



Politechnika Wroclawska

Wydział Informatyki i Zarządzania

**Opracowanie ćwiczenia laboratoryjnego
dotyczącego wykorzystania
sieci przemysłowej Profibus**

**DODATEK NR 2
Opis konfiguracji sieci Profibus**

Opracował: Paweł Obraniak

Wrocław 2014

Spis treści

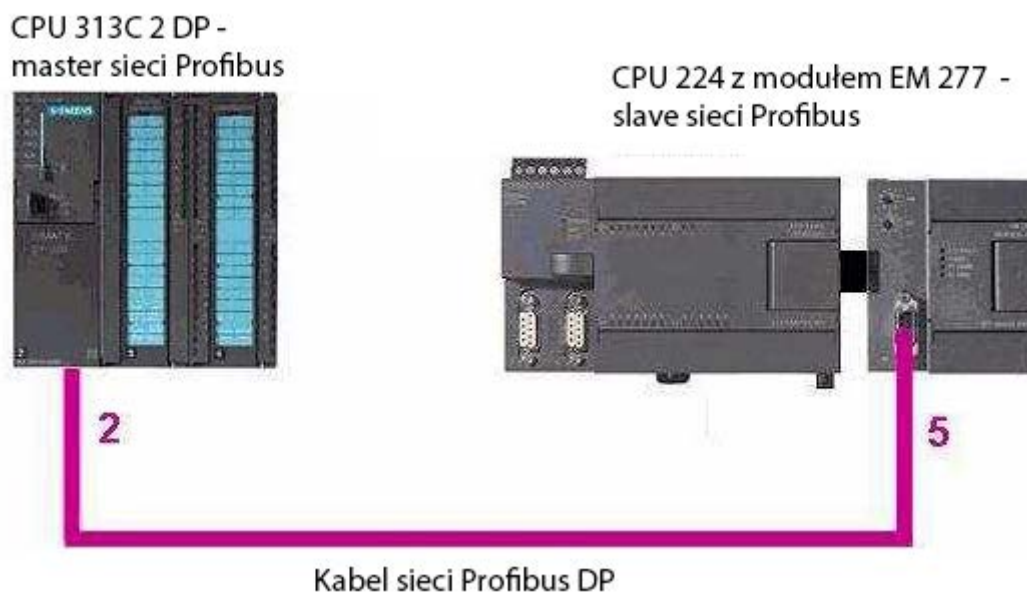
Przygotowanie komunikacji pomiędzy sterownikami Siemens s7-300 oraz Siemens s7-200 za pomocą protokołu Profibus DP	2
Konfiguracja sterownika S7-300 jako mastera sieci Profibus	3
Przygotowanie projektu w programie Simatic Manager	3
Konfiguracja sprzętowa sterownika	5
Instalacja plików GSD dla urządzeń nie znajdujących się w katalogu	16
Dodanie stacji slave EM277 do projektu	21
Zapisanie i Kompilacja konfiguracji sprzętowej	29
Ustawienie adresu sprzętowego na module EM277	30
Przykładowy program do testu komunikacji Profibus dla S7-300	31
Konfiguracja połączenia komputer – sterownik S7-300	31
Podstawowe informacje o edytorze programowym Step7	36
Przykładowy program do testu komunikacji dla S7-300	40
Przesłanie programu do sterownika	44
Przykładowy program do testu komunikacji Profibus dla S7-200	46
Konfiguracja połączenia komputer – sterownik S7-200	46
Wybór typu sterownika i przypisanie go do projektu	48
Przykładowy program do testu komunikacji	52

Przygotowanie komunikacji pomiędzy sterownikami Siemens s7-300 oraz Siemens s7-200 za pomocą protokołu Profibus DP

Sieć PROFIBUS DP pozwala na komunikację Master-Slave, w której S7-200 wraz z modulem EM277 pracują w trybie Slave, natomiast S7-300 pracuje w trybie Master.

Typowa sieć PROFIBUS to jedno urządzenie Master i kilka urządzeń Slave. Urządzenie Master zawiera dane konfiguracyjne współpracujących urządzeń Slave oraz ich adresy.

Master inicjalizuje komunikację oraz weryfikuje, czy stacje Slave są zgodne z danymi konfiguracyjnymi. Master w sposób ciągły zapisuje dane wyjściowe do stacji Slave oraz odczytuje z nich dane wejściowe.



Rys. 1. Połączenie Profibus w ćwiczeniu.

W tym ćwiczeniu skonfigurujemy połączenie między sterownikiem Siemens S7-300 (CPU 313C 2 DP) jako Masterem sieci oraz sterownikiem Siemens S7-200 CPU 224 z dołączonym modulem EM277 jako Slave sieci Profibus.

Konfiguracja sterownika S7-300 jako mastera sieci Profibus

Przygotowanie projektu w programie Simatic Manager

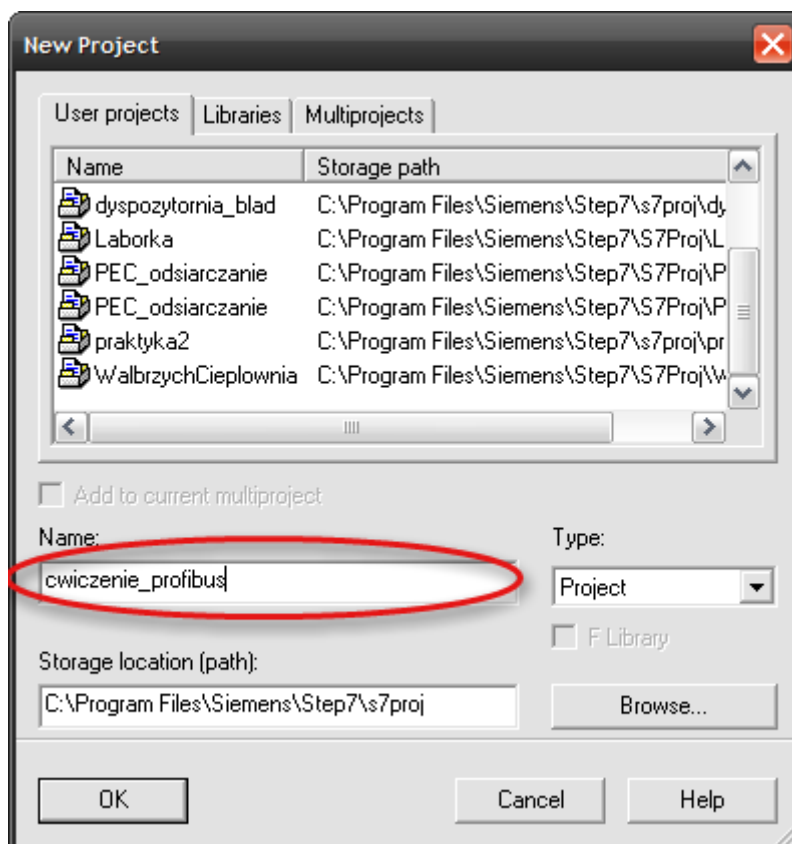
Pierwszym krokiem będzie utworzenie nowego projektu w środowisku Step7. W tym celu otwieramy program **SIMATIC Manager** dostępny w menu Start w katalogu SIMATIC.

Po uruchomieniu programu wybieramy menu **File -> New**



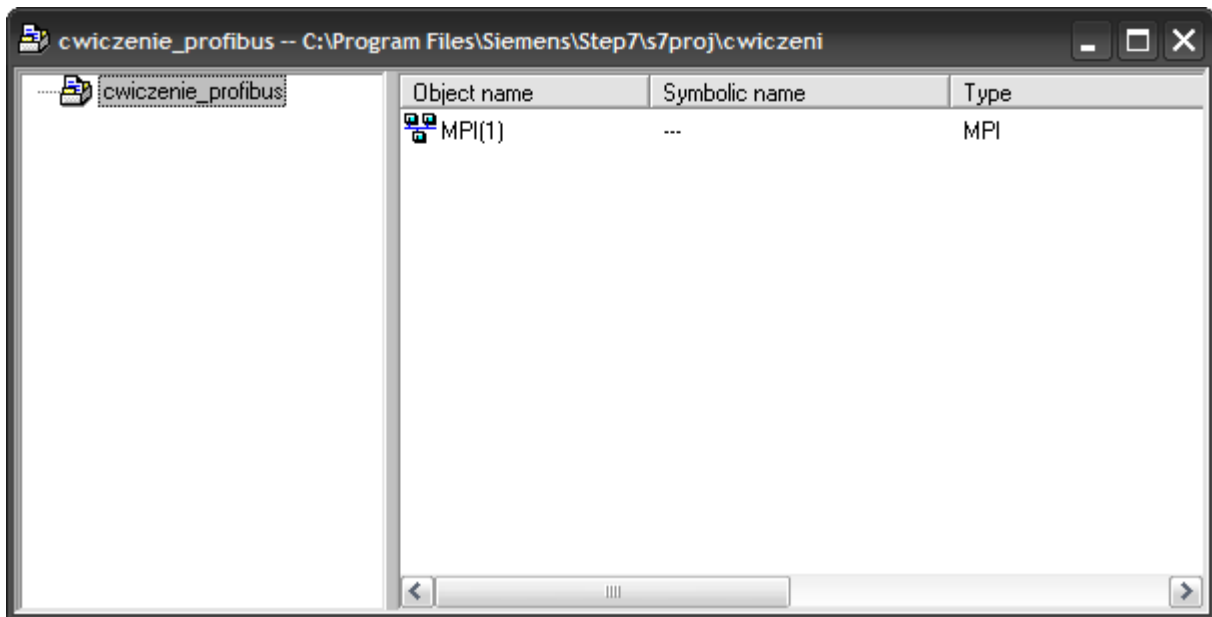
Rys. 2. Utworzenie nowego projektu w Simatic Managerze.

W nowym oknie wpisujemy nazwę dla naszego projektu np.: **cwiczenie_profibus**, resztę ustawień zostawiamy domyślnie jak na poniższym rysunku i klikamy **OK**.



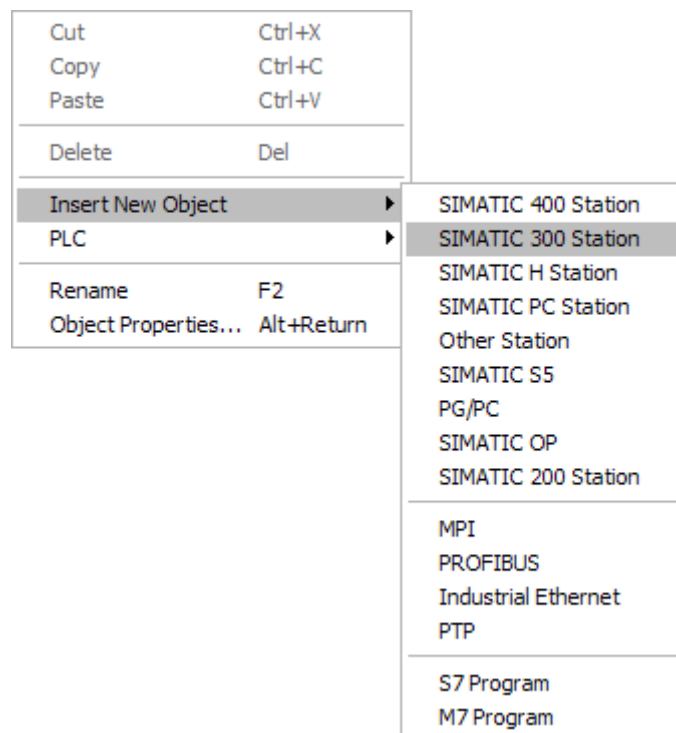
Rys. 3. Określenie nazwy i lokalizacji nowego projektu.

Otworzy się okno z zawartością naszego projektu. Na razie projekt nie zawiera sterownika i wygląda jak poniżej.



Rys. 4. Widok nowo utworzonego projektu.

Musimy teraz dodać do projektu sterownik który będzie masterem sieci. W naszym przypadku jest to Siemens serii S7-300, więc klikamy prawym przyciskiem na nazwie projektu w lewej części okna (u nas cwiczenie_profibus) i wybieramy kolejno **Insert New Object** a następnie **SIMATIC 300 Station**.



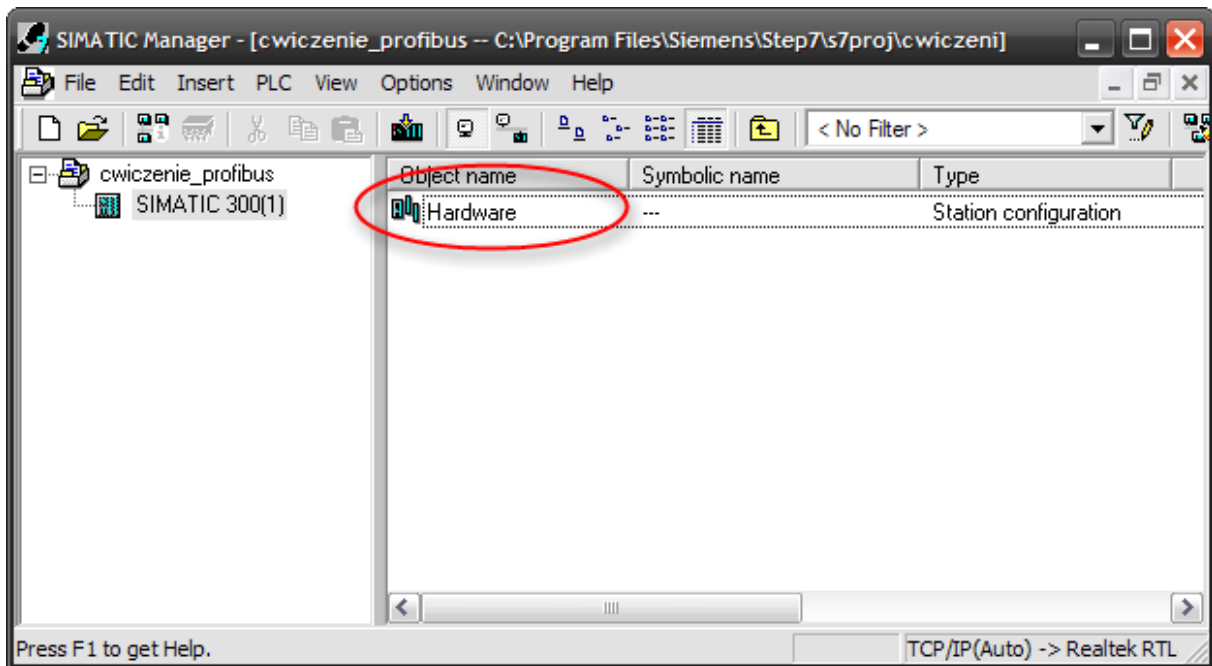
Rys. 5. Dodanie stacji serii 300 do projektu.

W projekcie pojawił się sterownik o domyślnej nazwie **SIMATIC 300(1)**. Teraz należy dokonać jego konfiguracji sprzętowej za pomocą edytora HW Config dostępnego z poziomu Simatic Managera.

Konfiguracja sprzętowa sterownika

Rozwijamy drzewo naszego projektu i zaznaczamy dodany przed chwilą sterownik SIMATIC 300(1).

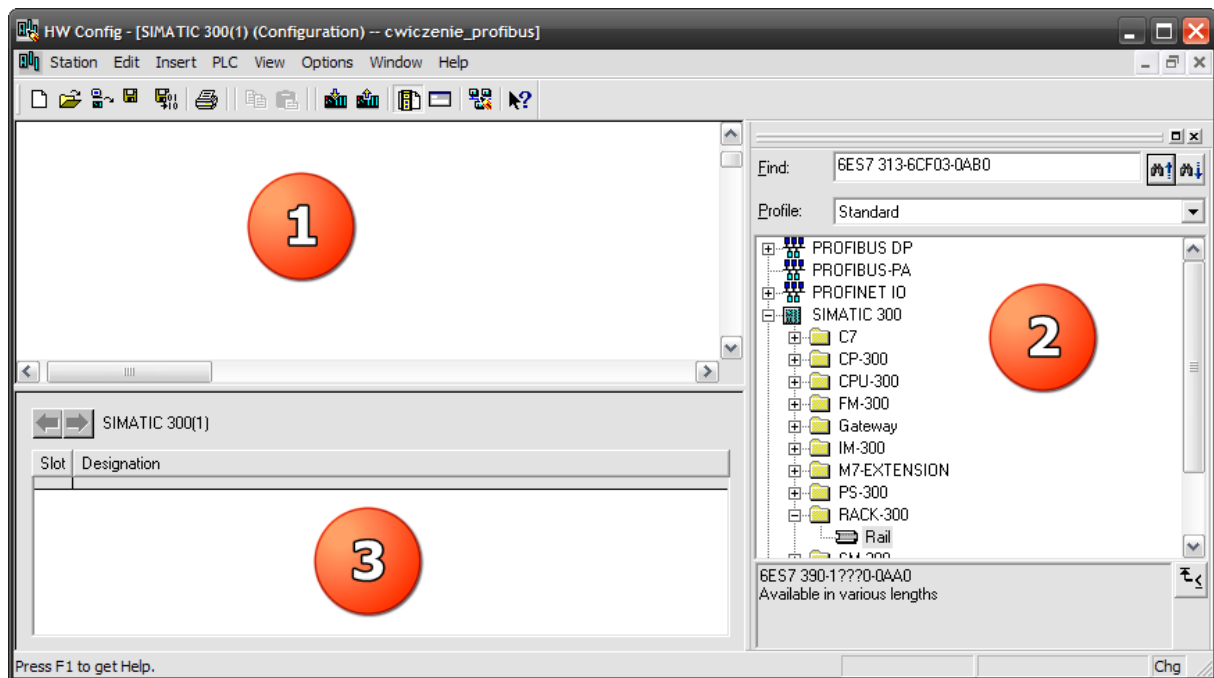
Klikamy dwukrotnie na ikonie Hardware aby przejść do edytora HW Config.



Rys. 6. Przejście do edytora konfiguracji sprzętowej.

Edytor HW Config umożliwia konfigurację sprzętową sterownika tj. wybór typu jednostki centralnej oraz modułów sygnałowych, komunikacyjnych itd. w które wyposażony jest nasz sterownik. Narzędzie to służy również do konfiguracji sieci komunikacyjnych w których sterownik będzie pracował.

Okno edytora podzielone jest na trzy części (rysunek poniżej).



Rys. 7. Okno edytora HW Config.

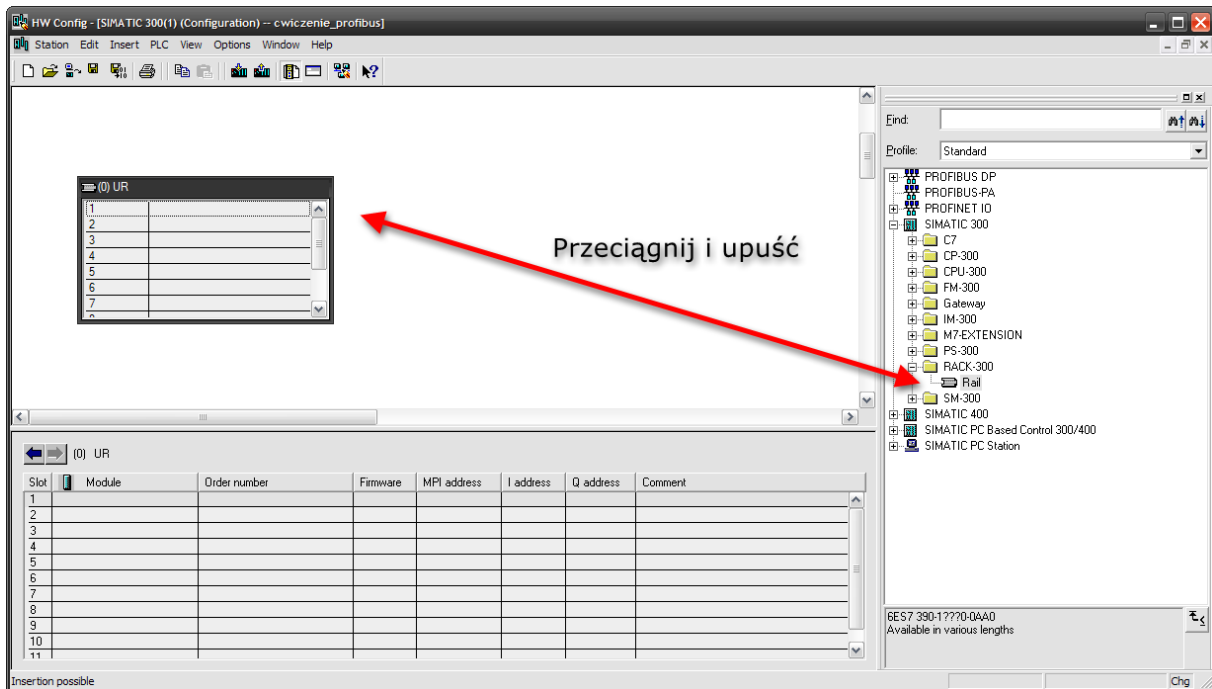
W pierwszej części (1) widzimy podgląd całego projektu ze sterownikiem i zainstalowanymi modułami (sygnałowymi, komunikacyjnymi itd.) umieszczonymi na szynach (rack) oraz sieci komunikacyjne w których sterownik pracuje wraz z urządzeniami podłączonymi do każdej z nich.

W drugiej części (2) znajduje się katalog modułów i elementów składających się na system SIMATIC. Są tutaj pogrupowane w kategoriach moduły sygnałowe, komunikacyjne, jednostki centralne i inne elementy systemu. Z tego katalogu wybieramy elementy wchodzące w skład naszego projektu.

