SIEMENS

SINAMICS DCC V18

SINAMICS DCC standard blocks

Function Manual

10/2022
A5E52288783B AA

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
Digital Industries
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E52288783B AA
Ⓟ 09/2022 Subject to change

Table of contents

1 Safety information

1.1 Fundamental safety instructions

1.1.1 General safety instructions

⚠ WARNING
Danger to life if the safety instructions and residual risks are not observed
If the safety instructions and residual risks in the associated hardware documentation are not observed, 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
Malfunctions of the machine as a result of incorrect or changed parameter settings
As a result of incorrect or changed parameterization, machines can malfunction, which in turn can lead to injuries or death. • Protect the parameterization against unauthorized access. • Handle possible malfunctions by taking suitable measures, e.g. emergency stop or emergency off.

1.1.2 Warranty and liability for application examples

Application examples are not binding and do not claim to be complete regarding configuration, equipment or any eventuality which may arise. Application examples do not represent specific customer solutions, but are only intended to provide support for typical tasks.

As the user you yourself are responsible for ensuring that the products described are operated correctly. Application examples do not relieve you of your responsibility for safe handling when using, installing, operating and maintaining the equipment.

1.1.3 Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that may be implemented, please visit
https://www.siemens.com/industrialsecurity (https://www.siemens.com/industrialsecurity).

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under
https://www.siemens.com/cert (https://www.siemens.com/cert).

Further information is provided on the Internet:

Industrial Security Configuration Manual (https://support.industry.siemens.com/cs/ww/en/view/108862708)

⚠ **WARNING**

Unsafe operating states resulting from software manipulation

Software manipulations, e.g. viruses, Trojans, or worms, can cause unsafe operating states in your system that may lead to death, serious injury, and property damage.

- Keep the software up to date.
- Incorporate the automation and drive components into a holistic, state-of-the-art industrial security concept for the installation or machine.
- Make sure that you include all installed products into the holistic industrial security concept.
- Protect files stored on exchangeable storage media from malicious software by with suitable protection measures, e.g. virus scanners.
- On completion of commissioning, check all security-related settings.

1.2 Industrial Security Manual

Industrial Security Manual

Additional information can be found in the configuration manual "Industrial Security" for SINAMICS, SINUMERIK and SIMOTION at this address.

Pay particular attention to the explanatory notes regarding the cell protection concept in the section "General security measures - network segmentation".

1.3 Unsafe operating states due to manipulation of the Safety Integrated parameters after the acceptance test

⚠ WARNING

Unsafe operating states due to manipulation of the Safety Integrated parameters after the acceptance test

Incorrect parameter changes to Safety Integrated Functions after an acceptance test can result in unwanted motion with subsequent severe injury or death.

- To prevent access to your plants and systems by unauthorized persons, implement access restrictions and take the precautions described in the security information.
- To avoid incorrect changes to the configuration and parameters of the Safety Integrated Functions, take the precautions described in the "Acceptance test" chapter of the SINAMICS S120 Safety Integrated Function Manual.
- Check the safety log book of SINAMICS Safety Integrated at regular intervals. Verify that no changes have been made to the parameters since the last acceptance test was performed.
- If any changes have been made and they are intentional, repeat the acceptance test for the Safety Integrated Functions affected. The purpose of the acceptance test is to ensure and document safe operation of the plant. Correct any unintentional changes back to the original values and repeat the acceptance test.

1.4 Protection of sensitive data in Startdrive project and drive configuration

Note

Extraction of sensitive data for unprotected transfer of projects

The parameters of the SINAMICS drives contain your know-how and sensitive configuration data as well as the configuration for protection against modifications for some drive functions like Safety Integrated. After an upload from the device this configuration is stored in the project. If a project is transferred unencrypted via unprotected channels (e.g. email) or stored in an unencrypted form (e.g. in cloud storage), unauthorized persons can extract this configuration from the project files.

- Activate the project protection in SINAMICS Startdrive to encrypt all drive parameters in the Startdrive project.
- Encrypt the exported files and project files with some other software.
- To prevent access to relevant data memory by unauthorized persons, implement access restrictions (e.g. password protection) and take the precautions described in the security information.

NOTICE

Protecting parameters in the SINAMICS drive memory

The parameters stored in the SINAMICS drive can be read out by unauthorized third parties without protection. Unauthorized persons can therefore cause damage.

- In addition to the project protection or the specific encryption of project files, also activate the know-how protection of the SINAMICS drive.
- If setting up know-how protection is not an option, as an alternative, prevent unauthorized persons from accessing your plants and systems. Implement access restrictions and take the precautions described in the security information.

1.5 Unsafe configuration after importing and downloading files from unknown or untrustworthy sources

⚠ WARNING

Unsafe configuration after importing and downloading files from unknown or untrustworthy sources

If you use project files or files (e.g. from EPLAN, Microsoft Excel) from unknown or untrustworthy sources or import such files into your Startdrive project, inconsistencies in the project or malfunctions of Startdrive may result. If the appropriate safety precautions are not observed, any untested changes in the system can cause unsafe operating states in your system that may lead to death, serious injury, and property damage.

If project files or imported files are transferred unsigned via unprotected channels (e.g. email) or stored without access protection (e.g. in cloud memories or local memories), unauthorized persons can change the system configuration, thus causing unsafe operating states in your system that may lead to death, serious injury, and property damage.

- Only use projects and files from sources that you know to be trustworthy.
- For the consistency check, use functions such as "Flash LED" in the "Go online" dialog or the parameter comparison in the parameter view.
- Check whether the machine behavior with the changed system configuration meets your expectations and perform an acceptance test of the Safety Integrated Functions to ensure and document the safe operation of the system.
- Take the precautions described in the security information.

1.6 Data security for libraries

Note

Data security for libraries

In order to ensure data security for global libraries and project libraries (e.g. imported DCB Extension libraries generated with SINAMICS DCB Studio), only use secure data storage and work only with libraries from secure sources, e.g. from signed e-mails or from SIOS. Do not load any data from unknown servers.

General information 2

2.1 Compliance with the General Data Protection Regulation

Siemens respects the principles of data privacy, in particular the data minimization rules (privacy by design).

For the SINAMICS Startdrive product – including the installed SINAMICS DCC option package – this means the following:
The product only sends personal data to SIEMENS AG if the user explicitly requests this. This occurs in the following cases:

- If the SINAMICS Startdrive program and the SINAMICS DCC option package end unexpectedly, then the user is given the opportunity to send diagnostics information to SIEMENS AG for analysis. If the user avails themselves of this option, then their email address will be collected, transmitted and saved so that they can be contacted in the event of queries.
- The TIA Administrator enables the user to check whether updates are available for SINAMICS Startdrive and the SINAMICS DCC option package and to install them. As the TIA Automation Update Server is used for verification and installation purposes, the IP address of the device used is transmitted for technical reasons.
- Feedback and diagnostics data is collected when using SINAMICS Startdrive and the SINAMICS DCC option package. This data, also including the IP address of the device used, is transmitted to a SIEMENS server. For more information on this topic, refer to the Online Help in the "Notes on the TIA Portal" chapter, keywords "Collecting feedback and diagnostics data".

Beyond the previously mentioned information, the product only saves personal data in the project. The user is therefore responsible for ensuring compliance with the statutory data protection provisions. This applies in particular to the transfer of projects.

The following data must be taken into account.

- Windows login
  In the standard configuration, the product saves the login details of the Windows user together with technical function data (e.g. time stamp) in the project. The specified data is saved in order to trace changes in large configurations.
  For SINAMICS Startdrive and the SINAMICS DCC option package, reference to specific persons can be established via the project and all elements contained within it (e.g. devices and DCC charts).
  The specified data can be viewed in the properties of the project and the elements in SINAMICS Startdrive and the SINAMICS DCC option package ("Author" property) and, with the exception of the most recent change to the project, subsequently modified.
- Usernames in the user administration
  The product only processes and saves additional personal data when the user explicitly enables one of the following functions.
  Usernames created by the user are saved by the user administration (security settings) to be able to verify them during subsequent authentication.
- Login for multi-user engineering
  For multi-user engineering, various technical function data (e.g. time stamp) are saved together with the login of the Windows user concerned in order to be able to trace project changes.

For the three previously mentioned points, the details relating to the specified functions must be observed in the relevant chapter in the information system of SINAMICS Startdrive and the SINAMICS DCC option package.

By generating the login or username, personal data can be pseudonymized for the functions. Deleting the project will cause all personal data saved within it to be deleted too. The particularities of multi-user engineering should be taken into consideration here (e.g. that the project not only needs to be deleted locally from the user's PC, but also from the server used).

2.2 Display of the information system for Chinese user interface language

Incorrect display of texts under Windows 7 SP1

If you install Startdrive or the SINAMICS DCC option package under Windows 7 SP1 and set the user interface language to Chinese, errors in the display of texts in the information system may occur. This incorrect display of texts is known i.a. as mojibake.

Remedy: Installation of Internet Explorer 11

Install Microsoft Internet Explorer 11 or later to ensure the texts are displayed correctly in the information system.

2.3 General information about SINAMICS documentation

SINAMICS documentation

The SINAMICS documentation is organized in the following categories:

- General documentation/catalogs
- User documentation
- Manufacturer/service documentation

Standard scope

The scope of the functionality described in this document can differ from that of the drive system that is actually supplied.

- Other functions not described in this documentation might be able to be executed in the drive system. However, no claim can be made regarding the availability of these functions when the equipment is first supplied or in the event of service.
- The documentation can also contain descriptions of functions that are not available in a particular product version of the drive system. Please refer to the ordering documentation only for the functionality of the supplied drive system.
- Extensions or changes made by the machine manufacturer must be documented by the machine manufacturer.

For reasons of clarity, this documentation does not contain all of the detailed information on all of the product types, and cannot take into consideration every conceivable type of installation, operation and service/maintenance.

Target group

This documentation is intended for machine manufacturers, commissioning engineers, and service personnel who use the SINAMICS drive system.

Benefits

This manual provides all of the information, procedures and operator actions required for the particular usage phase.

Siemens MySupport/Documentation

You can find information on how to create your own individual documentation based on Siemens content and adapt it for your own machine documentation at the following address.

Additional information

You can find information on the topics below at the following address:

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

Questions relating to the technical documentation

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

FAQs

You can find Frequently Asked Questions about SINAMICS under Product Support.

Siemens Support while on the move

With the "Siemens Industry Online Support" app, you can access more than 300,000 documents for Siemens Industry products – any time and from anywhere. The app supports you in the following areas, for example:

- Resolving problems when executing a project
- Troubleshooting when faults develop
- Expanding a system or planning a new system

Furthermore, you have access to the Technical Forum and other articles that our experts have drawn up:

- FAQs
- Application examples
- Manuals
- Certificates
- Product announcements and much more

The "Siemens Industry Online Support" app is available for Apple iOS and Android.

Data matrix code on the rating plate

The data matrix code on the rating plate contains the specific device data. This code can be read-in with any smartphone and technical information for the appropriate device can be displayed via the "Siemens Industry Online Support" mobile app.

Websites of third-party companies

This document includes hyperlinks to websites of third-party companies. Siemens is not responsible for and shall not be liable for these websites or their content, as Siemens has not checked the information contained in the websites and is not responsible for the content or information they provide. The use of such websites is at the user's own risk.

2.4 Usage phases and their documents/tools

Usage phase	Document/tool
Orientation	SINAMICS S Sales Documentation
Planning/configuration	• SIZER Engineering Tool • Configuration Manuals, Motors
Deciding/ordering	SINAMICS S120 catalogs • SINAMICS S120 and SIMOTICS (Catalog D 21.4) • SINAMICS Converters for Single-Axis Drives and SIMOTICS Motors (Catalog D 31) • SINAMICS Converters for Single-Axis Drives – Built-In Units (D 31.1) • SINAMICS Converters for Single-Axis Drives – Distributed Converters (D 31.2) • SINAMICS S210 Servo Drive System (D 32) • SINUMERIK 840 Equipment for Machine Tools (Catalog NC 62)
Installation/assembly	• SINAMICS S120 Equipment Manual for Control Units and Supplementary System Components • SINAMICS S120 Equipment Manual for Booksize Power Units • SINAMICS S120 Equipment Manual for Chassis Power Units, Air-cooled • SINAMICS S120 Equipment Manual for Chassis Power Units, Liquid-cooled • SINAMICS S120 Equipment Manual for Chassis Power Units, Water-cooled for common cooling circuits • SINAMICS S120 Equipment Manual for AC Drives • SINAMICS S120 Equipment Manual Combi • SINAMICS S120M Equipment Manual Distributed Drive Technology • SINAMICS HLA System Manual Hydraulic Drives
Programming Drive Control Charts	• Configuring SINAMICS DCC (TIA V17) • SINAMICS DCC Getting Started (TIA V16) • SINAMICS DCC standard blocks
Commissioning	• Startdrive Commissioning Tool • SINAMICS S120 Getting Started with Startdrive • SINAMICS S120 Commissioning Manual with Startdrive • SINAMICS S120 Function Manual Drive Functions • SINAMICS S120 Safety Integrated Function Manual • SINAMICS S120 Function Manual Communication • SINAMICS S120/S150 List Manual • SINAMICS HLA System Manual Hydraulic Drives

Usage phase	Document/tool
Usage/operation	• SINAMICS S120 Commissioning Manual with Startdrive • SINAMICS S120/S150 List Manual • SINAMICS HLA System Manual Hydraulic Drives
Maintenance/servicing	• SINAMICS S120 Commissioning Manual with Startdrive • SINAMICS S120/S150 List Manual
References	• SINAMICS S120/S150 List Manual

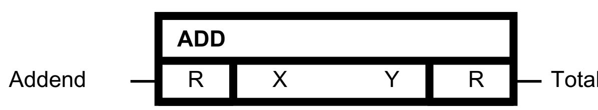
Arithmetic

3

3.1 ADD

Adder (REAL type)

Symbol



Brief description

Adder with up to four inputs of the REAL type

Method of operation

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	Default	Value range	Attrib-utes
X	Addend	0.0	REAL	
Y	Total	0.0	REAL	

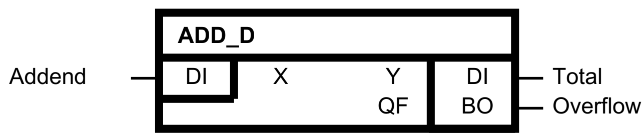
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Can be configured in	Cyclic tasks
Special character-istics	X comprises up to four inputs (X1 to X4)

3.2 ADD_D

Adder (double integer type)

Symbol



Brief description

Adder with up to four inputs of the double integer type

Method of operation

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

Algorithm:

Y = X1 + X2 + X3 + X4

Block connections

Block connection	Description	Default	Value range	Attrib-utes
X	Addend	0	DINT	
Y	Total	0	DINT	
QF	Overflow	0	0/1	

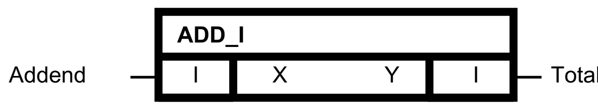
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	X comprises up to four inputs (X1 to X4)

3.3 ADD_I

Adder (integer type)

Symbol



Brief description

Adder with up to four inputs of the integer type

Method of operation

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	Default	Value range	Attrib-utes
X	Addend	0	INT	
Y	Total	0	INT	

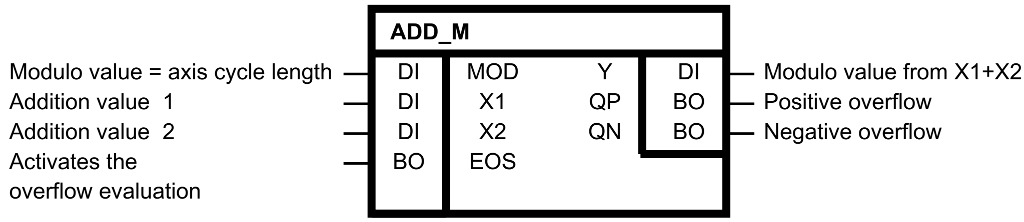
Configuration data

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

3.4 ADD_M

Modulo adder for addition in correct axis cycle

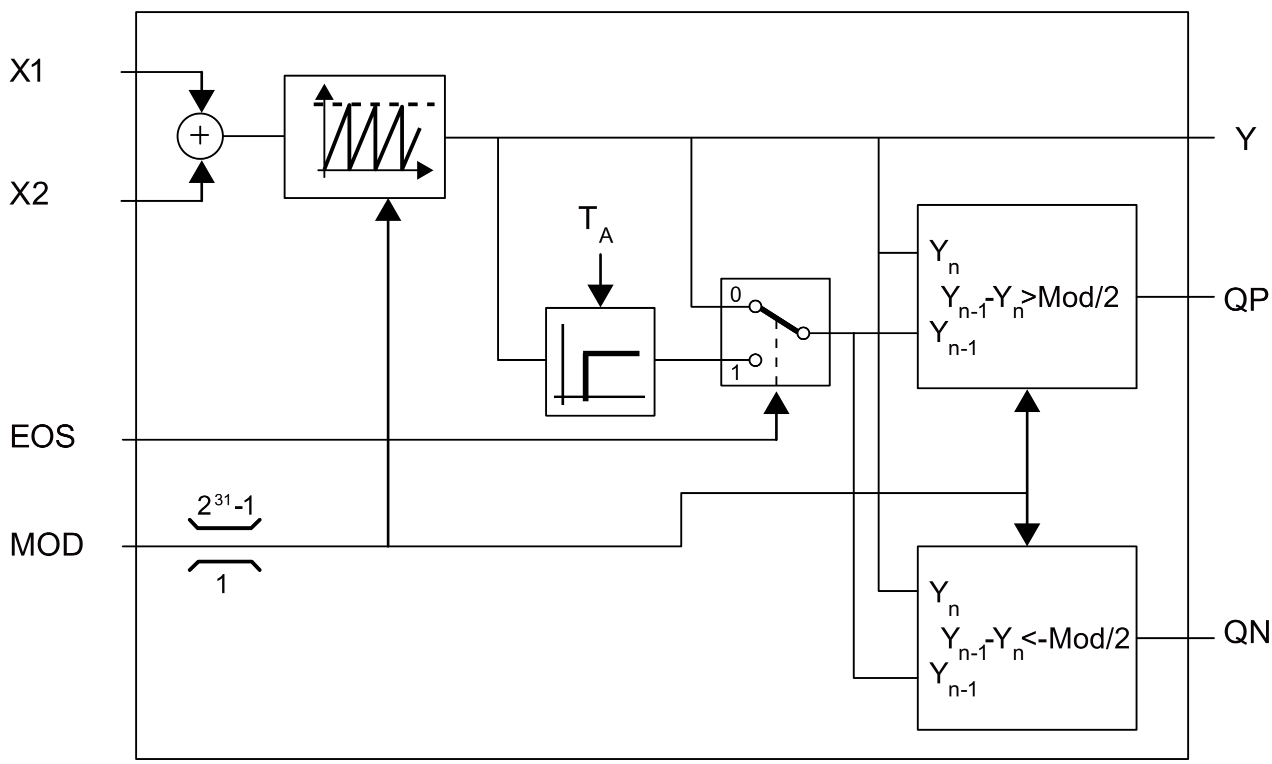
Symbol



Brief 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



Method of operation

This block adds the input values X1 and X2. A modulo value can be specified at the MOD input. The module value must be in the range of 1 to 2^{31}-1 and is 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 = 1:

Positive 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

This enables the overflow evaluation to be deactivated when setting offsets. When the modulo value is changed, the overflow evaluation is switched off for one cycle.

Block connections

Block connection	Description	Default	Value range	Attrib-utes
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	

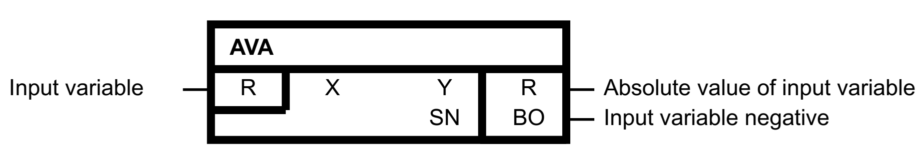
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special character-istics	-

3.5 AVA

Absolute value generator, with sign evaluation

Symbol



Brief description

Arithmetic function block for absolute value generation of type real

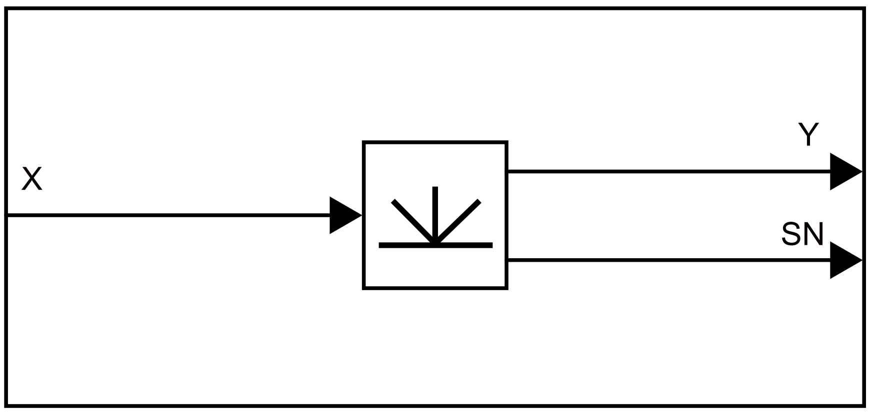
Method of operation

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	Default	Value range	Attrib-utes
X	Input variable	0.0	REAL	
Y	Absolute value of input variable	0.0	REAL	
SN	Input variable negative	0	0/1	

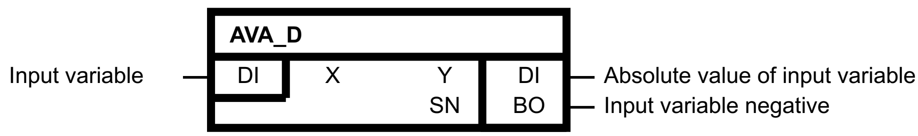
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

3.6 AVA_D

Absolute value generator (double integer)

Symbol



Brief description

Arithmetic function block for absolute value generation of type DOUBLE INTEGER

Method of operation

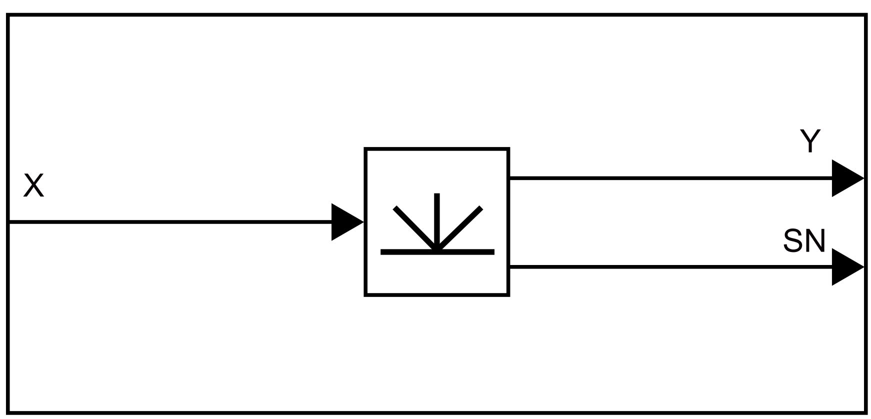
The block generates the absolute value of the value at input X (input variable). The result is output at output Y.

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	Default	Value range	Attrib-utes
X	Input variable	0	DINT	
Y	Absolute value of input variable	0	DINT	
SN	Input variable negative	0	0/1	

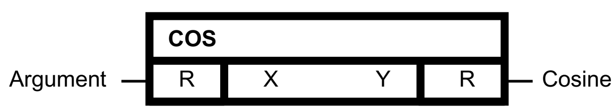
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

3.7 COS

Cosine function

Symbol



Brief description

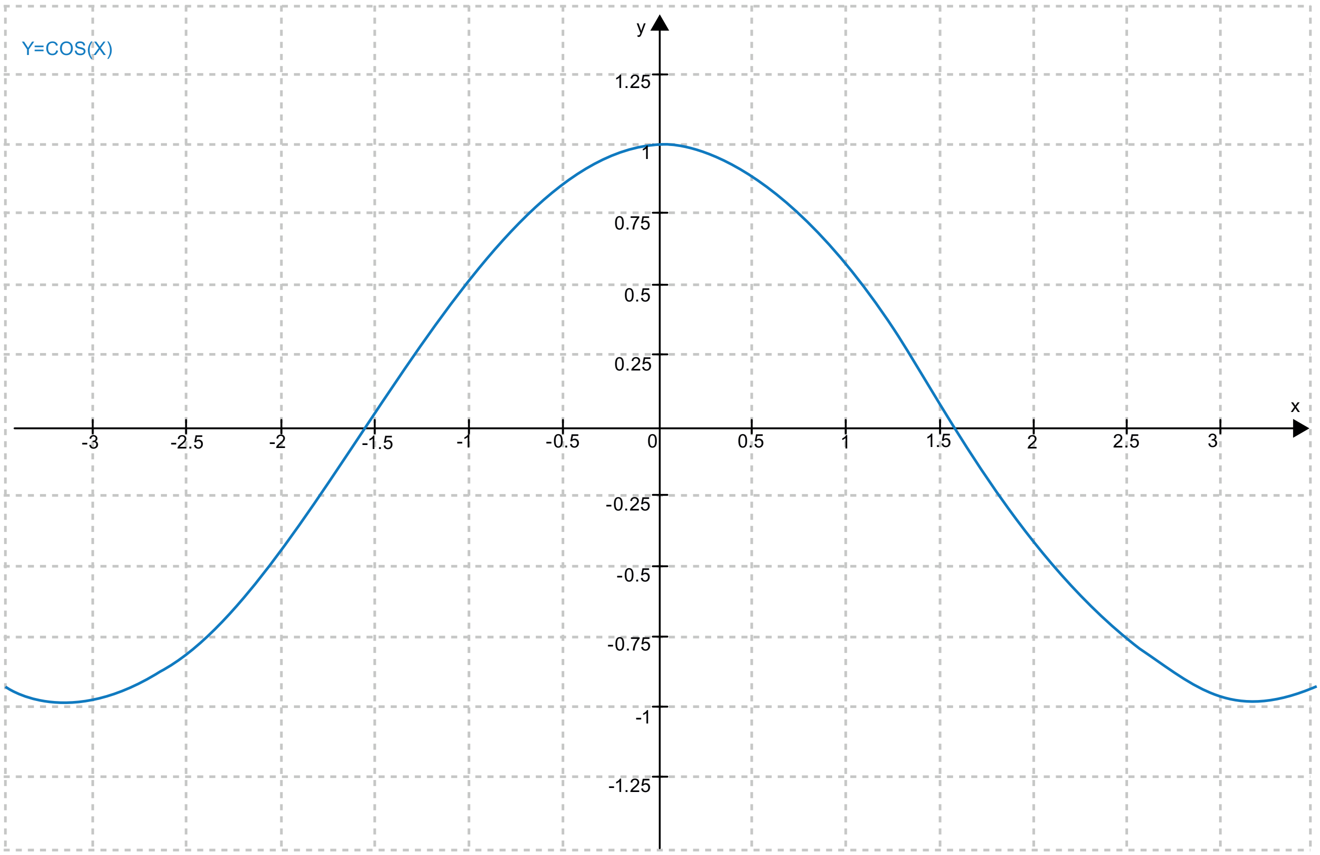
Determination of the cosine value for an argument

Method of operation

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 ∏

Block connections

Block connection	Description	Default	Value range	Attrib-utes
X	Argument	0.0	REAL	
Y	Cosine	1	REAL	

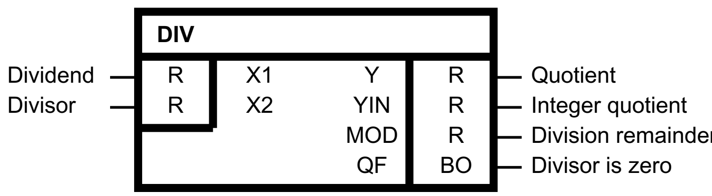
Configuration data

SIMOTION	✓ (as of V4.1)
SINAMICS	✓ (as of V4.4)
Can be loaded on-line	Yes
Special character-istics	-

3.8 DIV

Divider (REAL type)

Symbol



Brief description

Divider with two real-type inputs

Method of operation

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 a range of approximately -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 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	Default	Value range	Attrib-utes
X1	Dividend	0.0	REAL	
X2	Divisor	1	REAL	
Y	Quotient	0.0	REAL	
YIN	Integer quotient	0.0	REAL	
MOD	Division remainder	0.0	REAL	
QF	Divisor is zero	0	0/1	

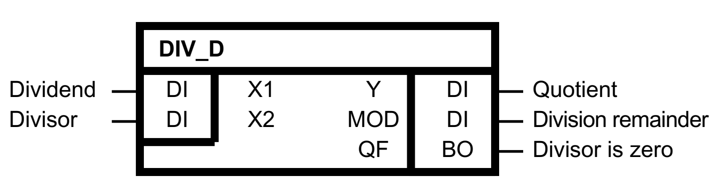
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special character-istics	-

3.9 DIV_D

Divider (double integer type)

Symbol



Brief description

Divider with two inputs of the double integer type

Method of operation

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

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

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

MOD = X1 MOD X2

When output value Y exceeds the permissible range of -2147483648 (2^{31}) to +2147483647 (2^{31}-1) (because divisor X2 is zero), then the limit value of the output range with the correct sign is output at connection Y. 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	Default	Value range	Attrib-utes
X1	Dividend	0	DINT	
X2	Divisor	1	DINT	
Y	Quotient	0	DINT	
MOD	Division remainder	0	DINT	
QF	Divisor is zero	0	0/1	

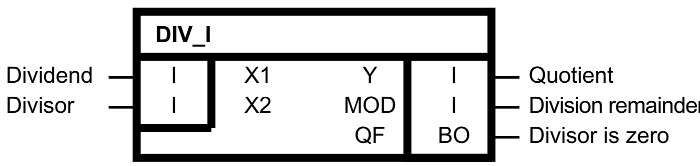
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special character-istics	-

3.10 DIV_I

Divider (integer type)

Symbol



Brief description

Divider with two integer-type inputs

Method of operation

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

MOD = X1 MOD X2

When output value Y exceeds the permissible range of -32768 to +32767 (because the divisor is zero), then the limit value of the output range with the correct sign is output at connection Y. 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	Default	Value range	Attrib-utes
X1	Dividend	0	INT	
X2	Divisor	1	INT	
Y	Quotient	0	INT	
MOD	Division remainder	0	INT	
QF	Divisor is zero	0	0/1	

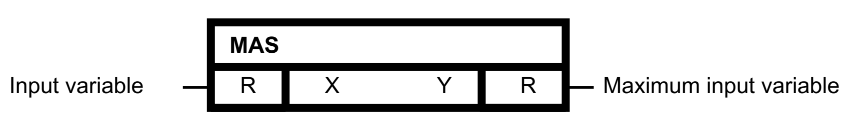
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special character-istics	-

3.11 MAS

Maximum evaluator

Symbol



Brief description

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

Method of operation

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	Default	Value range	Attrib-utes
X	Input variable	-3.402823 E38	REAL	
Y	Maximum input variable	0.0	REAL	

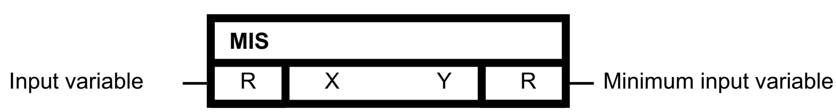
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	X comprises up to four inputs (X1 to X4)

3.12 MIS

Minimum evaluator

Symbol



Brief description

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

Method of operation

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	Default	Value range	Attrib-utes
X	Input variable	3.402823 E38	REAL	
Y	Minimum input variable	0.0	REAL	

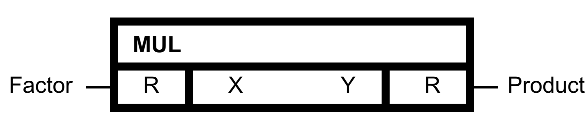
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	X comprises up to four inputs (X1 to X4)

3.13 MUL

Multiplier (REAL type)

Symbol



Brief description

Multiplier with up to four real-type inputs

Method of operation

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 x X2 x X3 x X4

Block connections

Block connection	Description	Default	Value range	Attrib-utes
X	Factor	1.0	REAL	
Y	Product	0.0	REAL	

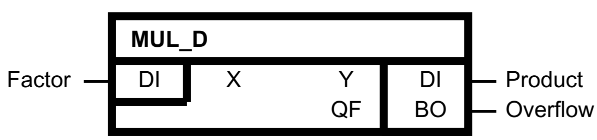
Configuration data

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

3.14 MUL_D

Multiplier (double integer type)

Symbol



Brief description

Multiplier with up to four double integer-type inputs

Method of operation

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

Y = X1 x X2 x X3 x X4

Block connections

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

Configuration data

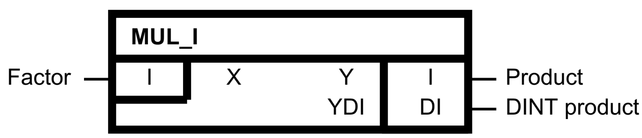
SIMOTION	✓
SINAMICS	✓

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

3.15 MUL_I

Multiplier (integer type)

Symbol



Brief description

Multiplier with up to four integer-type inputs

Method of operation

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 x X2 x X3 x X4

Block connections

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

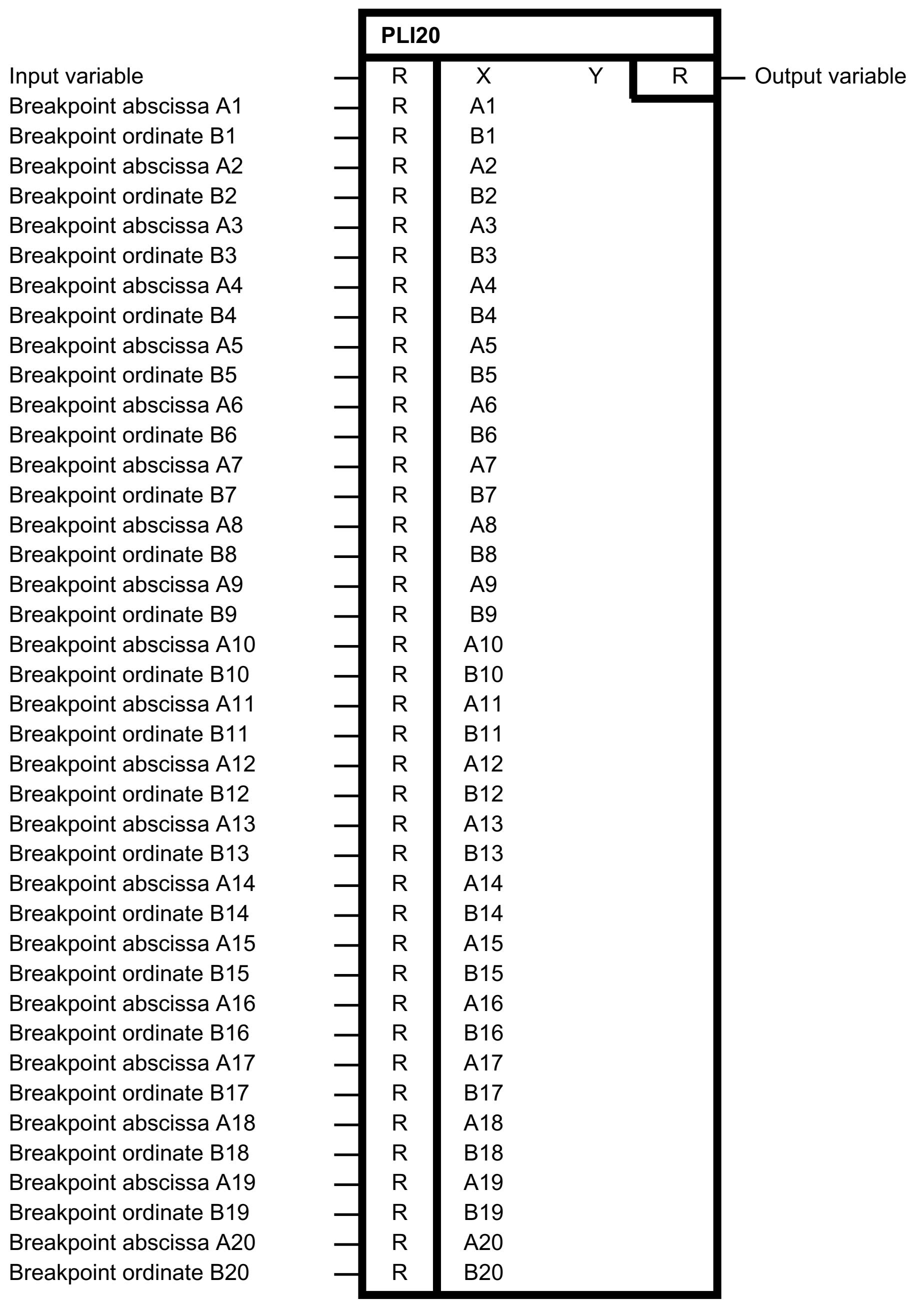
Configuration data

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

3.16 PLI20

Polyline, 20 breakpoints

Symbol



Brief description

Block of the REAL type

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

Method of operation

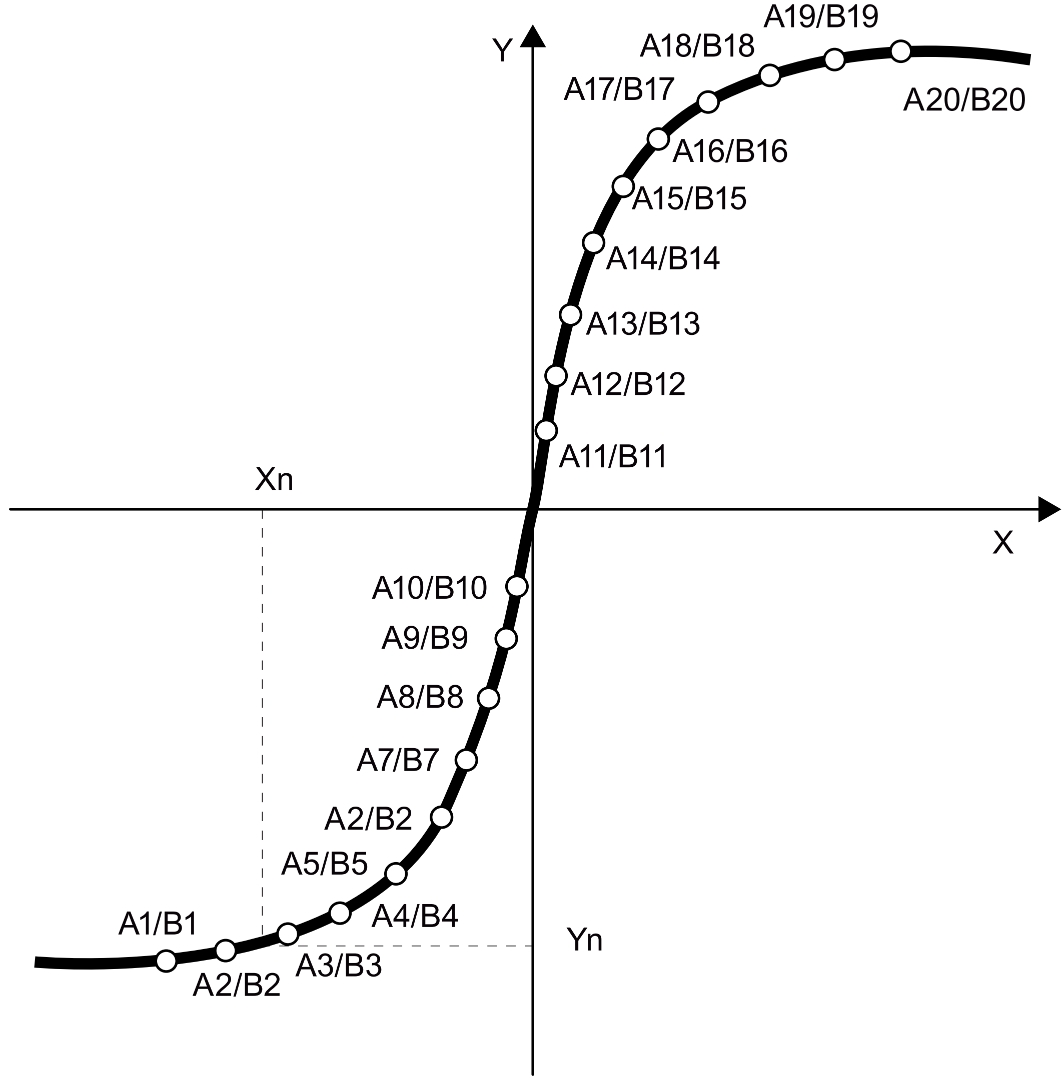
- The block adapts output variable Y arbitrarily to input variable X by means of up to 20 breakpoints in four quadrants.
- Interpolation is carried out linearly between the breakpoints. Outside of A1 and A20, the characteristic curve runs horizontally.

Configuration notes

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 the magnetization characteristic curve

Block connections

Block connection	Description	Default	Value range	Attrib-utes
X	Input variable	0.0	REAL	
A1	Breakpoint abscissa A1	0.0	REAL	
B1	Breakpoint ordinate B1	0.0	REAL	
O2	Breakpoint abscissa A2	0.0	REAL	
B2	Breakpoint ordinate B2	0.0	REAL	
O3	Breakpoint abscissa A3	0.0	REAL	
B3	Breakpoint ordinate B3	0.0	REAL	

Block connection	Description	Default	Value range	Attrib-utes
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	

Configuration data

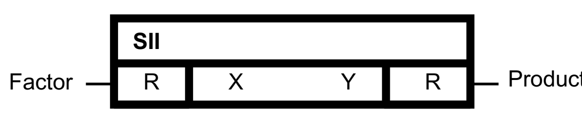
SIMOTION	✓
SINAMICS	✓

Can be loaded online	Yes
Special characteristics	-

3.17 SII

Inverter

Symbol



Brief description

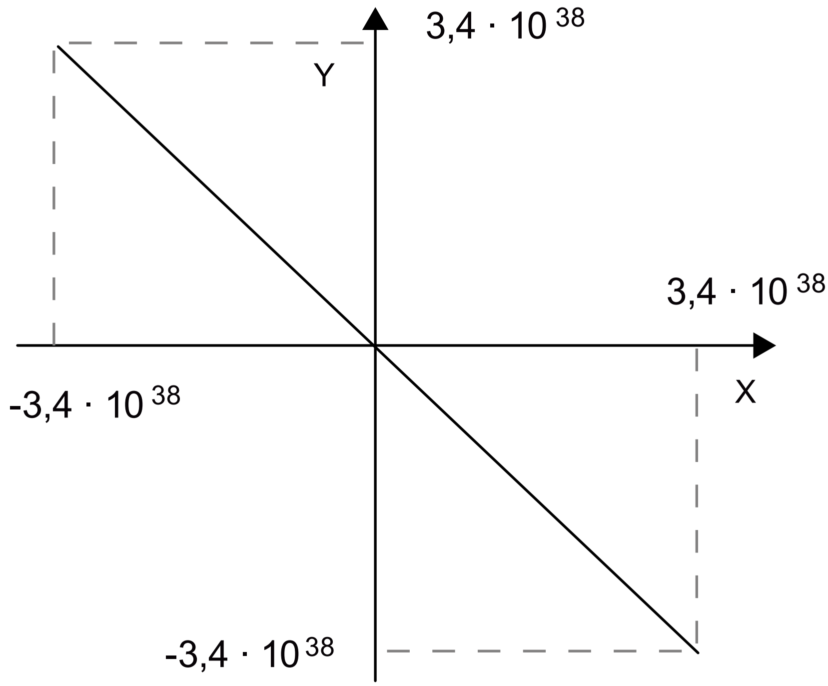
- Inverter with one Real-type input
- Arithmetic function block for sign reversal

Method of operation

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	Default	Value range	Attrib-utes
X	Factor	0.0	REAL	
Y	Product	0.0	REAL	

Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

3.18 SIN

Sine function

Symbol

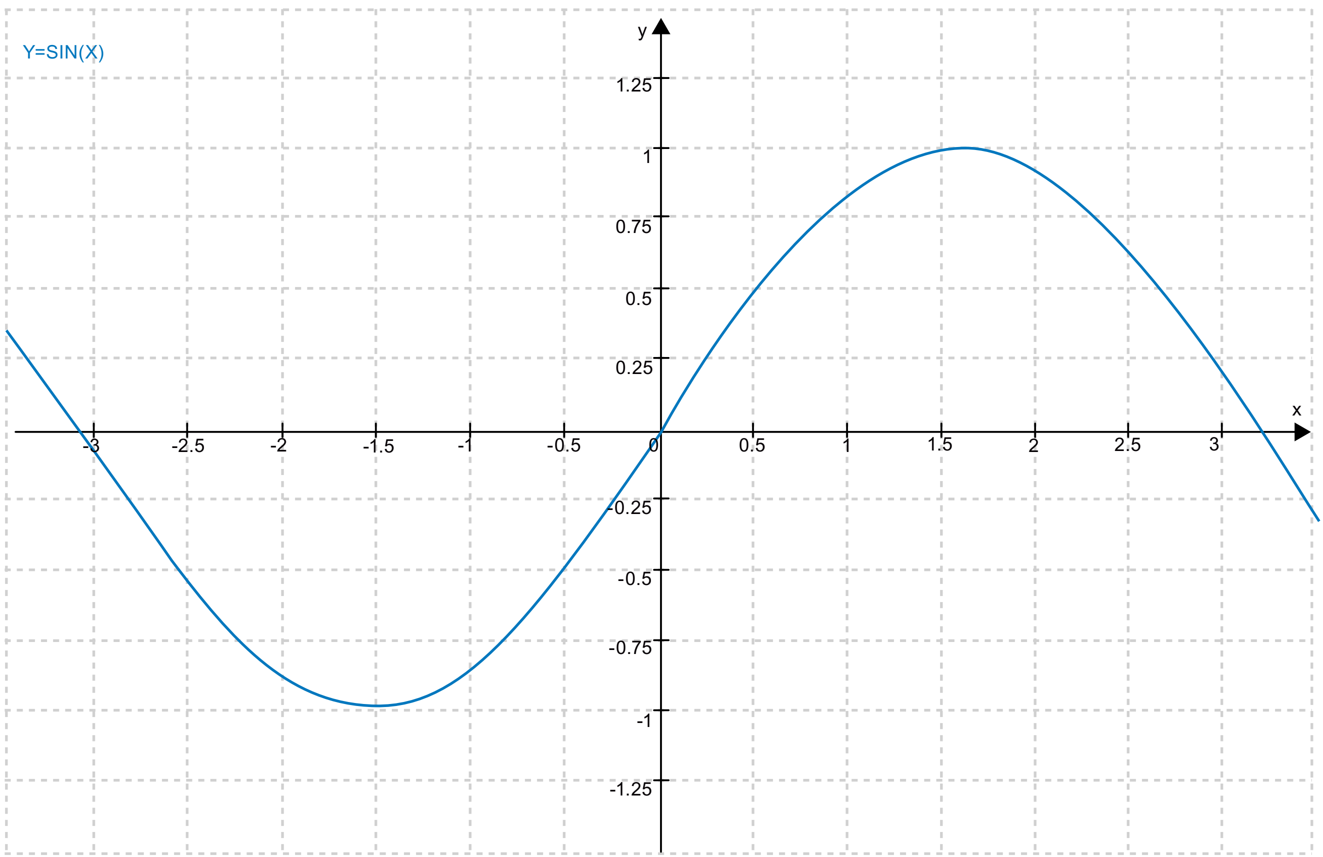
Brief description

Determination of the sine value for an argument

Method of operation

- 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 ∏

Block connections

Block connection	Description	Default	Value range	Attrib-utes
X	Argument	0.0	REAL	
Y	Sinusoidal	0.0	REAL	

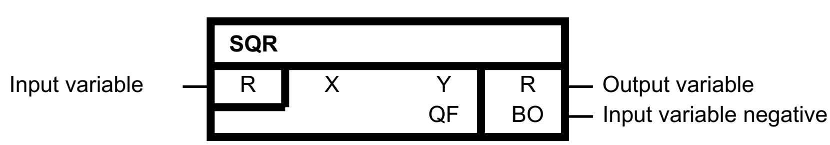
Configuration data

SIMOTION	✓ (as of V4.1)
SINAMICS	✓ (as of V4.4)
Can be loaded on-line	Yes
Special character-istics	-

3.19 SQR

Square-root extractor

Symbol



Brief description

Arithmetic function block for determining the square root

Method of operation

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)

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

Block connections

Block connection	Description	Default	Value range	Attrib-utes
X	Input variable	0.0	REAL	
Y	Output variable	0.0	REAL	
QF	Input variable negative	0	0/1	

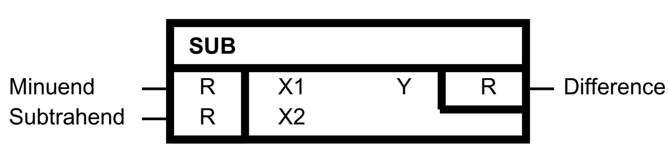
Configuration data

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

3.20 SUB

Subtractor (REAL type)

Symbol



Brief description

Subtractor with two Real-type inputs

Method of operation

- 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	Default	Value range	Attrib-utes
X1	Minuend	0.0	REAL	
X2	Subtrahend	0.0	REAL	
Y	Difference	0.0	REAL	

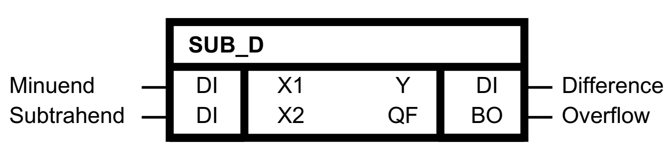
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

3.21 SUB_D

Subtractor (DOUBLE INTEGER type)

Symbol



Brief description

Subtractor with two inputs of the DOUBLE INTEGER type

Method of operation

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 -2147483648 (2^{31}) to +2147483647 (2^{31}-1), is output at output Y. An overflow is indicated at the binary output with QF=1.

Y= X1-X2

Block connections

Block connection	Description	Default	Value range	Attrib-utes
X1	Minuend	0	DINT	
X2	Subtrahend	0	DINT	
Y	Difference	0	DINT	
QF	Overflow	0	0/1	

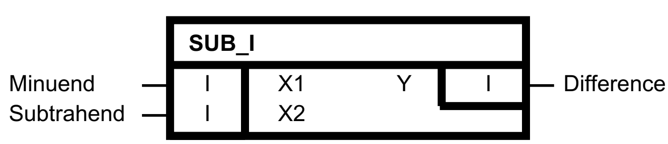
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

3.22 SUB_I

Subtractor (INTEGER type)

Symbol



Brief description

Subtractor with two inputs of the INTEGER type

Method of operation

- 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	Default	Value range	Attrib-utes
X1	Minuend	0	INT	
X2	Subtrahend	0	INT	
Y	Difference	0	INT	

Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

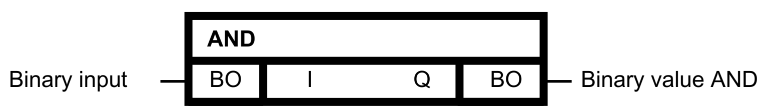
Logic

4

4.1 AND

Logic AND operation (BOOL type)

Symbol



Brief description

AND block with up to four inputs of the BOOL type

Method of operation

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

* Arbitrary

Block connections

Block connection	Description	Default	Value range	Attrib-utes
I	Binary input	1	0/1	
Q	Binary value AND	0	0/1	

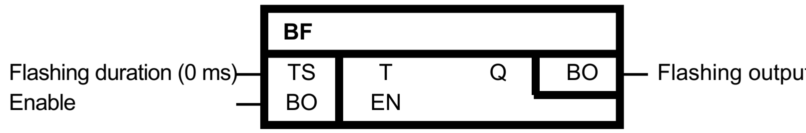
Configuration data

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

4.2 BF

Flashing function (BOOL type)

Symbol



Brief description

Block of the BOOL type

- For controlling signal encoders
- As clock generator

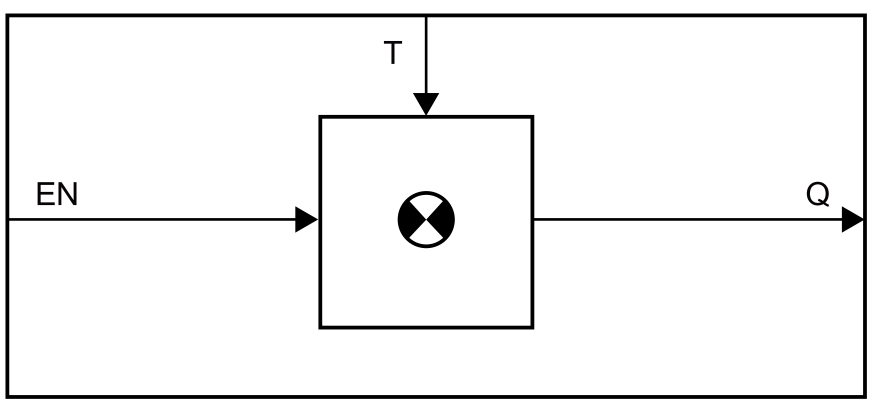
Method of operation

This block sets its output Q alternately to 1 and 0 at a frequency of interval T, as long as input EN = 1.

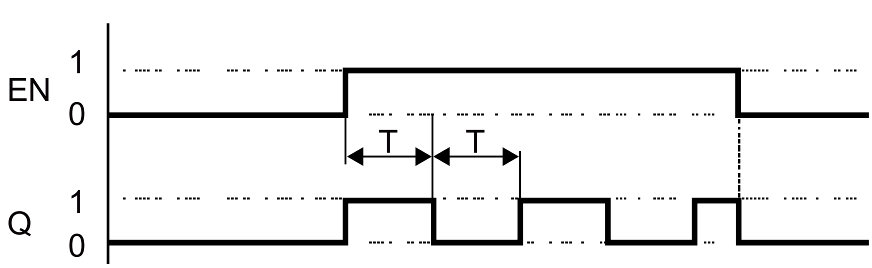
If enable input EN = 0, then output Q = 0.

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	Default	Value range	Attributes
T	Flashing duration (0 ms)	0	SDTIME	
EN	Enable	0	0/1	
Q	Flashing output	0	0/1	

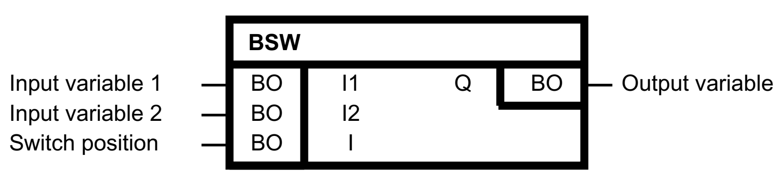
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

4.3 BSW

Binary change-over switch (BOOL type)

Symbol



Brief description

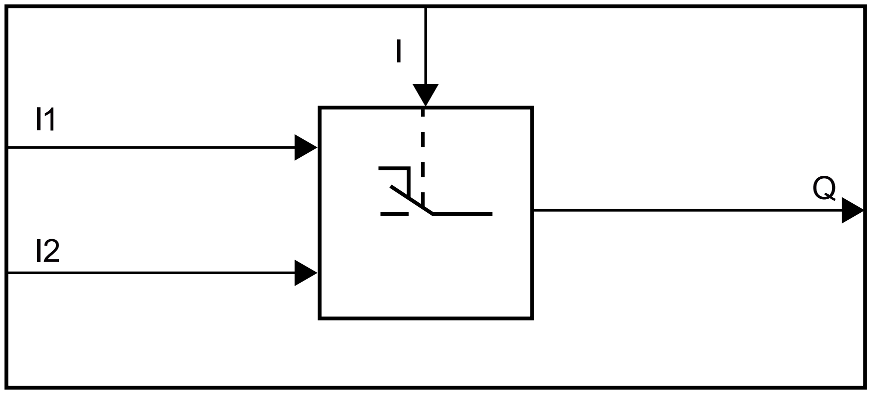
The block switches one of two binary input variables to the output.

