optimize! softing

User Guide

Field Device Software

Version: EN-122022-3.60

Disclaimer of liability

Trademarks

OpenSource

To comply with international software licensing terms, we offer the source files of open source software used in our products. For details see https://opensource.softing.com/

If you are interested in our source modifications and sources used, please contact: info@softing.com

Softing Industrial Automation GmbH

Richard-Reitzner-Allee 6
85540 Haar / Germany
https://industrial.softing.com

+ 49 89 4 56 56-340

info.automation@softing.com
support.automation@softing.com

https://industrial.softing.com/support/support-form

Scan the QR code to find the latest documentation on the product web page under Downloads.

Table of Contents

Chapter 1 About this guide 5

1.1 Read me first 5

1.2 Target audience 5

1.3 Typographic conventions 5

1.4 Document feedback 6

1.5 Release history 6

Chapter 2 FF SCP Mode 7

2.1 SCP Configuration 7

2.2 Activating the SCP mode 7

2.3 Deactivating the SCP mode 8

Chapter 3 FF PID Block 9

3.1 Parameter overview 10

3.2 Parameter description 13

3.3 Mode parameter 18

3.3.1 Out of service mode 22

3.3.2 Initialization manual mode 23

3.3.3 Local override mode 23

3.3.4 Manual mode 24

3.3.5 Automatic mode 25

3.3.6 Cascade mode 26

3.3.7 Remote cascade mode 27

3.3.8 Remote output mode 28

3.4 PV filter 29

3.5 Control algorithm 30

3.5.1 Start of the PID control algorithm 30

3.5.2 Stop of the PID control algorithm 31

3.5.3 Out Limits and Anti Windup 31

3.5.4 Feed forward algorithm 31

3.5.5 Status options 32

3.5.6 Shed option 33

3.5.7 Control options 34

3.5.8 Block errors 35

Chapter 4 FF Integrator Block 36

4.1 Block options 36

4.2 Block errors ... 37

Chapter 5 PA Totalizer Block ... 38

5.1 Function block vs transducer block ... 38

5.2 Parameter overview ... 39

5.3 Parameter descriptions ... 41

5.4 Additional block parameter BLOCK_ERR_DESC ... 44

5.5 Totalizer algorithm ... 44

5.6 Supported flow and totalizer units ... 45

5.6.1 Volume flow units ... 45

5.6.2 Mass flow units ... 46

5.6.3 UNIT_TOT units ... 46

5.6.3.1 UNIT_TOT volume units ... 46

5.6.3.2 UNIT_TOT mass units ... 47

Chapter 6 Transducer Blocks ... 48

6.1 Transducer blocks manual mode behavior ... 48

6.2 Softing diagnostic transducer block ... 48

6.2.1 Hardware and protocol stack diagnostic information ... 49

6.2.2 FF only: Function block execution times ... 50

6.3 Softing startHART demo application ... 51

6.3.1 Overview ... 51

6.3.2 Fieldbus representation of the startHART device ... 52

6.3.2.1 FF field device ... 52

6.3.2.2 PA field device ... 53

6.3.3 FF transducer blocks ... 54

6.3.3.1 DEMO_TB parameters ... 54

6.3.3.2 HART_TB parameters ... 56

6.3.3.3 Device specific data structures ... 57

6.3.4 PA transducer blocks ... 58

6.3.4.1 DEMO_TB parameters ... 58

6.3.4.2 HART_TB parameters ... 60

6.3.5 Operation ... 61

6.3.5.1 Process value generation ... 62

6.3.5.2 HART mapping ... 66

Chapter 7 Glossary ... 68

1 About this guide

1.1 Read me first

Please read this guide carefully before using the device to ensure safe and proper use. Softing does not assume any liability for damages due to improper installation or operation of this product. This document is not warranted to be error-free. The information contained in this document is subject to change without prior notice. To obtain the most current version of this guide, visit the download center on our website at: https://industrial.softing.com/en/downloads

1.2 Target audience

This document is addressed to field device developers and software engineers with a good working knowledge of HART/Modbus and FF or PA technology and HART device descriptions.

1.3 Typographic conventions

The following typographic conventions are used throughout Softing customer documentation:

Keys, buttons, menu items, commands and other elements involving user interaction are set in bold font and menu sequences are separated by an arrow	Open **Start** → **Control Panel** → **Programs**
Buttons from the user interface are enclosed in brackets and set to bold typeface	Press **[Start]** to start the application
Coding samples, file extracts and screen output is set in Courier font type	`MaxDlsapAddressSupported=23`
Filenames and directories are written in italic	Device description files are located in *C:\\<Application name>\delivery\software\Device Description files*

CAUTION

CAUTION indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

Note

This symbol is used to call attention to notable information that should be followed during installation, use or servicing of this device.

1.4 Document feedback

We would like to encourage you to provide feedback and comments to help us improve the documentation. You can write your comments and suggestions to the PDF file using the editing tool in Adobe Reader and email your feedback to support.automation@softing.com.

If you prefer to write your feedback directly as an email, please include the following information with your comments:

- document name
- document version (as shown on cover page)
- page number

1.5 Release history

Product version	Modifications compared to previous version
3.40	First release of document with description of FF PID-FB.
3.50	Document renamed to "Field Device Software User Guide". Descriptions of FF SCP mode, FF IT-FB, and PA TOT-FB added.
3.60	Descriptions of demo applications and DIAG-TB added.

2 FF SCP Mode

The FieldCommGroup has released the Standardized Connection Point (SCP) specification for easy configuration of a field device. When SCP mode is activated in the field device the I/O blocks (Analog and Discrete Input and Output Blocks) that are enabled by SCP will be executed with a well-known configuration that permits a host to consume or provide data without further configuration of the device.

The details of the Standardized Connection Points are described in the FieldCommGroup document FCG TS30916 (FF-916). Yet as some SCP aspects can be defined by the device vendor the following sections to this chapter describe which objects are configured automatically, which functions blocks are controlled by SCP mode and how configuration objects are impacted.

2.1 SCP Configuration

The following configuration applies to an activated the SCP mode:

- Link objects, VCRs and Function Block Schedule are cleared
- Every I/O block used in SCP mode is scheduled internally. This is not visible in the FB start entries. The period of the block execution is controlled by the SCP host macrocycle value in the SCP block
- One link object and one VCR are configured for Alarm/Event reporting
- One publisher link object and one publisher VCR are created for an input function block
- One subscriber and one publisher link object as well as one subscriber and one publisher VCR are created for an output function block
- Channels and XD_SCALE units of the SCP I/O blocks are configured
- SCP I/O blocks are set to Auto mode

2.2 Activating the SCP mode

When a device is taken out of the box it typically operates in standard FF mode or in SCP mode. You can specify the initial SCP behavior of your device in the commScript (see commScripter user manual, section FF_SCP_SETTINGS). The initial operation state is also assumed after writing a RESTART to a Resource Block parameter to revert to RestartWithFactory defaults or when the node address or the PD tag of the device is assigned using System Management services SM_CLEAR_ADDRESS/SM_SET_ADDRESS resp. SM_SET_PD_TAG. When the device is using an operational node address again the SCP mode will assume its initial state.

For the Field Device Software from Softing you can control this via the commScript describing the device application behavior. Entry ScpActive in section FF_SCP_SETTINGS can be used for this purpose. During operation, the SCP mode can be activated by writing value 100 (RestartInSCPMode) to the RESTART parameter of the Resource Block. The device will perform an immediate reset and restart in SCP mode.

The current SCP mode is indicated by the actual mode of the SCP block which will be Auto when the SCP mode is active or OutOfService if the SCP mode is deactivated. It is also possible to read the FEATURES_SEL parameter of the Resource Block which uses Bit 14 (0x0002) to indicate that the SCP mode is active.

2.3 Deactivating the SCP mode

In order to maintain compability with host implemenations not supporting SCP configurations the SCP mode is deactivated automatically when a device is configured. This includes writing to the following objects:

- Overwriting link objects or VCR objects created due to the SCP mode
- Creating a link object of type publisher or subscriber in an unused link object
- Creating a VCR of type publisher or subscriber in an unused VCR object
- Writing to an FB start entry object
- Writing RestartWithDefaults (3) to the Resource Block parameter RESTART
- Changing the backup Link Master role by writing to MIB parameter BootOperatFunctionalClass

Note

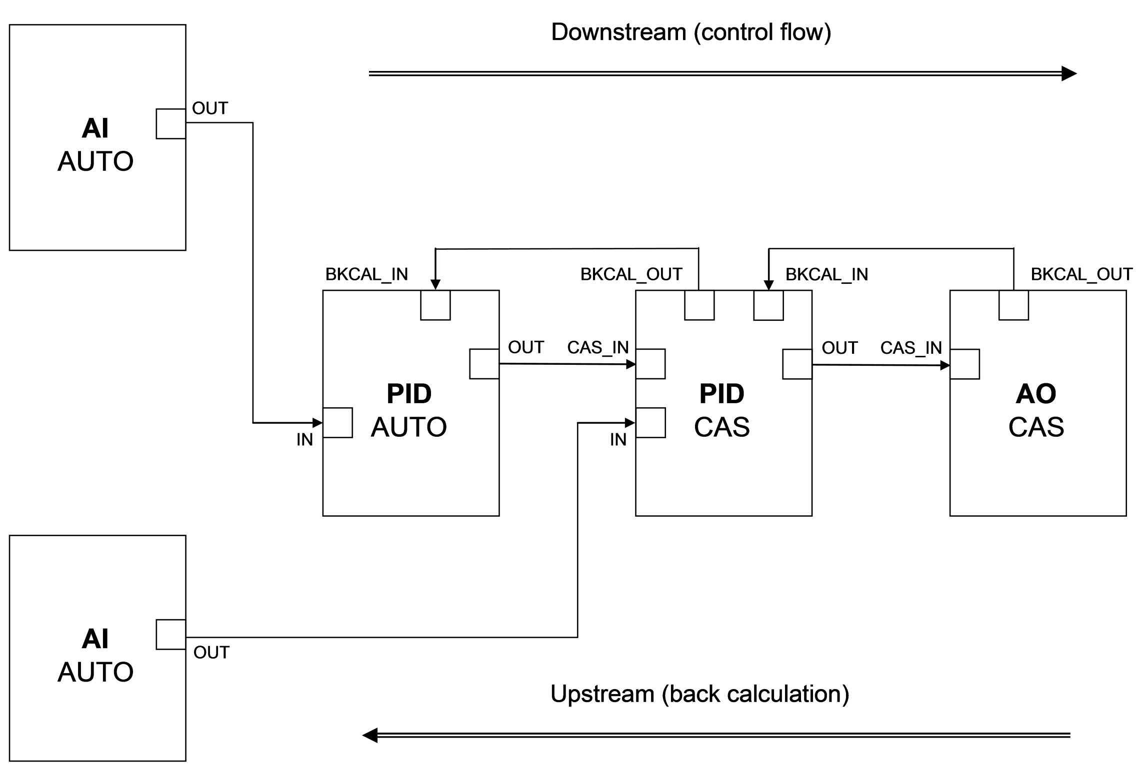
Writing SCP mode bit (Bit 14) in the Resource Block FEATURES_SEL does not impact on the SCP mode behavior. It is not possible to change the SCP mode in this way.

3 FF PID Block

This chapter describes the Proportional Integral Derivative (PID) function block. The PID controller calculation (algorithm) involves three separate parameters; the proportional, the integral and derivative values. The proportional value determines the reaction to the current error, the integral value determines the reaction based on the sum of recent errors, and the derivative value determines the reaction based on the rate at which the error has been changing. The weighted sum of these three actions is used to adjust the process via a control element such as the position of a control valve or the power supply of a heating element.

By tuning the three constants in the PID controller algorithm, the controller can provide control action designed for specific process requirements.

PID in a Control Application



3.1 Parameter overview

Index	Parameter Name	Datatype (Simple Var) Data Structure (Record)	Size	Store	Use / Related Scale or Unit	Valid Range	Initial Value	Write Access *(1)*
0	Block-Header	Block Structure (DS-64)	62	S	Contained			Mixed *(2)*
1	ST_REV	Unsigned16	2	N	Contained		0	Read only
2	TAG_DESC	Octet String	32	S	Contained		Blanks	
3	STRATEGY	Unsigned16	2	S	Contained		0	
4	ALERT_KEY	Unsigned8	1	S	Contained	1 - 255	0	
5	MODE_BLK	Mode Structure (DS-69)	4	Mix	Contained		OOS	Mixed *(3)*
6	BLOCK_ERR	Bit String	2	D	Contained			Read only
7	PV	Float Value & Status (DS-65)	5	D	Contained / PV_SCALE		Bad – non specific / 0	Read only
8	SP	Float Value & Status (DS-65)	5	N	Contained / PV_SCALE	PV_SCALE +/- 10 %	GOOD_CAS / 0	OOS | MAN | AUTO
9	OUT	Float Value & Status (DS-65)	5	N	Output / OUT_SCALE	OUT_SCALE +/- 10 %	BAD – Out of service 0	OOS | MAN
10	PV_SCALE	Scaling Structure (DS-68)	11	S	Contained		0-100%	OOS
11	OUT_SCALE	Scaling Structure (DS-68)	11	S	Contained		0-100%	OOS
12	GRANT_DENY	Access Permissions (DS-70)	2	S	Contained		0,0	
13	CONTROL_OPTS	Bit String	2	S	Contained		0	OOS
14	STATUS_OPTS	Bit String	2	S	Contained		0	OOS
15	IN	Float Value & Status (DS-65)	5	N	Input / PV_SCALE		Bad – Not connected / 0	
16	PV_FTIME	Float	4	S	Contained / seconds	Positive or zero	0	
17	BYPASS	Unsigned8	1	S	Contained / Enumeration	0: Uninitialized 1: Off, 2: On	Uninitialized	OOS | MAN
18	CAS_IN	Float Value & Status (DS-65)	5	D	Input / PV_SCALE		Bad – Not connected / 0	
19	SP_RATE_DN	Float	4	S	Contained / PV units per second	Positive or zero	+INF	
20	SP_RATE_UP	Float	4	S	Contained / PV units per second	Positive or zero	+INF	
21	SP_HI_LIM	Float	4	S	Contained / PV_SCALE	PV_SCALE +/- 10 %	100	
22	SP_LO_LIM	Float	4	S	Contained / PV_SCALE	PV_SCALE +/- 10 %	0	

Index	Parameter Name	Datatype (Simple Var) Data Structure (Record)	Size	Store	Use / Related Scale or Unit	Valid Range	Initial Value	Write Access *(1)*
23	GAIN	Float	4	S	Contained / no unit	Positive or zero	0	
24	RESET	Float	4	S	Contained / second	Positive or zero	+INF	
25	BAL_TIME	Float	4	S	Contained / second	Positive or zero	0	
26	RATE	Float	4	S	Contained / second	Positive or zero	0	
27	BKCAL_IN	Float Value & Status (DS-65)	5	D	Input / OUT_SCALE		Bad – Not connected / 0	
28	OUT_HI_LIM	Float	4	S	Contained / OUT_SCALE	OUT_SCALE +/- 10 %	100	
29	OUT_LO_LIM	Float	4	S	Contained / OUT_SCALE	OUT_SCALE +/- 10 %	0	
30	BKCAL_HYS	Float	4	S	Contained / percent	0 – 50 %	0.5 %	
31	BKCAL_OUT	Float Value & Status (DS-65)	5	D	Output / PV_SCALE		Bad – Non specific / 0	Read only
32	RCAS_IN	Float Value & Status (DS-65)	5	D	Contained / PV_SCALE		Bad – Out of service / 0	
33	ROUT_IN	Float Value & Status (DS-65)	5	D	Contained / OUT_SCALE		Bad – Out of service / 0	
34	SHED_OPT	Unsigned8	1	S	Contained / Enumeration		Uninitialized	
35	RCAS_OUT	Float Value & Status (DS-65)	5	D	Contained / PV_SCALE		Bad – Non specific / 0	Read only
36	ROUT_OUT	Float Value & Status (DS-65)	5	D	Contained / OUT_SCALE		Bad – Non specific / 0	Read only
37	TRK_SCALE	Scaling Structure (DS-68)	11	S	Contained		0-100%	OOS
38	TRK_IN_D	Discrete Value & Status (DS-66)	2	N	Input	0: Off not 0: On	Bad – Not connected / Off	
39	TRK_VAL	Float Value & Status (DS-65)	5	N	Input / TRK_SCALE		Bad – Not connected / 0.0	
40	FF_VAL	Float Value & Status (DS-65)	5	N	Input / FF_SCALE		Bad – Not connected / 0.0	
41	FF_SCALE	Scaling Structure (DS-68)	11	S	Contained		0-100%	OOS
42	FF_GAIN	Float	4	S	Contained		0.0	OOS | MAN
43	UPDATE_EVT	Event Update (DS-73)	14	D	Contained			Mixed *(4)*

Index	Parameter Name	Datatype (Simple Var) Data Structure (Record)	Size	Store	Use / Related Scale or Unit	Valid Range	Initial Value	Write Access *(1)*
44	BLOCK_ALM	Alarm Discrete (DS-72)	13	D	Contained			Mixed *(4)*
45	ALARM_SUM	Alarm Summary (DS-74)	8	Mix	Contained		All Alarms enabled	Mixed *(5)*
46	ACK_OPTION	Bit string	2	S	Contained / Enumeration	0: Auto Ack Disabled 1: Auto Ack Enabled	Auto ack Disabled	
47	ALARM_HYS	Float	4	S	Contained / Percent of PV_SCALE	0 – 50 %	0.5 %	
48	HI_HI_PRI	Unsigned8	1	S	Contained	0 – 15	0	
49	HI_HI_LIM	Float	4	S	Contained / PV_SCALE	PV_SCALE, +INF	+INF	
50	HI_PRI	Unsigned8	1	S	Contained	0 – 15	0	
51	HI_LIM	Float	4	S	Contained / PV_SCALE	PV_SCALE, +INF	+INF	
52	LO_PRI	Unsigned8	1	S	Contained	0 – 15	0	
53	LO_LIM	Float	4	S	Contained / PV_SCALE	PV_SCALE, -INF	-INF	
54	LO_LO_PRI	Unsigned8	1	S	Contained	0 – 15	0	
55	LO_LO_LIM	Float	4	S	Contained / PV_SCALE	PV_SCALE, -INF	-INF	
56	DV_HI_PRI	Unsigned8	1	S	Contained	0 – 15	0	
57	DV_HI_LIM	Float	4	S	Contained / PV_SCALE units	0 to PV span, +INF	+INF	
58	DV_LO_PRI	Unsigned8	1	S	Contained	0 – 15	0	
59	DV_LO_LIM	Float	4	S	Contained / PV_SCALE units	0 to PV span, -INF	-INF	
60	HI_HI_ALM	Alarm Float (DS-71)	16	D	Contained			Mixed *(4)*
61	HI_ALM	Alarm Float (DS-71)	16	D	Contained			Mixed *(4)*
62	LO_ALM	Alarm Float (DS-71)	16	D	Contained			Mixed *(4)*
63	LO_LO_ALM	Alarm Float (DS-71)	16	D	Contained			Mixed *(4)*
64	DV_HI_ALM	Alarm Float (DS-71)	16	D	Contained			Mixed *(4)*
65	DV_LO_ALM	Alarm Float (DS-71)	16	D	Contained			Mixed *(4)*
66	BLOCK_ERR_DESC_1	Bit string	4	D	Contained			

(1) Some of the parameters are not writeable in all target modes. The Write Access column shows the target modes that allow write access to the parameter
(2) The block header allows the elements 'Block Tag', 'Period of Execution' and 'Next FB to Execute' to be written, all other record elements are read-only
(3) The mode parameter the elements 'Target', 'Permitted' and 'Normal' to be written, the element 'Actual' is read-only

(4) Alarm and event parameters allow the element 'Unacknowledged' to be written. All other elements are read-only.
(5) The alarm summary allows the element 'Disabled' to be written. All other elements are read-only

3.2 Parameter description

ST_REV (Static Revision)
This parameter shows the revision level of the static data associated with the function block. To support tracking changes in static parameter attributes, the static revision parameter is incremented each time a static parameter attribute value is changed.

