

# Instrukcja oprogramowania

ITKP-22-01-04-20-PL

# PROFIBUS

Protokół komunikacji: Miernik wagowy PUE HX7



Znajdziesz tam więcej przydatnych informacji w przystępnej formie! Zeskanuj kod QR, aby obejrzeć dodatkowe materiały naukowe, które mogąCię zainteresować radwag.com

## KWIECIEŃ 2020

# SPIS TREŚCI

1.	KONFIGURACJA USTAWIEŃ MIERNIKA WAGOWEGO	4
2.	STRUKTURA DANYCH	4
	2.1. Adres wejściowy	4
	2.2. Opis rejestrów wejściowych	. 5
	2.3. Adres wyjściowy	7
	2.4. Opis rejestrów wyjściowych	7
3.	KONFIGURACJA MODUŁU PROFIUS W ŚRODOWISKU TIA PORTAL V13	10
	3.1. Import GSD	11
	3.2. Konfiguracja modułu	13
4.	APLIKACJA DIAGNOSTYCZNA	14

# 1. KONFIGURACJA USTAWIEŃ MIERNIKA WAGOWEGO

Konfiguracji ustawień miernika do komunikacji z wykorzystaniem protokołu Profibus dokonujemy w podmenu **<SETUP / Urządzenia / Moduł anybus>**. Konfiguracja ustawień jest szczegółowo opisana w instrukcji "**Miernik PUE HX7**".

### 2. STRUKTURA DANYCH

#### 2.1. Adres wejściowy

#### Wykaz zmiennych wejściowych:

Zmienna	Offset	Długość [WORD]	Typ danych
Masa platformy 1	0	2	float
Tara platformy 1	4	2	float
Jednostka platformy 1	8	1	word
Status platformy 1	10	1	word
Próg Lo platformy 1	12	2	float
Masa platformy 2	16	2	float
Tara platformy 2	20	2	float
Jednostka platformy 2	24	1	word
Status platformy 2	26	1	word
Próg Lo platformy 2	28	2	float
Status procesu (Stop, Start)	64	1	word
Stan wejść	66	1	word
Min	68	2	float
Мах	72	2	float
Numer serii	84	2	dword
Operator	88	1	word
Towar	90	1	word
Kontrahent	92	1	word
Opakowanie	94	1	word
Receptura	100	1	word
Proces dozowania	102	1	word

#### 2.2. Opis rejestrów wejściowych

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

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

<u>Jednostka platformy</u> – określa aktualną (wyświetlaną) jednostkę masy danej 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 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: 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</u> – zwraca wartość progu LO w jednostce kalibracyjnej danej platformy.

Status procesu – określa status procesu dozowania lub recepturowania:

0x00 - proces nieaktywny,

0x01 – proces uruchomiony,

0x02 - proces przerwany,

0x03 – proces zakończony.

<u>Stan wejść</u> – maska bitowa wejść miernika. Pierwsze 4 najmłodsze bity reprezentują wejścia terminala wagowego.

#### Przykład:

Odczytana wartość HEX: 0x000B

B1/7	B1/6	B1/5	B1/4	B1/3	B1/2	B1/1	B1/0	B0/7	B0/6	B0/5	B0/4	B0/3	B0/2	B0/1	B0/0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1

Wejścia numer 1,2 i 3 terminala wagowego znajdują się w stanie wysokim.

MIN – zwraca wartość ustawionego progu MIN w jednostce aktualnej.

MAX – zwraca wartość ustawionego progu MAX w jednostce aktualnej.

<u>Numer serii</u> – zwraca wartość numeru serii. Akceptowane są tylko wartości numeryczne! Wszystkie inne znaki są pomijane.

**Operator** – zwraca wartość kodu zalogowanego operatora.

**Towar** – zwraca wartość kodu wybranego towaru.

Kontrahent – zwraca wartość kodu wybranego kontrahenta.

**<u>Opakowanie</u>** – zwraca wartość kodu wybranego opakowania.

Receptura – zwraca wartość kodu wybranej receptury.

Proces dozowania – zwraca wartość kodu wybranego procesu dozowania.

#### 2.3. Adres wyjściowy

#### Wykaz zmiennych wyjściowych:

Zmienna	Offset	Długość [WORD]	Typ danych
Komenda	0	1	word
Komenda z parametrem	2	1	word
Platforma	4	1	word
Tara	6	2	float
Próg LO	10	2	float
Stan wyjść	14	1	word
Min	16	2	float
Мах	20	2	float
Numer serii	32	2	dword
Operator	36	1	word
Towar	38	1	word
Kontrahent	40	1	word
Opakowanie	42	1	word
Receptura	48	1	word
Proces dozowania	50	1	word

#### 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
2	Wyczyść statystyki
3	Zapisz/Drukuj
4	Start procesu
5	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.

