ABB MEASUREMENT & ANALYTICS | INTERFACE DESCRIPTION

TTX300

Temperature transmitter

FOUNDATION Fieldbus

Measurement made easy

—

TTX300-FF

Additional Information

Additional documentation on TTX300 is available for download free of charge at www.abb.com/temperature.
Alternatively simply scan this code:

TTF300 **TTH300**

Table of contents

1 Introduction

This manual describes the communication-specific properties of the TTX300-FF transmitter.
General information on operation, sensor configuration, connection, or explosion protection can be found in the operating instructions and commissioning instructions.
The TTX300 transmitter conforms to FF specification FF007-2008.3 and is certified in accordance with ITK 5.10. A device driver in the form of an EDD (Electronic Device Description) is required for commissioning purposes.
The EDD can be downloaded from the Fieldbus Foundation® website and from www.abb.com.
The functional scope of the EDD has been expanded to support additional languages. This means that if it is used in conjunction with older host applications, in particular, you may encounter compatibility problems. For this reason, there is currently a version that conforms to the old standard and one that conforms to the new standard.
You can find out whether the host application supports the new EDD standard by referring to the documentation for the system in question.
It is recommended that you only use certified host systems such as ABB's Industrial IT System 800xA, as these (like the field device being described here) have been tested against the FF standard by an independent body. The parameters described in this documentation are available via the EDD in FF-compliant host applications. The way in which the individual parameters are displayed and arranged, as well as their names, may vary from host application to host application.

2 Block overview

The transmitter contains the following FF blocks:

Number	Block
1	Resource Block
4	AI (analog input) block
1	ePID block (PID controller with expanded features)
2	DI (discrete input) block
1	AO (analog output) block
1	Temperature transducer block
1	Extended diagnostics transducer block
1	HMI transducer block (LCD)

