Function Manual 04/2014 Edition

SINAMICS/SIMOTION

Description of the standard DCC blocks

Motion Control

SIEMENS

SIEMENS

SINAMICS/SIMOTION

Motion Control SINAMICS/SIMOTION Description of the Standard DCC Blocks

Function Manual

6SL3097-4AQ00-0BP3

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ DANGER

indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ WARNING

indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ CAUTION

indicates that minor personal injury can result if proper precautions are not taken.

NOTICE

indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ WARNING

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

Order number: 6SL3097-4AQ00-0BP3

Preface

SIMOTION documentation

An overview of the SIMOTION documentation can be found in the SIMOTION Documentation Overview document.

This documentation is included as electronic documentation in the scope of delivery of SIMOTION SCOUT. It comprises ten documentation packages.

The following documentation packages are available for SIMOTION V4.4:

- SIMOTION Engineering System Handling
- SIMOTION System and Function Descriptions
- SIMOTION Service and Diagnostics
- SIMOTION IT
- SIMOTION Programming
- SIMOTION Programming - References
- SIMOTION C
- SIMOTION P
- SIMOTION D
- SIMOTION Supplementary Documentation

SINAMICS documentation

The SINAMICS documentation is organized into two parts:

- General documentation/catalogs
- Manufacturer/service documentation

A current overview of the documentation can be found on the Internet in the available languages:

http://www.siemens.com/motioncontrol

Select the menu items "Support" --> "Technical Documentation" --> "Overview of Publications".

The Internet version of DOConCD (DOConWEB) is available on the Internet:

http://www.automation.siemens.com/doconweb

Information on the range of training courses and FAQs (Frequently Asked Questions) are available on the Internet:

http://www.siemens.com/motioncontrol

Select the menu command "Support".

Further documentation for the DCC editor

- SINAMICS/SIMOTION DCC Editor Description

Additional information

Using the following link, you can find information on the topics:

- Ordering documentation / overview of publications
- Additional links to download documents
- Using documentation online (find and search manuals/information)

http://www.siemens.com/motioncontrol/docu

Please send any questions about the technical documentation (e.g. suggestions for improvement, corrections) to the following e-mail address:

docu.motioncontrol@siemens.com

My Documentation Manager

Click the following link for information on how to compile documentation individually on the basis of Siemens content and how to adapt it for the purpose of your own machine documentation:

http://www.siemens.com/mdm

Training

Click the following link for information on SITRAIN - Siemens training courses for automation products, systems and solutions:

http://www.siemens.com/sitrain

FAQs

Frequently Asked Questions can be found in SIMOTION Utilities & Applications, which are included in the scope of delivery of SIMOTION SCOUT, and in the Service & Support pages in Product Support:

http://support.automation.siemens.com

Technical support

Country-specific telephone numbers for technical support are provided on the Internet under Contact:

http://www.siemens.com/automation/service&support

Table of contents

Fundamental safety instructions 1

1.1 General safety instructions

⚠ WARNING

Risk of death if the safety instructions and remaining risks are not carefully observed

If the safety instructions and residual risks are not observed in the associated hardware documentation, accidents involving severe injuries or death can occur.

- Observe the safety instructions given in the hardware documentation.
- Consider the residual risks for the risk evaluation.

⚠ WARNING

Danger to life or malfunctions of the machine as a result of incorrect or changed parameterization

As a result of incorrect or changed parameterization, machines can malfunction, which in turn can lead to injuries or death.

- Protect the parameterization (parameter assignments) against unauthorized access.
- Respond to possible malfunctions by applying suitable measures (e.g. EMERGENCY STOP or EMERGENCY OFF).

1.2 Industrial security

Note

Industrial security

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens' products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. For more information about industrial security, visit http://www.siemens.com/industrialsecurity.

To stay informed about product updates as they occur, sign up for a product-specific newsletter. For more information, visit http://support.automation.siemens.com

⚠ WARNING

Danger as a result of unsafe operating states resulting from software manipulation

Software manipulation (e.g. by viruses, Trojan horses, malware, worms) can cause unsafe operating states to develop in your installation which can lead to death, severe injuries and/or material damage.

- Keep the software up to date.
  Information and newsletters can be found at:
  http://support.automation.siemens.com
- Incorporate the automation and drive components into a state-of-the-art, integrated industrial security concept for the installation or machine.
  For more detailed information, go to:
  http://www.siemens.com/industrialsecurity
- Make sure that you include all installed products into the integrated industrial security concept.

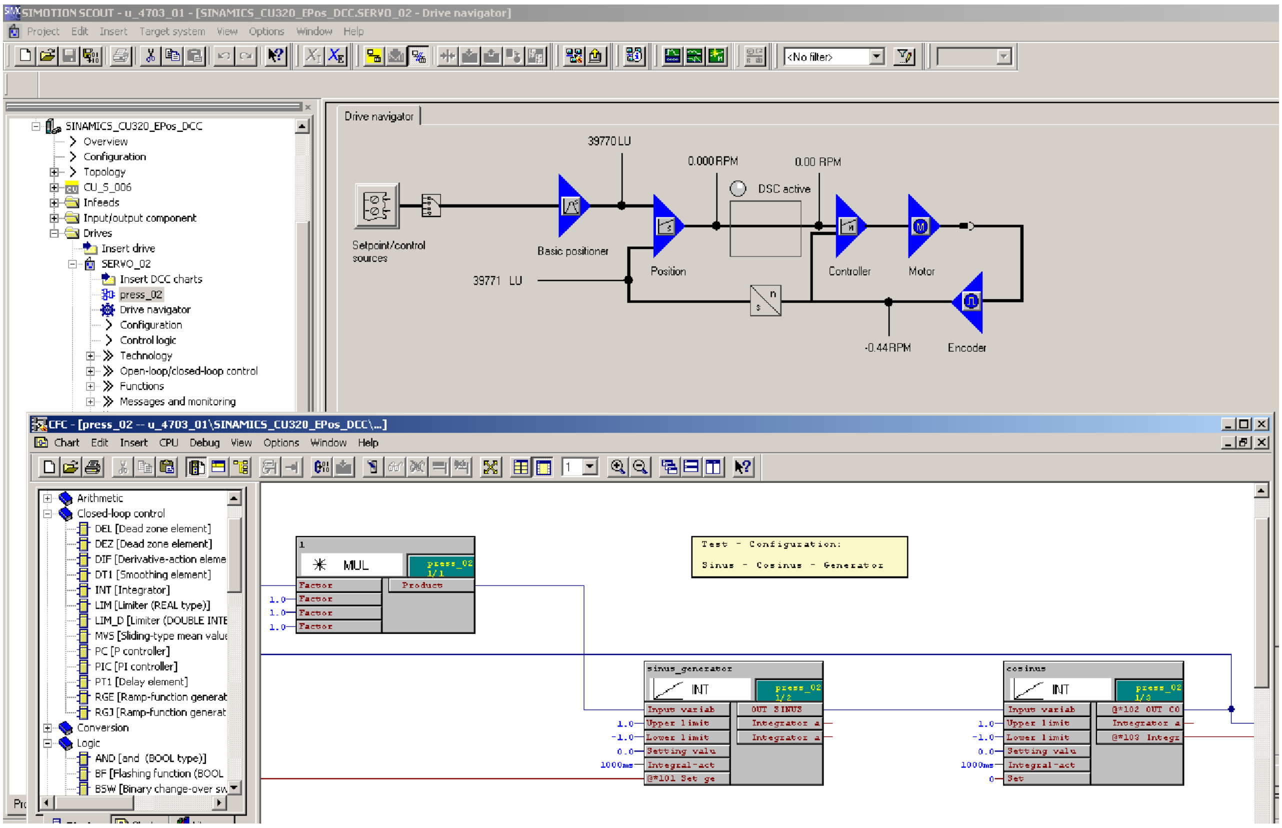
Introduction

2

2.1 Introduction to the Drive Control Chart (DCC)

Drive Control Chart (DCC) for SINAMICS and SIMOTION means graphic configuration and expansion of the device functionality by means of freely available control, calculation and logic blocks

Drive Control Chart (DCC) expands the facility for the simplest possible configuring of technological functions both for the SIMOTION motion control system and the SINAMICS drive system. This opens up a new dimension for users for adapting the specified systems to the specific functions of their machines. DCC has no restriction with regard to the number of usable functions; this is only limited by the performance capability of the target platform.



The user-friendly **DCC editor** enables easy graphic configuration and a clear representation of control loop structures as well as a high degree of reusability of existing charts.

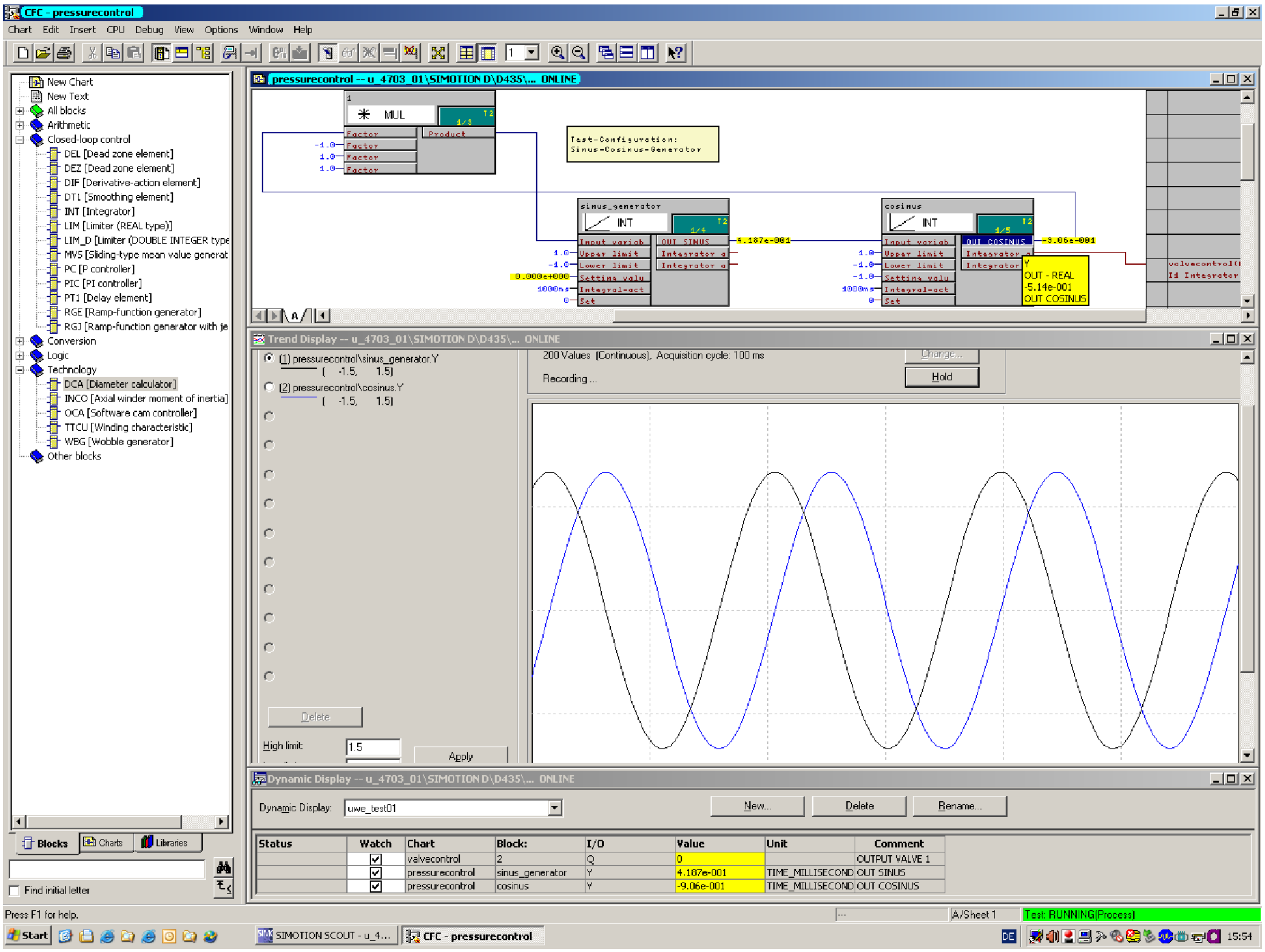
The open-loop and closed-loop control functionality is defined by using multi-instance-capable blocks (**Drive Control Blocks, DCBs**) from a pre-defined library (**DCB library**) that are selected and graphically linked by dragging and dropping. Test and diagnostic functions allow verification of program behavior or the identification of causes in the event of errors.

The block library contains a large selection of control, arithmetic and logic blocks as well as extensive open-loop and closed-loop control functions.

All commonly used logic functions are available for selection (AND, XOR, On/Off delay, RS flip-flop, counters, etc.) for the logic operation, evaluation and acquisition of binary signals. Numerous calculation functions, such as summation, division and minimum/maximum evaluation are available for monitoring and evaluating numeric variables. In addition to the drive control, axial winder functions, PI controllers, ramp-function generators or sweep generators can be configured simply and without problem.

Almost unlimited programming of control structures is possible in conjunction with the SIMOTION motion control system. These can then be combined with other program sections to form an overall program.

Drive Control Chart for SINAMICS drives also provides a convenient basis for resolving drive-level open-loop and closed-loop control tasks directly in the converter. This results in further adaptability of SINAMICS for the tasks set. On-site processing in the drive supports modular machine concepts and results in increased overall machine performance.



Dynamic Display: uwe_test01

Status	Watch	Chart	Block:	I/O	Value	Unit	Comment
	☑	valvecontrol	2	Q	0		OUTPUT VALVE 1
	☑	pressurecontrol	sinus_generator	Y	4.187e-001	TIME_MILLISECOND	OUT SINUS
	☑	pressurecontrol	cosinus	Y	-9.06e-001	TIME_MILLISECOND	OUT COSINUS

2.2 Libraries

Blocks are located in libraries that are imported as technology packages in the DCC editor.

There are two different libraries:

1. The *SIMOTION library* contains the SIMOTION blocks identified in this document.
2. The *SINAMICS library* contains the SINAMICS blocks identified in this document.

To find out which of the blocks described here are available within SIMOTION and/or SINAMICS, you can use both the overview in Appendix A3 and the sections on block descriptions.

Compatibility

SIMOTION

The standard library is given the version dcblibV4_1_simotion4_4. You can use this library for SIMOTION devices as of V4.4.

For target devices of version SIMOTION V4.3, the dcblibV4_0_simotion4_3 is installed in SIMOTION SCOUT by default.

You can use the dcblibV4_0_simotion4_3 library also for SIMOTION V4.4 devices.

For target devices of version SIMOTION V4.2, the dcblibV3_0_simotion4_2 is installed in SIMOTION SCOUT by default.You can use the dcblibV3_0_simotion4_2 library also for SIMOTION V4.3 and V4.4 devices.

For target devices of version SIMOTION V4.1, the dcblibV2_0_simotion4_1_5 is installed in SIMOTION SCOUT by default.You can use the dcblibV2_0_simotion4_1_5 library also for SIMOTION V4.2, V4.3 and V4.4 devices.

SINAMICS

With SINAMICS V4.6, the following standard libraries are executable:

- SINAMICS V4.6 (dcblibV3_0_sinamics4_6)
- SINAMICS V4.5 (dcblibV3_0_sinamics4_5)
- SINAMICS V4.4 (dcblibV3_0_sinamics4_4)
- SINAMICS V4.3 (dcblibV2_0_sinamics4_3)

With SINAMICS V4.5, the following standard libraries are executable:

- SINAMICS V4.5 (dcblibV3_0_sinamics4_5)
- SINAMICS V4.4 (dcblibV3_0_sinamics4_4)
- SINAMICS V4.3 (dcblibV2_0_sinamics4_3)

With SINAMICS V4.4, the following standard libraries are executable:

- SINAMICS V4.4 (dcblibV3_0_sinamics4_4)
- SINAMICS V4.3 (dcblibV2_0_sinamics4_3)

With SINAMICS V4.3.x, the following standard libraries are executable:

- SINAMICS V4.3 (dcblibV2_0_sinamics4_3)

With SINAMICS V2.6.x, the following standard libraries are executable:

- SINAMICS V2.6 (dcblibV2_0_sinamics2_6)

With SINAMICS V2.5.SP1, the following standard libraries are executable:

- SINAMICS V2.5 (dcblibV2_0_sinamics2_5_1)

2.3 Nomenclature of the blocks

A block is displayed as follows:

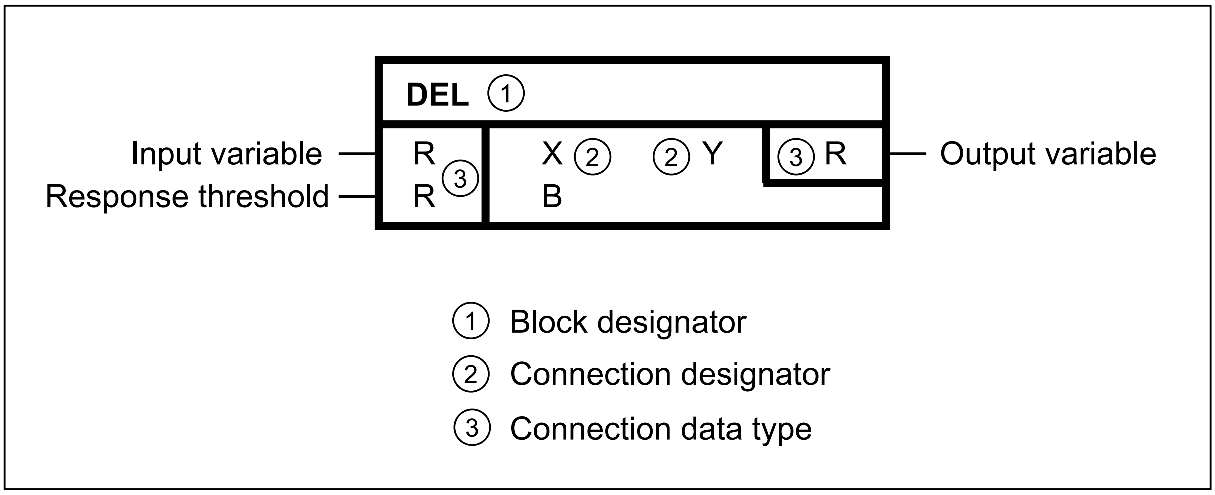


Figure 2-1 Nomenclature

It is identified using the following attributes:

Block designator

Each data type has its own block type. To simplify differentiation between the blocks for various data types with the same functionality, these are provided with a postfix corresponding to the data type, whereby postfix is not usually used for the Real and Bool data types (e.g. MUL_I: Integer-type multiplier, MUL: Real-type multiplier).The following table lists commonly used extensions:

Table 2-1 Block designator

Postfix for block designator	Data type of the input/output variable
_I	Integer
_D	Double_Integer
_W	Word
_R	Real (optional)
_B	Bool (optional)
_SI	Short Integer
_M	Modulo
_BY	Byte
_UI	Unsigned Integer
_US	Unsigned Short Integer
_UD	Unsigned Double Integer

Postfix for block designator	Data type of the input/output variable
_DW	Double Word
_LR	Long Real

Connection designator

- To identify a field from the input or output variable, the designator is extended by an index beginning with 1 (e.g. X1, X2, X3, etc.).
- With a generic number of inputs (e.g. ADD), the connection name is indexed beginning with 1 (e.g. X1, X2, X3, etc.).

The following table shows an overview of common connection designators:

Table 2-2 Connection designator

Connection designator	Use
X, X1, X2...	Numeric input variable
Y, Y1, Y2...	Numeric output variable
I, I1, I2...	Binary input variable
Q, Q1, Q2...	Binary output variable
IS	Bit string input (Word)
QS	Bit string output (Word)

If further inputs and outputs are used along with the primary input and output variables (e.g. limit values, time data, substitute values, status displays), the designators from the pool of the primary input/output variables are not used. The following table shows the preferred designators for secondary variables:

Connection designator	Use
LU	Input: Upper limit value
LL	Input: Lower limit value.
SV	Input: Setting value
S	Input: Setting the setting value
R	Input: Resetting the setting value
QU	Output: Upper limit reached
QL	Output: Lower limit reached
QF	Output: Error indicator
QE	Output Y equals input X
QN	Inverted binary variable

Connection data type

The abbreviated designators of the data types are listed in the following table:

Abbreviation	Bit width	Data type in line with IEC 61131-3	Description
BO	1	BOOL	BOOLEAN
BY	8	BYTE	Bit string, Unsigned Integer
SI	8	SINT	Signed Short Integer
DI	32	DINT	Signed Double Integer
DW	32	DWORD	Bit string, Unsigned Integer
I	16	INT	Signed Integer
R	32	REAL	Floating Point Single Precision in line with IEEE 754
LR	64	LREAL	Floating Point Double Precision in line with IEEE 754
T	32	SDTIME	Floating Point Single Precision in line with IEEE754
W	16	WORD	Bit string, Unsigned Integer
AID	32	-	Alarm ID

2.4 Block connections

Block connections display the interface of the DCBs, via which interconnection between the blocks can be performed. A distinction is made between

- Block output
- Block input

here, and these have the following properties:

- Inputs are positioned on the left of the block and are the target of an interconnection
- Outputs are positioned on the right of the block and are the source of an interconnection

2.5 Byte ordering

When interconnecting the blocks, the byte ordering of the data does not have to be taken into consideration. During data type conversions and arithmetic operations, the byte ordering of the target system is implicitly taken into consideration. Any byte swapping required for handling data beyond the system boundaries is carried out by the system (e.g. byte swapping may have to be carried out in Big Endian format before transferring data via PROFIBUS).

2.6 Direct interconnection of different data types

When interconnecting blocks, the target and source must be of the same data type. If the data types are different, there are special conversion blocks available which allow the data type to be converted.

The following permissible implicit conversions are an exception. The table below lists the permissible conversions.

The following data types, which can be interconnected without a conversion block, are another exception. In this case, the binary value of the output variable is transferred unchanged as the input variable.

Table 2-3 Conversions

Input	Output	Description
WORD	INT	Interconnection of a word variable to an integer variable
INT	WORD	Interconnection of an integer variable to a word variable
DWORD	DINT	Interconnection of a double word variable to a double integer variable
DINT	DWORD	Interconnection of a double integer variable to a double word variable
BYTE	SINT	Interconnection of a byte variable to a short integer variable
SINT	BYTE	Interconnection of a short integer variable to a byte variable
USINT	BYTE	Interconnection of an unsigned short integer variable to a byte variable
BYTE	USINT	Interconnection of a byte variable to an unsigned short integer variable
USINT	SINT	Interconnection of an unsigned short integer variable to a short integer variable
SINT	USINT	Interconnection of a short integer variable to an unsigned short integer variable
UINT	WORD	Interconnection of an unsigned integer variable to a word variable
WORD	UINT	Interconnection of a word variable to an unsigned integer variable
UINT	INT	Interconnection of an unsigned integer variable to an integer variable
INT	UINT	Interconnection of an integer variable to an unsigned integer variable
UDINT	DWORD	Interconnection of an unsigned double integer variable to a double word variable
DWORD	UDINT	Interconnection of a double word variable to an unsigned double integer variable
UDINT	DINT	Interconnection of an unsigned double integer variable to a double integer variable
DINT	UDINT	Interconnection of a double integer variable to an unsigned double integer variable
SDTIME	REAL	Interconnection of an SDTime variable to a real variable

2.7 Initialization of the blocks

Initialization determines the starting condition of the block. It is carried out by the system before the cyclical processing[1)] of the block. The sequence for initializing the individual blocks is implemented in line with the configured priority and process sequence. At the time of initialization, the configured interconnections and constants for a block are already active. At this point, the values from the interconnection source are already available in a block. Should a block behave in a special way during initialization, this is described in the respective block description under "Initialization". In the case of initialization, the blocks must be assigned in a time slice (SINAMICS) or to a task (SIMOTION).

1) As of SIMOTION 4.1 SP2, initialization is performed during a STOP/RUN transition (SIMOTION) or during the transition to cyclical operation (SINAMICS).

2.8 Implementing complex functions in a sample configuration

Sample configurations are available for the "convenient ramp-function generator" and "technology controller". These are based on the "free blocks" of the SIMOVERT MASTERDRIVES frequency converter series.

The functionality of the convenient ramp-function generator is implemented by interconnecting individual DCBs. They are provided in the form of a sample configuration.

Notes relating to the technology controller

- The smoothing filters cannot be deactivated using the time constant T = 0, as the time constant is limited to the sampling time of the block. The filters must be deactivated explicitly by a signal. The corresponding binary input must be provided in the sample configuration.
- The D component cannot be deactivated using the delay time Tv = 0. This must be deactivated explicitly by a binary signal. The corresponding binary input must be provided in the sample configuration.
- The I component cannot be deactivated using Tn = 0. For this, the I component of PIC must be reset explicitly using SV = 0 and S = 1.

Note

Here, you should now also use the advantage of DCC and configure/transfer only the blocks that are required, i.e. not fall back on the sample configurations from the outset and start with the basic blocks, such as RGJ (ramp-function generator with rounding) or PIC (PI controller) and expand, if necessary.

Import sample configuration

The sample configuration is archived in the form of exported DCC charts as of SCOUT/ STARTER version V4.3.

SIMOTION

In order to import the DCC chart in SCOUT, a SIMOTION device needs to be present in the project. In the Programs folder, select the path of the exported DCC chart via the context menu Expert -> Import object. The files can usually be found under the following path:

C:\Program Files\Siemens\Step7\Examples\dcc\SIMOTION

SINAMICS

To import the DCC chart in SCOUT/STARTER, an S120 drive unit with a drive object (DO) must be present in the project. On the drive object, select the path of the exported DCC chart via the context menu Expert -> Import object. The files can usually be found under the following path:

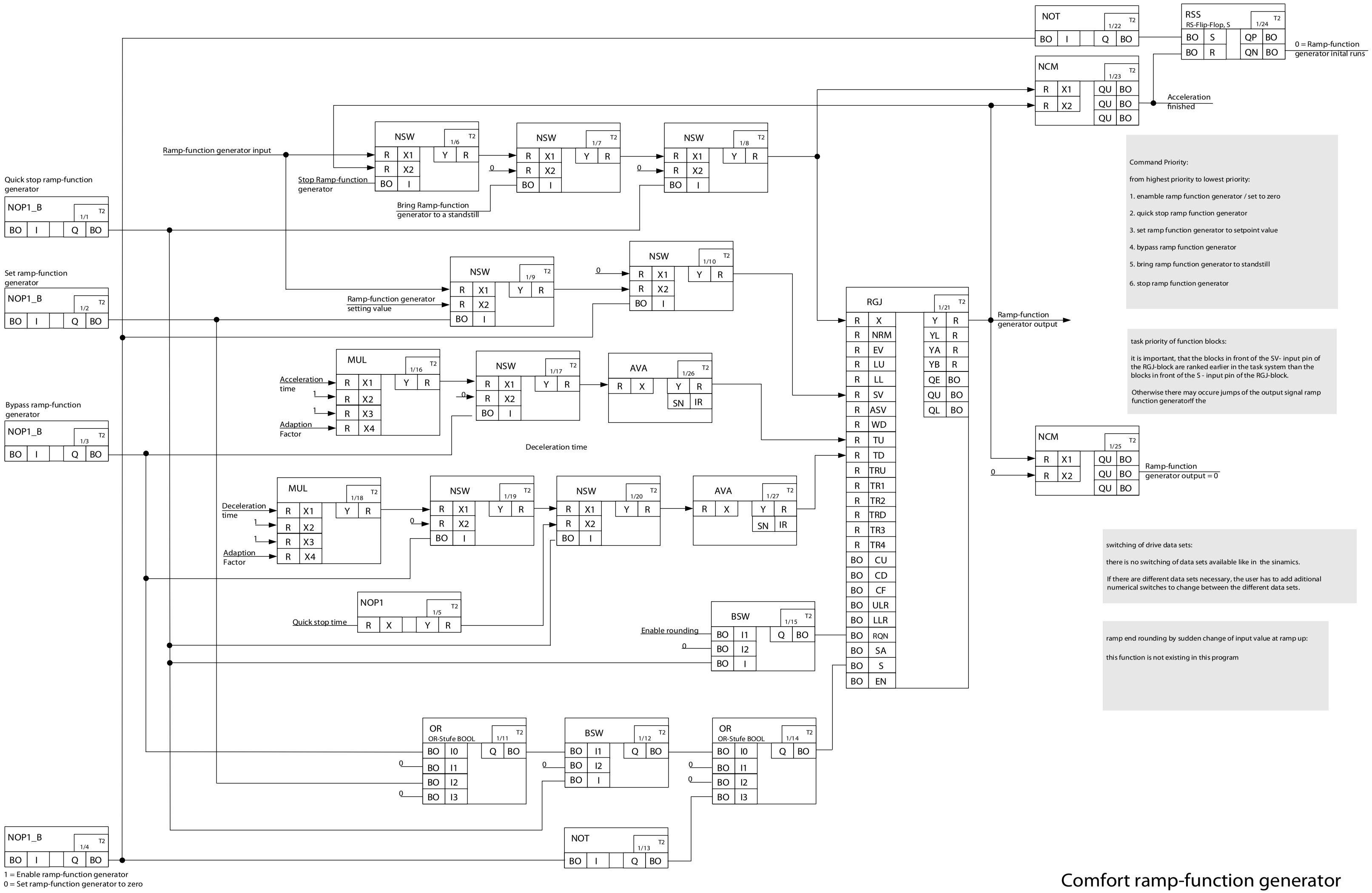
C:\Program Files\Siemens\Step7\Examples\dcc\SINAMICS

Up to SCOUT/STARTER version V4.3, the sample configuration was archived as a project export under the following path:

C:\Program Files\Siemens\Step7\Examples\dcc\Examples_CRGE_TCLR.xml

Note

The sample project has been created on the basis of a SCOUT project and thus contains both the configurations for SINAMICS_Integrated and SINAMICS stand-alone/CU320. When importing the sample project with STARTER (stand-alone), the SIMOTION components are naturally rejected; the CU320 components, however, continue to be accurately imported and are reproducible.


Quick stop ramp-function generator
NOP1_B 1/1 T2
Set ramp-function generator
NOP1_B 1/2 T2
Bypass ramp-function generator
NOP1_B 1/3 T2
NOP1_B 1/4 T2
1 = Enable ramp-function generator
0 = Set ramp-function generator to zero
Ramp-function generator input
Stop Ramp-function generator
Bring Ramp-function generator to a standstill
NSW 1/6 T2
NSW 1/7 T2
NSW 1/8 T2
NSW 1/9 T2
NSW 1/10 T2
Ramp-function generator setting value
Acceleration time
Adaption Factor
MUL 1/16 T2
NSW 1/17 T2
AVA 1/26 T2
Deceleration time
Deceleration time
Adaption Factor
MUL 1/18 T2
NSW 1/19 T2
NSW 1/20 T2
AVA 1/27 T2
Quick stop time
NOP1 1/5 T2
Enable rounding
BSW 1/15 T2
OR OR-Stufe BOOL 1/11 T2
BSW 1/12 T2
OR OR-Stufe BOOL 1/14 T2
NOT 1/13 T2
RGJ 1/21 T2
X NRM EV LU LL SV ASV WD TU TD TRU TR1 TR2 TRD TR3 TR4 CU CD CF ULR LLR RQN SA S EN
Y YL YA YB QE QU QL
Ramp-function generator output
NOT 1/22 T2
RSS RS-Flip-Flop, S 1/24 T2
0 = Ramp-function generator inital runs
NCM 1/23 T2
Acceleration finished
NCM 1/25 T2
Ramp-function generator output = 0

Command Priority:

from highest priority to lowest priority:

1. enamble ramp function generator / set to zero

2. quick stop ramp function generator

3. set ramp function generator to setpoint value

4. bypass ramp function generator

5. bring ramp function generator to standstill

6. stop ramp function generator

task priority of function blocks:

it is important, that the blocks in front of the SV- input pin of the RGJ-block are ranked earlier in the task system than the blocks in front of the S - input pin of the RGJ-block.

Otherwise there may occure jumps of the output signal ramp function generator!f the

switching of drive data sets:

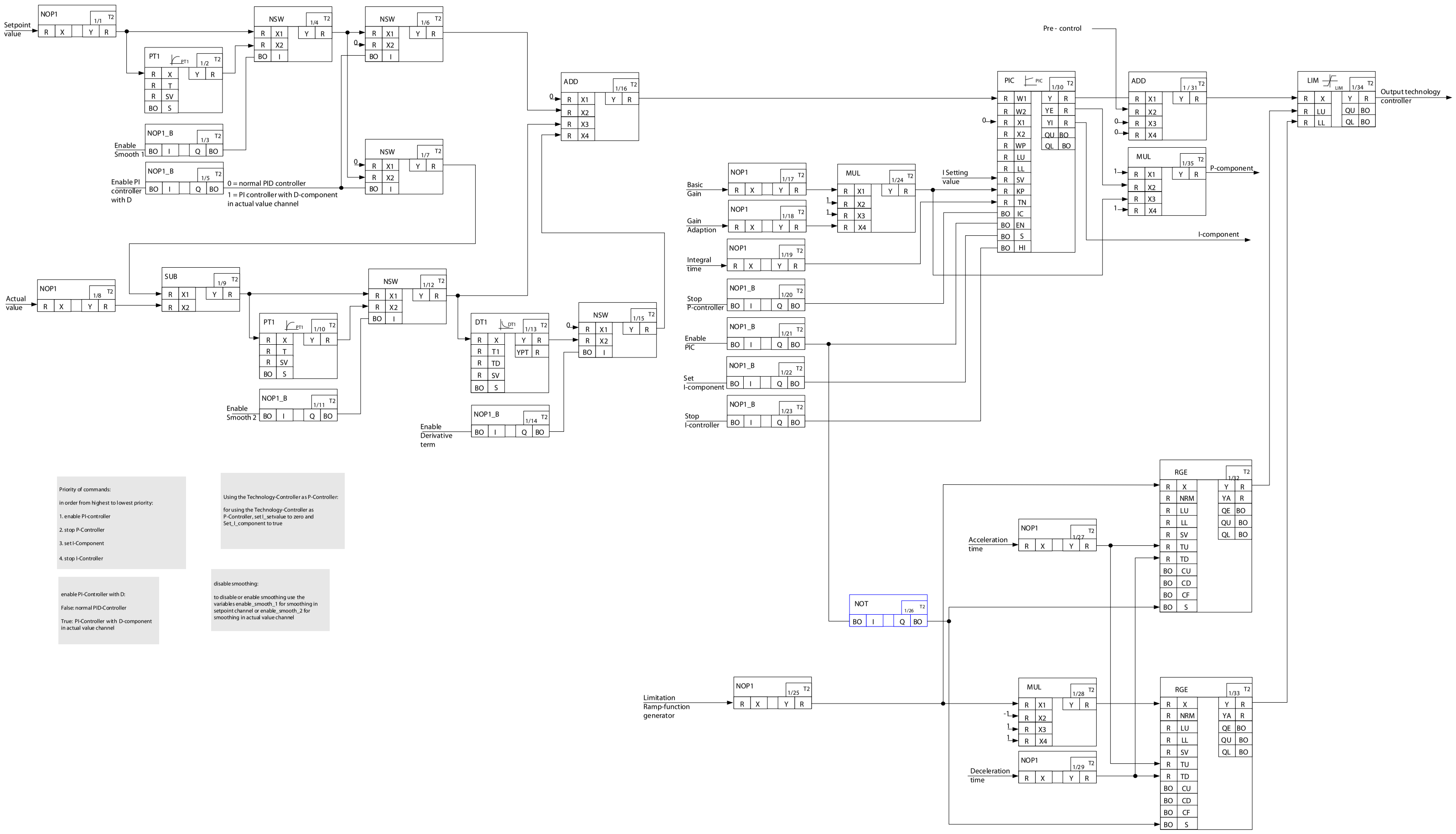
there is no switching of data sets available like in the sinamics.

If there are different data sets necessary, the user has to add aditional numerical switches to change between the different data sets.

ramp end rounding by sudden change of input value at ramp up:

this function is not existing in this program

Comfort ramp-function generator



Priority of commands:

in order from highest to lowest priority:

1. enable PI-controller

2. stop P-Controller

3. set I-Component

4. stop I-Controller

Using the Technology-Controller as P-Controller:

for using the Technology-Controller as P-Controller, set I_setvalue to zero and Set_I_component to true

enable PI-Controller with D:

False: normal PID-Controller

True: PI-Controller with D-component in actual value channel

disable smoothing:

to disable or enable smoothing use the variables enable_smooth_1 for smoothing in setpoint channel or enable_smooth_2 for smoothing in actual value channel

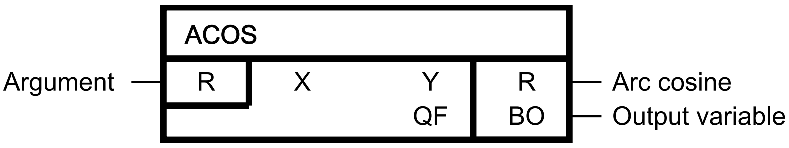
Technology Controller

Arithmetic 3

3.1 ACOS Arc cosine function

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

Determination of the arc cosine value for an argument

Operation mode

The block determines the associated arc cosine value in radian measure for an argument to be entered at input X and outputs the result at output Y.

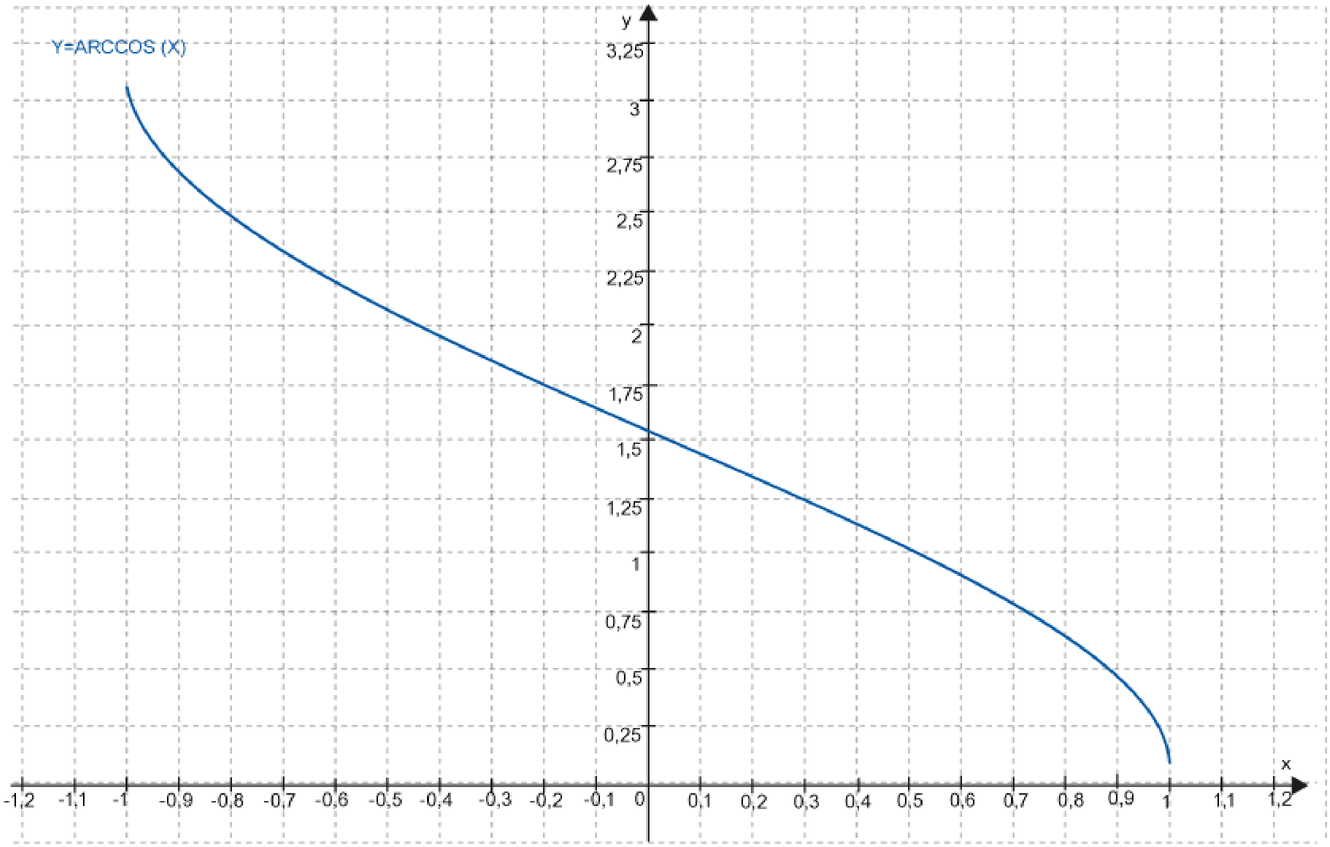
Y = arccos X

Permissible input range: -1.0 <= X <= +1.0

Output range: 0.0 <= Y <= π

If the argument lies outside of the permitted input range, output Y is limited to π (when X < -1.0) or 0.0 (when X > +1.0) and binary output QF = 1 is set at the same time.

Transfer function



Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Argument	0.0	REAL	
Y	Arc cosine	π/2	REAL	
QF	Output variable	0	0/1	

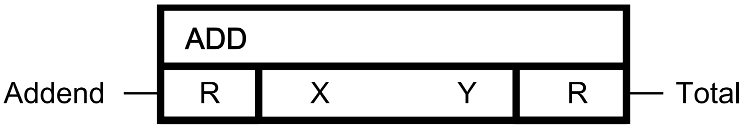
Project data

Can be loaded online	Yes
Special characteristics	-

3.2 ADD Adder (REAL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Adder with up to four inputs of the REAL type

Operation mode

The block adds the values entered at the X inputs, taking account of the sign. The result, limited to the range of -3.402823 E38 to 3.402823 E38, is output at output Y.

Algorithm:

Y = X1 + X2 + X3 + X4

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Addend	0.0	REAL	
Y	Total	0.0	REAL	

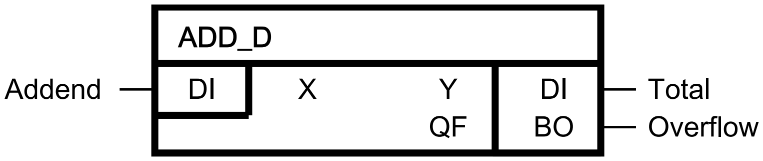
Project data

Can be loaded online	Yes
Can be configured in	Cyclic tasks
Special characteristics	X comprises up to four inputs (X1 to X4)

3.3 ADD_D Adder (double integer type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Adder with up to four inputs of the double integer type

Operation mode

The block adds the values entered at the X inputs, taking account of the sign. The result, limited to the range of -2147483648 (2^{31}) to +2147483647 (2^{31}-1), is output at output Y.

Algorithm:

Y = X1 + X2 + X3 + X4

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Addend	0	DINT	
Y	Total	0	DINT	
QF	Overflow	0	0/1	

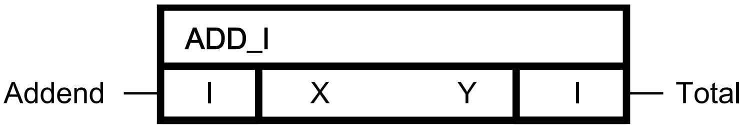
Project data

Can be loaded online	**Yes**
Special characteristics	X comprises up to four inputs (X1 to X4)

3.4 ADD_I Adder (integer type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Adder with up to four inputs of the integer type

Operation mode

The block adds the values entered at the X inputs, taking account of the sign. The result, limited to the range of -32768 to +32767, is output at output Y.

Algorithm:

Y = X1 + X2 + X3 + X4

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Addend	0	INT	
Y	Total	0	INT	

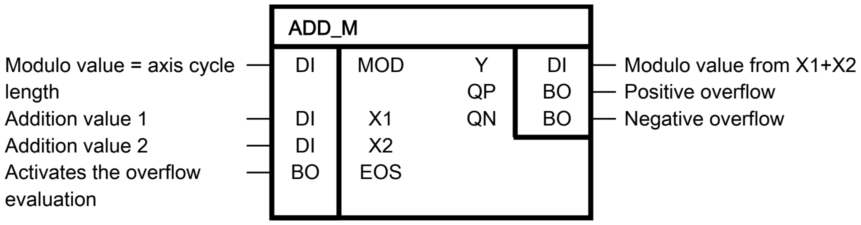
Project data

Can be loaded online	Yes
Special characteristics	X comprises up to four inputs (X1 to X4)

3.5 ADD_M Modulo adder for addition in correct axis cycle

☑ SIMOTION ☑ SINAMICS

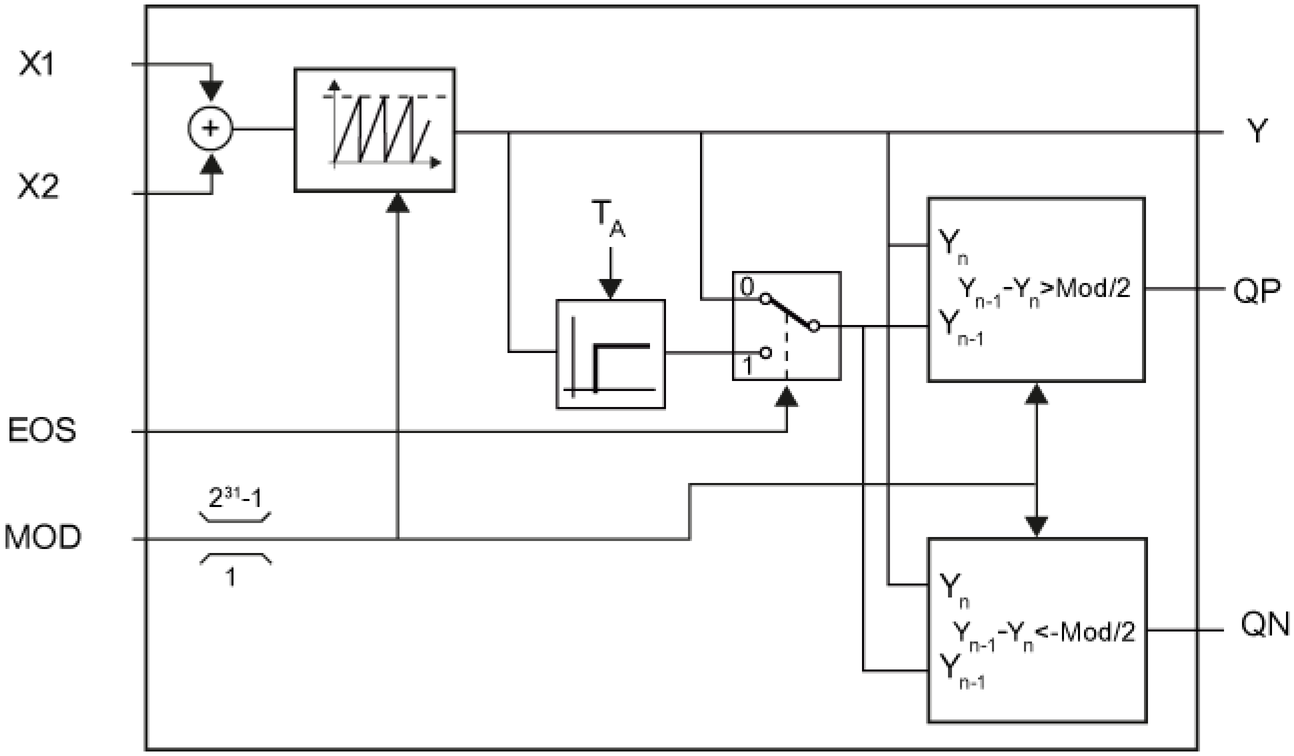
Symbol



Short description

The ADD_M block is used to add position values. It can be used to "add up" offsets for position setpoints or for dead time compensation in the real master.

Block diagram



Operation mode

The block adds the input values X1 and X2. A modulo value, which is limited to 1..2^{31}-1, can be specified at the MOD input, and applied to the sum of X1 and X2. Thus, the result Y of the modulo operation is always in the band from 0 to MOD.

The EOS input can be used to activate an overflow evaluation. When EOS = 1Positive overflow: QP = Y_{n-1}-Y_n> MOD/2 Negative overflow: QN = Y_{n-1} – Y_n < -MOD/2

When EOS = 0:QP = 0 QN = 0

In this way, it is possible to deactivate the overflow evaluation when setting offsets.When changing the modulo value, the overflow evaluation is deactivated for one cycle.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
MOD	Modulo value = axis cycle length	1	DINT	
X1	Addition value 1	0	DINT	
X2	Addition value 2	0	DINT	
EOS	Activates the overflow evaluation	0	0/1	
Y	Modulo value from X1+X2	0	DINT	
QP	Positive overflow	0	0/1	
QN	Negative overflow	0	0/1	

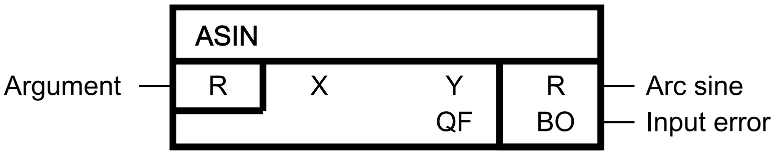
Project data

Can be inserted online	Yes
Special characteristics	-

3.6 ASIN Arc sine function

☑ SIMOTION □ SINAMICS

Symbol



Short description

Determination of the arc sine value for an argument

Operation mode

The block determines the associated arc sine value in radian measure for an argument to be entered at input X and outputs the result at output Y.

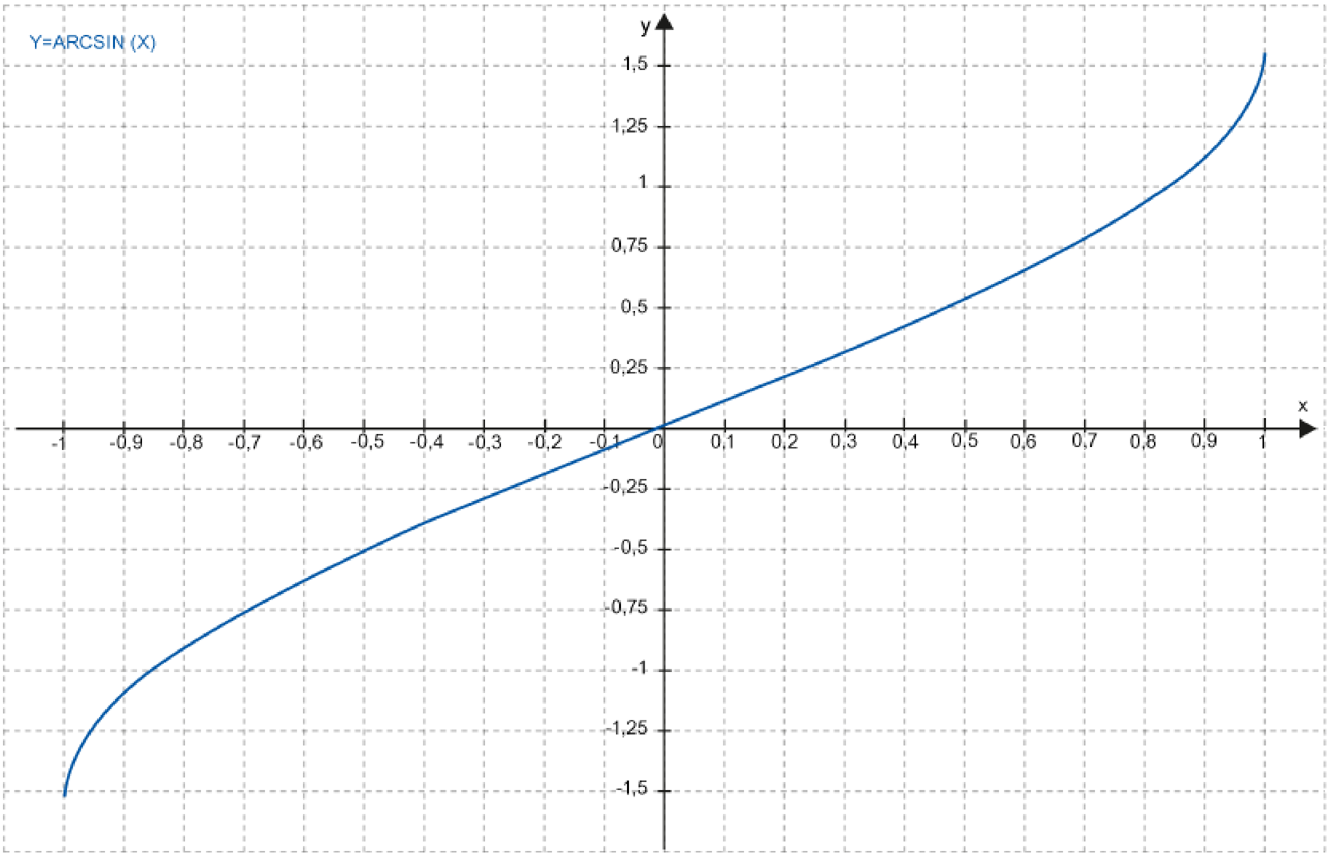
Y = arcsin X

Permissible input range: -1.0 <= X <= +1.0

Output range: -π/2 <= Y <= π/2

If the argument lies outside of the permitted input range of |X| <= 1.0, output Y is limited to -π/2 (when X < -1.0) or π/2 (when X > +1.0) and binary output QF = 1 is set at the same time.

XY diagram



Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Argument	0.0	REAL	
Y	Arc sine	0.0	REAL	
QF	Input error	0	0/1	

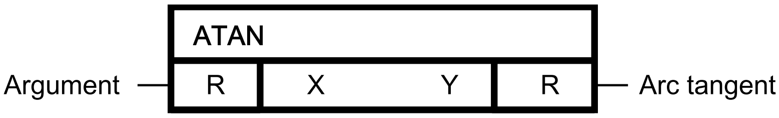
Project data

Can be inserted online	Yes
Special characteristics	-

3.7 ATAN Arc tangent function

☑ SIMOTION □ SINAMICS

Symbol



Short description

Determination of the arc tangent value for an argument

Operation mode

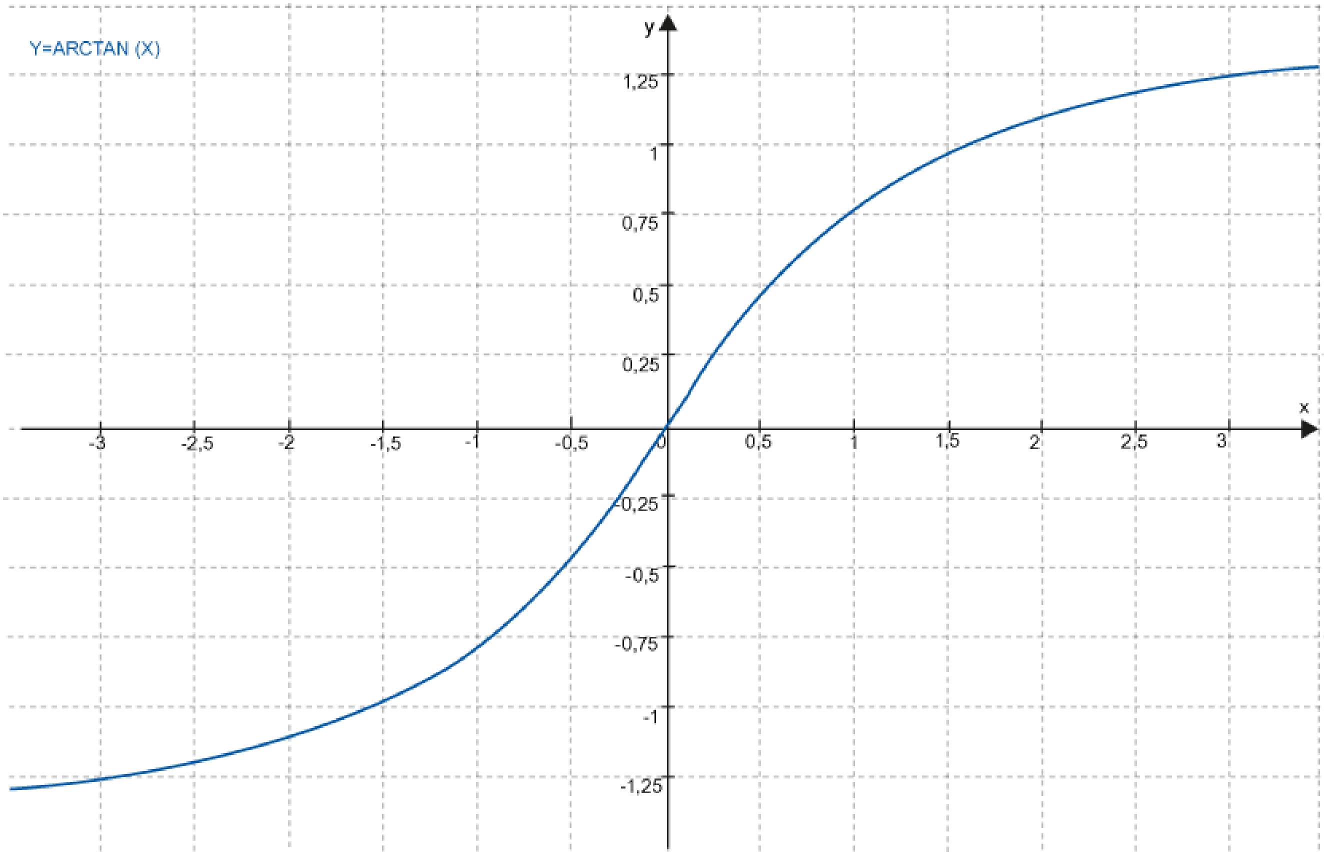
The block determines the associated arc tangent value in radian measure for an argument to be entered at input X and outputs the result at output Y.

Y = arctan X

Permissible input range: -3.402823 E38 to 3.402823 E38

Output range: -π/2 <= Y <= π/2

XY diagram



Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Argument	0.0	REAL	
Y	Arc tangent	0.0	REAL	

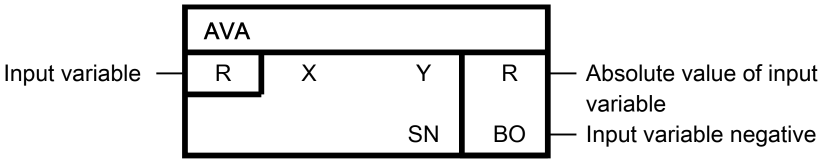
Project data

Can be inserted online	Yes
Special characteristics	-

3.8 AVA Absolute value generator, with sign evaluation

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Arithmetic function block for absolute value generation of type real

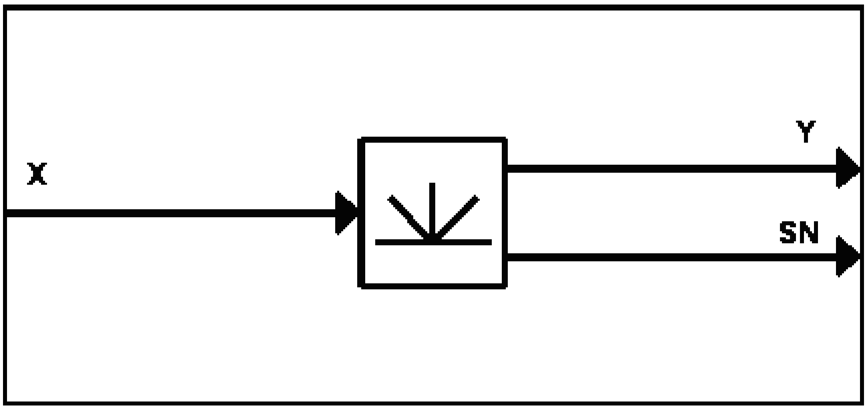
Operation mode

The block generates the absolute value of the value at input X (input variable). The result is output at output Y.

Y = |X|

If the input variable is negative, binary output SN = 1 is set at the same time.

Block diagram



Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
Y	Absolute value of input variable	0.0	REAL	
SN	Input variable negative	0	0/1	

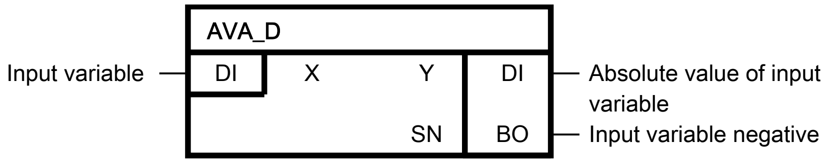
Project data

Can be loaded online	Yes
Special characteristics	-

3.9 AVA_D Absolute value generator (double integer)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Arithmetic function block for absolute value generation of type DOUBLE INTEGER

Operation mode

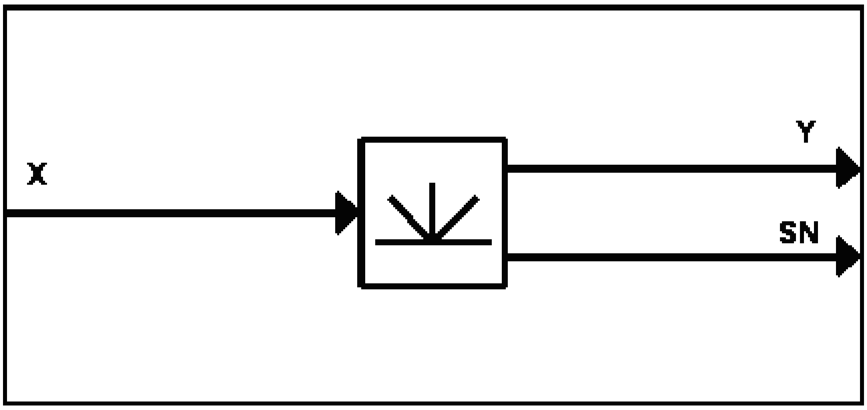
The block generates the absolute value of the value at input X (input variable). The result is output at output Y.

Y = |X|

If the input variable is negative, binary output SN = 1 is set at the same time.

Output values Y -2147483648 and SN 1 are set for input value -2147483648

Block diagram



Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	DINT	
Y	Absolute value of input variable	0	DINT	
SN	Input variable negative	0	0/1	

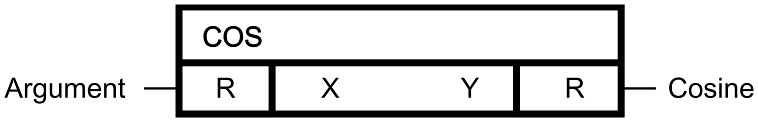
Project data

Can be loaded online	Yes
Special characteristics	-

3.10 COS Cosine function

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

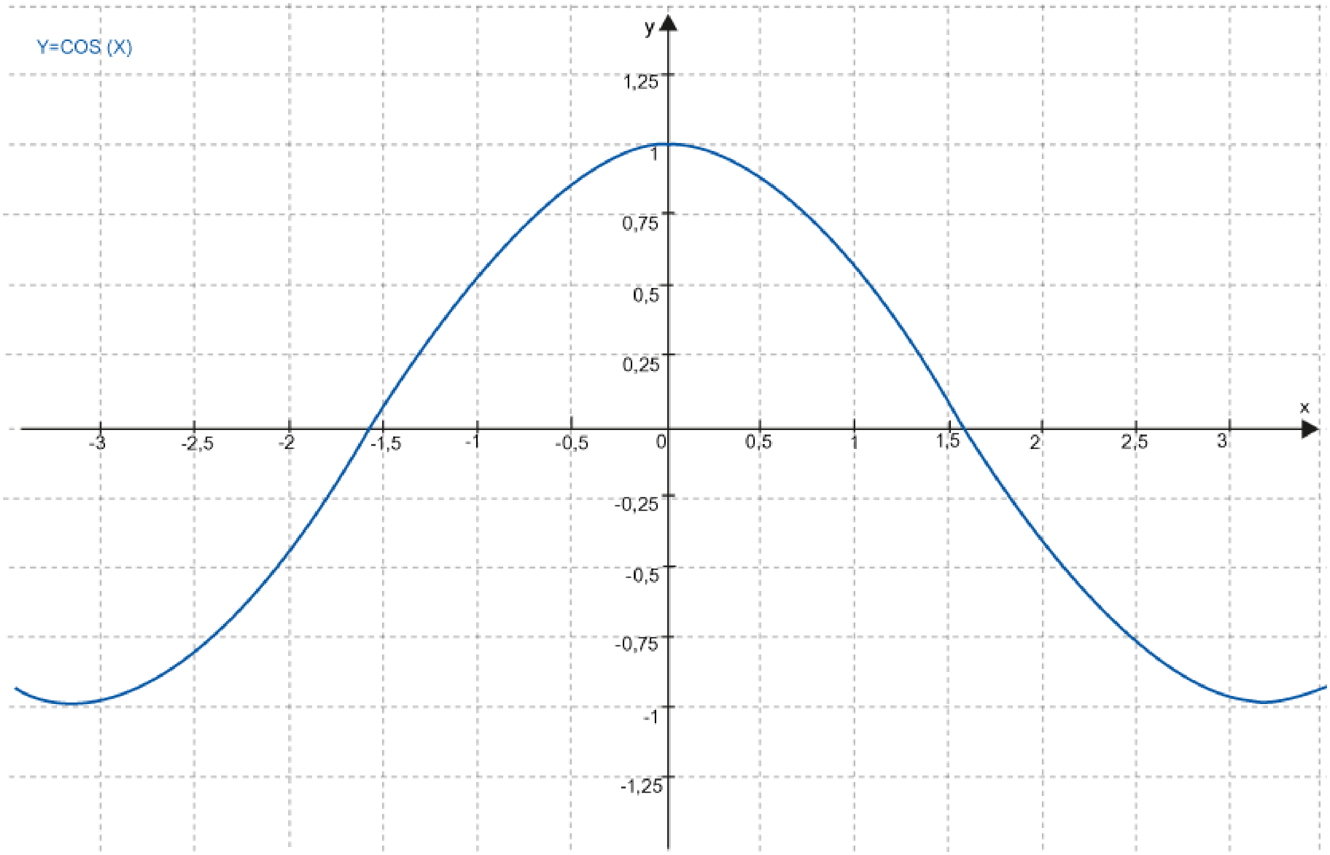
Determination of the cosine value for an argument.

Operation mode

The block determines the associated cosine value in radian measure for an argument to be entered at input X and outputs the result at output Y.

Y = cos X

XY diagram



X is modular π

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Argument	0.0	REAL	
Y	Cosine	1	REAL	

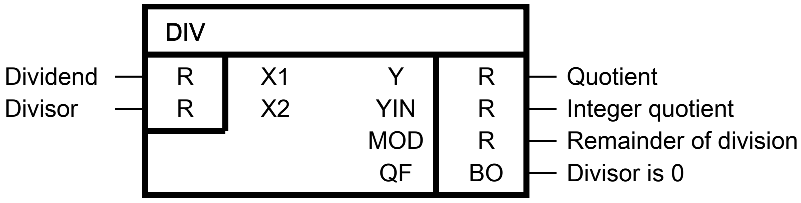
Project data

SIMOTION	as of V4.1
SINAMICS	as of V4.4
Can be loaded online	Yes
Special characteristics	-

3.11 DIV Divider (REAL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Divider with two real-type inputs

Operation mode

The block divides the value entered at connection X1 by the value entered at connection X2.

The result is output at outputs Y, YIN, and MOD:

- The Y output contains the quotient with integer places and decimal places
- The YIN output contains the integer quotient
- The MOD output contains the division remainder (absolute residual value)

The Y output is limited to the range of approx. -3.4 E38 to +3.4 E38.

$$Y = \frac{X1}{X2}$$

MOD = (Y – YIN) * X2

If the output value Y violates the permissible value range of -3.402823 E38 to 3.402823 E38 (because the divisor X2 is very small or less than zero), then the limit value of the output range is output at connection Y with the correct sign. The binary output QF=1 is set at the same time. If X2 is zero, then the outputs YIN and MOD retain their last values.

With division of 0/0, the block output Y remains unchanged. The binary output QF is set to 1. With a division by zero, output MOD retains its last value.

Truth table(s)

The following truth table lists the block responses in the cases specified above.

X1/X2	Y	YIN	MOD	OF
X/0	Limit value with correct sign	YIN n-1	MOD n-1	1
0/0	Y n-1	YIN n-1	MOD n-1	1
0/X	0	0	0	0

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X1	Dividend	0.0	REAL	
X2	Divisor	1	REAL	
Y	Quotient	0.0	REAL	
YIN	Integer quotient	0.0	REAL	
MOD	Remainder of division	0.0	REAL	
QF	Divisor is 0	0	0/1	

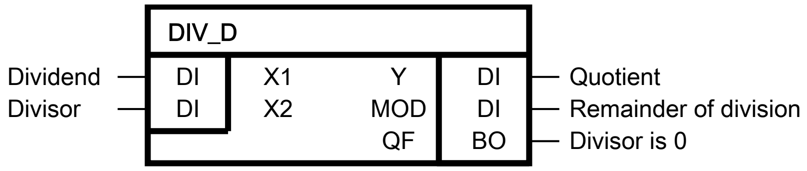
Project data

Can be inserted online	Yes
Special characteristics	-

3.12 DIV_D Divider (double integer type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Divider with two inputs of the double integer type

Operation mode

The block divides the value entered at connection X1 by the value entered at connection X2 taking account of the sign. The quotient is limited to the range of -2147483648 (2^{31}) to 2147483647 (2^{31} -1) and output at connection Y.

$$Y = \frac{X1}{X2}$$

The remainder of division is output at connection MOD. The sign of the division remainder MOD matches that of dividend X1.

MOD = X1 MOD X2

If output value Y violates the permissible value range of -2147483648 (2^{31}) to +2147483647 (2^{31}-1) (when the divisor X2 is zero), then the limit value of the output range is output at connection Y with the correct sign. The binary output QF = 1 is set at the same time.

With division of 0/0, the block output Y remains unchanged. The binary output QF is set to 1. With a division by zero, output MOD retains its last value.

Truth table(s)

The following truth table lists the block responses in the cases specified above.

X1/X2	Y	MOD	OF
X/0	Limit value with correct sign	MOD n-1	1
0/0	Y n-1	MOD n-1	1
0/X	0	0	0

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X1	Dividend	0	DINT	
X2	Divisor	1	DINT	
Y	Quotient	0	DINT	
MOD	Remainder of division	0	DINT	
QF	Divisor is 0	0	0/1	

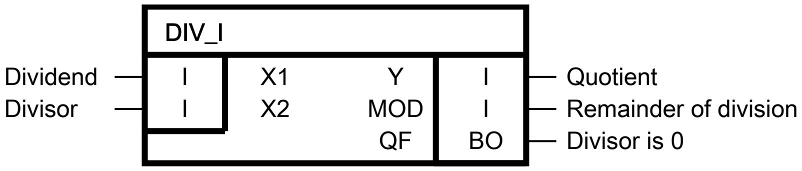
Project data

Can be inserted online	Yes
Special characteristics	-

3.13 DIV_I Divider (integer type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Divider with two integer-type inputs

Operation mode

The block divides the value entered at connection X1 by the value entered at connection X2 taking account of the sign. The quotient is limited to the range of -32768 to +32767 and output at connection Y.

$$Y = \frac{X1}{X2}$$

The remainder of division is output at connection MOD. The sign of the division remainder MOD matches that of dividend X1.

MOD = X1 MOD X2

If output value Y violates the permissible value range of -32768 to +32767 (when the divisor is zero, then the limit value of the output range is output at connection Y with the correct sign. The binary output QF = 1 is set at the same time.

With division of 0/0, the block output Y remains unchanged. The binary output QF is set to 1. With a division by zero, output MOD retains its last value.

Truth table(s)

The following truth table lists the block responses in the cases specified above.

X1/X2	Y	MOD	OF
X/0	Limit value with correct sign	MOD n-1	1
0/0	Y n-1	MOD n-1	1
0/X	0	0	0

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X1	Dividend	0	INT	
X2	Divisor	1	INT	
Y	Quotient	0	INT	
MOD	Remainder of division	0	INT	
QF	Divisor is 0	0	0/1	

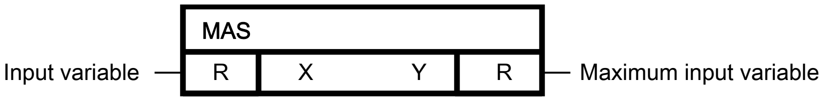
Project data

Can be inserted online	Yes
Special characteristics	-

3.14 MAS Maximum evaluator

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Comparison block with up to four inputs of the REAL type to determine the largest input value present at the time of processing

Operation mode

The block determines the largest of the values present at inputs X 1-4.

The result is output at output Y.

Y = max. {X1,X2,X3,X4}

If the same value is present at all inputs, this value is output as the maximum input variable.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	-3.402823 E38	REAL	
Y	Maximum input variable	0.0	REAL	

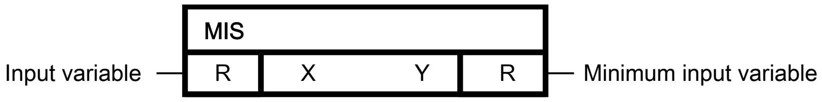
Project data

Can be loaded online	**Yes**
Special characteristics	X comprises up to four inputs (X1 to X4)

3.15 MIS Minimum evaluator

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Comparison block with up to four REAL-type inputs to determine the smallest input value present at the time of processing.

Operation mode

The block determines the smallest of the values present at inputs X 1-4.

The result is output at output Y.

Y = min. {X1,X2, X3, X4}

If the same value is present at all inputs, this value is output as the minimum input variable.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	3.402823 E38	REAL	
Y	Minimum input variable	0.0	REAL	

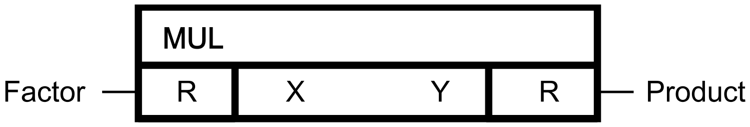
Project data

Can be loaded online	Yes
Special characteristics	X comprises up to four inputs (X1 to X4)

3.16 MUL Multiplier (REAL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Multiplier with up to four real-type inputs

Operation mode

The block multiplies the values entered at the generic inputs X 1-4 taking account of the sign. The result, limited to the range of -3.402823 E38 to +3.402823 E38, is output at output Y.

Y = X1·X2·X3·X4

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Factor	1.0	REAL	
Y	Product	0.0	REAL	

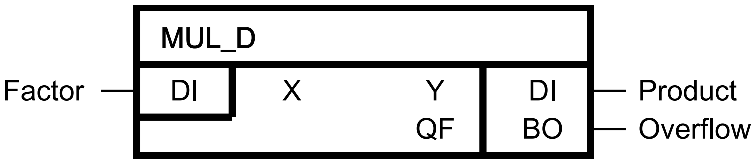
Project data

Can be loaded online	Yes
Special characteristics	X comprises up to four inputs (X1 to X4)

3.17 MUL_D Multiplier (double integer type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Multiplier with up to four double integer-type inputs

Operation mode

The block multiplies the values entered at the generic inputs X 1-4 taking account of the sign. The result, limited to the range of -2147483648 (2^{31}) to +2147483647 (2^{31}-1), is output at output Y.

Y = X1·X2·X3·X4

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Factor	1	DINT	
Y	Product	0	DINT	
QF	Overflow	0	0/1	

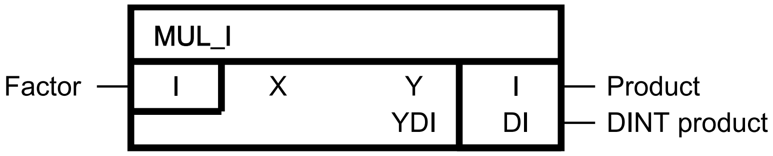
Project data

Can be loaded online	**Yes**
Special characteristics	X comprises up to four inputs (X1 to X4)

3.18 MUL_I Multiplier (integer type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Multiplier with up to four integer-type inputs

Operation mode

The block multiplies the values entered at the generic inputs X 1-4 taking account of the sign. The result, limited to the range of -32768 to +32767, is output at output Y. In addition, the result, limited to a range of -2147483648 (2^{31}) to +2147483647 (2^{31}-1), is output at output YDI.

Y = X1·X2·X3·X4

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Factor	1	INT	
Y	Product	0	INT	
YDI	DINT product	0	DINT	

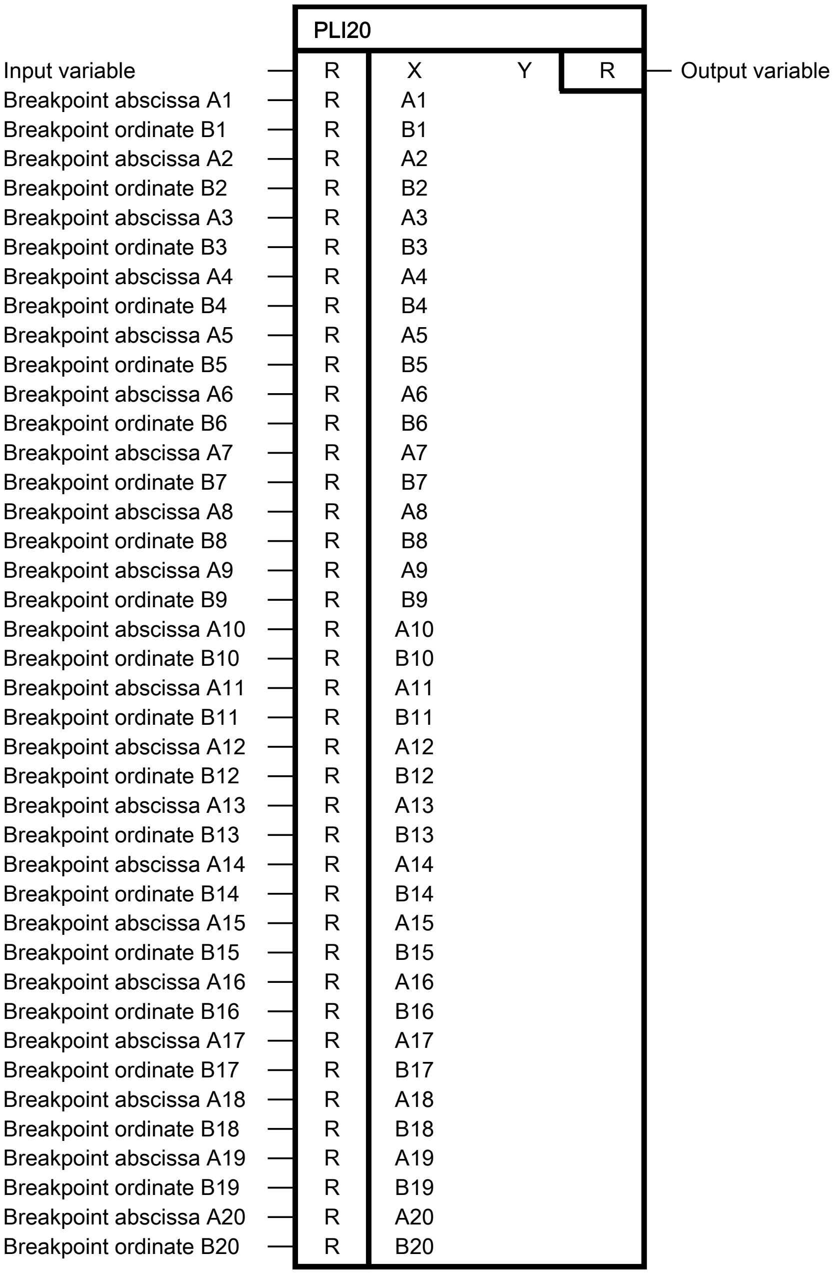
Project data

Can be loaded online	Yes
Special characteristics	X comprises up to four inputs (X1 to X4)

3.19 PLI20 Polyline, 20 breakpoints

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Block of the REAL type

- For linearization of characteristic curves
- For simulation of non-linear transfer elements
- For controller gain defined in sections

Operation mode

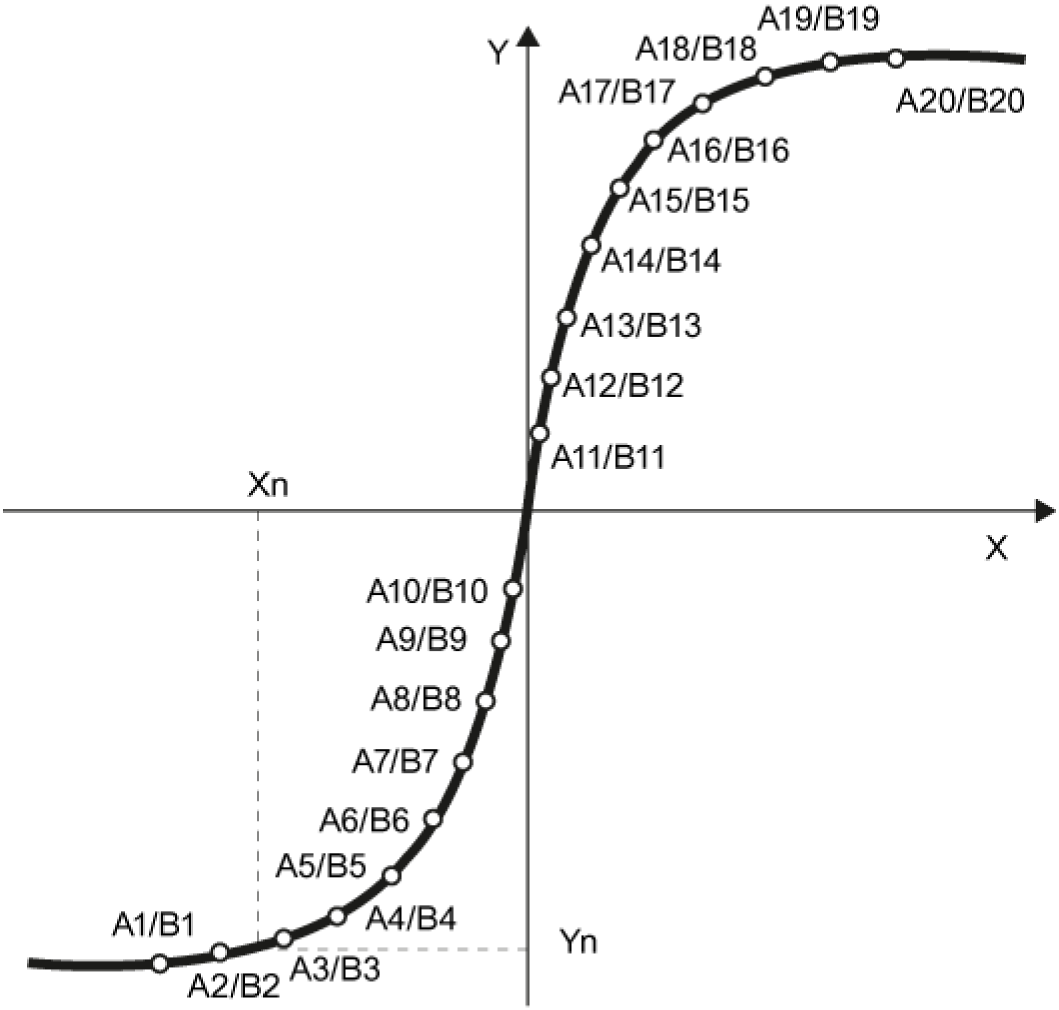
- This block arbitrarily adapts the output variable Y to the input variable X via max. 20 breakpoints in 4 quadrants.
- Interpolation is carried out linearly between the breakpoints. The characteristic is horizontal outside of A1 or A20.

Configuring guide

During configuration, you must ensure that the values of A1 to A20 are sorted in ascending order otherwise incorrect values are output. The ordinate values B1 to B20 can be selected arbitrarily, i.e. irrespective of the preceding value.

If breakpoints are not needed (e.g. as of A16/B16), the following abscissas and ordinates (A16/B16 to A20/B20) must be assigned the same values as A15/B15.

Example



Simulation of magnetization characteristic curve

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
A1	Breakpoint abscissa A1	0.0	REAL	
B1	Breakpoint ordinate B1	0.0	REAL	
A2	Breakpoint abscissa A2	0.0	REAL	
B2	Breakpoint ordinate B2	0.0	REAL	
A3	Breakpoint abscissa A3	0.0	REAL	
B3	Breakpoint ordinate B3	0.0	REAL	
A4	Breakpoint abscissa A4	0.0	REAL	
B4	Breakpoint ordinate B4	0.0	REAL	
A5	Breakpoint abscissa A5	0.0	REAL	
B5	Breakpoint ordinate B5	0.0	REAL	
A6	Breakpoint abscissa A6	0.0	REAL	
B6	Breakpoint ordinate B6	0.0	REAL	
A7	Breakpoint abscissa A7	0.0	REAL	
B7	Breakpoint ordinate B7	0.0	REAL	
A8	Breakpoint abscissa A8	0.0	REAL	
B8	Breakpoint ordinate B8	0.0	REAL	
A9	Breakpoint abscissa A9	0.0	REAL	
B9	Breakpoint ordinate B9	0.0	REAL	
A10	Breakpoint abscissa A10	0.0	REAL	
B10	Breakpoint ordinate B10	0.0	REAL	
A11	Breakpoint abscissa A11	0.0	REAL	
B11	Breakpoint ordinate B11	0.0	REAL	
A12	Breakpoint abscissa A12	0.0	REAL	
B12	Breakpoint ordinate B12	0.0	REAL	
A13	Breakpoint abscissa A13	0.0	REAL	
B13	Breakpoint ordinate B13	0.0	REAL	
A14	Breakpoint abscissa A14	0.0	REAL	
B14	Breakpoint ordinate B14	0.0	REAL	
A15	Breakpoint abscissa A15	0.0	REAL	
B15	Breakpoint ordinate B15	0.0	REAL	
A16	Breakpoint abscissa A16	0.0	REAL	
B16	Breakpoint ordinate B16	0.0	REAL	
A17	Breakpoint abscissa A17	0.0	REAL	
B17	Breakpoint ordinate B17	0.0	REAL	
A18	Breakpoint abscissa A18	0.0	REAL	
B18	Breakpoint ordinate B18	0.0	REAL	
A19	Breakpoint abscissa A19	0.0	REAL	
B19	Breakpoint ordinate B19	0.0	REAL	
A20	Breakpoint abscissa A20	0.0	REAL	
B20	Breakpoint ordinate B20	0.0	REAL	
Y	Output variable	0.0	REAL	

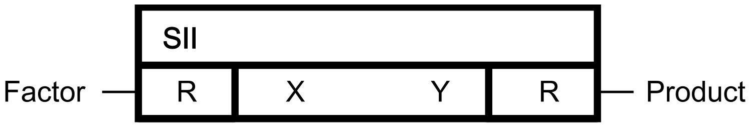
Project data

Can be loaded online	Yes
Special characteristics	-

3.20 SII Inverter

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

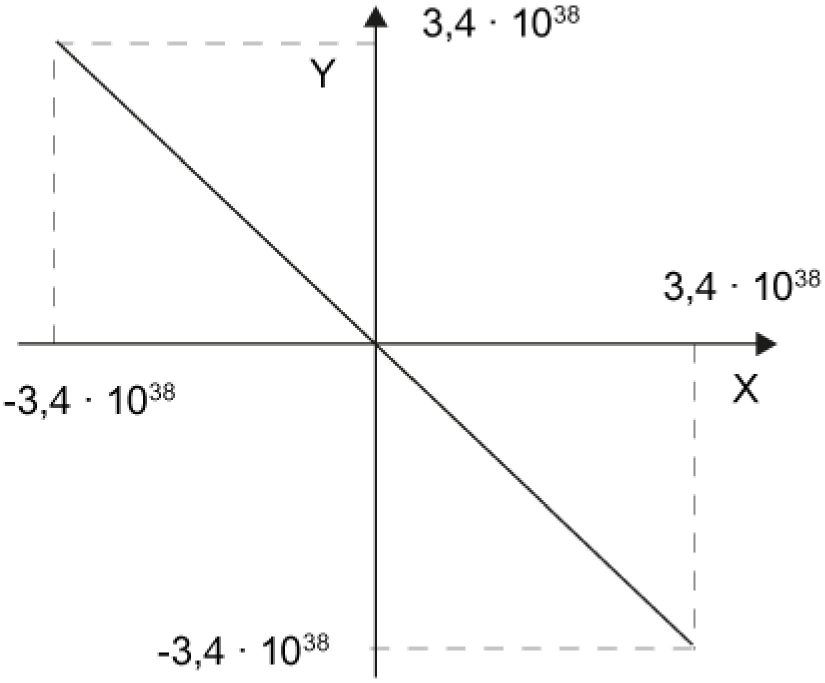
- Inverter with one Real-type input
- Arithmetic function block for sign reversal

Operation mode

The block inverts input variable X and outputs the result at block output Y (in accordance with the following transmission characteristic).

Y=-X

Transfer function



Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Factor	0.0	REAL	
Y	Product	0.0	REAL	

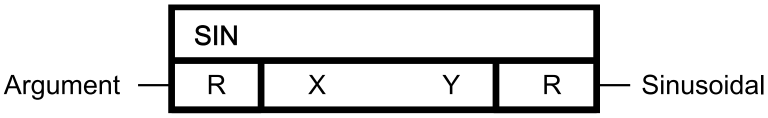
Project data

Can be loaded online	Yes
Special characteristics	-

3.21 SIN Sine function

☑ SIMOTION ☑ SINAMICS

Symbol



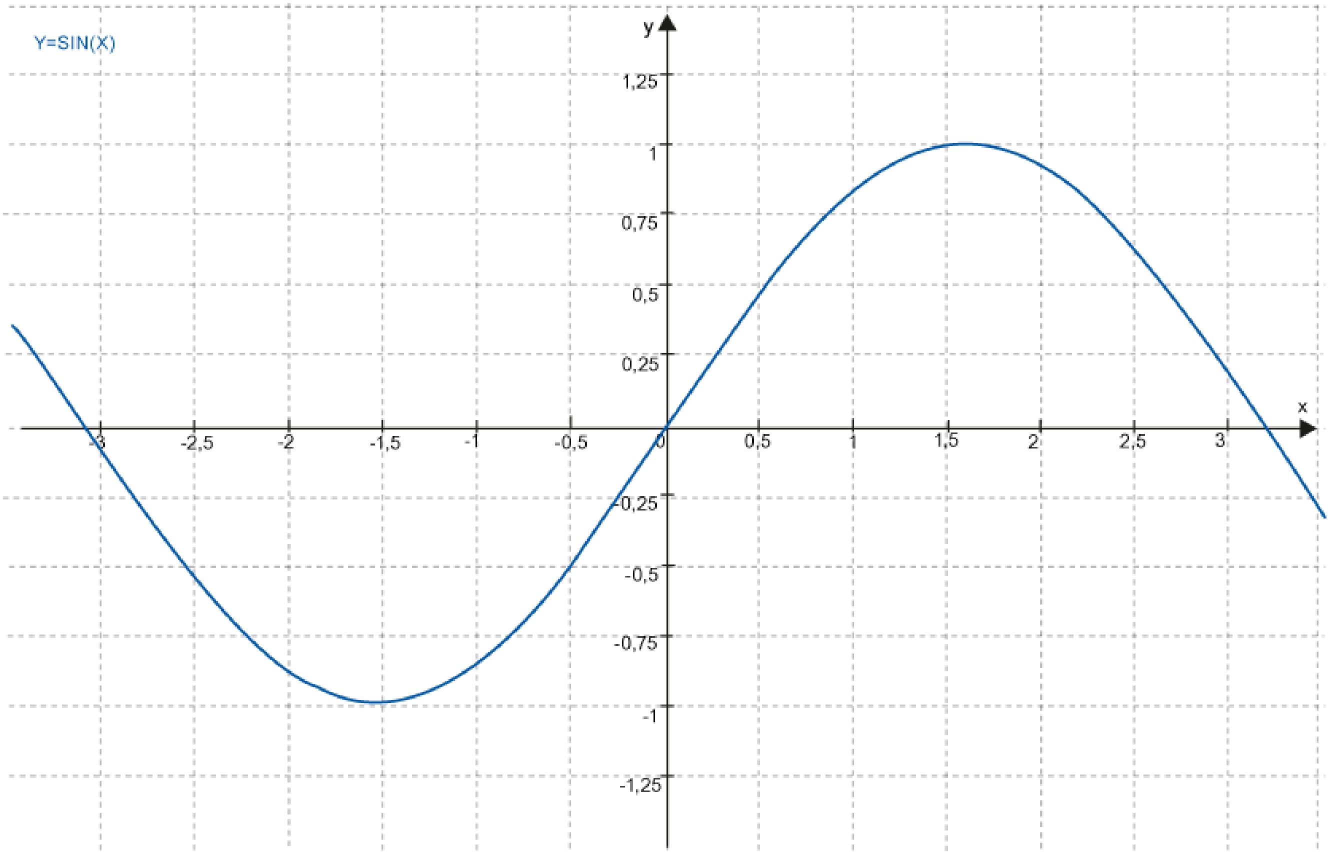
Short description

Determination of the sine value for an argument.

Operation mode

- The block determines the associated sine value in radian measure for an argument to be entered at input X and outputs the result at output Y.
- Y = sin X

XY diagram



X is modular π

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Argument	0.0	REAL	
Y	Sinusoidal	0.0	REAL	

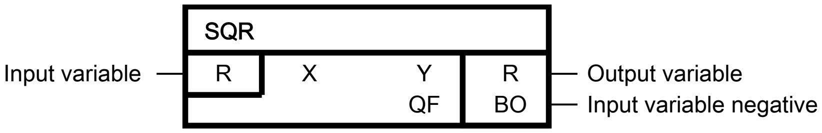
Project data

SIMOTION	as of V4.1
SINAMICS	as of V4.4
Can be inserted online	Yes
Special characteristics	-

3.22 SQR Square-root extractor

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

Arithmetic function block for determining the square root.

The SQR block is available as of SIMOTION V4.1 / SINAMICS V4.4.

Operation mode

The block calculates the square root of the value entered at connection X. The result is output at connection Y.

$Y = \sqrt{X}$

If the input variable is negative, the value zero is output at connection Y. The binary output QF = 1 is set at the same time.

Truth table(s)

Conditions	Y	QF
X > 0	SQR(X)	0
X = 0	0	0
X < 0	0	1

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
Y	Output variable	0.0	REAL	
QF	Input variable negative	0	0/1	

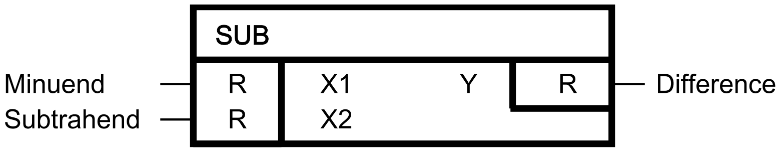
Project data

Can be inserted online	Yes
Special characteristics	-

3.23 SUB Subtractor (REAL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Subtractor with two Real-type inputs

Operation mode

- The block subtracts the value entered at connection X2 from the value entered at connection X1, taking account of the sign. The result is limited to the range of -3.402823 E38 to 3.402823 E38 and output at output Y.
- Y= X1-X2

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X1	Minuend	0.0	REAL	
X2	Subtrahend	0.0	REAL	
Y	Difference	0.0	REAL	

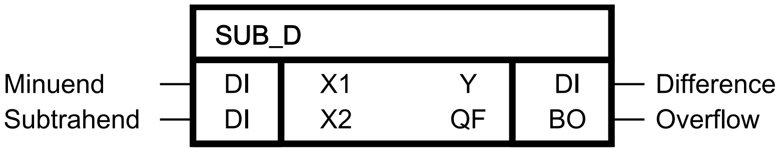
Project data

Can be loaded online	Yes
Special characteristics	-

3.24 SUB_D Subtractor (double integer type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Subtractor with two double integer-type inputs

Operation mode

The block subtracts the value entered at connection X2 from the value entered at connection X1, taking account of the sign. The result is limited to the range of -2147483648 (2^{31}) to +2147483647 (2^{31}-1) and output at output Y. An overflow is displayed at the binary output with QF=1.

Y= X1-X2

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X1	Minuend	0	DINT	
X2	Subtrahend	0	DINT	
Y	Difference	0	DINT	
QF	Overflow	0	0/1	

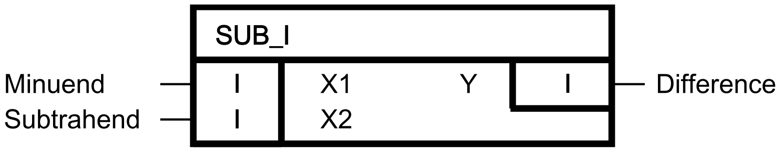
Project data

Can be loaded online	Yes
Special characteristics	-

3.25 SUB_I Subtractor (integer type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Subtractor with two Integer-type inputs

Operation mode

- The block subtracts the value entered at connection X2 from the value entered at connection X1, taking account of the sign. The result, limited to a range of approximately -32768 to 32767, is output at output Y.
- Y= X1-X2

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X1	Minuend	0	INT	
X2	Subtrahend	0	INT	
Y	Difference	0	INT	

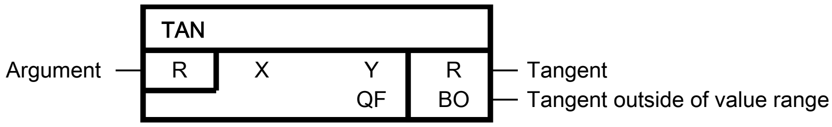
Project data

Can be loaded online	**Yes**
Special characteristics	-

3.26 TAN Tangent

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

Determination of the tangent value for an angle

Operation mode

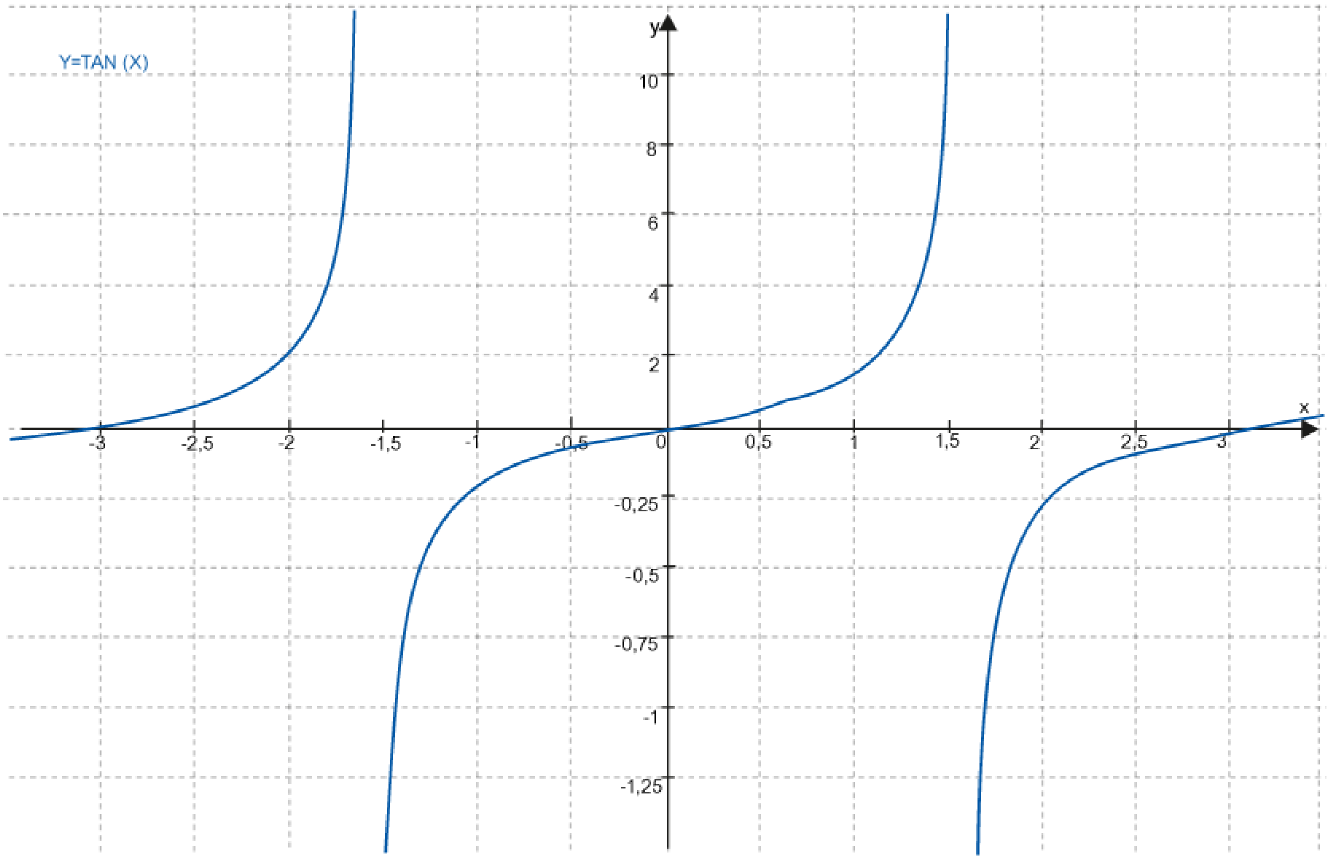
The block determines the associated tangent value in radian measure for an argument to be entered at input X and outputs the result at output Y.

Y = tan X

Output range: -3.402823 E38 to 3.402823 E38

If the determined tangent value is outside the range of -3.402823 E38 to 3.402823 E38, block output Y is set to -3.402823 E38 or +3.402823 E38 and binary output QF = 1 is set at the same time.

Transfer function



Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Argument	0.0	REAL	
Y	Tangent	0.0	REAL	
QF	Tangent outside of value range	0	0/1	

Project data

Can be inserted online	Yes
Special characteristics	-

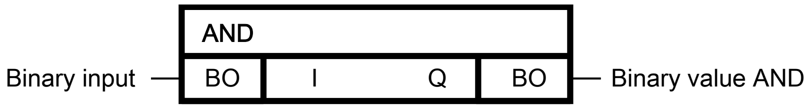
Logic

4

4.1 AND Logic AND operation (BOOL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

AND block with up to four inputs of the BOOL type

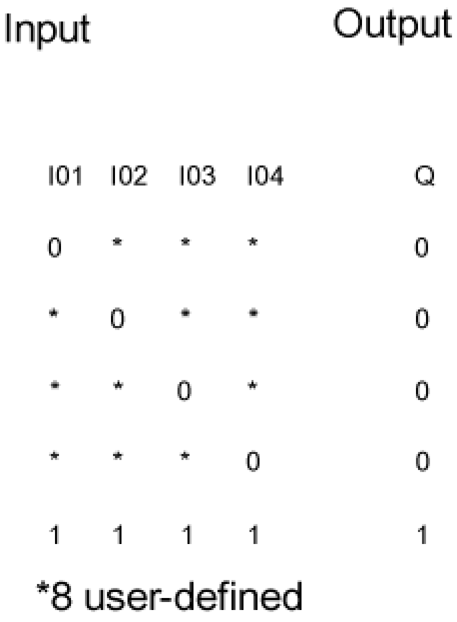
Operation mode

The block combines the binary values at the inputs I 1-4 to a logic AND and outputs the result at its binary output Q.

