





MINISTERSTWO EDUKACJI i NAUKI

Józef Butkowski

Programowanie sterowników PLC 311[50].Z3.01

Poradnik dla ucznia

Wydawca Instytut Technologii Eksploatacji – Państwowy Instytut Badawczy Radom 2005

Recenzenci: mgr inż. Andrzej Rodak mgr inż. Maria Suliga

Opracowanie redakcyjne: mgr inż. Katarzyna Maćkowska

Konsultacja: dr inż. Janusz Figurski

Korekta: mgr Joanna Iwanowska

Poradnik stanowi obudowę dydaktyczną programu jednostki modułowej 311[50].Z3.01. Programowanie Sterowników PLC zawartego w modułowym programie nauczania dla zawodu technik mechatronik.

SPIS TREŚCI

1. Wprowadzenie	3
2. Wymagania wstępne	4
3. Cele kształcenia	5
4. Materiał nauczania	6
4.1. Sterowanie, regulacja, klasyfikacja sterowników PLC	6
4.1.1. Materiał nauczania	6
4.1.2. Pytania sprawdzające	10
4.1.3. Ćwiczenia	10
4.1.4. Sprawdzian postępów	11
4.2. Budowa i zasada działania sterowników PLC	12
4.2.1. Materiał nauczania	12
4.2.2. Pytania sprawdzające	21
4.2.3. Ćwiczenia	21
4.2.4. Sprawdzian postępów	23
4.3. Zasady programowania sterowników PLC, języki	
programowania LAD, STL, FBD	24
4.3.1. Materiał nauczania	24
4.3.2. Pytania sprawdzające	49
4.3.3. Ćwiczenia	49
4.3.4. Sprawdzian postępów	53
4.4. Obsługa programów komputerowych do programowania	
sterowników PLC, uruchamianie i testowanie	
sterowników PLC	54
4.4.1. Materiał nauczania	54
4.4.2. Pytania sprawdzające	60
4.4.3. Ćwiczenia	61
4.4.4. Sprawdzian postępów	64
5. Sprawdzian osiągnięć	65
6. Literatura	68

1. WPROWADZENIE

Poradnik będzie Ci pomocny w przyswajaniu wiedzy o sterownikach PLC, ich programowaniu, zastosowaniu oraz perspektywach rozwoju.

W poradniku zamieszczono:

- wymagania wstępne, wykaz umiejętności, jakie powinieneś mieć już ukształtowane, abyś bez problemów mógł korzystać z poradnika,
- cele kształcenia, wykaz umiejętności, jakie ukształtujesz podczas pracy z poradnikiem,
- materiał nauczania, "pigułkę" wiadomości teoretycznych niezbędnych do opanowania treści jednostki modułowej,
- zestaw pytań przydatny do sprawdzenia, czy już opanowałeś podane treści,
- ćwiczenia pomogą Ci zweryfikować wiadomości teoretyczne oraz ukształtować umiejętności praktyczne,
- sprawdzian osiągnięć, przykładowy zestaw zadań i pytań. Pozytywny wynik sprawdzianu potwierdzi, że dobrze pracowałeś podczas lekcji i że nabrałeś wiedzy i umiejętności z zakresu tej jednostki modułowej,
- literaturę uzupełniającą.

[&]quot;Projekt współfinansowany ze środków Europejskiego Funduszu Społecznego"

2. WYMAGANIA WSTĘPNE

Przystępując do realizacji programu nauczania jednostki modułowej powinieneś umieć:

- stosować jednostki miar układ SI,
- znać zasady działania układów przekaźnikowych i stycznikowych,
- znać podstawy elektrotechniki, elektroniki, techniki cyfrowej, techniki mikroprocesorowej,
- obsługiwać komputer na poziomie podstawowym,
- korzystać z różnych źródeł informacji.

[&]quot;Projekt współfinansowany ze środków Europejskiego Funduszu Społecznego"

3. CELE KSZTAŁCENIA

W wyniku procesu kształcenia powinieneś umieć:

- określić parametry techniczne sterownika PLC,
- zidentyfikować elementy sterownika PLC,
- opisać strukturę sterownika PLC,
- dobrać sterownik do określonego zastosowania,
- zaprojektować graficzny schemat rozwiązania zadania sterowniczego,
- zaadresować wejścia i wyjścia w sterowniku,
- utworzyć program do sterownika w wybranym języku programowania,
- obsłużyć wybrany programator,
- posłużyć się oprogramowaniem specjalistycznym do programowania sterowników PLC,
- przesłać program do sterownika PLC,
- uruchomić i przetestować poprawność działania sterownika PLC.

[&]quot;Projekt współfinansowany ze środków Europejskiego Funduszu Społecznego"

4. MATERIAŁ NAUCZANIA

4.1. Sterowanie, regulacja, klasyfikacja sterowników PLC

4.1.1. Materiał nauczania

Zanim przejdziemy do klasyfikacji sterowników podamy określenia takich pojęć, jak regulacja i sterowanie.

Regulacja jest działaniem, polegającym na takim oddziaływaniu na wielkość regulowaną mierzoną na bieżąco, aby była zbliżona do wielkości zadanej. Rozróżniamy regulację stałowartościową i nadążną. Jeżeli zadaniem regulacji jest utrzymanie danej wielkości na stałym poziomie wartości, to taką regulację nazywamy stałowartościową (np. utrzymanie temperatury na stałym poziomie). W zależności od sposobu oddziaływania na proces, mamy regulację analogową, jeżeli oddziaływanie na proces odbywa się w sposób analogowy; binarną, jeżeli stosuje się działanie binarne; cyfrową, jeżeli stosuje się oddziaływanie cyfrowe.

Proces regulacji przebiega w układzie zamkniętym, w którym sygnały wyjściowe poszczególnych członów są doprowadzane do wejścia układu. Schemat blokowy układu regulacji pokazany jest na rysunku 4.1.1.