W trzeciej części okna (3) znajduje się widok szczegółowy dla aktualnie wybranego w oknie (1) elementu. Tutaj widzimy np. jakie adresy zostały przypisane poszczególnym modułom sygnałowemu wchodzącym w skład zaznaczonej szyny UR czy właściwości zaznaczonej stacji Profibus.

Jak widać nasz projekt jest jeszcze pusty, więc dodamy kolejno wszystkie używane moduły.

W pierwszej kolejności musimy dodać szynę montażową (UR) na której zainstalowany będzie sterownik oraz moduły. W tym celu wybieramy z katalogu pozycję SIMATIC 300, następnie z listy dostępnych kategorii wybieramy RACK-300. Szyna jest oznaczona w katalogu jako Rail.



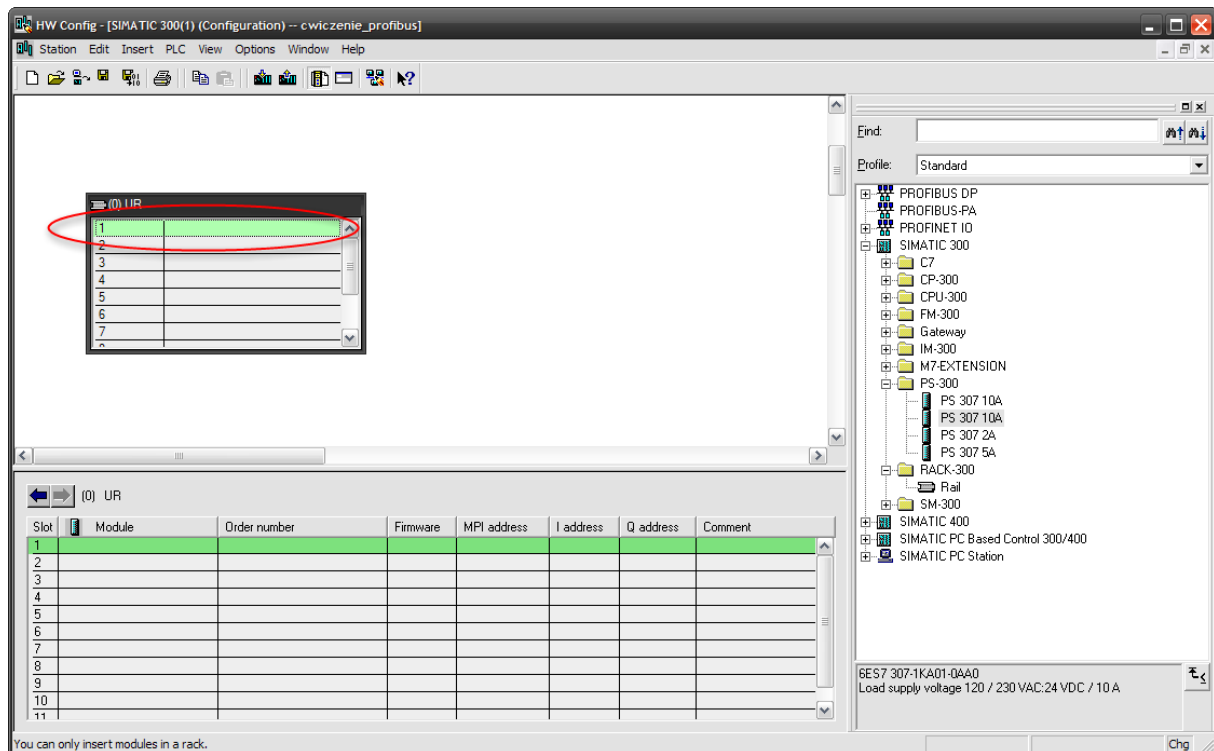
Rys. 8. Dodanie szyny (rail) do konfiguracji sprzętowej.

Przeciągamy pozycję Rail do głównego okna po lewej stronie. Teraz nasz projekt w HW Config powinien wyglądać jak na rysunku powyżej.

Szyna wyświetlana jest w formie tabeli. W systemie SIMATIC wiersze nr 1,2 oraz 3 mają ściśle określone zastosowanie.

Wiersz (slot) nr 1 służy zawsze do umieszczenia zasilacza, slot nr 2 jest przeznaczony na jednostkę centralną (CPU) a slot nr 3 na moduł komunikacyjny IM. Pozostałe sloty możemy dowolnie obsadzać modułami sygnałowymi, komunikacyjnymi itd.

Edytor HW Config podpowiada w którym miejscu na szynie możemy umieścić zaznaczony w katalogu moduł podświetlając możliwe lokalizacje na zielono.



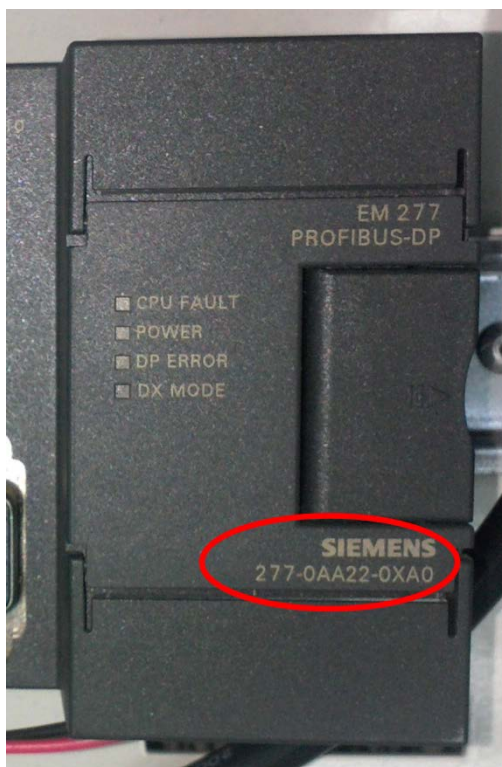
Rys. 9. Umieszczanie modułów na szynie w edytorze HW Config.

Pierwszy dodamy do projektu zasilacz umieszczony w katalogu w sekcji **SIMATIC 300** -> **PS-300**. Dodanie zasilacza nie jest obowiązkowe, ponieważ nie wymienia on żadnych informacji ze sterownikiem, jednak jest to zalecane w celach informacyjnych.

Wybieramy zasilacz PS 307 10A i przeciągamy go na slot 1 podświetlony na zielono.

Po dodaniu zasilacza dodajemy jednostkę centralną. Prosty sposób na wyszukiwanie CPU oraz innych modułów w katalogu jest posługiwanie się numerem zamówieniowym Siemens'a widniejącym na każdym z modułów.

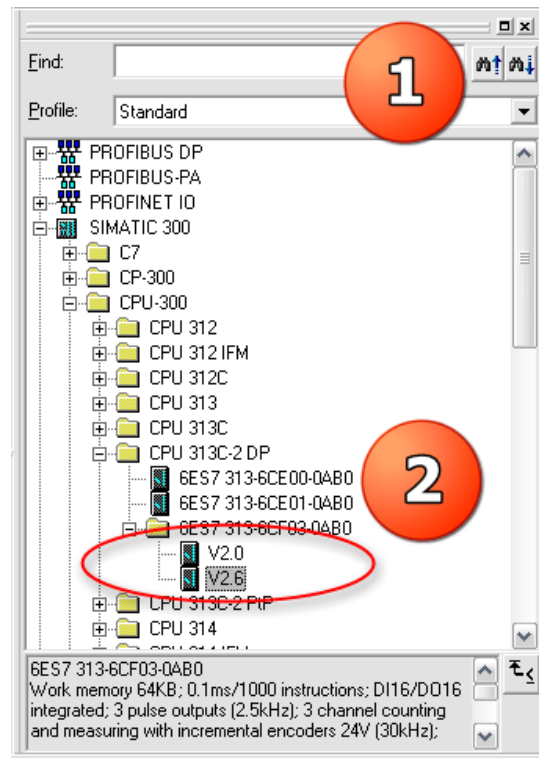
Na rysunku poniżej widać przykładowy moduł EM277 i zaznaczony na nim numer zamówieniowy. Podobne numery znajdują się na wszystkich elementach systemu SIMATIC.



Rys. 10. Odczytywanie numeru zamówieniowego na elementach systemu SIMATIC.

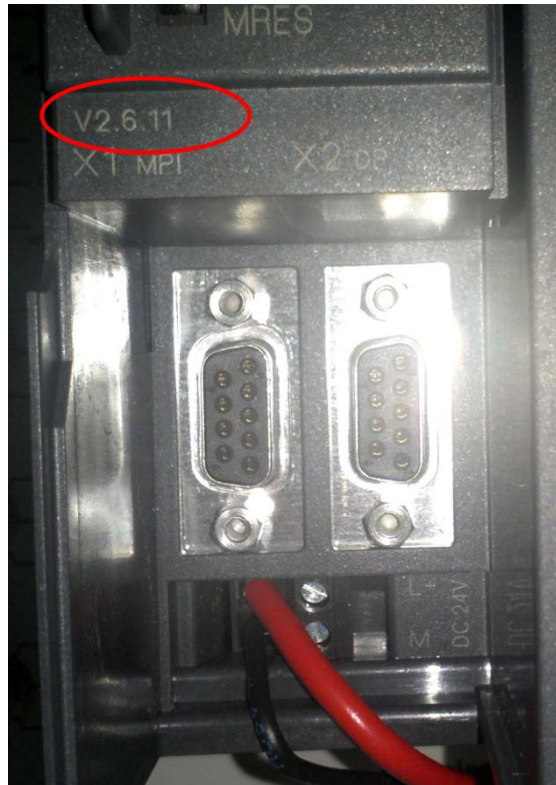
Aby wyszukać moduł o znanym numerze zamówieniowym wpisujemy w polu Find (1) numer zamówieniowy i wciskamy enter (lub klikamy ikonę lornetki z prawej strony pola).

Jak widać wpisując numer zamówieniowy naszego CPU (6ES7 313-6CF03-0AB0) pojawił się folder z 2 pozycjami (2).



Rys. 11. Wybór wersji sprzętowej procesora S7-300.

Niektóre CPU występują w kilku wersjach. W celu dokładnego zidentyfikowania wersji sprzętu jaki posiadamy, podnosimy na CPU klapkę zasłaniającą porty komunikacyjne i odczytujemy naszą wersję (rysunek poniżej).



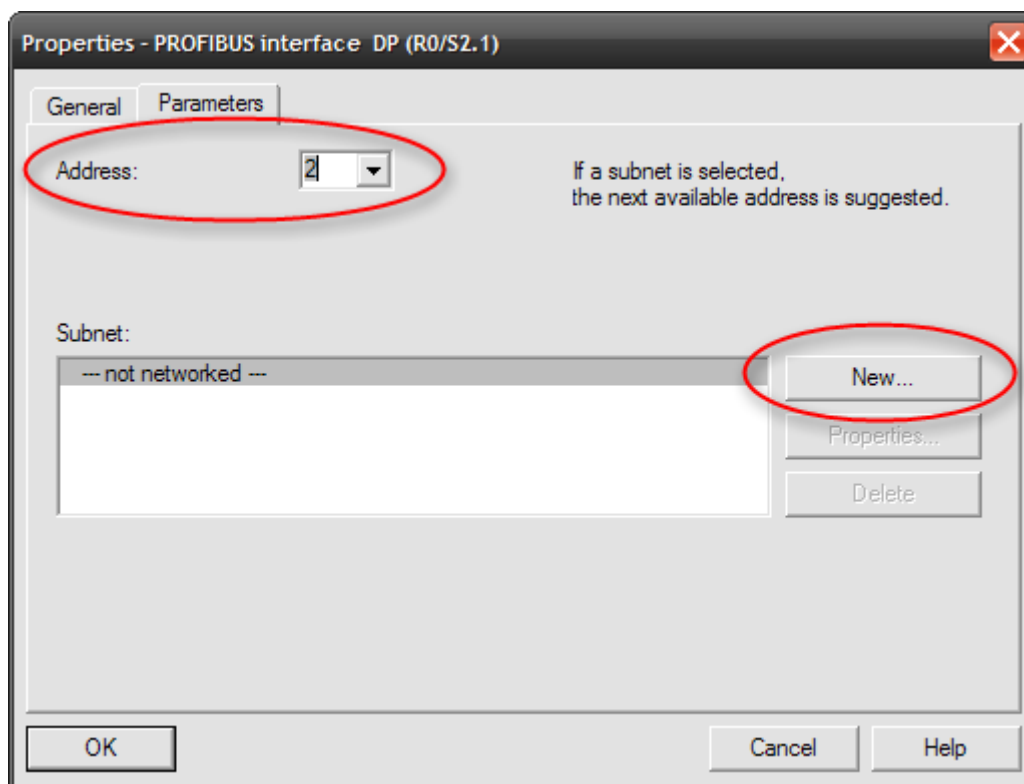
Rys. 12. Odczytanie wersji sprzętowej procesora S7-300.

Wybieramy CPU w wersji 2.6 i przeciągamy na szynę analogicznie jak zasilacz, tym razem na slot drugi.

Po przeciągnięciu CPU na właściwe miejsce na szynie pojawi się okno do parametryzacji interfejsu Profibus w który sterownik jest wyposażony.

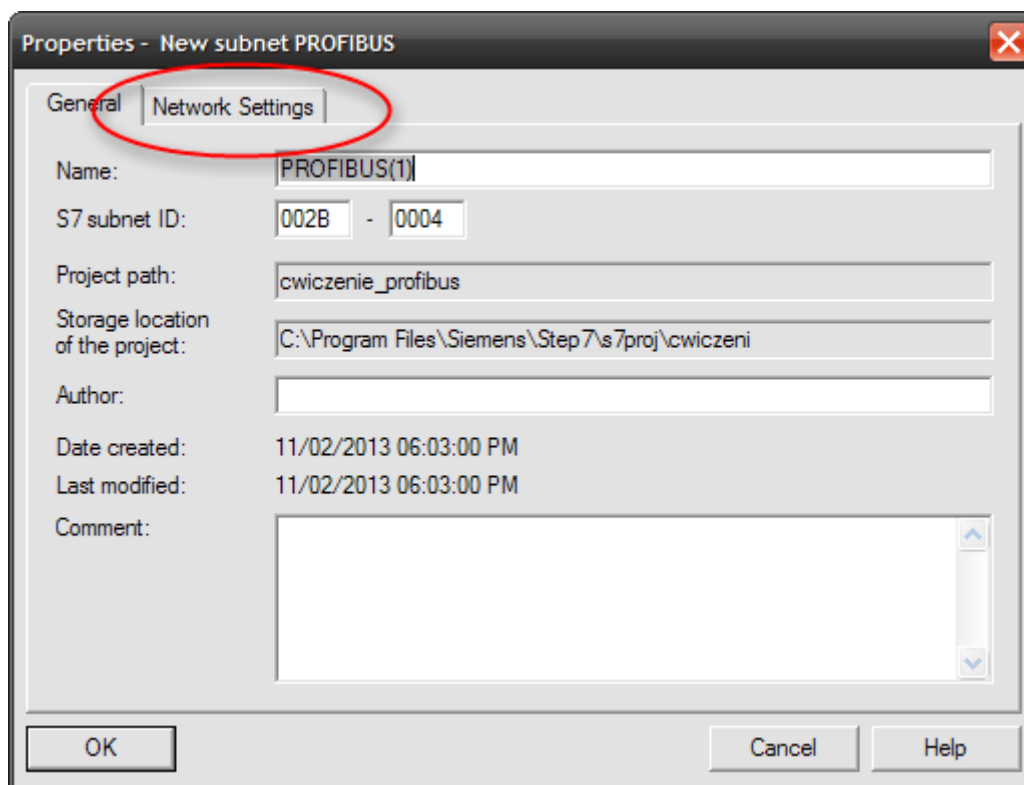
Wybieramy adres dla sterownika, domyślnie jest to 2, taki też ustawiamy.

Następnie klikamy **New...** aby utworzyć sieć Profibus dla wybranego interfejsu.



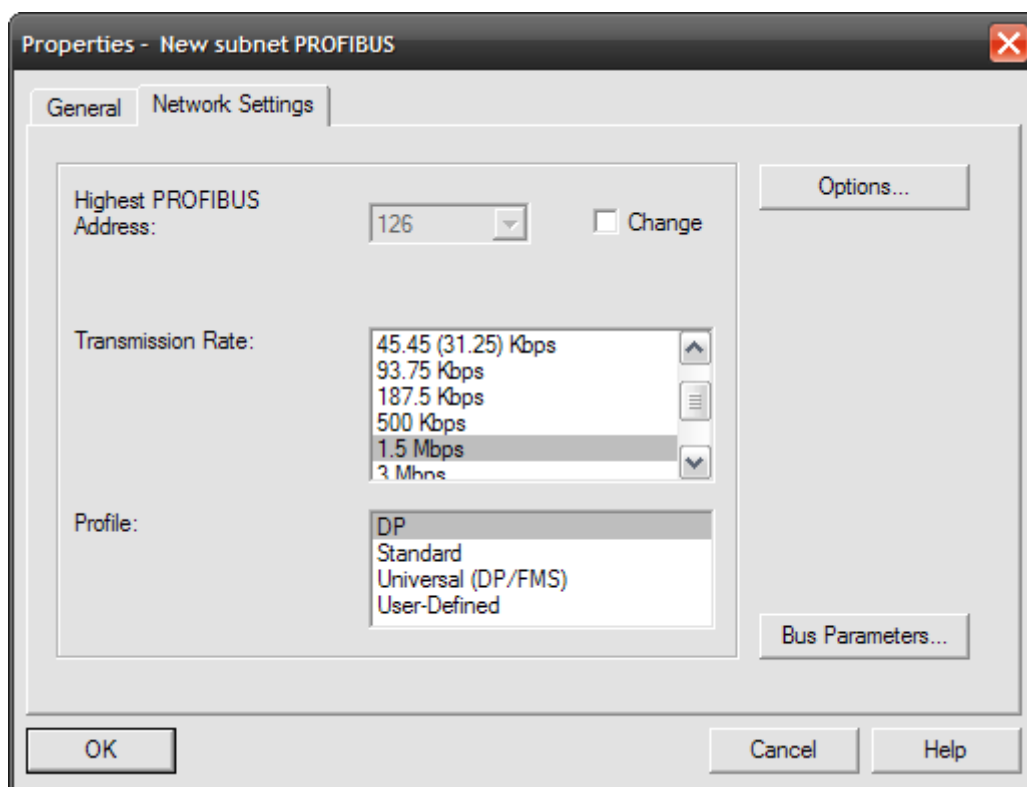
Rys. 13. Utworzenie nowej sieci Profibus.

W otwartym oknie przechodzimy na zakładkę **Network Settings** gdzie możemy wybrać prędkość sieci oraz profil protokołu.



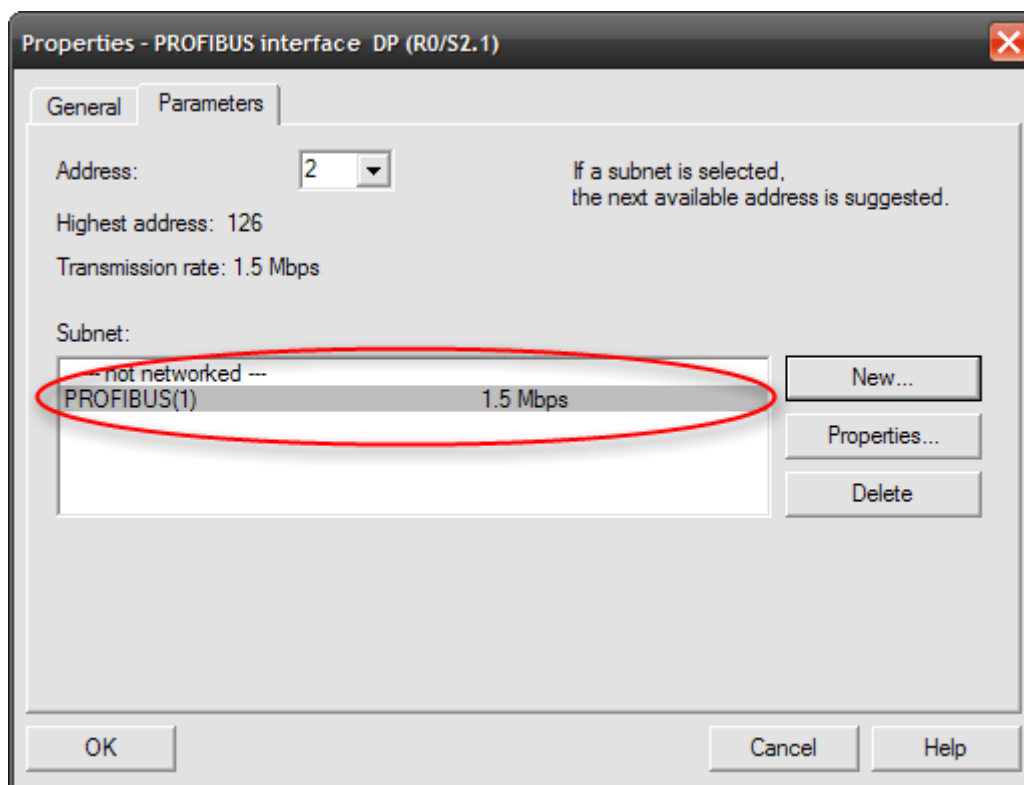
Rys. 14. Okno właściwości sieci Profibus.

Pozostawiamy opcje zaznaczone jak na rysunku poniżej i klikamy OK.