	٥

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 numeru serii
3	Ustawienie stanu wyjść
4	Wybór operatora
5	Wybór produktu
6	Wybór opakowania
7	Ustawienie wartości progu MIN
8	Wybór kontrahenta
9	Wybór magazynu źródłowego
10	Wybór magazynu docelowego
11	Wybór procesu dozowania
12	Ustawienie wartości progu MAX

Komenda złożona wymaga ustawienia odpowiedniego parametru (offset od 4 do 50 – 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 dla 1-szej platformy

Wykonanie komendy wymaga zapisania 3 rejestrów:

offset 2 – komenda z parametrem - wartość 0x01 – czyli ustawienie tary.

offset 4 – numer platformy wagowej, do której chcemy przypisać tarę- wartość 0x01 dla pierwszej platformy.

offset 6 – wartość tary w formacie float - 1.0

<u>Platforma</u> – parametr komendy złożonej: numer platformy wagowej (1 lub 2).

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 miernika wagowego i modułu komunikacyjnego.

#### 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).

<u>Numer serii</u> – parametr komendy złożonej: wartość numeru serii. Akceptowane są tylko wartości numeryczne! Wszystkie inne znaki są pomijane.

**Operator** – parametr komendy złożonej: kod operatora (tylko numeryczny).

Towar – parametr komendy złożonej: kod towaru (tylko numeryczny).

<u>Kontrahent</u> – parametr komendy złożonej: kod kontrahenta (tylko numeryczny).

<u>Opakowanie</u> – parametr komendy złożonej: kod opakowania (tylko numeryczny)

Receptura – parametr komendy złożonej: kod receptury (tylko numeryczny).

**<u>Proces dozowania</u>** - parametr komendy złożonej: kod procesu dozowania (tylko numeryczny).

#### 3. KONFIGURACJA MODUŁU PROFIUS 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.

Device name: PLC_2	<ul> <li>✓ Controllers</li> <li>&gt; SIMATIC \$7-1200</li> <li>&gt; MATIC \$7-1500</li> <li>✓ SIMATIC \$7-300</li> </ul>	<u>^</u>	Device:	1 and 1 == 1
PLC_2		^	Device:	
		^	Device:	E mark
		^	Device:	
	Controllers     Call SIMATIC S7-1200     Call SIMATIC S7-1500     Call SIMATIC S7-300	^	Device:	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
	Lim SIMATIC \$7-1200      Lim SIMATIC \$7-1500      Tim SIMATIC \$7-300			
	Imatic \$7-1500     Imatic \$7-300			
	SIMATIC \$7-300			AA I
Controllers				
	✓ Im CPU			
	CPU 312			CPU 313C-2 DP
	CPU 312C			
	CPU 313C		Articla po :	6557 212 60502 0480
	▼ ☐ CPU 313C-2 DP		Article no	6E37 515-6CF05-0A80
HMI	6ES7 313-6CF03-0AB0		Version:	V2.6
	6ES7 313-6CG04-0AB0			
	CPU 313C-2 PtP	≡	Description:	
	CPU 314		Work memory	64KB; 0.1ms/1000 instructions;
	CPU 314C-2 DP		3 channels co	unting and measuring with 24 V
DC australia	CPU 314C-2 PN/DP		(30kHz) incren	nental encoders; MPI+DP interface
PC systems	CPU 314C-2 PtP		(DP master or	DP slave); multi-tier configuration
	CPU 315-2 DP		receiving in di	rect data exchange; constant bus
	CPU 315-2 PN/DP		cycle time; rou	iting; S7 communication (loadable
	CPU 317-2 DP		FBs/FCs); firmv	vare V2.6; also available as SIPLUS
	CPU 317-2 PN/DP		2ABO.	nucle number oxar 515-oct 05-
	CPU 319-3 PN/DP			
	CPU 315F-2 DP			
	CPU 315F-2 PN/DP			
	CPU 317F-2 DP			
	CPU 317F-2 PN/DP			
	CPU 319F-3 PN/DP	~		
	<			

#### 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	Iuser/Downloads\RadwanProfibus3 5_V13_SP1 FX\Additiona Files\GSD Przeglądanie w poszukiwaniu folderu	
Content of imported p		
File hms_1810.gsd		Info
<	OK Anuluj	>
	Delete Install	Cancel

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



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

R Network Connections HMI connection	n 💌 🕎 🗒 🔂 🍳 ±
PLC_1 CPU 313C-2 DP	
PROFI	BUS_1
	_
	HX7Profibus Anybus-IC PDP PLC_1

#### 3.2. Konfiguracja modułu

Ger Ger PRO Ger Wat SYN Diag

W dalszej kolejności należy określić adres modułu. Ten parametr musi być zgodny z adresem ustawionym w menu wagi.

