, RMS)

Function	Provides the mean value of a signal over time. The period is determined by trigger events. The output is not updated continuously, it is rather updated at trigger events only. The trigger event is determined by either 1.When reaching a given number of acquired samples 2.The control input becomes greater than the threshold 3.The control input becomes smaller than the threshold 4.Rising edge at the digital input 5.Falling edge at the digital input
Inputs	in0: The signal to be averaged control: The control input that is compared to the threshold input. Only relevant with trigger events 2, 3. threshold: The threshold input. Only relevant with trigger events 2, 3. trigger input: The digital trigger input. Only relevant with trigger events 4, 5
Outputs	mean: Either the arithmetic or the quadratic mean count: The number of samples that have been averaged

Parameters	trigger event: see above number of samples: Number of samples to be averaged. Samples acquisition rate is the Calculation Rate, see above. Only relevant with trigger event 1. number of triggers: The mean value is calculated and updated after this number of trigger events. Only relevant with trigger events 2…5. root mean square: false: The arithmetic mean is output true: The quadratic mean (root mean spare) is output maximum samples count: The ‘mean’ output becomes invalid when this number of samples has been acquired without having detected a trigger event. Maximum of this parameter is 100,000.
Default	in0: Constant 0.0 control: Constant 0.0 threshold: Constant 0.0 trigger input: invalid trigger event: 2 number of samples: 2 number of triggers: 1 root mean square: false maximum samples count: 100,000
Exception handling	The ‘mean’ output becomes invalid when number of samples exceeded maximum samples count. In this case, the ‘count’ output reads the maximum number of samples. ‘mean’ becomes valid at next trigger detection. The first output mean value after initialization or after a samples overrun is usually not a plausible value. The output becomes sensible after the second trigger. Invalid values at inputs in0, control, and threshold are not detected and thus do not affect the validity of the ‘mean’ output.

13.2.2.8 Moving Average

Function	A moving average for an assigned number of samplings. The filter settling time is the number of samplings multiplied by the computation cycle (default 1s/19200, see section 1).
Inputs	X: : the signal being averaged
Outputs	Y: : the moving average of input X
Parameters	The number of samplings (filter width) Range 1...385
Default	Number of samplings: p. 385
Exception handling	If the input is marked as invalid (invalid bit), the filter is deactivated and the output is blocked and marked as invalid. Invalid input values are not fed into the filter. If the input becomes invalid, the filter is reactivated and the output immediately becomes valid again. A filter settling time is required until the filter buffer is filled with sequential samplings.

The moving average is an efficient option for eliminating narrow band noise, for example from the power supply (50 or 60 Hz).
Width = 19200 Hz/50 Hz = 384 eliminates 50-Hz noise (and the harmonics at 100, 150, 200, 250, etc. Hz)

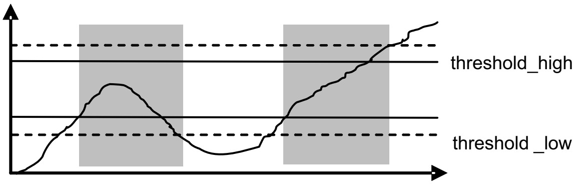
Width = 19200 Hz/60 Hz = 320 eliminates 60-Hz-noise (and the harmonics at 120, 180, 240, 300... Hz) ...where 19200 is the sampling rate.

This function requires a lot of internal memory. Generally it is not possible to create more than 6 or 7 blocks of moving averages, or even fewer if other blocks have already been created.

If there is not enough memory for a new function block, a message is logged.

13.2.2.9 Trigger function (range)

Function	Provide a digital trigger signal dependent on an input value. The trigger gets high (low with isHighActive = false) when threshold_low <= input <= threshold_high The trigger gets low (high) when input < (threshold_low – hysteresis) OR input > (threshold_high + hysteresis) threshold_high threshold_low

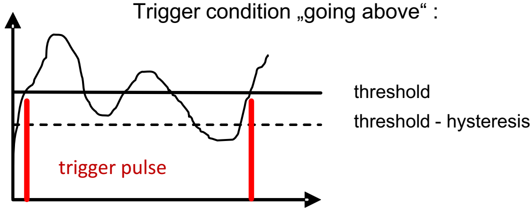
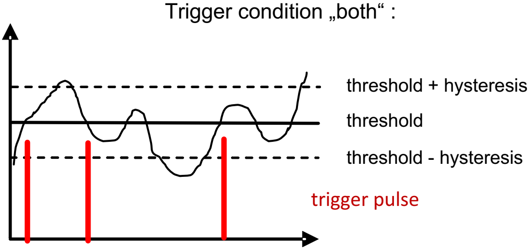


Inputs	input: A value from a sensor or a calculated channel threshold high: A value from a sensor or a calculated channel threshold low: A value from a sensor or a calculated channel
Outputs	the trigger signal, output in a flag
Parameters	hysteresis: Effective when the input leaves the threshold span delay in ms: Trigger gets high (low) not before the input signal was delayed ms inside the threshold span. delay is not effective when leaving the threshold span. isHighActive: If yes, the trigger output is active-high, else active low.
Default	input: Constant 0.0 threshold high: Constant 1.0 threshold low: Constant 0.0 hysteresis: 0.0 delay: 0.0 ms isHighActive: yes
Exception handling	If the input is marked as invalid (Invalid bit), the trigger gets low (high). The threshold inputs are not checked for validity.

Parameters	Interval [s]: Maximum 100,000 s. Single Shot (On/Off) Pulse Length [s]: Defines the high-time of the Elapse Flag. Maximum 100,000 s. When equal to zero, the Elapse Flag gets high for one calculation cycle.
Default	Run input: Constant high Interval: 1 s Single Shot: Off Pulse Length: 0.1 s
Exception handling	None. The output is always valid.

13.2.2.10 Trigger function (pulse)

Function	Provide a digital trigger signal dependent on an input value. The trigger gets high (low with isHighActive = false) when threshold_low <= input <= threshold_high The trigger gets low (high) when input < (threshold_low – hysteresis) OR input > (threshold_high + hysteresis) threshold_high threshold_low

	Trigger condition „going above“ :  Trigger condition „both“ : 
Inputs	input: A value from a sensor or a calculated channel threshold high: A value from a sensor or a calculated channel threshold low: A value from a sensor or a calculated channel
Outputs	the trigger signal, output in a flag
Parameters	hysteresis: Effective when the input leaves the threshold span delay in ms: Trigger gets high (low) not before the input signal was delayed ms inside the threshold span. delay is not effective when leaving the threshold span. isHighActive: If yes, the trigger output is active-high, else active low.

Default	input: Constant 0.0 threshold high: Constant 1.0 threshold low: Constant 0.0 hysteresis: 0.0 delay: 0.0 ms isHighActive: yes
Exception handling	If the input is marked as invalid (Invalid bit), the trigger gets low (high). The threshold inputs are not checked for validity.
Parameters	Interval [s]: Maximum 100,000 s. Single Shot (On/Off) Pulse Length [s]: Defines the high-time of the Elapse Flag. Maximum 100,000 s. When equal to zero, the Elapse Flag gets high for one calculation cycle.
Default	Run input: Constant high Interval: 1 s Single Shot: Off Pulse Length: 0.1 s
Exception handling	None. The output is always valid.

13.2.3 Mathematical functions

13.2.3.1 Adder

Function	Adds four summands, each weighted with one factor each $out0 = in0 * m0 + in1 * m1 + in2 * m2 + in3 * m3$
Inputs	Summands in0 .. in3
Outputs	Sum *out0*
Parameters	**Factors** m0 .. m3
Default	*in0* .. *in3* connected with constant 0.0 *m0* .. *m3* = 1.0
Exception handling	If one or more inputs are marked as invalid (Invalid Bit), then the output is also marked as invalid. The calculation is still implemented and output. If the range is overrun, Not a Number (+/- inf) is output.

13.2.3.2 Multiplier

Function	Multiplies *four* signals
Inputs	Inputs *in0, in1*, in2, in3
Outputs	Product *out0*
Parameters	
Default	*in0, in1, in2, in3* connected with constant 1.0
Exception handling	If one or more inputs are marked as invalid (Invalid Bit), then the output is also marked as invalid. The calculation is still implemented and output. If the range is overrun, Not-a-Number (+/- inf) is output.

13.2.3.3 Divider

Function	Division y = Dividend / Divisor
Inputs	Inputs *dividend, divisor*
Outputs	Quotient *out0*
Parameters	
Default	*dividend*: Constant 1.0 *divisor*: Constant 1.0
Exception handling	If one or more inputs are marked as invalid (Invalid Bit), then the output is also marked as invalid. The calculation is still implemented and output. If the range is overrun, Not-a-Number (+/- inf) is output.

13.2.3.4 Counter

Function	Counts transitions of a digital signal. Maximum count 2^{32} -1. Count mode is either positive edge, negative edge, or both edges. A digital input enables/disables counting. A flag is set when the count exceeds a defined value. The count is cleared when a defined period without countable transitions has elapsed.
Inputs	input: The digital signal to be counted enable: A digital signal that enables the counter reset: A digital signal that clears the counter (level sensitive)
Outputs	count: The current counter value flag: Set when counter >= match value

Parameters	count mode: positive edge / negative edge / both edges timeout in ms: Resets the counter when no countable edge has been detected. Maximum 50 x 10 ms. 0 ms means the timeout is disabled A value that controls the flag
Default	input: Constant low enable: Constant high reset: Constant low count mode: Positive edge timeout: 0 ms (= no timeout) flag control value: 1
Exception handling	None. The output is always valid.

13.2.3.5 Integrator

Function	Integrates a signal up to the user defined limits.
Inputs	In: The signal to be integrated Reset Flag: Resets the output value to the value at input Init Value. This flag is level sensitive. While being high, the Init Value is output. Init Value: The value at this input is applied with the Reset Flag.
Outputs	Out: The integrator value
Parameters	Integration Time [s]: Defines the time, after which the output equals a constant input (step function applied at t=0). Resolution = 1/Calculation Rate, see above. Y Max, Y Min: The limits of the output. While limited, the integrator does not wind up in the background.

Default	In: Constant 0.0 Reset Flag: Constant low Init Value: Constant 0.0 Integration Time: 1 s Y Max: 1e6 Y Min: -1e6
Exception handling	While the input is marked as invalid (Invalid bit), the integration is stopped and the output is marked invalid as well.

13.2.3.6 Differentiator

Function	Provides the derivative of a signal X, Y = delta_X / delta_T. delta_T = 4 calculation cycles = 4/Calculation Rate, see table below. It is recommended to limit the bandwidth of the input signal.
Inputs	input X
Outputs	output Y
Parameters	Y Max, Y Min: The limits of the output
Default	Input X: Constant 0.0 Y Max: 1e6 Y Min: -1e6
Exception handling	While the input is marked as invalid (Invalid bit), the calculation is stopped, the last good output value is held, and the output is marked invalid.

Calculation Rate	-3dB at	Transfer function = 0 at
19,200/s (default)	3.6 kHz	4.8 kHz
38,400/s	7.2 kHz	9.6 kHz

13.2.3.7 Cartesian to polar coordinates

Function	Conversion of a point (x,y) to polar coordinates radius r and angle Θ (theta). Θ is output in degrees in the interval [-180°, 180°] . To convert to radian measure multiply with π/180. (1.0) → Θ = 0° (-1.0) → Θ = 180°(0.0) → Θ = 0°, r = 0
Inputs	X-coordinate Y-coordinate
Outputs	Radius r, same unit as x,y Angle Θ (Theta) in degrees, interval [-180°, 180°]
Parameters	none
Default	X-coordinate Y-coordinate
Exception handling	If one or a number of inputs is marked as invalid (Invalid bit), the output is also marked as invalid. The calculation is nonetheless done and output.

13.2.3.8 Polar coordinates to Cartesian coordinates

Function	Conversion of polar coordinates radius and angle to a point (x,y).
Inputs	Angle in degrees, interval (-360°, 360°) Radius r
Outputs	X-Koordinate = r*cos(angle) Y-Koordinate = r*sin(angle)
Parameters	none

Default	Angle: Konstant 0.0 R: Constant 0.0
Exception handling	If one or a number of inputs is marked as invalid (Invalid bit), the output is also marked as invalid. The computation is nonetheless done and output.

13.2.3.9 Modulo function

Function	Calculates the remainder of the division input X/divisor
Inputs	X
Outputs	remainder
Parameters	divisor
Default	input X: Constant 0.0 divisor: 1.0
Exception handling	While the input is marked as invalid (Invalid bit), the calculation is stopped, the last good output value is held, and the output is marked invalid.

13.2.3.10 Constant signal

User-defined constant signal.

13.2.4 Technology functions

13.2.4.1 2-state controller

Function	2-state controller with feedback
Inputs	- Input *in0* target value (*setpoint*) - Input *in1* actual value (*feedback*)
Outputs	Flag out0, (Menu Digital outputs, "Calculated Channel Flag")
Parameters	- **Hysteresis** - **Feedback-amplification** (Kr) Common amplification of both parallel PT1 feedback paths - **Feedback-time constant** 1 (Tr1) [seconds] PT1 time constant of negative feedback path Tr1 < Tr2 - **Feedback-time constant 2** (Tr2) [seconds] PT1 time constant of positive feedback path Tr1 < Tr2
Default	Hysteresis: 1.0 Tr1, Tr2: 1e38 (the feedback paths are effectively switched off) Kr: 0.0 (feedback paths are switched off)
Exception handling	If at least one of the inputs is invalid (invalid bit), then - the controller output is switched off (low) - the values of the feedback paths are held

If there is doubt regarding the use of the feedback paths, they should be left in the default settings.

13.2.4.2 PID controller

Function	Quasi-linear PID controller in parallel structure with Anti-Windup $$Kp\left(1 + \frac{1}{Ti * s} + \frac{Td * s}{Tp * s + 1}\right)$$ Tp is the parasitic time constant, see below
Inputs	- Input *in0* target value (setpoint) - Input *in1* actual value (feedback) - Digital enable input (enableId)
Outputs	Controller output *out0*
Parameters	- **Amplification** Kp, P-proportion - **Reset time** Ti [seconds], I-proportion - **Derivative time** Td [seconds], D-proportion - **Upper limit** of controller output ymax - **Lower limit of controller** output ymin - **Default output** default is output, when enable input = low
Default	*Kp* = 0.0 *Ti* = 1e38 s *Td* = 0.0 s *ymax* = 1e20 *ymin* = -1e20 *enableId* = constant High, i.e. constantly active *default* = 0.0
Exception handling	If at least one of the inputs is invalid (invalid bit), then - the output is frozen and also marked as invalid - the controller is stopped

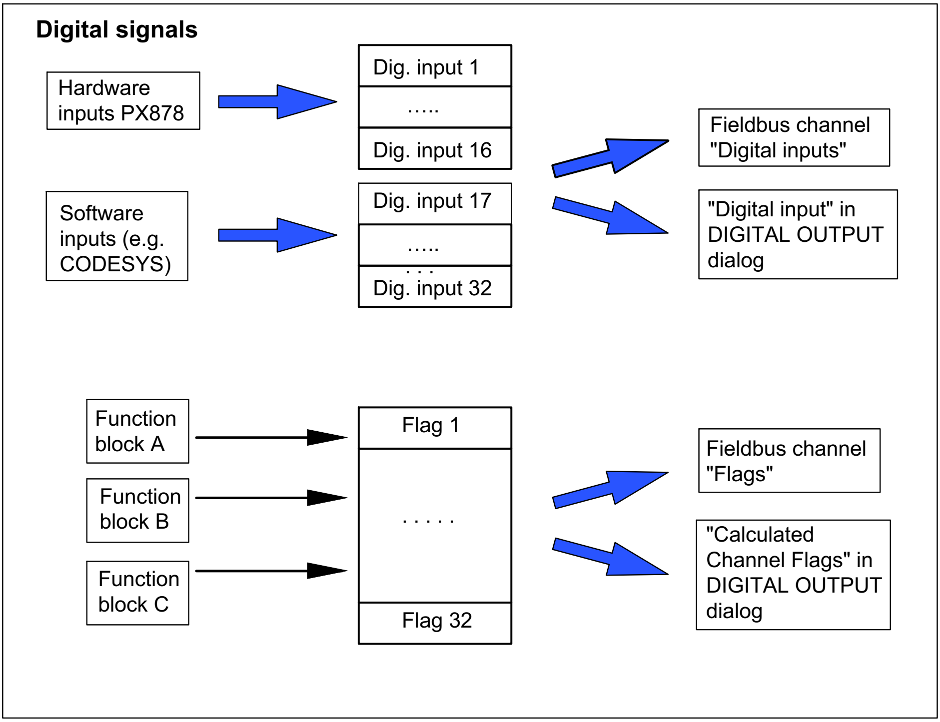
Due to recursive calculations, not suitable as a purely P-controller. There should always be an I-proportion present.

Parasitic time constant $Tp = \frac{1}{calculationrate}$

$$Tp = \frac{1}{19200}s = 52\,\mu s \; Default$$

13.2.4.3 RTD PT100 at PX455

Function	Temperature measurement with a RTD Pt100 resistor connected at PX455. Range -100°C ... +500°C. A fixed 100 Ohms precision resistor must complement the RTD to form a half bridge. The complement resistor is connected at the positive bridge supply. The complement resistor can be connected directly at the PX455 front clamps without wiring. **The details are outlined in a PMX Tech Note.**
Inputs	The half bridge signal in mV/V
Outputs	Temperature in °C
Parameters	R_wire [Ohms]: The resistance of the wire to the RTD (single distance). This parameter must be set, when the complement resistor is connected without wires directly at the PX455 clamps. If the complement resistor is connected with the same wire length as the RTD 100, then the wire resistances are compensated by the bridge layout and R_wire must be zero.
Default	R_wire = 0 Ohms
Exception handling	If the input is marked as invalid (Invalid bit), the operation stops and the output gets frozen and is marked as invalid. When the temperature is out of range (-100 .. +500°C), then -333.3°C is output and the output is marked as invalid.



Tip

Examples regarding calculation channels can be found in the Technical notes on the supplied System CD and in http://www.hbm.com/en/menu/support/software-firmware-downloads/industrial-amplifiers/

13.2.4.4 Signal generators (rectangle, triangle, sine)

Function	Generates a periodic or non-periodic signal
Inputs	Enable bit (digital input) Starts the generation. When low, the generator stops and outputs the offset value. When enable is high again, the generator does not continue where it stopped but starts with a new period.
Outputs	Generates signal *out0*
Parameters	- **Wave form** [sine, rectangle, noise, counter, constants, ramp] - **Frequency** 0 .. Data rate/4 (default 19200/4 Hz = 4800 Hz) (only effective with sine, rectangle, triangle) - **Amplitude** - **Offset** - **Periods** Number of periods to be played. Only effective with periodical waveforms. Zero means endless operation. Maximum value 2^{31}-1. When the periods have been played, the offset value is output. For restart, the enable input must toggle to low, then high. The generation stops immediately when the enable input gets low, even when the periods were not fully played.
Default	Sine, amplitude 1.0, offset 0.0, Frequency 100 Hz, 0 periods (=endless) Enable input = constant high
Exception handling	If the range is overrun, Not-a-Number is output.

The **noise signal** is generated from a pseudo-random series of numbers with the period 2^{31}.

The **counter** counts upwards with the data rate (default 19,200 Hz).

Range: -Amplitude to +Amplitude (with offset = 0).

For an acceptable curve form, the frequency should not be set higher than data rate/10.

13.2.4.5 Logic modules (AND, OR…)

Function	Provides either of the logic functions 1 x AND (4 inputs, 1 output) 1 x NAND (4 inputs 1 output) 1 x OR (4 inputs 1 output) 1 x NOR (4 inputs 1 output) 2 x XOR (2 inputs each, 1 output each) 2 x XNOR (2 inputs each, 1 output each) 4 x NOT (1 input each, 1 output each)
Inputs	digital signals A, B, C, D
Outputs	Y1 Y2: only used with XOR, XNOR, NOT Y3: only used with NOT Y4: only used with NOT
Parameters	mode: either AND, NAND, OR, NOR, XOR, XNOR or NOT
Default	inputs: Constant low mode: AND
Exception handling	None

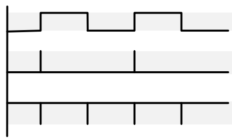
13.2.4.6 Multiplexer 4:1

Function	Multiplexes two inputs to one output, controlled by digital input Control bit 0 0 1 0 1 Control bit 1 0 1 0 1 Output = input 0 input 1 input 2 input 3
Inputs	Inputs *in0, in1*, in2, in3
Outputs	Output *out0*
Parameters	Control inputs *muxBit0Id*, muxBit1Id
Default	*in0, in1, in2, in3* combined with constant 0.0 *muxBits* combined with constant low
Exception handling	The outputs invalid bit follows the currently selected input's invalid bit.

13.2.4.7 Dead time

Function	Holds the output value as long as input and output differ less than the threshold. When the difference output – input is higher than the threshold, the output gets equal to the input.
Inputs	X
Outputs	Y
Parameters	deltaY: The minimum signal change
Default	input X: Constant 0.0 deltaY: 1.0
Exception handling	While the input is marked as invalid (Invalid bit), the calculation is stopped, the last good output value is held, and the output is marked invalid.

13.2.4.8 Edge detector

Function	Detect edges of a digital signal. A transition at the input produces a pulse at the output. A pulse lasts one calculation cycle, see Chapter 1. Detection modes are rising edge, falling edge or both. The output pulse is either high active or low active. This block comprises two detectors A and B, that are independent from each other. Input Rising edge, high active Both edges, low active 
Inputs	Two digital signals in A and in B
Outputs	*Two flag outputs*
Parameters	Detection mode - detect falling edge - detect rising edge - or detect both high Active: If yes, the output is active high, else active low
Default	Input: Constant 0.0 Detection mode: rising edge High active
Exception handling	-

13.2.4.9 Pulse width measurement

Function	Measures the time between two edges at the digital inputs. The period duration of a periodic signal can also be measured. The output is in ms, s or as frequency 1/s. Time resolution: Equal to 1/calculation rate (s.a.) Default 1/19,200 = 52 μs Maximum measurable time: 1/calculation rate * 0x800,000 Default approx. 436 s
Inputs	Digital input Start: Starts the time measurement Digital input Stop: Stops the time measurement If the Start and Stop conditions (input, edge) are identical, the period duration is measured. Otherwise, the pulse length is measured.
Outputs	out0 contains the pulse duration in s (or ms) or the frequency in 1/s

Parameters	Falling edge Start (startFallEdge): - true: Start at falling edge - false: Start at rising edge Falling edge Stop (stopFallEdge): - true: Stop at falling edge - false: Stop at rising edge result type (resultType): 0: Pulse duration output in s 1: Pulse duration output in ms 2: Frequency output in Hz Only recommended for periodic signals.
Default	Digital input Start: invalid Digital input Stop: invalid Falling edge Start: false Falling edge Stop: false Frequency: false result type: 0 (pulse duration in s)
Exception handling	If the pulse or period length is greater than the maximum time, the measurement is stopped, the output marked as invalid and the start conditions is waited for again.

Measurement uncertainty related to the measured value

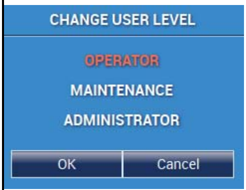
At data rate 19,200/s

Pulse/period duration [ms]	..corresponds to frequency [1/s]	Uncertainty [%]
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

13.2.4.10 Timer

Function	Either single shot or continuously running timer. When elapsed, a flag is set to high level. Maximum interval 100,000 s. Resolution = 1/Calculation Rate, see above.
Inputs	Run (digital): In single shot mode, a low to high transition starts the timer. A high to low transition does not stop the timer. In continuous mode, a high level starts the timer. A low level stops the timer immediately.
Outputs	Elapse flag: Is set to high when the timer elapses In **single shot mode**, the flag gets high for one calculation cycle when the parameter Pulse Length equals 0. Otherwise the flag gets high until the next timer start. In **continuous mode**, the flag gets high for the time defined with parameter Pulse Length. Time [s]: The current timer value. While the timer is stopped, Interval is output.

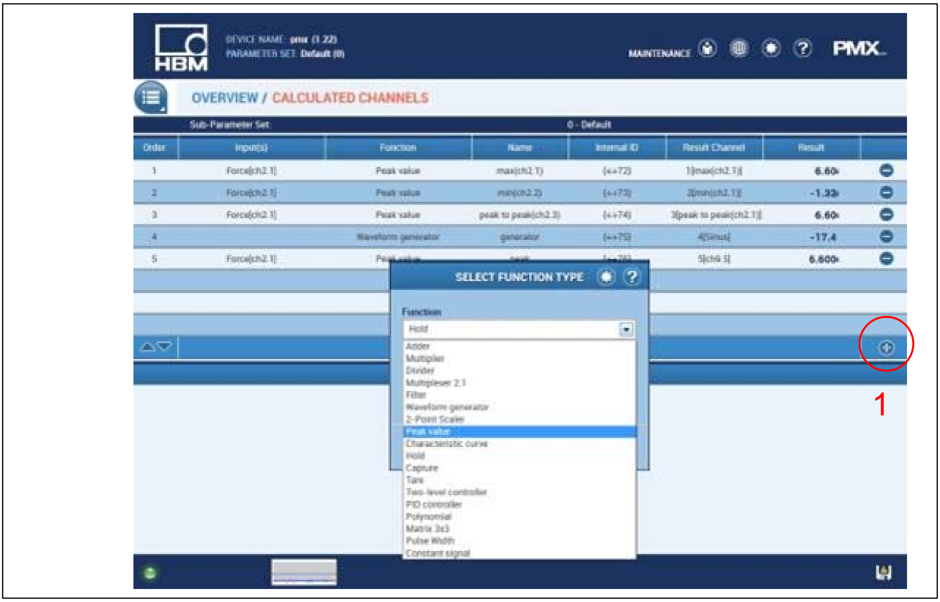
13.2.4.11 Connection with (optional) delay (CODESYS)

Function 	output Y = input X Purpose: Easy connection of a CODESYS channel to a calculated channel output The connection forwards the input signal on the output, optionally with a delay. You can duplicate the input signal with this function to forward it to other functions such as filters. For PMX versions with CODESYS (WGX001), the function is also suitable for forwarding values or signals that have been determined to other channels or outputs.
Inputs	X
Outputs	Y
Parameters	-
Default	Input X: constant 0.0
Exception handling	If the input is marked as invalid (Invalid bit), the output are also marked as invalid. The input value is nonetheless output.

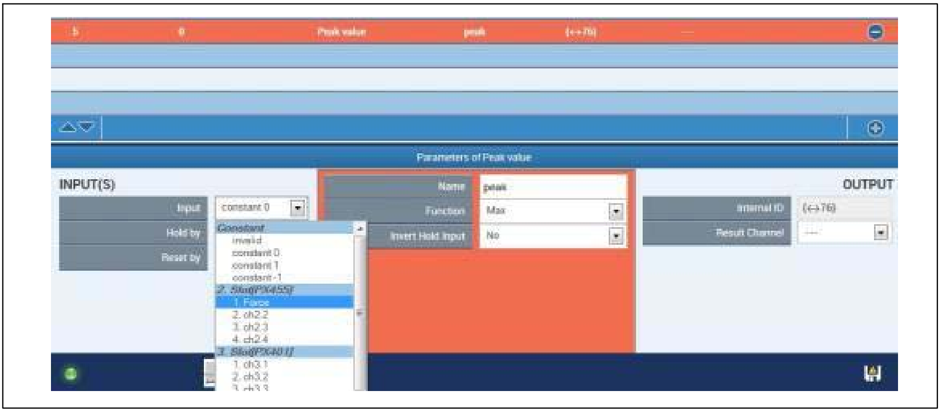
13.3 Example calculations

13.3.1 Generation of peak values

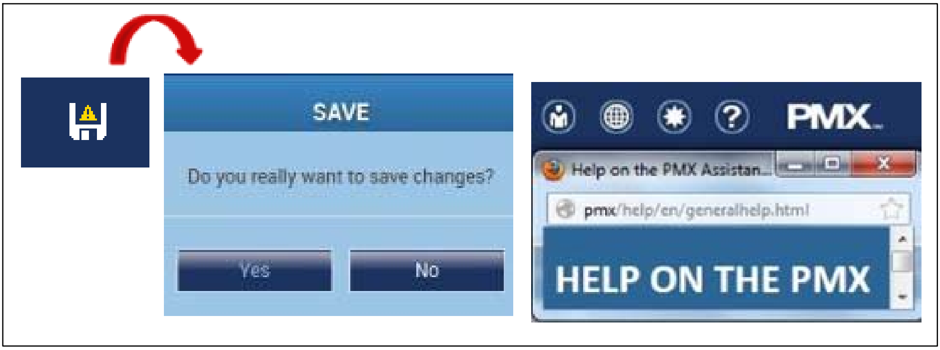
Before starting your configuration, make sure you have the necessary rights to make any changes.
Click the *person symbol* in the top right corner and choose *maintenance* or *administrator*.
Now click > *Settings* > *Calculated channels*.
Click “add function”[1] - you are now able to choose one out of many functions for every single channel. In this example we choose “Peak value” for channel 5.



The next step is to configure your function. Choose an input channel, give it a name, define the output channel, and so on. You will be able to change those settings later anytime; however the primary function cannot be changed. It is very important to define a “Result Channel” 1, as you won’t be able to see any output without it.



Do not forget to save your changes. Simply press the “floppy disk” symbol in the bottom right corner.



Note: If you need help, please use the PMX WebBrowser Help function at the top!

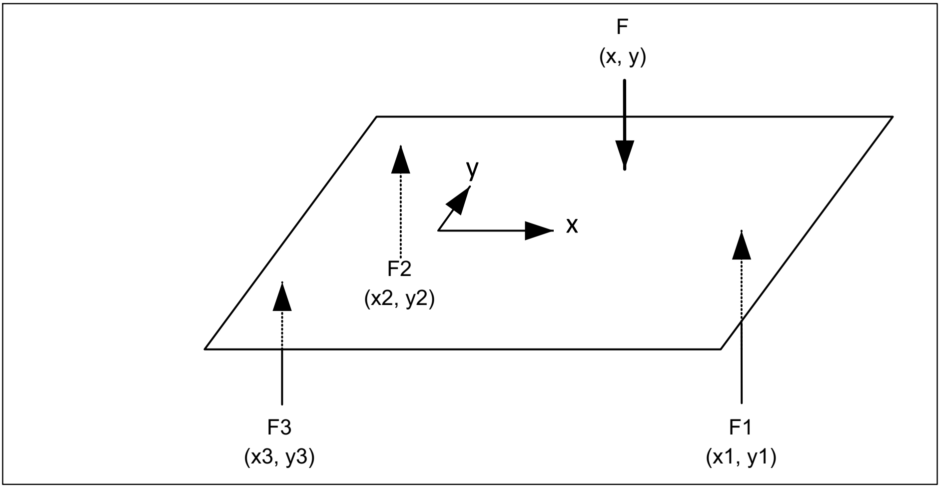
13.3.2 Calculating the force introduction point

Short introduction

Determining the force introduction point on a plate with three force transducers

Introduction

The coordinates of a force F to be measured can easily be determined with three force transducers F1, F2 and F3.



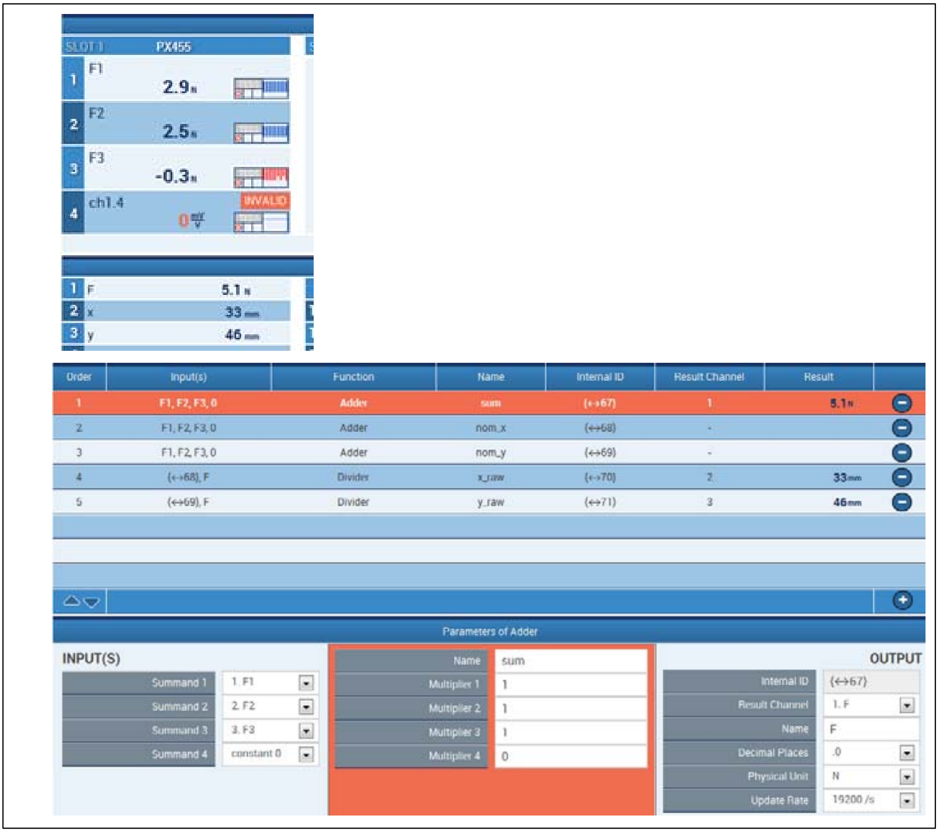
A plate is supported by three transducers and force F is applied against them orthogonally. The point at which the force is introduced is derived from the equilibrium of moments relative to origin

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

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

Procedure

Force F is the sum of the three individual forces:

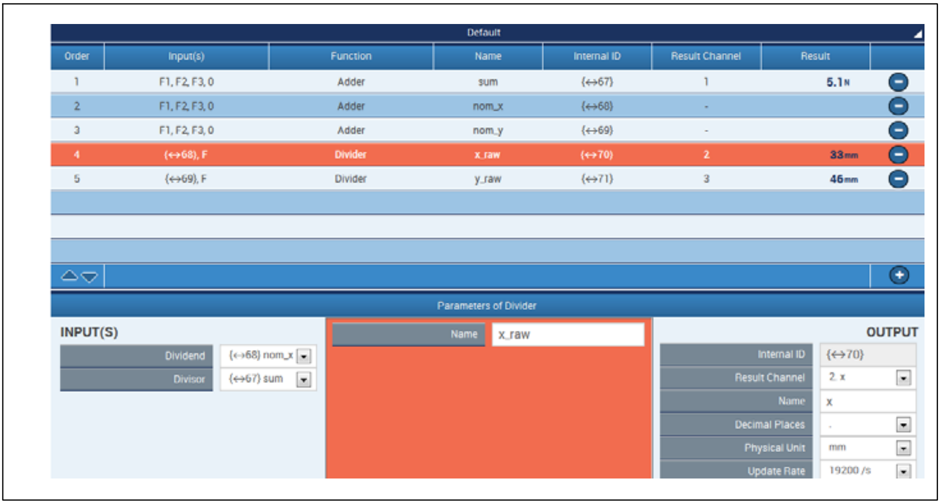


The counters for the x and y calculation are determined in an interim step. Coordinates x1, y1, x2,…. are in the factors of the summands.

For x:

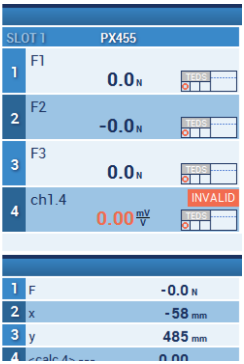
For y:

Finally x and y are calculated with two divisions. The calculation for x is shown here (y is similar):



Implausible values in unloaded state

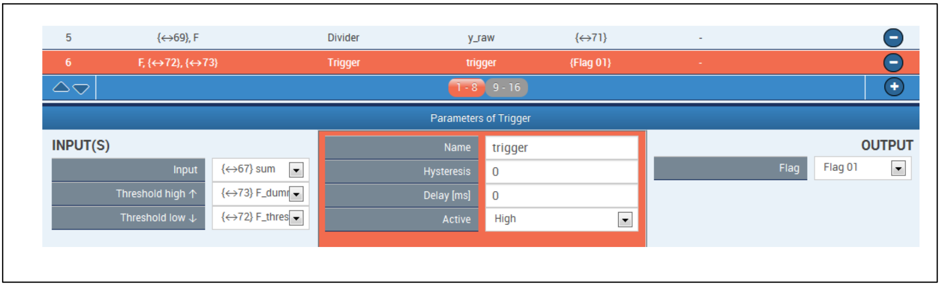
Noise predominates when F is close to zero. Implausible values are returned for x and y:



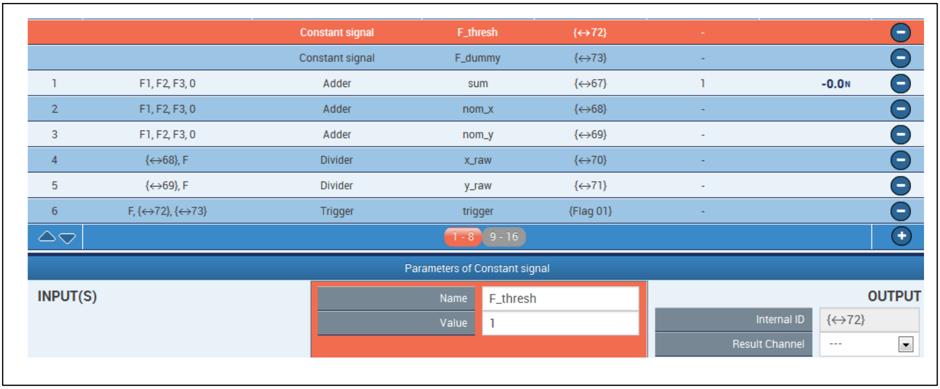
Remedy:

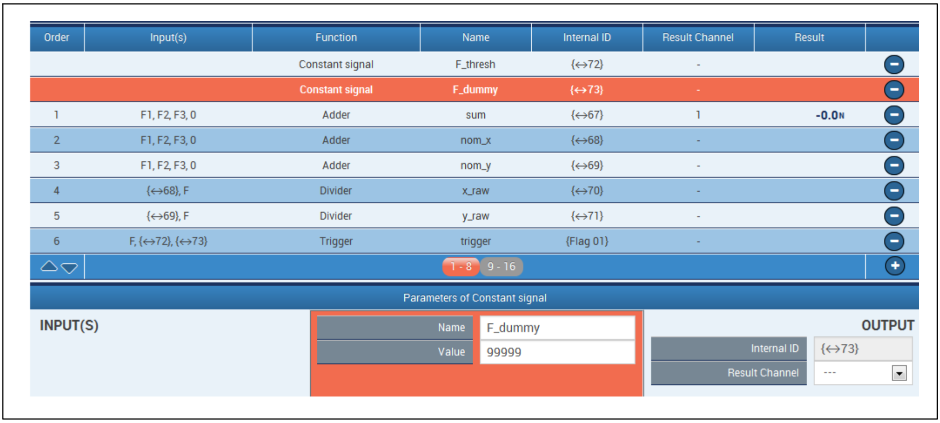
Output for x and y is not regular until F is greater than 1 N, for example. Otherwise zero will be returned each time.

A trigger block sets Flag_01 if F is greater than the minimum value:



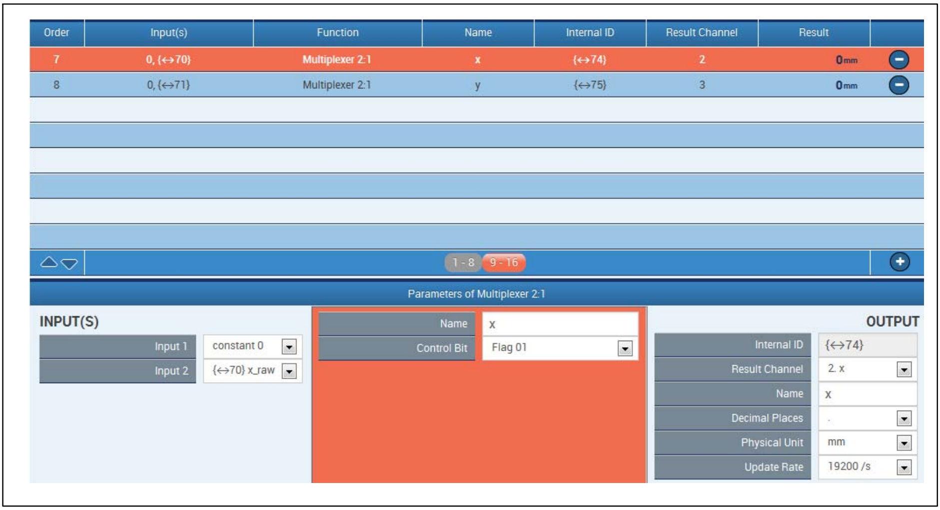
The two limit values for the trigger. Only the lower switching threshold is required for 1 N. A value is selected for the upper threshold that is far above the measuring range:





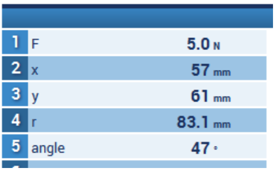
Two multiplexer blocks switch between zero and the calculated values.

