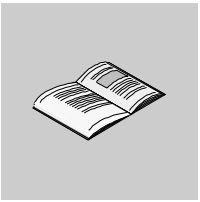
Concept
IEC Block Library
Part: ANA_IO

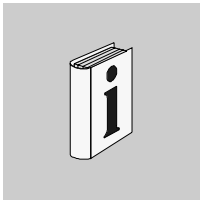
840 USE 494 00 eng Version 2.5

33001606.01

Table of Contents



About the book



At a Glance

Document Scope

This documentation is designed to help with the configuration of functions and function blocks.

Validity Note

This documentation applies to Concept 2.5 under Microsoft Windows 98, Microsoft Windows 2000 and Microsoft Windows NT 4.x.

Note: There is additional up to date tips in the README data file in Concept.

Related Documents

Title of Documentation	Reference Number
Concept Installation Instructions	840 USE 492 00
Concept User Manual	840 USE 493 00
Concept EFB User Manual	840 USE 495 00
Concept LL984 Block Library	840 USE 496 00

User Comments

We welcome your comments about this document. You can reach us by e-mail at TECHCOMM@modicon.com

General information on the ANA_IO block library

I

Overview

At a Glance

This section contains general information on the ANA_IO block library.

What's in this part?

This Part contains the following Chapters:

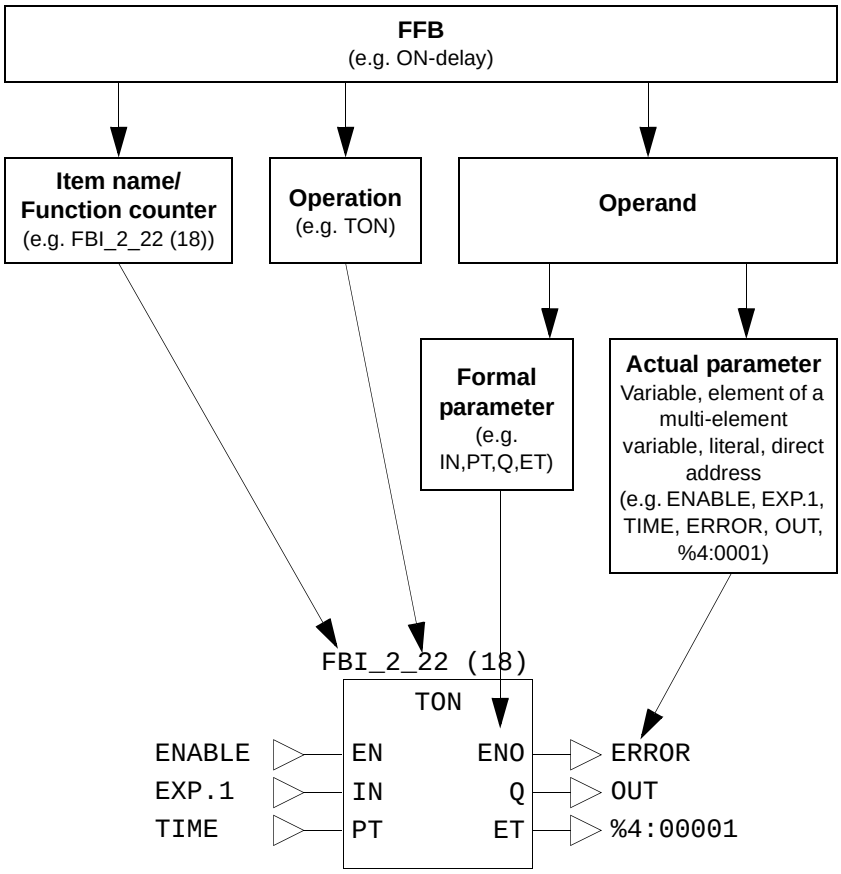
Parameterizing functions and function blocks

1

Parameterizing functions and function blocks

General

Each FFB consists of an operation, the operands needed for the operation and an instance name or function counter.



Operation

The operation determines which function is to be executed with the FFB, e.g. shift register, conversion operations.

Operand

The operand specifies what the operation is to be executed with. With FFBs, this consists of formal and actual parameters.

Formal/actual parameters

The formal parameter holds the place for an operand. During parameterization, an actual parameter is assigned to the formal parameter.

The actual parameter can be a variable, a multi-element variable, an element of a multi-element variable, a literal or a direct address.

Conditional/ unconditional calls

"Unconditional" or "conditional" calls are possible with each FFB. The condition is realized by pre-linking the input EN.

- Displayed EN
  conditional calls (the FFB is only processed if EN = 1)
- EN not displayed
  unconditional calls (FFB is always processed)

Note: If the EN input is not parameterized, it must be disabled. Any input pin that is not parameterized is automatically assigned a "0" value. Therefore, the FFB should never be processed.

Calling functions and function blocks in IL and ST

Information on calling functions and function blocks in IL (Instruction List) and ST (Structured Text) can be found in the relevant chapters of the user manual.

Operating Analog Modules

2

Overview

At a Glance

This block library contains EFBs to operate the analog modules and EFBs to operate the INTERBUS on the Compact.

Note: This block library's EFBs are available in every IEC application. It is possible to use these platform specific EFBs on a PLC platform for which they were not intended (Quantum EFBs on a Compact PLC).

The EFBs for the analog modules can be found in the following groups:

- Quantum I/O Config
- Compact I/O Config
- Analog I/O Config
- Analog I/O Scaling
- Analog I/O Debug

What's in this Chapter?

This Chapter contains the following Maps:

Editing analog values

Introduction

This block library contains EFBs for the operation of analog modules. The EFBs are designed in such a way that it enables the FBD program configuration to be largely independent of the hardware module used. Hardware-dependent EFBs (e.g. Group: Quantum IO Config) are used to evaluate project-specific information on the PLC and to process it in the data structures ANL_IN und ANL_OUT. Hardware-independent EFBs operate with these data structures to read the raw values from the Input words (3x), scale them and convert them into REAL values. This means that changes in direct addresses or changes in input/output parameters can be detected automatically by the EFBs.

Division into sections

As the detection of configuration data only occurs once after loading, it is advisable to divide the EFBs in the ANA_I/O library into at least two sections.

Division into at least two sections is recommended.
- Scaling section
- Configuration section

This division into a configuration section and several scaling sections can lead to a reduction of the CPU load, as the configuration part (configuration section) must only execute once, (after a cold restart or a warm restart). As a rule, the scaling sections must be executed continuously.

Scaling and configuration sections

Scaling section

Scaling sections are used for actual analog value processing.

Configuration section

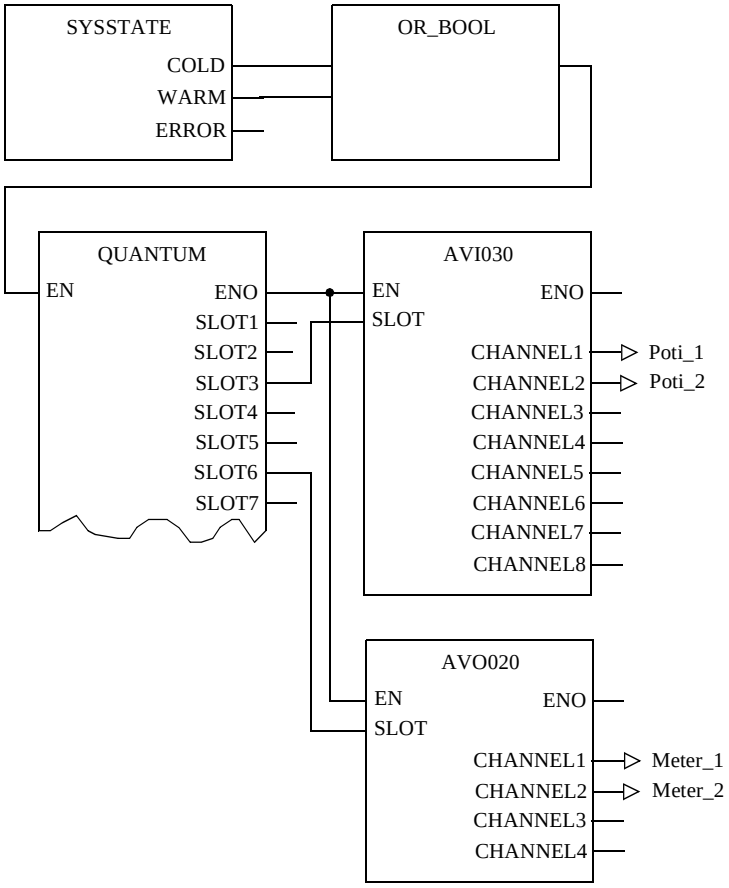
The configuration section is used to configure the analog I/O modules and controls the data exchange between analog EFBs, the state RAMand the configuration data. The configuration section should be called "CfgAnaIo" to ensure compatibility with future Concept versions.

Two options are available to control a configuration section:
- using the EN inputs of the individual EFBs
- by enabling or disabling the configuration section

Example 1 Control using the EN inputs

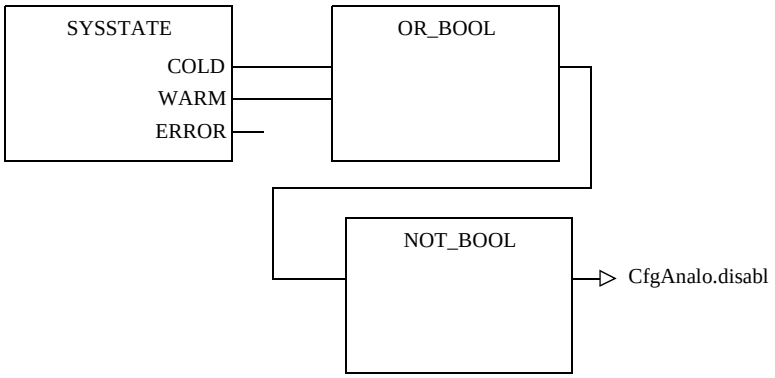
Control of the configuration section is possible through the EN inputs of this section's individual EFBs. The EFBs are enabled through the SYSSTATE EFB which has COLD or WARM outputs that are set to 1 for one cycle after either a cold or a warmstart.

Example of a configuration section "CfgAnaIo" for Quantum

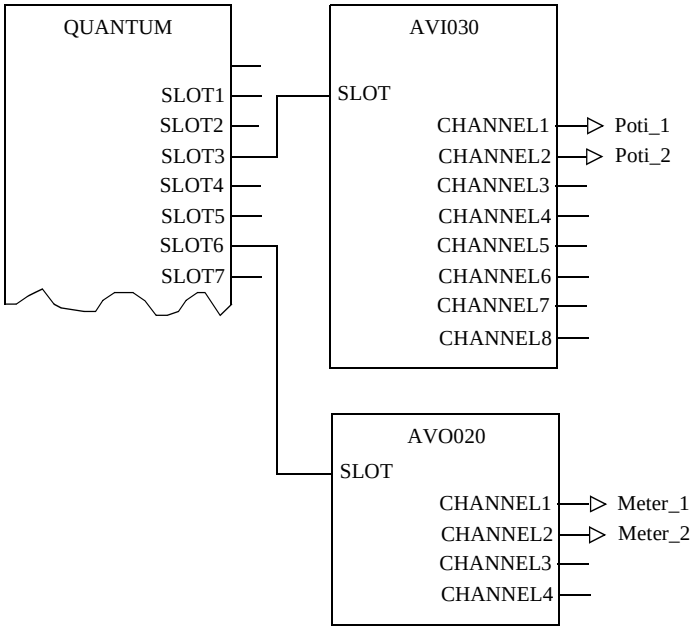


Example 2 Control by section enabling

The configuration section can be controlled by enabling and disabling this section. The configuration section is enabled in a section of its own through the SYSSTATE EFB which has COLD or WARM outputs that are set to 1 for one cycle after either a cold or a warmstart. This 1-signal is used for configuration section enable and disable. An EN and ENO linking of the EFBs is not required in this solution.
Example of a control section "Config_Ctrl" for Quantum



Example of a configuration section "CfgAnaIo" for Quantum



Configuration EFBs

At a Glance

Configuration data from analog input/output modules is accessed using EFBs from the block library groups "Quantum IO Config" and "Compact IO Config".
A difference is made in the following cases:

- *Procedure for only local I/O (Quantum/Compact), p. 21*
- *Procedure for expansion of the local backplane using XBE modules (Quantum), p. 23*
- *Procedure for distributed I/O (RIO, DIO) (Quantum), p. 24*
- *Procedure for expansion of the distributed backplane using XBE modules (Quantum), p. 25*

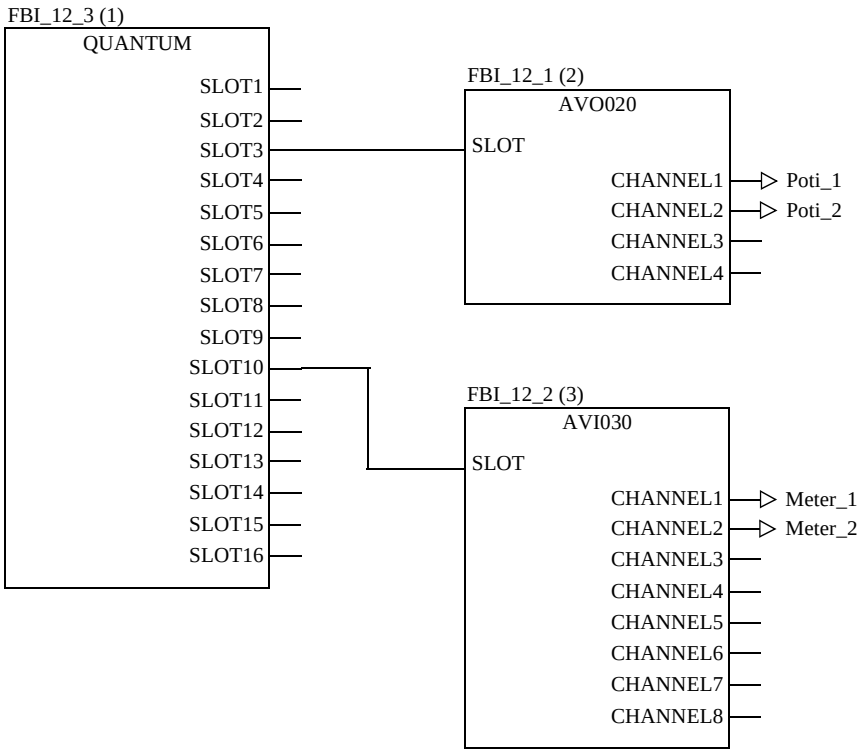
Procedure for only local I/O (Quantum/ Compact)

Place a single QUANTUM/COMPACT-EFB in the configuration section (CfgAnaIo). Each analog module has its own analog I/O EFB. Connect the desired analog I/O EFB to the corresponding slot on the QUANTUM/COMPACT EFB. The analog I/O EFB provides a variable of data type ANL_IN or ANL_OUTas an output. These values are further processed using the scaling EFBs in the scaling sections. To do this, they are connected with the relevant scaling EFBs using Unlocated variables .

Note: Do not specify Literals at the SLOT inputs of the configuration EFBs. SLOT inputs must be connected to SLOT outputs.

Example of only local I/O

Example of a configuration section "CfgAnaIo"



The EFB's mode of functioning can be found in the table below.

EFB	Function mode
Quantum	The EFB is used to edit the configuration data of a primary subrack for transfer by the analog in/out EFBs.
AVI030	Quantum module AVI 030 00 configuration. The EFB is used to edit the configuration data of the Quantum module AVI 030 00 for continued processing by the scaling EFBs. The 3x references specified in the I/O map are automatically assigned internally to the individual channels and can therefore only be occupied by Unlocated variables.
AVO020	Quantum module AVO 020 00 configuration. The EFB is used to edit the configuration data of the Quantum module AVO 020 00 for continued processing by the scaling EFBs. The 3x references and 4x references specified in the I/O map are automatically assigned internally to the individual channels and can therefore only be occupied by Unlocated variables.

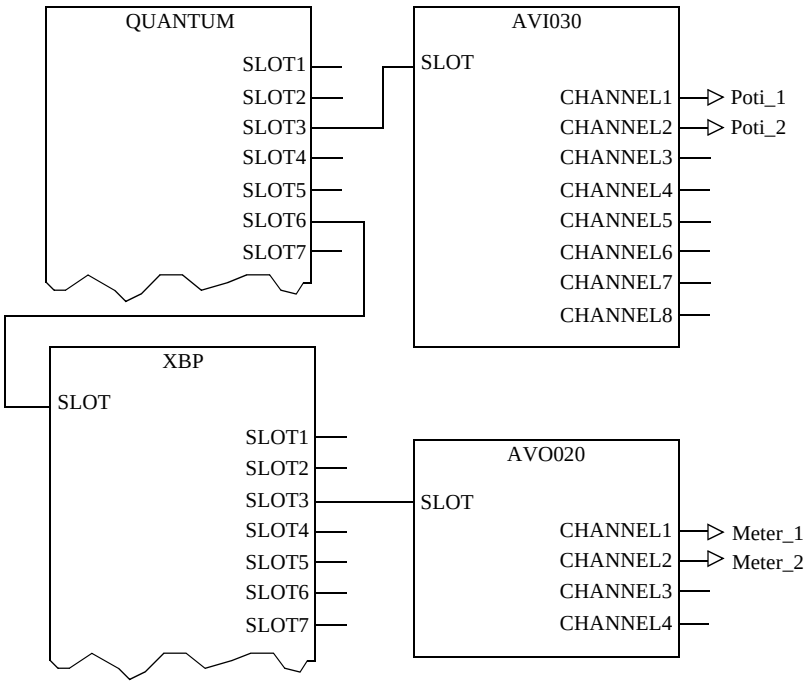
Procedure for expansion of the local backplane using XBE modules (Quantum)

Place a single QUANTUM EFB in the configuration section (CfgAnaIo).
The configuration data for the backplane expander is accessed using XBP EFBs.
Connect the XBP EFB to the XBE module slot.

Note: Do not specify Literals at the SLOT inputs of the configuration EFBs. SLOT inputs must be connected to SLOT outputs.

Example of expansion of the local backplane using XBE modules

Example of a configuration section "CfgAnaIo"



The EFB's mode of functioning can be found in the table below.

EFB	Function mode
Quantum	The EFB is used to edit the configuration data of a primary subrack for transfer by the analog in/out EFBs.
XBP	The EFB is used to prepare the backplane expander configuration data for transfer by the analog in/out EFBs.
AVI030	See *Example of only local I/O, p. 22*
AVO020	See *Example of only local I/O, p. 22*

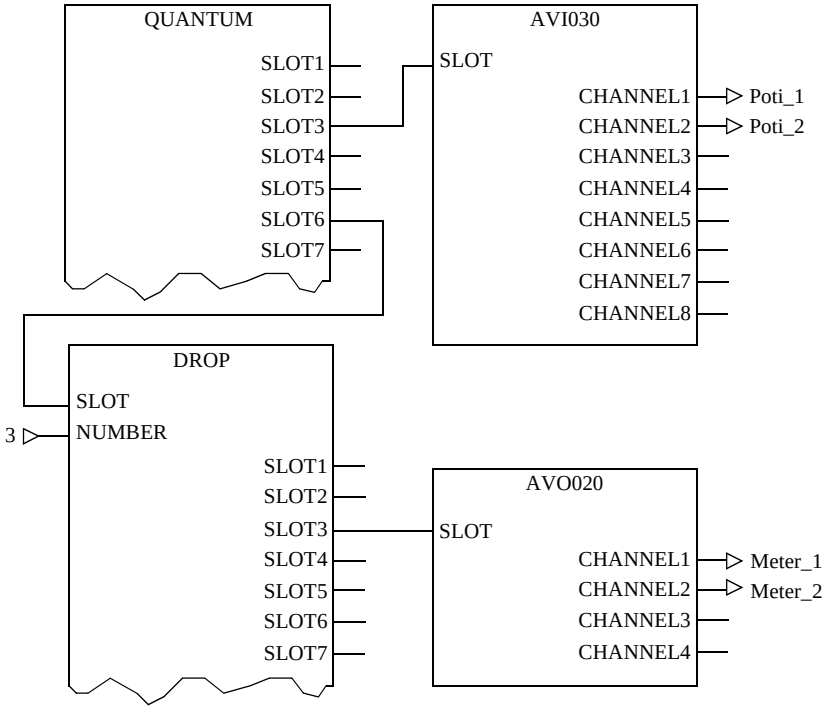
Procedure for distributed I/O (RIO, DIO) (Quantum)

Place a single QUANTUM EFB in the configuration section (CfgAnaIo).
The configuration data from remote I/O (RIO) and distributed I/O (DIO) or network option I/O (NOM) is accessed via DROP EFBs. The DROP EFB is applicable to all three I/O station types. If RIO is used, connect the DROP EFB to the slot on the RIO communication module. If DIO is used, connect the DROP EFB to the slot on the CPU or the NOM module. Each I/O station has its own address. Specify this number at the NUMBER input of the DROP EFB.

Note: Do not specify Literals at the SLOT inputs of the configuration EFBs. SLOT inputs must be connected to SLOT outputs.

Example of distributed I/O (RIO, DIO or NOM)

Example of a configuration section "CfgAnaIo"



The EFB's mode of functioning can be found in the table below.

EFB	Function mode
Quantum	The EFB is used to edit the configuration data of a primary subrack for transfer by the analog in/out EFBs.
Drop	The EFB is used to edit the configuration data of an I/O subrack for transfer by the analog in/out EFBs.
AVI030	See *Example of only local I/O, p. 22*
AVO020	See *Example of only local I/O, p. 22*

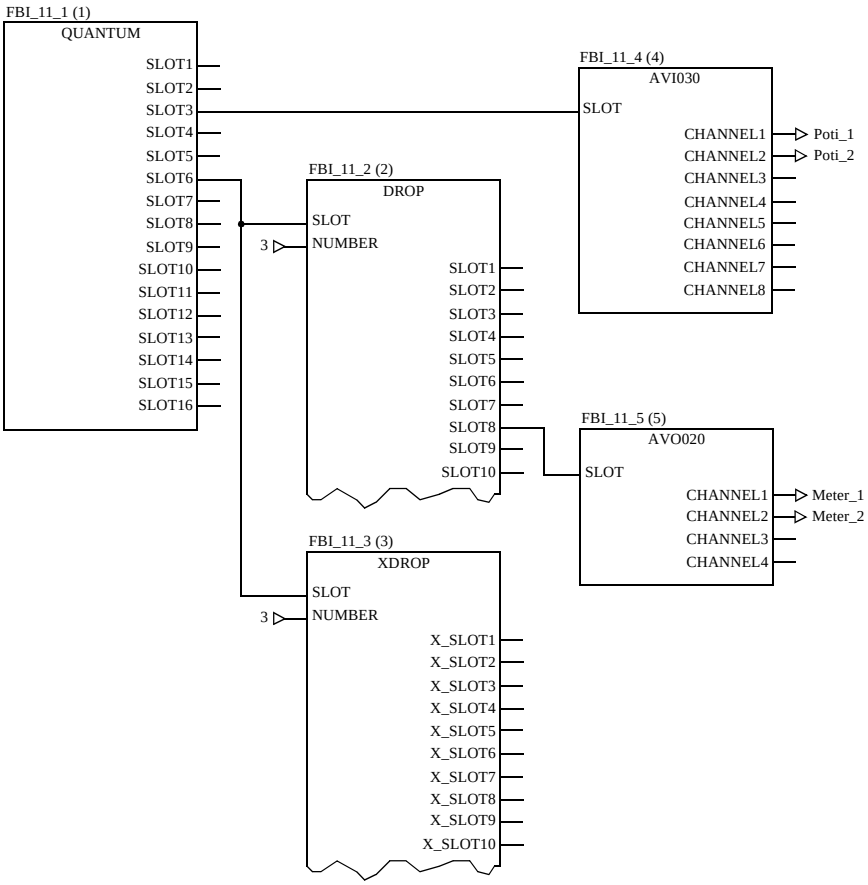
Procedure for expansion of the distributed backplane using XBE modules (Quantum)

Place a single QUANTUM EFB in the configuration section (CfgAnaIo).
The configuration data from remote I/O (RIO), distributed I/O (DIO) or network option I/O (NOM) is accessed via DROP EFBs. The DROP EFB is applicable to all three I/O station types. If RIO or NOM is used, connect the DROP EFB to the slot on the RIO communication module or the NOM module. If DIO is used, connect the DROP EFB to the slot on the CPU. Each I/O station has its own number. Specify this number at the NUMBER input of the DROP EFB.
The configuration data for the distributed backplane expander is accessed using XDROP EFBs. Connect the SLOT input of the XDROP EFB with the SLOT input of the DROP EFB. Enter the **same** number for the input NUMBER of the XDROP EFB as for the input NUMBER of the DROP EFB.

Note: Do not specify Literals at the SLOT inputs of the configuration EFBs. SLOT inputs must be connected to SLOT outputs.

Example of expansion of the distributed backplane using XBE modules

Example of a configuration section "CfgAnaIo"



The EFB's mode of functioning can be found in the table below.

EFB	Function mode
Quantum	The EFB is used to edit the configuration data of a primary subrack for transfer by the analog in/out EFBs.
XDROP	The EFB is used to prepare the distributed backplane expander configuration data for transfer by the analog in/out EFBs.
AVI030	See *Example of only local I/O, p. 22*
AVO020	See *Example of only local I/O, p. 22*

Scaling EFBs

At a Glance

The analog values are scaled using the EFBs of the "Analog IO Scaling" block library group in the scaling sections.
The analog I/O EFBs operate hardware-independent with the data types ANL_IN and ANL_OUT.

Scaling EFBs available

The following scaling EFBs are available:

- I_RAW, O_RAW:
  Raw value, no scaling
- I_RAWSIM:
  Raw value, simulation
- I_NORM, I_NORM_WARN, O_NORM, O_NORM_WARN:
  Standardization, representation in a range from 0.0 to 1.0
- I_PHYS, I_PHYS_WARN, O_PHYS, O_PHYS_WARN:
  Physical, physical range
- I_SCALE, I_SCALE_WARN, O_SCALE, O_SCALE_WARN:
  Scaled, representation in a user-defined range from MN to MX

Handy hints

Please take note of the following hints about using scaling EFBs:

- When using these EFBs the messages in **Online** → **Event viewer** should definitely be observed. That is where execution errors for these EFBs are recorded.
- For tasks not requiring the physical units and/or only scaling by 0-100% the NORM-EFBs, PHYS- or SCALE-EFBs are preferred.
- If scaling is required PHYS-EFBs should be used. PHYS-EFBs do not work without information about the physical units however, for these the SCALE-EFBs are used.
- SCALE-EFBs can also be used if physical scaling is not required or is not possible.
- SCALE-EFBs do not work in conjunction with input/output modules which provide direct physical values (e.g. decimal values). This applies, for example, to temperature or resistance modules not set to raw values.
- EFBs with "WARN" can not be used for all input/output modules. The descriptions of the individual EFBs explains which modules they can be used with.
- "Open circuit" is categorized as an error rather than a warning. "Open circuit" generates an online error message which can be accessed using **Online** → **Event viewer** and sets the output "ENO" of the "WARN"-EFBs to "0".
- The RAW-EFB is not usually required. It merely represents a simple way to make additional use of the raw values.

Debugging-EFBs

At a Glance

The data types ANL_IN and ANL_OUT are monitored by the EFBs of the "Analog I/O Debug" block library group in the scaling sections.
Ordinarily, the debug EFBs are not necessary. Although they can be used by system specialists for input/output diagnostics, e.g. to monitor the raw values in State RAM.

Scaling EFBs available

The following debug EFBs are available:

- I_DBSET,
- I_DEBUG,
- O_DBSET,
- O_DEBUG

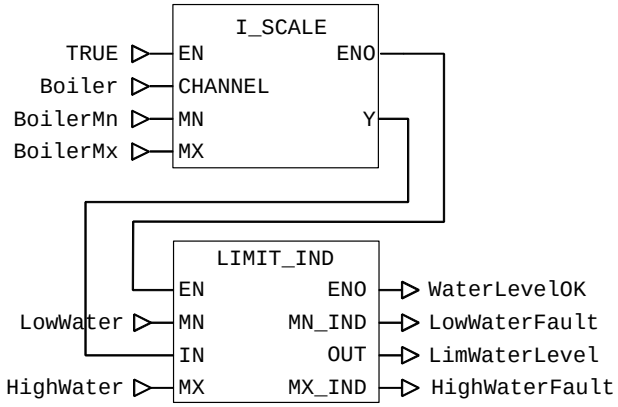
Application example for Quantum

At a Glance

To precisely monitor the output values, it is advisable to implement the scaling with two EFBs. The first EFB (scaling EFBs) scales the analog value and the second EFB monitors the scaled value for ranges preset by the process. In the following process, either the original Y output of the scaling EFB or the limited OUT output of the Limiter EFB can be used.

Application example

A simple example shows how the EFBs can be used.
The example assumes a boiler with a capacity of 350 liters. The input voltage ranges from 0.0 Volt for 0 liters to 10.0 Volt for 1000 liters. A PI controller should guarantee a volume between 200 and 300 liters. The Limiter EFB detects violations in this range and will limit the output.
Given values:
BoilerMn: 0
BoilerMx: 1 000
LowWater: 199
HighWater: 301
"Boiler" is an unlocated variable of the ANL_IN type and is linked to an AVI030-EFB.
Application example



Operating INTERBUS with Compact

3

Diagram of the INTERBUS structure with EFBs

Introduction

The IBS group in the ANA_IO block library contains all the Function blocks required for operating INTERBUS with TSX Compact for Concept PLC programming software. If the hardware is reconfigured (modules added or removed), the same modifications must be made to the PLC program as to the hardware. Modification of the addresses of all following modules is not necessary.

Note: The modules of the IBS group can also be set with the NOA 61100 and NOA 611 10.

Structure of an INTERBUS line

INTERBUS is constructed as a linear structure. This linear structure begins with the master and ends with an INTERBUS module. The diagram illustrates the INTERBUS hardware structure.
Hardware structure

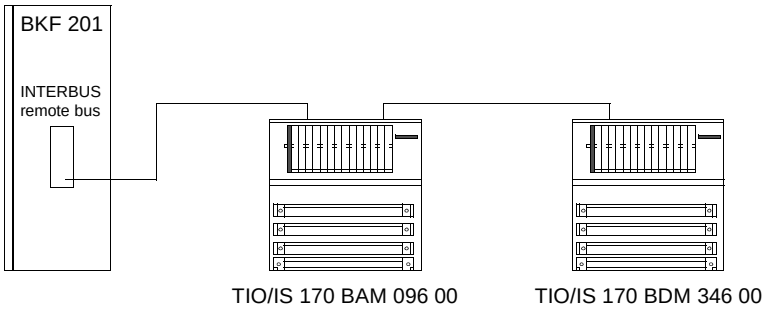
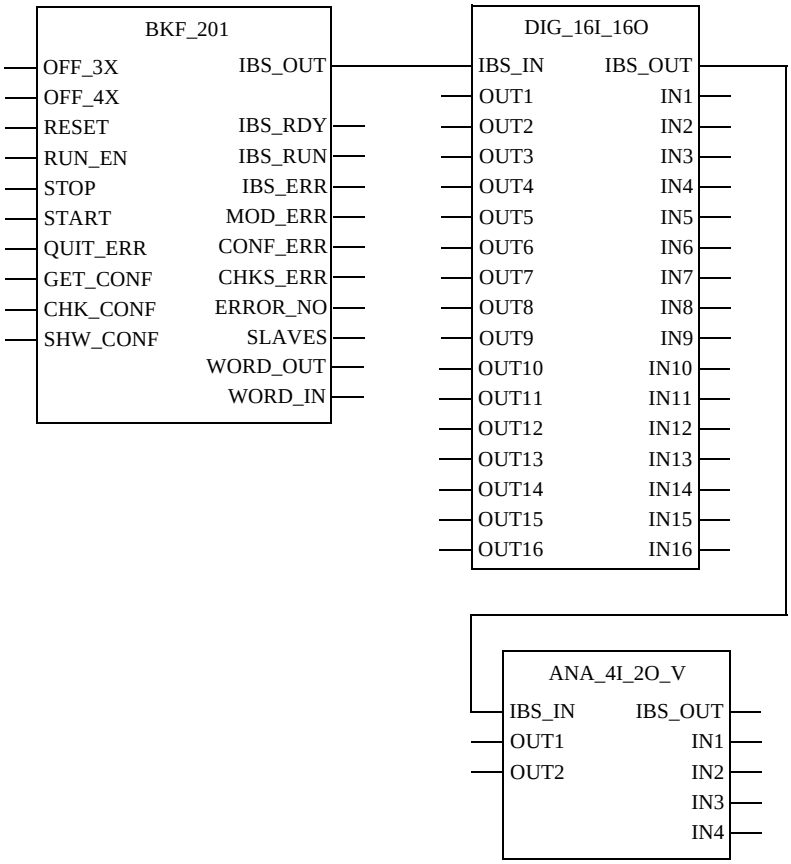


Diagram of the INTERBUS structure with EFBs

The diagram illustrates the same structure with the function blocks from the INTERBUS library.

Software structure with function blocks



Behavior of function blocks in simulator operation

IBS group function blocks in the ANA_IO library behave in the same way in simulator operation as with hardware connected. It should be noted, however, that the function block outputs labeled IN can only be assigned values if the outputs are assigned to a variable. A value can be assigned to the variables in the reference data editor. With hardware connected, no values can be assigned to the variables on the function block outputs.

Analog value processing with Momentum

4

At a Glance

Basic principle

Analog input data of I/O units from the Momentum product family are placed onto addresses with 3x references in State RAM. These are still raw values, although they were already linearized by the firmware of the I/O units. To make them available in the REAL format as voltage, current or temperature values, it is still necessary to first convert them in the user program. The required algorithms for this are in the user's guide "I/O Units for TSX Momentum".
The same procedure in reverse must be followed for the output values.
This conversion must always be done, regardless of the form in which data reaches State RAM (via Modbus Plus PeerCop, I/O Bus, ...).
For easier handling, Concept offers EFBs that perform this conversion.

Affected modules

This includes all analog I/O units and TIO modules.

What's in this Chapter?