$Q = I_{01} \wedge ... \wedge I_{04}$

Output Q = 1 when the value 1 is present at all generic inputs I1 to I4. In all other cases, output Q = 0

Truth table(s)



Input				Output
I01	I02	I03	I04	Q
0	*	*	*	0
*	0	*	*	0
*	*	0	*	0
*	*	*	0	0
1	1	1	1	1

*8 user-defined

Block connections

Block connection	Description	Preassignment	Value range	Attributes
I	Binary input	1	0/1	
Q	Binary value AND	0	0/1	

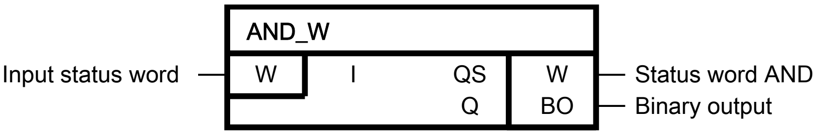
Project data

Can be loaded online	Yes
Special characteristics	I comprises up to four connections (I1 to I4)

4.2 AND_W Logic AND operation (WORD type)

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

AND_W block with up to four inputs of the WORD type

Operation mode

16 binary states are combined in a status word.

The block combines the status words I_{01} to I_{16} bit-by-bit according to the logic AND function. The corresponding bits of status word AND are then set at block output QS.

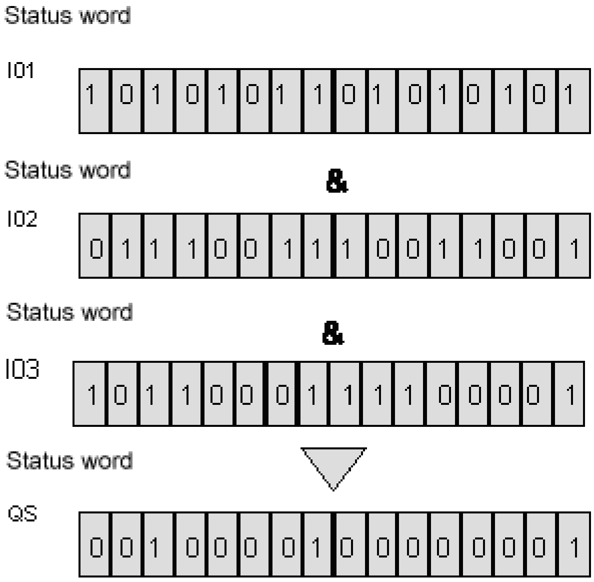
The following applies to the kth bit of the "and" status word:

$$QS_k = I_{01_k} \wedge .. \wedge I_{nn_k} \quad , k = 1 ... 16$$

A bit of the status word is equal to 0 when at least one of the equivalent bits on the block inputs I1 to I4 is equal to 0.

The binary output Q is 1 if at least one bit of the status word AND is equal to 1.

Following state diagram (for 3 inputs)



Status word

I01

1 0 1 0 1 0 1 1 0 1 0 1 0 1 0 1

Status word

&

I02

0 1 1 1 0 0 1 1 1 0 0 1 1 0 0 1

Status word

&

I03

1 0 1 1 0 0 0 1 1 1 1 0 0 0 0 1

Status word

QS

0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1

Block connections

Block connection	Description	Preassignment	Value range	Attributes
I	Input status word	16#FFFF	WORD	
QS	Status word AND	16#0000	WORD	
Q	Binary output	0	0/1	

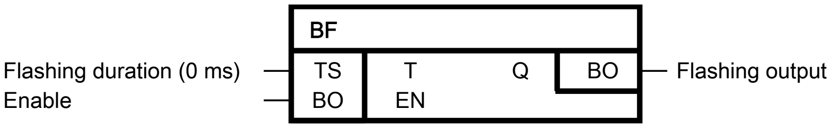
Project data

Can be loaded online	Yes
Special characteristics	I comprises up to four inputs (I1 to I4)

4.3 BF Flashing function (BOOL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Block of the BOOL type

- for controlling signal encoders
- as clock generators

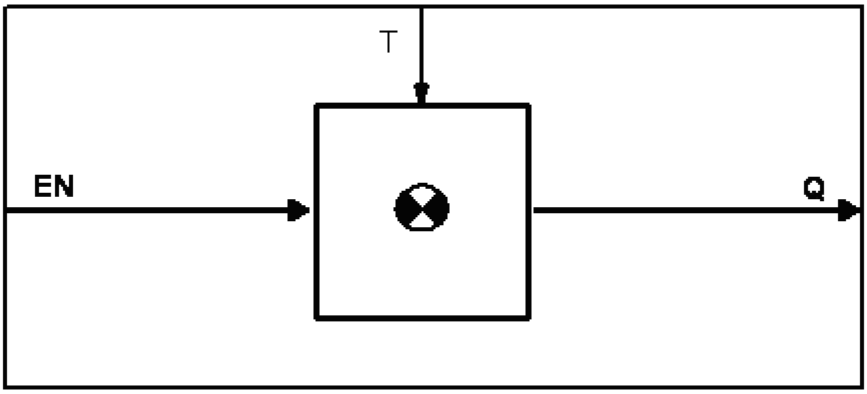
Operation mode

In the rhythm of the time interval T, the block sets its output Q alternately to 1 and 0, when input EN = 1.

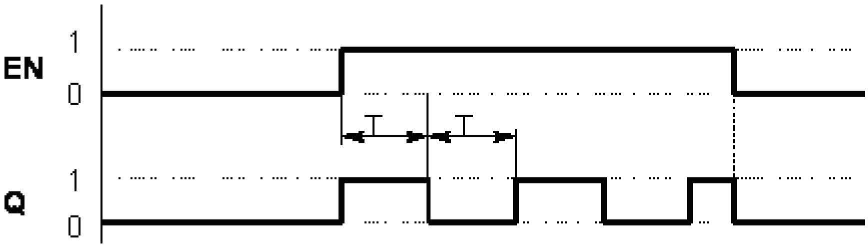
If enable input EN = 0, then output Q = 0.

In this case, T is both the light duration and the dark duration.

Block diagram



Time diagram



Flashing pulse Q subject to flashing duration T and the enable EN

Block connections

Block connection	Description	Preassignment	Value range	Attributes
T	Flashing duration (0 ms)	0	SDTIME	
EN	Enable	0	0/1	
Q	Flashing output	0	0/1	

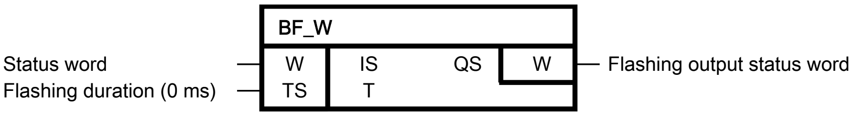
Project data

Can be loaded online	Yes
Special characteristics	-

4.4 BF_W Image function for status word (Type WORD)

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

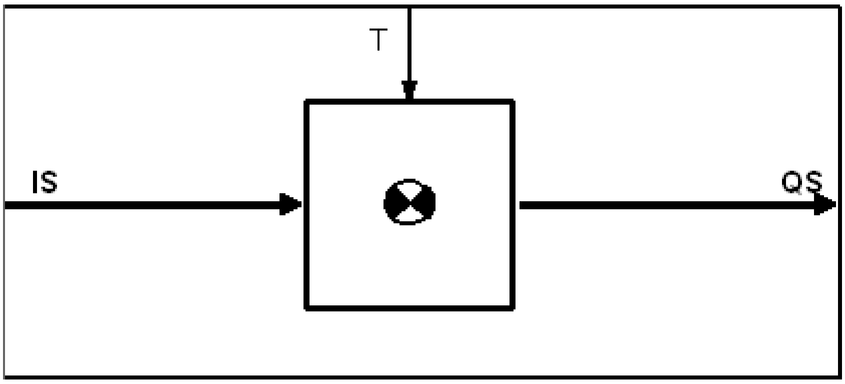
Block of the WORD type for control of the sensor combinations

Operation mode

In the rhythm of the time interval T, the block sets all bits of the input status word IS that have the value logic 1, alternately to 1 and 0 in the output status word QS.

In this case, T is both the light duration and the dark duration.

Block diagram



Block connections

Block connection	Description	Preassignment	Value range	Attributes
IS	Status word	16#0000	WORD	
T	Flashing duration (0 ms)	0	SDTIME	
QS	Flashing output status word	16#0000	WORD	

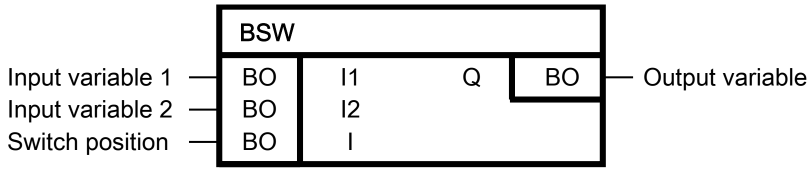
Project data

Can be loaded online	Yes
Special characteristics	-

4.5 BSW Binary change-over switch (BOOL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

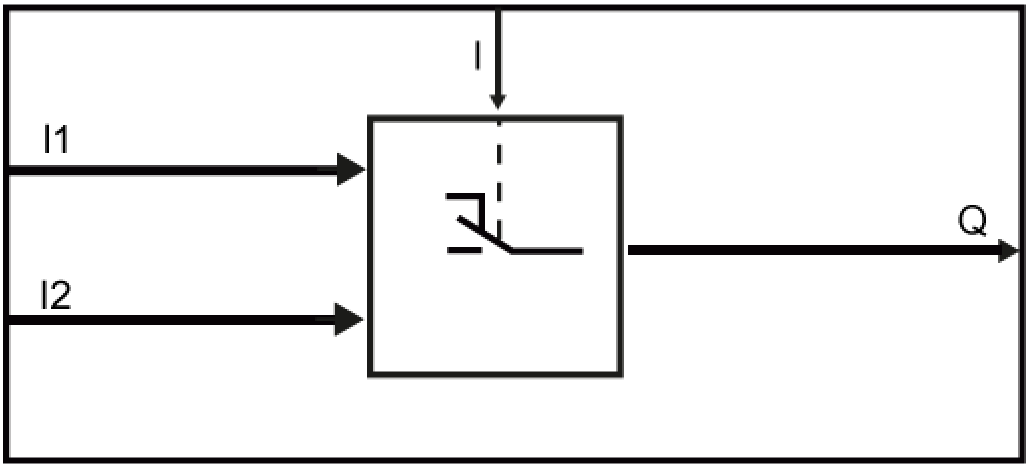
The block switches one of two binary input variables to the output

Operation mode

If input I = 0, then I1 is given to output Q.

If input I = 1, then I2 is given to output Q.

Block diagram



Truth table(s)

Switch position 1	Output variable Q
0	Q = I1
1	Q = I2

Initialization

If input I = 0, then I1 is given to output Q.

If input I = 1, then I2 is given to output Q.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
I1	Input variable 1	0	0/1	
I2	Input variable 2	0	0/1	
I	Switch position	0	0/1	
Q	Output variable	0	0/1	

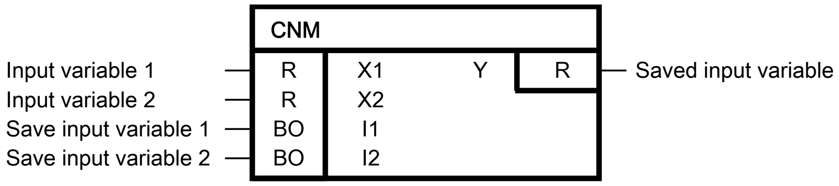
Project data

Can be loaded online	Yes
Special characteristics	-

4.6 CNM Controllable numeric memory (REAL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Block of the REAL type for saving a current input value (Engl. Sample and hold function) with

- Selectable input
- Selectable save time
- Rising edge-initiated triggering

The blocks CNM_I and CNM_ D fulfill the same function. They only differ in the data type used.

Operation mode

On a rising edge at I1, X1 is switched through to output Y.

On a rising edge at I2, X2 is switched through to output Y.

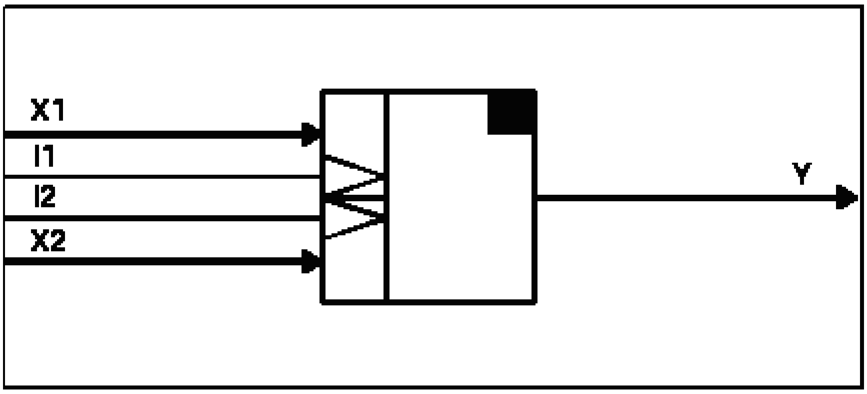
The saved input variable remains pending on Y until the next rising edge at I1 or I2 switches through the next momentary value.

In the case of a simultaneous rising edge at I1 and I2, I1 receives priority, and X1 is switched through to Y.

Initialization

If input I1 or I2 receives the value 1 during initialization of an upstream output, the block does not detect a positive edge during the first cyclic pass. Otherwise, the block detects a positive edge during the first cyclic pass. In START mode (edge trigger flag), the values for I1 and I2 are stored temporarily.

Block diagram



Truth table(s)

Input		Output Y at the time of triggering
I1	I2	
*	*	$Y_n = Y_{n-1}$
*	0 -> 1	$Y_n = X2_n$
0 -> 1	*	$Y_n = X1_n$
0 -> 1	0 -> 1	$Y_n = X1_n$

*: No rising edge0 -> 1: Rising edge

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X1	Input variable 1	0.0	REAL	
X2	Input variable 2	0.0	REAL	
I1	Save input variable 1	0.0	0/1	
I2	Save input variable 2	0.0	0/1	
Y	Saved input variable	0.0	REAL	

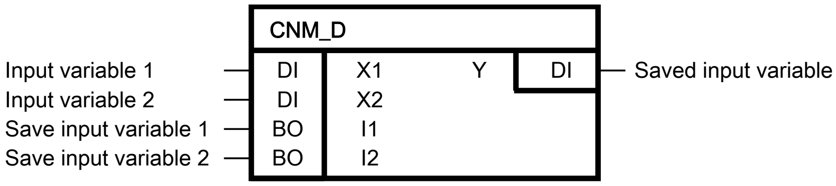
Project data

Can be loaded online	Yes
Special characteristics	-

4.7 CNM_D Controllable numeric memory (DOUBLE INTEGER type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Block of the DOUBLE INTEGER type for saving a current input value (Engl. Sample and hold function) with

- Selectable input
- Selectable save time
- Rising edge-initiated triggering

The blocks CNM and CNM_ i fulfill the same function. They only differ in the data type used.

Operation mode

On a rising edge at I1, X1 is switched through to output Y.

On a rising edge at I2, X2 is switched through to output Y.

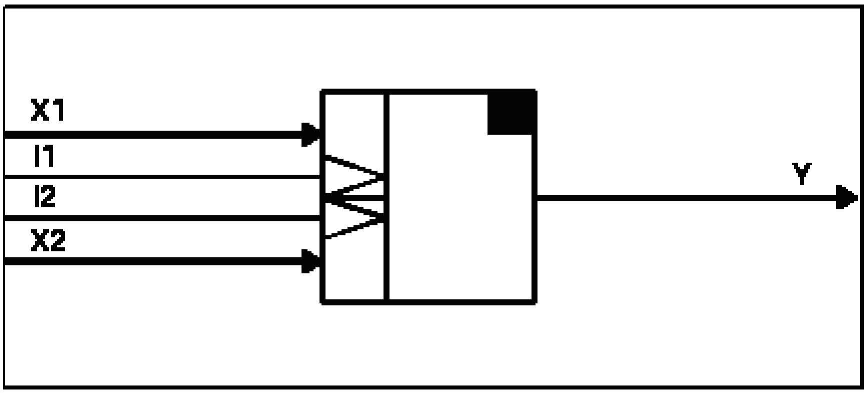
The saved input variable remains pending on Y until the next rising edge at I1 or I2 switches through the next momentary value.

In the case of a simultaneous rising edge at I1 and I2, I1 receives priority, and X1 is switched through to Y.

Initialization

If input I1 or I2 receives the value 1 during initialization of an upstream output, the block does not detect a positive edge during the first cyclic pass. The block detects a positive edge during the first cyclic pass. In START mode, the values for I1 and I2 are stored temporarily.

Block diagram



Truth table(s)

Input		Output Y at the time of triggering
I1	I2	
*	*	$Y_n = Y_{n-1}$
*	0 -> 1	$Y_n = X2_n$
0 -> 1	*	$Y_n = X1_n$
0 -> 1	0 -> 1	$Y_n = X1_n$

*: No rising edge0 -> 1: Rising edge

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X1	Input variable 1	0	DINT	
X2	Input variable 2	0	DINT	
I1	Save input variable 1	0	0/1	
I2	Save input variable 2	0	0/1	
Y	Saved input variable	0	DINT	

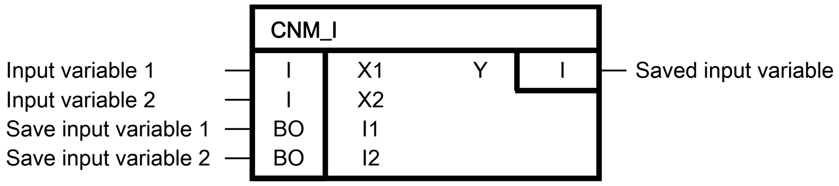
Project data

Can be loaded online	Yes
Special characteristics	-

4.8 CNM_I Controllable numeric memory (INTEGER type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Block of the INTEGER type for saving a current input value (Engl. Sample and hold function) with

- Selectable input
- Selectable save time
- Rising edge-initiated triggering

The CNM and CNM_D blocks have the same function. They only differ in the data type used.

Operation mode

On a rising edge at I1, X1 is switched through to output Y.

On a rising edge at I2, X2 is switched through to output Y.

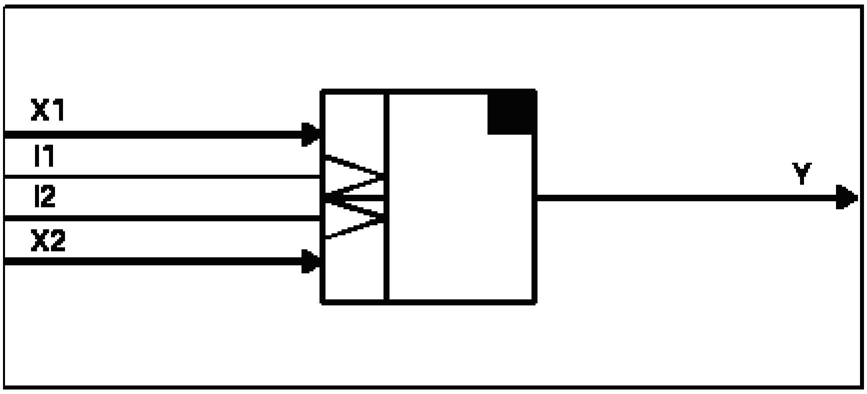
The saved input variable remains pending on Y until the next rising edge at I1 or I2 switches through the next momentary value.

In the case of a simultaneous rising edge at I1 and I2, I1 receives priority, and X1 is switched through to Y.

Initialization

If input I1 or I2 receives the value 1 during initialization of an upstream output, the block does not detect a positive edge during the first cyclic pass. The block detects a positive edge during the first cyclic pass. In START mode, the values for I1 and I2 are stored temporarily.

Block diagram



Truth table(s)

Input		Output Y at the time of triggering
I1	I2	
*	*	$Y_n = Y_{n-1}$
*	0 -> 1	$Y_n = X2_n$
0 -> 1	*	$Y_n = X1_n$
0 -> 1	0 -> 1	$Y_n = X1_n$

*: No rising edge0 -> 1: Rising edge

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X1	Input variable 1	0	INT	
X2	Input variable 2	0	INT	
I1	Save input variable 1	0	0/1	
I2	Save input variable 2	0	0/1	
Y	Saved input variable	0	INT	

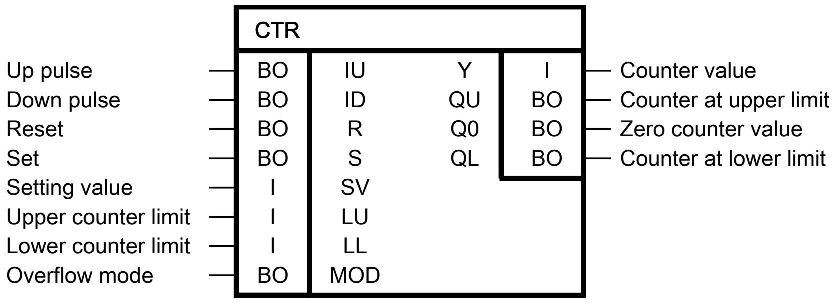
Project data

Can be loaded online	Yes
Special characteristics	-

4.9 CTR Counter (BOOL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Block for Up/Down counting with the following counter functions:

- Set counter to zero
- Hold counter at zero (disable)
- Set counter to initial value

Independent setting of upper and lower counter limit.

Operation mode

This block forms an edge-triggered Up/Down counter. With a rising edge of a pulse at input IU, the counter value is incremented.

With a rising edge of a pulse at input ID, the counter value is decremented. The counter value is present at output Y. Controlling the counter (see also truth table). With S=1, the counter value Y can be preset with the set value SV.

However, the reset input R has priority over the set input. As long as R is logic 1, Y is kept at 0. The counter is locked If Y is not in the counting range between LL and LU, the output at R= 1 will be set to the effective limit value.

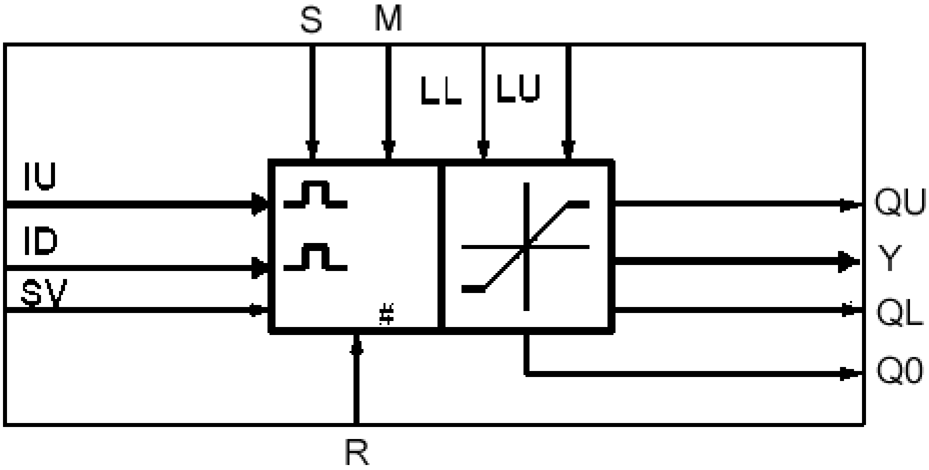
The working area of the counter can be specified via LU (upper counter limit value) or LL (lower counter limit value).

The setting value (SV) is in the range from LL >= SV >= LU.

MOD=0	When these limits are reached, the counter stops counting and the display QU (counter at upper limit) or QL (counter at lower limit) is set.
MOD=1	When the upper limit (LU) is reached, the counter value is set to the lower limit when the next Up pulse occurs; QU = 1 indicates positive overflow for a cycle.
	When the lower limit (LL) is reached, the counter value is set to the upper limit when the next Down pulse occurs; QL = 1 indicates negative overflow for a cycle.

When the counter value is zero, the output Q0 is set to 1.

Block diagram



Truth table(s)

Binary command	Binary command	Counter value Y
S	R	
0	0	Y is retained
0	1	Y is reset
1	0	Y = SV (set value)
1	1	Y is reset

Counter value when set/reset command is given

Initialization

The initialization defines the start value for the first cyclic pass. If input ID or IU is preset with 1, the block cannot detect a positive edge during the first cyclic pass.

Boundary conditions:

- LL <= Y <= LU for LL <LU
- Y = LU for LL >= LU

Block connections

Block connection	Description	Preassignment	Value range	Attributes
IU	Up pulse	0	0/1	
ID	Down pulse	0	0/1	
R	Reset	0	0/1	
S	Set	0	0/1	
SV	Setting value	0	INT	
LU	Upper counter limit	0	INT	
LL	Lower counter limit	0	INT	
MOD	Overflow mode	0	0/1	
Y	Counter value	0	INT	
QU	Counter at upper limit	0	0/1	
Q0	Zero counter value	0	0/1	
QL	Counter at lower limit	0	0/1	

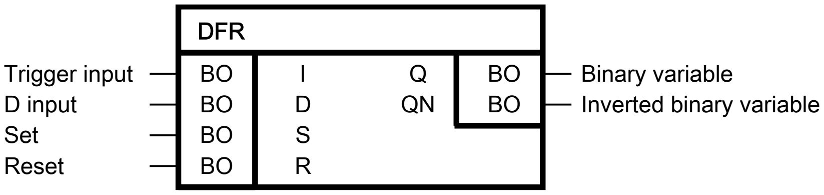
Project data

Can be loaded online	Yes
Special characteristics	-

4.10 DFR Reset-dominant D-type flip-flop (BOOL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Block of the BOOL type for use as reset-dominant D-type flip-flop

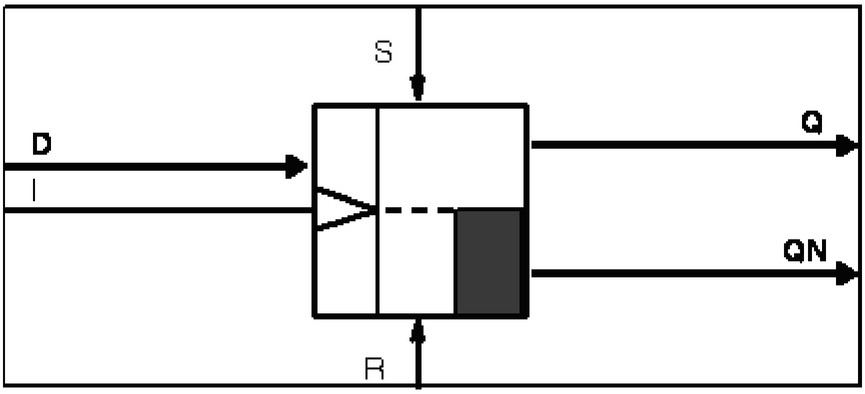
Operation mode

If the two inputs S and R are logic 0, the D input information is switched through to output Q on a rising edge at trigger input I. Output QN always has the value inverse to Q. With logic 1 at input S, output Q is set to logic 1. If input R is set to logic 1, then output Q is set to logic 0. If both inputs are logic 0, then Q does not change. However, if the two inputs S and R are logic 1, then Q is logic 0, since the reset input dominates.

Initialization

If input I receives the value 1 during initialization from an upstream output, the block does not detect a positive edge during the first cyclic pass. The block detects a positive edge during the first cyclic pass. In START mode, the value for I is stored temporarily.

Block diagram

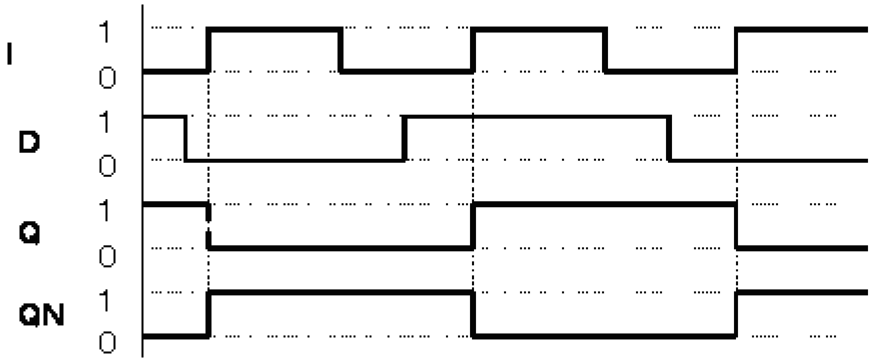


Truth table(s)

D	I	Binary command		Output states	
		S	R	Q	QN
0	0 -> 1	0	0	0	1
1	0 -> 1	0	0	1	0
*	1 -> 0	0	0	Q_{n-1}	Q_{n-1}
*	*	0	1	0	1
*	*	1	0	1	0
*	*	1	1	0	1

Time diagram

With D and I



Output pulse Q subject to the D input and input pulse I for S = R = 0

Block connections

Block connection	Description	Preassignment	Value range	Attributes
I	Trigger input	0	0/1	
D	D input	0	0/1	
S	Set	0	0/1	
R	Reset	0	0/1	
Q	Binary variable	0	0/1	
QN	Inverted binary variable	1	0/1	

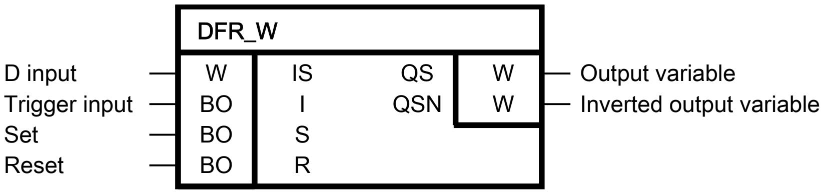
Project data

Can be loaded online	Yes
Special characteristics	-

4.11 DFR_W Reset-dominant D-type flip-flop (WORD type)

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

Block of the WORD type for use as reset-dominant D-type flip-flop

Operation mode

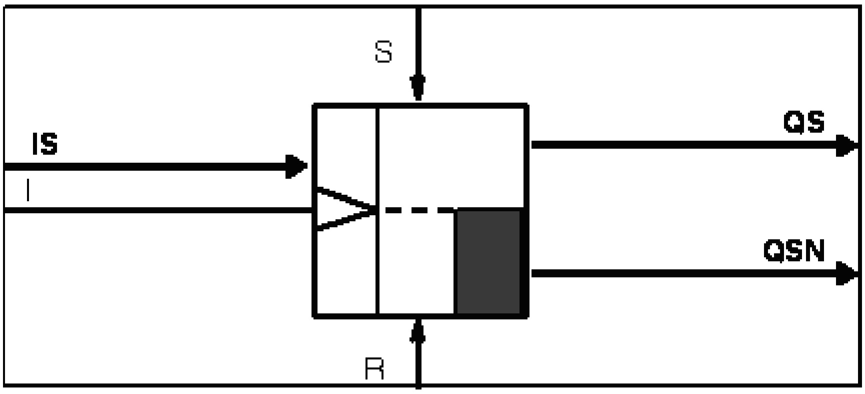
If the two inputs S and R = 0, the D input information is switched through to output QS on a rising edge at trigger input I. Output QSN always has the value inverse to QS. If S = 1, all bits of output variable QS are set to 1. If R = 1, all bits of output variable QS are set to 0. If both inputs S and R = 0, then QS does not change. If the two inputs S and R = 1, all bits of output variable QS are set to 0, since the reset input R dominates.

Initialization

If input I receives the value 1 during initialization from an upstream output, the block does not detect a positive edge during the first cyclic pass.

The block detects a positive edge during the first cyclic pass. In START mode, the value for I is stored temporarily.

Block diagram



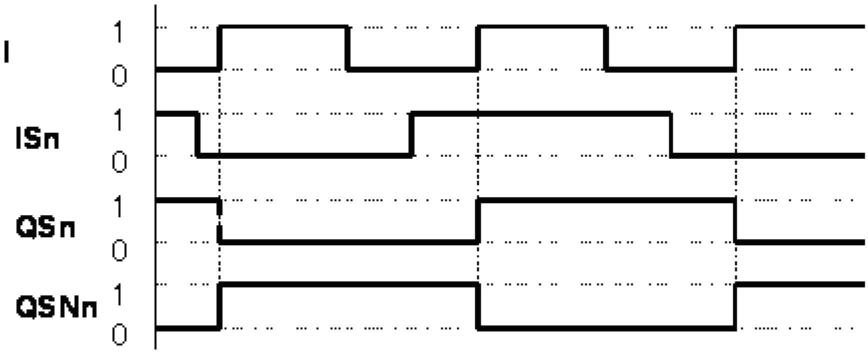
Truth table(s)

I	Binary command		Output states	
	S	R	QS	QSN
0 -> 1	0	0	IS	IS inverted
*	0	1	0	1
*	1	0	1	0
*	1	1	0	1

* Arbitrary

Time diagram

With I and IS



Output variables QS and QSN subject to the trigger input I and D-input IS for S = R = 0 (n is bit number)

Block connections

Block connection	Description	Preassignment	Value range	Attributes
IS	D input	16#0000	WORD	
I	Trigger input	0	0/1	
S	Set	0	0/1	
R	Reset	0	0/1	
QS	Output variable	16#0000	WORD	
QSN	Inverted output variable	16#FFFF	WORD	

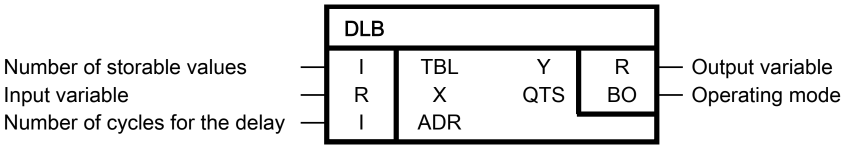
Project data

Can be loaded online	Yes
Special characteristics	-

4.12 DLB Delay element (REAL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Block of the REAL type for the output of an input variable which is delayed by a specifiable number of sampling times.

Operation mode

If the operating mode is QTS = 1, the block contains a delay memory of the TBL variable. The input variable specified at input X is output after a delay as output variable Y. The delay is specified by the integer multiple ADR of the sampling time (time slice in which the block is calculated). When operating mode QTS = 0, the delay memory is not activated. In this case, the input variable specified at input X is output immediately as output variable Y.

Initialization

During initialization, the delay memory is requested for the purpose of acquiring TBL input variables. The delay memory can contain a maximum of 1,000 values. If TBL < 0, TBL is limited to 0. QTS = 1 indicates that the delay memory requested in TBL is available. QTS = 0 indicates that the system was not able to make the memory available, due to a lack of resources, or a TBL value > 1,000 has been defined. In this case, output Y is corrected to input X during cyclic operation.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
TBL	Number of storable values	100	0...1000	
X	Input variable	0.0	REAL	
ADR	Number of cycles for the delay	0	0...1000	
Y	Output variable	0.0	REAL	
QTS	Operating mode	0	0/1	

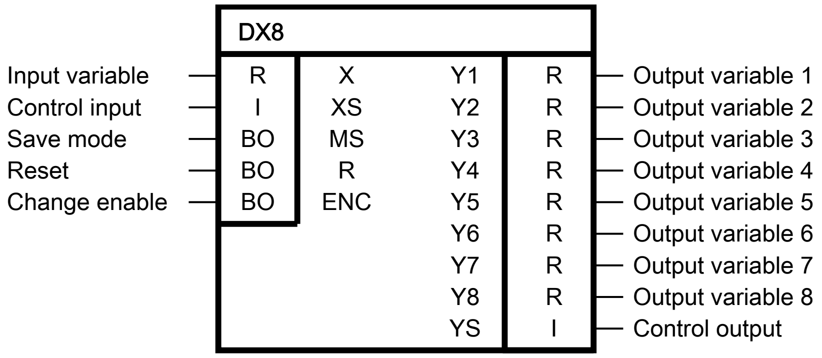
Project data

Can be loaded online	**Yes**
Special characteristics	-

4.13 DX8 Demultiplexer, 8 outputs, cascadable (REAL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Block of the REAL type for demultiplex operation. This block is cascadable.

Operation mode

Depending on ENC, R, MS and XS = 1 to 8, the block switches through its input X to one of the eight selectable outputs Y1 to Y8 (example: XS = 3 means Y3 = X).

When XS = 0 or XS >= 9, none of the block inputs Y1 to Y8 is selected. Non-selected outputs are either set to zero or retain their previous value until the next change.

The following priority sequence applies for the control inputs:

ENC before R before MS

When ENC = 0, all outputs Y1 to Y8 remain unchanged, regardless of R and MS.

When ENC = 1, outputs Y1 to Y8 are enabled for change.

When R = 1, all outputs Y1 to Y8 receive the value 0, irrespective of MS.

When MS = 0 (non-latching mode), all outputs Y1 to Y8 not selected by XS receive the value 0.

When MS = 1 (storing mode), all outputs not selected by XS remain unchanged.

Truth table(s)

ENC	R	MS	XS	Outputs Y1 to Y8
0	*	*	*	The previous values are retained
1	1	*	*	Y1 to Y8 = 0
1	0	0	1 <= XS <= 8	Selected output = X non-selected output = 0
1	0	0	XS = 0 orXS >= 9	Y1 to Y8 = 0
1	0	1	1 <= XS <= 8	Selected output = X non-selected outputs are unchanged
1	0	1	XS = 0 orXS >= 9	All previous values remain unchanged

Cascading

The block output YS must be connected to the block input XS of the following block.

For XS = 0 to 8, YS = 0

When XS > 8: YS = XS-8

(use for cascading)

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
XS	Control input	0	INT	
MS	Save mode	0	0/1	
R	Reset	0	0/1	
ENC	Change enable	0	0/1	
Y1	Output variable 1	0.0	REAL	
Y2	Output variable 2	0.0	REAL	
Y3	Output variable 3	0.0	REAL	
Y4	Output variable 4	0.0	REAL	
Y5	Output variable 5	0.0	REAL	
Y6	Output variable 6	0.0	REAL	
Y7	Output variable 7	0.0	REAL	
Y8	Output variable 8	0.0	REAL	
YS	Control output	0	INT	

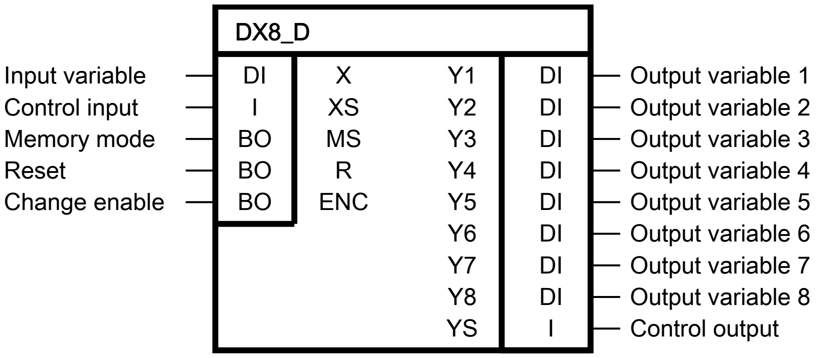
Project data

Can be loaded online	Yes
Special characteristics	-

4.14 DX8_D Demultiplexer, 8 outputs, cascadable (Double integer type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Block of the DOUBLE INTEGER type for demultiplex operation. This block is cascadable.

Operation mode

Depending on ENC, R, MS and XS = 1 to 8, the block switches through its input X to one of the eight selectable outputs Y1 to Y8 (example: XS = 3 means Y3 = X).

When XS = 0 or XS >= 9, none of the block inputs Y1 to Y8 is selected. Non-selected outputs are either set to zero or retain their previous value until the next change.

The following priority sequence applies for the control inputs:

ENC before R before MS

When ENC = 0, all outputs Y1 to Y8 remain unchanged, irrespective of R and MS. When ENC = 1, outputs Y1 to Y8 are enabled for change When R = 1, all outputs Y1 to Y8 receive the value 0, irrespective of M When MS = 0 (non-latching mode), all outputs Y1 to Y8 not selected by XS receive the value When MS = 1 (latching mode), all outputs not selected by XS remain unchanged.

Truth table(s)

ENC	R	MS	XS	Outputs Y1 to Y8
0	*	*	*	The previous values are retained
1	1	*	*	Y1 to Y8 = 0
1	0	0	1 ≤ XS ≤ 8	Selected output = X non-selected output = 0
1	0	0	XS = 0 or XS ≥ 9	Y1 to Y8 = 0
1	0	1	1 ≤ XS ≤ 8	Selected output = X non-selected outputs are unchanged
1	0	1	XS = 0 or XS ≥ 9	All previous values remain unchanged

* Arbitrary

For XS = 0 to 8, YS = 0.When XS > 8: YS = XS - 8(use for cascading).

Cascading

The block output YS must be connected to the block input XS of the following block.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	DINT	
XS	Control input	0	INT	
MS	Memory mode	0	0/1	
R	Reset	0	0/1	
ENC	Change enable	0	0/1	
Y1	Output variable 1	0	DINT	
Y2	Output variable 2	0	DINT	
Y3	Output variable 3	0	DINT	
Y4	Output variable 4	0	DINT	
Y5	Output variable 5	0	DINT	
Y6	Output variable 6	0	DINT	
Y7	Output variable 7	0	DINT	
Y8	Output variable 8	0	DINT	
YS	Control output	0	INT	

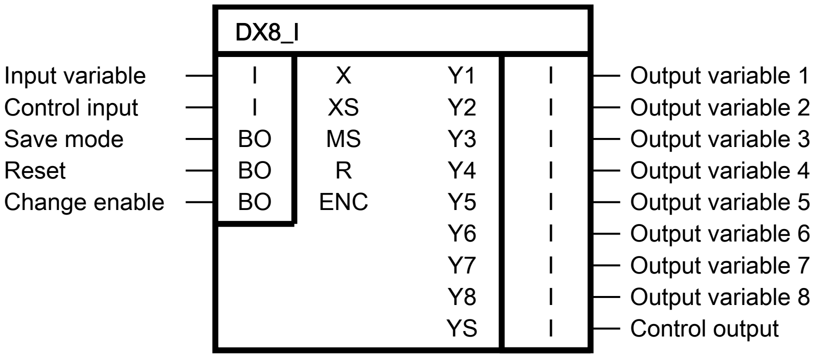
Project data

Can be inserted online	Yes
Special characteristics	-

4.15 DX8_I Demultiplexer, 8 outputs, cascadable (INTEGER type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Block of the INTEGER type for demultiplex operation. This block is cascadable.

Operation mode

Depending on ENC, R, MS and XS = 1 to 8, the block switches through its input X to one of the eight selectable outputs Y1 to Y8 (example: XS = 3 means Y3 = X).

When XS = 0 or XS >= 9, none of the block inputs Y1 to Y8 is selected. Non-selected outputs are either set to zero or retain their previous value until the next change.

The following priority sequence applies for the control inputs:

ENC before R before MS

When ENC = 0, all outputs Y1 to Y8 remain unchanged, regardless of R and MS.

When ENC = 1, outputs Y1 to Y8 are enabled for change.

When R = 1, all outputs Y1 to Y8 receive the value 0, irrespective of MS.

When MS = 0 (non-latching mode), all outputs Y1 to Y8 not selected by XS receive the value 0.

When MS = 1 (storing mode), all outputs not selected by XS remain unchanged.

Truth table(s)

ENC	R	MS	XS	Outputs Y1 to Y8
0	*	*	*	The previous values are retained
1	1	*	*	Y1 to Y8 = 0
1	0	0	1 <= XS <= 8	Selected output = X non-selected output = 0
1	0	0	XS = 0 orXS >= 9	Y1 to Y8 = 0
1	0	1	1 <= XS <= 8	Selected output = X non-selected outputs are unchanged
1	0	1	XS = 0 orXS >= 9	All previous values remain unchanged

* Arbitrary

For XS = 0 to 8, YS = 0.When XS > 8: YS = XS - 8(use for cascading).

Cascading

The block output YS must be connected to the block input XS of the following block.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	INT	
XS	Control input	0	INT	
MS	Save mode	0	0/1	
R	Reset	0	0/1	
ENC	Change enable	0	0/1	
Y1	Output variable 1	0	INT	
Y2	Output variable 2	0	INT	
Y3	Output variable 3	0	INT	
Y4	Output variable 4	0	INT	
Y5	Output variable 5	0	INT	
Y6	Output variable 6	0	INT	
Y7	Output variable 7	0	INT	
Y8	Output variable 8	0	INT	
YS	Control output	0	INT	

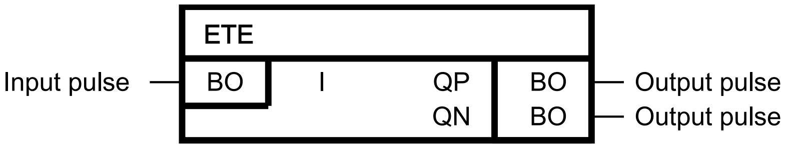
Project data

Can be loaded online	Yes
Special characteristics	-

4.16 ETE Edge evaluator (BOOL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Edge evaluation

Operation mode

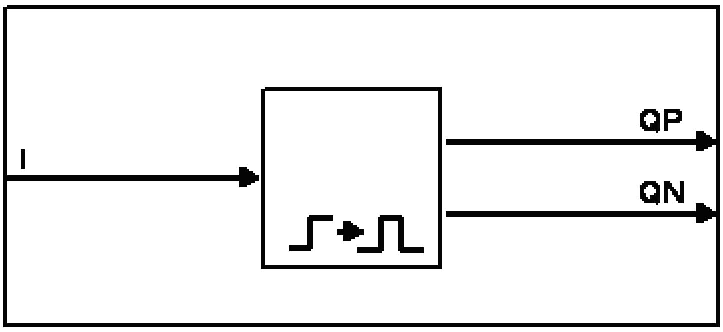
The block detects a signal change at input I. With a positive edge (0→1) at input I, output QP = 1 is set for scan time TA.

With a negative edge (1→0) at input I, output QN = 1 is set for scan time TA.

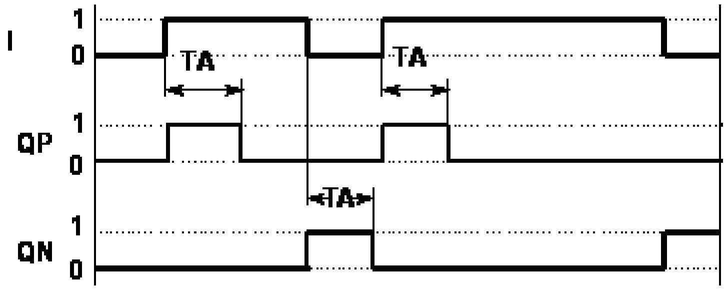
Initialization

The initialization defines the start value for the first cyclic pass. If input I receives the value 1 during initialization from an upstream block, the block cannot detect a positive edge during the first cyclic pass. If input I receives the value 0 during initialization from an upstream block, the block cannot detect a negative edge during the first cyclic pass.

Block diagram



Time diagram



Output pulses QP and QN as a function of scan time TA and input pulse I

Block connections

Block connection	Description	Preassignment	Value range	Attributes
I	Input pulse	0	0/1	
QP	Output pulse	0	0/1	
QN	Output pulse	0	0/1	

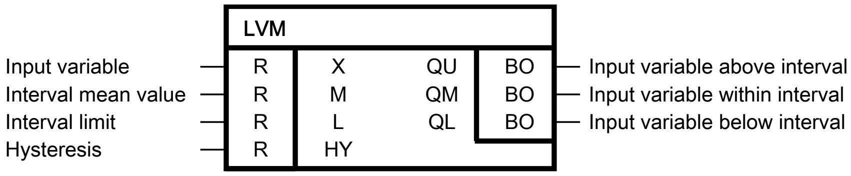
Project data

Can be loaded online	Yes
Special characteristics	-

4.17 LVM Double-sided limit monitor with hysteresis (type BOOL)

☑ SIMOTION ☑ SINAMICS

Symbol


LVM
Input variable
Interval mean value
Interval limit
Hysteresis
R X
R M
R L
R HY
QU BO
QM BO
QL BO
Input variable above interval
Input variable within interval
Input variable below interval

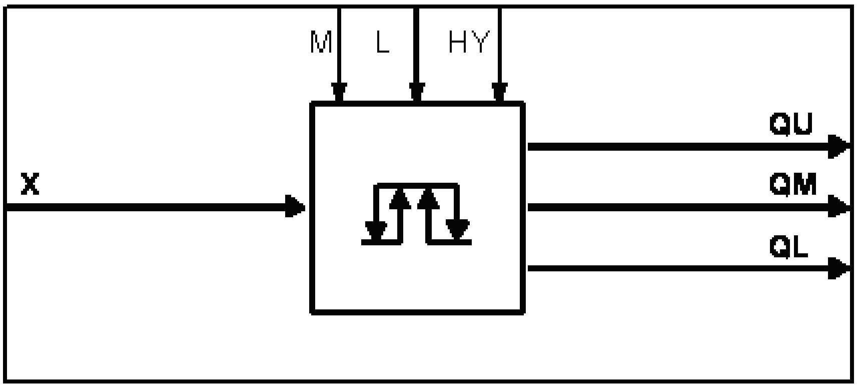
Short description

- The block of the BOOL type monitors an input variable through comparison with selectable reference variables.
- Can be used for monitoring setpoints, actual values and measured values, suppression of frequent switching (chatter)
- The block provides a window discriminator function.

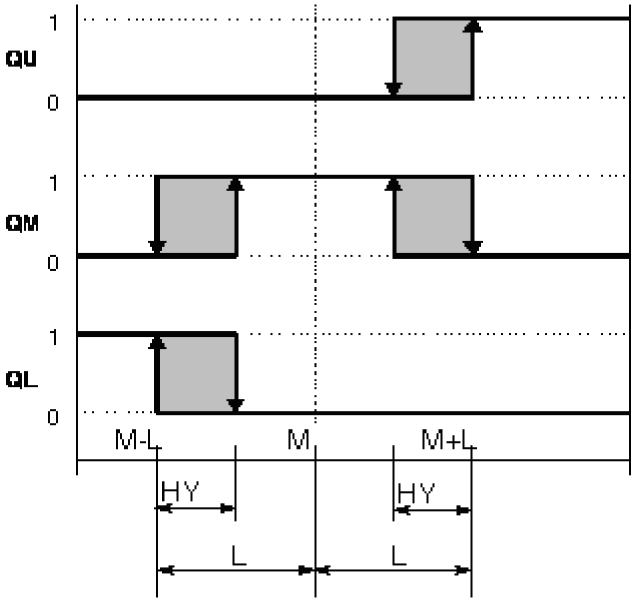
Operation mode

The block calculates an intermediate value based on a transmission characteristic (see Transmission characteristic) with hysteresis. This intermediate value is compared with the interval limits, and the result is output at outputs QU, QM, and QL. The transfer characteristic is configured with the values for the mean value M, the interval limit L and the hysteresis HY.

Block diagram



Transfer function



Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
M	Interval mean value	0.0	REAL	
L	Interval limit	0.0	REAL	
HY	Hysteresis	0.0	REAL	
QU	Input variable above interval	0	0/1	
QM	Input variable within interval	0	0/1	
QL	Input variable below interval	0	0/1	

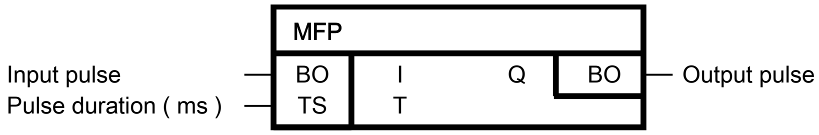
Project data

Can be loaded online	Yes
Special characteristics	-

4.18 MFP Pulse generator (type BOOL)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

- Timer for generating a pulse with a fixed duration
- Used as a pulse-contracting or pulse-stretching monoflop

Operation mode

The rising edge of a pulse at input I sets output Q to 1 for the pulse duration T. The pulse generator cannot be retriggered. When T=0, a pulse duration of 1 cycle is active.

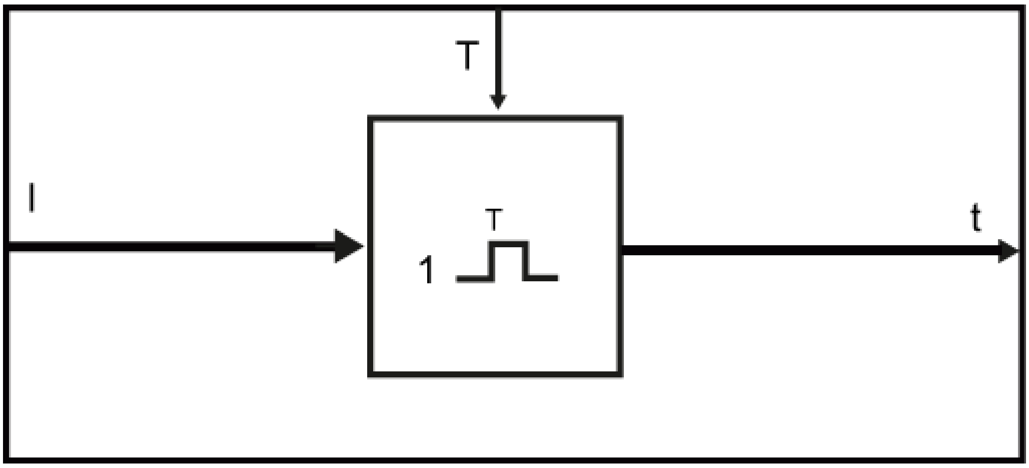
Initialization

The initialization defines the start value for the first cyclic pass.

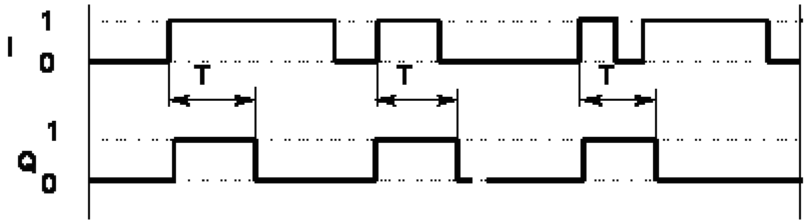
If input I receives the value 1 during initialization from the upstream block output, the block cannot detect a positive edge during the first cyclic pass.

If output Q receives the default value 1, output Q = 1 is set after initialization for the pulse duration T.

Block diagram



Time diagram



Output pulse Q as a function of pulse duration T
and input pulse I

Block connections

Block connection	Description	Preassignment	Value range	Attributes
I	Input pulse	0	0/1	
T	Pulse duration (ms)	0	SDTIME	
Q	Output pulse	0	0/1	

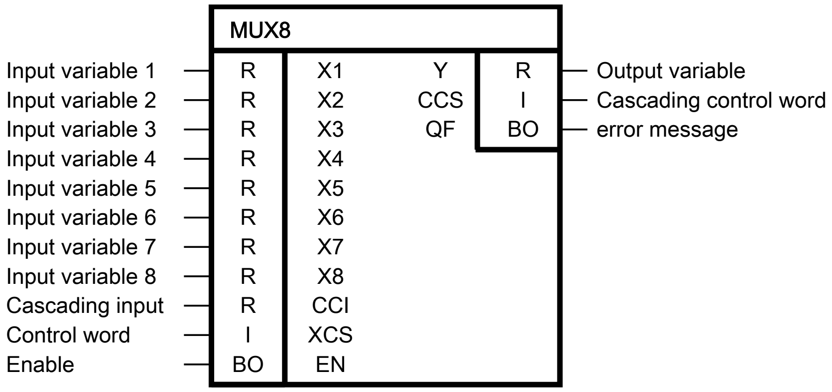
Project data

Can be loaded online	Yes
Special characteristics	-

4.19 MUX8 Multiplexer, cascadable (REAL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Block of the REAL type for 8-fold multiplex operation. This block is cascadable.

Operation mode

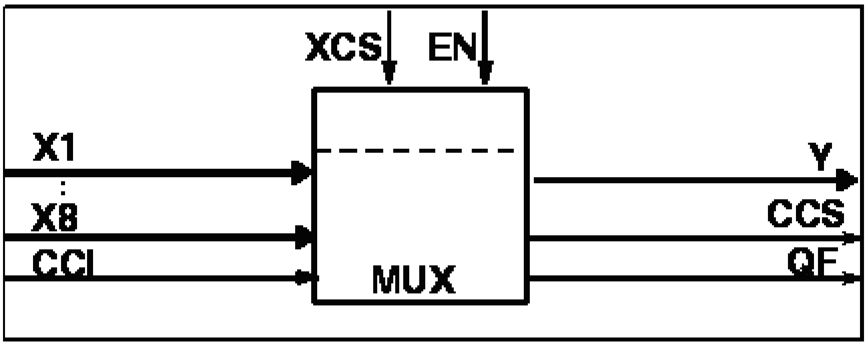
This block outputs the value of the cascading input CCI at output Y as long as the enable input EN is logic 0.

When EN is logic 1, one of the input variables X1 to X8 is switched through to output Y as long as the 16-bit control word XCS assumes a value between 1 and 8.

If the value of the input XCS > 8, output Y assumes the value 0, and output QF becomes logic 1. The cascading control word assumes the value CCS = XCS-8, see truth table.

The outputs Y, CCS, and QF can be used to cascade the blocks. In this case, output Y of the first block is connected to input CCI of the downstream multiplexer, output CCS to the following XCS, and output QF to the following input EN.

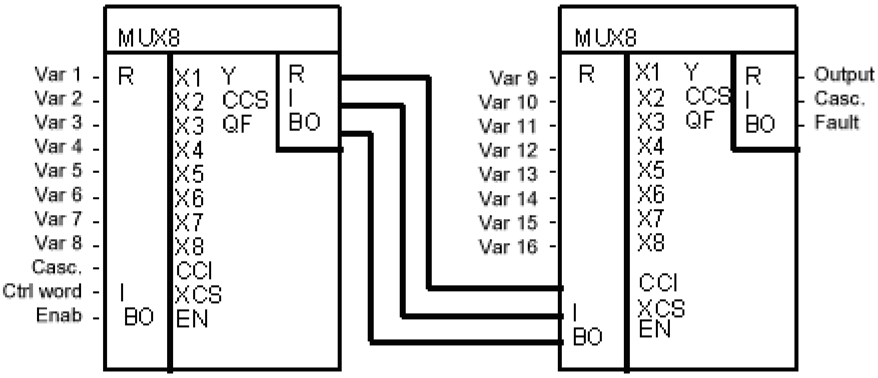
Block diagram



Truth table(s)

EN	XCS	Y	CSS	QF
0	Arbitrary	CCI	0	0
1	0	0	0	1
1	1	X1	0	0
1	2	X2	0	0
1	3	X3	0	0
1	4	X4	0	0
1	5	X5	0	0
1	6	X6	0	0
1	7	X7	0	0
1	8	X8	0	0
1	>8	0	XCS-8	1

Cascading



Block connections

Block connection	Description	Preassignment	Value range	Attributes
X1	Input variable 1	0.0	REAL	
X2	Input variable 2	0.0	REAL	
X3	Input variable 3	0.0	REAL	
X4	Input variable 4	0.0	REAL	
X5	Input variable 5	0.0	REAL	
X6	Input variable 6	0.0	REAL	
X7	Input variable 7	0.0	REAL	
X8	Input variable 8	0.0	REAL	
CCI	Cascading input	0.0	REAL	
XCS	Control word	0	0…32767	
EN	Enable	0	0/1	
Y	Output variable	0.0	REAL	
CCS	Cascading control word	0	0…32767	
QF	error message	0	0/1	

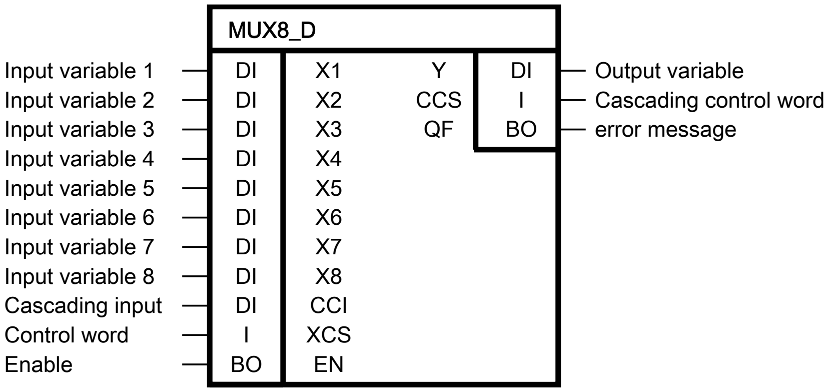
Project data

Can be loaded online	Yes
Special characteristics	-

4.20 MUX8_D Multiplexer, cascadable (double integer type)

☑ SIMOTION ☑ SINAMICS

Symbol



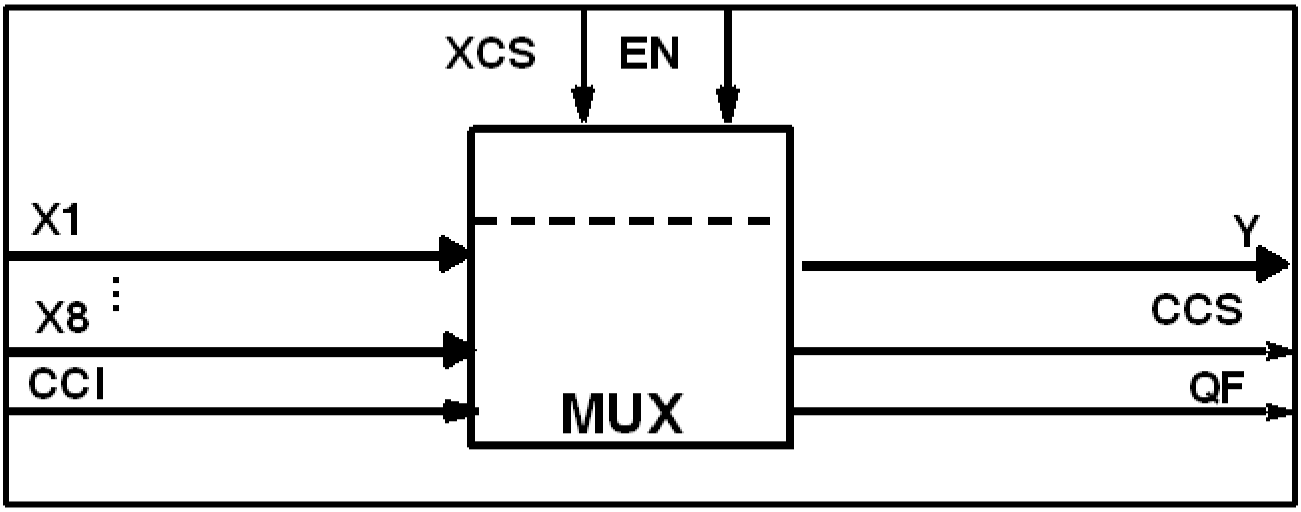
Short description

Block of the DOUBLE INTEGER type for 8-fold multiplex operation. This block is cascadable.

Operation mode

The block outputs the value of the cascading input CCI at output Y as long as the enable input EN is logic 0. When EN is logic 1, one of the input variables X1 to X8 is switched through to output Y as long as the 16-bit control word XCS assumes a value between 1 and 8. If the value of the input XCS > 8, output Y assumes the value 0 and output QF becomes logic 1. The cascading control word assumes the value CCS = XCS-8, see truth table. The outputs Y, CCS and QF can be used to cascade the blocks. In this case, output Y of the first block is connected to input CCI of the downstream multiplexer, output CCS to the following XCS, and output QF to the following input EN.

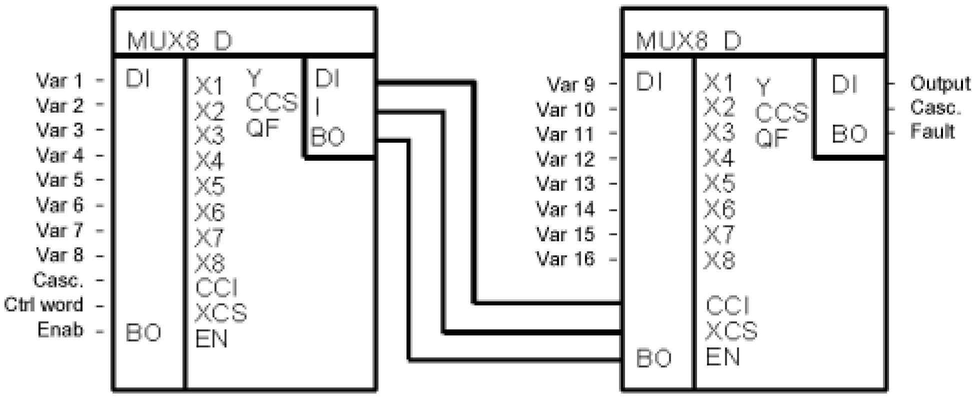
Block diagram



Truth table(s)

EN	XCS	Y	CSS	QF
0	Arbitrary	CCI	0	0
1	0	0	0	1
1	1	X1	0	0
1	2	X2	0	0
1	3	X3	0	0
1	4	X4	0	0
1	5	X5	0	0
1	6	X6	0	0
1	7	X7	0	0
1	8	X8	0	0
1	>8	0	XCS-8	1

Cascading



Block connections

Block connection	Description	Preassignment	Value range	Attributes
X1	Input variable 1	0	DINT	
X2	Input variable 2	0	DINT	
X3	Input variable 3	0	DINT	
X4	Input variable 4	0	DINT	
X5	Input variable 5	0	DINT	
X6	Input variable 6	0	DINT	
X7	Input variable 7	0	DINT	
X8	Input variable 8	0	DINT	
CCI	Cascading input	0	DINT	
XCS	Control word	0	0…32767	
EN	Enable	0	0/1	
Y	Output variable	0	DINT	
CCS	Cascading control word	0	0…32767	
QF	error message	0	0/1	

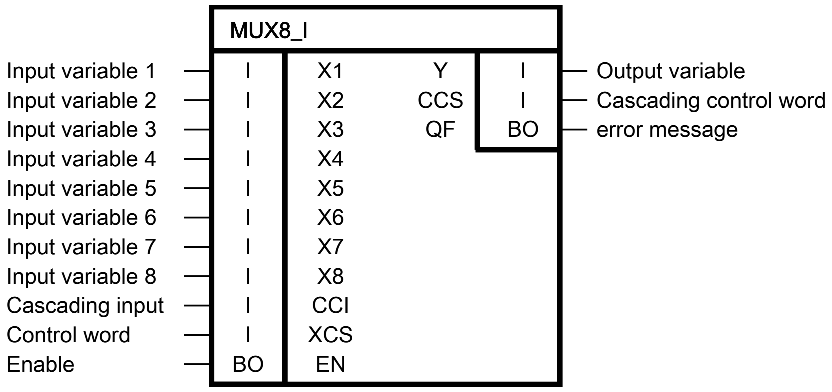
Project data

Can be inserted online	Yes
Special characteristics	-

4.21 MUX8_I Multiplexer, cascadable (INTEGER type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Block of the INTEGER type for 8-fold multiplex operation. This block is cascadable.

Operation mode

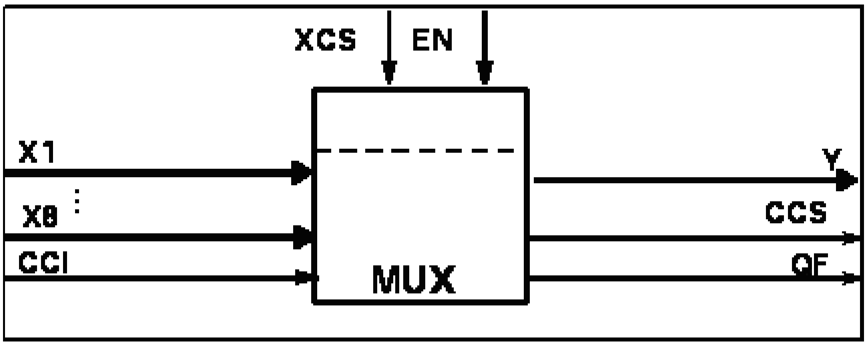
This block outputs the value of the cascading input CCI at output Y as long as the enable input EN is logic 0.

When EN is logic 1, one of the input variables X1 to X8 is switched through to output Y as long as the 16-bit control word XCS assumes a value between 1 and 8.

If the value of the input XCS > 8, output Y assumes the value 0, and output QF becomes logic 1. The cascading control word assumes the value CCS = XCS-8, see truth table.

The outputs Y, CCS, and QF can be used to cascade the blocks. In this case, output Y of the first block is connected to input CCI of the downstream multiplexer, output CCS to the following XCS, and output QF to the following input EN.

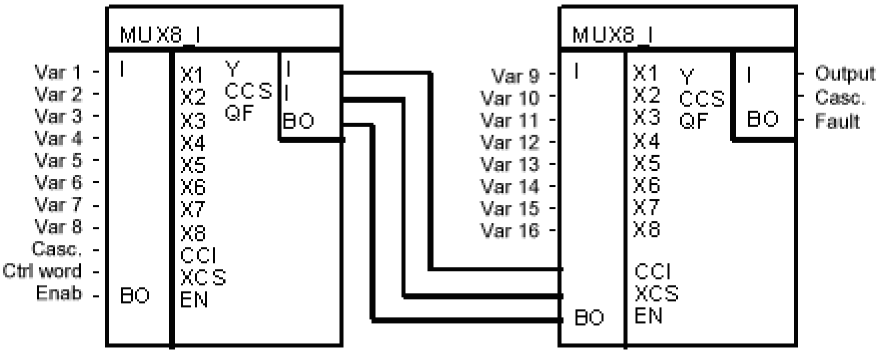
Block diagram



Truth table(s)

EN	XCS	Y	CSS	QF
0	Arbitrary	CCI	0	0
1	0	0	0	1
1	1	X1	0	0
1	2	X2	0	0
1	3	X3	0	0
1	4	X4	0	0
1	5	X5	0	0
1	6	X6	0	0
1	7	X7	0	0
1	8	X8	0	0
1	>8	0	XCS-8	1

Cascading



Block connections

Block connection	Description	Preassignment	Value range	Attributes
X1	Input variable 1	0	INT	
X2	Input variable 2	0	INT	
X3	Input variable 3	0	INT	
X4	Input variable 4	0	INT	
X5	Input variable 5	0	INT	
X6	Input variable 6	0	INT	
X7	Input variable 7	0	INT	
X8	Input variable 8	0	INT	
CCI	Cascading input	0	INT	
XCS	Control word	0	0…32767	
EN	Enable	0	0/1	
Y	Output variable	0	INT	
CCS	Cascading control word	0	0…32767	
QF	error message	0	0/1	

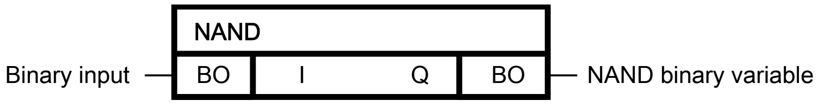
Project data

Can be loaded online	Yes
Special characteristics	-

4.22 NAND Logic AND operation (BOOL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

NAND block with up to four inputs of the BOOL type

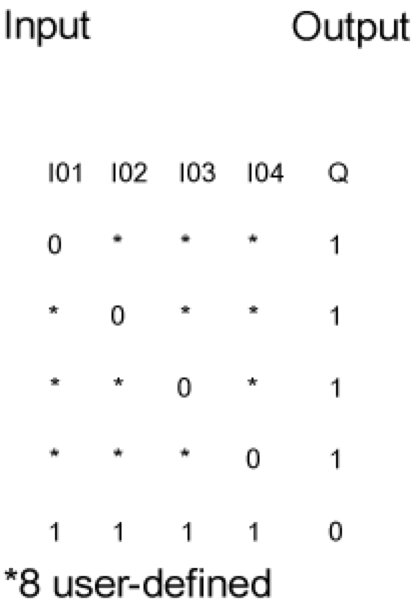
Operation mode

The block combines the binary values at the inputs I 1-4 to a logic AND, inverts the result and outputs it at binary output Q.

$$Q = \overline{I_{01} \wedge \cdots \wedge I_{nn}}$$

Output Q = 0, when the value 1 is present at all generic inputs I1 to I4. In all other cases, output Q = 1.

Truth table(s)



Input Output

I01	I02	I03	I04	Q
0	*	*	*	1
*	0	*	*	1
*	*	0	*	1
*	*	*	0	1
1	1	1	1	0

*8 user-defined

Block connections

Block connection	Description	Preassignment	Value range	Attributes
I	Binary input	1	0/1	
Q	NAND binary variable	0	0/1	

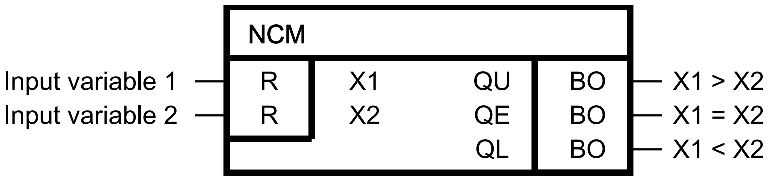
Project data

Can be loaded online	**Yes**
Special characteristics	I comprises up to four inputs (I1 to I4)

4.23 NCM Numeric comparator (REAL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Block for compare operations of two numeric variables of the REAL type

Operation mode

The input variables X1 and X2 are compared and one of binary outputs QU, QE, or QL is set depending on the result of the compare operation.

Truth table(s)

Comparison of input variables	Output signals	Output signals Y	Output signals Y
	QU	QE	QL
X1 > X2	1	0	0
X1 = X2	0	1	0
X1 < X2	0	0	1

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X1	Input variable 1	0	REAL	
X2	Input variable 2	0	REAL	
QU	X1 > X2	0	0/1	
QE	X1 = X2	1	0/1	
QL	X1 < X2	0	0/1	

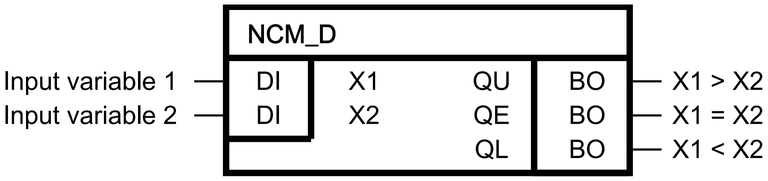
Project data

Can be loaded online	Yes
Special characteristics	If the execution group of the DCC chart is set to "Do not calculate", the default value is set at the output.

4.24 NCM_D Numeric comparator (DOUBLE_INTEGER type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Block for compare operations of two numeric variables of the DOUBLE INTEGER type

Operation mode

The input variables X1 and X2 are compared and one of binary outputs QU, QE, or QL is set depending on the result of the compare operation.

Truth table(s)

Comparison of input variables	Output signals	Output signals Y	Output signals Y
	QU	QE	QL
X1 > X2	1	0	0
X1 = X2	0	1	0
X1 < X2	0	0	1

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X1	Input variable 1	0	DINT	
X2	Input variable 2	0	DINT	
QU	X1 > X2	0	0/1	
QE	X1 = X2	1	0/1	
QL	X1 < X2	0	0/1	

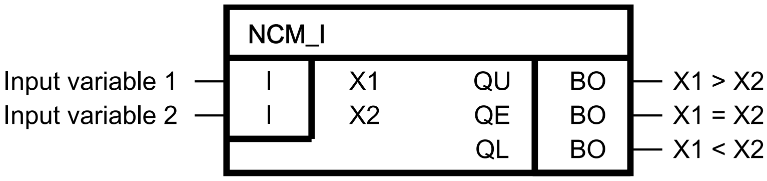
Project data

Can be loaded online	Yes
Special characteristics	If the execution group of the DCC chart is set to "Do not calculate", the default value is set at the output.

4.25 NCM_I Numeric comparator (INTEGER type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Block for compare operations of two numeric variables of the INTEGER type

Operation mode

The input variables X1 and X2 are compared and one of binary outputs QU, QE, or QL is set depending on the result of the compare operation.

Truth table(s)

Comparison of input variables	Output signals	Output signals Y	Output signals Y
	QU	QE	QL
X1 > X2	1	0	0
X1 = X2	0	1	0
X1 < X2	0	0	1

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X1	Input variable 1	0	INT	
X2	Input variable 2	0	INT	
QU	X1 > X2	0	0/1	
QE	X1 = X2	1	0/1	
QL	X1 < X2	0	0/1	

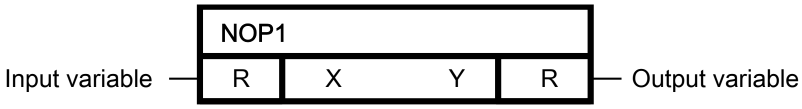
Project data

Can be loaded online	Yes
Special characteristics	If the execution group of the DCC chart is set to "Do not calculate", the default value is set at the output.

4.26 NOP1 Dummy blocks (REAL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

The block of the REAL type is used as dummy block (No Operation).

Operation mode

The block outputs the value present at input X without change at output Y. This is a so-called DUMMY or No Operation block.

Initialization

The block outputs the value present at input X without change at output Y, thus providing a common constant for the initialization of several other blocks.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
Y	Output variable	0.0	REAL	

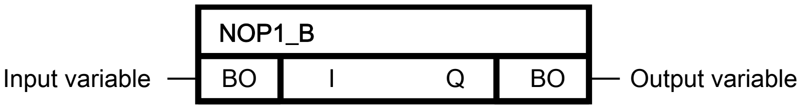
Project data

Can be loaded online	Yes
Special characteristics	-

4.27 NOP1_B Dummy block (BOOL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

The block of the BOOL type is used as dummy block (No Operation).

Operation mode

The block outputs the value present at input I without change at output Q. This is a so-called DUMMY or No Operation block.

Initialization

The block outputs the value present at input I without change at output Q, thus providing a common constant for the initialization of several other blocks.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
I	Input variable	0	0/1	
Q	Output variable	0	0/1	

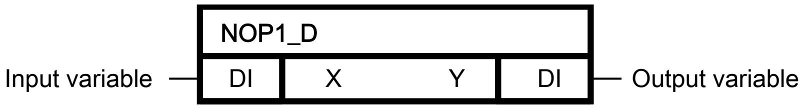
Project data

Can be loaded online	Yes
Special characteristics	-

4.28 NOP1_D Dummy block (DOUBLE INTEGER type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

The block of the DOUBLE INTEGER type is used as dummy block (No Operation).

Operation mode

The block outputs the value present at input X without change at output Y. This is a so-called DUMMY or No Operation block.

Initialization

The block outputs the value present at input X without change at output Y, thus providing a common constant for the initialization of several other blocks.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	DINT	
Y	Output variable	0	DINT	

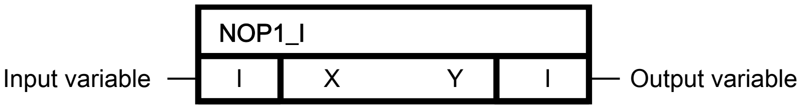
Project data

Can be loaded online	Yes
Special characteristics	-

4.29 NOP1_I Dummy block (INT type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

The block of the INT type is used as dummy block (No Operation).

Operation mode

The block outputs the value present at input X without change at output Y. This is a so-called DUMMY or No Operation block.

Initialization

The block outputs the value present at input X without change at output Y, thus providing a common constant for the initialization of several other blocks.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	INT	
Y	Output variable	0	INT	

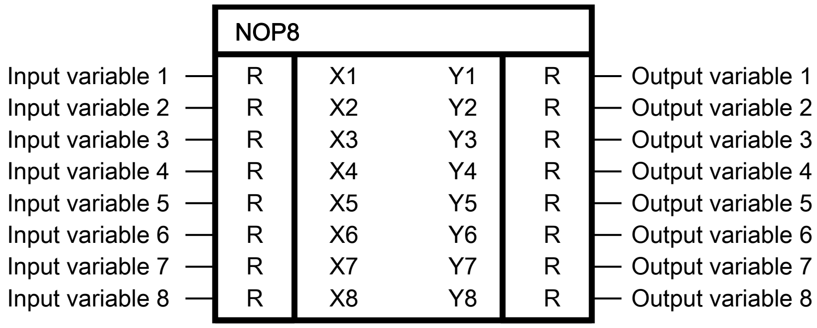
Project data

Can be loaded online	Yes
Special characteristics	-

4.30 NOP8 Dummy blocks (REAL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

The block of the REAL type is used as dummy block (No Operation).

Operation mode

The block outputs the values present at inputs X1-X8 without change at outputs Y1 to Y8. This is a so-called DUMMY or No Operation block.

Initialization

The block outputs the values present at inputs X1-X8 without change at outputs Y1 to Y8, thus providing a common constant for the initialization of several other blocks.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X1	Input variable 1	0.0	REAL	
X2	Input variable 2	0.0	REAL	
X3	Input variable 3	0.0	REAL	
X4	Input variable 4	0.0	REAL	
X5	Input variable 5	0.0	REAL	
X6	Input variable 6	0.0	REAL	
X7	Input variable 7	0.0	REAL	
X8	Input variable 8	0.0	REAL	
Y1	Output variable 1	0.0	REAL	
Y2	Output variable 2	0.0	REAL	
Y3	Output variable 3	0.0	REAL	
Y4	Output variable 4	0.0	REAL	
Y5	Output variable 5	0.0	REAL	
Y6	Output variable 6	0.0	REAL	
Y7	Output variable 7	0.0	REAL	
Y8	Output variable 8	0.0	REAL	

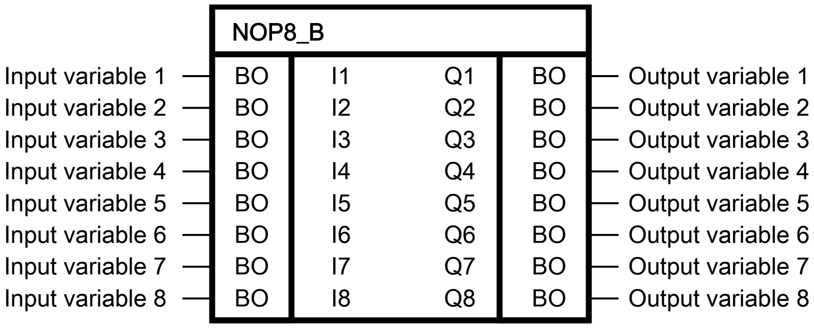
Project data

Can be loaded online	Yes
Special characteristics	-

4.31 NOP8_B Dummy blocks (BOOL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

The block of the BOOL type is used as dummy block (No Operation).

Operation mode

The block outputs the values present at inputs I1-I8 without change at outputs Q1 to Q8. This is a so-called DUMMY or No Operation block.

Initialization

The block outputs the values present at inputs I1-I8 without change at outputs Q1 to Q8, thus providing a common constant for the initialization of several other blocks.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
I1	Input variable 1	0	0/1	
I2	Input variable 2	0	0/1	
I3	Input variable 3	0	0/1	
I4	Input variable 4	0	0/1	
I5	Input variable 5	0	0/1	
I6	Input variable 6	0	0/1	
I7	Input variable 7	0	0/1	
I8	Input variable 8	0	0/1	
Q1	Output variable 1	0	0/1	
Q2	Output variable 2	0	0/1	
Q3	Output variable 3	0	0/1	
Q4	Output variable 4	0	0/1	
Q5	Output variable 5	0	0/1	
Q6	Output variable 6	0	0/1	
Q7	Output variable 7	0	0/1	
Q8	Output variable 8	0	0/1	

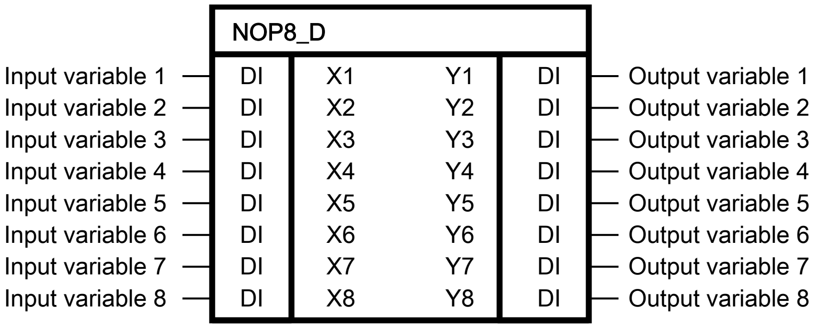
Project data

Can be loaded online	Yes
Special characteristics	-

4.32 NOP8_D Dummy blocks (DOUBLE INTEGER type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

The block of the DOUBLE INTEGER type is used as dummy block (No Operation).

Operation mode

The block outputs the values present at inputs X1-X8 without change at outputs Y1 to Y8. This is a so-called DUMMY or No Operation block.

Initialization

The block outputs the values present at inputs X1-X8 without change at outputs Y1 to Y8, thus providing a common constant for the initialization of several other blocks.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X1	Input variable 1	0	DINT	
X2	Input variable 2	0	DINT	
X3	Input variable 3	0	DINT	
X4	Input variable 4	0	DINT	
X5	Input variable 5	0	DINT	
X6	Input variable 6	0	DINT	
X7	Input variable 7	0	DINT	
X8	Input variable 8	0	DINT	
Y1	Output variable 1	0	DINT	
Y2	Output variable 2	0	DINT	
Y3	Output variable 3	0	DINT	
Y4	Output variable 4	0	DINT	
Y5	Output variable 5	0	DINT	
Y6	Output variable 6	0	DINT	
Y7	Output variable 7	0	DINT	
Y8	Output variable 8	0	DINT	

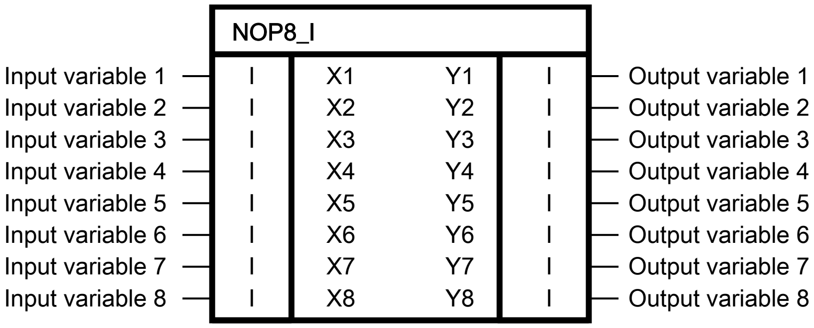
Project data

Can be loaded online	Yes
Special characteristics	-

4.33 NOP8_I Dummy blocks (INTEGER type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

The block of the INTEGER type is used as dummy block (No Operation).

Operation mode

The block outputs the values present at inputs X1-X8 without change at outputs Y1 to Y8. This is a so-called DUMMY or No Operation block.

Initialization

The block outputs the values present at inputs X1-X8 without change at outputs Y1 to Y8, thus providing a common constant for the initialization of several other blocks.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X1	Input variable 1	0	INT	
X2	Input variable 2	0	INT	
X3	Input variable 3	0	INT	
X4	Input variable 4	0	INT	
X5	Input variable 5	0	INT	
X6	Input variable 6	0	INT	
X7	Input variable 7	0	INT	
X8	Input variable 8	0	INT	
Y1	Output variable 1	0	INT	
Y2	Output variable 2	0	INT	
Y3	Output variable 3	0	INT	
Y4	Output variable 4	0	INT	
Y5	Output variable 5	0	INT	
Y6	Output variable 6	0	INT	
Y7	Output variable 7	0	INT	
Y8	Output variable 8	0	INT	

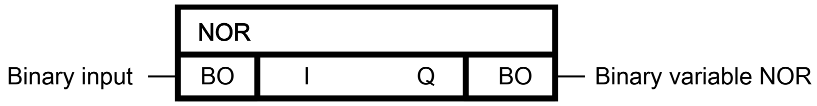
Project data

Can be loaded online	Yes
Special characteristics	-

4.34 NOR Logic OR operation (BOOL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

NOR block with up to four inputs of the BOOL type

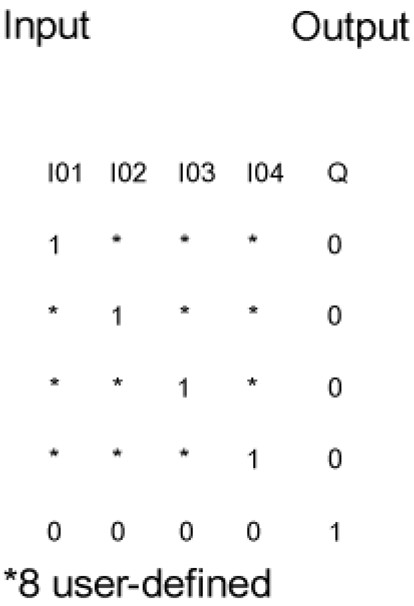
Operation mode

The block combines the binary values at the inputs I 1-4 to a logic OR and outputs the result at its binary output Q.

$$Q = \overline{I_{01} \vee \ldots \vee I_{nn}}$$

Output Q = 1, when the value 0 is present at all inputs I1 to I4. In all other cases, output Q = 0

Truth table(s)



Input				Output
I01	I02	I03	I04	Q
1	*	*	*	0
*	1	*	*	0
*	*	1	*	0
*	*	*	1	0
0	0	0	0	1

*8 user-defined

Block connections

Block connection	Description	Preassignment	Value range	Attributes
I	Binary input	0	0/1	
Q	Binary variable NOR	1	0/1	

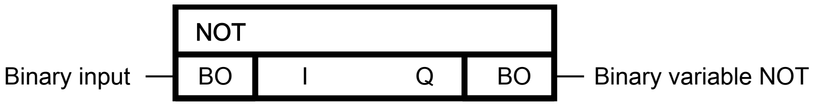
Project data

Can be loaded online	Yes
Special characteristics	I comprises up to four connections (I1 to I4)

4.35 NOT Inverter (BOOL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

BOOL-type inverter

Operation mode

The block inverts the binary variable at input I and outputs the result at output Q.

$$Q = \overline{I}$$

Truth table(s)

Input 1	Output Q
1	0

Block connections

Block connection	Description	Preassignment	Value range	Attributes
I	Binary input	0	0/1	
Q	Binary variable NOT	1	0/1	

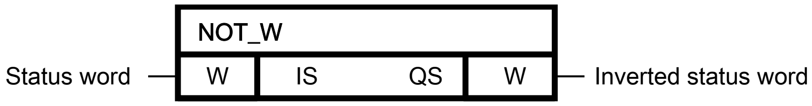
Project data

Can be loaded online	Yes
Special characteristics	-

4.36 NOT_W Status word inverter (WORD type)

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

- Inverter for WORD-type status word
- One's complement formation of IS

Operation mode

16 binary states are combined in a status word.

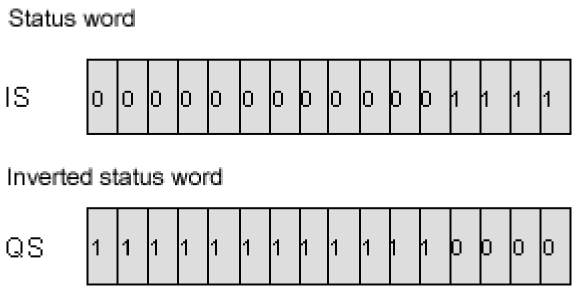
The block inverts the status word IS bit-by-bit and outputs it at output QS.

The following applies for the bit k of the inverted status word:

$$QS_k = \overline{IS_k}$$

Complement formation

Example: IS = 15 -> QS = -16



Status word

IS	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Inverted status word

QS	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0

Block connections

Block connection	Description	Preassignment	Value range	Attributes
IS	Status word	16#0000	WORD	
QS	Inverted status word	16#FFFF	WORD	

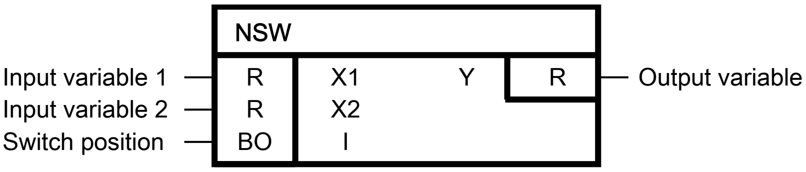
Project data

Can be loaded online	Yes
Special characteristics	-

4.37 NSW Numeric change-over switch (REAL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

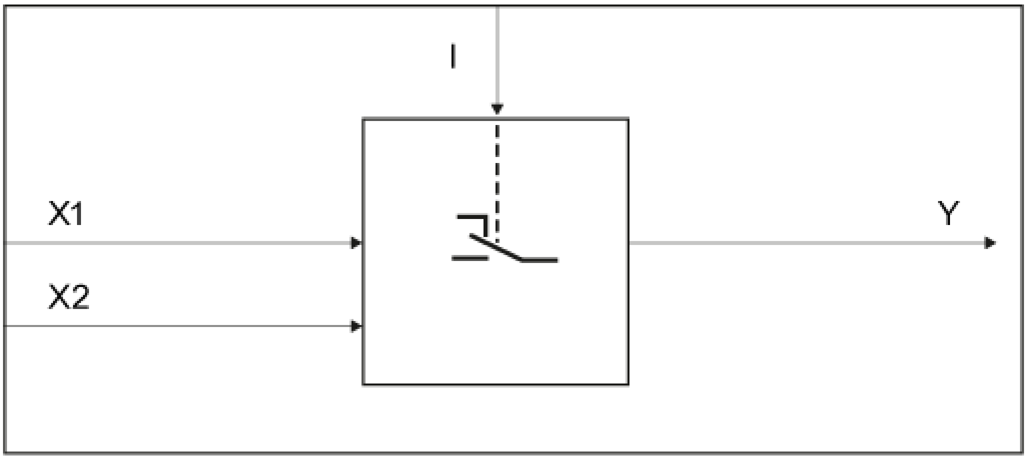
The block switches one of two numeric input variables (REAL type) to the output

Operation mode

If input I = 0, then X1 is given to output Y.

If input I = 1, then X2 is given to output Y.

Block diagram



Truth table(s)

Switch position 1	Output variable Y
0	Y = X1
1	Y = X2

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X1	Input variable 1	0	REAL	
X2	Input variable 2	0	REAL	
I	Switch position	0	0/1	
Y	Output variable	0	REAL	

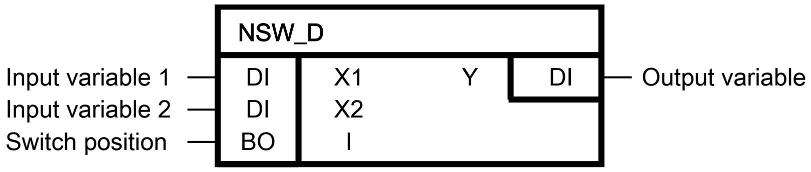
Project data

Can be loaded online	Yes
Special characteristics	-

4.38 NSW_D Numeric change-over switch (DOUBLE INTEGER type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

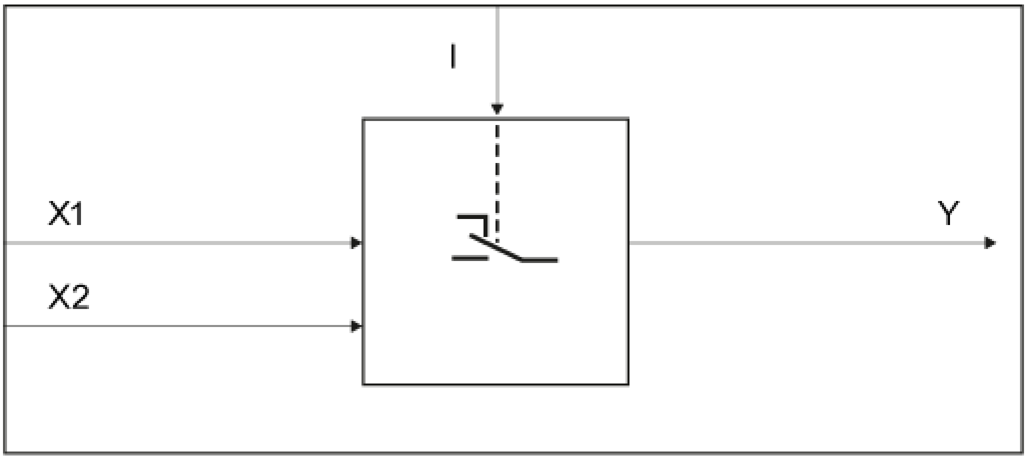
The block switches one of two numeric input variables (DOUBLE INTEGER type) to the output

Operation mode

If input I = 0, then X1 is given to output Y.

If input I = 1, then X2 is given to output Y.

Block diagram



Truth table(s)

Switch position 1	Output variable Y
0	Y = X1
1	Y = X2

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X1	Input variable 1	0	DINT	
X2	Input variable 2	0	DINT	
I	Switch position	0	0/1	
Y	Output variable	0	DINT	

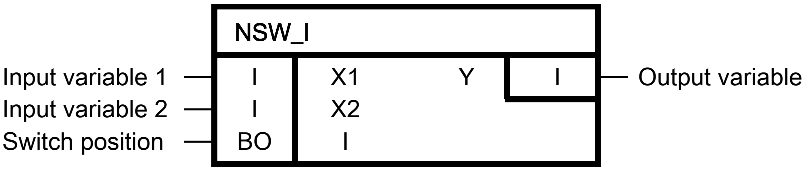
Project data

Can be loaded online	Yes
Special characteristics	-

4.39 NSW_I Numeric change-over switch (INTEGER type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

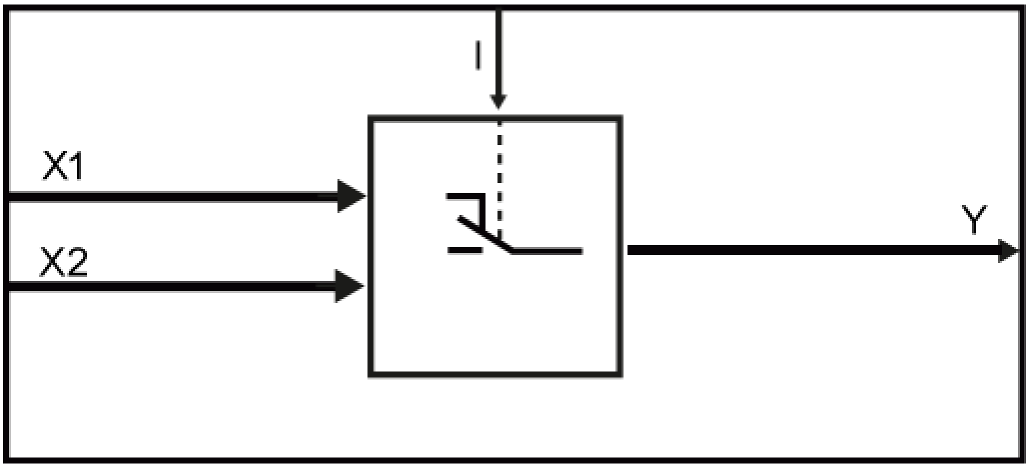
The block switches one of two numeric input variables (INTEGER type) to the output

Operation mode

If input I = 0, then X1 is given to output Y.

If input I = 1, then X2 is given to output Y.

Block diagram



Truth table(s)

Switch position 1	Output variable Y
0	Y = X1
1	Y = X2

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X1	Input variable 1	0	INT	
X2	Input variable 2	0	INT	
I	Switch position	0	0/1	
Y	Output variable	0	INT	

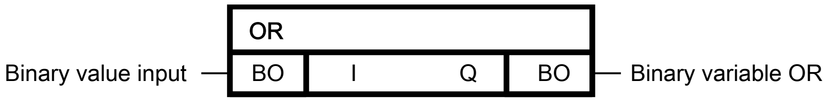
Project data

Can be loaded online	Yes
Special characteristics	-

4.40 OR Logic OR operation (BOOL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

OR block with up to four inputs of the BOOL type

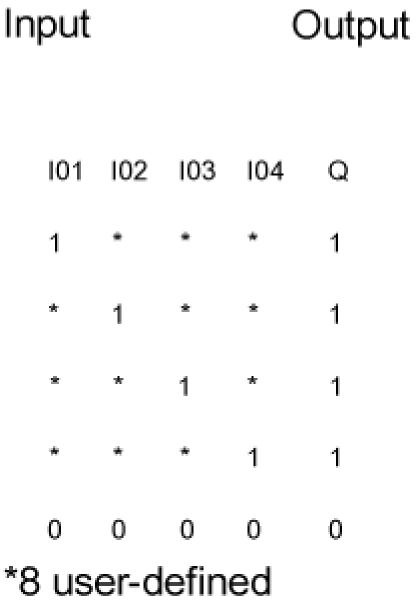
Operation mode

The block combines the binary values at the inputs I 1-4 to a logic OR (disjunction) and outputs the result at its binary output Q.

$Q = I_{01}v...vI_{04}$

Output Q = 0, when the value 0 is present at all inputs I1 to I4. In all other cases, output Q = 1.

Truth table(s)



Input Output

I01	I02	I03	I04	Q
1	*	*	*	1
*	1	*	*	1
*	*	1	*	1
*	*	*	1	1
0	0	0	0	0

*8 user-defined

Block connections

Block connection	Description	Preassignment	Value range	Attributes
I	Binary value input	0	0/1	
Q	Binary variable OR	0	0/1	

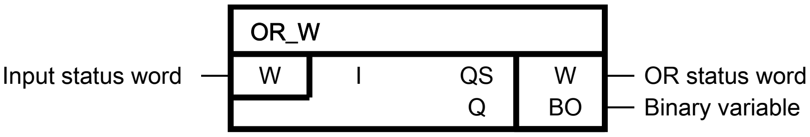
Project data

Can be loaded online	Yes
Special characteristics	I comprises up to four connections (I1 to I4)

4.41 OR_W Logic OR operation (WORD type)

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

OR block with up to four inputs of the WORD type

Operation mode

16 binary states are combined in a status word.

The block combines the status words I1 to I4 bit-by-bit according to the logic OR function.

The result is given to the block output QS (status word OR).

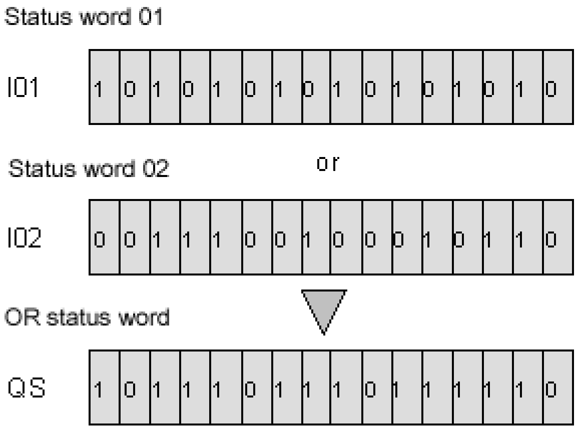
The following applies for bit k of status word OR:

QS_k = $I02_k$v $I02_k$, k = 1...16

A bit of the OR status word is equal to 1 when at least one of the equivalent bits on the block inputs I1 to I4 is equal to 1.

The binary output Q is 1 if at least one bit of the status word OR is equal to 1.

Following state diagram (for 3 inputs)



I01	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
I02	0	0	1	1	1	0	0	1	0	0	0	1	0	1	1	0
QS	1	0	1	1	1	0	1	1	1	0	1	1	1	1	1	0

Block connections

Block connection	Description	Preassignment	Value range	Attributes
I	Input status word	16#0000	WORD	
QS	OR status word	16#0000	WORD	
Q	Binary variable	0	0/1	

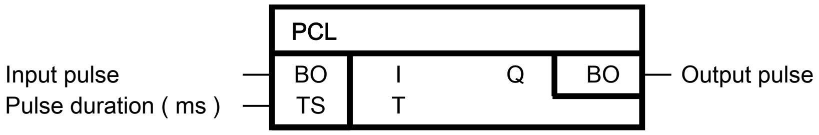
Project data

Can be loaded online	Yes
Special characteristics	I comprises up to four connections (I1 to I4)

4.42 PCL Pulse shortening device (BOOL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Timer for limiting the pulse duration

Operation mode

The rising edge of a pulse at input I sets output Q to 1. Output Q becomes 0 when input I becomes 0 (input I = 0) or pulse duration T has expired. When T=0, a pulse duration of 1 cycle is active.

