We reserve all rights in this document and in the information contained therein. Reproduction, use or disclosure to third parties without express authority is strictly forbidden. © ABB

FUNCTIONAL DESCRIPTION
PP_Element Library

Contents

1 Introduction

The Pulp & Paper Element Library for AC800M controller Series is a set of software functions that give the programmer additional functions for programming of controls for various kinds of industrial processes.

The system provides programming in compliance with the IEC 6-1131. 1131 is based on software implemented function blocks, FB. The Function Blocks are connected to each other into programs which form a complete control function.

You use 1131 in a graphical environment with a personal computer as engineering stations.

2 Call Name

Each function block has its own unique call name. Many of the function blocks have call parameters which allow size, function, data type and other properties to be determined.

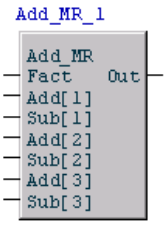
The mnemonic call name gives information about the basic function of the function block.

3 Call Parameters

The call parameters for function function blocks may specify the size, the data type, the number of inputs and outputs etc.

4 Graphic Symbol

Each Function Block has a graphic symbol similar to the figure below,



Normally, the call name of the function block is written in the uppermost part of the symbol.

5 Connection of Function Block Inputs and Outputs

Only inputs and outputs with the same data type can be interconnected. If, for example, an output with data type dint is to be connected to an input of type real, a code conversion function block, for example, DintToReal must be inserted in the data path.

6 IEC 61131-3 Standard Data Types

Data type	Description
Bool	Boolean
Date_and_time	Date and time of day (Year, Month, Day, Hour, Minute, Second, Millisecond)
Dint	Signed double integer (32 bits). Range - 2,147,483,648 to 2,147,483, 647
Dword	Double word (32-bit) bit string
Int	Signed integer (16-bit). Range - 32,768 to 32,767
Real	Floating point values in the range +/- 10^(+/-38)
String	String data type for character strings. Maximum string length 140 characters
Time	Duration (Day, Hour, Minute, Second, Millisecond)
Uint	Unsigned integer (16-bit) with the range 0...65535
Word	Word (16-bit) bit string

7 PP_Element Data Types

Data type	Description		
Real16	Structured type	Name	Type
		R1	real
		R2	real
		R3	real
		R4	real
		R5	real
		R6	real
		R7	real
		R8	real
		R9	real
		R10	real
		R11	real
		R12	real
		R13	real
		R14	real
		R15	real
		R16	real

8 Element Libraries

Pulp and Paper BU has developed an element library, which contains most of the Function Blocks that are required for process control.

9 Programming with 1131

When you program a controller, choose Function Blocks from the library and insert them into the programming pane. The function blocks are then interconnected on the diagram, to describe the exchange of data between them. The engineering station translates the program into an object code, which is loaded into the target system.

10 Function Block description

10.1 Summary

An function block description starts with a short Summary that describes the main function.

To the right of the summary, the graphic symbol of the function block is shown. Symbols with a variable number of connections are shown, as in the figure below.

Under the heading Call, the calling of the function block is described.

10.2 FUNCTION BLOCK (C1, C2, ... C n)

If the function block has call/function parameters, these are described in a table where their significance and permissible values are stated.

Function block terminals are described under the heading Connections. This section is normally dominated by a table giving each terminal name, type and description.

The type of the connection is given as in, if it is an input terminal, or out if it is an output terminal

10.3 Function

The descriptions end with a detailed functional description under the heading Function.

11 Execution Times

The execution times, specified in the table below, apply for AC800M version xx? and AC800M version yy? Controllers.

For more information about execution times for I/O, see AC800M User's Guide.

Name	AC800M (PM860) Execution Time, typical (ms)	AC800C, Execution Time, typical (ms)	Notes
ADD_MR			
Alarm_Lim2			
Alarm_Lim4			
Avg			
AnalogDelay			
Comp_I			
Comp_R			
Con_PU1			
CONV_BI			
CONV_IB			
Count_L			
DeMux_MI_IL			
DeMux_MI_R			
Div_R			
DERivator			
Filt_1P			
Filt_2P			

Name	AC800M (PM860) Execution Time, typical (ms)	AC800C, Execution Time, typical (ms)	Notes
Fung_1V			
INTegrator			
LIM			
Lim_N			
Max_R			
Min_R			
Mono			
Move_B			
Move_IL			
Move_R			
Mux_I_B			
Mux_I_IL			
Mux_I_R			
Mux_MI_IL			
Mux_MI_R			
Mux_MN_IL			
Mux_MN_R			
Mux_N_IL			
Mux_N_R			
MuxA_I			
Osc_B			
Ramp			
Reg_IL			
Reg_R			
SR_D			
SW_B			
SW_C_B			
SW_R			
SW_C_R			
TimerPuls			
Tresh_L			

12 Memory Space Requirements

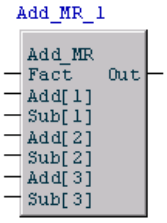
The following pages show the memory space requirements for Pulp and Paper Function Blocks. The memory requirements described in the table below, are added to the memory space requirements for P&P Functional Units.

Name	1 st instance of FB (byte)	Additional instances of FB (byte)	Notes
ADD_MR			
Alarm_Lim2			
Alarm_Lim4			
AnalogDelay			
Avg			
Comp_I			
Comp_R			
Con_PU1			
CONV_BI			
CONV_IB			
Count_L			
DeMux_MI_IL			
DeMux_MI_R			
Div_R			
DERivator			
Filt_1P			
Filt_2P			
Fung_1V			
INTegrator			
LIM			
Lim_N			
Max_R			
Min_R			
Mono			
Move_B			
Move_IL			
Move_R			
Mux_I_B			
Mux_I_IL			
Mux_I_R			
Mux_MI_IL			
Mux_MI_R			
Mux_MN_IL			
Mux_MN_R			
Mux_N_IL			
Mux_N_R			

Name	1 st instance of FB (byte)	Additional instances of FB (byte)	Notes
MuxA_I			
Osc_B			
Ramp			
Reg_IL			
Reg_R			
SR_D			
SW_B			
SW_C_B			
SW_R			
SW_C_R			
TimerPuls			
Tresh_L			

13 Function Description

13.1 ADD_MR, Adder with Multiplied Output



Summary

ADD_MR (ADDer - with Multiplier Real numbers) is used for addition of an optional number of real numbers. The numbers are added in two groups, after which the second group is subtracted from the first. The result is multiplied with a real number.

Call Name: ADD_MR [n]

Parameter	Significance	Permissible values
N	Number of positive and negative inputs in the function block.	1 - n

Terminal Description

Name	Data Type	Description
Fact	Real	Input for multiplication factor.
Add[1]	Real	Input for augend with positive weight factor.
Sub[1]	Real	Input for addend with negative weight factor.
Add[n]	Real	Input for addend with positive weight factor.
Sub[n]	Real	Input for addend with negative weight factor.
Out	Real	Output for result.

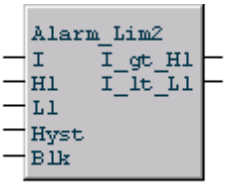
Function

The sum of the real numbers at the inputs Sub[1] .. Sub[X] is subtracted from the sum of the inputs Add[1] .. Add[X], after which the result is multiplied with the value at input Fact. The result is stored at the output Out.

Overflow

If the maximum positive or negative real value is exceeded, the output is limited to the greatest or lowest representable value respectively.

13.2 Alarm_Lim2, Comparator with 2 limits



Summary

Alarm_Lim2 is used for supervision of 2 limits, high and low, for alarm purpose.

Call Name: Alarm_Lim2

Parameter	Significance	Permissible values

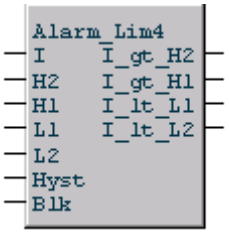
Terminal Description

Name	Data Type	Description
I	Real	Input for value to be checked.
H1	Real	Input for high alarm limit.
L1	Real	Input for low alarm limit.
Hyst	Real	Input for hysteresis.
Blk	Bool	Input for blocking of function.
I_gt_H1	Bool	Output if input is greater than H1.
I_lt_L1	Bool	Output if input is less than L1.

Function

The input is compared with the high and low limits and if exceeded the corresponding output is set. The output is reset when the input value is less/greater than the limit minus the Hysteresis. The input Blk disables the limit check and sets the outputs to zero.

13.3 Alarm_Lim4, Comparator with 4 limits



Summary

Alarm_Lim4 is used for supervision of 4 limits, high high, high, low and low low, for alarm purpose.

Call Name: Alarm_Lim4

Parameter	Significance	Permissible values

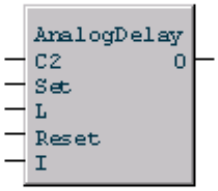
Terminal Description

Name	Data Type	Description
I	Real	Input for value to be checked.
H2	Real	Input for high high alarm limit.
H1	Real	Input for high alarm limit.
L1	Real	Input for low alarm limit.
L2	Real	Input for low low alarm limit.
Hyst	Real	Input for hysteresis.
Blk	Bool	Input for blocking of function.
I_gt_H2	Bool	Output if input is greater than H2.
I_gt_H1	Bool	Output if input is greater than H1.
I_lt_L1	Bool	Output if input is less than L1.
I_lt_L2	Bool	Output if input is less than L2.

Function

The input is compared with the high high, high, low and low low limits and if exceeded the corresponding output is set. The output is reset when the input value is less/greater than the limit minus the Hysteresis. The input Blk disables the limit check and sets the outputs to zero.

13.4 AnalogDelay, Delay of analog signals



Summary

AnalogDelay is used for delay of analog values.

Call Name: AnalogDelay

Parameter	Significance	Permissible values
C2	Number of places in delay queue	2 …

Terminal Description

Name	Data Type	Description
Set	Bool	Loading of new values each sample.
L	Bool	Dynamic loading of data.
Reset	Bool	Reset of delay queue.
I	Real	Input.
O	Real	Output for delayed input value.

Function

New data is entered into the element by replacing the oldest sample with the current value of “I”. The output is delayed by the number of places in queue (C2) multiplied by the sampling time of the control task.

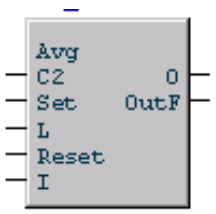
When the input “Set” is 1, data is loaded into the delay queue each sample and the time delay is incremented by TS sec. When the input “Set” is reset (to 0), the delay queue is halted. Data is also loaded when input “L” goes from 0 to 1.