Shown here for x

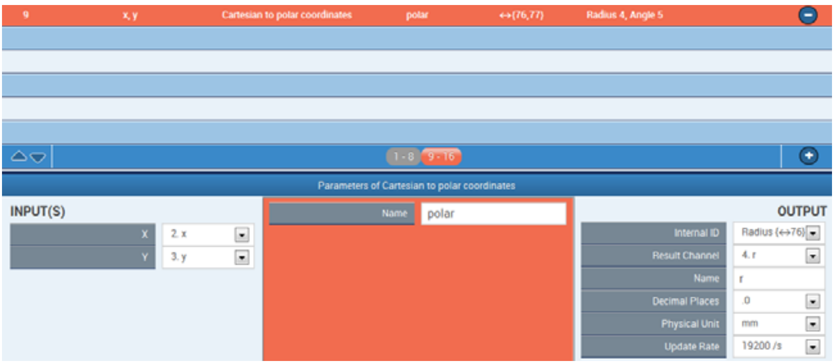


Tips

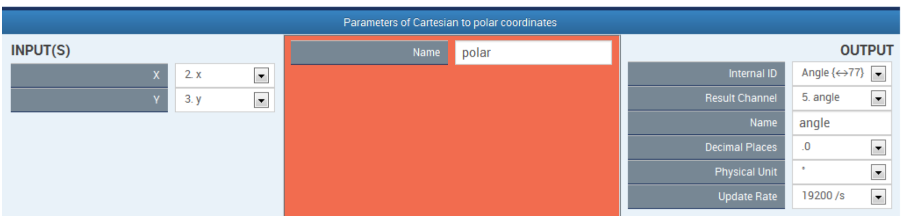
1. In case of division by zero, a divisor block returns Not-a-Number (NaN).
2. Polar coordinates can also be returned if necessary:



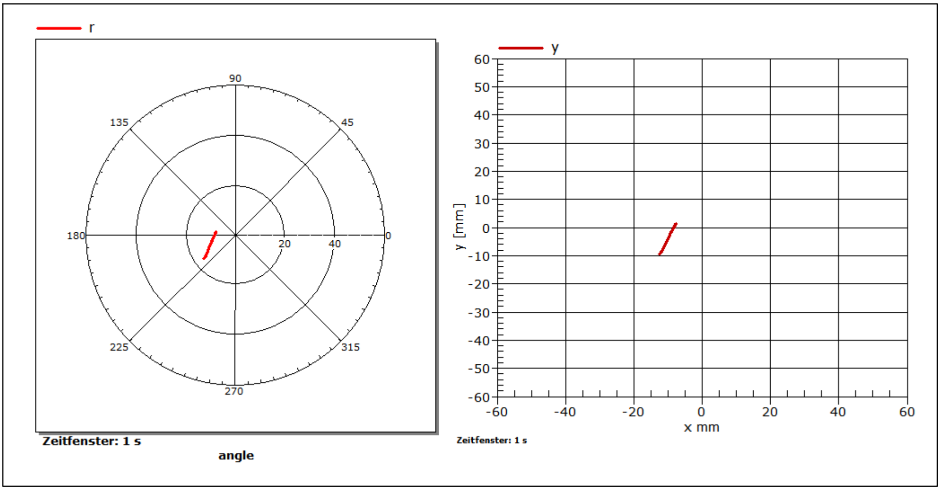
The settings for radius….:



….and angle:



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 measurement work

Introduction

Mechanical work performed W will be measured by integrating force F over displacement s.

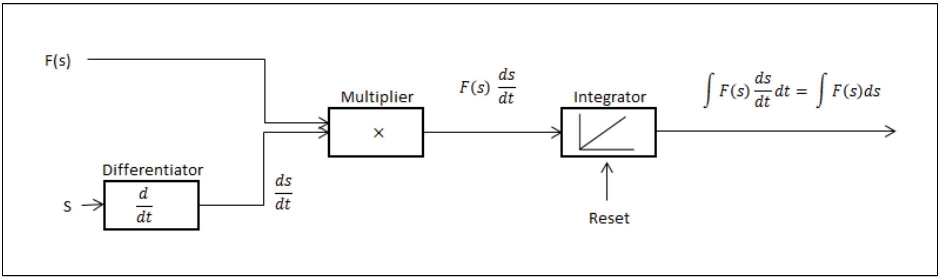
$$W = \int F(s) * ds$$

(Because the processing is time-discrete, this is actually a summation, however the term integration will still be used here.)
The beginning and end of the integration are determined

by measurable events such as fixed displacement or force values or signal edges on a digital input.

Procedure

Integration over displacement is achieved by first deriving the displacement based on the time, multiplying by F and then integrating with time again:

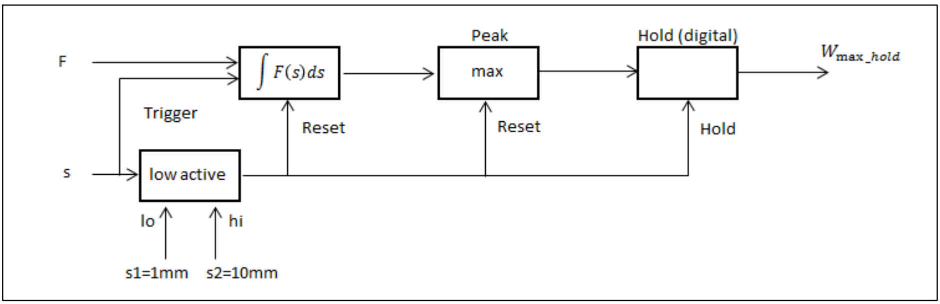


Example A

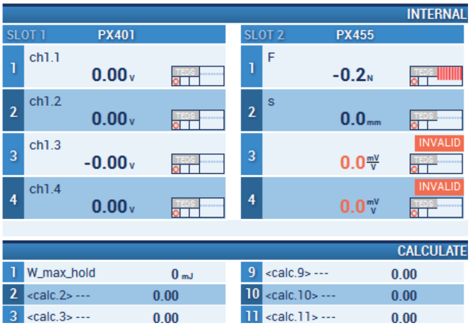
Integration over **displacement from s1 to s2**:

$$W = \int_{s1}^{s2} F(s) * ds$$.

In this case events s=s1 and s=2 determine the beginning and end of the integration. These points on the displacement axis must in any case be reached during the process. Otherwise the beginning or end will not be detected.



Overview with measurands F and s as well as the calculated work



Overview of glyphs:

Order	Input(s)	Function	Name	Internal ID	Result Channel	Result
		Constant signal	s_low	{↔70}	-	
		Constant signal	s_high	{↔71}	-	
1	s	Differentiator	d_s/d_t	{↔67}	-	
2	s, {↔70}, {↔71}	Trigger	gate	{Flag 01}	-	
3	F, {↔67}	Multiplier	d_W/d_t	{↔69}	-	
4	{↔69}, Flag 01, 0	Integrator	W	{↔72}	-	
5	{↔68}	Hold (digital)	W_max_hold	{↔74}	1	328 mJ
6	{↔72}	Peak value	W_max	{↔68}	-	

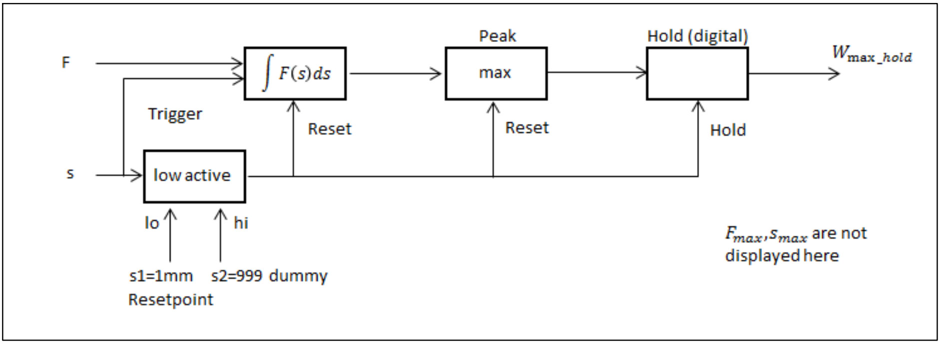
Example B

Measuring the **maximum value** of work.
This is useful, for example, if the full scale of the displacement is undetermined or cannot be reached reliably.

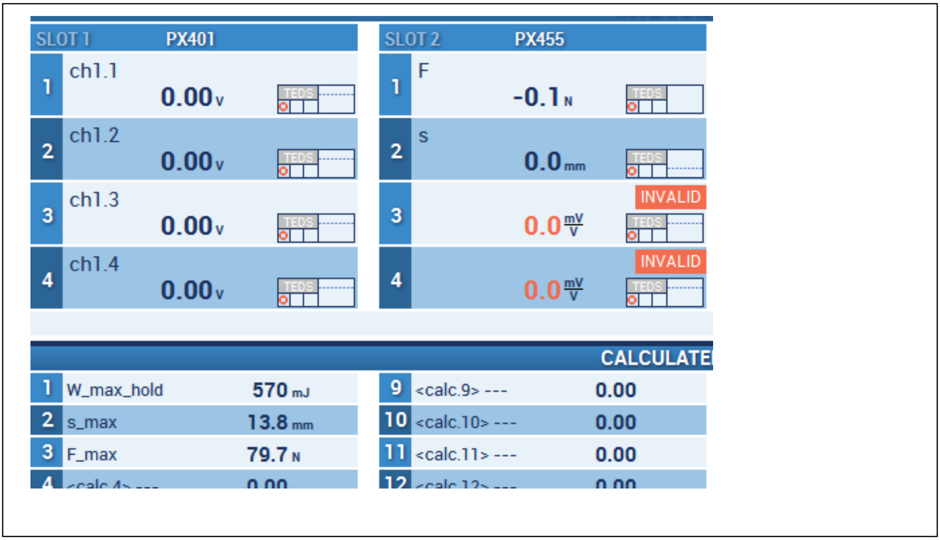
Integration begins and ends at the same point on the displacement axis s=s1:

$$W = \oint_{s1} F(s) * ds.$$

Work may decrease during the return stroke, for example due to spring relaxation. The maximum value is retained with a Peak block.



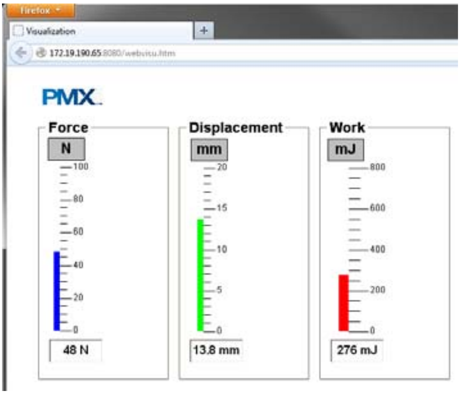
The max. of F and s are also generated for example A:



Overview of glyphs:

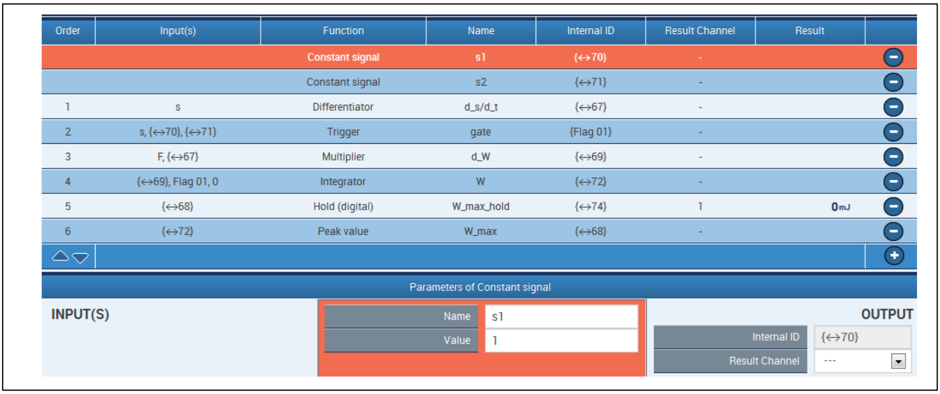
Order	Input(s)	Function	Name	Internal ID	Result Channel	Result	
		Constant signal	s1	{↔70}	-		
		Constant signal	s2	{↔71}	-		
1	s	Differentiator	d_s/d_t	{↔67}	-		
2	s, {↔70}, {↔71}	Trigger	reset	{Flag 01}	-		
3	F, {↔67}	Multiplier	dW	{↔69}	-		
4	{↔69}, Flag 01, 0	Integrator	W	{↔72}	-		
5	{↔68}	Hold (digital)	W_max_hold	{↔74}	1	179 mJ	
6	{↔72}	Peak value	W_max	{↔68}	-		
		1 - 8 9 - 16					

Example of visualization with CODESYS WebVisu

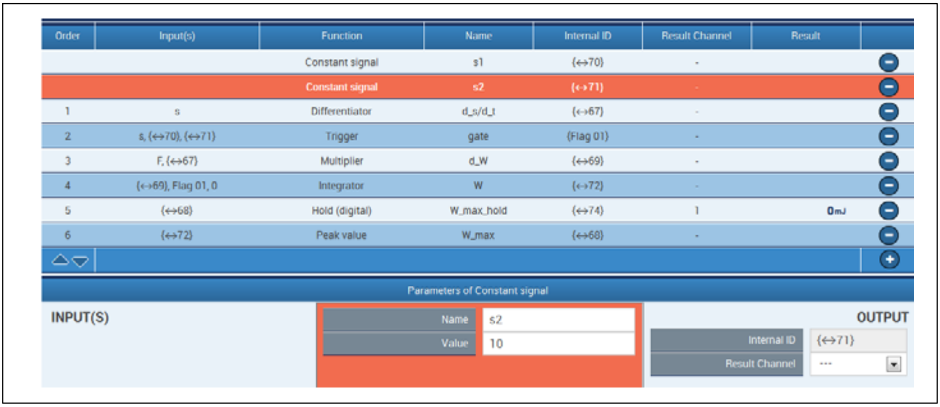


Appendix

Details of the glyphs in example A:



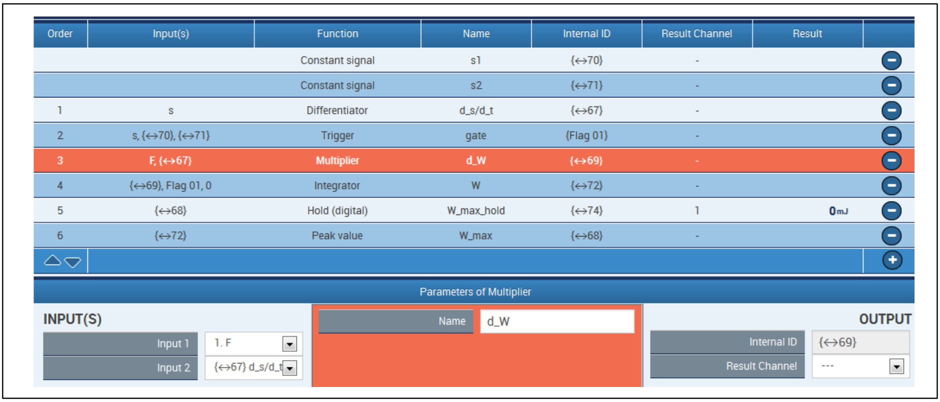
Order	Input(s)	Function	Name	Internal ID	Result Channel	Result
		Constant signal	s1	{↔70}	-	
		Constant signal	s2	{↔71}	-	
1	s	Differentiator	d_s/d_t	{↔67}	-	
2	s, {↔70}, {↔71}	Trigger	gate	{Flag 01}	-	
3	F, {↔67}	Multiplier	d_W	{↔69}	-	
4	{↔69}, Flag 01, 0	Integrator	W	{↔72}	-	
5	{↔68}	Hold (digital)	W_max_hold	{↔74}	1	0 mJ
6	{↔72}	Peak value	W_max	{↔68}	-	

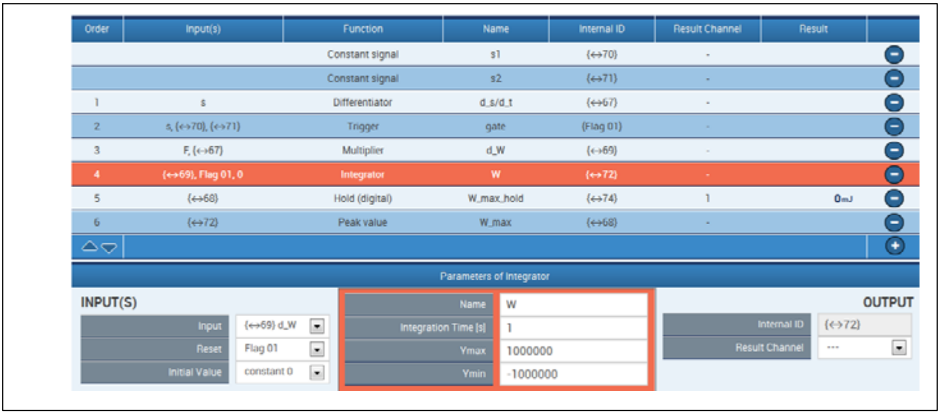


Order	Input(s)	Function	Name	Internal ID	Result Channel	Result
		Constant signal	s1	{↔70}	-	
		Constant signal	s2	{↔71}	-	
1	s	Differentiator	d_s/d_t	{↔67}	-	
2	s, {↔70}, {↔71}	Trigger	gate	{Flag 01}	-	
3	F, {↔67}	Multiplier	d_W	{↔69}	-	
4	{↔69}, Flag 01, 0	Integrator	W	{↔72}	-	
5	{↔68}	Hold (digital)	W_max_hold	{↔74}	1	0 mJ
6	{↔72}	Peak value	W_max	{↔68}	-	

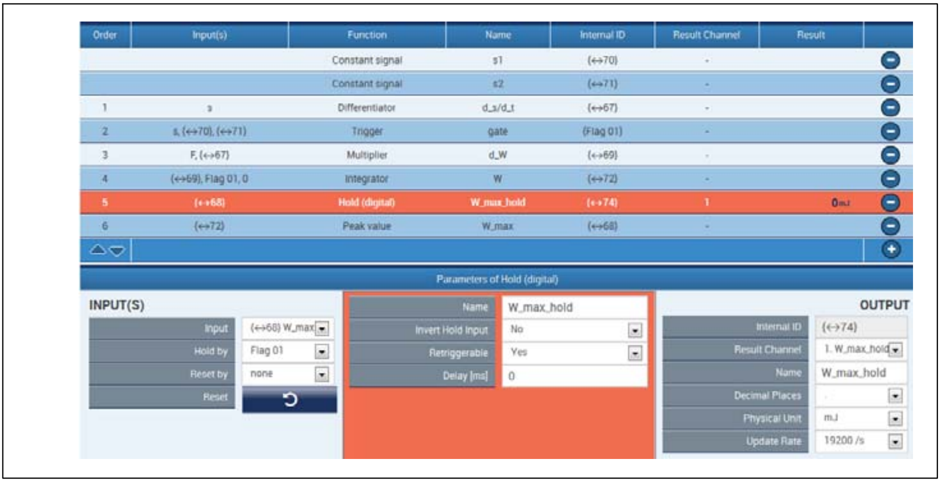

Order
Input(s)
Function
Name
Internal ID
Result Channel
Result
Constant signal s1 {↔70} -
Constant signal s2 {↔71} -
1 s Differentiator d_s/d_t {↔67} -
2 s, {↔70}, {↔71} Trigger gate {Flag 01} -
3 F, {↔67} Multiplier d_W {↔69} -
4 {↔69}, Flag 01, 0 Integrator W {↔72} -
5 {↔68} Hold (digital) W_max_hold {↔74} 1 0 mJ
6 {↔72} Peak value W_max {↔68} -
Parameters of Differentiator
INPUT(S)
Input
2. s
Name
d_s/d_t
Ymax
1000000
Ymin
-1000000
OUTPUT
Internal ID
{↔67}
Result Channel

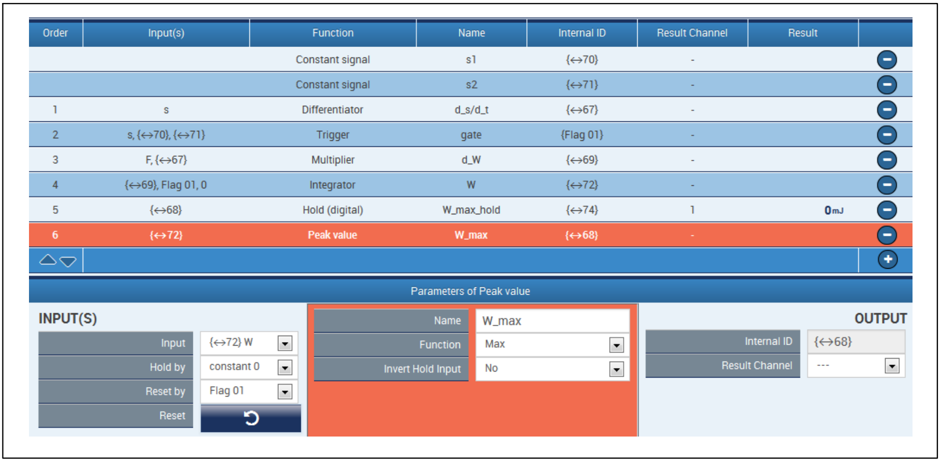

Order
Input(s)
Function
Name
Internal ID
Result Channel
Result
Constant signal s1 {↔70} -
Constant signal s2 {↔71} -
1 s Differentiator d_s/d_t {↔67} -
2 s, {↔70}, {↔71} Trigger gate {Flag 01} -
3 F, {↔67} Multiplier d_W {↔69} -
4 {↔69}, Flag 01, 0 Integrator W {↔72} -
5 {↔68} Hold (digital) W_max_hold {↔74} 1 0 mJ
6 {↔72} Peak value W_max {↔68} -
Parameters of Trigger
INPUT(S)
Input
2. s
Threshold high ↑
{↔71} s2
Threshold low ↓
{↔70} s1
Name
gate
Hysteresis
0.1
Delay [ms]
0
Active
Low
OUTPUT
Flag
Flag 01





Please note that the Hold block comes **before** the Peak block in the order of calculations. Thus the edge of Flag_01 first results in the Hold function and then causes the Peak block to be reset.


Order | Input(s) | Function | Name | Internal ID | Result Channel | Result
Constant signal | s1 | {↔70} | -
Constant signal | s2 | {↔71} | -
1 | s | Differentiator | d_s/d_t | {↔67} | -
2 | s, {↔70}, {↔71} | Trigger | gate | {Flag 01} | -
3 | F, {↔67} | Multiplier | d_W | {↔69} | -
4 | {↔69}, Flag 01, 0 | Integrator | W | {↔72} | -
5 | {↔68} | Hold (digital) | W_max_hold | {↔74} | 1 | 0 mJ
6 | {↔72} | Peak value | W_max | {↔68} | -
Parameters of Hold (digital)
INPUT(S)
Input | {↔68} W_max
Hold by | Flag 01
Reset by | none
Reset
Name | W_max_hold
Invert Hold Input | No
Retriggerable | Yes
Delay [ms] | 0
OUTPUT
Internal ID | {↔74}
Result Channel | 1. W_max_hold
Name | W_max_hold
Decimal Places
Physical Unit | mJ
Update Rate | 19200 /s


Order | Input(s) | Function | Name | Internal ID | Result Channel | Result
Constant signal | s1 | {↔70} | -
Constant signal | s2 | {↔71} | -
1 | s | Differentiator | d_s/d_t | {↔67} | -
2 | s, {↔70}, {↔71} | Trigger | gate | {Flag 01} | -
3 | F, {↔67} | Multiplier | d_W | {↔69} | -
4 | {↔69}, Flag 01, 0 | Integrator | W | {↔72} | -
5 | {↔68} | Hold (digital) | W_max_hold | {↔74} | 1 | 0 mJ
6 | {↔72} | Peak value | W_max | {↔68} | -
Parameters of Peak value
INPUT(S)
Input | {↔72} W
Hold by | constant 0
Reset by | Flag 01
Reset
Name | W_max
Function | Max
Invert Hold Input | No
OUTPUT
Internal ID | {↔68}
Result Channel | ---

Tip

In example A the Hold block is already reset during the return. If the value needs to be held for longer, the block could for example be reset externally by a digital signal or by an additional Trigger block.

13.3.4 Testing force at specific points on the displacement axis

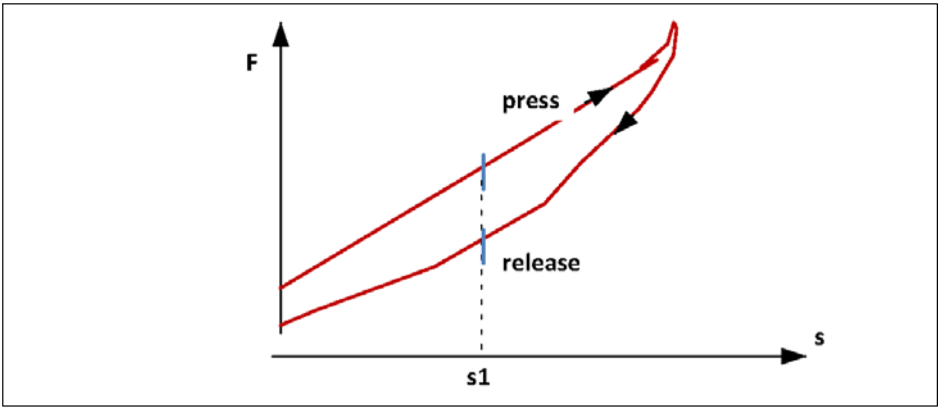
Short description

During a stroke motion with measurement of force and displacement, the force is measured at a specific point on the displacement axis. The measured force must fall within an acceptance band and a good/bad decision will be made.

The force is checked once during the outward stroke and once during the return.

Introduction

In this example the force curve is approximately as follows:



Two "Hold (analog)" glyphs hold the force measured value at point s1. One block during the outward stroke and the other during return. Two limit switches check whether the values are within their respective acceptance band. The rising force is the "press", the falling force the "release".

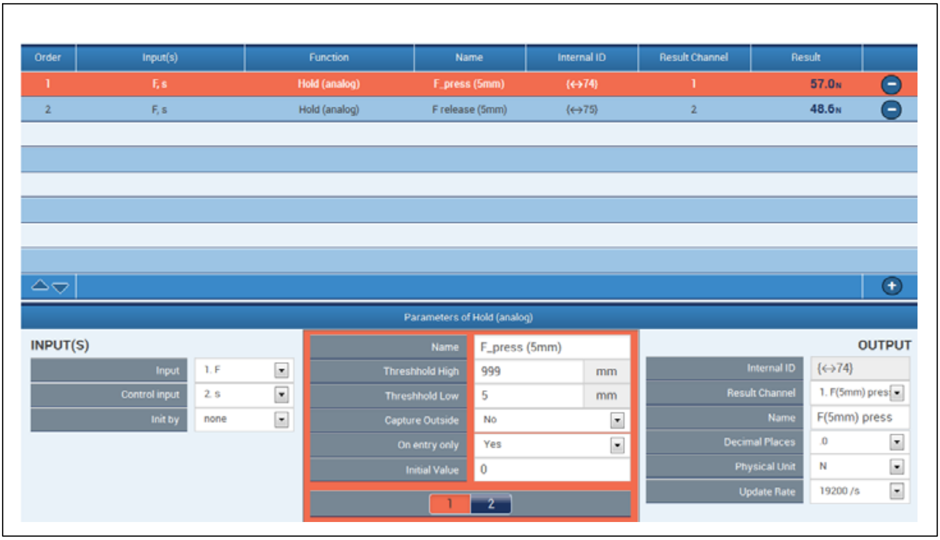
Two digital outputs show the result of the limit switches.

Procedure

"Press" for rising stroke:

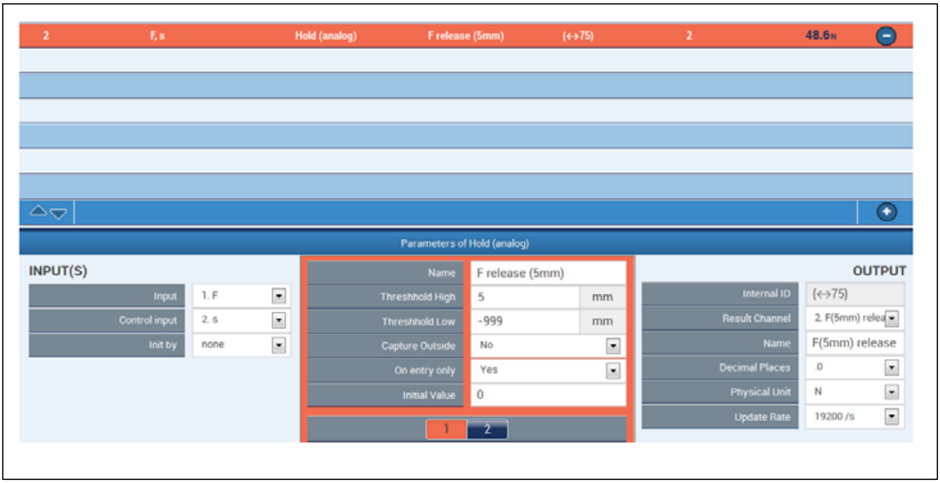
- The measured force value F is retained at point s1=5mm with a "Hold (analog)" glyph. Displacement s controls the hold function. It is therefore located at the control input.
- At s=5mm (Lower Limit, Threshold Low) the glyph reads force F and holds it at the output.

- The upper value of the hold range (Upper Limit, Threshold High) is not needed in this case. A dummy value outside of the measuring range is is assigned to it.
- The setting "On entry only" means that the force is read and held exactly one time for each entry into the hold range [5 mm, 999 mm].



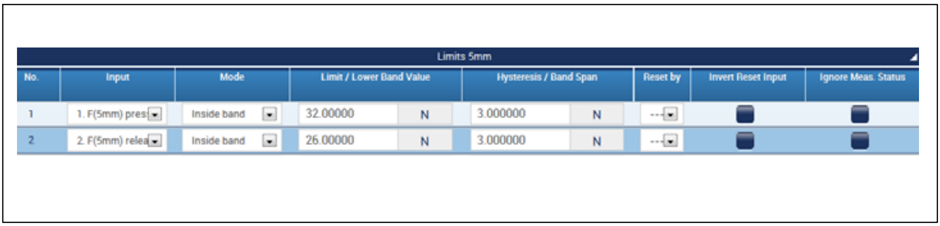
For falling stroke "Release":

- The hold range now includes the range s = [-999 mm, 5 mm]; -999 is a dummy value again here. The block holds the measured force value if the displacement s at 5mm <u>from above</u> enters into the hold range.

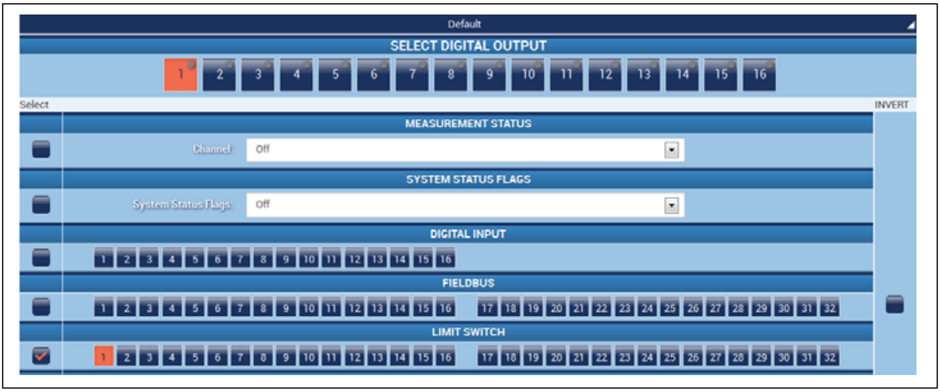


Evaluation by limit switch:

- The outputs of the Hold blocks are the inputs of two limit switches
- Switch No. 1 report the correct force value for rising stroke. It is active if the "F(5mm) press" signal is within the interval [32 N , 35N].
- The settings for switch No. 2 are similar. The acceptance interval is [26 N, 29 N].

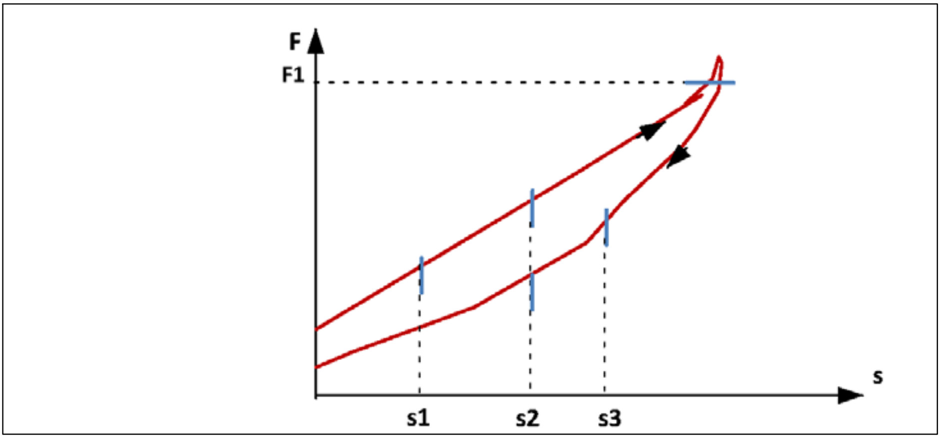


Digital outputs No. 1 and 2 output the states of limit switches Nos. 1 and 2:


Default
SELECT DIGITAL OUTPUT
Select
INVERT
MEASUREMENT STATUS
Channel:
Off
SYSTEM STATUS FLAGS
System Status Flags:
Off
DIGITAL INPUT
FIELDBUS
LIMIT SWITCH


Default
SELECT DIGITAL OUTPUT
Select
INVERT
MEASUREMENT STATUS
Channel:
Off
SYSTEM STATUS FLAGS
System Status Flags:
Off
DIGITAL INPUT
FIELDBUS
LIMIT SWITCH

Tips



- Violation of the acceptance band (as opposed to compliance) can be indicated by inverting the logic, for example with the limit switches.
- Instead of checking the force at points on the displacement axis, of course it is also possible to check displacement at certain points
- Additional direction-dependent check points can also be set up with the appropriate number of Hold blocks.
- Parameter sets can be used to switch the values of the measurement points and/or acceptance bands to other values. A copy of the sub-parameter set Acquisition in which the numeric values are changed has been created for this purpose. Sub-parameter sets are then assigned to different parameter sets, which are switched for example via fieldbus or digital inputs. Instead of individual numeric values, the structure of calculated channels can also be switched. The limit switches are switched by the sub-parameter set Limit Switches.

- The task can also be performed with "Trigger (pulse)" and "Hold (digital)" blocks.

13.3.5 Force/displacement measurement with relative zero point

Short description

The force F and displacement s during a stroke movement are measured. To compensate for random offsets in measurands, new, signals F_tared and s_tared are generated without offset.

Case A

The force is set to zero at point s on the displacement axis: s=s0.

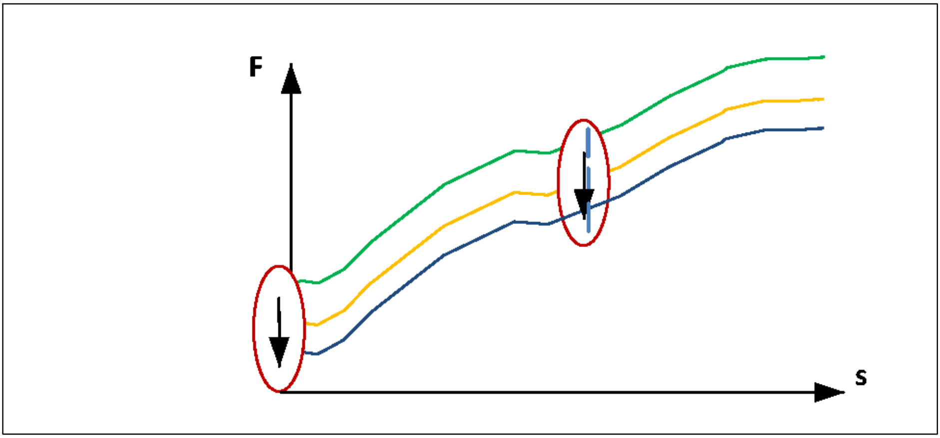
Case B

Force and displacement are set to zero for force F=F0

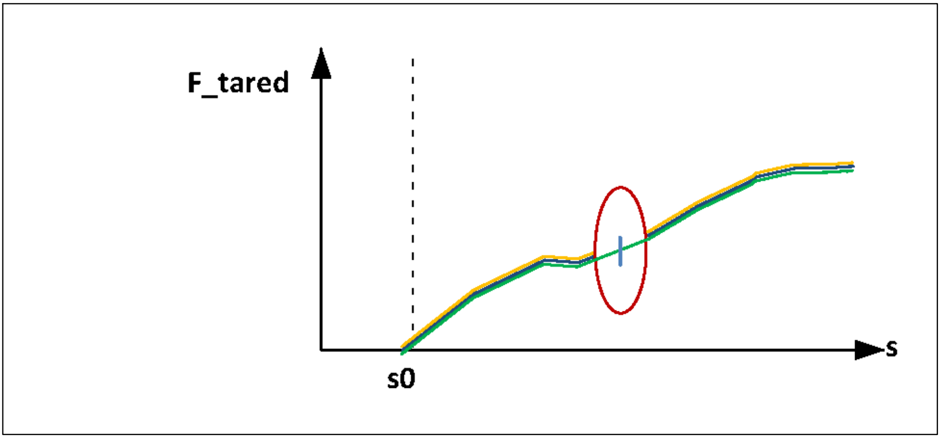
Introduction

Case A

If a displacement-dependent force F is being checked for compliance with the limit values, an offset of the force from stroke to stroke is disruptive. The force limit values would have to be adjusted for each stroke movement.

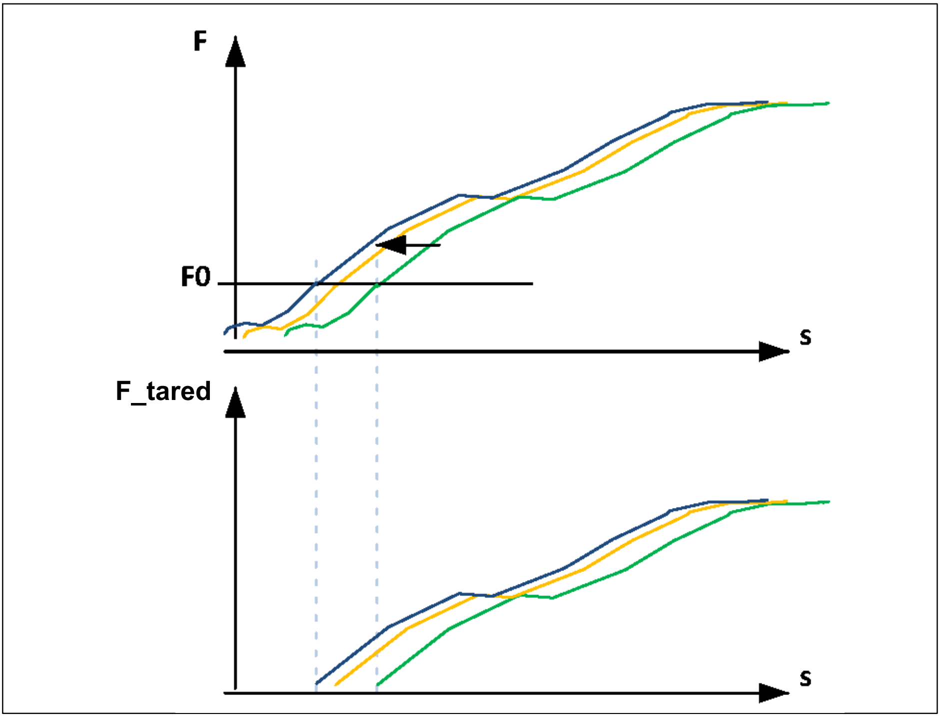


Two glyphs "Trigger" and "Taring" are used to set the force to zero below point s0. The force curve below s0 has no significance for zeroing.

