Politechnika Lubelska
Katedra Automatyki i Metrologii

Laboratorium

Podstawy Automatyki

MECHATRONIKA

Ćwiczenie nr 7

Temat: Realizacja układów sterowania binarnego na bazie sterownika PLC

Lublin 2015

7.1 Wstęp

Programowalne sterowniki logiczne PLC (ang. *PLC - Programmable Logic Controllers*) stanowią cyfrowe urządzenia mikroprocesorowe służące do automatyzacji (sterowania) procesów przemysłowych. W swojej 30-to letniej historii przeszły bardzo głęboką ewolucję - od programowalnych układów sterowania binarnego, zastępujących "przekaźnikowe szafy sterownicze"- do złożonych systemów mikrokomputerowych, realizujących oprócz zadań sterowania logicznego, złożone zadania regulacji cyfrowej, obliczeń, diagnostyki i komunikacji w zdecentralizowanym systemie automatyzacji kompleksowej.

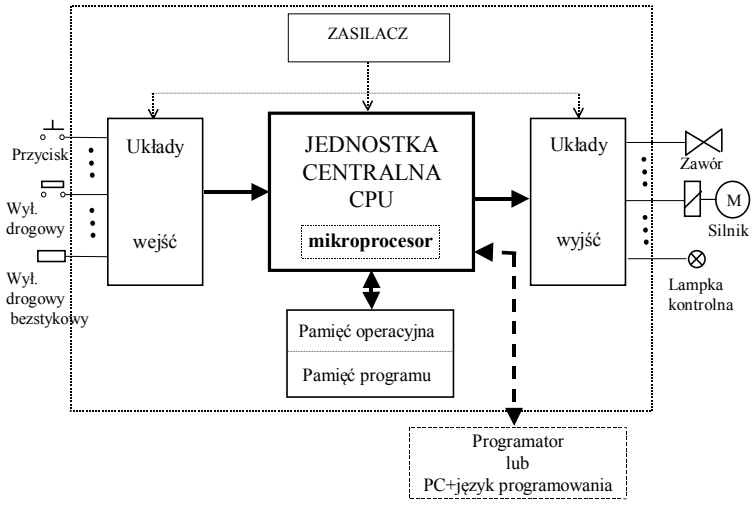
Obecnie zaciera się granica w możliwościach funkcjonalnych i mocach obliczeniowych pomiędzy sterownikami PLC, komputerami przemysłowymi i komputerami klasy PC. Daje się zauważyć postępujący proces unifikacji sterowników z akcentowaniem takich cech jak: niezawodność, uniwersalność, ciągłość produkcji, otwartość i kompatybilność z innymi sterownikami, sprawny serwis oraz możliwości komunikacyjne. Producenci proponują całe rodziny różnej „wielkości" modeli sterowników, obejmujących zarówno "małe" (mikro, mini) zintegrowane systemy typu kompakt (o liczbie we/ wy rzędu kilkunastu), jak i "duże" systemy modułowe (zestawiane w zależności od potrzeb użytkownika), mogące realizować złożone zadania sterowania binarnego, zadania regulacyjne, komunikacyjne (praca w sieci) jak i złożone obliczenia optymalizacyjne. Światowymi liderami na rynku sterowników PLC są obecnie takie firmy jak: Siemens, Allen-Bradley, GE-Fanuc, Mitsubishi, AEG - Modicon, Omron.

W związku z coraz powszechniejszym stosowaniem sterowników PLC, pojawiła się konieczność ich standaryzacji. W 1993 roku International Electronical Commission opracowała i wydała normę IEC 1131 „Programmable Controllers", dotyczącą standaryzacji sprzętu i języków programowania sterowników PLC. Wprowadzono w niej ujednoliconą koncepcję programowania PLC w językach tekstowych i graficznych, dzięki której użytkownik może być w stanie programować bez większego trudu różne, zgodne z nią, systemy PLC.

7.2 Charakterystyka sterowników PLC

7.2.1 Budowa sterowników PLC

Zastosowanie w sterownikach logicznych mikroprocesorowej jednostki centralnej 8080 w 1977 roku (firma Allen-Bradley) zapoczątkowało ich dynamiczny rozwój. Obecnie większość sterowników budowana jest na bazie mikroprocesorów specjalizowanych. Ogólny schemat strukturalny mikroprocesorowego sterownika PLC przedstawiono na rys. 9.1.



Rys. 7.1. Uproszczony schemat struktury mikroprocesorowego sterownika logicznego

Jednostka centralna CPU (ang. *Central Processing Unit*) jest najczęściej projektowana jako układ wieloprocesorowy. Liczba oraz typ mikroprocesorów, pracujących w jednostce centralnej ma wpływ przede wszystkim na szybkość działania sterownika, liczbę obsługiwanych obwodów wejściowo-wyjściowych jak

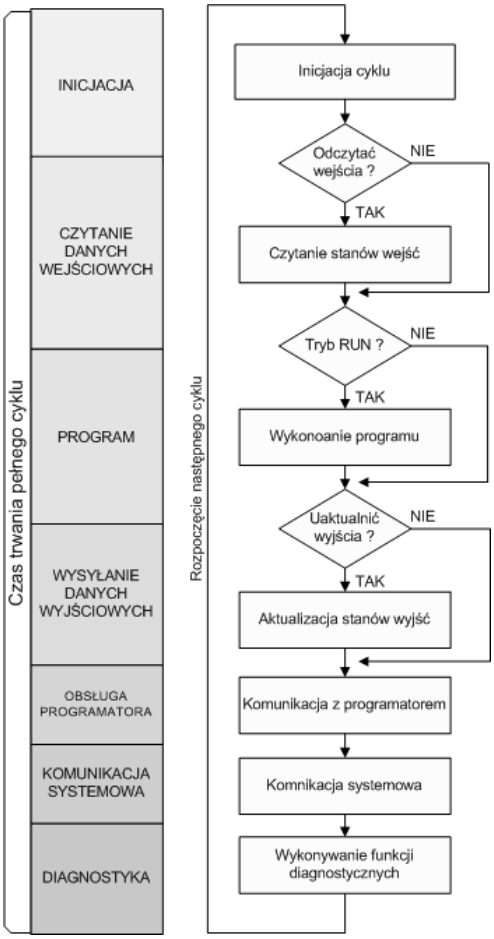
również pojemność pamięci. Każda firma produkująca sterowniki oferuje z reguły kilka ich typów przeznaczonych do realizacji zadań o różnym wymiarze. Najmniejsze obsługują kilkanaście kanałów wejść i wyjść (przeważnie z przewagą liczby wejść). Największe przystosowane są do sterowania dużymi obiektami i oprócz możliwości obsługi wejść i wyjść cyfrowych (dwustanowych) posiadają zdolność obsługi sygnałów analogowych. CPU zapewnia cykliczność pracy sterownika. Typowy cykl programowy sterownika składa się z następujących faz: inicjacja cyklu, czytanie sygnałów wejściowych, wykonanie programu użytkownika, aktualizacja sygnałów wyjściowych, transmisja danych, komunikacja systemowa, diagnostyka.

Większość sterowników posiada możliwość pracy w trzech trybach:
- RUN – uruchomienia programu użytkownika,
- STOP – zatrzymanie wykonywania programu użytkownika,
- REMOTE – zdalnego sterowania, wówczas tryb pracy ustawiany jest z poziomu programatora lub nadrzędnej jednostki sterującej.

Niektóre z powyższych faz mogą być w pewnych trybach pracy sterownika pomijane, co prezentuje algorytm pracy sterownika zamieszczony na rys. 4.2.

Program użytkownika wykonywany jest szeregowo tzn. od pierwszej do ostatniej instrukcji. Wykonanie programu polega przede wszystkim na obliczeniu i ustawianiu stanów sygnałów wyjściowych na podstawie odczytanych przed rozpoczęciem wykonywania programu użytkownika stanów sygnałów wejściowych. Zmiany sygnałów wejściowych, które nastąpiły po rozpoczęciu cyklu będą mogły być uwzględnione dopiero w cyklu następnym. Istnieją często odstępstwa od tej reguły (mechanizm przerwań).

Konsekwencją cykliczności wykonywania programu jest:
- Czas wykonywania programu zależy od jego długości i parametrów sterownika. Opóźnienie wnoszone do układu sterowania przez sterownik w najgorszym przypadku może wynosić dwa czasy cyklu.
- Jeżeli sygnał wejściowy trwa krócej niż czas cyklu, to może być on przez sterownik nie wzięty pod uwagę. Tę niekorzystną cechę eliminują rozwiązania polegające na przerwaniach alarmowych.
- W trakcie wykonywania przez sterownik cyklu programu stany wejść zachowują takie same wartości logiczne, chyba że korzysta się z mechanizmu przerwań.