Example1: If “C2” is 10, the controller task time is 1 sec and input “Set” is 1, the output is delayed by 10 sec.

Example2: If “C2” is 10, the controller task time is 1 sec and input “L” is triggered every 2 second the output is delayed by 20 sec.

When the input “Reset” is 1, all data in the queue is cleared. Changing the number of places in the delay queue, with parameter “C2”, requires a “cold start” of the controller to take place.

13.5 Avg, Moving Average



Summary

Avg is used for calculation of a moving average of n number of samples.

Call Name: Avg

Parameter	Significance	Permissible values
C2	Number of samples	2 …

Terminal Description

Name	Data Type	Description
Set	Bool	Loading of new values each sample.
L	Bool	Dynamic loading of data.
Reset	Bool	Reset of queue.
I	Real	Input.
O	Real	Output for moving average.
OutF	Real	Output for first value in queue.

Function

New data is entered into the queue by replacing the oldest sample with the current value of “I”. The floating average is calculated by summing all the values in the queue and then dividing by the number of queue places C2 as shown in figure 1 below.

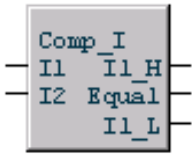
$$O = \left(\frac{1}{C2} \sum_{n=1}^{C2} In \right)$$

Figure 1. Floating Average Calculation.

When the input “Set” is 1, data is loaded to the outputs “O” and “OutF”. The number of values samples is set to 1. When the input “Set” is reset (to 0), the last value loaded “OutF” remains and “O” is calculated as in figure 1 above. Data is also loaded when input “L” goes from 0 to 1.

If the number of sampled values, i.e., the number of times the function block has been executed after “Set” has gone from 1 to 0, is less than “C2” then this value is substituted as the divisor in the calculation. When the input “Reset” is 1, all data in the queue is cleared. Changing the number of samples in the queue, with parameter “C2”, requires a “cold start” of the controller to take place.

13.6 Comp_I, Comparator (Integer Value)



Summary

Comp_I is used for comparison of two integer values.

Call Name: Comp_I

Parameter	Significance	Permissible values

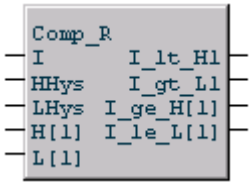
Terminal Description

Name	Data Type	Description
I1	Dint	Input value 1.
I2	Dint	Input value 2.
I1_H	Bool	Output which is set if input I1 is greater than input I2.
Equal	Bool	Output which is set if input I1 is equal to input I2.
I1_L	Bool	Output which is set if input I1 is less than input I2.

Function

The values at the two inputs I1 and I2 are compared and the result og the comparison can be read at the outputs I1_H, Equal and I1_L.

13.7 Comp_R, Comparator (Real Value)



Summary

Comp_R is used for comparison of two integer values.

Call Name: Comp_R [n]

Parameter	Significance	Permissible values
N	Number of high and low limits to compare.	1 - n

Terminal Description

Name	Data Type	Description
I	Real	Input value.
Hhys	Real	High hysteresis.
Lhys	Real	Low hysteresis.
H[1]	Real	High limit 1.
L[1]	Real	Low limit 1.
H[n]	Real	High limit n.
L[n]	Real	Low limit n.
I_lt_H1	Bool	Input less than H1.
I_gt_L1	Bool	Input greater than L1.
I_ge_H[1]	Bool	Input greater than or equal to H1.
I_le_L[1]	Bool	Input less than or equal to L1.

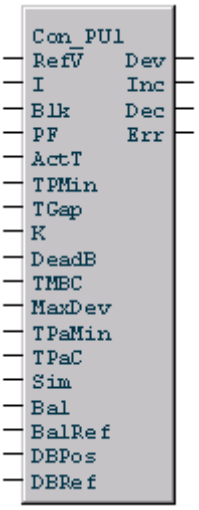
Function

Input signal I is compared with the limit value specified at the inputs H1 – H[n] and L1 – L[n]. For upper limits, the output for the different limits will be set when the input I becomes equal to, or greater than, the limit value. For lower limits, the output is set when I becomes less than or equal to the limit value.

Hysteresis

The input HHys gives the hysteresis for all limits at high level and LHys for all limits at low level. The hysteresis is given as the difference between the I values for which the different outputs are set and the values for which they are reset. At high level the outputs will be reset when I becomes lower than the limit minus the hysteresis HHys. At low level the outputs will be reset when I becomes greater than the limit value plus the hysteresis LHys.

13.8 Con_PU1, Three State Controller



Summary

CON_PU1 is a three state controller intended to control, by means of pulses (increase/decrease), a process via an integrating actuator such as a shifting motor or a solenoid valve. CON_PU1 can be combined with, for example, a PI regulator for process control.

Call Name: Con_PU1

Parameter	Significance	Permissible values

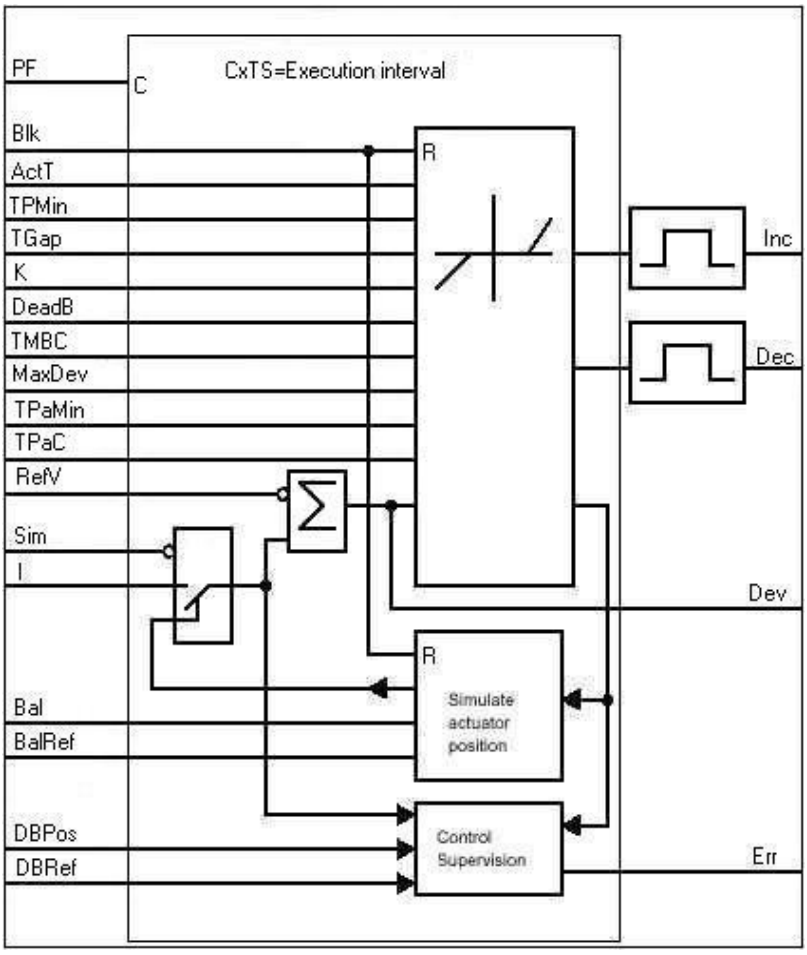
Terminal Description

Name	Data Type	Description
RefV	Real	Reference Value
I	Real	Measured Value
Blk	Bool	Block pulse calculation
PF	Dint	Pulse Frequency factor
ActT	Time	Actuator Time [s]
TPMin	Time	Time Pulse Minimum
Tgap	Time	Time Gap
K	Real	Pulse length factor
DeadB	Real	Deadband
TMBC	Time	Time Minimum Between Change
MaxDev	Real	Maximum Deviation
TpaMin	Time	Time Pause Minimum
TpaC	Real	Time Pause Constant
Sim	Bool	Simulate actuator position
Bal	Bool	Balance
BalRef	Real	Balance Reference
DBPos	Real	Deadband for actuator Position Change [%]

DBRef	Real	Deadband Reference for position change alarm
Dev	Real	Deviation
Inc	Bool	Increase
Dec	Bool	Decrease
Err	Bool	Error

Function

The three state controller CON-PU1 is intended to control, by means of pulses at the outputs Inc and Dec (increase/decrease respectively), a process via, for example, a shifting motor or a solenoid valve. The following inputs should be in unit % (0-100): RefV, I, DeadB, MaxDev, BalRef, DBPos, DBRef.



The feedback can be obtained from the actuator or the function block can simulate the actuator position internally. The effective pulse duration is proportional to the control error and is calculated as follows:

PL_{eff} =| Dev x K x ActT | / 100, where:

Dev=Control deviation=I–RefV (%)

K=Pulse length factor

ActT=Actuator delay time interval in seconds

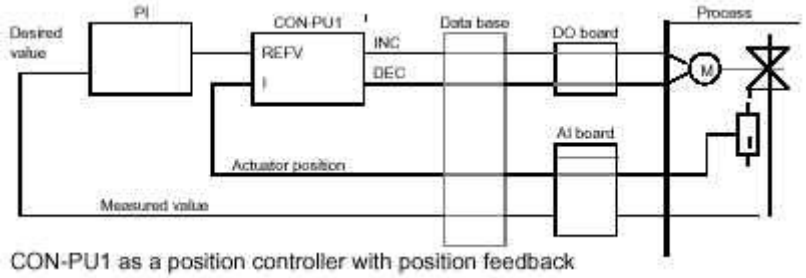
To permit gradual correction of a large control error, | Dev |, is limited to MaxDev. The actual time pulse duration is calculated as:

PL=(| DevxKxActT |)/100+TGap+TPMin where:

TPMin=Min pulse duration for excitation of the shifting motor or solenoid valve. This is the shortest pulse length which affects the actuator.

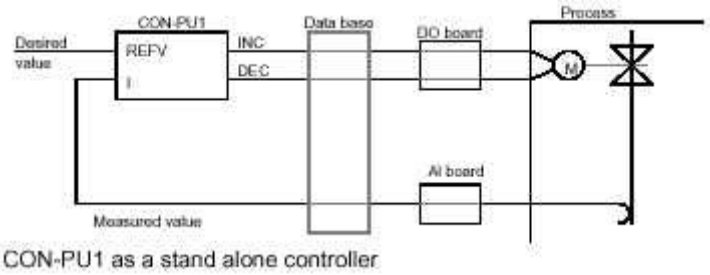
TGap=Play compensation. Added to the first pulse after a polarity change.