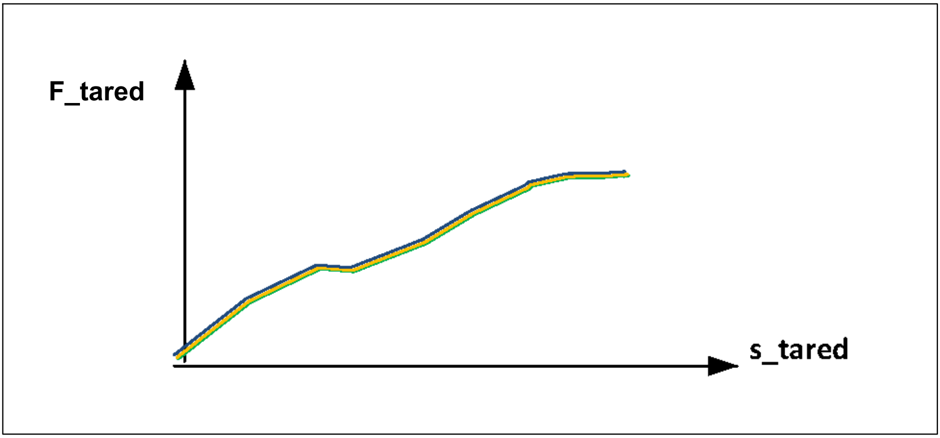


Case B

To compensate for a displacement offset, the new force zero point should be determined by force F0 rather than by the displacement.



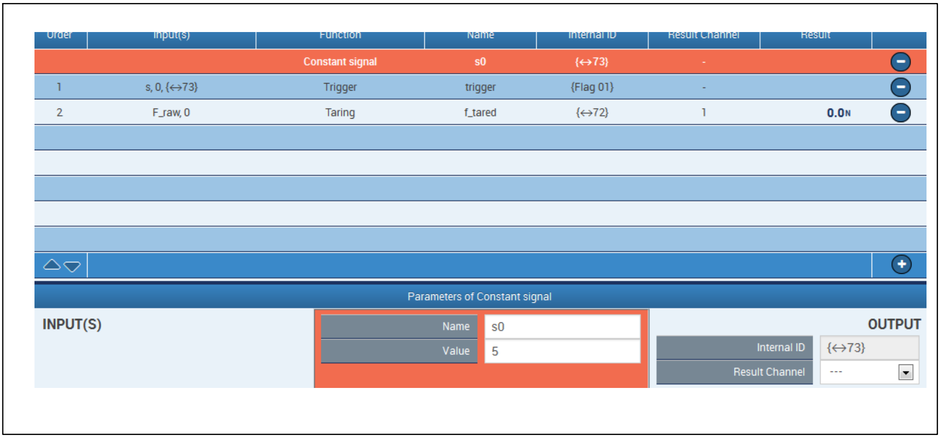
Case A illustrates how the displacement zero point can also be moved. The same condition (F=F0) is used to do this. Both quantities, force and displacement, now have a new zero point.



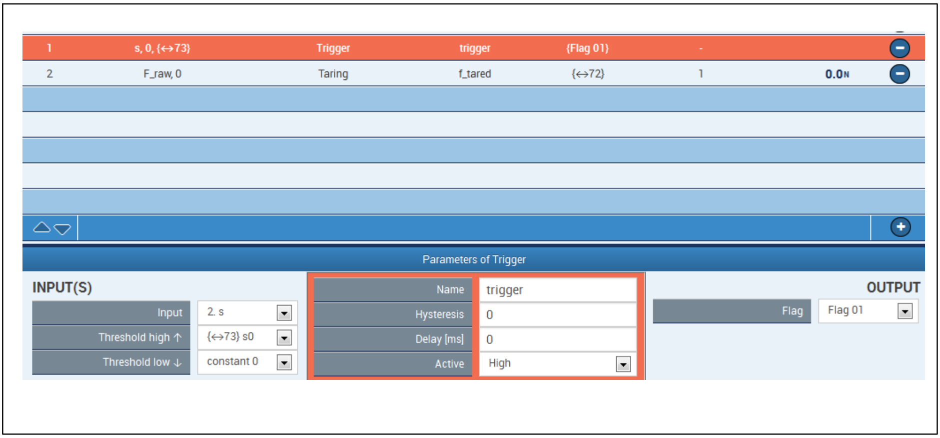
Procedure

Case A

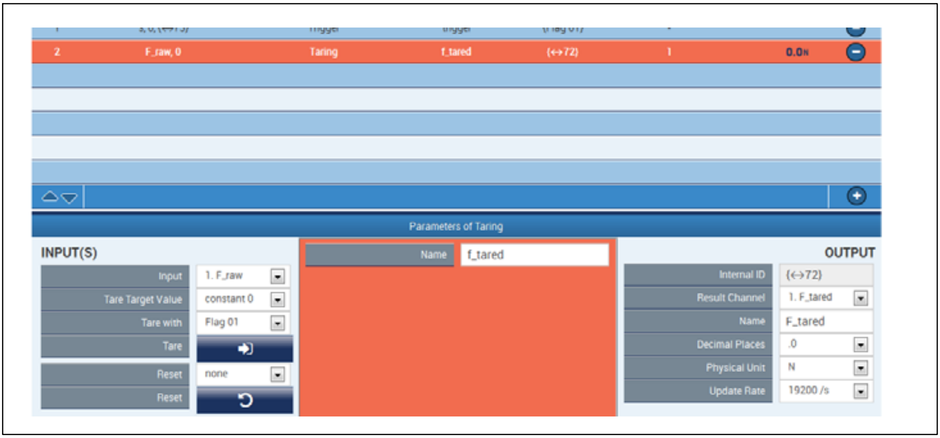
A constant signal returns s0, in this case 5 mm:



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

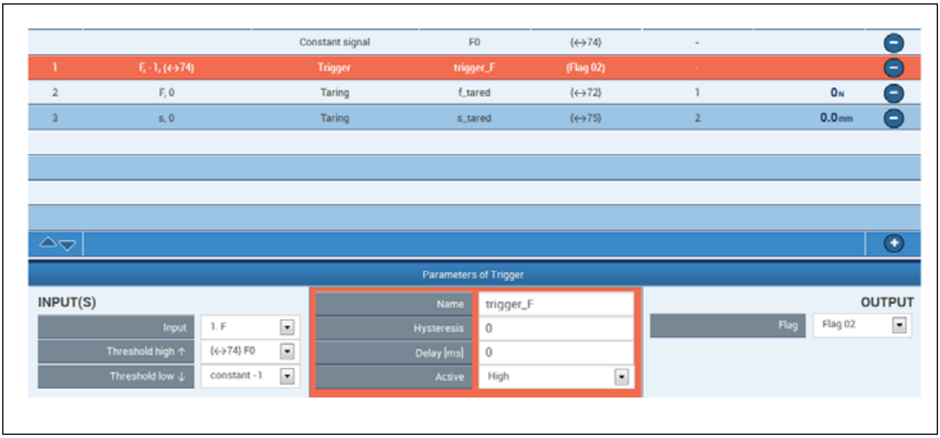


Finally the Trigger block sets the raw value of the force F_raw below 5 mm to zero. F_tared is the force value with the offset removed:

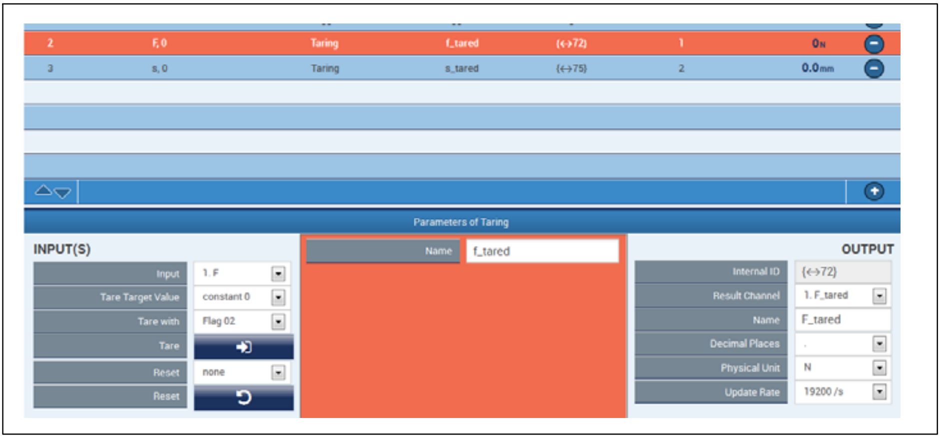


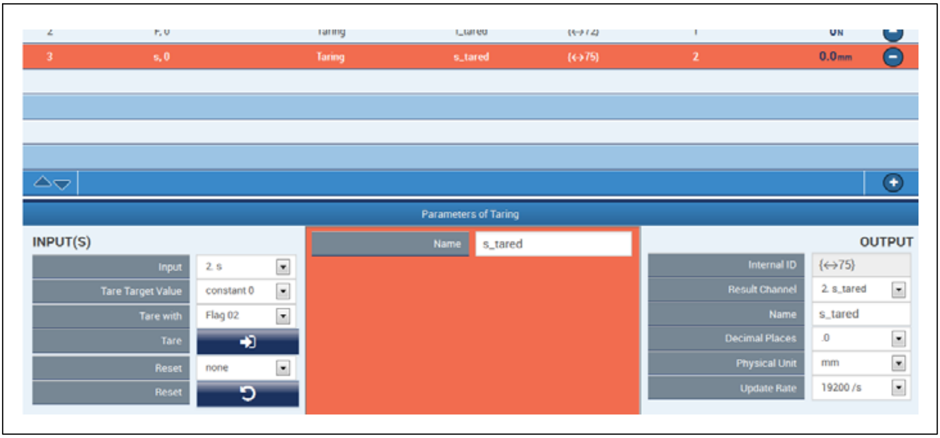
Case B

The Trigger block now checks force F against F0



Force and displacement are both set to zero with signal "Flag02":





13.3.6 Checking force against a tolerance band

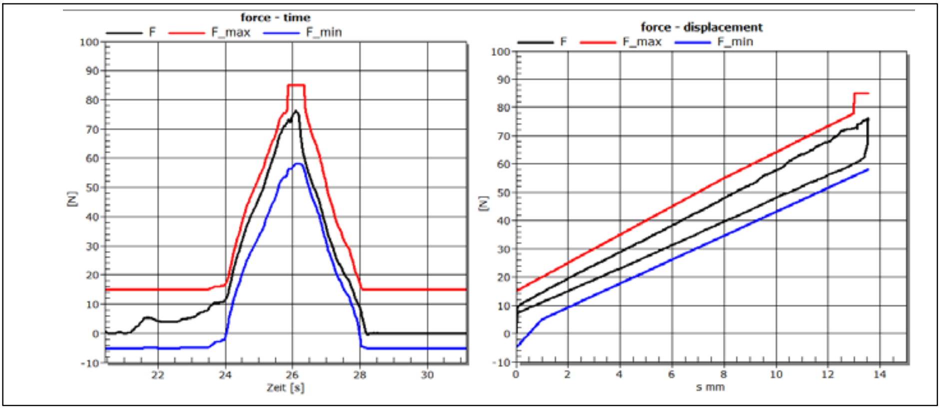
Short description

During a stroke movement with measurement of force and displacement, the force is continuously checked against a tolerance band. The tolerance limits are defined according to displacement. A counter counts the tolerance violations and a digital output reports if the counter state is greater than zero. The counter state is automatically deleted at the beginning of each stroke movement.

(The necessary glyphs are available beginning with firmware version 1.34.)

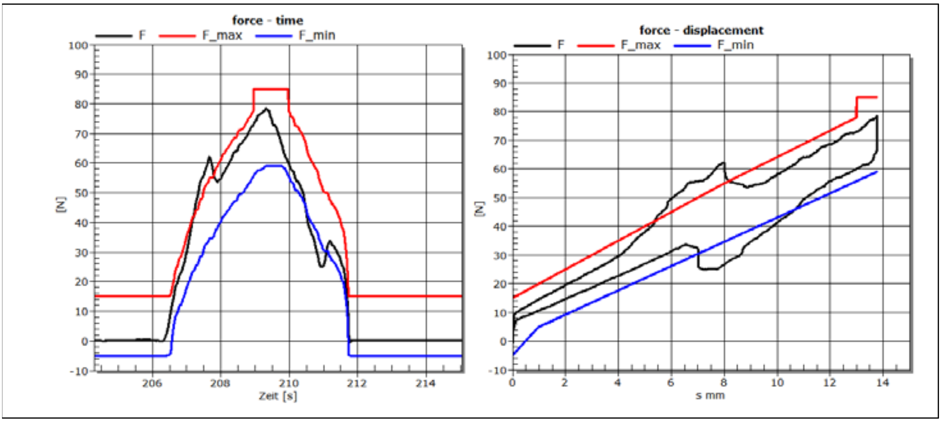
Introduction

The measured force F is checked against an upper and a lower tolerance (F_max and F_min). Two value tables return tolerance values depending on the measured displacement s. The diagram shows the force/time curve and the force/displacement curve of a stroke movement with no violations of the tolerance limits.



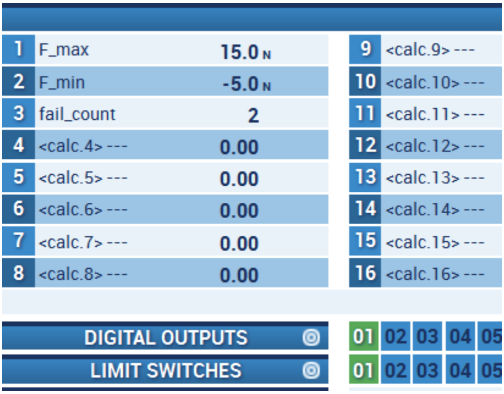
A stroke movement with no tolerance violations

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



Two tolerance violations

The overview after two tolerance violations. Digital output 01 indicates the error.

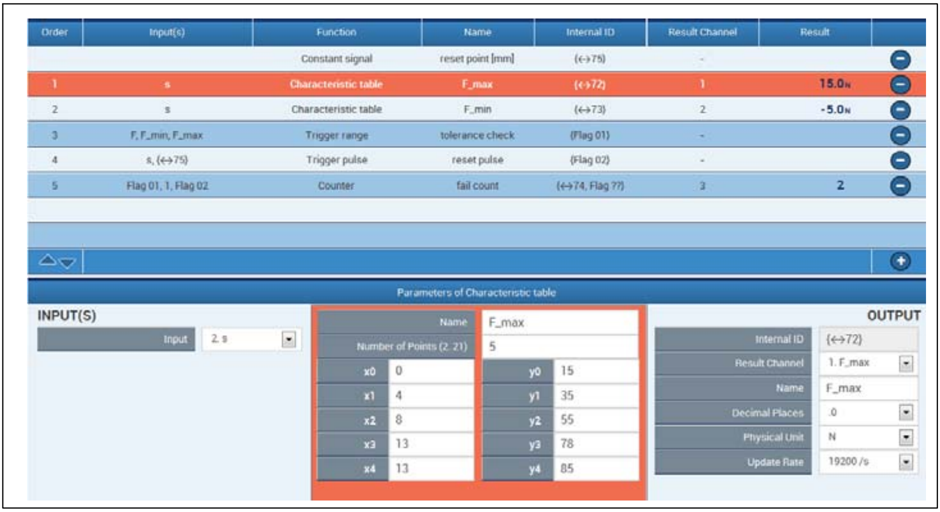


Procedure

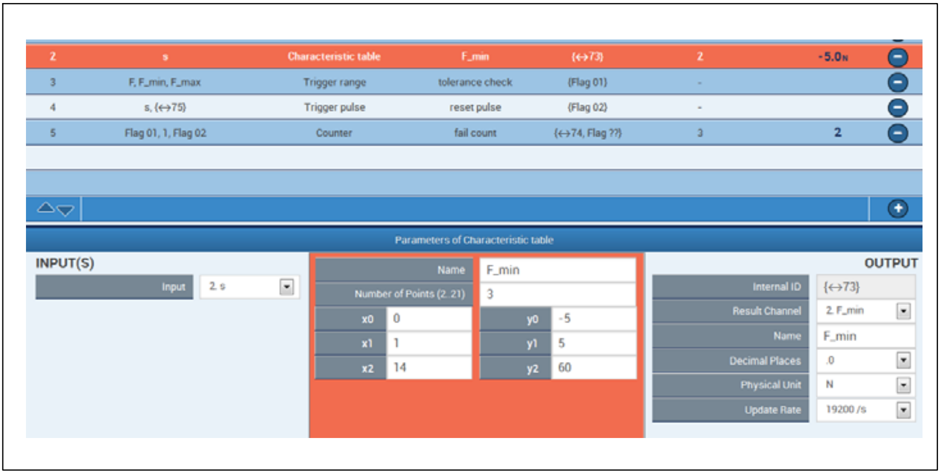
The table for the upper tolerance line has five points on the displacement axis. The x values are the displacement in mm and the y values are the tolerance limit of the force in N.

Tip

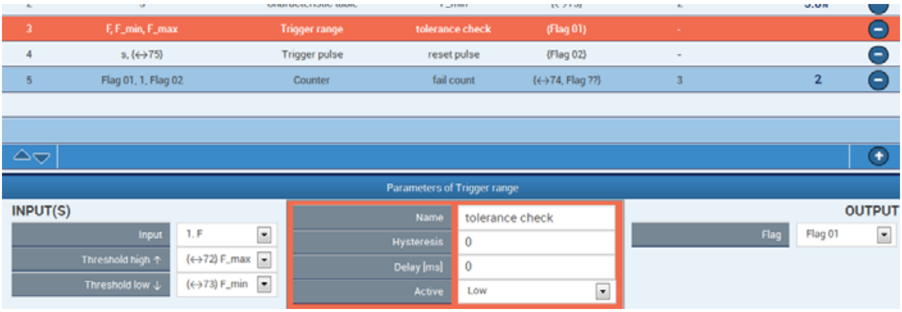
Points with the same x values produce a jump in the characteristic curve, in this case at x3=x4=13.



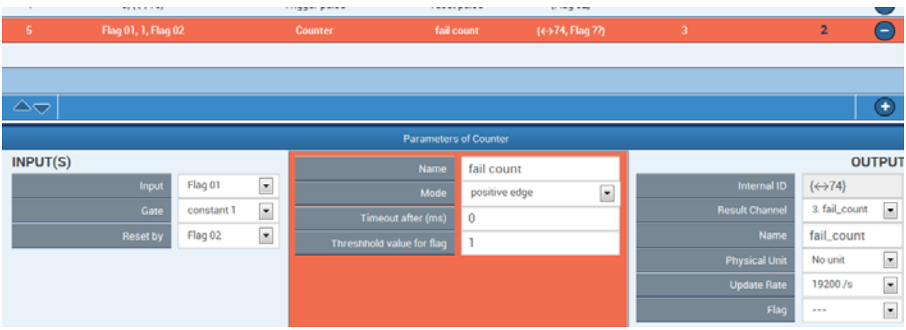
The table for the lower tolerance line:



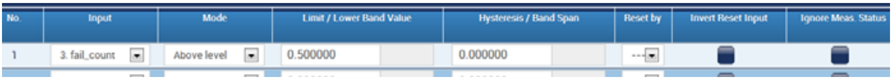
The trigger block compares F with the tolerance limits. Output "Flag_01" is low within the tolerance.



The counter block counts tolerance violations. It is deleted with Flag_02:

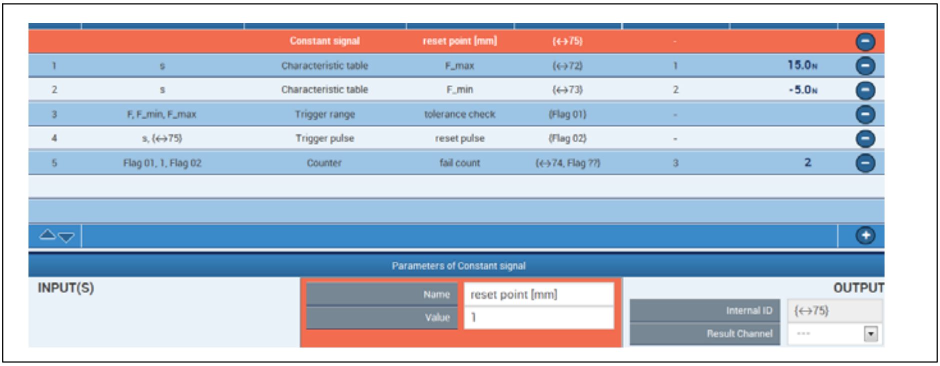


Limit switch No. 1 detects counter states >= 1:



Finally digital output No. 1 outputs the state of limit switch No. 1:

The "Trigger (pulse)" block returns the pulse to delete the counter if the displacement exceeds 1 mm

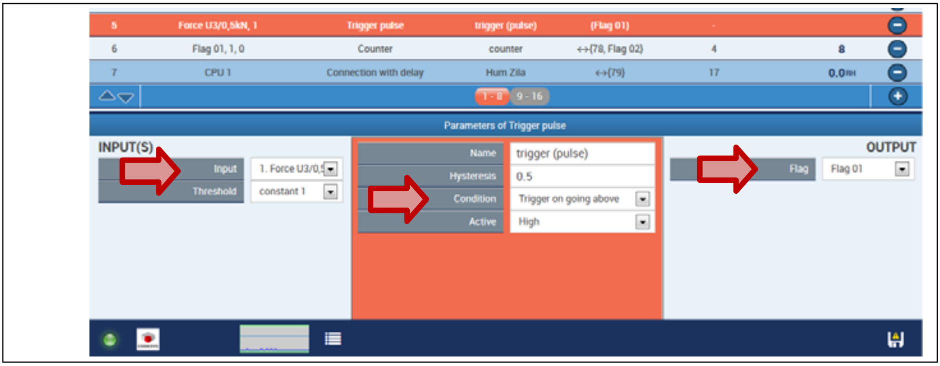


13.3.7 Event counter

The following example shows how to setup an event counter in PMX. Two Calculated Channels are required therefore.

Setup a Calculated Channel „Trigger Pulse“

- The measured force value is the input signal
- A predefined value can be set as limit
- A Condition is selected „Trigger on going above“
- The output is given to „Flag01“

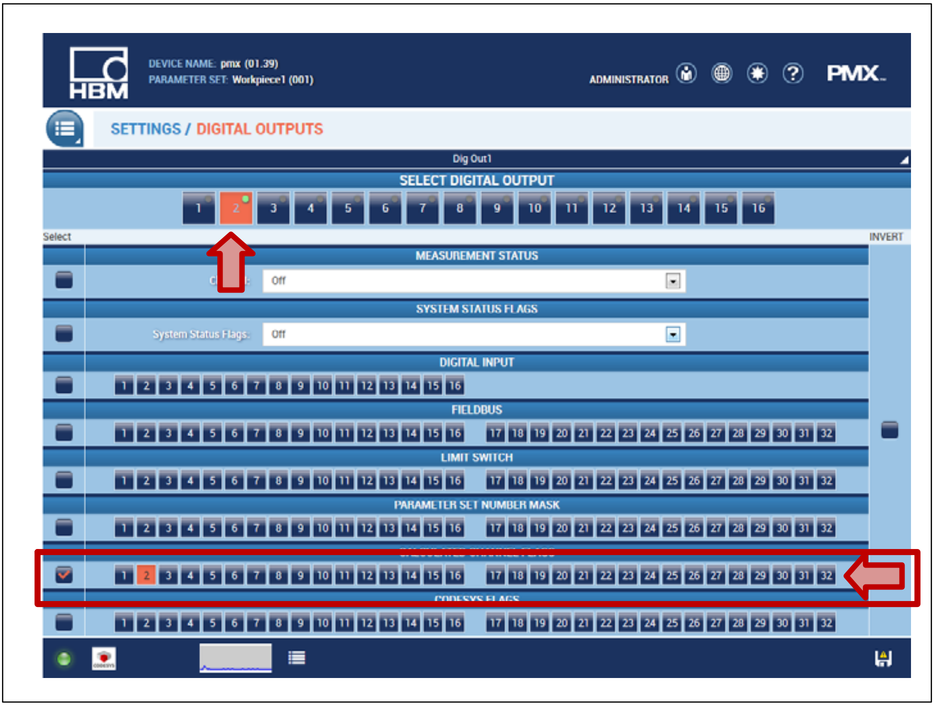


Setup a Calculated Channel „Counter“:

- Flag01 is the input signal
- counted is only with a rising edge (steigende Flanke)
- by setting a specific limit value for the flag a signal can be initiated, which will be set active when the limit value is reached

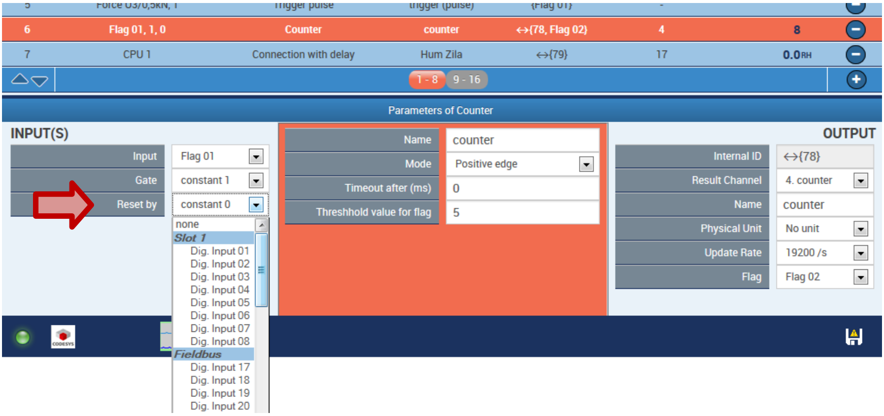
- This activates Flag02, as soon as the counter reaches a value of 5 or higher.
- These Flags can be retrieved by the PLC via a field-bus

- If the result (i.e. achieving of the counter value at a digital output[PX878]) shall be signalized, this flag is set as input for a digital output. In this example digital output 2 is activated by flag 2.



Resetting the event counter can be done by:

- Digital input (PX878)
- By a fieldbus-control word, this is indicated by digital inputs 17..32 of the PMX
- other flags of different Calculated Channels.


6
Flag 01, 1, 0
Counter
counter
↔{78, Flag 02}
4
8
7
CPU 1
Connection with delay
Hum Zila
↔{79}
17
0.0 RH
1 - 8
9 - 16
Parameters of Counter
INPUT(S)
Input
Flag 01
Gate
constant 1
Reset by
constant 0
none
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
Name
counter
Mode
Positive edge
Timeout after (ms)
0
Threshhold value for flag
5
OUTPUT
Internal ID
↔{78}
Result Channel
4. counter
Name
counter
Physical Unit
No unit
Update Rate
19200 /s
Flag
Flag 02

14 Test signals and signal generators

Test signals

Different types of signals can be generated and transmitted with the PMX. This can be a test signal during the starting-up phase, to simulate measured values and thus already test the functioning of the components. During activation of a test signal, 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 calculated channel "Signal generators". The following functions are available in the function parameters:

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

Other parameters include:

frequency, amplitude and offset

The signal generator can also be activated with an on-switch, with repetitions being defined by specifying the 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. The test profile can be controlled by a ramp (timer). First the ramp for controlling the timing of the signal is generated.

To do this, create a new calculated channel "Timer" under the "Technology" category. The period and the ramp interval can be specified in its function parameters, as well as switching between continuous and single shot mode.

Specific signal forms can be generated by a CODESYS application or a dotNET-API application.

Signal output

The generated signals are usually present in the PMX and can be transmitted over the existing interfaces in the PMX. Please note the maximum output rates for each medium.

Signal form	PMX signal generators	CODESYS	dotNET-APILabView
Rectangle, triangle, sine, noise	x	x	x
21-point characteristic	x	x	x
Free signal form	-	x	x
Output rate (max.):			
PX878 (+- 10V)	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 data storage examples can be found in the tips for PMX use on the system CD or in the PMX download area at http://www.hbm.com/en/menu/support/software-firmware-downloads/industrial-amplifiers/

15 Parameter sets (measurement programs)

The PMX has over 1000 independent parameter sets/ measurement programs that are stored failsafe in the device Flash memory. This means that measurement tasks or test sections can be predefined and used at a later date during ongoing operation without additional retooling times. Both the active and not active parameter sets (EDIT MODE) can be configured. Switch-over can be implemented via digital inputs, fieldbus or even via Ethernet, i.e. PMX command set, dotNET-API, LabView driver or CODESYS application..

Important

The configuration of the parameter sets, particularly the active parameter set, should not be implemented in ongoing operation. To prevent faults, parameter sets should only be switched after the respective measurement/test task is complete. In all cases, status bits/information are available to signal fault-free switching (digital outputs, system status, cyclical device data in fieldbus operation).

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

- Sensor data
- Measured value acquisition (also includes the calculated channels)
- Limit values
- Digital outputs

A parameter set is built up out of the individual sub-parameter sets. The settings are in the sub-parameter sets. If a sub-parameter set is not to be changed, "ignore" can be defined for it. Then this section is not changed during switching.

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

Sensor data	Acquisition of measured values	Limit values	Digital outputs	Average switching time typ. (ms)
1200	-	-	-	1200
-	950	-	-	950
1200	950	-	-	2150
-	-	100	-	100
1200	950	100	-	2250
	-	-	80	80
1200	950	100	80	2330

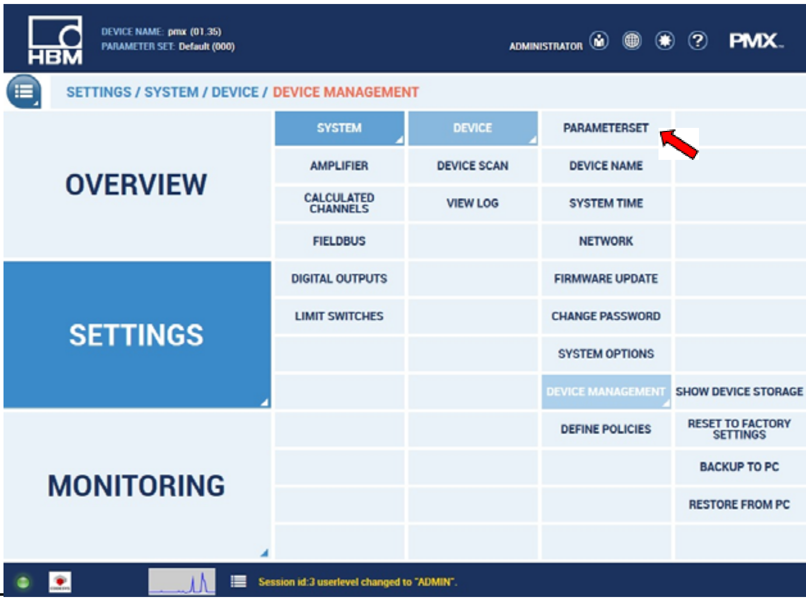
Important

If the voltage supply fails during the saving of a parameter set, this will destroy the parameter set and the PMX will boot up with the factory setting after being switched back on again. To avoid this, backing up the device settings to PC is urgently recommended.
The settings and parameter sets are retained through a firmware update. For safety sake, however, a backup of the parameter sets should be made to PC.

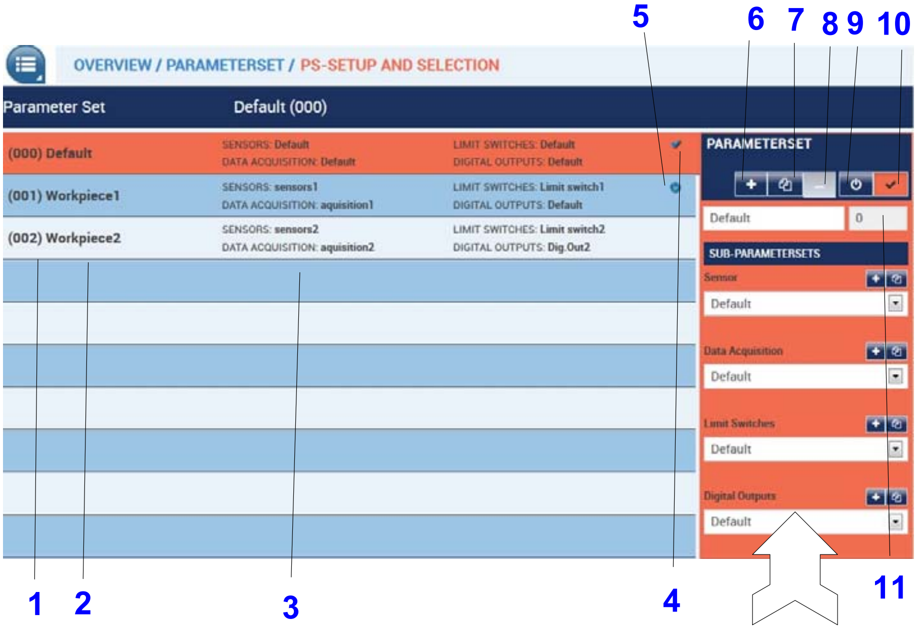
15.1 Setting parameter sets

Follow these steps:

1. First generate an entire parameter set (copy)
2. Then call one of the partial parameter sets and change or adjust the relevant parameters
3. After all the entries are complete, save the parameter sets power failsafe in PMX by clicking on the diskette icon in the status bar



Administration of parameter sets



Create, copy, allocate and delete sub-parameter sets

1 Unique parameter set index

2 Parameter set name, freely selectable

3 The parameter set consists of these sub-parameter sets

4 Is now active

5 Created during booting

6 Create parameter set with factory setting

7 Clone marked parameter set

8 Delete marked parameter set

9 Set as Boot parameter set

10 Set as active parameter set

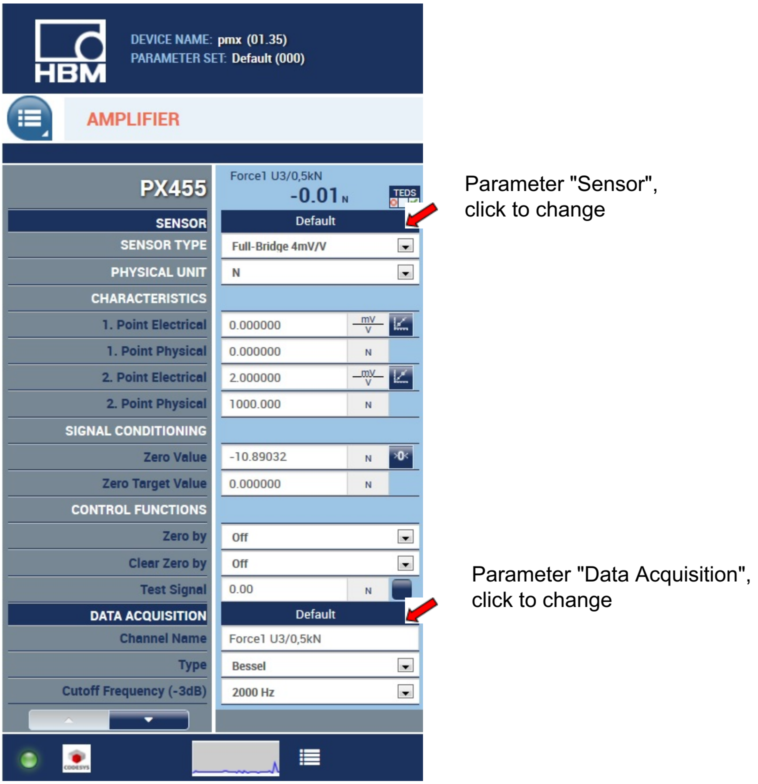
11 Parameter set index

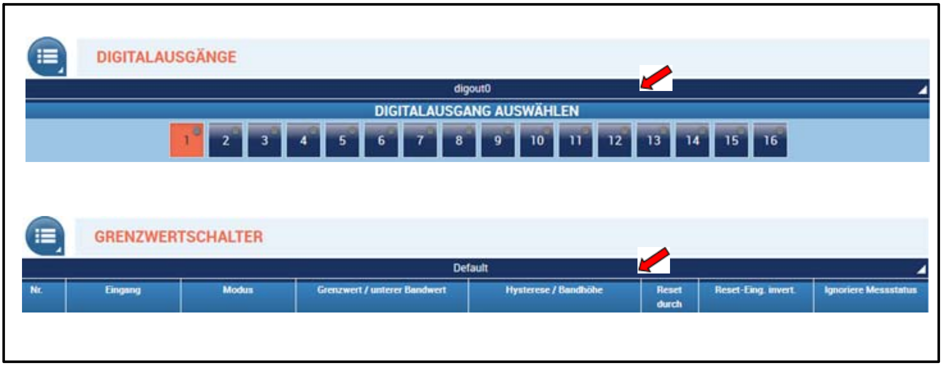
Important

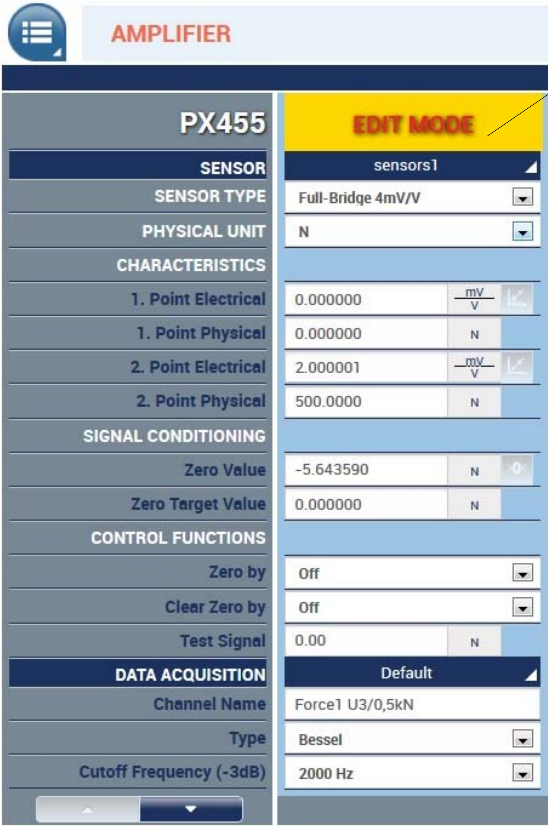
Parameter sets can be copied, deleted and combined in the "Parameter set" dialog. The specific settings for sub-parameter sets are made in the dialogs for the measuring cards.

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.




DIGITALAUSGÄNGE
digout0
DIGITALAUSGANG AUSWÄHLEN
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
GRENZWERTSCHALTER
Default
Nr.
Eingang
Modus
Grenzwert / unterer Bandwert
Hysterese / Bandhöhe
Reset durch
Reset-Eing. invert.
Ignoriere Messstatus



One of the sub-parameter sets is now not active. Settings can be edited, but they will not be effective immediately. They become effective when a parameter set that contains this sub-parameter set is activated.

15.3 Saving and loading measuring 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 the Disksymbol on the bottom right in the web browser.

Saving and loading to and from a PC

The "Backup to PC" menu item is used to create an XML data record that can be used as a backup or transferred to other devices fitted with the same measuring 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

This method does not change passwords and network settings (also see chapetr 24.6).
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 it is inadvisablet.

Example:

The table shows the parameter set - sub-parameter set allocation:

Parameter set	Sensor	Measured value acquisition	Limit values	Digital outputs
000 Factory settings	Default	Default	Default	Default
001 Workpiece A	Default	Fast filter	Limit values Workpiece A	ignore
002 Workpiece B	Default	Slow filter	Limit values Workpiece B	ignore

Switching from parameter set 000 to 001 causes the following:

Sensor Default -> Default	No change, but after switching the "Sensor Default" settings are guaranteed to be present
Acquisition of measured values Default -> "Fast filter"	The settings in "Fast filter" become active
Limit values Default -> "Limit values Workpiece A"	The settings in "Limit values Workpiece A" become active
Digital outputs Default -> ignore	No change, the "Digital outputs" settings remain as they are, they are dependent on the prior history

16 Communication with a control system

Digital inputs and outputs, as well as digital interfaces (EtherCAT®, PROFINET or EtherNet/IP), are available for linking the PMX to machine or system controls.

