Manual

05/02 MN05010005Z-EN
replaces 05/02 AWB2786-1454GB

Application Software
Motion Control Toolbox

Reference Guide for Function Blocks

Powering Business Worldwide

Emergency On Call Service
Please call your local representative:
http://www.eaton.com/moeller/aftersales
or
Hotline After Sales Service:
+49 (0) 180 5 223822 (de, en)
AfterSalesEGBonn@eaton.com

Original Operating Instructions
The German-language edition of this document is the original operating manual.

Translation of the original operating manual
All editions of this document other than those in German language are translations of the original German manual.

1st published 2002, edition date 05/02

Production: Thomas Kracht
Translation: globaldocs GmbH

Contents

1 Basic Information and Technical Data

General

Tool for IEC/EN 61131-3 programming with easySoft-CoDeSys

The Motion Control Toolbox provides a flexible means of implementing both remote and local positioning tasks. Its tailored integration capability allow movements to be optimally adapted to each automation task.

The Motion Control Toolbox can be imported into easySoft-CoDeSys in the form of a library.

What can the Motion Control Toolbox be used for?

- Ansynchronous point-to-point axis movements
- Electrical and hydraulic axes
- Positioning processes whose actual and setpoint positions have a high resolution and a prefix (approx. –1 000 000 000 to 1 000 000 000)
- Controlled acceleration and deceleration ramps in positive and negative directions can be assigned as required.
- Controlled rotational or linear speeds with contouring error monitoring
- For positioning processes with differing power requirements, position controllers are available.
- The function block-interface link with the sensor systems and the actuators can be established with various hardware combinations. For “slow” positioning tasks (where no low cycle times are required), variants using the fieldbus systems can also be used in combination with the remote expansions.
- Positioning rotary axes with optimized traversing paths via the zero point
- Typical camshaft applications
- Incremental coordinate positioning

- Master-slave positioning
  - Interpolation
  - Path control
- Electronic gearing
  - User-selectable transmission ratio
  - "Internal", "external" or "virtual" master

What can the Motion Control Toolbox not be used for?

Positioning tasks with very short cycle times cannot be implemented with the Motion Control Toolbox.

For which PLCs can the Motion Control Toolbox be used?

The Motion Control Toolbox can be used with all PLCs which can be programmed with easySoft-CoDeSys, i.e. the following types:

- XC-CPU600
- XC-CPU400
- XC-CPU200
- XC-CPU100

Most positioning tasks require extremely short PLC cycle times. The "fast" "XC-CPU600", "XC-CPU400" and "XC-CPU200" PLCs are therefore the most common choice.

Descriptive variable and function block names

The variables and function blocks of the Motion Control Toolbox have been given descriptive, plain-text names to make the toolbox as easy as possible to use. Most function blocks can be incorporated in the user program and parameterized without reference to the documentation. The names of the variables contain all the information required for defining the parameters (name, units and data type).

Example:
Setpoint_position_DINT
Deceleration_time_negative_ms_UINT

Parameterizing replaces programming

The Motion Control Toolbox removes the need for programming. Instead, programs are created by parameterizing, which significantly reduces the time needed to generate programs.

Fast integration using libraries

Using libraries, function blocks can be quickly integrated in easySoft-CoDeSys.

Technical data and other technical information

General information about code sizes and the cycle time demand of the Motion Control Toolbox function blocks

Code is generated only once for each declared function block. For further instances, (i.e. declaration of the same function block with different instance name), only an additional data field is created. Due to the hierarchic modular structure of the Motion Control Toolbox, calling even a single function block can result in large volumes of code, data and instances.

Single and multiple instancing

If a function block does not have to save data for the next call, it is sufficient to instance it only once (i.e. assign only one instance name in the declaration section). This applies, for example, to the following function blocks of the Motion Control Toolbox:

- P_UINT_rapid_traverse_crawl_speed
- P_DINT_rapid_traverse_crawl_speed
- P_AAINT_INT_analog_output
- P_IP2_DINT_interpolation
- P_camshaft
- P_hydraulic
- P_referencing

Example:
In sample application “Interpol”, instancing for “two-point interpolation” is used for two different tasks. First, an analog value is scaled to 12 bits, then a percentage value (+/–) is scaled to 14 bits.

Sample application for function block “P_IP2_DINT_interpolation” in program “Interpol”

```
PROGRAM Interpol


VAR
      Two_point_interpolation : P_IP2_DINT_INTERPOLATION ;
      Analog_value_4_till_20_mA_WORD : WORD ;
      Analog_value_12bit_DINT : DINT ;
      Value_percent_DINT : DINT ;
      Value_14bit_DINT : DINT ;
END_VAR
```

```
(*1st call of the interpolation FB*)
(*==========================*)
LD    Analog_value_4_till_20_mA_WORD
WORD_TO_DINT
ST    Two_point_interpolation.X_DINT
CAL Two_point_interpolation(
      Suppress_extrapolation_BOOL :=1,
      X1_DINT :=819,
      X2_DINT :=4095,
      Y1_DINT :=0,
      Y2_DINT :=4095,
      Y_DINT=>Analog_value_12bit_DINT)

(*2nd call of the interpolation FB*)
(*==========================*)
CAL Two_point_interpolation(
      X_DINT :=Value_percent_DINT,
      Suppress_extrapolation_BOOL :=0,
      X1_DINT :=-100,
      X2_DINT :=100,
      Y1_DINT :=0,
      Y2_DINT :=16383,
      Y_DINT=>Value_14bit_DINT)

END_PROGRAM
```

Function blocks that save data until the next call-up (marked with a "*" in the table on page 9), must be instantiated several times. If, for example, function block "P_closed_loop_position_control" is used for several zones, it must be instantiated once for each zone.

This applies, for example, to the following function blocks of the Motion Control Toolbox:

- P_basic_position_control
- P_UINT_characteristics_control
- P_DINT_characteristics_control
- P_UINT_rapid_traverse_crawl_speed
- P_DINT_rapid_traverse_crawl_speed
- P_closed_loop_position_control
- P_Sq10_sequencer
- P_axis_simulation
- P_incremental_encoder_evaluation
- P_ramp_deceleration
- P_hydraulic
- P_referencing

Declaration example for two-axis positioning:

```
VAR
     position_control_axis1 : P_closed_loop_position_control;
     position_control_axis2 : P_closed_loop_position_control;
END_VAR
```

“Instances per call-up“, taking into account any fuction sub-blocks

→ The data should be used only for guidance. For technical reasons, the actual values may deviate from those given here. No guarantee can therefore be given for the correctness of this data.

Table 1: Code size, data size, cycle time demand and "instances per call-up", taking into account function sub-blocks

Function blocks	Function sub-block	Instances
P_AAINT_INT_analog_output	2	2
P_axis_simulation*	20	32
P_basic_position_control*	27	38
P_camshaft	3	3
P_closed_loop_position_control*	35	92
P_CYCS_cycle_time_setpoint_value*	3	3
P_DINT_characteristics_control*	24	29
P_DIN_rapid_traverse_crawl_speed	8	8
P_FM1_BOOL_frequency_measuring*	5	5
P_FM1_UINT_frequency_measuring*	5	5
P_FM10_BOOL_frequency_measuring*	6	44
P_FM10_UINT_frequency_measuring*	6	44
P_FM3_BOOL_frequency_measuring*	6	16
P_FM3_UINT_frequency_measuring*	6	16
P_hydraulic	10	10
P_incremental_encoder_evaluation	3	3
P_IP10_DINT_interpolation	4	6
P_IP2_DINT_interpolation	3	5
P_IP20_DINT_interpolation	4	6
P_IP3_DINT_interpolation	4	6
P_IP4_DINT_interpolation	4	6
P_IP60_DINT_interpolation	6	12
P_load_compensation*	14	34
P_Osc1000_oscilloscope*	3	8
P_Osc255_oscilloscope*	6	8

Function blocks	Function sub-block	Instances
P_PI_controller*	14	18
P_ramp_deceleration*	11	13
P_referencing*	5	6
P_semi_ellipsis	3	3
P_semicircle	3	3
P_Sq10_sequencer*	3	3
P_SYC_syncronisation_controller*	7	12
P_SYCI_syncronisation_controller*	30	84
P_SYTP_syncronisation_times*	4	7
P_UINT_characteristics_control*	22	28
P_UIN_rapid_traverse_crawl_speed	7	7
P_ZSFB01_special_FB*	27	36
P_ZSFB02_special_FB*	31	49
P_ZSFB03_special_FB*	15	19

* The function block saves data until the next call-up.
If the function block is used several times, it must therefore be instantiated several times.

Sample applications

A sample application is listed at the end of each function block description. Sample applications "Pos_01" to "Pos_12" are supplied on the installation disk. The programs "Pos_10", "Pos_11" and "Pos_12" are a combination of axis simulation and the following positioning blocks:

- Pos_10 = Basic positioning and axis simulation and incremental encoder processing
- Pos_11 = Characteristics positioning and axis simulation and incremental encoder processing
- Pos_12 = Closed-loop positioning control and axis simulation

Example:
Sample application "Pos_10" demonstrates how basic positioning can be linked for test purposes with axis simulation and incremental encoder processing. The following functions are assigned to the digital inputs:

- Digital input "0" = Change setpoint position
- Digital input "1" = Activate basic position control
- Digital input "2" = Referencing on rising edge
- Digital input "3" = Activate axis simulation
- Digital input "4" = Accept setpoint positions (automatic mode)

Application of function block "P_basic_position_control" in program "Pos_10"

```
PROGRAM Pos_10
VAR
        basic_position_control_01 : P_basic_position_control ;
        axis_simulation_01 : P_axis_simulation ;
        incremental_encoder_evaluation_01 : P_incremental_encoder_evaluation ;
        DI_0_0_BOOL : BOOL ;
        DI_0_1_BOOL : BOOL ;
        DI_0_2_BOOL : BOOL ;
        DI_0_3_BOOL : BOOL ;
        Setpoint_position_01_DINT : DINT ;
        Actual_position_01_DINT : DINT ;
END_VAR

(*========================================================*)
(*====== setpoint position variation =====================*)
(*========================================================*)
LD      DI_0_0_BOOL
JMPCN   SETPOINT_POSITION_02
        LD    2000
        ST    Setpoint_position_01_DINT
        JMP   E_SETPOINT_POSITION_02
SETPOINT_POSITION_02:
        LD    60000
        ST    Setpoint_position_01_DINT
E_SETPOINT_POSITION_02:
```

```
(*=========================================================*)
(*== incremental encoder evaluation of the simulation =========*)
(*=========================================================*)

LD      axis_simulation_01.incremental_encoderincremental_encoder_Output_UDINT
UDINT_TO_DINT
ST      incremental_encoder_evaluation_01.Increments_DINT
CAL incremental_encoder_evaluation_01(
        Machine_zero_point_DINT :=2000,
        Maximum_incremental_encoder_DINT :=16777215,
        Activate_BOOL :=1,
        Absolute_value_transmitter_BOOL :=1,
        Absolute_value_transmitter_without_referencing_BOOL :=0,
        Accept_machine_zero_point_BOOL :=DI_0_2_BOOL,
        Reference_signal_BOOL :=0,
        Actual_value_DINT=>Actual_position_01_DINT
        )

(*=========================================================*)
(*======= basic position control ========================*)
(*=========================================================*)

CAL basic_position_control_01(
        Setpoint_position_DINT :=Setpoint_position_01_DINT,
        Actual_position_DINT :=Actual_position_01_DINT,
        Activate_BOOL :=DI_0_1_BOOL,
        Manipulated_variable_negation_BOOL :=0,
        Accept_setpoint_position_BOOL :=1,
        Cycle_time_demand_optimize_BOOL :=0,
        Ramp_time_100increase_maximum_ms_UINT :=500,
        Deceleration_position_deviation_UDINT :=30000,
        Rounding_position_deviation_15bit_UINT :=1000,
        Manipulated_variable_max_11Bit_UINT :=1500,
        Tolerance_positioning_zone_UINT :=30
        )
```

```
(*==========================================================*)
(*====== simulation of the axis ===========================*)
(*==========================================================*)

LD      basic_position_control_01.Manipulated_variable_12Bit_INT
ST      axis_simulation_01.Manipulated_variable_12Bit_INT
CAL axis_simulation_01(
        Activate_BOOL :=DI_0_3_BOOL,
        Accept_manual_value_BOOL :=0,
        Nominal_revolutions_per_minute_INT :=2000,
        Manual_value_UDINT :=0,
        Increments_per_revolution_UINT :=100
        )

END_PROGRAM
```

2 The Principle of Drive Positioning

The position controller is part of the position control circuit (→ fig. 1). The variable to be controlled – the actual position (s_{actual}) of a mechanical moving unit – is read and compared to the reference input variable (desired path s_{set}). From the difference between the actual position and the control variable, the Contouring error (Δs) is calculated ($\Delta s = s_{actual} - s_{set}$). In dependence of the current Contouring error (Δs), a speed reference value (Δ_{set}) is sent to the servo drive in the form of an analog voltage signal. Function block "P_closed_loop_position_control" calculates the reference input variable from the defined target position (setpoint position) and the specified speed of each axis.

The drive's internal speed control compares the actual speed (Δ_{actual}) with the speed reference value (Δ_{set}) and energizes the current regulator accordingly.

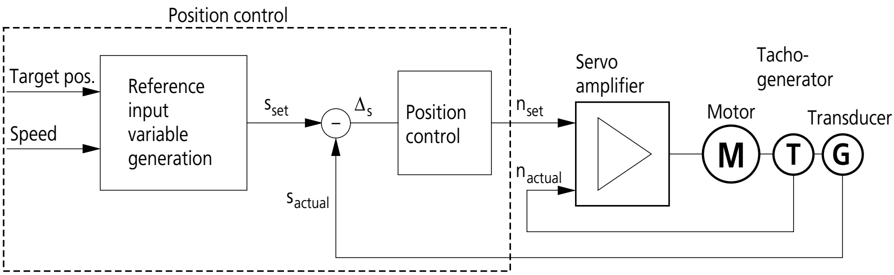


Figure 1: The position control circuit

Generating reference input variables

The reference input variable is calculated in each cycle depending on the maximum speed reached (V_{max}) and the target position. This results in continuous movement of the drive. The resulting axis movement is split into three phases of motion:

- Acceleration phase
- Constant speed phase
- Deceleration phase

The changeover between each phase depends on the drive's specified acceleration time (t_h) to its nominal speed. The generation of reference input variables ensures constant acceleration and deceleration in the respective phases, resulting in a symmetric speed ramp profile. An example of the path, speed and acceleration profile is shown in figure 2.

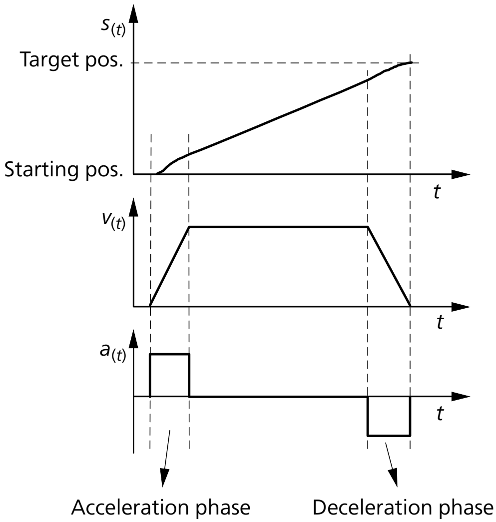


Figure 2: Axis movement as a function of time

The generation of reference input variables allows the current traversing speed of an axis to be changed at any time. The movement is then accelerated or decelerated according to the drive's current speed (→ fig. 3). The speed is specified in percent of the defined nominal speed.

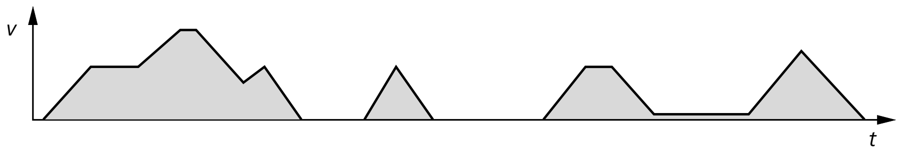


Figure 3: Examples of speed as a function of time

Axis movement to the target position

Servo axes have a finite dynamic, i.e. they have a limited rate of acceleration. During a movement, the axis lags behind the calculated setpoint position. The difference between the setpoint and actual position is called contouring error. The contouring error depends on the axis dynamic and the gain of the position controller. Positioning is completed when the actual position lies inside the in-position zone. The in-position zone reflects the tolerance with which an approached position is declared as having been reached. If the in-position zone is very small, it may take a long time until the axis reaches its exact target position.

Figure 4 shows the effect of the contouring error on the path and speed of an axis. Once the axis reports that it has reached its in-position zone, it is ready to process a new traversing job.

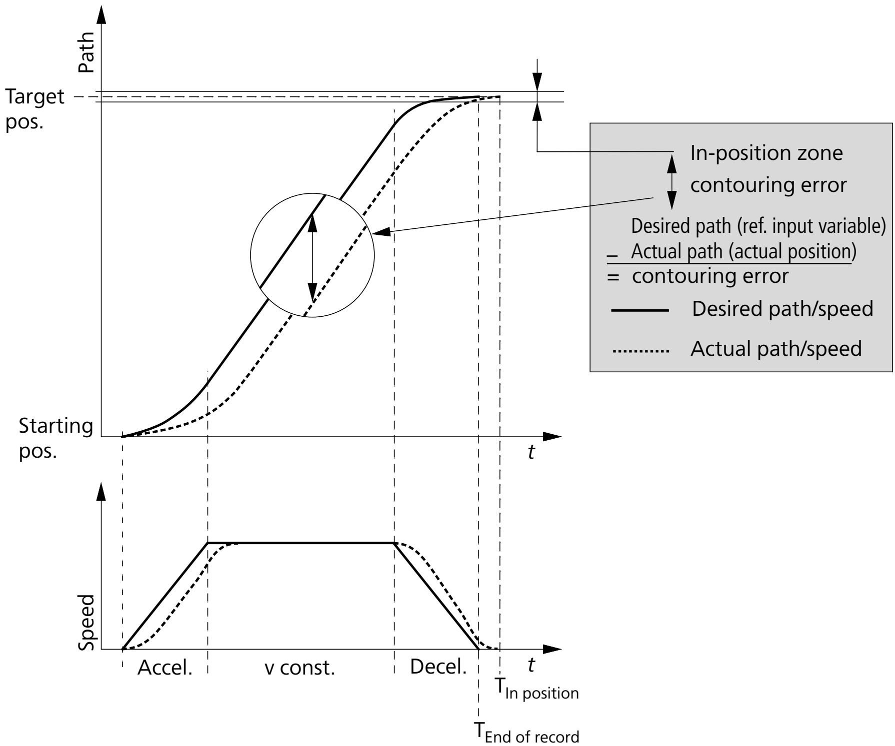


Figure 4: Dynamic axis movement

Positioning procedure with several axes

Function block "P_closed_loop_position_control" implements asynchronous Point-to-point positioning, allowing independent positioning of several axes. An initial synchronization between individual axes can be defined in the program. Initially, the axis accelerates at its maximum rate and, if the distance to be covered is long enough, traverses at maximum speed. As it approaches the target point, it decelerates.

If positioning of several axes is started at the same time, this means that the axes usually reach their target at different times (→ fig. 5), depending on

- the distance to be traversed,
- the specified acceleration time to nominal speed,
- the current maximum speed values.

Figure 6 shows an example of an asynchronous positioning process. The status signal contains corresponding bits for issuing commands to the axes.

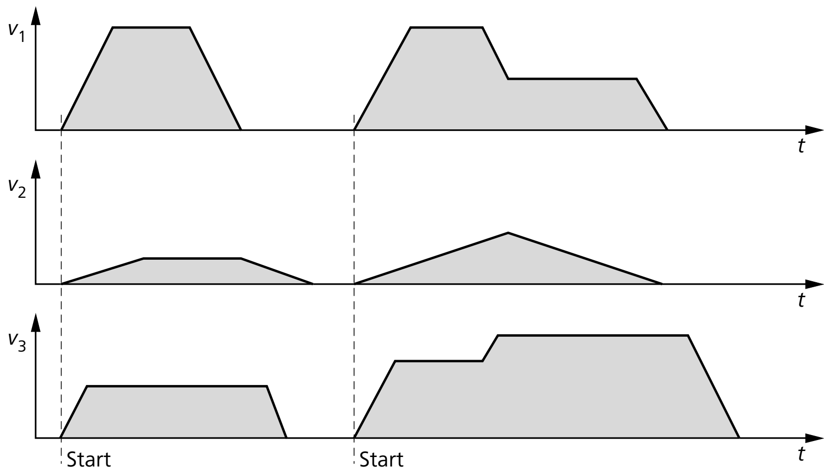


Figure 5: Point-to-point positioning with three axes, synchronous start

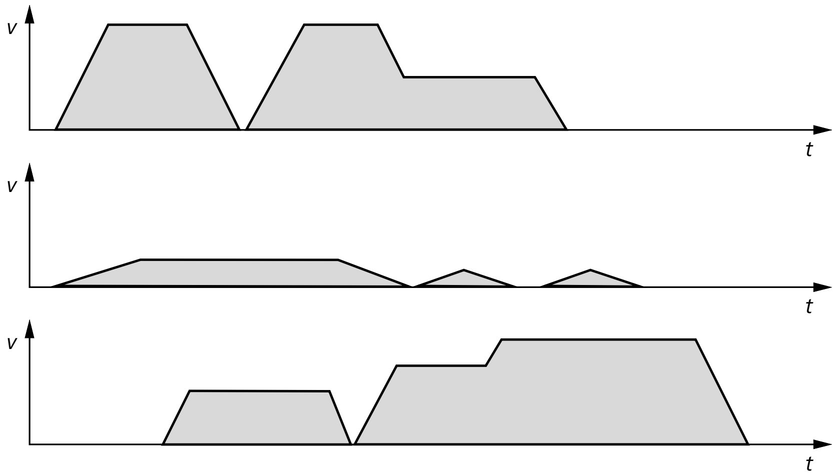
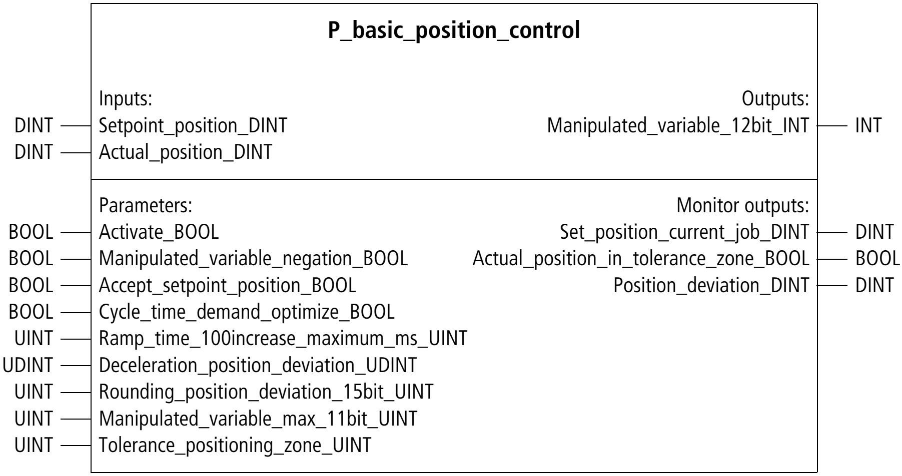


Figure 6: Asynchronous point-to-point positioning with three axes

3 Position Controller

Basic position control

P_basic_position_control
Basic position control



Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Setpoint_position_DINT	Setpoint position (target position)	-10^9 to 10^9
Actual_position_DINT	Actual position	-10^9 to 10^9
Parameters		
Activate_BOOL	Activate (deactivate = reset) the function block	0/1
Manipulated_variable_ negation_BOOL	Negate manipulated variable (reverse prefix)	0/1
Accept_setpoint_ position_BOOL	Control mode: Accept applied setpoint positions (automatic mode). This mode overrides manual mode.	0/1
Cycle_time_demand_ optimize_BOOL	Control mode: The cycle time demand of the function block can be optimized at the expense of the rounding transition at the end of the positioning process	0/1
Ramp_time_100increase_ maximum_ms_UINT	Ramp time required at a setpoint position change for an increase of 100 increments	0 to 65535
Deceleration_position_ deviation_UDINT	Position deviation from which a ramped delay is applied to the manipulated variable	0 to 10^9
Rounding_position_deviation_ 15bit_UINT	Position deviation from which rounding of the manipulated variable delay begins (→ fig. 7)	0 to 32767
Manipulated_variable_max_ 11bit_UINT	Greatest manipulated variable	0 to 2048
Tolerance_positioning_ zone_UINT	Tolerance range (±) of the positioning zone	0 to 32767
Outputs		
Manipulated_variable_ 12bit_INT	Manipulated variable for controlling the axis actuators (setpoint rotational speed, setpoint linear speed)	–2048 to 2047

Designation	Function	Value range
Monitor outputs		
Set_position_current_job_DINT	Setpoint position of current job	-10^9 to 10^9
Actual_position_in_tolerance_zone_BOOL	Status: Actual position is within tolerance zone	0/1
Position_deviation_DINT	Position deviation	-10^9 to 10^9

Caution!
The positioning hardware must be designed so that it cannot be damaged due to an incorrect start-up. Position end switches and a foot and palm switch must therefore be fitted for disconnecting the power supply. Incorrect starting can result in a sudden acceleration of the positioning axes at their maximum rate. During a start-up, approach the moving machine parts with caution.

Description

This function block provides a simple means of implementing positioning with acceleration and deceleration ramps. The manipulated variable output begins as soon as the function block is activated. Deactivating the function block results in its reset and a manipulated variable of "0". When the value "1" is assigned to parameter "Accept_setpoint_position_BOOL", the pending setpoint positions are accepted. If "Accept_setpoint_position_BOOL" is changed from "1" to "0", deceleration takes place with a delay and rounding. For referencing the actual position, the following function blocks can be used:

- "P_incremental_encoder_evaluation" (→ application example on page 27) or
- "P_referencing" (for automatic referencing with reference cam searching, → page 209).

→ If the actual positions drift away from the setpoint positions after the function block is activated, the most likely cause is the transducer’s direction of rotation. This can be corrected by changing the prefix of the manipulated variable with “Manipulated_variable_negation_BOOL =1”.

If parameter “Cycle_time_demand_optimize_BOOL” is set to “1”, the function block’s maximum cycle time demand is lowered by about 0.3 ms. The transition to the manipulated variable rounding at the end of the positioning procedure may then be slightly coarser.

Parameter “Manipulated_variable_max_11bit_UINT” is used to specify the maximum manipulated variable value that can be reached after completion of the acceleration ramp. The rotary or linear axis speeds are normally proportional to this value, with which the actuators are controlled.

An acceleration ramp can be defined with “Ramp_time_100increase_maximum_ms_UINT”. Within the entered ramp time, the manipulated variable can ramp up by up to 100 increments.

Example:
A ramp time “Ramp_time_100increase_maximum_ms_UINT = 50” was specified. This results in an increase of the manipulated variable (after the setpoint position step) by 100 increments every 50 ms. An increase from 0 to 2000 (≈ maximum manipulated variable) at this ramp time would therefore take 1000 ms.

Parameter "Deceleration_position_deviation_UDINT" can be used to specify that the manipulated variable ramps down when the position deviation falls below this value. To minimize the time taken by the positioning process, this "deceleration ramp starting value" should be kept as low as possible.

For small position deviations or at standstill, a steep deceleration ramp can result in vibration. To prevent this, the ramps can be rounded near the setpoint position (target position) with parameter "Rounding_position_deviation_15bit_UINT".

→ At standstill (setpoint position = actual position) or in case of slight axis vibrations, parameter "Rounding_position_deviation_15bit_UINT" can best be optimized. Observe the manipulated variable at small position deviations:

If the manipulated variables are too large (resulting in axis vibration), increase parameter "Rounding_position_deviation_15bit_UINT".

If the manipulated variables are too small (resulting in excessive steady-state position deviations), reduce parameter "Rounding_position_deviation_15bit_UINT".

The general rule is:
For a fast deceleration phase, the selected delay position deviation should be about 10 to 20 times greater than the rounding position deviation (→ fig. 7) and about 20 to 100 times greater for slow deceleration.

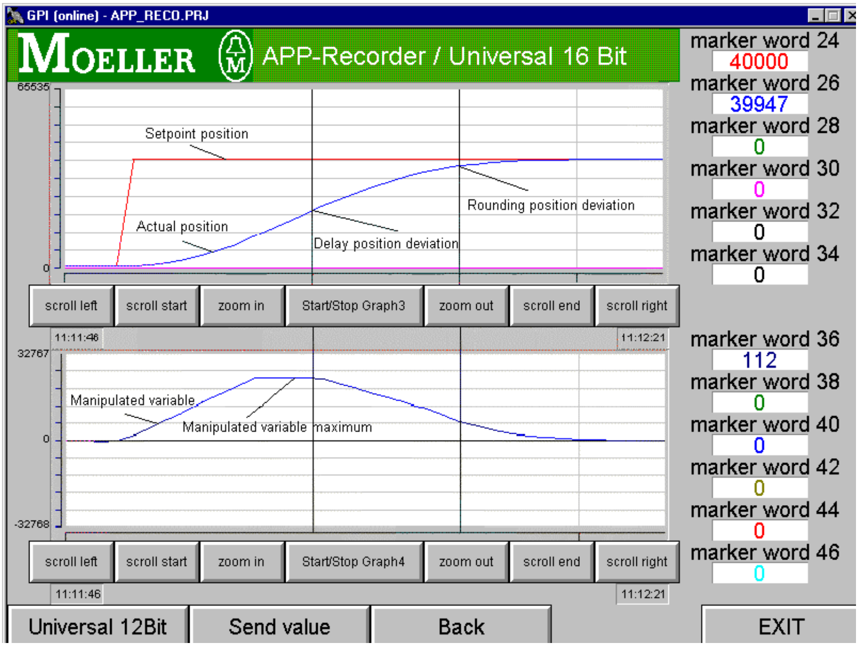


Figure 7: Positioning with ramp deceleration and end rounding of the manipulated variable

A tolerance position zone (setpoint position and tolerance value) can be defined. If the actual position is within this positioning zone, monitor output "Actual_position_in_tolerance_zone_BOOL" carries a "1". The monitor output is provided by the setpoint position of the current positioning job.

→ To start up slow axis movements, start with a low "maximum manipulated variable" and high acceleration and deceleration ramps. Start, for example with the following values:

- Ramp_time_100increase_maximum_ms_UINT = 200
- Deceleration_position_deviation_UDINT = 30 % of the traverse path
- Rounding_position_deviation_15bit_UINT = 3 % of the traverse path
- Manipulated_variable_max_11bit_UINT = 300

Example:
In this application example, two incremental encoder values are recorded. With function block "P_incremental_encoder_evaluation", the incremental values are processed in such a way as to prevent a data over-range. The zero points (referencing) are specified on a rising edge of digital inputs "2" and "3". With digital input "0", the setpoint position can be changed. Digital input "1" activates the basic positioning block.

Application of function block "P_basic_position_control" in program "Pos_01"

```
PROGRAM Pos_01
VAR
        incremental_encoder_evaluation_01 : P_incremental_encoder_evaluation ;
        incremental_encoder_evaluation_02 : P_incremental_encoder_evaluation ;
        axis_01 : P_basic_position_control ;
        axis_02 : P_basic_position_control ;
        AAINT_INT_analog_output : P_AAINT_INT_analog_output ;
        DI_0_0_BOOL : BOOL ;
        DI_0_1_BOOL : BOOL ;
        DI_0_2_BOOL : BOOL ;
        DI_0_3_BOOL : BOOL ;
```

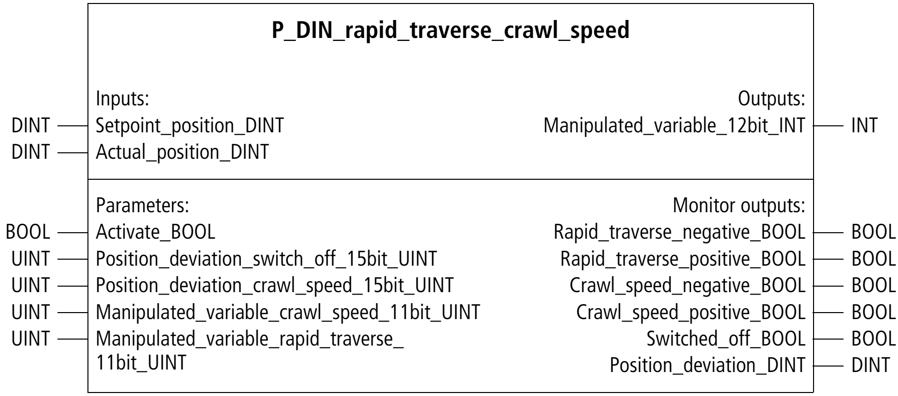
```
        AO_0_0_2_0 : INT ;
        AO_0_0_2_2 : INT ;
        Setpoint_position_01_DINT : DINT ;
        Setpoint_position_02_DINT : DINT ;
        Actual_position_01_DINT : DINT ;
        Actual_position_02_DINT : DINT ;
        incremental_encoder_01 : UDINT ;
        incremental_encoder_02 : UDINT ;
END_VAR
LD      DI_0_0_BOOL
JMPCN   SETPOINT_POSITION_02
        LD    5000
        ST    Setpoint_position_01_DINT
        LD    20000
        ST    Setpoint_position_02_DINT
        JMP   E_SETPOINT_POSITION_02
SETPOINT_POSITION_02:
        LD    200000
        ST    Setpoint_position_01_DINT
        LD    80000
        ST    Setpoint_position_02_DINT
E_SETPOINT_POSITION_02:

LD      incremental_encoder_01
UDINT_TO_DINT
ST      incremental_encoder_evaluation_01.Increments_DINT
```

```
CAL incremental_encoder_evaluation_01(
        Machine_zero_point_DINT :=2000,
        Maximum_incremental_encoder_DINT :=16777215,
        Activate_BOOL :=1,
        Absolute_value_transmitter_BOOL :=1,
        Absolute_value_transmitter_without_referencing_BOOL :=0,
        Accept_machine_zero_point_BOOL :=DI_0_2_BOOL,
        Reference_signal_BOOL :=0,
        Actual_value_DINT=>Actual_position_01_DINT
        )

CAL axis_01(
        Setpoint_position_DINT :=Setpoint_position_01_DINT,
        Actual_position_DINT :=Actual_position_01_DINT,
        Activate_BOOL :=DI_0_1_BOOL,
        Manipulated_variable_negation_BOOL :=0,
        Accept_setpoint_position_BOOL :=1,
        Cycle_time_demand_optimize_BOOL :=0,
        Ramp_time_100increase_maximum_ms_UINT :=50,
        Deceleration_position_deviation_UDINT :=20000,
        Rounding_position_deviation_15bit_UINT :=2000,
        Manipulated_variable_max_11Bit_UINT :=1500,
        Tolerance_positioning_zone_UINT :=20
        )

CAL AAINT_INT_analog_output(
        Input_value_INT :=axis_01.Manipulated_variable_12Bit_INT,
        |
        Analog_output_INT=>AO_0_0_2_0
)

LD      incremental_encoder_02
UDINT_TO_DINT
ST           incremental_encoder_evaluation_02.Increments_DINT
```

```
CAL incremental_encoder_evaluation_02(
        Machine_zero_point_DINT :=2000,
        Maximum_incremental_encoder_DINT :=16777215,
        Activate_BOOL :=1,
        Absolute_value_transmitter_BOOL :=1,
        Absolute_value_transmitter_without_referencing_BOOL :=0,
        Accept_machine_zero_point_BOOL :=DI_0_3_BOOL,
        Reference_signal_BOOL :=0,
        Actual_value_DINT=>Actual_position_02_DINT
        )

CAL axis_02(
        Setpoint_position_DINT :=Setpoint_position_02_DINT,
        Actual_position_DINT :=Actual_position_02_DINT,
        Activate_BOOL :=DI_0_1_BOOL,
        Manipulated_variable_negation_BOOL :=0,
        Accept_setpoint_position_BOOL :=1,
        Cycle_time_demand_optimize_BOOL :=0,
        Ramp_time_100increase_maximum_ms_UINT :=50,
        Deceleration_position_deviation_UDINT :=20000,
        Rounding_position_deviation_15bit_UINT :=2000,
        Manipulated_variable_max_11Bit_UINT :=1500,
        Tolerance_positioning_zone_UINT :=20
        )

CAL AAINT_INT_analog_output(
        Input_value_INT :=axis_02.Manipulated_variable_12Bit_INT,
        Analog_output_INT=>AO_0_0_2_2
)

END_PROGRAM
```

Rapid traverse crawl speed

P_DIN_rapid_traverse_crawl_speed Rapid crawl speed with double integer setpoint and actual positions



Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Setpoint_position_DINT	Setpoint position (target position)	-10^9 to 10^9
Actual_position_DINT	Actual position	-10^9 to 10^9
Parameters		
Activate_BOOL	Activate (deactivate = reset) the function block	0/1
Position_deviation_switch_off_15bit_UINT	If this position deviation is breached, axis movement is stopped (manipulated variable = 0)	0 to 32 767
Position_deviation_crawl_speed_15bit_UINT	If this position deviation is breached, axis movement changes from rapid traverse to crawl speed	0 to 32 767

Designation	Function	Value range
Manipulated_variable_crawl_speed_ 11bit_UINT	Manipulated variable of crawl speed	0 to 2048
Manipulated_variable_rapid_ traverse_11bit_UINT	Manipulated variable of rapid traverse	0 to 2048
Outputs		
Manipulated_variable_12bit_INT	Manipulated variable for controlling the axis actuators (setpoint rotational speed, setpoint linear speed)	–2048 to 2047
Monitor outputs		
Rapid_traverse_negative_BOOL	Status: negative rapid	0/1
Rapid_traverse_positive_BOOL	Status: positive rapid	0/1
Crawl_speed_negative_BOOL	Status: negative crawl	0/1
Crawl_speed_positive_BOOL	Status: positive crawl	0/1
Switched_off_BOOL	Status: switched off	0/1
Position_deviation_DINT	Position deviation	-10^9 to 10^9

Caution!
The positioning hardware must be designed so that it cannot be damaged due to an incorrect start-up. Position end switches and a foot and palm switch must therefore be fitted for disconnecting the power supply. Incorrect starting can result in a sudden acceleration of the positioning axes at their maximum rate. During a start-up, approach the moving machine parts with caution.