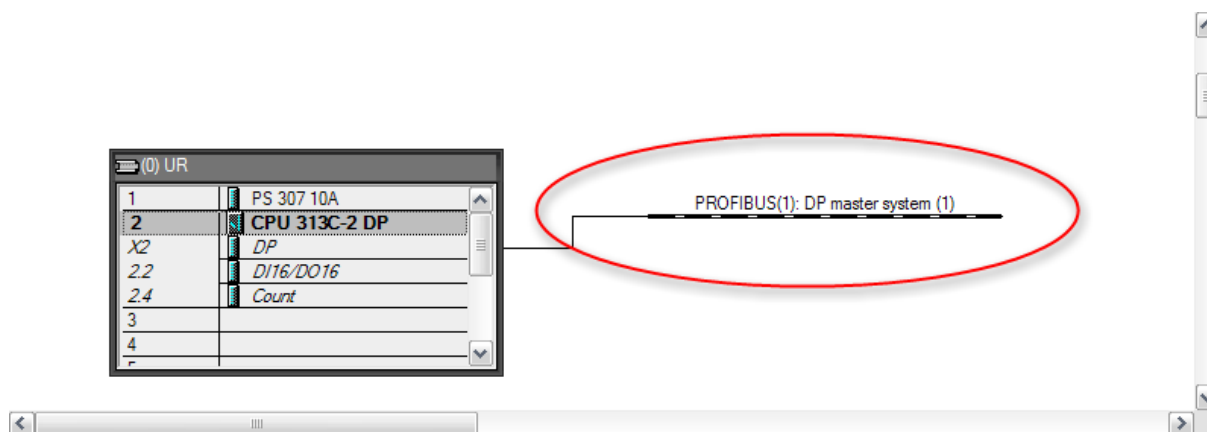
Rys. 15. Wybór prędkości i profilu sieci.

Wrócićmy do poprzedniego okna. Zaznaczamy utworzoną sieć PROFIBUS(1) 1.5 Mbps i klikamy OK.



Rys. 16. Okno ze skonfigurowaną sieć PROFIBUS(1).

Teraz nasz projekt powinien wyglądać jak poniżej.



Rys. 17. Okno edytora HW Config z dodaną siecią Profibus.

Sterownik S7-300 (master) mamy skonfigurowany, pozostało nam dodanie do projektu stacji slave czyli modułu EM277.

Domyślnie moduł ten nie znajduje się w katalogu i należy go doinstalować. Proces instalacji pliku GSD opisano w następnym punkcie. Jeżeli plik GSD dla stacji EM277 został już dodany wcześniej i jest dostępny w katalogu urządzeń, możemy pominąć ten punkt instrukcji.

Instalacja plików GSD dla urządzeń nie znajdujących się w katalogu

Dodaliśmy już sterownik i skonfigurowaliśmy parametry sieci Profibus, której będzie on masterem. Nadaliśmy również adres w sieci Profibus pod którym będzie widziany sterownik S7-300. Teraz do naszej sieci musimy dodać urządzenia slave z którymi sterownik będzie utrzymywał komunikację.

W naszym ćwiczeniu jedyną stacją poza sterownikiem master będzie stacja EM 277 ze sterownikiem S7-200.

Ponieważ domyślnie modułu EM 277 nie ma w katalogu Edytora HW Config, należy doinstalować brakujący plik GSD z opisem modułu. Pliki GSD zawierają opis możliwości urządzenia i są dostarczane przez producentów lub udostępnione do pobrania na ich stronie, w tym wypadku przez Siemens, który jest producentem modułu EM277.

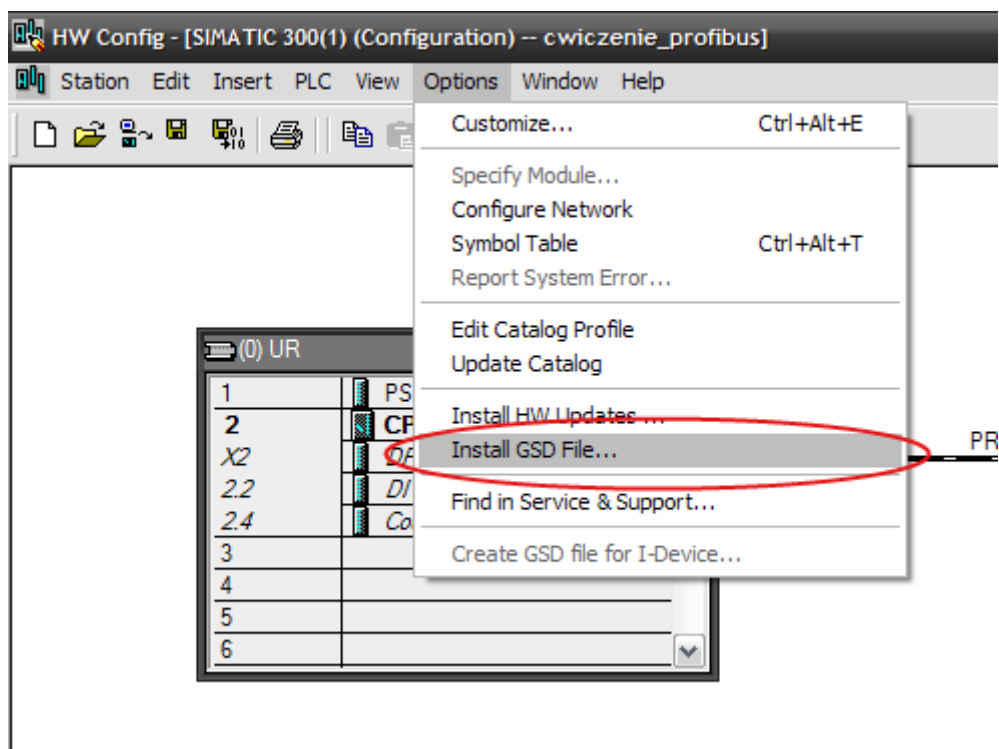
Przechodzimy pod adres:

<http://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&objId=113652&ehbid=113652&hlnav=yes&skm=1&content=adsearch%2Fadsearch%2Easpx&cssearchengine=new&lang=en&siteid=cseus&aktprim=0&objaction=csoopen&extranet=standard&viewreg=WW>

W oknie wyszukiwarki internetowej szukamy tekstu em277 (CTRL + F) i pobieramy archiwum em277.zip z potrzebnym plikiem GSD.

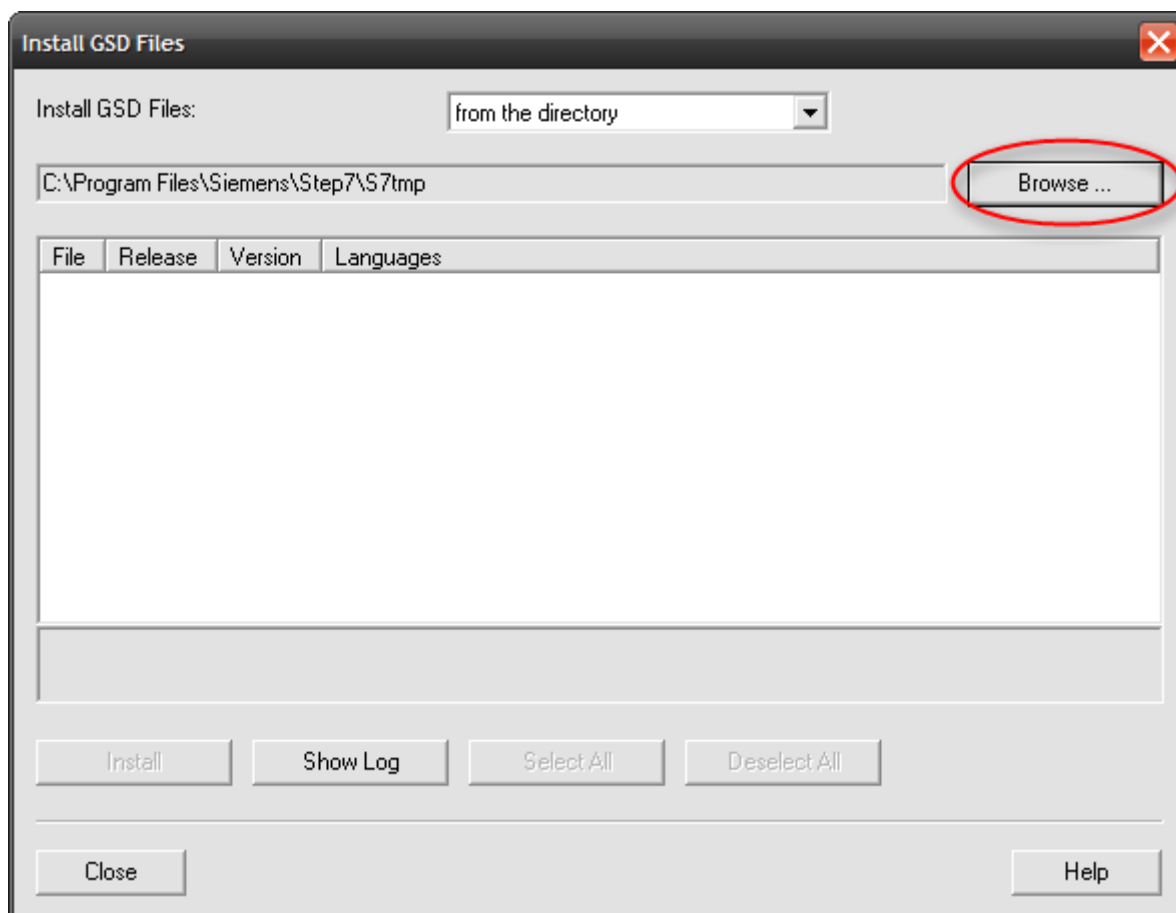
Po zakończeniu pobierania rozpakowujemy plik do wybranego przez nas katalogu np. **C:/EM277**.

W edytorze HW Config klikamy w górnym menu pozycję **Options**, następnie z rozwijanego menu wybieramy **Install GSD File...**



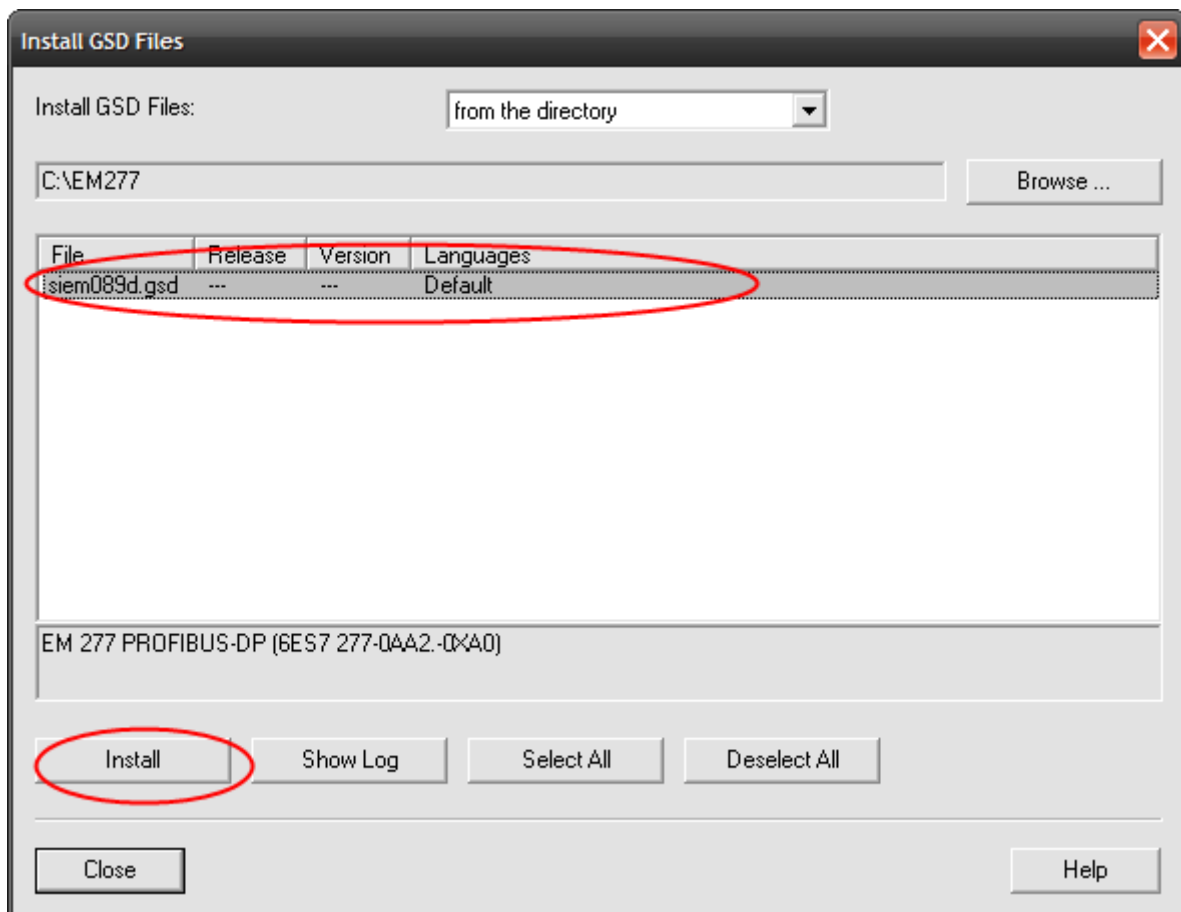
Rys. 18. Menu instalacji pliku GSD.

W oknie które się otworzy klikamy **Browse...** i wskazujemy katalog do którego rozpakowaliśmy pobrany plik GSD.



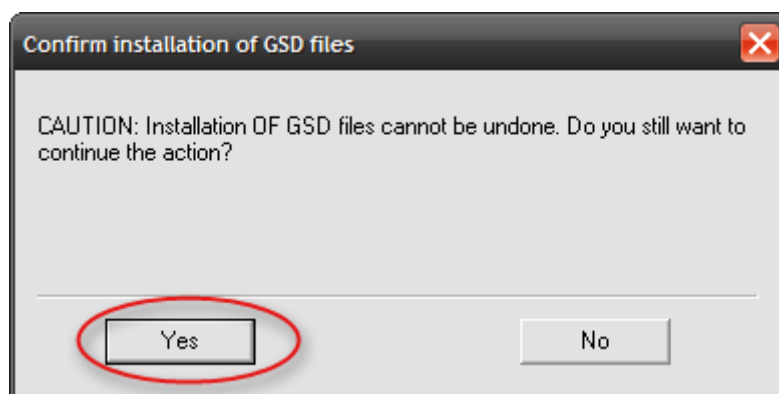
Rys. 19. Wybór katalogu z pobranym plikiem GSD.

Jeżeli wybraliśmy prawidłowy katalog, w oknie wyświetli się lista plików GSD znajdujących się w nim.



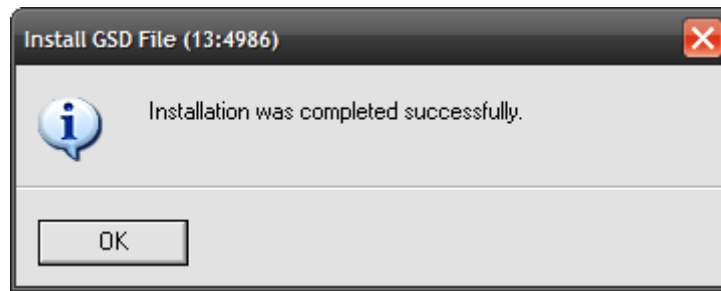
Rys. 20. Wybór pliku GSD do zainstalowania.

Zaznaczamy pozycję **siem089d.gsd**. W dolnej części okna wyświetla się nazwa modułu dla którego przeznaczony jest plik oraz jego numer zamówieniowy. Klikamy **Install**. Potwierdzamy instalację – klikamy **Yes**.



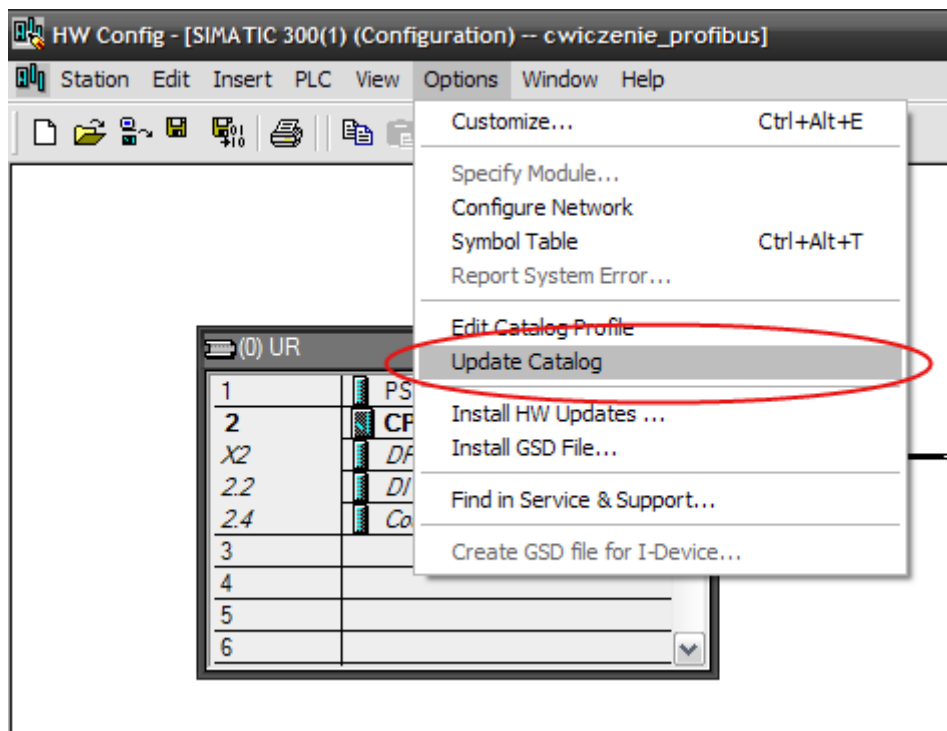
Rys. 21. Potwierdzenie instalacji pliku GSD.

Jeżeli wszystko przebiegło pomyślnie program wyświetli komunikat.



Rys. 22. Komunikat o zakończonej pomyślnie instalacji.

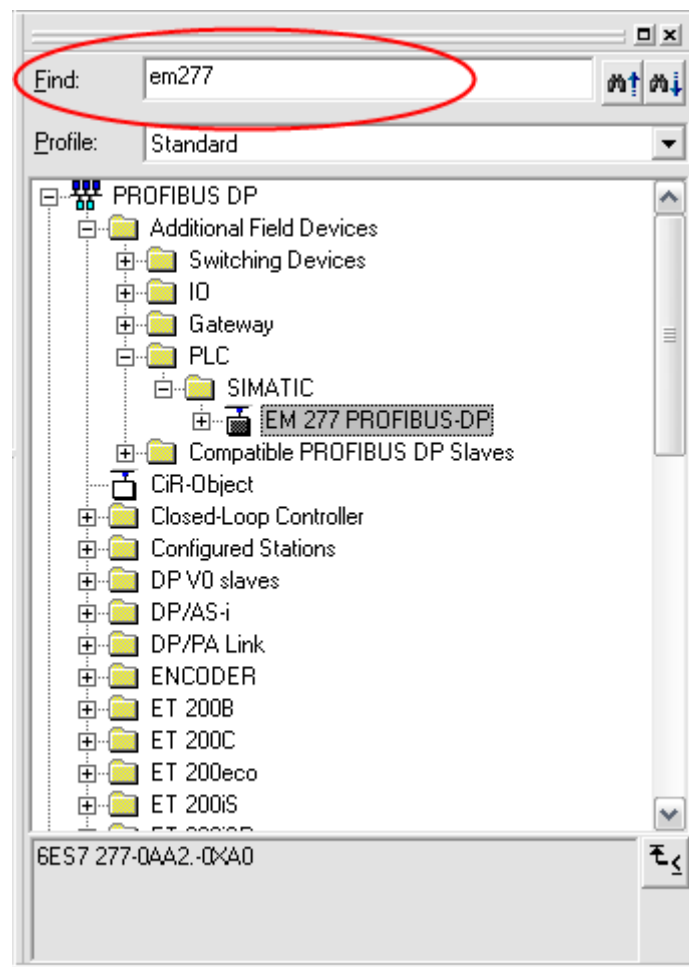
Postało nam odświeżyć katalog aby możliwe było korzystanie z nowych urządzeń. W tym celu klikamy w menu **Options** -> **Update Catalog**.



Rys. 23. Odświeżenie zawartości katalogu urządzeń.

