

Protokół komunikacji: Przetwornik masy MW-01-A

INSTRUKCJA OPROGRAMOWANIA

ITKP-42-01-11-21-PL



LISTOPAD 2021

SPIS TREŚCI

1.	KONFIGURACJA USTAWIEŃ PRZETWORNIKA MASY	4
2.	STRUKTURA DANYCH	4
	2.1. Adres wejściowy	4
	2.2. Opis rejestrów wejściowych	4
	2.3. Adres wyjściowy	7
	2.4. Opis rejestrów wyjściowych	7
3.	KONFIGURACJA MODUŁU PROFIBUS W ŚRODOWISKU TIA PORTAL V13	10
	3.1. Import GSD	10
	3.2. Konfiguracja modułu	12
4.	APLIKACJA DIAGNOSTYCZNA	14

1. KONFIGURACJA USTAWIEŃ PRZETWORNIKA MASY

Konfiguracji ustawień przetwornika masy MW-01-A do komunikacji z wykorzystaniem protokołu Profibus dokonujemy za pomocą programu komputerowego "**MwManager**" w zakładce **<Parametry / Komunikacja / Moduły dodatkowe>**. Konfiguracja ustawień jest szczegółowo opisana w instrukcji programu komputerowego "**MwManager**".

2. STRUKTURA DANYCH

2.1. Adres wejściowy

Zmienna	Offset	Długość [WORD]	Typ danych
Masa platformy	0	2	float
Tara platformy	4	2	float
Jednostka platformy	8	1	word
Status platformy	10	1	word
Próg Lo platformy	12	2	float
Status procesu	64	1	word
Stan wejść	66	1	word
Min	68	2	float
Max	72	2	float
Próg dozowania szybkiego	76	2	float
Próg dozowania wolnego	80	1	float

2.2. Opis rejestrów wejściowych

Należy zwrócić uwagę, że dane pobierane z przetwornika masy MW-01-A posidają odwróconą kolejność bajtów w rejestrach. I tak zmienne typu float posiadają kolejność DCBA a zmienne word BA. Żeby móc prawidłowo odczytać te rejestry należy zamienić ich kolejność.

<u>Masa platformy</u> – zwraca wartość masy danej platformy w jednostce aktualnej.

Przykład:

Odczytany rejestr o offsecie 0 posiada wartość hex równą 0x00001041. Przed zamianą na float należy odwrócić wartość kolejność bajtów do porządku ABCD w wyniku czego otrzymamy 0x41100000.

Po zamianie na float otrzymujemy 9.0 co stanowi bieżące wskazanie masy ładunku.

<u>Tara platformy</u> – zwraca wartość tary danej platformy w jednostce kalibracyjnej.

Jednostka platformy – określa aktualną (wyświetlaną) jednostkę masy.

Bity jedr	Bity jednostki								
0	Gram [g]								
1	Kilogram [kg]								
2	Karat [ct]								
3	Funt [lb]								
4	Uncja [oz]								
5	Newton [N]								

Przykład:

Wartość odczytana HEX 0x0200. Postać binarna:

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

Po zamianie z porzadku BA na AB otrzymamy 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

Jednostką wagi jest kilogram [kg].

<u>Status platformy</u> – określa stan danej platformy wagowej.

Bity s	tatusu
0	Pomiar prawidłowy (waga nie zgłasza błędu)
1	Pomiar stabilny
2	Waga jest w zerze
3	Waga jest wytarowana
4	Waga jest w drugim zakresie
5	Waga jest w trzecim zakresie
6	Waga zgłasza błąd NULL
7	Waga zgłasza błąd LH
8	Waga zgłasza błąd FULL

Przykład:

Odczytana wartość HEX: 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

Po zamianie z porzadku BA na AB otrzymamy 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

Waga nie zgłasza błędu, pomiar stabilny w drugim zakresie.

Próg LO – zwraca wartość progu LO w jednostce kalibracyjnej.

Status procesu – określa status procesu dozowania:

- 0x00 proces nieaktywny
- 0x01 proces uruchomiony
- 0x02 proces przerwany
- 0x03 proces zakończony

<u>Stan wejść</u> – maska bitowa wejść miernika. Pierwsze 3 najmłodsze bity reprezentują wejścia terminala wagowego.

Przykład:

Odczytana wartość HEX: 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

Po zamianie z porzadku BA na AB otrzymamy 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

Wejścia numer 1 i 2 przetwornika masy znajdują się w stanie wysokim.

