

Sterownik PLC
ELPM-8DI8DO z aplikacją
„ELPM-8DI8DOasRoleta”

wersja v1.1 2018 09 09

1. Opis ogólny

Moduły ELPM-... są to sterowniki swobodnie programowalne PLC (bez zegara RTC). Mogą być używane jako autonomiczne sterowniki lub jako moduły rozszerzeń do rozbudowy innych sterowników PLC o dodatkową funkcjonalność. Ich kompaktowa budowa, duży zasób wejść i wyjść pozwala budować duże systemy sterowania o kompaktowych rozmiarach. Cechy szczególne urządzenia to: nieograniczone możliwości konfiguracji, obsługa protokołów BACnet oraz Modbus, zarządzanie urządzeniami podrzędnymi (falowniki, inne moduły), obsługa wszystkich typów HMI z komunikacją na łączu RS485 producenta EL-Piast.

Moduł ELPM-8DI8DO z aplikacją ELPM-8DI8DOasRoleta pozwala na sterowanie roletą z dwóch przycisków monostabilnych, z nadrzędnego sterownika lub z systemu BMS za pomocą listy zmiennych Modbus lub BacNet. Do działania modułu w sterowaniu z nadrzędnego systemu nie są wymagane przyciski sterujące, stanowią one jedynie opcję dodatkową.

Pojedyncze wciśnięcie przycisku "góra" podłączonego do wejścia DI1 (krótsze niż 0,5s) powoduje przełączenie wyjścia Re1 z wyłączonego na załączony i otwieranie rolet aż do 100% lub do chwili w której nastąpi kolejne pojedyncze wciśnięcie przycisku "góra", analogiczna sytuacja jest dla DI3 - Re3, DI5 - Re5, DI7 - Re7.

Pojedyncze wciśnięcie przycisku "dół" podłączonego do wejścia DI2 (krótsze niż 0,5s) powoduje przełączenie wyjścia Re2 z wyłączonego na załączony i zamykanie rolet aż do 0% lub do chwili w której nastąpi kolejne pojedyncze wciśnięcie przycisku "dół", analogiczna sytuacja jest dla DI4 - Re4, DI6 - Re6, DI8 - Re8.

Przytrzymanie przycisku "góra" podłączonego do wejścia DI1 (powyżej 1s) powoduje przełączenie wyjścia Re1 z wyłączonego na załączony i otwieranie rolet aż do 100% lub do chwili w której nastąpi puszczenie przycisku "góra", analogiczna sytuacja jest dla DI3 - Re3, DI5 - Re5, DI7 - Re7.

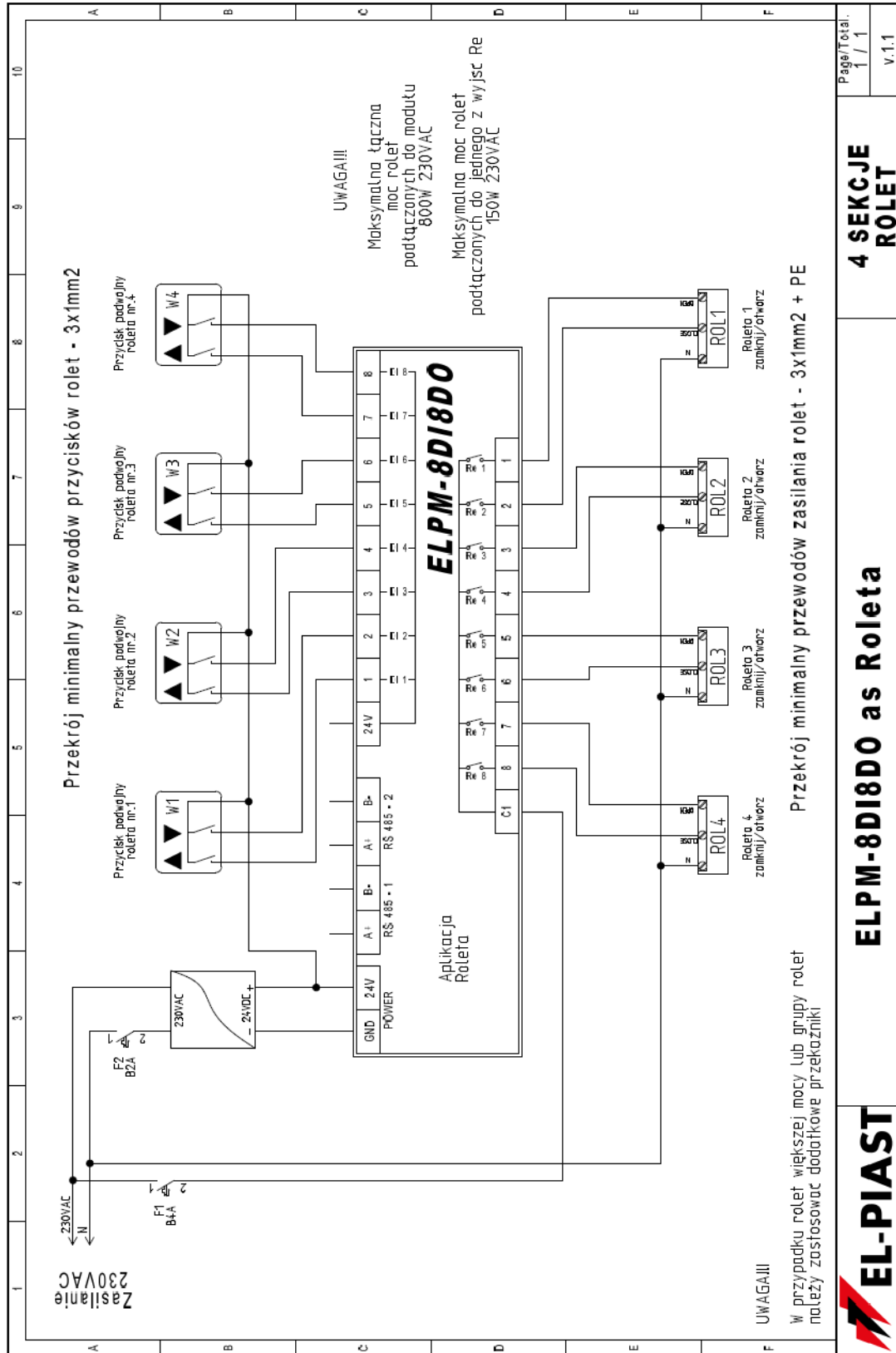
Przytrzymanie przycisku "dół" podłączonego do wejścia DI2 (powyżej 1s) powoduje przełączenie wyjścia Re2 z wyłączonego na załączony i zamykanie rolet aż do 0% lub do chwili w której nastąpi puszczenie przycisku "dół", analogiczna sytuacja jest dla DI4 - Re4, DI6 - Re6, DI8 - Re8.

