Application Library

OEN_BaseBlocks 5.00.11

Sysmac Function Block Library
for General use

User's Manual

Contents

Introduction 4
Functions and FunctionBlocks 8
1. Alarms 9
2. GroupAlarms 13
3. AnaScaling 15
4. ASCIItoHEX 18
5. BinSelect 20
6. BitControl 22
7. BitDataMonitor 23
8. BOOL_To_UINT 25
9. UINT_To_BOOL 26
10. CountDownTimer 27
11. CPU_System 28
12. DateAndTime 30
13. DevicePort_Setup 32
14. DST_TimeChange 34
15. EC_Control 35
16. EIP_Control 38
17. FahrenheitToCelsius 40
18. FIFOStack 41
19. GroupTrigger 44
20. HasChanged 46
21. InchTomm 48
22. JoyStick_Control 49
23. Loop 51
24. LowPassFilter_IIR 53
25. MeanMaxMin 55
26. MovingHolidays 57
27. NX_ECC_Status 59
28. NX_RS1201 61
29. OS32C_DM 64
30. OunceToGrams 65
31. PRM21_Control 66
32. psiTobar 70
33. PulseConverter 71
34. PulseGenerator 73
35. PWM 74
36. RealTo 76
37. SCU_PortSetup 77
38. SD_Logging10Col 81
39. Sequencer 83
40. Stack_Integers 85
41. Stack_Reals 87
42. SummationCounter 89
43. SunsetAndSunrise 90
44. Swap4Bytes 92
45. Table_Logging10Col 94
46. Tank_Volume 96
47. TCP_Connect 98
48. TCP_Messaging 100
49. TimeDateSwitch 102
50. TimeToTIME 105

51. Tx16bitToWORD 107
52. Tx32bitToDWORD 109
53. Tx4bitTo4bit 111
54. Tx6bitTo6bit 113
55. Tx8bitTo8bit 114
56. Tx8bitToByte 116
57. TxByteTo8bit 118
58. TxDWORDTo32bit 120
59. TxWORDTo16bit 122
60. UINT_To_BOOL 124
61. USGallonsToLiters 125
62. ValueSelect 126
63. WordsTo 128
64. WordsToDINT 130
65. ZoneCheck 131
66. Template .... **Error! Bookmark not defined.**

Introduction

Thank you for using the Application Library: **OEN_BaseBlocks**

Use it when programming with the automation software Sysmac Studio.

This manual contains information that is necessary to use the Library with Sysmac Studio.

Hereinafter, the function blocks are described as FB, functions as FNs.

1.1. Notice

This manual describes the necessary information to use the Application Library. Refer also to the user's manuals for Application Library, the *Sysmac Studio Version1 Operation Manual* (Cat.No. W504)

Please read and understand this manual before using the Library. Keep this manual in a safe place where it will be available for reference during operation.

1.2. Terms and Conditions Agreement

1 NO WARRANTY

1）The functions and function block Library is distributed as a sample in the hope that it will be useful, but without any warranty. It is provided “as is” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the function block is with you. Should the function block prove defective, you assume the cost of all necessary servicing, repair or correction.

2）In no event unless required by applicable law the author will be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use the function block (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a failure of the function block to operate with any other programs), even if the author has been advised of the possibility of such damages.

2 LIMITATION OF LIABILITY

1) OMRON SHALL HAVE NO LIABILITY FOR DEFECT OF THE SOFTWARE.

2) OMRON SHALL HAVE NO LIABILITY FOR SOFTWARE PARTS DEVELOPED BY THE USER OR ANY THIRD PARTY USING THE FUNCTION BLOCK DESCRIBED ON THIS MANUAL.

3 APPLICABLE CONDITIONS

USER SHALL NOT USE THE SOFTWARE FOR THE PURPOSE THAT IS NOT PROVIDED IN THE ATTACHED USER MANUAL.

4 CHANGE IN SPECIFICATION

The software specifications and accessories may be changed at any time based on improvements and other reasons.

5 ERRORS AND OMISSIONS

The information in this manual has been carefully checked and is believed to be accurate; however, no responsibility is assumed for clerical, typographical, or proofreading errors, or omissions.

1.3. Safety Precautions

Definition of Precautionary Information

The following notation is used in this manual to provide precautions required to ensure safe usage of OEN_BaseBlocks Library.
The safety precautions that are provided are extremely important to safety. Always read and heed the information provided in all safety precautions.

The following notation is used.

⚠ WARNING	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury. Additionally, there may be severe property damage.

⚠ Caution	Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury, or property damage.

Precautions for Safe Use

Indicates precautions on what to do and what not to do to ensure safe usage of the product.

Precautions for Correct Use

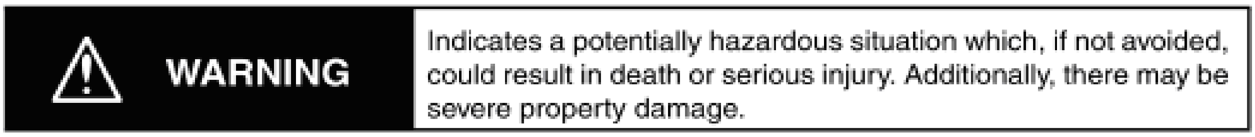
Indicates precautions on what to do and what not to do to ensure proper operation and performance.

Additional Information

Additional information to read as required.
This information is provided to increase understanding or make operation easier.

⚠	The triangle symbol indicates precautions (including warnings). The specific operation is shown in the triangle and explained in text. This example indicates a general precaution.
❗	The filled circle symbol indicates operations that you must do. The specific operation is shown in the circle and explained in text. This example shows a general precaution for something that you must do.

Warning list



⚠ WARNING

Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury. Additionally, there may be severe property damage.

Emergency stop circuits, interlock circuits, hardware limit and similar safety measures must be provided in external control circuits.	⚠
Using this FB in a device, confirm that the program and FB operate properly. Design a program so that safety measures such as fail-safe circuits are implemented outside of the FB	⚠

Caution list

⚠ Caution

Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury, or property damage.

Confirming an operation of the control program, including this FB. Trial operation such as the concerned motor runs in low velocity is recommended.	⚠
Performing adjustment of the device controlled by the program with this FB, secure the safety of the machine.	⚠
Do not use this FB for the system with devices and versions not specified in this document. To use, contact your OMRON representative	⚠
If a Task Period Exceeded Error occurred by executing this FB, the CPU Unit shifts to an error state. Make sure to set the execution task period to an appropriate value by referring to the execution time of this FB.	⚠
Do not delete the instances from the program with online editing during an execution of this FB. Program communication will stop in error.	⚠
Make sure to set the input parameters of this FB appropriately in accordance with the actual device. Make settings as described in this manual.	⚠

Functions and FunctionBlocks

Applications

The **OEN_BaseBlocks** is a set of functions and function blocks for general use. If not notified, these function blocks are compatible with all Sysmac series PLCs.

Library Change Log

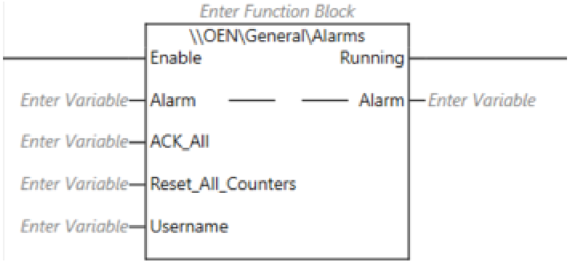
See more details on each Function/FunctionBlock	
5.00.9	
5.00.10	DateAndTime
5.00.11	CPU_System, ReadIPAddress, TCP_Connect

1. *Alarms*

This is a function block that uses an Alarm array to keep track of the status of alarms. The same array is used to delay, limit values, acknowledge and disable. See more about this in the table below. The alarm array is easily connected to the alarm function in the NA. It can handle several hundred alarms in a loop.

Best practice is to define an array of FB instance and then loop the FB using FOR/NEXT.

1.1. FB Layout



1.2. Input Variables

Name	Data type	Description
Enable	BOOL	Enable functionblock
ACK_All	BOOL	Acknowledge all active alarms. Set to TRUE to Autoacknowledge
Reset_All_Counters	BOOL	Reset all alarm counters
Username	STRING[16]	User that Acknowledged

1.3. In-Out Variables

Name	Data type	Valid Range	Description
Alarm	OEN/nGeneral/sAlarms	ARRAY[*] OF	Alarm structure. Use an array here with size same as number of alarms

1.4. Output Variables

Name	Data Type	Description
Running	BOOL	Enable is TRUE

1.5. Table Settings

Settings for each alarm	
ShortName	Text to identify alarm
Group	Group number (1-99) when using **GroupAlarms** Function
HighHigh_Limit	HighHigh Analog Level.
High_Limit	High Analog Level
Low_Limit	Low Analog Level
LowLow_Limit	LowLow Analog Level
Delay	Delay before setting alarm

1.6. Table Inputs

Activate these from your program - Numbers in ( ) are bit positions in Status	
AnalogInput	Analog value to observe (Use either Analog or Digital)
DigitalInput (0)	Digital input to observe (Use either Analog or Digital)
Disabled (15)	Alarm disabled/Not in use
ACK	Ackowledge
Reset_Counter	Reset of alarm counter
ACK_Username	Name of user that ACKed

1.7. Table Outputs

Status for each alarm - Numbers in ( ) are bit positions	
ActiveNoACK (8)	Alarm is active and not ACKed from ACK (ON until ACK)
NormalNoACK (10)	Signal is back to Normal but Alarm not ACKed
ActiveAndACK (9)	Alarm is ACKed but still active
DigitalAlarm (1)	The alarm is from a Digital Input
GroupDigital_Alarm (12)	DigitalAlarm with OFF_ON Delayed Trigger for each alarm
HighHigh_Alarm (4)	AnalogInput >= HighHigh Analog Level. Delayed.
High_Alarm (5)	AnalogInput >= High Analog Level. Delayed.
Low_Alarm (6)	AnalogInput <= Low Analog Level. Delayed.
LowLow_Alarm (7)	AnalogInput <=LowLow Analog Level. Delayed
Counter	Counting number of alarm occurences
Alarm_TimeStamp	When alarm was set
CriticalAlarm_TimeStamp	When HighHigh or LowLow was set
ACK_TimeStamp	When ACKed
Inactive_TimeStamp	When signal went back to Normal
Status	Collection of alarm status signals. See bitno in ( ) in this table

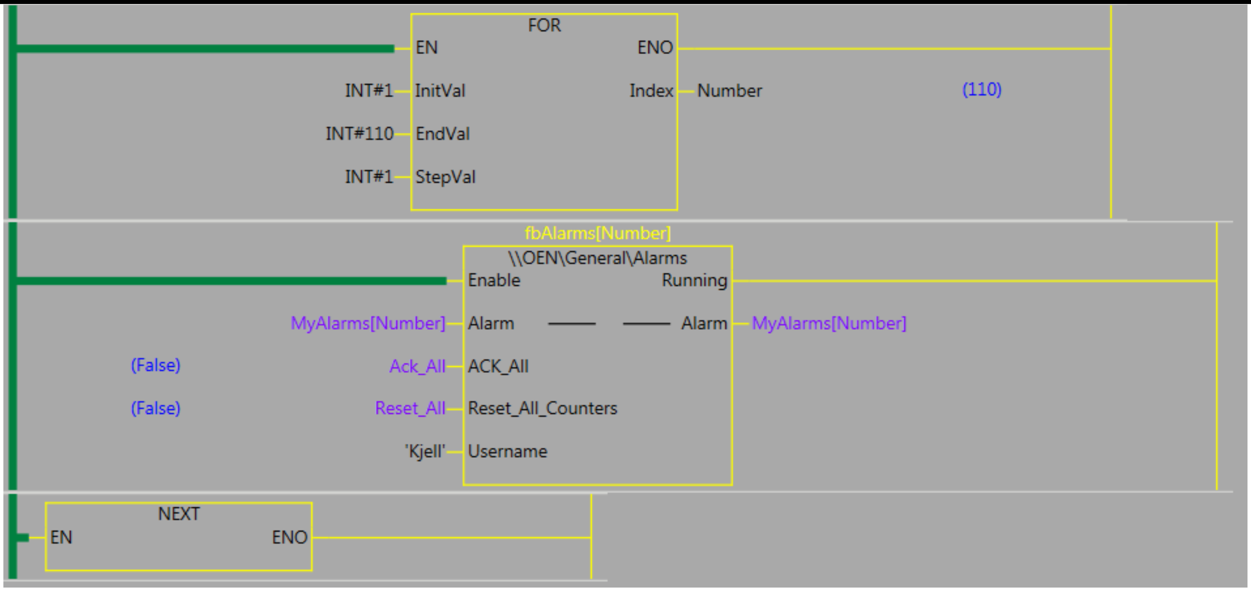
1.8. Revisions

Revision	In Library	Correction
2.2.0	5.00.9	

1.9. Credits

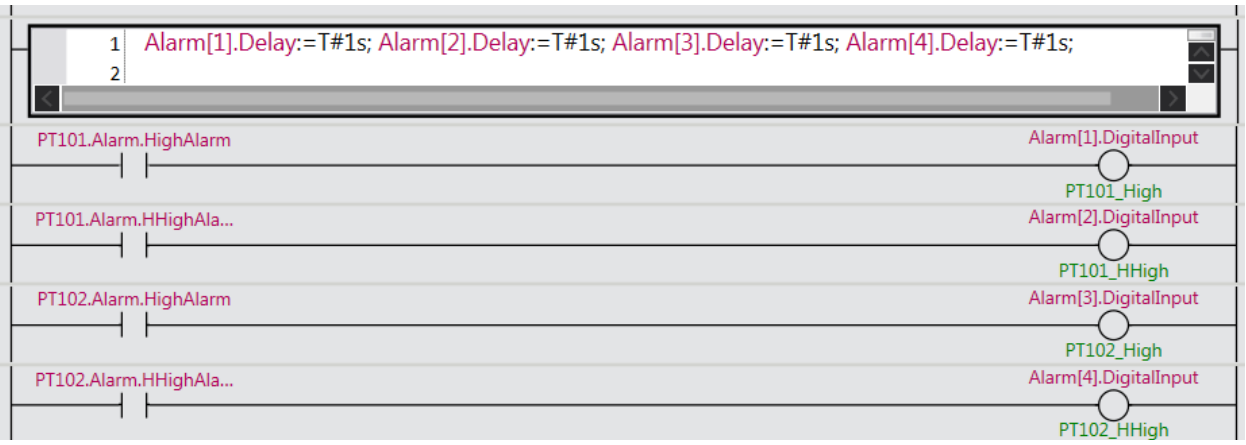
	Name
Omron - Norway	Kjell Baardsgaard

1.10. Example



1.10.1. Digital Alarm

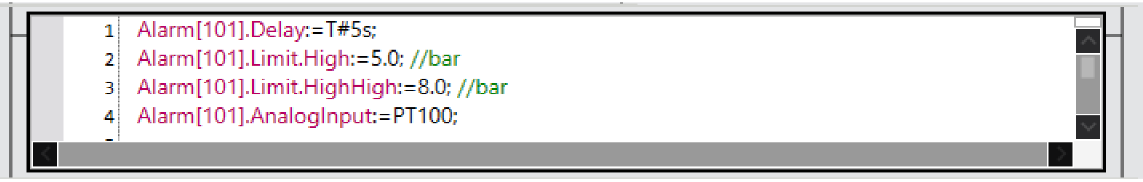
Rung#1 is the Alarm Settings for Alarm 1-4.
Rung#2-5 is the alarms that triggers DigitalInput in the Alarm table. Alternatively, the PTs could connect to AnalogInput and High/HighHigh be triggered from there. See 3.7.2.



```
Alarm[1].Delay:=T#1s; Alarm[2].Delay:=T#1s; Alarm[3].Delay:=T#1s; Alarm[4].Delay:=T#1s;
```

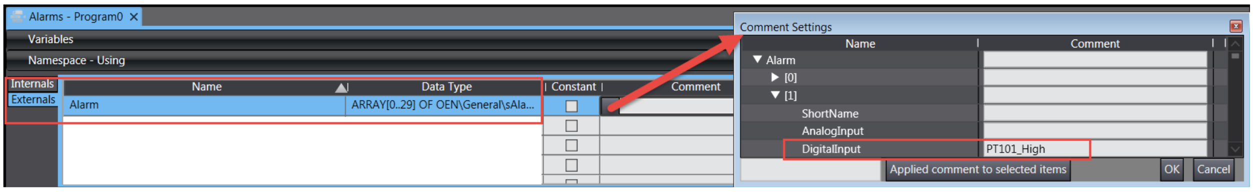
1.10.2. Analog alarm

Settings for the analog alarm



```
Alarm[101].Delay:=T#5s;
Alarm[101].Limit.High:=5.0; //bar
Alarm[101].Limit.HighHigh:=8.0; //bar
Alarm[101].AnalogInput:=PT100;
```

This means that you do not have to make separate alarms for all 4 alarm levels such as in the example of digital alarms.



Alarms - Program0

Variables

Namespace - Using

Internals
Externals

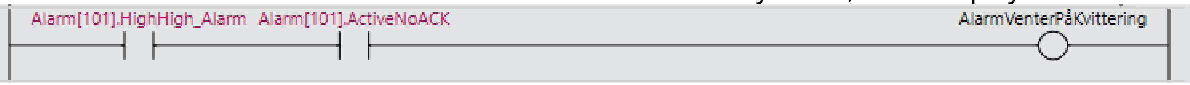
Name	Data Type	Constant	Comment
Alarm	ARRAY[0..29] OF OEN\General\sAla...	□	

Comment Settings

Name	Comment
▼ Alarm	
▶ [0]	
▼ [1]	
ShortName	
AnalogInput	
DigitalInput	PT101_High

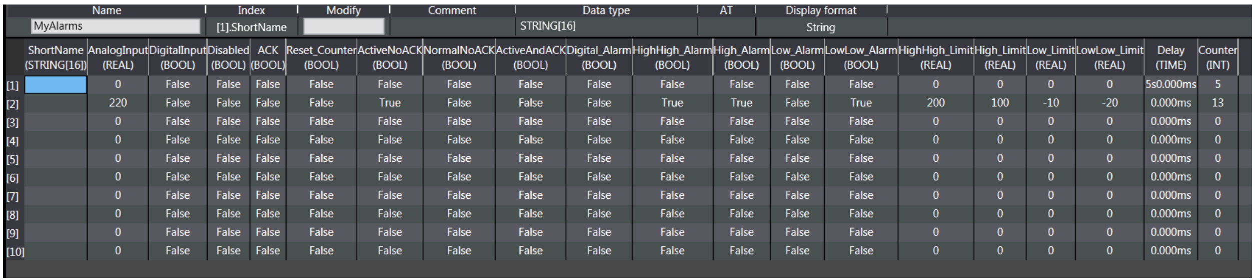
The green comment text on the alarms can be entered via Comment in the External Variable List Or by clicking under the output in ladder.

Alarm status is extracted from the Alarm Structure and directly in NA, or for display in Ladder.



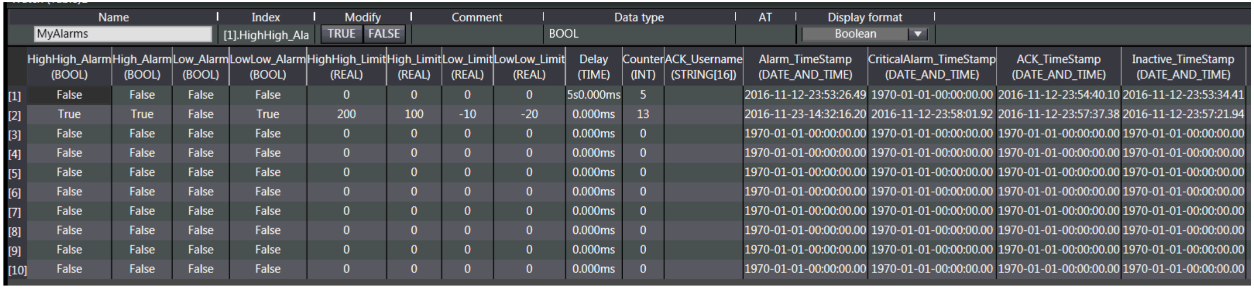
In the NA, there will now only be a single variable on the variable list to deal with. Neat and simple.

1.10.3. Table view



Name	Index	Modify	Comment	Data type	AT	Display format
MyAlarms	[1].ShortName			STRING[16]		String

	ShortName (STRING[16])	AnalogInput (REAL)	DigitalInput (BOOL)	Disabled (BOOL)	ACK (BOOL)	Reset_Counter (BOOL)	ActiveNoACK (BOOL)	NormalNoACK (BOOL)	ActiveAndACK (BOOL)	Digital_Alarm (BOOL)	HighHigh_Alarm (BOOL)	High_Alarm (BOOL)	Low_Alarm (BOOL)	LowLow_Alarm (BOOL)	HighHigh_Limit (REAL)	High_Limit (REAL)	Low_Limit (REAL)	LowLow_Limit (REAL)	Delay (TIME)	Counter (INT)
[1]		0	False	False	False	False	False	False	False	False	False	False	False	False	0	0	0	0	5s0.000ms	5
[2]		220	False	False	False	False	True	False	False	False	True	True	False	True	200	100	-10	-20	0.000ms	13
[3]		0	False	False	False	False	False	False	False	False	False	False	False	False	0	0	0	0	0.000ms	0
[4]		0	False	False	False	False	False	False	False	False	False	False	False	False	0	0	0	0	0.000ms	0
[5]		0	False	False	False	False	False	False	False	False	False	False	False	False	0	0	0	0	0.000ms	0
[6]		0	False	False	False	False	False	False	False	False	False	False	False	False	0	0	0	0	0.000ms	0
[7]		0	False	False	False	False	False	False	False	False	False	False	False	False	0	0	0	0	0.000ms	0
[8]		0	False	False	False	False	False	False	False	False	False	False	False	False	0	0	0	0	0.000ms	0
[9]		0	False	False	False	False	False	False	False	False	False	False	False	False	0	0	0	0	0.000ms	0
[10]		0	False	False	False	False	False	False	False	False	False	False	False	False	0	0	0	0	0.000ms	0

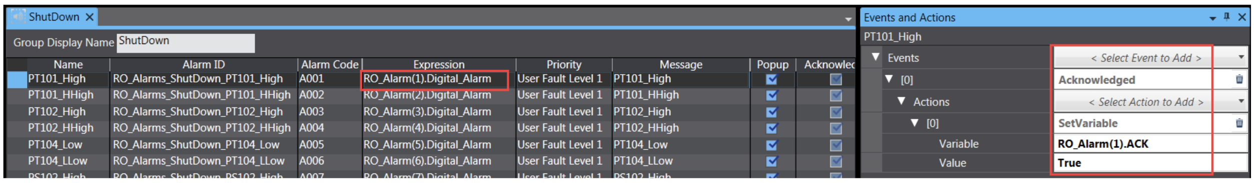


Name	Index	Modify	Comment	Data type	AT	Display format
MyAlarms	[1].HighHigh_Ala	TRUE FALSE		BOOL		Boolean ▼

	HighHigh_Alarm (BOOL)	High_Alarm (BOOL)	Low_Alarm (BOOL)	LowLow_Alarm (BOOL)	HighHigh_Limit (REAL)	High_Limit (REAL)	Low_Limit (REAL)	LowLow_Limit (REAL)	Delay (TIME)	Counter (INT)	ACK_Username (STRING[16])	Alarm_TimeStamp (DATE_AND_TIME)	CriticalAlarm_TimeStamp (DATE_AND_TIME)	ACK_TimeStamp (DATE_AND_TIME)	Inactive_TimeStamp (DATE_AND_TIME)
[1]	False	False	False	False	0	0	0	0	5s0.000ms	5		2016-11-12-23:53:26.49	1970-01-01-00:00:00.00	2016-11-12-23:54:40.10	2016-11-12-23:53:34.41
[2]	True	True	False	True	200	100	-10	-20	0.000ms	13		2016-11-23-14:32:16.20	2016-11-12-23:58:01.92	2016-11-12-23:57:37.38	2016-11-12-23:57:21.94
[3]	False	False	False	False	0	0	0	0	0.000ms	0		1970-01-01-00:00:00.00	1970-01-01-00:00:00.00	1970-01-01-00:00:00.00	1970-01-01-00:00:00.00
[4]	False	False	False	False	0	0	0	0	0.000ms	0		1970-01-01-00:00:00.00	1970-01-01-00:00:00.00	1970-01-01-00:00:00.00	1970-01-01-00:00:00.00
[5]	False	False	False	False	0	0	0	0	0.000ms	0		1970-01-01-00:00:00.00	1970-01-01-00:00:00.00	1970-01-01-00:00:00.00	1970-01-01-00:00:00.00
[6]	False	False	False	False	0	0	0	0	0.000ms	0		1970-01-01-00:00:00.00	1970-01-01-00:00:00.00	1970-01-01-00:00:00.00	1970-01-01-00:00:00.00
[7]	False	False	False	False	0	0	0	0	0.000ms	0		1970-01-01-00:00:00.00	1970-01-01-00:00:00.00	1970-01-01-00:00:00.00	1970-01-01-00:00:00.00
[8]	False	False	False	False	0	0	0	0	0.000ms	0		1970-01-01-00:00:00.00	1970-01-01-00:00:00.00	1970-01-01-00:00:00.00	1970-01-01-00:00:00.00
[9]	False	False	False	False	0	0	0	0	0.000ms	0		1970-01-01-00:00:00.00	1970-01-01-00:00:00.00	1970-01-01-00:00:00.00	1970-01-01-00:00:00.00
[10]	False	False	False	False	0	0	0	0	0.000ms	0		1970-01-01-00:00:00.00	1970-01-01-00:00:00.00	1970-01-01-00:00:00.00	1970-01-01-00:00:00.00

1.10.4. How to use it in the NA

Here is an example of setup in the NA to show alarm texts, and to acknowledge the alarm in the PLC when the alarm is cleared on the screen:



ShutDown

Group Display Name: ShutDown

Name	Alarm ID	Alarm Code	Expression	Priority	Message	Popup	Acknowlec
PT101_High	RO_Alarms_ShutDown_PT101_High	A001	RO_Alarm(1).Digital_Alarm	User Fault Level 1	PT101_High	☑	☑
PT101_HHigh	RO_Alarms_ShutDown_PT101_HHigh	A002	RO_Alarm(2).Digital_Alarm	User Fault Level 1	PT101_HHigh	☑	☑
PT102_High	RO_Alarms_ShutDown_PT102_High	A003	RO_Alarm(3).Digital_Alarm	User Fault Level 1	PT102_High	☑	☑
PT102_HHigh	RO_Alarms_ShutDown_PT102_HHigh	A004	RO_Alarm(4).Digital_Alarm	User Fault Level 1	PT102_HHigh	☑	☑
PT104_Low	RO_Alarms_ShutDown_PT104_Low	A005	RO_Alarm(5).Digital_Alarm	User Fault Level 1	PT104_Low	☑	☑
PT104_LLow	RO_Alarms_ShutDown_PT104_LLow	A006	RO_Alarm(6).Digital_Alarm	User Fault Level 1	PT104_LLow	☑	☑
PS102_High	RO_Alarms_ShutDown_PS102_High	A007	RO_Alarm(7).Digital_Alarm	User Fault Level 1	PS102_High	☑	☑

Events and Actions

PT101_High

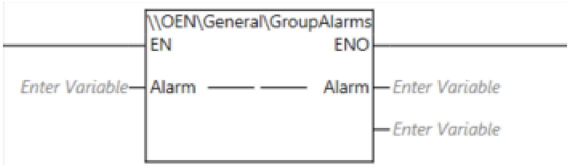
▼ Events	< Select Event to Add >
▼ [0]	Acknowledged
▼ Actions	< Select Action to Add >
▼ [0]	SetVariable
Variable	RO_Alarm(1).ACK
Value	True

The alarm table in this example is RO_Alarm.

2. GroupAlarms

This Function is used in combination with **OEN\General\Alarms**. In the Alarm table, specify the group to which group the alarm belongs to. One can have many alarms in each group. It is recommended to collect single alarms belonging to the same group in number order. The group number refers to Alarm[1] up to Alarm[99]. When using group alarms, Alarm[1-99] is therefore reserved. Alarms going into groups must have numbers from Alarm[101 ->]. Group alarms have their own GroupDigital_Alarm. It is activated by a single alarm in the group but has the property that it goes off for a period defined in Delay on the group alarm. In that way you get a group trigger for each new alarm in the group. See the graphic below in 2.9.

2.1. FN Layout



2.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function

2.3. In-Out Variables

Name	Data type	Valid Range	Description
Alarm	OEN/nGeneral/sAlarms	ARRAY[*] OF	Alarm Table used in FunctionBlock Alarms

2.4. Output Variables

Name	Data Type	Description
ENO	BOOL	TRUE if EN is TRUE

2.5. Table Settings

Settings for each alarm	
Group	Group number (1-99)
Delay	Delay before setting alarm or group alarm

2.6. Table Outputs

Status for each alarm - Number in ( ) is bit position	
GroupDigital_Alarm (12)	DigitalAlarm with OFF_ON Delayed Trigger for each alarm

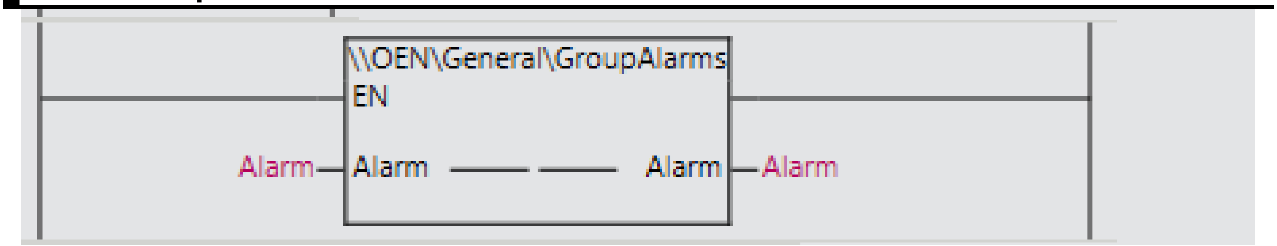
2.7. Revisions

Revision	In Library	Correction
2.1.0	5.00.9	

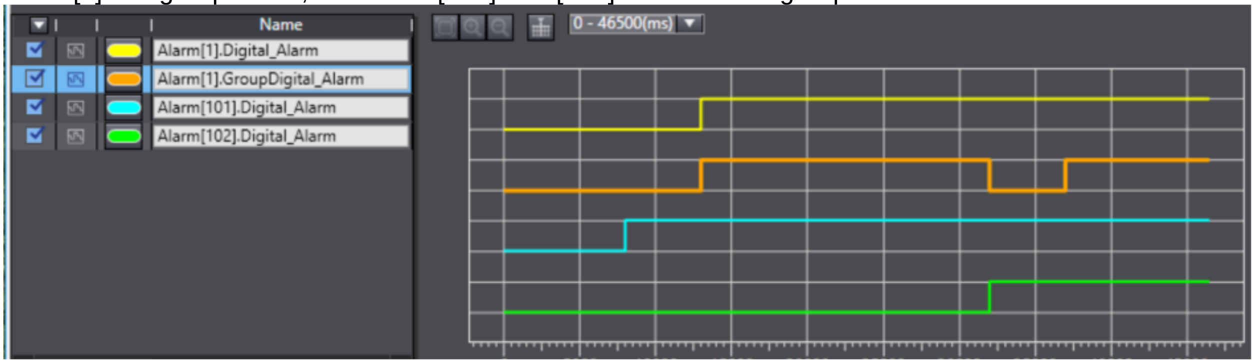
2.8. Credits

	Name
Omron - Norway	Kjell Baardsgaard

2.9. Example



Alarm[1] is a group alarm, and Alarm[101] and [102] are added to group 1.

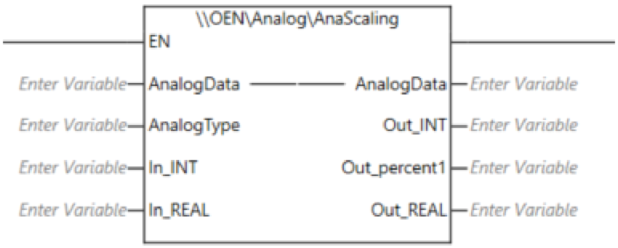


Note the difference between group 1 Digital_Alarm and group 1 GroupDigital_Alarm.

3. *AnaScaling*

A function for scaling analog signals. It can be used for both inputs and outputs. You can choose whether to use AnalogData.In (REAL) or In_INT as well as AnalogData.Out(REAL), Out_INT or Out_REAL.
You can also choose to use AnalogType that automatically detects the correct resolution for NX units, or manually enter the module's resolution with AnalogData.Out.Min/Max or AnalogData.In.Min/Max. With FilterFactor, you can smooth out the signal through an IIR Low Pass Filter. FilterFactor is in ms and relative to scan time.

3.1. FN Layout



3.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
AnalogType	OEN\eNX_Atype	Optional selection of NX-unit
In_INT	INTEGER	Optional input. If omitted, In_Real or Analogdata.In (REAL) has to be used
In_Real	REAL	Optional input. If omitted, In_INT or Analogdata.In (REAL) has to be used

AnalogData structure has more inputs. See 3.6

3.3. In-Out Variables

Name	Data type	Description
AnalogData	OEN\nGeneral\sAnalog	Analog signal data structure

3.4. Output Variables

Name	Data Type	Description
ENO	BOOL	EN is TRUE
Out_INT	INT	Scaled value
Out_percent1	REAL	Scaled value as 0-100%
Out_REAL	REAL	Scaled value

AnalogData structure has more outputs. See 3.7

3.5. Errors

ErrorID	Error Description	Correction

3.6. AnalogData structure inputs

Input	
AnalogData.In	Input (X) if In_INT and In_REAL are omitted
AnalogData.In_Min	Xmin if AnalogType is omitted
AnalogData.In_Max	Xmax if AnalogType is omitted
AnalogData.Out_Min	Ymin – Scaled value
AnalogData.Out_Max	Ymax – Scaled value
AnalogData.FilterConstant	LowPassFilter time (ms) based on CPU scantime
AnalogData.Limit	Variable for AlarmLimits used in other FB/FN
AnalogData.AvgSamples	Variable for MovingAverage Function if used
AnalogData.AvgIndex	Variable for MovingAverage Function if used
AnalogData.Extrapolate	Set TRUE to Extrapolate output. Else Out is set to min/max outside range.

