

Communication Protocol: MW-01-A Mass Converter

SOFTWARE MANUAL

ITKP-42-01-11-21-EN



NOVEMBER 2021

CONTENTS

1.	MASS CONVERTER SETTINGS AND CONFIGURATION	4
2.	DATA STRUCTURE	4
	2.1. Input Address	4
	2.2. Input Registers	4
	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	10
	3.2. Module Configuration	12
4.	DIAGNOSTICS APP	14

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.

<u>Platform unit</u> – 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].

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

Statu	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: 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.

<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.

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
Мах	20	2	float
Fast dosing threshold	24	2	float
Slow dosing threshold	28	1	float

2.4. Output Registers

<u>Basic command</u> – 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).

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 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.

 \underline{MIN} – complex command parameter: MIN threshold value (in the unit set for the active working mode).

 \underline{MAX} – complex command parameter: MAX threshold value (in the unit set for the active working mode).

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

<u>Slow dosing threshold</u> - 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.

Manage general station	description files
manage general station	
Source path: C:\Users	user/Downloads\RadwagProfibus3.5_V13_SP1_EX\AdditionalFiles\GSD
	Przeglądanie w poszukiwaniu folderu X
Content of imported p	
File	Info
hms_1810.gsd	
	E System
	\mu тмр
	UserFiles
	RadwagProfibus3.5_V13_SP1EX.backup
	T imatic
<	OK Anuluj >
	Delete Install Cancel

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

						х	Hardware catalog	
-	Topology view 🛛 🛔 Net	work view	I Y I	Device vi	iew		Options	
	Network overview	Connectio	ons]	•	•		
^	Device		Type				✓ Catalog	
	S7300/ET200M st	tation 1	57300/	ET200Mst	ation	,	<search></search>	ini init
	✓ PLC 1		CPU 31	3C-2 DP				
≡	GSD device 1		GSD de	vice			Filter	
	MW-01 Profib	us	Anybus	-IC PDP			Controllers	
			,				▶ 📺 HMI	
							C systems	
							Im Drives & starters	
							Inetwork components	
							Detecting & Monitoring	
							Distributed I/O	
							Power Supplies	
							Im Field devices	
							AS-Interface	
							Commanding and signaling devices	
							Im SIPLUS HCS	
	•						Other field devices	
							 Drives Encoder 	
							Catavara	
							Gateways	
							General	
							Anybus IC BDB	
							Anybus-IC PDP	
							SIEMENS AG	
							Ident systems	

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

RadwagProfibus_MW-01V13_SP1 → Devic	es & networks	
		📱 Topology view
Network Connections HMI connection		🛛 🔍 🛨 🔤
PLC_1 CPU 313C-2 DP	5_1)	
	WW-01 Profibur	
	Anybus-IC PDP	P-NORM

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.

RadwagProfibus_M	W-01V13_SP1	PLC_1 [CPU 313C-2 DP]	Distributed I/O	DP-Mastersystem (1): PROFIBI	US_1 ▶ MW-01_I
					2
MW-01_Profibus	•	🖽 🖭 🍊 🖽 🔍 ±			=
	WOI Potous	,			
	[]	DP-NORM			
					~
<				> 100% -	

MW-01_Profibus [Module]								
General IO tags Sys	stem constants Texts							
✓ General Catalog information	PROFIBUS address							
PROFIBUS address	Interface networked with	orked with						
General DP parameters Watchdog	Subnet:	PROFIBUS_1						
SYNC/FREEZE Diagnostics addresses		Add new subnet						
	Parameters							
	Address:	1						
	Highest address:	126						
	Transmission speed:	1.5 Mbps						

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:

