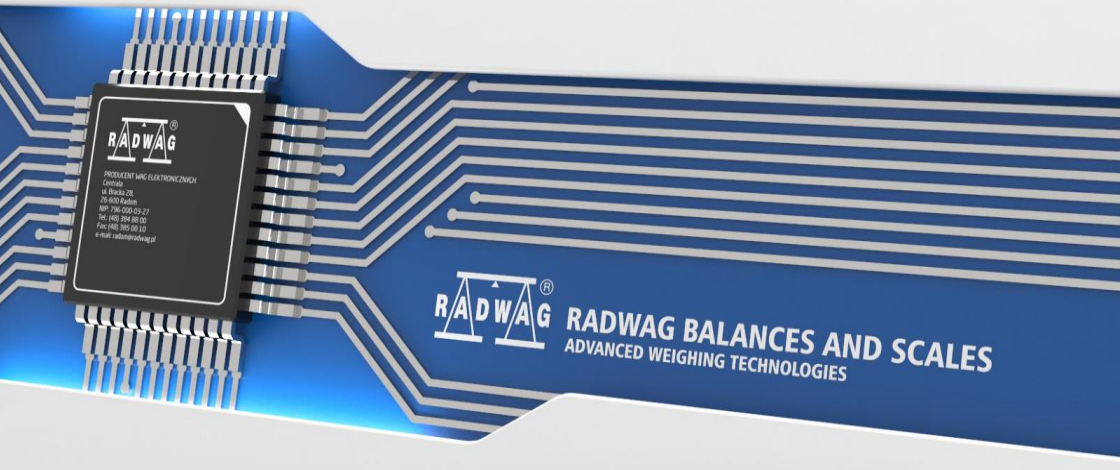
PROFINET

Communication protocol:
PUE HX5.EX Weighing Indicator

SOFTWARE MANUAL

ITKP-02-04-04-23-EN



APRIL 2023

CONTENTS

1. WEIGHING INDICATOR SETTINGS - CONFIGURATION 4
2. DATA STRUCTURE 4
2.1. Input Address 4
2.2. Input Registers 5
2.3. Output Address 7
2.4. Output Registers 7
3. CONFIGURATION OF PROFINET MODULE IN TIA PORTAL V14 11
3.1. GSD Import 11
3.2. Module Configuration 13
4. PLC SOFTWARE SAMPLE 18

1. WEIGHING INDICATOR SETTINGS - CONFIGURATION

To set indicator communication via PROFINET protocol go to **<SETUP / IM01.EX communication module / Additional modules / Anybus module>**. For detailed description of settings configuration read **„PUE HX5.EX - Software manual”** user manual.

2. DATA STRUCTURE

2.1. Input Address

Input variables list:

Variable	Offset	Length [WORD]	Data type
Mass	0	2	float
Tare	4	2	float
Unit	8	1	word
Platform status	10	1	word
LO threshold	12	2	float
Process status (Stop, Start)	16	1	word
Inputs status	66	1	word
Min	68	2	float
Max	72	2	float
Fast dosing threshold	76	2	float
Slow dosing threshold	80	2	float
Lot number	84	2	dword
Operator	88	1	word
Product	90	1	word
Customer	92	1	word
Packaging	94	1	word
Formulation	100	1	word
Dosing process	102	1	word

2.2. Input Registers

Platform mass – returns platform mass in current unit.

Platform tare – returns platform tare in an adjustment unit.

Platform unit – determines current mass unit of a given platform.

Unit bits	
0	gram [g]
1	kilogram [kg]
2	carat [ct]
3	pound [lb]
4	ounce [oz]
5	Newton [N]

Example:

Read HEX value: 0x02. Binary form:

B1/7	B1/6	B1/5	B1/4	B1/3	B1/2	B1/1	B1/0	B0/7	B0/6	B0/5	B0/4	B0/3	B0/2	B0/1	B0/0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

The unit of the weighing instrument is kilogram [kg].

Platform status – determines state of a given weighing platform.

Status bits	
0	Measurement correct (the weighing instrument does not report any error)
1	Measurement stable
2	Weighing instrument indicates zero
3	Weighing instrument tared
4	Weighing instrument in II weighing range
5	Weighing instrument in III weighing range
6	Weighing instrument reports NULL error
7	Weighing instrument reports LH error
8	Weighing instrument reports FULL error

Example:

Read HEX value: 0x13

B1/7	B1/6	B1/5	B1/4	B1/3	B1/2	B1/1	B1/0	B0/7	B0/6	B0/5	B0/4	B0/3	B0/2	B0/1	B0/0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1

The weighing instrument does not report any error, measurement stable in II weighing range.

LO threshold – returns value of platform's **LO** threshold in an adjustment unit.

Process status – determines dosing or formulations process status:

0x00 – process disabled
0x01 – process stopped
0x02 – process aborted
0x03 – process finished

Input state – bitmask of indicator inputs. The first 4 least significant bits represent weighing terminal inputs.

Example:

Read HEX value: 0x000B

B1/7	B1/6	B1/5	B1/4	B1/3	B1/2	B1/1	B1/0	B0/7	B0/6	B0/5	B0/4	B0/3	B0/2	B0/1	B0/0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1

Inputs 1, 2 and 3 of the weighing indicator take HI state.

MIN - response: **MIN** threshold value (in current unit).

MAX - response: **MAX** threshold value (in current unit).

Fast dosing threshold - returns fast dosing threshold value in an adjustment unit.

Slow dosing threshold - returns slow dosing threshold value in an adjustment unit.

Lot number – response: lot number. Only numerical values are accepted! All other characters are skipped.

Operator – response: code of logged in operator.

Product – response: code of selected product.

Customer – response: code of selected customer.

Packaging – response: code of selected packaging.

Formulation – response: code of selected formulation.

Dosing process – response: code of selected dosing process.