MIN – zwraca wartość ustawionego progu MIN w jednostce kalibracyjnej.

MAX – zwraca wartość ustawionego progu MAX w jednostce kalibracyjnej.

Próg dozowania szybkiego – wraca wartość ustawionego progu dozowania szybkiego w jednostce kalibracyjnej.

<u>Próg dozowania wolnego</u> – wraca wartość ustawionego progu dozowania wolnego w jednostce kalibracyjnej.

2.3. Adres wyjściowy

Wykaz zmiennych wejściowych:

Zmienna	Offset	Długość [WORD]	Typ danych
Komenda	0	1	word
Komenda z parametrem	2	1	word
Tara	6	2	float
Próg LO	10	2	float
Stan wyjść	14	1	word
Min	16	2	float
Мах	20	2	float
Próg dozowania szybkiego	24	2	float
Próg dozowania wolnego	28	1	float

2.4. Opis rejestrów wyjściowych

<u>Komenda podstawowa</u> – zapisanie rejestru odpowiednią wartością spowoduje wywołanie następujących akcji:

Numer bitu	Akcja
0	Zeruj platformę
1	Taruj platformę
5	Start procesu
6	Zatrzymanie procesu

Przykład:

Zapisanie rejestru wartością 0x02 skonwertowaną do porządku BA 0x0200

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

Spowoduje wytarowanie wagi.



Komenda wykonywana jest jednorazowo, po wykryciu ustawienia danego jej bitu. Jeżeli konieczne jest ponowne wykonanie komendy z ustawionym tym samym bitem, należy go najpierw wyzerować a następnie ustawić na żądaną wartość ponownie.

<u>Komenda złożona</u> – ustawienie odpowiedniej wartości realizuje zadanie, zgodnie z tabelą:

Numer bitu	Akcja				
0	Ustawienie wartości tary dla danej platformy				
1 Ustawienie wartości progu LO dla danej platformy					
2	Ustawienie stanu wyjść				
3	Ustawienie wartości progu MIN				
4	Ustawienie wartości progu MAX				
5	Ustawienie progu dozowania szybkiego				
6	Ustawienie progu dozowania wolnego				

Komenda złożona wymaga ustawienia odpowiedniego parametru (offset od 6 do 36 – patrz tabela rejestrów wyjściowych)
Komenda z parametrem wykonywana jest jednorazowo, po wykryciu ustawienia danego jej bitu. Jeżeli konieczne jest ponowne wykonanie komendy z ustawionym tym samym bitem, należy go najpierw wyzerować a następnie ustawić na żądaną wartość ponownie.

Przykład:

Wysłanie do wagi tary o wartości 1.0.

Wykonanie komendy wymaga zapisania 2 rejestrów:

offset 2 – komenda z parametrem - wartość 0x0100 – po konwersji 0x0100. offset 6 – wartość tary w formacie float - 1.0 po konwersji do formatu DCBA 0x0000803F. Tara – parametr komendy złożonej: wartość tary (w jednostce kalibracyjnej).

<u>Próg LO</u> – parametr komendy złożonej: wartość progu LO (w jednostce kalibracyjnej).

<u>Stan wyjść</u> – parametr komendy złożonej: określający stan wyjść przetwornika masy.

Przykład:

Ustawienie w stan wysoki wyjść nr 1 i 3 terminala wagowego.

Maska wyjść będzie miała postać:

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

Po konwersji na HEX otrzymamy 0x05

Wykonanie komendy wymaga zapisania 2 rejestrów:

offset 2 – komenda z parametrem - wartość 0x08 – czyli zapis stanu wyjść.

offset 14 – maska wyjść 0x05.

W efekcie wyjścia numer 1 i 3 zostaną ustawione w stan wysoki.

<u>MIN</u> – parametr komendy złożonej: wartość progu MIN (w jednostce aktualnie używanego modu pracy).

MAX – parametr komendy złożonej: wartość progu MAX (w jednostce aktualnie używanego modu pracy).

Próg dozowania szybkiego - parametr komendy złożonej – wartość progu dozowania szybkiego (w jednostce kalibracyjnej).

<u>Próg dozowania wolnego</u> - parametr komendy złożonej – wartość progu dozowania wolnego (w jednostce kalibracyjnej).

