

Przetwornik masy MW-01-A

INSTRUKCJA OPROGRAMOWANIA

ITKP-40-01-11-21-PL



LISTOPAD 2021

SPIS TREŚCI

1.	KONFIGURACJA USTAWIEŃ PRZETWORNIKA MASY	4
2.	STRUKTURA DANYCH	4
	2.1. Adres wejściowy	4
	2.2.1 Opis rejestrów wejściowych	4
	2.2. Adres wyjściowy	7
	2.2.1 Opis rejestrów wyjściowych	7
3.	KONFIGURACJA MODUŁU PROFINET W ŚRODOWISKU TIA PORTAL V16	10
	3.1. Import GSD	10
	3.2. Konfiguracja modułu	12
4.	APLIKACJA DIAGNOSTYCZNA	16

1. KONFIGURACJA USTAWIEŃ PRZETWORNIKA MASY

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

2. STRUKTURA DANYCH

Wszystkie rejestry mają postać 2 bajtową (WORD). Dane zmiennoprzecinkowe (jak masa czy tara) przechowywane są w dwóch kolejnych rejestrach i mają postać FLOAT. Jeżeli pierwszy rejestr składa się z dwóch bajtów AB a drugi z dwóch bajtów CD to FLOAT będzie miał postać HEX ABCD

2.1. Adres wejściowy

Wykaz zmiennych wejściowych:

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	2	float
Status kalibracji	100	1	word

2.2.1 Opis rejestrów wejściowych

Masa platformy – zwraca wartość masy netto platformy w jednostce aktualnej.

Przykład:

Odczytany rejestr o offsecie 0 posiada wartość hex równą 0x43E28000, po zamianie na float otrzymujemy 453.0 co stanowi bieżące wskazanie masy ładunku.

Tara platformy – zwraca wartość tary platformy w jednostce kalibracyjnej.

<u>Jednostka platformy</u> – określa aktualną (wyświetlaną) jednostkę masy netto platformy.

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 0x02. 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	0	0	0	0	0	0	0	0	1	0

Jednostką wagi jest kilogram [kg].

Status platformy – określa stan 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: 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

Waga nie zgłasza błędu, pomiar stabilny w drugim zakresie.

<u>Próg LO platformy</u> – zwraca wartość progu LO w jednostce kalibracyjnej platformy.

Status procesu – określa status procesu dozowania lub recepturowania:

Wartość HEX	Opis
0x00	Proces nieaktywny
0x01	Proces uruchomiony
0x02	Proces przerwany
0x03	Proces zakończony

<u>Stan wejść</u> – maska bitowa wejść przetwornika masy. Pierwsze 3 najmłodsze bity reprezentują stan wejść przetwornika masy.

Przykład:

Odczytana wartość HEX: 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

Wejścia numer 1 i 3 terminala wagowego 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 – zwraca wartość ustawionego progu dozowania szybkiego w jednostce kalibracyjnej.

Próg dozowania wolnego – zwraca wartość ustawionego progu dozowania wolnego w jednostce kalibracyjnej

<u>Status kalibracji</u> – określa status procesu kalibracji.

Wartość HEX	Opis
0x00	Proces zakończony poprawnie
0x01	Trwa wyznaczanie masy startowej / współczynnika kalibracji
0x02	Przekroczony zakres
0x03	Przekroczony czas
0x04	Proces przerwany

2.2. Adres wyjściowy

Wykaz zmiennych wyjściowych:

Zmienna	Offset	Długość [WORD]	Typ danych
Komenda podstawowa	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	2	float

2.2.1 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.

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

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 (z parametrem)</u> – ustawienie odpowiedniej wartości realizuje zadanie, zgodnie z tabelą:

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

Komenda złożona wymaga ustawienia odpowiedniego parametru (offset od 6 do 28 – 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 0x01 – czyli ustawienie tary.

offset 6 – wartość tary w formacie float - 1.0.

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 przetwornika masy.

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 0x04 – 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 kalibracyjnej).

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

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

Próg dozowania wolnego – parametr komendy złożonej: wartość progu dozowania dokładnego(w jednostce kalibracyjnej).