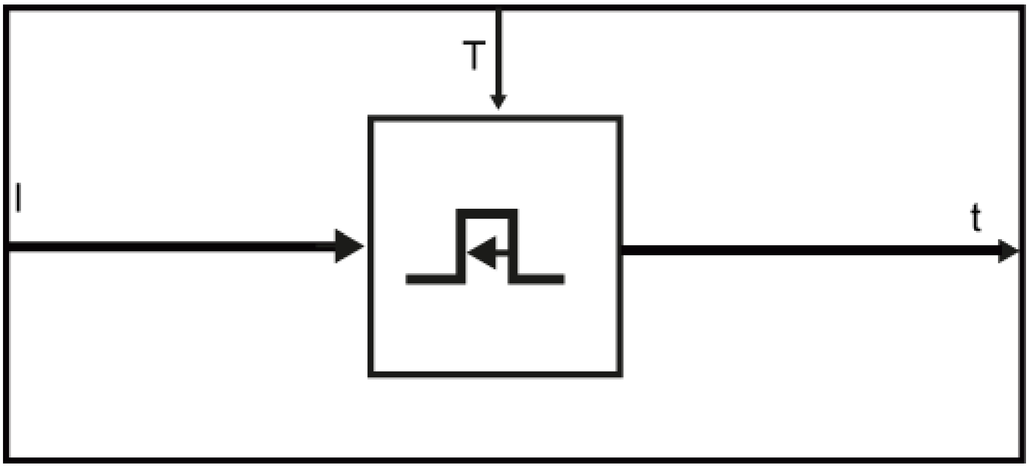
Initialization

The initialization defines the start value for the first cyclic pass.

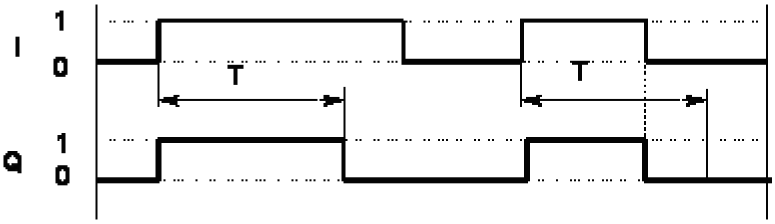
If input I receives the value 1 during initialization from the upstream block output, the block cannot detect a positive edge during the first cyclic pass.

If output Q receives the default value 1, output Q = 1 is set after initialization for the pulse duration T.

Block diagram



Time diagram



Output pulse Q as a function of pulse duration T and input pulse I

Block connections

Block connection	Description	Preassignment	Value range	Attributes
I	Input pulse	0	0/1	
T	Pulse duration (ms)	0	SDTIME	
Q	Output pulse	0	0/1	

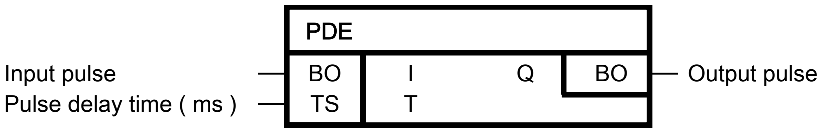
Project data

Can be loaded online	Yes
Special characteristics	-

4.43 PDE On-delay device (BOOL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

BOOL-type timer with on-delay

Operation mode

The pulse delay time at the input T is taken over with the rising edge at input I. After this time has elapsed, output Q is set to 1.

Output Q becomes 0 when I becomes 0 (I = 0).

If the duration of input pulse I is less than pulse delay time T, then Q remains at 0.

If time T is so large that the maximum displayable internal value (T/ta as 32-bit value, where ta = sampling time) is exceeded, limitation is performed to the maximum value (e.g when ta=1 ms approx. 50 days).

When T=0, a pulse delay time of 1 cycle is active.

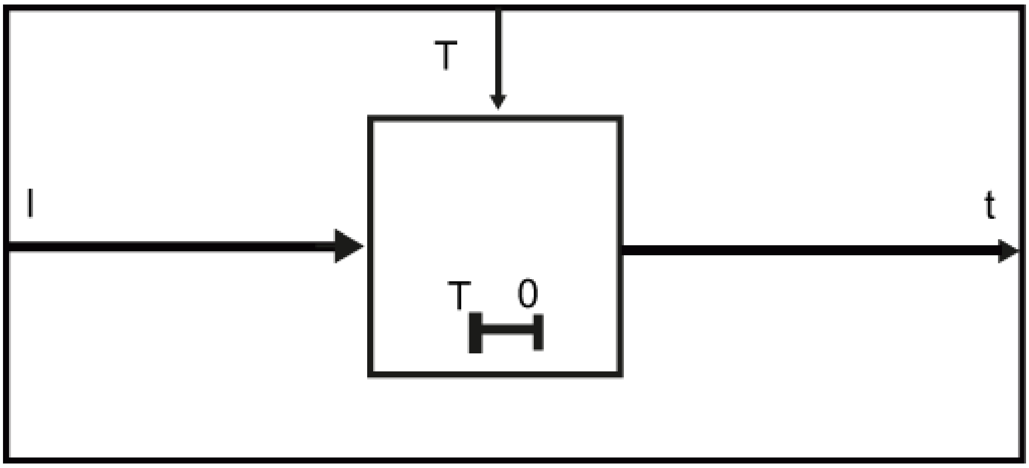
Initialization

The initialization defines the start value for the first cyclic pass.

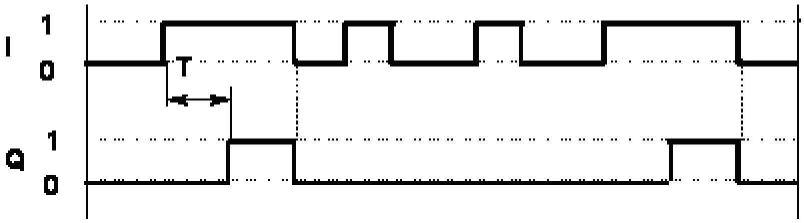
If input I receives the value 1 during initialization from the upstream block output, the block cannot detect a positive edge during the first cyclic pass. The pulse delay time T is therefore not taken over in the first cyclic pass with I = 1, the specified time from the initialization remains effective.

If output Q receives a value of 1 during initialization, then output Q = 1 is set immediately after initialization when I = 1.

Block diagram



Time diagram



Output pulse Q as a function of pulse duration T
and input pulse I

Block connections

Block connection	Description	Preassignment	Value range	Attributes
I	Input pulse	0	0/1	
T	Pulse delay time (ms)	0	SDTIME	
Q	Output pulse	0	0/1	

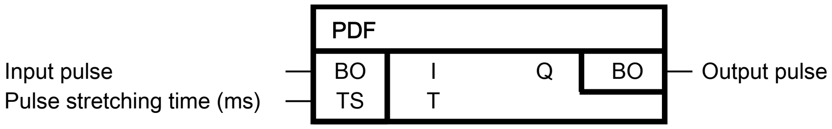
Project data

Can be loaded online	Yes
Special characteristics	-

4.44 PDF Off-delay device (BOOL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

BOOL-type timer with off-delay

Operation mode

The falling edge of a pulse at block input I resets output Q to 0 after pulse stretching time T.

Output Q becomes 1 when I becomes 1 (I = 0).

Output Q becomes 0 when input pulse I = 0 and the off-delay time T has expired.

If input I is reset to 1 before time T expires, then output Q remains at 1.

When T=0, a pulse stretching time of 1 cycle is active.

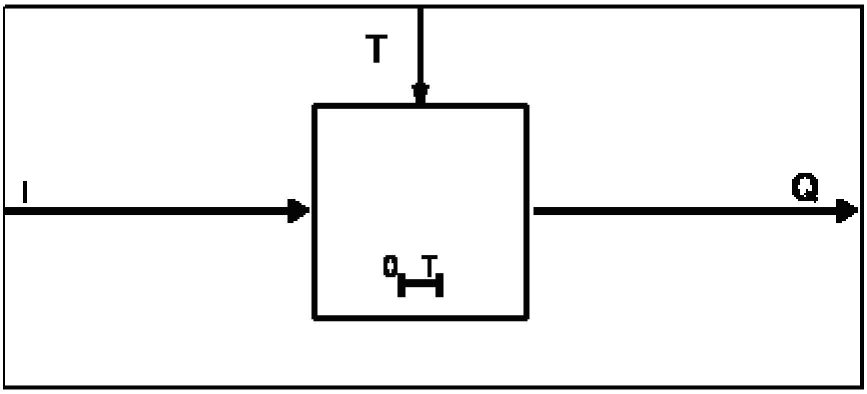
Initialization

The initialization defines the start value for the first cyclic pass.

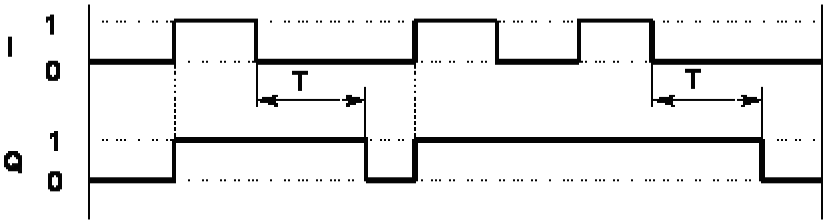
If input I receives the value 1 during initialization from the upstream block output, the block cannot detect a negative edge during the first cyclic pass.

If output Q receives the value 1 during initialization, output Q = 1 is set after initialization for the pulse stretching time T.

Block diagram



Time diagram



Output pulse Q as a function of pulse duration T
and input pulse I

Block connections

Block connection	Description	Preassignment	Value range	Attributes
I	Input pulse	0	0/1	
T	Pulse stretching time (ms)	0	SDTIME	
Q	Output pulse	0	0/1	

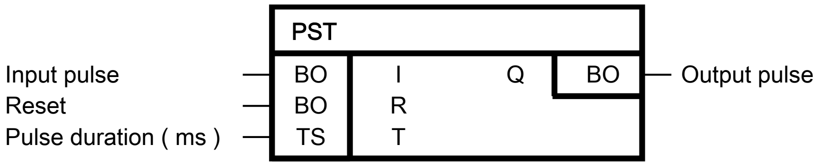
Project data

Can be loaded online	Yes
Special characteristics	-

4.45 PST Pulse stretching block (BOOL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Block for the generation of a pulse with a minimum duration and with additional reset input.

Operation mode

The rising edge of a pulse at input I sets output Q to 1.

Output Q does not fall back to 0 until input pulse I = 0 and the pulse duration T has expired.

Output Q can be set to zero at any time by means of the reset input R with R = 1.

When T=0, a pulse duration of 1 cycle is active.

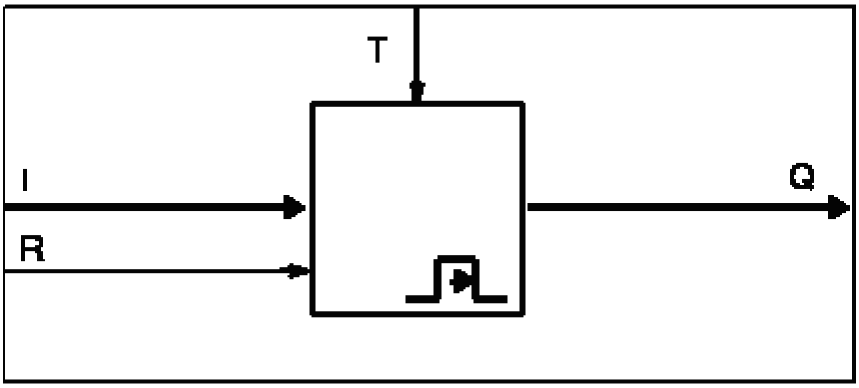
Initialization

The initialization defines the start value for the first cyclic pass.

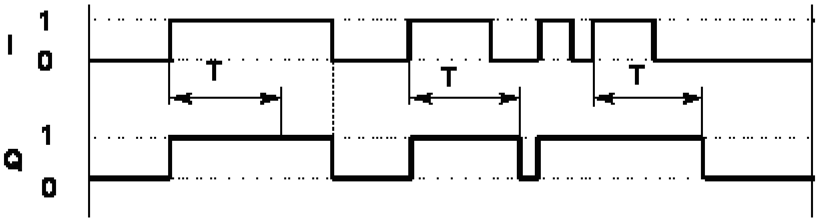
If input I receives the value 1 during initialization from the upstream block output, the block cannot detect a positive edge during the first cyclic pass.

If output Q receives the value 1 during initialization, output Q = 1 is set after initialization for the pulse duration T.

Block diagram



Time diagram



Output pulse Q as a function of pulse duration T and input pulse I (if R=0)

Block connections

Block connection	Description	Preassignment	Value range	Attributes
I	Input pulse	0	0/1	
R	Reset	0	0/1	
T	Pulse duration (ms)	0	SDTIME	
Q	Output pulse	0	0/1	

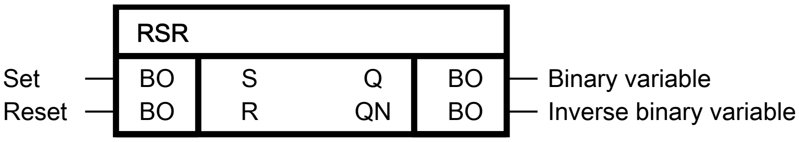
Project data

Can be loaded online	Yes
Special characteristics	-

4.46 RSR RS flip-flop, R-dominant (BOOL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

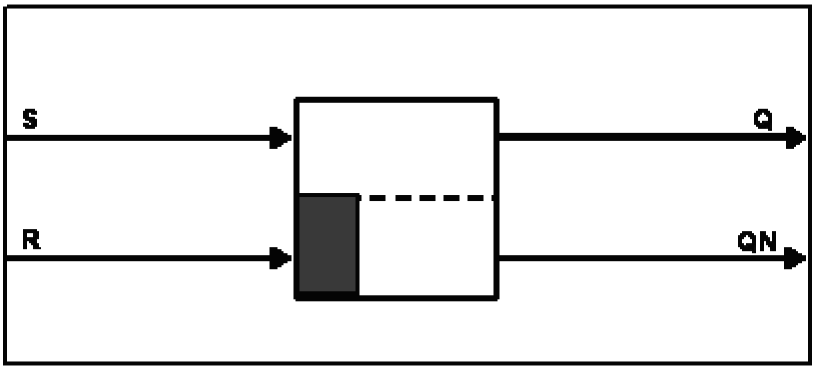
Used as static binary value memory

Operation mode

With logic 1 at input S, output Q is set to logic 1. If input R is set to logic 1, then output Q is set to logic 0. If both inputs are logic 0, then Q does not change. However, if the two inputs are logic 1, then Q is logic 0, since the reset input dominates.

Output QN always has the value inverse to Q.

Block diagram



Truth table(s)

Binary values when set/reset command is given

Binary command		Output status Q
S	R	
0	0	Q does not change
0	1	Q = 0
1	0	Q = 1
1	1	Q = 0

Block connections

Block connection	Description	Preassignment	Value range	Attributes
S	Set	0	0/1	
R	Reset	0	0/1	
Q	Binary variable	0	0/1	
QN	Inverse binary variable	1	0/1	

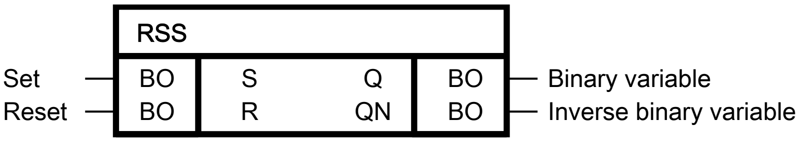
Project data

Can be loaded online	Yes
Special characteristics	-

4.47 RSS RS flip-flop, S-dominant (BOOL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

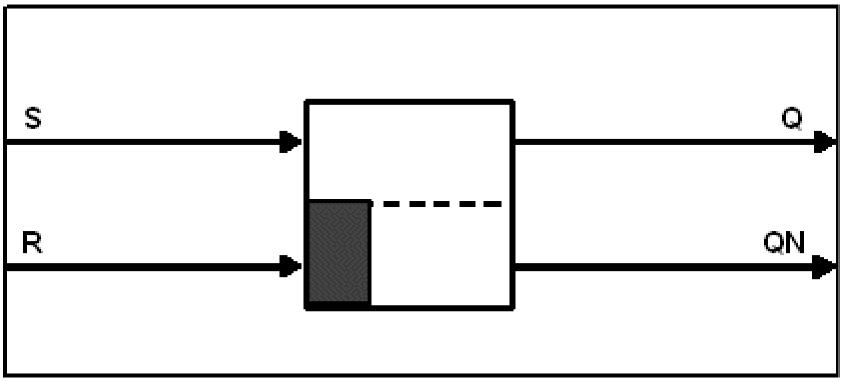
Block of the BOOL type for use as a static binary value memory

Operation mode

With logic 1 at input S, output Q is set to logic 1. If input R is set to logic 1, then output Q is set to logic 0. If both inputs are logic 0, then Q does not change. However, if the two inputs are logic 1, then Q is also logic 1, since the setting input dominates.

Output QN always has the value inverse to Q.

Block diagram



Truth table(s)

Binary values when set/reset command is given

Binary command		Output status Q
S	R	
0	0	Q does not change
0	1	Q = 0
1	0	Q = 1
1	1	Q = 1

Block connections

Block connection	Description	Preassignment	Value range	Attributes
S	Set	0	0/1	
R	Reset	0	0/1	
Q	Binary variable	0	0/1	
QN	Inverse binary variable	1	0/1	

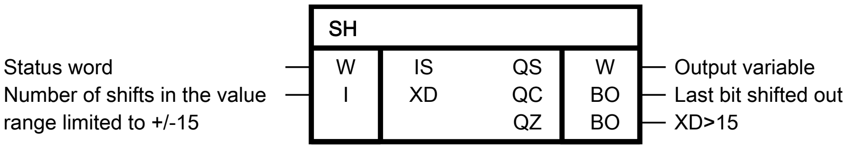
Project data

Can be loaded online	Yes
Special characteristics	-

4.48 SH Shift block (WORD type)

☑ SIMOTION □ SINAMICS

Symbol



Short description

The block of the WORD type shifts a status word bit-by-bit to the left or right.

Operation mode

The block shifts the status word present at input IS bit-by-bit by the number of positions specified at input XD.

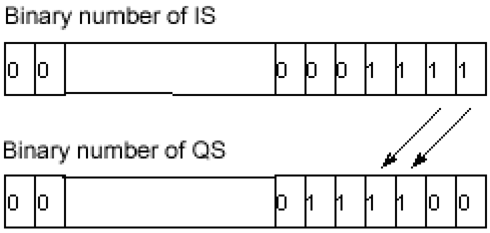
During shifting, new positions in output variable QS are filled with zeros irrespective of the shift direction.

The last bit shifted out is output on output QC. When XD = 0, QC = 0 is always true. When |XD| > 15, QC = 0, QS = 0, and QZ = 1 are always true.

Shift to the left - example:

XD = 2; IS = 15

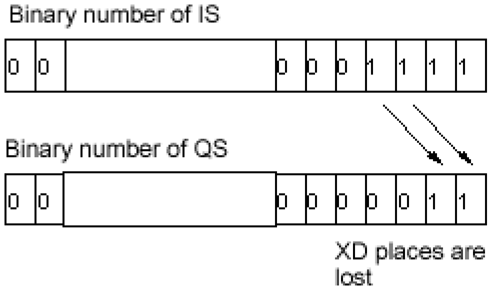
-> QS = 60; QC = 0



Shift to the right - example:

XD = -2; IS = 15

-> QS = 3 (remainder is omitted); QC = 1



Block connections

Block connection	Description	Preassignment	Value range	Attributes
IS	Status word	16#0000	WORD	
XD	Number of shifts in the value range limited to +/-15	0	INT	
QS	Output variable	16#0000	WORD	
QC	Last bit shifted out	0	0/1	
QZ	XD>15	0	0/1	

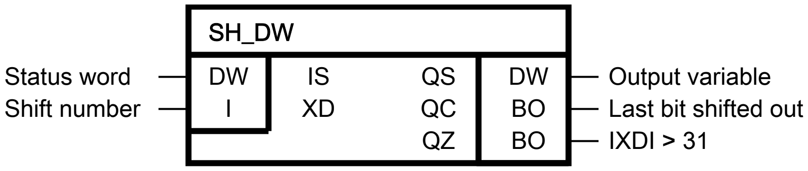
Project data

Can be loaded online	Yes
Special characteristics	-

4.49 SH_DW Shift block (DWORD type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

The block of the DWORD type shifts a status word bit-by-bit to the left or right.

Operation mode

The block shifts the status word present at input IS bit-by-bit by the number of positions specified at input XD.

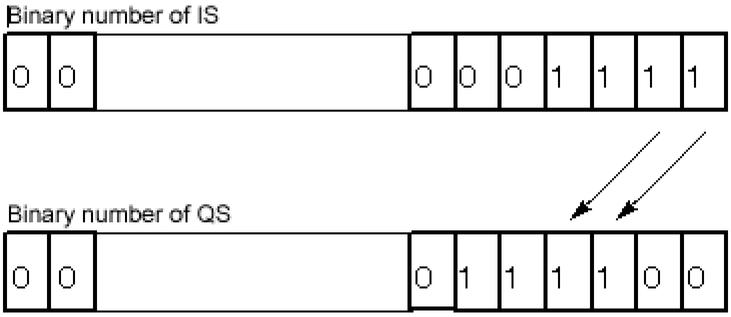
During shifting, new positions in output variable QS are filled with zeros irrespective of the shift direction.

The last bit shifted out is output on output QC. When XD = 0, QC = 0 is always true. When |XD| > 31, QC = 0, QS = 0, and QZ = 1 are always true.

Shift to the left - example:

XD = 2; IS = 15

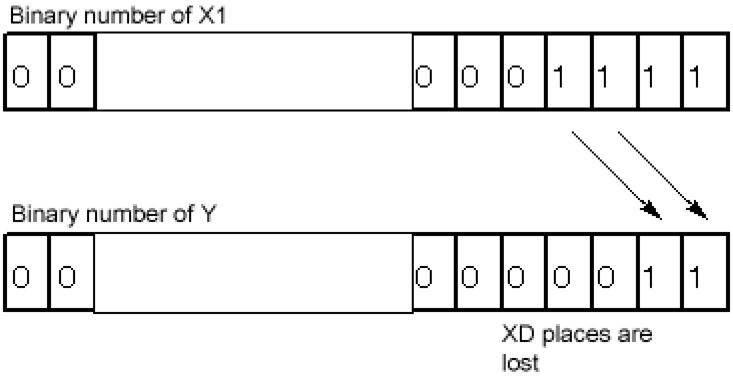
-> QS = 60; QC = 0



Shift to the right - example:

XD = -2; IS = 15

-> QS = 3 (remainder is omitted); QC = 1



Block connections

Block connection	Description	Preassignment	Value range	Attributes
IS	Status word	16#00000000	DWORD	
XD	Shift number	0	+/-31	
QS	Output variable	16#00000000	DWORD	
QC	Last bit shifted out	0	0/1	
QZ	IXDI > 31	0	0/1	

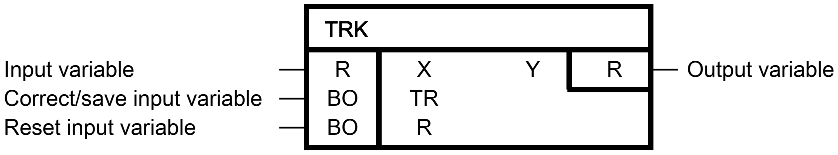
Project data

Can be loaded online	Yes
Special characteristics	-

4.50 TRK Correction/memory element (REAL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Block of the REAL type for saving a current input value with the following properties:

- Edge-controlled latch functions for the input value
- Level-controlled correction of the output value

Operation mode

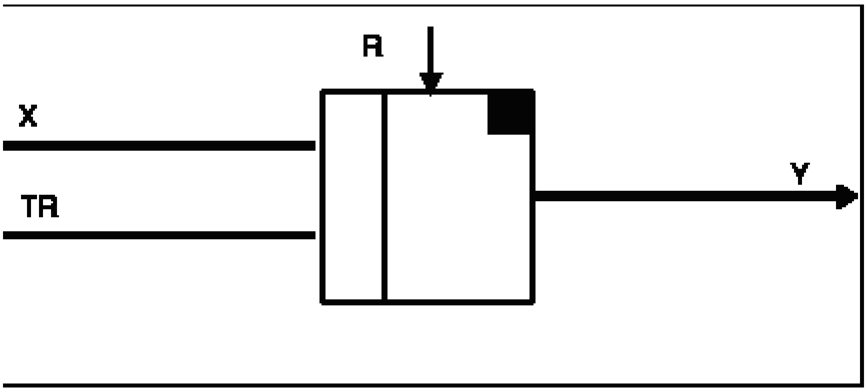
TRACK	TR = 1	Direct correction of output value Y = X.
	TR = 1-> 0	With a negative edge at TR, the current input variable is saved and output on output Y
	TR = 0	The value at output Y does not change
RESET	R = 1	Output Y is reset to 0. The reset input is dominant.

Initialization

If input TR receives the value 1 during initialization of an upstream block output, a negative edge can be detected during the first cyclic pass. In START mode, the value for TR is stored temporarily.

If input TR receives the value 0 during initialization of the upstream block output, the block cannot detect a negative edge during the first cyclic pass.

Block diagram



Truth table(s)

Input		Output Y at the time of triggering
TR	R	
0	0	$Y_n = Y_{n-1}$
1	0	$Y_n = X_n$
1	1	$Y_n = 0$
1->0	0	$Y_n = X_n$
1->0	1	$Y_n = 0$

1 -> 0: fall

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
TR	Correct/save input variable	0	0/1	
R	Reset input variable	0	0/1	
Y	Output variable	0.0	REAL	

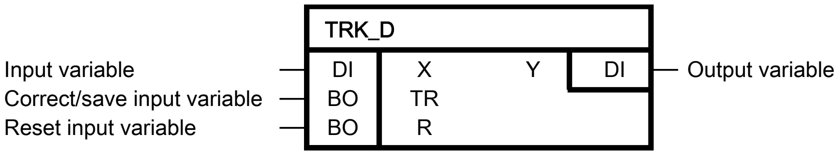
Project data

Can be loaded online	Yes
Special characteristics	-

4.51 TRK_D Correction/memory element (DOUBLE INTEGER type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Block of the DOUBLE INTEGER type for saving a current input value with the following properties:

- Edge-controlled latch functions for the input value
- Level-controlled correction of the output value

Operation mode

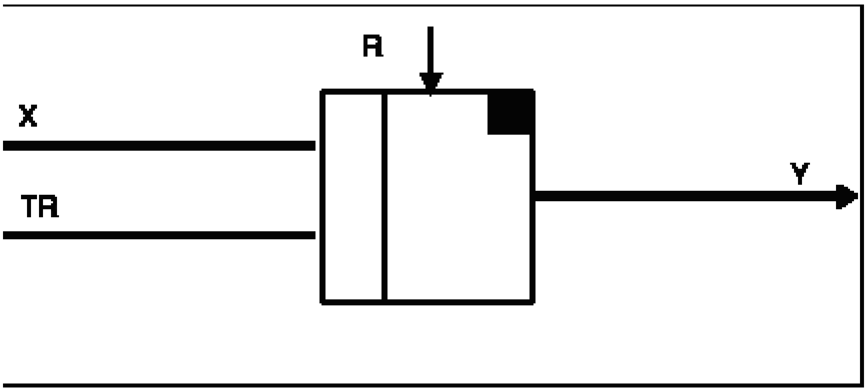
TRACK	TR = 1	Direct correction of output value Y = X.
	TR = 1-> 0	With a negative edge at TR, the current input variable is saved and output on output Y
	TR = 0	The value at output Y does not change
RESET	R = 1	Output Y is reset to 0. The reset input is dominant.

Initialization

If input TR receives the value 1 during initialization of an upstream block output, a negative edge can be detected during the first cyclic pass. In START mode, the value for TR is stored temporarily.

If input TR receives the value 0 during initialization of the upstream block output, the block cannot detect a negative edge during the first cyclic pass.

Block diagram



Truth table(s)

Input		Output Y at the time of triggering
TR	R	
0	0	$Y_n = Y_{n-1}$
1	0	$Y_n = X_n$
1	1	$Y_n = 0$
1->0	0	$Y_n = X_n$
1->0	1	$Y_n = 0$

1 -> 0 : Fall

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	DINT	
TR	Correct/save input variable	0	0/1	
R	Reset input variable	0	0/1	
Y	Output variable	0	DINT	

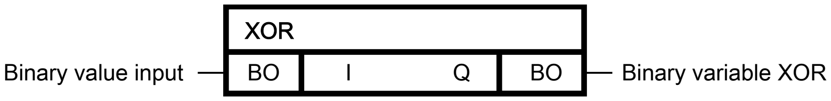
Project data

Can be loaded online	Yes
Special characteristics	-

4.52 XOR Logic exclusive OR operation (BOOL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

XOR block with up to four inputs of the BOOL type

Operation mode

The block combines the binary values at the inputs I 1-4 according to the logic exclusive OR function and outputs the result at its binary output Q.

Output Q is 0, when a 0 is present at all inputs I1 to I4 or when a 1 is present at an even number of the inputs I1 to I4.

Output Q is 1, when a 1 is present at an odd number of the inputs I1 to I4.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
I	Binary value input	0	0/1	
Q	Binary variable XOR	0	0/1	

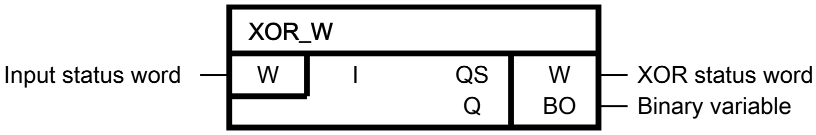
Project data

Can be loaded online	Yes
Special characteristics	I comprises up to four connections (I1 to I4)

4.53 XOR_W Logic exclusive OR operation (WORD type)

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

XOR block with up to four inputs of the WORD type

Operation mode

The block combines the status words I1 to I4 bit-by-bit according to the logic exclusive OR function.

The result is given to the block output QS (status word XOR).

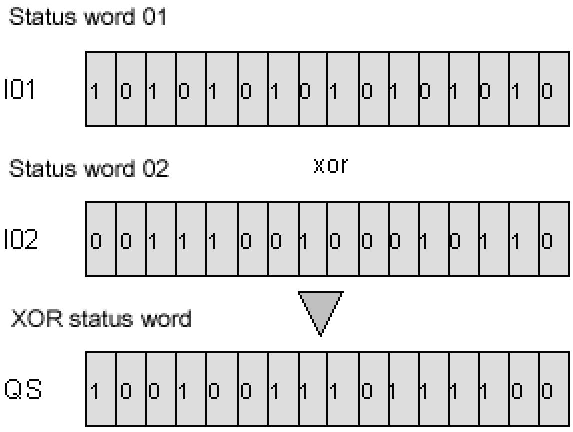
The following applies for bit k of status word XOR:

$$QS_k = (\overline{I01_k} \wedge I02_k) \vee (I01_k \wedge \overline{I02_k}),\quad k = 1...16$$

A bit of the XOR status word is equal to 1 when an odd number of the equivalent bits on the block inputs I1 to I4 is equal to 1.

The binary output Q is 1 if at least one bit of the status word XOR is equal to 1.

Following state diagram (for 3 inputs)


Status word 01
I01 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
Status word 02
xor
I02 0 0 1 1 1 0 0 1 0 0 0 1 0 1 1 0
XOR status word
QS 1 0 0 1 0 0 1 1 1 0 1 1 1 1 0 0

Block connections

Block connection	Description	Preassignment	Value range	Attributes
I	Input status word	16#0000	WORD	
QS	XOR status word	16#0000	WORD	
Q	Binary variable	0	0/1	

Project data

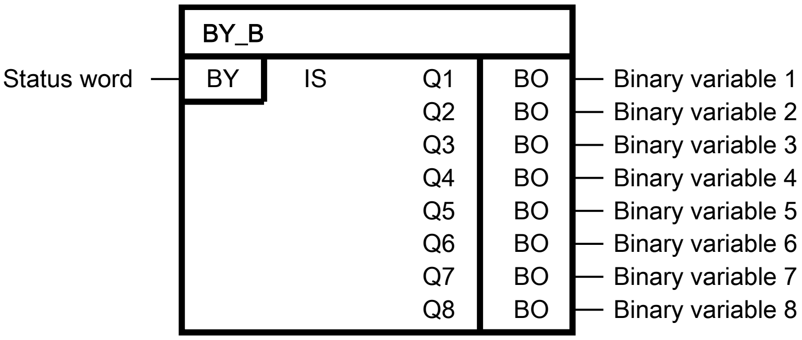
Can be loaded online	Yes
Special characteristics	Up to four connections (I1 to I4)

Conversion 5

5.1 BY_B Converter status byte to 8 binary variables

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

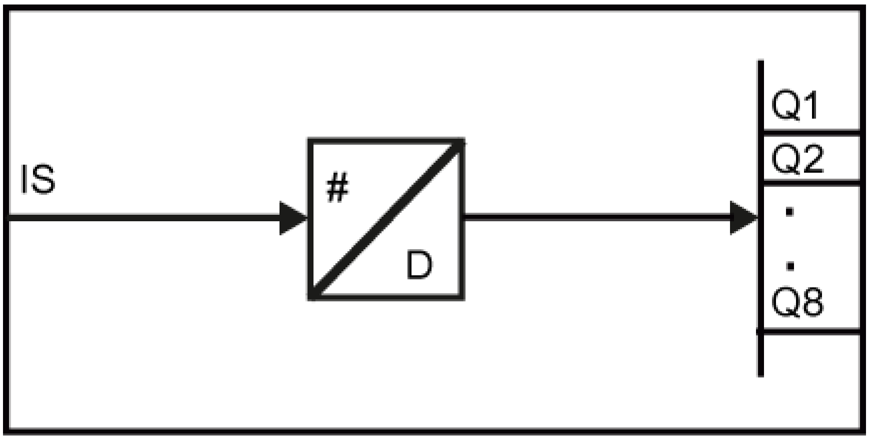
Status word decoding to 8 binary variables

Operation mode

This block decodes the status word IS to 8 binary variables and gives the result to its outputs Q1 to Q8.

The binary variable of outputs Q1 to Q8 is assigned to each dual equivalent 2^0 to 2^7 of the status byte.

Block diagram



Mapping scheme

Bit position (dual equivalent) of status byte IS	Output variable
0 (2^0)	Q1
1 (2^1)	Q2
2 (2^2)	Q3
3 (2^3)	Q4
4 (2^4)	Q5
5 (2^5)	Q6
6 (2^6)	Q7
7 (2^7)	Q8

Block connections

Block connection	Description	Preassignment	Value range	Attributes
IS	Status word	16#00	BYTE	
Q1	Binary variable 1	0	0/1	
Q2	Binary variable 2	0	0/1	
Q3	Binary variable 3	0	0/1	
Q4	Binary variable 4	0	0/1	
Q5	Binary variable 5	0	0/1	
Q6	Binary variable 6	0	0/1	
Q7	Binary variable 7	0	0/1	
Q8	Binary variable 8	0	0/1	

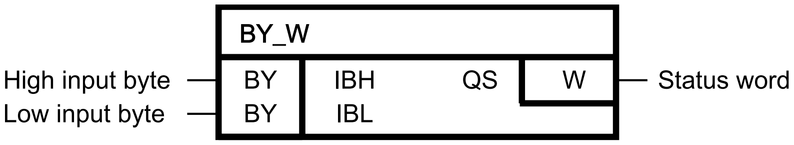
Project data

Can be inserted online	Yes
Special characteristics	-

5.2 BY_W Status byte to status word converter

☑ SIMOTION ☑ SINAMICS

Symbol



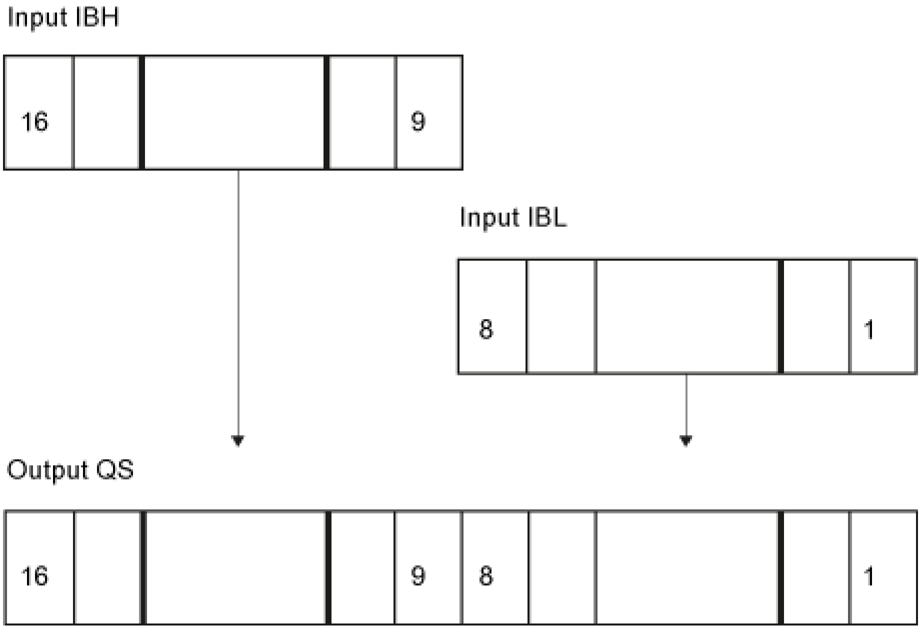
Short description

Combining 2 bytes into one word

Operation mode

The block combines two bytes into a word. The low byte of the output word is assigned to the input byte IBL and the high byte of the output word is assigned to the input byte IBH. The output word is present on QS according to the following conversion scheme.

Conversion scheme



Block connections

Block connection	Description	Preassignment	Value range	Attributes
IBH	High input byte	16#00	BYTE	
IBL	Low input byte	16#00	BYTE	
QS	Status word	16#0000	WORD	

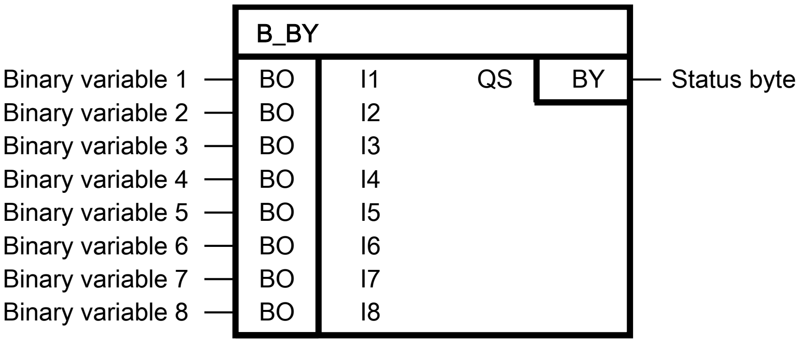
Project data

Can be inserted online	Yes
Special characteristics	-

5.3 B_BY Converter 8 binary variable to status byte

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

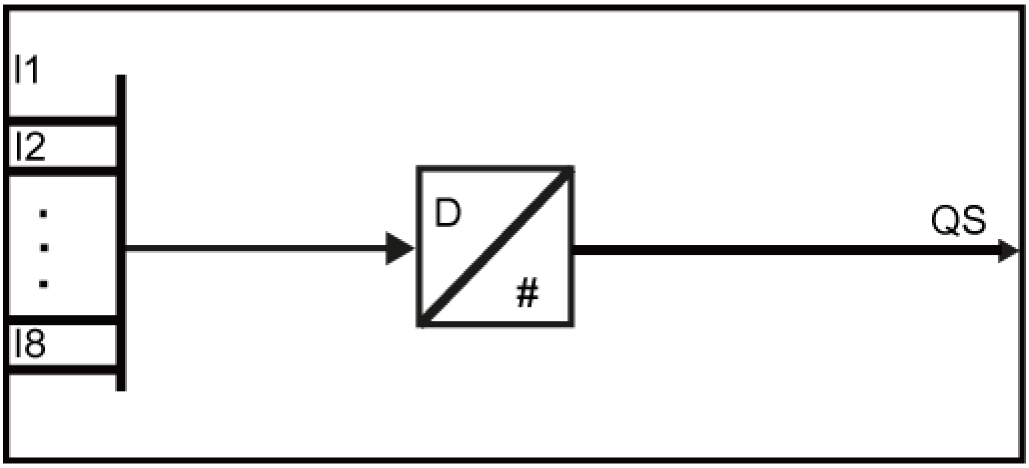
Status byte generation from 8 binary variables

Operation mode

This block combines the binary variables from I1 to I8 into a status byte and gives the result to its output QS.

Each binary variable of inputs I1 to I8 is assigned the dual equivalent 2^0 to 2^7 from which the status word is generated.

Block diagram



Mapping scheme

Input variable	Bit position (dual equivalent) of status byte QS
I1	0 (2^0)
I2	1 (2^1)
I3	2 (2^2)
I4	3 (2^3)
I5	4 (2^4)
I6	5 (2^5)
I7	6 (2^6)
I8	7 (2^7)

Block connections

Block connection	Description	Preassignment	Value range	Attributes
I1	Binary variable 1	0	0/1	
I2	Binary variable 2	0	0/1	
I3	Binary variable 3	0	0/1	
I4	Binary variable 4	0	0/1	
I5	Binary variable 5	0	0/1	
I6	Binary variable 6	0	0/1	
I7	Binary variable 7	0	0/1	
I8	Binary variable 8	0	0/1	
QS	Status byte	16#00	BYTE	

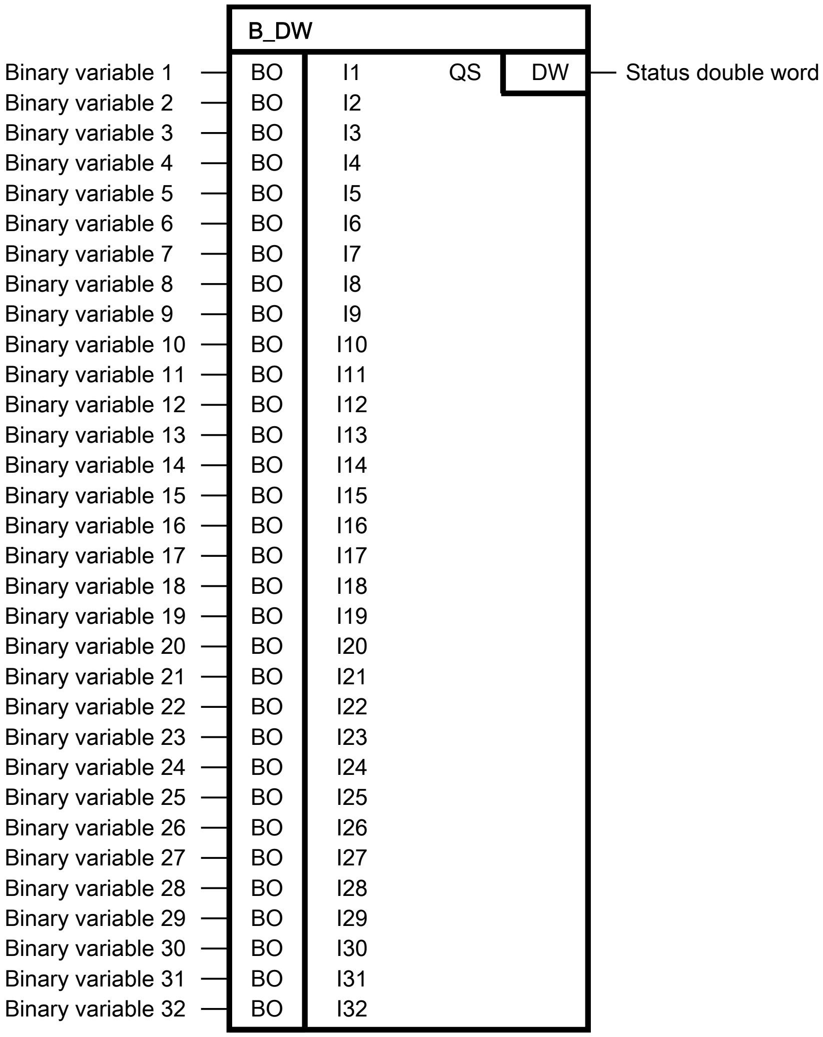
Project data

Can be inserted online	Yes
Special characteristics	-

5.4 B_DW Converter 32 binary variables to status double word

☑ SIMOTION ☑ SINAMICS

Symbol



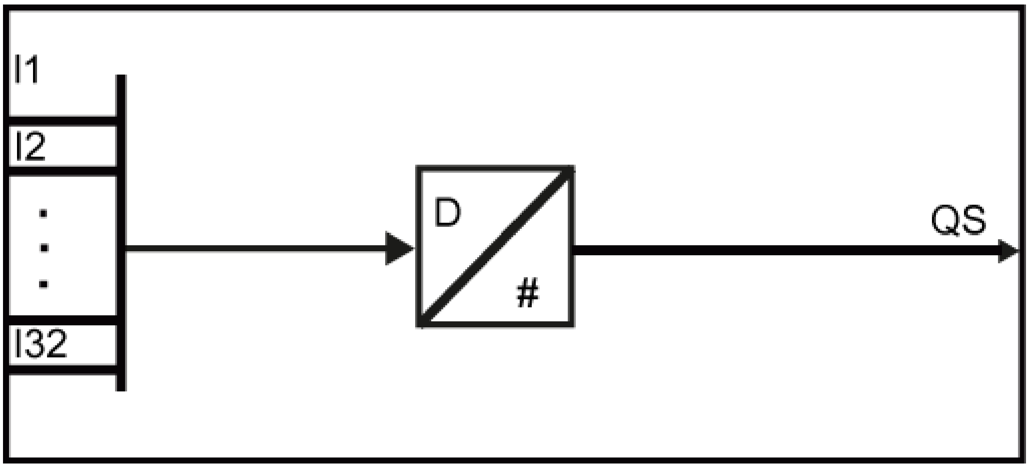
Short description

Status double word generation from 32 binary variables

Operation mode

The block combines the binary variables of I1 to I32 into a status double word and outputs the result at output QS. Each binary variable of inputs I1 to I32 is assigned the dual equivalent 2^0 to 2^{31} from which the status double word is generated.

Block diagram



Mapping scheme

Input parameters	Bit position (dual equivalent) of status byte QS
|1	0 (2^0)
|2	1 (2^1)
|3	2 (2^2)
...	...
|32	31 (2^{31})

Block connections

Block connection	Description	Preassignment	Value range	Attributes
I1	Binary variable 1	0	0/1	
I2	Binary variable 2	0	0/1	
I3	Binary variable 3	0	0/1	
I4	Binary variable 4	0	0/1	
I5	Binary variable 5	0	0/1	
I6	Binary variable 6	0	0/1	
I7	Binary variable 7	0	0/1	
I8	Binary variable 8	0	0/1	
I9	Binary variable 9	0	0/1	
I10	Binary variable 10	0	0/1	
I11	Binary variable 11	0	0/1	
I12	Binary variable 12	0	0/1	
I13	Binary variable 13	0	0/1	
I14	Binary variable 14	0	0/1	
I15	Binary variable 15	0	0/1	
I16	Binary variable 16	0	0/1	
I17	Binary variable 17	0	0/1	
I18	Binary variable 18	0	0/1	
I19	Binary variable 19	0	0/1	
I20	Binary variable 20	0	0/1	
I21	Binary variable 21	0	0/1	
I22	Binary variable 22	0	0/1	
I23	Binary variable 23	0	0/1	
I24	Binary variable 24	0	0/1	
I25	Binary variable 25	0	0/1	
I26	Binary variable 26	0	0/1	
I27	Binary variable 27	0	0/1	
I28	Binary variable 28	0	0/1	
I29	Binary variable 29	0	0/1	
I30	Binary variable 30	0	0/1	
I31	Binary variable 31	0	0/1	
I32	Binary variable 32	0	0/1	
QS	Status double word	16#00000000	DWORD	

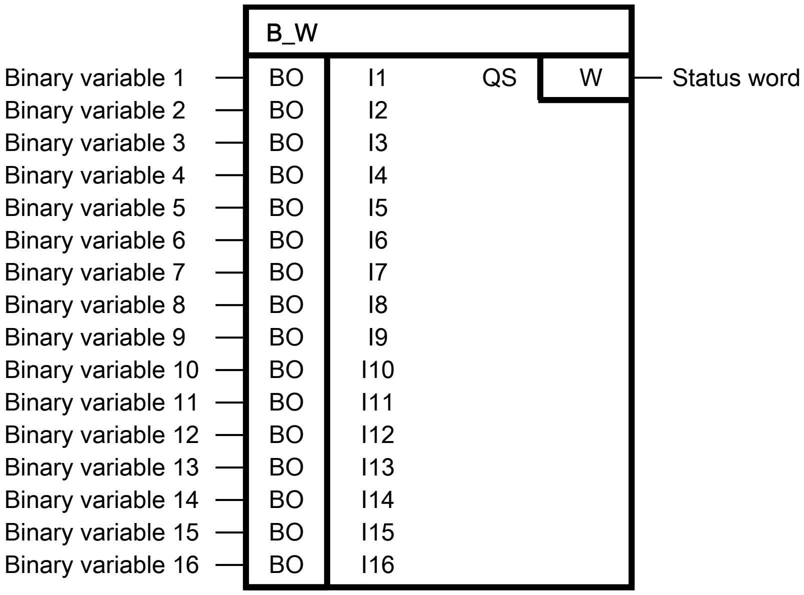
Project data

Can be inserted online	Yes
Special characteristics	-

5.5 B_W Converter 16 binary variables to status word

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

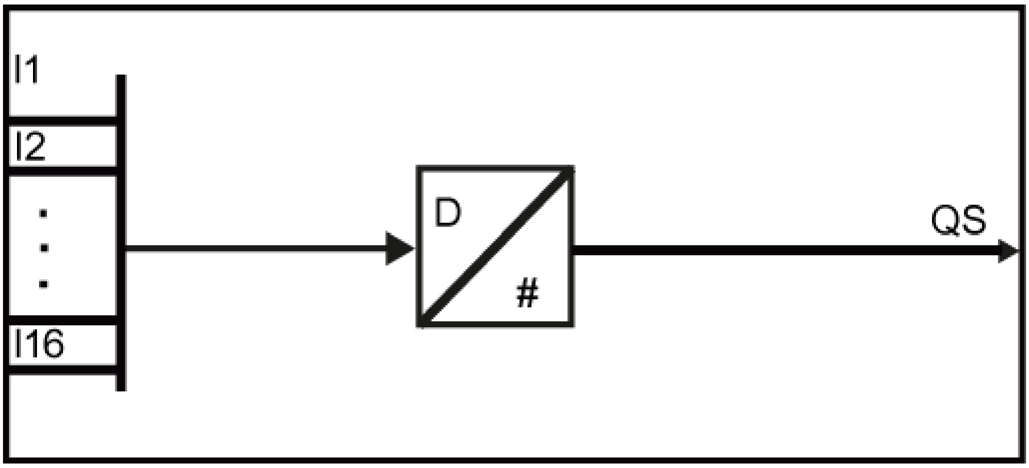
Status word generation from 16 binary variables.

Operation mode

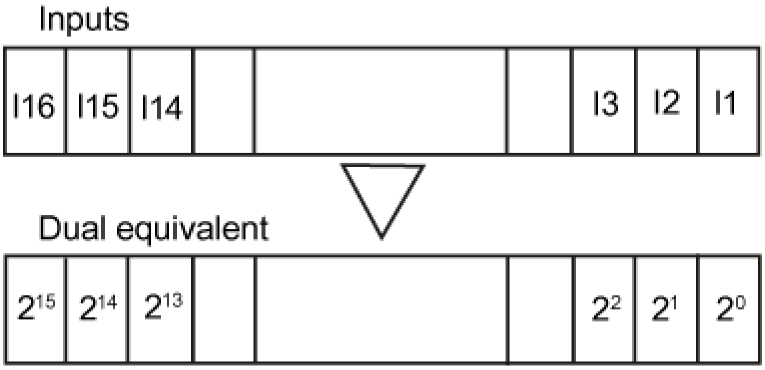
The block combines the binary variables of I1 to I16 into a status word and outputs the result at output QS.

Each binary variable of inputs I1 to I16 is assigned the dual equivalent 2^0 to 2^{15} from which the status word is generated.

Block diagram



Conversion scheme



Block connections

Block connection	Description	Preassignment	Value range	Attributes
I1	Binary variable 1	0	0/1	
I2	Binary variable 2	0	0/1	
I3	Binary variable 3	0	0/1	
I4	Binary variable 4	0	0/1	
I5	Binary variable 5	0	0/1	
I6	Binary variable 6	0	0/1	
I7	Binary variable 7	0	0/1	
I8	Binary variable 8	0	0/1	
I9	Binary variable 9	0	0/1	
I10	Binary variable 10	0	0/1	
I11	Binary variable 11	0	0/1	
I12	Binary variable 12	0	0/1	
I13	Binary variable 13	0	0/1	
I14	Binary variable 14	0	0/1	
I15	Binary variable 15	0	0/1	
I16	Binary variable 16	0	0/1	
QS	Status word	16#0000	WORD	

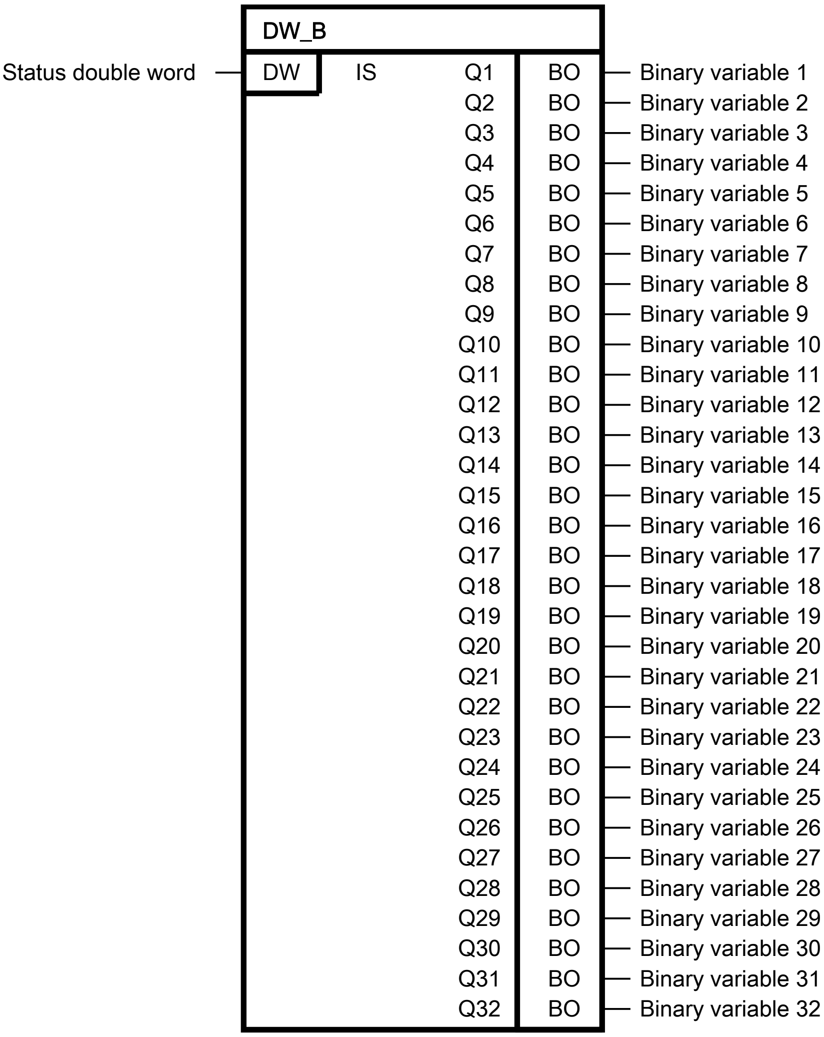
Project data

Can be inserted online	Yes
Special characteristics	-

5.6 DW_B Converter status double word to 32 binary variables

☑ SIMOTION ☑ SINAMICS

Symbol


DW_B
Status double word
DW
IS
Q1 BO Binary variable 1
Q2 BO Binary variable 2
Q3 BO Binary variable 3
Q4 BO Binary variable 4
Q5 BO Binary variable 5
Q6 BO Binary variable 6
Q7 BO Binary variable 7
Q8 BO Binary variable 8
Q9 BO Binary variable 9
Q10 BO Binary variable 10
Q11 BO Binary variable 11
Q12 BO Binary variable 12
Q13 BO Binary variable 13
Q14 BO Binary variable 14
Q15 BO Binary variable 15
Q16 BO Binary variable 16
Q17 BO Binary variable 17
Q18 BO Binary variable 18
Q19 BO Binary variable 19
Q20 BO Binary variable 20
Q21 BO Binary variable 21
Q22 BO Binary variable 22
Q23 BO Binary variable 23
Q24 BO Binary variable 24
Q25 BO Binary variable 25
Q26 BO Binary variable 26
Q27 BO Binary variable 27
Q28 BO Binary variable 28
Q29 BO Binary variable 29
Q30 BO Binary variable 30
Q31 BO Binary variable 31
Q32 BO Binary variable 32

Short description

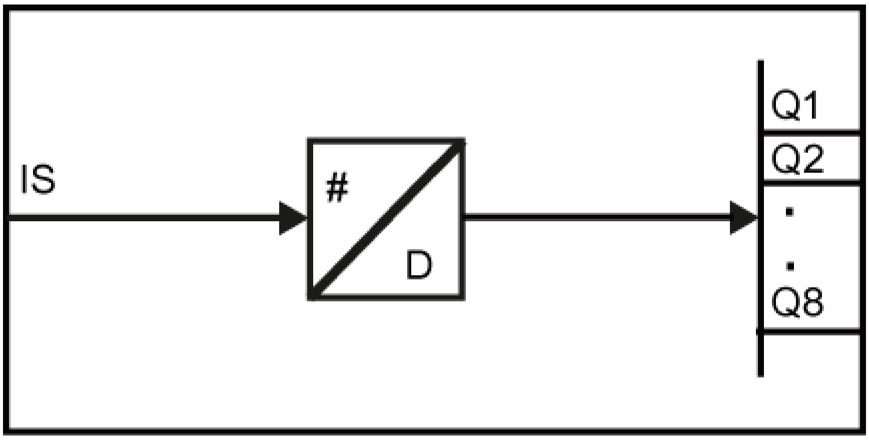
Status double word decryption to 32 binary variables

Operation mode

This block decodes the status double word IS to 32 binary variables and gives the result to its outputs Q1 to Q32.

The binary variable of outputs Q1 to Q32 is assigned to each dual equivalent 2^0 to 2^{31} of the status word.

Block diagram



Mapping scheme

Bit position (dual equivalent) of status double word IS	Output variable
0 (2^0)	Q1
1 (2^1)	Q2
2 (2^2)	Q3
...	...
31 (2^{31})	Q32

Block connections

Block connection	Description	Preassignment	Value range	Attributes
IS	Status double word	16#00000000	DWORD	
Q1	Binary variable 1	0	0/1	
Q2	Binary variable 2	0	0/1	
Q3	Binary variable 3	0	0/1	
Q4	Binary variable 4	0	0/1	
Q5	Binary variable 5	0	0/1	
Q6	Binary variable 6	0	0/1	
Q7	Binary variable 7	0	0/1	
Q8	Binary variable 8	0	0/1	
Q9	Binary variable 9	0	0/1	
Q10	Binary variable 10	0	0/1	
Q11	Binary variable 11	0	0/1	
Q12	Binary variable 12	0	0/1	
Q13	Binary variable 13	0	0/1	
Q14	Binary variable 14	0	0/1	
Q15	Binary variable 15	0	0/1	
Q16	Binary variable 16	0	0/1	
Q17	Binary variable 17	0	0/1	
Q18	Binary variable 18	0	0/1	
Q19	Binary variable 19	0	0/1	
Q20	Binary variable 20	0	0/1	
Q21	Binary variable 21	0	0/1	
Q22	Binary variable 22	0	0/1	
Q23	Binary variable 23	0	0/1	
Q24	Binary variable 24	0	0/1	
Q25	Binary variable 25	0	0/1	
Q26	Binary variable 26	0	0/1	
Q27	Binary variable 27	0	0/1	
Q28	Binary variable 28	0	0/1	
Q29	Binary variable 29	0	0/1	
Q30	Binary variable 30	0	0/1	
Q31	Binary variable 31	0	0/1	
Q32	Binary variable 32	0	0/1	

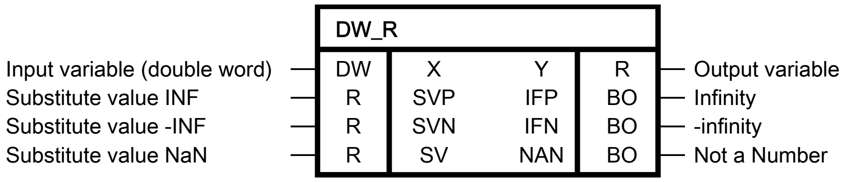
Project data

Can be inserted online	Yes
Special characteristics	-

5.7 DW_R Accepting bit string as real value

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

This block accepts the bit string at the input as a real variable and checks the value for validity

Operation mode

The DW_R block accepts the bit string at the input as a real variable and supplies it at output Y.

The bit pattern of input variable X is checked. If the bit pattern according to IEEE 754 corresponds to the representation for +/-infinity or NaN, the relevant binary outputs IFP, IFN or NAN are set to 1 and the respective specified substitute values are effective at output Y.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable (double word)	16#00000000	DWORD	
SVP	Substitute value INF	3.402823 E38	REAL	
SVN	Substitute value -INF	-3.402823 E38	REAL	
SV	Substitute value NaN	0.0	REAL	
IFP	Infinity	0	0/1	
IFN	-infinity	0	0/1	
NAN	Not a Number	0	0/1	
Y	Output variable	0.0	REAL	

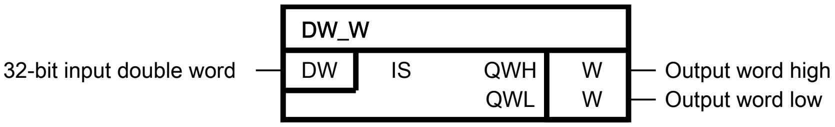
Project data

Can be inserted online	**Yes**
Special characteristics	-

5.8 DW_W Status double word to status word converter

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

A 32-bit double word is divided into two 16-bit words.

Operation mode

Output variables are calculated in line with the following regulation:

QWL = IS mod 2^{16}

QWH = IS / 2^{16}

Block connections

Block connection	Description	Preassignment	Value range	Attributes
IS	32-bit input double word	16#00000000	DWORD	
QWH	Output word high	16#0000	WORD	
QWL	Output word low	16#0000	WORD	

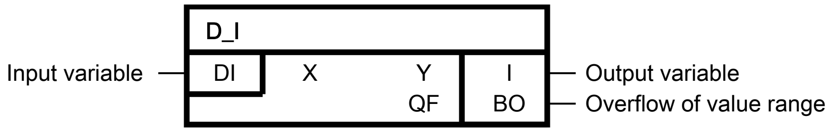
Project data

Can be inserted online	Yes
Special characteristics	-

5.9 D_I DOUBLE INTEGER to INTEGER converter

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Conversion of a DOUBLE INTEGER variable to an INTEGER variable

Operation mode

This block converts a DOUBLE INTEGER variable to an INTEGER variable, i.e., the low word of the DOUBLE INTEGER input variable is applied to the output variable Y.

If the value of input variable X exceeds the value range of output variable Y, then QF = 1 is set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	DINT	
Y	Output variable	0	INT	
QF	Overflow of value range	0	0/1	

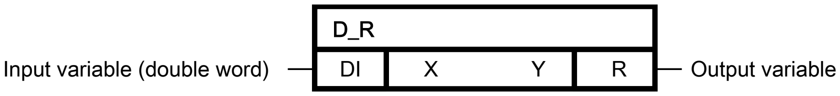
Project data

Can be inserted online	Yes
Special characteristics	-

5.10 D_R DOUBLE-INTEGER to REAL converter

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Conversion of a DOUBLE INTEGER variable to a REAL variable

Operation mode

This block converts a DOUBLE INTEGER variable to a REAL variable.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable (double word)	0	DINT	
Y	Output variable	0.0	REAL	

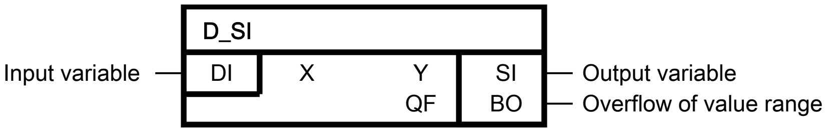
Project data

Can be inserted online	Yes
Special characteristics	-

5.11 D_SI DOUBLE INTEGER to SHORT INTEGER converter

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

Conversion of a DOUBLE INTEGER variable to a SHORT INTEGER variable

Operation mode

This block converts a DOUBLE INTEGER variable to a SHORT INTEGER variable, i.e., the least significant byte of the DOUBLE INTEGER input variable is applied to output variable Y.

If the value of input variable X exceeds the value range of output variable Y, then QF = 1 is set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	DINT	
Y	Output variable	0	SINT	
QF	Overflow of value range	0	0/1	

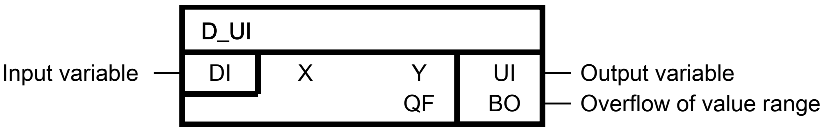
Project data

Can be inserted online	**Yes**
Special characteristics	-

5.12 D_UI DOUBLE INTEGER to UNSIGNED INTEGER converter

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Conversion of a DOUBLE INTEGER variable to an UNSIGNED INTEGER variable

Operation mode

This block converts a DOUBLE INTEGER variable to an UNSIGNED INTEGER variable, i.e. the low word of the DOUBLE INTEGER input variable is applied to the output variable.

If the value of input variable X exceeds the value range of output variable Y, then QF = 1 is set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	DINT	
Y	Output variable	0	UINT	
QF	Overflow of value range	0	0/1	

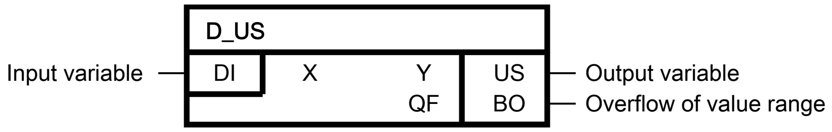
Project data

Can be inserted online	Yes
Special characteristics	-

5.13 D_US DOUBLE INTEGER to UNSIGNED SHORT INTEGER converter

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Conversion of a DOUBLE INTEGER variable to an UNSIGNED SHORT INTEGER variable

Operation mode

This block converts a DOUBLE INTEGER variable to an UNSIGNED SHORT INTEGER variable, i.e., the low word of the DOUBLE INTEGER input variable is applied to the output variable.

If the value of input variable X exceeds the value range of output variable Y, then QF = 1 is set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	DINT	
Y	Output variable	0	USINT	
QF	Overflow of value range	0	0/1	

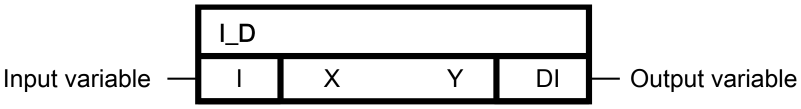
Project data

Can be inserted online	Yes
Special characteristics	-

5.14 I_D INTEGER to DOUBLE_INTEGER converter

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Conversion of an INTEGER variable to a DOUBLE INTEGER variable

Operation mode

This block converts an integer variable to a double integer variable.

The input variable of data type INTEGER is copied to the low word of the output variable. If the input variable has a positive sign, the high word of the output variable is filled with 16#0000 - if, on the other hand, the sign is negative, the high word receives the value 16#FFFF.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	INT	
Y	Output variable	0	DINT	

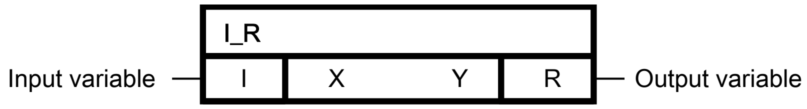
Project data

Can be inserted online	Yes
Special characteristics	-

5.15 I_R INTEGER to REAL converter

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Conversion of an INTEGER variable to a REAL variable

Operation mode

This block converts an integer variable to a real variable.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	INT	
Y	Output variable	0.0	REAL	

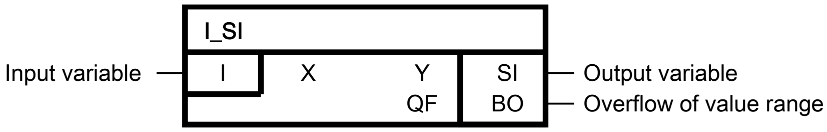
Project data

Can be inserted online	Yes
Special characteristics	-

5.16 I_SI INTEGER to SHORT INTEGER converter

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

Conversion of an INTEGER variable to a SHORT INTEGER variable

Operation mode

This block converts an INTEGER variable to a SHORT INTEGER variable, i.e., the low-order byte of the INTEGER input variable is applied to output variable Y.

If the value of input variable X exceeds the value range of output variable Y, then QF = 1 is set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	INT	
Y	Output variable	0	SINT	
QF	Overflow of value range	0	0/1	

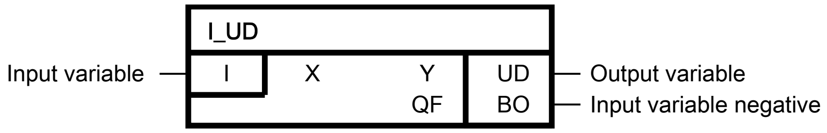
Project data

Can be inserted online	Yes
Special characteristics	-

5.17 I_UD INTEGER to UNSIGNED DOUBLE INTEGER converter

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Conversion of an INTEGER variable to an UNSIGNED DOUBLE INTEGER variable

Operation mode

This block converts an INTEGER variable to an UNSIGNED DOUBLE INTEGER variable.

The input variable of data type INTEGER is copied to the low word of the output variable.

The high word of the output variable is filled with 16#0000.

If the value of the input variable is negative, QF = 1 is set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	INT	
Y	Output variable	0	UDINT	
QF	Input variable negative	0	0/1	

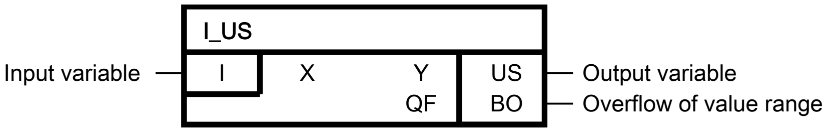
Project data

Can be inserted online	Yes
Special characteristics	-

5.18 I_US INTEGER to UNSIGNED SHORT INTEGER converter

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Conversion of an INTEGER variable to an UNSIGNED SHORT INTEGER variable

Operation mode

This block converts an INTEGER variable to an UNSIGNED SHORT INTEGER variable, i.e. the least significant byte of the DOUBLE INTEGER input variable is applied to output variable Y.

If the value of input variable X exceeds the value range of output variable Y, then QF = 1 is set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	INT	
Y	Output variable	0	USINT	
QF	Overflow of value range	0	0/1	

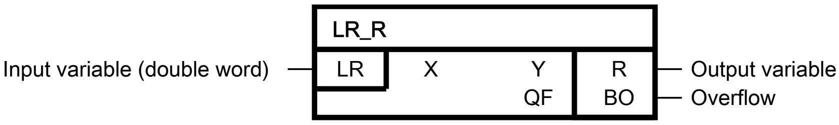
Project data

Can be inserted online	Yes
Special characteristics	-

5.19 LR_R LONG REAL to REAL converter

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

Conversion of a LONG REAL variable to a real variable

Operation mode

This block converts a long real variable to a real variable. The result is limited to the maximum range of data type REAL. If the output variable has been limited, then QF = 1 is set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable (double word)	0	LREAL	
Y	Output variable	0.0	REAL	
QF	Overflow	0	0/1	

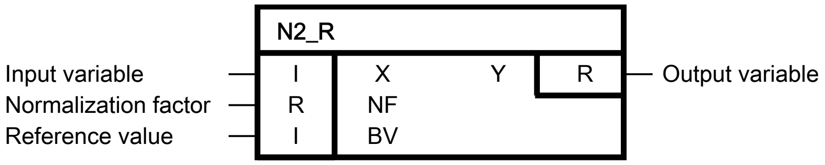
Project data

Can be inserted online	Yes
Special characteristics	-

5.20 N2_R Converting 16-bit fixed-point format (N2) to REAL

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Conversion of a 16-bit fixed-point variable to a REAL variable. For the case X and BV= 16348 (corresponds to 100% in normalized Profidrive representation), output Y assumes the value at input NF.

Operation mode

Input variable X is mapped to output Y according to the following formula:

$$Y = \frac{(X \cdot NF)}{BV}$$

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	INT	
NF	Normalization factor	1.0	REAL	
BV	Reference value	16384	INT	
Y	Output variable	0.0	REAL	

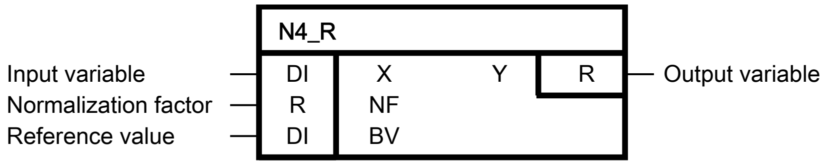
Project data

Can be inserted online	Yes
Special characteristics	-

5.21 N4_R Converting 32-bit fixed-point format (N4) to REAL

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Conversion of a 32-bit fixed-point variable to a REAL variable. For the case X and BV= 1073741824 (corresponds to 100% in normalized Profidrive representation), output Y assumes the value at input NF.

Operation mode

Input variable X is mapped to output Y according to the following formula:

$$Y = \frac{(X \cdot NF)}{BV}$$

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	DINT	
NF	Normalization factor	1.0	REAL	
BV	Reference value	1073741824	DINT	
Y	Output variable	0.0	REAL	

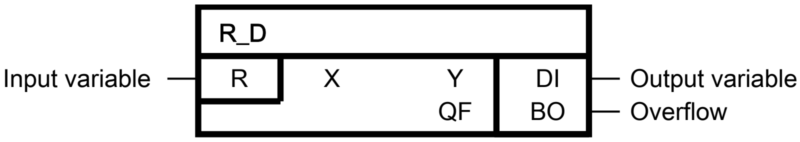
Project data

Can be inserted online	Yes
Special characteristics	-

5.22 R_D REAL to DOUBLE INTEGER converter

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Conversion of a REAL variable to a DOUBLE INTEGER variable

Operation mode

This block converts a real variable to a double integer variable. During the conversion, decimal places of the input variable are truncated.

Note: There is no rounding.

The result is limited to the data type of the output variable corresponding to -2^{31} or 2^{31}-1. If the output variable has been limited, then QF = 1 is set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
Y	Output variable	0	DINT	
QF	Overflow	0	0/1	

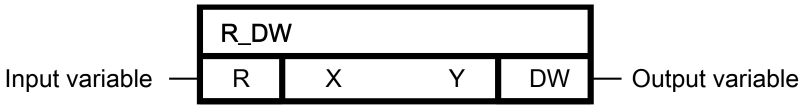
Project data

Can be inserted online	Yes
Special characteristics	-

5.23 R_DW Bit string transfer as DWORD

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

This block copies the bit string of the input variable to the output variable.

Operation mode

This block copies the bit string of input variable X to the output variable Y.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
Y	Output variable	16#00000000	DWORD	

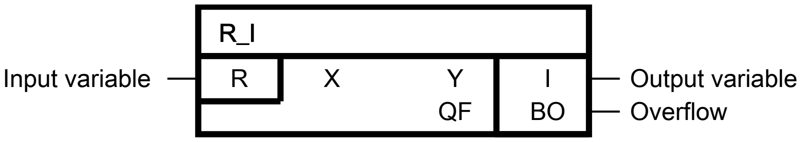
Project data

Can be inserted online	Yes
Special characteristics	-

5.24 R_I REAL to INTEGER converter

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Conversion of a REAL variable to an INTEGER variable

Operation mode

This block converts a real variable to an integer variable. During the conversion, decimal places of the input variable are truncated. The number is not rounded off. The result is limited to the data type of the output variable corresponding to +32767 or -32768. If the output variable has been limited, then QF = 1 is set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
Y	Output variable	0	INT	
QF	Overflow	0	0/1	

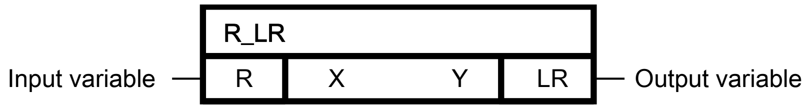
Project data

Can be inserted online	**Yes**
Special characteristics	-

5.25 R_LR REAL to LONG REAL converter

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

Conversion of a REAL variable to a LONG REAL variable

Operation mode

This block converts a real variable to a long real variable.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
Y	Output variable	0.0	LREAL	

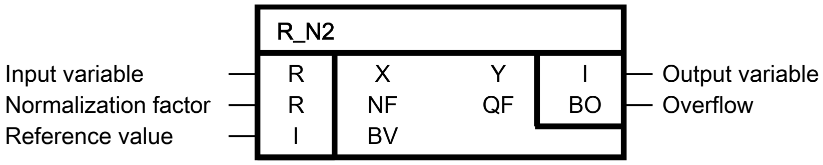
Project data

Can be inserted online	Yes
Special characteristics	-

5.26 R_N2 Converting REAL to 16-bit fixed-point format (N2)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Conversion of a REAL variable to a 16-bit fixed-point variable. For the case X = NF and BV = 16384 (default), output Y assumes the value 16384 (corresponds to 100% in normalized Profidrive representation).

Operation mode

Input variable X is mapped to output Y according to the following formula (result is rounded):

$$Y = \frac{X \cdot BV}{NF}$$

Y is limited to the range -32768 ≤ Y ≤ 32767 (corresponds to -200% ≤ Y < 200%).

Output QF (overflow) is set to ‘1’ if X cannot be mapped on Y because of a range violation, or if NF = 0 has been set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
NF	Normalization factor	1.0	REAL	
BV	Reference value	16384	INT	
Y	Output variable	0	INT	
QF	Overflow	0	0/1	

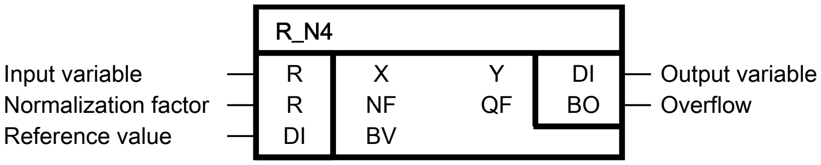
Project data

Can be inserted online	Yes
Special characteristics	-

5.27 R_N4 Converting REAL to 32-bit fixed-point format (N4)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Conversion of a REAL variable to a 32-bit fixed-point variable. For the case X = NF and BV = 1073741824 (default), output Y assumes the value 1073741824 (corresponds to 100%).

Operation mode

Input variable X is mapped to output Y according to the following formula (result is rounded):

$$Y = \frac{X \cdot BV}{NF}$$

Y is limited to the range -2147483648 ≤ Y ≤ 2147483647 (decimal) or 16#8000000 ≤ Y ≤ 16#7FFFFFF (hexadecimal) (corresponds to -200% ≤ Y < 200%).

Output QF (overflow) is set to ‘1’ if X cannot be mapped on Y because of a range violation, or if NF = 0 has been set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
NF	Normalization factor	1.0	REAL	
BV	Reference value	1073741824	DINT	
Y	Output variable	0	DINT	
QF	Overflow	0	0/1	

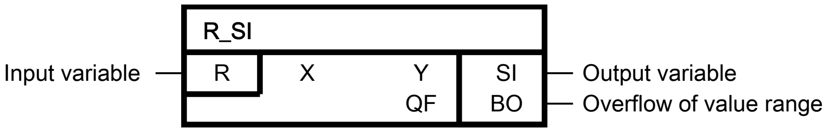
Project data

Can be inserted online	Yes
Special characteristics	-

5.28 R_SI REAL to SHORT INTEGER converter

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

Conversion of a REAL variable to a SHORT INTEGER variable

Operation mode

This block converts a real variable to a short integer variable. During the conversion, decimal places of the input variable are truncated. The number is not rounded off. The result is limited to the data type of the output variable corresponding to -128 or 127. If the output variable has been limited, then QF = 1 is set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
Y	Output variable	0	SINT	
QF	Overflow of value range	0	0/1	

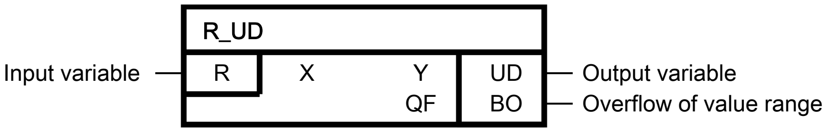
Project data

Can be inserted online	Yes
Special characteristics	-

5.29 R_UD REAL to UNSIGNED DOUBLE INTEGER converter

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Conversion of a REAL variable to an UNSIGNED DOUBLE INTEGER variable

Operation mode

This block converts a real variable to an unsigned double integer variable. During the conversion, decimal places of the input variable are truncated. The number is not rounded off. Depending on the data type of the output variable, the result is limited to 0 or 2^{32} -1. If the output variable has been limited, then QF = 1 is set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
Y	Output variable	0	UDINT	
QF	Overflow of value range	0	0/1	

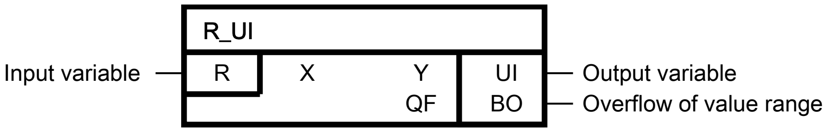
Project data

Can be inserted online	Yes
Special characteristics	-

5.30 R_UI REAL to UNSIGNED INTEGER converter

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Conversion of a REAL variable to an UNSIGNED INTEGER variable

Operation mode

This block converts a real variable to a unsigned integer variable. During the conversion, decimal places of the input variable are truncated. The number is not rounded off. Depending on the data type of the output variable, the result is limited to 0 or 2^{16}-1. If the output variable has been limited, then QF = 1 is set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
Y	Output variable	0	UINT	
QF	Overflow of value range	0	0/1	

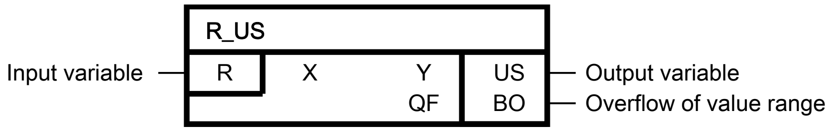
Project data

Can be inserted online	Yes
Special characteristics	-

5.31 R_US REAL to UNSIGNED SHORT INTEGER converter

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Conversion of a REAL variable to an UNSIGNED SHORT INTEGER variable

Operation mode

This block converts a real variable to an unsigned short integer variable.

During the conversion, decimal places of the input variable are truncated. The number is not rounded off. Depending on the data type of the output variable, the result is limited to 0 or 2^8-1. If the output variable has been limited, then QF = 1 is set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
Y	Output variable	0	USINT	
QF	Overflow of value range	0	0/1	

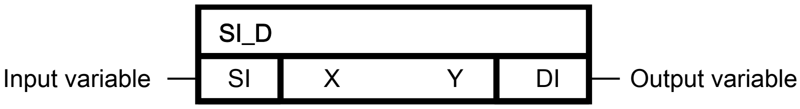
Project data

Can be inserted online	Yes
Special characteristics	-

5.32 SI_D SHORT INTEGER to DOUBLE INTEGER converter

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

Conversion of a SHORT INTEGER variable to a DOUBLE INTEGER variable

Operation mode

This block converts a short integer variable to a double integer variable.

The input variable of data type SHORT INTEGER is copied to the low-order byte of the output variable. If the input variable has a positive sign, high-order bytes of the output variable is filled with 16#00 - if, on the other hand, the sign is negative, the high-order bytes receive the value 16#FF.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	SINT	
Y	Output variable	0	DINT	

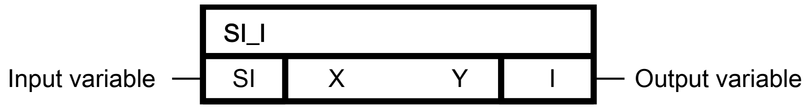
Project data

Can be inserted online	Yes
Special characteristics	-

5.33 SI_I SHORT INTEGER to INTEGER converter

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

Conversion of a SHORT INTEGER variable to an INTEGER variable

Operation mode

This block converts a SHORT INTEGER variable to an INTEGER variable.

The input variable of data type SHORT INTEGER is copied to the low-order byte of the output variable. If the input variable has a positive sign, the high-order byte of the output variable is filled with 16#00 - if, on the other hand, the sign is negative, the high-order byte contains the value 16#FF.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	SINT	
Y	Output variable	0	INT	

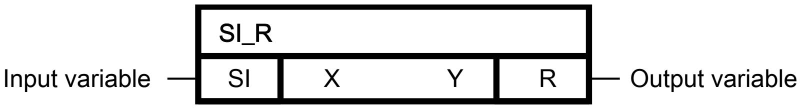
Project data

Can be inserted online	Yes
Special characteristics	-

5.34 SI_R SHORT INTEGER to REAL converter

☑ SIMOTION □ SINAMICS

Symbol



Short description

Conversion of a SHORT INTEGER variable to a REAL variable

Operation mode

This block converts a short integer variable to a real variable.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	SINT	
Y	Output variable	0.0	REAL	

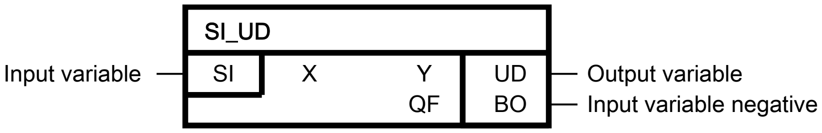
Project data

Can be inserted online	Yes
Special characteristics	-

5.35 SI_UD SHORT INTEGER to UNSIGNED DOUBLE INTEGER converter

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

Conversion of a SHORT INTEGER variable to an UNSIGNED DOUBLE INTEGER variable

Operation mode

This block converts a short integer variable to an unsigned double integer variable.

The input variable of data type SHORT INTEGER is copied to the low-order byte of the output variable. The high-order bytes of the output variable are filled with 16#00. If the value of the input variable is negative, QF = 1 is set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	SINT	
Y	Output variable	0	UDINT	
QF	Input variable negative	0	0/1	

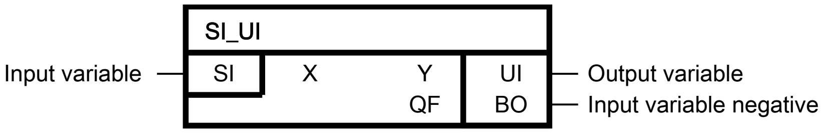
Project data

Can be inserted online	**Yes**
Special characteristics	-

5.36 SI_UI SHORT INTEGER to UNSIGNED INTEGER converter

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

Conversion of a SHORT INTEGER variable to an UNSIGNED INTEGER variable

Operation mode

This block converts a short integer variable to an unsigned integer variable.

The input variable of data type SHORT INTEGER is copied to the low-order byte of the output variable. The high-order byte of the output variable is filled with 16#00. If the value of the input variable is negative, QF = 1 is set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	SINT	
Y	Output variable	0	UDINT	
QF	Input variable negative	0	0/1	

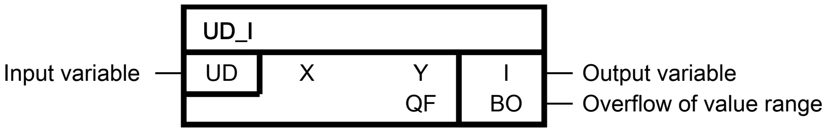
Project data

Can be inserted online	**Yes**
Special characteristics	-

5.37 UD_I UNSIGNED DOUBLE INTEGER to INTEGER converter

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Conversion of an UNSIGNED DOUBLE INTEGER variable to an INTEGER variable

Operation mode

This block converts an UNSIGNED DOUBLE INTEGER variable into an INTEGER variable, i.e. the low-order word of the UNSIGNED DOUBLE INTEGER input variable is taken over in output variable Y.

If the value of input variable X exceeds the value range of output variable Y, then QF = 1 is set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	UDINT	
Y	Output variable	0	INT	
QF	Overflow of value range	0	0/1	

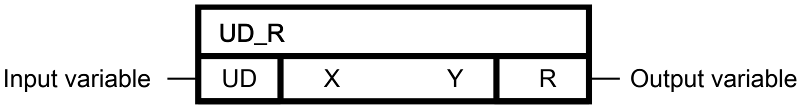
Project data

Can be inserted online	Yes
Special characteristics	-

5.38 UD_R UNSIGNED DOUBLE INTEGER to REAL converter

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Conversion of an UNSIGNED DOUBLE INTEGER variable to a REAL variable

Operation mode

This block converts an UNSIGNED DOUBLE INTEGER variable into a REAL variable.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	UDINT	
Y	Output variable	0.0	REAL	

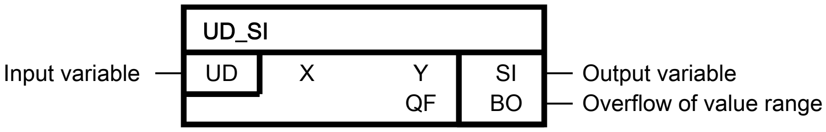
Project data

Can be inserted online	Yes
Special characteristics	-

5.39 UD_SI UNSIGNED DOUBLE INTEGER to SHORT INTEGER converter

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

Conversion of an UNSIGNED DOUBLE INTEGER variable to a SHORT INTEGER variable

Operation mode

This block converts an UNSIGNED DOUBLE INTEGER variable into a SHORT INTEGER variable, i.e. the low-order word of the UNSIGNED DOUBLE INTEGER input variable is taken over in output variable Y. If the value of input variable X exceeds the value range of output variable Y, QF = 1 is set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	UDINT	
Y	Output variable	0	SINT	
QF	Overflow of value range	0	0/1	

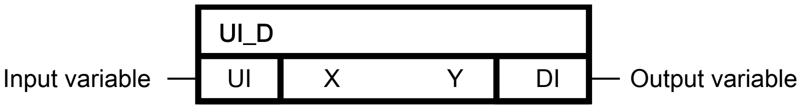
Project data

Can be inserted online	Yes
Special characteristics	-

5.40 UI_D UNSIGNED INTEGER to DOUBLE INTEGER converter

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Conversion of an UNSIGNED INTEGER variable to a DOUBLE INTEGER variable

Operation mode

This block converts an UNSIGNED INTEGER variable into a DOUBLE INTEGER variable.

The input variable of data type UNSIGNED INTEGER is copied to the low word of output variable Y. The high word is filled with 16#0000.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	UINT	
Y	Output variable	0	DINT	

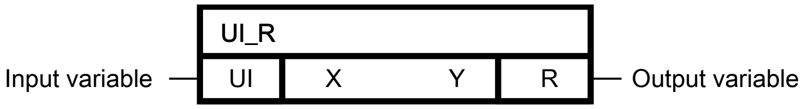
Project data

Can be inserted online	Yes
Special characteristics	-

5.41 UI_R UNSIGNED INTEGER to REAL converter

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Conversion of an UNSIGNED INTEGER variable to a REAL variable

Operation mode

This block converts an UNSIGNED INTEGER variable into a REAL variable.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	UINT	
Y	Output variable	0.0	REAL	

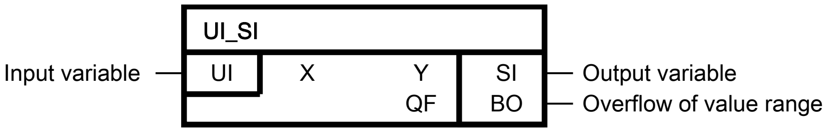
Project data

Can be inserted online	Yes
Special characteristics	-

5.42 UI_SI UNSIGNED INTEGER to SHORT INTEGER converter

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

Conversion of an UNSIGNED INTEGER variable to a SHORT INTEGER variable

Operation mode

This block converts an UNSIGNED INTEGER variable into a SHORT INTEGER variable, i.e. the low-order byte of the UNSIGNED INTEGER input variable is taken over in output variable Y. If the value of input variable X exceeds the value range of output variable Y, QF = 1 is set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	UINT	
Y	Output variable	0	SINT	
QF	Overflow of value range	0	0/1	

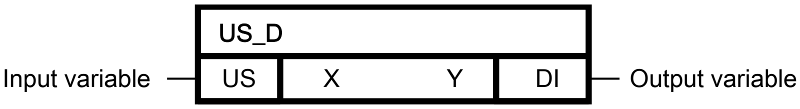
Project data

Can be inserted online	Yes
Special characteristics	-

5.43 US_D UNSIGNED SHORT INTEGER to DOUBLE INTEGER converter

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Conversion of an UNSIGNED SHORT INTEGER variable to a DOUBLE INTEGER variable

Operation mode

This block converts an UNSIGNED SHORT INTEGER variable into a DOUBLE INTEGER variable.

The input variable of data type UNSIGNED SHORT INTEGER is copied to the low-order byte of output variable Y. The remaining high-order bytes are filled with 16#00.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	USINT	
Y	Output variable	0	DINT	

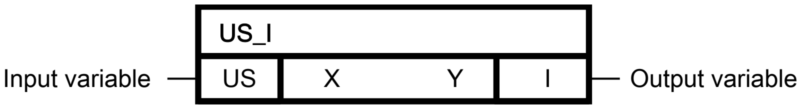
Project data

Can be inserted online	Yes
Special characteristics	-

5.44 US_I UNSIGNED SHORT INTEGER to INTEGER converter

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Conversion of an UNSIGNED SHORT INTEGER variable to an INTEGER variable

Operation mode

This block converts an UNSIGNED SHORT INTEGER variable into an INTEGER variable.

The input variable of data type UNSIGNED SHORT INTEGER is copied to the low-order byte of output variable Y. The remaining high-order bytes are filled with 16#00.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	USINT	
Y	Output variable	0	INT	

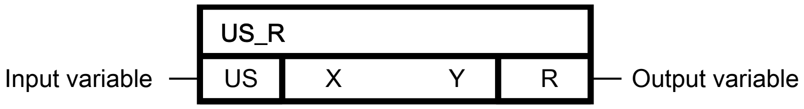
Project data

Can be inserted online	Yes
Special characteristics	-

5.45 US_R UNSIGNED SHORT INTEGER to REAL converter

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Conversion of an UNSIGNED SHORT INTEGER variable to a REAL variable

Operation mode

This block converts an UNSIGNED SHORT INTEGER variable into a REAL variable.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	USINT	
Y	Output variable	0.0	REAL	

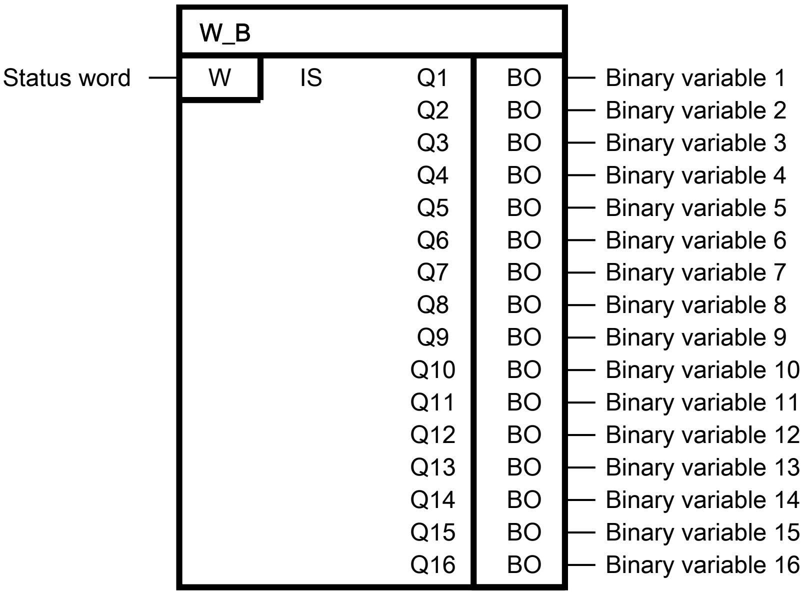
Project data

Can be inserted online	**Yes**
Special characteristics	-

5.46 W_B Converter, status word to 16 binary variables

☑ SIMOTION ☑ SINAMICS

Symbol



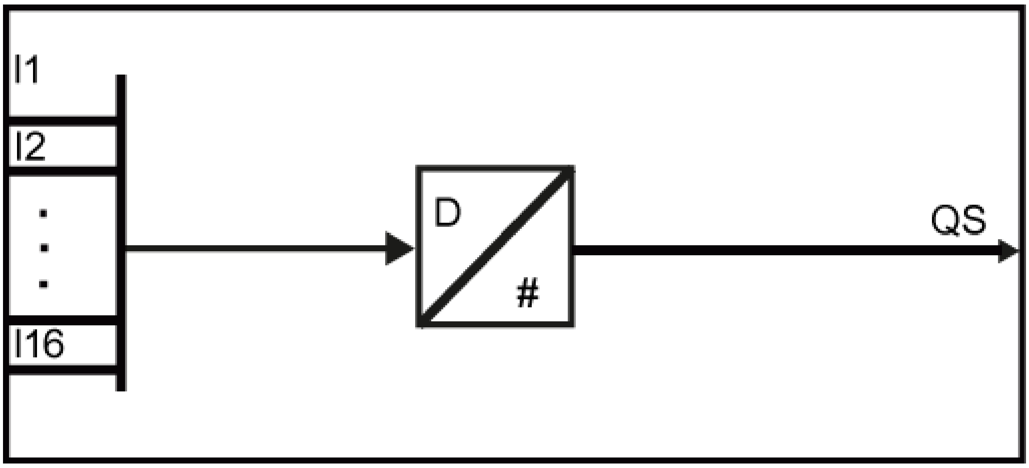
Short description

Status word decryption to 16 binary variables

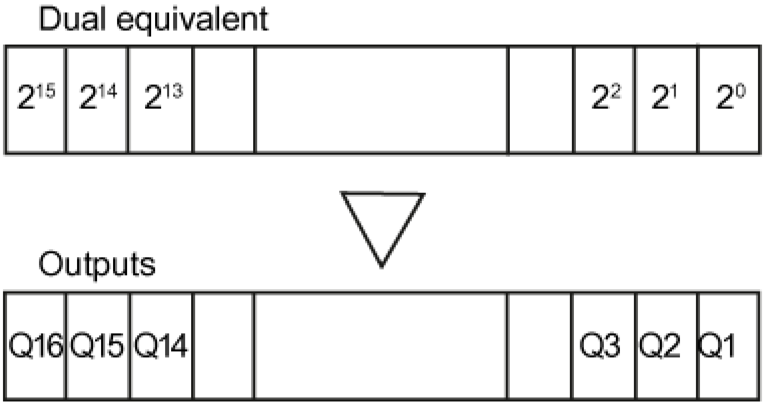
Operation mode

This block decodes the status word IS to 16 binary variables and gives the result to its outputs Q1 to Q16.

The binary variable of outputs Q1 to Q16 is assigned to each dual equivalent 2^0 to 2^{15} of the status word.



Conversion scheme



Block connections

Block connection	Description	Preassignment	Value range	Attributes
IS	Status word	16#0000	WORD	
Q1	Binary variable 1	0	0/1	
Q2	Binary variable 2	0	0/1	
Q3	Binary variable 3	0	0/1	
Q4	Binary variable 4	0	0/1	
Q5	Binary variable 5	0	0/1	
Q6	Binary variable 6	0	0/1	
Q7	Binary variable 7	0	0/1	
Q8	Binary variable 8	0	0/1	
Q9	Binary variable 9	0	0/1	
Q10	Binary variable 10	0	0/1	
Q11	Binary variable 11	0	0/1	
Q12	Binary variable 12	0	0/1	
Q13	Binary variable 13	0	0/1	
Q14	Binary variable 14	0	0/1	
Q15	Binary variable 15	0	0/1	
Q16	Binary variable 16	0	0/1	

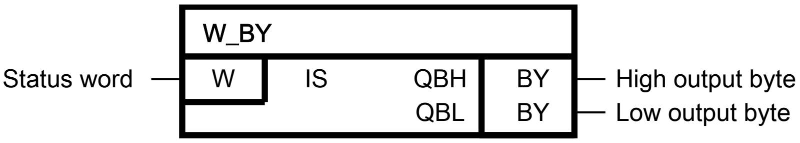
Project data

Can be inserted online	Yes
Special characteristics	-

5.47 W_BY Status word to status byte converter

☑ SIMOTION ☑ SINAMICS

Symbol



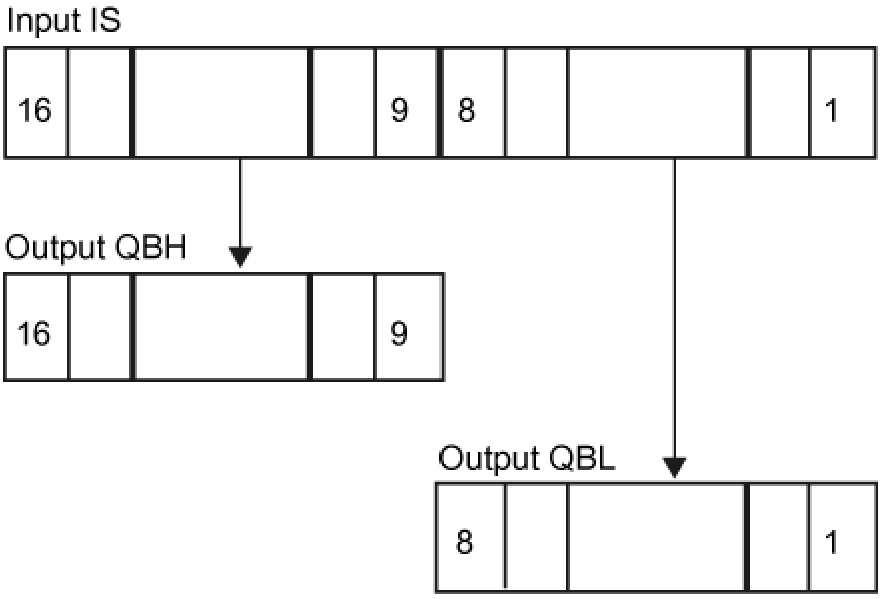
Short description

Conversion of a word to 2 bytes

Operation mode

This block splits the input word at IS into two bytes. These can be output to the I/O via the SBQ block. The high-order byte of the word at input IS is output at output QBH, and the low-order byte the word at input IS is output at output QBL (see conversion scheme below):

Conversion scheme



Block connections

Block connection	Description	Preassignment	Value range	Attributes
IS	Status word	16#0000	WORD	
QBH	High output byte	16#00	BYTE	
QBL	Low output byte	16#00	BYTE	

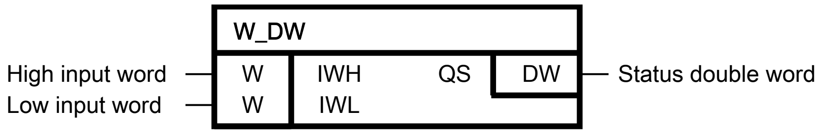
Project data

Can be inserted online	Yes
Special characteristics	-

5.48 W_DW Status word to status double word converter

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Two 16-bit words are copied to one 32-bit double word

Operation mode

The input variables are mapped according to the formula

QS = (IWL + IWH) * 2^{16}

to output QS.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
IWH	High input word	16#0000	WORD	
IWL	Low input word	16#0000	WORD	
QS	Status double word	16#00000000	DWORD	

Project data

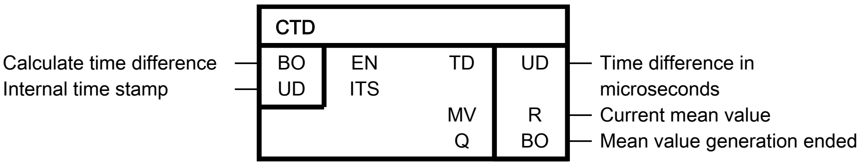
Can be inserted online	Yes
Special characteristics	-

System 6

6.1 CTD Time difference determination from an internal time stamp

☑ SIMOTION ☐ SINAMICS

Symbol


CTD
Calculate time difference
Internal time stamp
BO
UD
EN
ITS
TD
MV
Q
UD
R
BO
Time difference in microseconds
Current mean value
Mean value generation ended

Short description

Block for determining a time difference in microseconds.

