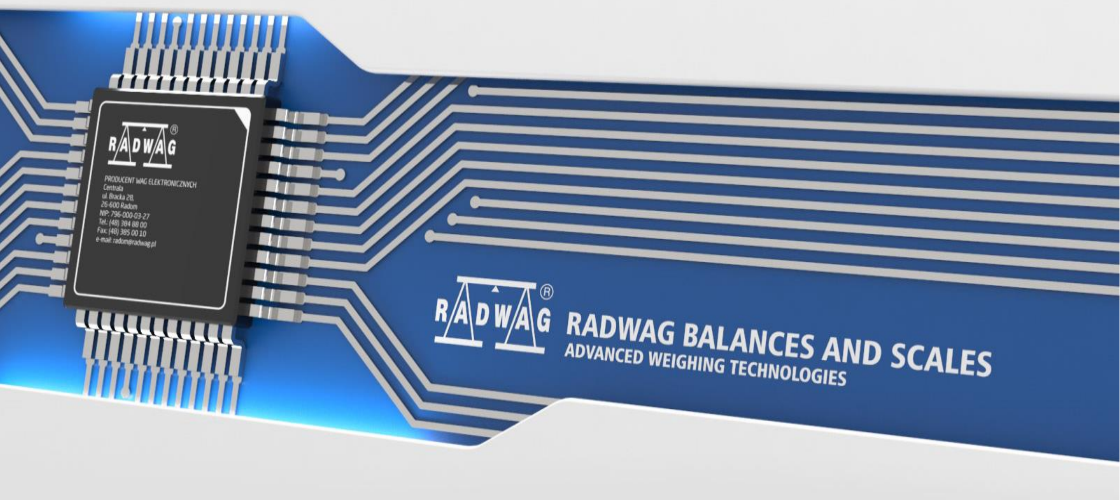
# PROFIBUS

**Communication Protocol:**
MW-01-A Mass Converter

SOFTWARE MANUAL

ITKP-42-01-11-21-EN



NOVEMBER 2021

# CONTENTS

# 1. MASS CONVERTER SETTINGS AND CONFIGURATION

To configure MW-01-A mass converter settings for communication via **Profibus** protocol run **“MwManager”** PC software and go to **<Parameters / Set Communication / Additional modules>** submenu. For detailed description of settings configuration read **“MwManager”** user manual.

# 2. DATA STRUCTURE

## 2.1. Input Address

**Input variables:**

| Variable | Offset | Length [WORD] | Data type |
|---|---|---|---|
| Platform mass | 0 | 2 | float |
| Platform tare | 4 | 2 | float |
| Platform unit | 8 | 1 | word |
| Platform status | 10 | 1 | word |
| Platform LO threshold | 12 | 2 | float |
| Process status | 64 | 1 | word |
| Input status | 66 | 1 | word |
| Min | 68 | 2 | float |
| Max | 72 | 2 | float |
| Fast dosing threshold | 76 | 2 | float |
| Slow dosing threshold | 80 | 1 | float |

## 2.2. Input Registers

Note that data uploaded from the MW-01-A mass converter have reverse order of register bytes, i.e. float data order is DCBA, and word variable order is BA. To correctly read these registers it is necessary to reorder them.

**Platform mass** – returns platform mass a in current unit.

**Example:**

Hex value of a register with offset 0 is 0x00001041. Prior to change to float, it is necessary to set the bytes in a reverse order, i.e. ABCD, which results with 0x41100000.

This, when changed to flow, gives 9.0 as a current mass of the load.

**Platform tare** – returns platform tare in an adjustment unit.

**Platform unit** – determines a 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: 0x0200. 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 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Upon reordering from BA to AB the result is 0x0002

| 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: 0x1300

| 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 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Upon reordering from BA to AB the result is 0x0013

| 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 **LO** threshold in an adjustment unit.

**Process status** – determines status of the dosing process.

0x00 – process disabled
0x01 – process activated
0x02 – process aborted
0x03 – process completed

**Input state** – bitmask of indicator inputs. The first 3 least significant bits represent weighing indicator inputs.

**Example:**

Read HEX value: 0x0300

| 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 | 1 |

Upon reordering from BA to AB the result is 0x0003

| 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 | 1 |

Inputs 1 and 2 of the mass converter take HI state.

**MIN** – returns **MIN** threshold value in an adjustment unit.

**MAX** – returns **MAX** threshold value in an adjustment 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.

### 2.3. Output Address

**Output variables:**

| Variable | Offset | Length [WORD] | Data type |
|---|---|---|---|
| Command | 0 | 1 | word |
| Command with parameter | 2 | 1 | word |
| Tare | 6 | 2 | float |
| LO threshold | 10 | 2 | float |
| Output state | 14 | 1 | word |
| Min | 16 | 2 | float |
| Max | 20 | 2 | float |
| Fast dosing threshold | 24 | 2 | float |
| Slow dosing threshold | 28 | 1 | float |

### 2.4. Output Registers

**Basic command** – record of the register via a given value triggers a respective operation:

| Bit No. | Operation |
|---|---|
| 0 | Zero the platform |
| 1 | Tare the platform |
| 5 | Process start |
| 6 | Process stop |

**Example:**

Record of the register by 0x02 value converted to BA 0x0200 order.

| 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 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Scale taring is triggered.