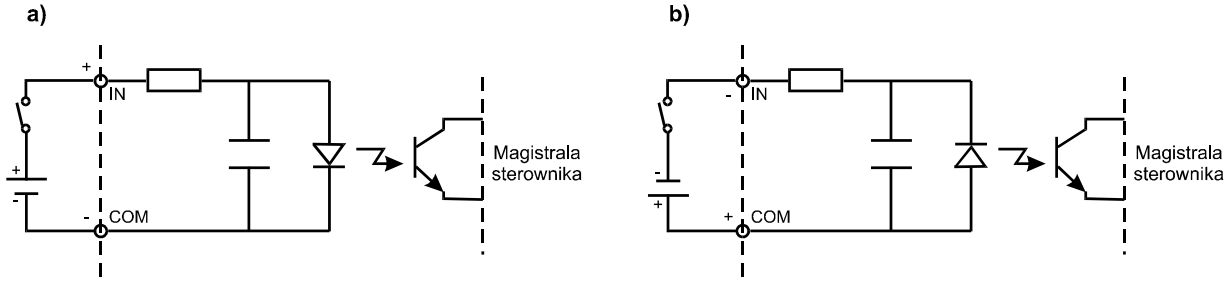
Rys. 7.2. Fazy cyklu pracy sterownika PLC

Pamięć w sterowniku służy do przechowywania programu oraz informacji pośrednich, powstających w trakcie jego wykonywania. Jest to pamięć typu RAM, nieulotna np. EPROM lub EEPROM. Podział pamięci na pamięć operacyjną i pamięć programu nie jest sztywny. Najczęściej w trakcie uruchamiania i testowania, program jest zapisywany w pamięci operacyjnej RAM. Ostateczna jego wersja może być tam pozostawiona albo zapisana na "trwałe" w pamięci stałej.

Układy wejść i wyjść stanowią połączenie sterownika ze sterowanym obiektem. W sterownikach PLC stosowane są dwa rodzaje wejść/wyjść: dyskretne i analogowe. Układy wejść/wyjść dyskretnych ze występują niemal we wszystkich sterownikach PLC. Z kolei układy wejść/wyjść analogowych ze względu na swoją bardziej złożoną budowę (konieczność przetwarzania sygnału analogowego na cyfrowy i odwrotnie) są rzadszym elementem sterowników.

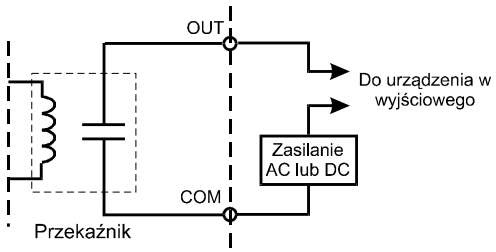
a) wejścia dyskretne, nazywane również wejściami cyfrowymi (ang. *digital inputs*) zamieniają pochodzące z urządzeń (przyciski, przełączniki, wyłączniki krańcowe, etc.) sygnały prądu stałego lub przemiennego na sygnały logiczne (dwustanowe) akceptowane przez sterownik. W produkowanych obecnie sterownikach do takiej zamiany wykorzystywany jest zazwyczaj przetwornik optyczny, zapewniający dodatkowo optoizolację pomiędzy obwodami wejściowymi a magistralą sterownika (patrz rys. 7.3). W przypadku wejść prądu stałego polaryzacja źródła zasilania obwodów wejściowych zależy od typu zastosowanego układu wejściowego:

- ujście (ang. *SINK IN*) tzn. z polaryzacją dodatnią (patrz rys. 7.3 a) nazywane układami o logice dodatniej (najczęściej spotykane),
- źródło (ang. *SOURCE IN*) tzn. z polaryzacją ujemną (patrz rys. 4.3 b) nazywane układami o logice ujemnej.



Rys. 7.3. Schemat pojedynczego obwodu układu wejść cyfrowych: a) z polaryzacją dodatnią (typu ujście), b) z polaryzacją ujemną (typu źródło)

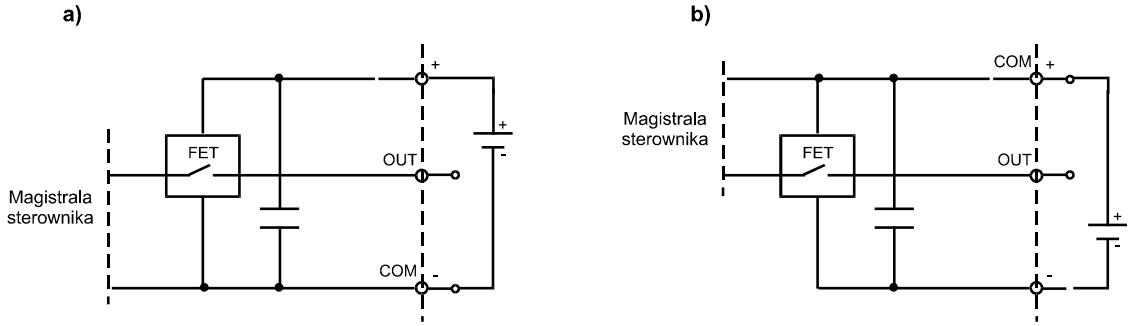
b) wyjścia dyskretne, nazywane również wyjściami cyfrowymi (ang. *digital outputs*) zamieniają sygnały binarne sterownika na sygnały prądu stałego lub przemiennego potrzebne do wysterowania urządzeń wyjściowych (cewki styczników, lampki kontrolne, etc.). Zamiany tych sygnałów dokonuje się poprzez zamykanie lub otwieranie zasilanych z zewnętrznego źródła obwodów wyjściowych za pomocą przekaźników (wyjścia przekaźnikowe, ang. *Relay Output* – rys. 7.4) lub łączników tranzystorowych (wyjście „napięciowe").



Rys. 7.4. Schemat pojedynczego obwodu układu wyjść przekaźnikowych

W przypadku obwodów wyjściowych z łącznikami tranzystorowymi istnieją dwa rozwiązania (podobnie jak w przypadku wejść prądu stałego):

- źródło (ang. *SOURCE OUT*) - najczęściej spotykane (patrz rys. 7.5 a),
- ujście (ang. *SINK OUT*) przedstawione na rys. 7.5 b.



Rys. 7.5. Schemat pojedynczego obwodu układu wyjść z łącznikami tranzystorowymi: a) ze „wspólną masą" (typu źródło), b) ze „wspólnym plusem" (typu ujście)

W zależności od typu i wykonania sterownika dwustanowe sygnały wejściowe mogą mieć postać sygnałów napięciowych prądu stałego lub przemiennego o wartości "1"od 5V do 220V (najbardziej rozpowszechnione jest 24V).
c) wejścia analogowe, (ang. *analog input*) zamieniają pochodzące z czujników sygnały analogowe (ciągłe) na sygnały cyfrowe. Konwersja tych sygnałów realizowana jest za pomocą przetworników analogowo-cyfrowych ADC (ang. *Analog to Digital Converter*).
d) wyjścia analogowe, (ang. *analog output*) zamieniają sygnały cyfrowe na sygnały ciągłe sterujące urządzeniami wykonawczymi. Konwersja tych sygnałów realizowana jest za pomocą przetworników cyfrowo-analogowych DAC (ang. *Digital to Analog Converter*).
Parametrami charakteryzującymi przetworniki ADC i DAC są:
- zakres napięć wejściowych/wyjściowych (najczęściej ±10 V),
- rozdzielczość – napięcie przypadające na najmniej znaczący bit przetwornika,
- czas przetwarzania,
- częstotliwość przetwarzania.

Zależnie od rodzaju sterownika PLC przedstawione powyżej jego elementy składowe mogą być zintegrowane w jednej obudowie (sterownik kompaktowy) lub mogą stanowić oddzielne moduły montowane w gniazdach (ang. *slots*) płyty łączeniowej sterownika zwanej kasetą (ang. *rack*) – sterownik modułowy.

7.2.2 Programowanie sterowników PLC

Sterowniki PLC programowane są za pomocą specjalnych urządzeń mikrokomputerowych zwanych programatorami lub komputerów PC z zainstalowanym oprogramowaniem narzędziowym (język programowania). Języki programowania sterowników można podzielić na dwie grupy: języki tekstowe i graficzne.

Do grupy **języków tekstowych** należą:

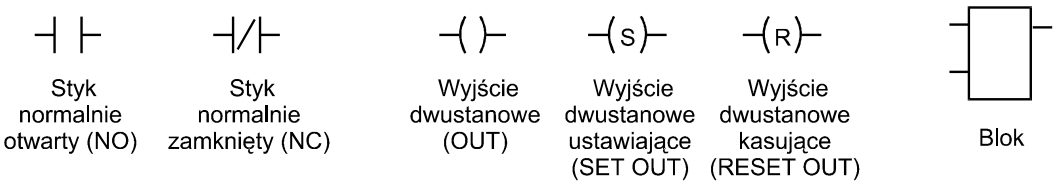
- **Lista instrukcji IL** (ang. *Instruction List*) - jest językiem niskiego poziomu, zbliżonym do języka typu assembler. Program w tym języku jest zestawem instrukcji mnemotechnicznych realizujących algorytm sterowania. Język wykorzystuje zbiór instrukcji, obejmujących operacje logiczne, arytmetyczne, relacji, funkcje przerzutników, czasomierzy, liczników itp. Język tego typu może znaleźć zastosowanie w programowaniu małych i prostych aplikacji.
- **Język strukturalny ST** (ang. *Structured Text*) - jest odpowiednikiem algorytmicznego języka wyższego poziomu, zawierającego struktury -programowe takie jak:

  If...then...else...end_if,
  For...to...do...end_for,
  While...do...end_while

  Język tego typu może być używany do obliczania złożonych wyrażeń, zawierających wielkości analogowe i binarne.
- **Lista instrukcji STL** (ang. *StatemenT List*) – stanowi połączenie języków IL oraz ST.