TAG_DESC (Tag Description)
This rarely used parameter allows a user to describe the intended function the block is performing.

STRATEGY (Strategy)
This rarely used parameter allows the user to group blocks by means of a set of group-identifiers. The strategy parameter is not checked or evaluated by the PID.

ALERT_KEY (Alert Key)
This rarely used parameter allows the user to identify different plant units. This information may be used in the host for sorting alarms, etc. In case of a write access the PID checks the valid range of the parameter. The parameter, however, does not influence the behavior of the PID.

MODE_BLK (Mode of Block)
This parameter allows the user to configure the target, permitted and normal modes of the PID and it shows the actual mode of the block.

BLOCK_ERR (Block Error)
This bit-string parameter reflects the error status of the PID (see Block Errors [37]).

PV (Process Value)
This parameter shows the process value to be controlled.

SP (Setpoint)
This parameter contains the target process value. The control algorithm is designed to drive the difference between process value and setpoint to zero.

OUT (Output Value)
The primary output value calculated as a result of the PID functions.

PV_SCALE (PV Scale)
With its high and low scale values this parameter defines the operating range and of the SP and PV parameter. Scale parameters have got an engineering unit. The engineering unit is not evaluated by the PID.

OUT_SCALE (Out Scale)
With its high and low scale values this parameter defines the operating range and of the OUT parameter. Scale parameters have got an engineering unit. The unit is not evaluated by the PID.

GRANT_DENY (Grant and Deny Access Permission)
The grant-deny parameter is used to allow the operator to grant and deny access permission to sets of function block parameters by other devices. The parameter has two elements named Grant and Deny. The PID does not evaluate this parameter; the operation of the parameter depends on the philosophy of the plant.

CONTROL_OPTS (Control Options)
This parameter allows the user to configure the options adapting the calculations done in a control block to the requirements of the controlled process.

STATUS_OPTS (Status Options)
This parameter allows the user to configure the PID behavior in response to different status conditions.

IN (Input value)
The process value to be controlled is subscribed via the IN parameter. The IN value is passed through a low-pass filter whose time constant is PV_FTIME. The filtered value is shown in the PV parameter.

PV_FTIME (PV filter time)
Time constant of the low-pass filter for the PV.

BYPASS (Bypass)
The normal control algorithm may be bypassed through this parameter. When bypass is enabled the setpoint value is re-scaled from PV_SCALE to OUT_SCALE and directly transferred to the OUT parameter.

CAS_IN (Cascade Input Value)
In CAS mode this parameter provides the setpoint value. The parameter subscribes its values from an other function block or from a supervisory host.

SP_RATE_DN (Setpoint Rate Downward)
This parameter allows the user to configure the ramp rate at which downward setpoint changes are acted on in AUTO mode. The unit of this parameter is ‘PV_SCALE units per second’. If the ramp rate is set to zero, then the setpoint won’t be ramped.

SP_RATE_UP (Setpoint Rate Upward)
This parameter allows the user to configure the ramp rate at which upward setpoint changes are acted on in AUTO mode. The unit of this parameter is ‘PV_SCALE units per second’. If the ramp rate is set to zero, then the setpoint won’t be ramped.

SP_HI_LIM (Setpoint High Limit)
Defines the highest setpoint value that can be set by an operator

SP_LO_LIM (Setpoint Low Limit)
Defines the lowest setpoint value that can be set by an operator

GAIN (Proportional Gain)
This dimensionless tuning parameter of the control algorithm is used in the proportional term to amplify (or damp) the deviation between the process value and the working setpoint value RESET (Integral Reset Time) This tuning parameter of the control algorithm is used in the integral term to amplify (or damp) the deviation between the process value and the working setpoint value. Its unit is seconds per repetition.

BAL_TIME (Balance Time)
This tuning parameter is used to avoid integral windup. It specifies the maximum time interval the OUT parameter stays in its limit after the deviation, that drove it into the limit, disappeared.

RATE (Derivative Time)
This tuning parameter of the control algorithm is used in the derivative term to amplify (or damp) the derivation of the deviations between the process value and the working setpoint value.

BKCAL_IN (Back-calculation Input Value)
Via this parameter the PID subscribes the setpoint (or optionally the process value) of its downstream block. Missing subscription or an inadequate parameter status forces the PID into IMAN mode. The parameter value is used for a bumpless cascade initialization.

OUT_HI_LIM (Output High Limit)
High limit of OUT parameter.

OUT_LO_LIM (Output Low Limit)
Low limit of OUT parameter.

BKCAL_HYS (Back-calculation hysteresis)
The amount that the output must change away from its output limit before the limit status is turned off, expressed as a percent of the span of the OUT_SCALE.

BKCAL_OUT (Back-calculation Output Value)
In CAS mode the PID publishes its setpoint (or optionally its process value) via this parameter. An upstream block subscribes status and value via its BKCAL_IN parameter. The upstream block may use its BKCAL_IN parameter to prevent integral windup and to provide a bumpless transfer to into a closed cascade loop.

RCAS_IN (Remote-Cascade Input Value)
In RCAS mode this parameter provides the setpoint value. The parameter has to be written periodically by a supervisory host.

ROUT_IN (Remote-Output Input Value)
In ROUT mode this parameter provides the output value. The parameter has to be written periodically by a supervisory host.

SHED_OPT (Shed Option)
The shed option parameter is an enumerated parameter which is used to configure the desired behavior when mode shedding is necessary. This parameter determines the actual shed mode when the setpoint or output are not updated within a time-out limit in the RCAS or ROUT mode. Also, it determines if the shed mode is maintained once the RCAS_IN in or ROUT_IN parameter is updated after shedding has occurred.

RCAS_OUT (Remote-Cascade Output Value)
In RCAS mode the PID provides its setpoint (or optionally the process value) via this parameter. A supervisory host has to read the parameter. The supervisory host may use status and value to prevent integral windup and to provide a bumpless transfer to into a closed remote-cascade loop. ROUT_OUT (Remote-Output Output Value) In ROUT mode the PID provides its output value via this parameter. A supervisory host has to read the parameter. The supervisory host may use status and value to provide a bumpless transfer to into a closed remote-output loop.

TRK_SCALE (Tracking Scale)
With its high and low scale values this parameter defines the operating range and of the TRK_VAL parameter. Scale parameters have got an engineering unit. The engineering unit is not evaluated by the PID.

TRK_IN_D (Discrete Tracking Input Value)
Via this discrete input parameter the PID subscribes the command to initiate the external tracking of the OUT parameter to the value specified by TRK_VAL.

TRK_VAL (Tracking Value)
Via this input parameter the PID subscribes the track value used when external tracking is enabled by TRK_IN_D.

FF_VAL (Feed-Forward Value)
Via this input parameter the PID subscribes the feed-forward value. The feed-forward value is multiplied by FF_GAIN before it is added to the result of the control algorithm to compensate an external disturbance or load.

FF_SCALE (Tracking Scale)
With its high and low scale values this parameter defines the operating range and of the FF_VAL parameter. Scale parameters have got engineering units. The units are not evaluated by the PID. FF_GAIN (Feed-Forward Gain) The gain that the feed-forward value is multiplied by before it is added to the result of the control algorithm.

UPDATE_EVT (Update Event)
The event update parameter captures the dynamic information associated with a write to a static parameter within the block. The information contained in the event update parameter is transferred by an alert update object when the parameter update is reported to an external host. The event update has a fixed priority of 2.

BLOCK_ALM (Block Alarm)
The block alarm parameter captures the dynamic information associated with the BLOCK_ERR parameter. The information contained in the block alarm parameter is transferred to an alert object when the alarm is reported to an external host. The block alarm has a fixed priority of 2.

ALARM_SUM (Alarm Summary)
This parameter summarizes the status of all PID alarms. For each alarm, the current state, unacknowledged state, unreported state, and disabled state is maintained.

ACK_OPTION (Acknowledge Option)
This parameter allows the user to configure a set of alarms the PID is allowed to acknowledge by itself.

ALARM_HYS (Alarm Hysteresis)
Amount the process value must return within the alarm limits before the alarm condition clears. Alarm hysteresis is expressed as a percent of the PV span.

HI_HI_PRI (High High Alarm Priority)
This parameter allows the user to configure the priority of critically high alarms.

HI_HI_LIM (High High Alarm Limit)
This parameter allows the user to configure the limit which, when reached by the process value, triggers a critically high alarm.

HI_PRI (High Alarm Priority)
This parameter allows the user to configure the priority of advisory high alarms.

HI_LIM (High Alarm Limit)
This parameter allows the user to configure the limit which, when reached by the process value, triggers an advisory high alarm.

LO_PRI (Low Alarm Priority)
This parameter allows the user to configure the priority of advisory low alarms.

LO_LIM (Low Alarm Limit)
This parameter allows the user to configure the limit which, when reached by the process value, triggers an advisory low alarm.

LO_LO_PRI (Low Low Alarm Priority)
This parameter allows the user to configure the priority of critically low alarms.

LO_LO_LIM (Low Low Alarm Limit)
This parameter allows the user to configure the limit which, when reached by the process value, triggers a critically low alarm.

DV_HI_PRI (Deviation High Priority)
This parameter allows the user to configure the priority of deviation high alarms.

DV_HI_LIM (Deviation High Limit)
This parameter allows the user to configure the high limit of the 'PV to SP' deviation which, when reached by the process value, triggers a deviation high alarm.

DV_LO_PRI (Deviation Low Priority)
This parameter allows the user to configure the priority of deviation low alarms.

DV_LO_LIM (Deviation Low Limit)
This parameter allows the user to configure the low limit of the 'PV to SP' deviation which, when reached by the process value, triggers a deviation low alarm.

HI_HI_ALM (High High Alarm)
The high high alarm parameter captures the dynamic information associated with critically high alarms. The information contained in the high high alarm parameter is transferred to an alert object when the alarm is reported to an external host.

HI_ALM (High Alarm)
The high alarm parameter captures the dynamic information associated with advisory high alarms. The information contained in the high alarm parameter is transferred to an alert object when the alarm is reported to an external host.

LO_ALM (Low Alarm)
The low alarm parameter captures the dynamic information associated with advisory low alarms. The information contained in the low alarm parameter is transferred to an alert object when the alarm is reported to an external host.

LO_LO_ALM (Low Low Alarm)
The low low alarm parameter captures the dynamic information associated with critically low alarms. The information contained in the low low alarm parameter is transferred to an alert object when the alarm is reported to an external host.

DV_HI_ALM (Deviation High Alarm)
The deviation high alarm parameter captures the dynamic information associated with deviation high alarms. The information contained in the deviation high alarm parameter is transferred to an alert object when the alarm is reported to an external host.

BLOCK_ERR_DESC_1 (Block Error Description)
The block error description is used to report more specific details regarding configuration errors that are reported through BLOCK_ERR.

3.3 Mode parameter

The PID supports the target modes

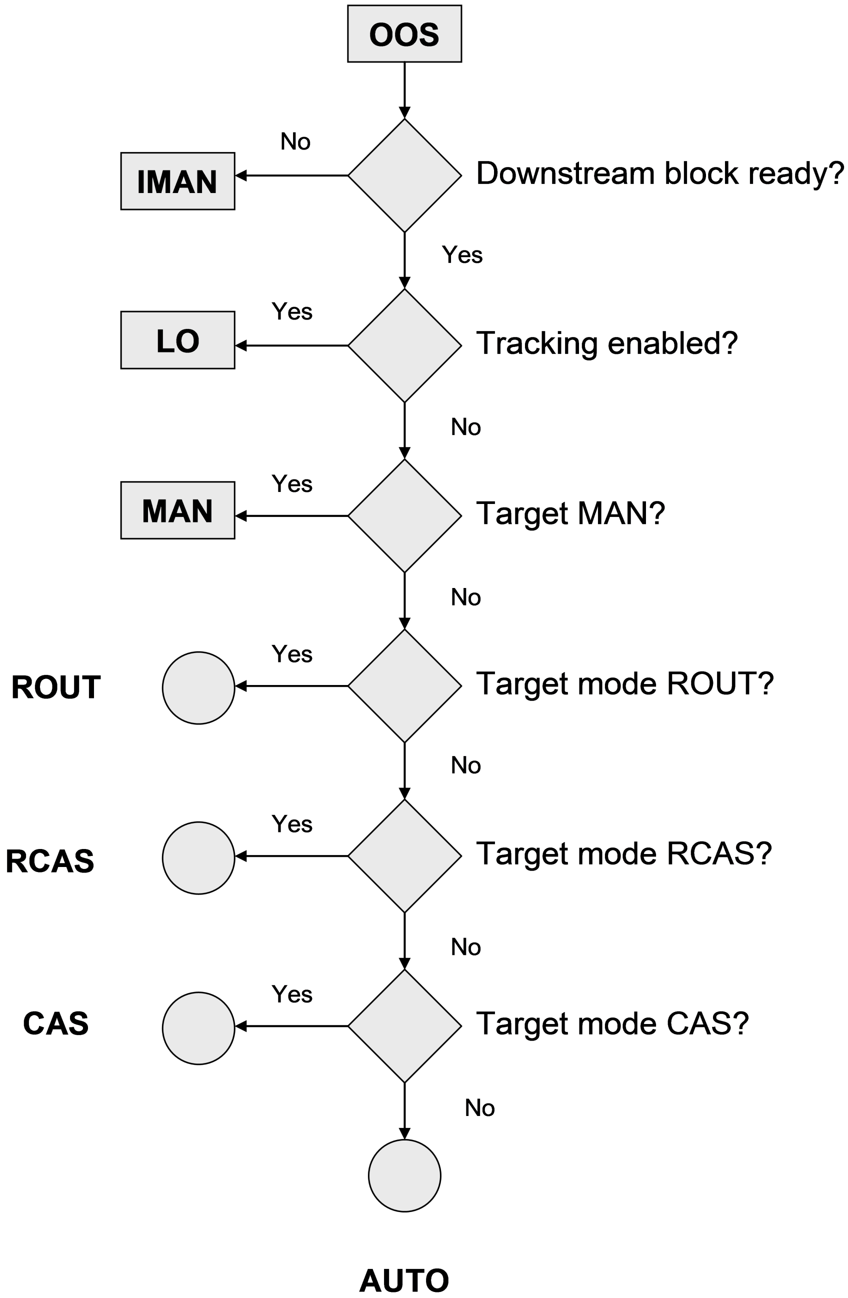
OOS	Out of Service
MAN	Manual
AUTO	Automatic
CAS	Cascade
RCAS	Remote Cascade
ROUT	Remote Output