This Chapter contains the following Maps:

Procedure for Analog value processing with Momentum

Concept EFBs

To calculate analog input or output data for Momentum special concept EFBs exist. The following table can be used to decide which EFB is necessary for which module:

Module	EFB
170 AAI 140 00	ANA_16I (See *ANA_16I: Configuring the module AAI 140 00, p. 89*)
170 AAI 520 40	ANA_4I_M (See *ANA_4I_M: Configuring the module AAI 520 40, p. 93*)
170 AAO 120 00	ANA_4O (See *ANA_4O: Configuring the module BAO 126 00, p. 117*)
170 AAO 921 00	ANA_4O (See *ANA_4O: Configuring the module BAO 126 00, p. 117*)
170 AMM 090 00	MIX_4I_2O (See *MIX_4I_2O: Configuring the AMM module 090 00, p. 241*)
170 BAI 036 00	ANA_8I (See *ANA_8I: Configuring the module AAI 030 00, BAI 036 00, p. 123*)
170 BAM 096 00	ANA_4I_2O (See *ANA_4I_2O: Configuring the TIO-module BAM 096 00, p. 101*) ANA_4I_2O_C (See *ANA_4I_2O_C: Configuring the TIO-module BAM 096 00, p. 109*) (current) ANA_4I_2O_V (See *ANA_4I_2O_V: Configuring the TIO-module BAM 096 00, p. 113*) (voltage)
170 BAO 126 00	ANA_4O (See *ANA_4O: Configuring the module BAO 126 00, p. 117*)

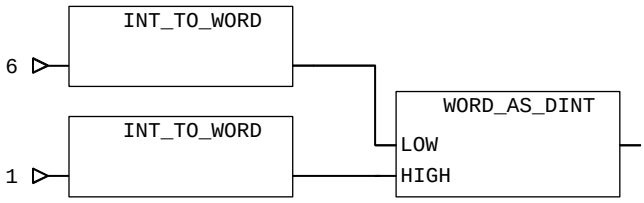
Procedure

Procedure for converting analog values

Step	Action
1	Incorporate the EFB that matches your module into your program.
2	Assign the parameters for the channels of your modules to the pins PM_OUT ... and PM_IN... (enter decimal values). The codes for the parameters are given in the user's guide "I/O Units for TSX Momentum". They must be consistent with the parameter assignment in the I/O map.
3	Assign the module address to the IBS_IN pin (data type DINT) (address in State RAM after which module data is given). The high word contains the offset of the first 3x address of the module, the low word the offset of the first 4x address. The parameter addresses have to be considered as well.

Examples

Example 1:
The following logic simplifies the value determination for IBS_IN, in the example here, a 170 AAI 140 00 with the addresses 300 001 .... 300 016 and 400 006 .... 400 021. For the AAI 140, the EFB ANA_16I (See *ANA_16I: Configuring the module AAI 140 00, p. 89*) will always occupy 16 output words! Consequently, enter the offset for the outputs on top and for the inputs on the bottom.
If your network has several I/O units, the storage of this logic in a DFB is recommended.
DFB logic



Example 2:
If using Peer Cop and placing the measuring values of a 170 AAI 140 onto the 300 001 .... 300 116 addresses, the same values can be entered with the consideration that the EFB does not only occupy the addresses 400 006 ... 400 009 for the I/O unit parameters, but internally also the addresses 400 010 ... 400 021.
Example 3:
If it is necessary to convert REAL values into raw values and have these on the addresses 400 101 …. 400 104 in order to place them onto the outputs of a 170 AAO 120, for instance via Peer Cop, the value entered on top must be 100 and the value on the bottom must be 0 (the EFB for the AAO does not occupy any input words). Please note, the EFB has to have the address 400 100 for the I/O unit parameters.

Result of the described procedure

The correct values of the modules are now present as data type REAL. They can be scaled, can be used for calculations or closed-loop control.

Note: Do not forget to divide the values by 4 when working with voltages. (Please refer to the EFB documentation as well).

Example analog value processing with Momentum

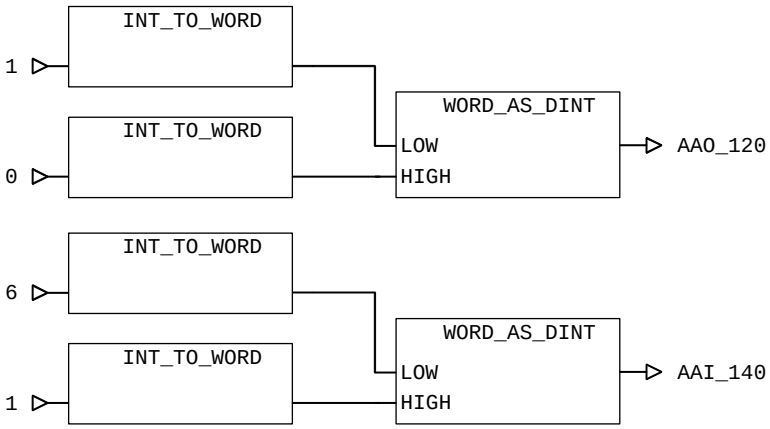
Task

There are two analog modules on the I/O bus:

- a 170 AAO 120 00, addressed to 400 001 400.005, disconnect behavior: all outputs to 0
- a 170 AAI 140 00, addressed to 300 001 300 016 and 400 006 400 021, channels 1 ... 4 are parametered for 4 … 20 mA, all other channels are disconnected.

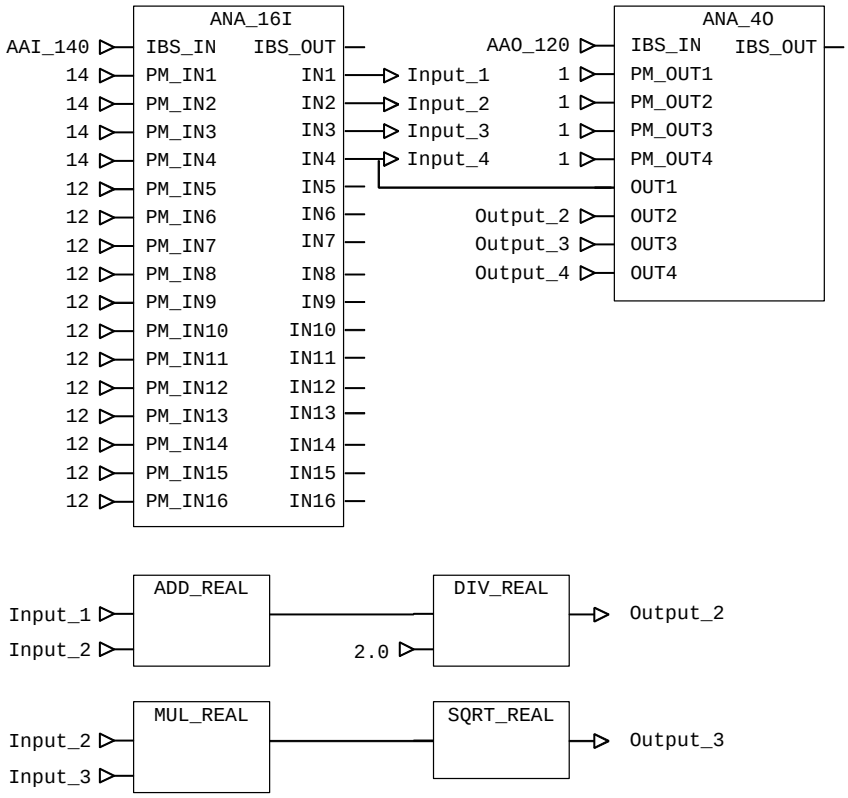
Addressing

The following network can be used for the addressing:



Network with analog value EFBs

The raw values in this example are in the 3x and 4x registers, the converted REAL values for the inputs are carried in the unlocated variables Input_1…Input_4 and for the outputs in Output_1 …Output_4.
The network with the analog value EFBs will look as follows:



Also note, the values can be easily used to continue computing or to be placed again directly onto an EFB for an output module.
If the module addresses are continuous and have no gaps, the pins IBS_OUT and IBS_IN from EFBs of two consecutive modules can be connected while only the first address has to be entered.

Note for digital Momentum I/O units

Concept also makes EFBs available for digital I/O units and modules. They convert the word references to bit references (and vice versa).
The following EFBs are available:

Module	EFB
170 ADI 340 00	DIG_16I (See *DIG_16I: Configuring the TIO-module BDI 346 00 / 546 50 / 746 50, p. 171*)
170 ADI 350 00	DIG_16I (See *DIG_16I: Configuring the TIO-module BDI 346 00 / 546 50 / 746 50, p. 171*) (2 EFBs connected in series, IBS_OUT connected with IBS_IN)
170 ADI 540 00	DIG_16I (See *DIG_16I: Configuring the TIO-module BDI 346 00 / 546 50 / 746 50, p. 171*)
170 ADM 350 x0	DIG_16I_16O (See *DIG_16I_16O: Configuring the TIO-module BDM 346 00, p. 181*)
170 ADM 370 10	DIG_16I_16O (See *DIG_16I_16O: Configuring the TIO-module BDM 346 00, p. 181*)
170 ADM 390 10	DIG_16I_12O_MON (See *DIG_16I_12O_MON: Configuring the module ADM 390 10, p. 175*)
170 ADM 390 30	DIG_16I_16O (See *DIG_16I_16O: Configuring the TIO-module BDM 346 00, p. 181*)
170 ADM 690 50	DIG_16I_16O (See *DIG_16I_16O: Configuring the TIO-module BDM 346 00, p. 181*)
170 ADO 340 00	DIG_16O (See *DIG_16I_16O: Configuring the TIO-module BDM 346 00, p. 181*)
170 ADO 350 00	DIG_16I (See *DIG_16I_16O: Configuring the TIO-module BDM 346 00, p. 181*) (2 EFBs connected in series, IBS_OUT connected with IBS_IN)
170 ADO 5x0 50	DIG_16O (See *DIG_16I_16O: Configuring the TIO-module BDM 346 00, p. 181*)
170 ADO 7x0 50	DIG_16O (See *DIG_16I_16O: Configuring the TIO-module BDM 346 00, p. 181*)

Note: Modules that have the same Ident code can use the same EFB.

EFB descriptions

II

Overview

At a Glance

These EFB descriptions are listed in alphabetical order.

What's in this part?

This Part contains the following Chapters:

ACI030: Configuring the Quantum module ACI 030 00

5

Overview

At a Glance

This chapter describes the block ACI030.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

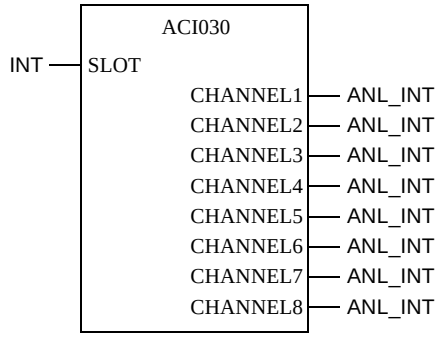
Function description

The Function block is used to edit the configuration data of an ACI 030 00 Quantum module for subsequent use by the scaling EFBs.
This module has 8 unipolar input channels for mixed voltage and current processing.
For the configuration of an ACI 030 00 the function block in the configuration section is connected to the corresponding slot output of the QUANTUM or DROP Function block. The 3x references specified in the I/O map are automatically assigned internally to the individual channels and can therefore only be occupied by Unlocated ariables.
The analog values can be further processed in Scaling Sections using the I_DEBUG, I_NORM, I_RAW and I_SCALE Function blocks.
EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
SLOT	INT	Slot
CHANNEL1	ANL_IN	Channel 1
CHANNEL2	ANL_IN	Channel 2
CHANNEL3	ANL_IN	Channel 3
CHANNEL4	ANL_IN	Channel 4
CHANNEL5	ANL_IN	Channel 5
CHANNEL6	ANL_IN	Channel 6
CHANNEL7	ANL_IN	Channel 7
CHANNEL8	ANL_IN	Channel 8

Runtime error

Runtime error

If no ACI 030 00 module has been configured for the specified SLOT input, an error message appears.
The status information "Open circuit or undervoltage on channel" can be collected via the status register defined in the I/O map.

ACI040: Configuring the Quantum module ACI 040 00

6

Overview

At a Glance

This chapter describes the block ACI040.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

The Function block is used to edit the configuration data of the ACI 040 00 Quantum module for subsequent use by the scaling EFBs.

The module has 16 channels which, according to requirement, can be used as differential or single inputs for the processing of current. When processing current the ranges are 0…20 mA, 0…25 mA and 4…20 mA.

For the configuration of an ACI 040 00 the function block in the configuration section is connected to the corresponding SLOT output of the QUANTUM or DROP Function block. The 3x references specified in the I/O map are assigned internally to individual channels automatically. The channels can only be occupied byUnlocated variables.

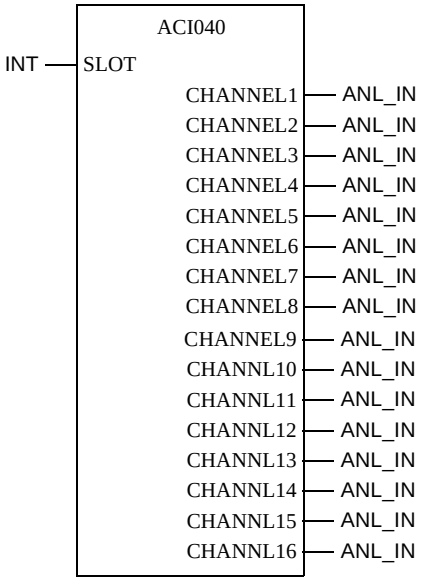
The analog values can be processed further in Scaling Sections using the I_NORM, I_PHYS, I_RAW and I_SCALE Function blocks. I_DEBUG and I_DBSET are also available for testing purposes.

EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
SLOT	INT	Slot
CHANNEL1	ANL_IN	Channel 1
:	:	:
CHANNEL9	ANL_IN	Channel 9
CHANNL10	ANL_IN	Channel 10
:	:	:
CHANNL16	ANL_IN	Channel 16

Runtime error

Runtime error

If no ACI 040 00 module has been configured for the specified SLOT input, an error message appears.

In the "4 to 20 mA" mode, the "open circuit on channel" status information is available. It can be collected via the 3 references of the module (3x+16) defined in the I/O mapping.

ACO020: Configuring the Quantum module ACO 020 00

7

Overview

At a Glance

This chapter describes the block ACO20.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

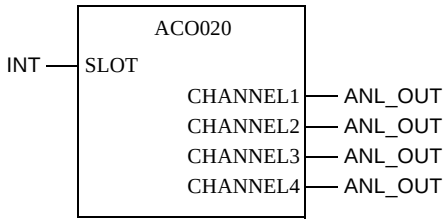
Function description

The Function block is used to edit the configuration data of an ACO 020 00 Quantum module for subsequent use by the scaling EFBs.
This module has 4 output channels for the processing of current in the range of 4 ... 20 mA.
For the configuration of an ACO 020 the function block in the configuration section is connected to the corresponding slot output of the QUANTUM or DROP Function block. The 4x references specified in the I/O map and the status information (if configured) are automatically assigned internally to the individual channels and can therefore only be occupied by Unlocated variables.
The analog values can be further processed in Scaling sections using the O_DEBUG, O_NORM, O_RAW and O_SCALE function blocks. The parameters EN and ENO can additionally be projected.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
SLOT	INT	Slot
CHANNEL1	ANL_OUT	Channel 1
CHANNEL2	ANL_OUT	Channel 2
CHANNEL3	ANL_OUT	Channel 3
CHANNEL4	ANL_OUT	Channel 4

Runtime error

Runtime error	If no ACO 020 00 module has been configured for the specified SLOT input, an error message appears. The status information "Open circuit on channel" can be collected via the status register defined in the I/O map.

ACO130: Configuring the Quantum module ACO 130 00

8

Overview

At a Glance

This chapter describes the block ACO130.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

The Function block is used to edit the configuration data of an ACO 130 00 Quantum module for subsequent use by the scaling EFBs.

This module has 8 output channels for controlling and supervising the currents in the ranges 0 to 20 mA, 0 to 25 mA and 4 to 20 mA.

For the configuration of an ACO 130 00 the function block in the configuration section is connected to the corresponding SLOT output of the QUANTUM or DROP Function block. The 4x referencesspecified in the I/O map are automatically assigned internally to individual channels. The channels can only be occupied byUnlocated variables.

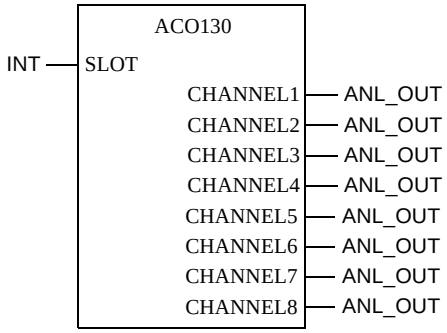
The analog values can be further processed in the Scaling Sections using the O_NORM, O_PHYS, O_RAW and O_SCALE Function blocks. O_DEBUG and O_DBSET are also available for testing purposes.

EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
SLOT	INT	Slot
CHANNEL1	ANL_OUT	Channel 1
:	:	:
CHANNEL8	ANL_OUT	Channel 8

Runtime error

Runtime error

If no ACO 130 00 module has been configured for the specified SLOT input, an error message will appear.

In the "4 to 20 mA" mode, the "open circuit on channel" status information is available. It can be collected via the status register defined in the I/O mapping.

ADU204: Configuring the Compact module ADU 204

9

Overview

At a Glance

This chapter describes the block ADU204.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

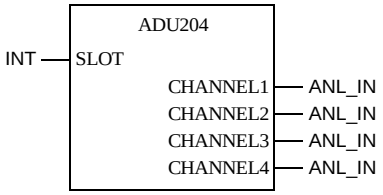
Function description

The Function block is used to edit the configuration data of a ADU 204 Compact module for subsequent use by the scaling EFBs.
This module has four input channels for measuring temperature and resistance.
For the configuration of an ADU 204, the function block in the Configuration section is connected to the corresponding SLOT output of the COMPACT Function block.
The 3x references specified in the I/O map are automatically assigned internally to the individual channels and can therefore only be occupied by Unlocated Variables.
The analog values can be further processed in Scaling sections using the I_DEBUG, I_NORM, I_SCALE, I_PHYS, I_DBSET, I_RAW Function blocks.
EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
SLOT	INT	Slot
CHANNEL1	ANL_IN	Channel 1
CHANNEL2	ANL_IN	Channel 2
CHANNEL3	ANL_IN	Channel 3
CHANNEL4	ANL_IN	Channel 4

Runtime error

Runtime error

If no ADU 204 module has been configured for the specified SLOT input, an error message appears.
The status information "Range violation" on a channel can be collected via the status entry in the ANL_IN data structure.

ADU205: Configuring the Compact module ADU 205

10

Overview

At a Glance

This chapter describes the block ADU205.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

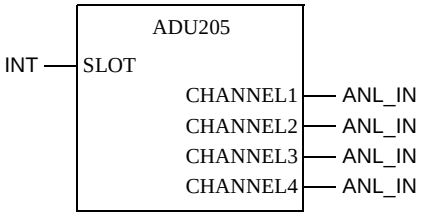
Function description

The Function block is used to edit the configuration data of a ADU 205 Compact module for subsequent use by the scaling EFBs.
This module has four input channels for combined bipolar and unipolar voltage and current processing.
For the configuration of an ADU 205, the function block in the Configuration section is connected to the corresponding SLOT output of the COMPACT Function block.
The 3x references specified in the I/O map are automatically assigned internally to the individual channels and can therefore only be occupied by Unlocated Variables.
The analog values can be further processed in Scaling sections using the I_DEBUG, I_NORM, I_SCALE, I_PHYS, I_DBSET, I_RAW Function blocks.
EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
SLOT	INT	Slot
CHANNEL1	ANL_IN	Channel 1
CHANNEL2	ANL_IN	Channel 2
CHANNEL3	ANL_IN	Channel 3
CHANNEL4	ANL_IN	Channel 4

Runtime error

Runtime error

If no ADU 205 module has been configured for the specified SLOT input, an error message appears.
A range warning for the channels is evaluated by the processing Function blocks (See *Function description, p. 62*).
The status information "Range violation" or "Open circuit" on a channel can be collected via the status entry in the ANL_IN data structure.

ADU206: Configuring the Compact module ADU 206/ ADU 256

11

Overview

At a Glance

This chapter describes the block ADU206.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

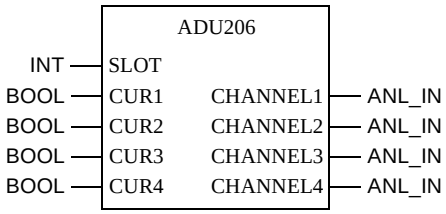
Function description

The Function block is used to edit the configuration data of an ADU 206 / ADU256 Compact module for subsequent use by the scaling EFBs.
This module has four input channels for combined bipolar and unipolar voltage and current processing.
For the configuration of an ADU 206 / ADU 256 the function block in the configuration section
is connected to the corresponding output of the function block COMPACT. The 3x references specified in the I/O map are automatically assigned internally to the individual channels and can therefore only be occupied by Unlocated Variables.
The CURx parameters indicate whether a channel is configured to process current (TRUE) or voltage (FALSE). As this card has range settings (in the parametering of this card in the I/O map) which do not allow current processing, the relevant CURx input must in this case be FALSE.
The analog values can be further processed in Scaling sections using the I_DEBUG, I_NORM, I_SCALE, I_PHYS, I_DBSET, I_RAW Function blocks.
EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
SLOT	INT	Slot
CUR1	BOOL	0: Channel 1 processing voltage 1: Channel 1 processing current
CUR2	BOOL	0: Channel 2 processing voltage 1: Channel 2 processing current
CUR3	BOOL	0: Channel 3 processing voltage 1: Channel 3 processing current
CUR4	BOOL	0: Channel 4 processing voltage 1: Channel 4 processing current
CHANNEL1	ANL_IN	Channel 1
CHANNEL2	ANL_IN	Channel 2
CHANNEL3	ANL_IN	Channel 3
CHANNEL4	ANL_IN	Channel 4

Runtime error

Runtime error

If no ADU 206 / ADU 256 module has been configured for the specified SLOT input, an error message will appear.
If current processing is selected (TRUE at the relevant CURx input) for a channel, which is only configured to allow voltage processing, an error message appears with the number of the affected channel (1-4).
A range warning for the channels is evaluated by the processing Function blocks (See *Function description, p. 66*).
The status information "Open circuit or range violation on channel" can be collected via the status register (3x reference) defined in the I/O map.3

ADU214: Configuring the Compact module ADU 214

12

Overview

At a Glance

This chapter describes the block ADU214.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

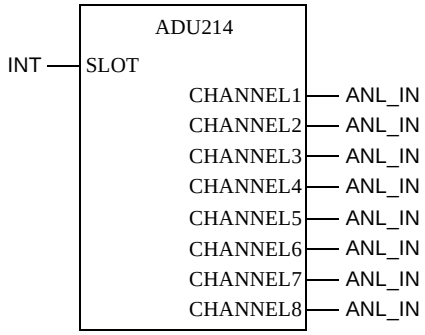
Function description

The Function block is used to edit the configuration data of a ADU 214 Compact module for subsequent use by the scaling EFBs.
The module has a maximun of 8 analog input channels for voltage, current, resistance and temperature measurement. They can be used as unipole or as bipolar inputs. Mixed operation is also allowed.
For the configuration of an ADU 214, the function block in the Configuration section is connected to the corresponding SLOT output of the COMPACT Function block.
The 3x references specified in the I/O map are automatically assigned internally to the individual channels and can therefore only be occupied by Unlocated variables.
The analog values can be further processed in Scaling sections using the I_DEBUG, I_NORM, I_SCALE, I_PHYS, I_DBSET, I_RAW Function blocks.
EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
SLOT	INT	Module slot
CHANNEL1	ANL_IN	Channel 1
CHANNEL2	ANL_IN	Channel 2
CHANNEL3	ANL_IN	Channel 3
CHANNEL4	ANL_IN	Channel 4
CHANNEL5	ANL_IN	Channel 5
CHANNEL6	ANL_IN	Channel 6
CHANNEL7	ANL_IN	Channel 7
CHANNEL8	ANL_IN	Channel 8

Runtime error

Runtime error

If no ADU 214 module has been configured for the specified SLOT input, an error message appears.
The error information for a channel, e.g. "line break or range exceeded" can be collected via the status register defined in the I/O map. If errors occur simultaneously on several channels, the error of the lowest channel number will be shown until it is removed. Then the error message for the next higher channel number is given, etc.

AII330: Configuring the Quantum module AII 330 00

13

Overview

At a Glance

This chapter describes the block AII330.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

The Function block is used to edit the configuration data of an AII 330 00 Quantum module for subsequent use by the scaling EFBs.

The module has 8 intrinsically safe channels and can be used either as a resistor temperature sensor (RTD) or as a thermoelement/millivolt input module.

For the configuration of an AII 330 00 the function block in the configuration section is connected to the corresponding SLOT output of the QUANTUM or DROP Function block. The 3x referencesspecified in the I/O map are automatically assigned internally to individual channels. The channels can only be occupied byUnlocated variables.

The analog values can be further processed inScaling sections using the I_NORM, I_NORM_WARN, I_PHYS, I_PHYS_WARN, I_RAW, I_SCALE and I_SCALE_WARN Function blocks. I_DEBUG and I_DBSET are also available for testing purposes.

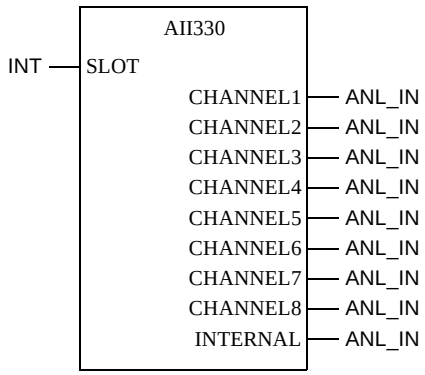
Note: I_SCALE and I_SCALE_WARN can not be used to set parameters for physical or temperature values.

EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameters	Data type	Meaning
SLOT	INT	Module slot
CHANNEL1	ANL_IN	Channel 1
CHANNEL2	ANL_IN	Channel 2
CHANNEL3	ANL_IN	Channel 3
CHANNEL4	ANL_IN	Channel 4
CHANNEL5	ANL_IN	Channel 5
CHANNEL6	ANL_IN	Channel 6
CHANNEL7	ANL_IN	Channel 7
CHANNEL8	ANL_IN	Channel 8
INTERNAL	ANL_IN	Temperature of module

Runtime error

Runtime error

If no AII 330 00 module has been configured for the specified SLOT input, an error message will appear.

The range warning for the channels can be evaluated using the function block I_NORM_WARN, I_SCALE_WARN or I_PHYS_WARN.

The status information "Open circuit or range violation on channel" can be collected via the 3 references (3x-8; module status register) defined in the I/O map or via the status register defined in the I/O map. (The information in the status register is a copy of the 3x+8 module status register (High-Byte).

AII33010: Configuring the Quantum module AII 330 10

14

Overview

At a Glance

This chapter describes the block AII33010.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

The Function block is used to edit the configuration data of an AII 330 10 Quantum module for subsequent use by the scaling EFBs.

The module has 8 unipolar intrinsically safe channels. The following ranges can be selected: 0 to 20 mA , 0 to 25 mA and 4 to 20 mA.

For the configuration of an AII 330 00 the function block in the configuration section is connected to the corresponding SLOT output of the QUANTUM or DROP Function block. The 3x referencesspecified in the I/O map are automatically assigned internally to individual channels. The channels can only be occupied byUnlocated variables.

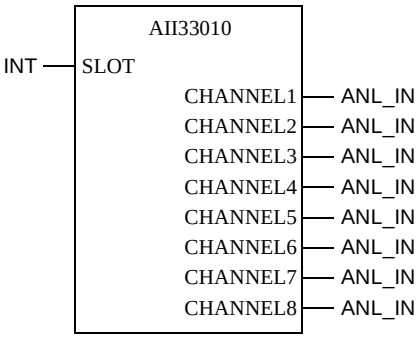
The analog values can be further processed in Scaling Sections using the I_NORM, I_PHYS, I_RAW and I_SCALE Function blocks. I_DEBUG and I_DBSET are also available for testing purposes.

EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
SLOT	INT	Slot
CHANNEL1	ANL_IN	Channel 1
CHANNEL2	ANL_IN	Channel 2
CHANNEL3	ANL_IN	Channel 3
CHANNEL4	ANL_IN	Channel 4
CHANNEL5	ANL_IN	Channel 5
CHANNEL6	ANL_IN	Channel 6
CHANNEL7	ANL_IN	Channel 7
CHANNEL8	ANL_IN	Channel 8

Runtime error

Runtime error

If no AII 330 10 module has been configured for the specified SLOT input, an error message will appear.

In the "4 to 20 mA" mode, the "open circuit on channel" status information is available. It can be collected via the status register defined in the I/O mapping.

AIO330: Configuring the Quantum module AIO 330 00

15

Overview

At a Glance

This chapter describes the block AIO330.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

The Function block is used to edit the configuration data of an AIO 330 00 Quantum module for subsequent use by the scaling EFBs.

This module has 8 intrinsically safe symmetrical output channels for controlling and supervising the currents in the ranges 0 to 20 mA, 0 to 25 mA and 4 to 20 mA.

For the configuration of an AIO 330 00 the function block in the configuration section is connected to the corresponding SLOT output of the QUANTUM or DROP Function block. The 4x referencesspecified in the I/O map are automatically assigned internally to individual channels. The channels can only be occupied byUnlocated variables.

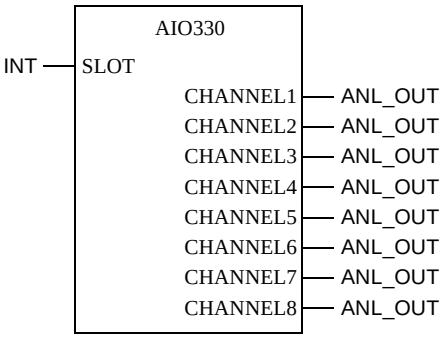
The analog values can be further processed in the Scaling Sections using the O_NORM, O_PHYS, O_RAW and O_SCALE Function blocks. O_DEBUG and O_DBSET are also available for testing purposes.

EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
SLOT	INT	Slot
CHANNEL1	ANL_OUT	Channel 1
CHANNEL2	ANL_OUT	Channel 2
CHANNEL3	ANL_OUT	Channel 3
CHANNEL4	ANL_OUT	Channel 4
CHANNEL5	ANL_OUT	Channel 5
CHANNEL6	ANL_OUT	Channel 6
CHANNEL7	ANL_OUT	Channel 7
CHANNEL8	ANL_OUT	Channel 8

Runtime error

Runtime error

If no AIO 330 00 module has been configured for the specified SLOT input, an error message will appear.

The status information "Open circuit or range violation on channel" can be collected via the status register defined in the I/O map.

AMM090: Configuring the Quantum module AMM 090

16

Overview

At a Glance

This chapter describes the block AMM090.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

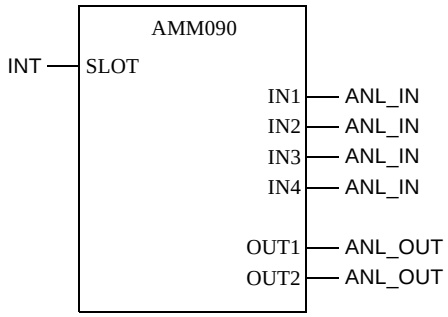
Function description

The Function block is used to edit the configuration data of an AMM 090 00 Quantum module for subsequent use by the scaling EFBs.
This module has 4 bipolar input channels for mixed voltage and current processing. The module also has 2 current output channels.
For the configuration of an AMM 090 00 the function block in the configuration section is connected to the corresponding slot output of the QUANTUM or DROP Function block. The 3x references and 4x references specified in the I/O map are automatically assigned internally to the individual channels and can therefore only be occupied by Unlocated variables.
The analog values can be further processed in Scalings sections using the function blocks I_DEBUG, I_NORM, I_NORM_WARN, I_PHYS, I_PHYS_WARN, I_RAW, I_SCALE, I_SCALE_WARN for the inputs and O_DEBUG, O_NORM, O_SCALE, O_PHYS, O_DBSET , O_RAW for the outputs.
EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
SLOT	INT	Slot
IN1	ANL_IN	Input channel 1
IN2	ANL_IN	Input channel 2
IN3	ANL_IN	Input channel 3
IN4	ANL_IN	Input channel 4
OUT1	ANL_OUT	Output channel 1
OUT2	ANL_OUT	Output channel 2

Runtime error

Runtime error

If no AMM 090 00 module has been configured for the specified SLOT input, an error message appears.
The range warning for the input channels can be evaluated using the I_NORM_WARN, I_PHYS_WARN or I_SCALE_WARN Function blocks.
The outputs return no warnings.
The status message "Open circuit or range violation on channel" can be requested via the status register defined in the I/O map.