For control errors less than a deadband (DeadB, %) the correction is considered to be complete and no further pulse is calculated. TMBC is a delay time that gives the shortest pause time after a change of the pulse outputs Inc/Dec. Calculation of a new pulse duration is activated at the interval PF x Ts where Ts is the sample time of the function, that is, the cycle time of the function block.

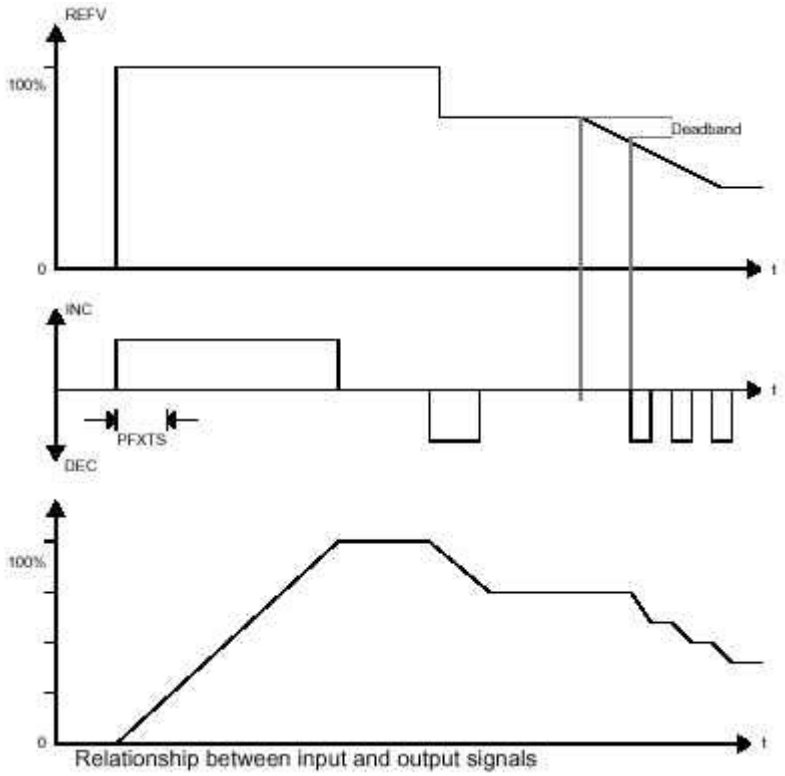


CON-PU1 as a position controller with position feedback

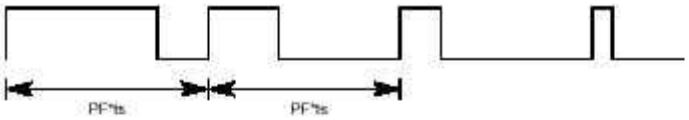
The least permitted pause time between two pulses is determined however by the parameters TPaMin and TPaC so that the Pause time=TPaMin+TPaC x PL where PL is the pulse duration calculated most recently. These possibilities of setting the pause time are normally only used when CON_PU1 is used as a stand alone regulator and provide great flexibility in trimming the regulator. By setting PF x Ts relatively large and TPaMin=TPaC=0, a constant pulse frequency is obtained. To obtain a constant pause time between the pulses, TPaC is set to 0 and TPaMin to the required pause time. This is necessary when controlling processes with long time constants.



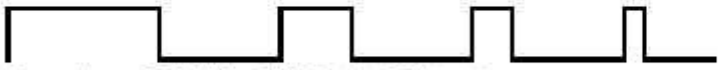
CON-PU1 as a stand alone controller



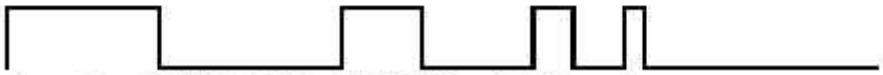
Relationship between input and output signals



Pause time = TPAMIN + TPAC x PL, TPAMN =TPAC = 0



Pause time = TPAMIN + TPAC x PL, TPAC = 0



Pause time=TPAMIN + TPAC x PL. TPAMIN and TPAC > > 0

In processes with long delay times, pause times ≥ the delay time are selected.

The pause time can be made to be dependent on the duration of the latest pulse calculated with the parameter TPaC. This function is used when a rapid control of small errors is required but when limits on the speed of change in the process response require long pause times with large changes.

When CON_PU1 is used as a position regulator for a motor actuator, different building up processes are obtained depending on a parameter G, this being in principle the relationship between the real data of the actuator and the parameters set in CON_PU1.

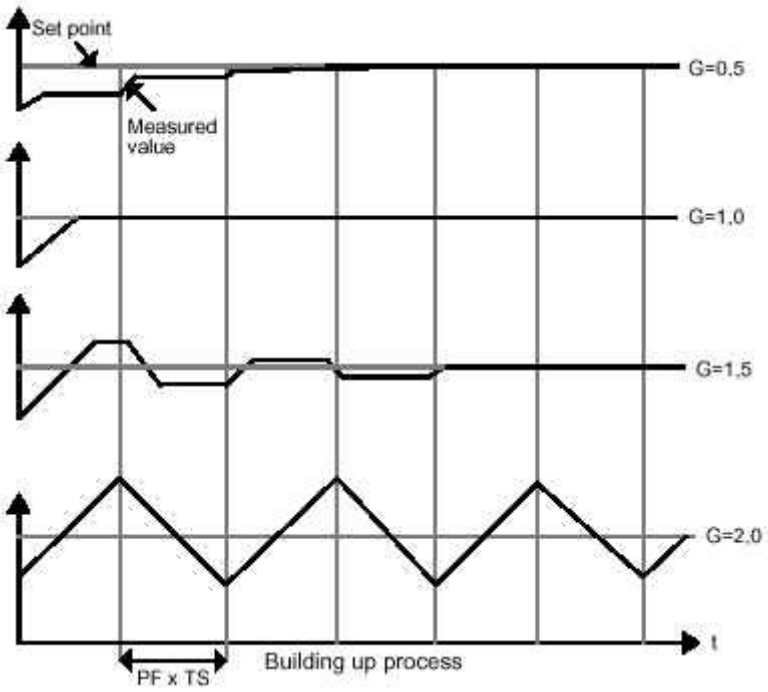
$G=(kxActT)/ActT_v +100x(TGapzTGap_v +TPMinzTPMin_v)/ (ActT\ v\ x|\ Dev\ |);$

$TGap_v$ is the actuator play.

$TPMin_v$ the least pulse length to which the actuator reacts.

$ActT_v$ the time delay time interval of the actuator.

The figure below shows the building up process for different G, where only (k x ActT)/$ActT_v$ is included. The system is unstable when G≥2



Simulation of Actuator Position

If it is impossible to feed back the actuator position to the function block, it can be simulated internally in the function block by integrating Dev x K x ActT. The simulated value is limited to 0-100%. When Bal is set, BalRef is used as an integrator value, for example, with a limit switch on the actuator to set the integrator to a known value or forced control when controlling the process manually. With simulation, the preceding regulator should have an output signal >0–100% (–100–200%).

Supervision

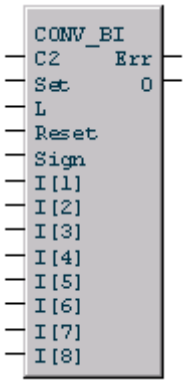
When actuator position is calculated to have changed > DBPos, a check is made to determine if | I(t=n) - I(t=0) | < DBRef, if so, Err is set, otherwise Err is cleared. I(t=0) is the input value when the calculation started and I(t=n) is the input value at the time when calculated actuator position has changed > DBPos. DBPos is given in % of the span the actuator is to have moved in ActT seconds. The calculation starts all over after exceeding DBPos, when Dev changes sign or when DBPos is not exceeded after 9 pulses. Err is reset when Dev changes sign, or if DBPos is not exceeded after 9 pulses. Theoretically you can set DBRef=DBPos. This means that the actuator follows the controller without delay. In practice, you will have to set DBRef<DBPos.

Ordinary Function

With ordinary function only pulse generation according to

PL= (| Dev | x KI x ActT) / 100+TPMin+TGap is performed. If | Dev | <DeadB no pulse is obtained.

13.9 CONV_BI, Code Converter (Boolean to Integer)



Summary

CONV_BI (**CONV**erter _ **B**oolean to **I**nteger) converts data from BC, BCD, or 1-of-N formed from Boolean variables into integers. The function block has a memory function for storage of converted data.

Call Name: CONV_BI

Parameter	Significance	Permissible values
N	Number of bits to convert	1 – 31

Terminal Description

Name	Data Type	Description
C2	Dint	Type of conversion. 1=BC, 2=BCD, 3=1-of-N
Set	Bool	Input for storage of new value each time the function block is executed.
L	Bool	Load. Dynamic input for loading of data.
Reset	Bool	Input for clearing of data. Reset overrides Set and L.
Sign	Bool	Input set to “1” with negative data
I[1]	Bool	Input value 1.
I[2]	Bool	Input value 2.
I[n]	Bool	Input value n.
Err	Bool	Error. Set to “1” if more than one input is set with conversion of 1-of-N code or if integer value cannot be represented
O	Dint	Output.

Function

Converts binary code (BC), binary coded decimal code (BCD) or 1-of-N code to integers of type dint (32 bit). SIGN input can be used as a sign bit (C2=1, 2 or 3).

Example of different decodings

Input terminal	Output Value O		
	C2=1, BC	C2=2, BCD	C2=3, 1-of-N(1)
I[1]	1	1	1
I[2]	2	2	2
I[3]	4	4	3
I[4]	8	8	4
I[5]	16	10	5
I[6]	32	20	6
I[7]	64	40	7
I[8]	128	80	8
I[31]	2^{30}	$4x10^{7}$	31
Sign	x-1	x-1	x-1

(1) Only one of the inputs may be set with 1-of-N code.

Storage of Data

The code at the inputs I[1] .. I[n] is converted and stored immediately the input L is set. If input Set is set, the input code is converted and the integer is stored each time the function block is executed. When Set is reset (to 0) after having been set, the data stored most recently remains. The input Set overrides the input L, that, is when Set is set, L has no effect.