Method of operation

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 = I1

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	Default	Value range	Attrib-utes
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	

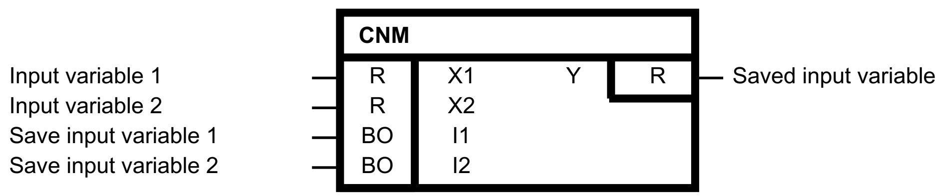
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

4.4 CNM

Controllable numeric memory (REAL type)

Symbol



Brief description

Block of the REAL type for saving an instantaneous input value (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.

Method of operation

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.

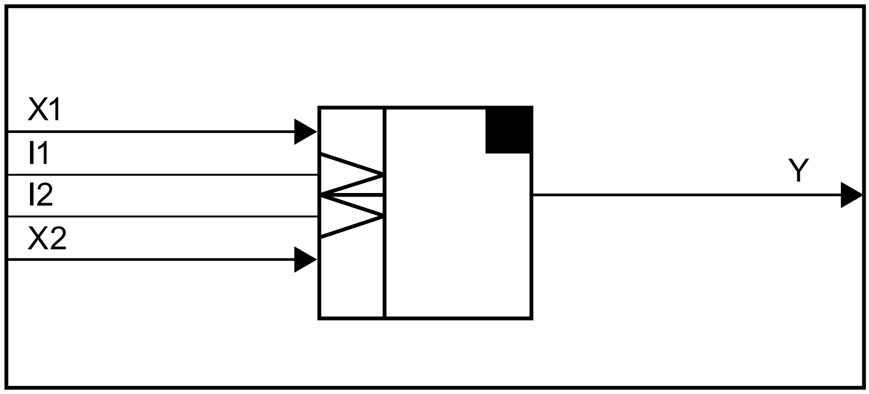
The saved input variable remains pending on Y until the next rising edge at I1 or I2 switches through the next instantaneous value.

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 memory bit), 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 edge, 0 -> 1: Rising edge

Block connections

Block connection	Description	Default	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	

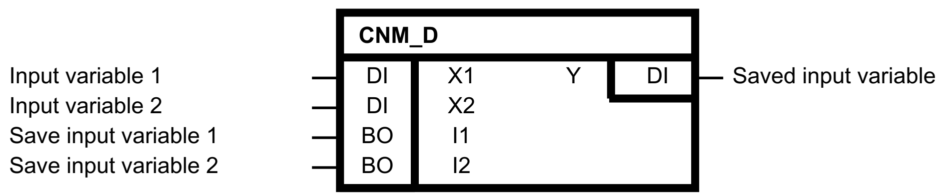
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

4.5 CNM_D

Controllable numeric memory (DOUBLE INTEGER type)

Symbol



Brief description

Block of the DOUBLE INTEGER type for saving a current input value (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.

Method of operation

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.

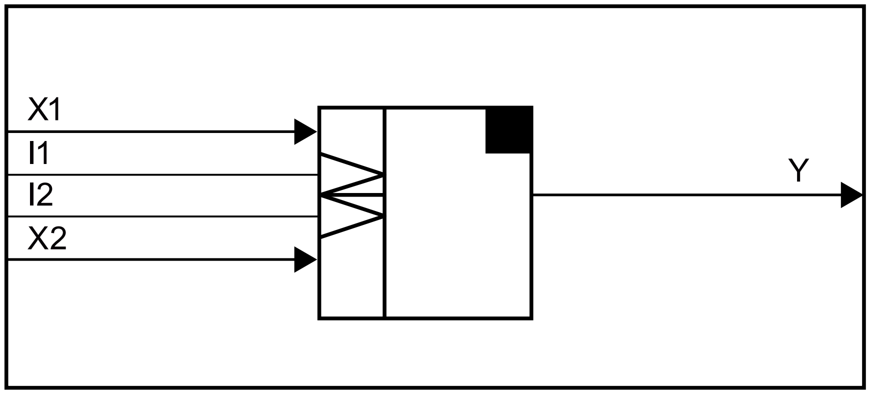
The saved input variable remains pending on Y until the next rising edge at I1 or I2 switches through the next momentary value.

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 edge, 0 -> 1: Rising edge

Block connections

Block connection	Description	Default	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	

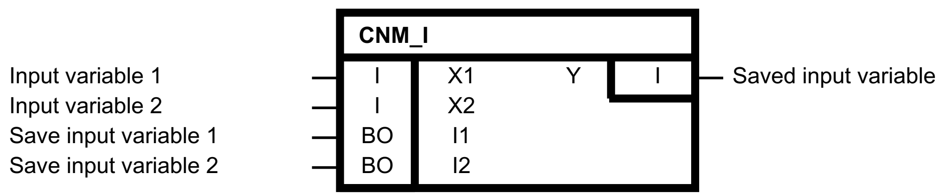
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

4.6 CNM_I

Controllable numeric memory (INTEGER type)

Symbol



Brief description

Block of the INTEGER type for saving a current input value (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.

Method of operation

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.

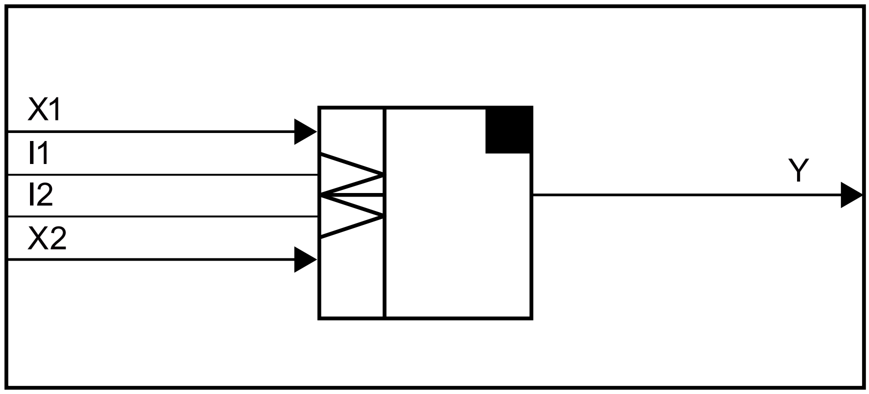
The saved input variable remains pending on Y until the next rising edge at I1 or I2 switches through the next momentary value.

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 edge, 0 -> 1: Rising edge

Block connections

Block connection	Description	Default	Value range	Attrib-utes
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	

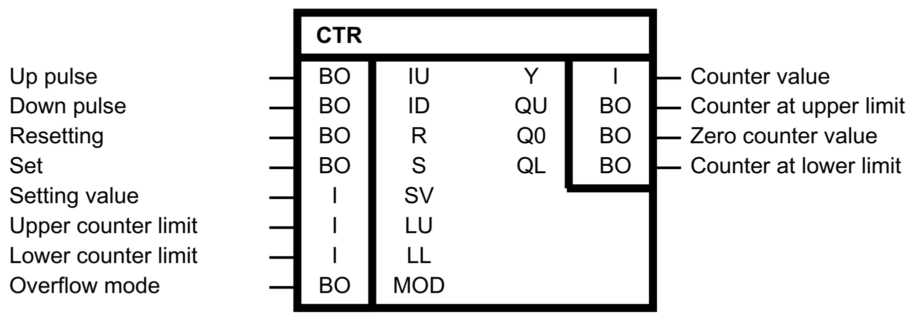
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

4.7 CTR

Counter (BOOL type)

Symbol



Brief description

Block for counting up and down 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.

Method of operation

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 held at 0. The counter is disabled. If Y is not in the counting range between LL and LU, the output is set to the active limit value when R = 1.

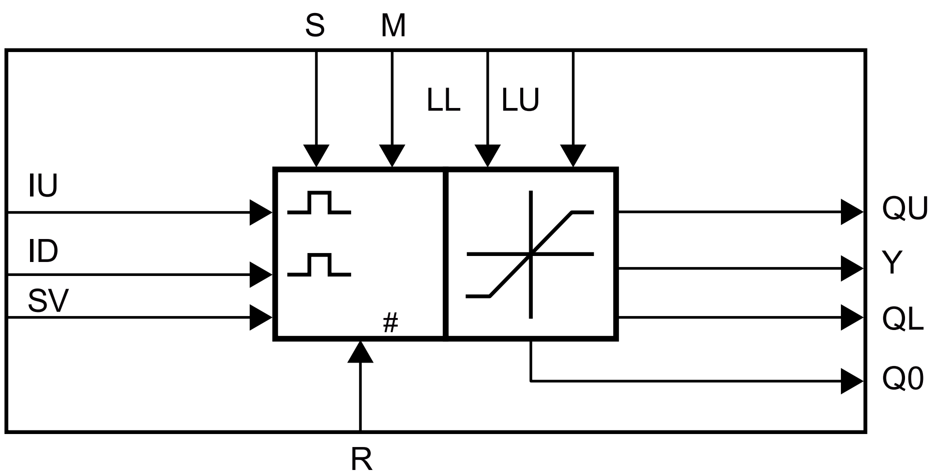
The working area of the counter can be specified via LU (upper counter limit value) or LL (lower counter limit value).

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	Default	Value range	Attrib-utes
IU	Up pulse	0	0/1	
ID	Down pulse	0	0/1	
R	Resetting	0	0/1	
S	Set	0	0/1	
SV	Setting value	0	INT	

Block connection	Description	Default	Value range	Attrib-utes
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	

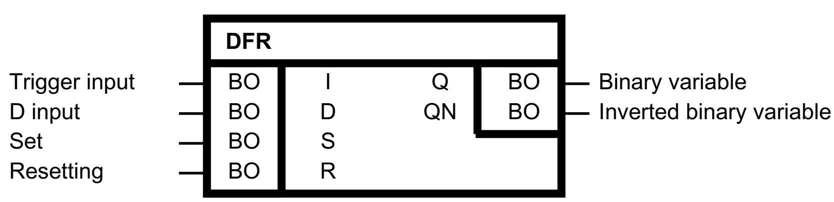
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

4.8 DFR

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

Symbol



Brief description

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

Method of operation

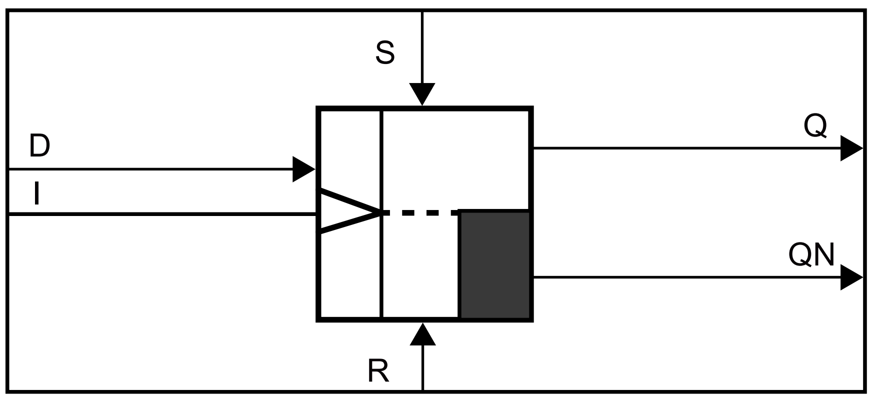
If the two inputs S and R are logic 0, the D input information is switched through to output Q on a rising edge at trigger input I. Output QN always has the value inverse to Q. With logic 1 at input S, output Q is set to logic 1. If input R is set to logic 1, then output Q is set to logic 0. If both 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.

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

Block diagram

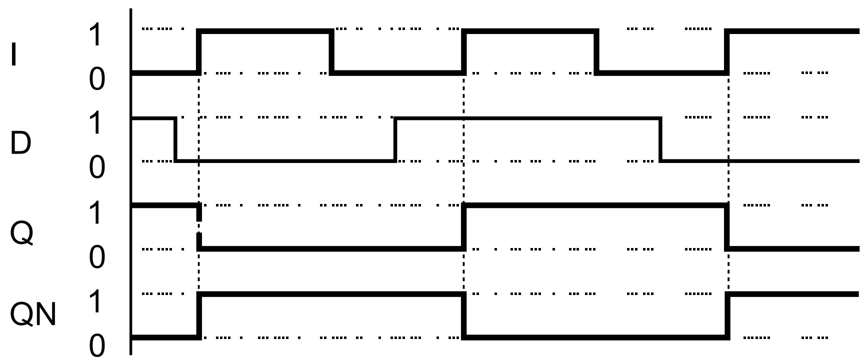


Truth table(s)

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	Default	Value range	Attrib-utes
I	Trigger input	0	0/1	
D	D input	0	0/1	
S	Set	0	0/1	
R	Resetting	0	0/1	
Q	Binary variable	0	0/1	
QN	Inverted binary variable	1	0/1	

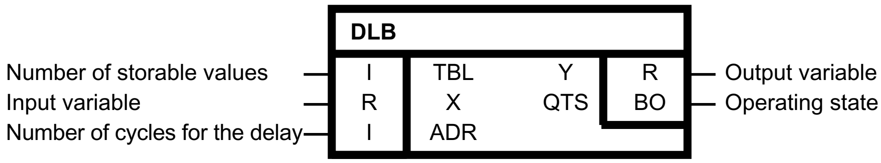
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

4.9 DLB

Delay element (REAL type)

Symbol



Brief description

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

Method of operation

If the operating state 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 1000 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 > 1000 has been defined. In this case, output Y is corrected to input X during cyclic operation.

Block connections

Block connection	Description	Default	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	INT	
QTS	Operating state	0	0/1	

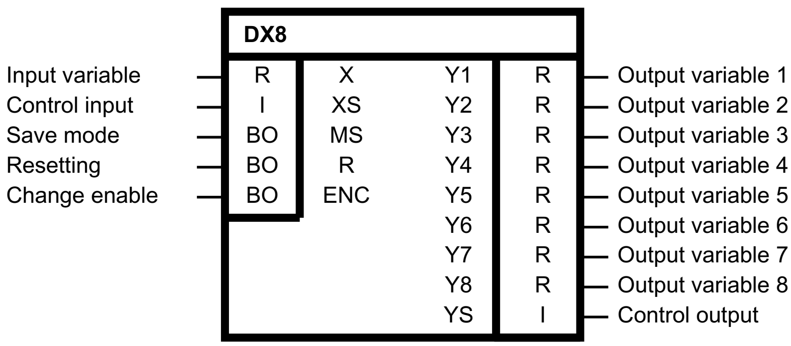
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

4.10 DX8

Demultiplexer, eight outputs, cascadable (REAL type)

Symbol



Brief description

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

Method of operation

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 • Output not selected = 0
1	0	0	XS = 0 or XS >= 9	Y1 to Y8 = 0

ENC	R	MS	XS	Outputs Y1 to Y8
1	0	1	1 <= XS <= 8	• Selected output = X • Outputs not selected remain unchanged
1	0	1	XS = 0 or XS >= 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	Default	Value range	Attrib-utes
X	Input variable	0.0	REAL	
XS	Control input	0	INT	
MS	Save mode	0	0/1	
R	Resetting	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	

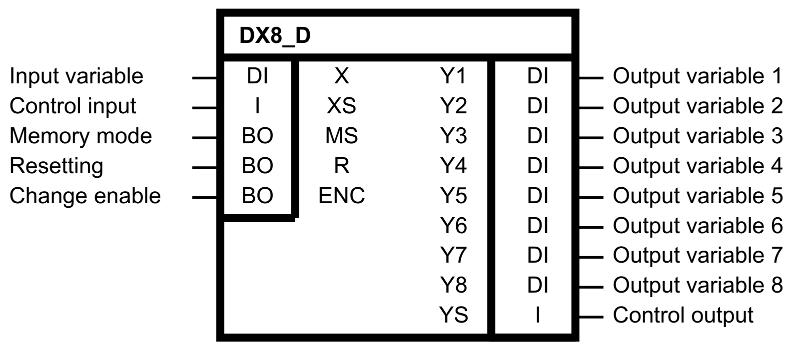
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

4.11 DX8_D

Demultiplexer, eight outputs, cascadable (DOUBLE INTEGER type)

Symbol



Brief description

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

Method of operation

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 • Output not selected = 0
1	0	0	XS = 0 or XS >= 9	Y1 to Y8 = 0

ENC	R	MS	XS	Outputs Y1 to Y8
1	0	1	1 <= XS <= 8	• Selected output = X • Outputs not selected remain 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	Default	Value range	Attrib-utes
X	Input variable	0	DINT	
XS	Control input	0	INT	
MS	Memory mode	0	0/1	
R	Resetting	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	

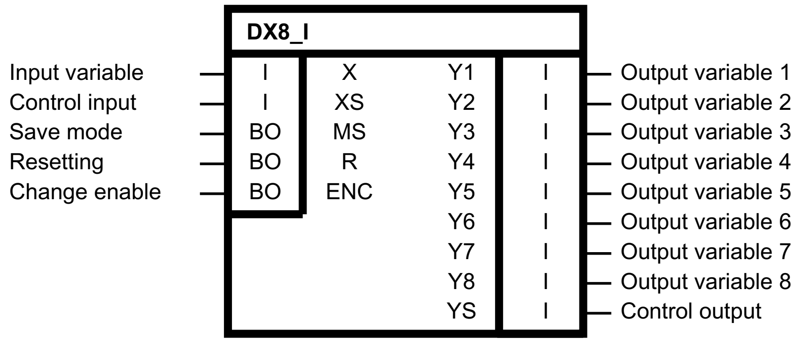
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special character-istics	-

4.12 DX8_I

Demultiplexer, eight outputs, cascadable (INTEGER type)

Symbol



Brief description

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

Method of operation

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 • Output not selected = 0
1	0	0	XS = 0 or XS >= 9	Y1 to Y8 = 0
1	0	1	1 <= XS <= 8	• Selected output = X • Outputs not selected remain 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	Default	Value range	Attrib-utes
X	Input variable	0	INT	
XS	Control input	0	INT	
MS	Save mode	0	0/1	
R	Resetting	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	

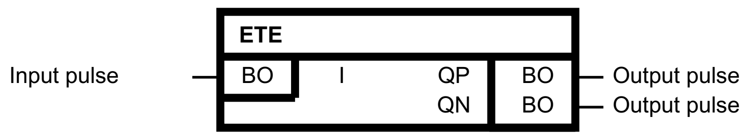
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

4.13 ETE

Edge evaluator (BOOL type)

Symbol



Brief description

Edge evaluation

Method of operation

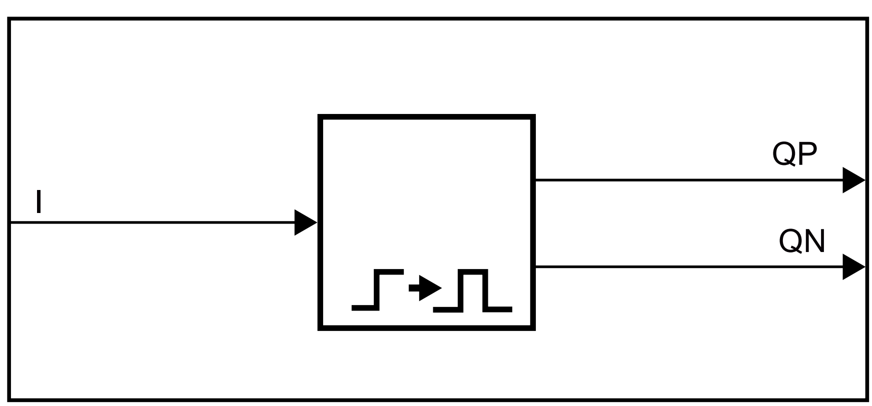
This block detects a signal change at input I. With a positive edge (0→1) at input I, output QP = 1 is set for scan time TA.

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

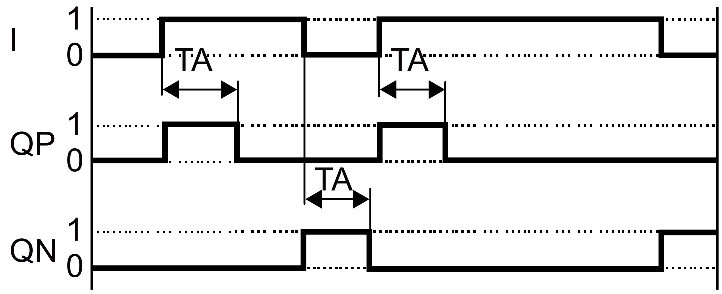
Initialization

The initialization defines the start value for the first cyclic pass. If input I receives the value 1 during initialization of an upstream block, the block cannot detect a positive edge during the first cyclic pass. If input I receives the value 0 during initialization of 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	Default	Value range	Attrib-utes
I	Input pulse	0	0/1	
QP	Output pulse	0	0/1	
QN	Output pulse	0	0/1	

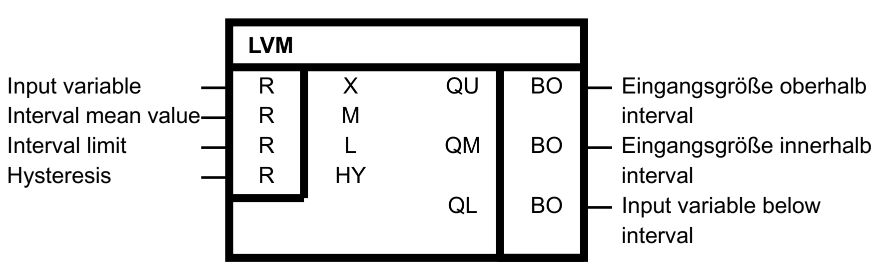
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

4.14 LVM

Double-sided limit monitor with hysteresis (BOOL type)

Symbol



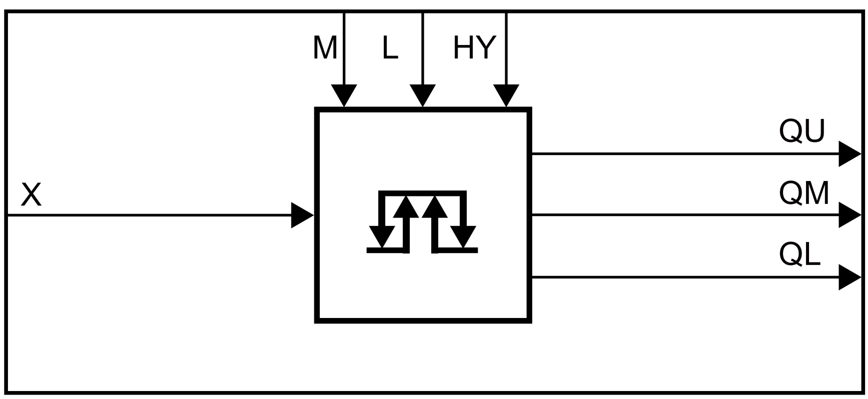
Brief description

- 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, as well as for the suppression of frequent switching (chatter).
- The block provides a window discriminator function.

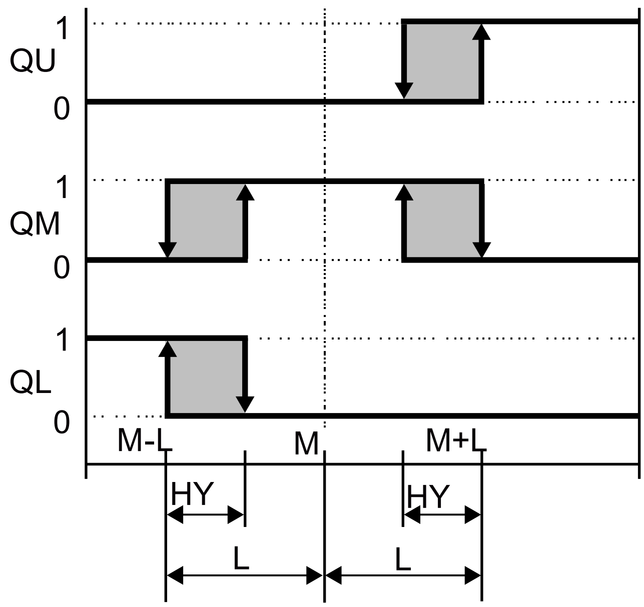
Method of operation

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	Default	Value range	Attrib-utes
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 the interval	0	0/1	
QM	Input variable above the interval	0	0/1	
QL	Input variable above the interval	0	0/1	

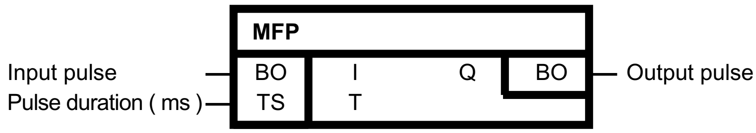
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

4.15 MFP

Pulse generator (BOOL type)

Symbol



Brief description

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

Method of operation

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.

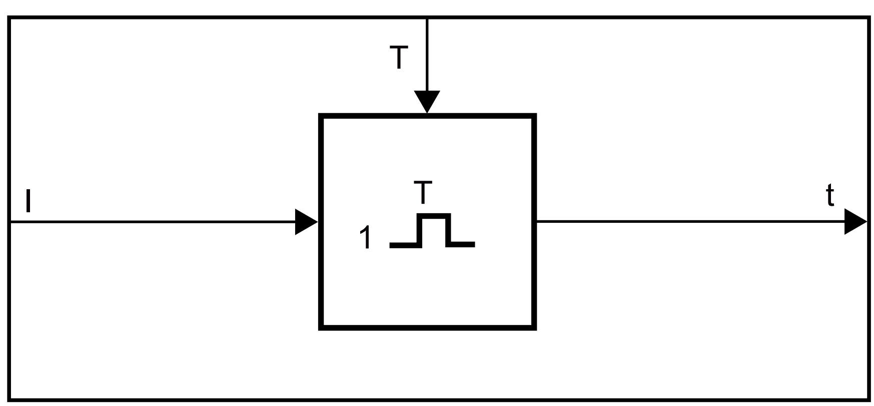
Initialization

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

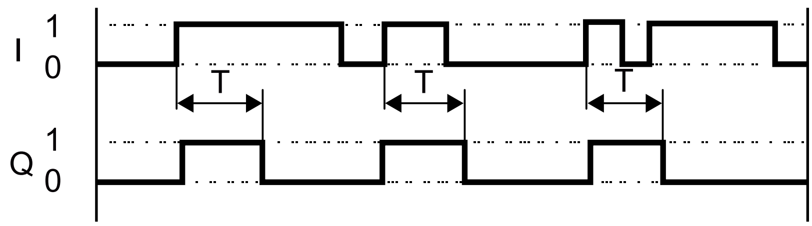
If input I receives the value 1 during initialization from the upstream block output, the block cannot detect a positive edge during the first cyclic pass.

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	Default	Value range	Attrib-utes
I	Input pulse	0	0/1	
T	Pulse duration (ms)	0	SDTIME	
Q	Output pulse	0	0/1	

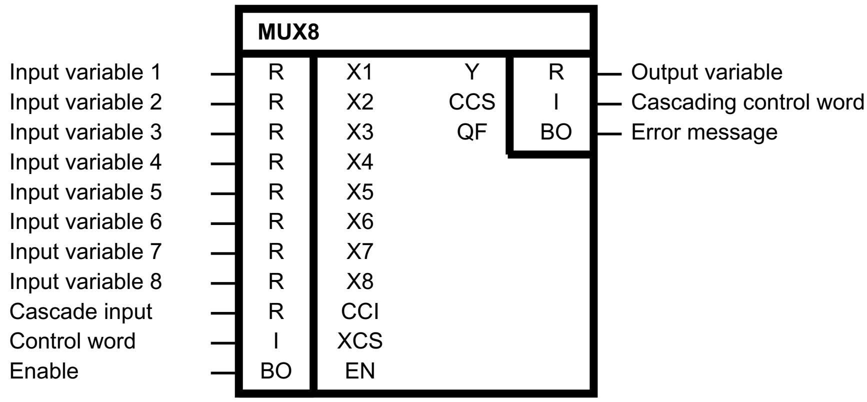
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

4.16 MUX8

Multiplexer, cascadable (REAL type)

Symbol



Brief description

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

Method of operation

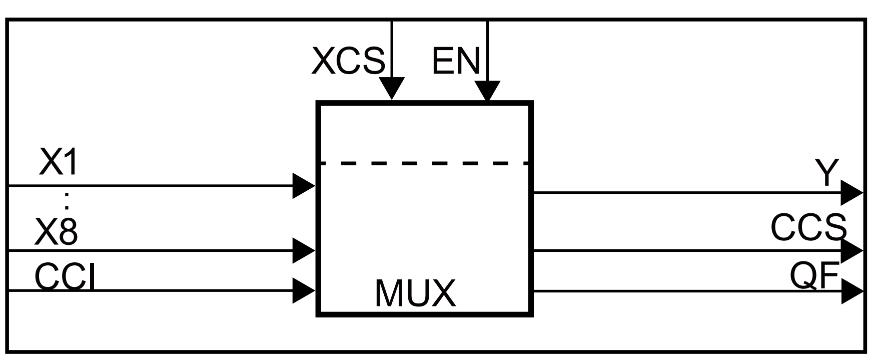
This block outputs the value of the cascading input CCI at output Y as long as the enable input EN is logic 0.

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.

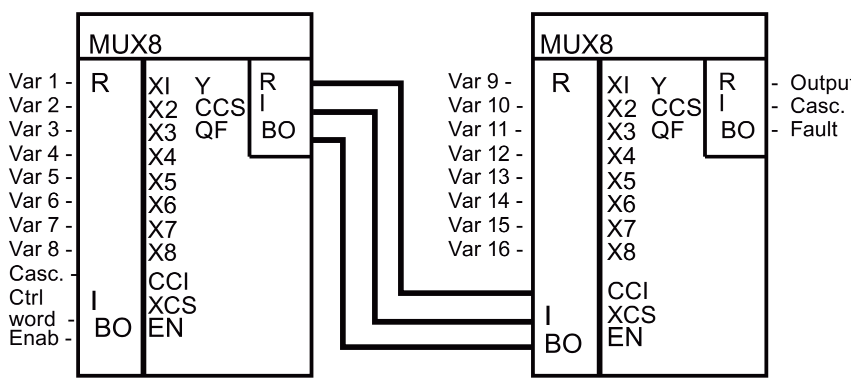
Block diagram



Truth table(s)

EN	XCS	Y	CSS	QF
0	Any	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	Default	Value range	Attrib-utes
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	Cascade input	0.0	REAL	
XCS	Control word	0	0...32767	
EN	Enable	0	0/1	
Y	Output variable	0.0	REAL	

Block connection	Description	Default	Value range	Attrib-utes
CCS	Cascading control word	0	0...32767	
QF	Error message	0	0/1	

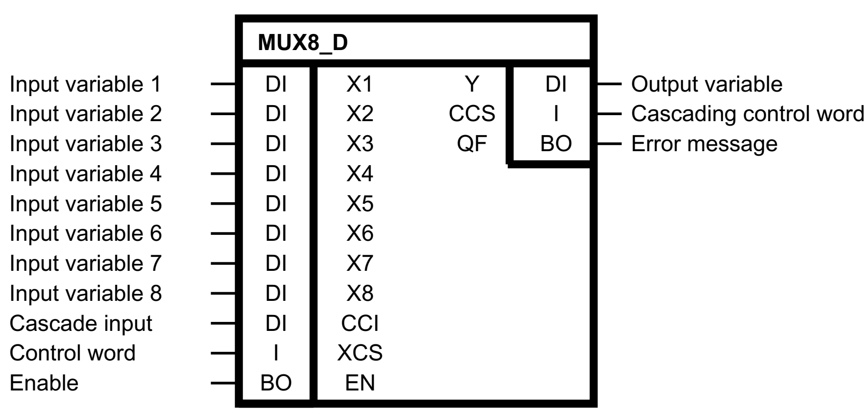
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

4.17 MUX8_D

Multiplexer, cascadable (DOUBLE-INTEGER type)

Symbol



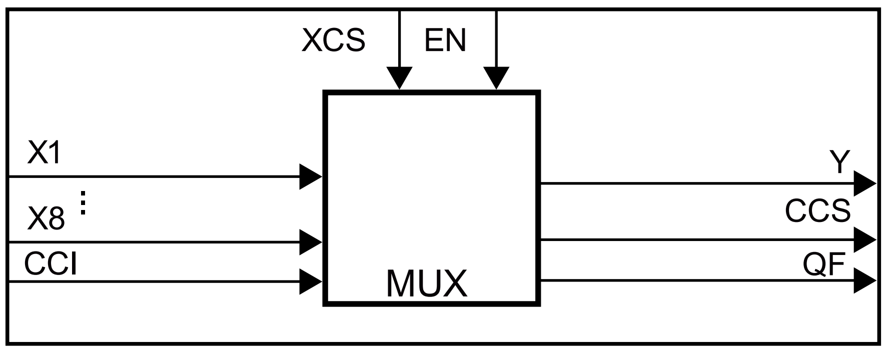
Brief description

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

Method of operation

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.

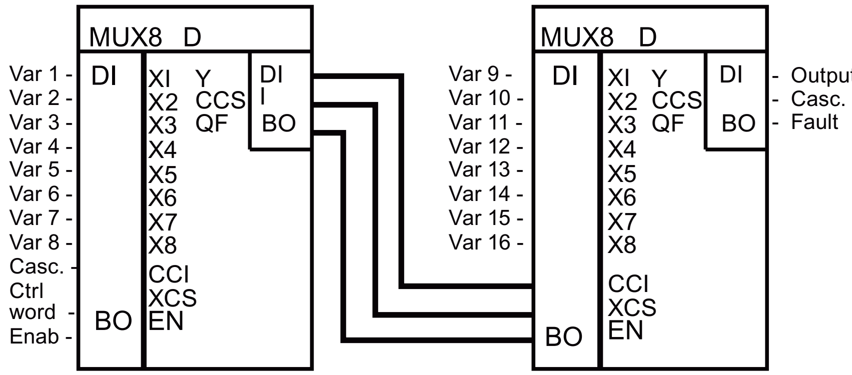
Block diagram



Truth table(s)

EN	XCS	Y	CSS	QF
0	Any	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	Default	Value range	Attrib-utes
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	Cascade 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	

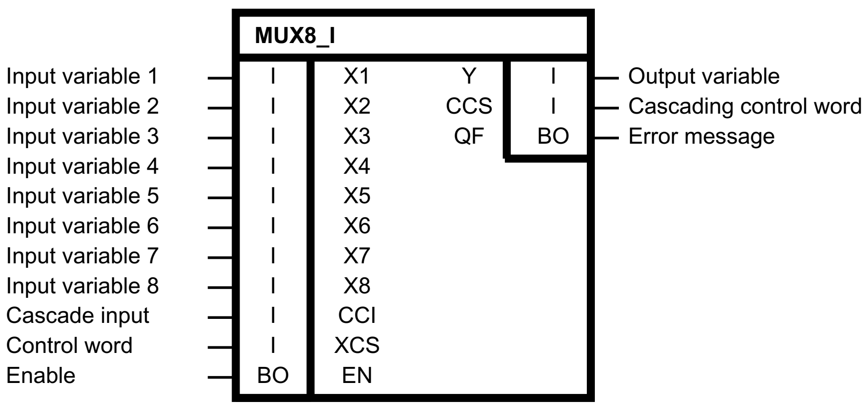
Configuration data

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

4.18 MUX8_I

Multiplexer, cascadable (INTEGER type)

Symbol



Brief description

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

Method of operation

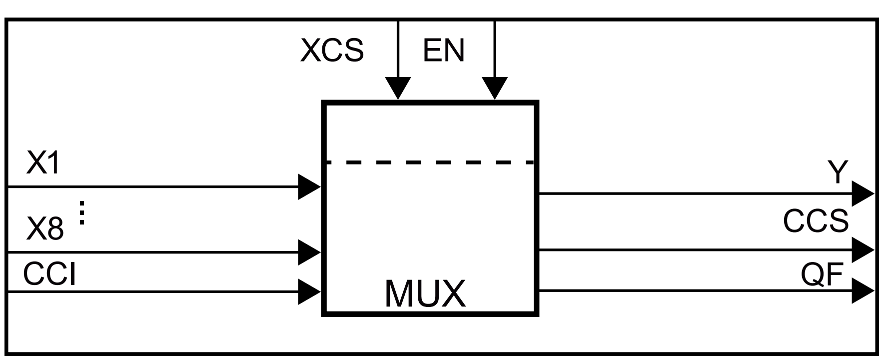
This block outputs the value of the cascading input CCI at output Y as long as the enable input EN is logic 0.

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.

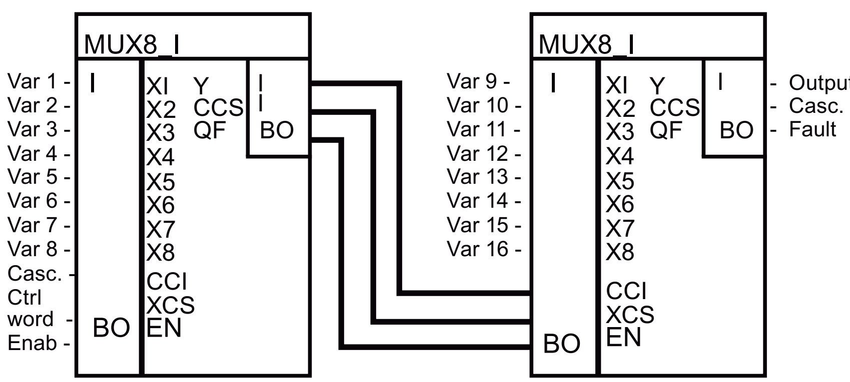
Block diagram



Truth table(s)

EN	XCS	Y	CSS	QF
0	Any	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	Default	Value range	Attrib-utes
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	Cascade input	0	INT	
XCS	Control word	0	0...32767	
EN	Enable	0	0/1	
Y	Output variable	0	INT	

Block connection	Description	Default	Value range	Attrib-utes
CCS	Cascading control word	0	0...32767	
QF	Error message	0	0/1	

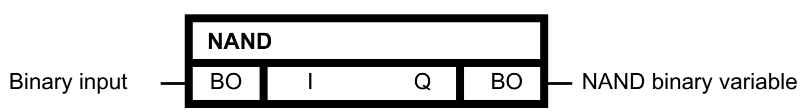
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

4.19 NAND

Logic AND operation (BOOL type)

Symbol



Brief description

NAND block with up to four inputs of the BOOL type

Method of operation

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.

$$Q = \overline{I_{01} \wedge \ldots \wedge I_{04}}$$

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

* Arbitrary

Block connections

Block connection	Description	Default	Value range	Attrib-utes
I	Binary input	1	0/1	
Q	NAND binary variable	0	0/1	

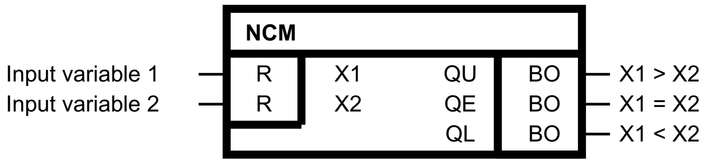
Configuration data

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

4.20 NCM

Numeric comparator (REAL type)

Symbol



Brief description

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

Method of operation

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 comparison 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	Default	Value range	Attrib-utes
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	

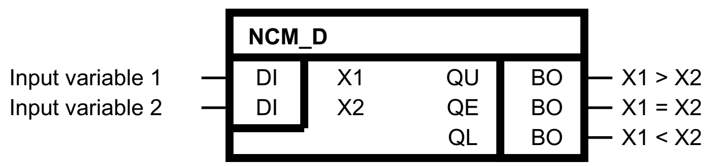
Configuration data

SIMOTION	✓
SINAMICS	✓
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.

4.21 NCM_D

Numeric comparator (DOUBLE INTEGER type)

Symbol



Brief description

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

Method of operation

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 comparison operation.

Truth table(s)

Comparison of input variables	Description	Default	Value range
	QU	QE	QL
X1 > X2	1	0	0
X1 = X2	0	1	0
X1 < X2	0	0	1

Block connections

Block connection	Description	Default	Value range	Attrib-utes
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	

Configuration data

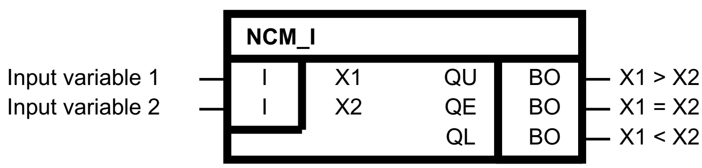
SIMOTION	✓
SINAMICS	✓

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.

4.22 NCM_I

Numeric comparator (INTEGER type)

Symbol



Brief description

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

Method of operation

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 comparison 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	Default	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	

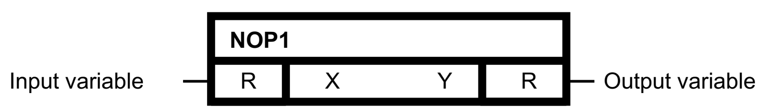
Configuration data

SIMOTION	✓
SINAMICS	✓
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.

4.23 NOP1

Dummy block (REAL type)

Symbol



Brief description

The block of the REAL type is used as dummy block (No Operation). It is used to provide a constant value for several blocks.

Method of operation

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

Block connections

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

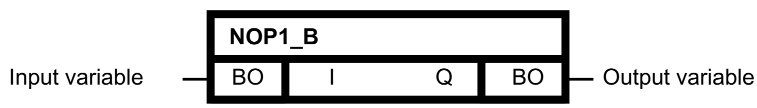
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

4.24 NOP1_B

Dummy block (BOOL type)

Symbol



Brief description

The block of the BOOL type is used as dummy block (No Operation). It is used to provide a constant value for several blocks.

Method of operation

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

Block connections

Block connection	Description	Default	Value range	Attrib-utes
I	Input variable	0	0/1	
Q	Output variable	0	0/1	

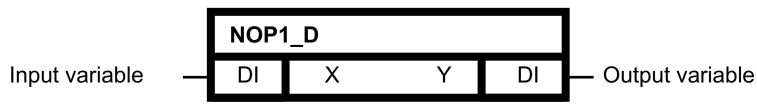
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

4.25 NOP1_D

Dummy block (DOUBLE INTEGER type)

Symbol



Brief description

The block of the DOUBLE INTEGER type is used as dummy block (No Operation). It is used to provide a constant value for several blocks.

Method of operation

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

Block connections

Block connection	Description	Default	Value range	Attrib-utes
X	Input variable	0	DINT	
Y	Output variable	0	DINT	

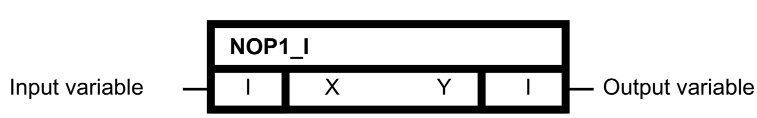
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

4.26 NOP1_I

Dummy block (INT type)

Symbol



Brief description

The block of the INT type is used as dummy block (No Operation). It is used to provide a constant value for several blocks.

Method of operation

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

Block connections

Block connection	Description	Default	Value range	Attrib-utes
X	Input variable	0	INT	
Y	Output variable	0	INT	

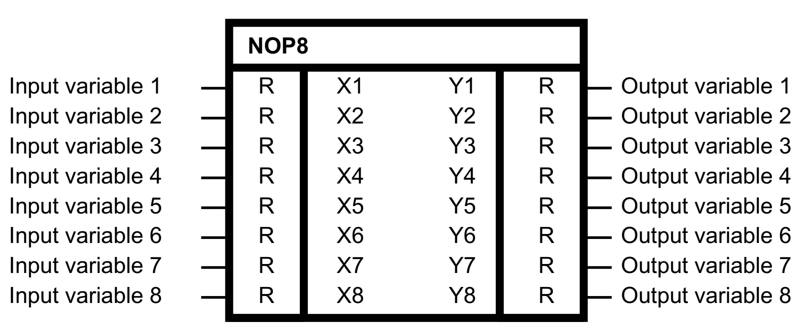
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

4.27 NOP8

Dummy block (REAL type)

Symbol



Brief description

The block of the REAL type is used as dummy block (No Operation). It is used to provide up to eight constant values for several blocks.

Method of operation

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.

Block connections

Block connection	Description	Default	Value range	Attrib-utes
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	

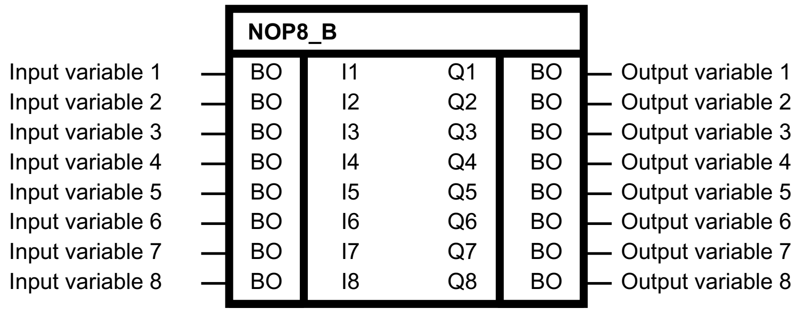
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

4.28 NOP8_B

Dummy block (BOOL type)

Symbol



Brief description

The block of the BOOL type is used as dummy block (No Operation). It is used to provide up to eight constant values for several blocks.

Method of operation

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.

Block connections

Block connection	Description	Default	Value range	Attrib-utes
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	

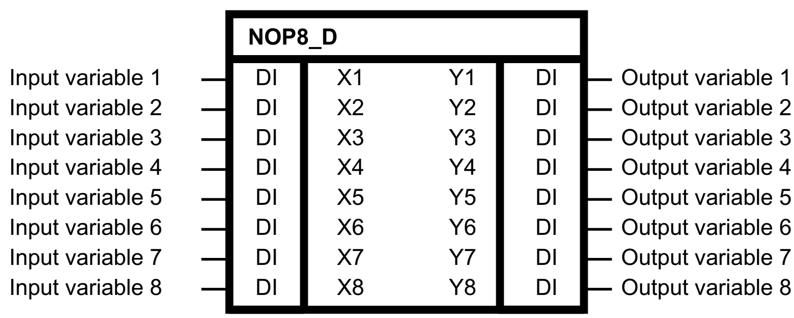
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

4.29 NOP8_D

Dummy block (DOUBLE INTEGER type)

Symbol



Brief description

The block of the DOUBLE INTEGER type is used as dummy block (No Operation). It is used to provide up to eight constant values for several blocks.

Method of operation

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.

Block connections

Block connection	Description	Default	Value range	Attrib-utes
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	

Block connection	Description	Default	Value range	Attrib-utes
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	

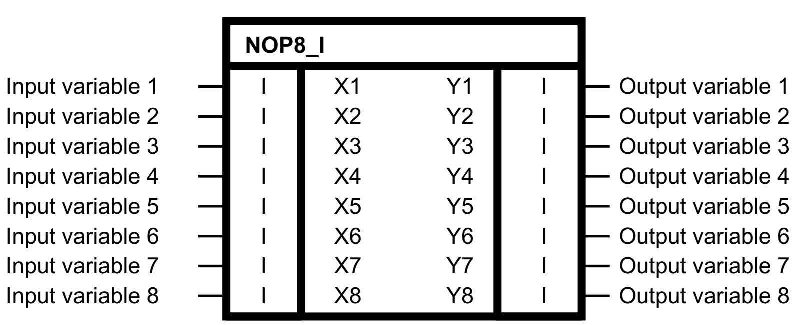
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

4.30 NOP8_I

Dummy block (INTEGER type)

Symbol



Brief description

The block of the INTEGER type is used as dummy block (No Operation). It is used to provide up to eight constant values for several blocks.

Method of operation

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.

Block connections

Block connection	Description	Default	Value range	Attrib-utes
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	

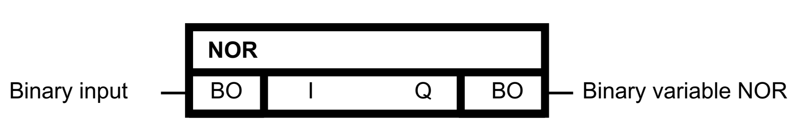
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

4.31 NOR

Logic OR operation (BOOL type)

Symbol



Brief description

NOR block with up to four inputs of the BOOL type

Method of operation

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

$$Q = \overline{I_{01} \vee \ldots \vee I_{04}}$$

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

* Arbitrary

Block connections

Block connection	Description	Default	Value range	Attrib-utes
I	Binary input	0	0/1	
Q	Binary variable NOR	1	0/1	

Configuration data

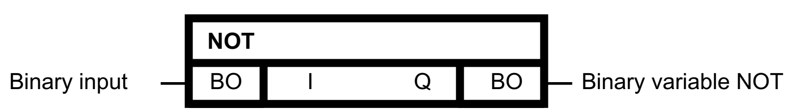
SIMOTION	✓
SINAMICS	✓

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

4.32 NOT

Inverter (BOOL type)

Symbol



Brief description

Inverter of the BOOL type

Method of operation

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	Default	Value range	Attributes
I	Binary input	0	0/1	
Q	Binary variable NOT	1	0/1	

Configuration data

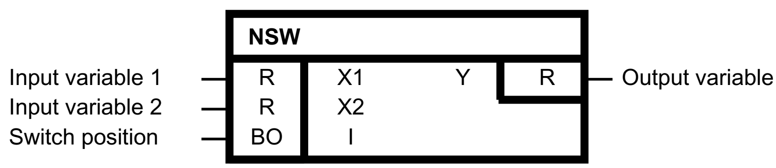
SIMOTION	✓
SINAMICS	✓

Can be loaded online	Yes
Special characteristics	-

4.33 NSW

Numeric change-over switch (REAL type)

Symbol



Brief description

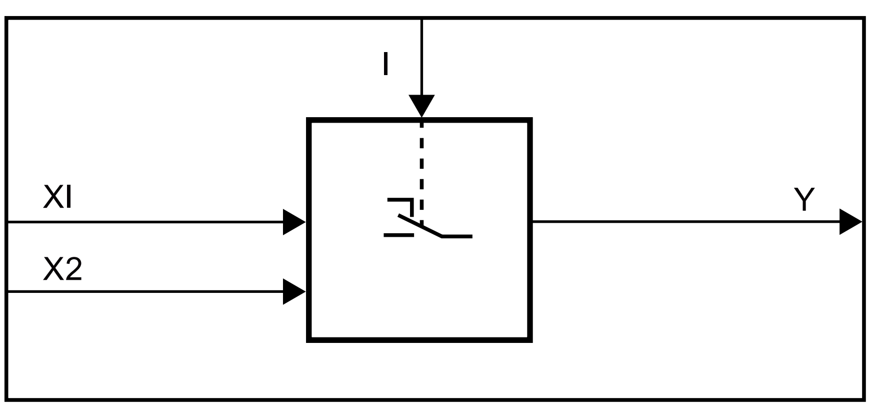
The block switches one of two numeric input variables (REAL type) to the output.

Method of operation

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	Default	Value range	Attrib-utes
X1	Input variable 1	0	REAL	
X2	Input variable 2	0	REAL	
I	Switch position	0	0/1	
Y	Output variable	0	REAL	

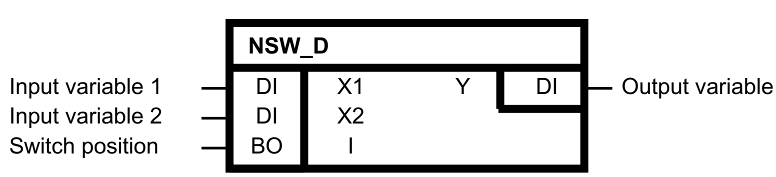
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

4.34 NSW_D

Numeric change-over switch (DOUBLE INTEGER type)

Symbol



Brief description

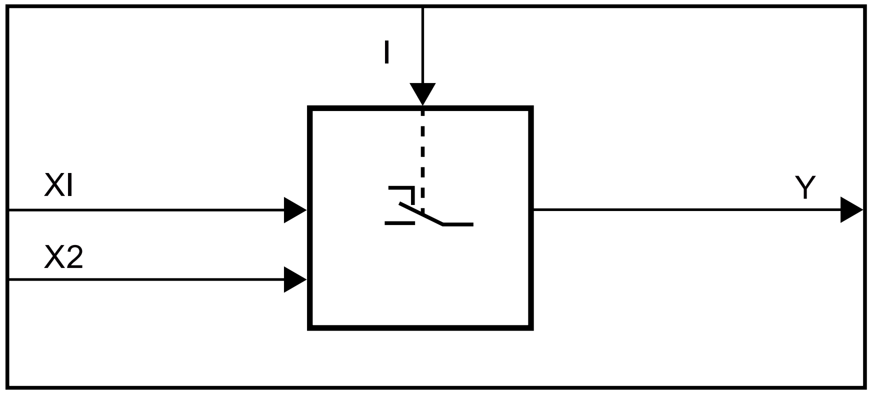
The block switches one of two numeric input variables (DOUBLE INTEGER type) to the output

Method of operation

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	Default	Value range	Attrib-utes
X1	Input variable 1	0	DINT	
X2	Input variable 2	0	DINT	
I	Switch position	0	0/1	
Y	Output variable	0	DINT	

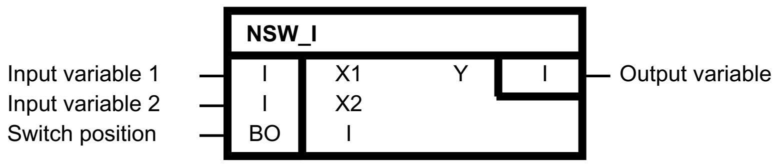
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

4.35 NSW_I

Numeric change-over switch (INTEGER type)

Symbol



Brief description

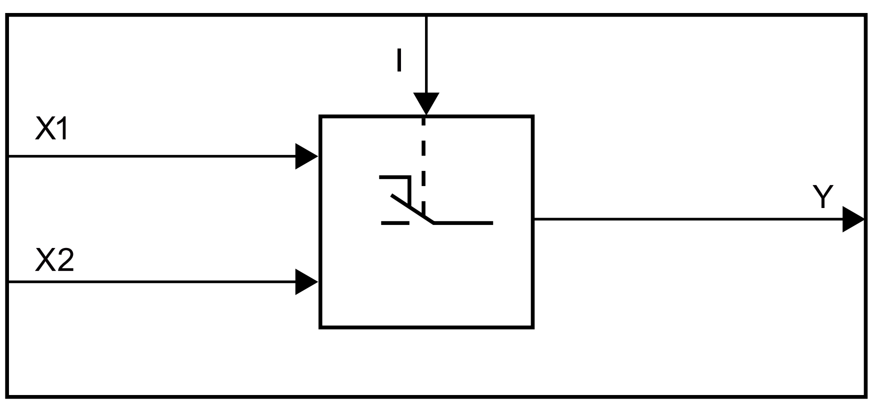
The block switches one of two numeric input variables (INTEGER type) to the output.

Method of operation

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	Default	Value range	Attrib-utes
X1	Input variable 1	0	INT	
X2	Input variable 2	0	INT	
I	Switch position	0	0/1	
Y	Output variable	0	INT	

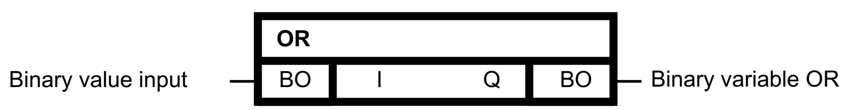
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

4.36 OR

Logic OR operation (BOOL type)

Symbol



Brief description

OR block with up to four inputs of the BOOL type

Method of operation

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.

