USER GUIDE

PUE HX7

Radwag

Distributed by AKRIBIS

Software manual

ITKP-22-01-05-20-EN

PROFIBUS

Communication Protocol:
PUE HX7 Weighing Terminal

radwag.com
Scan the QR code to view additional scientific material that may be of interest to you.
There you will find more useful information in an accessible format!

AKRIBIS www.akribis.info contacto@akribis.info

MAY 2020

CONTENTS

1. INDICATOR SETTINGS AND 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 PROFIBUS MODULE IN TIA PORTAL V13 ENVIRONMENT 10
3.1. Import GSD 11
3.2. Module Configuration 13
4. DIAGNOSTICS APP 14

1. INDICATOR SETTINGS AND CONFIGURATION

To configure indicator settings for communication via Profibus prorotcol go to **<SETUP / Peripherals / Anybus module>** submenu. For detailed configuration read **„PUE HX7 Indicator”** user manual.

2. DATA STRUCTURE

2.1. Input Address

Input variables:

Variable	Offset	Length [WORD]	Data type
Platform 1 mass	0	2	float
Platform 1 tare	4	2	float
Platform 1 unit	8	1	word
Platform 1 status	10	1	word
Platform 1 LO threshold	12	2	float
Platform 2 mass	16	2	float
Platform 2 tare	20	2	float
Platform 2 unit	24	1	word
Platform 2 status	26	1	word
Platform 2 LO threshold	28	2	float
Process status (Stop, Start)	64	1	word
Inputs status	66	1	word
Min	68	2	float
Max	72	2	float
Series 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 a current unit.

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: 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 status of the dosing\formulation:

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

Input state – bitmask of indicator inputs. 4 first least significant bits represent weighing indicator 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 number 1, 2 and 3 take HI state.

MIN – returns **MIN** threshold value (in a current unit).

MAX - returns **MAX** threshold value (in a current unit).

Serial number – returns serial number value. Numerical values exclusively! Any non-numerical values are neglected.

Operator – returns the value of a logged-in operator code.

Product – returns the value of a selected product code.

Customer – returns the value of a selected customer code.

Packaging – returns the value of a selected packaging code.

Formulation – returns the value of a selected formulation code.

Dosing process – returns the value of a selected dosing process code.

2.3. Output Address

Output 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
Serial 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 – record of the register via a given value triggers a respective operation:

Bit No.	Operation
0	Zero the platform
1	Tare the platform
2	Clear statistics
3	Save/Print
4	Process start
5	Process stop

Example:

Record of the register by 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

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 series number
3	Setting output status
4	Operator selection
5	Product selection
6	Packaging selection
7	Setting MIN threshold value
8	Customer selection
9	Source warehouse selection
10	Target warehouse 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 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 for platform 1 to the scale.

Performance of the command requires record of 3 registers:

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

offset 4 – number of a platform to which tare is to be assigned - value 0x01 for platform 1,

offset 6 – tare value in float format - 1.0 .

Platform – complex command parameter: platform number (1 or 2).

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: status of weighing indicator and communication module 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).

Serial number – complex command parameter: serial number value. Numerical values exclusively! Any non-numerical values are neglected

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

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

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

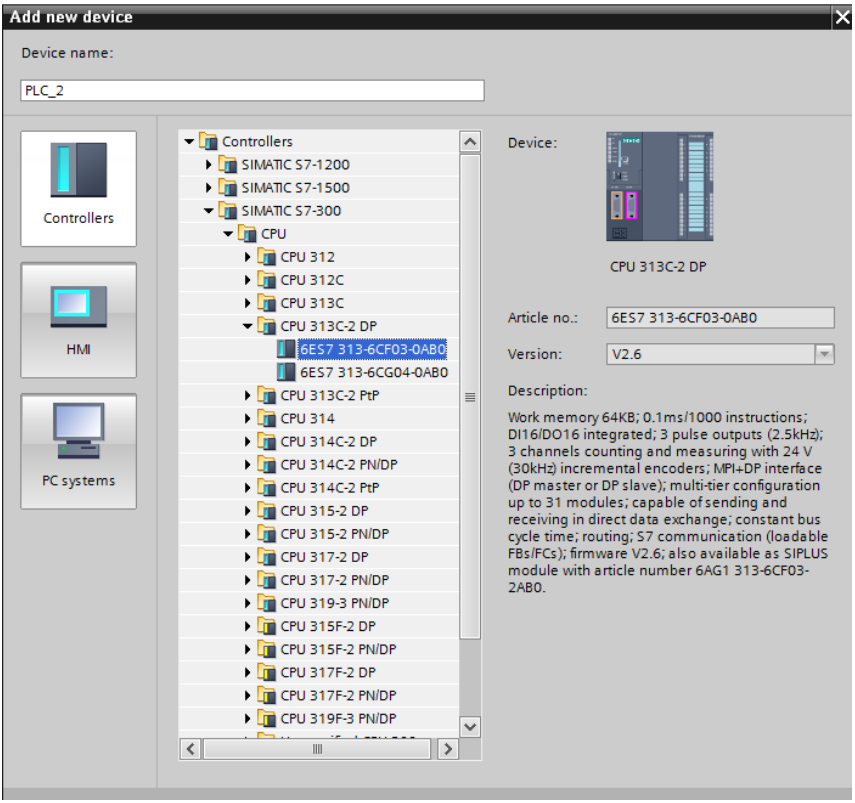
Packaging – complex command parameter: packaging code (digits only)