3.7. AnalogData structure outputs

Input	
AnalogData.Out	Scaled value as REAL. Same as Out_REAL
AnalogData.Out_percent1	Scaled value as 0-100%
AnalogData.Out_percent2	Scaled value as -100 - 100%
AnalogData.ERR_BrokenWire	TRUE if Broken Wire output
AnalogData.ERR_InBelowMin	TRUE if Input value below Xmin and NOT Extrapolate
AnalogData.ERR_InAboveMax	TRUE if Input value above Xmax and NOT Extrapolate

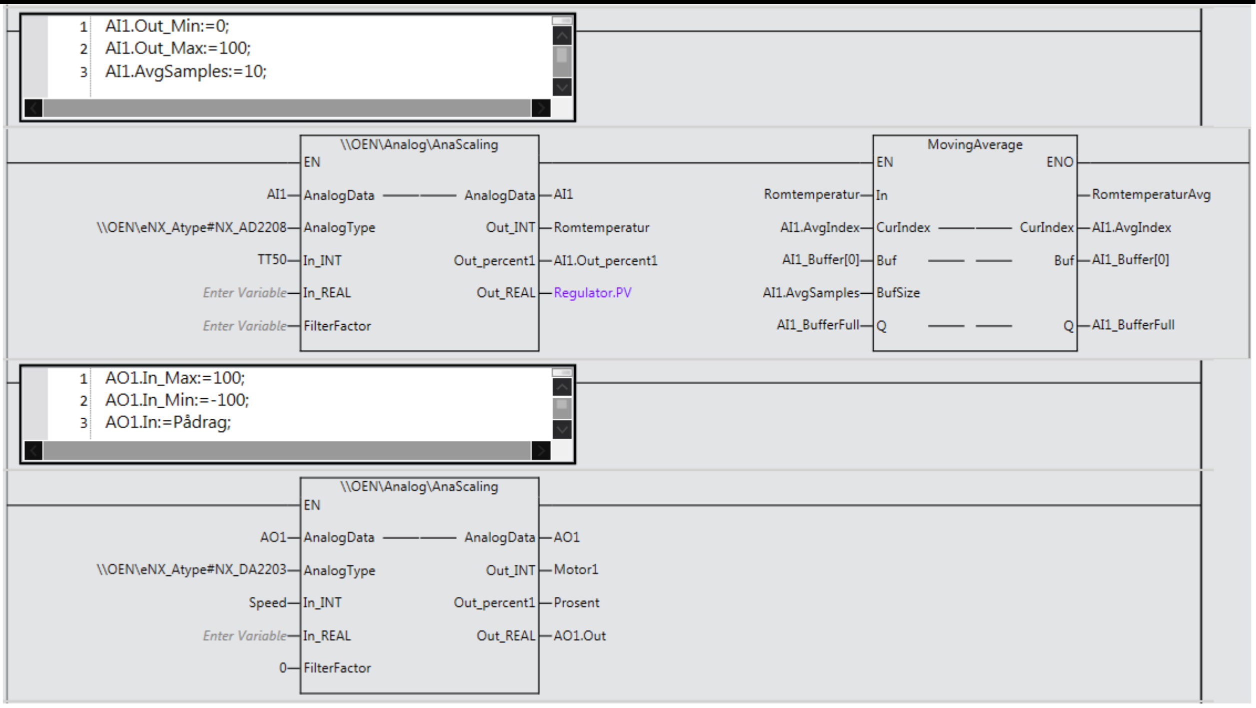
3.8. Revisions

Revision	In Library	Correction
2.8.0	5.00.9	

3.9. Credits

	Name
Omron - Norway	Kjell Baardsgaard

3.10. Example



```
AI1.Out_Min:=0;
AI1.Out_Max:=100;
AI1.AvgSamples:=10;
```

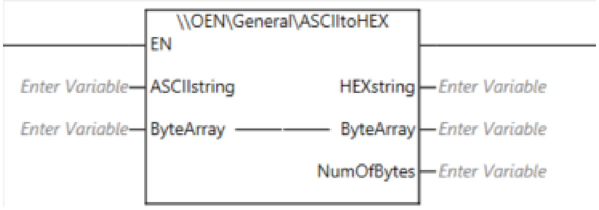
```
AO1.In_Max:=100;
AO1.In_Min:=-100;
AO1.In:=Pådrag;
```

The **MovingAverage** Function can be found in the SysmacStudio Toolbox and can also be used to smooth out the signal. See Help or Manual.

4. *ASCIItoHEX*

A function that converts a text to a HEX string and ByteArray of ASCII codes. At the same time, you also get the number of HEX codes (bytes) in these two. This is not the same as the number of characters in the text, as many special characters provide multiple bytes. (See ASCI Tables on UTF8). Make sure that ByteArray is at least 256 bytes as ASCIIstring is defined to do so.

4.1. FN Layout



4.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
ASCIIString	STRING[256]	String of text to be converted

4.3. In-Out Variables

Name	Data type	Description
ByteArray	ARRAY[*] OF BYTE	Byte array to hold the HEX-codes

4.4. Output Variables

Name	Data Type	Description
ENO	BOOL	EN is TRUE
HEXString	STRING[256]	String of converted text as a HEX-string
NumOfBytes	UINT	Number of bytes in byte array and HEX-string

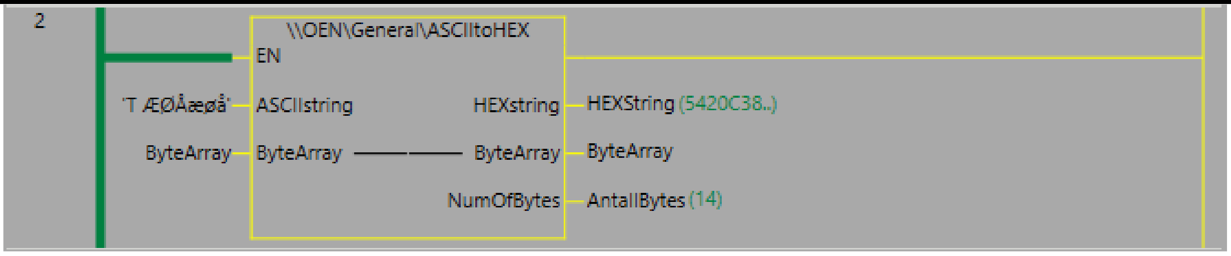
4.5. Revisions

Revision	In Library	Correction
1.0.0	5.00.9	

4.6. Credits

	Name
Omron - Norway	Kjell Baardsgaard

4.7. Example

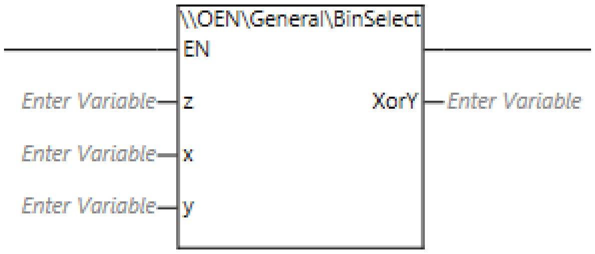


Number of characters are 8, but number of bytes are 14.

5. *BinSelect*

A function that sends the status of one of two inputs out onto the output using a third input. If Z=FALSE, the output is equal to X, if Z=TRUE, the output is equal to Y.

5.1. FN Layout



5.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
Z	BOOL	If FALSE, status of X is output, else status of Y is output
X	BOOL	Signal selected by Z
Y	BOOL	Signal selected by Z

5.3. Output Variables

Name	Data Type	Description
Return	BOOL	
XorY	BOOL	If Z=FALSE, status of X is output, else status of Y is output

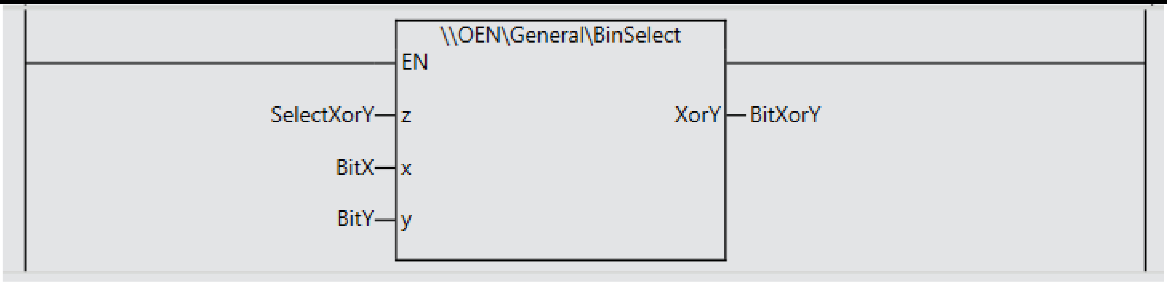
5.4. Revisions

Revision	In Library	Correction
1.1.0	5.00.9	

5.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

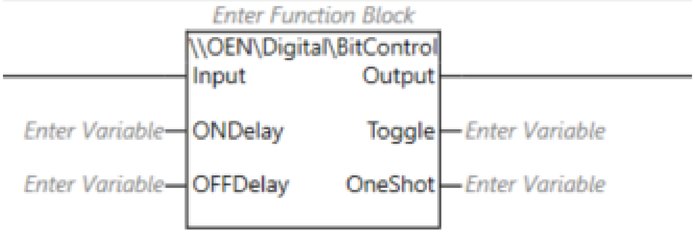
5.6. Example


\\OEN\General\BinSelect
EN
SelectXorY
z
BitX
x
BitY
y
XorY
BitXorY

6. *BitControl*

A multifunction input with 3 outputs with different properties.

6.1. FB Layout



6.2. Input Variables

Name	Data type	Default	Description
Input	BOOL	FALSE	Input signal
ONDelay	TIME	T#0	Time from Input TRUE until Output TRUE
OFFDelay	TIME	T#0	Time from Input FALSE until Output FALSE

6.3. Output Variables

Name	Data Type	Description
Output	BOOL	Output with ONDelay and OFFDelay
Toggle	BOOL	Output that toggles TRUE/FALSE with ONDelay
OneShot	BOOL	Output with ONDelay when Input TRUE. Goes FALSE after OFFDelay independent of Input status.

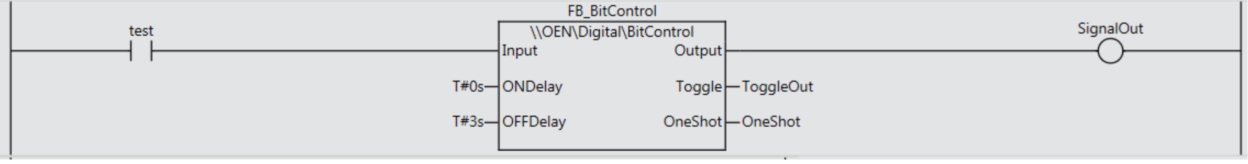
6.4. Revisions

Revision	In Library	Correction
2.0.0	5.00.9	

6.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

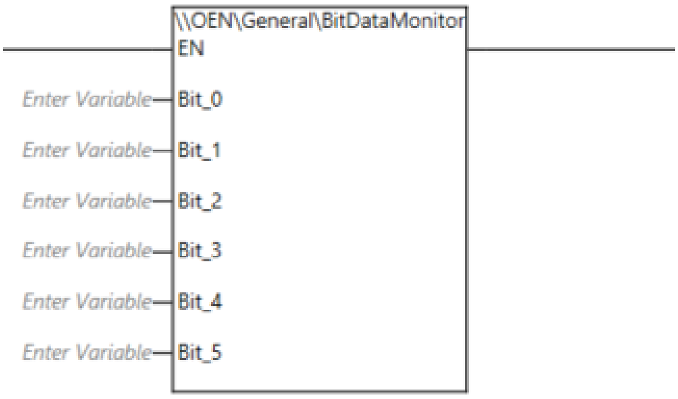
6.6. Example



7. *BitDataMonitor*

A simple feature to collect bit in the program to get better overview.

7.1. FN Layout



\\\\OEN\General\BitDataMonitor
EN
Enter Variable— Bit_0
Enter Variable— Bit_1
Enter Variable— Bit_2
Enter Variable— Bit_3
Enter Variable— Bit_4
Enter Variable— Bit_5

7.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable output
Bit_0	BOOL	Bool variable status monitor
Bit_1	BOOL	Bool variable status monitor
Bit_2	BOOL	Bool variable status monitor
Bit_3	BOOL	Bool variable status monitor
Bit_4	BOOL	Bool variable status monitor
Bit_5	BOOL	Bool variable status monitor

7.3. Output Variables

Name	Data Type	Description
Return	BOOL	EN is TRUE

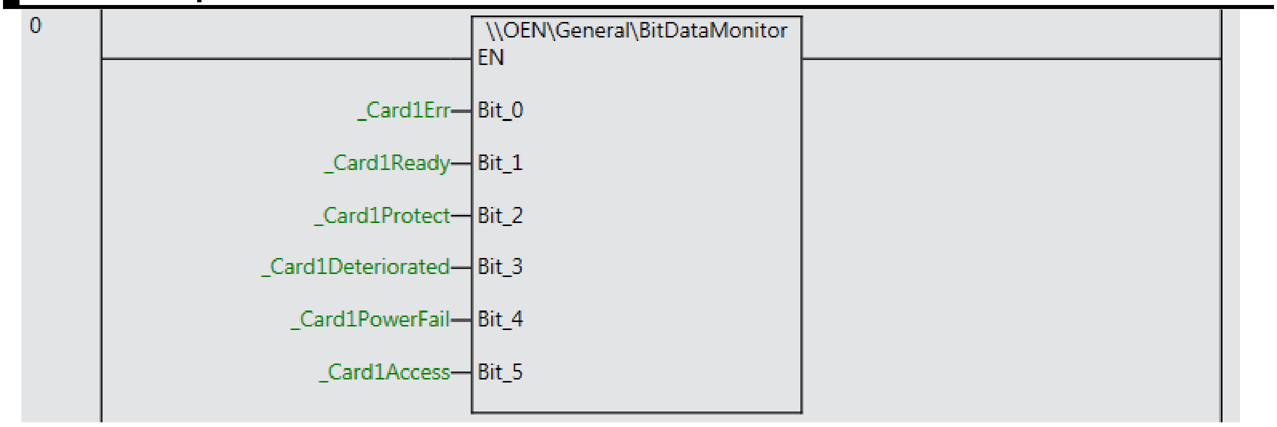
7.4. Revisions

Revision	In Library	Correction
1.0.0	5.00.9	

7.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

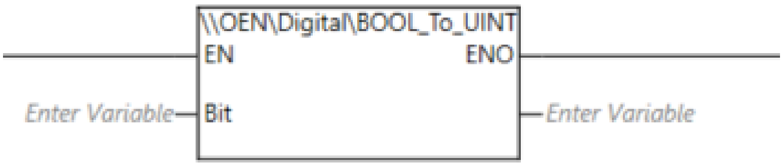
7.6. Example


0
\\OEN\General\BitDataMonitor
EN
_Card1Err
Bit_0
_Card1Ready
Bit_1
_Card1Protect
Bit_2
_Card1Deteriorated
Bit_3
_Card1PowerFail
Bit_4
_Card1Access
Bit_5

8. *BOOL_To_UINT*

Conversion of a bit to integer value 1 or 0.

8.1. FN Layout



8.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
Bit	BOOL	

8.3. Output Variables

Name	Data Type	Description
ENO	BOOL	EN is TRUE
Return	UINT	0 if Bit is FALSE, 1 if Bit is TRUE

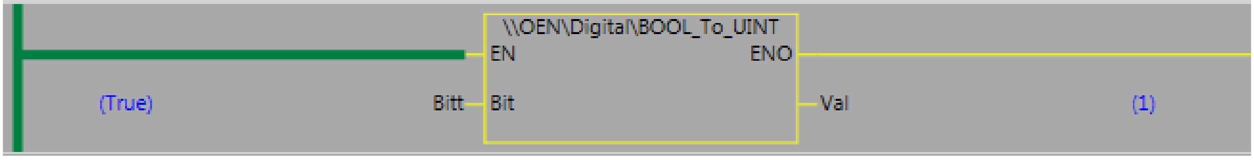
8.4. Revisions

Revision	In Library	Correction
1.0.0	5.00.9	

8.1. Credits

	Name
Omron - Norway	Kjell Baardsgaard

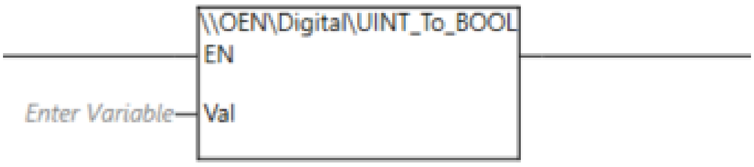
8.2. Example



9. *UINT_To_BOOL*

Conversion of the integer value to a bit.

9.1. FN Layout



9.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
Val	UINT	

9.3. Output Variables

Name	Data Type	Description
Return	BOOL	FALSE if Val is 0, TRUE if not

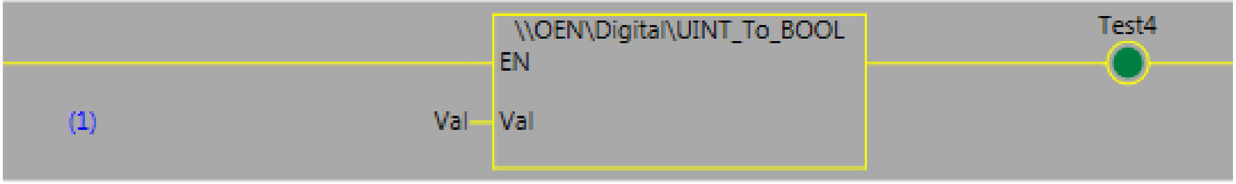
9.4. Revisions

Revision	In Library	Correction
1.0.0	5.00.9	

9.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

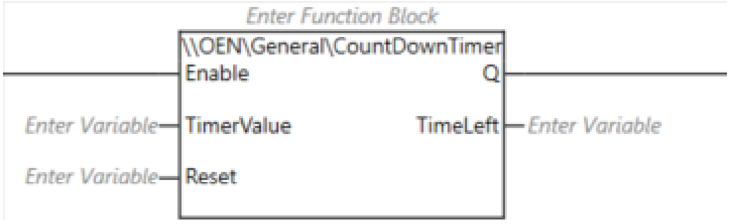
9.6. Example



10. *CountDownTimer*

A timer that counts down from a specific value as long as Enable is active. The value is retained in the event of a power failure.
TimeLeft is automatically reset when Enable goes on and TimeLeft=0.

10.1. FB Layout



10.2. Input Variables

Name	Data type	Valid Range	Default	Description
Enable	BOOL		FALSE	Enable function
TimerValue	TIME			Time to start from
Reset	BOOL			Set TimeLeft to TimerValue

10.3. Output Variables

Name	Data Type	Description
Q	BOOL	TimeLeft is 0
TimeLeft	TIME	Time before Q becomes TRUE

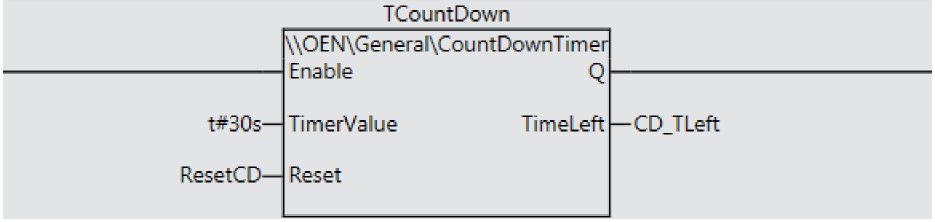
10.4. Revisions

Revision	In Library	Correction
1.0.0	5.00.9	

10.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

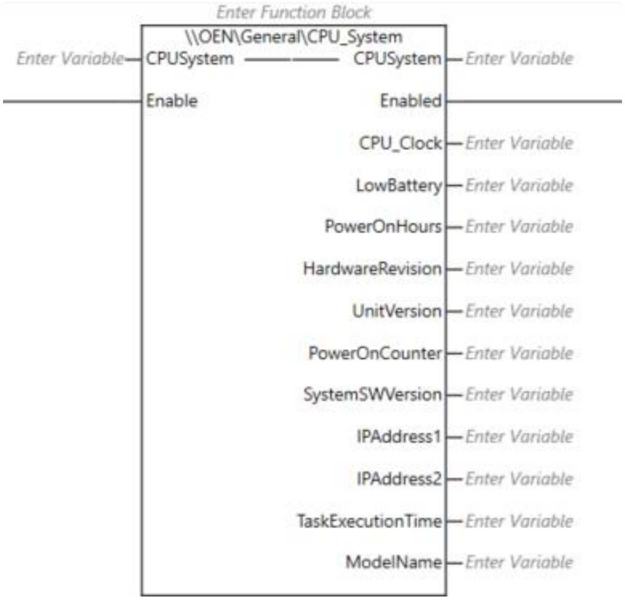
10.6. Example



11. *CPU_System*

A function block that displays system information. CPU_System collects everything in a data structure.
*Applicable CPUs are NJ, NX1P2, NX102.

11.1. FB Layout



11.2. Input Variables

Name	Data type	Description
Enable	BOOL	Enable function

11.3. In-Out Variables

Name	Data type	Description
CPUSystem	OEN\nGeneral\sCPUSystem	Structure containing CPU system data

11.4. Output Variables

Name	Data Type	Description
Enabled	BOOL	Enable is TRUE
CPU_Clock	STRING[30]	CPU Clock
LowBattery	BOOL	Low Battery
PowerOnHours	UDINT	How many CPU run hours ever
HardwareRevision	STRING[2]	
UnitVersion	STRING[7]	
PowerOnCounter	UDINT	How many CPU power ON ever
SystemSWVersion	STRING[10]	Firmware version
IPAddress1	STRING[16]	IP Address of Port#1
IPAddress2	STRING[16]	IP Address of Port#2
TaskExecutionTime	TIME	CPU Cycle time
ModelName	OEN\nGeneral\eCPUModelName	Readable model name as eNum

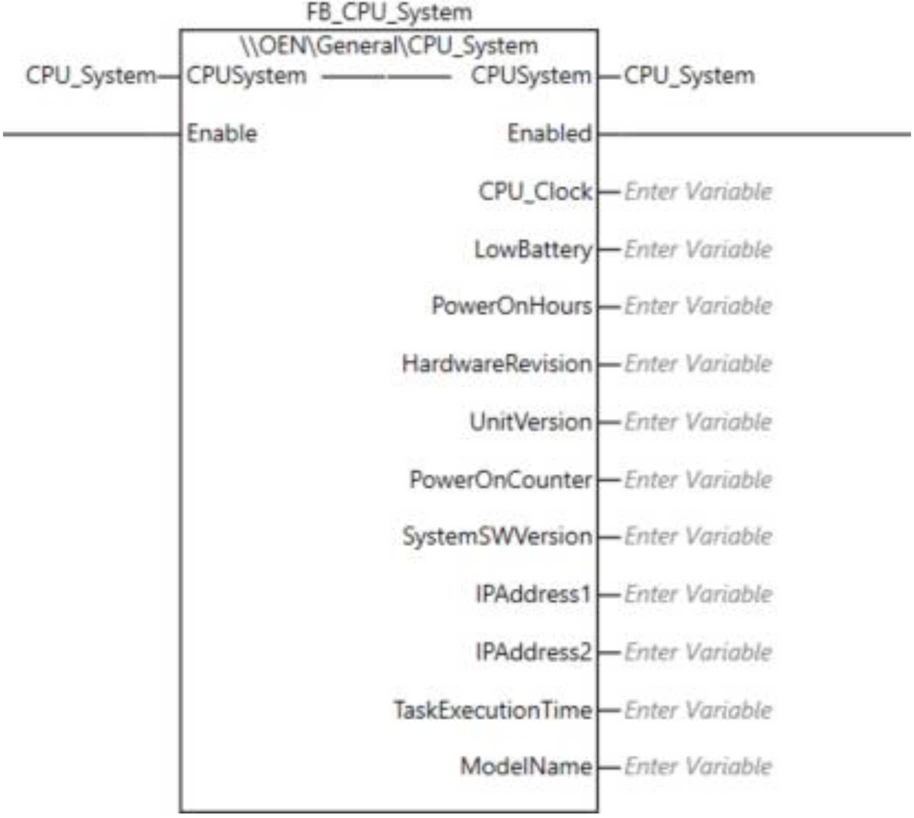
11.5. Revisions

Revision	In Library	Correction
2.0.1	5.00.9	
3.0.0	5.00.11	Added reading IP address of Port#2

11.6. Credits

	Name
Omron - Norway	Kjell Baardsgaard
Omron - France	Arnaud Croizit

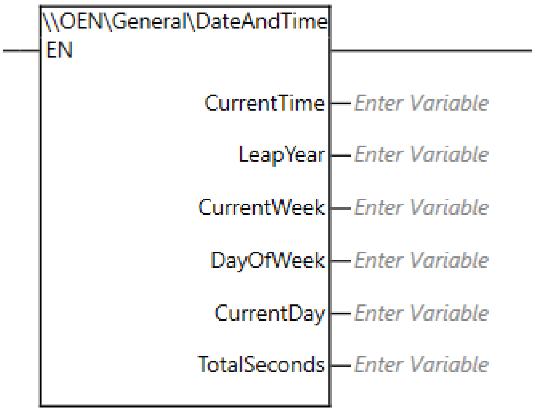
11.7. Example



12. *DateAndTime*

This Function gives you a data structure that contains the current time and date from the System. In addition, both leap year, week number, day of the year and the number of seconds that have passed since January 1st 1970. (Nice to know).

12.1. FN Layout



12.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function

12.3. Output Variables

Name	Data Type	Description
Return	BOOL	EN=TRUE
CurrentTime	_sDT	Current time as a data structure. Updated once per cycle
LeapYear	BOOL	TRUE if Leap year
CurrentWeek	USINT	Current week
DayOfWeek	_eDAYOFWEEK	Current day of week as eNum
CurrentDay	UINT	Current Day
TotalSeconds	LINT	Seconds since 1.1.1970 00:00.00

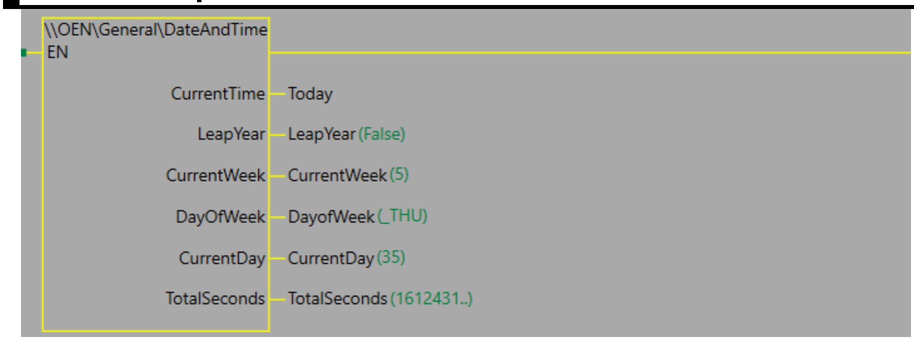
12.4. Revisions

Revision	In Library	Correction
1.1.0	5.00.9	
1.2.0	5.00.10	Fixed Current week according to ISO8601. Added DayOfWeek output.
1.3.0	5.00.10	Change system time reading from _CurrentTime (10Hz) to GetTime(each Cycle)

12.1. Credits

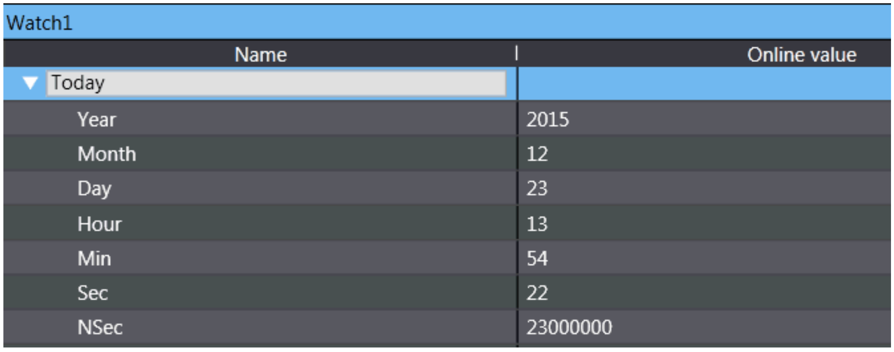
	Name
Omron - Norway	Kjell Baardsgaard

12.2. Example



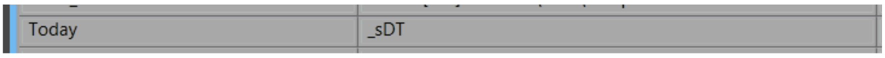
\\OEN\General\DateAndTime
EN

CurrentTime — Today
LeapYear — LeapYear (False)
CurrentWeek — CurrentWeek (5)
DayOfWeek — DayofWeek (_THU)
CurrentDay — CurrentDay (35)
TotalSeconds — TotalSeconds (1612431..)



Watch1

Name	Online value
▼ Today	
Year	2015
Month	12
Day	23
Hour	13
Min	54
Sec	22
NSec	23000000



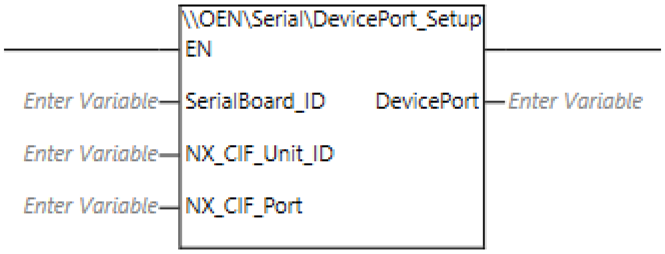
Today	_sDT

Today uses a predefined data structure. We see that because it starts with _s.

13. *DevicePort_Setup*

DevicePort_Setup simplifies the setup of function blocks that use serial ports. Use *DisplayNodeLocationPort* in I/O Map to create a port ID. Remember to select protocol, baud, parity and stopbit for the actual port in the Port Setup screen.

13.1. FN Layout



13.2. Input Variables

Name	Data type	Valid Range	Description
EN	BOOL		Enable function
SerialBoardID	_sOPTBOARD_ID		Serial board ID created in I/O Map or
NX_CIF_Unit_ID	_sNXUNIT_ID		…NX Unit ID created in I/O Map
NX_CIF_Port	USINT	0..2	NX Unit Port Number

13.3. Output Variables

Name	Data Type	Description
Return	BOOL	EN=TRUE
DevicePort	_sDEVICE_PORT	DevicePort Setup Structure

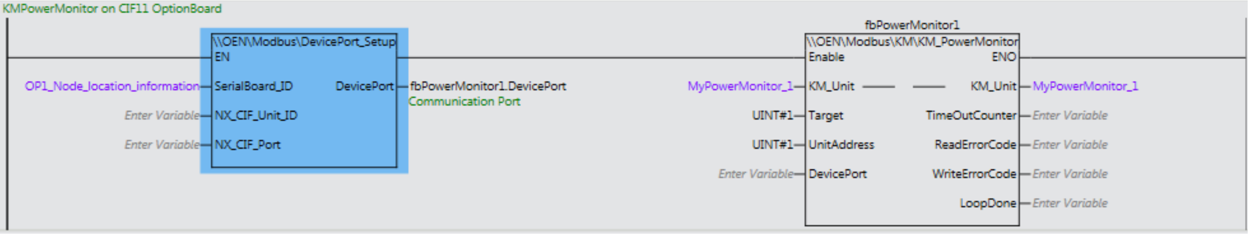
13.4. Revisions

Revision	In Library	Correction
1.2.0	5.00.9	

13.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

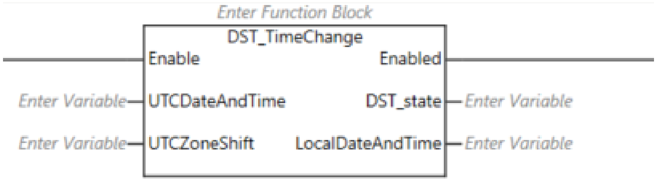
13.6. Example


KMPowerMonitor on CIF11 OptionBoard
\\OEN\Modbus\DevicePort_Setup
EN
OP1_Node_location_information
SerialBoard_ID
DevicePort
fbPowerMonitor1.DevicePort
Communication Port
Enter Variable
NX_CIF_Unit_ID
Enter Variable
NX_CIF_Port
fbPowerMonitor1
\\OEN\Modbus\KM\KM_PowerMonitor
Enable
ENO
MyPowerMonitor_1
KM_Unit
KM_Unit
MyPowerMonitor_1
UINT#1
Target
TimeOutCounter
Enter Variable
UINT#1
UnitAddress
ReadErrorCode
Enter Variable
Enter Variable
DevicePort
WriteErrorCode
Enter Variable
LoopDone
Enter Variable

14. *DST_TimeChange*

A daylight saving time function block. Note that the input should be based on UTC, typically from an NTP Server.

14.1. FB Layout



14.2. Input Variables

Name	Data type	Description
Enable	BOOL	Enable function
UTCDateAndTime	DATE_AND_TIME	UTC Current time and date
UTCZoneShift	TIME	Timezone offset for your location

14.3. Output Variables

Name	Data Type	Description
Enabled	BOOL	Enable=TRUE
DST_State	BOOL	TRUE if Summertime
LocalDateAndTime	DATE_AND_TIME	Local Date and Time

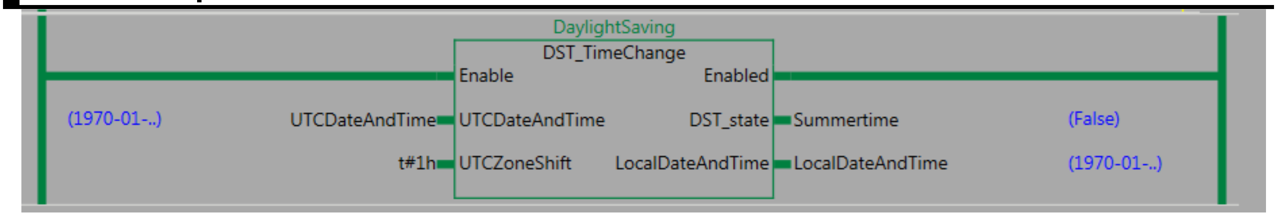
14.4. Revisions

Revision	In Library	Correction
1.0.0	5.00.9	

14.1. Credits

	Name
ATC- Barcelona	
	http://delphiforfun.org/programs/math_topics/dstcalc.htm

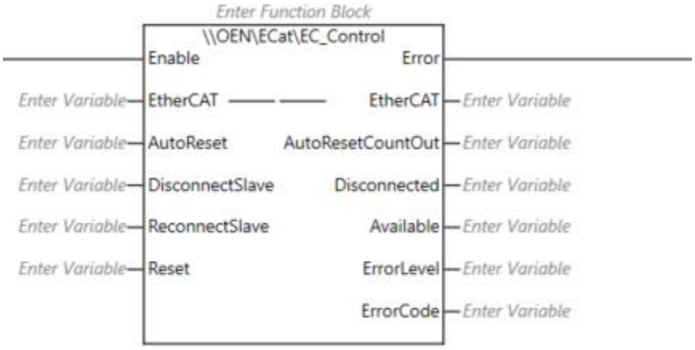
14.2. Example



15. *EC_Control*

With this function block, you can reset the EtherCAT network in case of errors, as well as disconnect nodes out and out of the loop.
The EtherCAT output is a data structure where you can read the status of up to 192 nodes.