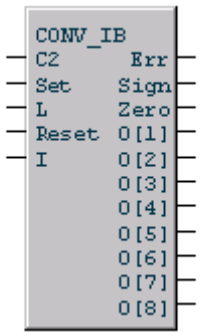
Clearing

The input Reset clears the output and prevents all further storage of data while Reset is set.

Supervision

When converting 1-of-N code the setting of only one of the inputs I[1] .. I[n] is supervised. If two or more inputs are set, the value of the input signal with the lowest number is stored. The error signal output ERR is also set. If the integer value to be stored at the output exceeds the value which can be represented, the error signal output is set. In addition, the output is limited to the upper or lower limit value.

13.10 CONV_IB, Code Converter (Integer to Boolean)



Summary

CONV_IB (**CONV**erter _ **I**nteger to **B**oolean) converts data from integer to Boolean variables using BC, BCD, or 1-of-N conversion. The function block has a memory function for storage of converted data.

Call Name: CONV_IB

Parameter	Significance	Permissible values
N	Number of converted data	1 – 32

Terminal Description

Name	Data Type	Description
C2	Dint	Type of conversion. 1=BC, 2=BCD, 3=1-of-N
Set	Bool	Input for storage of new value each time the function block is executed.
L	Bool	Load. Dynamic input for loading of data.
Reset	Bool	Input for clearing of data. Reset overrides Set and L.
I	Dint	Input.
Err	Bool	Error. Set to “1” if integer value to be converted cannot be represented at the outputs.
Sign	Bool	Input set to “1” with negative data
Zero	Bool	Set to “1” when the input value is zero or input Reset = 1
O[1]	Bool	Output 1.
O[2]	Bool	Output 2.
O[n]	Bool	Output n.

Function

CONV-BI converts integer values of data dint into Boolean variables, using binary code (BC), binary coded decimal code (BCD), 1-of-N code, or uncoded unpacking. The code conversion to be performed is specified by the call parameter C2. With positive data values, the output SIGN is reset (to 0). With negative values, it is set (to 1).

Examples of Output Values with Different Decodings

Input Value I			Affected
C2=1, BC	C2=2, BCD	C2=3, 1-of-N(1)	Output
1	1	1	O[1]
2	2	2	O[2]
4	4	3	O[3]
8	8	4	O[4]
16	10	5	O[5]
32	20	6	O[6]
64	40	7	O[7]
128	80	8	O[8]
256	100	9	O[9]
512	200	10	O[10]
2^{30}	$4x10^7$	31	O[31]
-	$4x10^7$	32	O[32]
(1)	(1)	(1)	Sign

(1) Sign is used as a sign. Only one of the outputs is set with 1-of-N code.

Storage of Data

The integer at input I is converted immediately when input L is set. If input Set is set, the input code is stored, and the result is converted each time the function block is executed. When Set is reset (to 0), after having been set, the data stored most recently remains until the function block once more is executed with one of the inputs Set, L or Reset set. The input Set overrides the input L, that is, when Set is set, L has no effect.

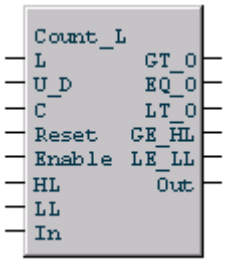
Reset

The input Reset clears the data outputs O[1] to O[n], and prevents all further storage of data while Reset is set.

Supervision

When converting data from integers to 1-of-N code, the numbers are restricted to –32 to +32. If the integer value to be converted exceeds the value which can be represented at the outputs, the error signal output is set. In addition, the output is limited to the upper or lower limit value.

13.11 Count_L, Up/Down Counter with Limits



Summary

Count_L (Counterer - Limiter) is a presettable counter for counting up or down. The counter also checks the counter value relative to 0 and optional max and min values.

Call Name: Count_L

Parameter	Significance	Permissible values

Terminal Description

Name	Data Type	Description
L	Bool	Loads the counter with the value at input In.
U_D	Bool	1 = Counting Up, 0 = Counting Down
C	Bool	Clock. When changes from 0 to 1 the counter counts up or down according to input U_D.
Reset	Bool	Clears the counter
Enable	Bool	Input has to be 1 for counting or loading to take place.
HL	Dint	High Limit
LL	Dint	Low Limit
In	DInt	Input for new value when loading.
GT_0	Bool	Output greater than zero.
EQ_0	Bool	Output equal to zero.
LT_0	Bool	Output less than zero.
GE_HL	Bool	Output greater than or equal to HL.
LE_LL	Bool	Output less than or equal to LL.
Out	DInt	Output.

Function

The counter value increases or decreases by 1 immediately when input C is set. The value increases if U_D=1, and decreases if U_D=0. The duration of the counter period may not be less than 2 x the cycle time of the program.

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

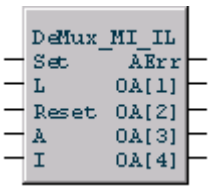
Clearing

The input Reset clears the counter and prevents all further counting or loading. Reset overrides Enable.

Supervision

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

13.12 DeMux_MI, Demultiplexer with Memory



Summary

DeMux_MI (**DEMU**ltiple**X**er **-** with **M**emory and **I**nteger address) is used as a demultiplexer with memory function. DeMux-MI has an optional number of outputs. The data type can be dint or real numbers.

Call Name: DeMux_MI_<C1>

Parameter	Significance	Permissible values
C1	Data Type	IL, R
N	Number of outputs	1 – n

Terminal Description

Name	Data Type	Description
Set	Bool	Input for loading of new value each time the function block is executed.
L	Bool	Load. Dynamic input for loading of data.
Reset	Bool	Input for clearing of data. Reset overrides Set and L.
A	Dint	Address for selection of which output data from input should be stored.
I	C1	Input.
Aerr	Bool	Address Error. Set to “1” if address is greater than the number of outputs, or negative.
OA[1]	C1	Output address 1.
OA[2]	C1	Output address 2.
OA[n]	C1	Output address n.

Function

At which output, OA[1] – OA[n] the data value at input I is to be stored is specified at input A with an address (dint 1 - n).

If the address A is 0, the value 0 is stored at all outputs.

Not addressed outputs are not updated and keep consequently the value from previous execution.

Loading

The value is stored at the instant that the input L is set (to 1). If input Set is set, new data is loaded each time the function block is executed. When Set is reset after having been set, the data most recently loaded remains until the function block is executed with one of the inputs Set, L or Reset set. The input Set overrides the input L, that is, when Set is set, L has no effect.

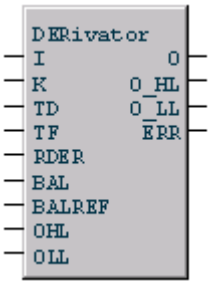
Clearing

Setting the input Reset clears the data outputs and prevents further storage of data.

Supervision

The address A is monitored and if its value is greater than the number of outputs or is negative, the error signal output AErr is set. The data value 0 is then stored at all outputs.

13.13 DERivator, Derivator



Summary

DERivator is used to give derivation effect. The derivation effect can be limited with the filter function, which serves as a low pass filter. The output signal can be limited with limit values specified at special inputs. The balancing function permits the output signal to follow an external reference and permits a bumpless return to the normal function. All transfers from static states are bumpless.

Call Name: DERivator

Parameter	Significance	Permissible values

Terminal Description

Name	Data Type	Description
I	Real	Input.
K	Real	Input for Gain.
TD	Time	Time Derivation.
TF	Time	Time Filter.
RDER	Bool	Reset of Derivator.
BAL	Bool	Balance.
BALREF	Real	Balance Reference.
OHL	Real	Output High Limit.
OLL	Real	Output Low Limit.
O	Real	Output.
O_HL	Bool	Output = High Limit.
O_LL	Bool	Output = Low Limit.
ERR	Bool	Error.

Function

The step response in the time plane for a derivator is:

O(t)=K (TD/TF)e-t/TF x I(t)

where I(t) specifies the magnitude of the step.

The transfer function for a DER function is:

G(s)=K(s x TD)/(1+s x TF)

This has been implemented in the DERivator element.

Gain, Derivation Time, Filter Time and Sampling Time

Certain constants are precalculated and stored internally in the element. When calculating a test is performed to check whether TD and TF>=2 x TS. If not, TD and/or TF is set equal to 2 x TS.

Clearing of the Derivator

Both output O(t) and the internal state of the output are cleared when RDER goes to 1.

Following

If BAL is set to 1, the derivator immediately goes into following 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 limit value concerned. Return to dynamic state is bumpless.

Limitation Function

The limitation function limits the output signal to the values at the inputs OHL for upper limit, and OLL for 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 element 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 sample before, in which the error occurred. After an error, the return to a dynamic state is bumpless, in the same way as in the case above.

13.14 Div_R, Divider Real Values

Summary

Div_R (**Div**ider **R**eal Values) is used for division of two real numbers. When dividing integers, the quotient is obtained with the remainder at a separate input.

Call Name: Div_R

Parameter	Significance	Permissible values

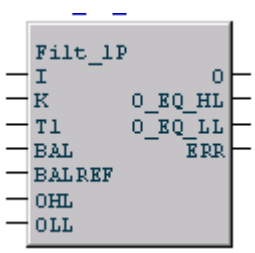
Terminal Description

Name	Data Type	Description
I1	Real	Input for Dividend.
I2	Real	Input for Divisor.
Out	Real	Output.
OutInt	Real	Integer part of Output.
Rem	Real	Remainder

Function

The value at input I1 is divided by the value at input I2. The quotient is stored at output Out. The integer part of the output is available at output OutInt and the remainder at output Rem.

13.15 Filt_1P, Filter 1 Pole



Summary

Filt-1P (**Filt**er **- 1 P**ole) is used as a single pole low pass filter. The output signal can be limited with limit values specified at special inputs. The balancing function permits the output signal to follow an external reference and permits a bumpless return to the normal function. All transfers from static states are bumpless.

Call Name: Filt_1P

Parameter	Significance	Permissible values

Terminal Description

Name	Data Type	Description
I	Real	Input value.
K	Real	Input for Gain.
T1	Time	Input for filter time constant.
BAL	Bool	Balance.
BALREF	Real	Balance Reference.
OHL	Real	Output High Limit
OLL	Real	Output Low Limit
O	Real	Output
O_EQ_HL	Bool	Output Equal to High Limit
O_EQ_LL	Bool	Output Equal to Low Limit
ERR	Bool	Error

Function

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

O(t)=I(t) K (1–e-t/T1).

