

## Spis treści

<b>1</b>	<b>Opis zagadnienia omawianego w dokumencie</b>	<b>2</b>
<b>2</b>	<b>Komunikacja Modbus TCP</b>	<b>3</b>
<b>3</b>	<b>Wstęp do nowego projektu</b>	<b>4</b>
3.1	Nowy projekt	3
3.2	Dodanie nowego urządzenia	5
<b>4</b>	<b>Modbus Serwer</b>	<b>6</b>
4.1	Konfiguracja dwóch połączeń w jednym sterowniku	6
4.2	Wyjaśnienie parametrów wejściowych „MB_SERVER”	8
<b>5</b>	<b>Modbus</b>	<b>9</b>
5.1	Konfiguracja dwóch połączeń w jednym sterowniku „MB_CLIENT”	9
5.2	Wyjaśnienie parametrów wejściowych „MB_CLIENT”	13
<b>6</b>	<b>Watch tables</b>	<b>13</b>
<b>7</b>	<b>Zakres adresów</b>	<b>16</b>
<b>8</b>	<b>Kody warunkowe</b>	<b>17</b>
8.1	Wartość „STATUS” dla „MB_SERVER”	17
8.2	Wartość „STATUS” dla „MB_CLIENT”	17

# 1 Opis zagadnienia omawianego w dokumencie

W tym dokumencie opisano sposób uruchomienia komunikacji w protokole modbus TCP dla sterowników S7-1200.

Minimalna konfiguracja składa się z dowolnego CPU S7-1200.

Wykaz urządzeń:

Urządzenie	Liczba	Numer katalogowy
Sterownik SIMATIC S7-1200 , model CPU 1214C DC/DC/DC	1	6ES7 214-1AE30-0XB0
Sterownik SIMATIC S7-1200 , model CPU 1212C AC/DC/RLY	1	6ES7 212-1BD30-0XB0

Wykaz oprogramowania:

Oprogramowanie	Liczba	Numer katalogowy
STEP 7 Basic v11	1	6ES7 822-0AA01-0YA0

Dodatkowe akcesoria:

Urządzenie	Liczba	Numer katalogowy
Kabel SIMATIC NET PROFIBUS	1	6XV 1830-0EH10
Wtyczka 90 stopni PROFIBUS	2	6ES7 972-0BB52-0XA0
Switch ethernetowy	1	6GK7277-1AA10-0AA0

## 2 Komunikacja Modbus TCP

Modbus TCP (Transmission Control Protocol) jest standardowym protokołem komunikacyjnym wykorzystującym interfejs PROFINET w sterowniku dla komunikacji TCP/IP. Nie są potrzebne żadne dodatkowe moduły komunikacyjne.

Modbus TCP bazuje na połączeniach Open User Communication (OUC) jako protokół komunikacyjny Modbus. Można skonfigurować kilka połączeń klient-serwer, oprócz połączenia między S7-1200 i STEP 7. Możliwa jest jednoczesna obsługa różnych typów połączeń (S7-connection, HMI-connection...), których liczba jest ściśle określona (patrz FAQ:36932465 lub manual S7-1200).

Każde połączenie „MB\_SERVER” musi posiadać przypisany unikalny blok danych typu instancje DB oraz numer portu IP. Możliwe jest tylko jedno połączenie na każdy port. „MB\_SERVER” musi zostać wywołany indywidualnie dla każdego połączenia.

Uwaga:

Modbus TCP może poprawnie pracować tylko ze sterownikiem z firmware’em w wersji v2.1 i oprogramowaniem STEP 7 V11 SP1 lub wyższej. Próba wywołania instrukcji Modbus na sterowniku ze starszą wersją firmware’u spowoduje błąd.

Klient (master) Modbusa TCP musi kontrolować połączenie klient-serwer za pomocą parametru DISCONNECT. Podstawowe akcje klienta Modbusa wyszczególniono poniżej:

1. Inicjalizacja połączenia do poszczególnego serwera (slave’a) poprzez adres IP i numer portu IP.
2. Inicjalizacja klienta do transmisji wiadomości Modbus i otrzymywania odpowiedzi od serwera.
3. W razie potrzeby inicjalizacja odłączenia klienta i serwera w celu połączenia z innym serwerem.

### Dostępne instrukcje Modbus TCP

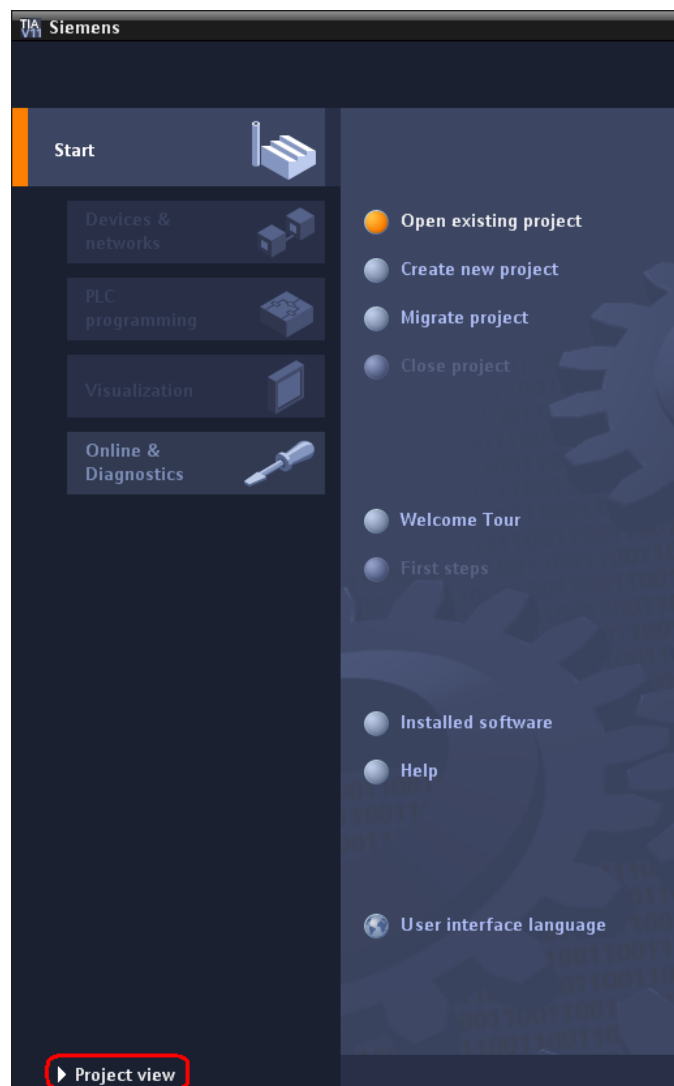
- MB\_CLIENT: tworzy połączenie TCP klient-serwer, wysyła polecenia, otrzymuje odpowiedzi i kontroluje przerwanie połączenia od serwera.
- MB\_SERVER: łączy się z klientem Modbusa TCP na żądanie, otrzymuje wiadomości Modbus i wysyła odpowiedź.

## 3 Wstęp do nowego projektu

### 3.1 Nowy projekt

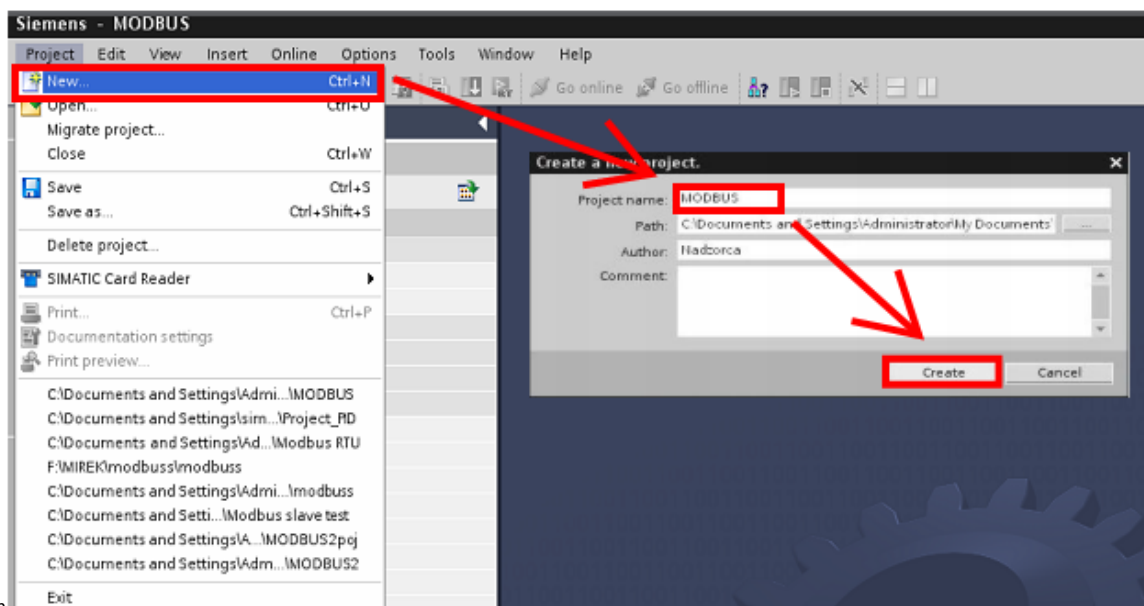
W celu utworzenia nowego projektu należy uruchomić:

**Totally Integrated Automation Portal (TIA),**  
a następnie przejść do **Project view**.



Tworzenie nowego projektu można rozpocząć wybierając **Project > New**.

Należy nadać nazwę projektu np. „FAQ\_Modbus\_TCP”, żeby zakończyć tworzenie projektu należy kliknąć **Create**.



Powinno pojawić się okno projektu sterownika SIMATIC S7-1200.

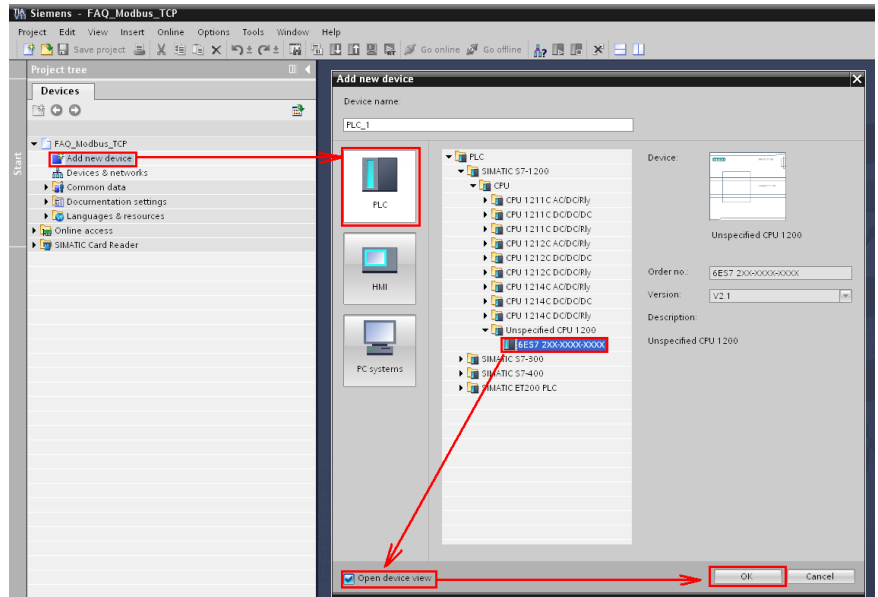
### 3.2 Dodanie nowego urządzenia

Z lewej strony okna w drzewie projektu „**Project tree**” należy kliknąć dwukrotnie na zakładkę **Add new device**, pojawi się okno o nazwie „**Add new device**”, należy wpisać nazwę nowego urządzenia np. „**Modbus\_Client**”.

Następnie trzeba rozwinąć zakładkę **Unspecified CPU 1200**, zaznaczyć **6ES7 2xx-xxxx-xxxx**.

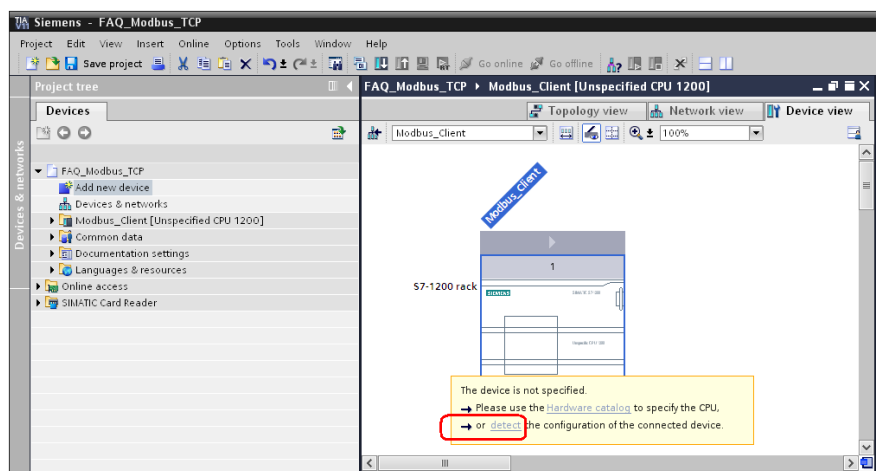
Ostatnim krokiem w tym oknie jest zaznaczenie **Open device view**, następnie należy kliknąć **OK**.

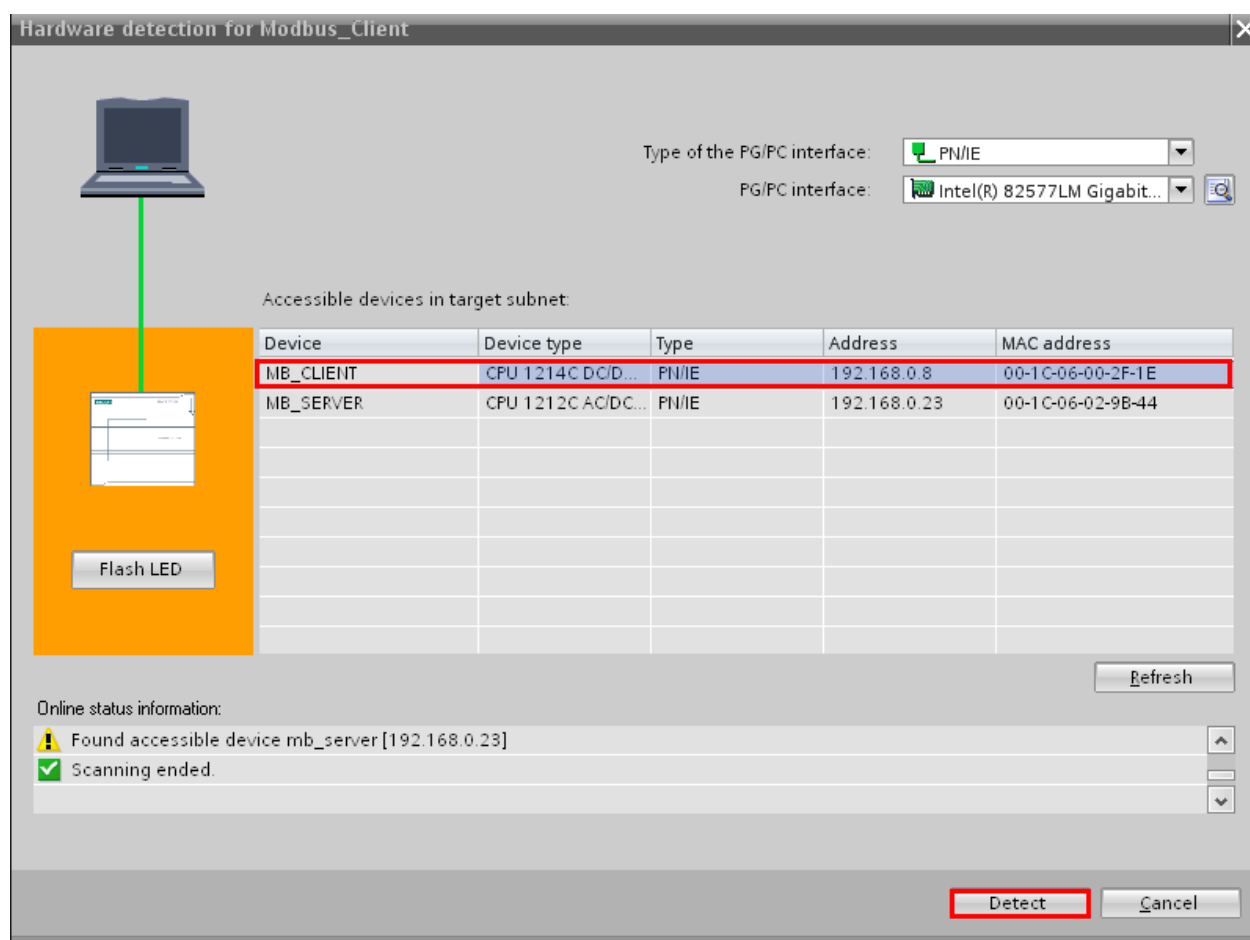
Po wprowadzonych zmianach automatycznie powinno otworzyć się okno projektu sterownika.



Najszybszym i najłatwiejszym sposobem konfiguracji sprzętowej jest bezpośrednie ściągnięcie jej z dostępnego fizycznie sterownika. Dokonuje się tego klikając opcję **detect**.

Otwiera się okno, w którym wyświetlone są dostępne, podłączone urządzenia. Wybieramy sterownik, z którym chcemy się połączyć i klikamy **detect**. Sprzęt zostanie wykryty automatycznie.





Dokładnie w ten sam sposób dodajemy drugi sterownik i nadajemy mu nazwę, np. „Modbus\_Server”.

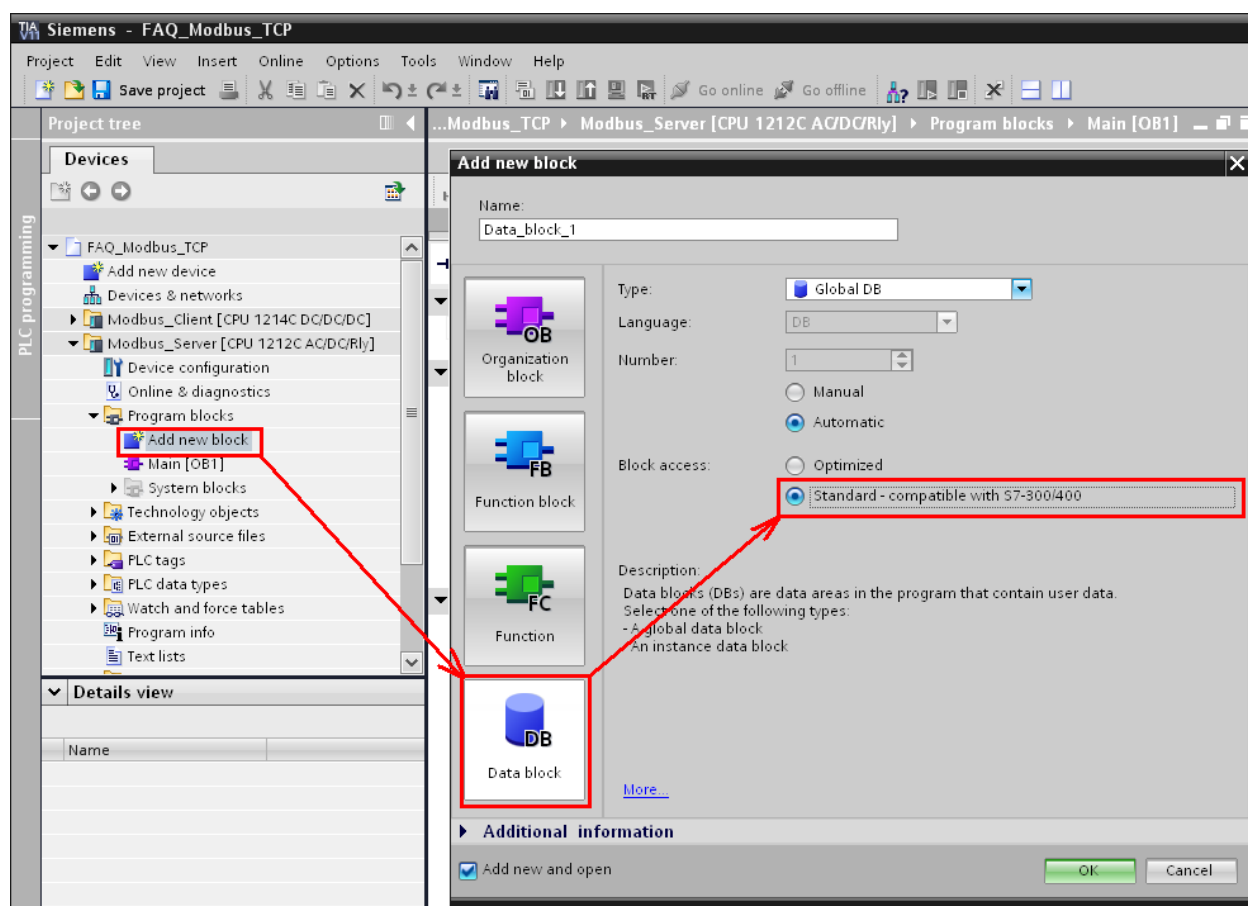
## 4 Modbus Serwer

### 4.1 Konfiguracja dwóch połączeń w jednym sterowniku „MB\_SERVER”

Możliwe jest wykonanie kilku połączeń Modbus TCP server. Aby tego dokonać należy wywołać niezależnie dla każdego połączenia instrukcję MB\_SERVER. Każde wywołanie musi posiadać przypisany indywidualny blok danych typu instancje DB, ID połączenia „CONNECT\_ID” oraz numer portu IP „IP\_PORT”.

Zostaną stworzone dwa połączenia, z których jedno będzie służyło do przesyłu obszaru pamięci typu M, a drugie do przesyłu danych z bloku danych DB. Instrukcje „MB\_SERVER” umieszczone zostaną w sterowniku, który został przez nas nazwany „Modbus\_Server”

W pierwszej kolejności stworzony zostanie blok danych, w którym będą przechowywane zmienne, które mają być przesyłane.



W bloku danych tworzymy 10 zmiennych typu Int (razem 20 bajtów).

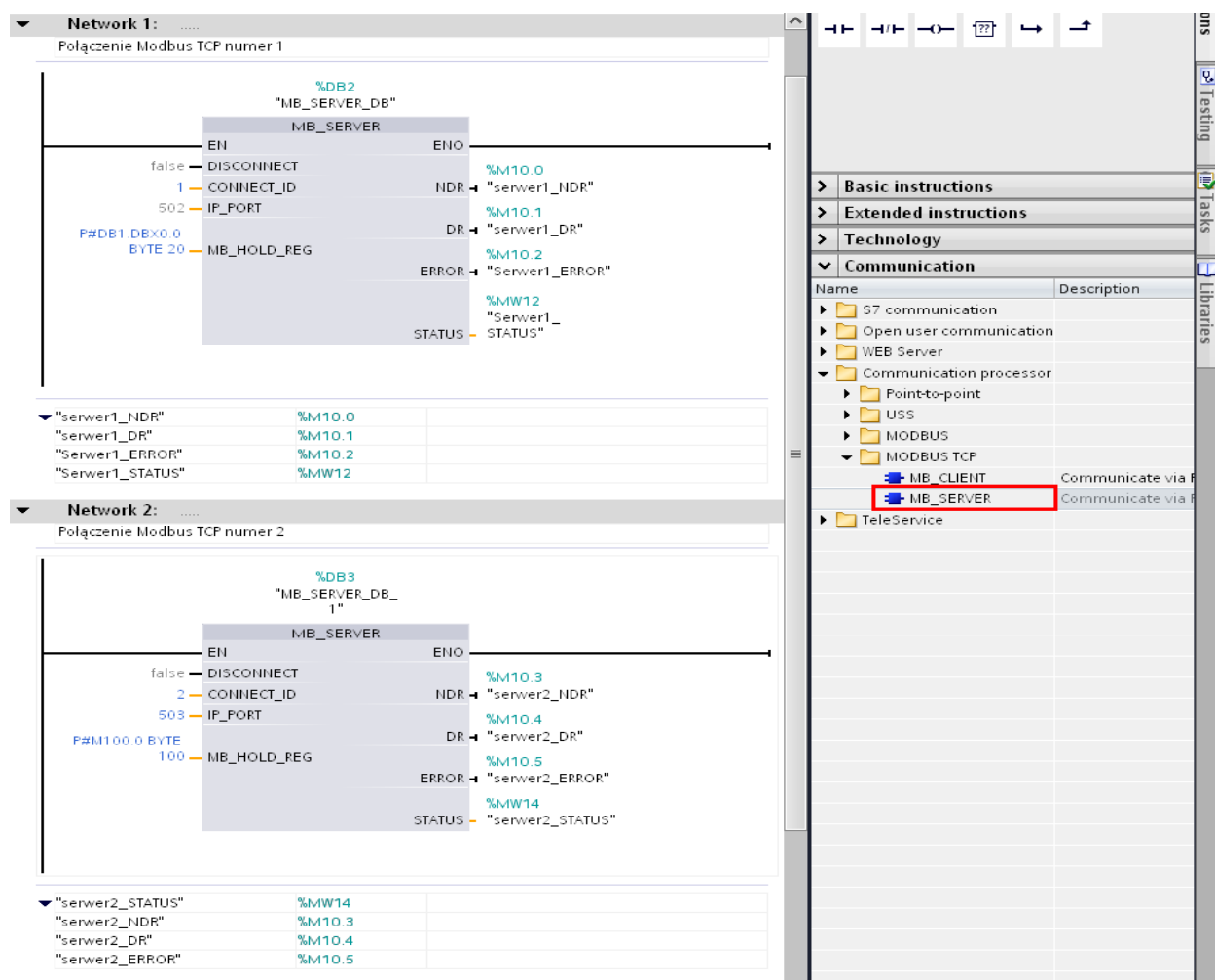
Data_block_1							
	Name	Data type	Offset	Start value	Retain	Visible in ...	Comm
1	Static						
2	MB_1	Int	0.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
3	MB_2	Int	2.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
4	MB_3	Int	4.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
5	MB_4	Int	6.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
6	MB_5	Int	8.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
7	MB_6	Int	10.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
8	MB_7	Int	12.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
9	MB_8	Int	14.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
10	MB_9	Int	16.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
11	MB_10	Int	18.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
12	<Add new>				<input type="checkbox"/>	<input type="checkbox"/>	

Następnie w bloku OB1 „MAIN” wywołujemy dwukrotnie instrukcję „MB\_SERVER” i określamy parametry połączenia:

- „DISCONNECT” ustawiamy na zero. Oznacza to, że może zostać zainicjowane połączenie pasywne
- „CONNECT\_ID” unikalny numer połączenia – „1” dla połączenia pierwszego, „2” dla drugiego
- „IP\_PORT” numer portu IP – każde połączenie musi być przypisane do odrębnego portu.

Więcej na temat numerów portów znajduje się helpie STEP 7

- „MB\_HOLD\_REG” wskaźnik obszaru pamięci zawiera zmienne, do których dostęp będzie miał klient Modbusa używający funkcji 3 (read), 6 (write) oraz 16 (read)



#### 4.2 Wyjaśnienie parametrów wejściowych „MB\_SERVER”

Parametr	Typ danych	Opis
DISCONNECT	Bool	„MB_SERVER” próbuje nawiązać pasywne połączenie z partnerem. Oznacza to, że serwer jest w trybie pasywnym i nasłuchuje wywołania. Jeżeli „DISCONNECT” = 0 i połączenie jeszcze nie jest nawiązane, wtedy nowe połączenie może zostać zainicjowane. Jeżeli „DISCONNECT” = 1 i połączenie jest nawiązane, wtedy połączenie jest przerywane. Pozwala to programowi kontrolować połączenie po jego nawiązaniu. Gdy parametr ma wartość „1”, nie można zainicjować połączenia.
CONNECT_ID	UInt	Parametr ten identyfikuje unikalne połączenia wewnątrz PLC. Każdy unikalny blok danych funkcji „MB_CLIENT” lub „MB_SERVER” musi posiadać unikalny numer ID połączenia.
IP_PORT	UInt	Wartość domyślna = 502: Numer portu IP identyfikuje port IP, który będzie monitorowany w celu wykrycia zapytania od klienta Modbusa. Poniższe numery TCP portu nie są dozwolone dla pasywnego połączenia „MB_SERVER”: 20, 21, 25, 80, 102, 123, 5001, 34962 oraz 34964.



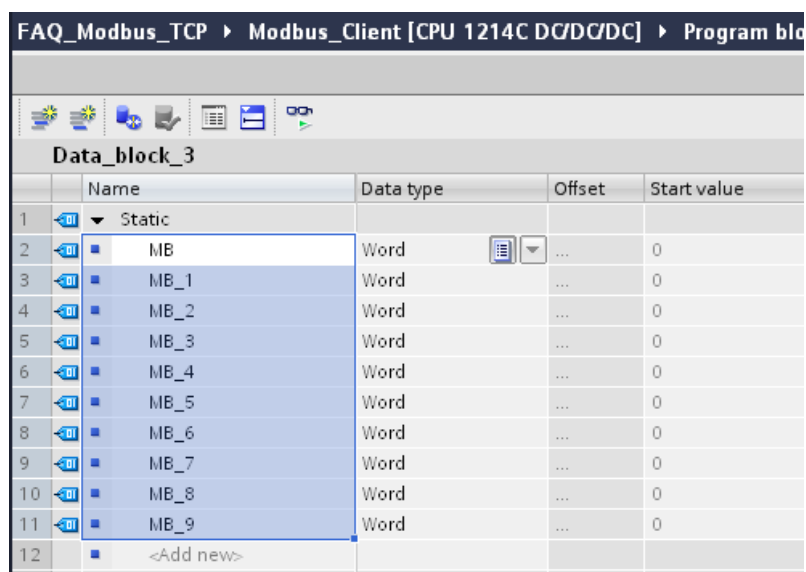
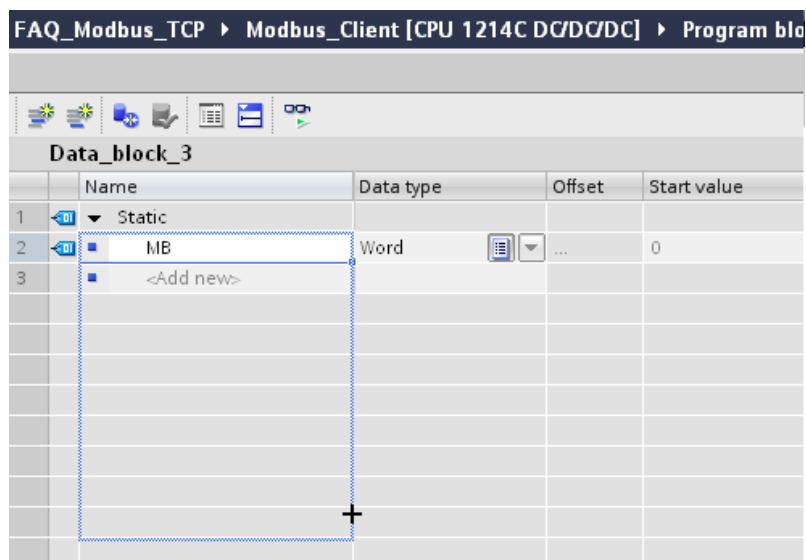
MB_HOLD_REG	Variant	Wskaźnik do rejestru pamięci serwera: Rejestr pamięci musi być blokiem danych o standardowym dostępie lub obszarem pamięci M. Ten obszar pamięci wykorzystywany jest do przechowywania danych, do których będzie miał dostęp klient Modbusa używając funkcji 3 (read), 6 (write) oraz 16 (write)
-------------	---------	--

## 5 Modbus Klient

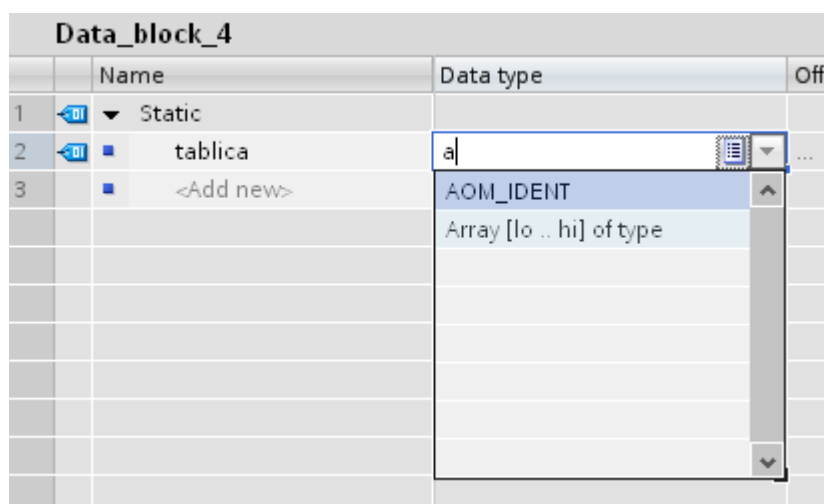
### 5.1 Konfiguracja dwóch połączeń w jednym sterowniku „MB\_CLIENT”

Zapytania klienta Modbusa mogą być rozsyłane za pośrednictwem kilku połączeń. Aby było to możliwe należy w programie umieścić instrukcję „MB\_CLIENT” (po jednej dla każdego połączenia), jednocześnie należy pamiętać, że każde jej wywołanie musi mieć oddzielny blok instancji DB. Adres IP musi być różny, jeżeli łączymy się z różnymi sterownikami lub taki sam, jeżeli realizujemy kilka połączeń w obrębie tego samego sterownika. Jeżeli nawiązywanych jest kilka połączeń z tym samym sterownikiem należy pamiętać, aby odwołać się do odpowiedniego portu IP. Poniższy przykład pokazuje, w jaki sposób zrealizować dwa połączenia z jednym sterownikiem (jeden adres IP, dwa numery portu). Instrukcje „MB\_Client” zostaną umieszczone w bloku OB1 w sterowniku, który nazwalismy „Modbus\_Client”.

Pierwszym krokiem jest określenie obszarów pamięci, do których mają zostać zapisane dane odczytane z serwera. W tym celu utworzymy dwa bloki danych. Pierwszy będzie składał się z dziesięciu zmiennych typu WORD (20 bajtów), a w drugim umieścimy tablicę składającą się ze 100 elementów o wielkości jednego bajtu. W tym celu klikamy dwukrotnie na „Add New block”, wybieramy Data block i ustawiamy standardowe adresowanie (Block Access >> Standard – Compatible with S7-300/400). W pierwszym bloku danych umieszczamy 10 zmiennych o nazwie „MB” typu WORD. W tym celu wpisujemy nazwę zmiennej „MB”, typ danych ustawiamy WORD i chwytając za prawy dolny róg pola z nazwą zmiennej przeciągamy w dół, dzięki czemu automatycznie utworzy nam się pożądana liczba kolejno ponumerowanych zmiennych.



W drugim bloku danych umieszczamy tablicę „array” składającą się ze 100 elementów typu bajt. W tym celu w polu „Name” wpisujemy nazwę tablicy, np. „tablica”. Następnie w polu „Data type” wpisujemy literę „a” i czekamy aż rozwinie się pole wyboru zmiennej”, po czym wybieramy typ „Array [lo .. hi] of type”. Wpisujemy zakres i określamy typ zmiennych.



Data_block_4				
	Name	Data type	Offset	Start value
1	Static			
2	tablica	Array [0 .. 99] of Byte		
3	tablica[0]	Byte		0
4	tablica[1]	Byte		0
5	tablica[2]	Byte		0
6	tablica[3]	Byte		0
7	tablica[4]	Byte		0
8	tablica[5]	Byte		0
9	tablica[6]	Byte		0
10	tablica[7]	Byte		0
11	tablica[8]	Byte		0
12	tablica[9]	Byte		0
13	tablica[10]	Byte		0
14	tablica[11]	Byte		0
15	tablica[12]	Byte		0
16	tablica[13]	Byte		0
17	tablica[14]	Byte		0
18	tablica[15]	Byte		0
19	tablica[16]	Byte		0
20	tablica[17]	Byte		0
21	tablica[18]	Byte		0
22	tablica[19]	Byte		0
23	tablica[20]	Byte		0
24	tablica[21]	Byte		0
25	tablica[22]	Byte		0

Z biblioteki instrukcji przeciągamy dwukrotnie do OB1 instrukcję „MB\_CLIENT” (podobnie jak MB\_SERVER) oraz przypisujemy odpowiednie parametry:

- „REQ” – żądanie połączenia z serwerem Modbusa (zbrocze narastające)
  - „DISCONNECT” – ustawiamy „0”, jeżeli chcemy, aby nawiązanie połączenia było możliwe.
- Po nawiązaniu połączenia można je zerwać za pomocą tego parametru ustawiając „1”
- „CONNECT\_ID” – ustawiamy odpowiednio „1” i „2”
  - „IP\_OCTET\_1.4” – wpisujemy adres IP sterownika, z którym będziemy się łączyć
  - „IP\_PORT” – numer portu nawiązuje do numeracji portów w serwerze
  - „MB\_MODE” – ustawiamy „0”, dzięki czemu możliwy będzie odczyt z serwera

- „MB\_DATA\_ADDR” – wpisujemy 40001, co oznacza, że zaczniemy odczyt od początku zadeklarowanego w serwerze rejestru
  - „MB\_DATA\_LEN” – parametr określa ilość danych jaka ma być odczytana z serwera. Wpisujemy odpowiednio 10 (10 słów) oraz 100 (100 słów)
  - „MB\_DATA\_PTR” – jest to wskaźnik do lokalnego obszaru pamięci. Określa obszar danych, do którego mają zostać zapisane dane odczytane z serwera.
- Opcjonalnie możemy dodać instrukcję „MOVE”, dzięki czemu możliwe będzie zatrzaśnięcie statusu w momencie pojawienia się błędu.

The image displays two screenshots of the Siemens SIMATIC Manager software, specifically the LAD editor for a Modbus client program. The project is named 'FAQ\_Modbus\_TCP' and the program is 'Main [OB1]'.

**Network 1:** This network shows the configuration for a Modbus client. The block 'MB\_CLIENT' is used with the following parameters:
 

- EN: %M10.0 (Client1\_REQ)
- DISCONNECT: FALSE
- CONNECT\_ID: 1
- IP\_OCTET\_1: 192
- IP\_OCTET\_2: 168
- IP\_OCTET\_3: 0
- IP\_OCTET\_4: 23
- IP\_PORT: 502
- MB\_MODE: 0
- MB\_DATA\_ADDR: 40001
- MB\_DATA\_LEN: 10
- MB\_DATA\_PTR: P#DB3 DBX0.0 BYTE 20

 The output signals are:
 

- DOHE: %M11.0 (client1\_DONE)
- BUSY: %M11.1 (client1\_BUSY)
- ERROR: %M11.2 (client1\_ERROR)
- STATUS: %MW12 (client1\_STATUS)

 A 'MOVE' instruction is used to set the output ZATRZ (client1\_STAT\_ZATRZ) to the status %MW12 when an error occurs (%M11.2).

**Network 2:** This network shows the configuration for a second Modbus client. The block 'MB\_CLIENT' is used with the following parameters:
 

- EN: %M10.1 (client2\_REQ)
- DISCONNECT: FALSE
- CONNECT\_ID: 2
- IP\_OCTET\_1: 192
- IP\_OCTET\_2: 168
- IP\_OCTET\_3: 0
- IP\_OCTET\_4: 23
- IP\_PORT: 503
- MB\_MODE: 0
- MB\_DATA\_ADDR: 40001
- MB\_DATA\_LEN: 50
- MB\_DATA\_PTR: P#DB4 DBX0.0 BYTE 100

 The output signals are:
 

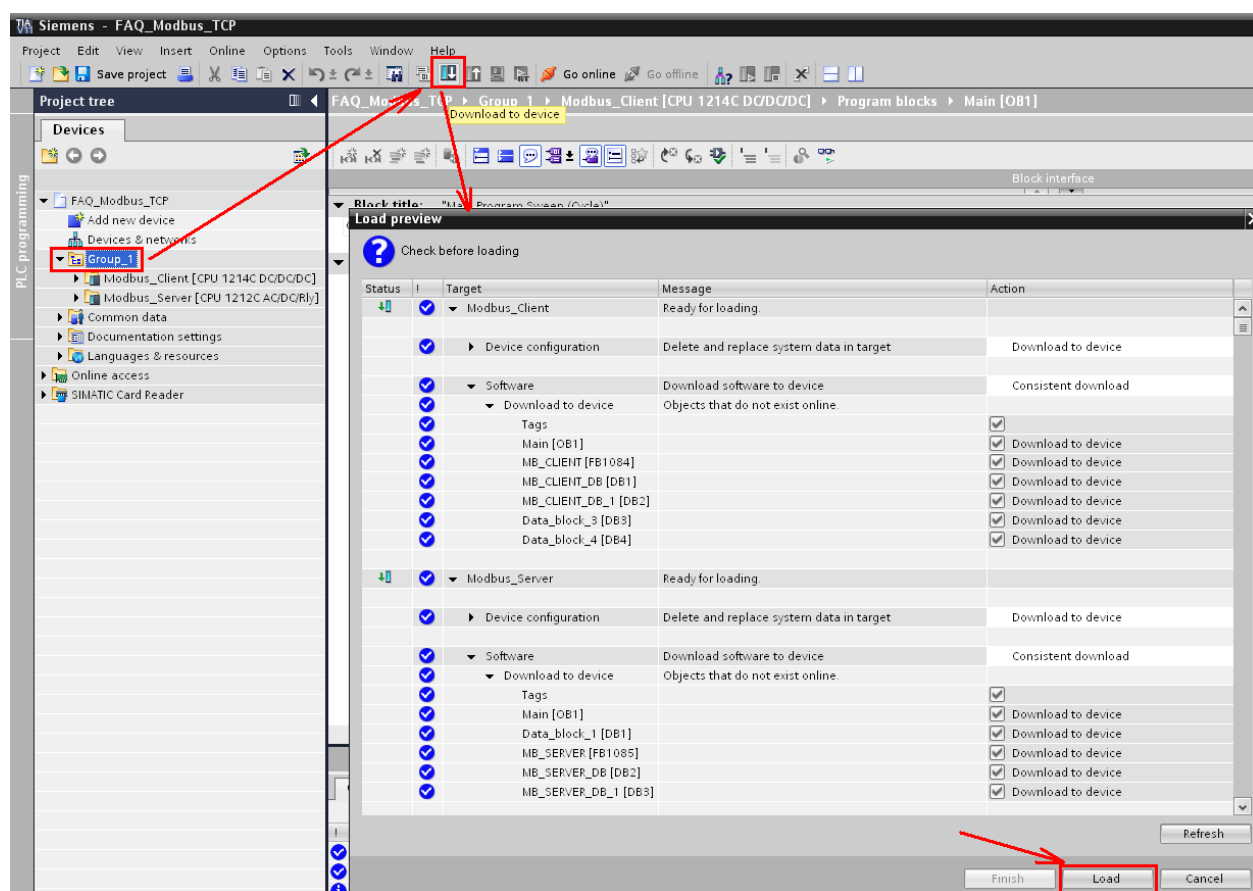
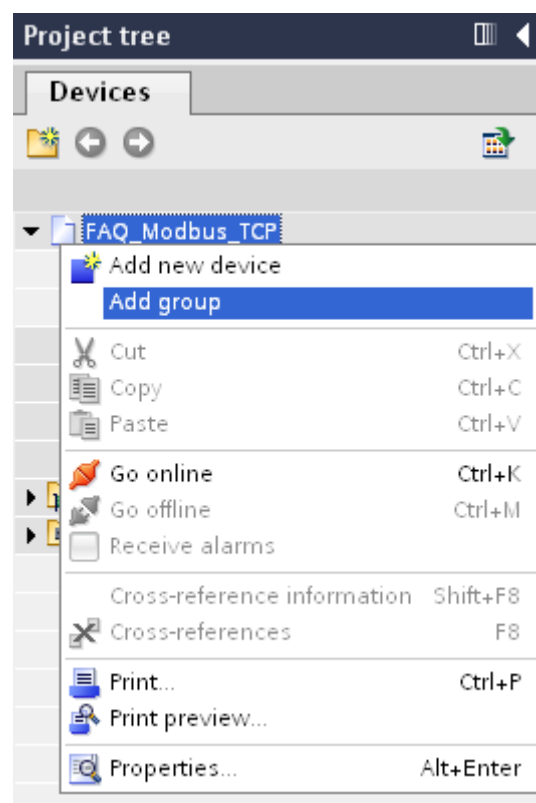
- DOHE: %M11.3 (client2\_DONE)
- BUSY: %M11.4 (client2\_BUSY)
- ERROR: %M11.5 (client2\_ERROR)
- STATUS: %MW14 (client2\_STATUS)

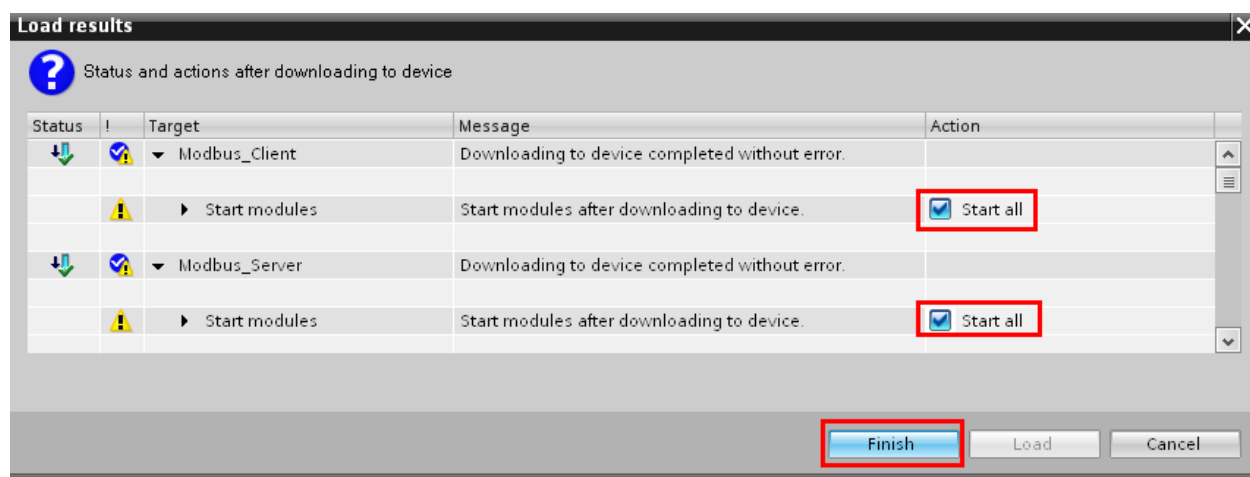
 A 'MOVE' instruction is used to set the output ZATRZ (client2\_STAT\_ZATRZ) to the status %MW14 when an error occurs (%M11.5).

Po wykonaniu wszystkich powyższych kroków ładujemy programy do sterowników. Jeżeli chcemy załadować program do wszystkich sterowników jednocześnie możemy je umieścić w jednej grupie.

W tym celu klikamy prawym przyciskiem myszy na nazwę projektu w drzewie projektu i wybieramy „Add group”. Foldery poszczególnych sterowników przeciągamy do nowo utworzonej grupy, zaznaczamy ją lewym przyciskiem myszy i klikamy „Download to device”

Jeżeli w programie nie było żadnych błędów, zostanie on załadowany. Następnie zaznaczamy opcję „Start all”, co spowoduje przejście sterowników w tryb „RUN”





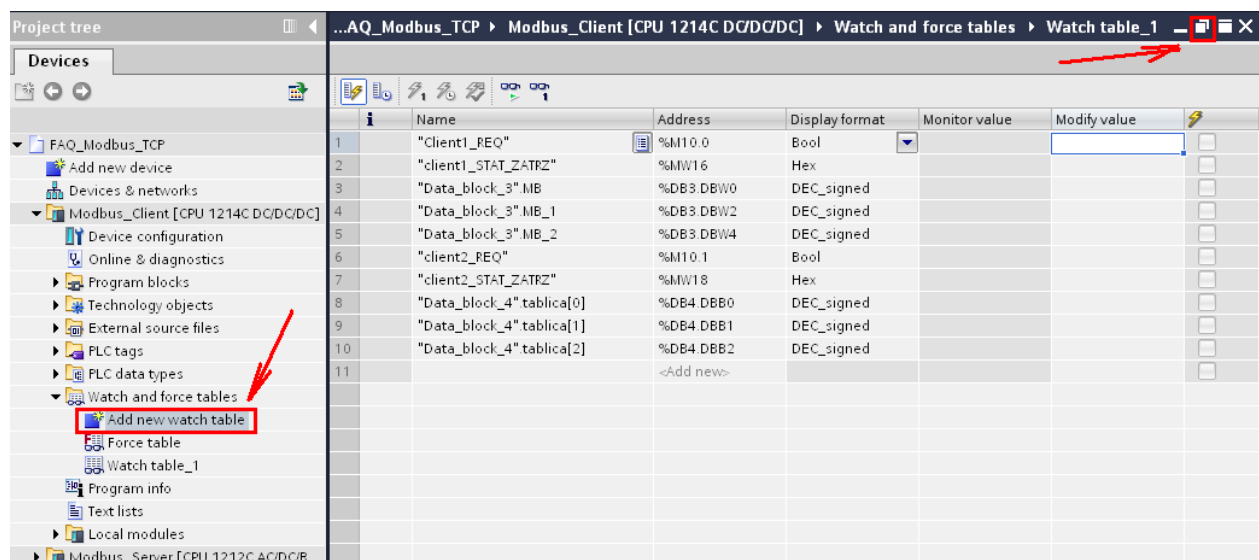
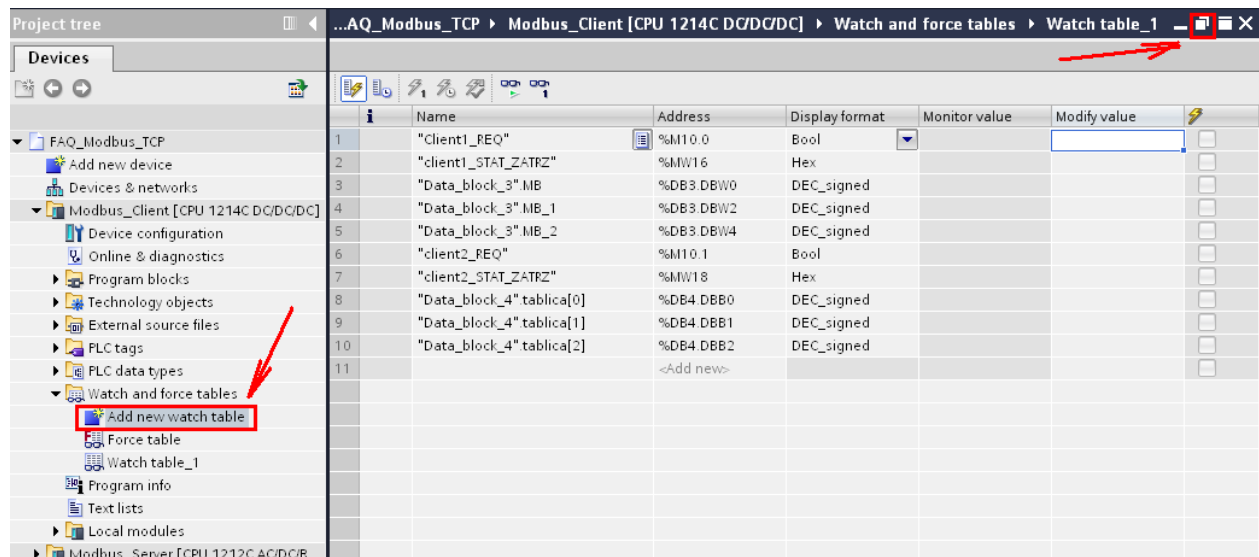
Oczywiście można również załadować program do każdego sterownika osobno bez tworzenia grupy urządzeń.

## 5.2 Wyjaśnienie parametrów wejściowych „MB\_CLIENT”

Parametr	Typ danych	Opis
REQ	Bool	FALSE = brak żądania komunikacji Modbus TRUE = żądanie komunikacji z serwerem Modbus
DISCONNECT	Bool	Parametr ten pozwala na kontrolę połączenia z serwerem Modbus przez program. Jeżeli „DISCONNECT” = 0 i połączenie nie zostało jeszcze nawiązane, wtedy „MB_CLIENT” próbuje nawiązać połączenie z przypisanym adresem IP i numerem portu Jeżeli „DISCONNECT” = 1 i połączenie istnieje, wtedy następuje przerwanie połączenia. Jeżeli ten parametr ma wartość 1, wtedy nie zostanie nawiązane żadne połączenie
CONNECT_ID	UInt	Parametr ten identyfikuje unikalne połączenia wewnątrz PLC. Każdy unikalny blok danych funkcji „MB_CLIENT” lub „MB_SERVER” musi posiadać unikalny numer ID połączenia
IP_OCTET_1	USInt	Adres IP serwera Modbus TCP: Oktet 1 8 bitowa część 32 bitowego adresu IP IPv4 serwera Modbus TCP z którym łączy się klient za pośrednictwem protokołu Modbus TCP
IP_OCTET_2	USInt	Adres IP serwera Modbus TCP: Oktet2
IP_OCTET_3	USInt	Adres IP serwera Modbus TCP: Oktet3
IP_OCTET_4	USInt	Adres IP serwera Modbus TCP: Oktet4
IP_PORT	UInt	Wartość domyślna = 502: Numer portu IP identyfikuje port IP serwera, do którego klient będzie się odwoływał przystępując do nawiązania połączenia używając protokołu TCP/IP.

## 6 Watch tables

Aby móc przetestować aplikację stworzymy tablice do obserwacji i modyfikacji zmiennych „Watch table”. Klikamy „Add new watch table” i wpisujemy w niej zmienne, do których chcemy mieć dostęp. Tworzymy dwie takie tablice. Jedną w „Modbus\_Client”, a drugą w „Modbus\_Server”. Aby móc obserwować dwie tablice jednocześnie, klikamy ikonkę „float”.



Klikamy podgląd (okularki), w kolumnie „Modify value” wpisujemy wartości, które chcemy przesłać i klikamy ikonkę „modify”.

The screenshot shows the SIMATIC Manager interface with the project tree on the left and the Watch table on the right. The Watch table is titled 'Watch table\_1' and contains the following data:

Name	Address	Display format	Monitor value	Modify value
"Client1_REQ"	%M10.0	Bool	FALSE	
"client1_STAT_ZATRZ"	%MW16	Hex	16#0000	
"Data_block_3" MB	%DB3.DBW0	DEC_signed	0	
"Data_block_3" MB_1	%DB3.DBW2	DEC_signed	0	
"Data_block_3" MB_4	%DB3.DBW8	Hex	16#0000	
"client2_REQ"	%M10.1	Bool	FALSE	
"client2_STAT_ZATRZ"	%MW18	Hex	16#0000	
"Data_block_4" tablica[0]	%DB4.DBB0	DEC_signed	0	
"Data_block_4" tablica[1]	%DB4.DBB1	DEC_signed	0	
"Data_block_4" tablica[99]	%DB4.DBB99	DEC_signed	0	

W „Watch table” klienta uruchamiamy odczyt poprzez zmianę parametru REQ. Parametr REQ może być zmieniony np. poprzez kliknięcie prawym przyciskiem myszy na wiersz, w którym znajduje się parametr i wybraniu opcji „modify to 1”. Następnie ustawiamy parametr powtórnie na „0”. Operację wykonujemy najpierw dla „Client1\_REQ”, a następnie dla „Client2\_REQ”.

The screenshot shows the context menu for the 'Client1\_REQ' variable in the Watch table. The menu options are:

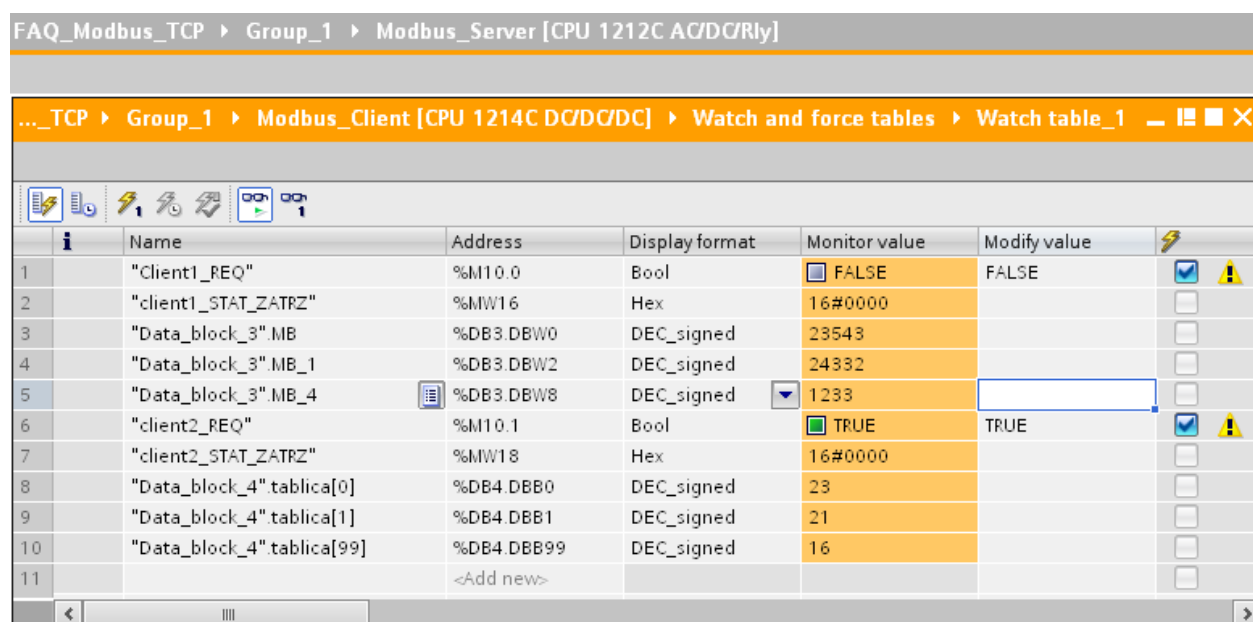
- Modify (selected)
- Monitor all
- Monitor now
- Cut (Ctrl+X)
- Copy (Ctrl+C)
- Paste (Ctrl+V)
- Delete (Del)
- Rename (F2)
- Expanded Mode

The 'Modify' sub-menu shows the following options:

- Modify to 0 (Ctrl+Shift+0)
- Modify to 1 (Ctrl+Shift+1)
- Modify now
- Modify with trigger
- Enable peripheral outputs

Zmienne zostały odczytane, a parametry: „client1\_STAT\_ZATRZ” oraz „client2\_STAT\_ZATRZ” mają wartość zero (brak błędu podczas przesylu).





## 7 Zakres adresów

Zamiast numeru funkcji instrukcja „MB\_CLIENT” używa parametru „MB\_MODE”. Parametr „MB\_DATA\_ADDR” używany jest do określania adresu startowego obszaru danych, jaki chcemy przesłać. Kombinacja tych dwóch parametrów definiuje kod funkcji Modbus. Poniższa tabela zawiera zależność między parametrem „MB\_MODE”, kodem funkcji Modbus oraz zakresem adresów.

Parametr „MB_MODE”	Kod funkcji Modbus	Parametr „MB_DATA_LEN”	Funkcjonalność i typ danych	Parametr „MB_DATA_ADDR”
0	01	1 do 2000	Czytanie bitów wyjściowych: 1 do 2000 bitów podczas jednego wywołania	1 do 9999
0	02	1 do 2000	Czytanie bitów wejściowych: 1 do 2000 bitów podczas jednego wywołania	10001 do 19999
0	03	1 do 125	Odczyt rejestrów pamiętających: 1 do 125 zmiennych typu WORD podczas jednego wywołania	40001 do 49999
0	04	1 do 125	Odczyt słów wejściowych: 1 do 125 słów podczas jednego wywołania	30001 do 39999
1	05	1	Zapis pojedynczego bitu wyjściowego: Jeden bit podczas jednego wywołania	1 do 9999
1	06	1	Zapis rejestru pamiętającego: Jedno słowo podczas jednego wywołania	40001 do 49999
1	15	2 do 1968	Zapis bitów wyjściowych: 2 do 1968 bitów podczas jednego wywołania	1 do 9999
1	16	2 do 123	Zapis rejestrów pamiętających: 2 do 123 słów podczas jednego wywołania	40001 do 49999



## 8 Kody warunkowe

### 8.1 Wartość „STATUS” dla „MB\_SERVER”

- 8187 – Nieprawidłowy wskaźnik dla parametru „MB\_HOLD\_REG”: obszar pamięci jest zbyt mały
- 818C - Parametr „DATA\_PTR” wskazuje na zoptymalizowany blok danych (musi być standardowy blok DB lub obszar pamięci M)
- 8381 – Nieobsługiwany kod funkcji
- 8382 – Błąd długości danych
- 8383 – Błąd adresu danych lub próba dostępu poza granice określone przez „MB”HOLD\_REG”
- 8384 – Błąd wartości danych
- 8385 – Nieobsługiwana wartość kodu diagnostycznego (kod funkcji 08).

### 8.2 Wartość „STATUS” dla „MB\_CLIENT”

- 80C8 – Serwer nie odpowiada w określonym czasie
- 8188 – Nieprawidłowa wartość parametru “MODE”
- 8189 – Nieprawidłowa wartość parametru “DATA\_ADDR”
- 818A – Nieprawidłowa wartość parametru „DATA\_LEN”
- 818B – Nieprawidłowy wskaźnik obszaru pamięci „DATA\_PTR”. Błąd może być spowodowany niepoprawną kombinacją parametrów „MB\_DATA\_ADDR” oraz „MB\_DATA\_LEN”
- 818C – Parametr „DATA\_PTR” wskazuje na zoptymalizowany blok danych (musi być standardowy blok DB lub obszar pamięci M)
- 8200 – Port jest zajęty przetwarzaniem innego żądania połączenia
- 8380 – Otrzymana ramka Modbusa jest zniekształcona lub otrzymano zbyt dużą ilość bajtów
- 8387 – Przypisany numer ID połączenia jest różny od ID z poprzedniego zapytania. Może być tylko jeden numer ID połączenia użyty wewnątrz przypisanego do „MB\_CLIENT” bloku danych DB
- 8388 – Serwer Modbus zwrócił ilość danych inną niż był odpytany. Tyczy się tylko funkcji 15 i 16.