15.1. FB Layout



15.2. Input Variables

Name	Data type	Description
Enable	BOOL	Enable function
AutoReset	BOOL	Activate AutoReset if you want to automatically reset errors on the network when slaves fall out and reconnect.
DisconnectSlave	UINT	Set DisconnectSlave to a node number and the node will be disconnected immediately. EtherCAT.CTRL.DisconnectSlave is automatically set to 0 when the slave is offline. Offline slaves can be disconnected from the network without giving an error message.
ReconnectSlave	UINT	Set ReconnectSlave to a node number and the node will be connected immediately. EtherCAT.CTRL.ReconnectSlave is automatically set to 0 when the slave is connected.
Reset	BOOL	With a pulse on Reset, you can manually reset errors if you do not want to use AutoReset.

15.3. In-Out Variables

Name	Data type	Description
EtherCAT	OEN\nECAT\sEC_Network	Structure to hold Network data

15.4. Output Variables

Name	Data Type	Description
Error	BOOL	EtherCAT Error, see ErrorLevel and ErrorCode
AutoresetCountOut	BOOL	AutoResetCountOut indicates that the selected number of resets has been achieved without the error being corrected. The reset interval is 1s.
Disconnected	UINT	Number of Nodes in StandBy.
Available	UINT	Number of Nodes Online+OnlineStandBy
ErrorLevel	UINT	0: No error, 2: Partial Fault, 3: Minor Fault
ErrorCode	DWORD	See GetECError Error Codes in Manual W503

15.5. Optional Settings

Name	Data type	Default	Description
<EtherCAT>.CTRL.AutoresetCount	UINT	#3	May be set to another value if one does not want to use 3.
<EtherCAT>.CTRL.AutoresetDelay		1s	May be set to another value if one does not want to use 1s.

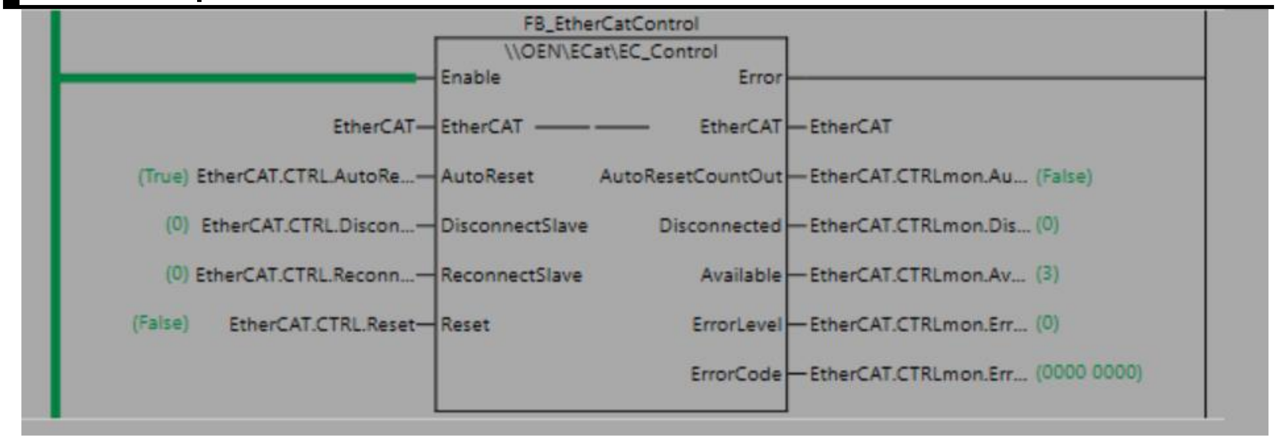
15.6. Revisions

Revision	In Library	Correction
3.3.0	5.00.9	

15.7. Credits

	Name
Omron - Norway	Kjell Baardsgaard

15.8. Example

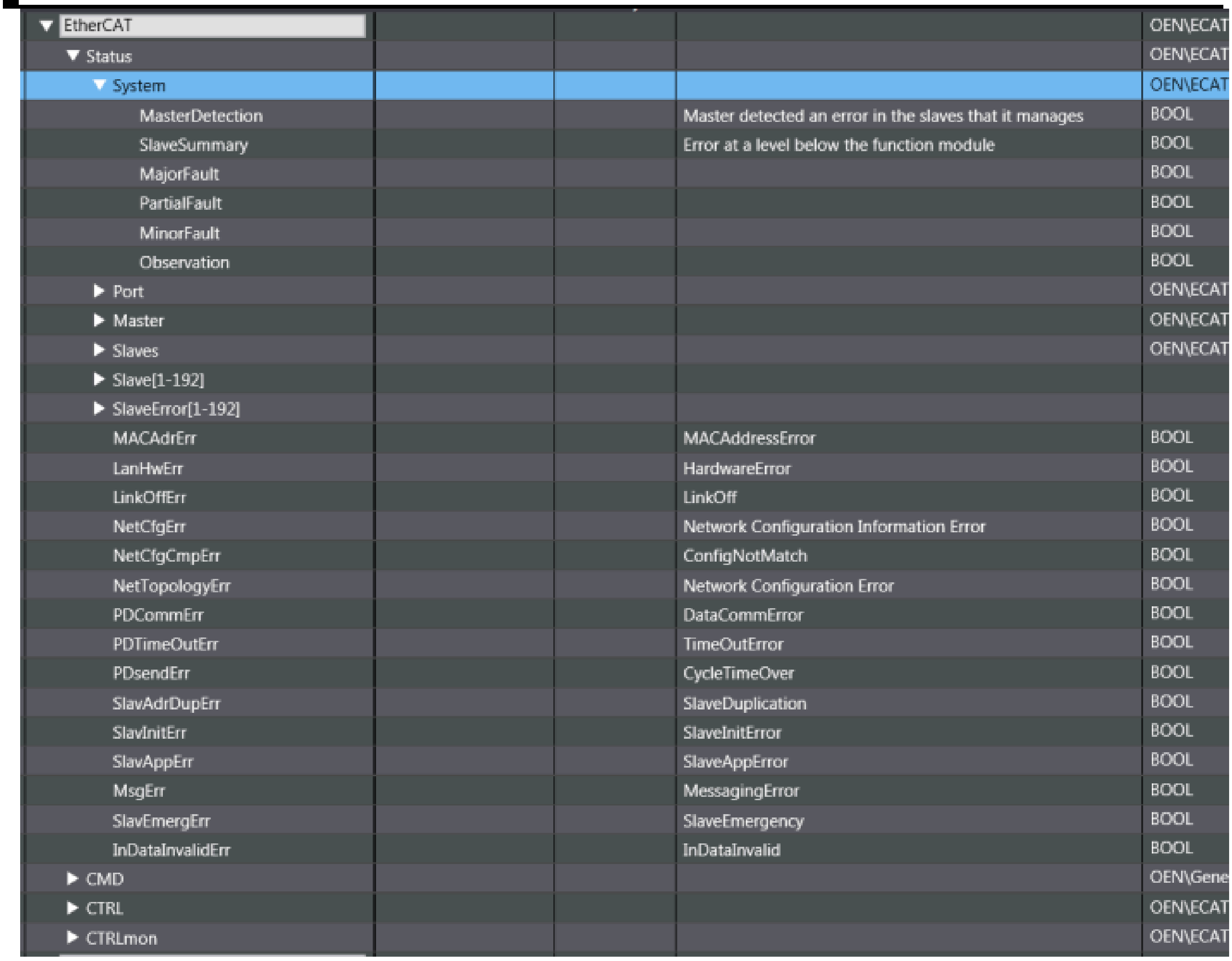


FB_EtherCatControl
\\OEN\ECat\EC_Control

Enable
EtherCAT — EtherCAT
(True) EtherCAT.CTRL.AutoRe... — AutoReset
(0) EtherCAT.CTRL.Discon... — DisconnectSlave
(0) EtherCAT.CTRL.Reconn... — ReconnectSlave
(False) EtherCAT.CTRL.Reset — Reset

Error
EtherCAT — EtherCAT
AutoResetCountOut — EtherCAT.CTRLmon.Au... (False)
Disconnected — EtherCAT.CTRLmon.Dis... (0)
Available — EtherCAT.CTRLmon.Av... (3)
ErrorLevel — EtherCAT.CTRLmon.Err... (0)
ErrorCode — EtherCAT.CTRLmon.Err... (0000 0000)

15.9. Structure

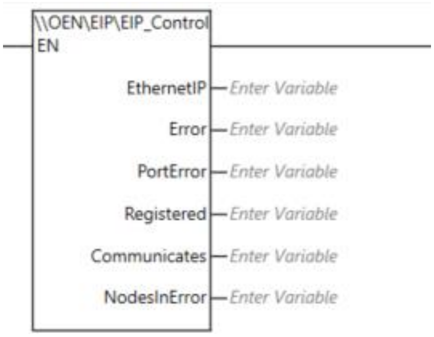


▼ EtherCAT				OEN\ECAT
▼ Status				OEN\ECAT
▼ System				OEN\ECAT
MasterDetection			Master detected an error in the slaves that it manages	BOOL
SlaveSummary			Error at a level below the function module	BOOL
MajorFault				BOOL
PartialFault				BOOL
MinorFault				BOOL
Observation				BOOL
▶ Port				OEN\ECAT
▶ Master				OEN\ECAT
▶ Slaves				OEN\ECAT
▶ Slave[1-192]				
▶ SlaveError[1-192]				
MACAdrErr			MACAddressError	BOOL
LanHwErr			HardwareError	BOOL
LinkOffErr			LinkOff	BOOL
NetCfgErr			Network Configuration Information Error	BOOL
NetCfgCmpErr			ConfigNotMatch	BOOL
NetTopologyErr			Network Configuration Error	BOOL
PDCommErr			DataCommError	BOOL
PDTimeOutErr			TimeOutError	BOOL
PDsendErr			CycleTimeOver	BOOL
SlavAdrDupErr			SlaveDuplication	BOOL
SlavInitErr			SlaveInitError	BOOL
SlavAppErr			SlaveAppError	BOOL
MsgErr			MessagingError	BOOL
SlavEmergErr			SlaveEmergency	BOOL
InDataInvalidErr			InDataInvalid	BOOL
▶ CMD				OEN\Gene
▶ CTRL				OEN\ECAT
▶ CTRLmon				OEN\ECAT

16. *EIP_Control*

This Function shows the status of the EIP communication in a data structure. Registered, Communicates, and NodesInError display the number of nodes defined, that communicate and that have error status. Applicable for NJ and NX1P2.

16.1. FN Layout



16.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function

16.3. Output Variables

Name	Data Type	Description
Return	BOOL	EN=TRUE
EthernetIP	OEN\nEIP\sEIP_Network	Structure containing status for all EIP Nodes
Error	WORD	Status of System variable _EIPErrSta
PortError	WORD	Status of System variable _EIP_PortErr. See Manual.
Registered	USINT	Number of Nodes registered
Communicates	USINT	Number of Nodes Communicating
NodesInError	USINT	Number of Nodes in Error

16.1. Optional Settings

Name	Data type	Default	Description
<EthernetIP>.Cmd.Reset	BOOL		Not in use

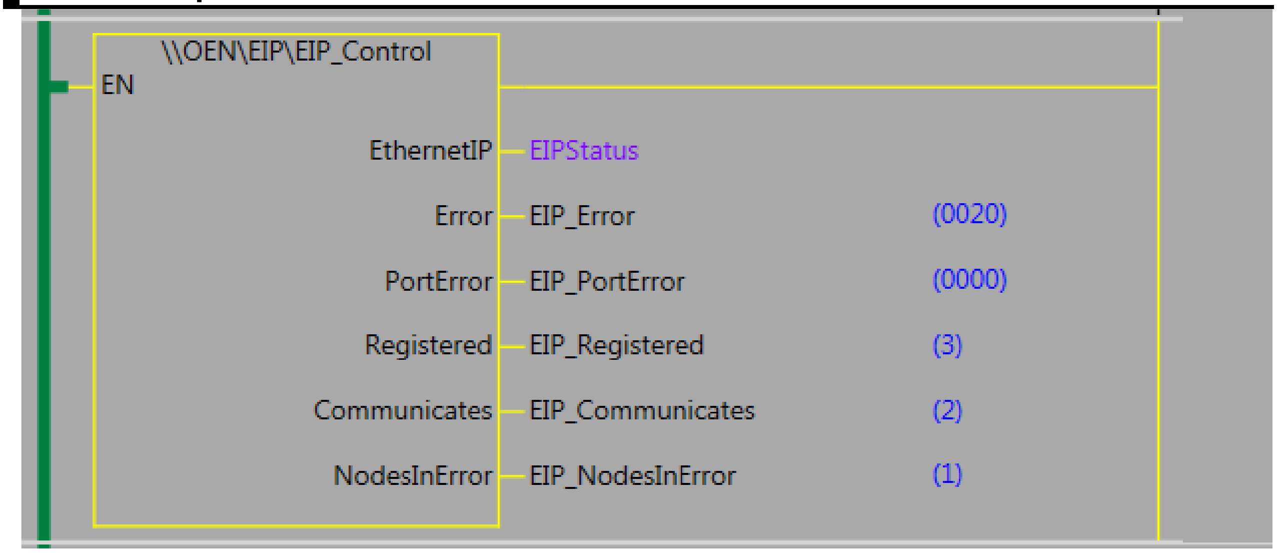
16.2. Revisions

Revision	In Library	Correction
2.0.0	5.00.9	

16.3. Credits

	Name
Omron - Norway	Kjell Baardsgaard

16.4. Example

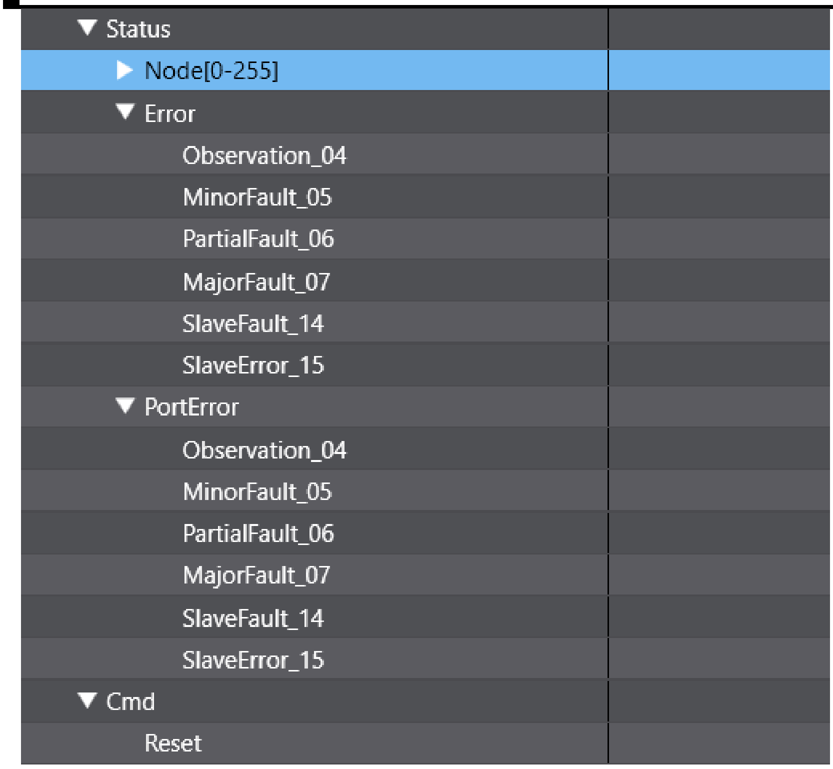


\\OEN\EIP\EIP_Control

EN

Pin	Variable	Value
EthernetIP	EIPStatus	
Error	EIP_Error	(0020)
PortError	EIP_PortError	(0000)
Registered	EIP_Registered	(3)
Communicates	EIP_Communicates	(2)
NodesInError	EIP_NodesInError	(1)

16.5. Structure

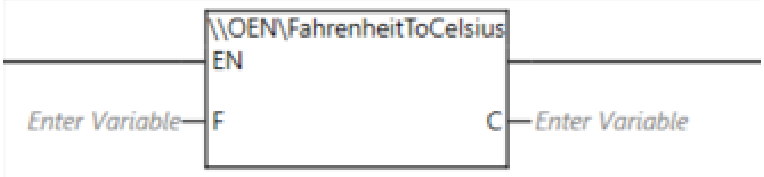


- ▼ Status
  - ▶ Node[0-255]
  - ▼ Error
    - Observation_04
    - MinorFault_05
    - PartialFault_06
    - MajorFault_07
    - SlaveFault_14
    - SlaveError_15
  - ▼ PortError
    - Observation_04
    - MinorFault_05
    - PartialFault_06
    - MajorFault_07
    - SlaveFault_14
    - SlaveError_15
- ▼ Cmd
  - Reset

17. *FahrenheitToCelsius*

Conversion of degrees Fahrenheit to Celsius.

17.1. FN Layout



17.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
F	REAL	Fahrenheit

17.3. Output Variables

Name	Data Type	Description
Return	BOOL	EN is TRUE
C	REAL	Celsius

17.4. Revisions

Revision	In Library	Correction
1.1.0	5.00.9	

17.5. Credits

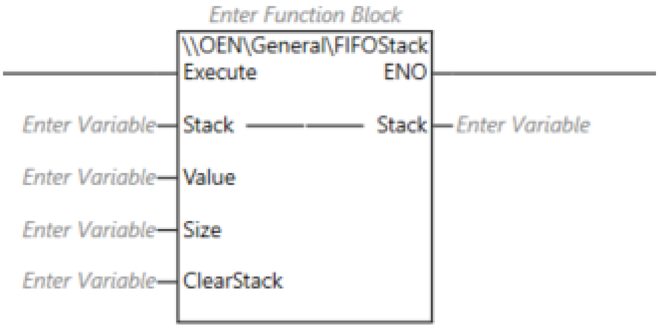
	Name
Omron - Norway	Kjell Baardsgaard

17.6. Example

18. *FIFOStack*

Simple data logging where the table "Stack" (Array[0..99] of REAL) stores the "Size" last "Value". This can be displayed as a floating curve in BrokenLineGraph in NA.

18.1. FB Layout



18.2. Input Variables

Name	Data type	Valid Range	Description
Execute	BOOL		Execute function block
Value	REAL		Value to add to Stack
Size	UINT	1..99	Stack Size
ClearStack	BOOL		Set all values in Stack to 0

18.3. In-Out Variables

Name	Data type	Description
Stack	ARRAY[0..99] OF REAL	Stack to hold logged data

18.4. Output Variables

Name	Data Type	Description
ENO	BOOL	

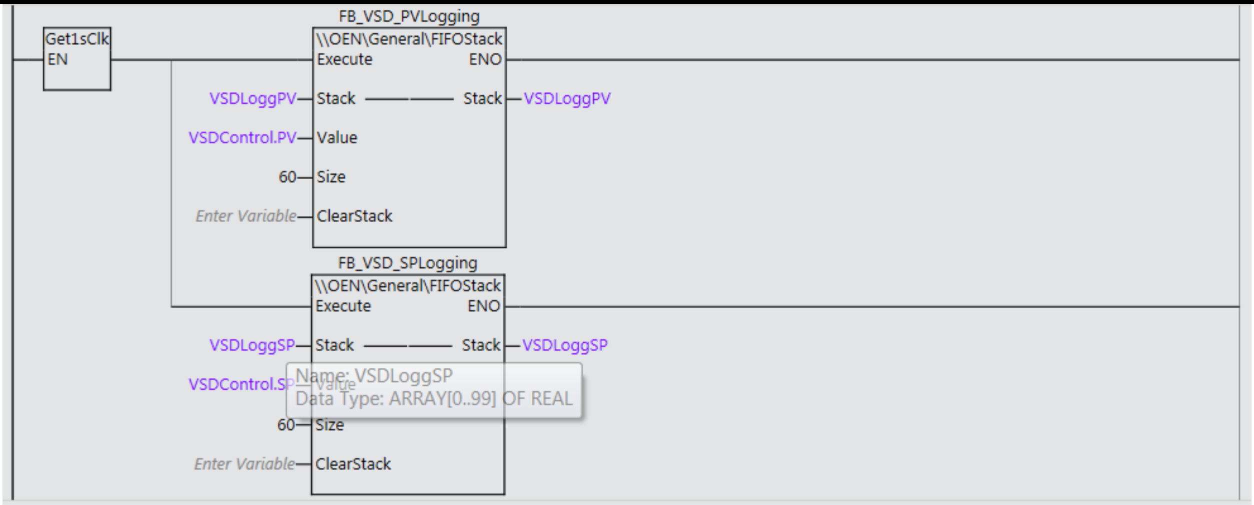
18.5. Revisions

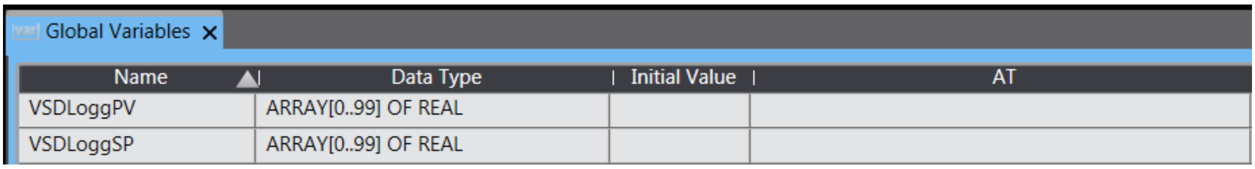
Revision	In Library	Correction
1.0.0	5.00.9	

18.6. Credits

	Name
Omron - Norway	Kjell Baardsgaard

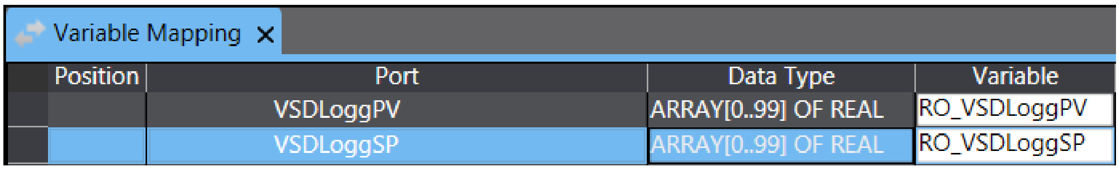
18.7. Example



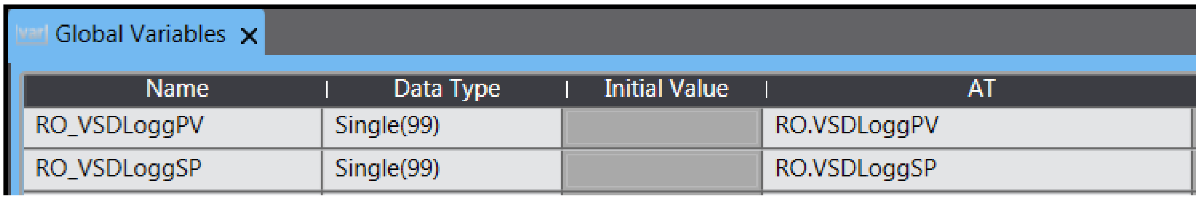


Name	Data Type	Initial Value	AT
VSDLoggPV	ARRAY[0..99] OF REAL		
VSDLoggSP	ARRAY[0..99] OF REAL		

Configure the Graph and create the variables in NA:

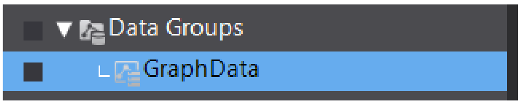


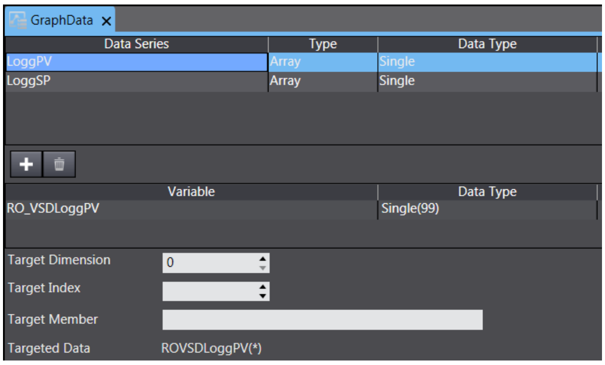
Position	Port	Data Type	Variable
	VSDLoggPV	ARRAY[0..99] OF REAL	RO_VSDLoggPV
	VSDLoggSP	ARRAY[0..99] OF REAL	RO_VSDLoggSP



Name	Data Type	Initial Value	AT
RO_VSDLoggPV	Single(99)		RO.VSDLoggPV
RO_VSDLoggSP	Single(99)		RO.VSDLoggSP

Then we create a Data Group with two variables connected to these:





Data Series	Type	Data Type
LoggPV	Array	Single
LoggSP	Array	Single

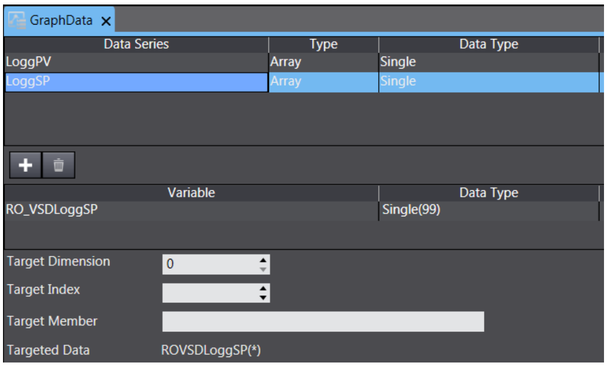
Variable	Data Type
RO_VSDLoggPV	Single(99)

Target Dimension 0

Target Index

Target Member

Targeted Data ROVSDLoggPV(*)



Data Series	Type	Data Type
LoggPV	Array	Single
LoggSP	Array	Single

Variable	Data Type
RO_VSDLoggSP	Single(99)

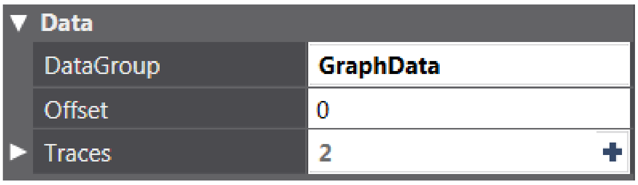
Target Dimension 0

Target Index

Target Member

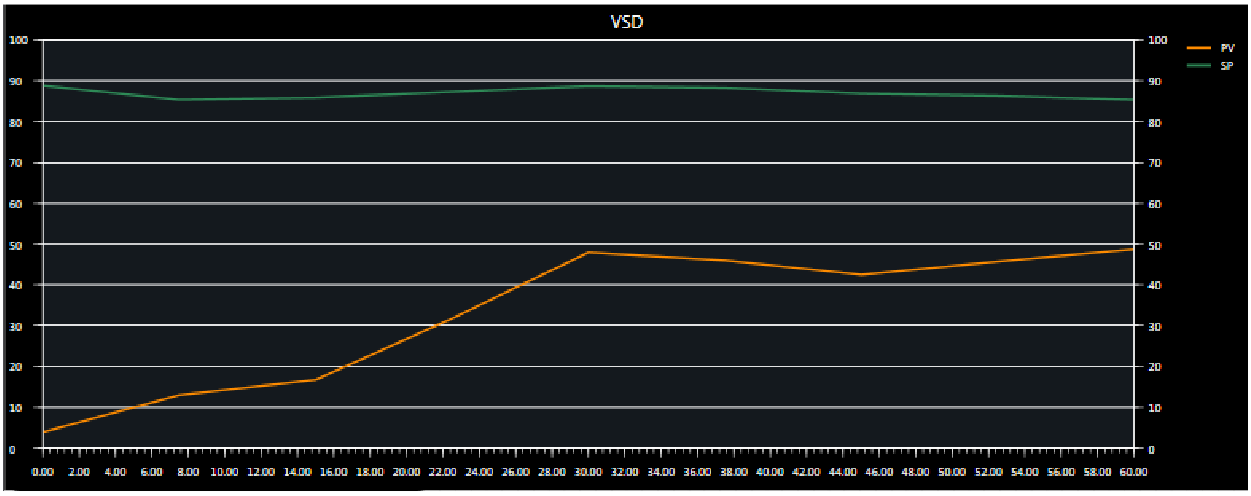
Targeted Data ROVSDLoggSP(*)

Finally, we create BrokenLineGraph by DataGroup:



▼ Data

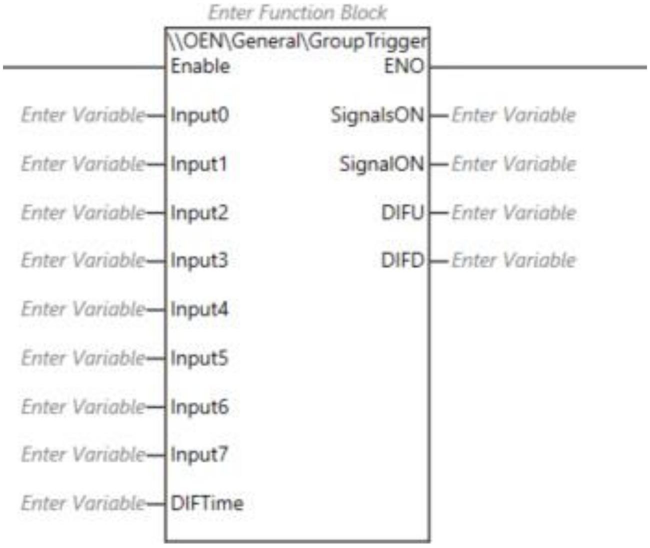
DataGroup	GraphData
Offset	0
▶ Traces	2 ✚



19. *GroupTrigger*

A function block that looks at a change on 8 inputs.

19.1. FB Layout



19.2. Input Variables

Name	Data type	Description
Enable	BOOL	Enable function
Input0..Input7	BOOL	8 inputs to detect signal change
DIFTime	TIME	Pulse length for DIFU and DIFD

19.3. Output Variables

Name	Data Type	Description
ENO	BOOL	Enable is TRUE
SignalsON	UINT	Number of inputs ON
SignalON	BOOL	TRUE if SignalsON>0
DIFU	BOOL	Pulse when an input goes ON
DIFD	BOOL	Pulse when an input goes OFF

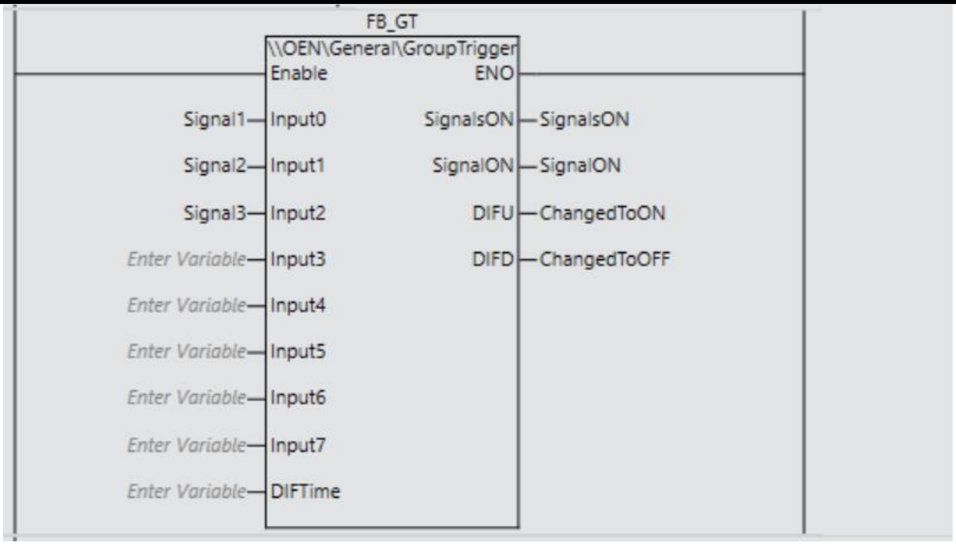
19.4. Revisions

Revision	In Library	Correction
1.0.0	5.00.9	

19.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

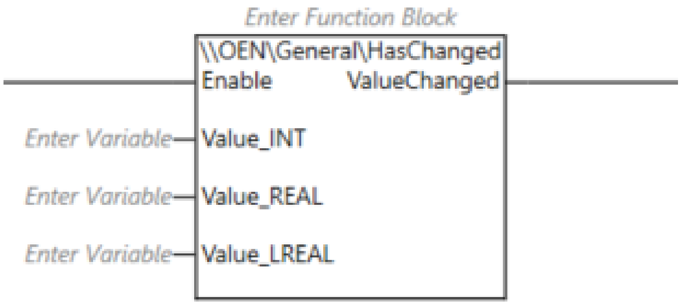
19.6. Example


FB_GT
\\OEN\General\GroupTrigger
Enable
ENO
Signal1
Input0
Signal2
Input1
Signal3
Input2
Enter Variable
Input3
Enter Variable
Input4
Enter Variable
Input5
Enter Variable
Input6
Enter Variable
Input7
Enter Variable
DIFTime
SignalsON
SignalsON
SignalON
SignalON
DIFU
ChangedToON
DIFD
ChangedToOFF

20. *HasChanged*

A function block that gives a pulse if the value of the input changes. One can choose to use one of the three inputs or all.

20.1. FB Layout



20.2. Input Variables

Name	Data type	Description
Enable	BOOL	Enable function
Value_INT	INT	
Value_REAL	REAL	
Value_LREAL	LREAL	

20.3. Output Variables

Name	Data Type	Description
ValueChanged	BOOL	Pulse output if any input value changes

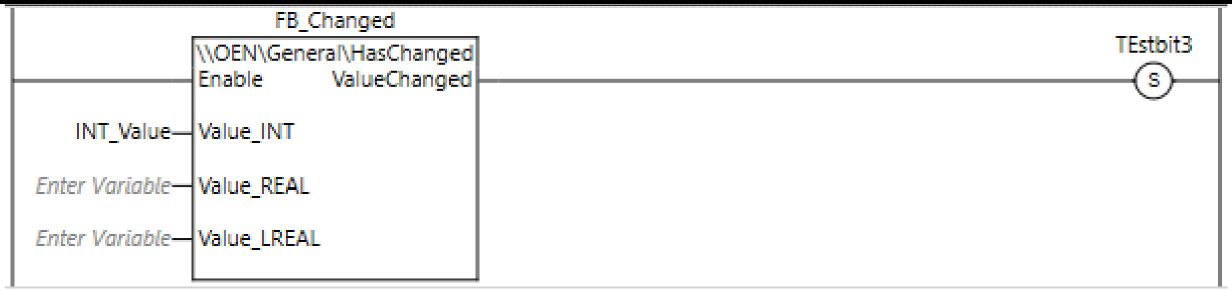
20.4. Revisions

Revision	In Library	Correction
1.0.0	5.00.9	

20.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

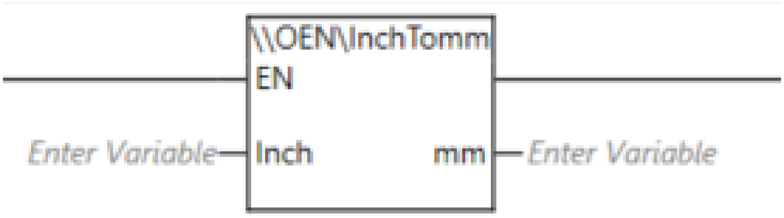
20.6. Example


FB_Changed
\\OEN\General\HasChanged
Enable
ValueChanged
INT_Value
Value_INT
Enter Variable
Value_REAL
Enter Variable
Value_LREAL
TEstbit3
S

21. *InchTomm*

Conversion from Inch to mm

21.1. FN Layout



21.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
Inch	REAL	Inches

21.3. Output Variables

Name	Data Type	Description
Return	BOOL	EN is TRUE
mm	REAL	Millimeters

21.4. Revisions

Revision	In Library	Correction
1.0.0	5.00.9	

21.5. Credits

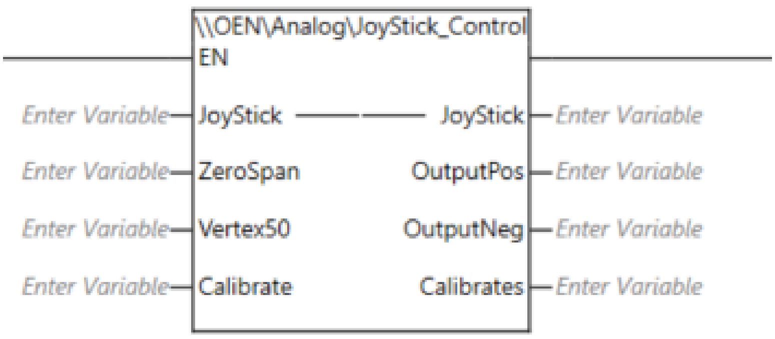
	Name
Omron - Norway	Kjell Baardsgaard

21.6. Example

22. *JoyStick_Control*

This function allows you to split an analog signal into two equal parts so you can drive up/down or back/forth.