The transfer function for a single pole low pass filter is:

G(s)=K(1/(1+sT1)).

This has been implemented in the FILT-1P function block.

Gain, Filter Time and Sampling Time

Certain constants are precalculated to make the execution time of the function block as short as possible. The result is stored internally in the function block. When recalculating, a test is performed to check if T1>2 x TS. If not, T1 is set equal to 2 x TS.

Following

If BAL is set to 1, the filter immediately goes into following and output O is set to the value of input BALREF. If the value at BALREF exceeds the output signal limits, the output is set to the limit value concerned. Return to dynamic state is bumpless.

Limitation Function

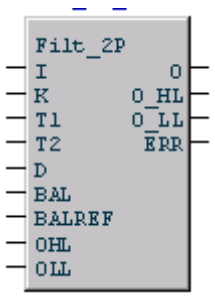
The limitation function limits the output signal to the limit values at input OHL for upper limit value, and input OLL for lower limit value.

If the actual value exceeds the upper limit value, the output O_EQ_HL is set to 1. If it falls below the lower limit value, the output O_EQ_LL is set to 1.

When the limitation status has been detected, a check is made each time the function block is executed to determine if K x I(t) exceeds the output signal limitations. If so, the limitation status remains. If not, the calculation of the output signal is performed by the algorithm in the normal way. Return from limitation to a dynamic state is bumpless. The function block checks that the upper limit value OHL is greater that the lower limit value OLL. If not, the output ERR is set to 1.

While the error status persists, the outputs O_EQ_HL, O_EQ_LL and O retain the values they had in the sample before, in which the error occurred. After an error, the return to a dynamic state is bumpless, in the same way as in the case above.

13.16 Filt_2P, Filter 2 Pole



Summary

Filt-2P (**Filt**er **- 2 P**ole) is used as a 2- pole low pass filter. The output signal can be limited with limit values specified at special inputs. The balancing function permits the output signal to follow an external reference and permits a bumpless return to the normal function. All transfers from static states are bumpless.

Call Name: Filt_2P

Parameter	Significance	Permissible values

Terminal Description

Name	Data Type	Description
I	Real	Input value.
K	Real	Input for Gain.
T1	Real	Input for 1/resonance angular frequency
T2	Real	Input for derivation time constant.
D	Real	Input for damping constant.
BAL	Bool	Balance.
BALREF	Real	Balance Reference.
OHL	Real	Output High Limit
OLL	Real	Output Low Limit
O	Real	Output
O_EQ_HL	Bool	Output Equal to High Limit
O_EQ_LL	Bool	Output Equal to Low Limit
ERR	Bool	Error

Function

The transfer function for a 2-pole low pass filter is:

$G(s)=K(1+sT1)/(1+s2DT2+(sT2)^2)$.

This has been implemented in the FILT-2P function block.

Gain, Filter Time, Damping Factor and Sampling Time

Certain constants are precalculated to make the execution time of the function block as short as possible. The result is stored internally in the function block. When recalculating, a test is performed to check if T1>2 x TS. If not, T1 is set equal to 2 x TS. In the same way a test is made to check if D<0.1, in which case D is set to 0.1.

T1 can be set to 0, thus making the FB a pure low pass filter.

Following

If BAL is set to 1, the filter immediately goes into following and output O is set to the value of input BALREF. If the value at BALREF exceeds the output signal limits, the output is set to the limit value concerned. Return to dynamic state is bumpless.

Limitation Function

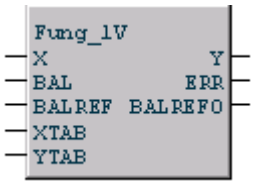
The limitation function limits the output signal to the limit values at input OHL for upper limit value, and input OLL for lower limit value.

If the actual value exceeds the upper limit value, the output O_EQ_HL is set to 1. If it falls below the lower limit value, the output O_EQ_LL is set to 1.

The Function Block checks that the upper limit value OHL is greater that the lower limit value OLL. If not, the output ERR is set to 1.

While the error status persists, the outputs O_EQ_HL, O_EQ_LL and O retain the values they had in the sample before, in which the error occurred. After an error, the return to a dynamic state is bumpless, in the same way as in the case above.

13.17 FUNG_1V, Function Generator with 1 Variable



Summary

Fung_1V (**Fun**ction **G**enerator - **1 V**ariable) is used for generation of an optional function of one variable, y=f(x).

The function is described by a number of co-ordinates. Linear interpolation is used for values between these coordinates. Maximum 16 coordinates can be specified.

Call FUNG-1V

Parameter	Significance	Permissible values

Terminal Description

Name	Data Type	Description
X	Real	Input for X value
BAL	Bool	Input for activation of balancing
BALREF	Real	Value which the output is to adopt with balancing.
XTAB	Real16	Groupdata for the X-table values.
YTAB	Real16	Groupdata for the Y-table values.
Y	Real	Output
ERR	Bool	Error, set (to 1) if X is outside the value of XTAB or if Y, on balancing, is outside the YTAB values.
BALREFO	Real	Output for calculated X-value with balancing.

Function

FUNG-1V 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 vectors XTAB and YTAB. For each X-value in XTAB, there is a corresponding Y-value in YTAB. The Y-value at the output is calculated by means of linear interpolation between the two X-values in XTAB which are nearest the value at the input X. The values in X-tab must be strictly increasing from low to high serial numbers in the table.

Interpolation

The function generated can be illustrated by the following figure. The interpolation is performed as follows:

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

Balancing

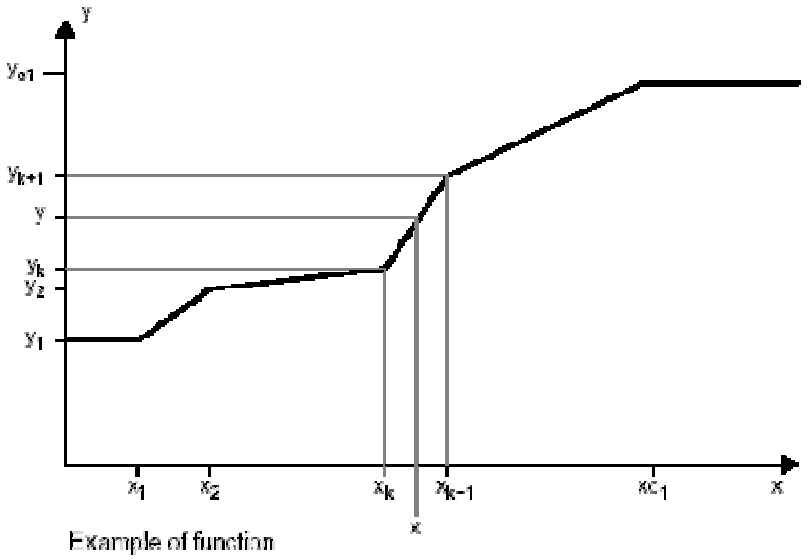
On activation of the balancing input BAL, the value at Y is set to the value at the input BALREF. The X-value which corresponds to this Y-value is obtained at the output BALREFO. On balancing, the X-value is calculated by interpolation in the same way as

the Y-value is calculated during normal operation. To permit the balancing, the values in YTAB must be strictly increasing or decreasing from low to high serial numbers in the table.

Error signal

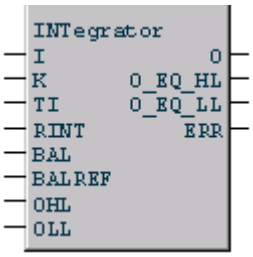
If the input signal X is outside the range defined by XTAB, the ERR output is set to 1. The Y-value is then set to the greatest or lowest value resp. in YTAB.

ERR is also set to 1 if BALREF is equal to or outside the YTAB value range when BAL is set to 1. The value at Y is then set to the value at the input BALREF and BALREFO is set to the greatest or lowest value resp. in XTAB.



Example of function

13.18 INTegrator, Integrator



Summary

INTegrator is used to give an integration effect. The output signal can be limited with limit values specified at special inputs. The balancing function permits the output signal to follow an external reference and permits a bumpless return to the normal function.

Call Name: INTegrator

Parameter	Significance	Permissible values

Terminal Description

Name	Data Type	Description
I	Real	Input.
K	Real	Gain.
TI	Time	Time constant for integration.
RINT	Bool	Reset Integrator.
BAL	Bool	Balance.
BALREF	Real	Balance Reference.
OHL	Real	Output High Limit
OLL	Real	Output Low Limit
O	Real	Output
O_EQ_HL	Bool	Output Equal to High Limit
O_EQ_LL	Bool	Output Equal to Low Limit
ERR	Bool	Error

Function, Transfer Function

The INTegrator function can be written in the time plane as O(t)=K/TI(òt I(t) d t). The main property when controlling is that the output signal retains its value when the input signal I(t)=0.The step response in the time plane is O(t)=k x I(t) x t/TI. The transfer function for an integrator is G(s)=K(1/sTI).

Gain, Integration Time Constant and Sampling Time

The constant K x TS/TI is precalculated and the result is stored internally in the function block. This constant is recalculated if TI or K is changed or if the sampling time TS is changed. When recalculating a test is made to see whether TS/TI<1.TS/TI is otherwise set equal to 1.

Clearing of Integrator

The algorithm is cleared when RINT goes to 1.

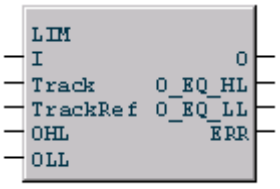
Following

If BAL is set to 1, the regulator immediately goes into following 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 limit value concerned. On return to the normal function, the value of output O during the last sample in following remains a further sample time, 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_EQ_HL is set to 1 and if it falls below the lower limit, the output O_EQ_LL is set to 1. The function block checks that the upper limit value OHL is greater than the lower limit value OLL. If this is not the case, the output ERR is set to 1. While the error status persists, the outputs O_EQ_HL, O_EQ_LL and O retain the values they had in the sample before that in which the error occurred. After limitation or error status, normal integration is performed from the current value.

13.19 LIM, Limiter



Summary

LIM (**LIM**iter) is used for limitation of real numbers.