Rys. 4.1.1. Schemat blokowy układu regulacji

Sterowaniem nazywa się proces, w którym jedna lub kilka wielkości wejściowych oddziaływa na wielkości wyjściowe. Sposób oddziaływania zależy od właściwości układu sterowania. Sterowanie odbywa się w układzie otwartym, w którym sygnały oddziaływają tylko w jednym kierunku tzn. nie są doprowadzane zwrotnie ani do wejścia ani do żadnego innego miejsca układu (rys. 4.1.2). Każdy układ sterowania składa się z urządzeń wejściowych, urządzenia sterującego, urządzeń wejściowych i obiektu sterowania.



Rys. 4.1.2. Schemat blokowy układu sterowania

Do urządzeń wejścia zalicza się łączniki i przyciski elektryczne oraz różnego rodzaju czujniki (sensory), np. fotokomórki, czujniki temperatury, liczniki, czujniki prędkości obrotowej itp.

Do urządzeń wyjścia zalicza się człony wykonawcze (aktuatory) oddziałujące bezpośrednio na obiekt sterowania. Zaliczamy tu przekaźniki, styczniki, tranzystory mocy, tyrystory, triaki, zawory hydrauliczne, zawory pneumatyczne.

Urządzenie sterujące to ta część układu, która przez człon wykonawczy oddziaływa na obiekt sterowania. W skład urządzenia sterującego wchodzą urządzenia wejścia, człony realizujące funkcje logiczne i urządzenia wyjścia. Człony realizujące funkcje logiczne przetwarzają sygnały wejściowe według zadanego programu.

Cechą charakterystyczną procesu sterowania jest to, że sterowanie odbywa się w układzie otwartym i jego celem jest osiągnięcie określonego stanu końcowego układu, przy czym stan ten nie ma wpływu na proces sterowania urządzenia sterującego i obiektu sterowania. W zależności od sposobu przetwarzania i wykorzystania sygnałów rozróżnia się układy sterowania kombinacyjne i sekwencyjne.

W sterowaniu kombinacyjnym sygnał sterujący powstaje z powiązania (kombinacji) kilku sygnałów. Na przykład: tokarkę można uruchomić tylko wtedy, gdy jest zamknięta osłona zabezpieczająca i detal jest zamocowany w uchwycie. Sterowanie kombinacyjne jest sterowaniem binarnym. Układy kombinacyjne projektuje się w oparciu o algebrę dwuwartościową i przedstawia się ich działanie za pomocą równań tej algebry, schematów połączeń, tablic i schematów działania. W układach sterowania sekwencyjnego poszczególne czynności sterujące odbywają się krok po kroku. Rozpoczęcie kolejnego kroku jest uzależnione od czasu lub stanu procesu. W czasowo-sekwencyjnych układach sterowania wykorzystywane są generatory impulsów, zegary taktujące lub przekaźniki czasowe.

Prostym przykładem sterowania sekwencyjnego o działaniu zależnym od czasu jest układ automatycznego rozruchu "gwiazda/trójkąt" silnika trójfazowego. Na początku silnik jest uruchamiany w układzie "gwiazda", po upływie oszacowanego czasu rozruchu (z uwzględnieniem pewnej rezerwy czasowej) następuje przełączenie w układ "trójkąt" i silnik jest gotowy do normalnej pracy. Sterowanie sekwencyjne przedstawia się w postaci schematu działania. W układach sterowania procesowo-sekwencyjnego przejście do następnego kroku jest powodowane zmianami stanu procesu. W przypadku układu rozruchu silnika trójfazowego potrzebny może być sensor sygnalizujący stan "osiągnięto obroty biegu jałowego". Po zasygnalizowaniu tego stanu następuje automatyczne przełączenie w układ "trójkąt". Sterowanie sekwencyjne zależne od procesu przedstawia się w postaci schematów działania lub diagramów przebiegu programu lub diagramów przemieszczeń, jeżeli kolejne przełączenie zależy od wartości położenia (przemieszczenia).

Sterowanie procesowo-sekwencyjne jest w zasadzie rozwiązaniem lepszym od sterowania czasowo-sekwencyjnego, ponieważ w przypadku zakłóceń przebieg procesu sterowania zostaje przerwany lub biegnie prawidłowo dalej, ale wolniej. Przykładowo, przy bardzo mocno obciążonym silniku trójfazowym przełączenie "w trójkąt" nastąpi dopiero po osiągnięciu wystarczająco wysokich obrotów.

Układy sekwencyjne tworzy się i przedstawia przy pomocy diagramów drogowych (przemieszczeń) lub schematów działania. Te formy przedstawiania układów. w połaczeniu ze wspomagającymi (edycyjnymi) programami komputerowymi, umożliwiają szybkie projektowanie układów sterowania. Niezależnie od formy przedstawienia układu należy na początku procesu projektowania układu sekwencyjnego dokładnie przeanalizować przebieg przeprowadzić automatyzowanego procesu, a więc systematyczną analize i ewentualne określenie poszczególnych stanów. Następnie, w ramach opracowania układu sterowania:

- proces sterowania zostaje podzielony na pojedyncze czynności cząstkowe,

- sporządza się słowny opis pojedynczych czynności,

- formułuje się warunki zabezpieczenia i czynności w stanach awaryjnych,

- sporządza się opis elementów sterowania i przyrządów sygnalizacyjnych,

– ustala się konfigurację przyrządów,

- sporządza się program sterowania,
- sprawdza się poprawność programu sterowania, o ile to możliwe, przy pomocy symulacji komputerowej i ewentualnie następnie koryguje się go,
- zestawia się przyrządy niezbędne dla realizacji programu i realizuje się go jako układ połączeń elektrycznych,
- testuje się układ sterowania bez udziału automatyzowanego procesu, przy czym sygnały pochodzące z procesu są symulowane przyciskami (łącznikami) ręcznymi,
- dokonuje się rozruchu układu sterowania łącznie z automatyzowanym procesem.