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

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

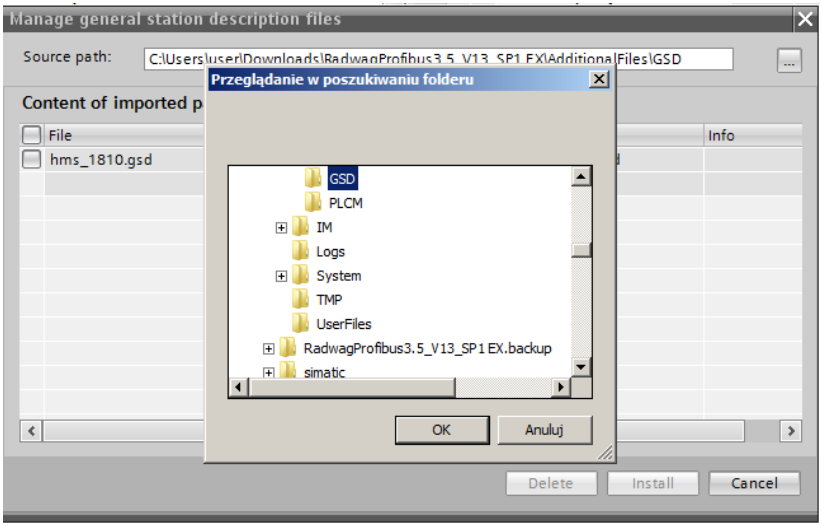
3. CONFIGURATION OF PROFIBUS MODULE IN TIA PORTAL V13 ENVIRONMENT

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.