***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.***

**<u>Complex command</u>** – setting a respective value triggers 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 output status |
| 3 | Setting MIN threshold value |
| 4 | Setting MAX threshold value |
| 5 | Setting fast dosing threshold |
| 6 | Setting slow dosing threshold |

***Complex command requires setting a respective parameter (offset from 6 to 36 – refer to output registers table)***

***A command with a parameter is executed once when its bit setting is detected. 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.

Performance of the command requires record of 2 registers:

offset 2 – command with parameter - value 0x0100 – after conversion 0x0100.
offset 6 – tare value in float format - 1.0 after conversion to DCBA 0x0000803F format.

**<u>Tare</u>** – complex command parameter: tare value (in an adjustment unit).

**<u>LO threshold</u>** – complex command parameter: LO threshold value (in an adjustment unit).

**<u>Output state</u>** – complex command parameter: state of mass converter outputs.

**Example:**

Setting high state for output 1 and 3 of the indicator.

Output mask:

| 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 |

Upon conversion to HEX the result is 0x05.

Performance of the command requires record of 2 registers:

offset 2 – command with parameter - value 0x08 – i.e. output state record,

offset 14 – output mask 0x05.

This results with HI state of outputs number 1 and 3.

**MIN** – complex command parameter: MIN threshold value (in the unit set for the active working mode).

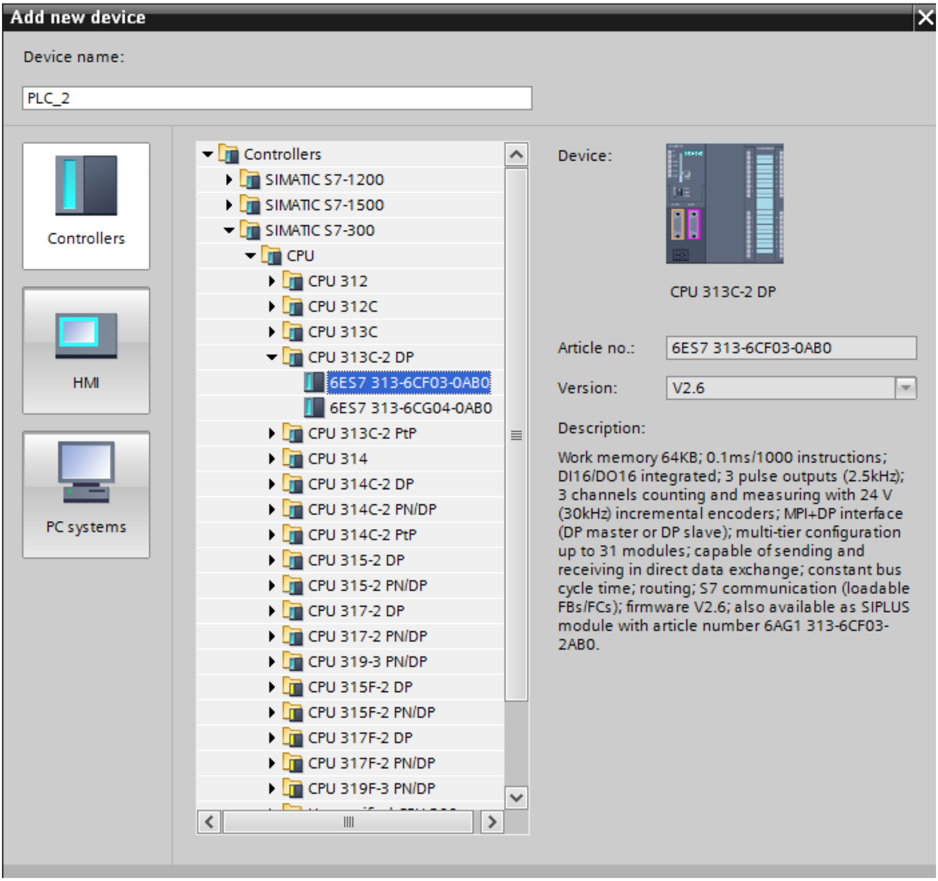
**MAX** – complex command parameter: MAX threshold value (in the unit set for the active working mode).