Układy sterowania sekwencyjnego projektuje się korzystając z diagramów drogowych (diagramów przemieszczeń), diagramów stanów lub schematów działania. Diagramy drogowe i diagramy stanów pokazują wzajemne oddziaływania elementów składowych i urządzeń wykonawczych automatyzowanego procesu. Środkami przedstawiania są odpowiednie symbole elementów sygnałowych, sterujących i wykonawczych realizowanych przemieszczeń.

Na rys. 4.1.3. przedstawiono spotykane układy sterowania.



Rys. 4.1.3. Układy sterowania

W przypadku układu przekaźnikowo-stycznikowego, po sprecyzowaniu zadania sterowniczego montuje się przekaźniki i styczniki w szafce sterowniczej oraz wykonuje się ich okablowanie (sterowanie konwencjonalne). Przy stosowaniu przełączników wybierakowych lub krzyżowych można w ograniczonym zakresie zmienić zadanie sterownicze. W układach przekaźnikowo-stycznikowych proces sterowania zależy od sposobu połączeń przekaźników i styczników, a w układach z programowalną pamięcią od programu sterowniczego.

W przypadku układów z programowalną pamięcią PLC schemat blokowy układu sterowania pokazany jest na rysunku 4.1.4.



Rys. 4.1.4. Układ sterowania z użyciem sterownika PLC

Sterowniki programowalne PLC (ang. Programmable Logic Controller) są to urządzenia wykonane w technice komputerowej, używane do monitorowania stanu wejść, podejmowania decyzji w oparciu o jego program i uaktualnianie stanu wyjść. Służą one do realizacji procesów sterowania i regulacji. Sposób działania sterownika opracowany jest w formie listy instrukcji tworzących tak zwany program użytkownika. Program ten może być dowolnie zmieniany, poprawiany, modyfikowany bez konieczności zmiany elementów, czy okablowania użytkownika. Tworzenie programu, według którego pracuje sterownik, wykonuje się przy pomocy programatora. Programatory to urządzenia budowane specjalnie do tego celu lub komputery PC wyposażone w oprogramowanie do tworzenia programów pracy sterownika i współpracy ze sterownikiem. Budowane są także sterowniki wyposażone w narzędzia ich programowania. Możliwość programowania sterowników jest ich zasadniczą zaletą. Mają bardzo szeroki zakres funkcji, są także bardzo niezawodne i mają małe rozmiary.

Inne układy automatyki, takie jak przekaźnikowe układy sterowania, czy układy elektroniczne zbudowane z funktorów logicznych, realizują stały program pracy. Program ten jest ustalony doborem elementów i okablowaniem, względnie wykonaniem ścieżek na płytach drukowanych. Niektóre z tych układów dopuszczają pewne zmiany działania umożliwiane stosowaniem zwór, przełączników, lub wymienialnych modułów. Zasadnicza zmiana ich programu pracy wymaga praktycznie nowego doboru elementów i wykonania nowego okablowania. Niezawodność sterowników i ich odporność na wpływy środowiskowe pozwalaja na ich stosowanie w budownictwie okrętowym, na statkach i platformach wiertniczych. Zakres zastosowań sterowników jest bardzo szeroki, od kompleksowej automatyzacji procesów wytwarzania do aplikacji w gospodarstwie domowym. Przykładem zastosowań kompleksowych jest automatyzacja produkcji samochodów. Poszczególne etapy produkcji, od montażu wyłączników oświetlenia do zgrzewania karoserii, lakierowania i magazynowania gotowych samochodów są automatyzowane, koordynowane i nadzorowane przez hierarchiczne struktury pracujących w sieci sterowników programowalnych. Do zastosowań najprostszych, np. sterowania pompy odwadniającej studzienkę osadową lub sterowania bramą garażu stosowane są sterowniki o kilku wejściach i wyjściach. Sterownik programowalny jest wykorzystywany jako część centralna, przetwarzająca, układu sterowania. Sterowniki, o maksymalnej liczbie 300 DI/DO (DI, ang. Discrete Input = wejście dyskretne, dwustanowe; DO, ang. Discrete Output = wyjście dyskretne, dwustanowe) wykonuje się jako kompaktowe (nazywane także mikrosterownikami); sterowniki o liczbie 65500 DI/DO nazywamy średnimi lub dużymi, gdy liczba DI/DO wynosi około 130000. Sterowniki średnie i duże wykonuje się jako modułowe. Na rysunku 4.1.5. pokazano te trzy grupy sterowników.



Rys. 4.1.5. Sterowniki PLC: 1. Sterownik kompaktowy, 2, 3. Sterowniki modułowe

4.1.2. Pytania sprawdzające

Odpowiadając na pytania, sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

- 1. Co to jest sterowanie? Wymień jego części składowe.
- 2. Co to są sensory? Podaj kilka przykładów.
- 3. Co to są aktuatory? Podaj kilka przykładów.
- 4. Co to jest regulacja, jaka jest różnica między sterowaniem a regulacją?
- 5. Jakie znasz układy sterowania, jakie są ich istotne cechy?
- 6. Co to jest sterownik PLC? Podaj klasyfikację sterowników PLC.
- 7. Jakie są części składowe układu sterowania z użyciem sterownika PLC?
- 8. Jakie są istotne różnice między układami sterowania przekaźnikowo-stycznikowymi a układami sterowania z programowalną pamięcią?
- 9. Co to jest sterowanie kombinacyjne, podaj przykłady?
- 10. Co to jest sterowanie sekwencyjne, podaj przykłady?

4.1.3. Ćwiczenia

Ćwiczenie 1

Wymień podstawowe części układu sterowania i omów rolę każdego z nich.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zapoznać się z kartą katalogową urządzenia mechatronicznego,
- 2) rozpoznać i zanotować podstawowe części układu sterowania przy nazwie urządzenia.

Wyposażenie stanowiska pracy:

- karty katalogowe urządzeń mechatronicznych,
- literatura zgodna z punktem 6 Poradnika dla ucznia.