Description

This function block can be used to implement positioning operations with a crawl speed/rapid traverse combination. Parameter “Position_deviation_crawl_speed_15bit_UINT” can be used to specify the position deviation at which the changeover from “rapid traverse” to “crawl speed” takes place.

With “Position_deviation_switch_off_15Bit_UINT”, the position deviation can be defined at which the axis is switched off (manipulated variable = 0).

The manipulated variable in the function block’s output range must be linked with the actuators. The individual traversing states and the position deviation are available as monitor output variables.

Example:
In this application example, two incremental encoder values are recorded. With the “P_incremental_encoder_ evaluation” function block, the incremental values are processed in such a way as to prevent a data over-range. The zero points (referencing) are specified on a rising edge of digital inputs “2” and “3”. With digital input “0”, the setpoint position can be changed. Digital input “1” activates the function block.

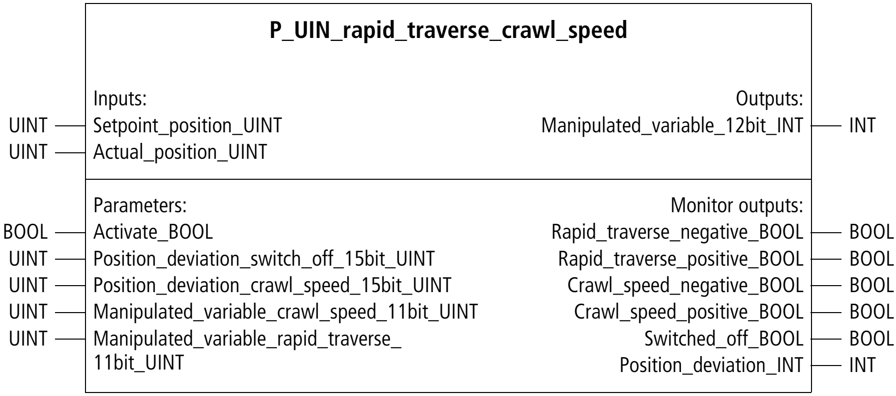
Application of function block "P_DIN_rapid_traverse_crawl_speed" in program "Pos_02"

```
PROGRAM Pos_02
VAR
        incremental_encoder_evaluation_01 : P_incremental_encoder_evaluation ;
        incremental_encoder_evaluation_02 : P_incremental_encoder_evaluation ;
        axis_01 : P_DIN_rapid_traverse_crawl_speed ;
        axis_02 : P_DIN_rapid_traverse_crawl_speed ;
        AAINT_INT_analog_output : P_AAINT_INT_analog_output ;
        DI_0_0_BOOL : BOOL ;
        DI_0_1_BOOL : BOOL ;
        DI_0_2_BOOL : BOOL ;
        DI_0_3_BOOL : BOOL ;
        AO_0_0_2_0 : INT ;
        AO_0_0_2_2 : INT ;
        Setpoint_position_01_DINT : DINT ;
        Setpoint_position_02_DINT : DINT ;
        Actual_position_01_DINT : DINT ;
        Actual_position_02_DINT : DINT ;
        incremental_encoder_01 : UDINT ;
        incremental_encoder_02 : UDINT ;
END_VAR
LD      DI_0_0_BOOL
JMPCN   SETPOINT_POSITION_02
        LD     5000
        ST     Setpoint_position_01_DINT
        LD     20000
        ST     Setpoint_position_02_DINT
        JMP    E_SETPOINT_POSITION_02
```

```
SETPOINT_POSITION_02:
        LD    200000
        ST    Setpoint_position_01_DINT
        LD    80000
        ST    Setpoint_position_02_DINT
E_SETPOINT_POSITION_02:

LD      incremental_encoder_01
UDINT_TO_DINT
ST      incremental_encoder_evaluation_01.Increments_DINT
CAL incremental_encoder_evaluation_01(
        Machine_zero_point_DINT :=2000,
        Maximum_incremental_encoder_DINT :=16777215,
        Activate_BOOL :=1,
        Absolute_value_transmitter_BOOL :=1,
        Absolute_value_transmitter_without_referencing_BOOL :=0,
        Accept_machine_zero_point_BOOL :=DI_0_2_BOOL,
        Reference_signal_BOOL :=0,
        Actual_value_DINT=>Actual_position_01_DINT
        )

CAL axis_01(
        Setpoint_position_DINT :=Setpoint_position_01_DINT,
        Actual_position_DINT :=Actual_position_01_DINT,
        Activate_BOOL :=DI_0_1_BOOL,
        Position_deviation_switch_off_15bit_UINT :=200,
        Position_deviation_crawl_speed_15bit_UINT :=2000,
        Manipulated_variable_crawl_speed_11bit_UINT :=100,
        Manipulated_variable_rapid_traverse_11bit_UINT :=1000
        )
```

```
CAL AAINT_INT_analog_output(
        Input_value_INT :=axis_01.Manipulated_variable_12Bit_INT,
        Analog_output_INT=>AO_0_0_2_0
        )

LD      incremental_encoder_02
UDINT_TO_DINT
ST      incremental_encoder_evaluation_02.Increments_DINT
CAL incremental_encoder_evaluation_02(
        Machine_zero_point_DINT :=2000,
        Maximum_incremental_encoder_DINT :=16777215,
        Activate_BOOL :=1,
        Absolute_value_transmitter_BOOL :=1,
        Absolute_value_transmitter_without_referencing_BOOL :=0,
        Accept_machine_zero_point_BOOL :=DI_0_3_BOOL,
        Reference_signal_BOOL :=0,
        Actual_value_DINT=>Actual_position_02_DINT
        )

CAL axis_02(
        Setpoint_position_DINT :=Setpoint_position_02_DINT,
        Actual_position_DINT :=Actual_position_02_DINT,
        Activate_BOOL :=DI_0_1_BOOL,
        Position_deviation_switch_off_15bit_UINT :=200,
        Position_deviation_crawl_speed_15bit_UINT :=4000,
        Manipulated_variable_crawl_speed_11bit_UINT :=300,
        Manipulated_variable_rapid_traverse_11bit_UINT :=1000
        )

CAL AAINT_INT_analog_output(
        Input_value_INT :=axis_02.Manipulated_variable_12Bit_INT,
        Analog_output_INT=>AO_0_0_2_2
        )

END_PROGRAM
```

P_UIN_rapid_traverse_crawl_speed
Rapid crawl speed with unsigned integer setpoint and actual positions



Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Setpoint_position_UINT	Setpoint position (target position)	0 to 65535
Actual_position_UINT	Actual position	0 to 65535
Parameters		
Activate_BOOL	Activate (deactivate = reset) the function block	0/1
Position_deviation_switch_off_15bit_UINT	If this position deviation is breached, axis movement is stopped (manipulated variable = 0)	0 to 32767
Position_deviation_crawl_speed_15bit_UINT	If this position deviation is breached, axis movement changes from rapid traverse to crawl speed	0 to 32767

Designation	Function	Value range
Manipulated_variable_crawl_speed_11bit_UINT	Manipulated variable of crawl speed	0 to 2048
Manipulated_variable_rapid_traverse_11bit_UINT	Manipulated variable of rapid traverse	0 to 2048
Outputs		
Manipulated_variable_12bit_INT	Manipulated variable for controlling the axis actuators (setpoint rotational speed, setpoint linear speed)	–2048 to 2047
Monitor outputs		
Rapid_traverse_negative_BOOL	Status: negative rapid	0/1
Rapid_traverse_positive_BOOL	Status: positive rapid	0/1
Crawl_speed_negative_BOOL	Status: negative crawl	0/1
Crawl_speed_positive_BOOL	Status: positive crawl	0/1
Switched_off_BOOL	Status: switched off	0/1
Position_deviation_INT	Position deviation	–32768 to 32767

Description

This function block is identical to function block "P_DIN_rapid_traverse_crawl_speed", except that the setpoint and actual positions are entered as unsigned integers. For further information, see the description for this function block (from page 31).

Characteristics control

P_DINT_characteristics_control
Characteristics control with double integer setpoint and actual positions

P_DINT_characteristics_control

Type	Inputs:	Outputs:	Type
DINT	Setpoint_position_DINT	Manipulated_variable_12bit_INT	INT
DINT	Actual_position_DINT		
	Parameters:	**Monitor outputs:**	
BOOL	Activate_BOOL	Actual_position_in_tolerance_zone_BOOL	BOOL
BOOL	Manipulated_variable_negation_BOOL	Position_deviation_DINT	DINT
UINT	Ramp_time_100increase_maximum_ms_UINT		
INT	Position_deviation_off_INT		
INT	Position_deviation_minimum_INT		
INT	Position_deviation_middle_INT		
INT	Position_deviation_maximum_INT		
INT	Manipulated_variable_minimum_11Bit_INT		
INT	Manipulated_variable_middle_11Bit_INT		
INT	Manipulated_variable_maximum_11Bit_INT		
UINT	Tolerance_positioning_zone_UINT		

Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Setpoint_position_DINT	Setpoint position (target position)	-10^9 to 10^9
Actual_position_DINT	Actual position	-10^9 to 10^9

Designation	Function	Value range
Parameters		
Activate_BOOL	Activate (deactivate = reset) the function block	0/1
Manipulated_variable_negation_BOOL	Negate manipulated variable (reverse prefix)	0/1
Ramp_time_100increase_maximum_ms_UINT	Ramp time required for an increase of 100 increments at the setpoint position change	0 to 65535
Position_deviation_off_INT	Position deviation at which positioning is switched off	0 to 32767
Position_deviation_minimum_INT	Smallest position deviation	0 to 32767
Position_deviation_middle_INT	Mean position deviation	0 to 32767
Position_deviation_maximum_INT	Greatest position deviation	0 to 32767
Manipulated_variable_minimum_11Bit_INT	Smallest manipulated variable	0 to 2048
Manipulated_variable_middle_11Bit_INT	Mean manipulated variable	0 to 2048
Manipulated_variable_maximum_11Bit_INT	Greatest manipulated variable	0 to 2048
Tolerance_positioning_zone_UINT	Tolerance range (±) of the positioning zone	0 to 32767
Outputs		
Manipulated_variable_12bit_INT	Manipulated variable for controlling the axis actuators (setpoint rotational speed, setpoint linear speed)	–2048 to 2047
Monitor outputs		
Actual_position_in_tolerance_zone_BOOL	Status: Actual position is within tolerance zone	0/1
Position_deviation_DINT	Position deviation	-10^9 to 10^9

Caution!
The positioning hardware must be designed so that it cannot be damaged due to an incorrect start-up. Position end switches and a foot and palm switch must therefore be fitted for disconnecting the power supply. Incorrect starting can result in a sudden acceleration of the positioning axes at their maximum rate. During a start-up, approach the moving machine parts with caution.

Description

With this function block, axis position control can be easily implemented with little parameterization. Unlike function block "P_closed_loop_position_control" this "characteristics control" cannot be used to generate controlled speeds and ramps. Depending on the position deviation, defined manipulated variables are interpolated or output between the defined interpolation points. This allows slow deceleration of the positioning process with smaller steady-state position deviations. One function block must be instanced per axis.

From the setpoint and actual position input, a position deviation is automatically generated (→ Monitor outputs). Four position deviations (off, min., mean and max.) can be defined. The manipulated variable for position deviation "off" is automatically set to "0". To the other position deviations, any manipulated variables between 0 and 2048 can be assigned. Between the defined interpolation points, the manipulated variable is determined through linear interpolation (→ fig. 8).

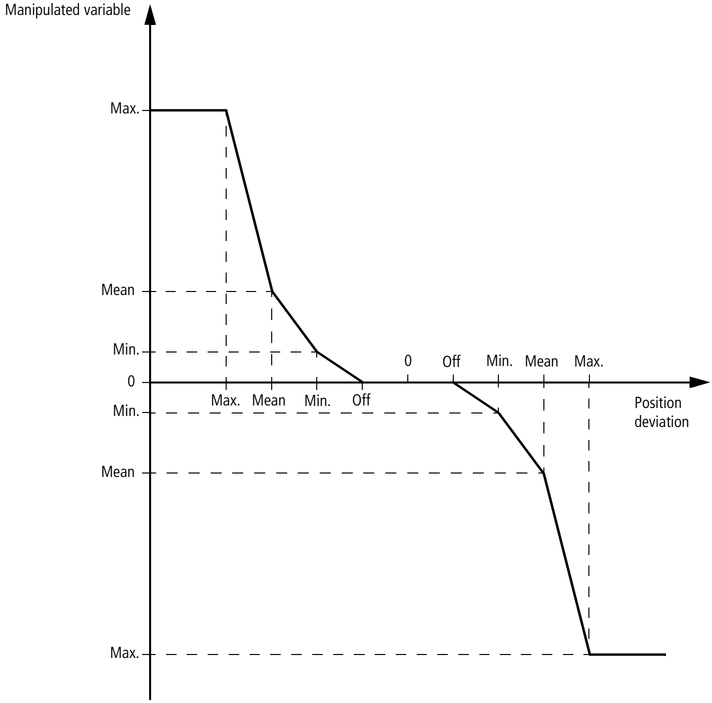


Figure 8: Manipulated variable in dependence on the position deviation

If the position change is delayed, the manipulated variable behaves as shown in figure 9. A tolerance position zone (setpoint position and tolerance value) can be defined. If the actual position is inside this positioning zone, monitor output "Actual_position_in_tolerance_zone_BOOL" carries status "1".

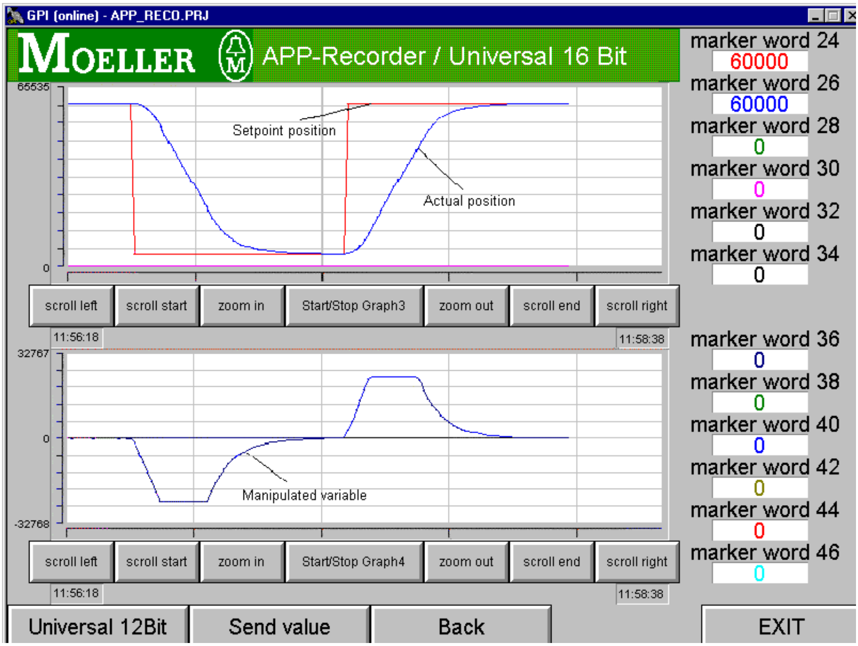


Figure 9: Acceleration and deceleration behaviour of the manipulated variable

The manipulated variable output begins as soon as the function block is activated. Deactivation results in a function block reset. If a prefix reversal of the manipulated variable is required, the value "1" must be assigned to parameter "Manipulated_variable_negation_BOOL".

On setpoint position step changes, the manipulated variable also changes suddenly. With parameter "Ramp_time_100increase_maximum_ms_UINT", this step change can be changed to a ramp. Within the entered ramp time, the manipulated variable can ramp up by up to 100 increments.

Example:
The value "90" was entered for parameter "Ramp_time_100increase_maximum_ms_UINT". This results in an increase of the manipulated variable (after the

setpoint value step change) by 100 increments every 90 ms. An increase from 0 to 1000 at this ramp time would therefore take 900 ms.

In the application example, the characteristics control function blocks were parameterized for two axes (→ fig. 10). In this program, two incremental encoder values are recorded. With function block "P_incremental_ encoder_evaluation", the incremental values are processed in such a way as to prevent a data over-range. The zero points (referencing) are specified on a rising edge of digital inputs "2" and "3". With digital input "0", the setpoint position can be changed. Digital input "1" activates the basic positioning block.

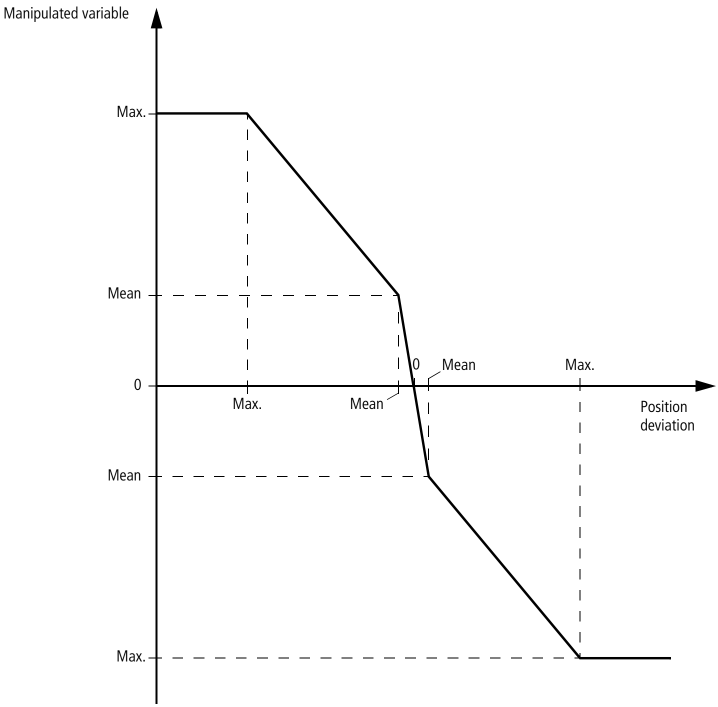


Figure 10: Optimizing the deceleration process and the steady-state position deviation using characteristics control in the application example

Application of function block "P_DINT_characteristics_control" in program "Pos_03"

```
PROGRAM Pos_03
VAR
	incremental_encoder_evaluation_01 : P_incremental_encoder_evaluation ;
	incremental_encoder_evaluation_02 : P_incremental_encoder_evaluation ;
	axis_01 : P_DINT_characteristics_control ;
	axis_02 : P_DINT_characteristics_control ;
	AAINT_INT_analog_output : P_AAINT_INT_analog_output ;
	DI_0_0_BOOL : BOOL ;
	DI_0_1_BOOL : BOOL ;
	DI_0_2_BOOL : BOOL ;
	DI_0_3_BOOL : BOOL ;
	AO_0_0_2_0 : INT ;
	AO_0_0_2_2 : INT ;
	Setpoint_position_01_DINT : DINT ;
	Setpoint_position_02_DINT : DINT ;
	Actual_position_01_DINT : DINT ;
	Actual_position_02_DINT : DINT ;
	incremental_encoder_01 : UDINT ;
	incremental_encoder_02 : UDINT ;
END_VAR
```

```
LD      DI_0_0_BOOL
JMPCN   SETPOINT_POSITION_02
        LD    5000
        ST    Setpoint_position_01_DINT
        ST    20000
        ST    Setpoint_position_02_DINT
        JMP   E_SETPOINT_POSITION_02
SETPOINT_POSITION_02:
        LD    200000
        ST    Setpoint_position_01_DINT
        LD    80000
        ST    Setpoint_position_01_DINT
E_SETPOINT_POSITION_02:

LD      incremental_encoder_01
UDINT_TO_DINT
ST      incremental_encoder_evaluation_01.Increments_DINT
CAL incremental_encoder_evaluation_01(
        Machine_zero_point_DINT :=2000,
        Maximum_incremental_encoder_DINT :=16777215,
        Activate_BOOL :=1,
        Absolute_value_transmitter_BOOL :=1,
        Absolute_value_transmitter_without_referencing_BOOL :=0,
        Accept_machine_zero_point_BOOL :=DI_0_2_BOOL,
        Reference_signal_BOOL :=0,
        Actual_value_DINT=>Actual_position_01_DINT
        )
```

```
CAL axis_01(
        Setpoint_position_DINT :=Setpoint_position_01_DINT,
        Actual_position_DINT :=Actual_position_01_DINT,
        Activate_BOOL :=DI_0_1_BOOL,
        Manipulated_variable_negation_BOOL :=0,
        Ramp_time_100increase_maximum_ms_UINT :=60,
        Position_deviation_off_INT :=0,
        Position_deviation_minimum_INT :=300,
        Position_deviation_middle_INT :=3000,
        Position_deviation_maximum_INT :=3000,
        Manipulated_variable_minimum_11Bit_INT :=200,
        Manipulated_variable_middle_11Bit_INT :=1000,
        Manipulated_variable_maximum_11Bit_INT :=1000,
        Tolerance_positioning_zone_UINT :=20
        )

CAL AAINT_INT_analog_output(
        Input_value_INT :=axis_01.Manipulated_variable_12Bit_INT,
        Analog_output_INT=>AO_0_0_2_0
        )

LD      incremental_encoder_02
UDINT_TO_DINT
ST      incremental_encoder_evaluation_02.Increments_DINT
CAL incremental_encoder_evaluation_02(
        Machine_zero_point_DINT :=2000,
        Maximum_incremental_encoder_DINT :=16777215,
        Activate_BOOL :=1,
        Absolute_value_transmitter_BOOL :=1,
        Absolute_value_transmitter_without_referencing_BOOL :=0,
        Accept_machine_zero_point_BOOL :=DI_0_3_BOOL,
        Reference_signal_BOOL :=0,
        Actual_value_DINT=>Actual_position_02_DINT
        )
```

```
CAL axis_02(
        Setpoint_position_DINT :=Setpoint_position_02_DINT,
        Actual_position_DINT :=Actual_position_02_DINT,
        Activate_BOOL :=DI_0_1_BOOL,
        Manipulated_variable_negation_BOOL :=0,
        Ramp_time_100increase_maximum_ms_UINT :=60,
        Position_deviation_off_INT :=0,
        Position_deviation_minimum_INT :=300,
        Position_deviation_middle_INT :=3000,
        Position_deviation_maximum_INT :=3000,
        Manipulated_variable_minimum_11Bit_INT :=200,
        Manipulated_variable_middle_11Bit_INT :=1000,
        Manipulated_variable_maximum_11Bit_INT :=1000,
        Tolerance_positioning_zone_UINT :=20
        )

CAL AAINT_INT_analog_output(
        Input_value_INT :=axis_02.Manipulated_variable_12Bit_INT,
        Analog_output_INT=>AO_0_0_2_2
        )

END_PROGRAM
```

P_UINT_characteristics_control Characteristics control with unsigned integer setpoint and actual positions

P_UINT_characteristics_control

	Inputs:	Outputs:	
UINT	Setpoint_position_UINT	Manipulated_variable_12bit_INT	INT
UINT	Actual_position_UINT		
	Parameters:	Monitor outputs:	
BOOL	Activate_BOOL	Actual_position_in_tolerance_zone_BOOL	BOOL
BOOL	Manipulated_variable_negation_BOOL	Position_deviation_INT	INT
UINT	Ramp_time_100increase_maximum_ms_UINT		
INT	Position_deviation_off_INT		
INT	Position_deviation_minimum_INT		
INT	Position_deviation_middle_INT		
INT	Position_deviation_maximum_INT		
INT	Manipulated_variable_minimum_11Bit_INT		
INT	Manipulated_variable_middle_11Bit_INT		
INT	Manipulated_variable_maximum_11Bit_INT		
UINT	Tolerance_positioning_zone_UINT		

Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Setpoint_position_UINT	Setpoint position (target position)	0 to 65535
Actual_position_UINT	Actual position	0 to 65535
Parameters		
Activate_BOOL	Activate (deactivate = reset) the function block	0/1
Manipulated_variable_negation_BOOL	Negate manipulated variable (reverse prefix)	0/1

Designation	Function	Value range
Ramp_time_100increase_ maximum_ms_UINT	Ramp time required for an increase of 100 increments at the setpoint position change	0 to 65535
Position_deviation_off_INT	Position deviation at which positioning is switched off	0 to 32767
Position_deviation_minimum_INT	Smallest position deviation	0 to 32767
Position_deviation_middle_INT	Mean position deviation	0 to 32767
Position_deviation_maximum_INT	Greatest position deviation	0 to 32767
Manipulated_variable_minimum_ 11Bit_INT	Smallest manipulated variable	0 to 2048
Manipulated_variable_middle_ 11Bit_INT	Mean manipulated variable	0 to 2048
Manipulated_variable_maximum_ 11Bit_INT	Greatest manipulated variable	0 to 2048
Tolerance_positioning_zone_UINT	Tolerance range (±) of the positioning zone	0 to 32767
Outputs		
Manipulated_variable_12bit_INT	Manipulated variable for controlling the axis actuators (setpoint rotational speed, setpoint linear speed)	–2048 to 2047
Monitor outputs		
Actual_position_in_tolerance_ zone_BOOL	Status: Actual position is within tolerance zone	0/1
Position_deviation_INT	Position deviation	–32768 to 32767

Description

This function block is identical to function block "P_DINT_characteristics_control", except that the setpoint and actual positions are entered as unsigned integers. For further information, see the description for this function block (from page 39).

Closed-loop position control

P_closed_loop_position_control Closed-loop position control

P_closed_loop_position_control

Type	Inputs:	Outputs:	Type
DINT	Setpoint_position_DINT	Manipulated_variable_12bit_INT	INT
DINT	Incremental_encoder_value_DINT		
DINT	Setpoint_position_order_master_axis_DINT		

Type	Parameters:	Monitor outputs:	Type
BOOL	Operating_mode_rotary_axis_BOOL	Set_position_current_job_DINT	DINT
BOOL	Operating_mode_slave_axis_BOOL	Actual_position_DINT	DINT
BOOL	Absolute_value_transmitter_BOOL	Position_deviation_DINT	DINT
BOOL	Absolute_value_transmitter_without_referencing_BOOL	Contouring_error_INT	INT
		Maximum_contouring_error_UINT	UINT
BOOL	Activate_BOOL	Setpoint_position_order_slave_axis_DINT	DINT
BOOL	Fault_signal_acknowledge_BOOL		
BOOL	Manipulated_variable_negation_BOOL	Positioning_job_finished_BOOL	BOOL
BOOL	Accept_setpoint_position_BOOL	Actual_position_in_tolerance_zone_BOOL	BOOL
BOOL	Manual_control_negative_BOOL	Parameterization_error_BOOL	BOOL
BOOL	Manual_control_positive_BOOL	Maximum_contouring_error_overrange_BOOL	BOOL
BOOL	Manual_control_suppresion_before_referencing_BOOL		
		Loop_controller_not_ready_BOOL	BOOL
BOOL	Manual_control_without_closed_loop_BOOL	Emergency_stop_triggered_BOOL	BOOL
BOOL	Referencing_negative_BOOL	Software_limit_overrange_BOOL	BOOL
BOOL	Referencing_positive_BOOL	Software_limit_attained_BOOL	BOOL
BOOL	Reference_signal_cam_BOOL	Loop_controller_enabled_BOOL	BOOL
BOOL	Reference_output_signal_incremental_encoder_BOOL	Ready_for_job_BOOL	BOOL
		Referencing_finished_BOOL	BOOL
BOOL	Emergency_stop_BOOL	Reference_input_signal_incremental_encoder_BOOL	BOOL
BOOL	Loop_controller_ready_BOOL		
UINT	Rotational_speed_negative_percent_UINT		
UINT	Rotational_speed_positive_percent_UINT		
UINT	Acceleration_time_negative_ms_UINT		
UINT	Acceleration_time_positive_ms_UINT		
UINT	Deceleration_time_negative_ms_UINT		
UINT	Deceleration_time_positive_ms_UINT		
UINT	Position_control_gain_negative_percent_UINT		
UINT	Position_control_gain_positive_percent_UINT		
UINT	Position_control_degressive_gain_percent_UINT		
UINT	Speed_precontrol_negative_percent_UINT		
UINT	Speed_precontrol_positive_percent_UINT		
UINT	Nominal_revolutions_min_UINT		
UINT	Tolerance_positioning_zone_UINT		

```
        |  Parameters (continued):
DINT  --|  Machine_zero_point_DINT
DINT  --|  Software_limit_negative_DINT
DINT  --|  Software_limit_positive_DINT
UINT  --|  Reference_point_acceleration_time_ms_UINT
UINT  --|  Reference_cam_searching_rotational_speed_percent_UINT
UINT  --|  Zero_mark_searching_rotational_speed_percent_UINT
UINT  --|  Contouring_error_maximum_UINT
UINT  --|  Manual_rotational_speed_percent_UINT
UINT  --|  Manual_acceleration_time_ms_UINT
DINT  --|  Manual_control_step_width_DINT
UINT  --|  Crawl_speed_rotational_speed_percent_UINT
DINT  --|  Crawl_speed_zone_DINT
UINT  --|  Hydraulic_zero_point_coverage_negative_UINT
UINT  --|  Hydraulic_zero_point_coverage_positive_UINT
UINT  --|  Hydraulic_stick_slip_compensation_negative_11Bit_UINT
UINT  --|  Hydraulic_stick_slip_compensation_positive_11Bit_UINT
UINT  --|  Hydraulic_stick_slip_zone_negative_UINT
UINT  --|  Hydraulic_stick_slip_zone_positive_UINT
DINT  --|  Maximum_incremental_encoder_DINT
UINT  --|  Encoder_increments_per_revolution_UINT
```

Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Setpoint_position_DINT	Setpoint position (target position)	-10^9 to 10^9
Incremental_encoder_value_DINT	Reading from the incremental or absolute encoder	-10^9 to 10^9
Setpoint_position_order_ master_axis_DINT	In “Slave axis” mode, the positioning process uses this input. The actual position of the master axis or monitor output “Setpoint_position_order_ slave_axis_DINT” of the master axis can, for example, be connected to this input.	-10^9 to 10^9

Designation	Function	Value range
Parameters		
Operating_mode_rotary_axis_BOOL	Control mode: "Normal axis" = 0, "Rotary axis" = 1. In Rotary Axis control mode, the shortest path to the setpoint position from 0 to 360° can be selected, e.g. from 350 to 10°, the difference of 20° is traversed directly across the zero point.	0/1
Operating_mode_slave_axis_BOOL	Operating mode: slave axis	0/1
Absolute_value_transmitter_BOOL	Transmitter type: "incremental encoder" = 0, "Absolute encoder" = 1. For "incremental encoder", referencing must be carried out with a reference cam signal and the zero marker signal from the incremental encoder; for "absolute encoder", only the reference cam signal is used.	0/1
Absolute_value_transmitter_ without_referencing_BOOL	Regardless of the settings for "absolute_value_transmitter_BOOL", no referencing is carried out here.	0/1
Activate_BOOL	Activates the function block. Deactivation results in a complete reset (re-initialization) of the function block.	0/1
Fault_signal_acknowledge_BOOL	Acknowledgment of the fault signal. The fault signals and the maximum contouring error are reset with each rising edge.	0/1
Manipulated_variable_negation_ BOOL	Negate manipulated variable (reverse prefix). Negation of the manipulated variable also affects the manual and referencing modes.	0/1
Accept_setpoint_position_BOOL	Control mode: Accept applied setpoint positions (automatic mode). This mode overrides manual mode.	0/1

Designation	Function	Value range
Manual_control_negative_BOOL	Control mode: Manual control (and jogging for non-zero manual control step widths), negative	0/1
Manual_control_positive_BOOL	Control mode: Manual control (and jogging for non-zero manual control step widths), positive	0/1
Manual_control_suppression_ before_referencing_BOOL	Control mode: Suppression of manual control before referencing in incremental encoder mode is completed.	0/1
Manual_control_without_closed_ loop_BOOL	Control mode: Manual control without closed loop.	0/1
Referencing_negative_BOOL	Control mode: negative referencing	0/1
Referencing_positive_BOOL	Control mode: positive referencing	0/1
Reference_signal_cam_BOOL	Signal: Reference cam	0/1
Reference_output_signal_ incremental_encoder_BOOL	Signal: reference output of the incremental encoder	0/1
Emergency_stop_BOOL	Signal: Emergency Stop	0/1
Loop_controller_ready_BOOL	Signal: controller, e.g. of a frequency inverter, ready	0/1
Rotational_speed_negative_ percent_UINT	Negative speed in % of nominal speed	0 to 65535
Rotational_speed_positive_ percent_UINT	Positive speed in % of nominal speed	0 to 65535
Acceleration_time_negative_ms_ UINT	Acceleration time, negative, in ms	0 to 65535
Acceleration_time_positive_ms_ UINT	Acceleration time, positive, in ms	0 to 65535
Deceleration_time_negative_ms_ UINT	Deceleration time, negative, in ms	0 to 65535
Deceleration_time_positive_ms_ UINT	Deceleration time, positive, in ms	0 to 65535
Position_control_gain_ negative_percent_UINT	Position control gain Kv, negative, in % (100 = 1.00)	0 to 65535

Designation	Function	Value range
Position_control_gain_positive_percent_UINT	Position control gain Kv, positive, in % (100 = 1.00)	0 to 65535
Position_control_degressive_gain_percent_UINT	The position control degressive gain ensures a linear rise of the position control gain within the tolerance positioning zone. At an input of "300 %", for example, it would rise by a factor of 3. This minimizes the steady-state position deviations and ensures smooth movement.	0 to 65535
Speed_precontrol_negative_percent_UINT	Speed precontrol factor Kg, negative, in % (100 = 1.00)	0 to 65535
Speed_precontrol_positive_percent_UINT	Speed precontrol factor Kg, positive, in % (100 = 1.00)	0 to 65535
Nominal_rotational_speed_min_UINT	Nominal rotational speed in rpm	0 to 65535
Tolerance_positioning_zone_UINT	Tolerance range (±) of the positioning zone	0 to 65535
Machine_zero_point_DINT	Machine zero point	-10^9 to 10^9
Software_limit_negative_DINT	Software limit, negative	-10^9 to 10^9
Software_limit_positive_DINT	Software limit, positive	-10^9 to 10^9
Reference_point_acceleration_time_ms_UINT	Acceleration and deceleration time for the referencing process	0 to 65535
Reference_cam_searching_rotational_speed_percent_UINT	Searching speed for the reference cam % of the nominal speed	0 to 65535
Zero_mark_searching_rotational_speed_percent_UINT	Searching speed for the zero marker signal from the incremental encoder in % of the nominal speed	0 to 65535
Contouring_error_maximum_UINT	Maximum permissible contouring error. If exceeded, positioning is cancelled.	0 to 65535
Manual_rotational_speed_percent_UINT	Speed for manual operation in % of the nominal speed	0 to 65535
Manual_acceleration_time_ms_UINT	Acceleration and deceleration time for manual control mode in ms	0 to 65535

Designation	Function	Value range
Manual_control_step_width_DINT	Step width for manual mode (values ≠ 0 => jogging)	-10^9 to 10^9
Crawl_speed_percent_UINT	Crawl speed at the end of the positioning process	0 to 65535
Crawl_speed_window_DINT	Crawl speed range about setpoint position	0 to 65535
Hydraulic_zero_point_coverage_negative_UINT	Negative zero point coverage for hydraulic axes	0 to 65535
Hydraulic_zero_point_coverage_positive_UINT	Positive zero point coverage for hydraulic axes	0 to 65535
Hydraulic_stick_slip_compensation_negative_11Bit_UINT	Negative stick-slip compensation for hydraulic axes	0 to 65535
Hydraulic_stick_slip_compensation_positive_11Bit_UINT	Positive stick-slip compensation for hydraulic axes	0 to 65535
Hydraulic_stick_slip_zone_negative_UINT	Negative stick-slip compensation is active only within this zone	0 to 65535
Hydraulic_stick_slip_zone_positive_UINT	Positive stick-slip compensation is active only within this zone	0 to 65535
Maximum_incremental_encoder_DINT	Maximum value from the incremental encoder (before overflow)	0 to 10^9
Encoder_increments_per_revolution_UINT	Encoder increments per revolution	0 to 65535
Outputs		
Manipulated_variable_12bit_INT	Manipulated variable for controlling the axis actuators (setpoint rotational speed, setpoint linear speed)	–2048 to 2047
Monitor outputs		
Set_position_current_job_DINT	Setpoint position of current job	-10^9 to 10^9
Actual_position_DINT	Actual position (= encoder value, machine zero point and overflow correction of incremental encoder => negative values possible)	-10^9 to 10^9
Position_deviation__DINT	Position deviation	-10^9 to 10^9

Designation	Function	Value range
Contouring_error_INT	Contouring error (internal setpoint position order – actual position)	–32768 to 32767
Maximum_contouring_error_UINT	Maximum determined contouring error (without prefix)	0 to 32767
Setpoint_position_order_ slave_axis_DINT	This monitor output provides the setpoint position of the current positioning process. This value can be applied to a slave axis. The slave and master axes are then regulated to the same setpoint position. Alternative: The actual position of this function block is applied to the slave axis.	-10^9 to 10^9
Positioning_job_finished_BOOL	Status: Positioning job finished	0/1
Actual_position_in_tolerance_ zone_BOOL	Status: Actual position is within tolerance zone	0/1
Parameterization_error_BOOL	Error: Parameterization error	0/1
Maximum_contouring_error_ overrange_BOOL	Error: Maximum contouring error overrange	0/1
Loop_controller_not_ready_BOOL	Error: Loop controller, e.g. of a frequency inverter, not ready (→ Parameter)	0/1
Emergency_stop_triggered_BOOL	Error: Emergency Stop was triggered	0/1
Software_limit_overrange_BOOL	Error: The actual position has exceeded the software limits by more than 100 increments.	0/1
Software_limit_attained_BOOL	Status: The setpoint position is outside the software limits.	0/1
Loop_controller_enabled_BOOL	Status: Controller ready (no faults)	0/1
Ready_for_job_BOOL	Status: Ready for job (no fault, previous job and referencing completed)	0/1
Referencing_finished_BOOL	Status: Referencing finished	0/1
Reference_input_signal_ incremental_encoder_BOOL	Reference input signal for incremental encoder	0/1

Description

With this function block, position control of an axis can be implemented. One function block must be instanced per axis. The terms “negative” and “positive” refer to the prefix of the manipulated variable.

