Modbus RTU (VersaMax)			
Julita Wójcik **Jakub Szczypek**	**26 IV 2022**	**wtorek, 19:45**	**3A**

Celem ćwiczenia jest zrealizowanie komunikacji za pomocą protokołu MODBUS RTU na platformie VersaMax.

Realizacja zadań

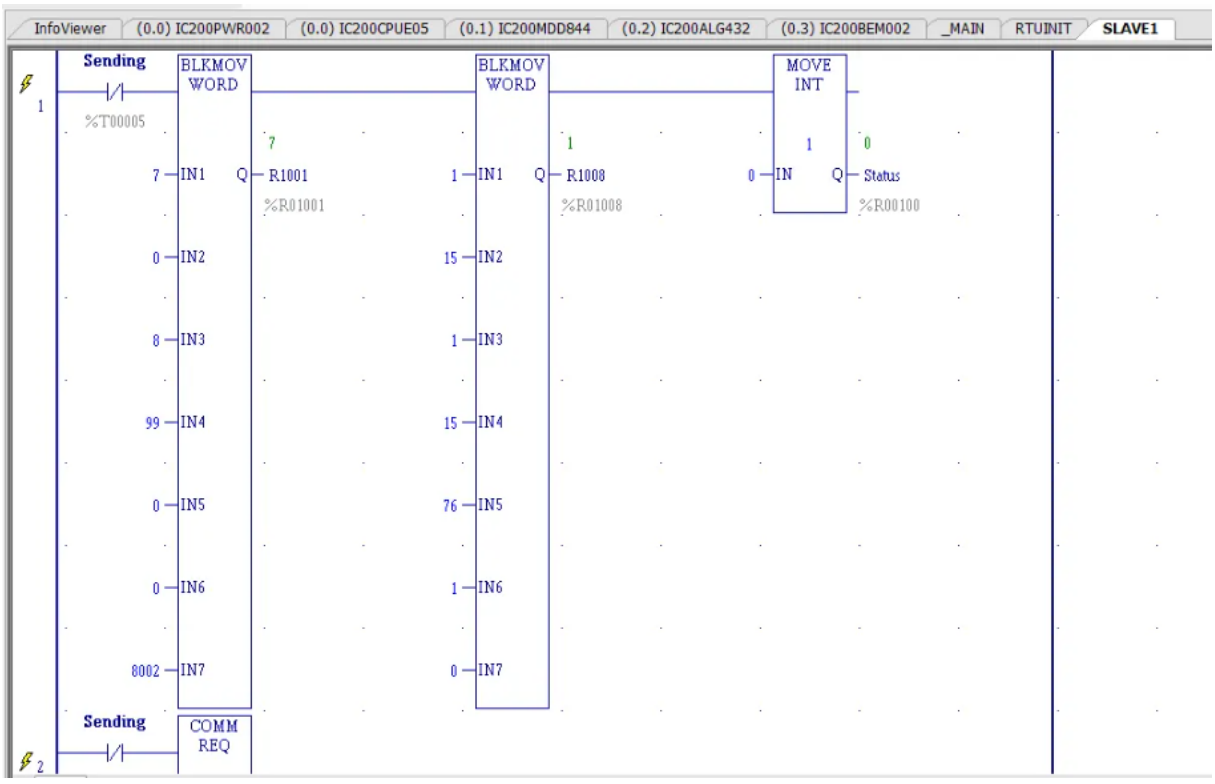
1.1. Ustawienie pierwsze 16 cewek Slave'a na wartość 1 za pomocą jednego polecenia COMMREQ

W celu realizacji zadania uzupełniono odpowiednie parametry w drugim bloku BLKMOV WORD w podprogramie Slave :

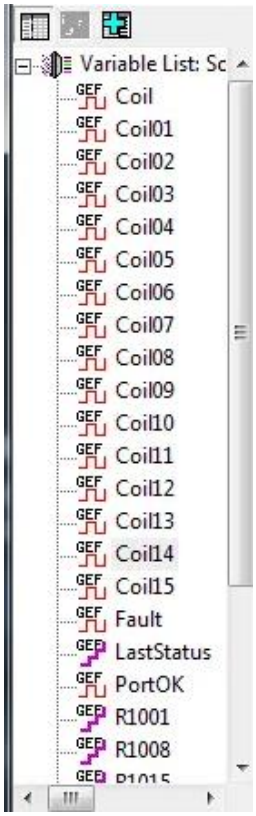
- Input 2 – funkcją MODBUS nr 15
- Input 4 – liczbą cewek

W celu szybszego utworzenia potrzebnej liczby zmiennych – cewek wykorzystano funkcję Duplicate w zakładce Variables/PPM

Gotowy bloczek przedstawiono na Rysunku 1. Przygotowane zmienne przedstawiono na Rysunku 2.

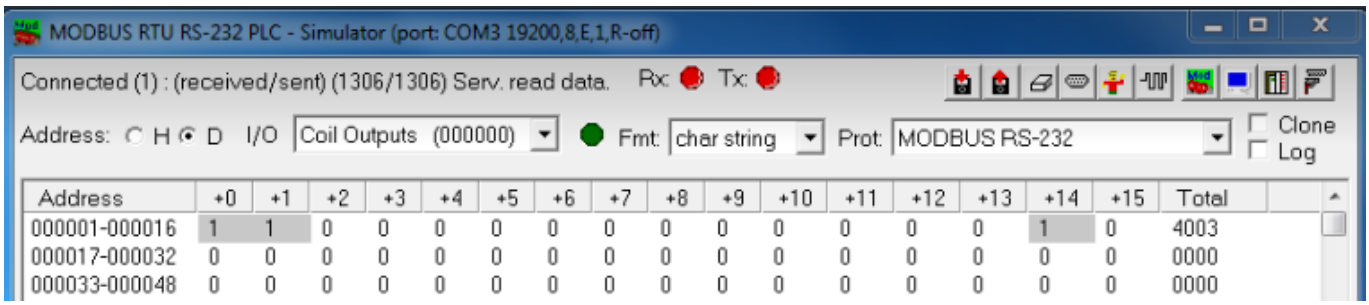


Rys. 1. Przygotowany bloczek BLKMOV WORD do realizacji zadania



Rys. 2. Zmienne Coil

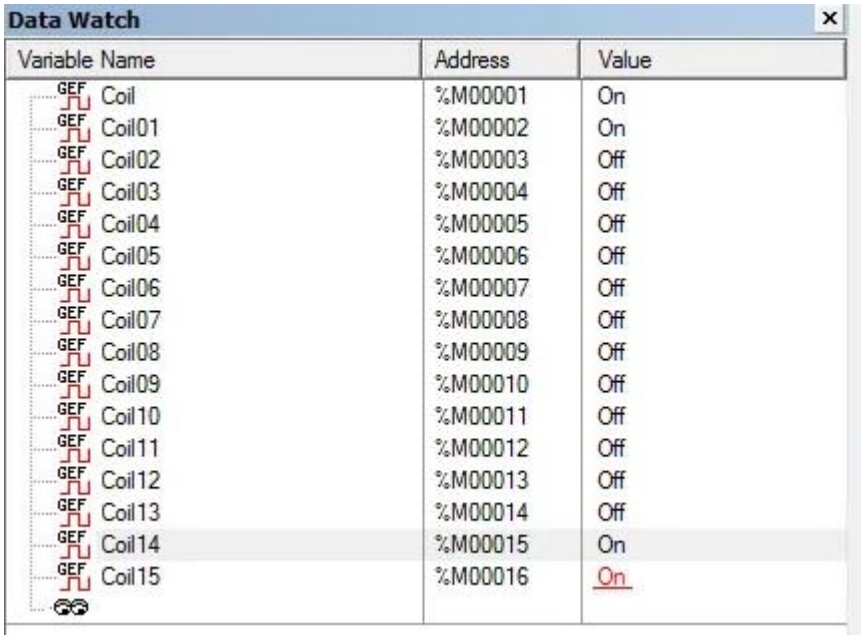
Rozpoczęto symulację wykorzystując oprogramowanie ModRSsim2 – okno Proficy zostało przedstawione na Rysunku 3. Zmieniono typ wyjścia na Coil Ouputs. W celu sprawdzenia poprawności komunikacji zmieniano wartość tagów Coil na 1 – używano okienka Data Watch przedstawionego na Rysunku 4.



Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11	+12	+13	+14	+15	Total
000001-000016	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	4003
000017-000032	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
000033-000048	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

Rys. 3. Okno symulatora

Cewki, które należy obserwować zaczynają się adresem 000001-000016



Variable Name	Address	Value
Coil	%M00001	On
Coil01	%M00002	On
Coil02	%M00003	Off
Coil03	%M00004	Off
Coil04	%M00005	Off
Coil05	%M00006	Off
Coil06	%M00007	Off
Coil07	%M00008	Off
Coil08	%M00009	Off
Coil09	%M00010	Off
Coil10	%M00011	Off
Coil11	%M00012	Off
Coil12	%M00013	Off
Coil13	%M00014	Off
Coil14	%M00015	On
Coil15	%M00016	On

Rys. 4.Okno Data Watch

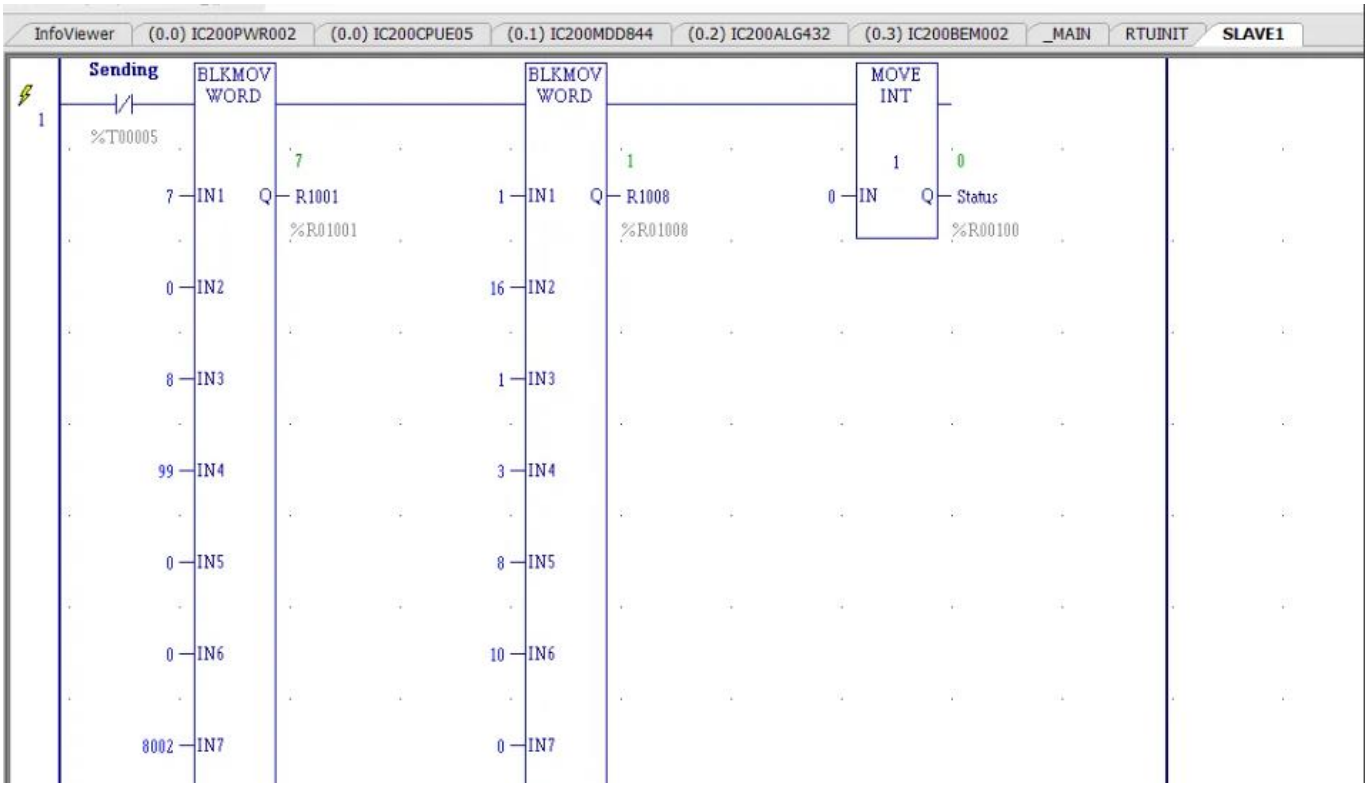
Zaobserwowano, że wraz ze zmianą wartości taga danej cewki (value = On) w oknie Proficy zmieniał się status cewki o danym numerze na wartość 1 . Niestety, podczas realizacji zadania wkradł się błąd – w bloczku BLKMOV WORD przekazano 15 cewek zamiast 16, wynikiem czego próba zmiany szesnastej cewki skutkowała błędem w postaci podkreślenia na czerwono.
Jednoczesne pojawienie się ciągu ‘jedynek’ nie było możliwe, gdyż zmiany w Data Watch odbywały się pojedynczo – po jednej cewce. Aby uniknąć tego problemu należałoby dopisać dodatkową funkcję.

1.2. Przesłanie do Slave'a wiadomości tekstowej "AGH" za pomocą jednego polecenia COMMREQ.

W celu realizacji zadania uzupełniono odpowiednie parametry w drugim bloku BLKMOV WORD w podprogramie Slave :

- Input 2 – funkcją MODBUS nr 16
- Input 4 – liczbą rejestrów – 3
- Input 5 – typ obszaru pamięci w PLC - odczytany z dokumentacji dla rejestrów - 8
- Input 6 – adres obrazu pamięci w PLC

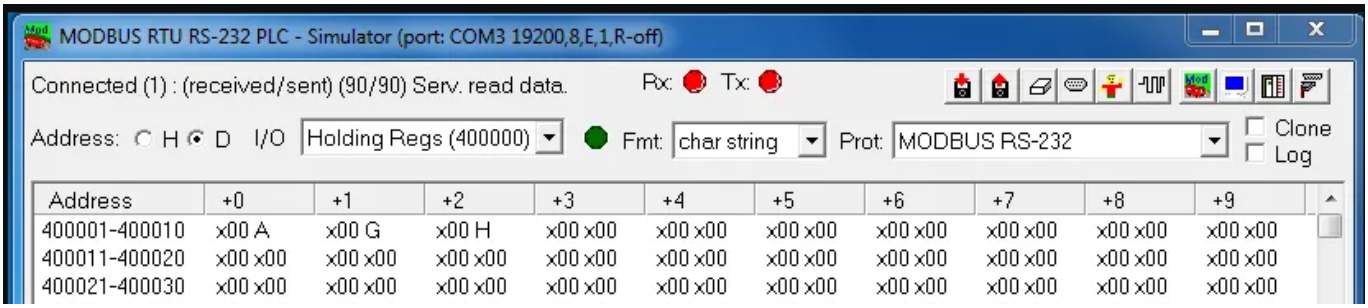
Przygotowany bloczek przedstawiono na Rysunku 5.



Rys. 5. Przygotowany bloczek BLKMOV WORD do realizacji zadania

Każdemu z trzech rejestrów przypisano wartość litery w kodzie ASCII : A – 65, G -71, H-72

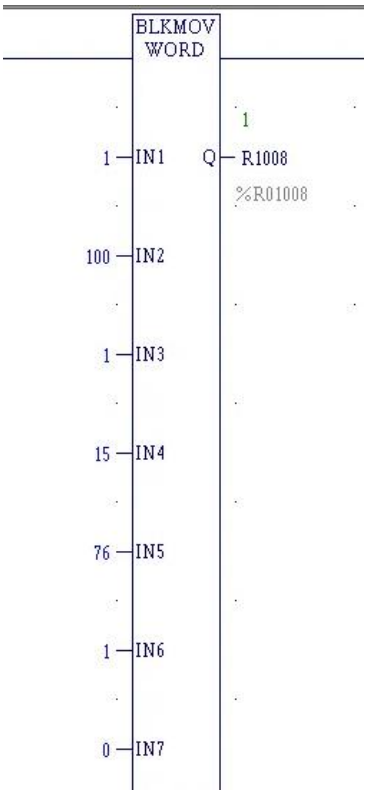
Rozpoczęto symulację, pamiętając o zmianie typu I/O na Holding Regs i formacie na char string



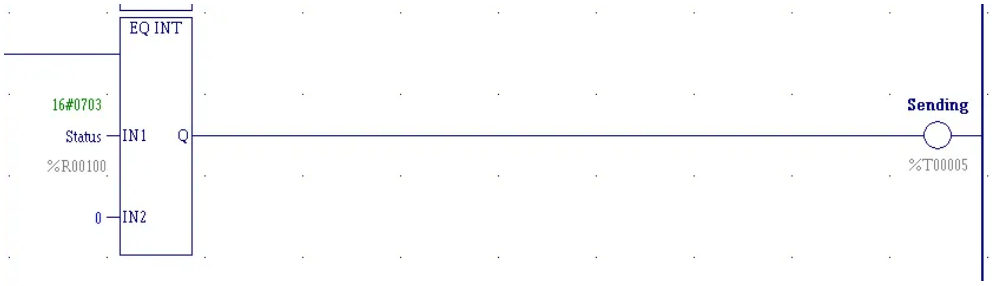
1.3. Sprawdzenie kodów błędów w przypadku

- Wysłania niepoprawnego kodu funkcji MODBUS

W celu realizacji zadania w drugim bloku BLKMOV WORD w podprogramie Slave zmieniono numer funkcji na 100 – nie ma takiego numeru. Przygotowany bloczek przedstawiono na Rysunku 6., otrzymany status błędu w bloczku EQ INT – 16#0703 na Rysunku 7. , fragment tabeli z kodami błędów na Rysunku 8.



Rysunek 6. Przygotowany bloczek



Rysunek 7. Otrzymany kod błędu

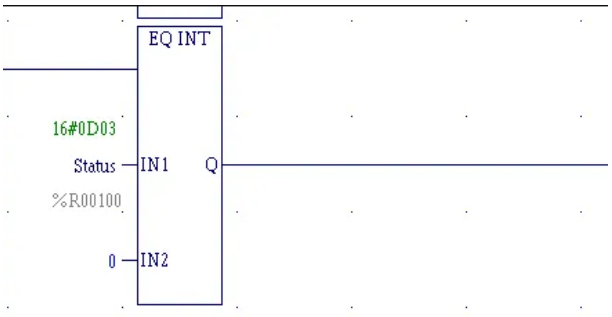
PARAMETER_ ERROR	NOT_A_COMMREQ	0103h	A Message received from CPU is not a COMMREQ.
	WAIT_COMMREQ	0203h	WAIT-mode COMMREQs are not supported.
	UNSUPP_COMMREQ_CMD	0303h	The COMMREQ Command is unsupported.
	COMMREQ_LEN_INVALID	0403h	The COMMREQ data length is too small.
	PORT_DATA_INVALID	0503h	Port Setup COMMREQ data is invalid.
	DEV_ADDRESS_INVALID	0603h	The RTU slave device is invalid.
	FUNC_CODE_UNSUPPORTED	0703h	The RTU function code is not supported.
	FUNC_INVALID_FOR_BCAST	0803h	The specified function code requires a response.

Rysunek 8. Kody błędów zaczerpnięte z dokumentacji

Program zinterpretował błąd jako FUNC_CODE_UNSUPPORTED, co jest sensowną odpowiedzią i ułatwia znalezienie błędu.

- ustawienia niepoprawnego zakresu danych

Zapisano 2000 cewek w bloczku BLKMOV WORD, nie tworząc tyle zmiennych. Otrzymany kod błędu przedstawiony na Rysunku 9, a interpretację błędu zaczerpniętą z dokumentacji na Rysunku 10.



Rysunek 9. Otrzymany kod błędu

DATA_MEM_OFFSET_INVALID	0D03h	The specified memory location for the data source/destination is invalid for local PLC CPU.

Rysunek 10. Opis błędu na podstawie dokumentacji

Wnioski

Protokół MODBus jest intuicyjny w obsłudze. Należy jednak zwracać uwagę na typ obszaru pamięci danych w PLC, ponieważ różnią się one między rejestrami, analogowymi wejściami/wyjściami, czy dyskretnymi wejściami/wyjściami – konieczne jest poprawny dobór parametrów. MODbus zwraca także klarowne odpowiedzi na błędy, dzięki którym łatwiej znaleźć je w programie. FBD jest językiem graficznym, więc cechuje go intuicyjność i łatwość programowania, zaletą jest równoległe wykonywanie instrukcji.