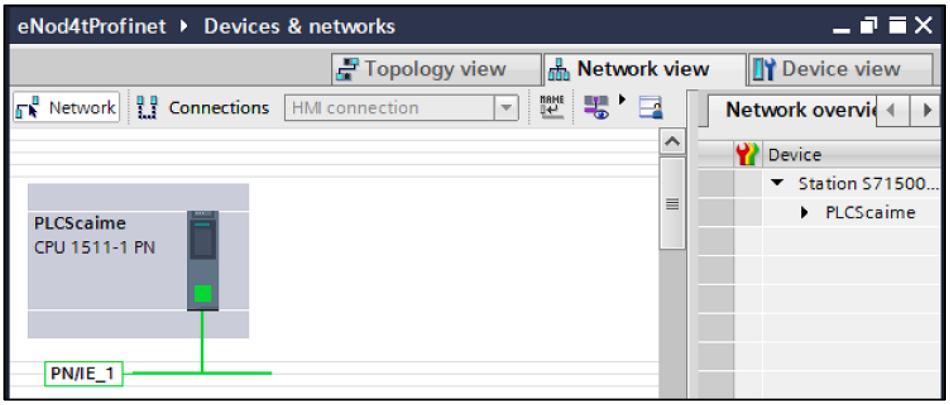
User Manual

Function Block for eNod4-T PNS

For TIA Portal, Siemens



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

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.

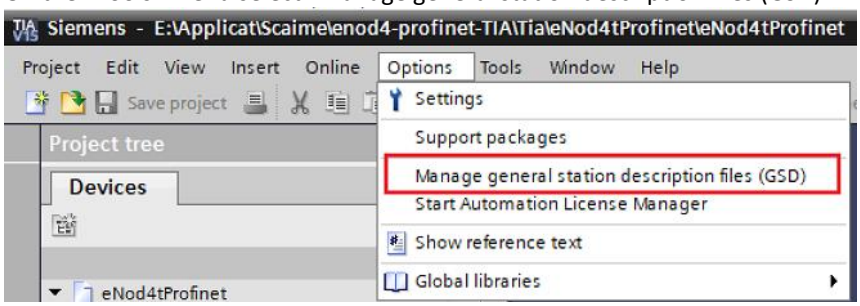
The eNod4-T Block function for TIA Portal is an open source code that eases the integration of an eNod4 with Siemens PLC. This tool integrates pre-programmed functions for TIA Portal. You can quickly get your measurement data, calibrate or restore your eNod4 using this code.

This user manual describes how to integrate the block function in TIA Portal and communicate with an eNod4-T Profinet through a Siemens PLC.

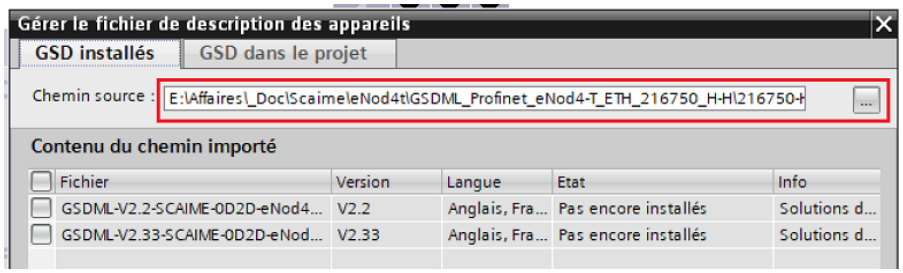
2 INSTALLING GSDML FILE IN TIA PORTAL

Before using eNod4 in application, GSD file must be imported in TIA Portal to add the eNod4 module in hardware catalog.

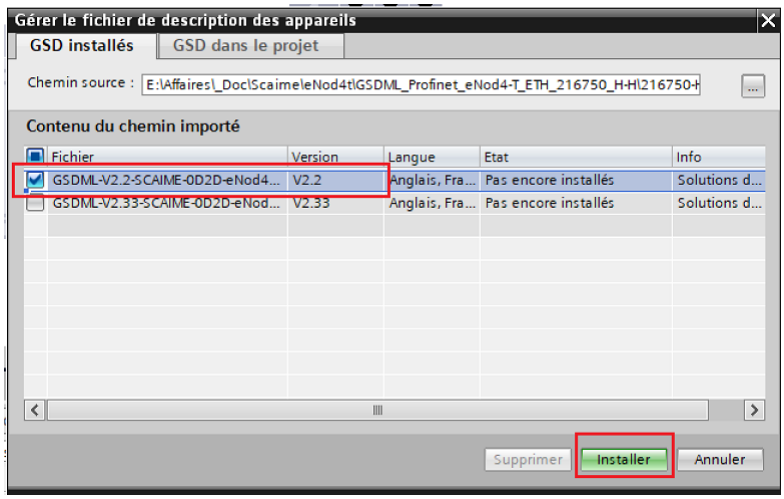
On the “Tools” menu select “Manage general station description files (GSD)”.



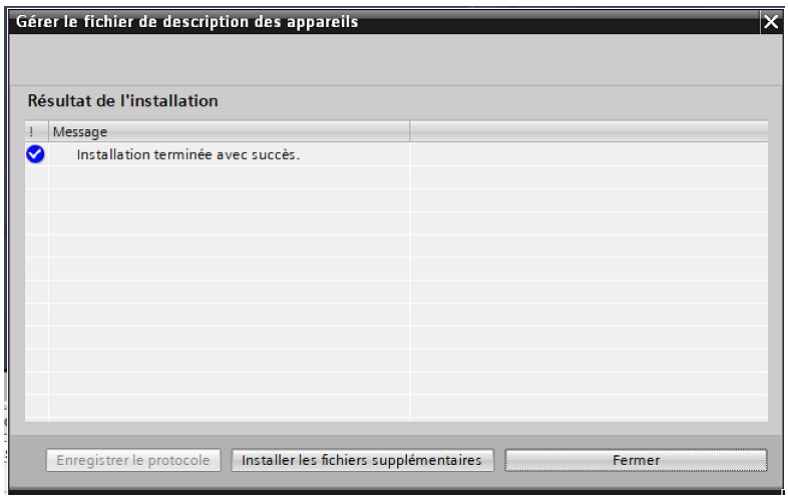
On the next window select the directory where the GSD files are available. It automatically displays the two versions of the GSD files.



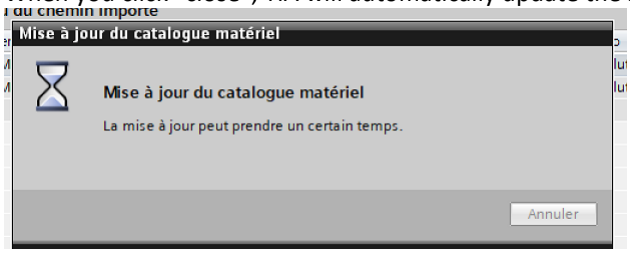
Select the version 2.2 and click on “Install” button.



When the installation is completed, you have the confirmation window:



When you click “close”, TIA will automatically update the hardware catalog.

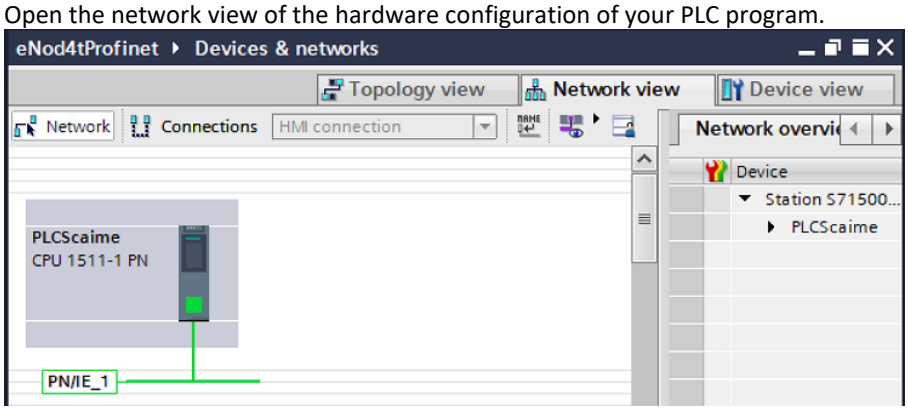


When the update is finished, the eNod4T is available on the Hardware catalog.

