User Manual

Function Block for eNod4-T EIP

For EcoStruxure Machine Expert, Schneider Electric

257707-A_NU-eNod4-T-FtBk_EIP_MachineExpert_E.docx
SCAIME SAS – Technosite Altéa – 294, Rue Georges Charpak – 74100 JUVIGNY - FRANCE
Tél. : +33 (0)4 50 87 78 64 – www.scaime.com

1 GENERAL PRESENTATION 3
2 INSTALLING EDS FILE IN MACHINE EXPERT 4
3 INSTALLING ENOD LIBRARY IN MACHINE EXPERT 6
4 ADDING ENOD4T IN THE PROJECT 7
5 USING ADD-ON INSTRUCTIONS IN THE PROJECT 9
5.1 Main bloc “TeNod4TFBMain” 9
5.1.1 Creating the instance 9
5.1.2 Select global data for the eNod. 10
5.1.3 Select implicit data from hardware configuration 10
5.1.4 Enter the Ethernet/IP address of the eNod 10
5.1.5 Using instructions 10
5.1.6 Compatibility with different Machine Expert versions 13
5.2 “TeNod4TFBBackupRestore” to restore and backup configuration 13
5.2.1 Creating the instance 13
5.2.2 Select global data for the eNod. 14
5.2.3 Using the instruction 14
5.3 “TeNod4TFBCalibration” to calibrate a scale 15
5.3.1 Creating the instance 15
5.3.2 Select global data for the eNod. 15
5.3.3 Using the instruction 15
5.4 “TeNod4TFBFilters” to read and write filter configuration 20
5.4.1 Creating the instance 20
5.4.1 Select global data for the eNod 20
5.4.2 Using the instruction 20
5.5 “TeNod4TFBLegal” to manage DSD legal values 21
5.5.1 Creating the instance 21
5.5.2 Select global data for the eNod. 21
5.5.3 Using the instruction 21
6 COMMUNICATION ERROR CODES 23
7 BASIC COMMUNICATION 24

1 GENERAL PRESENTATION

eNod4 is a high-speed digital process transmitter with programmable functions and powerful signal processing capabilities. ***eNod4*** offers operating modes for advanced static and dynamic process control.

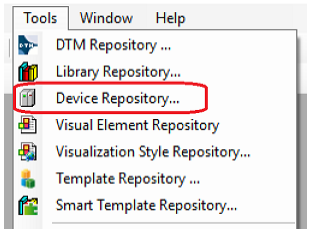
The eNod4-T Block function for EcoStruxure Machine Expert is an open-source code that eases the integration of an eNod4 with Schneider Electric M24x/25x PLC. This tool integrates pre-programmed functions for EcoStruxure Machine Expert. You can quickly get your measurement data, calibrate or restore your eNod4 using this code.

This user manual describes how to integrate the block function in EcoStruxure Machine Expert and communicate with an eNod4-T Ethernet/IP through a Schneider Electric M24x/25x PLC.

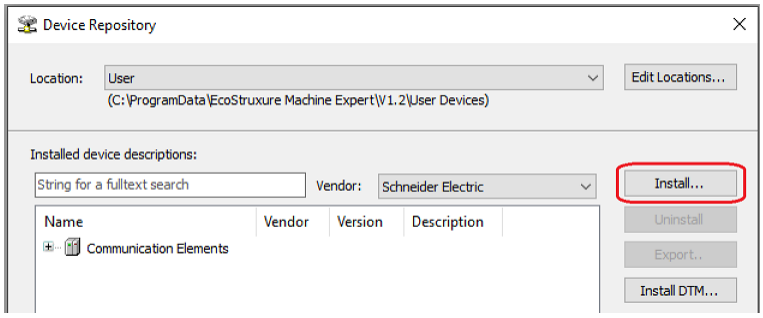
2 INSTALLING EDS FILE IN MACHINE EXPERT

Before using eNod4 in application, EDS file must be imported in Machine Expert to add the eNod4 module in hardware catalog.

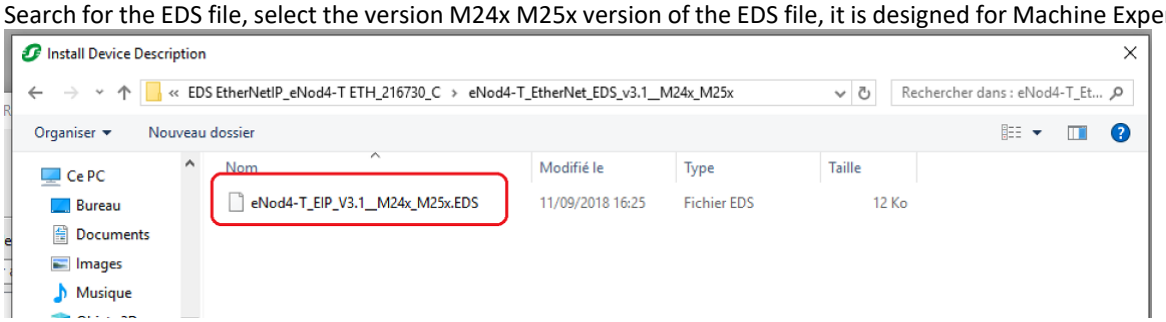
On the “Tools” menu go to “Devices repository…”



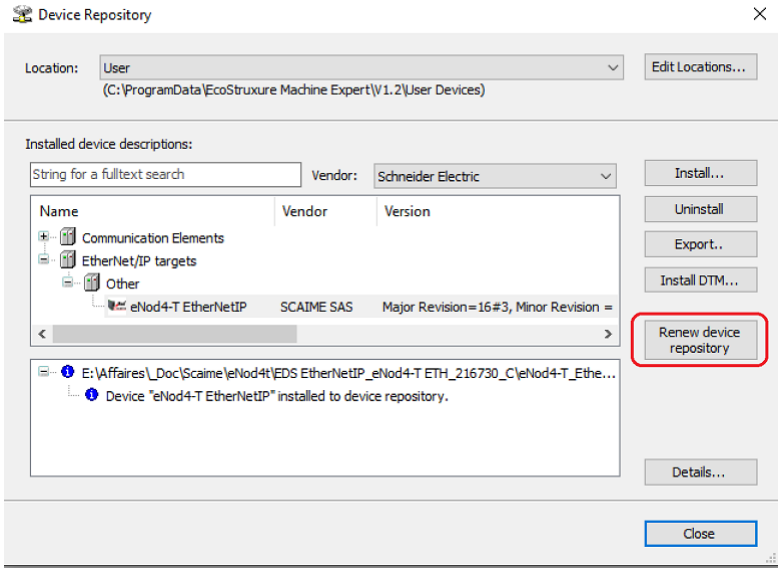
Click on the “Install…” button:



Search for the EDS file, select the version M24x M25x version of the EDS file, it is designed for Machine Expert program:

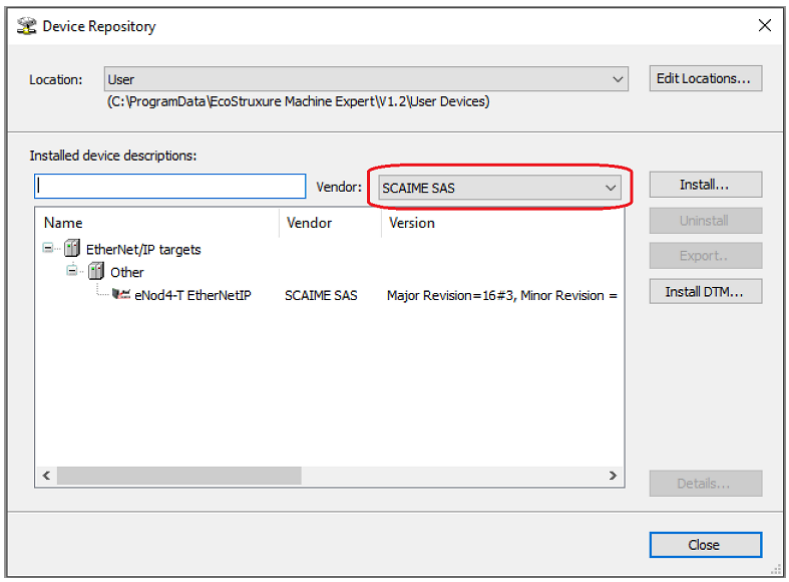


The eNod4 appears on the list, and then click on “Renew device repository”



Click on “Yes” on the confirmation message.

