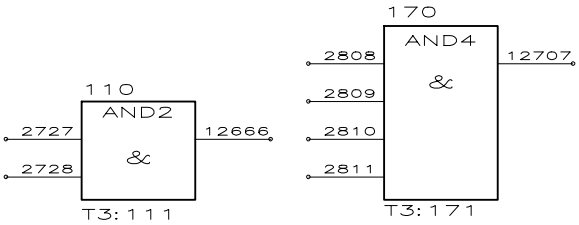
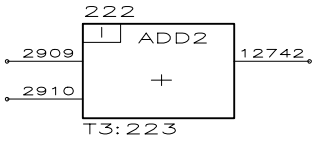
DCS 500

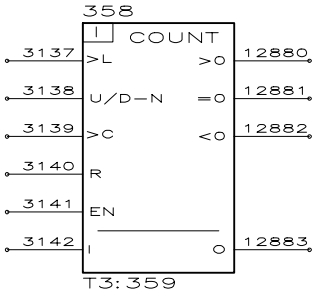
Application blocks

DCS 500 Standard Drives

DC Drives 25 to 5150 A







DCS 500 Standard Drives
DC Drives 25 to 5150 A

Application Blocks

Code: 3ADW 000 048 R0301 Rev C

IND/AME: ap_blke.doc

EFFECTIVE: 2.Dec.96
SUPERSEDES: Rev B

Contents

Contents Page

1. Introduction

1.1 DCS 500 programming

This manual is describing programming of the DCS 500 standard DC-drive.

Programming is done by setting the DCS 500 pointers. They are used to define connections between function blocks and to set operating values for standard and application function blocks.

DCS 500 drive application can be created in three different ways:
- with a PC-based engineering tool, the GAD
- with the CDP 310 control panel
- with a PC-based commissioning tool, the CMT/DCS 500.

CDP 310 panel and CMT/DCS 500 should be used only for small size applications. GAD is the right tool for larger applications.

Drive application created with the GAD is downloaded to the DCS 500 drive with the CMT/DCS 500 commissioning tool.

The GAD can also generate diagram files for the CMT/DCS 500. These files contain all graphical information about the application, so that the CMT/DCS 500 can display the function block application in the same format as it was displayed in the GAD.

CDP 310 panel and CMT/DCS 500 commissioning tool can be used to make on-line changes to the drive application.

1.2 DCS 500 Function Blocks

General

Function blocks are software implemented elements which perform some basic function. With them the DCS 500 user can create small and medium size application programs to meet the needs of his drive application.

The Programming Language

The Function Block (FB) is the smallest unit of the application programming language. Each FB performs a complete function, such as a logical AND gate, a timer or a feedback control block.

Call

Each block has an unique call name that identifies its type and instance number (e.g. ADD2_222). The block is selected with the individual block instance number.

Graphic Symbol

Each Function Block has a graphic symbol. Normally, the call name of the block is written in the uppermost part of the symbol. For certain functions, e.g. AND and OR, the name is replaced by a special symbol.

The instance number of the block is written above the symbol.

Connections of Function Block Inputs and Outputs

Only inputs and outputs with the same data type (e.g. integer or Boolean) should be interconnected.
The GAD-tool, CMT/DCS 500 or the operators panel make no checks on the data type of connected pins. The responsibility of correctly made connections is left to the user.

1.3 Function Block Libraries for the DCS 500

The function block libraries of the DCS 500 contain a fixed set of function blocks to be used in different drive applications.

In the GAD engineering tool there are four sub-libraries for the DCS 500:

1. Standard Program FBs
2. Application Program FBs
3. Application Information FBs
4. Template FBs

Standard Program FBs

These Function Blocks describe the different functions of the DCS 500 Standard Program. There are blocks like Analog Input1, Ramp and Speed Control.
These functions are described within the DCS 500 documentation. This document describes in detail only the Application Program blocks.
The user cannot affect the execution on Standard Program FBs. It is handled totally inside the DCS 500 Drive Program. The user is only able to set the block parameters and connect the inputs.
Simple drive applications can be done by using only Standard Program FBs.

Application Program FBs

If the user has a drive application which cannot be created only with Standard Program FBs, he can use the Application Program FBs.

Application Program FBs are general purpose blocks which can be connected to each other and to Standard Program FBs to create a complete drive application.

When the application is created, the user has to specify which blocks are used, the connections between blocks, the execution order and the execution interval.

Application Information FBs

These FBs are used to give some information about the application in hand and about the DCS 500.

Template FBs

These FBs are used to create the template which is used by the GAD engineering tool, when the application is designed.
The user is able to customise these FBs to better suit his needs and his documentation standard.

1.4 Programming with the FB Language

When programming a DCS 500 drive application with the GAD-tool, FBs are selected from the DCS 500 FB libraries and copied to the application design template. The Function Blocks are then interconnected to define the data flow between the FBs and correct values are entered for the FB parameters.

The compiler of the GAD-tool translates the program into a load file, which is loaded into the DCS 500 with the CMT/DCS 500-tool.

Also the CDP 310 operators panel or the CMT/DCS 500 can be used to create an application. This is convenient if the application is quite small.

FB Application tasks

The number of function block instances is fixed. The instance numbers for each block type are presented in the detailed description of the block.

There are three cyclic tasks with different intervals available:

Task1:	5 ms
Task2:	20 ms
Task3:	200 ms

A scheduler program will call all blocks in the execution list of the task one after another. The execution order is the same as the order of the blocks in the list. The execution order is defined when the application is created with the GAD-tool or with the operators panel.

Execution order of a function block is defined so that the first block in a task gets the execution order value **1**. Other blocks in the task get the value which is equal to the block instance number of the preceding block.

When GAD is used to create the application, the execution order is defined with symbol attribute values. Each block has a special attribute **ExecutionOrder** whose value must be set when the block is added to the application.
Also the task where the block is added must be set with the **Task** attribute. Execution order and task are not needed for Standard Program FBs.

When the panel or the CMT/DCS 500 is used, the user is prompted for the task number and execution order.

Each task can have a different maximum number of blocks:

Task1:	25
Task2:	100
Task3:	250

Fast Tasks 1 and 2 should be used only when Task 3 is too slow for the application. A FB in Task2 creates 10 times more processor load compared to the same FB in Task3.

Only Application Program FBs are entered to the execution order list. Standard Program FBs are executed by the DCS 500 Drive Program in a fixed order.

It is not possible to synchronise the execution of Application Program FBs and Standard Program FBs.

1.5 Starting and stopping of the FB application

When the application is downloaded to the DCS 500 from the CMT/DCS 500 or when it has been created with the panel or the CMT/DCS 500, the application is normally not running and function blocks are not executed.

The application is started by setting the parameter 2504, FB_APPL_ENABLE, to value 1 (ENABLE). This can be done either with the panel or the CMT.

The application is stopped by setting the parameter 2504, FB_APPL_ENABLE, to value 0 (DISABLE). When the application is stopped, outputs of the application program function blocks are not updated.

It should be carefully considered, what stopping of the application means for the whole drive application. The application should be built so that safety functions are not disabled, when the application is not running.

1.6 Application backup function.

The application is saved to the parameter backup memory together with standard program parameters.

The parameter backup memory has two parameter sets for standard program parameters and one set for application parameters.

When standard program parameter set 1 or 2 is saved, also application parameters are saved.

Application parameters are read only when specifically ordered with the panel or the CMT/DCS 500.
At power-up standard program set 1 and application parameters are read.

Notice that if the application was stopped, when the parameters were saved, the application is not automatically started when the parameters are read from the backup memory at power-up or when specifically ordered.

1.7 Programming with the operator’s panel

The function block application can also be created with the operator's panel, at least when the application is quite small.

The panel has a special mode for FB application programming.
This mode is entered by pressing the FUNC-key and after that the DOUBLE ARROW-key. The panel will display the text: FB PROGRAMMING. Press ENTER-key to start FB programming.

The display will show following texts:
FB PROGRAMMING
FUNC MENU:
TASK EXECUTION LISTS

In this mode you can select between five sub-functions by pressing ARROW UP-or ARROW DOWN-key. These functions are:
- TASK EXECUTION LISTS
- ADD FUNCTION BLOCK
- DEL FUNCTION BLOCK
- MOVE FUNCTION BLOCK
- TERMINATE FB PROGR!

Task Execution lists

You can select this function by pressing the ENTER-key when the text TASK EXECUTION LISTS is on the bottom row.

Next you have to enter the task number. Select 1, 2 or 3 with the single arrow-keys and press ENTER-key.

Now you can go through the list of function blocks in the selected task in the order they are executed. Each block is shown on the bottom row:
TASK EXECUTION LISTS
TASK[3]
1 :[INV (10)]
1 means the position in the execution order list. On the brackets there is the function block type and instance number. LAST (0) means the end of the execution order list. The LAST entry in the list marks the end of task. FBs which come after LAST are not executed.

You can go back to the FB programming menu by pressing the ENTER-key.

Add function block

You can select this function by pressing the ENTER-key when the text ADD FUNCTION BLOCK is on the bottom row.

Next you have to enter the task number. Select 1, 2 or 3 with the single arrow-keys and press ENTER-key.

Now you can go through the list of available function blocks by pressing arrow keys. Single arrow keys are used to select the instance number and double arrow keys to select the block type. The function block is shown on the bottom row:
ADD FUNCTION BLOCK

EXTER FB NUMBER
INV (10)

INV means the function block type and the instance number is in parentheses.
Select the block by pressing the ENTER-key.

Next step is to enter the position of the new block in the execution order list. Function blocks in the execution order list are shown on the bottom row:

ADD FUNCTION BLOCK
EXTER FB TASK POS:
1: [INV (34)]

1 means the position in the execution order list. On the brackets there is the function block type and instance number. LAST (0) means the end of the execution order list.
Press the ENTER-key when you have the correct position on the display. The new function block will be added **before** the block shown on the display.

Delete function block

You can select this function by pressing the ENTER-key when the text DEL FUNCTION BLOCK is on the bottom row.

Next you have to enter the task number. Select 1, 2 or 3 with the single arrow-keys and press ENTER-key.

The function block to be deleted is selected from the execution order list. Function blocks in the execution order list are shown on the bottom row:

DEL FUNCTION BLOCK
EXTER FB TASK POS:
1 :[INV (10)]

1 means the position in the execution order list. On the brackets there is the function block type and instance number. LAST (0) means the end of the execution order list.
Press the ENTER-key when you have the block to be deleted on the display.

Move function block

You can select this function by pressing the ENTER-key when the text MOVE FUNCTION BLOCK is on the bottom row.

Next you have to enter the task number. Select 1, 2 or 3 with the single arrow-keys and press ENTER-key.

The function block to be moved is selected from the execution order list. Function blocks in the execution order list are shown on the bottom row:

MOVE FUNCTION BLOCK
SELECT FB IN TASK[3]
2: [INV (34)]

2 means the position in the execution order list. On the brackets there is the function block type and instance number. LAST (0) means the end of the execution order list.
Press the ENTER-key when you have the block to be moved on the display.

Next step is to enter the new position of the function block in the execution order list. Function blocks in the execution order list are shown on the bottom row:

```
INV    (34)
1:  [INV    (10)]
```

Press the ENTER-key when you have the correct position on the display. The function block will be moved before the block shown on the display.

Terminate FB programming

Select this function by pressing the ENTER-key, when you want to exit from the function block programming.

Connecting FB inputs

FB inputs are connected by setting the corresponding pointers.

2. The Function Block Data Sheets

A Function Block description starts with a short summary that describes the main function of the FB. The graphic symbol of the block is shown above or to the right of the summary.

The type of the block is described under the heading **Call**.

FB pins are described under the heading **Connections**. This section provides a table giving each terminals name, type and usage.

A detailed functional description is presented under the heading **Function**.

The block instance number and pin numbers are given in the table under the heading **Block numbers and pin addresses**.

The description finishes with a guide for using the function block in the Graphical Application Designer (GAD) engineering tool.

Data types

The type of the pin is given as xy, where x is I, O or P if it is an input, an output or a parameter terminal, and y denotes the data type described in Table 2.

Table 1 FB pin types

Designation	Pin type	Comment
I	Input:	can be connected
P	Parameter:	can be given a value
O	Output:	output from the FB

Table 2 Data types

Designation	Data type	Range
I	Integer with sign, 16 bit.	(-32768...32767)
U	Integer without sign, 16 bit.	(0...65535)

Handling of Boolean values

There are a lot of function blocks which handle Boolean input and output values. All logic blocks (e.g. AND, OR etc.) and also many other blocks have Boolean inputs and outputs.

Boolean inputs are following the rule:

- if the value is not equal to 0 (≠ 0), it is logical TRUE
- if the value is equal to 0 (= 0), it is logical FALSE.

Boolean outputs are following the rule:

- if the output is logical TRUE, the output is set to -1
- if the output is logical FALSE, the output is set to 0.

3. DCS 500 Function Blocks listed by type

Function Blocks valid for Software Versions:
21.105, 21.106, 21.107, 21.108, 21.120, 21.121, 21.122, 21.123

Logic blocks

Inverter	**INV**
Two input OR gate	**OR2**
Four input OR gate	**OR4**
Two input AND gate	**AND2**
Four input AND gate	**AND4**
Memory block	**SR**
Memory block with Data input	**SR-D**
Exclusive OR gate	**XOR**

Arithmetic blocks

Two input Adder	**ADD2**
Four input Adder	**ADD4**
Two input Subtraction	**SUB2**
Two input Multiplier	**MUL**
Two input Divider	**DIV2**
Integer Scaling	**MULDIV**
Absolute Value	**ABS**

Selecting blocks

Minimum Selector	**MIN2**
Maximum Selector	**MAX2**
Switch with Changeover	**SW-C1**
One of four Selector	**MUX-I4**

Timing blocks

Time Delay On	**TON**
Time Delay Off	**TOFF**
Retriggerable pulse	**MONO**
Pulse Counter	**COUNT**
Oscillator	**OSC-B**

Comparing or limiting blocks

Comparator	**COMP-I**
Limiter	**LIM-N1**

Control and adjustment blocks

Low-pass Filter	**FILT-I**
Integrator	**INTEG**
PI controller	**PI-I**
Function generator (broken line)	**FUNG-1V**

Constant setting and data transfer blocks

Data transfer, one value	**WRITE1**
Data transfer, four values	**WRITE4**
User parameters	**PAR**
Code Converter, Integer to Binary	**CONV-IB**
Code Converter, Binary to Integer	**CONV-BI**
Extract a single bit value	**BITGET**
Set a single bit value	**BITSET**

Function Blocks valid for Software Version:
21.121

Selecting blocks

Transmit Invert Integer	**TRII**

Control and adjustment blocks

Acceleration weighting	**ACW**
Diameter calculator	**DIAC**
Derivation with low filter	**DT**
Loss weighting	**LSW**
Special controller	**SC**
Special start logic	**SSL**
Tension reference	**TERE**
Torque from tension	**TOTE**
Velocity reference	**VERE**
Winder logic	**WILO**

Constant setting and data transfer blocks

Diameter Relation	**DIAREL**
Parameter of winder	**PARW**

Function Blocks valid for Software Version:
21.122

Constant setting and data transfer blocks

Disable Fault	**DISF**
User parameters	**PARW**
Set Fault	**SETF**

Function Blocks valid for Software Version:
21.123

Constant setting and data transfer blocks

Switch parameter and transfer	**SW-P-T**

Left free for further extensions

Chapter 4 - Function Blocks listed in alphabetic order

4. DCS 500 Function Blocks listed in alphabetic order with short form description

Function Blocks valid for Software Versions:
21.105, 21.106, 21.107, 21.108, 21.120, 21.121, 21.122, 21.123

ABS	Calculates the absolute value of an integer .
ADD2	Adds two integers.
ADD4	Adds four integers.
AND2	Gives the Boolean product of two Boolean values.
AND4	Gives the Boolean product of four Boolean values.
BITGET	Reads the value of the selected bit from an integer value.
BITSET	Sets the value of the selected bit in an integer value.
COMP-I	Compares two integers.
CONV-BI	Converts data in BC coded Boolean inputs into data in integer format.
CONV-IB	Converts data in integer format into BC coded Boolean variables.
COUNT	Pre-setable counter for counting pulses, up or down. With checking of the value relative to 0.
DIV2	Divides two integers.
FILT-I	Single pole low-pass filter for integer signals.
FUNG-1V	Generates a piece-wise linear function of one variable. Function is described by five (x,y) co-ordinates.
INTEG	Gives integration effect. The block has inputs and outputs for limit values, setting of gain and following an external reference.
INV	Inverts a Boolean value.
LIM-N1	Limits an integer value between high and low limits.
MAX2	Selects the larger of two integers.
MIN2	Selects the smaller of two integers.
MONO	Generates a pulse with a given duration when the input changes from 0 to <> 0. Pulse duration and retriggering are specified with parameters.
MUL2	Multiplies two integer values.
MULDIV	Scaling block for the integer values.
MUX-I4	Selector for 4 inputs. Integer address.
OR2	Gives the Boolean sum of two Boolean values.
OR4	Gives the Boolean sum of four Boolean values.
OSC-B	Oscillator for a Boolean variable with variable frequency and pulse time.
PAR	Defines an user parameter that can be stored in the parameter backup FPROM.
PI-I	PI controller. The block permits limiting of the output signal and has a follow function.
SR	Memory for Boolean variables. The block has one Set and one Reset input.

SR-D	Memory for Boolean variables. The block has one Set and one Reset input and Data and Clock inputs for clocking in data.
SUB2	Subtracts one integer from another.
SW-C1	Switching block with one change-over channel.
TOFF	Delay for turning a Boolean signal off.
TON	Delay for turning a Boolean signal on.
WRITE1	Data transfer of one value from source to destination.
WRITE4	Data transfer of four values.
XOR	Gives the exclusive Boolean sum of two Boolean values.

Function Blocks valid for Software Version: **21.121**

ACW	Acceleration weighting
DIAC	Diameter calculator
DT	Derivation with low filter
LSW	Loss Weighting
DIAREL	Parameter for diameter relation
PARW	Parameter for winder application
SC	Special controller
SSL	Special start logic
TERE	Tension reference
TOTE	Torque of tension
TRII	Transmit invert integer
VERE	Velocity reference
WILO	Winder logic

Function Blocks valid for Software Version: **21.122**

DISF	Disabled selected fault
PARW	User parameters
SETF	Set selected fault

Function Blocks valid for Software Version: **21.123**

SWPT Switch parameter and transfer

Chapter 5 - Execution Times and No. of Function Blocks

5. Execution Times and Number of Function Blocks

Function Blocks valid for Software Versions:
21.105, 21.106, 21.107, 21.108, 21.120, 21.121, 21.122, 21.123

Name	Execution time, typical (μs)	Number of blocks
ABS	9	4
ADD2	12	10
ADD4	15	2
AND2	9	40
AND4	12	10
BITGET	12	8
BITSET	17	8
COMP-I	11	10
CONV-BI	70	2
CONV-IB	67	2
COUNT	26	2
DIV2	15	4
FILT-I	27	5
FUNG-1V	37	1
INTEG	51	2
INV	8	25
LIM-N1	18	5
MAX2	10	5
MIN2	10	5
MONO	20	5
MUL2	12	4
MULDIV	23	10
MUX-I4	12	2
OR2	9	30
OR4	12	10

Name	Execution time, typical (µs)	Number of blocks
OSC-B	25	1
PAR	-	20
PI-I	75	1
SR	11	10
SR-D	16	5
SUB2	11	4
SW-C1	9	15
TOFF	17	10
TON	17	15
WRITE1	10	10
WRITE4	20	2
XOR	9	10

Function Blocks valid for Software Version:
21.121

Name	Execution time, typical (µs)	Number of blocks
ACW	101	1
DIAC	59	1
DT	52	2
LSW	43	1
DIAREL	0	1
PARW	0	11
SC	59	1
SSL	36	1
TERE	79	1
TOTE	55	1
TRII	24	4
VERE	46	1
WILO	30	1

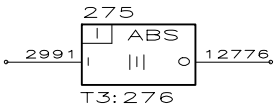
Function Blocks valid for Software Version: **21.122**

Name	Execution time, typical (µs)	Number of blocks
DISF_487	28	1
PARW_511 ... _525	0	15
SETF_486	28	1

Function Blocks valid for Software Version: **21.123**

Name	Execution time, typical (µs)	Number of blocks
DSWPT_488 ... _493	52	6

Absolute Value **ABS**



Summary

ABS (**ABS**olute value) is used to obtain the absolute value of an integer value.