Sterowanie roletami oparto o czas pełnego otwarcia rolet który należy zapisać w sterowniku za pomocą zmiennych Modbus, BacNet lub za pomocą opcjonalnego zadajnika HMI Compact lub Advanced podłączonego do łącza RS485-1.

Każdorazowe załączenie zasilania powoduje kalibrację rolet do położenia pełnego zamknięcia.

UWAGA!!! Do prawidłowego działania modułu w funkcji sterowania oświetleniem wymagane jest zasilanie 24VDC

2. Schemat połączeń



3. Dane techniczne

	ELPM-8DI8DO
Napięcie zasilania	24 V DC
Montaż	szyna TS35
Złącza komunikacyjne:	2 x RS-485
Protokoły komunikacyjne:	Modbus RTU, BACnet
Sygnalizacja stanu pracy WE/WY na diodach LED	✓
Sygnalizacja alarmu i komunikacji na diodach LED	✓
Konfiguracja trybu pracy na przelączniku	✓
Przełączniki typu NO	8 szt. 5A
Wejścia cyfrowe	8 szt. (napięcie znamionowe 24VDC)
Wejścia analogowe (napięciowe/prądowe)	—
Wyjścia analogowe	—
Wejścia temperaturowe	—
Wymiary	22,5 x 109 x 124

4. Nastawa parametrów komunikacji

Sterowniki – moduły serii ELPM-... stanowią rozszerzenie dedykowane dla sterowników posiadających port komunikacyjny Modbus RTU lub BACnet MS-TP w standardzie RS-485. Adres modułu ustalany jest za pomocą przełącznika typu DIP Switch w zakresie 1-256.

Nastawę prędkości komunikacji, parzystości, bitów stopu i ustawienie protokołu komunikacji wykonuje się poprzez zworki przełącznika DIP Switch i wykonanie następujących czynności:

- wyłączyć zasilanie modułu, ustawić wszystkie zworki na On i włączyć urządzenie
- po włączeniu przestawić wszystkie zworki na Off (diody COM i ALR powinny zacząć migać na zmianę)
- ustawić na zworkach wymagane parametry zgodnie z opisem:
 - a) zworki 1-4: prędkość komunikacji (0 - 2k4, 1 - 4k8, 2 - 9k6, 3 - 14k4, 4 - 19k2, 5 - 28k8, 6 - 38k4, 7 - 57k6, 8 - 76k8, 9 - 115k2, 10 - 230k4, 11 - 250k, 12 - 500k, 13 - 500k, 14 - 500k, 15 - 500k)
 - b) zworki 5-6: parzystość: (0 - brak, 1 - even, 2 - odd, 3 - odd)
 - c) zworka 7 - bity stopu (0 - 1bit, 1 - 2bity)
 - d) zworka 8 - tryb (0 - Modbus / ELPBus, 1 - BACnet MS/TP)

Przykład dla konfiguracji:

- prędkość komunikacji 9k6
- brak parzystości
- 2 bity stopu
- Modbus/ELPBus

d)	c)	b)	a)					
8	7	6	5	4	3	2	1	
	X					X		On
X		X	X	X	X		X	Off

Po ustawieniu należy wyłączyć moduł oraz ustawić za pomocą zworek przełącznika DIP Switch adres w zakresie 1-256 .