RadwagProfibus_MV	V-01V13_SP1 → PLC_1 [CPU 313C-2 DP]	Dist	tribute	d I/O I	DP-Mastersystem (1): PROF	BUS_	1 → MV	V-01_Profibus	5		_ # = ×
								🚽 Topolo	gy view	🔒 Network view	Device view
MW-01_Profibus	💌 🚍 🖾 🍕 🗄 🍳 🗉		4 🗊	Devic	e overview						
			_	**	Module	Rack	Slot	I address	Q address	Туре	Article no.
	1000				MW-01_Profibus	0	0	1022*		Anybus-IC PDP	
	A Pro-		_		INPUT: 32 Byte (16 word)_1	0	1	256287		INPUT: 32 Byte (1	
	10 ¹⁰		=	~	INPUT: 32 Byte (16 word)_2	0	2	288319		INPUT: 32 Byte (1	
	N. Contraction of the second s			~	INPUT: 16 Byte (8 word)_1	0	3	320335		INPUT: 16 Byte (8	
					INPUT: 4 Byte (2 word)_1	0	4	336339		INPUT: 4 Byte (2	
					OUTPUT: 32 Byte (16 word)	0	5		256287	OUTPUT: 32 Byte (
						0	6				
						0	7				
	DD NODM					0	8				
						0	9				
						0	10				
	•					0	11				
						0	12				
			•			0	13				
						0	14				
						0	15				

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:

Ra	dwa	agProf	ibus_MW-01V13_SP1	I ▶ PLC_1 [CPU 31	3C-2 DP]	Program blo	cks ▶ Har	dware 🕨 S	aveInput	HD_ProfibusInput [DB2]				
2	1	۶ 🎝	B IR B B B I	s 🖿 🔢 😤										
	HD_ProfibusInput													
		Name		Data type	Offset	Start value	Retain	Visible in	Setpoint	Comment				
1	-	▼ Sta	itic											
2	-	•	mass	Real	0.0	0.0								
3	-00	•	tare	Real	4.0	0.0								
4	-	•	unit	Word	8.0	16#0								
5	-	•	status	Word	10.0	16#0								
6	-00	•	lo	Real	12.0	0.0								
7	-	•	process_status	Word	16.0	16#0								
8	-00	•	inputs	Word	18.0	16#0								
9	-00	•	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								

						aveoutput	, up_uoupar [pp3]						
2 2 4 5 1 1 2 5 5 5 E 1 1 2 ⁹													
HD_ProfibusOutput													
me	Data type	Offset	Start value	Retain	Visible in	Setpoint	Comment						
Static													
command	Word	0.0	16#00										
complex_command	Word	2.0	16#00										
set_tare	Real	4.0	2.0										
set_lo	Real	8.0	0.5										
outputs	Word	12.0	16#03	\checkmark									
set_min	Real	14.0	10.0	\sim									
set_max	Real	18.0	20.0										
set_bulk_dosing_thre	Real	22.0	10.0										
set_fine_dosing_thres	Real 🔳	26.0	20.0										
r	IP	Image: set_fine_dosing_three Image: set_fine_dosing_three Image: set_fine_dosing_three	Image Data type Offset Static 000 000 command Word 0.0 set_tare Real 4.0 set_lo Real 8.0 outputs Word 12.0 set_max Real 14.0 set_buik_dosing_thre Real 18.0 set_fine_dosing_thre Real 22.0 set_fine_dosing_thre Real 12.0	Image: Second	Image: Second	Image: Second	Image: Construction of the set of t						