Call ABS_275...ABS_278

Connections *Table 1*

Name	Type	Description
I	II	Input for number, whose absolute value is to be obtained.
O	OI	Output for the absolute value of I.

Function

The absolute value of the input I is stored at the output O.

Block numbers and pin addresses *Table 2*

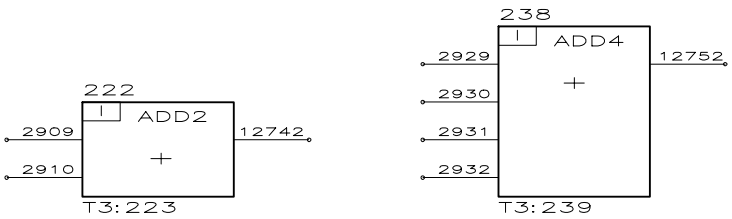
FB number	275	276	277	278
I	2991	2992	2993	2994
O	12776	12777	12778	12779

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Adder

ADD



Summary

ADD (**Add**er) is used to calculate the sum of integer values.

Call ADD2_222...ADD2_231, ADD4_238, ADD4_239

Connections *Table 1*

Name	Type	Description
IN1	II	Input for addend.
IN2	II	Input for addend
IN3	II	Input for addend.
IN4	II	Input for addend.
O	OI	Output for sum.

Function

The values at inputs IN1...IN4 are added and the sum is stored at output O.

Overflow

If the maximum positive or negative value is exceeded, the output is limited to the highest or lowest allowable value for the integer data type.

Block numbers and pin addresses *Table 2*

ADD2:

FB number	222	223	224	225	226	227	228	229	230	231
IN1	2909	2911	2913	2915	2917	2919	2921	2923	2925	2927
IN2	2910	2912	2914	2916	2918	2920	2922	2924	2926	2928
O	12742	12743	12744	12745	12746	12747	12748	12479	12750	12751

ADD4:

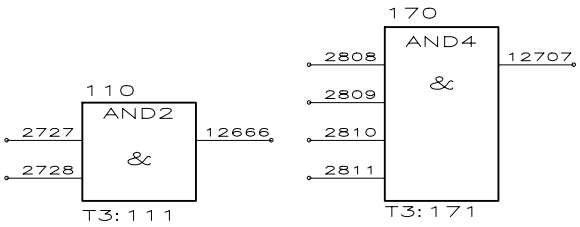
FB number	238	239
IN1	2929	2933
IN2	2930	2934
IN3	2931	2935
IN4	2932	2936
O	12752	12753

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

And Gate

AND



Summary

AND is used to form a Boolean product of Boolean input values.

Call AND2_110...AND2_149, AND4_170...AND4_179

Connections *Table 1*

Name	Type	Description
IN1	II	Input
IN2	II	Input
IN3	II	Input
IN4	II	Input
O	OI	Output

Function

The output signal is set to -1 if all inputs IN1..IN4 are <> 0. See the truth table below.

Truth table AND2 *Table 2*

IN1	IN2	O
= 0	= 0	0
= 0	≠ 0	0
≠ 0	= 0	0
≠ 0	≠ 0	-1

Block numbers and pin addresses *Table 3*

AND2:

FB number	110	111	112	113	114	115	116	117	118	119
IN1	2727	2729	2731	2733	2735	2737	2739	2741	2743	2745
IN2	2728	2730	2732	2734	2736	2738	2740	2742	2744	2746
O	12666	12667	12668	12669	12670	12671	12672	12673	12674	12675

FB number	120	121	122	123	124	125	126	127	128	129
IN1	2747	2749	2751	2753	2755	2757	2759	2761	2763	2765
IN2	2748	2750	2752	2754	2756	2758	2760	2762	2764	2766
O	12676	12677	12678	12679	12680	12681	12682	12683	12684	12685

FB number	130	131	132	133	134	135	136	137	138	139
IN1	2767	2769	2771	2773	2775	2777	2779	2781	2783	2785
IN2	2768	2770	2772	2774	2776	2778	2780	2782	2784	2786
O	12686	12687	12688	12689	12690	12691	12692	12693	12694	12695

FB number	140	141	142	143	144	145	146	147	148	149
IN1	2787	2789	2791	2793	2795	2797	2799	2802	2804	2806
IN2	2788	2790	2792	2794	2796	2798	2801	2803	2805	2807
O	12696	12697	12698	12699	12701	12702	12703	12704	12705	12706

AND4:

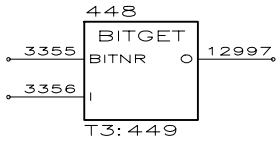
FB number	170	171	172	173	174	175	176	177	178	179
IN1	2808	2812	2816	2820	2824	2828	2832	2836	2840	2844
IN2	2809	2813	2817	2821	2825	2829	2833	2837	2841	2845
IN3	2810	2814	2818	2822	2826	2830	2834	2838	2842	2846
IN4	2811	2815	2819	2823	2827	2831	2835	2839	2843	2847
O	12707	12708	12709	12710	12711	12712	12713	12714	12715	12716

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Read One Bit

BITGET



Summary

BITGET block is used to read a selected bit from an integer value. It is used to read packed Boolean status information coming from a communication link or created by the drive.

Call BITG_448...BITG_455

Connections *Table1*

Name	Type	Description	Values
BITNR	II	Input for the selected **BIT N**umbe**R** to read	0...15
I	II	Input for the packed Boolean data	
O	OI	Status of the selected bit	

Function

The bit value of the input data I, specified by BITNR, is loaded to the output O. When selected input bit is 0 the output O is 0, and when input bit is 1 then the output is -1. If BITNR is not in the range 0...15 then the output O is set to 0.

Block numbers and pin addresses *Table 2*

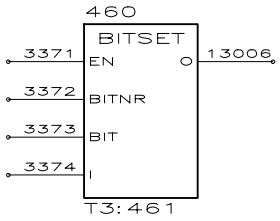
FB number	448	449	450	451	452	453	454	455
BITNR	3355	3357	3359	3361	3363	3365	3367	3369
I	3356	3358	3360	3362	3364	3366	3368	3370
O	12997	12998	12999	13001	13002	13003	13004	13005

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Set One Bit

BITSET



Summary

BITSET block is used to set a selected bit of an integer value. It is used when writing status information to a packed Boolean parameter or signal.

Call BITS_460...BITS_467

Connections *Table 1*

Name	Type	Description	Values
EN	II	**EN**able. Input for storage of a new value each time the block is executed. If this input is reset (to 0), then the current value of I is loaded unchanged to the output O.	
BITNR	II	Input for selecting the **BIT N**umbe**R** to set.	0...15
BIT	II	Value of the **BIT**.	
I	II	Data **I**nput.	
O	OI	**O**utput of the set data.	

Function

If the value of the input EN is set (to ≠ 0) then the value of bit number, determined by the input BITNR, of the input data I, is replaced by the value of the input BIT and the result is loaded to the output O.

If the value of EN is reset (to 0) then the value of the input data I is loaded unchanged to the output O.

If the BIT input value is 0 then the selected bit at the output data is 0, and if the BIT input value is ≠ 0 then the selected bit at output data is 1.

If BITNR is not in the range 0 to 15 then the output O is set to the value of I. If BITNR is 15 then the value of O will be negative.

Block numbers and pin addresses

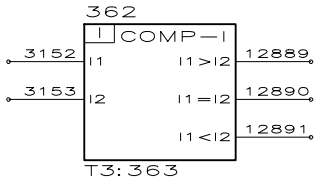
Table 2

FB number	460	461	462	463	464	465	466	467
EN	3371	3375	3379	3383	3387	3391	3395	3399
BITNR	3372	3376	3380	3384	3388	3392	3396	3401
BIT	3373	3377	3381	3385	3389	3393	3397	3402
I	3374	3378	3382	3386	3390	3394	3398	3403
O	13006	13007	13008	13009	13010	13011	13012	13013

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Comparator Integer — COMP-I



Summary

COMP-I (**COMP**arator - **I**nteger) is used to compare two integers.

Call COMP_362...COMP_371

Connections *Table 1*

Name	Type	Description
I1	II	**I**nput **1**. Input, whose value is compared with the value at input I2
I2	II	**I**nput **2.** Input, whose value is compared with the value at input I1
I1>I2	OI	**I**nput **1 > I**nput **2**. Output which is set to -1 if the value at input I1 is greater than the value at input I2.
I1=I2	OI	**I**nput **1 = I**nput **2**. Output which is set to -1 if the value at input I1 is equal to the value at input I2.
I1<I2	OI	**I**nput **1 < I**nput **2**. Output which is set to -1 if the value at input I1 is less than the value at input I2.

Function

The values at the two inputs are compared and the result of the comparison can be read at the outputs I1<I2, I1=I2 and I1>I2.

At start-up, output I1=I2 is set to -1 before the first execution of the block regardless of the values of inputs I1 and I2.

Block numbers and pin addresses

Table 2

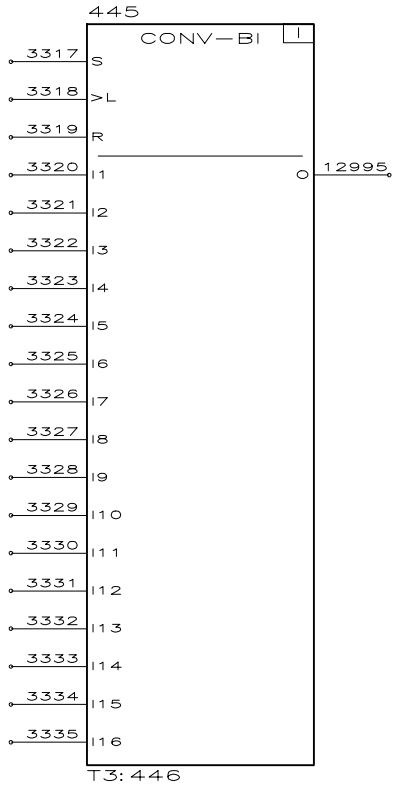
FB number	362	363	364	365	366	367	368	369	370	371
I1	3152	3154	3156	3158	3160	3162	3164	3166	3168	3170
I2	3153	3155	3157	3159	3161	3163	3165	3167	3169	3171
I1>I2	12889	12892	12895	12898	12902	12905	12908	12911	12914	12917
I1=I2	12890	12893	12896	12899	12903	12906	12909	12912	12915	12918
I1<I2	12891	12894	12897	12901	12904	12907	12910	12913	12916	12919

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language),*

Code Converter Binary to Integer

CONV-BI



Summary

CONV-BI (**CONV**erter-**B**oolean to **I**nteger) converts data from BC formed Boolean variables into integers. The block has a memory function for storage of converted data.

Call CVBI_445,..CVBI_446

Connections *Table 1*

Name	Type	Description
S	II	**S**et. Input for storage of a new value each time the block is executed. When this input is reset (to 0), the latest calculated value will remain at the outputs
L	II	**L**oad. Dynamic input for loading data.
R	II	**R**eset. Input for clearing the output. This input overrides S and L inputs.

Name	Type	Description
I1	II	**I**nput **1**. Input 1 for BC.
I2	II	**I**nput **2**. Input 2 for BC.
I3	II	**I**nput **3.** Input 3 for BC.
I4	II	**I**nput **4.** Input 4 for BC.
I5	II	**I**nput **5**. Input 5 for BC.
I6	II	**I**nput **6.** Input 6 for BC.
I7	II	**I**nput **7**. Input 7 for BC.
I8	II	**I**nput **8**. Input 8 for BC.
I9	II	**I**nput **9**. Input 9 for BC.
I10	II	**I**nput **10**. Input 10 for BC.
I11	II	**I**nput **11**. Input 11 for BC.
I12	II	**I**nput **12** Input 12 for BC.
I13	II	**I**nput **13** Input 13 for BC.
I14	II	**I**nput **14**. Input 14 for BC.
I15	II	**I**nput **15**. Input 15 for BC.
I16	II	**I**nput **16.** Input 16 for BC.
O	OI	**O**utput. Output for converted data.

Function

The block can convert 16 bit binary code (BC) to integer of data type I (16 bits) .

Examples of input values with different recordings

Table 2

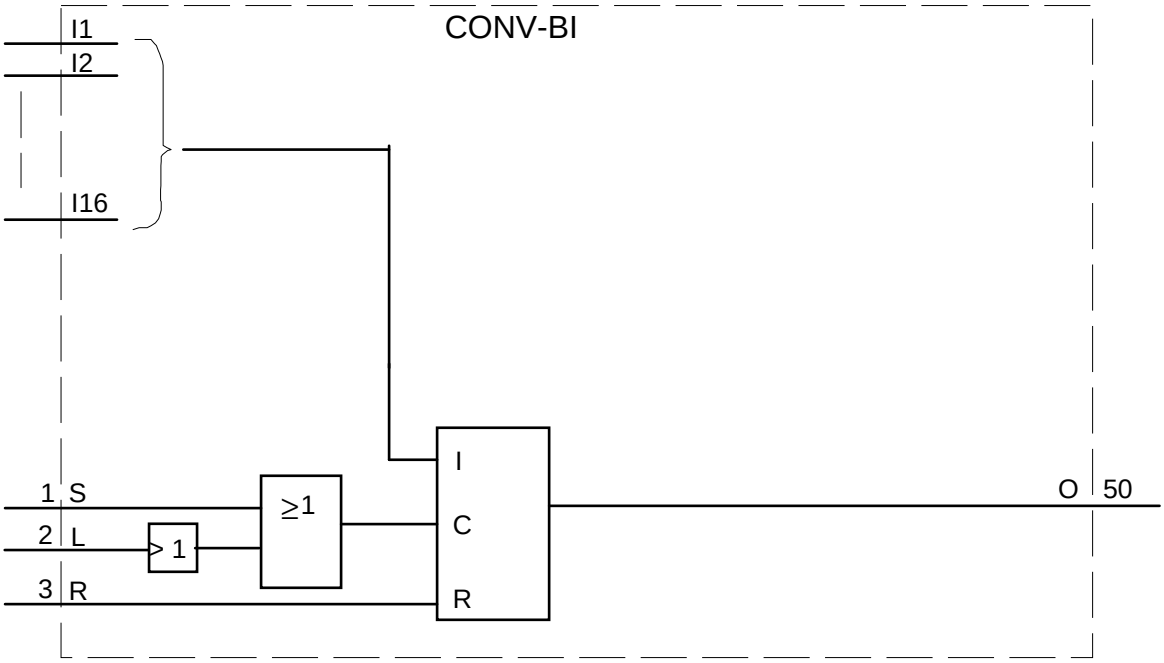
Input integer value	
Input	BC with sign
I1	1
I2	2
I3	4
I4	8
I5	16
I6	32
I7	64
I8	128
I9	256
.	.
I16	sign

Storage of data

When input L is set (from 0 to ≠ 0), the code at inputs I1 to I16 is immediately converted and stored. If input S is set (to ≠ 0), the input code is converted and the integer is stored each time the block is executed. When S is reset (to 0) after having been set, the data stored most recently remains. The input S overrides the input L, i.e. when S is set (to ≠ 0), L has no effect.

Clearing

The input R clears the output and prevents all further storage of data while R is set (to ≠ 0).



Function diagram

Block numbers and pin addresses

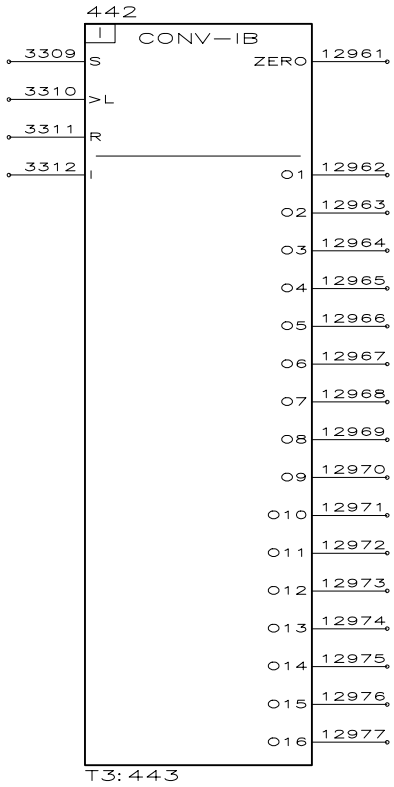
Table 3

FB number	445	446
S	3317	3336
L	3318	3337
R	3319	3338
I1	3320	3339
I2	3321	3340
I3	3322	3341
I4	3323	3342
I5	3324	3343
I6	3325	3344
I7	3326	3345
I8	3327	3346
I9	3328	3347
I10	3329	3348
I11	3330	3349
I12	3331	3350
I13	3332	3351
I14	3333	3352
I15	3334	3353
I16	3335	3354
O	12995	12996

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language),*

Code Converter Integer to Binary — CONV-IB



Summary

CONV-IB (**CONV**erter- **I**nteger to **B**oolean) converts data from integers into BC formed Boolean variables. The block has a memory function for storage of converted data.

Call CVIB_442, CVIB_443

Connections *Table 1*

Name	Type	Description
S	II	**S**et. Input for storage of a new value each time the block is executed. When this input is reset (to 0), the latest calculated value will remain at the outputs
L	II	**L**oad. Dynamic input for loading data.
R	II	**R**eset. Input for clearing the output. This input overrides S and L inputs.
I	II	**I**nput. Input for the integer value to be converted.

Name	Type	Description
ZERO	OI	Output set to -1 when the value at the input I is zero.
O1	OI	**O**utput **1**. Output 1 for BC.
O2	OI	**O**utput **2**. Output 2 for BC.
O3	OI	**O**utput **3**. Output 3 for BC.
O4	OI	**O**utput **4**. Output 4 for BC.
O5	OI	**O**utput **5**. Output 5 for BC.
O6	OI	**O**utput **6**. Output 6 for BC.
O7	OI	**O**utput **7**. Output 7 for BC.
O8	OI	**O**utput **8**. Output 8 for BC.
O9	OI	**O**utput **9**. Output 9 for BC.
O10	OI	**O**utput **10**. Output 10 for BC.
O11	OI	**O**utput **11**. Output 11 for BC.
O12	OI	**O**utput **12**. Output 12 for BC.
O13	OI	**O**utput **13**. Output 13 for BC.
O14	OI	**O**utput **14**. Output 14 for BC.
O15	OI	**O**utput **15**. Output 15 for BC.
O16	OI	**O**utput **16**. Output 16 for BC.

Function

The block can convert integers of the data type I.