ANA_16I: Configuring the module AAI 140 00

17

Overview

At a Glance

This chapter describes the block ANA_16I.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

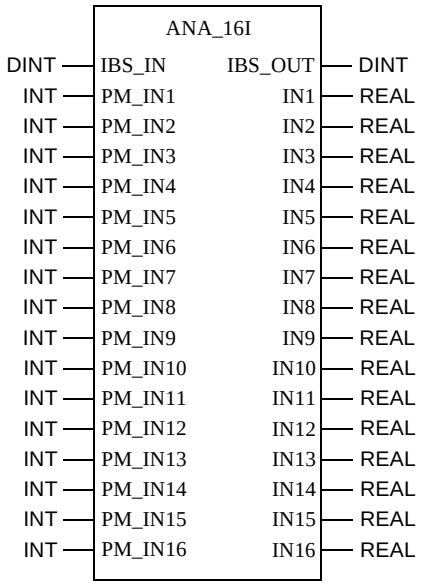
Function description

The Function block is a software connection to the InterBus-S Momentum/IS 170 AAI 140 00 hardware module.
The function block has 16 analog inputs. The function block must be parametered in the same way as the hardware module.
EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
IBS_IN	DINT	Incoming InterBus-S
PM_IN1	INT	Parameter for input 1
:	:	:
PM_INT	INT	Parameter for input 16
IBS_OUT	DINT	Outgoing InterBus-S

Parameter	Data type	Meaning
IN1	REAL	Input 1 of the module
:	:	:
IN16	REAL	Input 16 of the module

Detailed description

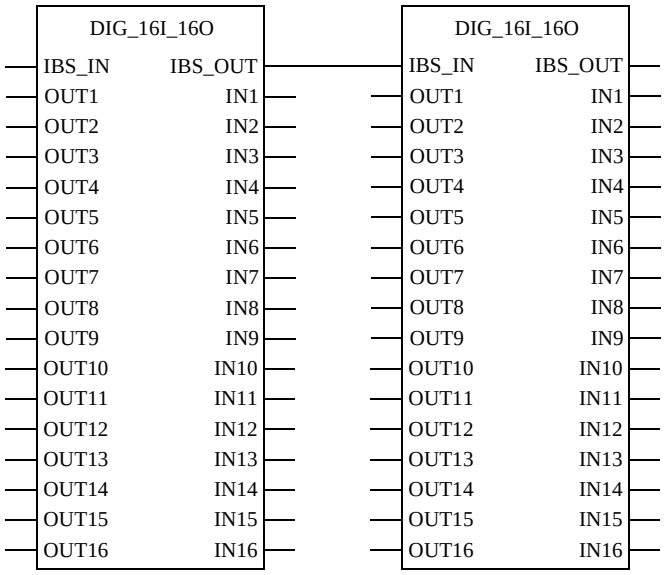
Detailed description

The funciton block occupies 16 input words and 16 outputwords in the Status-RAM.

Parameter description - inputs

IBS_IN

IBS_IN = Connection for the incoming remote bus part of InterBus-S
On the hardware, the male connector is on the top left of the module. Here, the module is connected to the outgoing remote bus (IBS_OUT) of the master (1st module on the bus) or the preceding module (see diagram). The link can be made via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two bus devices.
Connection of two InterBus-S modules



PM_INx

PM_INx = Parameters for the input channels
x stands for the digit 1 to 16 which indicates the particular input channel.
These parameters are used to parameter the input channels.
The meaning of the parameter values can be found in the table below.

Parameter value	Meaning
0	reserved
10	+/- 5V
11	+/- 10V
12	Channel inactive
14	4...20 mA

Example:
Input 3 should be 4 ...20 mA.
PM_IN3 = "14"

Note: The reserved parameter codes are not accepted by the module, i.e. the last parameter used will still apply.

Parameter description - Outputs

IBS_OUT

IBS_OUT = Connection for the outgoing remote bus part of InterBus-S
On the hardware, the male connector is on the top right of the module. The module is connected to the incoming remote bus (IBS_IN) of the following module, either via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two InterBus-S modules.

INx

INx = input channel x
x stands for the digit 1 to 16 which indicates the particular input channel. The analog process values are read into the InterBus-S module via the relevant input (INx).

Note: The values to be applied here are standardized, i.e. given as voltage in volts or as current in milliamperes. The input as current or voltage depends on the parameters set for the particular channel. .

ANA_4I_M: Configuring the module AAI 520 40

18

Overview

Introduction

This chapter describes the block ANA_41_M.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

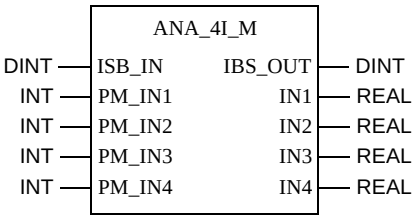
Function description

The ANA_4I_M Function block (M = measurement) is a software connection to the Momentum/IS 170 AAI 520 40 hardware module.
The function block has 4 analog inputs and is a special block for temperature and extra-low voltage measurements. The function block must be parametered in the same way as the hardware module.
EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
IBS_IN	DINT	Incoming InterBus-S
PM_IN1	INT	Parameter for input 1
:	:	:
PM_IN4	INT	Parameter for input 4
IBS_OUT	DINT	Outgoing InterBus-S
IN1	REAL	Input 1 of the module
:	:	:
IN4	REAL	Input 4 of the module

Detailed description

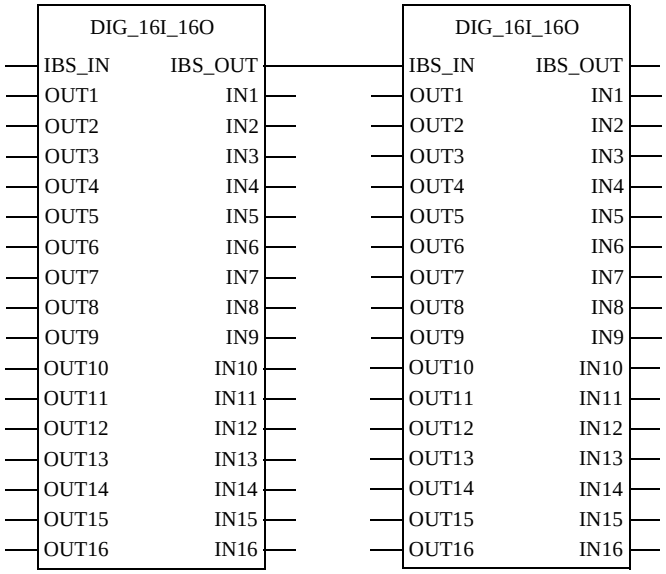
Detailed description

The block occupies 4 input words and 4 output words in the Status-RAM.

Parameter description - inputs

IBS_IN

IBS_IN = Connection for the incoming remote bus part of InterBus-S
On the hardware, the male connector is on the top left of the module.
The module is connected here to the outgoing remote bus (IBS_OUT) of the master (1st module on the bus) or the preceding module (see diagram). The link can be via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two bus devices.
Connection of two InterBus-S modules



PM_INx

PM_INx = Parameters for the input channels
x stands for the digit 1 to 4 which indicates the particular input channel.
These parameters are used to parameter the input channels.
The meaning of the parameter values can be found in the table below.
Table input range: Thermocouple

Parameter value	Cable break detection	Temperature unit	Input range
8705	no	1/10 degrees C	Thermocouple B
8961	yes		
8833	no	1/10 degrees F	
9089	yes		
4610	no	1/10 degrees C	Thermocouple E
4866	yes		
4738	no	1/10 degrees F	
4994	yes		
4611	no	1/10 degrees C	Thermocouple J
4867	yes		
4739	no	1/10 degrees F	
4995	yes		
4612	no	1/10 degrees C	Thermocouple K
4868	yes		
4740	no	1/10 degrees F	
4996	yes		
4613	no	1/10 degrees C	Thermocouple N
4869	yes		
4741	no	1/10 degrees F	
4997	yes		
8710	no	1/10 degrees C	Thermocouple R
8966	yes		
8838	no	1/10 degrees F	
9094	yes		
8711	no	1/10 degrees C	Thermocouple S
8967	yes		
8839	no	1/10 degrees F	
9095	yes		

Parameter value	Cable break detection	Temperature unit	Input range
8712	no	1/10 degrees C	Thermocouple T
8968	yes		
8840	no	1/10 degrees F	
9096	yes		

Table input range: IEC

Parameter value	Cable break detection	Temperature unit	Input range
2592	no	1/10 degrees C	IEC PT100 RTD 2 or 4 wire
2848	yes		
2720	no	1/10 degrees F	
2976	yes		
3616	no	1/10 degrees C	IEC PT100 RTD 3 wire
3872	yes		
3744	no	1/10 degrees F	
4000	yes		
545	no	1/10 degrees C	IEC PT100 RTD 2 or 4 wire
801	yes		
673	no	1/10 degrees F	
929	yes		
1569	no	1/10 degrees C	IEC PT100 RTD 3 wire
1825	yes		
1697	no	1/10 degrees F	
1953	yes		

Table input range: US/JIS

Parameter value	Cable break detection	Temperature unit	Input range
2656	no	1/10 degrees C	US/JIS PT100 RTD 2 or 4 wire
2912	yes		
2784	no	1/10 degrees F	
3040	yes		
3680	no	1/10 degrees C	US/JIS PT100 RTD 3 wire
3936	yes		
3808	no	1/10 degrees F	
4064	yes		
609	no	1/10 degrees C	US/JIS PT100 RTD 2 or 4 wire
865	yes		
737	no	1/10 degrees F	
993	yes		
1633	no	1/10 degrees C	US/JIS PT100 RTD 3 wire
1889	yes		
1761	no	1/10 degrees F	
2017	yes		

Table input range: DIN

Parameter value	Cable break detection	Temperature unit	Input range
2595	no	1/10 degrees C	DIN Ni100 RTD 2 or 4 wire
2851	yes		
2723	no	1/10 degrees F	
2979	yes		
3619	no	1/10 degrees C	DIN Ni100 RTD 3 wire
3875	yes		
3747	no	1/10 degrees F	
4003	yes		
546	no	1/10 degrees C	DIN Ni100 RTD 2 or 4 wire
802	yes		
674	no	1/10 degrees F	
930	yes		

Parameter value	Cable break detection	Temperature unit	Input range
1570	no	1/10 degrees C	DIN Ni100 RTD 3 wire
1826	yes		
1698	no	1/10 degrees F	
1954	yes		

Table input range: +/-25mV

Parameter value	Cable break detection	Temperature unit	Input range
8720	no		+/-25mV
8976	yes		
4625	no		+/-100mV
4881	yes		

Example:
Input 3 should be +/-25mV with wiring check.
PM_IN3 = "8976"

Parameter description - Outputs

IBS_OUT

IBS_OUT = Connection for the outgoing remote bus part of InterBus-S.
On the hardware, the male connector is on the top right of the module.
The module is connected to the incoming remote bus (IBS_IN) of the following module, either via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two InterBus-S modules.

INx

INx = input channel x
x stands for the digit 1 to 16 which indicates the particular input channel.
The analog process values are read into the InterBus-S module via the relevant input (INx).

Note: The values to be applied here are standardized, i.e. given as voltages in millivolts or as real values between -32000.0 and +32000.0. The input depends on the parametering of the particular channel.

ANA_4I_2O: Configuring the TIO-module BAM 096 00

19

Overview

Introduction

This chapter describes the block ANA_41_20.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

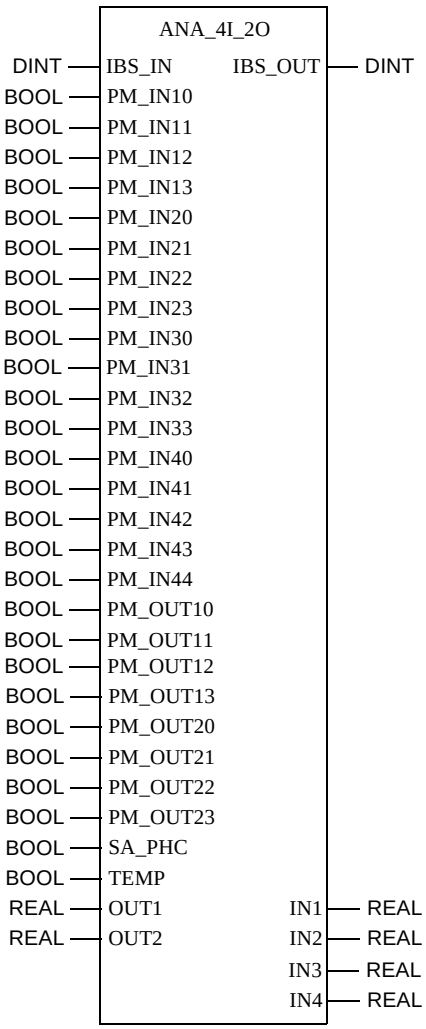
Function description

The ANA_4I_2O Function block is a software connection to the InterBus-S TIO/IS 170 BAM 096 00 hardware module.
The module has 4 analog inputs and 2 analog outputs. The function block must be configured corresponding to the hardware module (see *Detailed description, p. 105*). A different designation was selected in order to achieve a clearer relation between the name and the function of the module.
EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
IBS_IN	DINT	Incoming InterBus-S
PM_IN10	BOOL	Input 1, parameter bit 0
:	:	:
PM_IN13	BOOL	Input 1, parameter bit 3
PM_IN20	BOOL	Input 2, parameter bit 0
:	:	:
PM_IN23	BOOL	Input 2, parameter bit 3
PM_IN30	BOOL	Input 3, parameter bit 0
:	:	:
PM_IN33	BOOL	Input 3, parameter bit 3
PM_IN40	BOOL	Input 4, parameter bit 0
:	:	:
PM_IN43	BOOL	Input 4, parameter bit 3
PM_OUT10	BOOL	Output 1, parameter bit 0
:	:	:
PM_OUT13	BOOL	Output 1, parameter bit 3
PM_OUT20	BOOL	Output 2, parameter bit 0
:	:	:
PM_OUT23	BOOL	Output 2, parameter bit 3
SA_PHC	BOOL	Compatibility of the analog value representation Schneider Automation (SA) = 1, Phoenix Contact (PHC) = 0
TEMP	BOOL	Type of temperature representation, 0 = C, 1 = F
OUT1	REAL	Output 1 of the TIO
OUT2	REAL	Output 2 of the TIO
IBS_OUT	DINT	Outgoing InterBus-S
IN1	REAL	Input 1 of the TIO
IN2	REAL	Input 2 of the TIO
IN3	REAL	Input 3 of the TIO
IN4	REAL	Input 4 of the TIO

Detailed description

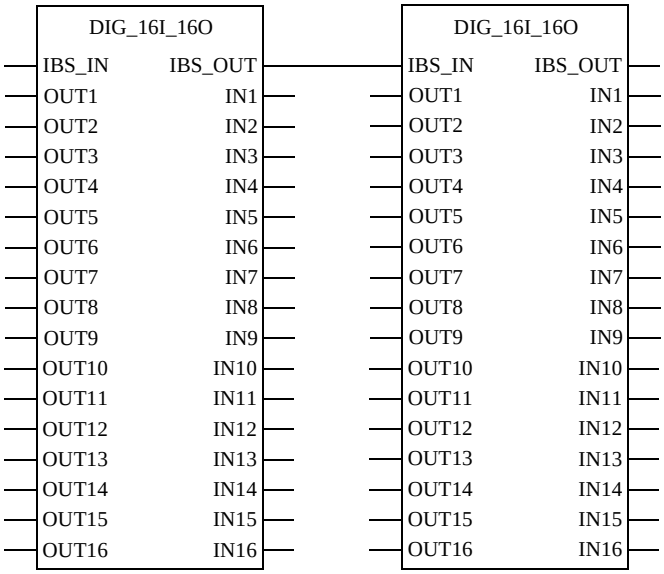
Detailed description

The block occupies 4 input words and 4 output words in the Status-RAM.

Parameter description - inputs

IBS_IN

IBS_IN = Connection for the incoming remote bus part of InterBus-S
On the hardware, the male connector is on the top left of the module.
The module is connected here to the outgoing remote bus (IBS_OUT) of the master (1st module on the bus) or the preceding module (see diagram). LinkConnection can be via a line or via a variable.line For the hardware, the type of connection corresponds to the InterBus-S cable between two bus devices.
Connection of two InterBus-S modules



PM_INxy

PM_INxy = Parameters for the input channels
x stands for the digit 1 to 4 which indicates the particular input channel.
y stands for the digit 0 to 3 which indicates the particular parameter bit.
Example: PM_IN23 = Parameters for input channel 2, bit 3
These parameters are used to parameter the input channels.
The meaning of the bits can be found in the table below.

Bit 3	Bit 2	Bit 1	Bit 0	Meaning
0	0	0	0	reserved; Channel inactive (default)
0	0	0	1	+/- 1 V
0	0	1	0	+/-20 mA (+/-5 V, when divided by 4)
0	0	1	1	+/- 10V
0	1	0	0	Channel inactive
0	1	0	1	0..0.1 V
0	1	1	0	0..0.5 V
0	1	1	1	0...10 V
1	0	0	0	reserved
1	0	0	1	0.2..0.1 V
1	0	1	0	420 mA (1...5 V, when divided by 4)
1	0	0	0	2...10 V
1	1	0	0	reserved
1	1	0	1	Pt100 with linearization
1	1	1	0	Ni100 with linearization
1	1	1	1	Resistance 0...2000 ohm

Example:
Input 3 should be 0 ...10 V.
- PM_IN30 = "1"
- PM_IN31 = "1"
- PM_IN32 = "1"
- PM_IN33 = "0"

Note: The reserved parameter codes are not accepted by the module, i.e. the last parameter used will still apply. The default parameters apply until a valid new parameter is entered.

PM_OUTxy

PM_OUTxy = Parameters for the output channels
x stands for the digit 1 or 2 which indicates the particular output channel.
y stands for the digit 0 to 3 which indicates the particular parameter bit.
Example: PM_OUT23 = Parameters for output channel 2, bit 3

These parameters are used to parameterize the output channels.
The meaning of the bits can be found in the table below.

Bit 3	Bit 2	Bit 1	Bit 0	Meaning	
				Output	Output after bus interrupt (timeout)
X	X	0	0	reserved; Channel inactive (default)	
0	0	0	1	0...20 mA	0 mA
0	0	1	0	4...20 mA	4 mA
0	0	1	1	+/-10 V/sensor supply	+0 V/sensor supply
0	1	0	0	0...20 mA	20 mA
0	1	1	1	4...20 mA	20 mA
0	1	1	0	+/-10 V/sensor supply	+10 V/sensor supply
1	0	0	1	0...20 mA	freezes
1	0	1	0	4...20 mA	freezes
1	0	1	1	+/-10 V/sensor supply	freezes
1	1	X	X	reserved	

Example:
Output 1 should be 0 20mA and be set to 0mA for bus failure.
- PM_OUT10 = "1"
- PM_OUT11 = "0"
- PM_OUT12 = "0"
- PM_OUT13 = "0"

Note: The reserved parameter codes are not accepted by the module, i.e. the last parameter used will still apply. The default parameters apply until a valid new parameter is entered.

PM_OUTxy

SA_PHC = Compatibility of the analog value representation
This parameter is used to enter the compatibility of the analog value representation; a "1" stands for Schneider Automation compatibility and a "0" for Phoenix Contact. The setting applies for inputs and outputs.

TEMP

TEMP = Type of temperature representation
This parameter can be used to select the temperature representation. If the bit is set to "0", the temperature is given in degrees Celsius, if it is "1", it is in degrees Fahrenheit.

OUTx

OUTx = output channel x
x stands for the number 1 or 2 which refers to the corresponding output. The analog values to be produced via the InterBus-S module are supplied to the process via the relevant output (OUTx).

Note: The values to be applied here are standardized, i.e. given as voltage in volts or as current in milliamperes. The input as current or voltage depends on the parametering of the particular channel.

Parameter description - Outputs

IBS_OUT

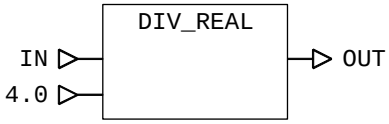
IBS_OUT = Connection for the outgoing remote bus part of InterBus-S
On the hardware, the male connector is on the top right of the module.
The module is connected to the incoming remote bus (IBS_IN) of the following module, either via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two InterBus-S modules.

INx

INx = input channel x
x stands for the number between 1 and 4 designating the corresponding input channel. The analog process values of the InterBus-S module are read via the corresponding input (INx).

Note: The values to be applied here are standardized, i.e. given as voltage in volts or as current in milliamperes. The input as current or voltage depends on the parameters set for the particular channel. .

If a channel is parametered in the +/-5 V- or 1...5 V range, the incoming values are given in milliamperes. To obtain these values as voltage, divide by 4.0 (see diagram).
Scaling an analog value



IN Analog value in mA
OUT Analog value in V

ANA_4I_2O_C: Configuring the TIO-module BAM 096 00

20

Overview

Introduction

This chapter describes the block ANA_41_20_C.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

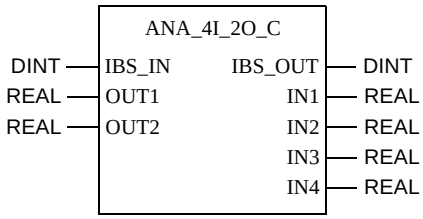
Function description

The function block ANA_4I_2O_C is a software connection to the InterBus-S TIO/IS 170 BAM 096 00 hardware module with preset current.
The module has 4 analog inputs and 2 analog outputs.
EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
IBS_IN	DINT	Incoming InterBus-S
OUT1	REAL	Output 1 of the TIO Range: 0...20 mA
OUT2	REAL	Output 2 of the TIO Range: 0...20 mA
IBS_OUT	DINT	Outgoing InterBus-S
IN1	REAL	Input 1 of the TIO range: +/-20 mA
IN2	REAL	Input 2 of the TIO range: +/-20 mA
IN3	REAL	Input 3 of the TIO range: +/-20 mA
IN4	REAL	Input 4 of the TIO range: +/-20 mA

Detailed description

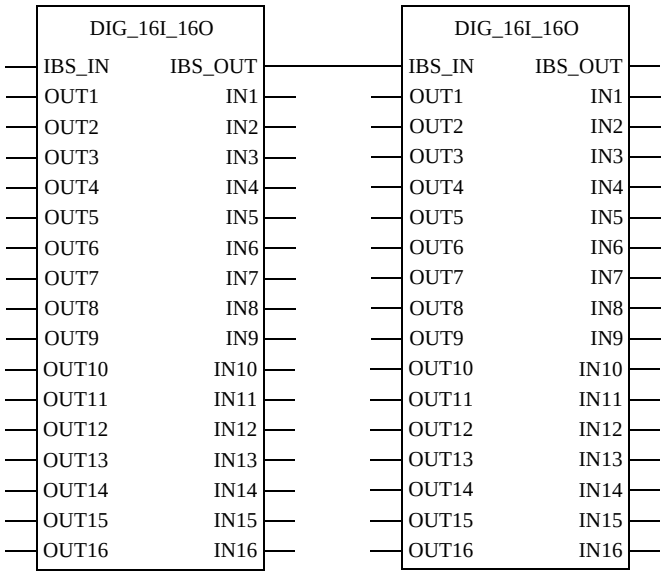
Detailed description

The function block ANA_4I_2O_C is a special version of ANA_4I_2O. The block is already set for current response (C = current), so the user does not need to set parameters for this.
On this block, all input channels are set to +/- 20 mA; the output channels are set to 0...20mA, Timeout A: set to 0 mA.
The block occupies 4 input words and 4 output words in the Status-RAM.

Parameter description - inputs

IBS_IN

IBS_IN = Connection for the incoming remote bus part of InterBus-S
On the hardware, the male connector is on the top left of the module.
The module is connected here to the outgoing remote bus (IBS_OUT) of the master (1st module on the bus) or the preceding module (see diagram). The link can be made via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two bus devices.
Connection of two InterBus-S modules



OUTx

OUTx = output channel x
x stands for the number 1 or 2 which refers to the corresponding output. The analog values to be produced via the InterBus-S module are supplied to the process via the relevant output (OUTx).

Note: The values to be applied here are standardized, i.e. given as current in milliamperes.

Parameter description - Outputs

IBS_OUT

IBS_OUT = Connection for the outgoing remote bus part of InterBus-S
On the hardware, the male connector is on the top right of the module.
The module is connected to the incoming remote bus (IBS_IN) of the following module, either via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two InterBus-S modules.

INx

INx = input channel x
x stands for the number between 1 and 4 designating the corresponding input channel. The analog process values of the InterBus-S module are read via the corresponding input (INx).

Note: The incoming values are standardized, i.e. given as current in milliamperes.

ANA_4I_2O_V: Configuring the TIO-module BAM 096 00

21

Overview

Introduction

This chapter describes the block ANA_41_20_V.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

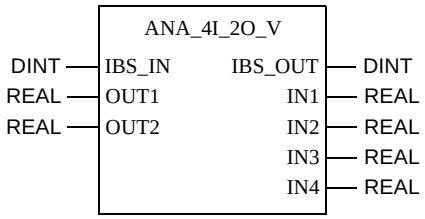
Function description

The function block ANA_4I_2O_V is a software connection to the InterBus-S TIO/IS 170 BAM 096 00 hardware module with preset voltage.
The module has 4 analog inputs and 2 analog outputs.
EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
IBS_IN	DINT	Incoming InterBus-S
OUT1	REAL	Output 1 of the TIO Range: +/-10 V
OUT2	REAL	Output 2 of the TIO Range: +/-10 V
IBS_OUT	DINT	Outgoing InterBus-S
INT1	REAL	Input 1 of the TIO Range: +/-10 V
INT2	REAL	Input 2 of the TIO Range: +/-10 V
INT3	REAL	Input 3 of the TIO Range: +/-10 V
INT4	REAL	Input 4 of the TIO Range: +/-10 V

Detailed description

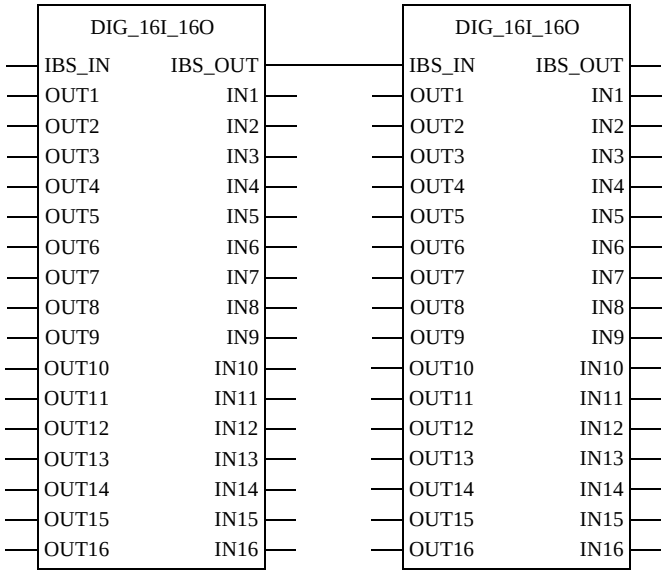
Detailed description

The function block ANA_4I_2O_V is a special version of ANA_4I_2O.
The block is already set for voltage response (V = voltage), so the user does not need to set parameters for this. With these blocks, all input channels and output channels are set to +/- 10 V and the Timeout A to 0 V.
The block occupies 4 input words and 4 output words in the Status-RAM.

Parameter description - inputs

IBS_IN

IBS_IN = Connection for the incoming remote bus part of InterBus-S
On the hardware, the male connector is on the top left of the module.
The module is connected here to the outgoing remote bus (IBS_OUT) of the master (1st module on the bus) or the preceding module (see diagram). LinkConnection can be via a line or via a variable.line For the hardware, the type of connection corresponds to the InterBus-S cable between two bus devices.
Connection of two InterBus-S modules



OUTx

OUTx = output channel x
x stands for the digit 1 or 2 which indicates the particular output channel.
The analog values to be produced via the InterBus-S module are supplied to the process via the relevant output (OUTx).

Note: The values to be applied here are standardized, ie. given as voltage in V.

Parameter description - Outputs

IBS_OUT

IBS_OUT = Connection for the outgoing remote bus part of InterBus-S
On the hardware, the male connector is on the top right of the module.
The module is connected to the incoming remote bus (IBS_IN) of the following module, either via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two InterBus-S modules.

INx

INx = input channel x
x stands for the digit 1 to 4 which indicates the particular input channel
The analog process values are read into the InterBus-S module via the relevant input (INx).

Note: The incoming values are standardized, i.e. given as voltage in V.

ANA_4O: Configuring the module BAO 126 00

22

Overview

Introduction

This chapter describes the block ANA_40.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

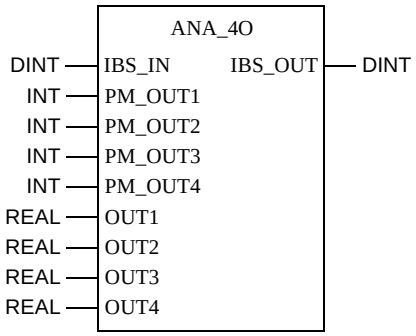
Function description

The ANA_40 Function block is a software connection to the InterBus-S TIO/IS 170 BAO 126 00 hardware module.
The function block has 4 analog outputs. The function block must be parametered in the same way as the hardware module.
EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
IBS_IN	DINT	Incoming InterBus-S
PM_OUT1	INT	Parameter output 1
:	:	:
PM_OUT4	INT	Parameter output 4
OUT1	REAL	Output 1 of the module
:	:	:
OUT4	REAL	Output 4 of the module
IBS_OUT	DINT	Outgoing InterBus-S

Detailed description

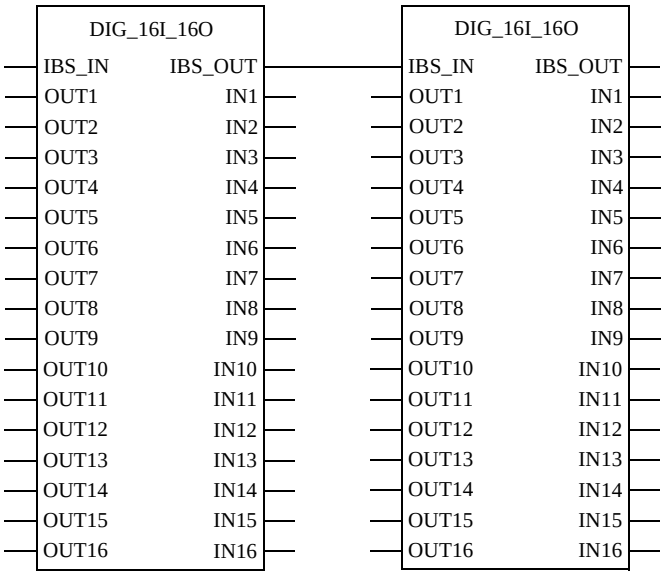
Detailed description

The function block occupies 5 output words in the Status-RAM.

Parameter description - inputs

IBS_IN

IBS_IN = Connection for the incoming remote bus part of InterBus-S
On the hardware, the male connector is on the top left of the module.
The module is connected here to the outgoing remote bus (IBS_OUT) of the master (1st module on the bus) or the preceding module (see diagram). The link can be made via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two bus devices.
Connection of two InterBus-S modules



PM_OUTx

PM_OUTx = Parameters for the output channels
x stands for the digit 1 or 4 which indicates the particular output channel.

Example:
PM_OUT2 = Parameters for output channel 2

These parameters are used to set parameters for the output channels.
The meaning of the parameter values can be found in the table below.

Parameter value	Meaning
0	reserved; Channel inactive (default)
1	0...20mA; Timeout A: 0mA
2	4...20mA; Timeout A: 4mA
3	+/- 10V; Timeout A: 0V
5	0...20mA; Timeout A: 20mA
6	4...20mA; Timeout A: 20mA
7	+/- 10V; Timeout A: +10V
9	0...20mA; Timeout A: freezes
10	4...20mA; Timeout A: freezes
11	+/- 10V; Timeout A: freezes

A = Output after bus interrupt

Note: All other parameter values are reserved.

Example:
Output 1 should be 0 ...20mA and set to 0mA for bus failure.
PM_OUT1 = "1"

Note: The reserved parameter codes are not accepted by the module, i.e. the last parameter used will still apply. The default parameters apply until a valid new parameter is entered.

OUTx

OUTx = output channel x
x stands for the digit 1 or 4 which indicates the particular output channel.
The analog values to be produced via the InterBus-S module are supplied to the process via the relevant output (OUTx).

Note: The values to be applied here are standardized, i.e. given as voltage in volts or as current in milliamperes. The input as current or voltage depends on the parametering of the particular channel.

Parameter description - Outputs

IBS_OUT

IBS_OUT = Connection for the outgoing remote bus part of InterBus-S
On the hardware, the male connector is on the top right of the module.
The module is connected to the incoming remote bus (IBS_IN) of the following module, either via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two InterBus-S modules.

ANA_8I: Configuring the module AAI 030 00, BAI 036 00

23

Overview

Introduction

This chapter describes the block ANA_8I.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

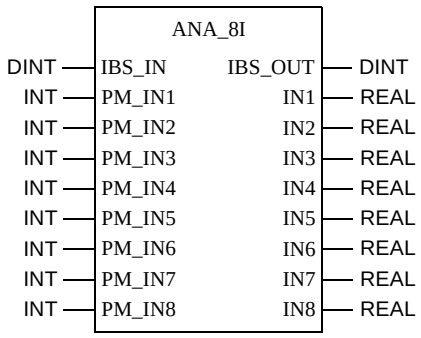
Function description

The ANA_81 Function block is a software connection to the InterBus-S, TIO/IS 170 BAI 036 00 and Momentum/IS 170 AAI 030 00 hardware modules.
The module has 8 analog inputs. The function block must be configured in the same way as the hardware module.
EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
IBS_IN	DINT	Incoming InterBus-S
PM_IN1	INT	Parameter for input 1
:	:	:
PM_IN8	INT	Parameter for input 8
IBS_OUT	DINT	Outgoing InterBus-S
IN1	REAL	Input 1 of the module
:	:	:
IN8	REAL	Input 8 of the module

Detailed description

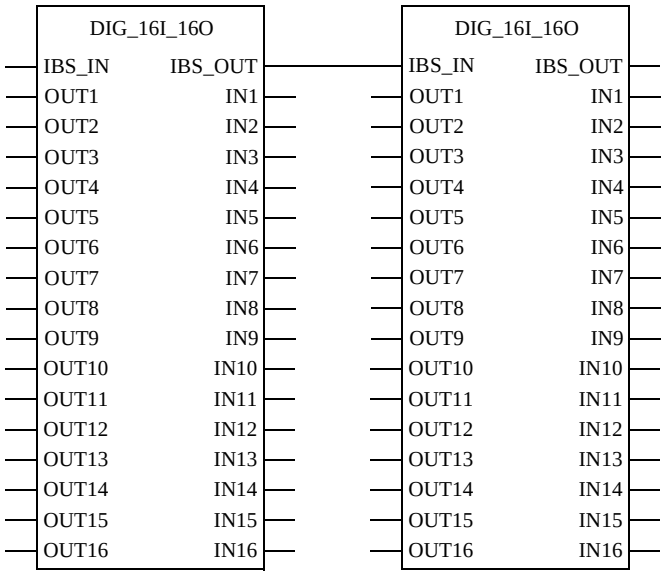
Detailed description

The function block occupies 8 input words and 8 output words in the Status-RAM.