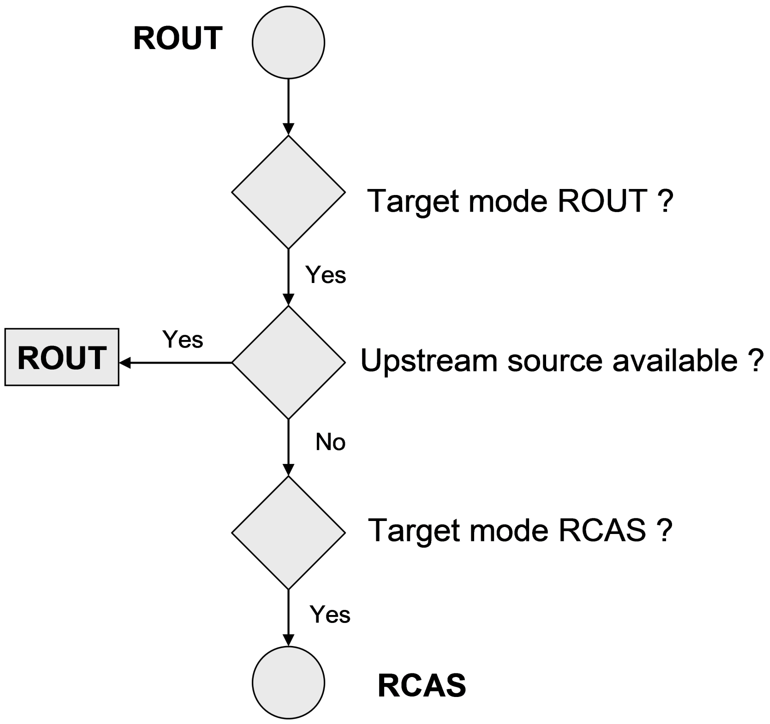
Additional to these target modes the PID supports an initialization and a tracking mode

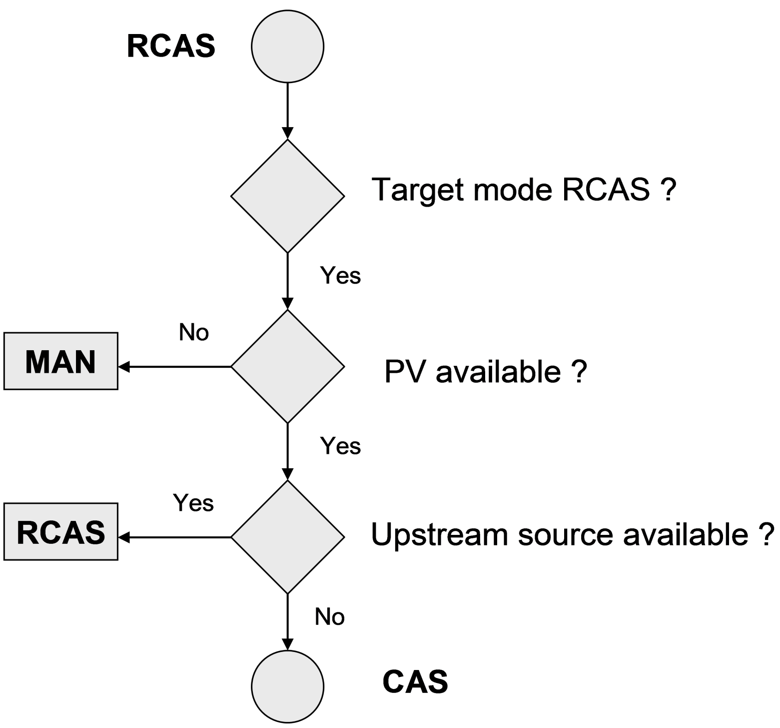
IMAN	Initialization Manual	(initialization mode)
LO	Local Override	(tracking mode)

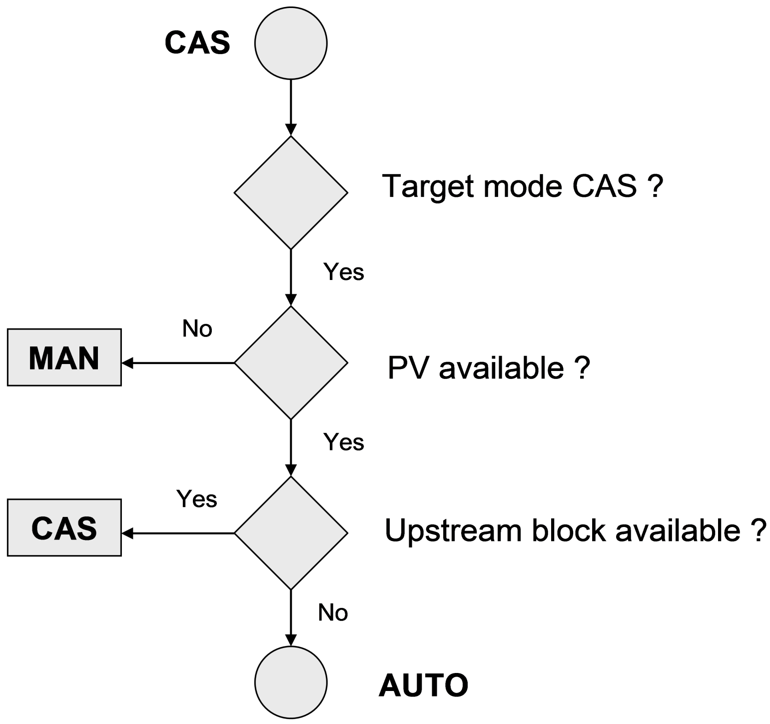
IMAN and LO are no selectable target modes. The PID switches to IMAN mode whenever its downstream block is not available. LO mode can be enabled via tracking option and TRD_IN_D parameter. Though it is allowed to select multiple target modes (ROUT and RCAS and MAN etc.) it is good practice to select one of the available target modes (ROUT or RCAS, AUTO or MAN etc.). Target mode is a bit-enumerated parameter. For the target modes ROUT, RCAS and CAS the bit of the selected mode and the bit of AUTO mode has to be set.

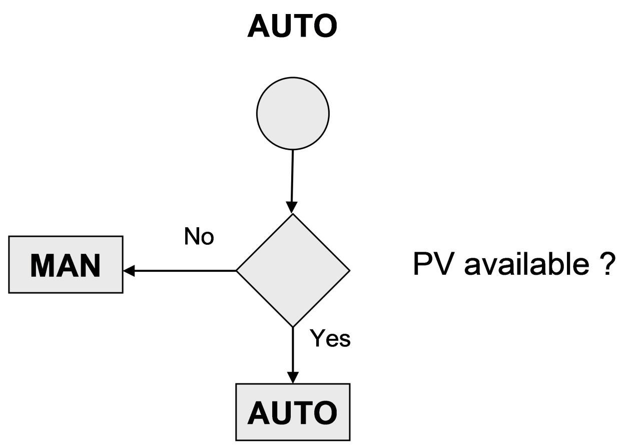
Calculation of Actual Mode



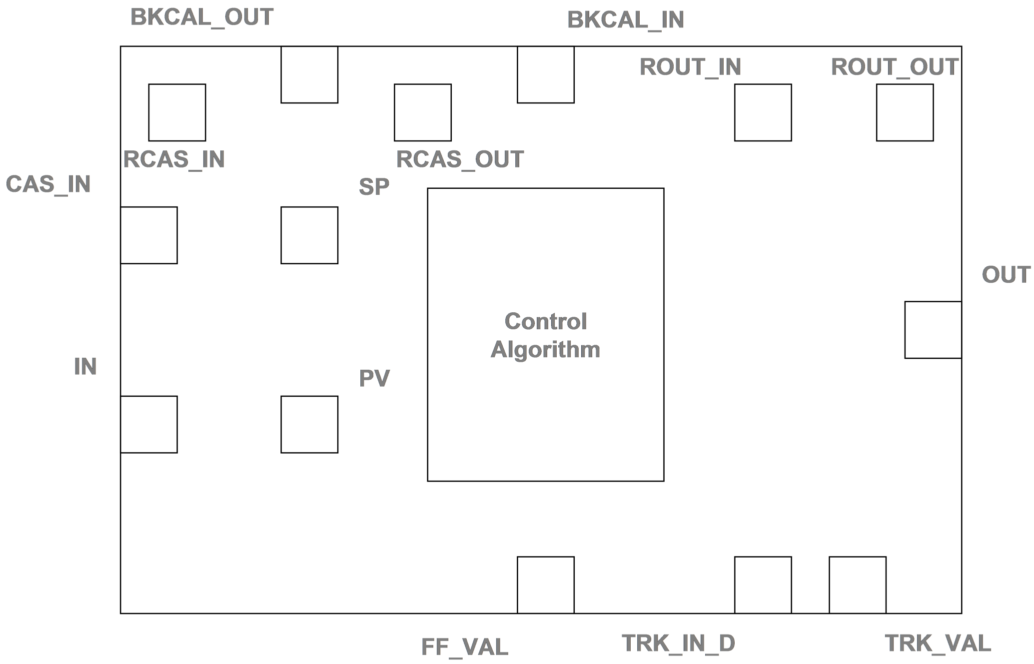

ROUT
Target mode ROUT ?
Yes
Yes
ROUT
Upstream source available ?
No
Target mode RCAS ?
Yes
RCAS


RCAS
Target mode RCAS ?
Yes
No
MAN
PV available ?
Yes
Yes
RCAS
Upstream source available ?
No
CAS


CAS
Target mode CAS ?
Yes
No
MAN
PV available ?
Yes
Yes
CAS
Upstream block available ?
No
AUTO


AUTO
No
MAN
PV available ?
Yes
AUTO

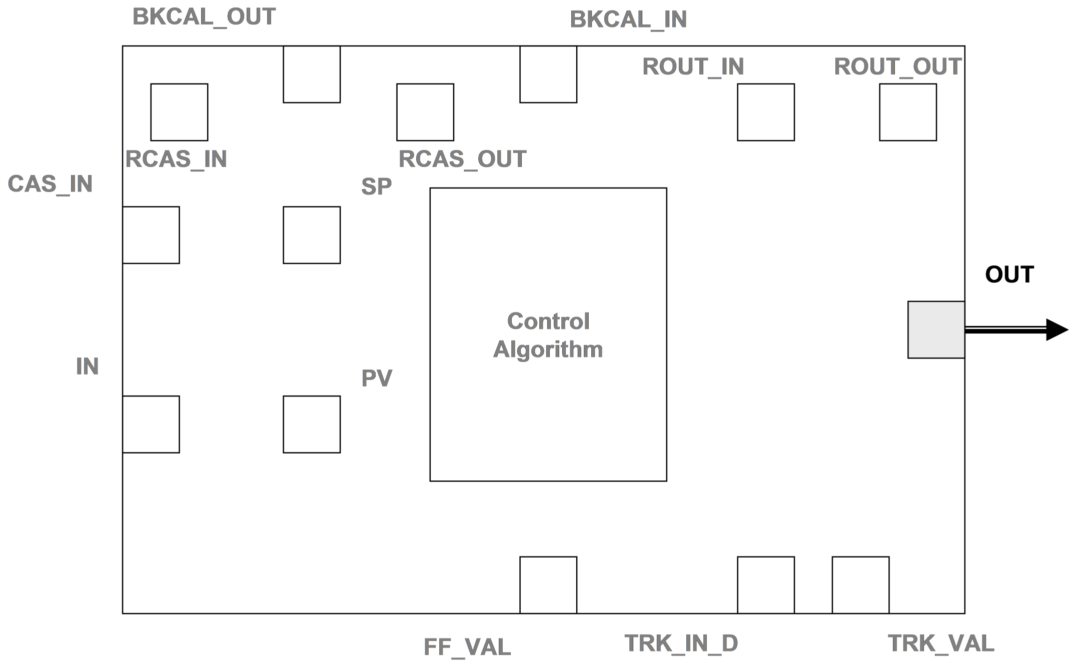
3.3.1 Out of service mode



The PID takes the OOS mode in the following instances:

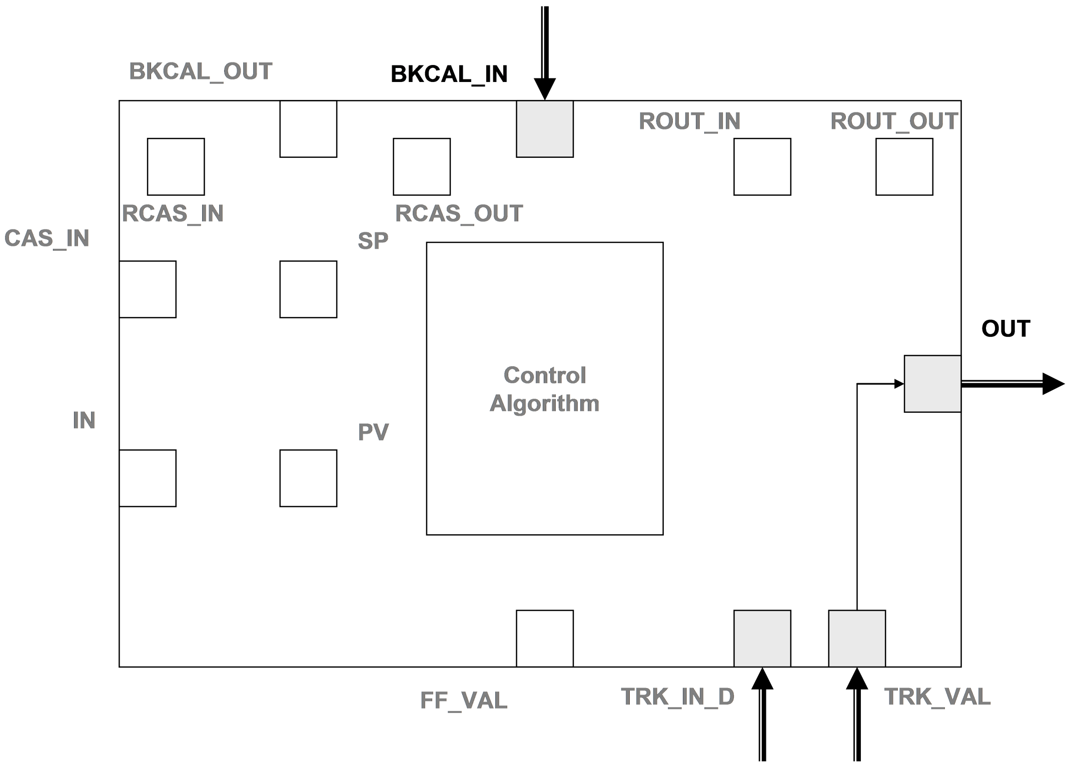
- The target mode is OOS
- The resource block is in OOS
- There is a PID configuration error
- There is no FB start object that triggers the PID execution
- The function block schedule is not active

3.3.2 Initialization manual mode



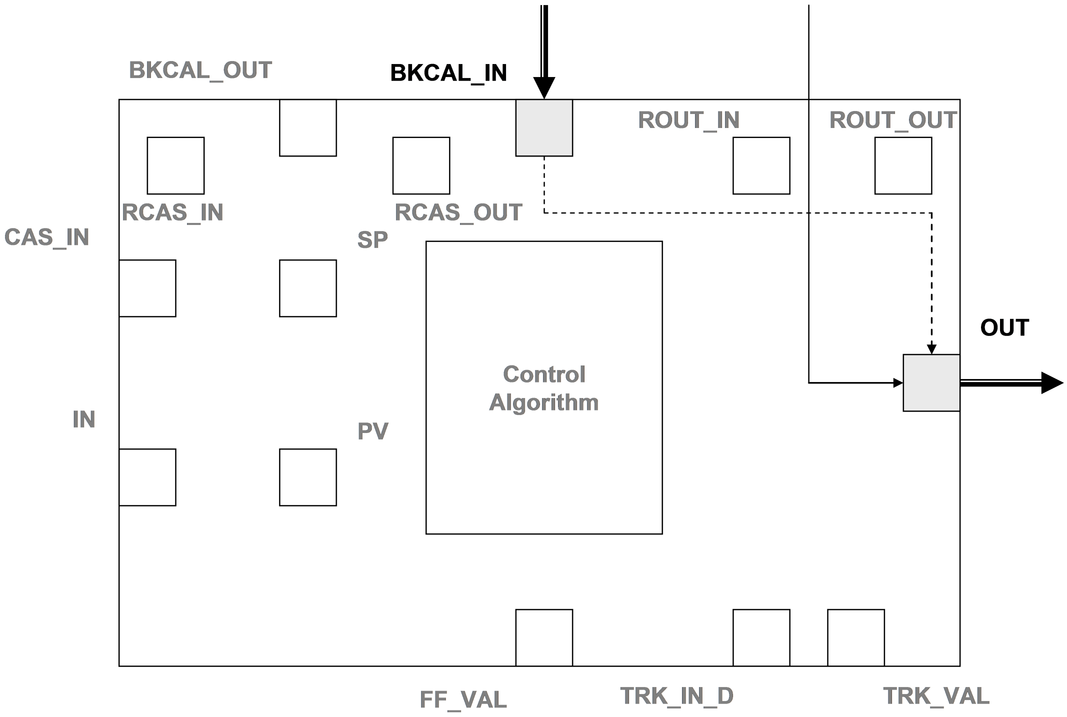
The PID takes the IMAN mode if its target mode is not OOS and the downstream block is not available.

3.3.3 Local override mode



The PID takes the LO mode if its target mode is not OOS and tracking is enabled.