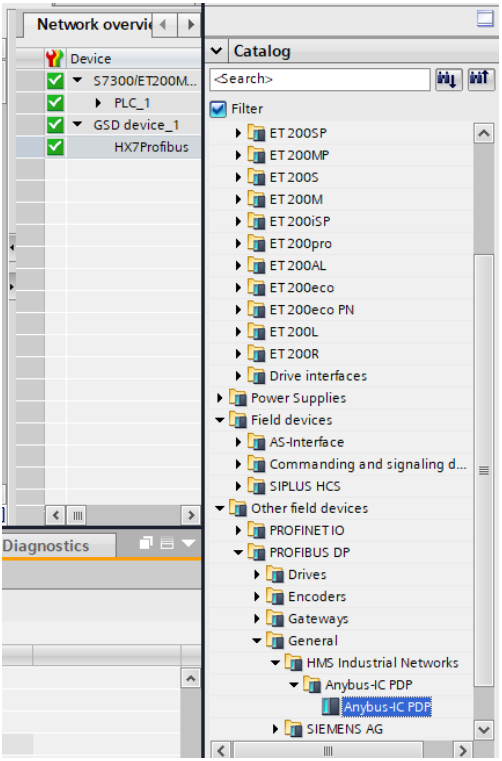


3.1. Import GSD

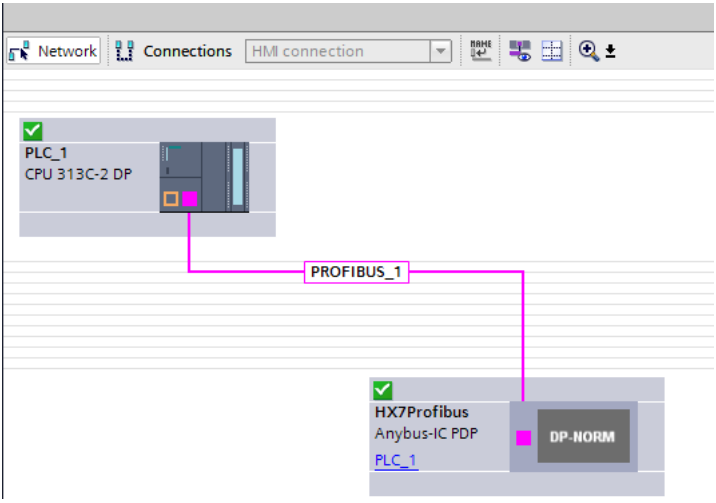
Using a delivered configuration file (GSD), add a new device into the environment. Open OPTIONS tab, next click MANAGE GENERAL STATION DESCRIPTION FILES (GSD) entry and select access path to the GSD file.



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

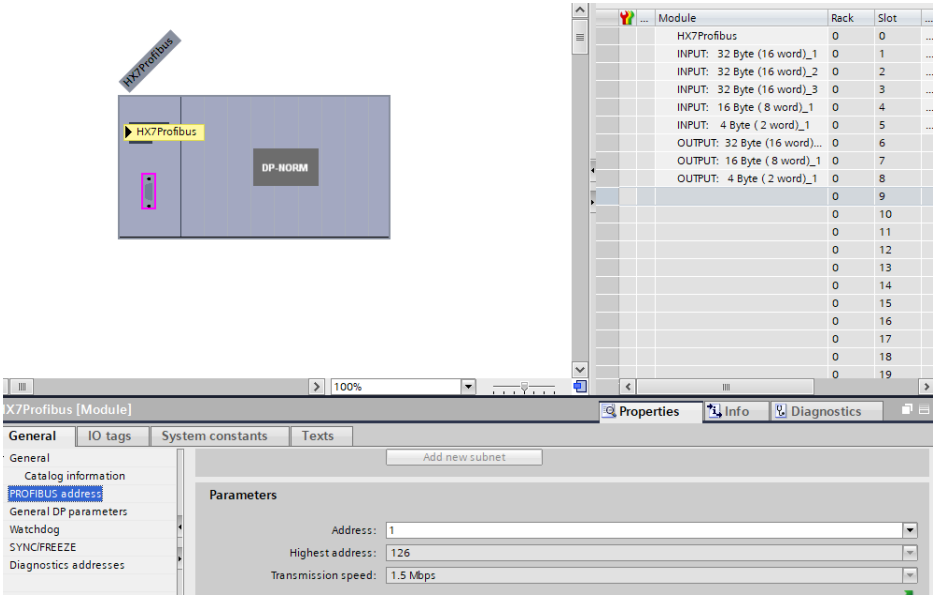


Now it is necessary to make a network comprising a MASTER controller and a newly added SLAVE module.