22.1. FN Layout



22.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
ZeroSpan	REAL	With ZeroSpan, add a percentage that corresponds to movement around the center of joystick that should output 0.
Vertex50	REAL	With Vertex50, you can decide how much the output should give in % when Joystick has 50% power.
Calibrate	BOOL	With Calibrate, you can learn joystick's zeropoint, maxpoint and minpoint.

22.3. In-Out Variables

Name	Data type	Description
JoyStick	OEN\nGeneral\sJoystick	Structure that holds the data.

22.4. Output Variables

Name	Data Type	Description
ENO	BOOL	EN is TRUE
OutputPos	REAL	Analog positive output (forward)
OutputNeg	REAL	Analog negative output (reverse)
Calibrates	BOOL	Calibrate is active

22.5. Revisions

Revision	In Library	Correction
1.1.0	5.00.9	

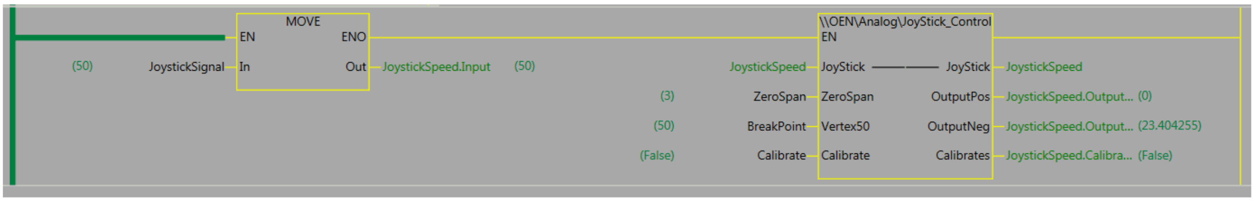
22.6. Credits

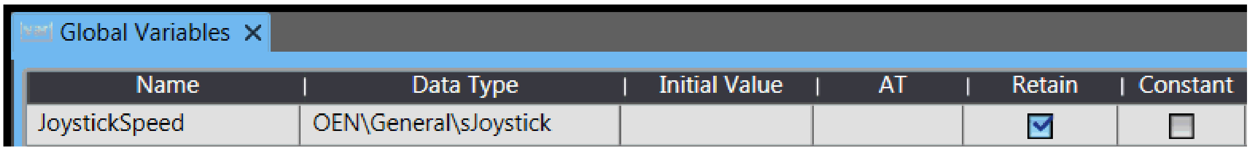
	Name
Omron - Norway	Kjell Baardsgaard

22.7. Example

HOW TO CALIBRATE

1) Put the Joystick in the center.
2) Activate Calibrate
3) Move joystick to Max
4) Move joystick to Min
5) Release Joystick
6) Turn off Calibrate



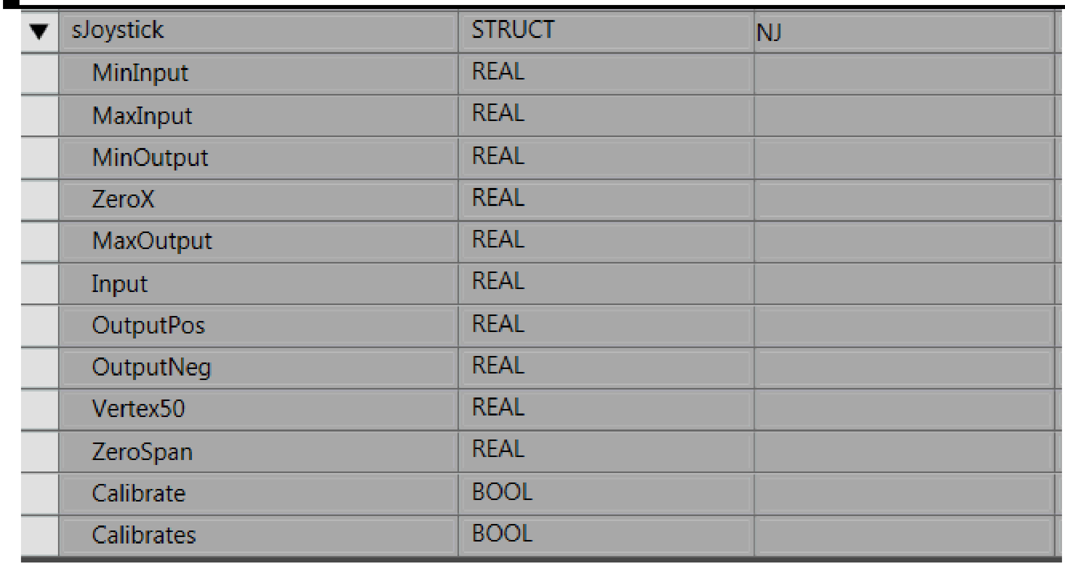


Global Variables

Name	Data Type	Initial Value	AT	Retain	Constant
JoystickSpeed	OEN\General\sJoystick			☑	☐

Don’t forget to Retain it.

22.8. Structure

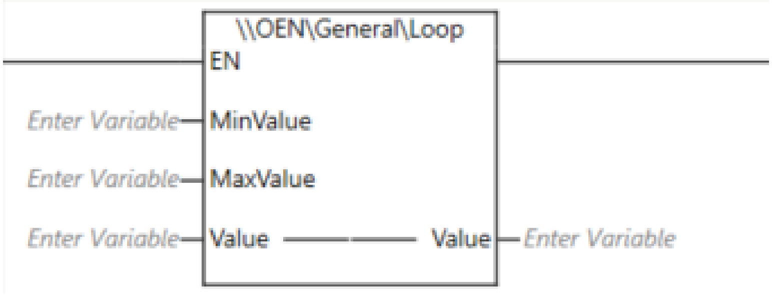


▼ sJoystick	STRUCT	NJ
MinInput	REAL	
MaxInput	REAL	
MinOutput	REAL	
ZeroX	REAL	
MaxOutput	REAL	
Input	REAL	
OutputPos	REAL	
OutputNeg	REAL	
Vertex50	REAL	
ZeroSpan	REAL	
Calibrate	BOOL	
Calibrates	BOOL	

23. *Loop*

A function that runs a value between MinValue and MaxValue for each scan EN is active.

23.1. FN Layout



23.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
MinValue	INT	Minimum loop value
MaxValue	INT	Maximum loop value

23.3. In-Out Variables

Name	Data type	Description
Value	INT	Variable that holds the loop value

23.4. Output Variables

Name	Data Type	Description
Return	BOOL	

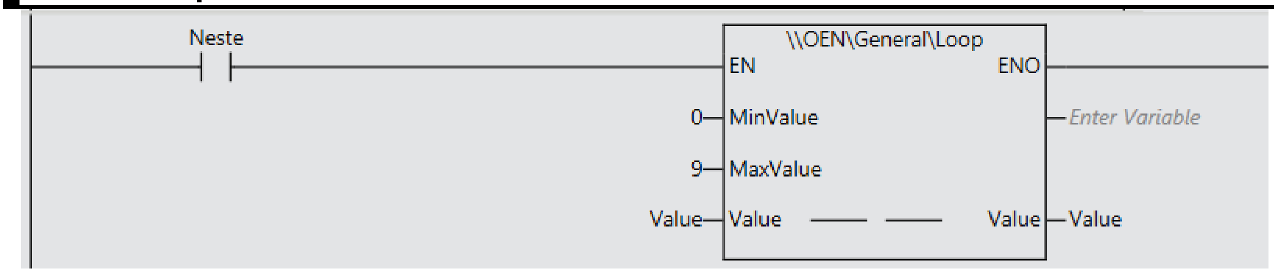
23.5. Revisions

Revision	In Library	Correction
1.2.0	5.00.9	

23.6. Credits

	Name
Omron - Norway	Kjell Baardsgaard

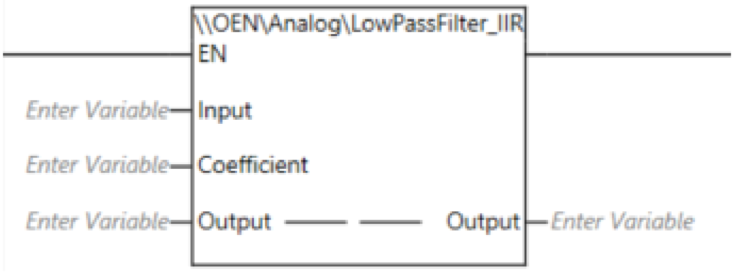
23.7. Example


Neste
\\OEN\General\Loop
EN
ENO
0
MinValue
Enter Variable
9
MaxValue
Value
Value
Value
Value

24. *LowPassFilter_IIR*

Low pass filter to smooth analog signals.

24.1. FN Layout



24.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
Input	REAL	Input
Coefficient	REAL	Factor divided by Scantime. 0.0=Disable filter

24.3. In-Out Variables

Name	Data type	Description
Output	REAL	Filtered output

24.4. Output Variables

Name	Data Type	Description
Return	BOOL	EN is TRUE

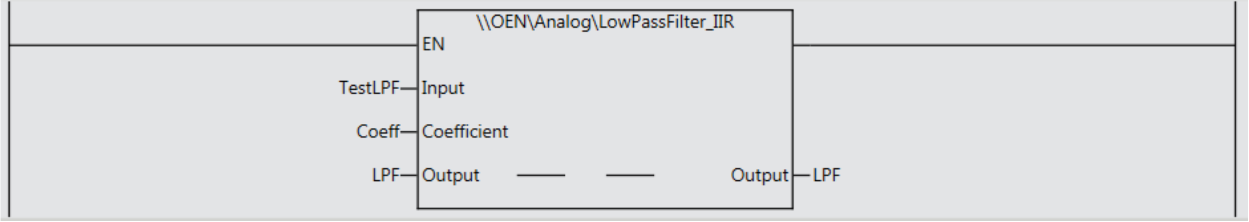
24.5. Revisions

Revision	In Library	Correction
1.1.0	5.00.9	

24.6. Credits

	Name
Omron - Norway	Kjell Baardsgaard
	Roger Eriksen

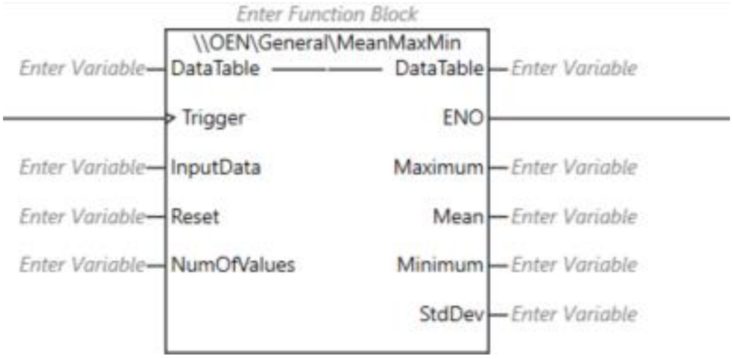
24.7. Example


\\OEN\Analog\LowPassFilter_IIR
EN
TestLPF
Input
Coeff
Coefficient
LPF
Output
Output
LPF

25. *MeanMaxMin*

A function block to find average, max value, min value, and standard deviation of the numbers in a FIFO table.

25.1. FB Layout



25.2. Input Variables

Name	Data type	Description
Trigger	BOOL	Add InputData into the DataTable
InputData	REAL	Value to be added to the table
Reset	BOOL	Clear the table
NumOfValues	UINT	Number of values to be a part of the calculation

25.3. In-Out Variables

Name	Data type	Description
DataTable	ARRAY[*] OF REAL	FIFO Table to hold the data

25.4. Output Variables

Name	Data Type	Description
ENO	BOOL	
Maximum	REAL	Largest value of the selected number of values
Mean	REAL	Average value of the selected number of values
Minimum	REAL	Lowest value of the selected number of values
StdDev	REAL	Standard deviation of the selected number of values.

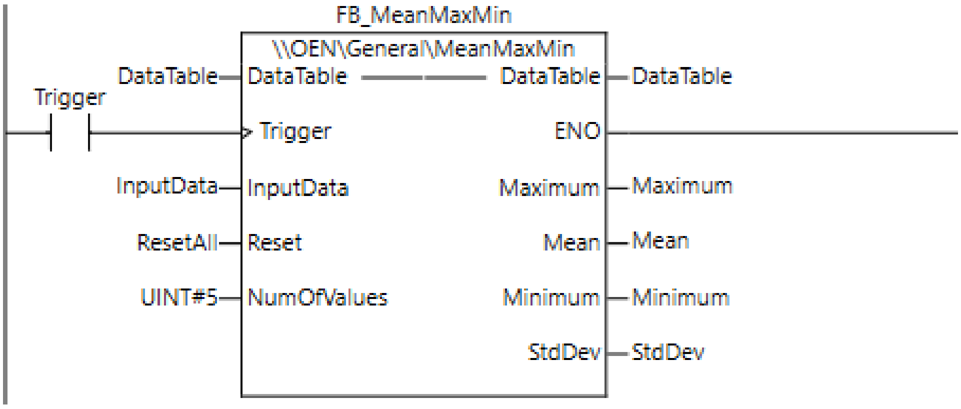
25.5. Revisions

Revision	In Library	Correction
1.0.0	5.00.9	

25.6. Credits

	Name
Omron - Norway	Kjell Baardsgaard

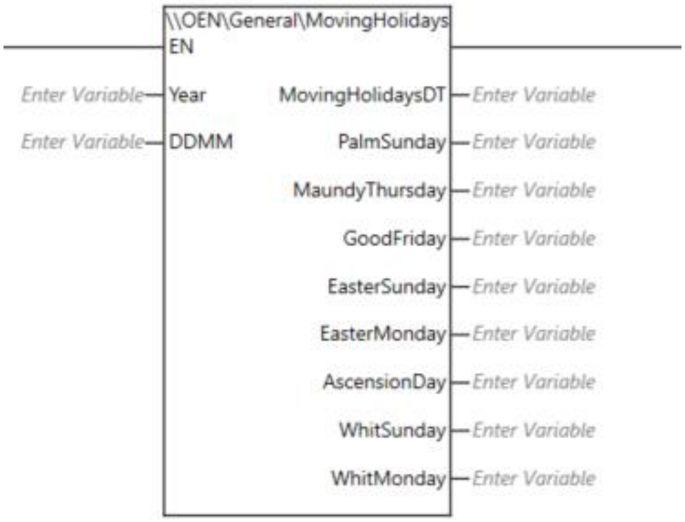
25.7. Example


FB_MeanMaxMin
\\OEN\General\MeanMaxMin
Trigger
DataTable
DataTable
DataTable
DataTable
Trigger
ENO
InputData
InputData
Maximum
Maximum
ResetAll
Reset
Mean
Mean
UINT#5
NumOfValues
Minimum
Minimum
StdDev
StdDev

26. *MovingHolidays*

With this feature, you can get the date of moving holidays like WORD or _sDT data type. Select the current year on Year. You can choose MMDD or DDMM format with the input DDMM. The MovingHolidaysDT variable contains all these days as a _sDT structure. It can be useful along with other function blocks.

26.1. FN Layout



26.2. Input Variables

Name	Data type	Valid Range	Default	Description
EN	BOOL		FALSE	Enable function
Year	UINT	1970-		Year to find moving holidays
DDMM	BOOL			FALSE=MMDD, TRUE=DDMM

26.3. Output Variables

Name	Data Type	Description
Return	BOOL	
MovingHolidaysDT	OEN\nGeneral\sMovingHolidaysDT	Moving holidays structure with dates
PalmSunday	WORD	Ex. April 10 as 0410 or 1004.
MaundyThursday	WORD	
GoodFriday	WORD	
EasterSunday	WORD	
EasterMonday	WORD	
AscensionDay	WORD	
WhitSunday	WORD	
WhitMonday	WORD	

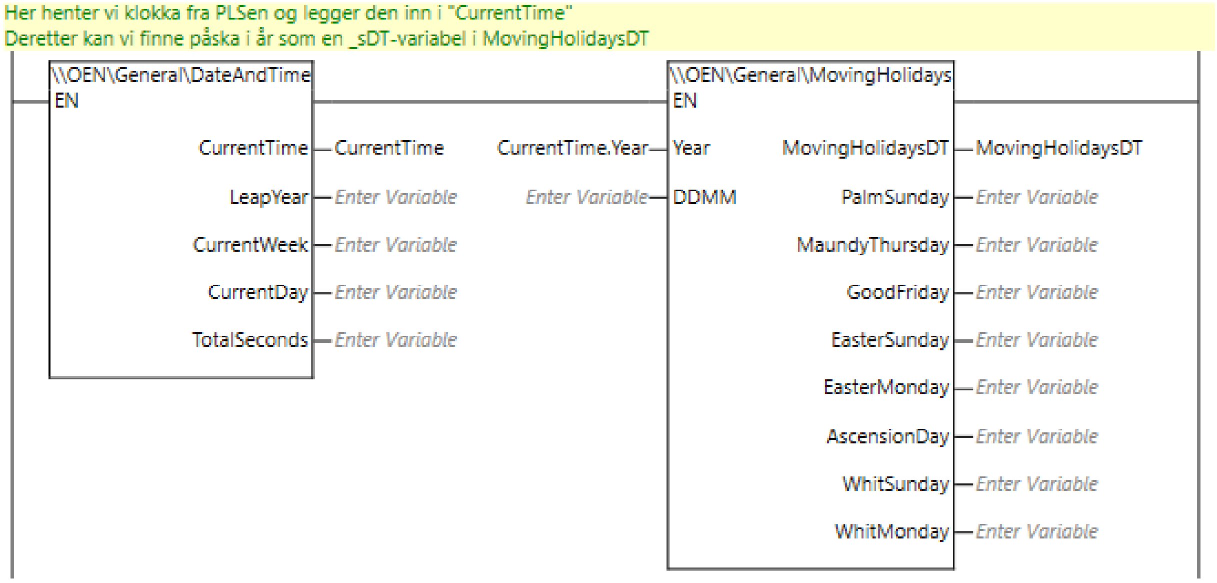
26.4. Revisions

Revision	In Library	Correction
2.0.0	5.00.9	

26.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

26.6. Example

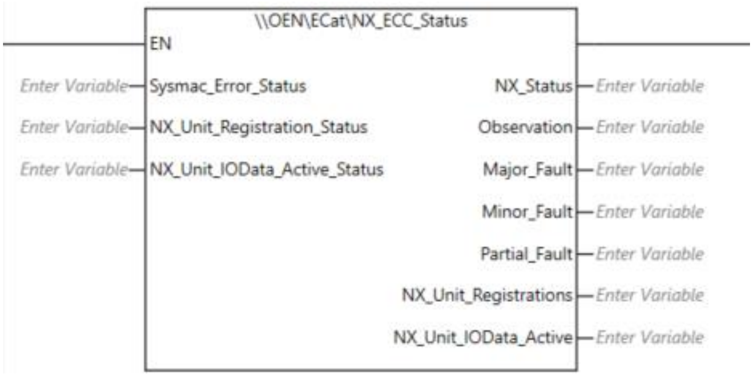


Her henter vi klokka fra PLSen og legger den inn i "CurrentTime"
Deretter kan vi finne påska i år som en _sDT-variabel i MovingHolidaysDT

27. *NX_ECC_Status*

A function block to get a better overview of status bit on an NX-ECC node. Everything is collected in the structure NX_Status for easier handling further.

27.1. FN Layout



27.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
Sysmac_Error_Status	BYTE	I/O Map Variable
NX_Unit_Registration_Status	ARRAY[0..63] OF BOOL	I/O Map variable. Registration Status for Units on NX-ECC
NX_Unit_IOData_Active_Status	ARRAY[0..63] OF BOOL	I/O Map variable Active Status for Units on NX-ECC

27.3. Output Variables

Name	Data Type	Description
Return	BOOL	
NX_Status	OEN\nGeneral\sNX_Status	Structure containing Units Status
Observation	BOOL	Observation signal
Major_Fault	BOOL	ECC-Coupler Major Fault
Minor_Fault	BOOL	ECC-Coupler Minor Fault
Partial_Fault	BOOL	ECC-Coupler Partial Fault
NX_Unit_Registrations	USINT	Registered slaves
NX_Unit_IOData_Active	USINT	Active slaves

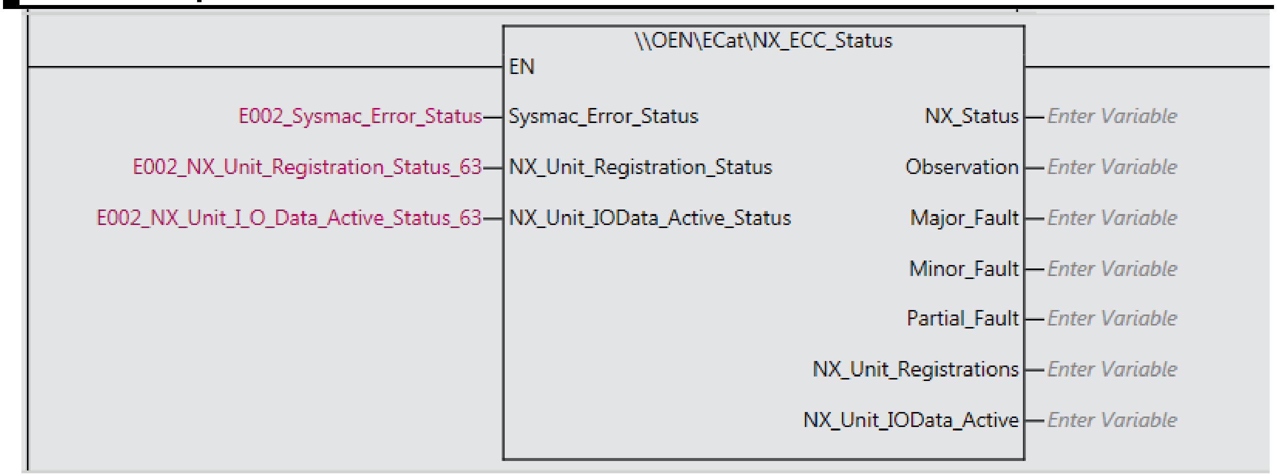
27.4. Revisions

Revision	In Library	Correction
1.1.0	5.00.9	

27.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

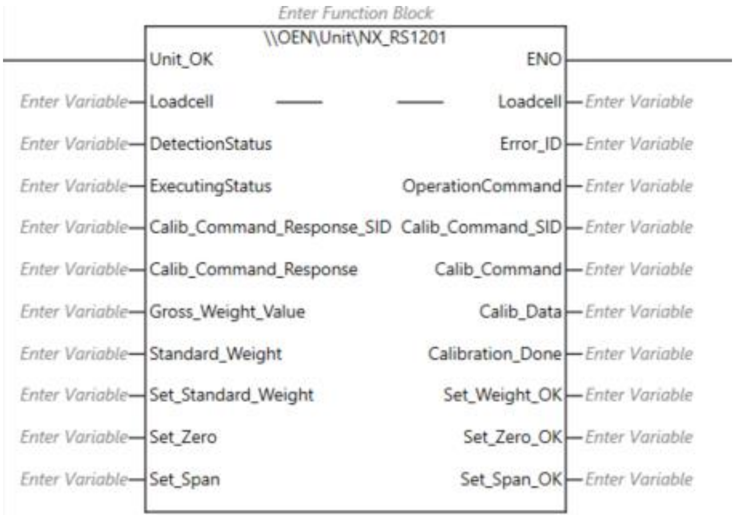
27.6. Example


\\OEN\ECat\NX_ECC_Status
EN
E002_Sysmac_Error_Status
Sysmac_Error_Status
E002_NX_Unit_Registration_Status_63
NX_Unit_Registration_Status
E002_NX_Unit_I_O_Data_Active_Status_63
NX_Unit_IOData_Active_Status
NX_Status
Observation
Major_Fault
Minor_Fault
Partial_Fault
NX_Unit_Registrations
NX_Unit_IOData_Active
Enter Variable
Enter Variable
Enter Variable
Enter Variable
Enter Variable
Enter Variable
Enter Variable

28. *NX_RS1201*

Simplifies the use of the RS1201 NX module for weighing cells. Use the Loadcell structure for configuration and control.

28.1. FB Layout



28.2. Input Variables

Name	Data type	Description
Unit_OK	BOOL	Activate input when NX-RS1201 is Ok
DetectionStatus	WORD	Variable from I/O Map
ExecutingStatus	WORD	Variable from I/O Map
Calib_Command_Response_SID	UINT	Variable from I/O Map
Calib_Command_Response	WORD	Variable from I/O Map
Gross_Weight_Value	REAL	Variable from I/O Map
Standard_Weight	REAL	A known load for calibration
Set_Standard_Weight	BOOL	Calibrate Step#1 Save the value for known load in Step#3
Set_Zero	BOOL	Calibrate Step#2 Set zeropoint for loadcell without any load
Set_Span	BOOL	Calibrate Step#3 Set span after putting the known load on the loadcell

28.3. In-Out Variables

Name	Data type	Description
Loadcell	OEN\nUnit\sRS1201	Structure to hold the data

28.4. Output Variables

Name	Data Type	Description
ENO	BOOL	Unit OK
Error_ID	WORD	Calibration Error
OperationCommand	WORD	Variable from I/O Map
Calib_Command_SID	UINT	Variable from I/O Map
Calib_Command	WORD	Variable from I/O Map
Calib_Data	REAL	Variable from I/O Map

Calibration_Done	BOOL	Calibration Done
Set_Weight_OK	BOOL	Step#1 Set Weight Done
Set_Zero_OK	BOOL	Step#2 Set Zero Done
Set_Span_OK	BOOL	Set#3 Set Span Done

28.5. Errors

ErrorID	Error Description	Correction

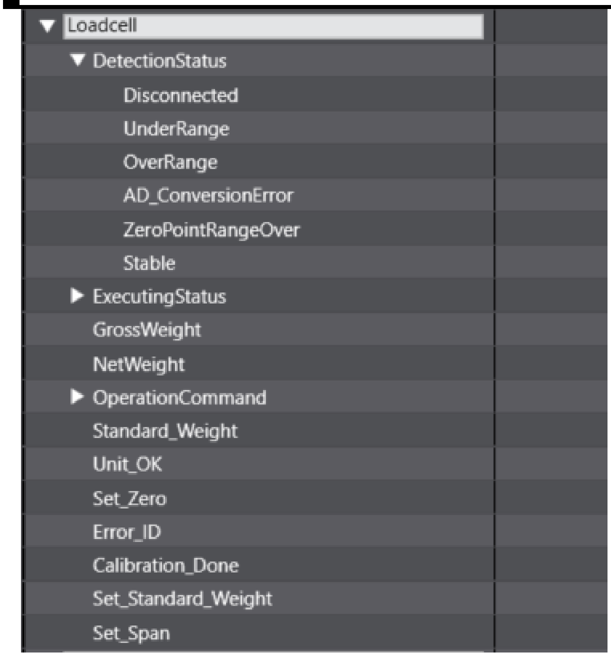
28.6. Revisions

Revision	In Library	Correction
2.2.0	5.00.9	

28.7. Credits

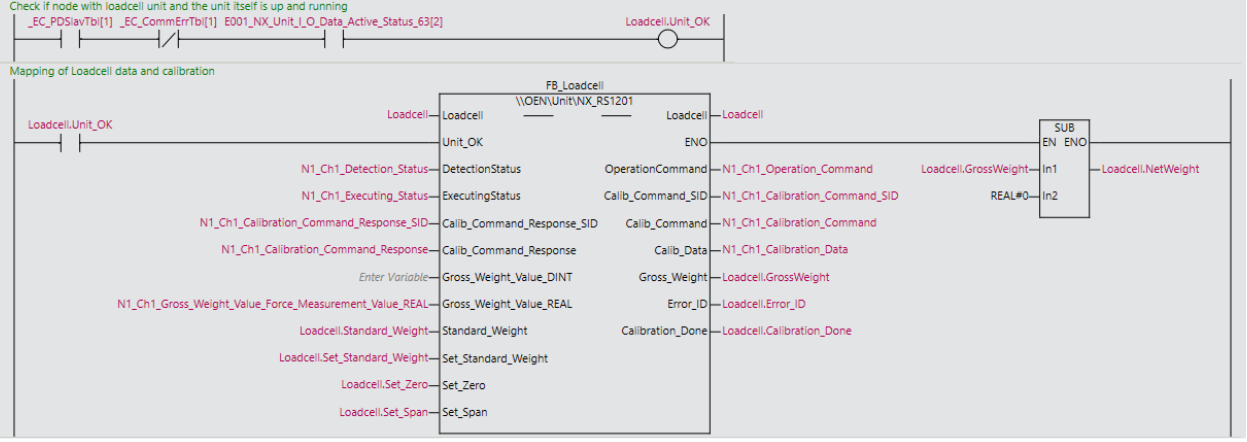
	Name
Omron - Norway	Kjell Baardsgaard

28.8. Structure



- ▼ Loadcell
  - ▼ DetectionStatus
    - Disconnected
    - UnderRange
    - OverRange
    - AD_ConversionError
    - ZeroPointRangeOver
    - Stable
  - ▶ ExecutingStatus
  - GrossWeight
  - NetWeight
  - ▶ OperationCommand
  - Standard_Weight
  - Unit_OK
  - Set_Zero
  - Error_ID
  - Calibration_Done
  - Set_Standard_Weight
  - Set_Span

28.9. Example

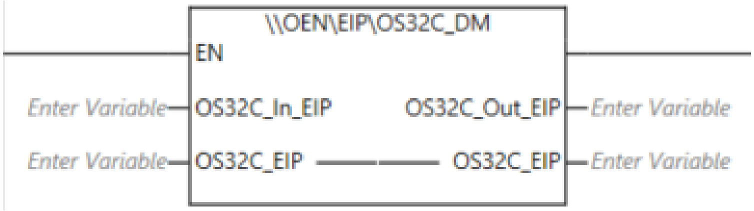

Check if node with loadcell unit and the unit itself is up and running

Mapping of Loadcell data and calibration

29. *OS32C_DM*

Reading data from safety scanner OS32C-DM via EthernetIP.

29.1. FN Layout



29.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
OS32C_In_EIP	OEN\nEIP\sOS32C_In_EIP	EIP Network data from OS32C

29.3. In-Out Variables

Name	Data type	Description
OS32C_EIP		

29.4. Output Variables

Name	Data Type	Description
Return	BOOL	EN is TRUE
OS32C_Out_EIP	OEN\nEIP\sOS32C_Out_EIP	EIP Network data to OS32C

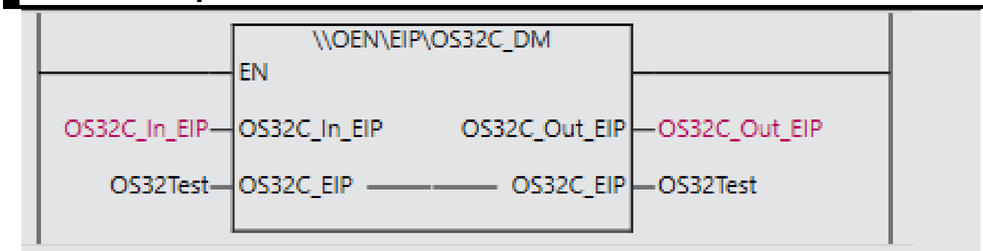
29.5. Revisions

Revision	In Library	Correction
1.0.0	5.00.9	

29.6. Credits

	Name
Omron - Norway	Kjell Baardsgaard

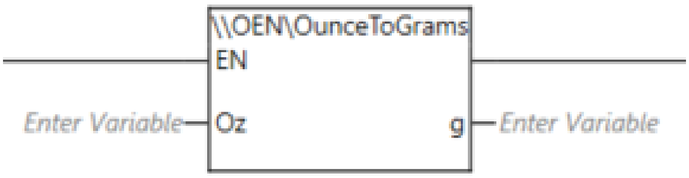
29.7. Example



30. *OunceToGrams*

Conversion of Ounce to Grams

30.1. FN Layout



30.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
Oz	REAL	

30.3. Output Variables

Name	Data Type	Description
Return	BOOL	EN is TRUE
g	REAL	

30.4. Revisions

Revision	In Library	Correction
1.1.0	5.00.9	

30.5. Credits

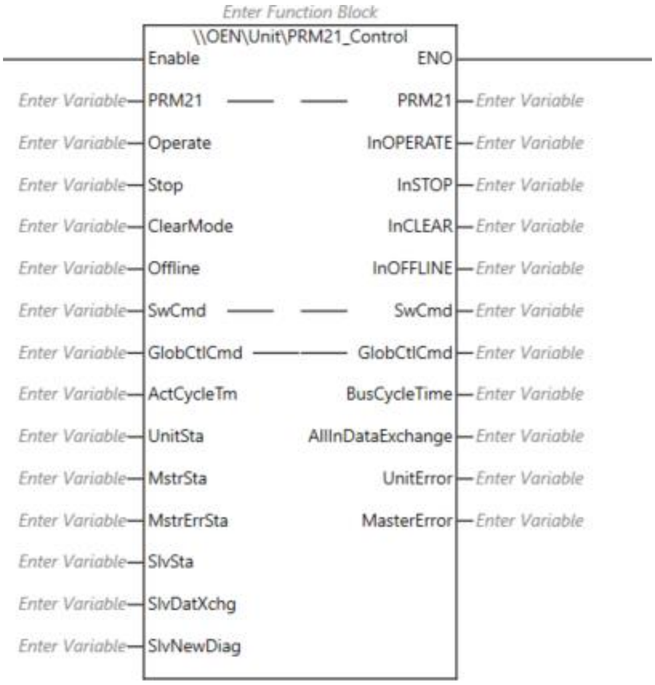
	Name
Omron - Norway	Kjell Baardsgaard

30.6. Example

31. *PRM21_Control*

Control and monitoring of Profibus Master CJ1W-PRM21 in NJ.
NX CPUs cannot be used.