3.3.4 Manual mode

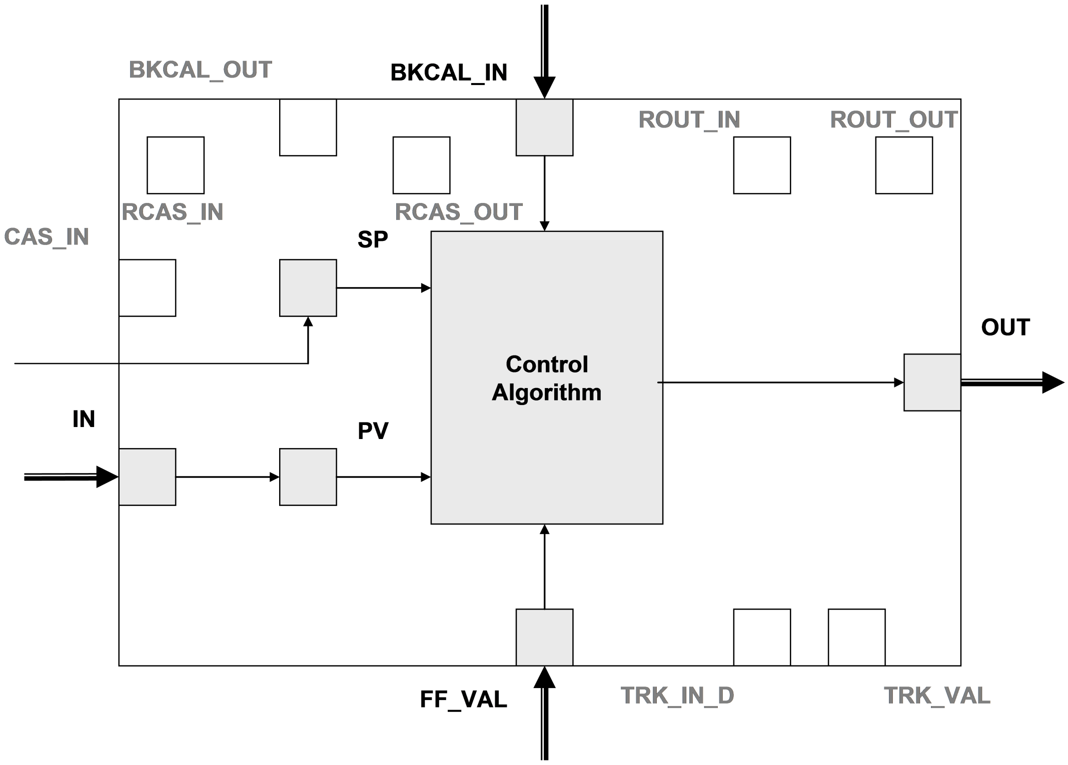


The PID takes the MAN mode in the following instances:

- The target mode is MAN.
- The target mode is AUTO, CAS or RCAS and the status of the IN parameter is BAD.

When the PID switches from IMAN to MAN it initializes the OUT parameter with the value received via BKCAL_IN.

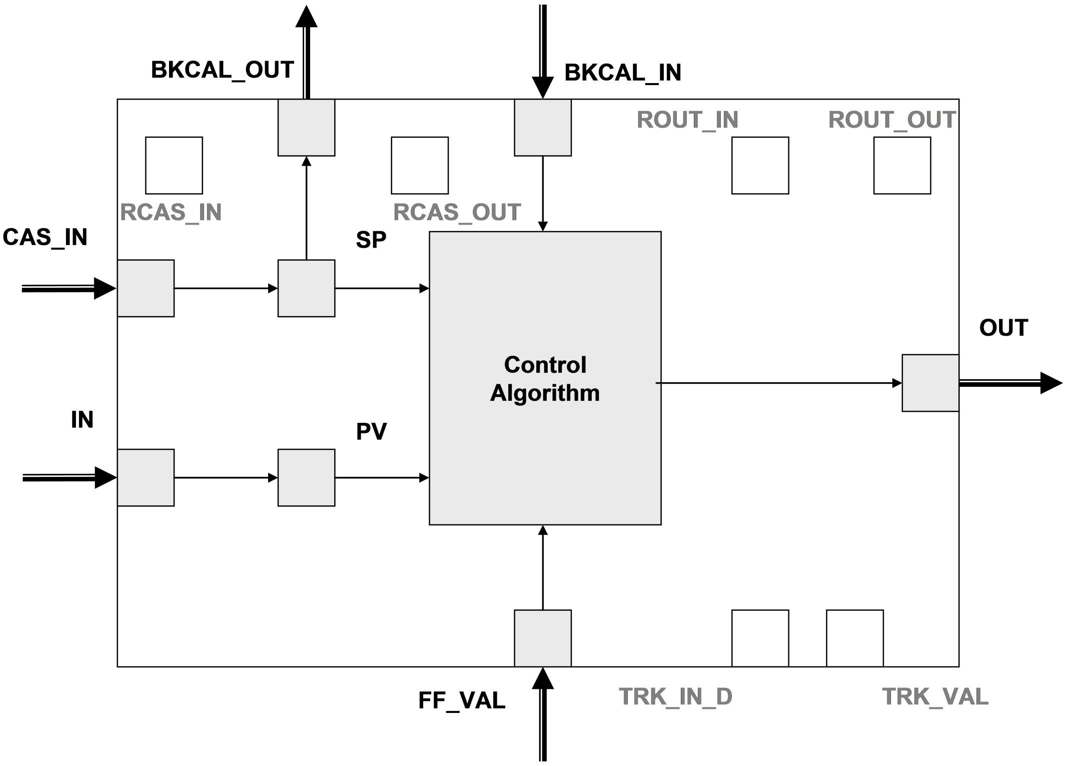
3.3.5 Automatic mode



The PID takes the AUTO mode in the following instances:

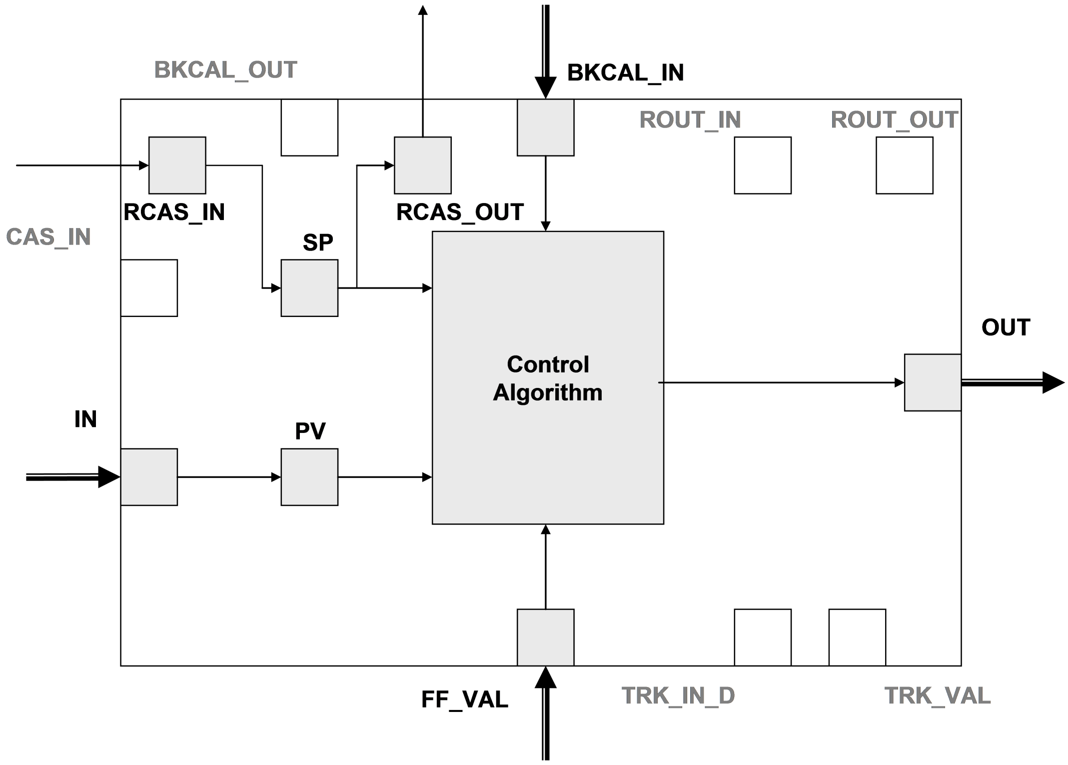
- The target mode is AUTO.
- The target mode is CAS, RCAS or ROUT and the upstream block (CAS) or supervisory host (RCAS, ROUT) is not available.

3.3.6 Cascade mode



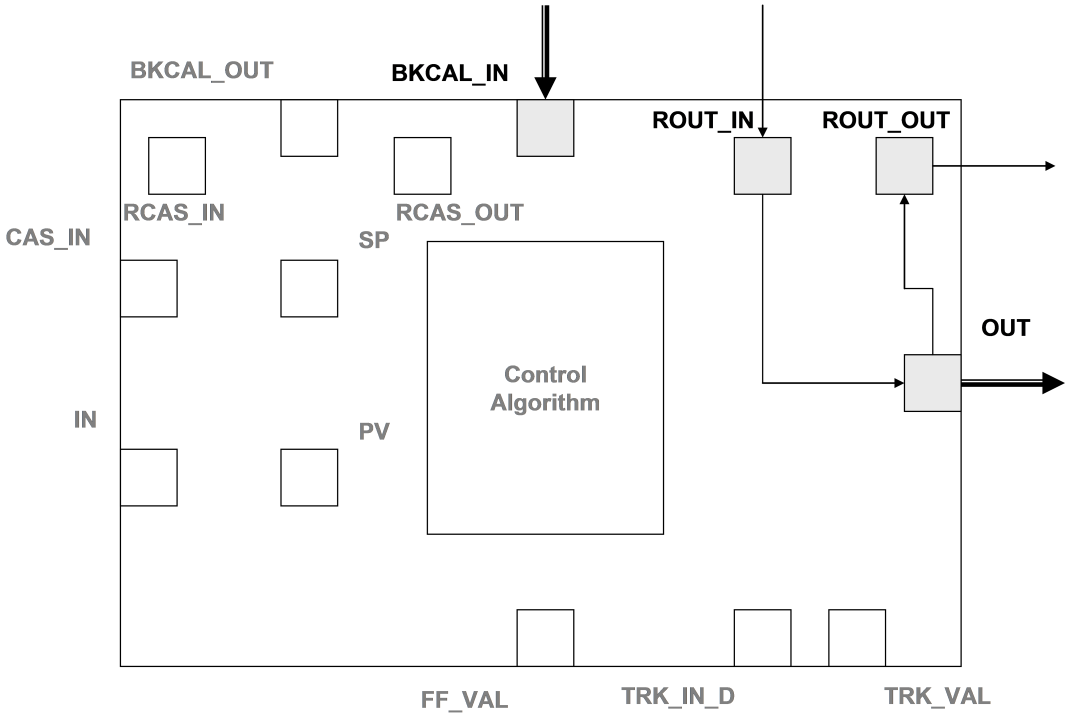
The PID takes the CAS mode if its target mode is AUTO | CAS and the upstream block is available.

3.3.7 Remote cascade mode



The PID takes the RCAS mode if its target mode is AUTO | RCAS and the supervisory host is available.

3.3.8 Remote output mode



The PID takes the ROUT mode if its target mode is AUTO | ROUT and the supervisory host is available.

3.4 PV filter

For some applications it may be necessary to filter out noise on the subscribed input value. For this reason the PID implements a low-pass filter.

Where:

`Period_of_exec`	Period of PID execution [unit sec]
`PV_FTIME`	PV filter time [unit sec] as configured by the user
`PV[n]`	Value of the PV parameter after the nth execution of the PID
`IN[n]`	Value of the IN parameter subscribed before the nth execution of the PID

The filter algorithm is:

```
pv_factor = Period_of_exec / (Period_of_exec + PV_FTIME);
PV[n]     = pv_factor * IN[n] + (1 – pv_factor) * PV[n-1];
```

If IN jumps from a value ***x*** to a new steady-state value y then the PV value will hold at about ***x + 0.63 * (y – x)*** after PV_FTIME, and it will hold at about ***x + 0.99 * (y – x)*** after 5 * PV_FTIME.

If the PV_FTIME is set to zero the IN value is directly passed to the PV parameter.

3.5 Control algorithm

3.5.1 Start of the PID control algorithm

The PID tries to initialize its control algorithm when its target mode is set to AUTO, CAS or RCAS.

The PID control algorithm requires a valid working setpoint, a valid process value (PV) and a valid back-calculation input value (BKCAL_IN). If one of these essential inputs is not available the PID does not initialize its control algorithm and it does not switch to its target mode.

The PID checks its inputs in the following sequence:

- If there is no valid BKCAL_IN value the PID will switch to IMAN mode
- If there is a valid BKCAL_IN value but not a valid PV value the PID will switch to MAN mode
- If there are valid BKCAL_IN and PV values but not a valid SP the PID will stay in AUTO mode

Valid back-calculation input value (BKCAL_IN)
A cascade loop needs two linkages between the PID and its downstream block. One link connects the OUT parameter of the PID with the CAS_IN parameter of its downstream block. The other link connects the BKCAL_OUT parameter of the downstream block with the BKCAL_IN parameter of the PID.

The cascade loop is initialized by the downstream block. If the downstream block is ready for cascade loop control it will send a “Good-Cascade | Initialization-Request” status to the BKCAL_IN parameter of the PID. If the PID is prepared to initialize the cascade loop it will send a “Good-Cascade | Initialization-Acknowledge” status to the CAS_IN parameter of the downstream block. With the reception of the acknowledge status the cascade control loop is initialized. When the cascade loop is established the downstream block sends a “Good-Cascade | Non-specific” status to the PID block. As long as it receives this status the PID takes the back-calculation input value as valid.

In addition to this classic cascade initialization the PID accepts a “Good Cascade | Non-specific” status as valid even if there was no preceding “Good Cascade | Initialization-Request”

The PID also accepts “Good Non-Cascade” as a valid back-calculation input status. When the “Good Non-Cascade” status is used a cascade loop initialization is not required.

Valid Process Value (PV)
The PV value is valid if its status is “Good Non-Cascade”.

The quality of the PV status is calculated from the status of the IN parameter. The IN parameter usually receives “Good Non-Cascade” status. A “Good Non-Cascade” status is propagated directly to the PV status. If the IN status is “Good Cascade” it will be propagated as “Good Non-Cascade” status to the PV. If the IN status is “Uncertain” it depends on the status option “Use Uncertain as Good” whether the PV becomes “Good Non-Cascade” or “Bad”.

Valid Setpoint (SP)
The SP value is valid if its status is “Good Cascade”.

3.5.2 Stop of the PID control algorithm

The PID control algorithm will stop if the BKCAL_IN or PV status becomes invalid.

3.5.3 Out Limits and Anti Windup

The Control_out value calculated by the control algorithm is limited by the OUT_HI_LIM and OUT_LO_LIM parameters.

If there is a long-time deviation between setpoint and PV, the integral term will be incremented or decremented with each PID execution and the Control_out value will run far beyond the OUT_HI_LIM or OUT_LO_LIM.

The BAL_TIME parameter is used for anti windup handling. BAL_TIME is used to specify how long the OUT parameter stays in its limit. The balance timer starts when the deviation disappears that causes the Control Value to exceed the out limit. The Integral term is balanced so that the Control_out value moves back into the unlimited OUT span no later than BAL_TIME.

When BAL_TIME parameter is zero, the OUT value will leave its limit immediately after the deviation disappears, which caused the Control_out value to exceed the out limit.

3.5.4 Feed forward algorithm

The PID supports a feed forward algorithm. The FF_VAL input brings in an external value which is proportional to some disturbance in the control loop. The value is converted to percent of output span using the values of parameter FF_SCALE. This value is multiplied by the FF_GAIN and added to the target output of the control algorithm. If the status of FF_VAL is BAD, the last usable value will be used to prevent the bumping of the output. When the status returns to GOOD, the block will adjust its integral term to maintain the previous output.

`FF_VAL[n]`	Value of the FF_VAL parameter subscribed prior to n_{th} execution of the PID
`FF_Value[n]`	Feed-forward value calculated during the nth execution of the PID

The feed-forward term is calculated as follows:

```
FF_Value[n] = FF_GAIN *(FF_VAL[n] – FF_VAL[n-1]);
```

If `FF_VAL[n]` or `FF_VAL[n-1]` is not available then `FF_Value[n]` is set to zero.
`FF_Value[n]` is rescaled and added to the integral term of the control algorithm.

3.5.5 Status options

Status options allow the user to configure the PID behavior in response to different status conditions.

IFS if BAD IN
When this option is enabled the PID will set *Initiate Fault State* status in its OUT parameter if the status of the IN parameter is BAD.

IFS if BAD CAS_IN
When this option is enabled the PID will set *Initiate Fault State* status in its OUT parameter if the status of the CAS_IN parameter is BAD.

Use Uncertain as Good
When this option is enabled the PID will treat an UNCERTAIN status of the IN parameter as GOOD. Otherwise, it will treat an UNCERTAIN status as BAD.

Target to Manual if BAD IN
When this option is enabled the PID will set its target mode to MAN if the status of the IN parameter is BAD. By means of this option the PID is latched to MAN mode if the IN parameter ever goes BAD.

Target to Next Permitted Mode if BAD CAS_IN
When this option is enabled the PID will set its target mode to the next permitted mode if the target mode is CAS and the status of CAS_IN parameter is BAD. By means of this option the PID is latched to the next permitted mode if the CAS_IN parameter is being used and the status ever goes BAD.

Target to MAN if BAD TRK_IN_D
When this option is enabled the PID will set its target mode to MAN if tracking is active and the status of the TRK_IN_D parameter is BAD. By means of this option the PID is latched to MAN if tracking is active and the status of the TRK_IN_D parameter ever goes BAD.

IFS if BAD TRK_IN_D
When this option is enabled the PID will set Initiate *Fault State Status* in its OUT parameter if tracking is active and the status of the TRK_IN_D parameter is BAD.

3.5.6 Shed option

The shed option applies to the modes RCAS and ROUT. These modes require a supervisory host to update the RCAS_IN or ROUT_IN parameter periodically. The resource block parameters SHED_RCAS and SHED_ROUT determine the time-limit for two consecutive parameter updates.

The SHED_OPT parameter determines the actual shed mode when the RCAS_IN or ROUT_IN parameter is not updated within the configured time-out limit. Also, it determines if the shed mode is maintained once the RCAS_IN or ROUT_IN parameter is updated after shedding has occurred.

SHED_OPT has the following enumerations:

Uninitialized

Normal shed, normal return - Actual mode changes to the next lowest priority non-remote mode permitted but returns to the target remote mode when the supervisory host completes the initialization handshake.