In all cases, they access the same device functionality. The input and output signals are available via the interface. These can be found in the Tables in Sections 16.4 and 16.5. the tables in chapter16.4 and 16.5.

16.1 Device description file

Physical properties are described in the device master data file (e.g. transmitted/received bytes). 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 internal PMX device memory
- on the Website www.hbm.com/en/menu/support/software-firmware-downloads/industrial-amplifiers and
- on the PMX system CD

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 2.0 and newer
PROFINET	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 interface card (PX01PN) note the version (software stack) of the card. It appears in the "Fieldbus" menu in the web browser.

Please use the appropriate GSDML file from the table below, depending on the software stack and firmware in the PMX.

PMX Firmware	PROFINET Stack 3.4.15	PROFINET 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 application V2.1.0	**GSDML-V2.25-HBM-PMX-20121025.xml (RT only)** **GSDML-V2.25-HBM-PMX_IRT-20130404.xml (RT and IRT)**	GSDML-V2.25-HBM-PMX-20121025.xml (RT only) GSDML-V2.3-HBM-PMX-IRT-noPLC -20141216.xml (RT and IRT)
1.44 **without PLC channels** PROFINET application V1.0.0	**GSDML-V2.25-HBM-PMX-20121025.xml (RT only)** **GSDML-V2.25-HBM-PMX_IRT-20130404.xml (RT and IRT)**	Not supported

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

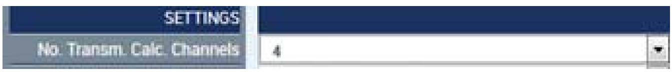
Click on the "Create GSDML File" button to create a device description file which exactly matches the PMX that is used with its plug-in card and the computing channels to be transferred. This eliminates the need for a manual adjustment in the configuration software of the control unit

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 PROFINET.

Important

The number of computed channels to be transferred must be calculated before generating.



Storing data in the device

The file is stored in public device storage. It can be downloaded or deleted there. Access via menu

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 the following in the address line:

http://pmx/public/PROFINET/ or
http://pmx/public/EtherCAT® or
http://pmx/public/EtherNet/IP

where "pmx" is the PMX network name. Names are case sensitive!

Alternative notation with the IP address, for example

http://172.19.201.184/public/PROFINET/

Exceptions

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

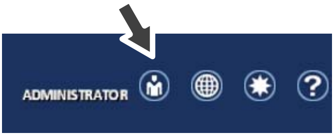
A firmware update (including one with the same version number as the installed firmware) will restore the template file.

If a measurement card is not installed and no calculated channels have been transferred yet, the generated file is not a valid file and will not be accepted by PROFINET configurators or. EtherCAT® or EtherNet/IP .

16.2 Setting the transmission speed of the fieldbus

The fieldbus transmission speed can be set.

► Set the user level to Administrator in the overview top right.



► In the menu Settings – Device – System options: Set the “Internal data transfer rate." to the required, reliable value.

The fieldbus update rate will follow this value up to the fieldbus-specific maximum.

The change is effective immediately.

► To save permanently, click on the floppy disk symbol  .

16.3 Data transmission by fieldbus

6 bytes are required per measured value via the fieldbus (EtherCat®, PROFINET or EtherNet/IP). This includes 4 bytes data + 1 byte control word and 1 byte status.

As a formula for the data volume:
46 bytes basic load + 6 bytes * Number of measurement channels and calculation channels

16.4 Input data, PMX ➪Controller (PLC)

16.4.1 Device data (cyclic)

Function		EtherCAT® Index	PROFINET slot.subslot	Data type
System status	See table „System status“ 13.4.2	6000.1	0.2 bytes 0..3	uint32
Parameter set	Currently active parameter set	6000.2	0.2 bytes 4..7	int32
GUI status	Answer object directory	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
Limit value reset acknowledgement	Acknowledgement of the "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 with 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 external consumers (transducer excitation)
6	catman® interface buffer overrun	Data transmission error, loss of data
7	Device not ready	Device running, but not delivering valid measured values
8	Calculated Channels Overrun	Computing time overrun in calculated channels

16.4.3 Measured values (cyclic)

Function		EtherCAT® Index	PROFI-NET slot.sub-slot	Data type
Flags	Status flags from calculated channels	6001.1	0.3 bytes 0..3	uint32
Flags status (reserved for future use)	always 0	6001.2	0.3 byte 4	uint8
Flags control word acknowledgement	'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 status	always 0	6002.2	0.4 byte 4	uint8
Digital inputs control word acknowledgement	'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
Measured value status	see 'Measured value status' table 16.4.4	60xy.2	x.y byte 4	uint8
Measured value control word acknowledgement (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	PROFI-NET slot.sub-slot	Data type
Calculated channel in slot 9.z		60xy.1	9.z bytes 0..3	float32
Status	see 'Measured value status' table 16.4.4	60xy.2	9.z byte 4	uint8
Control word acknowledgement	Control word return	60xy.3	9. byte 5	uint8
–	Number dependent on the number of calculated channels set on the fieldbus			

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 6051 to 60b4.

16.4.4 Measured value status

Bit	Function	
0	Working standard calibration invalid	-
1	Measured value invalid	Overflow, underflow, defective sensor, calibration ongoing
2	Autocalibration ongoing	Measurement channel with automatic 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, controller (PLC) ⇨ PMX

16.5.1 Device data (cyclic)

Function		EtherCAT® Index	PROFINET slot.sub-slot	Data type
Device control word	Bit0: LEDs flash for 30s Bit1 (value 0x02): Enable Object Directory server Bit2 (value 0x04): Save parameters (same function as floppy disk symbol on web interface) Edge triggered 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	Command object directory	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

Function		EtherCAT® Index	PROFINET slot.sub-slot	Data type
Limit value 0	Limit value No. 0	7000.6	0.2 bytes 20..23	float32
…				
Limit value 15	Limit value No. 15	7000.21	0.2 bytes 80..83	float32
Digital outputs	Setting digital outputs Digital output x = bit x (this set bit is assigned to a digital output on a PX878 via the "Digital output" menu in the PMX web browser)	7000.22	0.2 bytes 84..87	uint32
Digital inputs	Bits 16..31 of the digital outputs are also transmitted as "Digital inputs 17..32" to the calculated channels. This enables function blocks to be controlled.	7000.22	0.2 bytes 84..87	uint32
PLC channel 1 (from firmware version 2.00)	unrestricted use	7002.1	0.2 bytes 88…91	Float 32
PLC channel 2 (from firmware version 2.00)	unrestricted use	7002.2	0.2 bytes 92...95	Float 32

Function		EtherCAT® Index	PROFINET slot.sub-slot	Data type
PLC channel 3 (from firmware version 2.00)	unrestricted use	7002.3	0.2 bytes 96...99	Float 32
PLC channel 4 (from firmware version 2.00)	unrestricted use	7002.4	0.2 bytes 100…103	Float 32
PLC channel 5 (from firmware version 2.00)	unrestricted use	7002.5	0.2 bytes 104…107	Float 32
PLC channel 6 (from firmware version 2.00)	unrestricted use	7002.6	0.2 bytes 108…111	Float 32
PLC channel 7 (from firmware version 2.00)	unrestricted use	7002.7	0.2 bytes 112…115	Float 32
PLC channel8 (from firmware version 2.00)	unrestricted use	7002.8	0.2 bytes 116…119	Float 32

16.5.2 Measured value control words (cyclic)

Function		EtherCAT® Index	PROFINET slot.sub-slot	Data type
Flags control word	(reserved for future use)	7006.1	0.3	uint8
Digital inputs control word	(reserved for future use)	7006.2	0.4	uint8
Control word for measured value slot x.y	Function see 16.5.3	70xy.1	x.y	uint8

Function		EtherCAT® Index	PROFINET slot.sub-slot	Data type
---	Number of control words, subject to the connected measurement cards. One control word per measurement channel.			
Control word for the calculated channel slot 9.z	Function see 16.5.3	70xy.1	9.z	uint8
---	Number of control words, subject to the connected measurement cards. One control word per measurement channel.			

Important

Note on calculated channels

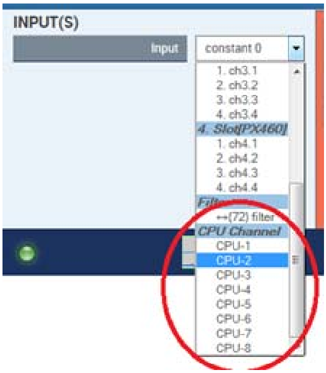
In the PMX device, the calculated channels are assigned to virtual slot 9. For technical reasons, 9 cannot be the third digit in the EtherCAT® indexes.
Calculated channels currently appear in indexes 7051 to 70b4.

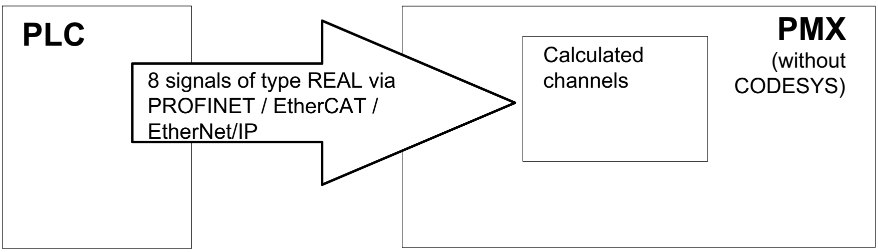
16.5.3 Measured value control words

Bit	Function	Responds to	Applicable to
0	Set to zero	Edge 0 -> 1	Measurement channel
1	Offset = 0.0	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), Refers only to PX455

16.5.4 Fieldbus channels

A controller (PLC) can feed up to 8 signals into the PMX as CPU channels via fieldbus (Industrial Ethernet). The channels will then be available in the calculating channels for further processing. This function is available in the PMX from firmware version 2.00 for ProfiNet, EtherCat and Ethernet/IP.



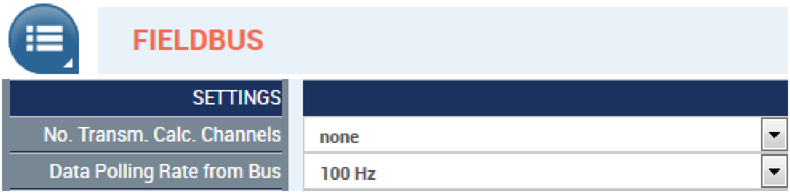


Important

They are not available in devices that are equipped with CODESYS (Grundgerät WGX001)..

The processing speed for fieldbus channel signal transmission from the PLC to the PMX is set with "Data Polling Rate from Bus".

Once the signals are present in the PMX, processing continues at the standard processing rate of 19200/s or 38460/s.

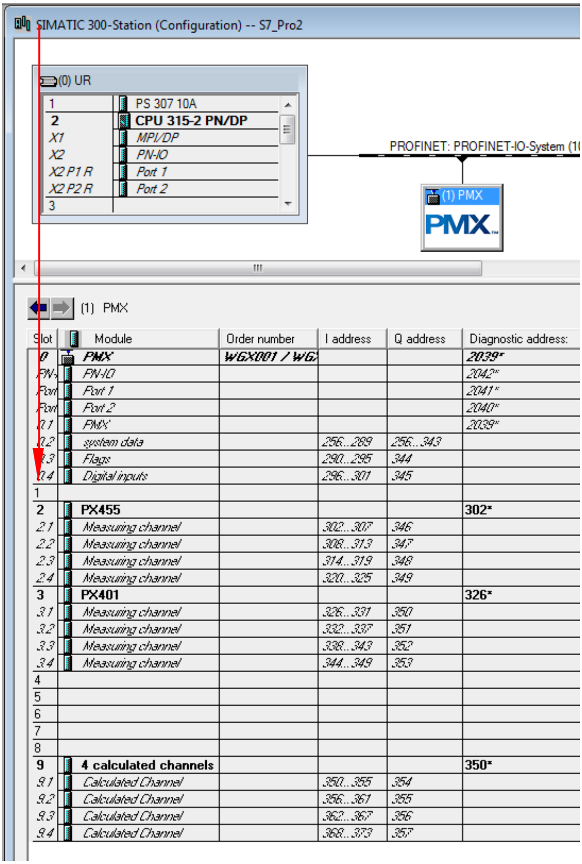


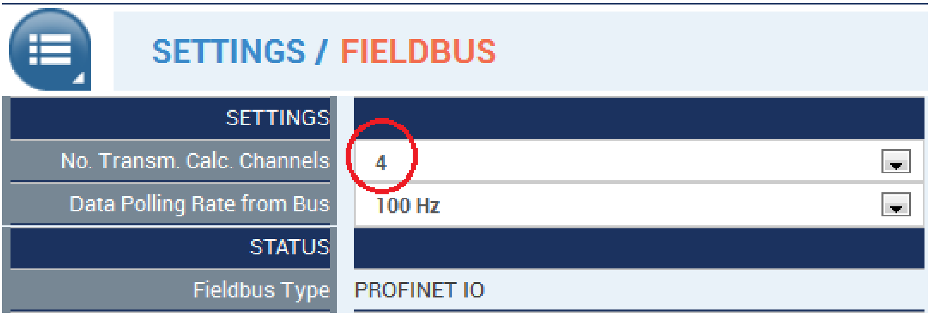
16.6 PROFINET

- Network settings
  The PROFINET-related network settings (IP address, device name, etc.) are selected using the PROFINET configuration tool and set via the PROFINET cable. These data can be read for checking in the "Fieldbus" dialog of the PMX user interface and (from PMX firmware version 3.0), they can also be set.
- The PROFINET configuration must match the installed PMX cards.
- The cable length must be entered in the master configuration in IRT mode, as otherwise there may be transmission errors with long cables.

Example

	Slot 1	Slot 2	Slot 3	Slot 4	Slot 9 (virtual)
Installed in PMX	PX878	PX455	PX401	empty	Calculated channels
PROFINET configuration	No data for PROFINET. Leave this slot empty, see below.	PX455	PX401	empty	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) PMX
PMX
(1) PMX
Slot
Module
Order number
I address
Q address
Diagnostic address:
0 PMX WGX001 / WG 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



Important

Examples regarding calculation channels can be found in the Technical notes on the supplied System CD and in www.hbm.com/en/menu/support/software-firmware-downloads/industrial-amplifiers

16.7 EtherCAT®

The EtherCAT® Master configuration must match the installed cards.

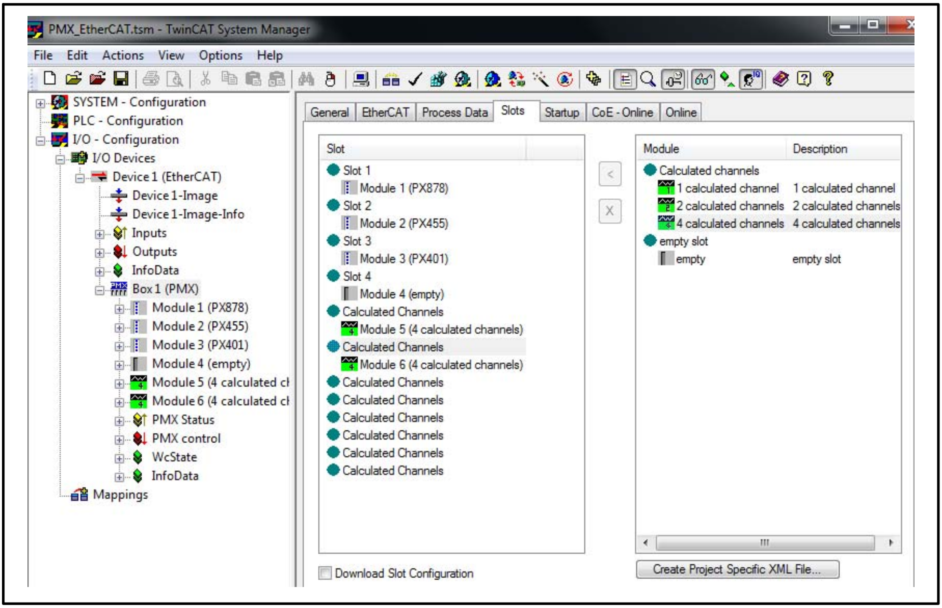
Calculated channels

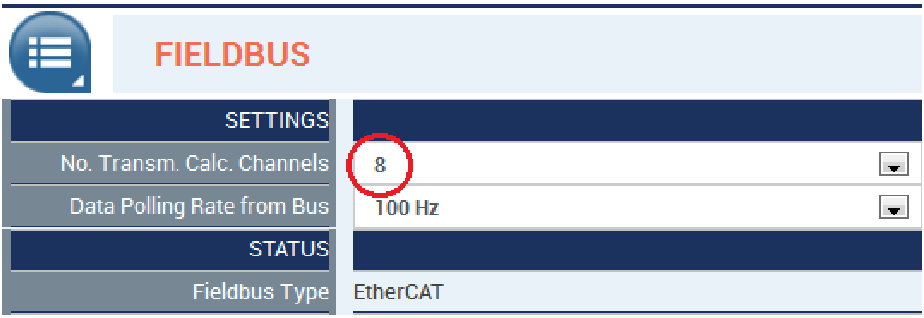
The number must match the PMX setting (Fieldbus menu).

The calculated channels are distributed in the EtherCAT® Master to virtual "Calculated Channels" slots.

The distribution across the slots does not play a role, but the total number of the channels must match.

Example with eight calculated channels:


PMX_EtherCAT.tsm - TwinCAT System Manager
File Edit Actions View Options Help
SYSTEM - Configuration
PLC - Configuration
I/O - Configuration
I/O Devices
Device 1 (EtherCAT)
Device 1-Image
Device 1-Image-Info
Inputs
Outputs
InfoData
Box 1 (PMX)
Module 1 (PX878)
Module 2 (PX455)
Module 3 (PX401)
Module 4 (empty)
Module 5 (4 calculated ch
Module 6 (4 calculated ch
PMX Status
PMX control
WcState
InfoData
Mappings
General EtherCAT Process Data Slots Startup CoE - Online Online
Slot
Slot 1
Module 1 (PX878)
Slot 2
Module 2 (PX455)
Slot 3
Module 3 (PX401)
Slot 4
Module 4 (empty)
Calculated Channels
Module 5 (4 calculated channels)
Calculated Channels
Module 6 (4 calculated channels)
Calculated Channels
Calculated Channels
Calculated Channels
Calculated Channels
Calculated Channels
Calculated Channels
Module Description
Calculated channels
1 calculated channel 1 calculated channel
2 calculated channels 2 calculated channels
4 calculated channels 4 calculated channels
empty slot
empty empty slot
Download Slot Configuration
Create Project Specific XML File...


FIELDBUS
SETTINGS
No. Transm. Calc. Channels
8
Data Polling Rate from Bus
100 Hz
STATUS
Fieldbus Type
EtherCAT

16.8 Using the PMX CoE Object Dictionary

CoE stands for CAN over EtherCAT

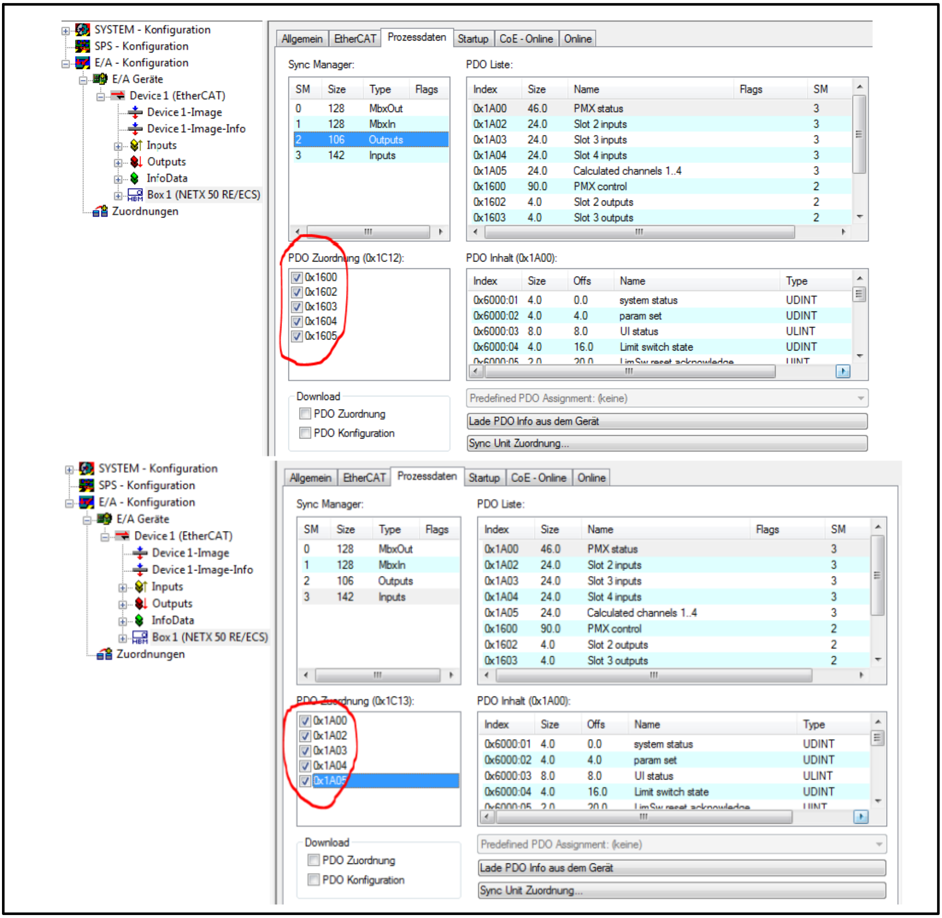
This means there is a great variety available of "CANopen™ devices and application profiles for device classes and applications: Starting from the I/O modules via drives (e.g. drive profile CiA 402 standardized as IEC 61800-7-201/301), encoders (CiA 406), proportional valves and hydraulic regulators (CiA 408) right up to application profiles.

The procedure with TwinCAT:

1. Before starting the TwinCAT, the PMX-ESI file must be deleted from the TwinCAT folder (default C:\TwinCAT\Io\EtherCAT).
   Alternatively, the .xml ending can be made unrecognizable,
   e.g. "HBM_PMX .xml.doNotUseYet".
2. TwinCAT locates the PMX with the device scan.
   PMX does not support partial PDO selection.

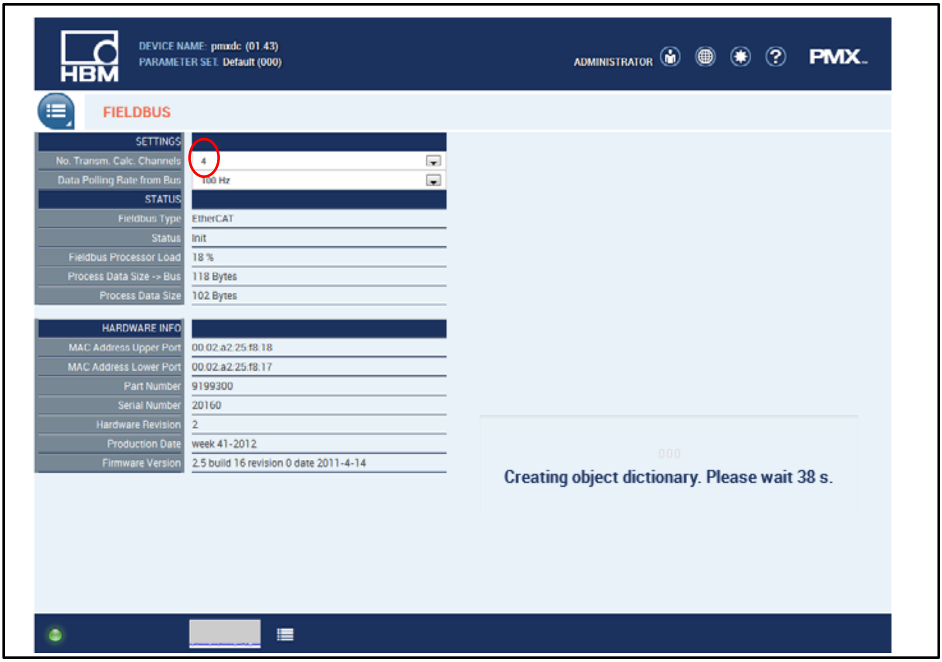
Important

All PDOs must be selected manually, otherwise the Sync-Manager parameters will not agree (it is unfortunately not possible to parameterize the PDOs as Fixed and Mandatory).



3. Subsequent procedure is identical to that for an ESI file.

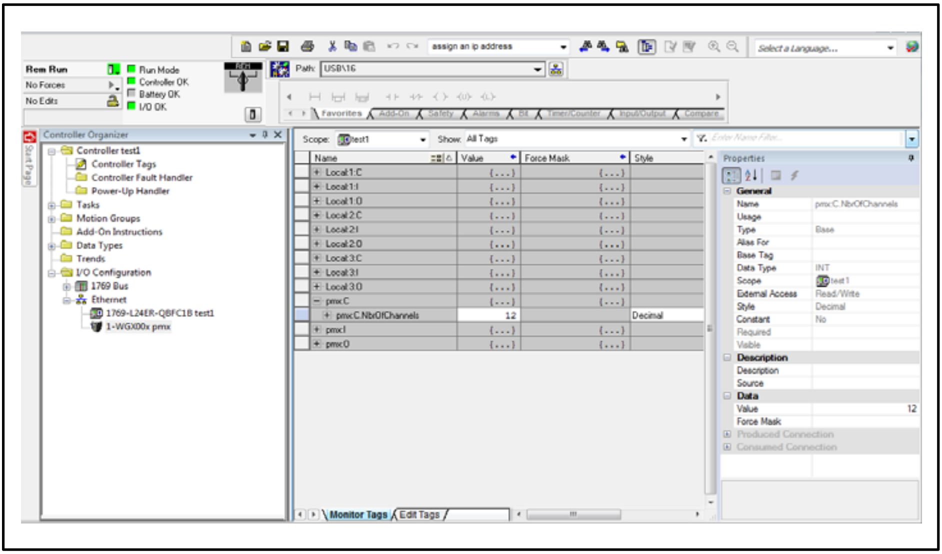
4. If calculated channels need to be sent via EtherCAT, set the required number in the dialog Settings-Fieldbus.


DEVICE NAME: pmxdc (01.43)
PARAMETER SET: Default (000)
ADMINISTRATOR
PMX
HBM
FIELDBUS
SETTINGS
No. Transm. Calc. Channels
4
Data Polling Rate from Bus
100 Hz
STATUS
Fieldbus Type
EtherCAT
Status
Init
Fieldbus Processor Load
18 %
Process Data Size -> Bus
118 Bytes
Process Data Size
102 Bytes
HARDWARE INFO
MAC Address Upper Port
00:02:a2:25:f8:18
MAC Address Lower Port
00:02:a2:25:f8:17
Part Number
9199300
Serial Number
20160
Hardware Revision
2
Production Date
week 41-2012
Firmware Version
2.5 build 16 revision 0 date 2011-4-14
Creating object dictionary. Please wait 38 s.

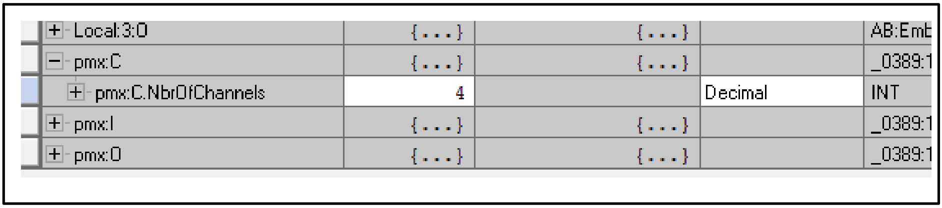
16.9 EtherNet/IP

16.9.1 Configuration

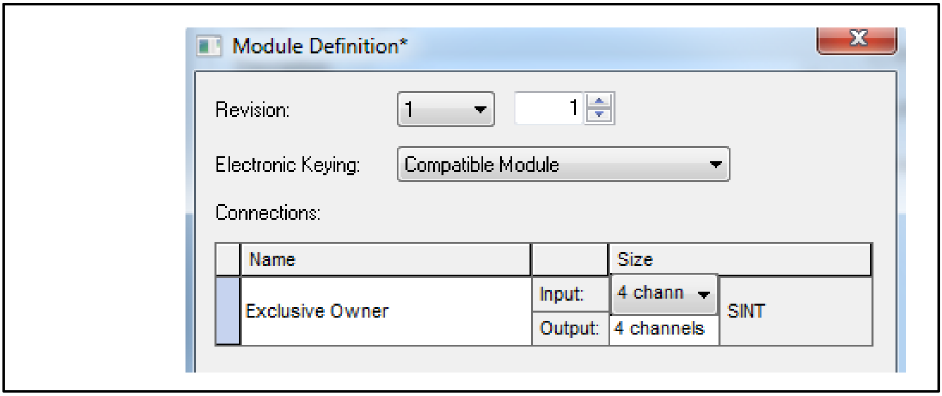
How to set the number the number of transmitted measurement channels:



1. Set the configuration object 199 "**NbrOfChannels**" (class 4, instance 199).
   That defines the number of channels that are copied into the data frame. Range 0..48.
   For example, this is how it looks like in RSLogix 5000



2. Select the size of both assembly instances 100 and 101. This number should match NbrOfChannels. Range 0..48 in steps of 4.
   For example, this is how it looks like in RSLogix 5000



Important

Examples regarding calculation channels can be found in the Technical notes on the supplied System CD and in [*http://www.hbm.com/en/menu/support/software-firmware-downloads/industrial-amplifiers/*](http://www.hbm.com/en/menu/support/software-firmware-downloads/industrial-amplifiers/)

16.9.2 Channel settings

The fitted measurement cards provide the channels in the order they are fitted into the PMX rack. The card in slot 1 provides data starting with channel 1. An empty slot does not provide any channel, the EtherNet/IP channels are populated gapless.
The calculated channels provide data behind 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 Use	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

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 transmitted channels data depends on the configuration, see below.

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	
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	
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	
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	
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	
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 transmitted channels data depends on the configuration, see below.

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
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

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

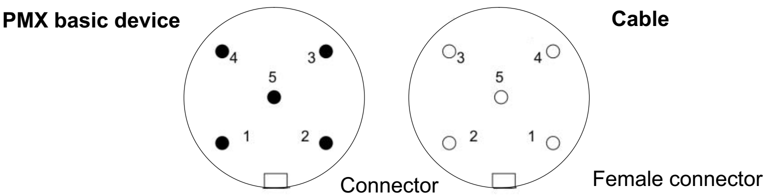
17 CAN interface (WGX001 only)

17.1 General

The basic device WGX001 has a CAN interface as per ISO11898. In combination with the CODESYS Soft-PLC, the PMX can be operated as a CANopen slave or CANopen master. To do this, a CAN component and then a CANopen Stack must be added. The corresponding PMX package and a collection of useful example programs for code generation, web visualization and inclusion of CANopen modules are also included.

The CD is included in the scope of supply or freely available at HBM: *www.hbm.com/en/menu/support/software-firmware-downloads/industrial-amplifiers*

17.2 CAN pin assignment



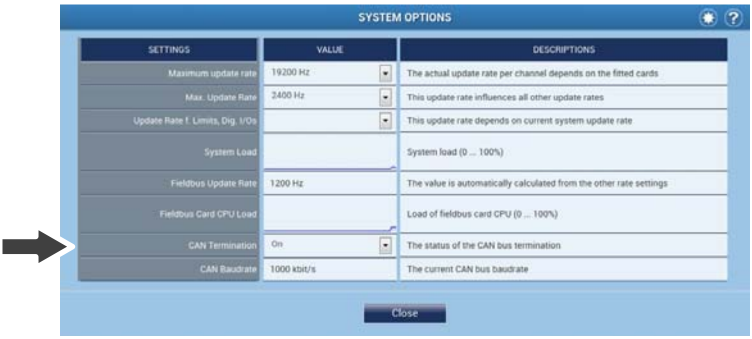
Pin	Signal	Description
1	SHLD	CAN shielding
2		Not in use
3	GND	Ground
4	CAN_H	CAN_H data lead (high)
5	CAN_L	CAN_L data lead (low)

Important

The node supply in the CAN network must be implemented separately and not via the CAN connection (female connector M12) of the PMX.

The bus termination is implemented via a 120 ohms terminating resistor at each end.

The terminating resistor is integrated in PMX and can be connected via the menu Settings>System>Device>System options>CAN termination.



17.3 CANopen master / slave operation

Master operation

Connected devices are not directly addressed during data transmission via CANbus. A unique identifier denotes the contents of a message (e.g. pressing force or pressing stroke).

The identifier also signals the priority of the message.
Message = Identifier + Signal + Additional information
Connected devices = Nodes.

In Master mode, CAN modules such as digiCLIP, PMX, SomatCR or third party devices can be connected via the CODESYS programming environment.

The CAN modules' device description files (EDS or DCF) are used for establishing the connection.
The bus speed must be identical in all CAN modules and is limited by the bus length. The transmission rate can be set to between 100 kbit and 1 Mbit in the CODESYS programming environment and then displayed in the "System options" PMX web browser menu.

Please contact the CAN module suppliers for information about how to set the bus speed.

Slave operation

In slave operation, PMX can transmit SDOs via EDS or DCF device description files and transmit PDOs of all measurement channels and calculation channels. A maximum of 128 PDO streams with a total maximum of 128 Byte data size and maximum 199 SDO*255 subID´s are available. The PDO streams can be sent triggered by timer-control up to min. 300 Hz or measured value-control up to 1.2 kHz or via SYNC message.

The SDOs and PDOs are generated and stored in the CODESYS programming environment. The transmission rate can be set in the CODESYS programming environment to between 100 kBit and 1 MBit.
This means that various SDO channels, module-dependent PDO Mapping and CAN Low Level libraries are available, then displayed in the PMX web browser menu "System options".

This means that various SDO channels, module-dependent PDO Mapping and CAN Low Level libraries are available.

A maximum of 30 CAN data can in turn be made available as measured values via the PMX calculation channel "Connection with (CODESYS)" in PMX and are immediately "time-stamped".
This enables directly acquired measured quantities and CAN messages to be acquired and analyzed in parallel and synchronously in the entire system.

Important

The CANbus must be terminated at both ends and the appropriate baud rate set for all bus nodes. In PMX, this is implemented by the menu selection: System-Options.

Error states

If errors occur when the CAN interface is active, they can be indicated in the CODESYS application. These errors are also recorded in the log file in the PMX (see Section 23.7.2 Log entries). There is no direct CAN error display via the PMX browser.

18 CODESYS-V3 Soft PLC (WGX001 only)

18.1 General

The basic housing WGX001, together with the software platform CODESYS V3, makes PMX suitable for numerous tasks in industrial automation technology. It contains everything needed for programming, fieldbus and I/O configuration, visualization, MotionControl and other tasks. The basis of the CODESYS V3 software platform is the IEC 61131-3 programming system. All IEC-61131-3 programming languages are supported.

PMX and CODESYS V3 means that applications can not only be automated, they can also be displayed and operated in realtime. The appropriate web visualization is created in the CODESYS software and runs together with the application in PMX. The visualization can be used on all browser-based devices via the Ethernet-TCP/IP interface of the amplifier.

A CODESYS runtime license is already included in PMX with the basic device WGX001. The supplied PMX-CODESYS CD contains the CODESYS software V3. The corresponding PMX package and a collection of useful example programs for code generation, web visualization and inclusion of CANopen modules are also included.

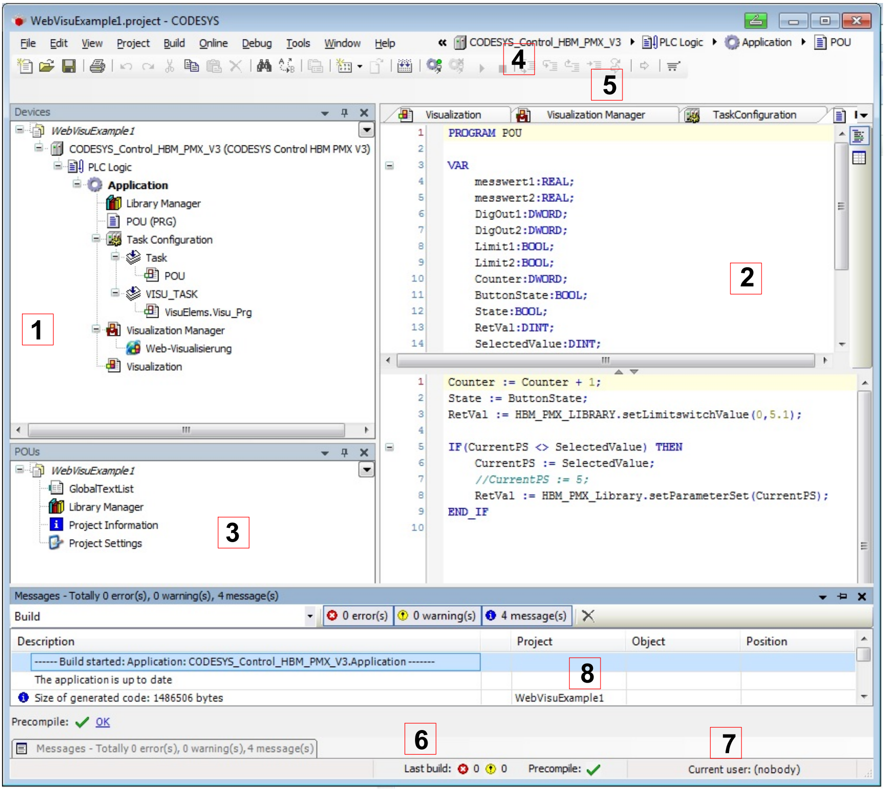
The CD is included in the scope of supply or freely available at HBM: *www.hbm.com/en/menu/support/software-firmware-downloads/industrial-amplifiers*

Important

A CODESYS application or web visualization running on the PMX will also be retained after a firmware update or if a card is changed (firmware 2.00 or higher). Please note that signal mapping in CODESYS is fixed and if the measuring cards are moved or the PMX computing channels are changed, the signal mapping must be checked and corrected if necessary.
A running application can be stopped directly in the CODESYS development environment. As from firmware V3.00, CODESYS applications and visualizations in the "CODESYS" menu can be started, stopped, reset and deleted individually.
CODESYS projects transferred to a PMX via a CODESYS development environment can be saved and loaded to a PC via this menu.

18.2 CODESYS development environment

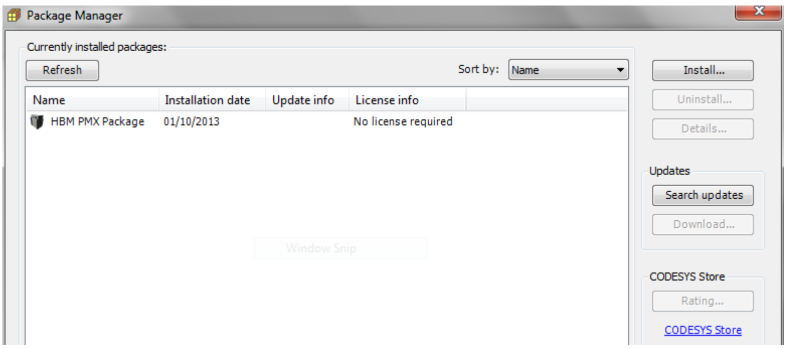
The operator interface provides menus and toolbars, windows for editor views, object organization, monitoring and message output as well as an information and status line.