Examples of output values

Table 2

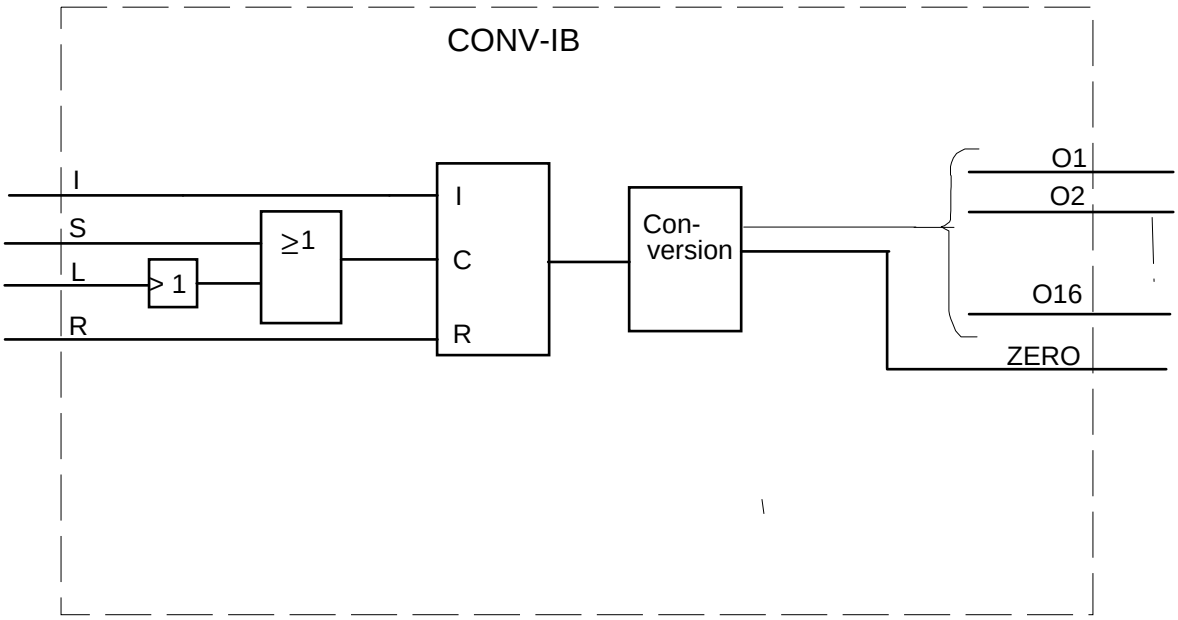
Output integer value	
Output	BC
O1	1
O2	2
O3	4
O4	8
O5	16
O6	32
O7	64
O8	128
O9	256
O10	512
.	.
.	.
O16	2^{15}

Storage of data

When input L is set (from 0 to ≠ 0), the integer at input I is immediately converted and if input S is set (to ≠ 0), the integer input is stored each time the block is executed. When S is reset (to 0) after having been set (to ≠ 0), the data stored most recently remains until the block is executed once more with one of the inputs S, L or R set. The input S overrides the input L, i.e. when S is set (to ≠ 0), L has no effect.

Clearing

The input R clears the output to 0 and prevents all further storage of data while R is set (to ≠ 0).



Function diagram

Block numbers and pin addresses *Table 3*

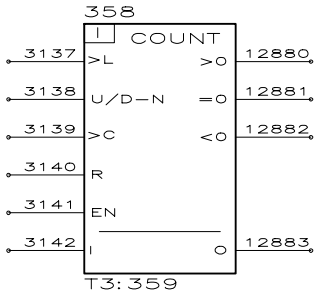
FB number	442	443
S	3309	3313
L	3310	3314
R	3311	3315
I	3312	3316
ZERO	12961	12978
O1	12962	12979
O2	12963	12980
O3	12964	12981
O4	12965	12982
O5	12966	12983
O6	12967	12984
O7	12968	12985
O8	12969	12986
O9	12970	12987
O10	12971	12988
O11	12972	12989
O12	12973	12990
O13	12974	12991
O14	12975	12992
O15	12976	12993
O16	12977	12994

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language),*

Counter

COUNT



Summary

COUNT (**COUNT**er) is a presetable counter for counting pulses, up or down. The counter also monitors the relation of the counter value to 0.

Call COUN_358,..COUN_359

Connections *Table 1*

Name	Type	Description
L	II	**L**oad. Loads the counter with the value at input I
U/D-N	II	**U**p/**D**own-**N**. Input which determines if the counter is to count up (U/D-N ≠ 0) or down (U/D-N = 0)
C	II	**C**lock. When this input changes from 0 to ≠ 0 the counter counts up or down in accordance with the status of the input U/D-N.
R	II	**R**eset. Input which clears the counter and prevents all further counting or loading. **This input overrides all other inputs!**
EN	II	**EN**able. Input which is set to ≠ 0 to permit counting or loading. Reset execution is, however, performed independently of EN.
I	II	**I**nput. Input for new value when loading.
>0	OI	Output which is set to -1 when the value of the counter is greater than 0.
=0	OI	Output which is set to -1 when the value of the counter is 0.
<0	OI	Output which is set to -1 when the value of the counter is less than 0.
O	OI	**O**utput. Output for counter value.

Function

When input C is set, the counter value immediately increases or decreases. The value increases if U/D-N≠ 0 and decreases if U/D-N = 0. The duration of the counter period cannot be less than 2 * the cycle time of the program.

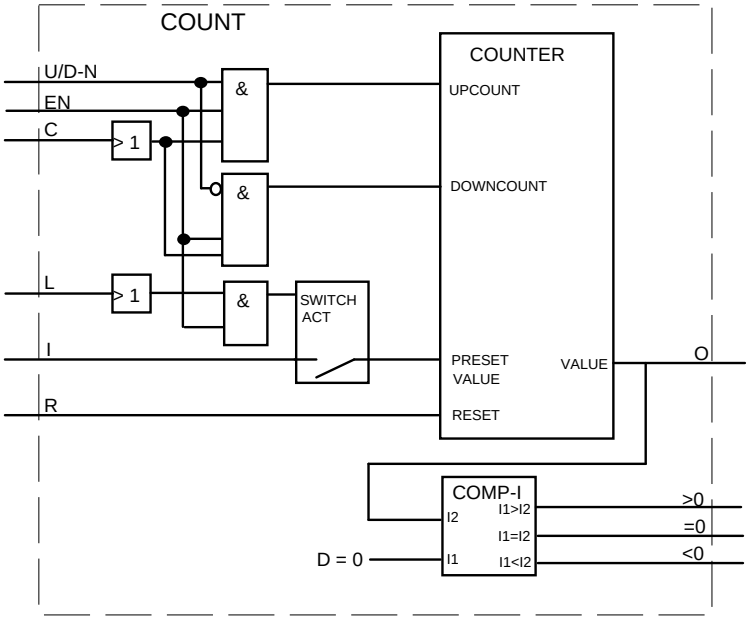
When the input L is set to ≠ 0, the counter is loaded with the value at input I. If both input L and C are set to ≠ 0 simultaneously, the counter is first loaded after which an up or down count is performed. The input EN must be set to ≠ 0 for the counter to count or load a new value.

Clearing

The input R clears the counter and prevents all further counting or loading. R overrides EN.

Supervision

The status outputs specify the relation of the counter value to zero (> 0, = 0, < 0). When the counter reaches its least or greatest value for the data type, all counting ceases.



Function diagram

Block numbers and pin addresses *Table 2*

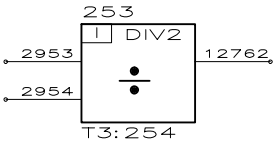
FB number	358	359
L	3137	3143
U/D-N	3138	3144
C	3139	3145
R	3140	3146
EN	3141	3147
I	3142	3148
>0	12880	12884
=0	12881	12885
<0	12882	12886
O	12883	12887

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Divider

DIV



Summary

DIV is used for division of two integers.

Call DIV_253...DIV_256

Connections *Table 1*

Name	Type	Description
IN1	II	Input for dividend.
IN2	II	Input for divisor.
O	OI	Output for quotient.

Function

The value at input IN1 is divided by the value at input IN2. The quotient is stored at output O.

Overflow

If the maximum positive or negative value is exceeded, the output is limited to the highest or lowest allowable value for the integer type.

Block numbers and pin addresses *Table 2*

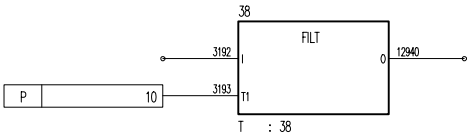
FB number	253	254	255	256
IN1	2953	2955	2957	2959
IN2	2954	2956	2958	2960
O	12762	12763	12764	12765

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language).*

Filter Integer

FILT-I



Summary

FILTer **I**nteger block is used as a single pole low pass filter for integer values.

Call FILT_383...FILT_387

Connections *Table 1*

Name	Type	Description	Values
I	II	**I**nput.	
T1	PI	Parameter. Filter time constant 1=1ms.	0...32767 [ms]
O	OI	**O**utput. Filtered value.	

Function

The step response in the time plane for a single low pass filter is:

$$O(t) = I(t)(1 - e^{-t/T1})$$

The transfer function for a single low pass filter is:

$$G(s) = 1/(1 + sT1)$$

The filtering algorithm is calculated using the following formula:

$$O = \frac{I + (T1/TS) \times O_{n-1}}{T1/TS + 1}$$

Where TS is the cycle time of the block in milliseconds and O_{n-1} is the output from the previous execution cycle. If T1/TS < 1 then the output O is equal to the input I. Internal calculation of the algorithm is done with 32-bit accuracy to avoid offset errors.

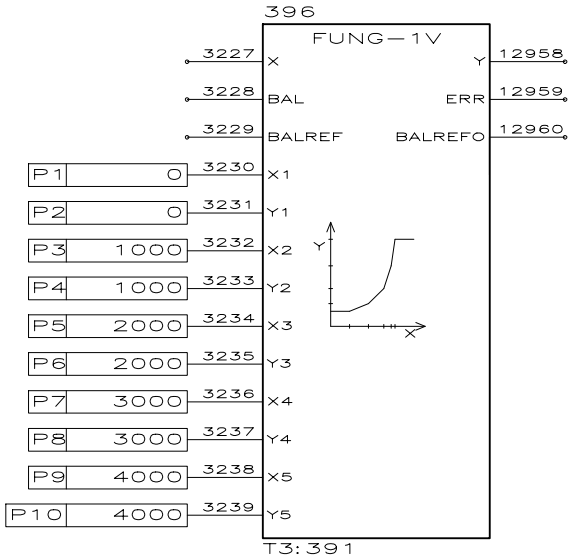
Block numbers and pin addresses *Table 2*

FB number	383	384	385	386	387
I	3192	3194	3196	3198	3201
T1	3193	3195	3197	3199	3202
O	12940	12941	12942	12943	12944

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Function Generator FUNG-1V



Summary

FUNG-1V (**FUN**ction **G**enerator **1 V**ariable) is used for generation of an user defined function of one variable, y = f(x). The function is defined by 5 (x,y) co-ordinates. Linear interpolation is used for values between these co-ordinates.

Call FUNG_396

Connections *Table 1*

Name	Type	Description
X	II	**X**-value. Input for X-value.
BAL	II	**BAL**ance. Input for activation of balancing.
BALREF	II	**BAL**ance **REF**erence. Value which the Y-output is to adopt with balancing.
X1	PI	Co-ordinate 1 on X-axis.
Y1	PI	Co-ordinate 1 on Y-axis.
X2	PI	Co-ordinate 2 on X-axis.
Y2	PI	Co-ordinate 2 on Y-axis.
X3	PI	Co-ordinate 3 on X-axis.
Y3	PI	Co-ordinate 3 on Y-axis.
X4	PI	Co-ordinate 4 on X-axis.
Y4	PI	Co-ordinate 4 on Y-axis.

Name	Type	Description
X5	PI	Co-ordinate 5 on X-axis.
Y5	PI	Co-ordinate 5 on Y-axis.
Y	OI	**Y**-value. Output for Y-value.
ERR	OI	**ERR**or. Output set (to -1) if X outside the value of XTAB or if Y, on balancing, is outside the YTAB values.
BALREFO	OI	**BAL**ance **REF**erence **O**utput. Output for calculated X-value with balancing.

Function

The function generator FUNG-1V for one variable calculates an output signal Y for a value at the input X. The calculation is performed in accordance with a piece by piece linear function which is determined by the inputs X1,Y1...X5,Y5. For each X-value, there is a corresponding Y-value. The Y-value at the output is calculated by means of linear interpolation between the two X-values which are the next lower and higher values for the input value X. The values X1...X5 must be in ascending order.

Interpolation

The function generated is performed as follows:

$y = y_k+(x-x_k)*(y_{k+1}-y_k)/(x_{k+1}-x_k)$

Balancing

If BAL is set to $\neq 0$, the value at Y is set to the value of the input BALREF. The X-value which corresponds to this Y-value is obtained at the output BALREFO. On balancing, the X-value is calculated by interpolation in the same way the Y-value is calculated during normal operation. If balancing is used, the values Y1...Y5 must be in ascending or descending order.

Error Signal

If the input signal X is outside the range defined by X1...X5, the ERR output is set to -1. The Y-value is then set to Y5 or Y1 value depending on value of X.
ERR is also set to -1 if BALREF is outside the Y1...Y5 value range when BAL is set to $\neq 0$. The value at Y is then set to the value at the input BALREF and BALREFO is set to X5 or X1 value depending on value of BALREF.

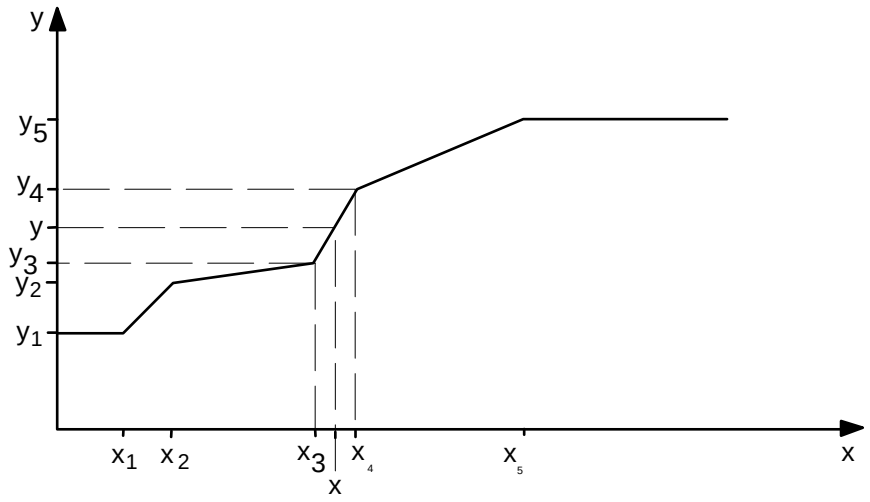


Figure 2. Example of function

Block numbers and pin addresses

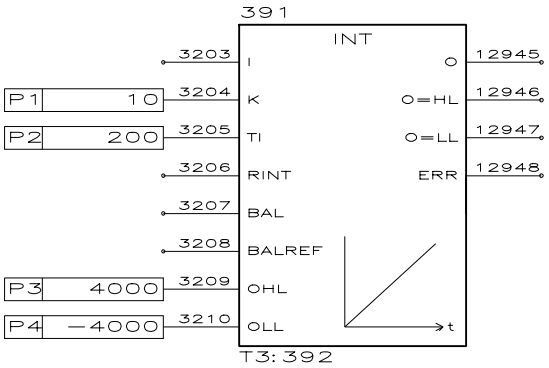
Table 2

FB number	396
X	3227
BAL	3228
BALREF	3229
X1	3230
Y1	3231
X2	3232
Y2	3233
X3	3234
Y3	3235
X4	3236
Y4	3237
X5	3238
Y5	3239
Y	12958
ERR	12959
BALREFO	12960

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Integrator INT



Summary

INT (**INT**egrator) is used to give an integration effect. The output signal can be limited with limit values specified with parameters. The balancing function permits the output signal to track an external reference and permits a bumpless return to the normal function.

Call INT_391,..INT_392

Connections *Table 1*

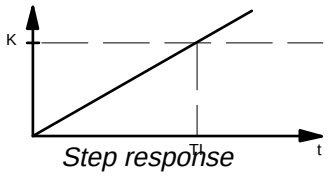
Name	Type	Description
I	II	**I**nput. Input for actual value.
K	PI	Parameter for setting proportional gain. 100 = gain=1.
TI	PI	**T**ime **I**ntegration. Parameter for time constant for integration. 1 = 1ms.
RINT	II	**R**eset **INT**egrator. Input for clearing the integrator.
BAL	II	**BAL**ance. Input for activation of tracking
BALREF	II	**BAL**ance **REF**erence. Input for reference value when tracking.
OHL	PI	**O**utput **H**igh **L**imit. Parameter for upper limit value.
OLL	PI	**O**utput **Low L**imit. Parameter for lower limit value.
O	OI	**O**utput. Output signal.
O=HL	OI	**O**utput **= H**igh **L**imit. Output which is set to -1 if the output reaches the higher limit value.
O=LL	OI	**O**utput **= L**ow **L**imit. Output which is set to -1 if the output reaches the lower limit value.
ERR	OI	**ERR**or. Set to -1 if OHL is less than OLL

Function

The INT function can be written in the time plane as:

$O(t) = K/TI(\neq I(t)dt)$

The main property when controlling is that the output signal retains its value when the input signal I(t) = 0.



Step response

The step response in the time plane is

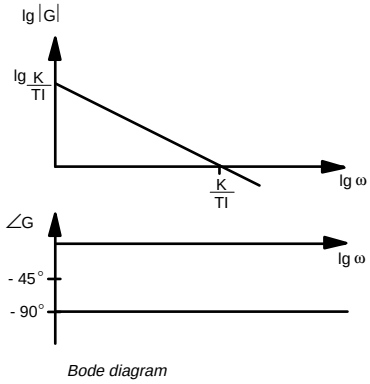
$O(t) = k*I(t)*t/T1.$

The transfer function for an integrator is:

$G(s) = K(1/sTI).$

Gain, Integration Time Constant and Sampling Time

The constant K*TS/TI is calculated in advance to reduce the execution time of the block to a minimum. The result is stored internally in the block. This constant is calculated again if TI or K has changed, or if the sampling time TS has changed. When calculating, a test is performed to check whether TS/TI < 1. In this case TS/TI is set equal to 1.



Bode diagram

Clearing of Integrator

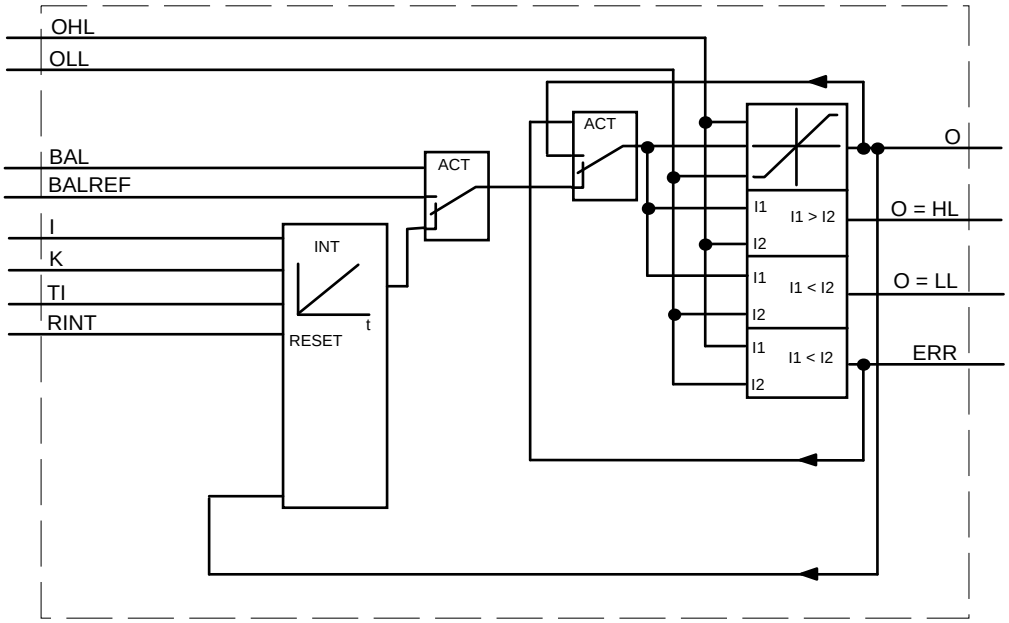
The algorithm is cleared when RINT goes to $\neq 0$.