Normal shed, no return - Target mode changes to the next lowest priority non-remote mode permitted. The target remote mode is lost, so there is no return to it.

Shed to Auto, normal return

Shed to Auto, no return - Target mode changes to AUTO on detection of a shed condition.

Shed to Manual, normal return

Shed to Manual, no return - Target mode changes to MAN on detection of a shed condition.

Shed to Retained target, normal return*

Shed to Retained target, no return - Target mode changes to retained target on detection of a shed condition.

** Retained target: If the PID normally runs in CAS mode but if it currently set to RCAS mode then the host may store the normal target CAS. In this case the target mode shows RCAS | CAS | AUTO where CAS is the 'retained target' mode.*

3.5.7 Control options

Bypassing of control algorithm
When bypass is active the setpoint is passed to the output, bypassing the control algorithm. Bypassing is enabled by setting the BYPASS parameter to *On*. The bypassing option has to be enabled by setting the control option *Enable Bypass*.

Setpoint tracking
Setpoint tracking means the setpoint (SP) tracking the process value (PV) in manual-like modes that do not execute the control algorithm. Setpoint tracking keeps the difference between SP and PV at zero and when the control algorithm starts again there will be no bump in the PV – SP deviation. In dependence on the controlled process setpoint tracking can be useful for the target modes CAS and RCAS.

SP-PV Track in MAN
Permits the setpoint to track the process variable when the target mode of the block is MAN.

SP-PV Track in ROUT
Permits the setpoint to track the process variable when the actual mode of the block is ROUT.

SP-PV Track in LO or IMAN
Permits the setpoint to track the process variable when the actual mode of the block is LO or IMAN.

SP Track retained target
Permits the setpoint to track the RCAS or CAS parameter based on the retained target mode when the actual mode of the block is IMAN, LO, MAN, or ROUT. If SP-PV track options are enabled, then SP Track retained target will have precedence in the selection of the value to track when the actual mode is MAN, IMAN, ROUT and LO.

Direct acting
This option defines the relationship between a change in PV and the corresponding change of the output. When *Direct Acting* is selected, an increase in PV results in an increase in the output.

Tracking mode
In certain cases it may be necessary to drive the PID output to a safety or final value. To override the calculated output value the tracking mode has to be enabled.

Track Enable
This option enables the external tracking function. If enabled, the value in TRK_VAL will replace the value of OUT if TRK_IN_D becomes true and the target mode is not MAN.

Track in Manual
This option enables TRK_VAL to replace the value of OUT when the target mode is MAN and TRK_IN_D is true. The actual mode will then be LO.

Track if Bad TRK_IN_D
If tracking is enabled and the status of TRK_IN_D is BAD, then the TRK_VAL will replace the value of the OUT parameter.

Further control options

Use PV for BKCAL_OUT
The BKCAL_OUT and RCAS_OUT values are normally the working setpoint. If this option is enabled, then the PV value will be used.

Obey SP limits if CAS or RCAS
Normally the setpoint will not be restricted to the setpoint limits only if the target mode is AUTO. However, if this option is selected, the setpoint will be restricted to the setpoint absolute limits in the CAS and RCAS modes too.

No OUT limits in Manual
Do not apply OUT_HI_LIM or OUT_LO_LIM when target mode is MAN. Trust the operator to do the right thing.

3.5.8 Block errors

The PID has two error conditions:

Out-of-Service
The PID shows the Out-of-Service error if its actual mode is OOS.

Block Configuration Error
The PID shows a block configuration error if at least one of the following conditions is valid:

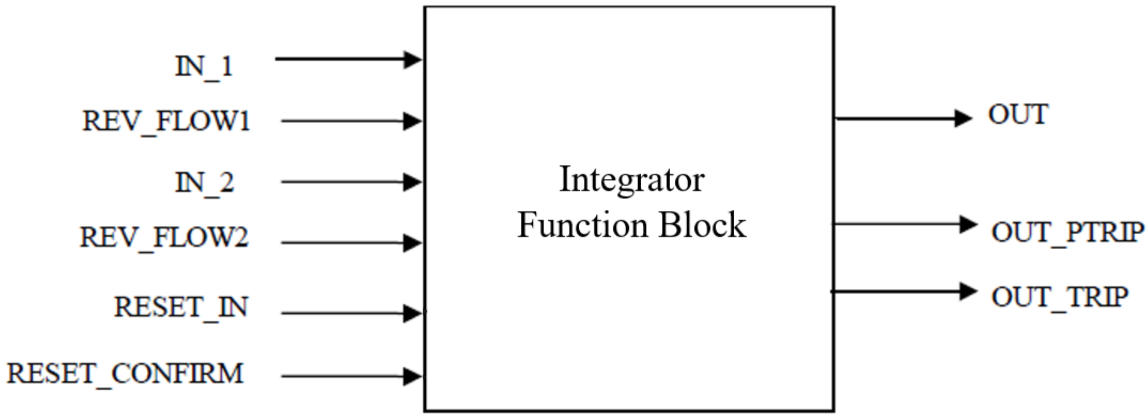
- 'Period of Execution' in the block header is zero
- Target mode is RCAS or ROUT and the SHED_OPT is 'Uninitialized'
- BYPASS is 'Uninitialized'
- 'EU at 100%' is equal or less than the 'EU at 0%' in one of the SCALE parameters
- SP_HI_LIM is less than the SP_LO_LIM
- OUT_HI_LIM is less than the OUT_LO_LIM

In case of a block configuration error the parameter BLOCK_ERR_DESC_1 shows the conditions preventing the block from proper operation.

4 FF Integrator Block

The Fieldbus Foundation Integrator block integrates a variable as a function of the time or accumulates the counts from a pulse input block. The Integrator block may be used as a totalizer that counts until reset or as a batch totalizer that has a setpoint, where the integrated or accumulated value is compared to pre-trip and trip settings, generating discrete signals when these settings are reached.

The block has two flow inputs so that it can calculate and integrate net flow.



4.1 Block options

Integration Options

Bit	Meaning	
0	Input 1 accumulate	Input 1 comes from an output, which generates a continuous accumulation of pulse counts.
1	Input 2 accumulate	Input 2 comes from an output, which generates a continuous accumulation of pulse counts.
2	Flow forward	Only positive flows are totalized. The negative values are treated as zero.
3	Flow reverse	Only negative flows are totalized. The positive values are treated as zero.
4	Use Uncertain	If the input status is Uncertain the increment status is taken as Good, and the new value is used. Otherwise, the increment status is Bad, and the last good value is used.
5	Use Bad	If the input status is Bad the increment status is taken as Good, and the new value is used. Otherwise, the increment status is Bad, and the last good value is used.
6	Carry	Reset always clears the internal registers Total, Atotal and Rtotal, except that when the option UP_AUTO or DN_AUTO is selected, a residual value beyond the trip value is carried to the next integration if the option “Carry” is set. In this case, TOTAL_SP is subtracted from Total, leaving the residual value.
7	Add zero if bad	The increment to set to zero if its input status is Bad
8	Confirm reset	This option, if set, prevents another reset from occurring until the value “Reset” has been written to RESET_CONFIRM

Status Option

Bit	Meaning	
8	Uncertain if Man mode	If this option is selected and the block mode is Manual, then the status of OUT, OUT_PTRIP, and OUT_TRIP is Uncertain constant. Otherwise, if the block mode is Manual, then the status of these three outputs is Good(NC) constant

4.2 Block errors

The Integrator block has two error conditions

Out-of-Service
The Integrator block shows the Out-of-Service error if its actual mode is OOS.

Block Configuration Error
The Integrator block shows a block configuration error if at least one of the following conditions is valid:

- 'Period of Execution' in the block header is zero
- The INTEG_TYPE is 'Uninitialized'
- TIME_UNIT1 or TIME_UNIT2 is 'Uninitialized'

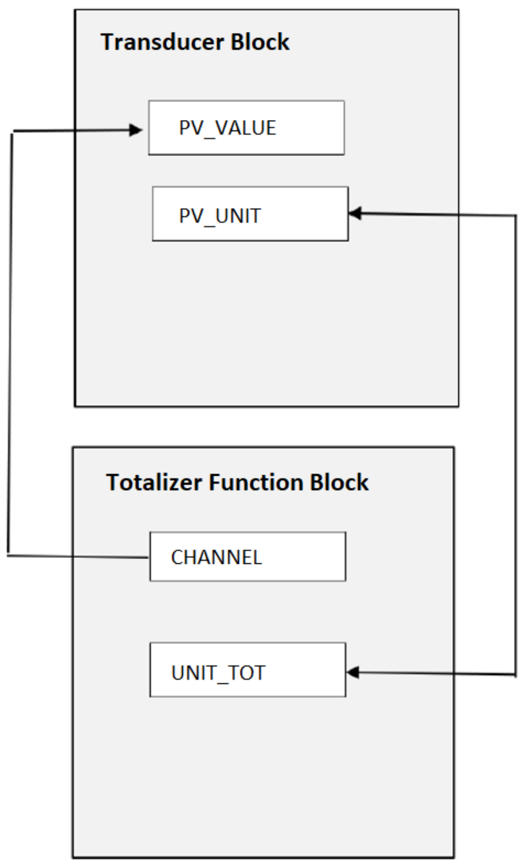
In case of a block configuration error the parameter BLOCK_ERR_DESC_1 shows the conditions preventing the block from proper operation.

5 PA Totalizer Block

This chapter describes the PROFIBUS Process Automation totalizer function block which can be used to totalize a volume or mass flow to a volume or a mass quantity.

The transducer block units for the rate and for the totalized quantity must fit together. This means that if the channel refers to mass flow (kg/s) parameter, the totalized quantity must be a compatible mass unit: kg, ton, etc.

5.1 Function block vs transducer block



- Function block CHANNEL refers to transducer PV_VALUE
- Function block UNIT_TOT and transducer PV_UNIT must fit together

5.2 Parameter overview

The following table shows the parameters of the PA totalizer block as specified by PA Profile V3.02.

Index	Parameter Name	Datatype (Simple Var) Data Structure (Record)	Size	Store	Use / Related Scale or Unit	Valid Range	Initial Value	Access
0	Block-Header	Block Structure (DS-64)	62	S	Contained			RW
1	ST_REV	Unsigned16	2	N	Contained		0	R
2	TAG_DESC	Octet String	32	S	Contained		Blanks	RW
3	STRATEGY	Unsigned16	2	S	Contained		0	RW
4	ALERT_KEY	Unsigned8	1	S	Contained	1 - 255	0	RW
5	MODE_BLK	Mode Structure (DS-69)	4	Mix	Contained	AUTO, OOS,CAS,RCAS,ROUT	OOS	RW
6	BLOCK_ERR	Bit String	2	D	Contained			R
7	PV	Float Value & Status (DS-65)	5	D	Contained / PV_SCALE		Bad – non specific / 0	R
8	SP	Float Value & Status (DS-65)	5	N	Contained / PV_SCALE	PV_SCALE +/- 10 %	GOOD_CAS / 0	RW (OOS | MAN | AUTO)
9	OUT	Float Value & Status (DS-65)	5	N	Output / OUT_SCALE	OUT_SCALE +/- 10 %	BAD – Out of service 0	RW (OOS | MAN)
10	PV_SCALE	Scaling Structure (DS-68)	11	S	Contained		0-100%	RW (OOS)
11	OUT_SCALE	Scaling Structure (DS-68)	11	S	Contained		0-100%	RW (OOS)
12	GRANT_DENY	Access Permissions (DS-70)	2	S	Contained		0,0	
13	CONTROL_OPTS	Bit String	2	S	Contained		0	RW (OOS)
14	STATUS_OPTS	Bit String	2	S	Contained		0	RW (OOS)
15	IN	Float Value & Status (DS-65)	5	N	Input / PV_SCALE		Bad – Not connected / 0	
16	PV_FTIME	Float	4	S	Contained / seconds	Positive or zero	0	
17	BYPASS	Unsigned8	1	S	Contained / Enumeration	0: Uninitialized 1: Off, 2: On	Uninitialized	RW (OOS | MAN)
18	CAS_IN	Float Value & Status (DS-65)	5	D	Input / PV_SCALE		Bad – Not connected / 0	
19	SP_RATE_DN	Float	4	S	Contained / PV units per second	Positive or zero	+INF	
20	SP_RATE_UP	Float	4	S	Contained / PV units per second	Positive or zero	+INF	
21	SP_HI_LIM	Float	4	S	Contained / PV_SCALE	PV_SCALE +/- 10 %	100	
22	SP_LO_LIM	Float	4	S	Contained / PV_SCALE	PV_SCALE +/- 10 %	0	

Index	Parameter Name	Datatype (Simple Var) Data Structure (Record)	Size	Store	Use / Related Scale or Unit	Valid Range	Initial Value	Access
23	GAIN	Float	4	S	Contained / no unit	Positive or zero	0	
24	RESET	Float	4	S	Contained / second	Positive or zero	+INF	
25	BAL_TIME	Float	4	S	Contained / second	Positive or zero	0	
26	RATE	Float	4	S	Contained / second	Positive or zero	0	
27	BKCAL_IN	Float Value & Status (DS-65)	5	D	Input / OUT_SCALE		Bad – Not connected / 0	
28	OUT_HI_LIM	Float	4	S	Contained / OUT_SCALE	OUT_SCALE +/- 10 %	100	
29	OUT_LO_LIM	Float	4	S	Contained / OUT_SCALE	OUT_SCALE +/- 10 %	0	
30	BKCAL_HYS	Float	4	S	Contained / percent	0 – 50 %	0.5 %	
31	BKCAL_OUT	Float Value & Status (DS-65)	5	D	Output / PV_SCALE		Bad – Non specific / 0	R
32	RCAS_IN	Float Value & Status (DS-65)	5	D	Contained / PV_SCALE		Bad – Out of service / 0	
33	ROUT_IN	Float Value & Status (DS-65)	5	D	Contained / OUT_SCALE		Bad – Out of service / 0	
34	SHED_OPT	Unsigned8	1	S	Contained / Enumeration		Uninitialized	
35	RCAS_OUT	Float Value & Status (DS-65)	5	D	Contained / PV_SCALE		Bad – Non specific / 0	R
36	ROUT_OUT	Float Value & Status (DS-65)	5	D	Contained / OUT_SCALE		Bad – Non specific / 0	R
37	TRK_SCALE	Scaling Structure (DS-68)	11	S	Contained		0-100%	RW (OOS)
38	TRK_IN_D	Discrete Value & Status (DS-66)	2	N	Input	0: Off not 0: On	Bad – Not connected / Off	

(1) *The initial value is specified in the commScript*
(2) *Write access is restricted. For write access to parameters UNIT_TOT and CHANNEL see Section Write access to parameters CHANNEL and UNIT_TOT*

5.3 Parameter descriptions

ST_REV (Static Revision)
This parameter shows the revision level of the static data associated with the Totalizer block. To support tracking changes in static parameter attributes, the static revision parameter is incremented each time a static parameter attribute value is changed.

TAG_DESC (Tag Description)
This rarely used parameter allows a user to describe the intended function the block is performing.

STRATEGY (Strategy)
This rarely used parameter allows the user to group blocks by means of a set of group-identifiers. The strategy parameter is not checked or evaluated by the Totalizer block.

ALERT_KEY (Alert Key)
This rarely used parameter allows the user to identify different plant units. This information may be used in the host for sorting alarms, etc. The parameter does not influence the behavior of the Totalizer block.

TARGET_MODE (Target Mode)

The TARGET_MODE parameter indicates which mode is desired for the block. Only one mode from those allowed by the "Permitted" element of the MODE_BLK parameter can be selected.

MODE_BLK (Mode of Block)
The mode of the block is a structured parameter composed of the actual, the normal and the permitted mode.

- The actual mode is calculated by the block during its execution to reflect the mode used during execution.
- Automatic is the normal operating mode of the Totalizer block.
- The permitted mode shows the target modes valid for the Totalizer block:
  - Out of Service (OOS)
  - Manual (Man)
  - Automatic (Auto)

ALARM_SUM (Alarm Summery)
The parameter ALARM_SUM summarizes the status of up to 16 block alarms. For each alarm, the current states, unacknowledged states, unreported states, and disabled states are maintained. This feature is not fully supported by PROFIBUS-PA profile V3.02. For this profile, the current state part of the alarm is used only.

BATCH (Batch)
This parameter is intended to be used in Batch applications according to IEC 61512-1. The parameter does not influence the behavior of the Totalizer block.

TOTAL (Totalized Quantity)
The total parameter contains the integrated quantity and the associated status.

UNIT_TOT (Unit of totalized quantity)
Unit of the totalized quantity. For supported units see Supported flow and totalizer units (45).

CHANNEL (Channel)
Reference to the transducer block parameter, which provides the measured flow rate to the Totalizer block.

SET_TOT (Set/Reset Totalization)
This parameter is used to set the totalized value to a preset value or to reset the totalized value to zero. The following selections are possible:

0: TOTALIZE	Normal operation of the Totalizer
1: RESET	Reset totalized value to zero
2: PRESET	Set totalized value to the value of PRESET_TOT

MODE_TOT (Mode of Totalization)
Controls the mode of totalization. The following selections are possible:

0: BALANCED	True arithmetic integration of the incoming rate values.
1: POS_ONLY	Totalization of positive incoming rate values only.
2: NEG_ONLY	Totalization of negative incoming rate values only.
3: HOLD	Totalization is stopped.

FAIL_TOT (Fail-safe mode)
Controls the behavior of the Totalizer block during the occurrence of input values with Bad status. The following selections are possible:

0: RUN	Totalization is continued using the input values despite the Bad status.
1: HOLD	Totalization is stopped during occurrence of Bad status of incoming values.
2: MEMORY	Totalization is continued based on the last incoming value with Good status.