No.	Meaning
1	Device window
2	Editor window
3	POU window (program organization unit)
4	Menu bar
5	Toolbar
6	Info Position Editor
7	Info active user
8	Message window

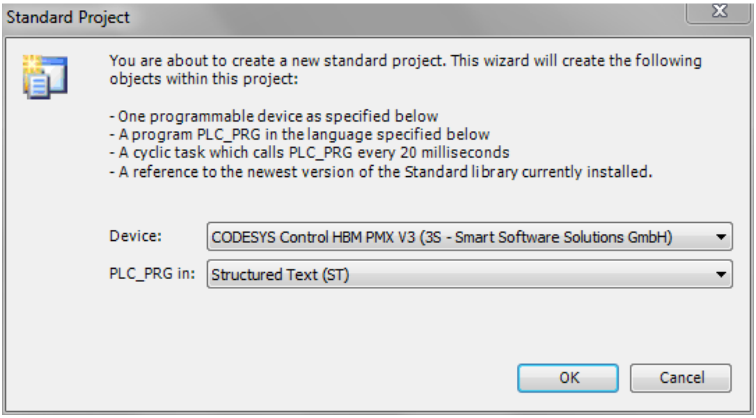
18.3 Preparation

- Install "Setup_CODESYS…..exe"
- Start CODESYS with administrator rights. (To do this, e.g. click on the symbol with the shift key and right mouse button, then select Run as administrator.)
- Install the PMX package:
  In the menu Tools – Package Manager – Install0, find and select the file "hbm-pmx.package" on the CD. Select Typical Setup. The Package Manager now contains the PMX package:



18.4 Create project

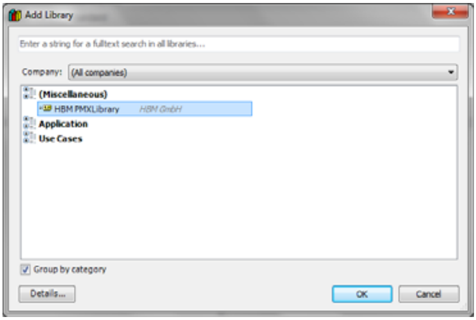
- Select File – New project – Standard project.
  Select "CODESYS Control HBM PMX V3" as the device type:



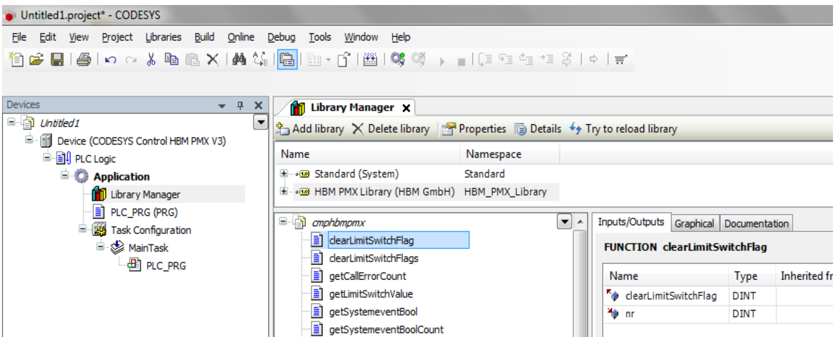
- After creating the project, double-click on "CODESYS Control HBM PMX V3"
  Add a gateway if necessary in the communication settings. (The gateway type is usually "TCP/IP" if PMX is connected via Ethernet with the PC. IP address: "localhost" or permanent device address or PMX device name.)
- Mark gateway entry and click on Scan Network. The target device should now be displayed. Double-click to make it the active device.
- The program can be created under "PLC_PRG (PRG)".

18.5 Add PMX library

- Double-click on "Library Manager", then click on "Add library" and select "HBM PMXLibrary” under "Miscellaneous"



- The library functions are explained in the Online help, e.g.



18.6 PMX library

Description of PMX reference library functions, version 0.94.

Function: clearLimitSwitchFlag

Clears one limitswitch flag.

Name	Data type	Inherited from	Adress	Initial value	Comment
clearLimitSwitchFlag	DINT				
nr	DINT				nr of limit switch flag to clear. Valid: 0 … 32

Function: clearLimitSwitchFlags

Clears a number of limit switches

Name	Data type	Inherited from	Adress	Initial value	Comment
clearLimitSwitchFlags	DINT				
mask	DWORD				Bitmask: every limit switch flag is cleared where corresponding bit is set

Function: diskfree

Get available storage space

Name	Data type	Inherited from	Adress	Initial value	Comment
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 occured during function calls which return a call handle. This function should always return zero on normal operation.

Name	Data type	Inherited from	Adress	Initial value	Comment
getCallErrorCount	DINT				

Function: GetLimitSwitchValue

Returns the limit switch value. The limitswitch value is the limit on which the limitswitch flag gets set.

Name	Data type	Inherited from	Adress	Initial value	Comment
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**	**Inherited from**	**Adress**	**Initial value**	**Comment**
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 in context of a Task -> External Event -> System Event. The number of arguments depends on the type of the signal determined by the getSystemeventNr function. This function returns the n-th argument of type "BOOL" for the corresponding systemevent. Every PMX device carries a xml file describing the valid systemevents to be retrieved via http://pmx/data/systemevent.xml.

Name	Data type	Inherited from	Adress	Initial value	Comment
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<>

Function: GetSystemeventBoolCount

This function is only valid in context of a Task -> External Event -> System Event. The number of arguments depends on the type of the signal determined by the getSystemeventNr function. This function returns the available number "BOOL"-type arguments for the current systemevent. Every PMX device carries a xml file describing the valid systemevents to be retrieved via http://pmx/data/systemevent.xml.

Name	Data type	Inherited from	Adress	Initial value	Comment
Get SystemeventBool-Count	BYTE				

Function: GetSystemeventByte

This function is only valid in context of a Task -> External Event -> System Event. The number of arguments depends on the type of the signal determined by the getSystemeventNr function. This function returns the n-th argument of type "BYTE" for the corresponding systemevent. Every PMX device carries a xml file describing the valid systemevents to be retrieved via http://pmx/data/systemevent.xml.

Name	Data type	Inherited from	Adress	Initial value	Comment
Get SystemeventByte	BYTE				
idx	DINT				Index value of the n-th Byte of the systemevent. Possible values: 0<=idx<5 and idx<>

Function: GetSystemeventByteCount

This function is only valid in context of a Task -> External Event -> System Event. The number of arguments depends on the type of the signal determined by the getSystemeventNr function. This function returns the available number "BYTE"-type arguments for the current systemevent. Every PMX device carries a xml file describing the valid systemevents to be retrieved via http://pmx/data/systemevent.xml.

Name	Data type	Inherited from	Adress	Initial value	Comment
getSystemeventByteCount	DINT				

Function: GetSystemeventDInt

This function is only valid in context of a Task -> External Event -> System Event. The number of arguments depends on the type of the signal determined by the getSystemeventNr function. This function returns the n-th argument of type "DINT" for the corresponding systemevent. Every PMX device carries a xml file describing the valid systemevents to be retrieved via http://pmx/data/systemevent.xml.

Name	Data type	Inherited from	Adress	Initial value	Comment
getSystemeventDint	DINT				
	DINT				Index value of the n-th DINT of the systemevent. Possible values: 0<=idx<5 and idx<>

Function: GetSystemeventDIntCount

This function is only valid in context of a Task -> External Event -> System Event. The number of arguments depends on the type of the signal determined by the getSystemeventNr function. This function returns the available number "DINT"-type arguments for the current systemevent. Every PMX device carries a xml file describing the valid systemevents to be retrieved via http://pmx/data/systemevent.xml.

Name	Data type	Inherited from	Adress	Initial value	Comment
clearLimitSwitchFlag					

Function getSystemeventInt

This function is only valid in context of a Task -> External Event -> System Event. The number of arguments depends on the type of the signal determined by the getSystemeventNr function. This function returns the n-th argument of type "INT" for the corresponding systemevent. Every PMX device carries a xml file describing the valid systemevents to be retrieved via http://pmx/data/systemevent.xml.

Name	Data type	Inherited from	Adress	Initial value	Comment
getSystemeventInt	INT				
idx	DINT				Index value of the n-th Int of the systemevent. Possible values: 0<=idx<5 and idx<>

Function getSystemeventIntCount

This function is only valid in context of a Task -> External Event -> System Event. The number of arguments depends on the type of the signal determined by the getSystemeventNr function. This function returns the available number "INT"-type arguments for the current systemevent. Every PMX device carries a xml file describing the valid systemevents to be retrieved via http://pmx/data/systemevent.xml.

Name	Data type	Inherited from	Adress	Initial value	Comment
getSystemeventInt-Count	DINT				

Function: GetSystemeventLInt

This function is only valid in context of a Task -> External Event -> System Event. The number of arguments depends on the type of the signal determined by the getSystemeventNr function. This function returns the n-th argument of type "LINT" for the corresponding systemevent. Every PMX device carries a xml file describing the valid systemevents to be retrieved via http://pmx/data/systemevent.xml.

Name	Data type	Inherited from	Adress	Initial value	Comment
getSystemeventInt	INT				
idx	DINT				Index value of the n-th Int of the systemevent. Possible values: 0<=idx<5 and idx<>

Function: GetSystemeventLIntCount

This function is only valid in context of a Task -> External Event -> System Event. The number of arguments depends on the type of the signal determined by the getSystemeventNr function. This function returns the available number "LINT"-type arguments for the current systemevent. Every PMX device carries a xml file describing the valid systemevents to be retrieved via http://pmx/data/systemevent.xml.

Name	Data type	Inherited from	Adress	Initial value	Comment
getSystemeventIntCount	DINT				

Function: GetSystemeventLReal

This function is only valid in context of a Task -> External Event -> System Event. The number of arguments depends on the type of the signal determined by the getSystemeventNr function. This function returns the n-th argument of type "REAL" for the corresponding systemevent. Every PMX device carries a xml file describing the valid systemevents to be retrieved via http://pmx/data/systemevent.xml.

Name	Data type	Inherited from	Adress	Initial value	Comment
get-SystemeventLReal	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<>

Function: GetSystemeventLRealCount

This function is only valid in context of a Task -> External Event -> System Event. The number of arguments depends on the type of the signal determined by the getSystemeventNr function. This function returns the available number "REAL"-type arguments for the current systemevent. Every PMX device carries a xml file describing the valid systemevents to be retrieved via http://pmx/data/systemevent.xml.

Name	Data type	Inherited from	Adress	Initial value	Comment
get-SystemeventLReal-Count	DINT				

Function: GetSystemeventNr

This function is only valid in context of a Task -> External Event -> System Event. This function returns the systemeventnr for the corresponding systemevent. Every PMX device carries a xml file describing the valid systemevents to be retrieved via http://pmx/data/systemevent.xml.

Name	Data type	Inherited from	Adress	Initial value	Comment
getSystemeventNr	DINT				

Function: GetSystemeventString

This function is only valid in context of a Task -> External Event -> System Event. The number of arguments depends on the type of the signal determined by the getSystemeventNr function. This function returns the n-th argument of type "STRING" for the corresponding systemevent. Every PMX device carries a xml file describing the valid systemevents to be retrieved via http://pmx/data/systemevent.xml.

Name	Data type	Inherited from	Adress	Initial value	Comment
getSystemeventString	STRING				
idx	DINT				Index value of the n-th String of the systemevent. Possible values: 0<=idx<5 and idx<>

Function: GetSystemeventStringCount

This function is only valid in context of a Task -> External Event -> System Event. The number of arguments depends on the type of the signal determined by the getSystemeventNr function. This function returns the available number "STRING"-type arguments for the current systemevent. Every PMX device carries a xml file describing the valid systemevents to be retrieved via http://pmx/data/systemevent.xml.

Name	Data type	Inherited from	Adress	Initial value	Comment
getSystemevent-StringCount	DINT				

Function: GetSystemeventUDInt

This function is only valid in context of a Task -> External Event -> System Event. The number of arguments depends on the type of the signal determined by the getSystemeventNr function. This function returns the n-th argument of type "UDINT" for the corresponding systemevent. Every PMX device carries a xml file describing the valid systemevents to be retrieved via http://pmx/data/systemevent.xml.

Name	Data type	Inherited from	Adress	Initial value	Comment
getSystemeventUDint	UDINT				
idx	DINT				Index value of the n-th UDint of the systemevent. Possible values: 0<=idx<5 and idx<>

Function: GetSystemeventUDIntCount

This function is only valid in context of a Task -> External Event -> System Event. The number of arguments depends on the type of the signal determined by the getSystemeventNr function. This function returns the available number "UDINT"-type arguments for the current systemevent. Every PMX device carries a xml file describing the valid systemevents to be retrieved via http://pmx/data/systemevent.xml.

Name	Data type	Inherited from	Adress	Initial value	Comment
getSystemeventUDint-Count	DINT				

Function: GetSystemeventUInt

This function is only valid in context of a Task -> External Event -> System Event. The number of arguments depends on the type of the signal determined by the getSystemeventNr function. This function returns the n-th argument of type "UINT" for the corresponding systemevent. Every PMX device carries a xml file describing the valid systemevents to be retrieved via http://pmx/data/systemevent.xml.

Name	Data type	Inherited from	Adress	Initial value	Comment
getSystemeventUInt	UINT				
idx	DINT				Index value of the n-th UIint of the systemevent. Possible values: 0<=idx<5 and idx<>

Function: GetSystemeventUIntCount

This function is only valid in context of a Task -> External Event -> System Event. The number of arguments depends on the type of the signal determined by the getSystemeventNr function. This function returns the available number "UINT"-type arguments for the current systemevent. Every PMX device carries a xml file describing the valid systemevents to be retrieved via http://pmx/data/systemevent.xml.

Name	Data type	Inherited from	Adress	Initial value	Comment
getSystemeventUIntCount	DINT				

Function: GetSystemeventULInt

This function is only valid in context of a Task -> External Event -> System Event. The number of arguments depends on the type of the signal determined by the getSystemeventNr function. This function returns the n-th argument of type "LINT" for the corresponding systemevent. Every PMX device carries a xml file describing the valid systemevents to be retrieved via http://pmx/data/systemevent.xml.

Name	Data type	Inherited from	Adress	Initial value	Comment
getSystemeventULint	ULINT				
idx	DINT				Index value of the n-th ULint of the systemevent. Possible values: 0<=idx<5 and idx<>

Function: GetSystemeventULIntCount

This function is only valid in context of a Task -> External Event -> System Event. The number of arguments depends on the type of the signal determined by the getSystemeventNr function. This function returns the available number "ULINT"-type arguments for the current systemevent. Every PMX device carries a xml file describing the valid systemevents to be retrieved via http://pmx/data/systemevent.xml.

Name	Data type	Inherited from	Adress	Initial value	Comment
getSystemeventULint-Count	DINT				

Function: IsFinished

For all functions returning a handle the state of the call can be queried. The function returns TRUE if the coresponding function call to the handle is finished.

Name	Data type	Inherited from	Adress	Initial value	Comment
callHandle	DINT				Handle of the corresponding function call e.g.recalibrate.

Function: Recalibrate

Recalibrate the coresponding hardware channel. This function only works if the channel is populated and has calibration hardware available.

Name	Data type	Inherited from	Adress	Initial value	Comment
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

Set a two point characteristic curve for a signal.

Name	Data type	Inherited from	Adress	Initial value	Comment
Set2Pointcharacteristic	DINT				
slot	DINT				
signal	DINT				(* slot of the hardware (valid 1-4 depending on hardware)*)
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/releases a peak value.

Name	Data type	Inherited from	Adress	Initial value	Comment
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

Set the limit switch value. The limitswitch value is the limit on which the coresponding limitswitch flag gets set.

Name	Data type	Inher-ited from	Adress	Initial value	Comment
setLimitswitchValue	DWORD				
nr	DINT				Nr of the limit switch starting with 0
value	REAL				New value of the limit switch

Function: SetParameterSet

Set the current parameterset. The current parameterset is available via the HBM PMX CODESYS IO. The parametersets have to be configured via web interface beforehand. This function returns a handle which can be queried via the isFinished function. It is nevertheless possible that the parameter switch is not finished on the completion of this call as this function only triggers the start of the parameterset switching. Use systemevent where systemeventnr = 2000 to get a trigger if parameter set switching was successful.

Name	Data type	Inherited from	Adress	Initial value	Comment
setParameterSet	DINT				
paremeternr	DINT				The parameterof the desires parameterset

Function: SetResetPeak

Reset the peak value. This function should be called two times for a full reset cycle.

Name	Data type	Inherited from	Adress	Initial value	Comment
setResetPeak	DINT				
slot	DINT				Slot of peak value (valid 1-4 depending on hardware).

Name	Data type	Inherited from	Adress	Initial value	Comment
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

Set the Shunt state of a signal.

Name	Data type	Inherited from	Adress	Initial value	Comment
setShuntState	DINT				Handle: check with isFin-ished(handle)
slot	DINT				The slot to mod-ify, 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

Set offset such that measval gets zero. Be aware that this function affects current parameterset. Revert this with setUserOffset(...,0.0)

Name	Data type	Inherited from	Adress	Initial value	Comment
setToZero	DINT				
slot	DINT				Slot of corres-ponding measval (valid 1-4)
signal	DINT				Signal of corres-ponding measval (valid 1-4 de-pending on hard-ware)

Function: setUserOffset

Set a custom measval offset. Be aware that this function affects the current parameter set.

Name	Data type	Inherited from	Adress	Initial value	Comment
setUserOffset	DINT				
slot	DINT				Slot of correspond-ing measval, use 9 for computed channels
signal	DINT				Signal of corres-ponding measval, starting with 1
offset	REAL				The new offset value

Function: setUserOffset

Set a custom measval offset. Be aware that this function affects the current parameter set.

Name	Data type	Inherited from	Adress	Initial value	Comment
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

Specifying a Zero Target Value allows you to add a constant value to the currently measured value for a specified signal.

Name	Data type	Inherited from	Adress	Initial value	Comment
startZeroTargetValue	DINT				
slot	DINT				
signal	DINT				(* slot of the hardware (shoukld be 9 for calculated channels) *)
value	DINT				(* nr of the calculated channel *)

Function: startGetShuntState

Initiate a change in the shunt state of a PX460.

Name	Data type	Inherited from	Adress	Initial value	Comment
startGetShuntState	DINT				Handle: query with get-ShuntState(han-dle)
slot	DINT				Sthe slot to mod-ify, valid 1-4 and Card PX460 only
signal	DINT				The signal to modify valid 2,4

Function: startLedEffect

Various LED Effects e.g. to localize the PMX device or give some feedback to the user in front of the device.

Name	Data type	Inherited from	Adress	Initial value	Comment
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: startTedsSetup

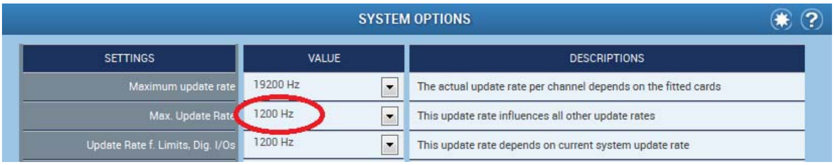
Reinitialize the Teds setup for the Specified channel.

Name	Data type	Inherited from	Adress	Initial value	Comment
StartTedsSetup	DINT				Handle which can be queried by isFinished
slot	DINT				Slot of the hard-ware (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 Main-Task:

- **Cyclic**
  The task is started asynchronously to the measured values. **The interval must be at least 4 ms**, that is the shortest possible resolution.
- **External - Measval Event**
  The task is started synchronously with the acquired measured values. The calling frequency is set in the "System Options" dialog. Default: 1,200 Hz, i.e. at a sampling rate of 19,200 Hz, the task is started after every 16th measured value.



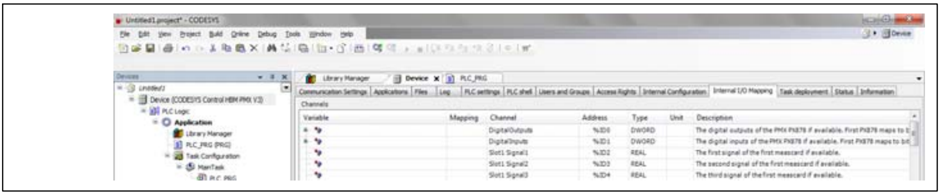
- **External - SystemEvent**
  The task is started when a PMX system event occurs (this event is also displayed in the device system log). The number of the event is delivered within the task by the library function getSystemeventNo (see chapter 18.10). .
  The system events can be called up via the browser with the device path http://<pmx devicename>/data/systemevent.xml.

Where possible, only select these task types

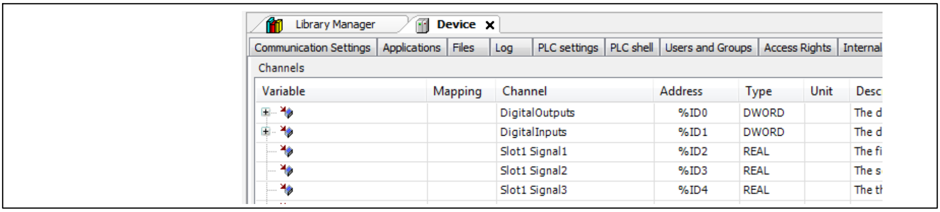
18.8 Cyclic data

The data exchanged cyclically with the PMX firmware are displayed as follows:

- Double-click on the left in the project tree "CODESYS Control HBM PMX V3".
  Select the tab "Internal I/O Mapping".



- To connect with an existing program variable of the same type, double-click in the "Variable" column in the required cell.

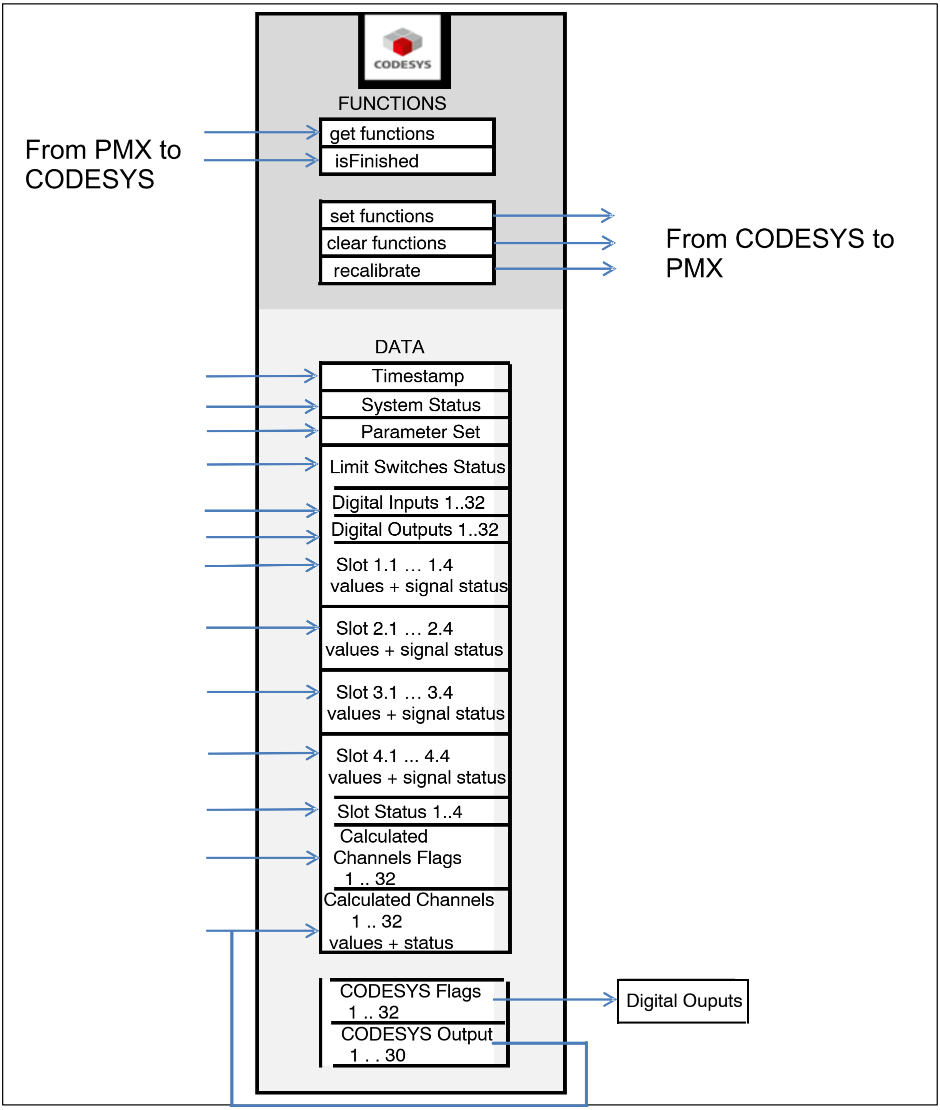


18.9 Signal run diagram (I/O-Mapping)

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
CODESYS Output 1 . . 30
Digital Ouputs

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 list of events, enter the following call in the browser line:
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:firmwareUpdate
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: interger:%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: Administator passwort reset failed: invalid signature!

Id: **1014** Name:pwResetFileError
Description: Administator passwort reset failed: file operation failed!

Id: **1015** Name:pwResetFormatError
Description: Administrator passwort reset failed: file format invalid!

Id: **1016** Name:pwResetHostnameError
Description: Administrator passwort reset failed: hostname does not match!

Id: **1017** Name:pwResetMacError
Description: Administrator passwort reset failed: mac does not match!

Id: **1018** Name:pwResetSuccessful
Description: Administrator passwort reset successfull!

Id: **1019** Name:codesysFileRemoved
Argument:deletedCODESYSFile Type:string
Description: The codesys application file:%1 has been deleted!

Id: **1020** Name:canError
Argument:errorString Type:string
Argument: error Type: enum
Description: CAN error occured: %1

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?.

18.10.5com.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: physSignalVal_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: tara 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.

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 field-bus.

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. Connectioons requested. Oldest terminated. Last activity %2s ago.

Id: **8003** Name:newConnectionEstablished
Description: New Eth. Connection on port 55000 establis-hed.

Id: **8004** Name:connectionClosed
Description: Eth. connection closed.

18.10.10 com.hbm.meassrv

Id: **9001** Name:bufferOverrun
Description: Buffer overrun occured.

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.

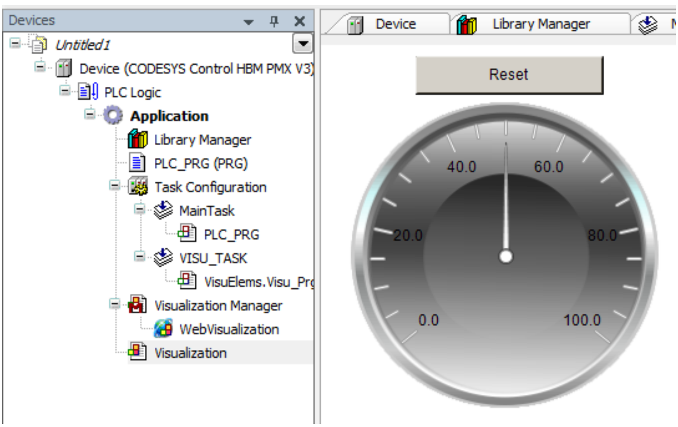
Id: **12020** Name:dataloggerStartRequested
Description: Datalogger start requested.

Id: **12021** Name:dataloggerDirectoryRemoved
Argument: filename Type:string
Description: Datalogger start requested %1.

18.11 WebVisualisation

CODESYS in PMX contains the WebVisu. This means that the process can be visualized and controlled via a freely configurable website. The web server runs here in CODESYS in PMX.

- Right-click in the project tree on "Application". Then "Add Object" – "Visualization".
- Graphic elements can now be added and connected with program variables. Example:



- After starting WebVisu in PMX, the website of a web browser is accessible under

  pmx:8080/webvisu.htm.

  "pmx" here is the device name, it may be necessary to replace it with the actual device name or an IP address. "webvisu.htm" is the default name set by

CODESYS. It can be changed in the Visualization Manager.

- In the PMX user interface, the WebVisu is linked in the footer via the CODESYS symbol. The prerequisite for this is the default name "webvisu.htm"

Tip

Examples regarding WebVisu can be found in the Technical notes on the supplied System CD and in *www.hbm.com/en/menu/support/software-firmware-downloads/industrial-amplifiers*

18.12 CAN interface

The device can be run with CODESYS as a CANOpen slave or master. To do this, a CAN component and then a CANOpen Stack must be added. Several examples are supplied in the package for this purpose.

Tip

Examples regarding CODESYS can be found in the Technical notes on the supplied System CD and in *www.hbm.com/en/menu/support/software-firmware-downloads/industrial-amplifiers*

18.13 CAN-Master and Slave mode

Short description

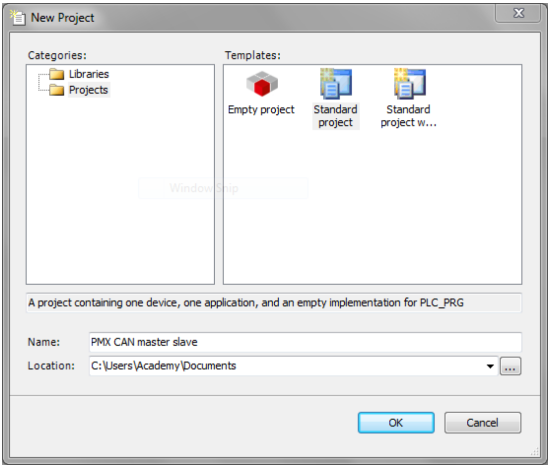
This is a guide to starting CODESYS applications for PMX. Basic experience with CODESYS is assumed. Experienced users are free to adopt a different approach. Further help is available from the examples that are installed as standard on the desktop when the PMX package is installed and from the online Help for the package.

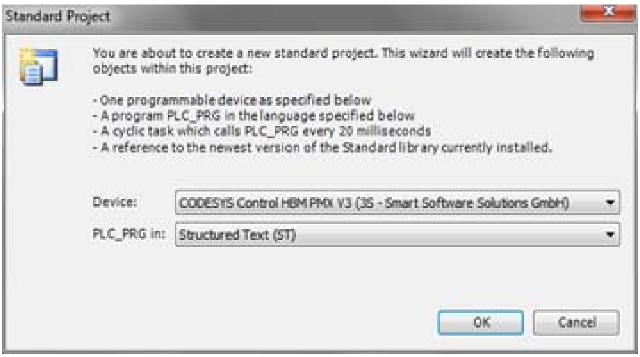
The necessary files are on the "PMX CODESYS" CD that is included in the scope of supply of every PMX containing CODESYS or can be downloaded from the support page at hbm.com.

In this example, two PMX are linked via the CANopen interface. One PMX operates as master, the second PMX as slave in the network. In the second PMX, a PDO with 4 measured values is created which transmits the measured values to the first PMX (master) for display in 4 calculation channels.

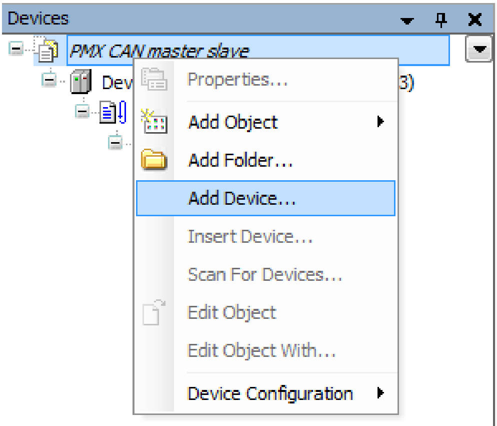
Start

- Open CODESYS development software. Create a standard project and select the PMX as device “CODESYS Control HBM PMX V3”.

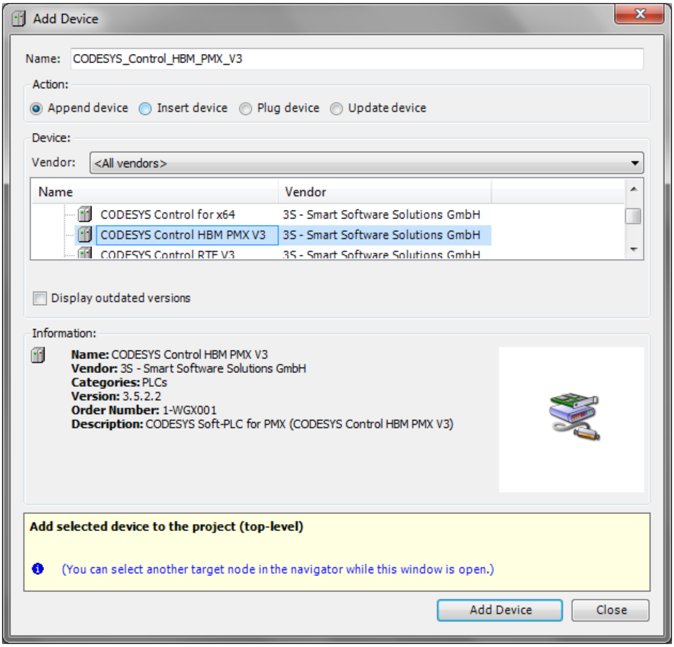




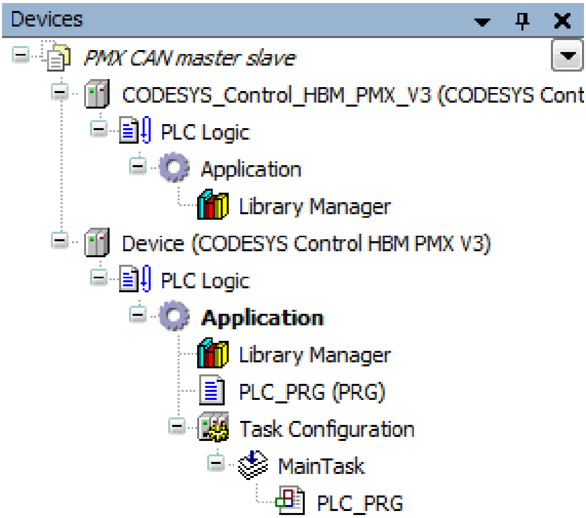
- When you have opened the project, right-click on the file name in the device structure on the left-hand side and select "Add Device"..



- …and select another PMX.

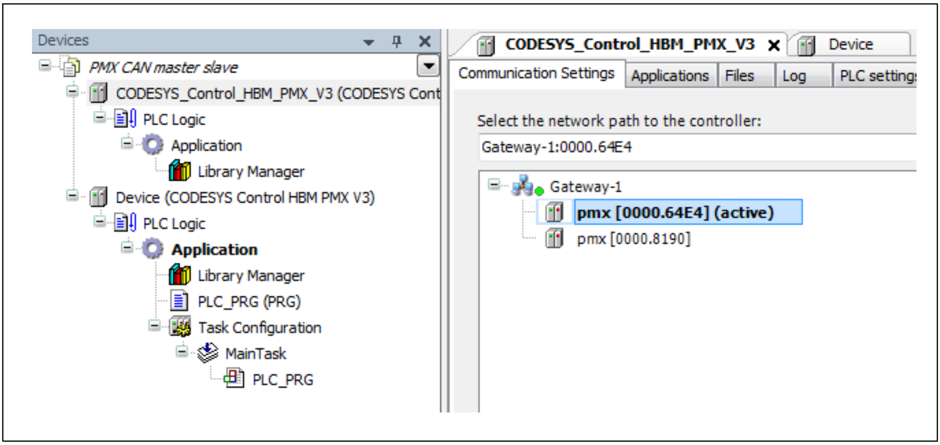


- This results in the following structure with two PMX devices.

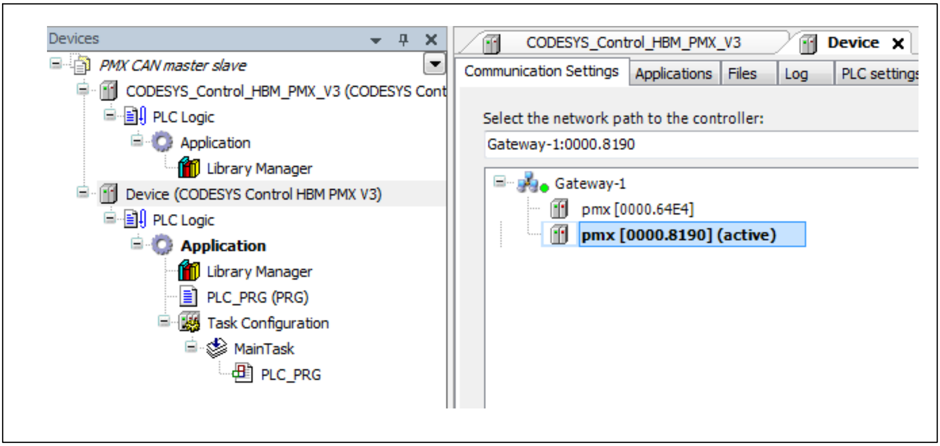

Devices
PMX CAN master slave
CODESYS_Control_HBM_PMX_V3 (CODESYS Cont
PLC Logic
Application
Library Manager
Device (CODESYS Control HBM PMX V3)
PLC Logic
Application
Library Manager
PLC_PRG (PRG)
Task Configuration
MainTask
PLC_PRG

Activate gateways

- Double-click the first PMX (CODESYS_Control). In the window that opens on the right, double-click the gateway to update the status. Then double-click on one of the PMX devices (in this case: [0000.64E4]) to activate the gateway.

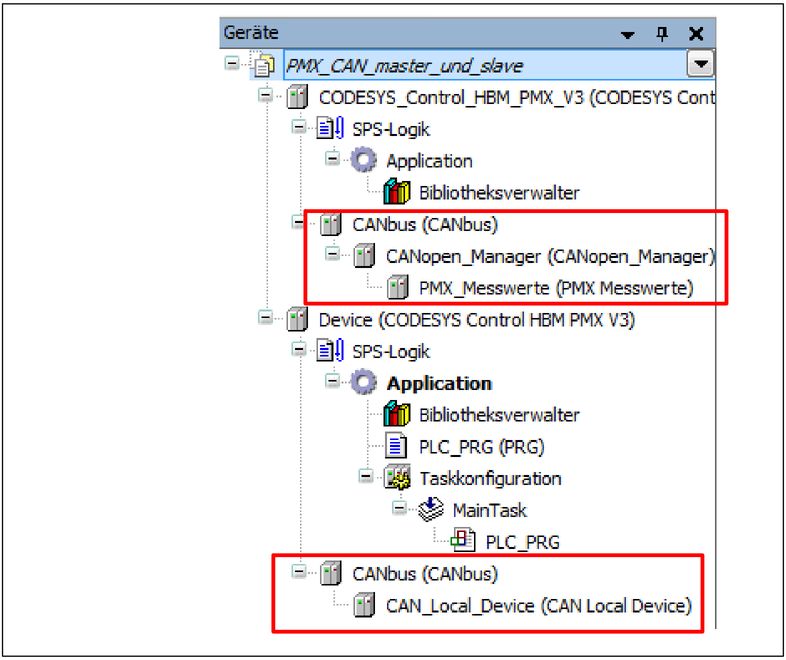


- Repeat the procedure for the second PMX (Device) in the structure and assign in accordance with the other PMX (in this case: [0000.8190]).