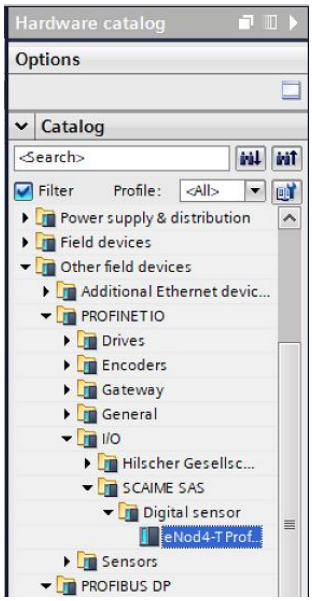
3 ADDING ENOD4T IN THE PROJECT

3.1 Adding eNod4T on the Ethernet configuration

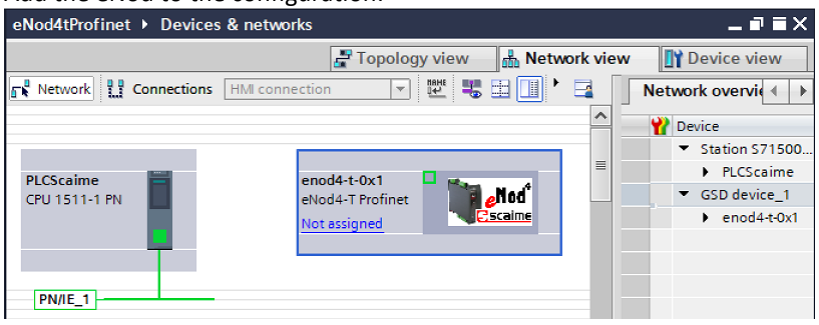
Open the network view of the hardware configuration of your PLC program.



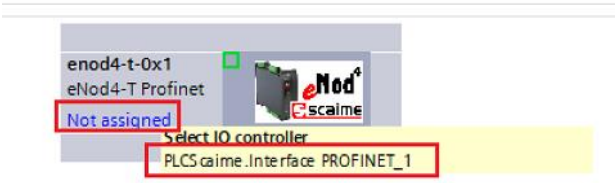
On the right, on the hardware catalog, search for the “eNod” or select this path:
Others field devices
Profinet IO
I/O
SCAIME SAS
Digital sensor
eNod4-T Profinet



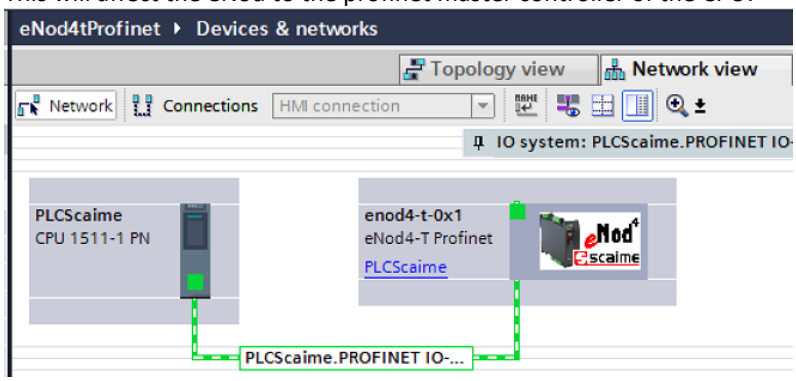
Add the eNod to the configuration:



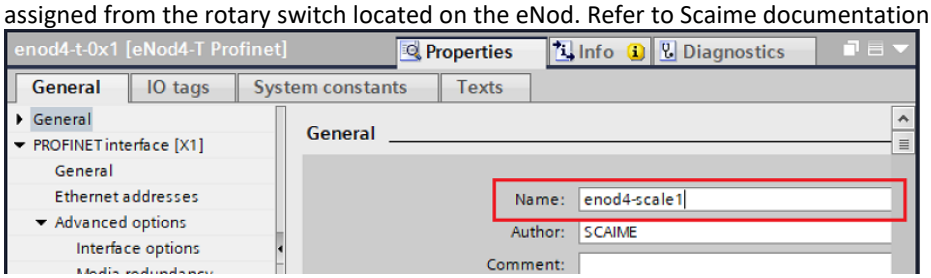
Click on the link “Not assigned” to the select the master controller for the eNod.



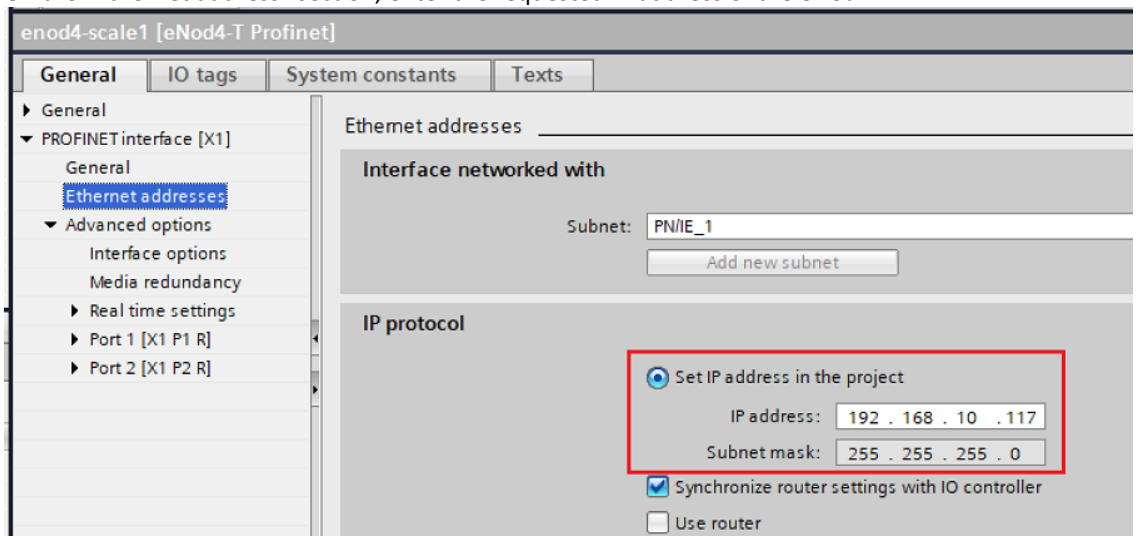
This will affect the eNod to the profinet master controller of the CPU:



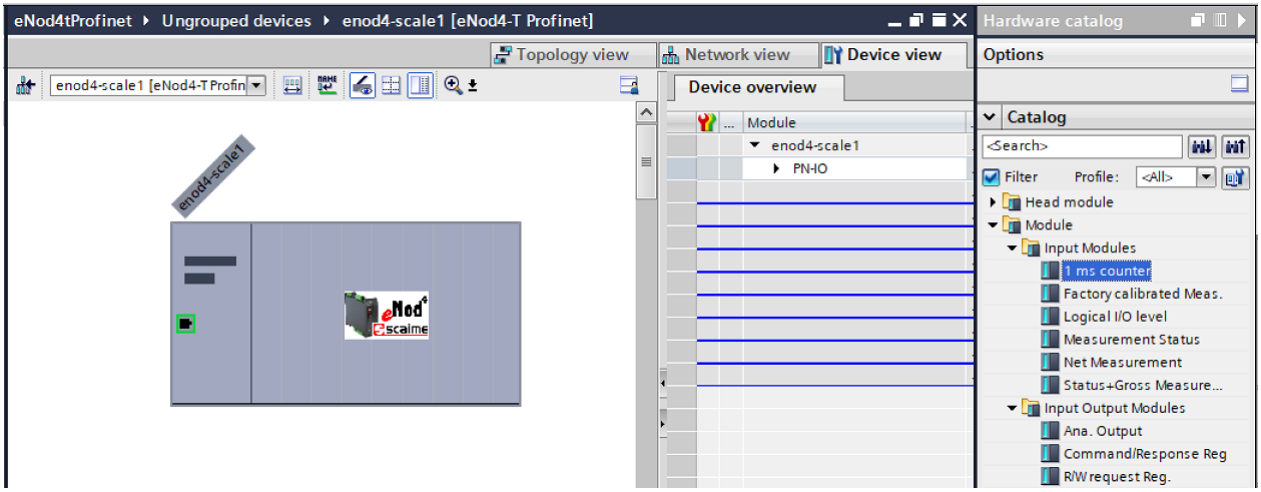
Double click on the eNod to open the equipment properties.
Then enter the equipment name. This name must match the name set on the eNod. The name can be set manually or assigned from the rotary switch located on the eNod. Refer to Scaime documentation about how to set the name.