$Q = I_{01} \vee \ldots \vee I_{04}$

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

* Arbitrary

Block connections

Block connection	Description	Default	Value range	Attrib-utes
I	Binary value input	0	0/1	
Q	Binary variable OR	0	0/1	

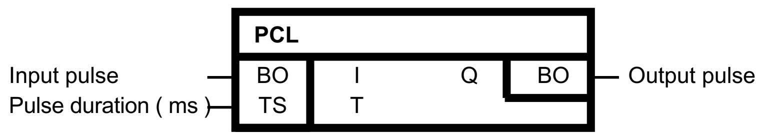
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	I comprises up to four connections (I1 to I4)

4.37 PCL

Pulse shortener (BOOL type)

Symbol



Brief description

Timer for limiting the pulse duration

Method of operation

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

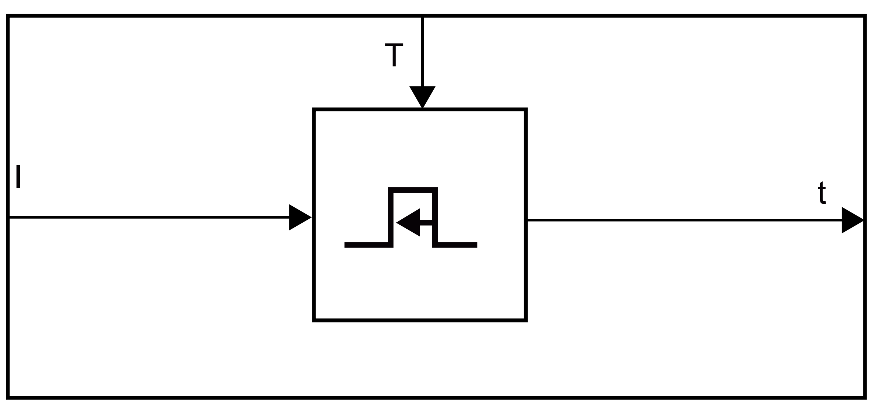
Initialization

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

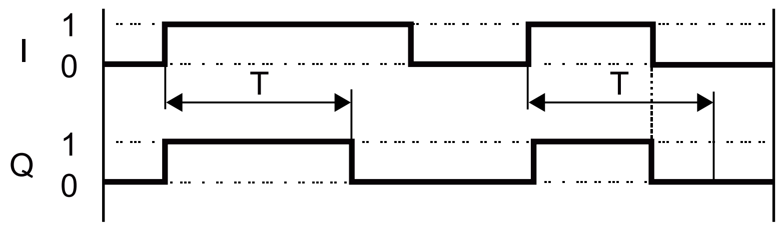
If input I receives the value 1 during initialization from the upstream block output, the block cannot detect a positive edge during the first cyclic pass.

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	Default	Value range	Attrib-utes
I	Input pulse	0	0/1	
T	Pulse duration (ms)	0	SDTIME	
Q	Output pulse	0	0/1	

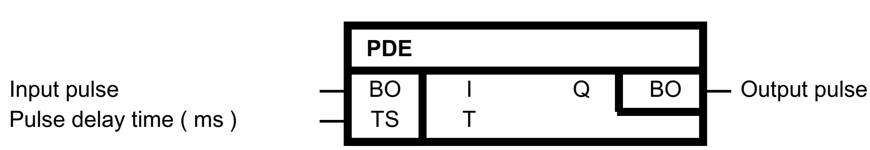
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

4.38 PDE

Switch-on delay (BOOL type)

Symbol



Brief description

BOOL-type timer with on-delay

Method of operation

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 = 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 long that the maximum value that can be displayed internally (T/ta as 32-bit value, where ta = sampling time) is exceeded, the maximum value is set (e.g. when ta = 1 ms, approx. 50 days).

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

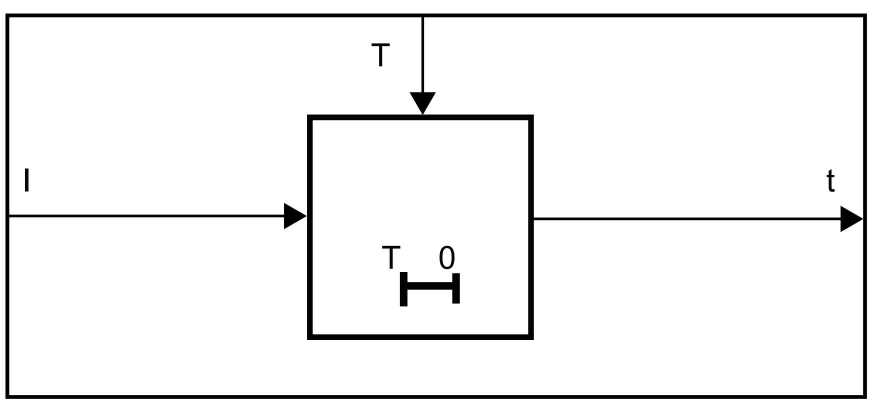
Method of operation

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

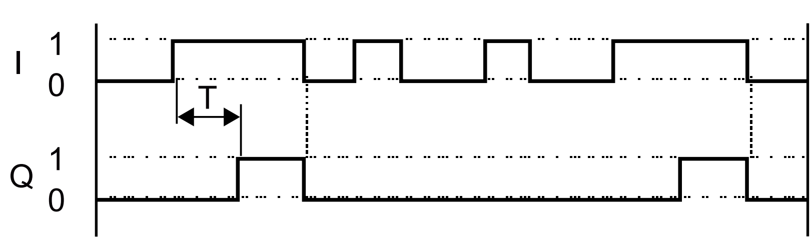
If input I receives the value 1 during initialization from the upstream block output, the block cannot detect a positive edge during the first cyclic pass. The pulse delay time T is therefore not taken over in the first cyclic pass with I = 1, the specified time from the initialization remains effective.

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	Default	Value range	Attrib-utes
I	Input pulse	0	0/1	
T	Pulse delay time (ms)	0	SDTIME	
Q	Output pulse	0	0/1	

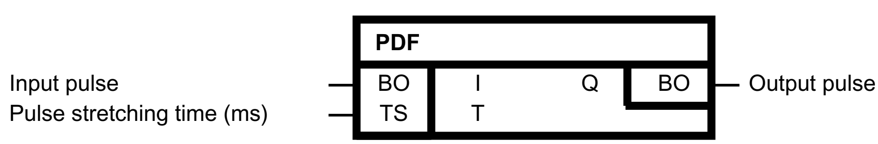
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

4.39 PDF

Switch-off delay (BOOL type)

Symbol



Brief description

BOOL-type timer with off-delay

Method of operation

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 = 1.

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.

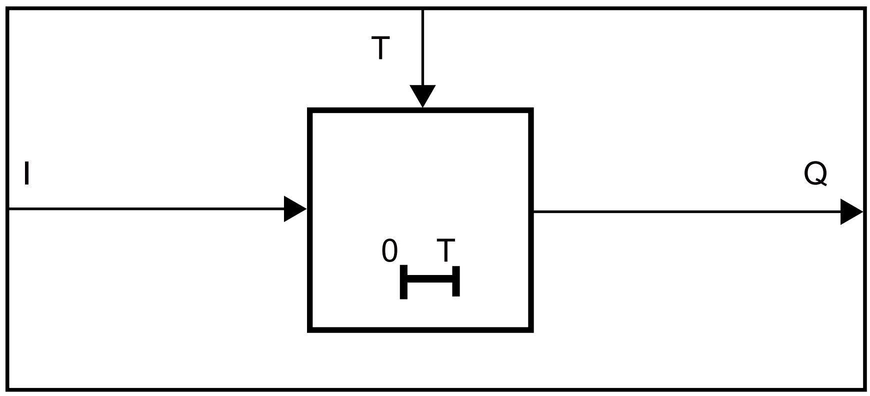
Initialization

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

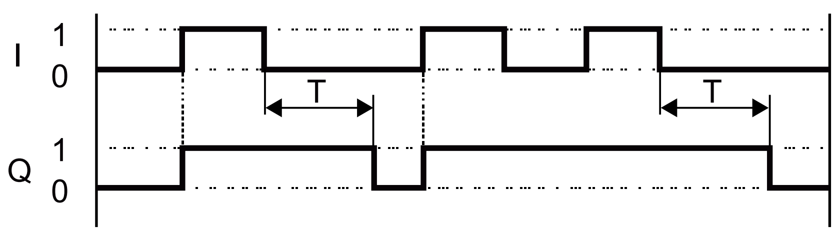
If input I receives the value 1 during initialization from the upstream block output, the block cannot detect a negative edge during the first cyclic pass.

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	Default	Value range	Attrib-utes
I	Input pulse	0	0/1	
T	Pulse stretching time (ms)	0	SDTIME	
Q	Output pulse	0	0/1	

Configuration data

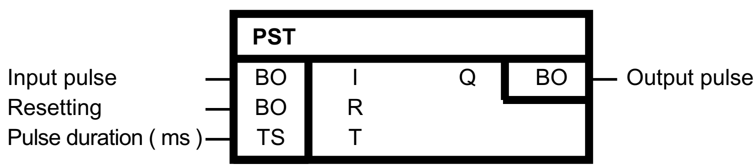
SIMOTION	✓
SINAMICS	✓

Can be loaded online	Yes
Special characteristics	-

4.40 PST

Pulse stretching block (BOOL type)

Symbol



Brief description

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

Method of operation

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.

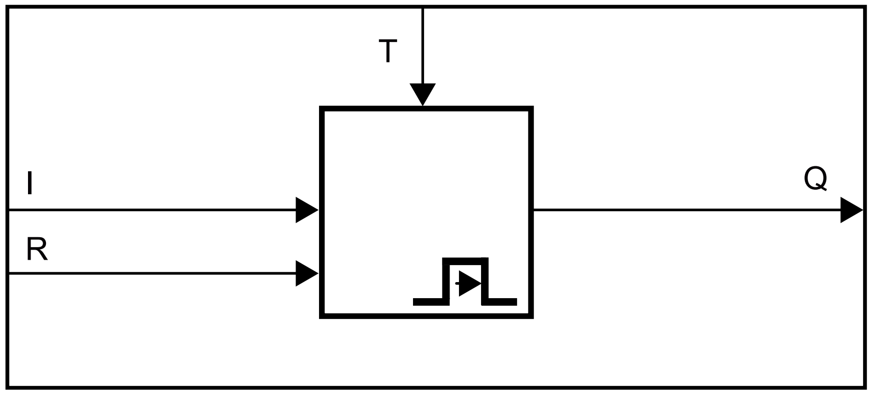
Initialization

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

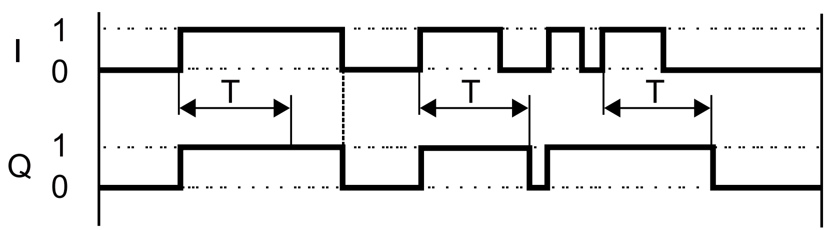


Figure 4-1 Output pulse Q as a function of pulse duration T and input pulse I (with R = 0)

Block connections

Block connection	Description	Default	Value range	Attrib-utes
I	Input pulse	0	0/1	
R	Resetting	0	0/1	
T	Pulse duration (ms)	0	SDTIME	
Q	Output pulse	0	0/1	

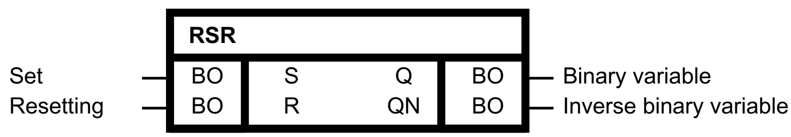
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

4.41 RSR

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

Symbol



Brief description

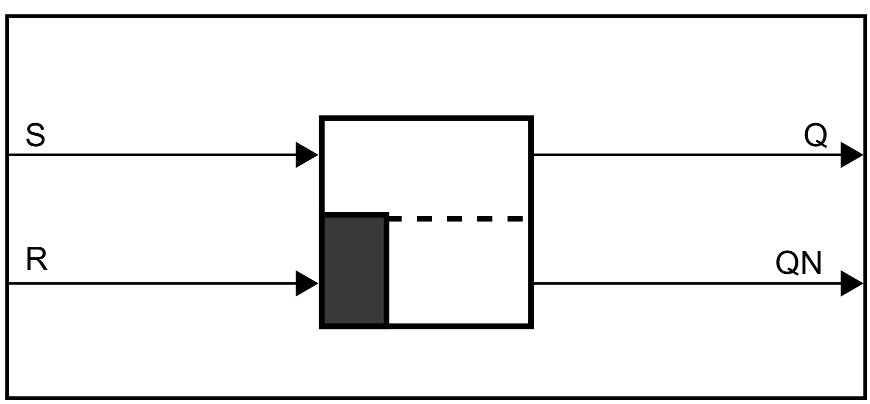
Used as static binary value memory

Method of operation

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	Default	Value range	Attrib-utes
S	Set	0	0/1	
R	Resetting	0	0/1	
Q	Binary variable	0	0/1	
QN	Inverse binary variable	1	0/1	

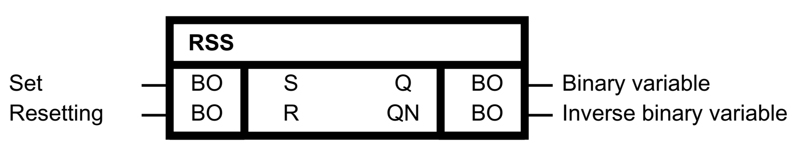
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

4.42 RSS

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

Symbol



Brief description

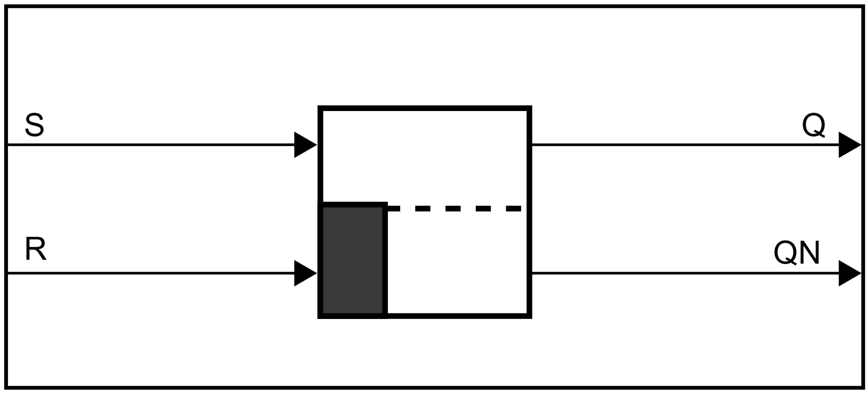
Block of the BOOL type for use as a static binary value memory

Method of operation

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 = 0

Block connections

Block connection	Description	Default	Value range	Attrib-utes
S	Set	0	0/1	
R	Resetting	0	0/1	
Q	Binary variable	0	0/1	
QN	Inverse binary variable	1	0/1	

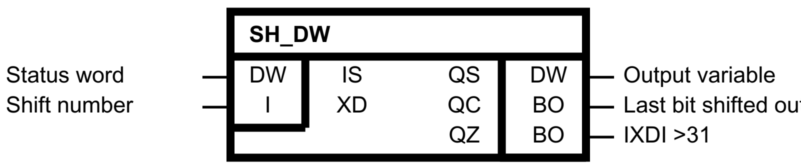
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

4.43 SH_DW

Shift block (DWORD type)

Symbol



Brief description

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

Method of operation

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

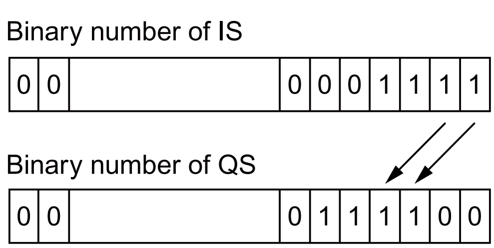
New positions created by the shifting are filled with zeros irrespective of the shift direction.

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

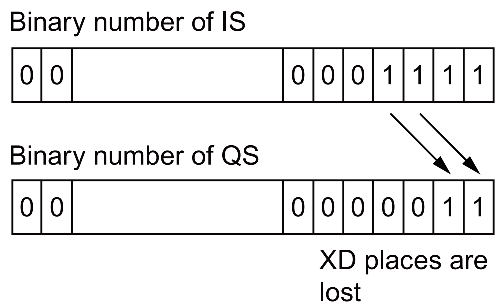
-> QS = 60; QC = 0



Shift to the right - example:

XD = -2; IS = 15

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



Block connections

Block connection	Description	Default	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	

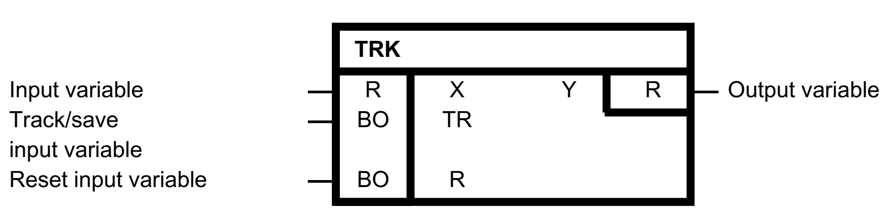
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

4.44 TRK

Tracking/memory element (REAL type)

Symbol



Brief 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

Method of operation

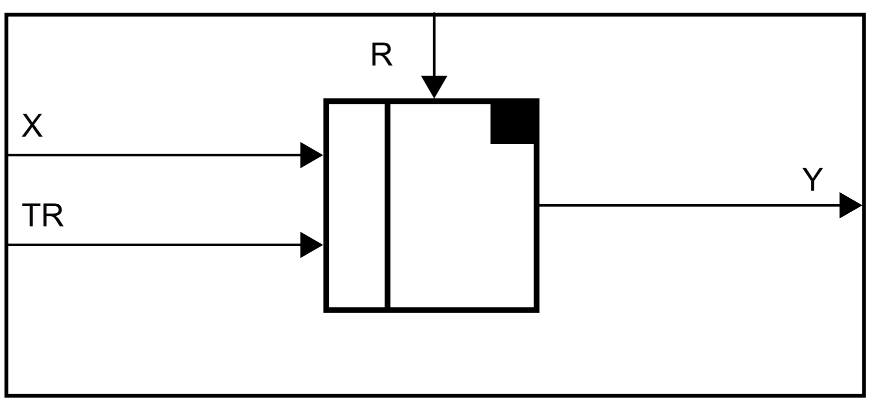
TRACK	TR = 1	Direct tracking of output value Y = X.
	TR = 1-> 0	With a negative edge at TR, the current input variable is saved and output on output Y.
	TR = 0	The value at output Y does not change.
RESET	R = 1	Output Y is reset to 0. The reset input is dominant.

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	Default	Value range	Attrib-utes
X	Input variable	0.0	REAL	
TR	Track/save input variable	0	0/1	
R	Reset input variable	0	0/1	
Y	Output variable	0.0	REAL	

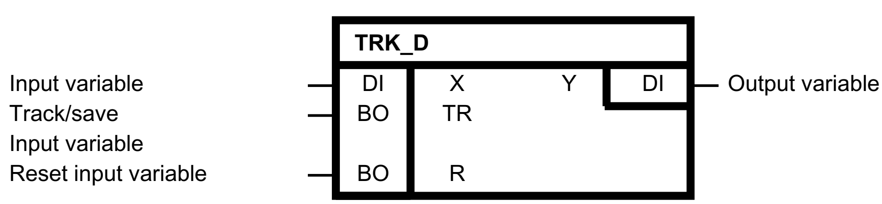
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

4.45 TRK_D

Tracking/memory element (DOUBLE INTEGER type)

Symbol



Brief 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

Method of operation

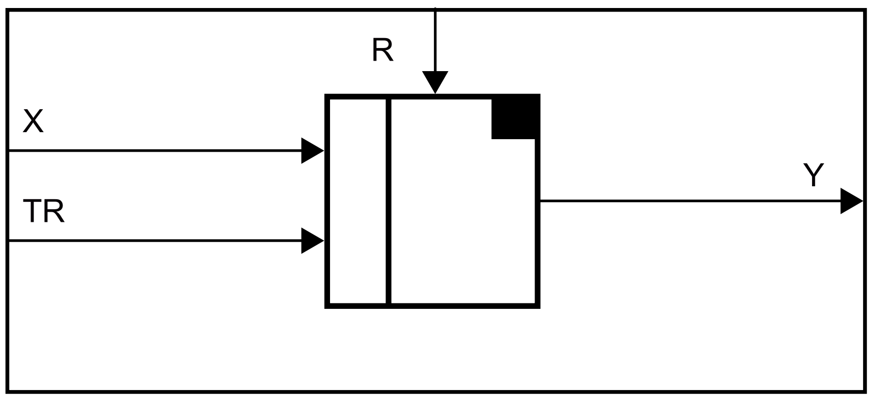
TRACK	TR = 1	Direct tracking of output value Y = X.
	TR = 1-> 0	With a negative edge at TR, the current input variable is saved and output on output Y.
	TR = 0	The value at output Y does not change.
RESET	R = 1	Output Y is reset to 0. The reset input is dominant.

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	Default	Value range	Attributes
X	Input variable	0	DINT	
TR	Track/save input variable	0	0/1	
R	Reset input variable	0	0/1	
Y	Output variable	0	DINT	

Configuration data

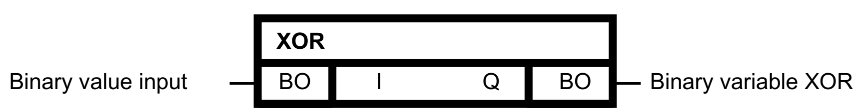
SIMOTION	✓
SINAMICS	✓

Can be loaded online	Yes
Special characteristics	-

4.46 XOR

Logic exclusive OR operation (BOOL type)

Symbol



Brief description

XOR block with up to four inputs of the BOOL type

Method of operation

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	Default	Value range	Attributes
I	Binary value input	0	0/1	
Q	Binary variable XOR	0	0/1	

Configuration data

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

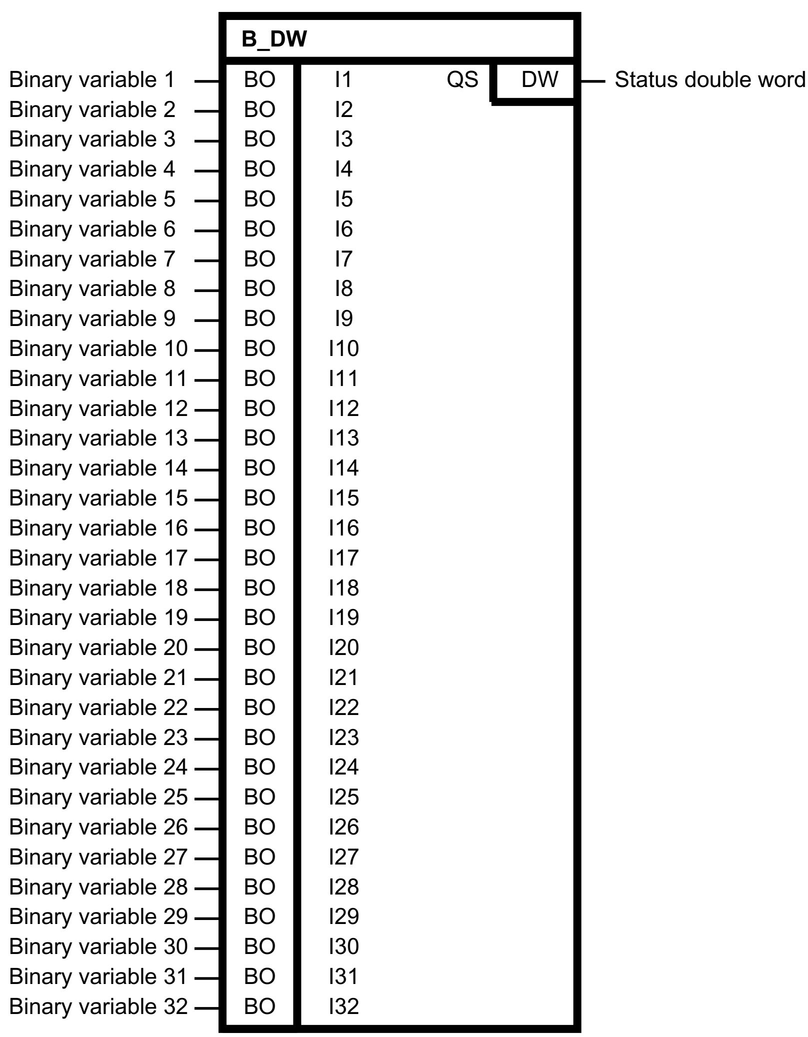
Conversion

5

5.1 B_DW

32 binary variables to status double word converter

Symbol



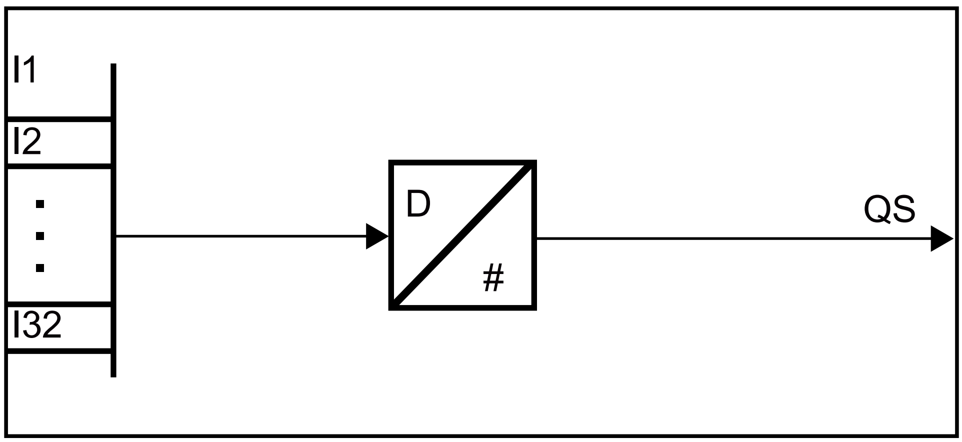
Brief description

Status double word generation from 32 binary variables

Method of operation

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	Default	Value range	Attrib-utes
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	

Block connection	Description	Default	Value range	Attrib-utes
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	

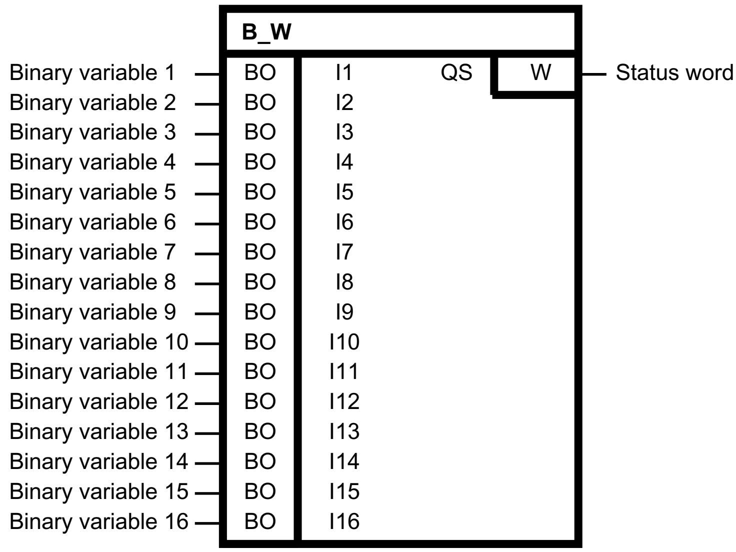
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special character-istics	-

5.2 B_W

16 binary variables to status word converter

Symbol



Brief description

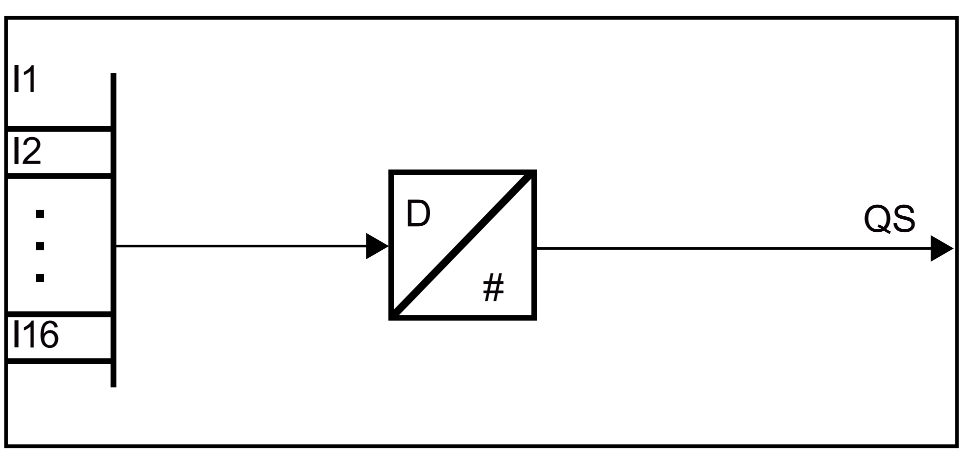
Status word generation from 16 binary variables

Method of operation

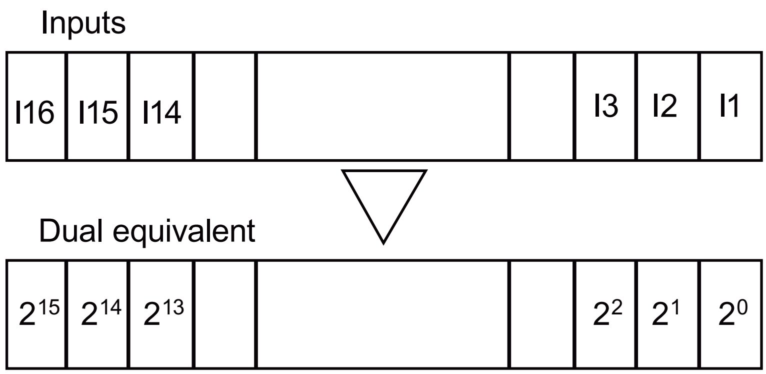
This block combines the binary variables from I1 to I16 into a status word and gives the result to its output QS.

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	Default	Value range	Attrib-utes
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	

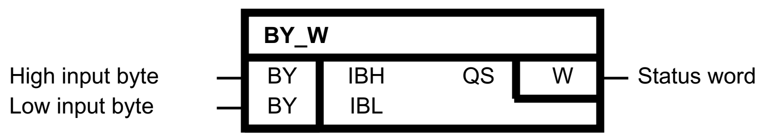
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special character-istics	-

5.3 BY_W

Status byte to status word converter

Symbol



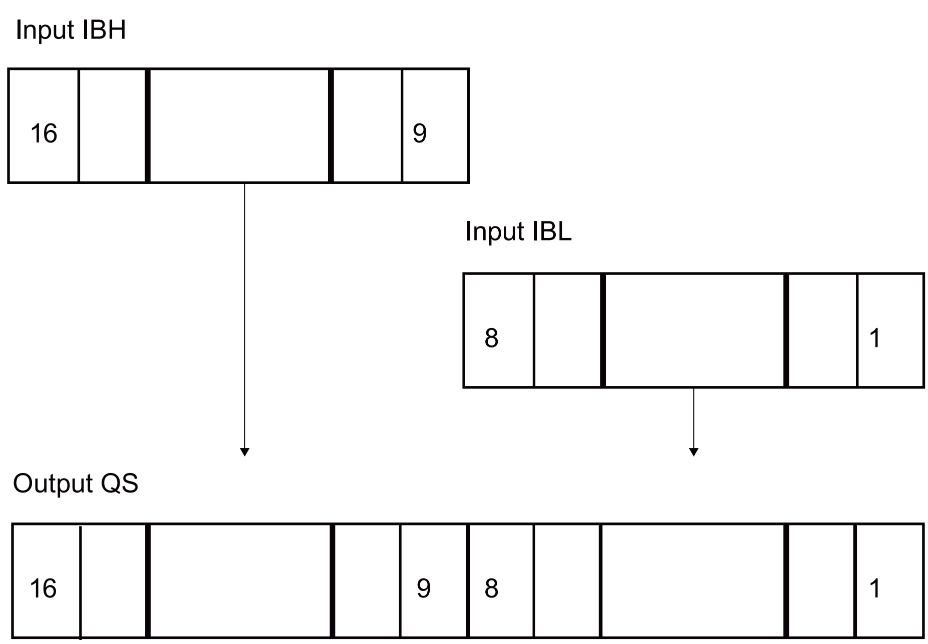
Brief description

Combining two bytes into one word

Method of operation

The block combines two bytes into one 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	Default	Value range	Attrib-utes
IBH	High input byte	16#00	BYTE	
IBL	Low input byte	16#00	BYTE	
QS	Status word	16#0000	WORD	

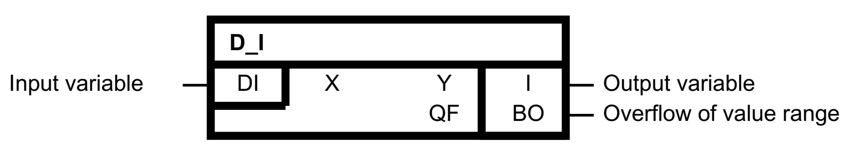
Configuration data

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

5.4 D_I

DOUBLE INTEGER to INTEGER converter

Symbol



Brief description

Conversion of a DOUBLE INTEGER variable to an INTEGER variable

Method of operation

This block converts a DOUBLE INTEGER variable to an INTEGER variable, i.e. the least significant 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	Default	Value range	Attrib-utes
X	Input variable	0	DINT	
Y	Output variable	0	INT	
QF	Overflow of value range	0	0/1	

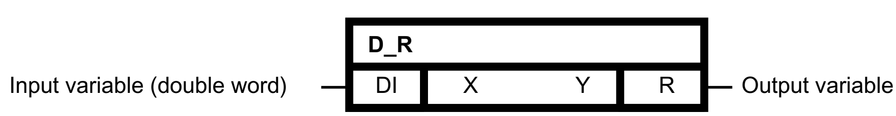
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special character-istics	-

5.5 D_R

DOUBLE INTEGER to REAL converter

Symbol



Brief description

Conversion of a DOUBLE INTEGER variable to a REAL variable

Method of operation

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

Block connections

Block connection	Description	Default	Value range	Attrib-utes
X	Input variable (double word)	0	DINT	
Y	Output variable	0.0	REAL	

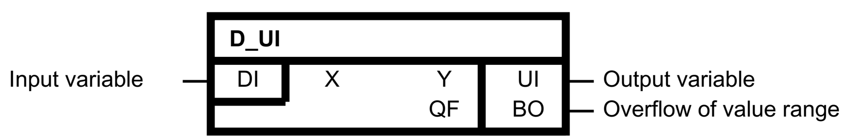
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special characteristics	-

5.6 D_UI

DOUBLE INTEGER to UNSIGNED INTEGER converter

Symbol



Brief description

Conversion of a DOUBLE INTEGER variable to an UNSIGNED INTEGER variable

Method of operation

This block converts a DOUBLE INTEGER variable to an UNSIGNED INTEGER variable, i.e. the least significant 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	Default	Value range	Attributes
X	Input variable	0	DINT	
Y	Output variable	0	UINT	
QF	Overflow of value range	0	0/1	

Configuration data

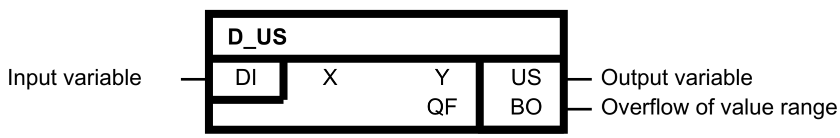
SIMOTION	✓
SINAMICS	✓

Can be inserted online	Yes
Special characteristics	-

5.7 D_US

DOUBLE INTEGER to UNSIGNED SHORT INTEGER converter

Symbol



Brief description

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

Method of operation

This block converts a DOUBLE INTEGER variable to an UNSIGNED SHORT INTEGER variable, i.e. the least significant 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	Default	Value range	Attributes
X	Input variable	0	DINT	
Y	Output variable	0	USINT	
QF	Overflow of value range	0	0/1	

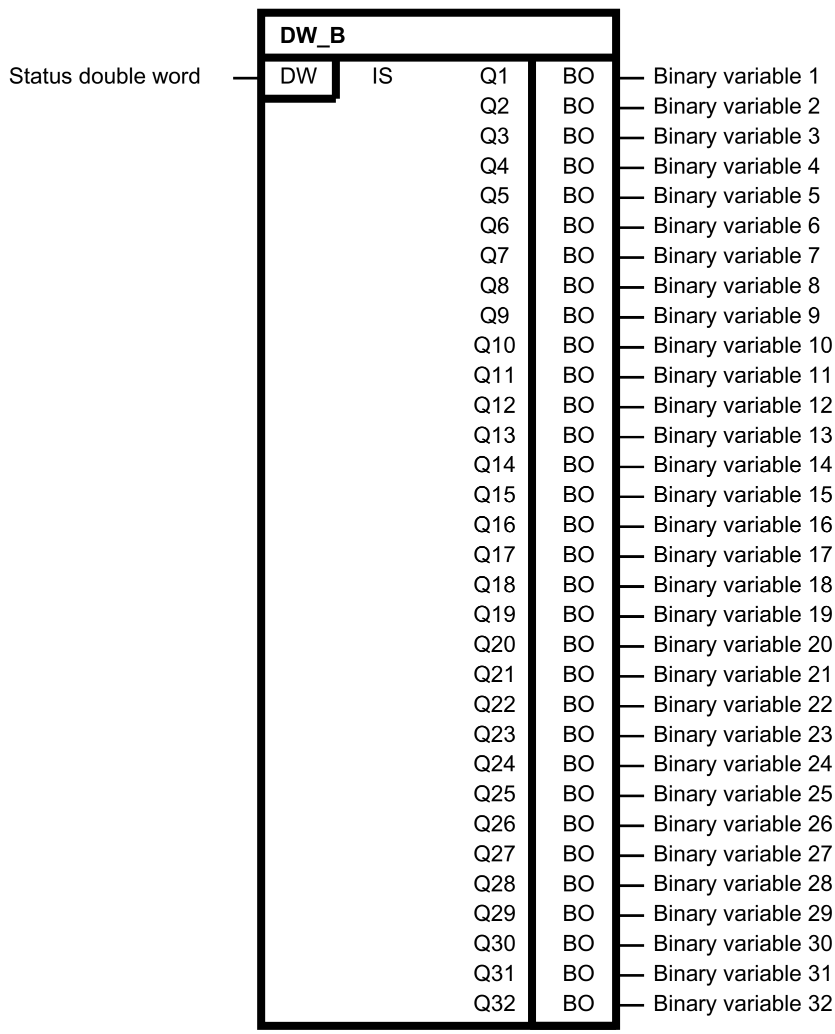
Configuration data

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

5.8 DW_B

Status double word to 32 binary variables converter

Symbol



Brief description

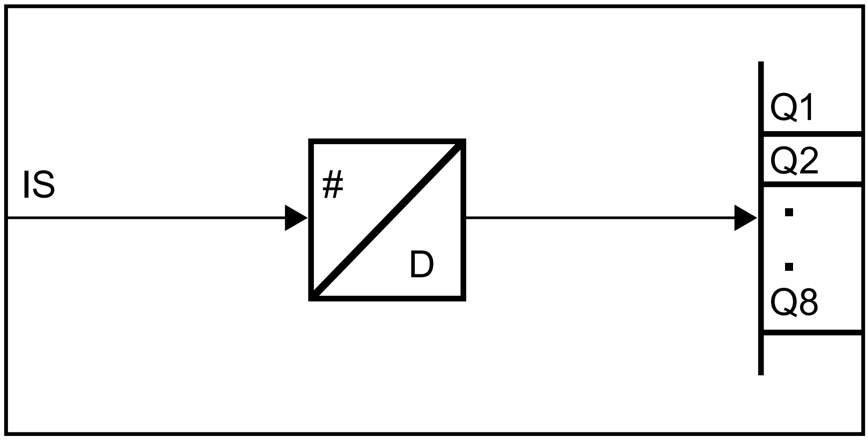
Status double word decryption to 32 binary variables

Method of operation

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	Default	Value range	Attrib-utes
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	

Block connection	Description	Default	Value range	Attrib-utes
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	

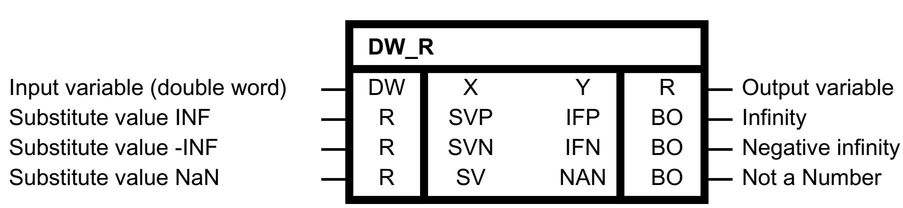
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special character-istics	-

5.9 DW_R

Acceptance of bit string as REAL value

Symbol



Brief description

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

Method of operation

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 corresponds to the representation for +/-infinite or NaN according to IEEE 754, the corresponding binary outputs IFP, IFN, and NAN are set to 1, and the substitute value predefined for each are applied at output Y.

Block connections

Block connection	Description	Default	Value range	Attrib-utes
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	Negative infinity	0	0/1	
NAN	Not a Number	0	0/1	
Y	Output variable	0.0	REAL	

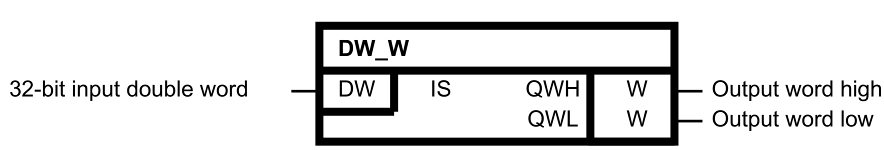
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special character-istics	-

5.10 DW_W

Status double word to status word converter

Symbol



Brief description

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

Method of operation

Output variables are calculated according to the following regulation:

QWL = IS mod 2^{16}

QWH = IS / 2^{16}

Block connections

Block connection	Description	Default	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	

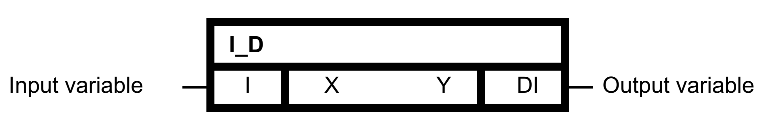
Configuration data

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

5.11 I_D

INTEGER to DOUBLE INTEGER converter

Symbol



Brief description

Conversion of an INTEGER variable to a DOUBLE INTEGER variable

Method of operation

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

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

Block connections

Block connection	Description	Default	Value range	Attrib-utes
X	Input variable	0	INT	
Y	Output variable	0	DINT	

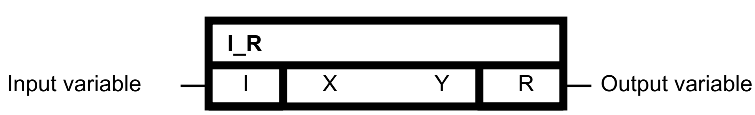
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special character-istics	-

5.12 I_R

INTEGER to REAL converter

Symbol



Brief description

Conversion of an INTEGER variable to a REAL variable

Method of operation

This block converts an INTEGER variable to a REAL variable.

Block connections

Block connection	Description	Default	Value range	Attrib- utes
X	Input variable	0	INT	
Y	Output variable	0.0	REAL	

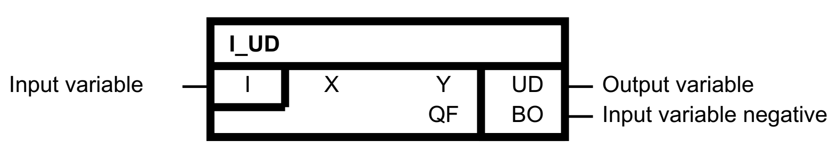
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on- line	Yes
Special character- istics	-

5.13 I_UD

INTEGER to UNSIGNED DOUBLE INTEGER converter

Symbol



Brief description

Conversion of an INTEGER variable to an UNSIGNED DOUBLE INTEGER variable

Method of operation

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

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

The most significant 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	Default	Value range	Attrib-utes
X	Input variable	0	INT	
Y	Output variable	0	UDINT	
QF	Input variable negative	0	0/1	

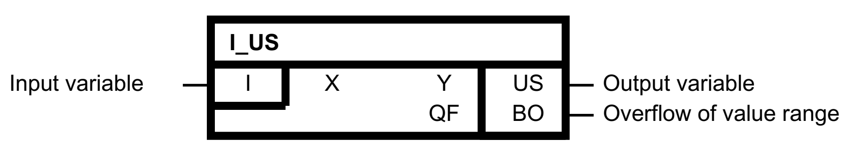
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special character-istics	-

5.14 I_US

INTEGER to UNSIGNED SHORT INTEGER converter

Symbol



Brief description

Conversion of an INTEGER variable to an UNSIGNED SHORT INTEGER variable

Method of operation

This block converts an INTEGER variable to an UNSIGNED SHORT INTEGER variable, i.e. the least significant 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	Default	Value range	Attrib-utes
X	Input variable	0	INT	
Y	Output variable	0	USINT	
QF	Overflow of value range	0	0/1	

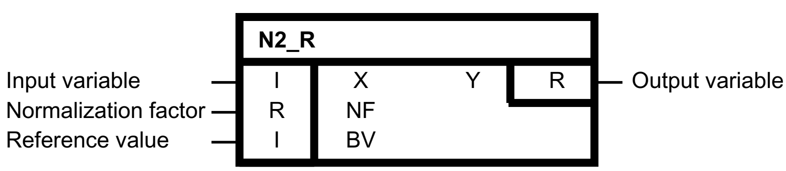
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special character-istics	-

5.15 N2_R

16-bit fixed-point format (N2) to REAL converter

Symbol



Brief description

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

Method of operation

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

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

Block connections

Block connection	Description	Default	Value range	Attrib-utes
X	Input variable	0	INT	
NF	Normalization factor	1.0	REAL	
BV	Reference value	16384	INT	
Y	Output variable	0.0	REAL	

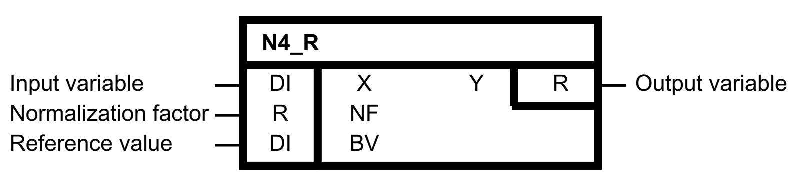
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special character-istics	-

5.16 N4_R

32-bit fixed-point format (N4) to REAL converter

Symbol



Brief 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.

Method of operation

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

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

Block connections

Block connection	Description	Default	Value range	Attrib-utes
X	Input variable	0	DINT	
NF	Normalization factor	1.0	REAL	
BV	Reference value	1073741824	DINT	
Y	Output variable	0.0	REAL	

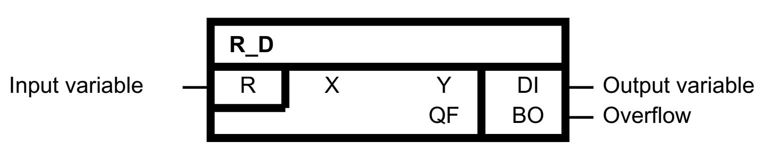
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special character-istics	-

5.17 R_D

REAL to DOUBLE INTEGER converter

Symbol



Brief description

Conversion of a REAL variable to a DOUBLE INTEGER variable

Method of operation

This block converts a REAL variable to a DOUBLE INTEGER variable. During the conversion, decimal places of the input variable are truncated.

Note

The number is not rounded off.

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	Default	Value range	Attrib-utes
X	Input variable	0.0	REAL	
Y	Output variable	0	DINT	
QF	Overflow	0	0/1	

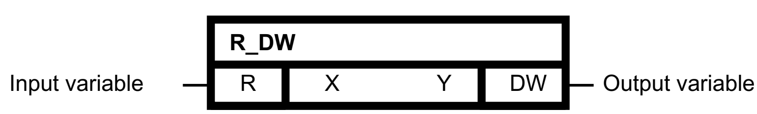
Configuration data

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

5.18 R_DW

Acceptance of bit string as DWORD

Symbol



Brief description

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

Method of operation

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

Block connections

Block connection	Description	Default	Value range	Attrib-utes
X	Input variable	0.0	REAL	
Y	Output variable	16#00000000	DWORD	

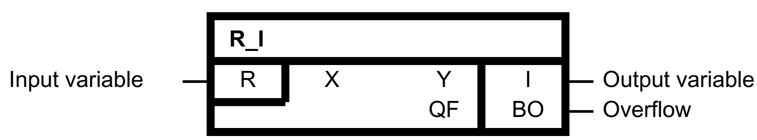
Configuration data

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

5.19 R_I

REAL to INTEGER converter

Symbol



Brief description

Conversion of a REAL variable to an INTEGER variable

Method of operation

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	Default	Value range	Attributes
X	Input variable	0.0	REAL	
Y	Output variable	0	INT	
QF	Overflow	0	0/1	

Configuration data

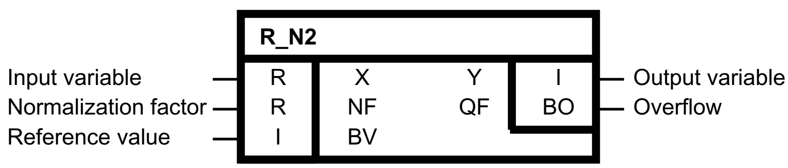
SIMOTION	✓
SINAMICS	✓

Can be inserted on-line	Yes
Special characteristics	-

5.20 R_N2

REAL to 16-bit fixed-point format (N2) converter

Symbol



Brief 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).

Method of operation

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	Default	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	

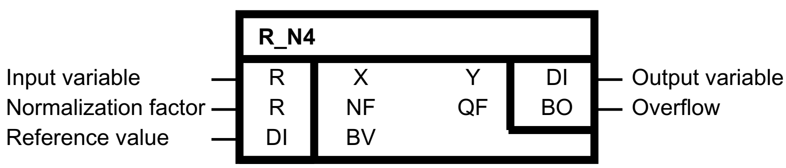
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special character-istics	-

5.21 R_N4

REAL to 32-bit fixed-point format (N4) converter

Symbol



Brief 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%).

Method of operation

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	Default	Value range	Attrib-utes
X	Input variable	0.0	REAL	
NF	Normalization factor	1.0	REAL	
BV	Reference value	1073741824	DINT	

Block connection	Description	Default	Value range	Attrib-utes
Y	Output variable	0	DINT	
QF	Overflow	0	0/1	

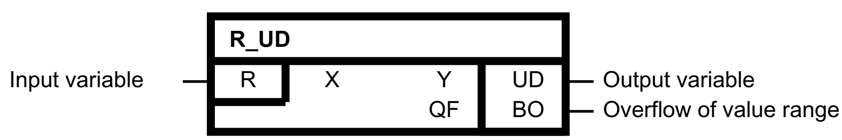
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special character-istics	-

5.22 R_UD

REAL to UNSIGNED DOUBLE INTEGER converter

Symbol



Brief description

Conversion of a REAL variable to an UNSIGNED DOUBLE INTEGER variable

Method of operation

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. The result is limited to the data type of the output variable corresponding to 0 or 2^{32}-1. If the output variable has been limited, then QF = 1 is set.

Block connections

Block connection	Description	Default	Value range	Attrib-utes
X	Input variable	0.0	REAL	
Y	Output variable	0	UDINT	
QF	Overflow of value range	0	0/1	

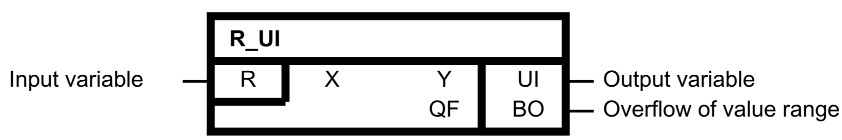
Configuration data

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

5.23 R_UI

REAL to UNSIGNED INTEGER converter

Symbol



Brief description

Conversion of a REAL variable to an UNSIGNED INTEGER variable

Method of operation

This block converts a REAL variable to an UNSIGNED 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 0 or 2^{16}-1. If the output variable has been limited, then QF = 1 is set.

Block connections

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

Configuration data

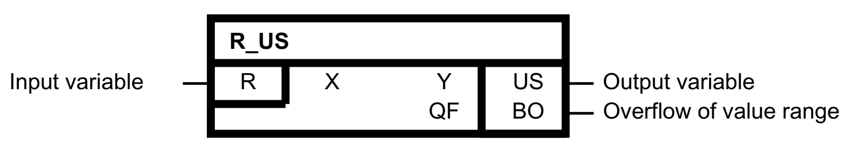
SIMOTION	✓
SINAMICS	✓

Can be inserted online	Yes
Special characteristics	-

5.24 R_US

REAL to UNSIGNED SHORT INTEGER converter

Symbol



Brief description

Conversion of a REAL variable to an UNSIGNED SHORT INTEGER variable

Method of operation

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. The result is limited to the data type of the output variable corresponding to 0 or 2^8 -1. If the output variable has been limited, then QF = 1 is set.

Block connections

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

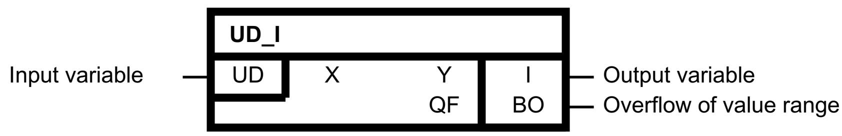
Configuration data

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

5.25 UD_I

UNSIGNED DOUBLE INTEGER to INTEGER converter

Symbol



Brief description

Conversion of an UNSIGNED DOUBLE INTEGER variable to an INTEGER variable

Method of operation

This block converts an UNSIGNED DOUBLE INTEGER variable to an INTEGER variable, i.e. the least significant word of the UNSIGNED 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	Default	Value range	Attrib-utes
X	Input variable	0	UDINT	
Y	Output variable	0	INT	
QF	Overflow of value range	0	0/1	

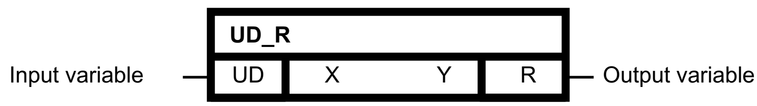
Configuration data

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

5.26 UD_R

UNSIGNED DOUBLE INTEGER to REAL converter

Symbol



Brief description

Conversion of an UNSIGNED DOUBLE INTEGER variable to a REAL variable

Method of operation

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

Block connections

Block connection	Description	Default	Value range	Attrib- utes
X	Input variable	0	UDINT	
Y	Output variable	0.0	REAL	

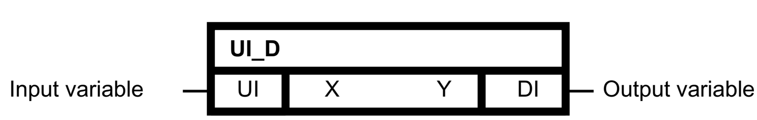
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special character-istics	-

5.27 UI_D

UNSIGNED INTEGER to DOUBLE INTEGER converter

Symbol



Brief description

Conversion of an UNSIGNED INTEGER variable to a DOUBLE INTEGER variable

Method of operation

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

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

Block connections

Block connection	Description	Default	Value range	Attrib-utes
X	Input variable	0	UINT	
Y	Output variable	0	DINT	

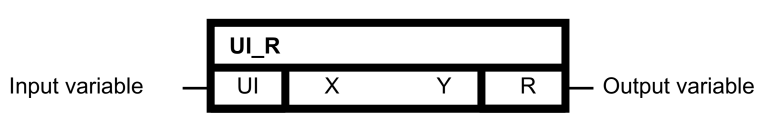
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special character-istics	-

5.28 UI_R

UNSIGNED INTEGER to REAL converter

Symbol



Brief description

Conversion of an UNSIGNED INTEGER variable to a REAL variable

Method of operation

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

Block connections

Block connection	Description	Default	Value range	Attrib-utes
X	Input variable	0	UINT	
Y	Output variable	0	DINT	

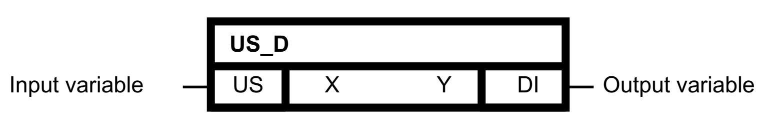
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special character-istics	-

5.29 US_D

UNSIGNED SHORT INTEGER to DOUBLE INTEGER converter

Symbol



Brief description

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

Method of operation

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

The input variable of data type UNSIGNED SHORT INTEGER is copied to the least significant byte of output variable Y. The remaining most significant bytes are filled with 16#00.