31.1. FB Layout



31.2. Input Variables

Name	Data type	Description
Enable	BOOL	Enable function
Operate	BOOL	Profibus Operate Mode
Stop	BOOL	Profibus Stop Mode
ClearMode	BOOL	Profibus Clear Mode
Offline	BOOL	Profibus Offline
ActCycleTm	WORD	Variable from I/O Map
UnitSta	WORD	Variable from I/O Map
MstrSta	WORD	Variable from I/O Map
MstrErrSta	WORD	Variable from I/O Map
SlvSta	WORD	Variable from I/O Map
SlvDatXchg	ARRAY[1..125] OF BOOL	Variable from I/O Map
SlvNewDiag	ARRAY[1..125] OF BOOL	Variable from I/O Map

31.3. In-Out Variables

Name	Data type	Description
PRM21	OEN\nUnit\sPRM21	Structure to hold the Profibus data
SwCmd	WORD	Variable from I/O Map
GlobCtlCmd	WORD	Variable from I/O Map

31.4. Output Variables

Name	Data Type	Description
ENO	BOOL	EN is TRUE
InOPERATE	BOOL	Profibus In OPERATE
InSTOP	BOOL	Profibus In STOP
InCLEAR	BOOL	Profibus In CLEAR
InOFFLINE	BOOL	Profibus OFFLINE
BusCycleTime	REAL	Profibus Cycle Time
AllInDataExchange	BOOL	All slaves in Data Exchange
Uniterror	BOOL	Unit Error
MasterError	BOOL	Master Error

31.5. Errors

ErrorID	Error Description	Correction

31.6. Revisions

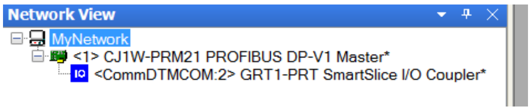
Revision	In Library	Correction
0.99.0	5.00.9	Beta version

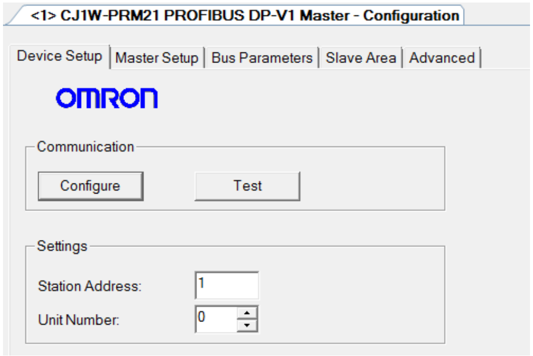
31.7. Credits

	Name
Omron - Norway	Kjell Baardsgaard

31.8. Example

Start CxConfigurator FDT and add the master and slaves.


Network View
MyNetwork
<1> CJ1W-PRM21 PROFIBUS DP-V1 Master*
<CommDTMCOM:2> GRT1-PRT SmartSlice I/O Coupler*



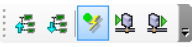
Station Address: 1

Unit Number: 0

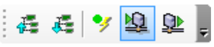
Station Address is the master's Profibus address. Feel free to use #1.
UnitNumber is the wheel at the front of the module and is the module's Rack address on the PLC.
Select the communication type against the PLC with the "Configure" button.
"Test" that it works afterwards.

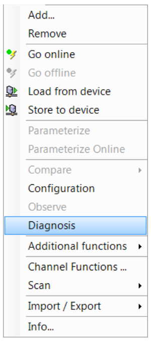
If the slaves are configured correctly, they will be autoaddressed in the master. Check this out under "Slave Area."
Press the "Save" button in the lower-right corner.

Go Online with the Master.



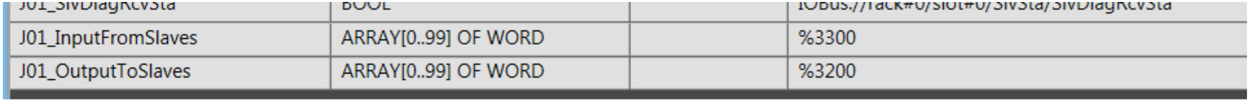
Transfer configuration





Use the Diagnostic tool and see that everything is running correctly.

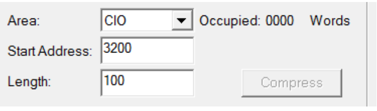
When auto addressing, data enters the CIO3300- addresses and from the CIO3200- addresses. We need to catch this in NJ. Create variables that fit Default Setup this way:



J01_SlvDiagRcvSta	BOOL		IOBus://rack#0/slot#0/SlvSta/SlvDiagRcvSta
J01_InputFromSlaves	ARRAY[0..99] OF WORD		%3300
J01_OutputToSlaves	ARRAY[0..99] OF WORD		%3200

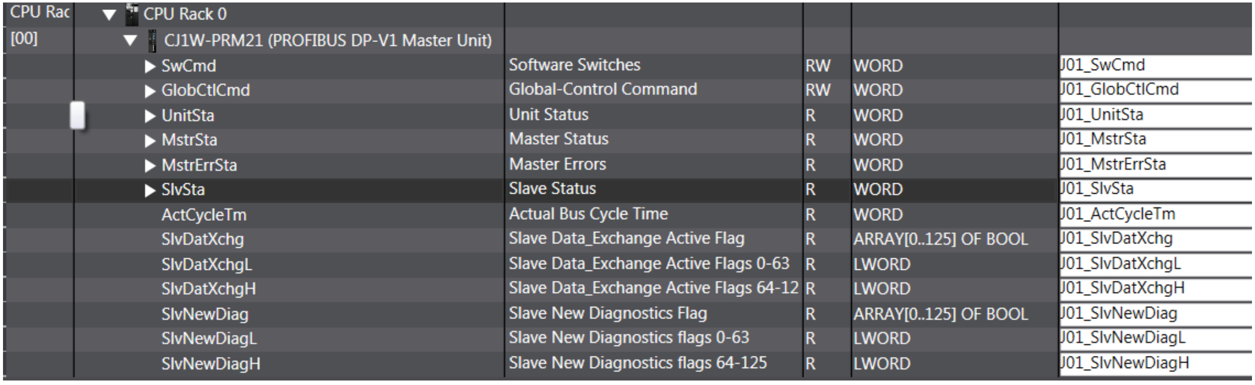
These will contain data from/to the Profibus slaves.

The number of words you choose yourself, but CxConfigurator reserves only 100 addresses as default.



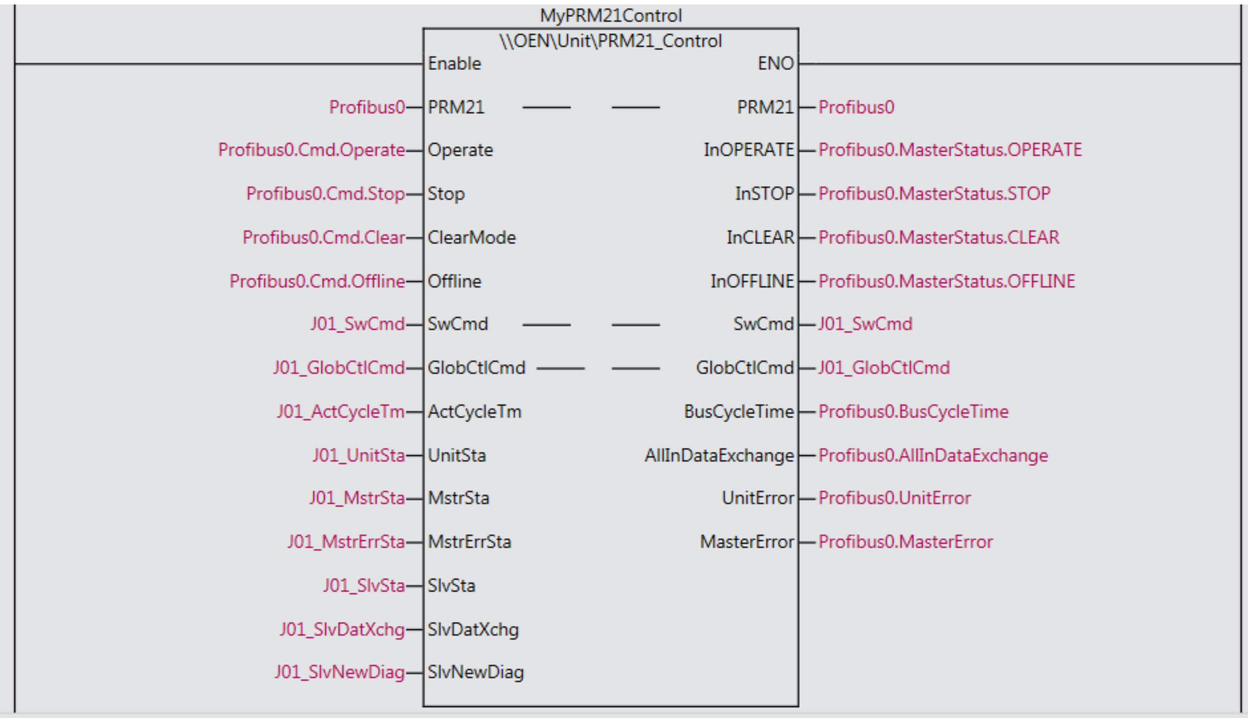
Area: CIO   Occupied: 0000 Words
Start Address: 3200
Length: 100
Compress

Autogenerate variables in IOMap in SysmacStudio.

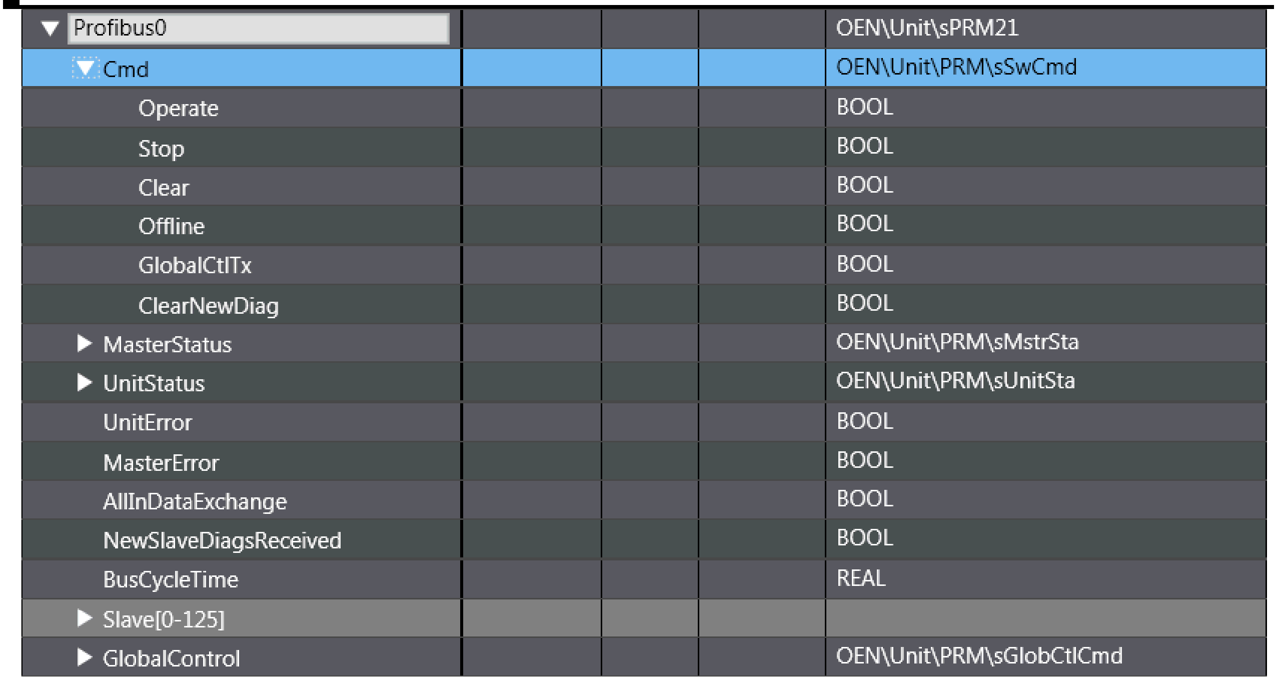


CPU Rac	CPU Rack 0				
[00]	CJ1W-PRM21 (PROFIBUS DP-V1 Master Unit)				
	SwCmd	Software Switches	RW	WORD	J01_SwCmd
	GlobCtlCmd	Global-Control Command	RW	WORD	J01_GlobCtlCmd
	UnitSta	Unit Status	R	WORD	J01_UnitSta
	MstrSta	Master Status	R	WORD	J01_MstrSta
	MstrErrSta	Master Errors	R	WORD	J01_MstrErrSta
	SlvSta	Slave Status	R	WORD	J01_SlvSta
	ActCycleTm	Actual Bus Cycle Time	R	WORD	J01_ActCycleTm
	SlvDatXchg	Slave Data_Exchange Active Flag	R	ARRAY[0..125] OF BOOL	J01_SlvDatXchg
	SlvDatXchgL	Slave Data_Exchange Active Flags 0-63	R	LWORD	J01_SlvDatXchgL
	SlvDatXchgH	Slave Data_Exchange Active Flags 64-12	R	LWORD	J01_SlvDatXchgH
	SlvNewDiag	Slave New Diagnostics Flag	R	ARRAY[0..125] OF BOOL	J01_SlvNewDiag
	SlvNewDiagL	Slave New Diagnostics flags 0-63	R	LWORD	J01_SlvNewDiagL
	SlvNewDiagH	Slave New Diagnostics flags 64-125	R	LWORD	J01_SlvNewDiagH

Add these to the function block to collect all data in a data structure (PRM21) and to get a better overview. The PRM21 data structure is also a great way to get status data into a NA operator panel.


MyPRM21Control
\\OEN\Unit\PRM21_Control
Enable
ENO
Profibus0 — PRM21
PRM21 — Profibus0
Profibus0.Cmd.Operate — Operate
InOPERATE — Profibus0.MasterStatus.OPERATE
Profibus0.Cmd.Stop — Stop
InSTOP — Profibus0.MasterStatus.STOP
Profibus0.Cmd.Clear — ClearMode
InCLEAR — Profibus0.MasterStatus.CLEAR
Profibus0.Cmd.Offline — Offline
InOFFLINE — Profibus0.MasterStatus.OFFLINE
J01_SwCmd — SwCmd
SwCmd — J01_SwCmd
J01_GlobCtlCmd — GlobCtlCmd
GlobCtlCmd — J01_GlobCtlCmd
J01_ActCycleTm — ActCycleTm
BusCycleTime — Profibus0.BusCycleTime
J01_UnitSta — UnitSta
AllInDataExchange — Profibus0.AllInDataExchange
J01_MstrSta — MstrSta
UnitError — Profibus0.UnitError
J01_MstrErrSta — MstrErrSta
MasterError — Profibus0.MasterError
J01_SlvSta — SlvSta
J01_SlvDatXchg — SlvDatXchg
J01_SlvNewDiag — SlvNewDiag

31.9. Structure

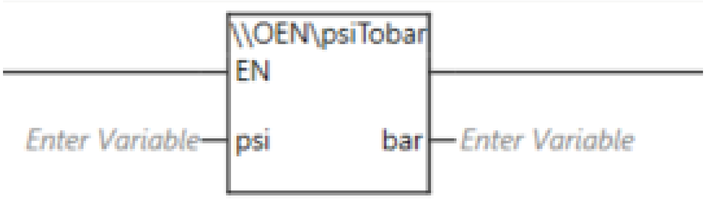


▼ Profibus0				OEN\Unit\sPRM21
▼ Cmd				OEN\Unit\PRM\sSwCmd
Operate				BOOL
Stop				BOOL
Clear				BOOL
Offline				BOOL
GlobalCtlTx				BOOL
ClearNewDiag				BOOL
▶ MasterStatus				OEN\Unit\PRM\sMstrSta
▶ UnitStatus				OEN\Unit\PRM\sUnitSta
UnitError				BOOL
MasterError				BOOL
AllInDataExchange				BOOL
NewSlaveDiagsReceived				BOOL
BusCycleTime				REAL
▶ Slave[0-125]				
▶ GlobalControl				OEN\Unit\PRM\sGlobCtlCmd

32. *psiTobar*

Conversion of psi to bar.

32.1. FN Layout



32.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
psi	REAL	

32.3. Output Variables

Name	Data Type	Description
ENO	BOOL	EN is TRUE
bar	REAL	

32.4. Errors

ErrorID	Error Description	Correction

32.5. Revisions

Revision	In Library	Correction
1.1.0	5.00.9	

32.6. Credits

	Name
Omron - Norway	Kjell Baardsgaard

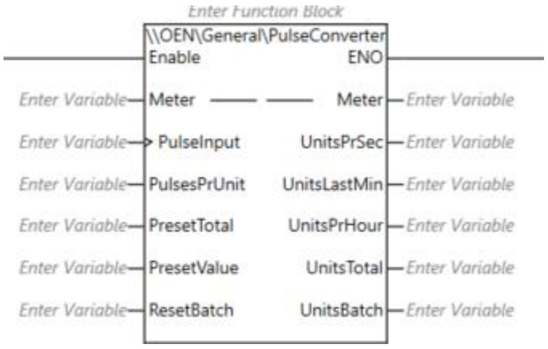
32.7. Example

33. *PulseConverter*

This block converts pulses to units/unit of time and total quantity. Typically used for flowmeter and energy measurement.

The total can be preset to a selected value and Batch can be reset. All data can be monitored in the Meter structure. Note the arrow on *PulseInput* indicating differential input.

33.1. FB Layout



33.2. Input Variables

Name	Data type	Description
Enable	BOOL	Enable function
PulseInput	BOOL	Pulses from metering instrument
PulsesPrUnit	REAL	Number of pulses per unit. Ex. Pulses/liter
PresetTotal	BOOL	Move PresetValue to UnitsTotal
PresetValue	REAL	Value to be set in UnitsTotal using PresetTotal
ResetBatch	BOOL	Set UnitsBatch to 0.

33.3. In-Out Variables

Name	Data type	Description
Meter	OEN\nGeneral\sPulseMeter	Output variables as a structure. See 33.8

33.4. Output Variables

Name	Data Type	Description
ENO	BOOL	Enable is TRUE
UnitsPrSec	REAL	Ex. Liters/second
UnitsLastMin	REAL	Ex. Liters last minute
UnitsPrHour	REAL	Ex. Liters/hour
UnitsTotal	REAL	Ex. Total liters
UnitsBatch	REAL	Ex. Total liters since last ResetBatch

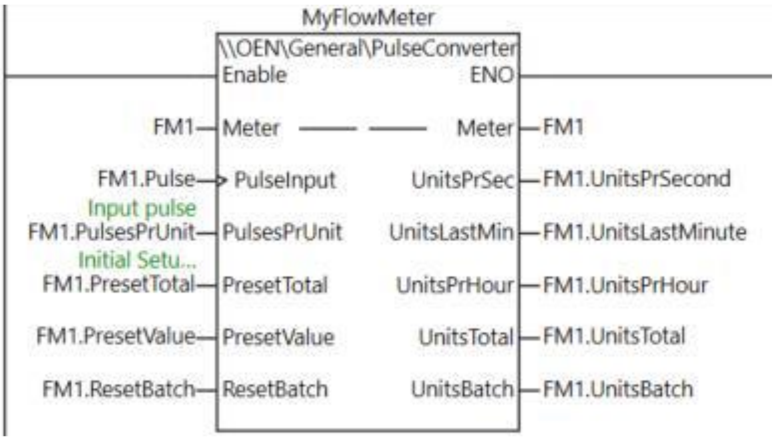
33.5. Revisions

Revision	In Library	Correction
1.6.0	5.00.9	

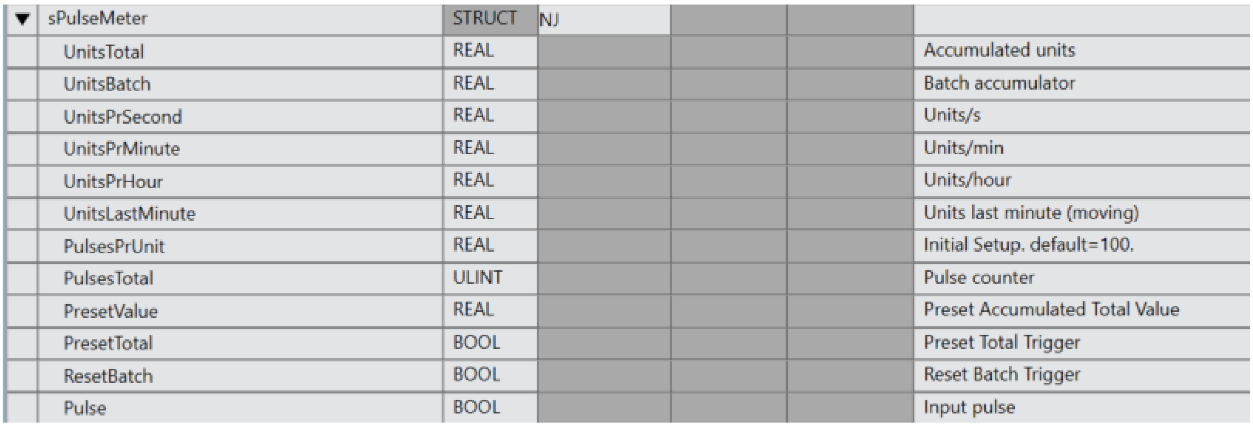
33.6. Credits

	Name
Omron - Norway	Kjell Baardsgaard

33.7. Example



33.8. Structure

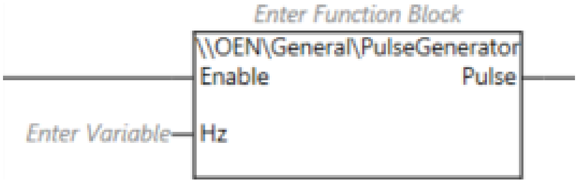


▼ sPulseMeter	STRUCT	NJ			
UnitsTotal	REAL				Accumulated units
UnitsBatch	REAL				Batch accumulator
UnitsPrSecond	REAL				Units/s
UnitsPrMinute	REAL				Units/min
UnitsPrHour	REAL				Units/hour
UnitsLastMinute	REAL				Units last minute (moving)
PulsesPrUnit	REAL				Initial Setup. default=100.
PulsesTotal	ULINT				Pulse counter
PresetValue	REAL				Preset Accumulated Total Value
PresetTotal	BOOL				Preset Total Trigger
ResetBatch	BOOL				Reset Batch Trigger
Pulse	BOOL				Input pulse

34. *PulseGenerator*

The output provides 50/50 pulses at a specified frequency. Please note that the PLC CycleTime will affect the accuracy of the frequency.

34.1. FB Layout



34.2. Input Variables

Name	Data type	Valid Range	Description
Enable	BOOL		Enable function
Hz	REAL		Frequency

34.3. Output Variables

Name	Data Type	Description
Pulse	BOOL	Pulses 50/50 Duty based on Frequency input

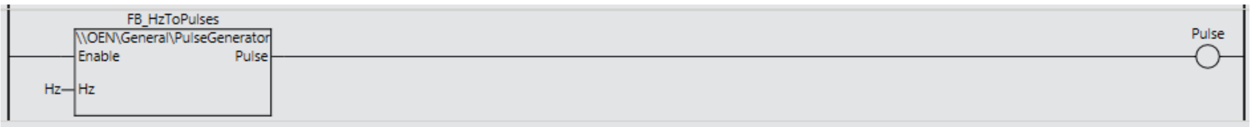
34.4. Revisions

Revision	In Library	Correction
1.1.0	5.00.9	

34.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

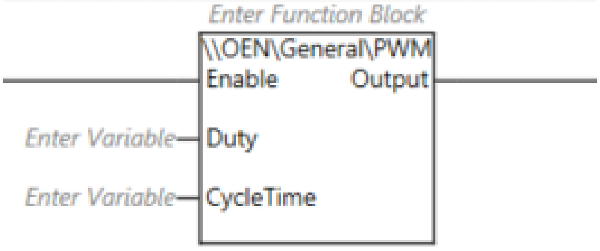
34.6. Example



35. *PWM*

A function block for pulse width modulated output. Duty is set in % and determines how large part of CycleTime Output should be on. By setting CycleTime to low values, the output can be used for dimming light. A CycleTime approaching the PLC's CycleTime will result in poor resolution/inaccuracy.

35.1. FB Layout



35.2. Input Variables

Name	Data type	Valid Range	Description
Enable	BOOL		Enable function
Duty	REAL		% part of CycleTime to be on
CycleTime	TIME		Time between each rising edge of output

35.3. Output Variables

Name	Data Type	Description
Output	BOOL	Pulse output

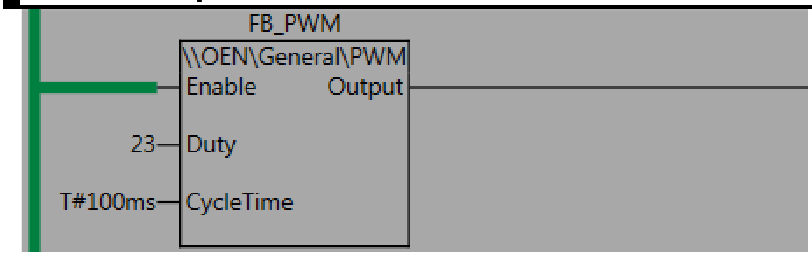
35.4. Revisions

Revision	In Library	Correction
1.1.0	5.00.9	

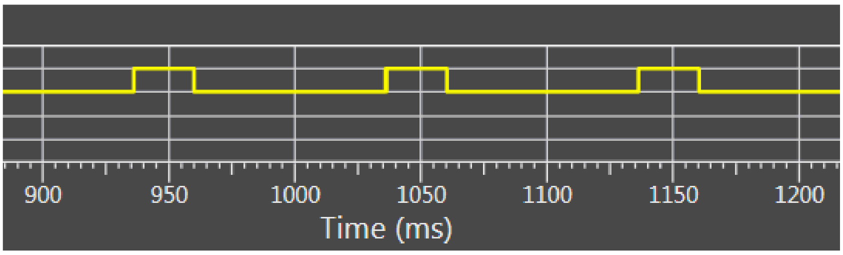
35.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

35.6. Example



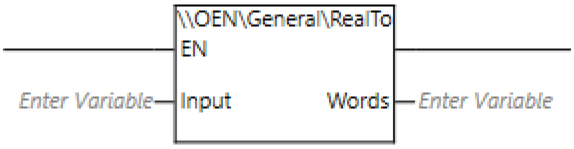
10Hz with PLC CycleTime 4ms:



36. *RealTo*

Conversion from Real to 2 Words

36.1. FN Layout



36.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
Input	REAL	

36.3. Output Variables

Name	Data Type	Description
Return	BOOL	
Words	ARRAY[0..1] OF WORDS	

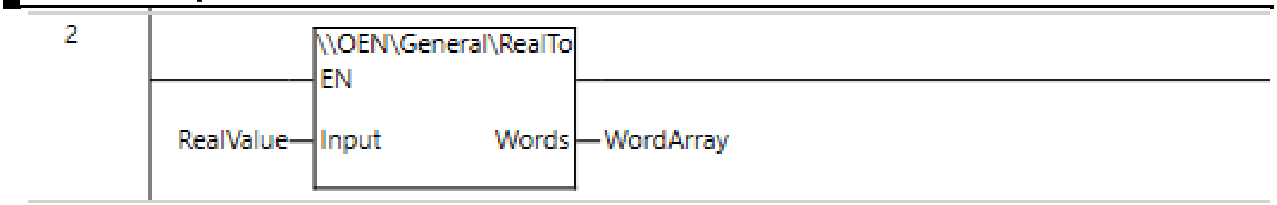
36.4. Revisions

Revision	In Library	Correction
1.0.0	5.00.9	

36.1. Credits

	Name
Omron - Norway	Kjell Baardsgaard

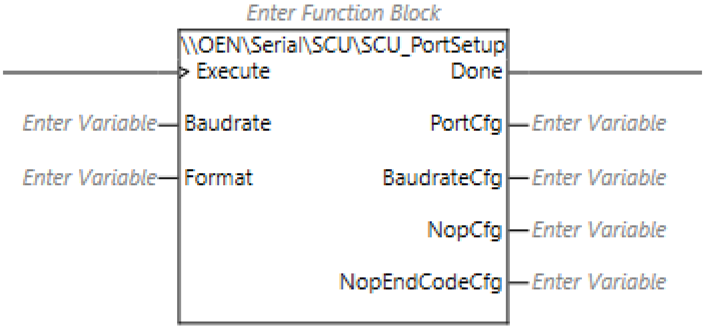
36.2. Example



37. *SCU_PortSetup*

A function block to simplify the setup of an SCU port.
All 4 outputs on the block must be connected to the SCU module's variables in IOMap.
NX CPUs cannot be used.

37.1. FB Layout



37.2. Input Variables

Name	Data type	Valid Range	Description
Execute	BOOL		Enable function
Baudrate	DINT	9600..230400	
Format	STRING[4]	7E1,7E2, 8N1,8E1	

StartCode is Default.
EndCode is set to <CR><LF>.

37.3. Output Variables

Name	Data Type	Description
Done	BOOL	Setup done
PortCfg	WORD	Connect this to the I/O Map variable
BaudrateCfg	USINT	Connect this to the I/O Map variable
NopCfg	WORD	Connect this to the I/O Map variable
NopEndCodeCfg	USINT	Connect this to the I/O Map variable

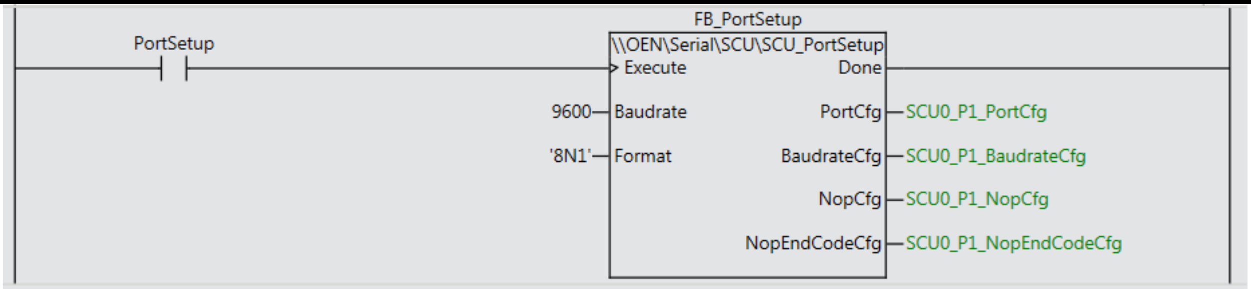
37.4. Revisions

Revision	In Library	Correction
0.1.0	5.00.9	

37.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

37.6. Example



FB_PortSetup

PortSetup

\\OEN\Serial\SCU\SCU_PortSetup

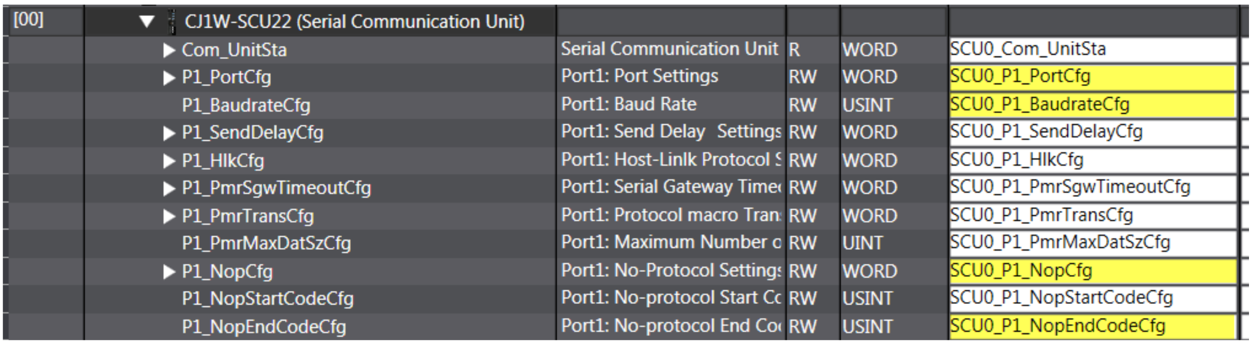
Execute — Done

9600 — Baudrate — PortCfg — SCU0_P1_PortCfg

'8N1' — Format — BaudrateCfg — SCU0_P1_BaudrateCfg

NopCfg — SCU0_P1_NopCfg

NopEndCodeCfg — SCU0_P1_NopEndCodeCfg

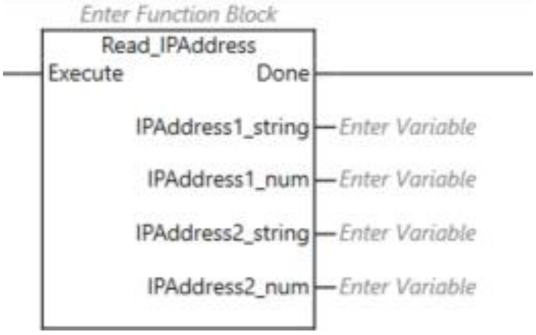


[00]	▼ CJ1W-SCU22 (Serial Communication Unit)				
	▶ Com_UnitSta	Serial Communication Unit	R	WORD	SCU0_Com_UnitSta
	▶ P1_PortCfg	Port1: Port Settings	RW	WORD	SCU0_P1_PortCfg
	P1_BaudrateCfg	Port1: Baud Rate	RW	USINT	SCU0_P1_BaudrateCfg
	▶ P1_SendDelayCfg	Port1: Send Delay Settings	RW	WORD	SCU0_P1_SendDelayCfg
	▶ P1_HlkCfg	Port1: Host-Linlk Protocol S	RW	WORD	SCU0_P1_HlkCfg
	▶ P1_PmrSgwTimeoutCfg	Port1: Serial Gateway Time	RW	WORD	SCU0_P1_PmrSgwTimeoutCfg
	▶ P1_PmrTransCfg	Port1: Protocol macro Tran	RW	WORD	SCU0_P1_PmrTransCfg
	P1_PmrMaxDatSzCfg	Port1: Maximum Number o	RW	UINT	SCU0_P1_PmrMaxDatSzCfg
	▶ P1_NopCfg	Port1: No-Protocol Setting	RW	WORD	SCU0_P1_NopCfg
	P1_NopStartCodeCfg	Port1: No-protocol Start Cc	RW	USINT	SCU0_P1_NopStartCodeCfg
	P1_NopEndCodeCfg	Port1: No-protocol End Co	RW	USINT	SCU0_P1_NopEndCodeCfg

38. *Read_IPAddress*

Read IP address of local ethernet ports. First Execute reads Port#1 next execute reads Port#2.