Parameter description - inputs

IBS_IN

IBS_IN = Connection for the incoming remote bus part of InterBus-S
On the hardware, the male connector is on the top left of the module.
The module is connected here to the outgoing remote bus (IBS_OUT) of the master (1st module on the bus) or the preceding module (see diagram). The link can be made via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two bus devices.
Connection of two InterBus-S modules



PM_INx

PM_INx = Parameters for the input channels
x stands for the digit 1 to 8 which indicates the particular input channel.
These parameters are used to configure the input channels.
The meaning of the parameter values can be found in the table below.

Parameter value	Meaning
0	Channel inactive (default)
2	+/- 20mA (+/- 5 V, when divided by 4)
3	+/- 10V
4	Channel inactive
6	020mA (0...5 V, when divided by 4)
7	0...10V
10	420mA (1...5 V, when divided by 4)

A = Output after bus interrupt

Note: All other parameter values are reserved.

Example:
Input 3 should 4 ...20mA.
PM_IN3 = "10"

Note: The reserved parameter codes are not accepted by the module, i.e. the last parameter used will still apply. The default parameters apply until a valid new parameter is entered.

Parameter description - Outputs

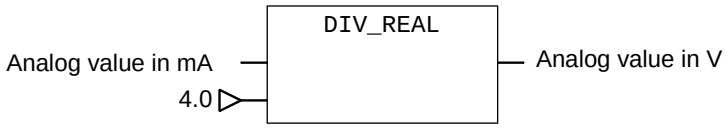
IBS_OUT

IBS_OUT = Connection for the outgoing remote bus part of InterBus-S
On the hardware, the male connector is on the top right of the module.
The module is connected to the incoming remote bus (IBS_IN) of the following module, either via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two InterBus-S modules.

INx

INx = input channel x
x stands for the digit 1 to 8 which indicates the particular input channel.
The analog process values are read into the InterBus-S module via the relevant input (INx).

Note: The values to be applied here are standardized, i.e. given as voltage in volts or as current in milliamperes. The input as current or voltage depends on the configuration of the particular channel. If a channel is configured in the +/-5V- or 1...5V range, the incoming values are given in milliamperes. To obtain these values as voltage, divide by 4.0.
Scaling an analog value



ARI030: Configuring the Quantum module ARI 030 10

24

Overview

Introduction

This chapter describes the block ARI030.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

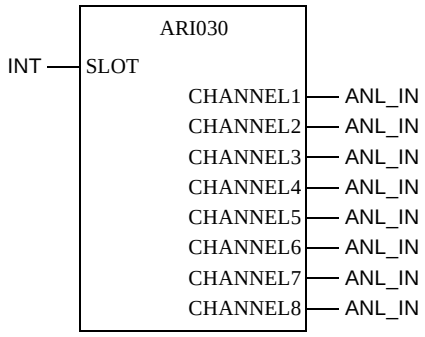
Function description

The Function block is used to edit the configuration data of an ARI 030 10 Quantum module for subsequent use by the scaling EFBs.
This module has 8 resistor temperature sensor input channels (RTD) for the processing of four-wire RTD sensors.
For the configuration of an ARI 030 10 the function block in the configuration section is connected to the corresponding SLOT output of the QUANTUM or DROP Function block. The 3x references specified in the I/O map are automatically assigned internally to the individual channels and can therefore only be occupied by Unlocated variables.
The analog values can be further processed in Scaling Sections using the I_DEBUG, I_NORM, I_NORM_WARN, I_PHYS, I_PHYS_WARN and I_RAW Function blocks.
EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
SLOT	INT	Slot
CHANNEL1	ANL_IN	Channel 1
CHANNEL2	ANL_IN	Channel 2
CHANNEL3	ANL_IN	Channel 3
CHANNEL4	ANL_IN	Channel 4
CHANNEL5	ANL_IN	Channel 5
CHANNEL6	ANL_IN	Channel 6
CHANNEL7	ANL_IN	Channel 7
CHANNEL8	ANL_IN	Channel 8

Runtime error

Runtime error

If no ARI 030 10 module has been configured for the specified SLOT input, an error message appears.
The range warning for the channels can be evaluated through the I_NORM_WARN or I_PHYS_WARN Function block.
The status message "Open circuit or range violation on channel" can be requested via the status register defined in the I/O map.

ATI030: Configuring the Quantum module ATI 030 00

25

Overview

Introduction

This chapter describes the block ATI030.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

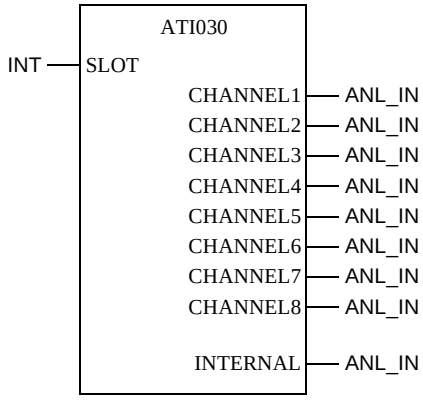
Function description

The Function block is used to edit the configuration data of an ATI 030 00 Quantum module for subsequent use by the scaling EFBs.
This module has 8 thermocouple input channels.
For the configuration of an ATI 030 00 the function block in the configuration section is connected to the corresponding SLOT output of the QUANTUM or DROP Function block. The 3x references specified in the I/O map are automatically assigned internally to the individual channels and can therefore only be occupied by Unlocated variables.
The analog values can be further processed in Scaling sections using the fuction blocks I_DEBUG, I_NORM, I_NORM_WARN and I_RAW.
EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
SLOT	INT	Slot
CHANNEL1	ANL_IN	Channel 1
CHANNEL2	ANL_IN	Channel 2
CHANNEL3	ANL_IN	Channel 3
CHANNEL4	ANL_IN	Channel 4
CHANNEL5	ANL_IN	Channel 5
CHANNEL6	ANL_IN	Channel 6
CHANNEL7	ANL_IN	Channel 7
CHANNEL8	ANL_IN	Channel 8
INTERNAL	ANL_IN	Temperature of module

Runtime error

Runtime error

If no ATI 030 00 module has been configured for the specified SLOT input, an error message appears.
The rangewarning for the input channels can be evaluated using the function block I_NORM_WARN or I_PHYS_WARN.
The status information "Range violation on channel" can be collected via the status register defined in the I/O map.

AVI030: Configuring the Quantum module AVI 030 00

26

Overview

Introduction

This chapter describes the block AVI030.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

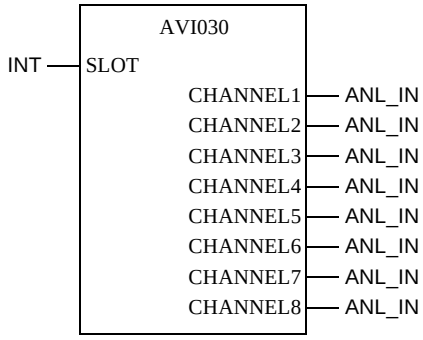
Function description

The Function block is used to edit the configuration data of an AVI 030 00 Quantum module for subsequent use by the scaling EFBs.
This module has 8 bipolar input channels for mixed voltage and current processing. For the configuration of an AVI 030 00 the function block in the configuration section is connected to the corresponding SLOT output of the QUANTUM or DROP Function block. The 3x references specified in the I/O map are automatically assigned internally to the individual channels and can therefore only be occupied by Unlocated variables.
The analog values can be further processed inScaling sections using the I_DEBUG, I_NORM, I_NORM_WARN, I_PHYS, I_PHYS_WARN, I_RAW, I_SCALE and I_SCALE_WARN Function blocks.
EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
SLOT	INT	Slot
CHANNEL1	ANL_IN	Channel 1
CHANNEL2	ANL_IN	Channel 2
CHANNEL3	ANL_IN	Channel 3
CHANNEL4	ANL_IN	Channel 4
CHANNEL5	ANL_IN	Channel 5
CHANNEL6	ANL_IN	Channel 6
CHANNEL7	ANL_IN	Channel 7
CHANNEL8	ANL_IN	Channel 8

Runtime error

Runtime error

If no AVI 030 00 module has been configured for the specified SLOT input, an error message appears.
The range warning for the channels can be evaluated using the I_NORM_WARN, I_PHYS_WARN or I_SCALE_WARN Function blocks.
The status information "Open circuit or range violation on channel" can be collected via the status register defined in the I/O map.

AVO020: Configuring the Quantum module AVO 020 00

27

Overview

Introduction

This chapter describes the block AVO20.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

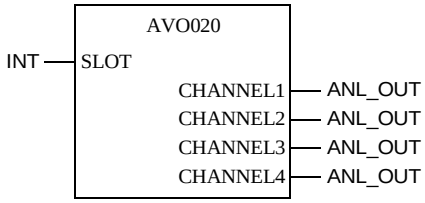
Function description

The Function block is used to edit the configuration data of the AVO 020 00 Quantum module for subsequent use by the scaling EFBs.
This module has 4 voltage output channels with mixed modes and levels.
For the configuration of an AVO 020 the function block in the configuration section is connected to the corresponding SLOT output of the QUANTUM or DROP Function block. The 4x references specified in the I/O map are automatically assigned internally to the individual channels and can therefore only be occupied by Unlocated variables.
The analog values can be further processed in Scaling Sections using the O_DEBUG, O_NORM, O_RAW and O_SCALE Function blocks.
EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
SLOT	INT	Slot
CHANNEL1	ANL_OUT	Channel 1
CHANNEL2	ANL_OUT	Channel 2
CHANNEL3	ANL_OUT	Channel 3
CHANNEL4	ANL_OUT	Channel 4

Runtime error

Runtime error

If no AVO 020 00 module has been configured for the specified SLOT input, an error message appears.

BKF_201: Configuring the Compact module BKF 201

28

Overview

Introduction

This chapter describes the block BKF_201.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

The BKF_201 Function block is the software connection to an InterBus-S BKF 201 master module.
It ensures that the data on the InterBus-S is correctly transferred to and read by the relevant module. The BKF 201 controls the bus and monitors operational performance.

Varying quantities of InterBus-S data can be transferred/read according to the slot.

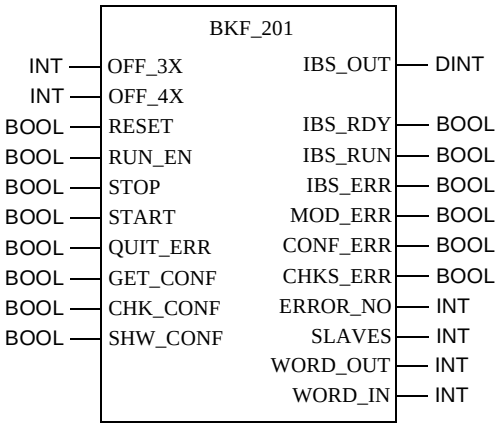
- Slot in main rack (BKF201(64 W))
  63 input words and 63 output words
- Slot in rack (BKF201(16 W))
  15 input words and 15 output words

The BKF 201 occupies 16 or 64 input words and 16 or 64 output words in PLC memory.
The first input word and the first output word are occupied by the BKF 201 itself. These contain the BKF 201 control bits and status bits. The remaining 15 or 63 input words and 15 or 63 output words contain the I/O data for the InterBus-S modules. EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
OFF_3X	INT	Offset for 3x address
OFF_4X	INT	Offset for 4x address
RESET	BOOL	Reset and reconfigure the BKF 201
RUN_EN	BOOL	CPU STOP routine (acc. to operations software Version 1.01)
STOP	BOOL	Emergency stop
START	BOOL	Start cycle
QUIT_ERR	BOOL	Error acknowledgment
GET_CONF	BOOL	Get configuration
CHK_CONF	BOOL	Check configuration
SHW_CONF	BOOL	Show configuration
IBS_OUT	DINT	Outgoing InterBus-S
IBS_RDY	BOOL	InterBus-S ready
IBS_RUN	BOOL	InterBus-S data is being transmitted
IBS_ERR	BOOL	InterBus-S faulty
MOD_ERR	BOOL	Module error
CONF_ERR	BOOL	InterBus-S configuration faulty
CHKS_ERR	BOOL	Checksum error
ERROR_NO	INT	Error number
SLAVES	INT	Number of InterBus-S devices
WORD_OUT	INT	Number of process data output words
WORD_IN	INT	Number of process data input words

Detailed description

Detailed description

The BKF_201 Function block works in the same way as its hardware counterpart. However, it has been possible to greatly simplify operation through the software connection. The module only processes words. It occupies either 64 (main rack) or 16 (rack) 3x addresses and either 64 (main rack) or 16 (rack) 4x addresses in the PLC.

Note: The dip switches on the back of the BKF 201 can be used to select the data width (16/64 words) of the module (see HW description for the module).

Parameter description - inputs

The BKF_201 inputs in Concept differ only slightly from those for the hardware module. Compared to the hardware, inputs OFF_3X, OFF_4X and RESET are new. The input abbreviations are assigned the following module functions or control bits.

OFF_3X, OFF_4X

OFF_3X = Offset 3x address
OFF_4X = Offset 4x address
On the function block, the relevant address offsets for 3x and 4x addresses are given at the two inputs (start addresses of the HW modules in I/O map).

Example:
The BFK 201 (16 W) is entered in the PLC configurator, as shown in the table.

Slot	Module	In Ref	In End	Out Ref	Out End
4	BKF201(16 W)	300020	300035	400020	400035

If the initial addresses for the BKF 201 are 3:20 for the input words and 4:20 for the output words, then

- OFF_3X = 20 and
- OFF_4X = 20.

RESET

RESET = reset and reconfigure the BKF 201
If RESET is set to "1", the BKF 201 is reset and a new InterBus-S configuration is started. RESET also performs error acknowledgment, see also "Time diagram for error acknowledgment".

RUN_EN

RUN_EN = CPU STOP routine (according to operations software Version 1.01)
0 = the IOBUS is deactivated if the CPU is stopped
1 = the IOBUS remains active if the CPU is stopped

STOP

STOP = emergency stop
If this bit is set, the InterBus-S stops immediately and all outputs are set to zero. When this bit is set, the other control bits have no effect.

START

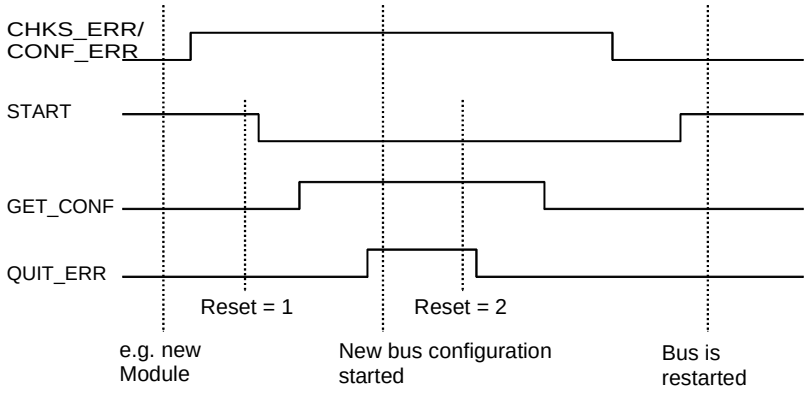
START = start cycle
The master transmits data to the InterBus-S devices when this bit is set. If the bit is deleted, the device outputs remain at their last value. Input information is maintained.

QUIT_ERR

QUIT_ERR = error acknowledgment
As its name suggests, this is used to acknowledge errors.
This bit should not be set permanently, otherwise any errors occurring will be immediately acknowledged and therefore deleted.

GET_CONF

GET_CONF = get configuration
When this bit is set, a new InterBus-S configuration is started. This is useful when the bus structure has been changed and the master has set either the CONF_ERR or CHKS_ERR status bit (see Time diagram). All other bits must first be set to zero. After the GET_CONF has been set, the QUIT_ERR should be set in a way that ensures both control bits are present simultaneously.
Vergessen Sie nicht beide Bits wieder zurückzusetzen und den Bus wieder zu starten.
Zeitdiagramm zur Fehlerquittierung



CHK_CONF

CHK_CONF = Konfiguration überprüfen
If this bit is set, the BKF 201 compares the actual configuration checksum with the desired configuration checksum. If they differ, the bus is stopped and CONF_ERR is set.

SHW_CONF	SHW_CONF = Show configuration The actual InterBus-S configuration is displayed; all other control bits must be "0". The configuration is displayed at the outputs SLAVES, WORD_OUT and WORD_IN. Until this function is used, the outputs are set to "0".
Parameter description - Outputs	The BKF_201 outputs in Concept differ only slightly from those for the hardware module. Compared to the hardware, the status indicators CHKS_ERR and ERROR_NO are new or have been changed. Outputs IBS_RDY to CONF_ERR correspond to the status bits of the BFK 201. The input abbreviations are assigned the following module functions or control bits.
IBS_OUT	IBS_OUT = Connection for the outgoing remote bus part of InterBus-S InterBus-S connection on the front panel of the BKF 201. From here, the first module on InterBus-S is connected to the master, either via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable from the master to the first module on InterBus-S.
IBS_RDY	IBS_RDY = InterBus-S ready InterBus-S is ready and error-free. This bit corresponds to LED No. 3 on the BKF 201.
IBS_RUN	IBS_RUN = InterBus-S data is being transmitted InterBus-S is operating without error, process data is being exchanged. This bit corresponds to LED No. 4 on the BKF 201.
IBS_ERR	IBS_ERR = InterBus-S error Indicates a bus error. This can be due to an open circuit, short circuit or voltage failure on a device or break-down during data transmission. This bit corresponds to LED No. 5 on the BKF 201.
MOD_ERR	MOD_ERR = module error An error has occurred on an InterBus-S module. The error does not stop InterBus-S. This bit corresponds to LED No. 6 on the BKF 201.
CONF_ERR	CONF_ERR = configuration of InterBus-S faulty This error can be due to wiring errors, changes to the configuration during operation, devices not ready or similar. This bit corresponds to LED No. 7 on the BKF 201.

CHKS_ERR

CHKS_ERR = checksum error
The InterBus-S configuration was changed during loss of voltage on the BKF 201. For an error to be detected, data loss in the PLC must be excluded. The checksum error does not occur when a program is loaded for the first time. The checksum error is not detected by the BKF 201 hardware, which can in certain circumstances lead to errors on InterBus-S, because the BKF 201 configures itself after voltage recovery. Error detection is possible because the old checksum is stored and compared with the new checksum. The error is deleted using QUIT_ERR or RESET.

ERROR_NO

ERROR_NO = Device error number
Indicates the device number of a faulty device on the bus. This indication corresponds to LEDs numbered. 14 (Significance 1) to 21 (Significance 80) on the BKF 201.
Example:

- Bus connection between device 1 and device 2 interrupted.
  Display: 2
- Voltage failure on device 1.
  Display: 1

SLAVES

SLAVES = number of InterBus-S devices
Indicates the number of bus devices connected to the master.

WORD_OUT

WORD_OUT = number of process data output words
Indicates the number of output words (4x register) occupied in the master. If the value indicated is 63 (main rack), or 15 (rack) then no additional InterBus-S module which occupies one or more output word in the master can be connected to InterBus-S.

WORD_IN

WORD_IN = number of process data input words
Indicates the number of input words (3x register) occupied in the master. If the value indicated is 63 (main rack), or 15 (rack) then no additional InterBus-S module which occupies more than one input word in the master can be connected to InterBus-S.

Runtime error

Runtime error

An error message (E_INPUT_VALUE_OUT_OF_RANGE) appears if the offset for the 3x or 4x addresses < 0 and/or more than the maximum permissible value.

BNO_671: Configuring the TIO-module BNO 671 00

29

Overview

Introduction

This chapter describes the block BNO_671.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

The BNO_671 Function block in Concept functions in the same way as its hardware counterpart.
By programming the bus as a function block in Concept, it can be divided into different segments even without 170 BNO 671 00 hardware. As only unidirectional connections are possible in Concept, unlike with hardware, the IBS_OUT output of the last module in the remote bus spur must be connected to the LRB_IN input of the BNO_671. The module does not occupy a word in the master.

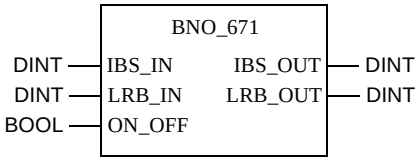
Note: If the 170 BNO 671 00 is used, its status cannot be indicated using the BNO_671.

EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

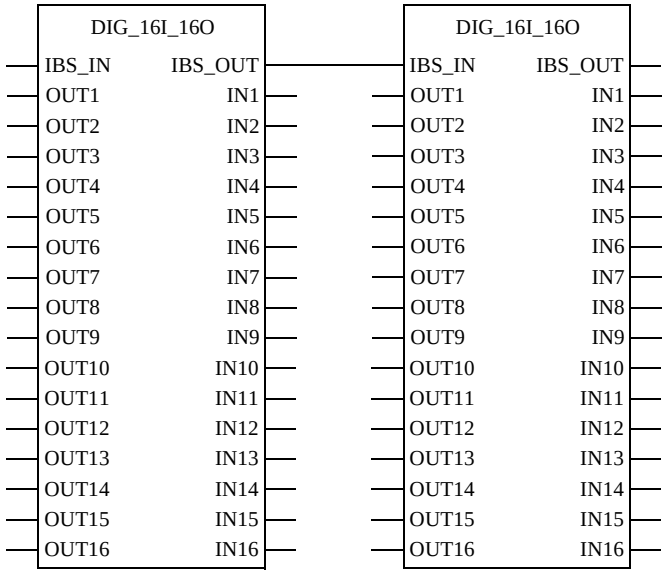
Parameters	Data type	Meaning
IBS_IN	DINT	Incoming InterBus-S
LRB_IN	DINT	Incoming remote bus branch and/or local bus (Local and/or Remote Bus)
ON_OFF	BOOL	On or Off switch for the remote bus branch output
IBS_OUT	DINT	Outgoing InterBus-S
LRB_OUT	DINT	Outgoing remote bus branch and/or local bus (Local and/or Remote Bus)

Detailed description

Parameter description - inputs

IBS_IN

IBS_IN = Connection for the incoming remote bus part of InterBus-S
On the hardware, the male connector is on the top left of the module.
The module is connected here to the outgoing remote bus (IBS_OUT) of the master (1st module on the bus) or the preceding module (see diagram). The link can be made via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two bus nodes.
Connection of two InterBus-S modules



LRB_IN

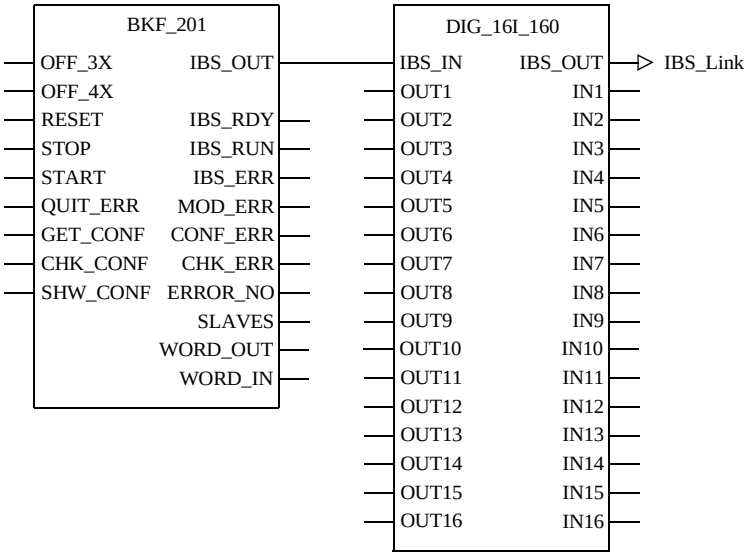
LRB_IN = Local remote bus input (incoming remote bus branch)
The IBS_OUT output of the last module on the remote bus branch is connected to this input to ensure that data is passed on correctly in InterBus-S.

ON_OFF	ON_OFF = Input for switching the remote bus branch on and off A"0" at input ON_OFF means that all outputs of an INTERBUS module are set to "0" on the remote bus branch. A "1" means that the values predefined in the program (e.g. UNI_I_O) are written to the corresponding 4x register and transmitted to the process. ON_OFF has no effect on the inputs of the INTERBUS module. Data traffic in the remote bus branch is independent from the valence of ON_OFF, i.e. it is not affected.
Parameter description - Outputs	
IBS_OUT	IBS_OUT = Connection for the outgoing remote bus part of InterBus-S On the hardware, the male connector is on the top right of the module. The module is connected to the incoming remote bus (IBS_IN) of the following module, either via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two InterBus-S modules.
LRB_OUT	LRB_OUT = Connection for the outgoing InterBus-S remote bus branch On the hardware, the male connector is on the front panel of the module and is marked as Local Remote Bus. The module is connected to the incoming remote bus (IBS_IN) of the first module on the remote bus branch, either via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two InterBus-S modules.

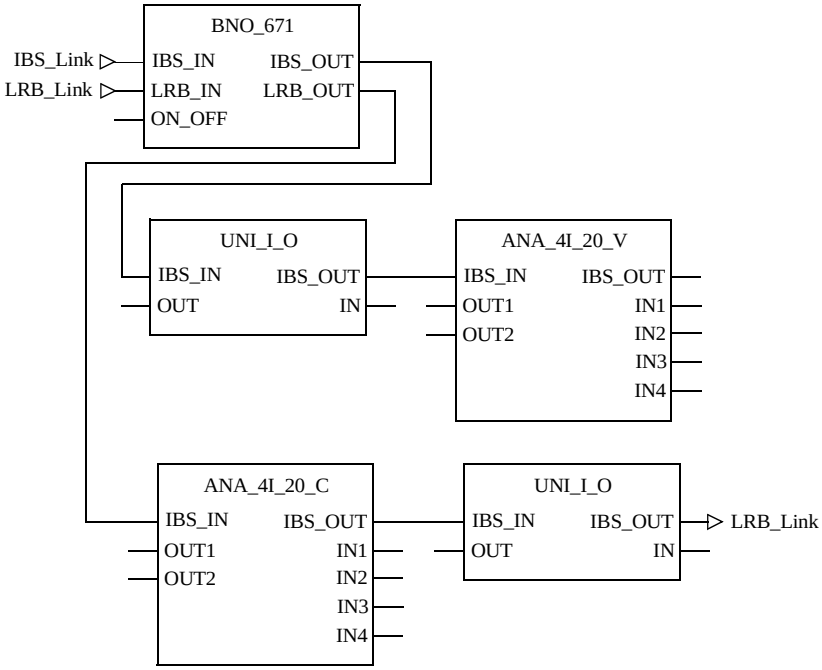
Structure with BNO_671

A possible InterBus-S structure with the bus terminal module is illustrated.

Structure with BNO_671 - Part 1



Structure with BNO_671 - Part 2



COMPACT: Configuring a main rack

30

Overview

Introduction

This chapter describes the block COMPACT.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

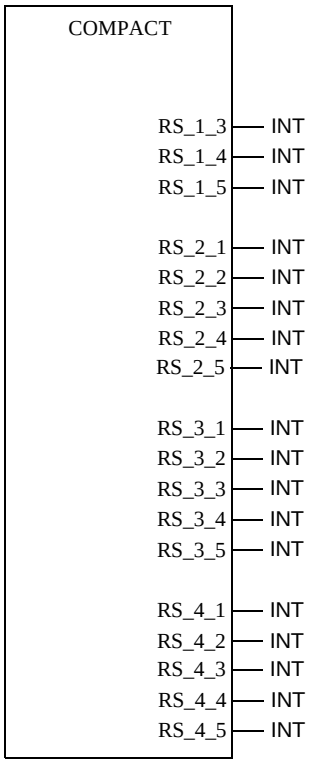
Function description

The Function block is used to edit the configuration data of a Compact primary backplane for subsequent use by the scaling EFBs.
To configure a COMPACT primary backplane, the COMPACT function block is inserted into the configuration section. The function block for configuring the analog module is connected to the SLOT outputs.
EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
RS_1_3	INT	Rack 1, slot 3
RS_1_4	INT	Rack 1, slot 4
RS_1_5	INT	Rack 1, slot 5
RS_2_1	INT	Rack 2, slot 1
:	:	:
RS_2_5	INT	Rack 2, slot 5
RS_3_1	INT	Rack 3, slot 1
:	:	:
RS_3_5	INT	Rack 3, slot 5
RS_4_1	INT	Rack 4, slot 1
:	:	:
RS_4_5	INT	Rack 4, slot 5

Runtime error

Runtime error

Internal I/O map errors will cause an Error message.

DAU202: Configuring the Compact module DAU 202 / DAU 252 / DAU 282

31

Overview

Introduction

This chapter describes the block DAU202.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

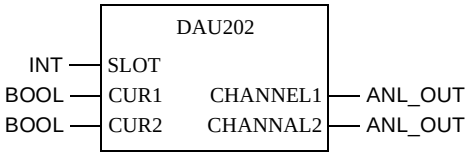
Function description

The function block is used to edit the configuration data of a DAU 202 / DAU 252 / DAU 282 Compact module for subsequent use by the scaling EFBs.
This module has two bipolar output channels for mixed voltage and current processing.
For the configuration of a DAU 202 / DAU 252 / DAU 282, the function block in the Configuration section will be connected to the corresponding SLOT output of the COMPACT Function block. The 3x references and 4x references specified in the I/O map are automatically assigned internally to the individual channels and can therefore only be occupied by Unlocated variables.
The analog values can be further processed in Scaling sections using the O_DEBUG, O_NORM, O_SCALE, O_PHYS, O_DBSET, O_RAW Function blocks.
EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
SLOT	INT	Slot
CUR1	BOOL	0: Channel 1 processing voltage 1: Channel 1 processing current
CUR2	BOOL	0: Channel 2 processing voltage 1: Channel 2 processing current
CHANNEL1	ANL_OUT	Channel 1
CHANNEL2	ANL_OUT	Channel 2

Runtime error

Runtime error

If no DAU 202 / DAU 252 / DAU 282 module has been configured for the specified SLOT input, an error message will appear.
Status information cannot be requested for this module.

DAU204: Configuring the Compact module DAU 204

32

Overview

Introduction

This chapter describes the block DAU204.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

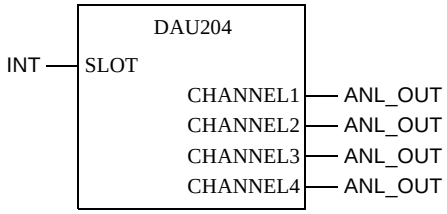
Function description

The function block is used to edit the configuration data of a DAU 204 Compact module for subsequent use by the scaling EFBs.
This module has four output channels for combined bipolar and unipolar voltage and current processing.
For the configuration of an DAU 204, the function block in the Configuration section is connected to the corresponding SLOT output of the COMPACT Function block.
The 4x references specified in the I/O map (as well as the 3 references for the status information) are automatically assigned internally to the individual channels and can therefore only be occupied by Unlocated variables.
The analog values can be further processed in Scaling sections using the O_DEBUG, O_NORM, O_SCALE, O_PHYS, O_DBSET, O_RAW function blocks.
EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
SLOT	INT	Slot
CHANNEL1	ANL_OUT	Channel 1
CHANNEL2	ANL_OUT	Channel 2
CHANNEL3	ANL_OUT	Channel 3
CHANNEL4	ANL_OUT	Channel 4

Runtime error

Runtime error

If no DAU 204 module has been configured for the specified SLOT input, an error message appears.
A range warning for the channels is not provided by the processing Function blocks (See *Function description, p. 164*).
The status information "Open circuit on channel" and "Open circuit on one or several channels" can be collected via the status register (3x reference) defined in the I/O map.

DAU208: Configuring the Compact module DAU 208

33

Overview

Introduction

This chapter describes the block DAU208.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

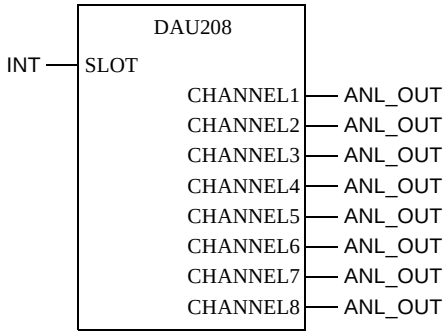
Function description

The function block is used to edit the configuration data of a DAU 208 Compact module for subsequent use by the scaling EFBs.
This module has eight bipolar output channels for voltage processing.
For the configuration of an DAU 208, the function block in the Configuration section is connected to the corresponding SLOT output of the COMPACT Function block.
The 3x references and 4x references specified in the I/O map are automatically assigned internally to the individual channels and can therefore only be occupied by Unlocated variables.
The analog values can be further processed in Scaling sections using the O_DEBUG, O_NORM, O_SCALE, O_PHYS, O_DBSET, O_RAW function blocks.
EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
SLOT	INT	Slot
CHANNEL1	ANL_OUT	Channel 1
CHANNEL2	ANL_OUT	Channel 2
CHANNEL3	ANL_OUT	Channel 3
CHANNEL4	ANL_OUT	Channel 4
CHANNEL5	ANL_OUT	Channel 5
CHANNEL6	ANL_OUT	Channel 6
CHANNEL7	ANL_OUT	Channel 7
CHANNEL8	ANL_OUT	Channel 8

Runtime error

Runtime error

If no DAU 208 module has been configured for the specified SLOT input, an error message appears.
Status information cannot be requested for this module.