2.3. Output Address

Input variables:

Variable	Offset	Length [WORD]	Data type
Command	0	1	word
Command with parameter	2	1	word
Platform	4	1	word
Tare	6	2	float
LO threshold	10	2	float
Output state	14	1	word
Min	16	2	float
Max	20	2	float
Lot number	32	2	dword
Operator	36	1	word
Product	38	1	word
Customer	40	1	word
Packaging	42	1	word
Formulation	48	1	word
Dosing process	50	1	word

2.4. Output Registers

Basic command – writing the register with respective value triggers the following actions:

Bit No.	Operation
0	Zero the platform
1	Tare the platform
2	Delete statistics
3	Save/Print
4	Start

5	Breakdown (STOP without confirmation)
6	Tare/Zero the platform
7	Lock keypad
8	Unlock keypad

A command is executed once upon detecting that its bit has been set. If the command is to be executed more than once, it is necessary to zero the bit first, and reset it to the required value next.

Example:

Writing the register with value 0x02

B1/7	B1/6	B1/5	B1/4	B1/3	B1/2	B1/1	B1/0	B0/7	B0/6	B0/5	B0/4	B0/3	B0/2	B0/1	B0/0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

This causes scale taring.

Complex command – setting a respective value results with performance of a given task, see the table:

Bit No.	Operation
0	Setting tare value for a given platform
1	Setting LO threshold value for a given platform
2	Setting lot number
3	Setting outputs status
4	Operator selection
5	Product selection
6	Packaging selection
7	Setting MIN threshold value
8	Customer selection
11	Dosing process selection
12	Setting MAX threshold value

Complex command requires setting a respective parameter (offset from 4 to 50 – refer to output registers table)

A command with a parameter is executed once upon detecting that its bit has been set. If the command is to be executed more than once, it is necessary to zero the bit first, and reset it to the required value next.

Example:

Sending tare of 1.0 value for the 1st platform.

Carrying out the command requires writing 3 registers:

offset 2 – command with parameter - value 0x01 – i.e. tare setting.

offset 4 – number of a weighing platform to which the tare is to be written - 0x01 value for the 1st first platform.

offset 6 – tare value in float format - 1.0.

Platform – complex command parameter: weighing platform number.

Tare – complex command parameter: tare value (in an adjustment unit).

LO threshold – complex command parameter: LO threshold value (in an adjustment unit).

Output state – complex command parameter: determines state of the weighing indicator and communication module outputs.

Example:

Setting high state to outputs 1 and 3 of the weighing indicator.

Output mask has the following format:

B1/7	B1/6	B1/5	B1/4	B1/3	B1/2	B1/1	B1/0	B0/7	B0/6	B0/5	B0/4	B0/3	B0/2	B0/1	B0/0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

After conversion to HEX the result is 0x05

Carrying out the command requires writing 2 registers:

offset 2 – command with parameter - value 0x08 – i.e. record of outputs state.

offset 14 – outputs mask 0x05

As a result, outputs number 1 and 3 take high state.

MIN – complex command parameter: MIN threshold value (in a unit of current working mode).

MAX – complex command parameter: MAX threshold value (in a unit of current working mode).

Lot number – complex command parameter: lot number value. Only numerical values are accepted! All other characters are skipped.

Operator – complex command parameter: operator code (digits only).

Product – complex command parameter: product code (digits only).

Customer – complex command parameter: customer code (digits only).

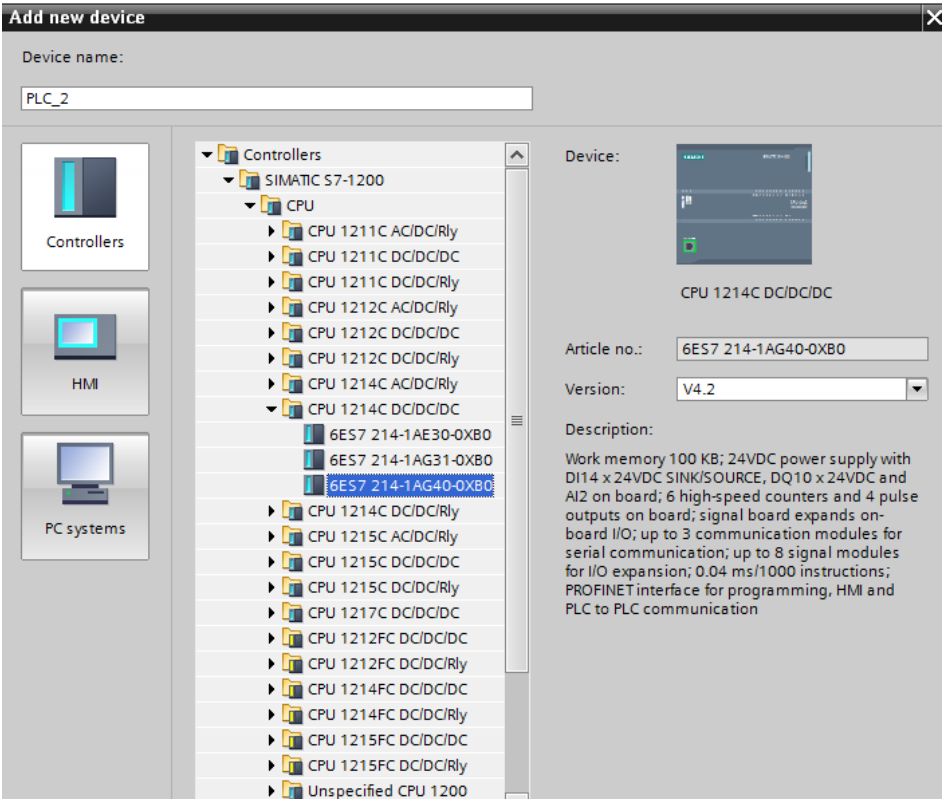
Packaging – complex command parameter: packaging code (digits only)

Formulation – complex command parameter: formulation code (digits only).