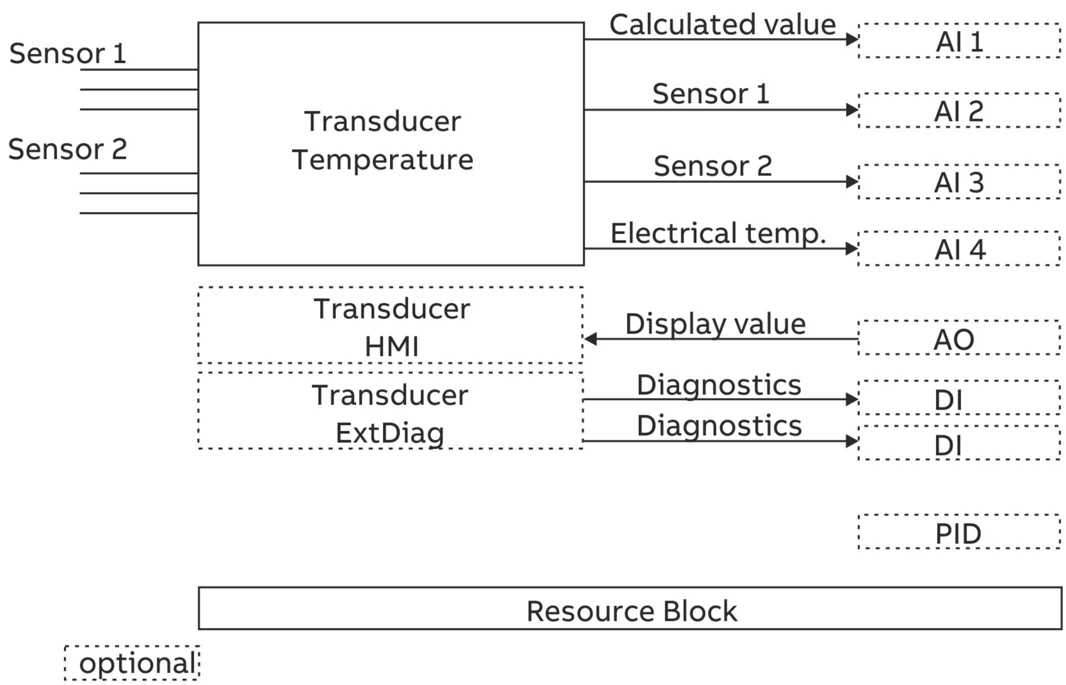

Figure 1: Block structure

AI, DI, and AO are standard FF blocks. 'Resource Block', 'Transducer Temperature', and 'ePID' are extended standard FF blocks. 'Transducer HMI' and 'Transducer ExtDiag' are device/manufacturer-specific blocks.

Resource block

The ‘Resource Block’ contains general information about the fieldbus device, such as the manufacturer, device type, version number, and so on.

Parameter [EN]	Description
Serial Number	Serial number of field device
Assembly Date	Assembly date of field device
Bus Voltage	Fieldbus power supply in volts
Software Version	Firmware version of field device
Hardware Version	Hardware version of field device
Running Hours	Running hours counter
Running Hours at Device Temperature	Running hours at certain electronics temperature classes
Device Temperature	Actual temperature of electronic unit
Min / Max Device Temperature	Absolute minimum and maximum electronics temperature

Analog input block

An AI block performs various tasks, such as rescaling, alarm handling, simulation, and so on.
This is described in detail in FF document FF891.
To make it easier to configure the transmitter, the channel parameter (CHANNEL) is already preset to the relevant channel for the ‘Transducer Temperature’ block:

AI1:	PRIMARY_VALUE_3 = Value calculated from sensor 1 and sensor 2 (differential, average, etc.)
AI2:	PRIMARY_VALUE_1 = Measured value for sensor 1
AI3:	PRIMARY_VALUE_2 = Measured value for sensor 2
AI4:	SECONDARY_VALUE = Temperature of the reference junction or device temperature with internal reference junction

PID block

The PID function block contains a proportional-integral-differential controller, as well as all the components required for scaling, limiting, alarm handling, disturbance variable feedforward control, cascading, and so on. For details, please refer to FF specification FF-891.

… 2 Block overview

… PID block

PID block diagram

The block is structured as shown below:

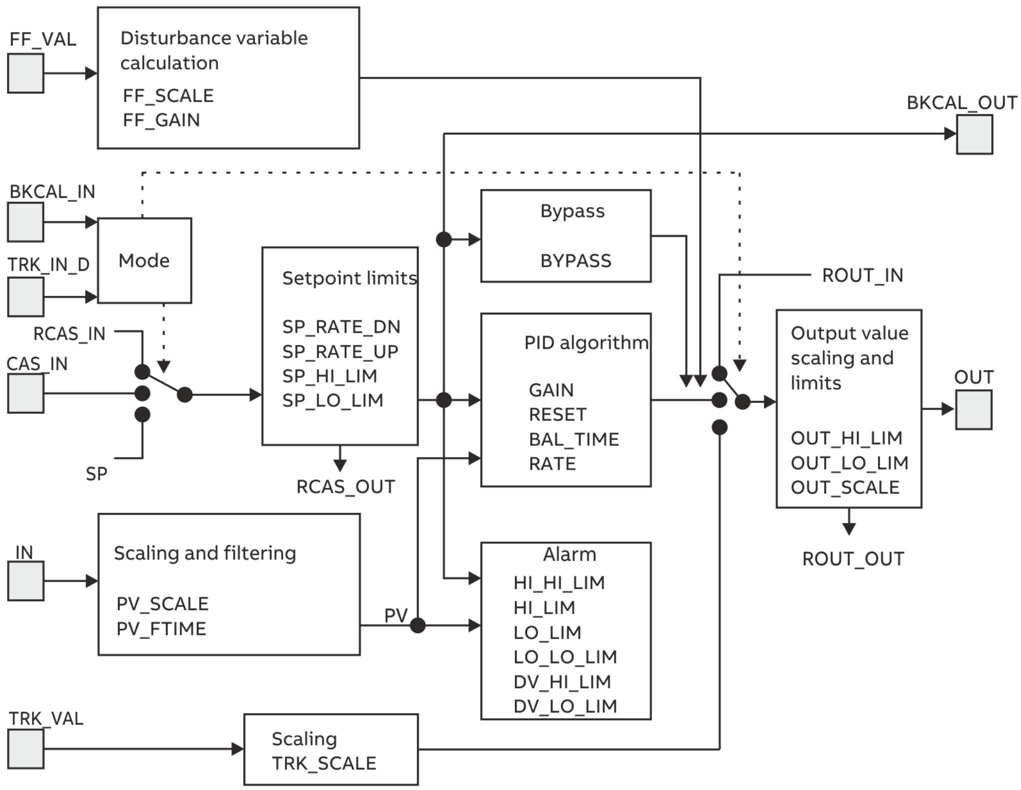

Figure 2: PID block diagram

The controlled variable (actual value) is sent to the IN input. It is scaled using the PV_SCALE parameter and is routed via a filter with the time constant PV_FTIME. The value processed in this way is called the PV (primary analog value).
The mode determines the way in which the setpoint is specified.
In automatic mode (AUTO), the setpoint is specified by the SP parameter. In ‘Cascade’ mode (CAS), the setpoint is specified by the CAS_IN input of a different function block. In ‘Remote Cascade’ mode (RCAS), the setpoint is specified by a control system in the RCAS_IN parameter.
The setpoint range is limited by parameters SP_HI_LIM and SP_LO_LIM, while the maximum rate of change (only applies to AUTO mode) is limited by parameters SP_RATE_DN and SP_RATE_UP. The setpoint limited in this way is called RCAS_OUT and is available for use as a feedback value by control systems (this is necessary for ‘Remote Cascade’ mode).

The PID algorithm is composed of the following:

Proportional component:	The output value (manipulated variable) is proportional to the control deviation (= difference between setpoint and actual value). The proportionality factor is the ‘Gain’ parameter. The drawback of using a purely P controller is its persistent control deviation. An I component can compensate for this, however.
Integral component:	The control deviation is integrated in this. The time constant used here is the ‘Rest’ parameter. The manipulated variable is the value of the integral.
Differential component:	In this case, the control deviation changes are taken as the manipulated variable. The time constant is called the ‘rate’.

The manipulated variable for the PID algorithm is the total of the manipulated variables from all three components.
A bypass is available at the same point as the PID algorithm: This allows the PID algorithm to be bypassed. In this case, the setpoint is immediately taken as the manipulated variable.
A known disturbance variable can be fed forward to input FF_VAL; it is scaled using FF_SCALE and FF_GAIN. The disturbance variable scaled in this way is added to the PID algorithm's manipulated variable.
The manipulated variable is scaled using OUT_SCALE, limited by OUT_LO_LIM and OUT_HI_LIM, and output via OUT.
In AUTO, CAS, and RCAS modes, the value from the PID algorithm (or bypass) is taken as the output value. In ROUT (Remote Out) mode, however, the ROUT_IN value specified by a control system is taken. Tracking is active in LO (Local Overwrite) mode, which means that the tracking value is taken as the output value. The user can set the output value in MAN or OOS mode.
The value for a tracking procedure is specified via the TRK_VAL input and scaled using TRK_SCALE. In order to use the tracking function, 'Track enable' or 'Track in Manual' must be activated in parameter CONTROL_OPTS. Tracking can then be activated using TRK_IN_D. The mode will change to LO (Local Overwrite) when you do this.

Examples of PID block applications

Straightforward control loop, constant setpoint

The flow in a pipeline is to be controlled by a butterfly valve. A permanent setpoint has been specified.