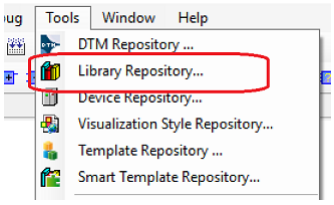
Now the eNod4T is ready to be used on the PLC projects. You can see it if you filter by vendor:


Device Repository
Location:
User
(C:\ProgramData\EcoStruxure Machine Expert\V1.2\User Devices)
Edit Locations...
Installed device descriptions:
Vendor:
SCAIME SAS
Install...
Uninstall
Export...
Install DTM...
Name
Vendor
Version
EtherNet/IP targets
Other
eNod4-T EtherNetIP
SCAIME SAS
Major Revision=16#3, Minor Revision =
Details...
Close

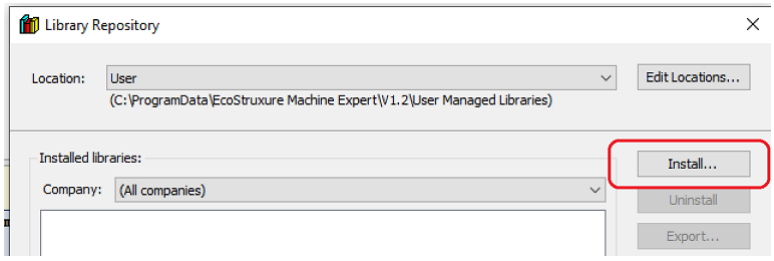
3 INSTALLING ENOD LIBRARY IN MACHINE EXPERT

To get eNod4t functions blocs in all projects, you need to install the library in Machine Expert

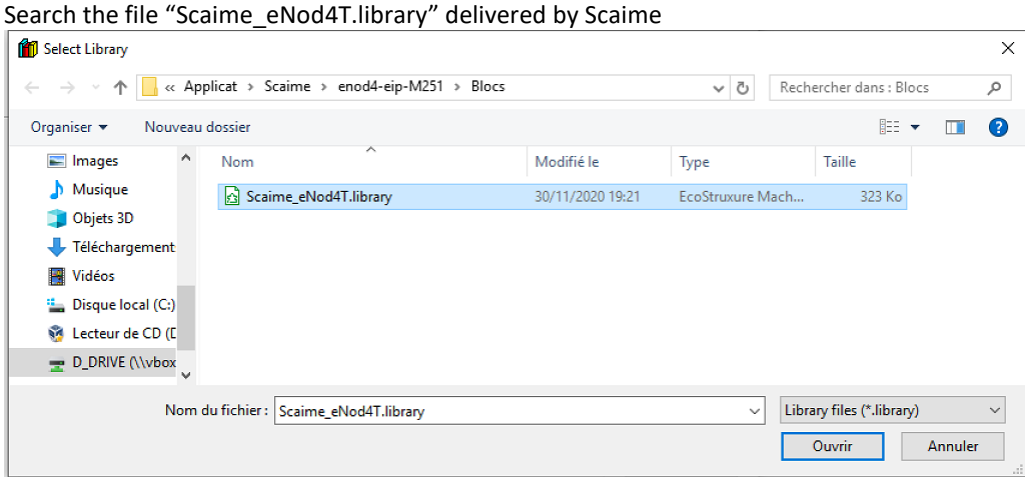
On “Tools” menu go to “Libraries repository”



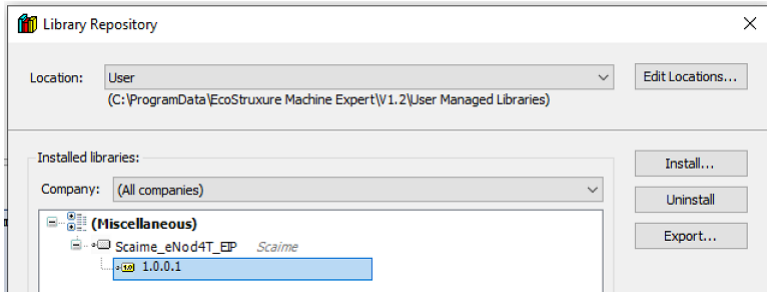
Click on “Install…” button.



Search the file “Scaime_eNod4T.library” delivered by Scaime

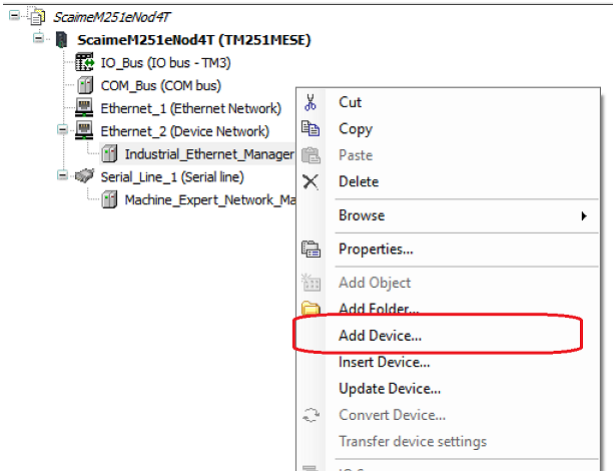


The library is added to the list, you can close the repository:

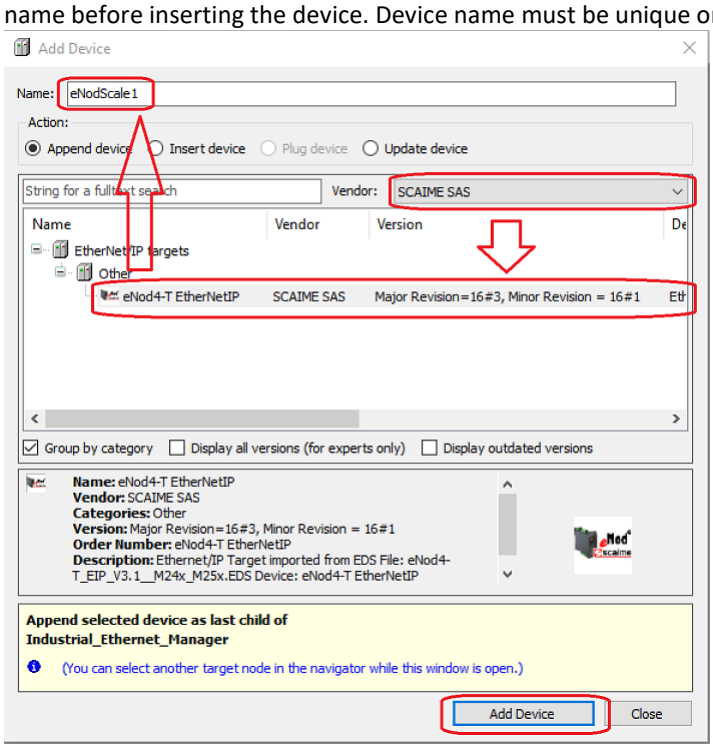


4 ADDING ENOD4T IN THE PROJECT

On the device tree, right click on the “Industrial_Ethernet_Manager” and select “Add device…”.