Tracking

If BAL is set to $\neq 0$, the integrator immediately goes into tracking and the output O is set to the value of the input BALREF. If the value at BALREF exceeds the output signal limits, the output is set to the applicable limit value. On return to the normal function, the value of output O during the previous execution cycle in tracking mode is kept for one cycle, after which integration will be performed for this value.

Limitation Function

The limitation function limits the output signal to the values at the inputs OHL for upper limit and OLL for the lower limit. If the actual value exceeds the upper limit, the output O = HL is set to -1. If it falls below the lower limit, the output O = LL is set to -1. The block checks that the upper limit value OHL is greater than the lower limit value OLL. If not, the output ERR is set to -1. While the error status persists, the outputs O = HL, O = LL and O retain the values they had in the previous execution cycle before the error occurred. After limitation or error status, normal integration is performed from the current value.



Function diagram

Block numbers and pin addresses *Table 2*

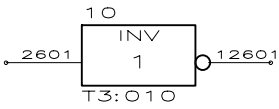
FB number	391	392
I	3203	3211
K	3204	3212
TI	3205	3213
RINT	3206	3214
BAL	3207	3215
BALREF	3208	3216
OHL	3209	3217
OLL	3210	3218
O	12945	12949
O=HL	12946	12950
O=LL	12947	12951
ERR	12948	12952

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Inverter

INV



Summary

INV is used for inverting Boolean variables.

Call INV_010...INV_034

Connections *Table 1*

Name	Type	Description
IN	II	Input.
O	OI	Output of inverted input value.

Function

The output signal from the INV block is set (to -1) if the input signal to the block is 0, and reset (to 0) when the input signal is ≠ 0. See the truth table below.

Truth Table INV *Table 2*

IN	O
= 0	-1
≠ 0	0

Block numbers and pin addresses *Table 3*

FB number	10	11	12	13	14	15	16	17	18	19
IN	2601	2602	2603	2604	2605	2606	2607	2608	2609	2610
O	12601	12602	12603	12604	12605	12606	12607	12608	12609	12610

FB number	20	21	22	23	24	25	26	27	28	29
IN	2611	2612	2613	2614	2615	2616	2617	2618	2619	2620
O	12611	12612	12613	12614	12615	12616	12617	12618	12619	12620

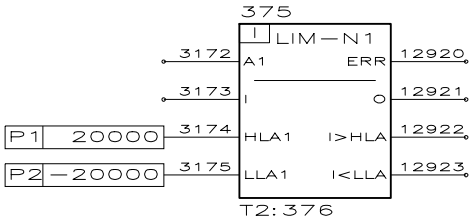
FB number	30	31	32	33	34
IN	2621	2622	2623	2624	2625
O	12621	12622	12623	12624	12625

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Limiter

LIM-N



Summary

LIM-N (**LIM**iter - 1 of - **N** address) is used for limitation of integer values.

Call LIMN_375...LIMN_379

Connections *Table 1*

Name	Type	Description
A1	II	**A**ddress **1**. Input which, when set to ≠ 0, limits the output O to the limit values defined by the parameters HLA1 and HLL1.
I	II	**I**nput. Input which is connected to the signal being limited.
HLA1	PI	**H**igh **L**imit **A**ddress **1**. Parameter for upper limit value which limits the signal when the input A1 is set to ≠ 0.
LLA1	PI	**L**ow **L**imit **A**ddress **1**. Parameter for lower limit value which limits the signal when the input A1 is set to ≠ 0.
ERR	OI	**ERR**or. Output which is set to -1 when the limit for high level is less than the limit for low level.
O	OI	**O**utput. Output for the limited signal.
I>HLA	OI	**I**nput **>** **H**igh **L**imit **A**ddress. Output which is set to -1 when the upper limit of the block is reached.
I<LLA	OI	**I**nput **>** **L**ow **L**imit **A**ddress. Output which is set to -1 when the lower limit of the block is reached.

Function

LIM-N is used to limit a signal with given limits. Boolean output signals indicate when the output is limited.

Selection of Limit Value

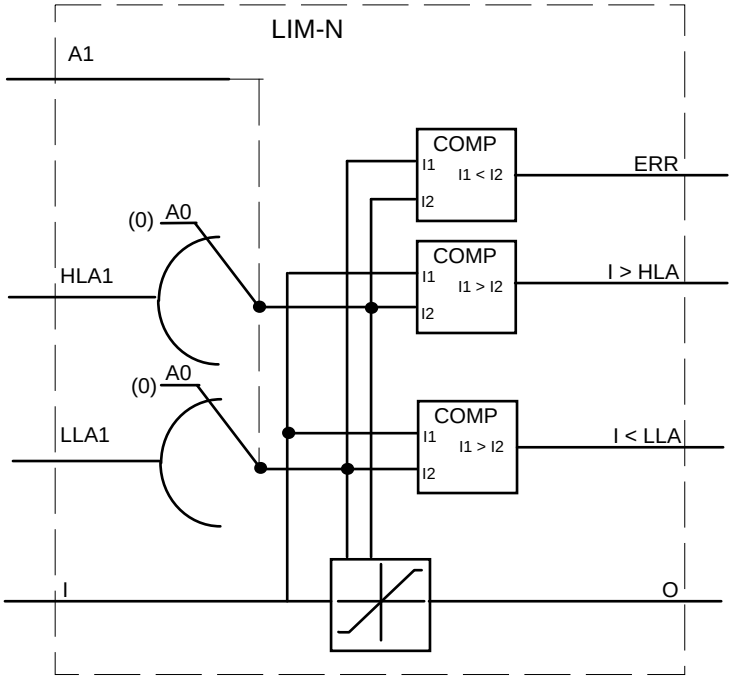
The limit value parameters HLA1 and LLA1 are activated with the input A1. If the input A1 is≠ 0, the output is limited by HLA1 and LLA1. If A1 is 0, output O is limited to the data value 0.

Limiting

When the input I exceeds the selected limit, the output O is limited to the limit value. One of the outputs I > HLA or I < LLA will then be set to -1 depending on which limit was exceeded.

Supervision of Limit Values

The block checks that the limit value HLA1 is greater than the limit LLA1. If HLA1 is less than LLA1, the error signal output ERR is set to -1. The output O and the limit value outputs I > HLA and I < LLA are retained from the previous execution cycle before the error status developed.



Function diagram

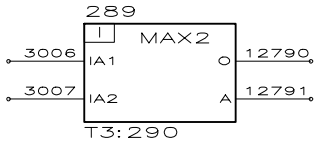
Block numbers and pin addresses *Table 2*

FB number	375	376	377	378	379
A1	3172	3176	3180	3184	3188
I	3173	3177	3181	3185	3189
HLA1	3174	3178	3182	3186	3190
LLA1	3175	3179	3183	3187	3191
ERR	12920	12924	12928	12932	12936
O	12921	12925	12929	12933	12937
I>HLA	12922	12926	12930	12934	12938
I<HLA	12923	12927	12931	12935	12939

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Maximum Selector MAX



Summary

MAX (**MAX**imum selector) is used to select the highest of two integer values.

Call MAX2_289...MAX2_293

Connections *Table 1*

No	Name	Type	Description
IA1	II		**I**nput **A**ddress **1**. Input which is compared with the IA2 input
IA2	II		**I**nput **A**ddress **2**. Input which is compared with the IA1 input
O	OI		**O**utput. Output for the highest value.
A	OI		**A**ddress. Output for the number of the input having the highest value.

Function

The values at the inputs IA1 and IA2 are compared and the greatest value is copied to the output O. The number of the input with the highest value is written to the output A. If the two highest signal values are equal when the block is executed for the first time, the signal with the lowest connection number will be the highest.

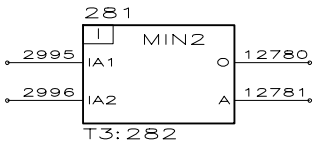
Block numbers and pin addresses *Table 2*

FB number	289	290	291	292	293
IA1	3006	3008	3010	3012	3014
IA2	3007	3009	3011	3013	3015
O	12790	12792	12794	12796	12798
A	12791	12793	12795	12797	12799

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Minimum Selector MIN



Summary

MIN (**MIN**imum selector) is used to select the lowest of two integer values.

Call MIN2_281...MIN2_285

Connections *Table 1*

Name	Type	Description
IA1	II	**I**nput **A**ddress **1**. Input which is compared with the IA2 input
IA2	II	**I**nput **A**ddress **2**. Input which is compared with the IA1 input
O	OI	**O**utput. Output for the lowest value.
A	OI	**A**ddress. Output for the number of the input having the lowest value.

Function

The values at the inputs IA1 and IA2 are compared and the lowest value is copied to the output O. The number of the input with the lowest value is written to the output A. If the two lowest signal values are equal when the block is executed for the first time, the signal with the lowest connection number will be the lowest.

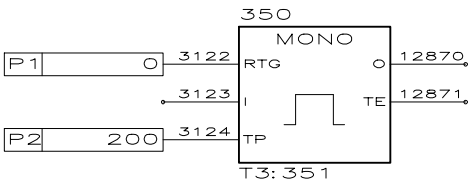
Block numbers and pin addresses *Table 2*

FB number	281	282	283	284	285
IA1	2995	2997	2999	3002	3004
IA2	2996	2998	3001	3003	3005
O	12780	12782	12784	12786	12788
A	12781	12783	12785	12787	12789

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Mono Function — MONO



Summary

MONO function block can be used e.g. for time limitation of output operation and for automatic functions. The block can also be used for impulse extension, stall alarm generation, etc.

Call MONO_350...MONO_354

Connections *Table 1*

Name	Type	Description
RTG	PI	**R**e **T**rig**G**. Parameter to select whether or not the MONO function is retriggable. If RTG is set to ≠ 0 , the retriggable mode is selected.
I	II	Input. Input which starts the MONO function when the input value changes from 0 to ≠ 0.
TP	PI	**T**ime **P**ulse. Parameter for setting of the pulse time. 1 = 10ms. Max. 650s
O	OI	**O**utput. Output which is set when the MONO function starts and reset when the time set TP has elapsed.
TE	OI	**T**ime **E**lapsed. Output for time elapsed when output O is set. When the time set TP has elapsed TE will stop. 1 = 10ms.

Function

A memory is set when the input I is set. The output then goes to -1. When the time set in the timer has elapsed, the memory is cleared and the output O goes to 0.

MONO function, Not Retriggable

If the parameter RTG is set to 0, the MONO function is not retriggable. If a new pulse occurs at the input I before the timer has elapsed, it does not affect the timer. Only when the time set has elapsed and the output O is 0, the MONO function can be restarted by the input value I going from 0 to ≠ 0.

MONO function, Retriggable

If the parameter RTG is set to ≠ 0, the MONO function is retriggable. I.e. the timer starts from 0 each time a new pulse occurs at the input I. If a new pulse occurs at the input I before the timer has elapsed, the timer will restart from zero.

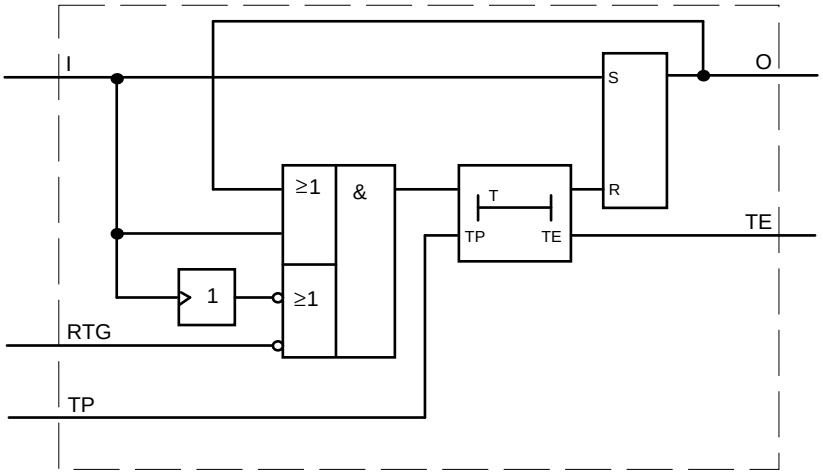


Figure 2. Function diagram

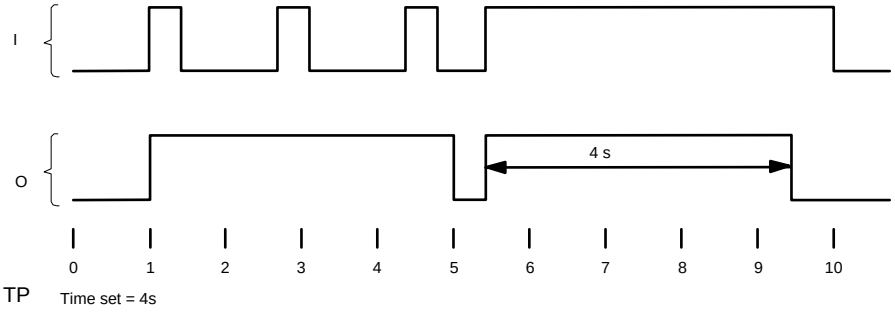


Figure 3. Time diagram MONO function, not retriggable

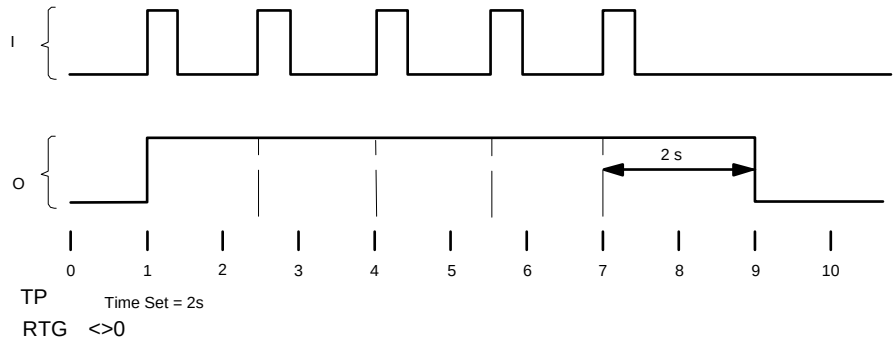


Figure 4. Time diagram MONO function, retriggable

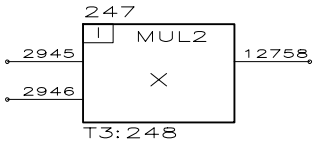
Block numbers and pin addresses *Table 2*

FB number	350	351	352	353	354
RTG	3122	3125	3128	3131	3134
I	3123	3126	3129	3132	3135
TP	3124	3127	3130	3133	3136
O	12870	12872	12874	12876	12878
TE	12871	12873	12875	12877	12879

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Multiplier MUL



Summary

MUL (**MUL**tiplier) is used for multiplication of two integer values.

Call MUL2_247...MUL2_250

Connections *Table 1*

Name	Type	Description
IN1	II	Input for multiplicand.
IN2	II	Input for multiplier
O	OI	Output for product.

Function

The values at the inputs IN1 and IN2 are multiplied and the product is stored to the output O.

Overflow

If the maximum positive or negative value is exceeded, the output is limited to the highest or lowest allowable value for the integer data type.

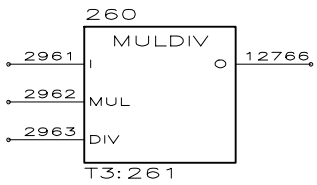
Block numbers and pin addresses *Table 2*

FB number	247	248	249	250
IN1	2945	2947	2949	2951
IN2	2946	2948	2950	2952
O	12758	12759	12760	12761

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Integer Scaling Block — MULDIV



Summary

MULtiplier **DIV**ider block is used to scale integer values by multiplying the input value with a fractional number.

Call MULD_260...MULD_269

Connections *Table 1*

Name	Type	Description
I	II	**I**nput for multiplicand
MUL	II	Input for **MUL**tiplier.
DIV	II	Input for **DIV**isor
O	OI	**O**utput for quotient.

Function

The product of input I and input MUL is divided by the input DIV. The quotient is loaded to the output O. The block uses internally 32 bit accuracy to perform the multiplication.

Overflow

If the maximum positive value is exceeded, the output O is limited to +32767. If the minimum negative value is exceeded, the output O is limited to -32768.

Block numbers and pin addresses *Table 2*

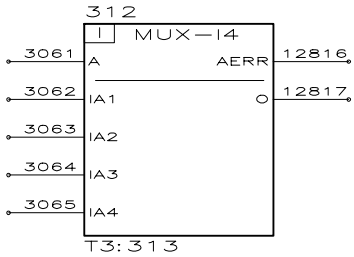
FB number	260	261	262	263	264	265	266	267	268	269
I	2961	2964	2967	2970	2973	2976	2979	2982	2985	2988
MUL	2962	2965	2968	2971	2974	2977	2980	2983	2986	2989
DIV	2963	2966	2969	2972	2975	2978	2981	2984	2987	2990
O	12766	12767	12768	12769	12770	12771	12772	12773	12774	12775

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Multiplexer with Integer Address

MUX-I



Summary

MUX-I (**MU**ltiple**X**er - with Integer address) is used as a selector.

Call MUX4_312...MUX4_313

Connections *Table 1*

Name	Type	Description
A	II	**A**ddress. Input for address value which specifies which input is to be connected to the output O. When A = 0, the output is set to 0.
IA1	II	**I**nput **A**ddress **1**. Input number 1 to the selector.
IA2	II	**I**nput **A**ddress **2**. Input number 2 to the selector.
IA3	II	**I**nput **A**ddress **3**. Input number 3 to the selector.
IA4	II	**I**nput **A**ddress **4**. Input number 4 to the selector.
AERR	OI	**A**ddress **ERR**or. Output which is set to -1 when the address value is greater than the number of inputs, or negative.
O	OI	**O**utput. Data output from the selector.

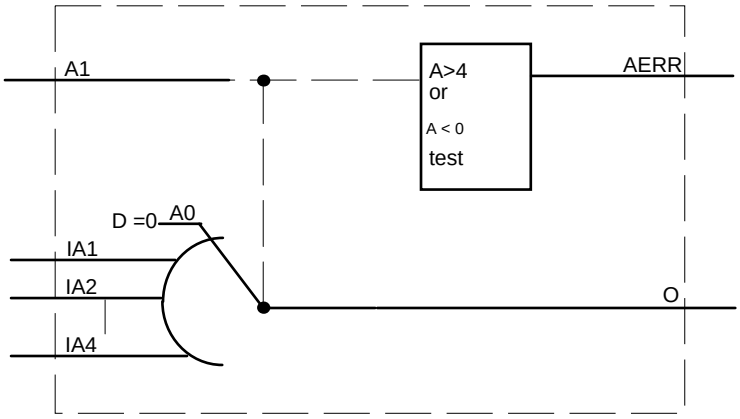
Function

Addressing

The input data value (IA1...IA4) connected to the output O is selected with an address (1...4) in the input A. If the address is 0, the data value 0 is written to the output O.

Supervision

The address A is monitored and if its value is greater than the number of inputs or negative, the error signal output AERR is set. The data value 0 is then written to the output.



Function diagram

Block numbers and pin addresses *Table 2*

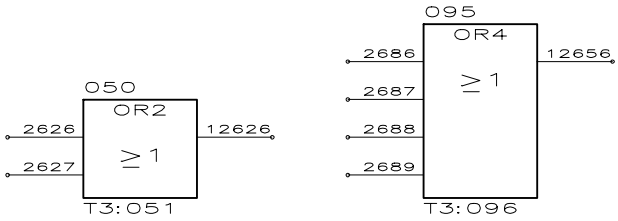
FB number	312	313
A	3061	3066
IA1	3062	3067
IA2	3063	3068
IA3	3064	3069
IA4	3065	3070
AERR	12816	12818
O	12817	12819

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Or Gate

OR



Summary

OR is used to form a logical OR-function with Boolean variables

Call OR2_050...OR2_079, OR4_095...OR4_104

Connections *Table 1*

Name	Type	Description
IIN1	II	Input
IN2	II	Input
IN3	II	Input
IN4	II	Input
O	OI	Output

Function

The output signal is set to -1 if any input IN1..IN4 is ≠ 0. See the truth table below.