38.1. FN Layout



38.2. Input Variables

Name	Data type	Valid Range	Default	Description
Execute	BOOL		FALSE	Execute function

38.3. Output Variables

Name	Data Type	Description
IPAddress1_string	STRING[16]	IP address of Ethernet Port#1
IPAddress1_num	ARRAY[0..3] OF UINT	IP address of Ethernet Port#1
IPAddress2_string	STRING[16]	IP address of Ethernet Port#2
IPAddress2_num	ARRAY[0..3] OF UINT	IP address of Ethernet Port#2

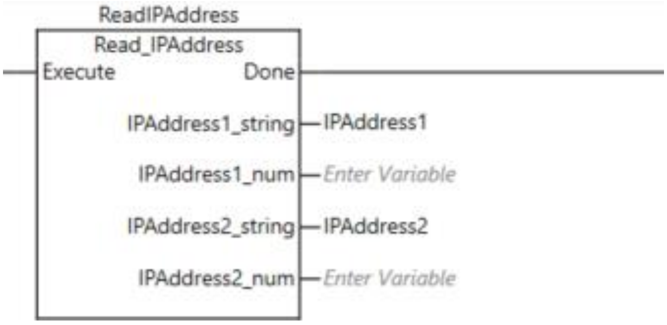
38.4. Revisions

Revision	In Library	Correction
1.0.0	5.00.9	
2.0.0	5.00.11	Added reading of Port#2

38.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard
Omron - France	Arnaud Croizit

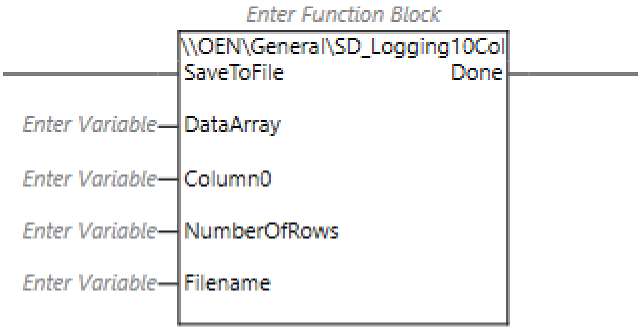
38.6. Example


ReadIPAddress
Read_IPAddress
Execute
Done
IPAddress1_string
IPAddress1
IPAddress1_num
Enter Variable
IPAddress2_string
IPAddress2
IPAddress2_num
Enter Variable

39. *SD_Logging10Col*

This function block creates a file on SD Memory Card in the CPU. See also TableLogging10Col.

39.1. FB Layout



39.2. Input Variables

Name	Data type	Description
SaveToFile	BOOL	Save data. Use EDGE Trigger
DataArray	ARRAY[1..9,0..9] OF INT	DataTable that contains data to be saved
Column0	ARRAY[0..9} OF STRING[20]	String data to insert into first Column in each row
NumberOfRows	UINT	Number of rows from DataArray to save
Filename	STRING[256]	Filename for SD Memory Card

Number of saves to a MemoryCard is limited to 50k-100k saves. Do not save every cycle!

39.3. Output Variables

Name	Data Type	Description
Done	BOOL	Saving complete

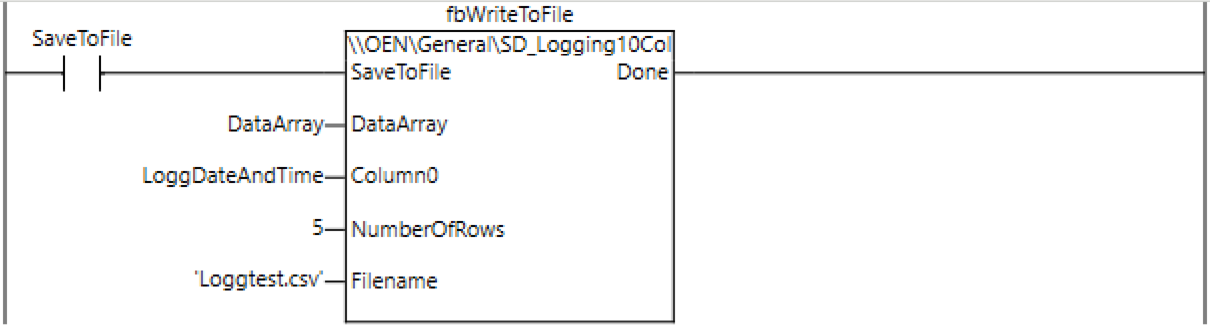
39.4. Revisions

Revision	In Library	Correction
1.0.1	5.00.9	

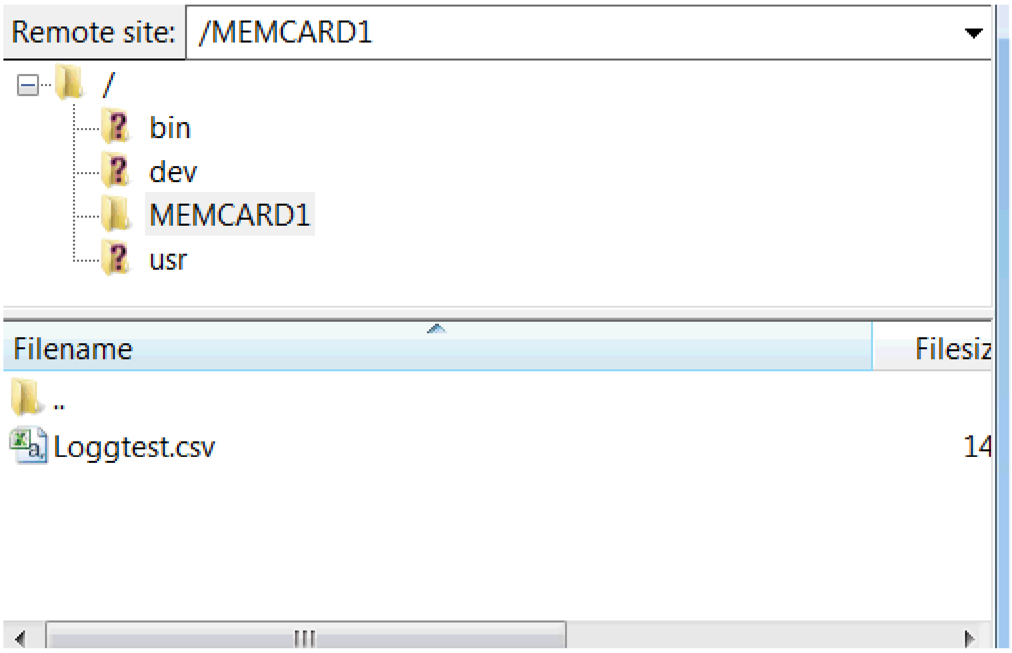
39.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

39.6. Example



When SaveToFile, the file will appear in MEMCARD1 folder as Loggtest.csv

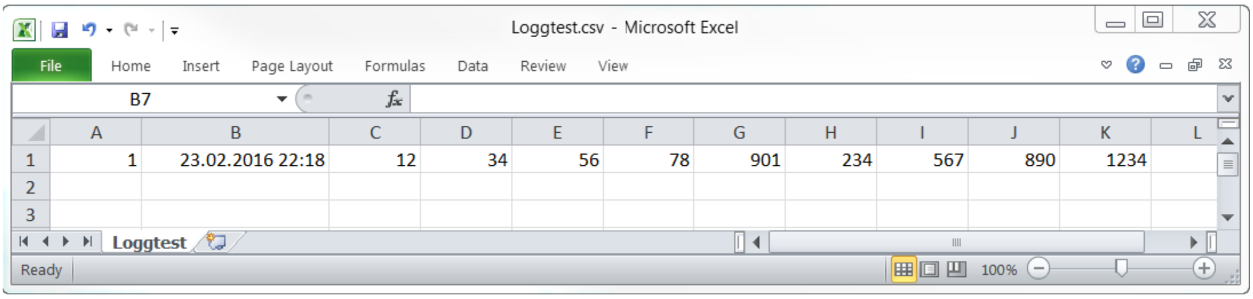


Remote site: /MEMCARD1

- /
  - bin
  - dev
  - MEMCARD1
  - usr

Filename	Filesiz
..	
Loggtest.csv	14

Open the file in Excel and you will see your data in Column B (0) through K.

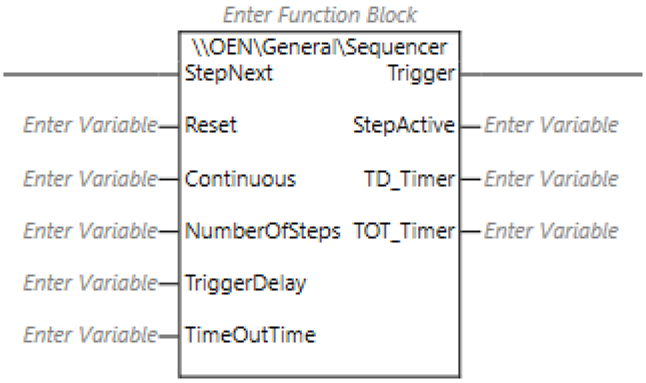


	A	B	C	D	E	F	G	H	I	J	K	L
1	1	23.02.2016 22:18	12	34	56	78	901	234	567	890	1234	
2												
3												

40. *Sequencer*

A Function Block that can be used to run sequences.

40.1. FB Layout



40.2. Input Variables

Name	Data type	Default	Description
StepNext	BOOL		Pulse input to index to next Step
Reset	BOOL		Set StepActive to 0
Continuous	BOOL		If FALSE stop at StepNext=NumberOfsteps+1 If TRUE return StepNext to 1 after reaching NumberOfSteps
NumberOfSteps	INT	0	Number of steps in the sequencer
TriggerDelay	TIME	T#0s	Delay Trigger after StepNext
TimeOutTime	TIME	T#0s	If >0 then StepNext if no StepNext input

40.3. Output Variables

Name	Data Type	Description
Trigger	BOOL	Pulse output when StepActive changes
StepActive	INT	Active Step number
TD_Timer	TIME	TriggerDelay elapsed time
TOT_Timer	TIME	TimeOutTime elapsed time

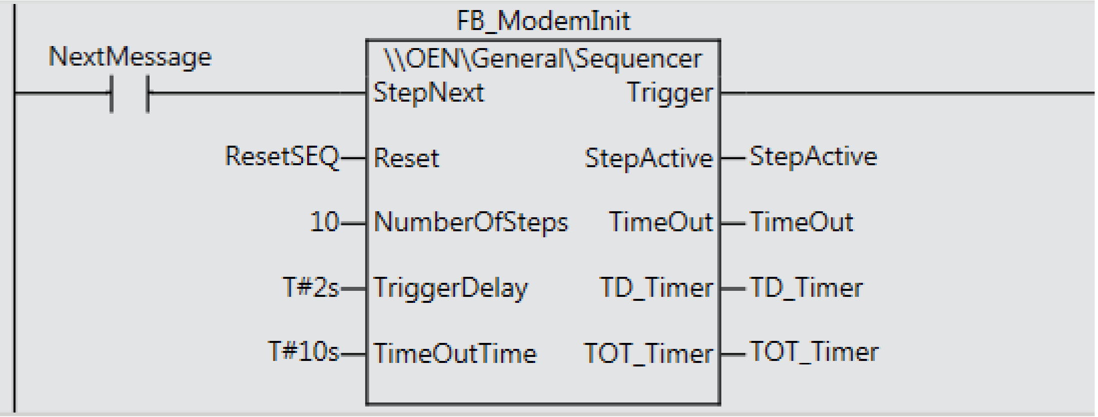
40.4. Revisions

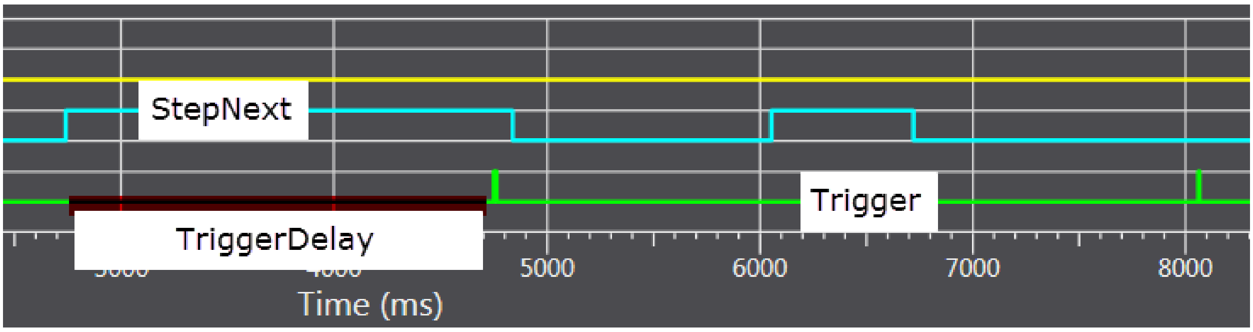
Revision	In Library	Correction
1.1.0	5.00.9	

40.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

40.6. Example

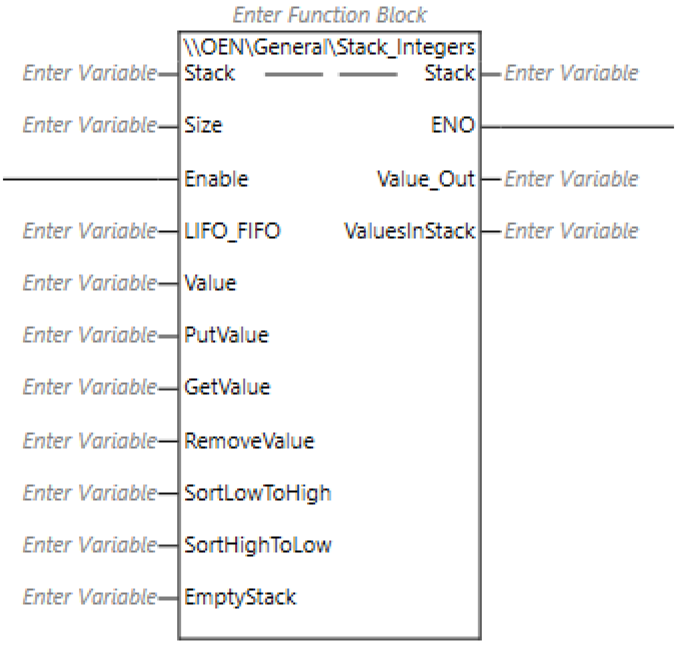

FB_ModemInit
\\OEN\General\Sequencer
NextMessage
StepNext
Trigger
ResetSEQ
Reset
StepActive
StepActive
10
NumberOfSteps
TimeOut
TimeOut
T#2s
TriggerDelay
TD_Timer
TD_Timer
T#10s
TimeOutTime
TOT_Timer
TOT_Timer


StepNext
Trigger
TriggerDelay
3000
4000
5000
6000
7000
8000
Time (ms)

41. *Stack_Integers*

A function block for storing, sorting, and filtering integer data.

41.1. FB Layout



41.2. Input Variables

Name	Data type	Default	Description
Size	UINT		Stack Size
Enable	BOOL		Enable function
LIFO_FIFO	BOOL		If FALSE, last value added moved out using GetValue
Value	INT		Value to add to the end of the stored list of values
PutValue	BOOL		Trigger to add Value to the list
GetValue	BOOL		Move first or last value from list and output to Value_out
RemoveValue	BOOL		Remove values in the list equal to Value
SortLowToHigh	BOOL		Sort values in the list Lowest to Highest
SortHighToLow	BOOL		Sort values in the list Highest to Lowest
EmptyStack	BOOL		Set values in Stack and ValuesInStack to 0

41.3. In-Out Variables

Name	Data type	Description
Stack	ARRAY[*] OF INT	Array to hold values

41.4. Output Variables

Name	Data Type	Description
ENO	BOOL	
Value_Out	INT	Value moved from the list using GetValue
ValuesInStack	UINT	Number of values added to Stack

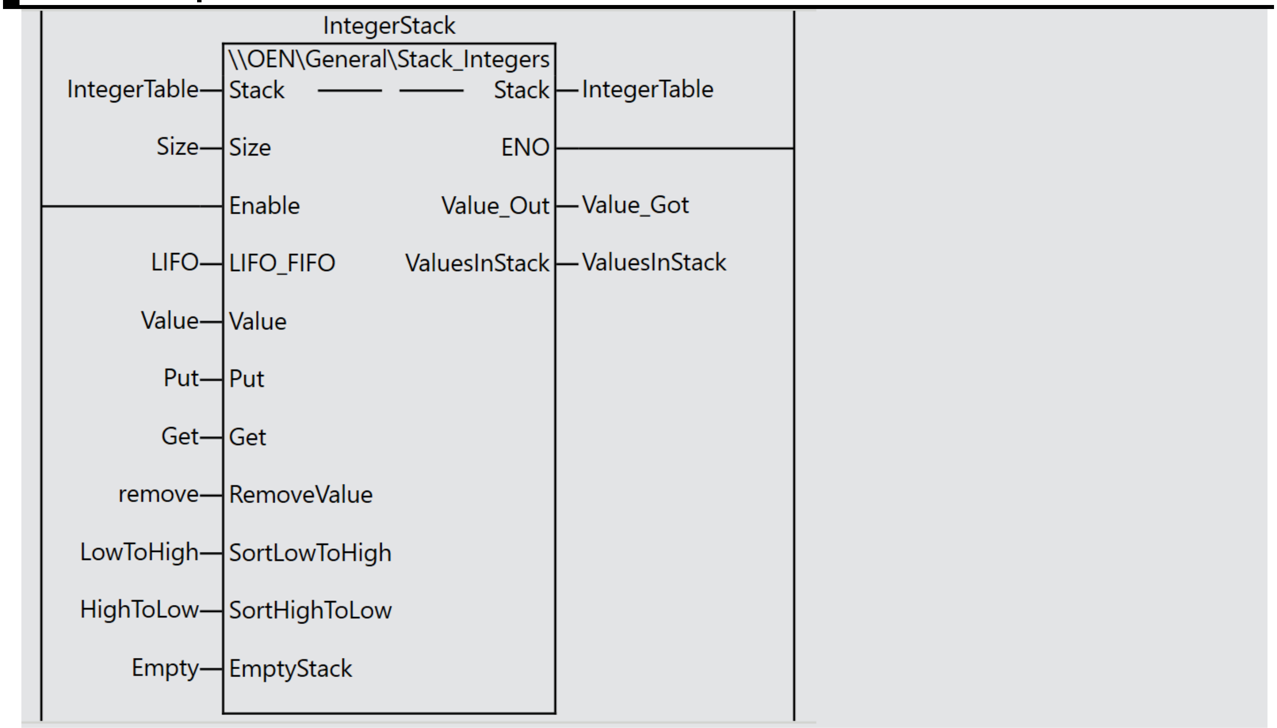
41.5. Revisions

Revision	In Library	Correction
1.1.1	5.00.9	

41.6. Credits

	Name
Omron - Norway	Kjell Baardsgaard

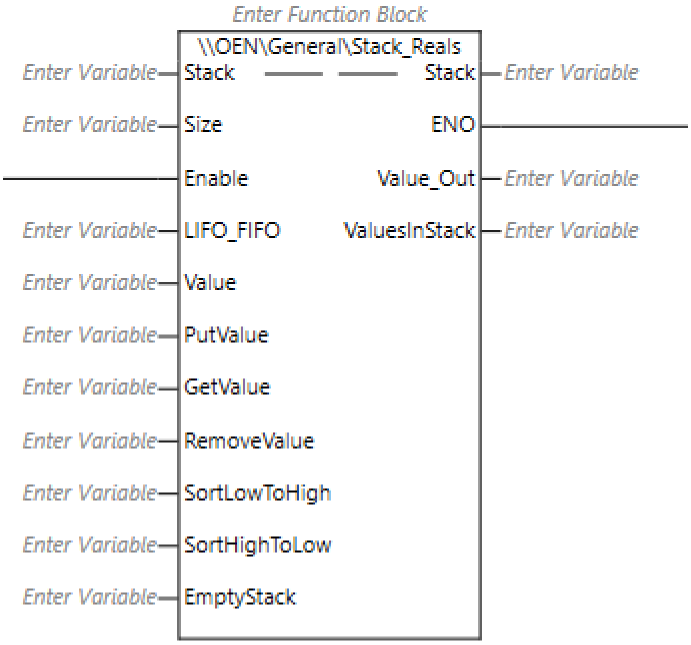
41.7. Example



42. *Stack_Reals*

A function block for storing, sorting, and filtering real data.

42.1. FB Layout



42.2. Input Variables

Name	Data type	Description
Size	UINT	Stack Size
Enable	BOOL	Enable function
LIFO_FIFO	BOOL	If FALSE, last value added moved out using GetValue
Value	REAL	Value to add to the end of the stored list of values
PutValue	BOOL	Trigger to add Value to the list
GetValue	BOOL	Move first or last value from list and output to Value_out
RemoveValue	BOOL	Remove values in the list equal to Value
SortLowToHigh	BOOL	Sort values in the list Lowest to Highest
SortHighToLow	BOOL	Sort values in the list Highest to Lowest
EmptyStack	BOOL	Set values in Stack and ValuesInStack to 0

42.3. In-Out Variables

Name	Data type	Description
Stack	ARRAY[*] OF REAL	Array to hold values

42.4. Output Variables

Name	Data Type	Description
ENO	BOOL	
Value_Out	REAL	Value moved from the list using GetValue
ValuesInStack	UINT	Number of values added to Stack

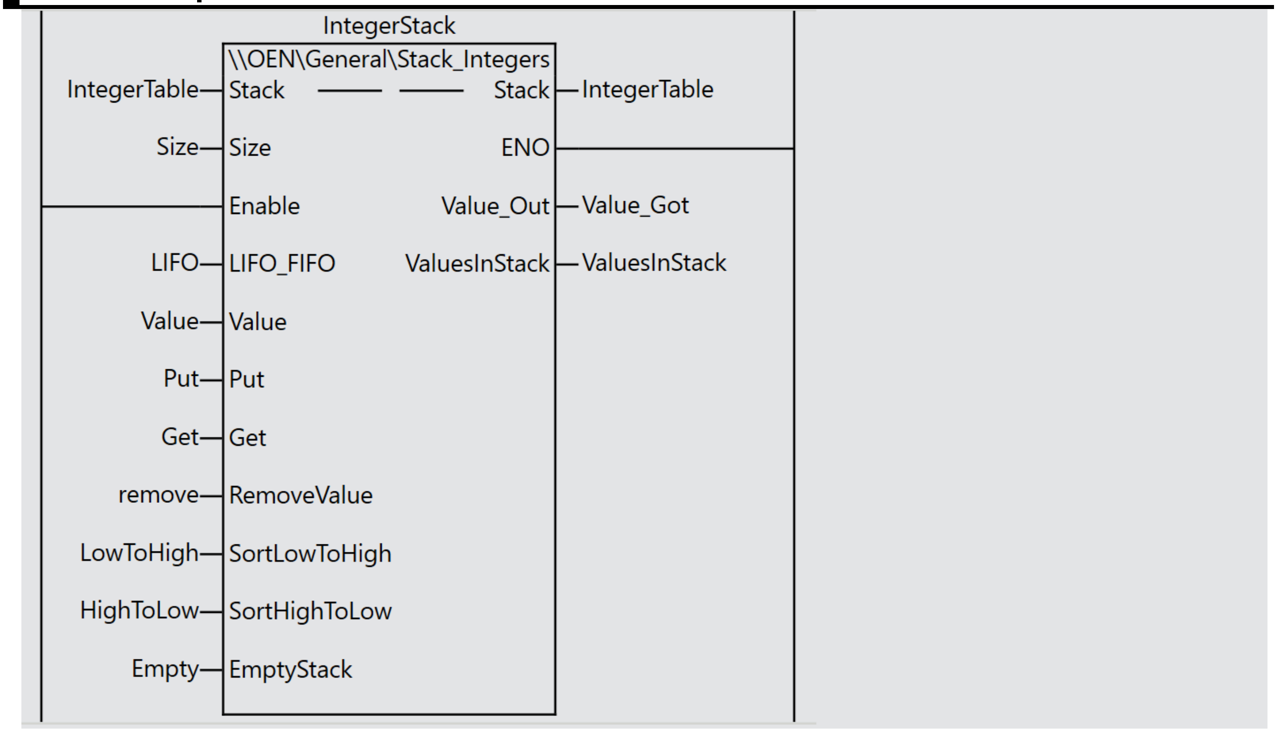
42.5. Revisions

Revision	In Library	Correction
1.0.1	5.00.9	

42.6. Credits

	Name
Omron - Norway	Kjell Baardsgaard

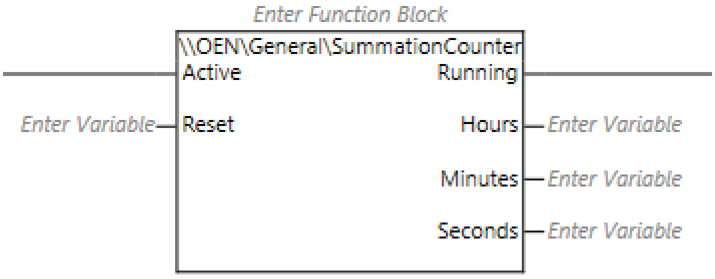
42.7. Example



43. *SummationCounter*

A function block that can be used for monitoring operating hours. The values are retained in the event of a power failure.

43.1. FB Layout



43.2. Input Variables

Name	Data type	Valid Range	Default	Description
Active	BOOL		FALSE	If TRUE then Count time
Reset	BOOL			Reset time

43.3. Output Variables

Name	Data Type	Description
Running	BOOL	Active is TRUE
Hours	UDINT	Total active hours
Minutes	UINT	
Seconds	UINT	

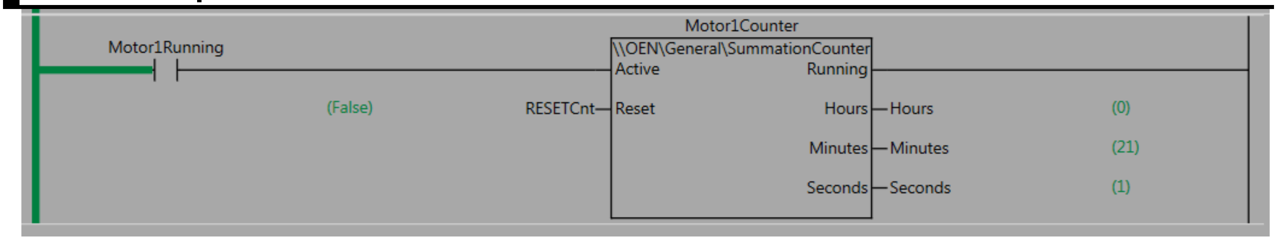
43.4. Revisions

Revision	In Library	Correction
1.1.0	5.00.9	

43.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

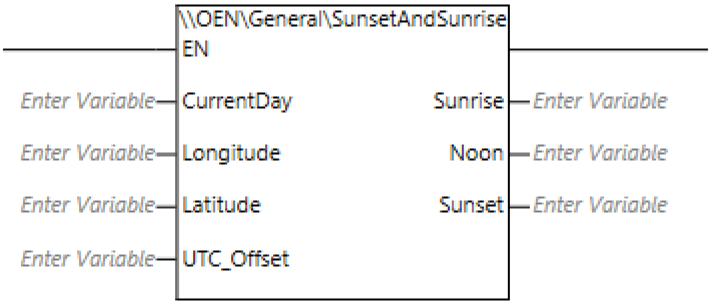
43.6. Example



44. *SunsetAndSunrise*

A function block that gives you the time of sunrise, sunset and when the sun is highest above the horizon.

44.1. FN Layout



44.2. Input Variables

Name	Data type	Valid Range	Description
EN	BOOL		Enable function
CurrentDay	UINT	1-366	Day of year to determine SunsetAndSunrise. You can use CurrentDay FB.
Longitude	LREAL	-180,+180	Degrees, East=Negative
Latitude	LREAL	-180,+180	Degrees, South=Negative
UTC_Offset	INT	-12,+12	Local UTC offset. GMT=0.

44.3. Output Variables

Name	Data Type	Description
Return	BOOL	
Sunrise	TIME	Sunrise on location the day input
Noon	TIME	Noon on location the day input
Sunset	TIME	Sunset on location the day input

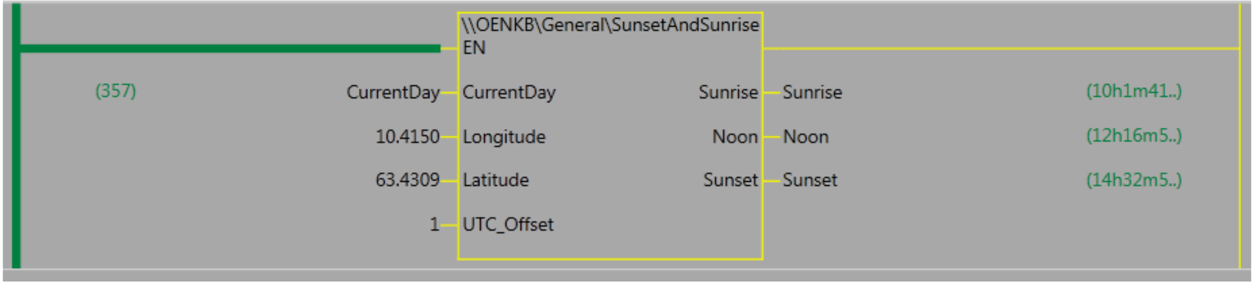
44.4. Revisions

Revision	In Library	Correction
1.1.1	5.00.9	

44.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard
	http://www.esrl.noaa.gov/gmd/grad/solcalc/solareqns.PDF

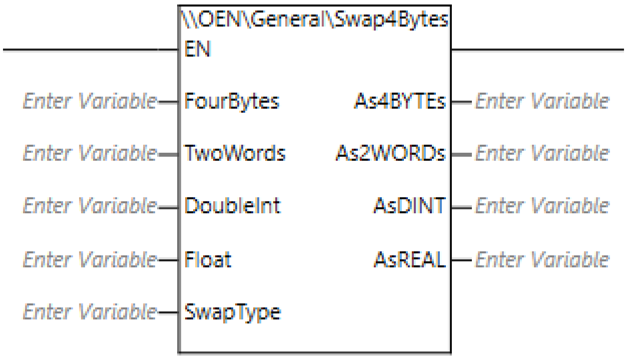
44.6. Example


\\OENKB\General\SunsetAndSunrise
EN
(357)
CurrentDay
CurrentDay
Sunrise
Sunrise
(10h1m41..)
10.4150
Longitude
Noon
Noon
(12h16m5..)
63.4309
Latitude
Sunset
Sunset
(14h32m5..)
1
UTC_Offset

45. *Swap4Bytes*

An FB to swap bytes and words. This is useful for data communication where the bytes do not come in the desired order. This is relatively common for Profibus, CAN and Modbus.

45.1. FN Layout



45.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
FourBytes	ARRAY[0..3] OF BYTE	Four bytes to Swap or
TwoWords	ARRAY[0..1] OF WORD	Two words to Swap or
DoubleInt	DINT	Double Integer to Swap or
Float	REAL	Real to Swap
SwapType	OEN\eSwap	#Bytes switches 2 and 2 bytes. ABCD becomes BADC #Words replaces the first two bytes with the last two. ABCD becomes CDAB. #BytesAndWords does both. ABCD will be DCBA.

45.3. Output Variables

Name	Data Type	Description
Return	BOOL	
As4BYTEs	ARRAY[0..3] OF BYTE	Swap result as 4 Bytes
As2WORDs	ARRAY[0..1] OF WORD	Swap result as 2 Words
AsDINT	DINT	Swap result as Double Integer
AsREAL	REAL	Swap result as Real

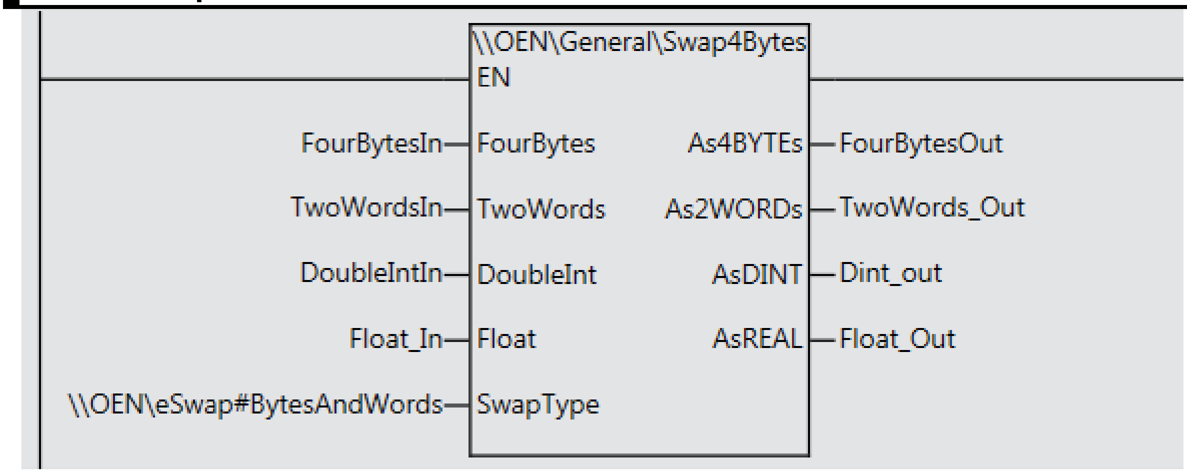
45.4. Revisions

Revision	In Library	Correction
1.1.0	5.00.9	

45.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

45.6. Example

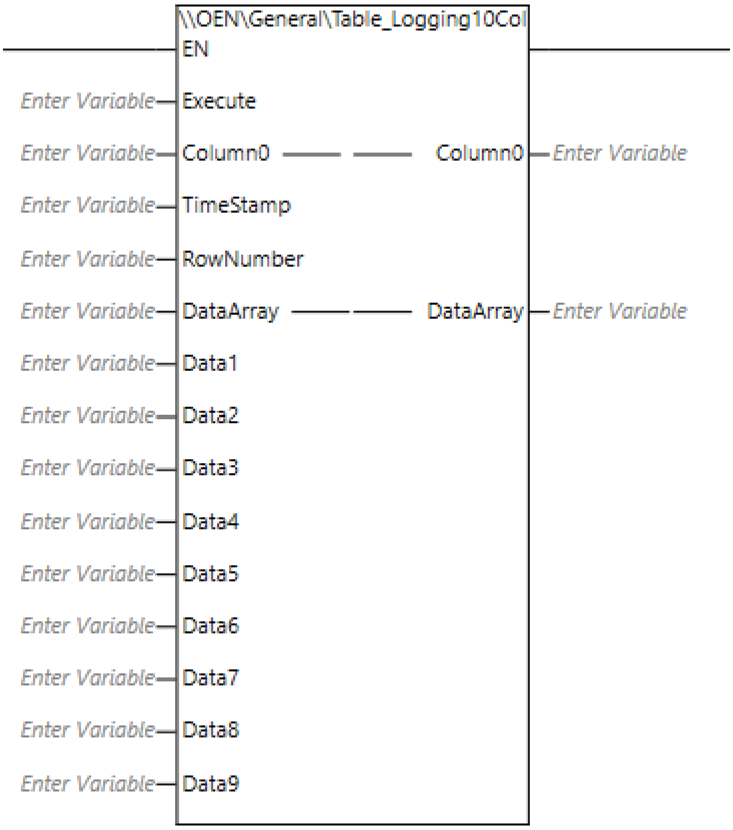


- Do not use more than one input as the result are output in 4 different ways. Example is not 100% correct.