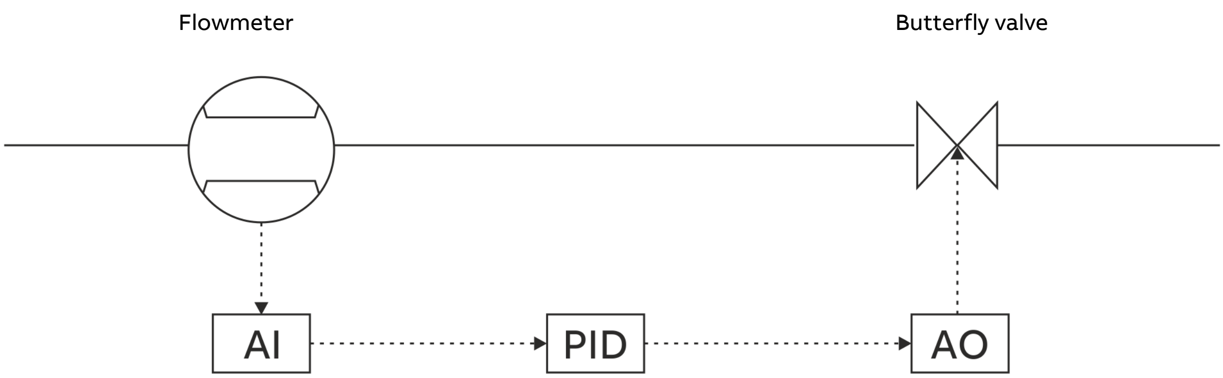

Figure 3: Straightforward control loop, constant setpoint (Example)

The actual value is measured by the flowmeter and made available as an AI block. The setpoint is set in parameter SP in the PID block. The manipulated variable is sent to the AO block of the butterfly valve. It is absolutely essential for a feedback value to be sent from the AO block to the PID block so that the system can switch between modes smoothly. The PID block is in AUTO mode.

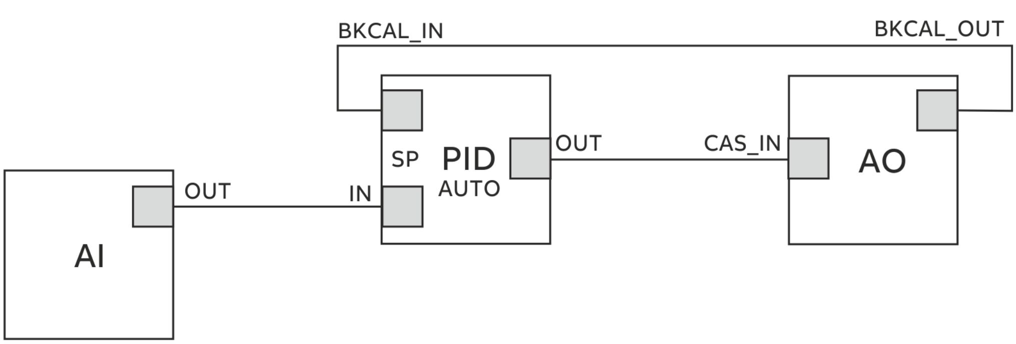

Figure 4: Straightforward control loop, constant setpoint (Example)

… 2 Block overview

… PID block

Straightforward control loop, external setpoint specification
An external setpoint from a different function block (here, AI 1) is sent to the CAS_IN input of the PID block. To enable it to be used, the PID block enters CAS mode.