Truth table OR2 *Table 2*

IN1	IN2	O
= 0	= 0	0
= 0	≠ 0	-1
≠ 0	= 0	-1
≠ 0	≠ 0	-1

Block numbers and pin addresses *Table 3*

OR2:

FB number	50	51	52	53	54	55	56	57	58	59
IN1	2626	2628	2630	2632	2634	2636	2638	2640	2642	2644
IN2	2627	2629	2631	2633	2635	2637	2639	2641	2643	2645
O	12626	12627	12628	12629	12630	12631	12632	12633	12634	12635

FB number	60	61	62	63	64	65	66	67	68	69
IN1	2646	2648	2650	2652	2654	2656	2658	2660	2662	2664
IN2	2647	2649	2651	2653	2655	2657	2659	2661	2663	2665
O	12636	12637	12638	12639	12640	12641	12642	12643	12644	12645

FB number	70	71	72	73	74	75	76	77	78	79
IN1	2666	2668	2670	2672	2674	2676	2678	2680	2682	2684
IN2	2667	2669	2671	2673	2675	2677	2679	2681	2683	2685
O	12646	12647	12648	12649	12650	12651	12652	12653	12654	12655

OR4:

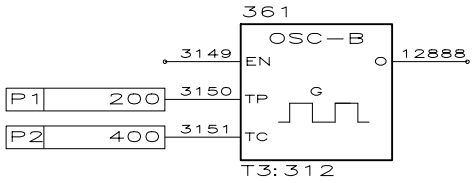
FB number	95	96	97	98	99	100	101	102	103	104
IN1	2686	2690	2694	2698	2703	2707	2711	2715	2719	2723
IN2	2687	2691	2695	2699	2704	2708	2712	2716	2720	2724
IN3	2688	2692	2696	2701	2705	2709	2713	2717	2721	2725
IN4	2689	2693	2697	2702	2706	2710	2714	2718	2722	2726
O	12656	12657	12658	12659	12660	12661	12662	12663	12664	12665

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Oscillator Boolean Variables

OSC-B



Summary

OSC-B (**OSC**illator - **B**oolean variables) with variable frequency and pulse time. OSC-B is used when pulse trains with period from 2 * cycle time up to 650 sec are needed.

Call OSCB_361

Connections *Table 1*

Name	Type	Description
EN	II	**EN**able. Input for starting the oscillator.
TP	PI	**T**ime **P**ulse. Parameter for setting the pulse time. 1 == 10ms.
TC	PI	**T**ime **C**ycle. Parameter for setting the period. The period must be greater than the pulse duration. 1 == 10ms. Max. 650s.
O	OI	**O**utput. Oscillator output.

Function

The oscillator starts when the input EN is set (to ≠ 0). Figure 2. shows an example of an oscillator time diagram.

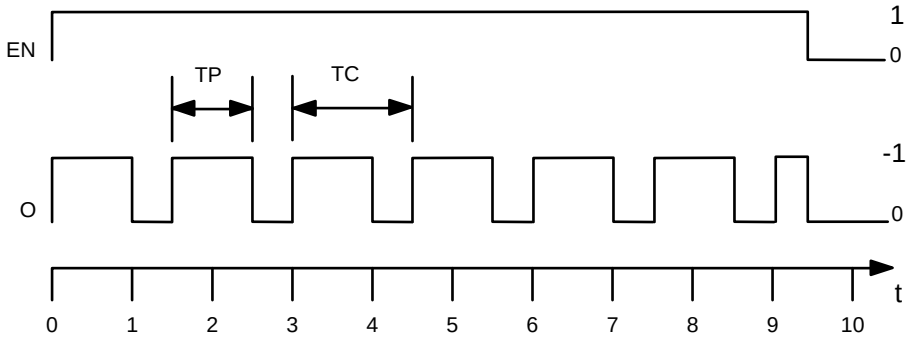


Figure 2. Example of time diagram with TP = 1 s and TC = 1.5 s

Block numbers and pin addresses *Table 2*

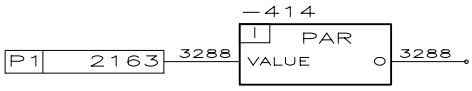
FB number	361
EN	3149
TP	3150
TC	3151
O	12888

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

User Parameter

PAR



Summary

The **PAR** block can be used to present parameters of the user application. These parameter values are stored in the Parameter Flash PROM together with other DCS 500 parameters and they are preserved during the "Power down". DCS 500 documentation tells you how to save the parameters to the Parameter Flash PROM. PAR blocks are not included in execution order list of Task1, Task2 or Task3!

Call PAR_414...PAR_433

Connections *Table 1*

Name	Type	Description
VALUE	II	User defined parameter.
O	OI	**O**utput value of the parameter.

Block numbers and pin addresses *Table 2*

FB number	414	415	416	417	418	419	420	421	422	423
VALUE, O	3288	3289	3290	3291	3292	3293	3294	3295	3296	3297

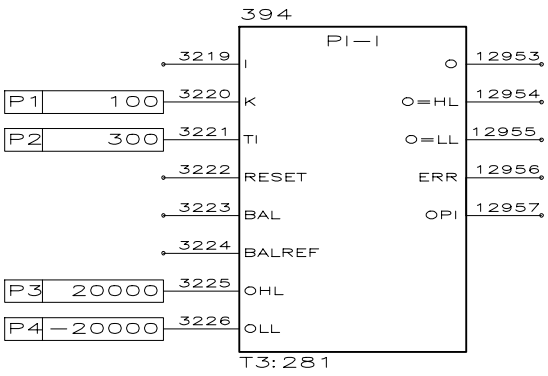
FB number	424	425	426	427	428	429	430	431	432	433
VALUE, O	3298	3299	3301	3302	3303	3304	3305	3306	3307	3308

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

PI-function Integer Values

PI-I



Summary

PI-I (**P**roportional **I**ntegrating regulator with **I**nteger calculation) is used as a PI-regulator for serial compensation in the closed loop control system. Control deviation is the input signal for the controller. The output signal can be limited to limits specified by input parameters. The tracking function forces the output signal to follow an external reference. The transfer from tracking to normal function with PI algorithm is bumpless.

Call PI-I_394

Connections *Table 1*

Name	Type	Description	Values
I	II	**I**nput. Input for deviation value.	-32768..32767
K	PI	Parameter for setting gain.	0..32767(%)
TI	PI	**T**ime **I**ntegration. Parameter for integration time constant.	0..32767(ms)
RESET	II	**RESET** the regulator outputs OPI and O to "0".	
BAL	II	**BAL**ance. Input for activation of tracking.	
BALREF	II	**BAL**ance **REF**erence. Input for reference value when tracking.	-32768..32767
OHL	PI	**O**utput **H**igh **L**imit. Parameter for upper limit value.	-32768..32767
OLL	PI	**O**utput **L**ow **L**imit. Parameter for lower limit value.	-32768..32767
O	OI	**O**utput. Output signal.	-32768..32767
O = HL	OI	**O**utput **= H**igh **L**imit. Output which is set to -1 if the output reaches to the higher limit value	
O = LL	OI	**O**utput **= L**ow **L**imit. Output which is set to -1 if the output reaches to the lower limit value	
ERR	OI	**ERR**or. Output which is set to -1 if OHL is less than OLL.	
OPI	OI	**O**utput of **PI** regulator before limitation.	-32768..32767

Gain and Integration Time Constant calibration

The calibration of the Proportional Gain **K** is in "%". It means that value K = 100 corresponds to a multiplication factor of 1.0 (100%). The calibration of the Integration Time Constant **TI** is directly in milliseconds.

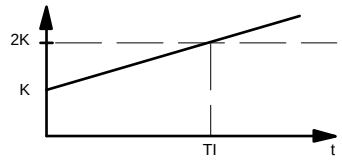


Figure 2. Step response

Transfer Function

P and I parts of the controller are calculated independently.

The PI function can be written in the time plane as:

O(t) = K*[I(t)+≠(1/T1)*I(t)*dt]

The transfer function of the PI controller is:

G(s) = K≠[1+1/(s≠TI)]

This has been implemented in the PI-I block with a recursive algorithm. The basic form of this algorithm is:

1. INT(k)=INT(k-1) + K≠I(k)≠TS/TI
2. O(k)=K ≠I(k) +INT(k)

where: TS is an execution interval of the controller.
INT is the integral component

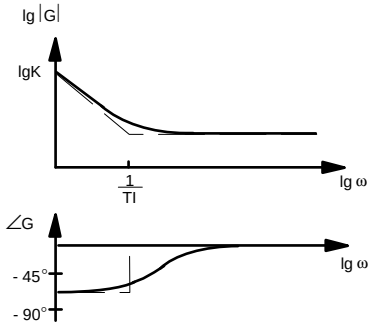


Figure 3. Bode diagram

Clearing the Regulator Output

When the RESET input signal is set to ≠ 0 the output of controller is set to 0. The Integrator part of regulator can be cleared by setting input TI to 0. When TI is set again to a value ≠ 0, the transfer to normal operation is bumpless.

Limitation Function

The OLL and OHL parameters provide the minimum and maximum limit values for the controller output signal. If the actual value exceeds the maximum limit value, then the output "O=HL" is set to -1 and if it falls below the minimum limit value, then the output "O=LL" is set to -1. The block also compares OHL with OLL. If OLL value is greater then OHL, then the ERR output is set to -1. While the error status persists, the output limits "O=HL", "O=LL" retain their previous values.

Tracking

When BAL input is set to ≠ 0, the regulator output "O" is forced to follow the reference value from the BALREF input. If the BALREF signal exceeds the output limits, the output is set to the applicable limit value.

Bumpless transfer between Tracking and Normal Mode

The transfer from **Tracking** mode is bumpless. This is achieved by internal re-calculation of the integration part in the tracking mode according to:

INT(k) = O(k) - K≠I(k)

There is no explicit feature in the regulator to support directly the bumpless transfer to the Tracking mode. However, such function can be easily implemented, by using other function blocks to connect the output "O" of the controller to its BALREF input.

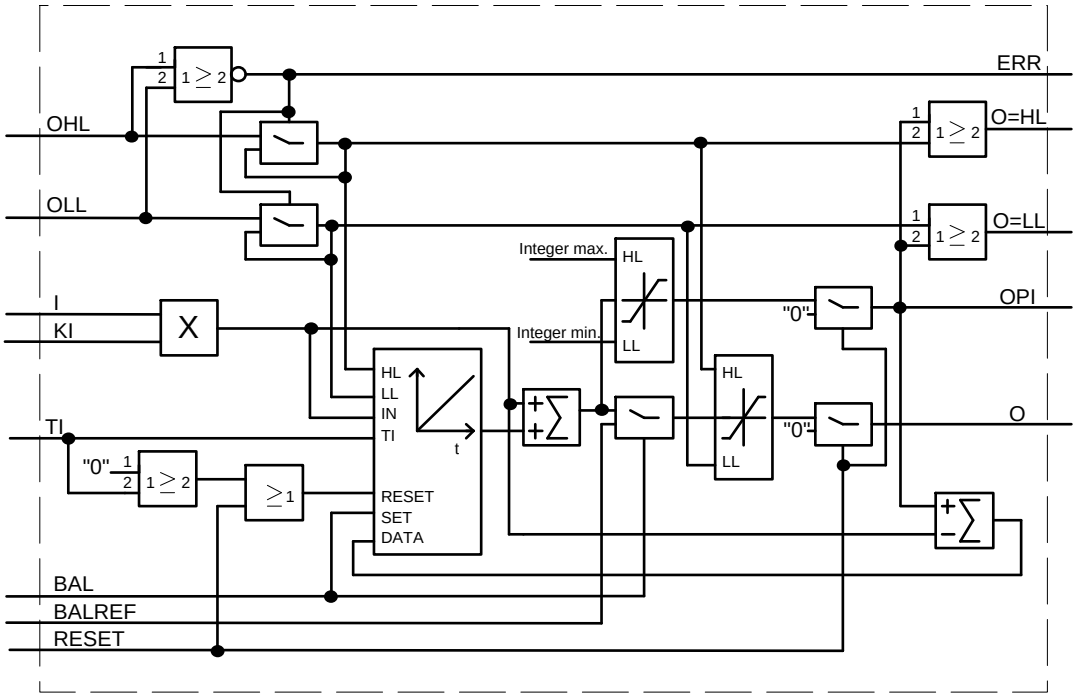


Figure 4. Function diagram

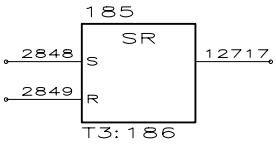
Block numbers and pin addresses *Table 2*

FB number	394
I	3219
K	3220
TI	3221
RESET	3222
BAL	3223
BALREF	3224
OHL	3225
OLL	3226
O	12953
O = HL	12954
O = LL	12955
ERR	12956
OPI	12957

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Memory Block — SR



Summary

The memory block **SR** (**S**et **R**eset memory) is used as a memory for Boolean variables.

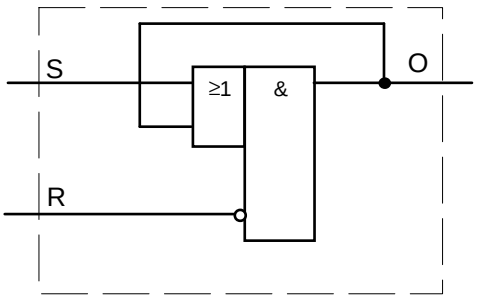
Call SR_185...SR_194

Connections *Table 1*

Name	Type	Description
S	II	**S**et input
R	II	**R**eset input which overrides the set input.
O	OI	**O**utput from the memory block.

Function

The block output is set (to -1) if the set input is set (to ≠ 0) while the reset input is reset (to 0). If the reset input is set (to ≠ 0), the output is unconditionally reset (to 0).



Function diagram

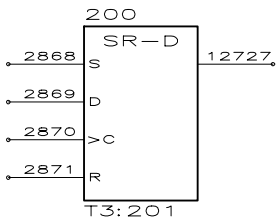
Block numbers and pin addresses *Table 2*

FB number	185	186	187	188	189	190	191	192	193	194
S	2848	2850	2852	2854	2856	2858	2860	2862	2864	2866
R	2849	2851	2853	2855	2857	2859	2861	2863	2865	2867
O	12717	12718	12719	12720	12721	12722	12723	12724	12725	12726

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Memory Block with Data Input — SR-D



Summary

The memory block **SR-D** (**S**et **R**eset memory - **D**ata input) is used as a memory for Boolean variables.

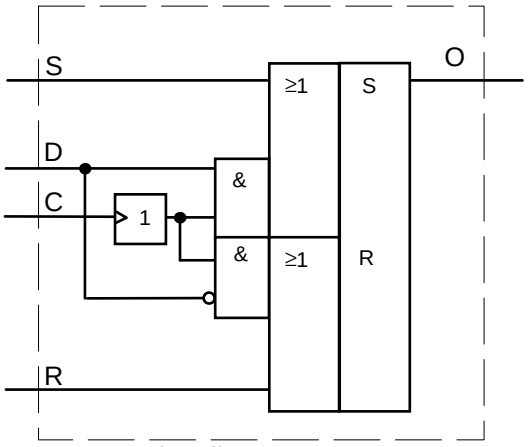
Call SR-D_200...SR-D_204

Connections *Table 1*

Name	Type	Description
S	II	**S**et input
D	II	**D**ata input
C	II	**C**lock. Dynamic input for entry of data from the D input.
R	II	**R**eset input which overrides the set input.
O	OI	Output from the memory block.

Function

If only the S and R inputs are used, SR-D functions as an ordinary SR block. When the input R is reset (to 0) and the input C goes to ≠ 0, the value at the input D is stored at the output O. If D is ≠ 0 , the output is set to -1. Otherwise output is 0. When the input R is set (to ≠ 0), the output O is unconditionally reset, i.e. R overrides the other inputs.



Function diagram

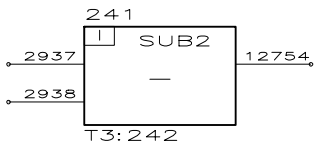
Block numbers and pin addresses *Table 2*

FB number	200	201	202	203	204
S	2868	2872	2876	2880	2884
D	2869	2873	2877	2881	2885
C	2870	2874	2878	2882	2886
R	2871	2875	2879	2883	2887
O	12727	12728	12729	12730	12731

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Subtractor SUB



Summary

SUB is used for subtraction of two integer values.

Call SUB2_241...SUB2_244

Connections *Table 1*

Name	Type	Description
IN1	II	Input for minuend.
IN2	II	Input for subtrahend.
O	OI	Output for difference.

Function

The value at input IN2 is subtracted from value at input IN1 and the result is stored at output O.

Overflow

If the maximum positive or negative value is exceeded, the output is limited to the highest or lowest allowable value for the integer data type.

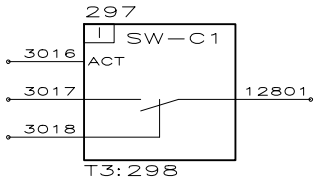
Block numbers and pin addresses *Table 2*

FB number	241	242	243	244
IN1	2937	2939	2941	2943
IN2	2938	2940	2942	2944
O	12754	12755	12756	12757

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Switch with Changeover SW-C



Summary

SW-C1 (**SW**itch - **C**hangeover) is used as a connection block for data and has one channel with closing function.

Call SWC1_297...SWC1_311

Connections *Table 1*

Name	Type	Description
ACT	II	**ACT**ivate. Input for activation of the switch. When the input is set to ≠ 0 the switch is activated.
IN1	II	Input to channel 1 which is connected to the output O when the switch is activated.
IN2	II	Input to channel 1 which is connected to the output O when the switch is not activated.
O	OI	Output.

Function

When the control input ACT is 0, the data from the input IN2 is connected to the output O. When ACT is set (to ≠ 0) , data comes from the input IN1.

Block numbers and pin addresses *Table 2*

FB number	297	298	299	300	301	302	303	304	305	306
ACT	3016	3019	3022	3025	3028	3031	3034	3037	3040	3043
IN1	3017	3020	3023	3026	3029	3032	3035	3038	3041	3044
IN2	3018	3021	3024	3027	3030	3033	3036	3039	3042	3045
O	12801	12802	12803	12804	12805	12806	12807	12808	12809	12810

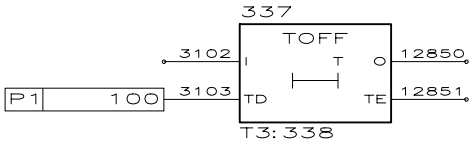
FB number	307	308	309	310	311
ACT	3046	3049	3052	3055	3058
IN1	3047	3050	3053	3056	3059
IN2	3048	3051	3054	3057	3060
O	12811	12812	12813	12814	12815

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Time Delay Off

TOFF



Summary

Time delay off **TOFF** (**T**imer **OFF**-delay) is used for Boolean off state delay.

Call TOFF_337...TOFF_346

Connections *Table 1*

Name	Type	Description
I	II	**I**nput for start of time delay.
TD	PI	**T**ime **D**elay. Parameter for pre-set time. 1 == 10ms. Max. 650s.
O	OI	**O**utput. Output which is reset when the time has elapsed.
TE	OI	**T**ime **E**lapsed. Output which specifies how long I has been reset. When the pre-set time (TD) has elapsed, TE stops. 1 == 10ms.