→ For detailed explanations of basic positioning terms, → chapter “The Principle of Drive Positioning” from page 15.

Start-up

Caution!
The positioning hardware must be designed so that it cannot be damaged due to an incorrect start-up. Position end switches and a foot and palm switch must therefore be fitted for disconnecting the power supply. Incorrect starting can result in a sudden acceleration of the positioning axes at their maximum rate. During a start-up, approach the moving machine parts with caution.

Parameters “Emergency_stop_BOOL” and “Loop_controller_ready_BOOL” must be linked with the corresponding hardware signals. If no hardware signals (position end switches) are present, “Emergency_stop_BOOL” must be set to “0” or “Loop_controller_ready_BOOL” to “1”. Monitor output “Loop_controller_not_ready_BOOL” must be linked to the actuators in such a way as to deactivate the actuators if “Loop_controller_not_ready_BOOL” is set to “1”.

Setpoint position

Input “Setpoint_position_DINT” should be linked, for example, to a sequencer or for visualization. A new positioning job is initiated by changing the setpoint position. The setpoint position of the current job is indicated at the function block’s output.

Actual position
Connect input "Incremental_encoder_value_DINT" to the corresponding signal from the incremental or absolute encoder. The function block internally monitors the incremental encoder value for overflow. Negative setpoint and actual positions from –1 000 000 000 to 1 000 000 000 can therefore also be approached. At the output of the function block, the actual position (possibly a referenced value) is then available with output actual_position_DINT.

Maximum transducer increments
Assign the maximum value from the incremental encoder to parameter "Maximum_incremental_encoder_DINT", for example 65 535 for 16 bit resolution or 16 777 215 for 24 bit resolution.

Transducer increments per revolution
Assign the corresponding value to parameter "Encoder_increments_per_revolution_UINT".

→ A multiplication of the software resolution may need to be considered if the number of "transducer increments per revolution" in the program is a multiple of those of the hardware. If the software resolution is set incorrectly, a 100 % nominal speed setting would result in an actual speed of, for example, only 50 or 25 %.

Example:
Hardware coding = 1024 increments per revolution
Multiplication of resolution by the software
=>
"Encoder_increments_per_revolution_UINT" must be set to 4096.

Software limits

Enter a negative (or minimum) software limit for the lowest setpoint position and a positive (or maximum) software limit for the highest setpoint position. If the setpoint position of a current job lies outside the software limits, it is then kept within the software limit. Monitor output "Software_limit_attained_BOOL" then indicates status "1". If the actual position lies outside the software limits by more than 100 increments, monitor output "Software_limit_overrange_BOOL" indicates status "1".

Caution!
In incremental encoder or absolute encoder with referencing mode, the software limits can be taken into account only after referencing. After referencing (in "absolute encoders without referencing" mode immediately), manual control and acceptance of the setpoint position are limited to fixed software limits.

Nominal speed

Enter the nominal speed of the axis in revolutions per minute.

→ This speed must be exceeded by a significant amount when the maximum setpoint variable is output, otherwise the actual position of the "internally specified setpoint position" cannot be approached, resulting in a growing contouring error.

Speeds, acceleration and deceleration times

Enter linear speeds, acceleration and deceleration times for the positive and negative direction. Speeds are entered in percent of the nominal speed.

Example:
Nominal speed = 1000 rpm
Speed, negative = 50 % = 500 rpm
Speed, positive = 20 % = 200 rpm

During start-up, check whether the specified speeds are reached by the actuators. If the entered speeds are small enough, a visual inspection can be made. Any deviations may be due to an incorrect value in parameter "Encoder_increments_per_revolution_UINT". The acceleration and deceleration times relate to the greatest possible acceleration and deceleration ramps that can occur between zero speed and nominal speed (→ fig. 11).

→ The acceleration and deceleration times can not be changed during the positioning process; the speeds, however, can be changed.

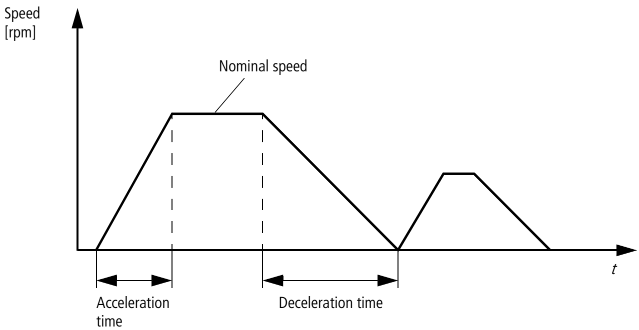


Figure 11: Speed sequence of a positioning process

Crawl speed

If parameters "Crawl_speed_percent_UINT" and "Crawl_speed_zone_DINT" are not equal to "0", the end of the positioning process (according to the selected parameters) is traversed at crawl speed.

Position control gain "Kv", speed precontrol factor "Kg" and position control degressive gain

To generate a manipulated variable, position control gain Kv is multiplied by the contouring error and speed precontrol factor Kg by the (internally defined) setpoint speed variation.

Your first step should be to optimize position control gain Kv. Using, for example, an oscilloscope to measure the analog output voltage, check whether the specified "acceleration and deceleration ramps" are linear.

If the ramps are not linear, Kv must be increased (for example by starting with Kv = 10 and increasing this value by a factor of 2 each time).

If positioning is uneven, Kv must be reduced.

If values above 100 are entered, the position control degressive gain causes a linear increase of position control gain Kv within the tolerance position zone (→ fig. 12). This allows a reduction of the steady-state position deviation at standstill. A gentler position control gain can therefore be used for the process than for standstill.

Example:
Negative position control gain KV = 100 %
Positive position control gain KV = 100 %
Position control degressive gain = 300 %
=>
When the actual position reaches the tolerance zone, the position control gain increases at a constant ramp by factor 3 to 300 % until the the actual position has reached the setpoint (target) position.

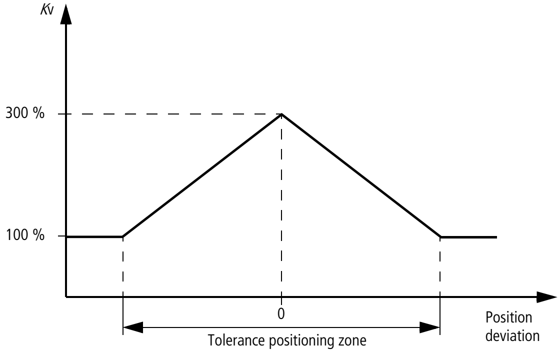


Figure 12: Position control gain Kv increases as the position deviations decrease.

For a positioning process, it is not vital to parameterize speed precontrol factor Kg. This factor can be used to reduce the contouring error. Check the magnitude of the contouring error at a movement in the positive direction (→ Monitor outputs), observing the prefix. By selecting a suitable value for Kg, this contouring error can be significantly reduced. Start with "Kg = 10" and observe the effect on the contouring error. The setting is acceptable when the (maximum) contouring error is reduced by a factor of about 5. If the contouring error prefix changes during a movement in the positive direction, Kg is too large.

Caution!
An excessively large value of Kg can result in excessive setpoint position overshoot.

→ The three factors can be changed at any time during the positioning process. Before the initial start-up, it is advisable to begin with a small Kv, a Kg of zero and a neutral position control degressive gain.

Example:
Position_control_gain_negative_percent_UINT= 10
Position_control_gain_positive_percent_UINT = 10
Speed_precontrol_negative_percent_UINT= 0
Speed_precontrol_positive_percent_UINT= 0
Position_control_degressive_gain_percent_UINT = 100

Maximum contouring error

In practice, axes (motor or power and mechanical equipment) have finite dynamics due to their limited acceleration capability. When the axis moves, this property results in a contouring error. This means that the actual position (specified internally in the function block) lags behind the by the amount of the contouring error.

The contouring error is directly proportional to the position control gain Kv and the setpoint speed. The contouring error is permanently monitored by the position control. An excessive contouring error during a traversing movement indicates an electrical or mechanical fault. The axis may, for example, have encountered an obstruction in its traversed path, or there may be a fault in the drive system. To limit any damage, the affected drive must then be switched off immediately.

An axis whose contouring error has exceeded the greatest permissible contouring error is disabled and the manipulated variable (which must be linked with the analog output for the actuators) is set to zero. A rising value of "Fault_signal_acknowledge_BOOL" can be used to restart positioning procedures.

The maximum contouring error for an axis can be worked out as follows. First, define a very large maximum contouring error (e.g. 20000 or 65535). Having optimized the position control gain and speed precontrol factors, traverse the axis at its maximum speed. Monitor output "Maximum_contouring_error_UINT" then indicates the maximum contouring error, which provides a direct measure for the maximum contouring error to be entered for stationary operation. This value should be doubled to prevent the contouring error monitor from being triggered as soon as the axis load changes.

→ Monitor output "Maximum_contouring_error_UINT" can be reset with a fault signal acknowledgment.

Tolerance positioning zone
When a setpoint position is reached, the actual position may oscillate slightly about the setpoint position, or a small steady-state position deviation may occur. A tolerance positioning zone greater than zero must therefore be defined to determine when the target position has been reached.
If the actual value is within this zone, monitor output "Actual_position_in_tolerance_zone_BOOL" carries a "1". This status signal can, for example, be linked to the stepping condition of a sequencer.

Example:
Tolerance_positioning_zone_UINT = 10
Set_position_current_job_DINT = 20000
=>
The tolerance positioning zone lies between 19990 and 20010.

Parameters for hydraulic axes

If you do not want to position a hydraulic axis, you can set all hydraulic parameters to zero. Unlike electrical servo motors, servo-hydraulic axes can exhibit a significant stick-slip effect (transitions between sticking and slipping friction). To compensate for the stick-slip effect, a stick-slip value Ks can each be defined in the positive and negative direction. Within a stick-slip zone, this Ks value is then added to the "normal" manipulated variable.

Example:
Hydraulic_stick_slip_compensation_negative_
11Bit_UINT = 500
Hydraulic_stick_slip_compensation_positive_
11Bit_UINT = 500
Hydraulic_stick_slip_zone_negative_UINT = 20
Hydraulic_stick_slip_zone_positive_UINT = 20
=>
At the start of a positioning process, the manipulated variable is increased by 500 until the actual position is removed from its starting position by more than 20 increments.

When servo-hydraulic axes are used, a zero-point overlap exists in general, i.e. small manipulated variables about the zero point have no effect. This can be corrected with parameters "Hydraulic_zero_point_coverage_negative_UINT" and "Hydraulic_zero_point_coverage_positive_UINT".

→ During commissioning, you can use the following procedure: Manually (through direct assignment of the analog outputs) determine the positive and negative manipulated variables at which the positioning axis responds and enter these values.

Control mode:
Rotary axis mode can be operated only in combination with "Absolute_value_transmitter_without_referencing_BOOL" = "1". In this mode, axis positioning takes place within no more than one revolution (0 to 360°). Manual mode can be used as usual.The following options are available for automatic mode:

1. The setpoint positions are always approached in a negative direction:

- Accept_setpoint_position_BOOL = 1
- Manual_control_negative_BOOL = 1
- Manual_control_positive_BOOL = 0

Example:
In a positioning process from 20 to 30°, the axis crosses its zero point (360°), traversing a total of 350° (and not 10°). In a positioning process from 30 to 20°, the axis does not cross its zero point. An angle of 10° is traversed.

2. The setpoint positions are always approached in a positive direction:

- Accept_setpoint_position_BOOL = 1
- Manual_control_negative_BOOL = 0
- Manual_control_positive_BOOL = 1

3. During positioning, the zero point is never exceeded:

- Accept_setpoint_position_BOOL = 1
- Manual_control_negative_BOOL = 0
- Manual_control_positive_BOOL = 0

4. During positioning, the shortest path is always chosen, if necessary crossing the zero point:

- Accept_setpoint_position_BOOL = 1
- Manual_control_negative_BOOL = 1
- Manual_control_positive_BOOL = 1

Example:
In a positioning process from 320 to 15°, the axis crosses its zero point (360°), traversing a total of 55° (and not 305°). In a positioning process from 15 to 320°, the axis also crosses its zero point. Again, an angle of 55° is traversed.

Control mode: incremental encoder/absolute encoder with/without referencing

Incremental encoder:
When parameters "Absolute_value_transmitter_BOOL" and "Absolute_value_transmitter_without_referencing_BOOL" are set to "0", the control mode is "incremental encoder". Positioning processes can then be carried out only after referencing. Referencing can be performed, for example, by resetting counter module.

Absolute encoder with referencing:
If you assign the value "1" to "Absolute_value_transmitter_BOOL", and "0" to parameter "Absolute_value_transmitter_without_referencing_BOOL", the active control mode will be "Absolute encoder with referencing". Positioning processes can then be carried out only after referencing. Referencing takes place exclusively with a reference cam signal.

Absolute encoder without referencing:
If you assign the value "0" or "1" to "Absolute_value_transmitter_BOOL", and "1" to parameter "Absolute_value_transmitter_without_referencing_BOOL", the active control mode will be "Absolute encoder without referencing". Positioning processes can then also be carried out without referencing.

Caution!
In this mode data overflow monitoring of the "Incremental_encoder_value_DINT" input does not take place.

Negating the manipulated variable

It may be necessary to reverse the prefix of the manipulated variable, for example if the encoder's direction of rotation does not match the sign of the manipulated variable. This is done by assigning a "1" to parameter "Manipulated_variable_negation_BOOL".

Hardware links

Parameters "Emergency_stop_BOOL" and "Loop_controller_ready_BOOL" must be linked with the hardware as described above. The "Manual control", "Referencing" and "Automatic control" functions (accept setpoint position) of the function block can only be used when "Emergency_stop_BOOL" has the value "0" and "Loop_controller_ready_BOOL" the value "1". If these hardware signals are not present, fixed values must be entered.

Activating the function block

Assign the "0" to the BOOL parameters of "Accept setpoint position", "Manula mode" and "Referencing" and activate the function block:

- Activate_BOOL = 1
- Accept_setpoint_position_BOOL = 0
- Manual_control_negative_BOOL = 0
- Manual_control_positive_BOOL = 0
- Referencing_negative_BOOL = 0
- Referencing_positive_BOOL = 0

After activation, the position control must become active, i.e. no axis drift can take place. The axes can then no longer be moved manually. Monitor output "Loop_controller_not_ready_BOOL" must output a "0". An output other than zero indicates a fault (→ Monitor outputs).

→ Only when "Activate_BOOL" or "Fault_signal_acknowledge_BOOL" has a rising edge are the fault messages reset to zero, so that monitor output "Loop_controller_not_ready_BOOL" can output a "0" and the function block responds again.

Control mode: Manual mode/Jog

Values must be assigned to parameters "Manual_rotational_speed_percent_UINT" and "Manual_acceleration_time_ms_UINT". The function of the manual control mode is linked with parameter "Manual_control_step_width_DINT":

- "Manual_control_step_width_DINT = 0"
  =>
  When manual control is activated, the axis is traversed in the direction of the positive or negative software limits (only after referencing; → below).
- "Manual_control_step_width_DINT ≠ 0"
  =>
  Each time manual control is activated, the corresponding step width is traversed (jogging).

→ For safety, enter a small initial value for parameter "Manual_control_step_width_DINT" during the first start-up, for example corresponding to one axis rotation.

To allow manual control, the BOOL parameters of "Referencing" and "Accept setpoint position" must be set to "0". Manual control is also locked if it is triggered in the positive and negative directions at the same time.

- Activate_BOOL = 1
- Accept_setpoint_position_BOOL = 0
- Manual_control_negative_BOOL = 1
- Manual_control_positive_BOOL = 0
- Referencing_negative_BOOL = 0
- Referencing_positive_BOOL = 0

or

- Activate_BOOL = 1
- Accept_setpoint_position_BOOL = 0
- Manual_control_negative_BOOL = 0
- Manual_control_positive_BOOL = 1
- Referencing_negative_BOOL = 0
- Referencing_positive_BOOL = 0

Once referencing is completed (and the referencing commands have been deactivated, → below), the axis can be traversed in manual or jog mode. The fixed software limits must be maintained, however. In "Absolute encoder without referencing" mode, movement is possible immediately. The dynamics defined with parameters "Manual_rotational_speed_percent_UINT" and "Manual_acceleration_time_ms_UINT" take effect.

→ Jogging means that the specified step width is traversed for each rising edge of "Manual_control_negative_BOOL" or "Manual_control_positive_BOOL".

→ When referencing is completed, the software limits are observed, otherwise they are ignored.

Manual control suppression

Meaningful software limits for position control can be observed only after referencing. Thereafter, these software limits can not be exceeded in manual control mode. Before referencing, no software limits are applied for manual control. Manual operation can be suppressed before referencing is complete by setting parameter "Manual_control_suppression_before_referencing_BOOL" to "1". This applies also to the "Absolute encoder with referencing" mode. Otherwise it is advisable to enter a lower value for "Manual_control_step_width_DINT" (→ above)

Referencing

During referencing, the hardware and software zero points are synchronized. For absolute encoders, it may not be necessary to define a zero point. If, however, the signals are provided by incremental encoders, the axes must be referenced. Otherwise the axis position on start-up is not known.

Referencing in incremental encoder control mode:
In this case, parameters "Absolute_value_transmitter_ BOOL" and "Absolute_value_transmitter_without_ referencing_BOOL" must be set to "0". Parameters "Referencing_negative_BOOL" and "Referencing_positive_BOOL" must be linked to switches. Parameters "Reference_signal_cam_BOOL" and "Reference_output_signal_incremental_encoder_BOOL" must be linked to the corresponding signals. Monitor output "Reference_input_signal_ incremental_encoder_BOOL" must be assigned to the corresponding incremental encoder input.

Values must be assigned to parameters "Reference_point_ acceleration_time_ms_UINT" and "Reference_cam_ searching_rotational_speed_percent_UINT", e.g. "Reference_point_acceleration_time_ms_UINT = 500" and "Reference_point_searching_rotational_speed_ percent_UINT = 10". These parameters specify the dynamics used to search for the reference cam. Parameter "Zero_mark_searching_rotational_speed_percent_UINT" specifies the rotary or linear speed with which the incremental encoder's zero-position is searched for. This value should be smaller than the reference cam search speed.

With a value ≠ 0 at parameter "Machine_zero_ point_DINT", a zero point deviation of the "referencing zero point" can be created. The following applies:

- Zero point = referencing zero point + Machine_zero_point_DINT (negative input possible)

Caution!
Software limits are not observed during referencing. The machine zero point must not be changed after referencing.

The referencing process takes place as follows (→ fig. 13). First, the reference cam is searched for in the specified direction at the selected reference cam search speed. Once the reference cam has been found, the search operation is terminated. The axis is then traversed in the opposite direction until the reference cam is left again (in this direction). Then, at the next zero marker signal from the incremental encoder, referencing is carried out.

The following illustration shows reference cycle sequence:

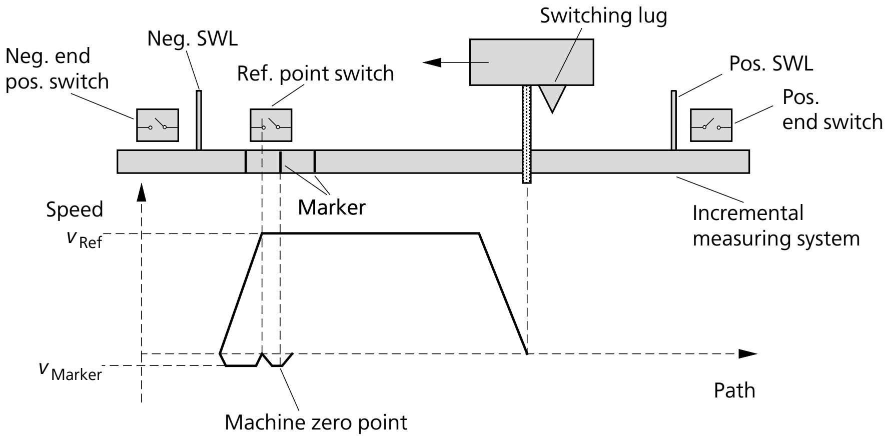


Figure 13: Reference point approach cycle for incremental measuring systems

Starting the referencing process:
Determine the direction (negative or positive) in which referencing needs to take place for the referencing cam to be passed or found. Start the referencing operation in this direction:

- Referencing_negative_BOOL = 1
- Referencing_positive_BOOL = 0

or

- Referencing_negative_BOOL = 0
- Referencing_positive_BOOL = 1

After successful completion of the referencing operation, monitor output “Referencing_finished_BOOL” carries status “1”. The “Ready for job” status for positioning becomes active when parameter “Referencing_negative_BOOL” or “Referencing_positive_BOOL” is reset to “0”. If referencing was not completed successfully, check whether all signals were linked correctly as described above.

→ If the referencing operation was started on the reference cam, e.g. in the negative direction, the search for the incremental encoder’s zero mark signal is carried out in the opposite direction (in this case in the positive direction).

Referencing in absolute encoder with referencing mode:

In this case, parameter “Absolute_value_transmitter_ BOOL” must be set to “1” and a “0” assigned to “Absolute_value_transmitter_without_referencing_BOOL”.

First possibility:

The following parameters are selected not equal to “0”

- Reference_point_acceleration_time_ms_UINT ≠ 0
- Reference_point_searching_rotational_speed_ percent_UINT ≠ 0

With one exception, the referencing process is the same as the referencing process in incremental encoder mode (→ page 72). Once the reference cam has been found, the search operation is terminated. The axis is then traversed in the opposite direction until the reference cam is left again (in this direction). When the reference cam is left, referencing is triggered, unlike in incremental encoder mode. “Reference_output_signal_incremental_encoder_BOOL” and “Reference_input_signal_incremental_ encoder_BOOL” cannot be linked in this referencing mode.

→ The repetition accuracy of the referencing operation depends on the reference point search manipulated variable. The smaller this manipulated variable, the greater the repetition accuracy of the referencing operation. If "Zero_mark_searching_rotational_speed_percent_UINT=0", the result is a speed of 0.25 %.

Second possibility:

The following parameters are selected not equal to "0"

- Reference_output_signal_incremental_encoder_BOOL = 0
- Reference_point_acceleration_time_ms_UINT = 0
- Reference_point_searching_rotational_speed_percent_UINT = 0

Referencing is triggered if the following three parameters all have a falling edge at the same time:

- Referencing_negative_BOOL = falling edge
- Referencing_positive_BOOL = rising edge
- Reference_signal_cam_BOOL = falling edge

To be able to work with negative position values, the zero point can be moved with the "Machine_zero_point_DINT" variables (this can also be done in absolute encoder without referencing mode").

Example:
The absolute encoder has a range of "0" to "1 000 000".
If you assign the value "–500 000" to parameter "Machine_zero_point_DINT", this range is offset to "–500 000 to 500 000".

On referencing, a "0" is assigned to the current value of "Incremental_encoder_value_DINT" and added to the machine zero point. This referencing process can also be used for incremental coordinate positioning processes.

Example:
Parameters “Reference_signal_cam_BOOL”, “Referencing_negative_BOOL” and “Referencing_positive_BOOL” were consecutively set from “1” to “0” (corresponding to a falling edge), while “Incremental_encoder_value_DINT” had the value “455620”. For the machine zero point, “5000” was entered. The encoder’s maximum value is 1000000. The following encoder values then correspond to the following actual positions:

Incremental_encoder_value_ DINT	**Actual_position_ DINT**
50620	–400000
430620	–20000
450620	0
455620 (current value at time of referencing)	5000
950620	500000

Control mode: automatic/accept setpoint positions
In Automatic control mode, the pending setpoint positions are approached automatically. The following parameters are “frozen” when the positioning job is started and can not be changed during positioning:

- Acceleration_time_negative_ms_UINT
- Acceleration_time_positive_ms_UINT
- Deceleration_time_negative_ms_UINT
- Deceleration_time_positive_ms_UINT
- Nominal_rotational_speed_min_UINT

If “Accept_setpoint_position_BOOL” has a falling edge (1 → 0), the current positioning job is cancelled.

→ This operating state has priority over manual control mode and no priority over referencing. The BOOL variables of the referencing function must therefore be set to "0". It is also advisable to set the BOOL variables for manual operation to zero to prevent inadvertent axis movement. The BOOL variables of the referencing function must therefore be set to "0".

- Activate_BOOL = 1
- Accept_setpoint_position_BOOL = 1
- Manual_control_negative_BOOL = 0
- Manual_control_positive_BOOL = 0
- Referencing_negative_BOOL = 0
- Referencing_positive_BOOL = 0

Automatic mode with sequencers:
If a sequencer is used, all that needs to be done is to link the sequencer's changeable setpoint positions, speeds, acceleration times, etc. with the function block. For the switching or stepping condition, you can use the following monitor outputs:

- Positioning_job_finished_BOOL
- Actual_position_in_tolerance_zone_BOOL

Master-slave positioning
The function block "P_closed_loop_position_control" can also be used as slave axis. If parameter "Operating_mode_slave_axis_BOOL" is set to "1", the axis follows the target setpoint positions of a master axis. To "Setpoint_position_order_master_axis_DINT", both "Actual_position_DINT" of a master axis and "Setpoint_position_order_slave_axis_DINT" can be assigned. In the latter case, both axes follow the same setpoint positions, so that the axes are synchronized without contouring error. This is useful when it is important that both the master and the slave axis arrive at their target positions at the same time.

Choosing the actual position of the master axis as target setpoint position has the advantage that the same contour can be traversed more accurately when the axis is to traverse back the way it came. When “Setpoint_position_order_ slave_axis_DINT” is used, the offset would be about twice as large as the current contouring error of the master axis.

Before the function block can be changed over from master axis mode to slave axis mode and vice versa, referencing must have been completed and the BOOL parameters for manual operation and referencing must have a zero value. Otherwise a parameterization error is indicated.

Also:

- Referencing_finished_BOOL = 1
- Referencing_negative_BOOL = 0
- Referencing_positive_BOOL = 0
- Manual_control_negative_BOOL = 0
- Manual_control_positive_BOOL = 0

At the time of changeover “Setpoint_position_order_ master_axis_DINT” must be in the range ±50 about the current actual position of the slave axis. Otherwise a parameterization error is indicated.

Virtual master axis:

In some applications, it is desirable to make several slave axes dependent on one virtual master axis. The master axis should, then be operated in the “absolute encoder without referencing”. Because a contouring error is not acceptable, the manipulated variable is generated exclusively with the speed precontrol factors. The following parameters must then be set accordingly:

- Absolute_value_transmitter_BOOL = 1
- Absolute_value_transmitter_without_ referencing_BOOL = 1
- Position_control_gain_negative_percent_UINT = 0
- Position_control_gain_positive_percent_UINT = 0
- Contouring_error_maximum_UINT = 65535

Examples:
Speed_precontrol_negative_percent_UINT = 50
Speed_precontrol_positive_percent_UINT = 50

Faults and controller readiness

The following faults can arise:

- Parameterization error:

A parameterization error is present in the following cases:

  - Acceleration or deceleration times of “0” were entered.
  - The specified negative software limit is greater than the positive software limit.
  - Control mode “rotary axis” was not selected in combination with “absolute encoder without referencing”.

- Maximum contouring error exceeded:

The greatest permissible (specified) contouring error was exceeded.

- Actuator controller not ready:

Parameter “Loop_controller_ready_BOOL” had status “0” after activation of the function block.

- Emergency Stop triggered:

Parameter “Emergency_stop_BOOL” had status “0” after activation of the function block.

- Software limit exceeded:

The actual position has exceeded the specified software limits by more than 100 increments. In incremental encoder control mode, this fault message is output only after referencing.

All error fault messages and the maximum contouring error are reset with a rising edge in “Fault_signal_ acknowledge_BOOL”. If a fault condition is present, “Loop_controller_enabled_BOOL” carries status “0”.

- Software limit reached (no fault):

The setpoint position is outside the specified software limits. This status does not result in termination of the positioning process. The function block's internal (current) setpoint positions are limited by the software limits.

Troubleshooting

Fault	Cause	Remedy
After activation of the function block, the axes exhibit uncontrolled acceleration.	The polarity of the encoders' setpoint positions and directions of rotation do not match.	Change the polarity of the analog output with "Manipulated_variable_negation_BOOL = 1".
After referencing, neither manual nor automatic mode work.	The BOOL variables of the referencing function are still set to "1" or the actual position lies outside the software limits.	Set the BOOL variables of the referencing function to "0" or set different software limits.
The axes do not reach the set percentage speeds.	Incorrect nominal speed or encoder resolution values were entered. Multiplication of the encoder resolution by the software may not have been taken into account.	Change nominal speed. It may be necessary (for incremental encoders) to enter an encoder resolution that is four times greater than the actual hardware resolution.
Once the function block is active, no position control is performed.	Fault/parameterization fault; e.g. no position control gain factors entered.	Rectify and acknowledge the fault or correct the parameter settings.
The referencing operation with reference cam and zero mark searching cannot be started.	The function block is not active or control mode "absolute encoder without referencing" is active.	Activation of function block or selection of control mode "Incremental encoder" or "Absolute encoder with referencing".
The steady-state control deviation after a positioning process is very large.	The selected position control gain values are too low.	Set the position control gain values and the position control degressive gain as high as possible. Oscillations and "chattering" must, however, be avoided.

Fault	Cause	Remedy
Axis travel is uneven (chattering)	The selected position control gain values are too high.	Lower the position control gain values (keeping them as high a possible, however)
Axis travel is even, but when the axis reaches its setpoint position, it starts to oscillate.	The selected position control degressive gain is too high.	Select a lower position control degressive gain.
The actual position oscillates beyond the setpoint position.	Axis deceleration is too rapid or the selected speed precontrol factors are too high.	Increase the deceleration times or reduce the speed precontrol factors.
After acceleration, the contouring error continues to increase.	The actual position cannot follow the internally defined setpoint position.	Select a lower nominal speed.

Example:
The application example implements positioning of two axes. The actual positions are determined with an incremental encoder. The referencing operation can be started in the negative direction with digital input "4". Digital input "1", activates the closed-loop positioning controllers. With digital inputs "2" and "3", manual control in the negative and positive direction is activated. Because "Manual_control_step_width_DINT" contains a non-zero value, the defined step width of 600 increments is traversed (jogging) with each manual control operation. Automatic mode (accept setpoint position) can be activated with digital input "5". With digital input "0", the setpoint position can be changed.

The parameterization illustrated in this application example results in a maximum speed of 1 440 rpm (1800 × 80 %) in the negative direction, and of 1080 rpm (1800 × 60 %) in the positive direction. Within the tolerance zone, the position control gain increases from 60 to 180 %. The incremental encoder provides up to 24 bits. Parameter "Maximum_incremental_encoder_DINT" was therefore given the value 16 777 215 (= 2^{24} –1). The incremental encoder hardware supplies 256 increments per revolution. Due to software multiplication, a 1024 was entered in parameter "Encoder_increments_per_revolution_UINT".

Application of function block "P_closed_loop_position_control" in program "Pos_04"

```
PROGRAM Pos_04
VAR
        Axis_01 : P_closed_loop_position_control ;
        Axis_02 : P_closed_loop_position_control ;
        AAINT_INT_analog_output : P_AAINT_INT_analog_output ;
        DI_0_0_BOOL : BOOL ;
        DI_0_1_BOOL : BOOL ;
        DI_0_2_BOOL : BOOL ;
        DI_0_3_BOOL : BOOL ;
        DI_0_4_BOOL : BOOL ;
        DI_0_5_BOOL : BOOL ;
        Reference_signal_cam_BOOL : BOOL ;
        AO_0_0_2_0 : INT ;
        AO_0_0_2_2 : INT ;
        Setpoint_position_01_DINT : DINT ;
        Setpoint_position_02_DINT : DINT ;
        Incremental_encoder_01 : UDINT ;
        Incremental_encoder_02 : UDINT ;
END_VAR
```

```
LD      DI_0_0_BOOL
JMPCN   SETPOINT_POSITION_02
        LD    5000
        ST    Setpoint_position_01_DINT
        LD    20000
        ST    Setpoint_position_02_DINT
        JMP   E_SETPOINT_POSITION_02
SETPOINT_POSITION_02:
        LD    300000
        ST    Setpoint_position_01_DINT
        LD    80000
        ST    Setpoint_position_02_DINT
E_SETPOINT_POSITION_02:

LD      incremental_encoder_01
UDINT_TO_DINT
ST      axis_01.Incremental_encoder_value_DINT
CAL axis_01(
        Setpoint_position_DINT :=Setpoint_position_01_DINT,
        Operating_mode_rotary_axis_BOOL :=0,
        Absolute_value_transmitter_BOOL :=0,
        Absolute_value_transmitter_without_referencing_BOOL :=0,
        Activate_BOOL :=DI_0_1_BOOL,
        Fault_signal_acknowledge_BOOL :=0,
        Manipulated_variable_negation_BOOL :=0,
        Accept_setpoint_position_BOOL :=DI_0_5_BOOL,
        Manual_control_negative_BOOL :=DI_0_2_BOOL,
        Manual_control_positive_BOOL :=DI_0_3_BOOL,
        Manual_control_suppression_before_referencing_BOOL :=1,
        Referencing_negative_BOOL :=DI_0_4_BOOL,
        Referencing_positive_BOOL :=0,
        Reference_signal_cam_BOOL :=Reference_signal_cam_BOOL,
        Emergency_stop_BOOL :=0,
        Loop_controller_ready_BOOL :=1,
```

```
Rotational_speed_negative_percent_UINT :=100,
Rotational_speed_positive_percent_UINT :=100,
Acceleration_time_negative_ms_UINT :=5000,
Acceleration_time_positive_ms_UINT :=3000,
Deceleration_time_negative_ms_UINT :=1600,
Deceleration_time_positive_ms_UINT :=1600,
Position_control_gain_negative_percent_UINT :=60,
Position_control_gain_positive_percent_UINT :=60,
Position_control_degressive_gain_percent_UINT :=300,
Speed_precontrol_negative_percent_UINT :=40,
Speed_precontrol_positive_percent_UINT :=40,
Nominal_revolutions_min_UINT :=1800,
Tolerance_positioning_zone_UINT :=20,
Machine_zero_point_DINT :=0,
Software_limit_negative_DINT :=-2000000,
Software_limit_positive_DINT :=2000000,
Reference_point_acceleration_time_ms_UINT :=3000,
Reference_cam_searching_rotational_speed_percent_UINT :=50,
Zero_mark_searching_rotational_speed_percent_UINT :=2,
Contouring_error_maximum_UINT :=3000,
Manual_rotational_speed_percent_UINT :=100,
Manual_acceleration_time_ms_UINT :=2500,
Manual_control_step_width_DINT :=600,
Crawl_speed_rotational_speed_percent_UINT :=0,
Crawl_speed_zone_DINT :=0,
Hydraulic_zero_point_coverage_negative_UINT :=0,
Hydraulic_zero_point_coverage_positive_UINT :=0,
Hydraulic_stick_slip_compensation_negative_11Bit_UINT :=0,
Hydraulic_stick_slip_compensation_positive_11Bit_UINT :=0,
Hydraulic_stick_slip_zone_negative_UINT :=0,
Hydraulic_stick_slip_zone_positive_UINT :=0,
Maximum_incremental_encoder_DINT :=16777215,
Encoder_increments_per_revolution_UINT :=1024
)
```

```
CAL AAINT_INT_analog_output(
        Input_value_INT :=axis_01.Manipulated_variable_12Bit_INT,
        Analog_output_INT=>AO_0_0_2_0
        )

LD      incremental_encoder_02
UDINT_TO_DINT
ST      axis_02.Incremental_encoder_value_DINT

CAL axis_02(
        Setpoint_position_DINT :=Setpoint_position_02_DINT,
        Operating_mode_rotary_axis_BOOL :=0,
        Absolute_value_transmitter_BOOL :=0,
        Absolute_value_transmitter_without_referencing_BOOL :=0,
        Activate_BOOL :=DI_0_1_BOOL,
        Fault_signal_acknowledge_BOOL :=0,
        Manipulated_variable_negation_BOOL :=0,
        Accept_setpoint_position_BOOL :=DI_0_5_BOOL,
        Manual_control_negative_BOOL :=DI_0_2_BOOL,
        Manual_control_positive_BOOL :=DI_0_3_BOOL,
        Manual_control_suppression_before_referencing_BOOL :=1,
        Referencing_negative_BOOL :=DI_0_4_BOOL,
        Referencing_positive_BOOL :=0,
        Reference_signal_cam_BOOL :=Reference_signal_cam_BOOL,
        Emergency_stop_BOOL :=0,
        Loop_controller_ready_BOOL :=1,
        Rotational_speed_negative_percent_UINT :=80,
        Rotational_speed_positive_percent_UINT :=60,
        Acceleration_time_negative_ms_UINT :=500,
        Acceleration_time_positive_ms_UINT :=500,
        Deceleration_time_negative_ms_UINT :=1000,
        Deceleration_time_positive_ms_UINT :=1000,
        Position_control_gain_negative_percent_UINT :=60,
        Position_control_gain_positive_percent_UINT :=60,
        Position_control_degressive_gain_percent_UINT :=300,
```

```
        Speed_precontrol_negative_percent_UINT :=40,
        Speed_precontrol_positive_percent_UINT :=40,
        Nominal_revolutions_min_UINT :=1800,
        Tolerance_positioning_zone_UINT :=20,
        Machine_zero_point_DINT :=0,
        Software_limit_negative_DINT :=-2000000,
        Software_limit_positive_DINT :=2000000,
        Reference_point_acceleration_time_ms_UINT :=3000,
        Reference_cam_searching_rotational_speed_percent_UINT :=50,
        Zero_mark_searching_rotational_speed_percent_UINT :=2,
        Contouring_error_maximum_UINT :=3000,
        Manual_rotational_speed_percent_UINT :=100,
        Manual_acceleration_time_ms_UINT :=2500,
        Manual_control_step_width_DINT :=600,
        Crawl_speed_rotational_speed_percent_UINT :=0,
        Crawl_speed_zone_DINT :=0,
        Hydraulic_zero_point_coverage_negative_UINT :=0,
        Hydraulic_zero_point_coverage_positive_UINT :=0,
        Hydraulic_stick_slip_compensation_negative_11Bit_UINT :=0,
        Hydraulic_stick_slip_compensation_positive_11Bit_UINT :=0,
        Hydraulic_stick_slip_zone_negative_UINT :=0,
        Hydraulic_stick_slip_zone_positive_UINT :=0,
        Maximum_incremental_encoder_DINT :=16777215,
        Encoder_increments_per_revolution_UINT :=1024
        )

CAL AAINT_INT_analog_output(
        Input_value_INT :=axis_02.Manipulated_variable_12Bit_INT,
        Analog_output_INT=>AO_0_0_2_2
        )

END_PROGRAM
```

Basic position control with parameterization options

P_ZSFB01_special_FB
Basic position control with parameterization options in positive and negative direction

P_ZSFB01_special_FB

Type	Inputs:	Outputs:	Type
DINT	Setpoint_position_DINT	Manipulated_variable_12bit_INT	INT
DINT	Actual_position_DINT		
	Parameters:	**Monitor outputs:**	
BOOL	Activate_BOOL	Set_position_current_job_DINT	DINT
BOOL	Accept_setpoint_position_BOOL	Actual_position_in_tolerance_zone_BOOL	BOOL
BOOL	Manipulated_variable_negation_BOOL	Position_deviation_DINT	DINT
BOOL	Cycle_time_demand_optimize_BOOL		
UINT	Ramp_time_100increase_maximum_negative_ms_UINT		
UINT	Ramp_time_100increase_maximum_positive_ms_UINT		
UDINT	Deceleration_position_deviation_negative_UDINT		
UDINT	Deceleration_position_deviation_positive_UDINT		
UINT	Rounding_position_deviation_negative_15bit_UINT		
UINT	Rounding_position_deviation_positive_15bit_UINT		
UINT	Manipulated_variable_max_neg_11bit_UINT		
UINT	Manipulated_variable_max_pos_11bit_UINT		
UINT	Tolerance_positioning_zone_UINT		

Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Setpoint_position_DINT	Setpoint position (target position)	-10^9 to 10^9
Actual_position_DINT	Actual position	-10^9 to 10^9
Parameters		
Activate_BOOL	Activate (deactivate = reset) the function block	0/1
Manipulated_variable_negation_ BOOL	Negate manipulated variable (reverse prefix)	0/1
Accept_setpoint_position_BOOL	Control mode: Accept applied setpoint positions (automatic mode). This mode overrides manual mode.	0/1
Cycle_time_demand_optimize_ BOOL	Control mode: The cycle time demand of the function block can be optimized at the expense of the rounding transition at the end of the positioning process	0/1
Ramp_time_100increase_ maximum_negative_ms_UINT	Negative ramp time required for an increase of 100 increments at a setpoint position change	0 to 65535
Ramp_time_100increase_ maximum_positive_ms_UINT	Positive ramp time required for an increase of 100 increments at a setpoint position change	0 to 65535
Deceleration_position_deviation_ negative_UDINT	Negative position deviation from which a ramped delay is applied to the manipulated variable.	0 to 10^9
Deceleration_position_deviation_ positive_UDINT	Positive position deviation from which a ramped delay is applied to the manipulated variable	0 to 10^9
Rounding_position_deviation_ negative_15bit_UINT	Negative position deviation from which rounding of the manipulated variable delay begins (→ fig. 7).	0 to 32767
Rounding_position_deviation_ positive_15bit_UINT	Positive position deviation from which rounding of the manipulated variable delay begins (→ fig. 7)	0 to 32767

Designation	Function	Value range
Manipulated_variable_max_neg_11bit_UINT	Maximum negative manipulated variable	0 to 2048
Manipulated_variable_max_pos_11bit_UINT	Maximum positive manipulated variable	0 to 2047
Tolerance_positioning_zone_UINT	Tolerance range (±) of the positioning zone	0/1
Outputs		
Manipulated_variable_12bit_INT	Manipulated variable for controlling the axis actuators (setpoint rotational speed, setpoint linear speed)	–2048 to 2047
Monitor outputs		
Set_position_current_job_DINT	Setpoint position of current job	-10^9 to 10^9
Position_deviation_DINT	Position deviation	-10^9 to 10^9
Actual_position_in_tolerance_zone_BOOL	Status: Actual position is within tolerance zone	0/1

Description

See function block "P_basic_position_control" from page 23. The only difference to "P_basic_position_control" is that the parameters of this function block can be entered in both positive and negative direction.

