



Politechnika Wrocławska

Wydział Informatyki i Telekomunikacji

Informatyczne systemy automatyki

Sprawozdanie IV - Komunikacja EGD VersaMax

Autorzy:

Damian Filipowski id. 272555

Konrad Landzberg id. 272508

Przedmiot:

ISP - laboratorium

6 marca 2025

Spis treści

1	Wstęp:	2
2	Konfiguracja:	2
3	Kod programu:	4
3.1	Target 1:	4
3.2	Target 2:	5
4	Wnioski:	6

Spis rysunków

1	Target 1 konfiguracja CPU, zasilacza oraz modułów.	2
2	Target 2 konfiguracja CPU, zasilacza oraz modułów.	2
3	Target 1 adres IP	2
4	Target 2 adres IP	2
5	Konfiguracja IP wymiany typu Producent po stronie sterownika Target 1.	3
6	Konfiguracja IP wymiany typu Konsument po stronie sterownika Target 1.	3
7	Konfiguracja IP wymiany typu Producent po stronie sterownika Target 2.	3
8	Konfiguracja IP wymiany typu Konsument po stronie sterownika Target 2.	3
9	Tablica zmiennych dla wymiany po stronie Producenta (Target 2).	3
10	Tablica zmiennych dla wymiany po stronie Konsumenta (Target 1).	3
11	Tablica zmiennych dla wymiany po stronie Producenta (Target 1).	4
12	Tablica zmiennych dla wymiany po stronie Konsumenta (Target 2).	4
13	Rungi dla sterownika Target 1 cz. 1.	4
14	Rungi dla sterownika Target 1 cz. 2.	5
15	Rungi dla sterownika Target 2 cz. 1.	5
16	Rungi dla sterownika Target 2 cz. 1.	6

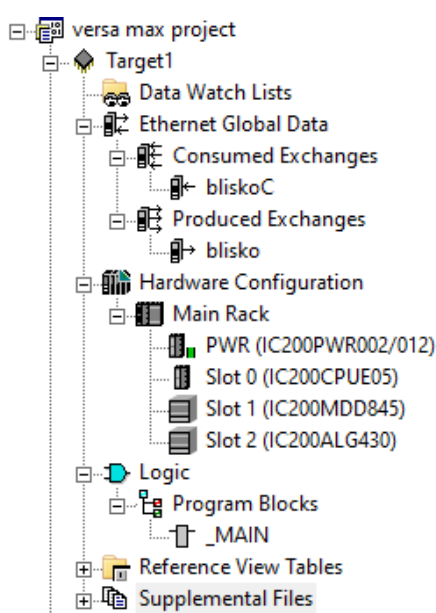
Spis tabel

1 Wstęp:

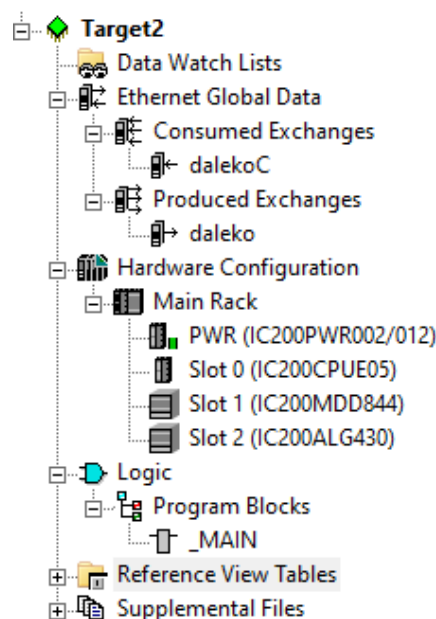
Celem laboratorium było skonfigurowanie komunikacji EGD (Ethernet Global Data) między dwoma sterownikami VersaMax w środowisku PAC Machine Edition oraz sprawdzenie poprawności wymiany danych. Zadanie obejmowało utworzenie połączenia, zdefiniowanie tablic wymienianych danych oraz implementację logiki sterowania w języku ladder.

2 Konfiguracja:

Na początku konfiguracji dodano moduł CPU oraz odpowiedni zasilacz zgodny ze sterownikiem używanym do komunikacji. Następnie dodano moduły wejść/wyjść (I/O), uwzględniając podział na szerokie i wąskie moduły.



Rysunek 1: Target 1 konfiguracja CPU, zasilacza oraz modułów.



Rysunek 2: Target 2 konfiguracja CPU, zasilacza oraz modułów.