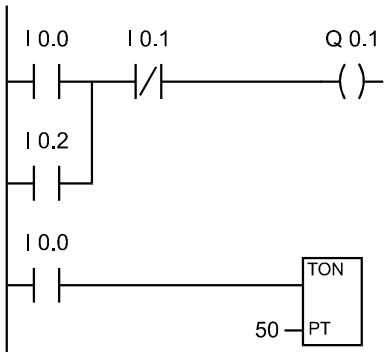
Do grupy **języków graficznych** zaliczane są następujące języki:

Język schematów drabinkowych LAD (ang. *LAdder Diagram*) - bazuje na symbolach logiki stykowo-przekaźnikowej. Podstawowymi symbolami języka LAD są przedstawione na rys. 7.6: styki (elementy wejściowe), wyjścia dwustanowe (odzwierciedlenie cewek przekaźnika) oraz bloki funkcyjne (liczniki, timery, operacje matematyczne, etc).



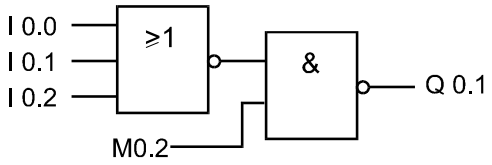
Rys.7.6. Podstawowe elementy języka LAD

Symbole te umieszcza się w obwodach (ang. *network*) w sposób podobny do szczebli (ang. *rungs*) w schematach drabinkowych dla przekaźnikowych układów sterowania (patrz rys. 7.7). Obwód LD ograniczony jest z lewej i prawej strony przez szyny prądowe. Prawa szyna może być rysowana w sposób jawny lub pozostawać w domyśle.



Rys.7.7. Przykładowa aplikacja zrealizowana w języku LAD

- **Język bloków funkcyjnych FBD** (ang. *Function Block Diagram*) - jest wzorowany na schematach blokowych układów scalonych. Realizacja programu w języku FBD opiera się na przepływie sygnału. Przepływ sygnału następuje z wyjścia funkcji lub bloku funkcyjnego do przyłączonego wejścia następnej funkcji lub bloku funkcyjnego (fragment programu realizowanego w języku FBD przedstawia rys. 7.8.



Rys.7.8. Przykładowa aplikacja zrealizowana w języku FBD

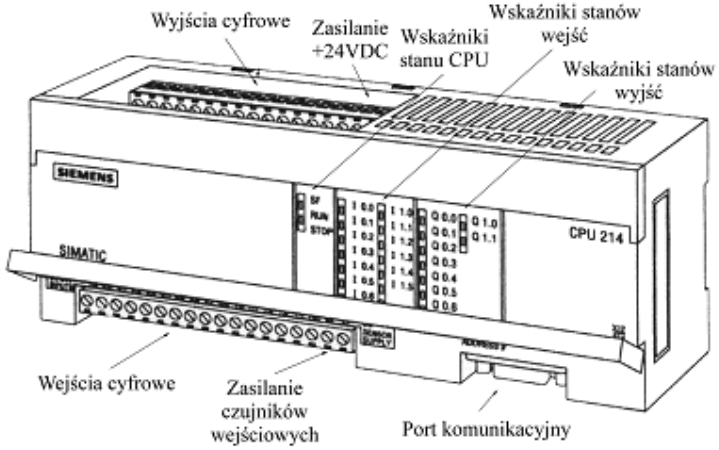
7.2 Cechy funkcjonalne sterownika SIMATIC S7-200

Sterownik S7-200 nazywany jest mikro PLC ze względu na swoje niewielkie wymiary (patrz rys. 7.9). Jednostka centralna S7-200 zbudowana jest w postaci bloku, ze zintegrowanymi układami wejść/wyjść (budowa kompaktowa). Może on być stosowany w mniejszych, samodzielnych aplikacjach przemysłowych, takich jak myjnie samochodowe, mieszarki, linie butelkowania i pakowania itp.

7.3.1. Budowa sterownika SIMATIC S7-200

Na rynku znajdują się dwa typy sterownika S7-200 z CPU 212 i CPU 214. W ćwiczeniu wykorzystywany jest sterownik z CPU 214, model 6ES7 214-1AC01-0XB0.
Poszczególne modele sterowników różnią się między sobą liczbą wejść i wyjść rodzajem zasilania (zintegrowany zasilacz lub nie). Parametry techniczne omawianego sterownika zostały zamieszczone w tablicy 7.1. Sterownik jest ponadto wyposażony w:

- dwa potencjometry analogowe (umieszczone pod pokrywą wyjść cyfrowych) pozwalające na ręczne nastawy dla dwóch zmiennych np. wartości zadanej),
- zegar/kalendarz czasu rzeczywistego TOD (ang. *Time-of-Day Clock*).



Rys.7.9.Wygląd zewnętrzny sterownika SIMATIC S7-200

Znaczenie poszczególnych wskaźników stanu CPU jest następujące:
- **SF** (dioda czerwona) – oznacza *błąd systemu* (ang. *System Fault*),
- **RUN** (dioda zielona) – sterownik w trybie RUN,
- **STOP** (dioda żółta) – sterownik w trybie STOP.

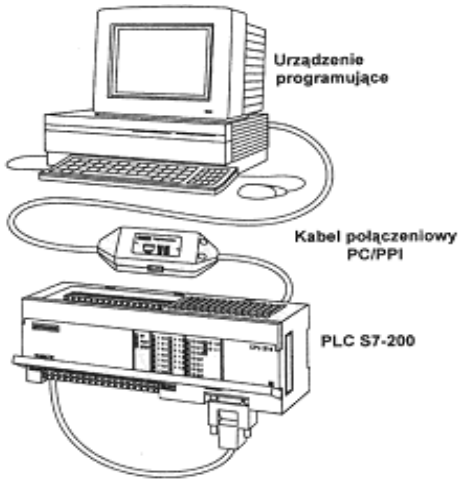
Znaczenie poszczególnych pozycji przełącznika trybu pracy sterownika (umieszczonego pod pokrywą wyjść cyfrowych) jest następujące:
- **RUN** – przełączenie w tryb wykonywania programu,
- **STOP** – przerwanie wykonywania programu. W tym trybie sterownik powinien się znajdować podczas edycji (on-line) programu lub podczas jego załadowywania do sterownika,
- **TERM** – zdalne (z poziomu programatora) przełączanie trybów pracy sterownika (ang. *Terminal*).

Tablica 7.1. Parametry techniczne sterownika *SIMATIC S7-200 model 6ES7 214-1AC01-0XB0*

Informacje ogólne		Układy wyjściowe	
Wymiary/Waga	197 x 80 x 62 mm / 0,4 kg	Max. obciążenie prądowe	0,75 A
Max. rozmiar programu użytkownika	2K słów /EEPROM		
Max. ilość danych	2K słów / RAM	Opóźnienie przełączania	25 μs ON, 120 μs
Liczba wejść/wyjść cyfrowych	14 wejść / 10 wyjść	Izolacja optyczna	500 VAC (1 minuta)
Max. liczba modułów zewnętrznych	7	Zabezpieczenie przed zwarciem	Brak
Max. liczba zewn. we/wy cyfrowych	64 wejść / 64 wyjść	Szybkie liczniki	2 sprzętowe (7 kHz max.), 1 programowy (2 kHz max.)
Max. liczba zewn. we/wy analogowych	16 wejść / 16 wyjść		
Szybkość wykonywania operacji log	0.8 μs / instrukcję		
Wewnętrznych bitów pamięci	256		
Timery	128		2 (4 kHz max.)
Liczniki	128		
Układy wejściowe		**Zasilanie**	
Napięcie w stanie aktywnym (ON)	15 – 30 VDC	Zakres napięć	20,4 – 28,8 VDC
Prąd wejścia w stanie aktywnym	4 mA (min.)	Max. pobór prądu	900 mA
Napięcie w stanie nieaktywnym (OFF)	0 - 5 VDC	**Zasilanie sensorów**	
Prąd wejścia w stanie nieaktywnym	1 mA	Zakres napięć	16,4 – 28,8 VDC
Izolacja optyczna	500 VAC (1 minuta)	Max. prąd zwr.	600 mA

7.3.2 Komunikacja z urządzeniami zewnętrznymi

Komunikacja z urządzeniami zewnętrznymi odbywa się poprzez port komunikacyjny. Urządzeniami tymi mogą być: programatory, komputer PC, wyświetlacze tekstowe, drukarki itp.
Komunikacja pomiędzy programatorem firmy Siemens (PG 720, PG 740, PG 702) i sterownikiem odbywa się za pośrednictwem protokołu PPI (ang. *Point-Point Interface*) - interfejs szeregowy RS-485.
Gdy jako urządzenie programujące używany jest komputer PC konieczne jest użycie specjalnego kabla PPI/PC (z konwerterem RS-485 na RS-232). Rozwiązanie takie prezentuje rys. 7.10. Przełączniki DIP na konwerterze PPI/PC służą do sprzętowego ustawienia prędkości transmisji.

