

MW-01-A Mass Converter

SOFTWARE MANUAL

ITKP-40-01-12-21-EN



DECEMBER 2021

CONTENTS

2. DATA STRUCTURE 2.1. Input Address 2.2.1 Input Registers 2.2. Output Address 2.2.1 Output Registers	
2.1. Input Address 2.2.1 Input Registers 2.2. Output Address 2.2.1 Output Registers	4
2.2.1 Input Registers 2.2. Output Address 2.2.1 Output Registers	4
2.2. Output Address	4
2.2.1 Output Registers	7
	7
3. CONFIGURATION OF THE PROFINET MODULE IN TIA PORTAL V16 ENVIRONMENT	10
3.1. Import GSD	10
3.2. Module Configuration	12
4. DIAGNOSTICS APP	16

1. MASS CONVERTER SETTINGS AND CONFIGURATION

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

2. DATA STRUCTURE

All registers are 2-byte type (WORD). Floating point data (such as mass and tare) are stored in two consecutive registers and are FLOAT-type data. If the first register consists of two AB bytes and the other of two CD bytes, then FLOAT value is HEX ABCD,

2.1. Input Address

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	2	float
Adjustment status	100	1	word

Input variables:

2.2.1 Input Registers

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

Example:

The read register with offset 0 has a hex value of 0x43E28000, after converting to float we get 453.0 which is the current indication of the load mass.

Platform tare – returns platform tare in an adjustment unit.

Platform unit – determines a current (displayed) net mass unit of the 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 mass unit is kilogram [kg].

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

Status	s bits
0	Measurement correct (the scale does not report any error)
1	Measurement stable
2	Scale indicates zero
3	Scale tared
4	Scale in II weighing range
5	Scale in III weighing range
6	Scale reports NULL error
7	Scale reports LH error
8	Scale 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 scale does not report any error, measurement stable in weighing range II.

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

Process status – determines status of the dosing\formulation process:

HEX value	Description
0x00	Process disabled
0x01	Process started
0x02	Process aborted
0x03	Process finished

<u>Input state</u> – bitmask of mass converter inputs. Three least significant bits represent the mass converter inputs state.

Example:

Read HEX value: 0x0005

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

Inputs number 1 and 3 of the weighing terminal take HI state.

MIN – returns MIN threshold value in an adjustment unit.

MAX – returns MAX threshold value in an adjustment unit.

<u>Fast dosing threshold</u> - returns fast dosing threshold value in an adjustment unit.

<u>Slow dosing threshold</u> - returns slow dosing threshold value in an adjustment unit.

Adjustment status - determines adjustment process status:

HEX value	Description
0x00	Process finished correctly
0x01	Start mass/adjustment coefficient determination in progress
0x02	Range exceeded
0x03	Time exceeded
0x04	Process aborted
0x05	Awaiting for data

2.2. Output Address

Output variables list:

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

2.2.1 Output Registers

<u>Basic command</u> – writing a respective value triggers direct performance of a given task, see the table:

Bit No.	Operation
0	Zero platform
1	Tare platform
5	Process start
6	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 (with parameter)</u> – setting a respective value triggers performance of a given task, see the table:

Bit No.	Operation
0	Setting tare value for a platform
1	Setting LO threshold value for a platform
2	Setting outputs state
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 28 – refer to output registers table)

~
(\mathbf{I})

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 to the scale.

Performance of the command requires writing of 2 registers:

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

offset 6 - tare value in float format - 1.0.

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: state of mass converter outputs

Example:

Setting high state for output 1 and 3 of the mass converter.

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 0x04 - i.e. output state record,

offset 14 - output mask 0x05.

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

 $\underline{\text{MIN}}$ – complex command parameter: MIN threshold value (in an adjustment unit).

 $\underline{\text{MAX}}$ – complex command parameter: MAX threshold value (in an adjustment unit).

<u>Fast dosing threshold</u> - complex command parameter: fast dosing threshold value (in adjustment unit).

<u>Slow dosing threshold</u> - complex command parameter: fine dosing threshold value (in adjustment unit).

3. CONFIGURATION OF THE PROFINET MODULE IN TIA PORTAL V16 ENVIRONMENT

Start operation in the environment by setting up a new project, where topology of PROFINET 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 to the environment.

Open OPTIONS tab, next click MANAGE GENERAL STATION DESCRIPTION FILES (GSD) entry and select access path to the GSD file.