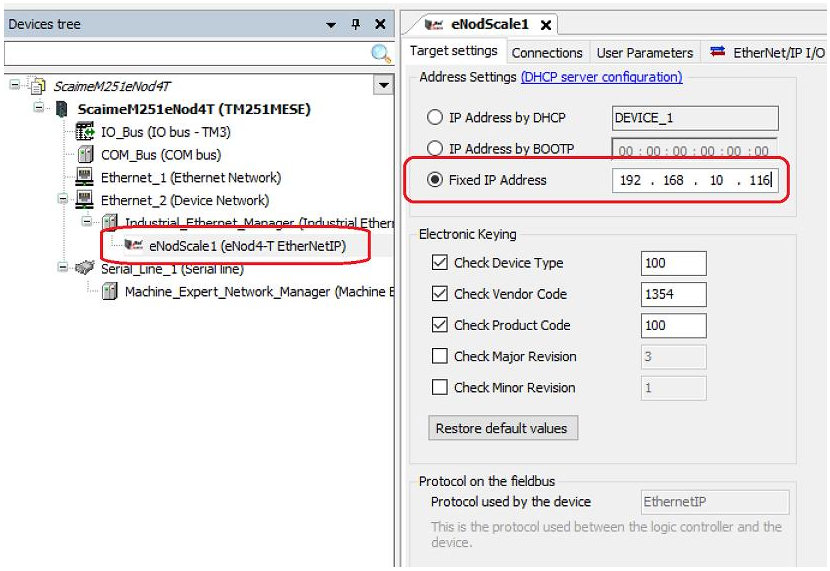
On the device selection window, select “Scaime SAS” as vendor name to easily find the eNod device on the list. Then, click once on it to select it. This set up by default a device name to “eNod4_T_EtherNetIP”, you need to change the name before inserting the device. Device name must be unique on the project. Let’s call it “eNodScale1”:



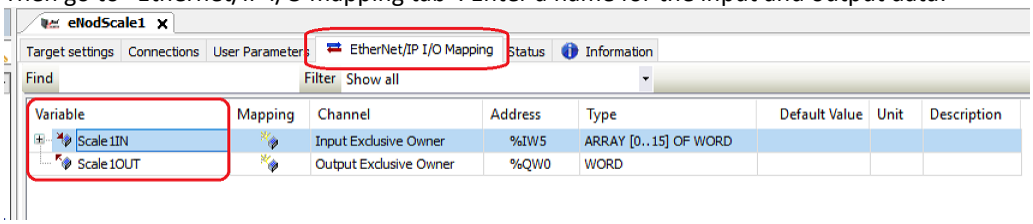
At the end click on “Add device” to add the eNod on the project.

The device windows stays open to allow you to insert easily a second eNod. Just “close” the window when the eNod(s) and added to the project.

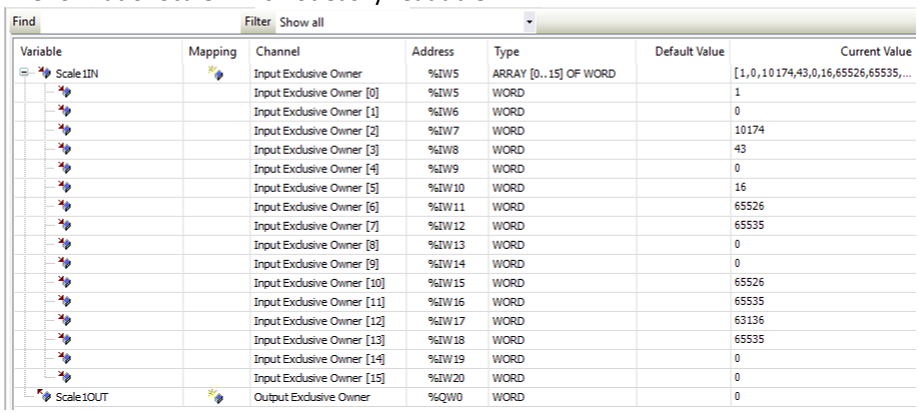
Machine Expert will the eNod in DHCP for the Ethernet IP address. To change the IP address of the eNod, double click on the eNod on the device tree. Then you can set the IP parameters, on this project we will you static IP:



Then go to “Ethernet/IP I/O mapping tab”. Enter a name for the input and output data:



Now you can generate and download the project. The two variables Scale1IN and Scale1OUT are implicitly exchanged. The format of Scale1IN is not easily readable:



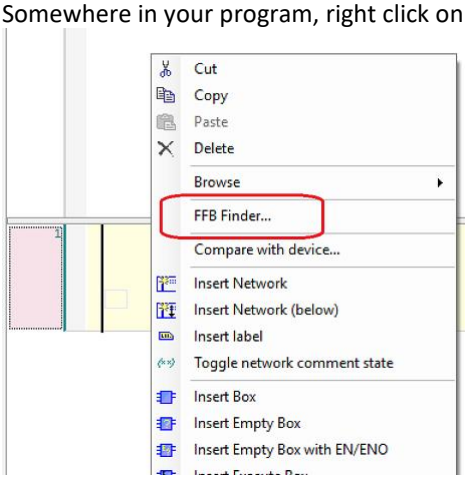
5 USING ADD-ON INSTRUCTIONS IN THE PROJECT

5.1 Main bloc "TeNod4TFBMain"

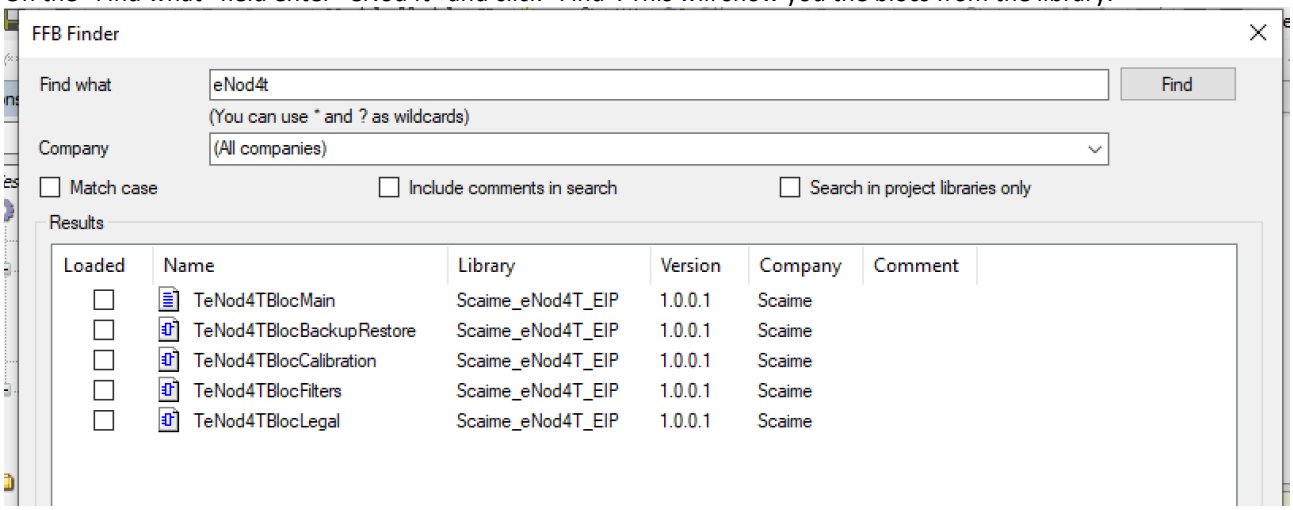
This bloc manages the communication with the eNod and basic commands. It is mandatory for the other blocs.

5.1.1 Creating the instance

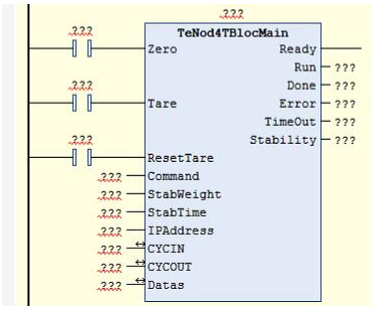
Somewhere in your program, right click on the grid and select "FFB finder…".



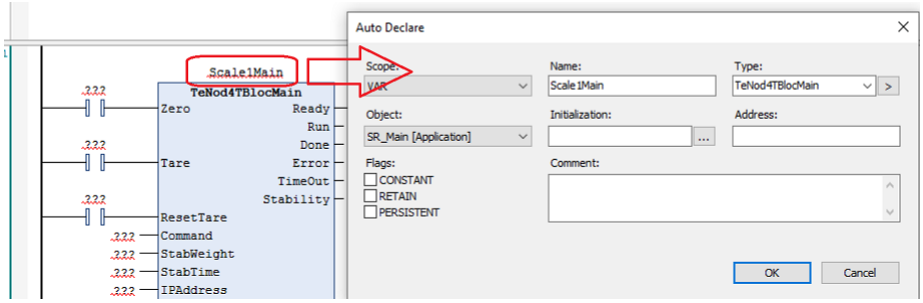
On the "Find what" field enter "eNod4t" and click "Find". This will show you the blocs from the library:



Double-click on the TeNod4TBlocMain to add it to the program.