	^					_
		<b>**</b>	Module	Rack	Slot	
359	≡		HX7Profibus	0	0	
rolite			INPUT: 32 Byte (16 word)_1	0	1	
atile			INPUT: 32 Byte (16 word)_2	0	2	
N.	_		INPUT: 32 Byte (16 word)_3	0	3	
			INPUT: 16 Byte ( 8 word)_1	0	4	
► HX7Profibus			INPUT: 4 Byte ( 2 word)_1	0	5	
			OUTPUT: 32 Byte (16 word)	0	6	
DP.NORM		-	OUTPUT: 16 Byte (8 word)_1	0	7	
		-	OUTPUT: 4 Byte ( 2 word)_1	0	8	
				0	9	
				0	10	
				0	11	
				0	12	
				0	13	
				0	14	
				0	15	
				0	16	
				0	17	
				0	18	
	Y			0	19	_
> 100%		<	10			>
		🔍 Proper	rties 🚺 Info 🖞 Diagi	nostics		1 🖂
neral IO tags System constants Texts		_				
eral	Add new subnet					_
atalog information	/ dd field Subject					
FIBLIS address						
eral DP parameters						
hdog Addres	s: 1				-	-
C/FREFZE					_	31
Highest addres	5: 126				-	×.
Transmission spee	d: 1.5 Mbps					w.

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 116 bajtów i tyle samo dla danych wyjściowych. W projekcie użyto domyślnych adresów początkowych – 256 dla modułu INPUT i 256 dla OUTPUT:

Rad	lwagProfi	busHX7_V13_SP1 → PLC_1	[CPU 31	13C-2 DP	] 🕨 Distri	buted I/O	<ul> <li>DP-Mastersystem (1)</li> </ul>	: PROFIBUS_1 → HX7F	rofibus	_ 7 7	i×
							2	Topology view 🔥 N	letwork view	Device view	
	Device	overview									
	<b>**</b>	Module	Rack	Slot	I address	Q address	Туре	Article no.	Firmware	Comment	
		HX7Profibus	0	0	1022*		Anybus-IC PDP		Version 2.12		^
		INPUT: 32 Byte (16 word)_1	0	1	256287		INPUT: 32 Byte (16 word)				
		INPUT: 32 Byte (16 word)_2	0	2	288319		INPUT: 32 Byte (16 word)				
		INPUT: 32 Byte (16 word)_3	0	з	320351		INPUT: 32 Byte (16 word)				
		INPUT: 16 Byte (8 word)_1	0	4	352367		INPUT: 16 Byte ( 8 word)				=
		INPUT: 4 Byte ( 2 word)_1	0	5	368371		INPUT: 4 Byte ( 2 word)				
		OUTPUT: 32 Byte (16 word).	0	6		256287	OUTPUT: 32 Byte (16 word)				
31		OUTPUT: 16 Byte (8 word)_	10	7		288303	OUTPUT: 16 Byte (8 word)				
		OUTPUT: 4 Byte ( 2 word)_1	0	8		304307	OUTPUT: 4 Byte ( 2 word)				
ki č			0	9							
å -			0	10							
			0	11							

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 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:

	wagProfibusHX7_V13_SP1 > PLC_1 [CPU 313C-2 DP] > Program blocks > Hardware > SaveOutput > HD_ProfibusOutput [DB3]											
1												
	HD_ProfibusOutput											
	Name Data type Offset Start value Monitor value Retain Visible in Setpoint											
1	-00	▼ S <sup>1</sup>	tatic									
2	-	•	command	Word	0.0	16#00	16#0000		<b></b>			
з	-00	•	complex_command	Word	2.0	16#00	16#0000					
4	-00	•	platform	Word	4.0	16#1	16#0001					
5	-0	•	set_tare	Real	6.0	2.0	2.0		<b></b>			
6	-	•	set_lo	Real	10.0	0.5	0.5					
7	-00	•	outputs	Word	14.0	16#03	16#0003					
8	-00	•	set_min	Real	16.0	1.1	1.1		<b></b>			
9	-	•	set_max	Real	20.0	1.4	1.4		<b></b>			
10	-00	•	set_lot	DWord	24.0	16#DE	16#0000_00DE					
11	-00	•	set_operator	Word	28.0	16#16	16#0016		<b></b>			
12	-0	•	set_article	Word	30.0	16#2	16#0002		<b></b>			
13	-00	•	set_customer	Word	32.0	16#2	16#0002					
14	-00	•	set_packeging	Word	34.0	16#01	16#0001		<b></b>			
15		•	set_formulation_process	Word	36.0	16#0	16#0000					
16		•	set_dosing_process	Word	38.0	16#0	16#0000		<b>~</b>			