W następnym etapie przypisaliśmy odpowiednie adresy IP dla poszczególnych sterowników Target 1 oraz Target 2.

Parameters	Values
Configuration Mode:	TCP/IP
IP Address:	192.168.22.15
Subnet Mask:	255.255.255.0
Gateway IP Address:	0.0.0.0
Status Address:	%I00105
Status Length:	80
NTP Time Server #1 IP Address	0.0.0.0
NTP Time Server #2 IP Address	0.0.0.0
NTP Time Server #3 IP Address	0.0.0.0

Rysunek 3: Target 1 adres IP

Parameters	Values
Configuration Mode:	TCP/IP
IP Address:	192.168.22.16
Subnet Mask:	255.255.255.0
Gateway IP Address:	0.0.0.0
Status Address:	%I00105
Status Length:	80
NTP Time Server #1 IP Address	0.0.0.0
NTP Time Server #2 IP Address	0.0.0.0
NTP Time Server #3 IP Address	0.0.0.0

Rysunek 4: Target 2 adres IP

W celu dodania możliwości wymiany danych za pomocą EGD dla każdego sterownika należało zdefiniować wymiane typu Producent - Konsument.

Produced Exchange	
Name	blisko
Exchange ID	2
Adapter Name	0.0
Destination Type	ip address
Destination	192.168.22.16
Produced Period	200
Reply Rate	0
Send Type	Always

Rysunek 5: Konfiguracja IP wymiany typu Producent po stronie sterownika Target 1.

Consumed Exchange	
Name	bliskoC
Producer ID	192.168.22.16
Group ID	0
Exchange ID	1
Adapter Name	0.0
Consumed Period	200
Update Timeout	10

Rysunek 6: Konfiguracja IP wymiany typu Konsument po stronie sterownika Target 1.

Produced Exchange	
Name	daleko
Exchange ID	1
Adapter Name	0.0
Destination Type	ip address
Destination	192.168.22.15
Produced Period	200
Reply Rate	0
Send Type	Always

Rysunek 7: Konfiguracja IP wymiany typu Producent po stronie sterownika Target 2.

Consumed Exchange	
Name	dalekoC
Producer ID	192.168.22.15
Group ID	0
Exchange ID	2
Adapter Name	0.0
Consumed Period	200
Update Timeout	0

Rysunek 8: Konfiguracja IP wymiany typu Konsument po stronie sterownika Target 2.

W ostatnim etapie konfiguracji dodano zmienne odpowiedzialne za wymianę danych pomiędzy sterownikami. Dla każdej wymiany danych (z tym samym Exchange ID) konieczne było zachowanie identycznej kolejności zmiennych, ich długości oraz typu zarówno po stronie producenta, jak i konsumenta.

Offset (Byte.Bit)	Variable	Ref Address	Ignore	Length	Type	Description
Status		%I00201	False	16	BIT	
0.0		%M00009	N/A	8	BIT	
1.0		%R00009	N/A	8	WORD	

Rysunek 9: Tablica zmiennych dla wymiany po stronie Producenta (Target 2).

Offset (Byte.Bit)	Variable	Ref Address	Ignore	Length	Type	Description
Status		%I00201	False	16	BIT	Status: This is where the Controller is to put the Status
TimeStamp		NOT USED	False	0	BYTE	Time Stamp: Optional place for the Controller to put the Time
0.0		%M00009	False	8	BIT	
1.0		%R00009	False	8	WORD	

Rysunek 10: Tablica zmiennych dla wymiany po stronie Konsumenta (Target 1).

W obu sterownikach zmienne przypisane do tych samych adresów spełniały identyczne funkcje. %M00009 - zmienna wewnętrzna typu bool służąca do przesyłania zmiennych cyfrowych z Target 2 na Target 1.

%R00009 - zmienna wewnętrzna typu word służąca do przesyłania zmiennych analogowych z Target 2 na Target 1.

Offset (Byte.Bit)	Variable	Ref Address	Ignore	Length	Type	Description
Status		%I00185	False	16	BIT	
0.0		%M00001	N/A	8	BIT	
1.0		%R00001	N/A	8	WORD	

Rysunek 11: Tablica zmiennych dla wymiany po stronie Producenta (Target 1).