Then enter an instance name for the bloc, the instance name must be different for every eNod of the application:



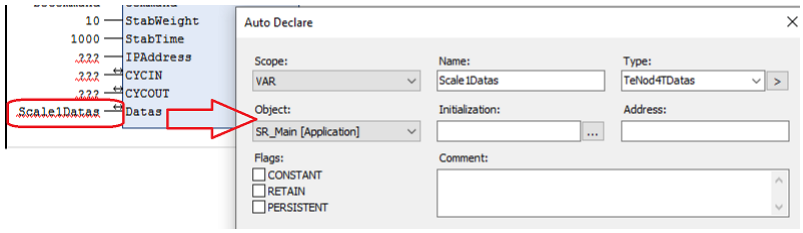
Machine Expert propose to create the new instance, click OK to confirm.

Note: Machine expert enters ??? for all fields, they are not al mandatory. If you do not use on field replace the value by “False” or 0 for an input field and leave the output fields blanks. There are four mandatory fields that are explain below. Even if a field is not filled on the bloc definition you can still use it elsewhere on the program using InstameName.FieldName, for example Scale1Main.Tare.

5.1.2 Select global data for the eNod.

This structure will receive all data from the eNod and will be a parameter for all instructions.
It must be unique for every scale on the project.

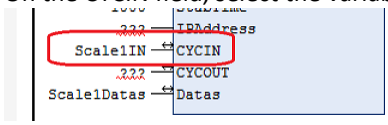
Add a “Scale1Datas” for the “Datas” field:



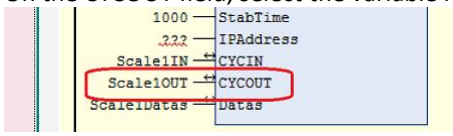
Click OK to automatically create the variable.

5.1.3 Select implicit data from hardware configuration

On the CYCIN field, select the variable name for the input data defined on §III.

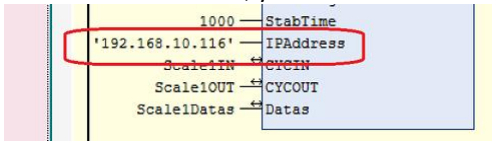


On the CYCOUT field, select the variable name for the input data defined on §III.



5.1.4 Enter the Ethernet/IP address of the eNod

On the IPAddress field, you must enter the IP address of the eNod. The format is a normal string value:



5.1.5 Using instructions

At this moment the configuration and basic program is done.
You can transfer the program to PLC to test communication

5.1.5.1 Implicit data

Implicit input data is decoded on the CYCIN field:

Scale1Datas	TeNod4TDatas	
CYCIN	TeNod4TCYCIN	
RefreshCounter	DINT	6559507
IOStatus	UINT	0
b00Reserved	BIT	FALSE
b01Reserved	BIT	FALSE
b02ErrorCode0	BIT	FALSE
b03ErrorCode1	BIT	FALSE
b04NoMotion	BIT	TRUE
b05ZeroWeight	BIT	FALSE
b06EEPROMFailure	BIT	FALSE
b07Reserved	BIT	FALSE
b08Input1ON	BIT	FALSE
b09Input2ON	BIT	FALSE
b10Output1ON	BIT	FALSE
b11Output2ON	BIT	FALSE
b12Output3ON	BIT	FALSE
b13Output4ON	BIT	FALSE
b14TareActive	BIT	TRUE
b15Reserved	BIT	FALSE
Gross	DINT	90
Tare	DINT	40
Net	DINT	50
ADPoints	DINT	14191
CommandEcho	UINT	0
CommandResult	UINT	0
CALDAT	TeNod4TCAL	

5.1.5.2 eNod configuration information

By default, the main bloc copies the full configuration from eNod to data structure.
This is done only once, when the eNod is online the first time.

Scale1Datas	TeNod4TDatas	
CYCIN	TeNod4TCYCIN	
CALDAT	TeNod4TCAL	
MeasureRange	DINT	3000
CalibSegment	INT	3
ResAlign1	INT	0
CalibSegment1	DINT	100
CalibSegment2	DINT	200
CalibSegment3	DINT	350
SensorSens	DINT	200000
StepMes	INT	1
ResAlign2	INT	0
CalibZero	DINT	-749
CalibCoeff1	REAL	16.671
CalibCoeff2	REAL	16.684
CalibCoeff3	REAL	16.6153336
CalibSpanCorr	DINT	1000000
gValueCalib	DINT	9805470
gValueLocal	DINT	9805470
ZeroOffset	DINT	0
FixedTare	DINT	40
SensorRef	DINT	0
SensorTol	INT	30
FILDAT	TeNod4TFILTER	
LEGAL	TeNod4TLEGAL	
IOCFG	TeNod4TIOCFG	

CALDAT section of the data structure contains all calibration data
FILDAT section of the structure contains all filter configuration data
IOCFG section of the structure contains all I/O configuration data
LEGAL section of the structure contains legal configuration data and legal DSD results.

All structures details can be found on the Scaime documentation “eNod4T Ethernet” chapter 16 “Ethernet/IP resgister map”. The structures keep the same names and order than the register map objects. Details of every register can be found in the same documentation.

5.1.5.3 Bloc parameters and commands

The bloc contains direct command parameters.

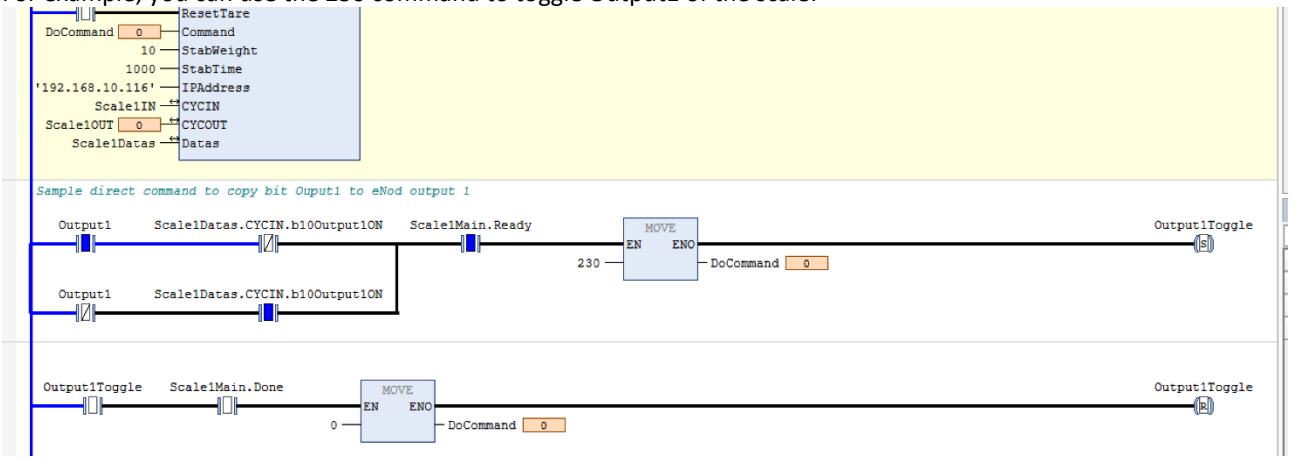
Inputs parameters and commands:

- “Zero”, execute a zero command on the scale
- “Tare”, execute a tare command on the scale
- “ResetTare”, suppress an existing tare
- “Command”, send a command to the scale. List of available commands are described Scaime documentation.
- “StabWeight”, weight tolerance used for weight stability calculation. It is in scale units.
- “StabTime”, delay of stability status when weight stays within the weight tolerance. It is in milliseconds units.