DIG_16I: Configuring the TIO-module BDI 346 00 / 546 50 / 746 50

34

Overview

Introduction

This chapter describes the block DIG_16I.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

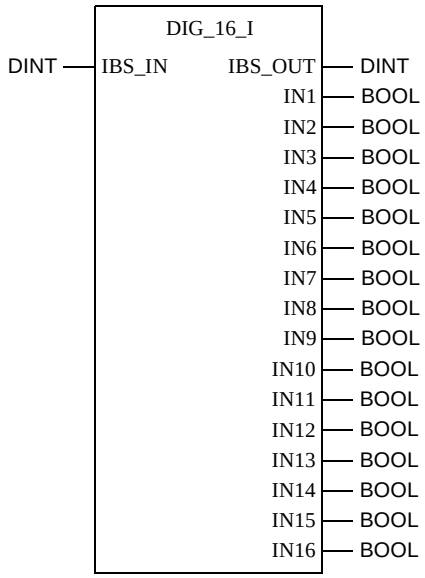
Function description

The DIG_16I Function block is a software connection from InterBus-S hardware modules with 16 binary inputs.
The function block corresponds to the hardware modules TIO/IS 170 BDI 346 00, TIO/IS 170 BDI 546 50 and TIO/IS 170 BDI 746 50. A different designation to that of the hardware was selected in order to clarify the relationship between the name and the function of the module.
EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
IBS_IN	DINT	Incoming InterBus-S
IBS_OUT	DINT	Outgoing InterBus-S
IN1	BOOL	Input 1 of the TIO
IN2	BOOL	Input 2 of the TIO

Parameter	Data type	Meaning
:	:	:
IN16	BOOL	Input 16 of the TIO

Detailed description

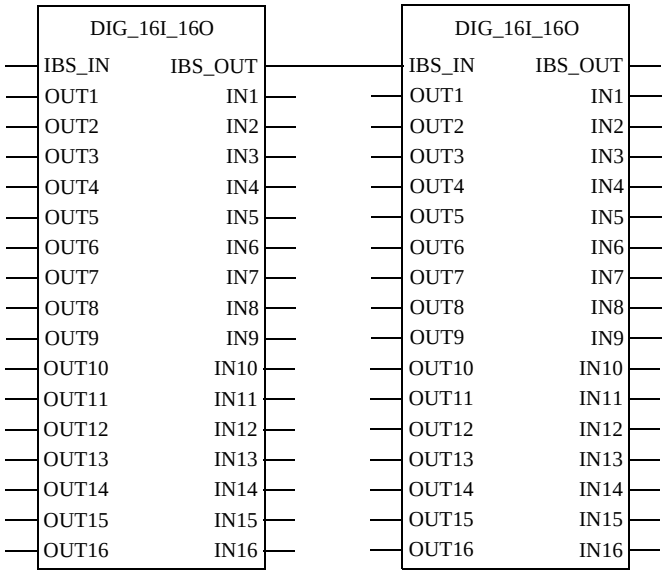
Detailed description

The function block occupies one input word in the Status-RAM.

Parameter description - inputs

IBS_IN

IBS_IN = Connection for the incoming remote bus part of InterBus-S
On the hardware, the male connector is on the top left of the module.
The module is connected to the outgoing master (1st module on the bus) remote bus (IBS_OUT) or the previous module (see also diagram). The link can be made via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two bus devices.
Connection of two InterBus-S modules



Parameter description - Outputs

IBS_OUT	IBS_OUT = Connection for the outgoing remote bus part of InterBus-S On the hardware, the male connector is on the top right of the module. The module is connected to the incoming remote bus (IBS_IN) of the following module, either via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two InterBus-S modules.
INx	INx = Input x x stands for the digit 1 to 16 which indicates the corresponding input. Binary process values are read into the InterBus-S module via the relevant input (INx).

DIG_16I_12O_MON: Configuring the module ADM 390 10

35

Overview

Introduction

This chapter describes the block DIG_16I_12O_MON.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

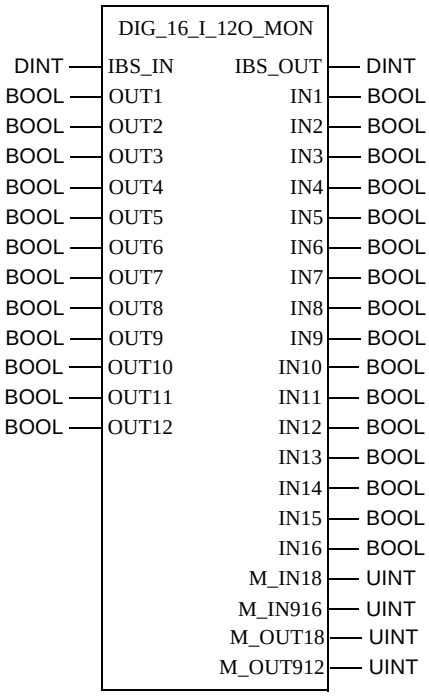
Function description

The DIG_16I_12O_MON Function block is a software connection to InterBus-S Momentum/IS 170 ADM 390 10 hardware modules.
The function block has 16 binary inputs and 12 binary outputs, which can be operated simultaneously or as inputs or outputs only. The status of the I/O is also indicated.
EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameters	Data type	Meaning
IBS_IN	DINT	Incoming InterBus-S
OUT1	BOOL	Output 1 of the module
:	:	:
OUT12	BOOL	Output 12 of the module
IBS_OUT	DINT	Outgoing InterBus-S
IN1	BOOL	Input 1 of the module
:	:	:
IN16	BOOL	Input 16 of the module
M_IN18	UDINT	Status indicator for inputs 1 to 8
M_IN916	UDINT	Status indicator for inputs 9 to 16
M_OUT18	UDINT	Status indicator for outputs 1 to 8
M_OUT912	UDINT	Status indicator for outputs 9 to 12

Detailed description

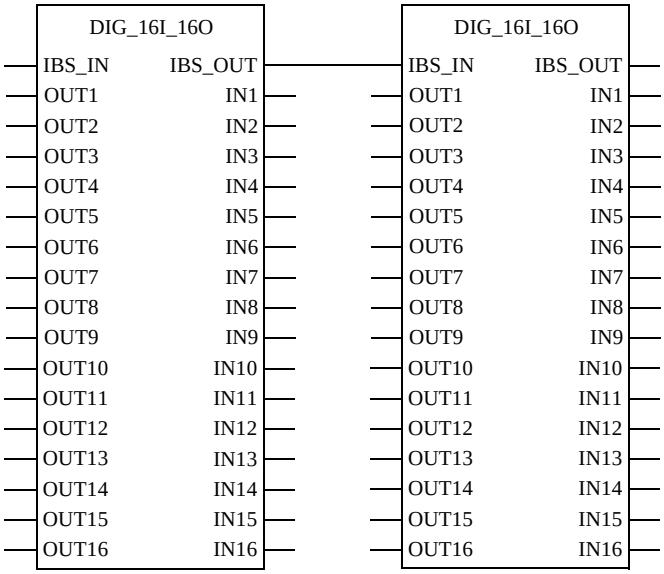
Detailed description

The function block occupies 3 input words and 3 output words in the Status-RAM.

Parameter description - inputs

IBS_IN

IBS_IN = Connection for the incoming remote bus part of InterBus-S
On the hardware, the male connector is on the top left of the module.
The module is connected here to the outgoing remote bus (IBS_OUT) of the master (1st module on the bus) or the preceding module (see diagram). The link can be made via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two bus devices.
Connection of two InterBus-S modules



OUTx = Output x

OUTx = Output x
x stands for the digit 1 to 16 which indicates the corresponding output.
The binary values to be produced via the InterBus-S module are supplied to the process via the relevant output (OUTx).

Parameter description - Outputs

IBS_OUT	IBS_OUT = Connection for the outgoing remote bus part of InterBus-S On the hardware, the male connector is on the top right of the module. The module is connected to the incoming remote bus (IBS_IN) of the following module, either via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two InterBus-S modules.
INx	INx = Input x x stands for the digit 1 to 16 which indicates the corresponding input. Binary process values are read into the InterBus-S module via the relevant input (INx).
M_IN18	M_IN18 = Status indicator for inputs 1 to 8 A number is indicated which refers to the faulty input. Example: M_IN18 = 10001, then input 1 and input 5 are faulty. They are counted from right to left.
M_IN916	M_IN916 = Status indicator for inputs 9 to 16 A number is indicated which refers to the faulty input. Example: M_IN916 = 1001, then input 9 and input 12 are faulty. They are counted from right to left.
M_OUT18	M_OUT18 = Status indicators for outputs 1 to 8 A number is indicated which refers to the faulty output. Example: M_OUT18 = 10001, then output 1 and output 5 are faulty. They are counted from right to left.
M_OUT912	M_OUT912 = Status indicators for outputs 9 to 12 A number is indicated which refers to the faulty output. Example: M_OUT912 = 1001, then output 9 and output 12 are faulty. They are counted from right to left.

DIG_16I_16O: Configuring the TIO-module BDM 346 00

36

Overview

Introduction

This chapter describes the block DIG_16I_16O.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

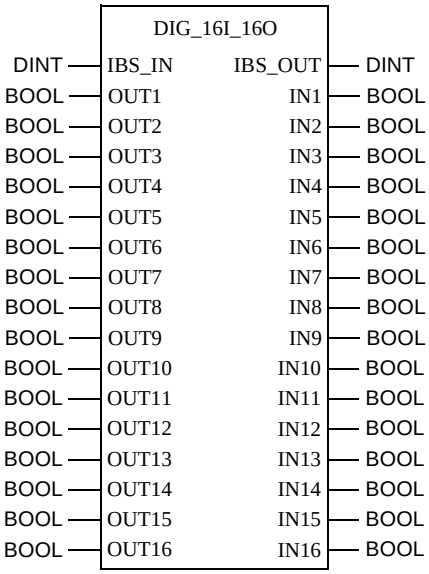
Function description

The DIG_16I_16O Function block in Concept functions in the same way as its hardware counterpart.
However, its operation has been simplified by programming it as a function block in Concept. The module occupies one input word and one output word in the master. EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
IBS_IN	DINT	Incoming InterBus-S
OUT1	BOOL	Output 1 of the TIO
OUT2	BOOL	Output 2 of the TIO
:	:	:
OUT16	BOOL	Output 16 of the TIO

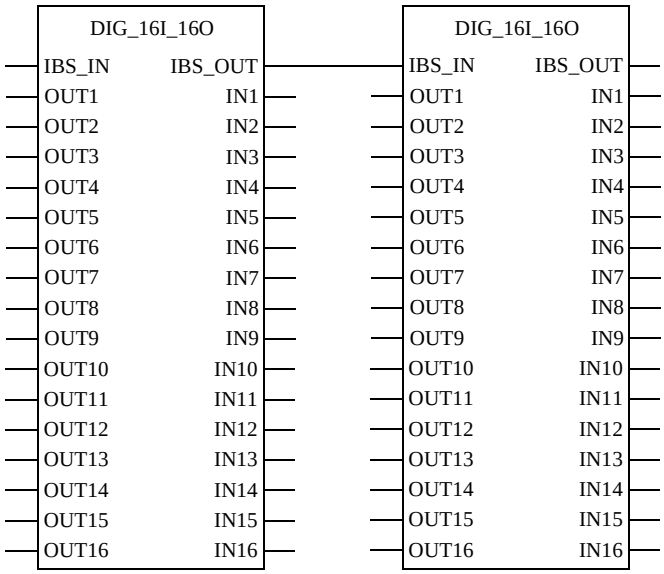
Parameter	Data type	Meaning
IBS_OUT	DINT	Outgoing InterBus-S
IN1	BOOL	Input 1 of the TIO
IN2	BOOL	Input 2 of the TIO
:	:	:
IN16	BOOL	Input 16 of the TIO

Detailed description

Parameter description - inputs

IBS_IN

IBS_IN = Connection for the incoming remote bus part of InterBus-S
On the hardware, the male connector is on the top left of the module.
The module is connected here to the outgoing remote bus (IBS_OUT) of the master (1st module on the bus) or the preceding module (see diagram). The link can be made via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two bus devices.
Connection of two InterBus-S modules



OUTx = Output x	OUTx = Output x x stands for the digit 1 to 16 which indicates the corresponding output. The binary values to be produced via the InterBus-S module are supplied to the process via the relevant output (OUTx).
Parameter description - Outputs	
IBS_OUT	IBS_OUT = Connection for the outgoing remote bus part of InterBus-S On the hardware, the male connector is on the top right of the module. The module is connected to the incoming remote bus (IBS_IN) of the following module, either via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two InterBus-S modules.
INx	INx = Input x x stands for the digit 1 to 16 which indicates the corresponding input. Binary process values are read into the InterBus-S module via the relevant input (INx).

DIG_16O: Configuring the TIO-modules BDO 346 00 / BDO 946 50

37

Overview

Introduction

This chapter describes the block DIG_16O.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

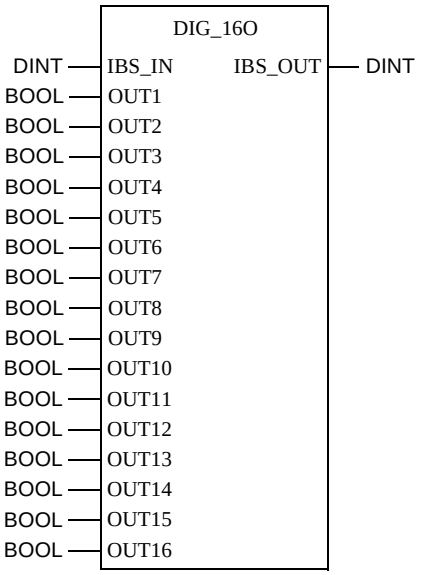
Function description

The DIG_16O Function block is a software connection to InterBus-S hardware modules with 16 binary outputs.
The function block corresponds to the hardware TIO/IS 170 BDO 346 00 and TIO/IS 170 BDO 946 50. A different designation was selected in order to achieve a clearer relation between the name and the function of the module.
EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
IBS_IN	DINT	Incoming InterBus-S
OUT1	BOOL	Output 1 of the TIO
OUT2	BOOL	Output 2 of the TIO
:	:	:

Parameter	Data type	Meaning
OUT16	BOOL	Output 16 of the TIO
IBS_OUT	DINT	Outgoing InterBus-S

Detailed description

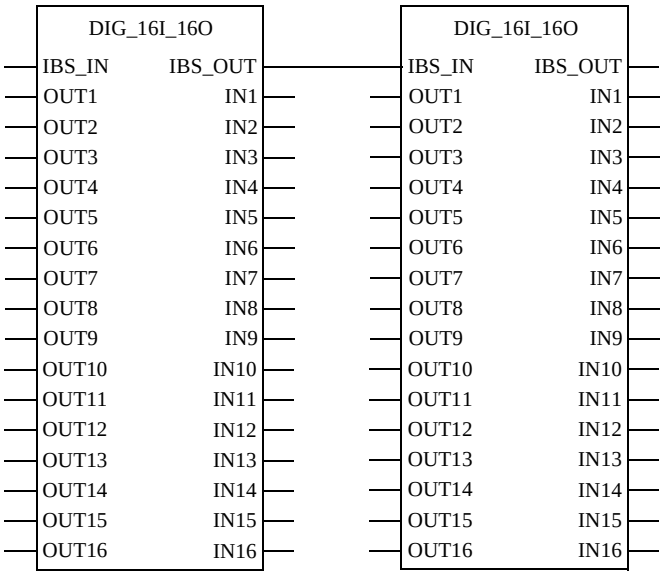
Detailed description

The function block occupies one output word in the Status-RAM.

Parameter description - inputs

IBS_IN

IBS_IN = Connection for the incoming remote bus part of InterBus-S
On the hardware, the male connector is on the top left of the module.
The module is connected here to the outgoing remote bus (IBS_OUT) of the master (1st module on the bus) or the preceding module (see diagram). The link can be made via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two bus devices.
Connection of two InterBus-S modules



OUTx	OUTx = Output x x stands for the digit 1 to 16 which indicates the corresponding output. The binary values to be produced via the InterBus-S module are supplied to the process via the relevant output (OUTx).
Parameter description - Outputs	
IBS_OUT	IBS_OUT = Connection for the outgoing remote bus part of InterBus-S On the hardware, the male connector is on the top right of the module. The module is connected to the incoming remote bus (IBS_IN) of the following module, either via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two InterBus-S modules.

DROP: Configuring a I/O Station subrack

38

Overview

At a Glance

This chapter describes the block DROP.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

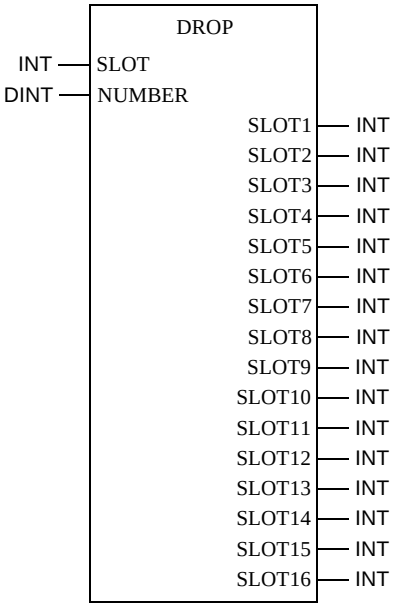
Function description

The Function block is used to edit the configuration data of a remote or distributed I/O station for subsequent processing by module configuration EFBs.
To configure an I/O station subrack, the DROP Function block in the configuration section is connected to the corresponding SLOT output of the QUANTUM Function block. The number of the I/O station defined in the I/O map has to be entered at the NUMBER input of the DROP Function block. The Function blocks for configuration of the analog modules of the I/O stations are connected to the SLOT outputs.
EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameters	Data type	Meaning
SLOT	INT	Slot for RIO, DIO, NOM
NUMBER	DINT	Number of RIO, DIO, NOM
SLOT1	INT	Slot 1
:	:	:
SLOT16	INT	Slot 16

Runtime error

Runtime error

If no "Head" has been configured for the I/O station subrack, an error message appears.

I_DBSET: Writing internal data structure ANL_IN

39

Overview

At a Glance

This chapter describes the block I_DBSET.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

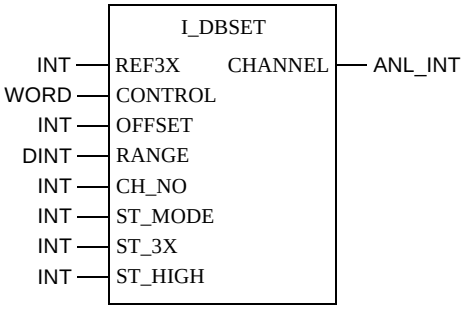
Note: The Function block is not usually needed.

The function block can be used to set information for the input channels (ANL_IN). EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
REF3X	INT	3x raw value register
CONTROL	WORD	Control word (internal use only)
OFFSET	INT	Input null shift
RANGE	DINT	Input range (resolution)
CH_NO	INT	Channel number
ST_MODE	INT	Status mode (internal use only)
ST_3X	INT	3x status register
ST_HIGH	INT	Identifies high byte or low byte of status register
CHANNEL	ANL_IN	Channel to be written

I_DEBUG: Monitoring internal data structure ANL_IN

40

Overview

At a Glance

This chapter describes the block I_DEBUG.

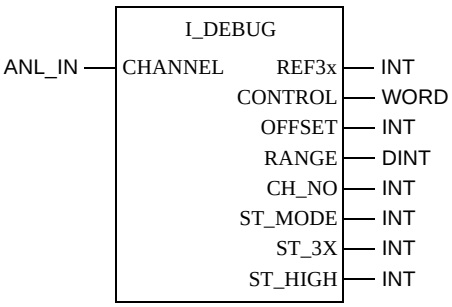
What's in this Chapter?

This Chapter contains the following Maps:

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
CHANNEL	ANL_IN	channel to be monitored
REF3X	INT	3x raw value register
CONTROL	WORD	Control word (internal use only)
OFFSET	INT	Input null shift
RANGE	DINT	Input range (resolution)
CH_NO	INT	Channel number
ST_MODE	INT	Status mode (internal use only)
ST_3X	INT	3x status register
ST_HIGH	INT	Identifies high byte or low byte of status register

Brief description

Function description

Note: The Function block is not usually needed.

The function block can be used to display information for the input channels (ANL_IN).
EN and ENO can be projected as additional parameters.

I_FILTER: Linearization for analog-inputs

41

Overview

At a Glance

This chapter describes the block I_FILTER.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

The function enables the adjustment of characteristic curves for analog input values.

3 different adjustments are available:
- Linearizing with square root (standardized range)
- Correction of the "Offset" (zero offset compensation)
- Correction of "Range" (gain)

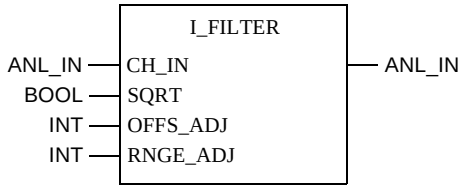
Note: Correction of the automatically set values for "Offset" and "Range" is not normally necessary.

EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

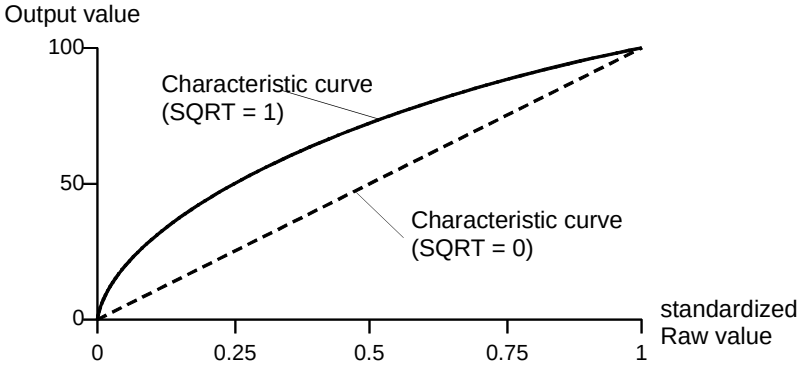
Block parameter description:

Parameter	Data type	Meaning
CH_IN	ANL_IN	Input value
SQRT	BOOL	Square root filter 1: Filter active 0: Filter inactive
OFFS_ADJ	INT	Adjusting offset
RNGE_ADJ	INT	Adjusting gain
OUT	ANL_IN	Output value

Detailed description

Linearizing with square root (standardized range)

Use the parameter SQRT to linearize an analog input value.
The square root filter acts according to the following functions:
f(0) = 0, f(0.5) = 0.707, f(1) = 1.
Characteristic curve of the square root filter



Correction of the "Offset" (zero offset compensation)

The OFFS_ADJ parameters can be used to modify (adjust) the calculated offset value of the output

Note: Correction of the automatically set value (OFFS_ADJ = 0) is not normally necessary. Nevertheless, if corrections are made, they should be monitored using the I_DEBUG Function block, because there will be a modification of the ANL_IN data type (of the output).

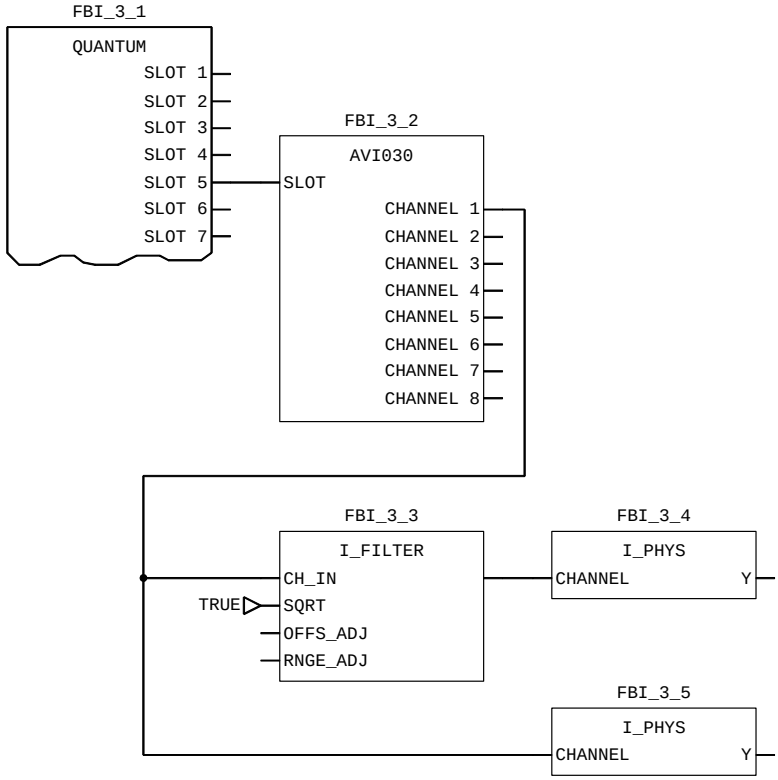
Correction of "Range" (gain)

The RNGE_ADJ parameter can be used to modify (adjust) the calculated gain of the output.

Note: Correction of the automatically set value (RNGE_ADJ = 0) is not normally necessary. If corrections are made, they should be monitored using the I_DEBUG Function block, because there will be a modification of the ANL_IN data type (of the output).

Example

Structure with I_FILTER



The outputs OFFS_ADJ and RNGE_ADJ of the I_FILTER (FBI_3_3) Function block are not used. They are therefore set per default to "0".
The following values apply for function block I_PHYS (FBI_3_4):

Input values (AVI030 10 V)	**Output values (I_PHYS)**
0 V	0.0
2.5 V	5.0
5 V	7.07
10 V	10.0

The following values apply for function block I_PHYS (FBI_3_5):

Input values (AVI030 10 V)	**Output values (I_PHYS)**
0 V	0.0
2.5 V	2.5
5 V	5.0
10 V	10.0

Runtime error

Runtime error

An error message appears if the input channel has not been configured. In this case, please check the connected I/O module EFB.

I_NORM: Standardized analog input

42

Overview

At a Glance

This chapter describes the block I_NORM.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

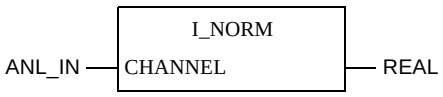
Function description

The function converts data from 16 bit integer format into REAL floating-point format. The configured integer input value is displayed with a floating-point value in the range of 0.0 to 1.0. If there are warning ranges for the current data format (e.g. 16 bit, +/- 10 V), the floating point value can be expanded (e.g. 1.016)
EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
CHANNEL	ANL_IN	Input value
OUT	REAL	Normalized value

Runtime error

Runtime error

AnError messageappears,

- if the input channel is not configured. In this case, please check the connected I/O module EFB.
- with an input value underflow (for example, -1 Volt instead of 0 ... 5 Volt).
- with an input value overflow (for example, 6 Volt instead of 0 ... 5 Volt).

Note: To evaluate the status information for the I/O module, use the function block I_NORM_WARN.

I_NORM_WARN: Standardized analog-input with warning status

43

Overview

At a Glance

This chapter describes the block I_NORM_WARN.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

The function block converts data from 16 bit integer format into REAL floating-point format. The configured integer input value is displayed with a floating-point value in the range of 0.0 to 1.0. If there are warning ranges for the current data format (e.g. 16 bit, +/- 10 V), the floating point value can be expanded (e.g. 1.016)
In addition the function block indicates at the WARN output whether a status warning has occurred in the connected analog input EFB.

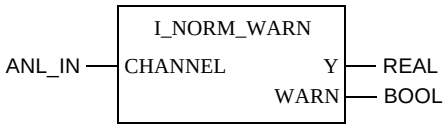
Note: This function block is not compatible with the ADU2xx function for Compact (the I_NORM Function block should be used instead). The I_NORM_WARN function block does not recognize the module range information, even though it is assigned to the 3x register area. Therefore, the area warn bits have to be taken directly.

The layout of the status information assigned to State RAM can be found in Online Help for Compact modules (**Help** → **Help on Compact** → **Compact I/O User's Guide**).
EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
CHANNEL	ANL_IN	Input value
Y	REAL	Normalized value
WARN	BOOL	0: no status warning on the connected analog input EFB 1: status warning on the connected analog input EFB

Runtime error

Runtime error

An error message appears,

- if the input channel is not configured. In this case, please check the connected I/O module EFB.
- with an input value overflow (outside the warning range, e.g. 6Volts instead of 0 ... 5Volts)
- if the connected analog input EFB is unable to generate status information, and the WARN output can, therefore, never become active. In this case, please use the I_NORM Function block.

I_PHYS: Physical analog-input

44

Overview

At a Glance

This chapter describes the block I_PHYS.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

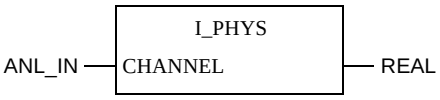
Function description

The Function outputs analog input values (voltage, current or temperature) as physical values in REAL floating-point format.
EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
CHANNEL	ANL_IN	Input value
OUT	REAL	Physical value

Runtime error

Runtime error

An error message appears,

- if the input channel is not configured. In this case, please check the connected I/O module EFB.
- with an input value underflow (for example, -1 Volt instead of 0 ... 5 Volt).
- in the case of an input value overflow (for example, 6 Volt instead of 0 ... 5 Volt).

Note: To evaluate the status information for the I/O module, use the Function block I_PHYS_WARN.

I_PHYS_WARN: Physical analog-input with warning-status

45

Overview

At a Glance

This chapter describes the block I_PHYS_WARN.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

The Function block provides analog input values (voltage, current or temperature) as physical values in REAL floating-point format.
In addition the function block indicates at the WARN output whether a status warning has occurred in the connected analog input EFB.

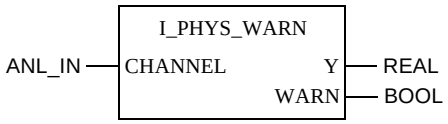
Note: This function block is not compatible with the ADU2xx function for Compact (the I_PHYS function block should be used instead). The I_PHYS_WARN Function block does not recognize the module range information, even though it is assigned to the 3x register area. Therefore, the area warn bits have to be taken directly.

The layout of the status information assigned to State RAM can be found in Online Help for Compact modules (**Help** → **Help on Compact** → **Compact I/O User's Guide**).
EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
CHANNEL	ANL_IN	Input value
Y	REAL	Physical value
WARN	BOOL	0: no status warning on the connected analog input EFB 1: status warning on the connected analog input EFB

Runtime error

Runtime error

An error message appears,

- if the input channel is not configured. In this case, please check the connected I/O module EFB.
- with an input value underflow (outside the warning range, e.g. -1Volt instead of 0 ... 5Volts).
- with an input value overflow (outside the warning range, e.g. 6Volts instead of 0 ... 5Volts).
- if the connected analog input EFB is unable to generate status information, and the WARN output can, therefore, never become active. In this case, please use the I_PHYS Function block.

I_RAW: Raw value analog input

46

Overview

At a Glance

This chapter describes the block I_RAW.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

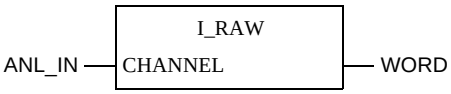
Function description

The function provides analog input values as raw values of the WORD data type. EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
CHANNEL	ANL_IN	Input value
OUT	WORD	Raw value

Runtime error

Runtime error

An error message appears,

- if the input channel is not configured. In this case, please check the connected I/O module EFB.
- if there is an input range violation.

I_RAWSIM: Simulated raw value analog input

47

Overview

At a Glance

This chapter describes the block I_RAWSIM.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

The Function block simulates raw value analog inputs on 3x registers. The function block acts to supplement for the reference data editor where 3x registers cannot be written.

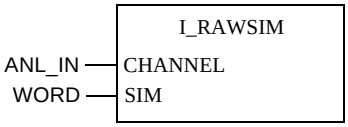
Note: Specify the processing sequence for the function blocks in a way that ensures the I_RAWSIM Function block will be executed before all the other function blocks which read the simulated raw value. To do this, connect the ENO output of the I_RAWSIM with the EN inputs of all the function blocks which read the simulated raw value.

EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
CHANNEL	ANL_IN	Simulated raw value
SIM	WORD	Input value

Runtime error

Runtime error

An error message appears,

- if the input channel is not configured. In this case, please check the connected I/O module EFB.

I_SCALE: Scaled analog input

48

Overview

At a Glance

This chapter describes the block I_SCALE.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

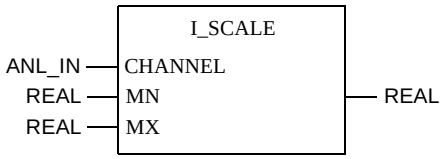
The function converts data from 16 bit integer format into REAL floating-point format. The scaling inputs MN and MX predefine the value range for the output. MN corresponds to 0 percent and MX to 100 percent. The integer input value is displayed in the floating-point range. If there are warning ranges for the current data format (e.g. 16 bit, +/- 10 V), the floating point value can be expanded to over 100 percent (e.g. 101.8 percent)
EN and ENO can be projected as additional parameters.

Note: The I_SCALE function can not be used to scale temperature measurements. Please use the I_PHYSfunction to scale temperature measurements:.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
CHANNEL	ANL_IN	Input value
MN	REAL	Scaling input, 0 percent
MX	REAL	Scaling input, 100 percent
OUT	REAL	Scaled value

Runtime error

Runtime error

An error message appears,

- if the input channel is not configured. In this case, please check the connected I/O module EFB.
- in the case of input value underflow (for example, -1 Volt instead of 0 ... 5 Volt).
- in the case of input value overflow (for example, 6 Volt instead of 0 ... 5 Volt).

Note: To evaluate the status information for the I/O module, use the I_SCALE_WARNFunction block.

I_SCALE_WARN: Scaled analog input with warnings status

49

Overview

At a Glance

This chapter describes the block I_SCALE_WARN.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

The function block converts data from 16 bit integer format into REAL floating-point format. The scaling inputs MN and MX predefine the value range for the output. MN corresponds to 0 percent and MX to 100 percent. The integer input value is displayed in the floating-point range. If there are warning ranges for the current data format (e.g. 16 bit, +/- 10 V), the floating point value can be expanded to over 100 percent (e.g. 101.6 percent)
In addition the function block indicates at the WARN output whether a status warning has occurred in the connected analog input EFB.

Note: This function block is not compatible with the ADU2xx function for Compact (the I_SCALE function block should be used instead). The I_SCALE_WARN Function block does not recognize the module range information, even though it is assigned to the 3x register area. Therefore, the area warn bits have to be taken directly.

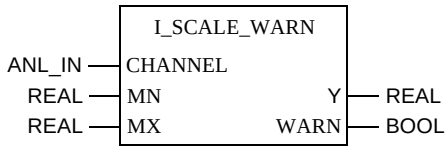
The layout of the status information assigned to State RAM can be found in Online Help for Compact modules (**Help** → **Help on Compact** → **Compact I/O User's Guide**).
EN and ENO can be configured as additional parameters.