Devices
PMX CAN master slave
CODESYS_Control_HBM_PMX_V3 (CODESYS Cont
PLC Logic
Application
Library Manager
Device (CODESYS Control HBM PMX V3)
PLC Logic
Application
Library Manager
PLC_PRG (PRG)
Task Configuration
MainTask
PLC_PRG
CODESYS_Control_HBM_PMX_V3
Device
Communication Settings
Applications
Files
Log
PLC settings
Select the network path to the controller:
Gateway-1:0000.8190
Gateway-1
pmx [0000.64E4]
pmx [0000.8190] (active)

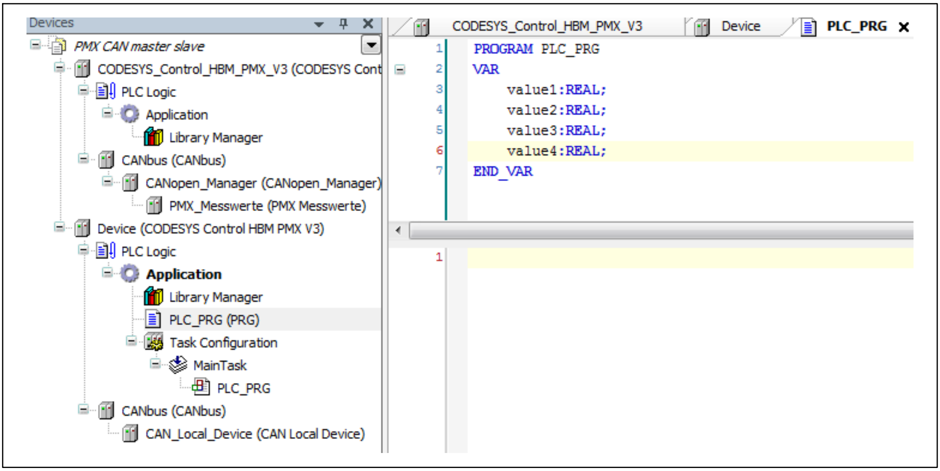
Add CAN devices

- *For master mode*: For CODESYS_Control_HBM_PMX_V3, add a CANbus, a CANopen_Manager and PMX_Messwerte using the "Add Device" menu.
- *For slave mode*: For Device, add a CANbus and a CAN_Local_Device.
- Set the baud rate for the CANbuses. (in this case: 100 000 bits/s)

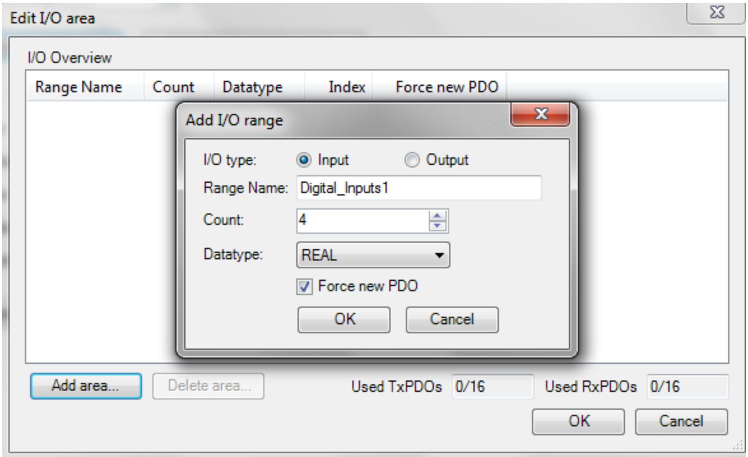


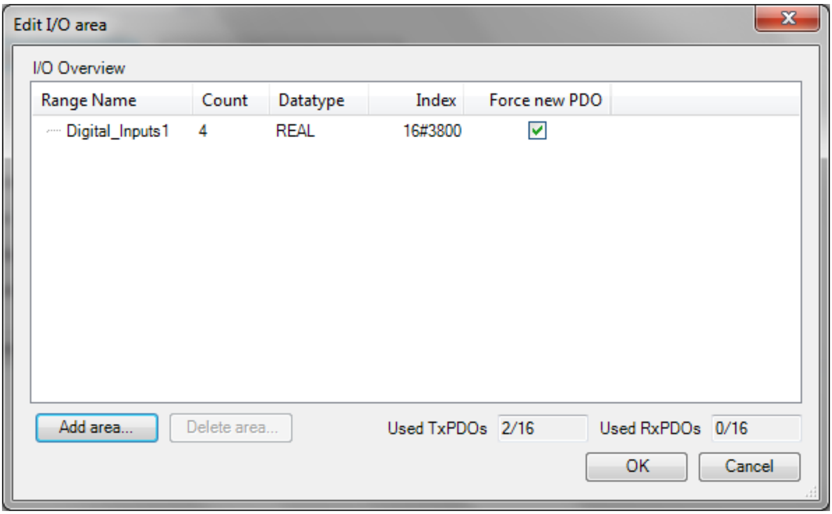
Declaration of variables and mapping to the device

- Open the PLC-PRG tab via the “Device” and declare variables as shown below.

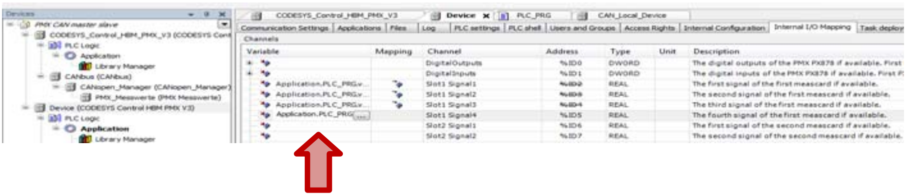


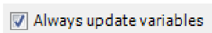
- Double-click CAN_Local_Device. In the window that opens, click the "Edit I/O area" button. In the "Edit I/O area" window click "Add area..." and add an area as shown below.


Edit I/O area
I/O Overview
Range Name
Count
Datatype
Index
Force new PDO
Add I/O range
I/O type:
Input
Output
Range Name:
Digital_Inputs1
Count:
4
Datatype:
REAL
Force new PDO
OK
Cancel
Add area...
Delete area...
Used TxPDOs
0/16
Used RxPDOs
0/16
OK
Cancel

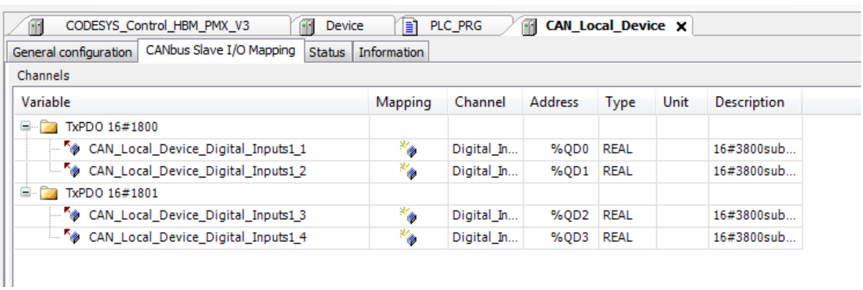

Edit I/O area
I/O Overview
Range Name
Count
Datatype
Index
Force new PDO
Digital_Inputs1
4
REAL
16#3800
Add area...
Delete area...
Used TxPDOs
2/16
Used RxPDOs
0/16
OK
Cancel

- Double-click “Device” or open the tab. Click the "Internal I/O Mapping" tab. In the Variable column, double-click the cell to open for mapping a variable to a slot.

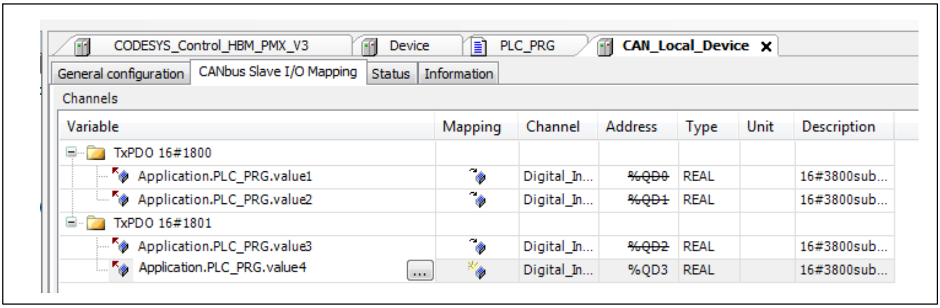


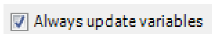
- Please note: Always activate the checkbox in the lower right corner 

- Open "CAN_Local_Device" again and click the "CANbus Slave I/O Mapping“ tab...

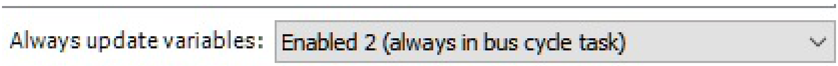


- ... map the variables in this case too.



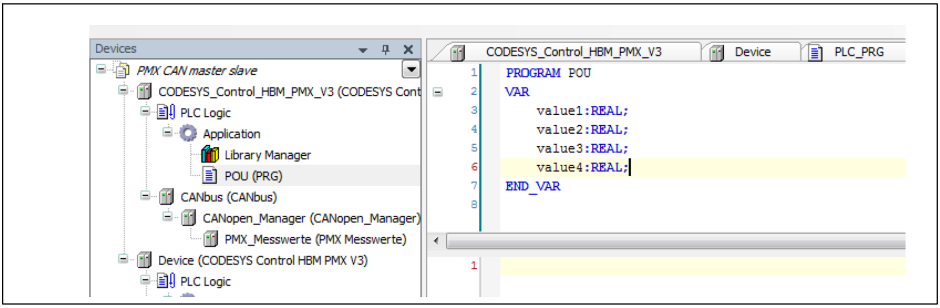
- Please note: Always activate the checkbox in the lower right corner 

resp.

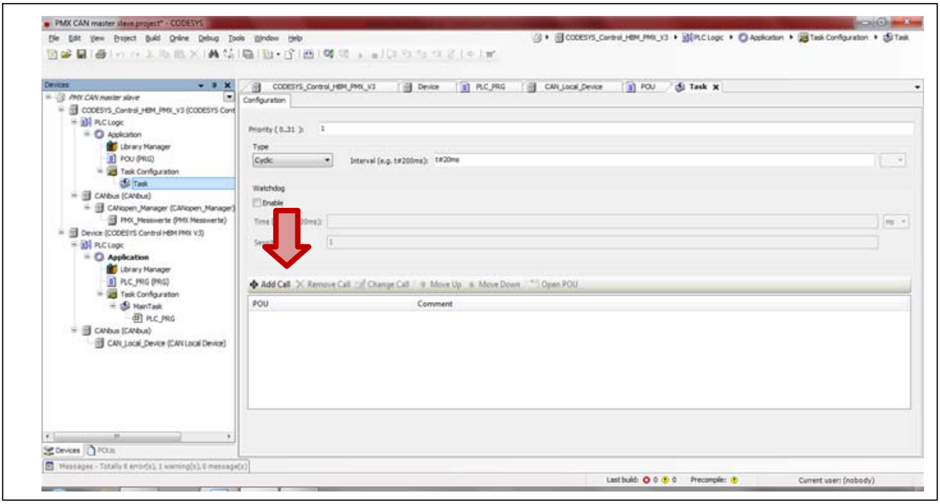


Declaration variables and mapping to CODESYS_Control

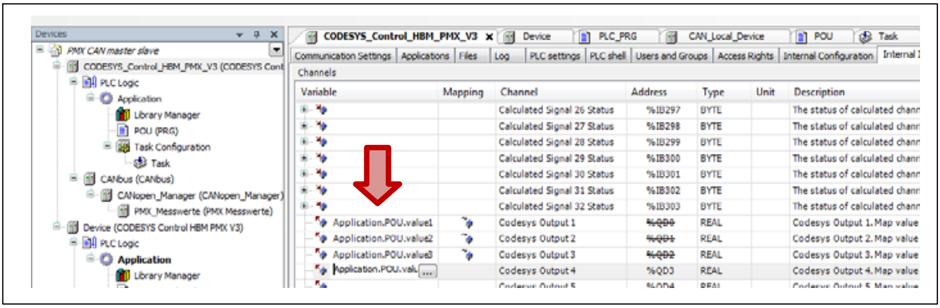
- Right-click Application under CODESYS_Control_HBM_PMX_V3, scroll to "Add Object" and select "POU". In this case, declare the variables analogously to the device declaration.

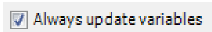


- Right-click Application and select "Add Object" and select a task configuration.

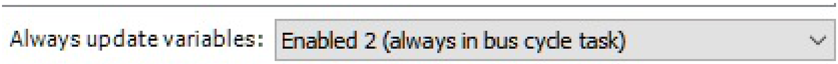


- In the window on the right, select "Add Call" and select "POU".
- Double-click CODESYS_Control_HBM_PMXV3 or open the tab. Click the "Internal I/O Mapping" tab. In the Variable column, double-click the cell to open for mapping of a variable to the corresponding CODESYS output channel.

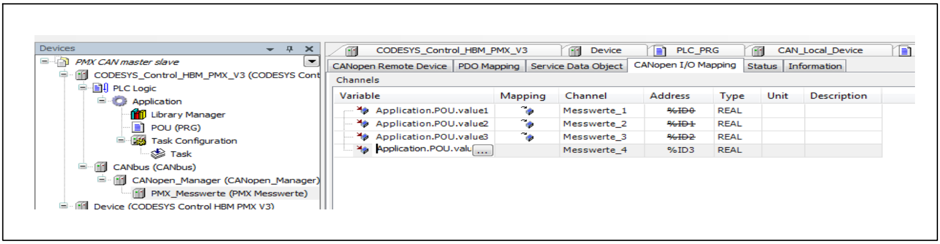


- Please note: Always activate the checkbox in the lower right corner 

resp.

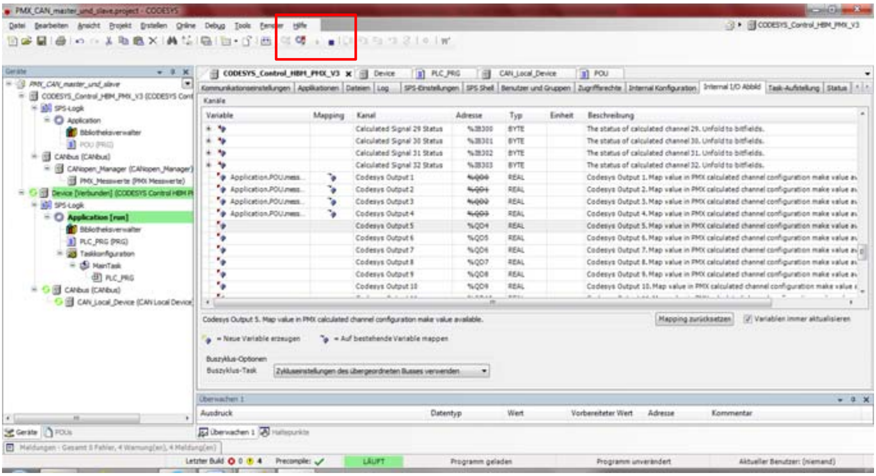


- Map the variables under “PMX_Messwerte”.

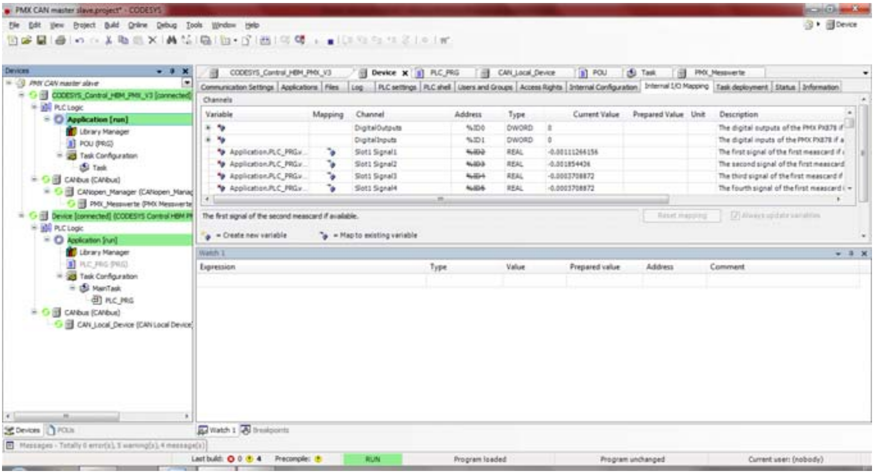


Run programs

- Click Device and log in; then press F5 to start the application The following screen is displayed.

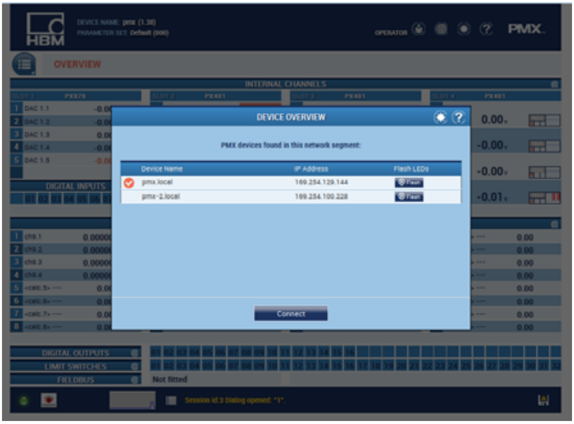


- Now log out (important: without stopping the application).
- Right-click Application under CODESYS_Control_HBM_PMX_V3 and select "set active application" and log in again.

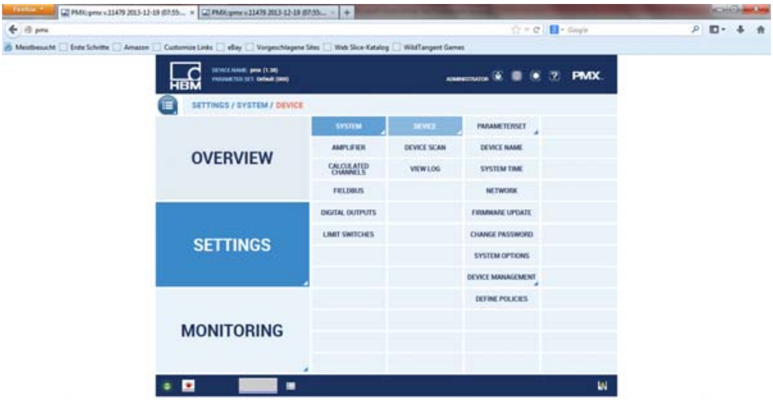


Configure Webbrowser

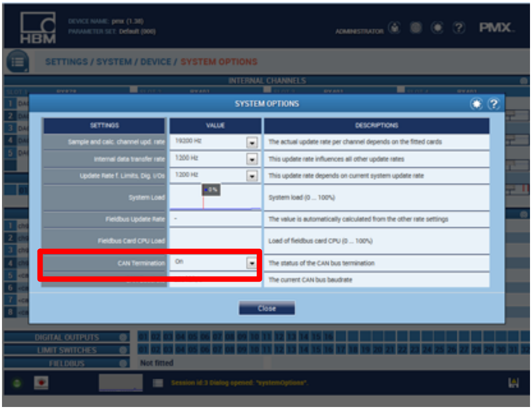
- Open PMX in the browser. Since two PMX devices are connected, the following overview is displayed. Copy the second PMX's IP address. Open each of the two PMX devices in a separate tab.



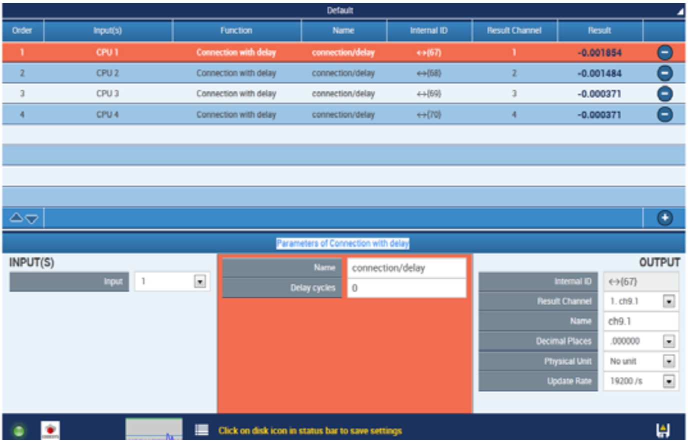
- Check the termination. For this purpose, use the Administrator to load the settings and system options for each device.



- Make sure that CAN Termination is ON for both devices.



- The browser enables the values to be displayed on the result channels; for this purpose, select "Connection with delay", select the respective CPU channel as input and specify a result channel as output. Allow for a sufficient number of decimal places.



Tip

Further information and help with creating CODESYS programs is available from the CODESYS online Help, on the Internet at https://www.codesys.com/# or in the CODESYS chat room http://forum.codesys.com/

Benefit from the knowledge and information available in the CODESYS Store. You'll find many examples of programs and solutions there, covering a wide range of task http://store.codesys.com/?___store=en&___from_store=en

18.14 PMX package

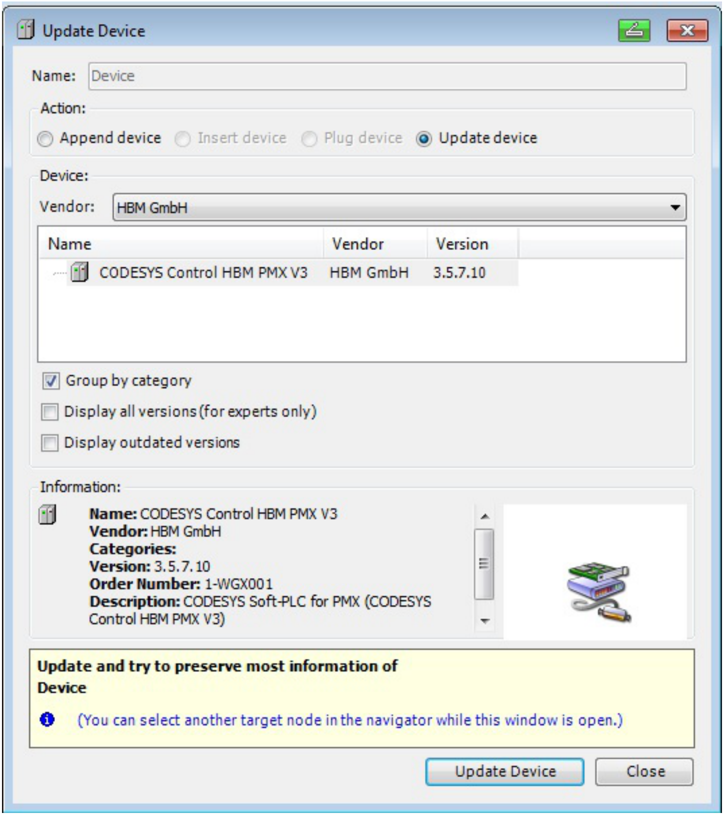
PMX package 0.94 includes some new functions for the PMX (see release notes). Proceed as follows when updating the PMX package from version 0.6 to 0.94:

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.

Notice

Package version 0.94 requires PMX firmware 3.0. If necessary, run a firmware update. The current PMX firmware can be found on the system CD, or at hbm.com: http://www.hbm.com/de/menu/support/software-firmware-downloads/industrie-messverstaerker/

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. Attention must be paid to the data volume, the memory speed and the storage location. The speed of data storage does not affect the measurement and data rates of the PMX.

DAQ (Data Acquisition):

The Ethernet interface of the PMX with a connected PC and DAQ software are suitable for storing large data volumes (DAQ). The HBM software catmanEASY/AP is available for this, as is the individual software that can be created using the PMX driver in dotNET, LabView or DIAdem. Using an Ethernet network, values from up to 20 PMX devices can be stored here synchronized to the measured values.

Monitoring:

For stand-alone monitoring applications, measured values can also be stored in the PMX device memory (1 GByte in size) or to a USB stick (max. size 32 GBytes). Only values that have been acquired by the particular PMX can be saved here. 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 on the system CD and in the collection of examples at http://www.hbm.com/en/menu/support/software-firmware-downloads/industrial-amplifiers/

The following table gives an overview of the possible signals and maximum storage rates.

Signals	catman	API/LabView/DIA-dem	CODESYS
Medium	Store on PC or server		Store internally on the PMX (1 GB) or USB stick (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 data storage examples can be found in the tips for PMX use on the PMX system CD or in the PMX download area at http://www.hbm.com/en/menu/products/industrial-amplifiers/pmx/

20 Data acquisition software (DAQ) catmanEASY/AP

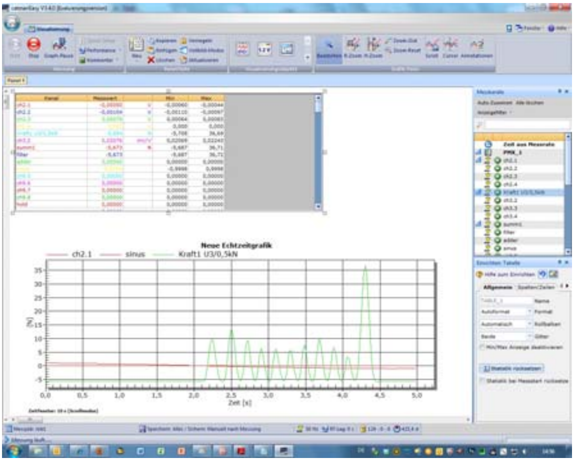
HBM's PC-based data acquisition software catmanEASY/AP can be used to professionally record data and for data analysis. You have many professional data acquisition functions (DAQ) available that are also helpful for starting up and for acquiring quality data:

- Visualization, storage and analysis of PMX measurement data, internal calculated channels, digital inputs/outputs and fieldbus data with up to 38.4 measured values per second and per channel.
- Simple PMX system and channel configuration (sensor database, TEDS editor, data rate, filters, etc.)
- Measured value trigger via PMX digital inputs/outputs, external PLC. (trigger prepost, cyclic storage, long-term measurement, etc.)
- Efficient data analysis (signal-to-signal, zoom, magnifier, ruler, min/max, FFT, cutting to size, removing outliers, etc.)
- Create automatic reports or display and export measurement data (MS Word, Excel, etc.)
- Create your own test and measurement routines with the script language catman Script.

Up to 20 PMX can be used simultaneously in a measurement with catman. Synchronization (< 1μs) occurs via sync connectors in the PMX (see section 9 Synchronization).

If the PMX is to be used in a measurement together with other measuring devices (e.g. MGCplus or QuantumX), the devices can be synchronized via the NTP protocol (1 ms to 100 ms). To allow the PMX to be used with other devices in catman, the parameter "ALLOWPMXHYBRID" in the registry file must first be set to the value 1.

This path for this file is: HKEY_CURRENT_USER\Software\VB and VBA Program\Settings\catmanEASY\SCAN and it can be changed by calling "regedit".



Important

Important: While a catman measurement is running, no signals may be added or removed in the PMX, as otherwise the catman measurement will be aborted.

Tip

A fully-operational demo version of catman is available as a free download at www.hbm.com

21 PMX command set (API)

With the command set (API), you can integrate the PMX into your own software applications. Customized solutions can then be implemented. These include Microsoft Windows and Linux-based software systems, such as e.g. VisualStudio, LabView, Delphi.

21.1 Requirements

- The PMX TCP-IP port is 55000
- All the commands are summarized in the command list (*Chapter 21.2)*.
- (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
- Virtual measuring subchannels use channel 9 (slot9), these are calculated channels, these are internally calculated channels
- Virtual digital subchannels (1 = dig in, 2 = dig out) use channel 10 (slot 10)
  Only the lowest 24 Bits are used due to binary representation of a float value. The 8 highest significant bits are always zero.
- Usually a positive response consists of a “0” followed by (y). A negative response normally is as “?” followed by (y)

- Character strings must be entered with a quotation mark“ at the beginning and at the end of the text. A quotation mark inside a character string is not permitted!

21.1.1 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 (if necessary, assign a suitable IP address to the PMX; DHCP is the factory setting default).

PMX command list example in a Telnet session under Microsoft Windows

The PMX must be connected to the PC (HOST) via an Ethernet cable or an Ethernet network

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:

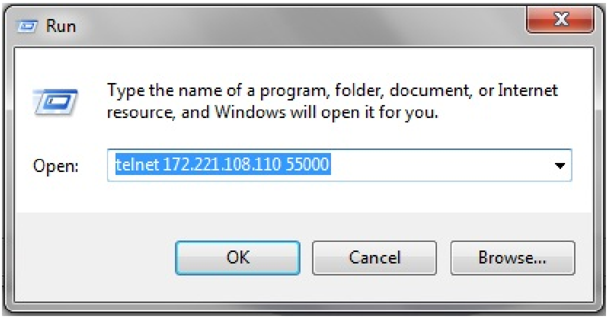
- under Windows XP/Vista/2000: **"Start" -> "Execute"**
- under Windows 7: **"Start" -> "All programs" -> "Accessories" -> "Run"**

(Under Windows 7, it may be necessary to activate a Telnet server. Click on **"Start" -> "Control panel" -> "Programs” -> "Turn Windows features on or off"**. mark **"Telnet client"** item and activate it. Then click on **"OK"** and wait a short while for the functions to be configured and applied.)

Start the Telnet session and establish the connection to the PMX:

"Telnet xxx.xxx.xxx.xxx 55000" (xxx.xxx.xxx.xxx = PMX IP address)

Example:



Get 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):* may be more than 16 characters

Example: IDN?(x)
HBM,PMX,1234-5678,1.12, 6415M*,0.20,myPMX (y)*

Company, 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 be changed	PX999 only for propietary commands
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 performs channel selection for the immediately following setup commands.

Syntax: PCS p1,.., pxx (x)

Parameters: 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). After that 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)

Parameters: p1: Any string "_____", max. 45 characters

Note: If the user wants to enter a channel name and comment which are both stored in the UCC string it is recommended to use a “;" to separate them.

Example: To store the channel name "Channel_Name_1" and the comment "Channel_Comment_1" in the amplifier, send the command:
UCC"Channel_Name_1;Channel_Comment_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 string, with a “ at the beginning and the end

Note: All comments of all selected subchannels of all selected channels (PCS und SPS) are returned **! All names (and comments) are separated by a “:" . Not a “," !!! .**

All the comments of all the selected subchannels of all the selected channels (PCS and SPS) are returned! *All the comments are separated by ":" and not by "," !!!*

EUN

Engineering Unit

Enter physical unit:

Syntax 1: EUNp1(x)

Parameters: p1: "UnitString"

Syntax 2: EUNp1(x)

Parameters: 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		

Code	Name	ASCII name
300	"m"	""
301	"µm"	"um"
302	"mm"	""
303	"cm"	""
304	"dm"	""
305	"km"	""
306	"inch"	"in"
307	"feet"	""
308	"yard"	""
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		

Code	Name	ASCII name
700	"K"	""
701	"°C"	"degC"
702	"°F"	"degF"
703	"°Rank"	"degRank"
704	"°R"	"degR"
// voltage/sensitivity		
1000	"V/V"	""
1001	"mV/V"	""
1002	"µV/V"	"uV/V"
// 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"

Code	Name	ASCII name
1403	"nF"	""
1404	"pF"	""
// charge r m kg s A K mol cd		
1500	"C"	""
1501	"nC"	""
1502	"pC"	""
// frequency		
1600	"Hz"	""
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"	""

Code	Name	ASCII name
// pressure		
2000	"Pa"	""
2001	"bar"	""
2002	"mbar"	""
2003	"kbar"	""
2004	"pas"	""
2005	"hPa"	""
2006	"kPa"	""
2007	"psi"	""
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"	""

Code	Name	ASCII name
	// speed	
2400	"m/s"	""
2401	"km/h"	""
2402	"mph"	""
2403	"fps"	""
2404	"m/h"	""
	// acceleration	
2500	"m/s^2"	"m/s2"
2501	"ga"	""
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	„%/^{o}C"	"%/degC"
3001	"%o/^{o}C"	"%o/degC"
3002	"ppm/^{o}C"	"ppm/degC"}

Code	Name	ASCII name
// numerical values		
3100	"Imp"	""
3101	"kImp"	""
// general physical units		
//	r m kg s A K mol cd	
5001	"%/decade"	""
5002	"dB"	""
5003	"l/l"	""
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 of the error status register or Systemstatus 1.

Syntax: ESR?(p1) (x)

Parameters: p1 is optional

Effect: The contents of the Standard Event Status Registers (ESR) are output in decimal equivalent. The Standard Event Status Register (ESR) is set when communication errors occur. Different fault causes set different bits, so that errors can be precisely identified.

Response: *p1 missing or p1=0*
q1(y)
q1: 8, 16 or 32 (or sum total)

Bit:	7	6	5	4	3	2	1	0

Bit 3 – Device-dependent error:
device-dependent error,
e. g. commend not permitted with this amplifier

Bit 4 – Execution error:
parameter error,
e. g. too many parameters

Bit 5 – Command error:
unknown command (syntax error),
e. g. *SER?

All other bits are unassigned.

Execution error: e. g.: command not valid for the selected channel (card).

ESR will be cleared after reading.

Response: *p1 : 1*
q1(*y*)
q1: sum of binary representation of single statusbits of Status 1, see table below

Status 1	binary Value	Remarks
No error	0	
FACTORYSETTINGS_ERROR	1	factory calibration for PMX-housing corrupted (not meascards !)
SYNC_MASTER	2	bit set:sync master bit cleared: sync slave
SYNCMESSAGE_ERR	4	missing or invalid sync-messages
SYNC_UNLOCKEDSLAVE_ERR	8	cannot lock, regulator is not able to synchronize to master
ALIVE	16	toggles with approx. 1Hz
POWEROVERLOAD	32	power supply overload somewhere
CAT_BUF_OVERRUN	64	“Catman” interface buffer overrun (TSV..) ==> meas. value stream “broken”
SYSTEM_NOT_READY	128	e.g. change of parameterset 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) deselects 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. 153,640 Hz. Time channels 17, 18 and 19 do not appear in commands PCS?/SPS?. Times do not have subchannel realizations (SMS command).
The binary ("measurement") data have a length of 8 bytes, where the highest valued two bytes are zero.
Time data are always placed at the end of one measval line.
There is an implicit connection between the Sample rate group and the three time channels:
Channel 17: Sample rate group 0
Channel 18: Sample rate group 1
Channel 19: Sample rate group 2

Note: The response depends on the SRB command. Have a look at STF and NTP command too. The multi IO-card PX878 is not selectable here. It does not create measurement 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)

Parameters: p1,.., pxx 1,.., 128 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. It is possible to select different signal combinations for the various channels. In particular, it is possible to record more than one signal for each sub channel.

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 increase the amount of data to be calculated and transmitted dramatically. Not everything will be possible.
For the reason PMX has internally a multi client software architecture and Catman is “only” one of these clients, the available signals (except gross) must be created at top level. Otherwise these signals are not available.
Response depends on SRB command.

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

Sample rate Group

Measurement signal selection for channels to be recorded

This signal assigns the Sample rate group to a selected channel or subchannel (PCS/SPS). Up to 3 synchronous Sample 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)

Parameters: p1: 0..2 Sample rate group
p2: 0..2 Sample rate group (optional)
p3: 0..2 Sample rate group (optional)

Note: The response depends on the SRB command.

MRG?

Sample rate Group Query

Output measurement signal selection for channels to be recorded

Syntax:	MRG? (x)
Response:	q1(y)
	q1: Sample 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 string)

Syntax:	NTP p1 (x)
Parameter:	p1: NTP-serves's IP-address or name: 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 string)

Syntax:	NTP? (x)
Parameter:	p1: is optional p1 missing or is 0 : call NTP server p1 =1; accuracy information as a character string

Response: for p1= 0 (or missing p1):
q1(,q2,q3..)(y)
q1 …: used NTP serevrs as strings, separated by a “,”

Note: This query replies the currently used NTP-server(s). The server(s) can differ from the one which was defined by the NTP-command before. This can happen if a DHCP-server defines a different NTP-server.

ICR

Internal Channel Recordingrate

Internal channel Sample rate

This command is used to set (only) one Sample rate per group. A second Sample rate in one group is not supported.

Syntax: ICR p1, p2(x)

Parameters: p1:Sample rate 1, see rate list below
p2: 0, 1, 2 ; Sample rate group

If parameter p2 is omitted, the command affects Sample rate group 0.

Status	Value	Comments
1 Hz	6300	
2	6301	
5	6302	
10	6303	
20	6326	
25	6304	
50	6305	
75	6307	
100	6308	
150	6309	

Status	Value	Comments
200	6310	
300	6311	
600	6313	
1200	6315	
2400	6317	Default
4800	6319	
9600	6320	
19200	6345	
38400	6346	

Note: The response depends on the SRB command.

ICR?

Internal Channel Recordingrate Query

Output the internal channel Sample rate

Syntax: ICR? p1(x)

Parameters: p1: Sample rate group 0, 1, 2

Response: q1 (y) Sample rate

Note: If parameter p1 is omitted, the command affects Sample rate group 0.

TSV

Transient Setup Values

This command defines and starts data acquisition.

Syntax: TSVp1 (x)

Parameters: p1: 0, 1,..,N Number of value lines to be measured in a single measurement:

1…N → Max. FIFO size 15 MB per Sample rate group
0 means infinite. → Default FIFO size 5 MB per Sample rate group

-1 means infinite in FIFO size of 1 line.

This allows the user to 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 "Message overflow" status bit, see TSV? query

TSV?

Transient Setup Values Query

This command defines and starts data acquisition.

Syntax: TSV? p1(x)

Parameters: p1: Sample 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 (wait until end)
3: Measurement ended (default)

q3: Status bits
Bit 0, (value = 1): FIFO buffer overflow, is cleared by the next TSV command
Bit 1, (Value=2): not yet in use
Bit 2, (value=4): not yet in use
Bit 3, (value=8): not yet in use
Bit 4, (value=16): not yet in use

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

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)

Parameters: p1: -N,.,N , Offset for moving the read pointer: -(max FIFO lines -1)…max. FIFO lines -1 max. FIFO lines after TSV command

P2: 0,..,2 Sample rate group 3 asynchronous FIFO

If the Sample rate group (p2) is not specified, Sample rate group 0 is affected.

Note: The response depends on the SRB command.

OMP?

Output Measuring Pointer Query

Syntax: OMP? p1(x)

Parameters: p1: Sample 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 Sample rate group is not specified, Sample 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: Sample rate group 0,..,2

If parameter p2 is omitted, the setting affects all Sample rate groups.

Note: The response depends on the SRB command.

MBF?

Measuring Buffer Format Query

Output the output format

Syntax: MBF? p1(x)

Parameters: p1: Sample rate group 0,..,2

Response: q1(y)
q1: Output format

If parameter p1 is omitted, you are given the output format of Sample 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 output 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)

Parameters: 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 reading the latest values (read pointer is unchanged) , not supported
6409 from the current read pointer; move read pointer forward by p1 lines

p3: Sample rate group 0,..,2

If the Sample rate group is not specified, Sample rate group 0 is affected.

RMV?

Read Current Measurement Value

Output measurement data.

Syntax: RMV? p1 (x)

Parameters: p1 Signal

p1	Signal
214	Gross
204	Min
205	Max
217	Peaktopeak

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 signals Min, Max, Peak/Peak have to be “defined/ created" at top level before (see command “MSS"), if they should 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 (slot9 / channel 9) do not support the subsignals min, max and peak-to-peak.

Digital values (slot10) and analog output values (PX878) don’t’ have min/max/peak-peak values itself. Analog output values (PX878) are updated approx.. every 0,5 seconds (2Hz). So if this command will be called more often the same value will be returned. In case that the signal source of the analog output has an invalid status 2e20 is returned (independently of the “Invalid Signal Value” entered in the Web-GUI. The voltage levels for the PX878 are the calculated ones. They are NOT (re-)measured at its 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, according to the measurement card.

p1=141 / 142	Cut-off frequency (Hz)			
p2	**Frequency in Hz**	**PX401**	**PX455**	**PX460**
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

p1=141 / 142	Cut-off frequency (Hz)			
p2	**Frequency in Hz**	**PX401**	**PX455**	**PX460**
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 “Sample and calc. channel update rate" (Browser User Interface: Settings -> System -> Device -> System Options) is set to 38,4k.
2) This value means that the digital filter is working with “neutral" coefficients and only the analog anti alias filter is active. This physical cut off frequency may be card dependant.

Virtual subchannels (slot9 / channel 9), digital subchannels (slot20 / channel10) and the PX878 multi IO-card do not support filters. It is possible to define parameters p1, p2 (and p3), 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 Limit 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 (channel 9) 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)

Parameters: 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 defines also the scaling of the PX878. That means the physical (p2, from signal source) and electrical (p3, output in Volt) values 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)

Parameters: 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, by implication. The ACL setting does NOT change! Only supported for PX455! Other (meas-)cards return “OK”(0).

Syntax: CAL(x)

Parameters: none

Note: In all CF bridge amplifiers, this command triggers a calibration. The 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 (meas-)cards return “OK”(0).

Syntax: CAL?(x)

Parameters: none

Response: *q1(y)*

0	Automatic calibration is not executed
1	Automatic calibration is executed

e.g. two boards 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 connected or the measurement signal overflows for a few seconds. Only supported for PX455! Other (meas-)cards return “OK”(0).

Syntax: ACLp1(x)

Parameters:

p1	Autocalibration
0	Off
1	On

Note: If on (enabled) cal is enabled for bridge or bridge-like sensors if sensor is plugged (poti / lvdt).
A ACL 0 command does not abort a running cal sequence. Only a further start will be suppressed.
Response depends on SRB command.

ACL?

Enable / Disable Autocal Query

Output of autocalibration status

All selected channels (PCS/SPS). Only supported for PX455! Other (meas-)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 an amplifier input signal. Only supported for PX455. Command is ignored for other (meas-)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

Amplifier input signal request. Only supported for PX455. Other (meas-)cards return 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 the 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: On clearing, peak value stores (Min or Max) are set to the current measured value. Peak-Peak is set to 0.0 .
Peak-Peak has its own Min/Max stores!
The peak value signals have to be created at top level parameterization before. Otherwise they are not available.
Virtual subchannels (slot9 / channel 9), digital subchannels (slot10 / channel 10) and the PX878 multi IO-card don’t 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). Suspend/enable peak value store updating.

Syntax: HPVp1,p2(x)

Parameters: p1: Peak-value memory 1 (Max), 2 (Min) or 3 (Peak-to-Peak)

p2=1: suspend update

p2=0: enable update (default)

Every time you switch on, the status is set to "enable apdating".