Block connections

Block connection	Description	Default	Value range	Attrib-utes
X	Input variable	0	USINT	
Y	Output variable	0	DINT	

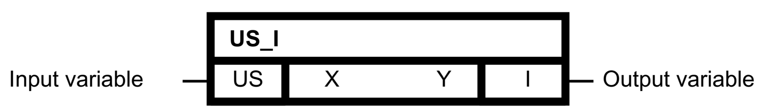
Configuration data

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

5.30 US_I

UNSIGNED SHORT INTEGER to INTEGER converter

Symbol



Brief description

Conversion of an UNSIGNED SHORT INTEGER variable to an INTEGER variable

Method of operation

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

The input variable of data type UNSIGNED SHORT INTEGER is copied to the least significant byte of output variable Y. The remaining most significant bytes are filled with 16#00.

Block connections

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

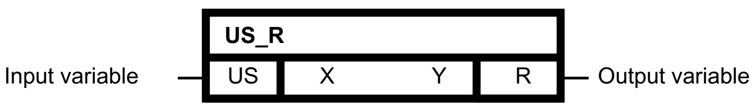
Configuration data

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

5.31 US_R

UNSIGNED SHORT INTEGER to REAL converter

Symbol



Brief description

Conversion of an UNSIGNED SHORT INTEGER variable to a REAL variable

Method of operation

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

Block connections

Block connection	Description	Default	Value range	Attrib-utes
X	Input variable	0	USINT	
Y	Output variable	0.0	REAL	

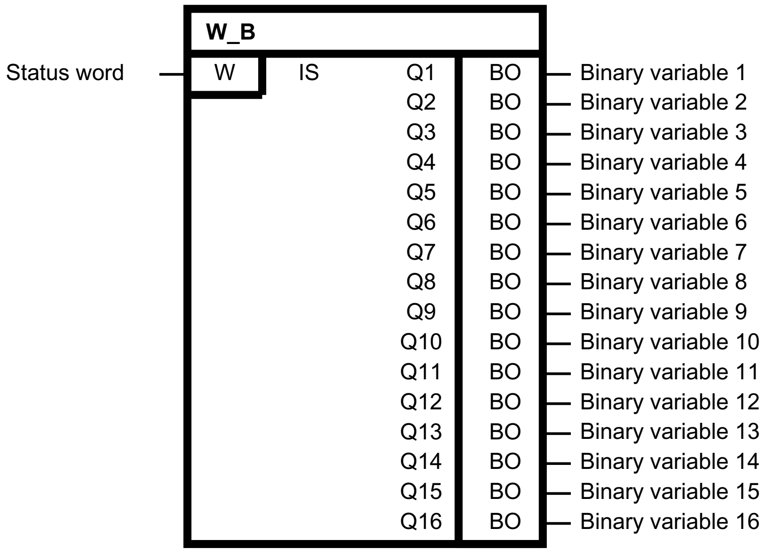
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special character-istics	-

5.32 W_B

Status word to 16 binary variables converter

Symbol



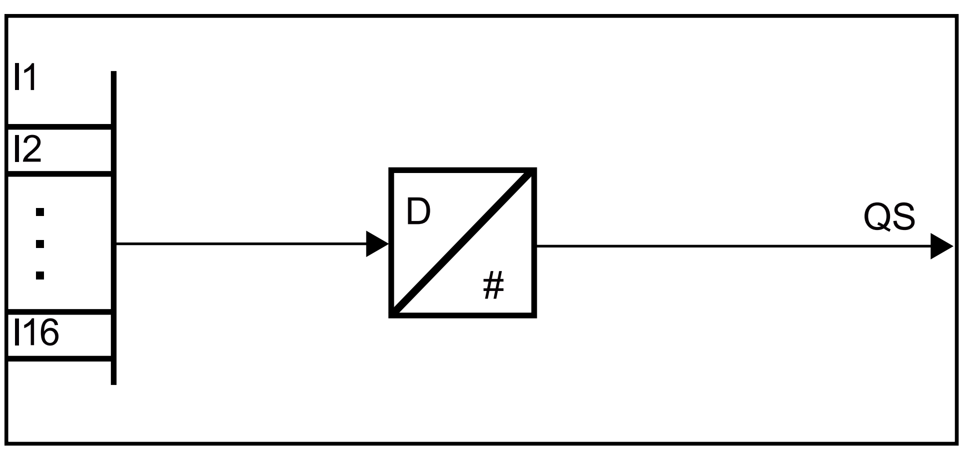
Brief description

Status word decoding to 16 binary variables

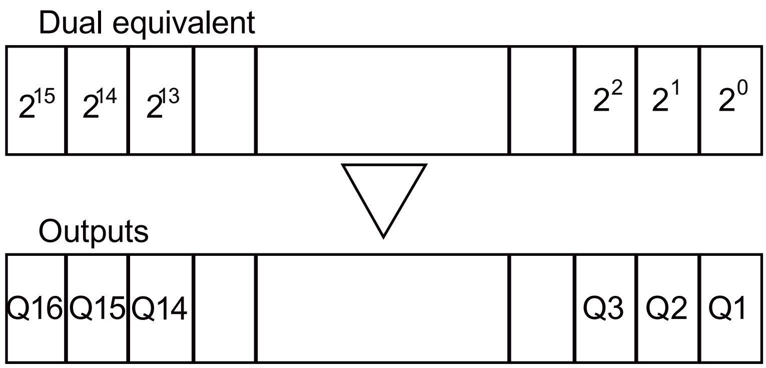
Method of operation

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	Default	Value range	Attrib-utes
IS	Status word	16#0000	WORD	
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	

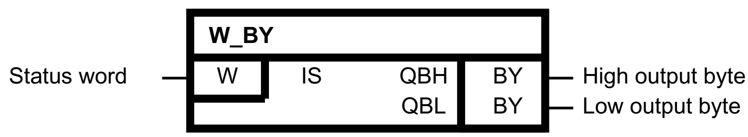
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special character-istics	-

5.33 W_BY

Status word to status byte converter

Symbol



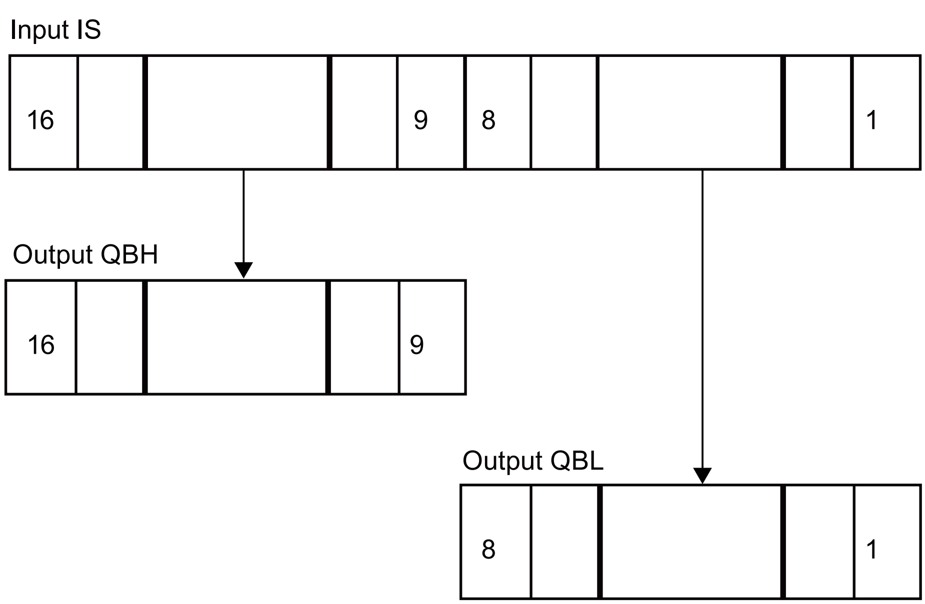
Brief description

Conversion of a word to 2 bytes

Method of operation

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

Conversion scheme



Block connections

Block connection	Description	Default	Value range	Attrib-utes
IS	Status word	16#0000	WORD	
QBH	High output byte	16#00	BYTE	
QBL	Low output byte	16#00	BYTE	

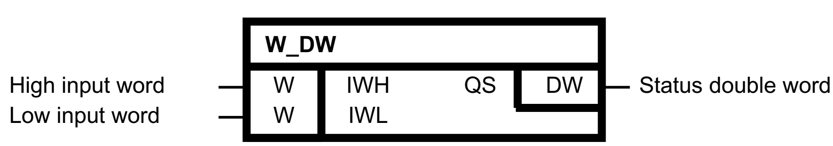
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special character-istics	-

5.34 W_DW

Status word to status double word converter

Symbol



Brief description

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

Method of operation

The input variables are mapped according to the formula

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

to output QS.

Block connections

Block connection	Description	Default	Value range	Attrib-utes
IWH	High input word	16#0000	WORD	
IWH	Low input word	16#0000	WORD	
QS	Status double word	16#00000000	DWORD	

Configuration data

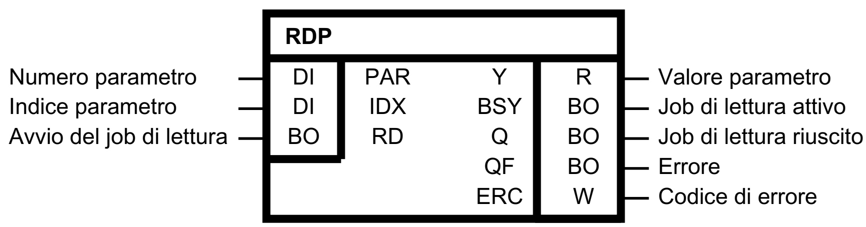
SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	Yes
Special character-istics	-

6 System

6.1 RDP

Reads drive parameters (REAL type)

Symbol



Brief description

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

Method of operation

The parameter number and the index of the parameter to be read must be specified 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.

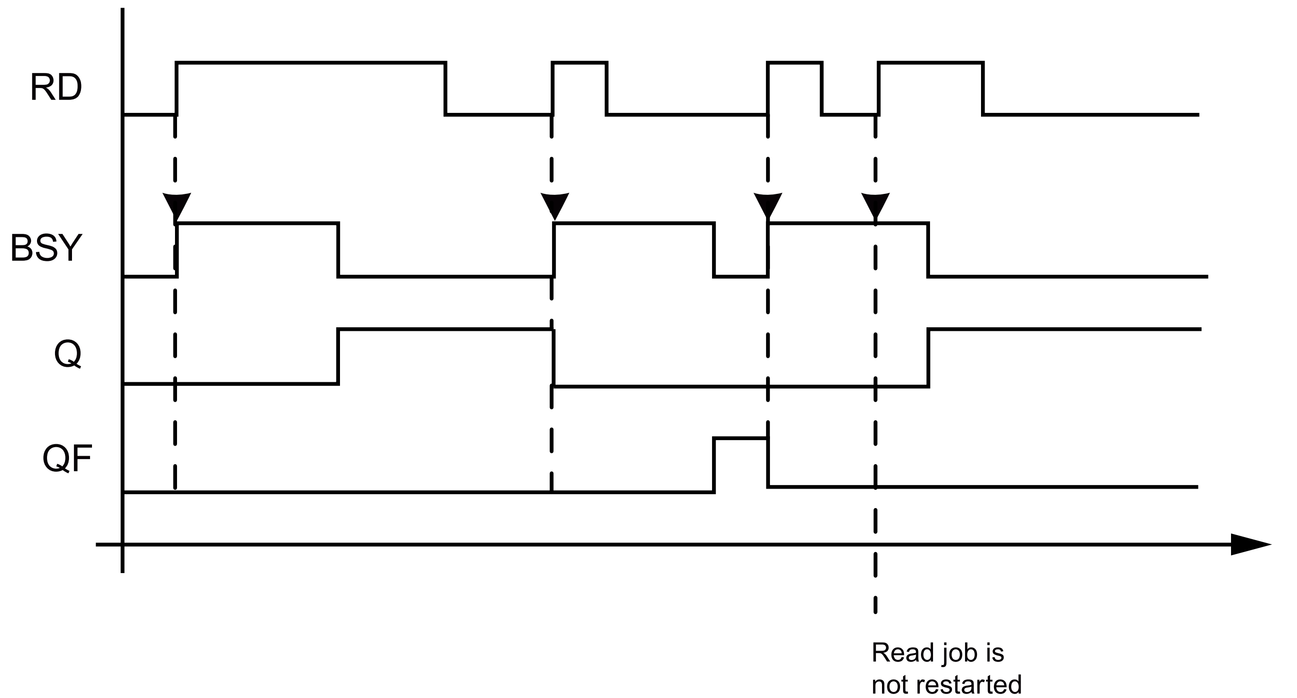
The asynchronous read job is started on a positive edge at input RD. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and can vary from job to job. During an active read job, any additional positive edges at input RD 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 an access, this is signaled with QF = 1. Output Y retains its last value.

For error diagnostics, the error code ERC can be evaluated. ERC corresponds to the error code for parameter accesses according to PROVIdrive 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	Default	Value range	Attrib-utes
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	

Configuration data

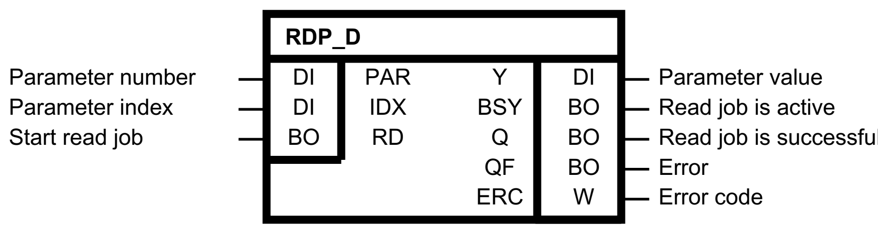
SIMOTION	-
SINAMICS	✓

Can be loaded online	No
Special characteristics	-

6.2 RDP_D

Reads drive parameters (DOUBLE INTEGER type)

Symbol



Brief description

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

Method of operation

The parameter number and the index of the parameter to be read must be specified 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.

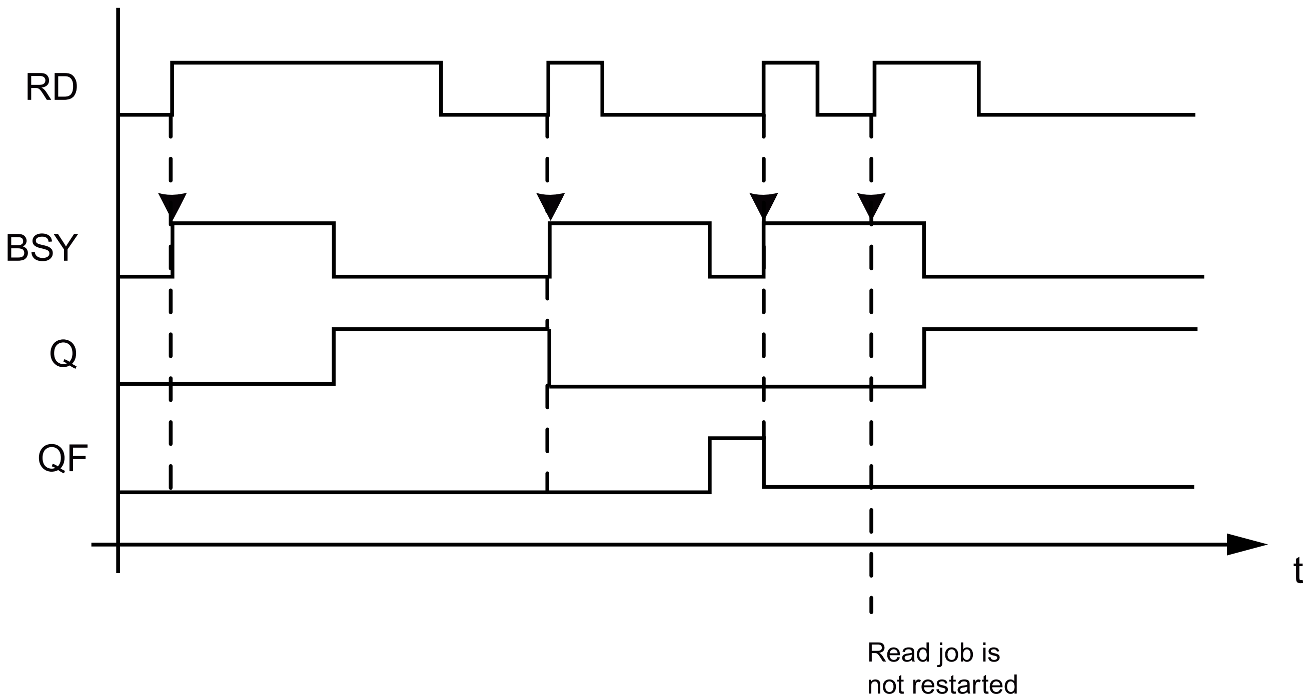
The asynchronous read job is started on a positive edge at input RD. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and can vary from job to job. During an active read job, any additional positive edges at input RD 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 an access, this is signaled with QF = 1. Output Y retains its last value.

For error diagnostics, the error code ERC can be evaluated. ERC corresponds to the error code for parameter accesses according to PROVIdrive 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	Default	Value range	Attrib-utes
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	

Configuration data

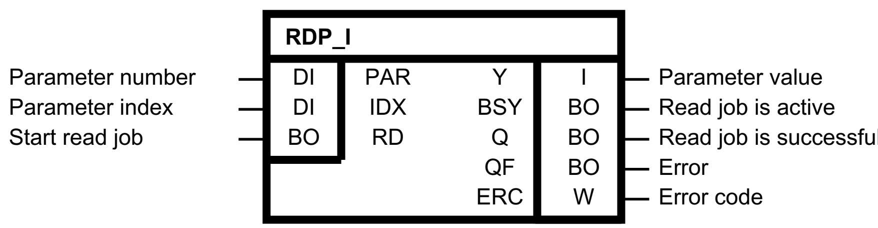
SIMOTION	-
SINAMICS	✓

Can be loaded online	No
Special characteristics	-

6.3 RDP_I

Reads drive parameters (INTEGER type)

Symbol



Brief description

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

Method of operation

The parameter number and the index of the parameter to be read must be specified 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.

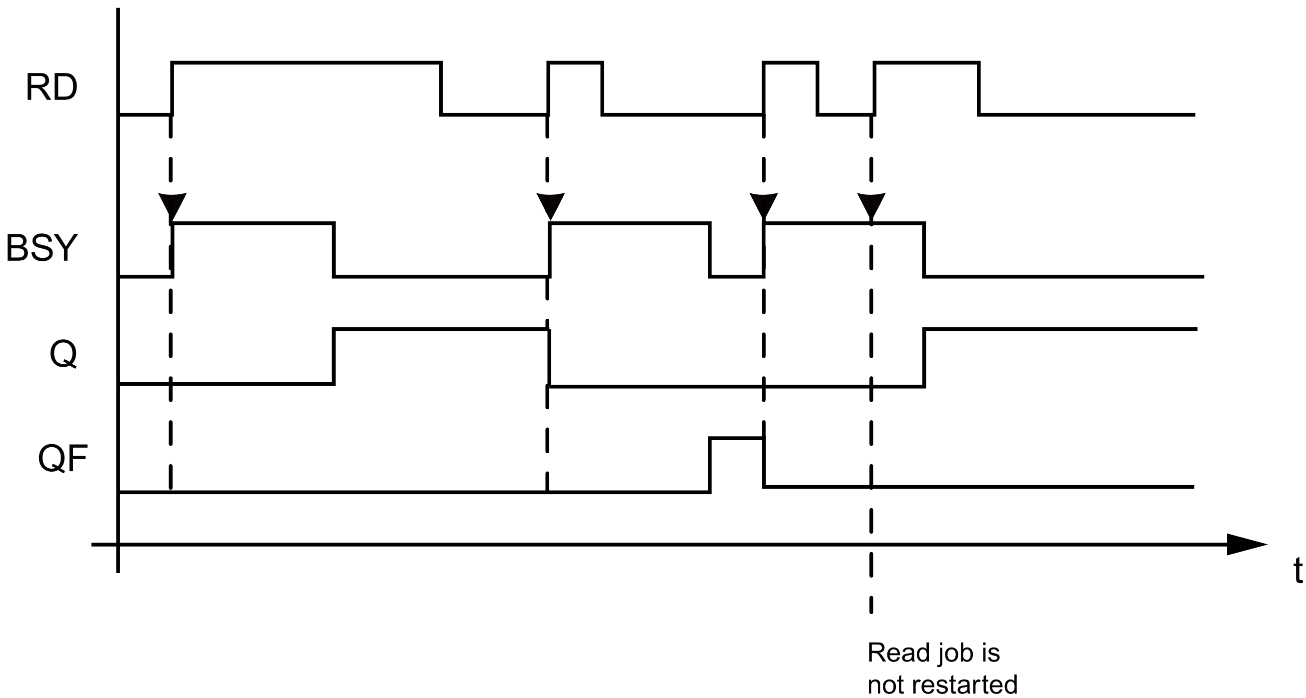
The asynchronous read job is started on a positive edge at input RD. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and can vary from job to job. During an active read job, any additional positive edges at input RD 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 an access, this is signaled with QF = 1. Output Y retains its last value.

For error diagnostics, the error code ERC can be evaluated. ERC corresponds to the error code for parameter accesses according to PROVIdrive 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	Default	Value range	Attrib-utes
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	

Configuration data

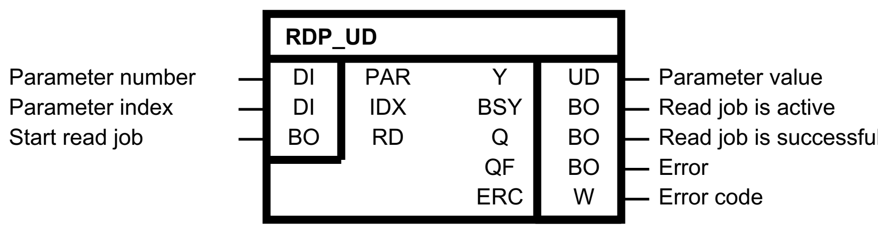
SIMOTION	-
SINAMICS	✓

Can be loaded online	No
Special characteristics	-

6.4 RDP_UD

Reads drive parameters (UNSIGNED DOUBLE INTEGER type)

Symbol



Brief description

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

Method of operation

The parameter number and the index of the parameter to be read must be specified 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.

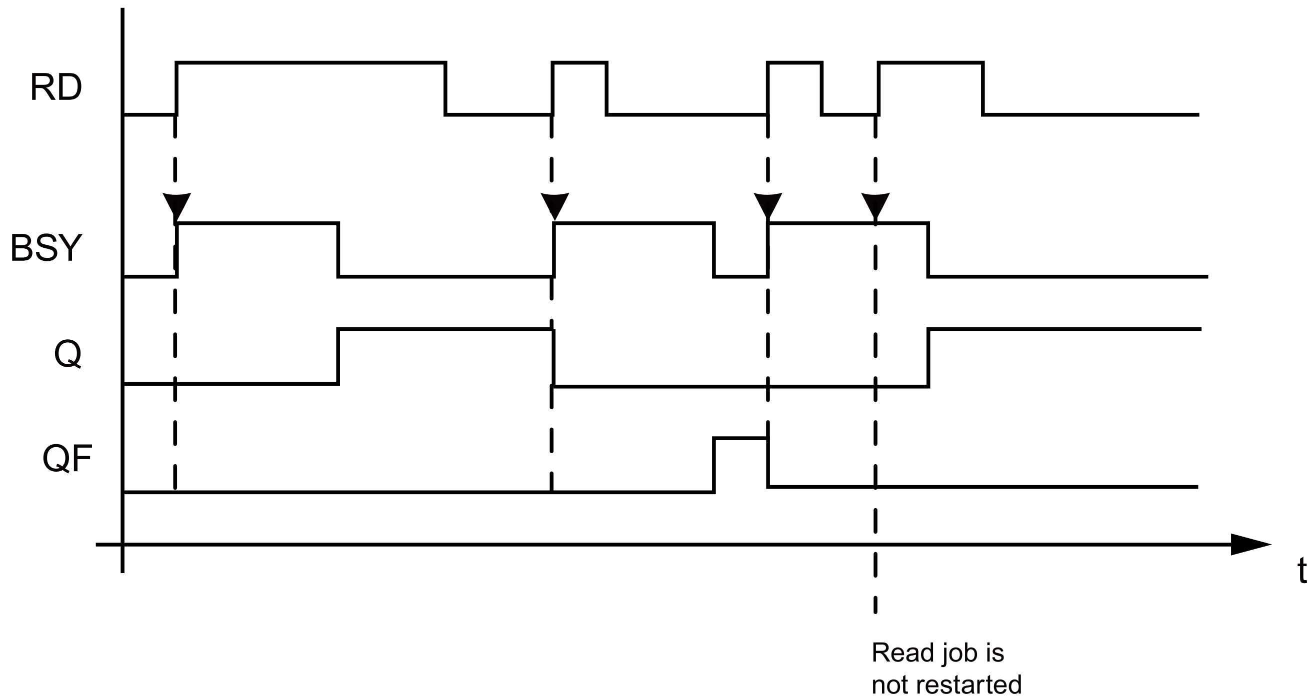
The asynchronous read job is started on a positive edge at input RD. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and can vary from job to job. During an active read job, any additional positive edges at input RD 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 an access, this is signaled with QF = 1. Output Y retains its last value.

For error diagnostics, the error code ERC can be evaluated. ERC corresponds to the error code for parameter accesses according to PROVIdrive 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	Default	Value range	Attrib-utes
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	

Configuration data

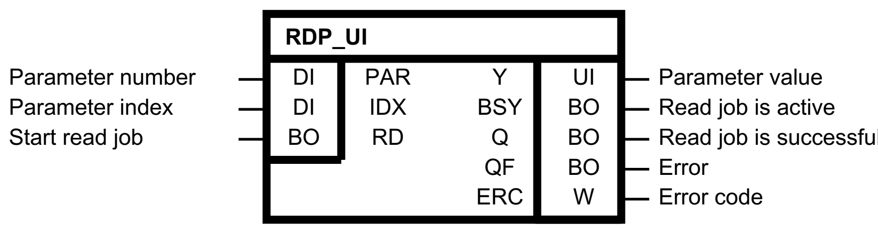
SIMOTION	-
SINAMICS	✓

Can be loaded online	No
Special characteristics	-

6.5 RDP_UI

Reads drive parameters (UNSIGNED INTEGER type)

Symbol



Brief description

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

Method of operation

The parameter number and the index of the parameter to be read must be specified 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.

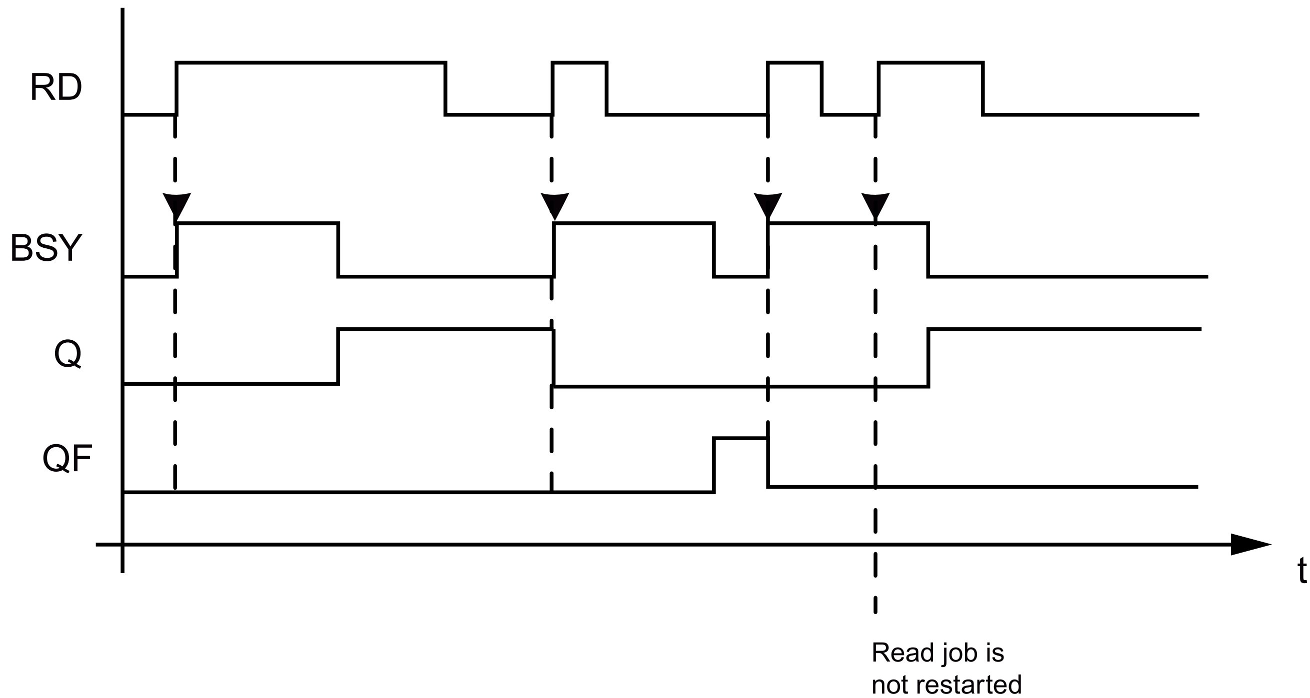
The asynchronous read job is started on a positive edge at input RD. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and can vary from job to job. During an active read job, any additional positive edges at input RD 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 an access, this is signaled with QF = 1. Output Y retains its last value.

For error diagnostics, the error code ERC can be evaluated. ERC corresponds to the error code for parameter accesses according to PROVIdrive 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	Default	Value range	Attrib-utes
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	

Configuration data

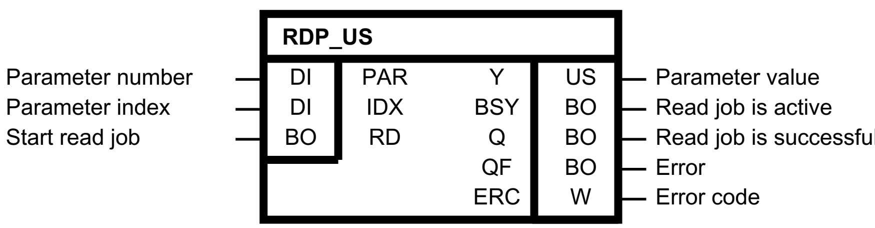
SIMOTION	-
SINAMICS	✓

Can be loaded online	No
Special characteristics	-

6.6 RDP_US

Reads drive parameters (UNSIGNED SHORT INTEGER type)

Symbol



Brief description

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

Method of operation

The parameter number and the index of the parameter to be read must be specified 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.

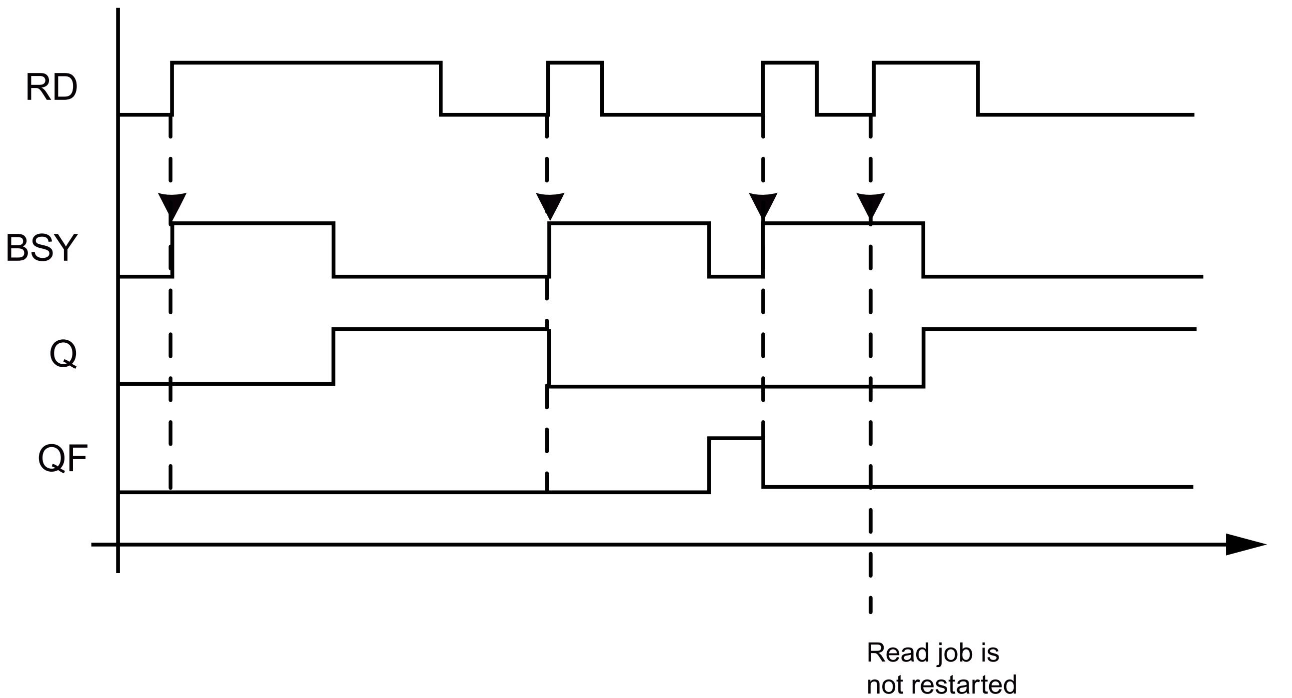
The asynchronous read job is started on a positive edge at input RD. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and can vary from job to job. During an active read job, any additional positive edges at input RD 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 an access, this is signaled with QF = 1. Output Y retains its last value.

For error diagnostics, the error code ERC can be evaluated. ERC corresponds to the error code for parameter accesses according to PROVIdrive 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	Default	Value range	Attrib-utes
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	

Configuration data

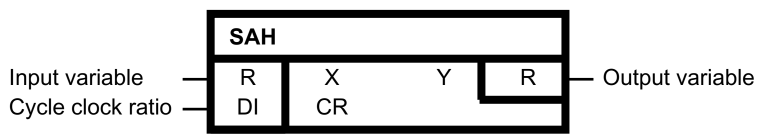
SIMOTION	-
SINAMICS	✓

Can be loaded online	No
Special characteristics	-

6.7 SAH

Sample & hold (REAL type)

Symbol



Brief description

Sample & hold block for equidistant value transfer (REAL type) between blocks with different scanning procedures.

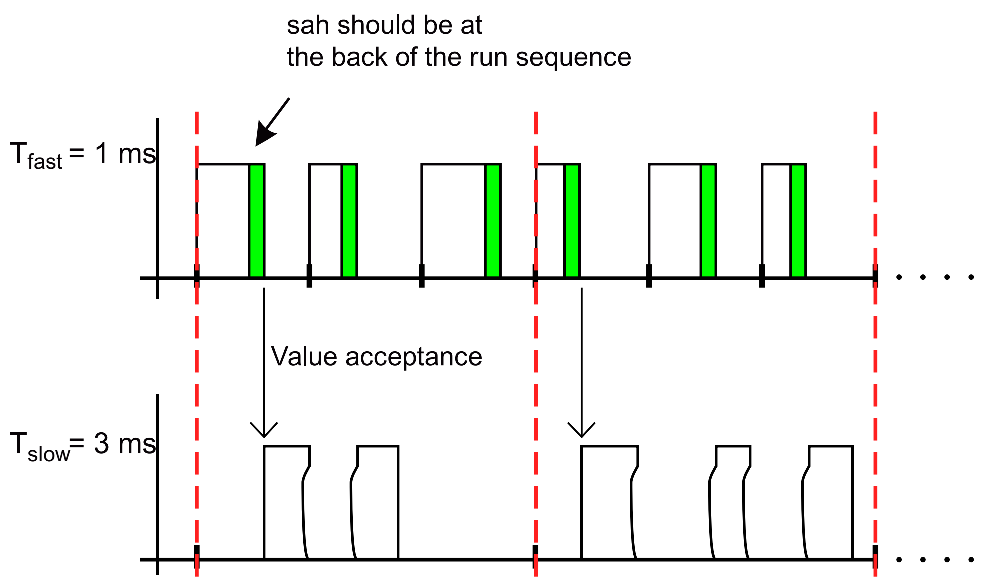
Method of operation

The value of the input variable X is taken over in the output variable Y in every CR cycle. The value transfer cycle clock 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 cycle clock 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.

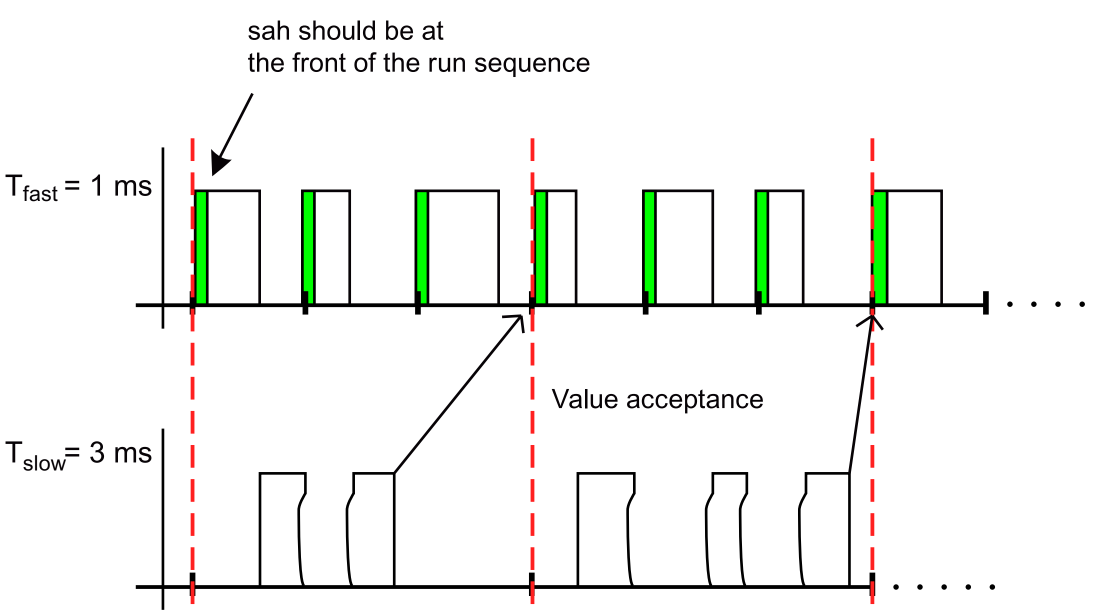
The following figure shows the transfer of values from a 1 ms level to a 3 ms level. The time diagram is shown for the calculation of the execution group.

$$CR = \frac{3ms}{1ms} = 3$$



The following figure shows the transfer of values from a 3 ms level to a 1 ms level. The time diagram is shown for the calculation of the execution group.

$$CR = \frac{3ms}{1ms} = 3$$



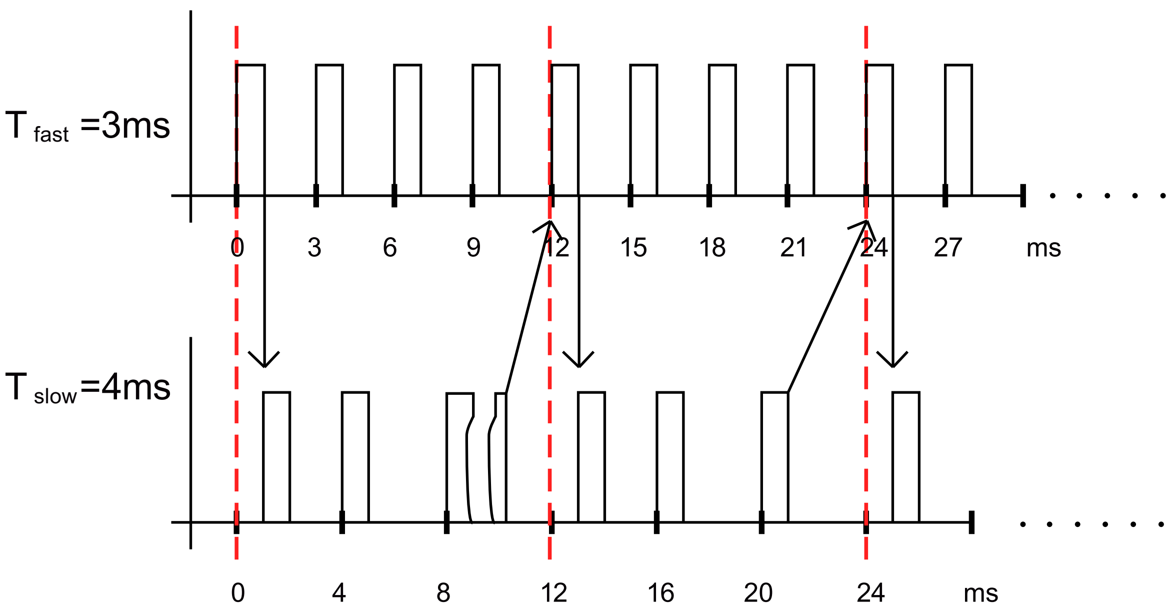
If the slower scan time is not a multiple of the faster scan time, the value can only be transferred consistently if both scanning procedures are restarted synchronously after CR cycles. This corresponds to the least common multiple of both scan times. CR is then calculated as follows:

$$CR = \frac{g\left(T_{fast}, T_{slow}\right)}{T_{fast}}$$

$g(T_{fast}, T_{slow})$: 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.

$$CR = \frac{g(3ms, 4ms)}{3ms} = \frac{12ms}{3ms} = 4$$



To enable values to be transferred at an optimal speed, it is recommended that the slower scan time is always a multiple of the faster scan time.

Block connections

Block connection	Description	Default	Value range	Attrib-utes
X	Input variable	0.0	REAL	
CR	Cycle clock ratio	1	0 - (2^{31}-1)	
Y	Output variable	0.0	REAL	

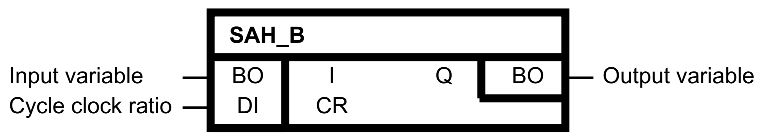
Configuration data

SIMOTION	-
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

6.8 SAH_B

Sample & hold (BOOL type)

Symbol



Brief description

Sample & hold block for equidistant value transfer (BOOL type) between blocks with different scanning procedures.

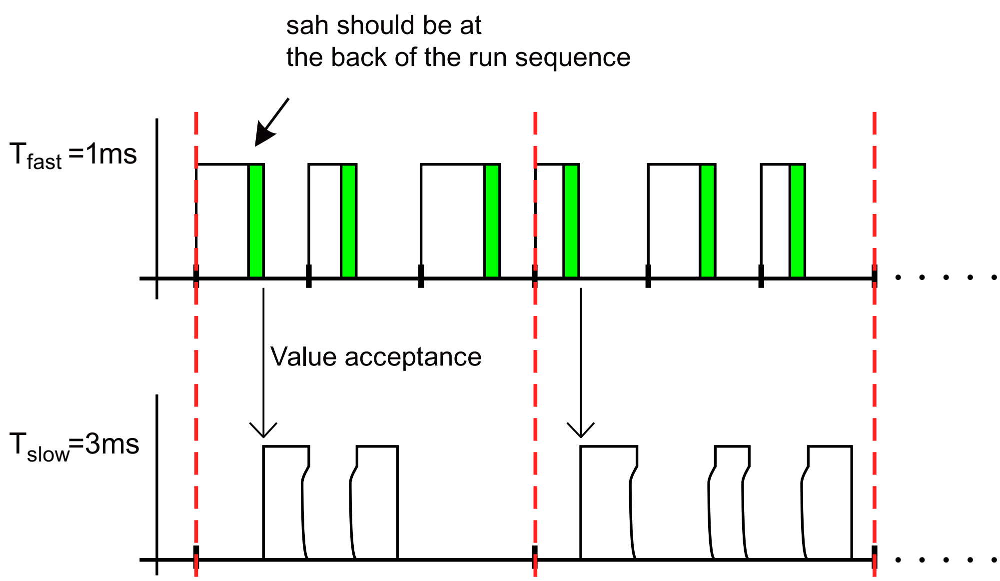
Method of operation

The value of the input variable I is taken over in the output variable Q in every CR cycle. The value transfer cycle clock 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 cycle clock 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.

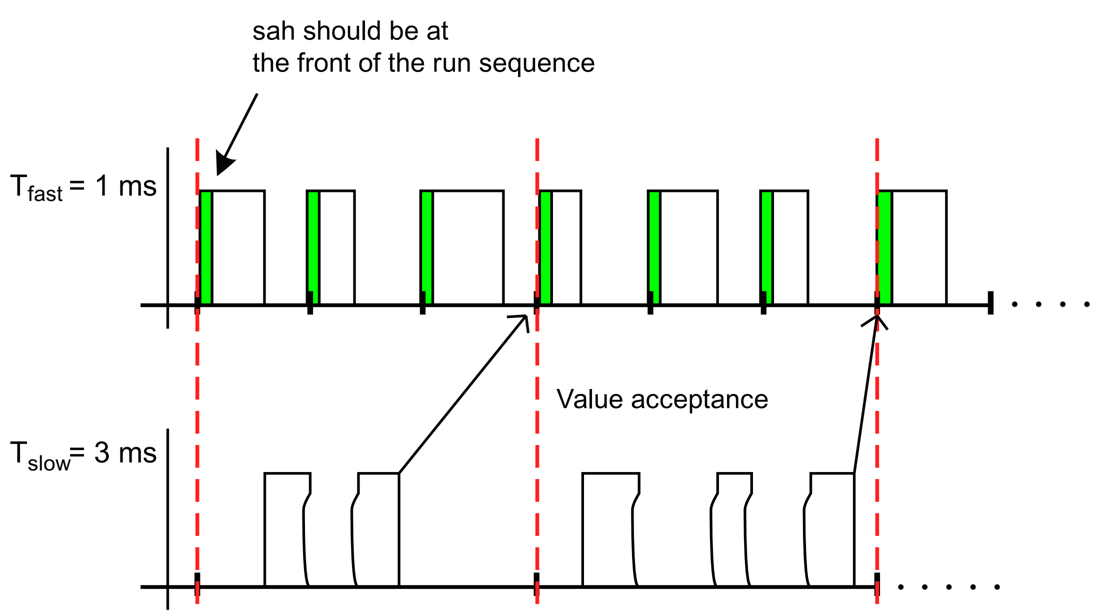
The following figure shows the transfer of values from a 1 ms level to a 3 ms level. The time diagram is shown for the calculation of the execution group.

$$CR = \frac{3ms}{1ms} = 3$$



The following figure shows the transfer of values from a 3 ms level to a 1 ms level. The time diagram is shown for the calculation of the execution group.

$$CR = \frac{3ms}{1ms} = 3$$



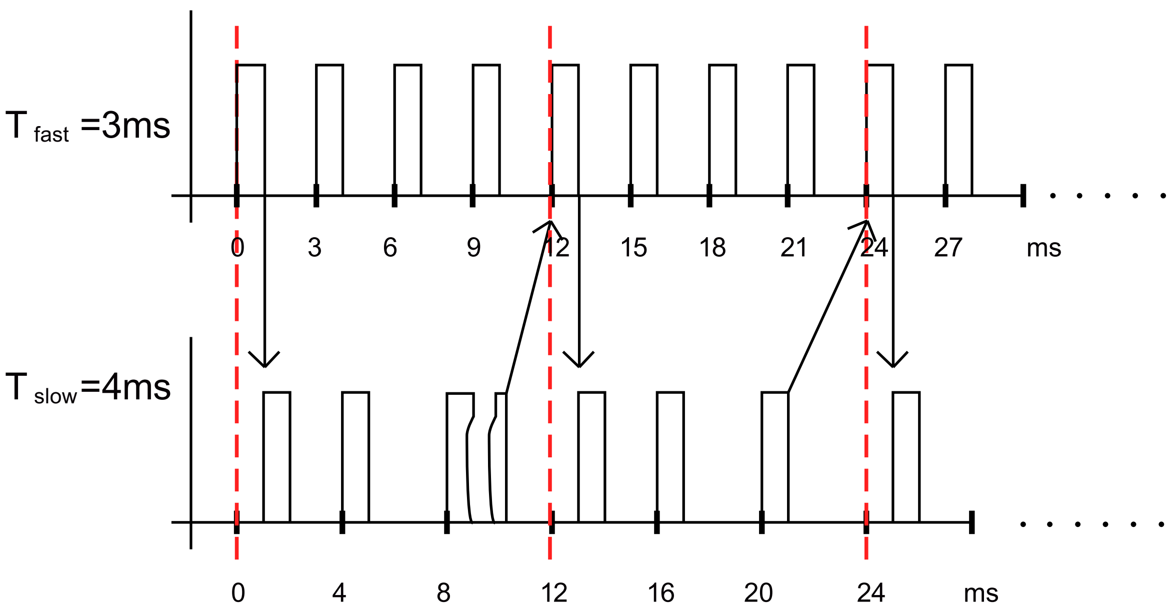
If the slower scan time is not a multiple of the faster scan time, the value can only be transferred consistently if both scanning procedures are restarted synchronously after CR cycles. This corresponds to the least common multiple of both scan times. CR is then calculated as follows:

$$CR = \frac{g\left(T_{fast}, T_{slow}\right)}{T_{fast}}$$

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.

$$CR = \frac{g(3ms, 4ms)}{3ms} = \frac{12ms}{3ms} = 4$$



To enable values to be transferred at an optimal speed, it is recommended that the slower scan time is always a multiple of the faster scan time.

Block connections

Block connection	Description	Default	Value range	Attrib-utes
I	Input variable	0	0/1	
CR	Cycle clock ratio	1	0 - (2^{31}-1)	
Q	Output variable	0	0/1	

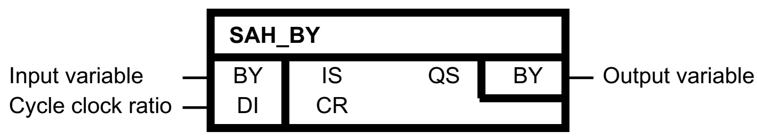
Configuration data

SIMOTION	-
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

6.9 SAH_BY

Sample & hold (BYTE type)

Symbol



Brief description

Sample & hold block for the equidistant value transfer (BYTE type) between blocks with different sampling times.

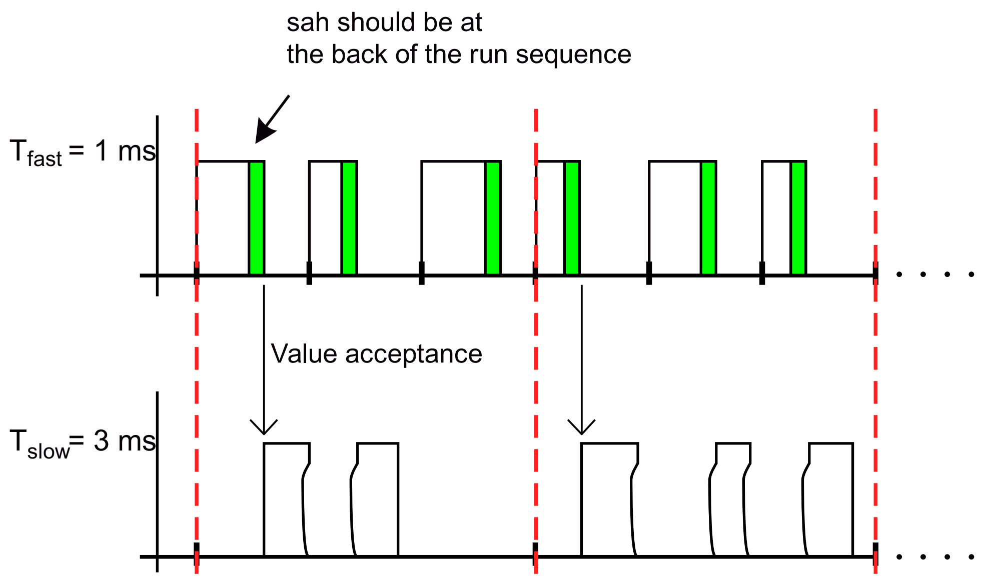
Method of operation

The value of the input variable IS is taken over in the output variable QS in every CR cycle. The value transfer cycle clock 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 cycle clock 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.

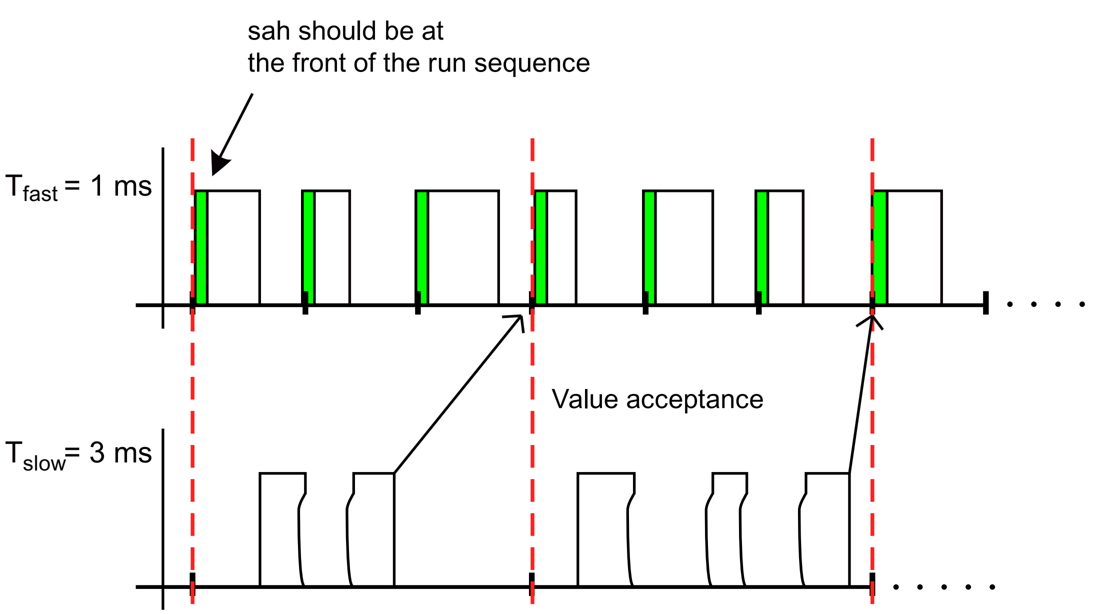
The following figure shows the transfer of values from a 1 ms level to a 3 ms level. The time diagram is shown for the calculation of the execution group.

$$CR = \frac{3ms}{1ms} = 3$$



The following figure shows the transfer of values from a 3 ms level to a 1 ms level. The time diagram is shown for the calculation of the execution group.

$$CR = \frac{3ms}{1ms} = 3$$



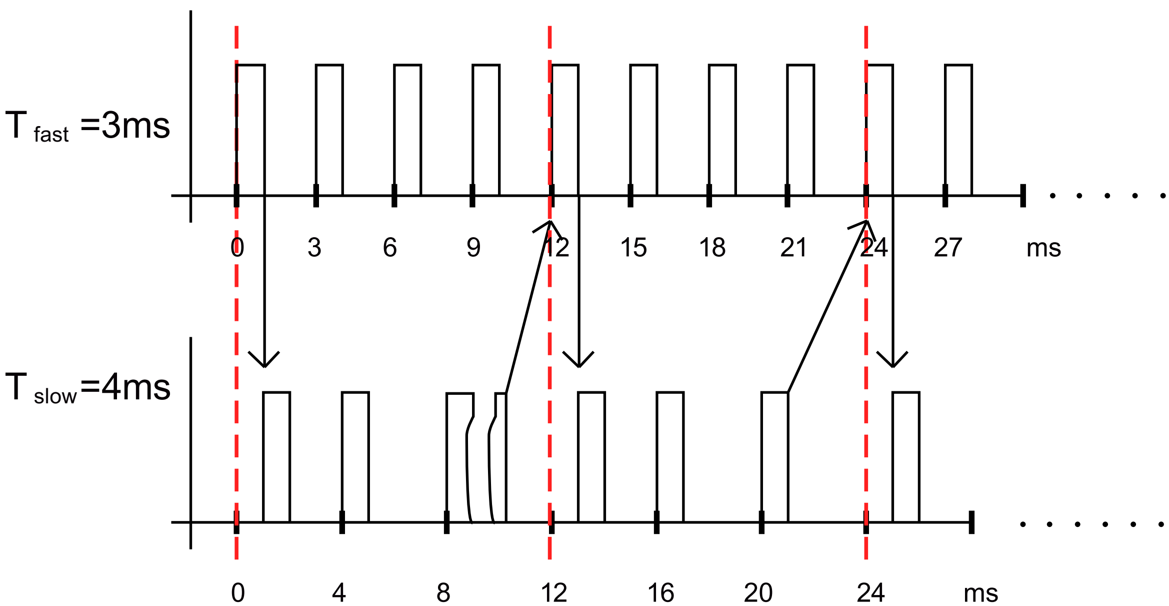
If the slower scan time is not a multiple of the faster scan time, the value can only be transferred consistently if both scanning procedures are restarted synchronously after CR cycles. This corresponds to the least common multiple of both scan times. CR is then calculated as follows:

$$CR = \frac{g\left(T_{fast}, T_{slow}\right)}{T_{fast}}$$

$g(T_{fast}, T_{slow})$: least common multiple

The following figure shows the value transfer for T_{fast} = 3 ms and T_{slow} = 4 ms. The value transfer is made in both directions.

$$CR = \frac{g(3ms, 4ms)}{3ms} = \frac{12ms}{3ms} = 4$$



To enable values to be transferred at an optimal speed, it is recommended that the slower scan time is always a multiple of the faster scan time.