**Fast dosing threshold** - complex command parameter – fast dosing threshold value (in an adjustment unit).

**Slow dosing threshold** - complex command parameter – slow dosing threshold value (in an adjustment unit).

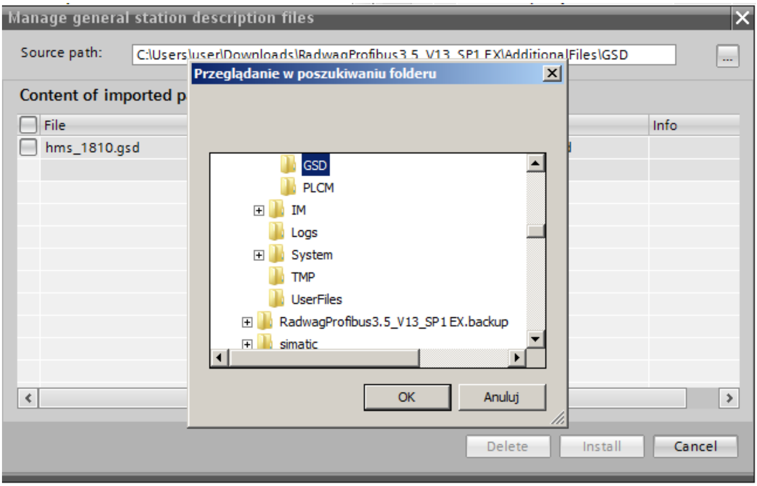
# 3. CONFIGURATION OF PROFIBUS MODULE IN TIA PORTAL V13 ENVIRONMENT

Start operation in the environment by setting up a new project, where topology of PROFIBUS network with MASTER controller will be specified, in this case the MASTER controller is SIEMENS controller of S7-300 series.



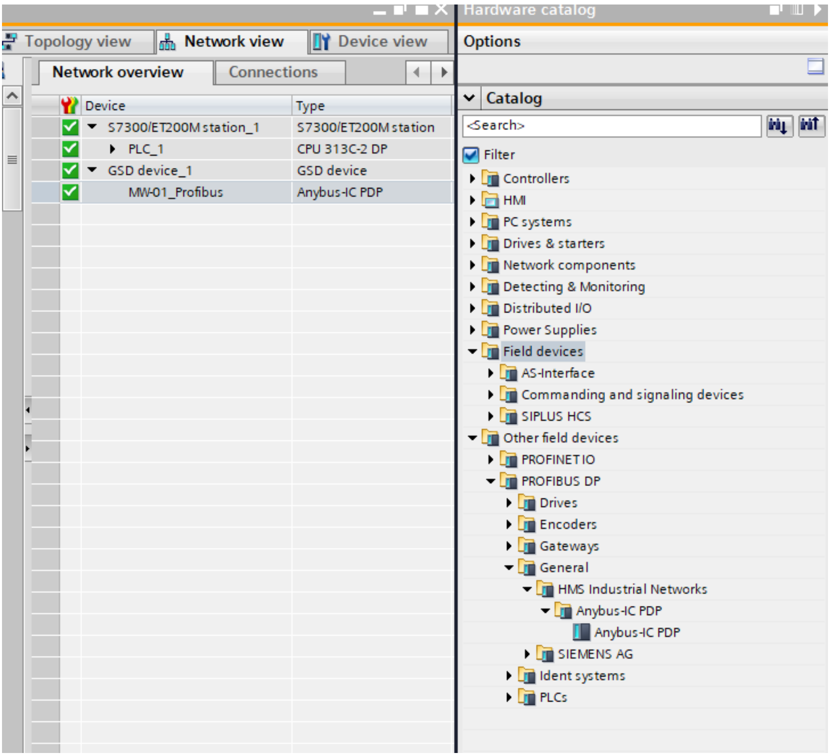

## 3.1. Import GSD

Using a delivered configuration file (GSD), add a new device to the environment. Open OPTIONS tab, next click MANAGE GENERAL STATION DESCRIPTION FILES (GSD) entry and select access path to the GSD file.

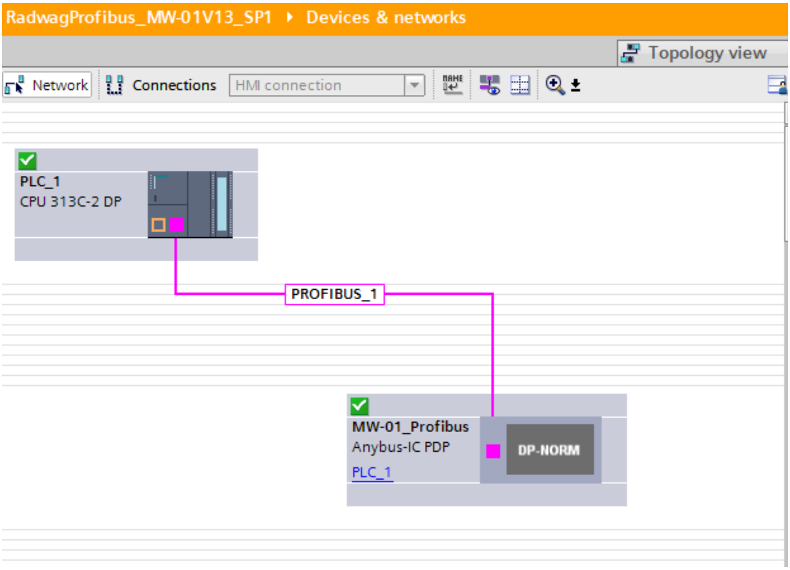




With the file successfully added, the Anybus-IC-PDP module is displayed on the list of the devices.