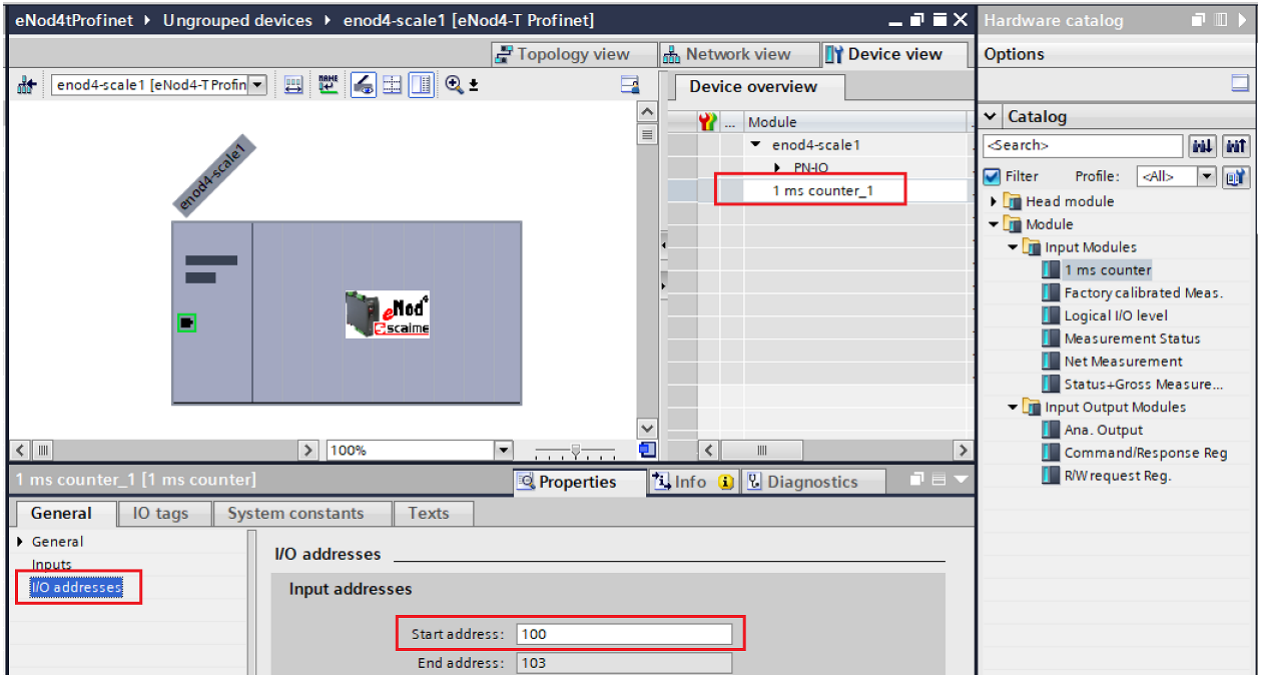
On the “Ethernet address” section, enter the requested IP address of the eNod4



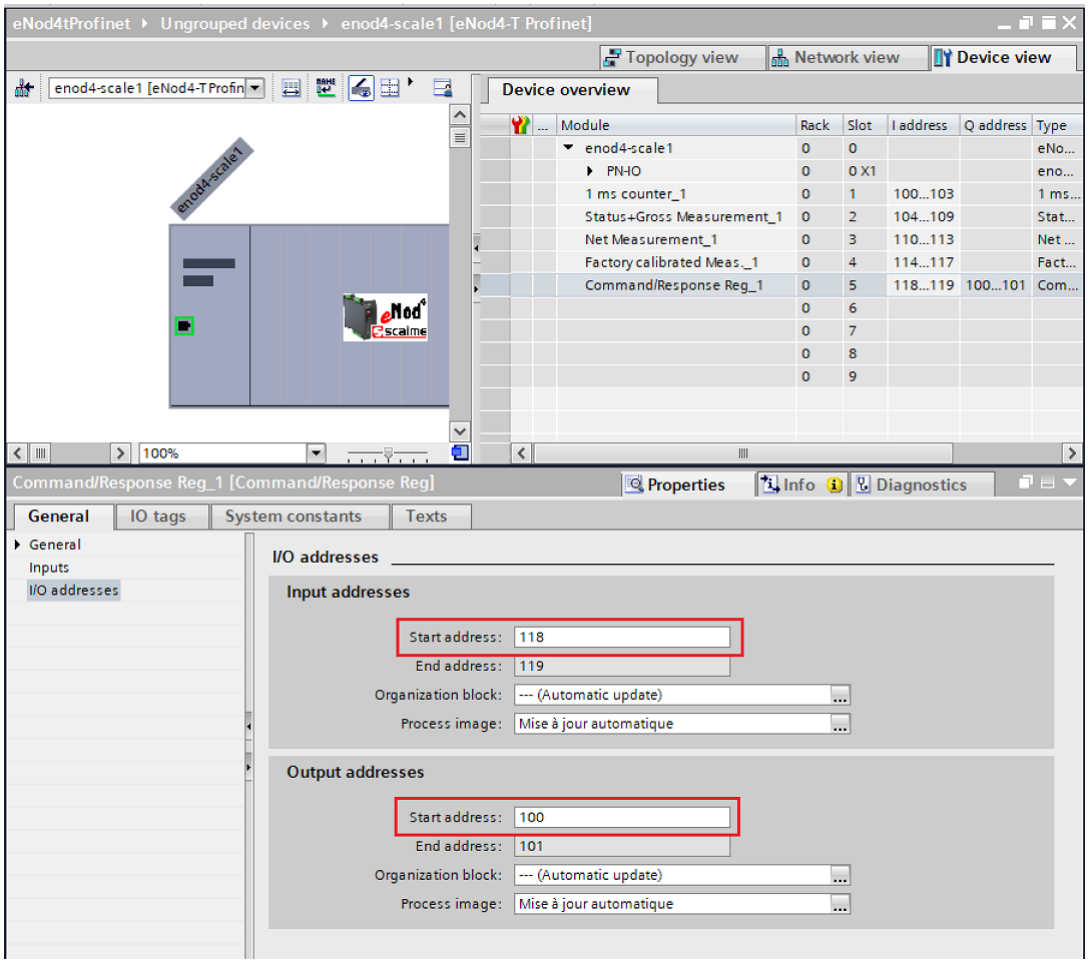
Now, open the “Device overview” to manage the sub-modules. This will define the data exchange by implicit messages.
On the right part of the screen you have the available modules.



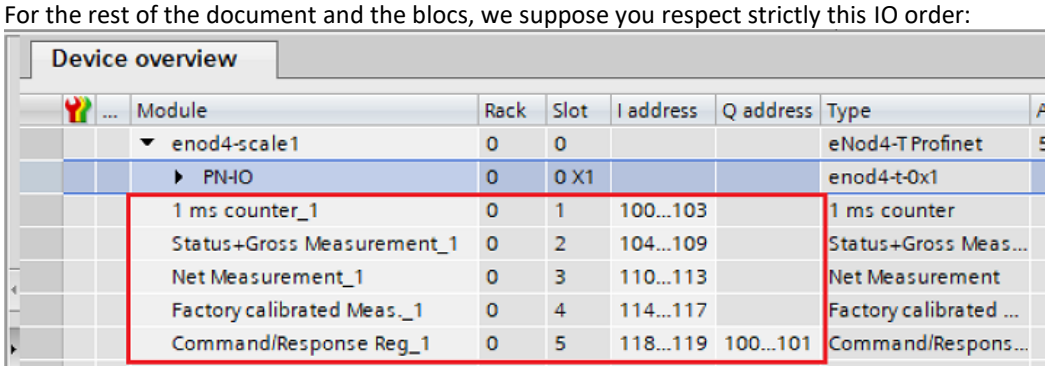
Select on the right the needed modules. For every module you need to set the I/O address.