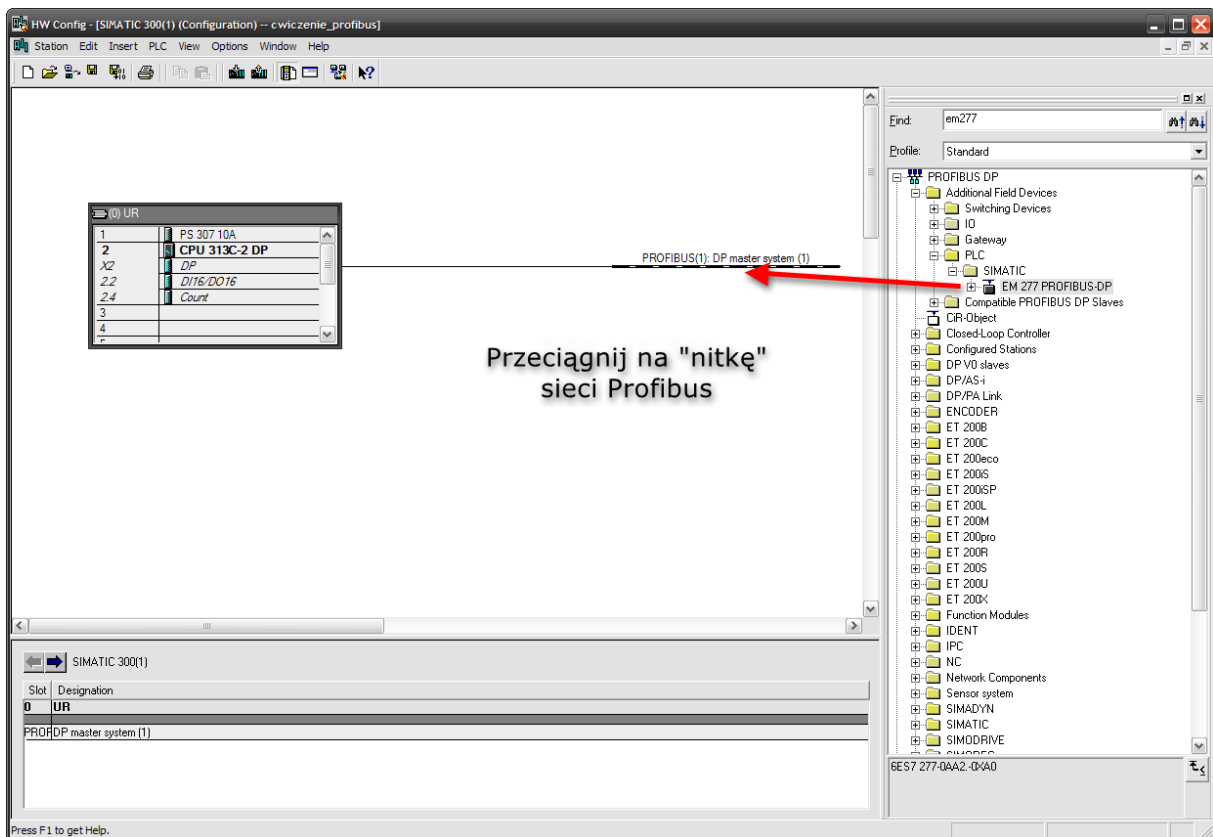
Dodanie stacji slave EM277 do projektu

W polu Find katalogu wpisujemy em277 (bądź nr zamówieniowy) i wciskamy enter. Podświetli się moduł EM 277 PROFIBUS-DP.



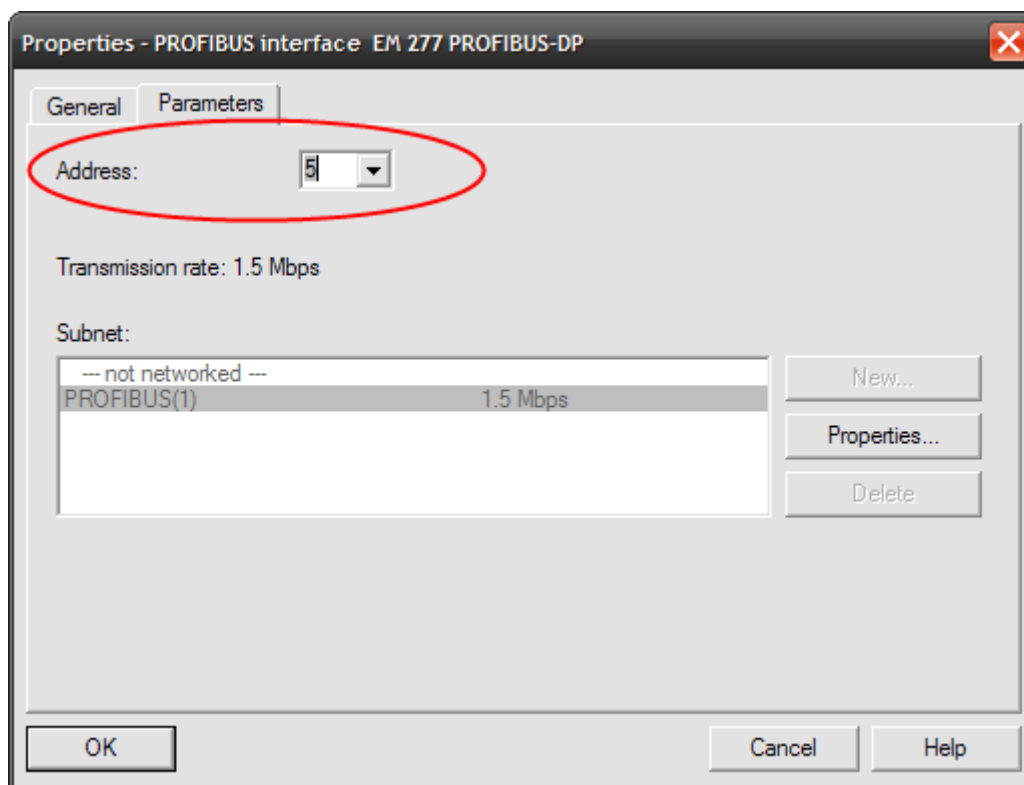
Rys. 24. Wybór z katalogu modułu EM277.

Przeciągamy teraz moduł EM277 na „nitkę” sieci **PROFIBUS(1)** widoczną w projekcie.



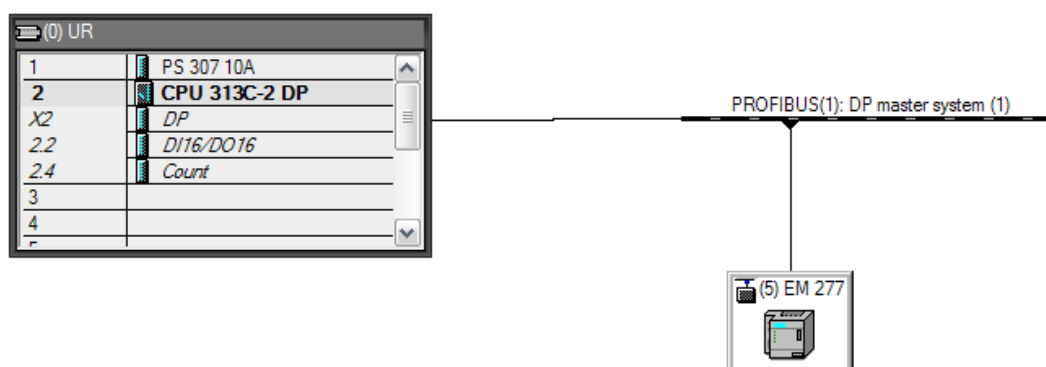
Rys. 25. Dodanie modułu EM277 do konfiguracji sieci.

Moduł zostanie dodany do projektu i podłączony do wybranej sieci Profibus. Otworzy się okno konfiguracji w którym wybieramy adres 5 dla modułu EM277 i klikamy OK.



Rys. 26. Konfiguracja adresu dla modułu EM277.

Teraz widok połączeń w naszym projekcie powinien wyglądać jak poniżej.



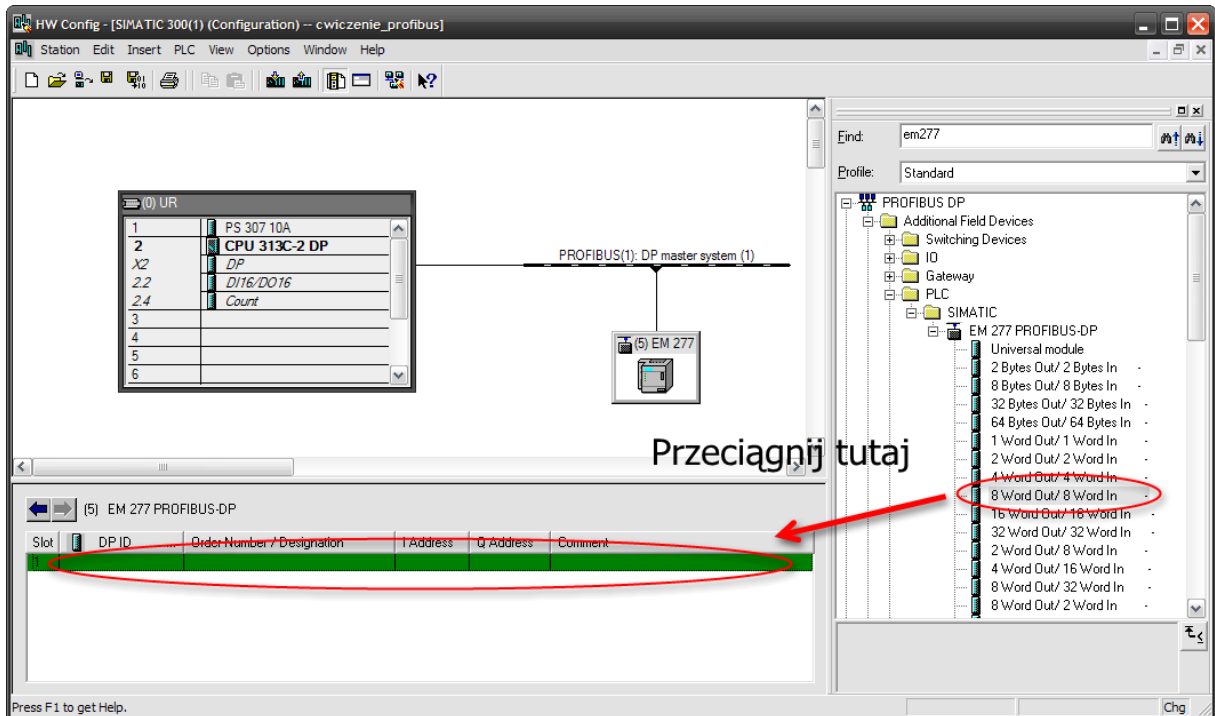
Rys. 27. Edytor HW Config z dodaną do sieci pod adresem 5 stacją EM277.

Musimy jeszcze skonfigurować w module EM277 rodzaj i ilość danych jaką będziemy wymieniać z masterem sieci S7-300.

Trzeba pamiętać, że jednocześnie może być dodany do modułu EM277 tylko element określający ilość danych wejściowych i wyjściowych (czyli tabela ze „slotami” tego modułu może zawierać jedną pozycję).

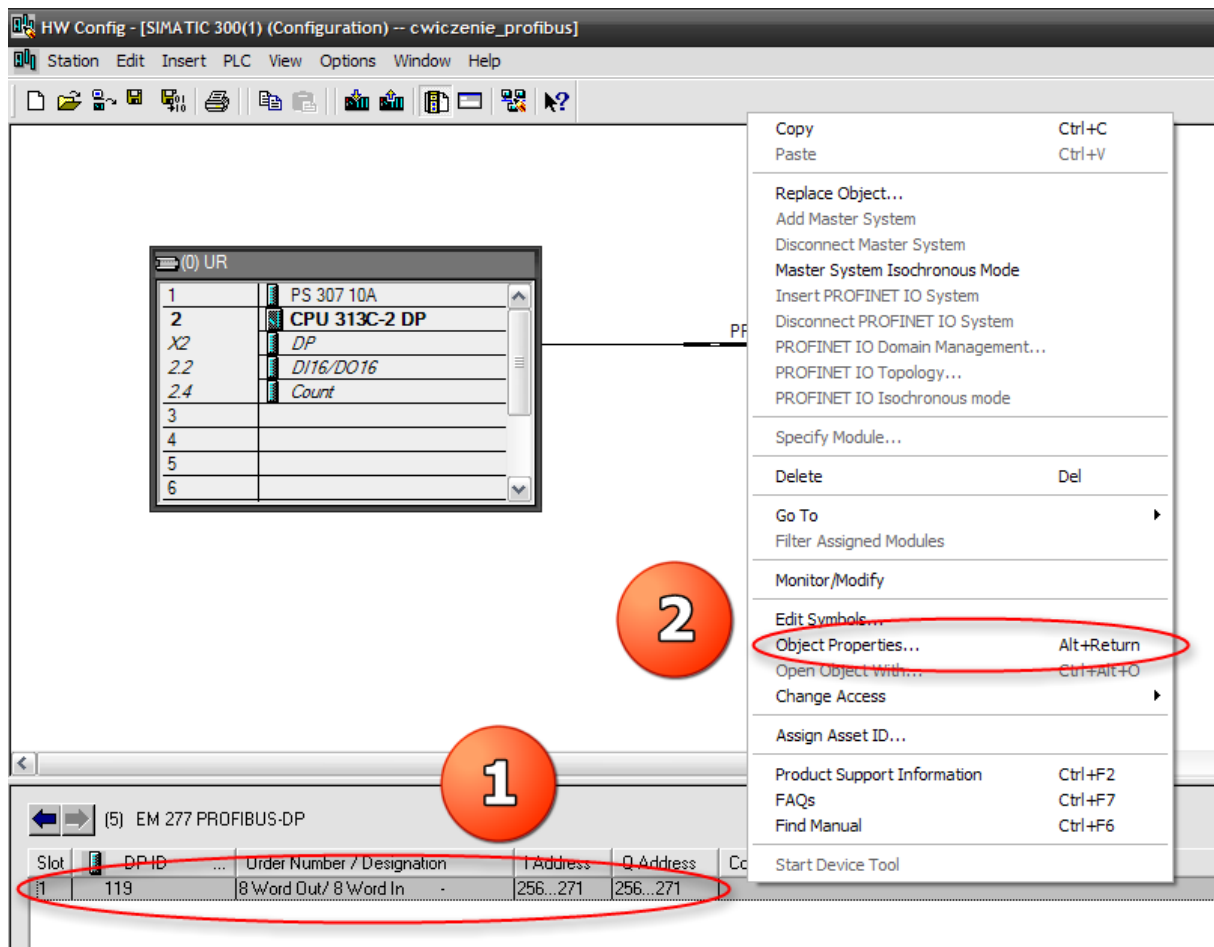
Aby zapewnić sobie odpowiedni obszar pamięci wejść/wyjść, wykorzystywany podczas komunikacji Profibus wybierzemy element wysyłający 8 słów oraz odbierający 8 słów.

Ponownie odnajdujemy w katalogu element EM 277 i z rozwiniętego drzewa wybieramy moduł **8 Word Out/8 Word In**, następnie przeciągamy go do tabeli jak na rysunku poniżej.



Rys. 28. Konfiguracja rozmiaru wymienianych danych przez moduł EM277.

Prawym przyciskiem kliknij na dodany moduł (1) i wybierz **Object Properties** (2).

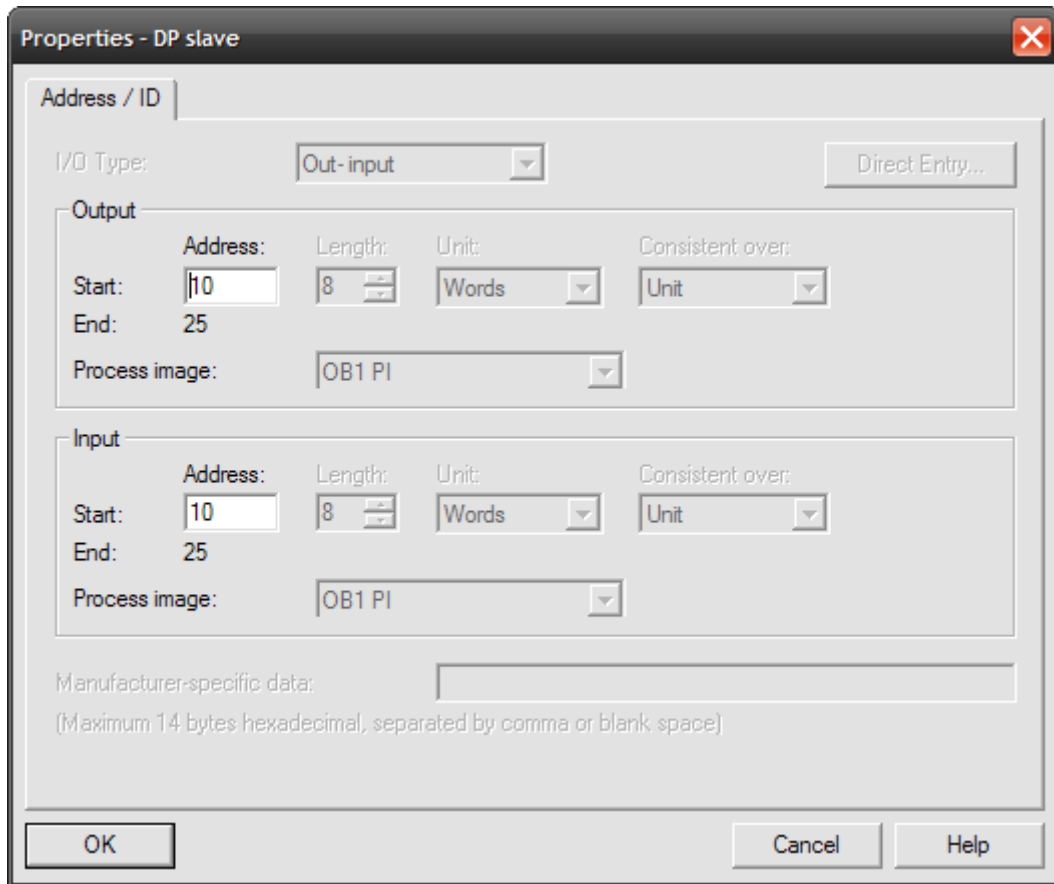


Rys. 29. Przydzielenie adresów w pamięci S7-300 na wymieniane dane.

W polu **Address** dla Output oraz Input wpisz **10**. Jest to początek przestrzeni pamięci przydzielonych na wysyłanie oraz odczytywanie danych w sterowniku S7-300.

W polu Consistent over widnieje napis Unit. Oznacza to, że dane mogą być przekazywane słowo po słowie.

Klikamy **OK**.



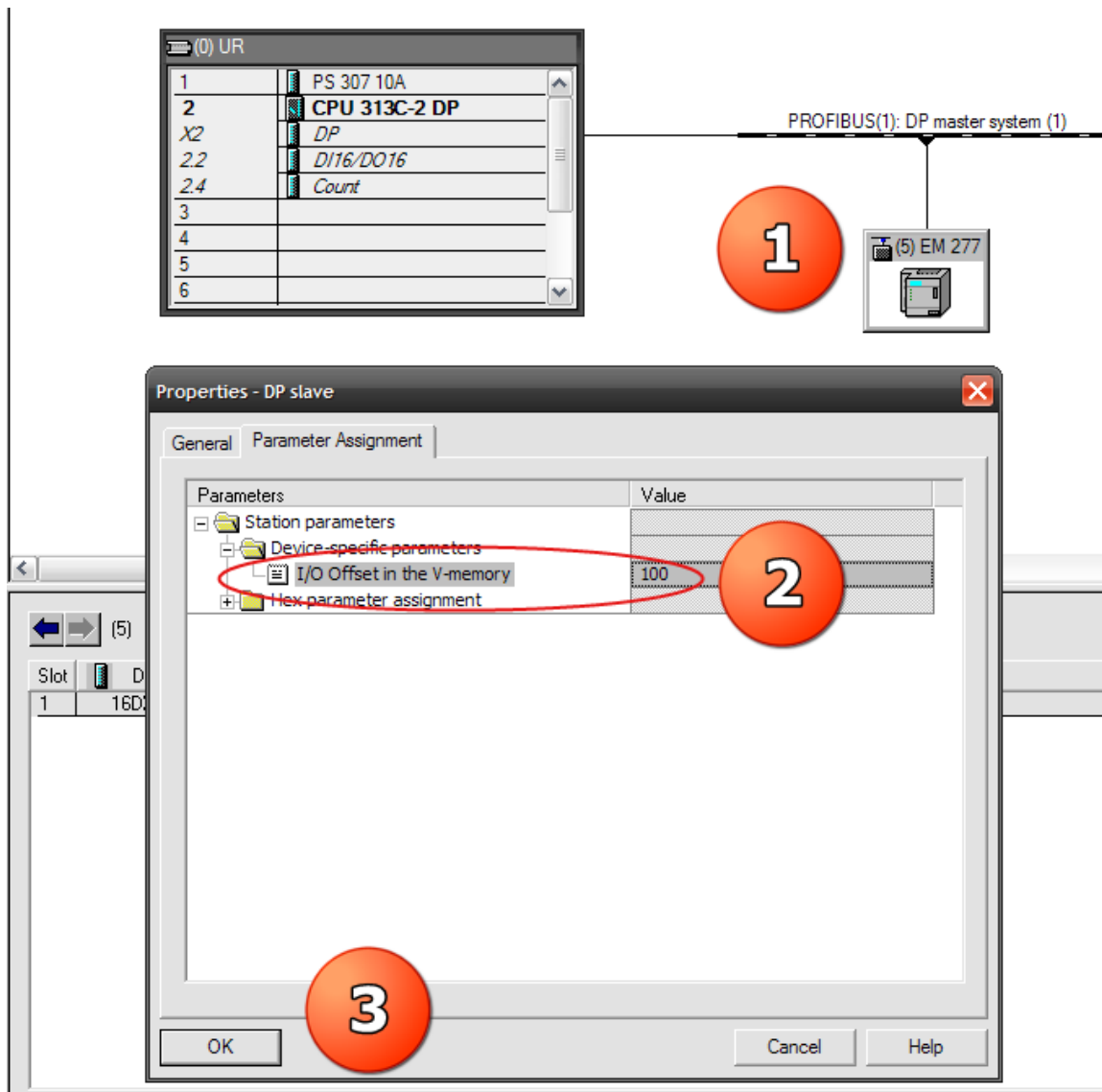
Rys. 30. Określenie adresów dla wejść i dla wyjść w pamięci S7-300.

W tej konfiguracji bajty IB10 do IB25 będą przechowywały informację odczytane z EM277 (wejścia), natomiast bajty QB10 do QB25 będą przechowywały informację do wysłania na EM277 (wyjścia).

Teraz klikamy prawym przyciskiem na ikonie stacji EM 277 w projekcie (1) i wybieramy **Object Properties**.

W oknie które się otworzy w zakładce **Parameter Assignment** określamy początek przestrzeni adresowej pamięci V przydzielonej na odbieranie oraz wysyłanie danych (2) w sterowniku S7-200. Wpisujemy **100** i klikamy ok. (3).

W ten sposób ustawiliśmy adres początkowy od którego w pamięci Variable Memory sterownika S7-200 będą dostępne informacje przesyłane z/do mastera.

