

SOFTWARE MANUAL

ITKP-23-01-05-20-EN



MAY 2020

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 ENVIRONMENT	10
	3.1. Import GSD	10
	3.2. Module Configuration	12
4.	DIAGNOSTICS APP	16

1. WEIGHING INDICATOR SETTINGS - CONFIGURATION

To set indicator communication via PROFINET protocol go to **SETUP / Peripherals/ Anybus module>**. For detailed description of settings configuration read **"PUE HX7 Weighing 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
Input status	66	1	word
Min	68	2	float
Max	72	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.

<u>Platform tare</u> – returns platform tare in an adjustment unit.

<u>Platform unit</u> – 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].

<u>Platform status</u> – determines state of a given weighing platform.

Status	s 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).

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

Operator – response: code of logged in operator.

<u>Product</u> – 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

 $\underline{\textbf{Basic command}}$ – 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	Process start
5	Process stop

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.



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 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
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 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.

<u>Platform</u> – complex command parameter: weighing 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).

<u>**Output state**</u> – 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.

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

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

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

<u>Operator</u> – complex command parameter: operator code (digits only).

<u>Product</u> – complex command parameter: product code (digits only).

<u>Customer</u> – complex command parameter: customer code (digits only).

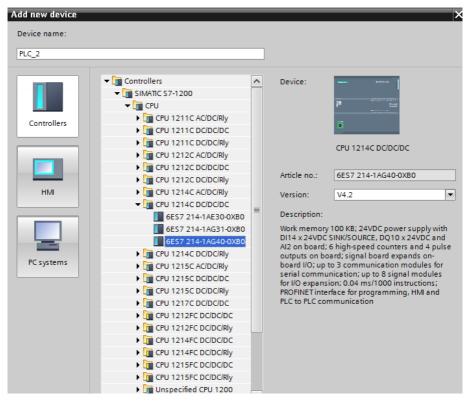
Packaging – complex command parameter: packaging code (digits only)

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 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 S7-1200 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.

Manage general station	description files	×
Source path: C:\Users	userIDesktonIRadwaoProfinetSamoleSCI 1 11AdditionalFiles GSD Przeglądanie w poszukiwaniu folderu	
Content of imported p		
GSDML-V2.3-HMS-ABICI	SSDML files HYPERTRM HICOVIN NEW E profinet SCI E Radwag Profinet Sample 1.0 E Radwag Profinet Sample 1.0.backup R Radwag Profinet Sample 1.0.backup R Radwag Profinet Sample 1.0.v13 SP1	Info Anybus-IC
<	OK Anuluj	>
	Delete Install	Cancel

With the file successfully added, the ABIC-PRT module is displayed on the list of the devices.

ırch in project>		Totally Integrated A	Automation PORTAL
	_ = = ×	Hardware catalog	1 1 •
logy view 🔒 Network view	Device view	Options	
Network overview Conner	ctions (
Y Device	Туре	✓ Catalog	
 S7-1200 station_1 	S7-1200 station	<pre>dearch></pre>	fini Lini
▶ PLC_1	CPU 1214C DC/DC/DC	Filter Profile: <all></all>	
 GSD device_1 	GSD device	Controllers	- 📑
ABIC-PRT	RT Migration (FW>=	HMI	
		PC systems	
		Drives & starters	
		Intel a state is Network components	
		Detecting & Monitoring	
		Distributed I/O	
		Power supply and distribution	
		Field devices	
		Other field devices	
		Additional Ethernet devices	
		Drives	
		Encoders	
		🕨 🛅 Gateway	
		🕶 🧊 General	
<	>	🕶 🛅 HMS Industrial Networks	
operties 🚺 Info 🔋 🛚 Diagr		🕶 🫅 Anybus-IC PRT	
spercies I sinto si si biagr	lostics	→ Im Migration	
		RT Migration (FW 1.xx)	
		RT Migration (FW>=2.00)	
		🚺 RT Standard	

Now you can make a network comprising a MASTER controller and a newly added SLAVE module.