Note that some modules have an input and an output address:

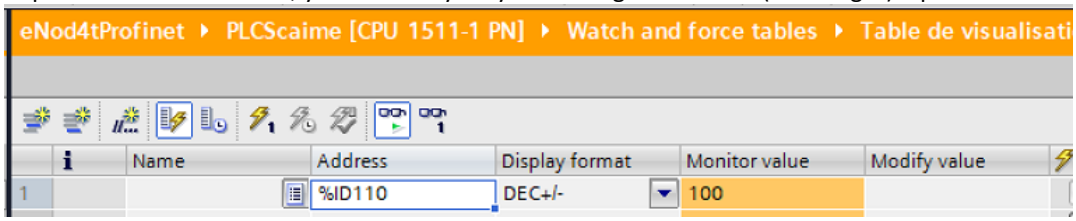


For the rest of the document and the blocs, we suppose you respect strictly this IO order:



The base address can be anyone and not it is not necessary to use the same for input and outputs. But on input and output the addresses must be continuous.

Now you can generate the hardware configuration and transfer the program to the PLC.
When you switch the PLC to RUN mode, the eNod will use automatically the IP address of the project.
Implicit data are refreshed, you can verify it by visualizing the %ID110 (net weight) input word:



3.2 Adding “FB instructions blocs” in the project.

3.2.1 Selection instructions

Various FB instructions have been developed for eNo4T integration.

TeNod4TFBMain.
This one is the base one and it is mandatory in all applications.
All other blocs need this one to communicate with the eNod.

TeNod4TFBBackupRestore
This bloc allows backup and restoring full eNod4T configuration and calibration in PLC memory.

TeNod4TFBCalibration
This bloc helps you to manage scale calibration in physical or theorical mode.

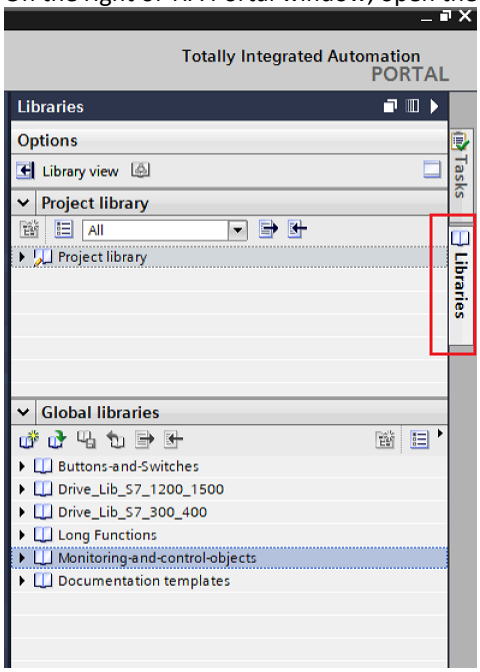
TeNod4TFBFilters
This bloc helps you to adjust eNod filters parameters.

TeNod4TFBLegal
This bloc helps you to use legal functions of eNod4T scale.

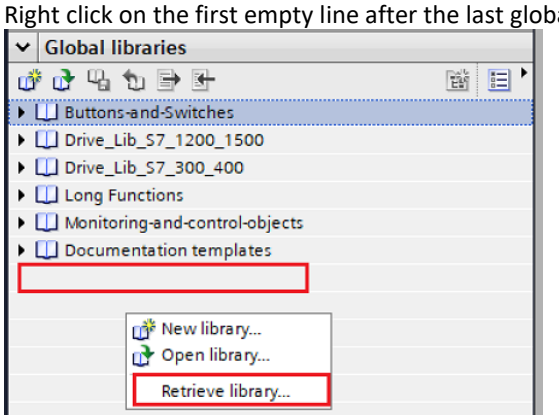
3.2.2 Installing Scaime eNod4-T library

This needs to be done only once in TIA portal. The library will be available for all TIA project.

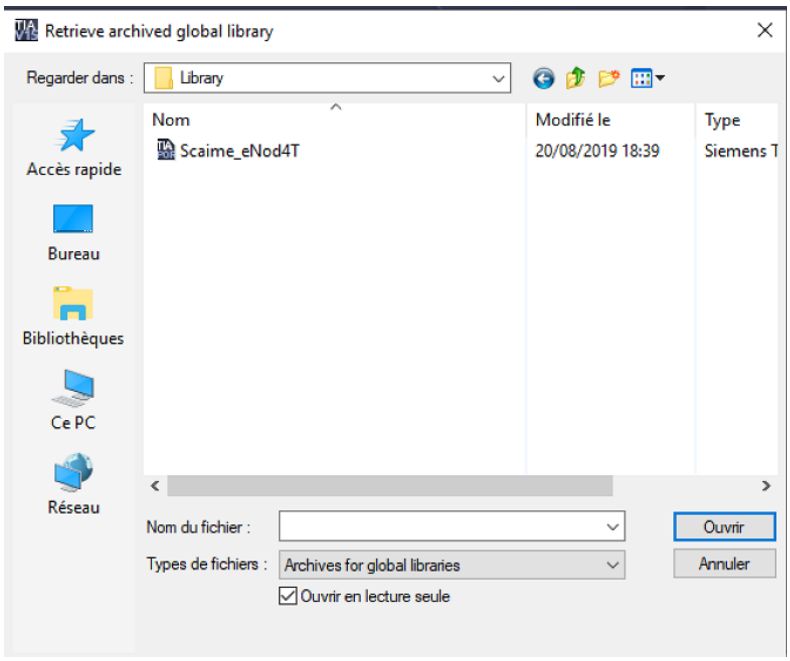
On the right of TIA Portal window, open the library tab.



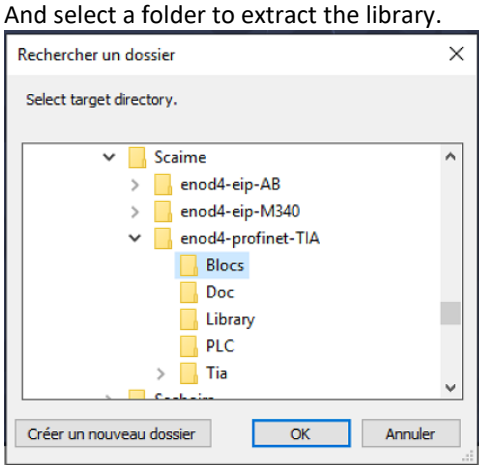
Right click on the first empty line after the last global library, and select “Retrieve library…”



Select the library file “Scaime_eNod4T.zal15” provide by Scaime.

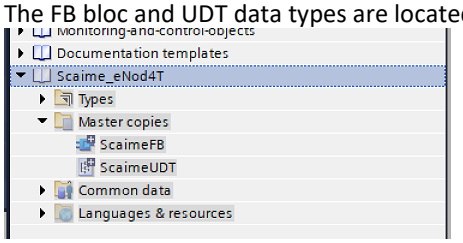


And select a folder to extract the library.



TIA will extract the files and open the library in read only mode.

The FB bloc and UDT data types are located under the master copies group:

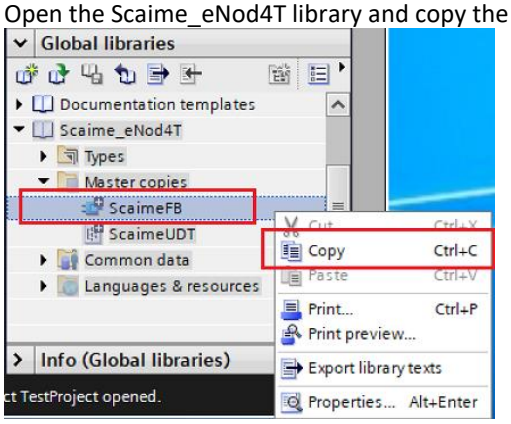


Now the library will be available for all project in TIA portal.