Virtual subchannels (slot9 / channel 9), digital subchannels (slot10 / channel 10) and the PX878 multi IO-card don’t 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)

Parameters: 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
:: indicates that this subchannel (between the two :) has no max, min or peak-to-peak values

Syntax 2: HPV??(x)

(y): Available peak-value memories: e. g. 1,2,3:1,2::1 :: indicates 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 (slot9 / channel 9), digital subchannels (slot10 / channel 10) and the PX878 multi IO-card don’t support peak values.

SAD

Sensor Adaption

Select transducer adaptation for all selected channels (PCS/SPS)

Syntax: SAD p1,p2,p3(x)

Parameters:

p1	Excitation voltage (or current), *see Tab. 21.3, see Tab. 21.6* for PX460
p2	Transducer type, *see Tab. 21.4* or Tab. 21.7 for PX460
p3	Sensitivity (optional), *see Tab. 21.5* (not usable for PX460)

Status	Value	Comments	Command
No excitation	10	PX401	
1V	11	Not supported	
1.25V	12	Not supported	
2.5V	13	PX455	
5V	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)

Status	Value	Comments
Inductive half bridge	357	PX455 (= HB 100mV/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	
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)
75mV DC	425	
10V DC	426	PX401
20mA DC	427	PX401
60V 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 ended)
24	Inductive (only for frequency measurement)

Tab. 21.6 Input type for PX460

p2:

Status	Transducer Type
520	Frequency (direct or inductive)
525	Impulse counter (only direct)
580	SSI (only direct)
527	PWM (only direct)

Tab. 21.7 Transducer types for PX460

Virtual subchannels (channel 9) use values p1=10, p2=575. Setting other values will be ignored (answer 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 Outut type for PX878 (p2)

Virtual subchannels (slot9 / channel 9) and digital subchannels (slot10 / channel10) use values p1=10, p2=575. Setting other values will be ignored (answer OK).

PX878 uses values p1=10, p2=290. Setting other values will be ignored and returns “?” as 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. 21.5*

	e. g. PX401: 10,426,-1:10,427,-1:….
Syntax 2:	SAD??(x)
Parameters:	none
Response:	*q1,..,qn(y):* possible excitation voltage or similar as per table 1 to 4 (SADCommand) 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 the 4th subchannel can be configured (SPS2 / SPS4). The 1st and the 3rd subchannel are fixed to frequency measurement with direct (digital) input.

The 1st and the 2nd subchannel use the same settings for glitch filter, digital input type and termination. The 3rd and the 4th subchannel are bound together in the same way.

Syntax: SAF p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11(x)

p2…p11 are optional

Parameter:

p1	glitch Filter	0: Off / 0.082, 1, 10, 100: On	Eliminates pulse widths < (p1) µs
p2	digital input type	0: differential 1: single ended	differential or single ended digital input (negative input set to “middle” voltage), default is 0
p3	termination	0: no	electrical termination for differential input mode
		termination 1: termination resistors active	default is 0
p4	sensortype is counter	525	Defines meaning and number of subsequent parameters

p5	F1+F2	0: Off / 1 (F2=90deg), 2 (F2=dir): On	F1+F2 signal is evaluated
p6	resolution quadrupling	0: Off / 1: On	Evaluates only one or all edges
p7	physical zero index input	0: Off / 1 On	Enables hardware (zero) input. If ON (every) zero-index-pulse the counter-result is set to p10
p8	factor	0: Off (disabled) / 1,2,3,4: On	Enables automatic resetting of counter after p8 rotations (p8 * p9 pulses)
p9	pulses per revolution	0: Off / 1...16000: On	If On, the counter is set to zero after reaching a count-result of p9. A physical zero-pulse (p7) can reset the counter result too. Usually these possibilities are used in combination.
p10	offset in pulses	0...16000	Reset-value of counter, usually not bigger than p9
p11	invert direction	0: standard 1: inverted	swaps counting direction
p12	use interpolation	0: OFF 1: ON	Similar to a filter, useful for slow pulses

Note: This command automatically resets the current counter-value to zero (NOT p10 !).
To reset the counter without changing the current settings (parameterization has been done before) only p1 is necessary.

Explanations:

Glitch Filter, p1

Input signals with pulse widths < x µs are not evaluated. Default Value is 1 (1µs).

F2 evaluation, p5

Activates detection and evaluation of direction of rotation using the second Hardware input. Default Value is 0 (off).

Resolution quadrupling, p6

All edges adjacent to F1 and F2 are evaluated. If F2 is not connected or F2-evaluation is turned off (p6=0), there is a resolution doubling. Default Value is 0 (off).

Zero index input, p7

Hardware input for the zero index signal. Relevant for incremental transducers. In count mode an active (electrical high level) zero index signal sets the counter value to p10. Default Value is 0.

Factor, p8

Factor for automatic resetting the counter value.to p10 after p8*p9 pulses (p8 rotations). Default Value is 0.

Pulses per revolution, p9

Used for rotary encoders. Pulse count for a single turn. If resolution quadrupling is activated 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” may also align (compensate) a quarter-pulse- width difference between these two competing ways of resetting the counter value.

Angle Offset in pulses, p10

Value which the counter gets when zero index is active. Default Value is 0.

Invertion of count-direction, p11

With this parameter it is possible to change the direction of counting. Default Value is 0.

Invertion of interpolation, p12

Works similar as a filter. Is useful for signals with slow pulses to smooth the measurement value.
Default Value is 0.

Configuration Examples

	physical zero index input, p7	Factor, p8	Pulses per revolution, p9	
Linear configuration 1	0	0	0	Counting up (and down if p5<>0), no reset to zero or p10
Linear configuration 2	1	0	0	Counting up (and down if p5<>0), reset to zero or p10 if physical zero index is active
Rotary configuration 1	0	0	1...16000	Counting always up (and down if p5<>0), no reset to zero or p10, no wrap around
Rotary configuration 2	0	1...4	1...16000 / p8	Reset counter each p8 rotations, no “physical zero index” used but automatic wrap around after p8 rotations
Rotary configuration 3	1	1...4	1...16000 / p8 *	Resets counter each p8 rotations, uses “physical zero index” and “pulses per revolution” (p9). Counter reset is synchronized with F1/F2 pulses.
Rotary 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 use Rotary configuration 3

*subject to be changed ....

Parameters for SSI transducers

Syntax: SAF p1,p2,p3,p4,p5,p6,p7(x)

p2…p7 are optional

Parameter:

p1	glitch Filter	0: Off / 0.082, 1, 10, 100: On	Eliminates pulse widths < (p1) µs
p2	digital input type	0: differential 1: single ended	differential or single ended digital input (negative input set to “middle” voltage), default is 0
p3	termination	0: no termination 1: termination resistors active	electrical termination for differential input mode default is 0
p4	sensortype is SSI	580	Defines meaning and number of subsequent parameters
p5	encoding	0, 1	0: Gray-code (default), 1: binary
p6	bit-length	6...32	transducer’s bit-length (resolution), usually 12,13,24,25(default)
p7	baudrate	1...5	baudrate for transducer’s clock: 1:10kBit, 2:100kBit (default), 3:200kBit, 4:500kBit, 5:1000kBit

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 depend on currently activated sensor

q1	glitch Filter	0: Off / 0.082, 1, 10, 100: On	all sensortypes
q2	digital input type	0: differential 1: single ended	all sensortypes
q3	termination	0: no termination 1: termination resistors active	all sensortypes
q4	sensortype		520 Frequency (direct or inductive) 525 Impulse counter 580 SSI 527 PWM
q5	F1+F2 or encoding		525 Impulse counter or 580 SSI
q6	resolution quadrupling or bit-length		525 Impulse counter or 580 SSI
q7	physical zero index input or baudrate		525 Impulse counter or 580 SSI
q8	factor		525 Impulse counter
q9	pulses per revolution		525 Impulse counter
q10	offset in pulses		525 Impulse counter
q11	invert direction		525 Impulse counter
q12	Use interpolationj		525 Impulse counter

SCL

Shunt Calibration output (only PX460)

Only the shunt-output of the 2nd and the 4th subchannel can be configured (SPS2 / SPS4, the PX460 has only 2 shunt-outputs)

Set shunt output on / off

Syntax: SCL p1(x)

Parameter:

p1	set shunt output
0	Off
1	On

SCL?

Shunt Calibration output Query (only PX460)

Syntax: SCL?(x)

Parameter: none

Response: *q1(y):*current status of shunt output setting

TDD

Transmit Device Data

Activate different amplifier settings (parameter sets)

Syntax: TDD p1,p2(x)

p2 is optional

Parameter:

p1	Parameter set
-2	Saves all current parameters and parametersets of the device and defines currently active parameterset as boot-parameterset. This command can take a plenty of time (e.g. 10…60s).
-1	Load factory default into currently active parameter set and activate it. p2 will not be used. Command returns immediately. Setting one parameterset to factory default (p1 = -1) sets all associated sub- parametersets to factory default. Be aware that these sub-parametersets could be used in other parametersets too !!
0,1,2,…	Load parameterset p1 and activate it. If changing of the parameterset has an error the response might be “OK” too. Please use TDD? to verify the parameterset which is currently in use.

p2	response behavior / timeout
missing or 0	don’t wait, no timeout detection, command responds immediately
>0.1,…	Timeout in seconds. Waits until parameter-set switching or saving (p1=-2) has finished or timeout occurred. If parameterset is set to default values (factory setup, p1 = -1) the timeout option p2 is not available!

Note: PMX’s parameterset-system consists of sub-parameter-sets which
are linked together to one main-parameterset which can be activated here (p1 >= 0).

TDD?

Transmit Device Data Query

Query, where amplifier set-up comes from

Syntax: TDD? p1(x)

p1 is optional

Parameter:

p1	command
missing or 0	get active parameter set
1	get SYSTEM_NOT_READY statusbit, → if set parameter switching is in progress

Response: *q1(y):* in the case of p1

p1	parameter from query
missing or 0	currently active parameter set
1	0:System ready, parameter switching finished , 1:System busy

CDT

Calibration Dead Load Target
Zero point 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 (slot9 / channel 9), digital subchannels (slot10 / channel 10) and the PX878 multi IO-card don’t support “dead load values”. Command is ignored with OK-answer.
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):* The current zero point value in the displayed units e. g. 0.01,0,10.5,10.502

Virtual subchannels (slot9 / channel 9), digital subchannels (slot10 / channel 10) and the PX878 multi IO-card don’t support “dead load values”. q1=0.

ATB

Application To Bus

Writes a 64-bit integer value that can be read by the fieldbus master

Syntax: ATBp1 (x)

Parameters: 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

Outputs the current 64-bit integer value as a hexadecimal number, that was previously written with the command ATB

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

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

STF Set Time Format

Defines the content and format of the time channels (MCS 17,18,19)

Syntax: STFp1 (x)

Parameters: p1 = 0:default, ticks (increasing counter) as a 64 bit integer value

p1 = 1: system time as two 32 bit integers, nanoseconds (1st 4 bytes) and seconds (2nd 4 bytes)
p1 = 2:system time as two 32 bit integers, microseconds and seconds
p1 = 3:system time as two 32 bit integers, 2^-32 seconds and seconds

The sum of the seconds and its fractions is the time passed since 01.01.1970.

The system time can be derived from the NTP-time. The accuracy is not 100% predictable.

Note: Response depends on SRB command.

STF?

STF? Set Time Format Query

Reads the time format currently in use

Syntax : STF?(x)

Parameters: none

Response: *q1(y):* current setting of the timeformat

BLK **Blink**

Enables or disables LED signalling of the whole device or signalling of selectes (sub-)-channels (PCS / SPS)

Syntax : BLK p1,p2,p3(x)

Parameter 1	Selection p1 = 0 : whole device p1 =1 : subchannel
Parameter 2	Blink mode see tables below for both selections of p1
Parameter 3	P1 = 0: time of LED signalling 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 are damaged or if TEDS is not available, we have the response q1 = "?". In this case, no TEDS data are transmitted to the amplifier (length = 0). If more than one TEDS transducer identification is read out (PCS/SPS), the error response is also output if only one TEDS transducer identification has a read error. TEDS data are 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 output, 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.

p1	p2	P3	Effect
1	Data length	Data (ASCII hex characte r string)	Transmits and writes data to the transducer memory. p2 = data length: number of bytes. p3: Data in ASCII-Hex format. e. g. "AB75e2…". If p2 (data length) is equal to 0, and p3 is an empty string "", (but has to be output), the data written to the transducer is taken over from the amplifier memory. Of course, this is 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.
100	-	-	Reads and interprets the TEDS data. If the data are damaged, or settings for the amplifier are not possible, we have the response 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 an eventually set TEDS-measval errorbit. This errorbit could be set if a TED100 command finds a valid TEDS content but device configuration failed. This could result in an invalid or only partly done configuration and therefore in potentially invalid measvalues. Virtual subchannels (channel 9) and the PX878 don’t support TEDS (not physically present). Command will be ignored and returns OK.

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 are separated by a **";"** Virtual subchannels (channel 9) do not support TEDS transducer identification (not physically present).
1	Reads the TEDS data **from** the memory of the **amplifier.** 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 (single-wire). e. g. 512 bytes. If more than one subchannel is selected (PCS/SPS), the data are separated by a ";". The minimum number of bytes should be 31 (1 checksum byte is deducted from the 32-byte page).
100	Gets warning / error-status of the TEDS-template handling (caused by TED100). **Not MGC-like** q1 = “0”: OK q1 = “?”: error or no information available
101	Gets TEDS-measval errorbit q1 = “0”: OK q1 = “?": errorbit set
102	Gets status of TEDS-setting. q1 = “0": Parameters defined by TEDS were changed afterwards q1 = “1": all parameters defined by TEDS are set in the amplifier, **Not yet supported**

TED?100

Response: q1,q2,q3

q1:0: No error
otherwise template-IDs with errors

q2: Bit position error in the template

q3: Error type:17000 template and amplifier are not compatible

17002	Value above limit value
17003	Value below limit value
17004	Value outside the limit values

TED?102

Response q1: Checks the status of the TEDS setting, does not read out 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)

Parameter:

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 hex string

Note: Virtual subchannels (channel 9) and the PX878 don’t support TEDS. The query returns 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 off response output
1	Switch on response output

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 the data being output in this type of command.

b.) The setting commands (e. g. SRB) generate feedback data (0 or ?). You can specify whether this data should be output in this type of command, by switching the option on or off.

Response:

Response	Significance
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

Output of the existing errors and warnings in list form for each selected sub channel. Each error/warning of a sub channel is separated by a comma (a sub channel can have more than one error/warning). Each sub channel is separated by a colon (**:**) for the error status values, see 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

Reads the time format currently in use

Syntax :	IDS?p1(x)
Parameters:	p1: numerical value of text-handle
Response:	*q1:* identification string in English for p1
Example:	IDS?15030(x) “Hardware underflow”(y)

Supported text-handles:
15001, 15020, 15023, 15030, 15031, 20031

LSS?

Limit Switch Status Query

Query limit value status

Syntax: LSS?(x)

Parameter: 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

Input limit value level

Syntax: LVL p1,p2(x)

Parameter: p1: numbers of limit switches (1...32)

p2: level in displayed units (floating) Response depends on SRB command.

LVL? Limit Value Level Query

Output limit value level

Syntax 1:	LVL?p1(x)
Parameter:	p1: number of limit switch (1...32)
Response:	*q1,q2(y)*
	q1: number of limit switch (1...32)
	q2: level in displayed units (floating)

Syntax 2:	LVL??(x)
Parameter:	none
Response:	*q1,q2(y):* available limit switches (range): 1, 32

Syntax 3:	LVL?,?(x)
Parameter:	none
Response:	*q1,q2(y):* possible input range of level (floating)

LVS

Limit Value Switch

Parameterizes limit switches.

Input value is used from the first signal which has been defined with PCS and SPS command.

Syntax: LVS p1,p2,p3,p4,p5,p6,p7,p8(x)

Parameter: p1: number of limit-switch (1...32)

p2: OPERATION (ON =1 or OFF=0)

P3	Direction
130	Above limit
131	Below limit
132	In band
133	Out of band

p4: level in displayed units (lower level in band mode) (floating)

p5: hysteresis (or width of band in band mode) in displayed units (floating)

p6: Reset behaviour (can be used for hysteresis management) : binary mask which is ANDed with all digital inputs. If result is >0 (true) this limit-switch is cleared. Value range: 0,1,2,4,8,16,....32768. Reset behaviour can be inverted with p7. Default is 0. Parameter is optional.

p7: 0 or 1. invert reset behaviour

0: Reset behaviour works like defined with p6

1: Reset behaviour is inverted

Default is 0. Parameter is optional.

p8: 0 or 1.

0: Only if measurement-value-status is OK limit-switch is evaluated, freezes limit-switch status if measurement-value-status has any error(s)

1: Ignore measurement-value-status

Default is 0. Parameter is optional.

LVS?

Limit Value Switch Query

Output parameter assignment of limit switches .

Syntax1: LVS?p1(x)

Parameter: p1: number of limit switch (1...32)

Response: *q1...q10(y)*

q1: number of limit switch (1...32) q2: enabled status (ON=1, OFF=0) q3: input channel (slot)

q4: input sub channel (signal)

q5: operating direction (-1(off),130,131,132,133, see LVS command)

q6: level or lower value of band in displayed units (floating) q7: hysteresis or width of band in displayed units (floating) q8: reset behavior (binary mask, see LVS command)

q9: invert reset behavior (0, 1, see LVS command)

q10: ignore measval-status (0, 1, see LVS command)

Syntax 2: LVS??(x)

Parameter: none

Response: *q1,q2(y):* available limit switches (1..32)

SOP

Setup Output

Parameterizes digital outputs.

Measurement-value-status is used from the first signal which has been defined with

PCS and SPS command.

Syntax: SOP p1,p2,p3,p4,...,p17(x) p4 ... p17 are optional. Default values are 0;

Parameter: p1: number of digital output (1...16)

p2: enable for limit-switches (ON=1, OFF=0). If enabled the status of the 32 limit-switches in combination with the mask for the limit- switches (p3) will be used to define the state of a digital output.

p3: binary mask for limit-switches which is ANDed with all limit- switches.

Bit 0 of this mask is used for Limit-switch 1, Bit 1 is used for limit- switch 2 and so on. If result is >0 (true) this digital output will be set.

Value range: 0,1,2,4,8,16,....32768,65536,....2^30,2^31. A mask with more than one bit set is possible too. This could be used to create a “window-behaviour” of a digital output.

p4: enable for measurement-value-status (ON=1, OFF=0). If enabled the status of a measurement-value-status will be used to define the state of a digital output. If this measval-status shows any error the digital output is set to 1 / switched on. (PCS / SPS)

p5: enable for fieldbus-bits (ON=1, OFF=0). If enabled the status of the 32 bit word which can be written by a field-bus master in combination with the mask for the fieldbus-bits (p6) will be used to define the state of a digital output.

p6: binary mask for the fieldbus-bits which is ANDed with the 32 fieldbus-bits. If result is >0 (true) this digital output will be set. A mask with more than one bit set is possible too. Value range: 0,1,2,3...(2^32-1)

p7: enable for digital inputs (ON=1, OFF=0). If enabled the status of the 16 digital inputs with the mask for the digital inputs (p8) will be used to define the state of a digital output.

p8: binary mask for the digital inputs which is ANDed with the 16 digital inputs. If result is >0 (true) this digital output will be set. A mask with more than one bit set is possible too. Value range: 0,1,2,3...(2^16-1)

p9: enable for parameter-set-number (ON=1, OFF=0). If enabled the parameter-set-number with the mask for the parameter-set-number (p10) will be used to define the state of a digital output.

p10: binary mask for the parameter-set-number which is ANDed with the currently active parameter-set-number. If result is >0 (true) this digital output will be set. A mask with more than one bit set is possible too. Value range: 0,1,2,3...(2^32-1)

p11: enable for calculated-channel-bits (ON=1, OFF=0). If enabled the status of the 32 bit word which can be defined by one or more calculated-channel(s) in combination with the mask for the calculated- channels (p12) will be used to define the state of a digital output.

p12: binary mask for the calculated-channels which is ANDed with the 32 calculated-channel -bits. If result is >0 (true) this digital output will be set. A mask with more than one bit set is possible too. Value range: 0,1,2,3...(2^32-1) p13: enable for CodeSys-bits (ON=1, OFF=0). If enabled the status of the 32 bit word which can be defined by CodeSys Application in combination with the mask for the CodeSys-bits (p14) will be used to define the state of a digital output. If CodeSys is not available, value is 0.

p14: binary mask for the CodeSys-bits which is ANDed with the 32 CodeSys-application bits . If result is >0 (true) this digital output will be set. A mask with more than one bit set is possible too. Value range: 0,1,2,3...(2^32-1).
If CodeSys is not available, value is 0.

p15: enable for systemstatus-bits (ON=1, OFF=0). If enabled the status of the 32 bit systemstatus in combination with the mask for the systemstatus (p16) will be used to define the state of a digital output.

p16: binary mask for the systemstatus-bits which is ANDed with the 32 systemstatus-bits. If result is >0 (true) this digital output will be set. A mask with more than one bit set is possible too. Value range: 0,1,2,3...(2^32-1)

p17: invert digital output (0: as described before, 1: inverted)

SOP?

Setup Output Query

Output parameter assignment of a certain digital output

Syntax1: SOP?p1(x)

Parameter: p1: number of digital output (1...16)

Response: *q1...q19(y)*

q1: number of digital output (1...16)

q2: enable for limit-switches (ON=1, OFF=0). If enabled the status of the 32 limit-switches in combination with the mask for the limit- switches (q3) will be used to define the state of a digital output.

q3: binary mask for limit-switches which is ANDed with all limit- switches.
Bit 0 of this mask is used for Limit-switch 1, Bit 1 is used for limit- switch 2 and so on. If result is >0 (true) this digital output will be set. Value range: 0,1,2,4,8,16,....32768,65536,....2^30,2^31. A mask with more than one bit set is possible too. This could be used to create a “window-behaviour” of a digital output.

q4: enable for measurement-value-status (ON=1, OFF=0). If enabled the status of a measurement-value-status will be used to define the state of a digital output. If this measval-status shows any error the digital output is set to 1 / switched on.

q5: input channel (slot)

q6: input sub channel (signal)

q7: enable for fieldbus-bits (ON=1, OFF=0). If enabled the status of the 32 bit word which can be written by a fieldbus master in combination with the mask for the fieldbus-bits (q8) will be used to define the state of a digital output.

q8: binary mask for the fieldbus-bits which is ANDed with the 32 fieldbus-bits. If result is >0 (true) this digital output will be set. A mask with more than one bit set is possible too. Value range: 0,1,2,3...(2^32-1)

q9: enable for digital inputs (ON=1, OFF=0). If enabled the status of the 16 digital inputs with the mask for the digital inputs (q10) will be used to define the state of a digital output.

q10: binary mask for the digital inputs which is ANDed with the 16 digital inputs. If result is >0 (true) this digital output will be set. A mask with more than one bit set is possible too. Value range: 0,1,2,3...(2^16-1)

q11: enable for parameter-set-number (ON=1, OFF=0). If enabled the parameter-set-number with the mask for the parameter-set-number (q12) will be used to define the state of a digital output.

q12: binary mask for the parameter-set-number which is ANDed with the currently active parameter-set-number. If result is >0 (true) this digital output will be set. A mask with more than one bit set is possible too. Value range: 0,1,2,3...(2^32-1)

q13: enable for calculated-channel-bits (ON=1, OFF=0). If enabled the status of the 32 bit word which can be defined by one or more calculated-channel(s) in combination with the mask for the calculated- channels (q14) will be used to define the state of a digital output.

q14: binary mask for the calculated-channels which is ANDed with the 32 calculated-channel -bits. If result is >0 (true) this digital output will be set. A mask with more than one bit set is possible too. Value range: 0,1,2,3...(2^32-1)

p15: enable for CodeSys-bits (ON=1, OFF=0). If enabled the status of the 32 bit word which can be defined by CodeSys Application in combination with the mask for the

	CodeSys-bits (p16) will be used to define the state of a digital output. If CodeSys is not available, value is 0.
	p16: binary mask for the CodeSys-bits which is ANDed with the 32 CodeSys-application bits . If result is >0 (true) this digital output will be set. A mask with more than one bit set is possible too. Value range: 0,1,2,3...(2^32-1). If CodeSys is not available, value is 0.
	p17: enable for systemstatus-bits (ON=1, OFF=0). If enabled the status of the 32 bit systemstatus in combination with the mask for the systemstatus (p18) will be used to define the state of a digital output.
	p18: binary mask for the systemstatus-bits which is ANDed with the 32 systemstatus-bits. If result is >0 (true) this digital output will be set. A mask with more than one bit set is possible too. Value range: 0,1,2,3...(2^32-1)
	q19: invert digital output (0: as described before, 1: inverted)
Syntax 2:	SOP??(x)
Parameter:	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 lower 8 bits represent the 8 inputs of the first PX878. The higher 8 bits represent the inputs of the second PX878.

The bits 16 … 31 are always virtual !

Example: RIP?(x)
1025(y)

Input 3 (bit 10 (8+2)) of the second PX878 and input 1 (bit 0) of the first PX878 are set (the input count starts with 1).

ROP

Set Digital Outputs

Set the digital outputs

Syntax: ROP p1, p2(x)

Parameters: p1: binary realization of all outputs, 0...65535
p2: optional, binary realization 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 higher 8 bits represent the outputs of the second PX878.

The outputs are available even if no PX878 is plugged in. In that case they are virtual. They can be set, read back, but are not electrically present.
p2 defines the selected bits which corresponding output is set or cleared with p1. The outputs which corresponding bits are 0 in p2 are not affected.

The bits 16 … 31 are always virtual !

Example: ROP2, 32770(x)
0(y)

Output 8 (bit 15) of the second PX878 is cleared and output 2 (bit 1) of the first PX878 is set (the output count starts with 1).
These two bits are the only ones 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. So you must be careful what you do!

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 lower 8 bits represent the 8 outputs of the first PX878. The higher 8 bits represent the outputs of the second PX878.
The outputs are available even if no PX878 is plugged in. In that case they are virtual. They can be set, read back, but are not electrically present.

Example: ROP?(x)
32770(y)

Output 8 (bit 15) of the second PX878 and output 2 (bit 1) of the first PX878 are set (the output count starts with 1).

OSP?

Output Signal Path Query (only PX878)
Analog outputs

Responds the source channel and sub channel of the analog output(s) of the PX878 which has (have) been selected with PCS and SPS before.

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 sub channel of the analog output(s) of the PX878 which has (have) been selected with PCS and SPS before and deactivates an eventually active test mode which could have been activated with SAO before.

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 the voltage(s) of the test signal(s) of the analog output(s) of the PX878 which has (have) been selected with PCS and SPS before. This does not mean that the test signal is active and that the returned voltage(s) is (are) currently forwarded to the output(s).

Syntax: SAO? (x)

Parameters: none

Response: voltage **,** voltage **,** … (y)

Example: SAO?(x)
1.1, -4.2, … (y)

SAO

Set analog Output (only PX878)

Analog outputs

Sets the voltage of all selected analog outputs of one or more PX878 multi IO cards to a given level (-10 V … +10 V). The command activates a test mode and “cuts” the connection of the analog output from its former-connected source signal üpath. To deactivate the test mode use the OSP command.

Syntax: SAO p1 (x)

Parameters: p1: voltage

Example: SAO 1.25(x)

Note: This command causes a lot of CPU load. Setting 10 values per second of a single analog output raises CPU load by about 15 %.

Important

This command is implemented in PMX firmware 2.00 and higher.

21.3 Examples

Simple case of measurement values configuration

Terminology:

PMX command list example in a Telnet session under Microsoft Windows

PMX Names	Catman Interface Names
Equipped 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 record subchannels that have been put into measurement rate groups.

Please do not change this example in this document, because an example 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 itself is part of query commands, which return values other than acknowledge.

This means:

Set a Butterworth 1000 Hz filter globally.

Select **all** cards	Sellect **all** their subchannels	Select filter characteristic Butterworth with **1000 Hz** cut-off frequence
pcs **0**	sps **0**	Sfc 141, **969**

Sort cards and their subchannels into measurement rate groupes.

Select card **1**	Sellect their subchannels **3** and **4**	Put the selection into measurement rate group **0**
pcs **1**	sps **3,4**	mrg **0**
Select card **2**	Sellect their subchannels **1** and **2**	Put the selection into measurement rate group **1**
pcs **2**	sps **1,2**	mrg **1**

Set recording rates for measurement rate groupes.

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 where put into measurement rate groupes above).

Select card **1**	Set their subchannels **3** and **4** for recording	Select subchannels **3,4** of the previously selected card 1	Set signal "**gross**" (= actual dynamic measurement values) for selected subchannels
pcs 1	sms 3,4	sps 3,4	mss214

Select card **2**	Set their subchannels **1** and **2** for recording	Select subchannels **1,2** of the previously selected card 2	Set signal "gross" (= actual dynamic measurement 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

Record **10** lines of values: 10 values for each signal for each measurement group = 10tsv • 2mrg • 1pcs • 2sms • 1mss = 40 values (the structure in the mrgs being identical)
tsv **10**

Check, if 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.

Get the values that have been recorded into the fifo buffer

Get **10** line of meas rate group **0** values and adjust the fifo internal read pointer (constant ADJUST_READ_POINTER = 6409)	Get **10** line of meas rate group **1** values and adjust the fifo internal read pointer
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 float format. (10 lines (sets) of two subchannels, each with one active signal.)	

Note:

- Spaces within or between commands are ignored and optional. The commas between the parameters are essential.
- Channel and subchannels count 1, 2… and 0 means “all”, where measurement rate groups count 0, 1, 2, as there is no “all”

- Some commands may be optional, because they only select the default. If the default has not been changed before, they can be omitted:
  pcs 0   sps 0   mss 214
- The default of the recording mask (if never set before) is: all equipped channels (except time channel), all their subchannels, and signal “gross” for all of them.
- A value *line* is also called a value *page*. It is a set of values defined by the measurement rate group configuration. For the above example, this means, the lines have the size of two float values for each measurement group, because we have put two subchannels (each has one active signal) into a group. A float value consists of 4 bytes. Therefore the measurement values will arrive as a multiple of 2 float values (size(line)=2) or 8 bytes, respectively
- In the above example selections via pcs and sps are done twice. That can be packed:

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 with:

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 directory

The OD is a collection of PMX setup and status parameters. A PC or PLC program can change PMX parameters, that formerly could be edited via the GUI only.

The PMX OD is **not** the EtherCAT dictionary of cyclic data objects. The EtherCAT objects are visible to the EtherCAT master only. The object dictionary is implemented in firmware version 3.02 and higher.

22.1 Accessible Data Objects

The object directory (OD) contains all relevant parameters of the PMX plug-in cards (except for the fieldbus card) and the computing channels that have been created in PMX. Objects for changing the operator level are also included.

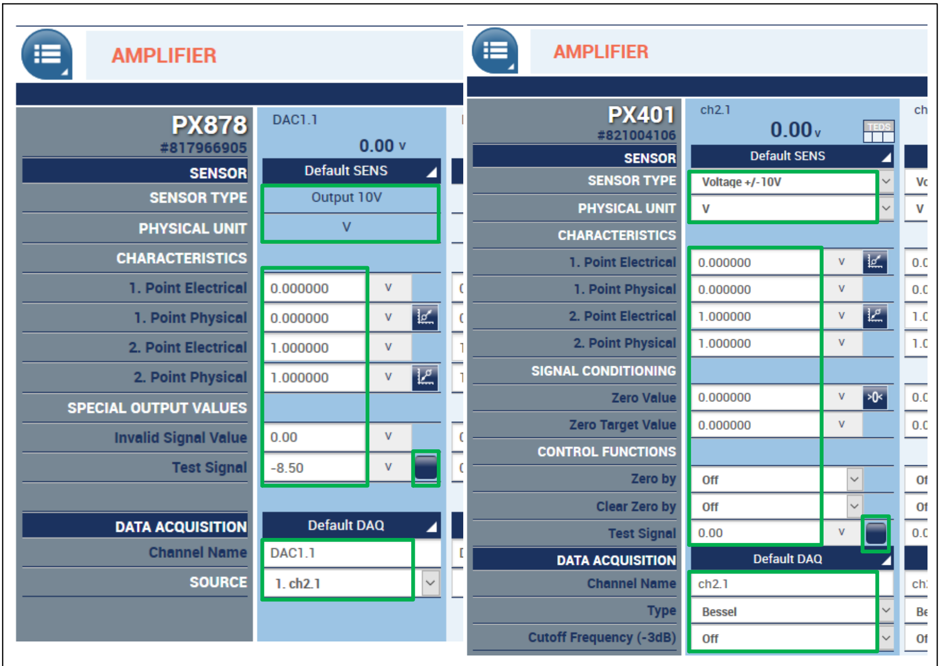
All other available device parameters can be found in the PMX command set (section 21), the dotNET-API or the device description files of the fieldbuses (section 16).

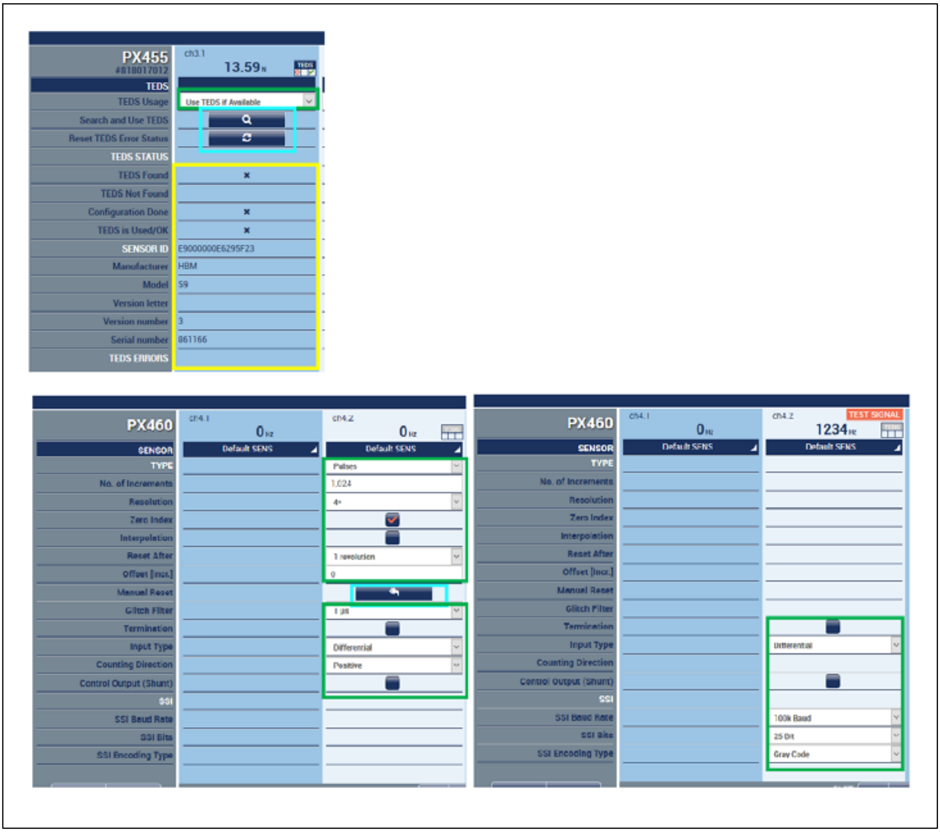
The marked parameters in the screenshots give an impression of the accessible data objects. Data objects that are in any case transferred periodically via the PC Ethernet interface or fieldbus are not visible in the OD.

Read only	Read / Write	Write only

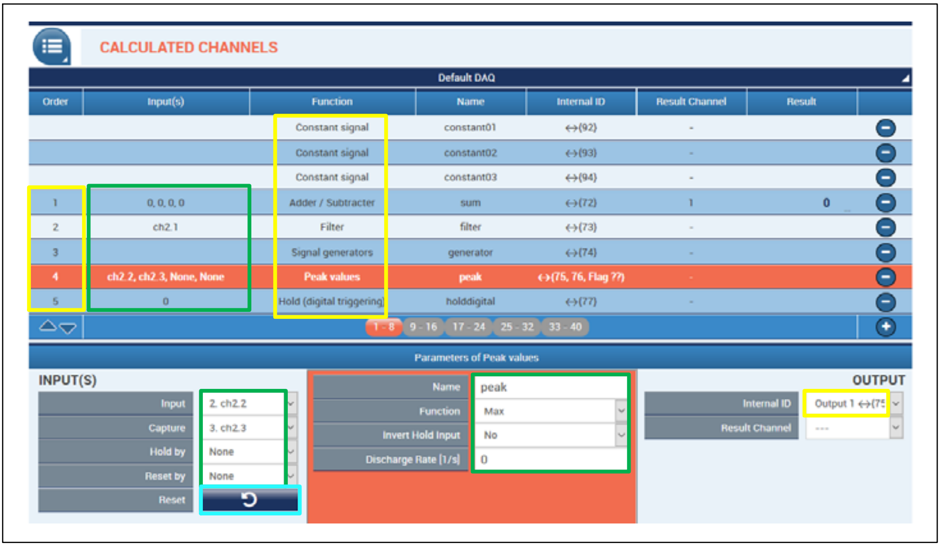
22.1.1 Measuring channels

The OD contains virtually all parameters from the amplifier dialog.




PX455
#818017012
ch3.1
13.59 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
E9000000E6295F23
Manufacturer
HBM
Model
S9
Version letter
Version number
3
Serial number
861166
TEDS ERRORS
PX460
ch4.1
ch4.2
0 Hz
Default SENS
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
Pulses
1,024
4x
1 revolution
0
1 µs
Differential
Positive
TEST SIGNAL
1234 Hz
100k Baud
25 Bit
Gray Code

22.1.2 Calculated Channels



22.2 The Numbering Scheme

A data object is addressed by

- The index 0x4000 ... 0x41ff, which is usually indicated hexadecimal.
- The subindex 0 ... 255, which is usually indicated decimal.

For example, 0x4123.45 denotes the data object with index 0x4123 and subindex 45.

22.2.1 General Objects

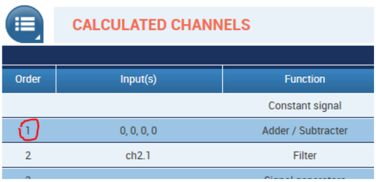
Index	Name	
0x4001.1	apply	Writing ‘1’ to this object applies the formerly changed parameters. Details in chapter 7.
0x4002.1	save_all_params	Save all settings non-volatile. That parameter calls the same function as the store icon  in the web GUI. Note: The command returns immediately, allthough the storage process takes some seconds.

22.2.2 Measuring channels

Index	Data objects of ...	Subindex
0x40**10**	Slot **1**	The subindices depend on the actually fitted module type. Find them listed in the generated files.
0x40**11**	Slot **1**, signal **1**	
0x40**1x**	Slot **1**, signal **x**	
0x40**20**	Slot **2**	
0x40**21**	Slot **2**, signal **1**	
0x40**30**	Slot **3**	
0x40**40**	Slot **4**	

22.2.3 Calculated Channels

Function block's index = 0x40A0 + calculation order

Index	Data objects of ...	Subindex
0x40A1	The function block at calculation order 1 	The subindices depend on the function block type. Find them listed in the generated files.
0x40A2	The function block at calculation order 2	
...		
0x40D0	The function block at calculation order 48	

22.2.4 The Constant Signals

0x40E1	User defined constant signal #1	8: Output signal ID 22: Name 30: Value
0x40E2	User defined constant signal #2	
...		

22.2.5 Passwords

The passwords in the PMX browser for the Maintenance level and Admin level can be turned off temporarily, for example by a higher-level PLC as service access.

The passwords are turned off by data object 0x4003 sub-index 1 in the object directory 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. Both bits can be set simultaneously (Admin level unlocked)

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: Maintenance level enabled for 1440 minutes

0x4003.1 = 0x0000 0000: Operator level, GUI locked by passwords.

22.3 Data Types

The OD supports these data types from IEC 61131

BOOL	1 bit	
USINT	8 bits unsigned	
SINT	8 bits signed	
UINT	16 bits unsigned	
INT	16 bits signed	
UDINT	32 bits unsigned	
DINT	32 bits signed	
ULINT	64 bits unsigned	not accessible via field bus
LINT	64 bits signed	not accessible via field bus
REAL	32 bits floating point	
LREAL	64 bits floating point	not accessible via field bus
STRING		not accessible via field bus

22.4 Access via Ethernet Command Interface

For general information about the command interface refer to the PMX Operating Manual, chapter PMX Comman Set (API).