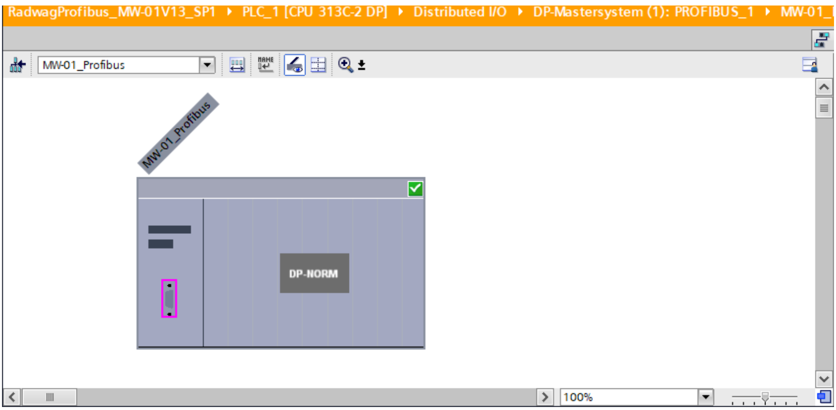


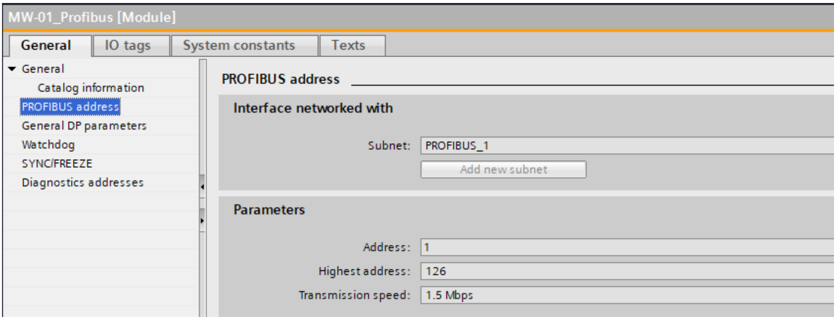
Now it is necessary to make a network comprising a MASTER controller and a newly added SLAVE module.



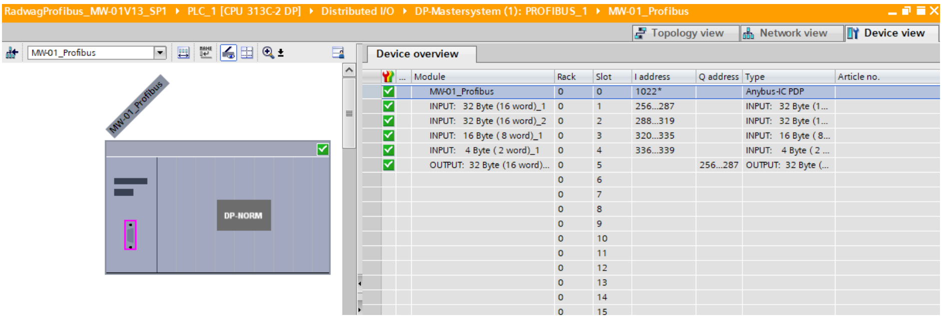

## 3.2. Module Configuration

Now, specify the module address. Make sure that the address is accordant with the address set via the MwManager in the menu.