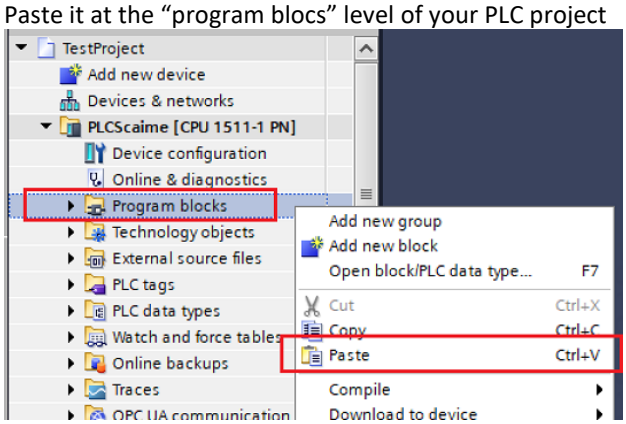
3.2.3 Copy the blocs from global library to PLC project

This needs to be done for every project.

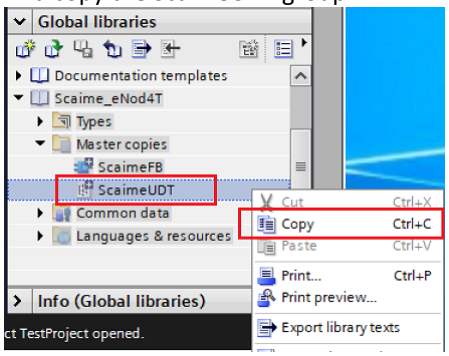
Open the Scaime_eNod4T library and copy the ScaimeFB group.



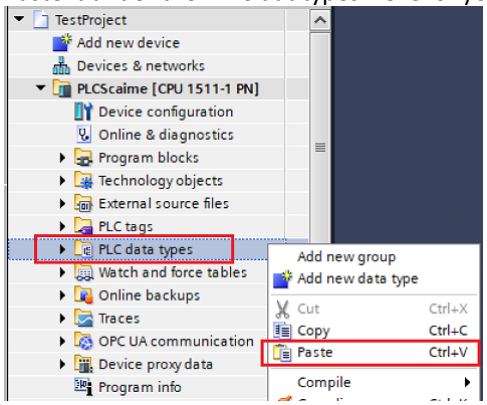
Paste it at the “program blocs” level of your PLC project



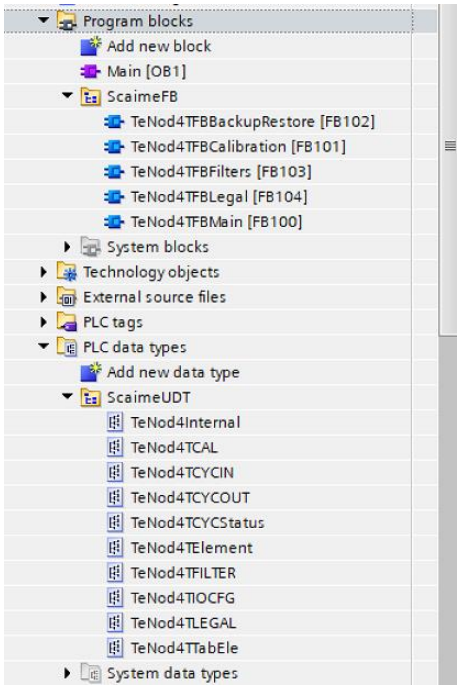
And copy the ScaimeUDT group.



Paste it under the “PLC dat types” level of your PLC project:



These two steps import the necessaries UDT and FB on the PLC project:

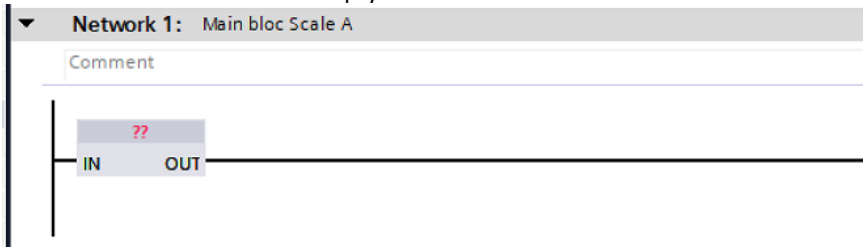

Program blocks
Add new block
Main [OB1]
ScaimeFB
TeNod4TFBBackupRestore [FB102]
TeNod4TFBCalibration [FB101]
TeNod4TFBFilters [FB103]
TeNod4TFBLegal [FB104]
TeNod4TFBMain [FB100]
System blocks
Technology objects
External source files
PLC tags
PLC data types
Add new data type
ScaimeUDT
TeNod4Internal
TeNod4TCAL
TeNod4TCYCIN
TeNod4TCYCOUT
TeNod4TCYCStatus
TeNod4TElement
TeNod4TFILTER
TeNod4TIOCFG
TeNod4TLEGAL
TeNod4TTabEle
System data types

4 USING ADD-ON INSTRUCTIONS IN THE PROJECT

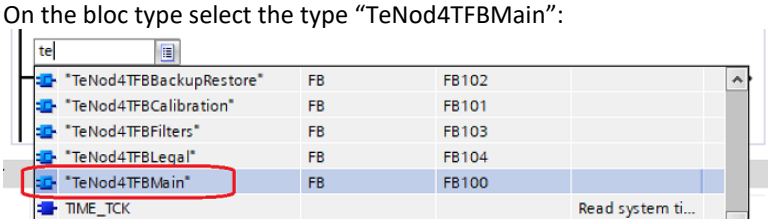
4.1 Main bloc "TeNod4TFBMain"

4.1.1 Creating the instance

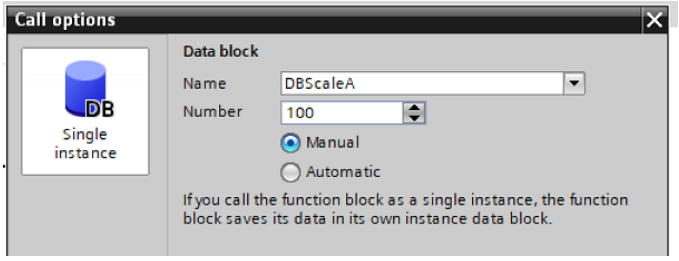
On a new network select an empty bloc.



On the bloc type select the type "TeNod4TFBMain":

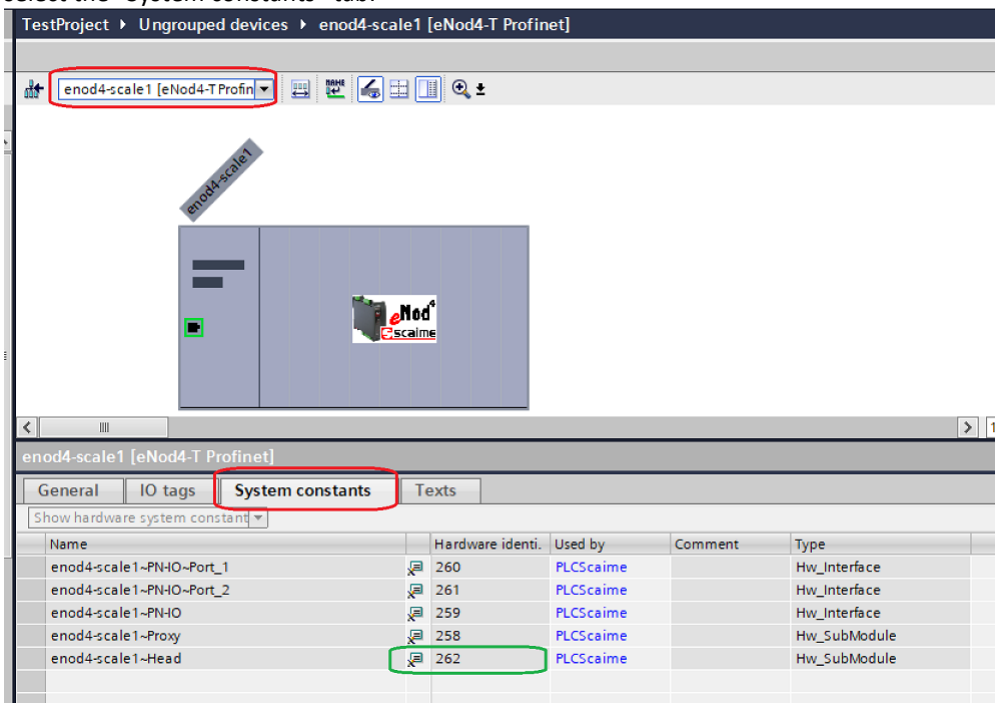


When you validate TIA automatically ask for the creation of the instance DB. This instance DB must be unique for every scale. It will also be used by the other eNod4T FB.
Give a name to the DB, the DB number can be automatic or manual, it does not matter. Let's call it DBScaleA:



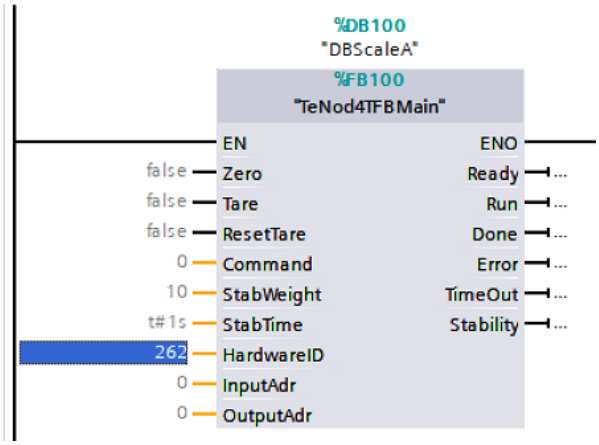
4.1.2 Select the hardware ID for the eNod.

This ID is generated by TIA and is unique for every hardware component on the project. It is used for all S7 communications.
To find the hardware ID, go to “device configuration”. Select the eNod; and go to property view. On the property view, select the “System constants” tab:



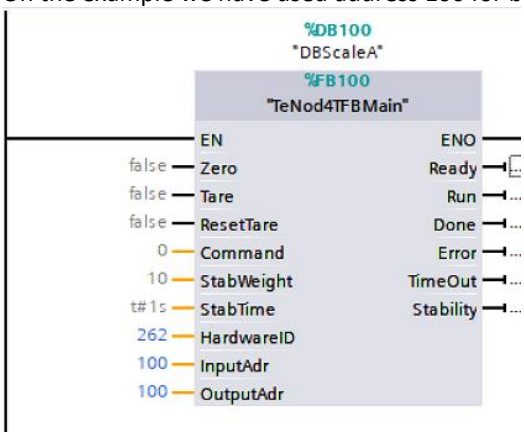
The hardware ID needed for the communication is the “Head” id.

Enter this ID on the main FB bloc:



4.1.3 Select the I/O addresses

Cyclic data are located on the I/O zone. To access this data, just enter on the bloc the address of the first input word used and the address of the first output word used.
On the example we have used address 100 for both inputs and outputs words.



4.1.4 Using the instruction

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

4.1.4.1.1 Implicit data.

Implicit input data are on the member CYCIN of the scale DB:

	CYCIN	"TeNod4TCYCIN"		
	RefresCounter	DWord	16#0	16#00F0_0586
	Status	"TeNod4TCYCStatus"		
	b00Reserved	Bool	false	FALSE
	b01Reserved	Bool	false	FALSE
	b02ErrorCode0	Bool	false	FALSE
	b03ErrorCode1	Bool	false	FALSE
	b04NoMotion	Bool	false	TRUE
	b05ZeroWeight	Bool	false	FALSE
	b06EEPROMFai...	Bool	false	FALSE
	b07Reserved	Bool	false	FALSE
	b08Input1ON	Bool	false	FALSE
	b09Input2ON	Bool	false	FALSE
	b10Output1ON	Bool	false	FALSE
	b11Output2ON	Bool	false	FALSE
	b12Output3ON	Bool	false	FALSE
	b13Output4ON	Bool	false	FALSE
	b14TareActive	Bool	false	FALSE
	b15Reserved	Bool	false	FALSE
	Gross	DInt	0	252
	Net	DInt	0	252
	ADPoints	DInt	0	39528
	CommandResult	Word	16#0	16#0000

Status word is decoded by bit.

4.1.4.1.2 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.

CAL	"TeNod4TCAL"		
MeasureRange	DInt	0	4000
CalibSegment	Int	0	2
CalibSegment1	DInt	0	200
CalibSegment2	DInt	0	300
CalibSegment3	DInt	0	30000
SensorSens	DInt	0	200000
StepMes	Int	0	2
CalibZero	DInt	0	-2441
CalibCoeff1	Real	0.0	16.702
CalibCoeff2	Real	0.0	16.702
CalibCoeff3	Real	0.0	16.702
CalibSpanCorr	DInt	0	1000000
gValueCalib	DInt	0	9805470
gValueLocal	DInt	0	9805470
ZeroOffset	DInt	0	0
FixedTare	DInt	0	0
SensorRef	DInt	0	0
SensorTol	Int	0	30
FILTER	"TeNod4TFILTER"		
LEGAL	"TeNod4TLEGAL"		
IOCFG	"TeNod4TIOCFG"		

CAL 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 register 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.

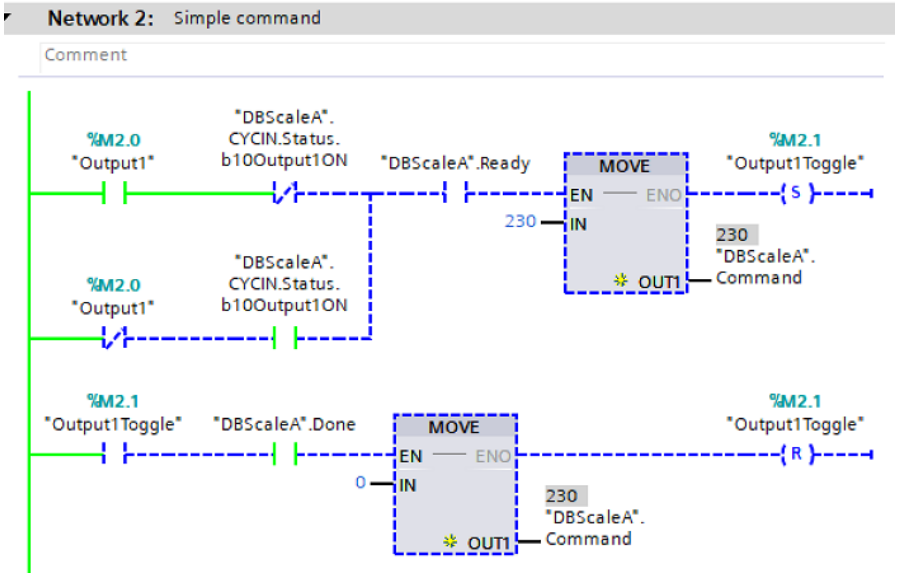
4.1.4.1.3 Bloc parameters and commands

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.



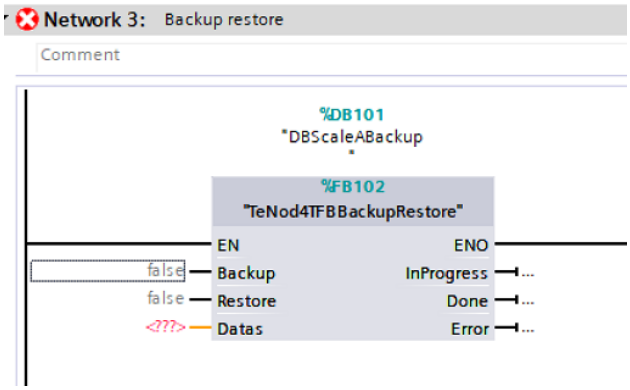
Not, this will work only if the output 1 is defined as “remote” on the scale configuration.