3. KONFIGURACJA MODUŁU PROFIBUS W ŚRODOWISKU TIA PORTAL V13

Pracę w środowisku należy rozpocząć od założenia nowego projektu, w którym określona zostanie topologia sieci PROFIBUS ze sterownikiem MASTER, którym w tym przykładzie będzie sterownik serii S7-300 firmy SIEMENS.



3.1. Import GSD

Korzystając z dołączonego pliku konfiguracyjnego GSD należy dodać nowe urządzenie w środowisku. W tym celu należy użyć zakładki OPTIONS a następnie MANAGE GENERAL STATION DESCRIPTION FILES (GSD) i wskazać ścieżkę dostępu do pliku GSD.

Manage general	station description files 🛛 🗙
Source path:	C:\Users\user\Downloads\Radwa@Profibus3.5_V13_SP1_EX\Additiona Files\GSD
Content of impo	prted p
File	Info
<	OK Anuluj
	Delete Install Cancel

Po pomyślnym dodaniu pliku w liście urządzeń możemy już odnaleźć interesujący nas moduł Anybus-IC PDP.

			_ =' = X	Hardware catalog	□' Ш ▶
.	Topology view 🛛 🛔 Netw	ork view 🛛 👔	Device view	Options	
	Network overview	Connections		-	
~				✓ Catalog	
	1 Device	Туре			
	S7300/ET200M stat	tion_1 \$7300/	ET200M station	<earcn></earcn>	wf wr
=	✓ ▶ PLC_1	CPU 31	3C-2 DP	🗹 Filter	
	GSD device_1	GSD de	vice	Controllers	
	MW-01_Profibus	Anybus	-IC PDP	🕨 🫅 HMI	
				PC systems	
				Drives & starters	
				Network components	
				Detecting & Monitoring	
				Distributed I/O	
				Power Supplies	
				✓ Im Field devices	
				AS-Interface	
	-			Commanding and signaling devices	
	4			SIPLUS HCS	
				 Other field devices 	
	-			PROFINETIO	
				✓ Im PROFIBUS DP	
				Drives	
				Encoders	
				Gateways	
				👻 🛅 General	
				👻 🛅 HMS Industrial Networks	
				👻 🛅 Anybus-IC PDP	
				🚺 Anybus-IC PDP	
				SIEMENS AG	
				Ident systems	
				PLCs	

Należy utworzyć sieć składającą się z jednego sterownika MASTER oraz dodanego modułu SLAVE:

RadwagProfibus_MW-01V13_SP1 → Devie	es & networks	
		📱 Topology view
Network Connections HMI connection	🔽 🔛 🔜 🛨	
✓		
PLC_1 CPU 313C-2 DP		
PROFIB	JS_1	
	MW-01_Profibus	
	PLC_1	

3.2. Konfiguracja modułu

W dalszej kolejności należy określić adres modułu. Ten parametr musi być zgodny z adresem ustawionymi w menu za pomocą programu MwManager.

Rad	wagProfibus_M	N-01V13_SF	1 → PLC_1	[CPU 313C-2 DP]	Distributed I/O	DP-Ma	istersystem (1): PROF	IBUS_1 →	MW-01_
									2
dit.	MW-01_Profibus		. 🖽 🖾	🚄 📖 🔍 ±					-
									^
		ofibur							=
		101.PT							
		PUR							
				DP-NORM					
									~
<						> 1	00%		1

General IO tags Sys	stem constants Texts									
General Catalog information	PROFIBUS address									
PROFIBUS address	Interface networked with									
Watchdog	Subnet:	PROFIBUS_1								
Diagnostics addresses		Add new subnet								
	Parameters									
	Address:	1								
	Highest address:	126								
	Transmission speed:	1.5 Mbps								

Możemy przejść do konfiguracji modułu. Na wstępie określamy rozmiar rejestrów wejściowych oraz wyjściowych a także definiujemy ich adresy początkowe. W tym celu z listy dostępnych modułów INPUT oraz OUTPUT wybieramy takie jak na zdjęciu poniżej. Maksymalny rozmiar danych wejściowych wynosi 84 bajtów a wyjściowych 32 bajty. W projekcie użyto domyślnych adresów początkowych – 256 dla modułu INPUT i 256 dla OUTPUT:

RadwagProfibus_MW	-01V13_SP1 PLC_1 [CPU 313C-2 DP]	• Distribut	ed I/O 🔸	DP-Mastersystem (1): PROF	IBU S_1	I → MM	V-01_Profibus			_ # # ×
							🚽 Topolog	y view	🔥 Network view	Device view
MW-01_Profibus	💌 📰 🖾 🖽 🍳 ±	1	Device	overview						
		^	- 11	Module	Rack	Slot	I address	Q address	Туре	Article no.
	50045		_	MW-01_Profibus	0	0	1022*		Anybus-IC PDP	
	2 P101		~	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	
	2.	_	~	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				

Na tym etapie można załadować do sterownika konfigurację sprzętową.