RadwagProfinetSampleSCL1.1 Devices & networks
💦 Network 🔡 Connections HM connection 🔽 📅 🖽 🛄 🔍 🛓
PLC_1 ABIC-PRT RT Migration (F PLC_1 PLC_1
PN/IE_1

3.2. Module Configuration

At this stage, it is necessary to create a network consisting of MASTER controller and SLAVE device (scale). Upon connecting to the mains, use ACCESSIBLE DEVICES function to search for devices in the environment. As a result, MASTER and SLAVE modules can be found on the list:

		Type of the PG/PC interface: PG/PC interface:	PN/IE Realtek PCIe GBE Family Controller					
	Accessible nodes of the	e selected interface:						
	Device	Device type	Interface type	Address	MAC address			
	Accessible device	S7-PC	ISO		00-16-76-25-13-51			
	pro2	RT Migration (FW 1.xx)	PN/IE	10.10.8.64	00-30-11-0D-EE-17			
	plc_1	CPU 1214C DC/DC/DC	PN/IE	10.10.8.244	28-63-36-9C-D1-12			
Flash LED								
nline status informatio	on:			Display only erro	<u>Start search</u> or messages			
Found accessible of	levice Accessible device [0	0-16-76-25-13-51]			1			
Scan completed. 3	devices found.							
? Retrieving device in	nformation							
Scan and information	on retrieval completed.				•			

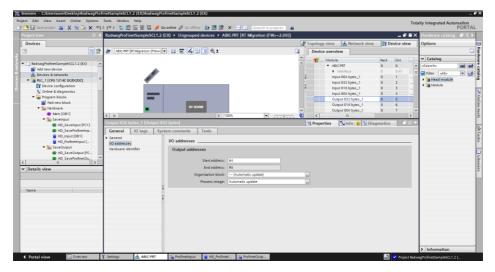
Now, specify the module IP address and its name in the PROFINET network. After selecting the module in the PROPERTIES tab, find the PROFINET INTERFACE submenu and enter the IP address and a name. Make sure that the setting is accordant with the parameters set in the scale menu. Make sure that IP SLAVE address is in the same subnet as the MASTER address.

< =	>	100%		2 <		1000	>
ABIC-PRT [RT]	digration (I	FW>2.00)]		S Propert	ies 🔄 Info	1 Diagnostics	
General	IO tags	System constants	Texts				
- General Catalog info	ormation			Add new s	ubnet]		^
PROFINET inter General Ethernet ad Advanced a Real time IO cyc Hardware iden	dresses options e settings de dentifier & Maintenanc	IP protocol		Subnet m	ess: 10 , 10 ask: 255 , 255 ess: 0 , 0		•
		C.0000	INET device name Converted name:	Generate PRO	FINET device name	e automatically	

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 screenshot 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 – 68 for the INPUT module and 64 for the OUTPUT module:

Mi Siemens - C:/Users/user/Desktop/RadwagProf	inetSampleSCL	1.2 (EX)/Radwa	•gProf	netSampleSCL1.2	(EX)		_				_		- •
Project Edit View Insert Online Options											Tota	ally integrated Autor	nation
🕒 🎦 🔚 Save project 🚢 🐰 🐚 🕞 🗙 🍤	± (= ± 🚯 🖪		🍠 Go	inline 🖉 Go offline	a 🖪 🖪 🕫	🗶 🖃 💷 🛛 Search in project> 🐪						ing integrated ristor	PORTAL
Project tree 🛛 🖬 🖣	RadwagProfi	netSampleSCL	1.2 (E	V Ingrouped	I devices + Al	IIC-PRT [RT Migration (FW>=2.00)]				_ •	∎×	Hardware catalog	
Devices							1	Topology view 🔥 Network view	0 10	evice vie	w	Options	
1 I I I I I I I I I I I I I I I I I I I	ABIC-PRT	RT Migration (FW	6=2 ¥		1 🛛 🔍 ±		1	Device overview					
							^	W Module	Rack	Slot		✓ Catalog	
🗧 💌 📑 RadwagProfinetSampleSCL1.2 (EX) 📃 🔨	1						-	ABIC-PRT	0	0	-	Search>	HAL HAT
Add new device	1		L					Interface	0	0.X1			
📥 Devices & networks	1	25						Input 064 bytes_1	0	1		Filter <al></al>	
 PLC_1 [CPU 1214C DC/DC/DC] 	1	~						Input 032 bytes_1	0	2		+ 📑 Head module	۲ ۲
Device configuration	1							Input 016 bytes_1	0	3		🕨 🧾 Module	
Conline & diagnostics	1		- 11					Input 004 bytes_1	0	4			
 Program blocks 								Output 032 bytes_1	0	5			
Add new block				00.000			•	Output 016 bytes_1	0	6			
Hardware Main [061]							2	Output 004 bytes_1	0	7			Cinite work
SaveInput									0	8			
HD_SaveInput [FC1]									0	9			10000
HD_SaveProfineting									0	10			
HD_Input [DB1]									0	11			
HD Profinetinput [0	12			2
 SaveOutput 									0	13			
HD SaveOutput IFC									0	14			
HD SaveProfinetOu	1								0	15			5
< = >							×		0	10	~		Lining
✓ Details view	< =)				> 1	10% ×	1	< II		_	>		
	ABIC-PRT [RT	Migration (F)	N>-2.	10)]				🖪 Properties 🚺 Info 🚯 🖏 Dia	gnostics		.		
	General	IO tags	Syste	m constants	Texts								
Name	▼ General		1	Catalog information							^		
		nformation		catalog informatio			_		_		- 1		
	· PROFINET int	erface [X1]											
	General			Sh	ort designation:	RT Migration (FW>=2.00)							
		addresses	- 4		Description:	This Device Access Point maybe used by IC	0 Cont	rollers that do not support extended PROFII	ET diagn	ostics.			
	 Advanced 	ne settings											
		ne settings ytle	- 1										
	Hardware												
		& Maintenance											
	Hardware id				Article no.:								_
					rmware version:				_			> Information	
Portal view Overview	Settions	ABIC-PRT				t ProfinetOute						anthrofinatCampleS(1) 1 2	

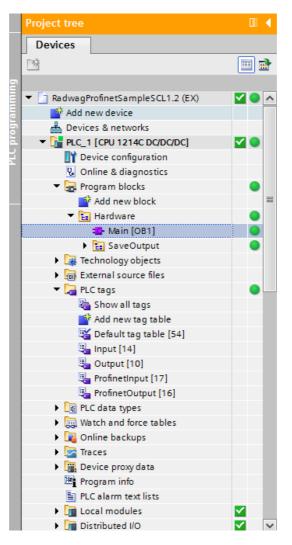
14 Siemens - C:IUsers\user\Desktop\RadwagProf	inetSampleSCL1.2 (EX)/RadwagPr	ofinetSampleSCL1.2 (EX)					_ # X
Project Edit View Insert Online Options					Tota	ally Integrated Auto	mation
📑 🛅 🖬 Save project 🏭 💥 🗟 🕞 🗙 🄊	± (#± 🖏 🖽 🖬 🔛 🖬 💋 🤇	Go online 🖉 Go offline 🛛 🛃 📳	🗶 🖃 🛄 < Search in project> 🖣	1		iny integrated risto	PORTAL
Project tree 🛛 🛛 🖣	RadwagProfinetSampleSCL1.2	(EX) • Ungrouped devices • /	BIC-PRT [RT Migration (FW>=2.00)]		_ # = ×	Hardware catalog	
Devices				Topology view 🔥 Network view	Device view	Options	03
	ABIC-PRT [RT Migration (FW)=2	- 🗉 🗹 🖌 🗐 🔍 ±		Device overview			
-				A	Rack Slot	✓ Catalog	
RadwagProfinetSampleSCL1.2 (EX)	1			ABIC-PRT	NACK SIDE _	Search>	HI WI
Add new device				Interface	0 0X1 =	Filter All>	
📩 Devices & networks	C.			Input 064 bytes_1	0 1	Head module	
PLC_1 [CPU 1214C DQ/DC/DC]	1			Input 032 bytes_1	0 2	Head module Module	-
Device configuration				 Input 016 bytes_1 	0 3	 Mobule 	
Online & diagnostics Frogram blocks	_			Input 004 bytes_1	0 4		3 Online
Add new block				Output 032 bytes_1	0 5		- III
Had new block	-	OP-NORM		Output 016 bytes_1	0 6		5
- Main [081]	-			Output 004 bytes_1	0 7 🗸		tools
 SaveInput 	< 1		100%				5
HD_SaveInput [FC1]	Input 064 bytes_1 [Input 064	bytes]		Properties 🚺 Info 🚺 💆 Diag	nostics		-
HD_SaveProfinetinp	General IO tags Sys	stem constants Texts					2
HD_Input [D81]	General Inputs	VO addresses					(ii) Tasks
SaveOutput HD_SaveOutput [FC	NO addresses Hardware identifier	Input addresses					
HD SaveProfinetOu	Hardware identifier	Start address					5
<		End address					ran
✓ Details view							6 S
			(Automatic update)				
		Process image	: Automatic update				
Name							
							_
						> Information	_
Portal view Dverview	Y Settings 🔥 ABIC-PRT	🛬 ProfinetInput 🧧 HD_Profi	retl 🍇 ProfinetOutp		📑 💙 Project Radw	agProfinetSampleSCL1.2	(