Combination of basic position control and referencing

P_ZSFB02_special_FB Combination of basic position control and referencing

P_ZSFB02_special_FB

Type	Inputs:	Outputs:	Type
DINT	Setpoint_position_DINT	Manipulated_variable_12bit_INT	INT
DINT	Increments_DINT		

Type	Parameters:	Monitor outputs:	Type
BOOL	Absolute_value_transmitter_BOOL	Set_position_current_job_DINT	DINT
BOOL	Absolute_value_transmitter_without_ referencing_BOOL	Actual_position_DINT	DINT
		Position_deviation_DINT	DINT
BOOL	Activate_BOOL	Actual_position_in_tolerance_zone_BOOL	BOOL
BOOL	Manipulated_variable_negation_BOOL	Parameterization_error_BOOL	BOOL
BOOL	Cycle_time_demand_optimize_BOOL	Software_limit_attained_BOOL	BOOL
BOOL	Accept_setpoint_position_BOOL	Referencing_finished_BOOL	BOOL
BOOL	Manual_control_negative_BOOL		
BOOL	Manual_control_positive_BOOL		
BOOL	Referencing_negative_BOOL		
BOOL	Referencing_positive_BOOL		
BOOL	Reference_signal_cam_BOOL		
UINT	Manual_manipulated_variable_11Bit_UINT		
DINT	Manual_control_step_width_DINT		
DINT	Machine_zero_point_DINT		
DINT	Software_limit_negative_DINT		
DINT	Software_limit_positive_DINT		
UINT	Reference_cam_searching_manipulated_variable_11Bit_UINT		
UINT	Reference_point_searching_manipulated_variable_11Bit_UINT		
UINT	Ramp_time_100increase_maximum_negative_UINT		
UINT	Ramp_time_100increase_maximum_positive_UINT		
UDINT	Deceleration_position_deviation_negative_UDINT		
UDINT	Deceleration_position_deviation_positive_UDINT		
UINT	Rounding_position_deviation_negative_15bit_UINT		
UINT	Rounding_position_deviation_positive_15bit_UINT		
UINT	Manipulated_variable_max_neg_11bit_UINT		
UINT	Manipulated_variable_max_pos_11bit_UINT		
UINT	Tolerance_positioning_zone_UINT		
DINT	Maximum_incremental_encoder_DINT		

Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Setpoint_position_DINT	Setpoint position of the automatic control. During referencing and manual operation, this value is overridden.	-10^9 to 10^9
Increments_DINT	Reading from the incremental or absolute encoder	-10^9 to 10^9
Parameters		
Absolute_value_transmitter_ BOOL	Transmitter type: "Incremental encoder" = "0", "Absolute encoder" = "1"; For "Incremental encoder", referencing with a reference cam scan can be carried out. With "Absolute encoder", referencing is carried out without reference cam scan.	0/1
Absolute_value_transmitter_ without_referencing_BOOL	In this mode, no referencing is carried out.	0/1
Activate_BOOL	Activates the function block	0/1
Manipulated_variable_negation_ BOOL	Negate manipulated variable (reverse prefix)	0/1
Cycle_time_demand_optimize_ BOOL	Control mode: The cycle time demand of the function block can be optimized at the expense of the rounding transition at the end of the positioning process	0/1
Accept_setpoint_position_BOOL	Control mode: Accept applied setpoint positions (automatic mode). This mode overrides manual mode.	0/1
Manual_control_negative_BOOL	Control mode: Manual control (and jogging for non-zero manual control step widths ≠ 0), negative	0/1
Manual_control_positive_BOOL	Control mode: Manual control (and jogging for non-zero manual control step widths ≠ 0), positive	0/1

Designation	Function	Value range
Referencing_negative_BOOL	Control mode: negative referencing	0/1
Referencing_positive_BOOL	Control mode: positive referencing	0/1
Reference_signal_cam_BOOL	Signal: Reference cam	0/1
Manual_manipulated_variable_ 11Bit_UINT	Maximum manipulated variable for manual mode	0 to 2048
Manual_control_step_width_ DINT	Step width for manual mode (values ≠ 0 => jogging)	-10^9 to 10^9
Machine_zero_point_DINT	Machine zero point	-10^9 to 10^9
Software_limit_negative_DINT	Negative software limit (effective only after referencing)	-10^9 to 10^9
Software_limit_positive_DINT	Positive software limit (effective only after referencing)	-10^9 to 10^9
Reference_cam_searching_ manipulated_variable_11Bit_UINT	Manipulated variable for reference cam searching	0 to 2048
Reference_point_searching_ manipulated_variable_11Bit_UINT	Manipulated variable for reference point searching (edge of reference cam)	0 to 2048
Ramp_time_100increase_ maximum_negative_ms_UINT	Negative ramp time required for an increase of 100 increments at a setpoint position change	0 to 65535
Ramp_time_100increase_ maximum_positive_ms_UINT	Positive ramp time required for an increase of 100 increments at a setpoint position change	0 to 65535
Deceleration_position_deviation_ negative_UDINT	Negative position deviation from which a ramped delay is applied to the manipulated variable	0 to 10^9
Deceleration_position_deviation_ positive_UDINT	Positive position deviation from which a ramped delay is applied to the manipulated variable	0 to 10^9
Rounding_position_deviation_ negative_15bit_UINT	Negative position deviation from which rounding of the manipulated variable delay begins (→ fig. 7).	0 to 32767
Rounding_position_deviation_ positive_15bit_UINT	Positive position deviation from which rounding of the manipulated variable delay begins (→ fig. 7).	0 to 32767

Designation	Function	Value range
Manipulated_variable_max_ neg_11bit_UINT	Maximum negative manipulated variable for automatic mode	0 to 2048
Manipulated_variable_max_ pos_11bit_UINT	Maximum positive manipulated variable for automatic mode	0 to 2047
Manipulated_variable_max_ positive_11bit_UINT	Maximum positive manipulated variable	0 to 2047
Tolerance_positioning_zone_UINT	Tolerance range (±) of the positioning zone	0/1
Maximum_incremental_encoder_ DINT	Maximum value from the incremental encoder (before overflow)	0/1
Outputs		
Manipulated_variable_12bit_INT	Manipulated variable for controlling the axis actuators (setpoint rotational speed, setpoint linear speed)	–2048 to 2047
Monitor outputs		
Set_position_current_job_DINT	Setpoint position of current job	-10^9 to 10^9
Actual_position_DINT	Actual position (= incremental encoder value + machine zero point + overflow correction of incremental encoder => negative values possible)	-10^9 to 10^9
Position_deviation_DINT	Position deviation	-10^9 to 10^9
Actual_position_in_tolerance_ zone_BOOL	Status: Actual position is within tolerance zone.	0/1
Parameterization_error_BOOL	Status: Parameterization error, e.g. simultaneous referencing in positive and negative direction	0/1
Software_limit_attained_BOOL	Status: The setpoint position lies outside the software limits and is constrained to within the software limits	0/1
Referencing_finished_BOOL	Status: Referencing finished	0/1

Description

See function blocks “P_ZSFB01_special_FB” and “P_referencing”. In this function block, “P_ZSFB01_special_FB” and “P_referencing” are linked to each other, as shown in the application example for “P_referencing”.

4 Sequencer

Ten-step sequencer

P_Sq10_sequencer
Ten-step sequencer

P_Sq10_sequencer

Type	Inputs:	Outputs:	Type
BOOL	Positioning_job_finished_BOOL	Axis1_selected_BOOL	BOOL
BOOL	Actual_position_in_tolerance_zone_BOOL	Axis2_selected_BOOL	BOOL
		Axis3_selected_BOOL	BOOL
		Rotational_speed_axis1_UINT	UINT
		Rotational_speed_axis2_UINT	UINT
		Rotational_speed_axis3_UINT	UINT
		Setpoint_position_axis1_DINT	DINT
		Setpoint_position_axis2_DINT	DINT
		Setpoint_position_axis3_DINT	DINT
		External_value1_INT	INT
		External_value2_INT	INT
		External_value3_INT	INT
		External_value4_INT	INT
		External_value5_INT	INT
		External_value6_INT	INT

Type	Parameters:	Monitor outputs:	Type
BOOL	Activate_BOOL	Step_number_USINT	USINT
USINT	Number_of_steps_USINT		
BOOL	Progression_condition_actual_position_in_tolerance_zone_BOOL		
BOOL	External_switching_condition_step1_BOOL		
BOOL	Enable_step1_BOOL		
BOOL	Selection_step1_axis1_BOOL		
BOOL	Selection_step1_axis2_BOOL		
BOOL	Selection_step1_axis3_BOOL		
UINT	Rotational_speed_step1_axis1_UINT		
UINT	Rotational_speed_step1_axis2_UINT		
UINT	Rotational_speed_step1_axis3_UINT		
DINT	Setpoint_position_step1_axis1_DINT		
DINT	Setpoint_position_step1_axis2_DINT		
DINT	Setpoint_position_step1_axis3_DINT		

	Parameters (continues):
INT	External_value1_step1_INT
INT	External_value2_step1_INT
INT	External_value3_step1_INT
INT	External_value4_step1_INT
INT	External_value5_step1_INT
INT	External_value6_step1_INT
.	.
.	.
.	.
BOOL	External_switching_condition_step10_BOOL
BOOL	Enable_step10_BOOL
BOOL	Selection_step10_axis1_BOOL
BOOL	Selection_step10_axis2_BOOL
BOOL	Selection_step10_axis3_BOOL
UINT	Rotational_speed_step10_axis1_UINT
UINT	Rotational_speed_step10_axis2_UINT
UINT	Rotational_speed_step10_axis3_UINT
DINT	Setpoint_position_step10_axis1_DINT
DINT	Setpoint_position_step10_axis2_DINT
DINT	Setpoint_position_step10_axis3_DINT
INT	External_value1_step10_INT
INT	External_value2_step10_INT
INT	External_value3_step10_INT
INT	External_value4_step10_INT
INT	External_value5_step10_INT
INT	External_value6_step10_INT

Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Positioning_job_finished_BOOL	Stepping condition: the positioning job is finished	0/1
Actual_position_in_tolerance_zone_BOOL	Stepping condition: actual position inside the tolerance zone	0/1
Parameters		
Activate_BOOL	Activates the function block. When it is deactivated (input of zero), the function block is reset	0/1
Number_of_steps_USINT	Number of steps	0 to 10
Progression_condition_actual_position_in_tolerance_zone_BOOL	Dependence of stepping condition selection: "1" => Actual_position_in_tolerance_zone_BOOL = 0/1 "0" => Positioning_job_finished_BOOL = 0/1	0/1
External_switching_condition_step1_BOOL	Step 1: Switching condition external (outside a positioning operation)	0/1
Enable_step1_BOOL	Step 1: Enable	0/1
Selection_step1_axis1_BOOL	Step 1: Selection of axis 1	0/1
Selection_step1_axis2_BOOL	Step 1: Selection of axis 2	0/1
Selection_step1_axis3_BOOL	Step 1: Selection of axis 3	0/1
Rotational_speed_step1_axis1_UINT	Step 1: Speed (rotational or linear), axis 1	0 to 65535
Rotational_speed_step1_axis2_UINT	Step 1: Speed (rotational or linear), axis 2	0 to 65535
Rotational_speed_step1_axis3_UINT	Step 1: Speed (rotational or linear), axis 3	0 to 65535

Designation	Function	Value range
Setpoint_position_step1_axis1_DINT	Step 1: Setpoint position (target position), axis 1	-10^9 to 10^9
Setpoint_position_step1_axis2_DINT	Step 1: Setpoint position (target position), axis 2	-10^9 to 10^9
Setpoint_position_step1_axis3_DINT	Step 1: Setpoint position (target position), axis 3	-10^9 to 10^9
External_value1_step1_INT	Step 1: External, or any value 1	–32 768 to 32 767
External_value2_step1_INT	Step 1: External, or any value 2	–32 768 to 32 767
External_value3_step1_INT	Step 1: External, or any value 3	–32 768 to 32 767
External_value4_step1_INT	Step 1: External, or any value 4	–32 768 to 32 767
External_value5_step1_INT	Step 1: External, or any value 5	–32 768 to 32 767
External_value6_step1_INT	Step 1: External, or any value 6	–32 768 to 32 767
. . .	. . .	. . .
External_switching_condition_ step10_BOOL	Step 10: Switching condition external (outside a positioning operation)	0/1
Enable_step10_BOOL	Step 10: Enable	0/1
Selection_step10_axis1_BOOL	Step 10: Selection of axis 1	0/1
Selection_step10_axis2_BOOL	Step 10: Selection of axis 2	0/1
Selection_step10_axis3_BOOL	Step 10: Selection of axis 3	0/1
Rotational_speed_step10_axis1_ UINT	Step 10: Speed (linear speed), axis 1	0 to 65 535
Rotational_speed_step10_axis2_ UINT	Step 10: Speed (linear speed), axis 2	0 to 65 535
Rotational_speed_step10_axis3_ UINT	Step 10: Speed (linear speed), axis 3	0 to 65 535
Setpoint_position_step10_axis1_ DINT	Step 10: Setpoint position (target position), axis 1	-10^9 to 10^9
Setpoint_position_step10_axis2_ DINT	Step 10: Setpoint position (target position), axis 2	-10^9 to 10^9
Setpoint_position_step10_axis3_ DINT	Step 10: Setpoint position (target position), axis 3	-10^9 to 10^9

Designation	Function	Value range
External_value1_step10_INT	Step 10: External, or any value 1	–32 768 to 32 767
External_value2_step10_INT	Step 10: External, or any value 2	–32 768 to 32 767
External_value3_step10_INT	Step 10: External, or any value 3	–32 768 to 32 767
External_value4_step10_INT	Step 10: External, or any value 4	–32 768 to 32 767
External_value5_step10_INT	Step 10: External, or any value 5	–32 768 to 32 767
External_value6_step10_INT	Step 10: External, or any value 6	–32 768 to 32 767
Outputs		
Axis1_selected_BOOL	Axis 1 was selected in this step	0/1
Axis2_selected_BOOL	Axis 2 was selected in this step	0/1
Axis3_selected_BOOL	Axis 3 was selected in this step	0/1
Rotational_speed_axis1_UINT	Speed (rotational or linear), axis 1	0 to 65 535
Rotational_speed_axis2_UINT	Speed (rotational or linear), axis 2	0 to 65 535
Rotational_speed_axis3_UINT	Speed (rotational or linear), axis 3	0 to 65 535
Setpoint_position_axis1_DINT	Setpoint position (target position), axis 1	-10^9 to 10^9
Setpoint_position_axis2_DINT	Setpoint position (target position), axis 2	-10^9 to 10^9
Setpoint_position_axis3_DINT	Setpoint position (target position), axis 3	-10^9 to 10^9
External_value1_INT	External or any value 1	–32 768 to 32 767
External_value2_INT	External or any value 2	–32 768 to 32 767
External_value3_INT	External or any value 3	–32 768 to 32 767
External_value4_INT	External or any value 4	–32 768 to 32 767
External_value5_INT	External or any value 5	–32 768 to 32 767
External_value6_INT	External or any value 6	–32 768 to 32 767
Monitor outputs		
Step_number_USINT	Step number of current step	0 to 10

Description

The sequencer can be used in conjunction with the position control function block. The number of steps can be specified with "Number_ of _steps_USINT". The steps are executed in ascending order. Individual steps can be enabled separately or skipped. Input "Positioning_job_finished_BOOL" or input "Actual_position_in_tolerance_zone_BOOL" can provide the stepping condition. Stepping is also subject to an additional external stepping condition. If the sequencer is not used to control processes other than conditioning, the external stepping conditions can be set to "1".

For positioning procedures (jobs), the following can be specified for three axes:

- Rotational or linear speed (per step)
- Setpoint position
- Six external (or any) values

The steps are processed sequentially. The outputs always carry the values that were entered for the current step. The current step is indicated by monitor output "Step_number_USINT".

For many positioning tasks, special sequencers are required. In general, these can be easily programmed. Function block "P_Sq10_sequencer" is primarily intended as an example of a sequencer. In particular in combination with axis simulation, this sequencer can be used for creating sample programs for the position control function blocks.

Example:
The application example implements positioning of two axes. The actual positions are determined with an incremental encoder. The referencing operation can be started in the negative direction with digital input "4". Digital input "1", activates the closed-loop positioning controllers. With digital inputs "2" and "3", manual control in the negative and positive direction is activated. Because "Manual_control_step_width_DINT" contains a value of "0", the axis traverses up to the software limits in manual control mode.

Automatic mode can be activated with digital input “5”. The setpoint positions, the (setpoint) speeds and the acceleration and deceleration times are defined by the sequencer for 10 steps. When the actual position has reached the tolerance zone, the new, pending, setpoint position and all parameters defined by the sequencer are accepted by the two closed-loop control function blocks. The next step is issued only when both axes have reached the target tolerance zone.

Application of function block “P_Sq10_sequencer” in program “Pos_05”

```
PROGRAM Pos_05
VAR
        Sq10_sequencer : P_Sq10_sequencer ;
        axis_01 : P_closed_loop_position_control ;
        axis_02 : P_closed_loop_position_control ;
        AAINT_INT_analog_output : P_AAINT_INT_analog_output ;
        DI_0_0_BOOL : BOOL ;
        DI_0_1_BOOL : BOOL ;
        DI_0_2_BOOL : BOOL ;
        DI_0_3_BOOL : BOOL ;
        DI_0_4_BOOL : BOOL ;
        DI_0_5_BOOL : BOOL ;
        Reference_signal_cam_BOOL : BOOL ;
        AO_0_0_2_0 : INT ;
        AO_0_0_2_2 : INT ;
        incremental_encoder_01 : UDINT ;
        incremental_encoder_02 : UDINT ;
END_VAR
LD      axis_01.Actual_position_in_tolerance_zone_BOOL
AND     axis_02.Actual_position_in_tolerance_zone_BOOL
ST      Sq10_sequencer.Actual_position_in_tolerance_zone_BOOL
```

```
CAL Sq10_sequencer(
        Positioning_job_finished_BOOL :=0,
        Activate_BOOL :=DI_0_0_BOOL,
        Number_of_steps_USINT :=10,
        Progression_condition_actual_position_in_tolerance_zone_BOOL :=1,
        External_switching_condition_step1_BOOL :=1,
        Selection_step1_axis1_BOOL :=1,
        Selection_step1_axis2_BOOL :=1,
        Selection_step1_axis3_BOOL :=1,
        Rotational_speed_step1_axis1_UINT :=50,
        Rotational_speed_step1_axis2_UINT :=50,
        Rotational_speed_step1_axis3_UINT :=0,
        Setpoint_position_step1_axis1_DINT :=10000,
        Setpoint_position_step1_axis2_DINT :=10000,
        Setpoint_position_step1_axis3_DINT :=0,
        External_value4_step1_INT :=2000,
        External_value5_step1_INT :=3000,
        External_switching_condition_step2_BOOL :=1,
        Selection_step2_axis1_BOOL :=1,
        Selection_step2_axis2_BOOL :=1,
        Selection_step2_axis3_BOOL :=0,
        Rotational_speed_step2_axis1_UINT :=100,
        Rotational_speed_step2_axis2_UINT :=100,
        Rotational_speed_step2_axis3_UINT :=0,
        Setpoint_position_step2_axis1_DINT :=12000,
        Setpoint_position_step2_axis2_DINT :=-20000,
        Setpoint_position_step2_axis3_DINT :=0,
        External_value4_step2_INT :=2000,
        External_value5_step2_INT :=1500,
        External_switching_condition_step3_BOOL :=1,
        Enable_step3_BOOL :=1,
        Selection_step3_axis1_BOOL :=1,
        Selection_step3_axis2_BOOL :=1,
        Selection_step3_axis3_BOOL :=0,
```

```
        Rotational_speed_step3_axis1_UINT :=100,
        Rotational_speed_step3_axis2_UINT :=100,
        Rotational_speed_step3_axis3_UINT :=0,
        Setpoint_position_step3_axis1_DINT :=40000,
        Setpoint_position_step3_axis2_DINT :=30000,
        Setpoint_position_step3_axis3_DINT :=0,
        External_value4_step3_INT :=500,
        External_value5_step3_INT :=500,
        External_switching_condition_step4_BOOL :=1,
        Selection_step4_axis1_BOOL :=1,
        Selection_step4_axis2_BOOL :=1,
        Selection_step4_axis3_BOOL :=0,
        Rotational_speed_step4_axis1_UINT :=100,
        Rotational_speed_step4_axis2_UINT :=100,
        Rotational_speed_step4_axis3_UINT :=0,
        Setpoint_position_step4_axis1_DINT :=25000,
        Setpoint_position_step4_axis2_DINT :=-25000,
        Setpoint_position_step4_axis3_DINT :=0,
        External_value4_step4_INT :=500,
        External_value5_step4_INT :=1000,
        External_switching_condition_step5_BOOL :=1,
        Selection_step5_axis1_BOOL :=1,
        Selection_step5_axis2_BOOL :=1,
        Selection_step5_axis3_BOOL :=0,
        Rotational_speed_step5_axis1_UINT :=100,
        Rotational_speed_step5_axis2_UINT :=100,
        Setpoint_position_step5_axis1_DINT :=45000,
        Setpoint_position_step5_axis2_DINT :=45000,
        Setpoint_position_step5_axis3_DINT :=0,
        External_value4_step5_INT :=100,
        External_value5_step5_INT :=1000,
        External_switching_condition_step6_BOOL :=1,
        Selection_step6_axis1_BOOL :=1,
        Selection_step6_axis2_BOOL :=1,
        Selection_step6_axis3_BOOL :=0,
```

```
        Rotational_speed_step6_axis1_UINT :=98,
        Rotational_speed_step6_axis2_UINT :=98,
        Setpoint_position_step6_axis1_DINT :=30000,
        Setpoint_position_step6_axis2_DINT :=30000,
        Setpoint_position_step6_axis3_DINT :=0,
        External_value4_step6_INT :=1500,
        External_value5_step6_INT :=1000,
        External_switching_condition_step7_BOOL :=1,
        Selection_step7_axis1_BOOL :=1,
        Selection_step7_axis2_BOOL :=1,
        Selection_step7_axis3_BOOL :=0,
        Rotational_speed_step7_axis1_UINT :=95,
        Rotational_speed_step7_axis2_UINT :=95,
        Setpoint_position_step7_axis1_DINT :=10000,
        Setpoint_position_step7_axis2_DINT :=-15000,
        External_value4_step7_INT :=3000,
        External_value5_step7_INT :=3000,
        External_switching_condition_step8_BOOL :=1,
        Selection_step8_axis1_BOOL :=1,
        Selection_step8_axis2_BOOL :=0,
        Selection_step8_axis3_BOOL :=0,
        Rotational_speed_step8_axis1_UINT :=100,
        Rotational_speed_step8_axis2_UINT :=100,
        Setpoint_position_step8_axis1_DINT :=30000,
        Setpoint_position_step8_axis2_DINT :=30000,
        Setpoint_position_step8_axis3_DINT :=0,
        External_value4_step8_INT :=1500,
        External_value5_step8_INT :=1500,
        External_switching_condition_step9_BOOL :=1,
        Selection_step9_axis1_BOOL :=1,
        Selection_step9_axis2_BOOL :=1,
        Selection_step9_axis3_BOOL :=0,
        Rotational_speed_step9_axis1_UINT :=80,
        Rotational_speed_step9_axis2_UINT :=80,
```

```
        Setpoint_position_step9_axis1_DINT :=5000,
        Setpoint_position_step9_axis2_DINT :=10000,
        Setpoint_position_step9_axis3_DINT :=0,
        External_value4_step9_INT :=1000,
        External_value5_step9_INT :=1000,
        External_switching_condition_step10_BOOL :=1,
        Selection_step10_axis1_BOOL :=1,
        Selection_step10_axis2_BOOL :=1,
        Selection_step10_axis3_BOOL :=0,
        Rotational_speed_step10_axis1_UINT :=100,
        Rotational_speed_step10_axis2_UINT :=100,
        Setpoint_position_step10_axis1_DINT :=50000,
        Setpoint_position_step10_axis2_DINT :=50000,
        Setpoint_position_step10_axis3_DINT :=0,
        External_value4_step10_INT :=1600,
        External_value5_step10_INT :=500
        )
LD      Sq10_sequencer.Setpoint_position_axis1_DINT
ST      axis_01.Setpoint_position_DINT

LD      Sq10_sequencer.Rotational_speed_axis1_UINT
ST      axis_01.Rotational_speed_negative_percent_UINT
ST      axis_01.Rotational_speed_positive_percent_UINT

LD      Sq10_sequencer.Setpoint_position_axis2_DINT
ST      axis_02.Setpoint_position_DINT

LD      Sq10_sequencer.Rotational_speed_axis2_UINT
ST      axis_02.Rotational_speed_negative_percent_UINT
ST      axis_02.Rotational_speed_positive_percent_UINT
```

```
LD      Sq10_sequencer.External_value4_INT
INT_TO_UINT
ST      axis_01.Acceleration_time_negative_ms_UINT
ST      axis_01.Acceleration_time_positive_ms_UINT
ST      axis_01.Deceleration_time_negative_ms_UINT
ST      axis_01.Deceleration_time_positive_ms_UINT

LD      Sq10_sequencer.External_value5_INT
INT_TO_UINT
ST      axis_02.Acceleration_time_negative_ms_UINT
ST      axis_02.Acceleration_time_positive_ms_UINT
ST      axis_02.Deceleration_time_negative_ms_UINT
ST      axis_02.Deceleration_time_positive_ms_UINT

LD      incremental_encoder_01
UDINT_TO_DINT
ST      axis_01.Incremental_encoder_value_DINT

CAL axis_01(
        Operating_mode_rotary_axis_BOOL :=0,
        Absolute_value_transmitter_BOOL :=0,
        Absolute_value_transmitter_without_referencing_BOOL :=0,
        Activate_BOOL :=DI_0_1_BOOL,
        Fault_signal_acknowledge_BOOL :=0,
        Manipulated_variable_negation_BOOL :=0,
        Accept_setpoint_position_BOOL :=DI_0_5_BOOL,
        Manual_control_negative_BOOL :=DI_0_2_BOOL,
        Manual_control_positive_BOOL :=DI_0_3_BOOL,
        Manual_control_suppression_before_referencing_BOOL :=1,
        Referencing_negative_BOOL :=DI_0_4_BOOL,
        Referencing_positive_BOOL :=0,
        Reference_signal_cam_BOOL :=Reference_signal_cam_BOOL,
        Emergency_stop_BOOL :=0,
        Loop_controller_ready_BOOL :=1,
```

```
        Position_control_gain_negative_percent_UINT :=60,
        Position_control_gain_positive_percent_UINT :=60,
        Position_control_degressive_gain_percent_UINT :=300,
        Speed_precontrol_negative_percent_UINT :=40,
        Speed_precontrol_positive_percent_UINT :=40,
        Nominal_revolutions_min_UINT :=1800,
        Tolerance_positioning_zone_UINT :=20,
        Machine_zero_point_DINT :=0,
        Software_limit_negative_DINT :=-2000000,
        Software_limit_positive_DINT :=2000000,
        Reference_point_acceleration_time_ms_UINT :=3000,
        Reference_cam_searching_rotational_speed_percent_UINT :=50,
        Zero_mark_searching_rotational_speed_percent_UINT :=2,
        Contouring_error_maximum_UINT :=3000,
        Manual_rotational_speed_percent_UINT :=100,
        Manual_acceleration_time_ms_UINT :=2500,
        Manual_control_step_width_DINT :=0,
        Crawl_speed_rotational_speed_percent_UINT :=0,
        Crawl_speed_zone_DINT :=0,
        Hydraulic_zero_point_coverage_negative_UINT :=0,
        Hydraulic_zero_point_coverage_positive_UINT :=0,
        Hydraulic_stick_slip_compensation_negative_11Bit_UINT :=0,
        Hydraulic_stick_slip_compensation_positive_11Bit_UINT :=0,
        Hydraulic_stick_slip_zone_negative_UINT :=0,
        Hydraulic_stick_slip_zone_positive_UINT :=0,
        Maximum_incremental_encoder_DINT :=16777215,
        Encoder_increments_per_revolution_UINT :=1024
        )
CAL AAINT_INT_analog_output(
        Input_value_INT :=axis_01.Manipulated_variable_12Bit_INT,
        Analog_output_INT=>AO_0_0_2_0
        )
LD      incremental_encoder_02
UDINT_TO_DINT
ST      axis_02.Incremental_encoder_value_DINT
```

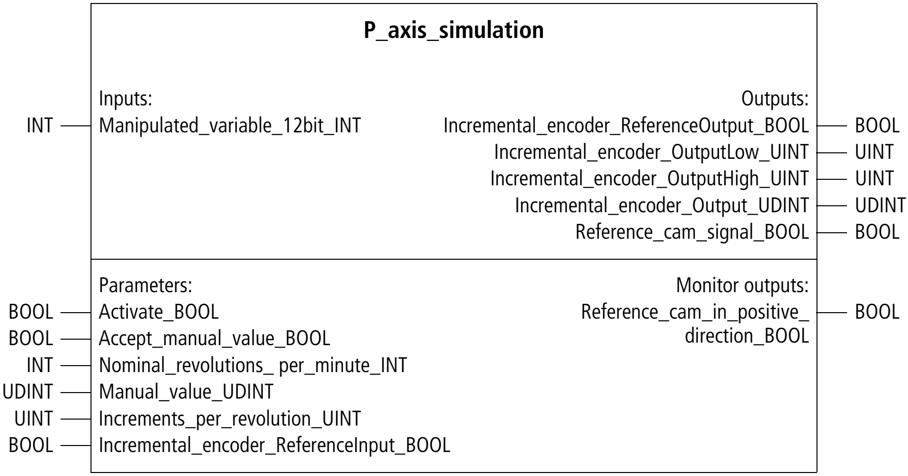
```
CAL axis_02(
        Operating_mode_rotary_axis_BOOL :=0,
        Absolute_value_transmitter_BOOL :=0,
        Absolute_value_transmitter_without_referencing_BOOL :=0,
        Activate_BOOL :=DI_0_1_BOOL,
        Fault_signal_acknowledge_BOOL :=0,
        Manipulated_variable_negation_BOOL :=0,
        Accept_setpoint_position_BOOL :=DI_0_5_BOOL,
        Manual_control_negative_BOOL :=DI_0_2_BOOL,
        Manual_control_positive_BOOL :=DI_0_3_BOOL,
        Manual_control_suppression_before_referencing_BOOL :=1,
        Referencing_negative_BOOL :=DI_0_4_BOOL,
        Referencing_positive_BOOL :=0,
        Reference_signal_cam_BOOL :=Reference_signal_cam_BOOL,
        Emergency_stop_BOOL :=0,
        Loop_controller_ready_BOOL :=1,
        Position_control_gain_negative_percent_UINT :=60,
        Position_control_gain_positive_percent_UINT :=60,
        Position_control_degressive_gain_percent_UINT :=300,
        Speed_precontrol_negative_percent_UINT :=40,
        Speed_precontrol_positive_percent_UINT :=40,
        Nominal_revolutions_min_UINT :=1800,
        Tolerance_positioning_zone_UINT :=20,
        Machine_zero_point_DINT :=0,
        Software_limit_negative_DINT :=-2000000,
        Software_limit_positive_DINT :=2000000,
        Reference_point_acceleration_time_ms_UINT :=3000,
        Reference_cam_searching_rotational_speed_percent_UINT :=50,
        Zero_mark_searching_rotational_speed_percent_UINT :=2,
        Contouring_error_maximum_UINT :=3000,
        Manual_rotational_speed_percent_UINT :=100,
        Manual_acceleration_time_ms_UINT :=2500,
        Manual_control_step_width_DINT :=0,
```

```
        Crawl_speed_rotational_speed_percent_UINT :=0,
        Crawl_speed_zone_DINT :=0,
        Hydraulic_zero_point_coverage_negative_UINT :=0,
        Hydraulic_zero_point_coverage_positive_UINT :=0,
        Hydraulic_stick_slip_compensation_negative_11Bit_UINT :=0,
        Hydraulic_stick_slip_compensation_positive_11Bit_UINT :=0,
        Hydraulic_stick_slip_zone_negative_UINT :=0,
        Hydraulic_stick_slip_zone_positive_UINT :=0,
        Maximum_incremental_encoder_DINT :=16777215,
        Encoder_increments_per_revolution_UINT :=1024
        )
CAL AAINT_INT_analog_output(
        Input_value_INT :=axis_02.Manipulated_variable_12Bit_INT,
        Analog_output_INT=>AO_0_0_2_2
        )
END_PROGRAM
```

5 Simulation

Axis simulation

P_axis_simulation
Simulating a positioning axis



Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Manipulated_variable_12bit_INT	Manipulated variable that can be linked with the position control function block	–2048 to 2048
Parameters		
Activate_BOOL	Activates the function block	0/1
Accept_manual_value_BOOL	Command: Accept manual value	0/1
Nominal_revolutions_ per_minute_INT	Nominal speed [revolutions per minute]	0 to 10000

Designation	Function	Value range
Manual_value_UDINT	Manual value	0 to 16777215
Increments_per_revolution_UINT	Increments per revolution	1 to 4096
Incremental_encoder_ ReferenceInput_BOOL	Reference input signal from the incremental encoder	0/1
Outputs		
Incremental_encoder_ ReferenceOutput_BOOL	Incremental encoder: Reference output signal	0/1
Incremental_encoder_ OutputLow_UINT	Incremental encoder: Output value low	0 to 65535
Incremental_encoder_ OutputHigh_UINT	Incremental encoder: Output value high	0 to 255
Incremental_encoder_Output_UDINT	Incremental encoder: Output value "double word"	0 to 16777215
Reference_cam_signal_BOOL	Reference cam signal	0/1
Monitor outputs		
Reference_cam_in_positive_ direction_BOOL	The reference cam is positioned in a positive direction	0/1

Description

This function block simulates a positioning axis. In the input range, parameter "Manipulated_variable_12bit_INT" must be linked with the output variables of a position controller. If a value of "1" has been assigned to parameter "Accept_manual_value_BOOL", the manually entered value is accepted. The simulation then begins with this "position value" when "Accept_manual_value_BOOL" is reset to "0". With parameters "Nominal_revolutions_ per_minute_INT" and "Increments_per_revolution_UINT", the output behaviour of the function block can be influenced.