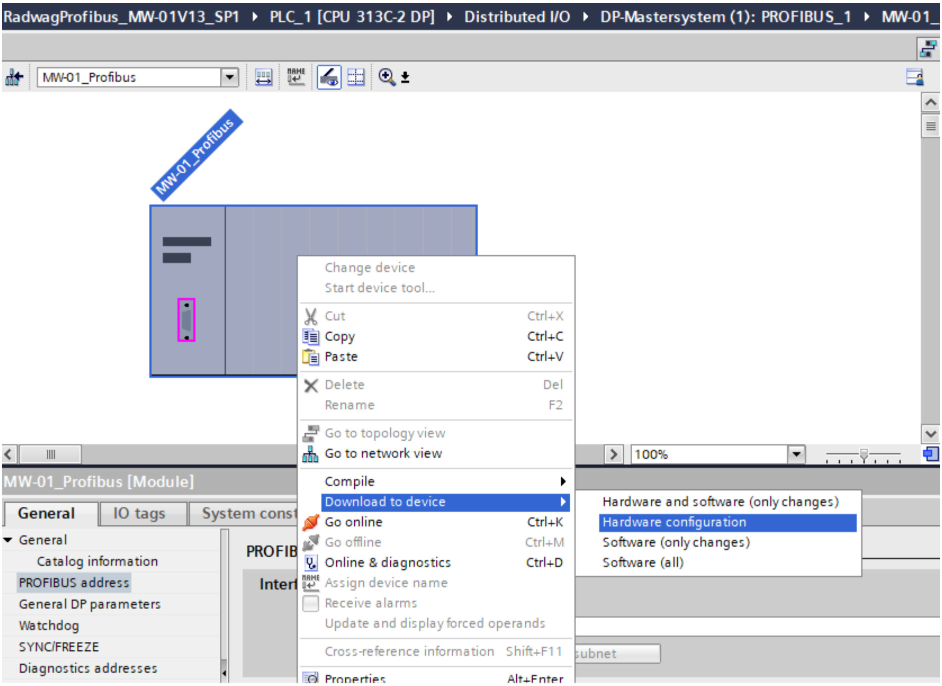





Next configure the module. First specify input and output registers size, define their start addresses. From the list of available INPUT and OUTPUT modules select such modules as presented in the picture below. Maximum size of input data is 84 bytes, maximum size of output data is 32 bytes. In the project, default start addresses have been used – 256 for the INPUT module and 256 for the OUTPUT module:



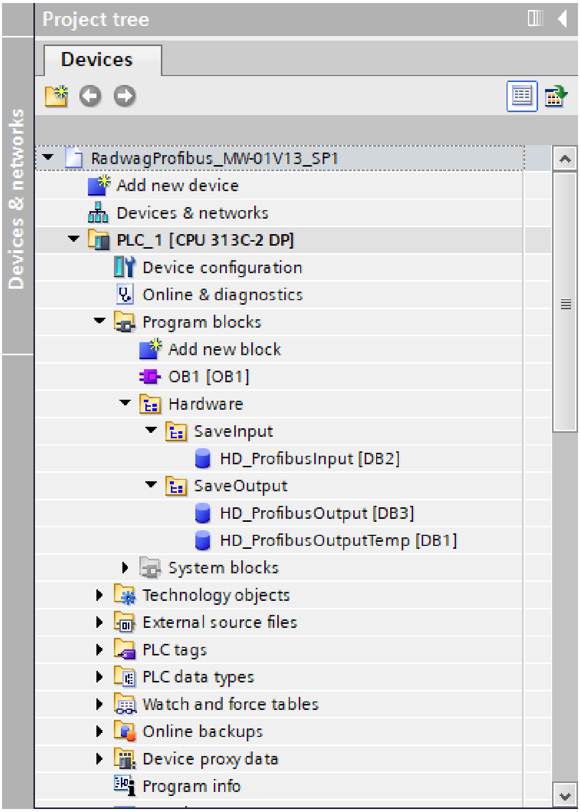

At this stage it is possible to upload the hardware configuration into the controller.




Upon successful compilation and code reading, the MASTER and SLAVE shall establish communication. Now proceed to the process of program code making.

## 4. DIAGNOSTICS APP

Start creating the app by defining names of symbolic input and output registers. Input and output registers of PROFIBUS module have been specified in data blocks, HD_ProfbusInput and HD_ProfbusOutput, in HARDWARE group in PROGRAM BLOCKS.

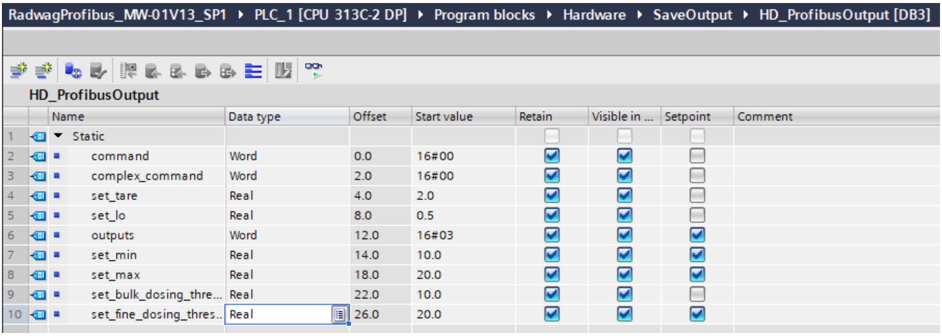



HD_ProfinetOutput and HD_ProfinetInput blocks represent input/output registers of the scale's PROFIBUS module. See the screenshots below:

RadwagProfibus_MW-01V13_SP1 ▸ PLC_1 [CPU 313C-2 DP] ▸ Program blocks ▸ Hardware ▸ SaveInput ▸ HD_ProfibusInput [DB2]