3. KONFIGURACJA MODUŁU PROFINET W ŚRODOWISKU TIA PORTAL V16

Pracę w środowisku należy rozpocząć od założenia nowego projektu, w którym określona zostanie topologia sieci PROFINET ze sterownikiem MASTER, którym w tym przykładzie będzie sterownik serii S7-1200 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 sta	ation description file	s		×
Installed GSDs	GSDs in the proje	ct		
Source path: D:1	ProiektvlTiaPortal\GSD_Pr Wybierz folder	rofinet\GSDML-V2-3-HMS-AB	BICPRT-20130219	
Content of import	← → • ↑ <mark> </mark>	« TiaPor » GSD_Profine	et ∽ Ō	, Przeszukaj: G
GSDML-V2.3-HMS	Organizuj 🔻 No	wy folder		
	Nazwa	^	Data modyfikacji	Тур
	GSDML-V2-3-HM	IS-ABICPRT-20130219	05.11.2021 13:09	Folder plików
	<			
		Folder: GSDML-V2-3-HM	IS-ABICPRT-20130219	
<				Wybierz folder
		_	Delete Ins	tall Cancel

Po pomyślnym dodaniu pliku w liście urządzeń możemy już odnaleźć interesujący nas moduł ABIC-PRT:

rch in project>	ni -				Totally Integrated Automation PORT
			_ # i	×	K Hardware catalog 📰 🔳
Topology view	view	Device viev	v	Options	
Network overview	v Conn	ections 4			
Y Device		Туре		A	✓ Catalog
▼ \$7-1200 st	ation_1	57-12	00 station		<search> MI M</search>
PLC_1		CPU 1	211C AC/DC/Rly		Filter Profile:
 GSD device 	<u>_1</u>	GSD d	levice		A Controller
ABIC-PR	г	RT Mig	gration (FW>=2.00)		
					PC systems
					Drives & starters
					Network components
					Detecting & Monitoring
					Distributed I/O
					Power supply and distribution
					Field devices
					Other field devices
					Additional Ethernet devices
					▼ ■ PROFINET IO
					Drives
					Encoders
					Gateway
					HMS Industrial Networks
					Anybus-IC PRI
					Migration
				_	RT Migration (FW 1.38)
<				>	RI Migration (FW>=2.00)
Q Properties	i Info	& Diagn	ostics		Ki Standard
					Sensors

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

Project_08_Profinet_	MW_01_New_instruk	cja 🕨 Devices &	networks	
			5	Topology view
Network	tions HMI connection		. 🗆 🖽 🛄 🍳	t t
PLC_1	-	ABIC-PRT	and the second se	
CPU 121	1C	RT Migration (F	DP-NORM	
		100_1		
		DN/15 1		
		PN/IE_1		

3.2. Konfiguracja modułu

Na tym etapie należy zbudować sieć złożoną ze sterownika MASTER, oraz urządzenia SLAVE (przetwornik masy MW-01-A). Po podłączeniu zasilania w środowisku można wyszukać urządzenia korzystając z funkcji ACCESSIBLE DEVICES. W efekcie powinniśmy odnaleźć na liście zarówno MASTER jak i SLAVE:

-		Type of the PG/PC in PG/PC in	terface: 🖳 PN/ terface: 🔝 Rea	IE Iltek PCIe GbE Family	Controller 💌 💽
	Accessible nodes	of the selected interface	2:		
	Device	Device type	Interface type	Address	MAC address
	pawelk	SIMATIC-PC	PN/IE	10.10.3.145	4C-ED-FB-44-CO-31
and the second sec	mw01-profinet	RT Migration (FW 1.xx)	PN/IE	10.10.8.100	00-30-11-34-44-66
.	tk-kacperczyk-m	SIMATIC-PC	PN/IE	10.10.22.21	F8-32-E4-A0-BF-29
Flash LED					
Online status informati	ion:			🗌 Display	Start search
Found accessible	device mw01-profinet			_ bishid	,,
Scan completed.	4 devices found.				
? Retrieving device i	information				
Scan and informat	tion retrieval complete	d.			

W dalszej kolejności należy określić adres IP modułu i jego nazwę w sieci PROFINET. Po zaznaczeniu modułu w zakładce PROPERTIES odnajdujemy podmenu PROFINET INTERFACE gdzie wpisujemy adres IP oraz nadajemy nazwę. Te ustawienia muszą być zgodne z parametrami ustawionymi w przetworniku masy MW-01. Należy pamiętać o tym żeby adres IP SLAVE znajdował się w tej samej podsieci co adres MASTER.