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

Rad	wag	Profibus	_MW-01V13_SP1	PLC_1 [CPU	313C-2 DP] 🕨 F	Program b	olocks	Hardware	SaveOutput	→ HD_Pro	ofibusOutp	utTemp [DB1]
*	2	🔩 🋃	N R. R. D. B) 🖿 🔢 °								
	HD_	Profibus	OutputTemp									
	N	lame			Data type		Offset	Start value	Retain	Visible in	Setpoint	Comment
1	•	 Static 										
2	•	set_	tare_inv		Real		0.0	2.0				
з -	•	set_	lo_inv		Real		4.0	0.5				
4	•	out	outs_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_thresho	old_inv	DWord		18.0	16#DE				
8	•	set_	_fine_dosing_thresho	ld_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_SP	1 PLC_1 [CPU 313C-2 DP]	 Program blocks > OB1 [OB1] 		_ # = X
		·원 GH 4D2 1_ 1 이야 [10]		
Name	Data type Offset	Default value Comment		
1 - 1 - Temp	ona ype	connent		~
2 - 1 Temp 0	Byte 🔠 0.0			=
3 - 1 = Temp 1	Byte 1.0			
	-			
CALL				
-				~
1 CALL DPI	RD DAT			-
2 LADDR	:=W#16#100		W#16#100	
3 RET VI	AL :="err read"		%MW4	
4 BECOBI	D :="HD ProfibusInput	".mass	\$DB2, DBD0	
5				
6				
7				
8				
9				
10				
11				
11				
 Network 3: 				
Commont				
Comment				
1 L "H	D ProfibusInput".mass		%DB2.DBD0	
2 CAD				
3 т "н	D ProfibusInput".mass		%DB2.DBD0	
4				
5				
RadwagProfibus_MW-01V13_SP	1 → PLC_1 [CPU 313C-2 DP]	▶ Program blocks OB1 [OB1]		_ # #
RadwagProfibus_MW-01V13_SP 成 kX 양 왕 등 등 등 등 등	1 → PLC_1 [CPU 313C-2 DP]	Program blocks OB1 [OB1] Set I = 1 = 0, ∞ II		- 7 = 2
RadwagProfibus_MW-01V13_SP (상 , K) 관 관 문 등 등 등 0B1	1 → PLC_1 [CPU 313C-2 DP]	> Program blocks > OB1 [OB1] 5 4		-**
RadwagProfibus_MW-01V13_SP ⊮& ⊮X ⊉ ⊉ ∎₀ ⋿ ⊟ ₪ OB1 Name	1 → PLC_1 [CPU 313C-2 DP]	 > Program blocks → OB1 [OB1] 5		- * =
RadwagProfibus_MW-01V13_SP k¾ k¾ ∰ ∰ ∰ ∰ ∰ ∰ ∰ Name 1	1 → PLC_1 [CPU 313C-2 DP]	Program blocks → OB1 [OB1] C C C C C C C		- # = 3
RadwagProfibus_MV-01V13_SP ki kX 20 20 100 100 100 OB1 Name 1 2 2 1 100 Temp 2 2 1 100 Temp_0	1 → PLC_1 [CPU 313C-2 DP] 2 ± ± ± = 2 ¢° ¢ Data type Offset Byte ■ 0.0	Program blocks → OB1 [OB1] C		_ # = 2
RadwagProfibus_MW-01V13_SP → → → → → → → → → → → → → → → → → → →	1 → PLC_1 [CPU 313C-2 DP] 2 2 ± 2 ± 2 2 ↓ 2 0 € Data type Offset Byte	Program blocks > OB1 [OB1] cE ←		- # = -
RadwagProfibus_MV-01V13_SP µ@ µ% ☆ ☆ ☆ � � b 註 臣 ⊆ Name 1	1 → PLC_1 (CPU 313C-2 DP) □ 2 ± 2 ± 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	Program blocks → OB1 [OB1] Cell QB1 Cell QB1 Default value Comment		_ # #
RadwagProfibus_MV-01V13_SP → → → → → → → → → → → → → → → → → → →	1 → P.C_1 (CPU 313C-2 DP)	Program blocks > OB1 [OB1] Cell { Cell {		- # #
RadwagProfibus_MV-01V13_SP µ@ µ% ഈ ഈ ₪ 世 🖾 💭 OB1 Name 1 → Y Temp 2 → Temp_0 3 → Temp_1 CAL ▼ Network 6:	1 → PLC_1 [CPU 313C-2 DP]	Program blocks > OB1 [OB1] Cefault value Comment		_ # = _
RadwagProfibus_MV-01V13_SP with with with an end of the second secon	1 → PLC_1 (CPU 313C-2 DP)	Program blocks > OB1 [OB1] Centre of the second s		_ # #