HD_ProfibusInput

| | Name | Data type | Offset | Start value | Retain | Visible in ... | Setpoint | Comment |
|---|---|---|---|---|---|---|---|---|
| 1 | ▼ Static | | | | ☐ | ☐ | ☐ | |
| 2 | mass | Real | 0.0 | 0.0 | ☑ | ☑ | ☐ | |
| 3 | tare | Real | 4.0 | 0.0 | ☑ | ☑ | ☐ | |
| 4 | unit | Word | 8.0 | 16#0 | ☑ | ☑ | ☐ | |
| 5 | status | Word | 10.0 | 16#0 | ☑ | ☑ | ☐ | |
| 6 | lo | Real | 12.0 | 0.0 | ☑ | ☑ | ☐ | |
| 7 | process_status | Word | 16.0 | 16#0 | ☑ | ☑ | ☐ | |
| 8 | inputs | Word | 18.0 | 16#0 | ☑ | ☑ | ☐ | |
| 9 | min | Real | 20.0 | 0.0 | ☑ | ☑ | ☐ | |
| 10 | max | Real | 24.0 | 0.0 | ☑ | ☑ | ☐ | |
| 11 | bulk_dosing_threshold | Real | 28.0 | 0.0 | ☑ | ☑ | ☐ | |
| 12 | fine_dosing_threshold | Real | 32.0 | 0.0 | ☑ | ☑ | ☐ | |

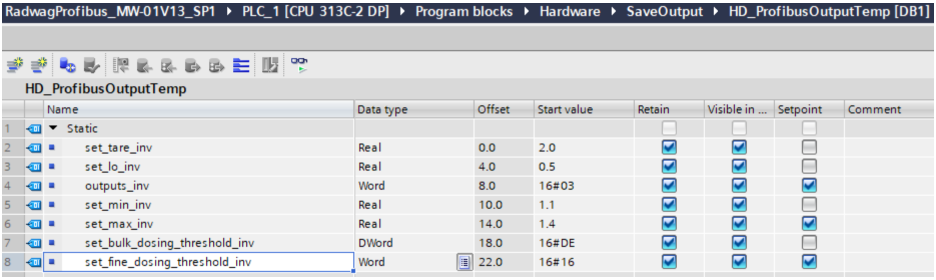


RadwagProfibus_MW-01V13_SP1 ▸ PLC_1 [CPU 313C-2 DP] ▸ Program blocks ▸ Hardware ▸ SaveOutput ▸ HD_ProfibusOutput [DB3]

HD_ProfibusOutput

| | Name | Data type | Offset | Start value | Retain | Visible in ... | Setpoint | Comment |
|---|---|---|---|---|---|---|---|---|
| 1 | ▼ Static | | | | ☐ | ☐ | ☐ | |
| 2 | command | Word | 0.0 | 16#00 | ☑ | ☑ | ☐ | |
| 3 | complex_command | Word | 2.0 | 16#00 | ☑ | ☑ | ☐ | |
| 4 | set_tare | Real | 4.0 | 2.0 | ☑ | ☑ | ☐ | |
| 5 | set_lo | Real | 8.0 | 0.5 | ☑ | ☑ | ☐ | |
| 6 | outputs | Word | 12.0 | 16#03 | ☑ | ☑ | ☑ | |
| 7 | set_min | Real | 14.0 | 10.0 | ☑ | ☑ | ☑ | |
| 8 | set_max | Real | 18.0 | 20.0 | ☑ | ☑ | ☑ | |
| 9 | set_bulk_dosing_thre... | Real | 22.0 | 10.0 | ☑ | ☑ | ☐ | |
| 10 | set_fine_dosing_thres... | Real | 26.0 | 20.0 | ☑ | ☑ | ☑ | |

HD_ProfibusOutputTemp block is for storing temporary data during register bytes reordering.



RadwagProfibus_MW-01V13_SP1 ▸ PLC_1 [CPU 313C-2 DP] ▸ Program blocks ▸ Hardware ▸ SaveOutput ▸ HD_ProfibusOutputTemp [DB1]