Offset (Byte.Bit)	Variable	Ref Address	Ignore	Length	Type	Description
Status		%I00185	False	16	BIT	Status: This is where the Controller is to put the Status
Time Stamp		NOT USED	False	0	BYTE	Time Stamp: Optional place for the Controller to put the Time
0.0		%M00001	False	8	BIT	
1.0		%R00001	False	8	WORD	

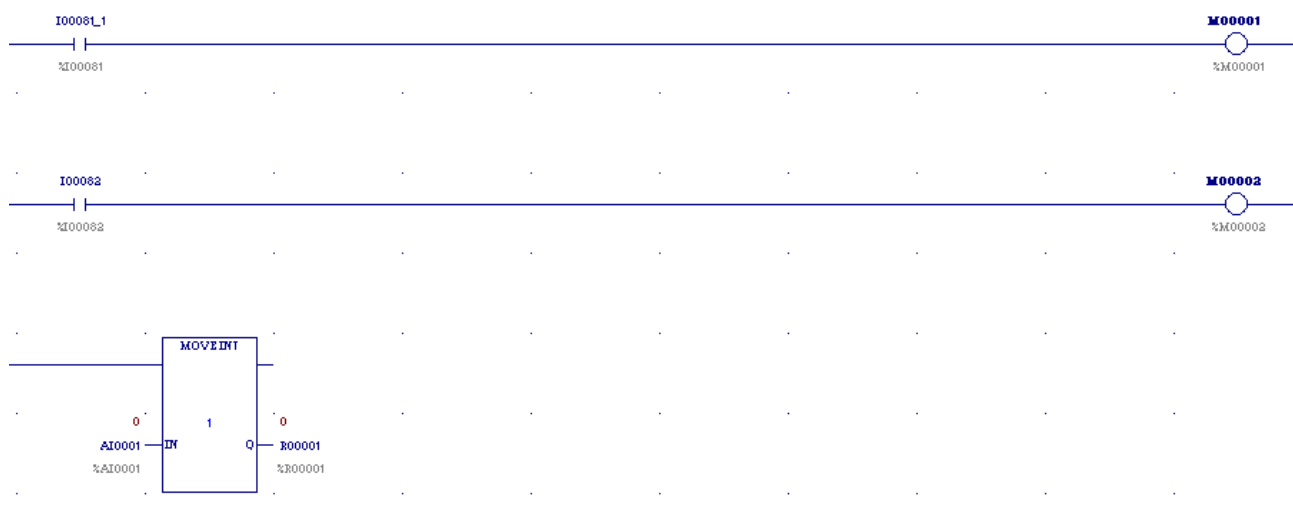
Rysunek 12: Tablica zmiennych dla wymiany po stronie Konsumenta (Target 2).

%M00001 - zmienna wewnętrzna typu bool służąca do przesyłania zmiennych cyfrowych z Target 1 na Target 2.

%R00001 - zmienna wewnętrzna typu word służąca do przesyłania zmiennych analogowych z Target 1 na Target 2.

3 Kod programu:

3.1 Target 1:



Rysunek 13: Rungi dla sterownika Target 1 cz. 1.

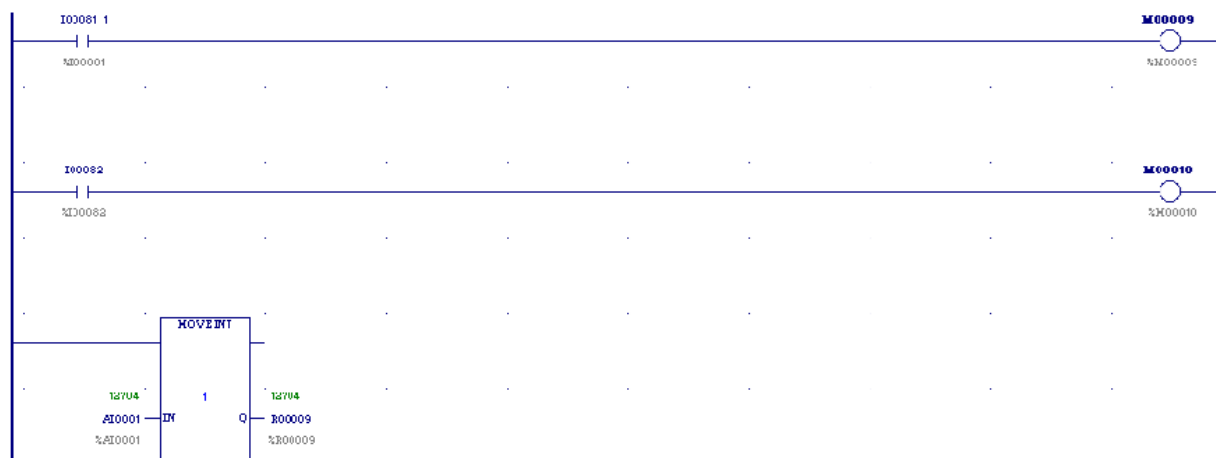
Powyższe rungi służyły do przypisania wartości pierwszych dwóch wejść cyfrowych (%I00081, %I00082) oraz pierwszego wejścia analogowego (%AI0001) z podpiętym zadajnikiem prądowym do zmiennych wewnętrznych odpowiedzialnych za komunikację.