Operation mode

If EN = 1, the time difference relative to time stamp ITS is determined and output at output TD. The time stamp IST must first be determined with the GTS block. The positive edge of EN starts the mean value generation of TD and the result is output at MV. After 10,000 mean value determinations, the mean value generation ends and output Q is set to 1. If input EN = 0 is set, the mean value generation and output Q is reset. Outputs TD and MV retain their last value.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
EN	Calculate time difference	0	0/1	
ITS	Internal time stamp	0	UDINT	
TD	Time difference in microseconds	0	UDINT	
MV	Current mean value	0	REAL	
Q	Mean value generation ended	0	0/1	

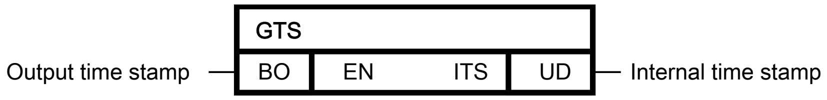
Project data

Can be loaded online	**Yes**
Special characteristics	-

6.2 GTS Reading out a time stamp

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

Blocks for reading out an internal time stamp for determination of runtimes. The determined time stamp can then be indicated at the CTD block for calculating a time difference in microseconds.

Operation mode

If EN = 1, an internal time stamp is determined and output at output TS. If EN = 0 is predefined, the last determined time stamp is output at TS.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
EN	Output time stamp	0	0/1	
ITS	Internal time stamp	0	UDINT	

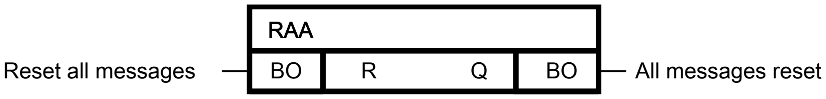
Project data

Can be loaded online	Yes
Special characteristics	-

6.3 RAA Reset all messages

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

All active messages are reset with the RAA (Reset all Alarms) block.

Operation mode

As long as input R = 1, all active messages are reset. Output Q indicates that the reset has been performed.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
R	Reset all messages	0	0/1	
Q	All messages reset	0	0/1	

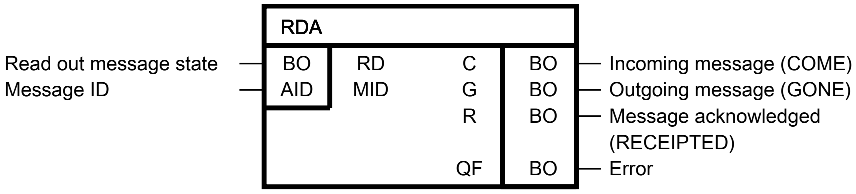
Project data

SIMOTION	4.3
SINAMICS	-
Can be inserted online	Yes
Special characteristics	-

6.4 RDA Reading out message

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

The block reads the state of a message and its acknowledgement state.

Operation mode

- The message is configured in SIMOTION SCOUT and referenced via a project-wide unique ID.
- Input MID contains the message ID, e.g. _alarm.Message.
- The state of the message is determined as long as input RD=1.
- A change of the message ID is possible. The state of the message ID specified at input MID is read out in each cycle.
- The outputs display the state of the message. The following combinations are possible:

C (incoming message)	G (outgoing message)	R (message acknowl-edged)	Meaning
0	1	0	Outgoing message, not acknowledged
1	0	0	Incoming message, not acknowledged
1	0	1	Incoming message, acknowledged
0	0	0	Message not in the message buffer *)

*) Message not in the message buffer - there are three reasons: - Message never triggered - Message triggered via AlarmS, but also gone - Message triggered via _AlarmSq, gone and acknowledged on display device

The outputs are refreshed as long as RD=1. With RD=0, the last state of the message buffer is retained.Output Q is set when an error occurs, e.g. Message ID not configured.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
RD	Read out message state	0	0/1	
MID	Message ID	STRUCTALARMID#NIL	StructAlarmId	
C	Incoming message (COME)	0	0/1	
G	Outgoing message (GONE)	0	0/1	
R	Message acknowledged (RECEIPTED)	0	0/1	
QF	Error	0	0/1	

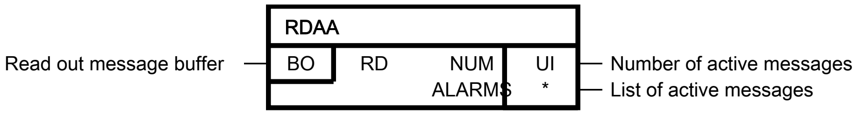
Project data

SIMOTION	4.3
SINAMICS	-
Can be inserted online	Yes
Special characteristics	-

6.5 RDAA Reading out of all messages

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

The list of active messages in the SIMOTION target device is read out.

Operation mode

- The reading out of all active messages is initiated with a rising edge at input RD.
- The number of active messages is returned at NUM output.
- A field of up to 40 active messages is displayed at the ALARMS output. The following is displayed for each alarm: - The ID of the message - The identifier for message not acknowledgeable (0), acknowledgeable message (1) - The state of the alarm: OUTGOING_ALARM (0), INCOMING_ALARM (1). The outputs are refreshed as long as RD=1. With RD=0, the last state of the message buffer is retained.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
RD	Read out message buffer	0	0/1	
NUM	Number of active messages	0	0..40	
ALARMS	List of active messages			
ALARMS[]	Up to 40 messages can be active	0		
ALARMS[].Id	Message ID	STRUCTALARMID#NIL	StructAlarmId	
ALARMS[].type	Corresponds to enumAlarmIdType (0: ALARM_S, 1: ALARM_SQ)	0	0/1	
ALARMS[].InOut	Corresponds to enumAlarmState OUTGOING_ALARM (0), INCOMING_ALARM (1)	0	0/1	

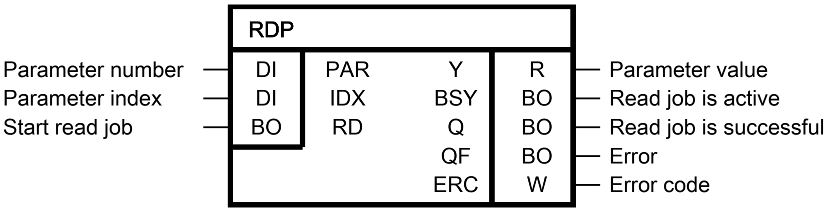
Project data

SIMOTION	4.3
SINAMICS	-
Can be inserted online	Yes
Special characteristics	-

6.6 RDP Reading drive parameters (REAL type)

□ SIMOTION ☑ SINAMICS

Symbol



Short description

The block enables the asynchronous reading of drive parameters of the REAL type on the local drive object.

Operation mode

The parameter number and the index of the parameter that is to be read are indicated at inputs PAR and IDX, respectively. If a parameter is not indexed, IDX = 0 must be set. The parameter is always read on the drive object on which the chart with the block is calculated. It is not possible to access parameters on several drive objects.

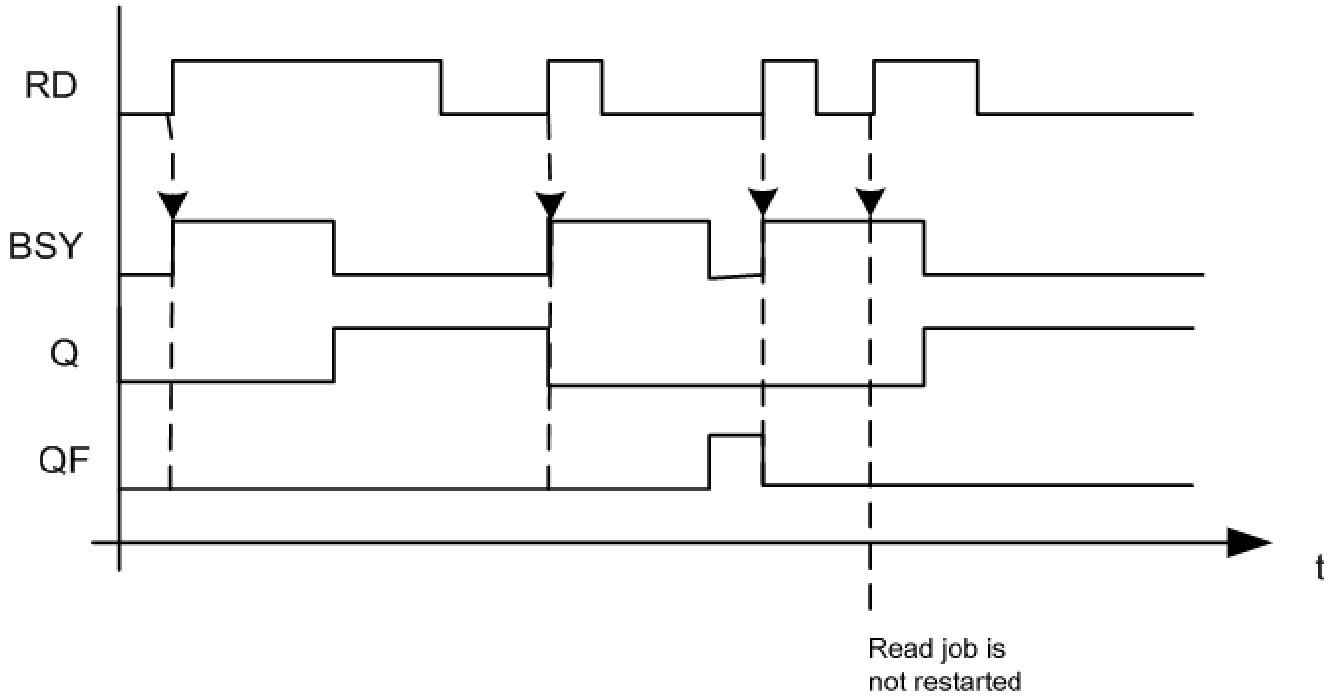
The asynchronous read job is started on a positive edge at input RD. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and can vary from job to job. During an active read job, any additional positive edges at input RD are ignored.

Output Q = 1 indicates that the parameter has been read successfully and the value is available at output Y. Y holds its value until a new value has been read. If an error occurs during the access, this is signaled with QF = 1. Output Y retains its last value.

For an error diagnostic, the error code ERC can be evaluated. ERC corresponds to the error code for parameter access according to PROFIdrive DPV1. The possible error codes can be found in Appendix A.2 of this document or in the SINAMICS Function Manual FH1 in Section PROFIBUS DP / PROFINET IO Communication and there in the Subsection Communication according to PROFIdrive → Acyclic communication → Configuration of the jobs and responses in Table Error values in DPV1 parameter responses.

ERC is only valid as long as QF = 1.

Time diagram



Quantity framework

Any number of asynchronous jobs of different block instances can be issued in parallel. Each block instance can only process one job.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
PAR	Parameter number	0	0..2^{16}	
IDX	Parameter index	0	0..2^{16}	
RD	Start read job	0	0/1	
Y	Parameter value	0.0	REAL	
BSY	Read job is active	0	0/1	
Q	Read job is successful	0	0/1	
QF	Error	0	0/1	
ERC	Error code	16#0000	DWORD	

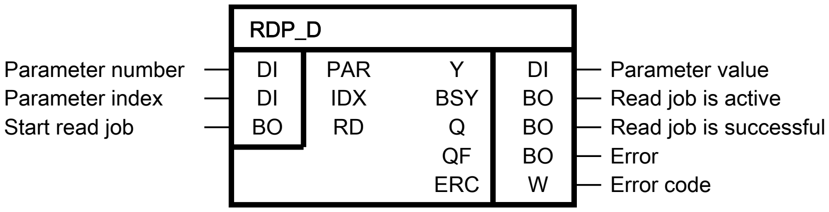
Project data

Can be loaded online	No
Special characteristics	-

6.7 RDP_D Reading drive parameters (DOUBLE INTEGER type)

□ SIMOTION ☑ SINAMICS

Symbol



Short description

The block enables the asynchronous reading of drive parameters of the DOUBLE INTEGER type on the local drive object.

Operation mode

The parameter number and the index of the parameter that is to be read are indicated at inputs PAR and IDX, respectively. If a parameter is not indexed, IDX = 0 must be set. The parameter is always read on the drive object on which the chart with the block is calculated. It is not possible to access parameters on several drive objects.

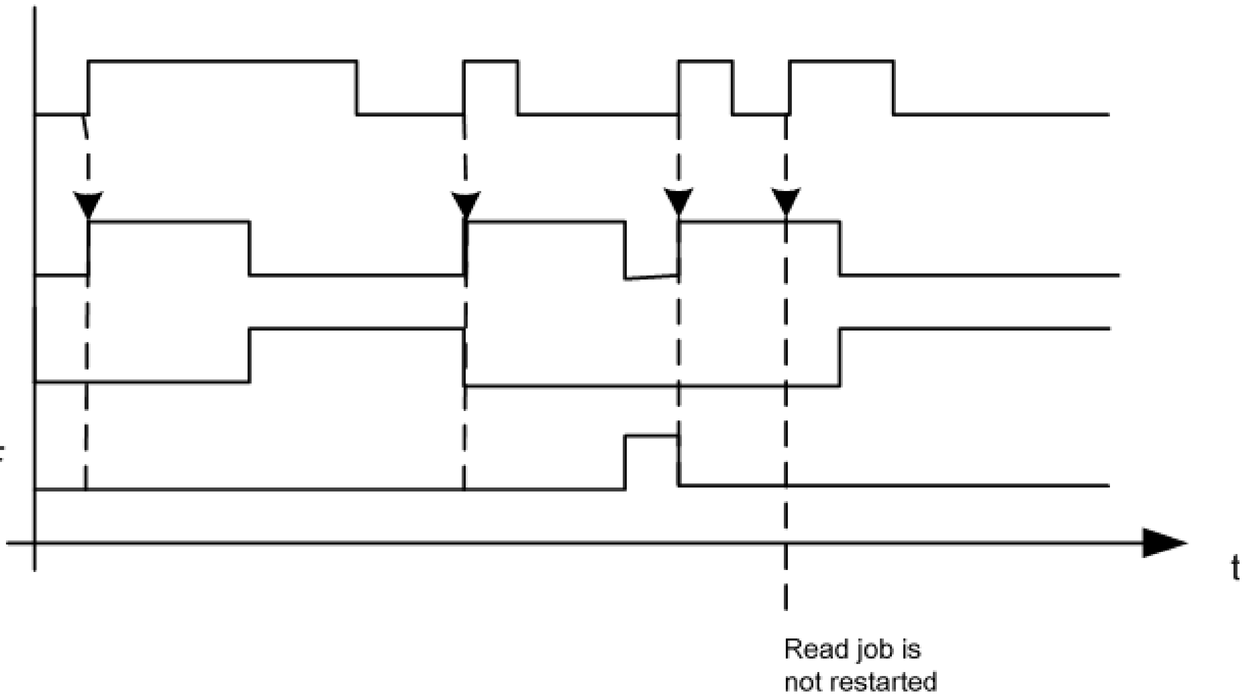
The asynchronous read job is started on a positive edge at input RD. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and can vary from job to job. During an active read job, any additional positive edges at input RD are ignored.

Output Q = 1 indicates that the parameter has been read successfully and the value is available at output Y. Y holds its value until a new value has been read. If an error occurs during the access, this is signaled with QF = 1. Output Y retains its last value.

For an error diagnostic, the error code ERC can be evaluated. ERC corresponds to the error code for parameter access according to PROFIdrive DPV1. The possible error codes can be found in Appendix A.2 of this document or in the SINAMICS Function Manual FH1 in Section PROFIBUS DP / PROFINET IO Communication and there in the Subsection Communication according to PROFIdrive → Acyclic communication → Configuration of the jobs and responses in Table Error values in DPV1 parameter responses.

ERC is only valid as long as QF = 1.

Time diagram



Quantity framework

Any number of asynchronous jobs of different block instances can be issued in parallel. Each block instance can only process one job.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
PAR	Parameter number	0	0..2^{16}	
IDX	Parameter index	0	0..2^{16}	
RD	Start read job	0	0/1	
Y	Parameter value	0	DINT	
BSY	Read job is active	0	0/1	
Q	Read job is successful	0	0/1	
QF	Error	0	0/1	
ERC	Error code	16#0000	WORD	

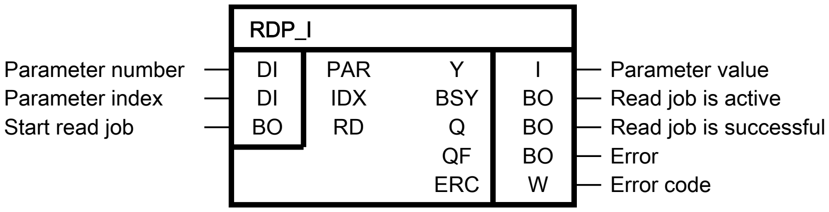
Project data

Can be loaded online	No
Special characteristics	-

6.8 RDP_I Reading drive parameters (INTEGER type)

□ SIMOTION ☑ SINAMICS

Symbol



Short description

The block enables the asynchronous reading of drive parameters of the INTEGER type on the local drive object.

Operation mode

The parameter number and the index of the parameter that is to be read are indicated at inputs PAR and IDX, respectively. If a parameter is not indexed, IDX = 0 must be set. The parameter is always read on the drive object on which the chart with the block is calculated. It is not possible to access parameters on several drive objects.

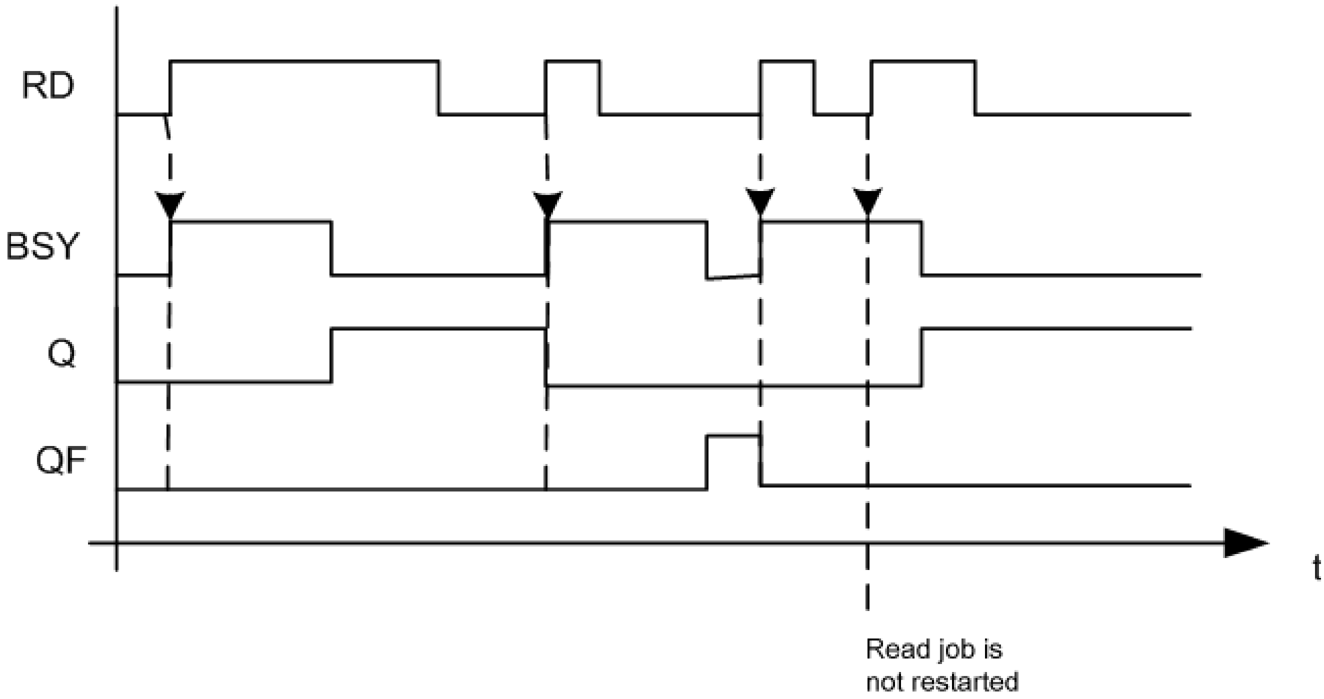
The asynchronous read job is started on a positive edge at input RD. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and can vary from job to job. During an active read job, any additional positive edges at input RD are ignored.

Output Q = 1 indicates that the parameter has been read successfully and the value is available at output Y. Y holds its value until a new value has been read. If an error occurs during the access, this is signaled with QF = 1. Output Y retains its last value.

For an error diagnostic, the error code ERC can be evaluated. ERC corresponds to the error code for parameter access according to PROFIdrive DPV1. The possible error codes can be found in Appendix A.2 of this document or in the SINAMICS Function Manual FH1 in Section PROFIBUS DP / PROFINET IO Communication and there in the Subsection Communication according to PROFIdrive → Acyclic communication → Configuration of the jobs and responses in Table Error values in DPV1 parameter responses.

ERC is only valid as long as QF = 1.

Time diagram



Quantity framework

Any number of asynchronous jobs of different block instances can be issued in parallel. Each block instance can only process one job.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
PAR	Parameter number	0	0..2^{16}	
IDX	Parameter index	0	0..2^{16}	
RD	Start read job	0	0/1	
Y	Parameter value	0	INT	
BSY	Read job is active	0	0/1	
Q	Read job is successful	0	0/1	
QF	Error	0	0/1	
ERC	Error code	16#0000	WORD	

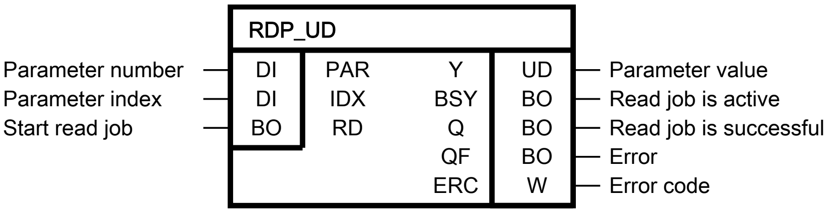
Project data

Can be loaded online	No
Special characteristics	-

6.9 RDP_UD Reading drive parameters (UNSIGNED DOUBLE INTEGER type)

□ SIMOTION ☑ SINAMICS

Symbol



Short description

RDP_UD (Read Parameter) enables the asynchronous reading of drive parameters of the UNSIGNED DOUBLE INTEGER type on the local drive object.

Operation mode

The parameter number and the index of the parameter that is to be read are indicated at inputs PAR and IDX, respectively. If a parameter is not indexed, IDX = 0 must be set. The parameter is always read on the drive object on which the chart with the block is calculated. It is not possible to access parameters on several drive objects.

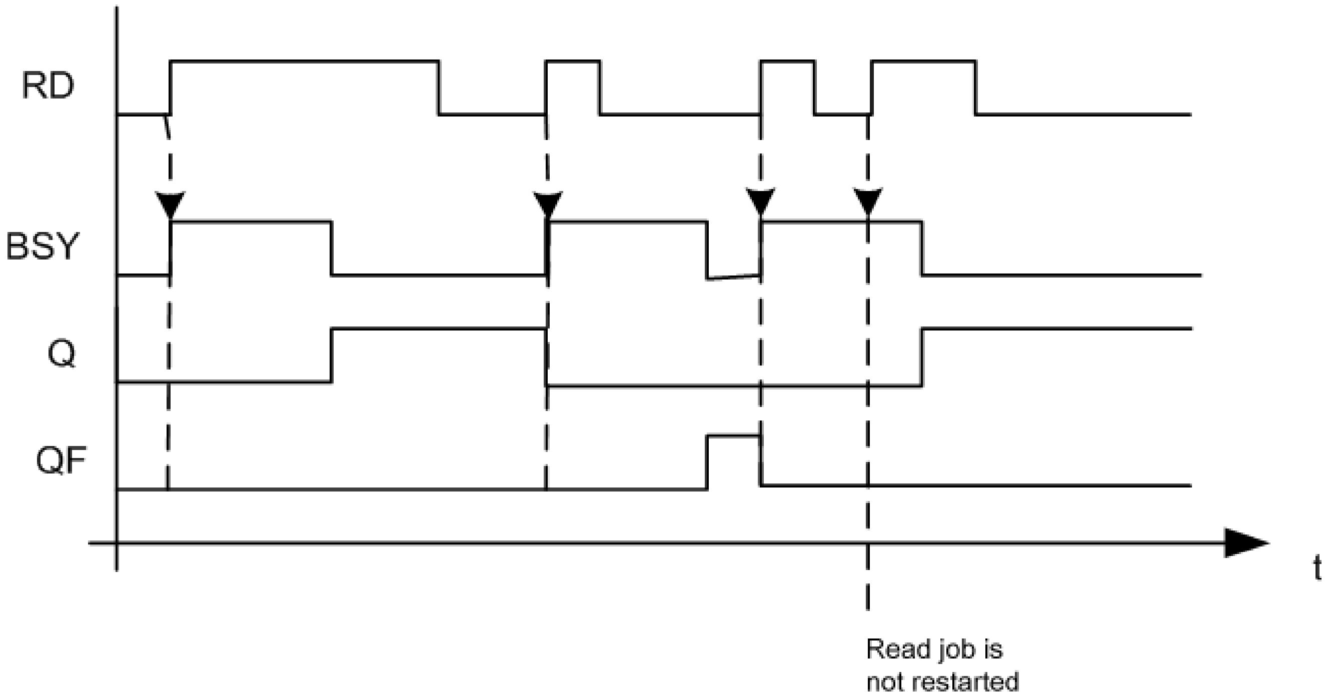
The asynchronous read job is started on a positive edge at input RD. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and can vary from job to job. During an active read job, any additional positive edges at input RD are ignored.

Output Q = 1 indicates that the parameter has been read successfully and the value is available at output Y. Y holds its value until a new value has been read. If an error occurs during the access, this is signaled with QF = 1. Output Y retains its last value.

For an error diagnostic, the error code ERC can be evaluated. ERC corresponds to the error code for parameter access according to PROFIdrive DPV1. The possible error codes can be found in Appendix A.2 of this document or in the SINAMICS Function Manual FH1 in Section PROFIBUS DP / PROFINET IO Communication and there in the Subsection Communication according to PROFIdrive → Acyclic communication → Configuration of the jobs and responses in Table Error values in DPV1 parameter responses.

ERC is only valid as long as QF = 1.

Time diagram



Quantity framework

Any number of asynchronous jobs of different block instances can be issued in parallel. Each block instance can only process one job.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
PAR	Parameter number	0	0..2^{16}	
IDX	Parameter index	0	0..2^{16}	
RD	Start read job	0	0/1	
Y	Parameter value	0	UDINT	
BSY	Read job is active	0	0/1	
Q	Read job is successful	0	0/1	
QF	Error	0	0/1	
ERC	Error code	16#0000	WORD	

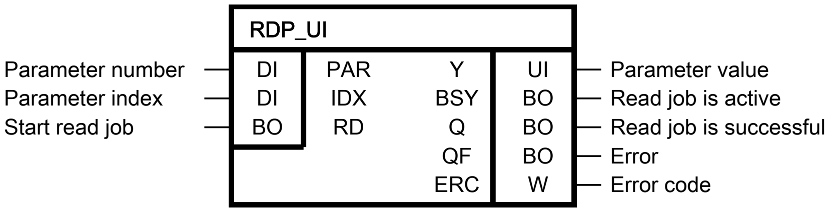
Project data

Can be loaded online	No
Special characteristics	-

6.10 RDP_UI Reading drive parameters (UNSIGNED INTEGER type)

□ SIMOTION ☑ SINAMICS

Symbol



Short description

RDP_UI (Read Parameter) enables the asynchronous reading of drive parameters of the UNSIGNED INTEGER type on the local drive object.

Operation mode

The parameter number and the index of the parameter that is to be read are indicated at inputs PAR and IDX, respectively. If a parameter is not indexed, IDX = 0 must be set. The parameter is always read on the drive object on which the chart with the block is calculated. It is not possible to access parameters on several drive objects.

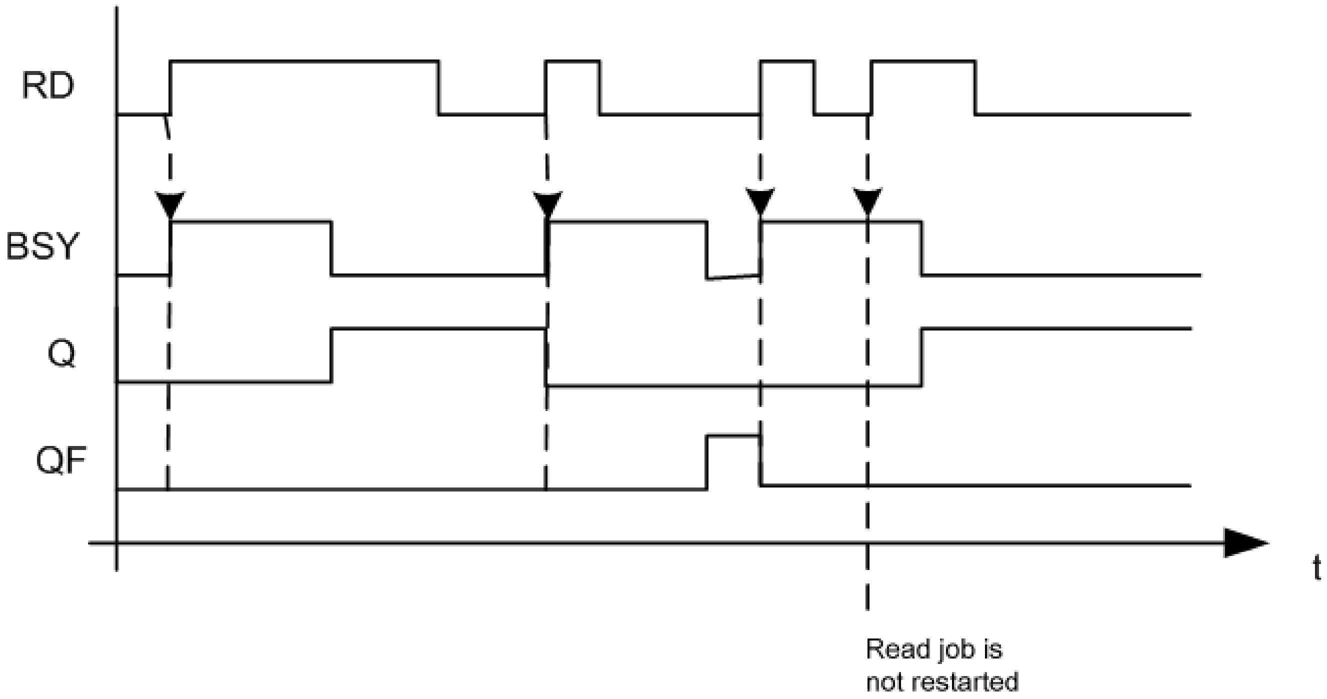
The asynchronous read job is started on a positive edge at input RD. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and can vary from job to job. During an active read job, any additional positive edges at input RD are ignored.

Output Q = 1 indicates that the parameter has been read successfully and the value is available at output Y. Y holds its value until a new value has been read. If an error occurs during the access, this is signaled with QF = 1. Output Y retains its last value.

For an error diagnostic, the error code ERC can be evaluated. ERC corresponds to the error code for parameter access according to PROFIdrive DPV1. The possible error codes can be found in Appendix A.2 of this document or in the SINAMICS Function Manual FH1 in Section PROFIBUS DP / PROFINET IO Communication and there in the Subsection Communication according to PROFIdrive → Acyclic communication → Configuration of the jobs and responses in Table Error values in DPV1 parameter responses.

ERC is only valid as long as QF = 1.

Time diagram



Quantity framework

Any number of asynchronous jobs of different block instances can be issued in parallel. Each block instance can only process one job.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
PAR	Parameter number	0	$0..2^{16}$	
IDX	Parameter index	0	$0..2^{16}$	
RD	Start read job	0	0/1	
Y	Parameter value	0	UINT	
BSY	Read job is active	0	0/1	
Q	Read job is successful	0	0/1	
QF	Error	0	0/1	
ERC	Error code	16#0000	WORD	

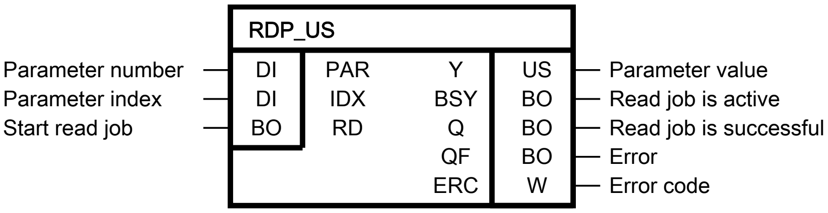
Project data

Can be loaded online	No
Special characteristics	-

6.11 RDP_US Reading drive parameters (UNSIGNED SHORT INTEGER type)

□ SIMOTION ☑ SINAMICS

Symbol



Short description

RDP_US (Read Parameter) enables the asynchronous reading of drive parameters of the UNSIGNED SHORT INTEGER type on the local drive object.

Operation mode

The parameter number and the index of the parameter that is to be read are indicated at inputs PAR and IDX, respectively. If a parameter is not indexed, IDX = 0 must be set. The parameter is always read on the drive object on which the chart with the block is calculated. It is not possible to access parameters on several drive objects.

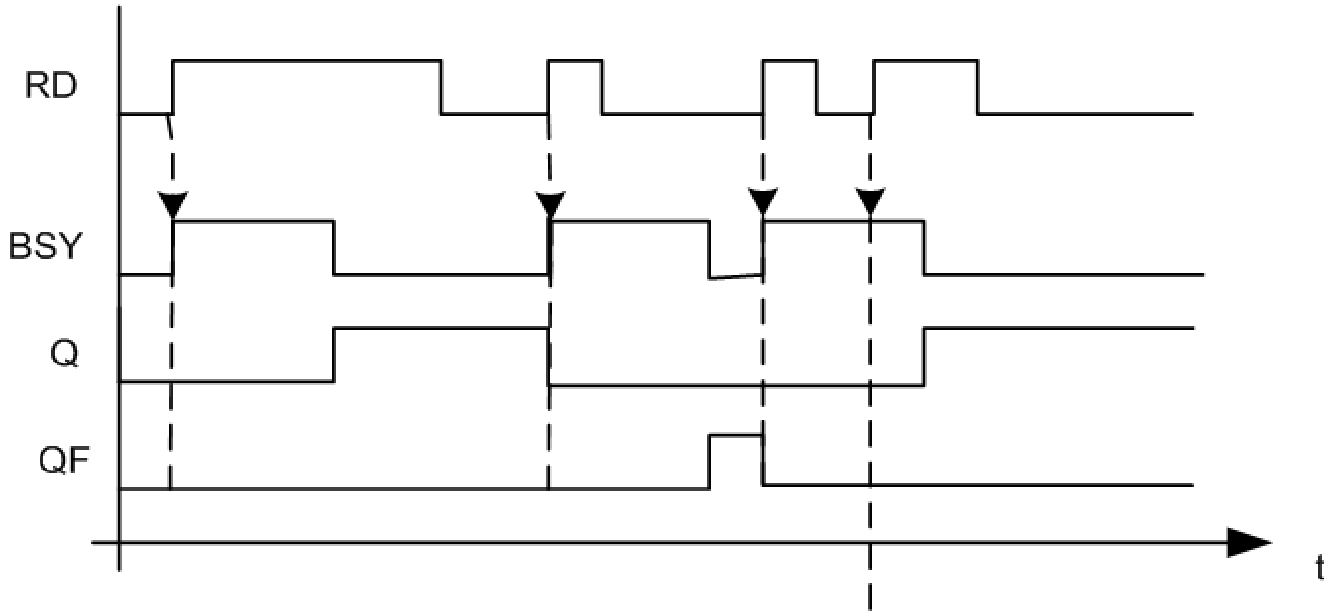
The asynchronous read job is started on a positive edge at input RD. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and can vary from job to job. During an active read job, any additional positive edges at input RD are ignored.

Output Q = 1 indicates that the parameter has been read successfully and the value is available at output Y. Y holds its value until a new value has been read. If an error occurs during the access, this is signaled with QF = 1. Output Y retains its last value.

For an error diagnostic, the error code ERC can be evaluated. ERC corresponds to the error code for parameter access according to PROFIdrive DPV1. The possible error codes can be found in Appendix A.2 of this document or in the SINAMICS Function Manual FH1 in Section PROFIBUS DP / PROFINET IO Communication and there in the Subsection Communication according to PROFIdrive → Acyclic communication → Configuration of the jobs and responses in Table Error values in DPV1 parameter responses.

ERC is only valid as long as QF = 1.

Time diagram



Read job is
not restarted

Quantity framework

Any number of asynchronous jobs of different block instances can be issued in parallel. Each block instance can only process one job.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
PAR	Parameter number	0	$0..2^{16}$	
IDX	Parameter index	0	$0..2^{16}$	
RD	Start read job	0	0/1	
Y	Parameter value	0	USINT	
BSY	Read job is active	0	0/1	
Q	Read job is successful	0	0/1	
QF	Error	0	0/1	
ERC	Error code	16#0000	WORD	

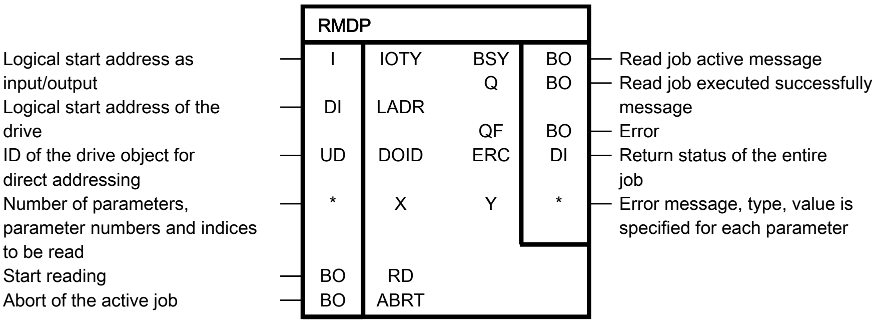
Project data

Can be loaded online	No
Special characteristics	-

6.12 RMDP Reading of drive parameters from the controller

☑ SIMOTION ☐ SINAMICS

Symbol


RMDP
Logical start address as input/output
I
IOTY
Logical start address of the drive
DI
LADR
ID of the drive object for direct addressing
UD
DOID
Number of parameters, parameter numbers and indices to be read
*
X
Start reading
BO
RD
Abort of the active job
BO
ABRT
BSY
BO
Read job active message
Q
BO
Read job executed successfully message
QF
BO
Error
ERC
DI
Return status of the entire job
Y
*
Error message, type, value is specified for each parameter

Short description

The RMDP block allows up to 39 SINAMICS parameters to be read from the DCC SIMOTION program.Only SINAMICS drives are supported.For error diagnostics, the error code ERC can be evaluated. ERC corresponds to the error code for parameter access according to PROVIdrive DPV1.The possible error codes can be found in Appendix A.2 of this document or in the SIMOTION Communication System Manual in Section PROFIdrive and there in the Subsection Acyclic communication (Base Mode Parameter Access) → Error evaluation in table Error codes in Base Mode Parameter Access responses.

The RMDP block is available as of SIMOTION V4.2.

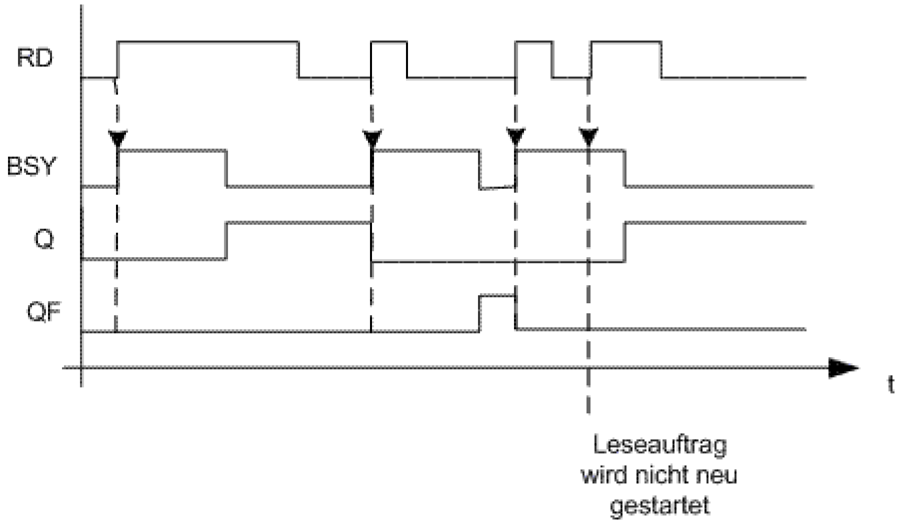
Operation mode

First the block inputs for addressing the drive are entered as well as the selection of the parameters to be read. The asynchronous read job is started by the positive edge at input RD. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and communication load and can vary from job to job. During an active read job, any additional positive edges at input RD are ignored.The actual reading/writing of the parameters is not performed in the DCC task. The block instance only controls the communication command. The results of the read-write job must be polled at the block outputs in the following task cycles. The evaluation is performed via global variables or user-defined block types.The output Q = 1 indicates that the parameters have been read successfully and the values are available at output Y. Y retains its value until a new job has been completed successfully. If an error occurs during an access, this is signaled with QF = 1. Output Y retains its last value. For an error diagnostic, the error code ERC can be evaluated.

If the entire job is successful (output Q=1) and individual jobs are not successful (ERC not equal to 0), the values that are read from the drive are displayed. The other parameters with ERC = 0 have been read error-free.

The error status of the individual read jobs can be evaluated on the parameter-specific return value PRES.An active job is aborted with the positive edge on the ABRT input. The ABRT signal must have the value 1 for at least one cycle.

Time diagram



Short description

The block enables up to 39 SINAMICS parameters to be read.

The following data sets can be used to read out the parameters:With PROFIBUS (external or integrated), data set 47 is always read out irrespective of whether the function is called with a valid (0>=doId<=254) or invalid'doId' (doId=255).With PROFINET, two data sets are available:− Base Mode Parameter Access - local (data set 0xB02E)This data set (DS) is used for SIMOTION if no or an invalid 'doId' (doId=255) is specified in the function. Access to the appropriate DO is then performed via the Parameter Access Point (PAP). The address of the PAP can be specified directly or the log. address of the cyclic data specified (e.g. 256 for a DO axis). SIMOTION then determines the associated PAP from this address and then accesses the correct address.PAP must always be at subslot 1 (configuration in HW Config).− Base Mode Parameter Access - global (data set 0xB02F)This data set is used if a valid 'doId' (0>=doId<=254) is entered. Any valid PAP or address can be specified because the assignment is only performed via the 'doId'.

Description of the block inputs

'IOTY': Input/output assignment of the logical start address of the drive.With 198: INPUT, the logical address of the drive is in the input range.With 199: OUTPUT, the logical address of the drive is in the output range.Diagnostics addresses are always of the INPUT type.

'LADR': Specification of the logical start address of the drive. If the optional parameter DOID is also used, any arbitrary address of the station (preferably the diagnostics address of the station) can be specified.With PROFINET, parameter access is via the Parameter Access Point (PAP) of a drive object.As an alternative to the logical start address of the drive, specification of the diagnostics address of the associated PAP is recommended.

'DOID': For the direct addressing of a drive object. Under the following conditions, the DOID may not be specified, or specified with an invalid value (>254):- Access via the DOID is not supported by the DP slave / I/O device (P978 not implemented).- Data set 0xB02F is not supported (PROFINET only).- Access is to be performed via the Parameter Access Point of a DO (PROFINET only).'X': The parameters to be read are specified under input X.

'X.NUMP': Number of parameters to be read.

'X.PAR[].NUM': Specification of the parameter number from which the values are to be read.

'X.PAR[].IDX': Parameter index; for indexed values, 0 means index 0. For non-indexed values, parameter index 0 must be specified.

'RD': Start read job.

'ABRT': Abort active job.

Description of the block outputs

'Q': Job completed without errors.

'QF': Job completed with errors.

'ERC': Corresponds to the values of the return value 'functionResult' of the _readDriveMultiParameter function. 'Y': Description of the parameter values. An error code, the data type and the value is read out for each parameter. Further information on the return value parameterResult can be found in the SIMOTION List Manual System Functions/Variables Devices → System Functions - Devices 1 → _readDriveMultiParameterDescription

'Y[].PRES': Corresponds to the parameter-specific return value. Coding corresponds to the return parameter parameterResult of the _readDriveMultiParameter ST function. Further information on the return parameter parameterResult can be found in the SIMOTION List Manual System Functions/ Variables Devices → System Functions - Devices 1 → _readDriveMultiParameterDescription

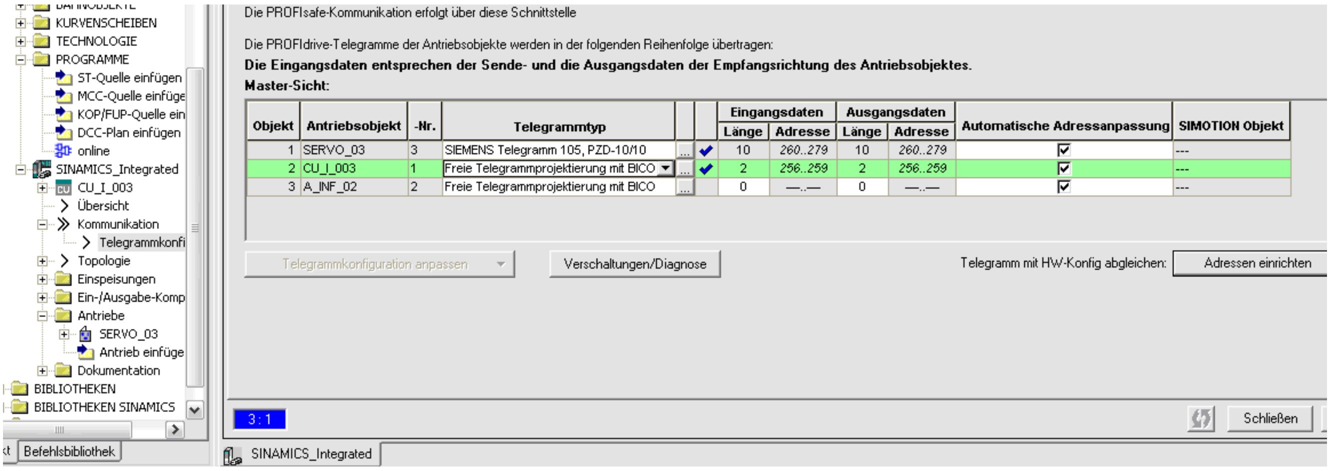
'Y[].DTYP': Returns the data type of the parameter (for the coding, see PROFIdrive profile).

'Y[].VAL': Parameter values read from the drive; the data type results from the returned data type. A conversion block must be called for different data types. When accessing REAL parameters, the conversion is performed via the conversion block DW_R.

Parameterization example

In order to be able to read certain parameters of a drive object (in the example: SERVO_03), proceed as follows:

First set the correct message frame configuration.



Die PROFIsafe-Kommunikation erfolgt über diese Schnittstelle

Die PROFIdrive-Telegramme der Antriebsobjekte werden in der folgenden Reihenfolge übertragen:

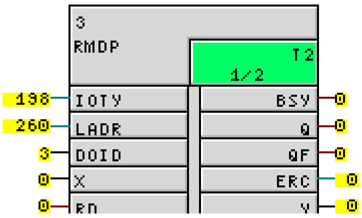
Die Eingangsdaten entsprechen der Sende- und die Ausgangsdaten der Empfangsrichtung des Antriebsobjektes.

Master-Sicht:

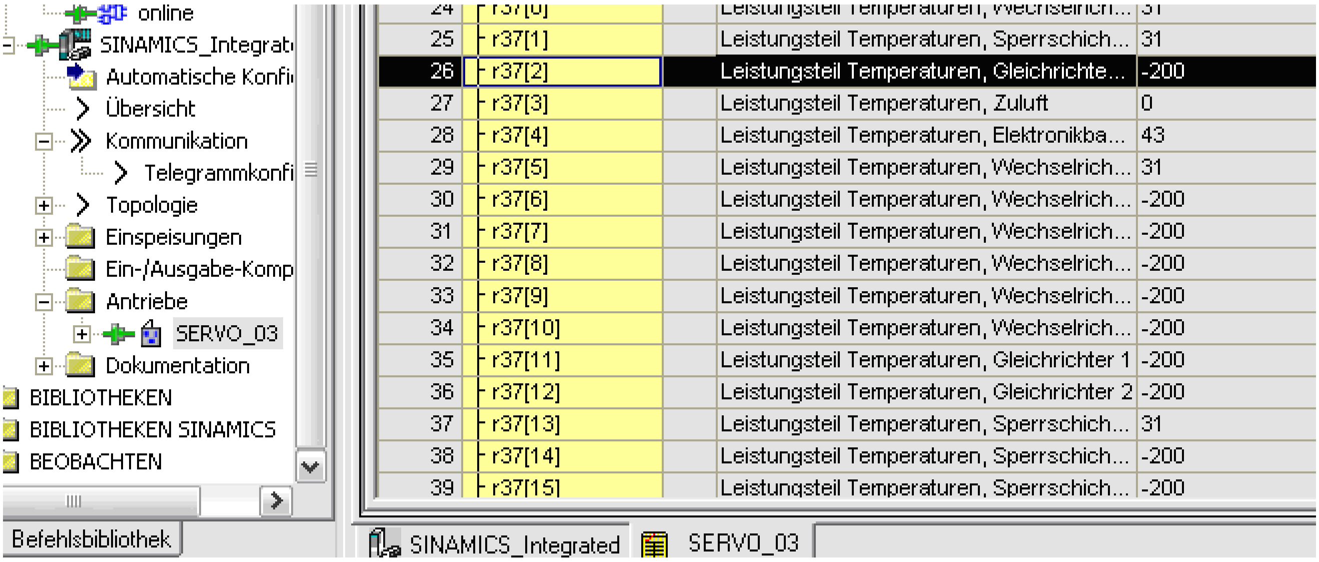
Objekt	Antriebsobjekt	-Nr.	Telegrammtyp			Eingangsdaten		Ausgangsdaten		Automatische Adressanpassung	SIMOTION Objekt
						Länge	Adresse	Länge	Adresse		
1	SERVO_03	3	SIEMENS Telegramm 105, PZD-10/10	...	✓	10	260..279	10	260..279	☑	---
2	CU_I_003	1	Freie Telegrammprojektierung mit BICO	...	✓	2	256..259	2	256..259	☑	---
3	A_INF_02	2	Freie Telegrammprojektierung mit BICO	...		0	—..—	0	—..—	☑	---

Telegramm mit HW-Konfig abgleichen:

Then set the desired DO address in the RMDP block. To do this, set the block input 'LADR' to the address (260) set in the message frame and the block input 'DOID' to the number (3) set in the message frame.

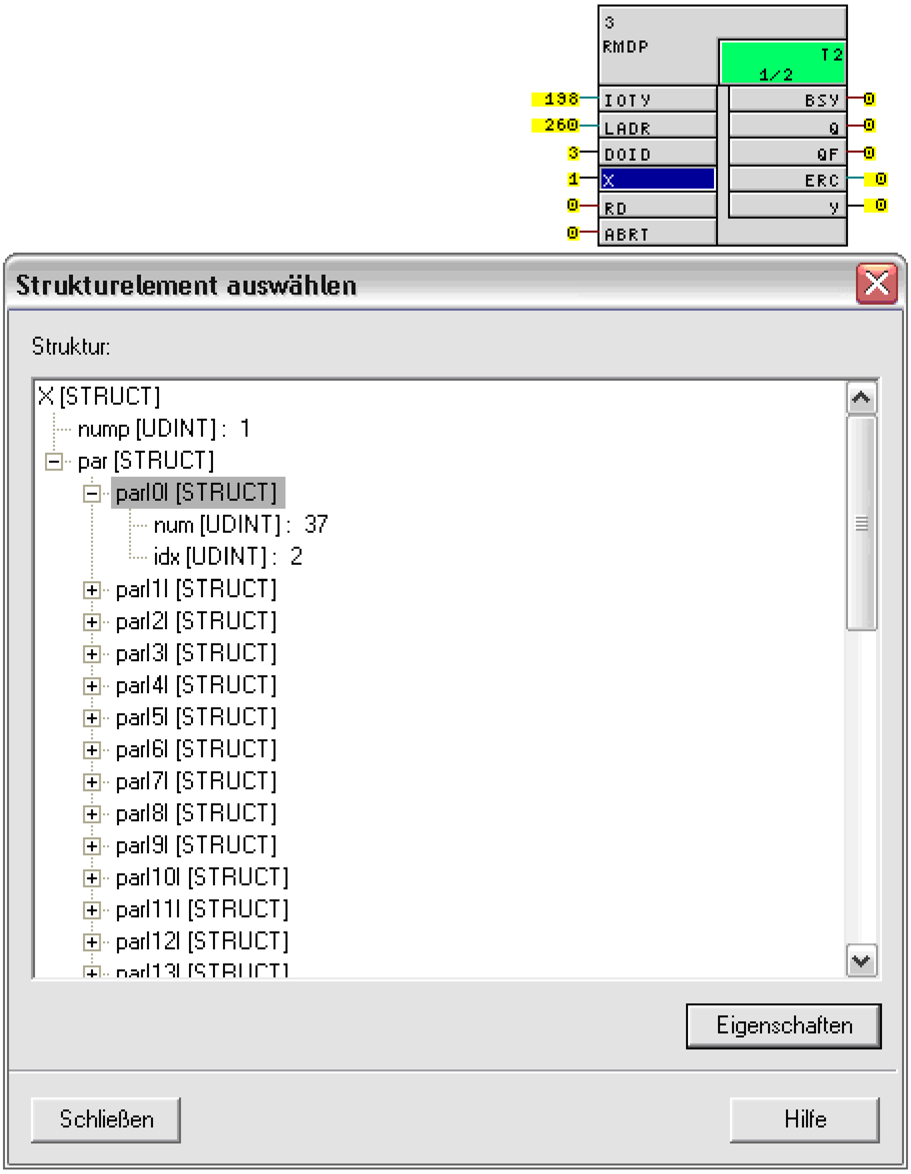


Select the parameters to be read at block input 'X', e.g. r37(2) power unit temperatures, rectifier maximum value in the expert list.



24	r37[0]	Leistungsteil Temperaturen, Wechselrich...	31
25	r37[1]	Leistungsteil Temperaturen, Sperrschich...	31
26	r37[2]	Leistungsteil Temperaturen, Gleichrichte...	-200
27	r37[3]	Leistungsteil Temperaturen, Zuluft	0
28	r37[4]	Leistungsteil Temperaturen, Elektronikba...	43
29	r37[5]	Leistungsteil Temperaturen, Wechselrich...	31
30	r37[6]	Leistungsteil Temperaturen, Wechselrich...	-200
31	r37[7]	Leistungsteil Temperaturen, Wechselrich...	-200
32	r37[8]	Leistungsteil Temperaturen, Wechselrich...	-200
33	r37[9]	Leistungsteil Temperaturen, Wechselrich...	-200
34	r37[10]	Leistungsteil Temperaturen, Wechselrich...	-200
35	r37[11]	Leistungsteil Temperaturen, Gleichrichter 1	-200
36	r37[12]	Leistungsteil Temperaturen, Gleichrichter 2	-200
37	r37[13]	Leistungsteil Temperaturen, Sperrschich...	31
38	r37[14]	Leistungsteil Temperaturen, Sperrschich...	-200
39	r37[15]	Leistungsteil Temperaturen, Sperrschich...	-200

To do this, double-click block input 'X', select the first structure element and enter the parameter number (37) at 'num' and the index (2) at 'idx'.

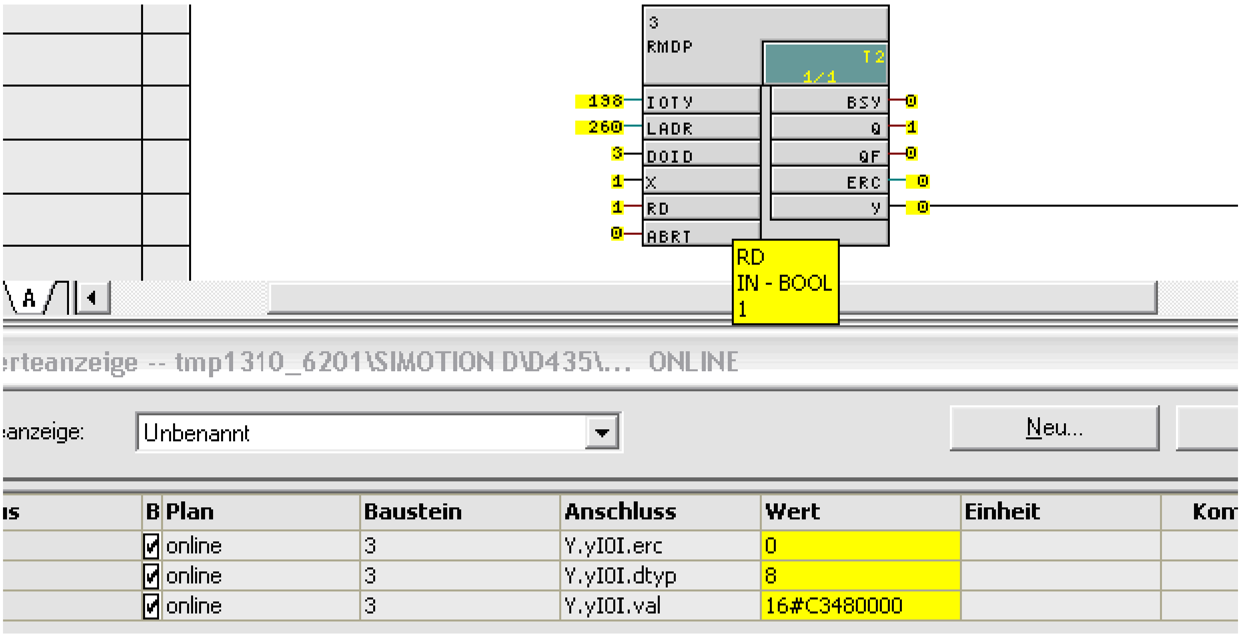


Strukturelement auswählen

Struktur:

```
X [STRUCT]
  nump [UDINT] :  1
  par [STRUCT]
    par[0] [STRUCT]
      num [UDINT] :  37
      idx [UDINT] :  2
    par[1] [STRUCT]
    par[2] [STRUCT]
    par[3] [STRUCT]
    par[4] [STRUCT]
    par[5] [STRUCT]
    par[6] [STRUCT]
    par[7] [STRUCT]
    par[8] [STRUCT]
    par[9] [STRUCT]
    par[10] [STRUCT]
    par[11] [STRUCT]
    par[12] [STRUCT]
    par[13] [STRUCT]
```

Then set the block input 'RD' to 1 to start the reading.



rteanzeige -- tmp1310_6201\SIMOTION D\D435\... ONLINE

anzeige: Unbenannt

ıs	B	Plan	Baustein	Anschluss	Wert	Einheit	Kor
	☑	online	3	Y.yIOI.erc	0		
	☑	online	3	Y.yIOI.dtyp	8		
	☑	online	3	Y.yIOI.val	16#C3480000		

Block connections

Block connection	Description	Preassignment	Value range	Attributes
IOTY	Logical start address as input/output	0	0: Not valid 198: Inputaddress 199: Outputaddress	
LADR	Logical start address of the drive	-1	DINT	
DOID	ID of the drive object for direct addressing	255	0 .. 254, 255: Not valid	
X	Number of parameters, parameter numbers and indices to be read			
X.NUMP	Number of parameters to be read	1	1..39	
X.PAR	Description of a parameter			
X.PAR[].NUM	Parameter number	1	1..65535	
X.PAR[].IDX	Parameter index	0	0..65535	
RD	Start reading	0	0/1	
ABRT	Abort of the active job	0	0/1	
BSY	Read job active message	0	0/1	
Q	Read job executed successfully message	0	0/1	
QF	Error	0	0/1	
ERC	Return status of the entire job	16#0000	DWORD	
Y	Error message, type, value is specified for each parameter			
Y[].PRES	Parameter-specific return value	0	DINT	
Y[].DTYP	Data type of the read parameter	0	USINT	
Y[].VAL	Parameter value read from the drive	0	DWORD	

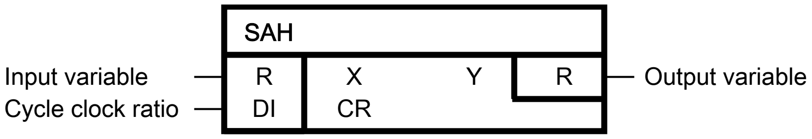
Project data

SIMOTION	as of V4.2
SINAMICS	-
Can be loaded online	No
Process context	Cyclic, equidistant
Special characteristics	-

6.13 SAH Sample & hold (REAL type)

□ SIMOTION ☑ SINAMICS

Symbol



Short description

Sample & hold block for the equidistant value transfer (REAL type) between blocks with different sampling times.

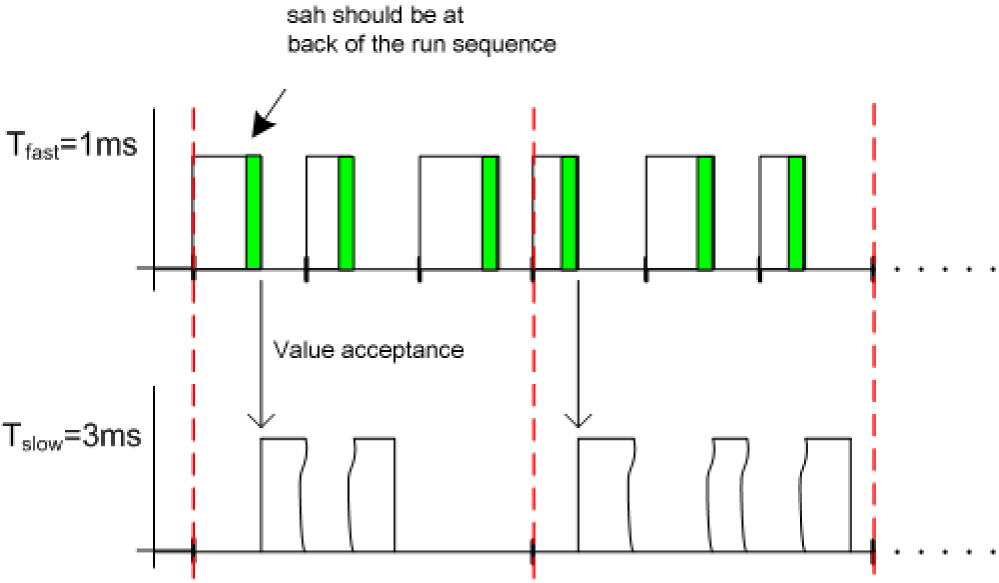
Operation mode

The value of the input variable X is taken over in the output variable Y in every CR cycle. The value transfer cycle is synchronized with the cycle control point of the execution system. The cycle control point defines the cycle clock in which all sampling times of the execution system are restarted.

A value transfer takes place every CR cycles relative to the cycle control point. The absolute value of CR is always generated for the cycle clock ratio. In the special case of CR = 0, the block behaves as for CR = 1. The block must always be configured in the faster sampling time. If the value from the slower scan time is transferred, it should be at the very beginning of the run sequence. If the value is to taken from the faster into the slower sampling time, the block should be computed last in the execution sequence.

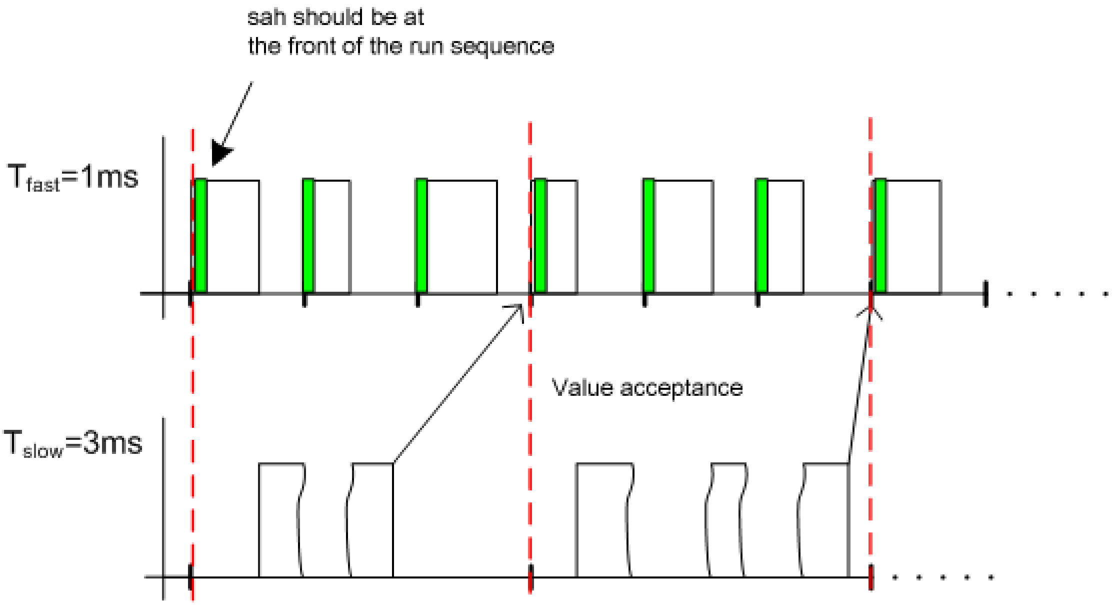
The following figure shows the transfer of values from a 1 ms level to a 3 ms level. The variation with time is shown for the calculation of the execution group.

$$CR = \frac{3\,ms}{1\,ms} = 3$$



The following figure shows the transfer of values from a 3 ms level to a 1 ms level. The variation with time is shown for the calculation of the execution group.

$$CR = \frac{3\,ms}{1\,ms} = 3$$



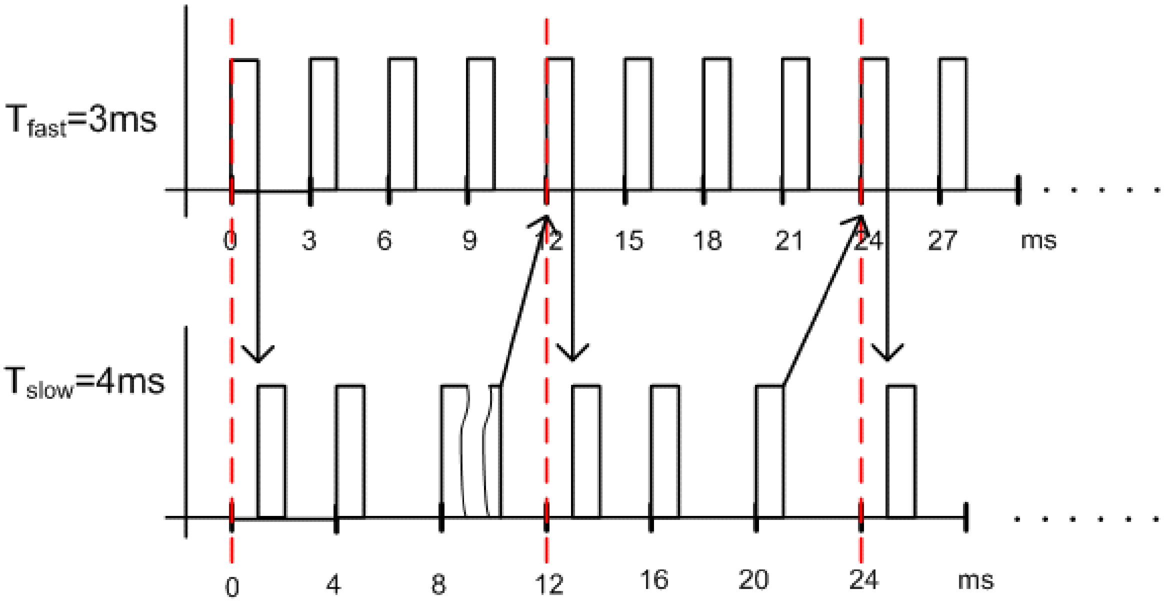
If the slower scan time is not a multiple of the faster scan time, the value can only be transferred consistently if both scanning procedures are restarted synchronously after CR cycles. This corresponds to the least common multiple of both scan times. CR is then calculated as follows:

$$CR = \frac{g(T_{fast}, T_{slow})}{T_{fast}}$$

$g(T_{fast}, T_{slow})$: least common multiple

The following shows the value transfer for T_{fast} = 3 ms and T_{slow} = 4 ms. The value transfer is made in both directions.

$$CR = \frac{g(3\,ms, 4\,ms)}{3\,ms} = \frac{12\,ms}{3\,ms} = 4$$



To enable values to be transferred at an optimal speed, it is, therefore, recommended that the slower scan time is always a multiple of the faster scan time.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
CR	Cycle clock ratio	1	0 - (2^{31}-1)	
Y	Output variable	0.0	REAL	

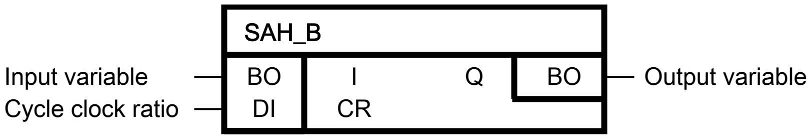
Project data

Can be loaded online	Yes
Special characteristics	-

6.14 SAH_B Sample & hold (BOOL type)

□ SIMOTION ☑ SINAMICS

Symbol



Short description

Sample & hold block for the equidistant value transfer (BOOL type) between blocks with different sampling times.

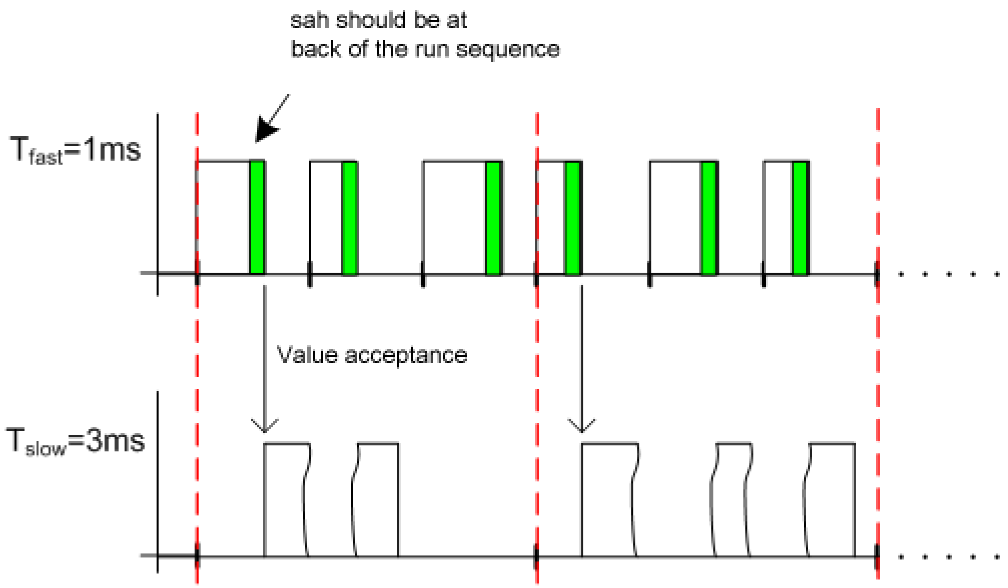
Operation mode

The value of the input variable I is taken over in the output variable Q in every CR cycle. The value transfer cycle is synchronized with the cycle control point of the execution system. The cycle control point defines the cycle clock in which all sampling times of the execution system are restarted.

A value transfer takes place every CR cycles relative to the cycle control point. The absolute value of CR is always generated for the cycle clock ratio. In the special case of CR = 0, the block behaves as for CR = 1. The block must always be configured in the faster sampling time. If the value from the slower scan time is transferred, it should be at the very beginning of the run sequence. If the value is to taken from the faster into the slower sampling time, the block should be computed last in the execution sequence.

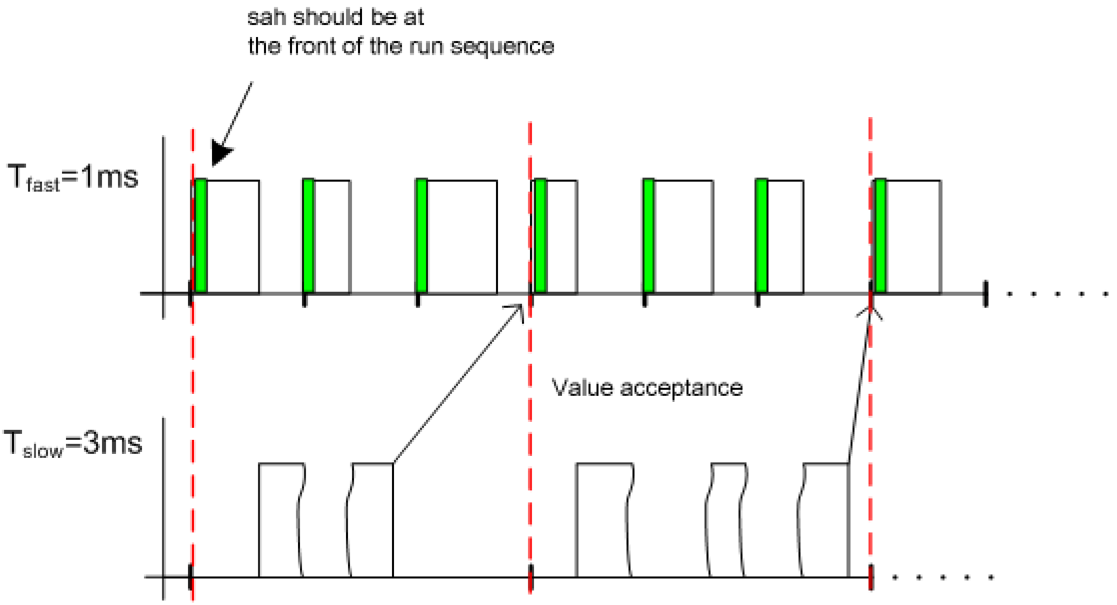
The following figure shows the transfer of values from a 1 ms level to a 3 ms level. The variation with time is shown for the calculation of the execution group.

$$CR = \frac{3\,ms}{1\,ms} = 3$$



The following figure shows the transfer of values from a 3 ms level to a 1 ms level. The variation with time is shown for the calculation of the execution group.

$$CR = \frac{3\,ms}{1\,ms} = 3$$



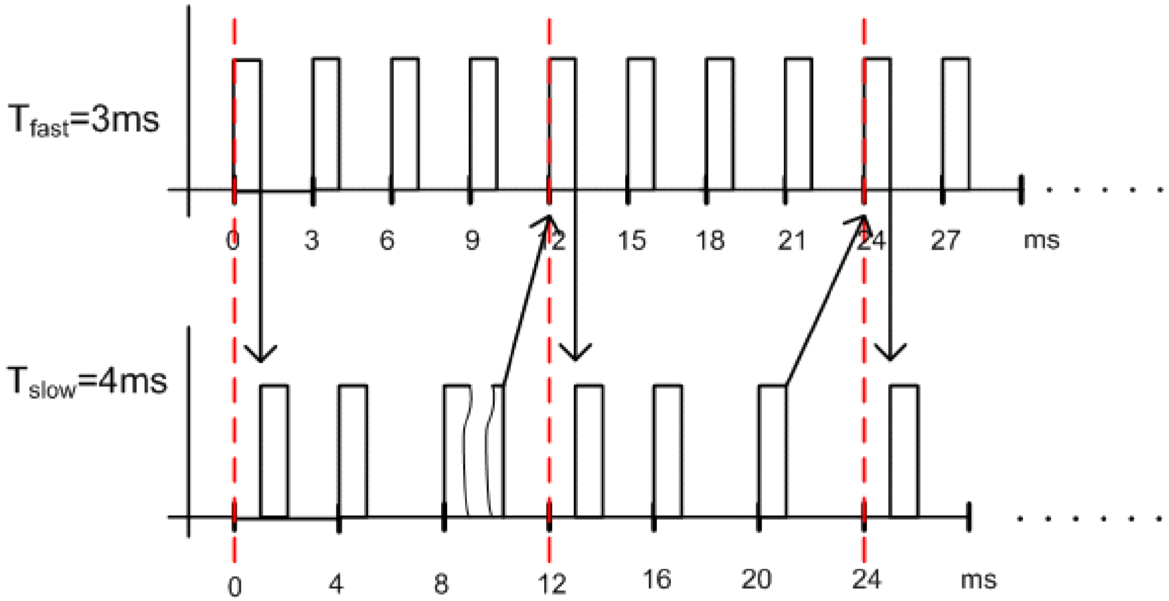
If the slower scan time is not a multiple of the faster scan time, the value can only be transferred consistently if both scanning procedures are restarted synchronously after CR cycles. This corresponds to the least common multiple of both scan times. CR is then calculated as follows:

$$CR = \frac{g(T_{fast}, T_{slow})}{T_{fast}}$$

$g(T_{fast}, T_{slow})$: least common multiple

The following shows the value transfer for T_{fast} = 3 ms and T_{slow} = 4 ms. The value transfer is made in both directions.

$$CR = \frac{g(3\,ms,\,4\,ms)}{3\,ms} = \frac{12\,ms}{3\,ms} = 4$$



To enable values to be transferred at an optimal speed, it is, therefore, recommended that the slower scan time is always a multiple of the faster scan time.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
I	Input variable	0	0/1	
CR	Cycle clock ratio	1	0 - (2^{31}-1)	
Q	Output variable	0	0/1	

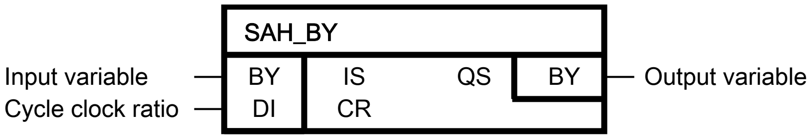
Project data

Can be loaded online	Yes
Special characteristics	-

6.15 SAH_BY Sample & hold (BYTE type)

□ SIMOTION ☑ SINAMICS

Symbol



Short description

Sample & hold block for the equidistant value transfer (BYTE type) between blocks with different sampling times.

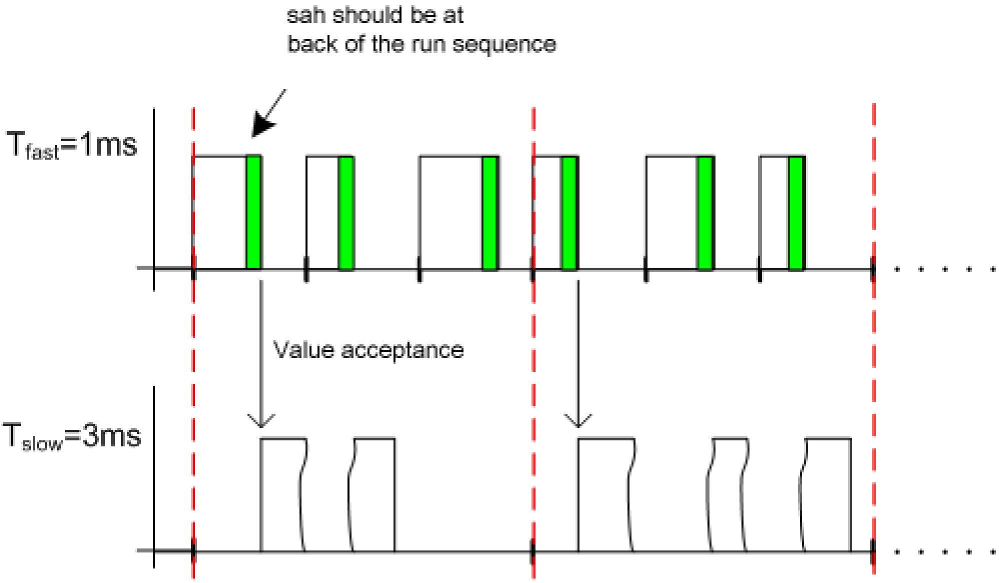
Operation mode

The value of the input variable IS is taken over in the output variable QS in every CR cycle. The value transfer cycle is synchronized with the cycle control point of the execution system. The cycle control point defines the cycle clock in which all sampling times of the execution system are restarted.

A value transfer takes place every CR cycles relative to the cycle control point. The absolute value of CR is always generated for the cycle clock ratio. In the special case of CR = 0, the block behaves as for CR = 1. The block must always be configured in the faster sampling time. If the value from the slower scan time is transferred, it should be at the very beginning of the run sequence. If the value is to taken from the faster into the slower sampling time, the block should be computed last in the execution sequence.

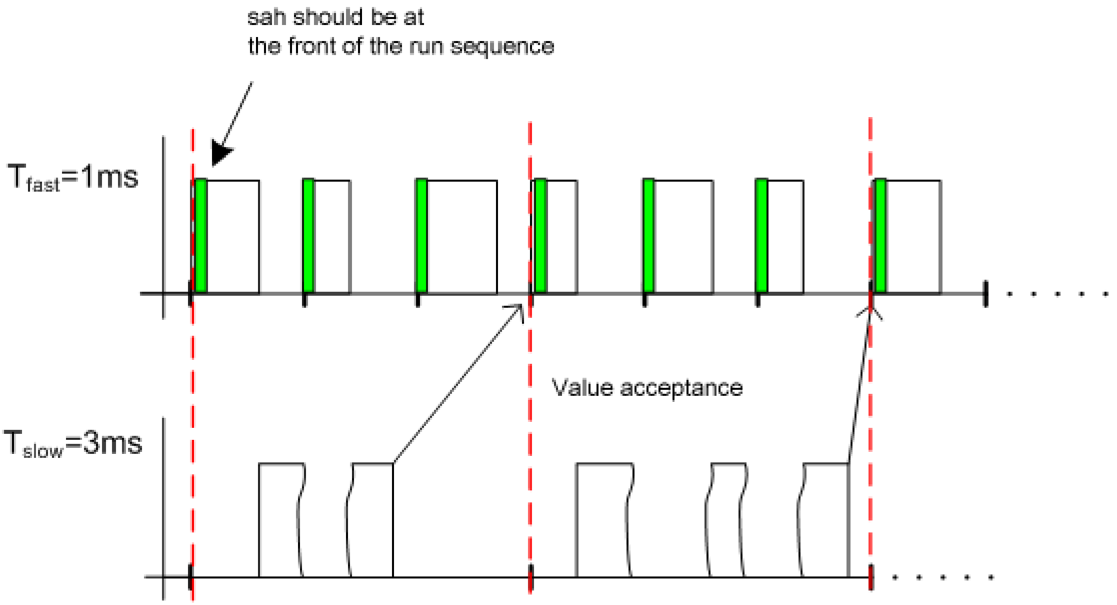
The following figure shows the transfer of values from a 1 ms level to a 3 ms level. The variation with time is shown for the calculation of the execution group.

$$CR = \frac{3\,ms}{1\,ms} = 3$$



The following figure shows the transfer of values from a 3 ms level to a 1 ms level. The variation with time is shown for the calculation of the execution group.

$$CR = \frac{3\,ms}{1\,ms} = 3$$



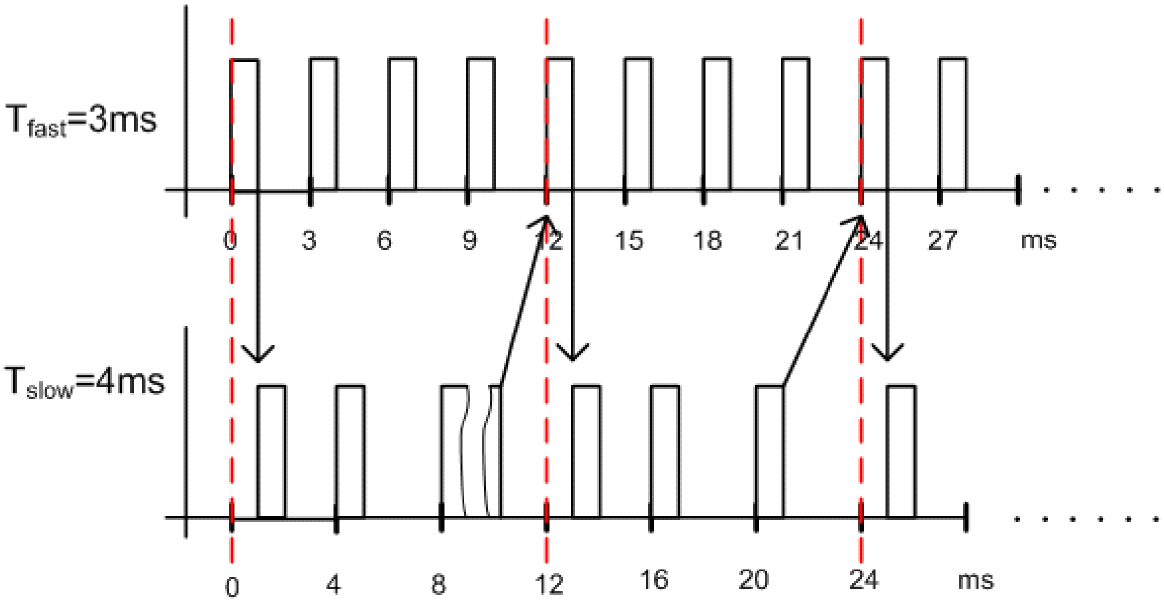
If the slower scan time is not a multiple of the faster scan time, the value can only be transferred consistently if both scanning procedures are restarted synchronously after CR cycles. This corresponds to the least common multiple of both scan times. CR is then calculated as follows:

$$CR = \frac{g(T_{fast}, T_{slow})}{T_{fast}}$$

$g(T_{fast}, T_{slow})$: least common multiple

The following shows the value transfer for T_{fast} = 3 ms and T_{slow} = 4 ms. The value transfer is made in both directions.

$$CR = \frac{g(3\,ms,\,4\,ms)}{3\,ms} = \frac{12\,ms}{3\,ms} = 4$$



To enable values to be transferred at an optimal speed, it is, therefore, recommended that the slower scan time is always a multiple of the faster scan time.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
IS	Input variable	16#00	BYTE	
CR	Cycle clock ratio	1	0 - (2^{31}-1)	
QS	Output variable	16#00	BYTE	

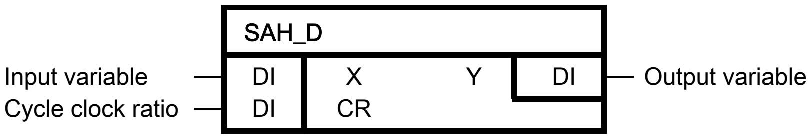
Project data

Can be loaded online	Yes
Special characteristics	-

6.16 SAH_D Sample & hold (DOUBLE INTEGER type)

□ SIMOTION ☑ SINAMICS

Symbol



Short description

Sample & hold block for the equidistant value transfer (DOUBLE INTEGER type) between blocks with different sampling times.

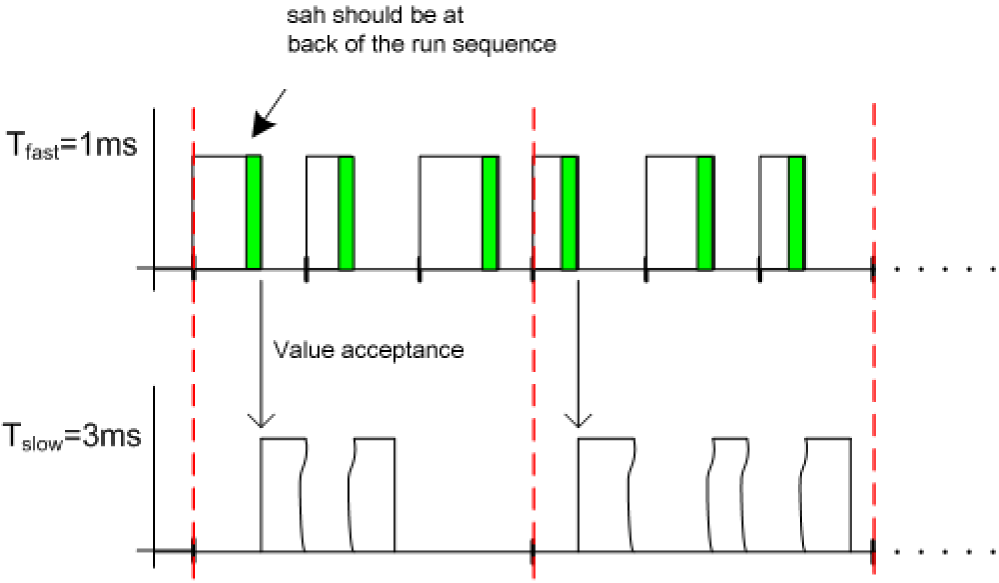
Operation mode

The value of the input variable X is taken over in the output variable Y in every CR cycle. The value transfer cycle is synchronized with the cycle control point of the execution system. The cycle control point defines the cycle clock in which all sampling times of the execution system are restarted.

A value transfer takes place every CR cycles relative to the cycle control point. The absolute value of CR is always generated for the cycle clock ratio. In the special case of CR = 0, the block behaves as for CR = 1. The block must always be configured in the faster sampling time. If the value from the slower scan time is transferred, it should be at the very beginning of the run sequence. If the value is to taken from the faster into the slower sampling time, the block should be computed last in the execution sequence.

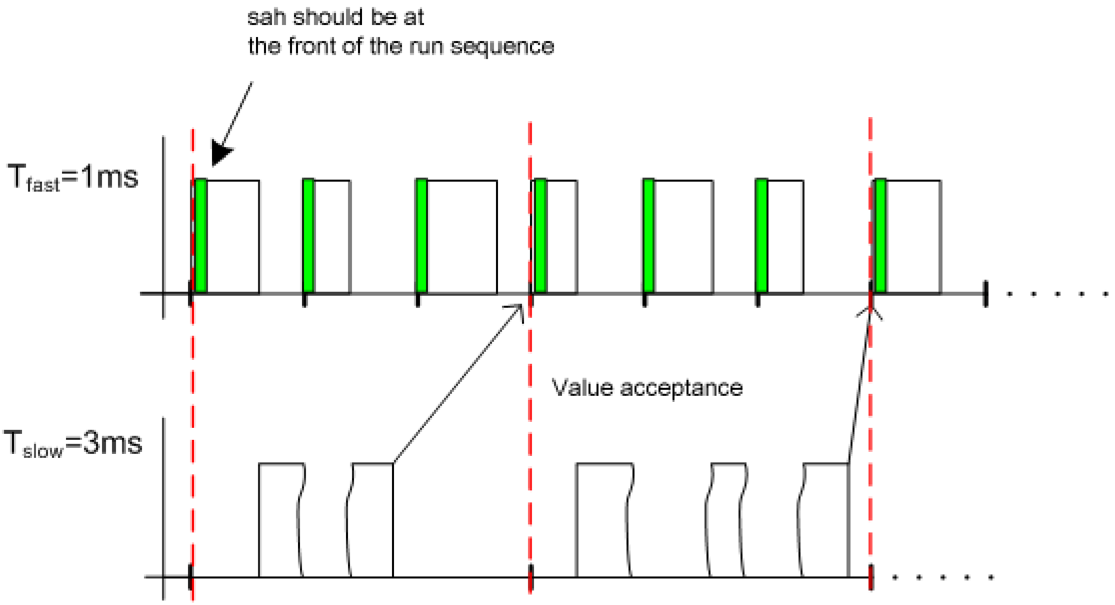
The following figure shows the transfer of values from a 1 ms level to a 3 ms level. The variation with time is shown for the calculation of the execution group.

$$CR = \frac{3\,ms}{1\,ms} = 3$$



The following figure shows the transfer of values from a 3 ms level to a 1 ms level. The variation with time is shown for the calculation of the execution group.

$$CR = \frac{3\,ms}{1\,ms} = 3$$



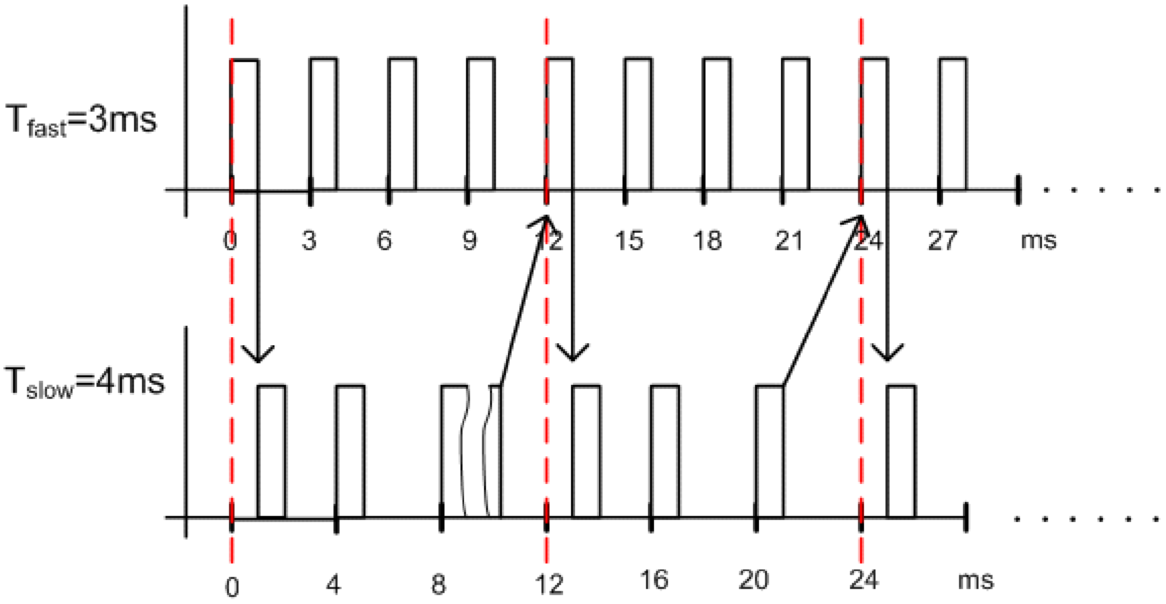
If the slower scan time is not a multiple of the faster scan time, the value can only be transferred consistently if both scanning procedures are restarted synchronously after CR cycles. This corresponds to the least common multiple of both scan times. CR is then calculated as follows:

$$CR = \frac{g(T_{fast}, T_{slow})}{T_{fast}}$$

$g(T_{fast}, T_{slow})$: least common multiple

The following shows the value transfer for T_{fast} = 3 ms and T_{slow} = 4 ms. The value transfer is made in both directions.

$$CR = \frac{g(3\,ms,\, 4\,ms)}{3\,ms} = \frac{12\,ms}{3\,ms} = 4$$



To enable values to be transferred at an optimal speed, it is, therefore, recommended that the slower scan time is always a multiple of the faster scan time.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	DINT	
CR	Cycle clock ratio	1	0 - (2^{31}-1)	
Y	Output variable	0	DINT	

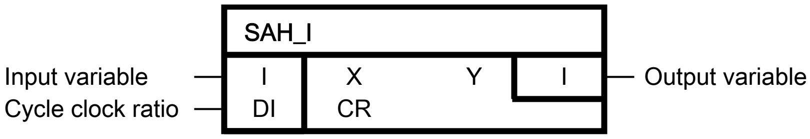
Project data

Can be loaded online	Yes
Special characteristics	-

6.17 SAH_I Sample & hold (INTEGER type)

□ SIMOTION ☑ SINAMICS

Symbol



Short description

Sample & hold block for the equidistant value transfer (INTEGER type) between blocks with different sampling times.

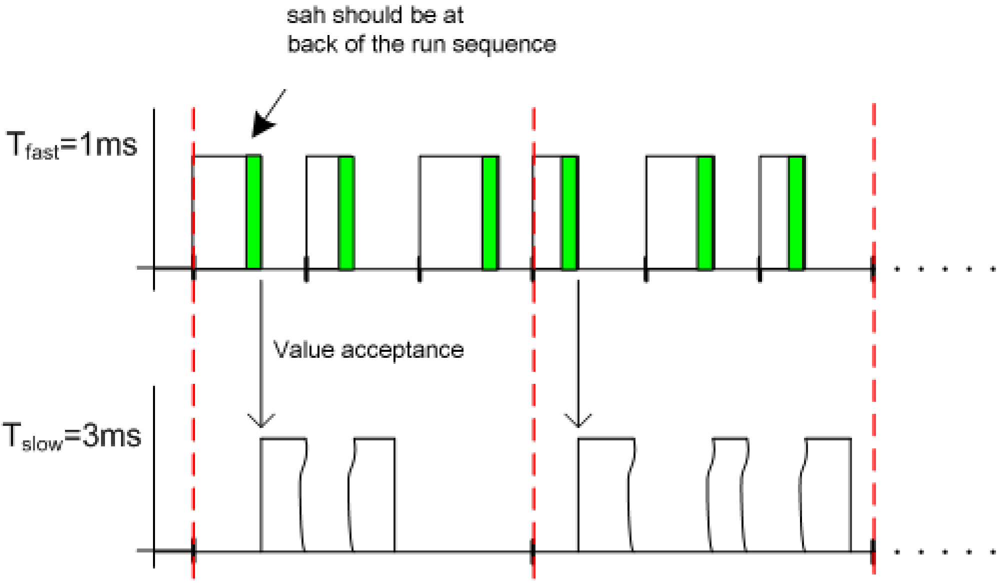
Operation mode

The value of the input variable X is taken over in the output variable Y in every CR cycle. The value transfer cycle is synchronized with the cycle control point of the execution system. The cycle control point defines the cycle clock in which all sampling times of the execution system are restarted.

A value transfer takes place every CR cycles relative to the cycle control point. The absolute value of CR is always generated for the cycle clock ratio. In the special case of CR = 0, the block behaves as for CR = 1. The block must always be configured in the faster sampling time. If the value from the slower scan time is transferred, it should be at the very beginning of the run sequence. If the value is to taken from the faster into the slower sampling time, the block should be computed last in the execution sequence.

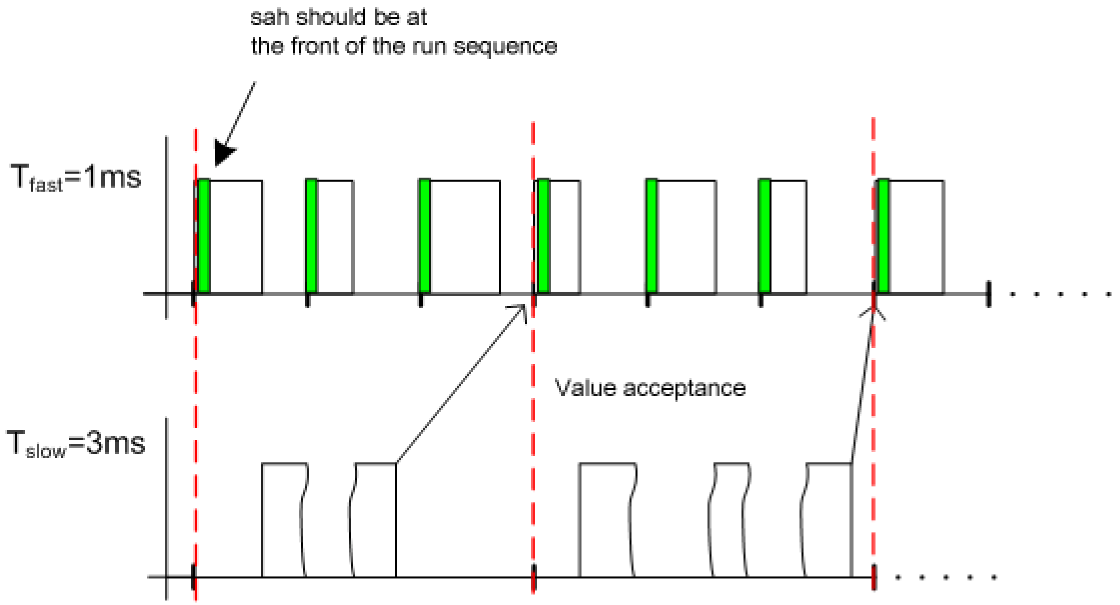
The following figure shows the transfer of values from a 1 ms level to a 3 ms level. The variation with time is shown for the calculation of the execution group.

$$CR = \frac{3\,ms}{1\,ms} = 3$$



The following figure shows the transfer of values from a 3 ms level to a 1 ms level. The variation with time is shown for the calculation of the execution group.

$$CR = \frac{3\,ms}{1\,ms} = 3$$



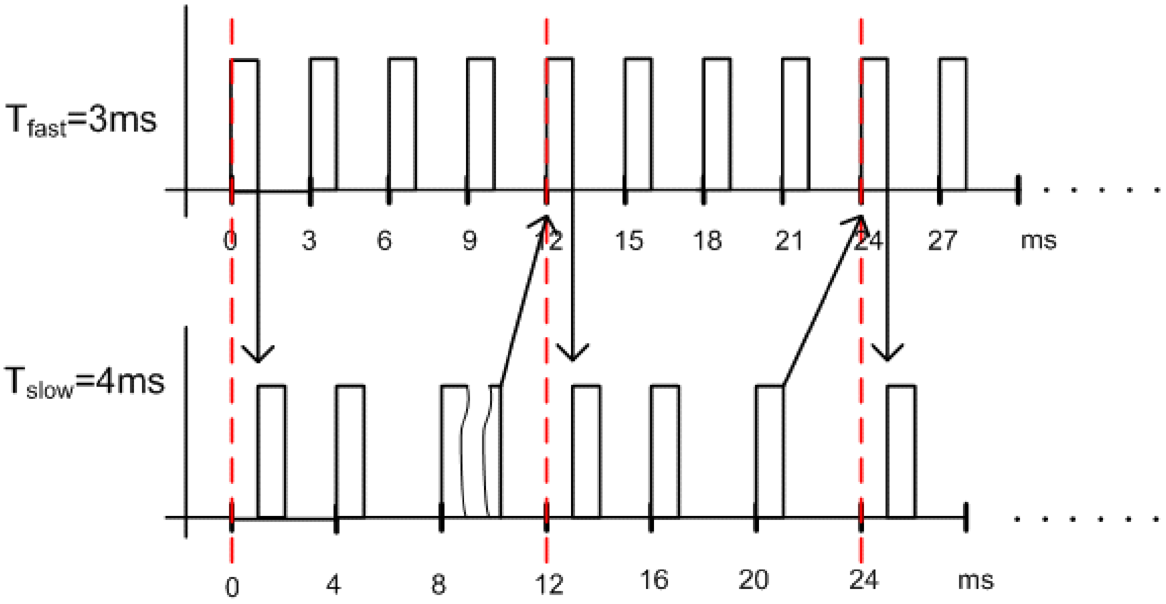
If the slower scan time is not a multiple of the faster scan time, the value can only be transferred consistently if both scanning procedures are restarted synchronously after CR cycles. This corresponds to the least common multiple of both scan times. CR is then calculated as follows:

$$CR = \frac{g(T_{fast}, T_{slow})}{T_{fast}}$$

$g(T_{fast}, T_{slow})$: least common multiple

The following shows the value transfer for T_{fast} = 3 ms and T_{slow} = 4 ms. The value transfer is made in both directions.

$$CR = \frac{g(3\,ms,\,4\,ms)}{3\,ms} = \frac{12\,ms}{3\,ms} = 4$$



To enable values to be transferred at an optimal speed, it is, therefore, recommended that the slower scan time is always a multiple of the faster scan time.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	INT	
CR	Cycle clock ratio	1	0 - (2^{31}-1)	
Y	Output variable	0	INT	

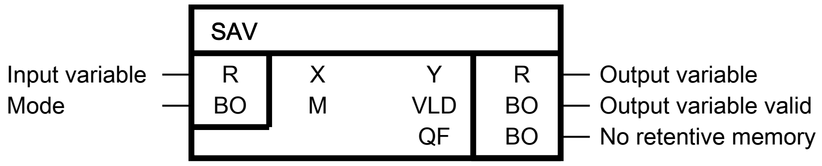
Project data

Can be loaded online	Yes
Special characteristics	-

6.18 SAV Value buffering (REAL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

SAV (Save) enables retentive storage of a REAL-type input variable.