Ćwiczenie 2

Uszereguj urządzenia mechatroniczne wg zastosowanych w nich różnych systemów sterowania.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) wyszukać w materiałach dydaktycznych informacji o rodzajach systemów sterowania,
- 2) zanotować rodzaje systemów,
- 3) wyszukać w katalogach urządzeń informacji, jakie systemy sterowania są w danych urządzeniach zastosowane,
- 4) zanotować przy nazwie urządzenia, jaki system został zastosowany,
- 5) krótko scharakteryzować systemy sterowania, korzystając z literatury.

Wyposażenie stanowiska pracy:

- literatura zgodna z punktem 6 Poradnika dla ucznia,
- karty katalogowe urządzeń mechatronicznych.

Ćwiczenie 3

Dokonaj klasyfikacji sterowników i podaj przykłady aplikacji dla każdej grupy.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) wyszukać w literaturze informacje o klasyfikacji sterowników,
- 2) wyszukać w katalogach różnych firm przykłady aplikacji sterowników,
- 3) wymienić znane mu przykłady użycia sterowników.

Wyposażenie stanowiska pracy:

- karty katalogowe urządzeń,
- literatura zgodna z punktem 6 Poradnika dla ucznia.

4.1.4. Sprawdzian postępów

	Tak	Nie
y potrafisz:	_	
wyjaśnić jaka jest różnica między sterowaniem a regulacją?		
wyjaśnić, co to jest sterownik PLC?		
podać, co decyduje o wyborze sterowników PLC	\square	
do pracy w urządzeniach mechatronicznych?		
omówić układ sterowania ze sterownikiem PLC?	\square	
wyjaśnić, co to są sensory i aktuatory oraz określić ich rolę		
w pracy sterowników?		
wyjaśnić różnice między sterowaniem kombinacyjnym a sterowaniem		
sekwencyjnym?		
wskazać dziedziny gospodarki, gdzie stosuje się sterowniki?		
	y potrafisz: wyjaśnić jaka jest różnica między sterowaniem a regulacją? wyjaśnić, co to jest sterownik PLC? podać, co decyduje o wyborze sterowników PLC do pracy w urządzeniach mechatronicznych? omówić układ sterowania ze sterownikiem PLC? wyjaśnić, co to są sensory i aktuatory oraz określić ich rolę w pracy sterowników? wyjaśnić różnice między sterowaniem kombinacyjnym a sterowaniem sekwencyjnym? wskazać dziedziny gospodarki, gdzie stosuje się sterowniki?	y potrafisz: Image: Construct a między sterowaniem a regulacją? Image: Construct a między sterowaniem a regulacją? wyjaśnić, co to jest sterownik PLC? Image: Construct a między sterowaniem a regulacją? Image: Construct a między sterowników PLC podać, co decyduje o wyborze sterowników PLC Image: Construct a między sterowników PLC? Image: Construct a między sterowania ze sterownikiem PLC? omówić układ sterowania ze sterownikiem PLC? Image: Construct a między sterowaniem prize sterowaniem a sterowaniem sekwencyjnym? wyjaśnić różnice między sterowaniem kombinacyjnym a sterowaniem sekwencyjnym? Image: Construct a między sterowaniem sterowaniem a sterowaniem a między sterowaniem się sterowniki?

4.2. Budowa i zasada działania sterowników PLC

4.2.1. Materiał nauczania

Sterowniki programowalne budowane są w wielu wersjach różniących się w zależności od przewidywanego zastosowania. Niezależnie od tych różnic w każdym sterowniku PLC można wyodrębnić następujące elementy składowe, zapewniające poprawne i funkcjonalne działanie jednostki. Są nimi:

– **Procesor** (CPU=ang. Central Processing Unit), jednostka przetwarzająca, podejmująca decyzję na podstawie programu i obsługująca urządzenia wejścia/wyjścia,

- Zasilacz, układ zasilający jednostkę centralną CPU i wszystkie przyłączone moduły rozszerzające,

- Wejścia/Wyjścia (ang. Input/Output) punkty połączenia PLC z obiektami zewnętrznymi,
- **Interfejs komunikacyjny**, układ z portem umożliwiający połączenie PLC z jednostką programującą (komputerem PC, programatorem),
- Wskaźniki stanu, układy diodowe informujące o trybie pracy CPU i stanie wejść i wyjść.

Wymienione układy stanowią podstawową strukturę sterownika. Ogólna struktura sterownika przedstawiona jest na rys. 4.2.1, zaś rys. 4.2.2 przedstawia płytę czołową sterownika.



Rys. 4.2.1. Struktura sterownika programowalnego PLC



Rys. 4.2.2. Elementy składowe sterownika PLC

Wyróżniamy dwa zasadnicze rozwiązania konstrukcji sterowników (rys. 4.2.3.):

- sterowniki o budowie kompaktowej,
- sterowniki o budowie modułowej.





W sterownikach kompaktowych (rys. 4.2.4.) konfiguracja podstawowa wymienionych wyżej części składowych (zasilacz, procesor CPU oraz moduły wejść/wyjść) znajdują się w jednej obudowie, na jednym elemencie nośnym. W celu zwiększenia liczby wejść/wyjść stosuje się moduły rozszerzające (rys. 4.2.5.). Powstaje w ten sposób sterownik kompaktowo–modułowy. Sterowniki kompaktowe i kompaktowo-modułowe to sterowniki o małych rozmiarach z możliwością obsługi niewielkiej ilości sygnałów sterujących, regulacji ciągłej PID i arytmetyką zmiennoprzecinkową. Są one stosowane do automatyzacji maszyn i urządzeń oraz

[&]quot;Projekt współfinansowany ze środków Europejskiego Funduszu Społecznego"

przeznaczone do tworzenia zdecentralizowanych struktur sterowania dla małych obiektów typu przepompownie, oczyszczalnie ścieków. Sterownik ma budowę modułową dzięki czemu może być łatwo dopasowany do wymagań użytkownika.