Przykład ustawienia adresu „,1”

8	7	6	5	4	3	2	1	
							X	On
X	X	X	X	X	X	X		Off

5. Komunikacja Modbus RTU

Sterownik – moduł serii ELPM-... posiada implementację protokołu Modbus RTU. Aby dokonać sprzęgu sieciowego należy podłączyć magistralę RS-485 do portu RS485-1 na listwie sterownika. Nastawę parametrów komunikacji opisano w pkt. 3 niniejszej instrukcji.

Domyślne parametry komunikacji:

- prędkość transmisji 9600 bps
(możliwość zmiany z poziomu nbudowanego lub zewnętrznego HMI)
- 8 bitów ramki
- 2 bity stopu
- brak parzystości

Wszystkie zmienne są 32-bitowymi wartościami które są przedstawiane w protokole Modbus jako *Input*, *Coil*, *Holding Register* lub *Input Register* w różnych przestrzeniach adresowych.

Odczyt i zapis danych typu *Input* i *Coil*:

Każda zmienna jest 32-bitową wartością. Dla przykładu zmienna o adresie w tabeli 0x0008 udostępnia bity pod adresami binarnymi 8*32 ... 9*32-1 dla *Input* i *Coil* w standardzie Modbus.

Odczyt i zapis danych typu *Holding Register* i *Input Register* :

Zmienne w tej postaci, dla ułatwienia integracji z systemami BMS, udostępniane są w różnych przestrzeniach adresowych.

- 0x0000 ... 0x1000 – tradycyjna reprezentacja wg. informacji poniżej
 - Multistate – wyszczególnionym całkowitym wartościom zmiennej odpowiadają opisane stany
 - Decimal – 32-bitowa wartość zmiennej jest traktowana jako typ całkowity ze znakiem
 - Fixed – typ stałopozycyjny w którym 8 najmniej znaczących bitów przeznaczone jest na część ułamkową, natomiast pozostałe 24 bity to część całkowita ze znakiem. Wynika z tego że dokładność wartości Fixed to 1/256. Aby przeskalować wartość reprezentowaną w postaci Fixed na docelową (właściwą) należy przemnożyć ją przez $1/256 = 0,00390625$.
- 0x1000 ... 0x2000 – zmienne formatu Fixed przedstawione jako wartości całkowite z pominięciem ułamka
- 0x2000 ... 0x3000 – zmienne formatu Fixed przedstawione jako wartości z dokładnością do jednego miejsca po przecinku w formacie dziesiętnym. Wartość 20,67 przedstawiona jest jako 206
- 0x3000 ... 0x4000 – zmienne formatu Fixed przedstawione jako wartości z dokładnością do dwóch miejsc po przecinku w formacie dziesiętnym. Wartość 20,67 przedstawiona jest jako 2067
- 0x4000 ... 0x5000 – analogicznie jak dla przestrzeni 0x0000 ... 0x1000 lecz zmienne są traktowane jako wartości 16-bitowe. Oznacza to że starsze 16-bitów nie są uwzględniane. Adresy należy podzielić przez dwa. Przykładowo zmienna z tabeli o adresie 0x0124 jest dostępna w formacie 16-bitowym pod adresem Modbus 0x4092
- 0x5000 ... 0x6000 – analogicznie jak dla przestrzeni 0x1000 ... 0x2000 lecz zmienne są traktowane jako wartości 16-bitowe. Oznacza to że starsze 16-bitów nie są uwzględniane.

- Adresy należy podzielić przez dwa. Przykładowo zmienna z tabeli o adresie 0x0124 jest dostępna w formacie 16-bitowym pod adresem Modbus 0x4092
- 0x6000 ... 0x7000 – analogicznie jak dla przestrzeni 0x2000 ... 0x3000 lecz zmienne są traktowane jako wartości 16-bitowe. Oznacza to że starsze 16-bitów nie są uwzględniane. Adresy należy podzielić przez dwa. Przykładowo zmienna z tabeli o adresie 0x0124 jest dostępna w formacie 16-bitowym pod adresem Modbus 0x4092
 - 0x7000 ... 0x8000 – analogicznie jak dla przestrzeni 0x2000 ... 0x3000 lecz zmienne są traktowane jako wartości 16-bitowe. Oznacza to że starsze 16-bitów nie są uwzględniane. Adresy należy podzielić przez dwa. Przykładowo zmienna z tabeli o adresie 0x0124 jest dostępna w formacie 16-bitowym pod adresem Modbus 0x4092