ABIC-PRT [RT Migration (FW>=	=2.00)]	🖳 Properties 🚺 Info 📳 Diagnostics 📰 🖃 🥆
General IO tags Sy	stem 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 IO controller
Identification & Maintenance		
	•	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
		~

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 102 bajty i 52 bajty dla danych wyjściowych. W projekcie użyto domyślnych adresów początkowych – 256 dla modułu INPUT i 256 dla OUTPUT o rozmiarze 102 bajtów danych wejściowych i 32 bajtów dla danych wyjściowych:





🗓 Info 🛛 🗓 Diagi **Properties** PLC tags
 PLC data types General IO tags System constants Texts Libraries Watch and force tables General I/O addresses Catalog information Traces IIO addresses Output addresses OPC UA co ✓ Details view Start address: 256 Name > Information ABIC-PRT STOFINET_Inp... Portal view

Na tym etapie można załadować do sterownika konfigurację sprzętową:

Project_08_Profinet_	MW_01_New ► Devices & networks				_ # =×
			🚽 Topology view	h Network view	Device view
Network 1 Connec	tions HMI connection	% 1 8	11 🔍 ±		
					^
					<u>=</u>
PLC_1 CPU 121	1C ABIC-PRT RT Migration (F	DP-NOR	м		
	PLC 1				- Vetw
	Change device				ork o
	🗶 Cut	Ctrl+X			<u>+ ដ</u>
	E Copy	Ctrl+C			
	aste	Ctrl+V			
	X Delete	Del			
	Aution to any DD and to (10 and to 11	12			~
<	Disconnect from DP master system / IC	system	> 100	%	
	Highlight DP master system / IO system	ı´	Sector Properties	🗓 Info 🛛 🗓 Diag	nostics 🔤 🗖 🗖 🤝 🤜
General Cross-	🚽 Go to topology view				
Show all m	Compile	•		7	
	Download to device		Hardware and software (only changes)		
! Message	Go online	Ctrl+K Ctrl+M	Hardware configuration Software (only changes)	Time	
The project Proje	🖳 Online & diagnostics	Ctrl+D	Software (all)	21 2:13:39 PM	^
Connection to PL	Assign device name		11512	021 2:14:32 PM	
Start downloadin	Update and display forced operands		11/5/2	2021 2:32:52 PM	
	Show catalog C	trl+Shift+C	11/5/2	2021 2:32:55 PM	
PLC_1:	Export module labeling strips		11/5/2	2021 2:32:57 PM	
✓ Hardw	🔯 Properties	Alt+Enter	11/5/2	2021 2:33:01 PM	

Po pomyślnej kompilacji i wczytaniu kodu MASTER i SLAVE powinny nawiązać połączenie. Można to sprawdzić przechodząc do połączenia ONLINE. Powinniśmy uzyskać wynik jak poniżej.



Dalszym etapem będzie tworzenie kodu programu.

4. APLIKACJA DIAGNOSTYCZNA

Tworzenie aplikacji najlepiej zacząć od zdefiniowania nazw symbolicznych rejestrów wejściowych i wyjściowych. W tym celu korzystamy z gałęzi drzewa projektu o nazwie PLC TAGS. Na potrzeby tego przykładu stworzono tablice tagów jak na rysunku poniżej:



Rejestry wejściowe i wyjściowe modułu PROFINET określono w tablicach TAG_ProfiNET_IN oraz TAG_ProfiNET_OUT. Poniższe rysunki prezentują nadane nazwy symboliczne i adresację:

Proj	ect_	08_Profinet_MW_01_New_ins	trukcja 🕨 PLC_	1 [CPU 12110	C AC/DC/RI	y] ► PL(C tags 🕨	TAG_	ProfiNET_IN [12]		_ - - - ×	ĸ
										🕣 Tags	User constants	1
۵	# # B B 🖤 🛍									-		
Т	AG_	ProfiNET_IN										
-		Name	Data type	Address	Retain	Acces	Writa	Visibl	Comment			
1	-0	AnyBus_Platform_mass	DWord	%ID256							^	^
2	-0	AnyBus_Platform_tare	DWord	%ID260								1
З	-0	AnyBus_Platform_unit	Word	%IW264								
4	-0	AnyBus_Platform_status	Word	%IW266							=	=
5	-0	AnyBus_Platform_LO_threshold	DWord	%ID268								
6	-0	AnyBus_Process_status	Word	%IW320								
7	-0	AnyBus_Inputs_status	Word	%IW322								
8	-0	AnyBus_Min	DWord	%ID324								
9	-0	AnyBus_Max	DWord	%ID328								
10	-0	AnyBus_Fast_dosing_threshold	DWord	%ID332								
11	-0	AnyBus_Slow_dosing_threshold	DWord	%ID336								
12	Ð	AnyBus_Adjustment_status	Word	%IW356	-						~	~