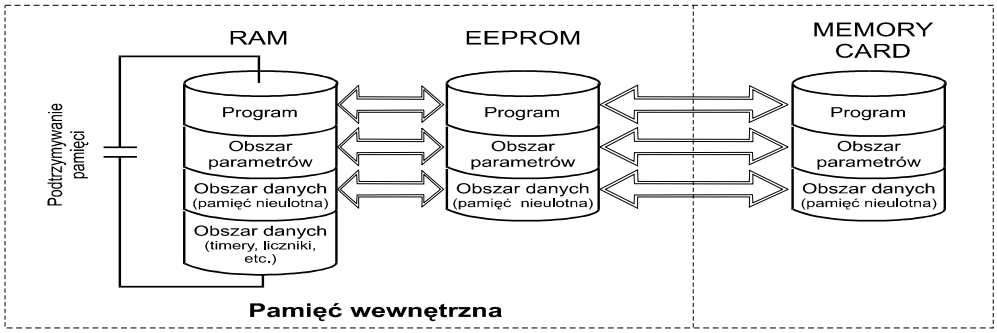


Rys.7.10. Komunikacja z komputerem

7.3.3 Organizacja pamięci sterownika

Pamięć sterownika SIMATIC S7-200 jest podzielona na trzy obszary (patrz rys. 7.11):

- obszar programu – przechowuje stworzone w języku drabinkowym (LAD) lub języku STL, instrukcje programu (program użytkownika),
- obszar parametrów – przechowuje parametry konfiguracyjne domyślne i definiowalne (hasło, adres stacji roboczej, itp.),
- obszar danych – używany jest jako obszar roboczy: wykonywanie obliczeń , pamięć tymczasowa(akumulator i rejestry). Obszar ten zajmowany jest również przez dane pamięciowe (ang. *Data Memory*) i przez dane specjalnych urządzeń (ang. *Data Objects*) jak timery, liczniki, itp. Zawartość obszaru danych oraz zakres i dostęp (adresowanie) do jego poszczególnych elementów przedstawia rys.7.11.

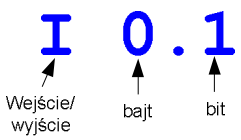


Rys. 7.11. Pamięć sterownika S7-200

Bity specjalne SM (ang. *Special Memory Bits*) dostarczają informacji statusowych (np. informacje o różnego rodzaju błędach), służą do wyboru i sterowania różnego rodzaju funkcjami oraz jako środek komunikacji pomiędzy systemem PLC i programem użytkownika.

7.3.4. Adresowanie wejść/wyjść

Wejścia i wyjścia cyfrowe sterownika S7-200 opisane są na listwach zaciskowych oraz przy diodowych wskaźnikach ich stanu. Znaczenie poszczególnych symboli w ich opisie wyjaśnia rys. 7.12.



Rys. 7.12. Znaczenie symboli w adresie

Symbol **I** przeznaczony jest dla wejść cyfrowych, natomiast symbol **Q** dla wyjść cyfrowych.

Adresowanie bezpośrednie i symboliczne

Parametry elementów sterownika (inaczej: instrukcje) programu można deklarować bezpośrednio (*absolutely*) lub symbolicznie (*symbolically*). Pierwszy sposób polega na określeniu obszaru pamięci oraz lokacji bitu lub bajtu do identyfikacji adresu. Deklarowanie symboliczne (pośrednie) wykorzystuje kombinację znaków alfanumerycznych do określenia adresu wejścia lub wyjścia.

Przykłady wyświetlania adresów przez Program editor:

`I0.0`	Adres bezpośredni jest deklarowany przez określenie obszaru pamięci oraz adresu
`%I0.0`	(SIMATIC)
`#INPUT1`	W standardzie IEC adres bezpośredni jest dodatkowo poprzedzony znakiem % (IEC)
`"INPUT1"`	Deklaracja zmiennej lokalnej przy użyciu znaku ‘ # ‘ (SIMATIC lub IEC)
`??.?`	Ujęcie w cudzysłów na potrzeby utworzenia zmiennej globalnej (SIMATIC lub IEC)
	Czerwone znaki zapytania wyróżniają nie zadeklarowany adres
	(należy je zdefiniować przed podjęciem kompilacji programu)

7.3.4.Szybkie liczniki i wyjścia impulsowe

Sterownik z CPU 214 posiada trzy szybkie liczniki (ang. *High Speed Counter*) HSC0, HSC1, HSC2. HSC0 jest dwukierunkowym licznikiem programowym (max. częstotliwość zliczania 2 kHz). HSC1 i HSC2 są licznikami sprzętowymi mogącymi pracować w jednym z dwunastu trybów pracy (max. częstotliwość zliczania 7 kHz). Liczniki te można konfigurować do pracy wspólnej wówczas max. częstotliwość zliczania wynosi 28 kHz. Jako wejścia dla tych liczników można użyć wejścia cyfrowe: I0.0 (HSC0), I0.6 – I1.1 (HSC1), I1.2 – I1.5 (HSC2).
W sterowniku z CPU 214 dostępne są instrukcje „szybkiego wyjścia” (wyjścia impulsowe). Wyjście 1 (Q0.0) i wyjście 2 (Q0.1) może służyć do generowania ciągu impulsów (PTO) lub impulsów z modulacją PWM.

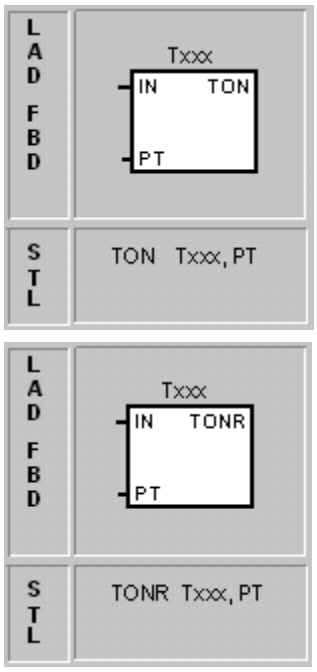
7.4.Programowanie sterownika SIMATIC S7-200

Do tworzenia programów roboczych dla sterowników SIMATIC S7-200 używane jest oprogramowanie STEP 7-Micro. Program użytkowy składa się z pewnej liczby instrukcji ułożonych w odpowiednim porządku logicznym odzwierciedlającym opis pracy sterowanego urządzenia. Instrukcje podzielone zostały tutaj na trzy grupy:

- instrukcje standardowe – podstawowe rozkazy procesora, instrukcje binarne, instrukcje opisujące pętle programowe, timery, liczniki, itp.,
- instrukcje specjalne – instrukcje używane do obsługi danych (rozkazy przesunięcia, grupowania w tablicach, szukania, konwersji,
- instrukcje szybkie – instrukcje umożliwiające obsługę zdarzeń w trybie przerwań, niezależnie od czasu skanowania PLC (instrukcje obsługi szybkich liczników, przerwań obiektowych, instrukcje transmisji).

7.4.1.Liczniki czasu (*ang. timers*).

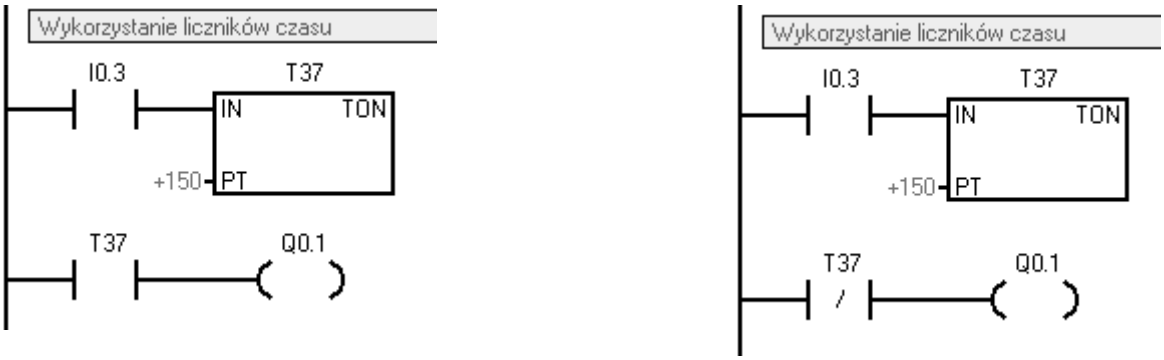
Timerami nazywane są funkcje pomiaru zadanych odcinków czasu. Timery umożliwiają wykonanie pewnych czynności w określonych chwilach, wynikających z charakteru zastosowania. Korzystając z licznika czasu, można na przykład włączyć silnik wirówki w pralce na 30 sekund albo w 2 sekundy po wydaniu rozkazu zamknięcia sprawdzić, czy brama wjazdowa do obiektu została zamknięta.