Block connections

Block connection	Description	Default	Value range	Attrib-utes
IS	Input variable	16#00	BYTE	
CR	Cycle clock ratio	1	0 - (2^{31}-1)	
QS	Output variable	16#00	BYTE	

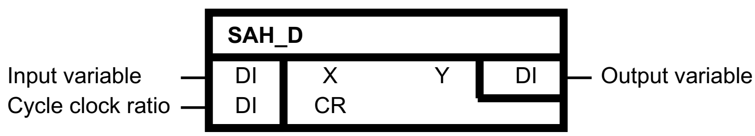
Configuration data

SIMOTION	-
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

6.10 SAH_D

Sample & hold (DOUBLE INTEGER type)

Symbol



Brief description

Sample & hold block for equidistant value transfer (DOUBLE INTEGER type) between blocks with different scanning procedures.

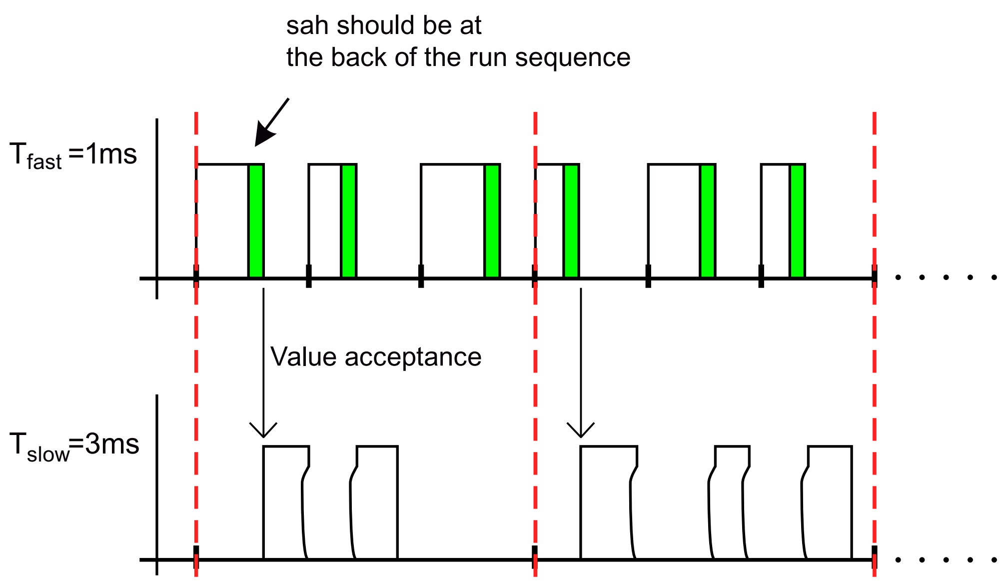
Method of operation

The value of the input variable X is taken over in the output variable Y in every CR cycle. The value transfer cycle clock 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 cycle clock 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.

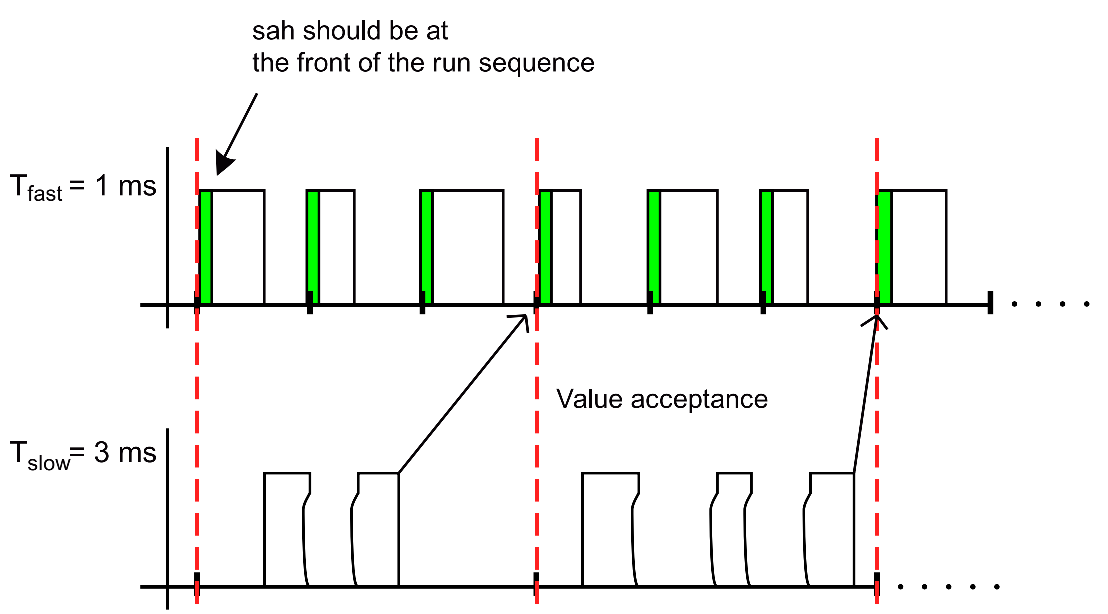
The following figure shows the transfer of values from a 1 ms level to a 3 ms level. The time diagram is shown for the calculation of the execution group.

$$CR = \frac{3ms}{1ms} = 3$$



The following figure shows the transfer of values from a 3 ms level to a 1 ms level. The time diagram is shown for the calculation of the execution group.

$$CR = \frac{3ms}{1ms} = 3$$



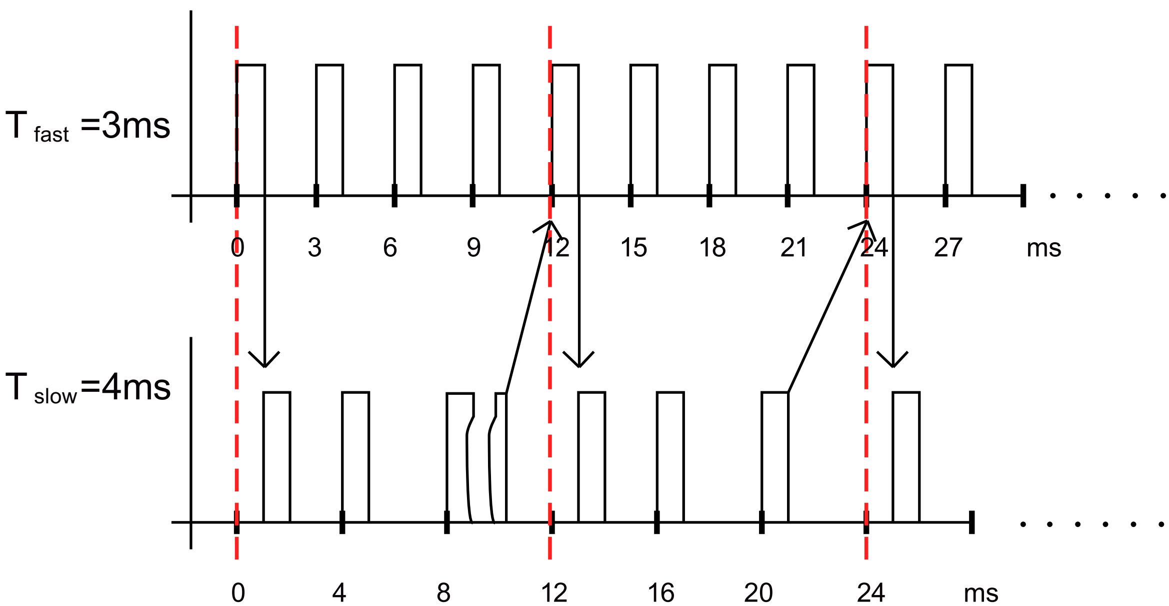
If the slower scan time is not a multiple of the faster scan time, the value can only be transferred consistently if both scanning procedures are restarted synchronously after CR cycles. This corresponds to the least common multiple of both scan times. CR is then calculated as follows:

$$CR = \frac{g\left(T_{fast}, T_{slow}\right)}{T_{fast}}$$

$g(T_{fast}, T_{slow})$: least common multiple

The following figure shows the value transfer for T_{fast} = 3 ms and T_{slow} = 4 ms. The value transfer is made in both directions.

$$CR = \frac{g(3ms, 4ms)}{3ms} = \frac{12ms}{3ms} = 4$$



To enable values to be transferred at an optimal speed, it is recommended that the slower scan time is always a multiple of the faster scan time.

Block connections

Block connection	Description	Default	Value range	Attrib-utes
X	Input variable	0	DINT	
CR	Cycle clock ratio	1	0 - (2^{31}-1)	
Y	Output variable	0	DINT	

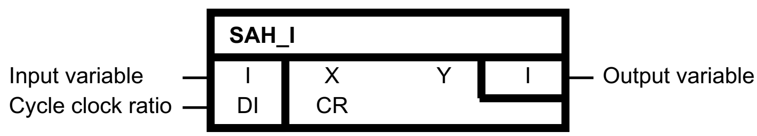
Configuration data

SIMOTION	-
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

6.11 SAH_I

Sample & hold (INTEGER type)

Symbol



Brief description

Sample & hold block for the equidistant value transfer (INTEGER type) between blocks with different sampling times.

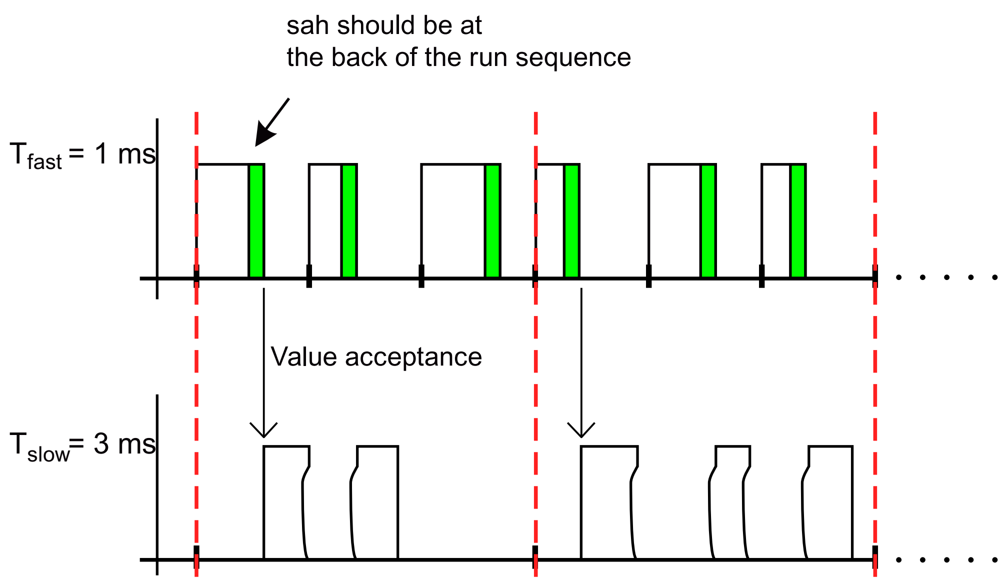
Method of operation

The value of the input variable X is taken over in the output variable Y in every CR cycle. The value transfer cycle clock 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 cycle clock 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.

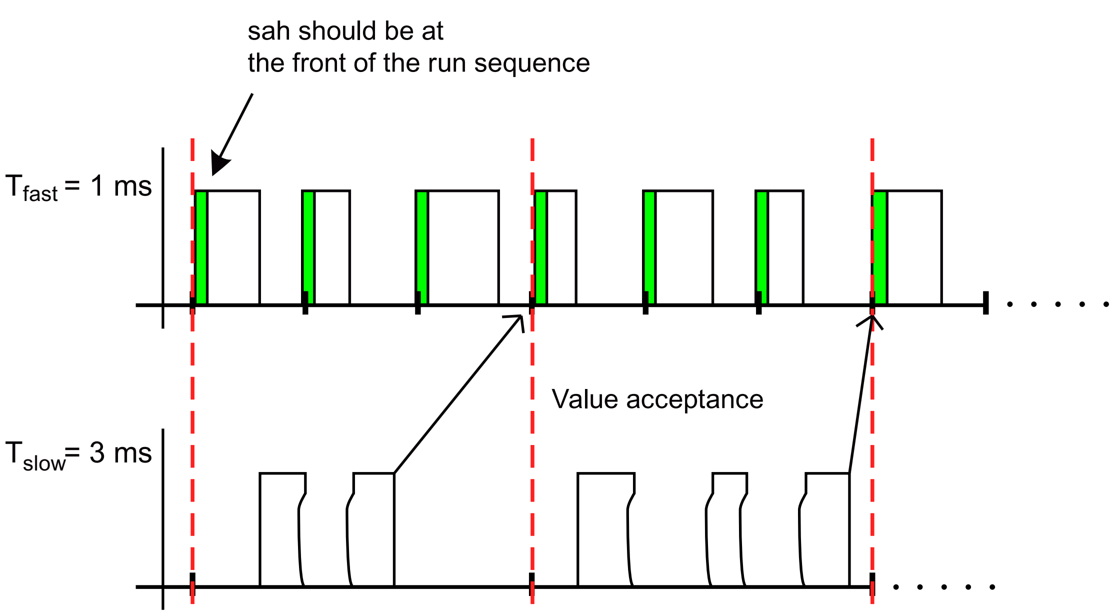
The following figure shows the transfer of values from a 1 ms level to a 3 ms level. The time diagram is shown for the calculation of the execution group.

$$CR = \frac{3ms}{1ms} = 3$$



The following figure shows the transfer of values from a 3 ms level to a 1 ms level. The time diagram is shown for the calculation of the execution group.

$$CR = \frac{3ms}{1ms} = 3$$



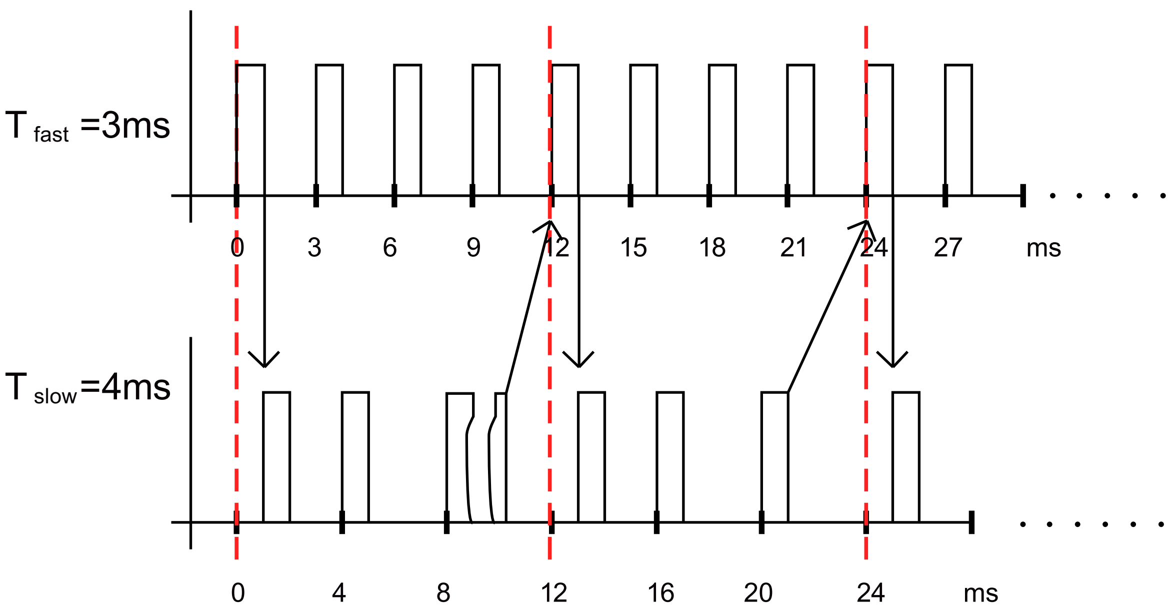
If the slower scan time is not a multiple of the faster scan time, the value can only be transferred consistently if both scanning procedures are restarted synchronously after CR cycles. This corresponds to the least common multiple of both scan times. CR is then calculated as follows:

$$CR = \frac{g\left(T_{fast}, T_{slow}\right)}{T_{fast}}$$

$g(T_{fast}, T_{slow})$: least common multiple

The following figure shows the value transfer for T_{fast} = 3 ms and T_{slow} = 4 ms. The value transfer is made in both directions.

$$CR = \frac{g(3ms, 4ms)}{3ms} = \frac{12ms}{3ms} = 4$$



To enable values to be transferred at an optimal speed, it is recommended that the slower scan time is always a multiple of the faster scan time.

Block connections

Block connection	Description	Default	Value range	Attrib-utes
X	Input variable	0	INT	
CR	Cycle clock ratio	1	0 - (2^{31}-1)	
Y	Output variable	0	INT	

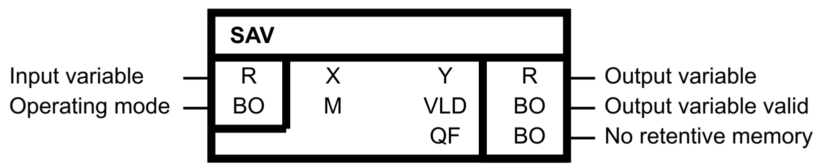
Configuration data

SIMOTION	-
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

6.12 SAV

Value buffering (REAL type)

Symbol



Brief description

SAV (Save) enables retentive storage of a REAL-type input variable.

Method of operation

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 on the target device 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 of a block has been changed and transferred to the target system.

The value is retained when:

- 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.
- Another SAV block has been added or removed.
- A download of the configuration is performed after an update of the DCBLIB.
- Another DO has been added or removed and downloaded to the target device.
- Another chart has been added or removed and downloaded to the target device.
- 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 was 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)

The instance name could look like this:

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	Default	Value range	Attrib-utes
X	Input variable	0.0	REAL	
M	Operating mode	0	0/1	
Y	Output variable	0.0	REAL	
VLD	Output variable valid	0	0/1	
QF	No retentive memory	0	0/1	

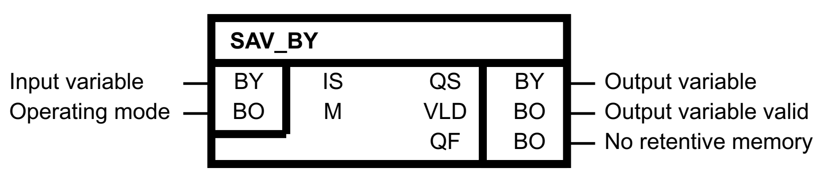
Configuration data

SIMOTION	✓
SINAMICS	✓
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 shortcut menu of the drive unit).

6.13 SAV_BY

Value buffering (BYTE type)

Symbol



Brief description

SAV_BY (Save) enables retentive storage of a BYTE-type input variable.

Method of operation

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 on the target device 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 of a block has been changed and transferred to the target system.

The value is retained when:

- 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.
- Another SAV block has been added or removed.

- A download of the configuration is performed after an update of the DCBLIB.
- Another DO has been added or removed and downloaded to the target device.
- Another chart has been added or removed and downloaded to the target device.
- 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 was 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)

The instance name could look like this:

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	Default	Value range	Attrib-utes
IS	Input variable	16#00	BYTE	
M	Operating mode	0	0/1	
QS	Output variable	16#00	BYTE	
VLD	Output variable valid	0	0/1	
QF	No retentive memory	0	0/1	

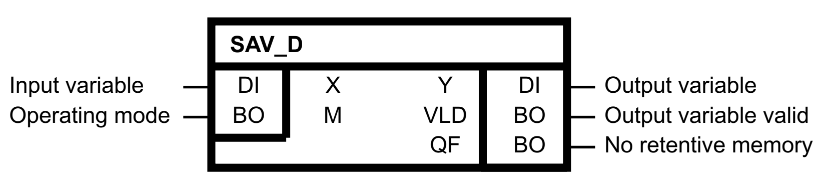
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	No
Special character-istics	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 shortcut menu of the drive unit).

6.14 SAV_D

Value buffering (DOUBLE INTEGER type)

Symbol



Brief description

SAV_D (Save) enables retentive storage of a DOUBLE INTEGER-type input variable.

Method of operation

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 on the target device 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 of a block has been changed and transferred to the target system.

The value is retained when:

- 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.
- Another SAV block has been added or removed.
- A download of the configuration is performed after an update of the DCBLIB.
- Another DO has been added or removed and downloaded to the target device.
- Another chart has been added or removed and downloaded to the target device.
- 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 was 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)

The instance name could look like this:

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	Default	Value range	Attrib-utes
X	Input variable	0	DINT	
M	Operating mode	0	0/1	
Y	Output variable	0	DINT	
VLD	Output variable valid	0	0/1	
QF	No retentive memory	0	0/1	

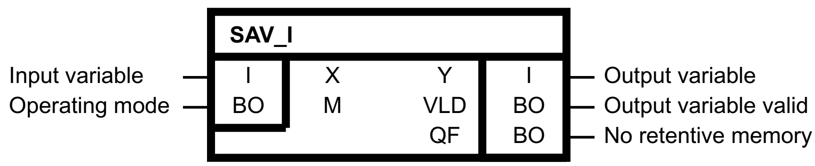
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be inserted on-line	No
Special character-istics	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 shortcut menu of the drive unit).

6.15 SAV_I

Value buffering (INTEGER type)

Symbol



Brief description

SAV_I (Save) enables retentive storage of an INTEGER-type input variable.

Method of operation

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 on the target device 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 of a block has been changed and transferred to the target system.

The value is retained when:

- 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.
- Another SAV block has been added or removed.
- A download of the configuration is performed after an update of the DCBLIB.
- Another DO has been added or removed and downloaded to the target device.
- Another chart has been added or removed and downloaded to the target device.
- 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 was 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)

The instance name could look like this:

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	Default	Value range	Attrib-utes
X	Input variable	0	INT	
M	Operating mode	0	0/1	
Y	Output variable	0	INT	
VLD	Output variable valid	0	0/1	
QF	No retentive memory	0	0/1	

Configuration data

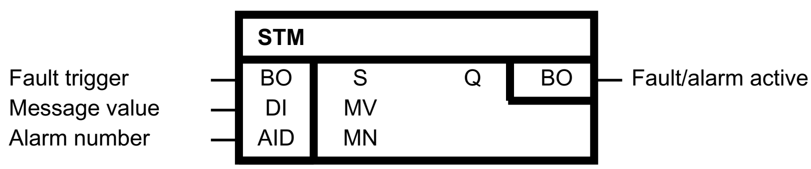
SIMOTION	✓
SINAMICS	✓

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 shortcut menu of the drive unit).

6.16 STM

Fault/alarm trigger

Symbol



Brief description

A predefined message (fault or alarm) 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 alarm buffer of the DO. The following specifications apply to this block type:

- The message number (fault/alarm number) assigned to an instance must be in the range 51050 to 51069 (default value is 51050).
- A message number can be repeated at multiple instances in the DO (message can be issued from different instances). However, for performance reasons, the STM block is not designed for multiple instantiation. The figure below shows the resulting behavior when there is multiple instantiation with the same message number for a fault on the same DO. Without additional RC circuitry, the block instances with the same message number are not coordinated (in any case, this would not be possible if the instances were running in different scan times). For this reason, we recommend assigning a unique message number in the DO for each instance.
- The message text is predefined and cannot be changed (see table below).
- The message type 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] "Set fault number for fault response" and
  - p2101[0..19] "Fault response setting"
- 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
  - p2127[0..19] "Acknowledgement mode setting"

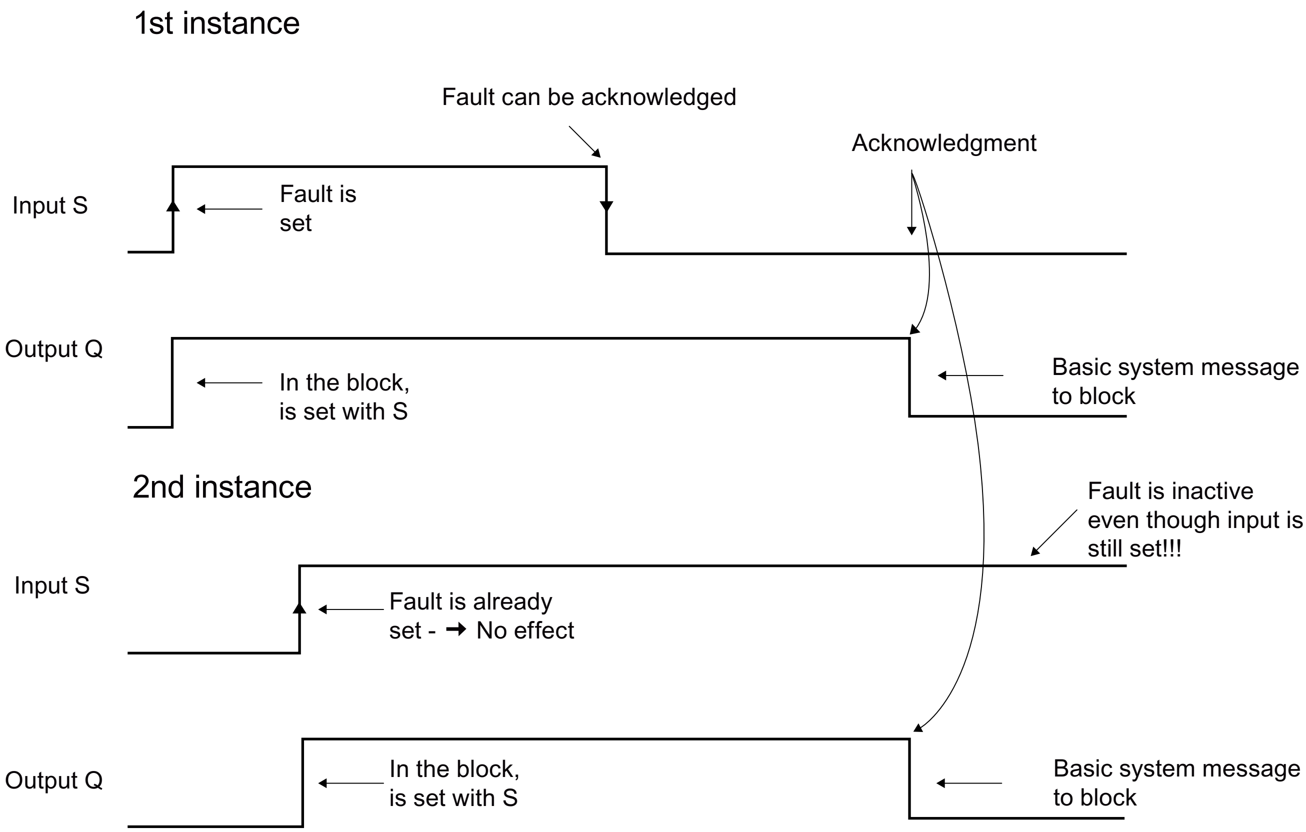
The table below specifies the default settings for the attributes. Possible options for the various settings can be found in the user documentation:

Message type	Alarm number	Reaction	Acknowledgment	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: Alarm A5106x Additional value: %d(x:= 0 to 9)

Method of operation

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).

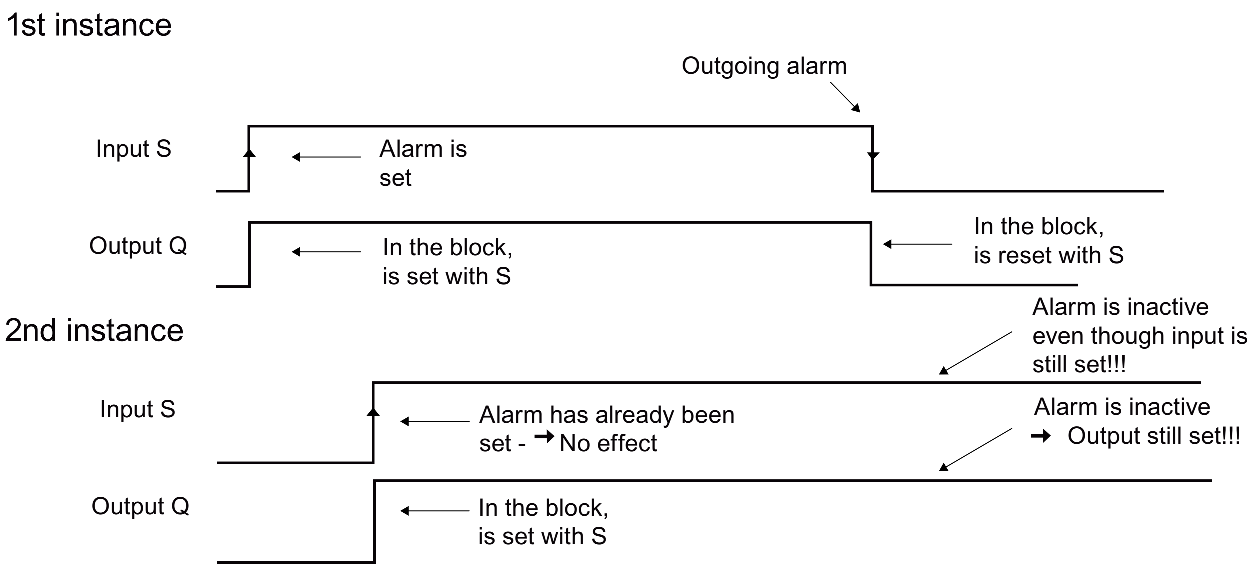
Input MV can be used to add additional information (fault value) for the fault. The value is transferred to input S when the fault is triggered on a positive edge and is entered in the fault buffer of the DO.



Example of two-fold instantiation with the same fault number on one DO (without additional RC circuitry)

Method of operation

The number of the alarm to be triggered (A51060 - A51069) must be specified at input MN. A positive edge at input S triggers the alarm 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. Alarms are self-acknowledging and are acknowledged when input S is reset (see figure below). Input MV can be used to provide additional information (alarm value) for the alarm, which is also entered in the alarm buffer.



Example of two-fold instantiation with the same alarm number at one DO (without additional RC circuitry)

Block connections

Block connection	Description	Default	Value range	Attrib-utes
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	

Configuration data

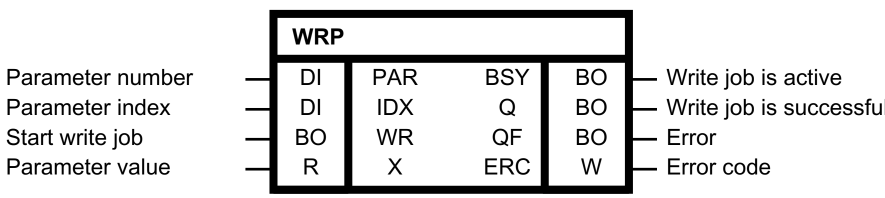
SIMOTION	-
SINAMICS	✓

Can be inserted online	No
Special characteristics	-

6.17 WRP

Writes drive parameters (REAL type)

Symbol



Brief description

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

NOTICE
Do not use with Safety Integrated
The block for programming a drive parameter must not be used for safety reasons to change parameters of the SINAMICS Safety Integrated Functions. DCC is not considered suitable for safety applications in the context of functional safety (Safety Integrated).

Method of operation

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 calculated. 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.

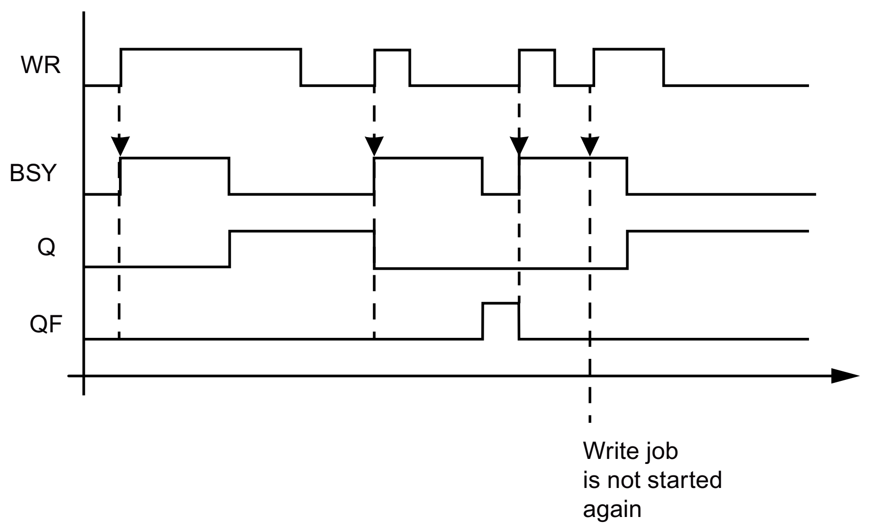
Output Q = 1 indicates that the parameter has been successfully written. If an error occurs during an access, this is signaled with QF = 1.

For error diagnostics, the error code ERC can be evaluated. ERC corresponds to the error code for parameter accesses according to PROVIdrive 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	Default	Value range	Attrib-utes
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	

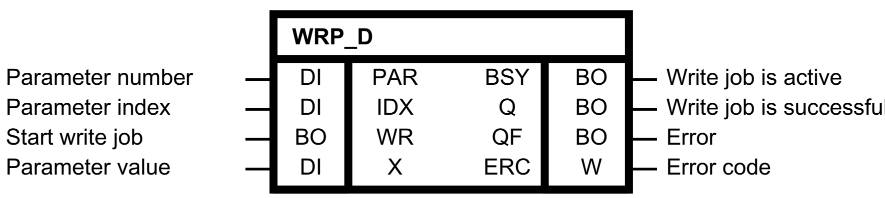
Configuration data

SIMOTION	-
SINAMICS	✓
Can be loaded on-line	No
Special character-istics	-

6.18 WRP_D

Writes drive parameters (DOUBLE INTEGER type)

Symbol



Brief description

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

NOTICE
Do not use with Safety Integrated
The block for programming a drive parameter must not be used for safety reasons to change parameters of the SINAMICS Safety Integrated Functions. DCC is not considered suitable for safety applications in the context of functional safety (Safety Integrated).

Method of operation

The parameter number and the index of the parameter to be written must be specified 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 calculated. It is not possible to access parameters on several drive objects.

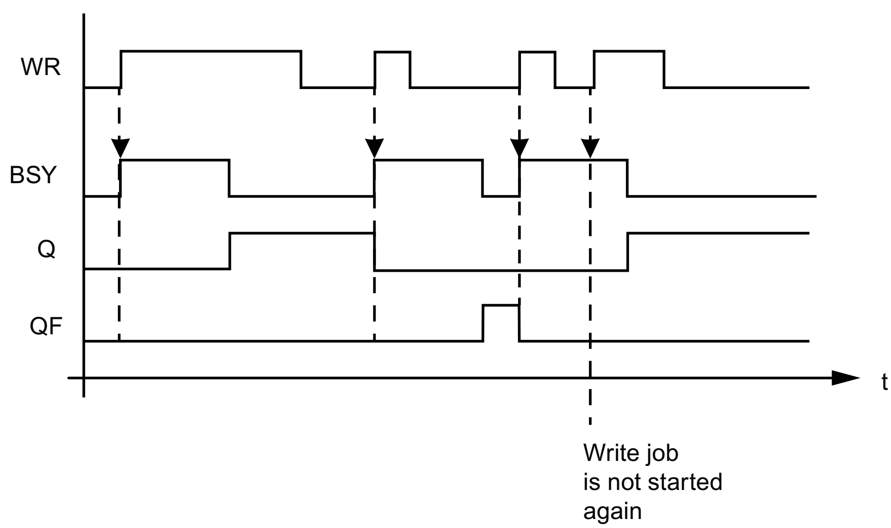
The parameter value is specified via input X. The asynchronous write job can be started on a positive edge at input WR. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and can vary from job to job. During an active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

Output Q = 1 indicates that the parameter has been successfully written. If an error occurs during an access, this is signaled with QF = 1.

For error diagnostics, the error code ERC can be evaluated. ERC corresponds to the error code for parameter accesses according to PROVIdrive 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	Default	Value range	Attrib-utes
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	

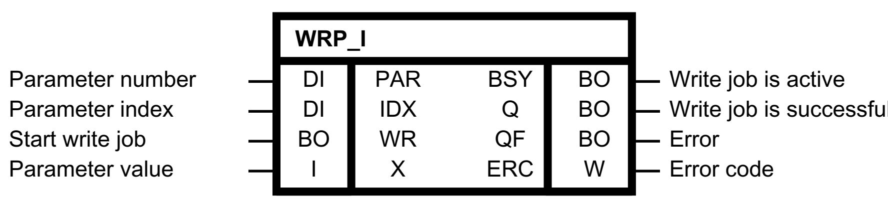
Configuration data

SIMOTION	-
SINAMICS	✓
Can be loaded on-line	No
Special character-istics	-

6.19 WRP_I

Writes drive parameters (INTEGER type)

Symbol



Brief description

The block allows asynchronous writing of drive parameters of the INTEGER type on the local drive object

NOTICE
Do not use with Safety Integrated
The block for programming a drive parameter must not be used for safety reasons to change parameters of the SINAMICS Safety Integrated Functions. DCC is not considered suitable for safety applications in the context of functional safety (Safety Integrated).

Method of operation

The parameter number and the index of the parameter to be written must be specified 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 calculated. It is not possible to access parameters on several drive objects.

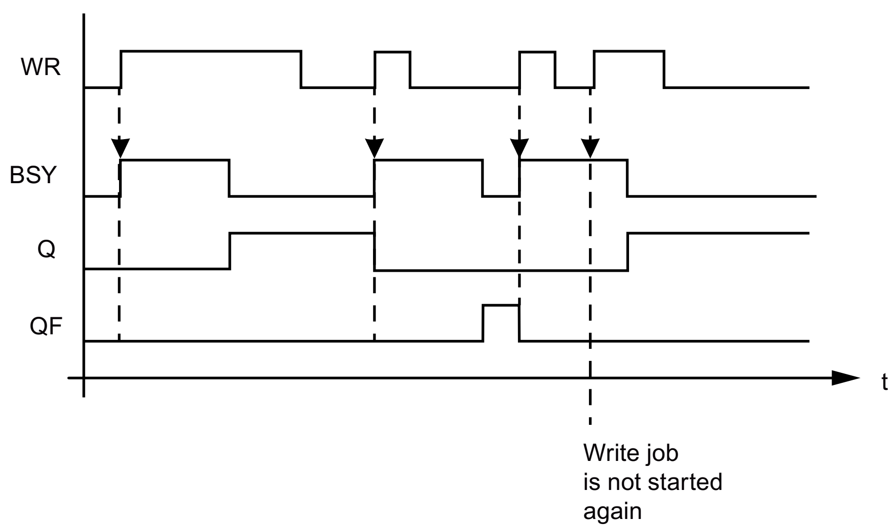
The parameter value is specified via input X. The asynchronous write job can be started on a positive edge at input WR. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and can vary from job to job. During an active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

Output Q = 1 indicates that the parameter has been successfully written. If an error occurs during an access, this is signaled with QF = 1.

For error diagnostics, the error code ERC can be evaluated. ERC corresponds to the error code for parameter accesses according to PROVIdrive 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	Default	Value range	Attrib-utes
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	

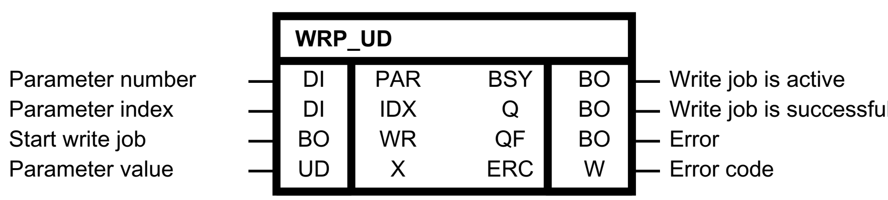
Configuration data

SIMOTION	-
SINAMICS	✓
Can be loaded online	No
Special characteristics	-

6.20 WRP_UD

Writes drive parameters (UNSIGNED DOUBLE INTEGER type)

Symbol



Brief description

WRP_UD (Write Parameter) enables the asynchronous writing of drive parameters of the UNSIGNED DOUBLE INTEGER type on the local drive object.

NOTICE
Do not use with Safety Integrated
The block for programming a drive parameter must not be used for safety reasons to change parameters of the SINAMICS Safety Integrated Functions. DCC is not considered suitable for safety applications in the context of functional safety (Safety Integrated).

Method of operation

The parameter number and the index of the parameter to be written must be specified 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 calculated. It is not possible to access parameters on several drive objects.

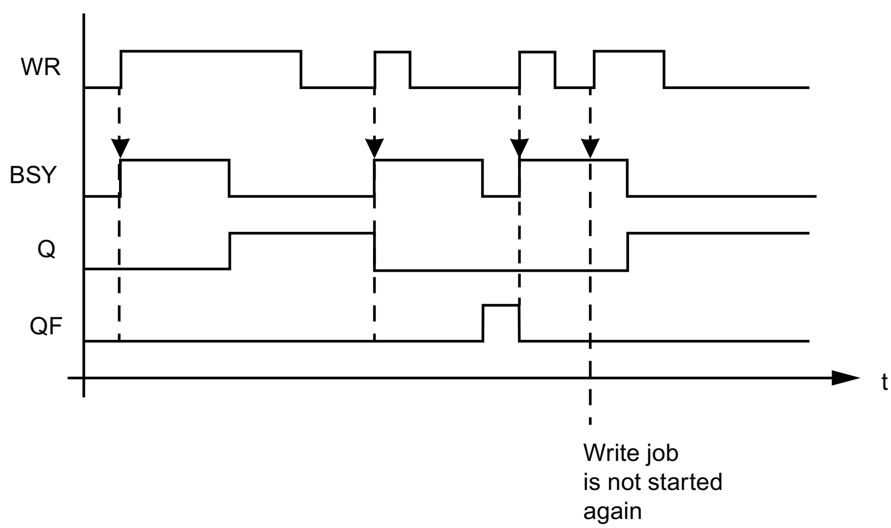
The parameter value is specified via input X. The asynchronous write job can be started on a positive edge at input WR. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and can vary from job to job. During an active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

Output Q = 1 indicates that the parameter has been successfully written. If an error occurs during an access, this is signaled with QF = 1.

For error diagnostics, the error code ERC can be evaluated. ERC corresponds to the error code for parameter accesses according to PROVIdrive 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	Default	Value range	Attrib-utes
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	

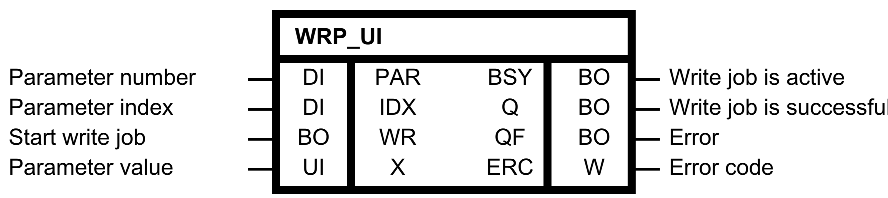
Configuration data

SIMOTION	-
SINAMICS	✓
Can be loaded on-line	No
Special character-istics	-

6.21 WRP_UI

Writes drive parameters (UNSIGNED INTEGER type)

Symbol



Brief description

WRP_UI (Write Parameter) enables the asynchronous writing of drive parameters of the UNSIGNED INTEGER type on the local drive object.

NOTICE
Do not use with Safety Integrated
The block for programming a drive parameter must not be used for safety reasons to change parameters of the SINAMICS Safety Integrated Functions. DCC is not considered suitable for safety applications in the context of functional safety (Safety Integrated).

Method of operation

The parameter number and the index of the parameter to be written must be specified 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 calculated. It is not possible to access parameters on several drive objects.

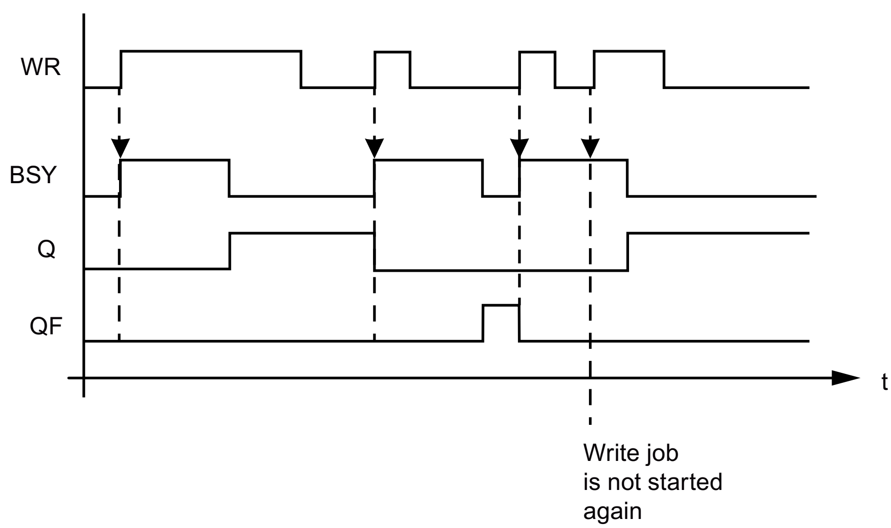
The parameter value is specified via input X. The asynchronous write job can be started on a positive edge at input WR. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and can vary from job to job. During an active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

Output Q = 1 indicates that the parameter has been successfully written. If an error occurs during an access, this is signaled with QF = 1.

For error diagnostics, the error code ERC can be evaluated. ERC corresponds to the error code for parameter accesses according to PROVIdrive 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	Default	Value range	Attrib-utes
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	

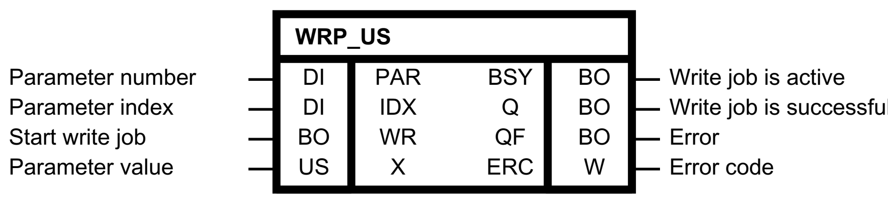
Configuration data

SIMOTION	-
SINAMICS	✓
Can be loaded on-line	No
Special character-istics	-

6.22 WRP_US

Writes drive parameters (UNSIGNED SHORT INTEGER type)

Symbol



Brief description

WRP_US (Write Parameter) enables the asynchronous writing of drive parameters of the UNSIGNED SHORT INTEGER type on the local drive object.

NOTICE
Do not use with Safety Integrated
The block for programming a drive parameter must not be used for safety reasons to change parameters of the SINAMICS Safety Integrated Functions. DCC is not considered suitable for safety applications in the context of functional safety (Safety Integrated).

Method of operation

The parameter number and the index of the parameter to be written must be specified 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 calculated. It is not possible to access parameters on several drive objects.

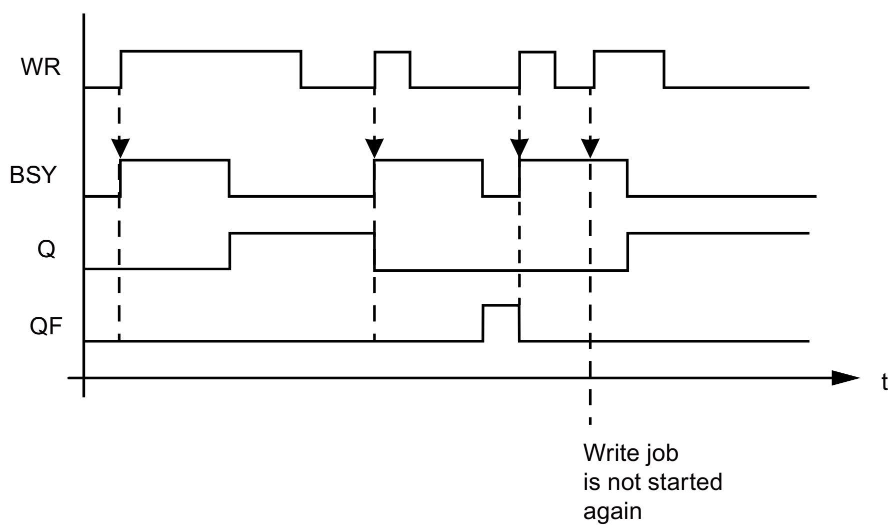
The parameter value is specified via input X. The asynchronous write job can be started on a positive edge at input WR. As long as the job is active, the BSY flag is set. The number of cycles for a parameter access is dependent on the system utilization and can vary from job to job. During an active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

Output Q = 1 indicates that the parameter has been successfully written. If an error occurs during an access, this is signaled with QF = 1.

For error diagnostics, the error code ERC can be evaluated. ERC corresponds to the error code for parameter accesses according to PROVIdrive 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	Default	Value range	Attrib-utes
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	

Configuration data

SIMOTION	-
SINAMICS	✓
Can be loaded online	No
Special characteristics	-

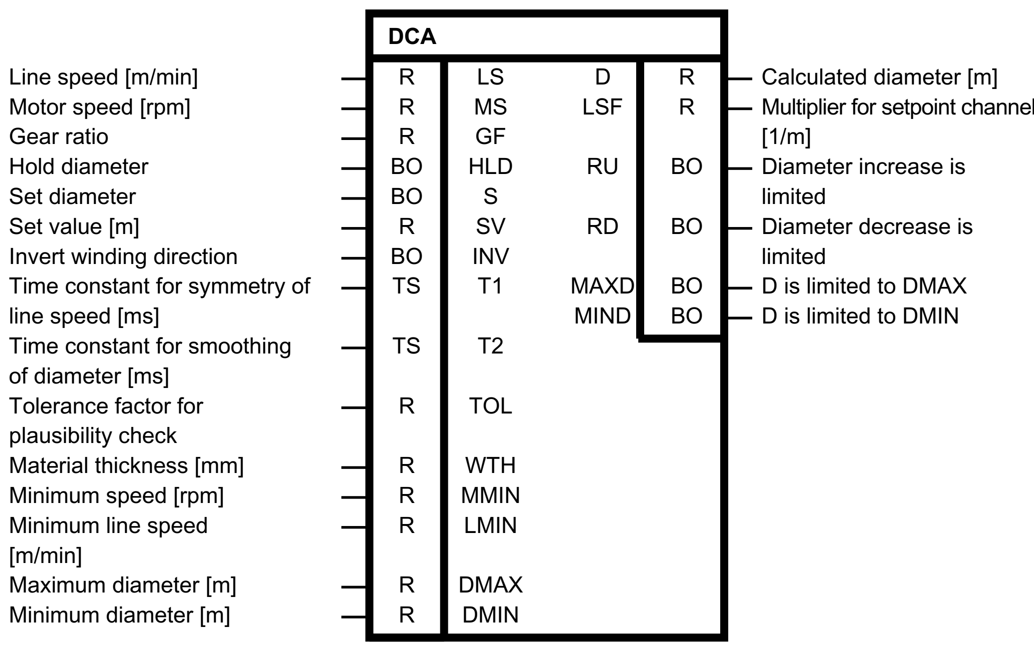
Technology

7

7.1 DCA

Diameter calculator

Symbol



Brief 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.