The command **oda** (Object Dictionary Access) writes or read single data objects via ethernet Port 55000..

Query	oda? *index*,*subindex*	Query the data object's value index: The data object's index in either decimal or hexadecimal notation subindex: The data object's subindex in either decimal or hexadecimal notation
Reply	index,subindex,value, error_code	Reply from PMX index: The index from the query in decimal notation subindex: The subindex from the query in decimal notation value: The data object's value error_code: 0 – no error, the returned value is valid 1 – access error (eg. read-attempt to a write-only object) 2 – format error (eg. the data type not supported) 4 – not found, the data object does not exist

Example:

oda? 0x4011,13
(read object 0x4011.13 , slot1.1 output voltage (PX878 analog output))

16401,13,0.125,0
(success, the voltage is 0.125 V)

oda? 0x4fff,1 (read object 0x4fff.1)

20479,1,0,4 (fail, the data object does not exist)

Set	oda *index*,*subindex,value*	Set value index: The data object's index in either decimal or hexadecimal notation subindex: The data object's subindex in either deci-mal or hexadecimal notation value: Value to be set. It is the responsibility of the programmer that the value can be converted into the data object's type.
Reply	index,subindex error_code	Reply from PMX index: The index from the query in decimal notation subindex: The subindex from the query in decimal notation error_code: 0 – no error, the parameter has been updated 1 – access error (eg. write-attempt to a read-only object) 2 – format error (eg. wrong data type) 4 – not found, the data object does not exist

Example:

oda 0x4011,13,1.2
(set slot1.1 test signal (=output voltage) to 1.2 V)

16401,13,0
(success)

oda 0x4011,14,1
(enable slot1.1 test signal)

16401,14,0
(success)

oda 0x4011,19,"my channel" (set the channel name of slot1.1)

16401,19,0
(success)

22.5 Access via Field Bus

For general information about field bus communication refer to the PMX Operating Manual, chapter

Communication with a Control System.

Note: Only data objects whose value fit into 32 bits are transfered on the field bus. That is BOOL, SINT, INT, DINT, USINT, UINT, UDINT, REAL.

LREAL values are transfered as REAL values with a certain loss of accuracy.

Before submitting a request via field bus, the service must be enabled with bit 1 in the Device Control Word: PROFINET, EtherCAT:

Output data controller (PLG) ⇒ PMX

Device data (cyclic)

PROFINET/EtherCAT

Function		EtherCAT® Index	PROFINET Slot.Subs-lot	Date type
Device control word	Bit1 (value 0x02): Enable Object Directory server	7000.1	0.2 bytes 0..3	uint32

EtherNet/IP

Function		EtherCAT® Index	PROFINET Slot.Subslot
Index	**Size in octets**	**Type**	**Tag**
0..3	4	UDINT	PMX Control

22.5.1 Submitting a Request

Read and write requests to the OD are submitted by the (so far unused) 'GUI signaling' data word. Find the bit allocation in chapter 16.5.1 and 16.9.3.

Output data, controller (PLC) ⇒ PMX

Device data (cyclic)

PROFINET/EtherCAT

Function		EtherCAT® Index	PROFINET Slot.Subs-lot	Daten-typ
GUI signaling	Command obejct directory	7000.3	0.2 bytes 8..15	uint64

Ethernet/IP

Index	Size in octets	Type	Tag	
8..15	8	ULINT	UiControl	SystemData (transmitted al-ways)

22.5.2 Bit Allocation

Bit allocation for both, request and reply

<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 zero !</td><td colspan="2">INT, UINT</td></tr>
<tr><td></td><td></td><td colspan="2"></td><td colspan="3">Not used, set 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 num-ber in octet**	**PLC sets control bits PMX sets status bits**	
63	7	control	Read request. Set this bit to read a data object.
62	6	control	Write request. Set this bit to write to a data object.
61	5	control	Repetitive read (not available with a write request) 0: PMX replies once 1: PMX updates the reply permanently until next request
60	4		not used
59	3		not used
58	2	status	Not found, the data object does not exist

57	1	status	Format error (eg. wrong data type)
56	0	status	Access error (eg. write-attempt to a read-only object)

Bit allocation of REAL values (32 bits floating point)

Sign	Exponent	Fraction
bit 31	bits 30..23	bits 22..0

22.5.3 The Reply from PMX

PMX replies in the (so far unused) 'GUI status' data word. Find the bit allocation in chapter 16.4 and 16.9.3

Input data, PMX ⇒ controller (PLC)

Device data (cyclic)

PROFINET/ EtherCAT

Function		EtherCAT® Index	PROFINET Slot.Subslot	Daten-typ
GUI status	Answer object directory	6000.3	0.2 bytes 8..15	uint64

Funktion		EtherCAT® Index	PROFINET Slot.Subslot	Daten-typ
Index	**Size in octets**	**Type**	**Tag**	
8..15	8	ULINT	UiStatus	System-Data (trans-mitted al-ways)

22.5.3.1 Reply to a Read Request

PMX copies the index, subindex, the control flags, and the requested value into the reply.

The request has been served successful, when bits 32..63 of the reply equal bits 32..63 of the request. Then bits 0..31 contain the requested value.

Don't evaluate the value when one of the error flags is set.

22.5.3.2 Reply to a Write Request

PMX copies all data from the request into the reply.

The request has been served successful, when all bits of the reply equals the request.

22.5.3.3 Retry

For retrying a failed request, at minimum one bit of the request must change. Generally, PMX waits for changes in the request.

Example:

For retrying a read request, toggle the read control bit (bit 63)

1. Set read control bit = 0
2. Wait for read control bit == 0 in reply
3. Set read control bit = 1
4. Check the reply

22.6 Make the New Value Effective

Most of the values require an explicit application after changing.

Write “1” to 0x4001.01 to make effective the changed values (command “oda 0x4001,1,1“).

It is good practice to change all parameters firstly, then set 0x4001.01 to apply them alltogether. (Note: In contrast, every parameter change in the web GUI becomes effective immediately.)

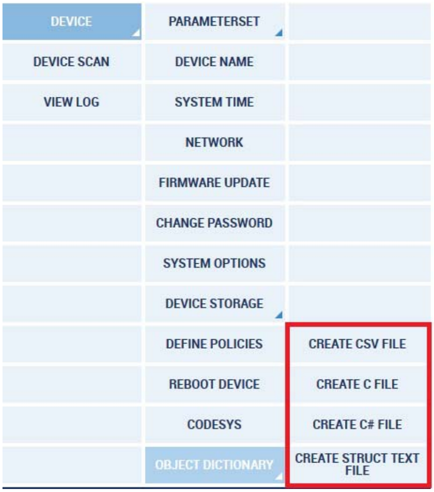
For these values an explicit application is not required, they are applied immediately when writing to:

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

PMX generates header files to ease your programming work.

First, setup the calculated channels with the web GUI. Then let PMX generate a csv, C, C#, or ST (Structured Text) file. In the trial firmware, the file creation buttons are located here:



Please note that two **C** files are created. The browser's download box appears two times.

The **Structured Text** (ST, SCL) file for PLCs is importable in selected PLC configuration tools only. If not, copy-and- paste the content in your source code.

The files contain a list of all data objects, types, and constants definitions.

Important

Note that changing the hardware configuration and adding, moving or deleting computed channels changes the object directory. After that the OD files must be generated again.

22.7.1 Value Ranges of the Objects

Most data objects are just numbers. The legal data range is indicated in the object's list.

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_CHARACTERIS-
TIC"}
```

Find the allowed values in the .h file

/* FILTER_CHARACTERISTIC */ enum FILTER_CHARACTERISTIC{ fltBessel = 0 , fltButterworth = 1 };

Important

Please note that most of the data objects are NOT CHECKED for range violation by the firmware. That's in your responsibility.

RANGE_AS_DATA_TYPE indicates an unconstrained range. The value can be virtually every number from the data type, but is usually limited by the context behind.

```
{ 0x4021, 15, 2, 1, odREAL , 1, ACCESS_RW,
"[slot2.1] test_signal","RANGE_AS_DATA_TYPE"}
```

22.8 Hints on using the Object directory

Best order to use calculated channels objects:

1. Setup the calcuated channels with the web GUI
2. Let PMX create the files with definitions and data objects
3. Edit the data objects with your PC or PLC program

Important

When calculated channels change their calculation order, the corresponding data objects change their index, too. When function blocks are created / deleted, the corresponding data objects are created / deleted as well. The most of the data objects are NOT checked for range violation. It's the responsibility of the user to write legal data. Performance when accessing data objects via fieldbus is usually 25..35 ms per request.

23 Firmware update

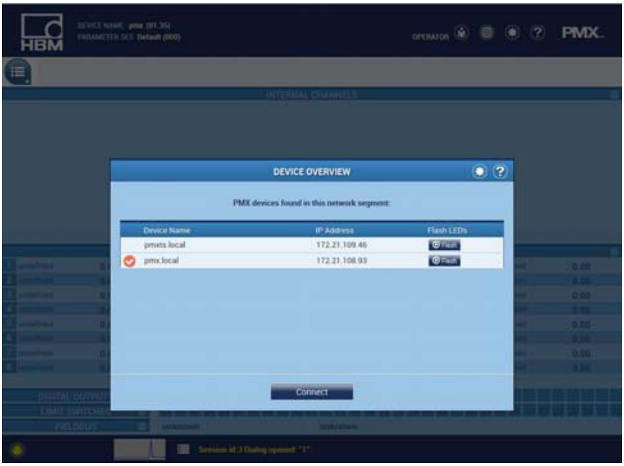
23.1 Preparation

An update can be applied to individual PMX devices or to several devices at the same time. The PMX must be connected to the PC (HOST).

In all cases, a firmware update will take approx.15 minutes. The device is *not* ready for measurement while the firmware is being updated.

To identify a specific device:

► Select a device (checkbox) and click on **FLASH**.
All the controllable LEDs (system LED, measurement card LED) on the selected PMX will flash to make the identification

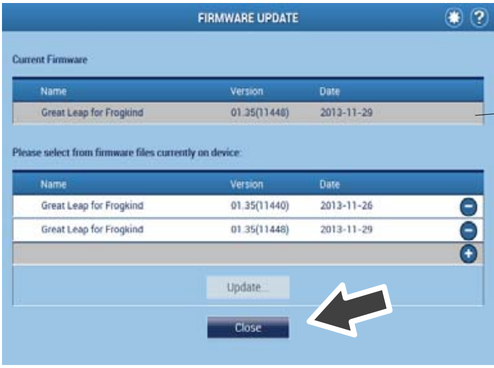


► Click **Connect**

- Copy the firmware file to the local PC (HOST), e.g.: "PMX_01.10-7412M.tgz"

23.2 Install firmware

1. Select the menu Settings / System / Device / **FIRMWARE UPDATE**



2. Select the firmware you want to update by clicking on the relevant entry. A tick will indicate your selection. If the firmware version you require is missing, it can be added by pressing the **+** button.
3. Press the **Update** button to transfer the firmware. The browser will then re-connect to the device.

Important

One of two cases may occur if the voltage supply fails during the firmware update.
The device comes online again with the old firmware after being switched on or the device loads and installs the new/loaded firmware and is then ready for operation after approx. 10-15 minutes.

Tip

The device setting and parameter sets are retained through a firmware update.
We recommend creating a backup on a PC before the firmware update.
In firmware version 2.00 and higher, CODESYS applications and CODESYS WebVisu are also retained after a firmware update to a higher firmware version.

To download the current firmware go to *http://www.hbm.com/menu/support/software-firmware-downloads/industrial-amplifier*

24 Diagnosis and maintenance

Before actually starting to measure, you should check your system.

24.1 Error messages / Operating state (LED display)

For the system to be ready for measurement, the LEDs on the basic device and modules must indicate the states described in *Chapters 8.2.3 to 8.2.5 and Chapter 8.1*.

If this is not the case, follow the instructions under "Remedy" in the tables below.

SYS LED:

LED	Status	Significance	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 ongoing	Check the mounting of the plug-in card and replace if necessary

PX01EC, EtherCAT®

LED	LED	Status	Significance
ERR Error status	-	OFF	No error
	red	Flashing	Configuration error, the PLC configuration (Master) must correspond exactly to the PMX configuration (Slave): Card types in slots 1 to 4 and number of calculated channels (see PMX web browser in the "Fieldbus" dialog)
		Single flash	Synchronization error
		Double flash	Application timeout error
		On	PDI timeout error

PX01PN, PROFINET

LED	LED	Status	Significance
SF System error	red	On	No connection or no valid license
		Flashing	Incorrect configuration, i.e. the configuration on the PROFINET Master side (PLC) must correspond **exactly** to the configuration of the PMX (card types in slots 1 to 4 and number of calculated channels). See PMX web browser in the dialog "Fieldbus"
BF Bus error	red	On	No connection or no valid license
		Flashing	Incorrect configuration, not all IO devices are connected

Ethernet / IP

LED	LED	Status	Significance
MS Modus status	**Duo LED red/green**		
	-	OFF	**Not turned on:** The device has not been turned on.
	green	On	**Device ready for operation:** The device is in operation and is working correctly.
		flashing	**Standby:** The device has not been configured.
	red	On	**Fatal error:** A fatal, unrecoverable error has been detected in the device.
		flashing	**Simple error:** A simple error that can be corrected has been detected in the device. The PLC configuration (Master) must correspond exactly to the PMX configuration (Slave): Card types in slots 1 to 4 and number of calculated channels (see PMX web browser in the "Fieldbus" dialog) Note: An incorrect or inconsistent configuration is classified as a simple error, for example.
	red/green	flashing	**Self-test:** The device is running a self-test.
	-	OFF	**Not turned on:** The device has not been turned on.

LED	LED	Status	Significance
NS Network status	**Duo LED red/green**		
	-	OFF	**Not turned on, no IP address:** The device has got no IP address (or has not been turned on).
	green	On	**Connected:** The device has at least one established connection to another device (also to the message router).
		flashing	**No connections:** The device has no established connection to any other device, however, it has got an IP address.
	red	On	**Duplicate IP:** The device has detected that its IP address is already in use.
		flashing	**Connection timeout:** One or several connections to this device are in timeout status. This status will only be terminated when all connections in timeout status have been re-established or when the device has been reset.
	red/green	flashing	**Self-test:** The device is running a self-test.

PX401, channel status

LED	Status	Significance	Remedy
green	On	No error	-
yellow	Flashing	Firmware update ongoing	
red	On	Parameters not OK, overloaded	Check: Sensor, sensor leads, TEDS module, send in the card if necessary

PX455, channel status

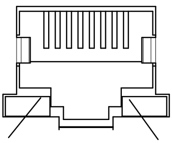
LED	Status	Significance	Remedy
green	On	No error	
yellow	On Flashing	No transducer connected or wire break (calibration ongoing) Firmware update ongoing	Transducer connection
red	On	Parameters not OK, transducer error, overloaded	Check: Sensor, sensor leads, TEDS module, send in the card if necessary

PX878, channel status

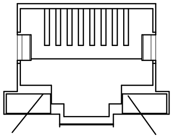
Analog			Remedy
green	On	Analog output configured	
yellow	Flashing	Firmware update ongoing	
red	On	Analog output overloaded, signal invalid or no signal assigned	Check the sensor signal, check the settings for the analog output channel

Synchronization SYNC

IN socket LEDs:

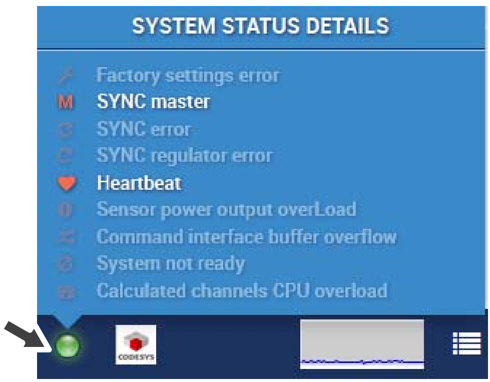
IN		Significance	Remedy
			
green	Off	Slave	
Off	Off	Master	
Off	yellow	Error	Check the cable connection to the master/slave

OUT socket LEDs:

OUT		Significance	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

24.2 Error messages of the device status

The device status is signaled directly on the PMX via the device LED (green=OK / red= error). In the case of an error message, detailed information can be called up about the device status via the web browser, double-click on the system LED, the PMX command set or the fieldbus.



24.2.1 Error in factory settings

Production data missing (serial No., Prod-Date,0). The device was not tested at the HBM final inspection place. The system LED is flashing yellow.
The device is still operational without restrictions.

24.2.2 SYNC Master

Status bit, no error. If set, the device is the Sync Master, i.e. no Sync signal was detected at the Sync IN socket.

24.2.3 SYNC error

Faulty or missing Sync telegrams. Indicates connection problems at the Sync IN socket.

24.2.4 SYNC controller error

The device cannot follow the Master as Slave. The controller is saturated. The time stamp and CF are not synchronized.

24.2.5 Heartbeat

System bit, no error. Flashing with approx. 1Hz. CPU error present if at standstill.

24.2.6 Sensor supply overloaded

The sensor supply was switched off due to excess current on at least one measurement card.

24.2.7 Buffer overflow in command interface

Measured values lost due to buffer overflow in catman®/Command interface.

24.2.8 System not ready

Device is damaged and not delivering valid measured values.
Set to temporary during parameter set switching (ok).
Set to static if parameter set does not agree with card configuration. Occurs when cards are removed/updated/replaced or if an unsuitable parameter set was imported.

24.2.9 CPU overload during calculations

Computing time overrun in calculated channels. The consequences can be gaps in the measured data flow.
Temporarily not harmful during parameter set switching, editing of calculated channels or self-calibration of bridge inputs.
If set during normal operation: Reduce number of function blocks and/or reduce global sampling rate (38.4kHz -> 19.2 kHz).

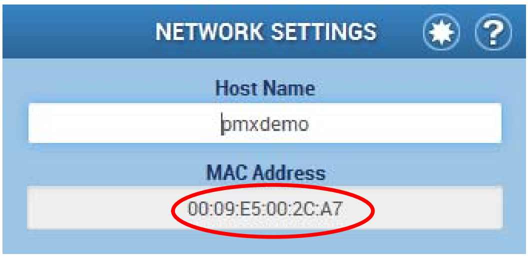
24.3 Resetting the PMX administrator password

This procedure has no effect on the password for maintenance work.

1. User

Send both the PMX host name and the MAC address to the HBM Technical Support Center.

You can find the host name on the overview screen ("Overview") and in the network dialog ("Network"). You can find the MAC address in the network dialog ("Network") and on the label on the base.



2. HBM Technical Support Center

The HBM Technical Support Center (TSC) will create a file with signature.

The signature will be saved in a file with the name "pmx-password-reset" and sent to the user.

3. User

Copy the "pmx-password-reset" to the root directory of a USB stick. Connect this stick to the USB socket of the PMX system while PMX is in normal operation.

The administrator password will be removed immediately. The file will be deleted from the USB stick.

You may want to keep a copy of the Reset file in order to be able to reset the password again in the future. This is possible with a specific PMX device as long as the host name is not changed.

24.4 Resetting the PMX to factory settings

All device settings can be reset in the menu Settings > System > Device > Device administration > Restore factory settings.

This function is not accessible in user level 1 (Worker).

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 various user levels (Worker, Service, Administrator)

24.5 Restoring lost PMX network settings and device names

If you cannot find the PMX in the network, you can use a USB memory stick to provide the network settings you require.

1. On a USB memory stick, create a text file called pmx.conf in the root directory
2. Example 1:

   This pmx.conf file sets the device name to "pmx_new_name" and switches PMX into the DHCP mode

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

3. Example 2:

   Sets the name to "pmx", as well as a permanent 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>
   ```

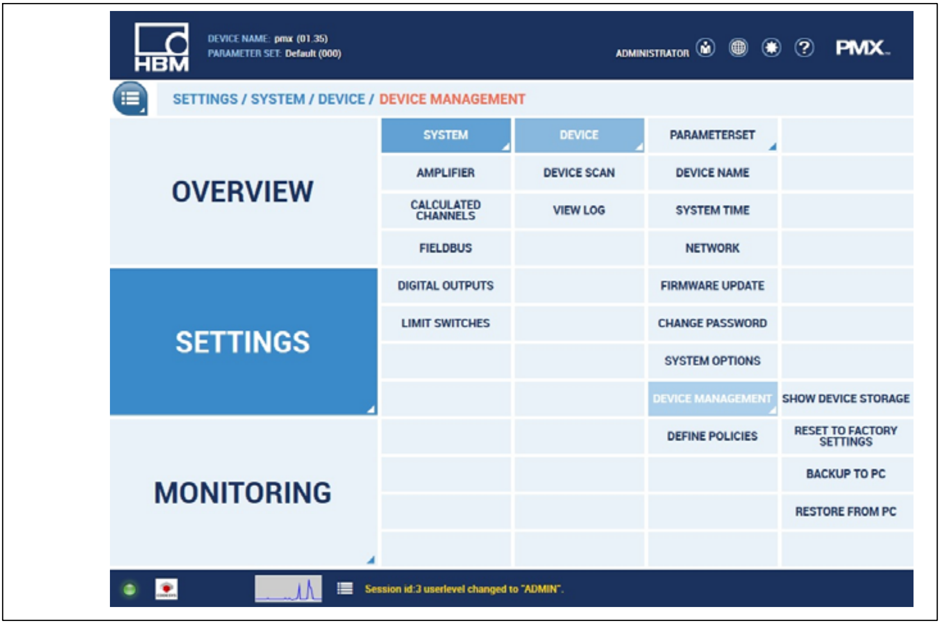
4. Plug the USB stick into the PMX device *while operation is ongoing*.

   The settings will change immediately, but will not be immediately apparent in the other network devices. So it is a good idea to to restart the PMX by interrupting the power supply.

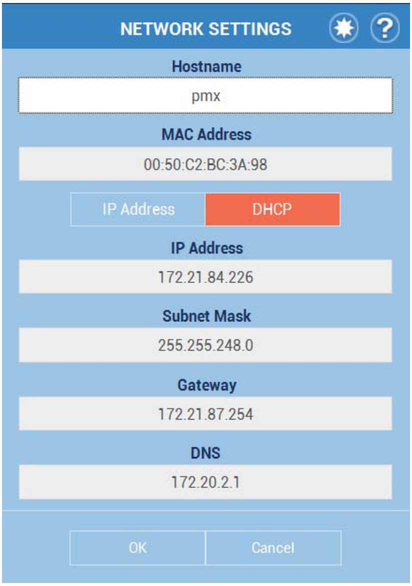
5. The PMX can be found in the network under the new settings.

6. Caution: This memory stick converts each PMX device as soon as you plug it in!
   Once you have used it, you should therefore delete the file, rename it or move it to a subdirectory.

Change the network settings



Network settings



24.6 Saving and restoring PMX device settings and CODESYS applications

If all the device settings, parameter sets, network settings, and the CODESYS application are to be transferred from one PMX to another PMX (cloning the device), you can use a USB memory stick. Passwords cannot be transferred in this way and must be modified manually via the PMX browser (also see Section (also see Chapetr 10.3.2).

WARNING

Make sure that the configuration of both devices is the same, as otherwise the settings cannot be transferred and malfunctions may occur.

1. On a USB memory stick, create a text file called "pmx.conf" in the root directory. 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:

2 Example 1:
Saving all the PMX device settings <u>with</u> the network settings to the USB flash drive:
<pmx type="save"> path="defaults.pmx" />
When saving via the PMX browser, the parameter set file “defaults.pmx” can also be created at: "Settings->System->Device->Device memory->Backup on PC".

3 Example 2:
Loading all the PMX device settings <u>without</u> the net-

work settings from the USB flash drive:
<pmx type="load" path="defaults.pmx" />

4 Example 3:
Restoring all the PMX device settings with the network settings from the USB flash drive:
<pmx type="restore"> path="defaults.pmx" codesys="codesys.tgz"/>

5 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. One or more applications must first be uploaded to the device with the CODESYS IDE (development environment). Then the "codesys.tgz" file can be saved under "Settings->System->Device->CODESYS->Backup on PC". This must be copied on the USB flash drive.

6 Insert the USB flash drive into the PMX while operation is ongoing. The settings will be transferred immediately.

WARNING

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.

24.7 Replacing measurement and communication cards

Measurement and communication cards can be retrofitted or removed at a later date. Please note the combination options (see Chapter 8.2.1).

After modification and switching on the supply voltage, the PMX automatically detects and initializes the hardware configuration.

Notice

If measurement cards are removed after being updated or mounted in different slots, the factory settings are loaded. All parameters must be reentered, also for the existing cards.
If communication cards (EtherCAT, PROFINET or EtherNet/IP) are updated, replaced or removed, the parameter settings are retained. Adaptation to the new fieldbus is necessary here via the configuration tool of the controller.

Important

A CODESYS application or web visualization running on the PMX will also be retained after a firmware update or if a card is changed (firmware 2.00 or higher). Please note that signal mapping in CODESYS is fixed and if the measuring cards are moved or the PMX computing channels are changed, the signal mapping must be checked and corrected if necessary.

24.8 Log file

PMX is equipped with an automatic log function to improve operating security. The user inputs in all three user levers and all (error) messages of the PMX are recorded and stored internally in the device.

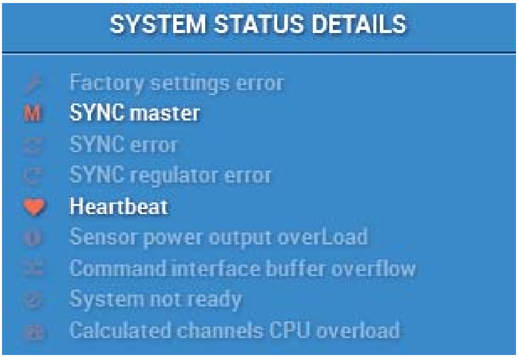
The device and channel statuses are also written and saved.

This enables simple and unique analysis in the case of error. User level 1 (worker) has no rights here to delete the file.

The file size can be used to set the size of the log files between 500 kByte and 20 MByte at user level 3 (administrator). Optionally, the messages can also be transferred 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 called via the symbol (see below) or the user menu (Settings>System>View Log).

24.8.1 System log entries for system status



- Sync-Master or –Slave:
  No sync input. Switching to master mode.
  Sync available. Switching to slave mode.
- Sync error
  Too many CRC errors on sync input
  Temporarily switching to master mode
- Sync regulator error
  Cannot lock to incoming sync signal
- Sensor power output overload:
  System status Sensor-Power-Output-Overload has been "activated"
  System status Sensor-Power-Output-Overload has been "deactivated"
- Command interface buffer overflow:
  System status Command-interface-buffer-overflow has been "activated"
  System status Command-interface-buffer-overflow has been "deactivated"
- Calculated channels CPU overflow
  Calculated channels runtime over

The statuses for:

"Error in factory settings"
"Heartbeat"
"System not ready"

are not written or saved.

24.8.2 System log entries for channel status / measured value status

- If the channel status changes from 0 to not equal to 0, i.e. at least 1 error is set again but no error was previously present, then the message: "Measval-status changed. New status: "invalid". Slot:X, Signal:Y" is generated.
  If an error was already present and a new one occurs, then no message is generated.
- If the measured value status does not contain any error entries (any more), i.e. it changes to valid, then: "Measval-status changed. New status: "valid". Slot:X, Signal:Y".

24.8.3 Log entries for CAN status

The following CAN interface error states are logged when they occur:

"transmit timeout",
"lost arbitration",
"controller problem",
"procrotocol violation",
"tranceiver status",

"no ack received on transmission",
"bus off",
"bus error",
"controller restarted"

These errors often occur if the physical connection of the CAN nodes is disturbed. You should therefore check the following:

- S correct cabling,
- S correct line termination (is a bus termination resistor connected?)
- S the conventional quantity value of the CAN network bus termination resistors (also see Chapter 17).

25 Quality certification and calibration certificates

Documented quality:

At the time of delivery, PDF documents of the HBM calibration certificates for the fitted measurement cards, and a certificate of compliance as per EN 10204 2.1 are already stored in the PMX device memory (public - certificates).

The PMX browser can be used to download them in the dialog > Device memory.

If the device is re-calibrated 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.

Should the user have deleted the PDF documents in the device memory, replacements can be obtained from HBM Technical Support at support@hbm.com

26 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 would have to be maintained?
No. The PMX manages without fans, etc. and is maintenance free.
- Are the connectors protected to prevent mix-ups?
At the time of delivery, no. But you can use the enclosed coding pins to integrate coding/prevent mix-ups.
- What connector options are available?
The standard multipoint connectors supplied are "Push-in" terminals and can also be ordered as screw-on plug terminals.
- What are the options for adjusting the amplifiers?
3 options: 1. Enter the sensor values (zero point/span) as a numerical value, 2. Measure the sensor values, 3. Load and automatically set the TEDS (Transducer Electronic Datasheet) sensor values from the TEDS chip in the PMX amplifier.
- What are the options for connecting the PMX to a web server?
1. Direct 1:1 connection via Ethernet. 2. An Ethernet connection via a network.
- Do I have to install operating software?
No. PMX has an internal web server for parameterization. All that you need on your PC is an Explorer, Windows Internet Explorer (Version 9 or later), Firefox or Google Chrome. You also have the

option to use the HBM catman®EASY/AP® software for recording and data analysis.

- What must I watch out for when connecting the PMX to a PC?
  The Ethernet cable must be connected. Both nodes (PMX (factory setting DHCP) and PC) must be set to DHCP. The connection is established by entering "PMX" in the browser bar.
- Can problems occur if the Ethernet, fieldbus and synchronization RJ45 connections are mixed up?
  No. All connections are short-circuit proof. Errors can be identified via the status LEDs on the device or in the PMX web browser.
- What must I watch out for when replacing the plug-in cards?
  The PMX must be de-energized! When you switch on, all the cards are automatically detected. Any new cards that are added have to be parameterized. The factory settings are loaded. All parameters must be reentered, also for the existing cards. This does not apply to the replacement of communication/fieldbus cards.
- How can I synchronize several PMX?
  By connecting with standard Ethernet cables using the SYNC sockets. The first PMX is automatically configured as the master, all the others are automatically slaves. There is a maximum number of 20 PMX devices that can be networked.
- 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 a wide variety of regulation and control tasks to be processed in the PMX using anything from peak-value calculation to PID control, and relieves the burden on downstream systems and PLCs.
- How high are sampling and calculating rates in the PMX?
  All channels, measurement channels and calculation channels are sampled and processed at 19,200 Hz or 38,400 Hz in the PX460. This means that extremely fast measurement data processing and automation are possible. The measuring bandwidths can be taken from the specifications of the individual measurement cards.
- Is the time for the measured value time stamp stopped after a power failure?
  No, the time stamp starts at zero again after a new start.
- 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 max. 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 does the PMX have and how high are the switching times?
  A maximum of 1,000 parameter sets can be used in the PMX. These are divided into 4 sub-parameter sets which can be switched separately.

Depending on how many sub-parameter sets are switched, the switching time takes between 0.1 and 2.5 seconds.

- Can the PMX also be used as a fieldbus master?
PMX can only be used as a slave in the Ethernet-based fieldbuses (EtherCAT, PROFINET and EtherNet/IP).
If the CANopen interface is used under CODESYS Soft-PLC, the PMX can be run either as a CAN master or CAN slave.
- What happens if the supply voltage fails during the saving of a parameter set?
This will destroy the parameter set and the PMX will boot up with the factory setting after being switched back on again. To avoid this, backing up the device settings to PC is urgently recommended.
- What happens if the supply voltage fails during a firmware update?
The device either comes online again with the old firmware after being switched on or the device loads and installs the new/loaded firmware and is then ready for operation after approx. 10-15 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 turned 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 knowhow protection.
However, when a program is created, the original source code can also be transferred to the PMX and

then loaded back into the CODESYS development environment.

- What happens with a CODESYS application after a firmware update or when a measuring card is changed in the PMX?
  A CODESYS application or web visualization running on the PMX will also be retained after a firmware update or if a card is changed (firmware 2.00 or higher). Please note that signal mapping in CODESYS is fixed and if the measuring 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?
  The current versions of the firmware/PMX web server and the device description files can be downloaded at www.hbm.com/support.
- Is there an electrical design tool for PMX?
  Yes. Ready-made ePLAN macros are available at *www.hbm.com/en/menu/support/software-firmware-downloads/industrial-amplifiers* and can be used without a license.
- Are there 3D (STEP files) for mechanical design (CAE) for PMX?
  Yes. PMX STEP files are available free of charge at *www.hbm.com/en/menu/support/software-firmware-downloads/industrial-amplifiers*
- How can I get support if I have any problems?
  If you have any technical questions, the HBM TSC (Technical Support Center) support@hbm.com can help. If you have any technical project planning and design problems, our colleagues from Application Engineering will be happy to answer your questions application-engineering@hbm.com, or can come to your premises.

27 Technical support

Should you have any questions when working with the PMX measuring amplifier system, HBM's technical support can provide:

E-mail support

support@hbm.com

Extended support can be obtained through a maintenance contract.

Fax support

06151 803 288 (within Germany)

+49 6151 803 288 (international)

The following options are also available:

HBM on the Internet
http://www.hbm.de

Download software updates from HBM
http://www.hbm.com/en/menu/products/industrial-amplifiers/pmx/

Headquarters world-wide

Europe

Hottinger Baldwin Messtechnik GmbH:
Im Tiefen See 45, 64293 Darmstadt, Germany
Tel. +49 6151 8030, Fax +49 6151 8039 100
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

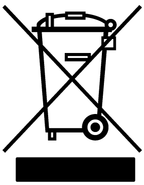
Asia

Hottinger Baldwin Measurement (Suzhou) Co., Ltd.
106 Heng Shan Road, Suzhou 215009, Jiangsu,
VR China
Tel. (+86) 512 68247776, Fax (+86) 512 68259343
E-Mail: hbmchina@hbm.com.cn

Up to date addresses of representatives can also be found on the Internet under: www.hbm.com/Contact/Worlwide-Sales-offices

28 Waste disposal and environmental protection

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 mark

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

29 Glossary

APIPA RFC	*Automatic Private IP Addressing*, APIPA is designed for being able to operate a TCP/IP network without needing to be confronted with IP addressing and IP parameters. In Microsoft Windows, automatic IP address assignment has been implemented since Windows 98. It does not however correspond completely with the RFC of the IETF. 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 measured value processing: Measurement, analysis and evaluation of large measurement data sets, including mathematical and graphical functions (statistics, signal analysis, digital filters)
CAN Bus	The **CAN-Bus** (**C**ontroller **A**rea **N**etwork) is a serial bus system and is one of the field bus systems.
CAT5-SFTP	CAT 5 screening. The S/FTP-version (Screened Foiled Twisted Pair) is configured similar to FTP, but has an additional complete screening (copper mesh sheathing) around the wire bundle
CODESYS	CODESYS is a development environment for programmable logic controllers (PLC) according to the IEC 61131-3 international industrial standard for the development of applications in industrial automation.

Communication card	The PMX basic device (WGX001/ WGX002) can be optionally equipped with a fieldbus communication card in Slot 0. This can enable the connection to a fieldbus master (PLC) via ProfiNET, EtherCAT or Ethernet/IP. This form of automation permits deterministic data transmission, i.e. data transmission at previously specified intervals.
Crossover cable	A *crossover cable* in computer network technology (LAN technology) is an eight-wire Twisted-Pair cable, in which *one* of the two RJ45 connectors has specific cable wires switched around. While an uncrossed (*straight through*) network cable connects a computer to switches, a crossover cable can directly connect two computers (or two switches) with each other.
DHCP	The *Dynamic Host Configuration Protocol (DHCP)* enables the assignment of network configurations to Clients via a Server. (WIKI)
EtherCAT®	*EtherCAT* ("Ethernet for Controller and Automation Technology") initiated by the company Beckhoff is a real-time Ethernet. The protocol made public in the IEC standard IEC61158 is suitable for hard and soft real-time requirements in automation technology. The main emphasis in the development of EtherCAT was focused on short cycle times (≤ 100 μs), low jitter for precise synchronization (≤ 1 μs) and low hardware costs.

Fieldbus	A *fieldbus* connects in one system field devices such as detecting elements (sensors) and control elements (actuators) for communication with a control device. 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). Standardized protocols exist for this, e.g. yyyy
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. Profinet-IO devices can be described with the XML-based GSD. International standards led to the development of the GSD file description language: GSDML (**G**eneric **S**tation **D**escription **M**arkup **L**anguage). As the name already states, this is an XML file (e**X**tensible **M**arkup **L**anguage) that is language-independent.
Greenline	HBM shielding design that ensures that HBM products can function safely without interference and that no interferences escape to the environment or that supply networks are not exceptionally overloaded.
GUI status	Control word for transmission of data via the PMX web browser to a connected PLC (this function is currently not activated).

Host	The host name (also site name) is the unique designation of a computer in a network. It is primarily used during electronic data exchanges (e.g. E-mail, Usenet, FTP) to provide communication partners with a human readable and visible format.
Industrial Ethernet	Ethernet-based fieldbus protocols are known as Industrial Ethernet (e.g. PROFINET).
Input/Output card	The PMX basic device (WGX001/WGX002) can be freely equipped 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 a measurement card for the acquisition of measurement signals in Slot 1-4.
NETBIOS	*NetBIOS* (English abbreviation for ***Net****work* ***B****asic* ***I****nput* ***O****utput* ***S****ystem*) is a programming interface (API) for communication between two programs via 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. The Profinet concept is modular in structure so that users can select the functionality themselves. This differs

	mainly by the type of data exchange in order to meet the speed requirements.
PROFINET-IRT protocol	
	The 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 up to maximum 1 µs. IRT is required, for example, in motion-control applications (positioning processes).
Push-In technology	Simple, tool-free wiring with terminal technology. This technology enables easy and direct connection of fixed and flexible wires with end sleeves from 0.34 mm^2 upwards. A contact spring opens automatically and provides the necessary pressure force against the current bar.
RJ45	RJ plug connectors are the plug connectors standardized by the USA 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 connector type. In the network sector, every fully equipped eight pin (8P8C) modular plug connector is usually called "RJ-45"
RailClip	Various electrical operating media (for instance, relays) can be pushed laterally onto a support rail with a

	U-shaped profile or pushed on from the front and locked in place. In English, the DIN rail is usually just called 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 the Internet Assigned Numbers Authority assigned it the UDP Ports 67 and 68.
Simatic manager	The SIMATIC manager administrates all data that belong to an automation project – irrespective of which target system (e.g. SIMATIC S7) they are realized on.
TEDS	The acronym TEDS stands for "Transducer Electronic Data Sheet" and refers to the electronic data sheet of a transducer or sensor that is stored in a small electronic chip or appropriate module which is permanently connected to the device. In addition, valuable metadata such as calibration data is provided, which gives important information for the traceability of measurements or tests. The electronic data sheet can be located in the transducer housing, in the inseparable cable or connector plug.
TwinCat	The Beckhoff *TwinCAT* software system converts almost all compatible PCs into a real-time controller with a Multi-PLC system, NC axis control, programming environment and operating station. TwinCAT substitutes conventional PLC and NC/CNC control systems as well as operating units

TCP-IP	*Transmission Control Protocol / Internet Protocol (TCP/IP)* is a family of network protocols and is, because of its great significance for the Internet, also called the Internet protocol family. The identification of computers participating in the network is implemented via IP addresses. A computer or any device with an IP address is called a *Host* in TCP/IP jargon. Originally, TCP was developed as a monolithic network protocol, but was subsequently split into the protocols IP and TCP. 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.
Telnet	Telnet (*Telecommunication Network*) is the name of a widely distributed Internet network protocol. This old and well-known Client/Server protocol is based on a 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 a unique time to an event
V/G-strip	The DIN standard **DIN 41612**, generally know as the *VG-strip*, 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 e. g. web browser. A web server refers either to a computer with web server software or 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 and worldwide according to the required purpose. The main task of a web server is the delivery of static files, e.g. non-modifiable HTML or image files, or dynamically generated files, e.g. pages with contents that are generated individually according to the profile of a logged in user.

30 Index

I

L

M

N

P

R

S

T

U

V

W

www.hbm.com

A4354-2.2 7-2002.4354 HBM: public

HBM Test and Measurement
Tel. +49 6151 803-0
Fax +49 6151 803-9100
info@hbm.com

measure and predict with confidence