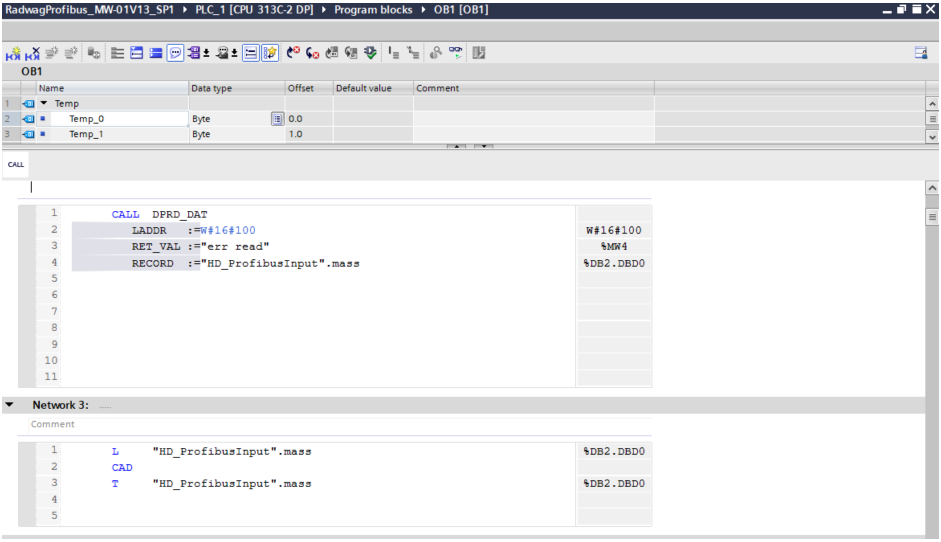
HD_ProfibusOutputTemp

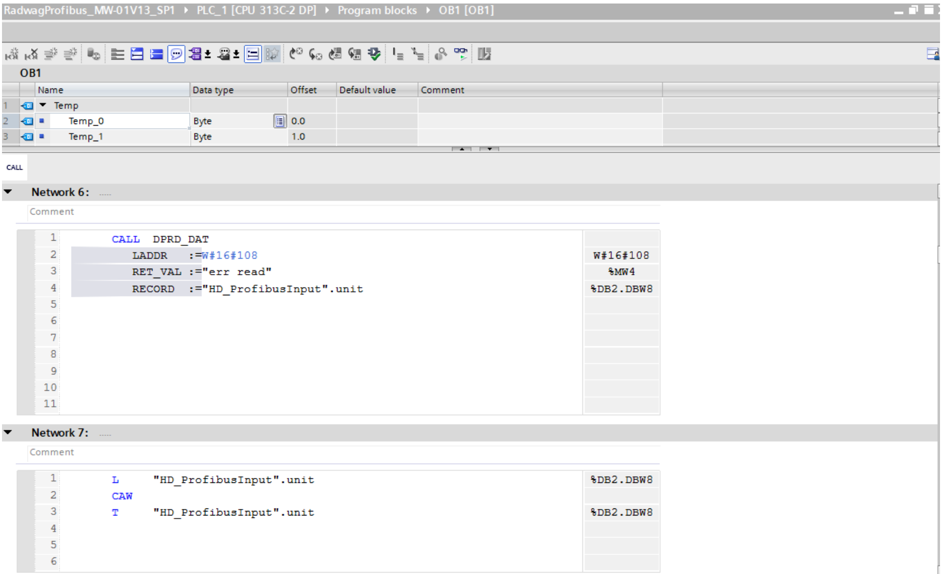
| | Name | Data type | Offset | Start value | Retain | Visible in ... | Setpoint | Comment |
|---|---|---|---|---|---|---|---|---|
| 1 | ▼ Static | | | | ☐ | ☐ | ☐ | |
| 2 | set_tare_inv | Real | 0.0 | 2.0 | ☑ | ☑ | ☐ | |
| 3 | set_lo_inv | Real | 4.0 | 0.5 | ☑ | ☑ | ☐ | |
| 4 | outputs_inv | Word | 8.0 | 16#03 | ☑ | ☑ | ☑ | |
| 5 | set_min_inv | Real | 10.0 | 1.1 | ☑ | ☑ | ☐ | |
| 6 | set_max_inv | Real | 14.0 | 1.4 | ☑ | ☑ | ☑ | |
| 7 | set_bulk_dosing_threshold_inv | DWord | 18.0 | 16#DE | ☑ | ☑ | ☐ | |
| 8 | set_fine_dosing_threshold_inv | Word | 22.0 | 16#16 | ☑ | ☑ | ☑ | |

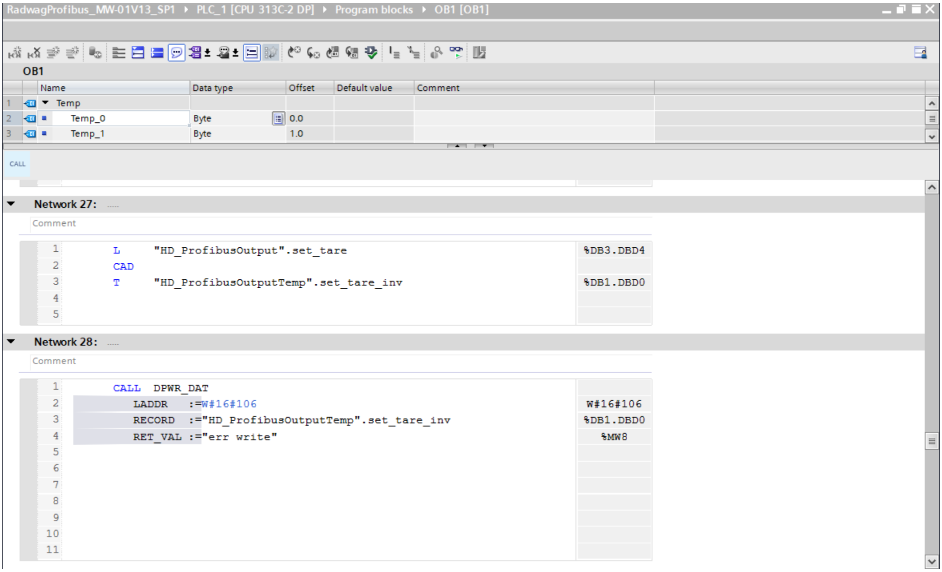
Now, in the main program loop, make function assigning states of physical scale registers to registers in HD_ProfibusInput and HD_ProfibusOutput data blocks. Functions may look as presented below. The example shows the method of mass and unit readout and record of 'tare' and 'command' registers.