Example 1:
Manipulated_variable_12bit_INT = –2048
(= negative maximum)
Nominal_revolutions_ per_minute_INT = 1200
Increments_per_revolution_UINT = 1024

$$\text{Increments per minute} = \frac{-2048}{2048} \times 1200 \times 1024 \text{ [1/min]} = -1228800 \text{ [1/min]} = -20480 \text{ [1/s]}$$

Example 2:
Manipulated_variable_12bit_INT = 1000
Nominal_revolutions_ per_minute_INT = 400
Increments_per_revolution_UINT = 100

$$\text{Increments per minute} = \frac{1000}{2048} \times 400 \times 100 \text{ [1/min]} = 19531 \text{ [1/min]} = 325 \text{ [1/s]}$$

Input variable "Incremental_encoder_ ReferenceInput_BOOL" can be linked with the output signal from function block "P_closed loop_position_control". At a falling edge, the a reset of the incremental outputs is triggered.

Outputs "Incremental_encoder_ReferenceOutput_BOOL" and "Reference_cam_signal_BOOL" must be linked with the corresponding inputs of "P_closed_loop_position_control". Outputs "Incremental_encoder_OutputLow_UINT", "Incremental_encoder_OutputHigh_UINT" and "Incremental_encoder_Output_UDINT" can be used as actual positions or incremental encoder inputs. Monitor output "reference_cam_in_positive_direction_BOOL" indicates if a reference cam signal will occur in the positive or negative direction.

Example:
In this application example, three axis simulations are called up in connection with three different positioning function blocks. The digital input signals are assigned as follows:

- Digital input "0" = Activate the sequencer
- Digital input "1" = Activate the positioning function blocks
- Digital input "2" = Manual mode for axis 1
- Digital input "3" = Manual mode, positive, for axis 1
- Digital input "4" = Referencing
- Digital input "5" = Automatic mode (Accept setpoint positions) axis 1
- Digital input "6" = Activate the simulation
- Digital input "7" = Absolute encoder = "1"/ incremental encoder = "0" for axis 1

To facilitate visual monitoring of the positioning processes, suitably high acceleration and deceleration time settings and a low number of increments per revolution have been chosen.

Function blocks "P_DINT_characteristics_control" and "P_basic_position_control" are each linked with a function block for incremental encoder processing. This provides a referencing option and prevents data overflow in the simulation.

Before the program can be operated, the simulations must be activated. If incremental encoder control mode was selected with "P_closed_loop_position_control", referencing must now be performed. Then, automatic control for axis 1 and the sequencer can be started. Axis 1 can also be operated in manual mode.

Application of function block "P_axis_simulation" in program "Pos_06"

```
PROGRAM Pos_06
VAR
        closed_loop_position_control_axis_01 : P_closed_loop_position_control ;
        basic_position_control_axis_02 : P_basic_position_control ;
        DINT_characteristics_control_axis_03 : P_DINT_characteristics_control ;
        axis_simulation_01 : P_axis_simulation ;
        axis_simulation_02 : P_axis_simulation ;
        axis_simulation_03 : P_axis_simulation ;
        Incremental_encoder_adaptation_02 : P_incremental_encoder_adaptation ;
        Incremental_encoder_adaptation_03 : P_incremental_encoder_adaptation ;
        Sq10_sequencer : P_Sq10_sequencer ;
        DI_0_0_BOOL : BOOL ;
        DI_0_1_BOOL : BOOL ;
        DI_0_2_BOOL : BOOL ;
        DI_0_3_BOOL : BOOL ;
        DI_0_4_BOOL : BOOL ;
        DI_0_5_BOOL : BOOL ;
        DI_0_6_BOOL : BOOL ;
        DI_0_7_BOOL : BOOL ;

        Setpoint_position_01_DINT : DINT ;
        Setpoint_position_02_DINT : DINT ;
        Setpoint_position_03_DINT : DINT ;
        Actual_position_01_DINT : DINT ;
        Actual_position_02_DINT : DINT ;
        Actual_position_03_DINT : DINT ;
        Reference_cam_signal_BOOL : BOOL ;
END_VAR

LD      closed_loop_position_control_axis_01.Actual_position_in_tolerance_zone_BOOL
AND     basic_position_control_axis_02.Actual_position_in_tolerance_zone_BOOL
AND     DINT_characteristics_control_axis_03.Actual_position_in_tolerance_zone_BOOL
ST      Sq10_sequencer.Actual_position_in_tolerance_zone_BOOL
```

```
CAL Sq10_sequencer(
        Positioning_job_finished_BOOL :=0,
        Activate_BOOL :=DI_0_0_BOOL,
        Number_of_steps_USINT :=10,
        Progression_condition_actual_position_in_tolerance_zone_BOOL :=1,
        External_switching_condition_step1_BOOL :=1,
        Selection_step1_axis1_BOOL :=1,
        Selection_step1_axis2_BOOL :=1,
        Selection_step1_axis3_BOOL :=1,
        Revolutions_step1_axis1_UINT :=50,
        Revolutions_step1_axis2_UINT :=50,
        Revolutions_step1_axis3_UINT :=50,
        Setpoint_position_step1_axis1_DINT :=0,
        Setpoint_position_step1_axis2_DINT :=0,
        Setpoint_position_step1_axis3_DINT :=0,
        External_value4_step1_INT :=3000,
        External_value5_step1_INT :=3000,
        External_value6_step1_INT :=3000,
        External_switching_condition_step2_BOOL :=1,
        Selection_step2_axis1_BOOL :=1,
        Selection_step2_axis2_BOOL :=1,
        Selection_step2_axis3_BOOL :=1,
        Revolutions_step2_axis1_UINT :=100,
        Revolutions_step2_axis2_UINT :=100,
        Revolutions_step2_axis3_UINT :=100,
        Setpoint_position_step2_axis1_DINT :=60000,
        Setpoint_position_step2_axis2_DINT :=60000,
        Setpoint_position_step2_axis3_DINT :=60000,
        External_value4_step2_INT :=2000,
        External_value5_step2_INT :=2000,
        External_value6_step2_INT :=2000,
        External_switching_condition_step3_BOOL :=1,
        Enable_step3_BOOL :=1,
```

```
        Selection_step3_axis1_BOOL :=1,
        Selection_step3_axis2_BOOL :=1,
        Selection_step3_axis3_BOOL :=1,
        Revolutions_step3_axis1_UINT :=100,
        Revolutions_step3_axis2_UINT :=100,
        Revolutions_step3_axis3_UINT :=100,
        Setpoint_position_step3_axis1_DINT :=20000,
        Setpoint_position_step3_axis2_DINT :=20000,
        Setpoint_position_step3_axis3_DINT :=20000,
        External_value4_step3_INT :=3000,
        External_value5_step3_INT :=3000,
        External_value6_step3_INT :=3000,
        External_switching_condition_step4_BOOL :=1,
        Selection_step4_axis1_BOOL :=1,
        Selection_step4_axis2_BOOL :=1,
        Selection_step4_axis3_BOOL :=1,
        Revolutions_step4_axis1_UINT :=40,
        Revolutions_step4_axis2_UINT :=40,
        Revolutions_step4_axis3_UINT :=40,
        Setpoint_position_step4_axis1_DINT :=5000,
        Setpoint_position_step4_axis2_DINT :=5000,
        Setpoint_position_step4_axis3_DINT :=5000,
        External_value4_step4_INT :=2500,
        External_value5_step4_INT :=2500,
        External_value6_step4_INT :=2500,
        External_switching_condition_step5_BOOL :=1,
        Selection_step5_axis1_BOOL :=1,
        Selection_step5_axis2_BOOL :=1,
        Selection_step5_axis3_BOOL :=1,
        Revolutions_step5_axis1_UINT :=100,
        Revolutions_step5_axis2_UINT :=100,
        Revolutions_step5_axis3_UINT :=100,
        Setpoint_position_step5_axis1_DINT :=60000,
        Setpoint_position_step5_axis2_DINT :=60000,
        Setpoint_position_step5_axis3_DINT :=60000,
```

```
        External_value4_step5_INT :=1000,
        External_value5_step5_INT :=1000,
        External_value6_step5_INT :=1000,
        External_switching_condition_step6_BOOL :=1,
        Selection_step6_axis1_BOOL :=1,
        Selection_step6_axis2_BOOL :=1,
        Selection_step6_axis3_BOOL :=1,
        Revolutions_step6_axis1_UINT :=98,
        Revolutions_step6_axis2_UINT :=98,
        Revolutions_step6_axis3_UINT :=98,
        Setpoint_position_step6_axis1_DINT :=0,
        Setpoint_position_step6_axis2_DINT :=0,
        Setpoint_position_step6_axis3_DINT :=0,
        External_value4_step6_INT :=1500,
        External_value5_step6_INT :=1500,
        External_value6_step6_INT :=1500,
        External_switching_condition_step7_BOOL :=1,
        Selection_step7_axis1_BOOL :=1,
        Selection_step7_axis2_BOOL :=1,
        Selection_step7_axis3_BOOL :=1,
        Revolutions_step7_axis1_UINT :=95,
        Revolutions_step7_axis2_UINT :=95,
        Revolutions_step7_axis3_UINT :=95,
        Setpoint_position_step7_axis1_DINT :=10000,
        Setpoint_position_step7_axis2_DINT :=10000,
        Setpoint_position_step7_axis3_DINT :=10000,
        External_value4_step7_INT :=3000,
        External_value5_step7_INT :=3000,
        External_value6_step7_INT :=3000,
        External_switching_condition_step8_BOOL :=1,
        Selection_step8_axis1_BOOL :=1,
        Selection_step8_axis2_BOOL :=0,
        Selection_step8_axis3_BOOL :=0,
```

```
        Revolutions_step8_axis1_UINT :=100,
        Revolutions_step8_axis2_UINT :=100,
        Revolutions_step8_axis3_UINT :=100,
        Setpoint_position_step8_axis1_DINT :=55000,
        Setpoint_position_step8_axis2_DINT :=55000,
        Setpoint_position_step8_axis3_DINT :=55000,
        External_value4_step8_INT :=1500,
        External_value5_step8_INT :=1500,
        External_value6_step8_INT :=1500,
        External_switching_condition_step9_BOOL :=1,
        Selection_step9_axis1_BOOL :=1,
        Selection_step9_axis2_BOOL :=1,
        Selection_step9_axis3_BOOL :=1,
        Revolutions_step9_axis1_UINT :=80,
        Revolutions_step9_axis2_UINT :=80,
        Revolutions_step9_axis3_UINT :=80,
        Setpoint_position_step9_axis1_DINT :=3200,
        Setpoint_position_step9_axis2_DINT :=3200,
        Setpoint_position_step9_axis3_DINT :=3200,
        External_value4_step9_INT :=1000,
        External_value5_step9_INT :=1000,
        External_value6_step9_INT :=1000,
        External_switching_condition_step10_BOOL :=1,
        Selection_step10_axis1_BOOL :=1,
        Selection_step10_axis2_BOOL :=1,
        Selection_step10_axis3_BOOL :=1,
        Revolutions_step10_axis1_UINT :=100,
        Revolutions_step10_axis2_UINT :=100,
        Revolutions_step10_axis3_UINT :=100,
        Setpoint_position_step10_axis1_DINT :=50000,
        Setpoint_position_step10_axis2_DINT :=50000,
        Setpoint_position_step10_axis3_DINT :=50000,
```

```
        External_value4_step10_INT :=1600,
        External_value5_step10_INT :=1600,
        External_value6_step10_INT :=1600
        )

(*=====================================================*)
(*====== setpoint positions ===========================*)
(*=====================================================*)
LD      Sq10_sequencer.Setpoint_position_axis1_DINT
ST      Setpoint_position_01_DINT

LD      Sq10_sequencer.Setpoint_position_axis2_DINT
ST      Setpoint_position_02_DINT

LD      Sq10_sequencer.Setpoint_position_axis3_DINT
ST      Setpoint_position_03_DINT

(*=====================================================*)
(*==== rotational speed and manipulated variables max ==*)
(*=====================================================*)
LD      Sq10_sequencer.Revolutions_axis1_UINT
ST      closed_loop_position_control_axis_01.Revolutions_negative_percent_UINT
ST      closed_loop_position_control_axis_01.Revolutions_positive_percent_UINT

LD      Sq10_sequencer.Revolutions_axis2_UINT
MUL     15
ST      basic_position_control_axis_02.Manipulated_variable_max_11Bit_UINT

LD      Sq10_sequencer.Revolutions_axis3_UINT
UINT_TO_INT
MUL     15
ST      DINT_characteristics_control_axis_03.Manipulated_variable_maximum_11Bit_INT
ST      DINT_characteristics_control_axis_03.Manipulated_variable_middle_11Bit_INT
DIV     5
ST      DINT_characteristics_control_axis_03.Manipulated_variable_minimum_11Bit_INT
```

```
(*=======================================================*)
(*====== running up and deceleration times ====================*)
(*=======================================================*)
LD      Sq10_sequencer.External_value4_INT
INT_TO_UINT
MUL     9
ST      closed_loop_position_control_axis_01.Acceleration_time_negative_ms_UINT
ST      closed_loop_position_control_axis_01.Acceleration_time_positive_ms_UINT
ST      closed_loop_position_control_axis_01.Deceleration_time_negative_ms_UINT
ST      closed_loop_position_control_axis_01.Deceleration_time_positive_ms_UINT

LD      Sq10_sequencer.External_value5_INT
INT_TO_UDINT
MUL     10
ST      basic_position_control_axis_02.Deceleration_position_deviation_UDINT
DIV     100
UDINT_TO_UINT
ST      basic_position_control_axis_02.Rounding_position_deviation_15bit_UINT

LD      Sq10_sequencer.External_value5_INT
INT_TO_UINT
DIV     2
ST      basic_position_control_axis_02.Ramp_time_100increase_maximum_ms_UINT

LD      Sq10_sequencer.External_value6_INT
MUL     10
ST      DINT_characteristics_control_axis_03.Position_deviation_maximum_INT
ST      DINT_characteristics_control_axis_03.Position_deviation_middle_INT
DIV     50
st      DINT_characteristics_control_axis_03.Position_deviation_minimum_INT
```

```
LD      Sq10_sequencer.External_value6_INT
INT_TO_UINT
DIV     2
ST      DINT_characteristics_control_axis_03.Ramp_time_100increase_maximum_ms_UINT

(*======================================================*)
(*====== axis 01 ========================================*)
(*======================================================*)

LD      axis_simulation_01.Incremental_encoder_ReferenceOutput_BOOL
ST      closed_loop_position_control_axis_01.Reference_output_signal_incremental_
        encoder_BOOL
LD      axis_simulation_01.Incremental_encoder_Output_UDINT
UDINT_TO_DINT
ST      closed_loop_position_control_axis_01.Incremental_encoder_value_DINT

LD      DI_0_7_BOOL
JMPCN   INCREMENT_SENSOR
        LD    0
        ST    closed_loop_position_control_axis_01.Referencing_positive_BOOL
        LD    DI_0_4_BOOL
        ST    closed_loop_position_control_axis_01.Reference_cam_signal_BOOL
LD      0
        ST    axis_simulation_01.Incremental_sensor_ReferenceInput_BOOL
JMP     E_INCREMENT_SENSOR
INCREMENT_SENSOR:
        LD    DI_0_4_BOOL
        ST    closed_loop_position_control_axis_01.Referencing_positive_BOOL
        LD    Reference_cam_signal_BOOL
        ST    closed_loop_position_control_axis_01.Reference_cam_signal_BOOL
LD      closed_loop_position_control_axis_01.Reference_input_signal_incremental_
        encoder_BOOL
        ST    axis_simulation_01.Incremental_sensor_ReferenceInput_BOOL
```

```
E_INCREMENT_SENSOR:
CAL closed_loop_position_control_axis_01(
        Setpoint_position_DINT :=Setpoint_position_01_DINT,
        Operating_mode_rotary_axis_BOOL :=0,
        Absolute_value_transmitter_BOOL :=DI_0_7_BOOL,
        Absolute_value_transmitter_without_referencing_BOOL :=0,
        Activate_BOOL :=DI_0_1_BOOL,
        Fault_signal_acknowledge_BOOL :=0,
        Manipulated_variable_negation_BOOL :=0,
        Accept_setpoint_position_BOOL :=DI_0_5_BOOL,
        Manual_control_negative_BOOL :=DI_0_2_BOOL,
        Manual_control_positive_BOOL :=DI_0_3_BOOL,
        Manual_control_oppresion_before_referencing_BOOL :=1,
        Referencing_negative_BOOL :=0,
        Position_control_gain_negative_percent_UINT :=60,
        Position_control_gain_positive_percent_UINT :=60,
        Position_control_degressive_gain_percent_UINT :=300,
        Speed_precontrol_negative_percent_UINT :=40,
        Speed_precontrol_positive_percent_UINT :=40,
        Nominal_revolutions_min_UINT :=2000,
        Tolerance_positioning_zone_UINT :=20,
        Machine_zero_point_DINT :=0,
        Software_limit_negative_DINT :=-2000000,
        Software_limit_positive_DINT :=2000000,
        Reference_point_acceleration_time_ms_UINT :=500,
        Reference_cam_searching_revolutions_percent_UINT :=50,
        Zero_mark_searching_revolutions_percent_UINT :=2,
        Contouring_error_maximum_UINT :=3000,
        Manual_revolutions_percent_UINT :=100,
        Manual_acceleration_time_ms_UINT :=2500,
        Manual_control_step_width_DINT :=0,
        Crawl_speed_revolutions_percent_UINT :=0,
        Crawl_speed_zone_DINT :=0,
```

```
        Hydraulic_zero_point_coverage_negative_UINT :=0,
        Hydraulic_zero_point_coverage_positive_UINT :=0,
        Hydraulic_stick_slip_compensation_negative_11Bit_UINT :=0,
        Hydraulic_stick_slip_compensation_positive_11Bit_UINT :=0,
        Hydraulic_stick_slip_zone_negative_UINT :=0,
        Hydraulic_stick_slip_zone_positive_UINT :=0,
        Maximum_incremental_encoder_DINT :=16777215,
        Encoder_increments_per_revolution_UINT :=100,
        Actual_position_DINT=>Actual_position_01_DINT
        )

LD      closed_loop_position_control_axis_01.Manipulated_variable_12Bit_INT
ST      axis_simulation_01.Manipulated_variable_12Bit_INT
CAL axis_simulation_01(
        Activate_BOOL :=DI_0_6_BOOL,
        Accept_manual_value_BOOL :=0,
        Nominal_revolutions_per_minute_INT :=2000,
        Manual_value_UDINT :=0,
        Increments_per_revolution_UINT :=100,
        Reference_cam_signal_BOOL=>Reference_cam_signal_BOOL
        )

(*=======================================================*)
(*====== axis 02 ========================================*)
(*=======================================================*)

LD      axis_simulation_02.Incremental_sensor_Output_UDINT
UDINT_TO_DINT
ST      Incremental_sensor_adaptation_02.Increments_DINT
CAL Incremental_sensor_adaptation_02(
        Machine_zero_point_DINT :=2000,
        Maximum_incremental_encoder_DINT :=16777215,
        Activate_BOOL :=1,
        Absolute_value_transmitter_BOOL :=1,
        Absolute_value_transmitter_without_referencing_BOOL :=0,
```

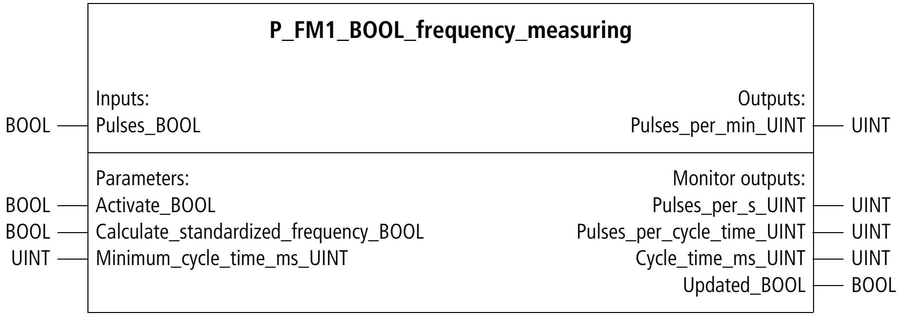
```
        Accept_machine_zero_point_BOOL :=DI_0_4_BOOL,
        Reference_signal_BOOL :=0,
        Actual_value_DINT=>Actual_position_02_DINT
        )

CAL basic_position_control_axis_02(
        Setpoint_position_DINT :=Setpoint_position_02_DINT,
        Actual_position_DINT :=Actual_position_02_DINT,
        Activate_BOOL :=DI_0_1_BOOL,
        Manipulated_variable_negation_BOOL :=0,
        Accept_setpoint_position_BOOL :=DI_0_5_BOOL,
        Cycle_time_demand_optimize_BOOL :=0,
        Tolerance_positioning_zone_UINT :=30
        )

LD      basic_position_control_axis_02.Manipulated_variable_12Bit_INT
ST      axis_simulation_02.Manipulated_variable_12Bit_INT
CAL axis_simulation_02(
        Activate_BOOL :=DI_0_6_BOOL,
        Accept_manual_value_BOOL :=0,
        Nominal_revolutions_per_minute_INT :=2000,
        Manual_value_UDINT :=0,
        Increments_per_revolution_UINT :=100
        )

(*=======================================================*)
(*====== axis 03 ========================================*)
(*=======================================================*)

LD      axis_simulation_03.Incremental_sensor_Output_UDINT
UDINT_TO_DINT
ST      Incremental_sensor_adaptation_03.Increments_DINT
```

```
CAL Incremental_sensor_adaptation_03(
        Machine_zero_point_DINT :=2000,
        Maximum_incremental_encoder_DINT :=16777215,
        Activate_BOOL :=1,
        Absolute_value_transmitter_BOOL :=1,
        Absolute_value_transmitter_without_referencing_BOOL :=0,
        Accept_machine_zero_point_BOOL :=DI_0_4_BOOL,
        Reference_signal_BOOL :=0,
        Actual_value_DINT=>Actual_position_03_DINT
        )

CAL DINT_characteristics_control_axis_03(
        Setpoint_position_DINT :=Setpoint_position_03_DINT,
        Actual_position_DINT :=Actual_position_03_DINT,
        Activate_BOOL :=DI_0_1_BOOL,
        Manipulated_variable_negation_BOOL :=0,
        Position_deviation_off_INT :=0,
        Tolerance_positioning_zone_UINT :=30
        )

LD      DINT_characteristics_control_axis_03.Manipulated_variable_12Bit_INT
ST      axis_simulation_03.Manipulated_variable_12Bit_INT
CAL axis_simulation_03(
        Activate_BOOL :=DI_0_6_BOOL,
        Accept_manual_value_BOOL :=0,
        Nominal_revolutions_per_minute_INT :=2000,
        Manual_value_UDINT :=0,
        Increments_per_revolution_UINT :=100
        )

END_PROGRAM
```

6 Frequency measuring

Measuring the frequency of a pulse sequence

P_FM1_BOOL_frequency_measuring
Measuring the frequency of a pulse sequence (speed measuring)



Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Pulses_BOOL	Pulses, e.g. from digital input	0/1
Parameters		
Activate_BOOL	Activates the function block	0/1
Calculate_standardized_frequency_ BOOL	The frequencies are calculated per second and per minute	0/1
Minimum_cycle_time_ms_UINT	The pulses are counted at least across this timespan	0 to 30000
Outputs		
Pulses_per_min_UINT	Pulses per minute	0 to 65535

Designation	Function	Value range
Monitor outputs		
Pulses_per_s_UINT	Pulses per second	0 to 65535
Pulses_per_cycle_time_UINT	Pulses per cycle time	0 to 65535
Cycle_time_ms_UINT	Actual cycle time	0 to 65535
Updated_BOOL	Update pulse indication	0/1

Description

With this function block, frequencies of pulses received at "Pulses_BOOL" can be calculated. The lowest possible frequency is 0.02 Hz, and the highest frequency must not be greater than half of the PLC cycle time.

Example:
PLC cycle time = 5 ms
=> PLC scan cycles = 1000 ms/5 ms = 200
=> Maximum frequency = 100 Hz

The calculation starts when the function block is activated. Specify a minimum cycle time, after which the pulse indication is to be updated. The actual cycle time can be a little longer, since the frequency calculation does not start until the next pulse is received after the minimum pulse time has expired. A standardized frequency calculation in minutes or seconds can be selected, but this increases the cycle time demand of the function block. The frequency output is updated when the cycle time has expired and is specified by output variable "Updated_BOOL".

Example:
In this application example, a minimum cycle time of 800 ms has been chosen. Within the actual cycle time of 802 ms, 111 pulses were registered. Because standardized frequencies are to be calculated, 8304 pulses per minute and 138 pulses per second are output.

Application of function block "P_FM1_BOOL_frequency_measuring" in program "frequency"

```
PROGRAM frequency
VAR
      Digital_input_0_BOOL : BOOL ;
      Frequency_measuring_digital_input_0 : P_FM1_BOOL_frequency_measuring ;
END_VAR

CAL Frequency_measuring_digital_input_0(
      Pulses_BOOL :=Digital_input_0_BOOL,
      Activate_BOOL :=1,
      Calculate_standardized_frequency_BOOL :=1,
      Minimum_cycle_time_ms_UINT :=800,
      Pulses_per_min_UINT=>8304,
      Pulses_per_s_UINT=>138,
      Pulses_per_cycle_time_UINT=>111,
      Cycle_time_ms_UINT=>802
      )
END_PROGRAM
```

P_FM3_BOOL_frequency_measuring
Measuring the frequency of a pulse sequence, superimposed three times

	P_FM3_BOOL_frequency_measuring		
	Inputs:	Outputs:	
BOOL	Pulses_BOOL	Pulses_per_min_UINT	UINT
	Parameters:	Monitor outputs:	
BOOL	Activate_BOOL	Pulses_per_s_UINT	UINT
BOOL	Calculate_standardized_frequency_BOOL	Pulses_per_cycle_time_UINT	UINT
UINT	Minimum_cycle_time_ms_UINT	Cycle_time_ms_UINT	UINT
		Updated_BOOL	BOOL

Function block prototype

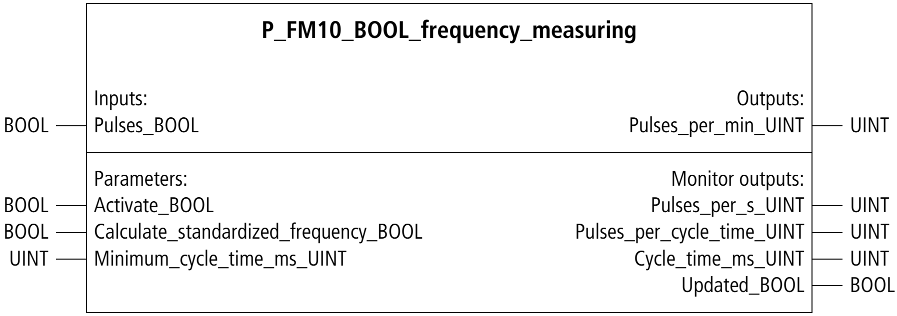
Meaning of the operands

Designation	Function	Value range
Inputs		
Pulses_BOOL	Pulses, e.g. from digital input	0/1
Parameters		
Activate_BOOL	Activates the function block	0/1
Calculate_standardized_frequency_BOOL	The frequencies are calculated per second and per minute	0/1
Minimum_cycle_time_ms_UINT	The pulses are counted at least across this timespan	0 to 30000
Outputs		
Pulses_per_min_UINT	Pulses per minute	0 to 65535
Monitor outputs		
Pulses_per_s_UINT	Pulses per second	0 to 65535
Pulses_per_cycle_time_UINT	Pulses per cycle time	0 to 65535
Cycle_time_ms_UINT	Actual cycle time	0 to 65535
Updated_BOOL	Update pulse indication	0/1

Description

See function block “P_FM1_BOOL_frequency_measuring”. The only difference to function block “P_FM1_BOOL_frequency_measuring”, frequency measuring with this function block is superimposed three times. The pulse indication is therefore updated three times faster, but the cycle time demand also increases.

P_FM10_BOOL_frequency_measuring Measuring the frequency of a pulse sequence, superimposed ten times



Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Pulses_BOOL	Pulses, e.g. from digital input	0/1
Parameters		
Activate_BOOL	Activates the function block	0/1
Calculate_standardized_frequency_BOOL	The frequencies are calculated per second and per minute	0/1
Minimum_cycle_time_ms_UINT	The pulses are counted at least across this timespan	0 to 30000
Outputs		
Pulses_per_min_UINT	Pulses per minute	0 to 65535
Monitor outputs		
Pulses_per_s_UINT	Pulses per second	0 to 65535
Pulses_per_cycle_time_UINT	Pulses per cycle time	0 to 65535
Cycle_time_ms_UINT	Actual cycle time	0 to 65535
Updated_BOOL	Update pulse indication	0/1

Description

See function block “P_FM1_BOOL_frequency_measuring”. The only difference to function block “P_FM1_BOOL_frequency_measuring”, frequency measuring with this function block is superimposed ten times. The pulse indication is therefore updated ten times faster, but the cycle time demand also increases.

Measuring the frequency of an incremental encoder value

P_FM1_UINT_frequency_measuring Measuring the frequency of a 16-bit incremental encoder value

P_FM1_UINT_frequency_measuring

Type	Inputs:	Outputs:	Type
UINT	Increments_16Bit_UINT	Pulses_per_s_INT	INT
	Parameters:	**Monitor outputs:**	
BOOL	Activate_BOOL	Pulses_per_cycle_time_INT	INT
BOOL	Calculate_standardized_frequency_BOOL	Cycle_time_ms_UINT	UINT
UINT	Minimum_cycle_time_ms_UINT	Updated_BOOL	BOOL

Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Increments_16Bit_UINT	Increments	0 to 65535
Parameters		
Activate_BOOL	Activates the function block	0/1
Calculate_standardized_frequency_BOOL	The frequencies are calculated per second and per minute	0/1
Minimum_cycle_time_ms_UINT	The pulses are counted at least across this timespan	0 to 30000
Outputs		
Pulses_per_s_INT	Pulses per second	–32768 to 32767
Monitor outputs		
Pulses_per_cycle_time_INT	Pulses per cycle time	–32768 to 32767
Cycle_time_ms_UINT	Actual cycle time	0 to 65535
Updated_BOOL	Update pulse indication	0/1

Description

With this function block, you can calculate frequencies of signals received at input "Increments_16Bit_UINT". The input can be linked with fast counter or incremental encoder signal. This function block is especially suitable for measuring higher frequencies (> 50 Hz). At lower frequencies, function block "P_FM1_BOOL_ frequency_measuring" yields more accurate results.

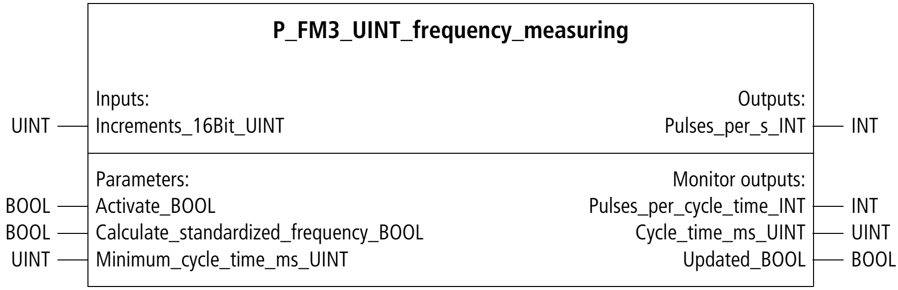
The calculation starts when the function block is activated. Specify a minimum cycle time, after which the pulse indication is to be updated. The actual cycle time can be a little greater, but only by up to one PLC cycle time. A standardized frequency calculation in seconds can be selected, but this increases the cycle time demand of the function block. The frequency output is updated when the cycle time has expired and is indicated by output variable "Updated_BOOL".

Example:
In this application example, a minimum cycle time of 200 ms has been chosen. Within the actual cycle time of 201 ms, 230 pulses were registered. Because standardized frequencies are to be calculated, 1 144 pulses per minute and pulses per second are output.

Application of function block ”P_FM1_UINT_frequency_measuring” in program “frequency”

```
PROGRAM frequency
VAR
      Fast_counter_UINT : UINT;
      Frequency_measuring_fast_counter : P_FM1_UINT_frequency_measuring ;
END_VAR

CAL Frequency_measuring_fast_counter(
      Increments_16Bit_UINT :=Fast_counter_UINT,
      Activate_BOOL :=1,
      Calculate_standardized_frequency_BOOL :=1,
      Minimum_cycle_time_ms_UINT :=200,
      Pulses_per_s_INT=>1144,
      Pulses_per_cycle_time_INT=>230,
      Cycle_time_ms_UINT=>201
      )
END_PROGRAM
```

P_FM3_UINT_frequency_measuring Measuring the frequency of a 16-bit incremental encoder value, superimposed three times



Function block prototype

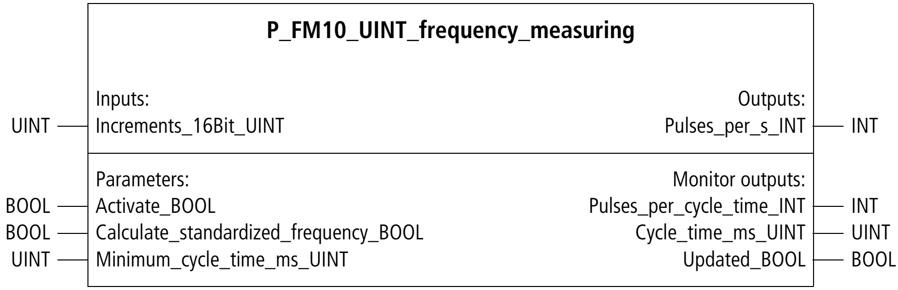
Meaning of the operands

Designation	Function	Value range
Inputs		
Increments_16Bit_UINT	Increments	0 to 65535
Parameters		
Activate_BOOL	Activates the function block	0/1
Calculate_standardized_ frequency_BOOL	The frequencies are calculated per second and per minute	0/1
Minimum_cycle_time_ms_UINT	The pulses are counted at least across this timespan	0 to 30000
Outputs		
Pulses_per_s_INT	Pulses per second	–32768 to 32767
Monitor outputs		
Pulses_per_cycle_time_INT	Pulses per cycle time	–32768 to 32767
Cycle_time_ms_UINT	Actual cycle time	0 to 65535
Updated_BOOL	Update pulse indication	0/1

Description

See function block "P_FM1_UINT_frequency_measuring". The only difference to function block "P_FM1_UINT_frequency_measuring", frequency measuring with this function block is superimposed three times. The pulse indication is therefore updated three times faster, but the cycle time demand also increases.

P_FM10_UINT_frequency_measuring Measuring the frequency of a 16-bit incremental encoder value, superimposed ten times



Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Increments_16Bit_UINT	Increments	0 to 65535
Parameters		
Activate_BOOL	Activates the function block	0/1
Calculate_standardized_frequency_BOOL	The frequencies are calculated per second and per minute	0/1
Minimum_cycle_time_ms_UINT	The pulses are counted at least across this timespan	0 to 30000
Outputs		
Pulses_per_s_INT	Pulses per second	–32768 to 32767
Monitor outputs		
Pulses_per_cycle_time_INT	Pulses per cycle time	–32768 to 32767
Cycle_time_ms_UINT	Actual cycle time	0 to 65535
Updated_BOOL	Update pulse indication	0/1

Description

See function block "P_FM1_UINT_frequency_measuring". The only difference to function block "P_FM1_UINT_frequency_measuring", frequency measuring with this function block is superimposed ten times. The pulse indication is therefore updated ten times faster, but the cycle time demand also increases.