Operation mode

The block is a retentive read/write memory for a REAL value.

The saved value of a SAV block is not retained when:

- The retentive memory has been cleared through a user action
- The chart on which the block was configured, has been deleted and the change transferred to the target device
- The block has been deleted and the change transferred to the target system
- The instance name has been changed and transferred to the target system

The value is retained:

- The instance name does not change during a download
- The target device ramps up without configuration data on the memory card. The memory of the missing SAV blocks is only released after a download. In this way, the data is also retained when the firmware is updated.
- When another SAV block has been added or removed
- When a download of the configuration is performed after an update of the DCBLIB
- When another DO has been added or removed and downloaded to the target device
- When another chart has been added or removed and downloaded to the target device
- When the target device ramps up with the same configuration as before the power failure

The block is only active when a 0 on output QF indicates that retentive memory space is available on the target device for storing the input values.

The block mode is set at input M:

Write mode (M =1)

- Input variable X is written cyclically to output Y.
- In addition, input variable X is transferred to the system for retentive storage. In so doing, an already saved value is overwritten.

Read mode (M = 0)

- The current saved value is output at output Y. The values at input X are not saved.
- Output VLD = 1 indicates the validity of Y. If the retentive memory of the system is recreated when the block is initialized, VLD = 0. In this case, Y is invalid and contains its default value. The status of VLD changes to 1 the first time a value is written (M = 1).

Initialization

The assignment between the SAV block and the value in the retentive memory is performed via the instance name of the block. The unique instance name is automatically generated by the DCC editor when the block is inserted in a chart. The instance name is made up of the call path of the block as follows:

(Chart name)/(Name of subchart 1)/(Name of subchart 2)/../(Name of the block)

An instance name could, for example, look like the following:

DCC_1/CFC1/CFC2/CFC3/SAV1

Chart name	DCC_1
Name of subchart 1	CFC1
Name of subchart 2	CFC2
Name of subchart 3	CFC3
Name of the block	SAV1

This instance name controls whether output Y is initialized with its default value or outputs the last saved value in the INIT mode. A check is made on the target device whether a retentive value has been saved for this instance name of the block. If this is not the case, the system recreates the memory space, the default value of output variable Y is transferred to the system for retentive storage and VLD is set to 0. If a retentive value has been saved for the instance name, it is read and written to output Y and status VLD = 1 is output.

If no retentive memory is available for the block, output QF = 1 is set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
M	Mode	0	0/1	
Y	Output variable	0.0	REAL	
VLD	Output variable valid	0	0/1	
QF	No retentive memory	0	0/1	

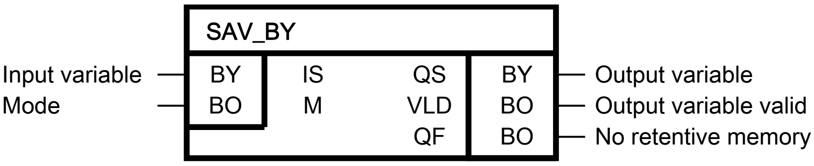
Project data

Can be inserted online	No
Special characteristics	A maximum of ten blocks can be used for the retentive storage (SAV, SAV_BY, SAV_I, SAV_D) for each SINAMICS or SINAMICS INTEGRATED drive unit. Retentive memory is available for a maximum of 40 bytes of user data. As of STARTER / SCOUT V4.2, the number of blocks is only checked in the consistency check (menu "Project -> Check consistency" or "Check consistency" in the context menu of the drive unit).

6.19 SAV_BY Value buffering (BYTE type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

SAV_BY (Save) enables retentive storage of a BYTE-type input variable.

Operation mode

The block is a retentive read/write memory for a BYTE value.

The saved value of a SAV block is not retained when:

- The retentive memory has been cleared through a user action
- The chart on which the block was configured, has been deleted and the change transferred to the target device
- The block has been deleted and the change transferred to the target system
- The instance name has been changed and transferred to the target system

The value is retained:

- The instance name does not change during a download
- The target device ramps up without configuration data on the memory card. The memory of the missing SAV blocks is only released after a download. In this way, the data is also retained when the firmware is updated.
- When another SAV block has been added or removed
- When a download of the configuration is performed after an update of the DCBLIB
- When another DO has been added or removed and downloaded to the target device
- When another chart has been added or removed and downloaded to the target device
- When the target device ramps up with the same configuration as before the power failure

The block is only active when a 0 on output QF indicates that retentive memory space is available on the target device for storing the input values.

The block mode is set at input M:

Write mode (M =1)

- Input variable IS is written cyclically to output QS.
- Input variable IS is also transferred to the system for retentive storage. In so doing, an already saved value is overwritten.

Read mode (M = 0)

- The currently saved value is output at output QS. The values at input IS are not saved
- Output VLD = 1 displays the validity of QS. If the retentive memory of the system is recreated when the block is initialized, VLD = 0. In this case, QS is invalid and contains its default value. The status of VLD changes to 1 the first time a value is written (M = 1).

Initialization

The assignment between the SAV block and the value in the retentive memory is performed via the instance name of the block. The unique instance name is automatically generated by the DCC editor when the block is inserted in a chart. The instance name is made up of the call path of the block as follows:

(Chart name)/(Name of subchart 1)/(Name of subchart 2)/../(Name of the block)

An instance name could, for example, look like the following:

DCC_1/CFC1/CFC2/CFC3/SAV1

Chart name	DCC_1
Name of subchart 1	CFC1
Name of subchart 2	CFC2
Name of subchart 3	CFC3
Name of the block	SAV1

This instance name controls whether output QS is initialized with its default value or outputs the last saved value in the INIT mode. A check is made on the target device whether a retentive value has been saved for this instance name of the block. If not, the memory space is recreated by the system, the default value of the output variable QS transferred to the system for retentive storage, and VLD = 0 set. If a retentive value has been saved for the instance name, this is read, written to output QS, and the status VLD = 1 output.

If no retentive memory is available for the block, output QF = 1 is set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
IS	Input variable	16#00	BYTE	
M	Mode	0	0/1	
QS	Output variable	16#00	BYTE	
VLD	Output variable valid	0	0/1	
QF	No retentive memory	0	0/1	

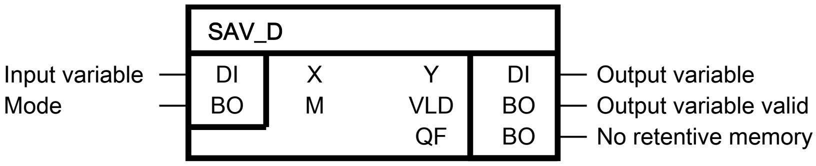
Project data

Can be inserted online	No
Special characteristics	A maximum of ten blocks can be used for the retentive storage (SAV, SAV_BY, SAV_I, SAV_D) for each SINAMICS or SINAMICS INTEGRATED drive unit. Retentive memory is available for a maximum of 40 bytes of user data. As of STARTER / SCOUT V4.2, the number of blocks is only checked in the consistency check (menu "Project -> Check consistency" or "Check consistency" in the context menu of the drive unit).

6.20 SAV_D Value buffering (DOUBLE INTEGER type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

SAV_D (Save) enables retentive storage of a DOUBLE INTEGER-type input variable.

Operation mode

The block is a retentive read/write memory for a DOUBLE INTEGER value.

The saved value of a SAV block is not retained when:

- The retentive memory has been cleared through a user action
- The chart on which the block was configured, has been deleted and the change transferred to the target device
- The block has been deleted and the change transferred to the target system
- The instance name has been changed and transferred to the target system

The value is retained:

- The instance name does not change during a download
- The target device ramps up without configuration data on the memory card. The memory of the missing SAV blocks is only released after a download. In this way, the data is also retained when the firmware is updated.
- When another SAV block has been added or removed
- When a download of the configuration is performed after an update of the DCBLIB
- When another DO has been added or removed and downloaded to the target device
- When another chart has been added or removed and downloaded to the target device
- When the target device ramps up with the same configuration as before the power failure

The block is only active when a 0 on output QF indicates that retentive memory space is available on the target device for storing the input values.

The block mode is set at input M:

Write mode (M =1)

- Input variable X is written cyclically to output Y.
- In addition, input variable X is transferred to the system for retentive storage. In so doing, an already saved value is overwritten.

Read mode (M = 0)

- The current saved value is output at output Y. The values at input X are not saved.
- Output VLD = 1 indicates the validity of Y. If the retentive memory of the system is recreated when the block is initialized, VLD = 0. In this case, Y is invalid and contains its default value. The status of VLD changes to 1 the first time a value is written (M = 1).

Initialization

The assignment between the SAV block and the value in the retentive memory is performed via the instance name of the block. The unique instance name is automatically generated by the DCC editor when the block is inserted in a chart. The instance name is made up of the call path of the block as follows:

(Chart name)/(Name of subchart 1)/(Name of subchart 2)/../(Name of the block)

An instance name could, for example, look like the following:

DCC_1/CFC1/CFC2/CFC3/SAV1

Chart name	DCC_1
Name of subchart 1	CFC1
Name of subchart 2	CFC2
Name of subchart 3	CFC3
Name of the block	SAV1

This instance name controls whether output Y is initialized with its default value or outputs the last saved value in the INIT mode. A check is made on the target device whether a retentive value has been saved for this instance name of the block. If this is not the case, the system recreates the memory space, the default value of output variable Y is transferred to the system for retentive storage and VLD is set to 0. If a retentive value has been saved for the instance name, it is read and written to output Y and status VLD = 1 is output.

If no retentive memory is available for the block, output QF = 1 is set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	DINT	
M	Mode	0	0/1	
Y	Output variable	0	DINT	
VLD	Output variable valid	0	0/1	
QF	No retentive memory	0	0/1	

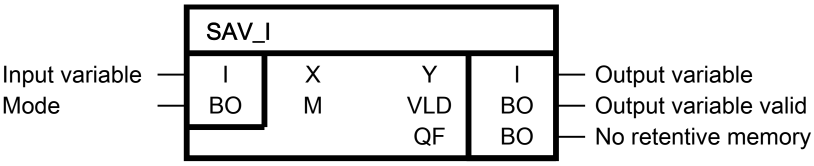
Project data

Can be inserted online	No
Special characteristics	A maximum of ten blocks can be used for the retentive storage (SAV, SAV_BY, SAV_I, SAV_D) for each SINAMICS or SINAMICS INTEGRATED drive unit. Retentive memory is available for a maximum of 40 bytes of user data. As of STARTER / SCOUT V4.2, the number of blocks is only checked in the consistency check (menu "Project -> Check consistency" or "Check consistency" in the context menu of the drive unit).

6.21 SAV_I Value buffering (INTEGER type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

SAV_I (Save) enables retentive storage of an INTEGER-type input variable.

Operation mode

The block is a retentive read/write memory for an INTEGER value.

The saved value of a SAV block is not retained when:

- The retentive memory has been cleared through a user action
- The chart on which the block was configured, has been deleted and the change transferred to the target device
- The block has been deleted and the change transferred to the target system
- The instance name has been changed and transferred to the target system

The value is retained:

- The instance name does not change during a download
- The target device ramps up without configuration data on the memory card. The memory of the missing SAV blocks is only released after a download. In this way, the data is also retained when the firmware is updated.
- When another SAV block has been added or removed
- When a download of the configuration is performed after an update of the DCBLIB
- When another DO has been added or removed and downloaded to the target device
- When another chart has been added or removed and downloaded to the target device
- When the target device ramps up with the same configuration as before the power failure

The block is only active when a 0 on output QF indicates that retentive memory space is available on the target device for storing the input values.

The block mode is set at input M:

Write mode (M =1)

- Input variable X is written cyclically to output Y.
- In addition, input variable X is transferred to the system for retentive storage. In so doing, an already saved value is overwritten.

Read mode (M = 0)

- The current saved value is output at output Y. The values at input X are not saved.
- Output VLD = 1 indicates the validity of Y. If the retentive memory of the system is recreated when the block is initialized, VLD = 0. In this case, Y is invalid and contains its default value. The status of VLD changes to 1 the first time a value is written (M = 1).

Initialization

The assignment between the SAV block and the value in the retentive memory is performed via the instance name of the block. The unique instance name is automatically generated by the DCC editor when the block is inserted in a chart. The instance name is made up of the call path of the block as follows:

(Chart name)/(Name of subchart 1)/(Name of subchart 2)/../(Name of the block)

An instance name could, for example, look like the following:

DCC_1/CFC1/CFC2/CFC3/SAV1

Chart name	DCC_1
Name of subchart 1	CFC1
Name of subchart 2	CFC2
Name of subchart 3	CFC3
Name of the block	SAV1

This instance name controls whether output Y is initialized with its default value or outputs the last saved value in the INIT mode. A check is made on the target device whether a retentive value has been saved for this instance name of the block. If this is not the case, the system recreates the memory space, the default value of output variable Y is transferred to the system for retentive storage and VLD is set to 0. If a retentive value has been saved for the instance name, it is read and written to output Y and status VLD = 1 is output.

If no retentive memory is available for the block, output QF = 1 is set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	INT	
M	Mode	0	0/1	
Y	Output variable	0	INT	
VLD	Output variable valid	0	0/1	
QF	No retentive memory	0	0/1	

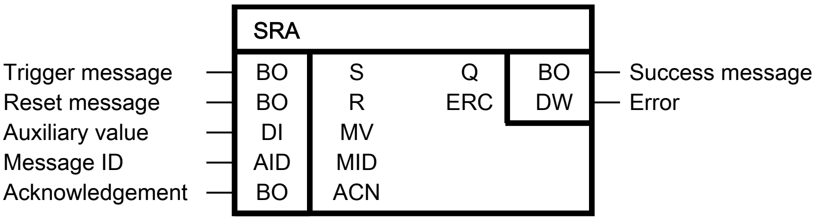
Project data

Can be inserted online	No
Special characteristics	A maximum of ten blocks can be used for the retentive storage (SAV, SAV_BY, SAV_I, SAV_D) for each SINAMICS or SINAMICS INTEGRATED drive unit. Retentive memory is available for a maximum of 40 bytes of user data. As of STARTER / SCOUT V4.2, the number of blocks is only checked in the consistency check (menu "Project -> Check consistency" or "Check consistency" in the context menu of the drive unit).

6.22 SRA Triggering/resetting of a message

☑ SIMOTION ☐ SINAMICS

Symbol



Short description

A message configured in SCOUT can be triggered or reset with the SRA (Set Alarm) block. The message is sent to the HMI and entered in the message buffer in the SIMOTION target device. The message buffer contains all active messages. The SRA block is configured for multiple instantiation. There can be several block instances that trigger the same message number.

Operation mode

- The message configured in SCOUT is entered at the MID message number, e.g. Message
- Input ACN =1 indicates that it is an acknowledgeable message. In this case, the message also only disappears after a reset when it has been acknowledged by the user on the HMI. ACN = 0 is configured for messages that cannot be acknowledged.
- A process value / auxiliary value must be entered at parameter MV if this has been specified in the configuration in SCOUT. A numerical value of the DINT type can be configured.The auxiliary value is inserted in the message text with a special syntax during the configuration of the messages: The call of a process value starts with @ and ends with @. The parameters in between specify the output of the value and the format.Only auxiliary values of the DINT type are possible for messages triggered with the SRA block.For detailed information on the syntax of auxiliary values in the message configuration, refer to the SIMOTION SCOUT online help.
- The message is triggered with a rising edge at input S. If the block is called with a new message number with 1 at input S, then the message is also triggered.Up to 40 messages can be entered in the message buffer. If the block SRA is activated with rising edge at input S at full message buffer, the block will be acknowledged with the error message.The message is not entered.
- The message is reset with a rising edge at input R.
- If a rising edge is set at both inputs S and R during a call, R takes priority, i.e. the message is reset.
- Block output Q =1 indicates that the message has been successfully set or reset.The outputs are set again with a rising edge at input S or R.
- With Q = 0, output ERC displays an error code that specifies the reason why the message could not be issued:
- The specified values can be displayed as OR logic operation between the constants.
- 16#0000 No error An entry is made in the message list for an incoming message. The entry is deleted from the message list for an outgoing message.
- 16#8001 Message name not permitted.
- 16#8002 Message loss through overflow There are already 40 entries in the message list. Entry has not been made in the message list.
- 16#8003 Message loss through overflow (signal not sent yet, signal overflow). Send buffer for notification of the clients is still occupied by the last event. Entry has not been made in the message list. Error may also occur when function calls with a rising and falling edge are made in quick succession
- 16#8004 Double message, message rejected (call with message came or went two times in succession). Entry has not been made in the message list.
- 16#8005 No display device signaled. Message is still entered in the list.
- 16#8007 A job has not been started yet with this message name (Initial call with S = FALSE.) Falling edge (outgoing message) came without a previous rising edge (incoming message).Entry has not been made in the message list.
- 16#8008 Message with this ID is already active in the message buffer.The message occurs when a message with the ID is in the message buffer and the same ID is set again. A new message buffer entry is not generated.
- 16#8009 Internal error
- 16#8010 Entry has been rejected; message acknowledgement memory is full.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
S	Trigger message	0	0/1	
R	Reset message	0	0/1	
MV	Auxiliary value	0	DINT	
MID	Message ID	STRUCTALARMID#NIL	StructAlarmId	
ACN	Acknowledgement	0	0/1	
Q	Success message	0	0/1	
ERC	Error	0	0- 0x80FF	

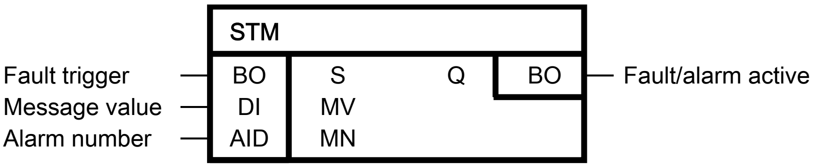
Project data

SIMOTION	4.3
SINAMICS	-
Can be inserted online	Yes
Special characteristics	-

6.23 STM Fault/alarm trigger

□ SIMOTION ☑ SINAMICS

Symbol



Short description

A predefined message (fault or warning) can be triggered on the DO with the STM (Set Message) block. The fault is displayed (e.g. STARTER, AOP) and entered in the fault buffer or warning buffer of the DO. The following applies for this block type:

- The message number (fault/warning number) assigned to an instance must be in the range 51050 to 51069 (preset value is 51050)
- A message number can be repeated on several instances in the DO (message can be set from different instances). However, for performance reasons, the SFM block is not configured for multiple instantiation The figure below shows the resulting behavior when there is multiple instantiation with the same message number for a fault at the same DO. Without an additional circuit, there is no coordination of the block instances with the same message number (this could not be implemented anyway when the instances were to run in different sampling times). For this reason, it is recommended that a unique message number be assigned for each instance in the DO.
- The message text is predefined and cannot be changed (see table below).
- The type of message cannot be changed (a fault cannot be redefined to an alarm, or vice versa)
- The default setting for the fault response is OFF2. This can be changed in the SINAMICS basic system parameter:
- p2100[0..19] "Setting the fault number for fault response" and
- 2101[0..19] "Setting the fault response

The default setting for the acknowledgement mode is IMMEDIATE. This can be changed in the SINAMICS basic system parameter:

- p2126[0..19] "Setting fault number for acknowledge mode" and
- 2127[0..19] "Sets acknowledgement mode

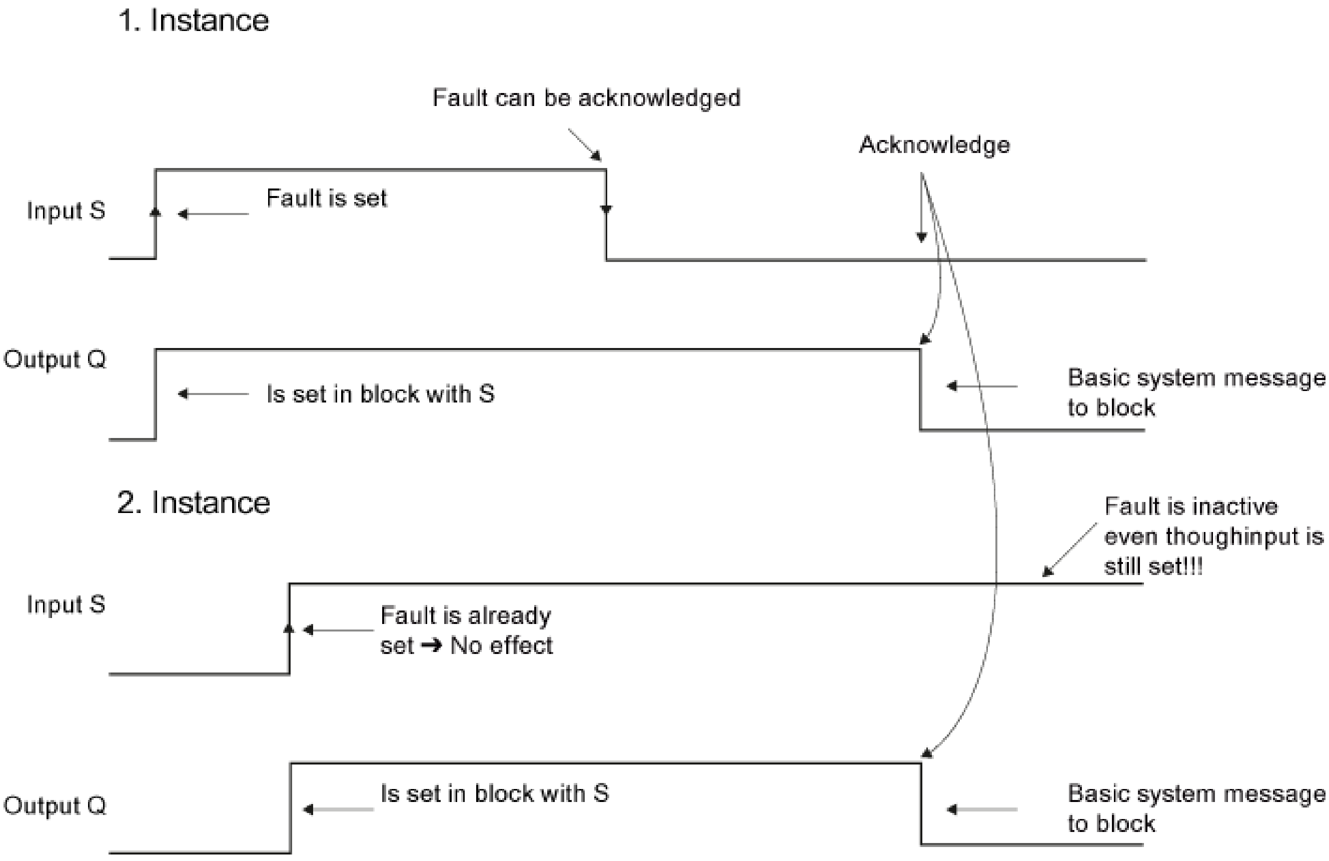
The table below specifies the default settings for the attributes. Possible options for the various settings can be found in the user documentation:

Message type	Alarm number	Response	Acknowledgement	Message text
Fault (cannot be changed)	F51050-F51059	OFF2 (can be changed via p2100/ p2101)	IMMEDIATE (can be changed via p2126/ p2127)	DCC: Fault F5105x additional value: %d(x:= 0 to 9)
Alarm (cannot be changed)	A51060-A51069			DCC: Warning A5106x additional value: %d(x:= 0 to 9)

Operation mode

The number of the fault to be triggered (F51050 - F51059) must be specified at input MN. A positive edge at input S triggers a fault at the DO. This is entered in the fault buffer of the DO and the specified response at the DO is executed. By doing this, output Q is set by the block. Output Q remains set as long as the fault is active. After a negative edge at input S, the fault can be acknowledged according to the acknowledgement attribute of the message (analog system faults: see first instance in figure below).

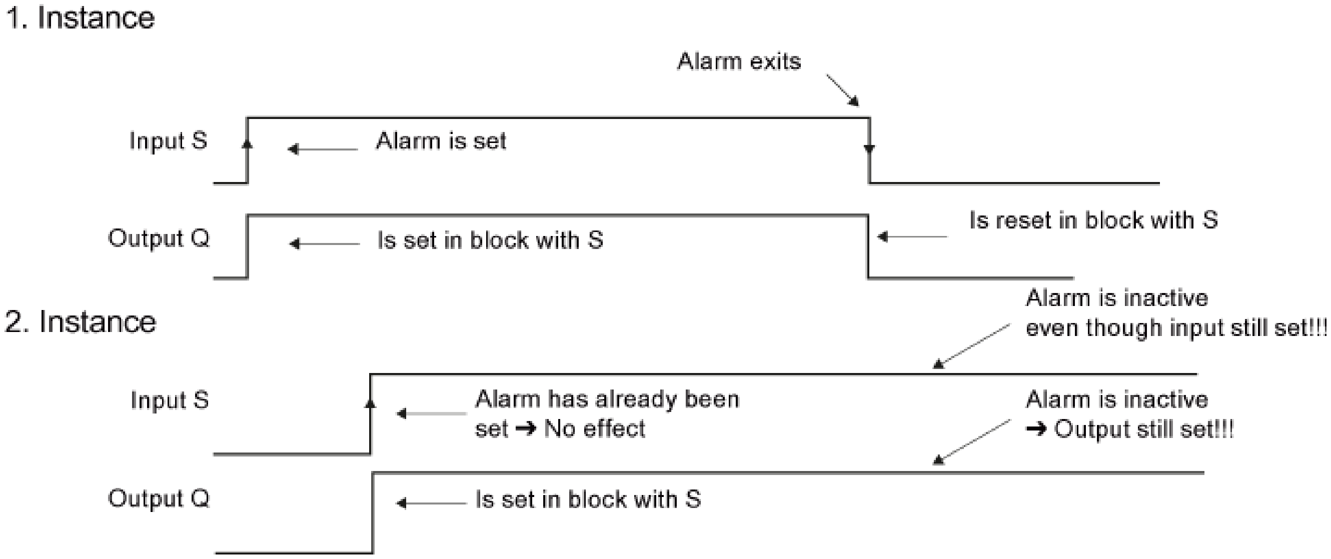
Input MV can be used to add additional information (fault value) for the fault. The value is transferred to input S when the fault is triggered on a positive edge and is entered in the fault buffer of the DO.


1. Instance
Fault can be acknowledged
Acknowledge
Input S
Fault is set
Output Q
Is set in block with S
Basic system message to block
2. Instance
Fault is inactive even thoughinput is still set!!!
Input S
Fault is already set ➔ No effect
Output Q
Is set in block with S
Basic system message to block

Example of two-fold instantiation with the same fault number at one DO (without additional RC circuitry)

Operation mode

The number of the warning to be triggered (A51060 -A51069) must be specified at input MN. A positive edge at input S triggers the warning assigned to the block. This is entered in the alarm buffer of the DO. In so doing, output Q is set. The output remains set as long as the alarm is active. Warnings are self-acknowledging and are acknowledged when input S is reset (see figure below). Additional information (warning value), which is also entered in the warning buffer, can be added to the warning via input MV.



Example of two-fold instantiation with the same alarm number at one DO (without additional RC circuitry)

Block connections

Block connec-tion	Description	Preassignment	Value range	Attributes
S	Fault trigger	0	0/1	
MV	Message value	0	DINT	
MN	Alarm number	F51050	F51050- F51059, A51060- A51069	
Q	Fault/alarm active	0	0/1	

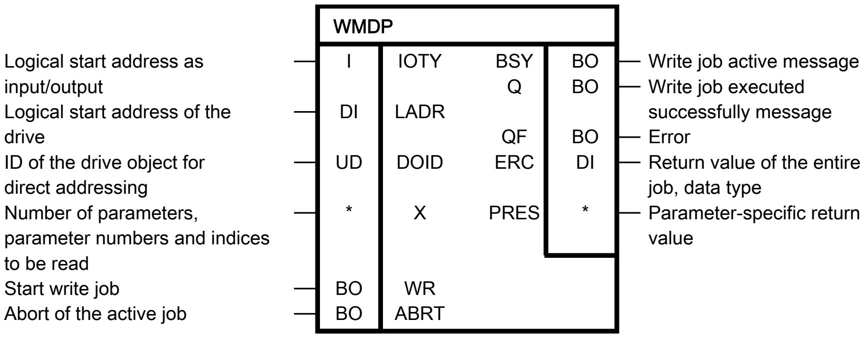
Project data

Can be inserted online	No
Special characteristics	-

6.24 WMDP Writing of drive parameters from the controller

☑ SIMOTION ☐ SINAMICS

Symbol


WMDP
Logical start address as input/output
I
IOTY
Logical start address of the drive
DI
LADR
ID of the drive object for direct addressing
UD
DOID
Number of parameters, parameter numbers and indices to be read
*
X
Start write job
BO
WR
Abort of the active job
BO
ABRT
BSY
BO
Write job active message
Q
BO
Write job executed successfully message
QF
BO
Error
ERC
DI
Return value of the entire job, data type
PRES
*
Parameter-specific return value

Short description

The WMDP block allows the writing of up to 23 SINAMICS parameters from the DCC SIMOTION program.Only SINAMICS drives are supported.For error diagnostics, the error code ERC can be evaluated. ERC corresponds to the error code for parameter access according to PROVIdrive DPV1.The possible error codes can be found in Appendix A.2 of this document or in the SIMOTION Communication System Manual in Section PROFIdrive and there in the Subsection Acyclic communication (Base Mode Parameter Access) → Error evaluation in table Error codes in Base Mode Parameter Access responses.

The WMDP block is available as of SIMOTION V4.2.

Operation mode

First the block inputs for addressing the drive as well as the selection of the parameters to be written and the values to be written are entered. If a parameter is not indexed, IDX = 0 must be set. The asynchronous write job is started with the positive edge at input WR. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and communication load and can vary from job to job. During an active write job, any additional positive edges at input WR are ignored.The actual reading/writing of the parameters is not performed in the DCC task. The block instance only controls the communication command. The results of the read-write job must be polled at the block outputs in the following task cycles. The evaluation is performed via global variables or user-defined block types.The output Q = 1 indicates that the parameters have been written successfully.If an error occurs during an access, this is signaled with QF = 1.For error diagnostics, the error code ERC can be evaluated.

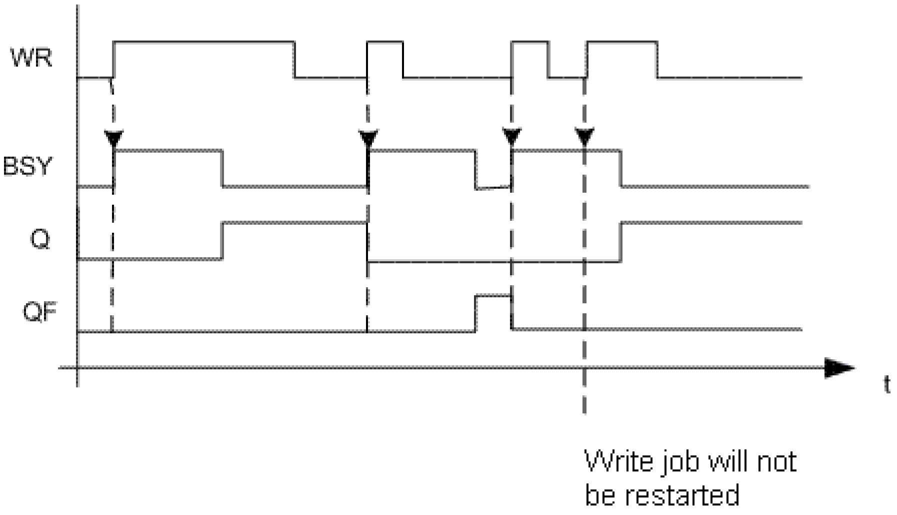
Despite the overall status Q =1, the writing of individual parameters may have been aborted with error. Write jobs for which there is no error in PRES, have been executed.

The error status of the individual write jobs can be evaluated on the parameter-specific return value PRES.An active job is aborted with the positive edge on the ABRT input. The ABRT signal must have the value 1 for at least one cycle.

Short description

The block enables up to 23 SINAMICS parameters to be written.

Time diagram



The following data sets can be used to write the parameters:With PROFIBUS (external or integrated), data set 47 is always read out irrespective of whether the function is called with a valid (0>=doId<=254) or invalid 'doId' (doId=255).With PROFINET, two data sets are available:− Base Mode Parameter Access - local (data set 0xB02E)This data set (DS) is used for SIMOTION if no or an invalid 'doId' (doId=255) is specified in the function. Access to the appropriate DO is then performed via the Parameter Access Point (PAP).- The address of the PAP can be specified directly or the log. address of the cyclic data specified (e.g. 256 for a DO axis). SIMOTION then determines the associated program flowchart from this address and then accesses the correct address.Program flowchart must always be at subslot 1 (configuration in HW Config).− Base Mode Parameter Access - global (data set 0xB02F)This data set is used if a valid 'doId' (0>=doId<=254) is entered. Any valid program flowchart or address can be specified because the assignment is only performed via the 'doId'.

Description of the block inputs

'IOTY': Input/output assignment of the logical start address of the drive.With 198: INPUT, the logical address of the drive is in the input range.With 199: OUTPUT, the logical address of the drive is in the output range.Diagnostics addresses are always of the INPUT type.

'LADR': Specification of the logical start address of the drive. If the optional parameter DOID is also used, any arbitrary address of the station (preferably the diagnostics address of the station) can be specified.With PROFINET, parameter access is via the Parameter Access Point (PAP) of a drive object.As an alternative to the logical start address of the drive, specification of the diagnostics address of the associated PAP is recommended.

'DOID': For the direct addressing of a drive object. Under the following conditions, the DOID may not be specified, or specified with an invalid value (>254):- Access via the DOID is not supported by the DP slave / I/O device (P978 not implemented).- Data set 0xB02F is not supported (PROFINET only).- Access is to be performed via the Parameter Access Point of a DO (PROFINET only).

'X.NUMP': Number of parameters to be written.

'X.PAR.NUM': Specification of the parameter numbers from which the values are to be written.

'X.PAR.IDX': Parameter index; for indexed values, 0 means index 0. For non-indexed values, parameter index 0 must be specified.'X.PAR.DTYP': Specifies the data type of the parameter (for the coding, see PROFIdrive profile). The data type must match the type of the parameter in the drive. The block performs the data type-specific transfer.If the specified data type does not match the actual data type of the parameter in SINAMICS, an error status is returned.'X.PAR.X': Data to be written to the drive, DWORD. Conversion blocks are required for different data types. Conversion block R_DW should be used to write a REAL parameter. Conversion block B_DW should be used to write a BYTE parameter.

'WR': Start write job.

'ABRT': Abort active job.

Description of the block outputs

'Q': Job completed without errors.

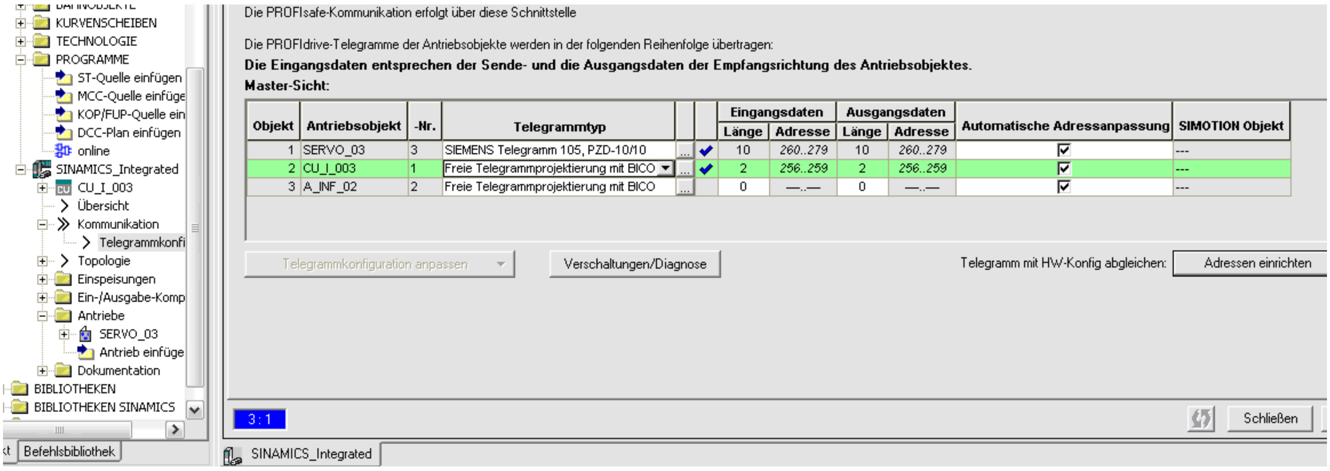
'QF': Job completed with errors.

'ERC': Corresponds to the values of the return value 'functionResult' of the _writeDriveMultiParameter function.

Parameterization example

In order to be able to write certain parameters of a drive object (in the example: SERVO_03), proceed as follows:

First set the correct message frame configuration.



Die PROFIsafe-Kommunikation erfolgt über diese Schnittstelle

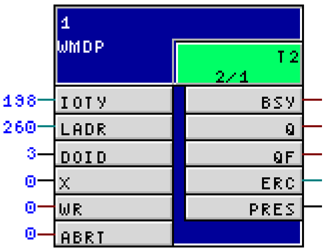
Die PROFIdrive-Telegramme der Antriebsobjekte werden in der folgenden Reihenfolge übertragen:

Die Eingangsdaten entsprechen der Sende- und die Ausgangsdaten der Empfangsrichtung des Antriebsobjektes.

Master-Sicht:

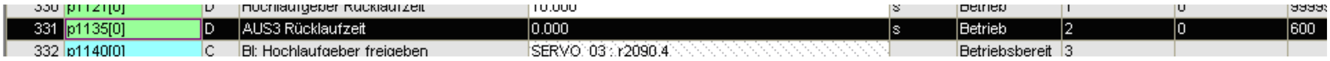
Objekt	Antriebsobjekt	-Nr.	Telegrammtyp			Eingangsdaten		Ausgangsdaten		Automatische Adressanpassung	SIMOTION Objekt
						Länge	Adresse	Länge	Adresse		
1	SERVO_03	3	SIEMENS Telegramm 105, PZD-10/10	...	✓	10	260..279	10	260..279	☑	---
2	CU_I_003	1	Freie Telegrammprojektierung mit BICO	...	✓	2	256..259	2	256..259	☑	---
3	A_INF_02	2	Freie Telegrammprojektierung mit BICO	...		0	—.—	0	—.—	☑	---

Then set the desired DO address in the WMDP block. To do this, set the block input 'LADR' to the address (260) set in the message frame and the block input 'DOID' to the number (3) set in the message frame.



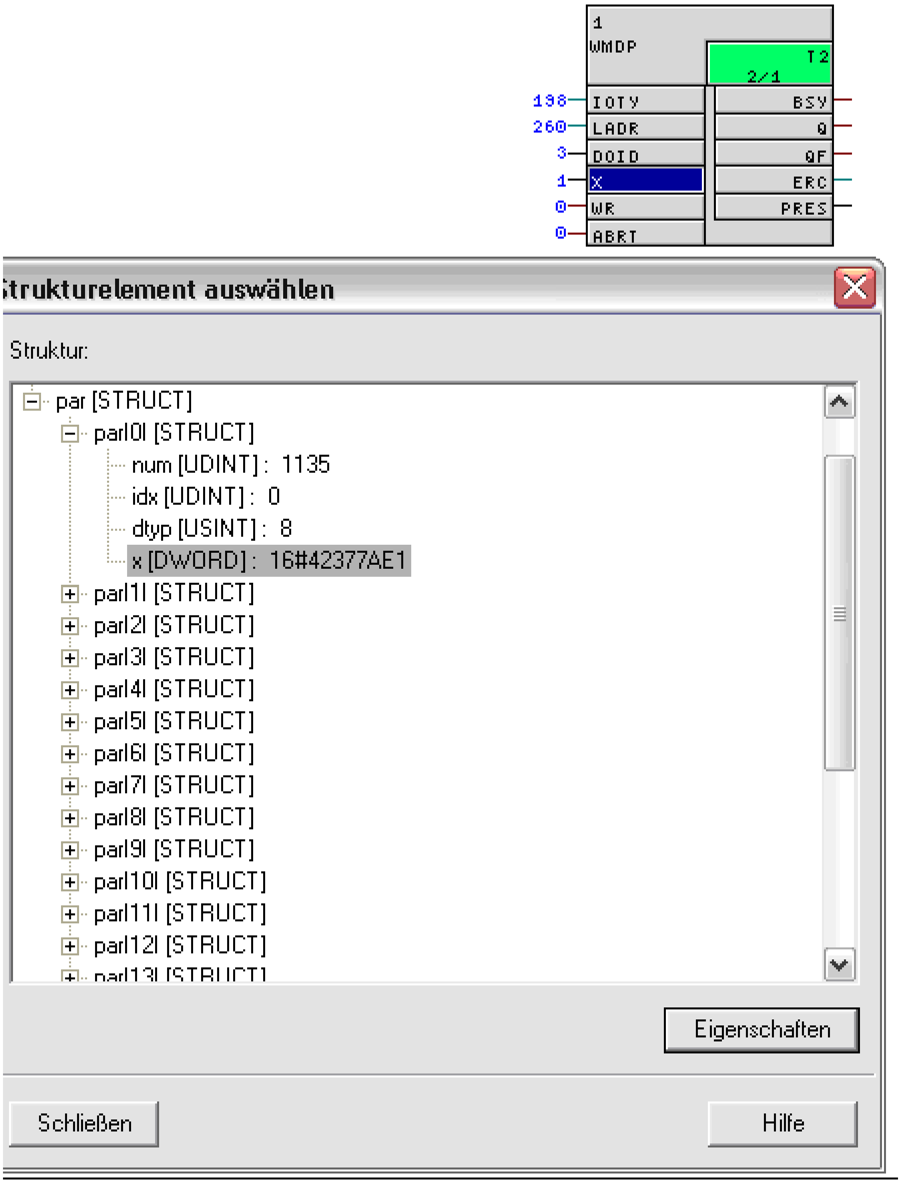
138 IOTY
260 LADR
3 DOID
0 X
0 WR
0 ABRT

Select the parameters to be written at block input 'X', e.g. P1135(0) OFF3 ramp-down time.



330	p1121[0]	D	Hochlaufgeber Rücklaufzeit	10.000	s	Betrieb	1	0	9999
331	p1135[0]	D	AUS3 Rücklaufzeit	0.000	s	Betrieb	2	0	600
332	p1140[0]	C	BI: Hochlaufgeber freigeben	SERVO_03 : r2090.4		Betriebsbereit	3		

To do this, double-click block input 'X', select the first structure element and enter the parameter number (1135) at 'num' and the index (0) at 'idx'. Enter the data type at ’dtyp’. It must correspond to the data type of the parameter, in our case 8 (floating-point). Enter the value 16#42377AE1 as DWORD at 'x'. The coding of the data types can be found in the SIMOTION List Manual 'System Functions/Variables Devices → System Functions - Devices 1 → _readDriveMultiParameterDescription'.



1
WMDP
T2
2/1

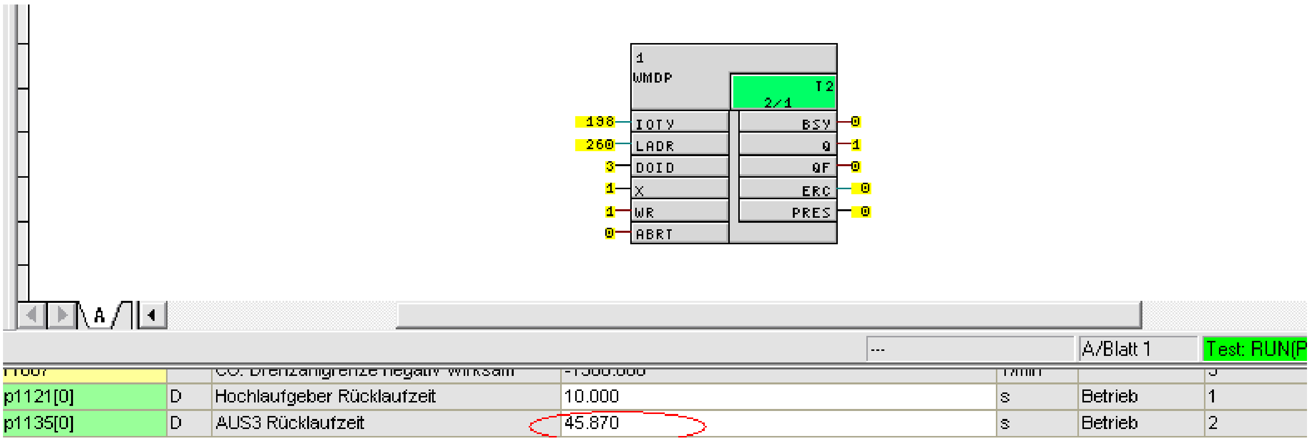
```
198—IOTY   BSY—
260—LADR     Q—
  3—DOID    QF—
  1—X      ERC—
  0—WR    PRES—
  0—ABRT
```

itrukturelement auswählen

Struktur:

```
⊟ par [STRUCT]
   ⊟ par[0] [STRUCT]
        num [UDINT] :  1135
        idx [UDINT] :  0
        dtyp [USINT] :  8
        x [DWORD] :  16#42377AE1
   ⊞ par[1] [STRUCT]
   ⊞ par[2] [STRUCT]
   ⊞ par[3] [STRUCT]
   ⊞ par[4] [STRUCT]
   ⊞ par[5] [STRUCT]
   ⊞ par[6] [STRUCT]
   ⊞ par[7] [STRUCT]
   ⊞ par[8] [STRUCT]
   ⊞ par[9] [STRUCT]
   ⊞ par[10] [STRUCT]
   ⊞ par[11] [STRUCT]
   ⊞ par[12] [STRUCT]
   ⊞ par[13] [STRUCT]
```

Then set the block input 'WR' to 1 to start the writing. The result can be viewed in the expert list.



p1121[0]	D	Hochlaufgeber Rücklaufzeit	10.000	s	Betrieb	1
p1135[0]	D	AUS3 Rücklaufzeit	45.870	s	Betrieb	2

Block connections

Block connection	Description	Preassignment	Value range	Attributes
IOTY	Logical start address as input/output	0	0: Not valid 198: Inputaddress 199: Outputaddress	
LADR	Logical start address of the drive	-1	DINT	
DOID	ID of the drive object for direct addressing	255	0 .. 254, 255: Not valid	
X	Number of parameters, parameter numbers and indices to be read			
X.NUMP	Number of parameters to be written	1	1 ..23	
X.PAR	Description of a parameter			
X.PAR.NUM	Number of the parameter	0	1 ..65535	
X.PAR.IDX	Index of the parameter	0	1 ..65535	
X.PAR.DTYP	Data type of the parameter to be written	0	USINT	
X.PAR.X	Value of the parameter	0	DWORD	
WR	Start write job	0	0/1	
ABRT	Abort of the active job	0	0/1	
BSY	Write job active message	0	0/1	
Q	Write job executed successfully message	0	0/1	
QF	Error	0	0/1	
ERC	Return value of the entire job, data type	16#0000	DWORD	
PRES	Parameter-specific return value	0	DWORD	

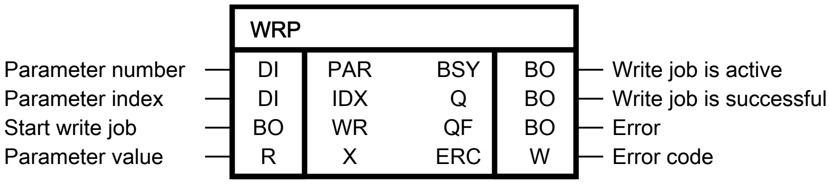
Project data

SIMOTION	as of V4.2
SINAMICS	-
Can be loaded online	No
Process context	Cyclic, equidistant
Special characteristics	-

6.25 WRP Writing drive parameters (REAL type)

□ SIMOTION ☑ SINAMICS

Symbol



Short description

The block enables the asynchronous writing of drive parameters of the REAL type on the local drive object.

Operation mode

The parameter number and the index of the parameter that is to be written are indicated at inputs PAR and IDX, respectively. If a parameter is not indexed, IDX = 0 must be set. The parameter is always written on the drive object on which the chart with the block is computed. It is not possible to access parameters on several drive objects.

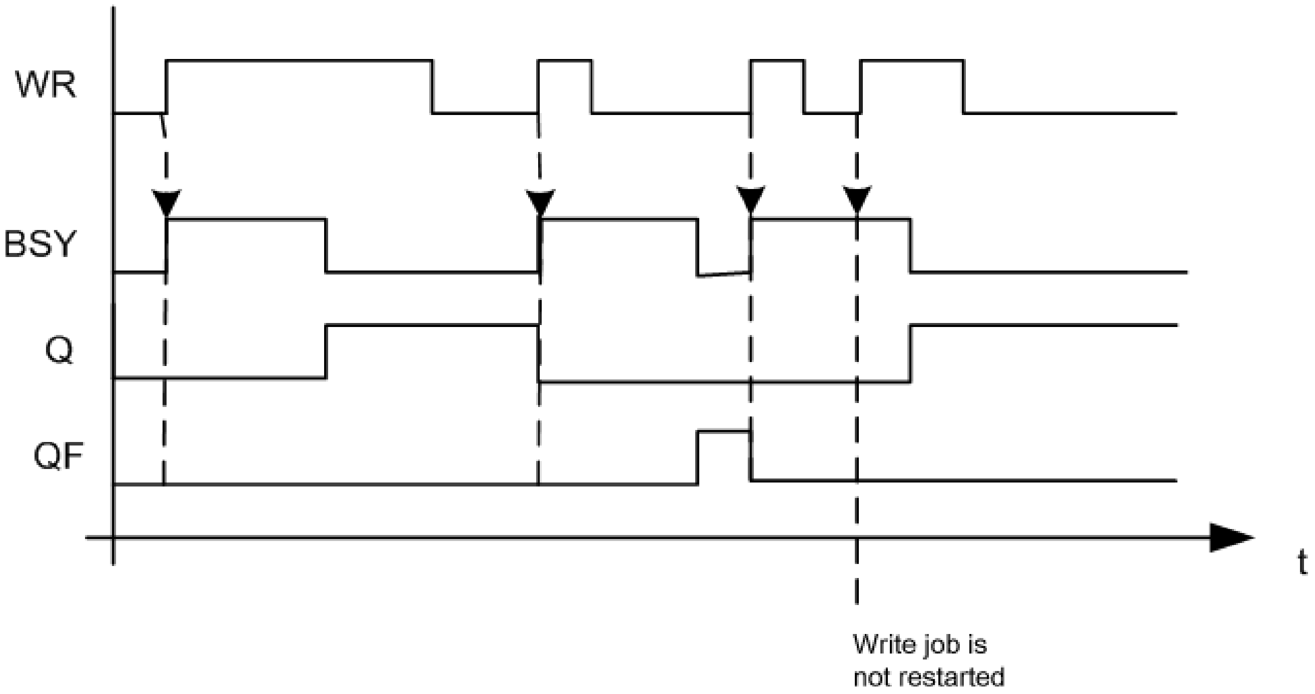
The parameter value is specified via input X. The asynchronous write job can be started on a positive edge at input WR. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and can vary from job to job. During an active write job, any additional positive edges at input WR are ignored.

Output Q = 1 indicates that the parameter has been written successfully. If an error occurs during the access, this is signaled with QF = 1.

For an error diagnostic, the error code ERC can be evaluated. ERC corresponds to the error code for parameter access according to PROFIdrive DPV1. The possible error codes can be found in Appendix A.2 of this document or in the SINAMICS Function Manual FH1 in Section PROFIBUS DP / PROFINET IO Communication and there in the Subsection Communication according to PROFIdrive → Acyclic communication → Configuration of the jobs and responses in Table Error values in DPV1 parameter responses.

ERC is only valid as long as QF = 1.

Time diagram



Quantity framework

Any number of asynchronous jobs of different block instances can be issued in parallel. Each block instance can only process one job.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
PAR	Parameter number	0	$0..2^{16}$	
IDX	Parameter index	0	$0..2^{16}$	
WR	Start write job	0	0/1	
X	Parameter value	0.0	REAL	
BSY	Write job is active	0	0/1	
Q	Write job is successful	0	0/1	
QF	Error	0	0/1	
ERC	Error code	16#0000	WORD	

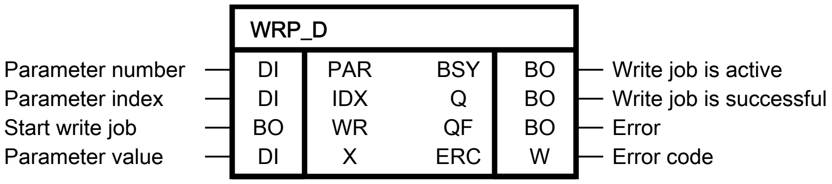
Project data

Can be loaded online	No
Special characteristics	-

6.26 WRP_D Writing drive parameters (DOUBLE INTEGER type)

□ SIMOTION ☑ SINAMICS

Symbol



Short description

The block enables the asynchronous writing of drive parameters of the DOUBLE INTEGER type on the local drive object.

Operation mode

The parameter number and the index of the parameter that is to be written are indicated at inputs PAR and IDX, respectively. If a parameter is not indexed, IDX = 0 must be set. The parameter is always written on the drive object on which the chart with the block is computed. It is not possible to access parameters on several drive objects.

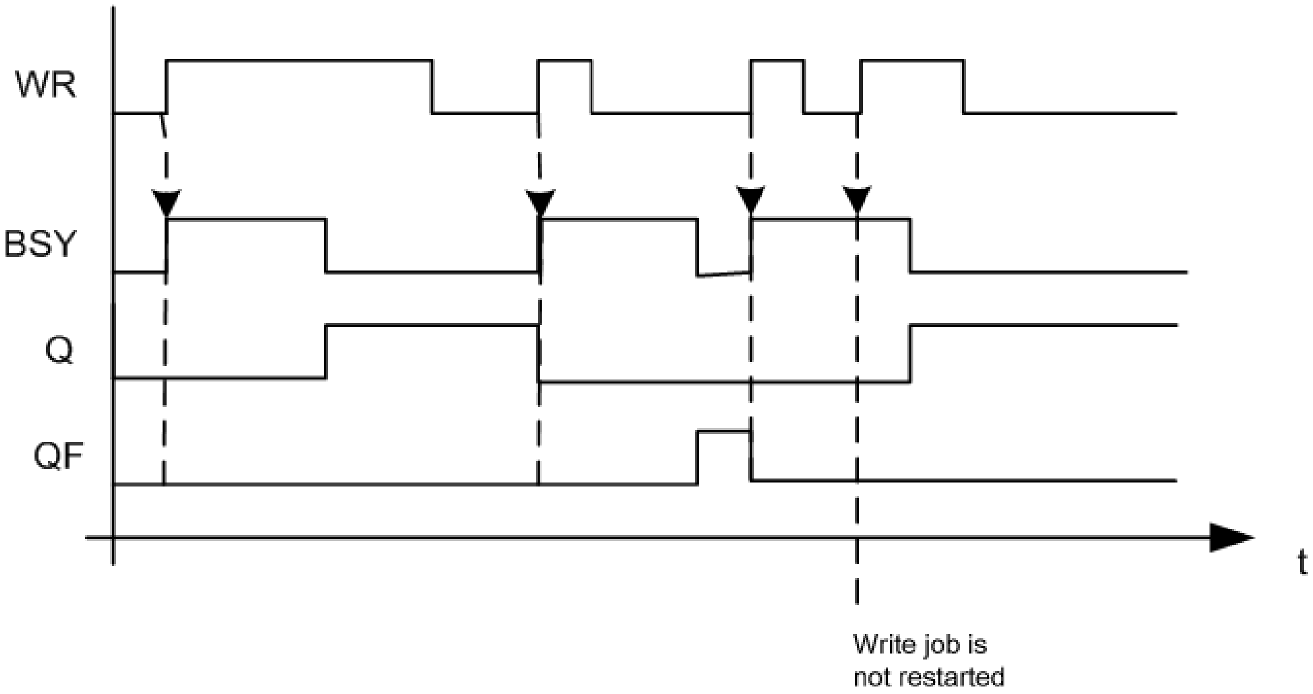
The parameter value is specified via input X. The asynchronous write job can be started on a positive edge at input WR. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and can vary from job to job. During an active write job, any additional positive edges at input WR are ignored.

Output Q = 1 indicates that the parameter has been written successfully. If an error occurs during the access, this is signaled with QF = 1.

For an error diagnostic, the error code ERC can be evaluated. ERC corresponds to the error code for parameter access according to PROFIdrive DPV1. The possible error codes can be found in Appendix A.2 of this document or in the SINAMICS Function Manual FH1 in Section PROFIBUS DP / PROFINET IO Communication and there in the Subsection Communication according to PROFIdrive → Acyclic communication → Configuration of the jobs and responses in Table Error values in DPV1 parameter responses.

ERC is only valid as long as QF = 1.

Time diagram



Quantity framework

Any number of asynchronous jobs of different block instances can be issued in parallel. Each block instance can only process one job.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
PAR	Parameter number	0	$0..2^{16}$	
IDX	Parameter index	0	$0..2^{16}$	
WR	Start write job	0	0/1	
X	Parameter value	0	DINT	
BSY	Write job is active	0	0/1	
Q	Write job is successful	0	0/1	
QF	Error	0	0/1	
ERC	Error code	16#0000	WORD	

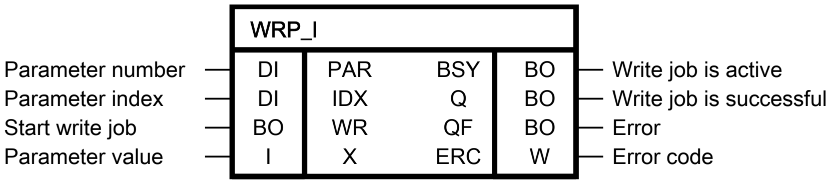
Project data

Can be loaded online	No
Special characteristics	-

6.27 WRP_I Writing drive parameters (INTEGER type)

□ SIMOTION ☑ SINAMICS

Symbol



Short description

The block allows asynchronous writing of drive parameters of the integer type on the local drive object

Operation mode

The parameter number and the index of the parameter that is to be written are indicated at inputs PAR and IDX, respectively. If a parameter is not indexed, IDX = 0 must be set. The parameter is always written on the drive object on which the chart with the block is computed. It is not possible to access parameters on several drive objects.

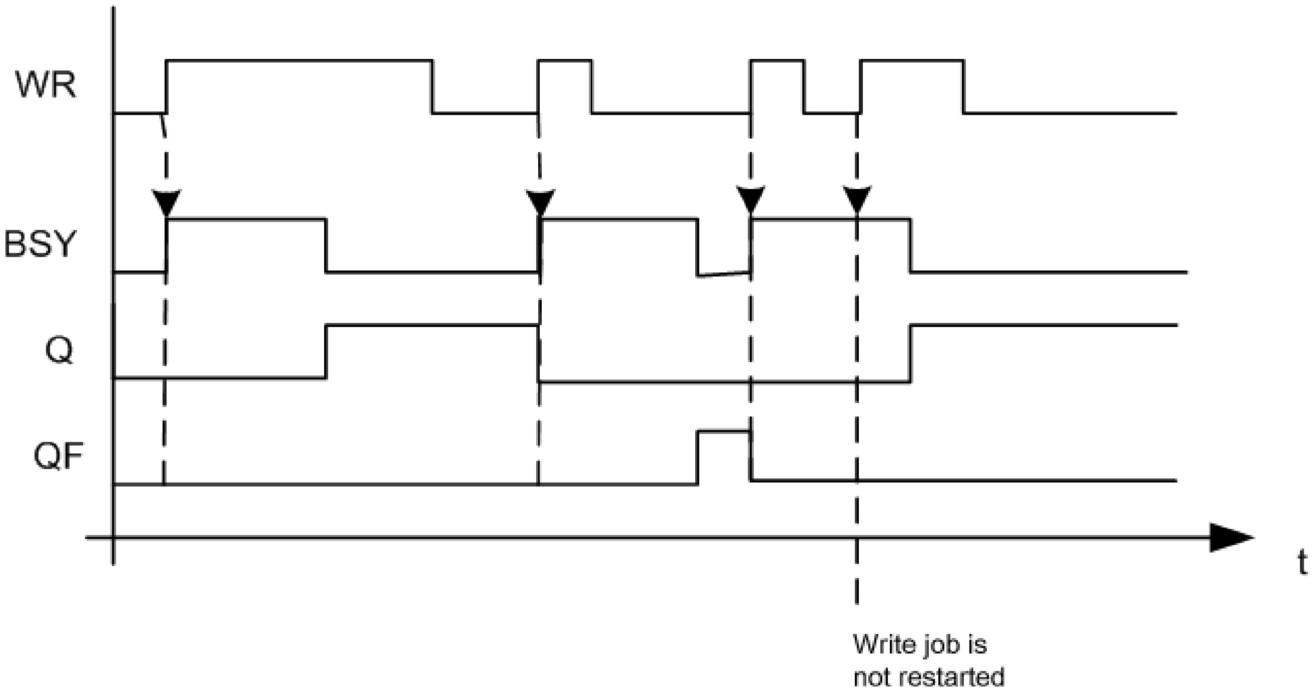
The parameter value is specified via input X. The asynchronous write job can be started on a positive edge at input WR. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and can vary from job to job. During an active write job, any additional positive edges at input WR are ignored.

Output Q = 1 indicates that the parameter has been written successfully. If an error occurs during the access, this is signaled with QF = 1.

For an error diagnostic, the error code ERC can be evaluated. ERC corresponds to the error code for parameter access according to PROFIdrive DPV1. The possible error codes can be found in Appendix A.2 of this document or in the SINAMICS Function Manual FH1 in Section PROFIBUS DP / PROFINET IO Communication and there in the Subsection Communication according to PROFIdrive → Acyclic communication → Configuration of the jobs and responses in Table Error values in DPV1 parameter responses.

ERC is only valid as long as QF = 1.

Time diagram



Quantity framework

Any number of asynchronous jobs of different block instances can be issued in parallel. Each block instance can only process one job.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
PAR	Parameter number	0	$0..2^{16}$	
IDX	Parameter index	0	$0..2^{16}$	
WR	Start write job	0	0/1	
X	Parameter value	0	INT	
BSY	Write job is active	0	0/1	
Q	Write job is successful	0	0/1	
QF	Error	0	0/1	
ERC	Error code	16#0000	WORD	

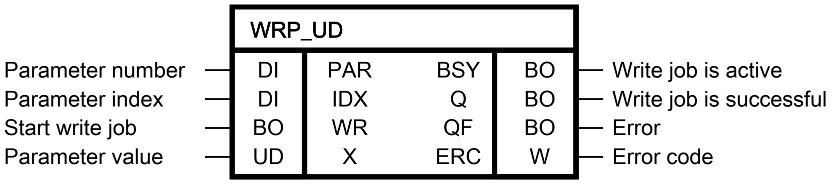
Project data

Can be loaded online	No
Special characteristics	-

6.28 WRP_UD Writing drive parameters (UNSIGNED DOUBLE INTEGER type)

□ SIMOTION ☑ SINAMICS

Symbol



Short description

WRP_UD (Write Parameter) enables the asynchronous writing of drive parameters of the UNSIGNED DOUBLE INTEGER type on the local drive object.

Operation mode

The parameter number and the index of the parameter that is to be written are indicated at inputs PAR and IDX, respectively. If a parameter is not indexed, IDX = 0 must be set. The parameter is always written on the drive object on which the chart with the block is computed. It is not possible to access parameters on several drive objects.

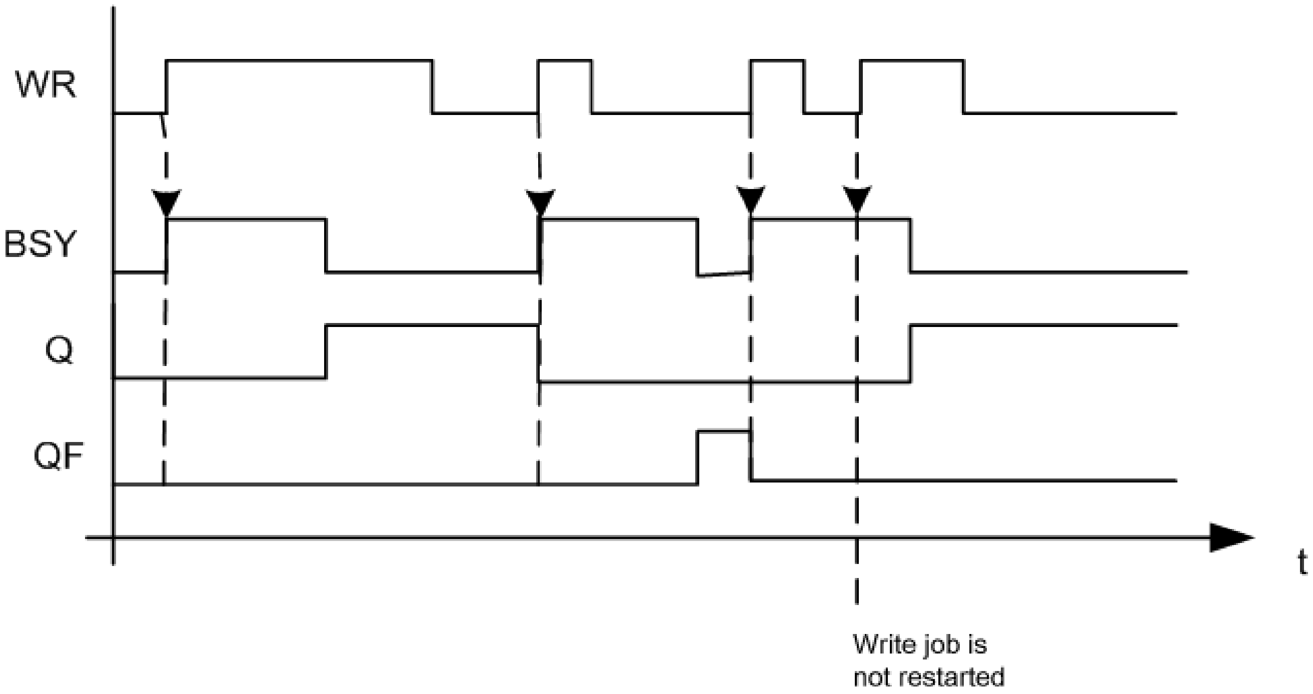
The parameter value is specified via input X. The asynchronous write job can be started on a positive edge at input WR. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and can vary from job to job. During an active write job, any additional positive edges at input WR are ignored.

Output Q = 1 indicates that the parameter has been written successfully. If an error occurs during the access, this is signaled with QF = 1.

For an error diagnostic, the error code ERC can be evaluated. ERC corresponds to the error code for parameter access according to PROFIdrive DPV1. The possible error codes can be found in Appendix A.2 of this document or in the SINAMICS Function Manual FH1 in Section PROFIBUS DP / PROFINET IO Communication and there in the Subsection Communication according to PROFIdrive → Acyclic communication → Configuration of the jobs and responses in Table Error values in DPV1 parameter responses.

ERC is only valid as long as QF = 1.

Time diagram



Quantity framework

Any number of asynchronous jobs of different block instances can be issued in parallel. Each block instance can only process one job.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
PAR	Parameter number	0	0..2^{16}	
IDX	Parameter index	0	0..2^{16}	
WR	Start write job	0	0/1	
X	Parameter value	0	UDINT	
BSY	Write job is active	0	0/1	
Q	Write job is successful	0	0/1	
QF	Error	0	0/1	
ERC	Error code	16#0000	WORD	

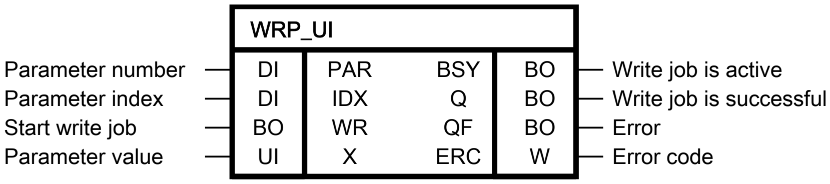
Project data

Can be loaded online	No
Special characteristics	-

6.29 WRP_UI Writing drive parameters (UNSIGNED INTEGER type)

□ SIMOTION ☑ SINAMICS

Symbol



Short description

WRP_UI (Write Parameter) enables the asynchronous writing of drive parameters of the UNSIGNED INTEGER type on the local drive object.

Operation mode

The parameter number and the index of the parameter that is to be written are indicated at inputs PAR and IDX, respectively. If a parameter is not indexed, IDX = 0 must be set. The parameter is always written on the drive object on which the chart with the block is computed. It is not possible to access parameters on several drive objects.

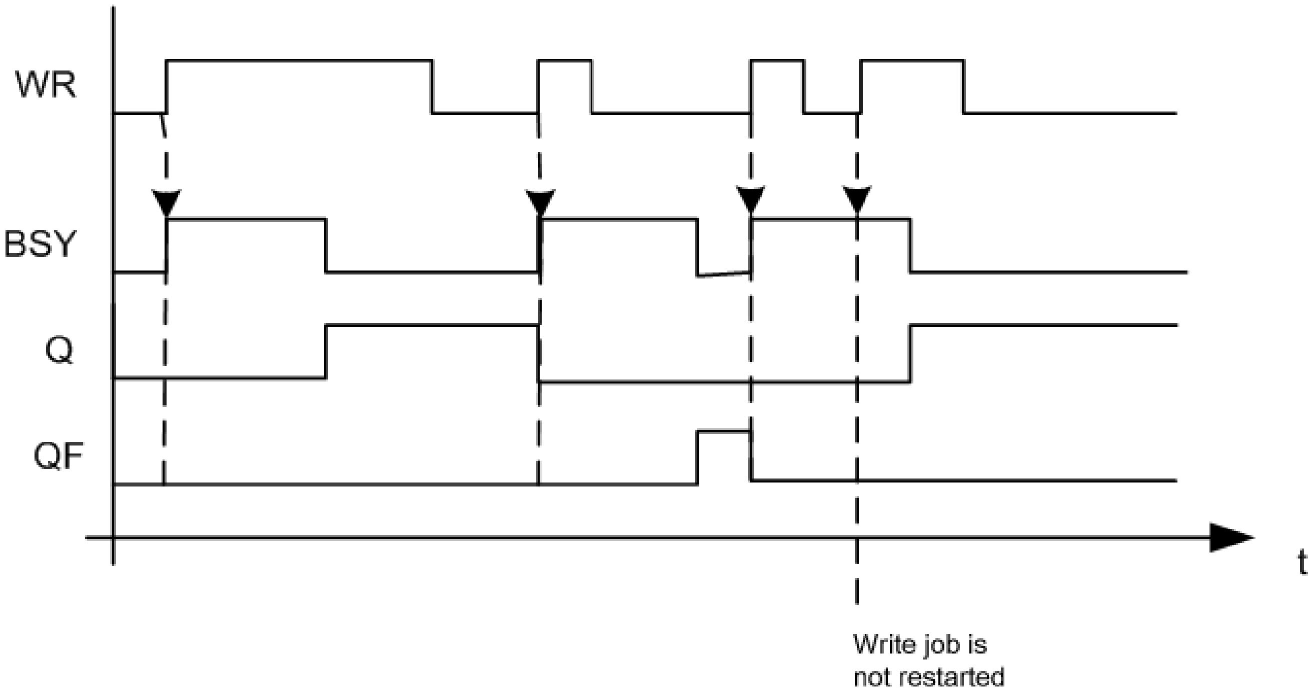
The parameter value is specified via input X. The asynchronous write job can be started on a positive edge at input WR. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and can vary from job to job. During an active write job, any additional positive edges at input WR are ignored.

Output Q = 1 indicates that the parameter has been written successfully. If an error occurs during the access, this is signaled with QF = 1.

For an error diagnostic, the error code ERC can be evaluated. ERC corresponds to the error code for parameter access according to PROFIdrive DPV1. The possible error codes can be found in Appendix A.2 of this document or in the SINAMICS Function Manual FH1 in Section PROFIBUS DP / PROFINET IO Communication and there in the Subsection Communication according to PROFIdrive → Acyclic communication → Configuration of the jobs and responses in Table Error values in DPV1 parameter responses.

ERC is only valid as long as QF = 1.

Time diagram



Quantity framework

Any number of asynchronous jobs of different block instances can be issued in parallel. Each block instance can only process one job.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
PAR	Parameter number	0	0..2^{16}	
IDX	Parameter index	0	0..2^{16}	
WR	Start write job	0	0/1	
X	Parameter value	0	UINT	
BSY	Write job is active	0	0/1	
Q	Write job is successful	0	0/1	
QF	Error	0	0/1	
ERC	Error code	16#0000	WORD	

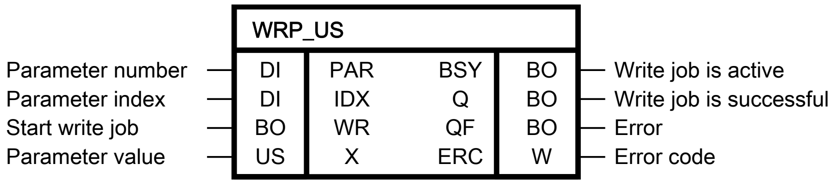
Project data

Can be loaded online	No
Special characteristics	-

6.30 WRP_US Writing drive parameters (UNSIGNED SHORT INTEGER type)

□ SIMOTION ☑ SINAMICS

Symbol


WRP_US
Parameter number — DI PAR BSY BO — Write job is active
Parameter index — DI IDX Q BO — Write job is successful
Start write job — BO WR QF BO — Error
Parameter value — US X ERC W — Error code

Short description

WRP_US (Write Parameter) enables the asynchronous writing of drive parameters of the UNSIGNED SHORT INTEGER type on the local drive object.

Operation mode

The parameter number and the index of the parameter that is to be written are indicated at inputs PAR and IDX, respectively. If a parameter is not indexed, IDX = 0 must be set. The parameter is always written on the drive object on which the chart with the block is computed. It is not possible to access parameters on several drive objects.

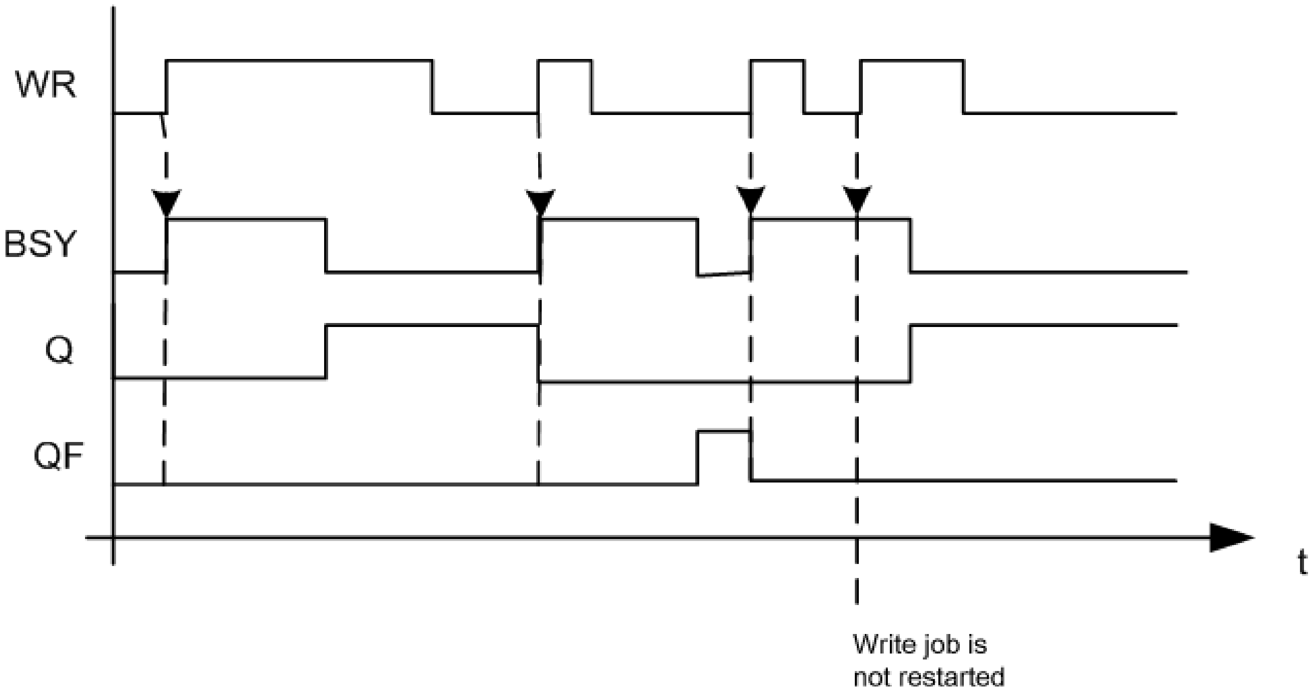
The parameter value is specified via input X. The asynchronous write job can be started on a positive edge at input WR. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and can vary from job to job. During an active write job, any additional positive edges at input WR are ignored.

Output Q = 1 indicates that the parameter has been written successfully. If an error occurs during the access, this is signaled with QF = 1.

For an error diagnostic, the error code ERC can be evaluated. ERC corresponds to the error code for parameter access according to PROFIdrive DPV1. The possible error codes can be found in Appendix A.2 of this document or in the SINAMICS Function Manual FH1 in Section PROFIBUS DP / PROFINET IO Communication and there in the Subsection Communication according to PROFIdrive → Acyclic communication → Configuration of the jobs and responses in Table Error values in DPV1 parameter responses.

ERC is only valid as long as QF = 1.

Time diagram



Quantity framework

Any number of asynchronous jobs of different block instances can be issued in parallel. Each block instance can only process one job.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
PAR	Parameter number	0	0..2^{16}	
IDX	Parameter index	0	0..2^{16}	
WR	Start write job	0	0/1	
X	Parameter value	0	USINT	
BSY	Write job is active	0	0/1	
Q	Write job is successful	0	0/1	
QF	Error	0	0/1	
ERC	Error code	16#0000	WORD	

Project data

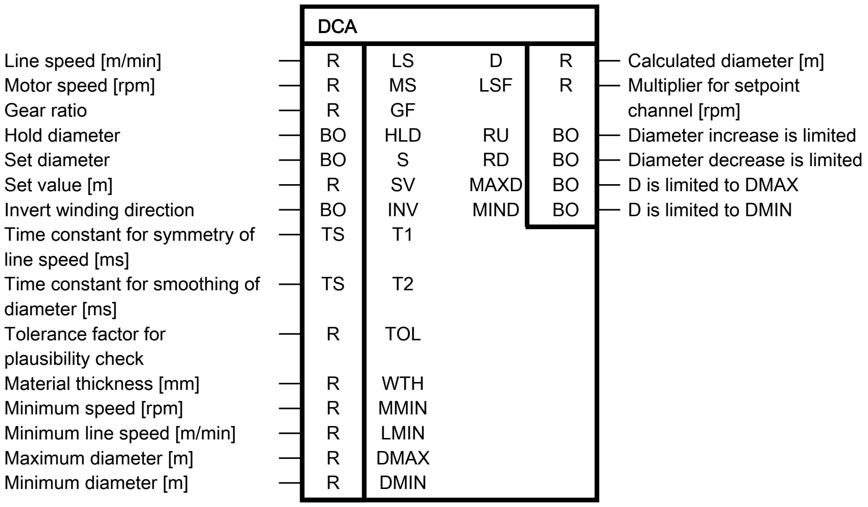
Can be loaded online	No
Special characteristics	-

Technology 7

7.1 DCA Diameter calculator

☑ SIMOTION ☑ SINAMICS

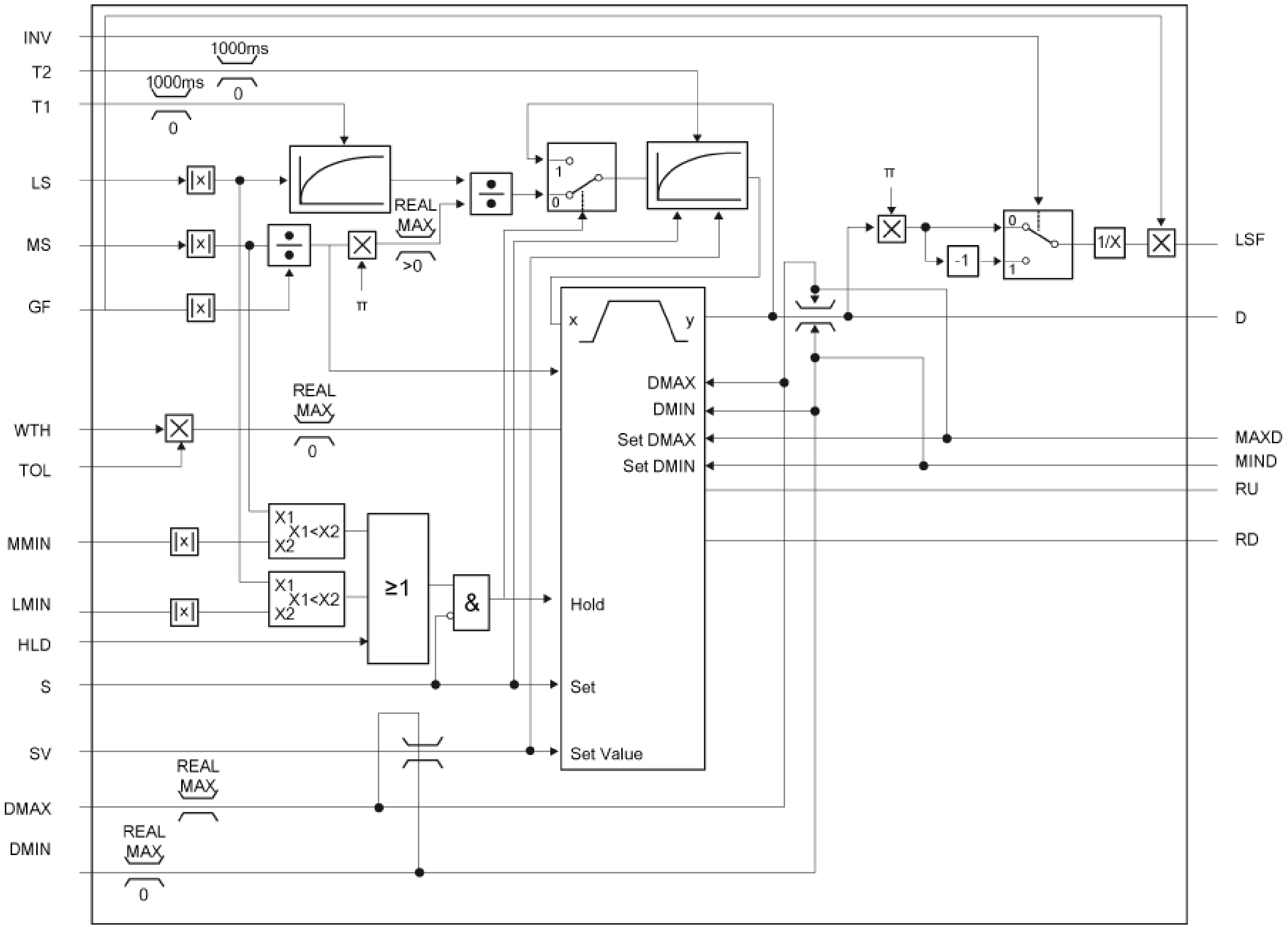
Symbol


DCA
Line speed [m/min] R LS
Motor speed [rpm] R MS
Gear ratio R GF
Hold diameter BO HLD
Set diameter BO S
Set value [m] R SV
Invert winding direction BO INV
Time constant for symmetry of line speed [ms] TS T1
Time constant for smoothing of diameter [ms] TS T2
Tolerance factor for plausibility check R TOL
Material thickness [mm] R WTH
Minimum speed [rpm] R MMIN
Minimum line speed [m/min] R LMIN
Maximum diameter [m] R DMAX
Minimum diameter [m] R DMIN
D R Calculated diameter [m]
LSF R Multiplier for setpoint channel [rpm]
RU BO Diameter increase is limited
RD BO Diameter decrease is limited
MAXD BO D is limited to DMAX
MIND BO D is limited to DMIN

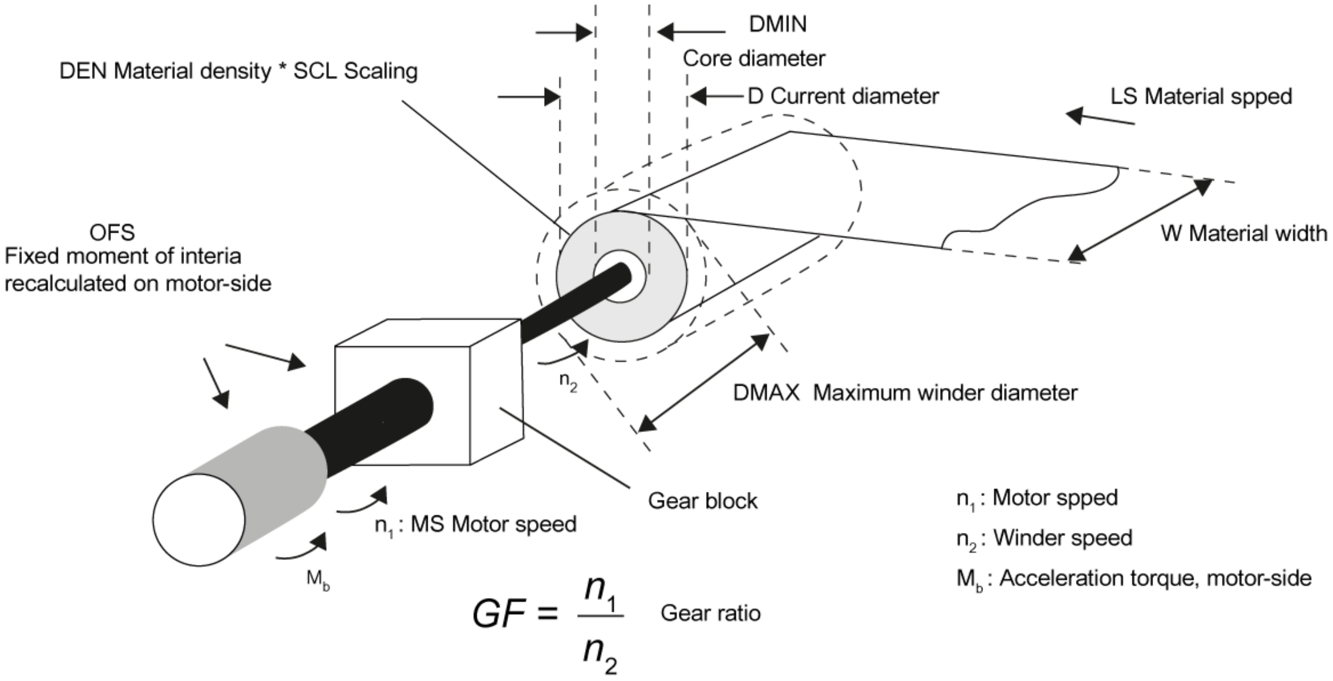
Short description

Technological block for axial winder applications. It is used to determine the current diameter of an axial winder based on the line speed and the motor speed. The calculated diameter is checked for plausibility.

Block diagram



Operation mode



The block cyclically calculates the diameter of an axial winder on the basis of the current line speed and the motor speed, which have to be supplied via the LS and MS inputs. The current motor speed is specified at input MS. The line speed can be delayed by the time T1 compared to the motor speed.

The current diameter is then calculated using the following formula:

$$\text{Durchmesser} = \frac{\text{Bahngeschwindigkeit} \bullet \text{Getriebeübersetzung}}{\text{Motordrehzahl} \bullet \pi}$$

The result can then be smoothed again using a smoothing element with time constant T2. The smoothing filters T1 and T2 have PT1 behavior. If the time constant T1 or T2 = 0, the smoothing input value is written directly to the output. The diameter calculation is only performed if the line speed LS or motor speed MS is greater than the respective threshold value LMIN or MMIN. Otherwise, the last calculated diameter value is held. With hold, the smoothing T2 is switched to the derived diameter D. The holding of diameter D can also be activated directly via the input HLD = 1. Input SV can be used to assign a preset value to the diameter; this diameter is applied when S =1. Smoothing element T2 is also initialized with this value. Only when S = 0 is the calculation for D and the smoothing T2 enabled. The setting of the diameter is dominant over the holding.

After smoothing element T2, the calculated diameter is checked for plausibility and corrected if a violation is identified. This test function is equivalent to that of a single ramp-function generator. The ramp-up time or ramp-down time is calculated dynamically from the material thickness WTH, tolerance factor TOL, and the winding speed. When material thickness WTH = 0, the plausibility check has no effect.

The maximum diameter change ΔD_{max} per scan interval is determined as follows:

$$\Delta D_{\max} = TOL \cdot 2 \cdot \frac{MS}{60 \cdot GF} \cdot \frac{WTH}{1000} \cdot T_A$$

with:

ΔD_{max}	Maximum diameter change [m] per scan interval
TOL	Tolerance factor
MS	Motor speed [rpm]
GF	Gear ratio
WTH	Material thickness [mm]
T_A	Block sampling time [s]

The resulting diameter D is limited as follows:

$D_n \leq D_{n-1} + \Delta Dmax_n$; for D_n(unlimited)$\geq D_{n-1}$ (ramp-up limiting)

$D_n \geq D_{n-1} - \Delta Dmax_n$; for D_n(unlimited)$\leq D_{n-1}$ (ramp-down limiting)

Output RU (ramp-up limiting) or RD (ramp-down limiting) is set in order to signal externally that limiting is in effect. If the limitation is deactivated again, the corresponding output is also set to zero. Both outputs are reset with Hold = 1 or Set = 1. When setting the diameter, the ramp-function generator has no effect. The plausibility check is connected downstream of a limiter. If the current diameter is limited to DMAX, output MAXD = 1 is set. A limitation to DMIN is signaled at output MIND. When the limitation is active, the ramp-function generator is corrected with the effective limit value in order to avoid a "large-scale integration" (anti-windup). In this case, the following applies to the next ramp-function generator cycle:

D_{n-1} = $DMAX_{n-1}$ if limitation is performed to DMAX

D_{n-1} = $DMIN_{n-1}$ if limitation is performed to DMIN

Output LSF cyclically supplies a multiplication factor for the setpoint channel in order to calculate the motor speed setpoint from the current line speed. The winding direction can be inverted with INV = 1.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
LS	Line speed [m/min]	0.0	0..REAL MAX	
MS	Motor speed [rpm]	1.0	0..REAL MAX	
GF	Gear ratio	1.0	0..REAL MAX	
HLD	Hold diameter	0	0/1	
S	Set diameter	0	0/1	
SV	Set value [m]	0.0	0..REAL MAX	
INV	Invert winding direction	0	0/1	
T1	Time constant for symmetry of line speed [ms]	0.0	0..REAL MAX	
T2	Time constant for smoothing of diameter [ms]	0.0	0..REAL MAX	
TOL	Tolerance factor for plausibility check	1,5	0..REAL MAX	
WTH	Material thickness [mm]	0.0	0..REAL MAX	
MMIN	Minimum speed [rpm]	1.0	0..REAL MAX	
LMIN	Minimum line speed [m/min]	0.1	0..REAL MAX	
DMAX	Maximum diameter [m]	0.1	0..REAL MAX	
DMIN	Minimum diameter [m]	0.01	0..REAL MAX	
D	Calculated diameter [m]	0.0	0..REAL MAX	
LSF	Multiplier for setpoint channel [rpm]	1.0	0..REAL MAX	
RU	Diameter increase is limited	0	0/1	
RD	Diameter decrease is limited	0	0/1	
MAXD	D is limited to DMAX	0	0/1	
MIND	D is limited to DMIN	0	0/1	

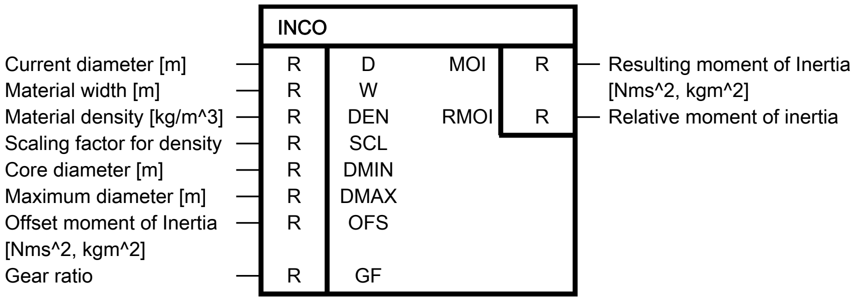
Project data

Can be loaded online	Yes
Special characteristics	-

7.2 INCO Axial winder moment of inertia

☑ SIMOTION ☑ SINAMICS

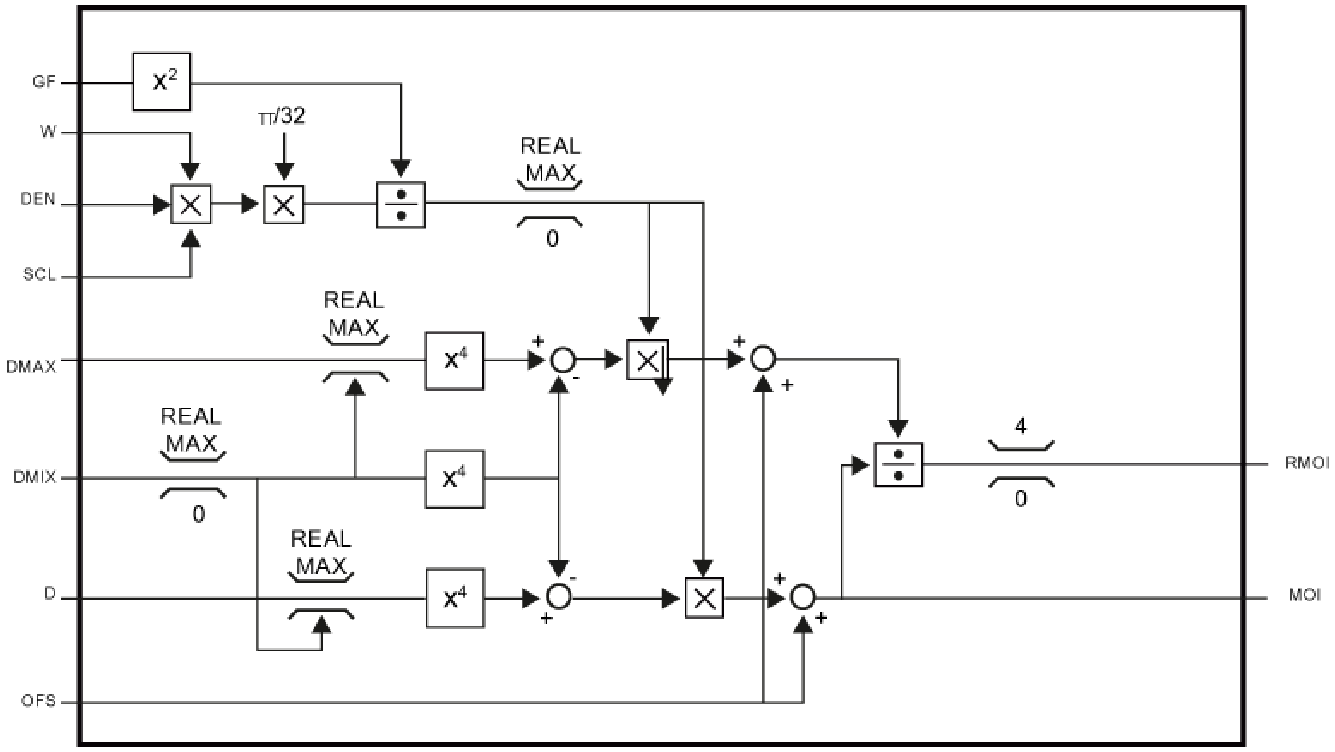
Symbol


INCO
Current diameter [m]
Material width [m]
Material density [kg/m^3]
Scaling factor for density
Core diameter [m]
Maximum diameter [m]
Offset moment of Inertia [Nms^2, kgm^2]
Gear ratio
R D
R W
R DEN
R SCL
R DMIN
R DMAX
R OFS
R GF
MOI R
RMOI R
Resulting moment of Inertia [Nms^2, kgm^2]
Relative moment of inertia

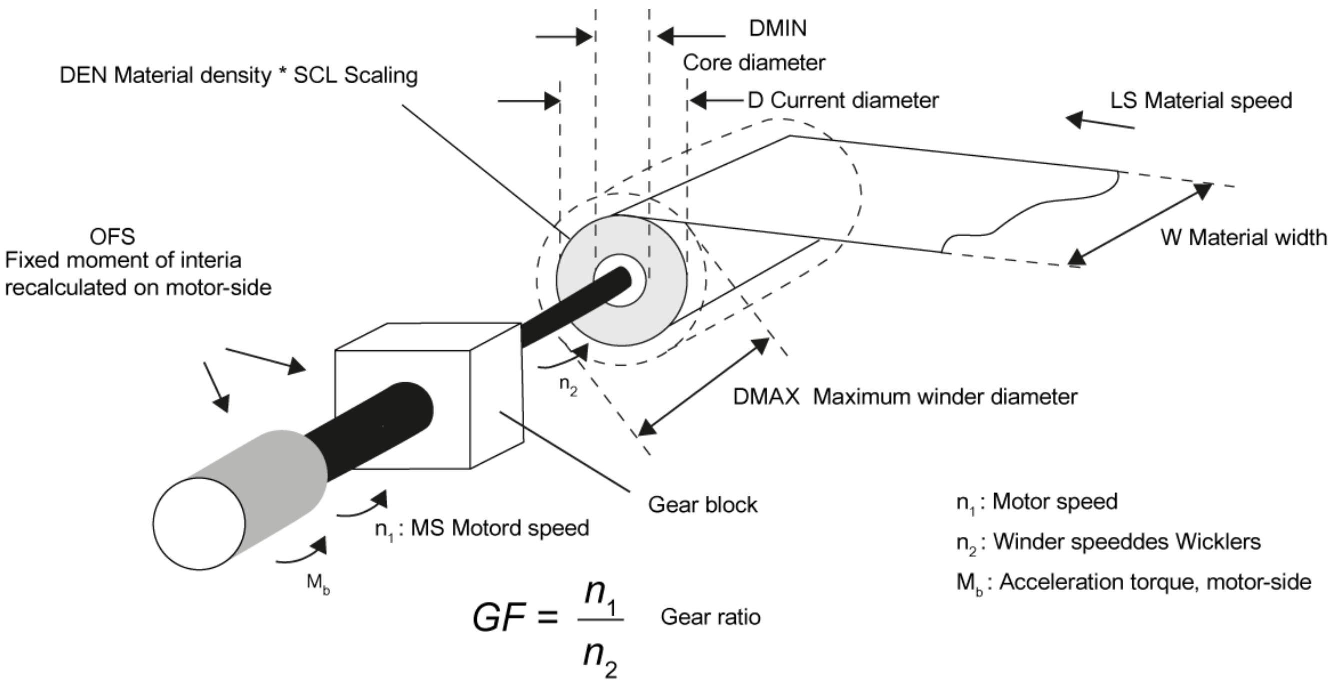
Short description

Technological block for axial winder applications. It is used for determining the moment of inertia of a winder, which is used to derive a torque feedforward control.