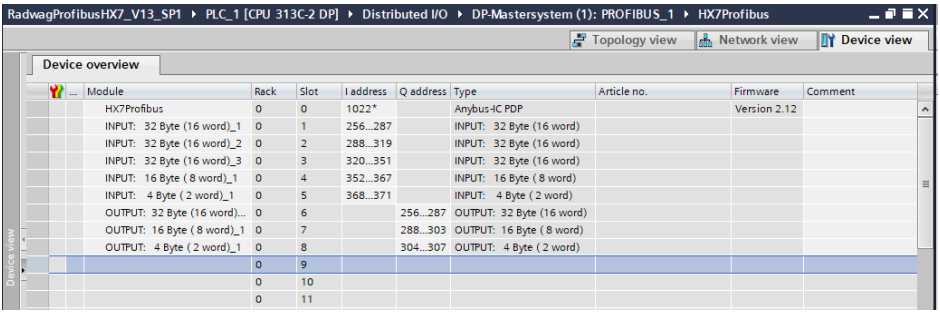


3.2. Module Configuration

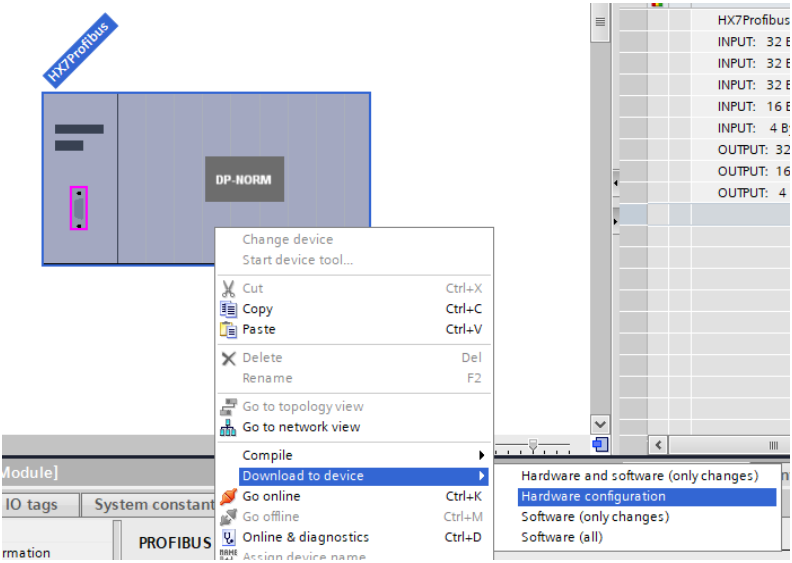
Now, specify the module address. Make sure that the address is accordant with the address set in the scale menu.



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 116 bytes, maximum size of output data is 116 bytes too. In the project, default start addresses have been used – 256 for the INPUT module and 256 for the OUTPUT module:



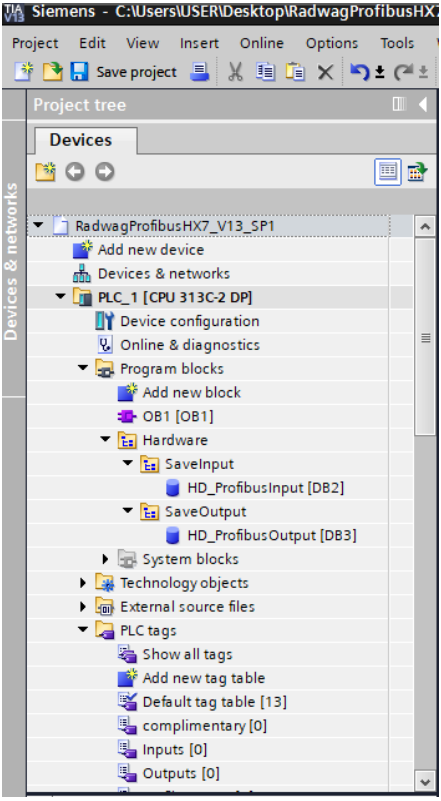
At this stage it is possible to upload the hardware configuration into the controller, and to start data upload.



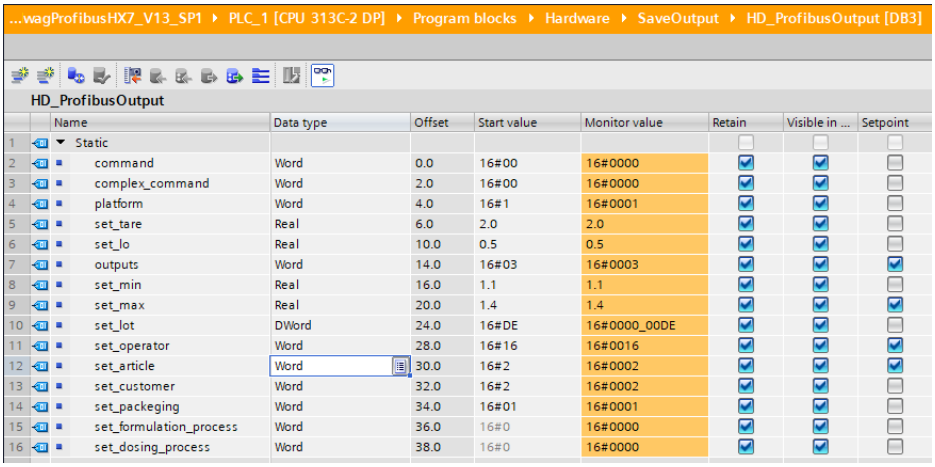
Upon successful compilation and code reading, the MASTER and SLAVE shall establish communication. Now proceed to the process of program code making.