Opóźnione załączenie. Timer zlicza jednostki czasu, gdy do jego wejścia IN zostanie doprowadzony sygnał IN = l, a jest zerowany wtedy, gdy sygnał IN = 0. Po ponownym pojawieniu się sygnału IN = l pomiar czasu rozpoczyna się od początku. Po doliczeniu do wartości określonej przez stałą podaną na wejście PT timer zwiera swój styk wyjściowy, oznaczony tą samą nazwą co nazwa timera. Maksymalny zakres zliczania wynosi 32767 jednostek czasu.
W sterowniku S7-214 są timery odmierzające czas z różną rozdzielczością. Timery **T32** i **T96** zliczają jednostki czasu o długości **l ms, T33 - T36** oraz **T97 - T100** zliczają jednostki czasu równe **10 ms,** a T37 - **T63** oraz **T101 -** T127 jednostki równe **100** ***ms.***

Opóźnione załączenie z podtrzymaniem. Timer z podtrzymaniem różni się od poprzedniego tym, że sygnał wejściowy IN = O nie zeruje zawartości licznika, tylko zawiesza zliczanie, które jest kontynuowane w chwili ponownego ustawienia sygnału IN = l. Timer mierzy więc sumaryczny czas trwania sygnału IN = l. Timer z podtrzymaniem można wyzerować za pomocą instrukcji **RESET** (jak w przerzutniku RESET). Po doliczeniu do wartości określonej przez stałą podaną na wejście PT timer zwiera swój styk wyjściowy, oznaczony tą samą nazwą, co jego nazwa. Maksymalny zakres zliczania wynosi 32767 jednostek czasu.
W sterowniku S7-214 timery z podtrzymaniem odmierzają czas z różną rozdzielczością. Timery o nazwach **TO - T64** zliczają jednostki czasu o długości **l ms, T65 - T68** zliczają jednostki **10 ms, a T69** -T95 jednostki **100 ms.**

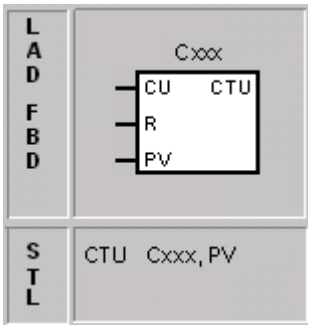
Przykład użycia timerów przedstawiony na rys. 7.13 dotyczy prostego układu składającego się z lampki włączanej za pomocą bistabilnego przycisku. Sterowanie ma zapewniać programowaną zwłokę zapalania (lub gaszenia) lampki w stosunku do momentu zmiany stanu przycisku.
Przycisk jest dołączony do wejścia 10.3, a lampka do wyjścia Q0.1. Po naciśnięciu przycisku wejście 10.3 jest równe jeden i wyzwala licznik T37 Podstawą czasu T37 jest 100 ms. Wartość zadana dla licznika PT = 150 zapewnia zwłokę równą 15 s, po której nastąpi zwarcie styku T37. Lampka zapali się więc po czasie równym 15 s od chwili wciśnięcia przycisku. Jeżeli przełącznik zostanie otwarty przed upływem 15 s, lampka nie będzie włączona. Ponowne wciśnięcie przycisku spowoduje odliczanie licznika od zera.
Przez przeprogramowanie styku T37 na „normalnie zamknięty", funkcja układu jest zmieniona i powoduje wyłączenie światła tylko wtedy, gdy licznik czasu odmierzy 15 s, czyli po upływie 15 s od wciśnięcia przycisku. Zmiana działania jest wykonana bez zmiany połączeń wejść i wyjść sterownika.



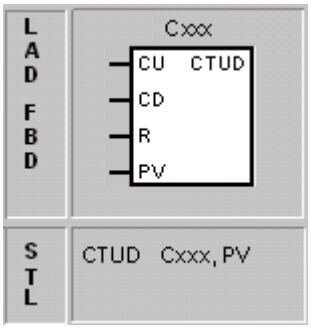
Rys.7.13. Przykład wykorzystania licznika czasu

7.4.2.Liczniki zdarzeń (*ang. counters*)

Liczniki zdarzeń to funkcje zliczania określonych stanów wybranych zmiennych (np. stanu sygnału z czujnika). Liczniki porównują wartość zliczoną z wartością zadaną. Wykorzystywane są do liczenia zdarzeń do chwili osiągnięcia nastawionej wartości zadanej w celu realizacji kolejnego kroku algorytmu. Na przykład maszyna pakująca butelki ma licznik zdarzeń do zliczania butelek w grupy po sześć.

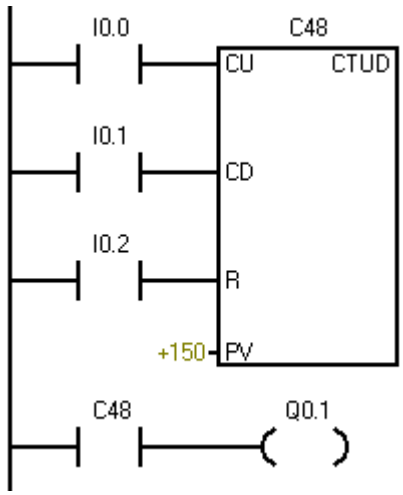


Licznik rosnący. Zlicza zmiany wartości z 0 na l sygnału podanego na wejście CU. Licznik jest zerowany, gdy na wejście kasujące R zostanie podany sygnał o wartości l. Po doliczeniu do wartości równej stałej podanej na wejściu PV, licznik zwiera swój styk wyjściowy, oznaczony tą samą nazwą, co jego nazwa (Cxx = l). Zakres zliczania: (0-32767).S7-214 ma 128 liczników o kolejnych adresach: **C0 - C127.**



Licznik dwukierunkowy. Jego zawartość może zarówno rosnąć, jak i maleć, wskutek zliczania impulsów na jego wejściach. Każda zmiana z 0 na l wartości sygnału podanego na wejście CU powoduje zwiększenie zawartości licznika o l, natomiast zmiana z 0 na l sygnału na wejściu CD powoduje zmniejszenie tej zawartości o l. Wejście R służy do zerowania licznika. Przy zrównaniu się liczby zliczonych impulsów z wartością zadaną na wejściu PV licznik zwiera swój styk wyjściowy, oznaczony tą samą nazwą, co jego nazwa. Zakres zliczania: (-32768, +32767). S7-214 ma 28 tych liczników, o kolejnych adresach: **C0 - C27**

Licznik może być wykorzystany na przykład do zapewnienia płynności poruszania się określonej liczby samochodów w obszarze parkingu. Prosty sterujący program jest pokazany na rys. 4.13. Kiedy samochód wjeżdża na parking przez bramę wjazdową, wartość licznika jest powiększana o l. Podczas wyjeżdżania samochodu z parkingu wartość licznika zmniejsza się o l. Kiedy parking zostanie zapełniony, a więc gdy zawartość licznika zrówna się z zadaną wartością PV, przy wjeździe na parking zapali się czerwone światło.



Czujnik otwarcia bramki wjazdowej jest podłączony do wejścia I0.0.

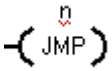
Czujnik otwarcia bramki wyjazdowej jest podłączony do wejścia I0.1.

Przełącznik kasowania, umieszczony w budce dyżurnego, jest podłączony do wejścia I0.2.

Parking ma 150 miejsc. Wyjście licznika, bit C48, steruje wyjście Q0.1, które jest podłączone do czerwonej lampki „parking pełen".

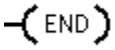
Rys. 7.14. Pętla programowa PLC

7.4.3.Bloki sterujące



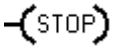
Skok warunkowy. Wykonanie instrukcji powoduje pominięcie części programu sterującego, umieszczonego między instrukcją *JUMP n* a etykietą *LABEL n.* Instrukcja skoku zostanie wykonana, gdy poprzedzające instrukcje w tym samym obwodzie schematu drabinkowego przekażą jej sygnał równy l

Etykieta. Etykieta określa miejsce docelowe *n,* do którego można wykonać skok. Samo zadeklarowanie etykiety nie wpływa na sposób wykonania programu. Program może zawierać co najwyżej 256 etykiet (n = 0 - 255).



Zakończenie warunkowe. Instrukcja powoduje zatrzymanie programu w miejscu, w którym występuje i rozpoczęcie cyklu wykonania programu od początku. Instrukcja zostanie wykonana, gdy poprzedzające instrukcje w tym samym obwodzie schematu drabinkowego przekażą jej sygnał równy l.