Outputs values:

- “Ready”, thus status indicates a new command can be done. This includes “Zero”, “Tare”, “ResetTare” and “Command” inputs parameters. If you use other add-on instructions, the “Ready” status can also change when other blocs send commands to the scale. It is recommended to test “Ready” status before sending a command.
- “Run”, it is ON when a command is in progress
- “Done”, it is ON when a command is finished successfully. “Done” stays ON until the command is reset.
- “Error”, it is ON when a command in finished with an error or cannot be executed. “Error” stays ON until the command is reset.
- “Timeout”, it is ON when there is no implicit communication with the eNod during more than 500ms. Note than some command can result with a “TimeOut” error like “RESET” command.
- “Stability”, calculated stability using implicit net weight, “StabWeight” and “StabTime” parameters.

For example, you can use the 230 command to toggle Output1 of the scale.

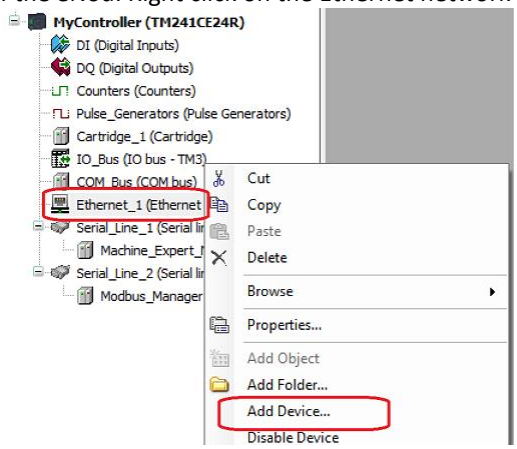


Note, this will work only if the output 1 is defined as “remote” on the scale configuration.

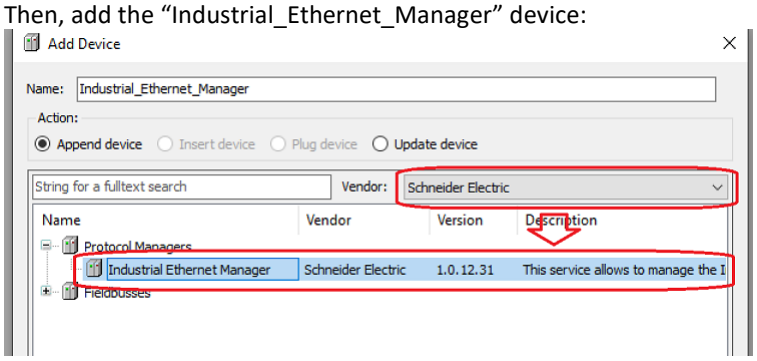
5.1.6 Compatibility with different Machine Expert versions

Blocs has been designed and tested with Machine Expert version 1.2. The CPU used for the test project is a M251MESE. For the previous versions, the XML exports of the blocs are provided to be imported directly in the project without loading the library.

For the M241 CPU, by default the Ethernet manager is not configured and must be added first to allow the configuration of the eNod. Right click on the Ethernet network and select “Add device…”



Then, add the “Industrial_Ethernet_Manager” device:



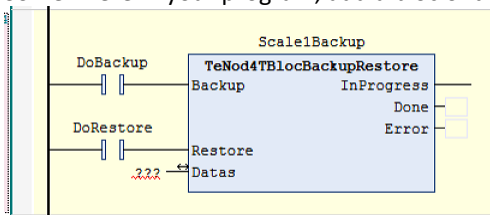
Then the eNod can be added as described earlier in this document.

5.2 “TeNod4TFBBackupRestore” to restore and backup configuration

This bloc allows backup and restoring full configuration and calibration data of the scale.

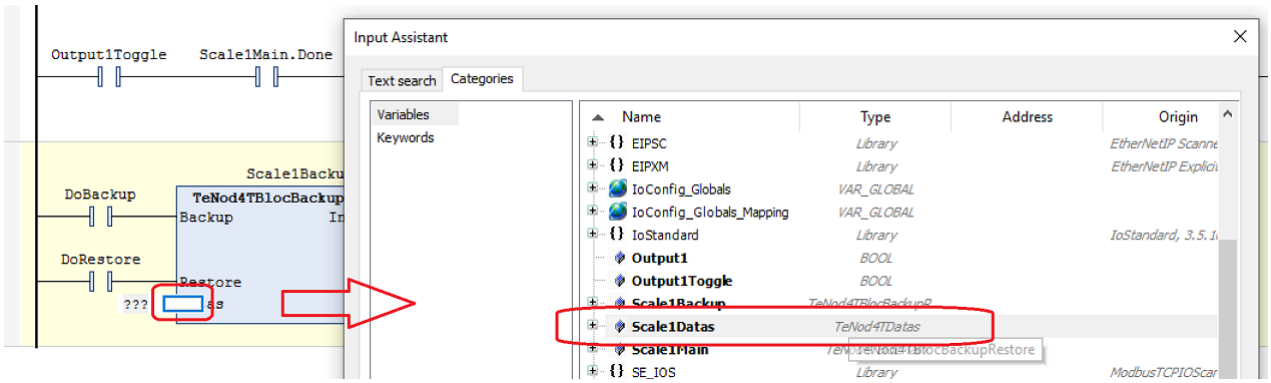
5.2.1 Creating the instance

Somewhere in your program, add a bloc of the Scaime instruction and select an instance name:



5.2.2 Select global data for the eNod.

This structure must be the one created for the main bloc.



5.2.3 Using the instruction

5.2.3.1 Bloc parameters and commands

The bloc contains direct command parameters.

Inputs parameters and commands:

- “Backup”, execute a copy of all parameters from the scale to the PLC
- “Restore”, execute a copy of all parameters from the PLC to the scale.

Outputs values:

- “InProgress”, it is ON when a command is in progress
- “Done”, it is ON when a command is finished successfully. “Done” stays ON until the command is reset.
- “Error”, it is ON when a command in finished with an error or cannot be executed. “Error” stays ON until the command is reset.

5.2.3.2 Backup of the configuration

To copy the information, just set to 1 the input “Backup” of the bloc. This will copy the whole part of the structure members

- CAL for calibration data
- FIL for filter data
- IOCFG for input/output configuration data
- LEGAL for legal configuration and data

Note that all existing data on the structure will be erased. The backup also read the current DSD values of the legal structure.
Backup can be done while the scale is in use.

5.2.3.3 Restoring the configuration

To restore the configuration, just set to 1 the input “Restore” of the bloc. This will send to the scale all configuration data:

- CAL for calibration data. The last saved calibration will be sent, so the scale will be calibrated with the last calibration data after the restore
- FIL for filter data
- IOCFG for input/output configuration data
- LEGAL, only the legal configuration fields will be restored: ZeroConfig, Stability, Decimal, Unit and HMIName.

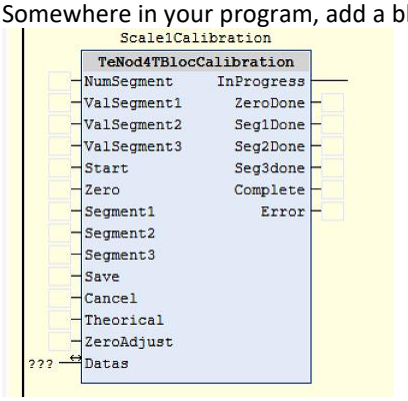
After the restore, an EEPROM save command is sent to save the data. Then a RESET command is sent to restart the eNod with the restored configuration. During this moment the communication will be temporally lost.

5.3 "TeNod4TFBCalibration" to calibrate a scale

This bloc allows to calibrate the scale.