Call Name: LIM

Parameter	Significance	Permissible values

Terminal Description

Name	Data Type	Description
I	Real	Input.
Track	Bool	Tracking.
TrackRef	Real	Input for tracking reference.
BAL	Bool	Balance.
BALREF	Real	Balance Reference.
OHL	Real	Output High Limit
OLL	Real	Output Low Limit
O	Real	Output
O_EQ_HL	Bool	Output Equal to High Limit
O_EQ_LL	Bool	Output Equal to Low Limit
ERR	Bool	Error

Function

LIM is used to limit a real value. Boolean output signals are given when the output is limited.

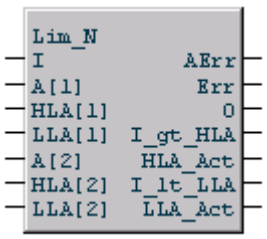
Limiting

When the input I exceeds the limit, the output O is limited to the limit value. One of the outputs O_EQ_HL or O_EQ_LL will then be set depending on which limit was exceeded.

Supervision of Limit Values

The function block checks that the limit value OHL is greater than the limit OLL. If OHL is less than OLL, the error signal output ERR is set. The output O and the limit value outputs IO_EQ_HL and O_EQ_LL are retained from the sample before that in which the error status developed.

13.20 Lim_N, Limiter



Summary

LIM_N (**LIM**iter **-** 1-of-**N** address) is used for limitation of real numbers. Several limit values can be selected.

Call Name: Lim_N

Parameter	Significance	Permissible values
N	Number of limits	1 – n

Terminal Description

Name	Data Type	Description
I	Real	Input value.
A[1]	Bool	Address 1.
HLA[1]	Real	High limit Address 1.
LLA[1]	Real	Low limit Address 1.
A[n]	Bool	Address 1.
HLA[n]	Real	High limit Address 1.
LLA[n]	Real	Low limit Address 1.
Aerr	Bool	Address error, is set if 2 or more inputs A[x] is set.
Err	Bool	Error, is set if limit for high level is less then low level.
O	Real	Output.
I_gt_HLA	Bool	Input > High Limit Address.
HLA_Act	Real	Actual High limit.
I_lt_LLA	Bool	Input < Low Limit Address.
LLA_Act	Real	Actual Low limit.

Function

LIM_N is used to limit up to n different limits. Boolean output signals are given when the output is limited.

Selection of Limit Value

Which of the limit value inputs HLA[1] – HLA[n] or LLA[1] – LLA[n] is to limit the value at output O is selected with the inputs A[1] – A[n]. If the input A[1] is 1, the output is limited by HLA[1] and LLA[1], if A[2] is 1, the output is limited by HLA[2] and LLA[2], and so on. If none of the inputs A[1] – A[n]is 1, O is limited to the data value 0. If 2 or more of the inputs A[1] – A[n] are set at the same time, the output is limited by the limit values corresponding to the lowest numbered set input. The error signal output AErr is set at the same time.

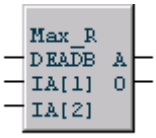
Limiting

When the input I exceeds the selected limit, the output O is limited to the limit value. One of the outputs I_gt_HLA or I_lt_LLA will then be set depending on which limit was exceeded. The value of the current limits for high and low level where limiting begins can be read at the outputs HLA_Act and LLA_Act.

Supervision of Limit Values

The function block checks that the limit value HLA is greater than the limit LLA. If HLA is less than LLA, the error signal output Err is set. The output O and the limit value outputs I_gt_HLA and I_lt_LLA are retained from the sample before that in which the error status developed.

13.21 Max_R, Maximum Selector



Summary

Max_R (**Max**imum selector **R**eal) is used to select the largest value of an optional number of real numbers.

Call Name: Max_R

Parameter	Significance	Permissible values
N	Number of inputs	1 – n

Terminal Description

Name	Data Type	Description
DEADB	Real	Deadband
IA[1]	Real	Input value 1.
IA[n]	Real	Input value n.
A	Dint	Address, output for the input having the greatest value.
O	Real	Output.

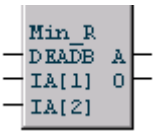
Function

The values at the inputs IA[1] – IA[n] are compared and the greatest value is obtained at the output O. The number of the input with the greatest value is obtained at the output A. If the two largest signal values are equal when the function block is executed the first time, the signal with the lowest connection number is selected as the greatest.

Deadband

The deadband specified at the input DEADB is symmetrical around the value of the greatest input. The upper and lower deadband limits are calculated from the value for the largest input in the preceding sample. To prevent rapid changes at the output A, the value at A is retained until the value at the corresponding input is less than the calculated lower deadband limit or until one of the other inputs exceeds the upper deadband limit.

13.22 Min_R, Minimum Selector



Summary

Min_R (**Min**imum selector **R**eal) is used to select the lowest value of an optional number of real numbers.

Call Name: Min_R

Parameter	Significance	Permissible values
N	Number of inputs	1 – n

Terminal Description

Name	Data Type	Description
DEADB	Real	Deadband
IA[1]	Real	Input value 1.
IA[n]	Real	Input value n.
A	Dint	Address, output for the input having the lowest value.
O	Real	Output.

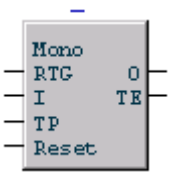
Function

The values at the inputs IA[1] – IA[n] are compared and the lowest value is obtained at the output O. The number of the input with the lowest value is obtained at the output A. If the two smallest signal values are equal when the function block is executed the first time, the signal with the lowest connection number is selected as the smallest.

Deadband

The deadband specified at the input DEADB is symmetrical around the value of the least input. The upper and lower deadband limits are calculated from the value for the lowest input in the preceding sample. To prevent rapid changes at the output A, the value at A is retained until the value at the corresponding input exceeds the calculated upper deadband limit or until one of the other inputs falls below the lower deadband limit.

13.23 Mono, Monostable



Summary

The mono function MONO can be used for time limiting of operation of outputs and automatic functions, impulse extension, stall alarm function, and so on.

Call Name: Mono

Parameter	Significance	Permissible values

Terminal Description

Name	Data Type	Description
RTG	Bool	Input for selection of if the Mono is to be retriggable or not.
I	Bool	Input.
Reset	Bool	Reset
TP	Time	Time Pulse.
O	Bool	Output.
TE	Time	Time Elapsed.

Function

A memory is set when the input I is set. The output then goes to 1, see function diagram. When the time set in the timer has elapsed, the memory is cleared and the output O goes to 0. The Reset input has priority over the I input. Reset = 1 clears the memory and the output goes to 0.

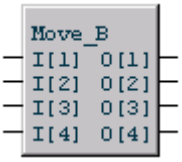
MONO Function, Not Retriggerable

If the input RTG is not set, a Mono function which is not retriggerable is obtained. If a new pulse is obtained at the input I before the time set in the timer has elapsed, it does not affect the timer. Only when the time set has elapsed and the output O is reset can the Mono function be restarted by the input I going from 0 to 1.

MONO Function, Retriggerable

If RTG is set, a retriggerable Mono function is obtained, that is, the timer starts from 0 each time a new pulse is obtained at the input I. If a new pulse is obtained at the input I before the time set in the timer has elapsed, the timer will restart, that is, the Mono function is retriggerable.

13.24 Move, Data copying



Summary

Move_<C1> copies the values at the input to the output and can be used to copy the value from one variable to another.

Call Name: Move_<C1>

Parameter	Significance	Permissible values
C1	Data Type	B, IL, R
N	Number of outputs	1 – n

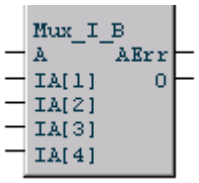
Terminal Description

Name	Data Type	Description
I[1]	C1	Input value 1.
I[2]	C1	Input value 2.
I[n]	C1	Input value n.
O[1]	C1	Output value 1.
O[2]	C1	Output value 2.
O[n]	C1	Output value n.

Function

The function block copies the values at the inputs to the respective output. The type of the data to be moved is determined with the call parameter C1 and the number of values moved is determined with the call parameter N.

13.25 Mux_I, Multiplexer with Integer Address



Summary

MUX_I (**Mu**ltiple**x**er **-** with **I**nteger address) is used as a selector and has an optional number of inputs. The data type can be boolean, integers or real numbers.

Call Name: Mux_I_<C1>

Parameter	Significance	Permissible values
C1	Data Type	B, IL, R
N	Number of inputs	1 – n

Terminal Description

Name	Data Type	Description
A	Dint	Address.
IA[1]	C1	Input value 1.
IA[2]	C1	Input value 2.
IA[n]	C1	Input value n.
Aerr	Bool	Address error, is set if address is larger than no of inputs or, negative.
O	C1	Output from selector.

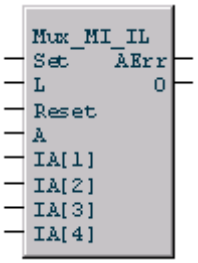
Function, Addressing

The input data value (IA[1] – IA[n]), which is to be connected to the output O, is specified with an address (integer 1 - n) at the input A. If the address is 0, the output terminal O is set to zero.

Supervision

The address A is monitored. If its value is negative or greater than the number of inputs, the error signal output AErr is set. The data value 0 is then obtained at the output.

13.26 Mux_MI, Multiplexer with Memory and Integer Address



Summary

Mux_MI (**Mu**ltiple**x**er **-** with **M**emory and **I**nteger address) is used as a selector with memory function and has an optional number of inputs. The data type can be integers or real numbers.

Call Name: Mux_MI_<C1>

Parameter	Significance	Permissible values
C1	Data Type	IL, R
N	Number of inputs	1 – n

Terminal Description

Name	Data Type	Description
Set	Bool	Input for loading of new value each time the function block is executed.
L	Bool	Load. Dynamic input for loading of data.
Reset	Bool	Input for clearing of data. Reset overrides Set and L.
A	Dint	Address.
IA[1]	C1	Input value 1.
IA[2]	C1	Input value 2.
IA[n]	C1	Input value n.
Aerr	Bool	Address error, is set if address is larger than no of inputs or, negative.
O	C1	Output from selector.

Function, Addressing

The input data value (IA[1] – IA[n]) that is to be stored at the output is specified with an address (integer 1 - n) at the input A. If the address is 0, the data value 0 is stored.

Loading

The value is loaded at the instant the input L is set to 1. If input Set is set, new data is loaded each time the function block is executed. When Set is reset after having been set, the data loaded most recently remains until the function block is once more executed with input Set, L or Reset set. Input Set overrides input L, that is, when Set is set, L has no function.