7 Synchronisation (electronic gearing)

Synchronisation controller

P_SYCI_syncronisation_controller Synchronisation controller with incremental encoder input from master and slave axis

P_SYCI_syncronisation_controller

Type	Inputs:	Outputs:	Type
BOOL	Synchronisation_signal_master_BOOL	Manipulated_variable_slave_12Bit_INT	INT
BOOL	Synchronisation_signal_slave_BOOL		
UDINT	Increments_master_UDINT		
UDINT	Increments_slave_UDINT		

Type	Parameters:	Monitor outputs:	Type
BOOL	Activate_BOOL	Manipulated_variable_slave_P_13Bit_INT	INT
BOOL	Angle_synchronisation_BOOL	Manipulated_variable_slave_I_13Bit_INT	INT
BOOL	Transformation_ratio_optimize_BOOL	Actual_position_master_DINT	DINT
BOOL	Stop_BOOL	Actual_position_slave_DINT	DINT
BOOL	Change_direction_BOOL	Deviation_master_slave_DINT	DINT
BOOL	Manipulated_variable_negation_BOOL	Transformation_ratio_optimized_ 10thpercent_UINT	UINT
BOOL	Accept_manual_manipulated_ variable_BOOL	Transformation_ratio_optimized_BOOL	BOOL
INT	Manual_manipulated_variable_12Bit_INT	Stopped_BOOL	BOOL
UINT	PI_controller_proportional_rate_ percent_UINT	Disturbance_maximum_manipulated_ variable_attained_BOOL	BOOL
UINT	PI_controller_reset_time_100ths_UINT	Disturbance_transformation_ratio_BOOL	BOOL
UINT	Transformation_ratio_10thpercent_UINT	Disturbance_BOOL	BOOL
UINT	Acceleration_ramp_ms_UINT		
UINT	Deceleration_ramp_ms_UINT		
INT	Asynchronisation_difference_INT		
UDINT	Maximum_incremental_master_encoder_UDINT		
UDINT	Maximum_incremental_slave_encoder_UDINT		
INT	Lower_limit_manipulated_variable_INT		
INT	Upper_limit_manipulated_variable_INT		

Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Synchronisation_signal_master_ BOOL	Angular synchronisation signal for the master axis	0/1
Synchronisation_signal_ slave_BOOL	Angular synchronisation signal for the slave axis	0/1
Increments_master_UDINT	Increments (from an encoder) of the master	-10^9 to 10^9
Increments_slave_UDINT	Increments (from an encoder) of the slave	-10^9 to 10^9
Parameters		
Activate_BOOL	Activates the function block	0/1
Angle_synchronisation_BOOL	Angular synchronisation	0/1
Transformation_ratio_optimize_ BOOL	Optimization of the transformation ratio	0/1
Stop_BOOL	Stop the slave using a deceleration ramp	0/1
Change_direction_BOOL	Direction reversal of the slave	0/1
Manipulated_variable_negation_ BOOL	Negation of the manipulated variable	0/1
Accept_manual_manipulated_ variable_BOOL	Accept the manual manipulated variable	0/1
Manual_manipulated_variable_ 12Bit_INT	Manual manipulated variable	–2048 to 2047
PI_controller_proportional_rate_ percent_UINT	Proportional gain of the PI controller in percent	0 to 6000
PI_controller_reset_time_100ths_ UINT	Reset time of the PI controller in 0.01 s	0 to 65535
Transformation_ratio_ 10thpercent_UINT	Transformation ratio between master and slave in 10ths of a percent	0 to 32000
Acceleration_ramp_ms_UINT	Acceleration ramp for engaging the coupling in ms	0 to 65535

Designation	Function	Value range
Deceleration_ramp_ms_UINT	Deceleration ramp for disengaging the coupling in ms	0 to 65535
Asynchronisation_difference_INT	Specified difference between master and slave (up to one rotation)	–32768 to 32767
Maximum_increments_master_encoder_UDINT	Maximum number of master encoder increments (before overflow occurs)	0 to 10^9
Maximum_increments_slave_encoder_UDINT	Maximum number of slave encoder increments (before overflow occurs)	0 to 10^9
Lower_limit_manipulated_variable_INT	Lower limit of manipulated variable range (default setting: –2048)	–4095 to 0
Upper_limit_manipulated_variable_INT	Upper limit of manipulated variable range (default setting: 2047)	0 to 4095
Outputs		
Manipulated_variable_slave_12Bit_INT	Manipulated variable of the slave axis (normally with a 12-bit resolution, e.g. –2048 to 2047)	–4095 to 4095
Monitor outputs		
Manipulated_variable_slave_P_13Bit_INT	Proportional component of the manipulated variable	–4095 to 4095
Manipulated_variable_slave_I_13Bit_INT	Integral component of the manipulated variable	–4095 to 4095
Actual_position_master_DINT	Actual position of the master axis	-10^9 to 10^9
Actual_position_slave_DINT	Actual position of the slave axis	-10^9 to 10^9
Deviation_master_slave_DINT	Controller deviation between master and slave axis	-10^9 to 10^9
Transformation_ratio_optimized_10thpercent_UINT	Optimized transformation ratio in thousandths	0 to 32000
Transformation_ratio_optimized_BOOL	Status: the transformation ratio is optimized	0/1
Stopped_BOOL	Status: the slave axis is stopped (slave manip. var. = 0)	0/1

Designation	Function	Value range
Error_max_manipulated_variable_reached_BOOL	Error: the slave's maximum manipulated variable was reached for several revolutions and no synchronisation has taken place	0/1
Error_transformation_ratio_BOOL	Error: the optimized transformation ratio is at least five times greater or smaller than the specified transformation ratio	0/1
Fault_BOOL	Error: An error exists and the slave axis has been stopped (manipulated variable = 0)	0/1

Description

With this function block, a slave axis can be synchronized with a master axis. A transformation ratio between the master and slave axes must be specified (electronic gearing). To synchronize the speeds, the position values from an incremental encoder must be supplied by the master and slave axes. The encoder's respective maximum value (before a data overflow occurs) must also be specified.

Example:
The increments from an incremental encoder are registered. If a 24-bit output is used, the maximum value must be set to "16777215"; if a 16-bit output is used, this value must be "65535".

If an angle synchronisation is to be carried out the digital synchronisation signals from the master and slave axes are needed at each revolution. To allow reliable registration of all signals, the synchronisation signals must be applied at least twice as long as the PLC cycle time.

After activation of the function block, the synchronisation process begins. Angular synchronisation takes place when both synchronisation signals have been applied once. If "Optimize_transformation_ratio_BOOL" is set to "1", the specified transformation ratio is corrected. This optimized value can be changed at input "Transformation_ratio_ 10thpercent_UINT", so that the next synchronisation starts with the correct transformation ratio. To avoid value jumps of the manipulated variable when synchronisation begins (during coupling), an acceleration ramp can be defined. Likewise, by entering a value other than "0" in "Deceleration_ramp_ms_UINT", disengagement without jumps in the manipulated variable can be achieved. If "Stop_BOOL" is changed from "0" to "1", the manipulated variable is decelerated according to this ramp. Monitor output "Stopped_BOOL" indicates that the deceleration process is finished.

If "Change_direction_BOOL" is set to "1", the direction of rotation of the slave axis is reversed.

If "Manipulated_variable_negation_BOOL" is set to "1", the manipulated variable is negated. The manipulated variable is overridden by a manual manipulated variable if "Accept_manual_manip_var_BOOL" is set to "1".

The difference (controller deviation) between master and slave (→ Monitor output) is the input variable of a PI controller which can be parameterized with proportional gain and the reset time. The effect of the proportional gain on the PI controller's integrator is reproportional, i.e. the smaller the gain, the faster does integration take place. If the controller is set too smooth (proportional gain too low), the slave axis does not follow the master axis efficiently when the speed or direction changes. If the controller is set too harsh (proportional gain too high or reset time too short), the slave axis oscillates. If the controller responds quickly to changes of the master, but a long time expires until the axes are fully synchronized again, the reset time may have been set too high.

With parameter "Asynchronisation_difference_INT", a difference (of up to one full rotation) between the synchronisation signals from master and slave can be specified. The manipulated variable limits are normally selected as follows:

- The positive and negative direction of the axis is –2048 to 2047
- Positive direction of rotation only is 0 to 4095 (or 0 to 2047)

Example:
In the application example, an "electronic gearing" with a transformation ratio of 1 to 3 has been selected, i.e. the slave axis rotates three times faster than the master axis. Engagement and disengagement take place with a ramped deceleration of five seconds.

Application of function block "P_SYCI_syncronisation_controller" in program "Synchro"

```
PROGRAM Synchro
VAR
      SYCI_synchronisation_controller : P_SYCI_syncronisation_controller ;
      DI_Master_BOOL : BOOL ;
      DI_Slave_BOOL : BOOL ;
      AO_0_0_2_0 : INT ;
      Incremental_encoder_master : UDINT ;
      Incremental_encoder_slave : UDINT ;
END_VAR
```

```
CAL SYCI_synchronisation_controller(
      Synchronisation_signal_master_BOOL :=DI_master_BOOL,
      Synchronisation_signal_slave_BOOL :=DI_slave_BOOL,
      Increments_master_UDINT :=Incremental_encoder_master,
      Increments_slave_UDINT :=Incremental_encoder_slave,
      Activate_BOOL :=1,
      Angle_synchronisation_BOOL :=1,
      Optimize_transformation_ratio_BOOL :=1,
      Stop_BOOL :=0,
      Change_direction_BOOL :=0,
      Manipulated_variable_negation_BOOL :=0,
      Accept_manual_manip_var_BOOL :=0,
      Manual_manipulated_variable_12Bit_INT :=0,
      PI_controller_proportional_rate_percent_UINT :=150,
      PI_controller_reset_time_100ths_UINT :=100,
      Transformation_ratio_10thpercent_UINT :=3000,
      Acceleration_ramp_ms_UINT :=5000,
      Deceleration_ramp_ms_UINT :=5000,
      Asynchronisation_difference_INT :=0,
      Maximum_increments_master_encoder_UDINT :=16777215,
      Maximum_increments_slave_encoder_UDINT :=16777215,
      Lower_limit_manipulated_variable_INT :=-2048,
      Upper_limit_manipulated_variable_INT :=2047
      )

LD    SYCI_synchronisation_controller.Manipulated_variable_slave_12Bit_INT
ST    AO_0_0_2_0

END_PROGRAM
```

P_SYC_syncronisation_controller Synchronisation controller with master axis speed input

	P_SYC_syncronisation_controller		
	Inputs:	Outputs:	
BOOL	Master_toggle_BOOL	Setpoint_rotational_speed_slave_UINT	UINT
BOOL	Slave_toggle_BOOL		
UINT	Time_difference_master_master_ms_UINT		
UINT	Time_difference_slave_slave_ms_UINT		
UINT	Time_difference_master_slave_ms_UINT		
UINT	Time_difference_slave_master_ms_UINT		
UINT	Actual_rotational_speed_master_UINT		
	Parameters:	Monitor outputs:	
BOOL	Activate_BOOL	Setpoint_speed_slave_transformation_ ratio_UINT	UINT
UINT	Maximum_rotational_speed_slave_UINT		
UINT	Transformation_ratio_10thpercent_UINT	Setpoint_speed_slave_synchronisation_INT	INT
BOOL	Transformation_ratio_optimize_BOOL	Transformation_ratio_optimized_ 10thpercent_UINT	UINT
UINT	Transformation_ratio_gain_percent_UINT		
BOOL	Synchronisation_ratio_indicate_BOOL	Synchronisation_ratio_10thpercent_UINT	UINT

Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Master_toggle_BOOL	Toggle bit of the master axis	0/1
Slave_toggle_BOOL	Toggle bit of the slave axis	0/1
Time_difference_master_master_ ms_UINT	Time difference between two synchronisation signals of the master axis	0 to 65535
Time_difference_slave_slave_ms_ UINT	Time difference between two synchronisation signals of the slave axis	0 to 65535
Time_difference_master_slave_ ms_UINT	Time difference between the synchronisation signals from the master axis to the slave axis	0 to 65535

Designation	Function	Value range
Time_difference_slave_master_ ms_UINT	Time difference between the synchronisation signals from the slave axis to the master axis	0 to 65535
Actual_rotational_speed_master_ UINT	Actual rotational speed of the master axis	0 to 32000
Parameters		
Activate_BOOL	Activates the function block	0/1
Maximum_rotational_speed_ slave_UINT	Maximum rotational speed of the slave axis	0 to 65535
Transformation_ratio_10thpercent_ UINT	transformation ratio in thousandths	0 to 65535
Transformation_ratio_optimize_ BOOL	Optimize the transformation ratio	0/1
Transformation_ratio_gain_percent_ UINT	Transformation ratio gain for optimization	0 to 16000
Synchronisation_ratio_indicate_ BOOL	Show the synchronisation ratio	0/1
Outputs		
Setpoint_rotational_speed_slave_ UINT	Setpoint speed of the slave axis	0 to 32000
Monitor outputs		
Setpoint_speed_slave_ transformation_ratio_UINT	Setpoint speed of the slave axis resulting from the transformation ratio	0 to 16000
Setpoint_speed_slave_ synchronisation_INT	Setpoint speed of the slave axis resulting from syncronization	–16000 to 16000
Transformation_ratio_optimized_ 10thpercent_UINT	Optimized transformation ratio	0 to 65535
Synchronisation_ratio_10thpercent_ UINT	Synchronisation ratio in thousandths	0 to 65535

→ If one incremental encoder each is connected to the master and slave axis, use function block “P_SYCI_syncronisation_controller”.

Description

This function block is used for synchronizing the rotational speed of a “slave axis” in relation to a “master axis”. A transformation ratio between master and slave axes can be specified (electronic gearing). The Master and slave axes must issue a synchronisation signal once per revolution. These signals can be processed by function block “P_SYTP_syncronisation_time”, whose outputs must be connected to the inputs of this function block. The setpoint speed of the slave axis should be specified as precisely as possible with “Actual_rotational_speed_master_UINT” and “Transformation_ratio_10thpercent_UINT”. If the synchronisation signals from the master and slave axes are offset in time, the slave axis is automatically accelerated or decelerated to synchronize these signals.

Example of a transformation ratio:
Actual speed of master axis = 1200
Transformation ratio = 3000 per 1000
=>
Setpoint speed of slave axis (resulting from transformation ratio) = 3600

When the function block is activated, the slave axis is initially regulated to this speed, as defined by the transformation ratio and the speed of the master axis. If “Transformation_ratio_optimize_BOOL” is set to “1”, the function block automatically corrects the specified transformation ratio according to the set optimization factor. The corrected value, which is output by “Transformation_ratio_optimized_10thpercent_UINT” must be applied to the corresponding input, so that the correct transformation ratio can be used for the next activation of the function block.

Although input “Actual_rotational_speed_master_UINT” must normally be linked to the actual speed of the master, the master axis’ setpoint speed or manipulated variable can also be applied to it if the transformation ratio between these variables and the synchronized slave axis speed is constant.

If “Show_synchronisation_ratio_BOOL” is set to “1”, “Synchronisation_ratio_10thpercent_UINT” indicates the extent of the synchronisation between master and slave axis. The value “1000” corresponds to full synchronisation.

“Setpoint_rotational_speed_slave_UINT” provides the output and the components “transformation ratio” and “synchronisation”, which are added together to arrive at the setpoint speed of the slave axis. For the slave axis, enter a maximum speed, to which “Setpoint_rotational_speed_slave_UINT” is limited.

To maintain a constant, controlled difference between the synchronisation signals from the master and slave axis, use the input “Asynchronisation_time_ms_INT” of function block “P_SYTP_syncronisation_time”. To ensure the accuracy of this function block, and therefore of the whole synchronisation process, keep the cycle time as short as possible. If necessary, optimize the user program (by segmenting and alternately calling the program sequences), so that the cycle time of the program sequence together with function block “P_SYTP_syncronisation_time” is as short as possible.

Example:
In the application example, a slave axis is synchronized with a master axis with a transformation ratio of 8.5. The synchronisation signals are read by the digital inputs and the actual speed of the master axis with the analog input. The setpoint speed of the slave axis is issued at the analog output. The maximum setpoint speed of the slave axis is limited to “4095”.

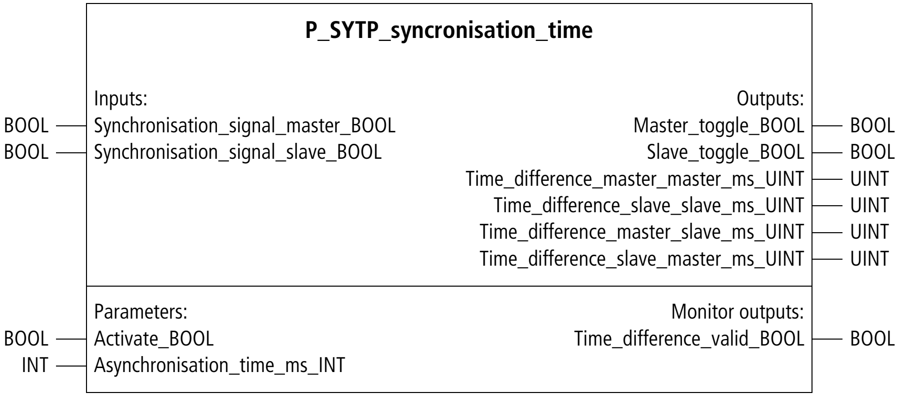
Application of function block"P_SYC_syncronisation_controller" in program "Synchro"

```
PROGRAM Synchro
VAR
      SYC_synchronisation_controller : P_SYR_syncronisation_controller ;
      SYTP_syncronisation_time : P_SYTP_syncronisation_time ;
      DI_0_0_BOOL : BOOL ;
      DI_0_1_BOOL : BOOL ;
      AI_0_0_2_0_UINT : UINT ;
      AO_0_0_2_0_UINT : UINT ;
END_VAR

CAL SYTP_syncronisation_time(
      Synchronisation_signal_master_BOOL :=DI_0_0_BOOL,
      Synchronisation_signal_slave_BOOL :=DI_0_1_BOOL,
      Activate_BOOL :=1,
      Asynchronisation_time_ms_INT :=0)

CAL SYC_synchronisation_controller(
      Master_toggle_BOOL :=SYTP_syncronisation_time.Master_toggle_BOOL,
      Slave_toggle_BOOL :=SYTP_syncronisation_time.Slave_toggle_BOOL,
      Time_difference_master_master_ms_UINT
      :=SYTP_syncronisation_time.Time_difference_master_master_ms_UINT,
      Time_difference_slave_slave_ms_UINT
      :=SYTP_synchronisation_time.Time_difference_slave_slave_ms_UINT,
      Time_difference_master_slave_ms_UINT
      :=SYTP_synchronisation_time.Time_difference_master_slave_ms_UINT,
      Time_difference_slave_master_ms_UINT
      :=SYTP_synchronisation_time.Time_difference_slave_master_ms_UINT,
```

```
      Actual_rotational_speed_master_UINT :=AI_0_0_2_0_UINT,
      Activate_BOOL :=1,
      Maximum_rotational_speed_slave_UINT :=4095,
      Transformation_ratio_10thpercent_UINT :=8500,
      Optimize_transformation_ratio_BOOL :=1,
      Transformation_ratio_gain_percent_UINT :=100,
      Synchronisation_ratio_indicate_BOOL :=1,
      Setpoint_rotational_speed_slave_UINT=>AO_0_0_2_0_UINT
      )
END_PROGRAM
```

Synchronisation time

P_SYTP_synchronisation_time
Time differences between angle synchronisation signal from master and slave axes



Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Synchronisation_signal_master_ BOOL	Synchronisation signal from the master axis	0/1
Synchronisation_signal_slave_ BOOL	Synchronisation signal from the slave axis	0/1

Designation	Function	Value range
Parameters		
Activate_BOOL	Activates the function block	0/1
Asynchronisation_time_ms_INT	Time difference between the synchronisation signals from the master and slave axes	–32768 to 32767
Outputs		
Master_toggle_BOOL	Toggle bit of the master axis	0/1
Slave_toggle_BOOL	Toggle bit of the slave axis	0/1
Time_difference_master_master_ms_UINT	Time difference between two synchronisation signals of the master axis	0 to 65535
Time_difference_slave_slave_ms_UINT	Time difference between two synchronisation signals of the slave axis	0 to 65535
Time_difference_master_slave_ms_UINT	Time difference between the synchronisation signals from the master axis to the slave axis	0 to 65535
Time_difference_slave_master_ms_UINT	Time difference between the synchronisation signals from the slave axis to the master axis	0 to 65535
Monitor outputs		
Time_difference_valid_BOOL	Current time differences	0/1

Description

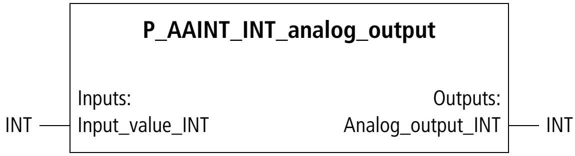
This function block is used in combination with function block “P_SYC_syncronisation_controller” to synchronize a slave axis with a master axis. To ensure the accuracy of this function block, and therefore of the whole synchronisation process, keep the cycle time as short as possible. If necessary, optimize the user program (by segmenting and alternately calling the program sequences), so that the cycle time of the program sequence together with function block “P_SYTP_syncronisation_time” is as short as possible. When this function block is activated, the times between the synchronisation signals from a master and a slave axis are measured. When, after an initial phase, all time differences have been measured, “Time_differences_current_BOOL” issues a “1”. The outputs of this function block must be connected to the corresponding inputs of “P_SYC_syncronisation_controller”.

For an application example for “P_SYC_syncronisation_controller”, → page 152.

8 Other Function Blocks

Converting manipulated variables for analog output

P_AAINT_INT_analog_output Converting manipulated variables for analog output



Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Input_value_INT	Bipolar input value: Negative range = –2048 to –1 Zero = 0 Positive range = 1 to 2047	–2048 to 2047
Outputs		
Analog_output_INT	Unipolar (positive) output value for analog output: Negative range = 2048 to 4095 Zero = 0 Positive range = 1 to 2047	0 to 4095

Description

The position controllers output bipolar (positive and negative) manipulated variables. For their signal output hardware they may require unipolar manipulated variables. This function block transforms the negative range “–2048 to –1” to “2048 to 4095”. The positive range remains unchanged in the range “0 to 2047”.

Application of function block “P_AAINT_INT_analog_output” in program “Output”

```
PROGRAM Output
VAR
      AAINT_INT_analog_output : P_AAINT_INT_analog_output ;
      Manipulated_variable_bipolar_12Bit_INT : INT ;
      Manipulated_variable_unipolar_12Bit_INT : INT ;
      AO_0_0_2_0 : INT ;
END_VAR
CAL AAINT_INT_analog_output(
      Input_value_INT :=Manipulated_variable_bipolar_12Bit_INT,
      Analog_output_INT=>Manipulated_variable_unipolar_12Bit_INT
      )
LD    Manipulated_variable_unipolar_12Bit_INT
ST    AO_0_0_2_0

END_PROGRAM
```

Incremental encoder processing

P_incremental_encoder_evaluation Incremental encoder processing

	P_incremental_encoder_evaluation		
	Inputs:	Outputs:	
DINT	Increments_DINT	Actual_value_DINT	DINT
	Parameters:	Monitor outputs:	
DINT	Machine_zero_point_DINT	Reset_incremental_encoder_ evaluation_BOOL	BOOL
DINT	Maximum_incremental_encoder_DINT		
BOOL	Activate_BOOL	Offset_machine_zero_point_DINT	DINT
BOOL	Absolute_value_transmitter_BOOL		
BOOL	Absolute_value_transmitter_without_referencing_BOOL		
BOOL	Accept_machine_zero_point_BOOL		
BOOL	Reference_signal_BOOL		

Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Increments_DINT	Increments from an incremental or absolute encoder	-10^9 to 10^9
Parameters		
Machine_zero_point_DINT	The mathematical zero point can be offset by the machine zero point.	-10^9 to 10^9
Maximum_incremental_encoder_ DINT	Maximum value of encoder increments, e.g. 65535 for 16 bit maximum or 16777215 for 24 bit maximum	-10^9 to 10^9
Activate_BOOL	Activates the function block	0/1
Absolute_value_transmitter_BOOL	Mode: Absolute encoder	0/1
Absolute_value_transmitter_ without_referencing_BOOL	Mode: Absolute encoder without referencing	0/1
Accept_machine_zero_point_BOOL	Accept machine zero point	0/1
Reference_signal_BOOL	Referencing signal	0/1

Designation	Function	Value range
Outputs		
Actual_value_DINT	Actual value with referencing and protection against data overflow	-10^9 to 10^9
Monitor outputs		
Reset_incremental_encoder_ evaluation_BOOL	Status: The incremental encoder processing was reset by a referencing signal.	0/1
Offset_machine_zero_point_DINT	Offset caused by acceptance of a machine zero point.	-10^9 to 10^9

Description

This function block is used for processing incremental encoder output. A zero point can be specified with referencing. The function block registers any data overflow of the incremental encoder and produces the steady, bipolar (positive and negative) output value "Actual_value_DINT".

This function block must be activated. On deactivation, it is reset. In absolute encoder mode, the current incremental encoder output is defined as the zero point at a rising edge of "Accept_machine_zero_point_BOOL". This zero point can be offset by entering the value of "Machine_zero_ point_DINT". The maximum incremental encoder value must be entered for identification of a data overflow. If parameter "Absolute_value_transmitter_without_ referencing_BOOL" has been set to "1", referencing cannot be triggered and a data over-range is not intercepted.

If parameter “Referencing signal_BOOL” carries a “1”, the current incremental encoder output is regarded as the referenced value (hardware referencing was performed). The machine zero point is added to this incremental encoder value.

Monitor output “Offset_machine_zero_point_DINT” indicates the value by which output variable “Actual_ value_DINT” and input variable “Increments_DINT” was offset when the machine zero point was accepted.

Example:
In the application example, an incremental encoder value is recorded and then processed with “P_incremental_ encoder_evaluation”. With digital input “0”, the setpoint position can be changed. The zero point (referencing) is defined on a rising edge of digital input “1”. The basic positioning function block is activated with digital input “2”.

Application of function block "P_incremental_encoder_evaluation" in program "Pos_07"

```
PROGRAM Pos_07
VAR
        Incremental_encoder_evaluation_01 : P_incremental_encoder_evaluation ;
        basic_position_control_01 : P_basic_position_control ;
        AAINT_INT_analog_output : P_AAINT_INT_analog_output ;
        DI_0_0_BOOL : BOOL ;
        DI_0_1_BOOL : BOOL ;
        DI_0_2_BOOL : BOOL ;
        AO_0_0_2_0 : INT ;
        Setpoint_position_DINT : DINT ;
        Actual_position_DINT : DINT ;
        Incremental_encoder_01 : UDINT ;
END_VAR

LD      Incremental_encoder_01
UDINT_TO_DINT
ST      Incremental_encoder_evaluation_01.Increments_DINT
CAL Incremental_encoder_evaluation_01(
        Machine_zero_point_DINT :=2000,
        Maximum_incremental_encoder_DINT :=16777215,
        Activate_BOOL :=1,
        Absolute_value_transmitter_BOOL :=1,
        Absolute_value_transmitter_without_referencing_BOOL :=0,
        Accept_machine_zero_point_BOOL :=DI_0_1_BOOL,
        Reference_signal_BOOL :=0,
        Actual_value_DINT=>Actual_position_DINT
        )
```

```
LD      DI_0_0_BOOL
JMPCN   SETPOINT_POSITION_02
        LD    5000
        ST    Setpoint_position_DINT
        JMP   E_SETPOINT_POSITION_02
SETPOINT_POSITION_02:
        LD    200000
        ST    Setpoint_position_DINT
E_SETPOINT_POSITION_02:

CAL basic_position_control_01(
        Setpoint_position_DINT :=Setpoint_position_DINT,
        Actual_position_DINT :=Actual_position_DINT,
        Activate_BOOL :=DI_0_2_BOOL,
        Manipulated_variable_negation_BOOL :=0,
        Accept_setpoint_position_BOOL :=1,
        Cycle_time_demand_optimize_BOOL :=0,
        Ramp_time_100increase_maximum_ms_UINT :=50,
        Deceleration_position_deviation_UDINT :=4500,
        Rounding_position_deviation_15bit_UINT :=450,
        Manipulated_variable_max_11Bit_UINT :=1500,
        Tolerance_positioning_zone_UINT :=20
        )

CAL AAINT_INT_analog_output(
        Input_value_INT :=basic_position_control_01.Manipulated_variable_12Bit_INT,
        Analog_output_INT=>AO_0_0_2_0
)
END_PROGRAM
```

Interpolation

At a linear interpolation, the application of a straight line between two interpolation points results in an interpolation profile. Outside the interpolation limits (X1 and X3 in this case), an extrapolation can be carried out (→ fig. 14 upper graph) or the interpolation limits can be applied (→ fig. 14 lower graph). For the interpolation point (X1 | Y1) and (X2 | Y2), the following functional association applies for interpolation (between the interpolation points) and extrapolation (outside the interpolation points):

$$F(x) = Y = \frac{(X - X_1) \times (Y1 - Y_2)}{X_1 - X_2} + Y_1$$

The graphs show any function profile, which is approximated by using an interpolation with three X/Y interpolation points. The deviation between the function profile and the approximated interpolation profile depends on the number and position of the interpolation points and on the curvature of the function profile.

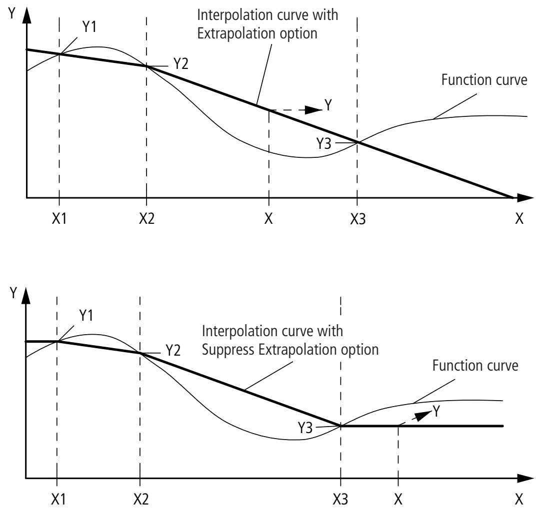


Figure 14: Interpolation profile

If the curve is extrapolated, the mathematical result can lie outside the double integer value range. The function blocks of the closed-loop control toolbox then use the value range limit as the result.

Typical interpolation example

A characteristic curve is defined by ten interpolation points. A linear profile is assumed between the interpolation points. An interpolation between the interpolation points results in the function profile (→ fig. 15). See also the sample application program for function block "P_IP10_DINT_interpolation" on page 178.

X	600	650	700	800	1000	1200	1500	1900	2500	2800
Y	0	500	1000	1500	2000	2500	3000	3500	4000	4095

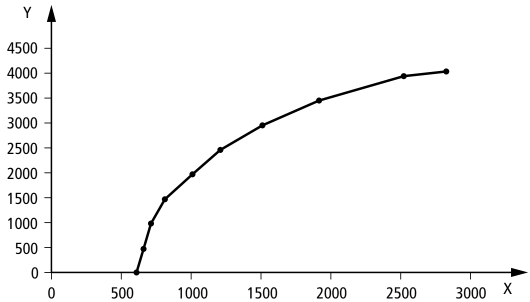


Figure 15: Function profile through interpolation

P_IP2_DINT_interpolation
Interpolation with two X/Y interpolation points and double integer values

	P_IP2_DINT_interpolation		
	Inputs:	Outputs:	
DINT —	X_DINT	Y_DINT	— DINT
	Parameters:		
BOOL —	Suppress_extrapolation_BOOL		
DINT —	X1_DINT		
DINT —	X2_DINT		
DINT —	Y1_DINT		
DINT —	Y2_DINT		

Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
X_DINT	Known X value	–2 147 483 648 to 2 147 483 647
Parameters		
Suppress_extrapolation_BOOL	For X-values outside the interpolation limits, the following can be selected: 0 => Extrapolate 1 => Extrapolation is suppressed. The interpolation boundaries are output	0/1

Designation	Function	Value range
X1_DINT	X value 1	–2 147 483 648 to 2 147 483 647
X2_DINT	X value 2	–2 147 483 648 to 2 147 483 647
Y1_DINT	Y value 1	–2 147 483 648 to 2 147 483 647
Y2_DINT	Y value 2	–2 147 483 648 to 2 147 483 647
Outputs		
Y_DINT	Interpolated (or extrapolated) Y-value	–2 147 483 648 to 2 147 483 647

Description

Between the X/Y interpolation points, a linear interpolated value is calculated with reference to the X-value applied to the input. Outside the X/Y interpolation points, a linearly extrapolated Y value is calculated if "Suppress_extrapolation_BOOL" is 0. If "Suppress_extrapolation_BOOL" is 1, the Y interpolation limits are output instead (→ fig. 14, page 164).

Example:
For the input parameters shown in the sample program below, the input value "Analog_value_4_to_ 20_mA :=1500" results in the output value "Analog_value_12bit_DINT :=852".

Application of function block "P_IP2_DINT_interpolation" in program "A_4_20mA"

```
PROGRAM A_4_20mA
VAR
        Scaling_12bit : P_IP2_DINT_interpolation ;
        Analog_value_4_to_20_mA_WORD : WORD ;
        Analog_value_12bit_DINT : DINT ;
END_VAR
LD      Analog_value_4_to_20_mA_WORD
WORD_TO_DINT
ST      Scaling_12bit.X_DINT
CAL Scaling_12bit(
        Suppress_extrapolation_BOOL :=1,
        X1_DINT :=819,
        X2_DINT :=4095,
        Y1_DINT :=0,
        Y2_DINT :=4095,
        Y_DINT=>Analog_value_12bit_DINT
        )
END_PROGRAM
```

P_IP3_DINT_interpolation
Interpolation with 3 X/Y interpolation points and integer values

	P_IP3_DINT_interpolation		
	Inputs:	Outputs:	
DINT	X_DINT	Y_DINT	DINT
	Parameters:		
BOOL	Suppress_extrapolation_BOOL		
DINT	X1_DINT		
DINT	X2_DINT		
DINT	X3_DINT		
DINT	Y1_DINT		
DINT	Y2_DINT		
DINT	Y3_DINT		

Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
X_DINT	Known X value	–2 147 483 648 to 2 147 483 647
Parameters		
Suppress_extrapolation_BOOL	For X-values outside the interpolation limits, the following can be selected: 0 => Extrapolate 1 => Extrapolation is suppressed. The interpolation boundaries are output	0/1

Designation	Function	Value range
X1_DINT	X value 1	–2147483648 to 2147483647
X2_DINT	X value 2	–2147483648 to 2147483647
X3_DINT	X value 3	–2147483648 to 2147483647
Y1_DINT	Y value 1	–2147483648 to 2147483647
Y2_DINT	Y value 2	–2147483648 to 2147483647
Y3_DINT	Y value 3	–2147483648 to 2147483647
Outputs		
Y_DINT	Interpolated (or extrapolated) Y value	–2147483648 to 2147483647

Description

Between the X/Y interpolation points, a linear interpolated value is calculated with reference to the X-value applied to the input. Outside the X/Y interpolation points, a linearly extrapolated Y value is calculated if "Suppress_extrapolation_BOOL" is 0. If "Suppress_extrapolation_BOOL" is 1, the Y interpolation limits are output instead (→ fig. 14, page 164).

→ The X values must be entered in ascending order.

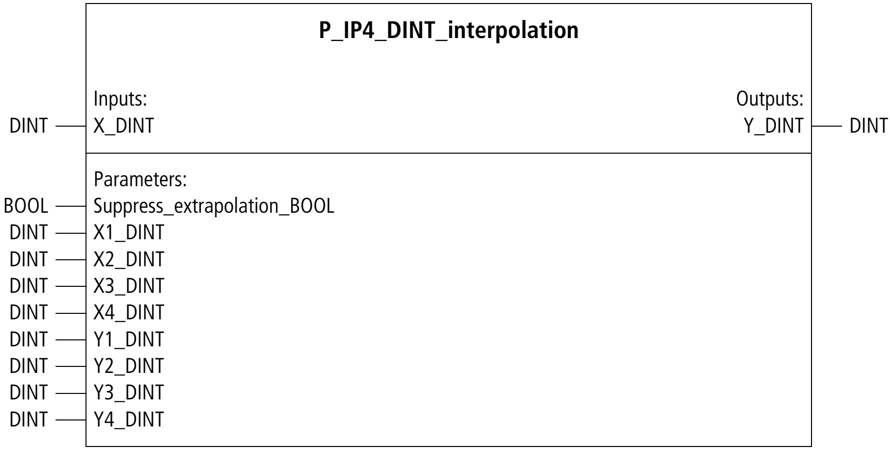
Example:
The following characteristic curve was generated with the application program below.

	1	2	3
X	0	800	4095
Y	–500	–1000	–2000

Application of function block “P_IP3_DINT_interpolation” in program “Charac_3”

```
PROGRAM Charac_3
VAR
      Characteristic_3Points : P_IP3_DINT_INTERPOLATION ;
      Analog_value_WORD : WORD ;
      Characteristic_function_value_DINT : DINT ;
END_VAR

LD    Analog_value_WORD
WORD_TO_DINT
ST    Characteristic_3Points.X_DINT
CAL Characteristic_3Points(
      Suppress_extrapolation_BOOL :=1,
      X1_DINT :=0,
      X2_DINT :=800,
      X3_DINT :=4095,
      Y1_DINT :=-500,
      Y2_DINT :=-1000,
      Y3_DINT :=-2000,
      Y_DINT=>Characteristic_function_value_DINT
)
END_PROGRAM
```

P_IP4_DINT_interpolation Interpolation with 4 X/Y-interpolation points and integer values



Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
X_DINT	Known X value	–2 147 483 648 to 2 147 483 647
Parameters		
Suppress_extrapolation_BOOL	For X-values outside the interpolation limits, the following can be selected: 0 => Extrapolate 1 => Extrapolation is suppressed. The interpolation boundaries are output	0/1

Designation	Function	Value range
X1_DINT	X value 1	–2 147 483 648 to 2 147 483 647
X2_DINT	X value 2	–2 147 483 648 to 2 147 483 647
X3_DINT	X value 3	–2 147 483 648 to 2 147 483 647
X4_DINT	X value 4	–2 147 483 648 to 2 147 483 647
Y1_DINT	Y value 1	–2 147 483 648 to 2 147 483 647
Y2_DINT	Y value 2	–2 147 483 648 to 2 147 483 647
Y3_DINT	Y value 3	–2 147 483 648 to 2 147 483 647
Y4_DINT	Y value 4	–2 147 483 648 to 2 147 483 647
Outputs		
Y_DINT	Interpolated (or extrapolated) Y value	–2 147 483 648 to 2 147 483 647

Description

Between the X/Y interpolation points, a linear interpolated value is calculated with reference to the X-value applied to the input. Outside the X/Y interpolation points, a linearly extrapolated Y value is calculated if "Suppress_extrapolation_BOOL" is 0. If "Suppress_extrapolation_BOOL" is 1, the Y interpolation limits are output instead (→ fig. 14, page 164).

→ The X values must be entered in ascending order.

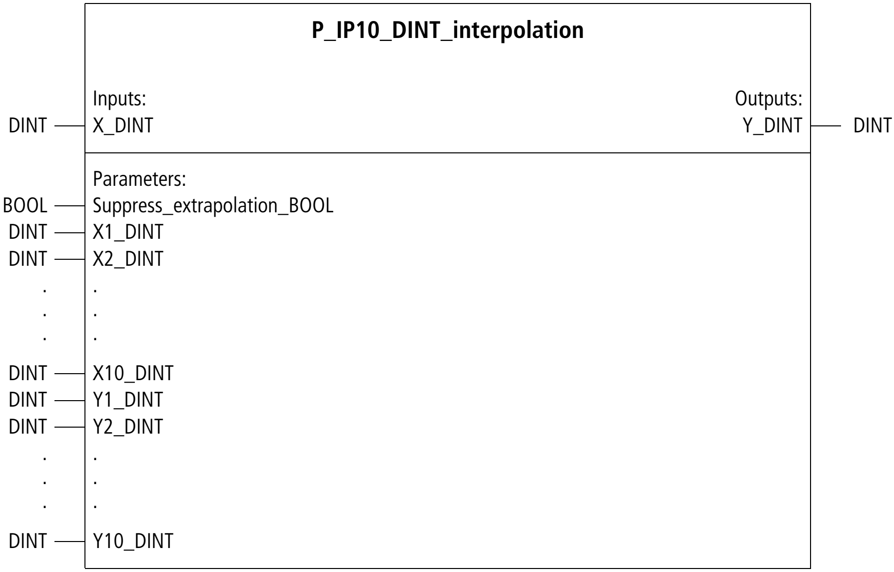
Example:
The following characteristic curve was generated with the application program below.