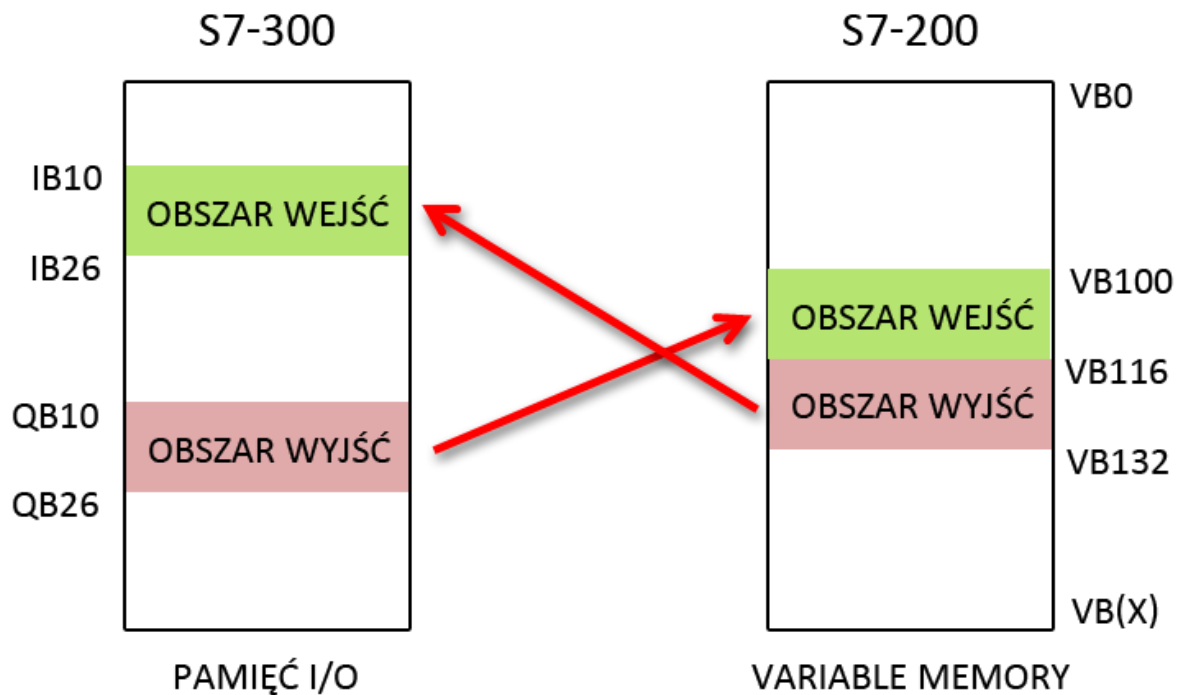


Rys. 31. Określenie obszaru w pamięci S7-200 do komunikacji Profibus.

Ponieważ wybraliśmy w konfiguracji EM277 moduł **8 Word Out/8 Word In**, więc z punktu widzenia S7-200 przydział pamięci (zapisany bajtowo) będzie następujący:

- VB100 do VB115 – dane odebrane (8 słów wychodzących z S7-300 i przychodzących do S7-200)
- VB116 oraz VB131 – dane do wysłania (8 słów wchodzących do S7-300 i wysłanych z S7-200).

Rozkład przestrzeni pamięci wejść (odczytu) i wyjść (zapisu) dla obu sterowników wygląda teraz następująco:



Rys. 32. Rozkład pamięci w obu sterownikach po zakończeniu konfiguracji.

Analogicznie wygląda adresowanie przy wyborze jednego z pozostałych modułów np. **2 Words Out/2 Words In**.

Najpierw ustawiamy adresy dla Input i Output (mogą być oczywiście różne np. 256 dla Input i 20 dla Output), jakie będą obowiązywały w S7-300 a ustawiony Offset modułu EM277 określa gdzie zaczyna się wykorzystywana Variable Memory w S7-200.

Przyjmijmy, że Input Address to 256. Czyli pierwsze słowo odbieramy w S7-300 pod adresem IW256 (drugie IW258).

Offset jaki wpisaliśmy to 200, więc słowo IW256, w sterowniku S7-200 będzie dostępne pod adresem VW204 (IW258 = VW206).

Jest tak dlatego, że adresy VW200 oraz VW202 są zarezerwowane dla 2 słów odbieranych ze sterownika S7-300 (nasze 2 Words Out wychodzące z S7-300 i wchodzące do S7-200).

Zapisanie i Kompilacja konfiguracji sprzętowej

Po zakończeniu konfigurowania sprzętu należy utworzoną konfigurację zapisać i skompilować do postaci bloku *System data*.

W tym celu kliknij na ikonie **Save & Compile** na pasku narzędziowym.



Rys. 33. Zapis i kompilacja konfiguracji sprzętowej sterownika S7-300.

Po skompilowaniu konfiguracja sprzętowa będzie widoczna w programie sterownika w katalogu Blocks jako blok System data (📁 System data). Po napisaniu programu zostanie ona wgrana wraz z nim do sterownika.

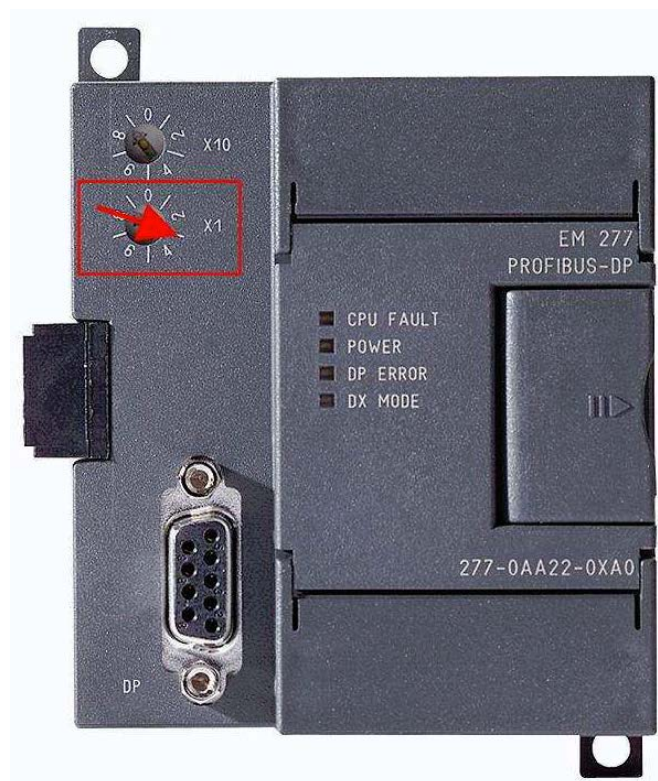
Po zapisaniu i kompilacji zamykamy edytor HW Config i przechodzimy do głównego okna projektu Simatic Manager.

Ustawienie adresu sprzętowego na module EM277

Zanim napiszemy program wykorzystujący komunikację Profibus, musimy jeszcze ustawić sprzętowo nadany w edytorze HW Config adres 5 dla stacji EM277.

Wszystkie czynności wykonujemy przy wyłączonym zasilaniu, więc najpierw odłączamy zasilanie od sterownika S7-200.

W tym celu obracamy za pomocą płaskiego wkrętaka widoczne na poniższym rysunku ustawniki dziesiętne, tak aby górny był w pozycji 0 a dolny zaznaczony na poniższym rysunku wskazywał 5.



Rys. 34. Sprzętowe ustawianie adresu na module EM277.

W ten sposób ustawiliśmy na module EM277 adres 5 w sieci Profibus, więc konfiguracja rozsyłana do stacji slave przez mastera przy starcie sieci zostanie prawidłowo odebrana przez moduł.

Teraz ponownie włączamy zasilanie sterownika S7-200.

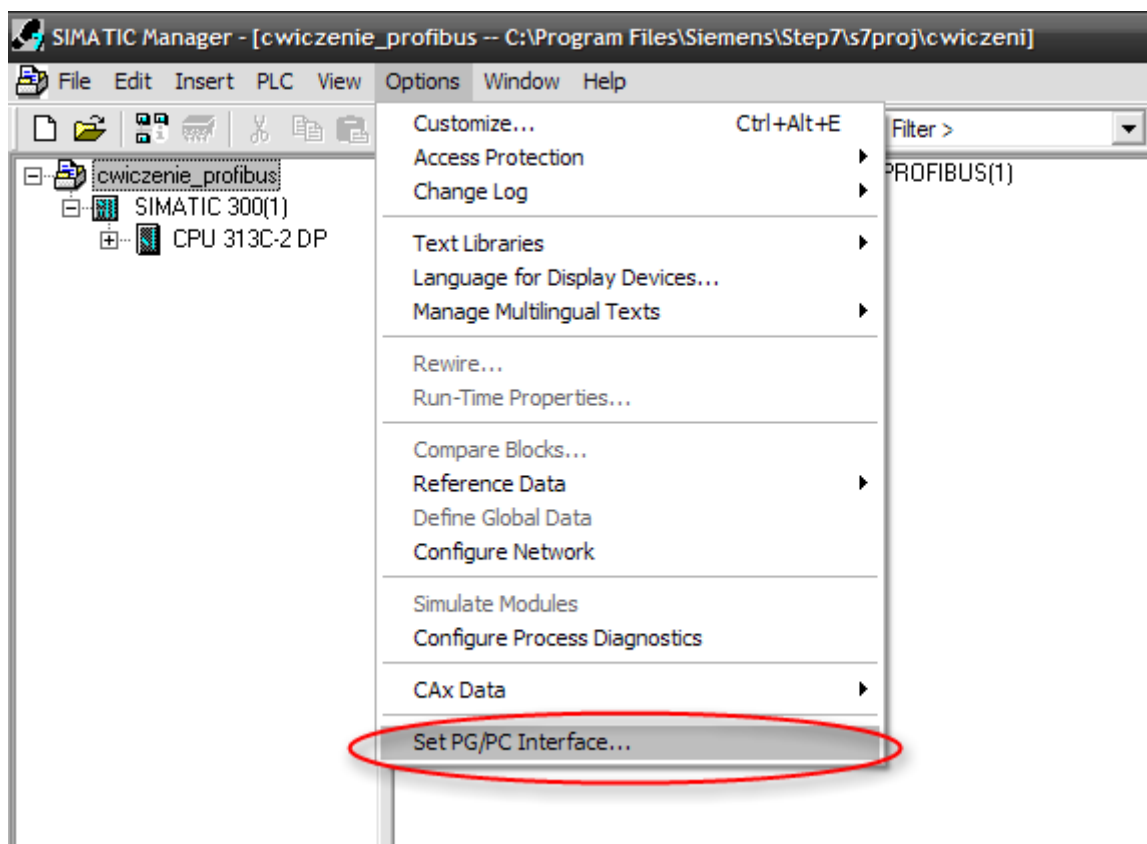
Przykładowy program do testu komunikacji Profibus dla S7-300

Konfiguracja połączenia komputer – sterownik S7-300

Do połączenia ze sterownikiem wykorzystamy interfejs Accon NetLink Pro Compact i protokół Profibus. Interfejs podłączamy do portu oznaczonego DP na sterowniku S7-300, a kabel sieciowy wychodzący z interfejsu podłączamy do karty sieciowej w komputerze na którym pracujemy.

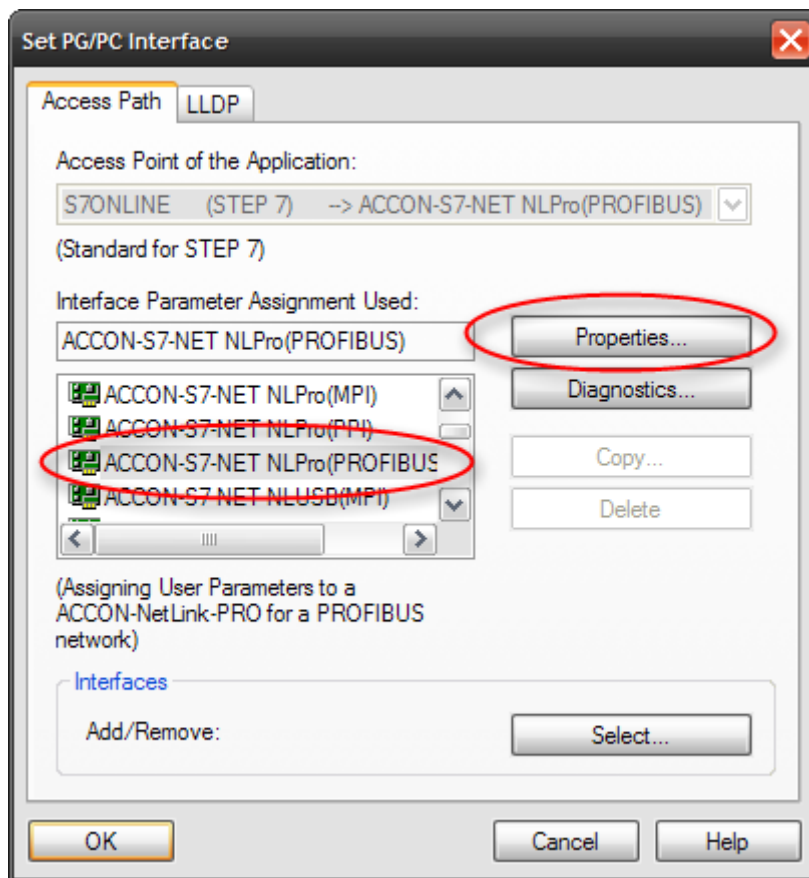
Karta sieciowa powinna mieć przypisany adres z tej samej podsieci jak interfejs (adres interfejsu to 10.0.0.1), a sterownik powinien być zasilony podczas poniższej konfiguracji, gdyż z jego portu zasilany jest Accon.

W celu konfiguracji połączenia komputera ze sterownikiem przechodzimy w programie Simatic Manager do menu **Options -> Set PG/PC Interface...**



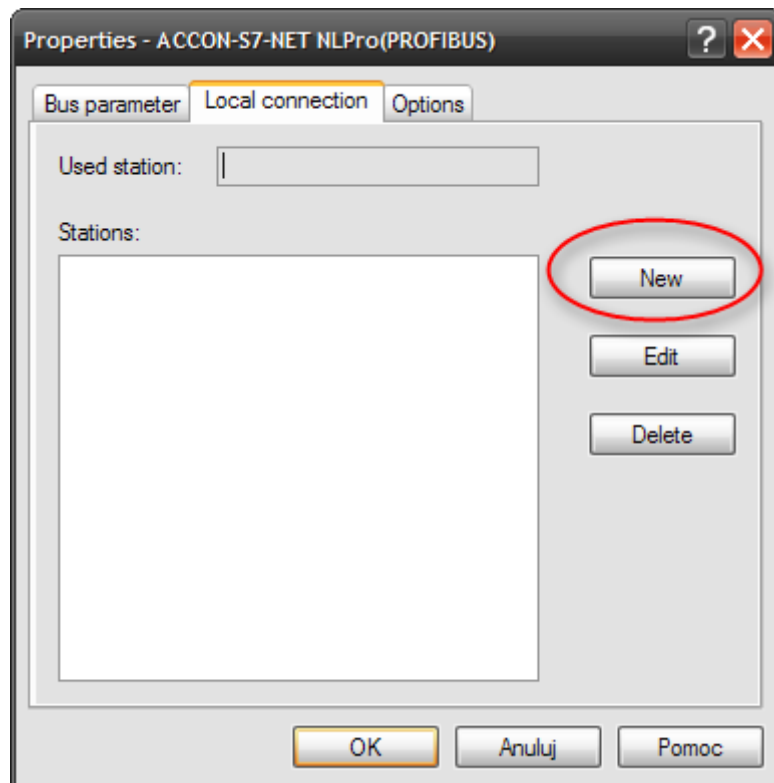
Rys. 35. Przejście do konfiguracji interfejsu PG/PC w programie Simatic Manager.

W otwartym oknie wybieramy interfejs Accon-S7-Net NLPro(PROFIBUS) i klikamy **Properties**.



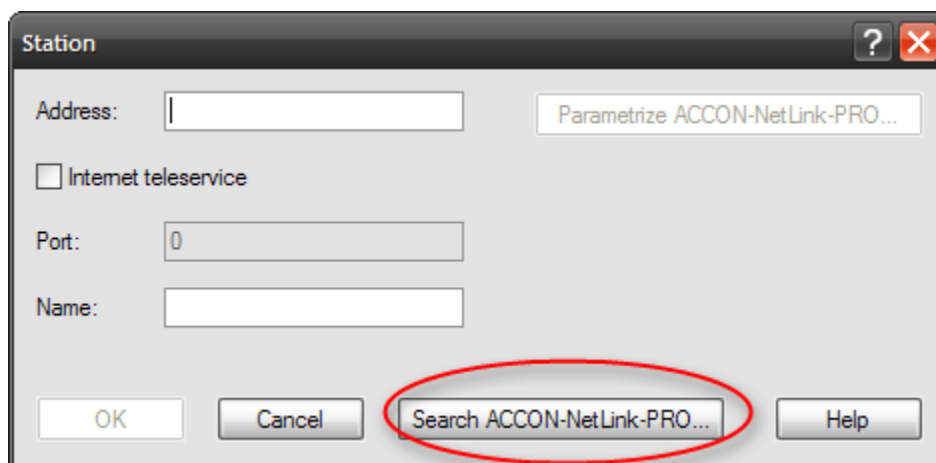
Rys. 36. Wybór interfejsu do konfiguracji.

Przechodzimy na zakładkę **Local connection** i jeżeli lista skonfigurowanych stacji (interfejsów Accon) jest pusta klikamy **New** (jeżeli został już skonfigurowany interfejs o adresie 10.0.0.1 po prostu go wybieramy i klikamy ok).



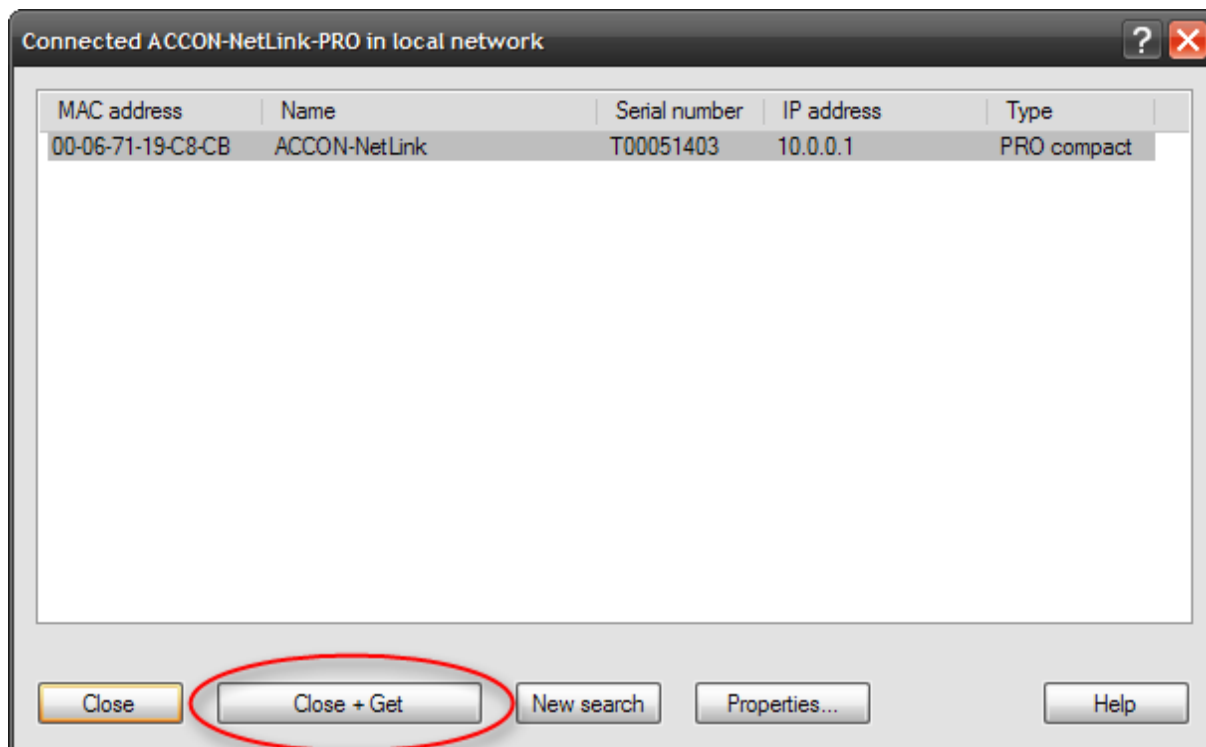
Rys. 37. Dodanie nowego interfejsu Accon do listy stacji.

W otwartym oknie klikamy Search Accon-NetLink-PRO aby wyszukać podłączony interfejs.



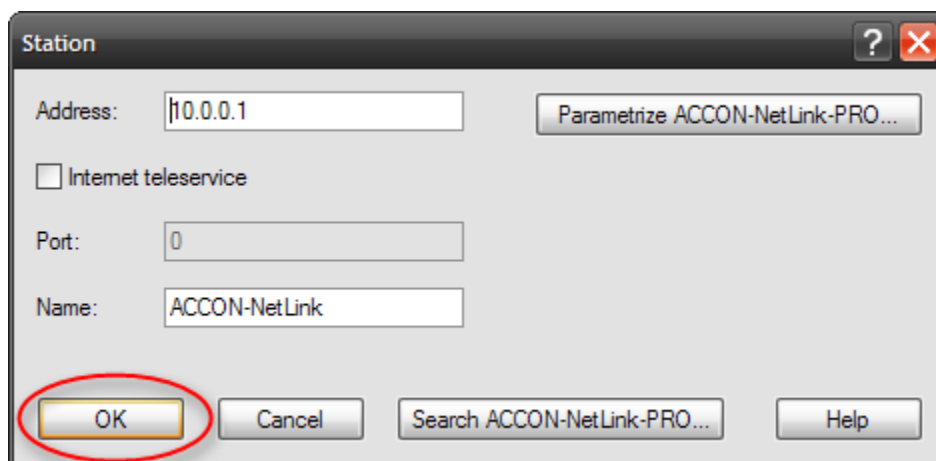
Rys. 38. Wyszukiwanie podłączonego interfejsu.