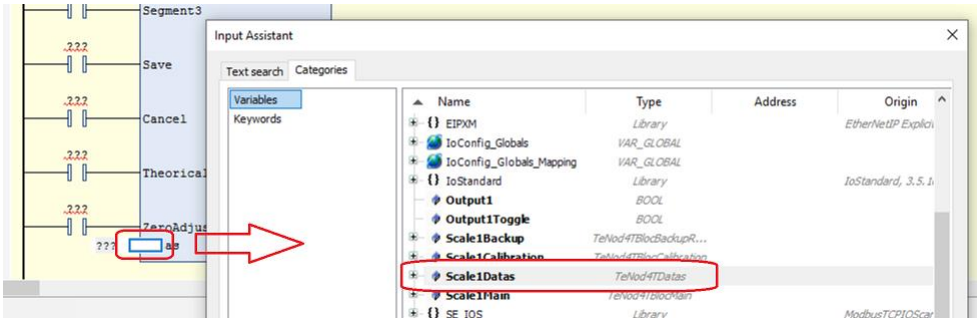
5.3.1 Creating the instance

Somewhere in your program, add a bloc of the Scaime instruction and select an instance name:



5.3.2 Select global data for the eNod.

This structure must be the one created for the main bloc.



5.3.3 Using the instruction

5.3.3.1 Bloc parameters and commands

The bloc contains direct command parameters.

Inputs parameters and commands:

- "NumSegment", number of calibration segment for calibration
- "ValSegment1", weight value in scale unit of the first calibration segment
- "ValSegment2", weight value in scale unit of the second calibration segment
- "ValSegment3", weight value in scale unit of the third calibration segment
- "Start", start of a physical calibration procedure.
- "Zero", execute to zero point of the calibration procedure
- "Segment1", execute the first point of the calibration procedure
- "Segment2", execute the second point of the calibration procedure
- "Segment3", execute the third point of the calibration procedure
- "Save", save the calibration procedure. This will save calibration point to EEPROM and copy the calibration data to PLC memory.
- "Cancel", cancel the physical calibration in progress.
- "Theorical", execute a theorical calibration.
- "ZeroAdjust", execute a zero-adjustment command.

Outputs values:

- “InProgress”, it is ON when a command is in progress. InProgress stays ON until the “Save” command is executed indicating a calibration is in progress.
- “ZeroDone”, it is ON when the “zero” is finished successfully. “ZeroDone” stays ON until the command is reset.
- “Seg1Done”, it is ON when the “Seg1Done” is finished successfully. “Seg1Done” stays ON until the command is reset.
- “Seg2Done”, it is ON when the “Seg2Done” is finished successfully. “Seg2Done” stays ON until the command is reset.
- “Seg3Done”, it is ON when the “Seg3Done” is finished successfully. “Seg3Done” stays ON until the command is reset.
- “Complete”, it is ON when the “Save” is finished successfully. “Complete” stays ON until the command is reset. When “Complete” is ON, “InProgress” is reset.
- “Error”, it is ON when a command in finished with an error or cannot be executed. “Error” stays ON until the command is reset.

Calibration must respect a command sequence; valid command sequences are:

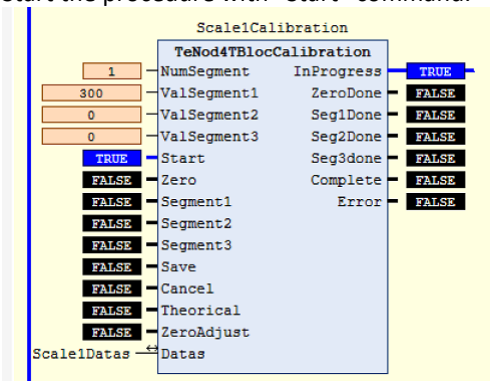
- Start, Zero, Segment1, Save
- Start, Zero, Segment1, Segment2, Save
- Start, Zero, Segment1, Segment2, Segment3, Save
- Start, Segment1, Save
- Zero Adjust, Save
- Theorical, Save

5.3.3.2 Full physical calibration

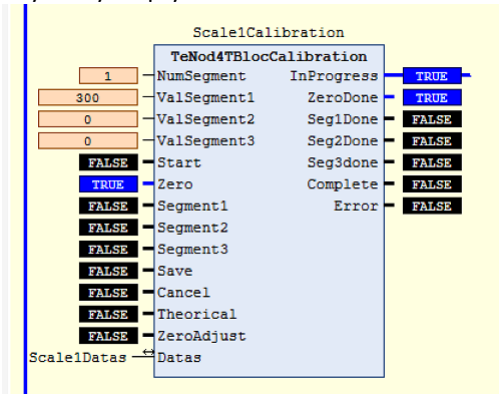
Calibration can be done using an HMI program that uses the input parameters of the bloc. You can also do it directly using Machine Expert by writing bloc parameters.

Before starting procedure, the inputs parameters NumSegment and ValSegment1 must be set. ValSegment2 must be set if NumSegment is 2 or 3. ValSegment3 must be set if NumSegment is 3.

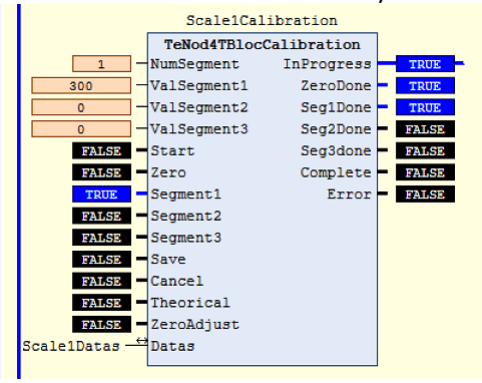
Start the procedure with “Start” command.



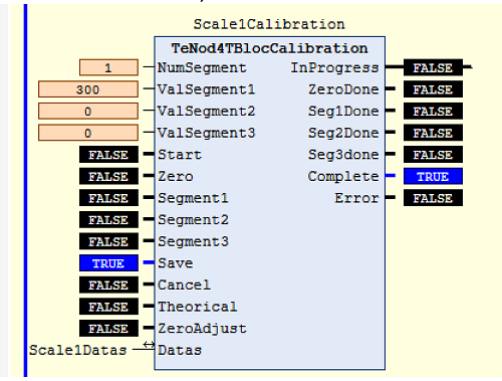
Physically empty the scale and send the “Zero” command



Put on the scale the standard load you need according the “ValSegment1” value and then send “Segment1” command



In case you have a 2 or 3 segments calibration, use Segment2 and Segment3 commands.
When it is finished, use save command



Note that ValSegmentX can be modified just before the corresponding command SegmentX is sent.

5.3.3.3 Zero adjust calibration

Calibration can be done using an HMI program that uses the input parameters of the bloc. You can also do it directly using TIA portal writing the main instance DB.

Send “Zero Adjust” command

Input			
NumSegment	Int	0	1
ValSegment1	DInt	0	300
ValSegment2	DInt	0	0
ValSegment3	DInt	0	0
Start	Bool	false	FALSE
Zero	Bool	false	FALSE
Segment1	Bool	false	FALSE
Segment2	Bool	false	FALSE
Segment3	Bool	false	FALSE
Save	Bool	false	FALSE
Cancel	Bool	false	FALSE
Theorical	Bool	false	FALSE
ZeroAdjust	Bool	false	TRUE
Output			
InProgress	Bool	false	TRUE
ZeroDone	Bool	false	FALSE
Seg1Done	Bool	false	FALSE
Seg2Done	Bool	false	FALSE
Seg3Done	Bool	false	FALSE
Complete	Bool	false	FALSE
Error	Bool	false	FALSE

Send “Save” command

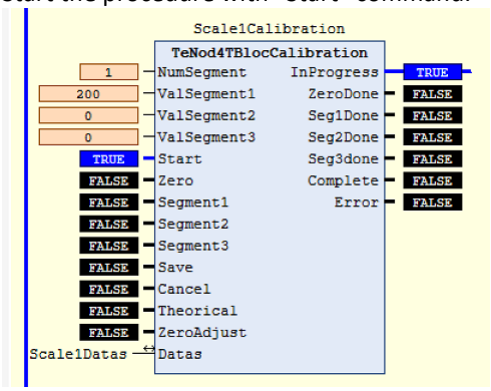
Input			
NumSegment	Int	0	1
ValSegment1	DInt	0	300
ValSegment2	DInt	0	0
ValSegment3	DInt	0	0
Start	Bool	false	FALSE
Zero	Bool	false	FALSE
Segment1	Bool	false	FALSE
Segment2	Bool	false	FALSE
Segment3	Bool	false	FALSE
Save	Bool	false	TRUE
Cancel	Bool	false	FALSE
Theorical	Bool	false	FALSE
ZeroAdjust	Bool	false	FALSE
Output			
InProgress	Bool	false	FALSE
ZeroDone	Bool	false	FALSE
Seg1Done	Bool	false	FALSE
Seg2Done	Bool	false	FALSE
Seg3Done	Bool	false	FALSE
Complete	Bool	false	TRUE
Error	Bool	false	FALSE