RadwagProfibus_MV-01V13_SP → → → → → → → → → → → → → → → → → → →	1 → PLC_1 [CPU 313C2 DP]	Program blocks > OB1 [OB1] Cefault value Comment Comment		- # =
RadwagProfibus_MV-01V13_SP +	1 → PLC_1 (CPU 313C-2 DP)	Program blocks > OB1 [OB1] Cell QB1 Comment Default value Comment Comment		_ # =
RadwagProfibus_MV-01V13_SP → → → → → → → → → → → → → → → → → → →	1 → PLC_1 (CPU 313C-2 DP)	Program blocks > OB1 [OB1] Cef en Comment Default value Comment	w#16#108	
RadwaigProfibus_MV-01V13_SP µiii µX ⇒ ⇒ ⊕ ⊨ □ 0B1 Name □ Temp_0 2 • Temp_1 CAL ▼ Network 6: 1 CALL DP1 2 LADDR 3	1 → PLC_1 [CPU 313C-2 DP]	Program blocks > OB1 [OB1] Cefault value Comment	N#16#108 55074	_ # =
RadwagProfibus_MV-01V13_SP → W → W → W → W → W → W → W → W → W → W	1 → PLC_1 (CPU 313C-2 DP)	> Program blocks > OB1 [OB1] a @ @ @ @ la la @ @ @ U Default value Comment	₩#16#108 %WW4 %DD2.DEM8	- # #
RadwagProfibus_MW-01V13_SP will will will will will will will will	1 → PLC_1 [CPU 313C-2 DP]	Program blocks > OB1 [OB1] Comment Default value Comment	w#16#108 %WW4 %DB2.DBW8	_ # #
RadwagProfibus_MV-01V13_SP will will will will will will will will	1 → PLC_1 (CPU 313C-2 DP)	Program blocks > OB1 [OB1] • E	##16#108 %MM4 %DB2.DBW8	- • • =
RadwagProfibus_MW-01V13_SP + № № № № № № № № № 0B1 Name 1 • Temp_0 3 • Temp_1 CAL * Network 6: 1 CALL DPI 2 LADDR 3 RET_Y, 4 5 6 7 7	1 → PLC_1 [CPU 313C2 DP] P 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	Program blocks > OB1 [OB1] Cefault value Comment Comment Comment Comment Comment Comment	W#16#108 %004 %DB2.DBW8	
RadwagProfibus_MV-01V13_SP +	<pre>1 > PLC_1 [CPU 313C-2 DP] Data type Data type Data type Data type Data type Dot Data type Dot Data Type Dot Data Type Dot Type Dot Type Dot Type Type Dot Type Type</pre>	Program blocks > OB1 [OB1] o e and other a	N#16#108 \$xxv4 \$DB2.DBW8	_ # #
RadwagProfibus_MV-01V13_SP will will will will will will will will	<pre>1 → PLC_1 [CPU 313C2 DP] P 1 ± 1 ± = ↓ ℓ ° € Data type Offset Byte 0 00 Byte 0 0 Byte 10 Byte 10 ET=#16\$108 LL := "err read" D := "HD_ProfibusInput</pre>	Program blocks > OB1 [OB1] a @ @ @ 0 1= 1= 0: 0: 0: 0: 0: 0: 0: 0: 0: 0: 0: 0: 0:	W#16#108 %WW4 %DB2.DBW8	
RadwagProfibus MV-01V13_SP will will see the second	<pre>1 > PLC_1 [CPU 313C-2 DP] p 2 ± 2 ± 2 ± 2 2 2 2 2 2 2 2 2 2 2 2 2</pre>	Program blocks > OB1 [OB1] • Program blocks > OB1 [OB1] • E • • • • • • • • • • • • • • • • • •	W#16#108 \$xxv4 \$DB2.DBW8	_ # #
RadwagProfibus_MV-01V13_SP will will will will will will will will	1 → PLC_1 [CPU 313C2 DP] P 2 ± ± ± = 2 0 0° € Data type Offset Byte 0 00 Byte 10 RD_DAT :=#\$16\$108 AL :="err read" D :=""HD_ProfibusInput	Program blocks > OB1 [OB1] Comment Default value Comment	W#1C#108 %MM4 %DB2.DBW8	= = .