PRESET_TOT (Preset for Totalized Value)
This value is used as a preset for the totalized value. The value gets effective if using the SET_TOT selection PRESET.

ALARM_HYS (Alarm Hysteresis)
Amount the totalized value must return within the alarm limits before the alarm condition clears. The hysteresis is expressed as value below high limit and above low limit in the engineering unit of UNIT_TOT

HI_HI_LIM (High High Limit)
Value for upper limit of alarms. If the totalizer value is equal to or higher than the upper limit value the “high limited” bit in the status of TOTAL, and the “HI_HI_Alarm” bit in the ALARM_SUM is set. The unit of this parameter is defined by UNIT_TOT.

HI_LIM (High Limit)
Value for upper limit of warnings. If the totalizer value is equal to or higher than the upper limit value the “high limited” bit in the status of TOTAL, and the “HI_Alarm” bit in the ALARM_SUM is set. The unit of this parameter is defined by UNIT_TOT.

LO_LIM (Low Limit)
Value for lower limit of warnings. If the totalizer value is equal to or lower than the lower limit value the “low limited” bit in the status of TOTAL, and the “LO_Alarm” bit in the ALARM_SUM is set. The unit of this parameter is defined by UNIT_TOT.

LO_LO_LIM (Low Low Limit)
Value for lower limit of alarms. If the totalizer value is equal to or lower than the lower limit value the “low limited” bit in the status of TOTAL, and the “LO_LO_Alarm” bit in the ALARM_SUM is set. The unit of this parameter is defined by UNIT_TOT.

HI_HI_ALM (High High Alarm)
Upper limit alarm. The alarm feature is not fully supported by PROFIBUS-PA profile V3.02. For this profile, only the “Alarm_State (no alarm / alarm active)” is supported.

HI_ALM (High Alarm)
Upper limit warning. The alarm feature is not fully supported by PROFIBUS-PA profile V3.02. For this profile, only the “Alarm_State (no alarm / alarm active)” is supported.

LO_ALM (Low Alarm)
Lower limit warning. The alarm feature is not fully supported by PROFIBUS-PA profile V3.02. For this profile, only the “Alarm_State (no alarm / alarm active)” is supported.

LO_LO_ALM (Low Low Alarm)
Lower limit alarm. The alarm feature is not fully supported by PROFIBUS-PA profile V3.02. For this profile, only the “Alarm_State (no alarm / alarm active)” is supported.

BLOCK_ERR_DESC (Block Error Description)
Softing specific block extension parameter. Indicates block error conditions. See Section Additional block parameter BLOCK_ERR_DESC [44].

5.4 Additional block parameter BLOCK_ERR_DESC

The parameter BLOCK_ERR_DESC is of type BIT_ENUMERATED(4) (4 bytes, 32 bits). It indicates two block error conditions:

0x00000002	The physical block is in mode OOS. In this case all function blocks also stay in mode OOS
0x10000000	There is mass/volume mismatch between CHANNEL, UNIT_TOT and TB unit parameters

5.5 Totalizer algorithm

The totalizer algorithm evaluates the PV unit parameter in the transducer block and the UNIT_TOT parameter in the totalizer function block:

- If the transducer PV unit and the UNIT_TOT unit fit together (volume flow/volume or mass flow/mass) the totalizer algorithm performs volume flow to volume or mass flow to mass unit conversions (e.g. transducer PV unit: liter/sec, UNIT_TOT: barrel)
- If the transducer channel unit and the UNIT_TOT unit do not match (volume/mass or mass/volume mismatch) then the totalizer block switches to error mode OOS.
- If the field device is power cycled, then the totalizer block restores the totalized value.
- If the transducer block provides a flow process value with BAD status then the totalizer block activates its fail-safe mode (for details of the fail-safe mode see “PROFIBUS PA Profile for Process Control Devices, V3.02”).
- If the totalizer block runs in MAN mode, then totalization is continued in background but the TOTAL parameter is not updated anymore. If the totalizer block switches back from MAN to AUTO mode, then the TOTAL parameter shows the continuously totalized value again.
- If the totalizer block is out-of-service, then totalization is stopped, and the TOTAL parameter is not updated anymore.

5.6 Supported flow and totalizer units

The transducer block unit parameters support the following volume flow and mass flow units:

5.6.1 Volume flow units

PA unit code	EDD keyword	Description
1347	m^3/s	Cubic meter per second
1348	m^3/min	Cubic meter per minute
1349	m^3/h	Cubic meter per hour
1350	m^3/d	Cubic meter per day
1351	L/S	Liter per second
1352	L/min	Liter per minute
1353	L/h	Liter per hour
1355	ML/d	Megaliter per day
1356	CFS	Cubic feet per second
1357	CFM	Cubic feet per minute
1358	CFH	Cubic feet per hour
1359	ft^3/d	Cubic feet per day
1360	SCFM	Standard cubic feet per minute
1361	SCFH	Standard cubic feet per hour (32°F, 1atm)
1362	gal/s	US gallon per second
1363	GPM	US gallon per minute
1364	gal/h	US gallon per hour
1365	gal/d	US gallon per day
1366	Mgal/d	Mega US gallon per day
1367	ImpGal/h	Imperial gallon per second
1368	ImpGal/min	Imperial gallon per minute
1369	ImpGal/h	Imperial gallon per hour
1370	ImpGal/d	Imperial gallon per day
1371	bbl/s	Barrel per second
1372	bbl/min	Barrel per minute
1373	bbl/h	Barrel per hour
1374	bbl/d	Barrel per day
1588	m3/s normal	Normal cubic meter per second (0°C, 1atm)
1589	m3/min normal	Normal cubic meter per minute (0°C, 1atm)
1590	m3/h normal	Normal cubic meter per hour (0°C, 1atm)
1591	m3/d normal	Normal cubic meter per day (0°C, 1atm)
1594	L/h normal	Normal liter per hour (0°C, 1atm)
1596	m3/s std.	Standard cubic meter per second (20°C,1atm)
1597	m3/min std.	Standard cubic meter per minute (20°C,1atm)
1598	m3/h std.	Standard cubic meter per hour (20°C, 1atm)
1599	m3/d std.	Standard cubic meter per day (20°C, 1atm)
1600	L/s std.	Standard liter per second (20°C, 1atm)

5.6.2 Mass flow units

PA unit code	EDD keyword	Description
1318	g/s	Gram per second
1319	g/min	Gram per minute
1320	g/h	Gram per hour
1322	kg/s	Kilogram per second
1323	kg/min	Kilogram per minute
1324	kg/h	Kilogram per hour
1325	kg/d	Kilogram per day
1327	t/min	Metric ton per minute
1328	t/h	Metric ton per hour
1329	t/d	Metric ton per day
1330	lb/s	Pound per second
1331	lb/min	Pound per minute
1332	lb/h	Pound per hour
1333	lb/d	Pound per day
1335	STon/min	Short ton per minute
1336	STon/h	Short ton per hour
1337	STon/d	Short ton per day
1340	LTon/h	Long ton per hour
1341	STon/d	Long ton per day

5.6.3 UNIT_TOT units

The following volume and mass units are supported:

5.6.3.1 UNIT_TOT volume units

PA unit code	EDD keyword	Description
1034	m^3	Cubic meters
1038	L	Liter
1043	ft3	Cubic feet
1053	SCF	Standard cubic feet U.S. System
1048	Gallon	US gallons
1049	ImpGal	Imperial gallons
1051	bbl	Barrel
1575	m3 std.	Standard cubic meter
1573	m3 normal	Normal cubic meter
1576	L std.	Standard liter
1574	L normal	Normal liter

5.6.3.2 UNIT_TOT mass units

PA unit code	EDD keyword	Description
1089	g	gram
1088	kg	kilogram
1092	t	metric ton
1094	lb	pound
1095	STon	short ton
1096	LTon	long tomn

6 Transducer Blocks

6.1 Transducer blocks manual mode behavior

Only transducer blocks which have at least one output process value support the block mode MAN. If manual mode is set, the transducer block parameters marked with a W_MAN flag in the commScript become writable and the process value parameter of the output channels will not be updated with the final values calculated by the PLC anymore.

This leads to a temporarily decoupling of the final elements of the device from the final values calculated by the PLC. Once the transducer block is set to manual mode, the status of a final value parameters will switch to GOOD_CAS.

6.2 Softing diagnostic transducer block

The Softing specific Diagnostic Transducer Block (DIAG_TB)provides information about internal states, errors and performance of the function block application including the communication layers thereby allowing for measuring execution times and scheduling the precision of the function blocks.

- Information on the revision of the FD-SW
- Information about the Fieldbus Controller
- Information on last fatal error
- Time since last device restart
- Information on last exception
- Time since last exception
- Chip temperature of the communication interface MCU
- Safety margin of power supply current as measured from MAU shunt regulator
- Operation and error statistics of the fieldbus controller
- FF only: Information on last DLL restart
- FF only: Time since last DLL restart
- FF only: Function block execution statistics

Note

Bear in mind that functions and structure of the DIAG_TB may change with future versions of the Softing FD-SW which is determined by the revision value of the DIAG_TB (parameter DIAG_TB_REVISION).

All parameters listed below appear after the protocol specific standard parameters of all transducer blocks.

6.2.1 Hardware and protocol stack diagnostic information

Parameter	Data Type	Meaning
DIAG_TB_REVISION	USIGN8	Version of the DIAG_TB structure, currently “2”
NV_STATUS	USIGN8	Status of internal NV data storage; 0 = status_ok
NV_DIAG	USIGN32[10]	Additional details of NV data storage diagnosis
FD_SW_REV	VSTRING[16]	SW revision of the core protocol implementation
FD_HW_REV	VSTRING[16]	HW revision of the communication interface
FB_CONTROLLER	USIGN8	1= UFC100, 2 = FIND1PLUS, 3=SPC4-2, 4 = RX-SOM, 5 = FPGA based, 6 = DAC8740
HW_TEST_INFO	BITSTRING(2)	Currently not used. Fixed values "HW Test 1 ok" and "HW Test 2 ok" bits set.
TIME_SINCE_LAST_ RESTART	USIGN32[4]	Time since last processor restart (warm start) in days, hours, minutes, seconds up to now. Note: With proceeding time these time values will increase.
FATAL_ERROR_INFO	USIGN32[4]	Internal detail information of the last fatal error
TIME_SINCE_LAST_ EXCEPTION	USIGN32[4]	Time since last exception (software error) in days, hours, minutes, seconds up to now. If the software error was fatal, also a processor restart is triggered. Note: With proceeding time these time values will increase.
EXCEPTION_INFO	USIGN32[4]	Internal detail information of the last exception
CHIP_TEMPERATURE	INT32[3]	Chip temperature (current, min, max) of the MCU / communication interface in units of 0.1°C
SHUNT_CURRENT	INT32[3]	Current of the MAU shunt regulator (current, min, max) in units of µA
SHUNT_HIST	USIGN32[16]	Shunt current histogram in slices of 2mA
TIME_SINCE_LAST_DLL_ RESTART	USIGN32[4]	**FF only:** Time since last data link layer restart in days, hours, minutes, seconds up to now. Note: With proceeding time these time values will increase.
DLL_RESTART_INFO	USIGN8[2]	**FF only:** Reason/State for last DLL restart
FBC_STATISTIC_CNT	USIGN32[1]	FBC statistic counters
FBC_ERROR_CNT	USIGN32[10]	FBC error counters

6.2.2 FF only: Function block execution times

Parameter	Data Type	Meaning
FB_EXEC_IDX:	USIGN16	Start index of the function block instance being monitored. For the function block start indices see commScripter generated .cff file. Starts the measurement of function block execution time for the function block with start index FB_EXEC_IDX, a value of 0 stops the measurement. Writing a valid function block index value to FB_EXEC_IDX will reset the measurement results to 0 and restart the measurement.
FB_EXEC_SUM	USIGN32[5]	Summary of FB execution statistics, with following info in detail:
FB_EXEC_SUM[0]:		Current number of measurements since start of measurement.
FB_EXEC_SUM[1]:		Maximum time from the scheduled start of function block execution until the function block execution is finished. Under normal conditions (function block execution is started in time) this is the maximum function block execution time in units of 1/32 msec.
FB_EXEC_SUM[2]:		Average time (from last 128 measurements) from the scheduled start of function block execution until the function block execution is finished. Under normal conditions (function block execution is started in time) this is the average function block execution time in units of 1/32 msec.
FB_EXEC_SUM[3]:		Maximum delay between scheduled start of function block execution and actual start of function block execution in units of 1/32 msec.
FB_EXEC_SUM[4]:		Maximum premature start of function block execution compared to the scheduled function block execution start time in units of 1/32 msec.
FB_EXEC_HIST[0..47]:	USIGN32[48]	Distribution of time measured from the scheduled function block start time until the function block execution is finished in intervals of 0.5ms. E. g. FB_EXEC_HIST[10] counts how often the measured time was between 5 and 5.5ms. FB_EXEC_HIST[47] counts all measured times greater than 23.5ms.
FB_EXEC_DEV_POS[0..31]:	USIGN32[32]	Distribution of the delay (positive deviation) between scheduled function block execution start time and actual start time in intervals of 0.125ms. E.g. FB_EXEC_DEV_POS[0] counts how often a delay between 0 and 0.125ms was measured. FB_EXEC_DEV_POS[31] count all measured times greater than 3.875ms.
FB_EXEC_DEV_NEG[0..31]:	USIGN32[32]	Distribution of premature actual function block execution start times (negative deviation) in intervals of 0.125ms. E. g. FB_EXEC_DEV_NEG[0] counts how often a premature function block execution start time between 0 and 0.125ms was measured.

6.3 Softing startHART demo application

6.3.1 Overview

The startHART demo application is a realization of an FF/PA device that provides a variety of function blocks that can be used for test purposes. It also contains two device specific transducer blocks which contain parameters that can be used within the input and output blocks. Furthermore it contains the Softing Diagnosis (DIAG_TB) that provides diagnosis information for the field device. The DIAG_TB is internally created within the Field Device software and will be available in every application.

The contained blocks are:

- Resource Block (FF), Physical Block (PA)
- Analog Input Block (2x)
- Discrete Input Block (2x)
- Analog Output Block (2x)
- Discrete Output Block (2x)
- Proportional integral derivative (PID) Block (FFonly)
- SCP_TB (FF only)
- DIAG_TB
- Transducer Block 1: DEMO_TB
- Transducer Block 2: HART_TB

Resource block/physical block are described in the *commScripter MBP User Guide*. No device specific parameters are defined as extension to the resource/physical block. Use and behaviour of the above function blocks are defined in the related protocol specifications are therefore not described in this documentation except for the PID block which is included in Chapter FF Integrator Block 36.

The application specific transducer blocks, however, are described in the next chapters.

6.3.2 Fieldbus representation of the startHART device

6.3.2.1 FF field device

The startHART FF device is using Softing manufacturer ID (0x1E6D11) and device type (0x0320). So the resulting FF Device ID is ‘1E6D110320 commMod<sssssssss>’ where <sssssssss> is the 9-digit serial number of the used commModule.

The following list shows the main FF attributes of the device:

Parameter	Value
Manufacturer ID	0x1E6D11
Device ID	0x0320
Device Revision	5 (v3.50), 6 (v3.60)
DD Revision	1
Input current	26 mA

For host system integration of the startHART FF device DD files and a CFF file are available supporting format 5 and 4. The files follow the standard naming conventions for DD files (060101.cff, 0601.ff5 / 0601.ffo and 0601.sy5 / 0601.sym).

The contained blocks are located as described in the table below:

Block	Index Block Object	No. of Parameters	View Object 1
Resource Block	500	72	1900-1903
Analog Input 1 FB	600	38	1950-1953
Analog Input 2 FB	700	38	2000-2003
Discrete Input 1 FB	800	26	2050-2053
Discrete Input 2 FB	900	26	2100-2103
Analog Output 1 FB	1000	32	2150-2153
Analog Output 2 FB	1100	32	2200-2203
Discrete Output 1 FB	1200	28	2250-2253
Discrete Output 2 FB	1300	28	2300-2303
PID FB	1400	67	2350-2353
SCP_TB	1500	36	2400-2403
DIAG_TB	1600	37	2450-2453
DEMO_TB	1700	51	2500-2503
HART_TB	1800	23	2550-2553

6.3.2.2 PA field device

The startHART PA device is using Softing PA manufacturer ID (279), the manufacturer ident number (TBD by PNO) and the profile ident number 0x9701 (2 Analog Input Blocks).

The following list shows the major PA attributes of the device:

Parameter	Value
Manufacturer ID	279
Manufacturer Ident No.	0x1169
Profile Ident No.	0x9701
Input current	26 mA

For integration of the startHART PA device a GSD file and EDD files are available. The GSD file is SOFT1169.gsd. The EDD description is distributed in several files as descripted in commScripter MBP user guide, Chapter 4.5.

Slot/Index List for the Blocks:

Block	Slot/Start Index	No. of Parameters	Index View Object 1	Parent Class/Class
Physical Block	0 / 16	38	230	Not used / not used
Analog Input 1 FB	1 / 16	46	230	Input / Analog Input
Analog Input 2 FB	2 / 16	46	230	Input / Analog Input
Discrete Input 1 FB	3 / 16	36	230	Input / Discrete Input
Discrete Input 2 FB	4 / 16	36	230	Input / Discrete Input
Analog Output 1 FB	5 / 16	50	230	Output / Analog
Analog Output 2 FB	6 / 16	50	230	Output / Analog
Discrete Output 1 FB	7 / 16	46	230	Output / Discrete
Discrete Output 2 FB	8 /16	46	230	Output / Discrete
DEMO_TB	9 / 16	46	230	Not used / not used
HART_TB	10 / 16	17	230	Not used / not used
DIAG_TB	11 / 16	24	230	Not used / not used

6.3.3 FF transducer blocks

As mentioned above two device specific transducer blocks are realized in the startHART device. The following chapters present an overview of the parameters of these transducer blocks and their attributes. A detailed description of the functionality of these parameters is given in Chapter Operation 61.