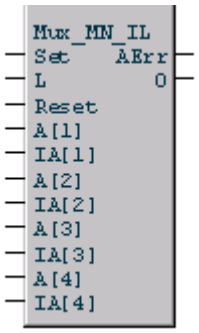
Clearing

The input Reset clears the data output and prevents further storage of data.

Supervision

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

13.27 Mux_MN, Multiplexer with Memory and 1 of N Address



Summary

Mux_MN (**Mu**ltiple**x**er **-** with **M**emory and 1-of-**N** address) is used as a selector with memory function. MUX-MN has an optional number of inputs. The data type can be integers or real numbers.

Call Name: Mux_MN_<C1>

Parameter	Significance	Permissible values
C1	Data Type	IL, R
N	Number of inputs	1 – n

Terminal Description

Name	Data Type	Description
Set	Bool	Input for loading of new value each time the function block is executed.
L	Bool	Load. Dynamic input for loading of data.
Reset	Bool	Input for clearing of data. Reset overrides Set and L.
A[1]	Bool	Address 1.
IA[1]	C1	Input value 1.
A[2]	Bool	Address 2.
IA[2]	C1	Input value 2.
A[n]	Bool	Address n.
IA[n]	C1	Input value n.
Aerr	Bool	Address error, is set if address is larger than no of inputs or, negative.
O	C1	Output from selector.

Function, Addressing

The input data value (A[1] – IA[n]) that is to be stored at the output is specified with the inputs A[1] – A[n]. If the input A1 is 1, the value from input IA[1] is stored at the output, if A[2] is 1, the value from input IA2, and so on. If none of the inputs A[1] – A[n] is set to 1, the data value 0 is stored at the output O.

Loading

The value is loaded at the instant the input L is set to 1. If input Set is set, new data is loaded each time the function block is executed. When Set is reset after having been set,

the data loaded most recently remains. Input Set overrides input L, that is, when Set is set, L has no function.

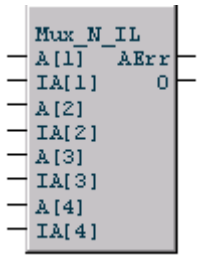
Clearing

The input Reset clears the data output and prevents further storage of data.

Supervision

If two or more of the inputs A[1] – A[n] are set simultaneously, the output AErr is set and the value from the input which correspond to the address input set with the lowest number is stored at the output O.

13.28 Mux_N, Multiplexer with 1 of N Address



Summary

Mux_N (**Mu**ltiple**x**er **-** with 1-of-**N** address) is used as a selector and has an optional number of inputs. The data type can be integers or real numbers.

Call Name: Mux_N_<C1>

Parameter	Significance	Permissible values
C1	Data Type	IL, R
N	Number of inputs	1 – n

Terminal Description

Name	Data Type	Description
A[1]	Bool	Address 1.
IA[1]	C1	Input value 1.
A[2]	Bool	Address 2.
IA[2]	C1	Input value 2.
A[n]	Bool	Address n.
IA[n]	C1	Input value n.
Aerr	Bool	Address error, is set if address is larger than no of inputs or, negative.
O	C1	Output from selector.

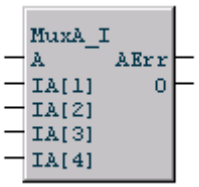
Function, Addressing

Which one of the inputs IA[1] – IA[n] is to be connected to the output O is selected with the inputs A[1] – A[n]. If input A1 is set, input IA[1] is connected to the output and if A[2] is set, input IA[2] is connected, and so on. If none of the inputs A[1] – A[n] is set, the data value 0 is connected to the output O.

Supervision

If two or more of the inputs A[1] – A[n] are set at the same time, the output AErr is set and the input which corresponds to the address input set with the lowest number is connected to the output O.

13.29 MuxA_I, Multiplexer for Text



Summary

MuxA_I (**Mu**ltiple**x**er **A**rray **-** with **I**nteger address) is used as a selector for text arrays and has an optional number of inputs. The number of characters in the text arrays is optional.

Call Name: MuxA_I

Parameter	Significance	Permissible values
N	Number of inputs	1 – n

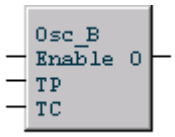
Terminal Description

Name	Data Type	Description
A	DInt	Address, specifies what input (IA) is connected to the output.
IA[1]	C1	Input value 1.
IA[2]	C1	Input value 2.
IA[n]	C1	Input value n.
Aerr	Bool	Address error, is set if address is larger than no of inputs or, negative.
O	C1	Output from selector.

Function

The function block takes address given at the input A and stores data from the corresponding input IA[x] at output O. If the address is greater than the number of inputs or if it is negative, the output AErr is set to 1. In addition a blank text string is stored on the output. If A is set to 0, a blank text string is issued.

13.30 Osc_B, Oscillator



Summary

Oscillator Osc_B (**Osc**illator **- B**oolean variables) with variable frequency and pulse time is used when pulse trains with periods from 2 x cycle time up to 24 hours are needed.

Call Name: Comp_I

Parameter	Significance	Permissible values

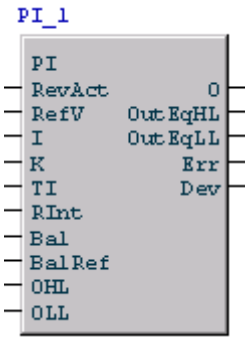
Terminal Description

Name	Data Type	Description
Enable	Bool	Enable, input for start of oscillator.
TP	Time	Time Pulse.
TC	Time	Time Cycle.
O	Bool	Output.

Function

The oscillator starts when the input Enable is set (to 1). TC and TP are rounded up to a multiple of the actual cycle time.

13.31 PI, Proportional Integrating regulator



Summary

PI (**P**roportional **I**ntegrating regulator) is used as a standard PI- regulator for serial compensation in feed back systems. The control deviation is calculated internally in the element. The output signal can be limited to limits specified at special inputs. The balancing function permits the output signal to follow an external reference and permits a bumpless return to normal function. All transfers from balancing or limited output signal are bumpless.

Call Name: PI

Parameter	Significance	Permissible values

Terminal Description

Name	Data Type	Description
RevAct	Real	Reverse Control Action.
RefV	Real	Reference Value (setpoint) .
I	Real	Input (Measured Value).
K	Real	Gain.
TI	Real	Time Integrationin seconds.
Rint	Bool	Reset Integrator.
Bal	Bool	Balance.
BalRef	Real	Balance Reference.
OHL	Real	Output High Limit
OLL	Real	Output Low Limit
O	Real	Output
Out_EqHL	Bool	Output Equal to High Limit
Out_EqLL	Bool	Output Equal to Low Limit
ERR	Bool	Error. Output is set if OHL is less than OLL.Input value.
Dev	Real	Deviation

Function

Control Deviation

Dev, the control deviation, is calculated as follows even when the regulator is following:
Dev = I – RefV

Reverse Control Action

RevAct is a function parameter which controls the direction of the output signal change in relation to the actual value. If RevAct is set to 0, direct action is obtained, that is, increasing actual value gives increasing output signal. When RevAct is set to 1, reversed action is obtained, that is, increasing actual value gives decreasing output signal.

Transfer Function

The transfer function for a PI function is:

$$G(s) = K\left(1 + \frac{1}{s \cdot T\,1}\right)$$

This has been implemented in the PI element.

Gain, Integration Time Constant and Sampling Time

The PI algorithm in the software performs the following tests:

If $\frac{TS}{TI} > 1$, then $\frac{TS}{TI}$ is set = 1

TS = the sampling time for the controller.

Clearing of Integrator

The integration section of the algorithm is cleared when RInt goes to 1. If the proportional section of the algorithm exceeds the limits of the output signal, the limit status remains and the internal status is updated in accordance with the section "Bumpless transfer from following or limitation". When RInt is set permanently to 1, the element functions as a P-regulator.

Following

If BaI is set to 1, the regulator immediately goes into following and the output O follows the value of the input BalRef. If the value at BalRef exceeds the output signal limits, the output is set to the limit value concerned. On return to a normal function the value of the output O during the last sample in the following remains during one sample time. See below under "Bumpless transfer from following or limiting".

Limitation Function

The limitation function limits the output signal to the limit values at the inputs OHL for upper limit value and OLL for the lower limit value. If the actual value exceeds the upper limit value, the output Out_EqHL is set to 1 and if it falls below the lower limit value, the output Out_EqLL is set to 1. The element checks that the upper limit value OHL is greater

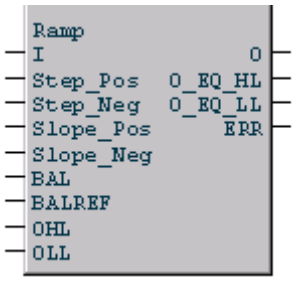
than the lower limit value OLL. If not, the output Err is set to 1. While the error status persists, the outputs Out_EqHL, Out_EqLL and O retain the values they had in the sample before, in which the error occurred.

Bumpless Transfer from Following or Limitation

Transfer from following status (Bal=1) or from a limited output signal is bumpless. This is performed by recalculation of internal states, that is, the integration part according to

INT(t)=O(t) - K x DEV(t).

13.32 Ramp, Ramp Generator



Summary

Ramp (**Ramp** generator) is used to limit the speed of change of a signal. The output signal can be limited with limit values specified at special inputs. The balancing function permits the output signal to follow an external reference.

Call Name: Ramp

Parameter	Significance	Permissible values

Terminal Description

Name	Data Type	Description
I	Real	Input.
Step_Pos	Real	The greatest allowed positive step change .
Step_Neg	Real	The greatest allowed negative step change .
Slope_Pos	Real	Positive ramp for the output. Must be a positive number.
Slope_Neg	Real	Negative ramp for the output. Must be a positive number.
BAL	Bool	Balance.
BALREF	Real	Balance Reference.
OHL	Real	Output High Limit
OLL	Real	Output Low Limit
O	Real	Output
O_EQ_HL	Bool	Output Equal to High Limit
O_EQ_LL	Bool	Output Equal to Low Limit
ERR	Bool	Error. Output is set if OHL is less than OLL.Input value.

Function

The main property of a Ramp function is that the output signal follows the input signal while the input signal is not changed more than the value specified at the step inputs. If the input signal is changed more than so, the output signal is first changed by Step_Pos or Step_Neg depending on the direction of change and then by Slope_Pos or Slope_Neg per second until the values at the input and output are equal. This means also that if Step_Neg = Step_Pos = 0, a pure Ramp, that is, Slope_x, is obtained at the output.