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_ProfibusInput and HD_ProfibusOutput, 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:



	Name	Data type	Offset	Start value	Monitor value	Retain	Visible in ...	Setpoint
1	Static							
2	command	Word	0.0	16#00	16#0000	☑	☑	☐
3	complex_command	Word	2.0	16#00	16#0000	☑	☑	☐
4	platform	Word	4.0	16#1	16#0001	☑	☑	☐
5	set_tare	Real	6.0	2.0	2.0	☑	☑	☐
6	set_lo	Real	10.0	0.5	0.5	☑	☑	☐
7	outputs	Word	14.0	16#03	16#0003	☑	☑	☑
8	set_min	Real	16.0	1.1	1.1	☑	☑	☐
9	set_max	Real	20.0	1.4	1.4	☑	☑	☑
10	set_lot	DWord	24.0	16#DE	16#0000_00DE	☑	☑	☐
11	set_operator	Word	28.0	16#16	16#0016	☑	☑	☑
12	set_article	Word	30.0	16#2	16#0002	☑	☑	☑
13	set_customer	Word	32.0	16#2	16#0002	☑	☑	☐
14	set_packeging	Word	34.0	16#01	16#0001	☑	☑	☐
15	set_formulation_process	Word	36.0	16#0	16#0000	☑	☑	☐
16	set_dosing_process	Word	38.0	16#0	16#0000	☑	☑	☐

RadwagProfibusHX7_V13_SP1 ▸ PLC_1 [CPU 313C-2 DP] ▸ Program blocks ▸ Hardware ▸ SaveInput ▸ HD_ProfibusInput [D

HD_ProfibusInput

	Name	Data type	Offset	Start value	Monitor value	Retain	Visible in ...	Setpoint
1	Static							
2	mass 1	Real	0.0	0.0	0.0	☑	☑	☐
3	tare 1	Real	4.0	0.0	0.0	☑	☑	☐
4	unit 1	Word	8.0	16#0	16#0002	☑	☑	☐
5	status 1	Word	10.0	16#0	16#0007	☑	☑	☐
6	lo 1	Real	12.0	0.0	0.5	☑	☑	☐
7	mass 2	Real	16.0	0.0	0.0	☑	☑	☐
8	tare 2	Real	20.0	0.0	0.0	☑	☑	☐
9	unit 2	Word	24.0	16#0	16#0000	☑	☑	☐
10	status 2	Word	26.0	16#0	16#0000	☑	☑	☐
11	lo 2	Real	28.0	0.0	0.0	☑	☑	☐
12	process_status	Word	32.0	16#0	16#0000	☑	☑	☐
13	inputs	Word	34.0	16#0	16#0008	☑	☑	☐
14	min	Real	36.0	0.0	0.0	☑	☑	☐
15	max	Real	40.0	0.0	0.0	☑	☑	☐
16	lot_number	DWord	44.0	16#0	16#0000_00DE	☑	☑	☐
17	operator	Word	48.0	16#0	16#0016	☑	☑	☐
18	article	Word	50.0	16#0	16#0002	☑	☑	☐
19	customer	Word	52.0	16#0	16#0002	☑	☑	☐
20	packaging	Word	54.0	16#0	16#0000	☑	☑	☐
21	source_warehouse	Word	56.0	16#0	16#0000	☑	☑	☐
22	target_warehouse	Word	58.0	16#0	16#0000	☑	☑	☐
23	formulation	Word	60.0	16#0	16#0000	☑	☑	☐
24	dosing	Word	62.0	16#0	16#0000	☑	☑	☐

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. In the example, mass readout method and record of "output state" and "min" registers are presented.

OB1

	Name	Data type	Offset	Default value	Comment
1	Temp				
2	Temp_0	Byte	0.0		
3	Temp_1	Byte	1.0		

CALL

Network 2:

Comment

```
CALL  DPRD_DAT
   LADDR   :=W#16#100                          W#16#100
   RET_VAL :="err read"                          %MW4
   RECORD  :="HD_ProfibusInput"."mass 1"       %DB2.DBD0
```

Network 3:

Comment

```
CALL  DPRD_DAT
   LADDR   :=W#16#104                          W#16#104
   RET_VAL :="err read"                          %MW4
   RECORD  :="HD_ProfibusInput"."tare 1"       %DB2.DBD4
```

Comment

```
CALL  DPWR_DAT
   LADDR   :=W#16#10E                          W#16#10E
   RECORD  :="HD_ProfibusOutput".outputs       %DB3.DBW14
   RET_VAL :="err write"                         %MW8
```

Network 29:

Comment

```
CALL  DPWR_DAT
   LADDR   :=W#16#110                          W#16#110
   RECORD  :="HD_ProfibusOutput".set_min       %DB3.DBD16
   RET_VAL :="err write"                         %MW8
```

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.

AKRIBIS www.akribis.info contacto@akribis.info