4.2 "TeNod4TFBBackupRestore" to restore and backup configuration

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

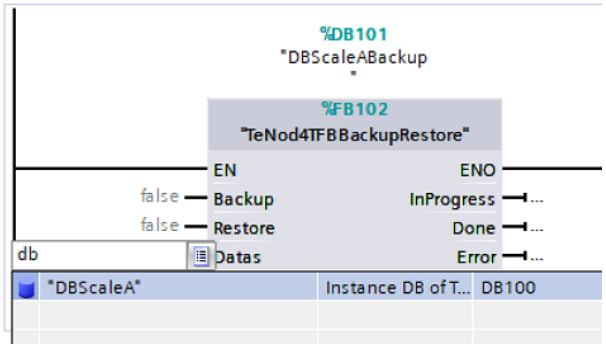
4.2.1 Creating the instance

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



4.2.2 Select main instance DB of the eNod.

This DB must be the one created with the main FB bloc.



4.2.3 Using the instruction

4.2.3.1.1 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.

4.2.3.1.2 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.

4.2.3.1.3 Restoring the configuration

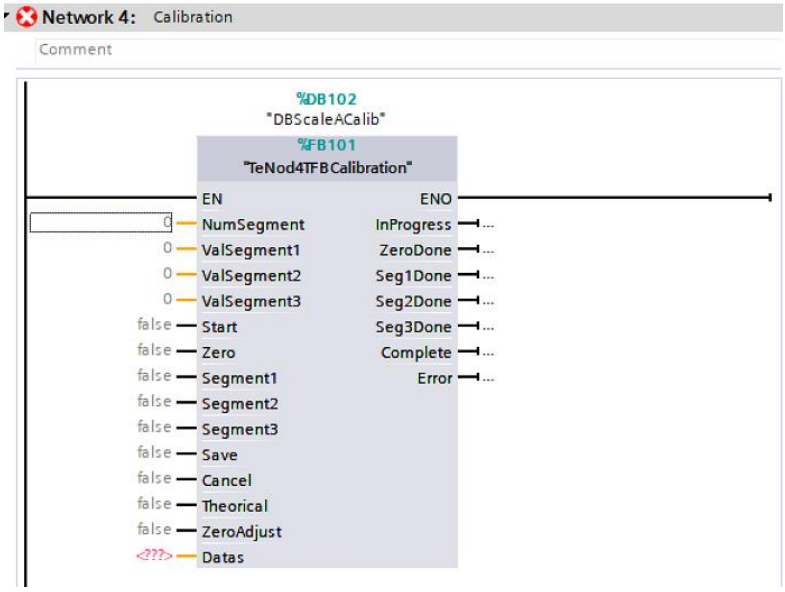
To restore the configuration, just set to 1 the input “Restore” of the bloc. This will send to the scale all configuration data:
CAL for calibration data. The last saved calibration will be sent, so the scale will be calibrated with the last calibration data after the restore
FIL for filter data
IOCFG 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.

4.3 “TeNod4TFBCalibration” to calibrate a scale

This bloc allows to calibrate the scale.

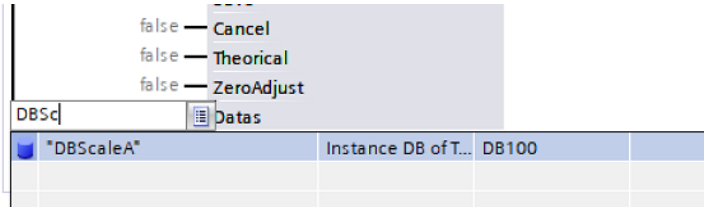
4.3.1 Creating the instance

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



4.3.2 Select main instance DB of the eNod.

This DB must be the one created with the main FB bloc.



4.3.3 Using the instruction

4.3.3.1.1 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
ZeroAdjust, Save
Theorical, Save

4.3.3.1.2 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 TIA portal writing the main instance DB.

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.

Input			
NumSegment	Int	0	1
ValSegment1	DInt	0	300
ValSegment2	DInt	0	0
ValSegment3	DInt	0	0
Start	Bool	false	TRUE
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	FALSE
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

Physically empty the scale and send the “Zero” 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	TRUE
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	FALSE
Output			
InProgress	Bool	false	TRUE
ZeroDone	Bool	false	TRUE
Seg1Done	Bool	false	FALSE
Seg2Done	Bool	false	FALSE
Seg3Done	Bool	false	FALSE
Complete	Bool	false	FALSE
Error	Bool	false	FALSE

Put on the scale the standard load you need according the “ValSegment1” value and then send “Segment1” 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	TRUE
Segment2	Bool	false	FALSE
Segment3	Bool	false	FALSE
Save	Bool	false	FALSE
Cancel	Bool	false	FALSE
Theorical	Bool	false	FALSE
ZeroAdjust	Bool	false	FALSE
Output			
InProgress	Bool	false	TRUE
ZeroDone	Bool	false	TRUE
Seg1Done	Bool	false	TRUE
Seg2Done	Bool	false	FALSE
Seg3Done	Bool	false	FALSE
Complete	Bool	false	FALSE
Error	Bool	false	FALSE

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

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

4.3.3.1.3 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 “ZeroAdjust” 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

4.3.3.1.4 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 TIA portal writing the main instance DB.

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

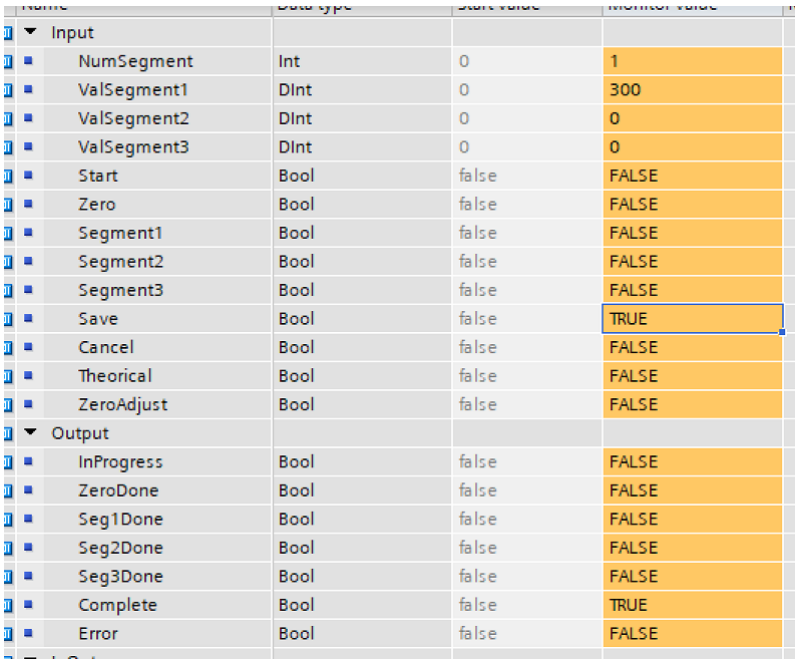
Start the procedure with “Start” command.

Input			
NumSegment	Int	0	1
ValSegment1	DInt	0	300
ValSegment2	DInt	0	0
ValSegment3	DInt	0	0
Start	Bool	false	TRUE
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	FALSE
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 the “Segment1” command

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

Send the “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

4.3.3.1.5 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 TIA portal writing the main instance DB.

Send “Theorical” 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	TRUE
ZeroAdjust	Bool	false	FALSE
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

4.3.3.1.6 Cancel calibration

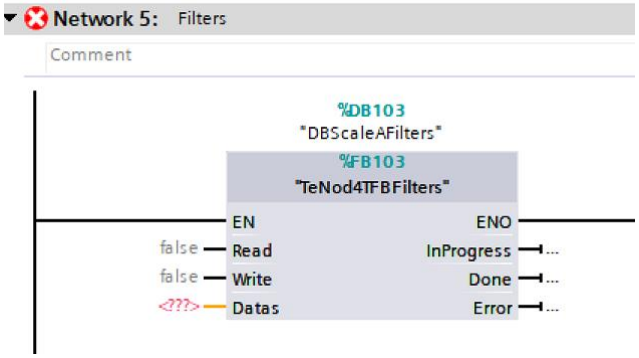
Calibration can be cancelled anytime until the save command is sent. “Cancel” command will reset the eNod and calibration will return to the last calibration values.