RadwagProfibus_MW-01V13_SP +% +% ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ ⇒	<pre>1 → PLC_1 [CPU 313C2 DP] p 2 ± 2 ± 1 = 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2</pre>	Program blocks > OB1 [OB1]	W#15#108 %XX4 %DB2.DBW8	
RadwagProfibus MV-01V13_SP +% +% ⇒ 0B1 > Neme1 > 1 > 2 • 1 > 2 • 1 Call 2 • 1 Call 2 • 1 Call 2 Comment 1 Call 2 Call 3 • 1 Call 2 Call 2 Call 3 • 1 Call 2 Call 1 Call	<pre>1 → PLC_1 [CPU 313C2 DP] p 2 ± 2 ± 1 = 1 = 2 e 2 e 2 e 2 e 2 e 2 e 2 e 2 e 2 e 2</pre>	Program blocks > OB1 [OB1] • E • • • • • • • • • • • • • • • • • •	##16#108 %XM4 %DB2.DBW8	_ # =
RadwagProfibus MV-01V13_SP will k% ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ ≡ ⇒ ≡ ⇒ ⇒ > OB1 Name ■ I Temp_0 > I Temp_1 Call I I Vetwork 6: Comment	<pre>1 → PLC_1 [CPU 313C-2 DP] P 1 ± 1 ± 1 = 1 = 2 C Q Deta type Deta type Byte Byte 0.0 Byte 1.0 RD_DAT :=##16#108 AL:=#"err read" D :=#"HD_ProfibusInput</pre>	Program blocks > OB1 [OB1] Image: Comment in the second	W#16#108 \$xW4 \$DB2.DBW8	
RadwagProfibus MV-01V13_SP + № ⇒ ⇒ 0B1 → Temp_0 2 • Temp_1 2 • Temp_1 CAL > Temp_1 CAL • 1 CALL DPI 2 • Temp_1 CAL • 2 • Temp_1 Comment CALL 2 • Temp_1 2 • Temp_1 Comment • 1 • CALL 9 10 • 10 11 • • Network 7: - 1 • •	1 → PLC_1 [CPU 313C2 DP] P 1 ± 1 ± 1 = 2 < 0 € Data type Offset Byte 0 00 Byte 0	Program blocks > OB1 [OB1] a @ @ @ @ 1 = 1 a @ @ @ U Default value Comment	¥#16#108 %NY4 %DS2.DSW8	
Radwaightofibus MV-01V13_SP Image: Second	<pre>1 > PLC_1 [CPU 313C2 DP] p 2 ± 2 ± 2 = 2 2 2 2 2 2 2 2 2 2 2 2 2 2</pre>	Program blocks > OB1 [OB1] • Program blocks > OB1 [OB1] • E • • • • • • • • • • • • • • • • • •	¥16#108 \$xm4 \$DB2.DBW8	
RadwagProfibus MV-01V13_SP Will will will will will will will will	<pre>1 → PLC_1 (CPU 313C2 DP) P 1 ± 1 ± = 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2</pre>	Program blocks > OB1 [OB1] a @ @ @ @ la la @ @ @ U Default value Comment	W#16#108 \$M074 \$D52.D5W8 \$D52.D5W8 \$D52.D5W8	
RadwagProfibus MW-01V13_SP will will will will will will will will	<pre>1 > PLC_1 [CPU 313C2 DP] p 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2</pre>	Program blocks > OB1 [OB1] • E • • • • • • • • • • • • • • • • • •	*DB2.DBW8 *DB2.DBW8	
RadwagProfibus MV-01V13_SP will kM => will kM => 0B1 will temp 0B1 will temp 2 Temp_0 2 Temp_1 Comment I 1 Call 1 Call 1 Call 1 Call 2 Image: Call 1 Call 2 Image: Call 1 Call 2 Image: Call 3 T 4 S 5 S	<pre>1 > PLC_1 [CPU 313C2 DP] p 1 ± 1 ± m 2 C Q Deta type Offset Byte 0 0 Byte 0 0 Byte 10 Byte 10 Byte 10 D = #164108 Lt := "err read" D := "HD_ProfibusInput".unit D_ProfibusInput".unit</pre>	Program blocks > OB1 [OB1] Program blocks > OB1 [OB1] Perfault value Comment Perfault value Comment	N#16#108 \$xxv4 \$DB2.DBW8 \$DB2.DBW8 \$DB2.DBW8	
RadwagProfibus_MW-01V13_SP will will will will will will will will	<pre>1 → PLC_1 [CPU 313C2 DP] P 1 ± 1 ± 1 = 2 < 4° € Data type Offset Byte B 00 Byte</pre>	Program blocks > OB1 [OB1] a E	*DB2.DBW8 *DB2.DBW8	