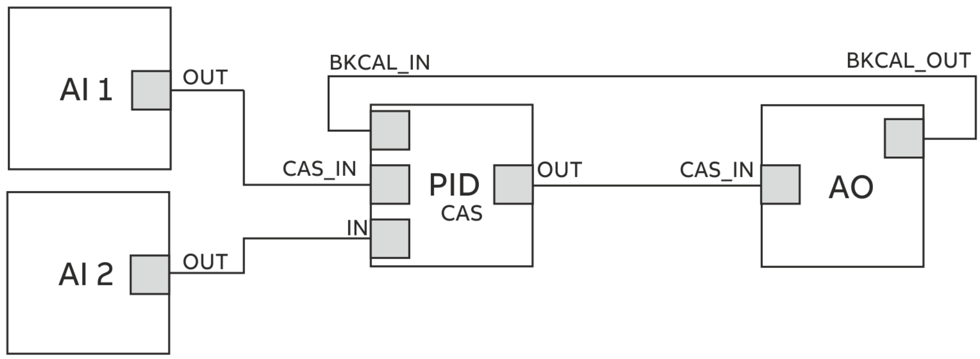

Figure 5: Straightforward control loop, external setpoint specification

Cascaded control loops
PID controllers can be cascaded. This example involves an internal control loop consisting of controller PID2, whose actual value (IN) comes from AI 3, and setpoint CAS_IN, which comes from an external controller (PID1). The external control loop (with controller PID1) receives its setpoint (CAS_IN) from AI 1 and its actual value (IN) from AI 2.
The external PID controller also receives feedback values – from BKCAL_OUT of PID2 in this case – to make it possible to switch between modes smoothly.

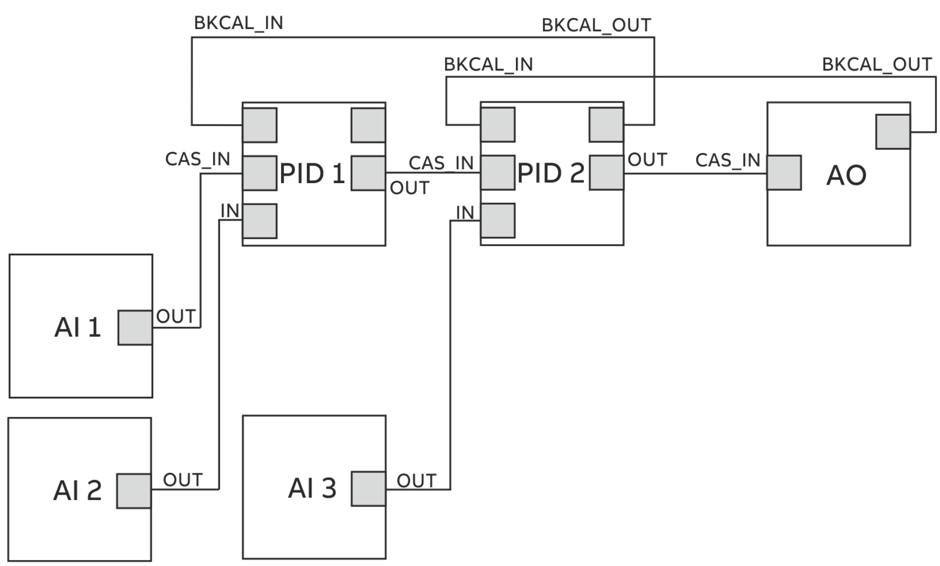

Figure 6: Cascaded control loops

Discrete input block

The discrete input block conforms to FF standard FF891 and is used by the TTX300 for cyclic reading out of extended diagnostics information. Both DI blocks are used in conjunction with the 'Transducer ExtDiag' block. See the section titled **Transducer – extended diagnostics**.

Analog output block

The analog output block conforms to FF standard FF891 and can be used as an option for outputting any cyclic analog value from the network. The value may come from a different field device or even from the host (i.e., from the control system). The TTX300 can be used as a display device for this purpose.

Transducer – temperature

The transducer block contains all the parameters and functions required for measuring and calculating temperature. The values that are measured and calculated are available as transducer block output values, and are called by the function blocks as 'channels'.
It is only possible to read out measured values cyclically from function blocks.