Rysunek 14: Rungi dla sterownika Target 1 cz. 2.

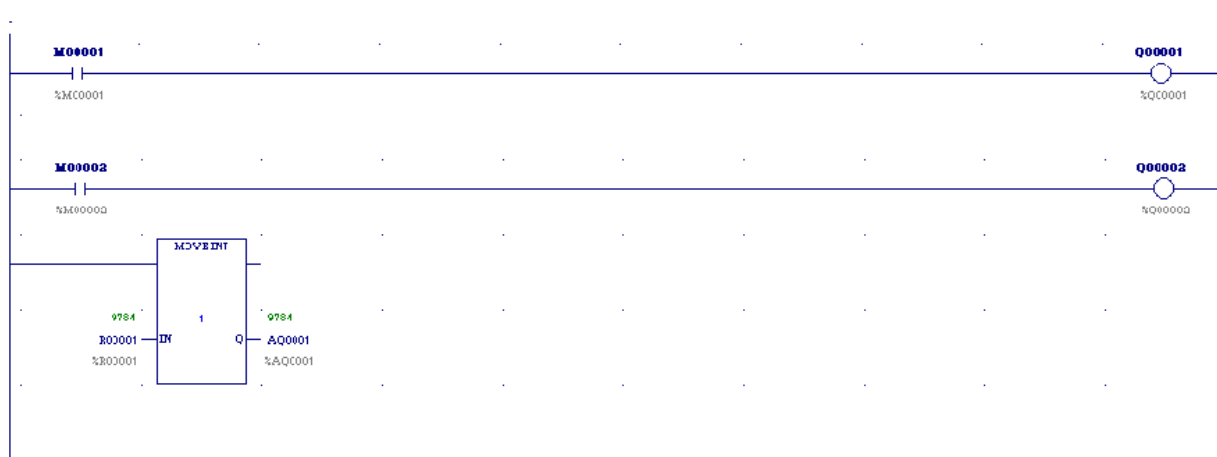
Powyższe rungi sterowały dwoma pierwszymi wyjściami sterownika Target 1 (%Q00001, %Q00002) oraz pierwszym wyjściem analogowym (%AQ00001) za pomocą wartości otrzymanych podczas komunikacji.

3.2 Target 2:



Rysunek 15: Rungi dla sterownika Target 2 cz. 1.

Powyższe rungi służyły do przypisania wartości pierwszych dwóch wejść cyfrowych (%I00081, %I00082) oraz pierwszego wejścia analogowego (%AI00001) z podpiętym zadajnikiem prądowym do zmiennych wewnętrznych odpowiedzialnych za komunikację.



Rysunek 16: Runi dla sterownika Target 2 cz. 1.

Powyższe runi sterowały dwoma pierwszymi wyjściami sterownika Target 1 (%Q00001, %Q00002) oraz pierwszym wyjściem analogowym (%AQ00001) za pomocą wartości otrzymanych podczas komunikacji.

4 Wnioski:

- Komunikacja odbywała się analogicznie w obie strony. Przesyłane były dwie zmienne cyfrowe oraz jedna analogowa. Poprawność komunikacji została sprawdzona przy użyciu amperomierzy podpiętych do wyjść analogowych sterowników oraz diod na sterownikach. Cała komunikacja odbyła się poprawnie w obie strony.
- Jedną z zalet komunikacji EGD jest jej automatyczne przesyłanie danych, co eliminuje konieczność korzystania z dedykowanych bloków w kodzie programu, takich jak w przypadku komunikacji typu „GET-PUT”.
- Komunikacja EGD opiera się na technologii Ethernet dzięki czemu możliwe jest przesyłanie dużych ilości danych w czasie rzeczywistym.