Maksymalna liczba we/wy cyfrowych (DI/DO): 248 Maksymalna liczba we/wy analogowych (AI/AO): 28



Rys. 4.2.4. Sterownik o budowie kompaktowej

Konstrukcja mechaniczna sterownika.

Sterownik w zależności od jednostki centralnej posiada różną ilość zintegrowanych wejść/ wyjść binarnych. Jednostki centralne zasilane napięciem 230V AC posiadają zintegrowany zasilacz. Każda jednostka centralna wyposażona jest w zasilacz 24V DC, który może być zastosowany do zasilania modułów rozszerzeń. Poszczególne elementy sterownika montowane są na pasywnej szynie montażowej DIN 35mm.



Rys. 4.2.5. Jednostka centralna z modułem rozszerzającym

Sterowniki o budowie modułowej:

a) do średnich aplikacji (automatyzacja maszyn, linii produkcyjnych i obiektów technologicznych), rys. 4.2.6.



Rys. 4.2.6. Sterownik o budowie modułowej do średnich aplikacji

Maksymalna liczba we/wy cyfr.: 65536 Maksymalna liczba we/wy analog.: 4096 Duży wybór modułów wejścia/wyjścia Możliwość budowy zdecentralizowanych struktur sterowania

Możliwość budowy zdecentralizowanych struktur sterowania

Duza moc obliczeniowa CPU

Możliwość łatwej rozbudowy

Kompletny sterownik składa się z modułu zasilacza, procesora CPU oraz modułów wejść/wyjść i/lub modułów komunikacyjnych i funkcyjnych. Poszczególne elementy sterownika montowane są na szynie DIN.

b) do średnich i dużych aplikacji (chemia, petrochemia, energetyka, transport, przetwórstwo surowców, ochrona środowiska, linie montażowe), rys. 4.2.7,

maks. liczba I/O cyfrowych: 131056/131056

maks. liczba I/O analogowych: 8192/8192

maks. pamięć programu: 4 MB,

komunikacja: MPI, PROFIBUS, Ethernet

bardzo krótki czas wykonania instrukcji, duży wybór procesorów, modułów I/O, opcji komunikacyjnych, modułów funkcyjnych, możliwość pracy wieloprocesorowej, interfejsy do systemów IT i sieci WWW.



Rys. 4.2.7. Sterownik o budowie modułowej do średnich i dużych aplikacji

Niezależnie od typu konstrukcji sterownika wszystkie elementy składowe połączone są magistralami:

- magistralą danych,
- magistralą sterującą,
- magistralą adresową.

Magistrale to szyny zbiorcze do przekazywania danych i rozkazów między elementami struktury sterownika. Przekazywanie danych i działanie magistral sterowane jest przez opisany poniżej układ sterowania.

Działanie sterownika programowalnego polega na cyklicznym przywoływaniu i wykonywaniu ciągu instrukcji tworzących program użytkownika. Zadania te realizuje układ sterowania. Realizację programu opisuje rys. 4.2.8. Po załączeniu zasilania układ sterowania powoduje wyzerowanie liczników, elementów czasowych, pamięci wyników pośrednich i innych elementów oprócz elementów o pamięci trwałej. Odczytane na początku każdego cyklu pracy sygnały wejściowe zostają zapisane do pamięci wejść procesowych (ang. Process Image Input). Instrukcje programu użytkownika załadowane są w uporządkowany sposób do

[&]quot;Projekt współfinansowany ze środków Europejskiego Funduszu Społecznego"

pamięci programu. Są tam ulokowane w kolejności wprowadzenia, w innych rozwiązaniach według numeracji linii programu.

Układ sterowania wybiera kolejne adresy z pamięci programu, przywołuje odpowiadające im rozkazy i dokonuje przetwarzania sygnałów pamiętanych w pamięci wejść procesowych.



Rys. 4.2.8. Realizacja programu przez sterownik PLC

Uzyskane wyniki podawane są do pamięci wyjść procesowych (ang: Process Image Output). Po wykonaniu instrukcji programu użytkownika sterownik sprawdza stan portów i realizuje zadania komunikacji. Kolejnym etapem pracy są testy i diagnostyka układu. Cykl pracy kończy zapisanie wyjść sterownika. Ogólny cykl pracy sterownika podaje rys. 4.2.9.



Rys. 4.2.9. Cykl programowy (ang. = scan cycle) sterownika

[&]quot;Projekt współfinansowany ze środków Europejskiego Funduszu Społecznego"

Po zakończeniu cyklu następuje rozpoczęcie następnego. Cykliczne wykonywanie programu jest charakterystyczną cechą sterownika uwzględnioną w konstrukcji urządzenia. Nie wymaga stosowania instrukcji powtarzania programu. Aktualny tryb pracy sterownika wskazują wskaźniki stanu umieszczone na płycie czołowej (rys. 4.2.10.).



Rys. 4.2.10. Wskaźniki stanu umieszczone na płycie czołowej sterownika

Sterownik może znajdować się w dwóch trybach pracy: START lub STOP, które mogą być wybierane przy użyciu trójpołożeniowego przełącznika STOP/TERM/START umieszczonego pod górną przykrywką sterownika (rys. 4.2.7.). Wyboru trybu pracy można też dokonać przy użyciu programatora, gdy przełącznik znajduje się w położeniu TERM, gdyż tylko wtedy występuje proces komunikacji między PLC, a programatorem. Ustawienie przełącznika w pozycję RUN(START) lub STOP powoduje to, iż po wyłączeniu i ponownym załączeniu zasilania sterownik nie zmienia trybu pracy. Ponadto tryb pracy STOP jest automatycznie wybierany po ponownym zasileniu sterownika, gdy przełącznik był w pozycji TERM.

W trybie pracy STOP można:

- ładować program sterujący do pamięci sterownika,
- przeglądać i zmieniać zawartość rejestrów wewnętrznych sterownika,

- zmieniać parametry konfiguracyjne sterownika.