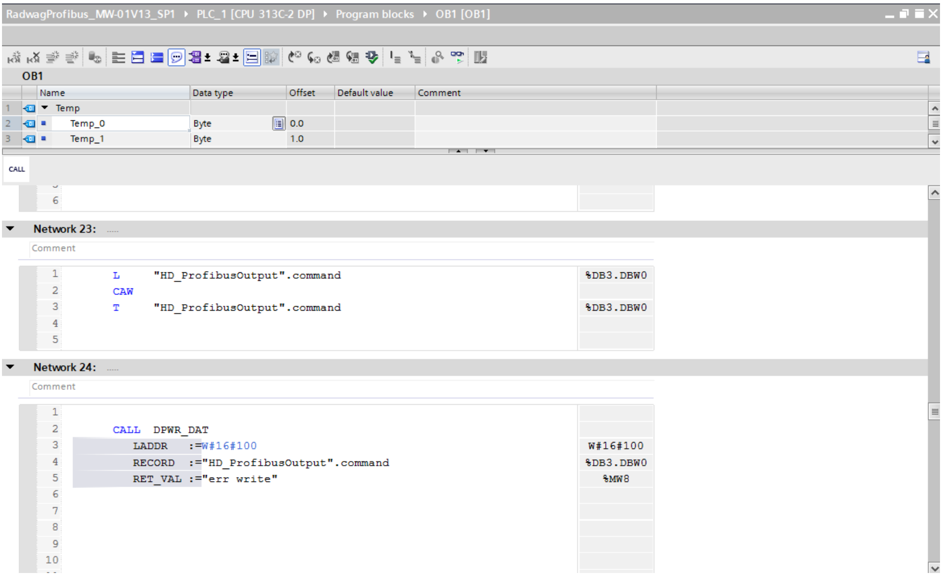
Note that data uploaded from and recorded to the MW-01-A mass converter have reverse order of register bytes, i.e. float data order is DCBA, and word variables order is BA. To correctly read these registers it is necessary to reorder them. In the presented example CAD command was used for variables of float type, and CAW command for variables of word type.

The same rule concerns variables recorded in the mass converter. Prior to the record it is necessary to set the bytes in a reverse order. Learn how it works on the example of record of tare of 1.5 value. Upon conversion to HEX the result is 0x3FC00000. Prior to record to the MW-01-A, conversion to DCBA order must be carried out. After use of CAD command, 0000C03F value is obtained, this value must be recorded to the scales register (log).


RadwagProfibus_MW-01V13_SP1 ▸ PLC_1 [CPU 313C-2 DP] ▸ Program blocks ▸ OB1 [OB1]
OB1
Name | Data type | Offset | Default value | Comment
Temp
Temp_0 Byte 0.0
Temp_1 Byte 1.0
CALL
CALL DPRD_DAT
LADDR :=W#16#100 W#16#100
RET_VAL :="err read" %MW4
RECORD :="HD_ProfibusInput".mass %DB2.DBD0
Network 3:
Comment
L "HD_ProfibusInput".mass %DB2.DBD0
CAD
T "HD_ProfibusInput".mass %DB2.DBD0


RadwagProfibus_MW-01V13_SP1 ▸ PLC_1 [CPU 313C-2 DP] ▸ Program blocks ▸ OB1 [OB1]
OB1
Name | Data type | Offset | Default value | Comment
Temp
Temp_0 Byte 0.0
Temp_1 Byte 1.0
CALL
Network 6:
Comment
CALL DPRD_DAT
LADDR :=W#16#108 W#16#108
RET_VAL :="err read" %MW4
RECORD :="HD_ProfibusInput".unit %DB2.DBW8
Network 7:
Comment
L "HD_ProfibusInput".unit %DB2.DBW8
CAW
T "HD_ProfibusInput".unit %DB2.DBW8




Upon compilation and upload of the program to the controller in data block it is possible to read input registers (MONITOR ALL) and to record output registers (e.g. by change of START VALUE and LOAD START VALUES AS ACTUAL) of the SLAVE module.

RADWAG BALANCES AND SCALES
ADVANCED WEIGHING TECHNOLOGIES