Kiedy wyszukiwanie dobiegnie końca, wybieramy znaleziony interfejs i klikamy **Close + Get**.



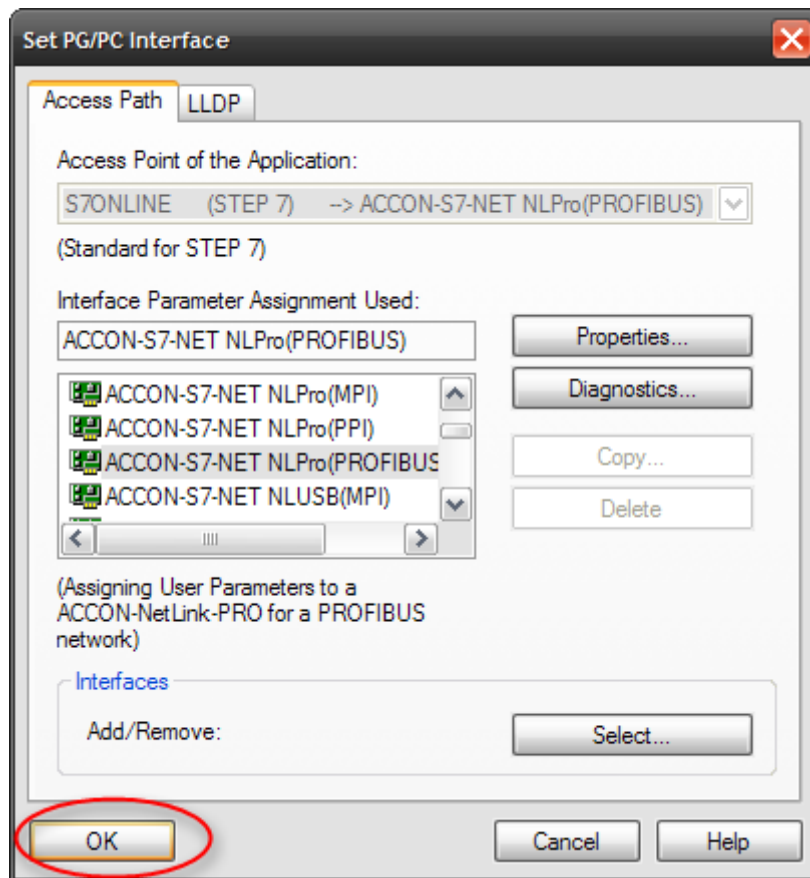
Rys. 39. Wyniki wyszukiwania.

Klikamy **OK**, aby potwierdzić adres interfejsu na którym pracujemy.



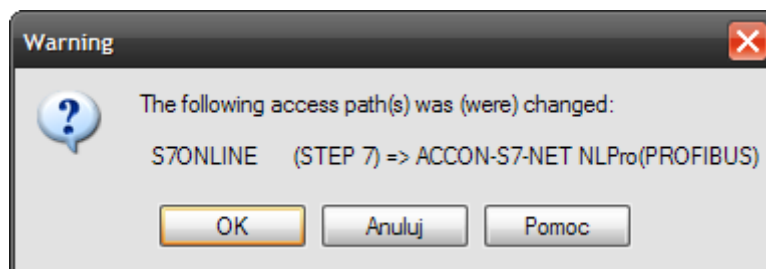
Rys. 40. Dodajemy znaleziony interfejs do listy stacji.

Wrócić do listy wyboru interfejsu PG/PC, klikamy **OK**, aby zamknąć okno.



Rys. 41. Potwierdzenie ustawień Set PG/PC Interface.

Program zapyta czy chcemy zmienić sposób komunikacji sterownika z komputerem (tzw. Access path), klikamy **OK**.



Rys. 42. Potwierdzenie zmiany sposobu dostępu do sterownika.

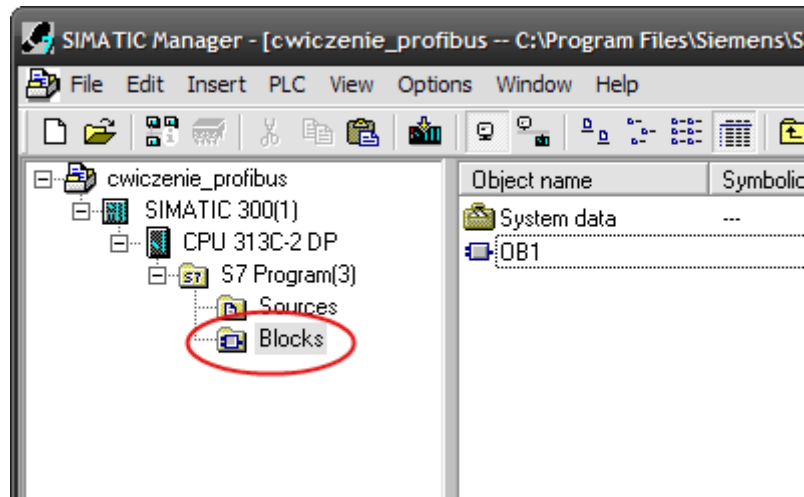
W ten sposób przygotowaliśmy nasz projekt do komunikacji ze sterownikiem.

Przygotujemy teraz prosty program do przetestowania komunikacji Profibus pomiędzy sterownikami w naszej sieci.

Podstawowe informacje o edytorze programowym Step7

Przechodzimy teraz do okna SIMATIC Manager i w drzewie projektu przechodzimy do gałęzi **S7 Program -> Blocks**.

Katalog ten zawiera bloki programowe składające się na nasz cały program.



Rys. 43. Widok katalogu bloków programowych po zakończonej konfiguracji sprzętowej.

Program sterowania z odpowiednimi danymi może być zapisany do sterownika w postaci różnych typów bloków, w zależności od wymagań procesowych.

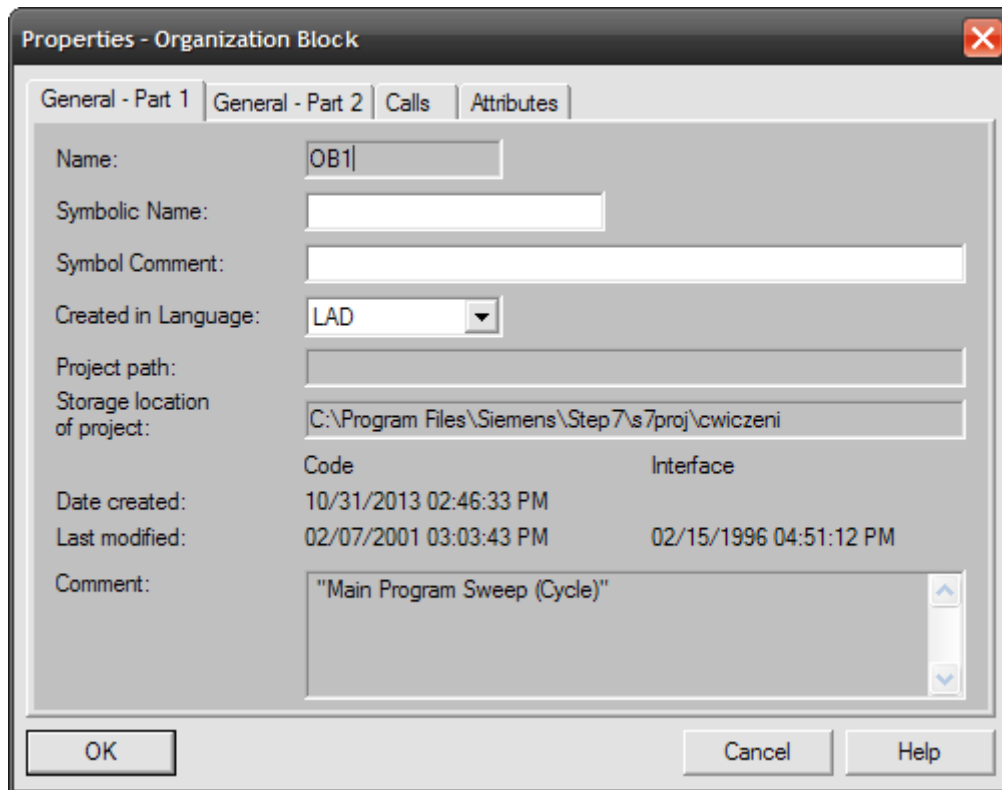
1. **Bloki organizacyjne OB** – łączą program sterowania z systemem operacyjnym CPU. Tylko program zapisany w OB1 jest wykonywany cyklicznie przez sterownik. W bloku OB1 wywołujemy również inne utworzone bloki programowe (FC,FB).
2. **Funkcje FC, SFC** – Bloki funkcyjne FC zawierają wydzieloną część programu sterowania. Mogą być wywoływane z parametrami, tak więc są wykorzystywane do elementów programowych powtarzających się lub do budowania kompletnych części funkcjonalności (np. funkcja odpowiedzialna za obliczenia matematyczne w programie).
Systemowe funkcje SFC są zintegrowane w systemie operacyjnym sterownika. Ich funkcjonalność jest zdefiniowana przez producenta a opis znajduje się w pomocy programu Step7,
3. **Bloki funkcyjne FB, SFB** – posiadają takie same możliwości jak funkcje. Dodatkowo posiadają przypisany obszar pamięci w postaci bloku danych.
4. **Bloki danych DB** – są to obszary w programie sterowania zdefiniowane i zarządzane przez użytkownika. Umożliwiają przechowywanie i organizację zmiennych wykorzystywanych np. przy komunikacji z systemem nadrzędnym.

Jak widać obecnie nasz program zawiera tylko skompilowane dane konfiguracyjne (System data) oraz domyślny główny blok programowy OB1.

Blok OB1 jest jedynym blokiem wykonywanym przez sterownik cyklicznie, tak więc wszystkie podprogramy (bloki FC, FB) muszą zostać w nim wywołane aby mogły funkcjonować.

Klikamy dwa razy na bloku OB1 aby otworzyć go w edytorze programu.

SIMATIC Manager zapyta nasz w jakim języku chcemy programować blok. Wybieramy LAD (język drabinkowy) lub FBD w zależności od preferencji i klikamy **OK**.



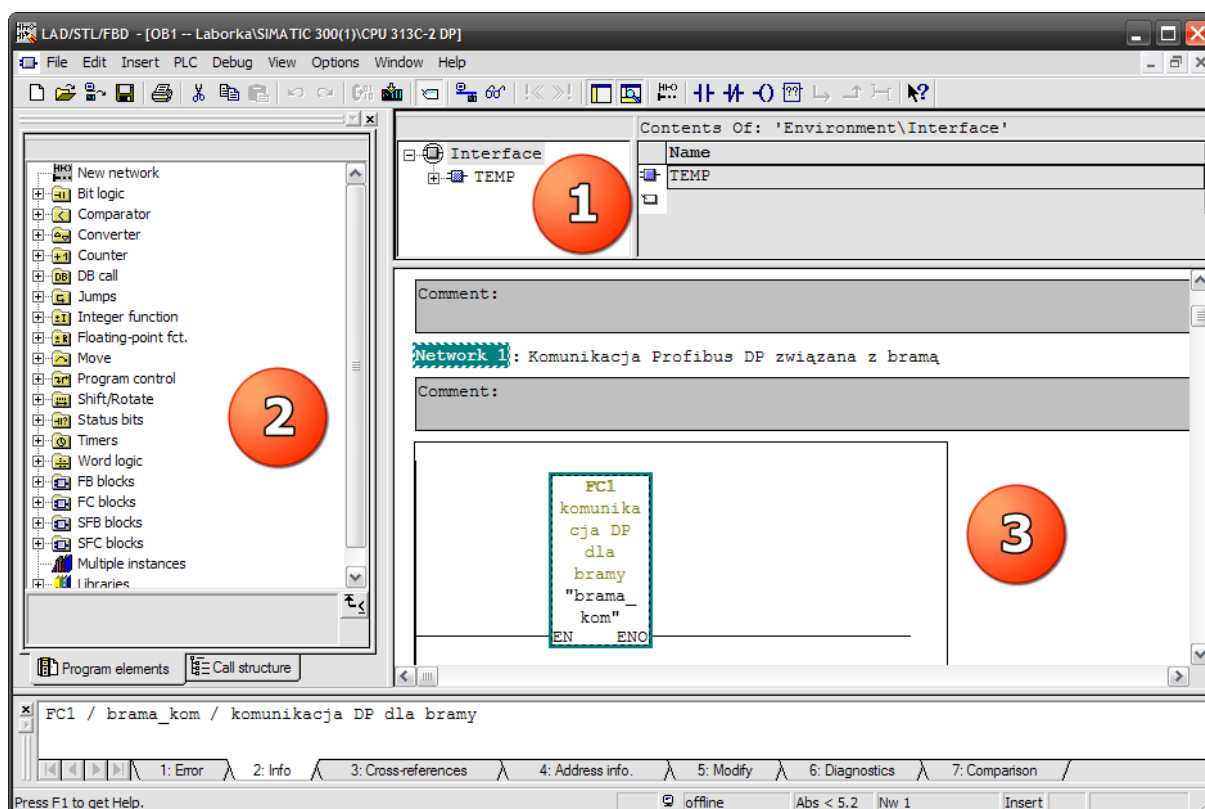
Rys. 44. Wybór języka programowania i opis bloku OB1.

Przy bardziej rozbudowanych programach blok OB1 służy głównie do wywoływania podprogramów zapisanych w innych blokach FC czy FB.

My utworzymy prosty program do przetestowania komunikacji Profibus pomiędzy S7-200 a S7-300, więc zawartość programu umieścimy w bloku OB1.

Okno edytora, które się otworzy wygląda jak poniższym rysunku. Jest ono podzielone na kilka części:

1. **Tablica deklaracji** – jest częścią otwartego bloku, zawiera zadeklarowane zmienne i parametry dla tego bloku programowego (lokalne). W blokach typu FB oraz FC definiujemy tutaj interfejs bloku, czyli jego wejścia i wyjścia.
2. **Elementy programowe** – okno to wyświetla listę instrukcji dla wybranego języka (LAD,FBD,STL). Elementy można wstawiać poprzez przeciąganie do części programowej,
3. **Część programowa** – tutaj znajduje się tworzony program. Okno jest podzielone na segmenty tzw. Networki. Edytor podczas pracy automatycznie sprawdza poprawności składni.



Rys. 45. Elementy edytora LAD/STL/FBD.

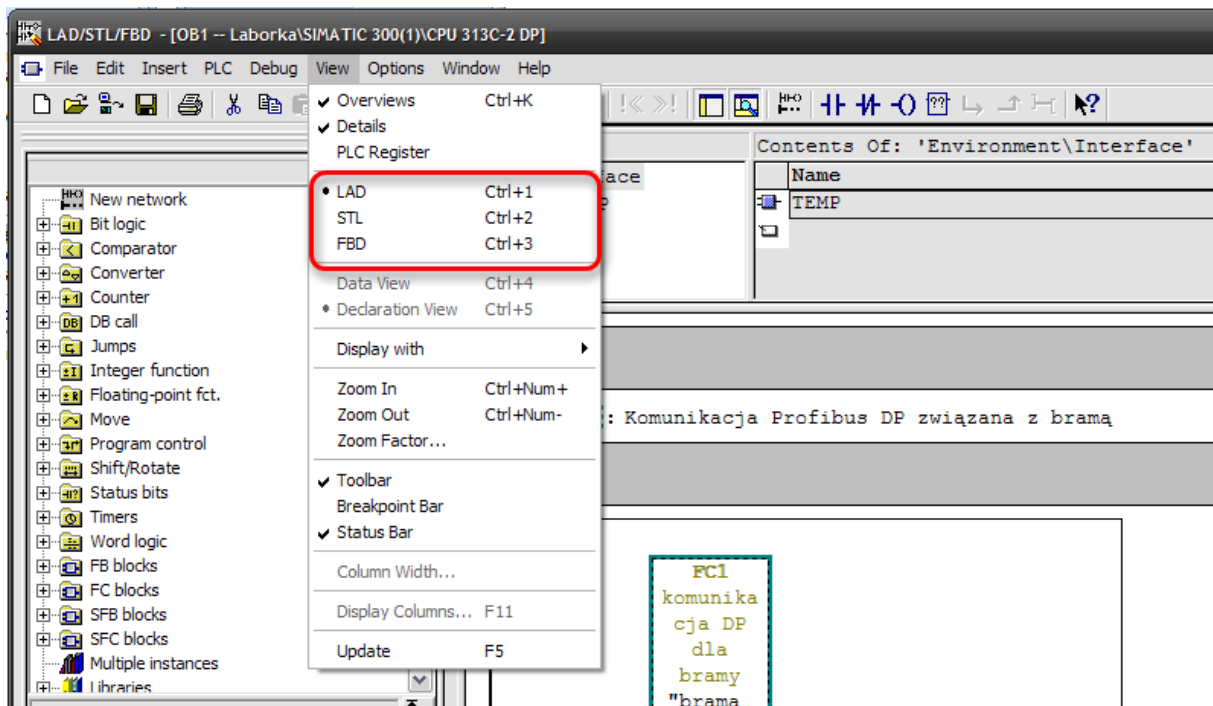
Step 7 posiada kilka języków programowania, które mogą być zastosowane przez użytkownika w zależności od preferencji i wiedzy. Zachowując odpowiednie reguły pisowni, program możemy utworzyć tak, aby możliwe było przełączanie edytora pomiędzy językami i konwersja następowała automatycznie.

Języki obsługiwane w Step7:

- LAD – język drabinkowy przypomina postać schematów stycznikowych. Wykorzystuje on symbole zestyków i cewek,
- STL – lista instrukcji środowiska Step7. Ten język pozwala na największą swobodę w tworzeniu programu. Jest zbliżony do assemblera,
- FBD – plan funkcyjny, wykorzystuje bloczki jako poszczególne funkcje logiczne (np. OR czy AND).

Poprzez menu **View** możliwa jest zmiana aktualnie używanego języka programowania. Podczas zamiany program jest konwertowany według następującego schematu:

- LAD/FBD -> STL – Część programu utworzona w języku graficznym LAD/FBD jest przekształcana na listę poleceń. Tak wykonana konwersja nie zawsze stanowi najbardziej optymalne rozwiązanie w języku STL,
- STL -> LAD/FBD – Program w części nadającej się do konwersji jest zamieniany na docelowy język LAD/FBD. Część, której nie można przekształcić na graficzną pozostaje w STL. Podczas konwersji, żadna część programu nie ulega skasowaniu.



Rys. 46. Wybór języka programowania.

Po zapoznaniu się z oknem edytora, napiszemy program przesyłający przykładowe zmienne bitowe oraz przykładowe zmienne liczbowe pomiędzy sterownikami.

Przykładowy program do testu komunikacji dla S7-300

Utworzymy tutaj najprostszy program przesyłający wartości bitowe (np. rozkaz otwarcia bramy) oraz wartości liczbowe (np. temperatura zmierzona w pomieszczeniu) pomiędzy sterownikami S7-200 i S7-300.

Jak pamiętamy konfiguracja adresów naszej sieci Profibus wygląda następująco:

- Sterownik S7-300:
 - Bajty od IB10 do IB25 (16 kolejnych bajtów) to obszar wejść odbierany ze sterownika S7-200,
 - Bajty od QB10 do QB25 (16 kolejnych bajtów) to obszar wyjść wysyłany do sterownika S7-200
- Sterownik S7-200:
 - Obszar pamięci Variable memory VB100 do VB115 to obszar wejść dla danych przesyłanych z S7-300,
 - Obszar pamięci Variable memory VB116 do VB131 to obszar wyjść dla danych wysyłanych do S7-300

W sterowniku S7-300 obszar pamięci wykorzystywany do komunikacji Profibus ze sterownikiem S7-200 jest widoczny jako standardowe wejścia (np. I10.0) oraz wyjścia (np. Q10.0).

Należy również pamiętać, że zadeklarowany w konfiguracji sposób przesyłanych danych **8 Word Out / 8 Word In** oznacza tylko sposób przesyłania danych między sterownikami (przesyłane są całe słowa). Natomiast w naszym programie sterowania możemy ten obszar adresować dowolnie:

- Jako bit np. I10.0, I10.1, I10.2 itd.
- Bajtowo np. IB10, IB11, IB12 itd,
- Słowa np. IW10, IW12, IW14 itd.,
- Czy podwójnymi słowami ID10, ID14, ID16...

Jak widać należy tak adresować swoje zmienne aby obszary pamięci się nie nakładały. Nie można np. zadeklarować zmiennych IW10 oraz IB11, ponieważ bajt 11 z IB11 nadpisuje bajt 11 ze słowa IW10.

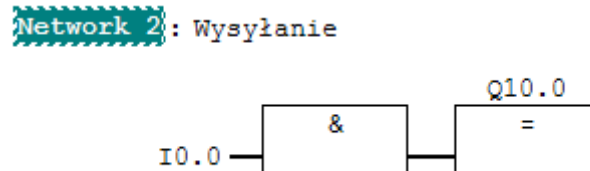
Liczba w adresowaniu zawsze oznacza bajtowo adres bazowy od którego zaczyna się adres naszej zmiennej tj. :

- I10.0 to bit 0 bajtu 10,
- IB10 to cały bajt 10,
- IW10 to całe słowo począwszy od adresu 10, czyli bajty 10 i 11,
- ID10 to podwójne słowo, czyli bajty 10,11,12,13.

Przesyłanie zmiennych binarnych

Aby przesyłać stany zmiennych binarnych wystarczy przepisać wejście (np. I0.0), marker (M10.0) lub inną zmienną binarną np. z Data Blocku (np. DB1.DBX0.1) do odpowiedniego bitu (wyjścia) w obszarze wyjść Profibus.

Poniższy fragment programu przesyła stan wejścia I0.0 do bitu w obszarze wyjść bufora komunikacji Profibus Q10.0.

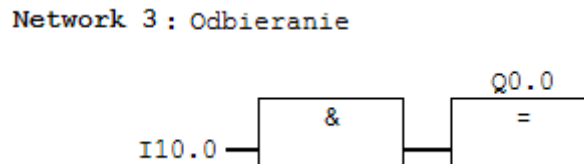


Rys. 47. Przesłanie stanu wejścia do obszaru wyjść Profibus sterownika S7-300.

Tak zapisany bit Q10.0 będzie w naszej konfiguracji przesłany do S7-200 i widoczny w jego obszarze wejść pod adresem **V100.0** (nie adresujemy VB100, ponieważ chcemy odczytać wartość jednego bitu V100.0 a nie całego bajtu VB100).

Analogicznie odbieramy dane z bufora komunikacji przesyłane ze sterownika S7-200.

Z obszaru wejść odczytujemy stan I10.0 i zapisujemy go na wyjście sterownika Q0.0.

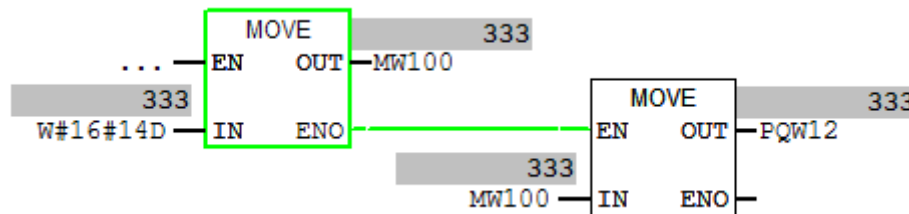


Rys. 48. Odebranie z obszaru wejść stanu zmiennej bitowej sterownika S7-200.

Wpisz do edytora taki przykładowy program odczytujący wejście I0.0 z S7-300 i przesyłający do obszaru wyjścia Profibus Q10.0, następnie przepisz wejście z obszaru Profibus I10.0 na wybrane wyjście sterownika np. Q0.0.

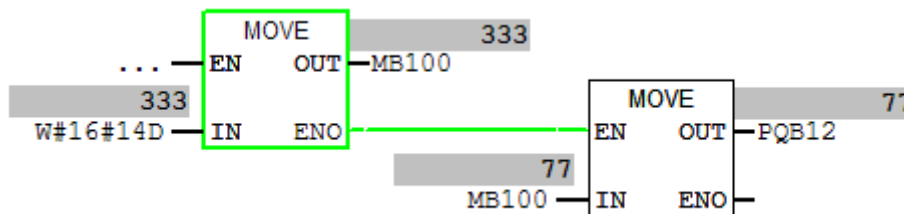
Przesyłanie wartości liczbowych

Podobnie przesyłamy wartości liczbowe. Do przesyłania wartości liczbowych korzystamy z bloczka Move. W naszym przykładzie przesyłamy wartość liczbową 333 do obszaru wyjść na adres PQW12. Przedrostek P przed adresem (QW12) oznacza, że wartość wejścia odczytamy z tzw. obrazu procesu (Process Image), a nie bezpośrednio z wejścia fizycznego. W ten sposób odczytujemy i zapisujemy wejścia/wyjścia analogowe.



Rys. 49. Przesyłanie wartości liczbowej.

Przy przesyłaniu wartości liczbowych (wartość analogowa, wyniki obliczeń) musimy pamiętać aby zmienne, które przesyłamy miały zadeklarowany odpowiedni obszar pamięci (Word, Double Word itp.) dla danego typu danych. Przykładowo próba przesłania liczby 333 jako bajt (szesnastkowo zapisujemy W#16#14D) da efekt jak poniżej.



Rys. 50. Błąd adresowania przy przesyłaniu wartości liczbowej.

Dziękuję się tak dlatego, że bajt może przechować maksymalnie 256 wartości, więc nasza wartość 333 została obcięta ($333 - 256 = 77$) i została przesłana liczba 77.

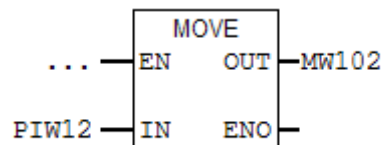
Elementarne typy danych dostępne w Step7 przedstawia poniższa tabela.

Oznaczenie	Długość (bity)	Przykład stałej tego typu
BOOL	1	1 lub 0
BYTE	8	B#16#A9
WORD	16	W#16#12AF
DWORD	32	DW#16#ADAC1EF5
CHAR	8	'w'
S5TIME	16	S5T#5s_200ms
INT	16	123
DINT	32	65539
REAL	32	1.2 lub 34.5E-12
TIME	32	T#2D_1H_3M_45S_12MS
DATE	16	D#1993-02-12
TIME_OF_DAY	32	TOD#12:23:45.12

Tabela 1. Typy danych dostępne z poziomu Step7.

W drugim kroku odbierzemy wartość przesyłaną ze sterownika S7-200, wchodzącą do obszaru wejść np. PIW12 i przepisujemy do pamięci M (markera) np. w słowie MW102.

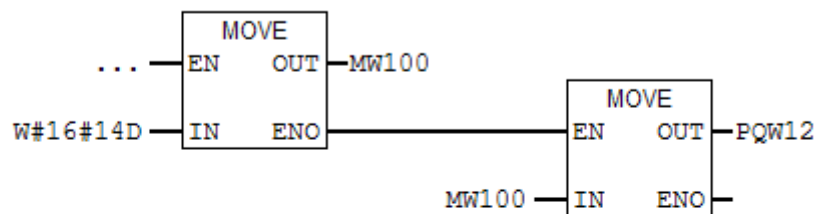
Program realizujący taką funkcję wygląda tak:



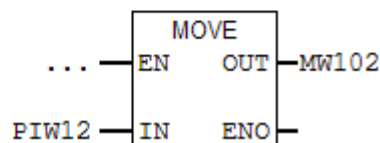
Rys. 51. Odebranie słowa z obszaru wejść Profibus.

Napiszmy więc taki prosty program wysyłający i odbierający wartość bitową oraz wysyłający i odbierający zmienną typu integer (zapisaną jako słowo).

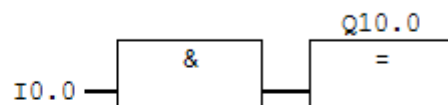
Network 1: Wysyłanie słowa



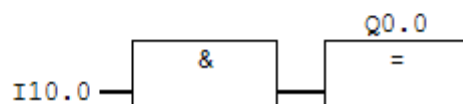
Network 2: Odbieranie słowa



Network 3: Wysyłanie wartości bitowej



Network 4: Odbieranie wartości bitowej



Rys. 52. Przykład prostego programu przesyłającego wartości liczbowe i bitowe.

Wartość, którą odbieramy w zmiennej PIW12 zdefiniujemy w programie S7-200.

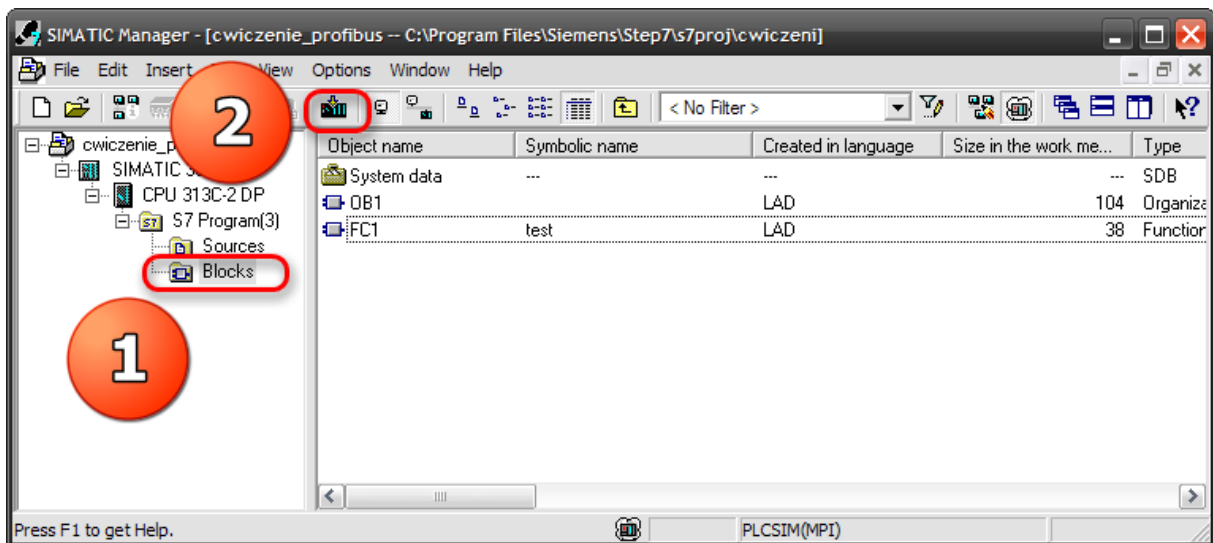
Do podglądania aktualnych wartości w programie wykorzystamy funkcję Monitor. W trakcie pracy z podłączonym sterownikiem, funkcje monitora uruchamiamy z paska narzędzi ikoną okularów.



Rys. 53. Uruchamianie monitora programu.

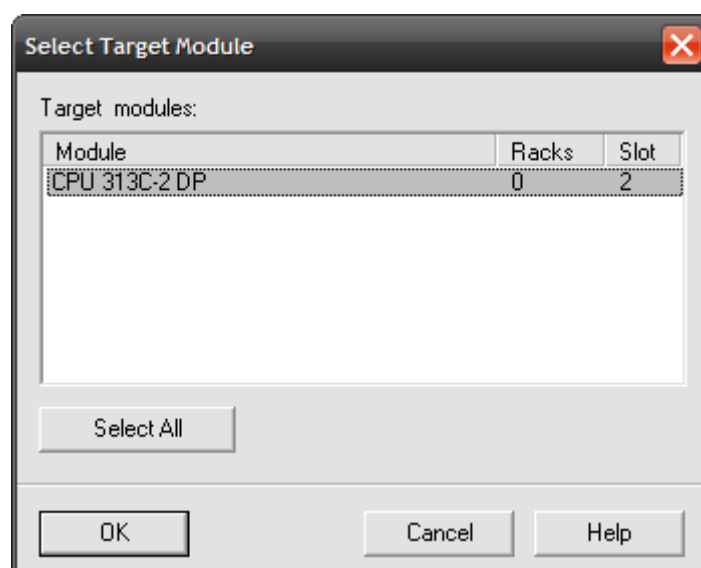
Przesłanie programu do sterownika

Po wprowadzeniu przykładowego programu przesyłamy go wraz z utworzoną wcześniej konfiguracją sprzętową do sterownika. W tym celu zaznaczamy cały katalog Blocks (1) i klikamy przycisk Download (2).



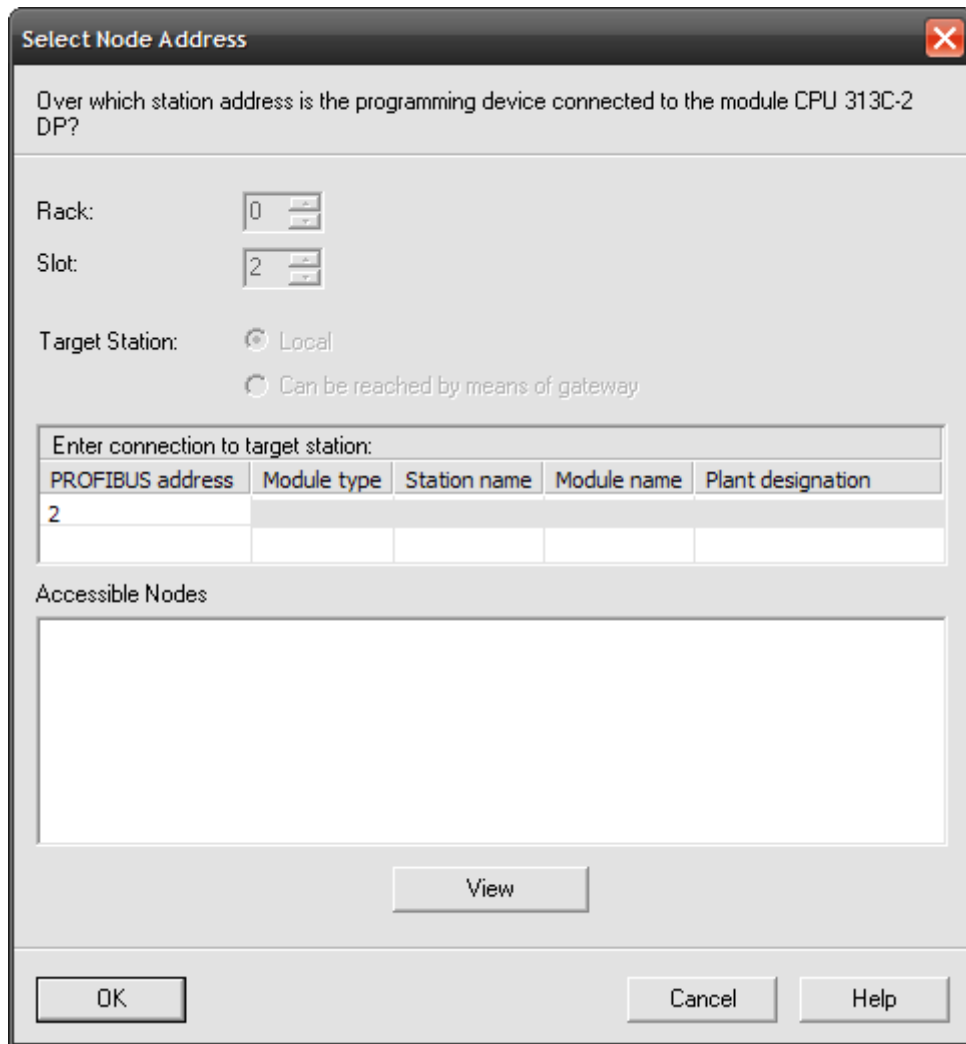
Rys. 54. Przesłanie programu do sterownika.

Wyświetli się okno z dostępnym w projekcie sterownikiem. Zaznaczamy go i klikamy OK.



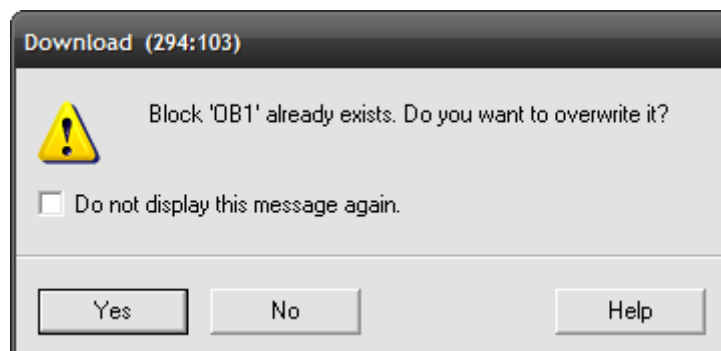
Rys. 55. Wybór sterownika.

Pojawi się okno z listą dostępnych podłączonych sterowników. Klikamy View aby odświeżyć zawartość. Pojawi się nasz sterownik. Zaznaczamy go i klikamy OK.



Rys. 56. Wybór stacji do przesłania programu.

Jeżeli pojawi się komunikat informujący o istnieniu bloku OB1, potwierdzamy nadpisanie.



Rys. 57. Ostrzeżenie o nadpisaniu bloku OB1.

Po wysłaniu programu do sterownika S7-300 trzeba jeszcze napisać analogiczny program zarządzający danymi w S7-200.

Przykładowy program do testu komunikacji Profibus dla S7-200

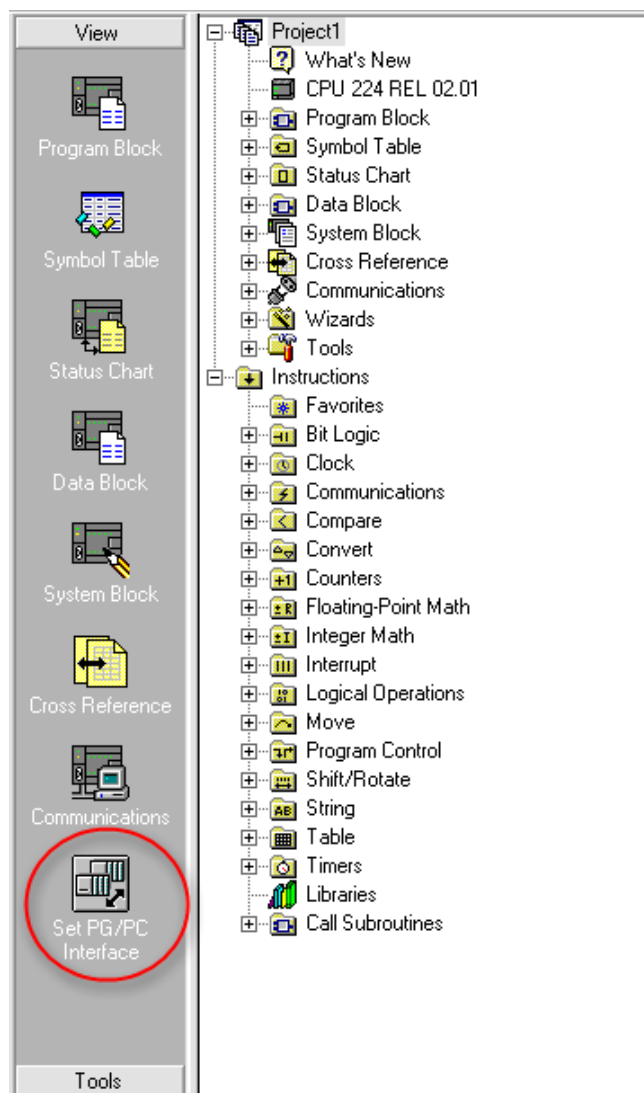
Uruchamiamy program STEP7-Micro/WIN z menu start. Otworzy się okno programu z utworzonym domyślnie nowym projektem. Gdyby jednak z jakiegoś powodu projekt nie został utworzony automatycznie, klikamy ikonę nowego projektu na górnym pasku narzędzi.



Rys. 58. Zakładanie projektu w Step7 Micro/WIN

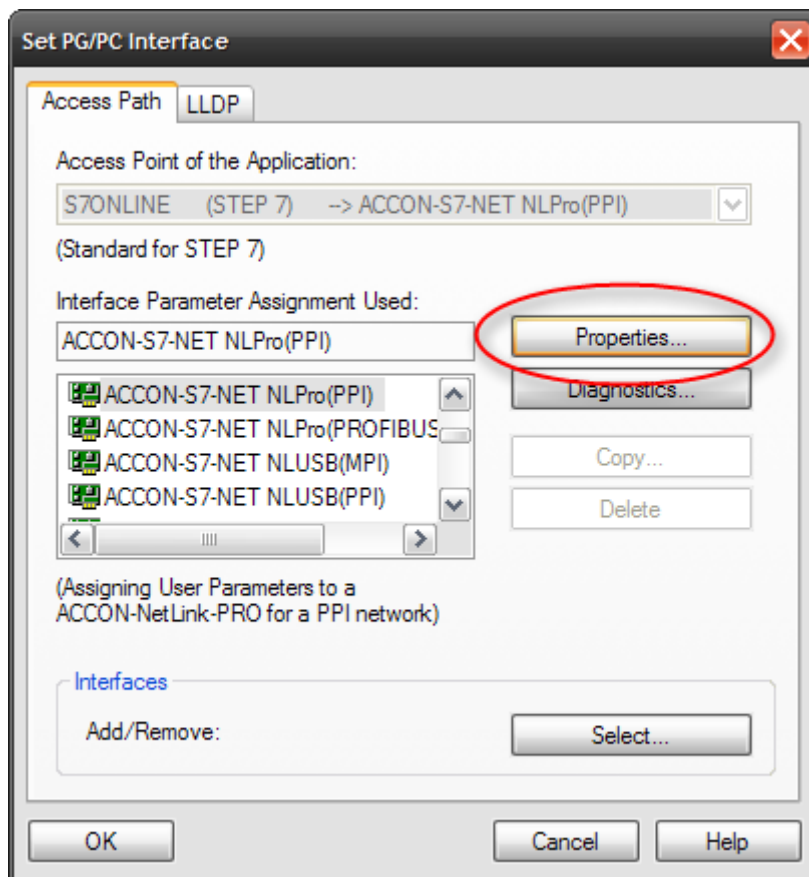
Konfiguracja połączenia komputer – sterownik S7-200

Połączenie PG/PC Interface ze sterownikiem S7-200 konfigurujemy analogicznie jak opisano powyżej dla sterownika S7-300. Aby przejść do okna konfiguracji wybieramy z głównego okna program ikonę Set PG/PC Interface.



Rys. 59. Przejście do ustawień połączenia.

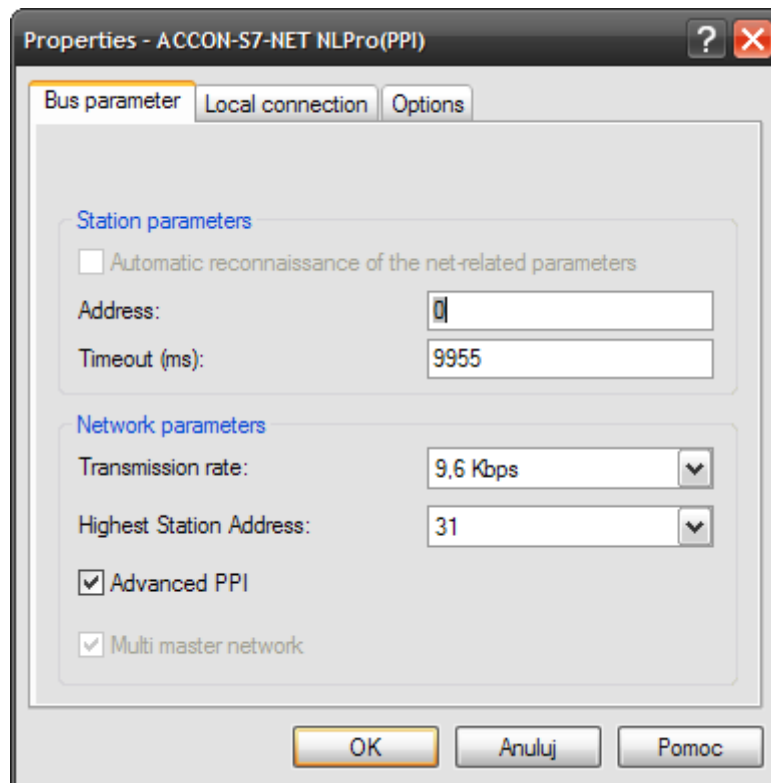
Ze sterownikiem S7-200 będziemy komunikować się za pośrednictwem protokołu PPI, więc wybieramy Accon-S7-Net NLPro (PPI)



Rys. 60. Konfiguracja interfejsu Accon dla trybu PPI.

Klikamy **Properties**, w oknie które się otworzy dokonujemy konfiguracji analogicznie jak przy połączeniu S7-300.

Jedyną różnicą polega na tym, że po dodaniu nowej stacji (10.0.0.1), przechodzimy na zakładkę **Bus parameter**, gdzie określamy prędkość połączenia komputera ze sterownikiem jak na rysunku poniżej.



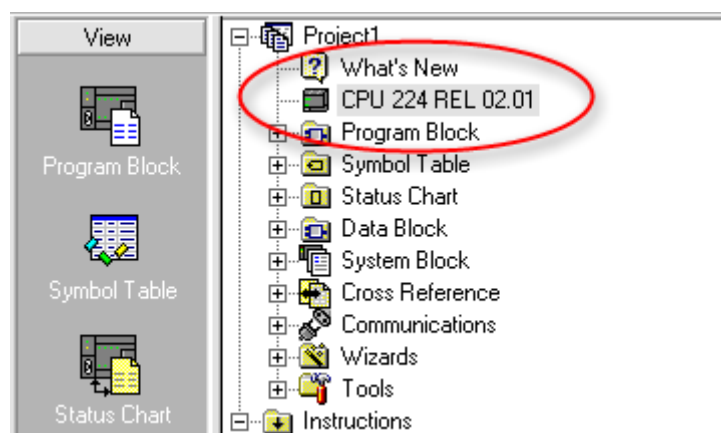
Rys. 61. Wybór prędkości połączenia.

Zamykamy konfigurację PG/PC Interface.

Wybór typu sterownika i przypisanie go do projektu

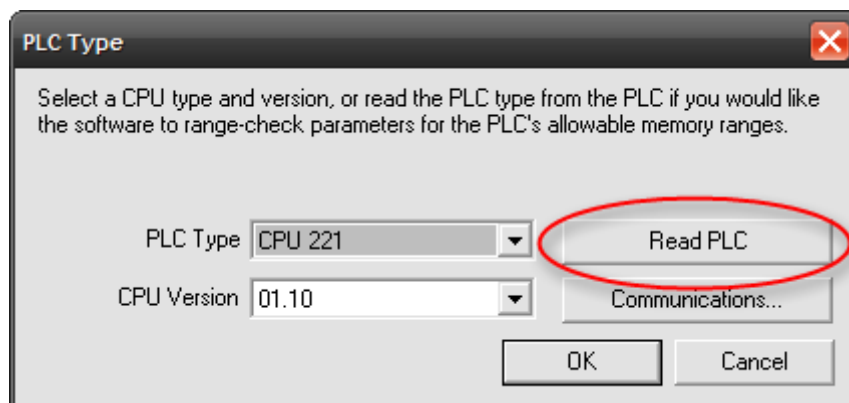
Po skonfigurowaniu połączenia ze sterownikiem przechodzimy do głównego okna programu STEP7-Micro/WIN w celu ustawienia typu sterownika na którym będziemy pracować.

Klikamy dwukrotnie na nazwie sterownika w środkowym oknie edytora projektu.



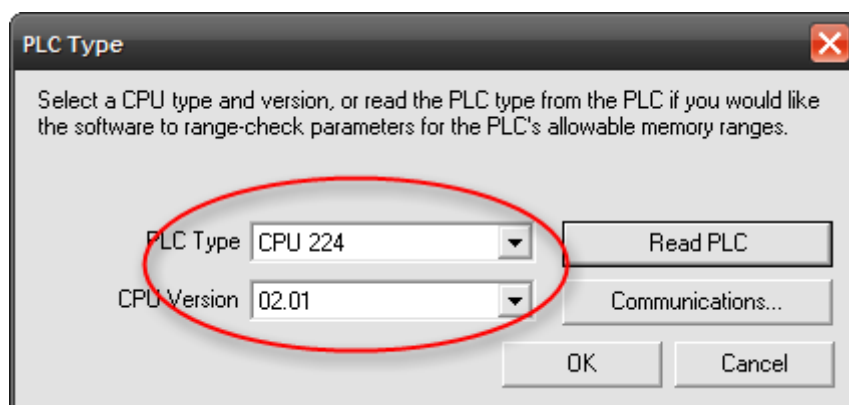
Rys. 62. Wybór typu sterownika w projekcie.

Aby automatycznie odczytać typ i wersję podłączonego sterownika klikamy Read PLC.



Rys. 63. Odczytanie typu podłączonego sterownika.

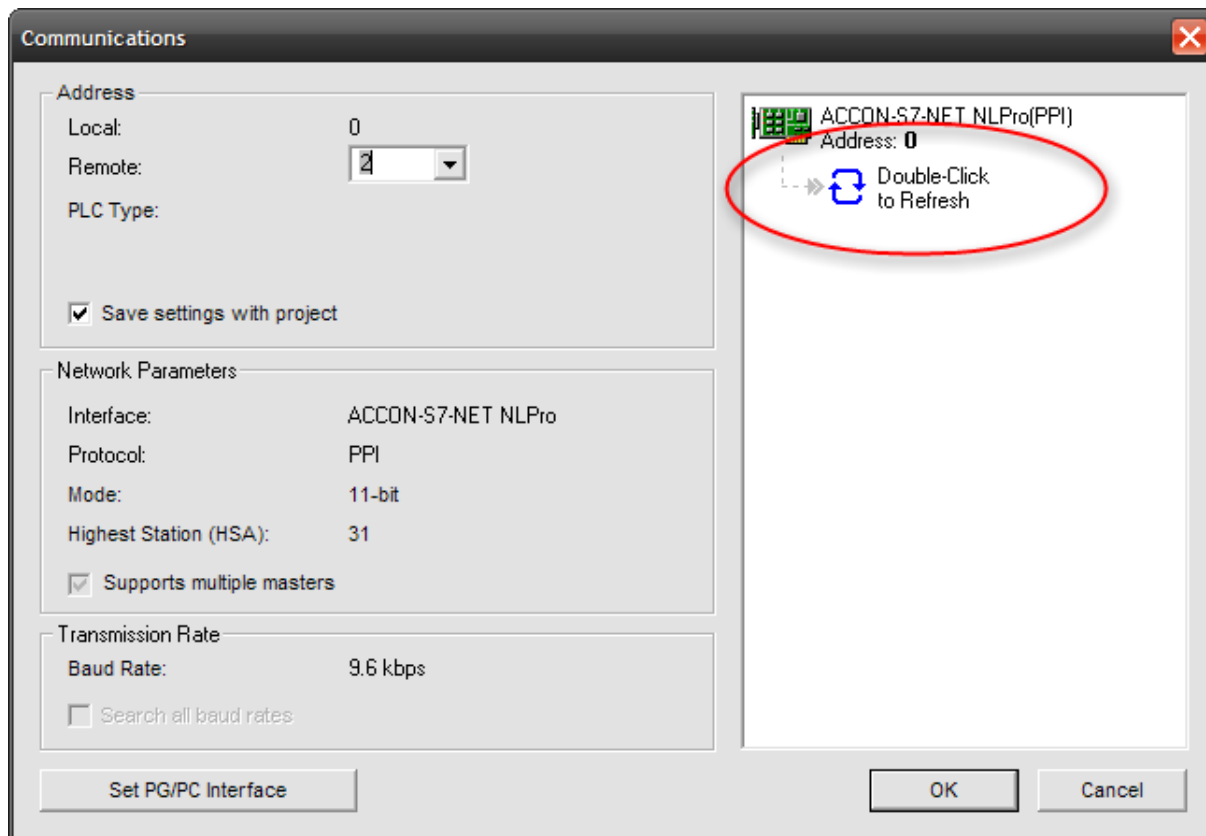
Typ sterownika zostanie odczytany. Klikamy przycisk **Communications**.



Rys. 64. Typ podłączonego sterownika.

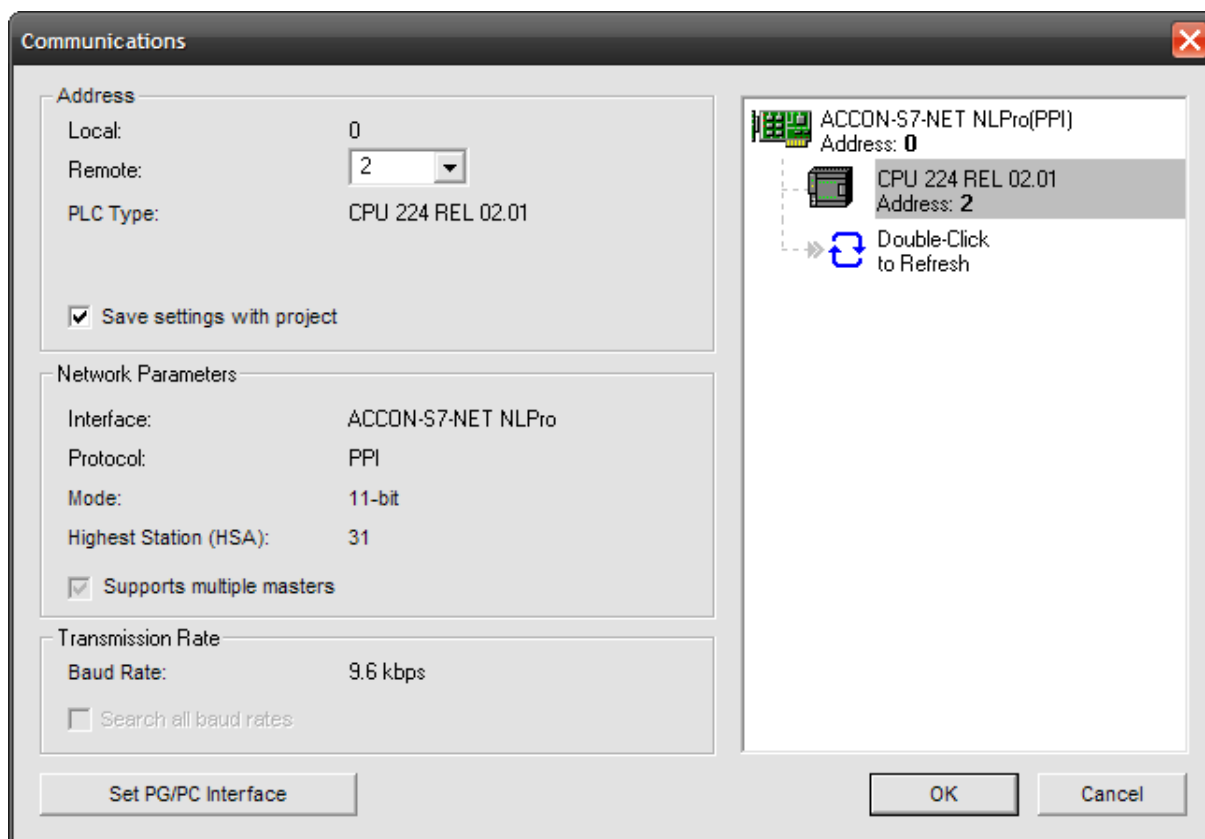
Możemy teraz wyświetlić wszystkie urządzenia dostępne w podłączonej sieci PPI oraz wybrać z wyświetlonej listy sterownik dla którego piszemy program.

Klikamy dwukrotnie na zaznaczonej ikonie strzałek aby odczytać dostępne w sieci sterowniki S7-200.



Rys. 65. Wyszukiwanie podłączonych sterowników.

W sieci znaleziono jeden sterownik dostępny pod adresem 2, wybieramy ten adres z listy Remote oraz zaznaczamy ptaszkiem pole „Save settings with Project” i klikamy **OK**.

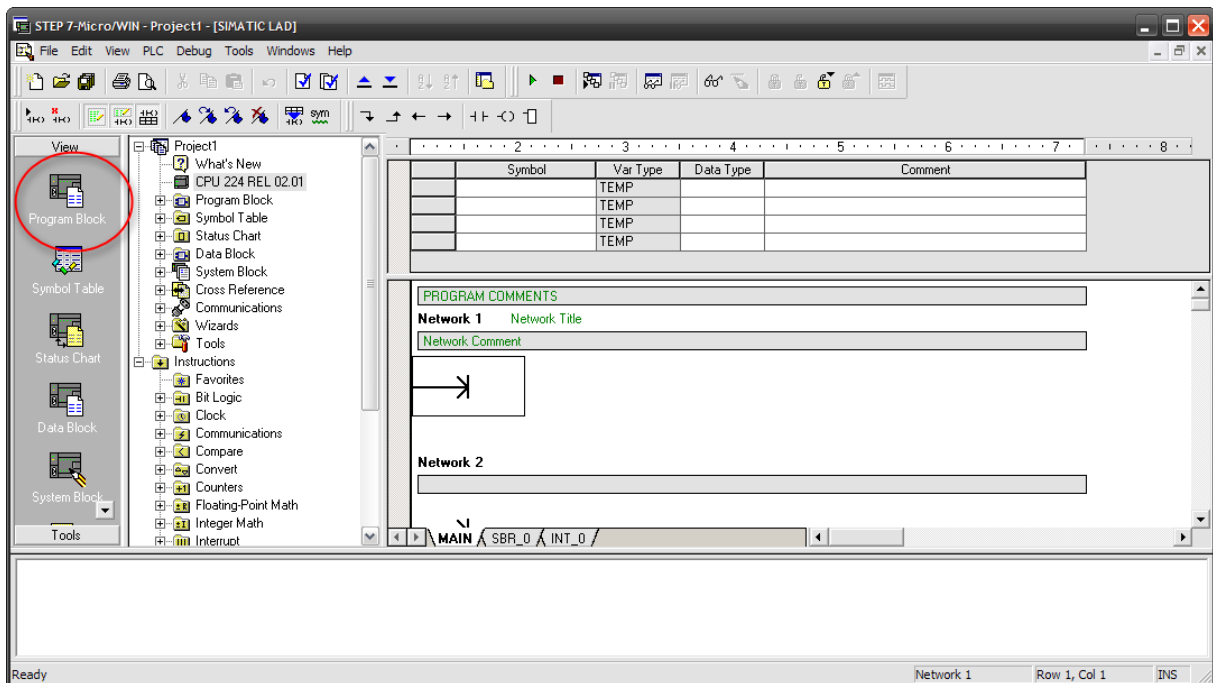


Rys. 66. Znaleziony sterownik S7-200.

Zakończyliśmy konfigurację połączenia ze sterownikiem, wybraliśmy też w projekcie sterownik dla którego piszemy program. Możemy rozpocząć pisanie programu.

Przykładowy program do testu komunikacji

Aby utworzyć przykładowy program przechodzimy do pozycji Program Block w pasku nawigacyjnym programu Step7 Micro/WIN

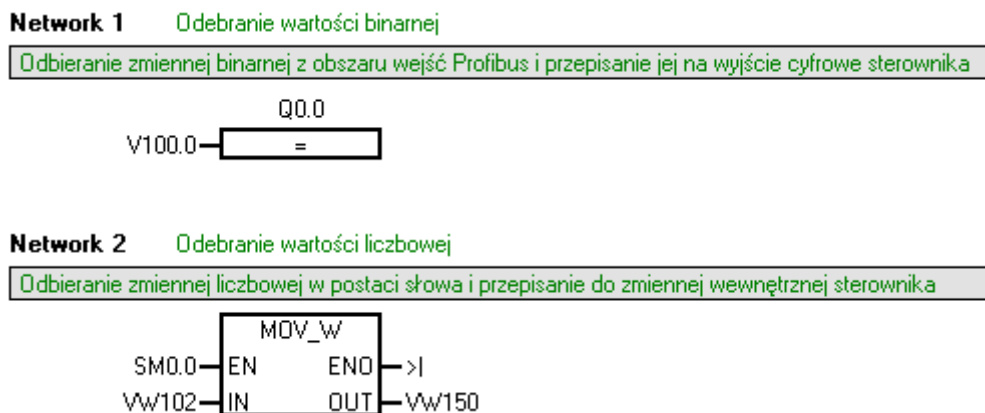


Rys. 67. Okno edytora programu.

Otworzy się okno edytora programu Step7 Micro/WIN podobne do edytora Step7 wykorzystywanego wcześniej. Teraz zachowując przyjęte w naszym projekcie adresy musimy odebrać dane wysłane przez S7-300 oraz wysłać dane do S7-300.

Wartość bitowa wysłana z S7-300 na adres Q10.0 będzie odebrana w sterowniku S7-200 pod adresem V100.0, natomiast wysłane do obszaru wyjść słowo PQW12 będzie zapisane do obszaru wejść sterownika S7-200 pod adresem VW102.

Program odbierający przesłane wartości wygląda więc następująco:



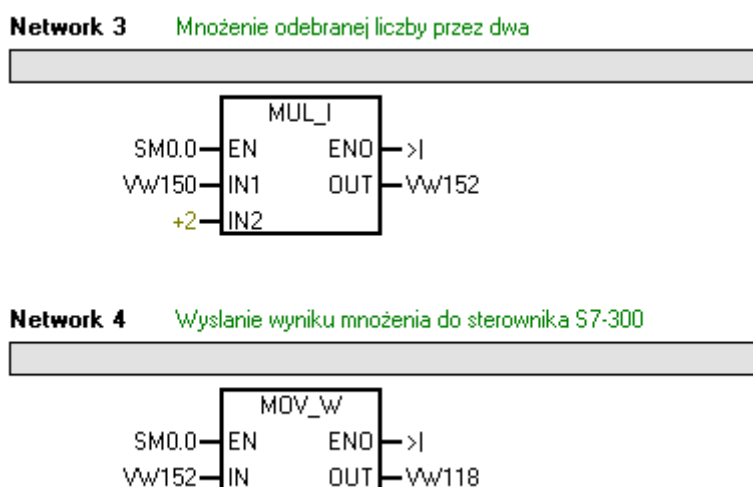
Rys. 68. Odbieranie danych z S7-300.

Bit SM0.0 jest specjalnym bitem (Special Memory), który zawsze posiada wartość 1. Stosujemy go na wejściu EN bloków funkcyjnych, aby były zawsze wykonywane.

W kolejnym kroku wartość liczbową odebraną z obszaru wejść (VW102) przepisujemy na naszą zmienną (VW150), a następnie pomnożymy przez dwa i wyślemy wynik (VW152) z powrotem do S7-300.

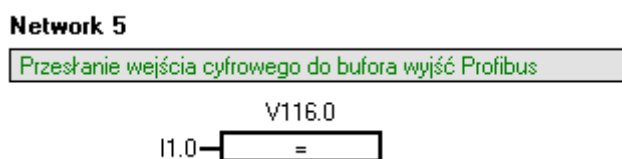
W sterowniku S7-300 odbieramy wartość pod adresem PIW12, więc ze sterownika S7-200 musi ona być wysłana z adresu VW118 (pamiętamy, że obszar wyjść S7-200 to IB116-IB131).

Algorytm realizujący to proste zadanie wygląda następująco:



Rys. 69. Mnożenie i wysłanie z powrotem do S7-300.

Ostatnią czynnością, będzie przesłanie wartości wejścia cyfrowego I1.0 do obszaru wyjść Profibus. Wartość ta jest odbierana w programie S7-300 pod adresem I10.0, więc aby tam dotarła należy ją przesłać do obszaru wyjść na adres V116.0. Instrukcja realizująca to zadanie wygląda tak:



Rys. 70. Przesłanie stanu wejścia cyfrowego do S7-300.

Tak przygotowany program należy przesłać do sterownika S7-200. Program wysyłamy za pomocą opcji Download dostępnej z paska narzędzi.



Rys. 71. Przesłanie programu do sterownika.

Po wgraniu programów do obu sterowników sprawdzamy działanie komunikacji:

- Załącz przełącznik **I0.0 na sterowniku S7-300** – powinna zapalić się dioda sygnalizująca stan wysoki przy wyjściu **Q0.0 sterownika S7-200**,
- Załącz przełącznik **I1.0 na sterowniku S7-200** - powinna zapalić się dioda sygnalizująca stan wysoki przy wyjściu **Q0.0 sterownika S7-300**,

Aby sprawdzić przesyłanie wartości liczbowych, przejdź do programu sterownika S7-300 w Step7 i wybierz ikonę okularków (Rys. 53). W trybie monitorowania można sprawdzić wartości wszystkich zmiennych widocznych na ekranie. Powinna być wysyłana wartość **333** i odbierana wartość **666**.