KX 🕾 🕾 🖦 🔚 🚍 🚍	, 2 ± 2 ± 1 1 0 € 6 6 6 9 1 1 1 6 % 1 1		
OB1	Detection Defectively Compared		
Name	Data type Offset Default value Comment		
II = Temp_0	Byte 🔳 0.0		
🖬 = Temp_1	Byte 1.0		
Network 27:			
Comment			
1 L "H	D_ProfibusOutput".set_tare	%DB3.DBD4	
2 CAD			
3 T "H	D_ProfibusOutputTemp".set_tare_inv	%DB1.DBD0	
5			
Network 28:			
Comment			
1 CALL DP	WR_DAT		
2 LADDR	:=W#16#106	W#16#106	
3 RECOR	D := "HD_ProfibusOutputTemp".set_tare_inv	\$DB1.DBD0	
5 KE1_V	AL :- err write	SDIN O	
6			
7			
8			
9			
10			
11			
11			
11			
11 wagProfibus_MW-01V13_SF	1 + PLC_1 [CPU 313C-2 DP] + Program blocks + OB1 [OB1]		_ (
11 wagProfibus_MW-01V13_SF	1 → PLC_1 [CPU 313C-2 DP] → Program blocks → OB1 [OB1]		
11 wagProfibus_MW-01V13_Sf	1 -> PLC_1 [CPU 313C2 DP] -> Program blocks -> OB1 [OB1] 중 월 : 골 : 프 등 및 산 등, 센 앤 장 1 = 1 등 유 약 및		- 1
11 wagProfibus_MW-01V13_SR kX ⊉ ⊉ № ⊨ ≘ ⊒ 281 Name	1 > PLC_1 [CPU 313C-2 DP] > Program blocks > OB1 [OB1]		- 1
11 wagProfibus_MW-01V13_Sf ,xX	1 → PLC_1 [CPU 313C-2 DP] → Program blocks → OB1 [OB1] 중 월호 월호 등 같은 도 ₀ 선택 대 장 I ₌ 1 ₌ 수 약 IV Data type Offset Default value Comment		- 1
11 wagProfibus_MW-01V13_SF kX ☞ ☞ 등 등 등 등 등 881 Name ☞ Temp ■ Temp_0	1 → PLC_1 [CPU 313C-2 DP] → Program blocks → OB1 [OB1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2		- 1
11 wagProfibus_MW-01V13_Sf wagProfibus_MW-01V13_Sf wagProfibus_MW-01V13_Sf wagProfibus_MW-01V13_Sf wagProfibus_MW-01V13_Sf wagProfibus_MW-01V13_Sf wagProfibus_MW-01V13_Sf wagProfibus_MW-01V13_Sf wagProfibus_MW-01V13_Sf wagProfibus_MW-01V13_Sf wagProfibus_MW-01V13_Sf wagProfibus_MW-01V13_Sf wagProfibus_MW-01V13_Sf wagProfibus_MW-01V13_Sf wagProfibus_MW-01V13_Sf wagProfibus_MW-01V13_Sf wagProfibus_MW-01V13_Sf wagProfibus_MW-01V13_Sf wagProfibus_MM-0	1 → PLC_1 [CPU 313C-2 DP] → Program blocks → OB1 [OB1] D3 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		-
11 NagProfibus_MW-01V13_Sf NagProfibus_MW-01V13_Sf NagProfibus_MW-01V13_Sf ■ 1 = 1 = 1 = 1 = 1 = 1 = 1 = 1 = 1 = 1	PI → PLC_1 [CPU 313C-2 DP] → Program blocks → OB1 [OB1] D 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2		-
11 wagProfibus_MW-01V13_Sf &	1 > PIC_1 [CPU 313C-2 DP] > Program blocks > OB1 [OB1] 중 월호 월호 표 한 약 약 전 약 전 약 전 1 = 1 은 약 약 1 Data type Offset Default value Comment Byte 월 0.0 Byte 1.0		-