46. *Table_Logging10Col*

A function block that logs 10 columns into 10 rows. In combination with SDLgging10Col, you can save this as a csv file on a memory card in NJ/NX.

46.1. FN Layout



46.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
Execute	BOOL	Add Column0 and data1..Data9 to the DataArray
TimeStamp	_sDT	Time to add to the row when logging
RowNumber	INT	Chose rownumber in DataArray to store the data
Data1..Data9	INT	Data to be inserted into the DataArray

46.3. In-Out Variables

Name	Data type	Description
Column0	ARRAY[0..9] OF STRING[20]	String data to insert into first Column in each row
DataArray	ARRAY[1..9,0..9] OF INT	DataTable that contains data to be saved

46.4. Output Variables

Name	Data Type	Description
Return	BOOL	

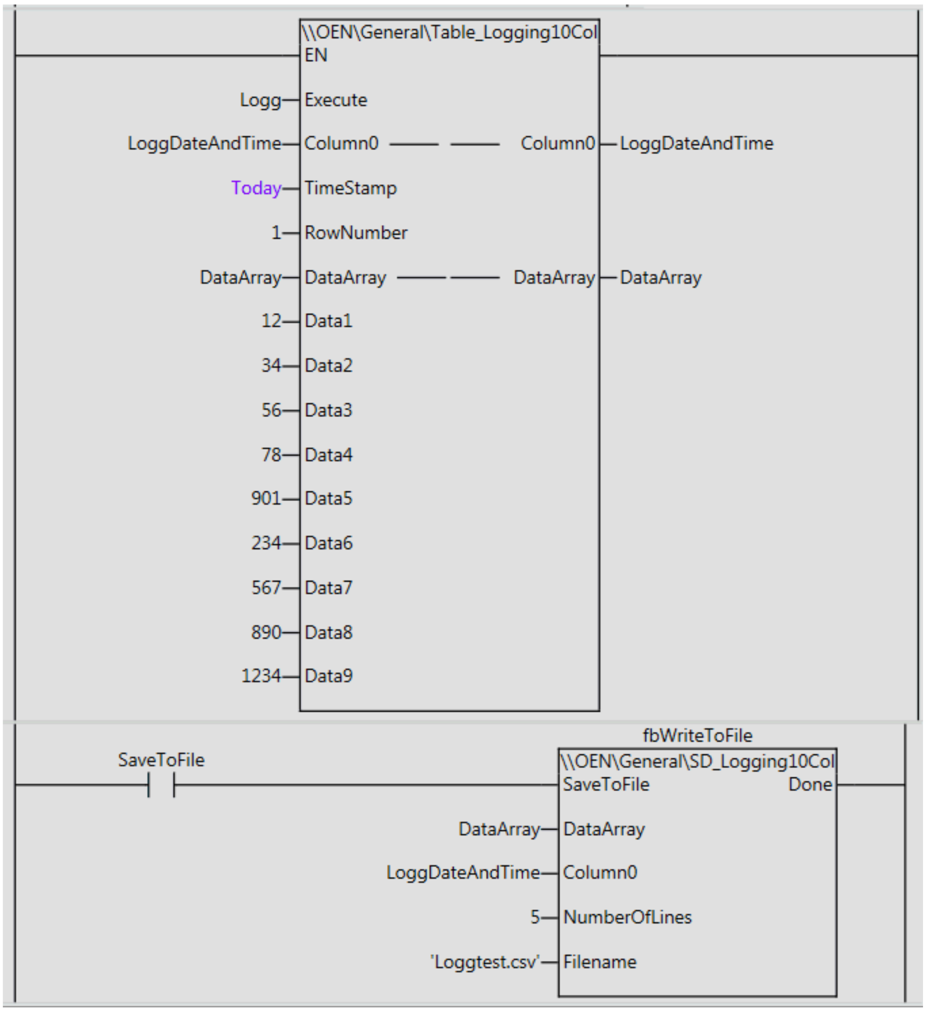
46.5. Revisions

Revision	In Library	Correction
1.1.0	5.00.9	

46.6. Credits

	Name
Omron - Norway	Kjell Baardsgaard

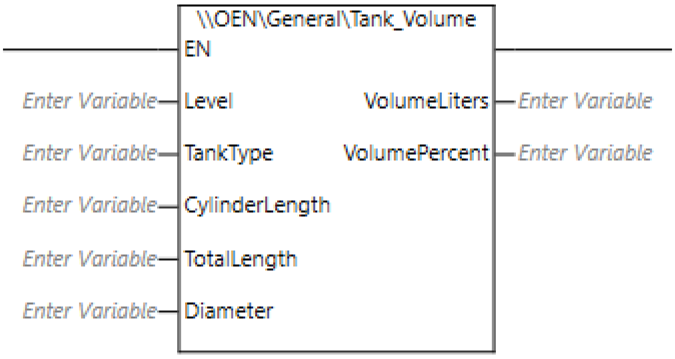
46.7. Example



47. *Tank_Volume*

Calculation of fluid volume in a vertical or horizontal tank based on level measurement.

47.1. FN Layout



47.2. Input Variables

Name	Data type	Valid Range	Description
EN	BOOL		Enable function
Level	REAL		Liquid level (mm)
TankType	OEN\eTank	Horizontal_FlatEnds Horizontal_EllipsoidalEnds Horizontal_SphericalEnds Vertical_FlatEnds Vertical_EllipsoidalEnds Vertical_SphericalEnds	
CylinderLength	REAL		Length of cylindrical part (mm)
TotalLength	REAL		Total tank length (mm)
Diameter	REAL		Cylinder diameter (mm)

47.3. Output Variables

Name	Data Type	Description
Return	BOOL	
VolumeLiters	REAL	Liquid volume (liters)
VolumePercent	REAL	Liquid volume in % of full tank

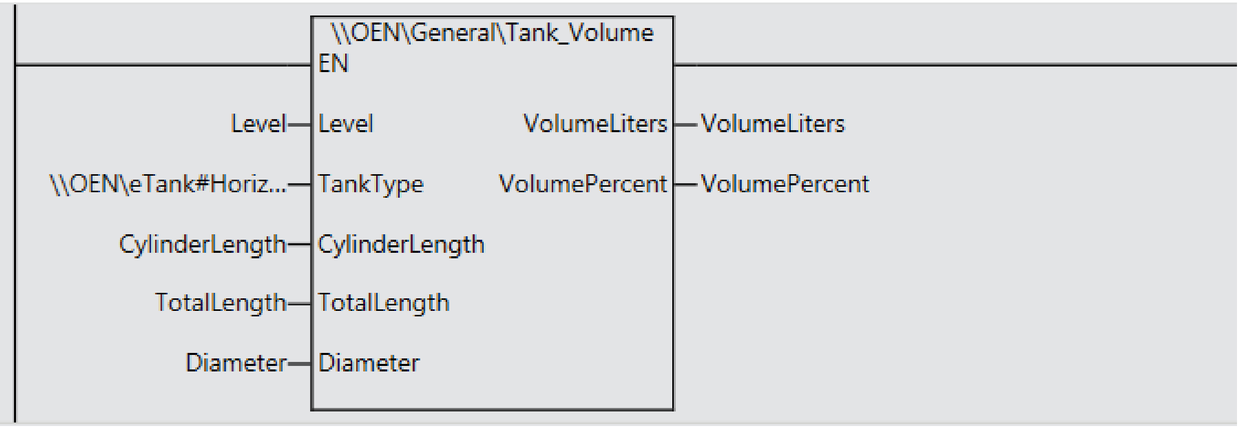
47.4. Revisions

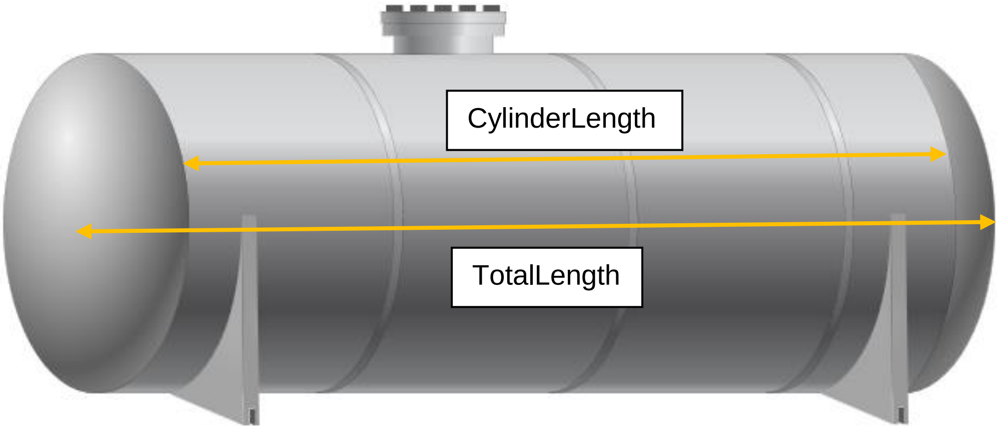
Revision	In Library	Correction
2.3.0	5.00.9	

47.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

47.6. Example


\\OEN\General\Tank_Volume
EN
Level
Level
VolumeLiters
VolumeLiters
\\OEN\eTank#Horiz...
TankType
VolumePercent
VolumePercent
CylinderLength
CylinderLength
TotalLength
TotalLength
Diameter
Diameter

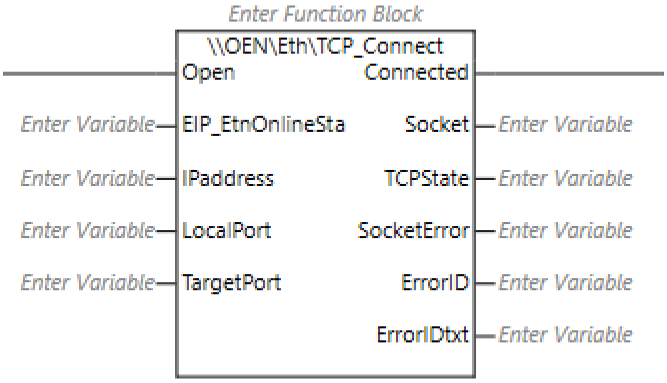

CylinderLength
TotalLength

48. *TCP_Connect*

A function block to simplify communication via SocketService. Open, Close, ClearBuf and GetTCPStatus are used in this, and the input Open can be used to open/close a Socket. E.g. ModbusTCP can then be run via the specified Socket. If you set LocalPort=0, it will find a free port automatically.

A socket is a kind of serial port. You open a socket and then you send and receive data on it. The CPU can have many sockets open on the same physical ethernet port, so this is much more rational than before, where one often had to have a serial port for each device to communicate with.

48.1. FB Layout



48.2. Input Variables

Name	Data type	Valid Range	Description
Open	BOOL		Enable function
EIP_EtnOnlineSta	BOOL		EIP port Status. Use System variable
IPAddress	STRING[200]		Example ‘192.168.250.3’
LocalPort	UINT		0=Auto Assignment
TargetPort	UINT		Remote TCP Target Port

48.3. Output Variables

Name	Data Type	Description
Connected	BOOL	Successfully connected
Socket	_sSocket	Socket status for monitoring purpose
TCPState	_eCONNECTION_STATE	Connection state as eNum
SocketError	BOOL	
ErrorID	WORD	Error Code
ErrorIDtxt	STRING[100]	ErrorID as readable message

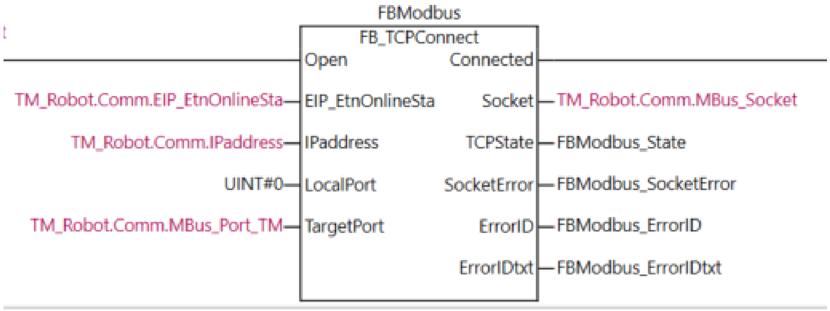
48.4. Revisions

Revision	In Library	Correction
2.1.0	5.00.9	
2.2.0	5.00.11	Counts 3 consecutive Port Errors before closing and reopening

48.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

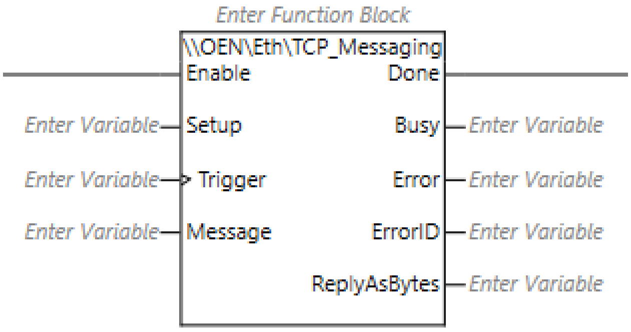
48.6. Example



49. *TCP_Messaging*

An FB to send custom data over TCP.

49.1. FB Layout



49.2. Input Variables

Name	Data type	Description
Enable	BOOL	Enable function
Setup	OEN\nEth\sSetup	Konfigurering av kommunikasjon mot TCP Server.
Trigger	BOOL	Trigger to send Message
Message	STRING[256]	Message to be send

49.3. Output Variables

Name	Data Type	Description
Done	BOOL	Message successfully send
Busy	BOOL	Connecting, Sending and waiting for reply
Error	BOOL	Error connecting or sending
ErrorID	WORD	Connecting or sending error code
ReplyAsBytes	ARRAY[0..255] OF BYTE	Reply from connected device

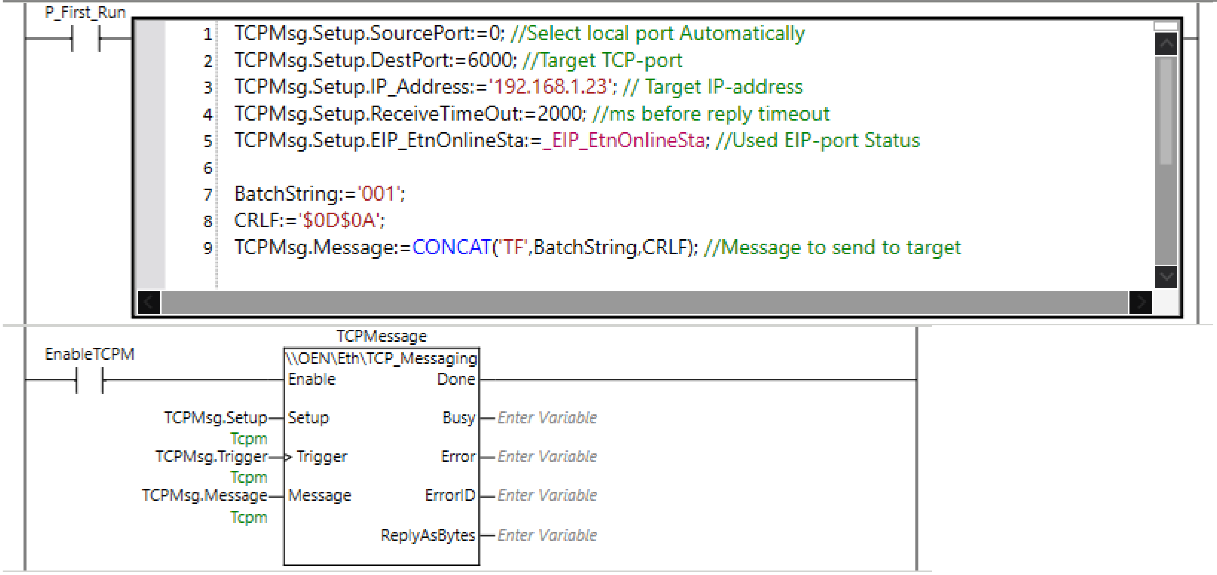
49.4. Revisions

Revision	In Library	Correction
1.1.0	5.00.9	

49.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

49.6. Example



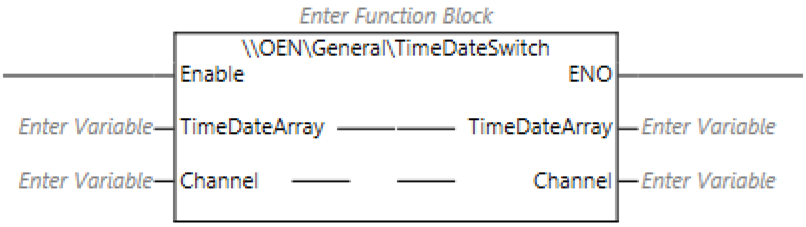
```
TCPMsg.Setup.SourcePort:=0; //Select local port Automatically
TCPMsg.Setup.DestPort:=6000; //Target TCP-port
TCPMsg.Setup.IP_Address:='192.168.1.23'; // Target IP-address
TCPMsg.Setup.ReceiveTimeOut:=2000; //ms before reply timeout
TCPMsg.Setup.EIP_EtnOnlineSta:=_EIP_EtnOnlineSta; //Used EIP-port Status

BatchString:='001';
CRLF:='$0D$0A';
TCPMsg.Message:=CONCAT('TF',BatchString,CRLF); //Message to send to target
```

50. *TimeDateSwitch*

A multichannel FB to start and stop events based on selected periods. Optional number of criteria and channels.

50.1. FB Layout



50.2. Input Variables

Name	Data type	Valid Range	Default	Description
Enable	BOOL		FALSE	Enable function

50.3. In-Out Variables

Name	Data type	Description
TimeDateArray	ARRAY[*] OF OEN\nGeneral\sTimeDateSwitch	Array containing start and stop times for Channel control
Channel	ARRAY[*] OF OEN\nGeneral\sTimeDateChannel	Channel with bools controlled by Start and Stop times. Several Start/Stop times can control the same channel.

50.4. Output Variables

Name	Data Type	Description
ENO	BOOL	Enable is TRUE

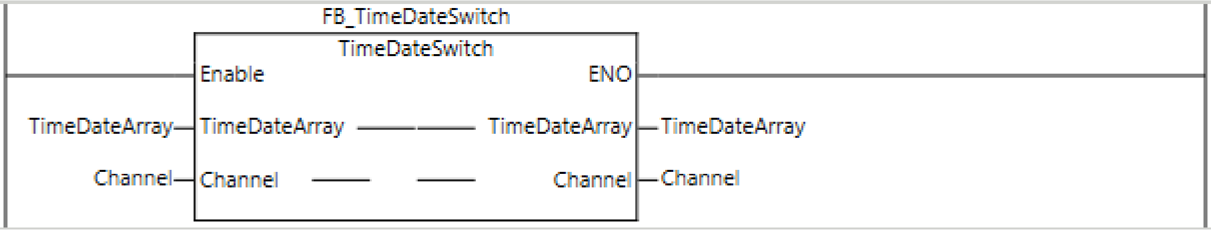
50.5. Revisions

Revision	In Library	Correction
1.0.0	5.00.9	

50.6. Credits

	Name
Omron - Norway	Kjell Baardsgaard

50.7. Example



In TimeDateArray, enter the time for Start and Stop, which channel to control, and whether the channel should be controlled on or off during this time interval. The number of rows in this Array you decide for yourself. Start with row 0. Channel is also an Array of optional size. You shouldn't use Channel 0, it can easily become a mess.
The section can be fixed day of the year, from one day to another day, between two times, on selected weekdays etc.
The program is based on Start time from inclusive until End time inclusive.

The following table explains this by four examples.
The first row in this table is Start Time. The second row is Stop Time.
[0]

Year	Month	Day	Hour	Min	Channel	Weekday	Channel Off	Description
0	5	17	0	0	1	0 0 0 0 0 0 0	TRUE	Channel 1 PowerOff=all day
0	5	18	0	0				

[1]

Year	Month	Day	Hour	Min	Channel	Weekday	Channel Off	Description
2020	7	1	0	0	2	0 0 0 0 0 0 0	TRUE	Channel 2 PowerOff=the whole holiday
2020	7	21	23	59				

[2]

Year	Month	Day	Hour	Min	Channel	Weekday	Channel Off	Description
0	0	0	08	0	3	1 1 1 1 1 0 0	False	Channel 3 PowerOn=on in working hours
0	0	0	16	0				

[3]

Year	Month	Day	Hour	Min	Channel	Weekday	Channel Off	Description
0	0	0	10	0	3	0 0 0 0 0 1 0	False	Channel 3 PowerOn=on Saturdays
0	0	0	14	0				

May 17th as in example [0] you can put as a fixed code in the PLC. It will then apply to all years.
TimeDateArray[0].Start.Month:=5;
TimeDateArray[0].Start.day:=17;
TimeDateArray[0].End.Mnd:=5;
TimeDateArray[0].End.Day:=18;
TimeDateArray[0].Channel:=1;
TimeDateArray[0].ChannelOff:=TRUE;
Instead of 18 on End.Day, you can also put 17 on End.Day, 23 at the End. Hour and 59 at End.Min. Then the output will be active 1 min shorter time.

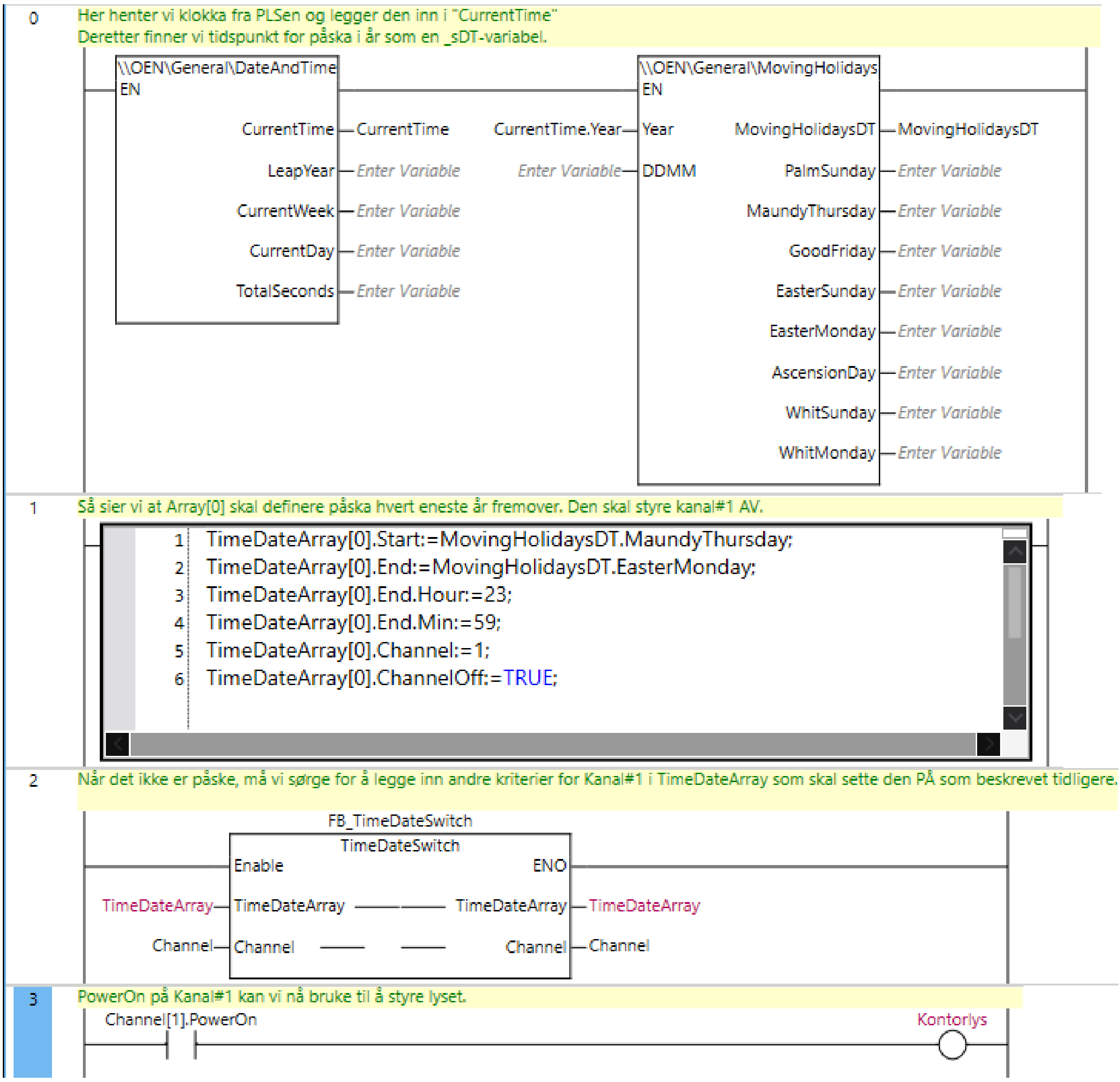
If we look at the holiday 2020 as another example, one would normally post your holiday via an HMI. The TimeDateArray variable must then be created as a GlobalVariable **Retain** and an Array of a certain size. The holiday is located here in example [1] above, which then corresponds to TimeDateArray[1].
Data Input fields must be created in HMI that connect to TimeDateArray[1].Start.Year, TimeDateArray[1]. End.Year etc. Which channel to control with this criterion is selected in TimeDateArray[1].Channel. For the holidays, we want to override all ON criteria and set TimeDateArray[1].ChannelOff=TRUE, preferably

with a click box in HMI. Weekdays are immaterial during the holidays, but a click box linked to TimeDateArray[2].Weekday[1] will correspond to Monday as shown in example [2].

In other words, each line in TimeDateArray controls channels. For example, these channels can be connected to light and/or heat in rooms.

Channel[3].PowerOn is a BOOL that can control the light in the offices during working hours. Note that example [2] and example [3] control the same channel, but with different times and days.

Here's how we can make sure the light doesn't go on during Easter for the next few years:



0 Her henter vi klokka fra PLSen og legger den inn i "CurrentTime"
Deretter finner vi tidspunkt for påska i år som en _sDT-variabel.

\\OEN\General\DateAndTime
EN
CurrentTime — CurrentTime
LeapYear — Enter Variable
CurrentWeek — Enter Variable
CurrentDay — Enter Variable
TotalSeconds — Enter Variable
\\OEN\General\MovingHolidays
EN
CurrentTime.Year — Year
Enter Variable — DDMM
MovingHolidaysDT — MovingHolidaysDT
PalmSunday — Enter Variable
MaundyThursday — Enter Variable
GoodFriday — Enter Variable
EasterSunday — Enter Variable
EasterMonday — Enter Variable
AscensionDay — Enter Variable
WhitSunday — Enter Variable
WhitMonday — Enter Variable

1 Så sier vi at Array[0] skal definere påska hvert eneste år fremover. Den skal styre kanal#1 AV.

```
TimeDateArray[0].Start:=MovingHolidaysDT.MaundyThursday;
TimeDateArray[0].End:=MovingHolidaysDT.EasterMonday;
TimeDateArray[0].End.Hour:=23;
TimeDateArray[0].End.Min:=59;
TimeDateArray[0].Channel:=1;
TimeDateArray[0].ChannelOff:=TRUE;
```

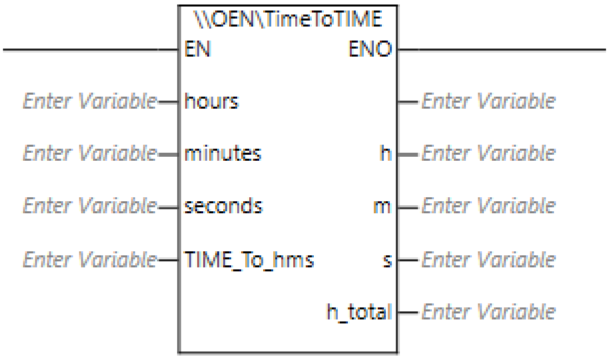
2 Når det ikke er påske, må vi sørge for å legge inn andre kriterier for Kanal#1 i TimeDateArray som skal sette den PÅ som beskrevet tidligere.

3 PowerOn på Kanal#1 kan vi nå bruke til å styre lyset.

51. *TimeToTIME*

Conversion of hours/minutes/seconds to Time or vice versa data type. Can be used to set the time of a Timer in the CPU from a field in an operator panel with data type UINT or REAL.

51.1. FN Layout



51.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
hours	UINT	Hours to be converted
minutes	UINT	Minutes to be converted
seconds	REAL	Seconds to be converted
TIME_To_hms	TIME	TIME to be extracted

51.3. Output Variables

Name	Data Type	Description
ENO	BOOL	EN is TRUE
Return	TIME	hours/minutes/seconds converted to TIME datatype
h	UINT	Hours extracted from TIME_To_hms
m	UINT	Minutes extracted from TIME_To_hms
s	REAL	Seconds extracted from TIME_To_hms
h_total	REAL	Time as a REAL value

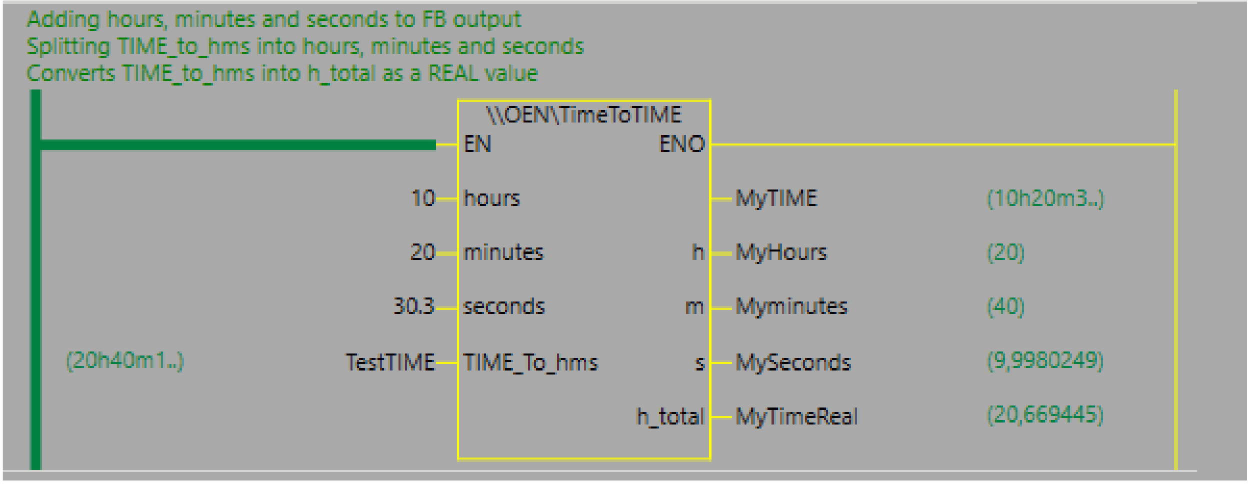
51.4. Revisions

Revision	In Library	Correction
2.1.0	5.00.9	

51.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

51.6. Example

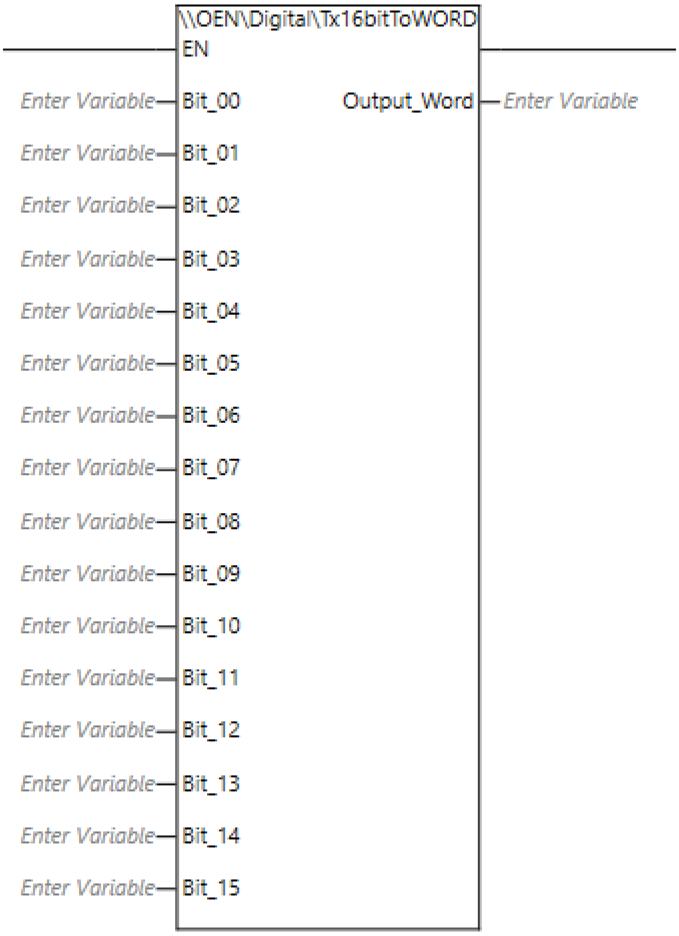


Adding hours, minutes and seconds to FB output
Splitting TIME_to_hms into hours, minutes and seconds
Converts TIME_to_hms into h_total as a REAL value

52. *Tx16bitToWORD*

Collect 16 bit into 1 Word

52.1. FN Layout



52.2. Input Variables

Name	Data type	Valid Range	Description
EN	BOOL		Enable function
Bit_00..Bit_15	BOOL		Bits to be collected into Output_Word

52.3. Output Variables

Name	Data Type	Description
Return	BOOL	
Output_Word	WORD	Collected bits

52.4. Revisions

Revision	In Library	Correction
1.0.1	5.00.9	

52.5. Credits

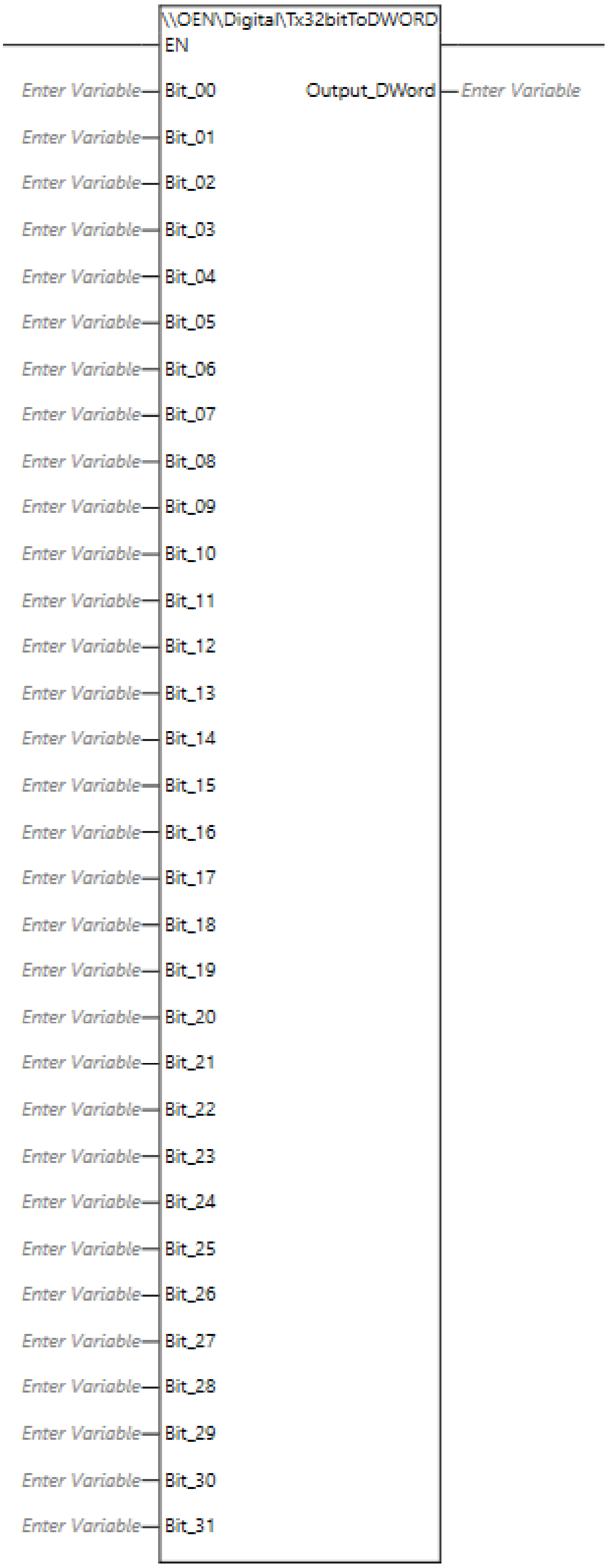
	Name
Omron - Norway	Kjell Baardsgaard

52.6. Example

53. *Tx32bitToDWORD*

Collect 32 bit into a Double Word

53.1. FN Layout



53.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
Bit_00..Bit_32	BOOL	Bits to be collected into Output_DWord

53.3. Output Variables

Name	Data Type	Description
Return	BOOL	
Output_DWord	DWORD	Collected bits

53.4. Revisions

Revision	In Library	Correction
1.0.1	5.00.9	

53.5. Credits

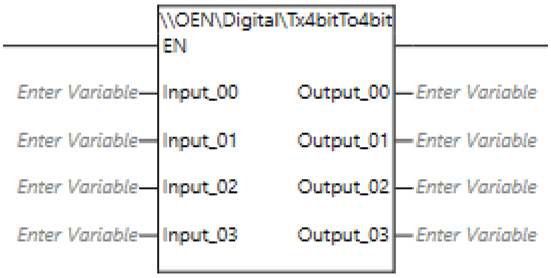
	Name
Omron - Norway	Kjell Baardsgaard

53.6. Example

54. *Tx4bitTo4bit*

Map 4 inputs to 4 outputs. Useful when connecting I/O data to/from structured variables.