Po pomyślnej kompilacji i wczytaniu kodu MASTER i SLAVE powinny nawiązać połączenie. Dalszym etapem będzie tworzenie kodu programu.

4. APLIKACJA DIAGNOSTYCZNA

Tworzenie aplikacji najlepiej zacząć od zdefiniowania nazw symbolicznych rejestrów wejściowych i wyjściowych. Rejestry wejściowe i wyjściowe modułu PROFINET określono w blokach danych HD_ProfbusInput oraz HD_ProfbusOutput i HD_ProfibusOutputTemp w grupie HARDWARE w gałęzi PROGRAM BLOCKS.



Bloki HD_ProfinetOutput, oraz HD_ProfinetInput reprezentują interesujące nas rejestry wejść/wyjść modułu PROFIBUS wagi. Wyglądają one jak poniżej:

Ra	dwa	agProfibus_MW-01\	(13_SP1 → PLC_1 [CP	PU 313C-2 DP) 🕨 Program I	blocks 🕨 Ha	rdware 🕨 S	aveInput	HD_ProfibusInput [DB2]			
2												
	HD_ProfibusInput											
		Name	Data type	Offset	Start value	Retain	Visible in	Setpoint	Comment			
1	-00	▼ Static										
2	-	mass	Real	0.0	0.0	~						
3	-	 tare 	Real	4.0	0.0							
4	-00	 unit 	Word	8.0	16#0							
5	-00	status	Word	10.0	16#0	~						
6	-	= lo	Real	12.0	0.0	~						
7	-	process_status	Word	16.0	16#0							
8	-00	 inputs 	Word	18.0	16#0							
9	-	min	Real	20.0	0.0	~						
10	-	max	Real	24.0	0.0							
11		bulk_dosing_th	reshold Real	28.0	0.0							
12	-00	fine_dosing_th	reshold Real	32.0	0.0							

a		3C-2 DPJ	Program blo	icks 🕨 Har	dware 🕨 S	aveOutput	▷ HD_ProfibusOutput [DB3]
	s 🖿 🔢 😤						
ProfibusOutput							
Name	Data type	Offset	Start value	Retain	Visible in	Setpoint	Comment
 Static 							
command	Word	0.0	16#00	\checkmark	\sim		
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	\checkmark		\sim	
set_bulk_dosing_thre	Real	22.0	10.0				
set_fine_dosing_thres	Real 🔳	26.0	20.0				
		ProfibusOutput Name Data type Static command Word complex_command Word set_tare set_lo cutputs Word set_min set_max set_bulk_dosing_thre Real set_fine_dosing_thres Real	ProfibusOutput Data type Offset Static Data type Offset • command Word 0.0 • complex_command Word 2.0 • set_tare Real 4.0 • set_to Real 4.0 • set_to Real 1.0 • set_min Real 14.0 • set_max Real 18.0 • set_bulk_dosing_thres. Real 12.0 • set_fine_dosing_thres. Real 12.0	Profibus 0 utput Data type Offset Start value * Static 0.0 16#00 • command Word 0.0 16#00 • complex_command Word 2.0 16#00 • set_tare Real 4.0 2.0 • set_lo Real 4.0 2.0 • set_mark Real 12.0 16#03 • set_mark Real 14.0 10.0 • set_mark Real 18.0 20.0 • set_fine_dosing_thres Real 2.0 10.0	ProfibusOutput Data type Offset Start value Retain • static 0.0 16#00 • • command Word 2.0 16#00 • • complex_command Word 2.0 16#00 • • set_tare Real 4.0 2.0 • • set_to Real 8.0 0.5 • • outputs Word 12.0 16#03 • • set_max Real 14.0 10.0 • • set_bluk_dosing_thre Real 12.0 16#03 • • set_max Real 18.0 20.0 • • set_fine_dosing_thres Real 18.0 20.0 •	Profibus O utput Data type Offset Start value Retain Visible in • Static • </th <th>Profibus O utput Data type Offset Start value Retain Visible in Setpoint • Static •</th>	Profibus O utput Data type Offset Start value Retain Visible in Setpoint • Static •

Blok HD_ProfibusOutputTemp służy do przechowywania danych tymczasowych podczas zamiany kolejności bajtów w rejestrach.