Block diagram



Operation mode



The block calculates the motor-side moment of inertia of an axial winder. Input variable D specifies the current diameter [m] of the winding. The density [kg/ m3] of the wound material can be specified via DEN and a correction factor for the density specified via SCL. The input variable DMIN [m] is used to specify the diameter of the coil core or the minimum diameter of the core. In order to be able to calculate the relative moment of inertia RMOI for a Kp adaptation of the speed controller, the block requires the maximum moment of inertia of the layout. To calculate this, the maximum winding diameter must be specified at input DMAX [m]. The total static moment of inertia (motor, empty winder and, if required gearbox) with regard to the motor side, can be specified via input OFS [Nms2, kgm2]. The gear ratio is specified at input GF. The current moment of inertia of the entire winder layout with regard to the motor side is specified at output MOI.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
D	Current diameter [m]	0.0	0..REAL MAX	
W	Material width [m]	0.0	0..REAL MAX	
DEN	Material density [kg/m^3]	0.0	0..REAL MAX	
SCL	Scaling factor for density	1.0	0..REAL MAX	
DMIN	Core diameter [m]	0.01	0..REAL MAX	
DMAX	Maximum diameter [m]	0.1	0..REAL MAX	
OFS	Offset moment of Inertia [Nms^2, kgm^2]	0.0	0..REAL MAX	
GF	Gear ratio	1.0	0..REAL MAX	
MOI	Resulting moment of Inertia [Nms^2, kgm^2]	0.0	0..REAL MAX	
RMOI	Relative moment of inertia	0.0	0..REAL MAX	

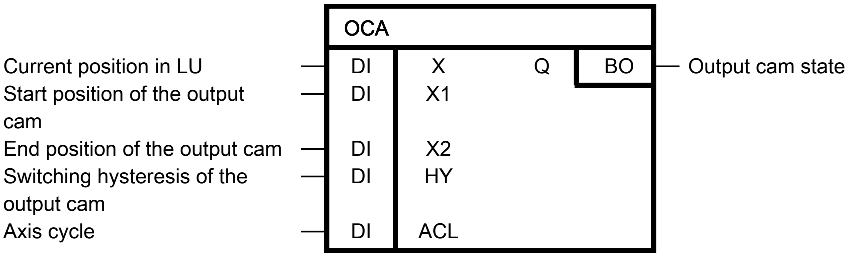
Project data

Can be loaded online	Yes
Special characteristics	-

7.3 OCA Software cam controller

☑ SIMOTION ☑ SINAMICS

Symbol

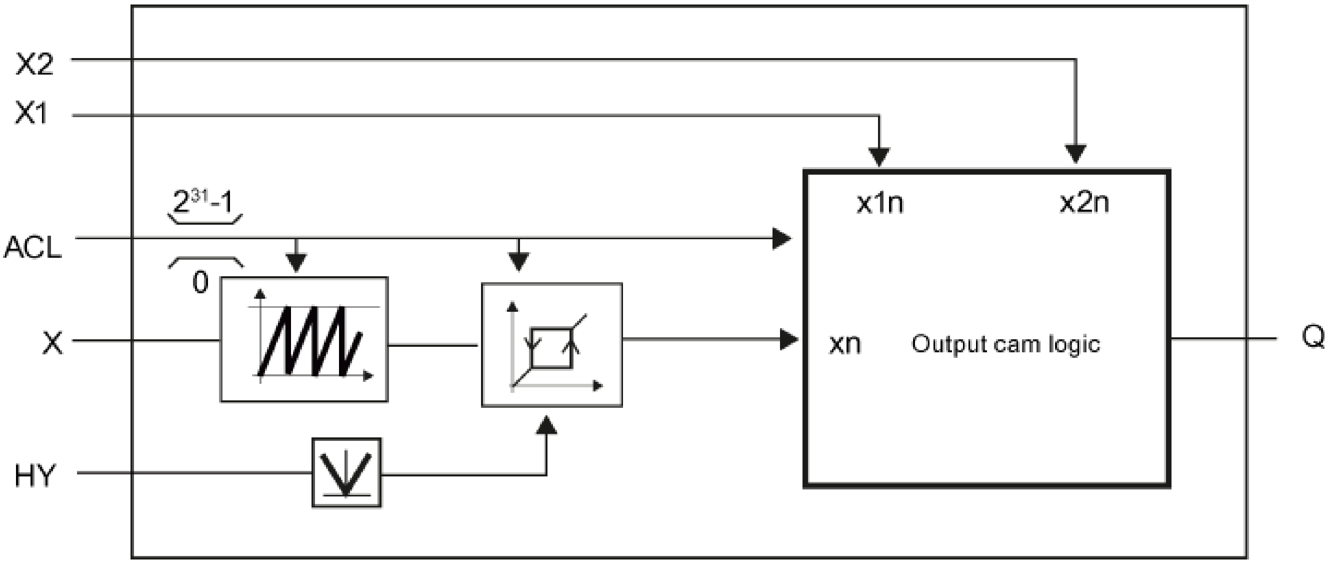

OCA
Current position in LU
DI X
Q BO
Output cam state
Start position of the output cam
DI X1
End position of the output cam
DI X2
Switching hysteresis of the output cam
DI HY
Axis cycle
DI ACL

Short description

Software cam controller with the following features:

- Position-based cam
- Switch-on/switch-off positions can be changed dynamically.
- Adjustable hysteresis for actual value-related output cam

Block diagram



Operation mode

The switch-on position of the position-based cam in positive direction of motion and the switch-off position in the negative direction are specified via input X1 [LU]. X2 [LU] specifies the switch-off position in positive direction of motion or the switch-on position in the negative direction. The axis cycle can be specified at input ACL so that the cam controller can also be operated with modulo axes. If ACL = 0, there is no modulo correction. A hysteresis band for input X can be set via HY, to ensure that no switching operations are performed at standstill for actual-value-related output cams.

The cam logic makes the following evaluation:

Non-modulo axis (ACL = 0)

x1n < x2n	Q = (x1n ≤ xn) AND (x2n > xn)
x1n ≥ x2n	Q = 0

Modulo axis (ACL <> 0):

x1n < x2n	Q = (x1n ≤ xn) AND (x2n > xn)
x1n > x2n	Q = (x1n ≤ xn) OR (x2n > xn)
x1n = x2n	Q = 0

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Current position in LU	0	DINT	
X1	Start position of the output cam	0	DINT	
X2	End position of the output cam	0	DINT	
HY	Switching hysteresis of the output cam	0	DINT	
ACL	Axis cycle	0	0...2^{31}-1	
Q	Output cam state	0	0/1	

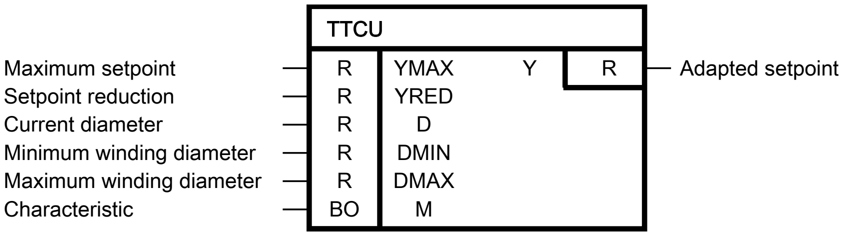
Project data

Can be loaded online	Yes
Special characteristics	-

7.4 TTCU Winding characteristic

☑ SIMOTION ☑ SINAMICS

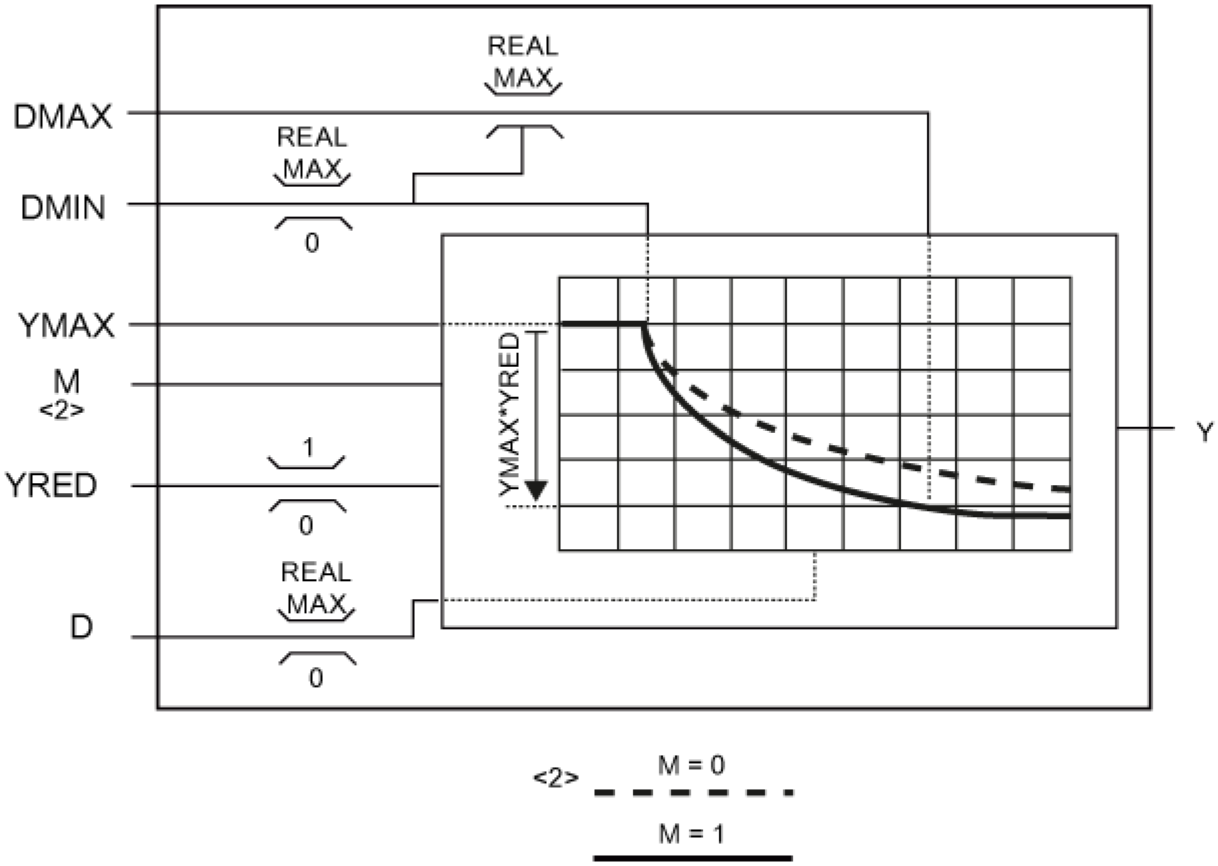
Symbol



Short description

Adaptation of a setpoint according to the specified characteristic. Block used for winder applications to determine the tension setpoint depending on the current winder diameter.

Block diagram



Operation mode

The reduction of the characteristic starts when D > DMIN is true. The input variable YRED specifies the degree of reduction in relation to the input variable YMAX. A curve characteristic can be preselected with input M which specifies the reduction behavior of the output variable with a rising input variable. If M = 0 has been preselected, the characteristic is reduced asymptotically by the factor YMAX*YRED. In this case, input variable DMAX is not taken into account. With the preselection M = 1, the input variable DMAX can be used to specify for which input variable D = DMAX, the characteristic runs through YMAX- YMAX*YRED.

The calculation of the characteristic is specified as follows:

D ≤ DMIN is true

$$Y = YMAX$$

D > DMIN and M = 0 (reaching of the reduction factor for D→∞)

$$Y = YMAX\left(1 - YRED\left(1 - \frac{DMIN}{D}\right)\right)$$

D > DMIN and M = 1 (attainment of reduction factor for D = DMAX)

DMAX > DMIN: $Y = YMAX\left(1 - YRED\frac{DMAX}{DMAX - DMIN}\left(1 - \frac{DMIN}{D}\right)\right)$

DMAX = DMIN: $Y = YMAX(1 - YRED)$

Block connections

Block connection	Description	Preassignment	Value range	Attributes
YMAX	Maximum setpoint	0.0	0..REAL MAX	
YRED	Setpoint reduction	0.0	0..1	
D	Current diameter	0.0	0..REAL MAX	
DMIN	Minimum winding diameter	1.0e-2	0..REAL MAX	
DMAX	Maximum winding diameter	0.1	0..REAL MAX	
M	Characteristic	1	0/1	
Y	Adapted setpoint	0.0	0..REAL MAX	

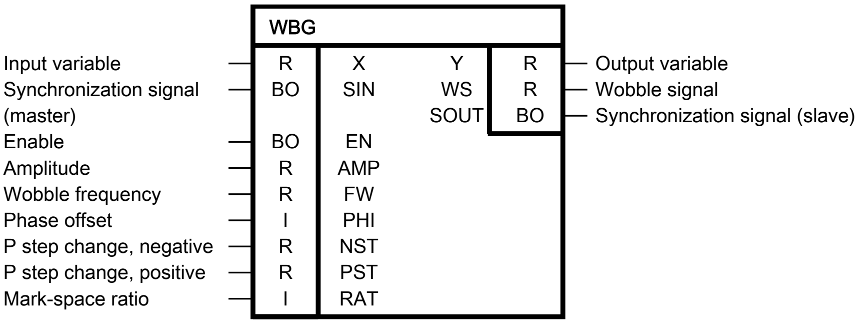
Project data

Can be loaded online	Yes
Special characteristics	-

7.5 WBG Wobble generator

☑ SIMOTION ☑ SINAMICS

Symbol

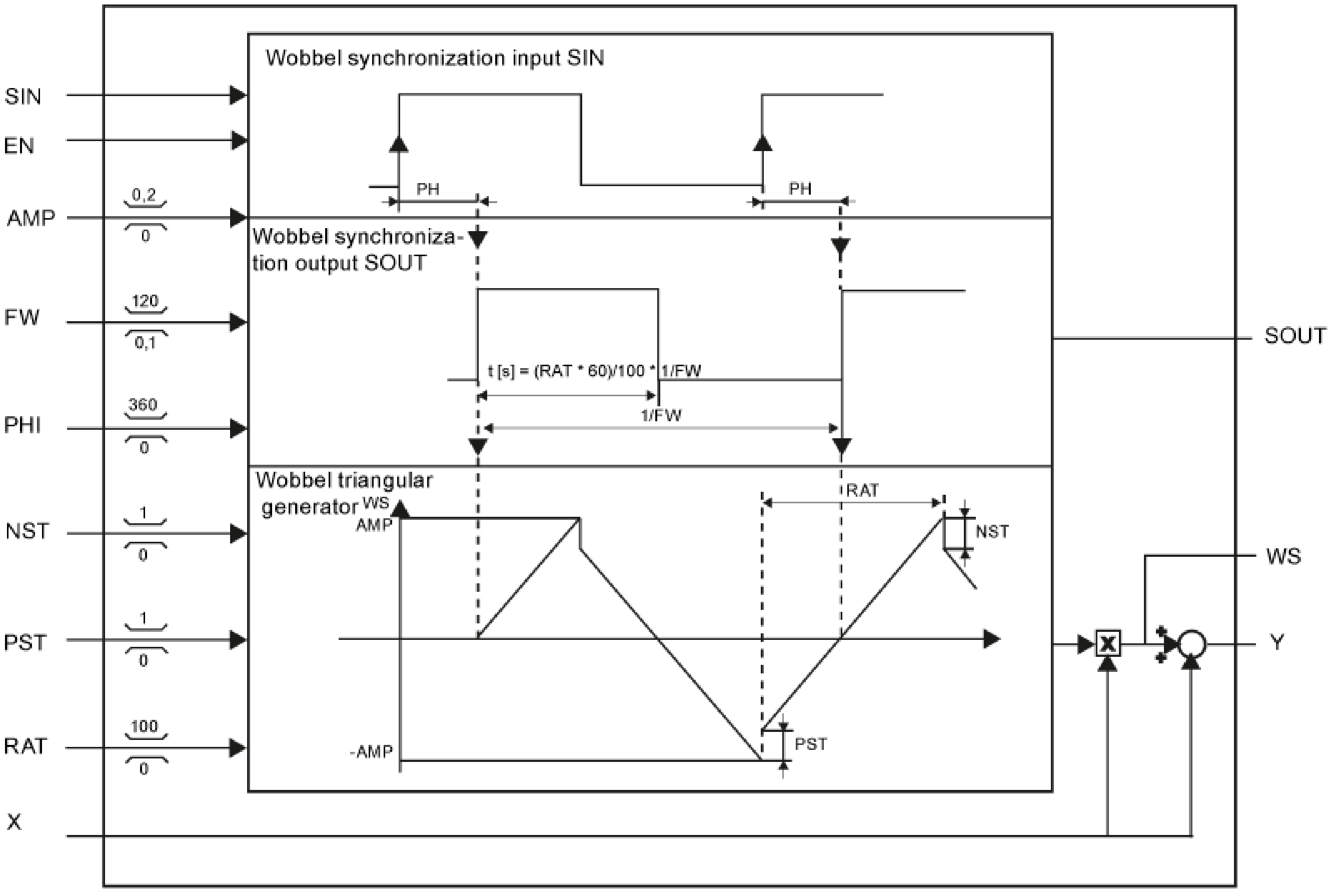


Short description

Triangular generator with adjustable frequency and amplitude for imprinting of "faults" on traversing drives for winding up textile threads. This generator has the following features:

- Positive and negative P step change, can be adjusted separately
- Synchronization to a master drive with an adjustable phase shift.
- Enabling input

Block diagram



Operation mode

The wobble generator is enabled with EN = 1. This triggers the output of wobble signal WS and synchronization signal SOUT. The signal generation is always started with a positive zero crossover or the positive edge of the synchronization output SOUT. If EN is reset again, wobble generation is continued until the next zero crossover of WS, and only then is the generator disabled and SOUT = 0. Input PHI (0-360°) can be used to set a phase shift between the positive edge of the synchronization input SIN and the start of the wobble signal. The signal is then generated for a signal period. For continuous signal generation, SIN must be used periodically as a trigger. If at a new start, the generation of the previous signal period is still running, it is cancelled. In the special case of PHI = 360, the free-running wobulation can be activated. The signal generation then runs periodically and is decoupled from the synchronization input SIN. The wobble signal is switched to input X and output at output Y.

Attributes of the wobble signal

Input	Value range	Description
AMP	0..0.2	Relative amplitude of the wobble signal
FW	0.1..120 rpm	Frequency of the wobble signal
PHI	0..360°	Phase shift of wobble signal relative to a positive edge at synchronization input SIN
NST	0.0..1.0	Relative, negative step change of wobble signal at the end of the positive signal edge
PST	0.0..1.0	Relative, positive step change of wobble signal at the end of the negative signal edge
RAT	0..100%	Ratio of rising signal edge / signal period

Effective amplitude(WS) = ABS(X) * AMP

Effective negative step change = -ABS(X) * AMP * NST

Effective positive step change = ABS(X) * AMP * PST

Ratio of rising edge / falling edge = RAT/(100-RAT)

If the attributes of the wobble signal are changed dynamically, the changed attributes take effect at the start of a new signal period (positive zero crossover).

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
SIN	Synchronization signal (master)	0	0/1	
EN	Enable	0	0/1	
AMP	Amplitude	0.0	0..0.2	
FW	Wobble frequency	60	0.1..120	
PHI	Phase offset	360	0..360	
NST	P step change, negative	0.0	0.0..1.0	
PST	P step change, positive	0.0	0.0..1.0	
RAT	Mark-space ratio	50	0..100	
Y	Output variable	0.0	REAL	
WS	Wobble signal	0.0	REAL	
SOUT	Synchronization signal (slave)	0	0/1	

Project data

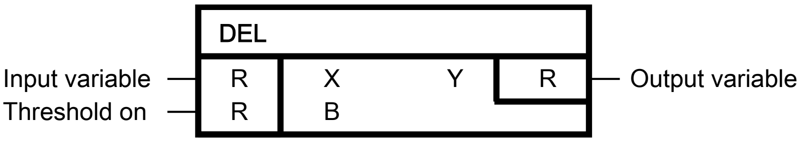
Can be loaded online	Yes
Special characteristics	-

Closed-loop control 8

8.1 DEL Dead zone element

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

- Adjustable dead band
- Set zero-point symmetric value range to zero.

Operation mode

- If the absolute value of X is less than B, then Y = 0
- If X is greater than or equal to B, then Y = X - B
- If X is less than or equal to B, then Y = X + B

The zero-point symmetric dead band can be set with operating value B.

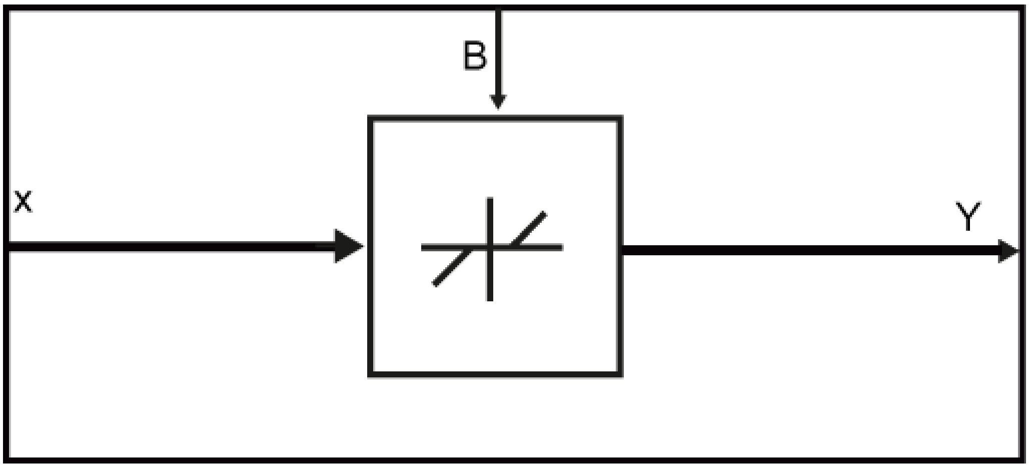
Algorithm:

with boundary condition B >=0

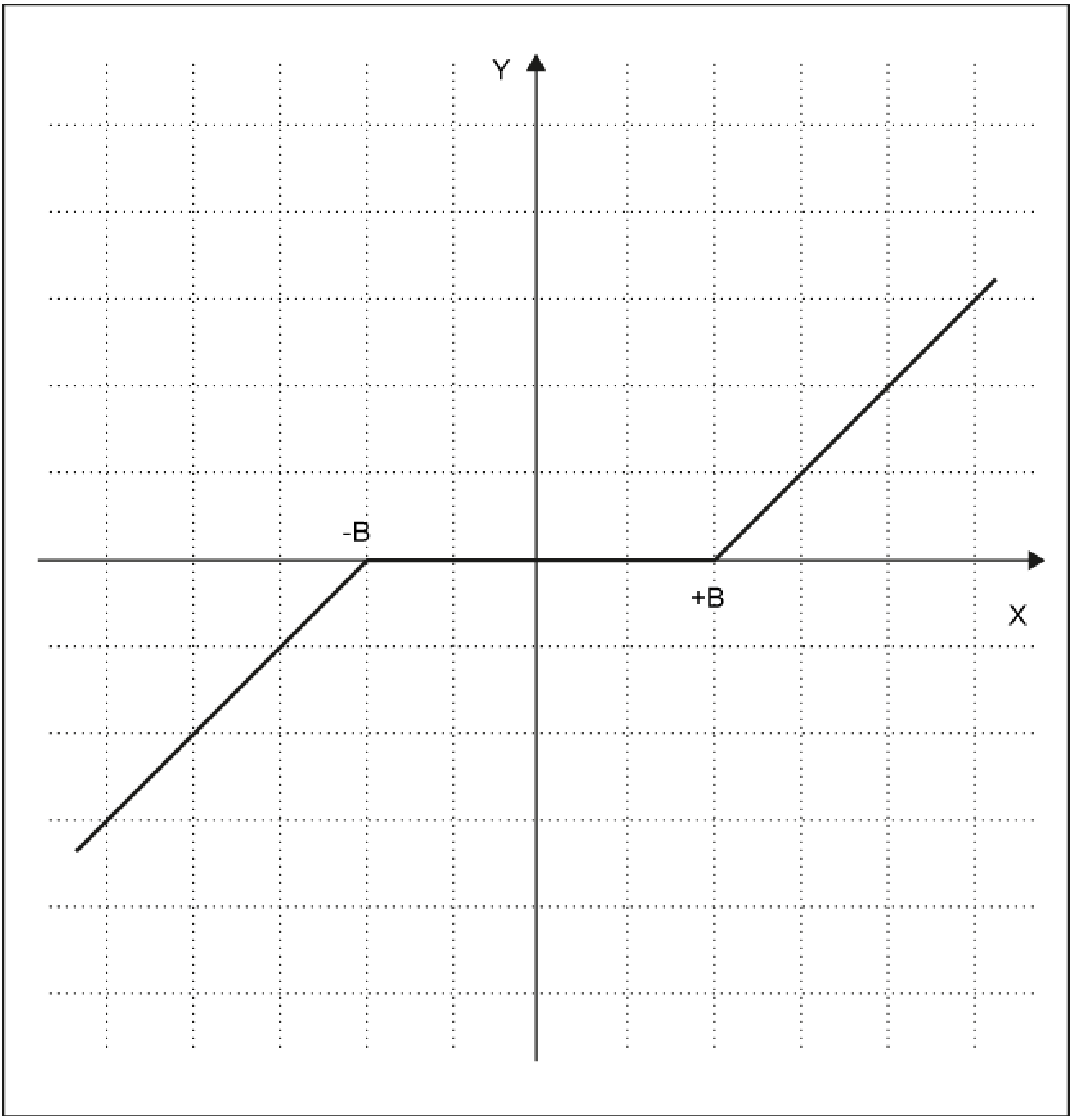
For B < 0, the following applies for all X: Y = X.

$$Y = \begin{cases} X + B \text{ for } X <= -B \\ 0 \text{ for } -B < X < B \\ X - B \text{ for } X >= B \end{cases}$$

Block diagram



XY diagram



Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
B	Threshold on	0.0	REAL	
Y	Output variable	0.0	REAL	

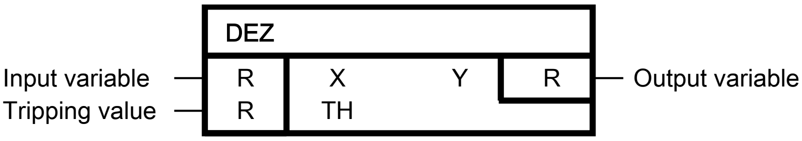
Project data

Can be loaded online	Yes
Special characteristics	-

8.2 DEZ Dead zone element

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

- Adjustable dead zone
- Set zero-point symmetric value range to zero

Operation mode

If the absolute value of X is less than TH, then Y = 0.

If the absolute value of X is greater than or equal to TH, then Y = X.

The zero-point symmetric dead zone can be set with operating value TH.

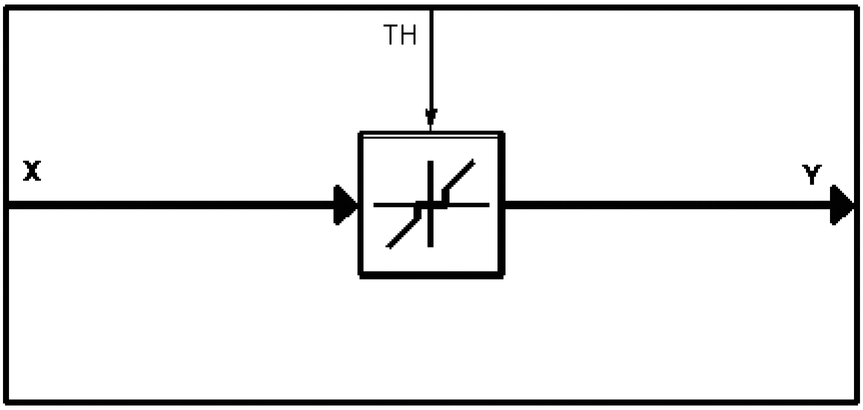
Algorithm:

$$Y = \begin{cases} X & \text{for } X <= -TH \\ 0 & \text{for } -TH < X < TH \\ X & \text{for } X >= TH \end{cases}$$

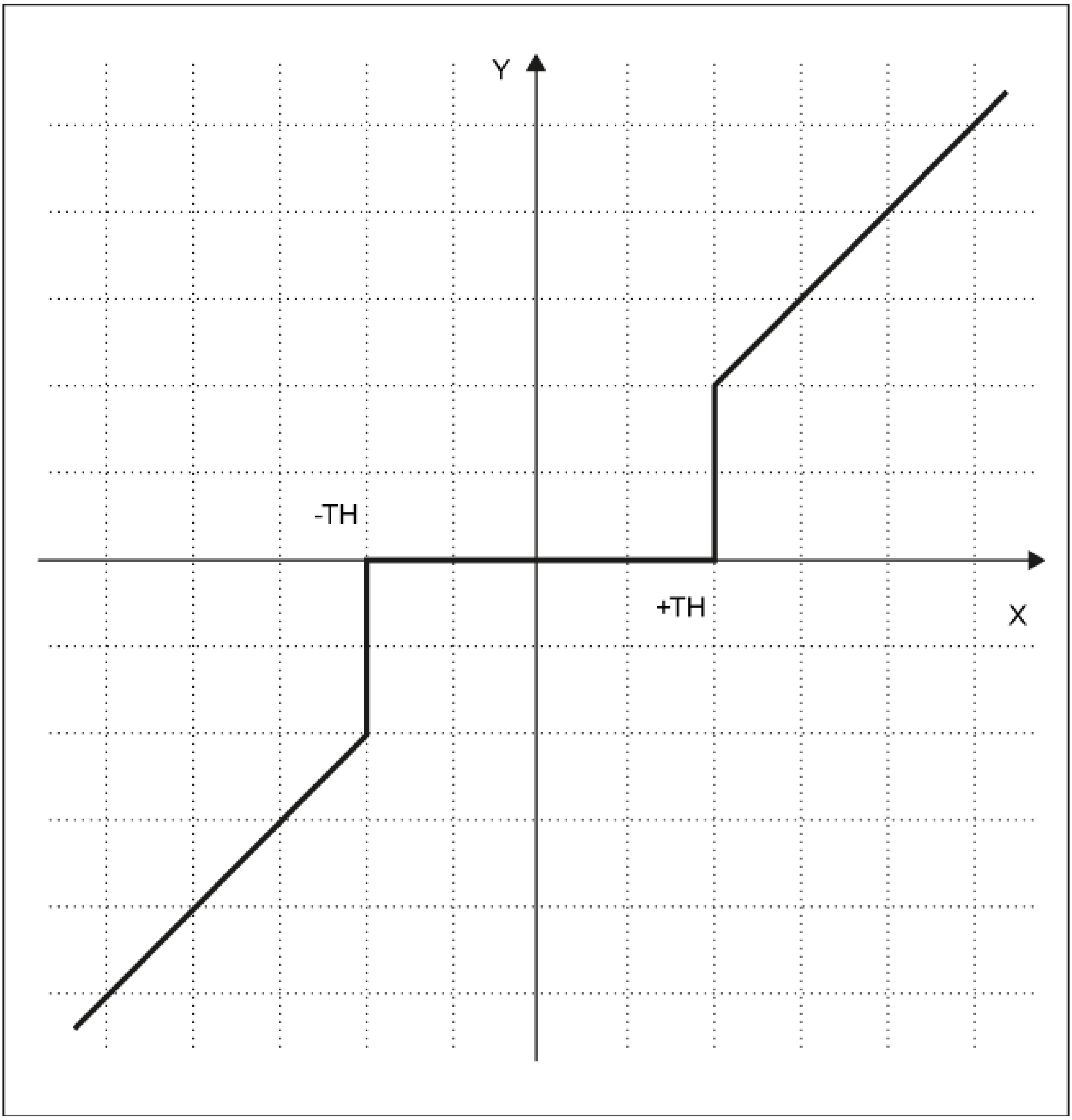
with boundary condition TH >=0

For TH< 0, the following applies for all X: Y = X.

Block diagram



XY diagram



Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
TH	Tripping value	0.0	REAL	
Y	Output variable	0.0	REAL	

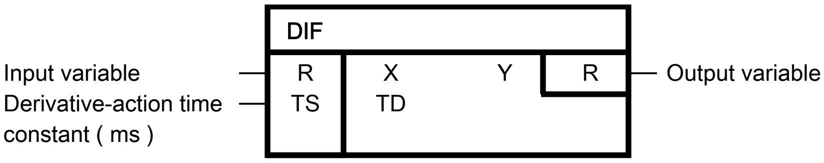
Project data

Can be loaded online	Yes
Special characteristics	-

8.3 DIF Derivative-action element

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

Block with derivative-action response

Operation mode

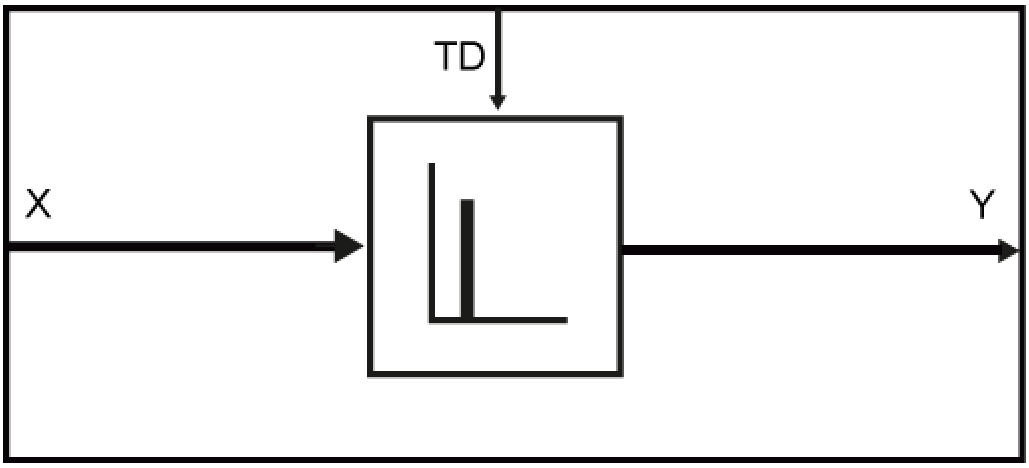
- Output variable Y is proportional to the change velocity of input variable X, multiplied by the derivative-action time constant TD.
- Discrete values are calculated according to the algorithm:

Algorithm:

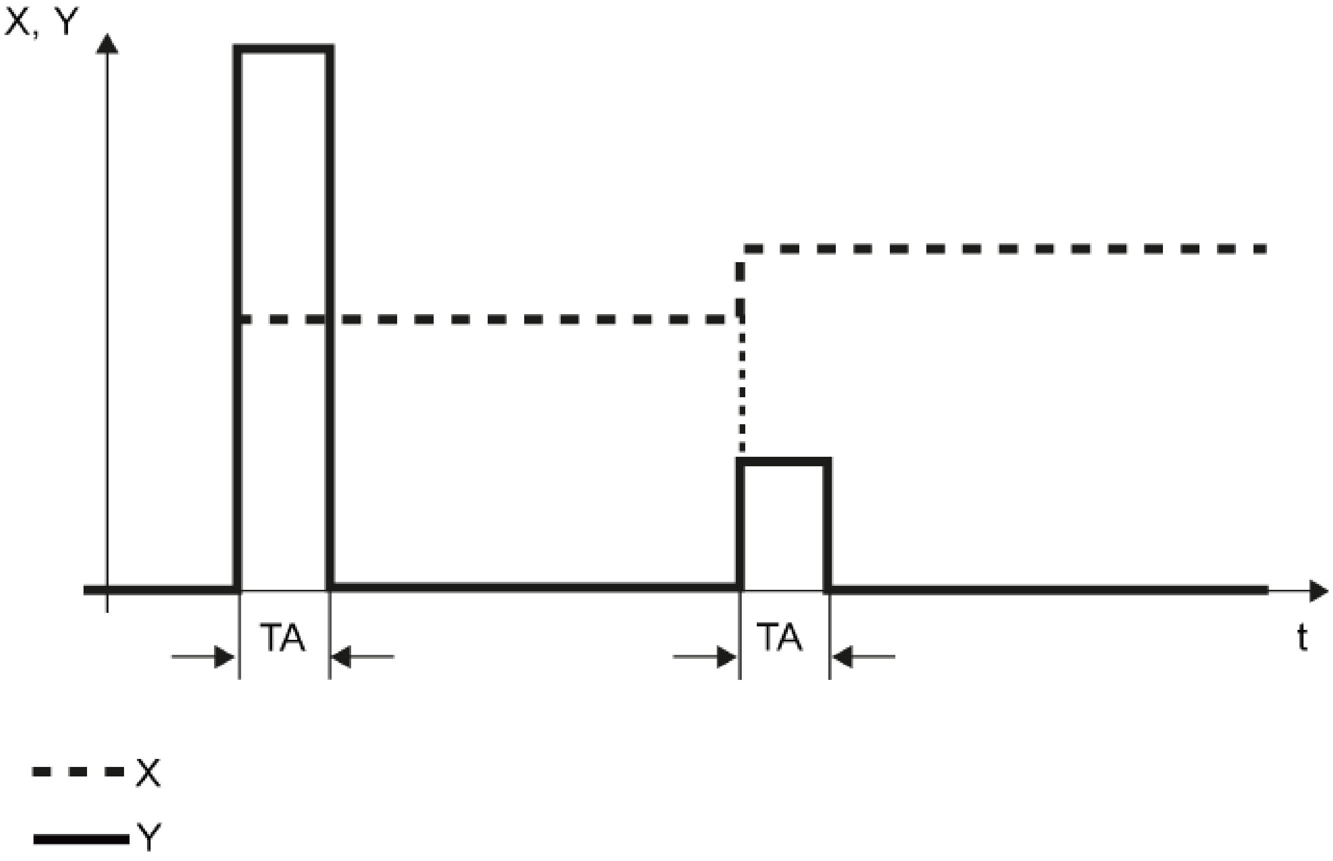
$$Y_n = \left(X_n - X_{n-1}\right) \cdot \frac{TD}{TA}$$

Yn	Value of Y in scan interval n
Xn	Value of X in scan interval n
Xn-1	Value of X in scan interval n-1

Block diagram



XY diagram



Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
TD	Derivative-action time constant (ms)	0	SDTIME	
Y	Output variable	0.0	REAL	

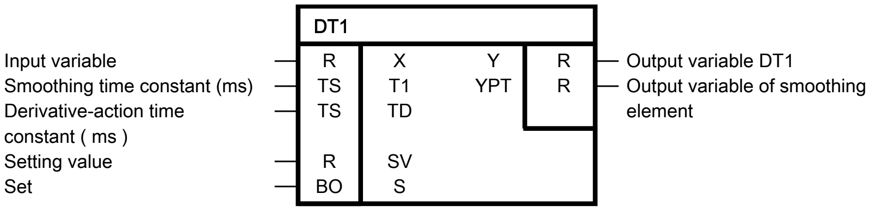
Project data

Can be loaded online	Yes
Special characteristics	-

8.4 DT1 Smoothing element

☑ SIMOTION ☑ SINAMICS

Symbol


DT1
Input variable
Smoothing time constant (ms)
Derivative-action time constant (ms)
Setting value
Set
R
TS
TS
R
BO
X
T1
TD
SV
S
Y
YPT
R
R
Output variable DT1
Output variable of smoothing element

Short description

Block with derivative-action response and smoothing. The DT1 output can be set.

Operation mode

Setting function not active (S = 0)

Input variable X, dynamically delayed by smoothing time constant T1, is given to a derivative-action element and the block output YPT.

Output variable Y of the entire DT1 element is proportional to the change velocity of YPT (differential quotient), multiplied by the derivative-action time constant TD.

T1 determines the steepness of the decline of the output variable. It specifies the time at which the transfer function has fallen to 37% of X·TD/T1 after the smoothing and differentiation . With sufficiently large T1/TA (T1/TA>10), the transfer function corresponds to the characteristic of

$Y(t) = X\cdot(TD/T1)\cdot e^{-t/T1}$

with t = n·TA

Algorithm:

$$Y_n = \frac{TD}{T1} \cdot \left(X_n - YPT_{n-1}\right)$$

$$YPT_n = YPT_{n-1} + \frac{TA}{T1} \cdot \left(X_n - YPT_{n-1}\right)$$

YPTn	Value of YPT in scan interval n
Yn	Value of Y in scan interval n
Xn	Value of X in scan interval n
YPTn-1	Value of YPT in scan interval n-1

The larger T1/TA is, the smaller is the amplitude change on Y and YPT from one sampling time to the next. TA is the sampling time in which the block is configured. The larger TD/TA is, the larger is the amplitude change on Y from one sampling time to the next. TD and T1 are limited internally: TD >= 0, T1 >= TA.

Setting function active (S = 1)

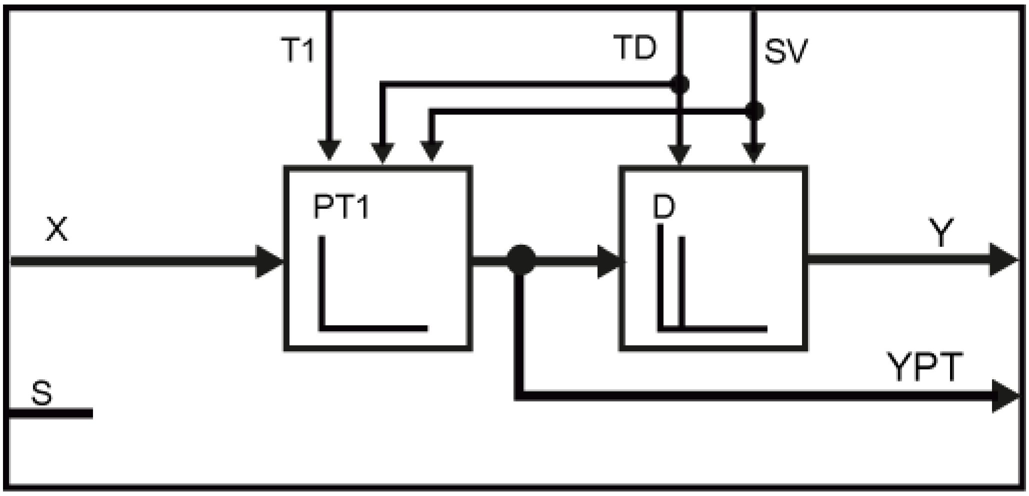
With active setting function, the setting value SV is applied at the dt1 output Y (Y=SV), the following results for the output of the smoothing element:

$$YPT_n = X_n - \frac{T1}{Td} \cdot SV_n$$

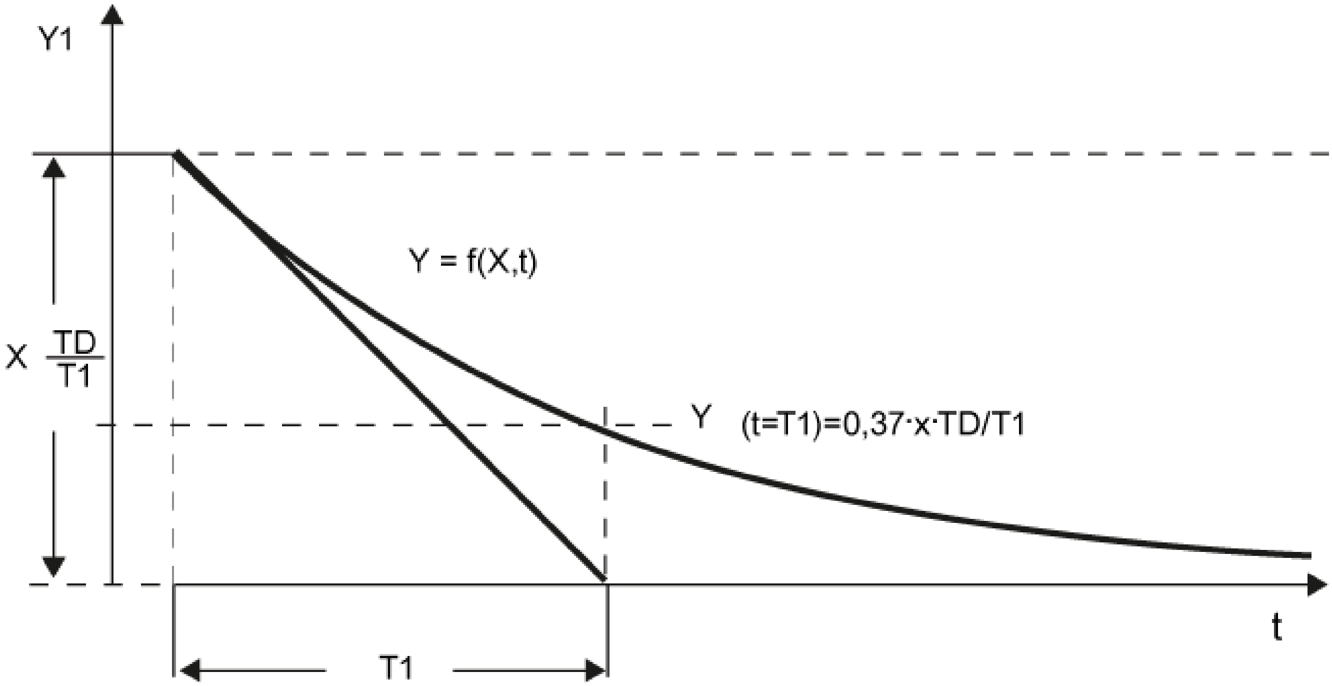
for TD ≠ 0

The internal limitations for T1 and TD apply in this case. When TD=0, the output variables remain unchanged, as long as S=1.

Block diagram



XY diagram



Initialization

If input S is logic 1 at the initialization, the setting value SV is applied at output Y and YPT = T1/TD*(X-SV) set.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
T1	Smoothing time constant (ms)	0.0	SDTIME	
TD	Derivative-action time constant (ms)	0.0	SDTIME	
SV	Setting value	0.0	REAL	
S	Set	0	BOOL	
Y	Output variable DT1	0.0	REAL	
YPT	Output variable of smoothing element	0.0	REAL	

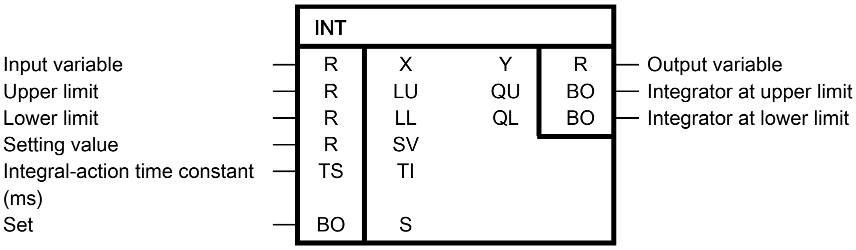
Project data

Can be loaded online	Yes
Special characteristics	-

8.5 INT Integrator

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

- Block with integral behavior
- Integrator functions:
- Set initial value
- Adjustable integral-action time constant
- Adjustable limits
- For normal integrator mode, a positive limit value must be specified for LU and a negative limit value must be specified for LL

Operation mode

The change in output variable Y is proportional to input variable X and inversely proportional to the integral-action time constant TI.

The output Y of the integrator can be limited via the inputs LU and LL. If the output reaches one of the two limits, a message is sent via the outputs QU or QL. If LL >= LU, then output Y = LU.

The calculation of the discrete values (TA is the sampling time in which the block is configured) is performed according to the following algorithm:

Algorithm:

$$Y_n = Y_{n-1} + \frac{TA}{TI} \cdot X_n$$

Yn	Value of Y in scan interval n
Yn-1	Value of Y in scan interval n-1
Xn	Value of X in scan interval n

When S = 1, the output variable Y is set to the setting value SV. Two functions can be realized via S:

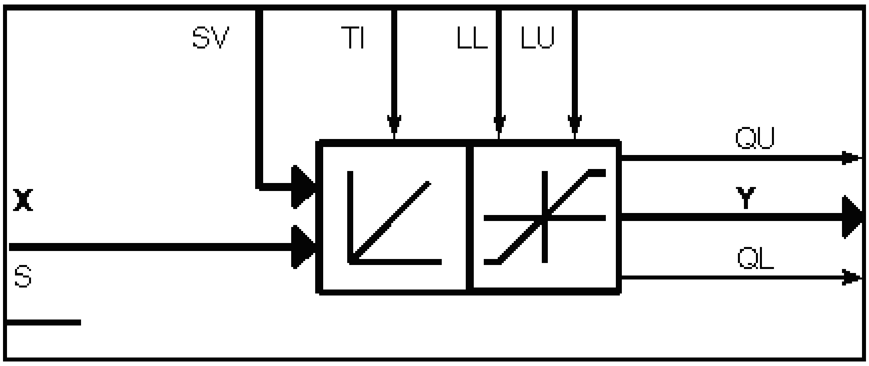
Track integrator (Y = SV)

The binary input is S = 1 and the setting value SV is changed. If applicable, the output makes a jump to the setting value immediately after the setting operation.

Set integrator to initial value SV

S is switched to 1. S is then set to 0, and the integrator starts from SV in the direction specified by the polarity of input variable X.

Block diagram



Truth table(s)

S	Conditions	Y	QU	QL	Mode
0	LL<Yn-1+X×TA/TI<LU	Yn	0	0	Integration
0	Yn-1+X×TA/TI >= LU	LU	1	0	INT at upper limit
0	Yn-1+X×TA/TI <= LL	LL	0	1	INT at lower limit
1	LL<SV<LU	SVn	0	0	Set
1	SV >= LU	LU	1	0	INT at upper limit
1	SV <= LL	LL	0	1	INT at lower limit

Truth table for LL>= LU

S	Conditions	Y	QU	QL	Mode
(any)	LL >= LU	LU	1	1	INT at upper limit

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
LU	Upper limit	0.0	REAL	
LL	Lower limit	0.0	REAL	
SV	Setting value	0.0	REAL	
TI	Integral-action time constant (ms)	0.0	SDTIME	
S	Set	0	0/1	
Y	Output variable	0.0	REAL	
QU	Integrator at upper limit	0	0/1	
QL	Integrator at lower limit	0	0/1	

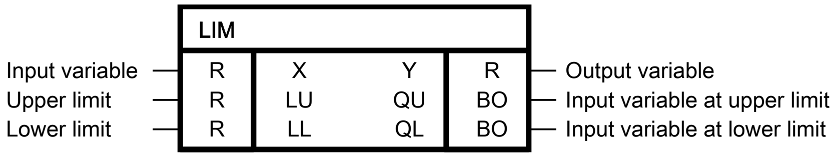
Project data

Can be loaded online	Yes
Special characteristics	-

8.6 LIM Limiter (REAL type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

- Block for the limitation
- Adjustable upper and lower limit
- Indication when set limits are reached

Operation mode

The block transfers the input variable X to its output Y. During which the input variable is limited depending on LU and LL.

If the input variable reaches the upper limit LU, then output QU = 1 is set.

If the input variable reaches the lower limit LL, then output QL = 1 is set.

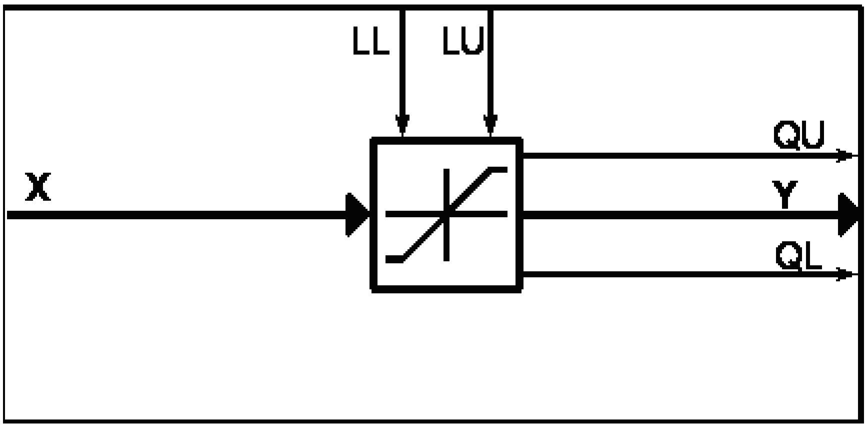
If the lower limit is greater than or equal to the upper limit, then output Y is set to the upper limit LU.

Algorithm:

$$Y = \begin{cases} LU \text{ for } X \geq LU \\ X \text{ for } LL < X < LU \\ LL \text{ for } X \leq LL \end{cases}$$

with boundary condition: LL < LU

Block diagram



Truth table(s)

Conditions	Y	QU	QL	Mode
LL < X < LU	X	0	0	
X >= LU	LU	1	0	Input variable at upper limit
X <= LL	LL	0	1	Input variable at lower limit

Truth table for LL >= LU

Conditions	Y	QU	QL	Mode
LL >= LU	LU	1	1	Input variable at upper limit

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
LU	Upper limit	0.0	REAL	
LL	Lower limit	0.0	REAL	
Y	Output variable	0.0	REAL	
QU	Input variable at upper limit	1	0/1	
QL	Input variable at lower limit	1	0/1	

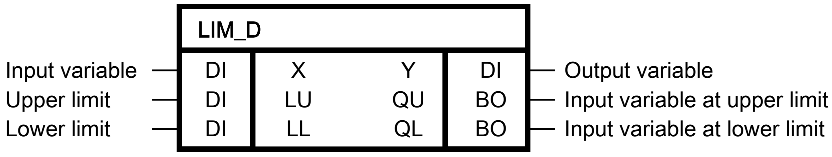
Project data

Can be loaded online	Yes
Special characteristics	-

8.7 LIM_D Limiter (DOUBLE INTEGER type)

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

- Block for the limitation of the DOUBLE INTEGER type
- Adjustable upper and lower limit
- Indication when set limits are reached

Operation mode

The block transfers the input variable X to its output Y. During which the input variable is limited depending on LU and LL.

If the input variable reaches the upper limit LU, then output QU = 1 is set.

If the input variable reaches the lower limit LL, then output QL = 1 is set.

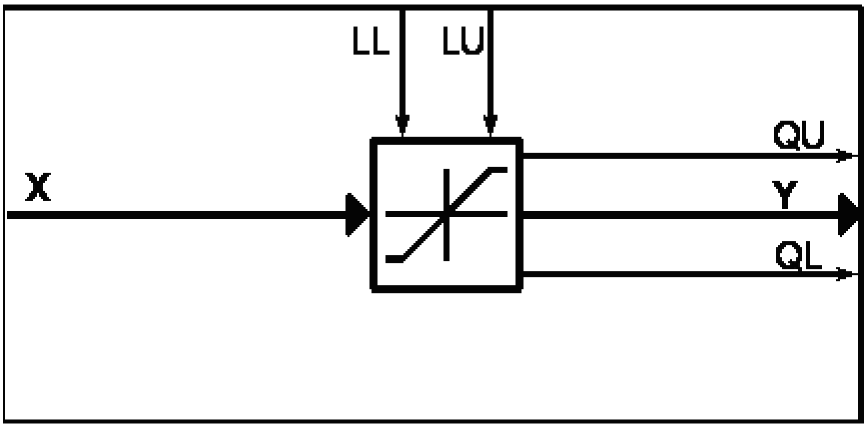
If the lower limit is greater than or equal to the upper limit, then output Y is set to the upper limit LU.

Algorithm:

$$Y = \begin{cases} LU \text{ for } X \geq LU \\ X \text{ for } LL < X < LU \\ LL \text{ for } X \leq LL \end{cases}$$

with boundary condition: LL < LU

Block diagram



Truth table(s)

Conditions	Y	QU	QL	Mode
LL < X < LU	X	0	0	
X >= LU	LU	1	0	Input variable at upper limit
X <= LL	LL	0	1	Input variable at lower limit

Truth table for LL >= LU

Conditions	Y	QU	QL	Mode
LL >= LU	LU	1	1	Input variable at upper limit

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0	DINT	
LU	Upper limit	0	DINT	
LL	Lower limit	0	DINT	
Y	Output variable	0	DINT	
QU	Input variable at upper limit	1	0/1	
QL	Input variable at lower limit	1	0/1	

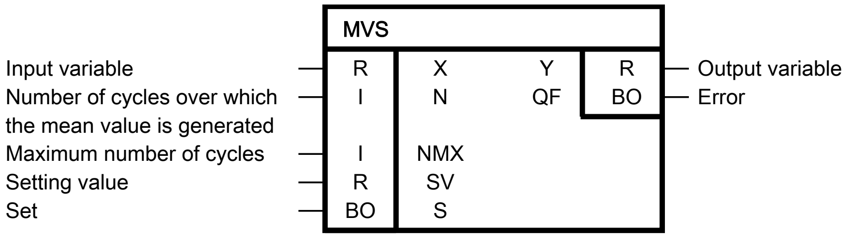
Project data

Can be loaded online	Yes
Special characteristics	-

8.8 MVS Sliding-type mean value generator

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

The block calculates a sliding-type mean value via the input variable X.

Operation mode

The mean value is generated over the last N cycles.

$$Y_k = \frac{1}{N} \cdot \sum_{i=k-(N-1)}^{k} X_i$$

X_k = X in cycle k

k = 0 is the current cycle

The number of cycles can be changed in the range 1 ≤ N ≤ NMX. The maximum number of cycles is specified through NMX and cannot be changed during operation. The block limits input N to the range of 1 ≤ N ≤ NMX. The buffer for the input values is always filled up to NMAX, irrespective of N. In this way, the block can re-determine the current mean value via all variables when there is a change in the window length.

The mean value is set to set value SV as long as S = 1.

Initialization

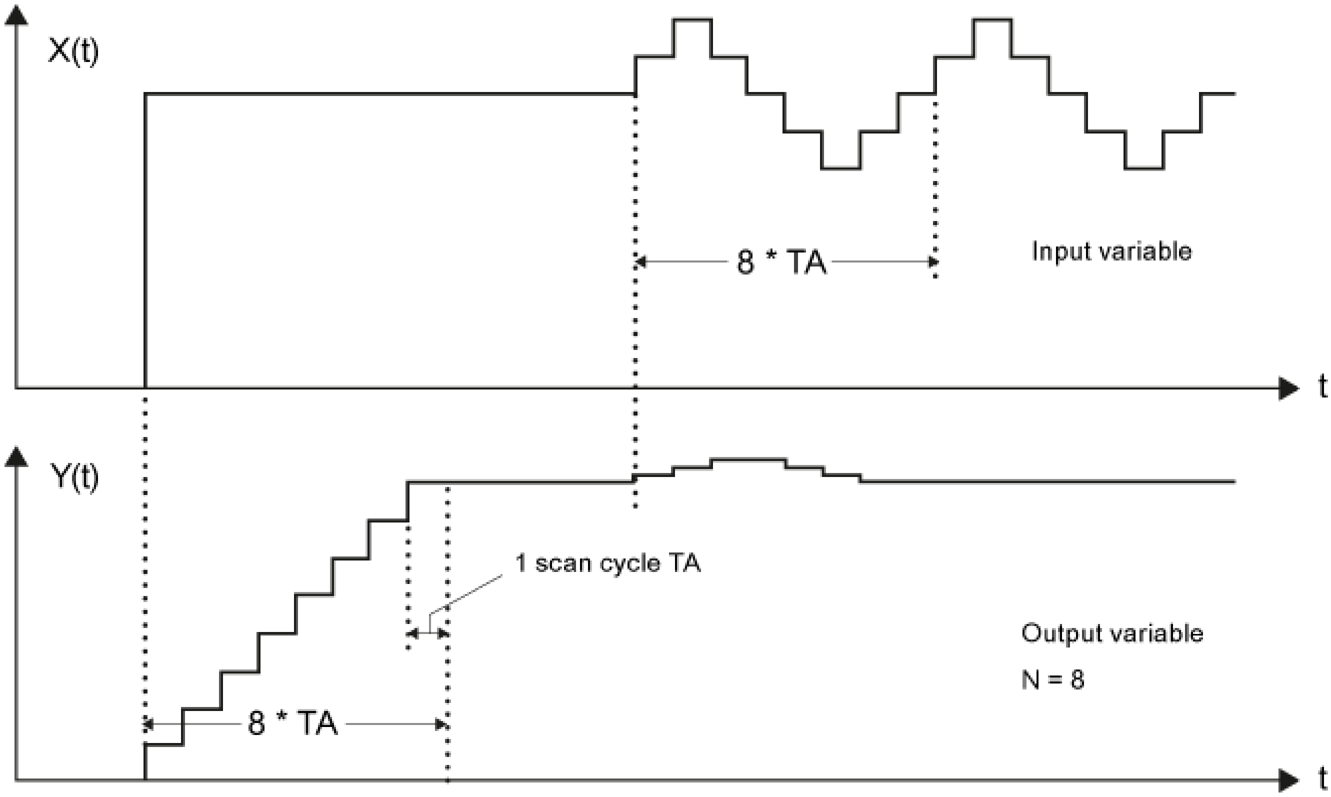
At the initialization, NMX is used to specify the maximum size of the mean value buffer for the X values. For this reason, NMX should be set to the maximum value of N required under operating conditions. The value of NMX is limited to the range of values between 1 and 1,000. If there is not enough memory for NMX on the target device, or if NMX is limited, output QF is set to 1 and output Y retains its default value during cyclic operation. As NMX cannot be changed dynamically during operation, NMX should be specified as a constant.

Fields of application

The block can be used as ramp-function generator or filter block for the mean value generation. It acts as a low pass and band-stop filter for frequencies f_k.

$$f_k = \frac{k}{N \cdot T_A}$$

k = 1, 2, ...



Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
N	Number of cycles over which the mean value is generated	10	1...1000	
NMX	Maximum number of cycles	100	1...1000	
SV	Setting value	0.0	REAL	
S	Set	0	0/ 1	
Y	Output variable	0.0	REAL	
QF	Error	0	0/1	

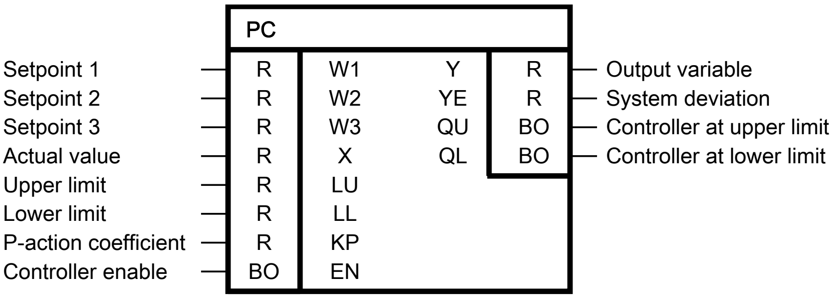
Project data

Can be inserted online	Yes
Special characteristics	-

8.9 PC P-action controller

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

- P controller with 3 setpoint inputs and 1 actual value input
- Sign reversal of actual value in block
- Indication when set limits are reached
- For normal controller operation, a positive limit value must be specified for LU and a negative limit value must be specified for LL.

Operation mode

The three setpoints W1, W2, and W3 are added and the actual value X is subtracted from the setpoint total. The result YE is multiplied by the proportional coefficient KP and given to output Y.

Algorithm:

$Y = KP \cdot YE = KP \cdot (W1 + W2 + W3 - X)$

$YE = W1 + W2 + W3 - X$

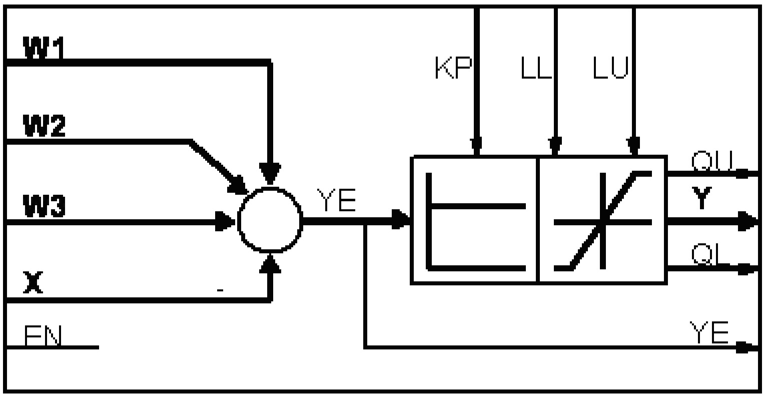
The system deviation YE is always calculated, irrespective of the operating mode, and is lead out separately.

The output Y of the controller can be limited via the inputs LU and LL. If the output Y reaches one of the two limits, a message is sent via the outputs QU and QL. If LL >= LU, then output Y = LU.

The controller is enabled with EN = 1. If EN = 0, the output variable Y is set to zero. The controller is disabled. The binary outputs QU and QL are treated in this case as if KP*YE equaled zero.

The controller operates inverted when a negative KP value is selected (inversion amplifier).

Block diagram



Truth table(s)

EN	Conditions	Y	QU	QL	Mode
0	LL< 0 <LU	0	0	0	Controller disable
0	LU <= 0	0	1	0	Controller disable
0	LL >= 0	0	0	1	Controller disable
1	LL<YE * KP<LU	KP × YE	0	0	Controller enable
1	YE * KP >= LU	LU	1	0	Controller at upper limit
1	YE * KP <= LL	LL	0	1	Controller at lower limit

Truth table for LL >= LU

EN	Conditions	Y	QU	QL	Mode
0	None	0	1	1	Controller disable
1	LL >= LU	LU	1	1	Controller at upper limit

Block connections

Block connection	Description	Preassignment	Value range	Attributes
W1	Setpoint 1	0.0	REAL	
W2	Setpoint 2	0.0	REAL	
W3	Setpoint 3	0.0	REAL	
X	Actual value	0.0	REAL	
LU	Upper limit	0.0	REAL	
LL	Lower limit	0.0	REAL	
KP	P-action coefficient	0.0	REAL	
EN	Controller enable	0	0/1	
Y	Output variable	0.0	REAL	
YE	System deviation	0.0	REAL	
QU	Controller at upper limit	1	0/1	
QL	Controller at lower limit	1	0/1	

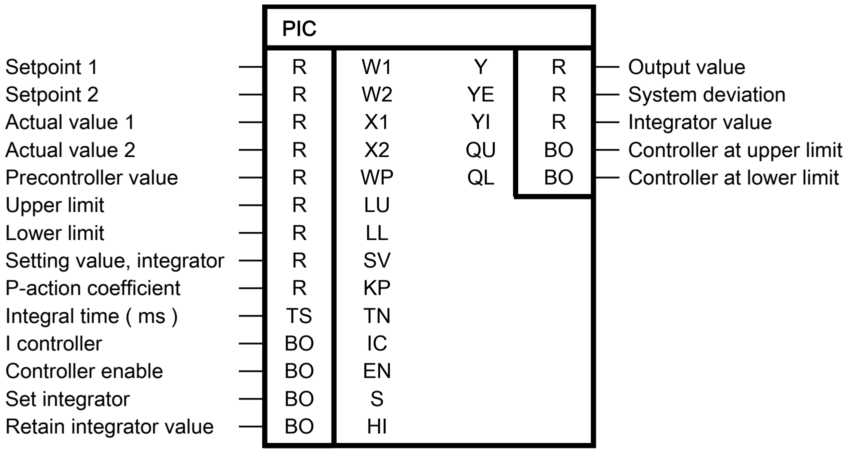
Project data

Can be loaded online	Yes
Special characteristics	-

8.10 PIC PI controller

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

- Universal PI controller, can be switched to P-controller or I-controller modes. Can be used as a speed controller or a primary controller. Suitable for dynamic override control
- Flexible integrator functions:
- Set initial value ⇒ Load SV in integrator
- Retain current value of integrator ⇒ P-controller
- Integrator control by SV
- Integrator control by controller limiting
- Gain shutdown ⇒ I-controller

Overall controller functions:

- Independent setting and modification of the following variables during operation:
- Proportional coefficient KP
- Integral action time TN
- Controller limits LU and LL
- Precontroller value WP, e.g. for acceleration injection
- Second actual value input X2, e.g. for static injection
- Indication when set limits are reached

Operation mode

The actual value total (X1+X2) is subtracted from the setpoint total (W1+W2) according to the equation:

YE = (W1 + W2) - (X1 + X2)

The result, system deviation YE, is then multiplied by the adjustable proportional coefficient KP. The product is carried to the output summation device and the integrator. The adjustable integral time TN determines the integration behavior of the controller. The change in output variable YI is proportional to input variable KP*YE and inversely proportional to the integral-action time TN. The integrator value YI is also given to the output summation device. Another value with the correct sign can be added to output value Y via input WP.

Discrete values are calculated according to the algorithm:

Algorithm:

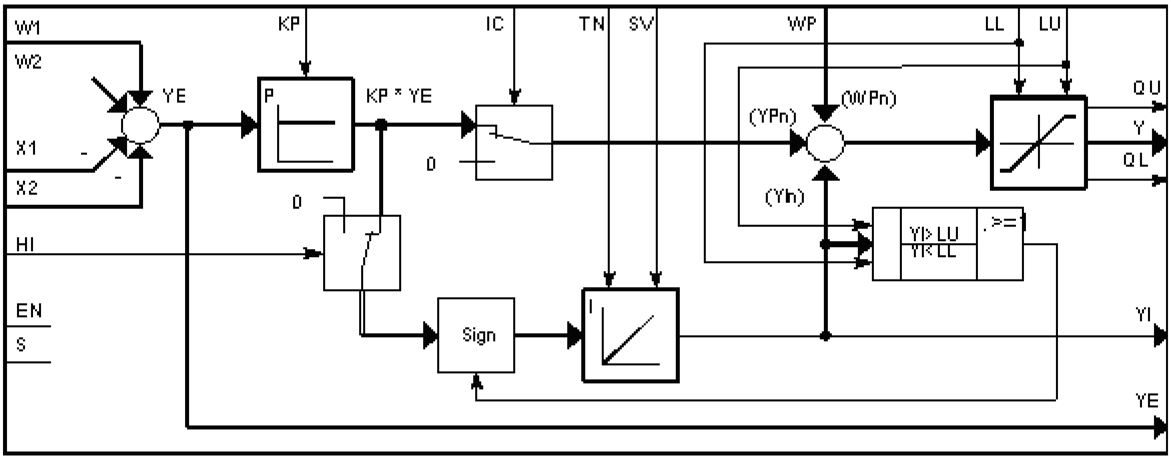
$$Y_n = Y_{n-1} + KP \cdot \left[\left(1 + \frac{TA}{TN} \right) \cdot YE_n - YE_{n-1} \right]$$

with boundary conditions: LL < Y < LU, and LL < LU

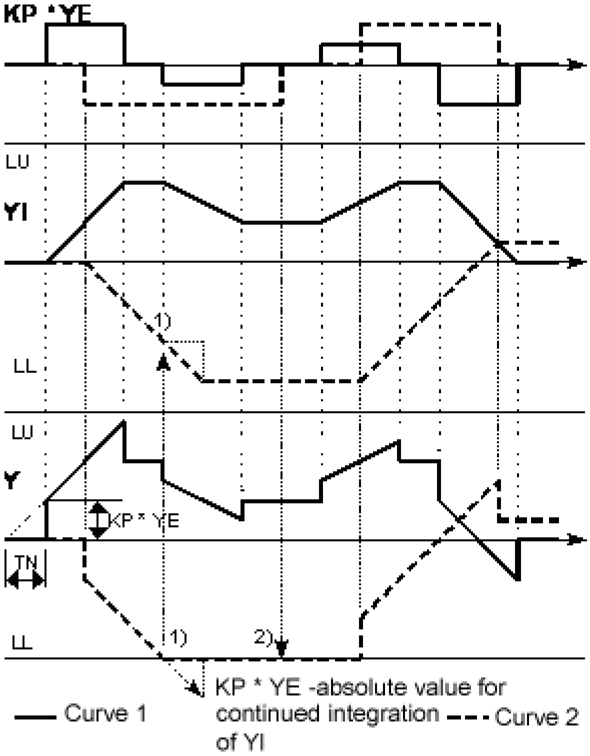
Y_n	Value of Y in scan interval n
Y_{n-1}	Value of Y in scan interval n-1

TA is the sampling time in which the block is configured.

Block diagram



XY diagram



Curves 1 and 2 show the characteristic of Y and YI during YE jumps:

- Curve 1, normal operation, no limiting
- Curve 2, with the use of limits (e.g. LL)

For 2) a decay of YE*KP is expected, but this is canceled by the continued integration in 1).

Operating modes and control of the controller

Output value Y and integrator value YI of the controller can be limited via the inputs LU and LL. When the set limits are reached by output variable Y, a message is issued with QU = 1 of QL = 1.

The following priority sequence applies for the control inputs:

EN before IC before S before HI.

Command input at the control inputs:

Control input	Value	Functions
EN	1	Controller enable
IC	1	Change-over from PI controller to I controller
S	1	Accept integrator setting value, do not integrate
HI	1	Retain integrator output YI, do not integrate

The combination of commands at the control inputs and the possible operating modes can be found in the truth tables.

In normal controller operation, LL <= 0 <= LU and LL < Y_n < LU. However, other settings, explained below, are possible. To this end, the algorithm is converted appropriately:

$Y_n = KP \cdot YE_n + YI_n + WP_n$

There are 5 different operating conditions in conjunction with LU and LL:

No.	Conditions	Y_n
LL < LU		
1	LL < KP*YE_n + YI_n + WP_n < LU	KP*YE_n+YI_n+WP_n
2	KP*YE_n + YI_n + WP_n >= LU	LU
3	KP*YE_n + Y_n + WP_n <= LL	LL
LL=LU		
4	None	LU
LL > LU		
5	None	LU

Integrator control by own limiting

If output Y reaches one of the set limits LL or LU during the control process, integrator YI may continue to run until it also reaches the limit and is held there.

If the controller is at the limit and the limit value is changed, output Y momentarily assumes the new value as long as an override is defined. However, the integrator is updated to the new limiting value at change velocity YIn.

Truth table(s)

Operating condition 1

EN	IC	S	HI	ΔYI_n	YI_n	Y_n	Mode	Comments
0	*	*	*	*	0	0	Controller disable	KP, RN, WP, LU, LL, YE any value

EN	IC	S	HI	ΔYI_n	YI_n	Y_n	Mode	Comments
1	0	0	0	KP*YE_n×TA/TN	YI_{n-1} +ΔYI n	KP*YE_n +YI_n +WP_n	PI controller	Controller enable, normal operation
1	1	0	0	KP*YE_n*TA/TN	YI_{n-1} +ΔYI n	YI_n +WP_n	I controller	P-action component = 0
1	0	1	*	*	SV_n	KP*YE_n +YI_n +WP_n	P-controller, integrator guidance	YI_n = SV_n
1	1	1	*	*	SV_n	YI_n +WP_n	I controller, integrator guidance	YI_n = SV_n
1	0	0	1	0	YI_{n-1}	KP*YE_n +YI_n +WP	P-controller, integrator = constant	YI_n = YI_{n-1}
1	1	0	1	0	YI_{n-1}	YI_n +WP_n	I-controller, integrator = constant	YI_n = YI_{n-1}

*= any value

Operating condition 2

EN	IC	S	HI	ΔYI_n	YI_n	Y_n	Mode	Comments
1	0	0	0	KP*YE_n×TA/TN	YI_{n-1}+ΔYI_n for YI_{n-1} < LUYI_{n-1}- ΔYI_n for YI_{n-1}> LULU for YI_{n-1} = LU	LU	PI controller at upper limit	YI_n integrated -> LU, possibly with (-)
1	1	0	0	KP*YE_n×TA/TN	YI_{n-1}+ΔYI_n for YI_{n-1} < LUYI_{n-1}- ΔYI_n for YI_{n-1} > LULU for YI_{n-1} = LU	LU	I controller at upper limit	YI_n integrated -> LU, possibly with (-)
1	0	1	*	*	SV_n for SV_n < LULU for SV_n >= LU	LU	P controller at upper limit	YI_n = SV_n or YI_n = LU
1	1	1	*	*	SV_n for SV_n < LULU for SV_n >= LU	LU	I controller at upper limit	YI_n = SV_n or YI_n = LU, P-action component = 0
1	0	0	1	0	YI_{n-1}	LU	P-controller, integrator = constant	YI_n = YI_{n-1} or YI_{n-1} = LU
1	1	0	1	0	YI_{n-1}	LU	I-controller, integrator = constant	YI_n = YI_{n-1} or YI_{n-1} = LU, P-action component = 0

*= any value

Operating condition 3

EN	IC	S	HI	ΔYI_n	YI_n	Y_n	Mode	Comments
1	0	0	0	KP*YE_n*TA/TN	YI_{n-1}+ΔYI_n for YI_{n-1} < LLYI_{n-1}- ΔYI_n for YI_{n-1} > LLLL for YI_{n-1} = LL	LL	PI controller at lower limit	YI_n integrated -> LL, possibly with (-)
1	1	0	0	KP*YE_n×TA/TN	YI_{n-1}+ΔYI_n for YI_{n-1} < LLYI_{n-1}- ΔYI_n for YI_{n-1} > LLLL for YI_{n-1} = LL	LL	I controller at lower limit	YI_n integrated -> LL, possibly with (-)
1	0	1	*	*	SV_n for SV_n > LLLL for SV_n <= LL	LL	P controller at lower limit	YI_n = SV_n or YI_n = LL
1	1	1	*	*	SV_n for SV_n > LLLL for SV_n <= LL	LL	I controller at lower limit	YI_n = SV_n or YI_n = LL, P-action component = 0
1	0	0	1	0	YI_{n-1}	LL	P-controller, integrator = constant	YI_n = YI_{n-1} or YI_{n-1} = LL
1	1	0	1	0	YI_{n-1}	LL	I-controller, integrator = constant	YI_n = YI_{n-1} or YI_{n-1} = LL, P-action component = 0

*= any value

Operating condition 4

EN	IC	S	HI	ΔYI_n	YI_n	Y_n	Mode	Comments
1	*	*	*	*	*	LL=LU	See operating conditions 2 or 3	-

*= any value

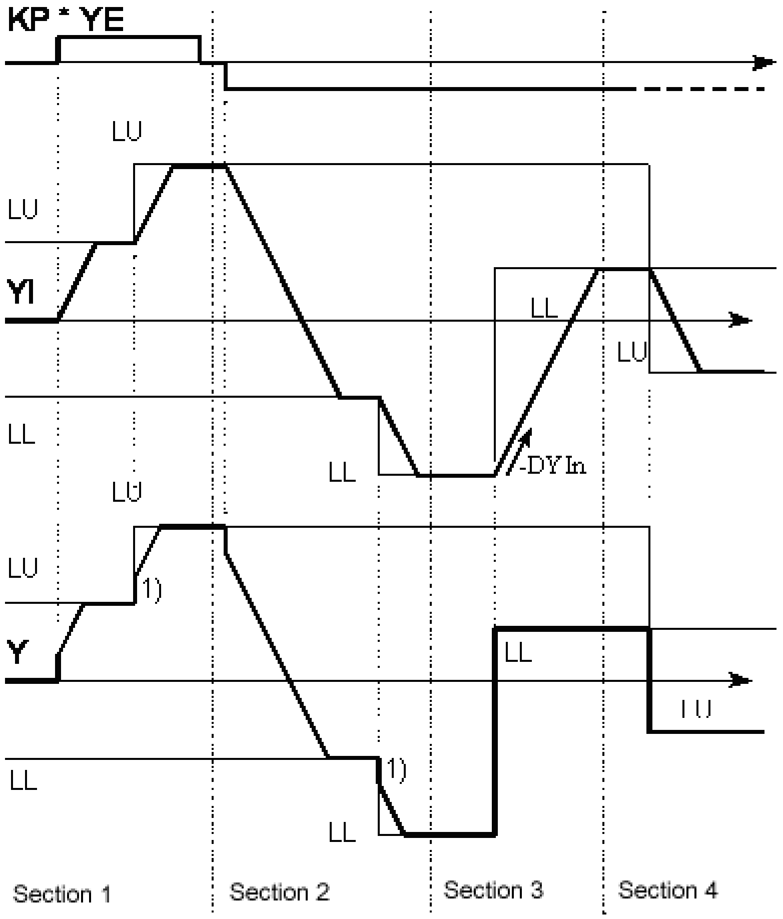
Operating condition 5

EN	IC	S	HI	ΔYI_n	YI_n	Y_n	Mode	Comments
1	*	*	*	KP*YE_nTA/TN	YI_{n-1}+ΔYI_n for YI_{n-1} < LUYI_{n-1}- ΔYI_n for YI_{n-1} > LULU for YI_{n-1} = LU	LU	PI controller at upper limit	-

Depending on the direction of the limit value change, the sign of the integration is inverted if necessary.

Transfer functions

Transfer function during controller override for conditions 2, 3 and 5:



Section 1: Characteristic with LUn > LUn-1 according to operating condition 2

Section 2: Characteristic with LLn < LLn-1 according to operating condition 3

Section 3: Characteristic with LLn > LLn-1 according to operating condition 3, for

Limit shift relative to the control direction with

sign inversion at the integrator input

Section 4: Characteristic with LLn > LUn according to operating condition 5

1) Jump by KP*YE, because the integrator was run up to the limit.

Change-over from PI mode to I mode

When EN = 1 and IC = 1, the P-action component is retained at 0, and the controller is switched from PI mode to I mode. Output Y assumes integrator value YI. If this occurs during the control process, then a jump by -KP * YE will occur at output Y. When resetting to IC = 0, the P component is reset to the current value of KP * YE. The controller again exhibits PI behavior. If this occurs during the control process, then a jump by KP * YE will occur at output Y.

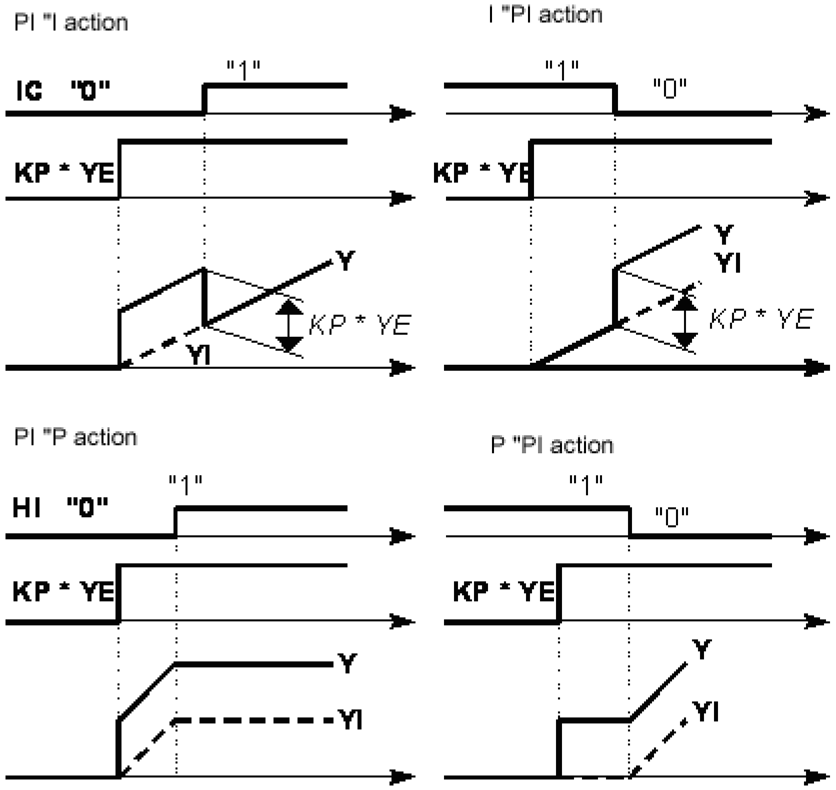
Change-over from PI mode to P mode

If the block inputs become EN = 1 and HI = 1, integrator YI is held and the controller is switched smoothly from PI to P mode. YI continues to act as an addend on output Y.

During a reset to HI = 0, the integrator is enabled again. The controller again exhibits PI behavior.

Transfer functions

Transfer functions during changeover without controller override: Examples for EN=1∧S=0



PI "I action

I "PI action

PI "P action

P "PI action

Block connections

Block connection	Description	Preassignment	Value range	Attributes
W1	Setpoint 1	0.0	REAL	
W2	Setpoint 2	0.0	REAL	
X1	Actual value 1	0.0	REAL	
X2	Actual value 2	0.0	REAL	
WP	Precontroller value	0.0	REAL	
LU	Upper limit	0.0	REAL	
LL	Lower limit	0.0	REAL	
SV	Setting value, integrator	0.0	REAL	
KP	P-action coefficient	0.0	REAL	
TN	Integral time (ms)	0.0	SDTIME	
IC	I controller	0	0/1	
EN	Controller enable	0	0/1	
S	Set integrator	0	0/1	
HI	Retain integrator value	0	0/1	
Y	Output value	0.0	REAL	
YE	System deviation	0.0	REAL	
YI	Integrator value	0.0	REAL	
QU	Controller at upper limit	1	0/1	
QL	Controller at lower limit	1	0/1	

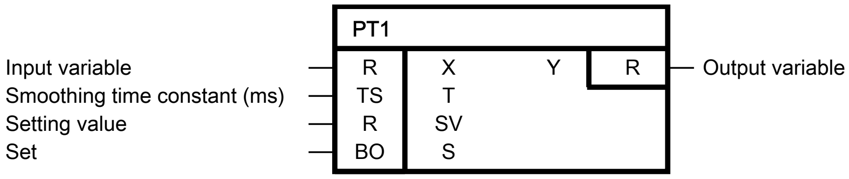
Project data

Can be loaded online	Yes
Special characteristics	-

8.11 PT1 Delay element

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

- First-order delay element with setting function
- Use as smoothing element

Operation mode

Setting function not active (S = 0)

Input variable X, dynamically delayed by smoothing time constant T, is given to output Y.

T determines the steepness of the rise of the output variable. It specifies the time at which the transfer function has risen to 63% of its end value.

After t = 3T, the transfer function reaches approx. 95% of its end value.

The internally fixed proportional gain is 1 and does not vary.

If T/TA (T/TA > 10) is sufficiently large, the transfer function corresponds to the characteristic of

$Y(t) = X\cdot(1-e^{-t/T})$

with t = n * TA.

Discrete values are calculated according to the algorithm

Algorithm:

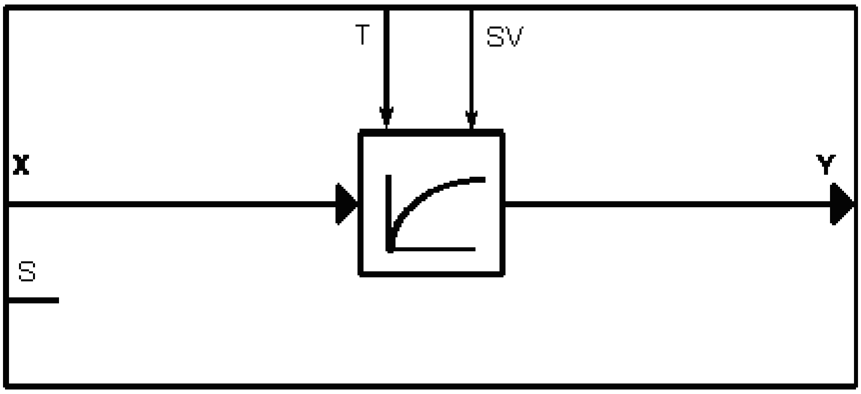
$$Y_n = Y_{n-1} + \frac{TA}{T} \cdot \left(X_n - Y_{n-1}\right)$$

Yn	Value of Y in scan interval n
Yn-1	Value of Y in scan interval n-1
Xn	Value of X in scan interval n

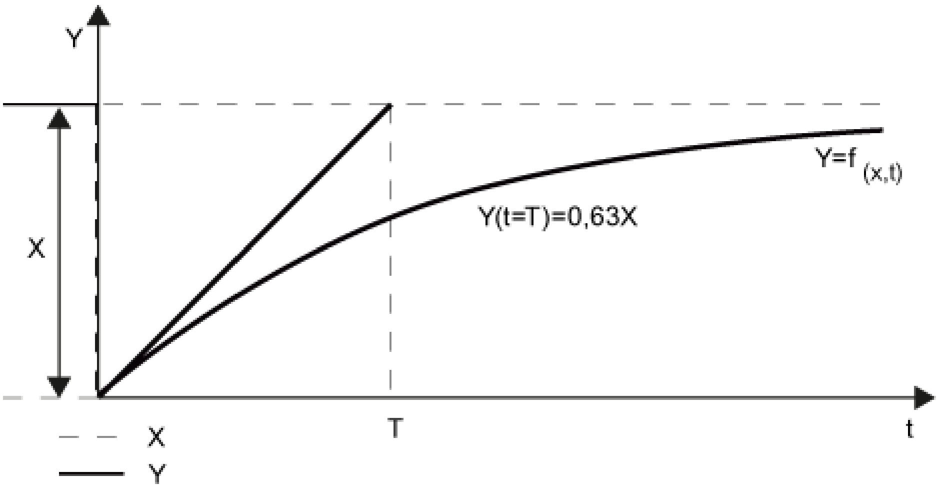
Setting function active (S = 1)

When the setting function is active, the current setting value SVn is accepted at the output variable: Yn = SVn

Block diagram



Time diagram



Initialization

If input S is logic 1 at the initialization, the setting value SV is applied at output Y.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
T	Smoothing time constant (ms)	0.0	SDTIME	
SV	Setting value	0.0	REAL	
S	Set	0	0/1	
Y	Output variable	0.0	REAL	

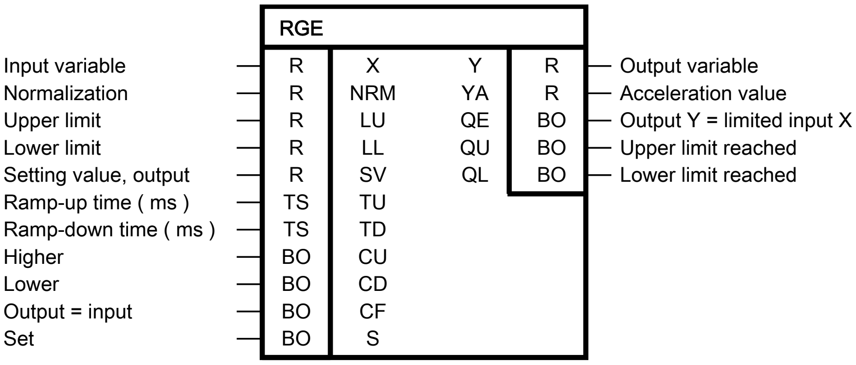
Project data

Can be loaded online	Yes
Special characteristics	-

8.12 RGE Ramp-function generator

☑ SIMOTION ☑ SINAMICS

Symbol



Short description

- Ramp-function generator for limiting the change velocity of input variable X
- Output variable can be limited:
- Independent setting and modification of the following variables during operation:
- Ramp-up and ramp-down time
- Output limits LU and LL
- Setting value

Flexible ramp-function generator functions:

- Integrating correction to setpoint X
- Setting of initial value for ramp-function generator output (-> load SV to integrator)
- Integrating increase and decrease of ramp-function generator output

Operation mode

The block contains an integrator with two integration time constants that can be set separately. Output Y changes according to the algorithm:

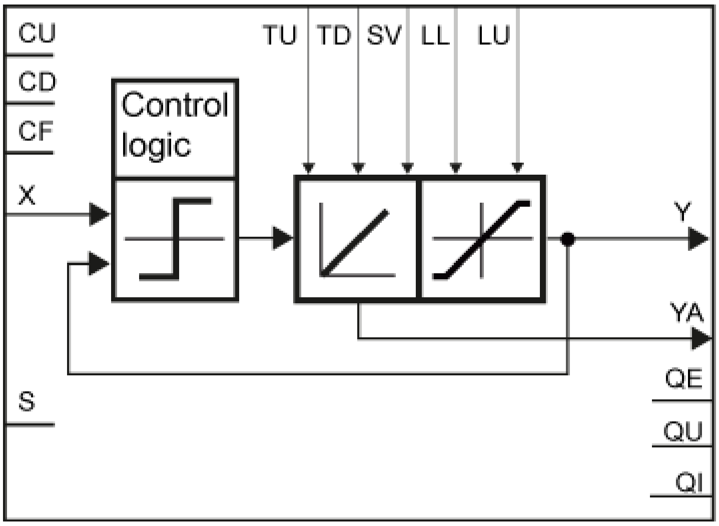
$Y_n = Y_{n-1}+YA_n$

The acceleration value YA is calculated separately for the ramp up and ramp down and is output on an output.

The process in which output value Y moves away from zero is called ramp up.

The process in which output value Y moves toward zero is called ramp down.

Block diagram



The following applies for the ramp up acceleration value:

$$YA = \frac{TA}{TU} * NRM\ \mathit{für}\,Y > 0$$

$$YA = -\frac{TA}{TU} * NRM\ \mathit{für}\,Y < 0$$

The following applies for the ramp down acceleration value:

$$YA = -\frac{TA}{TD} * NRM\ \mathit{für}\,Y > 0$$

$$YA = \frac{TA}{TD} * NRM\ \mathit{für}\,Y < 0$$

The change-over between ramp up time and ramp down time takes place during a direction change or at the zero crossover of the transfer function.

The operating mode is predefined by means of control logic, depending on the logic states of the control inputs S, CF, CU, and CD.

The output variable can be limited by means of the inputs LU and LL. When the set limits are reached by Y, the binary outputs QU or QL are set to 1. The binary output QE becomes 1 when Y = X.

Ramp-up time and ramp-down time

The ramp-up time TU is the time in which the absolute value of the output variable increases by NRM.

The ramp-down time TD is the time in which the absolute value of the output variable is reduced by NRM. Ramp-up time and ramp-down time can be selected differently.

The smaller TA/TU or TA/TD is, the smaller is the amplitude change on Y from one sampling time to the next..TA is the sampling time with which the block is processed.

The following priority sequence applies for the control inputs:

S before CF before CU and CD.

Function of control inputs:

S=1	Load setting value SV in integrator; do not integrate.
CF=1	Correct output Y to setpoint X with integration.
CU=1	Correct output Y in the direction LU with integration
CD=1	Correct output Y in the direction LL with integration

Operating modes and control of the ramp-function generator

The combination of commands at the control inputs and the possible operating modes can be found in the truth tables.

In normal ramp-up mode, LL <= 0 <= LU and LL< Yn < LU. However, other settings, explained below, are possible.

The following applies to the setting LL >= LU: The LU limit is dominant over the LL limit.

Behavior of the integrator at the limitation

If output Y reaches one of the set limits LL or LU during the control process, the integrator value is held. The output value Y is then kept constant until the integrator value leaves the limit due to changes in the input variables.

If the integrator is at the limit and the limit value is changed, the integrator behaves differently depending on the direction of the limit value change.

If the absolute value of a limit is increased, and the control logic specifies that the ramp-function generator is to run in the same direction, the integrator continues to integrate from the previously held value in accordance with the set ramp-up time, until the output reaches the limit again.

If the absolute value of the limit value is reduced, the integrator integrates from the previously held value according to the set ramp down time, until the output again reaches the limit value.

TD and TD are limited internally: TU >= TA, TD >= TA

Truth table(s)

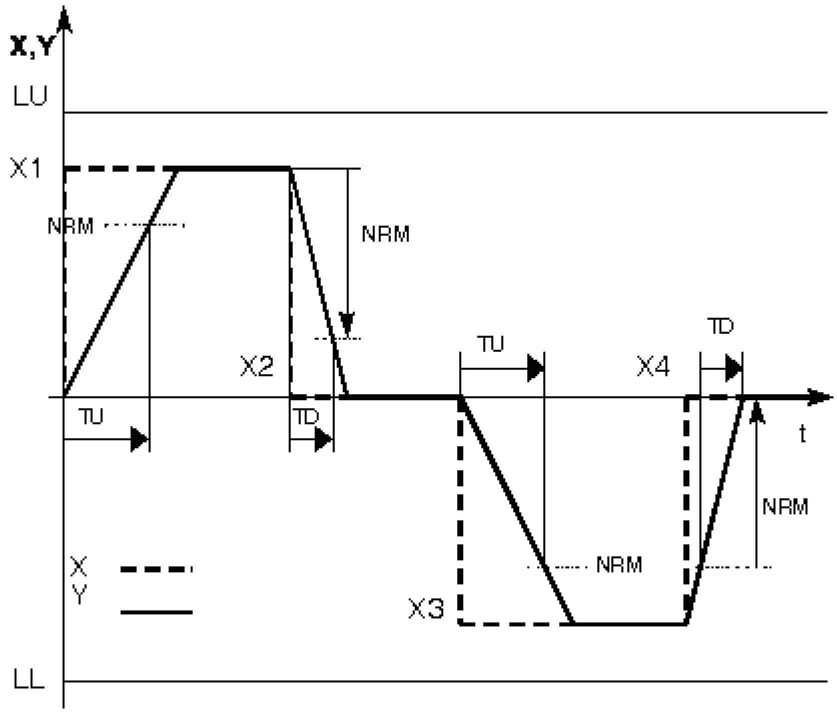
S	CF	CU	CD	Y_{An}	Yn	Mode	Comments
0	0	0	0	0	Y_{n-1}	Stop	Y is constant

LL < LU and LL < actual value Y_{n-1} < LU

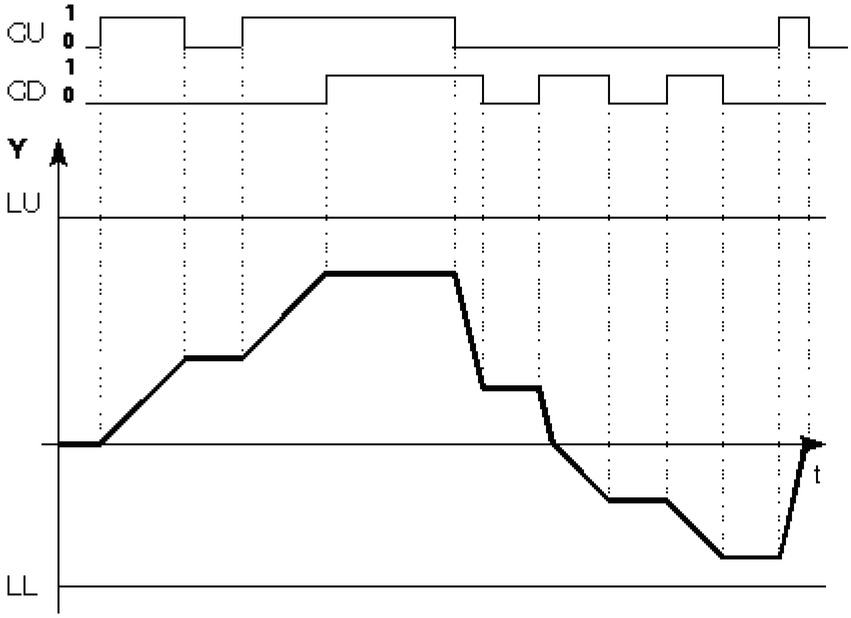
S	CF	CU	CD	YAn	Yn	Mode	Comments
1	*	*	*	Jump	SV_n	Set output to SV	Any SV, fixed or variable
0	1	*	*	TA/ TU;TA/TD	Y_{n-1} +YA n	Normal mode Y -> X	TU for [X > Y ∧ Y >= 0] ∨ [X < Y ∧ Y <= 0]TD for [X > Y ∧ Y < 0] ∨ [X < Y ∧ Y > 0]
0	0	1	0	TA/ TU(TA/ TD)	Y_{n-1} +YA n	Touch upperlimit value Y -> LU	TU, TD as before, depending on start position
0	0	0	1	TA/ TD(TA/ TU)	Y_{n-1} +YA n	Touch lowerlimit value Y -> LL	TU, TD as before, depending on start position

* Arbitrary

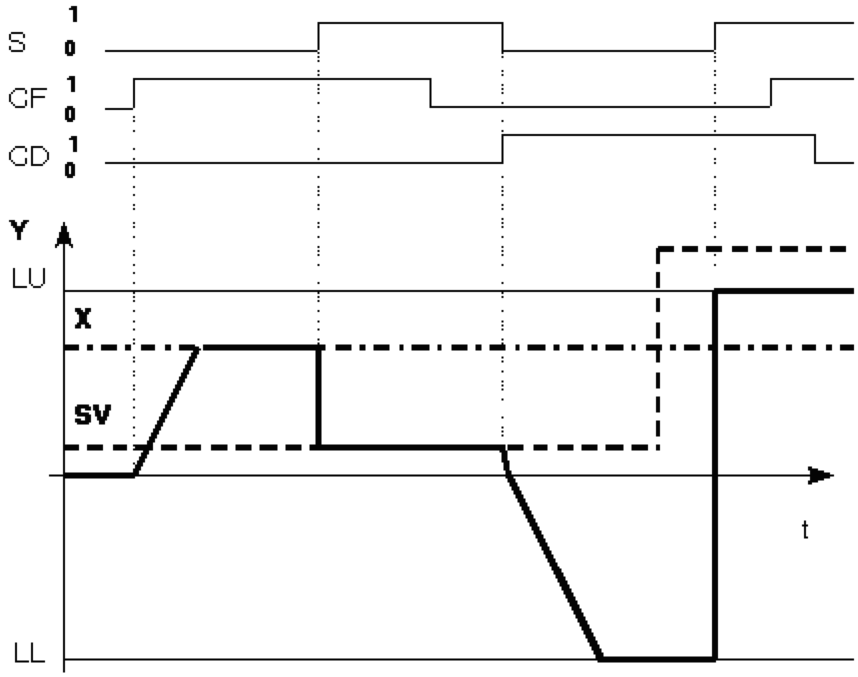
Transfer function



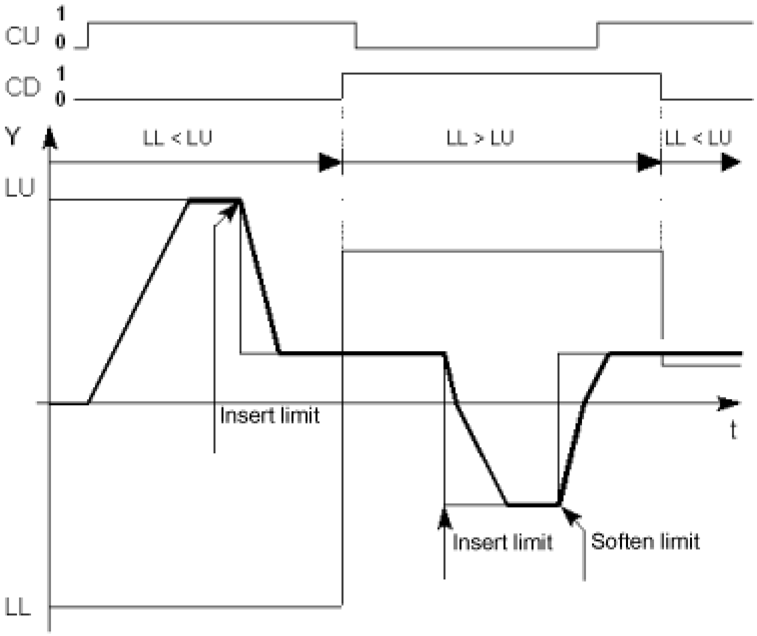
Example 1: CF = 1 with LL < LU and LL < X < LU, as well as X_1= 1.5, X_2 = X_4 = 0.0, X_3 = –1.5, LU = 2.0, LL = –2.0, TU > TD, NRM > 0



Example 2: Motor potentiometer function with CU and CD and with LL < LU



Example 3: Set integrator with LL < LU



Example 4: Change and interchange of the limits

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
NRM	Normalization	1.0	REAL	
LU	Upper limit	0.0	REAL	
LL	Lower limit	0.0	REAL	
SV	Setting value, output	0.0	REAL	
TU	Ramp-up time (ms)	0.0	SDTIME	
TD	Ramp-down time (ms)	0.0	SDTIME	
CU	Higher	0	0/1	
CD	Lower	0	0/1	
CF	Output = input	0	0/1	
S	Set	0	0/1	
Y	Output variable	0.0	REAL	
YA	Acceleration value	0.0	REAL	
QE	Output Y = limited input X	0	0/1	
QU	Upper limit reached	0	0/1	
QL	Lower limit reached	0	0/1	

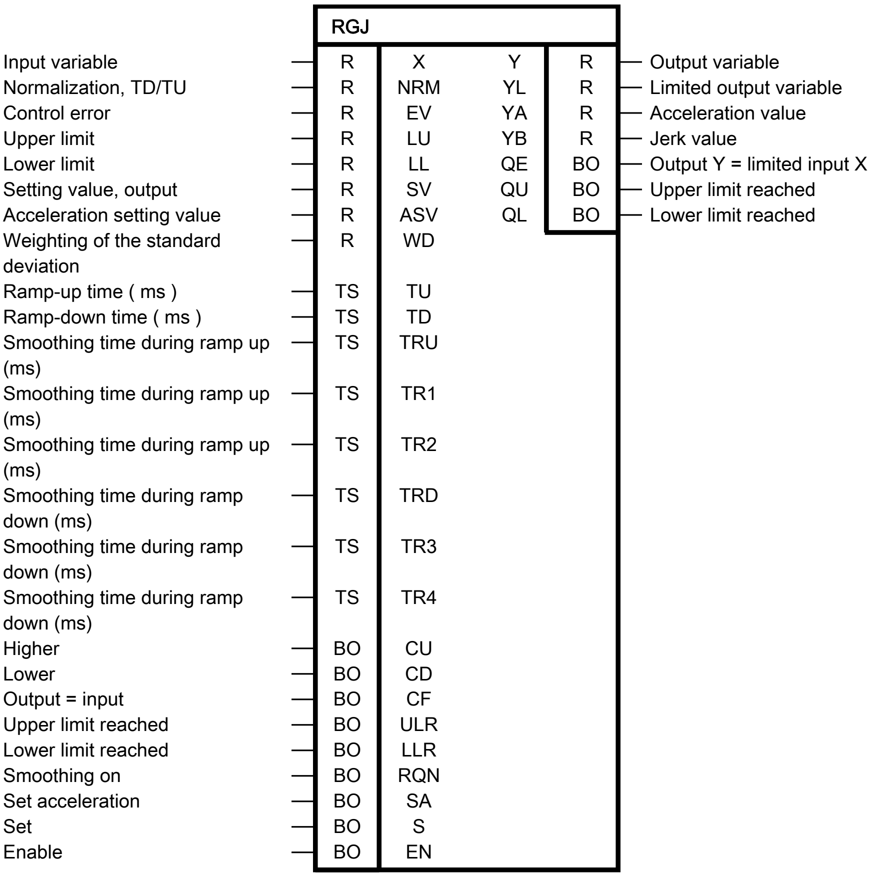
Project data

Can be loaded online	Yes
Special characteristics	-

8.13 RGJ Ramp-function generator with jerk limiting

☑ SIMOTION ☑ SINAMICS

Symbol


RGJ
Input variable — R X
Normalization, TD/TU — R NRM
Control error — R EV
Upper limit — R LU
Lower limit — R LL
Setting value, output — R SV
Acceleration setting value — R ASV
Weighting of the standard deviation — R WD
Ramp-up time (ms) — TS TU
Ramp-down time (ms) — TS TD
Smoothing time during ramp up (ms) — TS TRU
Smoothing time during ramp up (ms) — TS TR1
Smoothing time during ramp up (ms) — TS TR2
Smoothing time during ramp down (ms) — TS TRD
Smoothing time during ramp down (ms) — TS TR3
Smoothing time during ramp down (ms) — TS TR4
Higher — BO CU
Lower — BO CD
Output = input — BO CF
Upper limit reached — BO ULR
Lower limit reached — BO LLR
Smoothing on — BO RQN
Set acceleration — BO SA
Set — BO S
Enable — BO EN
Y R — Output variable
YL R — Limited output variable
YA R — Acceleration value
YB R — Jerk value
QE BO — Output Y = limited input X
QU BO — Upper limit reached
QL BO — Lower limit reached

Short description

- Ramp-function generator with jerk limiting and correction
- Ramp-function generator functions:
- Set output Y or acceleration YA
- Correct ramp-function generator output to setpoint X with integration and jerk limiting
- Integrating increase and decrease of ramp-function generator output
- Correction of the ramp-function generator according to the system deviation of a lower-level controller during limiting

Operation mode

The block limits the acceleration (change in velocity) and the jerk (change in acceleration) of setpoints.