5.3.3.4 Span adjust calibration

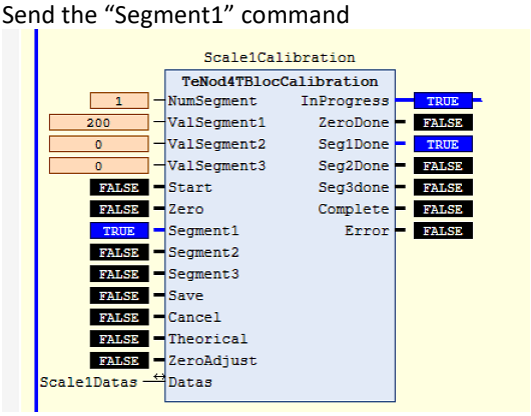
Calibration can be done using an HMI program that uses the input parameters of the bloc. You can also do it directly using Machine Expert by writing bloc parameters.

For this procedure, “NumSegment” must be set to 1 even if the original calibration procedure is 2 or 3 segments.

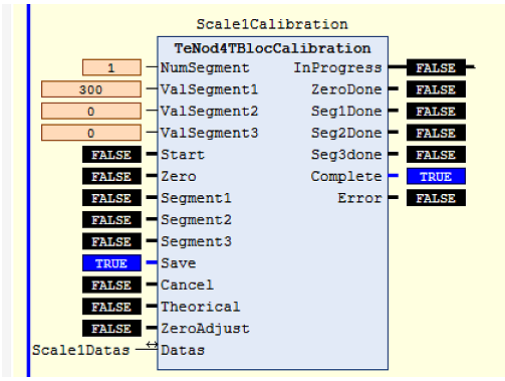
Start the procedure with “Start” command.



Send the “Segment1” command



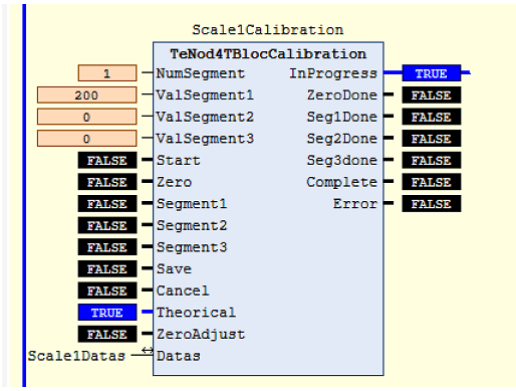
Send the “Save” command



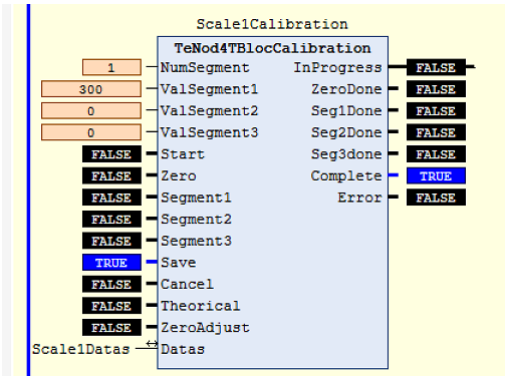
5.3.3.5 Theorical calibration

Calibration can be done using an HMI program that uses the input parameters of the bloc. You can also do it directly using Machine Expert by writing bloc parameters.

Send “Theorical” command



Send “Save” command



5.3.3.6 Cancel calibration

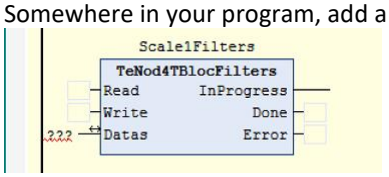
Calibration can be cancelled anytime until the save command is sent. “Cancel” command will reset the eNod and calibration will return to the last calibration values. The “Cancel” function do not have a “Done” value. Cancel is done when the “InProgress” output goes to zero.

5.4 "TeNod4TFBFilters" to read and write filter configuration

This bloc allows to read and write filter configuration of the scale.

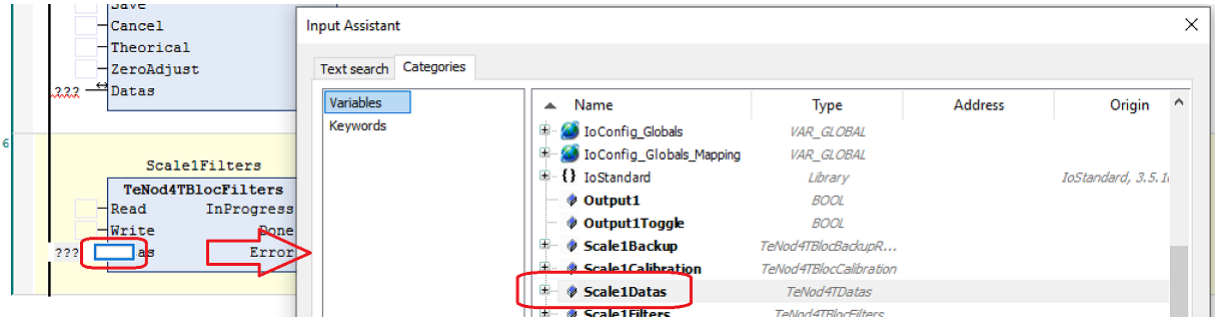
5.4.1 Creating the instance

Somewhere in your program, add a bloc of the Scaime instruction and select an instance name:



5.4.1 Select global data for the eNod

This structure must be the one created for the main bloc.



5.4.2 Using the instruction

5.4.2.1 Bloc parameters and commands

The bloc contains direct command parameters.

Inputs parameters and commands:

- "Read", execute a copy of all parameters from the scale to the PLC
- "Write", execute a copy of all parameters from the PLC to the scale.

Outputs values:

- "InProgress", it is ON when a command is in progress
- "Done", it is ON when a command is finished successfully. "Done" stays ON until the command is reset.
- "Error", it is ON when a command is finished with an error or cannot be executed. "Error" stays ON until the command is reset.

This bloc uses the FIL section of the data structure.

5.4.2.2 Read the filter configuration

To copy the information, just set to 1 the input "Read" of the bloc. This will copy the filter configuration to FIL member of the data structure.

5.4.2.3 Set filter configuration

First you need to set the desired filter configuration on the FIL section of the eNod data structure.
Then, you set the "Write" command of the bloc.
New configuration is directly is use. The command also saves the new configuration to EEPROM.

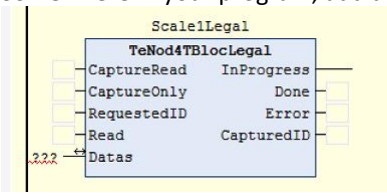
Note that if the A/D parameters values are changed, the changed will be used on the next reboot or reset of the eNod

5.5 "TeNod4TFBLegal" to manage DSD legal values

This bloc allows to create DSD values and read them

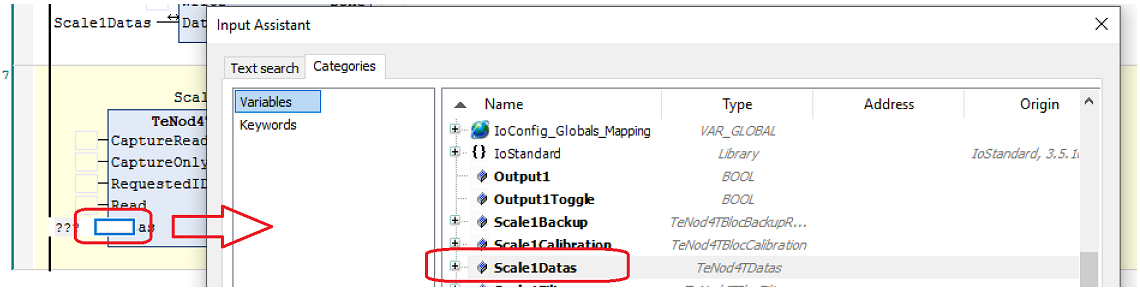
5.5.1 Creating the instance

Somewhere in your program, add a bloc of the Scaime instruction and select an instance name:



5.5.2 Select global data for the eNod.

This structure must be the one created for the main bloc.