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

	Help					
± 🖥 🗓 🛛	🚹 🖳 📮 💋 Go online 🚀 Go off	fline 🔥 🚺	X 🗄 🛄 🗠	arch in project> 🛛 🖣	ú .	
Radwag	ProfinetSampleSCL1.2 (EX) 🕨 D	evices & netwo	orks			_ •
			🛃 Тор	ology view 🛛 🛔 🖡	Network view	Device vie
Netwo	rk Connections HMI connectio	n 🔻 🛤	z =	Network overvi	ew Conne	ctions
			^	Pevice		Туре
			====	▼ \$7-1200	station_1	S7-1200 station
PLC_1	1010.000	- •	_	PLC_1		CPU 1214C DC/
CPU 121	Device configuration		NORM	 GSD devi 	ce_1	GSD device
	Change device			ABIC-F	PRT	RT Migration (F)
	🗶 Cut	Ctrl+X				
	E Copy	Ctrl+C				
	📋 Paste	Ctrl+V				
	🗙 Delete	Del				
	Rename	F2				
	Assign to new DP master / IO co	ntroller				
	Disconnect from DP master syste	em / IO system	~			
< 111	Highlight DP master system / IO	system		<	1111	
\$7-1200	🚽 Go to topology view		Q	Properties 🚺 II	nfo 😵 Diag	nostics
Genera	Compile	•			1_ 0	
-	Download to device	Þ	Hardware and so	ftware (only changes)		
▼ Genera		Ctrl+K	Hardware config	uration		
Proje	🔊 Go offline	Ctrl+M	Software (only ch	anges)		
	Q Online & diagnostics	Ctrl+D	Software (all)			
	Assign device name					
	Receive alarms					
	Update and display forced opera		Name: \$7-1200 s	tation_1		
~	Show catalog	Ctrl+Shift+C	omment:			~
-	Export data for TCSB					
	Proportion	Alt - Entor				

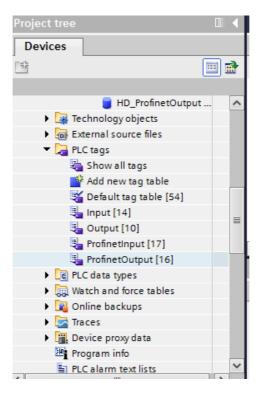
Upon successful compilation and code reading, the MASTER and SLAVE shall establish communication. It can be verified via ONLINE connection. The result should be as presented below:



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. For this purpose, use a branch of the PLC TAGS tree. For the purpose of this example, tag tables have been created as presented below:

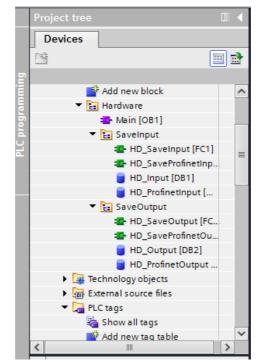


INPUT and OUTPUT tables refer to physical inputs/outputs of the MASTER controller and are not relevant in this application. ProfinetInput and ProfinetOutput blocks represent input/output registers of the scale's PROFINET module. See the screenshots below (names and addresses are examples only):

Ra	ıdw	agPr	ofinetHX7 ang 🕨 PLC	_1 [CPU 1214C D	C/DC/DC] ▶ Prog	ram blocks	 Hardware 	► Sav	eOutput ▸	HD_Profi	netOutput [DB4]	_∎∎×
1	2	1	a 🛃 🗮 🎌 Keep a	ctual values 🔒 🤮	Snapshot 🖷 🖷	Copysnaps	hots to start va	lues 🔣	🖳 Load	start values	as actual values 🛛 🖳 🕮	
	HD	_Pro	ofinetOutput									
		Nam	ne	Data type	Start value	Retain	Accessible f	Writa	Visible in	Setpoint	Comment	
1	-0	•	Static									
2	-00	•	command	Word	16#0000							
З	-10	•	complex command	Word	16#0000							
4	-0		platform	Word	16#01							
5		•	set tare	Real	2.0							
6		•	set lo	Real	4.0							
7	-0	•	outputs	Word	16#0002							
8	-00	•	set min	Real	2.2							
9	-0	•	set max	Real	3.3							
10	-0		set lot number	DWord	444							
11		•	set operator	Word	12							
12	-0	•	set article	Word	2							
13	-00	•	set customer	Word	16#0							
14	-00	•	set packaging	Word	16#0							
15	-0		set recipe	Word	16#44							
16	-0	-	set dosing	Word	16#0							
17			<add new=""></add>									

Rad	wagP	rofinetHX7 ang → PLC	_1 [CPU 1214C DC/D	C/DC] ► PLC	tags 🕨 Prot	finetInpu	ıt [20]				_ # = ×
										🕣 Tags	User constants
*	÷ [> ♥♥ III									
		etInput									
	1	Name	Data type	Address	Retain	Acces	Writa	Visibl	Comment		
1	-00	mass_1	Real	%ID68							
2	-	tare_1	Real	%ID72							
3	-00	unit_1	Word	%IW76							
4	-	status_1	Word	%IW78							
5	-	LO_1	Real	%ID80							
6	-0	mass_2	Real	%ID84							
7	-	tare_2	Real	%ID88							
8	-0	unit_2	Word	%IW92							
9	-	status_2	Word	%IW94							
10	-0	LO_2	Real	%ID96							
11	-	inputs	Word	%IW134							
12	-00	min	Real	%ID136							
13	-	max	Real	%ID140							
14	-	lot number	DWord	%ID152							
15	-	operator	Word	%IW156							
16	-	article	Word	%IW158							
17	-	customer	Word	%IW160							
18	-	packaging	Word	%IW162							
19	-	recipe	Word	%IW168							
20	-	dosing	Word	%IW170							

In order not to operate using module's physical inputs/outputs create data blocks containing representations of these registers and functions rewriting values between them. Create HARDWARE group in the PROGRAM BLOCKS branch and determine data blocks as below:



HD_OUTPUT and HD_INPUT blocks represent physical input/output registers of the MASTER controller and are not relevant in this project. HD_ProfinetOutput and HD_ProfinetInput blocks represent input/output registers of the scale's PROFINET module. See the screenshots below:

Ż			ep actual values 🔒 Sr	napshot 🖳 🖳	Copysnap	shots to start va	lues 🖳	E. Load	start values	as actual values	W, U,	-
		rofinetInput										
	Na		Data type	Start value	Retain	Accessible f	Writa	Visible in	Setpoint	Comment		
1	-											
2		mass_1	Real	0.0								
		tare_1	Real	0.0								
		unit_1	Word	16#0								
		status_1	Word	16#0								
		lo_1	Real	0.0								
	-00 =	mass_2	Real	0.0								
		tare_2	Real	0.0								
		unit_2	Word	16#0								
0	-00 =	status_2	Word	16#0								
1		lo_2	Real	0.0								
2		inputs	Word	16#0								
3	-01 =	min	Real	16#0								
4		max	Real	0.0								
5		lot numbet	DWord	16#444								
6		operator	Word	16#0								
7		article	Word	16#0	Ē							
8		customer	Word	16#0	Ē				Ā			
		packaging	Word	16#0	Ä				Ē			
		recipe	Word	16#0	Ä				Ä			
		dosing	Word	16#0	Ä				Ä			

RadwagProfinetHX7 ang 🔸 PLC_1 [CPU 1214C DC/DC/DC] + Program blocks + Hardware + SaveOutput + HD_ProfinetOutput [DB4]								_∎≡>				
÷	1		🛃 🔁 😤 Keep a	actual values 🔒	Snapshot 🔤 🛤	, Copysnap	hots to start va	lues 🖁	E. Load	start values	as actual values 🛛 🖳 🕮	1
	HD.	Prot	finetOutput									
		Name	•	Data type	Start value	Retain	Accessible f	Writa	Visible in	Setpoint	Comment	
	-	▼ S	tatic									
	-	•	command	Word	16#0000							
	-	•	complex command	Word	16#0000							
	-	•	platform	Word	16#01							
	-	•	set tare	Real	2.0							
	-	•	set lo	Real	4.0							
	-	•	outputs	Word	16#0002							
	-	•	set min	Real	2.2							
	-	•	set max	Real	3.3							
0	-	•	set lot number	DWord	444							
1	-	•	set operator	Word	12							
2	-	•	set article	Word	2							
3	-	•	set customer	Word	16#0							
4	-	•	set packaging	Word	16#0							
5	-	•	set recipe	Word	16#44							
6	-0		set dosing	Word	16#0							
7			<add new=""></add>									

Functions rewriting the values between module's physical inputs/outputs may look as presented below.

ageronnetHX7 ang V PLC_1 [CPU 1214C DODODC] V Program blocks V 1		
🖉 🖻 🔍 🗄 🔚 🛛 😰 🕼 🕪 ᅇ 🚱 🖉 🕼 🐨 😵 🖕 🖬 🖷 🏦 🖕 🍾	🛤 📢 🔗 🖤 🙄 🔒	
D_SaveProfinetInput		
Name Data type Default value Comment		
Input ■ Data type ■ Detail voide ■ comment		
 Output 		
- CASE FOR WHILE (**) REGION		
<pre>1 "HD_ProfinetInput".mass_1 := "mass_1";</pre>	"HD_ProfinetInput"	\$DB3
<pre>2 "HD_ProfinetInput".tare_1 := "tare_1";</pre>	HD_ProfinetInput"	%DB3
3 "HD ProfinetInput".unit 1 := "unit 1";	HD ProfinetInput"	\$DB3
4 "HD ProfinetInput".status 1 := "status 1";	HD ProfinetInput"	&DB3
<pre>5 "HD_ProfinetInput".lo_1 := "LO_1";</pre>	"HD ProfinetInput"	DB3
6 "HD_ProfinetInput".mass_2 := "mass_2";	"HD_ProfinetInput"	%DB3
7 "HD_ProfinetInput".tare_2 := "tare_2";	"HD ProfinetInput"	\$DB3
8 "HD ProfinetInput".unit 2 := "unit 2";	"HD ProfinetInput"	\$DB3
9 "HD ProfinetInput".status 2 := "status 2";	"HD_ProfinetInput"	\$DB3
<pre>10 "HD_ProfinetInput".lo_2 := "L0_2";</pre>	"HD_ProfinetInput"	%DB3
<pre>11 "HD_ProfinetInput".inputs := "inputs";</pre>	"HD_ProfinetInput"	\$DB3
<pre>12 "HD_ProfinetInput".min := "min";</pre>	"HD_ProfinetInput"	\$DB3
13 "HD_ProfinetInput".max := "max";	"HD_ProfinetInput"	DB3
14 "HD_ProfinetInput"."lot numbet" :="lot number";	"HD_ProfinetInput"	%DB3
15 "HD_ProfinetInput".operator:= "operator";	"HD_ProfinetInput"	\$DB3
HD_ProfinetInput".article := "article";	"HD_ProfinetInput"	\$DB3
<pre>17 "HD_ProfinetInput".customer := "customer";</pre>	"HD_ProfinetInput"	<pre>%DB3</pre>
18 "HD_ProfinetInput".packaging := "packaging";	"HD_ProfinetInput"	%DB3
<pre>19 "HD_ProfinetInput".recipe := "recipe";</pre>	"HD_ProfinetInput"	\$DB3
20 "HD ProfinetInput".dosing := "dosing";	"HD_ProfinetInput"	\$DB3