11 waghofibus_MW-01V13_SP Waghofibus_MW-01V13_SP 181 Name • Temp • Temp	PI → PIC_1 [CPU 313C-2 DP] → Program blocks → OB1 [OB1] Image: Set in the set		-
11 11 waghofibus_MV-01V13_SP W+01V13_SP 181 Name • Temp • Temp	PI → PIC_1 [CPU 313C-2 DP] → Program blocks → OB1 [OB1] P = = = = P = <td< td=""><td></td><td>-</td></td<>		-
11 wagProfibus_MV-01V13_SF wX ∰ ∰ ∰ ∰	PI → PLC_1 [CPU 313C-2 DP] → Program blocks → OB1 [OB1] P 3 ± 3 ± 1 ± 1 ± 0, ∞ D 2 to type O 5 to type O 5 to type O 5 to type D 6 to type D 7 to type D 7 to type D 8 to type D 9 to type		-
11 wagProfibus_MV-01V13_SF wagProfibus_MV-01V13_SF wagProfibus_MV-01V13_SF B1 Name ○ Temp_0 ○ Temp_0 ○ Temp_1 ○ Temp_1 ○ Temp_1 ○ Comment 1 L "H 2 CAW	P1 → PLC_1 [CPU 313C-2 DP] → Program blocks → OB1 [OB1] P 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	\$DB3.DBW0	-
11 wagProfibus_MV-01V13_Sf wagProfibus_MV-0	PI → PIC_1 [CPU 313C2 DP] → Program blocks → OB1 [OB1] → 2 ± 2 ± 1 ± 2 ± 2 ± 1 ± 2 ± 2 ± 1 ± 2 ± 2	\$DB3.DBW0 \$DB3.DBW0	-
11 wagProfibus_MV-01V13_SF w% ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ Name > Temp_0 • Temp_1 • Temp_1 • Comment 1 L "H 2 CAN 3 T "H	PI > PLC_1 [CPU 313C-2 DP] > Program blocks > OB1 [OB1] P = = = = P C < 0	\$DB3.DBW0 \$DB3.DBW0	-
11 wagProfibus_MV-01V13_SF w% ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ DB1 Name • Temp_0 • Temp_1 • Temp_1 • Comment 1 2 CaN 3 T "H 5	PI > PLC_1 [CPU 313C2 DP] > Program blocks > OB1 [OB1] P 2 2 2 1 10 20 0 00 00 00 00 00 00 00 00 00 00 00	\$DB3.DBW0 \$DB3.DBW0	-
11 wagProfibus_MV-01V13_SF w% ₽ ₽ ₽ ₽ ₽ ₽ ₽ BI Name ● Temp_0 ● Temp_1 ● Temp_1 ● Comment 1 L "H 2 CAN 3 T "H 5 Network 24:	PI > PLC_1 [CPU 313C2 DP] > Program blocks > OB1 [OB1] P 3 2 3 2 1 1 2 1 2 1 1 2 1 1 2 1 1 2 1 1 2 1 1 2 1 1 2 1 1 2 1 1 2 1 1 2 1 1 2 1	\$DB3.DBW0	-
11 wagProfibus_MV-01V13_SF wagProfibus_MV-01V13_SF wagProfibus_MV-01V13_SF wagProfibus_MV-01V13_SF BI Name ************************************	PI > PLC_1 [CPU 313C2 DP] > Program blocks > OB1 [OB1] P 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	\$DB3.DBW0 \$DB3.DBW0	_ (