Function

The input signal in the input I is copied with delay to the output O when the input signal changes from≠ 0 to 0 in accordance with the time pulse diagram in figure 2. The output signal returns to -1, when the input signal changes from 0 to ≠ 0.

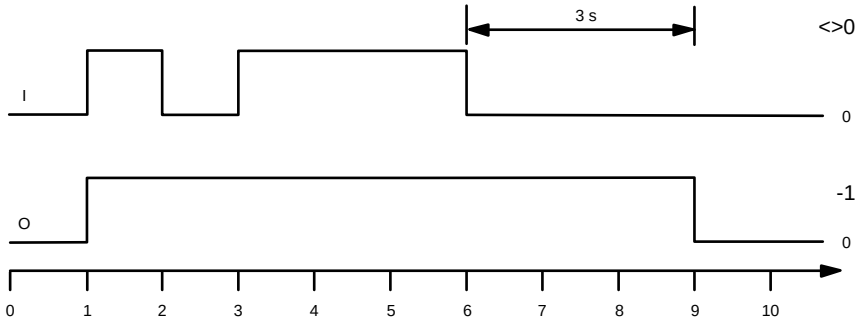


Figure 2. Example of time diagram for TOFF with preset time 3s.

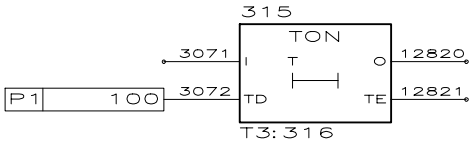
Block numbers and pin addresses *Table 2*

FB number	337	338	339	340	341	342	343	344	345	346
I	3102	3104	3106	3108	3110	3112	3114	3116	3118	3120
TD	3103	3105	3107	3109	3111	3113	3115	3117	3119	3121
O	12850	12852	12854	12856	12858	12860	12862	12864	12866	12868
TE	12851	12853	12855	12857	12859	12861	12863	12865	12867	12869

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Time Delay On — TON



Summary

Time delay on **TON** (**T**imer **ON**-delay) is used for Boolean on state delay.

Call TON

Connections *Table 1*

Name	Type	Description
I	II	**I**nput for start of time delay.
TD	PI	**T**ime **D**elay. Parameter for pre-set time. 1 == 10ms. Max. 650s.
O	OI	**O**utput. Output which is set when the time has elapsed.
TE	OI	**T**ime **E**lapsed. Output which specifies how long I has been set. When the pre-set time (TD) has elapsed, TE stops. 1 == 10ms.

Function

The input signal connected to input I is copied with delay to the output O, when the input signal changes from 0 to ≠ 0 in accordance with the time pulse diagram in figure 2. The output signal returns to 0, when the input signal changes from ≠ 0 to 0.

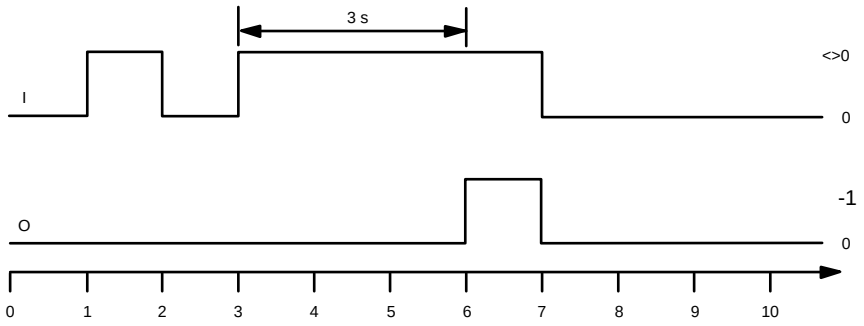


Figure 2. Example of time diagram for TON with preset time 3s.

Block numbers and pin addresses *Table 2*

FB number	315	316	317	318	319	320	321	322	323	324
I	3071	3073	3075	3077	3079	3081	3083	3085	3087	3089
TD	3072	3074	3076	3078	3080	3082	3084	3086	3088	3090
O	12820	12822	12824	12826	12828	12830	12832	12834	12836	12838
TE	12821	12823	12825	12827	12829	12831	12833	12835	12837	12839

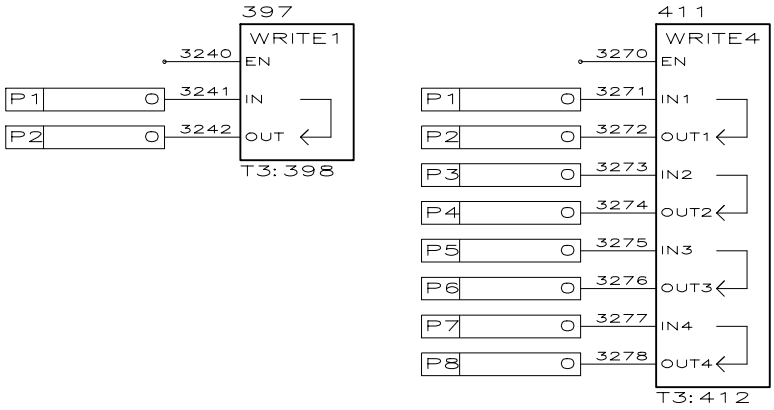
FB number	325	326	327	328	329
I	3091	3093	3095	3097	3099
TD	3092	3094	3096	3098	3101
O	12840	12842	12844	12846	12848
TE	12841	12843	12845	12847	12849

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Data Transfer

WRITE



Summary

WRITE function block performs data transfer from source to destination.

Call WRI1_397...WRI1_406; WRI4_411, WRI4_412

Connections *Table 1*

Name	Type	Description
EN	II	Enable data transfer.
IN1	PI	Source address 1
OUT1	PI	Destination address 1
IN2	PI	Source address 2
OUT2	PI	Destination address 2
IN3	PI	Source address 3
OUT3	PI	Destination address 3
IN4	PI	Source address 4
OUT4	PI	Destination address 4

Function

If EN ≠ 0 the value of a parameter or a signal indicated by the source address is written to the parameter or signal indicated by the destination address. When EN = 0, no data transfer occurs.

Block numbers and pin addresses *Table 2*

WRITE1:

FB number	397	398	399	400	401	402	403	404	405	406
EN	3240	3243	3246	3249	3252	3255	3258	3261	3264	3267
IN	3241	3244	3247	3250	3253	3256	3259	3262	3265	3268
OUT	3242	3245	3248	3251	3254	3257	3260	3263	3266	3269

WRITE4:

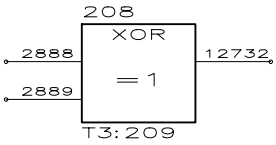
FB number	411	412
EN	3270	3279
IN1	3271	3280
OUT1	3272	3281
IN2	3273	3282
OUT2	3274	3283
IN3	3275	3284
OUT3	3276	3285
IN4	3277	3286
OUT4	3278	3287

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language),*

Exclusive Or Gate

XOR



Summary

XOR (e**X**clusive **OR**) is used to generate an exclusive OR function with Boolean variables.

Call XOR

Connections *Table 1*

Name	Type	Description
IN1	II	Input.
IN2	II	Input.
O	OI	Output.

Function

The output signal from the XOR block is -1 if the one of the input signals is 0 and the other is≠ 0, see table 2 below.

Truth Table XOR *Table 2*

IN1	IN2	O
= 0	= 0	0
= 0	≠ 0	-1
≠ 0	= 0	-1
≠ 0	≠ 0	0

Block numbers and pin addresses *Table 3*

FB number	208	209	210	211	212	213	214	215	216	217
IN1	2888	2890	2892	2894	2896	2898	2901	2903	2905	2907
IN2	2889	2891	2893	2895	2897	2899	2902	2904	2906	2908
O	12732	12733	12734	12735	12736	12737	12738	12739	12740	12741

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language*

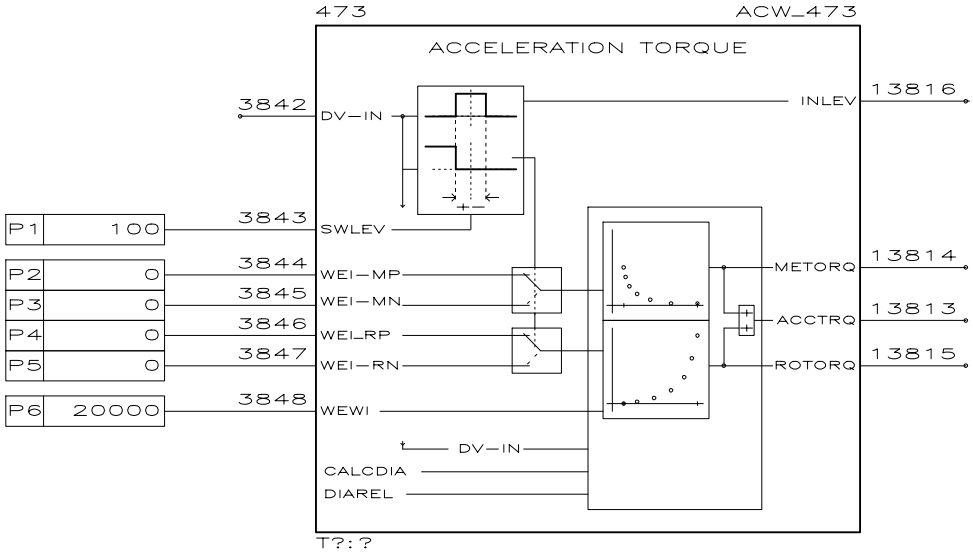
Function Blocks valid for Software Version:
21.121

Function Block	New	Modified
ACW_473	X	
DIAC_468	X	
DT_480	X	
DT_481	X	
LSW_470	X	
DIAREL	X	
PARW_501	X	
PARW_502	X	
PARW_503	X	
PARW_504	X	
PARW_505	X	
PARW_506	X	
PARW_507	X	
PARW_508	X	
PARW_509	X	
PARW_510	X	
SC_472	X	
SSL_471	X	
TERE_474	X	
TOTE_475	X	
TRII_482	X	
TRII_483	X	
TRII_484	X	
TRII_485	X	
VERE_469	X	
WILO_476	X	

❶ Modification of Software
❷ Modification of Diagram
❸ Modification of Connections

Acceleration weighting

ACW



Summary

ACW The function block **AC**celeration **W**eighting is used to calculate the torque of inertia. The input is the acceleration value. The weighting depends of the nominal current of the unit.

Call ACW_473

Connections

Table 1

Name	Type	Description
DV-IN	II	Input of the acceleration value
SWLEV	PI	Value for the switching hysteresis
WEI-MP	PI	Adjusting value : weighting of the moment of the mechanical inertia (positive acceleration)
WEI-MN	PI	Adjusting value : weighting of the moment of the mechanical inertia (negative acceleration)
WEI-RP	PI	Adjusting value : weighting of the moment of the roll inertia (positive acceleration)
WEI-RN	PI	Adjusting value : weighting of the moment of the roll inertia (negative acceleration)
WEWI	PI	Value of material web width
CALCDIA		Calculated diameter (output of DIAC_468)
DIAREL		Diameter relation (output of PARW_500)
INLEV	OI	Output signal of the switching hysteresis
METORQ	OI	Moment of the mechanical inertia torque
ROTORQ	OI	Moment of the roll inertia torque
ACCTRQ	OI	Total of the inertia torque

Function

The acceleration value is to create with a maximal value of 20000 for the minimum accelerating and decelerating time. It is better to make this value outside of the DCS, but it is also possible to do this with the function block DERIVATION WITH LOW FILTER (DT) and a MULTIPLIER (MUL).

Two binary signals are created with the acceleration value. The parameter P1 (SWLEV) is used to set the comparator level.

- The output INLEV get high (-1), if the amount of the acceleration level is lower than the value of P1.

- The second signal descripted the polarity of the acceleration signal (lower than minus P1) and is used to switch the adjusting parameters.

Parameters P2 (WEI-MP) and P3 (WEI-MN) serve for presetting the acceleration share of the complete mechanical equipment of the winder (incl. core). The parameter P2 is used for defining the positive acceleration share, parameter P3 for defining the negative acceleration share (deceleration). The thus adjusted current is attained at minimum diameter.

$$\mathbf{METORQ} = \mathbf{DV_{IN}} * \frac{\mathbf{(P2\ or\ P3)}}{\mathbf{5000}} * \frac{\mathbf{DIAREL}}{\mathbf{CALCDIA}}$$

Parameters P4 (WEI-PP) and P5 (WEI-PN) serve for presetting the acceleration share of the material roll. Parameter P4 is used for defining the positive acceleration share, parameter P5 for defining the negative acceleration share (deceleration).

Parameter P6 (WEWI) serves for presetting the web width as percentage value, which is taken into account for acceleration of the material roll. The value adjusted for acceleration is attained at maximum diameter and maximum web width (20000).

$$\mathbf{ROTORQ} = \mathbf{DV_{IN}} * \frac{\mathbf{(P4\ or\ P5)}}{\mathbf{5000}} * \frac{\mathbf{WEWI}}{\mathbf{20000}} * \frac{\mathbf{DIAmax}}{\mathbf{CALCDIA}} * \frac{\mathbf{CALCDIA_z^4 - DIAREL_z^4}}{\mathbf{DIAmax_z^4 - DIAREL_z^4}}$$

where

$$\mathbf{DIA_{max} = 4000}$$

$$\mathbf{DIA_{max}{}_z^4 = (4000^2 / 347)^2}$$

$$\mathbf{CALCDIA_z^4} = \mathbf{(CALCDIA^2 / 347)^2}$$

$$\mathbf{DIAREL_z^4} = \mathbf{(DIAREL^2 / 347)^2}$$

The sum of both torque values is given to the output ACCTRQ.

Block numbers and pin addresses *Table 2*

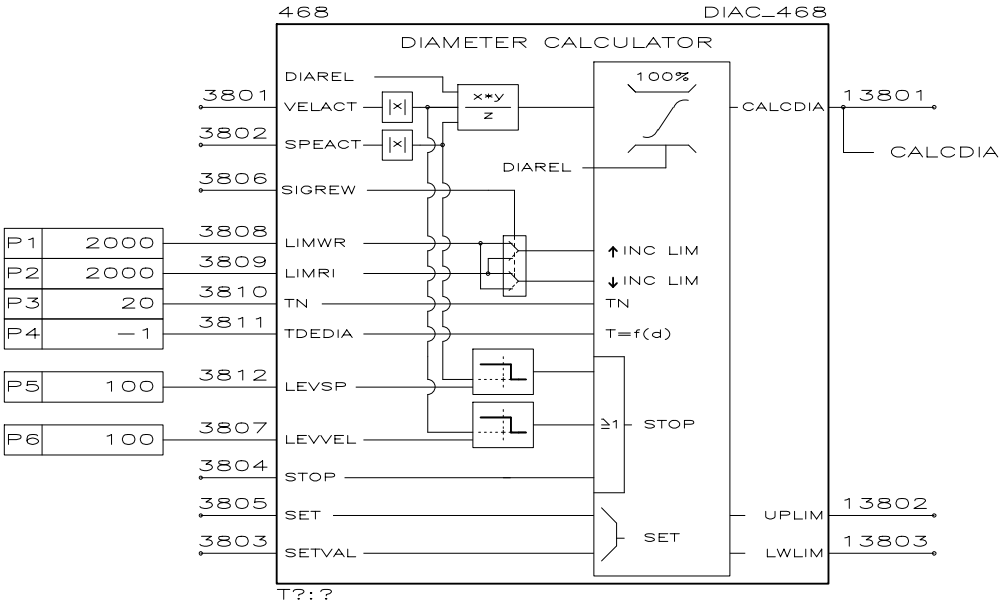
FB number	473		Scaling factor	Range of values max.	min.	Default	Unit
DV-IN	3842			-20000	+20000		
SWLEV	3843			0	+20000	0	
WEI-MP	3844		0,1	0	+1000	0	%
WEI-MN	3845		0,1	0	+1000	0	%
WEI-RP	3846		0,1	0	+1000	0	%
WEI-RN	3847		0,1	0	+1000	0	%
WEWI	3848			0	+20000	20000	
ACCTRQ	13813		TORQ	-4000	+4000		
METORQ	13814		TORQ	-4000	+4000		
ROTORQ	13815		TORQ	-4000	+4000		
INLEV	13816			0	-1		

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Diameter calculator

DIAC



Summary

DIAC The **DIA**meter **C**alculator is used to integrate the diameter from two inputs. The output is the relation of the minimal an maximal actual diameter.

Call DIAC_468

Connections *Table 1*

Name	Type	Description
VELACT	II	Input actual velocity reference
SPEACT	II	Input actual motor speed
SIGREW	II	Signal rewinder changing lower and upper limits
LIMWR	PI	Limit of the wrong calculation direction
LIMRI	PI	Limit of the right calculation direction
TN	PI	Integrating time
TDEDIA	PI	Integrating time is depended of the diameter (T = f(d))
LEVSP	PI	Value of the motor speed to release the calculation
LEVVEL	PI	Value of the velocity to release the calculation
STOP	II	Signal to stop the calculation
SET	II	Signal to set the calculation
SETVAL	II	Value for the setting function
CALCDIA	OI	Output of the calculated diameter
UPLIM	OI	Signal : Output is at the upper limit (4000 dec.)
LWLIM	OI	Signal : Output is at the lower limit (DIAREL)

Function

The diameter is calculated on the basis of the weighted velocity value, of the speed feedback value and the diameter relation DIAREL (PARW_500). To avoid jumps, the calculated diameter signal is passed through integrator. The time constant is adjusted by using parameter P3 (TN). Parameter P4 (TEDEDIA) serves for selecting whether this time constant also is to be weighted in dependence on the diameter (advisable in the case of a higher diameter relation).

The differential signal (diameter reference/feedback difference) is additionally provided with two direction-dependent limit values. The value assigned to parameter P1 (LIMWR) effects - irrespective of upwinder or unwinder operation - limitation of the wrong direction of calculation. The proper direction of calculation is limited with the aid of parameter P2 (LIMRI).

Example : The upwinder’s increase (increase in diameter) and the unwinder’s decrease (decrease in diameter) are limited through parameter P2.

Upon stopping of the diameter calculator, the currently present value is frozen and can only be overwritten via command "Diameter setting" (SET) or by switching off the electronics section.

Any of the following functions suffices to stop the calculator :

- Input STOP (= -1)
- Motor speed lower than the threshold adjusted with parameter [159]
- Velocity reference lower than the threshold adjusted with parameter [154]

Block numbers and pin addresses *Table 2*

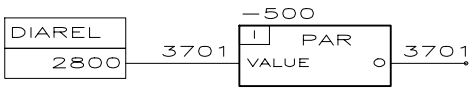
FB number	**468**		Scaling factor	Range of values max.	min.	Default	Unit
VELACT	3801			-20000	20000		
SPEACT	3802			-20000	20000		
SETVAL	3803			100	4000		
STOP	3804		BI	-1	0		
SET	3805		BI	-1	0		
SIGREW	3806		BI	-1	0		
LEVVEL	3807			0	20000	100	
LIMWR	3808			0	20000	2000	
LIMRI	3809			0	20000	2000	
TN	3810			1	20000	5	
TDEDIA	3811			-1	0	-1	
LEVSP	3812			0	20000	100	
CALCDIA	13801			100	4000	2800	
UPLIM	13802			-1	0		
LWLIM	13803			-1	0		

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Diameter Relation

DIAREL



Summary

The block **DIAREL** is fixed for the using as the **DIA**meter **REL**ation. This parameter is used by several other winder blocks like DIAC_468 ... .