Ra	dwa	agl	rofibus_MW-01V13_SP1 PL0	1 [CPU 313C-	2 DP] 🕨	Program	blocks	Hardware	SaveOutput	t ▶ HD_Pro	ofibusOut	outTemp [DB1]
1	1	8	• • • • • • • • • • •	12 😜								
	HD	_P	rofibusOutputTemp									
		Na	me		Data type		Offset	Start value	Retain	Visible in	Setpoint	Comment
1	-	•	Static									
2			set_tare_inv		Real		0.0	2.0				
3			set_lo_inv		Real		4.0	0.5				
4			outputs_inv		Word		8.0	16#03				
5			set_min_inv		Real		10.0	1.1				
6			set_max_inv		Real		14.0	1.4				
7			set_bulk_dosing_threshold_inv		DWord		18.0	16#DE				
8	-		set_fine_dosing_threshold_inv		Word		22.0	16#16				

Pozostaje w głównej pętli programu stworzyć funkcje przepisujące stany fizycznych rejestrów wagi do rejestrów w blokach danych HD_ProfibusInput i HD_ProfibusOutput. Funkcje mogą wyglądać jak poniżej. Na przykładzie zaprezentowano sposób odczytu masy, jednostki oraz zapisu rejestrów "tara" i "komenda".

Należy zwróić uwagę że dane pobierane i zapisywane w przetworniku masy MW-01-A posidają odwróconą kolejność bajtów w rejestrach. I tak zmienne typu float posiadają kolejność DCBA a zmienne word BA. Żeby móc proawidłowo odczytac te rejestry należy zamiernić ich kolejność. W tym przykładzie użyto poleceń CAD dla zmiennych typu float oraz CAW dla zmiennych typu word.

Podobna zasada dotyczy zmiennych zapisywanych w przetworniku masy. Przed zapisem należy odwrócić kolejność bajtów. Jako przykład podamy sposób zapisu tary o wartości 1.5. Po konwersji na hex otrzymamy – 0x3FC00000. Przed zapisem do MW-01-A musimy dokonac konwersji do porządku DCBA. Po użyciu polecenia CAD otrzymamy 0000C03F i taką wartość należy zapisać w rejestrze wagi.

RadwagProfibus_MW-01V13_SP	1 PLC_1 [CPU 313C-2 DP]	 Program blocks > OB1 [OB1] 		_ # = X
		·원 GH 4D2 1_ 1_ 2, 000 118		
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 ,X : 한 : 문 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 ﷺ ∰ ∰ ∰ ∰ ∰ ∰ ∰ 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 Temp 2 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 = 0.0 Byte = 1.0	Program blocks > OB1 [OB1] cell €		- # = -
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 HR HR # # # # # # # # # OB1 Name 1	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 µ@ xX ⊉ ⊉ ⊕ ⊨ ⊑ ⊑ Name Name 1 ▼ Temp_0 3 ♥ Temp_0 3 ♥ Temp_1 Collar Comment 1 CALL DPI	1 → PLC_1 (CPU 313C-2 DP)	Program blocks > OB1 [OB1] Cell QB1 Comment Default value Comment Comment		_ # =
RadwagProfibus MV-01V13 SP with with site site a a a OB1 Name a a a 1 - Temp_0 a a a 2 - Temp_0 a a b b a 2 - Temp_1 c	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 • Temp_1 Comment □ LADDR 3 • Call_L 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 @ @ @ 0 1 = 1 a @ 0 0 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	W#16#108 %004 %DB2.DBW8	
RadwagProfibus_MV-01V13_SP	<pre>1 > PLC_1 [CPU 313C-2 DP] Data type Office Dyte 0 0 Byte 0 0 Byte 10 Byte 10</pre>	Program blocks > OB1 [OB1] o e and the second se	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 ± 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2</pre>	Program blocks > OB1 [OB1] Image: Comment in the second	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] Image: Comment in the second	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] Center of the second s	W#15#108 %XX4 %DB2.DBW8	
RadwagProfibus MW-01V13_SP will will 0B1 will Neme i 1 Temp_0 2 Temp_1 Comment i Comment i 1 CALL 2 I 2 I 2 I 1 CALL 2 I 2 I 2 I 2 I 2 I 2 I 2 I 2 I 2 I 3 I	<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 Comment 1 CALL DPI CALL DPI 3 • TEMP_1 4 Second RET_V, 5 6 7 8 9 10 11 Temp • • Network 7:	1 → PLC_1 [CPU 313C2 DP] P 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 %D52.D5W8	
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 => 0B1 Name 1 V Temp 2 1 Comment Comment 1 Labor 1 Call 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 2 1 1 1 1 1 1 1 1 1 1 2 1 2 1	<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 Image: Specific state s	<pre>1 → PLC_1 [CPU 313C2 DP] P 1 ± 1 ± = = 2 € 3 € Data type Offset Byte B 00 Byte</pre>	Program blocks > OB1 [OB1] a E	*DB2.DBW8 *DB2.DBW8	