Manage general sta	tion description files		×
Installed GSDs	GSDs in the project		
Source path: D:\	ProiektvlTiaPortallGSD_ProfinetlGSDML-V2	2-3-HMS-ABICPRT-20130219	
Content of import	← → < ↑ 🔒 « TiaPor > G	SD_Profinet ~ 진	, Przeszukaj: G
GSDML-V2.3-HMS	Organizuj 👻 Nowy folder		
	Nazwa	Data modyfikacji	Тур
	GSDML-V2-3-HMS-ABICPRT-2013	0219 05.11.2021 13:09	Folder plików
	<		
	Folder: GSDML	-V2-3-HMS-ABICPRT-20130219	
<			Wybierz folder
		Delete Insta	Cancel

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

arch in project> 🛛 🔒					Totally Integrated	Automation PORTA
			_ 7	∎×	Hardware catalog	a 🗉 🕨
Topology view	h Netw	ork view	Device vie	w	Options	
Network overview	v Co	nnections		• •		6
Y Device		Туре		A	✓ Catalog	
▼ \$7-1200 sta	tion_1	\$7-12	00 station		<search></search>	féi léi
PLC_1		CPU 1	211C AC/DC/Rly			
 GSD device 	_1	GSD	device		ritter riolite.	
ABIC-PRT		RT Mig	gration (FW>=2.00)	Controllers	
					PC systems	
					Drives & starters	
					Detection & Manitaria	
					Detecting & Monitoring	
					Distributed no	*ia.a
					Fower supply and distribut	luon
					Tield devices	
					Additional Ethernet de	vicer
						VICES
					Driver	
					Encodorr	
					Cateway	
					T General	
					HMS Industrial N	etworks
					Anybus-IC PRT	CUNOIRS
					T Migration	
					RT Migra	tion (FW 1 xx)
	2002				RT Migra	tion (FW>=2.00)
N	m			/	RT Standar	d
Q Properties	🔄 Info	🖁 Diagn	ostics		► I I/O	
					> Sensors	

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

Project_08_Profinet_MW_01_New_instrukcja → Devices & networks										
			-	^a Topology view						
Network	ections HMI connection	- E =	. 🔳 🖽 🛄 🍳	±						
PLC 1		ABIC-PRT	_							
CPU 12	11C	RT Migration (F	DP-NORM							
		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 (MW-01-A mass converter). With the power supply on, you can use ACCESSIBLE DEVICES function to search for a given device in the environment. The list should display both the MASTER and the SLAVE.

		Type of the PG/PC in	terface: 📜 PN/	IE		
		PG/PC in	terface: 💹 Rea	ltek PCIe GbE Family	Controller	• 💎 🖸
	Accessible nodes	of the selected interface	e:			
	Device	Device type	Interface type	Address	MAC address	
	pawelk	SIMATIC-PC	PN/IE	10.10.3.145	4C-ED-FB-44-CO-31	
	mw01-profinet	RT Migration (FW 1.xx)	PN/IE	10.10.8.100	00-30-11-34-44-E6	
	plc_1	CPU 1211C AC/DC/Rly	PN/IE	10.10.8.222	E0-DC-A0-CF-59-E0	
	tk-kacperczyk-m	SIMATIC-PC	PN/IE	10.10.22.21	F8-32-E4-A0-BF-29	
Flash LED						
Online status informatio	n:			🗌 Display	vonlyerror messages	art search
Found accessible d	evice mw01-profinet				, ,	
Scan completed. 4	devices found.					
? Retrieving device in	formation					1
Scan and information	on retrieval complete	d.				

Next, specify the IP address of the module and its name in PROFINET network. To do it, select the module, in PROPERTIES tab find PROFINET INTERFACE submenu, click it and enter the IP address and name. Make sure that these settings are accordant with parameters set in the MW-01 mass converter. Remember that the SLAVE IP address and MASTER address must come from the same subnet.

ABIC-PRT [RT Migration (FW>=	2.00)]	🖾 Properties 🚺 Info 😨 Diagnostics 📑 🖃	•
General IO tags Sys	tem constants Texts		
▼ General		Add new subnet	^
Catalog information			
➡ PROFINET interface [X1]	IP protocol		
General			
Ethernet addresses		Set IP address in the project	
 Advanced options 		IP address: 10 . 10 . 8 . 100	≣
 Real time settings 		Subnet mask: 255 255 255 0	
IO cycle		Synchronize router settings with 10 controller	
Identification & Maintenance			
1 .		Use router	
		Router address: 0 . 0 . 0 . 0	
		O IP address is set directly at the device	
	PROFINET		
		Generate PROFINET device name automatically	
	PROFINET device name:	MW01-ProfiNet	
	Converted name:	mw01-profinet	
	Device number:	1	
			~

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 102 bytes, maximum size of output data is 52 bytes. The project uses the default starting addresses: 256 for INPUT and 256 for OUTPUT with a size of 102 bytes for input data and 32 bytes for output data, as presented below.