Dosing process – complex command parameter: dosing process code (digits only).

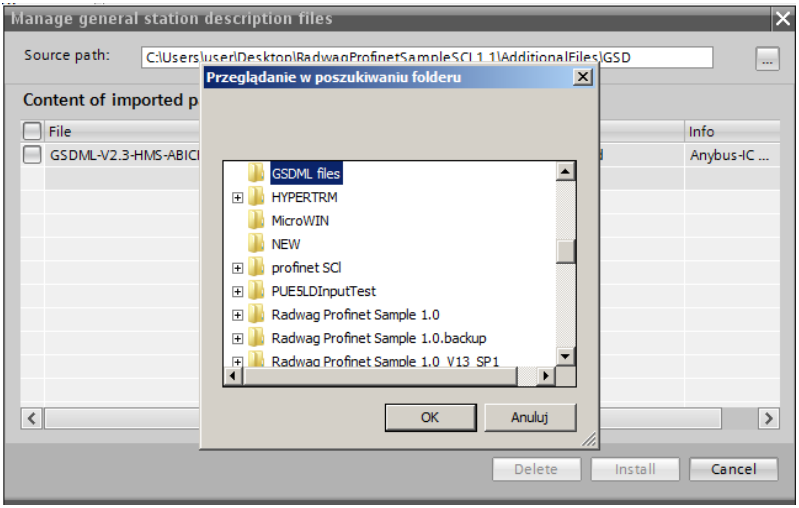
3. CONFIGURATION OF PROFINET MODULE IN TIA PORTAL V14

Operating the environment has to be preceded with creating a new project in which the topology of the PROFINET network with MASTER PLC is determined (in this example: SIEMENS S7-1200).

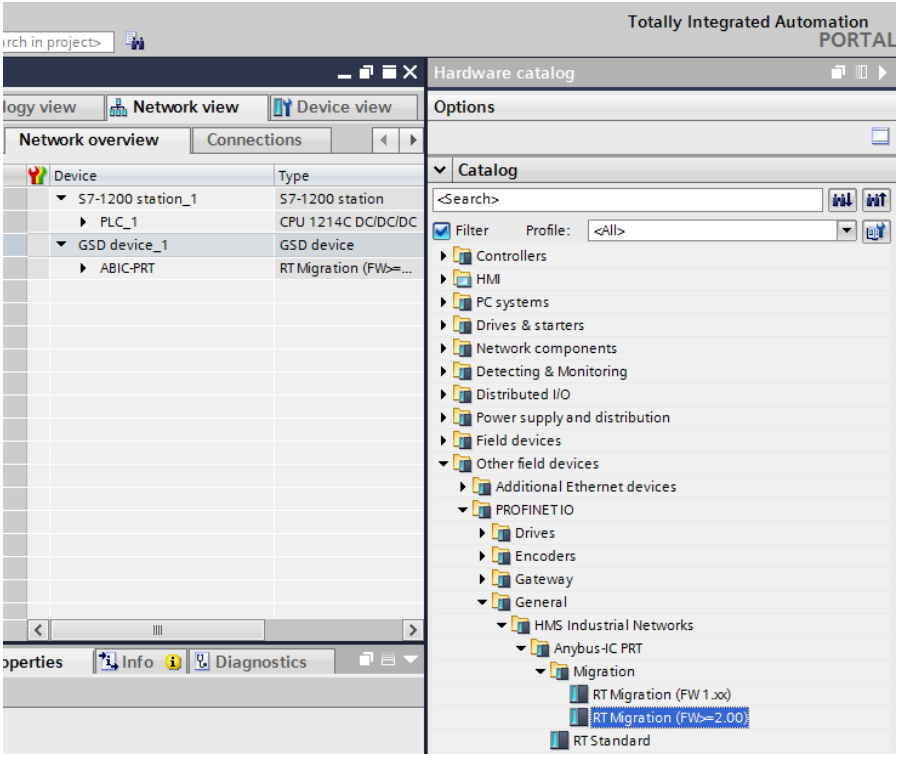


3.1. GSD Import

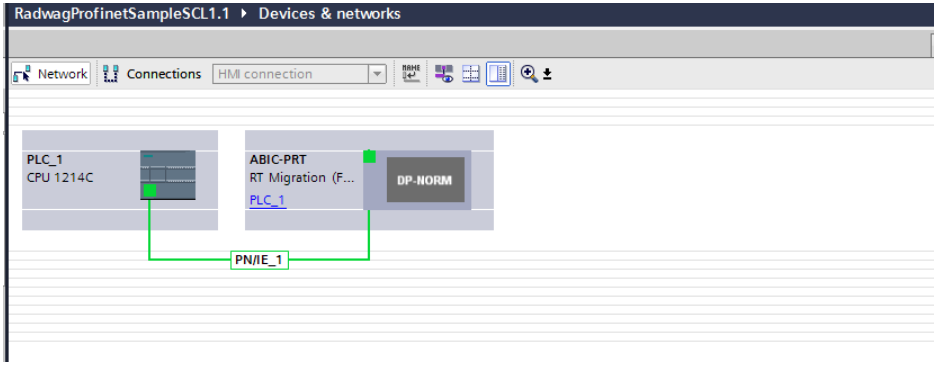
Using the included GSD configuration file add new device to the environment. Use OPTIONS tab first, MANAGE GENERAL STATION DESCRIPTION FILES (GSD) next and indicate the path to GSD file.