RadwagProfibusHX7\_V13\_SP1 → PLC\_1 [CPU 313C-2 DP] → Program blocks → Hardware → SaveInput → HD\_ProfibusInput [

-		- <b>-</b> •								
	HD	_Prof	ibusInput							
		Name		Data type	Offset	Start value	Monitor value	Retain	Visible in	Setpoint
1	-	▼ St	atic							
2	-	•	mass 1	Real	0.0	0.0	0.0	<b></b>	<b></b>	
3	-00	•	tare 1	Real	4.0	0.0	0.0	<b></b>	<b></b>	
4	-	•	unit 1	Word	8.0	16#0	16#0002	<b></b>	<b></b>	
5	-0	•	status 1	Word	10.0	16#0	16#0007	<b></b>	<b></b>	
6	-00	•	lo 1	Real	12.0	0.0	0.5	<b></b>	<b></b>	
7	-	•	mass 2	Real 🔳	16.0	0.0	0.0	<b></b>	<b></b>	
8	-00	•	tare 2	Real	20.0	0.0	0.0	<b></b>	<b></b>	
9	-00	•	unit 2	Word	24.0	16#0	16#0000	<b></b>	<b></b>	
10	-	•	status 2	Word	26.0	16#0	16#0000	<b></b>	<b></b>	
11	-00	•	lo 2	Real	28.0	0.0	0.0		<b></b>	
12	-00	•	process_status	Word	32.0	16#0	16#0000	<b></b>	<b></b>	
13	-	•	inputs	Word	34.0	16#0	16#0008	<b></b>	<b></b>	
14	-00	•	min	Real	36.0	0.0	0.0	<b></b>	<b></b>	
15	-00	•	max	Real	40.0	0.0	0.0	<b></b>	<b></b>	
16	-0	•	lot_number	DWord	44.0	16#0	16#0000_00DE	<b></b>	<b></b>	
17	-00	•	operator	Word	48.0	16#0	16#0016	<b></b>	<b></b>	
18	-00	•	article	Word	50.0	16#0	16#0002			
19	-0	•	customer	Word	52.0	16#0	16#0002	<b></b>	<b></b>	
20	-00	•	packaging	Word	54.0	16#0	16#0000	<b></b>	<b></b>	
21		•	source_warehouse	Word	56.0	16#0	16#0000	<b></b>		
22	-	•	target_warehouse	Word	58.0	16#0	16#0000	<b></b>	<b></b>	
23	-00	•	formulation	Word	60.0	16#0	16#0000			
24		•	dosing	Word	62.0	16#0	16#0000	<b></b>		

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 oraz zapisu rejestrów "stan wyjść" i "min".

_										
	OE	31								
		Name		Data type	Offset	Default value	Comment			
1	-00	▼ Te	mp							
2	-	•	Temp_0	Byte 🔳	0.0					
3	-00	•	Temp_1	Byte	1.0					
_						<b>A</b>				
CAL	L									
•		Netwo	Drk Z:							
_	C	omme	ent							
		1	CALL DPRD_DAT							
		2	LADDR :=W#1	.6#100				W#16#100		
	3         RET_VAL :="err read"         %MW4           4         RECORD :="HD_ProfibusInput"."mass 1"         %DB2.DBD0									
	5									
		6								
		7								
		8								
		9								
		10								
		11								
•		Netwo	ork 3:							
	C	Comme	ent							
		1	CALL DDDD DAT							
		1	CALL DPRD_DAT	C1104						
		2	DET VAL	.0#104				W#10#104		
		4	REI_VAL := er	r reau DesfibusTeeut" "	tono 1"			SPIN4 SDP2 DPD4		
		5	RECORD := HD	_Profibusinput"."	tare 1			SUD2.UDU4		
		6								
		7								
		8								
		9								
		-								
	C	lomme	ent							
		1								
		1	CALL DPWR_DAI							
		4	LADDR :=W#	16#10E				W#16#10E		
		3	RECORD :="H	D_ProfibusOutput	".outpu	123		<pre>%DB3.DBW14</pre>		
		4	RET_VAL :="e	err write"				*MW8		
		5								
		6								
		8								
		9								
		10								
		11								
•	1	Netwo	ork 29:							
	C	omme	ent							
		1	CALL DEWE DAT							
		2	LADDRU4	16#110				W#16#110		
		3	DECODD	D ProfibusOutrot	" act -	in.		SDB3 DBD16		
		4	RECORD :="H	m_rrorrousoutput	.set_M			SUDD.UDU16		
		5	KEI_VAL :="e	II WEILE				-ST-IM O		
		0								
		6								

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.