Note: The I_SCALE_WARN function can not be used to scale temperature measurements. Please use the I_PHYS_WARNfunction to scale temperature measurements.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
CHANNEL	ANL_IN	Input value
MN	REAL	Scaling input, 0 percent
MX	REAL	Scaling input, 100 percent
Y	REAL	Scaled value
WARN	BOOL	0: no status warning on the connected analog input EFB 1: status warning on the connected analog input EFB

Runtime error

Runtime error

An error message appears,

- if the input channel is not configured. In this case, please check the connected I/O module EFB.
- with an input value underflow (outside the warning range, e.g. -1Volt instead of 0 ... 5Volts).
- with an input value overflow (outside the warning range, e.g. 6Volts instead of 0 ... 5Volts).
- if the connected analog input EFB is unable to generate status information, and the WARN output can, therefore, never become active. In this case, please use the I_SCALE Function block.

I_SET: Set information from analog input channels

50

Overview

At a Glance

This chapter describes the block I_SET.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

The function block sets the informations for the analog input channels (ANL_IN). This block enables all scaling blocks of this library to be used.

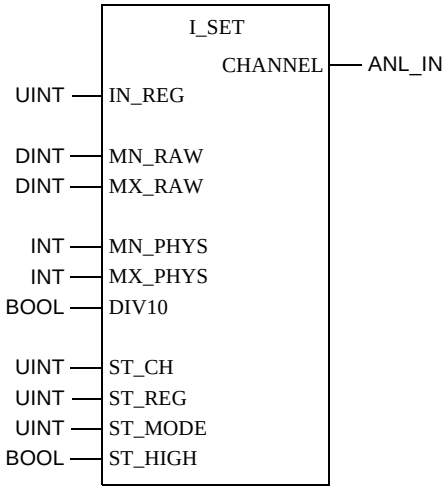
Note: The function block is only required if there is no specific block for a specific analog module available.

EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
IN_REG	UINT	Number of the raw value register (3X)
MN_RAW	DINT	0 % raw value (e.g. 768)
MX_RAW	DINT	100% raw value (e.g. 64768)
MN_PHYS	INT	lowest input value (e.g. -10 V as -10)
MX_PHYS	INT	greatest input value (e.g. +10 V as 10)
DIV10	BOOL	MN_PHYS and MX_PHYS divided by 10
ST_CH	UINT	channel number (1n) (e.g. 4)
ST_REG	UINT	Number of the status register (3X)
ST_MODE	UINT	Status mode (e.g. 1=AVI_STATUS_MODE)
ST_HIGH	BOOL	Status byte found in highbyte of the register
CHANNEL	ANL_IN	channel information to be described

Detailed description

Area of application

The function block can be used in three areas:

1. Raw value scaling, with the block: I_NORM and I_SCALE
2. Scaling in physical units, with the I_PHYS block
3. Evaluation of error information, with the blocks I_NORM, I_SCALE and I_PHYS and additional evaluation of status information (warnings) using the I_..._WARN block

Basic circuit connections

The input IN_REG must always be connected with the number of an input word (3x).

Raw value scaling

For raw value scaling, the inputs MN_RAW (minimum raw value, corresponds to 0%) and MX_RAW (maximum raw value, corresponds to 100%) must also be connected.

Scaling in physical units

For scaling in physical units the inputs MN_PHYS and MX_PHYS must also be connected.
DIV10 is an auxiliary input in the range 0.2 V ... 1 V floating point format to avoid. Set MN_PHYS=2, MX_PHYS=10 and DIV10=1 for this range.
For most ranges this input can remain open (or be assigned 0).
e.g. +/-20 mA: here is MN_PHYS=-20, MX_PHYS=20
The input value ranges supported by I_SET can be found in the section *Supported Value Ranges, p. 230*.

Evaluation of error information

For the evaluation of error information the inputs ST_CH, ST_REG and ST_MODE and ST_HIGH must also be configured.
ST_HIGH is an auxiliary input in case the status byte (status information) is located in the registers high byte. For most ranges this input can remain open (or be assigned FALSE).
The input channel number (1 ... n) is given to ST_CH.
If ST_CH is entered, ST_REG and ST_MODE must also be entered.
ST_REG must be connected with the number of an input word (3X), where the status information is located (error and/or warnings).
ST_MODE determines how the status word is evaluated.
The following 8 modes are defined:

Value	Mode	see also module description for
1	AVI_STATUS_MODE	AVI030
2	ACI_STATUS_MODE	ACI030
3	ACO_STATUS_MODE	ACO030
4	ADU_STATUS_MODE	ADU204
5	DAU204_STATUS_MODE	DAU204
6	ADU205_STATUS_MODE	ADU205
7	AMM090_STATUS_MODE	AMM090
8	ADU214_STATUS_MODE	ADU214

Supported Value Ranges

Voltage

Unipolar

Value range	MN_PHYS	MX_PHYS	DIV10
0 ... 0.5 V	0	5	1
0 ... 1.0 V	0	10	1
0 ... 5.0 V	0	5	0
0 ... 10 V	0	10	0
0 ... 20 V	0	20	0
0,1 ... 0.5 V	1	5	1
0,2 ... 1.0 V	2	10	1
1,0 ... 5.0 V	1	5	0
2,0 ... 10, 0 V	2	10	0

Bipolar

Value range	MN_PHYS	MX_PHYS	DIV10
+/- 25 mV	-25	25	0
+/-100 mV	-100	100	0
+/-0.5 V	-5	5	1
+/- 1 V	-1	1	0
+/-5 V	-5	5	0
+/-10 V	-10	10	0
+/-20 V	-20	20	0

Current

Unipolar

Value range	MN_PHYS	MX_PHYS	DIV10
0 .. 20 mA	0	20	0
4 ... 20 mA	4	20	0

Bipolar

Value range	MN_PHYS	MX_PHYS	DIV10
+/-20 mA	-20	20	0
+/-40 mA	-40	40	0

Resistance

Unipolar

Value range	MN_PHYS	MX_PHYS	DIV10
0 .. 400 Ω	0	400	0
0 .. 500 Ω	0	500	0
0 .. 766,6 Ω	0	7666	1
0 .. 1 kΩ	0	1000	0
0 .. 2 kΩ	0	2000	0
0 .. 4 kΩ	0	4000	0

Runtime error

Runtime error

The following error messages can be triggered:

Error message	Meaning
E_EFB_USER_ERROR_1	The input IN_REG is not connected with the number of an input word (3x).
E_EFB_USER_ERROR_2 with the parameters of the faulty number	The input IN_REG is connected with an invalid number of an input word (3x).
E_EFB_USER_ERROR_3 with parameter MN_RAW	MN_RAW ≥ MX_RAW)
E_EFB_USER_ERROR_4 with parameter MN_PHYS	Unknown value for MN_PHYS
E_EFB_USER_ERROR_5 with parameter MX_PHYS	Unknown value for MX_PHYS
E_EFB_USER_ERROR_11	ST_REG not entered
E_EFB_USER_ERROR_12	ST_REG too large
E_EFB_USER_ERROR_13	ST_CH not entered

IMIO_IN: Immediate I/O module input

51

Overview

At a Glance

This chapter describes the block IMIO_IN.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

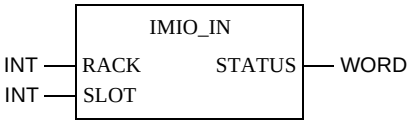
Function description

This Function block reads in I/O module signals immediately during processing. The input module must be in the local rack of the PLC.
EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameters	Data type	Meaning
RACK	INT	Subrack number (Quantum: 1; Compact: 1 ... 4)
SLOT	INT	Slot number (Quantum: 1...16; Compact: 1 ... 5)
STATUS	WORD	Status report

Detailed description

Detailed description

The input of signals takes place directly during block processing as well as during normal I/O processing at the end of a cycle.
The input module must be in the local rack of the PLC. It must also be entered into the I/O map of its configuration. The I/O module is addressed using subrack number and slot number.

Parameter description

The STATUS parameter may contain the following messages:

Status	Meaning
0000	Operation OK
2001	invalid operation type (e.g. the I/O module addressed is not an input module)
2002	Invalid rack or slot number (I/O map in the configurator contains no module entry for this slot)
2003	invalid slot number
F001	Module not OK

Runtime error

Runtime error

The ENO parameter can be used for error display:

ENO	Meaning
1	Operation OK (STATUS equals "0")
0	Operation OK (STATUS not equal to "0")

IMIO_OUT: Immediate I/O module output

52

Overview

At a Glance

This chapter describes the block IMIO_OUT.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

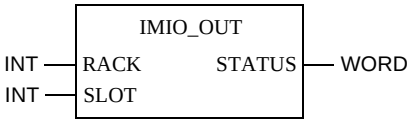
Function description

This Function block supplies the I/O module signals immediately during processing. The output module must be in the local rack of the PLC.
EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameters	Data type	Meaning
RACK	INT	Subrack number (Quantum: 1; Compact: 1 ... 4)
SLOT	INT	Slot number (Quantum: 1...16; Compact: 1 ... 5)
STATUS	WORD	Status report

Detailed description

Detailed description

The output of signals takes place immediately during block processing as well as during normal I/O processing at the end of a cycle.
The output module must be in the local rack of the PLC. It must also be entered into the I/O map of its configuration. The I/O module is addressed using subrack number and slot number.

Parameter description

Status report STATUS
The STATUS parameter may contain the following messages:

Status	Meaning
0000	Operation OK
2001	invalid operation type (e.g. the I/O module addressed is not an input module)
2002	Invalid rack or slot number (I/O map in the configurator contains no module entry for this slot)
2003	invalid slot number
F001	Module not OK

Runtime error

Runtime error

The ENO parameter can be used for error display:

ENO	Meaning
1	Operation OK (STATUS equals "0")
0	Operation OK (STATUS not equal to "0")

MIX_4I_2O: Configuring the AMM module 090 00

53

Overview

At a Glance

This chapter describes the block MIX_4I_2O.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

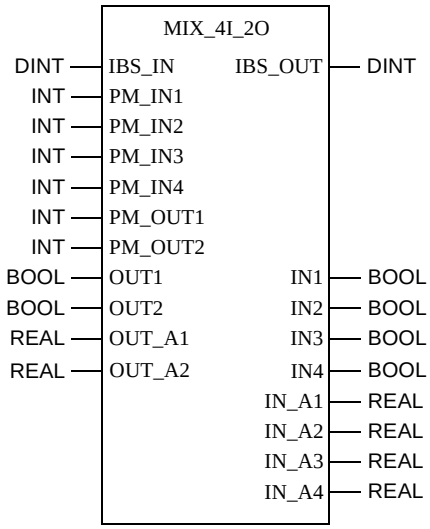
Function description

The MIX_4I_2O Function block is a software connection to an InterBus-S Momentum/IS 170 AMM 090 00 hardware module.
The function block has 4 analog inputs and 2 analog outputs, as well as 4 binary inputs and 2 binary outputs. The function block must be parametered in the same way as the hardware module.
The parameters EN and ENO can additionally be projected.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
IBS_IN	DINT	Incoming InterBus-S
PM_IN1	INT	Parameter input 1
:	:	:
PM_IN4	INT	Parameter input 4
PM_OUT1	INT	Parameter output 1
PM_OUT2	INT	Parameter output 2
OUT1	BOOL	Digital output 1
OUT2	BOOL	Digital output 2
OUT_A1	REAL	Analog output 1 of the module
OUT_A2	REAL	Analog output 2 of the module
IBS_OUT	DINT	Outgoing InterBus-S
IN1	BOOL	Digital input 1
:	:	:
IN4	BOOL	Digital input 4
IN_A1	REAL	Analog input 1 of the module
:	:	:
IN_A4	REAL	Analog input 4 of the module

Detailed description

Detailed description

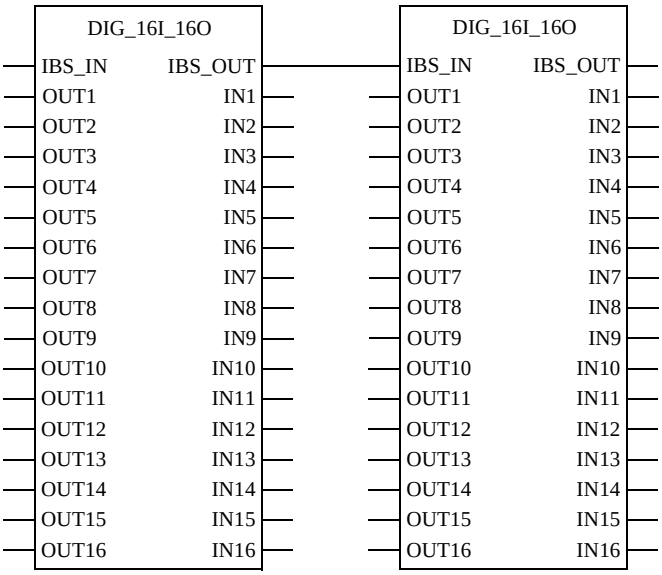
The function block occupies 5 input words and 5 output words in the Status-RAM.

Parameter description - inputs

IBS_IN

IBS_IN = Connection for the incoming remote bus part of InterBus-S
On the hardware, the male connector is on the top left of the module.
The module is connected here to the outgoing remote bus (IBS_OUT) of the master (1st module on the bus) or the preceding module (see diagram). The link can be made via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two bus devices.

Connection of two InterBus-S modules



PM_INx

PM_INx = Parameters for the input channels
x stands for the digit 1 to 4 which indicates the particular input channel.
These parameters are used to parameter the input channels.
The meaning of the parameter values can be found in the table below.

Parameter value	**Meaning**
2	+/- 20mA (+/- 5 V, when divided by 4)
3	+/- 10V
4	Channel inactive
10	420mA (1...5 V, when divided by 4)

A = Output after bus interrupt

Note: All other parameter values are reserved.

Example:
Input 3 should 4 ...20mA.
PM_IN3 = "10"

Note: The reserved parameter codes are not accepted by the module, i.e. the last parameter used will still apply. The default parameters apply until a valid new parameter is entered.

PM_OUTx

PM_OUTx = Parameters for the output channels
x stands for the digit 1 or 2 which indicates the particular output channel.

Example:
PM_OUT2 = Parameters for output channel 2
These parameters are used to parameterize the output channels.
The meaning of the parameter values can be found in the table below.

Parameter value	Meaning
1	0...20mA; Timeout A: 0mA
3	+/- 10V; Timeout A: 0V
4	Channel inactive (default)
5	0...20mA; Timeout A: 20mA
7	+/- 10V; Timeout A: +10V
9	0...20mA; Timeout A: freezes
11	+/- 10V; Timeout A: freezes

A = Output after bus interrupt

Note: All other parameter values are reserved.

Example:
Output 1 should be 0 ...20mA and set to 0mA for bus failure.
PM_OUT1 = "1"

Note: The reserved parameter codes are not accepted by the module, i.e. the last parameter used will still apply. The default parameters apply until a valid new parameter is entered.

OUTx

OUTx = Digital output channel x
x stands for the digit 1 or 2 which indicates the particular output channel.
The binary values to be produced via the InterBus-S module are supplied to the process via the relevant output (OUTx).

OUT_Ax

OUT_Ax = Analog output channel x
x stands for the digit 1 or 2 which indicates the particular output channel.
The analog values to be produced via the InterBus-S module are supplied to the process via the relevant output (OUTx).

Note: The values to be applied here are standardized, i.e. given as voltage in volts or as current in milliamperes. The input as current or voltage depends on the parametering of the particular channel.

Parameter description - Outputs

IBS_OUT

IBS_OUT = Connection for the outgoing remote bus part of InterBus-S
In the hardware, the male connector is located in the top right of the module. This is where the module is connected to the incoming remote bus (IBS_IN) of the next module via either a line or a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two InterBus-S modules.

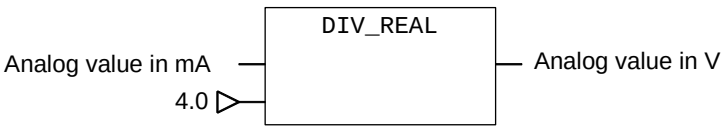
INx

INx = Digital input x
x stands for the digit 1 to 4 which indicates the corresponding input.
Binary process values are read into the InterBus-S module via the relevant input (INx).

IN_Ax

IN_Ax = Analog input channel x
x stands for the number between 1 and 4 designating the corresponding input channel. The analog process values of the InterBus-S module are read via the corresponding input (INx).

Note: The values to be applied here are standardized, i.e. given as voltage in volts or as current in milliamperes. The input as current or voltage depends on the parametering of the particular channel. If a channel is parametered in the +/-5V- or 1...5V range, the incoming values are given in milliamperes. To obtain these values as voltage, divide by 4.0
Scaling an analog value



NOA_611: Configuring the Quantum module NOA 611 00/ NOA 611 10

54

Overview

At a Glance

This chapter describes the block NOA_611.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

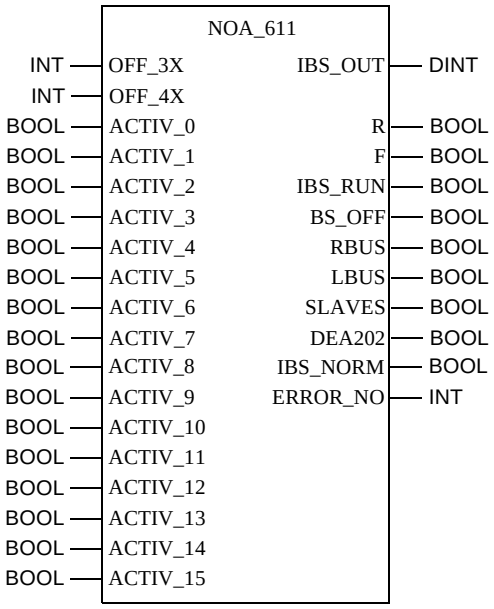
Function description

The Function block NOA_611 is the software connection for an InterBus-S NOA 611 10 master module. It ensures that the data on the InterBus-S is transferred to and read by the corresponding module. The NOA 611 10 controls the bus and monitors operational performance.
The NOA 611 10 occupies 267 input words and 264 output words in the PLC memory. The first input word and the first output word are occupied by the NOA 611 10 itself; it contains the NOA 61110 control bits and status bits. The remaining 256 input words and 256 output words contain the I/O data for the InterBus-S modules. EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
OFF_3X	INT	Offset for 3x address
OFF_4X	INT	Offset for 4x address
ACTIV_0	BOOL	Starts routines which are stored under active bit 0
:	:	:
ACTIV_15	BOOL	Starts routines which are stored under active bit 15
IBS_OUT	DINT	Outgoing InterBus-S
R	BOOL	Master ready
F	BOOL	Error on NOA
IBS_RUN	BOOL	Process data is being exchanged
BS_OFF	BOOL	One or more bus segments are switched off
RBUS	BOOL	Error on remote bus
LBUS	BOOL	Error on local bus
SLAVES	BOOL	InterBus-S device indicates error
DEA202	BOOL	Initialization error on DEA202
IBS_NORM	BOOL	InterBus-S is standardized. All outputs = 0.
ERROR_NO	INT	Number of the faulty InterBus-S module

Detailed description

Parameter description - inputs

OFF_3X and OFF_4X

The parameters of the inputs for the function block are assigned the following module functions.
OFF_3X = Offset 3x address
OFF_4X = Offset 4x address
On the function block, the relevant address offsets for 3x and 4x addresses are given at the two inputs.

Example:
The NOA 611 10 is entered in the PLC configurator, as shown in the table.

Slot	Module	Detected	In.Ref.	In.End	Out.Ref.	Out.End.
1	NOA-611-10		300020	300286	400020	400283

If the initial addresses for the NOA 611 10 are "3:20" for the input words and "4:20" for the output words, then
- OFF_3X = 20 and
- OFF_4X = 20.

ACTIV_x

ACTIV_x = Routine call of active bit x
x stands for the digit 0 to 15 which indicates the particular active bit. A positive transition on ACTIV_x calls the routine stored under ACTIV_x.

Parameter description - Outputs

IBS_OUT

IBS_OUT = Connection for the outgoing remote bus part of InterBus-S
InterBus-S connection on the front panel of the NOA 611. From here, the first module on InterBus-S is connected to the master, via either a line or a variable. For the hardware, the type of connection corresponds to the InterBus-S cable from the master to the first module on InterBus-S.

R

R = NOA 611 10 ready
The NOA 611 10 master module is ready and error-free.

F	F = NOA 611 10 faulty The NOA 611 10 master module is faulty.
IBS_RUN	IBS_RUN = InterBus-S data is being transmitted InterBus-S is operating without error, process data is being exchanged.
BS_OFF	BS_OFF = InterBus-S segment switched off One or more bus segments on InterBus-S are switched off.
RBUS	RBUS = Remote bus error An error has occurred on the remote bus.
LBUS	LBUS = Local bus error An error has occurred on a local bus.
SLAVES	SLAVES = Error on an InterBus-S device Indicates that a device on InterBus-S is faulty.
DEA202	DEA202 = Error on DEA202 Indicates an initialization error on the DEA202.
IBS_NORM	IBS_NORM = InterBus-S standardized The bus is standardized. All outputs on InterBus-S take the value "0".
ERROR_NO	ERROR_NO = Device error number Indicates the device number of a faulty device on the bus. Example: ● Bus connection between device 1 and device 2 interrupted. Display: 2 ● Voltage failure on device 1. Display: 1

Runtime error

Runtime error	An error message (E_INPUT_VALUE_OUT_OF_RANGE) appears if the offset for the 3x or 4x addresses is less than 0 or greater than the maximum permissible value.

O_DBSET: Write internal data structure ANL_OUT

55

Overview

At a Glance

This chapter describes the block O_DBSET.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

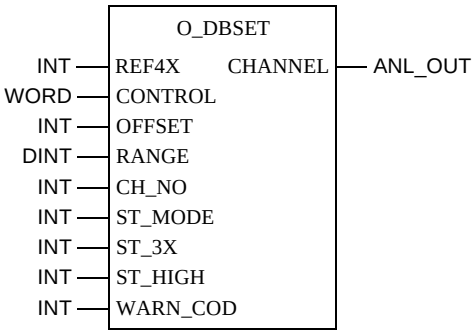
Note: The Function block is not usually needed.

The function block can be used to enter information for the output channels (ANL_OUT).
EN and ENO can be projected as additional parameters.

Representation

Symbol

Module representation:



Parameter description

Module parameter description:

Parameter	Data type	Meaning
REF4X	INT	4x raw value register
CONTROL	WORD	Control word (internal use only)
OFFSET	INT	Input null shift
RANGE	DINT	Input range (resolution)
CH_NO	INT	Channel number
ST_MODE	INT	Status mode (internal use only)
ST_3X	INT	3x status register
ST_HIGH	INT	Identifies high byte or low byte of status register
WARN_COD	INT	Warning mode (internal use only)
CHANNEL	ANL_OUT	Channel to be written

O_DEBUG: Monitoring internal data structure ANL_OUT

56

Overview

At a Glance

This chapter describes the block O_DEBUG.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

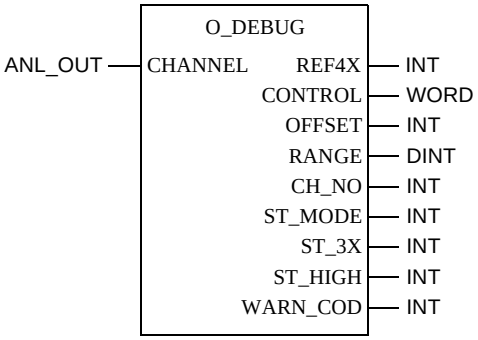
Note: The Function block is not usually needed.

The function block can be used to display information for the output channels (ANL_OUT).
EN and ENO can be projected as additional parameters.

Representation

Symbol

Module representation:



Parameter description

Module parameter description:

Parameter	Data type	Meaning
CHANNEL	ANL_OUT	channel to be monitored
REF3X	INT	4x raw value register
CONTROL	WORD	Control word (internal use only)
OFFSET	INT	Input null shift
RANGE	DINT	Input range (resolution)
CH_NO	INT	Channel number
ST_MODE	INT	Status mode (internal use only)
ST_3X	INT	3x status register
ST_HIGH	INT	Identifies high byte or low byte of status register
WARN_COD	INT	Warning mode (internal use only)

O_FILTER: Linearization for analog outputs

57

Overview

At a Glance

This chapter describes the block O_FILTER.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

The function enables the adjustment of characteristic curves for analog raw values. Different adjustments are available:

- Linearizing with square root (standardized range)
- Correction of the "Offset" (zero offset compensation)
- Correction of "Range" (gain)

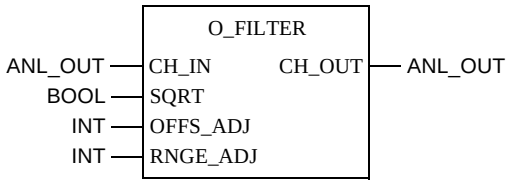
Note: Correction of the automatically set values for "Offset" and "Range" is not normally necessary.

EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

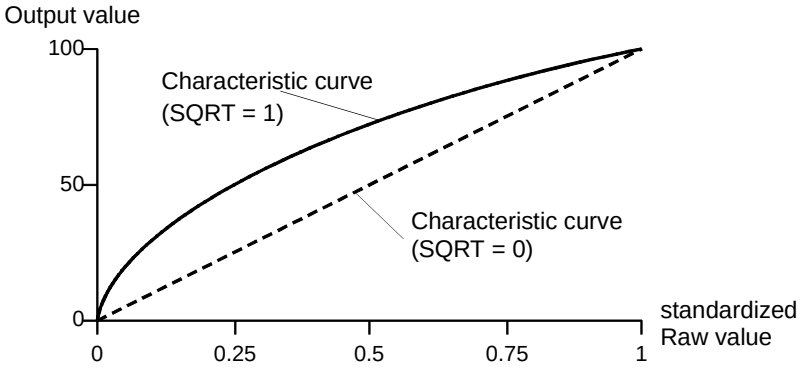
Block parameter description:

Parameter	Data type	Meaning
CH_IN	ANL_OUT	Raw value
SQRT	BOOL	Square root filter 1: Filter active 0: Filter inactive
OFFS_ADJ	INT	Adjusting offsets
RNGE_ADJ	INT	Adjusting gain
CH_OUT	ANL_OUT	Output value

Detailed description

Adjustment with square root (standardized range)

The SQRT parameter can be used to adjust an analog output value. The square root filter acts according to the following functions:
f(0) = 0, f(0.5) = 0.707, f(1) = 1.
Characteristic curve of the square root filter



Correction of the "Offset" (zero offset compensation)

Use the parameter OFFS_ADJ to modify (adjust) the calculated offset value of the output CH_OUT.

Note: Correction of the automatically set value (OFFS_ADJ = 0) is not normally necessary. If corrections are made, they should be monitored using the O_DEBUG Function block, because there will be a modification of the ANL_OUT data type (of the output).

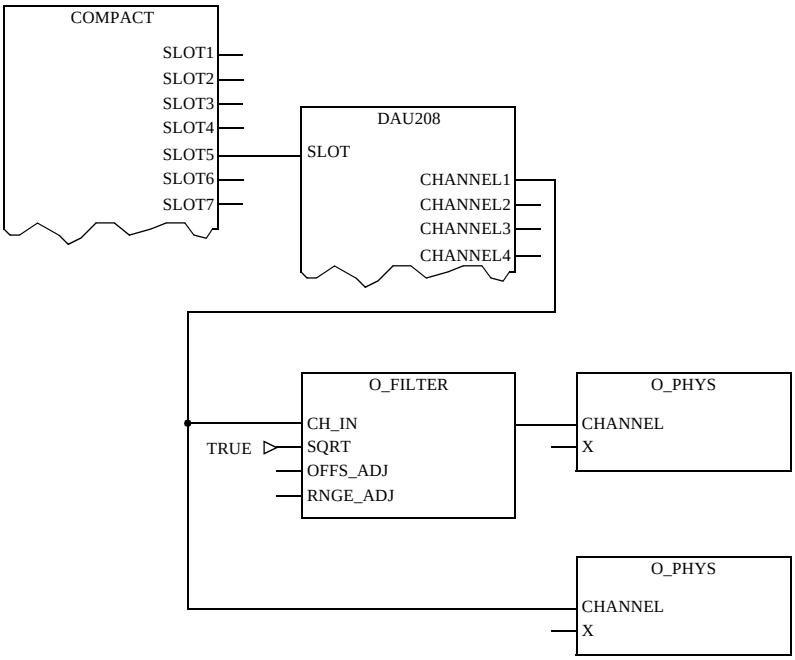
Correction of "Range" (gain)

The RNGE_ADJ parameter can be used to modify (adjust) the calculated gain of the output.

Note: Correction of the automatically set value (RNGE_ADJ = 0) is not normally necessary. If corrections are made, they should be monitored using the O_DEBUG Function block, because there will be a modification of the ANL_OUT data type (of the output)).

Example

Structure with O_FILTER



The outputs OFFS_ADJ and RNGE_ADJ of the O_FILTER (FBI_3_3) Function block are not used. They are set to "0"by default.
The following values apply for function block O_PHYS (FBI_3_4):

Input values (DAU208 10 V)	**Output values (O_PHYS)**
0 V	0.0
2.5 V	5.0
5 V	7.07
10 V	10.0

The following values apply for function block O_PHYS (FBI_3_5):

Input values (DAU208 10 V)	**Output values (O_PHYS)**
0 V	0.0
2.5 V	2.5
5 V	5.0
10 V	10.0

Runtime error

Runtime error	An error message appears if the input channel has not been configured. In this case, please check the connected I/O module EFB.

O_NORM: Standardized analog output

58

Overview

At a Glance

This chapter describes the block O_NORM.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

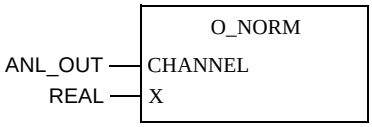
Function description

The Function block outputs values from floating point formatREAL as analog values in 16 bit integer format. The floating point value in the range of 0.0 to 1.0 is displayed onto the configured integer output value.
EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
CHANNEL	ANL_OUT	Output value
X	REAL	Normalized value

Runtime error

Runtime error

An error message appears,

- if the output channel is not configured. In this case, please check the connected I/O module EFB.
- with an output value underflow (arithmetic) (for example, -0.1 V instead of 0 ... 1.0 Volt)
- with an output value overflow (arithmetic) (for example, 1.1 instead of 0 ... 1.0 Volt)

Note: To evaluate the status information for the I/O module, use the O_NORM_WARN function block.

O_NORM_WARN: Standardized analog output with warning status

59

Overview

At a Glance

This chapter describes the block O_NORM_WARN.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

The Function block outputs values from floating point formatREAL as analog values in 16 bit integer format. The floating point value in the range of 0.0 to 1.0 is displayed onto the configured integer output value.
In addition the function block at the WARN_NEG and WARN_POS outputs indicate whether a status warning has occurred in the connected analog output EFB.

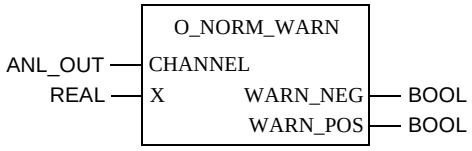
Note: This function block is not compatible with the ADU2xx and DAU2xx functions for Compact (the O_NORM Function block should be used instead). The O_NORM_WARN Function block does not recognize the module range information, even though it is assigned to the 3x register area. Therefore, the area warn bits have to be taken directly.

The layout of the status information assigned to State RAM can be found in Online Help for Compact modules (**Help** → **Help on Compact** → **Compact I/O User's Guide**).
EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
CHANNEL	ANL_OUT	Output value
X	REAL	Normalized value
WARN_NEG	BOOL	0: no output value underflow at the closed analog output EFB 1: output value underflow at the closed analog output EFB
WARN_POS	BOOL	0: no output value overflow at the closed analog output EFB 1: output value overflow at the closed analog output EFB

Runtime error

Runtime error

An error message appears,

- if the output channel is not configured. In this case, please check the connected I/O module EFB.
- with an output value underflow (arithmetic) (outside the warning range, eg. -0.1V instead of 0 ... 1.0V) 1.0Volt)
- with an output value overflow (arithmetic) (outside the warning range, eg. 1,1 instead of 0 ... 1.0V) 1.0Volt)
- if the connected analog output EFB is unable to generate status information and the warning outputs can, therefore, never become active. In this case, please use the O_NORM Function block.

O_PHYS: Physical analog output

60

Overview

At a Glance

This chapter describes the block O_PHYS.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

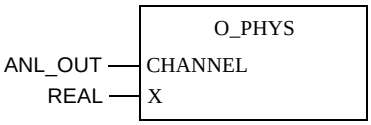
Function description

The Function block provides analog input values (voltage, current or temperature) as physical values in REAL floating-point format.
The function block is in output modules with configuration information (e.g. DAUs).
EN and ENOcan be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
CHANNEL	ANL_OUT	Output value
X	REAL	Physical value

Runtime error

Runtime error

An error message appears,
- if the output channel is not configured. In this case, please check the connected I/O module EFB.
- with an output value underflow (for example, -1 Volt instead of 0 ... 5 Volt).
- with an output value overflow (for example, 6 Volt instead of 0 ... 5 Volt).

Note: To evaluate the status information for the I/O module, use the O_PHYS_WARN function block.

O_PHYS_WARN: Physical analog output with warning-status

61

Overview

At a Glance

This chapter describes the block O_PHYS_WARN.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

The Function block provides analog input values (voltage, current or temperature) as physical values in REAL floating-point format.
The function block is used for output modules with configuration information (e.g. DAUs).
In addition the function block at the WARN_NEG and WARN_POS outputs indicate whether a status warning has occurred in the connected analog output EFB.

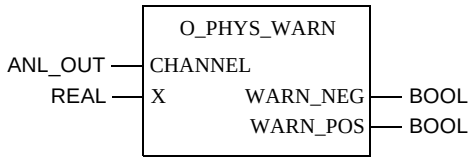
Note: This function block is not compatible with the DAU2xx function for Compact (the O_PHYS function block should be used instead). The O_PHYS_WARN Function block does not recognize the module range information, even though it is assigned to the 3x register area. Therefore, the area warn bits have to be taken directly.