Upon successful adding of the file using list of devices, find ABIC-PRT module:

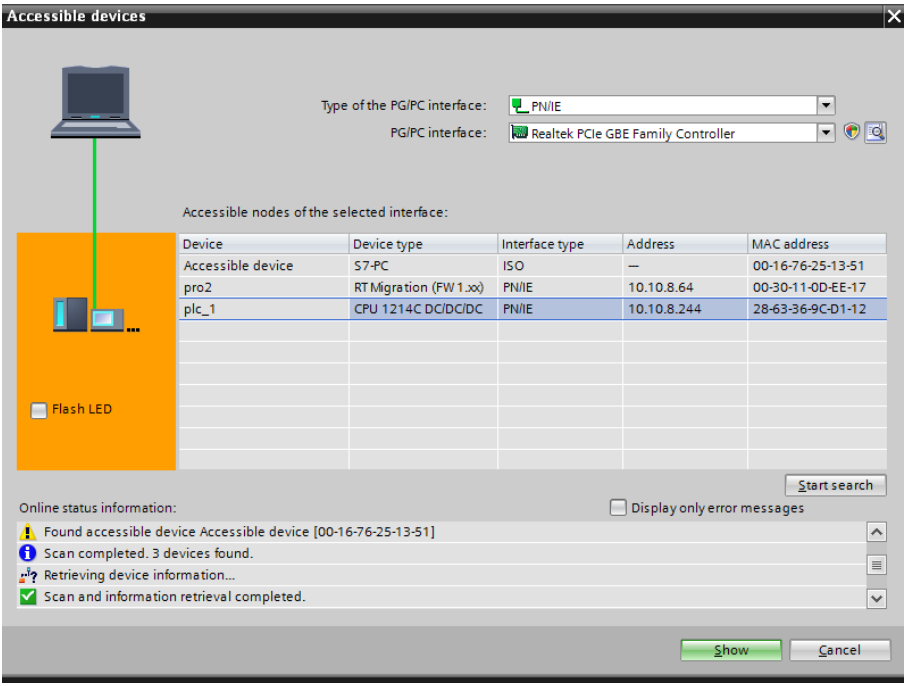


You can now create a network consisting of one MASTER PLC and added SLAVE module:

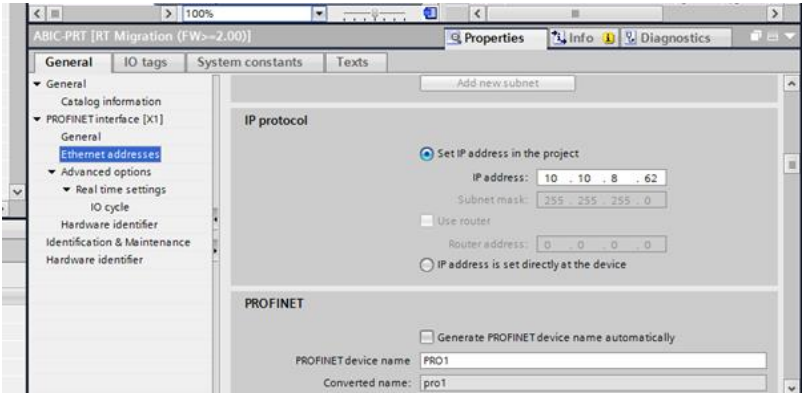


3.2. Module Configuration

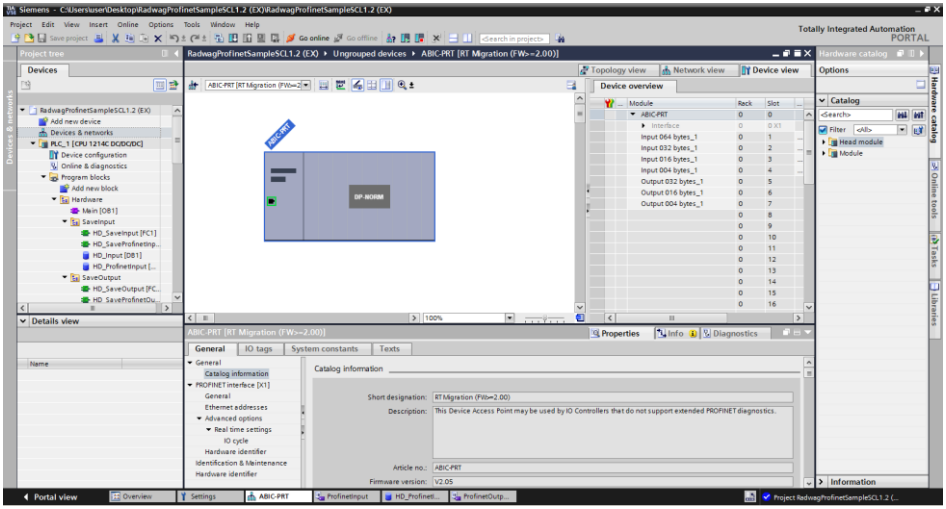
At this stage, create a network consisting of MASTER device and SLAVE device (weighing instrument). Upon connecting the power supply, search for device using ACCESSIBLE DEVICES function. The list should contain MASTER and SLAVE devices:

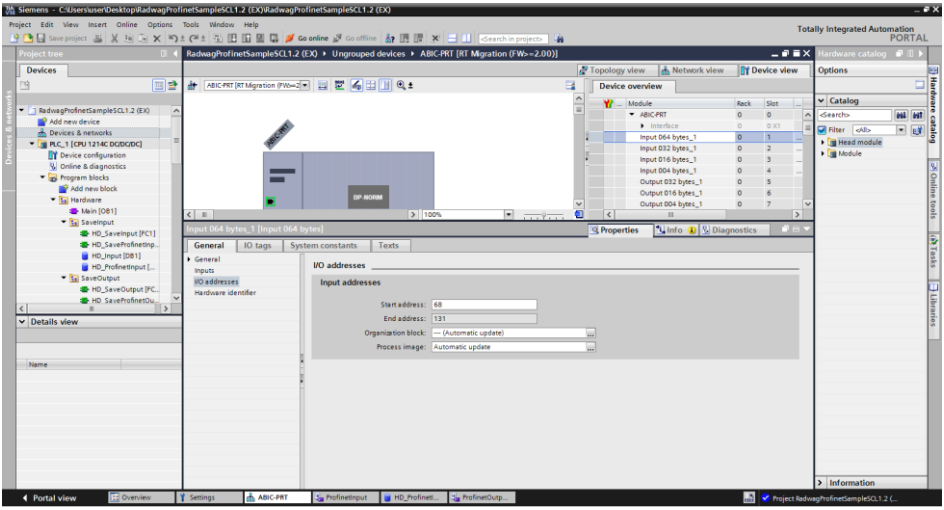


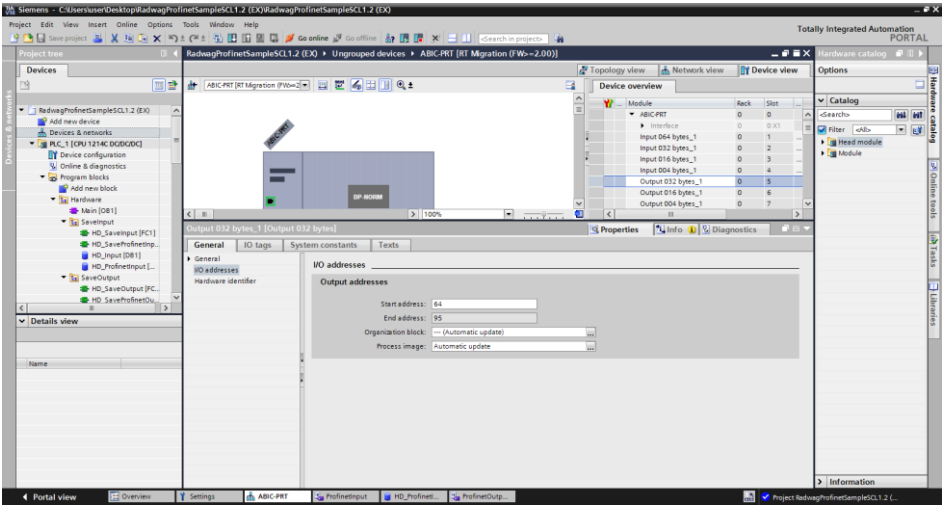
Next, specify the IP address of the module and its name in PROFINET network. Upon selecting the module in PROPERTIES tab, find PROFINET INTERFACE and enter IP address and name. Those settings have to be the same as the ones set in the weighing instrument menu. IP SLAVE address has to be part of the same subnet as the MASTER address.



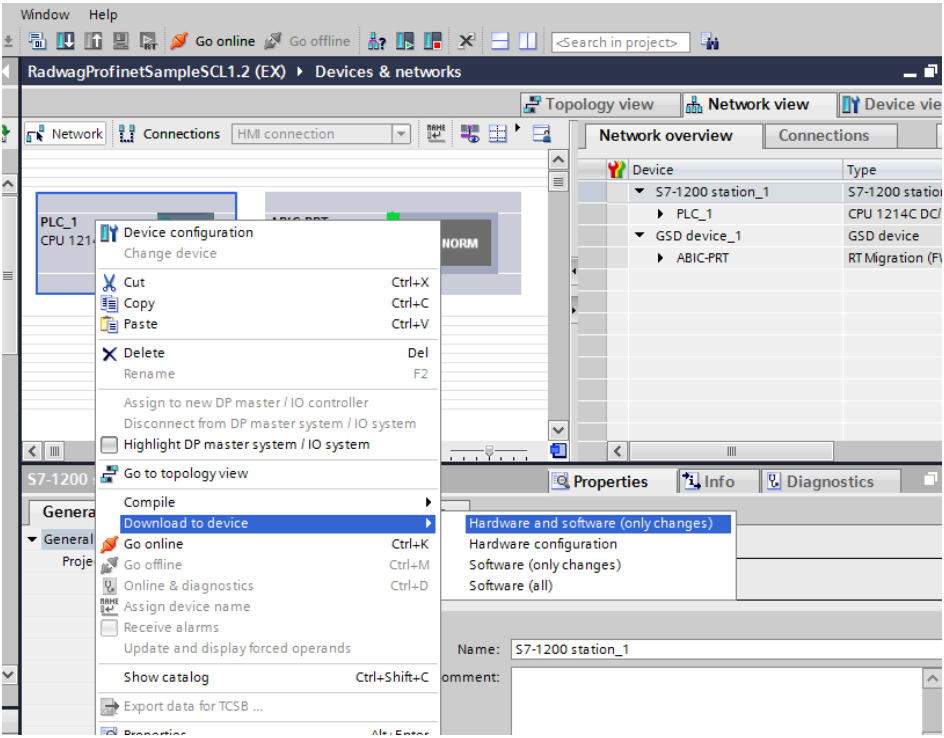
Proceed to module configuration. Start by determining the size and the starting address of input and output registers. To do this, select modules from the list of INPUT and OUTPUT modules as in the picture below. The maximum size of the input and output data is 116 bytes each. Default starting addresses were used in the project - 68 for INPUT module and 64 for OUTPUT module:



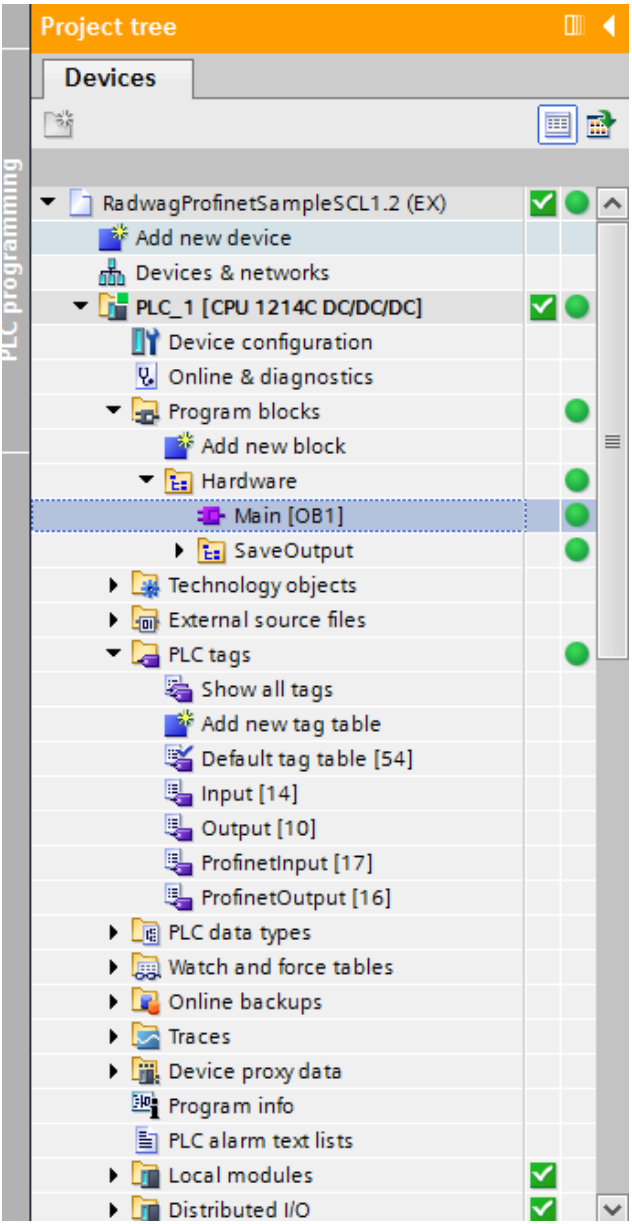

Siemens – C:\Users\user\Desktop\RadwagProfinetSampleSCL1.2 (EX)\RadwagProfinetSampleSCL1.2 (EX)
Totally Integrated Automation
PORTAL
RadwagProfinetSampleSCL1.2 (EX) ▸ Ungrouped devices ▸ ABIC-PRT [RT Migration (FW>=2.00)]
Input 064 bytes_1 [Input 064 bytes]
General
IO tags
System constants
Texts
I/O addresses
Input addresses
Start address: 68
End address: 131
Organization block: --- (Automatic update)
Process image: Automatic update


Siemens – C:\Users\user\Desktop\RadwagProfinetSampleSCL1.2 (EX)\RadwagProfinetSampleSCL1.2 (EX)
Totally Integrated Automation
PORTAL
RadwagProfinetSampleSCL1.2 (EX) ▸ Ungrouped devices ▸ ABIC-PRT [RT Migration (FW>=2.00)]
Output 032 bytes_1 [Output 032 bytes]
General
IO tags
System constants
Texts
I/O addresses
Output addresses
Start address: 64
End address: 95
Organization block: --- (Automatic update)
Process image: Automatic update

On this stage you can download hardware and software configuration to the device and download data to the device.



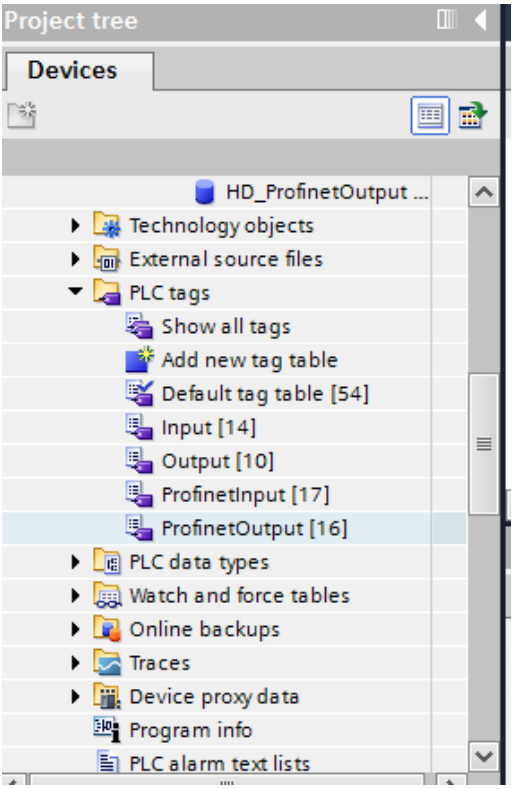
Upon successful compilation and loading of the code, MASTER and SLAVE modules should establish communication. You can check this by pressing GO ONLINE field. The result has to be similar to the result presented below.



The next step will be to create program code.

4. PLC SOFTWARE SAMPLE

Start creating the application by determining symbolic names of input and output registers. For this purpose, use the branch of the project tree: PLC TAGS. For the purpose of this example, the figure below contains exemplary tags tables:



INPUT and OUTPUT tables refer to the physical inputs/outputs of the MASTER device and are not relevant in terms of this application. The PROFINET module input and output registers are specified in ProfinetInput and ProfinetOutput tables.