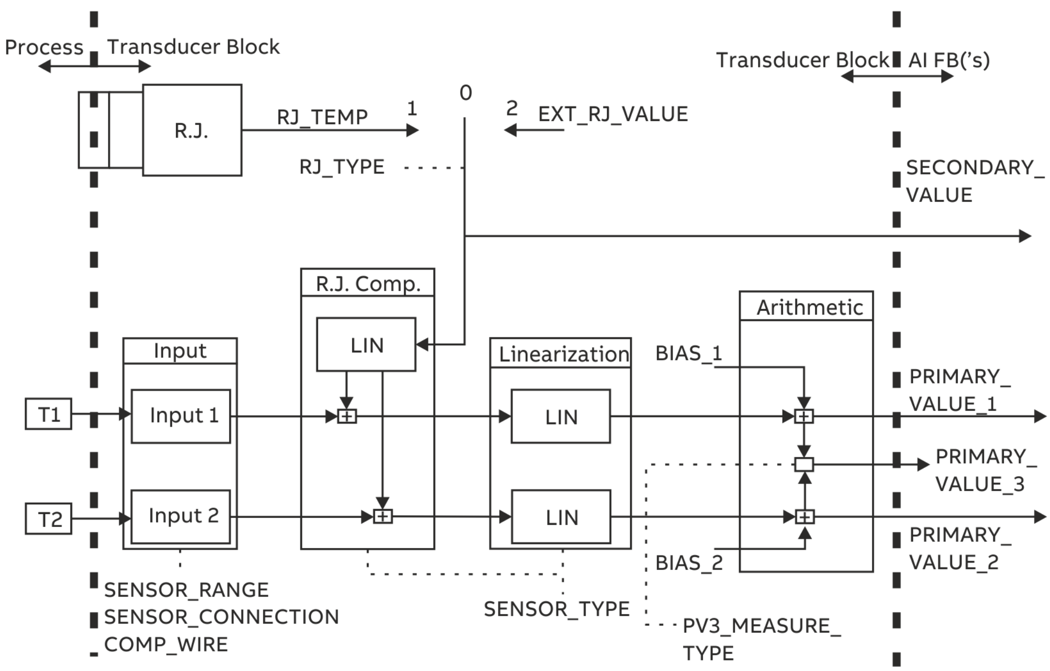

Figure 7: Transducer block

… 2 Block overview

… Transducer – temperature

Parameter [EN]	Beschreibung
PV1 / 2 Type	Identifies measurement type of sensor 1 / 2: Process temperature Non-process temperature Differential temperature
PV1 / 2 Range	Physical measurement range of sensor 1 / 2, depending on sensor type selected
Sensor 1 / 2	Sensor type setting for sensor 1 / 2. All types specified in the datasheet or manual are supported.
Sensor Range 1 / 2	Physical measurement range of sensor. Depends on sensor type selected.
Serial Number Sensor 1 / 2	Optional field for sensor serial number
Cal Method 1 / 2	Optional field for selecting calibration method
Cal Location 1 / 2	Optional field for specifying location of calibration
Cal Date 1 / 2	Optional field for specifying calibration date
Cal Person 1 / 2	Optional field for entering name of person who calibrated the sensor
Connection Sensor 1 / 2	Sensor connection type for RTD (2-, 3-, 4-wire)
SV (Device Temperature)	Displays the device temperature (secondary value)
SV Unit	SV unit (device temperature), always °C
PV3 (calc.value)	Displays the calculated value (PV3)
PV3 Unit	Determines the PV3 unit. Selection depends on selected sensor types 1 and 2.
PV3 Measure Type	PV3 measurement type. Selection depends on selected sensor types 1 and 2: PV1 (sensor 1) PV2 (sensor 2) PV1 (sensor 1) – PV2 (sensor 2) difference PV2 (sensor 2) – PV1 (sensor 1) difference Average Redundancy
Bias (Offset) 1 / 2	Offset to PV1 / 2 (sensor 1 / 2)
Max. Value Sensor 1 / 2	Drag indicator: Maximum value, sensor 1 / 2
Min. Value Sensor 1 / 2	Drag indicator: Minimum value, sensor 1 / 2
Device Temperature (Reference junction Temperature internal)	Temperature of reference junction