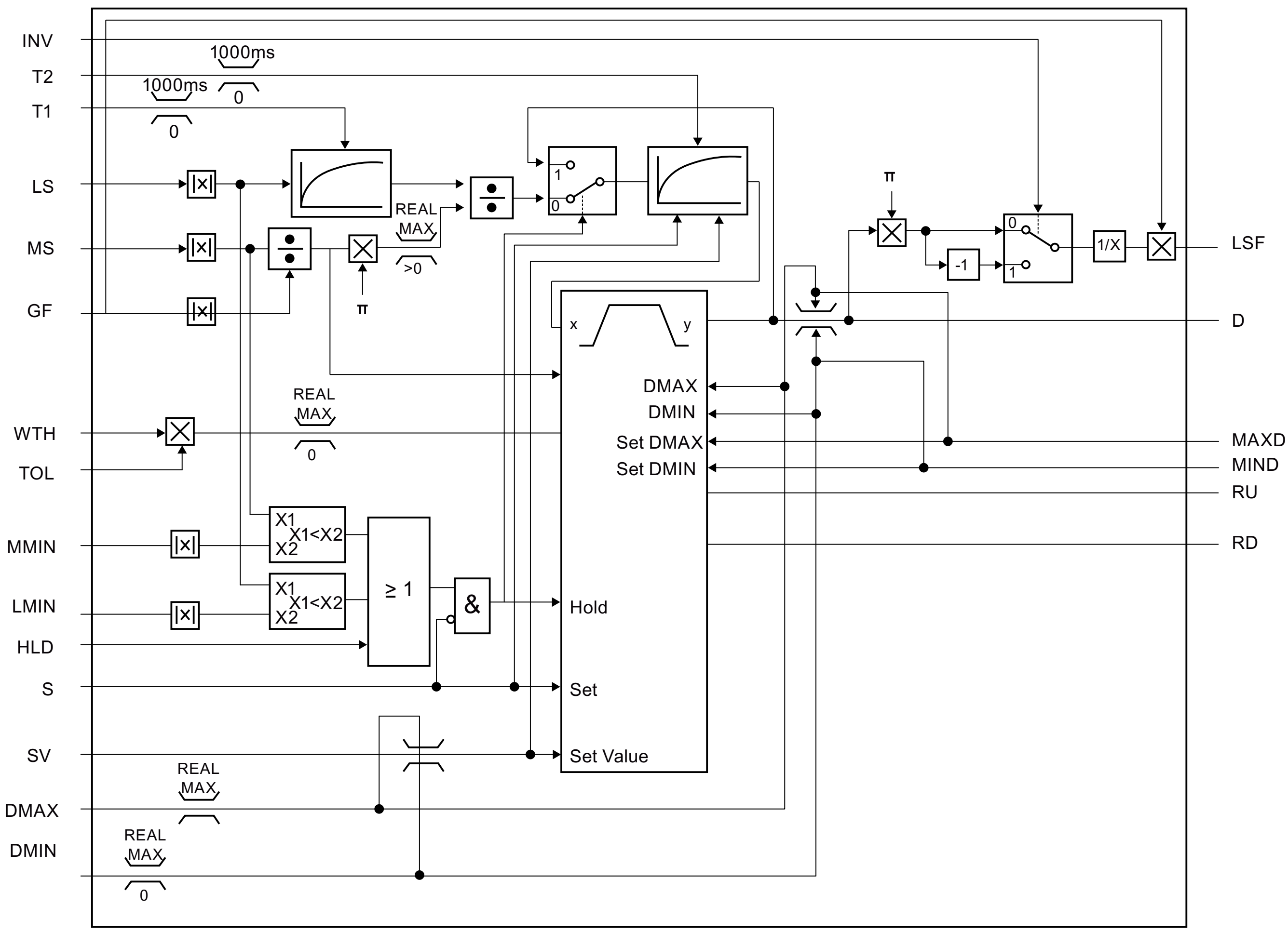
Note

DCC SINAMICS

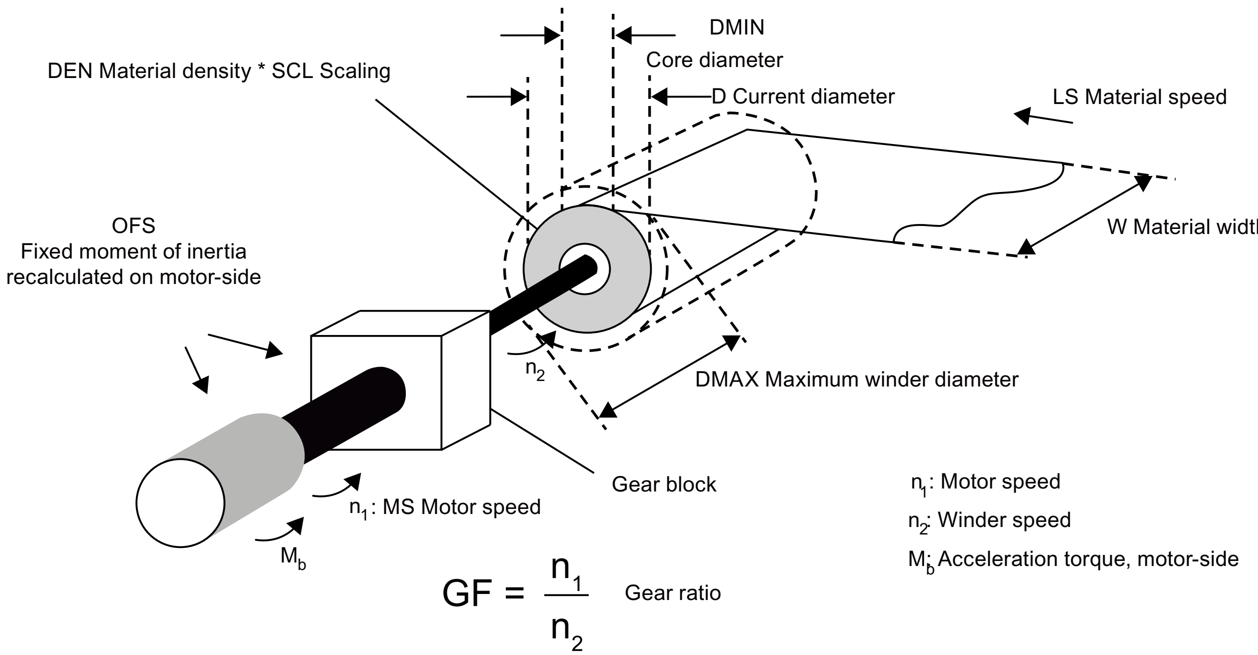
If you use this block in the following execution groups, you must assign parameter p2048 the value of the isochronous master clock:

- Receive AFTER IF1 PROFIdrive PZD
- Send BEFORE IF1 PROFIdrive PZD
- Receive AFTER IF1 PROFIdrive flexible PZD
- Receive AFTER IF2 PZD
- Send BEFORE IF2 PZD
- Receive AFTER IF2 flexible PZD

Block diagram



Method of operation



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 indicated at input MS. With deceleration time T1, the path velocity can be decelerated relative to the motor speed.

The current diameter is then calculated using the following formula:

$$\text{Diameter} = \frac{\text{Line speed} \cdot \text{Gear ratio}}{\text{Motor speed} \cdot \pi}$$

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 time constant T1 or T2 = 0, the input value of the smoothing is written directly to the output. The diameter is only calculated if the path velocity LS or motor speed MS is greater than the threshold value LMIN or MMIN, respectively. Otherwise, the last calculated diameter value is held. In this case, smoothing T2 switches over to the fed-back diameter D. It is also possible to trigger holding of diameter D directly by setting 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. When S = 0, the diameter calculation and smoothing T2 is enabled again. Setting the diameter has precedence over 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 limiting is no longer in effect, the corresponding output is reset to zero. When Hold = 1 or Set = 1, both outputs are reset. When the diameter is set, the ramp-function generator has no effect. The plausibility check is a downstream limiter. If the current diameter is limited to DMAX, output MAXD = 1 is set. If the current diameter is limited to DMIN, this is signaled at output MIND. If limiting is active, the ramp-function generator is corrected with the active limit value in order to avoid anti-windup. In this case, the following applies to the next ramp-function generator cycle:

D_{n-1} = $DMAX_{n-1}$ if diameter is limited to DMAX

D_{n-1} = $DMIN_{n-1}$ if diameter is limited to DMIN

Output LSF cyclically supplies a multiplication factor for the setpoint channel in order to calculate the speed setpoint of the motor from the current path velocity. If the INV input is set to the value 1, the winding direction is inverted.

Note

Note for using "Indirect tension control"

In the case of a web break, HLD should be set in order to keep the diameter value. Otherwise, block DCA again supplies an increasing/decreasing (unwinder/winder) diameter as a result of the diameter calculation based on the actual web velocity and the motor speed. As a consequence, the winder would accelerate.

Block connections

Block connection	Description	Default	Value range	Attrib-utes
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	

Block connection	Description	Default	Value range	Attrib-utes
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 [1/m]	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	

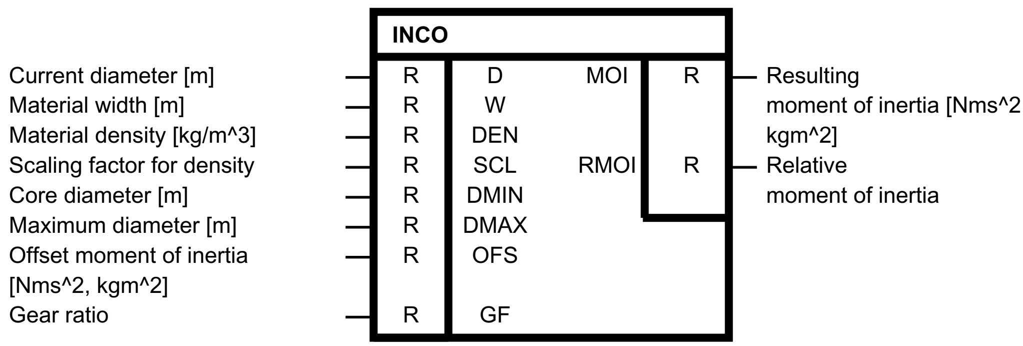
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

7.2 INCO

Axial winder moment of inertia

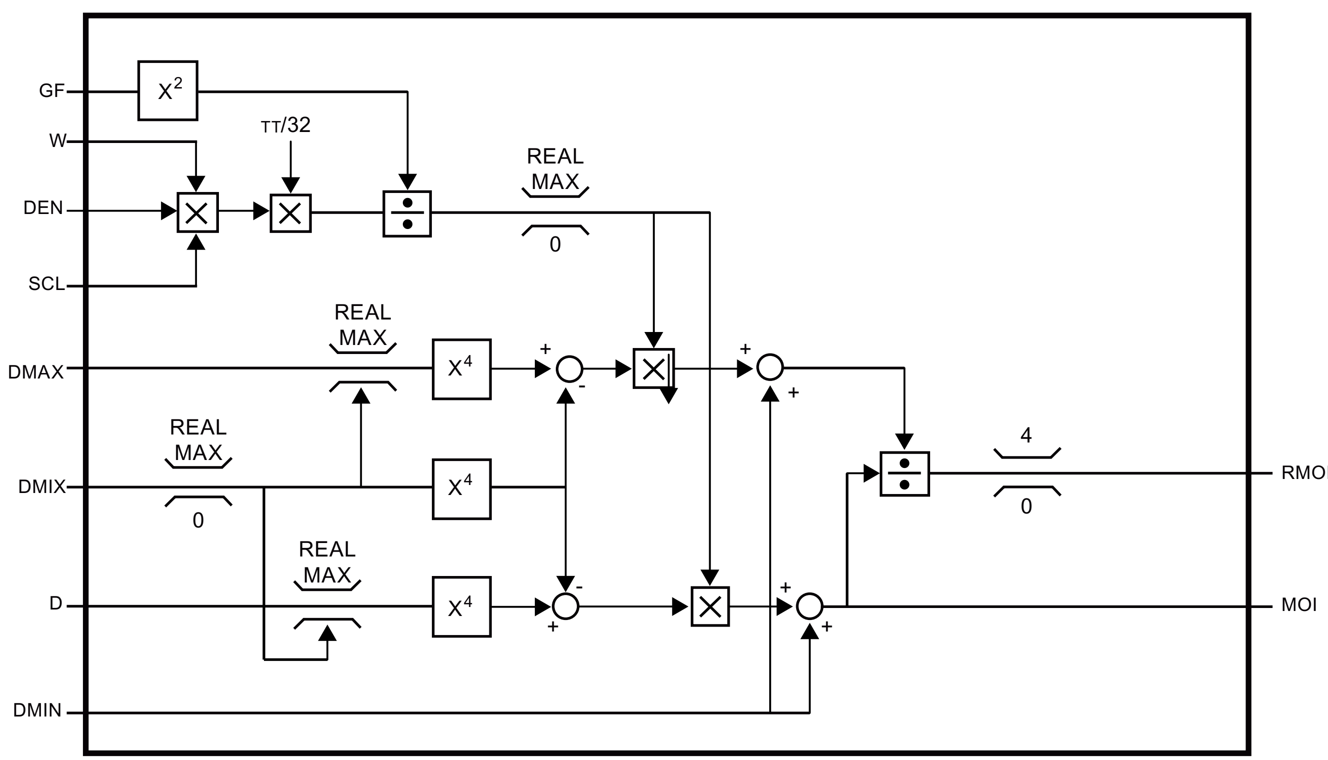
Symbol



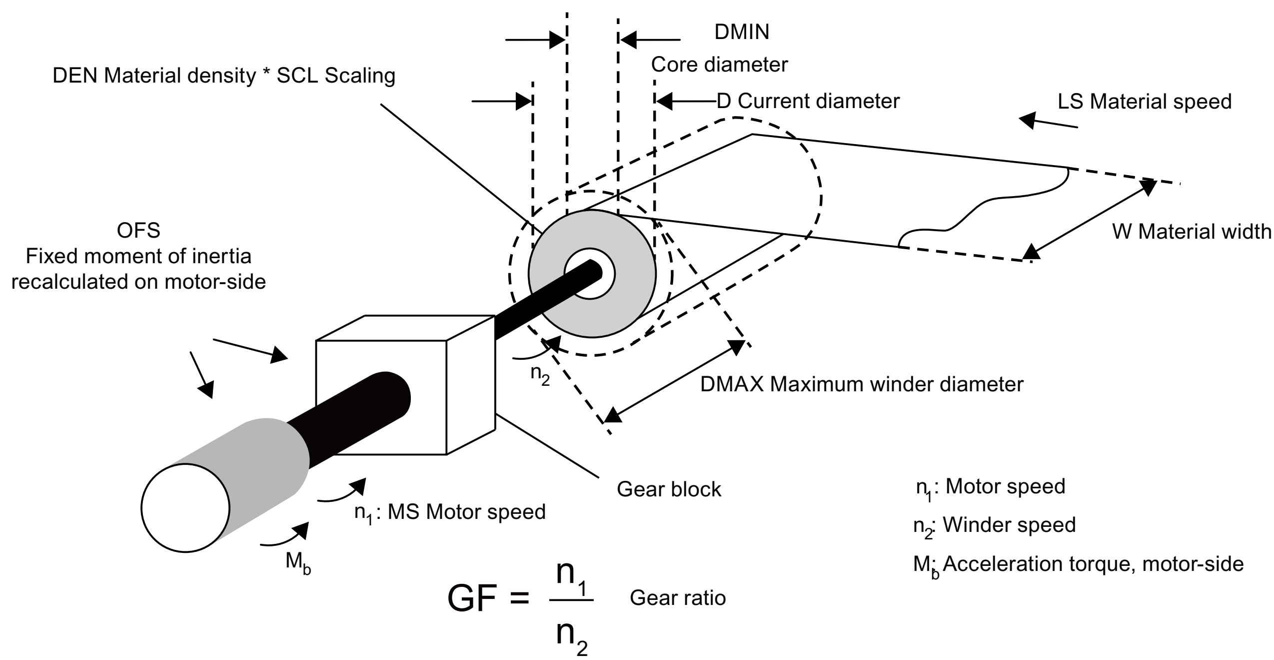
Brief description

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



Method of operation



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 winder can be specified via DEN, and a correction factor for the density can be specified via SCL. Input variable DMIN [m] is used to specify the diameter of the winding core or the minimum diameter of the wound material. In order to calculate the relative moment of inertia RMOI for an adaptation Kp 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) relative to the motor side can be specified via input OFS [Nms2, kgm2]. The transmission ratio is specified at input GF. The current moment of inertia of the entire winder layout relative to the motor side is output at output MOI.

Block connections

Block connection	Description	Default	Value range	Attrib-utes
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	

Block connection	Description	Default	Value range	Attrib-utes
MOI	Resulting moment of inertia [Nms^2, kgm^2]	0.0	0..REAL MAX	
RMOI	Relative moment of inertia	0.0	0..REAL MAX	

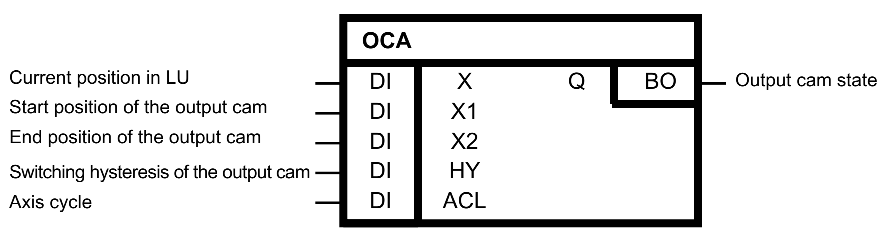
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

7.3 OCA

Software cam controller

Symbol

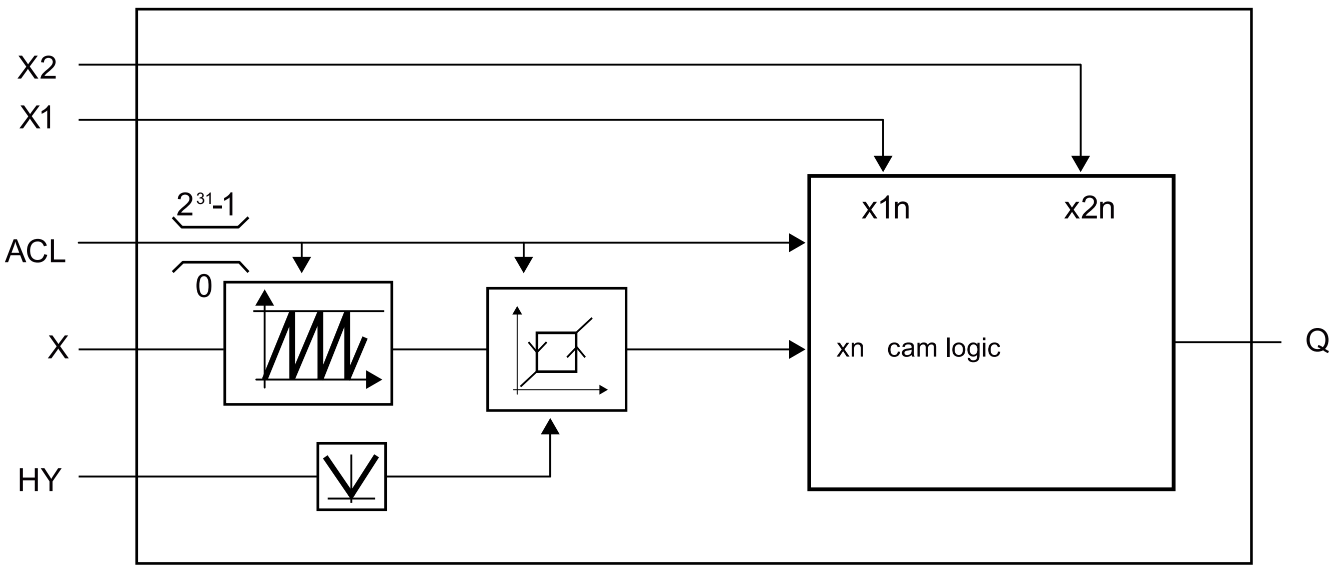


Brief description

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



Method of operation

The switch-on position of the position-based cam in the positive direction and the switch-off position in the negative direction is specified via input X1 [LU]. X2 [LU] specifies the switch-off position in the positive direction or the switch-on position in the negative direction. In order to be able to drive the cam controller with modulo axes, the axis cycle can be specified at input ACL. If ACL = 0, there is no internal modulo correction. A hysteresis band for input X can be set via HY. This means that switching operations do not occur when actual value-related output cams are at a standstill.

The hysteresis is used to avoid unwanted switch-on and switch-off operations of the output cam during actual value noise. After a switching operation, switching is only possible again for a direction reversal when the hysteresis range is exited.

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	Default	Value range	Attrib-utes
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	

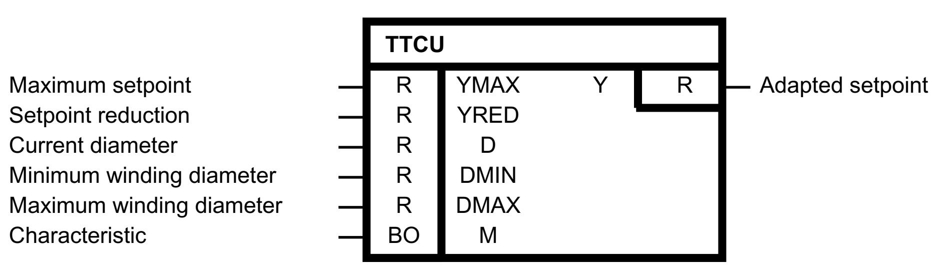
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

7.4 TTCU

Winding harshness characteristic

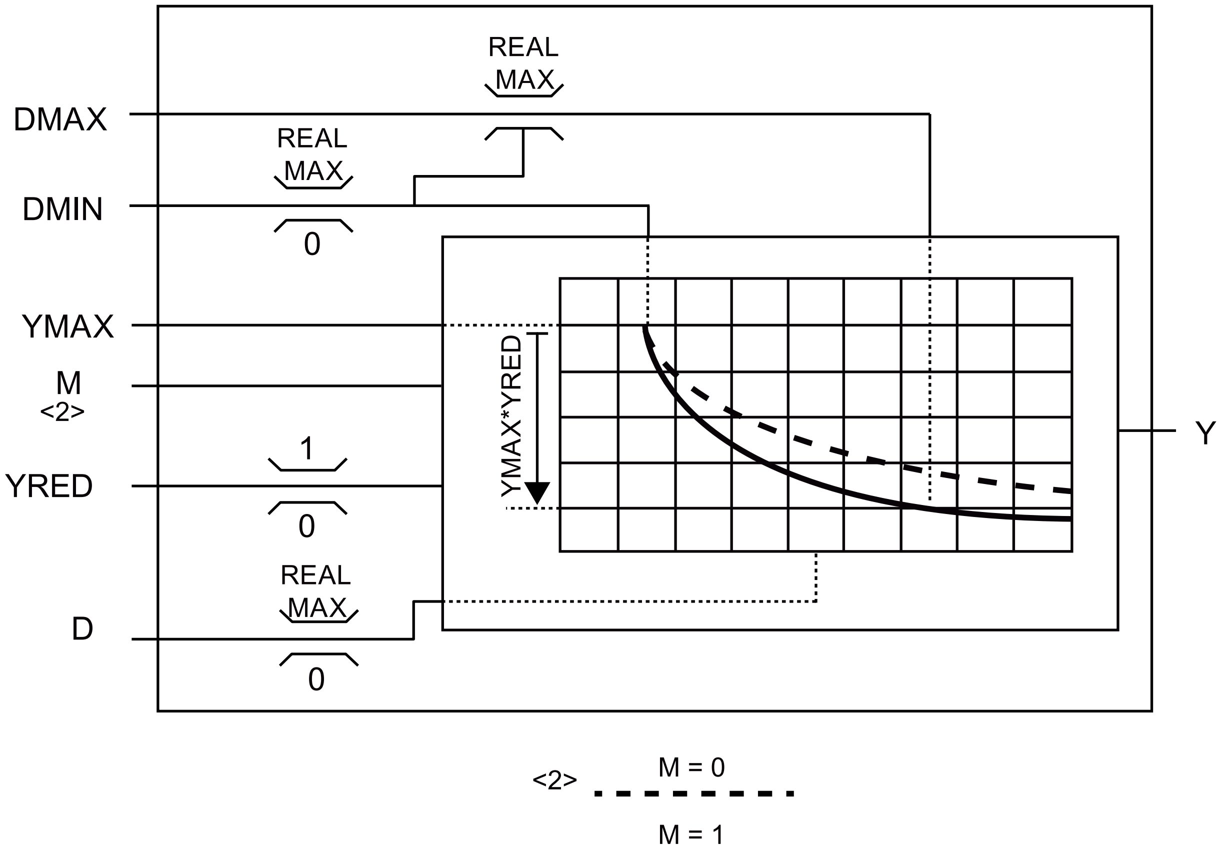
Symbol



Brief description

Adaptation of a setpoint according to the specified characteristic. The block used for winder applications to determine the tension setpoint depending on the current winder diameter.

Block diagram



Method of operation

The reduction of the characteristic starts when D > DMIN is true. Input variable YRED specifies the degree of reduction relative to input variable YMAX. Input M can be used to preselect a characteristic that defines the reduction behavior of the output variable as the input variable increases. 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. If M =1 has been preselected, input variable DMAX can be used to specify at 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	Default	Value range	Attrib-utes
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	

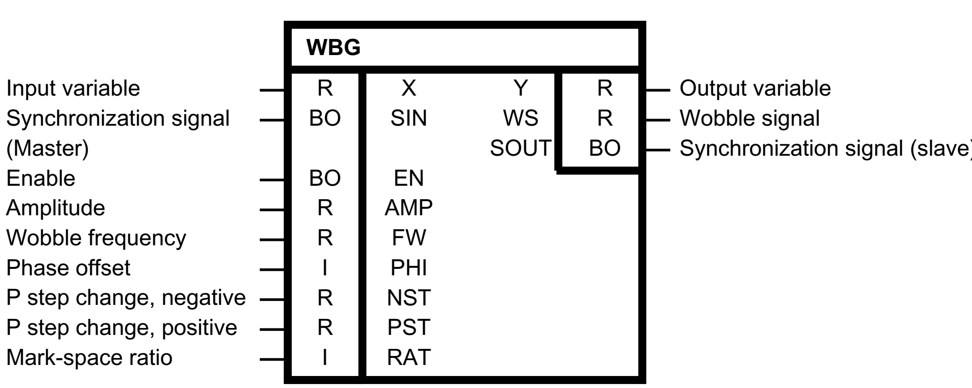
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

7.5 WBG

Wobble generator

Symbol



Brief 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.
- Enable input.

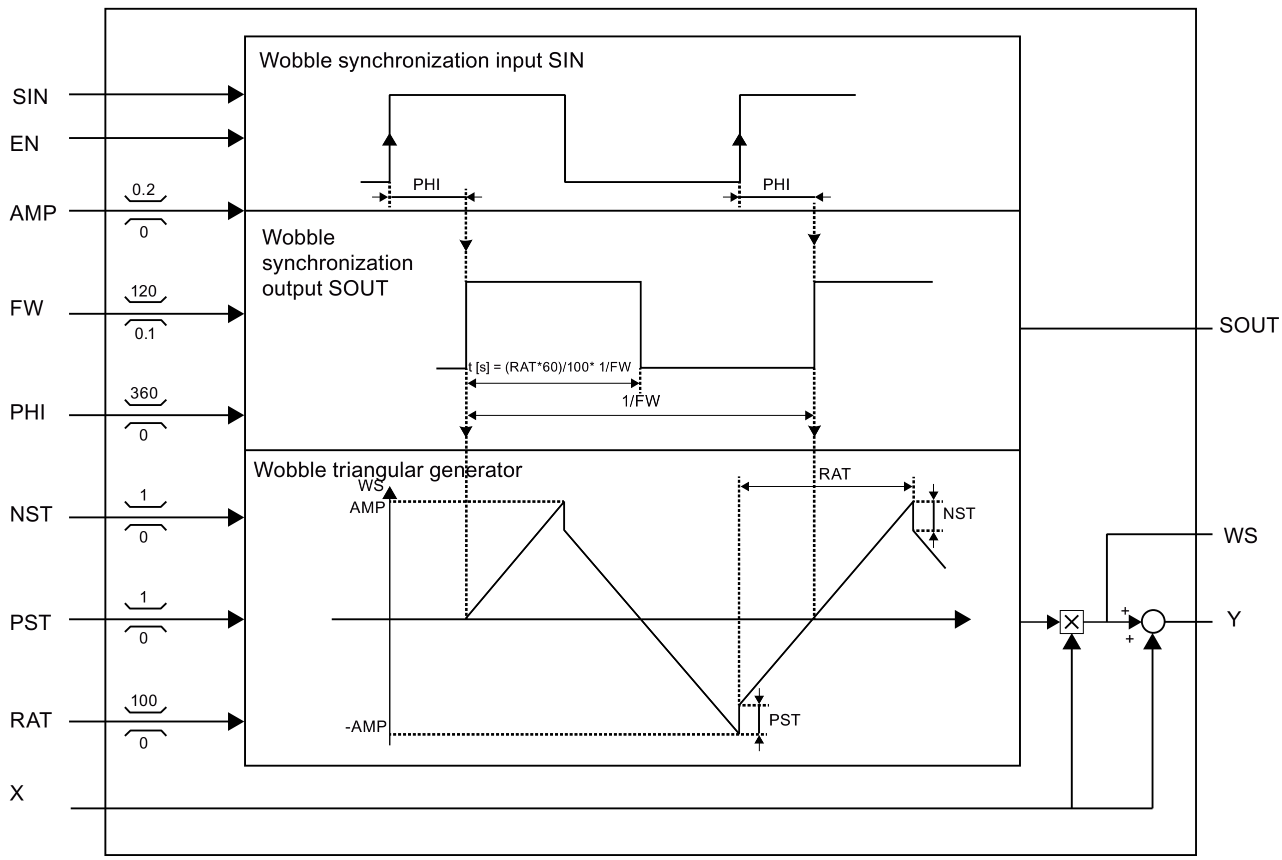
Note

DCC SINAMICS

If you use this block in the following execution groups, you must assign parameter p2048 the value of the isochronous master clock:

- Receive AFTER IF1 PROFIdrive PZD
- Send BEFORE IF1 PROFIdrive PZD
- Receive AFTER IF1 PROFIdrive flexible PZD
- Receive AFTER IF2 PZD
- Send BEFORE IF2 PZD
- Receive AFTER IF2 flexible PZD

Block diagram



Method of operation

The wobble generator is enabled with EN = 1. This triggers the output of wobble signal WS and synchronization signal SOUT. The signal generation always starts with a positive zero crossover or with a positive edge of synchronization output SOUT. If EN is reset again, wobble generation continues up to the next zero crossover of WS. Only then is the generator again inhibited and SOUT = 0 Input PHI (0-360°) enables a phase shift between the positive edge of the synchronization input SIN to be set along with 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 the generation of the preceding signal period is still running at a new start time, this generation is canceled. Special case PHI = 360 enables free-running wobble generation to be activated. The signal generation runs periodically and is decoupled from synchronization input SIN. The wobble signal is injected into 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

Input	Value range	Description
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	Default	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	

Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

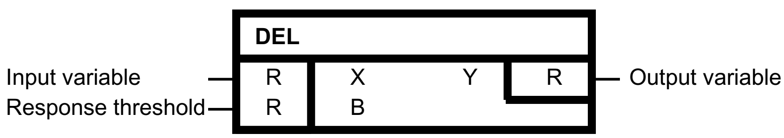
Closed-loop control

8

8.1 DEL

Dead zone element

Symbol



Brief description

- Adjustable dead band
- Set zero-point symmetric value range to zero

Method of operation

- 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.

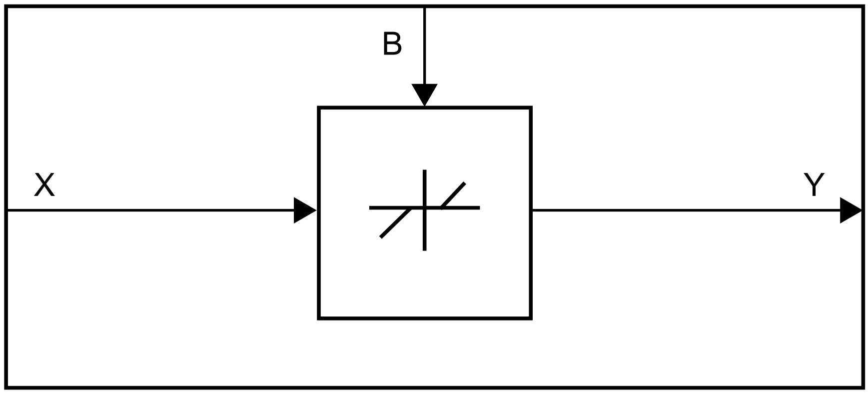
Algorithm:

with boundary condition B >=0

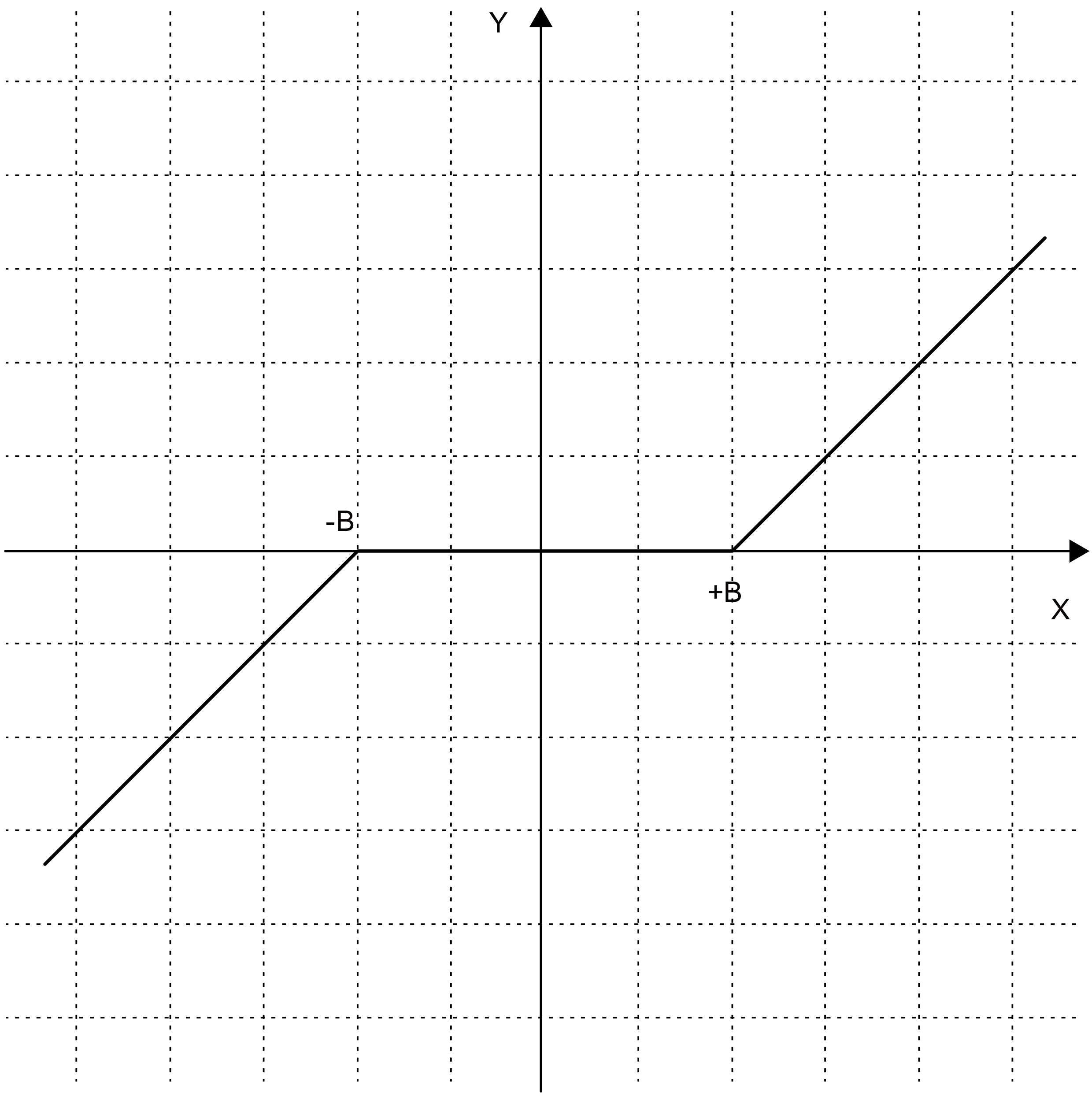
For B < 0, the following applies for all X: Y = X.

$$Y = \begin{cases} X + B \text{ for } X <= -B \\ 0 \text{ for } -B < X < B \\ X - B \text{ for } X >= B \end{cases}$$

Block diagram



XY diagram



Block connections

Block connection	Description	Default	Value range	Attrib-utes
X	Input variable	0.0	REAL	
B	Response threshold	0.0	REAL	
Y	Output variable	0.0	REAL	

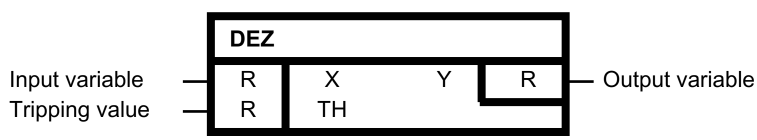
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

8.2 DEZ

Dead zone element

Symbol



Brief description

- Adjustable dead zone
- Set zero-point symmetric value range to zero

Method of operation

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.

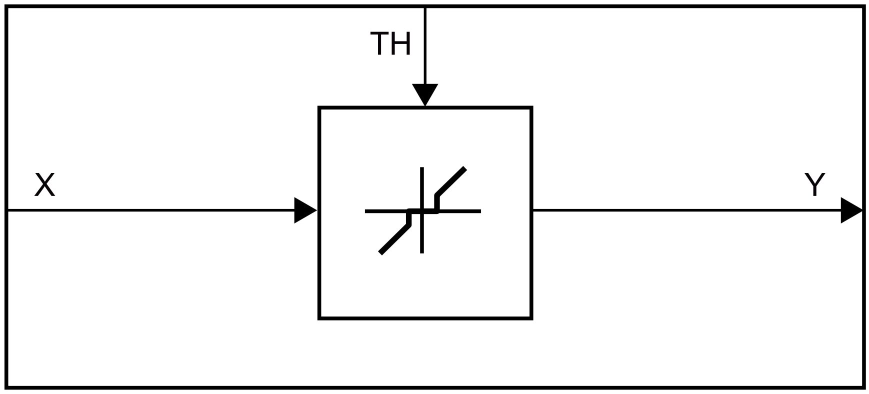
Algorithm:

$$Y = \begin{cases} X \text{ for } X <= -TH \\ 0 \text{ for } -TH < X < TH \\ X \text{ for } X >= TH \end{cases}$$

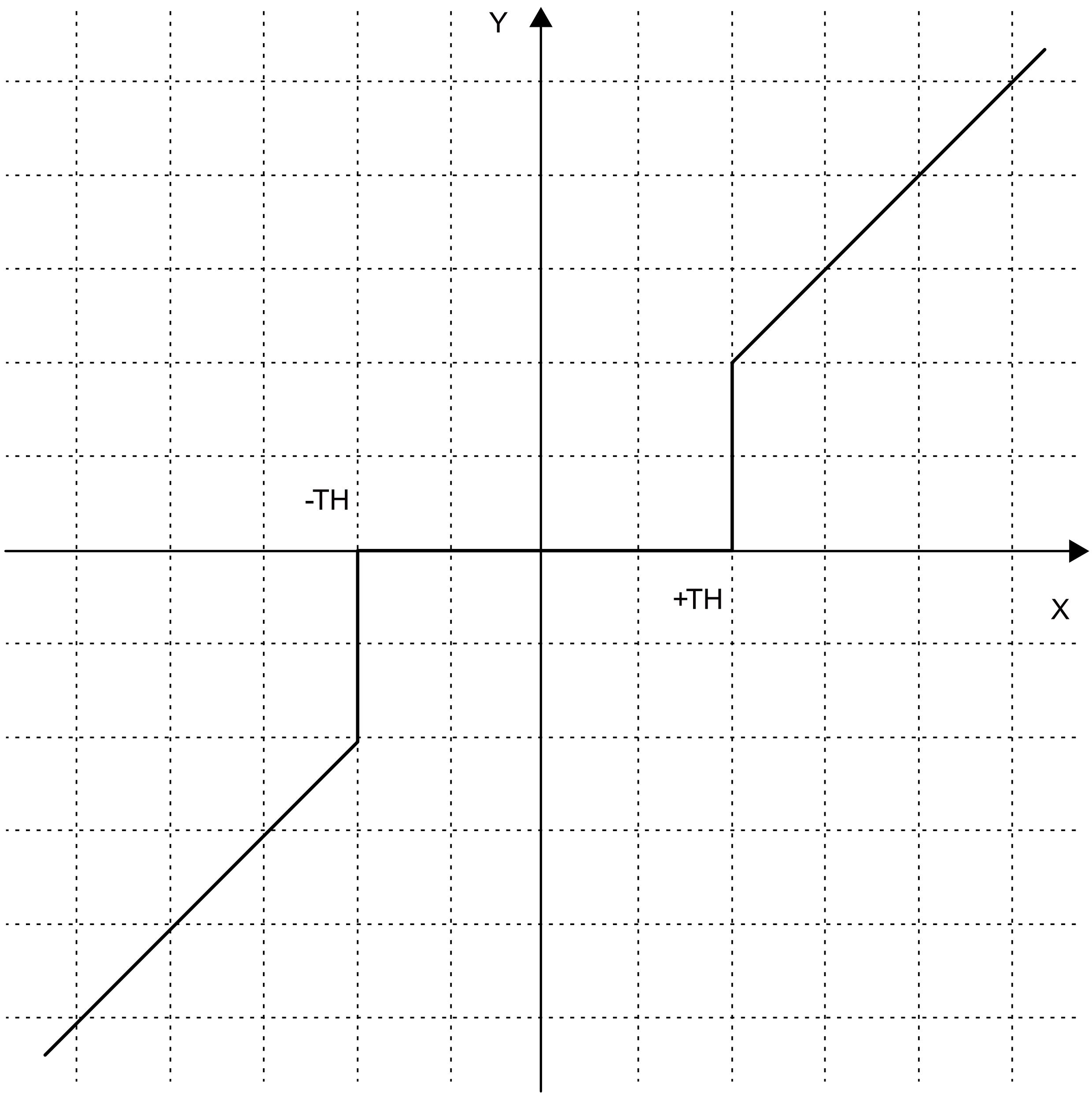
with boundary condition TH >=0

For TH< 0, the following applies for all X: Y = X.

Block diagram



XY diagram



Block connections

Block connection	Description	Default	Value range	Attrib-utes
X	Input variable	0.0	REAL	
TH	Tripping value	0.0	REAL	
Y	Output variable	0.0	REAL	

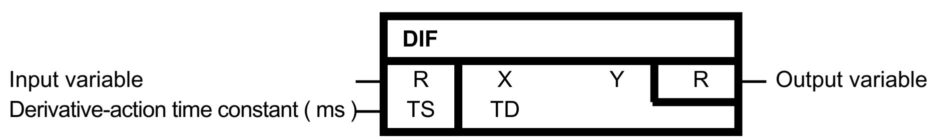
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

8.3 DIF

Derivative action element

Symbol



Brief description

Block with derivative-action response

Note

DCC SINAMICS

If you use this block in the following execution groups, you must assign parameter p2048 the value of the isochronous master clock:

- Receive AFTER IF1 PROFIdrive PZD
- Send BEFORE IF1 PROFIdrive PZD
- Receive AFTER IF1 PROFIdrive flexible PZD
- Receive AFTER IF2 PZD
- Send BEFORE IF2 PZD
- Receive AFTER IF2 flexible PZD

Method of operation

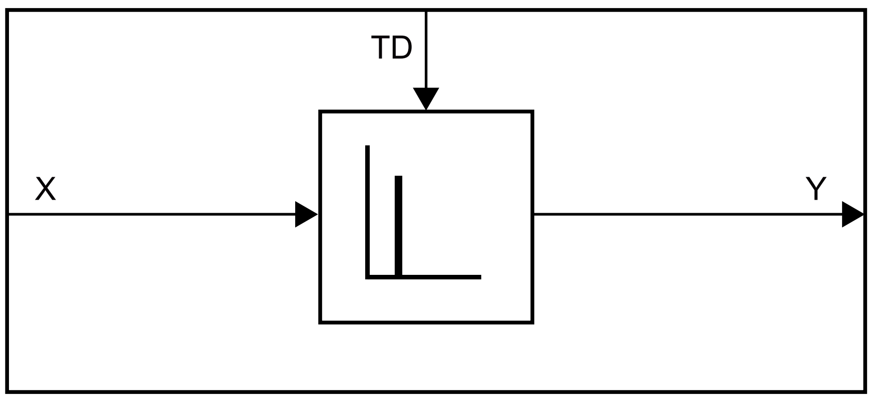
- Output variable Y is proportional to the change velocity of input variable X, multiplied by the derivative-action time constant TD.
- Discrete values are calculated according to the algorithm:

Algorithm:

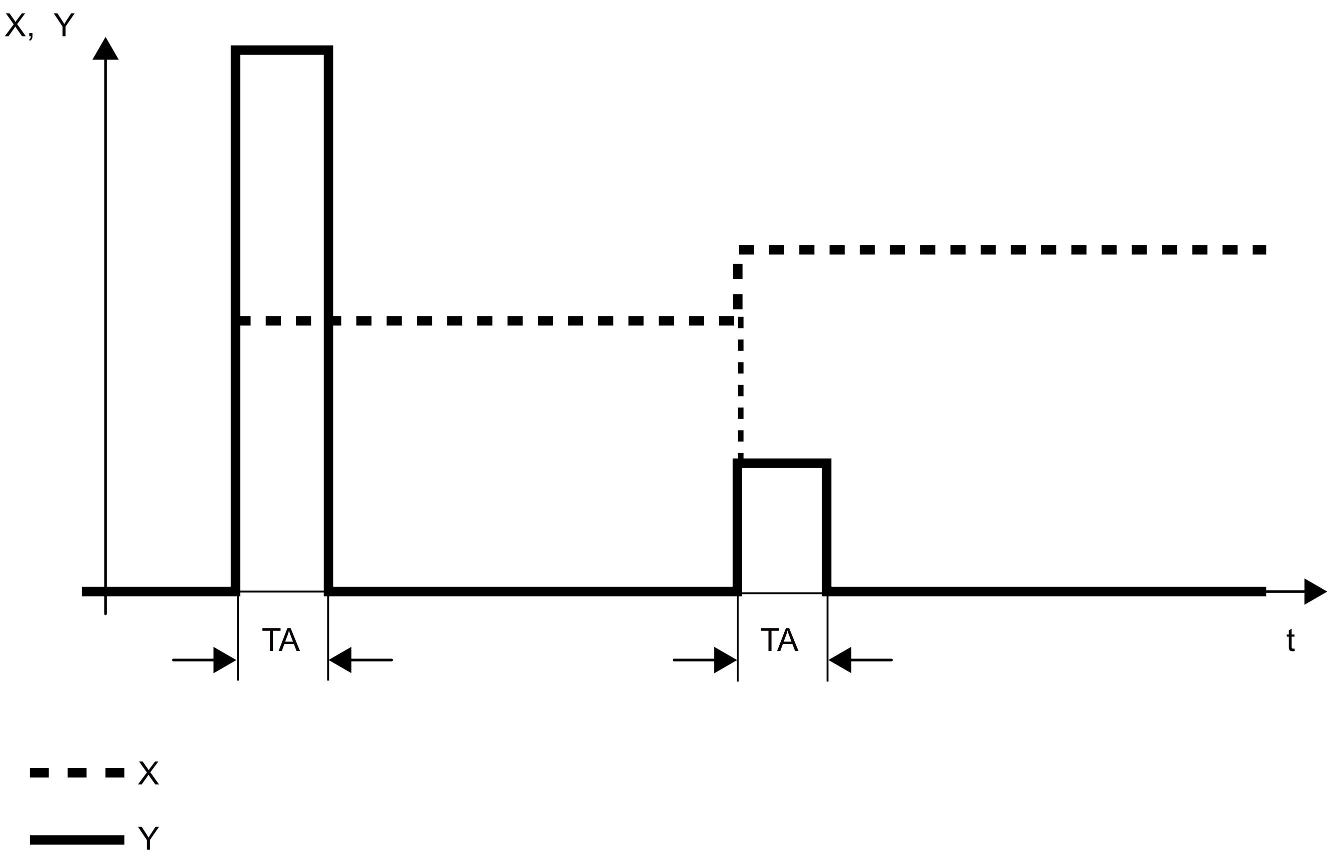
$$Y_n = (X_n - X_{n-1}) \cdot \frac{TD}{TA}$$

Y_n	Value of Y in scan interval n
X_n	Value of X in scan interval n
X_{n-1}	Value of X in scan interval n-1

Block diagram



XY diagram



Block connections

Block connection	Description	Default	Value range	Attrib-utes
X	Input variable	0.0	REAL	
TD	Derivative-action time constant (ms)	0	SDTIME	
Y	Output variable	0.0	REAL	

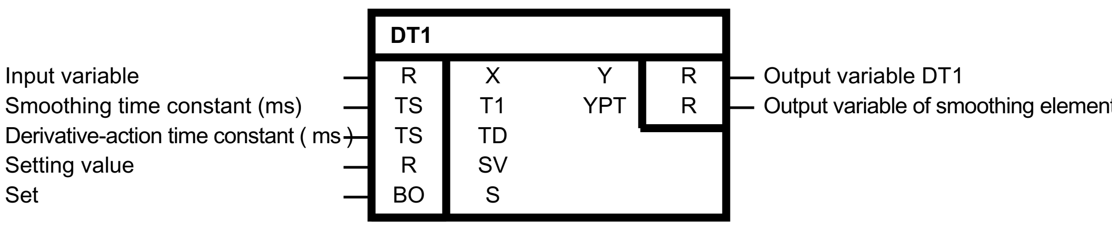
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

8.4 DT1

Smoothing element

Symbol



Brief description

Block with derivative-action response and smoothing. The DT1 output can be set.

Note

DCC SINAMICS

If you use this block in the following execution groups, you must assign parameter p2048 the value of the isochronous master clock:

- Receive AFTER IF1 PROFIdrive PZD
- Send BEFORE IF1 PROFIdrive PZD
- Receive AFTER IF1 PROFIdrive flexible PZD
- Receive AFTER IF2 PZD
- Send BEFORE IF2 PZD
- Receive AFTER IF2 flexible PZD

Method of operation

Setting function not active (S = 0)

Input variable X (dynamically delayed by smoothing time constant T1), is given to a derivative-action element and 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 fell to 37% of X-TD/T1 after smoothing and differentiation. If T1/TA is sufficiently large (T1/TA>10), the transfer function corresponds to the characteristic curve of

$Y(t) = X \cdot (TD/T1) \cdot e^{-t/T1}$

with $t = n \cdot TA$

Algorithm:

$$Y_n = \frac{TD}{T1} \cdot (X_n - YPT_{n-1})$$

$$YPT_n = YPT_{n-1} + \frac{TA}{T1} \cdot (X_n - YPT_{n-1})$$

YPT_n	Value of YPT in scan interval n
Y_n	Value of Y in scan interval n
X_n	Value of X in scan interval n
YPT_{n-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 that TD/TA is, the larger 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:

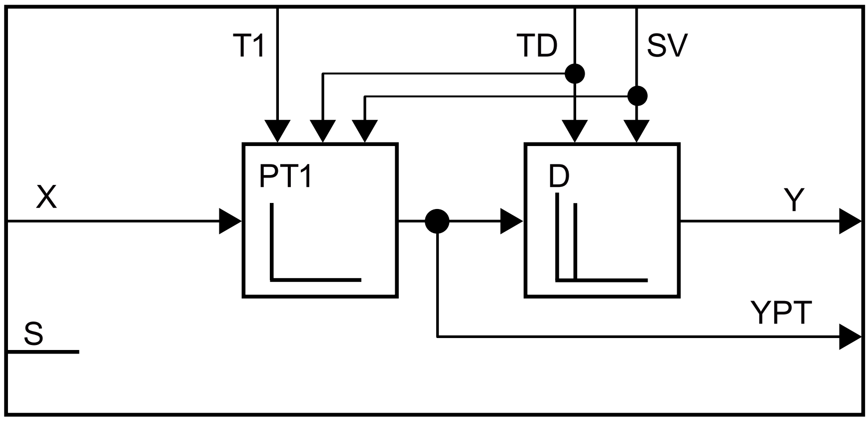
$$YPT_n = X_n - \frac{T1}{TD} \cdot SV_n$$

for TD ≠ 0

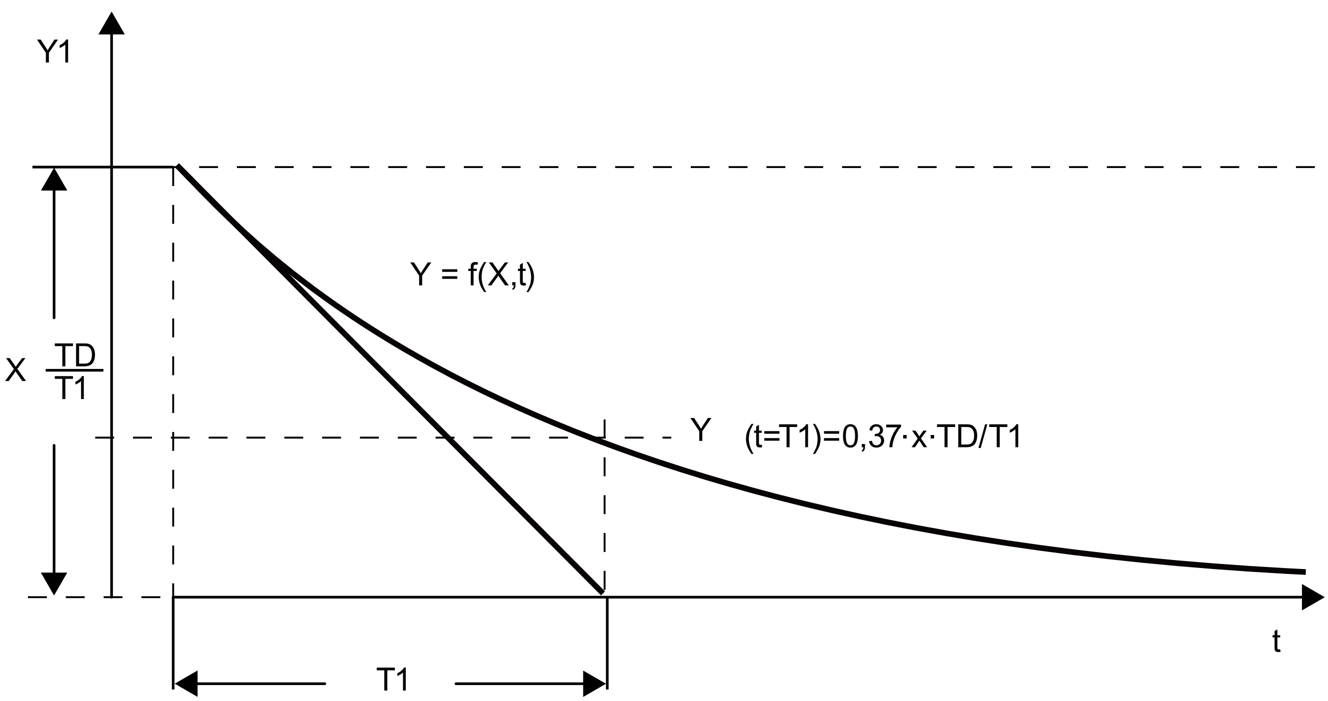
The internal limitations for T1 and TD apply in this case. When TD=0, the output variables remain unchanged, as long as S=1.

⚠ CAUTION
Overcontrol   Overcontrol is possible with the setting function both active as well as inactive!

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	Default	Value range	Attrib-utes
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	

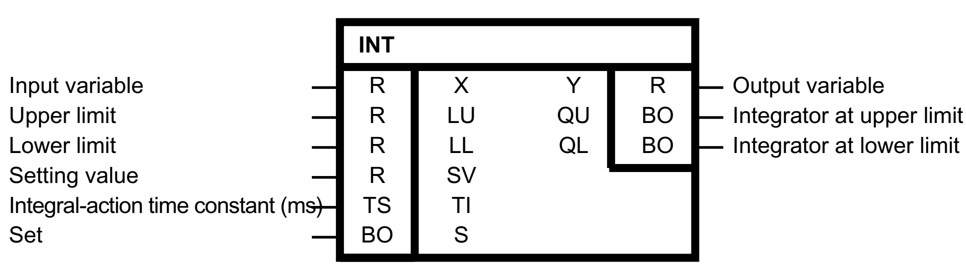
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

8.5 INT

Integrator

Symbol



Brief 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 for LL

Note

DCC SINAMICS

If you use this block in the following execution groups, you must assign parameter p2048 the value of the isochronous master clock:

- Receive AFTER IF1 PROFIdrive PZD
- Send BEFORE IF1 PROFIdrive PZD
- Receive AFTER IF1 PROFIdrive flexible PZD
- Receive AFTER IF2 PZD
- Send BEFORE IF2 PZD
- Receive AFTER IF2 flexible PZD

Method of operation

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:

$$Y_n = Y_{n-1} + \frac{TA}{TI} \cdot X_n$$

Y_n	Value of Y in scan interval n
Y_{n-1}	Value of Y in scan interval n-1
X_n	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

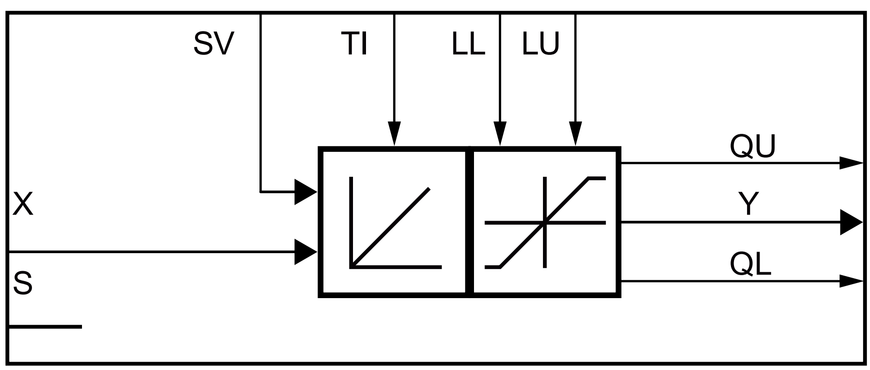
S is switched to 1. S is then set to 0, and the integrator starts from SV in the direction specified by the polarity of input variable X.