roject_08_Profinet_MW_01_New → Ungrouped devices → ABIC-PRT [RT Migration (FW>=2.00)]										
		🛃 To	pology view 🔒 🖁	Netwo	rk vie	w 🚺 D	evice viev	w		
🟕 ABIC-PRT [RT Migration (FW>=2 💌 🔡 🌃 🔛 🛄 🔍 🛨		Device ov	erview							
A	^		dule	Rack	Slot	I address	Q address			
SF.		-	ABIC-PRT	0	0			^		
ABIC	_		Interface	0	0 X1					
•			Input 064 bytes_1	0	1	256319		=		
			Input 032 bytes_1	0	2	320351				
	•		Input 004 bytes_1	0	3	352355				
			Input 002 bytes_1	0	4	356357				
DP-NORM	•		Output 032 bytes_1	0	5		256287			
	_			0	6					
				0	7					
				0	8					
				0	9					
				0	10			~		
< III > 100% •	- 7	<	111	0	11			>		



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

> Informati

Name

Portal view

ABIC-PRT

ProfiNET_Inp...

Project_08_Profinet_	MW_01_New ➤ Devices & networks		_ 🖬 🖬 X
		🛃 Topology view 🛛 🎄 Network view 👔	Device view
Network 1 Connec	tions HMI connection 💌 🛍 🐫 📲	1 🗊 🔍 ±	
			^
			=
PLC_1	ABIC-PRT		
CPU 121	1C RT Migration (F Dr	P-NORM	7
	PLC 1		etw
	Change device		, i i i i i i i i i i i i i i i i i i i
	🗶 Cut Ctrl+:	x	- 13
	Copy Ctrl+1	c	
		×	
	Rename F	2	
	Assign to new DP master / IQ controller		~
<	Disconnect from DP master system / IO system	> 100%	-
	Highlight DP master system / IO system	Roperties Info Diagnostic	s 🛛 🖛 🗸
General Cross-	🚽 Go to topology view		
🕄 🚹 🚺 Show all m	Compile	•	
	Download to device	Hardware and software (only changes)	
! Message	Go offline Ctrl+N	M Software (only changes)	
The project Proje	🖳 Online & diagnostics Ctrl+I	D Software (all)	^
Connection to PL	Assign device name		_
Start downloadin	Update and display forced operands	11/5/2021 2:32:52 PM 11/5/2021 2:32:55 PM	_
✓ TLC_1 ✓ Hardware	Show catalog Ctrl+Shift+	C 11/5/2021 2:32:55 PM	
PLC_1:	Export module labeling strips	11/5/2021 2:32:57 PM	
Y Hardwa	Roperties Alt+Ente	11/5/2021 2:33:01 PM	

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:



PROFINET input and output registers are specified in TAG_ProfiNET_IN and TAG_ProfiNET_OUT tables. The below screenshots present symbolic names and addresses:

Proje	roject_08_Profinet_MW_01_New_instrukcja → PLC_1 [CPU 1211C AC/DC/Rly] → PLC tags → TAG_ProfiNET_IN [12] _ 🗕 🖬 🗮 🗙											
										🕣 Tags	User const	ants
\$	P 👻 🖻 🗄 🕫 🛍											
Т	TAG_ProfiNET_IN											
-		Name	Data type	Address	Retain	Acces	Writa	Visibl	Comment			
1	-	AnyBus_Platform_mass	DWord	%ID256		Image: A start and a start						^
2	-0	AnyBus_Platform_tare	DWord	%ID260		Image: A start and a start						
3	-	AnyBus_Platform_unit	Word	%IW264		Image: A start and a start						
4	-0	AnyBus_Platform_status	Word	%IW266								=
5	-	AnyBus_Platform_LO_threshold	DWord	%ID268		Image: A start of the start						
6	-	AnyBus_Process_status	Word	%IW320		Image: A start of the start						
7	-	AnyBus_Inputs_status	Word	%IW322		Image: A start and a start						
8	-0	AnyBus_Min	DWord	%ID324								
9	-	AnyBus_Max	DWord	%ID328		Image: A start and a start						
10	-0	AnyBus_Fast_dosing_threshold	DWord	%ID332								
11	-	AnyBus_Slow_dosing_threshold	DWord	%ID336								
12	-	AnyBus_Adjustment_status	Word	%IW356								~