	1	2	3	4
X	0	800	2500	4095
Y	–500	–1000	1000	2000

Application of function block "P_IP4_DINT_interpolation" in program "Charac_4"

```
PROGRAM Charac_4
VAR
      Characteristic_4Points : P_IP4_DINT_INTERPOLATION ;
      Analog_value_WORD : WORD ;
      Characteristic_function_value_DINT : DINT ;
END_VAR

LD    Analog_value_WORD
WORD_TO_DINT
ST    Characteristic_4Points.X_DINT
CAL Characteristic_4Points(
      X1_DINT :=0,
      X2_DINT :=800,
      X3_DINT :=2500,
      X4_DINT :=4095,
      Y1_DINT :=-500,
      Y2_DINT :=-1000,
      Y3_DINT :=1000,
      Y4_DINT :=2000,
      Y_DINT=>Characteristic_function_value_DINT)
END_PROGRAM
```

P_IP10_DINT_interpolation
Interpolation with 10 X/Y-interpolation points and integer values



Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
X_DINT	Known X value	–2147483648 to 2147483647
Parameters		
Suppress_extrapolation_BOOL	For X-values outside the interpolation limits, the following can be selected: 0 => Extrapolate 1 => Extrapolation is suppressed. The interpolation boundaries are output	0/1
X1_DINT	X value 1	–2147483648 to 2147483647
X2_DINT	X value 2	–2147483648 to 2147483647
X3_DINT	X value 3	–2147483648 to 2147483647
X4_DINT	X value 4	–2147483648 to 2147483647
X5_DINT	X value 5	–2147483648 to 2147483647
X6_DINT	X value 6	–2147483648 to 2147483647
X7_DINT	X value 7	–2147483648 to 2147483647
X8_DINT	X value 8	–2147483648 to 2147483647
X9_DINT	X value 9	–2147483648 to 2147483647
X10_DINT	X value 10	–2147483648 to 2147483647

Designation	Function	Value range
Y1_DINT	Y value 1	–2 147 483 648 to 2 147 483 647
Y2_DINT	Y value 2	–2 147 483 648 to 2 147 483 647
Y3_DINT	Y value 3	–2 147 483 648 to 2 147 483 647
Y4_DINT	Y value 4	–2 147 483 648 to 2 147 483 647
Y5_DINT	Y value 5	–2 147 483 648 to 2 147 483 647
Y6_DINT	Y value 6	–2 147 483 648 to 2 147 483 647
Y7_DINT	Y value 7	–2 147 483 648 to 2 147 483 647
Y8_DINT	Y value 8	–2 147 483 648 to 2 147 483 647
Y9_DINT	Y value 9	–2 147 483 648 to 2 147 483 647
Y10_DINT	Y value 10	–2 147 483 648 to 2 147 483 647
Outputs		
Y_value_DINT	Interpolated (or extrapolated) Y value	–2 147 483 648 to 2 147 483 647

Description

Between the X/Y interpolation points, a linear interpolated value is calculated with reference to the X-value applied to the input. Outside the X/Y interpolation points, a linearly extrapolated Y value is calculated if "Suppress_extrapolation_BOOL" is 0. If "Suppress_extrapolation_BOOL" is 1, the Y interpolation limits are output instead (→ fig. 14, page 164).

→ The X values must be entered in ascending order.

Example:
The following characteristic curve was generated with the application program below.

	1	2	3	4	5	6	7	8	9	10
X	0	100	200	300	400	500	600	700	800	900
Y	–50	–100	–200	–400	–800	–1600	–3200	–6400	–12800	–25600

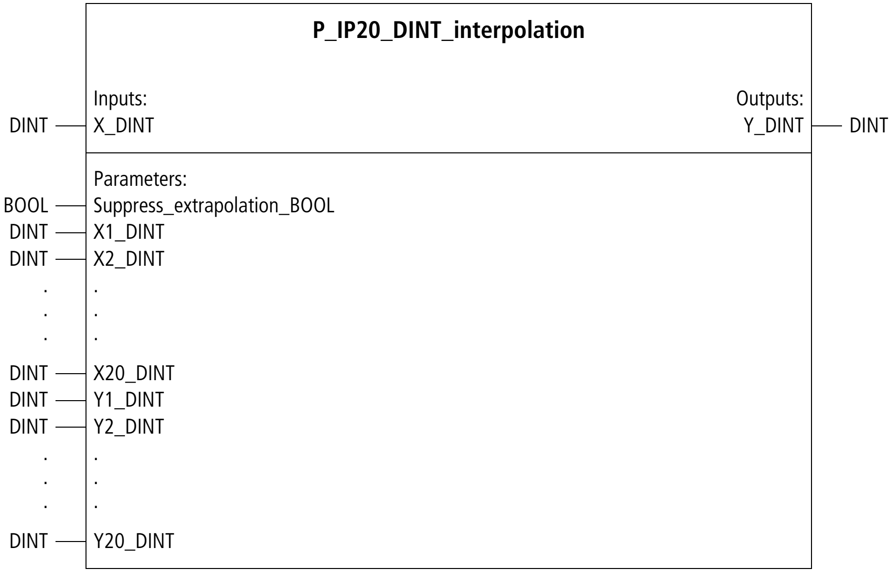
→ See also application example of "P_semicircle" (page 192) and "P_semi_ellipsis" (page 189).

Application of function block "P_IP10_DINT_interpolation" in program "Charac_10"

```
PROGRAM Charac_10
VAR
      Characteristic_10Points : P_IP10_DINT_INTERPOLATION ;
      Analog_value_WORD : WORD ;
      Characteristic_function_value_DINT : DINT ;
END_VAR

LD    Analog_value_WORD
WORD_TO_DINT
ST    Characteristic_10Points.X_DINT
```

```
CAL Characteristic_10Points(
      Suppress_extrapolation_BOOL :=1,
      X1_DINT :=0,
      X2_DINT :=100,
      X3_DINT :=200,
      X4_DINT :=300,
      X5_DINT :=400,
      X6_DINT :=500,
      X7_DINT :=600,
      X8_DINT :=700,
      X9_DINT :=800,
      X10_DINT :=900,
      Y1_DINT :=-50,
      Y2_DINT :=-100,
      Y3_DINT :=-200,
      Y4_DINT :=-400,
      Y5_DINT :=-800,
      Y6_DINT :=-1600,
      Y7_DINT :=-3200,
      Y8_DINT :=-6400,
      Y9_DINT :=-12800,
      Y10_DINT :=-25600,
      Y_DINT=>Characteristic_function_value_DINT
)
END_PROGRAM
```

P_IP20_DINT_interpolation Interpolation with 20 X/Y-interpolation points and integer values



Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
X_DINT	Known X value	–2 147 483 648 to 2 147 483 647
Parameters		
Suppress_extrapolation_BOOL	For X-values outside the interpolation limits, the following can be selected: 0 => Extrapolate 1 => Extrapolation is suppressed. The interpolation boundaries are output	0/1
X1_DINT	X value 1	–2 147 483 648 to 2 147 483 647
X2_DINT	X value 2	–2 147 483 648 to 2 147 483 647
X3_DINT	X value 3	–2 147 483 648 to 2 147 483 647
X4_DINT	X value 4	–2 147 483 648 to 2 147 483 647
X5_DINT	X value 5	–2 147 483 648 to 2 147 483 647
X6_DINT	X value 6	–2 147 483 648 to 2 147 483 647
X7_DINT	X value 7	–2 147 483 648 to 2 147 483 647
X8_DINT	X value 8	–2 147 483 648 to 2 147 483 647
X9_DINT	X value 9	–2 147 483 648 to 2 147 483 647
X10_DINT	X value 10	–2 147 483 648 to 2 147 483 647
X11_DINT	X value 11	–2 147 483 648 to 2 147 483 647

Designation	Function	Value range
X12_DINT	X value 12	–2 147 483 648 to 2 147 483 647
X13_DINT	X value 13	–2 147 483 648 to 2 147 483 647
X14_DINT	X value 14	–2 147 483 648 to 2 147 483 647
X15_DINT	X value 15	–2 147 483 648 to 2 147 483 647
X16_DINT	X value 16	–2 147 483 648 to 2 147 483 647
X17_DINT	X value 17	–2 147 483 648 to 2 147 483 647
X18_DINT	X value 18	–2 147 483 648 to 2 147 483 647
X19_DINT	X value 19	–2 147 483 648 to 2 147 483 647
X20_DINT	X value 20	–2 147 483 648 to 2 147 483 647
Y1_DINT	Y value 1	–2 147 483 648 to 2 147 483 647
Y2_DINT	Y value 2	–2 147 483 648 to 2 147 483 647
Y3_DINT	Y value 3	–2 147 483 648 to 2 147 483 647
Y4_DINT	Y value 4	–2 147 483 648 to 2 147 483 647
Y5_DINT	Y value 5	–2 147 483 648 to 2 147 483 647
Y6_DINT	Y value 6	–2 147 483 648 to 2 147 483 647
Y7_DINT	Y value 7	–2 147 483 648 to 2 147 483 647
Y8_DINT	Y value 8	–2 147 483 648 to 2 147 483 647

Designation	Function	Value range
Y9_DINT	Y value 9	–2 147 483 648 to 2 147 483 647
Y10_DINT	Y value 10	–2 147 483 648 to 2 147 483 647
Y11_DINT	Y value 11	–2 147 483 648 to 2 147 483 647
Y12_DINT	Y value 12	–2 147 483 648 to 2 147 483 647
Y13_DINT	Y value 13	–2 147 483 648 to 2 147 483 647
Y14_DINT	Y value 14	–2 147 483 648 to 2 147 483 647
Y15_DINT	Y value 15	–2 147 483 648 to 2 147 483 647
Y16_DINT	Y value 16	–2 147 483 648 to 2 147 483 647
Y17_DINT	Y value 17	–2 147 483 648 to 2 147 483 647
Y18_DINT	Y value 18	–2 147 483 648 to 2 147 483 647
Y19_DINT	Y value 19	–2 147 483 648 to 2 147 483 647
Y20_DINT	Y value 20	–2 147 483 648 to 2 147 483 647
Outputs		
Y_DINT	Interpolated (or extrapolated) Y value	–2 147 483 648 to 2 147 483 647

Description

Between the X/Y interpolation points, a linear interpolated value is calculated with reference to the X-value applied to the input. Outside the X/Y interpolation points, a linearly extrapolated Y value is calculated if "Suppress_extrapolation_BOOL" is 0. If "Suppress_extrapolation_BOOL" is 1, the Y interpolation limits are output instead (→ fig. 14, page 164).

→ The X values must be entered in ascending order.

Example:
The following characteristic curve was generated with the application program below. The result is an approximation of a cosine function for a range of 114 to 228° (deg).

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
X	114	120	126	132	138	144	150	156	162	168
Y	914	866	809	743	669	588	500	407	309	208

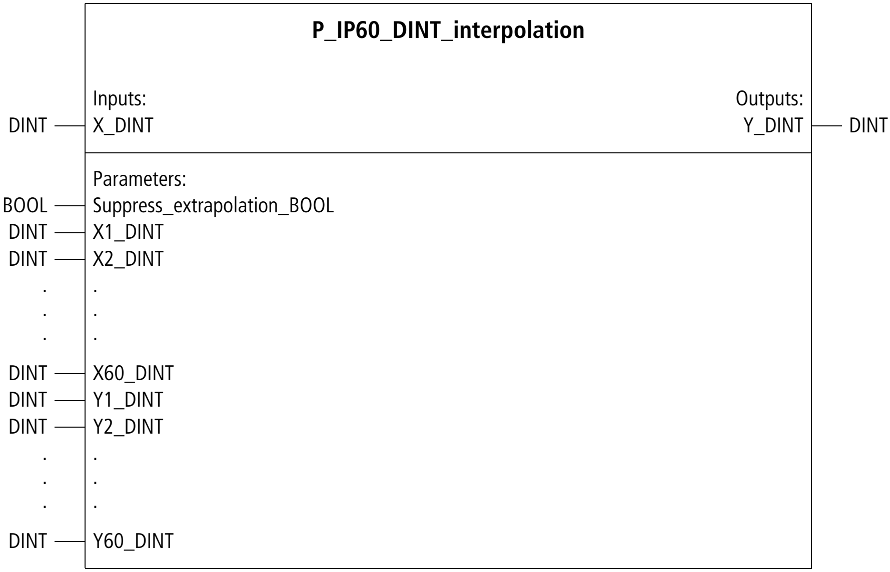
	11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
X	174	180	186	192	198	204	210	216	222	228
Y	105	0	–105	–208	–309	–407	–500	–588	–669	–743

Application of function block “P_IP20_DINT_interpolation” in program “Cos_228”

```
PROGRAM Cos_228
VAR
      Cosine_calculation_114_to_228_deg :P_IP20_DINT_INTERPOLATION ;
      Deg_114_to_228_DINT : DINT :=145;
      Cosine_per1000_DINT : DINT ;
END_VAR

CAL Cosine_calculation_114_to_228_deg(
      X_DINT :=Deg_114_to_228_DINT,
      Suppress_extrapolation_BOOL :=1,
      X1_DINT :=114,
      X2_DINT :=120,
      X3_DINT :=126,
      X4_DINT :=132,
      X5_DINT :=138,
      X6_DINT :=144,
      X7_DINT :=150,
      X8_DINT :=156,
      X9_DINT :=162,
      X10_DINT :=168,
      X11_DINT :=174,
      X12_DINT :=180,
      X13_DINT :=186,
      X14_DINT :=192,
      X15_DINT :=198,
      X16_DINT :=204,
      X17_DINT :=210,
      X18_DINT :=216,
      X19_DINT :=222,
      X20_DINT :=228,
```

```
      Y1_DINT :=914,
      Y2_DINT :=866,
      Y3_DINT :=809,
      Y4_DINT :=743,
      Y5_DINT :=669,
      Y6_DINT :=588,
      Y7_DINT :=500,
      Y8_DINT :=407,
      Y9_DINT :=309,
      Y10_DINT :=208,
      Y11_DINT :=105,
      Y12_DINT :=0,
      Y13_DINT :=-105,
      Y14_DINT :=-208,
      Y15_DINT :=-309,
      Y16_DINT :=-407,
      Y17_DINT :=-500,
      Y18_DINT :=-588,
      Y19_DINT :=-669,
      Y20_DINT :=-743,
      Y_DINT=>Cosine_per1000_DINT)
END_PROGRAM
```

P_IP60_DINT_interpolation
Interpolation with 60 X/Y-interpolation points and integer values

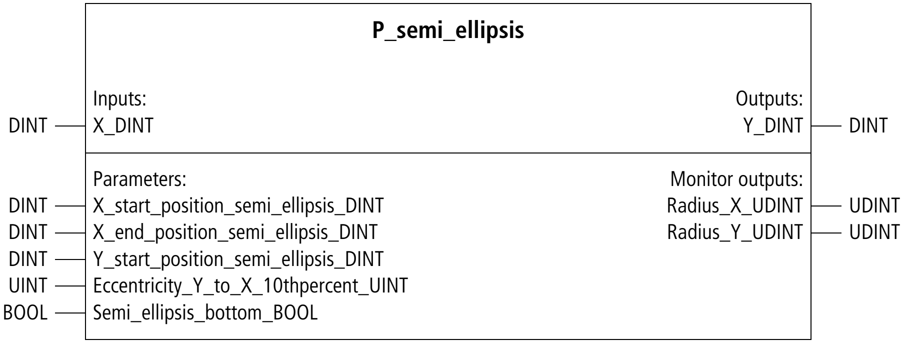


Function block prototype

→ Except for the number of X/Y interpolation points, this function block is identical with function block "P_IP20_DINT_Interpolation". For a description of this function block, → from page 184.

Geometry

P_semi_ellipsis
Semi-ellipsis



Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
X_DINT	X value of ellipsis	-10^9 to 10^9
Parameters		
X_start_position_semi_ellipsis_DINT	Starting X value of semi-ellipsis	-10^9 to 10^9
X_end_position_semi_ellipsis_DINT	End X value of semi-ellipsis	-10^9 to 10^9
Y_start_position_semi_ellipsis_DINT	Starting Y value of semi-ellipsis	-10^9 to 10^9
Eccentricity_Y_to_X_10thpercent_UINT	Eccentricity of radii Y to X of ellipsis	0 to 65535
Semi_ellipsis_bottom_BOOL	Selection whether the semi-ellipsis runs at top (0) or bottom (1)	0/1
Outputs		
Y_DINT	Y value of the semi-ellipsis	-10^9 to 10^9
Monitor outputs		
Radius_X_UDINT	Radius of the ellipsis in X direction	0 to 2×10^9
Radius_Y_UDINT	Radius of the ellipsis in Y direction	0 to 2×10^9

Description

This function block generates an elliptic curve of the X/Y values between the definable X/Y start and end values. This functionality can be used, for example, for a master-slave positioning process. The user can specify the eccentricity of the ellipsis in thousandths (1 000 = circle). With parameter "Semi_ellipsis_bottom_BOOL", the user can select whether the semi-ellipsis will be above (larger than the Y starting value) or below (smaller than the Y starting value). The monitor outputs are the radii of the ellipsis in the X and Y direction.

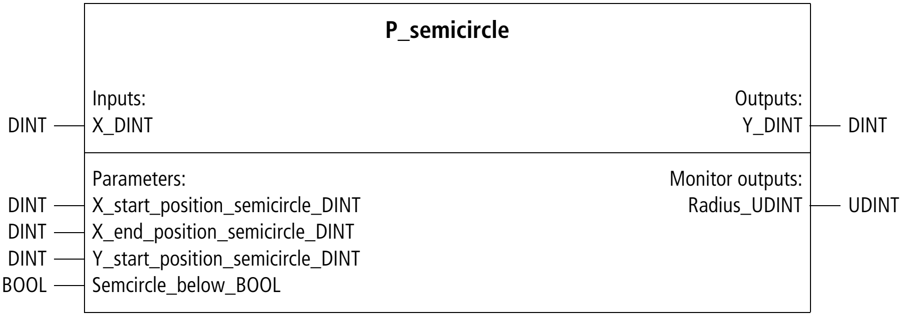
Example:
In the application example, "P_semi_ellipsis" is combined with a 10-point interpolation module. The X-Y curve results from combining master axis 1 and slave axis 2. Since "Semi_ellipsis_bottom_BOOL" is set to "0" and the specified eccentricity is "3", the semi-ellipsis runs between the X values (master) "60 000" and "100 000" and between the Y values (slave) "0" and "60 000".

Application of function block "P_semi_ellipsis" in program "Ellipsis"

```
PROGRAM Ellipsis
VAR
      Semi_ellipsis : P_semi_ellipsis ;
      IP10_DINT_interpolation : P_IP10_DINT_interpolation ;
      Axis_01 : P_closed_loop_position_control ;
      Axis_02 : P_closed_loop_position_control ;
END_VAR

CAL semi_ellipsis(
      X_DINT :=axis_01.actual_position_DINT,
      X_start_position_semi_ellipsis_DINT :=60000,
      X_end_position_semi_ellipsis_DINT :=100000,
      Y_start_position_semi_ellipsis_DINT :=0,
```

```
      Eccentricity_Y_to_X_10thpercent_UINT :=3000,
      Semi_ellipsis_bottom_BOOL :=0,
      Radius_X_UDINT=>20000,
      Radius_Y_UDINT=>60000
      )

CAL IP10_DINT_interpolation(
      X_DINT :=axis_01.actual_position_DINT,
      Suppress_extrapolation_BOOL :=1,
      X1_DINT :=0,
      X2_DINT :=20000,
      X3_DINT :=40000,
      X4_DINT :=60000,
      X5_DINT :=60000,
      X6_DINT :=100000,
      X7_DINT :=100000,
      X8_DINT :=100000,
      X9_DINT :=100000,
      X10_DINT :=100000,
      Y1_DINT :=0,
      Y2_DINT :=0,
      Y3_DINT :=20000,
      Y4_DINT :=0,
      Y5_DINT :=Semi_ellipsis.Y_DINT,
      Y6_DINT :=Semi_ellipsis.Y_DINT,
      Y7_DINT :=Semi_ellipsis.Y_DINT,
      Y8_DINT :=Semi_ellipsis.Y_DINT,
      Y9_DINT :=Semi_ellipsis.Y_DINT,
      Y10_DINT :=Semi_ellipsis.Y_DINT
      )
LD    IP10_DINT_interpolation.Y_DINT
ST    Axis_02.Setpoint_position_command_master_axis_DINT

END_PROGRAM
```

P_semicircle
Semicircle



Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
X_DINT	X value of the circle	-10^9 to 10^9
Parameters		
X_start_position_semicircle_DINT	Starting X value of the semicircle	-10^9 to 10^9
X_end_position_semicircle_DINT	End X value of the semicircle	-10^9 to 10^9
Y_start_position_semicircle_DINT	Starting Y value of the semicircle	-10^9 to 10^9
Semcircle_below_BOOL	Selection whether the semicircle runs at top (0) or bottom (1)	0/1
Outputs		
Y_DINT	Y value of the semicircle	-10^9 to 10^9
Monitor outputs		
Radius_UDINT	Radius of the circle	0 to 2×10^9

Description

This function block generates a circular curve of the X/Y values between the definable X/Y start and end values. This functionality can be used, for example, for a master-slave positioning process. With parameter "Semicircle_below_BOOL", the user can select whether the semicircle will be above (larger than the Y starting value) or below (smaller than the Y starting value). The monitor outputs are the radius of the circle in the X and Y direction.

Example:
In the application example, "P_semicircle" is combined with a 10-point interpolation module. The X-Y curve results from combining master axis 1 and slave axis 2. Since "Semicircle_below_BOOL" is set to "1" and the specified eccentricity is "", the semicircle runs between the X values (master) "60000" and "100000" and between the Y values (slave) "0" and "–20000".

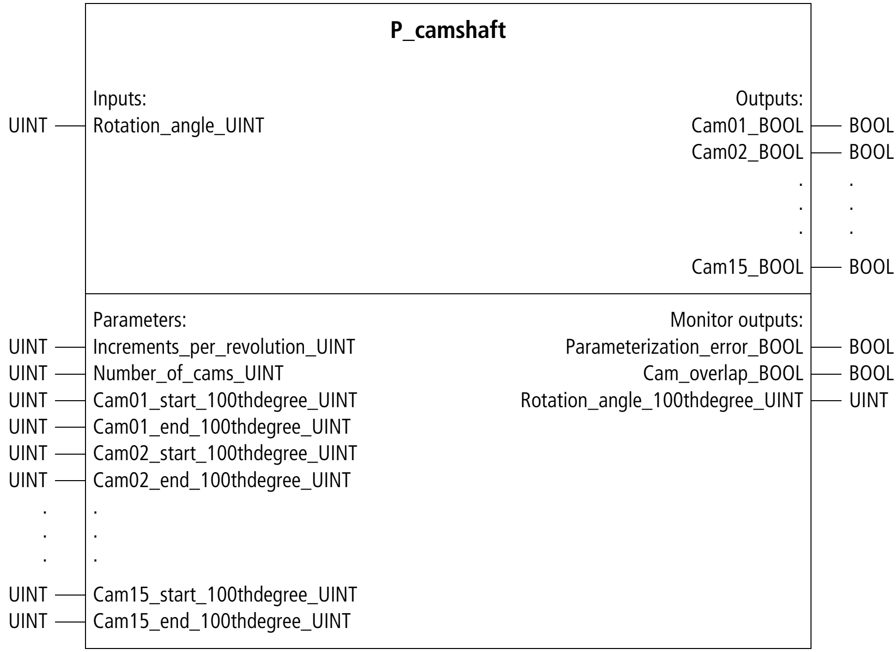
Application of function block "P_semicircle" in program "Circle"

```
PROGRAM Circle
VAR
      Semicircle : P_semicircle ;
      IP10_DINT_interpolation : P_IP10_DINT_interpolation ;
      Axis_01 : P_closed_loop_position_control ;
      Axis_02 : P_closed_loop_position_control ;
END_VAR

CAL semicircle(
      X_DINT :=axis_01.actual_position_DINT,
      X_start_position_semicircle_DINT :=60000,
      X_end_position_semicircle_DINT :=100000,
      Y_start_position_semicircle_DINT :=0,
      Semcircle_below_BOOL :=1
      )
```

```
CAL IP10_DINT_interpolation(
      X_DINT :=Axis_01.actual_position_DINT,
      Suppress_extrapolation_BOOL :=1,
      X1_DINT :=0,
      X2_DINT :=20000,
      X3_DINT :=40000,
      X4_DINT :=60000,
      X5_DINT :=60000,
      X6_DINT :=100000,
      X7_DINT :=100000,
      X8_DINT :=100000,
      X9_DINT :=100000,
      X10_DINT :=100000,
      Y1_DINT :=0,
      Y2_DINT :=0,
      Y3_DINT :=20000,
      Y4_DINT :=0,
      Y5_DINT :=semicircle.Y_DINT,
      Y6_DINT :=semicircle.Y_DINT,
      Y7_DINT :=semicircle.Y_DINT,
      Y8_DINT :=semicircle.Y_DINT,
      Y9_DINT :=semicircle.Y_DINT,
      Y10_DINT :=semicircle.Y_DINT
      )
LD    IP10_DINT_interpolation.Y_DINT
ST    Axis_02.Setpoint_position_command_master_axis_DINT

END_PROGRAM
```

Camshaft

P_camshaft
Camshaft with up to 15 cams



Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Rotation_angle_UINT	Angle of rotation in increments	0 to 65535
Parameters		
Increments_per_revolution_UINT	Increments per revolution	0 to 65535
Number_of_cams_UINT	Number of cams	1 to 15
Cam01_start_100thdegree_UINT	Start of cam 1 in hundredths of a degree	0 to 65535
Cam01_end_100thdegree_UINT	End of cam 1 in hundredths of a degree	0 to 65535

Designation	Function	Value range
Cam02_start_100thdegree_UINT	Start of cam 2 in hundredths of a degree	0 to 65535
Cam02_end_100thdegree_UINT	End of cam 2 in hundredths of a degree	0 to 65535
Cam03_start_100thdegree_UINT	Start of cam 3 in hundredths of a degree	0 to 65535
Cam03_end_100thdegree_UINT	End of cam 3 in hundredths of a degree	0 to 65535
Cam04_start_100thdegree_UINT	Start of cam 4 in hundredths of a degree	0 to 65535
Cam04_end_100thdegree_UINT	End of cam 4 in hundredths of a degree	0 to 65535
Cam05_start_100thdegree_UINT	Start of cam 5 in hundredths of a degree	0 to 65535
Cam05_end_100thdegree_UINT	End of cam 5 in hundredths of a degree	0 to 65535
Cam06_start_100thdegree_UINT	Start of cam 6 in hundredths of a degree	0 to 65535
Cam06_end_100thdegree_UINT	End of cam 6 in hundredths of a degree	0 to 65535
Cam07_start_100thdegree_UINT	Start of cam 7 in hundredths of a degree	0 to 65535
Cam07_end_100thdegree_UINT	End of cam 7 in hundredths of a degree	0 to 65535
Cam08_start_100thdegree_UINT	Start of cam 8 in hundredths of a degree	0 to 65535
Cam08_end_100thdegree_UINT	End of cam 8 in hundredths of a degree	0 to 65535
Cam09_start_100thdegree_UINT	Start of cam 9 in hundredths of a degree	0 to 65535
Cam09_end_100thdegree_UINT	End of cam 9 in hundredths of a degree	0 to 65535
Cam10_start_100thdegree_UINT	Start of cam 10 in hundredths of a degree	0 to 65535
Cam10_end_100thdegree_UINT	End of cam 10 in hundredths of a degree	0 to 65535
Cam11_start_100thdegree_UINT	Start of cam 11 in hundredths of a degree	0 to 65535
Cam11_end_100thdegree_UINT	End of cam 11 in hundredths of a degree	0 to 65535
Cam12_start_100thdegree_UINT	Start of cam 12 in hundredths of a degree	0 to 65535
Cam12_end_100thdegree_UINT	End of cam 12 in hundredths of a degree	0 to 65535
Cam13_start_100thdegree_UINT	Start of cam 13 in hundredths of a degree	0 to 65535
Cam13_end_100thdegree_UINT	End of cam 13 in hundredths of a degree	0 to 65535
Cam14_start_100thdegree_UINT	Start of cam 14 in hundredths of a degree	0 to 65535
Cam14_end_100thdegree_UINT	End of cam 14 in hundredths of a degree	0 to 65535
Cam15_start_100thdegree_UINT	Start of cam 15 in hundredths of a degree	0 to 65535
Cam15_end_100thdegree_UINT	End of cam 15 in hundredths of a degree	0 to 65535

Designation	Function	Value range
Outputs		
Cam01_BOOL	Signal: Cam 1	0/1
Cam02_BOOL	Signal: Cam 2	0/1
Cam03_BOOL	Signal: Cam 3	0/1
Cam04_BOOL	Signal: Cam 4	0/1
Cam05_BOOL	Signal: Cam 5	0/1
Cam06_BOOL	Signal: Cam 6	0/1
Cam07_BOOL	Signal: Cam 7	0/1
Cam08_BOOL	Signal: Cam 8	0/1
Cam09_BOOL	Signal: Cam 9	0/1
Cam10_BOOL	Signal: Cam 10	0/1
Cam11_BOOL	Signal: Cam 11	0/1
Cam12_BOOL	Signal: Cam 12	0/1
Cam13_BOOL	Signal: Cam 13	0/1
Cam14_BOOL	Signal: Cam 14	0/1
Cam15_BOOL	Signal: Cam 15	0/1
Monitor outputs		
Parameterization_error_BOOL	Status: A parameterization fault has occurred, for example a cam overlap or an input of zero or above 16 for the number of cams.	0/1
Cam_overlap_BOOL	Status: The cams overlap, because they were not parameterized in ascending order.	0/1
Rotation_angle_100thdegree_UINT	Angle of rotation in hundredths of a degree	0 to 36000

Description

This function block simulates a camshaft. For the input, enter an angle of rotation in any suitable unit (resolution). When "Increments_per_revolution_UINT" is entered, an internal conversion to "Rotation_angle_100thdegree_UINT" is performed. The start and end of each cam must be entered in hundredths of a degree. The number of cams must be in the range zero to 16.

At the function block's output, the BOOL variables indicate, whether the angle of rotation lies inside the range of a cam or not (→ fig. 16). Status signals "Parameterization error" and (inadmissible) "Cam overlap" are provided by two BOOL variables.

Special case

If a cam is to cover the zero point, this cam must be entered last (→ fig. 16). For example, a camshaft with three cams is to be configured. The third cam ranges from 320 to 20°. For this cam, a starting value of 32000 and an end value of 38000 (= 36000 + 2000) must be entered (→ sample application).

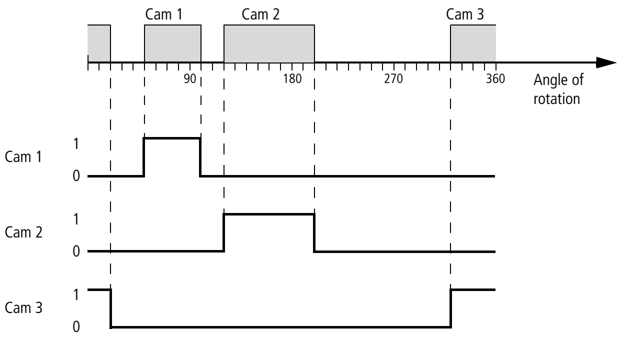


Figure 16: Camshaft with three cams. The third cam lies across the zero point.

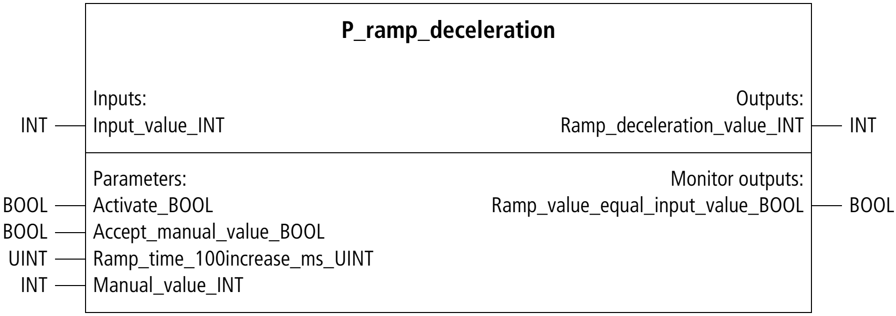
Application of function block "P_camshaft" in program "Cam03"

```
PROGRAM Cam03
VAR
        camshaft : P_camshaft ;
        Rotation_angle_100thdegree_UINT : UINT ;
END_VAR

CAL camshaft(
        Rotation_angle_UINT :=Rotation_angle_100thdegree_UINT,
        Increments_per_revolution_UINT :=3600,
        Number_of_cams_UINT :=3,
        Cam01_start_100thdegree_UINT :=5000,
        Cam01_end_100thdegree_UINT :=10000,
        Cam02_start_100thdegree_UINT :=12000,
        Cam02_end_100thdegree_UINT :=20000,
        Cam03_start_100thdegree_UINT :=32000,
        Cam03_end_100thdegree_UINT :=38000
        )
```

```
LD      camshaft.Cam01_BOOL
JMPCN   NO_ACTION_01
        (*
        1. program section
        *)
NO_ACTION_01:

LD      camshaft.Cam02_BOOL
JMPCN   NO_ACTION_02
        (*
        2. program section
        *)
NO_ACTION_02:

LD      camshaft.Cam03_BOOL
JMPCN   NO_ACTION_03
        (*
        3. program section
        *)
NO_ACTION_03:

END_PROGRAM
```

Ramp deceleration

P_ramp_deceleration
Ramp deceleration of an input value



Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Input_value_INT	Input value	–32768 to 32767
Parameters		
Activate_BOOL	Activates the function block	0/1
Accept_manual_value_BOOL	Accept manual value	0/1
Ramp_time_100increase_ms_UINT	Ramp time required for an increase of 100 increments, in ms	0 to 65535
Manual_value_INT	Bumpless acceptance of manual value	–32768 to 32767
Outputs		
Ramp_deceleration_value_INT	Ramp-delayed value	–32768 to 32767
Monitor outputs		
Ramp_value_equal_input_value_BOOL	Display: Ramp value corresponds with input value	0/1

Description

This function block delays the input value according to a maximum ramp slope (similar to PT1 filter). The ramp slope can be specified with “Ramp_time_100increase_ ms_UINT”. Within the entered ramp time, “Ramp_deceleration_value_INT” rises (or falls) by up to 100 increments.

Example:
A ramp time “Ramp_time_100increase_ms_UINT = 200” was specified. This results in an increase of the ramp deceleration value by 100 increments per 20 seconds if “Input_value_INT” is greater than “Ramp_deceleration_value_INT”, and falls by 100 increments per 20 seconds if “Input_value_INT” is smaller than “Ramp_deceleration_value_INT”.

If the input value and the (ramp-)decelerationed value match, “Ramp_value_equal_input_value_BOOL” carries a “1”; if they do not, it carries a “0”. With “Accept_manual_value_BOOL=1”, “Manual_value_INT” is accepted.

Application of function block "P_ramp_deceleration" in program "St_deceleration"

```
PROGRAM St_deceleration

VAR
      ramp_deceleration: P_ramp_deceleration;
      Manipulated_variable_12bit_UINT : UINT ;
      Manipulated_variable_delayed_12bit_UINT : UINT ;
END_VAR
LD    Manipulated_variable_12bit_UINT
UINT_TO_INT
ST    ramp_deceleration.Input_value_INT
CAL ramp_deceleration(
      Activate_BOOL :=1,
      Accept_manual_value_BOOL :=0,
      Ramp_time_100increase_10thms_UINT :=200,
      Manual_value_INT :=0
      )
LD    ramp_deceleration.Ramp_deceleration_value_INT
INT_TO_UINT
ST    Manipulated_variable_delayed_12bit_UINT

END_PROGRAM
```

Hydraulics

P_hydraulic
Functions for hydraulic axes

P_hydraulic

Type	Inputs:	Outputs:	Type
INT	Manipulated_variable_12bit_INT	Manipulated_variable_hydraulic_12Bit_INT	INT
DINT	Setpoint_position_DINT		
DINT	Actual_position_DINT		
BOOL	Actual_position_in_tolerance_zone_BOOL		
	Parameters:		
UINT	Hydraulic_zero_point_coverage_negative_UINT		
UINT	Hydraulic_zero_point_coverage_positive_UINT		
BOOL	Stick_slip_compensation_activate_BOOL		
UINT	Hydraulic_stick_slip_compensation_negative_11Bit_UINT		
UINT	Hydraulic_stick_slip_compensation_positive_11Bit_UINT		
UINT	Hydraulic_stick_slip_zone_negative_UINT		
UINT	Hydraulic_stick_slip_zone_positive_UINT		

Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Manipulated_variable_12bit_INT	Manipulated variable of a basic positioning function block	–2048 to 2048
Setpoint_position_DINT	Setpoint position (target position)	-10^9 to 10^9
Actual_position_DINT	Actual position	-10^9 to 10^9
Actual_position_in_tolerance_zone_BOOL	Actual position is within the tolerance zone of the basic positioning function block	0/1

Designation	Function	Value range
Parameters		
Hydraulic_zero_point_coverage_negative_UINT	Negative zero point coverage for hydraulic axes	0 to 65535
Hydraulic_zero_point_coverage_positive_UINT	Positive zero point coverage for hydraulic axes	0 to 65535
Stick_slip_compensation_activate_BOOL	Activate stick-slip compensation	0/1
Hydraulic_stick_slip_compensation_negative_11Bit_UINT	Stick-slip compensation Ks, negative	0 to 65535
Hydraulic_stick_slip_compensation_positive_11Bit_UINT	Stick-slip compensation Ks, positive	0 to 65535
Hydraulic_stick_slip_zone_negative_UINT	Negative stick-slip compensation is active only within this zone	0 to 65535
Hydraulic_stick_slip_zone_positive_UINT	Positive stick-slip compensation is active only within this zone	0 to 65535
Outputs		
Manipulated_variable_hydraulic_12Bit_INT	Manipulated variable for hydraulic axes	–2048 to 2047

Description

This function block can be linked with the basic positioning function blocks (→ chapter 3), in particular with function block “P_basic_position_control”. The manipulated variable and “Actual position in tolerance zone” must be linked with the outputs of the basic positioning function block. The setpoint position and the actual position must be associated with the corresponding variables.