RadwagProfinetHX7 ang > PLC_1 [CPU 1214C DC/DC/DC] > Program blocks > Hardware > SaveOutput > HD_SaveProfinetOutput [FC4]

1 [CDU 1214C DC/DC/DC] > D---

_ # #×

) # 핵심 🗄 🔠 월 🕹 😥 🧐 😡 🕾 📾 🗣 📭 🖬 🖬 🐂 🐂 📢 🌾	<u></u>		1
HD_SaveProfinetOutput			
Name Data type Default value Comment			
a v Input			-
Add new>			
 ✓ Output 			
■			
0F TO DO DO 1 1 LUION			
<pre>1 "cammand" := "HD_ProfinetOutput".command;</pre>	•	"cammand"	\$QW64
2 "complex command" := "HD_ProfinetOutput"."complex command"	; F	"complex command"	\$QW66
3 "set platform" := "HD_ProfinetOutput".platform;	•	"set platform"	%QW68
4 "set tare" := "HD_ProfinetOutput"."set tare";	•	"set tare"	%QD70
5 "set lo":="HD_ProfinetOutput"."set lo";	•	"set lo"	%QD74
6 "outputs" := "HD_ProfinetOutput".outputs;	•	"outputs"	\$QW78
7 "set min" := "HD_ProfinetOutput"."set min";	•	"set min"	%QD80
8 "set max" := "HD_ProfinetOutput"."set max";	•	"set max"	%QD84
9 "set lot number" := "HD_ProfinetOutput"."set lot number";	•	"set lot number"	\$QD96
10 "set operator" := "HD_ProfinetOutput"."set operator";	•	"set operator"	\$QW100
<pre>11 "set article" := "HD ProfinetOutput"."set article";</pre>	•	"set article"	%QW102
12 "set customer":= "HD_ProfinetOutput"."set customer";	•	"set customer"	%QW104
13 "set packag %OW104 / Word finetOutput"."set packaging";	•	"set packaging"	\$QW106
14 "set recipe := "HD_ProrinetOutput"."set recipe";	•	"set recipe"	\$QW112
15 "set dosing" := "HD ProfinetOutput"."set dosing";	•	"set dosing"	%QW114

Now, in the main program loop, trigger required functions.

RadwagProfinetSampleSCL1.2 (EX) → PLC_1 [CPU 1214C DC/DC/DC] → Program blocks → Hardware → Main [OB1]										
(2) (2) 안 10 (2) 한 10 (2) (2) (2) (2) (2) (2) (2) (2) (2) (2)										
	Main									
		Na	me	Data type	Default value	Comment				
1	-	٠	Input							
2	-	•	Initial_Call	Bool		Initial call of this OB				
3	-01	•	Remanence	Bool		=True, if remanent data a				
4	-	•	Temp							
5			<add new=""></add>							
6	-	٠	Constant							
IF CASE FOR WHILE (*) REGION										
		1	"HD_SaveInput"();				"HD SaveInput"		8FC1	
	<pre>2 "HD_SaveOutput"();</pre>						"HD_SaveOutput" %FC2			
	3 "HD_SaveProfinetInput"();						"HD_SaveProfinetInput" %FC3			
	4 "HD_SaveProfinetOutput"(); 5						"HD_SaveProfinetOu	tput"	%FC4	

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.