11 wagProfibus_MV-01V13_Sf w% ♥ ♥ ♥ ♥ ♥ ♥ ■ ■ ■ ■ DB1 Neme ● Temp_0 ● Temp_1 ● Temp_1 ● Comment 1 L "H 2 CAN 3 T "H 4 5 Network 24:	PI → PIC_1 [CPU 313C2 DP] → Program blocks → OB1 [OB1] D = 2 = 2 = 2 = 2 = 2 = 2 = 2 = 2 = 2 =	\$DB3.DBW0 \$DB3.DBW0	
11 wagProfibus_MV-01V13_Sf wag	PI > PLC_1 [CPU 313C-2 DP] > Program blocks > OB1 [OB1] P = = = = = C < 0	\$DB3.DBW0 \$DB3.DBW0	
11 wagProfibus_MV-01V13_SF w%gProfibus_MV-01V13_SF w%gProfibus_MV-01V13_SF w%gProfibus_MV-01V13_SF wetwork 20 [2000] 0 Temp_0 0 Temp_0 0 Temp_0 0 Temp_0 0 Temp_1 0	PI > PLC_1 [CPU 313C2 DP] > Program blocks > OB1 [OB1] P 3 2 3 2 1 1 2 1 2 1 1 2 1 1 2 1 1 2 1 1 1 1	\$DB3.DBW0 \$DB3.DBW0	
11 wagProfibus_MV-01V13_Sf w% ♥ ♥ ● ● ■ ● ■ ● ■ Name ● Temp_0 ● Temp_1 ● Temp_1 ● Comment 1 L "H 2 CAN 3 T "H 4 5 Network 24:	PI > PIC_1 [CPU 313C2 DP] > Program blocks > OB1 [OB1] Image: Section 2016 Image: Section 2016 Image: Section 2016 Image: Section 2016 D_ProfibusOutput".command D_ProfibusOutput".command D_ProfibusOutput".command D_ProfibusOutput".command D_ProfibusOutput".command D_ProfibusOutput".command D_ProfibusOutput".command	\$DB3.DBW0 \$DB3.DBW0 \$DB3.DBW0 \$DB3.DBW0	
11 wagProfibus_MV-01V13_SF wagProfibus_MV-0	<pre>Pl > PlC_1 [CPU 313C2 DP] > Program blocks > OB1 [OB1] P = = = = + + + + + + + + + + + + + + +</pre>	\$DB3.DBW0 \$DB3.DBW0 \$DB3.DBW0 \$DB3.DBW0	
11 wagProfibus_MV-01V13_SF wagProfibus_MV-0	<pre>PL > PLC_1 [CPU 313C2 DP] > Program blocks > OB1 [OB1] P 2 2 2 2 P 2 P 2 P 2 P 2 P 2 P 2 P 2 P</pre>	4DB3.DBW0 4DB3.DBW0 4DB3.DBW0 5DB3.DBW0 5DB3.DBW0 50B3.DBW0	
11 wagProfibus_MV-01V13_Sf wagProfibus_MV-0	<pre>Pl > PlC_1 (CPU 313C2 DP) > Program blocks > OB1 [OB1] D 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2</pre>	\$DB3.DEW0 \$DB3.DEW0 \$DB3.DEW0 \$DB3.DEW0 \$DB3.DEW0 \$DM8	
11 wagProfibus_MV-01V13_Sf wagProfibus_MV-01V13_Sf Name > Temp_0 Temp_0 Temp_1 6 Network 23: Comment 1 L "H 2 CAN 3 T "H 3 LADPB 3 LADPB 4 RECOR RET_V 6 7 8 9	PI > PIC_1 [CPU 313C2 DP] > Program blocks > OB1 [OB1] PI = III = IIII PIC Data type Offset Defaultvalue Comment Byre 0.0 Byre D_ProfibusOutput".command D_ProfibusOutput".command Byre D_ProfibusOutput".command AI If err write"	\$DB3.DBW0 \$DB3.DBW0 \$DB3.DBW0 \$DB3.DBW0 \$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.