All Pie Die Die Die Die Die Die Die Die Die D				
Unit Des hype Other Privativate Comment • • • • • • • • • • • • • • • • • • •	⊮X ⊉ € № 11 E	≡ ♥ 월 ± 월 ± 월 [2] (* 6₀ (8] 68 19 = 1= 6°	5 B	
• • • • • • • • • • • • • • • • • • •	Name	Data type Offset Default value Comme	nt	
• •	- Temp			
• Temp,1 Pro 10 • ************************************	Temp_0	Byte 🔳 0.0		
Account 1 1 100 profiburoutput*.set_tare_inv NoB3.0004 Account NoB3.0004 NoB3.0004 Account NoB4.004/11_SH 1 * NoL 1 (COU SISCOM) * NoB1 (MUW Comment NoB4.004 NoB4.004/11_SH 1 * NoL 1 (COU SISCOM) * NoB1 (MUW Comment NoB4.004 NoB4.004/11_SH 1 * NoL 1 (COU SISCOM) * NoB4.004 NoB5.0004 NoB4.004/11_	Temp_1	Byte 1.0		
Network 27:				
Network 20:	Notwork 27.			
1 1 "ND_Profibus0utput".set_tare NDD 2 7 "ND_Profibus0utputTemp".set_tare_inv NDD 2 2 3 4 2 3 4 5 3 4 5 6 7 6 7 6 6 7 <td>Comment</td> <td></td> <td></td> <td></td>	Comment			
important important important important important	1 .	"UD DwofibusOutput" act town	8DP2 DPD4	
1 1 *BD_PErifibusOutputTemp*.set_tare_inv Network 28: 1 1 </td <td>2 CAD</td> <td>ing_Fioribasoucput iset_tare</td> <td>0003.0004</td> <td></td>	2 CAD	ing_Fioribasoucput iset_tare	0003.0004	
* ************************************	3 т	"HD_ProfibusOutputTemp".set_tare_inv	%DB1.DBD0	
Network 28: Comment Image: Child DWMs_DAT Image: Child DWMs_D	4			
Network 28: Comment	5			
Connect CALL DFWR_DAT RECORD := "BD_ProfilueOutputTemp".set_tare_inv MI16f106 RECORD := "BD_ProfilueOutputTemp".set_tare_inv MI16f106 RECORD := "BO ProfilueOutputTemp".set_tare_inv MI16f106 Set Set Set Set Set Set Set Set	Network 28:			
Image of the set of the	Comment			
I LADOK :=%/16106 W16106 I LADOK :=%/16106 W16106 I LADOK :=%/16106 W16106 RET_VAL :="err write" W16106 NN00 NN00 I LADOK :=%/16106 NN00 I LADOK :=%/16106 NN00 NN00 NN00 I LADOK :=%/16106 NN00 I LADOK :=%/16106 NN00 I LADOK :=%/16100 NN00 I LADOK :=%/1	1 CALL	DPWR_DAT		
RECORD := ""B_PCfibusOutputTemp".set_tart_inv \$0B1.0BD0 SET_VAL :="err write" \$MN8 Mine \$MN8 Mare Data type Offet Defutivalue Comment \$MS3.DBM0 * Imp_0 byte byte 0 * Imp_1 byte byte 0 * Temp_1 byte * Temp_2	2 LJ	ADDR :=W#16#106	W#16#106	
Nework 23:	3 RI	CORD := "HD_ProfibusOutputTemp".set_tare_inv	\$DB1.DBD0	
i	5	T_VAL :- "err write"	400W 0	
************************************	6			
weighting weighting	7			
1 waphofibus_MW01V13_SP1 > PLC1 [CPU 313C2 DP] > Program blocks > OB1 [OB1] waphofibus_MW01V13_SP1 > PLC1 [CPU 313C2 DP] > Program blocks > OB1 [OB1] * * * * * * * * * * * * * * * * * * *	8			