Note

You must ensure that the sampling time is sufficiently short with reference to the integrating time constant TI.

TI is limited internally: TI >=TA. When limiting is active, the amplitude change at Y depends on the cycle time.

Block diagram



Truth table(s)

S	Condition	Y	QU	QL	Operating 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	Condition	Y	QU	QL	Operating mode
(any)	LL >= LU	LU	1	1	INT at upper limit

Block connections

Block connection	Description	Default	Value range	Attrib-utes
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	

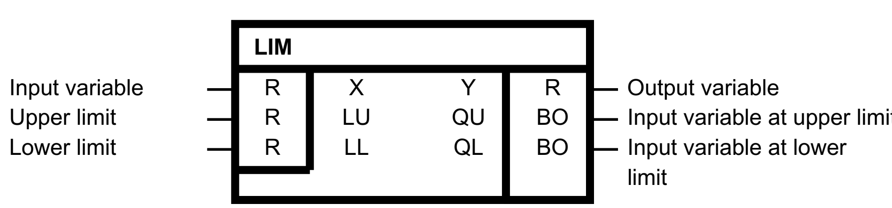
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

8.6 LIM

Limiter (REAL type)

Symbol



Brief description

- Block for the limitation
- Adjustable upper and lower limit
- Indication when set limits are reached

Method of operation

This 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.

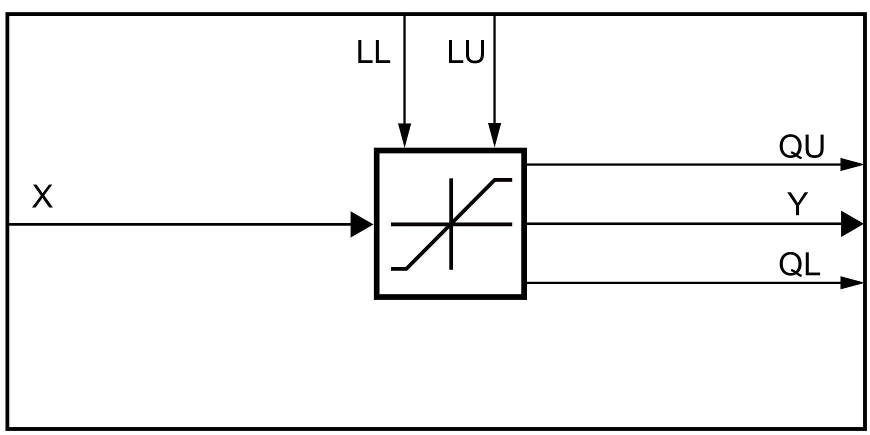
If the lower limit is greater than or equal to the upper limit, then output Y is set to the upper limit LU.

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 the boundary condition: LL<LU

Block diagram



Truth table(s)

Condition	Y	QU	QL	Operating 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

Condition	Y	QU	QL	Operating mode
LL >= LU	LU	1	1	Input variable at upper limit

Block connections

Block connection	Description	Default	Value range	Attrib-utes
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	

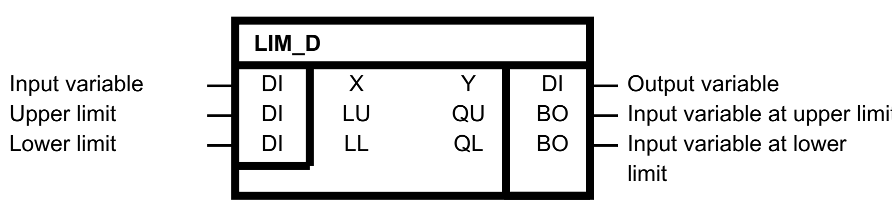
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

8.7 LIM_D

Limiter (DOUBLE INTEGER type)

Symbol



Brief description

- Block for the limitation of the DOUBLE INTEGER type
- Adjustable upper and lower limit
- Indication when set limits are reached

Method of operation

This 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.

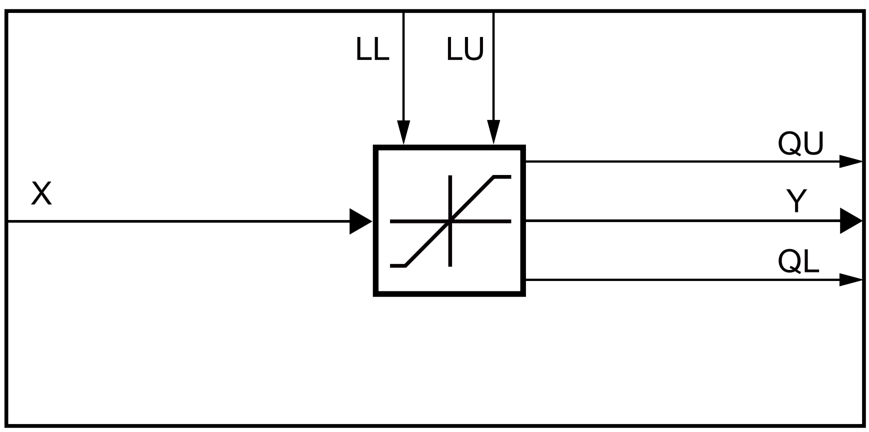
If the lower limit is greater than or equal to the upper limit, then output Y is set to the upper limit LU.

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 the boundary condition: LL<LU

Block diagram



Truth table(s)

Condition	Y	QU	QL	Operating 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

Condition	Y	QU	QL	Operating mode
LL >= LU	LU	1	1	Input variable at upper limit

Block connections

Block connection	Description	Default	Value range	Attrib-utes
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	

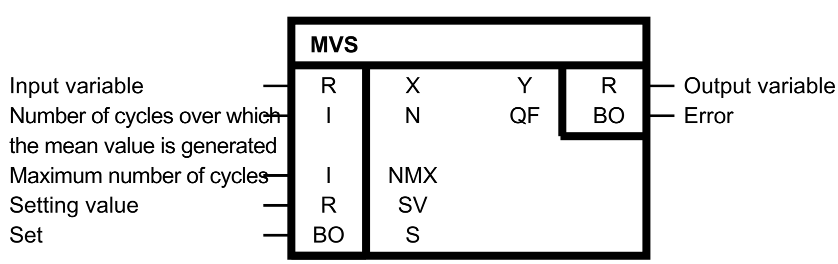
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

8.8 MVS

Sliding-type mean value generator

Symbol



Brief description

The block calculates a sliding-type mean value via the input variable X.

Method of operation

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

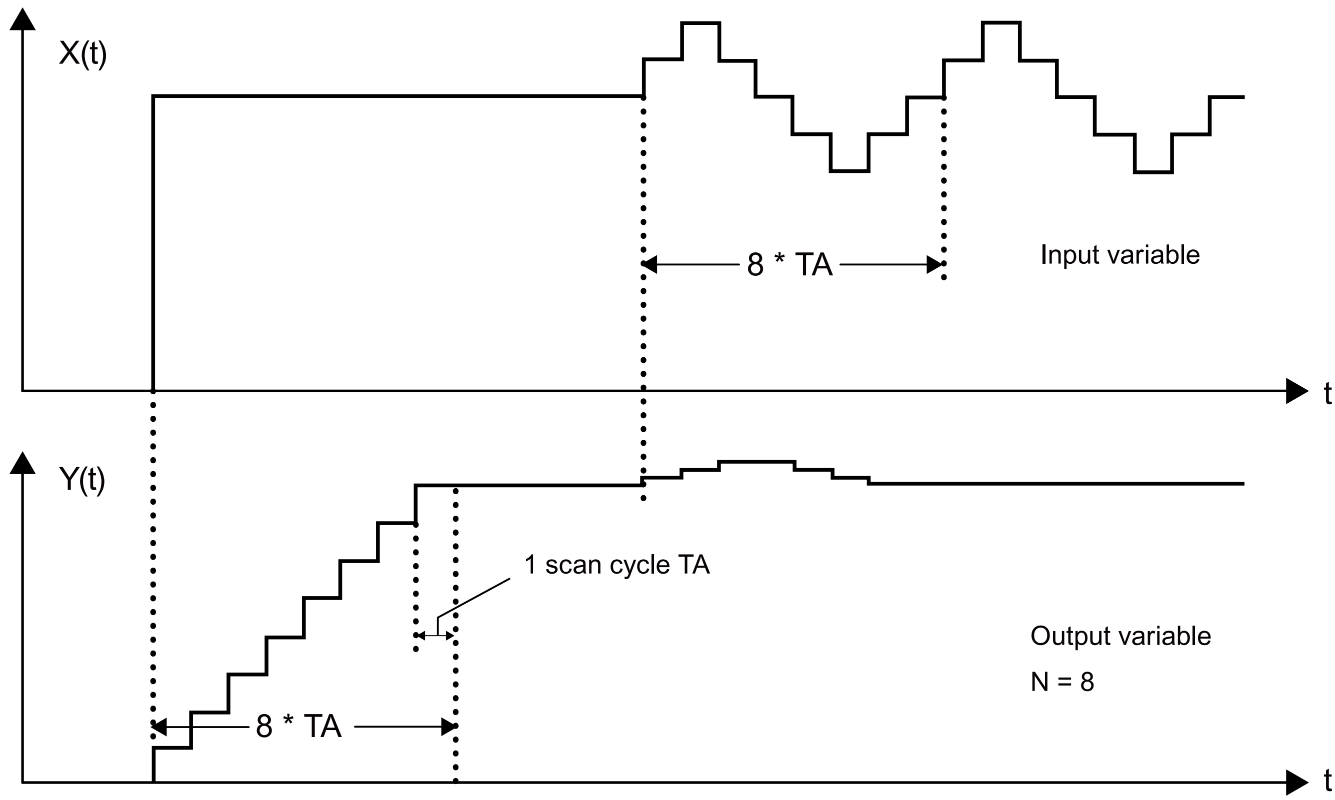
At the initialization, NMX is used to specify the maximum size of the mean value buffer for the X values. For this reason, NMX should be set to the maximum value of N required under operating conditions. The value of NMX is limited to the range of values between 1 and 1,000. If there is not enough memory 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.

Application areas

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	Default	Value range	Attrib-utes
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	

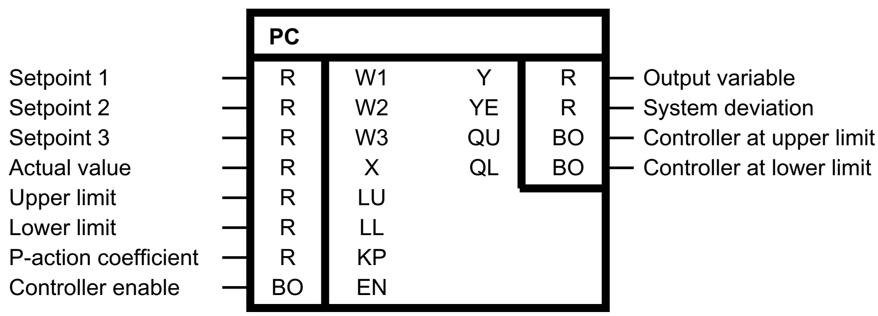
Configuration data

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

8.9 PC

P controller

Symbol



Brief 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 for LL.

Method of operation

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:

$Y = KP \cdot YE = KP \cdot (W1 + W2 + W3 - X)$

$YE = W1 + W2 + W3 - X$

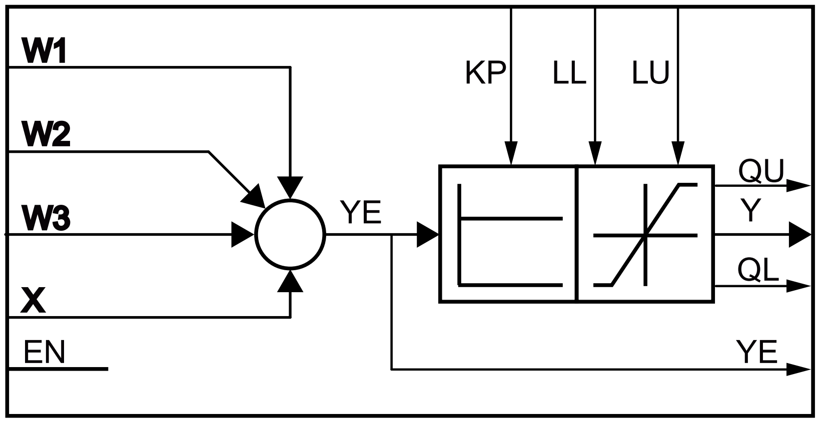
The system deviation YE is always calculated, irrespective of the operating mode, and is lead out separately.

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	Condition	Y	QU	QL	Operating 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 upper limit

Truth table for LL >= LU

EN	Condition	Y	QU	QL	Operating mode
0	None	0	1	1	Controller disable
0	LL >= LU	LU	1	1	Controller at upper limit

Block connections

Block connection	Description	Default	Value range	Attrib-utes
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	

Block connection	Description	Default	Value range	Attrib-utes
QU	Controller at upper limit	1	0/1	
QL	Controller at lower limit	1	0/1	

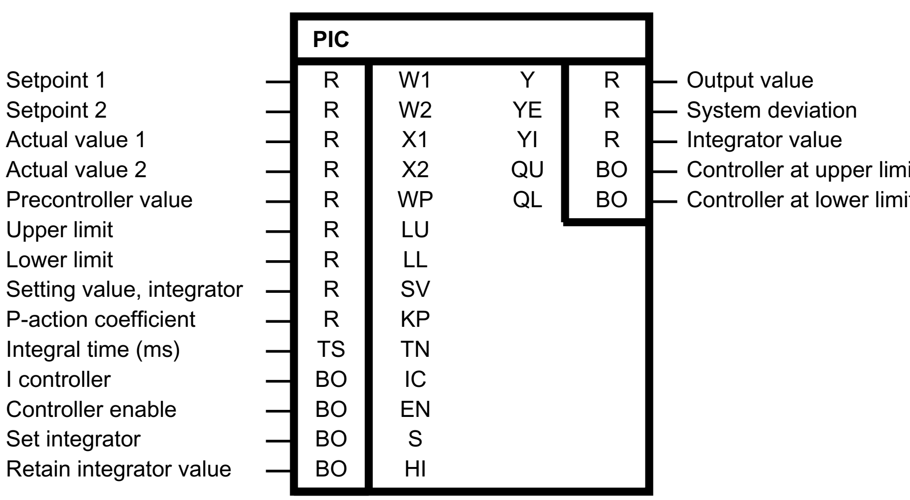
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

8.10 PIC

PI controller

Symbol



Brief 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 to 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 in 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 droop injection
- Indication when set limits are reached

Note

DCC SINAMICS

If you use this block in the following execution groups, you must assign parameter p2048 the value of the isochronous master clock:

- Receive AFTER IF1 PROFIdrive PZD
- Send BEFORE IF1 PROFIdrive PZD
- Receive AFTER IF1 PROFIdrive flexible PZD
- Receive AFTER IF2 PZD
- Send BEFORE IF2 PZD
- Receive AFTER IF2 flexible PZD

Method of operation

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:

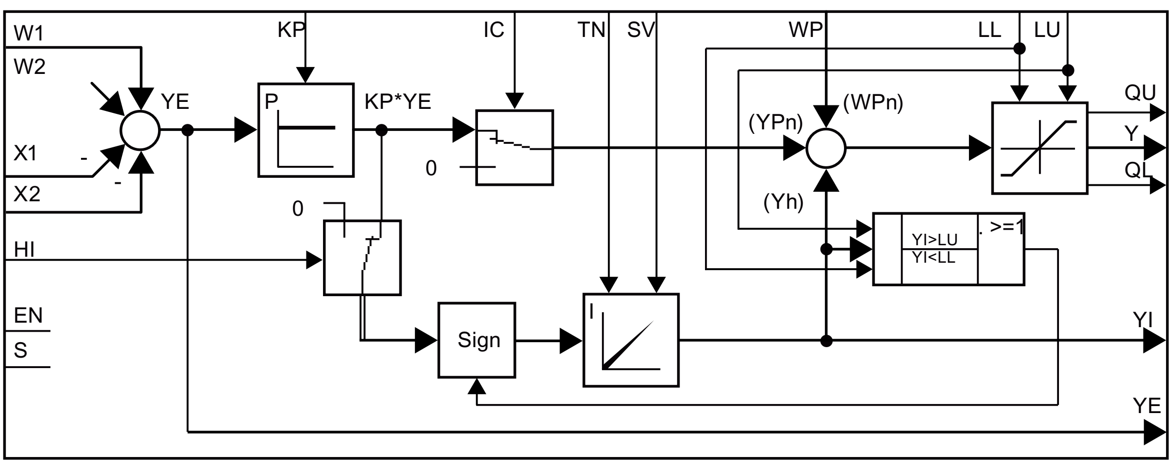
$$Y_n = Y_{n-1} + KP \cdot \left[\left(1 + \frac{TA}{TN} \right) \cdot YE_n - YE_{n-1} \right]$$

With the boundary conditions: LL<Y<LU and LL<LU

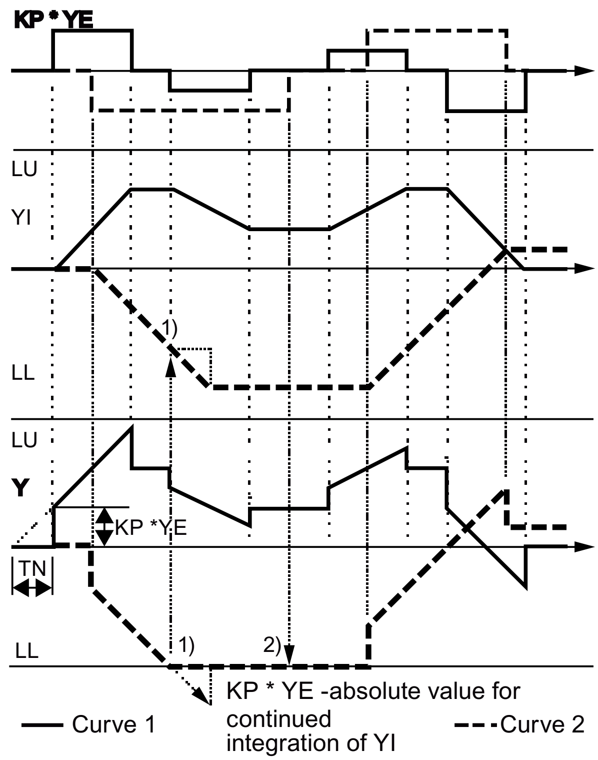
Y_n	Value of Y in scan interval n
Y_{n-1}	Value of Y in scan interval n-1

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 use of limiting (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 < Yn < 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.	Condition	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 comes up against one of the set limitations LL or LU during the control process, integrator YI will continue to run if applicable, until it comes up against the limitation itself and is retained 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	Operating mode	Remark
0	*	*	*	*	0	0	Controller disable	KP, RN, WP, LU, LL, YE any value

EN	IC	S	HI	ΔYI_n	YI_n	Y_n	Operating mode	Remark
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 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	0	0	YI_{n-1}	KP*YE_n+YI_n+WP	P controller, integrator = constant	YI_n = YI_{n-1}
1	1	0	0	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	Operating mode	Remark
1	0	0	0	$KP*YE_n \times TA/TN$	$YI_{n-1}+\Delta YI_n$ 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	$KP*YE_n*TA/TN$	$YI_{n-1}+\Delta YI_n$ 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 < LU-LU 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 < LU-LU for SV_n >= LU	LU	I controller at upper limit	$YI_n = SV_n$ or YI_n = LU, P 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 component = 0

*= any value

Operating condition 3

EN	IC	S	HI	ΔYI_n	YI_n	Y_n	Operating mode	Remark
1	0	0	0	$KP*YE_n \times TA/TN$	$YI_{n-1}+\Delta YI_n$ 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	$KP*YE_n*TA/TN$	$YI_{n-1}+\Delta YI_n$ 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 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 component = 0

*= any value

Operating condition 4

EN	IC	S	HI	ΔYI_n	YI_n	Y_n	Operating mode	Remark
1	*	*	*	*	*	LL=LU	See operating condition 2 or 3	-

*= any value

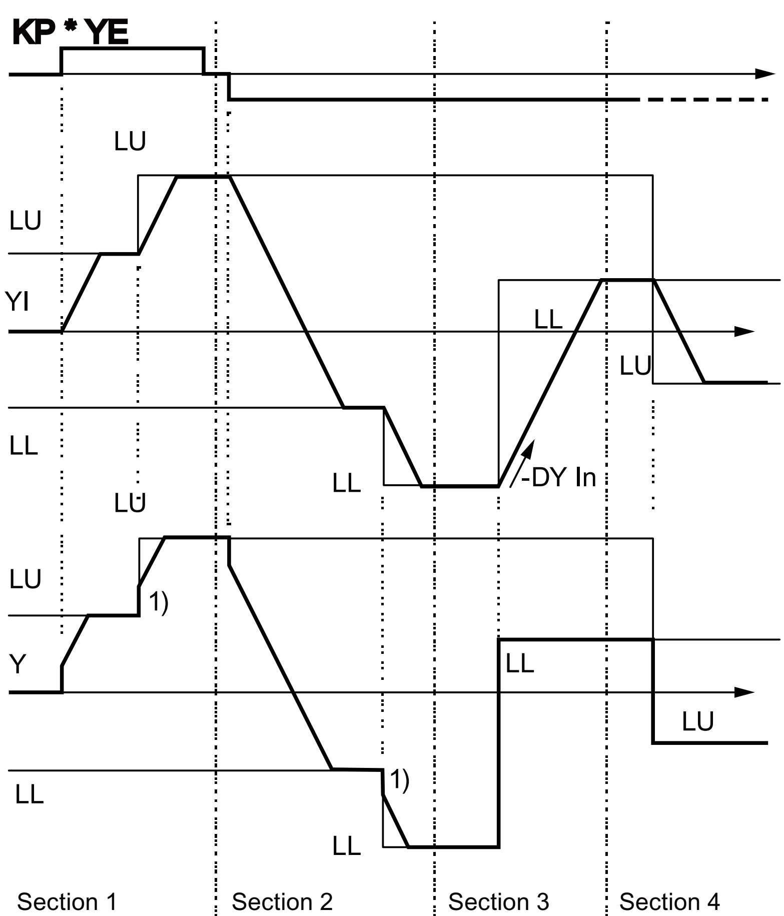
Operating condition 5

EN	IC	S	HI	ΔYI_n	YI_n	Y_n	Operating mode	Remark
1	*	*	*	$KP*YE_n TA/TN$	$YI_{n-1}+\Delta YIn$ for $YI_{n-1} < LU-YI_{n-1}-\Delta YI_n$ for $YI_{n-1} > LU-LU$ 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 operating 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 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 During a reset 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.

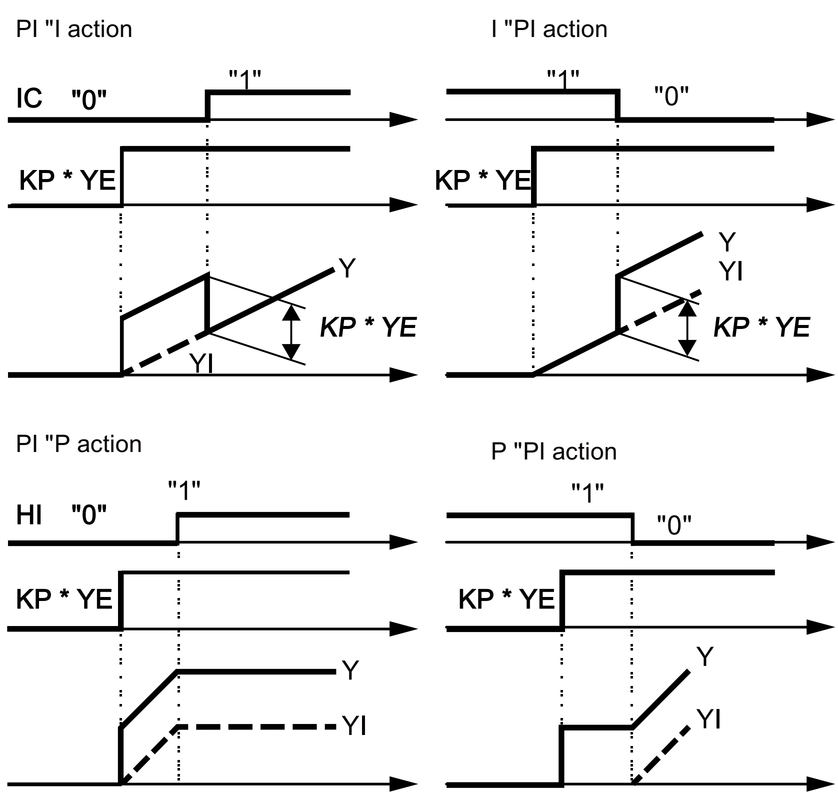
Change-over from PI mode to P mode

If block inputs EN = 1 and HI = 1, the integrator YI is retained, and a bumpless controller change-over takes place from PI mode 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



Note

The system deviation YE is always calculated and output, irrespective of the active control commands and the operating modes.

The integrator operates internally with increased accuracy, such that integration takes place even with a small system deviation. You must ensure that the sampling time is sufficiently short with reference to the integral time TN. TN is limited internally: TN >= TA.

Block connections

Block connection	Description	Default	Value range	Attrib-utes
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	

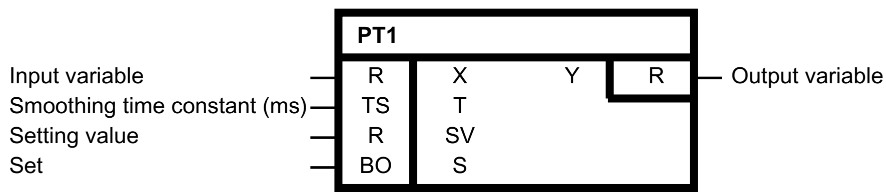
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

8.11 PT1

Delay element

Symbol



Brief description

- First-order delay element with setting function
- Use as smoothing element

Note

DCC SINAMICS

If you use this block in the following execution groups, you must assign parameter p2048 the value of the isochronous master clock:

- Receive AFTER IF1 PROFIdrive PZD
- Send BEFORE IF1 PROFIdrive PZD
- Receive AFTER IF1 PROFIdrive flexible PZD
- Receive AFTER IF2 PZD
- Send BEFORE IF2 PZD
- Receive AFTER IF2 flexible PZD

Method of operation

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 indicates the time at which the transfer function has risen to 63% of its end value.

After t = 3T, the transfer function reaches approximately 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:

$$Y_n = Y_{n-1} + \frac{TA}{T} \cdot (X_n - Y_{n-1})$$

Y_n	Value of Y in scan interval n
Y_{n-1}	Value of Y in scan interval n-1
X_n	Value of X in scan interval n

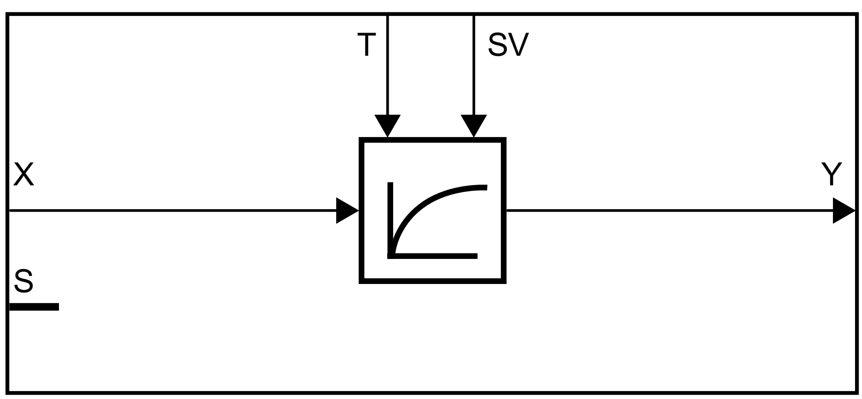
Setting function active (S = 1)

When the setting function is active, the actual setting value SVn is accepted at the output variable: Yn = SVn

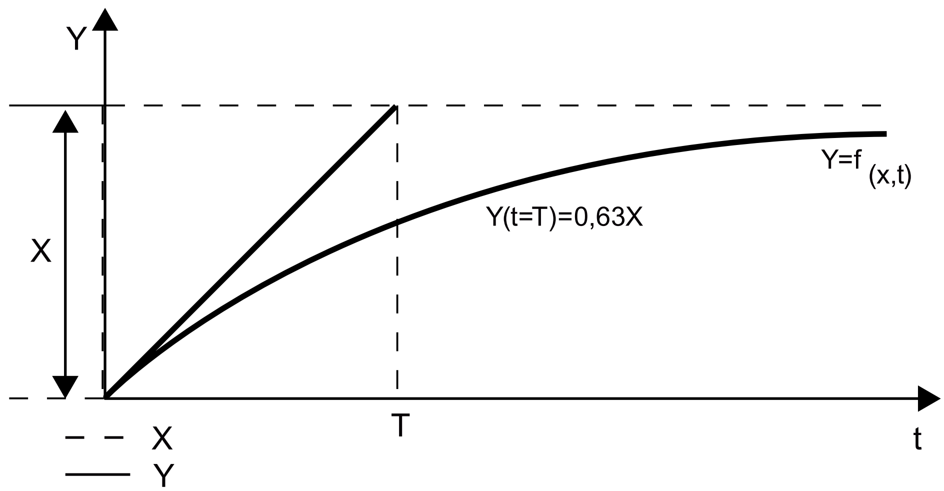
Note

The higher the T/TA, the smaller the amplitude change at Y from one sampling time to the next. TA is the sampling time in which the block is configured. T is limited internally: T>=TA.

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	Default	Value range	Attrib-utes
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	

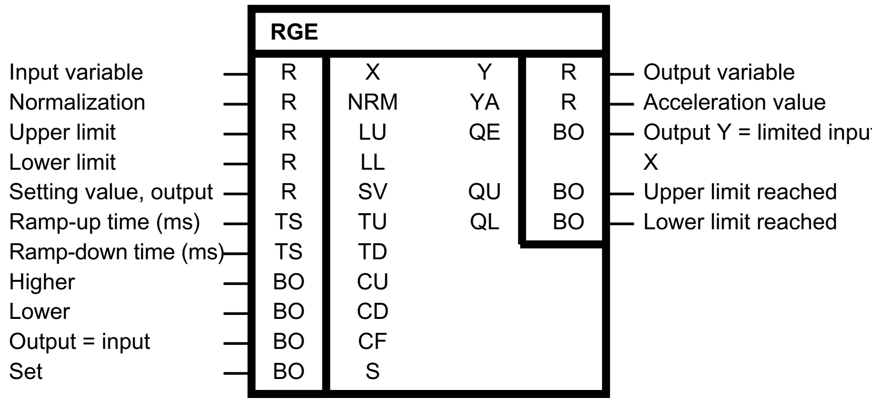
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded online	Yes
Special characteristics	-

8.12 RGE

Ramp-function generator

Symbol



Brief 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

---

Note

DCC SINAMICS

If you use this block in the following execution groups, you must assign parameter p2048 the value of the isochronous master clock:

- Receive AFTER IF1 PROFIdrive PZD
- Send BEFORE IF1 PROFIdrive PZD
- Receive AFTER IF1 PROFIdrive flexible PZD
- Receive AFTER IF2 PZD
- Send BEFORE IF2 PZD
- Receive AFTER IF2 flexible PZD

Method of operation

The block contains an integrator with two integration time constants that can be set separately. Output Y changes according to the algorithm:

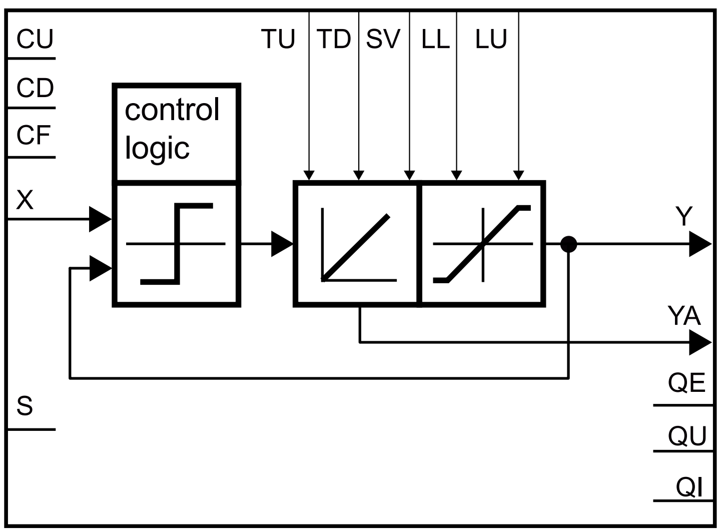
$Y_n = Y_{n-1} + YA_n$

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:

$$YA = \frac{TA}{TU} \cdot NRM \text{ for } Y > 0$$

$$YA = -\frac{TA}{TU} \cdot NRM \text{ for } Y < 0$$

The following applies for the ramp down acceleration value:

$$YA = -\frac{TA}{TD} \cdot NRM \text{ for } Y > 0$$

$$YA = \frac{TA}{TD} \cdot NRM \text{ for } Y < 0$$

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 decreases 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 scan time to the next. TA is the scan time in 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 comes up against one of the set limits LL or LU during the control process, the integrator value is retained. 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 value is increased and it has been defined in the control logic that the ramp-function generator should run in the same direction, the integrator continues to integrate from the previously held value according to the set ramp up time, until the output once again comes up against the limit value.

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.

Note

The integrator operates internally with increased accuracy, such that integration takes place even with a small setpoint-actual value difference. You must ensure that the sampling time is sufficiently short relative to the ramp-up or ramp-down time.

Note

If the absolute value of a limit value is reduced and the current value of the output is outside the limits, the integrator integrates from the current value according to the set ramp down time, until the output reaches the limit value. This process is applicable in all operating modes.

TU and TD are limited internally: TU >= TA, TD >= TA

Truth table(s)

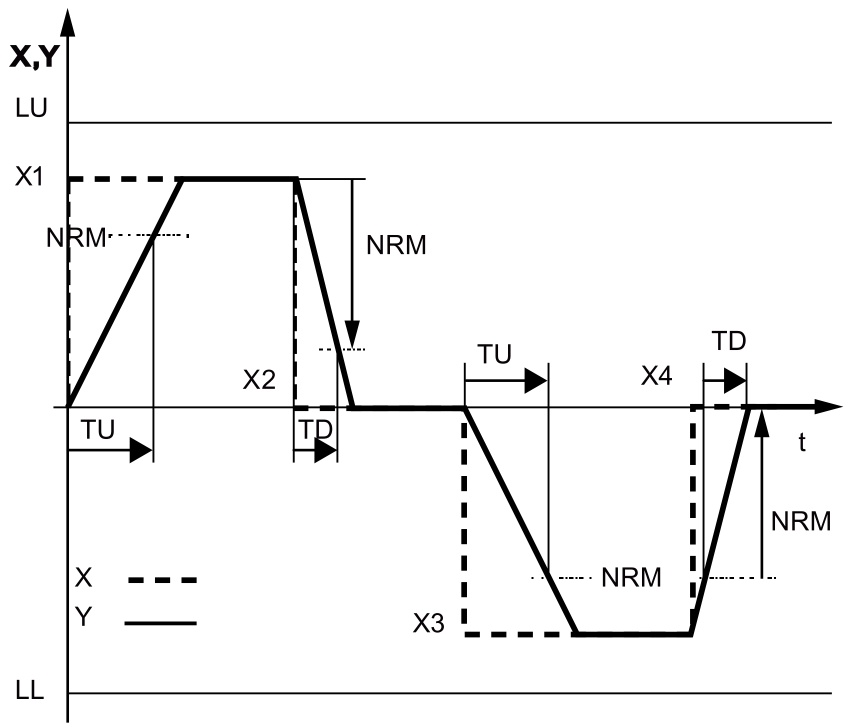
S	CF	CU	CD	Y_{An}	Y_n	Operating mode	Remark
0	0	0	0	0	Y_{n-1}	Stop	Y is constant

LL < LU and LL < actual value Y_{n-1} < LU

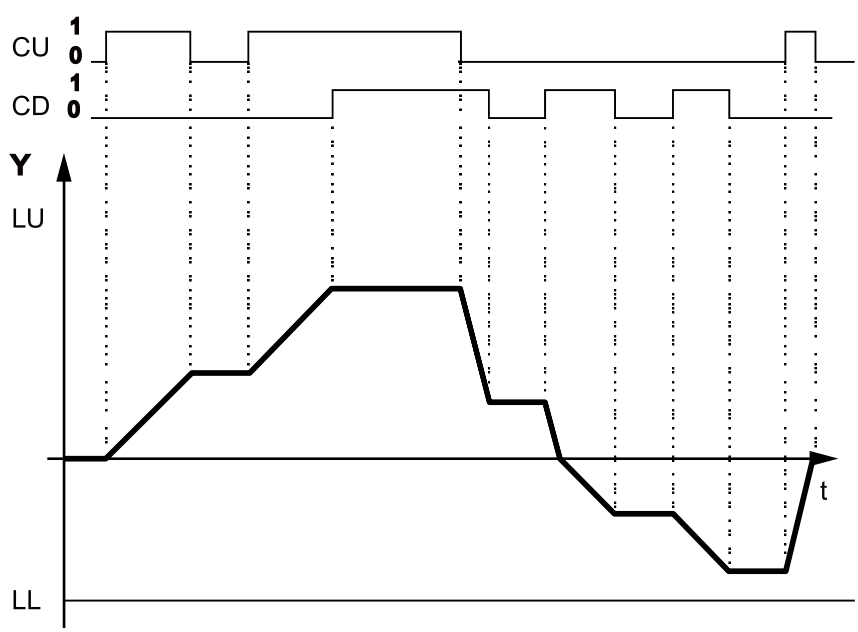
S	CF	CU	CD	Y_{An}	Y_n	Operating mode	Remark
1	*	*	*	Jump	SV_n	Set output to SV	Any SV, fixed or variable
0	1	*	*	TA/ TU;TA/TD	$Y_{n-1}+YA_n$	Normal mode Y -> X	TU for $[X > Y \wedge Y >= 0] \vee [X < Y \wedge Y <= 0]$ TD for $[X > Y \wedge Y < 0] \vee [X < Y \wedge Y > 0]$
0	0	1	0	TA/ TU(TA/TD)	$Y_{n-1}+YA_n$	Touch upper limit 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 lower limit value Y -> LL	TU, TD as before, depending on start position

* Arbitrary

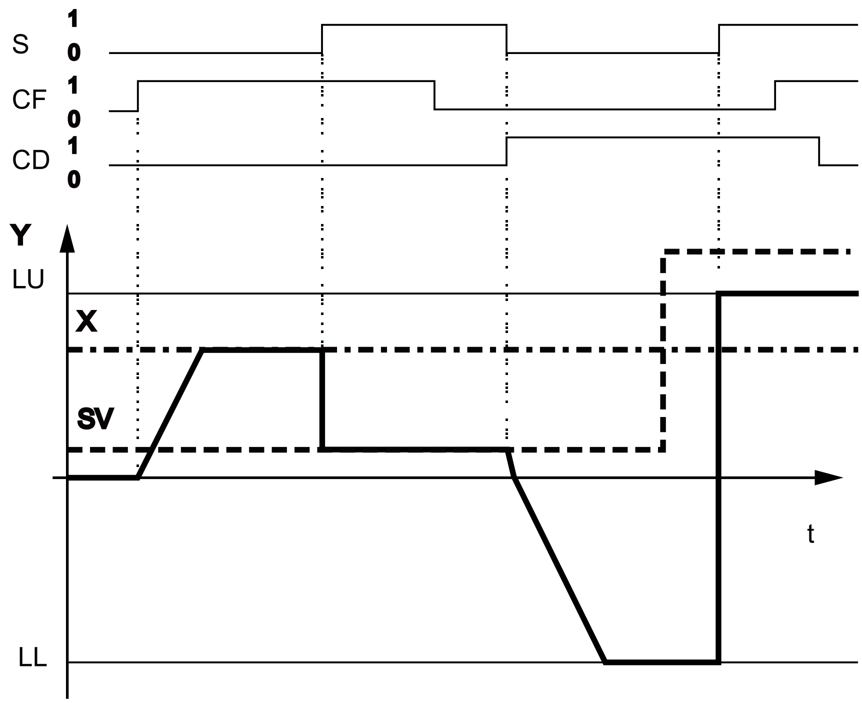
Transfer function



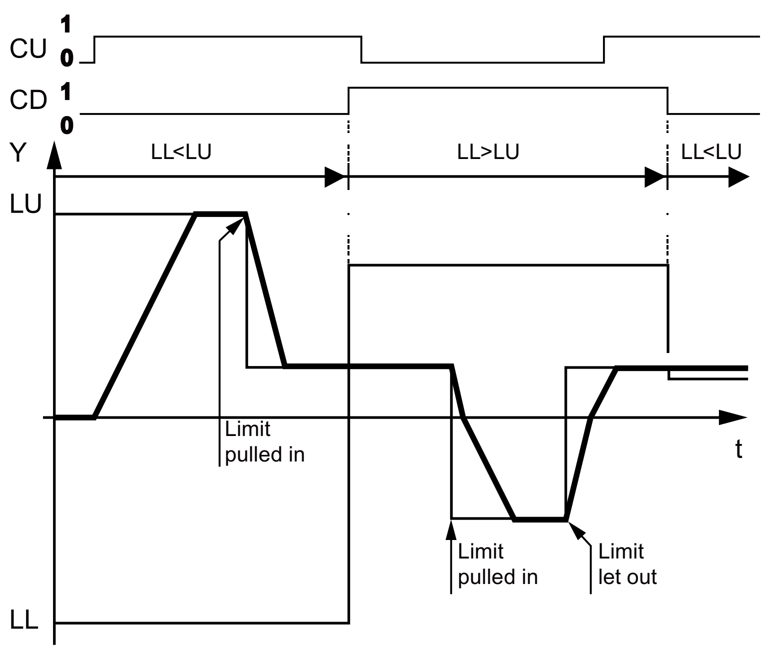
Example 1: CF = 1 with LL < LU and LL < X < LU, and X_1=1.5, X_2=X_4 =0.0, X_3=–1.5, LU=2.0, LL=–2.0, TU > TD



Example 2: Motor potentiometer function with CU and CD and with LL < LU



Example 3: Set integrator with LL < LU



Example 4: Change and swap limits

Block connections

Block connection	Description	Default	Value range	Attrib-utes
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	

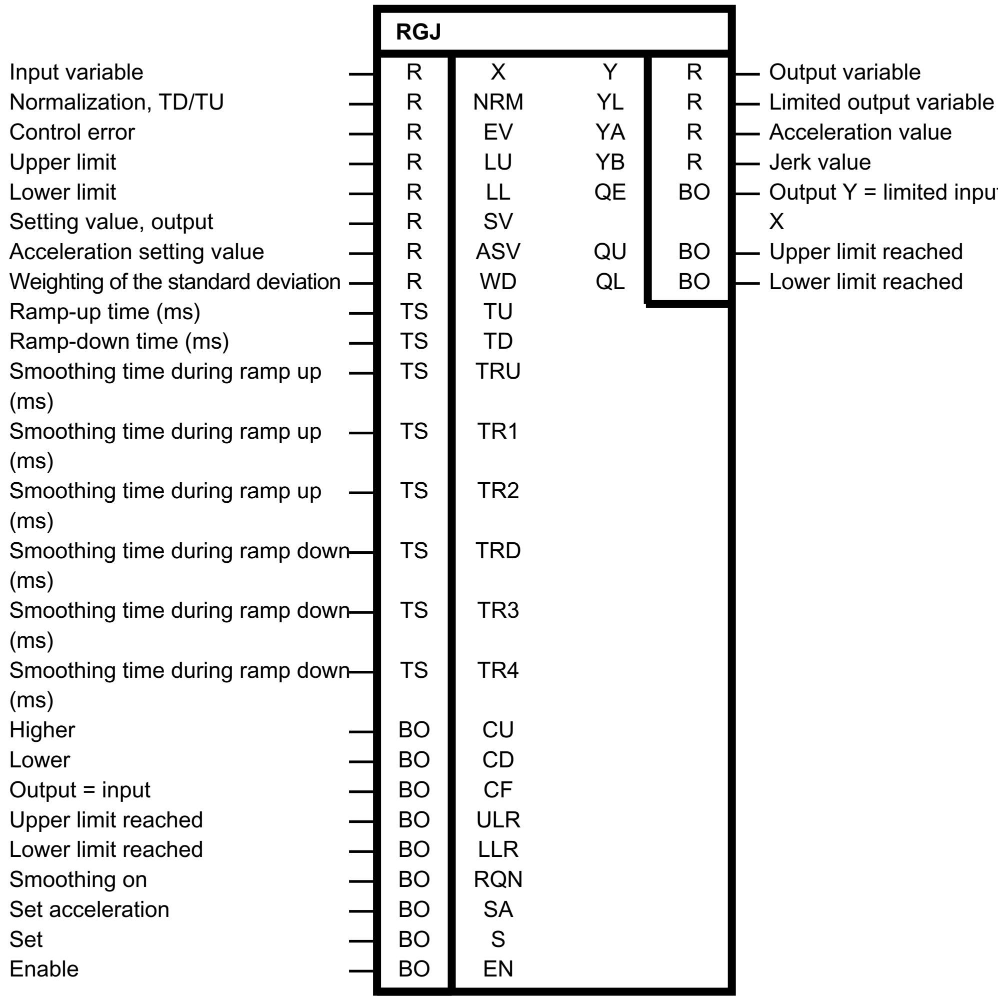
Configuration data

SIMOTION	✓
SINAMICS	✓
Can be loaded on-line	Yes
Special character-istics	-

8.13 RGJ

Ramp-function generator with jerk limiting

Symbol



Brief 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

---

Note

DCC SINAMICS

If you use this block in the following execution groups, you must assign parameter p2048 the value of the isochronous master clock:

- Receive AFTER IF1 PROFIdrive PZD
- Send BEFORE IF1 PROFIdrive PZD
- Receive AFTER IF1 PROFIdrive flexible PZD
- Receive AFTER IF2 PZD
- Send BEFORE IF2 PZD
- Receive AFTER IF2 flexible PZD

Method of operation

The block limits the acceleration (change in velocity) and the jerk (change in acceleration) of setpoints.

The following algorithms apply:

$$Y_n = Y_{n-1} + YA_n$$

$$YA_n = YA_{n-1} + YB_n$$

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 of the smoothing time during ramp up:

$$YA = YA_{max} = \frac{TA}{TU} \cdot NRM \text{ for } Y > 0$$

$$YA = YA_{max} = -\frac{TA}{TU} \cdot NRM \text{ for } Y < 0$$

The following applies for the acceleration value YA outside the smoothing time during ramp-down:

$$YA = YA_{max} = -\frac{TA}{TD} \cdot NRM \text{ for } Y > 0$$

$$YA = YA_{max} = \frac{TA}{TD} \cdot NRM \text{ for } Y < 0$$

The following applies for the jerk value YB during the smoothing time during ramp up:

$$YB = \frac{TA \cdot YA_{max}}{TRU}$$

or

$$YB = \frac{TA \cdot YA_{max}}{TR1} \quad YB = \frac{TA \cdot YA_{max}}{TR2}$$

The following applies for the jerk value YB during the smoothing time during ramp down:

$$YB = \frac{TA \cdot YA_{max}}{TRD}$$

or

$$YB = \frac{TA \cdot YA_{max}}{TR3} \quad YB = \frac{TA \cdot YA_{max}}{TR4}$$

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.

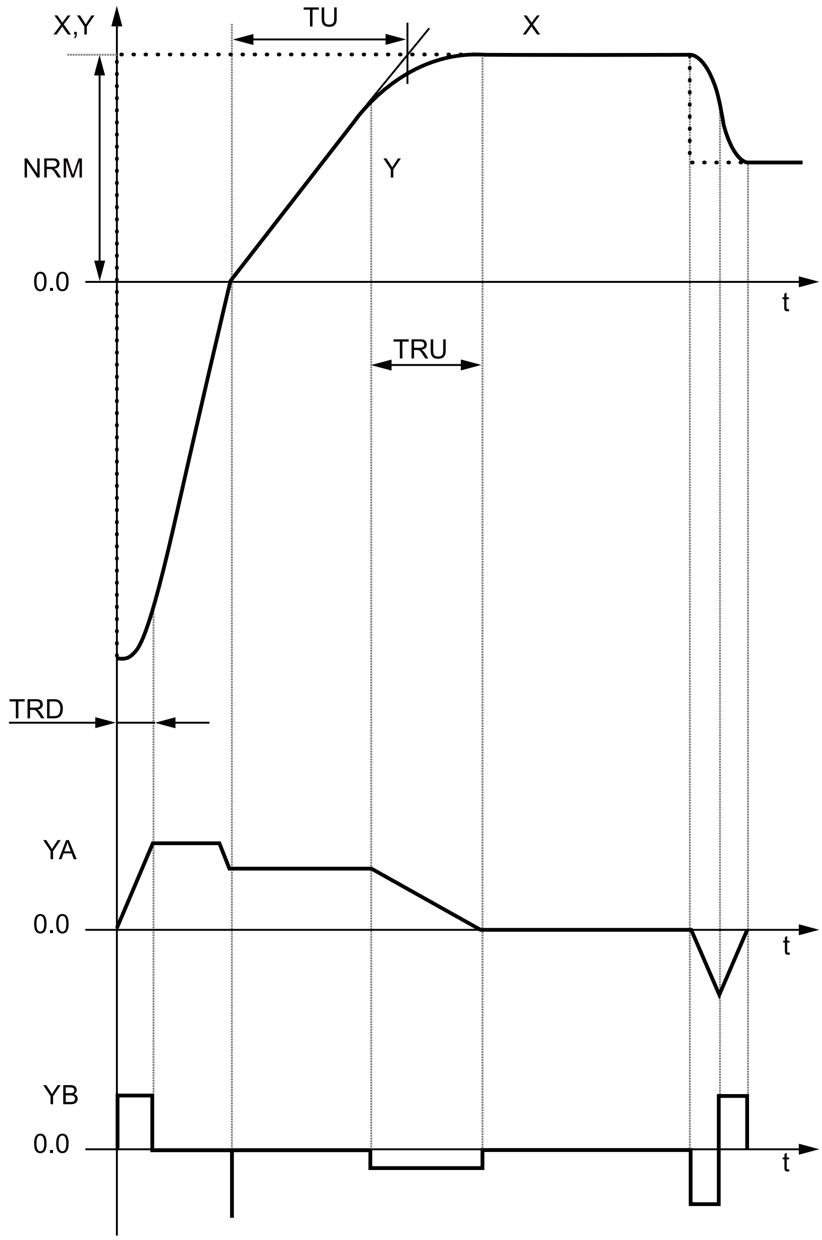
Binary output QE becomes 1 when output variable Y equals the limited value of input variable X.

A ramp up process is subdivided into three phases:

- Phase 1
  - When setpoint X increases, the maximum jerk YB (depending on TRU or TR1) is defined in the first part. 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).

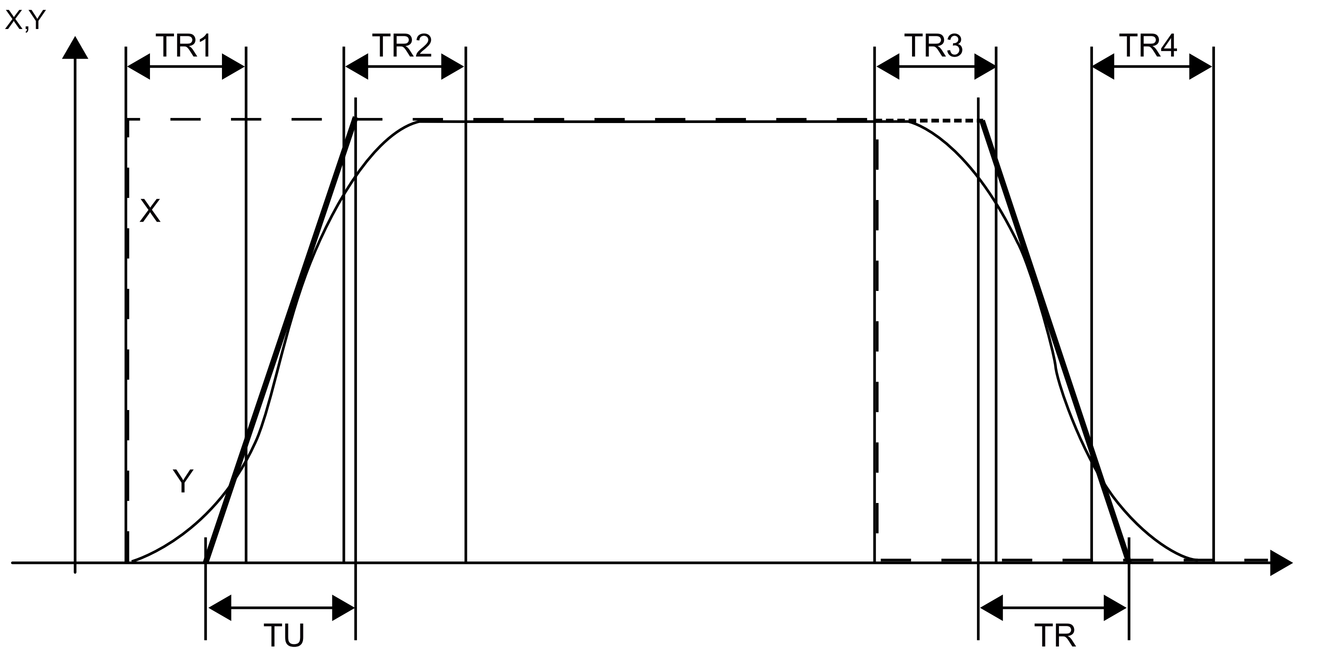
A ramp-down process proceeds analogously.

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 (compare with phase 1).

The smoothing time is also defined as the time in which the output variable reaches a constant final value starting from its maximum acceleration value (compare with phase 3). The smoothing time is defined with TRU or TR1 and TR2 during a ramp-up process and with TRD or TR3 and TR4 during a ramp-down process.

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 reduced during the ramp-down to the extent that no overshooting occurs in 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 block behaves the same as the RGE block.

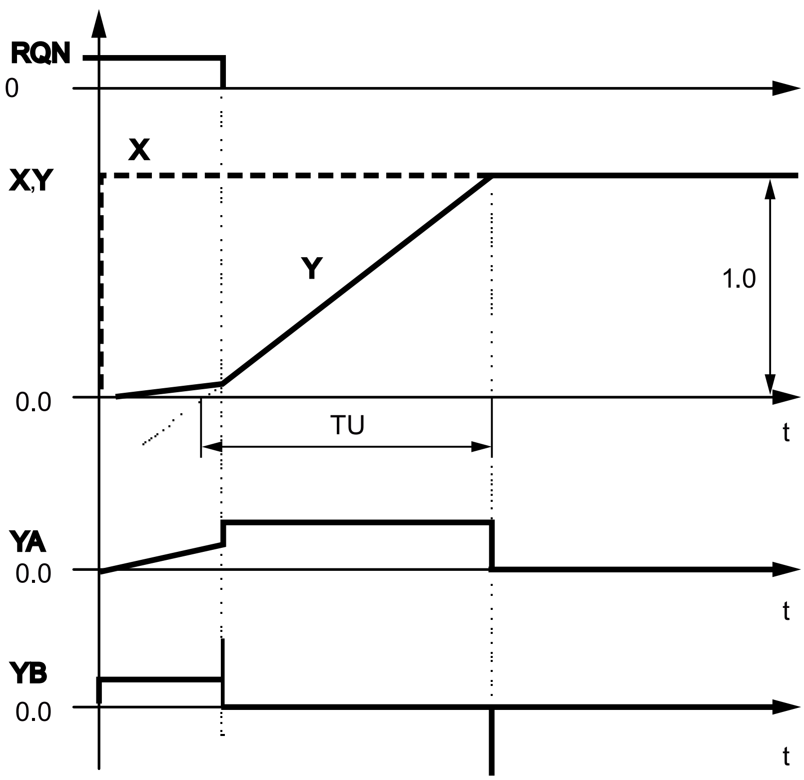
Note

Smoothing does not take place at the zero crossing. As a consequence, as a result of the inherent physics, for a difference between the ramp-up time and the ramp-down time, a step occurs in the acceleration. When required, smoothing can be initiated at the zero crossing by entering "intermediate setpoint = 0" at the input of the RGJ.