54.1. FN Layout



54.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
Input_00	BOOL	Outputs status to Output_00
Input_01	BOOL	Outputs status to Output_01
Input_02	BOOL	Outputs status to Output_02
Input_03	BOOL	Outputs status to Output_03

54.3. Output Variables

Name	Data Type	Description
Return	BOOL	
Output_00..Output_03	BOOL	Input status

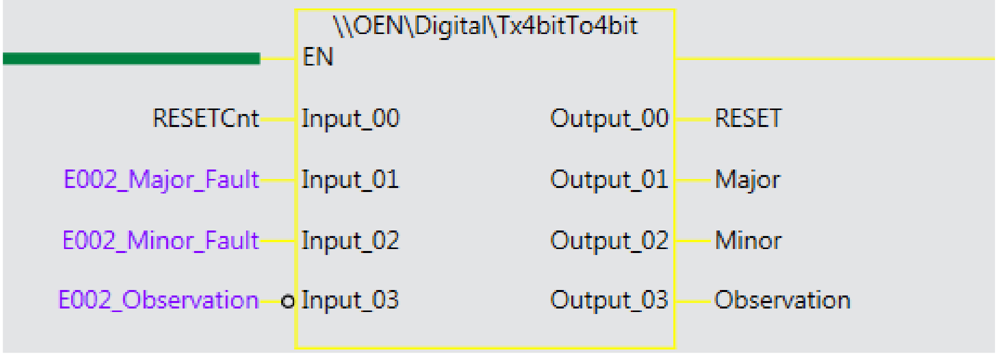
54.1. Revisions

Revision	In Library	Correction
1.0.1	5.00.9	

54.2. Credits

	Name
Omron - Norway	Kjell Baardsgaard

54.3. Example

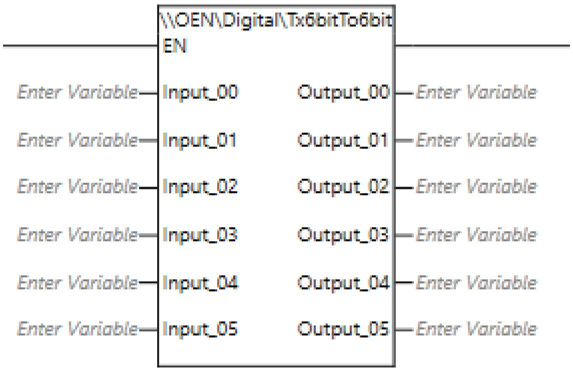


See above how you can Invert the signal on Input_03. Use RightClick.

55. *Tx6bitTo6bit*

Map 6 inputs to 6 outputs. Useful when connecting I/O data to/from structured variables.

55.1. FN Layout



55.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
Input_00	BOOL	Outputs status to Output_00
Input_01	BOOL	Outputs status to Output_01
Input_02	BOOL	Outputs status to Output_02
Input_03	BOOL	Outputs status to Output_03
Input_04	BOOL	Outputs status to Output_04
Input_05	BOOL	Outputs status to Output_05

55.3. Output Variables

Name	Data Type	Description
Return	BOOL	
Output_00..Output_05	BOOL	Input status

55.4. Revisions

Revision	In Library	Correction
1.0.1	5.00.9	

55.5. Credits

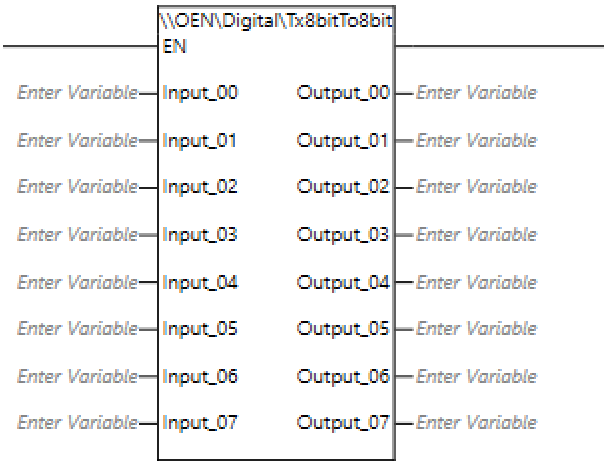
	Name
Omron - Norway	Kjell Baardsgaard

55.6. Example

56. *Tx8bitTo8bit*

Map 8 inputs to 8 outputs. Useful when connecting I/O data to/from structured variables.

56.1. FN Layout



56.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
Input_00	BOOL	Outputs status to Output_00
Input_01	BOOL	Outputs status to Output_01
Input_02	BOOL	Outputs status to Output_02
Input_03	BOOL	Outputs status to Output_03
Input_04	BOOL	Outputs status to Output_04
Input_05	BOOL	Outputs status to Output_05
Input_06	BOOL	Outputs status to Output_06
Input_07	BOOL	Outputs status to Output_07

56.3. Output Variables

Name	Data Type	Description
Return	BOOL	
Output_00..Output_07	BOOL	Input status

56.4. Revisions

Revision	In Library	Correction
1.0.1	5.00.9	

56.5. Credits

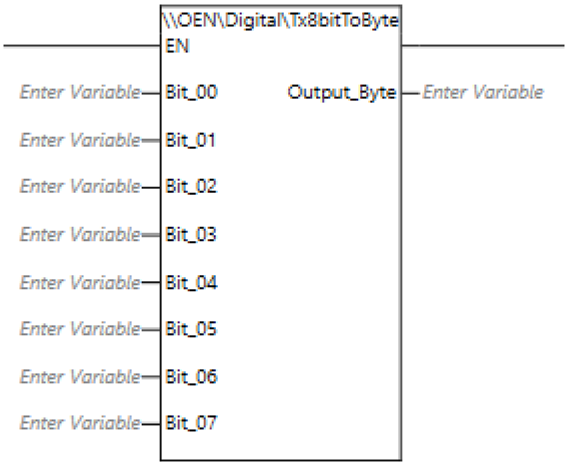
	Name
Omron - Norway	Kjell Baardsgaard

56.6. Example

57. *Tx8bitToByte*

Collect 8 bit into a Byte

57.1. FN Layout



57.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
Input_00	BOOL	Outputs status to bit0 in Byte
Input_01	BOOL	Outputs status to bit1 in Byte
Input_02	BOOL	Outputs status to bit2 in Byte
Input_03	BOOL	Outputs status to bit3 in Byte
Input_04	BOOL	Outputs status to bit4 in Byte
Input_05	BOOL	Outputs status to bit5 in Byte
Input_06	BOOL	Outputs status to bit6 in Byte
Input_07	BOOL	Outputs status to bit7 in Byte

57.3. Output Variables

Name	Data Type	Description
Return	BOOL	
Output_Byte	BOOL	Input status as a byte

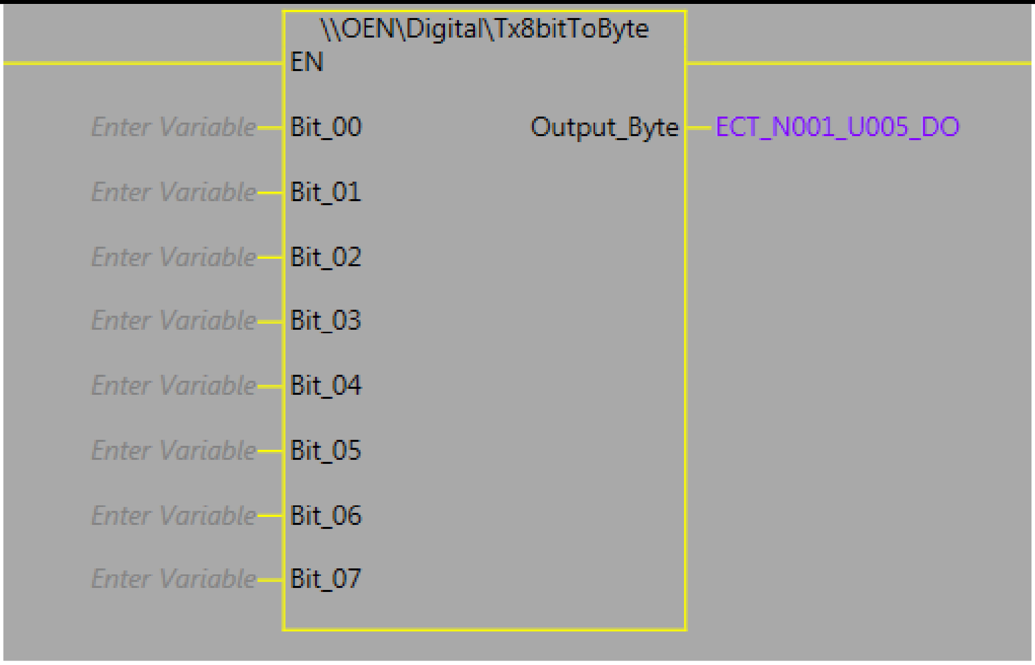
57.4. Revisions

Revision	In Library	Correction
1.0.1	5.00.9	

57.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

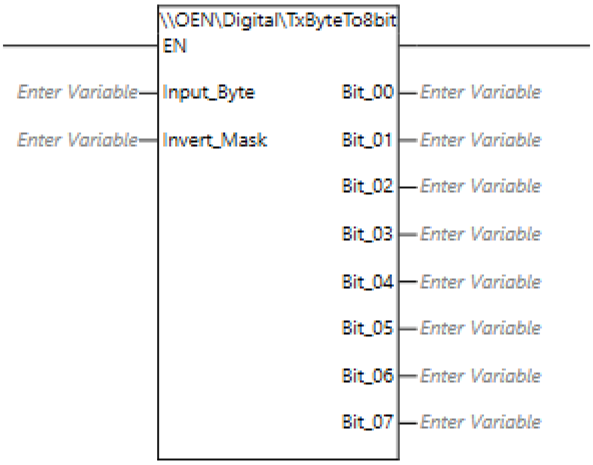
57.6. Example


\\OEN\Digital\Tx8bitToByte
EN
Enter Variable
Bit_00
Output_Byte
ECT_N001_U005_DO
Enter Variable
Bit_01
Enter Variable
Bit_02
Enter Variable
Bit_03
Enter Variable
Bit_04
Enter Variable
Bit_05
Enter Variable
Bit_06
Enter Variable
Bit_07

58. *TxByteTo8bit*

Distribute a byte into 8 bits. If using invert Mask, some or all bits can be inverted.

58.1. FN Layout



58.2. Input Variables

Name	Data type	Default	Description
EN	BOOL	FALSE	Enable function
Input_Byte	BYTE		Byte to distribute into 8 bits.
Invert_Mask	BYTE		Bits set TRUE in Byte Mask inverts corresponding output bit.

58.3. Output Variables

Name	Data Type	Description
Return	BOOL	
Bit_00..Bit_07	BOOL	Distributed bits from Input_Byte

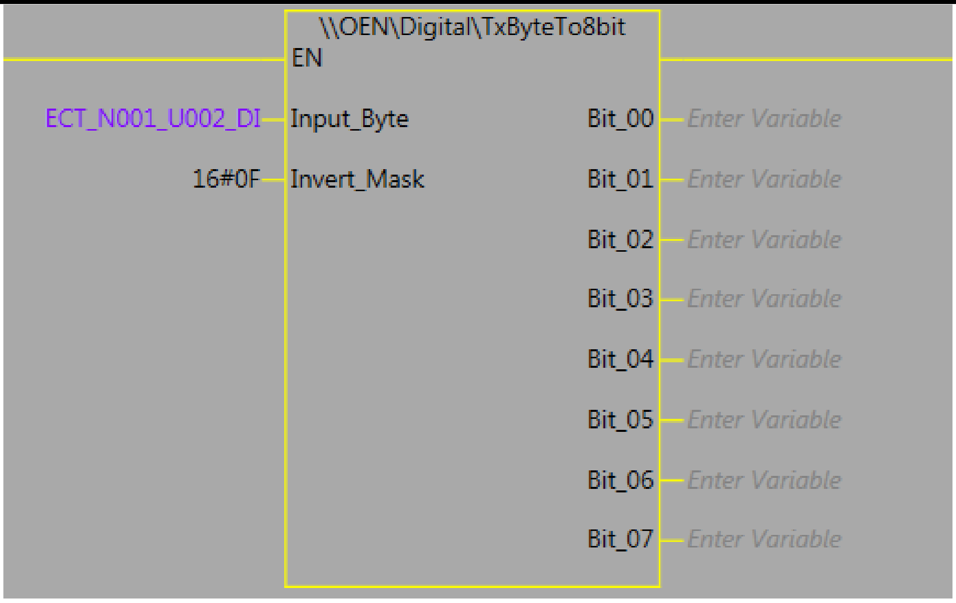
58.4. Revisions

Revision	In Library	Correction
2.0.1	5.00.9	

58.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

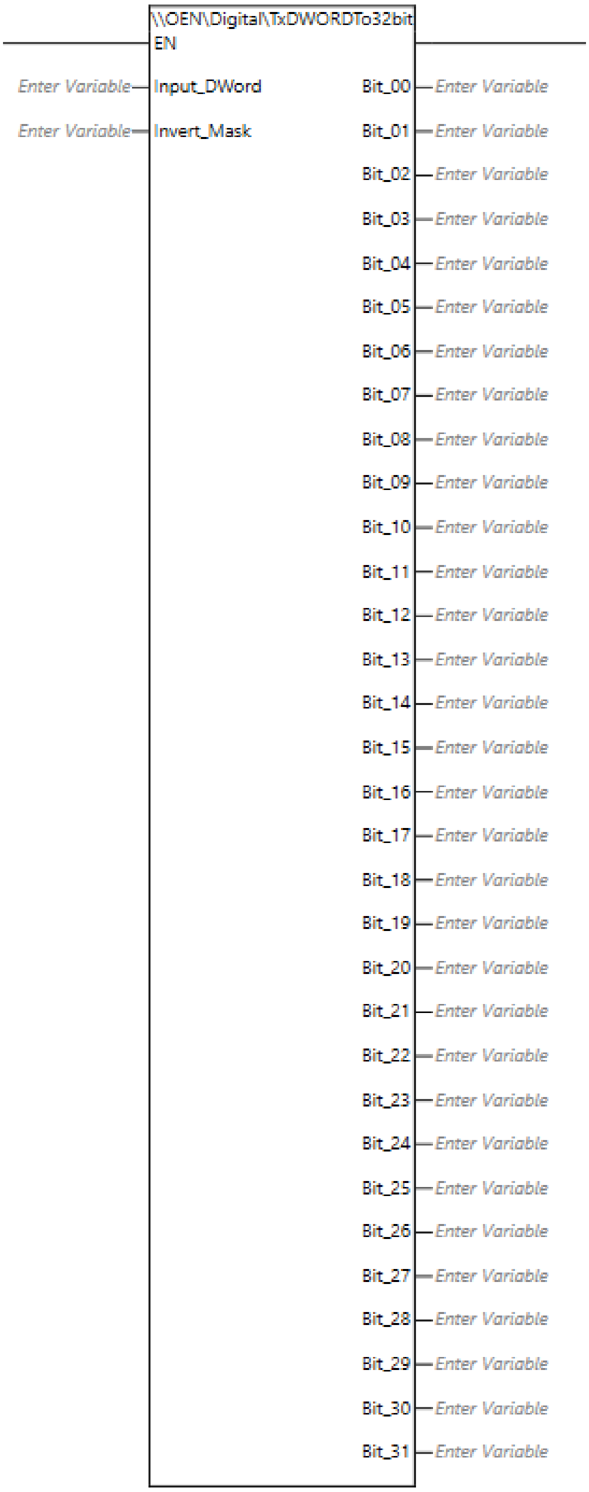
58.6. Example


\\OEN\Digital\TxByteTo8bit
EN
ECT_N001_U002_DI
Input_Byte
16#0F
Invert_Mask
Bit_00
Bit_01
Bit_02
Bit_03
Bit_04
Bit_05
Bit_06
Bit_07
Enter Variable
Enter Variable
Enter Variable
Enter Variable
Enter Variable
Enter Variable
Enter Variable
Enter Variable

59. *TxDWORDTo32bit*

Distribute a Dword into 32 bits. If using invert Mask, some or all bits can be inverted.

59.1. FN Layout



59.2. Input Variables

Name	Data type	Default	Description
EN	BOOL	FALSE	Enable function
Input_DWord	DWORD		DWord to distribute into 32 bit
Invert_Mask	DWORD		Mask to invert bit. Bits set TRUE in Mask inverts corresponding output bit.

59.3. Output Variables

Name	Data Type	Description
Return	BOOL	
Bit_00..Bit_31	BOOL	Distributed bits from Input_DWord

59.4. Revisions

Revision	In Library	Correction
2.0.1	5.00.9	

59.5. Credits

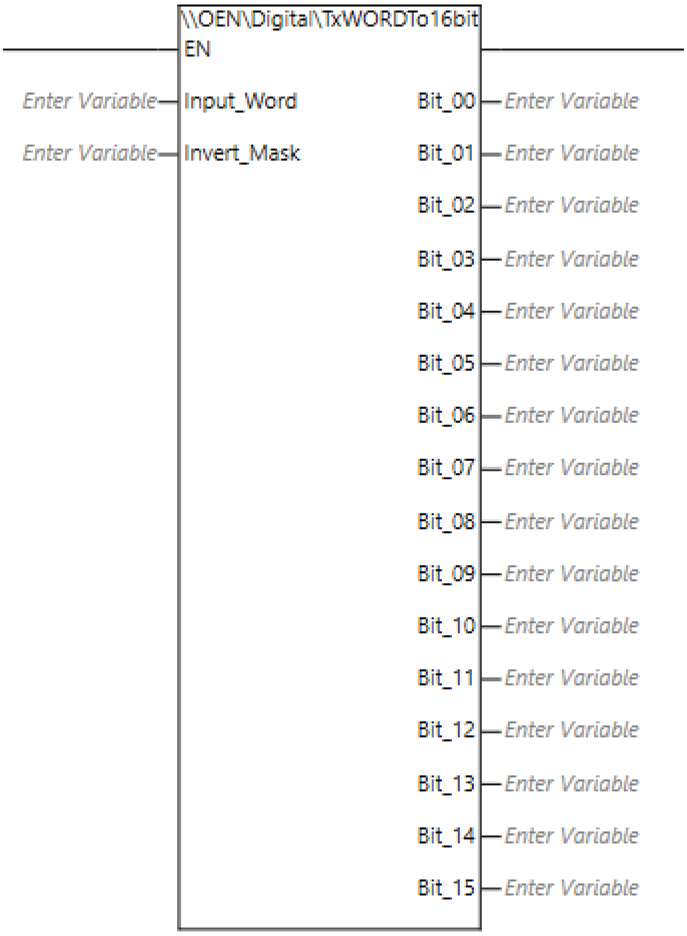
	Name
Omron - Norway	Kjell Baardsgaard

59.6. Example

60. *TxWORDTo16bit*

Distribute a Word into 16 bits. If using invert Mask, some or all bits can be inverted.

60.1. FN Layout



60.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
Input_Word	WORD	Word to distribute into 16 bit
Invert_Mask	WORD	Mask to invert bit. Bits set TRUE in Mask inverts corresponding output bit.

60.3. Output Variables

Name	Data Type	Description
Bit_00..Bit_15	BOOL	Distributed bits from Input_Word

60.4. Revisions

Revision	In Library	Correction
2.0.1	5.00.9	

60.5. Credits

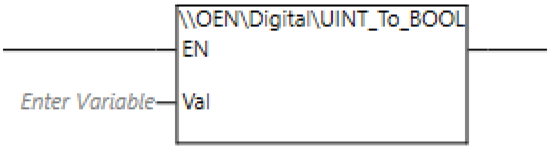
	Name
Omron - Norway	Kjell Baardsgaard

60.6. Example

61. *UINT_To_BOOL*

Conversion of the integer value 1 or 0 to a bit.

61.1. FN Layout



61.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
Val	UINT	Value to set digital output

61.3. Output Variables

Name	Data Type	Description
Return	BOOL	FALSE if Val=0 else TRUE

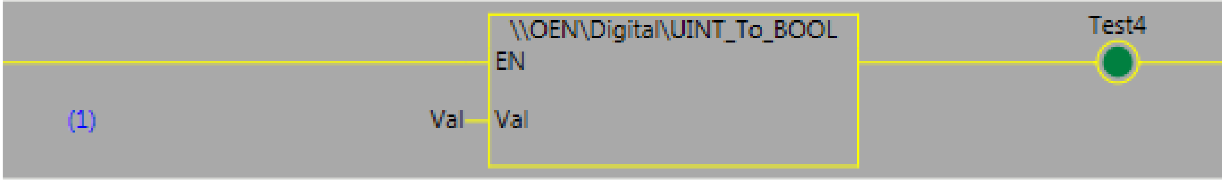
61.4. Revisions

Revision	In Library	Correction
1.0.0	5.00.9	

61.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

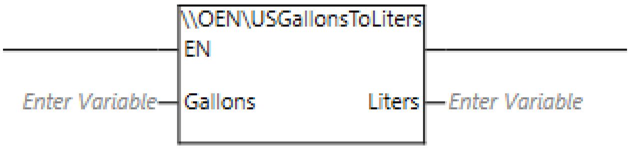
61.6. Example



62. *USGallonsToLiters*

Conversion of USGallons to Liters.

62.1. FN Layout



62.2. Input Variables

Name	Data type	Valid Range	Default	Description
EN	BOOL		FALSE	Enable function
Gallons	REAL			

62.3. Output Variables

Name	Data Type	Description
Return	BOOL	
Liters	REAL	

62.4. Revisions

Revision	In Library	Correction
1.1.0	5.00.9	

62.5. Credits

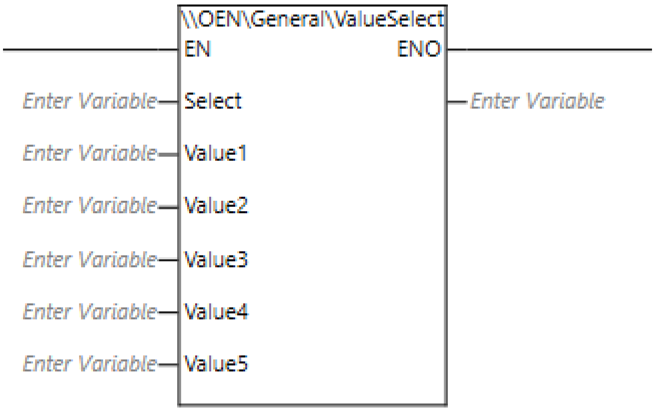
	Name
Omron - Norway	Kjell Baardsgaard

62.6. Example

63. *ValueSelect*

A function where you can select one of 5 values.
Select can be a number between 1 and 5, and the output will be set equal to the selected input.

63.1. FN Layout



63.2. Input Variables

Name	Data type	Valid Range	Default	Description
EN	BOOL		FALSE	Enable function
Select	UINT	0..5	0	
Value1..Value5	REAL			Value to choose from with Select

63.3. Output Variables

Name	Data Type	Description
ENO	BOOL	TRUE if EN=TRUE
Return	REAL	Selected value

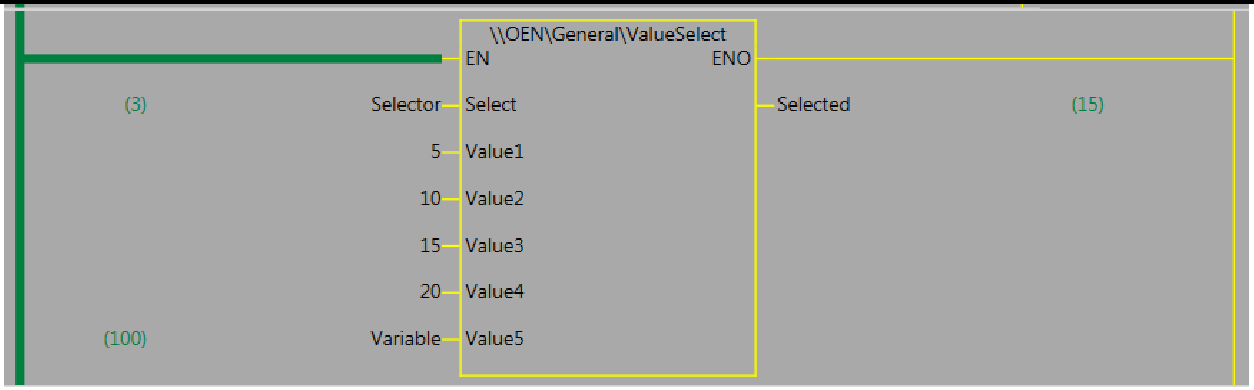
63.4. Revisions

Revision	In Library	Correction
1.1.0	5.00.9	

63.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

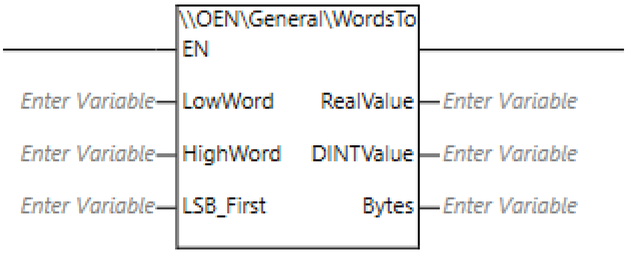
63.6. Example


\\OEN\General\ValueSelect
EN
ENO
(3)
Selector
Select
Selected
(15)
5
Value1
10
Value2
15
Value3
20
Value4
(100)
Variable
Value5

64. *WordsTo*

Merge two WORD variables into a REAL, DINT, or BYTE array. The order of bytes in the BYTE array is determined by LSB_First (TRUE/FALSE).

64.1. FN Layout



64.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
LowWord	WORD	Low Word to Merge
HighWord	WORD	HighWord to Merge
LSB_First	BOOL	Merger Order

64.3. Output Variables

Name	Data Type	Description
Return	BOOL	
RealValue	REAL	2 words as REAL
DINTValue	DINT	2 words as DINT
Bytes	ARRAY[0..3] OF BYTE	2 words as bytes

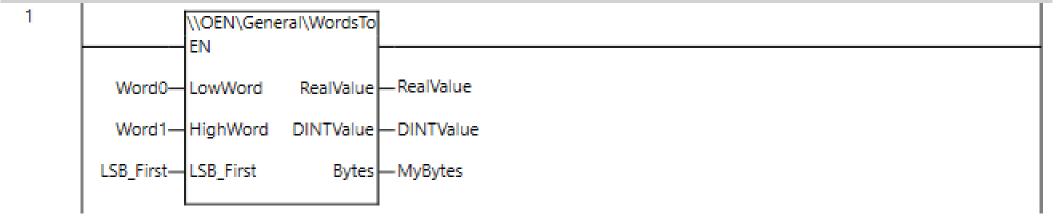
64.4. Revisions

Revision	In Library	Correction
2.1.0	5.00.9	

64.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

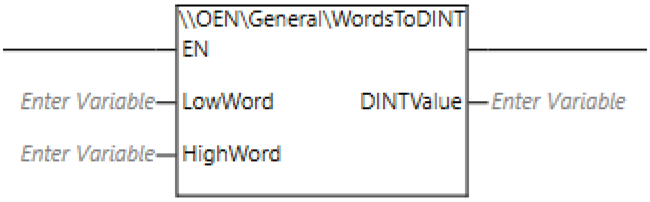
64.6. Example


1
\\OEN\General\WordsTo
EN
Word0
LowWord
RealValue
RealValue
Word1
HighWord
DINTValue
DINTValue
LSB_First
LSB_First
Bytes
MyBytes

65. *WordsToDINT*

Merge two WORD variables into a DINT.

65.1. FN Layout



65.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
LowWord	WORD	Low word to Merge
HighWord	WORD	High word to Merge

65.3. Output Variables

Name	Data Type	Description
Return	BOOL	
DINTValue	DINT	2 words as DINT

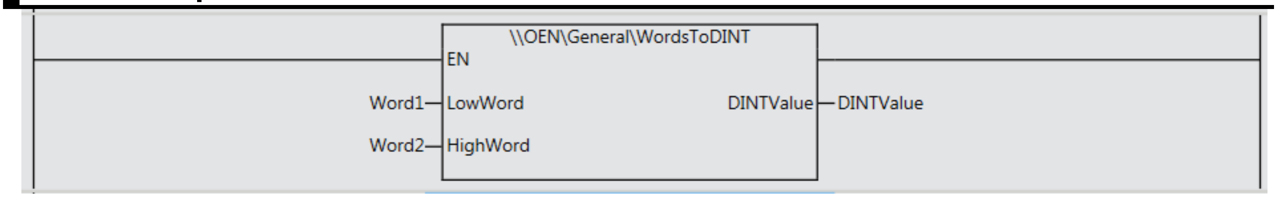
65.4. Revisions

Revision	In Library	Correction
1.1.1	5.00.9	

65.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

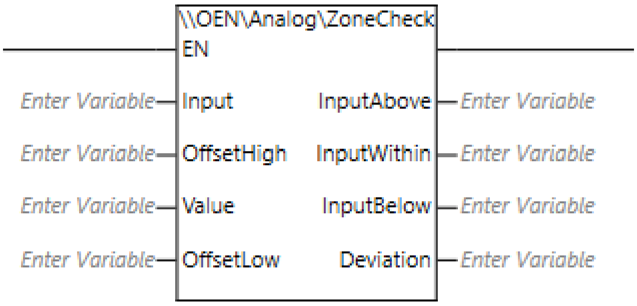
65.6. Example



66. *ZoneCheck*

A function that checks whether an analog value is within the Value+OffsetHigh and Value-OffsetLow interval. Deviation shows the discrepancy between Input and Value.

66.1. FN Layout



66.2. Input Variables

Name	Data type	Description
EN	BOOL	Enable function
Input	REAL	Value to monitor
OffsetHigh	REAL	Offset above Value
Value	REAL	Value to compare with
OffsetLow	REAL	Offset below value

66.3. Output Variables

Name	Data Type	Description
Return	BOOL	
InputAbove	BOOL	Input is above (Value+OffsetHight)
InputWithin	BOOL	Input is not above or below limits
InputBelow	BOOL	Input is below (Value-OffsetLow)
Deviation	REAL	Input-Value

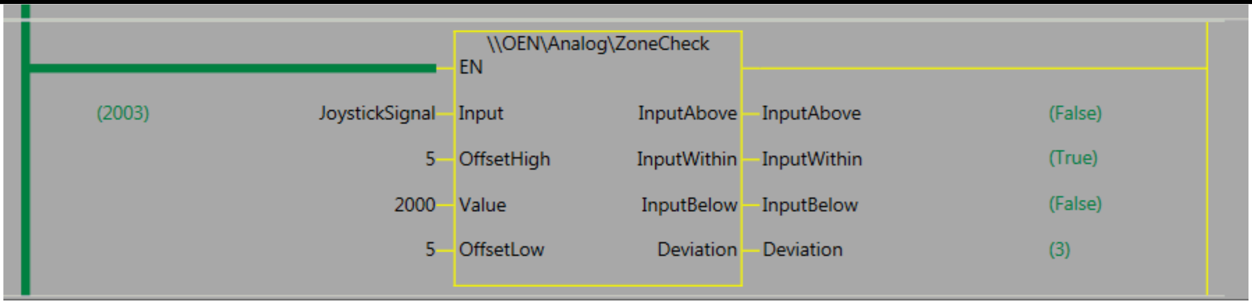
66.4. Revisions

Revision	In Library	Correction
1.1.0	5.00.9	

66.5. Credits

	Name
Omron - Norway	Kjell Baardsgaard

66.6. Example


\\OEN\Analog\ZoneCheck
EN
(2003)
JoystickSignal
Input
5
OffsetHigh
2000
Value
5
OffsetLow
InputAbove
InputWithin
InputBelow
Deviation
InputAbove
InputWithin
InputBelow
Deviation
(False)
(True)
(False)
(3)

67. *Template*

text

67.1. FN Layout

67.2. Input Variables

Name	Data type	Valid Range	Default	Description
EN	BOOL		FALSE	Enable function

67.3. In-Out Variables

Name	Data type	Description

67.4. Output Variables

Name	Data Type	Description
	BOOL	

67.5. Revisions

Revision	In Library	Correction
1.0.0	5.00.9	

67.6. Credits

	Name
Omron - Norway	Kjell Baardsgaard

67.7. Example