The pictures below present determined symbolic names and addresses:

RadwagProfinetSampleSCL1.2 ang(EX) ▸ PLC_1 [CPU 1214C DC/DC/DC] ▸ PLC tags ▸ ProfinetInput [17]

ProfinetInput

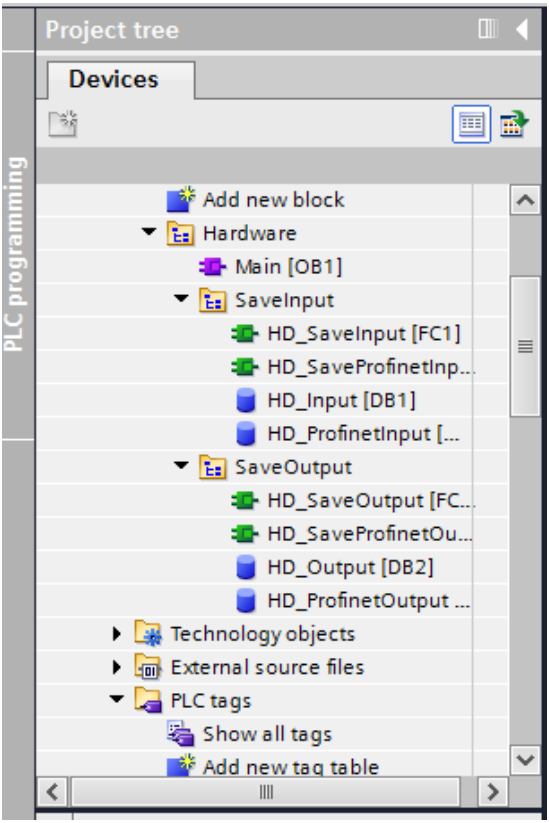
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Comment
1	mass	Real	%ID68		☑	☑	☑	
2	tare	Real	%ID72		☑	☑	☑	
3	unit	Word	%IW76		☑	☑	☑	
4	status	Word	%IW78		☑	☑	☑	
5	LO	Real	%ID80		☑	☑	☑	
6	process status	Word	%IW84		☑	☑	☑	
7	inputs	Word	%IW134		☑	☑	☑	
8	min	Real	%ID136		☑	☑	☑	
9	max	Real	%ID140		☑	☑	☑	
10	lot number	DWord	%ID152		☑	☑	☑	
11	operator	Word	%IW156		☑	☑	☑	
12	article	Word	%IW158		☑	☑	☑	
13	customer	Word	%IW160		☑	☑	☑	
14	packaging	Word	%IW162		☑	☑	☑	
15	source warehuse	Word	%IW164		☑	☑	☑	
16	target warehouse	Word	%IW166		☑	☑	☑	
17	formulation/dosing process	Word	%IW168		☑	☑	☑	
18	<Add new>				☑	☑	☑	

RadwagProfinetSampleSCL1.2 ang(EX) ▸ PLC_1 [CPU 1214C DC/DC/DC] ▸ PLC tags ▸ ProfinetOutput [16]

ProfinetOutput

	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Comment
1	cammand	Word	%QW64		☑	☑	☑	
2	complex command	Word	%QW66		☑	☑	☑	
3	platform	Word	%QW68		☑	☑	☑	
4	set tare	Real	%QD70		☑	☑	☑	
5	set lo	Real	%QD74		☑	☑	☑	
6	outputs	Word	%QW78		☑	☑	☑	
7	set min	Real	%QD80		☑	☑	☑	
8	set max	Real	%QD84		☑	☑	☑	
9	set lot	DWord	%QD96		☑	☑	☑	
10	set operator	Word	%QW100		☑	☑	☑	
11	set article	Word	%QW102		☑	☑	☑	
12	set customer	Word	%QW104		☑	☑	☑	
13	set source warehouse	Word	%QW108		☑	☑	☑	
14	set target warehouse	Word	%QW110		☑	☑	☑	
15	set formulation/dosing	Word	%QW112		☑	☑	☑	
16	set packaging	Word	%QW106		☑	☑	☑	
17	<Add new>				☑	☑	☑	

In order to avoid working directly on module physical inputs/outputs, create data blocks containing the representations of those registers and create function 'rewriting' the values between them. Create HARDWARE group in PROGRAM BLOCKS branch and determine data blocks in the same way as presented below:



HD_OUTPUT and HD_INPUT blocks refer to physical MASTER inputs/outputs and are not relevant in terms of this project. HD_ProfinetOutput and HD_ProfinetInput blocks refer to the PROFINET module input/output registers on a weighing instrument. They look as follows:

RadwagProfinetSampleSCL1.2 ang(EX) ▸ PLC_1 [CPU 1214C DC/DC/DC] ▸ Program blocks ▸ Hardware ▸ SaveInput ▸ HD_ProfinetInput [DB3]

Keep actual values | Snapshot | Copy snapshots to start values | Load start values as actual values

HD_ProfinetInput

	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	Static								
2	mass	Real	0.0		☑	☑	☑		
3	tare	Real	0.0		☑	☑	☑		
4	unit	Word	16#0		☑	☑	☑		
5	status	Word	16#0		☑	☑	☑		
6	lo	Real	0.0		☑	☑	☑		
7	process status	Word	16#0		☑	☑	☑		
8	inputs	Word	16#0		☑	☑	☑		
9	min	Real	0.0		☑	☑	☑		
10	max	Real	0.0		☑	☑	☑		
11	lot number	DWord	16#0		☑	☑	☑		
12	operator	Word	16#0		☑	☑	☑		
13	article	Word	16#0		☑	☑	☑		
14	customer	Word	16#0		☑	☑	☑		
15	packaging	Word	16#0		☑	☑	☑		
16	source warehouse	Word	16#0		☑	☑	☑		
17	target warehouse	Word	16#0		☑	☑	☑		
18	formulation/dosing pr...	Word	16#0		☑	☑	☑		