Enabling smoothing (jerk limitation)

Smoothing is active during ramp up and ramp down when RQN=1.

Transfer function: Switching off 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

Note

The following priority sequence applies regarding the control inputs:

EN before S before SA before CF before CU and CD.

The combination of commands at the control inputs and the possible operating modes can be found in the truth tables.

Normal ramp function generator operation provides for LL <= 0 <= LU and LL <= Yn <= LU. However, other settings, explained below, are also possible:

The following applies to the setting LL >= LU: The LU limit is dominant with respect to the LL limit.

During all transition processes, the values for acceleration and jerk are not exceeded.

Depending on the setpoint definition or limit offset, a characteristic with only smoothing transitions according to phases 1 and 3 appears, if applicable. In this case, a characteristic of output variable Y that is proportional over time does not occur.

Truth table(s)

Ramp-function generator stop

EN	S	SA	CF	CU	CD	Y_{An}	Y_n	Operating mode	Remark
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	SA	CF	CU	CD	YA_n	Y_n	Operating mode	Remark
1	1	*	*	*	*	Jump	SV_n	Set output to SV	Any SV, fixed or variable
1	0	1	*	*	*	ASV_n	$Y_{n-1}+YA_n$	Set output to integrator 1 on ASV	Any ASV, fixed or variable

EN	S	SA	CF	CU	CD	YA_n	Y_n	Operating mode	Remark
1	0	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] QE=1 is set when Y=X is reached.
1	0	0	0	1	0	TA/ TU(TA/TD)	$Y_{n-1}+YA_n$	Touch upper limit value Y -> LU	TU, TD as above, depending on the start position QU=1 and QE=1 are set when Y=LU is reached.
1	0	0	0	0	1	TA/ TD(TA/TU)	$Y_{n-1}+YA_n$	Touch lower limit value Y -> LL	TU, TD as above, depending on the start position QL=1 and QE=1 are set when Y=LL is reached.

Correction of the ramp-function generator

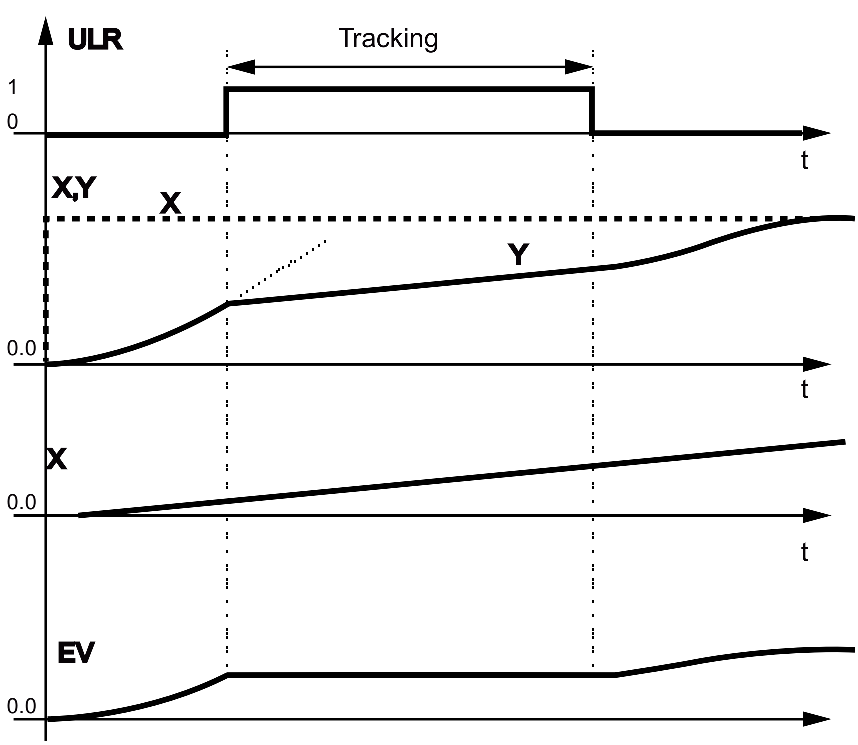
In general, output Y of the ramp-function generator is carried as a setpoint to a lower-level control loop (e.g. speed controller).

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)



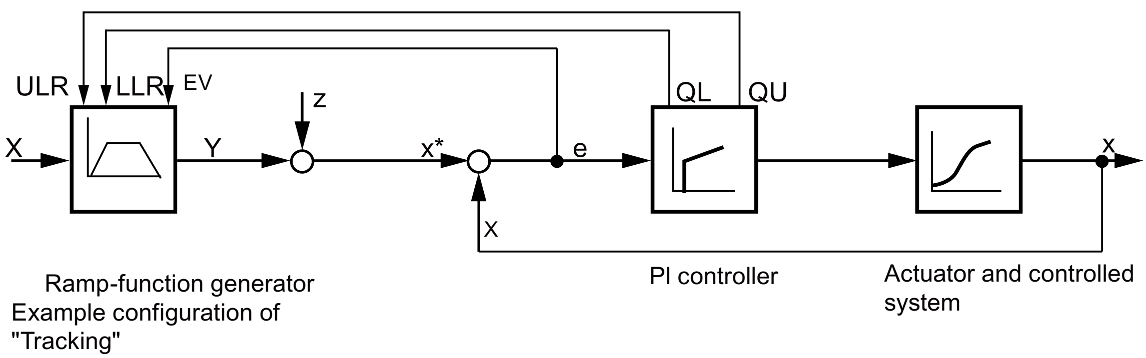
Example of tracking mode

In general, this correction can only be used for "classic control loops" (e.g. PI speed controllers). The controller limits must be set correctly (e.g. equal to the current limits).

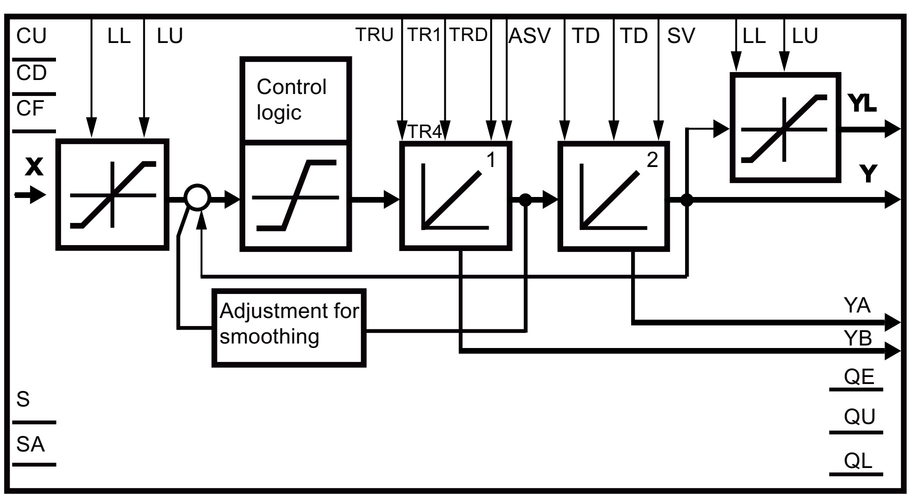
Generally, WD is 1.01 to 1.1 (> 1.0!). Jerk limiting is not active during the correction.

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	Default	Value range	Attrib-utes
X	Input variable	0.0	REAL	
NRM	Normalization, TD/TU	1.0	REAL	
EV	Control error	0.0	REAL	

Block connection	Description	Default	Value range	Attrib-utes
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	

Configuration data

SIMOTION	✓
SINAMICS	✓

Can be loaded online	Yes
Special characteristics	-

Messages and parameters

A

A.1 Messages

'All objects: DCC, DCC_DC

F51000	**DCC: Logon of the runtime group with sampling time management rejected**
Drive object:	'All objects
Message class:	General drive fault (19)
Message value:	-
Reaction:	NONE
Acknowledge:	IMMEDIATELY
Cause:	For the sampling time management of the basic SINAMICS system, Drive Control Chart (DCC) made an attempt to log on a sampling time that cannot be implemented. The logon was rejected. STARTER: Fault value (r0949, decimal interpretation): Number of the p21000 index of the runtime group where the sampling time was incorrectly set. Number of the runtime group = fault value StartDrive: Fault value (r0949, interpret decimal) indicates the position of the chart in the chart sequence. Number of the fault value = position of the chart (numbering starts with 1)
Remedy:	Attempt to assign this runtime group another fixed or free runtime 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
Message class:	General drive fault (19)
Message value:	%1
Reaction:	NONE
Acknowledge:	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. STARTER: Fault value (r0949, decimal interpretation): Number of the p21000 index of the runtime group where the sampling time was incorrectly set. Number of the runtime group = fault value StartDrive: Fault value (r0949, interpret decimal) indicates the position of the chart in the chart sequence. Number of the fault value = position of the chart (numbering starts with 1)
Remedy:	The fault can be immediately acknowledged, as the system runtime group 0 (corresponds to "Do not calculate") was assigned in p21000[x]. In Startdrive it is p21100+5*(chart ID-1). n= position of the chart in the chart sequence (first chart has position 1) Note: In window "Set runtime groups" in the context menu of the chart, p21000[0] (p21100 with Startdrive) is the topmost entry and p21000[9] is the lowest entry. The current assignment of hardware sampling times can be read-out in r21008.

F51004 DCC: Sampling time of the free runtime group differs at download

Drive object: 'All objects

Message class: General drive fault (19)

Message value: %1

Reaction: NONE

Acknowledge: IMMEDIATELY

Cause: In the STARTER/SCOUT project that was downloaded, the hardware sampling time of a free runtime group (1 <= p21000[i] or p21100+5*(n-1) with Startdrive <= 256) was set to a value that was either too low or too high. n= position of the chart in the chart sequence (first chart has position 1). The sampling time must lie between 1 ms and the value (r21003 - r21002).

If the sampling time of the selected free runtime group is < 1 ms, then 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. To prevent the fault, the determined software sampling time can be set in the runtime group (1001 <= p21000[i] <= 1096 with STARTER or 1001 <= p21100+5*(n-1) <= 1096 with Startdrive).

At least one block is assigned to the free runtime group involved.

After correcting the selection in p21000[i] (p21100+5*(n-1) with Startdrive) in the project, if this fault still occurs during download, please check which runtime group is involved based on the fault value (r0949). Only one F51004 fault is signaled at a time, even if several runtime groups have been incorrectly parameterized in p21000[] (p21100+5*(n-1) with Startdrive).

STARTER:

Fault value (r0949, decimal interpretation):

Number of the p21000 index of the runtime group where the sampling time was incorrectly set.

Number of the runtime group = fault value

StartDrive:

Fault value (r0949, interpret decimal) indicates the position of the chart in the chart sequence. Number of the fault value = position of the chart (numbering starts with 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 runtime group or remove all of the blocks from the runtime group.

F51005 DCC: Sampling time of the fixed runtime group differs online

Drive object: 'All objects

Message class: General drive fault (19)

Message value: %1

Reaction: NONE

Acknowledge: IMMEDIATELY

Cause: Generally, the sampling times of the fixed runtime groups correspond to the sampling times of the associated system function (e.g. the sampling time of the fixed runtime 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 runtime group belonging to this system function (1 ms). The sampling time is set to 1 ms. At least one block is assigned to the fixed runtime group involved.

STARTER:

Fault value (r0949, decimal interpretation):

Number of the p21000 index of the runtime group where the sampling time was incorrectly set.

Number of the runtime group = fault value

StartDrive:

Fault value (r0949, interpret decimal) indicates the position of the chart in the chart sequence. Number of the fault value = position of the chart (numbering starts with 1)

Remedy: Using parameter p0112 or p0115, increase the sampling time of the system function to the minimum permissible sampling time for the runtime groups of 1 ms or remove all of the blocks from the runtime group.

F51006 DCC: Sampling time of the fixed runtime group differs at download

Drive object: 'All objects

Message class: General drive fault (19)

Message value: %1

Reaction: NONE

Acknowledge: IMMEDIATELY

Cause: Generally, the sampling times of the fixed runtime groups correspond to the sampling times of the associated system function (e.g. the sampling time of the fixed runtime 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 runtime group belonging to this system function (1 ms). The sampling time is set to the smallest possible value (r21002 on the drive object).

STARTER:

Fault value (r0949, decimal interpretation):

Number of the p21000 index of the runtime group where the sampling time was incorrectly set.

Number of the runtime group = fault value

StartDrive:

Fault value (r0949, interpret decimal) indicates the position of the chart in the chart sequence. Number of the fault value = position of the chart (numbering starts with 1)

Remedy: Using parameter p0112 or p0115, increase the sampling time of the system function to the minimum permissible sampling time for the runtime groups of 1 ms or remove all of the blocks from the runtime group.

F51008 DCC: No NVRAM available

Drive object: 'All objects

Message class: General drive fault (19)

Message value: %1

Reaction: OFF2

Acknowledge: 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

Message class: General drive fault (19)

Message value: -

Reaction: OFF2

Acknowledge: 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.

A51032 DCC: Internal measurement active

Drive object: 'All objects
Message class: General drive fault (19)
Message value: -
Reaction: NONE
Acknowledge: NONE
Cause: A Siemens internal measurement has been activated.
Remedy: Carry out a POWER ON (switch off/on) for the Control Unit involved.

F51033 Licensing DCC application not sufficient

Drive object: 'All objects
Message class: General drive fault (19)
Message value: -
Reaction: NONE
Acknowledge: 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.

F51034 DCC: block runtimes are not measured

Drive object: 'All objects
Message class: General drive fault (19)
Message value: -
Reaction: NONE
Acknowledge: IMMEDIATELY
Cause: A block library created with DCB Studio contains blocks for which the runtime has not been measured. Contact the person that created the library.
Remedy:

F51035 DCC: DCC configuration error

Drive object: 'All objects
Message class: General drive fault (19)
Message value: -
Reaction: OFF2
Acknowledge: IMMEDIATELY
Cause: An error has occurred when powering up from a DCC configuration.
Remedy: - Evaluate fault buffer (r0945).
- Carry out a POWER ON for all components
(switch off/switch on)
- If required, check the data on the non-volatile memory (e.g. memory card).
- Upgrade firmware to later version.
- Contact the Hotline.

F51036 (A) DCC: Error for online changes in the DCC chart

Drive object: 'All objects
Message class: General drive fault (19)
Message value: %1
Reaction: Infeed: OFF2 (NONE, OFF1, OFF1_DELAYED)
Servo: OFF2 (ENCODER, IASC/DCBRK, NONE, OFF1, OFF1_DELAYED, OFF3, STOP2)
Vector: OFF2 (ENCODER, IASC/DCBRK, NONE, OFF1, OFF1_DELAYED, OFF3, STOP2)
Chopper: OFF2 (NONE)
Hla: OFF2 (ENCODER, NONE, OFF1, OFF3, STOP2)
Acknowledge: IMMEDIATELY (POWER ON)
Cause: The changes in the DCC chart were not able to be undone.
Remedy: - Restart the device and download the DCC configuration into the device.
- Upgrade firmware to later version.
- Contact the Hotline.
Reaction upon A: NONE
Acknowl. upon A: NONE

F51037 (A) DCC: Alarm for online changes in the DCC chart

Drive object: 'All objects
Message class: General drive fault (19)
Message value: %1
Reaction: Infeed: NONE (OFF1, OFF1_DELAYED, OFF2)
Servo: NONE (ENCODER, IASC/DCBRK, OFF1, OFF1_DELAYED, OFF2, OFF3, STOP2)
Vector: NONE (ENCODER, IASC/DCBRK, OFF1, OFF1_DELAYED, OFF2, OFF3, STOP2)
Chopper: NONE (OFF2)
Hla: NONE (ENCODER, OFF1, OFF2, OFF3, STOP2)
Acknowledge: IMMEDIATELY (POWER ON)
Cause: The changes in the DCC chart were not able to be undone.
Remedy: - Restart the device and download the DCC configuration into the device.
- Upgrade firmware to later version.
- Contact the Hotline.
Reaction upon A: NONE
Acknowl. upon A: NONE

A51038 DCC: Performance limits of online changes reached

Drive object: 'All objects
Message class: General drive fault (19)
Message value: %1
Reaction: NONE
Acknowledge: NONE
Cause: Fault value (r2124, interpret decimal):
0: Another fault has occurred.
1: Too many online changes were requested. Saving such a project to the CF card slows down the device restart, and it takes longer to execute online changes.
Remedy: Load into the PC/PG and then load the DCC configuration into the device.

F51050 DCC: Fault initiated by "Drive Control Chart"

Drive object: 'All objects
Message class: General drive fault (19)

Message value: %1

Reaction: Infeed: OFF2 (NONE, OFF1)
Servo: OFF2 (ENCODER, IASC/DCBRK, NONE, OFF1, OFF3, STOP2)
Vector: OFF2 (ENCODER, IASC/DCBRK, NONE, OFF1, OFF3, STOP2)
Chopper: OFF2 (NONE)
Hla: OFF2 (ENCODER, NONE, OFF1, OFF3, STOP2)

Acknowledge: IMMEDIATELY (POWER ON)

Cause: "Drive Control Chart" (DCC) had 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

Message class: General drive fault (19)

Message value: %1

Reaction: Infeed: OFF2 (NONE, OFF1)
Servo: OFF2 (ENCODER, IASC/DCBRK, NONE, OFF1, OFF3, STOP2)
Vector: OFF2 (ENCODER, IASC/DCBRK, NONE, OFF1, OFF3, STOP2)
Chopper: OFF2 (NONE)
Hla: OFF2 (ENCODER, NONE, OFF1, OFF3, STOP2)

Acknowledge: IMMEDIATELY (POWER ON)

Cause: "Drive Control Chart" (DCC) had 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

Message class: General drive fault (19)

Message value: %1

Reaction: Infeed: OFF2 (NONE, OFF1)
Servo: OFF2 (ENCODER, IASC/DCBRK, NONE, OFF1, OFF3, STOP2)
Vector: OFF2 (ENCODER, IASC/DCBRK, NONE, OFF1, OFF3, STOP2)
Chopper: OFF2 (NONE)
Hla: OFF2 (ENCODER, NONE, OFF1, OFF3, STOP2)

Acknowledge: IMMEDIATELY (POWER ON)

Cause: "Drive Control Chart" (DCC) had 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

Message class: General drive fault (19)
Message value: %1
Reaction: Infeed: OFF2 (NONE, OFF1)
Servo: OFF2 (ENCODER, IASC/DCBRK, NONE, OFF1, OFF3, STOP2)
Vector: OFF2 (ENCODER, IASC/DCBRK, NONE, OFF1, OFF3, STOP2)
Chopper: OFF2 (NONE)
Hla: OFF2 (ENCODER, NONE, OFF1, OFF3, STOP2)
Acknowledge: IMMEDIATELY (POWER ON)
Cause: "Drive Control Chart" (DCC) had 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
Message class: General drive fault (19)
Message value: %1
Reaction: Infeed: OFF2 (NONE, OFF1)
Servo: OFF2 (ENCODER, IASC/DCBRK, NONE, OFF1, OFF3, STOP2)
Vector: OFF2 (ENCODER, IASC/DCBRK, NONE, OFF1, OFF3, STOP2)
Chopper: OFF2 (NONE)
Hla: OFF2 (ENCODER, NONE, OFF1, OFF3, STOP2)
Acknowledge: IMMEDIATELY (POWER ON)
Cause: "Drive Control Chart" (DCC) had 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
Message class: General drive fault (19)
Message value: %1
Reaction: Infeed: OFF2 (NONE, OFF1)
Servo: OFF2 (ENCODER, IASC/DCBRK, NONE, OFF1, OFF3, STOP2)
Vector: OFF2 (ENCODER, IASC/DCBRK, NONE, OFF1, OFF3, STOP2)
Chopper: OFF2 (NONE)
Hla: OFF2 (ENCODER, NONE, OFF1, OFF3, STOP2)
Acknowledge: IMMEDIATELY (POWER ON)
Cause: "Drive Control Chart" (DCC) had 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
Message class: General drive fault (19)
Message value: %1
Reaction: Infeed: OFF2 (NONE, OFF1)
Servo: OFF2 (ENCODER, IASC/DCBRK, NONE, OFF1, OFF3, STOP2)
Vector: OFF2 (ENCODER, IASC/DCBRK, NONE, OFF1, OFF3, STOP2)
Chopper: OFF2 (NONE)
Hla: OFF2 (ENCODER, NONE, OFF1, OFF3, STOP2)
Acknowledge: IMMEDIATELY (POWER ON)
Cause: "Drive Control Chart" (DCC) had 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
Message class: General drive fault (19)
Message value: %1
Reaction: Infeed: OFF2 (NONE, OFF1)
Servo: OFF2 (ENCODER, IASC/DCBRK, NONE, OFF1, OFF3, STOP2)
Vector: OFF2 (ENCODER, IASC/DCBRK, NONE, OFF1, OFF3, STOP2)
Chopper: OFF2 (NONE)
Hla: OFF2 (ENCODER, NONE, OFF1, OFF3, STOP2)
Acknowledge: IMMEDIATELY (POWER ON)
Cause: "Drive Control Chart" (DCC) had 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
Message class: General drive fault (19)
Message value: %1
Reaction: Infeed: OFF2 (NONE, OFF1)
Servo: OFF2 (ENCODER, IASC/DCBRK, NONE, OFF1, OFF3, STOP2)
Vector: OFF2 (ENCODER, IASC/DCBRK, NONE, OFF1, OFF3, STOP2)
Chopper: OFF2 (NONE)
Hla: OFF2 (ENCODER, NONE, OFF1, OFF3, STOP2)
Acknowledge: IMMEDIATELY (POWER ON)
Cause: "Drive Control Chart" (DCC) had 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
Message class: General drive fault (19)
Message value: %1
Reaction: Infeed: OFF2 (NONE, OFF1)
Servo: OFF2 (ENCODER, IASC/DCBRK, NONE, OFF1, OFF3, STOP2)
Vector: OFF2 (ENCODER, IASC/DCBRK, NONE, OFF1, OFF3, STOP2)
Chopper: OFF2 (NONE)
Hla: OFF2 (ENCODER, NONE, OFF1, OFF3, STOP2)
Acknowledge: IMMEDIATELY (POWER ON)
Cause: "Drive Control Chart" (DCC) had 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.

A51060 DCC: alarm initiated by "Drive Control Chart"

Drive object: 'All objects
Message class: General drive fault (19)
Message value: %1
Reaction: NONE
Acknowledge: NONE
Cause: "Drive Control Chart" (DCC) had 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 "Drive Control Chart"

Drive object: 'All objects
Message class: General drive fault (19)
Message value: %1
Reaction: NONE
Acknowledge: NONE
Cause: "Drive Control Chart" (DCC) had 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 "Drive Control Chart"

Drive object: 'All objects
Message class: General drive fault (19)
Message value: %1
Reaction: NONE
Acknowledge: NONE

Cause: "Drive Control Chart" (DCC) had 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 "Drive Control Chart"

Drive object: 'All objects

Message class: General drive fault (19)

Message value: %1

Reaction: NONE

Acknowledge: NONE

Cause: "Drive Control Chart" (DCC) had 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 "Drive Control Chart"

Drive object: 'All objects

Message class: General drive fault (19)

Message value: %1

Reaction: NONE

Acknowledge: NONE

Cause: "Drive Control Chart" (DCC) had 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 "Drive Control Chart"

Drive object: 'All objects

Message class: General drive fault (19)

Message value: %1

Reaction: NONE

Acknowledge: NONE

Cause: "Drive Control Chart" (DCC) had 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 "Drive Control Chart"

Drive object: 'All objects

Message class: General drive fault (19)

Message value: %1

Reaction: NONE
Acknowledge: NONE
Cause: "Drive Control Chart" (DCC) had 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 "Drive Control Chart"

Drive object: 'All objects
Message class: General drive fault (19)
Message value: %1
Reaction: NONE
Acknowledge: NONE
Cause: "Drive Control Chart" (DCC) had 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 "Drive Control Chart"

Drive object: 'All objects
Message class: General drive fault (19)
Message value: %1
Reaction: NONE
Acknowledge: NONE
Cause: "Drive Control Chart" (DCC) had 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 "Drive Control Chart"

Drive object: 'All objects
Message class: General drive fault (19)
Message value: %1
Reaction: NONE
Acknowledge: NONE
Cause: "Drive Control Chart" (DCC) had 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.

A.2 Parameters

Version: 5201400

All objects: DCC, DCC_DC

p21000[0...9] Runtime group properties / RTG property

DCC

Can be changed: T	**Calculated:** -	**Access level:** 1
Data type: Integer16	**Dynamic index:** -	**Function diagram:** -
P-Group: -	**Unit group:** -	**Unit selection:** -
Not for motor type: -	**Scaling:** -	**Expert list:** 1
Min:	**Max:**	**Factory setting:**
0	4005	0

Description: Allocates properties to runtime 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 runtime group:

- p21000[0] is used to set the property of the runtime group 1

...

- p21000[9] is used to set the property of the runtime group 10

Value:

0:	Do not calculate runtime 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

Recommendation: On the drive objects of CU, TB30, TM15DI_DO, TM31, TM41, TM120, the sampling time for supplementary functions is preset to p0115[0] = 4 ms. If you wish to configure a DCC runtime group with a shorter sampling time on these drive objects, then you should first set the sampling time for supplementary functions p0115[0] on this drive object to the value of the shortest sampling time required.

Index:
[0] = Runtime group 1
[1] = Runtime group 2
[2] = Runtime group 3
[3] = Runtime group 4
[4] = Runtime group 5
[5] = Runtime group 6
[6] = Runtime group 7
[7] = Runtime group 8
[8] = Runtime group 9
[9] = Runtime group 10

Dependency: See also: r7903, r21008

⚠ CAUTION

The properties of the runtime groups must not be changed during operation as this could result in discontinuous signal transitions.

Note

For value = 1 ... 256 (free runtime group):
This selection value can only be selected online if the following applies for sampling time T_sample of this runtime 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.
For value > 2000 (fixed runtime group):
The fixed runtime 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 runtime group with a sampling time >= 1 ms should be selected. When selecting the fixed runtime 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.

For value = 4002, 4003, 4005 (IF2 runtime group):
On devices where IF2 does not exist (D4xx, CU310), when selecting the runtime groups that involve IF2, the corresponding runtime group for IF1 is automatically logged on.

p21000[0...9] Runtime group properties / RTG property

DCC

Can be changed: T	**Calculated:** -	**Access level:** 1
Data type: Integer16	**Dynamic index:** -	**Function diagram:** -
P-Group: -	**Unit group:** -	**Unit selection:** -
Not for motor type: -	**Scaling:** -	**Expert list:** 1
Min:	**Max:**	**Factory setting:**
0	4004	0

Description:	Allocates properties to runtime 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 runtime group: - p21000[0] is used to set the property of the runtime group 1 ... - p21000[9] is used to set the property of the runtime group 10
Value:	0: Do not calculate runtime 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

Recommendation: On the drive objects of CU, TB30, TM15DI_DO, TM31, TM41, TM120, the sampling time for supplementary functions is preset to p0115[0] = 4 ms. If you wish to configure a DCC runtime group with a shorter sampling time on these drive objects, then you should first set the sampling time for supplementary functions p0115[0] on this drive object to the value of the shortest sampling time required.

Index: [0] = Runtime group 1
[1] = Runtime group 2
[2] = Runtime group 3
[3] = Runtime group 4
[4] = Runtime group 5
[5] = Runtime group 6
[6] = Runtime group 7
[7] = Runtime group 8
[8] = Runtime group 9
[9] = Runtime group 10

Dependency: See also: r7903, r21008

⚠ CAUTION

The properties of the runtime groups must not be changed during operation as this could result in discontinuous signal transitions.

Note

For value = 1 ... 256 (free runtime group):
This selection value can only be selected online if the following applies for sampling time T_sample of this runtime 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.
For value > 2000 (fixed runtime group):
The fixed runtime 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 runtime group with a sampling time >= 1 ms should be selected. When selecting the fixed runtime 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] Runtime group properties / RTG property

All objects

Can be changed: T	**Calculated:** -	**Access level:** 1
Data type: Integer16	**Dynamic index:** -	**Function diagram:** -
P-Group: -	**Unit group:** -	**Unit selection:** -
Not for motor type: -	**Scaling:** -	**Expert list:** 1
Min:	**Max:**	**Factory setting:**
0	4005	0

Description: Allocates properties to runtime 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 runtime group:
- p21000[0] is used to set the property of the runtime group 1
...
- p21000[9] is used to set the property of the runtime group 10

Value:

0:	Do not calculate runtime 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

Recommendation: On the drive objects of CU, TB30, TM15DI_DO, TM31, TM41, TM120, the sampling time for supplementary functions is preset to p0115[0] = 4 ms. If you wish to configure a DCC runtime group with a shorter sampling time on these drive objects, then you should first set the sampling time for supplementary functions p0115[0] on this drive object to the value of the shortest sampling time required.

Index:
[0] = Runtime group 1
[1] = Runtime group 2
[2] = Runtime group 3
[3] = Runtime group 4
[4] = Runtime group 5
[5] = Runtime group 6
[6] = Runtime group 7
[7] = Runtime group 8
[8] = Runtime group 9
[9] = Runtime group 10

Dependency: See also: r7903, r21008

> ⚠ **CAUTION**
>
> The properties of the runtime groups must not be changed during operation as this could result in discontinuous signal transitions.

Note

For value = 1 ... 256 (free runtime group):

This selection value can only be selected online if the following applies for sampling time T_sample of this runtime 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.

For value > 2000 (fixed runtime group):

The fixed runtime 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 runtime group with a sampling time >= 1 ms should be selected. When selecting the fixed runtime 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.

For value = 4002, 4003, 4005 (IF2 runtime group):

On devices where IF2 does not exist (D4xx, CU310), when selecting the runtime groups that involve IF2, the corresponding runtime group for IF1 is automatically logged on.

p21000[0...9] Runtime group properties / RTG property

DCC

Can be changed: T	**Calculated:** -	**Access level:** 1
Data type: Integer16	**Dynamic index:** -	**Function diagram:** -
P-Group: -	**Unit group:** -	**Unit selection:** -
Not for motor type: -	**Scaling:** -	**Expert list:** 1
Min:	**Max:**	**Factory setting:**
0	4005	0

Description: Allocates properties to runtime 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 runtime group:

- p21000[0] is used to set the property of the runtime group 1

...

- p21000[9] is used to set the property of the runtime group 10

Value:

0:	Do not calculate runtime 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

Recommendation: On the drive objects of CU, TB30, TM15DI_DO, TM31, TM41, TM120, the sampling time for supplementary functions is preset to p0115[0] = 4 ms. If you wish to configure a DCC runtime group with a shorter sampling time on these drive objects, then you should first set the sampling time for supplementary functions p0115[0] on this drive object to the value of the shortest sampling time required.

Index:
[0] = Runtime group 1
[1] = Runtime group 2
[2] = Runtime group 3
[3] = Runtime group 4
[4] = Runtime group 5
[5] = Runtime group 6
[6] = Runtime group 7
[7] = Runtime group 8
[8] = Runtime group 9
[9] = Runtime group 10

Dependency: See also: r7903, r21008

⚠ CAUTION
The properties of the runtime groups must not be changed during operation as this could result in discontinuous signal transitions.

Note

For value = 1 ... 256 (free runtime group):

This selection value can only be selected online if the following applies for sampling time T_sample of this runtime 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.

For value > 2000 (fixed runtime group):

The fixed runtime 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 runtime group with a sampling time >= 1 ms should be selected. When selecting the fixed runtime 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.

For value = 4002, 4003, 4005 (IF2 runtime group):

On devices where IF2 does not exist (D4xx, CU310), when selecting the runtime groups that involve IF2, the corresponding runtime group for IF1 is automatically logged on.

p21000[0...9] Runtime group properties / RTG property

DCC

Can be changed: T	**Calculated:** -	**Access level:** 1
Data type: Integer16	**Dynamic index:** -	**Function diagram:** -
P-Group: -	**Unit group:** -	**Unit selection:** -
Not for motor type: -	**Scaling:** -	**Expert list:** 1
Min:	**Max:**	**Factory setting:**
0	4005	0

Description: Allocates properties to runtime 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 runtime group:

- p21000[0] is used to set the property of the runtime group 1

...

- p21000[9] is used to set the property of the runtime group 10

Value:

0: Do not calculate runtime 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

Recommendation: On the drive objects of CU, TB30, TM15DI_DO, TM31, TM41, TM120, the sampling time for supplementary functions is preset to p0115[0] = 4 ms. If you wish to configure a DCC runtime group with a shorter sampling time on these drive objects, then you should first set the sampling time for supplementary functions p0115[0] on this drive object to the value of the shortest sampling time required.

Index:
[0] = Runtime group 1
[1] = Runtime group 2
[2] = Runtime group 3
[3] = Runtime group 4
[4] = Runtime group 5
[5] = Runtime group 6
[6] = Runtime group 7
[7] = Runtime group 8
[8] = Runtime group 9
[9] = Runtime group 10

Dependency: See also: r7903, r21008

⚠ CAUTION
The properties of the runtime groups must not be changed during operation as this could result in discontinuous signal transitions.

Note

For value = 1 ... 256 (free runtime group):
This selection value can only be selected online if the following applies for sampling time T_sample of this runtime 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.
For value > 2000 (fixed runtime group):
The fixed runtime 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 runtime group with a sampling time >= 1 ms should be selected. When selecting the fixed runtime 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.

For value = 4002, 4003, 4005 (IF2 runtime group):
On devices where IF2 does not exist (D4xx, CU310), when selecting the runtime groups that involve IF2, the corresponding runtime group for IF1 is automatically logged on.

p21000[0...9] Runtime group properties / RTG property

DCC

Can be changed: T	**Calculated:** -	**Access level:** 1
Data type: Integer16	**Dynamic index:** -	**Function diagram:** -
P-Group: -	**Unit group:** -	**Unit selection:** -
Not for motor type: -	**Scaling:** -	**Expert list:** 1
Min:	**Max:**	**Factory setting:**
0	4005	0

Description: Allocates properties to runtime 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 runtime group:
- p21000[0] is used to set the property of the runtime group 1
...
- p21000[9] is used to set the property of the runtime group 10

Value:

0:	Do not calculate runtime 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

Recommendation: On the drive objects of CU, TB30, TM15DI_DO, TM31, TM41, TM120, the sampling time for supplementary functions is preset to p0115[0] = 4 ms. If you wish to configure a DCC runtime group with a shorter sampling time on these drive objects, then you should first set the sampling time for supplementary functions p0115[0] on this drive object to the value of the shortest sampling time required.

Index: [0] = Runtime group 1
[1] = Runtime group 2
[2] = Runtime group 3
[3] = Runtime group 4
[4] = Runtime group 5
[5] = Runtime group 6
[6] = Runtime group 7
[7] = Runtime group 8
[8] = Runtime group 9
[9] = Runtime group 10

Dependency: See also: r7903, r21008

⚠ CAUTION
The properties of the runtime groups must not be changed during operation as this could result in discontinuous signal transitions.

Note

For value = 1 ... 256 (free runtime group):

This selection value can only be selected online if the following applies for sampling time T_sample of this runtime 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.

For value > 2000 (fixed runtime group):

The fixed runtime 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 runtime group with a sampling time >= 1 ms should be selected. When selecting the fixed runtime 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.

For value = 4002, 4003, 4005 (IF2 runtime group):

On devices where IF2 does not exist (D4xx, CU310), when selecting the runtime groups that involve IF2, the corresponding runtime group for IF1 is automatically logged on.

p21000[0...9] Runtime group properties / RTG property

DCC			
	Can be changed: T	**Calculated:** -	**Access level:** 1
	Data type: Integer16	**Dynamic index:** -	**Function diagram:** -
	P-Group: -	**Unit group:** -	**Unit selection:** -
	Not for motor type: -	**Scaling:** -	**Expert list:** 1
	Min:	**Max:**	**Factory setting:**
	0	4005	0

Description: Allocates properties to runtime 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 runtime group:

- p21000[0] is used to set the property of the runtime group 1

...

- p21000[9] is used to set the property of the runtime group 10

Value:

0:	Do not calculate runtime 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
	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

Recommendation: On the drive objects of CU, TB30, TM15DI_DO, TM31, TM41, TM120, the sampling time for supplementary functions is preset to p0115[0] = 4 ms. If you wish to configure a DCC runtime group with a shorter sampling time on these drive objects, then you should first set the sampling time for supplementary functions p0115[0] on this drive object to the value of the shortest sampling time required.

Index:
[0] = Runtime group 1
[1] = Runtime group 2
[2] = Runtime group 3
[3] = Runtime group 4
[4] = Runtime group 5
[5] = Runtime group 6
[6] = Runtime group 7
[7] = Runtime group 8
[8] = Runtime group 9
[9] = Runtime group 10

Dependency: See also: r7903, r21008

⚠ CAUTION
The properties of the runtime groups must not be changed during operation as this could result in discontinuous signal transitions.

Note

For value = 1 ... 256 (free runtime group):

This selection value can only be selected online if the following applies for sampling time T_sample of this runtime 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.

For value > 2000 (fixed runtime group):

The fixed runtime 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 runtime group with a sampling time >= 1 ms should be selected. When selecting the fixed runtime 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] Runtime group properties / RTG property

DCC

Can be changed: T	**Calculated:** -	**Access level:** 1
Data type: Integer16	**Dynamic index:** -	**Function diagram:** -
P-Group: -	**Unit group:** -	**Unit selection:** -
Not for motor type: -	**Scaling:** -	**Expert list:** 1
Min:	**Max:**	**Factory setting:**
0	4004	0

Description: Allocates properties to runtime 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 runtime group:

- p21000[0] is used to set the property of the runtime group 1

...

- p21000[9] is used to set the property of the runtime group 10

Value:

0:	Do not calculate runtime 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

Recommendation: On the drive objects of CU, TB30, TM15DI_DO, TM31, TM41, TM120, the sampling time for supplementary functions is preset to p0115[0] = 4 ms. If you wish to configure a DCC runtime group with a shorter sampling time on these drive objects, then you should first set the sampling time for supplementary functions p0115[0] on this drive object to the value of the shortest sampling time required.

Index: [0] = Runtime group 1
[1] = Runtime group 2
[2] = Runtime group 3
[3] = Runtime group 4
[4] = Runtime group 5
[5] = Runtime group 6
[6] = Runtime group 7
[7] = Runtime group 8
[8] = Runtime group 9
[9] = Runtime group 10

Dependency: See also: r7903, r21008

⚠ CAUTION

The properties of the runtime groups must not be changed during operation as this could result in discontinuous signal transitions.

Note

For value = 1 ... 256 (free runtime group):

This selection value can only be selected online if the following applies for sampling time T_sample of this runtime 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.

For value > 2000 (fixed runtime group):

The fixed runtime 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 runtime group with a sampling time >= 1 ms should be selected. When selecting the fixed runtime 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] Runtime group properties / RTG property

DCC	**Can be changed:** T	**Calculated:** -	**Access level:** 1
	Data type: Integer16	**Dynamic index:** -	**Function diagram:** -
	P-Group: -	**Unit group:** -	**Unit selection:** -
	Not for motor type: -	**Scaling:** -	**Expert list:** 1
	Min:	**Max:**	**Factory setting:**
	0	4004	0

Description: Allocates properties to runtime 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 runtime group:

- p21000[0] is used to set the property of the runtime group 1

...

- p21000[9] is used to set the property of the runtime group 10

Value:

0:	Do not calculate runtime 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

Recommendation: On the drive objects of CU, TB30, TM15DI_DO, TM31, TM41, TM120, the sampling time for supplementary functions is preset to p0115[0] = 4 ms. If you wish to configure a DCC runtime group with a shorter sampling time on these drive objects, then you should first set the sampling time for supplementary functions p0115[0] on this drive object to the value of the shortest sampling time required.

Index:
[0] = Runtime group 1
[1] = Runtime group 2
[2] = Runtime group 3
[3] = Runtime group 4
[4] = Runtime group 5
[5] = Runtime group 6
[6] = Runtime group 7
[7] = Runtime group 8
[8] = Runtime group 9
[9] = Runtime group 10

Dependency: See also: r7903, r21008

> ⚠ **CAUTION**
>
> The properties of the runtime groups must not be changed during operation as this could result in discontinuous signal transitions.

Note

For value = 1 ... 256 (free runtime group):

This selection value can only be selected online if the following applies for sampling time T_sample of this runtime 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.

For value > 2000 (fixed runtime group):

The fixed runtime 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 runtime group with a sampling time >= 1 ms should be selected. When selecting the fixed runtime 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] Runtime group properties / RTG property

DCC

Can be changed: T	**Calculated:** -	**Access level:** 1
Data type: Integer16	**Dynamic index:** -	**Function diagram:** -
P-Group: -	**Unit group:** -	**Unit selection:** -
Not for motor type: -	**Scaling:** -	**Expert list:** 1
Min:	**Max:**	**Factory setting:**
0	4004	0

Description: Allocates properties to runtime 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 runtime group:

- p21000[0] is used to set the property of the runtime group 1

...

- p21000[9] is used to set the property of the runtime group 10

Value:

0:	Do not calculate runtime 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

Recommendation: On the drive objects of CU, TB30, TM15DI_DO, TM31, TM41, TM120, the sampling time for supplementary functions is preset to p0115[0] = 4 ms. If you wish to configure a DCC runtime group with a shorter sampling time on these drive objects, then you should first set the sampling time for supplementary functions p0115[0] on this drive object to the value of the shortest sampling time required.

Index:
[0] = Runtime group 1
[1] = Runtime group 2
[2] = Runtime group 3
[3] = Runtime group 4
[4] = Runtime group 5
[5] = Runtime group 6
[6] = Runtime group 7
[7] = Runtime group 8
[8] = Runtime group 9
[9] = Runtime group 10

Dependency: See also: r7903, r21008

> ⚠ **CAUTION**
>
> The properties of the runtime groups must not be changed during operation as this could result in discontinuous signal transitions.

Note

For value = 1 ... 256 (free runtime group):

This selection value can only be selected online if the following applies for sampling time T_sample of this runtime 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.

For value > 2000 (fixed runtime group):

The fixed runtime 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 runtime group with a sampling time >= 1 ms should be selected. When selecting the fixed runtime 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] Runtime group properties / RTG property

DCC

Can be changed: T	**Calculated:** -	**Access level:** 1
Data type: Integer16	**Dynamic index:** -	**Function diagram:** -
P-Group: -	**Unit group:** -	**Unit selection:** -
Not for motor type: -	**Scaling:** -	**Expert list:** 1
Min:	**Max:**	**Factory setting:**
0	4004	0

Description: Allocates properties to runtime 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 runtime group:

- p21000[0] is used to set the property of the runtime group 1

...

- p21000[9] is used to set the property of the runtime group 10

Value:

0:	Do not calculate runtime 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

Recommendation: On the drive objects of CU, TB30, TM15DI_DO, TM31, TM41, TM120, the sampling time for supplementary functions is preset to p0115[0] = 4 ms. If you wish to configure a DCC runtime group with a shorter sampling time on these drive objects, then you should first set the sampling time for supplementary functions p0115[0] on this drive object to the value of the shortest sampling time required.

Index:
[0] = Runtime group 1
[1] = Runtime group 2
[2] = Runtime group 3
[3] = Runtime group 4
[4] = Runtime group 5
[5] = Runtime group 6
[6] = Runtime group 7
[7] = Runtime group 8
[8] = Runtime group 9
[9] = Runtime group 10

Dependency: See also: r7903, r21008

⚠ CAUTION

The properties of the runtime groups must not be changed during operation as this could result in discontinuous signal transitions.

Note

For value = 1 ... 256 (free runtime group):

This selection value can only be selected online if the following applies for sampling time T_sample of this runtime 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.

For value > 2000 (fixed runtime group):

The fixed runtime 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 runtime group with a sampling time >= 1 ms should be selected. When selecting the fixed runtime 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.

r21001[0...9] Runtime group sampling time / RTG sampling time

All objects

Can be changed: -	**Calculated:** -	**Access level:** 1
Data type: FloatingPoint32	**Dynamic index:** -	**Function diagram:** -
P-Group: -	**Unit group:** -	**Unit selection:** -
Not for motor type: -	**Scaling:** -	**Expert list:** 1
Min:	**Max:**	**Factory setting:**
- [ms]	- [ms]	- [ms]

Description: Displays the current sampling time of the runtime groups.

Index:
[0] = Runtime group 1
[1] = Runtime group 2
[2] = Runtime group 3
[3] = Runtime group 4
[4] = Runtime group 5
[5] = Runtime group 6
[6] = Runtime group 7
[7] = Runtime group 8
[8] = Runtime group 9
[9] = Runtime group 10

r21002 Basis sampling time, hardware / Basis samp time HW

All objects

Can be changed: -	**Calculated:** -	**Access level:** 1
Data type: FloatingPoint32	**Dynamic index:** -	**Function diagram:** -
P-Group: -	**Unit group:** -	**Unit selection:** -
Not for motor type: -	**Scaling:** -	**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

Can be changed: -	**Calculated:** -	**Access level:** 1
Data type: FloatingPoint32	**Dynamic index:** -	**Function diagram:** -
P-Group: -	**Unit group:** -	**Unit selection:** -
Not for motor type: -	**Scaling:** -	**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

Dependency: Ensure that the basis sampling time on the SIMOTION D410 for the software time slices is always the same as the configured PROFIBUS/PROFINET clock cycle.

r21005[0...9] Computing time load of the runtime group / RTG load

All objects

Can be changed: -	**Calculated:** -	**Access level:** 3
Data type: FloatingPoint32	**Dynamic index:** -	**Function diagram:** -
P-Group: -	**Unit group:** -	**Unit selection:** -
Not for motor type: -	**Scaling:** -	**Expert list:** 1
Min:	**Max:**	**Factory setting:**
- [%]	- [%]	- [%]

Description: Share of the computing time load with which the DCC runtime group contributes to the utilization of the sampling time during which it is called.

Index: [0] = Runtime group 1
[1] = Runtime group 2
[2] = Runtime group 3
[3] = Runtime group 4
[4] = Runtime group 5
[5] = Runtime group 6
[6] = Runtime group 7
[7] = Runtime group 8
[8] = Runtime group 9
[9] = Runtime group 10

Note

The computing time load can only be displayed for the runtime 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...31] Hardware sampling times available / HW t_samp

All objects

Can be changed: -	**Calculated:** -	**Access level:** 3
Data type: FloatingPoint32	**Dynamic index:** -	**Function diagram:** -
P-Group: -	**Unit group:** -	**Unit selection:** -
Not for motor type: -	**Scaling:** -	**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
[25] = Hardware 26
[26] = Hardware 27
[27] = Hardware 28
[28] = Hardware 29
[29] = Hardware 30
[30] = Hardware 31
[31] = Hardware 32

Dependency:
See also: r7903, p21000
See also: F51001

NOTICE
For internal purposes, the drive unit always requires several 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...31] may be provided from the Control Unit. When selecting in this state, if a runtime group with a sampling time < r21003 (p21000 <= 255) is selected in p21000, only runtime groups whose sampling time is already provided in r21008[0...31] may be selected.

Note

A sampling time that is provided can be simultaneously used by system functions, several FBLOCK runtime groups and several DCC runtime groups.

The sampling time of runtime groups that have been assigned to the PROFIBUS runtime 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 requires several 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.

p21030 Runtime group, computing time measurement / RTG comp_ti_meas

All objects

Can be changed: T, U	**Calculated:** -	**Access level:** 4
Data type: Unsigned16	**Dynamic index:** -	**Function diagram:** -
P-Group: -	**Unit group:** -	**Unit selection:** -
Not for motor type: -	**Scaling:** -	**Expert list:** 1
Min:	**Max:**	**Factory setting:**
0	65535	0

Description: Only for internal Siemens service purposes.

Dependency: See also: p21032, r21035, r21036, r21037

p21031 Computing time measurement, blocks / Comp_ti_meas block

All objects

Can be changed: T, U	**Calculated:** -	**Access level:** 4
Data type: Unsigned32	**Dynamic index:** -	**Function diagram:** -
P-Group: -	**Unit group:** -	**Unit selection:** -
Not for motor type: -	**Scaling:** -	**Expert list:** 1
Min:	**Max:**	**Factory setting:**
0	4294967295	0

Description: Only for internal Siemens service purposes.

p21032 Computing time measurement, duration / Comp_ti_meas dur

All objects

Can be changed: T, U	**Calculated:** -	**Access level:** 4
Data type: Unsigned16	**Dynamic index:** -	**Function diagram:** -
P-Group: -	**Unit group:** -	**Unit selection:** -
Not for motor type: -	**Scaling:** -	**Expert list:** 1
Min:	**Max:**	**Factory setting:**
60 [s]	10000 [s]	60 [s]

Description: Only for internal Siemens service purposes.

Dependency: See also: p21030, r21035, r21036, r21037

p21033 Computing time measurement, number of individual measurements / Comp_ti_meas qty

All objects

Can be changed: T, U	**Calculated:** -	**Access level:** 4
Data type: Unsigned32	**Dynamic index:** -	**Function diagram:** -
P-Group: -	**Unit group:** -	**Unit selection:** -
Not for motor type: -	**Scaling:** -	**Expert list:** 1
Min:	**Max:**	**Factory setting:**
1	4294967295	10000

Description: Setting for the number of calls during the measurement of the individual blocks.

Dependency: See also: p21031

r21035[0...9] Computing time, minimum value / Computing time min

All objects

Can be changed: -	**Calculated:** -	**Access level:** 4
Data type: FloatingPoint32	**Dynamic index:** -	**Function diagram:** -
P-Group: -	**Unit group:** -	**Unit selection:** -
Not for motor type: -	**Scaling:** -	**Expert list:** 1
Min:	**Max:**	**Factory setting:**
- [µs]	- [µs]	- [µs]

Description: Only for internal Siemens service purposes.

Index: [0] = Runtime group 1
[1] = Runtime group 2
[2] = Runtime group 3
[3] = Runtime group 4
[4] = Runtime group 5
[5] = Runtime group 6
[6] = Runtime group 7
[7] = Runtime group 8
[8] = Runtime group 9
[9] = Runtime group 10

Dependency: See also: p21030, p21032, r21036, r21037

r21036[0...9] Computing time, mean value / Computing tim av

All objects

Can be changed: -	**Calculated:** -	**Access level:** 4
Data type: FloatingPoint32	**Dynamic index:** -	**Function diagram:** -
P-Group: -	**Unit group:** -	**Unit selection:** -
Not for motor type: -	**Scaling:** -	**Expert list:** 1
Min:	**Max:**	**Factory setting:**
- [µs]	- [µs]	- [µs]

Description: Only for internal Siemens service purposes.

Index: [0] = Runtime group 1
[1] = Runtime group 2
[2] = Runtime group 3
[3] = Runtime group 4
[4] = Runtime group 5
[5] = Runtime group 6
[6] = Runtime group 7
[7] = Runtime group 8
[8] = Runtime group 9
[9] = Runtime group 10

r21037[0...9] Computing time, maximum value / Computing time max

All objects

Can be changed: -	**Calculated:** -	**Access level:** 4
Data type: FloatingPoint32	**Dynamic index:** -	**Function diagram:** -
P-Group: -	**Unit group:** -	**Unit selection:** -
Not for motor type: -	**Scaling:** -	**Expert list:** 1
Min:	**Max:**	**Factory setting:**
- [µs]	- [µs]	- [µs]

Description: Only for internal Siemens service purposes.

Index: [0] = Runtime group 1
[1] = Runtime group 2
[2] = Runtime group 3
[3] = Runtime group 4
[4] = Runtime group 5
[5] = Runtime group 6
[6] = Runtime group 7
[7] = Runtime group 8
[8] = Runtime group 9
[9] = Runtime group 10

Dependency: See also: p21030, p21032, r21035, r21036

r21041[0...49] Block ID of the measured block / Block ID

All objects	**Can be changed:** -	**Calculated:** -	**Access level:** 4
	Data type: Unsigned16	**Dynamic index:** -	**Function diagram:** -
	P-Group: -	**Unit group:** -	**Unit selection:** -
	Not for motor type: -	**Scaling:** -	**Expert list:** 1
	Min:	**Max:**	**Factory setting:**
	-	-	-

Description: Block ID of the measured block (block runtime 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

Can be changed: -	**Calculated:** -	**Access level:** 4
Data type: Unsigned16	**Dynamic index:** -	**Function diagram:** -
P-Group: -	**Unit group:** -	**Unit selection:** -
Not for motor type: -	**Scaling:** -	**Expert list:** 1
Min:	**Max:**	**Factory setting:**
-	-	-

Description: In the block runtime measurements, the block runtimes are measured. R21039 indicates whether the measurement is the first or a subsequent call.

If the block type occurs only once in the runtime 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 runtime in us / Computing time min

All objects	**Can be changed:** -	**Calculated:** -	**Access level:** 4
	Data type: FloatingPoint32	**Dynamic index:** -	**Function diagram:** -
	P-Group: -	**Unit group:** -	**Unit selection:** -
	Not for motor type: -	**Scaling:** -	**Expert list:** 1
	Min:	**Max:**	**Factory setting:**
	- [μs]	- [μs]	- [μs]

Description: Minimum measured runtime of the measured block (block runtime 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 runtime in us / Computing tim av

All objects	**Can be changed:** -	**Calculated:** -	**Access level:** 4
	Data type: FloatingPoint32	**Dynamic index:** -	**Function diagram:** -
	P-Group: -	**Unit group:** -	**Unit selection:** -
	Not for motor type: -	**Scaling:** -	**Expert list:** 1
	Min:	**Max:**	**Factory setting:**
	- [μs]	- [μs]	- [μs]

Description: Average measured runtime of the measured block (block runtime 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 runtime in us / Computing time max

All objects	Can be changed: -	Calculated: -	Access level: 4
	Data type: FloatingPoint32	Dynamic index: -	Function diagram: -
	P-Group: -	Unit group: -	Unit selection: -
	Not for motor type: -	Scaling: -	Expert list: 1
	Min:	Max:	Factory setting:
	- [µs]	- [µs]	- [µs]

Description: Average measured runtime of the measured block (block runtime 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	**Can be changed:** -	**Calculated:** -	**Access level:** 4
	Data type: Unsigned32	**Dynamic index:** -	**Function diagram:** -
	P-Group: -	**Unit group:** -	**Unit selection:** -
	Not for motor type: -	**Scaling:** -	**Expert list:** 1
	Min:	**Max:**	**Factory setting:**
	-	-	-

Description: Library ID of the measured block (block runtime measurement via parameter p21031).
Measurements with blocks from different libraries can thereby be carried out in one runtime 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

p21047 Consistency entries number / Cons_Entr_No

All objects	**Can be changed:** T	**Calculated:** -	**Access level:** 4
	Data type: Unsigned16	**Dynamic index:** -	**Function diagram:** -
	P-Group: -	**Unit group:** -	**Unit selection:** -
	Not for motor type: -	**Scaling:** -	**Expert list:** 0
	Min:	**Max:**	**Factory setting:**
	2	65535	2

Description: ES enters the dynamic parameter number.

p21048[0...n] Consistency data / Cons_Data

All objects	**Can be changed:** T, U	**Calculated:** -	**Access level:** 4
	Data type: Unsigned32	**Dynamic index:** p21047	**Function diagram:** -
	P-Group: -	**Unit group:** -	**Unit selection:** -
	Not for motor type: -	**Scaling:** -	**Expert list:** 0
	Min:	**Max:**	**Factory setting:**
	0	4294967295	0

Dependency: Dynamic index for fingerprint parameter.

Index

S

T

U

W

X