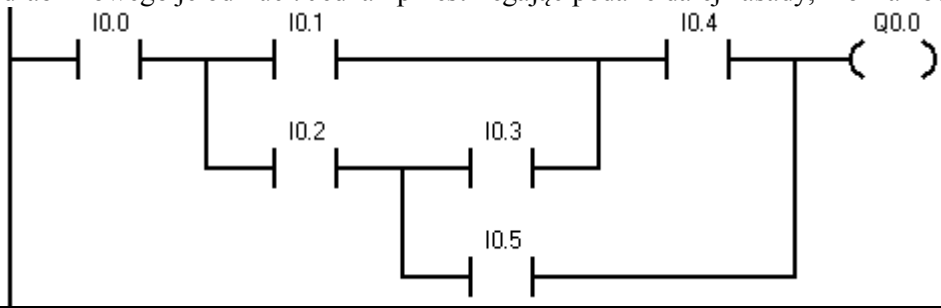
Zakończenie bezwarunkowe. Jest ostatnim elementem programu. Powoduje rozpoczęcie nowego cyklu wykonania programu od początku.



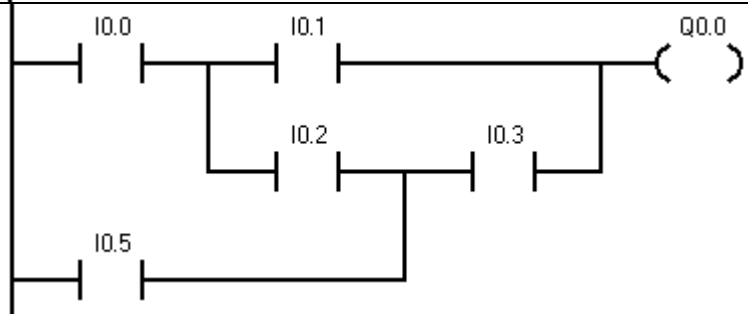
Stop warunkowy. Instrukcja kończy wykonywanie programu i powoduje natychmiastowe przejście sterownika do trybu STOP. Instrukcja zostanie wykonana, gdy poprzedzające instrukcje w tym samym obwodzie schematu drabinkowego przekażą jej sygnał równy l.

7.4.4.Ograniczenia struktury programu

Projektując szczeble drabiny programu należy pamiętać, że istnieją ograniczenia co do stopnia skomplikowania ich budowy. Niektóre konstrukcje są niedozwolone (rys. 7.15) i kompilator języka drabinkowego je odrzuci. Jednak przestrzegając podane dalej zasady, można zbudować poprawny program.



Rozgałęzienie (linia zawierająca styk I0.5) bierze początek w niewłaściwym miejscu szczebla (wewnątrz innego rozgałęzienia)



Styk I0.5 jest nieprawidłowo połączony z wnętrzem odgałęzienia zawierającego styki I0.2 i I0.3

Rys 7.15 Przykładowe konstrukcje są niedozwolone

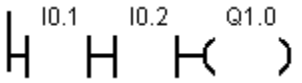
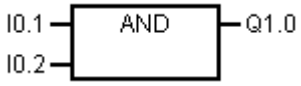
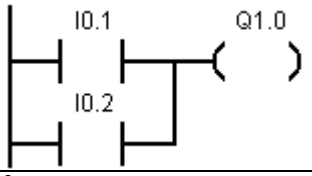
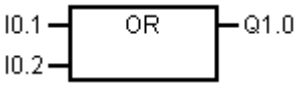
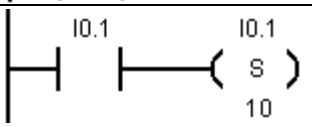
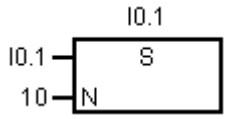
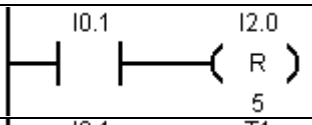
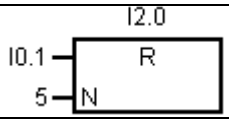
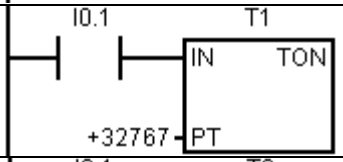
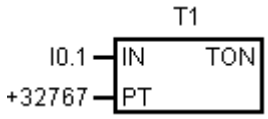
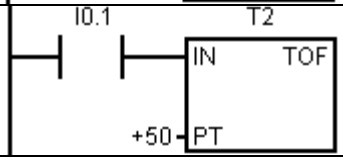
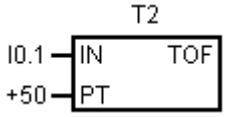
Konstrukcja obwodu programu podlega następującym ograniczeniom:

- Obwód może zawierać co najwyżej 16 linii równoległych, a linia nie może mieć więcej niż 16 elementów logicznych połączonych szeregowo.
- Ostatnim elementem szeregowego połączenia w danym obwodzie musi być przekaźnik, licznik lub blok sterujący.
- Obwód może zawierać co najwyżej 16 przekaźników.

- Obwód musi zawierać przynajmniej jeden styk przed wystąpieniem przekaźnika, bloku funkcyjnego lub połączenia pionowego.
- Nie może wystąpić rozgałęzienie mające początek lub koniec wewnątrz innego odgałęzienia.
- Nie może wystąpić rozgałęzienie mające koniec wewnątrz innego odgałęzienia

W tablicy 7.3 zestawiono reprezentacje podstawowych operacji logicznych w poszczególnych językach programowania.

Tablica 7.3. Realizacja podstawowych operacji w językach LAD, STL, FBD

Polecenie	Reprezentacja			Opis
	LAD	STL	FBD	
AND		LD I0.1 A I0.2 = Q1.0		
OR		LD I0.1 O I0.2 = Q1.0		
SET		LD I0.1 S I0.1, 10		N=10 ilość cykli
RESET		LD I0.1 R I2.0, 5		
TON On-Delay Timer		LD I0.1 TON T1, +32767		
TOF Off–Delay Timer		LD I0.1 TOF T2, +50		

7.4.4.STEP 7-Micro – wprowadzenie

W ćwiczeniu wykorzystano oprogramowanie STEP 7-Micro/WIN 32, w wersji ewaluacyjnej, pracującej pod systemem Windows. Umożliwia ono programowanie sterownika PLC w trzech językach STL (*Statement List* – język poleceń), LAD (*Ladder Diagram* – język drabinkowy)oraz FBI (*Function Block Diagram* – język bloków funkcyjnych)

Zmienne globalne i zmienne lokalne

Wartości symboliczne zapisywane są w Tabeli Symboli (Symbol Table / Global Variable Table) mają zasięg globalny. Z kolei wartości symboliczne zadeklarowane w tabeli zmiennych lokalnych (Local Variable Table) mają zasięg lokalny.

Program sterowania dla sterowników rodziny S7-200 składa się z następujących typów jednostek organizacyjnych (*Program Organizational Unit [POU]*):

Program główny (*Main program*) Miejscem gdzie umieszczone są instrukcje aplikacji sterowania jest główne ciało programu. Instrukcje te są wykonywane sekwencyjnie, jedna na cykl jednostki centralnej CPU.

Subrutyny (*Subroutines*) Podprogram, nazywany także subrutyną jest opcjonalnym zestawem instrukcji, umieszczonych w oddzielnym bloku. Jest on wykonywany tylko wtedy, gdy zostanie wywołany z programu głównego.

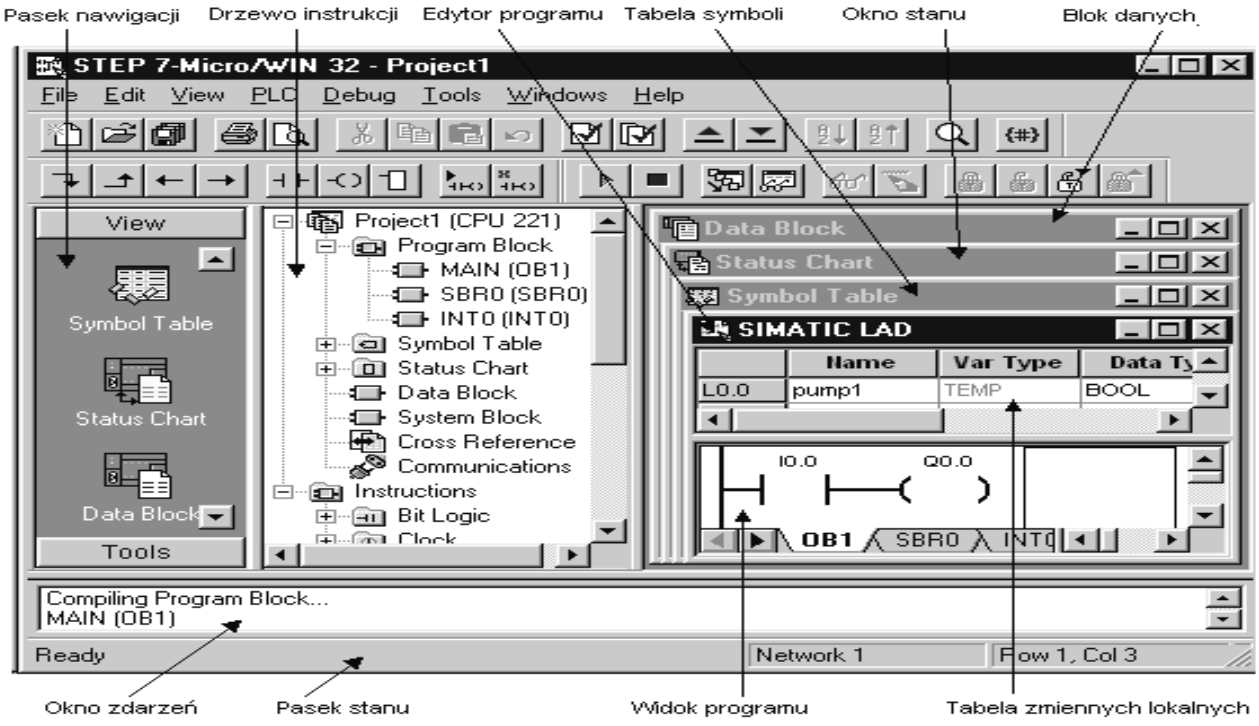
Rutyny przerwań (*Interrupt routines*) Rutyna przerwań jest opcjonalnym zestawem instrukcji, umieszczonych w oddzielnym bloku, wykonywana wówczas, gdy zachodzi zdarzenie przerwania.