The layout of the status information assigned to State RAM can be found in Online Help for Compact modules (**Help** → **Help on Compact** → **Compact I/O User's Guide**).
EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
CHANNEL	ANL_OUT	Output value
X	REAL	Physical value
WARN_NEG	BOOL	0: no output value underflow at the closed analog output EFB 1: output value underflow at the closed analog output EFB
WARN_POS	BOOL	0: no output value overflow at the closed analog output EFB 1: output value overflow at the closed analog output EFB

Runtime error

Runtime error

An error message appears,

- if the output channel is not configured. In this case, please check the connected I/O module EFB.
- with an output value underflow (outside the warning range, e.g. –1 Volt instead of 0 ... 5 Volt).
- with an output value overflow (outside the warning range, e.g. 6 Volt instead of 0 ... 5 Volt).
- if the connected analog output EFB is unable to generate status information and the warning outputs can, therefore, never become active. In this case, please use the O_PHYS Function block.

O_RAW: Raw value analog output

62

Overview

At a Glance

This chapter describes the block O_RAW.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

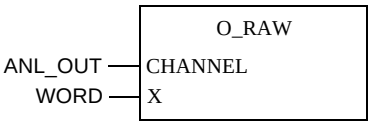
Function description

The Function block provides raw values of the WORD data type as analog output values.
EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameters	Data type	Meaning
CHANNEL	ANL_OUT	Output value
X	WORD	Raw value

Runtime error

Runtime error

An error message is created if the input channel has not been configured. In this case, please check the connected I/O module EFB.

O_SCALE: Scaled analog output

63

Overview

At a Glance

This chapter describes the block O_SCALE.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

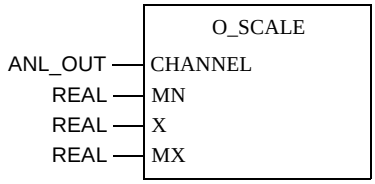
Function description

The Function block converts values from floating point formatREAL into 16 bit integer format. The scaling inputs MN and MX predefine the value range for the analog output. MN corresponds to 0 percent and MX to 100 percent of the output range (e.g. -10 ... 10 V).
EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
CHANNEL	ANL_OUT	Output value
MN	REAL	Scaling input, 0 percent
X	REAL	Floating-point value
MX	REAL	Scaling input, 100 percent

Runtime error

Runtime error

An error message appears,

- if the output channel is not configured. In this case, please check the connected I/O module EFB.
- if the values of MN and MX are identical causing an internal module division by zero.
- with an output value underflow (for example, -1 Volt instead of 0 ... 5 Volt).
- with an output value overflow (for example, 6 Volt instead of 0 ... 5 Volt).

Note: To evaluate the status information for the I/O module, use the O_SCALE_WARN function block.

O_SCALE_WARN: Scaled analog output with warnings status

64

Overview

At a Glance

This chapter describes the block O_SCALE_WARN.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

The Function block converts values from floating point formatREAL into 16 bit integer format. The scaling inputs MN and MX predefine the value range for the analog output. MN corresponds to 0 percent and MX to 100 percent of the output range (e.g. -10 ... 10 V).
In addition the function block at the WARN_NEG and WARN_POS outputs indicate whether a status warning has occurred in the connected analog output EFB.

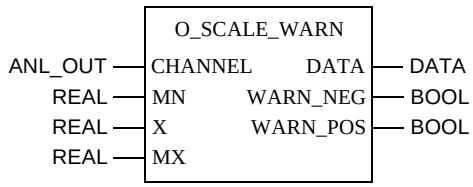
Note: This function block is not compatible with the DAU2xx function for Compact (the O_SCALE function block should be used instead). The O_SCALE_WARN Function block does not recognize the module range information, even though it is assigned to the 3x register area. Therefore, the area warn bits have to be taken directly.

The layout of the status information assigned to State RAM can be found in Online Help for Compact modules (**Help** → **Help on Compact** → **Compact I/O User's Guide**).
EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameters	Data type	Meaning
CHANNEL	ANL_OUT	Output value
MN	REAL	Scaling input, 0 percent
X	REAL	Floating-point value
MX	REAL	Scaling input, 100 percent
WARN_NEG	BOOL	0: no output value underflow at the closed analog output EFB 1: output value underflow at closed analog output EFB (X < MN)
WARN_POS	BOOL	0: no output value overflow at the closed analog output EFB 1: output value exceeded at closed analog output EFB (X > MX)

Runtime error

Runtime error

An error message appears,

- if the output channel is not configured. In this case, please check the connected I/O module EFB.
- if the values of MN and MX are identical causing an internal module division by zero.
- with an output value underflow (outside the warning range, e.g. –1 Volt instead of 0 ... 5 Volt).
- with an output value overflow (outside the warning range, e.g. 6 Volt instead of 0 ... 5 Volt).
- if the connected analog output EFB is unable to generate status information and the warning outputs can, therefore, never become active. In this case, please use the O_SCALE Function block.

O_SET: Set information from analog output channels

65

Overview

At a Glance

This chapter describes the block O_SET.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

The function block sets the information for the analog output channels (ANL_OUT). This block enables all scaling blocks of this library to be used.

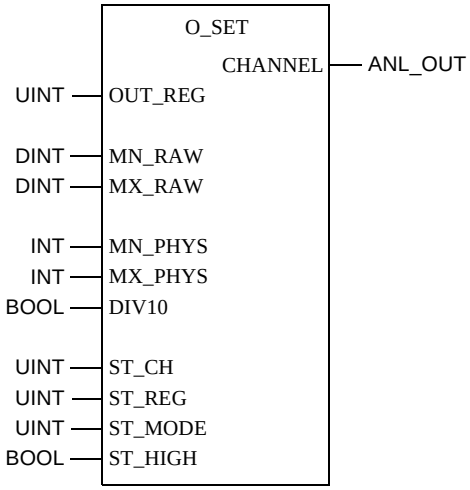
Note: The function block is only required if there is no specific block for a specific analog module available.

EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
OUT_REG	UINT	Number of the raw value register (4X)
MN_RAW	DINT	0 % raw value (e.g. 0)
MX_RAW	DINT	100% raw value (e.g. 4095)
MN_PHYS	INT	lowest output value (e.g. 0 V as 0)
MX_PHYS	INT	greatest output value (e.g. +10 V as 10)
DIV10	BOOL	MN_PHYS and MX_PHYS divided by 10
ST_CH	UINT	channel number (1n) (e.g. 4)
ST_REG	UINT	Number of the status register (3X)
ST_MODE	UINT	Status mode (e.g. 3=ACO_STATUS_MODE)
ST_HIGH	BOOL	Status byte found in highbyte of the register
CHANNEL	ANL_OUT	channel information to be described

Detailed description

Area of application

The function block can be used in three areas:

1. Raw value scaling, with the block: O_NORM and O_SCALE
2. Scaling in physical units, with the O_PHYS block
3. Evaluation of error information, with the blocks O_NORM, O_SCALE and O_PHYS and additional evaluation of status information (warnings) using the O_..._WARN block

Basic circuit connections

The input OUT_REG must always be connected with the number of an output word (4x).

Raw value scaling

For raw value scaling, the inputs MN_RAW (minimum raw value, corresponds to 0%) and MX_RAW (maximum raw value, corresponds to 100%) must also be connected.

Scaling in physical units

For scaling in physical units the inputs MN_PHYS and MX_PHYS must also be connected.
DIV10 is an auxiliary input in the range 0.2 V ... 1 V floating point format to avoid.
Set MN_PHYS=2, MX_PHYS=10 and DIV10=TRUE for this range.
For most ranges this input can remain open (or be assigned FALSE).
e.g. +/-20 mA: here is MN_PHYS=-20, MX_PHYS=20
The input value ranges supported by O_SET can be found in the section *Supported Value Ranges, p. 287*.

Evaluation of error information

For the evaluation of error information the inputs ST_CH, ST_REG and ST_MODE and ST_HIGH must also be configured.
ST_HIGH is an auxiliary input in case the status byte (error information) is located in the registers high byte. For most ranges this input can remain open (or be assigned FALSE).
The input channel number (1 ... n) is given to ST_CH.
If ST_CH is entered, ST_REG and ST_MODE must also be entered.
ST_REG must be connected with the number of an input word (3X), where the status information is located (error and/or warnings).
ST_MODE determines how the status word is evaluated.
The following 8 modes are defined:

Value	Mode	see also module description for
1	AVI_STATUS_MODE	AVI030
2	ACI_STATUS_MODE	ACI030
3	ACO_STATUS_MODE	ACO030
4	ADU_STATUS_MODE	ADU204
5	DAU204_STATUS_MODE	DAU204
6	ADU205_STATUS_MODE	ADU205
7	AMM090_STATUS_MODE	AMM090
8	ADU214_STATUS_MODE	ADU214

Supported Value Ranges

Voltage

Unipolar

Value range	MN_PHYS	MX_PHYS	DIV10
0 ... 0.5 V	0	5	1
0 ... 1.0 V	0	10	1
0 ... 5.0 V	0	5	0
0 ... 10 V	0	10	0
0 ... 20 V	0	20	0
0,1 ... 0.5 V	1	5	1
0,2 ... 1.0 V	2	10	1
1,0 ... 5.0 V	1	5	0
2,0 ... 10, 0 V	2	10	0

Bipolar

Value range	MN_PHYS	MX_PHYS	DIV10
+/- 25 mV	-25	25	0
+/-100 mV	-100	100	0
+/-0.5 V	-5	5	1
+/- 1 V	-1	1	0
+/-5 V	-5	5	0
+/-10 V	-10	10	0
+/-20 V	-20	20	0

Current

Unipolar

Value range	MN_PHYS	MX_PHYS	DIV10
0 .. 20 mA	0	20	0
4 ... 20 mA	4	20	0

Bipolar

Value range	MN_PHYS	MX_PHYS	DIV10
+/-20 mA	-20	20	0
+/-40 mA	-40	40	0

Runtime error

Runtime error

The following error messages can be triggered:

Error message	Meaning
E_EFB_USER_ERROR_1	The input OUT_REG is not connected with the number of an output word (4x).
E_EFB_USER_ERROR_2 with the parameters of the faulty number	The input OUT_REG is connected with an invalid number of an output word (4x).
E_EFB_USER_ERROR_3 with parameter MN_RAW	MN_RAW ≥ MX_RAW)
E_EFB_USER_ERROR_4 with parameter MN_PHYS	Unknown value for MN_PHYS
E_EFB_USER_ERROR_5 with parameter MX_PHYS	Unknown value for MX_PHYS
E_EFB_USER_ERROR_11	ST_REG not entered
E_EFB_USER_ERROR_12	ST_REG too large
E_EFB_USER_ERROR_13	ST_CH not entered

QPR_16I_12O: Configuring the TIO-module QPR 346 00 / 10 / 20 / 21

66

Overview

At a Glance

This chapter describes the block QPR_16I_12O.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

The Function block QPR_16I_12Ois a software connection for the following InterBus-S modules:

- TIO/IS 170 BAM 346 00
- TIO/IS 170 BAM 346 10
- TIO/IS 170 BAM 346 20
- TIO/IS 170 BAM 346 21

The function block has 16 binary inputs and 12 binary outputs, which can be operated simultaneously or just as inputs or outputs. In addition, the module can be programmed in ASCII code via the built-in RS 232 interface.
In an unprogrammed state, the module behaves in the same way as a TIO with 16 binary inputs and 12 binary outputs. In a programmed state, the internal QPR links have priority over signals created at the function block, i.e. the QPR outputs accept the value generated by the internal link. In this case, the output value indicated in Concept need not agree with the actual value in the QPR.

Example:

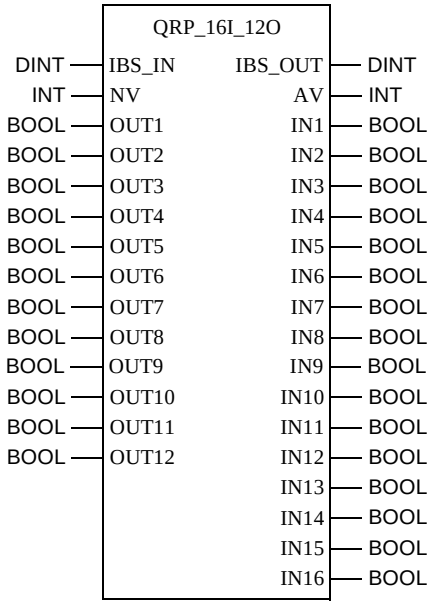
- With Concept, output OUT5 is set to "1".
- With the internal QPR link, output OUT5 is set to "0".
- The value at output 5 of the QPR is "0" in Concept, but a "1" is indicated
- Module programming is described in the user manual for TIO modules with preceding logic operation.

EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
IBS_IN	DINT	Incoming InterBus-S
NV	INT	Nominal value
OUT1	BOOL	Output 1 of the TIO
OUT2	BOOL	Output 2 of the TIO
:	:	:
OUT12	BOOL	Output 12 of the TIO
IBS_OUT	DINT	Outgoing InterBus-S
AV	INT	Actual value
IN1	BOOL	Input 1 of the TIO
IN2	BOOL	Input 2 of the TIO
:	:	:
IN16	BOOL	Input 16 of the TIO

Detailed description

Detailed description

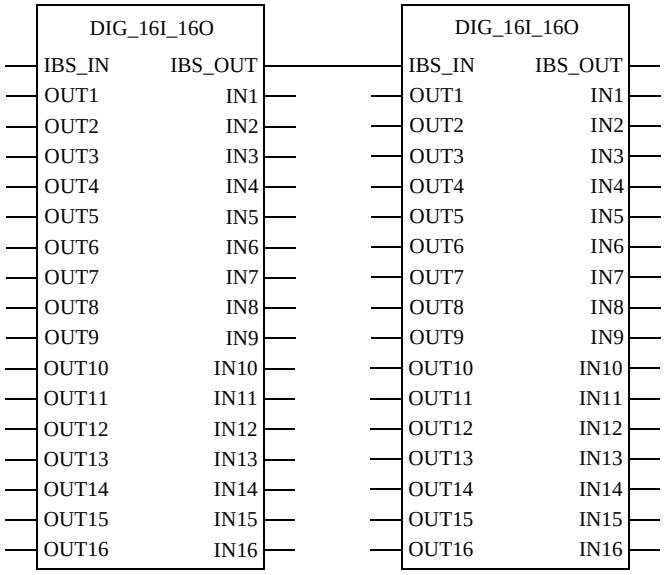
The QPR_16I_12O Function block in Concept functions in the same way as its hardware counterpart.
However, its operation has been simplified by programming it as a function block in Concept. The module occupies two input words and two output words in the master.

Note: The outputs programmed in the QPR cannot be represented by the QPR_16I_12O Function block, as the value is generated by the module itself. The PLC cannot influence the value which was generated by the module.

Parameter description - inputs

IBS_IN

IBS_IN = Connection for the incoming remote bus part of InterBus-S
On the hardware, the male connector is on the top left of the module.
The module is connected here to the outgoing remote bus (IBS_OUT) of the master (1st module on the bus) or the preceding module (see diagram). The link can be made via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two bus devices.
Connection of two InterBus-S modules



NV	NV = Nominal value A number for the programmable counter or a time for a delay switch can be entered here. For the times, 1 = 1ms i.e., 30000 = 30s.
OUTx	OUTx = Output x x stands for a number between 1 and 12 which refers to the corresponding output. The binary values displayed by the InterBus-S module are supplied to the process via the relevant output (OUTx).
Parameter description - Outputs	
IBS_OUT	IBS_OUT = Connection for the outgoing remote bus part of InterBus-S On the hardware, the male connector is on the top right of the module. The module is connected to the incoming remote bus (IBS_IN) of the following module, either via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two InterBus-S modules.
AV	AV = Actual value The actual number on a counter or the actual time of a delay switch is indicated at this output. For the times, 1 = 1ms i.e., 30000 = 30s.
INx	INx = Input x x stands for the digit 1 to 16 which indicates the particular input. Binary process values are read into the InterBus-S module via the relevant input (INx).

QUANTUM: Configuring a main rack

67

Overview

At a Glance

This chapter describes the block QUANTUM.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

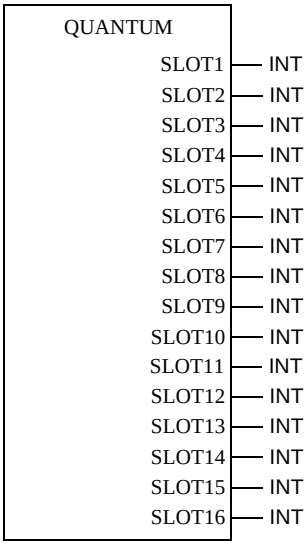
Function description

The Function block is used to edit the configuration data of a Quantum primary backplane for subsequent use by the scaling EFBs.
To configure a QUANTUM primary subrack, the QUANTUM Function block is inserted into the configuration section. The function blocks for the configuration of analog modules or the DROP Function block for the I/O station are connected at its SLOT outputs.
EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
SLOT1	INT	Slot 1
:	:	:
SLOT16	INT	Slot 16

Runtime error

Runtime error	Internal I/O map errors will cause an Error message.

R_INT_WORD: Type conversion (REAL -> INT -> WORD)

68

Overview

At a Glance

This chapter describes the block R_INT_REAL.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

This Function block converts a input value from data type REAL to data type INT and subsequently to data type WORD.

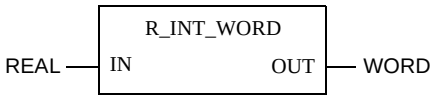
In contrast to the conversion block REAL_TO_WORD (IEC library), the R_INT_WORD block implements a conversion in INT value before the task of the REAL value. This results in the input value of –1.0, for example, being issued as an output value of FFFF (and not like the REAL_TO_WORD block which has an output value of 0).

EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
IN	REAL	Input value
OUT	WORD	Output value

Runtime error

Error message

An error message appears,
- an unauthorized floating point number is placed at the input,
- The value range of the data type INT is violated.

R_UINT_WORD: Type conversion (REAL -> UINT -> WORD)

69

Overview

At a Glance

This chapter describes the block R_UINT_WORD.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

This Function block converts a input value from data type REAL to data type UINT and subsequently to data type WORD.

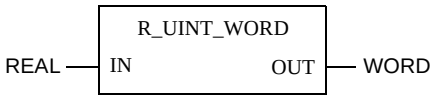
In contrast to the conversion block REAL_TO_WORD (IEC library), the R_UINT_WORD block implements a conversion in UINT value (value range 0.0 - 65535.5) before the output of the WORD value. This results in the input value of –1.0, for example, causes an error message, the output ENO is set and the output value is unchanged (and not like the REAL_TO_WORD block which has no error message and an output value of 0).

EN and ENO can be used as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
IN	REAL	Input value
OUT	WORD	Output value

Runtime error

Error message

An error message appears, if
- an unauthorized floating point number is placed at the input,
- The value range of the data type UINT is violated.

SCALRTOW: Scaling (REAL -> WORD)

70

Overview

At a Glance

This chapter describes the block SCALRTOW.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

The function scales a REAL-input value to a WORD-output value according to a given scale.

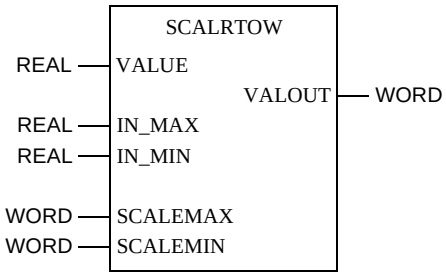
Note: The values for SCALEMAX and SCALEMIN are converted internally before the evaluation according to UINT.

EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Formulas

A linear scaling takes place according to the following formula:

$$VALOUT = (VALUE - IN\_MIN) \times \frac{SCALEMAX - SCALEMIN}{IN\_MAX - IN\_MIN} + SCALEMIN$$

Restrictions:

- If VALUE ≥ IN_MAX, then VALOUT = SCALEMAX.
- If VALUE ≤ IN_MIN, then VALOUT = SCALEMIN.

Parameter description

Block parameter description:

Parameter	Data type	Meaning
VALUE	REAL	Input value
IN_MAX	REAL	Upper limit for input value
IN_MIN	REAL	Lower limit for input value
SCALEMAX	WORD	Upper limit for output value
SCALEMIN	WORD	Lower limit for output value
VALOUT	WORD	Output value

Runtime error

Error message

An error message appears, if

- invalid REAL-values are placed on the inputs. In this case the output value is not changed.
- scaling is invalid, e.g. SCALEMAX < SCALEMIN. In this case the output value is not changed.
- The value of the VALUE-input is not between the given values for IN_MAX and IN_MIN. In this case ENO is set to "0" and the output value is set either to the value of SCALEMAX or SCALEMIN, depending on which value has been violated.

SCALWTOR: Scaling (WORD -> REAL)

71

Overview

At a Glance

This chapter describes the block SCALWTOR.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

The function scales a WORD-input value to a REAL-output value according to a given scale.

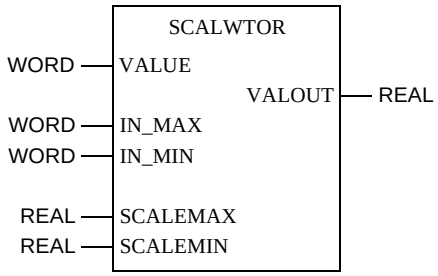
Note: The values for IN_MAX and IN_MIN are converted internally before the evaluation according to UINT

EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Formulas

A linear scaling takes place according to the following formula:

$$VALOUT = (VALUE - IN\_MIN) \times \frac{SCALEMAX - SCALEMIN}{IN\_MAX - IN\_MIN} + SCALEMIN$$

Restrictions:

- If VALUE ≥ IN_MAX, then VALOUT = SCALEMAX.
- If VALUE ≤ IN_MIN, then VALOUT = SCALEMIN.

Parameter description

Block parameter description:

Parameter	Data type	Meaning
VALUE	WORD	Input value
IN_MAX	WORD	Upper limit for input value
IN_MIN	WORD	Lower limit for input value
SCALEMAX	REAL	Upper limit for output value
SCALEMIN	REAL	Lower limit for output value
VALOUT	REAL	Output value

Runtime error

Error message

An error message appears, if

- invalid REAL-values are placed on the inputs. In this case the output value is not changed.
- scaling is invalid, e.g. SCALEMAX < SCALEMIN. In this case the output value is not changed.
- The value of the VALUE-input is not between the given values for IN_MAX and IN_MIN. In this case ENO is set to "0" and the output value is set either to the value of SCALEMAX or SCALEMIN, depending on which value has been violated.

UNI_I: Configuring universal TIO input modules

72

Overview

At a Glance

This chapter describes the block UNI_I.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

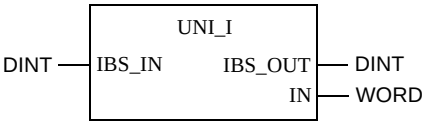
Function description

The UNI_I_O function block is a software connection to a universal InterBus-S hardware module (input only).
The function block has one output for this.
EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
IBS_IN	DINT	Incoming InterBus-S
IBS_OUT	DINT	Outgoing InterBus-S
IN	WORD	Input of an InterBus-S module

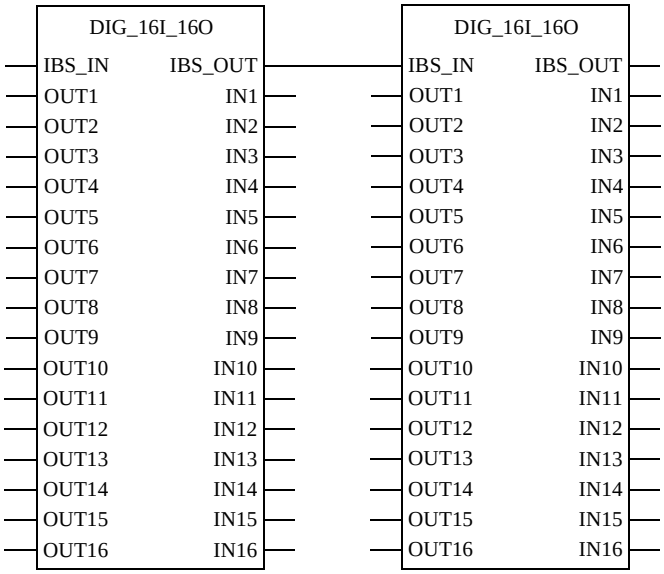
Detailed description

Detailed description

The function block occupies one input word in the Status-RAM.

IBS_IN

IBS_IN = Connection for the incoming remote bus part of InterBus-S
The module is connected here to the outgoing remote bus (IBS_OUT) of the master (1st module on the bus) or the preceding module (see diagram). The link can be made via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two bus devices.
Connection of two InterBus-S modules



IBS_OUT

IBS_OUT = Connection for the outgoing remote bus part of InterBus-S
The module is connected to the incoming remote bus (IBS_IN) of the following module, either via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two InterBus-S modules.

IN

IN = Input
The input reads input information from the InterBus-S module in the form of a word.

UNI_I_O: Configuring universal TIO input/output modules

73

Overview

At a Glance

This chapter describes the block UNI_I_O.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

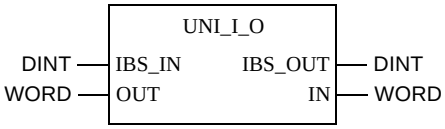
Function description

The UNI_I_O function block is a software connection to a universal InterBus-S hardware module (input/output).
The function block has one input and one output for this.
EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
IBS_IN	DINT	Incoming InterBus-S
OUT	WORD	Output of an InterBus-S module
IBS_OUT	DINT	Outgoing InterBus-S
IN	WORD	Input of an InterBus-S module

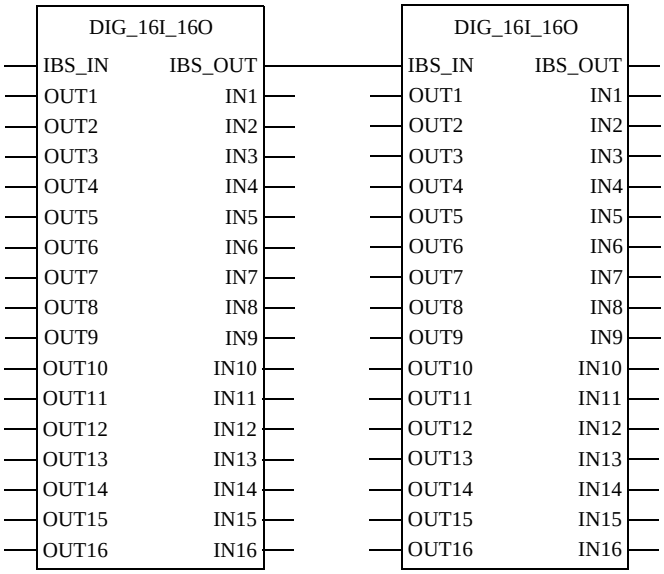
Detailed description

Detailed description

The function block occupies 1 input word and 1 output word in the Status-RAM.

IBS_IN

IBS_IN = Connection for the incoming remote bus part of InterBus-S
The module is connected here to the outgoing remote bus (IBS_OUT) of the master (1st module on the bus) or the preceding module (see diagram). The link can be made via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two bus devices.
Connection of two InterBus-S modules



OUT

OUT = Output
The output provides output information from the InterBus-S module in the form of a word.

IBS_OUT

IBS_OUT = Connection for the outgoing remote bus part of InterBus-S
The module is connected to the incoming remote bus (IBS_IN) of the following module, either via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two InterBus-S modules.

IN = Input

IN = Input
The input reads input information from the InterBus-S module in the form of a word.

UNI_O: Configuring universal TIO output modules

74

Overview

At a Glance

This chapter describes the block UNI_O.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

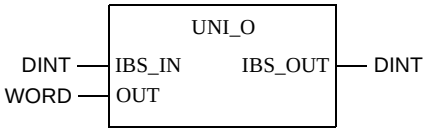
Function description

The UNI_I_O function block is a software connection to a universal InterBus-S hardware module (output only). The function block has one input for this.
EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
IBS_IN	DINT	Incoming InterBus-S
OUT	WORD	Output of an InterBus-S module
IBS_OUT	DINT	Outgoing InterBus-S

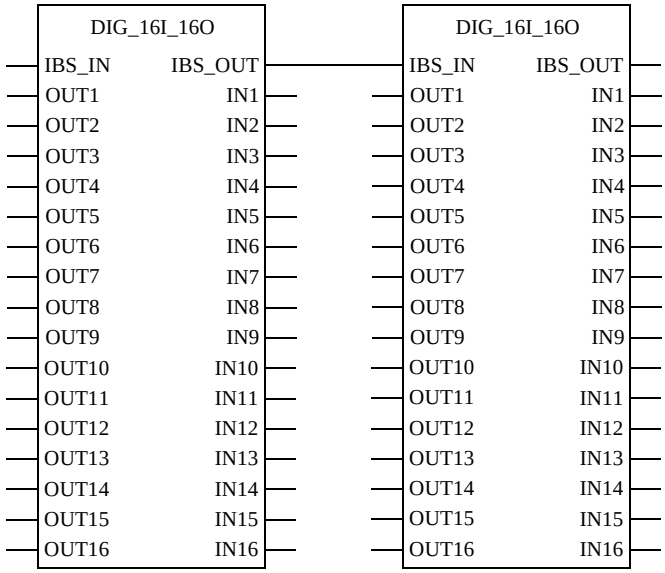
Detailed description

Detailed description

The function block occupies one output word in the Status-RAM.

IBS_IN

IBS_IN = Connection for the incoming remote bus part of InterBus-S
The module is connected here to the outgoing remote bus (IBS_OUT) of the master (1st module on the bus) or the preceding module (see diagram). The link can be made via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two bus devices.
Connection of two InterBus-S modules



OUT = Output
The output provides output information from the InterBus-S module in the form of a word.

IBS_OUT

IBS_OUT = Connection for the outgoing remote bus part of InterBus-S
The module is connected to the incoming remote bus (IBS_IN) of the following module, either via a line or via a variable. For the hardware, the type of connection corresponds to the InterBus-S cable between two InterBus-S modules.

W_INT_REAL: Type conversion (WORD -> INT -> REAL)

75

Overview

At a Glance

This chapter describes the block W_INT_REAL.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

This Function block converts a input value from data type WORD to data type INT and subsequently to data type REAL.

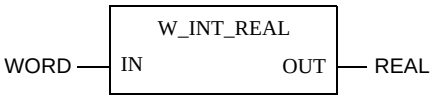
In contrast to the conversion block WORD_TO_REAL (IEC library), the W_INT_REAL block implements a conversion in INT value before the task of the REAL value. This results in the input value FFFF, for example, being issued as an output value of –1.0 (and not like the WORD_TO_REAL block which has an output value of 9.183409e-41).

EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
IN	WORD	Input value
OUT	REAL	Output value

W_UINT_REAL: Type conversion (WORD -> UINT -> REAL)

76

Overview

At a Glance

This chapter describes the block W_UINT_REAL.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

This Function block converts a input value from data type WORD to data type UINT and subsequently to data type REAL.

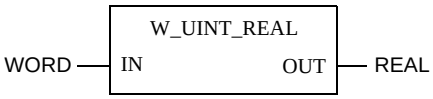
In contrast to the conversion block WORD_TO_REAL (IEC library), the W_INT_REAL block implements a conversion in UINT value before the task of the REAL value. This results in the input value FFFF, for example, being issued as an output value of –1.0 (and not like the WORD_TO_REAL block which has an output value of 9.183409e-41). This results in the input value FFFF, for example, being issued as an output value of 65535.0 (and not like the WORD_TO_REAL block which has an output value of 9.183409e-41).

EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
IN	WORD	Input value
OUT	REAL	Output value

XBP: Configuring a primary backplane expander

77

Overview

At a Glance

This chapter describes the block XBP.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

The Function block is used to edit the configuration data of a Quantum primary backplane expander for subsequent use by the scaling EFBs.

To configure a Quantum primary backplane expander, the XBP Function block is inserted into the configuration section (See *Procedure for expansion of the local backplane using XBE modules (Quantum), p. 23*). It is connected to its SLOT input at the corresponding SLOT x-output of the QUANTUM Function block. The function block for configuring the analog module is connected to the SLOT x-outputs.

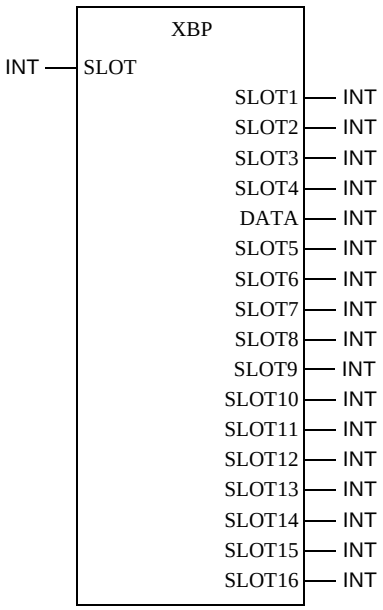
Note: The XBP function block is only used to configure **central** backplane expanders. To configure distributed expansions, use the function block XDROP (See *XDROP: Configuring a I/O Station Backplane , p. 331*).

EN and ENO can be configured as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

Parameter	Data type	Meaning
SLOT	INT	140 XBE 100 00 slot in the central rack
SLOT1	INT	Slot 1
:	:	:
SLOT16	INT	Slot 16

XDROP: Configuring a I/O Station Backplane

78

Overview

At a Glance

This section describes function block XDROP.

What's in this Chapter?

This Chapter contains the following Maps:

Brief description

Function description

The function block is used to prepare the configuration data of distributed I/O station for subsequent processing by module configuration EFBs.
To configure a expansion for an I/O station backplane, the SLOT input of XDROP function block is connected with the SLOT input of the DROP function block in the Configuration Section. The same number must be entered for the NUMBER input of the XDROP function block as for the NUMBER input of the DROP function block.
The Function blocks for configuration of the analog modules of the I/O stations are connected to the X_SLOT outputs.