This block is not included in execution order list of Task1, Task2 or Task3

Call DIAREL

Connections *Table 1*

Name	Type	Description
VALUE	II	User defined parameter
O	OI	Output value of the parameter

Function

The maximal diameter has the internal value of 4000. The minimum and the maximum diameter of the application is needed to calculate this value :

$$\textbf{DIAREL} = \textbf{4000} * \frac{\textbf{min. diameter}}{\textbf{max. diameter}}$$

Example : With a minimum diameter of 100 mm and a maximum diameter of 1000 mm the value of 400 has to be set into this parameter.

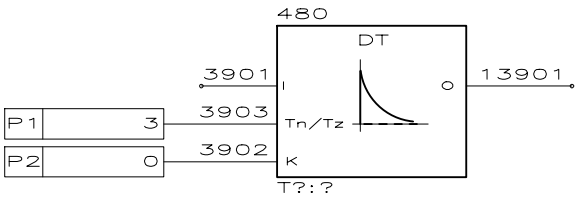
Block numbers and pin addresses *Table 2*

FB number	500		Scaling factor	Range of values max.	min.	Default	Unit
DIAREL	3701			100	4000	2800	

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number Comment text (several languages), General text (one language)*

Derivation with low filter **DT**



Summary

The **DT** block is used as a derivation block with a low pass filter.

Call DT_480, DT_481

Connections *Table 1*

Name	Type	Description
I	II	Input for analogue signals
Tn/Tz	PI	Parameter for filter
K	PI	Parameter for amplification
O	OI	Output of the derivation

Function

The input of the last task will be subtracted from the actual input. This difference is multiplicated with the constant K (P2). The result passed through the low pass filter with the time P1.

Block numbers and pin addresses Table 2

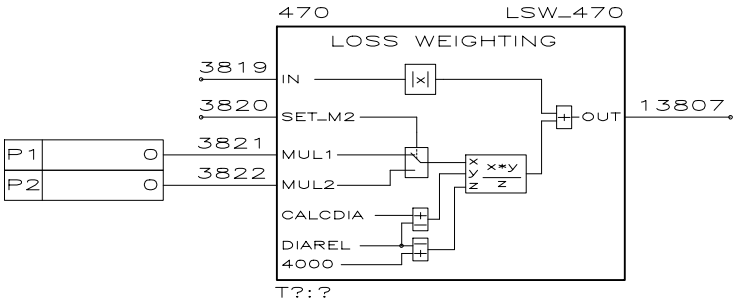
FB number	480	481		Scaling factor	Range of values max.	min.	Default	Unit
I	3901	3904			-20000	+20000		
K	3902	3905			0	20000	0	
Tn/Tz	3903	3906			0	20000	0	
O	13901	13902			-20000	+20000		

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Loss weighting

LSW



Summary

The current of the loss depending of the diameter are calculated and are added up with function block **L**os**S** **W**eighting.

Call LSW_470

Connections *Table 1*

Name	Type	Description
IN	II	Value of the constant and the current depending of the speed
SET_M2	II	Input switching MUL1 and MUL2
MUL1	PI	Weighting with SET_M2 = 0
MUL2	PI	Weighting with SET_M2 = -1
CALCDIA		Calculated diameter (fixed parameter)
DIAREL		Diameter relation (fixed parameter)
4000		Maximally diameter (Constant 4000)
OUT	OI	Sum of the current value

Function

The constant and speed-dependent loss values can to give to the input IN.

The diameter-dependent loss value is entered by using parameter P1 (MUL1) or P2 (MUL2). The preset value corresponds with the unit current at maximum diameter.

Block numbers and pin addresses *Table 2*

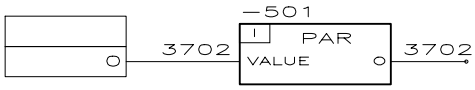
FB number	**470**		Scaling factor	Range of values max.	min.	Default	Unit
IN	3819		TORQ	0	4000		
SET_M2	3820		BI	-1	0		
MUL1	3821			-32768	32767	0	
MUL2	3822			-32768	32767	0	
OUT	13807			0	4000		

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Parameter of winder

PARW



Summary

The **PARW** block can be used to present parameters of the winder application.

PARW blocks are not included in execution order list of Task1, Task2 or Task3.

Call PARW_501 ... PARW_510

Connections *Table 1*

Name	Type	Description
VALUE	II	User defined parameter
O	OI	Output value of the parameter

Block numbers and pin addresses *Table 2*

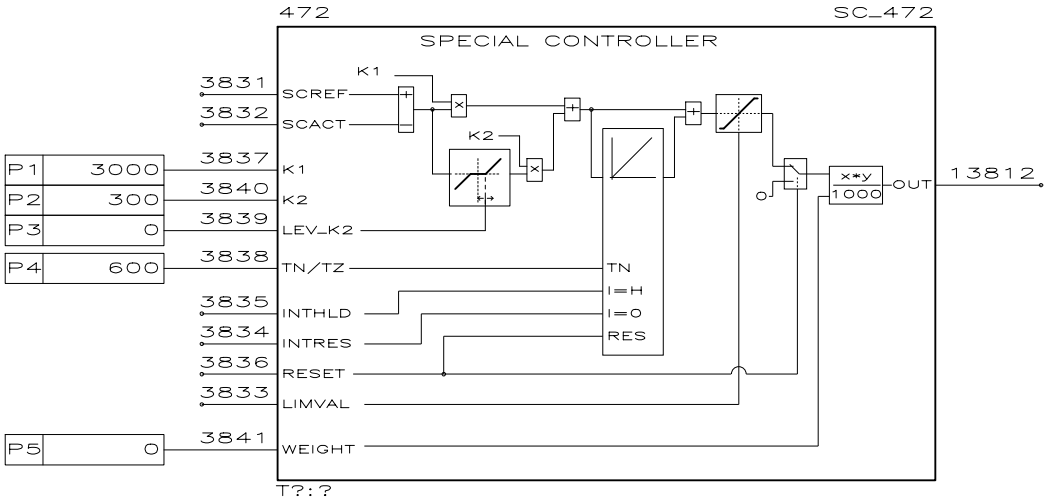
FB number	501	502	503	504	505		Scaling factor	Range of values max.	min.	Default	Unit
VALUE, O	3702	3703	3704	3705	3706			-32768	32767	0	

FB number	5016	507	508	509	510		Scaling factor	Range of values max.	min.	Default	Unit
VALUE, O	3707	3708	3709	3710	3711			-32768	32767	0	

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number, Comment text (several languages), General text (one language)*

Special controller SC



Summary

The **S**pecial **C**ontroller is used as a PI-regulator with a second P for serial compensation in the closed loop control system. The output is limited with an external value and is weighted with a parameter.

Call SC_472

Connections *Table 1*

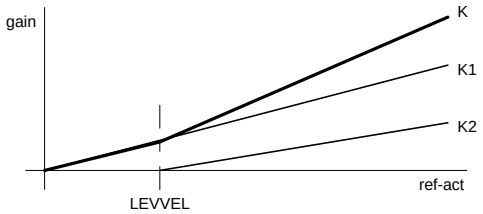
Name	Type	Description
SCREF	II	Reference value
SCACT	II	Actually value
K1	PI	Parameter 1 for setting the gain
K2	PI	Parameter 2 for setting the gain
LEV-K2	PI	Level to start with the second gain (K2)
TN/TZ	PI	Integration time
INTHLD	II	Hold the integration
INTRES	II	Reset the integration
RESET	II	Reset the controller
LIMVAL	II	Value for the limitation
WEIGHT	PI	Value for the weighting
OUT	OI	Output

Function

This controller is equipped with a PI controller and a second gain function depending on the difference between reference and feedback value.

The gain is preset by using parameter P1 (K1) and the reset time by using parameter P4 (TN/TZ).

If the feedback/reference deviation lies above the preset difference P3 (LEV_K2), a second gain function P2 (K2) becomes effective. This ensures that the control system's response is less forceful, if the deviation value is within the range of its desired value, and more forceful, if this range is exceeded.



The value of the integration is hold with the input INTHLD and is set to zero with INTRES. The output of the controller is set to zero, if the signal RESET is active.

The input LIMVAL provide the minimum and maximum limit values.

The controller output is weighted (0...100%) by using parameter P5 (WEIGHT).

Block numbers and pin addresses *Table 2*

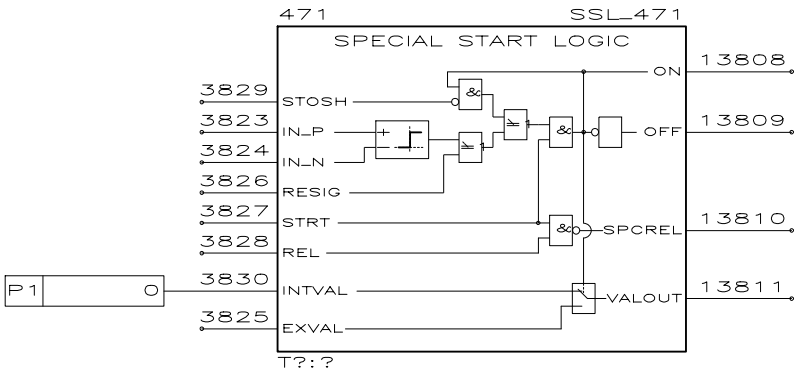
FB number	**472**		Scaling factor	Range of values max.	min.	Default	Unit
SCREF	3831			-20000	20000		
SCACT	3832			-20000	20000		
LIMVAL	3833			-1	0		
INTRES	3834		BI	-1	0		
INTHLD	3835		BI	-1	0		
RESET	3836		BI	-1	0		
K1	3837			0	32767	3000	
TN/TZ	3838			0	32767	600	
LEV_K2	3839			0	32767	100	
K2	3840			0	32767	300	
WEIGHT	3841		0,1	0	2000	0	%
OUT	13812			-20000	20000		

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Special start logic

SSL



Summary

This block **S**pecial **S**tart **L**ogic is used to create the control signals and the limit for die SC (Special Controller) depending of the actually value.

Call SSL_471

Connections *Table 1*

Name	Type	Description
STOSH	II	Reset of the self-holding logic
IN_P	II	Positive input of the comparator (actual value)
IN_N	II	Negative input of the comparator (parameter)
RESIG	II	Reset the start logic
STRT	II	Logic active
REL	II	Release
INTVAL	PI	Internal value for the limit
EXVAL	II	External value for the limit
ON	OI	Start logic on
OFF	OI	Start logic off
SPCREL	OI	Release for the controller
VALOUT	OI	Value for limit

Function

The description starts with the start situation, that all logical inputs are zero. The input IN_P is connected to the actually value, the input IN_N to an external value (parameter or constant). The parameter P1 is set to the start limit and the input EXVAL is set to the max. limit. in this case the start logic is ON (output).

The inputs REL and STRT (set to -1) set the output SPCREL (release signal for the controller). The start logic changed to off, when the input IN_P increased higher than the input IN_N **or** when the signal RESIG is set. The self-holding is active and can to be reset with the input STOSH.

The start logic is ON again, when the signal STRT is set to zero.

Block numbers and pin addresses *Table 2*

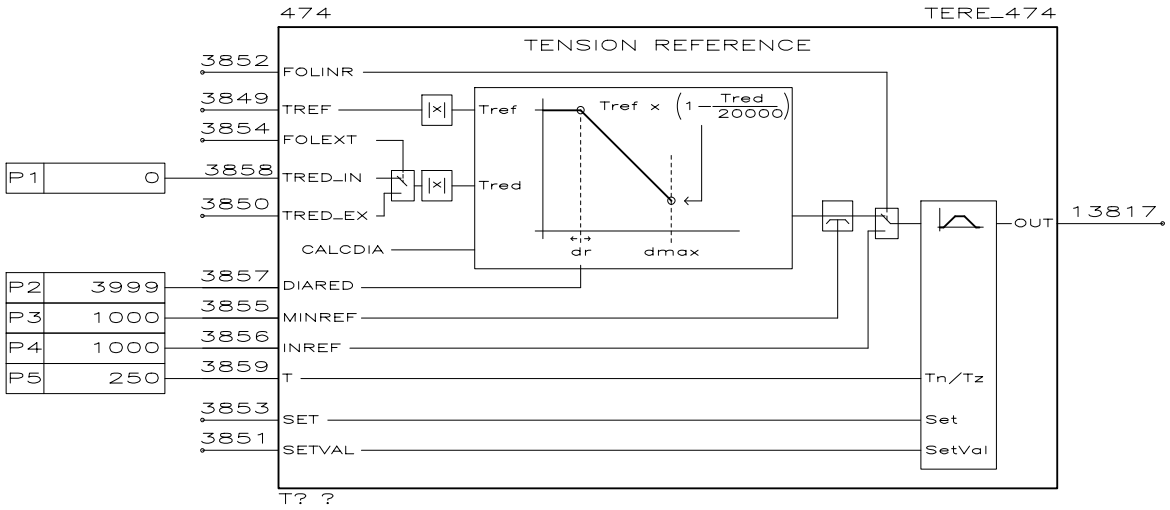
FB number	**471**		Scaling factor	Range of values max.	min.	Default	Unit
IN_P	3823			-20000	20000		
IN_N	3824			-20000	20000		
EXVAL	3825			0	20000		
RESIG	3826		BI	-1	0		
STRT	3827		BI	-1	0		
REL	3828		BI	-1	0		
STOSH	3829		BI	-1	0		
INTVAL	3830			0	20000	0	
ON	13808			-1	0		
OFF	13809			-1	0		
SPCREL	13810			-1	0		
VALOUT	13811			0	20000		

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Tension reference

TERE



Summary

The block **TE**nsion **RE**ference is used for the tension control with and without an actual tension value. This block consists on internal tension reference, tension reduction and tension ramp.

Call TERE_474

Connections

Table 1

Name	Type	Description
TREF	II	External tension reference
FOLEXT	II	Signal setting the external reduction value
TRED_IN	PI	Internal reduction value
TRED_EX	II	External reduction value
CALCDIA	-	Actual diameter
DIARED	PI	Value of diameter starting the reduction
MINREF	PI	Value of the minimum tension
FOLINR	II	Signal setting the internal tension reference
INREF	PI	Internal tension reference
T	PI	Ramp time
SET	II	Signal setting the ramp on SETVAL
SETVAL	II	External setting value
OUT	OI	Output

Function

The input reference value (TREF) is to connected to a tension reference. The tension of 100% is the internal value of 20000.

The parameter P1 (TRED_IN) or the input TRED_EX, chosen with FOLEXT, is used to reduce the tension depends of increasing diameter. The reduction begins with the diameter higher than P2 (DIARED)

At the maximum diameter (4000 dec.) the tension reference is :

$$\mathbf{OUT} = \mathbf{TREF} * \left(1 - \frac{\mathbf{TRED\_IN\ or\ TRED\_EX}}{\mathbf{20000}} \right)$$

Tension reference is limited at the lower value by the parameter P3 (MINREF).

Setting the signal FOLINR the tension value is from the parameter P4 (INREF).

The selected value is passed through the integrator. Time is to set with the parameter P5 (T).

The signal SET is used to set the output to the value of the input SETVAL.

Block numbers and pin addresses *Table 2*

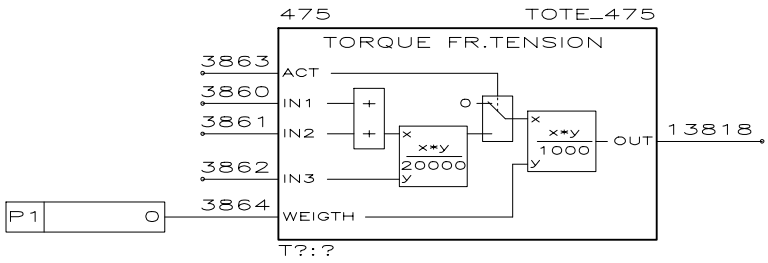
FB number	**474**		Scaling factor	Range of values max.	min.	Default	Unit
TREF	3849			0	20000		
TRED_EX	3850			0	20000		
SETVAL	3851			0	20000		
FOLINR	3852			-1	0		
SET	3853			-1	0		
FOLEXT	3854			-1	0		
MINREF	3855			0	20000	1000	
INREF	3856			0	20000	1000	
DIARED	3857			0	3999	3999	
TRED_IN	3858			0	20000	0	
T	3859			0	32767	250	
OUT	13817			0	20000		

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Torque from tension

TOTE



Summary

The block **TO**rque **TE**nsion is used to transfer the tension to torque and to weight the output.

Call TOTE_475

Connections *Table 1*

Name	Type	Description
ACT	II	Signal to activate the output
IN1	II	Input for the adjusted tension reference
IN2	II	Input for the tension controller
IN3	II	Multiplier for calculation
WEIGHT	PI	Value weighting the output
OUT	OI	Output

Function

This block is to activate the tension torque. The inputs IN1 and IN2 (max. 20000) are calculated with the input IN3, which is connected to the actual diameter CALCDIA. This means with the maximum input, that the output reached the maximum at the maximum diameter. The level of 20000 (reference) is also set to the level of 4000 (current).

The needed torque getting the tension is to be set with the parameter P1.

Block numbers and pin addresses *Table 3*

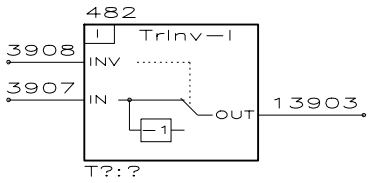
FB number	**475**		Scaling factor	Range of values max.	min.	Default	Unit
IN1	3860			0	20000		
IN2	3861			-20000	20000		
IN3	3862			0	4000		
ACT	3863		BI	-1	0		
WEIGHT	3864		0,1	0	1000	0	%
OUT	13818			-4000	4000		

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Transmit Invert Integer

TRII



Summary

TRII is used to transfer and to invert the integer values

Call TRII_482 ... TRII_485

Connections *Table 1*

Name	Type	Description
INV	II	Input for activation of the inversion
IN	II	Input
OUT	OI	Output is inverted when the switch is activated

Function

When the control input INV is 0, the output has the same value as the input.

When INV is set (≠ 0), data is inverted.

Block numbers and pin addresses *Table 2*

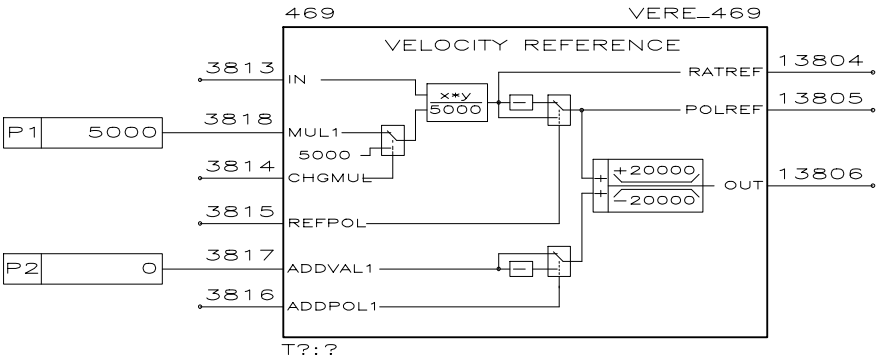
FB number	482	483	484	485
INV	3908	3910	3912	3914
IN	3907	3909	3911	3913
OUT	13903	13904	13905	13906

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Velocity reference

VERE



Summary

The block **VE**locity **RE**ference is used to adjust the value and set the polarity.

Call VERE_469

Connections *Table 1*

Name	Type	Description
IN	II	Reference input
MUL1	PI	Multiplier from parameter
CGHMUL	II	Signal to change multiplier from the internal value (5000)
REFPOL	II	Set polarity
ADDVAL1	PI	Value to add to the reference
ADDPOL1	II	Signal to set the polarity of the added value
RATREF	OI	Adjusted reference (rated)
POLREF	OI	Adjusted reference with the set polarity
OUT	OI	Output of this block

Function

The input IN get the main external velocity reference. For some applications it is necessary to adjust the reference to lower than 100%. This is possible with the parameter P1 (MUL1). The polarity is set with the signal REFPOL and includes the signal re- or unwinder and the winding direction.