W trybie pracy RUN, gdy wykonywany jest program sterujący, nie można dokonać próby ładowania programu do sterownika.

Komunikacja sterownika z rządzeniami sterującymi i sterowanymi odbywa się poprzez wejścia i wyjścia sterownika. Sygnały wejściowe są podawane do sterownika poprzez transoptory o wysokiej klasie izolacji. Zapewnia to galwaniczną separację wejść sterownika i zapobiega przedostaniu się do jego wnętrza impulsów zakłócających. Sterownik prawidłowo rozpoznaje sygnały logiczne 0 lub 1 pojawiające się na jego wejściach, jeżeli sygnały te mają odpowiedni poziom. Sygnał logiczny 1 może ulec zafałszowaniu, jeżeli szeregowo z czujnikiem jest włączona zbyt duża rezystancja. Sytuacja taka występuje wtedy, gdy przewód łączący czujnik z wejściem sterownika ma zbyt mały przekrój lub gdy tranzystor wyjściowy czujnika jest niedostatecznie wysterowany. W takich sytuacjach prąd płynący przez transoptor wejściowy sterownika jest zbyt mały i sygnał logiczny 0 ulega zafałszowaniu wtedy, gdy przy sygnale logicznym 0 na wejściu sterownika pojawia się napięcie różne od 0 V. Tak jest wówczas, gdy do wejścia sterownika jest podłączony łącznik lub przewód o uszkodzonej izolacji. Na wyjściu transoptora pojawia się napięcie, którego poziom nie odpowiada sygnałowi logicznemu 0.

[&]quot;Projekt współfinansowany ze środków Europejskiego Funduszu Społecznego"

Sygnał ten jest wówczas błędnie rozpoznawany przez sterownik. Wszystkie wejścia sterownika są wewnątrz dodatkowo zbocznikowane obwodami RC, zabezpieczającymi sterownik przed zakłóceniami impulsowymi (przepięcia w sieci lub urządzeniu). Filtry RC wprowadzają pewne opóźnienie sygnałów, co oznacza, że aby sygnał został rozpoznany przez sterownik, musi utrzymywać się na jego wejściu przez określony czas.

Ze względu na cykliczne opracowywanie programu przez sterownik, czas ten powinien być również dłuższy niż czas trwania jednego cyklu programowego. Jeżeli sygnał wejściowy trwa krócej niż cykl programowy, to nie jest wpisywany do rejestru pośredniego i jest ignorowany przez sterownik. Krótkotrwałe impulsy zakłóceniowe są zatem również ignorowane przez sterownik, o ile chwila ich wystąpienia nie zbiegnie się z odczytem danych z rejestru pośredniego.

Wejścia sterownika są zasilane z własnego źródła napięciowego. Ze względu na to, że zwykle wszystkie transoptory wejściowe mają wspólny biegun ujemny (połączenie wewnątrz sterownika), na zewnątrz sterownika wyprowadza się tylko biegun dodatni źródła, który łączy się poprzez czujniki z wejściami sterownika. Na rys. 4.2.11 pokazano obwód wejściowy sterownika.



Rys. 4.2.11. Obwód wejściowy sterownika

Stopnie wyjściowe sterowników są różnych rodzajów, a mianowicie:

– wyjścia przekaźnikowe

Wyjścia przekaźnikowe zapewniają pełną separację galwaniczną wewnętrznych układów elektronicznych sterownika i obwodów wyjściowych. Przekaźniki są mechanicznymi urządzeniami łączeniowymi o ograniczonej żywotności. Żywotność przekaźników zależy od obciążenia zestyków i wynosi od 500 000 do 3 000 000 cykli łączeniowych. Sterownik o wyjściach przekaźnikowych może sterować odbiornikami o stałym lub przemiennym napięciu znamionowym. Napięcie to nie przekracza zwykle wartości 230 V. W celu ograniczenia niebezpieczeństwa powstawania łuku elektrycznego na zestykach przekaźników, bocznikuje się je wewnątrz sterownika obwodami RC lub warystorami. W przypadku napięcia przemiennego, przy otwartych zestykach przekaźnika przez obwód RC płynie prąd, co może utrudnić sterowanie odbiorników. Dotyczy to głównie odbiorników "wysokoomowych" o niewielkiej mocy znamionowej. Stosując sterowniki z wyjściami przekaźnikowymi w obwodach prądu stałego, należy liczyć się z niebezpieczeństwem uszkodzenia zestyków przez łuki łączeniowe powstając przy wyłączaniu cewek (styczniki, silniki, elektrozawory). W celu ograniczenia tego niebezpieczeństwa równolegle do danej cewki należy włączyć diodę. Na rys. 4.2.12 pokazano sterownik z wyjściami przekaźnikowymi.

[&]quot;Projekt współfinansowany ze środków Europejskiego Funduszu Społecznego"



Rys. 4.2.12. Obwód wyjściowy sterownika z przekaźnikiem

- wyjścia tranzystorowe

Wyjścia tranzystorowe są stosowane tylko w przypadku zasilania odbiorników napięcia stałego. Zwykle napięcie to jest równe napięciu zasilania wewnętrznych obwodów sterownika (np. 24 V). Dopuszczalny prąd tranzystorów jest wystarczająco duży, by wysterować silniki małej mocy, cewki zaworów elektromagnetycznych lub lampy. Do ochrony tranzystorów przed przepięciami powstającymi przy wyłączaniu cewek stosuje się diody. Podczas montażu sterownika należy zwrócić uwagę na biegunowość napięcia stałego podłączanego do jego wyjść.

Sterownik z wyjściami tranzystorowymi pokazano na rys. 4.2.13.



Rys. 4.2.13. Sterownik z wyjściami tranzystorowymi

– wyjścia z triakami

Wyjścia z triakami stosuje się tylko w przypadku zasilania odbiorników prądu przemiennego, napięcie zasilania ma wówczas zwykle wartość 230V AC. Umieszczenie triaków wewnątrz

sterownika pogarsza warunki chłodzenia, a to ogranicza dopuszczalny prąd triaka. Prąd ten jest wystarczająco duży, by wysterować cewki styczników i zaworów elektromagnetycznych.