Proje	ect_	08_Profinet_MW_01_New_inst	trukcja 🕨 PLC_	1 [CPU 1211C A	C/DC/RI	y] 🕨 PLC	tags 🕨	• TAG_I	ProfiNET_OUT [9]		_∎×
										🕣 Tags	User constants
# :	ø	🖻 🛃 🙄 🛍								_	
T	AG_	ProfiNET_OUT									
		Name	Data type	Address	Retain	Acces	Writa	Visibl	Comment		
1	-01	AnyBus_Command	Word 🔳	%QW256 -	•		$\mathbf{\sim}$				
2	-	AnyBus_Command_with_para	Word	%QW258			~				
3	-	AnyBus_Tare	DWord	%QD262							
4	-00	AnyBus_LO_threshold	DWord	%QD266			~				
5	-	AnyBus_Output_state	Word	%QW270							
6	-	AnyBus_Set_Min	DWord	%QD272							
7	-00	AnyBus_Set_Max	DWord	%QD276			~				
8	-0	AnyBus_Set_Fast_dosing_thres	DWord	%QD280							
9	-0	AnyBus_Set_Slow_dosing_thre	DWord	%QD284			\checkmark				
10		<add new=""></add>				V	~	V			

Żeby nie pracować bezpośrednio na fizycznych wejściach/wyjściach modułu PROFINET warto stowrzyć bloki danych zawieracjące reprezentacje tych rejestrów oraz "przepisać" warości pomiędzy nimi. W tym celu definiujemy bloki danych jak poniżej:

Project tree	
Devices	
۲. The second	🔲 🖬
Name	
Program blocks	^
Add new block	
Hain [OB1]	
ProfiNET_Input [DB1]	
ProfiNET_Output [DB2]	
Technology objects	=
External source files	
🕨 🚂 PLC tags	
PLC data types	
Watch and force tables	
🕨 🙀 Online backups	
🕨 🔀 Traces	
OPC UA communication	
Device proxy data	
Program info	
PLC alarm text lists	
Local modules	
Distributed I/O	~

Bloki ProfiNET_Input oraz ProfiNET_Output reprezentują interesujące nas rejestry wejść/wyjść modułu PROFINET przetwornika masy MW-01-A. Wyglądają one jak poniżej:

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												
	Name		Data type	Offset	Start value	Retain	Accessible f	Writa	Visible in	Setpoint	Comment	
1 🕣	▼ St	atic										
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							
κ										>		

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)												
		Name	Data type	Offset	Start value	Retain	Accessible f	Writa	Visible in	Setpoint	Comment	
1 🔫		 Static 										
2 <		 Command 	Word	0.0	16#0							
3 🔫		 Command_with_parameter 	Word	2.0	16#0							
4 🔫	0	 Platform 	Word	4.0	16#0							
5 \prec	0	 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 ┥	0	 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 🔫	0	 Adjustment_weight_mass 	DWord	32.0	16#0							
	۲										>	

Do przepisania wartości pomiędzy fizycznymi wejściami/wyjściami modułu a rejestrami w blokach danych można wykorzystać np. instrukcję MOVE:

Project_08_Profinet_MW_01_New + PLC_1 [CPU 1211C AC/DC/Rly] + Program blocks + Main [OB1]	_₽■×							
🔫 1월 등 등 📄 🚍 🚍 🖻 제 7 월 7 월 7 월 7 월 9 년 6 년 6 월 9 월 6 년 년 년 6 월 9 8 8	🔒 📑							
Block interface								
-++/ (??) -→ - Ť ∧00								
▼ Block title: "Main Program Sweep (Cycle)"	^							
Comment								
 Network 1: mass 	-							
Comment								
*20256 *Angeu*PotNtT_input*. Platform_mass*N & OUTImass								
 Network 2: command 								
Comment MOVE EN EN *QW256 *Anglus Command N								
	100%							

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.