The following algorithms apply:

$$Y_n = \bar{Y}_{n-1} + \bar{Y}A_n$$
$$YA_n = YA_{n-1} + YB_n$$

Acceleration value YA and jerk YB are calculated separately for ramp up and ramp down. This requires configuration of the time values ramp up time TU and smoothing time during ramp up TRU as well as ramp down TD and smoothing time during ramp down TRD.

The following applies for the acceleration value YA outside the smoothing time during ramp-up:

$$YA = YA_{max} = \frac{TA}{TU} * NRM \quad \text{for Y > 0}$$

$$YA = YA_{max} = -\frac{TA}{TU} * NRM \quad \text{for Y < 0}$$

The following applies for the acceleration value YA outside the smoothing time during ramp-down:

$$YA = YA_{max} = -\frac{TA}{TD} * NRM \quad \text{for Y > 0}$$

$$YA = YA_{max} = \frac{TA}{TD} * NRM \quad \text{for Y < 0}$$

The following applies for the jerk value YB during the smoothing time during ramp up:

$$YB = \frac{TA \cdot YA_{max}}{TRU}$$

or

$$YB = \frac{TA \cdot YA_{max}}{TR1} \qquad YB = \frac{TA \cdot YA_{max}}{TR2}$$

The following applies for the jerk value YB during the smoothing time during ramp down:

$$YB = \frac{TA \cdot YA_{max}}{TRD}$$

or

$$YB = \frac{TA \cdot YA_{max}}{TR3} \qquad YB = \frac{TA \cdot YA_{max}}{TR4}$$

The operating mode is predefined by means of control logic, depending on the logic states of the binary variables EN, S, SA, CF, CU, and CD.

Input variable X and thus indirectly output variable Y are limited by means of the block inputs LU and LL. When the set limits are reached by Y, a message is issued to the binary outputs with QU = 1 or QL = 1.

Binary output QE becomes 1 when output variable Y equals the limited value of input variable X.

A ramp up process is subdivided into three phases:

Phase 1

If the setpoint X is increased, the max. jerk YB (depending on TRU or TR1) is specified in the first section. Thus, the acceleration increases proportionally over time; in this smoothing phase, output Y rises quadratically over time.

Phase 2

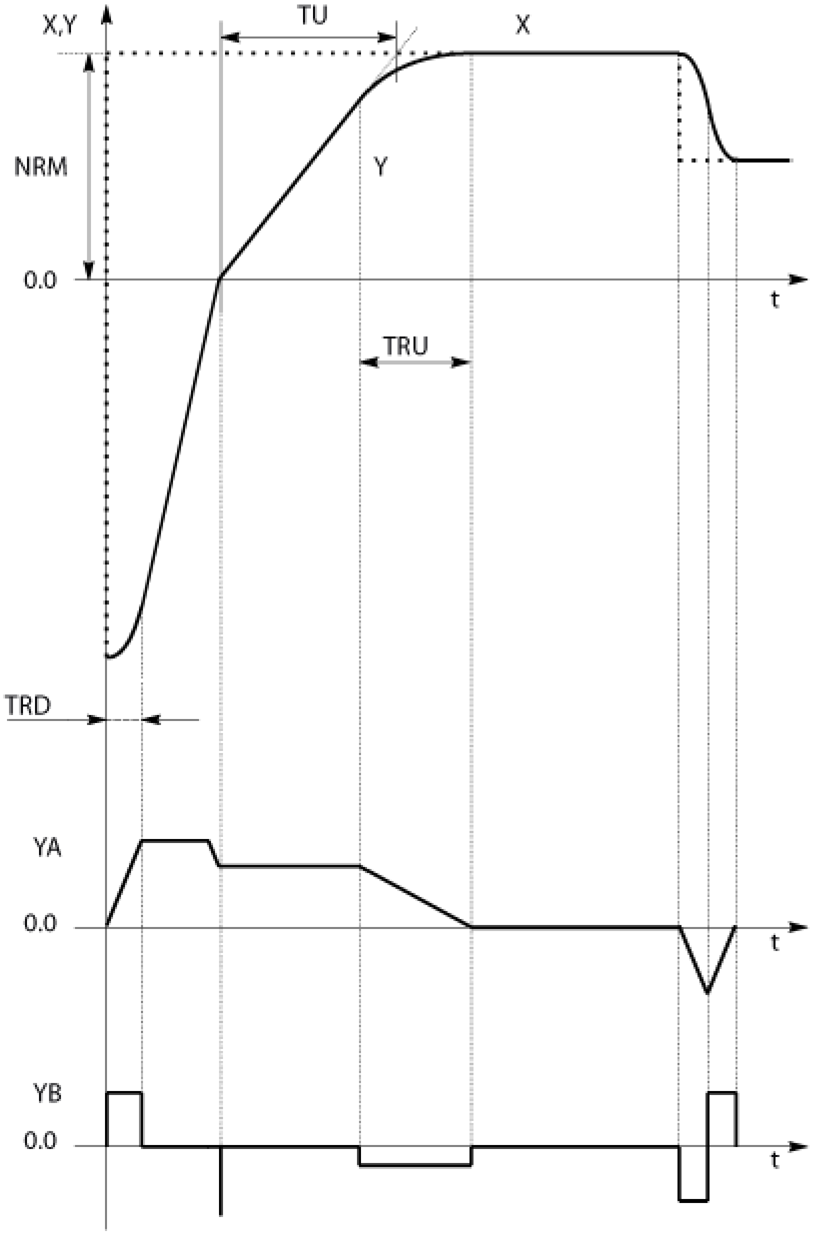
Once the maximum acceleration YA has been reached according to the defined ramp-up time TU, the acceleration is constant. Output variable Y rises proportionally over time.

Phase 3

In the third part, the acceleration is decreased proportionally over time. In this smoothing phase, output variable Y approaches input variable X on YB quadratically over time (depending on TRU or TR2).

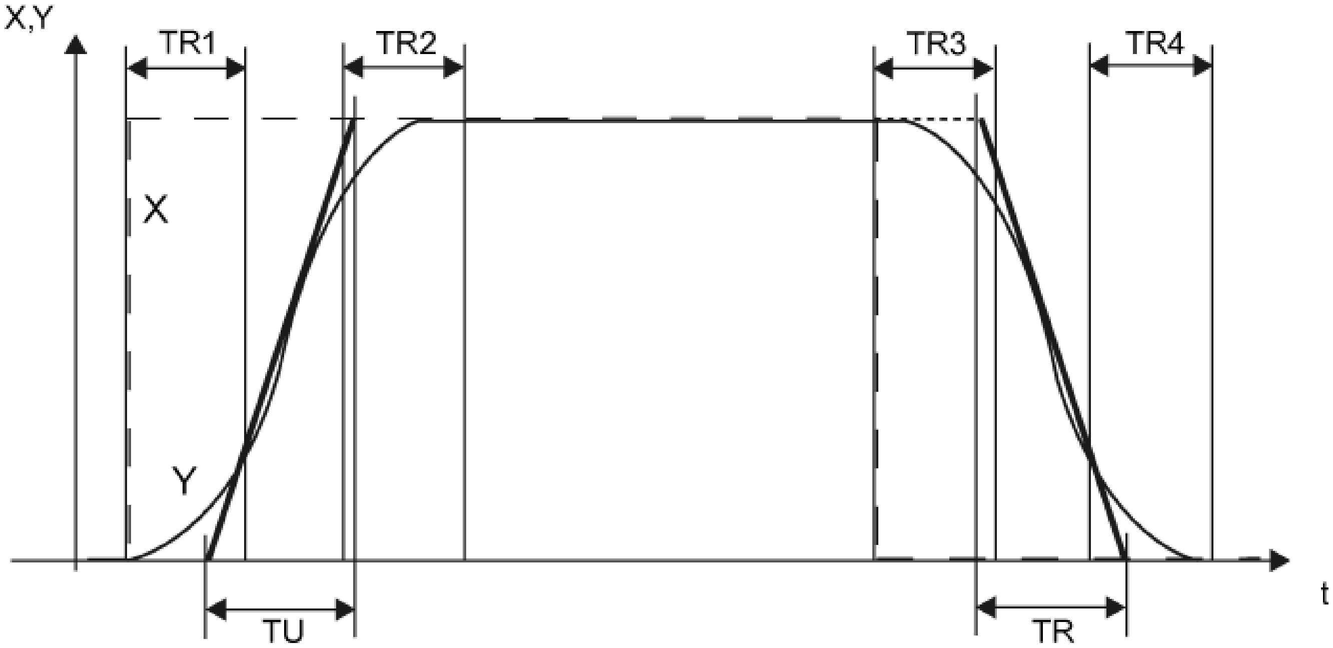
A ramp-down process proceeds analogously.

Transfer function



Ramp-up and ramp-down (not true to scale)

Rounding times if TRU=00.0 ms and TRD=0.0 ms:



Ramp-up and ramp-down (not true to scale)

Ramp-up time and ramp-down time

The ramp-up time TU is defined as the time in which the value of the output variable increases proportionally over time by the value NRM.

The ramp-down time TD is defined as the time in which the value of the output variable decreases proportionally over time by the value NRM.

Ramp up time and ramp down time can be selected differently.

Smoothing time during ramp up and ramp down

The smoothing time is defined as the time in which the output variable reaches the maximum acceleration value starting from a constant initial value. During this time, the jerk value is constant and not equal to zero (refer to phase 1).

The smoothing time is also defined as the time in which the output variable reaches a constant end value based on its maximum acceleration value (refer to phase 3). The smoothing time during a ramp-up is specified with TRU or TR1 and TR2, during a ramp-down with TRD or TR3 and TR4.

Each time the setpoint changes direction, the system switches from ramp-up to ramp-down or from ramp-down to ramp-up with the associated smoothing processes, depending on the initial position. The same applies accordingly when the ramp-up or ramp-down time is changes during operation.

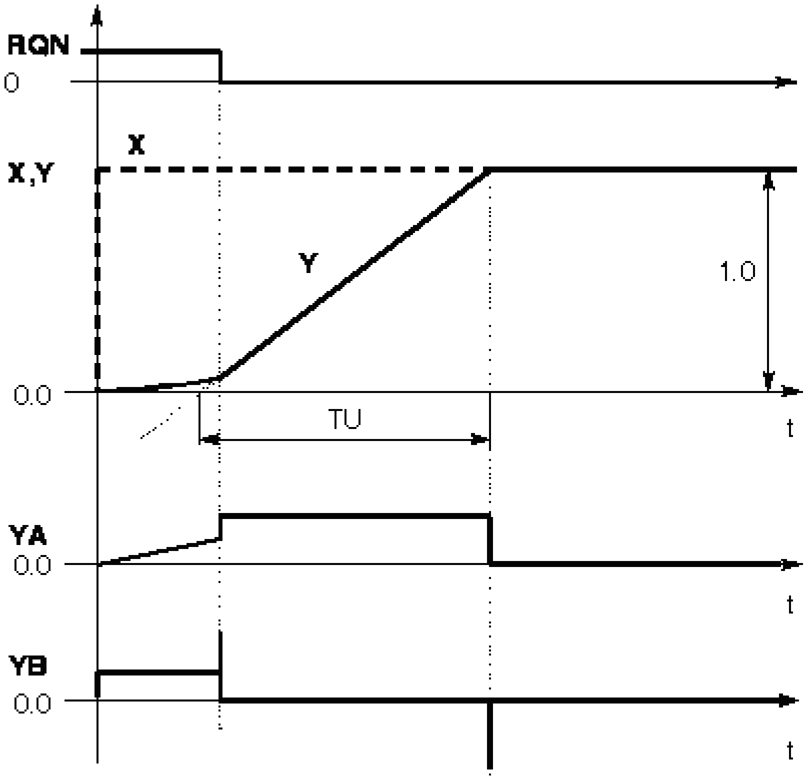
If a ramp-up follows a ramp-down, whereby TRD and TD are small and TRU and TU are large, YA is already reduced during the ramp-down so that an overshoot does not occur during the following ramp-up, as long as the target value (X, LL or LU) and the ramp-function generator times (TU, TD, TRU, TRD) do not change.

If the smoothing (RQN=0) and the correction (ULR=LLR=0) are switched off, then the RGJ behaves the same as the RGE block.

Enabling smoothing (jerk limitation)

Smoothing is active during ramp up and ramp down when RQN=1.

Transfer function: Deactivation of the smoothing during ramp-up



Smoothing is off when RQN = 0. Ramp up/down takes place according to the ramp-up/ramp-down time specified in TU or TD.

When the jerk limiting is switched off during the smoothing time, the remaining ramp up/down also occurs with the ramp-up/ramp-down time specified in TU or TD.

"Smoothing off" mode

If you want to operate the block in this mode, proceed as follows:

- Set the connections TRU, TR1, TR2, TRD, TR3, and TR4 to "0" (all smoothing times are "0").
- Set the RQN connection to "1" ("Smoothing on" mode).

With these settings, the RGJ block behaves as described in the "Smoothing off" mode (RQN = 0).

Operating modes and control of the ramp-function generator

The control inputs are defined as follows:

EN=1	Enable ramp-function generator
S=1	Set output Y to setting value SV; do not integrate
SA=1	Set acceleration YA to setting value ASV; do not integrate
CF=1	Correct output Y to setpoint X with integration.
CU=1	Correct output Y in the direction LU with integration
CD=1	Correct output Y in the direction LL with integration

Truth table(s)

EN	S	SF	CF	CU	CD	Y_{An}	Yn	Mode	Comments
0	*	*	*	*	*	0	0	Inhibit	Y=0
1	0	0	0	0	0	0	Y_{n-1}	Inhibit	Y = constant

*= any value

LL < LU and LL < actual value Y_{n-1} < LU

EN	S	SF	CF	CU	CD	YAn	Yn	Mode	Comments
1	1	*	*	*	*	Jump	SV_n	Set output toSV	Any SV, fixed or variable
1	0	1	*	*	*	ASV_n	Y_{n-1} $+YA_n$	Set outputto integrator 1on ASV	Any ASV, fixed or variable
1	0	0	1	*	*	TA/ TU(TA/TD)	Y_{n-1} $+YA_n$	Normal modeY->X	TU for [X>Y ∧ Y ≥ 0] v [X<Y ∧ Y<=0] TD for [X>Y ∧ Y<0] v [X<Y ∧ Y>0]QE=1 is set when Y=X is reached.
1	0	0	0	1	0	TA/ TU(TA/TD)	Y_{n-1} $+YA_n$	Touchupperlimit value Y -> LU	TU, TD as above, depending on start positionQU=1 and QE=1 are set when Y=LU is reached.
1	0	0	0	0	1	TA/ TD(TA/TU)	Y_{n-1} $+YA_n$	Touchupperlimit valueY -> LL	TU, TD as above, depending on start positionQL=1 and QE=1 are set when Y=LL is reached.

Correction of the ramp-function generator

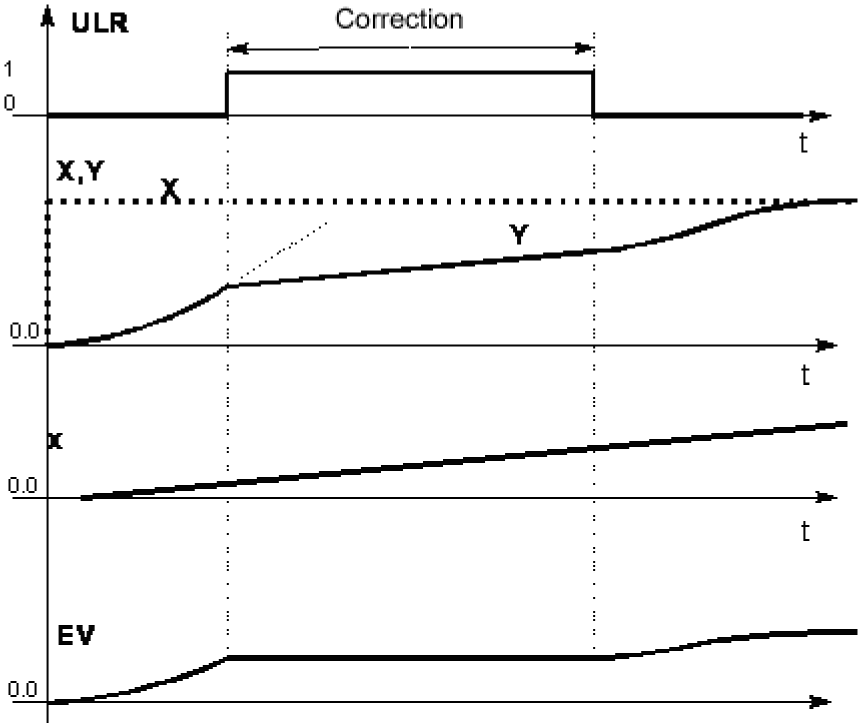
Generally, output Y of the ramp-function generator is led as setpoint to a subordinate control loop (e.g. speed controller).

If this controller reaches the limit because of a change (e.g. during a ramp-up), the ramp-function generator may not increase the output in accordance to the ramp-up times. In this case, output Y is corrected using the system deviation EV and the weighting factor WD:

$Y_n = Y_{n-1}-EV_n+WD\cdot EV_k$

$_n$= scan interval n

$_K$= the time at which the controller first reaches the limit (0 -> 1 edge on ULR or LLR)



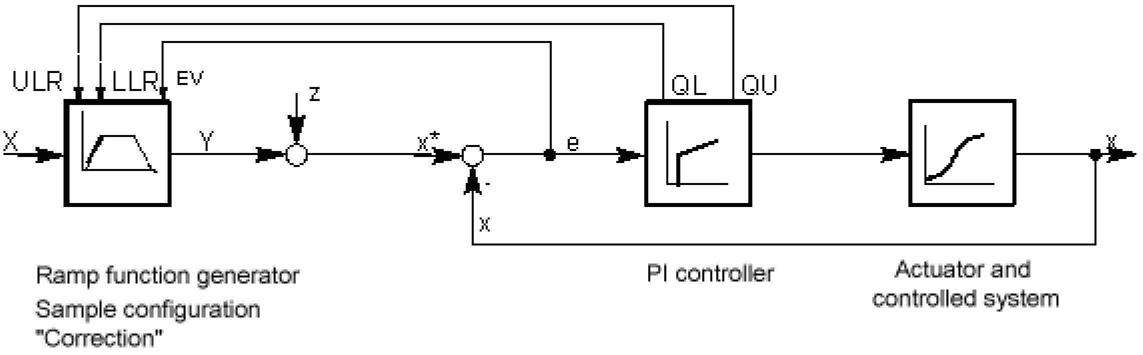
Example of correction mode

Generally, the correction can only be used for "conventional control loops" (e.g. PI speed controller). The controller limits must be set correctly (e.g. the same as the current limits).

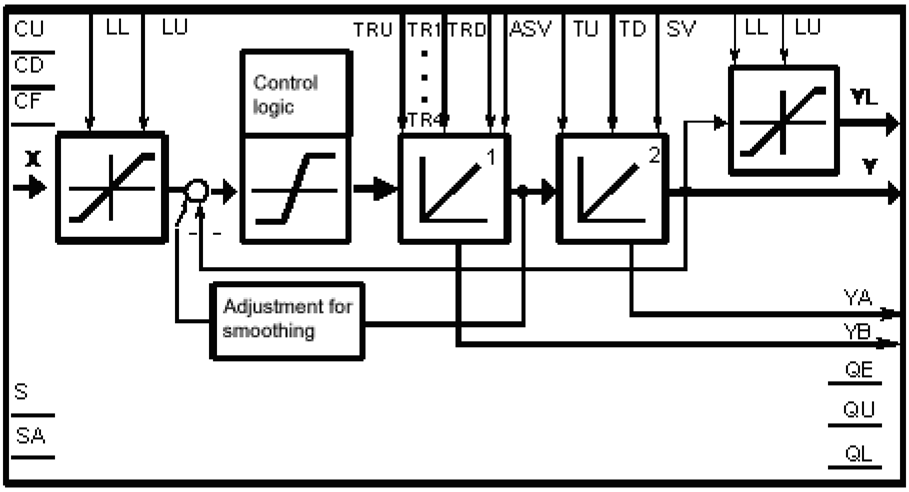
Generally, WD is 1.01 to 1.1 (> 1.0!). Jerk limiting is not active during the correction.

The binary outputs of the controller ("Upper/lower limit reached") are returned to the binary inputs ULR or LLR. When the limit is reached, one of the two binary inputs ULR = 1 or LLR = 1 is set via the feedback on the RGJ block, and therefore the correction activated.

If the correction is not to be used, ULR and LLR must be set to 0.



Block diagram



The input value NRM is set internally to 1.0 when NRM < 1.0e-18.

Block connections

Block connection	Description	Preassignment	Value range	Attributes
X	Input variable	0.0	REAL	
NRM	Normalization, TD/TU	1.0	REAL	
EV	Control error	0.0	REAL	
LU	Upper limit	0.0	REAL	
LL	Lower limit	0.0	REAL	
SV	Setting value, output	0.0	REAL	
ASV	Acceleration setting value	0.0	REAL	
WD	Weighting of the standard deviation	0.0	REAL	
TU	Ramp-up time (ms)	0.0	SDTIME	
TD	Ramp-down time (ms)	0.0	SDTIME	
TRU	Smoothing time during ramp up (ms)	0.0	SDTIME	
TR1	Smoothing time during ramp up (ms)	0.0	SDTIME	
TR2	Smoothing time during ramp up (ms)	0.0	SDTIME	
TRD	Smoothing time during ramp down (ms)	0.0	SDTIME	
TR3	Smoothing time during ramp down (ms)	0.0	SDTIME	
TR4	Smoothing time during ramp down (ms)	0.0	SDTIME	
CU	Higher	0	0/1	
CD	Lower	0	0/1	
CF	Output = input	0	0/1	
ULR	Upper limit reached	0	0/1	
LLR	Lower limit reached	0	0/1	
RQN	Smoothing on	0	0/1	
SA	Set acceleration	0	0/1	
S	Set	0	0/1	
EN	Enable	0	0/1	
Y	Output variable	0.0	REAL	
YL	Limited output variable	0.0	REAL	
YA	Acceleration value	0.0	REAL	
YB	Jerk value	0.0	REAL	
QE	Output Y = limited input X	0	0/1	
QU	Upper limit reached	0	0/1	
QL	Lower limit reached	0	0/1	

Project data

Can be loaded online	Yes
Special characteristics	-

Appendix A

A

A.1 Data types

The table lists the data types relevant for the DCBLIB.

Table A-1 Overview of data types of the block interfaces

Abbreviation	Data width	Data type in line with IEC 61131-3	Postfix for DCB designator	PIN designator - Input - Output	Can be interconnected with data type	Description
BO/B	1-bit	BOOL	_B*	I, I1, I2,.... Q, Q1, Q2;....	BOOL	Bool
BY	8-bit	BYTE	_BY	IS QS	BY, SINT, USINT	Bit string
W	16-bit	WORD	_W		WORD, INT, UINT	Bit string
DW	32-bit	DWORD	_DW		DWORD, DINT, UDINT	Bit string
SI	8-bit	SINT	_SI	X, X1, X2, ... Y, Y1, Y2, ...	SINT, USINT, BY	Signed Short Integer
I	16-bit	INT	_I		INT, UINT, WORD	Signed Integer
DI/D	32-bit	DINT	_D		DINT, UDINT, DWORD	Signed Double Integer
US	8-bit	USINT	_US		SINT, USINT, BY	Unsigned Short Integer
UI	16-bit	UINT	_UI		INT, UINT, WORD	Unsigned Integer
UD	32-bit	UDINT	_UD		DINT, UDINT, DWORD	Unsigned Double Integer
R	32-bit	REAL	_R*		REAL, SDTIME	Floating Point *Single* Precision in line with IEEE 754
LR	64-bit	LREAL	_LR		LREAL	Floating Point *Double* Precision in line with IEEE 754

Abbreviation	Data width	Data type in line with IEC 61131-3	Postfix for DCB designator	PIN designator - Input - Output	Can be interconnected with data type	Description
TS	32-bit	(SDTIME)	-	-	SDTIME, REAL	The SDTIME data type is derived from the REAL data type; 1.0 corresponds to 1.0 msNegative values are not defined
AID	32-bit	-	-	-	DINT, UDINT, DWORD	Alarm ID
*	-	Block-defined	-	-	See DCC editor description	See DCC editor description

The table lists the fields for DPV1 parameter request and response.

Table A-2 Overview of fields for DPV1 parameter request and response

Field	Data type	Values	Remark
Request reference	Unsigned8	0x01 to 0xFF	
	Unique identification of the request/response pair for the master. The master changes the request reference with each new request. The slave mirrors the request reference in its response.		
Request ID	Unsigned8	0x010x02	Read requestWrite request
	Specifies the type of request. In the case of a write request, the changes are made in the volatile memory (RAM). A save operation must be executed to transfer the changed data to the non-volatile memory (p0971, p0977).		
Response ID	Unsigned8	0x010x020x810x82	Read request (+)Write request (+)Read request (-)Write request (-)
	Mirrors the request ID and specifies whether the request was executed positively or negatively. Negative means:The request could completely or partially not be executed. For each subresponse, the error values are transferred instead of the values.		
Drive object number	Unsigned8	0x01 to 0x27	No. 1 to 39Limited by DPV1 telegram length
	For a multi-parameter request, it defines the number of the following areas: Parameter address and/or parameter value.For single requests, Number of parameters = 1.		
Attribute	Unsigned8	0x100x200x30	ValueDescriptionText (not implemented)
	Type of parameter element accessed		
Number of elements	Unsigned8	0x000x01 to 0x75	Special functionNo. 1 to 117Limited by DPV1 telegram length
	Number of array elements accessed		
Parameter number	Unsigned16	0x0001 to 0xFFFF	No. 1 to 65535
	Addresses the parameter accessed		

Field	Data type	Values	Remark
Subindex	Unsigned16	0x0000 to 0xFFFF	No. 0 to 65535
	Addresses the first array element of the parameter to be accessed		
Format	Unsigned8	0x020x030x040x050x060x070x08Other values0x40 0x410x420x430x44	Data type Integer8Data type Integer16Data type Integer32Data type Unsigned8Data type Unsigned16Data type Unsigned32Data type FloatingPointSee PROFIdrive Profile V3.1Zero (without values as a positive subresponse of a write request)ByteWordDouble wordError
	The format and number specify the adjoining space containing values in the telegram. For write access, it is preferable to specify data types according to the PROFIdrive profile. Bytes, words, and double words are also possible as a substitute.		
Number of values	Unsigned8	0x00 to 0xEA	No. 0 to 234Limited by DPV1 telegram length
	Specifies the number of subsequent values.		
Error values	Unsigned16	0x0000 to 0x00FF	Meaning of the error values, see Appendix A.2
	The error values in the event of a negative response. If the values make up an odd number of bytes, a zero byte is attached. This ensures the integrity of the word structure of the telegram.		
Values	Unsigned16	0x0000 to 0x00FF	
	The values of the parameter for read or write access. If the values make up an odd number of bytes, a zero byte is attached. This ensures the integrity of the word structure of the telegram.		

A.2 Error values in PROFIdrive parameter responses, data types

Table A-3 Error values in DPV1 parameter responses

Error value	Meaning	Remark	Additional info
0x00	Illegal parameter number.	Access to a parameter that is not available.	-
0x01	Parameter value cannot be changed.	Modification access to a parameter value that cannot be changed.	Subindex
0x02	Lower or upper value limit exceeded.	Modification access with value outside value limits.	Subindex
0x03	Invalid subindex.	Access to a subindex that is not available.	Subindex

Error value	Meaning	Remark	Additional info
0x04	No array.	Access with subindex to an unindexed parameter.	-
0x05	Wrong data type.	Modification access with a value that does not match the data type of the parameter.	-
0x06	Illegal set operation (only reset allowed).	Modification access with a value not equal to 0 in a case where this is not allowed.	Subindex
0x07	Description element cannot be changed.	Modification access to a description element that cannot be changed.	Subindex
0x09	No description data available.	Access to a description that is not available (parameter value is available).	-
0x0B	No operating priority.	Modification access with no operating priority.	-
0x0F	No text array available.	Access to a text array that is not available (parameter value is available).	-
0x11	Request cannot be executed due to operating status.	Access is temporarily not possible for unspecified reasons.	-
0x14	Illegal value.	Modification access with a value that is within the limits but is illegal for other permanent reasons (parameter with defined individual values).	Subindex
0x15	Response too long.	The length of the present response exceeds the maximum transfer length.	-
0x16	Illegal parameter address.	Illegal or unsupported value for attribute, number of elements, parameter number, subindex or a combination of these.	-
0x17	Illegal format.	Write request: Illegal or unsupported parameter data format.	-
0x18	Number of values inconsistent.	Write request: A mismatch exists between the number of values in the parameter data and the number of elements in the parameter address.	-
0x19	Drive object does not exist.	You have attempted to access a drive object that does not exist.	-
0x20	The text element of the parameter cannot be changed.	-	-
0x21	BMP service is not supported; invalid request ID.	-	-
0x22	Multi-parameter accessing is not supported.	-	-

Error value	Meaning	Remark	Additional info
0x65	Parameter presently deactivated.	Access to a parameter that, although available, is currently inactive (e.g. n control set and access to parameter from V/f control).	-
0x6B	Parameter %s [%s]: No write access for the enabled controller.	-	-
0x6C	Parameter %s [%s]: Unknown unit.	-	-
0x6D	Parameter %s [%s]: Write access only in the commissioning state, encoder (p0010 = 4).	-	-
0x6E	Parameter %s [%s]: Write access only in the commissioning state, motor (p0010 = 3).	-	-
0x6F	Parameter %s [%s]: Write access only in the commissioning state, power module (p0010 = 2).	-	-
0x70	Parameter %s [%s]: Write access only in quick commissioning (p0010 = 1).	-	-
0x71	Parameter %s [%s]: Write access only in the ready state (p0010 = 0).	-	-
0x72	Parameter %s [%s]: Write access only in the commissioning state,parameter reset (p0010 = 30).	-	-
0x73	Parameter %s [%s]: Write access only in the commissioning state, safety (p0010 = 95).	-	-
0x74	Parameter %s [%s]: Write access only in the commissioning state, tech. application/units (p0010 = 5).	-	-
0x75	Parameter %s [%s]: Write access only in the commissioning state (p0010 not equal to 0).	-	-
0x76	Parameter %s [%s]: Write access only in the commissioning state, download (p0010 = 29).	-	-

Error value	Meaning	Remark	Additional info
0x77	Parameter %s [%s] must not be written during download.	-	-
0x78	Parameter %s [%s]: Write access only in the commissioning state, drive configuration (device: p0009 = 3).	-	-
0x79	Parameter %s [%s]: Write access only in the commissioning state, define drive type (device: p0009 = 2).	-	-
0x7A	Parameter %s [%s]: Write access only in the commissioning state, data set basis configuration (device: p0009 = 4).	-	-
0x7B	Parameter %s [%s]: Write access only in the commissioning state, device configuration (device: p0009 = 1).	-	-
0x7C	Parameter %s [%s]: Write access only in the commissioning state, device download (device: p0009 = 29).	-	-
0x7D	Parameter %s [%s]: Write access only in the commissioning state, device parameter reset (device: p0009 = 30).	-	-
0x7E	Parameter %s [%s]: Write access only in the commissioning state, device ready (device: p0009 = 0).	-	-
0x7F	Parameter %s [%s]: Write access only in the commissioning state, device (device: p0009 not equal to 0).	-	-
0x81	Parameter %s [%s] must not be written during download.	-	-
0x82	Transfer of the master control is inhibited by BI: p0806.	-	-
0x83	Parameter %s [%s]: Requested BICO interconnection not possible.	BICO output does not supply float values, but the BICO input requires float values.	-

Error value	Meaning	Remark	Additional info
0x84	Parameter %s [%s]: Parameter change inhibited (refer to p0300, p0400, p0922).	-	-
0x85	Parameter %s [%s]: Access method not defined.	-	-
0xC8	Below currently valid limit.	Modification request for a value that, although within "absolute" limits, is below the currently valid lower limit.	-
0xC9	Above currently valid limit.	Modification request for a value that, although within "absolute" limits, is above the currently valid upper limit (e.g., specified by the actual converter rating).	-
0xCC	Write access not permitted.	Write access is not permitted because an access code is not available.	-
0xFF	Successful read/write procedure.	The value has been successfully read or written.	-

A.3 Block overview

For new blocks, the version as of which they are available is indicated.

Block	Description	SIMOTION	SINAMICS
ACOS	Arc cosine function	X	
ADD	Adder (REAL type)	X	X
ADD_D	Adder (double integer type)	X	X
ADD_I	Adder (integer type)	X	X
ADD_M	Modulo adder for addition in correct axis cycle	X	X
AND	Logic AND operation (BOOL type)	X	X
AND_W	Logic AND operation (WORD type)	X	
ASIN	Arc sine function	X	
ATAN	Arc tangent function	X	
AVA	Absolute value generator, with sign evaluation	X	X
AVA_D	Absolute value generator (double integer)	X	X
BF	Flashing function (BOOL type)	X	X
BF_W	Image function for status word (Type WORD)	X	
BSW	Binary change-over switch (BOOL type)	X	X
BY_B	Converter status byte to 8 binary variables	X	
BY_W	Status byte to status word converter	X	X
B_BY	Converter 8 binary variable to status byte	X	
B_DW	Converter 32 binary variables to status double word	X	X
B_W	Converter 16 binary variables to status word	X	X
CNM	Controllable numeric memory (REAL type)	X	X
CNM_D	Controllable numeric memory (DOUBLE INTEGER type)	X	X
CNM_I	Controllable numeric memory (INTEGER type)	X	X
COS	Cosine function	X	V4.4
CTD	Time difference determination from an internal time stamp	X	
CTR	Counter (BOOL type)	X	X
DCA	Diameter calculator	X	X
DEL	Dead zone element	X	X
DEZ	Dead zone element	X	X
DFR	Reset-dominant D-type flip-flop (BOOL type)	X	X
DFR_W	Reset-dominant D-type flip-flop (WORD type)	X	
DIF	Derivative-action element	X	X
DIV	Divider (REAL type)	X	X
DIV_D	Divider (double integer type)	X	X
DIV_I	Divider (integer type)	X	X
DLB	Delay element (REAL type)	X	X
DT1	Smoothing element	X	X
DW_B	Converter status double word to 32 binary variables	X	X
DW_R	Accepting bit string as real value	X	X
DW_W	Status double word to status word converter	X	X
DX8	Demultiplexer, 8 outputs, cascadable (REAL type)	X	X

DX8_D	Demultiplexer, 8 outputs, cascadable (Double integer type)	X	X
DX8_I	Demultiplexer, 8 outputs, cascadable (INTEGER type)	X	X
D_I	DOUBLE INTEGER to INTEGER converter	X	X
D_R	DOUBLE-INTEGER to REAL converter	X	X
D_SI	DOUBLE INTEGER to SHORT INTEGER converter	X	
D_UI	DOUBLE INTEGER to UNSIGNED INTEGER converter	X	X
D_US	DOUBLE INTEGER to UNSIGNED SHORT INTEGER converter	X	X
ETE	Edge evaluator (BOOL type)	X	X
GTS	Reading out a time stamp	X	
INCO	Axial winder moment of inertia	X	X
INT	Integrator	X	X
I_D	INTEGER to DOUBLE_INTEGER converter	X	X
I_R	INTEGER to REAL converter	X	X
I_SI	INTEGER to SHORT INTEGER converter	X	
I_UD	INTEGER to UNSIGNED DOUBLE INTEGER converter	X	X
I_US	INTEGER to UNSIGNED SHORT INTEGER converter	X	X
LIM	Limiter (REAL type)	X	X
LIM_D	Limiter (DOUBLE INTEGER type)	X	X
LR_R	LONG REAL to REAL converter	X	
LVM	Double-sided limit monitor with hysteresis (type BOOL)	X	X
MAS	Maximum evaluator	X	X
MFP	Pulse generator (type BOOL)	X	X
MIS	Minimum evaluator	X	X
MUL	Multiplier (REAL type)	X	X
MUL_D	Multiplier (double integer type)	X	X
MUL_I	Multiplier (integer type)	X	X
MUX8	Multiplexer, cascadable (REAL type)	X	X
MUX8_D	Multiplexer, cascadable (double integer type)	X	X
MUX8_I	Multiplexer, cascadable (INTEGER type)	X	X
MVS	Sliding-type mean value generator	X	X
N2_R	Converting 16-bit fixed-point format (N2) to REAL	X	X
N4_R	Converting 32-bit fixed-point format (N4) to REAL	X	X
NAND	Logic AND operation (BOOL type)	X	X
NCM	Numeric comparator (REAL type)	X	X
NCM_D	Numeric comparator (DOUBLE_INTEGER type)	X	X
NCM_I	Numeric comparator (INTEGER type)	X	X
NOP1	Dummy blocks (REAL type)	X	X
NOP1_B	Dummy block (BOOL type)	X	X
NOP1_D	Dummy block (DOUBLE INTEGER type)	X	X
NOP1_I	Dummy block (INT type)	X	X
NOP8	Dummy blocks (REAL type)	X	X
NOP8_B	Dummy blocks (BOOL type)	X	X
NOP8_D	Dummy blocks (DOUBLE INTEGER type)	X	X
NOP8_I	Dummy blocks (INTEGER type)	X	X

NOR	Logic OR operation (BOOL type)	X	X
NOT	Inverter (BOOL type)	X	X
NOT_W	Status word inverter (WORD type)	X	
NSW	Numeric change-over switch (REAL type)	X	X
NSW_D	Numeric change-over switch (DOUBLE INTEGER type)	X	X
NSW_I	Numeric change-over switch (INTEGER type)	X	X
OCA	Software cam controller	X	X
OR	Logic OR operation (BOOL type)	X	X
OR_W	Logic OR operation (WORD type)	X	
PC	P-action controller	X	X
PCL	Pulse shortening device (BOOL type)	X	X
PDE	On-delay device (BOOL type)	X	X
PDF	Off-delay device (BOOL type)	X	X
PIC	PI controller	X	X
PLI20	Polyline, 20 breakpoints	X	X
PST	Pulse stretching block (BOOL type)	X	X
PT1	Delay element	X	X
RAA	Reset all messages	V4.3	
RDA	Reading out message	V4.3	
RDAA	Reading out of all messages	V4.3	
RDP	Reading drive parameters (REAL type)		X
RDP_D	Reading drive parameters (DOUBLE INTEGER type)		X
RDP_I	Reading drive parameters (INTEGER type)		X
RDP_UD	Reading drive parameters (UNSIGNED DOUBLE INTEGER type)		X
RDP_UI	Reading drive parameters (UNSIGNED INTEGER type)		X
RDP_US	Reading drive parameters (UNSIGNED SHORT INTEGER type)		X
RGE	Ramp-function generator	X	X
RGJ	Ramp-function generator with jerk limiting	X	X
RMDP	Reading of drive parameters from the controller	V4.2	
RSR	RS flip-flop, R-dominant (BOOL type)	X	X
RSS	RS flip-flop, S-dominant (BOOL type)	X	X
R_D	REAL to DOUBLE INTEGER converter	X	X
R_DW	Bit string transfer as DWORD	X	X
R_I	REAL to INTEGER converter	X	X
R_LR	REAL to LONG REAL converter	X	
R_N2	Converting REAL to 16-bit fixed-point format (N2)	X	X
R_N4	Converting REAL to 32-bit fixed-point format (N4)	X	X
R_SI	REAL to SHORT INTEGER converter	X	
R_UD	REAL to UNSIGNED DOUBLE INTEGER converter	X	X
R_UI	REAL to UNSIGNED INTEGER converter	X	X
R_US	REAL to UNSIGNED SHORT INTEGER converter	X	X
SAH	Sample & hold (REAL type)		X
SAH_B	Sample & hold (BOOL type)		X
SAH_BY	Sample & hold (BYTE type)		X

SAH_D	Sample & hold (DOUBLE INTEGER type)		X
SAH_I	Sample & hold (INTEGER type)		X
SAV	Value buffering (REAL type)	X	X
SAV_BY	Value buffering (BYTE type)	X	X
SAV_D	Value buffering (DOUBLE INTEGER type)	X	X
SAV_I	Value buffering (INTEGER type)	X	X
SH	Shift block (WORD type)	X	
SH_DW	Shift block (DWORD type)	X	X
SII	Inverter	X	X
SIN	Sine function	X	V4.4
SI_D	SHORT INTEGER to DOUBLE INTEGER converter	X	
SI_I	SHORT INTEGER to INTEGER converter	X	
SI_R	SHORT INTEGER to REAL converter	X	
SI_UD	SHORT INTEGER to UNSIGNED DOUBLE INTEGER converter	X	
SI_UI	SHORT INTEGER to UNSIGNED INTEGER converter	X	
SQR	Square-root extractor	X	
SRA	Triggering/resetting of a message	V4.3	
STM	Fault/alarm trigger		X
SUB	Subtractor (REAL type)	X	X
SUB_D	Subtractor (double integer type)	X	X
SUB_I	Subtractor (integer type)	X	X
TAN	Tangent	X	
TRK	Correction/memory element (REAL type)	X	X
TRK_D	Correction/memory element (DOUBLE INTEGER type)	X	X
TTCU	Winding characteristic	X	X
UD_I	UNSIGNED DOUBLE INTEGER to INTEGER converter	X	X
UD_R	UNSIGNED DOUBLE INTEGER to REAL converter	X	X
UD_SI	UNSIGNED DOUBLE INTEGER to SHORT INTEGER converter	X	
UI_D	UNSIGNED INTEGER to DOUBLE INTEGER converter	X	X
UI_R	UNSIGNED INTEGER to REAL converter	X	X
UI_SI	UNSIGNED INTEGER to SHORT INTEGER converter	X	
US_D	UNSIGNED SHORT INTEGER to DOUBLE INTEGER converter	X	X
US_I	UNSIGNED SHORT INTEGER to INTEGER converter	X	X
US_R	UNSIGNED SHORT INTEGER to REAL converter	X	X
WBG	Wobble generator	X	X
WMDP	Writing of drive parameters from the controller	V4.2	
WRP	Writing drive parameters (REAL type)		X
WRP_D	Writing drive parameters (DOUBLE INTEGER type)		X
WRP_I	Writing drive parameters (INTEGER type)		X
WRP_UD	Writing drive parameters (UNSIGNED DOUBLE INTEGER type)		X
WRP_UI	Writing drive parameters (UNSIGNED INTEGER type)		X
WRP_US	Writing drive parameters (UNSIGNED SHORT INTEGER type)		X
W_B	Converter, status word to 16 binary variables	X	X
W_BY	Status word to status byte converter	X	X

W_DW	Status word to status double word converter	X	X
XOR	Logic exclusive OR operation (BOOL type)	X	X
XOR_W	Logic exclusive OR operation (WORD type)	X	

Appendix B

B

B.1 Messages

> Note
>
> The messages described in this chapter only apply for SINAMICS Version 4.70.03.

A51032 DCC: Internal measurement active
Drive object: All objects
Reaction: NONE
Acknowledgement: NONE
Cause: A Siemens internal measurement has been activated.
Remedy: Carry out a POWER ON (power off/on) for the Control Unit involved.

A51060 DCC: Alarm initiated by "Drive Control Chart"
Drive object: All objects
Reaction: NONE
Acknowledgement: NONE
Cause: "Drive Control Chart" (DCC) has initiated this message via the block "Set Message" (STM) or via a block (SINAMICS DCB Extension) generated using SINAMICS DCB Studio.
Alarm value (r2124, interpret decimal):
The configured message value is displayed in r2124.
Remedy: This message was configured with "Drive Control Chart" (DCC).
The cause and remedy depend on the project and should be described in the corresponding project documentation.

A51061 DCC: Alarm initiated by block STM
Drive object: All objects
Reaction: NONE
Acknowledgement: NONE
Cause: "Drive Control Chart" (DCC) has initiated this message via the block "Set Message" (STM) or via a block (SINAMICS DCB Extension) generated using SINAMICS DCB Studio.
Alarm value (r2124, interpret decimal):
The configured message value is displayed in r2124.
Remedy: This message was configured with "Drive Control Chart" (DCC).
The cause and remedy depend on the project and should be described in the corresponding project documentation.

A51062 DCC: Alarm initiated by block STM
Drive object: All objects
Reaction: NONE
Acknowledgement: NONE
Cause: "Drive Control Chart" (DCC) has initiated this message via the block "Set Message" (STM) or via a block (SINAMICS DCB Extension) generated using SINAMICS DCB Studio.
Alarm value (r2124, interpret decimal):
The configured message value is displayed in r2124.
Remedy: This message was configured with "Drive Control Chart" (DCC).
The cause and remedy depend on the project and should be described in the corresponding project documentation.

A51063	DCC: Alarm initiated by block STM
Drive object:	All objects
Reaction:	NONE
Acknowledgement:	NONE
Cause:	"Drive Control Chart" (DCC) has initiated this message via the block "Set Message" (STM) or via a block (SINAMICS DCB Extension) generated using SINAMICS DCB Studio. Alarm value (r2124, interpret decimal): The configured message value is displayed in r2124.
Remedy:	This message was configured with "Drive Control Chart" (DCC). The cause and remedy depend on the project and should be described in the corresponding project documentation.
A51064	DCC: Alarm initiated by block STM
Drive object:	All objects
Reaction:	NONE
Acknowledgement:	NONE
Cause:	"Drive Control Chart" (DCC) has initiated this message via the block "Set Message" (STM) or via a block (SINAMICS DCB Extension) generated using SINAMICS DCB Studio. Alarm value (r2124, interpret decimal): The configured message value is displayed in r2124.
Remedy:	This message was configured with "Drive Control Chart" (DCC). The cause and remedy depend on the project and should be described in the corresponding project documentation.
A51065	DCC: Alarm initiated by block STM
Drive object:	All objects
Reaction:	NONE
Acknowledgement:	NONE
Cause:	"Drive Control Chart" (DCC) has initiated this message via the block "Set Message" (STM) or via a block (SINAMICS DCB Extension) generated using SINAMICS DCB Studio. Alarm value (r2124, interpret decimal): The configured message value is displayed in r2124.
Remedy:	This message was configured with "Drive Control Chart" (DCC). The cause and remedy depend on the project and should be described in the corresponding project documentation.
A51066	DCC: Alarm initiated by block STM
Drive object:	All objects
Reaction:	NONE
Acknowledgement:	NONE
Cause:	"Drive Control Chart" (DCC) has initiated this message via the block "Set Message" (STM) or via a block (SINAMICS DCB Extension) generated using SINAMICS DCB Studio. Alarm value (r2124, interpret decimal): The configured message value is displayed in r2124.
Remedy:	This message was configured with "Drive Control Chart" (DCC). The cause and remedy depend on the project and should be described in the corresponding project documentation.

A51067 DCC: Alarm initiated by block STM
Drive object: All objects
Reaction: NONE
Acknowledgement: NONE
Cause: "Drive Control Chart" (DCC) has initiated this message via the block "Set Message" (STM) or via a block (SINAMICS DCB Extension) generated using SINAMICS DCB Studio.
Alarm value (r2124, interpret decimal):
The configured message value is displayed in r2124.
Remedy: This message was configured with "Drive Control Chart" (DCC).
The cause and remedy depend on the project and should be described in the corresponding project documentation.

A51068 DCC: Alarm initiated by block STM
Drive object: All objects
Reaction: NONE
Acknowledgement: NONE
Cause: "Drive Control Chart" (DCC) has initiated this message via the block "Set Message" (STM) or via a block (SINAMICS DCB Extension) generated using SINAMICS DCB Studio.
Alarm value (r2124, interpret decimal):
The configured message value is displayed in r2124.
Remedy: This message was configured with "Drive Control Chart" (DCC).
The cause and remedy depend on the project and should be described in the corresponding project documentation.

A51069 DCC: Alarm initiated by block STM
Drive object: All objects
Reaction: NONE
Acknowledgement: NONE
Cause: "Drive Control Chart" (DCC) has initiated this message via the block "Set Message" (STM) or via a block (SINAMICS DCB Extension) generated using SINAMICS DCB Studio.
Alarm value (r2124, interpret decimal):
The configured message value is displayed in r2124.
Remedy: This message was configured with "Drive Control Chart" (DCC).
The cause and remedy depend on the project and should be described in the corresponding project documentation.

F51000 DCC: Logon of the run-time group with sampling time management rejected
Drive object: All objects
Reaction: NONE
Acknowledgement: IMMEDIATELY
Cause: The OA application "Drive Control Chart" (DCC) attempted to log on a sampling time that cannot be implemented with the sampling time management of the basic SINAMICS system. The logon was rejected.
Remedy: Try to assign this run-time group another fixed or free run-time group.
The assignment is set in STARTER in the context menu of the DCC chart via sampling times.
Then compile the chart and download it again into the drive unit.

F51001 DCC: No further hardware sampling times available

Drive object: All objects

Reaction: NONE

Acknowledgement: IMMEDIATELY

Cause: The drive unit can no longer provide any additional hardware sampling times, whose sampling time deviates from the sampling times.already logged on.

Remedy: The fault can be immediately acknowledged, as the system run-time group 0 (corresponds to "Do not calculate") was assigned in p21000[x].
Fault value (r0949, interpret hexadecimal):
yyyyxxxx hex
yyyy: The upper 16 bits of the fault value specify the number of the drive object.
xxxx: The lower 16 bits specify the index of the run-time group in p21000.
Note:
In window "Set run-time groups" in the context menu of the chart, p21000[0] is the topmost entry and p21000[9] the lowest entry.
The current assignment of hardware sampling times can be read-out in r21008.

F51004 DCC: Sampling time of the free run-time group differs at download

Drive object: All objects

Reaction: NONE

Acknowledgement: IMMEDIATELY

Cause: In the STARTER/SCOUT project that was downloaded, the hardware sampling time of a free run-time group (1 <= p21000[i] <= 256) was set to a value that was either too low or too high. The sampling time must be between 1 ms and the value (r21003 - r21002).
If the sampling time of the selected free run-time group is < 1 ms, the equivalent value of 1 ms is used.
If the value >= r21003, then the sampling time is set to the next higher or the same software sampling time >= r21003.
The free run-time group involved is assigned as a minimum to one block.
If this fault still occurs during download after the selection in p21000[i] in the project has been corrected, please check which run-time group is involved on the basis of the fault value (r0949). Only one F51004 fault is signaled at a time, even if several run-time groups have been incorrectly parameterized in p21000[].
Fault value (r0949, decimal interpretation):
Number of the p21000 index of the run-time group where the sampling time was incorrectly set.
Number of the run-time group = fault value + 1
Note:
With SIMOTION D410, r21003 (unlike all the other Control Units) is automatically set the same as the PROFIBUS sampling time.

Remedy: Correctly set the sampling time of the run-time group or remove all of the blocks from the run-time group.

F51005 DCC: Sampling time of the fixed run-time group differs online

Drive object: All objects

Reaction: NONE

Acknowledgement: IMMEDIATELY

Cause: Generally, the sampling times of the fixed run-time groups correspond to the sampling times of the associated system function (e.g. the sampling time of the fixed run-time group "BEFORE speed controller" generally corresponds to the sampling of the speed controller p0115[1]).
The sampling time of a system function online was set to a lower value (e.g. with p0112, p0115, p0799, p4099) than the smallest permissible sampling time that is allowed for the fixed run-time group belonging to this system function (1 ms). The sampling time is set to 1 ms. The fixed run-time group involved is assigned as a minimum to one block.
Fault value (r0949, decimal interpretation):
Number of the p21000 index of the run-time group where the sampling time was incorrectly set.
Number of the run-time group = fault value + 1

Remedy: Using parameter p0112 or p0115, increase the sampling time of the system function to the minimum permissible sampling time for the run-time groups of 1 ms or remove all of the blocks from the run-time group.

F51006 DCC: Sampling time of the fixed run-time group differs at download

Drive object: All objects

Reaction: NONE

Acknowledgement: IMMEDIATELY

Cause: Generally, the sampling times of the fixed run-time groups correspond to the sampling times of the associated system function (e.g. the sampling time of the fixed run-time group "BEFORE speed controller" generally corresponds to the sampling of the speed controller p0115[1]).
During a download, the sampling time of a system function was set to a lower value (p0112, p0115) than the smallest permissible sampling time that is allowed for the fixed run-time group belonging to this system function (1 ms). The sampling time is set to the smallest possible value (r21002 on the drive object).
Fault value (r0949, decimal interpretation):
Number of the p21000 index of the run-time group where the sampling time was incorrectly set.
Number of the run-time group = fault value + 1

Remedy: Using parameter p0112 or p0115, increase the sampling time of the system function to the minimum permissible sampling time for the run-time groups of 1 ms or remove all of the blocks from the run-time group.

F51008 DCC: No NVRAM available

Drive object: All objects

Reaction: OFF2

Acknowledgement: IMMEDIATELY

Cause: The DCC project contains at least one block that requires remanent memory from the basic SINAMICS system (e.g. SAV, SAV_BY, SAV_D, SAV_I). The request for remanent memory was rejected by the basic SINAMICS system.
Fault value (r0949, decimal interpretation):
0: There is no more free remanent memory available on the drive unit.
1: The EPROM data of the drive unit indicates that there is no remanent memory on the module.

Remedy: For fault value = 0:
- Deactivate other applications on the drive unit that use remanent memory.
- Do not use blocks that require remanent memory in your DCC charts.
For fault value = 1:
- For modules D425 or D435, use hardware version D or higher.
Note:
You can read out the hardware version using SCOUT in online mode under Target system --> Device diagnostics --> tab "General" in the lower window, 3rd column in the line of the CPU.

F51009 DCC: Project data and block library are incompatible

Drive object: All objects

Reaction: OFF2

Acknowledgement: IMMEDIATELY

Cause: The block library and the saved or downloaded project data are incompatible.

Remedy: Make sure that the block library and project data match.

- Update the block library in SINAMICS by downloading the technology package.

or

- Update the project data in the DCC Editor by importing the correct block library.

F51033 Licensing DCC application not sufficient

Drive object: All objects

Reaction: NONE

Acknowledgement: IMMEDIATELY

Cause: There is a license error in a DCB block.

Remedy: -Obtain the necessary license.

-Later licensing is not possible online via p9920, 9921.

F51050 DCC: Fault initiated by "Drive Control Chart"

Drive object: All objects

Reaction: OFF2 OFF2 OFF2 OFF2 OFF2

Acknowledgement: IMMEDIATELY

Cause: "Drive Control Chart" (DCC) has initiated this message via the block "Set Message" (STM) or via a block (SINAMICS DCB Extension) generated using SINAMICS DCB Studio.

Fault value (r0949, decimal interpretation):

The configured message value is displayed in r0949.

Remedy: This message was configured with "Drive Control Chart" (DCC).

The cause and remedy depend on the project and should be described in the corresponding project documentation.

F51051 DCC: Fault initiated by "Drive Control Chart"

Drive object: All objects

Reaction: OFF2 OFF2 OFF2 OFF2 OFF2

Acknowledgement: IMMEDIATELY

Cause: "Drive Control Chart" (DCC) has initiated this message via the block "Set Message" (STM) or via a block (SINAMICS DCB Extension) generated using SINAMICS DCB Studio.

Fault value (r0949, decimal interpretation):

The configured message value is displayed in r0949.

Remedy: This message was configured with "Drive Control Chart" (DCC).

The cause and remedy depend on the project and should be described in the corresponding project documentation.

F51052 DCC: Fault initiated by "Drive Control Chart"

Drive object: All objects

Reaction: OFF2 OFF2 OFF2 OFF2 OFF2

Acknowledgement: IMMEDIATELY

Cause: "Drive Control Chart" (DCC) has initiated this message via the block "Set Message" (STM) or via a block (SINAMICS DCB Extension) generated using SINAMICS DCB Studio.

Fault value (r0949, decimal interpretation):

The configured message value is displayed in r0949.

Remedy: This message was configured with "Drive Control Chart" (DCC).

The cause and remedy depend on the project and should be described in the corresponding project documentation.

F51053 DCC: Fault initiated by "Drive Control Chart"

Drive object: All objects

Reaction: OFF2 OFF2 OFF2 OFF2 OFF2

Acknowledgement: IMMEDIATELY

Cause: "Drive Control Chart" (DCC) has initiated this message via the block "Set Message" (STM) or via a block (SINAMICS DCB Extension) generated using SINAMICS DCB Studio.
Fault value (r0949, decimal interpretation):
The configured message value is displayed in r0949.

Remedy: This message was configured with "Drive Control Chart" (DCC).
The cause and remedy depend on the project and should be described in the corresponding project documentation.

F51054 DCC: Fault initiated by "Drive Control Chart"

Drive object: All objects

Reaction: OFF2 OFF2 OFF2 OFF2 OFF2

Acknowledgement: IMMEDIATELY

Cause: "Drive Control Chart" (DCC) has initiated this message via the block "Set Message" (STM) or via a block (SINAMICS DCB Extension) generated using SINAMICS DCB Studio.
Fault value (r0949, decimal interpretation):
The configured message value is displayed in r0949.

Remedy: This message was configured with "Drive Control Chart" (DCC).
The cause and remedy depend on the project and should be described in the corresponding project documentation.

F51055 DCC: Fault initiated by "Drive Control Chart"

Drive object: All objects

Reaction: OFF2 OFF2 OFF2 OFF2 OFF2

Acknowledgement: IMMEDIATELY

Cause: "Drive Control Chart" (DCC) has initiated this message via the block "Set Message" (STM) or via a block (SINAMICS DCB Extension) generated using SINAMICS DCB Studio.
Fault value (r0949, decimal interpretation):
The configured message value is displayed in r0949.

Remedy: This message was configured with "Drive Control Chart" (DCC).
The cause and remedy depend on the project and should be described in the corresponding project documentation.

F51056 DCC: Fault initiated by "Drive Control Chart"

Drive object: All objects

Reaction: OFF2 OFF2 OFF2 OFF2 OFF2

Acknowledgement: IMMEDIATELY

Cause: "Drive Control Chart" (DCC) has initiated this message via the block "Set Message" (STM) or via a block (SINAMICS DCB Extension) generated using SINAMICS DCB Studio.
Fault value (r0949, decimal interpretation):
The configured message value is displayed in r0949.

Remedy: This message was configured with "Drive Control Chart" (DCC).
The cause and remedy depend on the project and should be described in the corresponding project documentation.

F51057 DCC: Fault initiated by "Drive Control Chart"
Drive object: All objects
Reaction: OFF2 OFF2 OFF2 OFF2 OFF2
Acknowledgement: IMMEDIATELY
Cause: "Drive Control Chart" (DCC) has initiated this message via the block "Set Message" (STM) or via a block (SINAMICS DCB Extension) generated using SINAMICS DCB Studio.
Fault value (r0949, decimal interpretation):
The configured message value is displayed in r0949.
Remedy: This message was configured with "Drive Control Chart" (DCC).
The cause and remedy depend on the project and should be described in the corresponding project documentation.

F51058 DCC: Fault initiated by "Drive Control Chart"
Drive object: All objects
Reaction: OFF2 OFF2 OFF2 OFF2 OFF2
Acknowledgement: IMMEDIATELY
Cause: "Drive Control Chart" (DCC) has initiated this message via the block "Set Message" (STM) or via a block (SINAMICS DCB Extension) generated using SINAMICS DCB Studio.
Fault value (r0949, decimal interpretation):
The configured message value is displayed in r0949.
Remedy: This message was configured with "Drive Control Chart" (DCC).
The cause and remedy depend on the project and should be described in the corresponding project documentation.

F51059 DCC: Fault initiated by "Drive Control Chart"
Drive object: All objects
Reaction: OFF2 OFF2 OFF2 OFF2 OFF2
Acknowledgement: IMMEDIATELY
Cause: "Drive Control Chart" (DCC) has initiated this message via the block "Set Message" (STM) or via a block (SINAMICS DCB Extension) generated using SINAMICS DCB Studio.
Fault value (r0949, decimal interpretation):
The configured message value is displayed in r0949.
Remedy: This message was configured with "Drive Control Chart" (DCC).
The cause and remedy depend on the project and should be described in the corresponding project documentation.

B.2 Parameters

Note

The parameters described in this chapter only apply for SINAMICS Version 4.70.03.

p21000[0...9]	**Run-time group properties / RTG property**		
A_INF	Changeable: -	Calculated: -	Access level: 1
	Data type: Integer16	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	0	4004	[0] 0

Description: Allocates properties to run-time groups 1 to 10.
This property comprises the sampling time and, for p21000[x] >= 2000, the instant of the call within the sampling time.
The index x + 1 of p21000 corresponds to the number of the run-time group:
- p21000[0] is used to set the property of the run-time group 1
...
- p21000[9] is used to set the property of the run-time group 10

Values:

0:	Do not calculate run-time group
1:	T = 1 * r21002
2:	T = 2 * r21002
3:	T = 3 * r21002
4:	T = 4 * r21002
5:	T = 5 * r21002
6:	T = 6 * r21002
7:	T = 7 * r21002
8:	T = 8 * r21002
9:	T = 9 * r21002
10:	T = 10 * r21002
11:	T = 11 * r21002
12:	T = 12 * r21002
13:	T = 13 * r21002
14:	T = 14 * r21002
15:	T = 15 * r21002
16:	T = 16 * r21002
17:	T = 17 * r21002
18:	T = 18 * r21002
19:	T = 19 * r21002
20:	T = 20 * r21002
21:	T = 21 * r21002
22:	T = 22 * r21002
23:	T = 23 * r21002
24:	T = 24 * r21002
25:	T = 25 * r21002
26:	T = 26 * r21002
27:	T = 27 * r21002
28:	T = 28 * r21002
29:	T = 29 * r21002

30:	T = 30 * r21002
31:	T = 31 * r21002
32:	T = 32 * r21002
33:	T = 33 * r21002
34:	T = 34 * r21002
35:	T = 35 * r21002
36:	T = 36 * r21002
37:	T = 37 * r21002
38:	T = 38 * r21002
39:	T = 39 * r21002
40:	T = 40 * r21002
41:	T = 41 * r21002
42:	T = 42 * r21002
43:	T = 43 * r21002
44:	T = 44 * r21002
45:	T = 45 * r21002
46:	T = 46 * r21002
47:	T = 47 * r21002
48:	T = 48 * r21002
49:	T = 49 * r21002
50:	T = 50 * r21002
51:	T = 51 * r21002
52:	T = 52 * r21002
53:	T = 53 * r21002
54:	T = 54 * r21002
55:	T = 55 * r21002
56:	T = 56 * r21002
57:	T = 57 * r21002
58:	T = 58 * r21002
59:	T = 59 * r21002
60:	T = 60 * r21002
61:	T = 61 * r21002
62:	T = 62 * r21002
63:	T = 63 * r21002
64:	T = 64 * r21002
65:	T = 65 * r21002
66:	T = 66 * r21002
67:	T = 67 * r21002
68:	T = 68 * r21002
69:	T = 69 * r21002
70:	T = 70 * r21002
71:	T = 71 * r21002
72:	T = 72 * r21002
73:	T = 73 * r21002
74:	T = 74 * r21002
75:	T = 75 * r21002
76:	T = 76 * r21002
77:	T = 77 * r21002
78:	T = 78 * r21002
79:	T = 79 * r21002
80:	T = 80 * r21002
81:	T = 81 * r21002

82:	T = 82 * r21002
83:	T = 83 * r21002
84:	T = 84 * r21002
85:	T = 85 * r21002
86:	T = 86 * r21002
87:	T = 87 * r21002
88:	T = 88 * r21002
89:	T = 89 * r21002
90:	T = 90 * r21002
91:	T = 91 * r21002
92:	T = 92 * r21002
93:	T = 93 * r21002
94:	T = 94 * r21002
95:	T = 95 * r21002
96:	T = 96 * r21002
97:	T = 97 * r21002
98:	T = 98 * r21002
99:	T = 99 * r21002
100:	T = 100 * r21002
101:	T = 101 * r21002
102:	T = 102 * r21002
103:	T = 103 * r21002
104:	T = 104 * r21002
105:	T = 105 * r21002
106:	T = 106 * r21002
107:	T = 107 * r21002
108:	T = 108 * r21002
109:	T = 109 * r21002
110:	T = 110 * r21002
111:	T = 111 * r21002
112:	T = 112 * r21002
113:	T = 113 * r21002
114:	T = 114 * r21002
115:	T = 115 * r21002
116:	T = 116 * r21002
117:	T = 117 * r21002
118:	T = 118 * r21002
119:	T = 119 * r21002
120:	T = 120 * r21002
121:	T = 121 * r21002
122:	T = 122 * r21002
123:	T = 123 * r21002
124:	T = 124 * r21002
125:	T = 125 * r21002
126:	T = 126 * r21002
127:	T = 127 * r21002
128:	T = 128 * r21002
129:	T = 129 * r21002
130:	T = 130 * r21002
131:	T = 131 * r21002
132:	T = 132 * r21002
133:	T = 133 * r21002

134:	T = 134 * r21002
135:	T = 135 * r21002
136:	T = 136 * r21002
137:	T = 137 * r21002
138:	T = 138 * r21002
139:	T = 139 * r21002
140:	T = 140 * r21002
141:	T = 141 * r21002
142:	T = 142 * r21002
143:	T = 143 * r21002
144:	T = 144 * r21002
145:	T = 145 * r21002
146:	T = 146 * r21002
147:	T = 147 * r21002
148:	T = 148 * r21002
149:	T = 149 * r21002
150:	T = 150 * r21002
151:	T = 151 * r21002
152:	T = 152 * r21002
153:	T = 153 * r21002
154:	T = 154 * r21002
155:	T = 155 * r21002
156:	T = 156 * r21002
157:	T = 157 * r21002
158:	T = 158 * r21002
159:	T = 159 * r21002
160:	T = 160 * r21002
161:	T = 161 * r21002
162:	T = 162 * r21002
163:	T = 163 * r21002
164:	T = 164 * r21002
165:	T = 165 * r21002
166:	T = 166 * r21002
167:	T = 167 * r21002
168:	T = 168 * r21002
169:	T = 169 * r21002
170:	T = 170 * r21002
171:	T = 171 * r21002
172:	T = 172 * r21002
173:	T = 173 * r21002
174:	T = 174 * r21002
175:	T = 175 * r21002
176:	T = 176 * r21002
177:	T = 177 * r21002
178:	T = 178 * r21002
179:	T = 179 * r21002
180:	T = 180 * r21002
181:	T = 181 * r21002
182:	T = 182 * r21002
183:	T = 183 * r21002
184:	T = 184 * r21002
185:	T = 185 * r21002

186:	T = 186 * r21002
187:	T = 187 * r21002
188:	T = 188 * r21002
189:	T = 189 * r21002
190:	T = 190 * r21002
191:	T = 191 * r21002
192:	T = 192 * r21002
193:	T = 193 * r21002
194:	T = 194 * r21002
195:	T = 195 * r21002
196:	T = 196 * r21002
197:	T = 197 * r21002
198:	T = 198 * r21002
199:	T = 199 * r21002
200:	T = 200 * r21002
201:	T = 201 * r21002
202:	T = 202 * r21002
203:	T = 203 * r21002
204:	T = 204 * r21002
205:	T = 205 * r21002
206:	T = 206 * r21002
207:	T = 207 * r21002
208:	T = 208 * r21002
209:	T = 209 * r21002
210:	T = 210 * r21002
211:	T = 211 * r21002
212:	T = 212 * r21002
213:	T = 213 * r21002
214:	T = 214 * r21002
215:	T = 215 * r21002
216:	T = 216 * r21002
217:	T = 217 * r21002
218:	T = 218 * r21002
219:	T = 219 * r21002
220:	T = 220 * r21002
221:	T = 221 * r21002
222:	T = 222 * r21002
223:	T = 223 * r21002
224:	T = 224 * r21002
225:	T = 225 * r21002
226:	T = 226 * r21002
227:	T = 227 * r21002
228:	T = 228 * r21002
229:	T = 229 * r21002
230:	T = 230 * r21002
231:	T = 231 * r21002
232:	T = 232 * r21002
233:	T = 233 * r21002
234:	T = 234 * r21002
235:	T = 235 * r21002
236:	T = 236 * r21002
237:	T = 237 * r21002

	238:	T = 238 * r21002
	239:	T = 239 * r21002
	240:	T = 240 * r21002
	241:	T = 241 * r21002
	242:	T = 242 * r21002
	243:	T = 243 * r21002
	244:	T = 244 * r21002
	245:	T = 245 * r21002
	246:	T = 246 * r21002
	247:	T = 247 * r21002
	248:	T = 248 * r21002
	249:	T = 249 * r21002
	250:	T = 250 * r21002
	251:	T = 251 * r21002
	252:	T = 252 * r21002
	253:	T = 253 * r21002
	254:	T = 254 * r21002
	255:	T = 255 * r21002
	256:	T = 256 * r21002
	1001:	T = 1 * r21003
	1002:	T = 2 * r21003
	1003:	T = 3 * r21003
	1004:	T = 4 * r21003
	1005:	T = 5 * r21003
	1006:	T = 6 * r21003
	1008:	T = 8 * r21003
	1010:	T = 10 * r21003
	1012:	T = 12 * r21003
	1016:	T = 16 * r21003
	1020:	T = 20 * r21003
	1024:	T = 24 * r21003
	1032:	T = 32 * r21003
	1040:	T = 40 * r21003
	1048:	T = 48 * r21003
	1064:	T = 64 * r21003
	1080:	T = 80 * r21003
	1096:	T = 96 * r21003
	3003:	BEFORE speed setpoint channel
	4000:	Receive AFTER IF1 PROFIdrive PZD
	4001:	Send BEFORE IF1 PROFIdrive PZD
	4004:	Receive AFTER IF1 PROFIdrive flexible PZD
Index:	[0]:	Run-time group 1
	[1]:	Run-time group 2
	[2]:	Run-time group 3
	[3]:	Run-time group 4
	[4]:	Run-time group 5
	[5]:	Run-time group 6
	[6]:	Run-time group 7
	[7]:	Run-time group 8
	[8]:	Run-time group 9
	[9]:	Run-time group 10

Caution: The properties of the run-time groups must not be changed during operation as this could result in discontinuous signal transitions.

Note: Re value = 1 ... 256 (free run-time group):
This selection value can only be selected online if the following applies for sampling time T_sample of this run-time group:
1 ms <= T_sample < r21003.
At download, a value that violates this condition is not rejected, but a permissible equivalent value is set automatically and fault F51004 is output.
Re value > 2000 (fixed run-time group):
The fixed run-time groups p21000[x] >= 2000 log on with the sampling time of the associated basic system function, subject to a minimum sampling time of 1 ms. If, as a result of this limit, the actual sampling time deviates from the sampling time of the basic system function, then fault F51005 (during F51006 download) is output. In this case, another run-time group with a sampling time >= 1 ms should be selected. When selecting the fixed run-time groups, a check is not made as to whether the associated system block exists.
Example:
"BEFORE speed setpoint channel" means before function charts 3010, 3020, 3030, 3040, etc. are calculated, if the setpoint channel is activated. If, e.g. for SERVO, a setpoint channel has not been configured (p0108.8 = 0), the calculation is made before function chart 3095.

p21000[0...9]	**Run-time group properties / RTG property**		
BMM2C,	Changeable: -	Calculated: -	Access level: 1
VECTORMV,	Data type: Integer16	Dynamic index: -	Function plan: -
VECTORGL	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	0	4005	[0] 0

Description: Allocates properties to run-time groups 1 to 10.
This property comprises the sampling time and, for p21000[x] >= 2000, the instant of the call within the sampling time.
The index x + 1 of p21000 corresponds to the number of the run-time group:
- p21000[0] is used to set the property of the run-time group 1
...
- p21000[9] is used to set the property of the run-time group 10

Values:		
	0:	Do not calculate run-time group
	1:	T = 1 * r21002
	2:	T = 2 * r21002
	3:	T = 3 * r21002
	4:	T = 4 * r21002
	5:	T = 5 * r21002
	6:	T = 6 * r21002
	7:	T = 7 * r21002
	8:	T = 8 * r21002
	9:	T = 9 * r21002
	10:	T = 10 * r21002
	11:	T = 11 * r21002
	12:	T = 12 * r21002
	13:	T = 13 * r21002
	14:	T = 14 * r21002
	15:	T = 15 * r21002
	16:	T = 16 * r21002
	17:	T = 17 * r21002
	18:	T = 18 * r21002
	19:	T = 19 * r21002
	20:	T = 20 * r21002
	21:	T = 21 * r21002
	22:	T = 22 * r21002
	23:	T = 23 * r21002
	24:	T = 24 * r21002
	25:	T = 25 * r21002
	26:	T = 26 * r21002
	27:	T = 27 * r21002
	28:	T = 28 * r21002
	29:	T = 29 * r21002
	30:	T = 30 * r21002
	31:	T = 31 * r21002
	32:	T = 32 * r21002
	33:	T = 33 * r21002
	34:	T = 34 * r21002
	35:	T = 35 * r21002
	36:	T = 36 * r21002

37:	T = 37 * r21002
38:	T = 38 * r21002
39:	T = 39 * r21002
40:	T = 40 * r21002
41:	T = 41 * r21002
42:	T = 42 * r21002
43:	T = 43 * r21002
44:	T = 44 * r21002
45:	T = 45 * r21002
46:	T = 46 * r21002
47:	T = 47 * r21002
48:	T = 48 * r21002
49:	T = 49 * r21002
50:	T = 50 * r21002
51:	T = 51 * r21002
52:	T = 52 * r21002
53:	T = 53 * r21002
54:	T = 54 * r21002
55:	T = 55 * r21002
56:	T = 56 * r21002
57:	T = 57 * r21002
58:	T = 58 * r21002
59:	T = 59 * r21002
60:	T = 60 * r21002
61:	T = 61 * r21002
62:	T = 62 * r21002
63:	T = 63 * r21002
64:	T = 64 * r21002
65:	T = 65 * r21002
66:	T = 66 * r21002
67:	T = 67 * r21002
68:	T = 68 * r21002
69:	T = 69 * r21002
70:	T = 70 * r21002
71:	T = 71 * r21002
72:	T = 72 * r21002
73:	T = 73 * r21002
74:	T = 74 * r21002
75:	T = 75 * r21002
76:	T = 76 * r21002
77:	T = 77 * r21002
78:	T = 78 * r21002
79:	T = 79 * r21002
80:	T = 80 * r21002
81:	T = 81 * r21002
82:	T = 82 * r21002
83:	T = 83 * r21002
84:	T = 84 * r21002
85:	T = 85 * r21002
86:	T = 86 * r21002
87:	T = 87 * r21002
88:	T = 88 * r21002

89:	T = 89 * r21002
90:	T = 90 * r21002
91:	T = 91 * r21002
92:	T = 92 * r21002
93:	T = 93 * r21002
94:	T = 94 * r21002
95:	T = 95 * r21002
96:	T = 96 * r21002
97:	T = 97 * r21002
98:	T = 98 * r21002
99:	T = 99 * r21002
100:	T = 100 * r21002
101:	T = 101 * r21002
102:	T = 102 * r21002
103:	T = 103 * r21002
104:	T = 104 * r21002
105:	T = 105 * r21002
106:	T = 106 * r21002
107:	T = 107 * r21002
108:	T = 108 * r21002
109:	T = 109 * r21002
110:	T = 110 * r21002
111:	T = 111 * r21002
112:	T = 112 * r21002
113:	T = 113 * r21002
114:	T = 114 * r21002
115:	T = 115 * r21002
116:	T = 116 * r21002
117:	T = 117 * r21002
118:	T = 118 * r21002
119:	T = 119 * r21002
120:	T = 120 * r21002
121:	T = 121 * r21002
122:	T = 122 * r21002
123:	T = 123 * r21002
124:	T = 124 * r21002
125:	T = 125 * r21002
126:	T = 126 * r21002
127:	T = 127 * r21002
128:	T = 128 * r21002
129:	T = 129 * r21002
130:	T = 130 * r21002
131:	T = 131 * r21002
132:	T = 132 * r21002
133:	T = 133 * r21002
134:	T = 134 * r21002
135:	T = 135 * r21002
136:	T = 136 * r21002
137:	T = 137 * r21002
138:	T = 138 * r21002
139:	T = 139 * r21002
140:	T = 140 * r21002

141:	T = 141 * r21002
142:	T = 142 * r21002
143:	T = 143 * r21002
144:	T = 144 * r21002
145:	T = 145 * r21002
146:	T = 146 * r21002
147:	T = 147 * r21002
148:	T = 148 * r21002
149:	T = 149 * r21002
150:	T = 150 * r21002
151:	T = 151 * r21002
152:	T = 152 * r21002
153:	T = 153 * r21002
154:	T = 154 * r21002
155:	T = 155 * r21002
156:	T = 156 * r21002
157:	T = 157 * r21002
158:	T = 158 * r21002
159:	T = 159 * r21002
160:	T = 160 * r21002
161:	T = 161 * r21002
162:	T = 162 * r21002
163:	T = 163 * r21002
164:	T = 164 * r21002
165:	T = 165 * r21002
166:	T = 166 * r21002
167:	T = 167 * r21002
168:	T = 168 * r21002
169:	T = 169 * r21002
170:	T = 170 * r21002
171:	T = 171 * r21002
172:	T = 172 * r21002
173:	T = 173 * r21002
174:	T = 174 * r21002
175:	T = 175 * r21002
176:	T = 176 * r21002
177:	T = 177 * r21002
178:	T = 178 * r21002
179:	T = 179 * r21002
180:	T = 180 * r21002
181:	T = 181 * r21002
182:	T = 182 * r21002
183:	T = 183 * r21002
184:	T = 184 * r21002
185:	T = 185 * r21002
186:	T = 186 * r21002
187:	T = 187 * r21002
188:	T = 188 * r21002
189:	T = 189 * r21002
190:	T = 190 * r21002
191:	T = 191 * r21002
192:	T = 192 * r21002

193:	T = 193 * r21002
194:	T = 194 * r21002
195:	T = 195 * r21002
196:	T = 196 * r21002
197:	T = 197 * r21002
198:	T = 198 * r21002
199:	T = 199 * r21002
200:	T = 200 * r21002
201:	T = 201 * r21002
202:	T = 202 * r21002
203:	T = 203 * r21002
204:	T = 204 * r21002
205:	T = 205 * r21002
206:	T = 206 * r21002
207:	T = 207 * r21002
208:	T = 208 * r21002
209:	T = 209 * r21002
210:	T = 210 * r21002
211:	T = 211 * r21002
212:	T = 212 * r21002
213:	T = 213 * r21002
214:	T = 214 * r21002
215:	T = 215 * r21002
216:	T = 216 * r21002
217:	T = 217 * r21002
218:	T = 218 * r21002
219:	T = 219 * r21002
220:	T = 220 * r21002
221:	T = 221 * r21002
222:	T = 222 * r21002
223:	T = 223 * r21002
224:	T = 224 * r21002
225:	T = 225 * r21002
226:	T = 226 * r21002
227:	T = 227 * r21002
228:	T = 228 * r21002
229:	T = 229 * r21002
230:	T = 230 * r21002
231:	T = 231 * r21002
232:	T = 232 * r21002
233:	T = 233 * r21002
234:	T = 234 * r21002
235:	T = 235 * r21002
236:	T = 236 * r21002
237:	T = 237 * r21002
238:	T = 238 * r21002
239:	T = 239 * r21002
240:	T = 240 * r21002
241:	T = 241 * r21002
242:	T = 242 * r21002
243:	T = 243 * r21002
244:	T = 244 * r21002

	245:	T = 245 * r21002
	246:	T = 246 * r21002
	247:	T = 247 * r21002
	248:	T = 248 * r21002
	249:	T = 249 * r21002
	250:	T = 250 * r21002
	251:	T = 251 * r21002
	252:	T = 252 * r21002
	253:	T = 253 * r21002
	254:	T = 254 * r21002
	255:	T = 255 * r21002
	256:	T = 256 * r21002
	1001:	T = 1 * r21003
	1002:	T = 2 * r21003
	1003:	T = 3 * r21003
	1004:	T = 4 * r21003
	1005:	T = 5 * r21003
	1006:	T = 6 * r21003
	1008:	T = 8 * r21003
	1010:	T = 10 * r21003
	1012:	T = 12 * r21003
	1016:	T = 16 * r21003
	1020:	T = 20 * r21003
	1024:	T = 24 * r21003
	1032:	T = 32 * r21003
	1040:	T = 40 * r21003
	1048:	T = 48 * r21003
	1064:	T = 64 * r21003
	1080:	T = 80 * r21003
	1096:	T = 96 * r21003
	3001:	BEFORE speed ctrl
	3003:	BEFORE speed setpoint channel
	3006:	BEFORE standard technology controller
	4000:	Receive AFTER IF1 PROFIdrive PZD
	4001:	Send BEFORE IF1 PROFIdrive PZD
	4002:	Receive AFTER IF2 PZD
	4003:	Send BEFORE IF2 PZD
	4004:	Receive AFTER IF1 PROFIdrive flexible PZD
	4005:	Receive AFTER IF2 flexible PZD
Index:	[0]:	Run-time group 1
	[1]:	Run-time group 2
	[2]:	Run-time group 3
	[3]:	Run-time group 4
	[4]:	Run-time group 5
	[5]:	Run-time group 6
	[6]:	Run-time group 7
	[7]:	Run-time group 8
	[8]:	Run-time group 9
	[9]:	Run-time group 10

Caution: The properties of the run-time groups must not be changed during operation as this could result in discontinuous signal transitions.

Note: Re value = 1 ... 256 (free run-time group):
This selection value can only be selected online if the following applies for sampling time T_sample of this run-time group:
1 ms <= T_sample < r21003.
At download, a value that violates this condition is not rejected, but a permissible equivalent value is set automatically and fault F51004 is output.
Re value > 2000 (fixed run-time group):
The fixed run-time groups p21000[x] >= 2000 log on with the sampling time of the associated basic system function, subject to a minimum sampling time of 1 ms. If, as a result of this limit, the actual sampling time deviates from the sampling time of the basic system function, then fault F51005 (during F51006 download) is output. In this case, another run-time group with a sampling time >= 1 ms should be selected. When selecting the fixed run-time groups, a check is not made as to whether the associated system block exists.
Example:
"BEFORE speed setpoint channel" means before function charts 3010, 3020, 3030, 3040, etc. are calculated, if the setpoint channel is activated. If, e.g. for SERVO, a setpoint channel has not been configured (p0108.8 = 0), the calculation is made before function chart 3095.
Re value = 4002, 4003, 4005 (IF2 run-time group):
On devices where IF2 does not exist (D4xx, CU310), when selecting the run-time groups that involve IF2, the corresponding run-time group for IF1 is automatically logged on.

p21000[0...9]	**Run-time group properties / RTG property**		
CU_DC, HLA	Changeable: -	Calculated: -	Access level: 1
	Data type: Integer16	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	0	4005	[0] 0

Description:

Allocates properties to run-time groups 1 to 10.
This property comprises the sampling time and, for p21000[x] >= 2000, the instant of the call within the sampling time.
The index x + 1 of p21000 corresponds to the number of the run-time group:
- p21000[0] is used to set the property of the run-time group 1
...
- p21000[9] is used to set the property of the run-time group 10

Values:

0:	Do not calculate run-time group
1:	T = 1 * r21002
2:	T = 2 * r21002
3:	T = 3 * r21002
4:	T = 4 * r21002
5:	T = 5 * r21002
6:	T = 6 * r21002
7:	T = 7 * r21002
8:	T = 8 * r21002
9:	T = 9 * r21002
10:	T = 10 * r21002
11:	T = 11 * r21002
12:	T = 12 * r21002
13:	T = 13 * r21002
14:	T = 14 * r21002
15:	T = 15 * r21002
16:	T = 16 * r21002
17:	T = 17 * r21002
18:	T = 18 * r21002
19:	T = 19 * r21002
20:	T = 20 * r21002
21:	T = 21 * r21002
22:	T = 22 * r21002
23:	T = 23 * r21002
24:	T = 24 * r21002
25:	T = 25 * r21002
26:	T = 26 * r21002
27:	T = 27 * r21002
28:	T = 28 * r21002
29:	T = 29 * r21002
30:	T = 30 * r21002
31:	T = 31 * r21002
32:	T = 32 * r21002
33:	T = 33 * r21002
34:	T = 34 * r21002
35:	T = 35 * r21002
36:	T = 36 * r21002

37:	T = 37 * r21002
38:	T = 38 * r21002
39:	T = 39 * r21002
40:	T = 40 * r21002
41:	T = 41 * r21002
42:	T = 42 * r21002
43:	T = 43 * r21002
44:	T = 44 * r21002
45:	T = 45 * r21002
46:	T = 46 * r21002
47:	T = 47 * r21002
48:	T = 48 * r21002
49:	T = 49 * r21002
50:	T = 50 * r21002
51:	T = 51 * r21002
52:	T = 52 * r21002
53:	T = 53 * r21002
54:	T = 54 * r21002
55:	T = 55 * r21002
56:	T = 56 * r21002
57:	T = 57 * r21002
58:	T = 58 * r21002
59:	T = 59 * r21002
60:	T = 60 * r21002
61:	T = 61 * r21002
62:	T = 62 * r21002
63:	T = 63 * r21002
64:	T = 64 * r21002
65:	T = 65 * r21002
66:	T = 66 * r21002
67:	T = 67 * r21002
68:	T = 68 * r21002
69:	T = 69 * r21002
70:	T = 70 * r21002
71:	T = 71 * r21002
72:	T = 72 * r21002
73:	T = 73 * r21002
74:	T = 74 * r21002
75:	T = 75 * r21002
76:	T = 76 * r21002
77:	T = 77 * r21002
78:	T = 78 * r21002
79:	T = 79 * r21002
80:	T = 80 * r21002
81:	T = 81 * r21002
82:	T = 82 * r21002
83:	T = 83 * r21002
84:	T = 84 * r21002
85:	T = 85 * r21002
86:	T = 86 * r21002
87:	T = 87 * r21002
88:	T = 88 * r21002

89:	T = 89 * r21002
90:	T = 90 * r21002
91:	T = 91 * r21002
92:	T = 92 * r21002
93:	T = 93 * r21002
94:	T = 94 * r21002
95:	T = 95 * r21002
96:	T = 96 * r21002
97:	T = 97 * r21002
98:	T = 98 * r21002
99:	T = 99 * r21002
100:	T = 100 * r21002
101:	T = 101 * r21002
102:	T = 102 * r21002
103:	T = 103 * r21002
104:	T = 104 * r21002
105:	T = 105 * r21002
106:	T = 106 * r21002
107:	T = 107 * r21002
108:	T = 108 * r21002
109:	T = 109 * r21002
110:	T = 110 * r21002
111:	T = 111 * r21002
112:	T = 112 * r21002
113:	T = 113 * r21002
114:	T = 114 * r21002
115:	T = 115 * r21002
116:	T = 116 * r21002
117:	T = 117 * r21002
118:	T = 118 * r21002
119:	T = 119 * r21002
120:	T = 120 * r21002
121:	T = 121 * r21002
122:	T = 122 * r21002
123:	T = 123 * r21002
124:	T = 124 * r21002
125:	T = 125 * r21002
126:	T = 126 * r21002
127:	T = 127 * r21002
128:	T = 128 * r21002
129:	T = 129 * r21002
130:	T = 130 * r21002
131:	T = 131 * r21002
132:	T = 132 * r21002
133:	T = 133 * r21002
134:	T = 134 * r21002
135:	T = 135 * r21002
136:	T = 136 * r21002
137:	T = 137 * r21002
138:	T = 138 * r21002
139:	T = 139 * r21002
140:	T = 140 * r21002

141:	T = 141 * r21002
142:	T = 142 * r21002
143:	T = 143 * r21002
144:	T = 144 * r21002
145:	T = 145 * r21002
146:	T = 146 * r21002
147:	T = 147 * r21002
148:	T = 148 * r21002
149:	T = 149 * r21002
150:	T = 150 * r21002
151:	T = 151 * r21002
152:	T = 152 * r21002
153:	T = 153 * r21002
154:	T = 154 * r21002
155:	T = 155 * r21002
156:	T = 156 * r21002
157:	T = 157 * r21002
158:	T = 158 * r21002
159:	T = 159 * r21002
160:	T = 160 * r21002
161:	T = 161 * r21002
162:	T = 162 * r21002
163:	T = 163 * r21002
164:	T = 164 * r21002
165:	T = 165 * r21002
166:	T = 166 * r21002
167:	T = 167 * r21002
168:	T = 168 * r21002
169:	T = 169 * r21002
170:	T = 170 * r21002
171:	T = 171 * r21002
172:	T = 172 * r21002
173:	T = 173 * r21002
174:	T = 174 * r21002
175:	T = 175 * r21002
176:	T = 176 * r21002
177:	T = 177 * r21002
178:	T = 178 * r21002
179:	T = 179 * r21002
180:	T = 180 * r21002
181:	T = 181 * r21002
182:	T = 182 * r21002
183:	T = 183 * r21002
184:	T = 184 * r21002
185:	T = 185 * r21002
186:	T = 186 * r21002
187:	T = 187 * r21002
188:	T = 188 * r21002
189:	T = 189 * r21002
190:	T = 190 * r21002
191:	T = 191 * r21002
192:	T = 192 * r21002

193:	T = 193 * r21002
194:	T = 194 * r21002
195:	T = 195 * r21002
196:	T = 196 * r21002
197:	T = 197 * r21002
198:	T = 198 * r21002
199:	T = 199 * r21002
200:	T = 200 * r21002
201:	T = 201 * r21002
202:	T = 202 * r21002
203:	T = 203 * r21002
204:	T = 204 * r21002
205:	T = 205 * r21002
206:	T = 206 * r21002
207:	T = 207 * r21002
208:	T = 208 * r21002
209:	T = 209 * r21002
210:	T = 210 * r21002
211:	T = 211 * r21002
212:	T = 212 * r21002
213:	T = 213 * r21002
214:	T = 214 * r21002
215:	T = 215 * r21002
216:	T = 216 * r21002
217:	T = 217 * r21002
218:	T = 218 * r21002
219:	T = 219 * r21002
220:	T = 220 * r21002
221:	T = 221 * r21002
222:	T = 222 * r21002
223:	T = 223 * r21002
224:	T = 224 * r21002
225:	T = 225 * r21002
226:	T = 226 * r21002
227:	T = 227 * r21002
228:	T = 228 * r21002
229:	T = 229 * r21002
230:	T = 230 * r21002
231:	T = 231 * r21002
232:	T = 232 * r21002
233:	T = 233 * r21002
234:	T = 234 * r21002
235:	T = 235 * r21002
236:	T = 236 * r21002
237:	T = 237 * r21002
238:	T = 238 * r21002
239:	T = 239 * r21002
240:	T = 240 * r21002
241:	T = 241 * r21002
242:	T = 242 * r21002
243:	T = 243 * r21002
244:	T = 244 * r21002

	245:	T = 245 * r21002
	246:	T = 246 * r21002
	247:	T = 247 * r21002
	248:	T = 248 * r21002
	249:	T = 249 * r21002
	250:	T = 250 * r21002
	251:	T = 251 * r21002
	252:	T = 252 * r21002
	253:	T = 253 * r21002
	254:	T = 254 * r21002
	255:	T = 255 * r21002
	256:	T = 256 * r21002
	1001:	T = 1 * r21003
	1002:	T = 2 * r21003
	1003:	T = 3 * r21003
	1004:	T = 4 * r21003
	1005:	T = 5 * r21003
	1006:	T = 6 * r21003
	1008:	T = 8 * r21003
	1010:	T = 10 * r21003
	1012:	T = 12 * r21003
	1016:	T = 16 * r21003
	1020:	T = 20 * r21003
	1024:	T = 24 * r21003
	1032:	T = 32 * r21003
	1040:	T = 40 * r21003
	1048:	T = 48 * r21003
	1064:	T = 64 * r21003
	1080:	T = 80 * r21003
	1096:	T = 96 * r21003
	4000:	Receive AFTER IF1 PROFIdrive PZD
	4001:	Send BEFORE IF1 PROFIdrive PZD
	4002:	Receive AFTER IF2 PZD
	4003:	Send BEFORE IF2 PZD
	4004:	Receive AFTER IF1 PROFIdrive flexible PZD
	4005:	Receive AFTER IF2 flexible PZD
Index:	[0]:	Run-time group 1
	[1]:	Run-time group 2
	[2]:	Run-time group 3
	[3]:	Run-time group 4
	[4]:	Run-time group 5
	[5]:	Run-time group 6
	[6]:	Run-time group 7
	[7]:	Run-time group 8
	[8]:	Run-time group 9
	[9]:	Run-time group 10

Caution: The properties of the run-time groups must not be changed during operation as this could result in discontinuous signal transitions.

Note: Re value = 1 ... 256 (free run-time group):
This selection value can only be selected online if the following applies for sampling time T_sample of this run-time group:

1 ms <= T_sample < r21003.
At download, a value that violates this condition is not rejected, but a permissible equivalent value is set automatically and fault F51004 is output.
Re value > 2000 (fixed run-time group):
The fixed run-time groups p21000[x] >= 2000 log on with the sampling time of the associated basic system function, subject to a minimum sampling time of 1 ms. If, as a result of this limit, the actual sampling time deviates from the sampling time of the basic system function, then fault F51005 (during F51006 download) is output. In this case, another run-time group with a sampling time >= 1 ms should be selected. When selecting the fixed run-time groups, a check is not made as to whether the associated system block exists.
Example:
"BEFORE speed setpoint channel" means before function charts 3010, 3020, 3030, 3040, etc. are calculated, if the setpoint channel is activated. If, e.g. for SERVO, a setpoint channel has not been configured (p0108.8 = 0), the calculation is made before function chart 3095.
Re value = 4002, 4003, 4005 (IF2 run-time group):
On devices where IF2 does not exist (D4xx, CU310), when selecting the run-time groups that involve IF2, the corresponding run-time group for IF1 is automatically logged on.

p21000[0...9]	Run-time group properties / RTG property		
CU_I, TM15BASE	Changeable: -	Calculated: -	Access level: 1
	Data type: Integer16	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	0	4004	[0] 0

Description: Allocates properties to run-time groups 1 to 10.
This property comprises the sampling time and, for p21000[x] >= 2000, the instant of the call within the sampling time.
The index x + 1 of p21000 corresponds to the number of the run-time group:
- p21000[0] is used to set the property of the run-time group 1
...
- p21000[9] is used to set the property of the run-time group 10

Values:

0:	Do not calculate run-time group
1:	T = 1 * r21002
2:	T = 2 * r21002
3:	T = 3 * r21002
4:	T = 4 * r21002
5:	T = 5 * r21002
6:	T = 6 * r21002
7:	T = 7 * r21002
8:	T = 8 * r21002
9:	T = 9 * r21002
10:	T = 10 * r21002
11:	T = 11 * r21002
12:	T = 12 * r21002
13:	T = 13 * r21002
14:	T = 14 * r21002
15:	T = 15 * r21002
16:	T = 16 * r21002
17:	T = 17 * r21002
18:	T = 18 * r21002
19:	T = 19 * r21002
20:	T = 20 * r21002
21:	T = 21 * r21002
22:	T = 22 * r21002
23:	T = 23 * r21002
24:	T = 24 * r21002
25:	T = 25 * r21002
26:	T = 26 * r21002
27:	T = 27 * r21002
28:	T = 28 * r21002
29:	T = 29 * r21002
30:	T = 30 * r21002
31:	T = 31 * r21002
32:	T = 32 * r21002
33:	T = 33 * r21002
34:	T = 34 * r21002
35:	T = 35 * r21002
36:	T = 36 * r21002

37:	T = 37 * r21002
38:	T = 38 * r21002
39:	T = 39 * r21002
40:	T = 40 * r21002
41:	T = 41 * r21002
42:	T = 42 * r21002
43:	T = 43 * r21002
44:	T = 44 * r21002
45:	T = 45 * r21002
46:	T = 46 * r21002
47:	T = 47 * r21002
48:	T = 48 * r21002
49:	T = 49 * r21002
50:	T = 50 * r21002
51:	T = 51 * r21002
52:	T = 52 * r21002
53:	T = 53 * r21002
54:	T = 54 * r21002
55:	T = 55 * r21002
56:	T = 56 * r21002
57:	T = 57 * r21002
58:	T = 58 * r21002
59:	T = 59 * r21002
60:	T = 60 * r21002
61:	T = 61 * r21002
62:	T = 62 * r21002
63:	T = 63 * r21002
64:	T = 64 * r21002
65:	T = 65 * r21002
66:	T = 66 * r21002
67:	T = 67 * r21002
68:	T = 68 * r21002
69:	T = 69 * r21002
70:	T = 70 * r21002
71:	T = 71 * r21002
72:	T = 72 * r21002
73:	T = 73 * r21002
74:	T = 74 * r21002
75:	T = 75 * r21002
76:	T = 76 * r21002
77:	T = 77 * r21002
78:	T = 78 * r21002
79:	T = 79 * r21002
80:	T = 80 * r21002
81:	T = 81 * r21002
82:	T = 82 * r21002
83:	T = 83 * r21002
84:	T = 84 * r21002
85:	T = 85 * r21002
86:	T = 86 * r21002
87:	T = 87 * r21002
88:	T = 88 * r21002

89:	T = 89 * r21002
90:	T = 90 * r21002
91:	T = 91 * r21002
92:	T = 92 * r21002
93:	T = 93 * r21002
94:	T = 94 * r21002
95:	T = 95 * r21002
96:	T = 96 * r21002
97:	T = 97 * r21002
98:	T = 98 * r21002
99:	T = 99 * r21002
100:	T = 100 * r21002
101:	T = 101 * r21002
102:	T = 102 * r21002
103:	T = 103 * r21002
104:	T = 104 * r21002
105:	T = 105 * r21002
106:	T = 106 * r21002
107:	T = 107 * r21002
108:	T = 108 * r21002
109:	T = 109 * r21002
110:	T = 110 * r21002
111:	T = 111 * r21002
112:	T = 112 * r21002
113:	T = 113 * r21002
114:	T = 114 * r21002
115:	T = 115 * r21002
116:	T = 116 * r21002
117:	T = 117 * r21002
118:	T = 118 * r21002
119:	T = 119 * r21002
120:	T = 120 * r21002
121:	T = 121 * r21002
122:	T = 122 * r21002
123:	T = 123 * r21002
124:	T = 124 * r21002
125:	T = 125 * r21002
126:	T = 126 * r21002
127:	T = 127 * r21002
128:	T = 128 * r21002
129:	T = 129 * r21002
130:	T = 130 * r21002
131:	T = 131 * r21002
132:	T = 132 * r21002
133:	T = 133 * r21002
134:	T = 134 * r21002
135:	T = 135 * r21002
136:	T = 136 * r21002
137:	T = 137 * r21002
138:	T = 138 * r21002
139:	T = 139 * r21002
140:	T = 140 * r21002

141:	T = 141 * r21002
142:	T = 142 * r21002
143:	T = 143 * r21002
144:	T = 144 * r21002
145:	T = 145 * r21002
146:	T = 146 * r21002
147:	T = 147 * r21002
148:	T = 148 * r21002
149:	T = 149 * r21002
150:	T = 150 * r21002
151:	T = 151 * r21002
152:	T = 152 * r21002
153:	T = 153 * r21002
154:	T = 154 * r21002
155:	T = 155 * r21002
156:	T = 156 * r21002
157:	T = 157 * r21002
158:	T = 158 * r21002
159:	T = 159 * r21002
160:	T = 160 * r21002
161:	T = 161 * r21002
162:	T = 162 * r21002
163:	T = 163 * r21002
164:	T = 164 * r21002
165:	T = 165 * r21002
166:	T = 166 * r21002
167:	T = 167 * r21002
168:	T = 168 * r21002
169:	T = 169 * r21002
170:	T = 170 * r21002
171:	T = 171 * r21002
172:	T = 172 * r21002
173:	T = 173 * r21002
174:	T = 174 * r21002
175:	T = 175 * r21002
176:	T = 176 * r21002
177:	T = 177 * r21002
178:	T = 178 * r21002
179:	T = 179 * r21002
180:	T = 180 * r21002
181:	T = 181 * r21002
182:	T = 182 * r21002
183:	T = 183 * r21002
184:	T = 184 * r21002
185:	T = 185 * r21002
186:	T = 186 * r21002
187:	T = 187 * r21002
188:	T = 188 * r21002
189:	T = 189 * r21002
190:	T = 190 * r21002
191:	T = 191 * r21002
192:	T = 192 * r21002

193:	T = 193 * r21002
194:	T = 194 * r21002
195:	T = 195 * r21002
196:	T = 196 * r21002
197:	T = 197 * r21002
198:	T = 198 * r21002
199:	T = 199 * r21002
200:	T = 200 * r21002
201:	T = 201 * r21002
202:	T = 202 * r21002
203:	T = 203 * r21002
204:	T = 204 * r21002
205:	T = 205 * r21002
206:	T = 206 * r21002
207:	T = 207 * r21002
208:	T = 208 * r21002
209:	T = 209 * r21002
210:	T = 210 * r21002
211:	T = 211 * r21002
212:	T = 212 * r21002
213:	T = 213 * r21002
214:	T = 214 * r21002
215:	T = 215 * r21002
216:	T = 216 * r21002
217:	T = 217 * r21002
218:	T = 218 * r21002
219:	T = 219 * r21002
220:	T = 220 * r21002
221:	T = 221 * r21002
222:	T = 222 * r21002
223:	T = 223 * r21002
224:	T = 224 * r21002
225:	T = 225 * r21002
226:	T = 226 * r21002
227:	T = 227 * r21002
228:	T = 228 * r21002
229:	T = 229 * r21002
230:	T = 230 * r21002
231:	T = 231 * r21002
232:	T = 232 * r21002
233:	T = 233 * r21002
234:	T = 234 * r21002
235:	T = 235 * r21002
236:	T = 236 * r21002
237:	T = 237 * r21002
238:	T = 238 * r21002
239:	T = 239 * r21002
240:	T = 240 * r21002
241:	T = 241 * r21002
242:	T = 242 * r21002
243:	T = 243 * r21002
244:	T = 244 * r21002

	245:	T = 245 * r21002
	246:	T = 246 * r21002
	247:	T = 247 * r21002
	248:	T = 248 * r21002
	249:	T = 249 * r21002
	250:	T = 250 * r21002
	251:	T = 251 * r21002
	252:	T = 252 * r21002
	253:	T = 253 * r21002
	254:	T = 254 * r21002
	255:	T = 255 * r21002
	256:	T = 256 * r21002
	1001:	T = 1 * r21003
	1002:	T = 2 * r21003
	1003:	T = 3 * r21003
	1004:	T = 4 * r21003
	1005:	T = 5 * r21003
	1006:	T = 6 * r21003
	1008:	T = 8 * r21003
	1010:	T = 10 * r21003
	1012:	T = 12 * r21003
	1016:	T = 16 * r21003
	1020:	T = 20 * r21003
	1024:	T = 24 * r21003
	1032:	T = 32 * r21003
	1040:	T = 40 * r21003
	1048:	T = 48 * r21003
	1064:	T = 64 * r21003
	1080:	T = 80 * r21003
	1096:	T = 96 * r21003
	2000:	Read-in AFTER digital inputs
	2001:	Output BEFORE digital outputs
	4000:	Receive AFTER IF1 PROFIdrive PZD
	4001:	Send BEFORE IF1 PROFIdrive PZD
	4004:	Receive AFTER IF1 PROFIdrive flexible PZD
Index:	[0]:	Run-time group 1
	[1]:	Run-time group 2
	[2]:	Run-time group 3
	[3]:	Run-time group 4
	[4]:	Run-time group 5
	[5]:	Run-time group 6
	[6]:	Run-time group 7
	[7]:	Run-time group 8
	[8]:	Run-time group 9
	[9]:	Run-time group 10

Caution: The properties of the run-time groups must not be changed during operation as this could result in discontinuous signal transitions.