Triaki zabezpiecza się przed przepięciami obwodami RC lub warystorami. Transoptorowe sterowanie triaków gwarantuje galwaniczną separację wewnętrznych układów elektronicznych sterownika od obwodów wyjściowych.

Sterownik z triakami na wyjściu pokazano na rys. 4.2.14.



Rys. 4.2.14. Sterownik z triakami na wyjściu

Żeby wybrać sterownik, który zostanie wykorzystany w konkretnej aplikacji należy zapoznać się z podstawowymi danymi technicznymi sterowników. Dane te podaje producent i zawarte są one w kartach katalogowych produkowanych przez niego urządzeń. Przykładową kartę katalogową pokazuje tabela 4.2.1.

Tabela 4	4.2.1. Dane	techniczne	sterowników
----------	-------------	------------	-------------

Sterownik	1	2
liczba wejść/wyjść dyskretnych	8/6	14/10
wyjście impulsowe	brak	2
pamięć programu (EEPROM)	1kB	4kB
pamięć danych (RAM)	512 słów	2000 słów
podtrzymanie pamięci danych (kondensatory)	50 godzin	190 godzin
czas wykonania 1000 instrukcji binarnych	1,3ms	0,8ms
timery	max. 64	max. 128
liczniki	max. 64	max. 128
szybki licznik impulsów (zdarzeń)	1 x 2kHz	1 x 2kHz,
		2 x 7kHz
interfejs komunikacyjny	RS - 485	RS - 485
protokół komunikacyjny	PPI/Freeport	PPI/Freeport
potencjometry analogowe	1	2
zegar czasu rzeczywistego	brak	tak
hasło dostępu	3 poziomy	3 poziomy
wymienna kaseta z pamięcią EEPROM	brak	opcja
wymienna kaseta z baterią zasilającą	brak	opcja
max. liczba modułów rozszeszeń	2	7
zasilanie sterownika	(85 - 264)VAC	(85 - 264)VAC
obwody wejściowe z optoizolacją:		
zasilanie	(15 - 30)VDC	(15 - 30)VDC
obwody wyjściowe (przekaźnikowe):		
zasilanie	(5-30)VDC/250VAC	(530)VDC/250VAC
max. obciążenie	2A	2A
-		

[&]quot;Projekt współfinansowany ze środków Europejskiego Funduszu Społecznego"

4.2.2. Pytania sprawdzające

Odpowiadając na pytania, sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

- 1. Jakie są podstawowe elementy składowe sterownika PLC?
- 2. Jakie znasz rodzaje rozwiązań konstrukcyjnych sterowników?
- 3. Z jakich elementów zbudowany jest sterownik?
- 4. Omów zasadę działania sterownika.
- 5. Jaka jest rola wskaźników znajdujących się na płycie czołowej sterownika?
- 6. Jakie są tryby pracy sterownika i co w każdym z nich jest realizowane?
- 7. Jakie znasz rodzaje stopni wejściowych i wyjściowych sterownika?
- 8. Co to są i jakich informacji dostarczają dane techniczne sterownika?

4.2.3. Ćwiczenia

Ćwiczenie 1

Wymień podstawowe elementy składowe sterownika PLC.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zapoznać się z kartą katalogową sterownika,
- 2) obejrzeć sterownik i wskazać jego części składowe,
- 3) wymienić podstawowe elementy składowe sterownika PLC.

Wyposażenie stanowiska pracy:

- karty katalogowe sterowników
- sterowniki, jakie znajdują się w szkole,
- literatura zgodna z punktem 6 Poradnika dla ucznia.

Ćwiczenie 2

Podaj zastosowania sterowników o budowie kompaktowej i modułowej w mechatronice.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zapoznać się z katalogami sterowników,
- 2) odwiedzić strony internetowe producentów sterowników,
- 3) podać zastosowanie sterowników.

Wyposażenie stanowiska pracy:

- karty katalogowe sterowników,
- stanowisko komputerowe z dostępem do Internetu,
- literatura zgodna z punktem 6 Poradnika dla ucznia.

Ćwiczenie 3

Wymień informacje, które możesz odczytać ze wskaźników znajdujących się na płycie czołowej sterownika.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zapoznać się ze wyglądem płyty czołowej sterownika,
- 2) wymienić nazwy i kolory wskaźników,
- 3) wskazać najistotniejszy ze wskaźników.

Wyposażenie stanowiska pracy:

- karty katalogowe sterowników,
- sterownik PLC,
- literatura zgodna z punktem 6 Poradnika dla ucznia.

Ćwiczenie 4.

Wymień rodzaje wejść i wyjść sterownika i omów ich budowę.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zapoznać się z katalogami sterowników,
- 2) wskazać wejścia i wyjścia sterownika, wskazać interfejs komunikacyjny sterownika i wyjaśnić jego rolę.

Wyposażenie stanowiska pracy:

- karty katalogowe sterowników,
- sterownik PLC,
- literatura zgodna z punktem 6 poradnika dla ucznia.

Ćwiczenie 5

Podaj dane techniczne sterownika znajdującego się w Pracowni Mechatroniki.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

1) zapoznać się z kartami katalogowymi sterowników,

2) odczytać i zanotować niezbędne dane techniczne.

Wyposażenie stanowiska pracy:

- karty katalogowe sterowników,
- sterownik PLC,
- literatura zgodna z punktem 6 Poradnika dla ucznia.

4.2.4. Sprawdzian postępów

Czy	v potrafisz:	Tak	Nie
1)	wymienić podstawowe części sterownika i określić ich rolę w prawidłowej pracy sterownika?		
2)	wymienić rodzaje sygnałów wejściowych i wyjściowych sterownika?		
3)	podać, czym się należy kierować przy wyborze sterownika dla danej aplikacji?		
4)	omówić zasadę działania sterownika?		
5)	podać, jakie są poziomy napięć sterujących wejściowych		
	i wyjściowych sterownika?		
6)	omówić rodzaje wejść i wyjść sterownika?		
7)	korzystać z danych technicznych sterownika?		