STEP 7-Micro/WIN 32 uporządkowuje program poprzez wyświetlanie osobnych zakładek w oknie edytora programu dla każdego podprogramu. Program główny, OB1, jest zawsze pierwszą zakładką, poprzedzającą utworzone przez programistę subrutyny oraz rutyny przerwań.
Każdy projekt posiada pięć podstawowych komponentów:

Blok Programu (*Program Block*) składa się z wykonywalnego kodu I komentarzy. Wykonywalny kod składa się z programu głównego (OB1) oraz ewentualnych subrutyn bądź rutyn przerwań. Jest on kompilowany i przesyłany do sterownika z pominięciem komentarzy.
Blok Danych (*Data Block*) w jego skład wchodzą dane w postaci początkowych wartości zmiennych pamięciowych oraz stałych. Dane te są kompilowane i przesyłane do sterownika.
Blok Systemowy (*System Block)* przechowuje parametry konfiguracyjne dotyczące komunikacji, zakresów danych, parametry wejść cyfrowych I analogowych a także hasło dostępu. Zawartość bloku systemowego jest przesyłana do sterownika.

Tabele symboli (*Symbol Tables*) pozwalają programiście na adresowanie symboliczne, przez co kod staje się czytelniejszy. Przed załadowaniem programu z adresowaniem symbolicznym do pamięci sterownika STEP-7 Micro konwertuje wszystkie użyte symbole na adresy bezpośrednie..

Następstwem uruchomienia programu STEP-7 Micro jest pojawienie się głównego ekranu programu, który prezentuje rys. 7.16. Z okna tego dostępne są następujące opcje:



Rys. 7.16. Ekran startowy systemu STEP7-Micro/WIN32

Poniżej zamieszczono krótkie opisy poszczególnych elementów programu STEP7-Micro:
Pasek głównego menu (Menu Bar) Pozwala na wykonywanie czynności przy użyciu myszki bądź klawiatury.
Paski narzędzi (Toolbars) Umożliwiają łatwy dostęp do najczęściej używanych poleceń oprogramowania STEP 7-Micro/WIN 32.
Pasek nawigacji (Navigation Bar) Pogrupowane przyciski odpowiadające za ustawienia specyfiki programowania:
Zakładka **View**—zawiera przyciski umożliwiające wyświetlenie okna edycji programu (Program Block), tabeli symboli (Symbol Table), okna stanu (Status Chart), bloku danych (Data Block), okna pozwalającego na dostosowanie parametrów systemu (System Block), okna z informacjami o elementach programu, użytych instrukcjach i połączeniach w sieci PLC (Cross Reference) oraz parametrów komunikacji ze sterownikiem (Communications).
Zakładka **Tools**— obejmuje dodatkowe narzędzia do tworzenia instrukcji (Instruction Wizard) oraz do oprogramowania zewnętrznego panelu (TD 200 Wizard).

Drzewo instrukcji (Instruction Tree) Wyświetla w postaci zhierarchizowanej wszystkie obiekty oraz instrukcje projektu dostępne w formie LAD, FBD lub STL. Po otwarciu folderu z określoną instrukcją można umieścić ją w oknie edycji programu przy użyciu techniki "drag and drop" bądź przez podwójne kliknięcie w (językach LAD i FBD).
Tabela zmiennych lokalnych (Local Variable Table) Zawiera odnośniki do wejść i wyjść sterownika w postaci utworzonych przez użytkownika zmiennych lokalnych.
Okno edycji programu (Program Editor Window) Zawiera tabelę ze zmiennymi lokalnymi oraz widok programu dla edytora LAD, FBD bądź STL. Po utworzeniu podprogramów (subroutines) i obsługi przerwań (interrupt routines) w programie głównym (OB1), wyświetlany jest u dołu pasek pozwalający na nawigację pomiędzy podprogramami.
Okno zdarzeń (Output Window) Wspiera wyświetlanie informacji podczas kompilacji programu. Po wystąpieniu błędów kompilacji, wystarczy podwójnie kliknąć na określonym komunikacie o błędzie zostanie wyświetlony komunikat w oknie edycji programu.
Pasek stanu (Status Bar) Wyświetla informacje o stanie wykonywanych informacji wykonywanych przez STEP 7-Micro/WIN 32.
Okno stanu (Status Chart Window) Pozwala na prześledzenie stanów wejść/wyjść oraz zmiennych programu umieszczając je w diagramie. Można tworzyć różne diagramy w celu obserwacji elementów z różnych części programu. Każdy diagram stanu ma swoją własną zakładkę w oknie stanu.
Blok danych (Data Block/Data Initializer Window) Umożliwia wyświetlanie oraz edycję zawartości bloku danych.

7.4.5. Pierwszy program w LAD i STL.

Język logiki drabinkowej LAD składa się z powszechnie używanego zestawu symboli, które reprezentują elementy kontroli oraz instrukcje. Wprowadzanie elementów do schematu drabinkowego odbywa się przez umieszczenie kursora w wybranym miejscu obwodu oraz wybór symbolu z drzewa instrukcji i przeniesienie go do obwodu. Następnie należy zaadresować dany element przez w prowadzenie kolejnych znaków adresu i zatwierdzenie klawiszem ENTER.

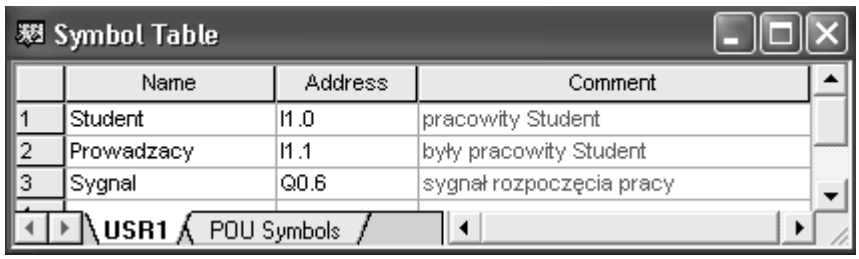
Najprostszy program może realizować sumę logiczną: „*Laboratorium automatyki może odbywać się, gdy stawi się na nie Student i prowadzący.*"(Student AND Prowadzący = zajęcia).

Założenia
W teście wykorzystane zostanie stanowisko laboratoryjne. W dostępnej 'klawiaturze', znajdującej się pod sterownikiem S7-200 pierwszym dwóm przyciskom przyporządkowane są adresy I1.0 oraz I1.1. Z kolei do wyjścia o adresie Q0.6 podłączony jest sygnalizator dźwiękowy.
Każda z osób spełniających warunek konieczny podany w zdaniu logicznym może przycisnąć tylko jeden przycisk.

Edycja programu
W przypadku pierwszego uruchomienia programu STEP-7Micro należy z menu głównego wybrać PLC/Type… i wybrać z pola kombi typ sterownika CPU 214.