Unlike electrical servo motors, servo-hydraulic axes can exhibit a significant stick-slip effect (transitions between sticking and slipping friction). To compensate for the stick-slip effect, a stick-slip value Ks can each be defined in the positive and negative direction. Within a stick-slip zone, this Ks value is then added to the "normal" manipulated variable.

Example:
Hydraulic_stick_slip_compensation_negative_11Bit_UINT = 500
Hydraulic_stick_slip_compensation_positive_11Bit_UINT = 500
Hydraulic_stick_slip_zone_negative_UINT = 20
Hydraulic_stick_slip_zone_positive_UINT = 20
=>
At the start of a positioning process, the manipulated variable is increased by 500 until the actual position is removed from its starting position by more than 20 increments.

When servo-hydraulic axes are used, a zero-point overlap exists in general, i.e. small manipulated variables about the zero point have no effect. This can be corrected with parameters "Hydraulic_zero_point_coverage_negative_UINT" and "Hydraulic_zero_point_coverage_positive_UINT".

→ For commissioning, proceed as follows:

Manually (through direct assignment of the analog outputs) determine the positive and negative manipulated variables at which the positioning axis responds and enter these values.

Example:
In this application example, an incremental encoder value is recorded. With function block “P_incremental_ encoder_evaluation”, the incremental values are processed in such a way as to prevent a data over-range. Referencing (definition of the zero points) is carried out when digital input “2” has a a rising edge. With digital input “0”, the setpoint position can be changed. Digital input “1” activates the basic positioning block. The hydraulic functions “Zero-point overlap” and “Stick-slip compensation” are created by function block instance “Hydraulic_01”.

Stick-slip compensation is activated with digital input “3”.

Application of function block “P_hydraulic” in program “Pos_08”

```
PROGRAM Pos_08
VAR
        hydraulic_01 : P_hydraulic ;
        Incremental_encoder_evaluation_01 : P_incremental_encoder_evaluation ;
        axis_01 : P_basic_position_control ;
        AAINT_INT_analog_output : P_AAINT_INT_analog_output ;
        DI_0_0_BOOL : BOOL ;
        DI_0_1_BOOL : BOOL ;
        DI_0_2_BOOL : BOOL ;
        DI_0_3_BOOL : BOOL ;
        AO_0_0_2_0 : INT ;
        Setpoint_position_01_DINT : DINT ;
        Actual_position_01_DINT : DINT ;
        Incremental_encoder_01 : UDINT ;
END_VAR
```

```
LD      DI_0_0_BOOL
JMPCN   SETPOINT_POSITION_02
        LD    5000
        ST    Setpoint_position_01_DINT
        JMP   E_SETPOINT_POSITION_02
SETPOINT_POSITION_02:
        LD    200000
        ST    Setpoint_position_01_DINT
E_SETPOINT_POSITION_02:

LD      Incremental_encoder_01
UDINT_TO_DINT
ST      Incremental_encoder_evaluation_01.Increments_DINT
CAL Incremental_encoder_evaluation_01(
        Machine_zero_point_DINT :=2000,
        Maximum_incremental_encoder_DINT :=16777215,
        Activate_BOOL :=1,
        Absolute_value_transmitter_BOOL :=1,
        Absolute_value_transmitter_without_referencing_BOOL :=0,
        Accept_machine_zero_point_BOOL :=DI_0_2_BOOL,
        Reference_signal_BOOL :=0,
        Actual_value_DINT=>Actual_position_01_DINT
        )
CAL axis_01(
        Setpoint_position_DINT :=Setpoint_position_01_DINT,
        Actual_position_DINT :=Actual_position_01_DINT,
        Activate_BOOL :=DI_0_1_BOOL,
        Manipulated_variable_negation_BOOL :=0,
        Accept_setpoint_position_BOOL :=1,
        Cycle_time_demand_optimize_BOOL :=0,
        Ramp_time_100increase_maximum_ms_UINT :=50,
        Deceleration_position_deviation_UDINT :=20000,
        Rounding_position_deviation_15bit_UINT :=2000,
        Manipulated_variable_max_11Bit_UINT :=1500,
        Tolerance_positioning_zone_UINT :=20
        )
```

```
LD      axis_01.Manipulated_variable_12Bit_INT
ST      hydraulic_01.Manipulated_variable_12Bit_INT
LD      axis_01.Actual_position_in_tolerance_zone_BOOL
ST      hydraulic_01.Actual_position_in_tolerance_zone_BOOL
LD      axis_01.Set_position_current_job_DINT
ST      hydraulic_01.Setpoint_position_DINT
CAL hydraulic_01(
        Actual_position_DINT :=Actual_position_01_DINT,
        Hydraulic_zero_point_coverage_negative_UINT :=0,
        Hydraulic_zero_point_coverage_positive_UINT :=0,
        Stick_slip_compensation_activate_BOOL :=DI_0_3_BOOL,
        Hydraulic_stick_slip_compensation_negative_11Bit_UINT :=500,
        Hydraulic_stick_slip_compensation_positive_11Bit_UINT :=500,
        Hydraulic_stick_slip_zone_negative_UINT :=10,
        Hydraulic_stick_slip_zone_positive_UINT :=10
        )
CAL AAINT_INT_analog_output(
        Input_value_INT :=hydraulic_01.Manipulated_variable_hydraulic_12Bit_INT,
        Analog_output_INT=>AO_0_0_2_0
        )
END_PROGRAM
```

Referencing

P_referencing
Automatic referencing with basic position control

P_referencing

Type	Inputs:	Outputs:	Type
DINT	Setpoint_position_DINT	Setpoint_position_basic_position_control_DINT	DINT
DINT	Increments_DINT	Actual_position_basic_position_control_DINT	DINT
		Manipulated_var_max_negative_basic_position_control_11Bit_UINT	UINT
		Manipulated_var_max_positive_basic_position_control_11Bit_UINT	UINT
		Accept_setpoint_position_basic_position_control_BOOL	BOOL

Type	Parameters:	Monitor outputs:	Type
BOOL	Absolute_value_transmitter_BOOL	Parameterization_error_BOOL	BOOL
BOOL	Absolute_value_transmitter_without_referencing_BOOL	Parameterization_error_BOOL	BOOL
		Referencing_finished_BOOL	BOOL
BOOL	Activate_BOOL	Software_limit_attained_BOOL	BOOL
BOOL	Accept_setpoint_position_BOOL		
BOOL	Manual_control_negative_BOOL		
BOOL	Manual_control_positive_BOOL		
BOOL	Referencing_negative_BOOL		
BOOL	Referencing_positive_BOOL		
BOOL	Reference_signal_cam_BOOL		
UINT	Manual_manipulated_variable_11Bit_UINT		
DINT	Manual_control_step_width_DINT		
DINT	Machine_zero_point_DINT		
DINT	Software_limit_negative_DINT		
DINT	Software_limit_positive_DINT		
UINT	Reference_cam_searching_manipulated_variable_11Bit_UINT		
UINT	Reference_point_searching_manipulated_variable_11Bit_UINT		
UINT	Manipulated_variable_max_negative_11bit_UINT		
UINT	Manipulated_variable_max_positive_11bit_UINT		
DINT	Maximum_incremental_encoder_DINT		

Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Setpoint_position_DINT	Setpoint position of the automatic control. This value is overridden during referencing	-10^9 to 10^9
Increments_DINT	Value from the incremental or absolute encoder	-10^9 to 10^9
Parameters		
Absolute_value_transmitter_BOOL	Transmitter type: "Incremental encoder" = 0, "Absolute encoder" = 1. For "Incremental encoder", referencing with a reference cam scan can be carried out. With "Absolute encoder", referencing is carried out without reference cam scan.	0/1
Absolute_value_transmitter_ without_referencing_BOOL	Regardless of the settings for "Absolute_value_transmitter_BOOL", no referencing is carried out here.	0/1
Activate_BOOL	Activates the function block	0/1
Accept_setpoint_position_BOOL	Control mode: Accept applied setpoint positions (automatic mode). This mode overrides manual mode.	0/1
Manual_control_negative_BOOL	Control mode: Manual control (and jogging for non-zero manual control step widths), negative	0/1
Manual_control_positive_BOOL	Control mode: Manual control (and jogging for non-zero manual control step widths), positive	0/1
Referencing_negative_BOOL	Control mode: Negative referencing	0/1
Referencing_positive_BOOL	Control mode: Positive referencing	0/1
Reference_signal_cam_BOOL	Signal: Reference cam	0/1
Manual_manipulated_variable_ 11Bit_UINT	Maximum manipulated variable for manual mode	0 to 2048

Designation	Function	Value range
Manual_control_step_width_DINT	Step width for manual mode (values ≠ 0 => jogging)	-10^9 to 10^9
Machine_zero_point_DINT	Machine zero point	-10^9 to 10^9
Software_limit_negative_DINT	Negative software limit (effective only after referencing)	-10^9 to 10^9
Software_limit_positive_DINT	Positive software limit (effective only after referencing)	-10^9 to 10^9
Reference_cam_searching_ manipulated_variable_11Bit_UINT	Manipulated variable for reference cam searching	0 to 2048
Reference_point_searching_ manipulated_variable_11Bit_UINT	Manipulated variable for reference point searching (edge of reference cam)	0 to 2048
manipulated_variable_max_ negative_11bit_UINT	Maximum negative manipulated variable for automatic mode	0 to 2048
Manipulated_variable_max_ positive_11bit_UINT	Maximum positive manipulated variable for automatic mode	0 to 2048
Maximum_incremental_encoder_ DINT	Maximum value from the incremental encoder (before overflow)	0 to 10^9
Outputs		
Setpoint_position_basic_position_ control_DINT	Setpoint position; must be linked with basic position control	-10^9 to 10^9
Actual_position_basic_position_ control_DINT	Actual position; must be linked with basic position control	-10^9 to 10^9
Manip_var_max_negative_basic_ position_control_11Bit_UINT	Maximum negative manipulated variable; must be linked with basic position control	0 to 2048
Manip_var_max_positive_basic_ position_control_11Bit_UINT	Maximum positive manipulated variable; must be linked with basic position control	0 to 2048
Accept_setpoint_position_basic_ position_control_BOOL	Acceptance of applied setpoint positions (automatic control), must be linked with basic position control	0/1

Designation	Function	Value range
Monitor outputs		
Parameterization_error_BOOL	Status: Parameterization error, e.g. simultaneous referencing in positive and negative direction	0/1
Software_limit_attained_BOOL	Status: The setpoint position lies outside the software limits and is constrained to within the software limits	0/1
Referencing_finished_BOOL	Status: Referencing finished	0/1

Description

With function block “P_referencing”, the following functions can be implemented in combination with function block “P_basic_position_control”:

- Manual mode
- Referencing
- Automatic mode (accept setpoint positions)

Variables “Setpoint_position_DINT”, “Manipulated_variable_max_negative_11bit_UINT” and “Manipulated_variable_max_positive_11bit_UINT” must contain the values that are to be supplied to basic position control with outputs “Setpoint_position_basic_position_control_DINT” and “Manipulated_variable_maximum_basic_position_control_11Bit_UINT” if no referencing is carried out and “Accept_setpoint_position_BOOL” carries a “1”. During referencing and manual control, deviating setpoint positions and maximum manipulated variables are generated, which specify the speed and direction of manual control and referencing.

The incremental or absolute encoder output values must be linked to “Increments_DINT” and the reference cam signal to “Reference_signal_cam_BOOL”.

The largest possible value from the incremental or absolute encoder must be assigned to parameter “Maximum_incremental_ encoder_DINT”, for example 65535 for 16 bit resolution or 16777215 for 24 bit resolution.

The function block’s output variables marked basic position control must be linked to the corresponding input variables of “P_basic_position_control”. If the software limits are reached or breached, the setpoint position is limited to the corresponding software limit and status signal “Software_limit_attained_BOOL=1” is output (provided referencing has taken place).

Manual mode

Parameter “Manual_manipulated_variable_11Bit_UINT” specifies the maximum manipulated variable at which traversing is possible in manual control mode. The function of the manual control mode is linked with parameter “Manual_control_step_width_DINT”. The following applies:

“Manual_control_step_width_DINT = 0”
=>
When manual control is activated, the axis is traversed in the direction of the positive or negative software limits (only after referencing; → below).

“Manual_control_step_width_DINT” ≠ “0”
=>
Each time manual control is activated, the corresponding step width is traversed (jogging).

Referencing with referencing

With this function block, an automatic referencing process with referencing can be carried out when parameters "Absolute_value_transmitter_BOOL" and "Absolute_value_transmitter_without_referencing_BOOL" are set to "0".

The referencing process then takes place as follows:

First, the reference cam is searched for in the specified direction at the selected reference cam search speed. Once the reference cam has been found, the search operation is cancelled. The axis is then traversed in the opposite direction at the specified reference point search speed until the reference cam is left again (in this direction). Referencing (definition of the zero point) takes place when the reference cam leaves the transducer. This machine zero point can be offset with parameter "Machine_zero_point_DINT".

→ Referencing takes place independently of an incremental encoder's zero marker signal. The referencing options of this function block differ from those of function block "P_closed_loop_position_control".

After activation of the function block, the referencing process in the direction of the reference cam can be started with "Referencing_negative_BOOL" or "Referencing_positive_BOOL". The end of the referencing operation is indicated by monitor output "Referencing_finished_BOOL". The referencing commands must then be set to "0" in both the positive and negative directions, so that automatic and manual operation can be started.

→ The repetition accuracy of the referencing operation depends on the reference point search manipulated variable. The smaller this manipulated variable, the greater the repetition accuracy of the referencing operation.

Referencing without reference cam search

An other possibility for referencing is to set "Absolute_value_transmitter_BOOL" to "1", "Absolute_value_transmitter_without_referencing_BOOL" to "0" and "Reference_cam_searching_manipulated_variable_11Bit_UINT" to "0". Referencing is triggered if the following three parameters have a falling flank at the same time:

- Referencing_negative_BOOL = falling edge
- Referencing_positive_BOOL = rising edge
- Reference_signal_cam_BOOL = falling edge

→ This referencing process can also be used for incremental coordinate positioning.

→ Before referencing (both processes), automatic and manual operation are not bound by the software limits. If, however, parameter "Absolute_position_transmitter_without_referencing_ BOOL" is set to "1", referencing cannot be carried out and automatic and manual operation are then limited by the software limits.

Example:
Sample application "Pos_09" demonstrates how basic positioning can be linked for test purposes with axis simulation and the referencing function block. At the end of the program, the setpoint value and the setpoint and actual positions are assigned to several marker words. The following functions are assigned to the digital inputs:

- Digital input "0" = Change setpoint position
- Digital input "1" = Activate basic position control
- Digital input "2" = Activate referencing function block
- Digital input "3" = Activate axis simulation
- Digital input "4" = Referencing in negative direction
- Digital input "5" = Referencing in positive direction
- Digital input "6" = Manual control in negative direction
- Digital input "7" = Manual control in positive direction

Application of function block "P_referencing" in program "Pos_09"

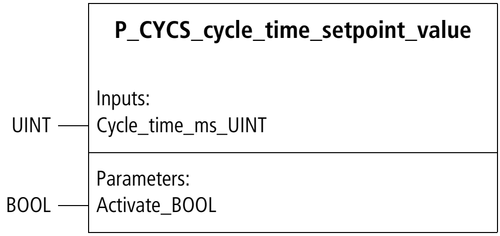
```
PROGRAM Pos_09
VAR
       referencing_01 : P_referencing ;
       basic_position_control_01 : P_basic_position_control ;
       axis_simulation_01 : P_axis_simulation ;
       DI_0_0_BOOL : BOOL ;
       DI_0_1_BOOL : BOOL ;
       DI_0_2_BOOL : BOOL ;
       DI_0_3_BOOL : BOOL ;
       DI_0_4_BOOL : BOOL ;
       DI_0_5_BOOL : BOOL ;
       DI_0_6_BOOL : BOOL ;
       Setpoint_position_01_DINT : DINT ;
       Actual_position_01_DINT : DINT ;
       Reference_signal_cam_01_BOOL : BOOL ;
END_VAR

(*========================================================*)
(*====== Setpoint position variation =====================*)
(*========================================================*)
LD     DI_0_0_BOOL
JMPCN  SETPOINT_POSITION_02
       LD    60000
       ST    Setpoint_position_01_DINT
       JMP   E_SETPOINT_POSITION_02
SETPOINT_POSITION_02:
       LD    2000
       ST    Setpoint_position_01_DINT
E_SETPOINT_POSITION_02:
```

```
(*=========================================================*)
(*= referencing with an integrated ======================*)
(*= increment encoder evaluation ===========================*)
(*=========================================================*)

LD      axis_simulation_01.Incremental_encoder_Output_UDINT
UDINT_TO_DINT
ST      referencing_01.Increments_DINT
CAL referencing_01(
        Setpoint_position_DINT :=Setpoint_position_01_DINT,
        Absolute_value_transmitter_BOOL :=0,
        Absolute_value_transmitter_without_referencing_BOOL :=0,
        Activate_BOOL :=DI_0_2_BOOL,
        Accept_setpoint_position_BOOL :=DI_0_6_BOOL,
        Referencing_negative_BOOL :=DI_0_4_BOOL,
        Referencing_positive_BOOL :=DI_0_5_BOOL,
        Reference_signal_cam_BOOL :=Reference_signal_cam_01_BOOL,
        Machine_zero_point_DINT :=2000,
        Software_limit_negative_DINT :=0,
        Software_limit_positive_DINT :=70000,
        Reference_cam_searching_manipulated_variable_11Bit_UINT :=150,
        Reference_point_searching_manipulated_variable_11Bit_UINT :=15,
        Manipulated_variable_max_negative_11Bit_UINT :=1500,
        Manipulated_variable_max_positive_11Bit_UINT :=1500,
        Maximum_incremental_encoder_DINT :=16777215,
        Actual_position_basic_position_control_DINT=>Actual_position_01_DINT
        )

(*=========================================================*)
(*======= basic position control =========================*)
(*=========================================================*)
```

```
LD     referencing_01.Setpoint_position_basic_position_control_DINT
ST     basic_position_control_01.Setpoint_position_DINT
LD     referencing_01.Manipulated_variable_max_neg_basic_position_control_11Bit_UINT
ST     basic_position_control_01.Manipulated_variable_maximum_11Bit_UINT
LD     referencing_01.Accept_setpoint_position_basic_position_control_BOOL
ST     basic_position_control_01.Accept_setpoint_position_BOOL
CAL basic_position_control_01(
       Actual_position_DINT :=Actual_position_01_DINT,
       Activate_BOOL :=DI_0_1_BOOL,
       Manipulated_variable_negation_BOOL :=0,
       Cycle_time_demand_optimize_BOOL :=0,
       Ramp_time_100increase_maximum_ms_UINT :=500,
       Deceleration_position_deviation_UDINT :=30000,
       Rounding_position_deviation_15bit_UINT :=1000,
       Tolerance_positioning_zone_UINT :=30
       )

(*==========================================================*)
(*====== simulation of the axis ==========================*)
(*==========================================================*)

LD     basic_position_control_01.Manipulated_variable_12Bit_INT
ST     axis_simulation_01.Manipulated_variable_12Bit_INT
CAL axis_simulation_01(
       Activate_BOOL :=DI_0_3_BOOL,
       Accept_manual_value_BOOL :=0,
       Nominal_revolutions_per_minute_INT :=2000,
       Manual_value_UDINT :=0,
       Increments_per_revolution_UINT :=1024,
       Reference_cam_signal_BOOL=>Reference_signal_cam_01_BOOL
       )

END_PROGRAM
```

Constant cycle time

P_CYCS_cycle_time_setpoint_value
Setting a constant cycle time to setpoint value



Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Cycle_time_ms_UINT	Desired PLC cycle time	1 to 250
Parameters		
Activate_BOOL	Activates the function block	0/1

Description

With this function block, a setpoint PLC cycle time can be entered. The cycle time is set when the greatest actual cycle times of the user program are shorter than this value and the PLC works cyclically. The ideal positioning behaviour is achieved in particular at a constant cycle time of 10 ms.

→ If the PLC cycle time exceeds the specified setpoint, nothing happens (i.e. the PLC does not switch to “Halt” if the PLC works cyclical); it means merely that the set cycle time cannot be achieved.

If the PLC operates periodically, this function block may not be used. It is not a requirement as automatic constant cycle times are present when the PLC operates periodically.

Example:
In the user program below, the instructions and function block calls generate an average cycle time of about 22 ms ±4 ms. Assigning a setpoint of 30 ms changes this cycle time to a constant value.

Application of function block “P_CYCS_cycle_time_setpoint_value” in program “scantcon”

```
PROGRAM scantcon
VAR
      CYCS_cycle_time_setpoint_value : P_CYCS_cycle_time_setpoint_value ;
END_VAR


(* Calling other function blocks result in PLC cycle times below or equal to 26 ms.
Function block "P_CYCS_cycle_time_setpoint_value" is called at the end of the
program*)


CAL CYCS_cycle_time_setpoint_value(
      Activate_BOOL :=1,
      Cycle_time_ms_UINT :=30)


END_PROGRAM
```

Load compensation

P_load_compensation
Load compensation at the end of a positioning process

P_load_compensation

Type	Inputs:	Outputs:	Type
INT	Manipulated_variable_position_controller_12Bit_INT	Manipulated_variable_12bit_INT	INT
DINT	Position_deviation_DINT		
INT	Contouring_error_INT		
BOOL	Positioning_job_finished_BOOL		
BOOL	Actual_position_in_tolerance_zone_BOOL		
	Parameters:	**Monitor outputs:**	
BOOL	Activate_negative_BOOL	Manipulated_variable_load_compensation_negative_12Bit_INT	INT
BOOL	Integration_external_start_negative_BOOL		
UDINT	Limiting_position_deviation_negative_UDINT	Manipulated_variable_load_compensation_positive_12Bit_INT	INT
UINT	Maximum_manipulated_variable_load_compensation_negative_UINT	Position_deviation_job_finished_DINT	DINT
		Delay_time_ms_UINT	UINT
UINT	Reset_time_negative_100ths_UINT	Integrator_started_negative_BOOL	BOOL
UINT	Waiting_time_negative_ms_UINT	Integrator_started_positive_BOOL	BOOL
BOOL	Activate_positive_BOOL		
BOOL	Integration_external_start_positive_BOOL		
UDINT	Limiting_position_deviation_positive_UDINT		
UINT	Maximum_manipulated_variable_load_compensation_positive_UINT		
UINT	Reset_time_positive_100ths_UINT		
UINT	Waiting_time_positive_ms_UINT		

Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Manipulated_variable_position_ controller_12Bit_INT	Manipulated variable of the position controller	–2048 to 2047
Position_deviation_DINT	Position deviation	-10^9 to 10^9
Contouring_error_INT	Contouring error	–32768 to 32767
Positioning_job_finished_BOOL	Positioning job finished	0/1
Actual_position_in_tolerance_zone_ BOOL	Actual position in the tolerance zone	0/1
Parameters		
Activate_negative_BOOL	Activate in the negative direction	0/1
Integration_external_start_ negative_BOOL	Start of external integration in negative direction	0/1
Limiting_position_deviation_ negative_UDINT	Negative limiting position deviation	0 to 10^9
Maximum_manipulated_variable_ load_compensation_negative_UINT	Maximum manipulated load compensation variable in the negative direction	0 to 1000
Reset_time_negative_100ths_UINT	Reset time in hundredths of a second in the negative direction	0 to 65535
Waiting_time_negative_ms_UINT	Waiting time in ms in the negative direction	0 to 65535
Activate_positive_BOOL	Activate in the positive direction	0/1
Integration_external_start_ positive_BOOL	Start of external integration in positive direction	0/1

Designation	Function	Value range
Limiting_position_deviation_ positive_UDINT	Positive limiting position deviation	0 to 10^9
Maximum_manipulated_variable_ load_compensation_positive_UINT	Maximum manipulated load compensation variable in the positive direction	0 to 1000
Reset_time_positive_100ths_UINT	Reset time in hundredths of a second in the positive direction	0 to 65535
Waiting_time_positive_ms_UINT	Waiting time in ms in the positive direction	0 to 65535
Outputs		
Manipulated_variable_12bit_INT	Manipulated variable	–2048 to 2047
Monitor outputs		
Manipulated_variable_load_ compensation_negative_12Bit_INT	Negative manipulated load compensation variable	–1000 to 0
Manipulated_variable_load_ compensation_positive_12Bit_INT	Positive manipulated load compensation variable	0 to 1000
Position_deviation_job_finished_ DINT	Position deviation at completion of the positioning job	-10^9 to 10^9
Delay_time_ms_UINT	Delay between completion of the positioning job and actual position in the tolerance zone	0 to 65535
Integrator_started_negative_BOOL	Integrator started in negative direction	0/1
Integrator_started_positive_BOOL	Integrator started in positive direction	0/1

Description

If large forces act in opposition to the positioning direction during positioning, a significant position deviation may be the result. By integrating the manipulated variable, this function block can eliminate this position deviation. When the positioning job is finished, the function block checks whether the position deviation is greater (or, in the negative direction, smaller) than the specified limiting position deviation. If it is, the integrator is started. When the positioning job is finished, a waiting time starts. If the actual position is not within the tolerance zone after the waiting time, the integrator is started.

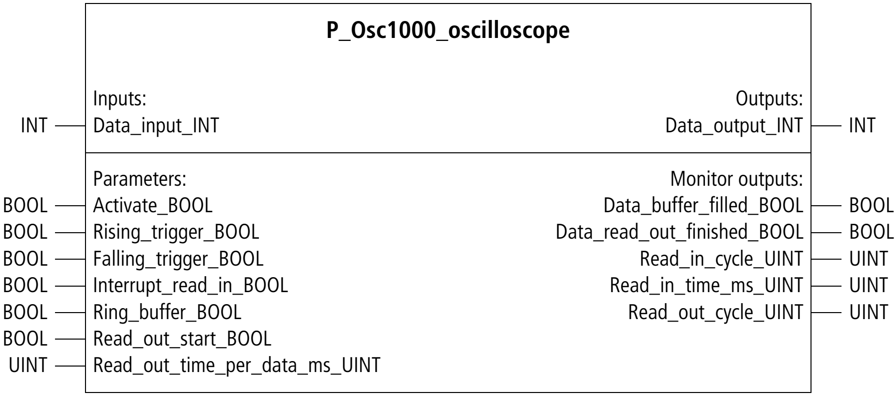
Example:
In the application example, a load compensation in the positive direction is carried out. If the position deviation is greater than 500 increments after completion of the positioning job, the integrator is activated in the positive direction. If, after completion of the positioning job and expiry of a waiting time of 1000 ms, the actual position lies outside the tolerance zone, the integrator is started in the positive direction.

Application if function block "P_load_compensation" in the program "load_c"

```
PROGRAM load_c
VAR
      Analog_output_INT : INT ;
      load_compensation : P_load_compensation ;
      axis1 : P_closed_loop_position_control ;
END_VAR

CAL load_compensation(
      Manipulated_variable_position_controller_12Bit_INT
      :=axis1.Manipulated_variable_12bit_INT,
      Position_deviation_DINT :=axis1.Position_deviation_DINT,
      Contouring_error_INT :=axis1.Contouring_error_INT,
      Positioning_job_finished_BOOL :=axis1.Positioning_job_finished_BOOL,
      Actual_position_in_tolerance_zone_BOOL
      :=axis1.Actual_position_in_tolerance_zone_BOOL,
      Activate_negative_BOOL :=0,
      Integration_external_start_negative_BOOL :=0,
      Limiting_position_deviation_negative_UDINT :=0,
      Maximum_manipulated_variable_load_compensation_negative_UINT :=0,
      Reset_time_negative_100ths_UINT :=0,
      Waiting_time_negative_ms_UINT :=0,
      Activate_positive_BOOL :=1,
      Integration_external_start_positive_BOOL :=0,
      Limiting_position_deviation_positive_UDINT :=500,
      Maximum_manipulated_variable_load_compensation_positive_UINT :=100,
      Reset_time_positive_100ths_UINT :=10,
      Waiting_time_positive_ms_UINT :=1000,
      Manipulated_variable_12bit_INT=>Analog_output_INT
      )

END_PROGRAM
```

Graphic representation of fast processes (oscilloscope)

P_Osc1000_oscilloscope Data buffer block for 1000 values for graphic representation of fast processes



Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Data_input_INT	Data entered in quick succession	–32768 to 32767
Parameters		
Activate_BOOL	Activates the function block	0/1
Rising_trigger_BOOL	If this parameter has a rising edge, the read process is restarted	0/1
Falling_trigger_BOOL	If this parameter has a falling edge, the read process is restarted	0/1
Interrupt_read_in_BOOL	Interruption of the read process	0/1
Ring_buffer_BOOL	Mode: recursive data read-in (ring buffer)	0/1
Read_out_start_BOOL	Start of read-out process	0/1
Read_out_time_per_value_ms_UINT	Read-out time per data value in ms	0 to 65535

Designation	Function	Value range
Outputs		
Data_output_INT	Data output slowly (in slow-motion)	–32768 to 32767
Monitor outputs		
Data_buffer_full_BOOL	Status: the data buffer is full	0/1
Data_read_out_finished_BOOL	Status: the data buffer has been read out	0/1
Read_in_cycle_UINT	Read-in cycle	0 to 1000
Read_in_time_ms_UINT	Duration of read-in process in ms	0 to 65535
Read_out_cycle_UINT	Read-out cycle	0 to 1000

Description

With this function block, 1000 values can be read in rapidly, for example one value per PLC cycle. These values can later be slowed down (read out in slow-motion). This allows fast processes, such as highly dynamic positioning processes, to be fully visually represented with conventional visualization tools. In combination with this function block, such tools can, for example, be used instead of an oscilloscope.

When the function block is activated, the read process starts when either a a rising edge or a falling edge is registered. In the ring buffer mode, 1000 values are read in and the first values then overwritten again by new values (ring buffer). The read-out time per value is specified in milliseconds. The monitor inputs indicate when the data buffer is full and when the data has been fully read out. In addition, the values “read-in cycle”, “read-in duration” and “read-out cycle” are available.

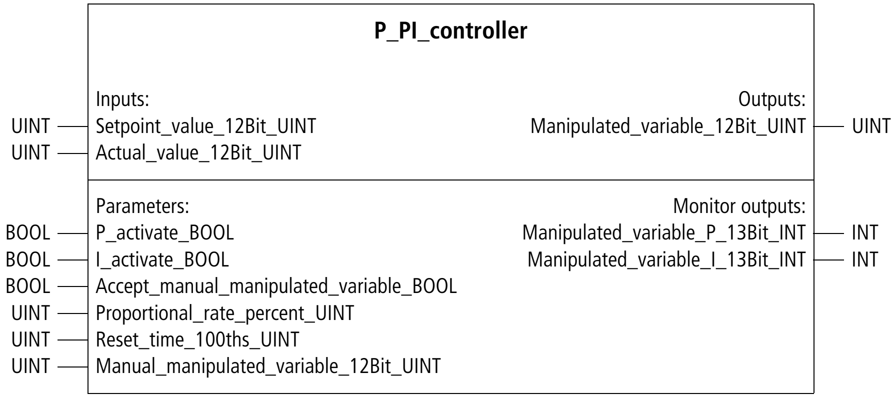
Example:
In the application example, the function block “P_Osc1000_oscilloscope” records the curve of the manipulated variable and reads out a curve value every 100 ms.

Application of function block "P_Osc1000_oscilloscope" in program "Oscillo"

```
PROGRAM Oscillo
VAR
      Osc1000_oscilloscope : P_Osc1000_oscilloscope ;
      Axis_01 : P_closed_loop_position_control ;
      MW12_INT : INT ;
END_VAR

CAL Osc1000_oscilloscope(
      Data_input_INT :=axis_01.Manipulated_variable_12bit_INT,
      Activate_BOOL :=1,
      Rising_trigger_BOOL :=0,
      Falling_trigger_BOOL :=axis_01.Positioning_job_finished_BOOL,
      Interrupt_read_in_BOOL :=0,
      Ring_buffer_BOOL :=0,
      Read_out_start_BOOL :=1,
      Read_out_time_per_value_ms_UINT :=100
      )
LD    Osc1000_Oscilloscope.Data_output_INT
MUL   2
ST    MW12_INT

END_PROGRAM
```

PI controller

P_PI_controller
PI controller with 12-bit inputs/outputs



Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Setpoint_value_12Bit_UINT	Setpoint value	0 to 4095
Actual_value_12Bit_UINT	Actual value	0 to 4095
Parameters		
P_activate_BOOL	Activates P component	0/1
I_activate_BOOL	Activates I component	0/1
Accept_manual_manipulated_variable_BOOL	"Smooth" acceptance of manual manipulated variable	0/1
Proportional_rate_percent_UINT	Proportional rate Kp [%]	0 to 65535
Reset_time_100ths_UINT	Reset time Tn [0.01 s]	0 to 65535
Manual_manipulated_variable_12Bit_UINT	Manual manipulated variable	0 to 4095
Outputs		
Manipulated_variable_12Bit_UINT	Manipulated variable (analog, 12 bit)	0 to 4095

Designation	Function	Value range
Monitor outputs		
Manipulated_variable_P_13Bit_INT	Manipulated variable P component	–4095 to 4095
Manipulated_variable_I_13Bit_INT	Manipulated variable I component	–4095 to 4095

Description

The PI controller is used for controlling the speed of rotating axes. The components of the controller can be activated (enabled) and deactivated separately with variables "P_activate_BOOL" and "I_activate_BOOL". Deactivating the I component resets the controller. The controller's parameters are set with the standardized variables "proportional rate" [%] and "reset time" [0.01 s].
The controller provides the analog output value "Manipulated_variable_12Bit_UINT". The PI components of the manipulated variable are used for (remote) diagnosis of control behaviour. The overall manipulated variable is created by adding the individual components.

Manual operation:
The controller can be operated by hand using the corresponding BOOL and UINT variables. If "Accept_manual_manip_var_BOOL" is "1", the controller outputs the value assigned to the variable "Manual_manipulated_variable_12Bit_UINT" to "Manipulated_variable_12Bit_UINT". If "Accept_manual_manip_var_BOOL" changes back to "0", the controller accepts the manual manipulated variable and continues constant (bumpless) control with this manipulated variable.

Example:
In application example "Zone1", a PI controller is called up with the following parameters:

- Proportional rate = 1.2
- Reset time = 3 s

Application of function block "P_PI_controller" in program "Zone1"

```
PROGRAM Zone1
VAR
      PI_controller_zone1 : P_PI_controller ;
      Setpoint_value_zone1 : UINT ;
      Actual_value_zone1 : UINT ;
      Enable_PI_controller: BOOL ;
      Manual_operation_zone1 : BOOL ;
      Manual_manipulated_variable_zone1 : UINT :=1000;
      Manipulated_variable_zone1 : UINT;
END_VAR

CAL PI_controller_zone1(
      Setpoint_value_12Bit_UINT :=Setpoint_value_zone1,
      Actual_value_12Bit_UINT :=Actual_value_zone1,
      P_activate_BOOL :=Enable_PI_controller,
      I_activate_BOOL :=Enable_PI_controller,
      Accept_manual_manip_var_BOOL :=Manual_operation_zone1,
      Proportional_rate_percent_UINT :=120,
      Reset_time_100ths_UINT :=300,
      Manual_manipulated_variable_12Bit_UINT :=Manual_manipulated_variable_zone1,
      Manipulated_variable_12Bit_UINT=>Manipulated_variable_zone1
      )

END_PROGRAM
```

P_ZSFB03_special_FB
PI controller with 12-bit integer inputs/outputs and manipulated variable limitation

P_ZSFB03_special_FB

Type	Inputs:	Outputs:	Type
INT	Setpoint value_12Bit_INT	Manipulated_variable_12bit_INT	INT
INT	Actual_value_12Bit_INT		
	Parameters:	**Monitor outputs:**	
BOOL	P_activate_BOOL	Manipulated_variable_P_13Bit_INT	INT
BOOL	I_activate_BOOL	Manipulated_variable_I_13Bit_INT	INT
BOOL	Accept_manual_manipulated_variable_BOOL		
UINT	Proportional_rate_percent_UINT		
UINT	Reset_time_100ths_UINT		
INT	Manual_manipulated_variable_12Bit_INT		
INT	Lower_limit_manipulated_variable_INT		
INT	Upper_limit_manipulated_variable_INT		

Function block prototype

Meaning of the operands

Designation	Function	Value range
Inputs		
Setpoint value_12Bit_INT	Setpoint value	–4095 to 4095
Actual_value_12Bit_INT	Actual value	–4095 to 4095
Parameters		
P_activate_BOOL	Activates P component	0/1
I_activate_BOOL	Activates I component	0/1
Accept_manual_manipulated_variable_BOOL	"Smooth" acceptance of manual manipulated variable	0/1
Proportional_rate_percent_UINT	Proportional rate Kp [%]	0 to 65535
Reset_time_100ths_UINT	Reset time Tn [0.01 s]	0 to 65535
Manual_manipulated_variable_12Bit_INT	Manual manipulated variable	–4095 to 4095
Lower_limit_manipulated_variable_INT	Lower limit of manipulated variable	–4095 to 4095
Upper_limit_manipulated_variable_INT	Upper limit of manipulated variable	–4095 to 4095
Outputs		
Manipulated_variable_12bit_INT	Manipulated variable (analog, 12 bit)	–4095 to 4095
Monitor outputs		
Manipulated_variable_P_13Bit_INT	Manipulated variable P component	–4095 to 4095
Manipulated_variable_I_13Bit_INT	Manipulated variable I component	–4095 to 4095

Description

See function block "P_PI_controller". The PI controller is suitable for controlling the speed of rotating axes, in particular if the direction of rotation changes repeatedly. The setpoint and actual values can therefore be entered as integers and the manipulated variable is also output as an integer. In addition, a lower and an upper limit value for the manipulated variable can be defined.

Index