6.3.3.1 DEMO_TB parameters

The DEMO_TB transducer block provides several parameters that can be used as process variables and the related units.

Note

The functionality provided by these parameters is implemented in the firmware and is activated only if the first transducer block is the Softing DEMO_TB without any change when used in the Softing startHART device. Using the block in another application is not possible. Therefore the commScripter MBP restricts to use a transducer block named 'DEMO_TB' in a vendor specific application.

The following table shows a list of transducer block parameters:

Index	Parameter Name	Datatype (Simple Var) Data Structure (Record)	Size	Store	Valid Range	Initial Value	Access
0	Block-Header	Block Structure (DS-64)	62	S			RW
1	ST_REV	Unsigned16	2	N		0	R
2	TAG_DESC	Octet String	32	S		Blanks	RW
3	STRATEGY	Unsigned16	2	S		0	RW
4	ALERT_KEY	Unsigned8	1	S	1-255	0	RW
5	MODE_BLK	Mode Structure (DS-69)	4	Mix	AUTO,OOS	target mode /actual mode = OOS default mode = AUTO permitted mode = OOS, AUTO	RW
6	BLOCK_ERR	Bit String	2	D		0	R
7	UPDATE_EVT	Event Update (DS-73)	14	D			RW
8	BLOCK_ALM	Alarm Discrete (DS-72)	13	D			RW
9	TRANSDUCER_DIRECTORY	Array with 1 Unsigned16	2	S			R
10	TRANSDUCER_TYPE	Unsigned16	2	S			R
11	TRANSDUCER_TYPE_VER	Unsigned16	2	N			R
12	XD_ERROR	Unsigned8	1	D		0	R
13	COLLECTION_DIRECTORY	Array with 1 Unsigned32	4	S			R
14	CONST_VALUE	PV Float (DS-65)	5	N		0x80, 0.0	RW
15	CONST_UNIT	Unsigned16	2	S		1342 (%)	R
16	RAMP_VALUE	PV Float (DS-65)	5	D		0x80, 0.0	R
17	RAMP_UNIT	Unsigned16	2	S		1342 (%)	R
18	RAMP_RATE	Float	4	S		1.0	RW

Index	Parameter Name	Datatype (Simple Var) Data Structure (Record)	Size	Store	Valid Range	Initial Value	Access
19	RAMP_MIN	Float	4	S		0.0	RW
20	RAMP_MAX	Float	4	S		100.0	RW
21	RECT_VALUE	PV Float (DS-65)	5	D		0x80, 0.0	R
22	RECT_UNIT	Unsigned16	2	S		1342 (%)	R
23	RECT_PERIOD	Unsigned16	2	S		10	RW
24	RECT_MIN	Float	4	S		10.0	RW
25	RECT_MAX	Float	4	S		90.0	RW
26	TEMP_VALUE	PV Float (DS-65)	5	D		0x80, 0.0	R
27	TEMP_UNIT	Unsigned16	2	S	1000 (K) 1001 (°C) 1002 (°F)	1001 (°C)	RW
28	ANALOG_INPUT_1_VALUE	PV Float (DS-65)	2	D		0x80, 0.0	R
29	ANALOG_INPUT_1_UNIT	Unsigned16	2	D		1342 (%)	R
30	ANALOG_INPUT_2_VALUE	PV Float (DS-65)	5	D		0x80, 0.0	R
31	ANALOG_INPUT_2_UNIT	Unsigned16	2	D		1342 (%)	R
32	CONST_VALUE_D	PV Discrete (DS-66)	2	S		0x80, 0	RW
33	RAMP_VALUE_D	PV Discrete (DS-66)	2	D		0x80, 0	R
34	RAMP_RATE_D	Unsigned8	1	S		1	RW
35	RAMP_MIN_D	Unsigned8	1	S		0	RW
36	RAMP_MAX_D	Unsigned8	1	S		100	RW
37	RECT_VALUE_D	PV Discrete (DS-66)	2	D		0x80, 0	R
38	RECT_PERIOD_D	Unsigned16	2	S		10	RW
39	RECT_MIN_D	Unsigned8	1	S		1	RW
40	RECT_MAX_D	Unsigned8	1	S		2	RW
41	DISCRETE_INPUT_1_VALUE	PV Discrete (DS-66)	2	D		0x80, 0	R
42	DISCRETE_INPUT_2_VALUE	PV Discrete (DS-66)	2	D		0x80, 0	R
43	FINAL_VALUE	PV Float (DS-65)	5	N		0x80, 0.0	RW
44	FINAL_POSITION_VALUE	PV Float (DS-65)	5	D		0x80, 0.0	R
45	FINAL_VALUE_RANGE	Scaling (DS-68)	11	D		100.0, 0.0, 1342 (%), 2	R
46	FINAL_VALUE_2	PV Float (DS-65)	5	D		0x80, 0.0	R
47	FINAL_VALUE_2_UNIT	Unsigned16	2	D		1342 (%)	R
48	FINAL_VALUE_D	PV Discrete (DS-66)	2	N		0x80, 0	RW
49	FINAL_POSITION_VALUE_D	PV Discrete (DS-66)	2	D		0x80, 0	R
50	FINAL_VALUE_D_2	PV Discrete (DS-66)	2	D		0x80, 0	R

6.3.3.2 HART_TB parameters

The HART_TB transducer block contains some parameters which are mapped to a HART device. The mapping uses only HART universal commands. Therefore it should be possible to connect a HART device when the UART connection to the commModule can be used without modifications.

See Chapter HART mapping (66) for HART mapping and UART attributes.

Index	Parameter Name	Data Type (Simple Var) Data Structure (Record)	Size	Store	Valid Range	Initial Value	Access
0	Block-Header	Block Structure (DS-64)	62	S			RW
1	ST_REV	Unsigned16	2	N		0	R
2	TAG_DESC	Octet String	32	S		Blanks	RW
3	STRATEGY	Unsigned16	2	S		0	RW
4	ALERT_KEY	Unsigned8	1	S	1-255	0	RW
5	MODE_BLK	Mode Structure (DS-69)	4	Mix	AUTO, OOS	target mode /actual mode = OOS default mode = AUTO permitted mode = OOS, AUTO	RW
6	BLOCK_ERR	Bit String	2	D		0	R
7	UPDATE_EVT	Event Update (DS-73)	14	D			RW
8	BLOCK_ALM	Alarm Discrete (DS-72)	13	D			RW
9	TRANSDUCER_DIRECTORY	Array with 1 Unsigned16	2	S			R
10	TRANSDUCER_TYPE	Unsigned16	2	S			R
11	TRANSDUCER_TYPE_VER	Unsigned16	2	N			R
12	XD_ERROR	Unsigned8	1	D		0	R
13	COLLECTION_DIRECTORY	Array with 1 Unsigned32	4	S			R
14	PRIMARY_VALUE	PV Float (DS-65)	5	D			R
15	PV_UNIT	Unsigned16	2	S			R
16	SECONDARY_VALUE	PV Float (DS-65)	5	D			R
17	SV_UNIT	Unsigned16	2	S			R
18	HART_DEVICE_IDENT	DS-256	9	D			R
19	HART_DEVICE_INFO	DS-259	18	D			R
20	HART_PV_INFO	DS-258	18	D			R
21	HART_TAG_DESC_DATE	DS-257	27	S			RW
22	HART_ADDITIONAL_DEV_STATUS	Array of 4 BitString(1)	4	S			R

6.3.3.3 Device specific data structures

Device specific data structures are used within this transducer blocks. The used data structures are defined as:

DS-256 (HART_CMD_0_STRUCT)

Subindex	Meaning	Data Type
1	HART Device Type	Unsigned16
2	HART Device Revision	Unsigned8
3	HART SW Revision	Unsigned8
4	HART HW Revision	Unsigned8
5	HART Device ID	Unsigned32

DS-257 (HART_TAG_DESC_DATE_STRUCT)

Subindex	Meaning	Data Type
1	Tag	VisibleString(8)
2	Descriptor	VisibleString(16)
3	Day	Unsigned8
4	Month	Unsigned8
5	Year	Unsigned8

DS-258 (HART_PV_INFO_STRUCT)

Subindex	Meaning	Data Type
1	Transducer Serial Number	Unsigned32
2	Unit Code	Unsigned16
3	Upper Transducer Limit	Float
4	LowerTransducer Limit	Float
5	Minimum Span	Float

DS-259 (HART_DEVICE_INFO_STRUCT)

Subindex	Meaning	Data Type
1	PV Alarm Selection Code	Unsigned8
2	PV Transfer Function Code	Unsigned8
3	PV Range Unit Code	Unsigned16
4	PV Upper Range Value	Float
5	PV Lower Range Value	Float
6	PV Damping Value	Float
7	Write Protect Code	Unsigned8
8	PV Analog Channel Flags	Unsigned8

6.3.4 PA transducer blocks

For the PA device the same transducer blocks are used as in the FF device using the same device specific transducer block parameter. As the standard parameters of a PA transducer block differ from standard FF transducer block parameters all parameters of the PA transducer blocks are listed in the following chapters. A detailed description is given in Chapters Operation [61] and HART mapping [66].

Process values in PA have the value + status while in FF the process values are status + value. So an analog process value which has the value 1.23 and status GoodNonCascade (=0x80) will be 1.23, 0x80 in PA and 0x80, 1.23 in FF.

6.3.4.1 DEMO_TB parameters

Index	Parameter Name	Data Type (Simple Var) Data Structure (Record)	Size	Store	Valid Range	Initial Value	Access
16	Block-Header	Block Structure (DS-100)	62	S			RW
17	ST_REV	Unsigned16	2	N		0	R
18	TAG_DESC	Octet String	32	S		Blanks	RW
19	STRATEGY	Unsigned16	2	S		0	RW
20	ALERT_KEY	Unsigned8	1	S	1-255	0	RW
21	TARGET_MODE	Unsigned8	1	S			RW
22	MODE_BLK	Mode Structure	3	D			R
23	ALARM_SUM	Alarm Discrete (DS-72)	13	D			RW
24	CONST_VALUE	PV Float (DS-101)	5	N		0.0, 0x80	RW
25	CONST_UNIT	Unsigned16	2	S		1342 (%)	R
26	RAMP_VALUE	PV Float (DS-101)	5	D		0.0, 0x80	R
27	RAMP_UNIT	Unsigned16	2	S		1342 (%)	R
28	RAMP_RATE	Float	4	S		1.0	RW
29	RAMP_MIN	Float	4	S		0.0	RW
30	RAMP_MAX	Float	4	S		100.0	RW
31	RECT_VALUE	PV Float (DS-101)	5	D		0.0, 0x80	R
32	RECT_UNIT	Unsigned16	2	S		1342 (%)	R
33	RECT_PERIOD	Unsigned16	2	S		10	RW
34	RECT_MIN	Float	4	S		10.0	RW
35	RECT_MAX	Float	4	S		90.0	RW
36	TEMP_VALUE	PV Float (DS-101)	5	D	1000 (K) 1001 (°C) 1002 (°F)	0.0, 0x80	R
37	TEMP_UNIT	Unsigned16	2	S		1001 (°C)	RW
38	ANALOG_INPUT_1_VALUE	PV Float (DS-101)	2	D		0.0, 0x80	R
39	ANALOG_INPUT_1_UNIT	Unsigned16	2	D		1342 (%)	R

Index	Parameter Name	Data Type (Simple Var) Data Structure (Record)	Size	Store	Valid Range	Initial Value	Access
40	ANALOG_INPUT_2_VALUE	PV Float (DS-101)	5	D		0.0, 0x80	R
41	ANALOG_INPUT_2_UNIT	Unsigned16	2	D		1342 (%)	R
42	CONST_VALUE_D	PV Discrete (DS-102)	2	S		0.0, 0x80	RW
43	RAMP_VALUE_D	PV Discrete (DS-102)	2	D		0.0, 0x80	R
44	RAMP_RATE_D	Unsigned8	1	S		1	RW
45	RAMP_MIN_D	Unsigned8	1	S		0	RW
46	RAMP_MAX_D	Unsigned8	1	S		100	RW
47	RECT_VALUE_D	PV Discrete (DS-102)	2	D		0.0, 0x80	R
48	RECT_PERIOD_D	Unsigned16	2	S		10	RW
49	RECT_MIN_D	Unsigned8	1	S		1	RW
50	RECT_MAX_D	Unsigned8	1	S		2	RW
51	DISCRETE_INPUT_1_VALUE	PV Discrete (DS-102)	2	D		0.0, 0x80	R
52	DISCRETE_INPUT_2_VALUE	PV Discrete (DS-102)	2	D		0.0, 0x80	R
53	FINAL_VALUE	PV Float (DS-101)	5	N		0.0, 0x80	RW
54	FINAL_POSITION_VALUE	PV Float (DS-101)	5	D		0.0, 0x80	R
55	FINAL_VALUE_RANGE	Scaling (DS-36)	11	D		100.0, 0.0, 1342 (%), 2	R
56	FINAL_VALUE_2	PV Float (DS-101)	5	D		0.0, 0x80	R
57	FINAL_VALUE_2_UNIT	Unsigned16	2	D		1342 (%)	R
58	FINAL_VALUE_D	PV Discrete (DS-102)	2	N		0.0, 0x80	RW
59	FINAL_POSITION_VALUE_D	PV Discrete (DS-102)	2	D		0.0, 0x80	R
60	FINAL_VALUE_D_2	PV Discrete (DS-102)	2	D		0.0, 0x80	R

6.3.4.2 HART_TB parameters

Index	Parameter Name	Data Type (Simple Var) Data Structure (Record)	Size	Store	Valid Range	Initial Value	Access
16	Block-Header	Block Structure (DS-100)	62	S			RW
17	ST_REV	Unsigned16	2	N		0	R
18	TAG_DESC	Octet String	32	S		Blanks	RW
19	STRATEGY	Unsigned16	2	S		0	RW
20	ALERT_KEY	Unsigned8	1	S	1-255	0	RW
21	TARGET_MODE	Unsigned8	1	S			RW
22	MODE_BLK	Mode Structure	3	D	AUTO, OOS		R
23	ALARM_SUM	Alarm Discrete (DS-72)	13	D			RW
24	PRIMARY_VALUE	PV Float (DS-101)	5	S			R
25	PV_UNIT	Unsigned16	2	S			R
26	SECONDARY_VALUE	PV Float (DS-101)	5	S			R
27	SV_UNIT	Unsigned16	2	S			R
28	HART_DEVICE_IDENT	DS-256	9	S			R
29	HART_DEVICE_INFO	DS-259	18	S			R
30	HART_PV_INFO	DS-258	18	S			R
31	HART_TAG_DESC_DATE	DS-257	27	S			RW
32	HART_ADDITIONAL_DEV_STATUS	Array of 4 BitString(1)	4	S			R

6.3.5 Operation

This chapter explains the various channels available in the startHART device. First a list of all supported channels is given to provide an overview of the existing channels. These channels are described in more detail based on the functionality. Here it is distinguished between variables mapping process values from a HART device to FF/PA and the process values generated in the DEMO_TB transducer block. The channels are available for FF and PA.

FF Channel No.	PA Channel No.	Process Variable	Transducer	Function Block Type	Description
1	0x0208	PRIMARY_VALUE	HART_TB	Analog Input	HART Primary Value
2	0x020A	SECONDARY_VALUE	HART_TB	Analog Input	HART Secondary Value
3	0x010F	RECT_VALUE	DEMO_TB	Analog Input	Rectangle
4	0x010A	RAMP_VALUE	DEMO_TB	Analog Input	Ramp
5	0x0108	CONST_VALUE	DEMO_TB	Analog Input	Constant Value
6	0x0114	TEMP_VALUE	DEMO_TB	Analog Input	Temperature
7	0x0116	ANALOG_INPUT_1_VALUE	DEMO_TB	Analog Input	Mirroring Channel
8	0x0118	ANALOG_INPUT_2_VALUE	DEMO_TB	Analog Input	Mirroring Channel
11	0x011F	RECT_VALUE_D	DEMO_TB	Discrete Input	Rectangle-Discrete
12	0x011B	RAMP_VALUE_D	DEMO_TB	Discrete Input	Ramp-Discrete
13	0x011A	CONST_VALUE_D	DEMO_TB	Discrete Input	Constant-Discrete
14	0x0123	DISCRETE_INPUT_1_VALUE	DEMO_TB	Discrete Input	Mirroring Channel
15	0x0124	DISCRETE_INPUT_2_VALUE	DEMO_TB	Discrete Input	Mirroring Channel
21	0x0125	FINAL_VALUE	DEMO_TB	Analog Output	Analog Output
22	0x0128	FINAL_VALUE_2	DEMO_TB	Analog Output	Analog Output
31	0x012A	FINAL_VALUE_D	DEMO_TB	Discrete Output	Discrete Output
32	0x012C	FINAL_VALUE_2_D	DEMO_TB	Discrete Output	Discrete Output

Note

The PA the channel number to be used is a Unsigned 16 value which is 0x100 * (*TransducerBlockNumber*) + *RelativeParamIndex*.
TransducerBlockNumber: DEMO_TB = 1, HART_TB = 2.
The *RelativeParamIndex* is the relative index of the parameter given in the parameter table fo the PA transducer block in chapters DEMO_TB parameters [58] and HART_TB parameters [60], decremented by 16 (which is the offset of the first parameter within the block). For example *CONST_VALUE* has the relative index 24 in DEMO_TB. The resulting channel number is 0x0108.