Zmienne w reprezentacji Multistate oraz Decimal nie należy używać w przestrzeniach adresowych 0x1000 ... 0x4000 oraz 0x5000 ... 0x8000 gdyż traci się najmniej znaczące 8 bitów każdej ze zmiennych. Adresy z tabeli są przeliczane dla protokołu Modbus w następujący sposób:

Przeźreźń adresowa	Obliczanie adresu
0x0000 ... 0x1000	Modbus Adres = Adr.
0x1000 ... 0x2000	Modbus Adres = 0x1000 + Adr.
0x2000 ... 0x3000	Modbus Adres = 0x2000 + Adr.
0x3000 ... 0x4000	Modbus Adres = 0x3000 + Adr.
0x4000 ... 0x5000	Modbus Adres = 0x4000 + (Adr. / 2)
0x5000 ... 0x6000	Modbus Adres = 0x5000 + (Adr. / 2)
0x6000 ... 0x7000	Modbus Adres = 0x6000 + (Adr. / 2)
0x7000 ... 0x8000	Modbus Adres = 0x7000 + (Adr. / 2)

UWAGA: nie można dokonać zapisu pojedynczego rejestru 16-bitowego w przestrzeniach adresowych 0x1000 ... 0x4000. W tym wypadku należy zapisywać rejestry parami komendą *Preset Multiple Registers* (0x10) na którą składa się pełna wartość 32-bitowej zmiennej. Oznacza to że adres początku zapisu oraz ilość rejestrów musi być liczbą parzystą.

6. Komunikacja BACnet MS-TP

Zmienne BACnet należy wyszukać po podłączeniu zasilonego sterownika oraz wprowadzeniu odpowiednich ustawień sieci BACnet (patrz pkt.4).

7. Adresy zmiennych Modbus RTU oraz BacNet MS-TP

Zmienne sterujące (bez limitu przełączeń, bez pamięci stanu z przed resetu zasilania) **Wg. ustawień fabrycznych nastawy oznaczone pogrubioną czcionką zmieniają stan wyjść.**

Adres (DEC)		Zmienna	Opis	Stany	Typ		Odczyt [R] /Zapis [W]
BacNet	Modbus				BacNet	Modbus	
0	0	SetProc1	Nastawa wysterowania rolety 1	1% = 256 (10% = $10 \cdot 256 = 2560 = 0xA00$)	AV	Register	R / W
1	2	SetProc2	Nastawa wysterowania rolety 2	1% = 256 (10% = $10 \cdot 256 = 2560 = 0xA00$)	AV	Register	R / W
2	4	SetProc3	Nastawa wysterowania rolety 3	1% = 256 (10% = $10 \cdot 256 = 2560 = 0xA00$)	AV	Register	R / W
3	6	SetProc4	Nastawa wysterowania rolety 4	1% = 256 (10% = $10 \cdot 256 = 2560 = 0xA00$)	AV	Register	R / W

Zmienne do odczytu wysterowania oraz nastawy czasu otwarcia

Adres (DEC)		Zmienna	Opis	Stany	Typ		Odczyt [R] /Zapis [W]
BacNet	Modbus				BacNet	Modbus	
4	8	ActProc1	Aktualne wysterowanie rolety 1	1% = 256 (10% = $10 \cdot 256 = 2560 = 0xA00$)	AV	Register	R
5	10	ActProc2	Aktualne wysterowanie rolety 2	1% = 256 (10% = $10 \cdot 256 = 2560 = 0xA00$)	AV	Register	R
6	12	ActProc3	Aktualne wysterowanie rolety 3	1% = 256 (10% = $10 \cdot 256 = 2560 = 0xA00$)	AV	Register	R
7	14	ActProc4	Aktualne wysterowanie rolety 4	1% = 256 (10% = $10 \cdot 256 = 2560 = 0xA00$)	AV	Register	R
8	16	OpenTime1	Nastawa czasu pełnego otwarcia rolety 1	1s = 256 (22s = $22 \cdot 256 = 5632 = 0x1600$)	AV	Register	R / W
9	18	OpenTime2	Nastawa czasu pełnego otwarcia rolety 2	1s = 256 (22s = $22 \cdot 256 = 5632 = 0x1600$)	AV	Register	R / W
10	20	OpenTime3	Nastawa czasu pełnego otwarcia rolety 3	1s = 256 (22s = $22 \cdot 256 = 5632 = 0x1600$)	AV	Register	R / W
11	22	OpenTime4	Nastawa czasu pełnego otwarcia rolety 4	1s = 256 (22s = $22 \cdot 256 = 5632 = 0x1600$)	AV	Register	R / W