Proje	ect_	08_Profinet_MW_01_New_ins	trukcja 🕨 PLC_	1 [CPU 1211C /	AC/DC/RI	y] ► PLC	tags 🕨	TAG_I	ProfiNET_OUT [9]		_∎≡×
										🕣 Tags	User constants
# :											
T	TAG_ProfiNET_OUT										
		Name	Data type	Address	Retain	Acces	Writa	Visibl	Comment		
1	-	AnyBus_Command	Word 🔳	%QW256	-						
2	-	AnyBus_Command_with_para	Word	%QW258		Image: A start and a start					
3	-	AnyBus_Tare	DWord	%QD262		Image: A start of the start					
4	-	AnyBus_LO_threshold	DWord	%QD266		Image: A start of the start	\sim				
5	-	AnyBus_Output_state	Word	%QW270							
6	-	AnyBus_Set_Min	DWord	%QD272							
7	-00	AnyBus_Set_Max	DWord	%QD276							
8	-	AnyBus_Set_Fast_dosing_thres	DWord	%QD280			\checkmark				
9	-	AnyBus_Set_Slow_dosing_thre	DWord	%QD284							
10		<add new=""></add>				V	 Image: A start of the start of	V			

So as not to work directly on the physical inputs/outputs of the module, it is recommended to create data blocks comprising representations of these registers, and "copy" values between them. For this purpose, define data blocks as follows:

Project tree	
Devices	
Ť	
Name	
🔻 🙀 Program blocks	^
📑 Add new block	
📲 Main [OB1]	
ProfiNET_Input [DB1]	
ProfiNET_Output [DB2]	
Technology objects	=
External source files	
🕨 🚂 PLC tags	
PLC data types	
Watch and force tables	
🕨 📴 Online backups	
🕨 🔛 Traces	
DPC UA communication	
Device proxy data	
Program info	
PLC alarm text lists	
Local modules	
Distributed I/O	~

ProfiNET_Input and ProfiNET_Output blocks represent input/output registers of the MW-01-A mass converter's PROFINET module. See the screenshots below:

Project_08_Profinet_MW_01_New_instrukcja > PLC_1 [CPU 1211C AC/DC/Rly] > Program blocks > ProfiNET_Input [DB1] _ 🗖 🖬										_ = = ×		
😴 💱 🛤 👼 🛅 🤷 Keep actual values 🔒 Snapshot 🏯 🦉 Copy snapshots to start values 😹 🚱 Load start values as actual values 💐 🕸												
ProfiNET_Input												
	Na	me	Data type	Offset	Start valu	ue Retain	Accessible f	Writa	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 🔩	•	platform _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 🖪	•	fast_dosing_threshold	Real	28.0	0.0							
12 🖪	•	slow_dosing_threshold	Real	32.0	0.0							
13 ◄	•	adjustment_status	Word	36.0	16#0		\checkmark					
									>			

Project_08_Profinet_MW_01_New_instrukcja > PLC_1 [CPU 1211C AC/DC/Rly] > Program blocks > ProfiNET_Output [DB2] 🗕 🖬 🖬									_ = = ×			
📝 🐏 🍓 👺 🗮 😤 🔚 🖤 Keep actual values 🔒 Snapshot 🦄 🦉 Copysnapshots to start values 😹 🐼 Load start values as actual values 💐 🕸												
ProfiNET_Output (snapshot created: 7/23/2021 11:03:40 AM)												
	N	lame	Data type	Offset	Start value	Retain	Accessible f	Writa	Visible in	Setpoint	Comment	
1 🖪	•	Static										
2 🔸		Command	Word	0.0	16#0							
3 \prec		Command_with_parameter	Word	2.0	16#0							
4 \prec	•	Platform	Word	4.0	16#0							
5 \prec	•	Tare	DWord	6.0	16#0							
6 🖪	•	LO_threshold	DWord	10.0	16#0							
7 🔩	•	Output_state	Word	14.0	16#0							
8 ◀	•	Min	DWord	16.0	16#0							
9 🔩	•	Max	DWord	20.0	16#0							
10 🔩	•	Fast_dosing_threshold	DWord	24.0	16#0							
11 🖪	•	Slow_dosing_threshold	DWord	28.0	16#0							
12 🛃	•	Adjustment_weight_mass	DWord	32.0	16#0							
	< m 12									>		

A MOVE instruction, for example, can be used to rewrite values between the physical inputs/outputs of the module and the registers in the data blocks:

Project_08_Profinet_MW_01_New + PLC_1 [CPU 1211C AC/DC/Rly] + Program blocks + Main [OB1]		_ # =×
	P 🔐	
Block interface		
- ト - / ト - → 1 ADD		
▼ Block title: "Main Program Sweep (Cycle)"		^
Comment		
Network 1: mass		-
Comment		
100256 *Anglous_ Platform_mass* N 4 OUTI - mass		
Network 2: command		_
Comment MOVE EN EN *Command *Anglus Command N		×
	100%	

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 changing START VALUE and LOAD START VALUES AS ACTUAL) of the SLAVE module.