6.3.5.1 Process value generation

This chapter describes the various channels that are based on parameters of the DEMO_TB transducer block. The description explains which variables are used as process values, the related unit variables for analog values and for output channels the feedback information via a readback parameter.

The DEMO_TB transducer block is executed cyclically every 1000 ms. So new values of the various process variables are calculated in this interval.

Temperature Channel

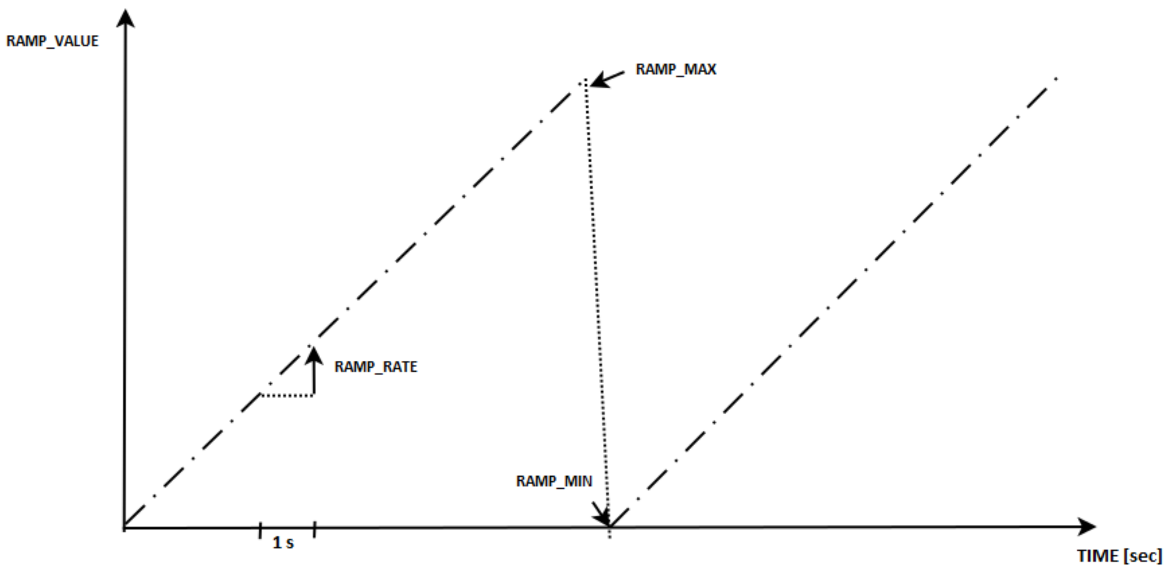
The commModule uses a Renesas RX64M processor. It contains an internal temperature sensor that allows measuring the internal temperature of the chip. This temperature is measured cyclically in the background. When the DEMO_TB transducer block is executed the chip temperature value is copied to parameter TEMP_VALUE. The value is based on the unit given in parameter TEMP_UNIT which can have the values °Celsius, Kelvin and °Fahrenheit. It is possible to change the unit by writing to TEMP_UNIT.

Constant Value Channel

This channel provides always a constant analog value that can be accessed via parameter CONST_VALUE. The value can be changed by writing to the parameter. Any value is permitted. The unit of the value is given in parameter CONST_UNIT. It is fixed to % and cannot be changed.

Ramp Channel

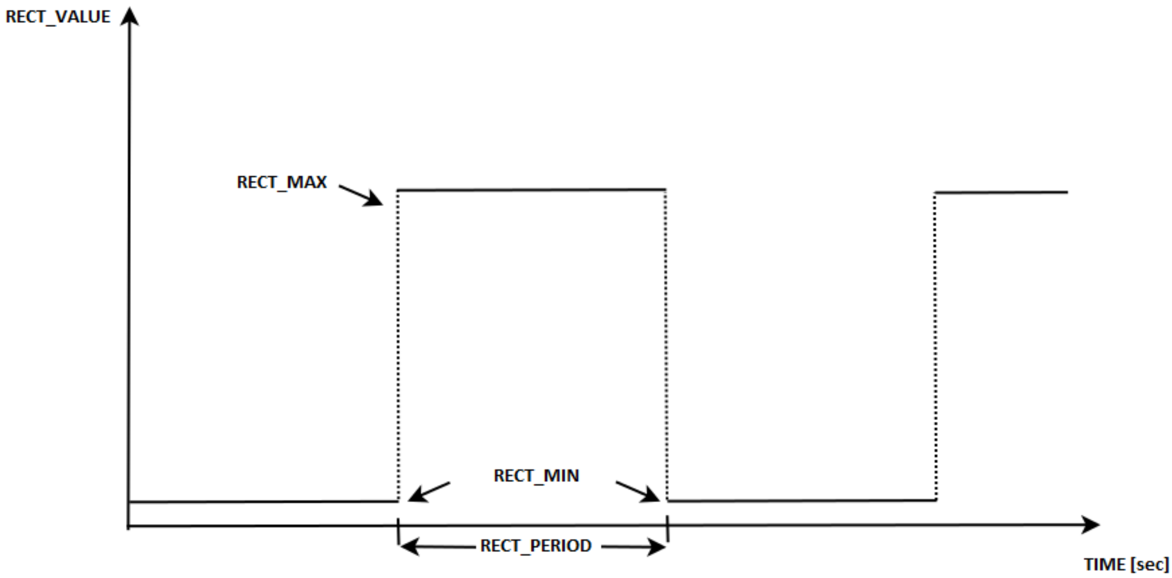
This channel provides a value that is changed upon the execution of the DEMO_TB transducer block i.e. every second. The value starts from a low value given by the value in parameter RAMP_MIN and is incremented in steps given by RAMP_RATE until the value of RAMP_MAX is reached or exceeded. The unit for the process value is given via RAMP_UNIT. It is set to % and cannot be modified. This is shown in the following diagram.



The values that impact the ramp (RAMP_MIN, RAMP_MAX, RAMP_RATE) can be changed by writing to the parameters. No checks are performed any values are permitted. When a parameter is changed the ramp is not restarted. It will continue using the new value in the next transducer block execution.

Rectangle Channel

The channel designated as rectangle channel provides two different output values that change in a given interval. The interval is determined by parameter **RECT_PERIOD**. The output values are given by parameter **RECT_MIN** and **RECT_MAX**. The unit of the process value is given by the value of the read-only parameter **RECT_UNIT.** The behavior is shown in the following picture.



The values that impact the rectangle can be changed by writing to the parameters. No checks on parameter values are performed. The current period will continue with the old interval value and will change in the next cycle.

Constant-Value Discrete Channel

This channel provides a constant discrete value in parameter CONST_ VALUE_D. It can be changed by writing to the parameter.

Ramp-Discrete Channel

This is similar to the analog ramp channel providing discrete outputs. The current ramp value is given by parameter RAMP_VALUE_D. The start and end of the ramp is defined by the values given in RAMP_MIN_D and RAMP_MAX_D. The increment per second is defined via RAMP_RATE_D. The values of the parameters influencing the ramp can be changed.

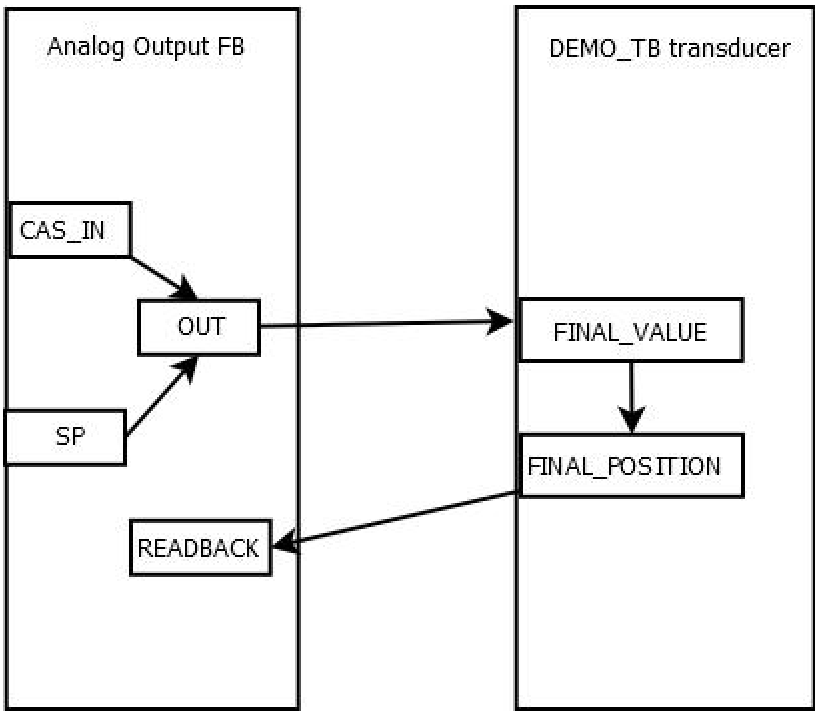
Rectangle-Discrete Channel

This is similar to the analog rectangle channel. The channel provides two different discrete values within the period RECT_PERIOD_D. The value is represented by parameter RECT_VALUE_D. The two discrete values are represented by parameters RECT_MIN_D and RECT_MAX_D. RECT_PERIOD_D, RECT_MIN_D and RECT_MAX_D can be altered to change RECT_VALUE_D. However, if one of these values is changed, the new process value is not executed immediately but only after the old RECT_MAX_D value has expired.

Analog Output Channels

The DEMO_TB transducer block has two parameters that can be used as the process values for the Analog Output function blocks. These are the parameters FINAL_VALUE and FINAL_VALUE_2. The value calculated by the AO function block will be set to these parameters. For FINAL_VALUE a feedback is simulated via parameter FINAL_POSITION_VALUE that is used as read-back for an AO function block (channel 21). For FINAL_VALUE_2 a scaling parameter is defined via FINAL_VALUE_2_UNIT.

The behavior using the channel with read-back value is shown below:

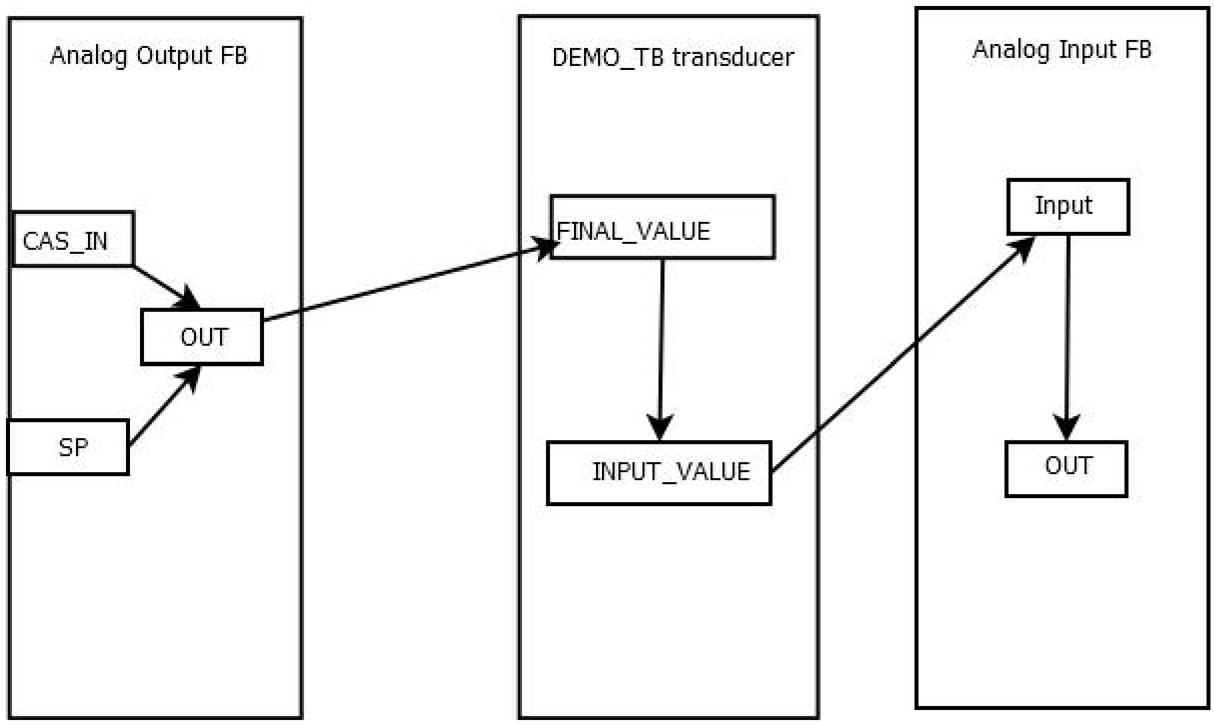


Discrete Output Channels

The DEMO_TB transducer block has two parameters that can be used as the process values for the Discrete Output function blocks. These are the parameters FINAL_VALUE_D and FINAL_VALUE_D_2. The value calculated by the DO function block will be set to these parameters. For FINAL_VALUE_D a feedback is simulated via parameter FINAL_POSITION_VALUE_D that is used as read-back for a DO function block.

Mirroring Channels

As described above the output values calculated by the analog and discrete output blocks will be set in the FINAL_VALUE(_2) resp. FINAL_VALUE(_D_2) parameters of the transducer blocks. The mirroring channels permit to map these values to other values that can be used as input values for analog or discrete input blocks as shown below.



The following table shows which parameters can be mapped:

Output final value	Resulting input
FINAL_VALUE	ANALOG_INPUT_1_VALUE
FINAL_VALUE_2	ANALOG_INPUT_2_VALUE
FINAL_VALUE_D	DISCRETE_INPUT_1_VALUE
FINAL_VALUE_D_2	DISCRETE_INPUT_2_VALUE

6.3.5.2 HART mapping

The commModule permits to connect a HART slave via the UART connection as described in commModule Evaluation Kit documentation, chapter 'External Interfaces'. The commModule in the startHART device is configured to use 26 mA input current from the fieldbus which is the maximum possible value. A HART slave can be powered via the 6.2 V or the 3.15 V output of the evaluation kit. Details for the serial communication between HART slave and commModule is given below but first the logical mapping of parameters of the HART slave to FF/PA parameters is described. These parameters reside within the HART_TB transducer block. Every value is read cyclically from the HART slave in the background. Two different cycles are used. One for all process value related parameters the other for all other parameters that are mapped. This is done in order to have a faster update rate for process values. The cycle times depend on the number of parameters and the transfer speed i.e. the UART baud rate. Consequently, to every FF/PA read request a response can be sent immediately without having to wait until the HART command for the parameter is sent.

The HART commands used in the startHART device are universal commands and should be supported by all HART slaves. The used commands are:

- Command 0
- Command 3
- Command 13
- Command 14
- Command 15
- Command 18
- Command 48

HART Process Values

The HART_TB parameters PRIMARY_VALUE and SECONDARY_VALUE can be used as process values provided by the HART slave. The values are provided by using Command 3. The related process value units are in parameters PV_UNIT for the primary value and SV_UNIT for the secondary value. These are also transmitted via CMD 3. The FF/PA parameters cannot be written. The values are always forwarded from the HART slave. Therefore, it might be necessary to adapt the unit in the XD_SCALE of an FF Analog Input block according to the value provided by the HART slave to avoid a configuration error when the HART process values are used in an AI block.

Other HART Parameters

The HART_TB has several other parameters which are mapped to parameters of the HART slave. The following table shows the HART commands that are used for reading/writing the parameters. Most of the parameters are read-only. One parameter can be written. The HART write command is sent when the FF/PA write has been received after the current HART read command is finished (after it has received a response from the HART slave or has timed out).

Parameter	HART Read Command	HART Write Command
HART_DEVICE_IDENT	0	-
HART_TAG_DESC_DATE	13	18
HART_PV_INFO	14	-
HART_DEVICE_INFO	15	-
HART_ADDITIONAL_DEV_STATUS	48	-

Note

HART command 48 is always used as conditions are mapped to bits in the command 48 response. So even if no explicit mapping would be done command 48 would be used.

HART Communication Parameters

The implementation of the HART master uses command 0 to check for responses from HART slaves trying different baud rates. The supported baud rates are 1200, 2400, 4800, 9600, 19200, 38400, 57600. The other UART attributes that are used:

- Parity: Odd
- Number of Stop Bits: 1
- Use RTS signal: Yes
- RTS active level: High
- RTS Setup time: 0 [bit time]

Note

If the HART slave requires no hardware handshake it still can be connected to the commModule evaluation kit. If the HART slaves requires 6V and RTS handshake the 6-pin connector of the evaluation kit cannot be used directly. See the commModule Evaluation Kit documentation for more information. If other UART parameters are changed, the commScript of the startHART device must also be modified. This requires commScripter and a flashing interface as described in the commScripter user manual.

7 Glossary

AI	Analog input block
ALM	Alarm
AO	Analog output block
AUTO	Automatic mode
CAS	Cascade mode
DCS	Distributed control system
DP	Decentralised Peripherals (PROFIBUS)
DS	Data structure
EU	Engineering unit
FB	Function block
FBS	Function Block Shell
FD	Field Device
FF	Fieldbus Foundation
IA	Initialization acknowledge
IMAN	Initialization manual mode
IR	Initialization request
LIM	Limit
LO	Local override mode
MAN	Manual mode
NI	Not invited
MIB	Management Information Base
OOS	Out of service mode
PA	Process Automation
PB	Physical Block (valid for PA device)
PID	Proportional integral derivative block
PLC	Programmable logic controller
RB	Resource Block (valid for FF device)
RCAS	Remote cascade mode
ROUT	Remote output mode
SCP	Standardized Connection Point
TB	Transducer Block
TOT	Totalization
VCR	Virtual Communication Relationship

This page is intentionally left blank.

Softing Industrial Automation GmbH

Richard-Reitzner-Allee 6
85540 Haar / Germany
https://industrial.softing.com

+ 49 89 45 656-340

info.automation@softing.com