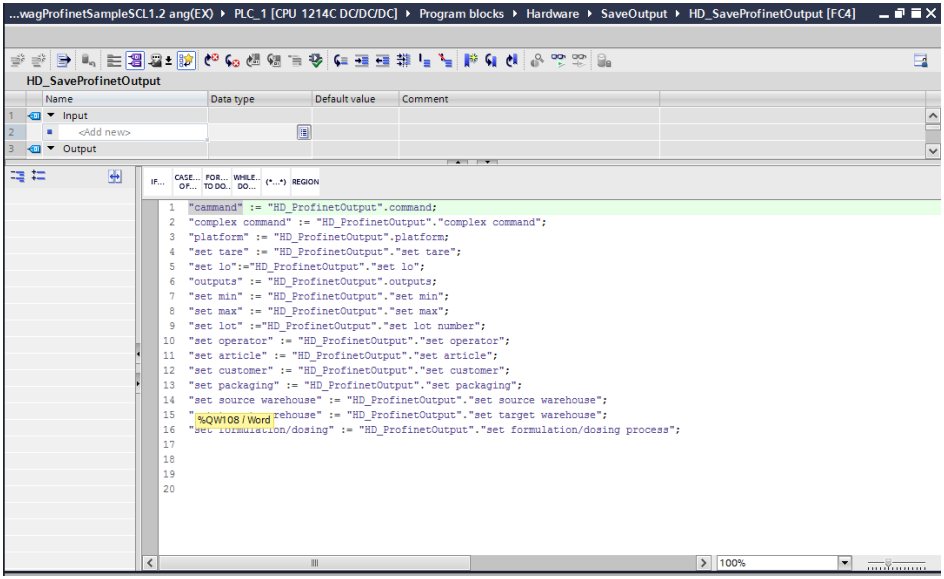
RadwagProfinetSampleSCL1.2 ang(EX) ▸ PLC_1 [CPU 1214C DC/DC/DC] ▸ Program blocks ▸ Hardware ▸ SaveOutput ▸ HD_ProfinetOutput [DB4]

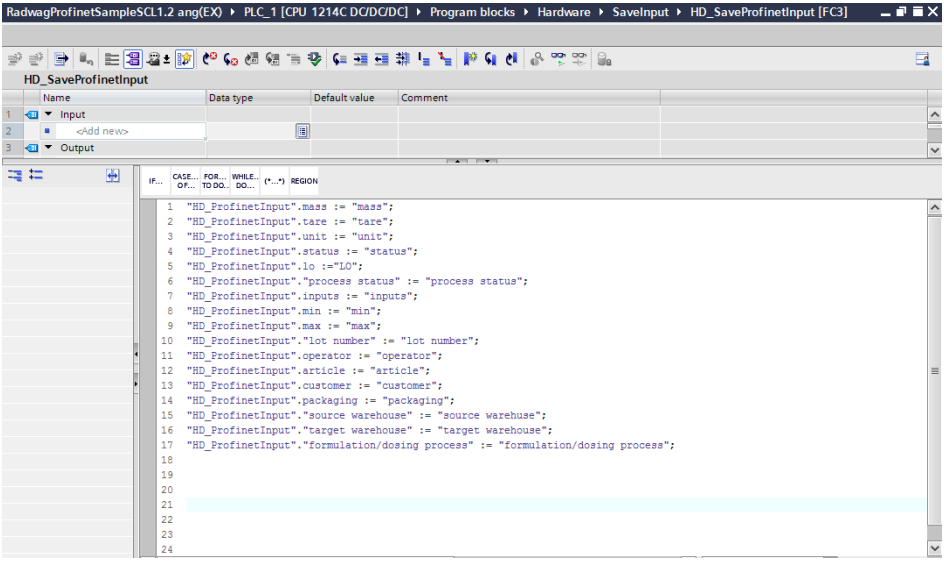
Keep actual values | Snapshot | Copy snapshots to start values | Load start values as actual values

HD_ProfinetOutput

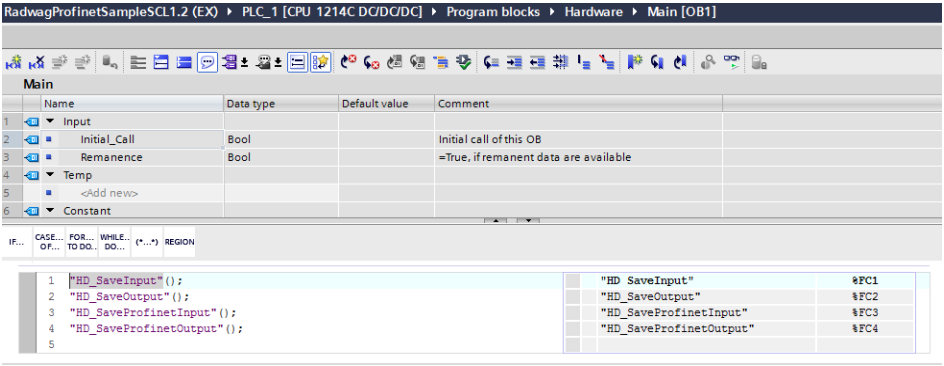
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	Static								
2	command	Word	16#02		☑	☑	☑		
3	complex command	Word	16#0008		☑	☑	☑	☑	
4	platform	Word	16#0001		☑	☑	☑		
5	set tare	Real	2.0		☑	☑	☑		
6	set lo	Real	1.5		☑	☑	☑		
7	outputs	Word	16#0000		☑	☑	☑	☑	
8	set min	Real	2.1		☑	☑	☑		
9	set max	Real	2.2		☑	☑	☑		
10	set lot number	DWord	16#0000		☑	☑	☑		
11	set operator	Word	16#0004		☑	☑	☑		
12	set article	Word	16#0001		☑	☑	☑		
13	set customer	Word	16#01		☑	☑	☑		
14	set packaging	Word	16#0004		☑	☑	☑		
15	set source warehouse	Word	16#0		☑	☑	☑		
16	set target warehouse	Word	16#0		☑	☑	☑		
17	set formulation/dosin...	Word	16#0		☑	☑	☑		

The functions that rewrite values between physical inputs/outputs of the module may look like this:





Invoke the functions in the main program loop.



Upon compiling and loading the program to the device in the data block you can read interesting output registers (MONITOR ALL) and save output registers (e.g. by changing START VALUE and LOAD START VALUES AS ACTUAL) of the SLAVE mode.

RADWAG BALANCES AND SCALES
ADVANCED WEIGHING TECHNOLOGIES