Parameter [EN]	**Beschreibung**	
Reference junction Compensation 1 / 2	Reference junction type:	
	not used	No compensation
	Internal	Measured internally (inside transmitter)
	External	Externally stabilized via thermostat
	Sensor 1	Measured via resistance thermometer at channel 1 (can only be set at channel 2)
Temperature fixed CJ 1 / 2	If an externally stabilized reference junction is being used, its temperature is entered here in °C.	
Line Resistance 1 / 2	Line resistance for sensor 1 / 2 if an RTD or linear resistor has been selected as a sensor and a two-wire circuit connection has been selected	
CvD Datensatz 1 / 2	Callendar-Van Dusen dataset 1 / 2. Coefficients R0, A, B, C	
FixPoint 1 / 2	User-specific characteristic with 32 pairs of reference junction each (X1..32, Y1..32), strictly monotonically increasing or decreasing	
Drift Limit	Detection level for drift monitoring between sensor 1 / 2	
Drift Time	Detection time for drift monitoring between sensor 1 / 2	
Drift Detection active	Switch sensor drift monitoring on / off	
Noise Suppression	The transmitter has a characteristic representing noise / interference suppression for the sensor measuring signals. This characteristic can be changed during runtime. The ‘slow’ setting improves the quality of the measurement for noisy measuring signals. The ‘fast’ setting reduces the response time of the transmitter, but requires higher-quality measuring signals. The quality of the measuring signals can be improved by using shielded measuring lines that are as short as possible.	

… 2 Block overview

Transducer – HMI

The ‘Transducer HMI’ block contains all the parameters and functions that are required for configuring the local LCD display. As an option, the display value can be specified using an AO block, via the fieldbus network.

Parameter [EN]	Description
Language	Language used for HMI (LCD display). Language of device driver in host system / configuration tool is not influenced by this setting.
Contrast	Contrast adjustment
Local Operation	Option of blocking local operation
View 1	Selects signal to show in 1-line view
View 2 line 1	Selects signal to show at line 1 in 2-line view
View 2 line 2	Selects signal to show at line 2 in 2-line view
Autoscroll	Activates or deactivates automatic changeover between view 1 (1-line) and 2 (2-line)

Transducer – extended diagnostics

FF devices supply diagnostic information via their ‘Resource Block’. This information can be read out by the device driver (EDD). Normally, it is not possible to access this data in the host from its application, meaning that there is also no way of responding to individual diagnostics events in an application-controlled manner. For example, a particular function could be started in the control system if a display indicating that maintenance is required appears as a result of redundancy switching in the sensor. The transmitter offers two DI (discrete input) blocks for this purpose.