The parameter ADDVAL1 is used to the overmodulation or to the commissioning.

Block numbers and pin addresses *Table 2*

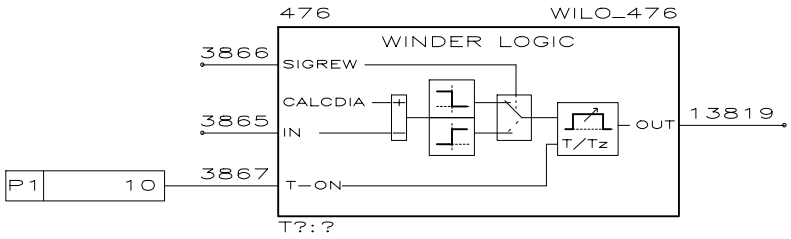
FB number	**469**		Scaling factor	Range of values max.	min.	Default	Unit
IN	3813			-20000	20000		
CGHMUL	3814		BI	-1	0		
REFPOL	3815		BI	-1	0		
ADDPOL1	3816		BI	-1	0		
ADDVAL1	3817			-20000	20000	0	
MUL1	3818			0	5000	5000	
RATREF	13804			-20000	20000		
POLREF	13805			-20000	20000		
OUT	13806			-20000	20000		

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Winder logic

WILO



Summary

In the depending of the diameter (CALCDIA) the output will set for a time of T_ON.

Call WILO_476

Connections *Table 1*

Name	Type	Description
SIGREW	II	Signal for the rewinding or unwinding
CALCDIA		Calculated diameter (internal connection)
IN	II	Input of the level
T-ON	PI	Time setting the output
OUT	OI	Output

Function

The output is set to -1, when the difference from CALCDIA and the input IN change the polarity from 0 to -1. The output stays at -1 during the time T-ON. The input IN can be connected to a parameter or to an analogue input.

The direction of the polarity is depended of the signal SIGREW.

When the signal SIGREW is zero (agree with an unwinder), the output goes to logic one decreasing the diameter CALCDIA lower than IN.

When the signal SIGREW is -1 (agree with a rewinder), the output goes to logic one increasing the diameter CALCDIA higher than IN.

Please remind, that the maximum value of the calculated diameter (CALCDIA) is 4000.

Block numbers and pin addresses *Table 2*

FB number	**470**		Scaling factor	Range of values max.	min.	Default	Unit
IN	3865			0	4000		
SIGREW	3866		BI	-1	0		
T-ON	3867			0	32767	10	
OUT	13819		BI	-1	0		

Guide for using Function Block in the Graphical Application Designer

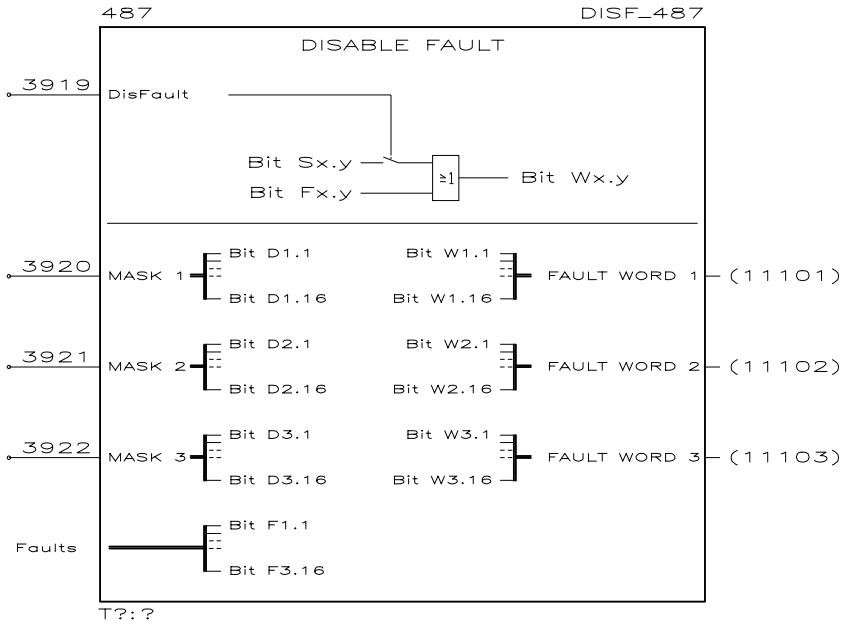
Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Function Blocks valid for Software Version:
21.122

Function Block	New	Modified
DISF_487	X	
PARW_501		❷
PARW_502		❷
PARW_503		❷
PARW_504		❷
PARW_505		❷
PARW_506		❷
PARW_507		❷
PARW_508		❷
PARW_509		❷
PARW_510		❷
PARW_511	X	
PARW_512	X	
PARW_513	X	
PARW_514	X	
PARW_515	X	
PARW_516	X	
PARW_517	X	
PARW_518	X	
PARW_519	X	
PARW_520	X	
PARW_521	X	
PARW_522	X	
PARW_523	X	
PARW_524	X	
PARW_525	X	
SETF_486	X	

❶ Modification of Software
❷ Modification of Diagram
❸ Modification of Connections

Disable Fault DISF



Summary

The block **DISF** (**DIS**able **F**ault) is used to disable one or more faults when activ fault signals occur.

Call DISF_487

Connections *Table 1*

Name	Type	Description
IN	II	Signal to disable the faults
Mask 1	II*	Input mask to disable the faults in Fault Word 1 (111.01)
Mask 2	II	Input mask to disable the faults in Fault Word 2 (111.02)
Mask 3	II	Input mask to disable the faults in Fault Word 3 (111.03)

Function

If the control input "DisFault" is set (to <> 0) and one or more bits in the inputs "Disable.[MASK1]" to "Disable.[MASK3]" are set, the same bits in the signals "Fault Word 1" to "Fault Word 3" (111.01 - 111.03) will not be set and the corresponding faults will not be activated when the reason for these faults occurs.

If the control input "DisFault" is reset (to 0) and the reasons for the faults are still activ, the corresponding fault will set immediately

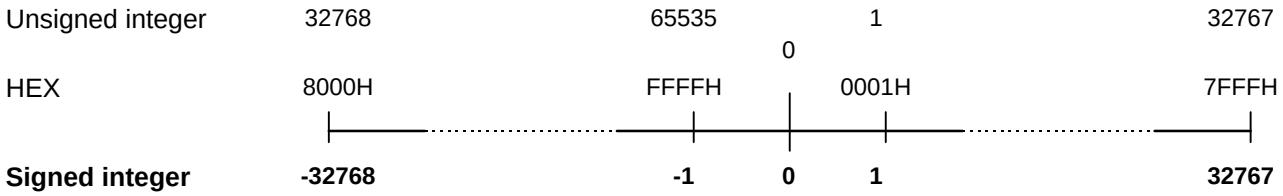
Table : Reaction of DCS500 when the occasion of the disabled fault occurs

Bit	Fault Word 1	(Rem.)	Fault Word 2	(Rem.)	Fault Word 3	(Rem.)
15	Field Ex.1 commu. error	F33 (3)	Motor overspeed	F37 (1)	Rev fault	F65 (1)
14	Field Ex.1 overcurrent	F32 (3)	Motor stalled	F23 (1)	Master-Slave fault	F66 (1)
13	Not in syncronism	F31 (3)	Field Ex. 2 not OK	F43 (3)	Timeout fieldbus	F60 (1)
12	Mains overvoltage	F30 (3)	Field Ex. 1 not OK	F42 (3)	Null	
11	Mains undervoltage	F29 (2)	Local & Disconnected	F20 (3)	Null	
10	No BRAKE ack.	F52 (1)	No C FAN ack	F50 (2)	Null	
09	Motor 2 overload (calc.)	F27 (1)	Par backup fault	F18 (1)	Null	
08	Mot-2 overtemp.(meas.)	F48 (1)	Type coding fault	F17 (4)	Null	
07	I/O board not found	F44 (3)	No main cont. ack	F41 (2)	Null	
06	Motor 1 overload (calc.)	F07 (1)	No ext. FAN ack	F40 (2)	Pfail wact	
05	Mot.1 overtemp.(meas.)	F06 (1)	Speed meas. fault	F14 (2)	Write backup alarm	A136
04	Earth fault	F05 (1)	No field ack	F39 (1)	Backup not allowed	A134
03	Converter overtemp.	F04 (1)	Phase sequence fault	F38 (1)	Param set 2 missing	A132
02	Armature overvoltage	F28 (1)	Field Ex. 2 comm. error	F36 (3)	Init value read, S2	A130
01	Overcurrent	F02 (2)	Field Ex. 2 overcurrent	F35 (3)	Type code changed	A129
00	Auxiliary undervoltage	F01 (3)	Arm. current ripple	F34 (4)	Panel disconnected	A128

Remarks

(1) The fault is not displayed. The drive is **not** blocked.
(2) The fault is not displayed. The drive is blocked because of the theshold of corresponding parameter or pointer.
(3) The fault is not displayed. The drive´s reaction cannot be supressed.
(4) The fault is not displayed. The reaction could not be tested

A parameter (PAR or PARW) connected to the input MASK need a signed integer value.



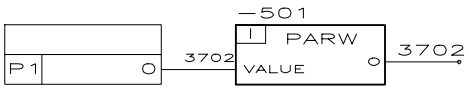
Block numbers and pin addresses *Table 2*

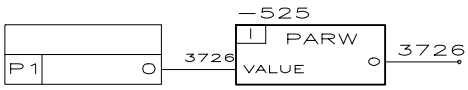
FB number	487
IN	3919
MASK 1	3920
MASK 2	3921
MASK 3	3922

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Parameter PARW





Summary

The **PARW** block can be used to.present parameters of the winder application.
PARW blocks are not included in execution order list of Task1, Task2 or Task3.

Call PARW_501 ... PARW_525

Connections *Table 1*

Name	Type	Description
VALUE	II	User defined parameter
O	OI	Output value of the parameter

Block numbers and pin addresses *Table 2*

FB number	501	502	503	504	505	506	507	508	509	510
VALUE, O	3702	3703	3704	3705	3706	3707	3708	3709	3710	3711

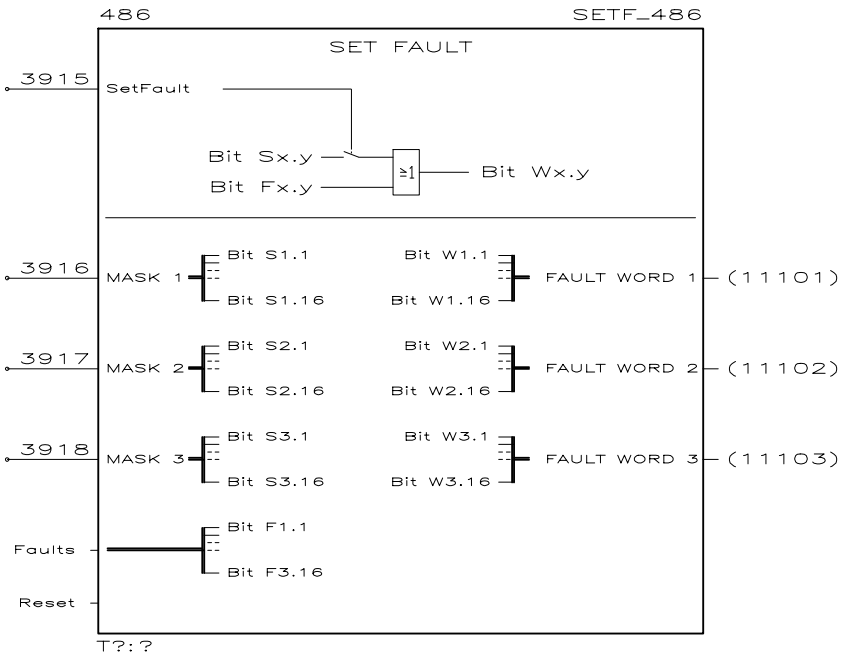
FB number	511	512	513	514	515	516	517	518	519	520
VALUE, O	3712	3713	3714	3715	3716	3717	3718	3719	3720	3721

FB number	521	522	523	524	525					
VALUE, O	3722	3723	3724	3725	3726					

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language), NAME*

SET FAULT SETF



Summary

The block **SETF** (**SET F**ault) is used to set one or more faults without activ fault signals.

Call SETF_486

Connections *Table 1*

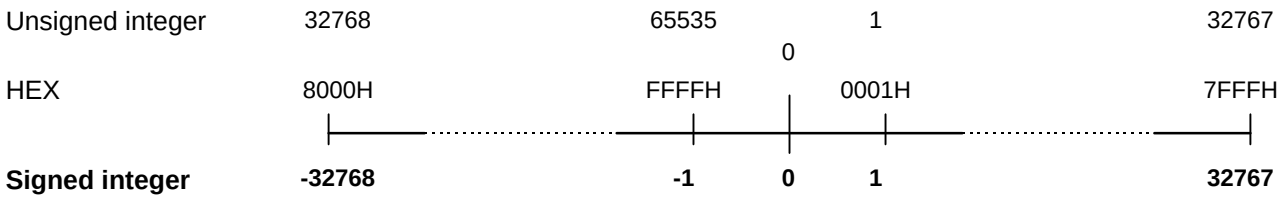
Name	Type	Description
IN	II	Signal to set the faults
Mask 1	II	Input mask to set the faults in Fault Word 1 (111.01)
Mask 2	II	Input mask to set the faults in Fault Word 2 (111.02)
Mask 3	II	Input mask to set the faults in Fault Word 3 (111.03)

Function

If the control input "SetFault" is set (to <> 0) and one or more bits in the inputs "Set.[MASK1]" to "Set.[MASK3]" are set, the same bits in the signals "Fault Word 1" to "Fault Word 3" (111.01 - 111.03) will be set and the corresponding fault will be activated.

If the control input "SetFault" is reset (to 0) and a RESET takes place, the bits in the "Fault Word 1" to "Fault Word 3" (111.01 - 111.03) will be reset and the corresponding fault will be cleared.

A parameter (PAR or PARW) connected to the input MASK need a signed integer value.



Block numbers and pin addresses *Table 2*

FB number	487
IN	3915
MASK 1	3916
MASK 2	3917
MASK 3	3918

Guide for using Function Block in the Graphical Application Designer

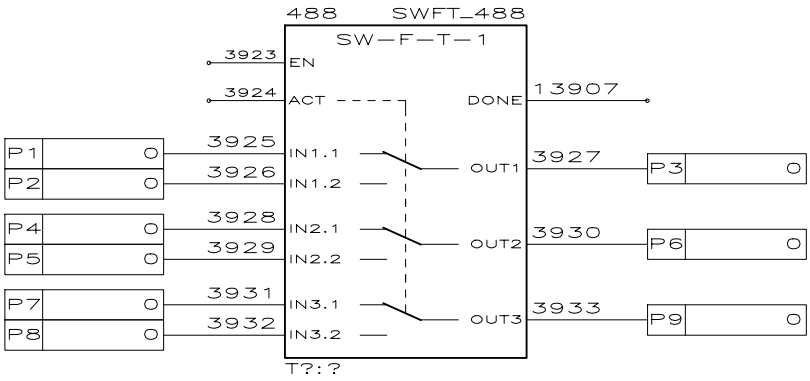
Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Function Blocks valid for Software Version:
21.123

Function Block	New	Modified
SWPT_488	X	
SWPT_489	X	
SWPT_490	X	
SWPT_491	X	
SWPT_492	X	
SWPT_493	X	

❶ Modification of Software
❷ Modification of Diagram
❸ Modification of Connections

Switch parameter and transfer — SWPT



Summary

The block **SWPT** (**SW**itch **P**arameter and **T**ransfer) is used to switch parameters with changover and to tranfer to another parameter.

Call SWPT_488

Connections *Table 1*

Name	Type	Description
EN	II	Enable data transfer
ACT	II	Activate the switching
IN1.1	PI	Value of parameter 1 for the switch 1
IN1.2	PI	Value of parameter 2 for the switch 1
OUT1	PI	Destination address for the transfer of switch 1
IN2.1	PI	Value of parameter 1 for the switch 2
IN2.2	PI	Value of parameter 2 for the switch 2
OUT2	PI	Destination address for the transfer of switch 2
IN3.1	PI	Value of parameter 1 for the switch 3
IN3.2	PI	Value of parameter 2 for the switch 3
OUT3	PI	Destination address for the transfer of switch 3
DONE	OI	Set to -1 when the data tranfer is completed

Function

When the input EN is <>0 the selected parameters will be written to the parameters indicated by the destination addresses. With EN = 0 the data transfer not occurs.

With the input ACT = 0 the parameters at INx.1 (with x in 1,2,3) are selected for the transfer. With active input ACT (<>0) the parameters at INx.2 are selected.

When the data transfer of this block starts, the output signal DONE is 0. This signal get -1 when the transfer of the whole block is completed.

Attention ! Let no destination address (P3,P6,P9) with 0. It can give rise to unintended reactions. (alternative e.g. set the address of a not used parameter)

Block numbers and pin addresses *Table 2*

FB number	488	489	490	491	492	493
EN	3923	3934	3945	3956	3967	3978
ACT	3924	3935	3946	3957	3968	3979
IN1.1	3925	3936	3947	3958	3969	3980
IN1.2	3926	3937	3948	3959	3970	3981
OUT1	3927	3938	3949	3960	3971	3982
IN2.1	3928	3939	3950	3961	3972	3983
IN2.2	3929	3940	3951	3962	3973	3984
OUT2	3930	3941	3952	3963	3974	3985
IN3.1	3931	3942	3953	3964	3975	3986
IN3.2	3932	3943	3954	3965	3976	3987
OUT3	3933	3944	3955	3966	3977	3988
DONE	13907	13908	13909	13910	13911	13912

Guide for using Function Block in the Graphical Application Designer

Definable attributes : *Task number / Execution order, Comment text (several languages), General text (one language)*

Index of Function Block description

Appendix A - Relationship between Application blocks and Firmware releases

related GAD Libraries	1.01	1.21	1.22	1.30
related Firmware release	≤ 21.108	21.121	21.122	21.123
Application Blocks				
ABS	x			
ACW		x		
ADD2	x			
ADD4	x			
AND2	x			
AND4	x			
BITGET	x			
BITSET	x			
COMP-I	x			
CONV-BI	x			
CONV-IB	x			
COUNT	x			
DIAC		x		
DIAREL		x		
DISF			x	
DIV2	x			
DT		x		
FILT-I	x			
FUNG-1V	x			
INTEG	x	P		
INV	x			
LIM-N1	x			
LSW	x			
MAX2	x			
MIN2	x			
MONO	x			
MUL	x			
MULDIV	x			
MUX-I4	x			
OR2	x			
OR4	x			
OSC-B	x			
PAR	x			
PARW501_510		x	G	
PARW511_525			x	
PI-I	x			
SC		x		
SETF			x	
SR	x			
SR-D	x			
SSL		x		
SUB2	x			
SW-C1	x			
SW-P-T				x
TERE		x		
TOFF	x			

related GAD Libraries	1.01	1.21	1.22	1.30
related Firmware release	≤ 21.108	21.121	21.122	21.123
Application Blocks				
TON	x			
TOTE		x		
TRII		x		
VERE		x		
WILO		x		
WRITE1	x			
WRITE4	x			
XOR	x			

Legend

x available from Firmware release ... on
C Place of **C**onnection modified
D Function Block **D**eleted
G **G**rafic modified
F **F**irmware modified
P **P**arameter value modified

Ident. Nr. : 3ADW 000 048 R0301 REV C

ABB Industrietechnik AG
Produktbereich Antriebstechnik
Postfach 1180
D-68619 Lampertheim
Telefon: (0 6206)-5 03-0
Fax: (0 6206)-5 03-5 63
Telex 4 62 411605 ab d