The greatest step change allowed at the output O is specified by the parameters Step_Pos and Step_Neg for the respective direction of change. The ramp which the output signal is to follow if the change at the input I exceeds Step_Pos or Step_Neg is

specified by the inputs Slope_Pos and Slope_Neg. All parameters are specified as absolute values with the same units as input I.

The values of the parameters are stored internally in the function block. New values are only entered under stationary conditions, that is, when I(t)=O(t).

Calculation of Output Signal

When calculating output signal O, three cases must be distinguished:

• I(t) = O(t-TS) results in O(t) = I(t)

• I(t) > O(t-TS) results in calculation of the value of the output in accordance with:

  O(t) = min(I(t)) and (Step_Pos + VPOS(t–TS) + Slope_Pos x TS), that is, the least of the values of I(t) and (Step_Pos + VPOS(t–TS) + Slope_Neg x TS)

• I(t) < O(t–TS) results in calculation of the value of the output in accordance with:

  O(t) = max(I(t)) and (VNEG (t-TS) – Step_Neg– Slope_Neg x TS) , that is, the largest of the values of I(t) and (VNEG (t-TS) – Step_Neg– Slope_Neg x TS).

VPOS and VNEG are auxiliary variables with positive and negative steps respectively. These are calculated in all cases according to:

• VPOS(t) = min(O(t), VPOS(t–TS) + Slope_Pos x TS)

• VNEG(t) = max(O(t), VNEG(t–TS)–Slope_Neg x TS).

If a new step with the same direction of change appears at the input before the internal auxiliary signal has been updated the step part of the output signal is reduced. A step with the opposite derivative, however, takes full effect. This is because the auxiliary variable for the opposite direction always has been updated to O(t).

Following

If BAL is set to 1, the filter immediately goes into following 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 limit value concerned.
During following VPOS(t) = VNEG(t) = O(t) = BALREF(t).
Return to normal function is done as if a unit step had occurred on the input.

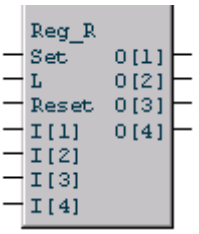
Limitation Function

The limitation function limits the output signal to the limit 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_EQ_HL is set to 1 and if it falls below the lower limit, the output O_EQ_LL is set to 1. In the limiting state VPOS(t), VNEG(t) and O(t) are set to the limit value concerned.

The function 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_EQ_HL, O_EQ_LL and O retain the values they had in the sample before that in which the error occurred.

13.33 Reg, Register



Summary

Register Reg (**Reg**ister) is used as a memory function block with an optional number of positions. The value of the data can be integer, real number.

Call Name: Reg_<C1>

Parameter	Significance	Permissible values
C1	Data Type	IL, R
N	Number of places	1 – n

Terminal Description

Name	Data Type	Description
Set	Bool	Input for loading of new value each time the function block is executed.
L	Bool	Load. Dynamic input for loading of data.
Reset	Bool	Input for clearing of data. Reset overrides Set and L.
I[1]	C1	Input data to position 1.
I[2]	C1	Input data to position 2.
I[n]	C1	Input data to position n.
O[1]	C1	Output data from position 1.
O[2]	C1	Output data from position 2.
O[n]	C1	Output data from position n.

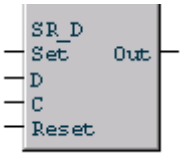
Function

When input L becomes 1, the register is loaded with data from inputs I[1] – I[n]. Data previously in the register is replaced. If input Set is set, loading is performed as above each time the register is executed. When Set is reset after having been set, the data most recently loaded remains until the function block is executed again, with input Set, L or Reset set. Input Set overrides L so that when input Set is set, L has no effect.

Clearing

When input Reset is set, the register is cleared and all further entry is prevented. Reset overrides both Set and L.

13.34 SR_D, Set/Reset Memory with Data Input



Summary

SR_D (**S**et **R**eset memory - **D**ata input) is used as a memory for Boolean variables. Besides the SR-function, it can also clock in data.

Call Name: SR_D

Parameter	Significance	Permissible values

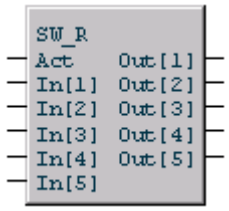
Terminal Description

Name	Data Type	Description
Set	Bool	Set Input.
D	Bool	Data input.
C	Bool	Clock, Dynamic input for entry of data from input D.
Reset	Bool	Reset, overrides all other inputs.
Out	Bool	Output value .

Function

If only the Set and Reset inputs are used, SR_D functions as an ordinary SR function block. When input Reset is reset and input C goes to 1, the value at input D is stored at the output. When input Reset is set, the output is unconditionally reset, that is, Reset overrides the other inputs.

13.35 SW, Switch



Summary

SW (**Sw**itch) is used as a connection element for data and has an optional number of channels with closing function.

Call Name: SW_<C1>

Parameter	Significance	Permissible values
C1	Data Type	B, R
N	Number of places	1 – n

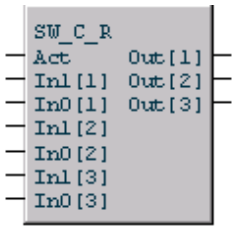
Terminal Description

Name	Data Type	Description
Act	Bool	Set Input.
In[1]	<C1>	Input data to channel 1.
Out[1]	<C1>	Output data from channel 1.
In[n]	<C1>	Input data to channel n.
Out[n]	<C1>	Output data from channel n.

Function

When the control input Act is 0, the output data is 0. When Act is set, data comes from the inputs In[1] – In[n].

13.36 SW_C, Changeover Switch



Summary

SW_C (**SW**itch **– C**hangeover) is used as a connection element for data and has an optional number of channels with switching function.

Call Name: SW_C<C1>

Parameter	Significance	Permissible values
C1	Data Type	B, R
N	Number of places	1 – n

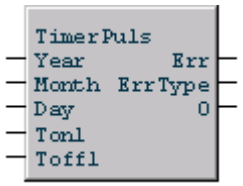
Terminal Description

Name	Data Type	Description
Act	Bool	Set Input.
In1[1]	<C1>	Input data to channel 1, which is connected to the output for channel 1 when the switch is activated.
In0[1]	<C1>	Input data to channel 1, which is connected to the output for channel 1 when the switch is deactivated.
Out[1]	<C1>	Output data from channel 1.
In1[n]	<C1>	Input data to channel 1, which is connected to the output for channel 1 when the switch is activated.
In0[n]	<C1>	Input data to channel 1, which is connected to the output for channel 1 when the switch is deactivated.
Out[n]	<C1>	Output data from channel n.

Function

When the control input Act is 0, the data from the inputs In0[1] – in0[n] are connected to the appropriate outputs. When Act is set, data comes from the inputs In1[1] – In1[n].

13.37 TimerPuls, Time Selector



Summary

TimerPuls is used to generate a pulse with a duration determined by two times in the same day (24 hour).

Call Name: TimerPuls

Parameter	Significance	Permissible values

Terminal Description

Name	Data Type	Description
Year	Dint	Specifies which year the pulse is to be given. "0" means every year.
Month	Dint	Specifies which month the pulse is to be given. "0" means every month.
Day	Dint	Specifies which day the pulse is to be given. "0" means every day.
Ton1	Time	Specifies which time the pulse starts.
Toff1	Time	Specifies which time the pulse stops.
Err	Bool	Error. Is set if any of the values are set outside the permitted range.
ErrType	String[12]	Shows the actual error.
O	Bool	Output.

Function

TimerPuls is used to produce a pulse between two times within one (24 hours) day. The day is specified by year, month and day number within the month. The duration of the pulse is determined by the inputs Ton1 and Toff1. The generated pulse can be used to start required functions.

YEAR Input

The year during which the pulse is to be obtained is specified at the input Year. If the year is given as 0, a pulse is given each year.

Month Input

The month during which the pulse is to be obtained is specified at the input Month. If the month is given as 0, a pulse is given each month. In addition to 0, month numbers 1 to 12 can be given.

DAY Input

The day of the month during which the pulse is to be obtained is specified at the input Day. If the day is given as 0, a pulse is given each day of the month. In addition to 0, day numbers 1 to 31 can be given.

TON and TOFF Inputs

At which time a pulse is to begin or end respectively is given at the inputs Ton1 and Toff1. Ton1 and Toff1 must be within the same 24 hour period, that is, Ton1<Toff1. Time specifications can vary from 00:00:00.000 to 23:59:59.999.

Examples of Selection of Day

Year	Month	Day	Significance
0	0	0	Each day.
0	0	1 – 31	A certain day of each month, every year.
0	1 - 12	0	Each day during a certain month every year.
0	1 - 12	1 - 31	A certain day of each year.
1980 – 2099	0	0	Each day during a certain year.
1980 - 2099	0	1 - 31	A certain day each month during a certain year.
1980 – 2099	1 – 12	0	Each day during a certain month during a certain year.
1980 - 2099	1 - 12	1 - 31	A particular day.

Supervision

If any input has a value outside the permitted range, the error signal Err is set (to 1) and the type of error is given at output ErrType.

13.38 Tresh_L, Treshold Function

Summary

Thresh_L (**Thresh**old **- L**ogic) is used to determine when more than, or equal to, a given number of Boolean signals are set (to 1).

Call Name: Tresh_L

Parameter	Significance	Permissible values
N	Number of inputs	1 – n

Terminal Description

Name	Data Type	Description
C1	Dint	Treshold Value
I[1]	Bool	Input for first boolean value.
I[2]	Bool	Input for first boolean value.
I[n]	Bool	Input for first boolean value.
O	Bool	Output.

Function

The Tresh_L function block is used to determine if more than, or equal to, a specified number of Boolean values in a group are set to 1. If so, the output O is set.

REVISION

Rev.	Page (P) Chapt. (C)	Description	Date Dept./Init.
A		PP_Lib version 1.3	02-11-22/MP
B		Type Real16 described	2002-11-21 RSTV/HL
C		PP_Lib version 2.0	03-03-07/MP
D		PP_Lib version 2.0/1, Mono changed	03-10-13/MP
E		TimePuls element. The day is defined as 0 – 31	08-09-23/BP