11 wapProfibus_MWOVV13_SP1 > PLC_1 (CPU 313C2 DP] > Program blocks > 0B1 (DB1)	9			
wogPofibus_MW-01V13_SP1 > RC_1 [CPU 313C-2 DP] > Program blocks > OB1 [OB1] - 1 X = P + E = P + E = P + E + E = P + E + P + P + E + P + P + E + P + P +	11			
waphofibus_MWO1V13_SP1 > PC_1 (CRU 313C2 DP) > Program blocks > OB1 (OB1) X Y O O E Y O O O O O O O O O O O O O O O				
Name Data type Offset Default value Name Data type Offset Default value Image: Temp_0 Byte 0.0 Image: Temp_1 Byte 1.0 Image: Temp_1 Byte Image: Temp_1 <th></th> <th></th> <th></th> <th></th>				
Name Data type Offs Default value Comment • Temp.0 Byte 0.0 • </th <th>kă 🖻 👻 🎭 🖿 🚍</th> <th></th> <th>∞ ⊳ 10</th> <th></th>	kă 🖻 👻 🎭 🖿 🚍		∞ ⊳ 10	
Imp_0 Byte 0.0 Imp_1 Byte 1.0 Imp_1 Byte 1.0 Imp_1 Byte 1.0	kă ≝' ≅ № 100 100 191	≡ ♥2:2:100 ° 60 8 9 1= 1= 8	200 F	
* Temp_1 Byte 10 6	kX ⊉ ⊉ № ⊫ ⊑ DB1 Name	E P 2 ± 2 ± E 2 ℓ° 60 € € 9 1 ± 1 ± 6	an III	
Image: CALL DPWR_DAT *DB3.DBW0 CALL DPWR_DAT *DB3.DBW0 Image: CALL DPWR_DAT	kX ∰ ∰ ₩	E Default value Comm Byre ■ 0.0	oo D	
6 Image: Second Secon	X ⇒ → → + + + + + + + + + + + + + + + + +	□ 2 3 3 4 3 3 4 3 3 4 3 3 4 3 3 4 3 3 4 3 3 4 3 3 4 3 3 4 3 3 4 3 3 4 3 3 4 3 3 4 3 3 4 3	oo int	
* * Network 23: Comment 1 L 2 CAN 3 T * HD_ProfibusOutput".command *DB3.DBW0 * DB3.DBW0 * DB3.DBW0 * DB3.DBW0 * DB3.DBW0 * CAL Comment Comment * CALL CALL IADDR * RECORD * RECORD * RET_VAL i="err write" * HM18	X ♥ ♥ ♥ ₩ ₩ Ε Ε B1 Name ▼ Temp ■ Temp_0 ■ Temp_1	Image: Second secon	ant	
Network 23:	xX	Image: Second secon	00 III	
1 L "HD_ProfibusOutput".command \$DB3.DBW0 2 CAN \$DB3.DBW0 3 T "HD_ProfibusOutput".command \$DB3.DBW0 8 S S S Network 24:	X	E P St 2 t = P P C Co di Ci	ant	
2 CAN Important for the second secon	xă ⊉ ⊉ to te ti B1 Name → Temp_0 → Temp_1 6 Network 23:	Image: Second secon	ant	
3 7 "HD_ProfibusOutput".command \$DB3.DBW0 4 5 Network 24:	kă ≇ ≇ ♣ ⊨	Byte Boof ibusefultrut ¹ constand	202 D	
4 5 comment Comment 1 CALL DBWR_DAT 3 LADDR :=##f16f100 4 RECORD :=HB_ProfibusOutput".command 5 RET_VAL :="err write" 9 10	kX ⇒ ⇒ a b b b b b b b b b b b b b b b b b	P ≥ 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	NOC DES.DENO	
S Network 24: Comment 1 2 CALL DEWR DAT 3 LADDR :==##fl6f100 4 RECORD :==#fl0 ProfibusOutput".command 5 RET_VAL :="err write" 9 10	kX ∰ ∰ 6 E E BB1 Name → Temp.0 → Temp.1 6 Network 23: Comment 1 L 2 CAW 3 T	<pre>"HD_ProfibusOutput".command</pre>	* DB3.DBW0 * DB3.DBW0	
Network 24:	kX 22 22 24 24 25 25 25 25 25 25 25 25 25 25 25 25 25	Byte 00 Byte 00 Byte 10	оор III III III III III III III III III	