5.5.3 Using the instruction

5.5.3.1 Bloc parameters and commands

The bloc contains direct command parameters.

Inputs parameters and commands:

- "CaptureRead", create a DSD measure and copy it to PLC memory.
- "CaptureOnly", create a DSD measure but to not read it on the scale. Measure can be readen later. "CaptureOnly" is faster than "CaptureRead" command dans do not erase the DSDRead datas.
- "RequestedID", parameter of the "Read" command
- "Read", reads "RequestedID" from scale memory and copies it to PLC memory.

Outputs values:

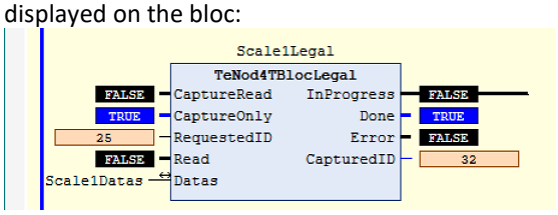
- "InProgress", it is ON when a command is in progress
- "Done", it is ON when a command is finished successfully. "Done" stays ON until the command is reset.
- "Error", it is ON when a command is finished with an error or cannot be executed. "Error" stays ON until the command is reset.
- "CapturedID", the last ID generated by "CaptureRead" or "CaptureOnly".

The “CaptureRead” or the “Read” commands read the scale memory and copy the DSD values on the LEGAL members of the data structure:

Scale1Datas	TeNod4TDatas	
CYCIN	TeNod4TCYCIN	
CALDAT	TeNod4TCAL	
FILDAT	TeNod4TFILTER	
LEGAL	TeNod4TLEGAL	
eNodVersion	INT	24787
ModbusCfg	INT	1793
LegalVersion	BYTE	2
LegalSwitch	BYTE	0
LegalCount	BYTE	2
ResAlign1	BYTE	0
LegalChecksum	INT	0
ZeroConfig	INT	0
Stability	BYTE	1
Decimal	BYTE	2
UNIT	STRING(4)	<Value of the ex...
ResAlign2	BYTE	0
ResAlign3	BYTE	0
DSDLastID	DINT	30
DSDOldestID	DINT	1
DSDCurrentID	DINT	25
DSDReadID	DINT	25
DSDReadNet	DINT	20
DSDReadTare	DINT	0
DSDReadStatus	INT	7298
DSDReadCheckSum	INT	-7343
HMIName	STRING(4)	'ABCD'

5.5.3.2 Capture a DSD value

Set the command “CaptureRead” or “CaptureOnly” of the bloc. When the command is done, the generated ID is displayed on the bloc:



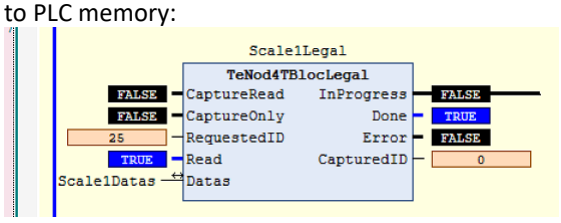
The last ID is also on the structure on the field

DSDLastID	DINT	32

If you have used the “CaptureRead” command, the DSD information are located on he DSDRead part of the LEGAL section of the structure.

5.5.3.3 Read a previous DSD value

Set “RequestedID” value with the ID you want to read, Set the command “Read” to get DSD information from the scale to PLC memory:



The DSD information are located on the DSDRead part of the LEGALsection of the structure:

DSDCurrentID	DINT	25
DSDReadID	DINT	25
DSDReadNet	DINT	20
DSDReadTare	DINT	0
DSDReadStatus	INT	7298
DSDReadCheckSum	INT	-7343
HMIName	STRING(4)	'ABCD'

6 COMMUNICATION ERROR CODES

In case a command returns an error, it can be for various reasons.
For direct command, refer to Scaime documentation for command conditions.

In some cases, “main” bloc cannot read or write data on the scale. This is mainly due to data format. As a lot of registers can be read or written, the data structure contains the current read or written register.
If a communication error occurs the bit “**Datas.INTER.MSGError**” is set.
In this case check the “**Datas.MSGGroup**” data field. If value is from 1 to 99, it is a read command. If value is from 101 to 199 it is a write command.
Then check the following fields:
Datas.MSGClass: It is the CIP class name of the register
Datas.MSGAttribute: It is the CIP attribute value of the register
With these two values, refer to chapter 16 to get the register name.

For example, class 105 attribute 2 is the number of segments in calibration procedure.

There are two error codes stored in CommError and OperError. These two codes are the ones from the Schneider library for explicit messaging under ethernet/IP.

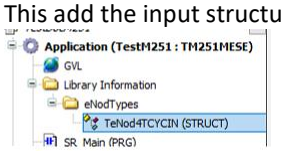
The description of the error codes is on the Schneider online documentation here:
https://product-help.schneider-electric.com/Machine%20Expert/V1.1/en/ESMEEtherNetIP/index.htm?#t=ESMEEtherNetIP%2FEtherNetIP_Explicit_Messaging_Library%2FEtherNetIP_Explicit_Messaging_Library-13.htm

7 BASIC COMMUNICATION

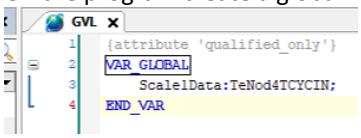
If you do not need any calibration or parameters changes of the eNod by the PLC, you can use this basic procedure that only read implicit data. For this solution you do not need the library.

First you need to configure the scale on the Ethernet network as describe on §III.

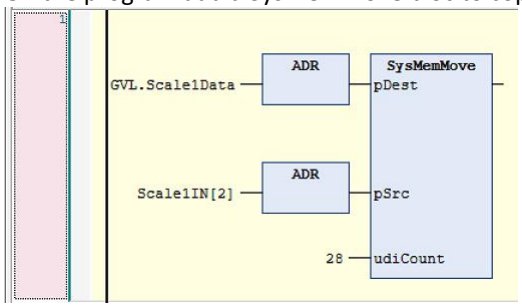
Then right click on the application tree and select “PLCOpenXML” import option.
Open the file TeNod4TCYCIN.xml supplied by Scaime.
This add the input structure in the application:



On the program create a global variable of this type:



On the program add a SysMemMove bloc to copy the data to the new variable:



If you prefer you can do it as ST language:
SysMemMove(pDest:=ADR(GVL.ScaleData) , pSrc:=ADR(Scale1IN[2]) , udiCount:= 28);

The new structure gets the readable data:

GVL | SR_Main

TestM251.Application.GVL

Expression	Type	Value
Scale1Data	TeNod4TCYCIN	
RefreshCounter	DINT	3986605
IOStatus	UINT	0
b00Reserved	BIT	FALSE
b01Reserved	BIT	FALSE
b02ErrorCode0	BIT	FALSE
b03ErrorCode1	BIT	FALSE
b04NoMotion	BIT	TRUE
b05ZeroWeight	BIT	FALSE
b06EEPROMFailure	BIT	FALSE
b07Reserved	BIT	FALSE
b08Input1ON	BIT	FALSE
b09Input2ON	BIT	FALSE
b10Output1ON	BIT	FALSE
b11Output2ON	BIT	FALSE
b12Output3ON	BIT	FALSE
b13Output4ON	BIT	FALSE
b14TareActive	BIT	TRUE
b15Reserved	BIT	FALSE
Gross	DINT	90
Tare	DINT	40
Net	DINT	50
ADPoints	DINT	14178
CommandEcho	UINT	0
CommandResult	UINT	0