Note: Re value = 1 ... 256 (free run-time group):
This selection value can only be selected online if the following applies for sampling time T_sample of this run-time group:
1 ms <= T_sample < r21003.

At download, a value that violates this condition is not rejected, but a permissible equivalent value is set automatically and fault F51004 is output.
Re value > 2000 (fixed run-time group):
The fixed run-time groups p21000[x] >= 2000 log on with the sampling time of the associated basic system function, subject to a minimum sampling time of 1 ms. If, as a result of this limit, the actual sampling time deviates from the sampling time of the basic system function, then fault F51005 (during F51006 download) is output. In this case, another run-time group with a sampling time >= 1 ms should be selected. When selecting the fixed run-time groups, a check is not made as to whether the associated system block exists.
Example:
"BEFORE speed setpoint channel" means before function charts 3010, 3020, 3030, 3040, etc. are calculated, if the setpoint channel is activated. If, e.g. for SERVO, a setpoint channel has not been configured (p0108.8 = 0), the calculation is made before function chart 3095.

p21000[0...9]	**Run-time group properties / RTG property**		
CU_S, _G, _GM, _GL	Changeable: -	Calculated: -	Access level: 1
	Data type: Integer16	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	0	4005	[0] 0

Description: Allocates properties to run-time groups 1 to 10.
This property comprises the sampling time and, for p21000[x] >= 2000, the instant of the call within the sampling time.
The index x + 1 of p21000 corresponds to the number of the run-time group:
- p21000[0] is used to set the property of the run-time group 1
...
- p21000[9] is used to set the property of the run-time group 10

Values:

0:	Do not calculate run-time group
1:	T = 1 * r21002
2:	T = 2 * r21002
3:	T = 3 * r21002
4:	T = 4 * r21002
5:	T = 5 * r21002
6:	T = 6 * r21002
7:	T = 7 * r21002
8:	T = 8 * r21002
9:	T = 9 * r21002
10:	T = 10 * r21002
11:	T = 11 * r21002
12:	T = 12 * r21002
13:	T = 13 * r21002
14:	T = 14 * r21002
15:	T = 15 * r21002
16:	T = 16 * r21002
17:	T = 17 * r21002
18:	T = 18 * r21002
19:	T = 19 * r21002
20:	T = 20 * r21002
21:	T = 21 * r21002
22:	T = 22 * r21002
23:	T = 23 * r21002
24:	T = 24 * r21002
25:	T = 25 * r21002
26:	T = 26 * r21002
27:	T = 27 * r21002
28:	T = 28 * r21002
29:	T = 29 * r21002
30:	T = 30 * r21002
31:	T = 31 * r21002
32:	T = 32 * r21002
33:	T = 33 * r21002
34:	T = 34 * r21002
35:	T = 35 * r21002
36:	T = 36 * r21002

37:	T = 37 * r21002
38:	T = 38 * r21002
39:	T = 39 * r21002
40:	T = 40 * r21002
41:	T = 41 * r21002
42:	T = 42 * r21002
43:	T = 43 * r21002
44:	T = 44 * r21002
45:	T = 45 * r21002
46:	T = 46 * r21002
47:	T = 47 * r21002
48:	T = 48 * r21002
49:	T = 49 * r21002
50:	T = 50 * r21002
51:	T = 51 * r21002
52:	T = 52 * r21002
53:	T = 53 * r21002
54:	T = 54 * r21002
55:	T = 55 * r21002
56:	T = 56 * r21002
57:	T = 57 * r21002
58:	T = 58 * r21002
59:	T = 59 * r21002
60:	T = 60 * r21002
61:	T = 61 * r21002
62:	T = 62 * r21002
63:	T = 63 * r21002
64:	T = 64 * r21002
65:	T = 65 * r21002
66:	T = 66 * r21002
67:	T = 67 * r21002
68:	T = 68 * r21002
69:	T = 69 * r21002
70:	T = 70 * r21002
71:	T = 71 * r21002
72:	T = 72 * r21002
73:	T = 73 * r21002
74:	T = 74 * r21002
75:	T = 75 * r21002
76:	T = 76 * r21002
77:	T = 77 * r21002
78:	T = 78 * r21002
79:	T = 79 * r21002
80:	T = 80 * r21002
81:	T = 81 * r21002
82:	T = 82 * r21002
83:	T = 83 * r21002
84:	T = 84 * r21002
85:	T = 85 * r21002
86:	T = 86 * r21002
87:	T = 87 * r21002
88:	T = 88 * r21002

89:	T = 89 * r21002
90:	T = 90 * r21002
91:	T = 91 * r21002
92:	T = 92 * r21002
93:	T = 93 * r21002
94:	T = 94 * r21002
95:	T = 95 * r21002
96:	T = 96 * r21002
97:	T = 97 * r21002
98:	T = 98 * r21002
99:	T = 99 * r21002
100:	T = 100 * r21002
101:	T = 101 * r21002
102:	T = 102 * r21002
103:	T = 103 * r21002
104:	T = 104 * r21002
105:	T = 105 * r21002
106:	T = 106 * r21002
107:	T = 107 * r21002
108:	T = 108 * r21002
109:	T = 109 * r21002
110:	T = 110 * r21002
111:	T = 111 * r21002
112:	T = 112 * r21002
113:	T = 113 * r21002
114:	T = 114 * r21002
115:	T = 115 * r21002
116:	T = 116 * r21002
117:	T = 117 * r21002
118:	T = 118 * r21002
119:	T = 119 * r21002
120:	T = 120 * r21002
121:	T = 121 * r21002
122:	T = 122 * r21002
123:	T = 123 * r21002
124:	T = 124 * r21002
125:	T = 125 * r21002
126:	T = 126 * r21002
127:	T = 127 * r21002
128:	T = 128 * r21002
129:	T = 129 * r21002
130:	T = 130 * r21002
131:	T = 131 * r21002
132:	T = 132 * r21002
133:	T = 133 * r21002
134:	T = 134 * r21002
135:	T = 135 * r21002
136:	T = 136 * r21002
137:	T = 137 * r21002
138:	T = 138 * r21002
139:	T = 139 * r21002
140:	T = 140 * r21002

141:	T = 141 * r21002
142:	T = 142 * r21002
143:	T = 143 * r21002
144:	T = 144 * r21002
145:	T = 145 * r21002
146:	T = 146 * r21002
147:	T = 147 * r21002
148:	T = 148 * r21002
149:	T = 149 * r21002
150:	T = 150 * r21002
151:	T = 151 * r21002
152:	T = 152 * r21002
153:	T = 153 * r21002
154:	T = 154 * r21002
155:	T = 155 * r21002
156:	T = 156 * r21002
157:	T = 157 * r21002
158:	T = 158 * r21002
159:	T = 159 * r21002
160:	T = 160 * r21002
161:	T = 161 * r21002
162:	T = 162 * r21002
163:	T = 163 * r21002
164:	T = 164 * r21002
165:	T = 165 * r21002
166:	T = 166 * r21002
167:	T = 167 * r21002
168:	T = 168 * r21002
169:	T = 169 * r21002
170:	T = 170 * r21002
171:	T = 171 * r21002
172:	T = 172 * r21002
173:	T = 173 * r21002
174:	T = 174 * r21002
175:	T = 175 * r21002
176:	T = 176 * r21002
177:	T = 177 * r21002
178:	T = 178 * r21002
179:	T = 179 * r21002
180:	T = 180 * r21002
181:	T = 181 * r21002
182:	T = 182 * r21002
183:	T = 183 * r21002
184:	T = 184 * r21002
185:	T = 185 * r21002
186:	T = 186 * r21002
187:	T = 187 * r21002
188:	T = 188 * r21002
189:	T = 189 * r21002
190:	T = 190 * r21002
191:	T = 191 * r21002
192:	T = 192 * r21002

193:	T = 193 * r21002
194:	T = 194 * r21002
195:	T = 195 * r21002
196:	T = 196 * r21002
197:	T = 197 * r21002
198:	T = 198 * r21002
199:	T = 199 * r21002
200:	T = 200 * r21002
201:	T = 201 * r21002
202:	T = 202 * r21002
203:	T = 203 * r21002
204:	T = 204 * r21002
205:	T = 205 * r21002
206:	T = 206 * r21002
207:	T = 207 * r21002
208:	T = 208 * r21002
209:	T = 209 * r21002
210:	T = 210 * r21002
211:	T = 211 * r21002
212:	T = 212 * r21002
213:	T = 213 * r21002
214:	T = 214 * r21002
215:	T = 215 * r21002
216:	T = 216 * r21002
217:	T = 217 * r21002
218:	T = 218 * r21002
219:	T = 219 * r21002
220:	T = 220 * r21002
221:	T = 221 * r21002
222:	T = 222 * r21002
223:	T = 223 * r21002
224:	T = 224 * r21002
225:	T = 225 * r21002
226:	T = 226 * r21002
227:	T = 227 * r21002
228:	T = 228 * r21002
229:	T = 229 * r21002
230:	T = 230 * r21002
231:	T = 231 * r21002
232:	T = 232 * r21002
233:	T = 233 * r21002
234:	T = 234 * r21002
235:	T = 235 * r21002
236:	T = 236 * r21002
237:	T = 237 * r21002
238:	T = 238 * r21002
239:	T = 239 * r21002
240:	T = 240 * r21002
241:	T = 241 * r21002
242:	T = 242 * r21002
243:	T = 243 * r21002
244:	T = 244 * r21002

	245:	T = 245 * r21002
	246:	T = 246 * r21002
	247:	T = 247 * r21002
	248:	T = 248 * r21002
	249:	T = 249 * r21002
	250:	T = 250 * r21002
	251:	T = 251 * r21002
	252:	T = 252 * r21002
	253:	T = 253 * r21002
	254:	T = 254 * r21002
	255:	T = 255 * r21002
	256:	T = 256 * r21002
	1001:	T = 1 * r21003
	1002:	T = 2 * r21003
	1003:	T = 3 * r21003
	1004:	T = 4 * r21003
	1005:	T = 5 * r21003
	1006:	T = 6 * r21003
	1008:	T = 8 * r21003
	1010:	T = 10 * r21003
	1012:	T = 12 * r21003
	1016:	T = 16 * r21003
	1020:	T = 20 * r21003
	1024:	T = 24 * r21003
	1032:	T = 32 * r21003
	1040:	T = 40 * r21003
	1048:	T = 48 * r21003
	1064:	T = 64 * r21003
	1080:	T = 80 * r21003
	1096:	T = 96 * r21003
	2000:	Read-in AFTER digital inputs
	2001:	Output BEFORE digital outputs
	4000:	Receive AFTER IF1 PROFIdrive PZD
	4001:	Send BEFORE IF1 PROFIdrive PZD
	4002:	Receive AFTER IF2 PZD
	4003:	Send BEFORE IF2 PZD
	4004:	Receive AFTER IF1 PROFIdrive flexible PZD
	4005:	Receive AFTER IF2 flexible PZD
Index:	[0]:	Run-time group 1
	[1]:	Run-time group 2
	[2]:	Run-time group 3
	[3]:	Run-time group 4
	[4]:	Run-time group 5
	[5]:	Run-time group 6
	[6]:	Run-time group 7
	[7]:	Run-time group 8
	[8]:	Run-time group 9
	[9]:	Run-time group 10

Caution: The properties of the run-time groups must not be changed during operation as this could result in discontinuous signal transitions.

Note: Re value = 1 ... 256 (free run-time group):

This selection value can only be selected online if the following applies for sampling time T_sample of this run-time group:
1 ms <= T_sample < r21003.
At download, a value that violates this condition is not rejected, but a permissible equivalent value is set automatically and fault F51004 is output.
Re value > 2000 (fixed run-time group):
The fixed run-time groups p21000[x] >= 2000 log on with the sampling time of the associated basic system function, subject to a minimum sampling time of 1 ms. If, as a result of this limit, the actual sampling time deviates from the sampling time of the basic system function, then fault F51005 (during F51006 download) is output. In this case, another run-time group with a sampling time >= 1 ms should be selected. When selecting the fixed run-time groups, a check is not made as to whether the associated system block exists.
Example:
"BEFORE speed setpoint channel" means before function charts 3010, 3020, 3030, 3040, etc. are calculated, if the setpoint channel is activated. If, e.g. for SERVO, a setpoint channel has not been configured (p0108.8 = 0), the calculation is made before function chart 3095.
Re value = 4002, 4003, 4005 (IF2 run-time group):
On devices where IF2 does not exist (D4xx, CU310), when selecting the run-time groups that involve IF2, the corresponding run-time group for IF1 is automatically logged on.

p21000[0...9]	**Run-time group properties / RTG property**		
DC_CTRL	Changeable: -	Calculated: -	Access level: 1
	Data type: Integer16	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	0	4005	[0] 0

Description: Allocates properties to run-time groups 1 to 10.
This property comprises the sampling time and, for p21000[x] >= 2000, the instant of the call within the sampling time.
The index x + 1 of p21000 corresponds to the number of the run-time group:
- p21000[0] is used to set the property of the run-time group 1
...
- p21000[9] is used to set the property of the run-time group 10

Values:	0:	Do not calculate run-time group
	1:	T = 1 * r21002
	2:	T = 2 * r21002
	3:	T = 3 * r21002
	4:	T = 4 * r21002
	5:	T = 5 * r21002
	6:	T = 6 * r21002
	7:	T = 7 * r21002
	8:	T = 8 * r21002
	9:	T = 9 * r21002
	10:	T = 10 * r21002
	11:	T = 11 * r21002
	12:	T = 12 * r21002
	13:	T = 13 * r21002
	14:	T = 14 * r21002
	15:	T = 15 * r21002
	16:	T = 16 * r21002
	17:	T = 17 * r21002
	18:	T = 18 * r21002
	19:	T = 19 * r21002
	20:	T = 20 * r21002
	21:	T = 21 * r21002
	22:	T = 22 * r21002
	23:	T = 23 * r21002
	24:	T = 24 * r21002
	25:	T = 25 * r21002
	26:	T = 26 * r21002
	27:	T = 27 * r21002
	28:	T = 28 * r21002
	29:	T = 29 * r21002
	30:	T = 30 * r21002
	31:	T = 31 * r21002
	32:	T = 32 * r21002
	33:	T = 33 * r21002
	34:	T = 34 * r21002
	35:	T = 35 * r21002
	36:	T = 36 * r21002

37:	T = 37 * r21002
38:	T = 38 * r21002
39:	T = 39 * r21002
40:	T = 40 * r21002
41:	T = 41 * r21002
42:	T = 42 * r21002
43:	T = 43 * r21002
44:	T = 44 * r21002
45:	T = 45 * r21002
46:	T = 46 * r21002
47:	T = 47 * r21002
48:	T = 48 * r21002
49:	T = 49 * r21002
50:	T = 50 * r21002
51:	T = 51 * r21002
52:	T = 52 * r21002
53:	T = 53 * r21002
54:	T = 54 * r21002
55:	T = 55 * r21002
56:	T = 56 * r21002
57:	T = 57 * r21002
58:	T = 58 * r21002
59:	T = 59 * r21002
60:	T = 60 * r21002
61:	T = 61 * r21002
62:	T = 62 * r21002
63:	T = 63 * r21002
64:	T = 64 * r21002
65:	T = 65 * r21002
66:	T = 66 * r21002
67:	T = 67 * r21002
68:	T = 68 * r21002
69:	T = 69 * r21002
70:	T = 70 * r21002
71:	T = 71 * r21002
72:	T = 72 * r21002
73:	T = 73 * r21002
74:	T = 74 * r21002
75:	T = 75 * r21002
76:	T = 76 * r21002
77:	T = 77 * r21002
78:	T = 78 * r21002
79:	T = 79 * r21002
80:	T = 80 * r21002
81:	T = 81 * r21002
82:	T = 82 * r21002
83:	T = 83 * r21002
84:	T = 84 * r21002
85:	T = 85 * r21002
86:	T = 86 * r21002
87:	T = 87 * r21002
88:	T = 88 * r21002

89:	T = 89 * r21002
90:	T = 90 * r21002
91:	T = 91 * r21002
92:	T = 92 * r21002
93:	T = 93 * r21002
94:	T = 94 * r21002
95:	T = 95 * r21002
96:	T = 96 * r21002
97:	T = 97 * r21002
98:	T = 98 * r21002
99:	T = 99 * r21002
100:	T = 100 * r21002
101:	T = 101 * r21002
102:	T = 102 * r21002
103:	T = 103 * r21002
104:	T = 104 * r21002
105:	T = 105 * r21002
106:	T = 106 * r21002
107:	T = 107 * r21002
108:	T = 108 * r21002
109:	T = 109 * r21002
110:	T = 110 * r21002
111:	T = 111 * r21002
112:	T = 112 * r21002
113:	T = 113 * r21002
114:	T = 114 * r21002
115:	T = 115 * r21002
116:	T = 116 * r21002
117:	T = 117 * r21002
118:	T = 118 * r21002
119:	T = 119 * r21002
120:	T = 120 * r21002
121:	T = 121 * r21002
122:	T = 122 * r21002
123:	T = 123 * r21002
124:	T = 124 * r21002
125:	T = 125 * r21002
126:	T = 126 * r21002
127:	T = 127 * r21002
128:	T = 128 * r21002
129:	T = 129 * r21002
130:	T = 130 * r21002
131:	T = 131 * r21002
132:	T = 132 * r21002
133:	T = 133 * r21002
134:	T = 134 * r21002
135:	T = 135 * r21002
136:	T = 136 * r21002
137:	T = 137 * r21002
138:	T = 138 * r21002
139:	T = 139 * r21002
140:	T = 140 * r21002

141:	T = 141 * r21002
142:	T = 142 * r21002
143:	T = 143 * r21002
144:	T = 144 * r21002
145:	T = 145 * r21002
146:	T = 146 * r21002
147:	T = 147 * r21002
148:	T = 148 * r21002
149:	T = 149 * r21002
150:	T = 150 * r21002
151:	T = 151 * r21002
152:	T = 152 * r21002
153:	T = 153 * r21002
154:	T = 154 * r21002
155:	T = 155 * r21002
156:	T = 156 * r21002
157:	T = 157 * r21002
158:	T = 158 * r21002
159:	T = 159 * r21002
160:	T = 160 * r21002
161:	T = 161 * r21002
162:	T = 162 * r21002
163:	T = 163 * r21002
164:	T = 164 * r21002
165:	T = 165 * r21002
166:	T = 166 * r21002
167:	T = 167 * r21002
168:	T = 168 * r21002
169:	T = 169 * r21002
170:	T = 170 * r21002
171:	T = 171 * r21002
172:	T = 172 * r21002
173:	T = 173 * r21002
174:	T = 174 * r21002
175:	T = 175 * r21002
176:	T = 176 * r21002
177:	T = 177 * r21002
178:	T = 178 * r21002
179:	T = 179 * r21002
180:	T = 180 * r21002
181:	T = 181 * r21002
182:	T = 182 * r21002
183:	T = 183 * r21002
184:	T = 184 * r21002
185:	T = 185 * r21002
186:	T = 186 * r21002
187:	T = 187 * r21002
188:	T = 188 * r21002
189:	T = 189 * r21002
190:	T = 190 * r21002
191:	T = 191 * r21002
192:	T = 192 * r21002

193:	T = 193 * r21002
194:	T = 194 * r21002
195:	T = 195 * r21002
196:	T = 196 * r21002
197:	T = 197 * r21002
198:	T = 198 * r21002
199:	T = 199 * r21002
200:	T = 200 * r21002
201:	T = 201 * r21002
202:	T = 202 * r21002
203:	T = 203 * r21002
204:	T = 204 * r21002
205:	T = 205 * r21002
206:	T = 206 * r21002
207:	T = 207 * r21002
208:	T = 208 * r21002
209:	T = 209 * r21002
210:	T = 210 * r21002
211:	T = 211 * r21002
212:	T = 212 * r21002
213:	T = 213 * r21002
214:	T = 214 * r21002
215:	T = 215 * r21002
216:	T = 216 * r21002
217:	T = 217 * r21002
218:	T = 218 * r21002
219:	T = 219 * r21002
220:	T = 220 * r21002
221:	T = 221 * r21002
222:	T = 222 * r21002
223:	T = 223 * r21002
224:	T = 224 * r21002
225:	T = 225 * r21002
226:	T = 226 * r21002
227:	T = 227 * r21002
228:	T = 228 * r21002
229:	T = 229 * r21002
230:	T = 230 * r21002
231:	T = 231 * r21002
232:	T = 232 * r21002
233:	T = 233 * r21002
234:	T = 234 * r21002
235:	T = 235 * r21002
236:	T = 236 * r21002
237:	T = 237 * r21002
238:	T = 238 * r21002
239:	T = 239 * r21002
240:	T = 240 * r21002
241:	T = 241 * r21002
242:	T = 242 * r21002
243:	T = 243 * r21002
244:	T = 244 * r21002

	245:	T = 245 * r21002
	246:	T = 246 * r21002
	247:	T = 247 * r21002
	248:	T = 248 * r21002
	249:	T = 249 * r21002
	250:	T = 250 * r21002
	251:	T = 251 * r21002
	252:	T = 252 * r21002
	253:	T = 253 * r21002
	254:	T = 254 * r21002
	255:	T = 255 * r21002
	256:	T = 256 * r21002
	1001:	T = 1 * r21003
	1002:	T = 2 * r21003
	1003:	T = 3 * r21003
	1004:	T = 4 * r21003
	1005:	T = 5 * r21003
	1006:	T = 6 * r21003
	1008:	T = 8 * r21003
	1010:	T = 10 * r21003
	1012:	T = 12 * r21003
	1016:	T = 16 * r21003
	1020:	T = 20 * r21003
	1024:	T = 24 * r21003
	1032:	T = 32 * r21003
	1040:	T = 40 * r21003
	1048:	T = 48 * r21003
	1064:	T = 64 * r21003
	1080:	T = 80 * r21003
	1096:	T = 96 * r21003
	3001:	BEFORE speed ctrl
	3003:	BEFORE speed setpoint channel
	4000:	Receive AFTER IF1 PROFIdrive PZD
	4001:	Send BEFORE IF1 PROFIdrive PZD
	4002:	Receive AFTER IF2 PZD
	4003:	Send BEFORE IF2 PZD
	4004:	Receive AFTER IF1 PROFIdrive flexible PZD
	4005:	Receive AFTER IF2 flexible PZD
Index:	[0]:	Run-time group 1
	[1]:	Run-time group 2
	[2]:	Run-time group 3
	[3]:	Run-time group 4
	[4]:	Run-time group 5
	[5]:	Run-time group 6
	[6]:	Run-time group 7
	[7]:	Run-time group 8
	[8]:	Run-time group 9
	[9]:	Run-time group 10

Caution: The properties of the run-time groups must not be changed during operation as this could result in discontinuous signal transitions.

Note: Re value = 1 ... 256 (free run-time group):

This selection value can only be selected online if the following applies for sampling time T_sample of this run-time group:
1 ms <= T_sample < r21003.
At download, a value that violates this condition is not rejected, but a permissible equivalent value is set automatically and fault F51004 is output.
Re value > 2000 (fixed run-time group):
The fixed run-time groups p21000[x] >= 2000 log on with the sampling time of the associated basic system function, subject to a minimum sampling time of 1 ms. If, as a result of this limit, the actual sampling time deviates from the sampling time of the basic system function, then fault F51005 (during F51006 download) is output. In this case, another run-time group with a sampling time >= 1 ms should be selected. When selecting the fixed run-time groups, a check is not made as to whether the associated system block exists.
Example:
"BEFORE speed setpoint channel" means before function charts 3010, 3020, 3030, 3040, etc. are calculated, if the setpoint channel is activated. If, e.g. for SERVO, a setpoint channel has not been configured (p0108.8 = 0), the calculation is made before function chart 3095.
Re value = 4002, 4003, 4005 (IF2 run-time group):
On devices where IF2 does not exist (D4xx, CU310), when selecting the run-time groups that involve IF2, the corresponding run-time group for IF1 is automatically logged on.

p21000[0...9]	**Run-time group properties / RTG property**		
INFEED, TM120	Changeable: -	Calculated: -	Access level: 1
	Data type: Integer16	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	0	4004	[0] 0

Description: Allocates properties to run-time groups 1 to 10.
This property comprises the sampling time and, for p21000[x] >= 2000, the instant of the call within the sampling time.
The index x + 1 of p21000 corresponds to the number of the run-time group:
- p21000[0] is used to set the property of the run-time group 1
...
- p21000[9] is used to set the property of the run-time group 10

Values:		
	0:	Do not calculate run-time group
	1:	T = 1 * r21002
	2:	T = 2 * r21002
	3:	T = 3 * r21002
	4:	T = 4 * r21002
	5:	T = 5 * r21002
	6:	T = 6 * r21002
	7:	T = 7 * r21002
	8:	T = 8 * r21002
	9:	T = 9 * r21002
	10:	T = 10 * r21002
	11:	T = 11 * r21002
	12:	T = 12 * r21002
	13:	T = 13 * r21002
	14:	T = 14 * r21002
	15:	T = 15 * r21002
	16:	T = 16 * r21002
	17:	T = 17 * r21002
	18:	T = 18 * r21002
	19:	T = 19 * r21002
	20:	T = 20 * r21002
	21:	T = 21 * r21002
	22:	T = 22 * r21002
	23:	T = 23 * r21002
	24:	T = 24 * r21002
	25:	T = 25 * r21002
	26:	T = 26 * r21002
	27:	T = 27 * r21002
	28:	T = 28 * r21002
	29:	T = 29 * r21002
	30:	T = 30 * r21002
	31:	T = 31 * r21002
	32:	T = 32 * r21002
	33:	T = 33 * r21002
	34:	T = 34 * r21002
	35:	T = 35 * r21002
	36:	T = 36 * r21002

37:	T = 37 * r21002
38:	T = 38 * r21002
39:	T = 39 * r21002
40:	T = 40 * r21002
41:	T = 41 * r21002
42:	T = 42 * r21002
43:	T = 43 * r21002
44:	T = 44 * r21002
45:	T = 45 * r21002
46:	T = 46 * r21002
47:	T = 47 * r21002
48:	T = 48 * r21002
49:	T = 49 * r21002
50:	T = 50 * r21002
51:	T = 51 * r21002
52:	T = 52 * r21002
53:	T = 53 * r21002
54:	T = 54 * r21002
55:	T = 55 * r21002
56:	T = 56 * r21002
57:	T = 57 * r21002
58:	T = 58 * r21002
59:	T = 59 * r21002
60:	T = 60 * r21002
61:	T = 61 * r21002
62:	T = 62 * r21002
63:	T = 63 * r21002
64:	T = 64 * r21002
65:	T = 65 * r21002
66:	T = 66 * r21002
67:	T = 67 * r21002
68:	T = 68 * r21002
69:	T = 69 * r21002
70:	T = 70 * r21002
71:	T = 71 * r21002
72:	T = 72 * r21002
73:	T = 73 * r21002
74:	T = 74 * r21002
75:	T = 75 * r21002
76:	T = 76 * r21002
77:	T = 77 * r21002
78:	T = 78 * r21002
79:	T = 79 * r21002
80:	T = 80 * r21002
81:	T = 81 * r21002
82:	T = 82 * r21002
83:	T = 83 * r21002
84:	T = 84 * r21002
85:	T = 85 * r21002
86:	T = 86 * r21002
87:	T = 87 * r21002
88:	T = 88 * r21002

89:	T = 89 * r21002
90:	T = 90 * r21002
91:	T = 91 * r21002
92:	T = 92 * r21002
93:	T = 93 * r21002
94:	T = 94 * r21002
95:	T = 95 * r21002
96:	T = 96 * r21002
97:	T = 97 * r21002
98:	T = 98 * r21002
99:	T = 99 * r21002
100:	T = 100 * r21002
101:	T = 101 * r21002
102:	T = 102 * r21002
103:	T = 103 * r21002
104:	T = 104 * r21002
105:	T = 105 * r21002
106:	T = 106 * r21002
107:	T = 107 * r21002
108:	T = 108 * r21002
109:	T = 109 * r21002
110:	T = 110 * r21002
111:	T = 111 * r21002
112:	T = 112 * r21002
113:	T = 113 * r21002
114:	T = 114 * r21002
115:	T = 115 * r21002
116:	T = 116 * r21002
117:	T = 117 * r21002
118:	T = 118 * r21002
119:	T = 119 * r21002
120:	T = 120 * r21002
121:	T = 121 * r21002
122:	T = 122 * r21002
123:	T = 123 * r21002
124:	T = 124 * r21002
125:	T = 125 * r21002
126:	T = 126 * r21002
127:	T = 127 * r21002
128:	T = 128 * r21002
129:	T = 129 * r21002
130:	T = 130 * r21002
131:	T = 131 * r21002
132:	T = 132 * r21002
133:	T = 133 * r21002
134:	T = 134 * r21002
135:	T = 135 * r21002
136:	T = 136 * r21002
137:	T = 137 * r21002
138:	T = 138 * r21002
139:	T = 139 * r21002
140:	T = 140 * r21002

141:	T = 141 * r21002
142:	T = 142 * r21002
143:	T = 143 * r21002
144:	T = 144 * r21002
145:	T = 145 * r21002
146:	T = 146 * r21002
147:	T = 147 * r21002
148:	T = 148 * r21002
149:	T = 149 * r21002
150:	T = 150 * r21002
151:	T = 151 * r21002
152:	T = 152 * r21002
153:	T = 153 * r21002
154:	T = 154 * r21002
155:	T = 155 * r21002
156:	T = 156 * r21002
157:	T = 157 * r21002
158:	T = 158 * r21002
159:	T = 159 * r21002
160:	T = 160 * r21002
161:	T = 161 * r21002
162:	T = 162 * r21002
163:	T = 163 * r21002
164:	T = 164 * r21002
165:	T = 165 * r21002
166:	T = 166 * r21002
167:	T = 167 * r21002
168:	T = 168 * r21002
169:	T = 169 * r21002
170:	T = 170 * r21002
171:	T = 171 * r21002
172:	T = 172 * r21002
173:	T = 173 * r21002
174:	T = 174 * r21002
175:	T = 175 * r21002
176:	T = 176 * r21002
177:	T = 177 * r21002
178:	T = 178 * r21002
179:	T = 179 * r21002
180:	T = 180 * r21002
181:	T = 181 * r21002
182:	T = 182 * r21002
183:	T = 183 * r21002
184:	T = 184 * r21002
185:	T = 185 * r21002
186:	T = 186 * r21002
187:	T = 187 * r21002
188:	T = 188 * r21002
189:	T = 189 * r21002
190:	T = 190 * r21002
191:	T = 191 * r21002
192:	T = 192 * r21002

193:	T = 193 * r21002
194:	T = 194 * r21002
195:	T = 195 * r21002
196:	T = 196 * r21002
197:	T = 197 * r21002
198:	T = 198 * r21002
199:	T = 199 * r21002
200:	T = 200 * r21002
201:	T = 201 * r21002
202:	T = 202 * r21002
203:	T = 203 * r21002
204:	T = 204 * r21002
205:	T = 205 * r21002
206:	T = 206 * r21002
207:	T = 207 * r21002
208:	T = 208 * r21002
209:	T = 209 * r21002
210:	T = 210 * r21002
211:	T = 211 * r21002
212:	T = 212 * r21002
213:	T = 213 * r21002
214:	T = 214 * r21002
215:	T = 215 * r21002
216:	T = 216 * r21002
217:	T = 217 * r21002
218:	T = 218 * r21002
219:	T = 219 * r21002
220:	T = 220 * r21002
221:	T = 221 * r21002
222:	T = 222 * r21002
223:	T = 223 * r21002
224:	T = 224 * r21002
225:	T = 225 * r21002
226:	T = 226 * r21002
227:	T = 227 * r21002
228:	T = 228 * r21002
229:	T = 229 * r21002
230:	T = 230 * r21002
231:	T = 231 * r21002
232:	T = 232 * r21002
233:	T = 233 * r21002
234:	T = 234 * r21002
235:	T = 235 * r21002
236:	T = 236 * r21002
237:	T = 237 * r21002
238:	T = 238 * r21002
239:	T = 239 * r21002
240:	T = 240 * r21002
241:	T = 241 * r21002
242:	T = 242 * r21002
243:	T = 243 * r21002
244:	T = 244 * r21002

	245:	T = 245 * r21002
	246:	T = 246 * r21002
	247:	T = 247 * r21002
	248:	T = 248 * r21002
	249:	T = 249 * r21002
	250:	T = 250 * r21002
	251:	T = 251 * r21002
	252:	T = 252 * r21002
	253:	T = 253 * r21002
	254:	T = 254 * r21002
	255:	T = 255 * r21002
	256:	T = 256 * r21002
	1001:	T = 1 * r21003
	1002:	T = 2 * r21003
	1003:	T = 3 * r21003
	1004:	T = 4 * r21003
	1005:	T = 5 * r21003
	1006:	T = 6 * r21003
	1008:	T = 8 * r21003
	1010:	T = 10 * r21003
	1012:	T = 12 * r21003
	1016:	T = 16 * r21003
	1020:	T = 20 * r21003
	1024:	T = 24 * r21003
	1032:	T = 32 * r21003
	1040:	T = 40 * r21003
	1048:	T = 48 * r21003
	1064:	T = 64 * r21003
	1080:	T = 80 * r21003
	1096:	T = 96 * r21003
	4000:	Receive AFTER IF1 PROFIdrive PZD
	4001:	Send BEFORE IF1 PROFIdrive PZD
	4004:	Receive AFTER IF1 PROFIdrive flexible PZD
Index:	[0]:	Run-time group 1
	[1]:	Run-time group 2
	[2]:	Run-time group 3
	[3]:	Run-time group 4
	[4]:	Run-time group 5
	[5]:	Run-time group 6
	[6]:	Run-time group 7
	[7]:	Run-time group 8
	[8]:	Run-time group 9
	[9]:	Run-time group 10

Caution: The properties of the run-time groups must not be changed during operation as this could result in discontinuous signal transitions.

Note: Re value = 1 ... 256 (free run-time group):
This selection value can only be selected online if the following applies for sampling time T_sample of this run-time group:
1 ms <= T_sample < r21003.
At download, a value that violates this condition is not rejected, but a permissible equivalent value is set automatically and fault F51004 is output.

Re value > 2000 (fixed run-time group):
The fixed run-time groups p21000[x] >= 2000 log on with the sampling time of the associated basic system function, subject to a minimum sampling time of 1 ms. If, as a result of this limit, the actual sampling time deviates from the sampling time of the basic system function, then fault F51005 (during F51006 download) is output. In this case, another run-time group with a sampling time >= 1 ms should be selected. When selecting the fixed run-time groups, a check is not made as to whether the associated system block exists.
Example:
"BEFORE speed setpoint channel" means before function charts 3010, 3020, 3030, 3040, etc. are calculated, if the setpoint channel is activated. If, e.g. for SERVO, a setpoint channel has not been configured (p0108.8 = 0), the calculation is made before function chart 3095.

p21000[0...9]	**Run-time group properties / RTG property**		
SERVO, VECTOR	Changeable: -	Calculated: -	Access level: 1
	Data type: Integer16	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	0	4005	[0] 0

Description: Allocates properties to run-time groups 1 to 10.
This property comprises the sampling time and, for p21000[x] >= 2000, the instant of the call within the sampling time.
The index x + 1 of p21000 corresponds to the number of the run-time group:
- p21000[0] is used to set the property of the run-time group 1
...
- p21000[9] is used to set the property of the run-time group 10

Values:	0:	Do not calculate run-time group
	1:	T = 1 * r21002
	2:	T = 2 * r21002
	3:	T = 3 * r21002
	4:	T = 4 * r21002
	5:	T = 5 * r21002
	6:	T = 6 * r21002
	7:	T = 7 * r21002
	8:	T = 8 * r21002
	9:	T = 9 * r21002
	10:	T = 10 * r21002
	11:	T = 11 * r21002
	12:	T = 12 * r21002
	13:	T = 13 * r21002
	14:	T = 14 * r21002
	15:	T = 15 * r21002
	16:	T = 16 * r21002
	17:	T = 17 * r21002
	18:	T = 18 * r21002
	19:	T = 19 * r21002
	20:	T = 20 * r21002
	21:	T = 21 * r21002
	22:	T = 22 * r21002
	23:	T = 23 * r21002
	24:	T = 24 * r21002
	25:	T = 25 * r21002
	26:	T = 26 * r21002
	27:	T = 27 * r21002
	28:	T = 28 * r21002
	29:	T = 29 * r21002
	30:	T = 30 * r21002
	31:	T = 31 * r21002
	32:	T = 32 * r21002
	33:	T = 33 * r21002
	34:	T = 34 * r21002
	35:	T = 35 * r21002
	36:	T = 36 * r21002

37:	T = 37 * r21002
38:	T = 38 * r21002
39:	T = 39 * r21002
40:	T = 40 * r21002
41:	T = 41 * r21002
42:	T = 42 * r21002
43:	T = 43 * r21002
44:	T = 44 * r21002
45:	T = 45 * r21002
46:	T = 46 * r21002
47:	T = 47 * r21002
48:	T = 48 * r21002
49:	T = 49 * r21002
50:	T = 50 * r21002
51:	T = 51 * r21002
52:	T = 52 * r21002
53:	T = 53 * r21002
54:	T = 54 * r21002
55:	T = 55 * r21002
56:	T = 56 * r21002
57:	T = 57 * r21002
58:	T = 58 * r21002
59:	T = 59 * r21002
60:	T = 60 * r21002
61:	T = 61 * r21002
62:	T = 62 * r21002
63:	T = 63 * r21002
64:	T = 64 * r21002
65:	T = 65 * r21002
66:	T = 66 * r21002
67:	T = 67 * r21002
68:	T = 68 * r21002
69:	T = 69 * r21002
70:	T = 70 * r21002
71:	T = 71 * r21002
72:	T = 72 * r21002
73:	T = 73 * r21002
74:	T = 74 * r21002
75:	T = 75 * r21002
76:	T = 76 * r21002
77:	T = 77 * r21002
78:	T = 78 * r21002
79:	T = 79 * r21002
80:	T = 80 * r21002
81:	T = 81 * r21002
82:	T = 82 * r21002
83:	T = 83 * r21002
84:	T = 84 * r21002
85:	T = 85 * r21002
86:	T = 86 * r21002
87:	T = 87 * r21002
88:	T = 88 * r21002

89:	T = 89 * r21002
90:	T = 90 * r21002
91:	T = 91 * r21002
92:	T = 92 * r21002
93:	T = 93 * r21002
94:	T = 94 * r21002
95:	T = 95 * r21002
96:	T = 96 * r21002
97:	T = 97 * r21002
98:	T = 98 * r21002
99:	T = 99 * r21002
100:	T = 100 * r21002
101:	T = 101 * r21002
102:	T = 102 * r21002
103:	T = 103 * r21002
104:	T = 104 * r21002
105:	T = 105 * r21002
106:	T = 106 * r21002
107:	T = 107 * r21002
108:	T = 108 * r21002
109:	T = 109 * r21002
110:	T = 110 * r21002
111:	T = 111 * r21002
112:	T = 112 * r21002
113:	T = 113 * r21002
114:	T = 114 * r21002
115:	T = 115 * r21002
116:	T = 116 * r21002
117:	T = 117 * r21002
118:	T = 118 * r21002
119:	T = 119 * r21002
120:	T = 120 * r21002
121:	T = 121 * r21002
122:	T = 122 * r21002
123:	T = 123 * r21002
124:	T = 124 * r21002
125:	T = 125 * r21002
126:	T = 126 * r21002
127:	T = 127 * r21002
128:	T = 128 * r21002
129:	T = 129 * r21002
130:	T = 130 * r21002
131:	T = 131 * r21002
132:	T = 132 * r21002
133:	T = 133 * r21002
134:	T = 134 * r21002
135:	T = 135 * r21002
136:	T = 136 * r21002
137:	T = 137 * r21002
138:	T = 138 * r21002
139:	T = 139 * r21002
140:	T = 140 * r21002

141:	T = 141 * r21002
142:	T = 142 * r21002
143:	T = 143 * r21002
144:	T = 144 * r21002
145:	T = 145 * r21002
146:	T = 146 * r21002
147:	T = 147 * r21002
148:	T = 148 * r21002
149:	T = 149 * r21002
150:	T = 150 * r21002
151:	T = 151 * r21002
152:	T = 152 * r21002
153:	T = 153 * r21002
154:	T = 154 * r21002
155:	T = 155 * r21002
156:	T = 156 * r21002
157:	T = 157 * r21002
158:	T = 158 * r21002
159:	T = 159 * r21002
160:	T = 160 * r21002
161:	T = 161 * r21002
162:	T = 162 * r21002
163:	T = 163 * r21002
164:	T = 164 * r21002
165:	T = 165 * r21002
166:	T = 166 * r21002
167:	T = 167 * r21002
168:	T = 168 * r21002
169:	T = 169 * r21002
170:	T = 170 * r21002
171:	T = 171 * r21002
172:	T = 172 * r21002
173:	T = 173 * r21002
174:	T = 174 * r21002
175:	T = 175 * r21002
176:	T = 176 * r21002
177:	T = 177 * r21002
178:	T = 178 * r21002
179:	T = 179 * r21002
180:	T = 180 * r21002
181:	T = 181 * r21002
182:	T = 182 * r21002
183:	T = 183 * r21002
184:	T = 184 * r21002
185:	T = 185 * r21002
186:	T = 186 * r21002
187:	T = 187 * r21002
188:	T = 188 * r21002
189:	T = 189 * r21002
190:	T = 190 * r21002
191:	T = 191 * r21002
192:	T = 192 * r21002

193:	T = 193 * r21002
194:	T = 194 * r21002
195:	T = 195 * r21002
196:	T = 196 * r21002
197:	T = 197 * r21002
198:	T = 198 * r21002
199:	T = 199 * r21002
200:	T = 200 * r21002
201:	T = 201 * r21002
202:	T = 202 * r21002
203:	T = 203 * r21002
204:	T = 204 * r21002
205:	T = 205 * r21002
206:	T = 206 * r21002
207:	T = 207 * r21002
208:	T = 208 * r21002
209:	T = 209 * r21002
210:	T = 210 * r21002
211:	T = 211 * r21002
212:	T = 212 * r21002
213:	T = 213 * r21002
214:	T = 214 * r21002
215:	T = 215 * r21002
216:	T = 216 * r21002
217:	T = 217 * r21002
218:	T = 218 * r21002
219:	T = 219 * r21002
220:	T = 220 * r21002
221:	T = 221 * r21002
222:	T = 222 * r21002
223:	T = 223 * r21002
224:	T = 224 * r21002
225:	T = 225 * r21002
226:	T = 226 * r21002
227:	T = 227 * r21002
228:	T = 228 * r21002
229:	T = 229 * r21002
230:	T = 230 * r21002
231:	T = 231 * r21002
232:	T = 232 * r21002
233:	T = 233 * r21002
234:	T = 234 * r21002
235:	T = 235 * r21002
236:	T = 236 * r21002
237:	T = 237 * r21002
238:	T = 238 * r21002
239:	T = 239 * r21002
240:	T = 240 * r21002
241:	T = 241 * r21002
242:	T = 242 * r21002
243:	T = 243 * r21002
244:	T = 244 * r21002

	245:	T = 245 * r21002
	246:	T = 246 * r21002
	247:	T = 247 * r21002
	248:	T = 248 * r21002
	249:	T = 249 * r21002
	250:	T = 250 * r21002
	251:	T = 251 * r21002
	252:	T = 252 * r21002
	253:	T = 253 * r21002
	254:	T = 254 * r21002
	255:	T = 255 * r21002
	256:	T = 256 * r21002
	1001:	T = 1 * r21003
	1002:	T = 2 * r21003
	1003:	T = 3 * r21003
	1004:	T = 4 * r21003
	1005:	T = 5 * r21003
	1006:	T = 6 * r21003
	1008:	T = 8 * r21003
	1010:	T = 10 * r21003
	1012:	T = 12 * r21003
	1016:	T = 16 * r21003
	1020:	T = 20 * r21003
	1024:	T = 24 * r21003
	1032:	T = 32 * r21003
	1040:	T = 40 * r21003
	1048:	T = 48 * r21003
	1064:	T = 64 * r21003
	1080:	T = 80 * r21003
	1096:	T = 96 * r21003
	3001:	BEFORE speed ctrl
	3003:	BEFORE speed setpoint channel
	3004:	BEFORE pos ctrl
	3005:	BEFORE basic positioner
	3006:	BEFORE standard technology controller
	3007:	BEFORE act p v
	4000:	Receive AFTER IF1 PROFIdrive PZD
	4001:	Send BEFORE IF1 PROFIdrive PZD
	4002:	Receive AFTER IF2 PZD
	4003:	Send BEFORE IF2 PZD
	4004:	Receive AFTER IF1 PROFIdrive flexible PZD
	4005:	Receive AFTER IF2 flexible PZD
Index:	[0]:	Run-time group 1
	[1]:	Run-time group 2
	[2]:	Run-time group 3
	[3]:	Run-time group 4
	[4]:	Run-time group 5
	[5]:	Run-time group 6
	[6]:	Run-time group 7
	[7]:	Run-time group 8
	[8]:	Run-time group 9
	[9]:	Run-time group 10

Caution: The properties of the run-time groups must not be changed during operation as this could result in discontinuous signal transitions.

Note: Re value = 1 ... 256 (free run-time group):

This selection value can only be selected online if the following applies for sampling time T_sample of this run-time group:

1 ms <= T_sample < r21003.

At download, a value that violates this condition is not rejected, but a permissible equivalent value is set automatically and fault F51004 is output.

Re value > 2000 (fixed run-time group):

The fixed run-time groups p21000[x] >= 2000 log on with the sampling time of the associated basic system function, subject to a minimum sampling time of 1 ms. If, as a result of this limit, the actual sampling time deviates from the sampling time of the basic system function, then fault F51005 (during F51006 download) is output. In this case, another run-time group with a sampling time >= 1 ms should be selected. When selecting the fixed run-time groups, a check is not made as to whether the associated system block exists.

Example:

"BEFORE speed setpoint channel" means before function charts 3010, 3020, 3030, 3040, etc. are calculated, if the setpoint channel is activated. If, e.g. for SERVO, a setpoint channel has not been configured (p0108.8 = 0), the calculation is made before function chart 3095.

Re value = 4002, 4003, 4005 (IF2 run-time group):

On devices where IF2 does not exist (D4xx, CU310), when selecting the run-time groups that involve IF2, the corresponding run-time group for IF1 is automatically logged on.

p21000[0...9]	**Run-time group properties / RTG property**		
SERVO (Dig IO)	Changeable: -	Calculated: -	Access level: 1
	Data type: Integer16	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	0	4005	[0] 0

Description: Allocates properties to run-time groups 1 to 10.
This property comprises the sampling time and, for p21000[x] >= 2000, the instant of the call within the sampling time.
The index x + 1 of p21000 corresponds to the number of the run-time group:
- p21000[0] is used to set the property of the run-time group 1
...
- p21000[9] is used to set the property of the run-time group 10

Values:	0:	Do not calculate run-time group
	1:	T = 1 * r21002
	2:	T = 2 * r21002
	3:	T = 3 * r21002
	4:	T = 4 * r21002
	5:	T = 5 * r21002
	6:	T = 6 * r21002
	7:	T = 7 * r21002
	8:	T = 8 * r21002
	9:	T = 9 * r21002
	10:	T = 10 * r21002
	11:	T = 11 * r21002
	12:	T = 12 * r21002
	13:	T = 13 * r21002
	14:	T = 14 * r21002
	15:	T = 15 * r21002
	16:	T = 16 * r21002
	17:	T = 17 * r21002
	18:	T = 18 * r21002
	19:	T = 19 * r21002
	20:	T = 20 * r21002
	21:	T = 21 * r21002
	22:	T = 22 * r21002
	23:	T = 23 * r21002
	24:	T = 24 * r21002
	25:	T = 25 * r21002
	26:	T = 26 * r21002
	27:	T = 27 * r21002
	28:	T = 28 * r21002
	29:	T = 29 * r21002
	30:	T = 30 * r21002
	31:	T = 31 * r21002
	32:	T = 32 * r21002
	33:	T = 33 * r21002
	34:	T = 34 * r21002
	35:	T = 35 * r21002
	36:	T = 36 * r21002

37:	T = 37 * r21002
38:	T = 38 * r21002
39:	T = 39 * r21002
40:	T = 40 * r21002
41:	T = 41 * r21002
42:	T = 42 * r21002
43:	T = 43 * r21002
44:	T = 44 * r21002
45:	T = 45 * r21002
46:	T = 46 * r21002
47:	T = 47 * r21002
48:	T = 48 * r21002
49:	T = 49 * r21002
50:	T = 50 * r21002
51:	T = 51 * r21002
52:	T = 52 * r21002
53:	T = 53 * r21002
54:	T = 54 * r21002
55:	T = 55 * r21002
56:	T = 56 * r21002
57:	T = 57 * r21002
58:	T = 58 * r21002
59:	T = 59 * r21002
60:	T = 60 * r21002
61:	T = 61 * r21002
62:	T = 62 * r21002
63:	T = 63 * r21002
64:	T = 64 * r21002
65:	T = 65 * r21002
66:	T = 66 * r21002
67:	T = 67 * r21002
68:	T = 68 * r21002
69:	T = 69 * r21002
70:	T = 70 * r21002
71:	T = 71 * r21002
72:	T = 72 * r21002
73:	T = 73 * r21002
74:	T = 74 * r21002
75:	T = 75 * r21002
76:	T = 76 * r21002
77:	T = 77 * r21002
78:	T = 78 * r21002
79:	T = 79 * r21002
80:	T = 80 * r21002
81:	T = 81 * r21002
82:	T = 82 * r21002
83:	T = 83 * r21002
84:	T = 84 * r21002
85:	T = 85 * r21002
86:	T = 86 * r21002
87:	T = 87 * r21002
88:	T = 88 * r21002

89:	T = 89 * r21002
90:	T = 90 * r21002
91:	T = 91 * r21002
92:	T = 92 * r21002
93:	T = 93 * r21002
94:	T = 94 * r21002
95:	T = 95 * r21002
96:	T = 96 * r21002
97:	T = 97 * r21002
98:	T = 98 * r21002
99:	T = 99 * r21002
100:	T = 100 * r21002
101:	T = 101 * r21002
102:	T = 102 * r21002
103:	T = 103 * r21002
104:	T = 104 * r21002
105:	T = 105 * r21002
106:	T = 106 * r21002
107:	T = 107 * r21002
108:	T = 108 * r21002
109:	T = 109 * r21002
110:	T = 110 * r21002
111:	T = 111 * r21002
112:	T = 112 * r21002
113:	T = 113 * r21002
114:	T = 114 * r21002
115:	T = 115 * r21002
116:	T = 116 * r21002
117:	T = 117 * r21002
118:	T = 118 * r21002
119:	T = 119 * r21002
120:	T = 120 * r21002
121:	T = 121 * r21002
122:	T = 122 * r21002
123:	T = 123 * r21002
124:	T = 124 * r21002
125:	T = 125 * r21002
126:	T = 126 * r21002
127:	T = 127 * r21002
128:	T = 128 * r21002
129:	T = 129 * r21002
130:	T = 130 * r21002
131:	T = 131 * r21002
132:	T = 132 * r21002
133:	T = 133 * r21002
134:	T = 134 * r21002
135:	T = 135 * r21002
136:	T = 136 * r21002
137:	T = 137 * r21002
138:	T = 138 * r21002
139:	T = 139 * r21002
140:	T = 140 * r21002

141:	T = 141 * r21002
142:	T = 142 * r21002
143:	T = 143 * r21002
144:	T = 144 * r21002
145:	T = 145 * r21002
146:	T = 146 * r21002
147:	T = 147 * r21002
148:	T = 148 * r21002
149:	T = 149 * r21002
150:	T = 150 * r21002
151:	T = 151 * r21002
152:	T = 152 * r21002
153:	T = 153 * r21002
154:	T = 154 * r21002
155:	T = 155 * r21002
156:	T = 156 * r21002
157:	T = 157 * r21002
158:	T = 158 * r21002
159:	T = 159 * r21002
160:	T = 160 * r21002
161:	T = 161 * r21002
162:	T = 162 * r21002
163:	T = 163 * r21002
164:	T = 164 * r21002
165:	T = 165 * r21002
166:	T = 166 * r21002
167:	T = 167 * r21002
168:	T = 168 * r21002
169:	T = 169 * r21002
170:	T = 170 * r21002
171:	T = 171 * r21002
172:	T = 172 * r21002
173:	T = 173 * r21002
174:	T = 174 * r21002
175:	T = 175 * r21002
176:	T = 176 * r21002
177:	T = 177 * r21002
178:	T = 178 * r21002
179:	T = 179 * r21002
180:	T = 180 * r21002
181:	T = 181 * r21002
182:	T = 182 * r21002
183:	T = 183 * r21002
184:	T = 184 * r21002
185:	T = 185 * r21002
186:	T = 186 * r21002
187:	T = 187 * r21002
188:	T = 188 * r21002
189:	T = 189 * r21002
190:	T = 190 * r21002
191:	T = 191 * r21002
192:	T = 192 * r21002

193:	T = 193 * r21002
194:	T = 194 * r21002
195:	T = 195 * r21002
196:	T = 196 * r21002
197:	T = 197 * r21002
198:	T = 198 * r21002
199:	T = 199 * r21002
200:	T = 200 * r21002
201:	T = 201 * r21002
202:	T = 202 * r21002
203:	T = 203 * r21002
204:	T = 204 * r21002
205:	T = 205 * r21002
206:	T = 206 * r21002
207:	T = 207 * r21002
208:	T = 208 * r21002
209:	T = 209 * r21002
210:	T = 210 * r21002
211:	T = 211 * r21002
212:	T = 212 * r21002
213:	T = 213 * r21002
214:	T = 214 * r21002
215:	T = 215 * r21002
216:	T = 216 * r21002
217:	T = 217 * r21002
218:	T = 218 * r21002
219:	T = 219 * r21002
220:	T = 220 * r21002
221:	T = 221 * r21002
222:	T = 222 * r21002
223:	T = 223 * r21002
224:	T = 224 * r21002
225:	T = 225 * r21002
226:	T = 226 * r21002
227:	T = 227 * r21002
228:	T = 228 * r21002
229:	T = 229 * r21002
230:	T = 230 * r21002
231:	T = 231 * r21002
232:	T = 232 * r21002
233:	T = 233 * r21002
234:	T = 234 * r21002
235:	T = 235 * r21002
236:	T = 236 * r21002
237:	T = 237 * r21002
238:	T = 238 * r21002
239:	T = 239 * r21002
240:	T = 240 * r21002
241:	T = 241 * r21002
242:	T = 242 * r21002
243:	T = 243 * r21002
244:	T = 244 * r21002

	245:	T = 245 * r21002
	246:	T = 246 * r21002
	247:	T = 247 * r21002
	248:	T = 248 * r21002
	249:	T = 249 * r21002
	250:	T = 250 * r21002
	251:	T = 251 * r21002
	252:	T = 252 * r21002
	253:	T = 253 * r21002
	254:	T = 254 * r21002
	255:	T = 255 * r21002
	256:	T = 256 * r21002
	1001:	T = 1 * r21003
	1002:	T = 2 * r21003
	1003:	T = 3 * r21003
	1004:	T = 4 * r21003
	1005:	T = 5 * r21003
	1006:	T = 6 * r21003
	1008:	T = 8 * r21003
	1010:	T = 10 * r21003
	1012:	T = 12 * r21003
	1016:	T = 16 * r21003
	1020:	T = 20 * r21003
	1024:	T = 24 * r21003
	1032:	T = 32 * r21003
	1040:	T = 40 * r21003
	1048:	T = 48 * r21003
	1064:	T = 64 * r21003
	1080:	T = 80 * r21003
	1096:	T = 96 * r21003
	2000:	Read-in AFTER digital inputs
	2001:	Output BEFORE digital outputs
	3001:	BEFORE speed ctrl
	3003:	BEFORE speed setpoint channel
	3004:	BEFORE pos ctrl
	3005:	BEFORE basic positioner
	3006:	BEFORE standard technology controller
	3007:	BEFORE act p v
	4000:	Receive AFTER IF1 PROFIdrive PZD
	4001:	Send BEFORE IF1 PROFIdrive PZD
	4002:	Receive AFTER IF2 PZD
	4003:	Send BEFORE IF2 PZD
	4004:	Receive AFTER IF1 PROFIdrive flexible PZD
	4005:	Receive AFTER IF2 flexible PZD
Index:	[0]:	Run-time group 1
	[1]:	Run-time group 2
	[2]:	Run-time group 3
	[3]:	Run-time group 4
	[4]:	Run-time group 5
	[5]:	Run-time group 6
	[6]:	Run-time group 7
	[7]:	Run-time group 8

[8]:	Run-time group 9
[9]:	Run-time group 10

Caution: The properties of the run-time groups must not be changed during operation as this could result in discontinuous signal transitions.

Note: Re value = 1 ... 256 (free run-time group):

This selection value can only be selected online if the following applies for sampling time T_sample of this run-time group:

1 ms <= T_sample < r21003.

At download, a value that violates this condition is not rejected, but a permissible equivalent value is set automatically and fault F51004 is output.

Re value > 2000 (fixed run-time group):

The fixed run-time groups p21000[x] >= 2000 log on with the sampling time of the associated basic system function, subject to a minimum sampling time of 1 ms. If, as a result of this limit, the actual sampling time deviates from the sampling time of the basic system function, then fault F51005 (during F51006 download) is output. In this case, another run-time group with a sampling time >= 1 ms should be selected. When selecting the fixed run-time groups, a check is not made as to whether the associated system block exists.

Example:

"BEFORE speed setpoint channel" means before function charts 3010, 3020, 3030, 3040, etc. are calculated, if the setpoint channel is activated. If, e.g. for SERVO, a setpoint channel has not been configured (p0108.8 = 0), the calculation is made before function chart 3095.

Re value = 4002, 4003, 4005 (IF2 run-time group):

On devices where IF2 does not exist (D4xx, CU310), when selecting the run-time groups that involve IF2, the corresponding run-time group for IF1 is automatically logged on.

p21000[0...9]	**Run-time group properties / RTG property**		
TB30, TM31	Changeable: -	Calculated: -	Access level: 1
	Data type: Integer16	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	0	4004	[0] 0

Description: Allocates properties to run-time groups 1 to 10.

This property comprises the sampling time and, for p21000[x] >= 2000, the instant of the call within the sampling time.

The index x + 1 of p21000 corresponds to the number of the run-time group:

- p21000[0] is used to set the property of the run-time group 1

...

- p21000[9] is used to set the property of the run-time group 10

Values:	0:	Do not calculate run-time group
	1:	T = 1 * r21002
	2:	T = 2 * r21002
	3:	T = 3 * r21002
	4:	T = 4 * r21002
	5:	T = 5 * r21002
	6:	T = 6 * r21002
	7:	T = 7 * r21002
	8:	T = 8 * r21002
	9:	T = 9 * r21002
	10:	T = 10 * r21002
	11:	T = 11 * r21002
	12:	T = 12 * r21002
	13:	T = 13 * r21002
	14:	T = 14 * r21002
	15:	T = 15 * r21002
	16:	T = 16 * r21002
	17:	T = 17 * r21002
	18:	T = 18 * r21002
	19:	T = 19 * r21002
	20:	T = 20 * r21002
	21:	T = 21 * r21002
	22:	T = 22 * r21002
	23:	T = 23 * r21002
	24:	T = 24 * r21002
	25:	T = 25 * r21002
	26:	T = 26 * r21002
	27:	T = 27 * r21002
	28:	T = 28 * r21002
	29:	T = 29 * r21002
	30:	T = 30 * r21002
	31:	T = 31 * r21002
	32:	T = 32 * r21002
	33:	T = 33 * r21002
	34:	T = 34 * r21002
	35:	T = 35 * r21002
	36:	T = 36 * r21002

37:	T = 37 * r21002
38:	T = 38 * r21002
39:	T = 39 * r21002
40:	T = 40 * r21002
41:	T = 41 * r21002
42:	T = 42 * r21002
43:	T = 43 * r21002
44:	T = 44 * r21002
45:	T = 45 * r21002
46:	T = 46 * r21002
47:	T = 47 * r21002
48:	T = 48 * r21002
49:	T = 49 * r21002
50:	T = 50 * r21002
51:	T = 51 * r21002
52:	T = 52 * r21002
53:	T = 53 * r21002
54:	T = 54 * r21002
55:	T = 55 * r21002
56:	T = 56 * r21002
57:	T = 57 * r21002
58:	T = 58 * r21002
59:	T = 59 * r21002
60:	T = 60 * r21002
61:	T = 61 * r21002
62:	T = 62 * r21002
63:	T = 63 * r21002
64:	T = 64 * r21002
65:	T = 65 * r21002
66:	T = 66 * r21002
67:	T = 67 * r21002
68:	T = 68 * r21002
69:	T = 69 * r21002
70:	T = 70 * r21002
71:	T = 71 * r21002
72:	T = 72 * r21002
73:	T = 73 * r21002
74:	T = 74 * r21002
75:	T = 75 * r21002
76:	T = 76 * r21002
77:	T = 77 * r21002
78:	T = 78 * r21002
79:	T = 79 * r21002
80:	T = 80 * r21002
81:	T = 81 * r21002
82:	T = 82 * r21002
83:	T = 83 * r21002
84:	T = 84 * r21002
85:	T = 85 * r21002
86:	T = 86 * r21002
87:	T = 87 * r21002
88:	T = 88 * r21002

89:	T = 89 * r21002
90:	T = 90 * r21002
91:	T = 91 * r21002
92:	T = 92 * r21002
93:	T = 93 * r21002
94:	T = 94 * r21002
95:	T = 95 * r21002
96:	T = 96 * r21002
97:	T = 97 * r21002
98:	T = 98 * r21002
99:	T = 99 * r21002
100:	T = 100 * r21002
101:	T = 101 * r21002
102:	T = 102 * r21002
103:	T = 103 * r21002
104:	T = 104 * r21002
105:	T = 105 * r21002
106:	T = 106 * r21002
107:	T = 107 * r21002
108:	T = 108 * r21002
109:	T = 109 * r21002
110:	T = 110 * r21002
111:	T = 111 * r21002
112:	T = 112 * r21002
113:	T = 113 * r21002
114:	T = 114 * r21002
115:	T = 115 * r21002
116:	T = 116 * r21002
117:	T = 117 * r21002
118:	T = 118 * r21002
119:	T = 119 * r21002
120:	T = 120 * r21002
121:	T = 121 * r21002
122:	T = 122 * r21002
123:	T = 123 * r21002
124:	T = 124 * r21002
125:	T = 125 * r21002
126:	T = 126 * r21002
127:	T = 127 * r21002
128:	T = 128 * r21002
129:	T = 129 * r21002
130:	T = 130 * r21002
131:	T = 131 * r21002
132:	T = 132 * r21002
133:	T = 133 * r21002
134:	T = 134 * r21002
135:	T = 135 * r21002
136:	T = 136 * r21002
137:	T = 137 * r21002
138:	T = 138 * r21002
139:	T = 139 * r21002
140:	T = 140 * r21002

141:	T = 141 * r21002
142:	T = 142 * r21002
143:	T = 143 * r21002
144:	T = 144 * r21002
145:	T = 145 * r21002
146:	T = 146 * r21002
147:	T = 147 * r21002
148:	T = 148 * r21002
149:	T = 149 * r21002
150:	T = 150 * r21002
151:	T = 151 * r21002
152:	T = 152 * r21002
153:	T = 153 * r21002
154:	T = 154 * r21002
155:	T = 155 * r21002
156:	T = 156 * r21002
157:	T = 157 * r21002
158:	T = 158 * r21002
159:	T = 159 * r21002
160:	T = 160 * r21002
161:	T = 161 * r21002
162:	T = 162 * r21002
163:	T = 163 * r21002
164:	T = 164 * r21002
165:	T = 165 * r21002
166:	T = 166 * r21002
167:	T = 167 * r21002
168:	T = 168 * r21002
169:	T = 169 * r21002
170:	T = 170 * r21002
171:	T = 171 * r21002
172:	T = 172 * r21002
173:	T = 173 * r21002
174:	T = 174 * r21002
175:	T = 175 * r21002
176:	T = 176 * r21002
177:	T = 177 * r21002
178:	T = 178 * r21002
179:	T = 179 * r21002
180:	T = 180 * r21002
181:	T = 181 * r21002
182:	T = 182 * r21002
183:	T = 183 * r21002
184:	T = 184 * r21002
185:	T = 185 * r21002
186:	T = 186 * r21002
187:	T = 187 * r21002
188:	T = 188 * r21002
189:	T = 189 * r21002
190:	T = 190 * r21002
191:	T = 191 * r21002
192:	T = 192 * r21002

193:	T = 193 * r21002
194:	T = 194 * r21002
195:	T = 195 * r21002
196:	T = 196 * r21002
197:	T = 197 * r21002
198:	T = 198 * r21002
199:	T = 199 * r21002
200:	T = 200 * r21002
201:	T = 201 * r21002
202:	T = 202 * r21002
203:	T = 203 * r21002
204:	T = 204 * r21002
205:	T = 205 * r21002
206:	T = 206 * r21002
207:	T = 207 * r21002
208:	T = 208 * r21002
209:	T = 209 * r21002
210:	T = 210 * r21002
211:	T = 211 * r21002
212:	T = 212 * r21002
213:	T = 213 * r21002
214:	T = 214 * r21002
215:	T = 215 * r21002
216:	T = 216 * r21002
217:	T = 217 * r21002
218:	T = 218 * r21002
219:	T = 219 * r21002
220:	T = 220 * r21002
221:	T = 221 * r21002
222:	T = 222 * r21002
223:	T = 223 * r21002
224:	T = 224 * r21002
225:	T = 225 * r21002
226:	T = 226 * r21002
227:	T = 227 * r21002
228:	T = 228 * r21002
229:	T = 229 * r21002
230:	T = 230 * r21002
231:	T = 231 * r21002
232:	T = 232 * r21002
233:	T = 233 * r21002
234:	T = 234 * r21002
235:	T = 235 * r21002
236:	T = 236 * r21002
237:	T = 237 * r21002
238:	T = 238 * r21002
239:	T = 239 * r21002
240:	T = 240 * r21002
241:	T = 241 * r21002
242:	T = 242 * r21002
243:	T = 243 * r21002
244:	T = 244 * r21002

	245:	T = 245 * r21002
	246:	T = 246 * r21002
	247:	T = 247 * r21002
	248:	T = 248 * r21002
	249:	T = 249 * r21002
	250:	T = 250 * r21002
	251:	T = 251 * r21002
	252:	T = 252 * r21002
	253:	T = 253 * r21002
	254:	T = 254 * r21002
	255:	T = 255 * r21002
	256:	T = 256 * r21002
	1001:	T = 1 * r21003
	1002:	T = 2 * r21003
	1003:	T = 3 * r21003
	1004:	T = 4 * r21003
	1005:	T = 5 * r21003
	1006:	T = 6 * r21003
	1008:	T = 8 * r21003
	1010:	T = 10 * r21003
	1012:	T = 12 * r21003
	1016:	T = 16 * r21003
	1020:	T = 20 * r21003
	1024:	T = 24 * r21003
	1032:	T = 32 * r21003
	1040:	T = 40 * r21003
	1048:	T = 48 * r21003
	1064:	T = 64 * r21003
	1080:	T = 80 * r21003
	1096:	T = 96 * r21003
	2000:	Read-in AFTER digital inputs
	2001:	Output BEFORE digital outputs
	2002:	Read-in AFTER analog inputs
	2003:	Output BEFORE analog outputs
	4000:	Receive AFTER IF1 PROFIdrive PZD
	4001:	Send BEFORE IF1 PROFIdrive PZD
	4004:	Receive AFTER IF1 PROFIdrive flexible PZD
Index:	[0]:	Run-time group 1
	[1]:	Run-time group 2
	[2]:	Run-time group 3
	[3]:	Run-time group 4
	[4]:	Run-time group 5
	[5]:	Run-time group 6
	[6]:	Run-time group 7
	[7]:	Run-time group 8
	[8]:	Run-time group 9
	[9]:	Run-time group 10

Caution: The properties of the run-time groups must not be changed during operation as this could result in discontinuous signal transitions.

Note: Re value = 1 ... 256 (free run-time group):

This selection value can only be selected online if the following applies for sampling time T_sample of this run-time group:
1 ms <= T_sample < r21003.
At download, a value that violates this condition is not rejected, but a permissible equivalent value is set automatically and fault F51004 is output.
Re value > 2000 (fixed run-time group):
The fixed run-time groups p21000[x] >= 2000 log on with the sampling time of the associated basic system function, subject to a minimum sampling time of 1 ms. If, as a result of this limit, the actual sampling time deviates from the sampling time of the basic system function, then fault F51005 (during F51006 download) is output. In this case, another run-time group with a sampling time >= 1 ms should be selected. When selecting the fixed run-time groups, a check is not made as to whether the associated system block exists.
Example:
"BEFORE speed setpoint channel" means before function charts 3010, 3020, 3030, 3040, etc. are calculated, if the setpoint channel is activated. If, e.g. for SERVO, a setpoint channel has not been configured (p0108.8 = 0), the calculation is made before function chart 3095.

p21000[0...9]	**Run-time group properties / RTG property**		
TM41	Changeable: -	Calculated: -	Access level: 1
	Data type: Integer16	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	0	4004	[0] 0

Description: Allocates properties to run-time groups 1 to 10.
This property comprises the sampling time and, for p21000[x] >= 2000, the instant of the call within the sampling time.
The index x + 1 of p21000 corresponds to the number of the run-time group:
- p21000[0] is used to set the property of the run-time group 1
...
- p21000[9] is used to set the property of the run-time group 10

Values:	0:	Do not calculate run-time group
	1:	T = 1 * r21002
	2:	T = 2 * r21002
	3:	T = 3 * r21002
	4:	T = 4 * r21002
	5:	T = 5 * r21002
	6:	T = 6 * r21002
	7:	T = 7 * r21002
	8:	T = 8 * r21002
	9:	T = 9 * r21002
	10:	T = 10 * r21002
	11:	T = 11 * r21002
	12:	T = 12 * r21002
	13:	T = 13 * r21002
	14:	T = 14 * r21002
	15:	T = 15 * r21002
	16:	T = 16 * r21002
	17:	T = 17 * r21002
	18:	T = 18 * r21002
	19:	T = 19 * r21002
	20:	T = 20 * r21002
	21:	T = 21 * r21002
	22:	T = 22 * r21002
	23:	T = 23 * r21002
	24:	T = 24 * r21002
	25:	T = 25 * r21002
	26:	T = 26 * r21002
	27:	T = 27 * r21002
	28:	T = 28 * r21002
	29:	T = 29 * r21002
	30:	T = 30 * r21002
	31:	T = 31 * r21002
	32:	T = 32 * r21002
	33:	T = 33 * r21002
	34:	T = 34 * r21002
	35:	T = 35 * r21002
	36:	T = 36 * r21002

37:	T = 37 * r21002
38:	T = 38 * r21002
39:	T = 39 * r21002
40:	T = 40 * r21002
41:	T = 41 * r21002
42:	T = 42 * r21002
43:	T = 43 * r21002
44:	T = 44 * r21002
45:	T = 45 * r21002
46:	T = 46 * r21002
47:	T = 47 * r21002
48:	T = 48 * r21002
49:	T = 49 * r21002
50:	T = 50 * r21002
51:	T = 51 * r21002
52:	T = 52 * r21002
53:	T = 53 * r21002
54:	T = 54 * r21002
55:	T = 55 * r21002
56:	T = 56 * r21002
57:	T = 57 * r21002
58:	T = 58 * r21002
59:	T = 59 * r21002
60:	T = 60 * r21002
61:	T = 61 * r21002
62:	T = 62 * r21002
63:	T = 63 * r21002
64:	T = 64 * r21002
65:	T = 65 * r21002
66:	T = 66 * r21002
67:	T = 67 * r21002
68:	T = 68 * r21002
69:	T = 69 * r21002
70:	T = 70 * r21002
71:	T = 71 * r21002
72:	T = 72 * r21002
73:	T = 73 * r21002
74:	T = 74 * r21002
75:	T = 75 * r21002
76:	T = 76 * r21002
77:	T = 77 * r21002
78:	T = 78 * r21002
79:	T = 79 * r21002
80:	T = 80 * r21002
81:	T = 81 * r21002
82:	T = 82 * r21002
83:	T = 83 * r21002
84:	T = 84 * r21002
85:	T = 85 * r21002
86:	T = 86 * r21002
87:	T = 87 * r21002
88:	T = 88 * r21002

89:	T = 89 * r21002
90:	T = 90 * r21002
91:	T = 91 * r21002
92:	T = 92 * r21002
93:	T = 93 * r21002
94:	T = 94 * r21002
95:	T = 95 * r21002
96:	T = 96 * r21002
97:	T = 97 * r21002
98:	T = 98 * r21002
99:	T = 99 * r21002
100:	T = 100 * r21002
101:	T = 101 * r21002
102:	T = 102 * r21002
103:	T = 103 * r21002
104:	T = 104 * r21002
105:	T = 105 * r21002
106:	T = 106 * r21002
107:	T = 107 * r21002
108:	T = 108 * r21002
109:	T = 109 * r21002
110:	T = 110 * r21002
111:	T = 111 * r21002
112:	T = 112 * r21002
113:	T = 113 * r21002
114:	T = 114 * r21002
115:	T = 115 * r21002
116:	T = 116 * r21002
117:	T = 117 * r21002
118:	T = 118 * r21002
119:	T = 119 * r21002
120:	T = 120 * r21002
121:	T = 121 * r21002
122:	T = 122 * r21002
123:	T = 123 * r21002
124:	T = 124 * r21002
125:	T = 125 * r21002
126:	T = 126 * r21002
127:	T = 127 * r21002
128:	T = 128 * r21002
129:	T = 129 * r21002
130:	T = 130 * r21002
131:	T = 131 * r21002
132:	T = 132 * r21002
133:	T = 133 * r21002
134:	T = 134 * r21002
135:	T = 135 * r21002
136:	T = 136 * r21002
137:	T = 137 * r21002
138:	T = 138 * r21002
139:	T = 139 * r21002
140:	T = 140 * r21002

141:	T = 141 * r21002
142:	T = 142 * r21002
143:	T = 143 * r21002
144:	T = 144 * r21002
145:	T = 145 * r21002
146:	T = 146 * r21002
147:	T = 147 * r21002
148:	T = 148 * r21002
149:	T = 149 * r21002
150:	T = 150 * r21002
151:	T = 151 * r21002
152:	T = 152 * r21002
153:	T = 153 * r21002
154:	T = 154 * r21002
155:	T = 155 * r21002
156:	T = 156 * r21002
157:	T = 157 * r21002
158:	T = 158 * r21002
159:	T = 159 * r21002
160:	T = 160 * r21002
161:	T = 161 * r21002
162:	T = 162 * r21002
163:	T = 163 * r21002
164:	T = 164 * r21002
165:	T = 165 * r21002
166:	T = 166 * r21002
167:	T = 167 * r21002
168:	T = 168 * r21002
169:	T = 169 * r21002
170:	T = 170 * r21002
171:	T = 171 * r21002
172:	T = 172 * r21002
173:	T = 173 * r21002
174:	T = 174 * r21002
175:	T = 175 * r21002
176:	T = 176 * r21002
177:	T = 177 * r21002
178:	T = 178 * r21002
179:	T = 179 * r21002
180:	T = 180 * r21002
181:	T = 181 * r21002
182:	T = 182 * r21002
183:	T = 183 * r21002
184:	T = 184 * r21002
185:	T = 185 * r21002
186:	T = 186 * r21002
187:	T = 187 * r21002
188:	T = 188 * r21002
189:	T = 189 * r21002
190:	T = 190 * r21002
191:	T = 191 * r21002
192:	T = 192 * r21002

193:	T = 193 * r21002
194:	T = 194 * r21002
195:	T = 195 * r21002
196:	T = 196 * r21002
197:	T = 197 * r21002
198:	T = 198 * r21002
199:	T = 199 * r21002
200:	T = 200 * r21002
201:	T = 201 * r21002
202:	T = 202 * r21002
203:	T = 203 * r21002
204:	T = 204 * r21002
205:	T = 205 * r21002
206:	T = 206 * r21002
207:	T = 207 * r21002
208:	T = 208 * r21002
209:	T = 209 * r21002
210:	T = 210 * r21002
211:	T = 211 * r21002
212:	T = 212 * r21002
213:	T = 213 * r21002
214:	T = 214 * r21002
215:	T = 215 * r21002
216:	T = 216 * r21002
217:	T = 217 * r21002
218:	T = 218 * r21002
219:	T = 219 * r21002
220:	T = 220 * r21002
221:	T = 221 * r21002
222:	T = 222 * r21002
223:	T = 223 * r21002
224:	T = 224 * r21002
225:	T = 225 * r21002
226:	T = 226 * r21002
227:	T = 227 * r21002
228:	T = 228 * r21002
229:	T = 229 * r21002
230:	T = 230 * r21002
231:	T = 231 * r21002
232:	T = 232 * r21002
233:	T = 233 * r21002
234:	T = 234 * r21002
235:	T = 235 * r21002
236:	T = 236 * r21002
237:	T = 237 * r21002
238:	T = 238 * r21002
239:	T = 239 * r21002
240:	T = 240 * r21002
241:	T = 241 * r21002
242:	T = 242 * r21002
243:	T = 243 * r21002
244:	T = 244 * r21002

	245:	T = 245 * r21002
	246:	T = 246 * r21002
	247:	T = 247 * r21002
	248:	T = 248 * r21002
	249:	T = 249 * r21002
	250:	T = 250 * r21002
	251:	T = 251 * r21002
	252:	T = 252 * r21002
	253:	T = 253 * r21002
	254:	T = 254 * r21002
	255:	T = 255 * r21002
	256:	T = 256 * r21002
	1001:	T = 1 * r21003
	1002:	T = 2 * r21003
	1003:	T = 3 * r21003
	1004:	T = 4 * r21003
	1005:	T = 5 * r21003
	1006:	T = 6 * r21003
	1008:	T = 8 * r21003
	1010:	T = 10 * r21003
	1012:	T = 12 * r21003
	1016:	T = 16 * r21003
	1020:	T = 20 * r21003
	1024:	T = 24 * r21003
	1032:	T = 32 * r21003
	1040:	T = 40 * r21003
	1048:	T = 48 * r21003
	1064:	T = 64 * r21003
	1080:	T = 80 * r21003
	1096:	T = 96 * r21003
	2000:	Read-in AFTER digital inputs
	2001:	Output BEFORE digital outputs
	2002:	Read-in AFTER analog inputs
	4000:	Receive AFTER IF1 PROFIdrive PZD
	4001:	Send BEFORE IF1 PROFIdrive PZD
	4004:	Receive AFTER IF1 PROFIdrive flexible PZD
Index:	[0]:	Run-time group 1
	[1]:	Run-time group 2
	[2]:	Run-time group 3
	[3]:	Run-time group 4
	[4]:	Run-time group 5
	[5]:	Run-time group 6
	[6]:	Run-time group 7
	[7]:	Run-time group 8
	[8]:	Run-time group 9
	[9]:	Run-time group 10

Caution: The properties of the run-time groups must not be changed during operation as this could result in discontinuous signal transitions.

Note: Re value = 1 ... 256 (free run-time group):
This selection value can only be selected online if the following applies for sampling time T_sample of this run-time group:

1 ms <= T_sample < r21003.
At download, a value that violates this condition is not rejected, but a permissible equivalent value is set automatically and fault F51004 is output.
Re value > 2000 (fixed run-time group):
The fixed run-time groups p21000[x] >= 2000 log on with the sampling time of the associated basic system function, subject to a minimum sampling time of 1 ms. If, as a result of this limit, the actual sampling time deviates from the sampling time of the basic system function, then fault F51005 (during F51006 download) is output. In this case, another run-time group with a sampling time >= 1 ms should be selected. When selecting the fixed run-time groups, a check is not made as to whether the associated system block exists.
Example:
"BEFORE speed setpoint channel" means before function charts 3010, 3020, 3030, 3040, etc. are calculated, if the setpoint channel is activated. If, e.g. for SERVO, a setpoint channel has not been configured (p0108.8 = 0), the calculation is made before function chart 3095.

p21030	**Run-time group, computing time measurement / RTG comp_ti_meas**		
All objects	Changeable: -	Calculated: -	Access level: 4
	Data type: Unsigned16	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	0	10	[0] 0

Description: Only for internal Siemens service purposes.

See also: p21032, r21035, r21036, r21037

p21031	**Computing time measurement, blocks / Comp_ti_meas block**		
All objects	Changeable: -	Calculated: -	Access level: 4
	Data type: Unsigned32	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	0	10	[0] 0

Description: Sets the number of the run-time group to be measured 1 ... 10.
This starts measurement of the entire run-time group and the individual blocks.
At the end of the measurement the parameter is reset automatically to 0.

Danger: During this measurement, the run-time group n is logged off and on again twice for the purpose of time slice management. Since this procedure runs in a background task, the run-time group is not calculated for an indefinite period on two occasions. This can lead to freezing or jumps in signal values. Logical signals can assume unexpected values. This measurement must therefore never be carried out when an application is running!

Note: The run-time group to be measured has to be logged on.
Only one measurement can take place at a time.
The results are displayed in parameters r21035, r21036, r21037.

p21032 **Computing time measurement, duration / Comp_ti_meas dur**

All objects	Changeable: -	Calculated: -	Access level: 4
	Data type: Unsigned16	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	60 [s]	10000 [s]	[0] 60 [s]

Description: Only for internal Siemens service purposes.

See also: p21030, r21035, r21036, r21037

p21033 **Computing time measurement, number of individual measurements / Comp_ti_meas qty**

All objects	Changeable: -	Calculated: -	Access level: 4
	Data type: Unsigned32	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	1	4294967295	[0] 10000

Description: Setting for the number of calls during the measurement of the individual blocks.

See also: p21031

r21001[0...9] **Run-time group sampling time / RTG sampling time**

All objects	Changeable: -	Calculated: -	Access level: 1
	Data type: FloatingPoint32	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	- [ms]	- [ms]	[] - [ms]

Description: Displays the current sampling time of the run-time groups.

Index:	[0]:	Run-time group 1
	[1]:	Run-time group 2
	[2]:	Run-time group 3
	[3]:	Run-time group 4
	[4]:	Run-time group 5
	[5]:	Run-time group 6
	[6]:	Run-time group 7
	[7]:	Run-time group 8
	[8]:	Run-time group 9
	[9]:	Run-time group 10

r21002 **Basis sampling time, hardware / Basis samp time HW**

All objects	Changeable: -	Calculated: -	Access level: 1
	Data type: FloatingPoint32	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	- [ms]	- [ms]	[] - [ms]

Description: Displays the basis sampling time effective at this drive object for values 1 to 256 of p21000.
Sampling time T = p21000 * r21002

r21003 **Basis sampling time, software / Basis samp time SW**

All objects	Changeable: -	Calculated: -	Access level: 1
	Data type: FloatingPoint32	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	- [ms]	- [ms]	[] - [ms]

Description: Displays the basis sampling time effective at this drive object for p21000 = 1002 to 1096 as factor.
Sampling time T = (p21000 - 1000) * r21003

r21005[0...9] **Computing time load of the run-time group / RTG load**

All objects	Changeable: -	Calculated: -	Access level: 3
	Data type: FloatingPoint32	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	- [%]	- [%]	[] - [%]

Description: Share of the computing time load with which the DCC run-time group contributes to the utilization of the sampling time during which it is called.

Index:		
	[0]:	Run-time group 1
	[1]:	Run-time group 2
	[2]:	Run-time group 3
	[3]:	Run-time group 4
	[4]:	Run-time group 5
	[5]:	Run-time group 6
	[6]:	Run-time group 7
	[7]:	Run-time group 8
	[8]:	Run-time group 9
	[9]:	Run-time group 10

Note: The computing time load can only be displayed for the run-time groups which are logged on (p21000[x] > 0). The value for the computing time load is calculated in the drive unit based on the project loaded plus DCC chart. Therefore, the values r21005[x] are not available in the offline mode of the SCOUT/STARTER.
In r21005 the computing time load is displayed, with which the DCC runtime group utilizes the sampling time in which it is called. The runtime groups "Receive AFTER IF1 PROFIdrive PZD" (p21000 = 4000), "Send BEFORE IF1 PROFIdrive PZD" (p21000 = 4001) , "Receive BEFORE IF2 PZD" (p21000 = 4002) and "Send BEFORE IF2 PZD" (p21000 = 4003) are called in the isochronous mode and in the non-isochronous mode, in different sampling times.
In the non-isochronous mode, these are IF1 / IF2 PZD sampling time (p2048 for p21000 = 4000 or 4001, p8848 for p21000 = 4002 or 4003). In the isochronous mode, this is the current controller sampling time (p115[0]) which is periodically called with the isochronous bus cycle time. The computing time load displayed in r21005 is always calculated for the (more unfavorable) case of isochronous operation. This is why this value does not (always) act to the full amount on the computing time load of the complete system.

r21008[0...24]	**Hardware sampling times available / HW t_samp**		
All objects	Changeable: -	Calculated: -	Access level: 3
	Data type: FloatingPoint32	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	- [ms]	- [ms]	[] - [ms]

Description: Displays the assignment of the available hardware sampling times of the drive unit.
The designated sampling times are those created as a multiple of the hardware basis sampling time (r21002) and which are always < r21003.

Index:

[0]:	Hardware 1
[1]:	Hardware 2
[2]:	Hardware 3
[3]:	Hardware 4
[4]:	Hardware 5
[5]:	Hardware 6
[6]:	Hardware 7
[7]:	Hardware 8
[8]:	Hardware 9
[9]:	Hardware 10
[10]:	Hardware 11
[11]:	Hardware 12
[12]:	Hardware 13
[13]:	Hardware 14
[14]:	Hardware 15
[15]:	Hardware 16
[16]:	Hardware 17
[17]:	Hardware 18
[18]:	Hardware 19
[19]:	Hardware 20
[20]:	Hardware 21
[21]:	Hardware 22
[22]:	Hardware 23
[23]:	Hardware 24
[24]:	Hardware 25

Notice: For internal purposes, the drive unit always requires at least two (or several, depending on how p0115[0] is parameterized for the drive objects) free hardware sampling times. Therefore the current number of free hardware sampling times can be read out in r7903.
If r7903=0, no additional sampling time different from r21008[0...24] may be provided from the Control Unit. When selecting in this state, if a run-time group with a sampling time < r21003 (p21000 <= 255) is selected in p21000, only run-time groups whose sampling time is already provided in r21008[0...24] may be selected.

Note: A sampling time that is provided can be simultaneously used by system functions, several FBLOCK run-time groups and several DCC run-time groups.
The sampling time of run-time groups that have been assigned to the PROFIBUS run-time groups (p21000 = 4000 ... 4004) is not displayed in r21008. For this sampling time, one of the internally and permanently assigned hardware sampling times is used.
If the value of r21008[x] != 0 (not equal to 0), then the sampling time is specified in ms.
If the value of r21008[x] = 0, this sampling time can still be freely assigned. It should be noted that the basic system, depending on the selected basis sampling times p0115[0], requires at least two (some-

times more) freely assignable hardware sampling times for internal functions. The number of hardware sampling times that can still be freely assigned can be read out in r7903.
If the value r21008[x] = 99999.00000, this hardware sampling time is not supported.

r21035[0...9]	Computing time, minimum value / Computing time min		
All objects	Changeable: -	Calculated: -	Access level: 4
	Data type: FloatingPoint32	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	- [µs]	- [µs]	[] - [µs]

Description: Only for internal Siemens service purposes.

Index:		
	[0]:	Run-time group 1
	[1]:	Run-time group 2
	[2]:	Run-time group 3
	[3]:	Run-time group 4
	[4]:	Run-time group 5
	[5]:	Run-time group 6
	[6]:	Run-time group 7
	[7]:	Run-time group 8
	[8]:	Run-time group 9
	[9]:	Run-time group 10

See also: p21030, p21032, r21036, r21037

r21036[0...9]	Computing time, mean value / Computing tim av		
All objects	Changeable: -	Calculated: -	Access level: 4
	Data type: FloatingPoint32	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	- [µs]	- [µs]	[] - [µs]

Description: Only for internal Siemens service purposes.

Index:		
	[0]:	Run-time group 1
	[1]:	Run-time group 2
	[2]:	Run-time group 3
	[3]:	Run-time group 4
	[4]:	Run-time group 5
	[5]:	Run-time group 6
	[6]:	Run-time group 7
	[7]:	Run-time group 8
	[8]:	Run-time group 9
	[9]:	Run-time group 10

r21037[0...9]	**Computing time, maximum value / Computing time max**		
All objects	Changeable: -	Calculated: -	Access level: 4
	Data type: FloatingPoint32	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	- [µs]	- [µs]	[] - [µs]

Description: Only for internal Siemens service purposes.

Index:		
	[0]:	Run-time group 1
	[1]:	Run-time group 2
	[2]:	Run-time group 3
	[3]:	Run-time group 4
	[4]:	Run-time group 5
	[5]:	Run-time group 6
	[6]:	Run-time group 7
	[7]:	Run-time group 8
	[8]:	Run-time group 9
	[9]:	Run-time group 10

See also: p21030, p21032, r21035, r21036

r21041[0...49]	**Block ID of the measured block / Block ID**		
All objects	Changeable: -	Calculated: -	Access level: 4
	Data type: Unsigned16	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	-	-	[] -

Description: Block ID of the measured block (block run-time measurement via parameter p21031. The blocks are measured in the same sequence as they have been programmed in the execution sequence.
The parameter is designed for the measurement of 50 block instances

Index:	[0]:	Block 1
	[1]:	Block 2
	[2]:	Block 3
	[3]:	Block 4
	[4]:	Block 5
	[5]:	Block 6
	[6]:	Block 7
	[7]:	Block 8
	[8]:	Block 9
	[9]:	Block 10
	[10]:	Block 11
	[11]:	Block 12
	[12]:	Block 13
	[13]:	Block 14
	[14]:	Block 15
	[15]:	Block 16
	[16]:	Block 17
	[17]:	Block 18
	[18]:	Block 19
	[19]:	Block 20
	[20]:	Block 21
	[21]:	Block 22
	[22]:	Block 23
	[23]:	Block 24
	[24]:	Block 25
	[25]:	Block 26
	[26]:	Block 27
	[27]:	Block 28
	[28]:	Block 29
	[29]:	Block 30
	[30]:	Block 31
	[31]:	Block 32
	[32]:	Block 33
	[33]:	Block 34
	[34]:	Block 35
	[35]:	Block 36
	[36]:	Block 37
	[37]:	Block 38
	[38]:	Block 39
	[39]:	Block 40
	[40]:	Block 41

[41]:	Block 42
[42]:	Block 43
[43]:	Block 44
[44]:	Block 45
[45]:	Block 46
[46]:	Block 47
[47]:	Block 48
[48]:	Block 49
[49]:	Block 50

r21042[0...49]	**First run / subsequent run identifiers / First**		
All objects	Changeable: -	Calculated: -	Access level: 4
	Data type: Unsigned16	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	-	-	[] -

Description: In the block run-time measurements, the block run-times are measured. R21039 indicates whether the measurement is the first or a subsequent call.
If the block type occurs only once in the run-time group, only the measured value for the first run will be supplied.
The parameter is designed for the measurement of 50 block instances

Index:	[0]:	Block 1
	[1]:	Block 2
	[2]:	Block 3
	[3]:	Block 4
	[4]:	Block 5
	[5]:	Block 6
	[6]:	Block 7
	[7]:	Block 8
	[8]:	Block 9
	[9]:	Block 10
	[10]:	Block 11
	[11]:	Block 12
	[12]:	Block 13
	[13]:	Block 14
	[14]:	Block 15
	[15]:	Block 16
	[16]:	Block 17
	[17]:	Block 18
	[18]:	Block 19
	[19]:	Block 20
	[20]:	Block 21
	[21]:	Block 22
	[22]:	Block 23
	[23]:	Block 24
	[24]:	Block 25
	[25]:	Block 26
	[26]:	Block 27
	[27]:	Block 28
	[28]:	Block 29
	[29]:	Block 30
	[30]:	Block 31
	[31]:	Block 32
	[32]:	Block 33
	[33]:	Block 34
	[34]:	Block 35
	[35]:	Block 36
	[36]:	Block 37
	[37]:	Block 38
	[38]:	Block 39

[39]:	Block 40
[40]:	Block 41
[41]:	Block 42
[42]:	Block 43
[43]:	Block 44
[44]:	Block 45
[45]:	Block 46
[46]:	Block 47
[47]:	Block 48
[48]:	Block 49
[49]:	Block 50

r21043[0...49]	**Minimum measured block run-time in us / Computing time min**		
All objects	Changeable: -	Calculated: -	Access level: 4
	Data type: FloatingPoint32	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	- [µs]	- [µs]	[] - [µs]

Description: Minimum measured run-time of the measured block (block run-time measurement via parameter p21031. The blocks are measured in the same sequence as they have been programmed in the execution sequence.
The parameter is designed for the measurement of 50 block instances

Index:

[0]:	Block 1
[1]:	Block 2
[2]:	Block 3
[3]:	Block 4
[4]:	Block 5
[5]:	Block 6
[6]:	Block 7
[7]:	Block 8
[8]:	Block 9
[9]:	Block 10
[10]:	Block 11
[11]:	Block 12
[12]:	Block 13
[13]:	Block 14
[14]:	Block 15
[15]:	Block 16
[16]:	Block 17
[17]:	Block 18
[18]:	Block 19
[19]:	Block 20
[20]:	Block 21
[21]:	Block 22
[22]:	Block 23
[23]:	Block 24
[24]:	Block 25
[25]:	Block 26
[26]:	Block 27
[27]:	Block 28
[28]:	Block 29
[29]:	Block 30
[30]:	Block 31
[31]:	Block 32
[32]:	Block 33
[33]:	Block 34
[34]:	Block 35
[35]:	Block 36
[36]:	Block 37
[37]:	Block 38
[38]:	Block 39
[39]:	Block 40

[40]:	Block 41
[41]:	Block 42
[42]:	Block 43
[43]:	Block 44
[44]:	Block 45
[45]:	Block 46
[46]:	Block 47
[47]:	Block 48
[48]:	Block 49
[49]:	Block 50

r21044[0...49]	**Average measured block run-time in us / Computing tim av**		
All objects	Changeable: -	Calculated: -	Access level: 4
	Data type: FloatingPoint32	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	- [µs]	- [µs]	[] - [µs]

Description: Average measured run-time of the measured block (block run-time measurement via parameter p21031. The blocks are measured in the same sequence as they have been programmed in the execution sequence.
The parameter is designed for the measurement of 50 block instances

Index:		
	[0]:	Block 1
	[1]:	Block 2
	[2]:	Block 3
	[3]:	Block 4
	[4]:	Block 5
	[5]:	Block 6
	[6]:	Block 7
	[7]:	Block 8
	[8]:	Block 9
	[9]:	Block 10
	[10]:	Block 11
	[11]:	Block 12
	[12]:	Block 13
	[13]:	Block 14
	[14]:	Block 15
	[15]:	Block 16
	[16]:	Block 17
	[17]:	Block 18
	[18]:	Block 19
	[19]:	Block 20
	[20]:	Block 21
	[21]:	Block 22
	[22]:	Block 23
	[23]:	Block 24
	[24]:	Block 25
	[25]:	Block 26
	[26]:	Block 27
	[27]:	Block 28
	[28]:	Block 29
	[29]:	Block 30
	[30]:	Block 31
	[31]:	Block 32
	[32]:	Block 33
	[33]:	Block 34
	[34]:	Block 35
	[35]:	Block 36
	[36]:	Block 37
	[37]:	Block 38
	[38]:	Block 39
	[39]:	Block 40

[40]:	Block 41
[41]:	Block 42
[42]:	Block 43
[43]:	Block 44
[44]:	Block 45
[45]:	Block 46
[46]:	Block 47
[47]:	Block 48
[48]:	Block 49
[49]:	Block 50

r21045[0...49]	**Maximum measured block run-time in us / Computing time max**		
All objects	Changeable: -	Calculated: -	Access level: 4
	Data type: FloatingPoint32	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	- [µs]	- [µs]	[] - [µs]

Description: Average measured run-time of the measured block (block run-time measurement via parameter p21031. The blocks are measured in the same sequence as they have been programmed in the execution sequence.
The parameter is designed for the measurement of 50 block instances

Index:		
	[0]:	Block 1
	[1]:	Block 2
	[2]:	Block 3
	[3]:	Block 4
	[4]:	Block 5
	[5]:	Block 6
	[6]:	Block 7
	[7]:	Block 8
	[8]:	Block 9
	[9]:	Block 10
	[10]:	Block 11
	[11]:	Block 12
	[12]:	Block 13
	[13]:	Block 14
	[14]:	Block 15
	[15]:	Block 16
	[16]:	Block 17
	[17]:	Block 18
	[18]:	Block 19
	[19]:	Block 20
	[20]:	Block 21
	[21]:	Block 22
	[22]:	Block 23
	[23]:	Block 24
	[24]:	Block 25
	[25]:	Block 26
	[26]:	Block 27
	[27]:	Block 28
	[28]:	Block 29
	[29]:	Block 30
	[30]:	Block 31
	[31]:	Block 32
	[32]:	Block 33
	[33]:	Block 34
	[34]:	Block 35
	[35]:	Block 36
	[36]:	Block 37
	[37]:	Block 38
	[38]:	Block 39
	[39]:	Block 40

[40]:	Block 41
[41]:	Block 42
[42]:	Block 43
[43]:	Block 44
[44]:	Block 45
[45]:	Block 46
[46]:	Block 47
[47]:	Block 48
[48]:	Block 49
[49]:	Block 50

r21046[0...49]	Library IDs of the measured blocks / Lib ID measured		
All objects	Changeable: -	Calculated: -	Access level: 4
	Data type: Unsigned32	Dynamic index: -	Function plan: -
	P group: -	Units group: -	Units selection: -
	Not for motor type: -		Expert list: 1
	Min	Max	Factory setting
	-	-	[] -

Description: Library ID of the measured block (block run-time measurement via parameter p21031.
Measurements with blocks from different libraries can thereby be carried out in one run-time group.
The blocks are measured in the same sequence as they have been programmed in the execution sequence.
The parameter is designed for the measurement of 50 block instances
Indices 0..49

Index:		
	[0]:	Block 1
	[1]:	Block 2
	[2]:	Block 3
	[3]:	Block 4
	[4]:	Block 5
	[5]:	Block 6
	[6]:	Block 7
	[7]:	Block 8
	[8]:	Block 9
	[9]:	Block 10
	[10]:	Block 11
	[11]:	Block 12
	[12]:	Block 13
	[13]:	Block 14
	[14]:	Block 15
	[15]:	Block 16
	[16]:	Block 17
	[17]:	Block 18
	[18]:	Block 19
	[19]:	Block 20
	[20]:	Block 21
	[21]:	Block 22
	[22]:	Block 23
	[23]:	Block 24
	[24]:	Block 25
	[25]:	Block 26
	[26]:	Block 27
	[27]:	Block 28
	[28]:	Block 29
	[29]:	Block 30
	[30]:	Block 31
	[31]:	Block 32
	[32]:	Block 33
	[33]:	Block 34
	[34]:	Block 35
	[35]:	Block 36
	[36]:	Block 37
	[37]:	Block 38

[38]:	Block 39
[39]:	Block 40
[40]:	Block 41
[41]:	Block 42
[42]:	Block 43
[43]:	Block 44
[44]:	Block 45
[45]:	Block 46
[46]:	Block 47
[47]:	Block 48
[48]:	Block 49
[49]:	Block 50

Index

Siemens AG

Automation and Drives
Motion Control Systems
Postfach 3180
91050 ERLANGEN
GERMANY
www.siemens.com/motioncontrol