4.4 "TeNod4TFBFilters" to read and write filter configuration

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

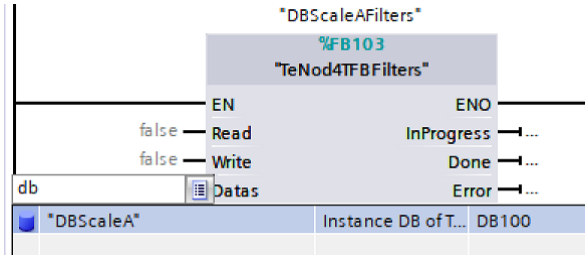
4.4.1 Creating the instance

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



4.4.2 Select main instance DB of the eNod.

This DB must be the one created with the main FB bloc.



4.4.3 Using the instruction

4.4.3.1.1 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.

4.4.3.1.2 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.

4.4.3.1.3 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.

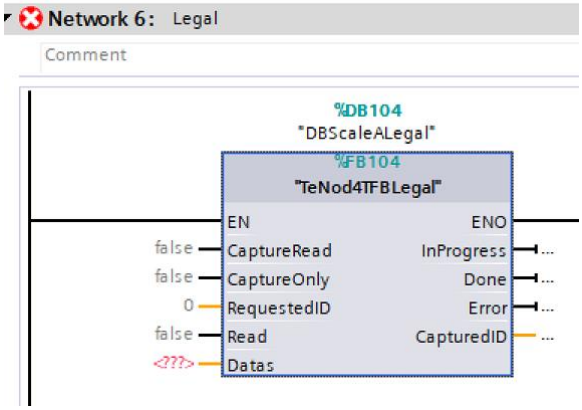
Note that if the A/D parameters values are changed, the changed will be used on the next reboot or reset of the eNod

4.5 "TeNod4TFBLegal" to manage DSD legal values

This bloc allows to create DSD values and read them

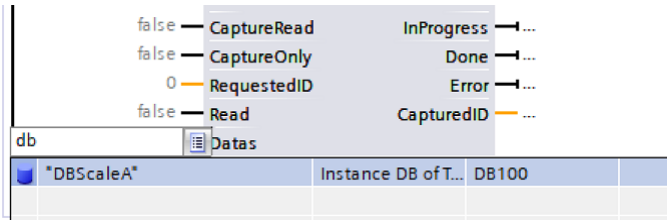
4.5.1 Creating the instance

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



4.5.2 Select main instance DB of the eNod.

This DB must be the one created with the main FB bloc.



4.5.3 Using the instruction

4.5.3.1.1 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 read later. “CaptureOnly” is faster than “CaptureRead” command dans do not erase the DSDRead data.
“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:

LEGAL	"TeNod4TLEGAL"		
eNodVersion	Int	0	0
ModbusCfg	Int	0	0
LegalVersion	Byte	16#0	16#02
LegalSwitch	Byte	16#0	16#00
LegalCount	Byte	16#0	16#00
LegalAlign	Byte	16#0	16#00
LegalChecksum	Int	0	0
ZeroConfig	Int	0	0
Stability	Byte	16#0	16#03
Decimal	Byte	16#0	16#02
Unit	Array[1..4] of Char		
DSDLastID	DInt	0	7
DSDOldestID	DInt	0	1
DSDCurrentID	DInt	0	7
DSDReadID	DInt	0	7
DSDReadNet	DInt	0	300
DSDReadTare	DInt	0	0
DSDReadStatus	Int	0	3202
DSDReadCheckSum	Int	0	-3509

4.5.3.1.2 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:

DBScaleALegal

Name	Data type	Start value	Monitor value
Input			
CaptureRead	Bool	false	TRUE
CaptureOnly	Bool	false	FALSE
RequestedID	DInt	0	0
Read	Bool	false	FALSE
Output			
InProgress	Bool	false	FALSE
Done	Bool	false	TRUE
Error	Bool	false	FALSE
CapturedID	DInt	0	8

The last ID is also on the structure on the field

DBScaleA

Name	Data type	Start value	Monitor value
LEGAL	"TeNod4TLEGAL"		
eNodVersion	Int	0	0
ModbusCfg	Int	0	0
LegalVersion	Byte	16#0	16#02
LegalSwitch	Byte	16#0	16#00
LegalCount	Byte	16#0	16#00
LegalAlign	Byte	16#0	16#00
LegalChecksum	Int	0	0
ZeroConfig	Int	0	0
Stability	Byte	16#0	16#03
Decimal	Byte	16#0	16#02
Unit	Array[1..4] of Char		
DSDLastID	DInt	0	8
DSDOldestID	DInt	0	1
DSDCurrentID	DInt	0	8

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

4.5.3.1.3 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:

DBScaleALegal

Name	Data type	Start value	Monitor value
Input			
CaptureRead	Bool	false	FALSE
CaptureOnly	Bool	false	FALSE
RequestedID	DInt	0	10
Read	Bool	false	TRUE
Output			
InProgress	Bool	false	FALSE
Done	Bool	false	FALSE
Error	Bool	false	TRUE
CapturedID	DInt	0	8

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

DSDReadID	DInt	0	8
DSDReadNet	DInt	0	350
DSDReadTare	DInt	0	0
DSDReadStatus	Int	0	3202
DSDReadCheckSum	Int	0	-3560

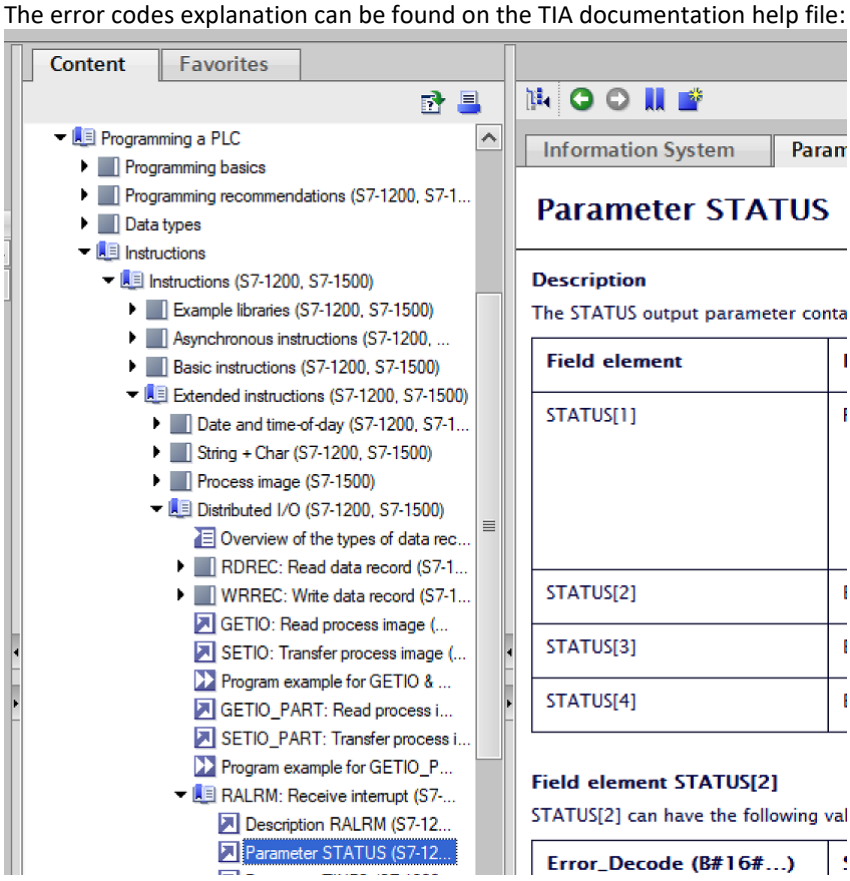
5 COMMUNICATION ERROR CODES

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 “**INTER.MSGError**” of the main instance DB is set.
In this case check the “**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 field “**MSGDataRecord**” to check what it the register that cannot be accessed. Go to chapter “15 Profinet IO” of Scaime documentation.

For example, data record 50 (0x32) is the “Low pass cut-off frequency”.

If the communication cannot be done, the field “**MSGErrorCode**” contains the error code returned by Status field of TIA functions RDREC or WRREC.
The error codes explanation can be found on the TIA documentation help file:



6 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.

First you need to configure the scale on the Ethernet network as describe on §2-A.

Then create tags on input data according the input address you have used: