

Zautomatyzowane Systemy Produkcyjne

Politechnika Poznańska
Katedra Sterowania i Inżynierii Systemów

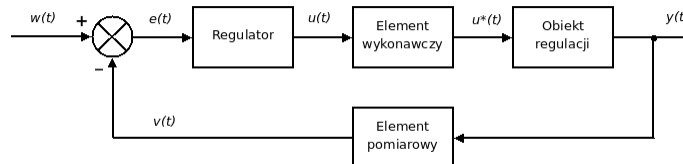
STEROWNIKI PLC

Celem ćwiczenia jest poznanie podstawowych cech sterowników z programowalną logiką (PLC). Podczas ćwiczenia omówione zostaną podstawowe informacje związane z procesem sterowania. Wyjaśnione zostaną budowa i zasada działania sterowników PLC. Poruszona zostanie również będzie tematyka programowania sterowników PLC. Wszystkie zagadnienia teoretyczne zostaną przetestowane w praktyce przy użyciu sterowników PLC firmy Siemens.

1 Wprowadzenie

1.1 Zagadnienie sterowania

Proces sterowania, inaczej regulacji, polega na takim oddziaływaniu na obiekt sterowania, aby jego sygnały wyjściowe osiągnęły pożądaną wartość. Schemat układu regulacji automatycznej, czyli nie wykonywanej ręcznie przez człowieka, jest przedstawiony na Rys. 1.



Rysunek 1: Układ regulacji automatycznej.

Użytkownik określa pewne zadanie do wykonania $w(t)$, nazwany inaczej sygnałem zadany. Stan aktualny układu (inaczej sygnał wyjściowy) jest określony jako $y(t)$. Za pomocą elementu pomiarowego dokonywany jest pomiar stanu aktualnego układu, nazywany sygnałem zwrotnym $v(t)$. Element porównujący oblicza różnicę pomiędzy $w(t)$ a $v(t)$, nazwaną uchybem $e(t)$. Uchyb jest przekazywany do regulatora, który na podstawie **prawa sterowania**, przekształca go w sygnał sterujący $u(t)$ i przekazuje do elementu wykonawczego. Dany sygnał sterujący, skutkuje wywołaniem przez element wykonawczy pewnego oddziaływania $u^*(t)$ na obiekt.

Przykładem układu sterowania może być układ zbiornika z wodą i pompy wprowadzającej lub wyprowadzającej wodę ze zbiornika. Wówczas jako sygnał zadany można przyjąć określony poziom wody w zbiorniku. Sygnałem wyjściowym byłby zatem aktualny stan wody w zbiorniku, a uchybem różnica pomiędzy aktualnym i zadany poziom wody. Regulatorem mógłby być pewien układ elektroniczny, który na podstawie informacji o uchybie określałby wartość sygnału sterującego (i o tej właśnie części układu sterowania będzie traktować opisywane ćwiczenie). Elementem wykonawczym nazwano by pompę, która wpływałaby na poziom wody w zbiorniku, czy też inaczej w obiekcie regulacji. Elementem pomiarowym mógłby być czujnik poziomu wody.

W przemyśle często rozwiązywanie zadań sterowania (zadań wykonywanych przez regulator) odbywa się przy wykorzystaniu metody stykowej, a więc odpowiedniej konfiguracji podłączenia styków przekaźników czy styczników. Metoda ta opiera się na algebrze Boole'a, czy inaczej na logice binarnej (dwuargumentowej). Wyróżniamy

Lp.	Równanie	Układ połączeń	Lp.	Równanie	Układ połączeń	Lp.	Równanie	Układ połączeń
1	$1 + a = 1$		3	$0 + a = a$		5	$\neg a + a \cdot b = \neg a + b$	
2	$a \cdot \neg a = 0$		4	$a + \neg a \cdot b = a + b$		6	$a + a \cdot b = a$	
„¬” – operator negacji „/a” oznacza negację argumentu a, podobnie jak „~a”								

Rysunek 2: Równania algebry Boole'a wraz z układami połączeń.

tu dwa stabilne stany styku: stan działania jako logiczna „1” oraz stan niedziałania jako logiczne „0”. Wybrane przykłady równania algebry Boole'a i odpowiadające im układy połączeń przedstawiono na Rys. 2.

Jednak metody stykowe posiadają liczne wady, m. in.:

- trudna diagnostyka układu sterowania, identyfikacja błędów czasem niemożliwa;
- konieczność stosowania kosztownego sprzętu diagnostycznego w celu kontroli wykonywanych połączeń
- pracochłonny demontaż połączeń elektrycznych w celu naprawy;
- sztywność algorytmu sterowania, uniemożliwiająca elastyczne wybranie innego algorytmu;
- brak możliwości realizacji sterowania nadrzędnego w przypadku konieczności konstrukcji systemów rozproszonych.

Alternatywą dla metody stykowej jest zastosowanie sterowników PLC.

1.2 Sterownik PLC - definicja

Sterowniki PLC (sterowniki programowalne logicznie) to urządzenia mikroprocesorowe przeznaczone do sterowania procesami przemysłowymi. Realizują zaprogramowany algorytm sterowania w czasie rzeczywistym. Podstawowym powodem zastosowania jest możliwość wbudowania logiki działania układu sterowania procesem do pamięci urządzenia, najczęściej o dostępie swobodnym, co umożliwia wymianę programu sterującego.

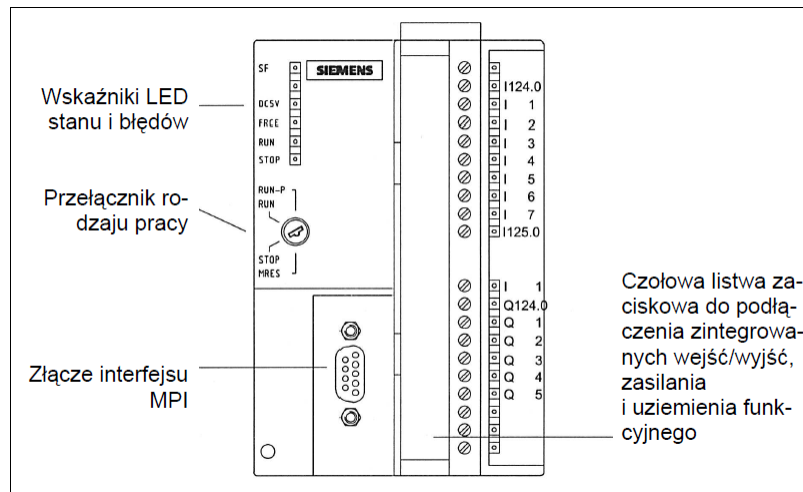
Najpopularniejsze sterowniki PLC to, m.in.:

- SIMATIC S7-200, S7-300, S7-400,
- B&R 2003, B&R 2005,
- SAIA PCD2, PCD4,
- Fanuc 90-70, 90-30,
- Schneider Modicon Quantum.

Sterowniki PLC posiadają budowę monolityczną lub modułową (konfigurowalną). Podczas opisywanych ćwiczeń, prace będą prowadzone na sterownikach modułowych SIMATIC S-7 300 CPU 312 IFM.

Koncepcja modułowa pozwala użytkownikowi wybrać i zestawić na płycie bazowej dowolną kombinację jednostek centralnych CPU, urządzeń komunikacyjnych, specjalistycznych modułów sterujących i dyskretnych wejść i wyjść.

Budowa jednostki centralnej jest przedstawiona na Rys. 3.



Rysunek 3: Budowa SIMATIC S-7 300 CPU 312 IFM.

Zatem sterownik PLC jest identyfikowany jako regulator z Rys. 1. Jego zadaniem jest obliczanie odpowiednich wartości sygnałów wysyłanych na urządzenia wykonawcze, na bazie informacji dostarczanych przez urządzenia pomiarowe. Zapisany na sterowniku program pozwala na obliczanie stanu wyjść na podstawie stanu wejść.

Elementy wejściowe sterownika PLC:

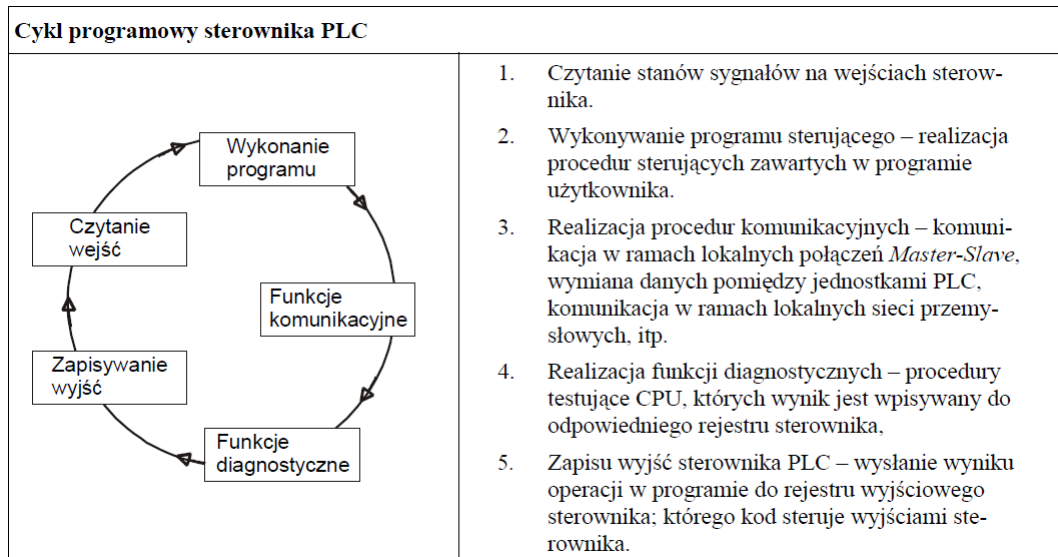
- generują sygnały dla układu sterowania,
- dostarczają niezbędną informację o danych procesu, które są podstawą do realizacji zadanego algorytmu,
- muszą generować sygnały analogowe i cyfrowe odpowiednio dopasowane do wejść sterownika PLC,
- stosuje się specjalizowane dopasowujące moduły sterownika, np. moduł wejść enkodera,
- przykłady: wyspa zaworowa, czujnik poziomu wody, czujnik pozycji kątowej (enkoder), krańcówka, przycisk, czujnik indukcyjny, czujnik termistorowy.

Elementy wyjściowe sterownika PLC:

- urządzenia wykonawcze, które pod wpływem sygnałów z układu sterowania zmieniają swój stan pracy,
- wyposaża się w układy pośredniczące w sterowaniu tymi elementami,
- możliwe jest wyposażenie sterownika w odpowiedni moduł, np. do sterowania silnikiem krokowym,
- przykłady: silnik prądu stałego, silnik asynchroniczny, siłownik pneumatyczny, pompa, lampka, stycznik, przekaźnik.

1.3 Zasada działania sterownika PLC

Zasada działania sterownika logicznego PLC polega na pracy cyklicznej, w której sterownik wykonuje pojedyncze rozkazy programu w takiej kolejności, w jakiej są one zapisane w programie.



1.4 Programowanie sterownika PLC

Przed rozpoczęciem procesu programowania należy wszystkim zmiennym w programie nadać odpowiednie adresy. Jest to niezbędna informacja dla sterownika - dzięki niej może utożsamić wirtualną zmienną programową z rzeczywistym sygnałem podłączonym do sterownika. Każdy adres zaczyna się od operandu określającego typ zmiennej. Dla elementu wejściowego korzysta się z oznaczenia *I*, elementowi wyjściowemu przypisany jest *Q*, natomiast element pamięci (pomocnicza zmienna) opisany jest za pomocą *M*. Po operandzie umieszcza się numer modułu, do którego podłączony jest dany element i po kropce numer zacisku. Każdy moduł posiada zawsze 8 zacisków (adresy 0-7). W przypadku wejść i wyjść numer modułu i zacisku należy odczytać bezpośrednio ze sterownika. W przypadku zmiennych pomocniczych można obrać wartości dowolne. Należy jednak pamiętać, że dana zmienna pomocnicza ma zarezerwowane określone miejsce w pamięci, dlatego różnym zmiennym należy nadawać różne adresy. Przykładowe adresy zmiennych przedstawiono w tabeli 1.

Adres	Znaczenie
I 1.3	czujnik podłączony do modułu 1 i zacisku 3
Q 124.0	silnik podłączony do modułu 124 i zacisku 0
M 0.0	zmienna pomocnicza

Tablica 1: Przykładowe adresy zmiennych.




Wyróżnia się cztery podstawowe metody zapisu programu dla sterownika PLC:

- Języki graficzne:
 - metoda drabinkowa LAD (ang. Ladder Diagram) – schemat drabinkowy,
 - język FBD (ang. Function Block Diagram) – funkcjonalny schemat blokowy,
- Języki tekstowe:
 - proceduralny STL (ang. Statement List) – tekst strukturalny,

- assemblerowy IL (ang. Instruction List) – lista rozkazów.

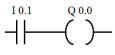
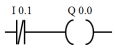
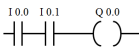
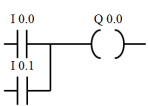
Podczas opisywanych ćwiczeń stosowana będzie jedynie metoda drabinkowa LAD. Metoda ta opiera się na tworzeniu schematu połączeń w postaci szczebli drabiny logicznej. Wykorzystuje się w niej symbole graficzne elementów stykowych i bloków, umieszczanych w odpowiednich miejscach szkieletu schematu. Bardzo powszechnie stosuje się połączenia symboli graficznych tak, aby możliwe było tworzenie np. funkcji sumy logicznej czy iloczynu logicznego.

Za pomocą określonych wartości wejść można ustalać wartości wyjść. Zmienne wejściowe utożsamiane są ze stykami, natomiast wyjścia z cewkami przekaźników (Tab. 2).

Postać LAD	Opis
	styk normalnie otwarty (wejście)
	styk normalnie zamknięty (wejście)
	cewka (wyjście)

Tablica 2: Podstawowe elementy języka LAD.

Wykonywanie programu polega na "przepływie prądu" z lewej strony do prawej. Tu również wykorzystywana jest algebra Boole'a, a więc zmienne mogą przyjmować jedynie logiczne "0" lub "1". Przykładowe wartości wejść i odpowiadające im wartości wyjść są przedstawione w tabeli. 3.

Postać szczebla	Wartości wejść	Wartości wyjść	Komentarz
	I 0.1 = 0	Q 0.0 = 0	Ładuj I 0.1
	I 0.1 = 1	Q 0.0 = 1	
	I 0.1 = 0	Q 0.0 = 1	Ładuj z negacją I 0.1
	I 0.1 = 1	Q 0.0 = 0	
	I 0.0 = 0; I 0.1 = 0	Q 0.0 = 0	Iloczyn logiczny (AND)
	I 0.0 = 1; I 0.1 = 0	Q 0.0 = 0	
	I 0.0 = 0; I 0.1 = 1	Q 0.0 = 0	
	I 0.0 = 1; I 0.1 = 1	Q 0.0 = 1	
	I 0.0 = 0; I 0.1 = 0	Q 0.0 = 0	Suma logiczna (OR)
	I 0.0 = 1; I 0.1 = 0	Q 0.0 = 1	
	I 0.0 = 0; I 0.1 = 1	Q 0.0 = 1	
	I 0.0 = 1; I 0.1 = 1	Q 0.0 = 1	

Tablica 3: Podstawowe zależności w języku LAD.

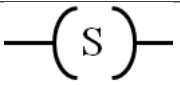

Kolejność wykonywania odbywa się od góry na dół (z wyjątkiem gdy kolejność ta ulega zmianie z powodu wprowadzenia elementów kontrolnych), oznacza to, że kolejność umieszczanych szczebli ma znaczenie. I tak na przykład wartość żadnego elementu nie powinna być wyznaczana dopóki nie wyznaczono wartości wszystkich jego wejść oraz wyznaczanie wartości elementu obwodu nie może być zakończone dopóki nie wyznaczono wartości dla wszystkich jego wyjść.

Szczebel drabiny logicznej (oznaczany w programie jako Network) musi posiadać odpowiedni format i składnię:

- każdy szczebel może zawierać do 16 linii równoległych każda po 16 elementów logicznych połączonych szeregowo,
- ostatnim elementem szeregowego połączenia w danym szczeblu musi być jedna z cewek lub blok funkcyjny,

- szczebel może zawierać maksymalnie do 16 cewek,
- szczebel musi zawierać przynajmniej jeden styk przed wystąpieniem cewki, bloku funkcyjnego lub połączenia pionowego,
- nie może wystąpić rozgałęzienie mające początek lub koniec wewnątrz innego rozgałęzienia.

Oprócz zwykłych cewek możliwe jest również zastosowanie cewek *Set* i *Reset* (Tab. 4).

Postać LAD	Opis
	Cewka <i>Set</i> : ustawianie bitu
	Cewka <i>Reset</i> : usuwanie bitu

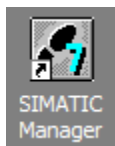
Tablica 4: Cewka Set i Reset.

Cewka *Set* powoduje, że raz ustawiona wartość logicznej “1” zostaje zapamiętana, mimo zmiany wartości styków poprzedzających cewkę. Aby zmienić wartość zmiennej z powrotem na “0” logiczne, należy zastosować cewkę *Reset*, oczywiście przy odpowiedniej konfiguracji styków poprzedzających.

2 Przebieg ćwiczenia

Podczas zajęć z przedmiotu Zautomatyzowane Systemy Produkcyjne, do programowania sterowników PLC wykorzystane będzie środowisko STEP7. Jest to podstawowy pakiet programowy SIMATIC służący do tworzenia programów sterowników SIMATIC S7-300/400 w LAD, FDB lub ST.

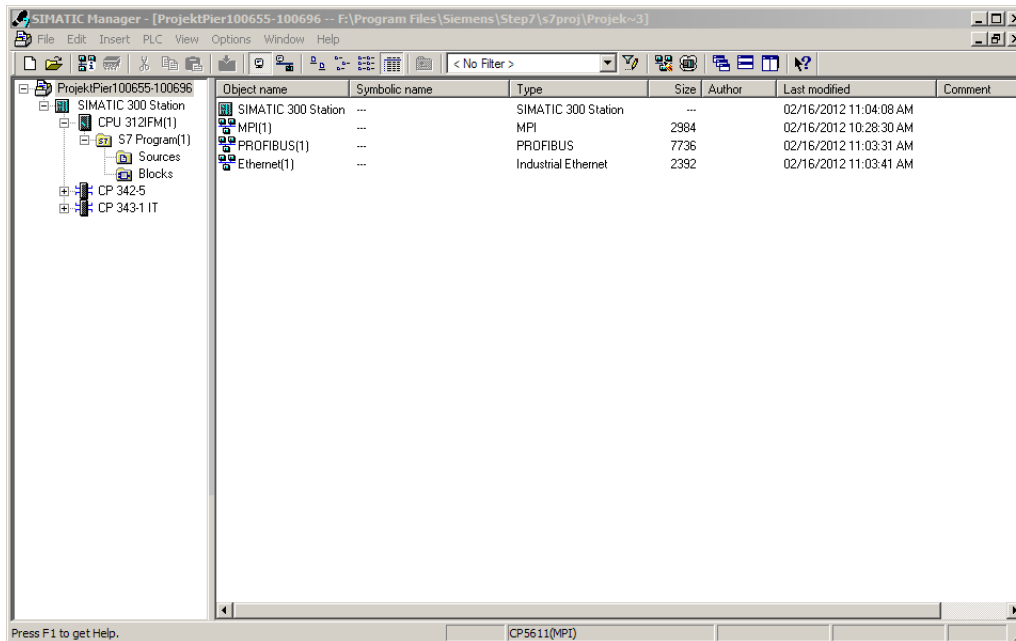
2.1 Utworzenie nowego projektu i konfiguracja sprzętowa



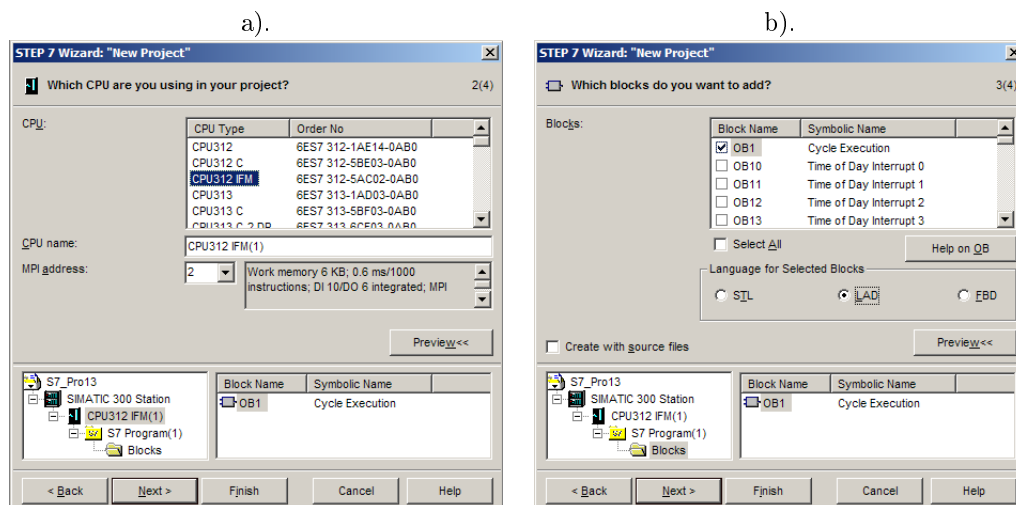
Rysunek 4: Ikona środowiska STEP7.

Aby uruchomić środowisko STEP7 należy wybrać ikonę przedstawioną na Rys. 4. Spowoduje to otwarcie okna programu (Rys. 5). Jeżeli w oknie programu otwarte są jakieś projekty należy je pozamykać. W tym celu należy wybrać *File->Close* z menu głównego programu.

Następnie należy wybrać z menu głównego programu opcję: *File->'New Project' Wizard*. Przechodząc przez kroki kreatora, należy skorzystać ze wskazówek przedstawionych na Rys. 6 - 7.

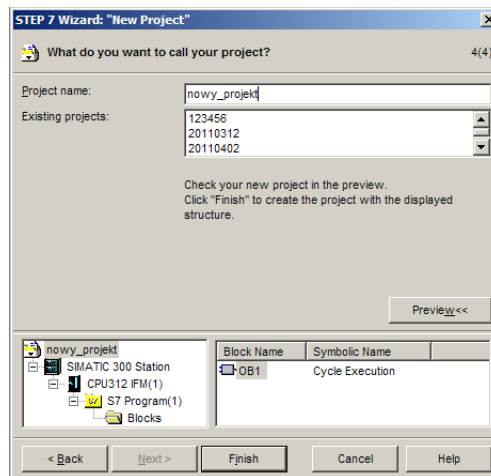


Rysunek 5: Okno środowiska STEP7.

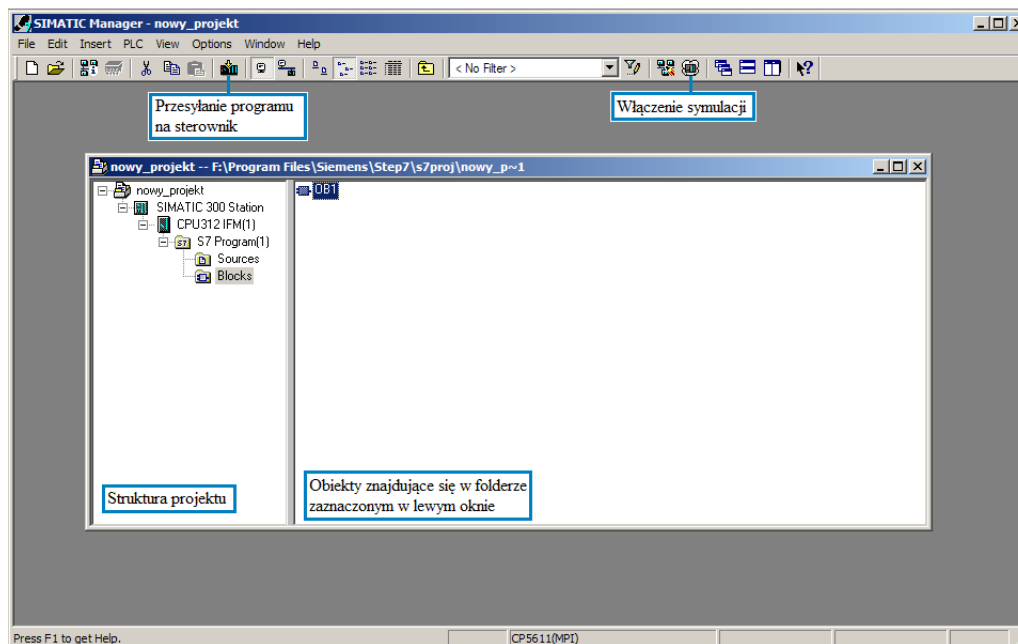


Rysunek 6: Tworzenie nowego projektu: a). wybór typu jednostki centralnej sterownika, b). wybór bloku danych do edycji.

Po wykonaniu powyższych kroków w oknie programu powinien pojawić się utworzony projekt, jak na Rys. 8.



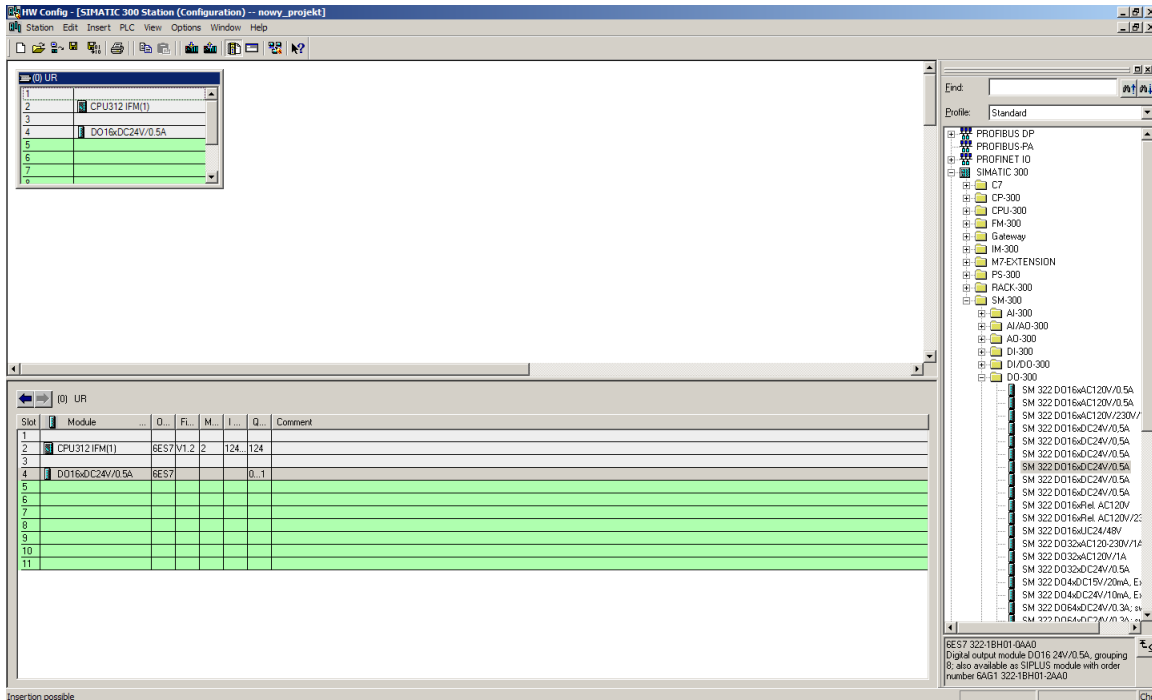
Rysunek 7: Tworzenie nowego projektu: nadanie wybranej nazwy projektu.



Rysunek 8: Okno programu STEP7 z nowym projektem.

W kolejnym etapie ćwiczenia należy zdefiniować w środowisku STEP7 jakie komponenty wchodzi w skład sterownika PLC, z którym nastąpi połączenie. Wystarczy opisać jakie moduły wejść/wyjść są podłączone do jednostki centralnej. Aby dokonać konfiguracji sprzętowej, należy w lewym oknie projektu zaznaczyć opcję *SIMATIC 300 Station*, po czym należy wybrać opcję *Hardware* w prawym oknie.

W wyniku powyższej operacji w programie zostanie otwarte nowe okno *HW Config*. (Rys. 9). W pierwszej kolejności należy upewnić się, czy wybrana jednostka centralna jest identyczna z zastosowaną w sterowniku. W tym celu, należy porównać nazwę i *Order number* z danymi umieszczonymi wprost na jednostce centralnej sterownika.



Rysunek 9: Okno programu STEP7 służące do zdefiniowania konfiguracji sprzętowej.

Aby dodać odpowiedni moduł wejść lub wyjść należy odnaleźć go na liście elementów w prawej części okna. Wybór komponentu odbywa się przez odnalezienie modułu o tej samej nazwie i *Order number* co podane na module sterownika. Elementy należy wyszukiwać w zakładce *Simatic300->SM-300->DO300*. Należy zaznaczyć wybrany element i przeciągnąć go do lewej części okna, pamiętając o tym, że *Slot 3* powinien pozostać pusty. Po dodaniu wszystkich potrzebnych elementów należy zapisać konfigurację oraz zapisać na kartce adresy przypisane wejściom i wyjściom (kolumny *I* oraz *Q*).

Po ukończeniu wszystkich czynności, należy zamknąć okno i powrócić do głównego okna programu (Rys. 8).

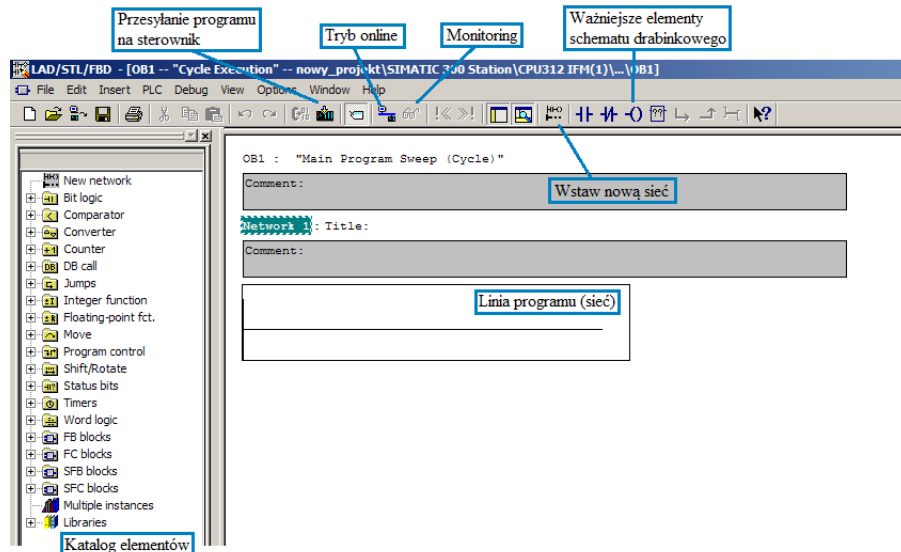
2.2 Pierwszy program

Na tym etapie ćwiczenia utworzony zostanie prosty program, którego działanie będzie przetestowane symulacyjnie oraz na sterowniku.

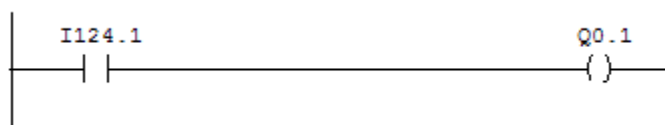
Aby rozpocząć edycję programu należy wybrać opcję *Blocks->OB1*, wówczas pojawi się nowe okno programu (Rys. 10).

Po zaznaczeniu linii (*Network*) możliwe jest umieszczanie na niej elementów. Jest to możliwe przy wykorzystaniu bibliotek elementów umieszczonych po lewej stronie okna. Jeżeli wykorzystane będą jedynie elementy podstawowe, można posłużyć się ikonami w menu podręcznym (Rys. 10). Na linii należy umieścić jeden styk normalnie otwarty oraz cewkę. Stykowi należy nadać adres $Ix.0$, przy czym numer x bajtu powinien być zgodny z adresem odczytanymi z konfiguracji sprzętowej. Odpowiednio cewce należy nadać adres $Qx.0$, przy czym numer x bajtu powinien być zgodny z adresem odczytanymi z konfiguracji sprzętowej. Po wykonaniu wszystkich czynności należy zapisać program. Tak utworzona sieć powinna prezentować się tak jak na Rys. 11 (z uwzględnieniem odpowiednich adresów).

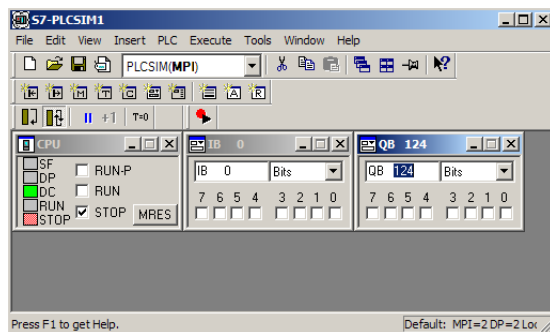
Aby przetestować działanie programu przed przegraniem go na sterownik, należy uruchomić symulację programową. W tym celu należy wybrać ikonę symulacji w oknie głównym projektu (Rys. 8). Spowoduje to otwarcie okna symulacji jak na rysunku 12.



Rysunek 10: Okno programu STEP7 z nowym programem w języku LAD.



Rysunek 11: Sieć przykładowego programu.



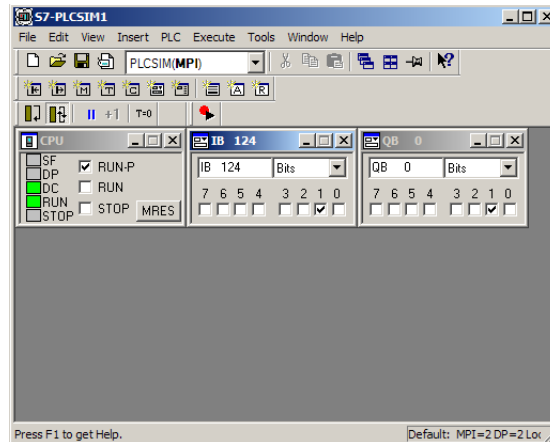
Rysunek 12: Okno symulacji programu STEP7.

W oknie symulacji należy umieścić (opcja *Insert* w menu głównym) okno odpowiadające wejściom (*IB*) i okno odpowiadające wyjściom (*QB*), o ile takie okna już nie znajdują się w symulacji. Wejściom i wyjściom należy przypisać takie adresy jak wcześniej umieszczono w programie (Rys. 11). Zaznaczenie poszczególnych pól w oknie wejść spowoduje przypisanie danego adresowi "1" logicznej.

Po skonfigurowaniu okna symulacji powinno się przesłać program na wirtualny sterownik. W tym celu należy wybrać przycisk *Download* w menu podręcznym okna głównego projektu (Rys.8) lub okna programu (Rys. 10)¹. Po wgraniu programu na sterownik można uruchomić symulację. W oknie symulacji (Rys. 12) należy wybrać

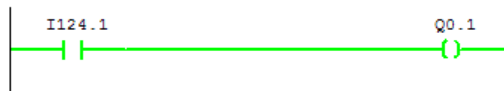
¹W przypadku wybrania przycisku *Download* przed otwarciem okna symulacji spowoduje przesłanie programu bezpośrednio na rzeczywisty sterownik

opcję *RUN-P*. Od tego momentu symulacja jest uruchomiona. Aby przetestować działanie programu należy zaznaczyć odpowiedni adres wejścia i sprawdzić czy zmienia się wartość odpowiedniego wyjścia, jak na rysunku. Nie należy samodzielnie zmieniać stanu wyjść.



Rysunek 13: Symulacja w programie STEP7.

Działanie programu można również obserwować bezpośrednio na schemacie sieci programu, przez wybranie opcji *Monitoring* w oknie programu (Rys. 10). Wówczas styki i cewki, które przyjmują stany logiczne "1" są podświetlone na zielono, jak na rysunku 14.



Rysunek 14: Symulacja programu na schemacie sieci.

Po poprawnym wykonaniu symulacji należy przeanalizować działanie programu na sterowniku. **Tę część ćwiczenia należy wykonać razem z prowadzącym ćwiczenia.**

2.3 Symulacja działania podnośnika A

W tej części ćwiczenia należy stworzyć program symulujący pracę podnośnika. Układ podnośnika składa się z następujących elementów:

- silnika, który realizuje dwa rodzaje ruchów: podnoszenie i opuszczanie ładunku;
- 2 przycisków mono-stabilnych: jeden odpowiada za wymuszenie podnoszenia (G), a drugi za opuszczanie ładunku (D);
- 2 czujników wykrywających obecność ładunku na danej wysokości (N - umieszczony niżej, W - umieszczony wyżej).

W opisywanym programie należy odpowiednio sterować ruchem silnika. Można założyć, że określone są dwa wyjścia układu:

- jedno (OUT1) przyjmuje logiczną "1", jeżeli silnik wykonuje podnoszenie, natomiast "0" gdy silnik nie wykonuje podnoszenia;
- drugie (OUT2) przyjmuje logiczną "1", jeżeli silnik wykonuje opuszczanie, natomiast "0" gdy silnik nie wykonuje opuszczania.

Silnik zaczyna podnosić ładunek ($OUT1 = 1$) jeżeli przycisk G jest naciskany ($G = 1$) i jeżeli ładunek nie jest na wysokości czujnika W ($W = 0$). Silnik zaczyna opuszczać ładunek ($OUT2 = 1$) jeżeli przycisk D jest naciskany ($D = 1$) i jeżeli czujnik N nie wykrywa ładunku na swojej wysokości ($N = 0$). W pozostałych wypadkach silnik się nie porusza ($OUT1 = 0$, $OUT2 = 0$). Założono, że zawsze może być naciskany tylko jeden z dwóch przycisków, a nigdy oba naraz.

Podczas tworzenia programu należy odpowiednio zdefiniować wejścia systemu. Po napisaniu programu należy przeprowadzić symulację bez użycia sterownika oraz przetestować działanie na sterowniku (**nie należy uruchamiać sterownika bez zgody prowadzącego ćwiczenia**).

2.4 Symulacja działania podnośnika B

Zadanie polega na stworzeniu programu, który jest modyfikacją programu z poprzedniego podpunktu. Układ z części A zaopatrzono w dodatkowy przycisk zatrzymania (Z).

Podnośnik B pracuje na podobnej zasadzie co podnośnik A. Ustalono, że ruch silnika jest aktywowany naciśnięciem przycisku G lub D, przy czym przycisk powinien być naciśnięty jednorazowo (a nie jak w części A - stale). Wybór powinien zostać “zapamiętany” przez układ sterowania. Przerwanie ruchu silnika następuje, gdy ładunek zostanie wykryty przez jeden z czujników (tak jak w części A) lub gdy zostanie wciśnięty przycisk Z.