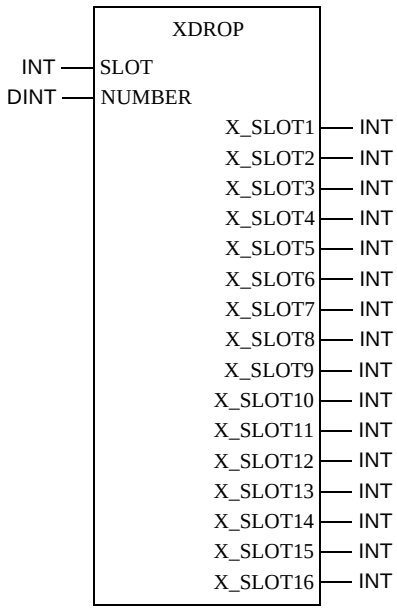
Note: The XDROP function block is only used to configure expansions for distributed backplanes. To configure **central** backplane expanders, use the function block XBP (See *XBP: Configuring a primary backplane expander, p. 327*).

EN and ENO can be projected as additional parameters.

Representation

Symbol

Block representation:



Parameter description

Block parameter description:

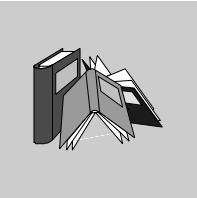
Parameters	Data type	Meaning
SLOT	INT	XBP slot in the distributed backplane
NUMBER	DINT	Number of the distributed station
X_SLOT1	INT	Expansion slot 1
:	:	:
X_SLOT16	INT	Expansion slot 16

Runtime error

Runtime error

If no "Head" has been configured for the I/O station backplane, an error message appears (E_EFB_NOT_CONFIGURED).

Glossary



A

active window The window, which is currently selected. Only one window can be active at any one given time. When a window is active, the heading changes color, in order to distinguish it from other windows. Unselected windows are inactive.

Actual parameter Currently connected Input/Output parameters.

Addresses (Direct) addresses are memory areas on the PLC. These are found in the State RAM and can be assigned input/output modules.
The display/input of direct addresses is possible in the following formats:
- Standard format (400001)
- Separator format (4:00001)
- Compact format (4:1)
- IEC format (QW1)

ANL_IN ANL_IN stands for the data type "Analog Input" and is used for processing analog values. The 3x References of the configured analog input module, which is specified in the I/O component list is automatically assigned the data type and should therefore only be occupied by Unlocated variables.

ANL_OUT ANL_OUT stands for the data type "Analog Output" and is used for processing analog values. The 4x-References of the configured analog output module, which is specified in the I/O component list is automatically assigned the data type and should therefore only be occupied by Unlocated variables.

ANY In the existing version "ANY" covers the elementary data types BOOL, BYTE, DINT, INT, REAL, UDINT, UINT, TIME and WORD and therefore derived data types.

ANY_BIT In the existing version, "ANY_BIT" covers the data types BOOL, BYTE and WORD.

ANY_ELEM In the existing version "ANY_ELEM" covers the elementary data types BOOL, BYTE, DINT, INT, REAL, UDINT, UINT, TIME and WORD.

ANY_INT In the existing version, "ANY_INT" covers the data types DINT, INT, UDINT and UINT.

ANY_NUM In the existing version, "ANY_NUM" covers the data types DINT, INT, REAL, UDINT and UINT.

ANY_REAL In the existing version "ANY_REAL" covers the data type REAL.

Application window The window, which contains the working area, the menu bar and the tool bar for the application. The name of the application appears in the heading. An application window can contain several document windows. In Concept the application window corresponds to a Project.

Argument Synonymous with Actual parameters.

ASCII mode American Standard Code for Information Interchange. The ASCII mode is used for communication with various host devices. ASCII works with 7 data bits.

Atrium The PC based controller is located on a standard AT board, and can be operated within a host computer in an ISA bus slot. The module occupies a motherboard (requires SA85 driver) with two slots for PC104 daughter boards. From this, a PC104 daughter board is used as a CPU and the others for INTERBUS control.

B

Back up data file (Concept EFB) The back up file is a copy of the last Source files. The name of this back up file is "backup??.c" (it is accepted that there are no more than 100 copies of the source files. The first back up file is called "backup00.c". If changes have been made on the Definition file, which do not create any changes to the interface in the EFB, there is no need to create a back up file by editing the source files (**Objects** → **Source**). If a back up file can be assigned, the name of the source file can be given.

Base 16 literals

Base 16 literals function as the input of whole number values in the hexadecimal system. The base must be denoted by the prefix 16#. The values may not be preceded by signs (+/-). Single underline signs (_) between figures are not significant.

Example
16#F_F or 16#FF (decimal 255)
16#E_0 or 16#E0 (decimal 224)

Base 8 literal

Base 8 literals function as the input of whole number values in the octal system. The base must be denoted by the prefix 3.63kg. The values may not be preceded by signs (+/-). Single underline signs (_) between figures are not significant.

Example
8#3_1111 or 8#377 (decimal 255)
8#34_1111 or 8#340 (decimal 224)

Basis 2 literals

Base 2 literals function as the input of whole number values in the dual system. The base must be denoted by the prefix 0.91kg. The values may not be preceded by signs (+/-). Single underline signs (_) between figures are not significant.

Example
2#1111_1111 or 2#11111111 (decimal 255)
2#1110_1111 or 2#11100000 (decimal 224)

Binary connections

Connections between outputs and inputs of FFBs of data type BOOL.

Bit sequence

A data element, which is made up from one or more bits.

BOOL

BOOL stands for the data type "Boolean". The length of the data elements is 1 bit (in the memory contained in 1 byte). The range of values for variables of this type is 0 (FALSE) and 1 (TRUE).

Bridge

A bridge serves to connect networks. It enables communication between nodes on the two networks. Each network has its own token rotation sequence – the token is not deployed via bridges.

BYTE

BYTE stands for the data type "Bit sequence 8". The input appears as Base 2 literal, Base 8 literal or Base 1 16 literal. The length of the data element is 8 bit. A numerical range of values cannot be assigned to this data type.

C

Cache The cache is a temporary memory for cut or copied objects. These objects can be inserted into sections. The old content in the cache is overwritten for each new Cut or Copy.

Call up The operation, by which the execution of an operation is initiated.

Coil A coil is a LD element, which transfers (without alteration) the status of the horizontal link on the left side to the horizontal link on the right side. In this way, the status is saved in the associated Variable/ direct address.

Compact format (4:1) The first figure (the Reference) is separated from the following address with a colon (:), where the leading zero are not entered in the address.

Connection A check or flow of data connection between graphic objects (e.g. steps in the SFC editor, Function blocks in the FBD editor) within a section, is graphically shown as a line.

Constants Constants are Unlocated variables, which are assigned a value that cannot be altered from the program logic (write protected).

Contact A contact is a LD element, which transfers a horizontal connection status onto the right side. This status is from the Boolean AND- operation of the horizontal connection status on the left side with the status of the associated Variables/direct Address. A contact does not alter the value of the associated variables/direct address.

D

Data transfer settings Settings, which determine how information from the programming device is transferred to the PLC.

Data types

The overview shows the hierarchy of data types, as they are used with inputs and outputs of Functions and Function blocks. Generic data types are denoted by the prefix "ANY".

- ANY_ELEM
  - ANY_NUM
    ANY_REAL (REAL)
    ANY_INT (DINT, INT, UDINT, UINT)
  - ANY_BIT (BOOL, BYTE, WORD)
  - TIME
- System data types (IEC extensions)
- Derived (from "ANY" data types)

DCP I/O station

With a Distributed Control Processor (D908) a remote network can be set up with a parent PLC. When using a D908 with remote PLC, the parent PLC views the remote PLC as a remote I/O station. The D908 and the remote PLC communicate via the system bus, which results in high performance, with minimum effect on the cycle time. The data exchange between the D908 and the parent PLC takes place at 1.5 Megabits per second via the remote I/O bus. A parent PLC can support up to 31 (Address 2-32) D908 processors.

DDE (Dynamic Data Exchange)

The DDE interface enables a dynamic data exchange between two programs under Windows. The DDE interface can be used in the extended monitor to call up its own display applications. With this interface, the user (i.e. the DDE client) can not only read data from the extended monitor (DDE server), but also write data onto the PLC via the server. Data can therefore be altered directly in the PLC, while it monitors and analyzes the results. When using this interface, the user is able to make their own "Graphic-Tool", "Face Plate" or "Tuning Tool", and integrate this into the system. The tools can be written in any DDE supporting language, e.g. Visual Basic and Visual-C++. The tools are called up, when the one of the buttons in the dialog box extended monitor uses Concept Graphic Tool: Signals of a projection can be displayed as timing diagrams via the DDE connection between Concept and Concept Graphic Tool.

Decentral Network (DIO)

A remote programming in Modbus Plus network enables maximum data transfer performance and no specific requests on the links. The programming of a remote net is easy. To set up the net, no additional ladder diagram logic is needed. Via corresponding entries into the Peer Cop processor all data transfer requests are met.

Declaration

Mechanism for determining the definition of a Language element. A declaration normally covers the connection of an Identifier with a language element and the assignment of attributes such as Data types and algorithms.

Definition data file (Concept EFB)	The definition file contains general descriptive information about the selected FFB and its formal parameters.
Derived data type	Derived data types are types of data, which are derived from the Elementary data types and/or other derived data types. The definition of the derived data types appears in the data type editor in Concept. Distinctions are made between global data types and local data types.
Derived Function Block (DFB)	A derived function block represents the Call up of a derived function block type. Details of the graphic form of call up can be found in the definition " Function block (Item)". Contrary to calling up EFB types, calling up DFB types is denoted by double vertical lines on the left and right side of the rectangular block symbol. The body of a derived function block type is designed using FBD language, but only in the current version of the programming system. Other IEC languages cannot yet be used for defining DFB types, nor can derived functions be defined in the current version. Distinctions are made between local and global DFBs.
DINT	DINT stands for the data type "double integer". The input appears as Integer literal, Base 2 literal, Base 8 literal or Base 16 literal. The length of the data element is 32 bit. The range of values for variables of this data type is from –2 exp (31) to 2 exp (31) –1.
Direct display	A method of displaying variables in the PLC program, from which the assignment of configured memory can be directly and indirectly derived from the physical memory.
Document window	A window within an Application window. Several document windows can be opened at the same time in an application window. However, only one document window can be active. Document windows in Concept are, for example, sections, the message window, the reference data editor and the PLC configuration.
Dummy	An empty data file, which consists of a text header with general file information, i.e. author, date of creation, EFB identifier etc. The user must complete this dummy file with additional entries.
DX Zoom	This property enables connection to a programming object to observe and, if necessary, change its data value.

E

Elementary functions/ function blocks (EFB)
Identifier for Functions or Function blocks, whose type definitions are not formulated in one of the IEC languages, i.e. whose bodies, for example, cannot be modified with the DFB Editor (Concept-DFB). EFB types are programmed in "C" and mounted via Libraries in precompiled form.

EN / ENO (Enable / Error display)
If the value of EN is "0" when the FFB is called up, the algorithms defined by the FFB are not executed and all outputs contain the previous value. The value of ENO is automatically set to "0" in this case. If the value of EN is "1" when the FFB is called up, the algorithms defined by the FFB are executed. After the error free execution of the algorithms, the ENO value is automatically set to "1". If an error occurs during the execution of the algorithm, ENO is automatically set to "0". The output behavior of the FFB depends whether the FFBs are called up without EN/ENO or with EN=1. If the EN/ENO display is enabled, the EN input must be active. Otherwise, the FFB is not executed. The projection of EN and ENO is enabled/disabled in the block properties dialog box. The dialog box is called up via the menu commands **Objects** → **Properties...** or via a double click on the FFB.

Error
When processing a FFB or a Step an error is detected (e.g. unauthorized input value or a time error), an error message appears, which can be viewed with the menu command **Online** → **Event display...** . With FFBs the ENO output is set to "0".

Evaluation
The process, by which a value for a Function or for the outputs of a Function block during the Program execution is transmitted.

Expression
Expressions consist of operators and operands.

F

FFB (functions/ function blocks)
Collective term for EFB (elementary functions/function blocks) and DFB (derived function blocks)

Field variables
Variables, one of which is assigned, with the assistance of the key word ARRAY (field), a defined Derived data type. A field is a collection of data elements of the same Data type.

FIR filter
Finite Impulse Response Filter

Formal parameters
Input/Output parameters, which are used within the logic of a FFB and led out of the FFB as inputs/outputs.

Function (FUNC)
A Program organization unit, which exactly supplies a data element when executing. A function has no internal status information. Multiple call ups of the same function with the same input parameter values always supply the same output values. Details of the graphic form of function call up can be found in the definition " Function block (Item)". In contrast to the call up of function blocks, the function call ups only have one unnamed output, whose name is the name of the function itself. In FBD each call up is denoted by a unique number over the graphic block; this number is automatically generated and cannot be altered.

Function block (item) (FB)
A function block is a Program organization unit, which correspondingly calculates the functionality values, defined in the function block type description, for the output and internal variables, when it is called up as a certain item. All output values and internal variables of a certain function block item remain as a call up of the function block until the next. Multiple call up of the same function block item with the same arguments (Input parameter values) supply generally supply the same output value(s).
Each function block item is displayed graphically by a rectangular block symbol. The name of the function block type is located on the top center within the rectangle. The name of the function block item is located also at the top, but on the outside of the rectangle. An instance is automatically generated when creating, which can however be altered manually, if required. Inputs are displayed on the left side and outputs on the right of the block. The names of the formal input/output parameters are displayed within the rectangle in the corresponding places.
The above description of the graphic presentation is principally applicable to Function call ups and to DFB call ups. Differences are described in the corresponding definitions.

Function block dialog (FBD)
One or more sections, which contain graphically displayed networks from Functions, Function blocks and Connections.

Function block type
A language element, consisting of: 1. the definition of a data structure, subdivided into input, output and internal variables, 2. A set of operations, which is used with the elements of the data structure, when a function block type instance is called up. This set of operations can be formulated either in one of the IEC languages (DFB type) or in "C" (EFB type). A function block type can be instanced (called up) several times.

Function counter
The function counter serves as a unique identifier for the function in a Program or DFB. The function counter cannot be edited and is automatically assigned. The function counter always has the structure: .n.m
n = Section number (number running)
m = Number of the FFB object in the section (number running)

G

Generic data type — A Data type, which stands in for several other data types.

Generic literal — If the Data type of a literal is not relevant, simply enter the value for the literal. In this case Concept automatically assigns the literal to a suitable data type.

Global derived data types — Global Derived data types are available in every Concept project and are contained in the DFB directory directly under the Concept directory.

Global DFBs — Global DFBs are available in every Concept project and are contained in the DFB directory directly under the Concept directory.

Global macros — Global Macros are available in every Concept project and are contained in the DFB directory directly under the Concept directory.

Groups (EFBs) — Some EFB libraries (e.g. the IEC library) are subdivided into groups. This facilitates the search for FFBs, especially in extensive libraries.

I

I/O component list — The I/O and expert assemblies of the various CPUs are configured in the I/O component list.

IEC 61131-3 — International norm: Programmable controllers – part 3: Programming languages.

IEC format (QW1) — In the place of the address stands an IEC identifier, followed by a five figure address:
- %0x12345 = %Q12345
- %1x12345 = %I12345
- %3x12345 = %IW12345
- %4x12345 = %QW12345

IEC name conventions (identifier)

An identifier is a sequence of letters, figures, and underscores, which must start with a letter or underscores (e.g. name of a function block type, of an item or section). Letters from national sets of characters (e.g. ö,ü, é, õ) can be used, taken from project and DFB names.
Underscores are significant in identifiers; e.g. "A_BCD" and "AB_CD" are interpreted as different identifiers. Several leading and multiple underscores are not authorized consecutively.
Identifiers are not permitted to contain space characters. Upper and/or lower case is not significant; e.g. "ABCD" and "abcd" are interpreted as the same identifier.
Identifiers are not permitted to be Key words.

IIR filter

Infinite Impulse Response Filter

Initial step (starting step)

The first step in a chain. In each chain, an initial step must be defined. The chain is started with the initial step when first called up.

Initial value

The allocated value of one of the variables when starting the program. The value assignment appears in the form of a Literal.

Input bits (1x references)

The 1/0 status of input bits is controlled via the process data, which reaches the CPU from an entry device.

Note: The x, which comes after the first figure of the reference type, represents a five figure storage location in the application data store, i.e. if the reference 100201 signifies an input bit in the address 201 of the State RAM.

Input parameters (Input)

When calling up a FFB the associated Argument is transferred.

Input words (3x references)

An input word contains information, which come from an external source and are represented by a 16 bit figure. A 3x register can also contain 16 sequential input bits, which were read into the register in binary or BCD (binary coded decimal) format.
Note: The x, which comes after the first figure of the reference type, represents a five figure storage location in the user data store, i.e. if the reference 300201 signifies a 16 bit input word in the address 201 of the State RAM.

Instantiation

The generation of an Item.

Instruction (IL)

Instructions are "commands" of the IL programming language. Each operation begins on a new line and is succeeded by an operator (with modifier if needed) and, if necessary for each relevant operation, by one or more operands. If several operands are used, they are separated by commas. A tag can stand before the instruction, which is followed by a colon. The commentary must, if available, be the last element in the line.

Instruction (LL984)

When programming electric controllers, the task of implementing operational coded instructions in the form of picture objects, which are divided into recognizable contact forms, must be executed. The designed program objects are, on the user level, converted to computer useable OP codes during the loading process. The OP codes are deciphered in the CPU and processed by the controller's firmware functions so that the desired controller is implemented.

Instruction list (IL)

IL is a text language according to IEC 1131, in which operations, e.g. conditional/ unconditional call up of Function blocks and Functions, conditional/unconditional jumps etc. are displayed through instructions.

INT

INT stands for the data type "whole number". The input appears as Integer literal, Base 2 literal, Base 8 literal or Base 16 literal. The length of the data element is 16 bit. The range of values for variables of this data type is from –2 exp (15) to 2 exp (15) –1.

Integer literals

Integer literals function as the input of whole number values in the decimal system. The values may be preceded by the signs (+/-). Single underline signs (_) between figures are not significant.

Example
-12, 0, 123_456, +986

INTERBUS (PCP)

To use the INTERBUS PCP channel and the INTERBUS process data preprocessing (PDP), the new I/O station type INTERBUS (PCP) is led into the Concept configurator. This I/O station type is assigned fixed to the INTERBUS connection module 180-CRP-660-01.
The 180-CRP-660-01 differs from the 180-CRP-660-00 only by a clearly larger I/O area in the state RAM of the controller.

Item name

An Identifier, which belongs to a certain Function block item. The item name serves as a unique identifier for the function block in a program organization unit. The item name is automatically generated, but can be edited. The item name must be unique throughout the Program organization unit, and no distinction is made between upper/lower case. If the given name already exists, a warning is given and another name must be selected. The item name must conform to the IEC name conventions, otherwise an error message appears. The automatically generated instance name always has the structure: FBI_n_m

FBI = Function block item
n = Section number (number running)
m = Number of the FFB object in the section (number running)

J

Jump

Element of the SFC language. Jumps are used to jump over areas of the chain.

K

Key words

Key words are unique combinations of figures, which are used as special syntactic elements, as is defined in appendix B of the IEC 1131-3. All key words, which are used in the IEC 1131-3 and in Concept, are listed in appendix C of the IEC 1131-3. These listed keywords cannot be used for any other purpose, i.e. not as variable names, section names, item names etc.

L

Ladder Diagram (LD)

Ladder Diagram is a graphic programming language according to IEC1131, which optically orientates itself to the "rung" of a relay ladder diagram.

Ladder Logic 984 (LL)

In the terms Ladder Logic and Ladder Diagram, the word Ladder refers to execution. In contrast to a diagram, a ladder logic is used by engineers to draw up a circuit (with assistance from electrical symbols),which should chart the cycle of events and not the existing wires, which connect the parts together. A usual user interface for controlling the action by automated devices permits ladder logic interfaces, so that when implementing a control system, engineers do not have to learn any new programming languages, with which they are not conversant.
The structure of the actual ladder logic enables electrical elements to be linked in a way that generates a control output, which is dependant upon a configured flow of power through the electrical objects used, which displays the previously demanded condition of a physical electric appliance.
In simple form, the user interface is one of the video displays used by the PLC programming application, which establishes a vertical and horizontal grid, in which the programming objects are arranged. The logic is powered from the left side of the grid, and by connecting activated objects the electricity flows from left to right.

Landscape format

Landscape format means that the page is wider than it is long when looking at the printed text.

Language element

Each basic element in one of the IEC programming languages, e.g. a Step in SFC, a Function block item in FBD or the Start value of a variable.

Library

Collection of software objects, which are provided for reuse when programming new projects, or even when building new libraries. Examples are the Elementary function block types libraries.
EFB libraries can be subdivided into Groups.

Literals

Literals serve to directly supply values to inputs of FFBs, transition conditions etc. These values cannot be overwritten by the program logic (write protected). In this way, generic and standardized literals are differentiated.
Furthermore literals serve to assign a Constant a value or a Variable an Initial value. The input appears as Base 2 literal, Base 8 literal, Base 16 literal, Integer literal, Real literal or Real literal with exponent.

Local derived data types

Local derived data types are only available in a single Concept project and its local DFBs and are contained in the DFB directory under the project directory.

Local DFBs

Local DFBs are only available in a single Concept project and are contained in the DFB directory under the project directory.

Local link

The local network link is the network, which links the local nodes with other nodes either directly or via a bus amplifier.

Local macros

Local Macros are only available in a single Concept project and are contained in the DFB directory under the project directory.

Local network nodes

The local node is the one, which is projected evenly.

Located variable

Located variables are assigned a state RAM address (reference addresses 0x,1x, 3x, 4x). The value of these variables is saved in the state RAM and can be altered online with the reference data editor. These variables can be addressed by symbolic names or the reference addresses.

Collective PLC inputs and outputs are connected to the state RAM. The program access to the peripheral signals, which are connected to the PLC, appears only via located variables. PLC access from external sides via Modbus or Modbus plus interfaces, i.e. from visualizing systems, are likewise possible via located variables.

M

Macro

Macros are created with help from the software Concept DFB.
Macros function to duplicate frequently used sections and networks (including the logic, variables, and variable declaration).
Distinctions are made between local and global macros.

Macros have the following properties:

- Macros can only be created in the programming languages FBD and LD.
- Macros only contain one single section.
- Macros can contain any complex section.
- From a program technical point of view, there is no differentiation between an instanced macro, i.e. a macro inserted into a section, and a conventionally created macro.
- Calling up DFBs in a macro
- Variable declaration
- Use of macro-own data structures
- Automatic acceptance of the variables declared in the macro
- Initial value for variables
- Multiple instancing of a macro in the whole program with different variables
- The section name, the variable name and the data structure name can contain up to 10 different exchange markings (@0 to @9).

MMI

Man Machine Interface

Multi element variables

Variables, one of which is assigned a Derived data type defined with STRUCT or ARRAY.
Distinctions are made between Field variables and structured variables.

N

Network A network is the connection of devices to a common data path, which communicate with each other via a common protocol.

Network node A node is a device with an address (164) on the Modbus Plus network.

Node address The node address serves a unique identifier for the network in the routing path. The address is set directly on the node, e.g. with a rotary switch on the back of the module.

O

Operand An operand is a Literal, a Variable, a Function call up or an Expression.

Operator An operator is a symbol for an arithmetic or Boolean operation to be executed.

Output parameters (Output) A parameter, with which the result(s) of the Evaluation of a FFB are returned.

Output/discretes (0x references) An output/marker bit can be used to control real output data via an output unit of the control system, or to define one or more outputs in the state RAM. Note: The x, which comes after the first figure of the reference type, represents a five figure storage location in the application data store, i.e. if the reference 000201 signifies an output or marker bit in the address 201 of the State RAM.

Output/marker words (4x references) An output/marker word can be used to save numerical data (binary or decimal) in the State RAM, or also to send data from the CPU to an output unit in the control system. Note: The x, which comes after the first figure of the reference type, represents a five figure storage location in the application data store, i.e. if the reference 400201 signifies a 16 bit output or marker word in the address 201 of the State RAM.

P

Peer processor The peer processor processes the token run and the flow of data between the Modbus Plus network and the PLC application logic.

PLC Programmable controller

Program The uppermost Program organization unit. A program is closed and loaded onto a single PLC.

Program cycle A program cycle consists of reading in the inputs, processing the program logic and the output of the outputs.

Program organization unit A Function, a Function block, or a Program. This term can refer to either a Type or an Item.

Programming device Hardware and software, which supports programming, configuring, testing, implementing and error searching in PLC applications as well as in remote system applications, to enable source documentation and archiving. The programming device could also be used for process visualization.

Programming redundancy system (Hot Standby) A redundancy system consists of two identically configured PLC devices, which communicate with each other via redundancy processors. In the case of the primary PLC failing, the secondary PLC takes over the control checks. Under normal conditions the secondary PLC does not take over any controlling functions, but instead checks the status information, to detect mistakes.

Project General identification of the uppermost level of a software tree structure, which specifies the parent project name of a PLC application. After specifying the project name, the system configuration and control program can be saved under this name. All data, which results during the creation of the configuration and the program, belongs to this parent project for this special automation.
General identification for the complete set of programming and configuring information in the Project data bank, which displays the source code that describes the automation of a system.

Project data bank The data bank in the Programming device, which contains the projection information for a Project.

Prototype data file (Concept EFB)

The prototype data file contains all prototypes of the assigned functions. Further, if available, a type definition of the internal

R

REAL

REAL stands for the data type "real". The input appears as Real literal or as Real literal with exponent. The length of the data element is 32 bit. The value range for variables of this data type reaches from 8.43E-37 to 3.36E+38.

Note: Depending on the mathematic processor type of the CPU, various areas within this valid value range cannot be represented. This is valid for values nearing ZERO and for values nearing INFINITY. In these cases, a number value is not shown in animation, instead NAN (**N**ot **A N**umber) oder INF (**INF**inite).

Real literal

Real literals function as the input of real values in the decimal system. Real literals are denoted by the input of the decimal point. The values may be preceded by the signs (+/-). Single underline signs (_) between figures are not significant.

Example
-12.0, 0.0, +0.456, 3.14159_26

Reference

Each direct address is a reference, which starts with an ID, specifying whether it concerns an input or an output and whether it concerns a bit or a word. References, which start with the code 6, display the register in the extended memory of the state RAM.
0x area = Discrete outputs
1x area = Input bits
3x area = Input words
4x area = Output bits/Marker words
6x area = Register in the extended memory

Note: The x, which comes after the first figure of each reference type, represents a five figure storage location in the application data store, i.e. if the reference 400201 signifies a 16 bit output or marker word in the address 201 of the State RAM.

Real literal with exponent

Real literals with exponent function as the input of real values in the decimal system. Real literals with exponent are denoted by the input of the decimal point. The exponent sets the key potency, by which the preceding number is multiplied to get to the value to be displayed. The basis may be preceded by a negative sign (-). The exponent may be preceded by a positive or negative sign (+/-). Single underline signs (_) between figures are not significant. (Only between numbers, not before or after the decimal poiont and not before or after "E", "E+" or "E-")

Example
-1.34E-12 or -1.34e-12
1.0E+6 or 1.0e+6
1.234E6 or 1.234e6

Register in the extended memory (6x reference)

6x references are marker words in the extended memory of the PLC. Only LL984 user programs and CPU 213 04 or CPU 424 02 can be used.

RIO (Remote I/O)

Remote I/O provides a physical location of the I/O coordinate setting device in relation to the processor to be controlled. Remote inputs/outputs are connected to the consumer control via a wired communication cable.

RP (PROFIBUS)

RP = Remote Peripheral

RTU mode

Remote Terminal Unit
The RTU mode is used for communication between the PLC and an IBM compatible personal computer. RTU works with 8 data bits.

Rum-time error

Error, which occurs during program processing on the PLC, with SFC objects (i.e. steps) or FFBs. These are, for example, over-runs of value ranges with figures, or time errors with steps.

S

SA85 module

The SA85 module is a Modbus Plus adapter for an IBM-AT or compatible computer.

Section

A section can be used, for example, to describe the functioning method of a technological unit, such as a motor.
A Program or DFB consist of one or more sections. Sections can be programmed with the IEC programming languages FBD and SFC. Only one of the named programming languages can be used within a section.
Each section has its own Document window in Concept. For reasons of clarity, it is recommended to subdivide a very large section into several small ones. The scroll bar serves to assist scrolling in a section.

Separator format (4:00001)

The first figure (the Reference) is separated from the ensuing five figure address by a colon (:).

Sequence language (SFC)

The SFC Language elements enable the subdivision of a PLC program organizational unit in a number of Steps and Transitions, which are connected horizontally by aligned Connections. A number of actions belong to each step, and a transition condition is linked to a transition.

Serial ports

With serial ports (COM) the information is transferred bit by bit.

Source code data file (Concept EFB)

The source code data file is a usual C++ source file. After execution of the menu command **Library** → **Generate data files** this file contains an EFB code framework, in which a specific code must be entered for the selected EFB. To do this, click on the menu command **Objects** → **Source**.

Standard format (400001)

The five figure address is located directly after the first figure (the reference).

Standardized literals

If the data type for the literal is to be automatically determined, use the following construction: 'Data type name'#'Literal value'.

Example
INT#15 (Data type: Integer, value: 15),
BYTE#00001111 (data type: Byte, value: 00001111)
REAL#23.0 (Data type: Real, value: 23.0)

For the assignment of REAL data types, there is also the possibility to enter the value in the following way: 23.0.
Entering a comma will automatically assign the data type REAL.

State RAM

The state RAM is the storage for all sizes, which are addressed in the user program via References (Direct display). For example, input bits, discretes, input words, and discrete words are located in the state RAM.

Statement (ST) Instructions are "commands" of the ST programming language. Instructions must be terminated with semicolons. Several instructions (separated by semi-colons) can occupy the same line.

Status bits There is a status bit for every node with a global input or specific input/output of Peer Cop data. If a defined group of data was successfully transferred within the set time out, the corresponding status bit is set to 1. Alternatively, this bit is set to 0 and all data belonging to this group (of 0) is deleted.

Step SFC Language element: Situations, in which the Program behavior follows in relation to the inputs and outputs of the same operations, which are defined by the associated actions of the step.

Step name The step name functions as the unique flag of a step in a Program organization unit. The step name is automatically generated, but can be edited. The step name must be unique throughout the whole program organization unit, otherwise an Error message appears.
The automatically generated step name always has the structure: S_n_m

S = Step
n = Section number (number running)
m = Number of steps in the section (number running)

Structured text (ST) ST is a text language according to IEC 1131, in which operations, e.g. call up of Function blocks and Functions, conditional execution of instructions, repetition of instructions etc. are displayed through instructions.

Structured variables Variables, one of which is assigned a Derived data type defined with STRUCT (structure).
A structure is a collection of data elements with generally differing data types (Elementary data types and/or derived data types).

SY/MAX In Quantum control devices, Concept closes the mounting on the I/O population SY/MAX I/O modules for RIO control via the Quantum PLC with on. The SY/MAX remote subrack has a remote I/O adapter in slot 1, which communicates via a Modicon S908 R I/O system. The SY/MAX I/O modules are performed when highlighting and including in the I/O population of the Concept configuration.

Symbol (Icon) Graphic display of various objects in Windows, e.g. drives, user programs and Document windows.

T

Template data file (Concept EFB)

The template data file is an ASCII data file with a layout information for the Concept FBD editor, and the parameters for code generation.

TIME

TIME stands for the data type "Time span". The input appears as Time span literal. The length of the data element is 32 bit. The value range for variables of this type stretches from 0 to 2exp(32)-1. The unit for the data type TIME is 1 ms.

Time span literals

Permitted units for time spans (TIME) are days (D), hours (H), minutes (M), seconds (S) and milliseconds (MS) or a combination thereof. The time span must be denoted by the prefix t#, T#, time# or TIME#. An "overrun" of the highest ranking unit is permitted, i.e. the input T#25H15M is permitted.

Example
t#14MS, T#14.7S, time#18M, TIME#19.9H, t#20.4D, T#25H15M, time#5D14H12M18S3.5MS

Token

The network "Token" controls the temporary property of the transfer rights via a single node. The token runs through the node in a circulating (rising) address sequence. All nodes track the Token run through and can contain all possible data sent with it.

Traffic Cop

The Traffic Cop is a component list, which is compiled from the user component list. The Traffic Cop is managed in the PLC and in addition contains the user component list e.g. Status information of the I/O stations and modules.

Transition

The condition with which the control of one or more Previous steps transfers to one or more ensuing steps along a directional Link.

U

UDEFB
User defined elementary functions/function blocks
Functions or Function blocks, which were created in the programming language C, and are available in Concept Libraries.

UDINT
UDINT stands for the data type "unsigned double integer". The input appears as Integer literal, Base 2 literal, Base 8 literal or Base 16 literal. The length of the data element is 32 bit. The value range for variables of this type stretches from 0 to 2exp(32)-1.

UINT
UINT stands for the data type "unsigned integer". The input appears as Integer literal, Base 2 literal, Base 8 literal or Base 16 literal. The length of the data element is 16 bit. The value range for variables of this type stretches from 0 to (2exp16)-1.

Unlocated variable
Unlocated variables are not assigned any state RAM addresses. They therefore do not occupy any state RAM addresses. The value of these variables is saved in the system and can be altered with the reference data editor. These variables are only addressed by symbolic names.

Signals requiring no peripheral access, e.g. intermediate results, system tags etc, should primarily be declared as unlocated variables.

V

Variables
Variables function as a data exchange within sections between several sections and between the Program and the PLC.
Variables consist of at least a variable name and a Data type.
Should a variable be assigned a direct Address (Reference), it is referred to as a Located variable. Should a variable not be assigned a direct address, it is referred to as an unlocated variable. If the variable is assigned a Derived data type, it is referred to as a Multi-element variable.
Otherwise there are Constants and Literals.

Vertical format
Vertical format means that the page is higher than it is wide when looking at the printed text.

W

Warning When processing a FFB or a Step a critical status is detected (e.g. critical input value or a time out), a warning appears, which can be viewed with the menu command **Online** → **Event display...** . With FFBs the ENO output remains at "1".

WORD WORD stands for the data type "Bit sequence 16". The input appears as Base 2 literal, Base 8 literal or Base 1 16 literal. The length of the data element is 16 bit. A numerical range of values cannot be assigned to this data type.

Index

A

B

C

D

F

I

L

M

N

O

P

Q

R

S

T

U

W

X