1 CALL DFWR_DAT W#16#100 3 LADDR :=##16#100 W#16#100 4 RECORD :="MD_ProfibusOutput".command %DB3.DBW0 5 RET_VAL :="err write" %MN8 9 10	kX ∰ ∰ ♣ E E B1 Name > Temp.0 = Temp.1 = Comment Comment 1 L 2 CAN 3 T 4 5	"HD_ProfibusOutput".command	SDB3.DBW0 SDB3.DBW0	
CALL DPWR_DAT Wflffloo LADDR i=#flfloo Wflffloo RECORD i="HD_ProfibusOutput".command \$D83.DBWO RET_VAL i="err write" \$MN8	xă ∰ ∰ ₩ ₩ ₩ ₩ Name > Temp.0 • Temp.0 • Temp.1 • Temp.1 • Temp.2 • Temp	Byte 00 1	SDB3.DBW0 SDB3.DBW0	
2 CALL DEWR_DAT 3 LADDR := ##\$16\$100 4 RECORD := "HD_ProfibusOutput".command 5 RET_VAL := "err write" 6 *MN8 9 10	kă ≇ ₽ № E E BBI Name Temp_0 Temp_1 Temp_1 Network 23: Comment L CAW 3 T Comment	Image: Second	DB3.DBW0 SDB3.DBW0	
4 RECORD :="HD_ProfibusOutput".command \$DB3.DBW0 5 RET_VAL :="err write" \$MM8 6	kă ≇ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽	Image: Second	************************************	
5 RET_VAL :="err write" \$M088 6	kă ≇ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽	"HD_ProfibusOutput".command "HD_ProfibusOutput".command	000 ₪ nt \$DB3.DBWD \$DB3.DBWD \$DB3.DBWD	
6 7 8 9 10	kă ⊯ ₽ ₩ E E OB1 Name Temp_0 Temp_0 Temp_1 Temp_1 Comment L Comment C Comment C Comment C Comment C Comment C Comment C Comment C Comment C Comment C Comment C Comment C C C C C C C C C C C C C	<pre>"HD_ProfibusOutput".command</pre>	* * * * * * * * * * * * * * * * * * *	
7 8 9 10	kă ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ >	Image: Second	SDB3.DBW0 SDB3.DBW0 SDB3.DBW0 SDB3.DBW0 SDB3.DBW0	
8 9 10	kă # # Image: Base of the second se	"HD_ProfibusOutput".command T_VAL :="err vrite"	000 ₪ nnt 100 0 100	
10	kă ⊯ ₩ ₩ E E BBI Name Temp_0 Temp_0 Temp_1 Temp_	<pre>"HD_ProfibusOutput".command "HD_ProfibusOutput".command "HD_ProfibusOutput".command "HD_ProfibusOutput".command "HD_ProfibusOutput".command DPWR_DAT DDDR :=##116#100 (CODD):="HD_ProfibusOutput".command tf_VAL :="err write"</pre>	N#166100 NMR8	
	kă ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ >	Image: Second	************************************	
	kX ∰ ∰ € E E B1 Name > Temp_0 • Temp_0 • Temp_1 • Temp_1	"HD_ProfibusOutput".command "T_VAL :="err write"	000 ₪ nnt 9DB3.DBWD 9DB3.DBWD 9DB3.DBWO 9DB3.DBWO 9DB3.DBWO 9DB3.DBWO	

Po kompilacji i załadowaniu programu do sterownika w bloku danych możemy odczytać interesujące nas rejestry wejściowe (MONITOR ALL) oraz zapisywać rejestry wyjściowe (np. poprzez zmianę START VALUE i LOAD START VALUES AS ACTUAL) modułu SLAVE.