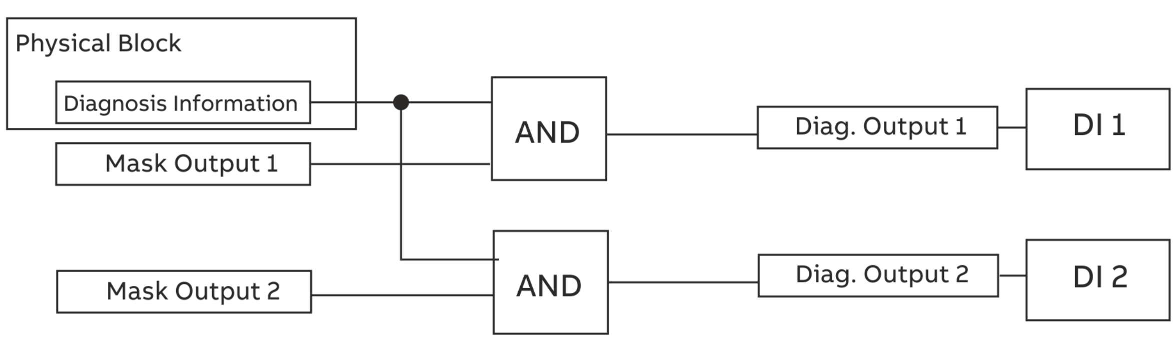

Figure 8: Transducer – extended diagnostics

The behavior of the binary signals can be parameterized in the ‘Transducer ExtDiag’ block using parameters ‘Mask Output 1’ and ‘Mask Output 2’. ANDing is performed bit by bit. The result is TRUE (not equal to 0) if at least one bit operation produces 1 logically; otherwise, it is FALSE (equal to 0). The result is sent on to the connected DI block. Both masks can be set independently of one another. The operation with the second mask provides the value for DI block 2.
Activating a checkbox sets a 1 in the mask.

Parameter [EN]	Description
Output 1 / 2	Displays output channel 1/2 (communicated via DI 1 / 2)
Mask Output 1 / 2	Masking of diagnostic conditions that will lead to a logic 1 signal at the block's output. The output is 1 (true) if at least one of the masked conditions is true. The setting does not influence diagnostics processing itself.

3 Possible commissioning errors

AI block cannot be switched to 'AUTO' mode

The following conditions must be fulfilled in order for an AI block to enter 'AUTO' mode:

- The resource block must be in 'AUTO' mode. There are no other preconditions for this.
- A valid 'Channel' (1 to 8) must be entered in the AI block.
- L_Type must be set to 'Direct' or 'Indirect' (indirect square root is also possible).
- The XD_Scale unit must be the same as the 'Channel Unit'.
- If L_Type is set to 'Direct', the XD_Scale and OUT_Scale structure must have exactly the same settings throughout.

If these conditions have been fulfilled and the 'Target Mode' of the AI block is switched to 'AUTO', the 'Actual Mode' and, therefore, the block itself are also switched to 'AUTO'.
You can see whether these conditions have been fulfilled in the BLOCK_ERR parameter (in the NI Configurator in the AI window, under the 'Diagnostics' tab). If 'Block Configuration Error' is signaled here, one of the conditions has not been fulfilled.
If the PD_Tag of the device or the block tag was adjusted after a 'Schedule' was loaded into the device, it may no longer be possible for blocks to be switched to 'AUTO' even if the conditions listed above have been fulfilled. In this case, a new 'Schedule' containing the 'new' blocks (= new tags = new name) must be created and loaded into the device.

PID block cannot be switched to 'AUTO' mode

The following conditions must be fulfilled in order for the PID block to enter 'AUTO' mode:

- The resource block must be in 'AUTO' mode. There are no other preconditions for this.
- Bypass must have the correct setting (must not be set to the default value 'uninitialized')
- Shed_Opt must have the correct setting (must not be set to the default value 'uninitialized')
- Gain and SP must be set

The actual mode of the PID remains set to 'Iman':
Check the downstream function block from which the BKCAL_IN parameter comes.

Trademarks

FOUNDATION Fieldbus is a registered trademark of FieldComm Group, Austin, Texas, USA.

ABB Limited
Measurement & Analytics
Howard Road, St. Neots
Cambridgeshire, PE19 8EU
UK
Tel: +44 (0)870 600 6122
Fax: +44 (0)1480 213 339
Email: enquiries.mp.uk@gb.abb.com

ABB Inc.
Measurement & Analytics
125 E. County Line Road
Warminster, PA 18974
USA
Tel: +1 215 674 6000
Fax: +1 215 674 7183

ABB Automation Products GmbH
Measurement & Analytics
Schillerstr. 72
32425 Minden
Germany
Tel: +49 571 830-0
Fax: +49 571 830-1806

abb.com/temperature

3KXT200001R4001

COM/TTX300/FF-EN Rev. A 02.2019