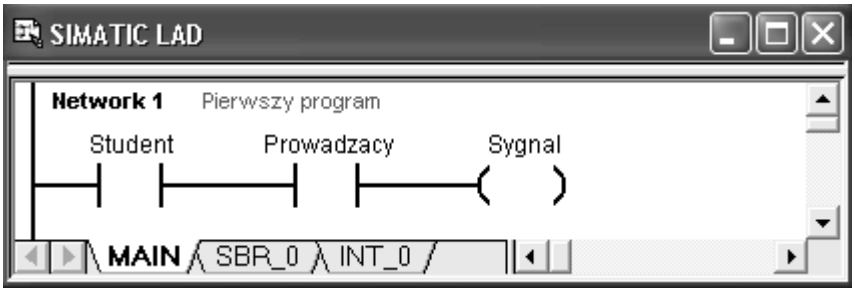
Dla adresowania pośredniego należy wybrać oraz ustalić zawartość tabeli symboli:

- Rozwinąć drzewo instrukcji i wybrać Symbol table/USR1
- Przejść do prawej części ekranu i uzupełnić tabelkę jak na rys 7.17



Rys.7.17. Deklaracje w tabeli symboli

- Wrócić do okna Simatic LAD i ustawić się na początku pierwszego obwodu (Network1),
- Znaleźć w drzewie instrukcji i rozwinąć grupy poleceń Instructions oraz Bit Logic,
- Ustawić się na symbolu styku normalnie otwartego (-| |-) i przeciągnąć go do obwodu,
- Poprzednią czynność powtórzyć, dołączając szeregowo kolejny styk normalnie otwarty,
- Ustawić się na symbolu wyjścia (**-( )**) i przeciągnąć go na koniec obwodu,
- Podwójne kliknięcie na ??.? umożliwi edycję opisu odpowiedniego elementu; należy opisać je według rysunku 7.18



Rys.7.18. Zapis funkcji logicznej AND w języku LAD

- Tak przygotowany program należy skompilować PLC/Compile all i przesłać program do sterownika (uprzednio upewniając się, ze znajduje się on w trybie pracy „STOP") używając kombinacji klawiszy CTRL+D lub przez wybór ikony  .

Testowanie programu

- Przestawić sterownik w tryb pracy „RUN", wybierając z menu głównego PLC/RUN.
- Przeprowadzić test przedstawiony w założeniach programu.

Program zapisany w języku STL – lista instrukcji stanowi zbiór instrukcji zapisanych w kolejnych liniach programu. Do najczęściej używanych instrukcji należą:

LD	- ładuj wartość bitu na stos,
A, O	- operacje logiczne AND, OR,
=	- Przypisanie wartości bitu,
S,R,	- Ustaw, Wyzeruj wartość bitu,
NOP	- Instrukcja pusta (bez znaczenia).

Zaproponowany program w języku STL przyjmie postać:

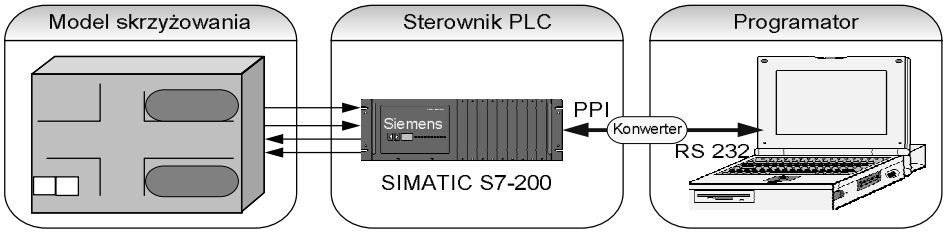
```
LD      Student
A       Prowadzacy
=       Sygnal
```

Wybór z menu głównego View / STL przełączy widok z zapisu w LAD na kod w STL.

7.5. Stanowisko laboratoryjne

Stanowisko laboratoryjne składa się z następujących elementów (rys. 7.19):

- sterownika PLC SIMATIC S7-200 z zasilaczem 24 VDC,
- komputera klasy IBM PC z oprogramowaniem użytkowym STEP7 firmy Siemens,
- modelu sygnalizacji świetlnej skrzyżowania ulicznego.



Rys. 7.19 Elementy składowe stanowiska laboratoryjnego

7.6. Instrukcja wykonania ćwiczenia

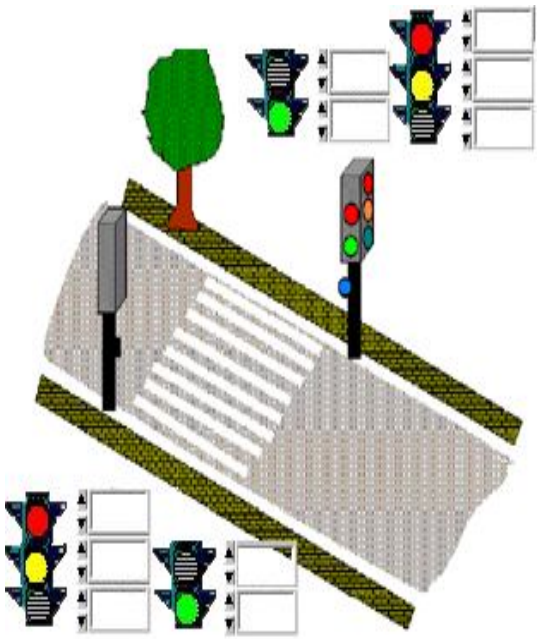
1. Zapoznać się z budową zewnętrzną sterownika PLC.
2. Prześledzić konfigurację połączeń elementów składowych stanowiska laboratoryjnego.
3. Zidentyfikować poszczególne wyjścia modelu.
4. Zapoznać się z oprogramowaniem STEP 7-Micro/WIN:

- wykorzystując wybrane elementy makiety skrzyżowania, zrealizować we wszystkich dostępnych w oprogramowaniu językach (edytorach) podstawowe funkcje logiczne,
- zapisać programy na dysk,
- przesłać programy do sterownika,
- sprawdzić poprawność działania programów.

5. Zrealizować w wybranym języku oprogramowania STEP-7 (np. w domu), program sterujący elementami sygnalizacji świetlnej modelu skrzyżowania (rys. 7.20) i sprawdzić jego działanie.

Układ powinien realizować sterowanie sygnalizacją świetlną typowego przejścia dla pieszych. Powinien składać się z sygnalizacji dla kierowców (światło czerwone, żółte i zielone), z sygnalizacji dla pieszych (światło czerwone i zielone) oraz przycisku żądania. W typowej sytuacji uaktywnione jest światło zielone dla kierowców i czerwone dla pieszych. Po naciśnięciu przycisku następuje zmiana światła dla kierowców z zielonego na żółte a następnie na czerwone, które włącza jednocześnie światło zielone dla pieszych. Po upływie 10 sekund światło powinno zmienić się z ciągłego na impulsowe i tak pozostać przez pięć sekund. Następnie sytuacja powinna się odwrócić.

Jeżeli przed upływem trzydziestu sekund nastąpi ponowne żądanie zmiany układu świateł, to polecenie to zostanie zapamiętane, ale zrealizowane dopiero po dopełnieniu czasu oczekiwania.



Rys.7.20. Model sygnalizacji świetlnej

7.7. Wykonanie sprawozdania

Sprawozdanie powinno zawierać opis stanowiska laboratoryjnego (sprzęt i oprogramowanie) „wzięty z natury", treść postawionych zadań, programy w różnych językach wraz z komentarzami. W sprawozdaniu należy też umieścić techniczną realizację rozwiązanych zadań wraz z uwagami eksploatacyjnymi.

LITERATURA

1. *Simatic S7-200. Podręcznik obsługi systemu sterownikowego. Siemens. W-wa 2006*
2. *K. Grandek, R. Rojek: Mikroprocesorowe sterowniki programowalne. Wyd. WSI, Opole 1991*
3. *T. Legierski, J. Wyrwał, J. Kasprzyk, J. Hajda: Programowanie sterowników PLC. Gliwice 1998*
4. *T. Mikulczyński,Z. Samsonowicz: Automatyzacja dyskretnych procesów produkcyjnych. WNT W-wa 1997*
5. *A. Niederliński: Systemy komputerowe automatyki przemysłowej t 1, 2, WNT 1984.*
6. *A.i J. Król: S5/S7Windows. Programowanie i symulacja sterowników PLC firmy SIEMENS, Nakom, 2000*
7. *T. Seta: Wprowadzenie do zagadnień sterowania, wykorzystanie programowalnych sterowników logicznych PLC. Wyd. MIKOM, W-wa 2002*

Wzór protokołu (lekko zaciemnione pola wypełnia prowadzący)

Laboratorium Podstaw Automatyki					
Temat: Realizacja układów sterowania binarnego na bazie sterownika PLC					Nr: **7**
Grupa:	**Imiona i nazwiska osób:** **1.**	**Podpisy:**	**Data wykonania:**	**Termin:** **[ ] - planowy** **[ ] – odróbkowy**	**Ocena:**
Zespół:	**2.** **3.**		**Data oddania:**	**Opóźnienie:**	**Dzień tygodnia:** **Godz. zajęć:**

Podsumowanie:

L.p.	**Etap**	**Wykonanie**		
		Poprawne	Poprawne, ale z małymi błędami	Z rażącymi błędami lub niewykonane
1.	Realizacja programu nr 1. zadanego przez prowadzącego w jęz. LAD / FBD / STL*			
2.	Realizacja programu nr 2. zadanego przez prowadzącego w jęz. LAD / FBD / STL*			
3.	Realizacja programu nr 3. zadanego przez prowadzącego w jęz. LAD / FBD / STL*			
4.	Realizacja programu nr 4. zadanego przez prowadzącego w jęz. LAD / FBD / STL*			
Uwagi:				

* - niepotrzebne skreślić