[&]quot;Projekt współfinansowany ze środków Europejskiego Funduszu Społecznego"

4.3. Zasady programowania sterowników PLC, języki programowania LAD, STL, FBD

4.3.1. Materiał nauczania.

Zastosowanie sterownika do określonego zadania wymaga jego oprogramowania, czyli przetworzenia przepisu sterowania na program działania sterownika. Program jest to uporządkowany ciąg instrukcji określających operacje wykonywane na wielkościach wejściowych sterownika. Program sterownika tworzony jest przez użytkownika w języku programowania właściwym dla danego sterownika. Do programowania sterowników PLC wykorzystuje się komputery PC lub specjalnie do tego przeznaczone programatory (rys. 4.3.1).



Rys. 4.3.1. Programowanie sterowników PLC z użyciem komputera lub programatora

Wyróżnia się dwie podstawowe grupy języków programowania:

- grupę języków tekstowych,
- grupę języków graficznych.
 - Do grupy języków tekstowych zalicza się:
- języki list instrukcji STL (ang. Statment List),
- języki strukturalne ST (ang. Structured Text).
- Do grupy języków graficznych zalicza się :
- języki schematów drabinkowych LAD (ang. Ladder Diagram),
- języki schematów blokowych FBD (ang. Function Block Diagram).

Oprócz tych grup języków stosowane są powszechnie w technice sterowników programowalnych dwie metody modelowania i programowania sekwencyjnych procesów produkcyjnych: metoda **Grafcet** oraz metoda **SFC** (ang. Sequential Function Chart), nazywana także metodą grafów sekwencji. Obie metody stanowią narzędzie wykorzystywane do tworzenia programów w wymienionych, standardowych językach programowania – właściwie są zorientowanymi graficznie metodami zapisu algorytmu procesu wymagającymi określenia logicznych zależności przyczynowo-skutkowych przebiegu sterowanego procesu. Te podane podziały, nazwy języków i metod opisu nie są jednak aprobowane przez wszystkich producentów.

Do najszerzej stosowanych języków należą języki drabinkowe (LAD), języki list instrukcji (STL) oraz języki schematów blokowych (FDB). Stosowane są w nich powszechnie znane, proste instrukcje i symbole, wykorzystywane także w innych technikach sterowania oraz w językach programowania komputerów. Programowanie w języku schematów drabinkowych jest bardzo podobne do tworzenia schematów stykowo-przekaźnikowych układów sterowania elektrycznego. Programowanie w oparciu o metodę Grafcet i grafy sekwencji (SFC) wychodzi z opisu zadań sterowania sekwencyjnego za pomocą grafów zawierających etapy (kroki) i warunki przejścia (tranzycie) między tymi etapami. Języki list instrukcji są najbardziej uniwersalną grupą języków programowania sterowników PLC.

[&]quot;Projekt współfinansowany ze środków Europejskiego Funduszu Społecznego"

Programowanie drabinkowe LAD

Podstawą schematu drabinkowego LAD jest tradycyjny schemat sterowania stycznikowego.

Schemat LAD jest graficznym rozwiązaniem zadania stycznikowego. Tworząc program w postaci schematu stykowego stosuje się symbole graficzne. Podstawowymi symbolami języka drabinkowego są: styki, cewki, liczniki, timery, itp. Symbole używane w schematach drabinkowych nie są tożsame ze stykami dołączonymi do sterownika. Poniżej pokazane są podstawowe elementy programowania drabinkowego.



Styk normalnie otwarty NO. Jeżeli cewka przekaźnika nie jest zasilona lub wyjście sterujące tym stykiem ma wartość zero, to styk jest rozwarty i nie przenosi napięcia sterującego.

Styk normalnie zamknięty NC. Styk jest zwarty, gdy cewka przekaźnika nie jest zasilona lub wejście sterujące tym stykiem ma wartość zero. Zasilenie cewki lub zmiana wartości wejścia na jeden powoduje rozwarcie styku.



Cewka przekaźnika. Cewka jest realizowana jako bit pamięci, sterujący stanem jakichś styków. Styki te mogą wchodzić w skład innych obwodów programu. Cewki mogą ponadto sterować stanem wyjść sterownika.



Prostokąty reprezentują różne instrukcje lub funkcje pomocnicze, które są wykonywane wtedy, gdy zostaną "zasilone". Typowymi elementami funkcyjnymi są: liczniki czasu (ang. timer), liczniki zdarzeń, oraz inne bloki funkcyjne.

Przykładowy program w języku drabinkowym jest przedstawiony na rys. 4.3.2.



Rys. 4.3.2. Przykład programu napisanego w języku schematów drabinkowych

Część logiczna programu sterującego składa się z umieszczonych jeden pod drugim tzw. szczebli programowych. Przypominają one typowy elektryczny schemat połączeń. W skład szczebla wchodzą: elementy logiczne (styki), przekaźniki, jak i bardziej złożone bloki funkcyjne. Schemat drabinkowy posiada symboliczne źródło zasilania. Zakłada się przepływ sygnału od szyny umieszczonej po lewej stronie schematu do przekaźników lub bloków funkcyjnych umieszczonych po prawej stronie danego szczebla. Kolejne szczeble drabiny odczytywane są kolejno od góry do dołu. Po dojściu do ostatniego szczebla proces śledzenia programu rozpoczyna się od początku. Szczebel drabiny logicznej (oznaczany w programie jako Network) musi posiadać odpowiedni format i składnię. Ostatnim elementem szeregowego połączenia w danym szczeblu musi być jeden z przekaźników lub blok funkcyjny, szczebel musi zawierać przynajmniej jeden styk przed wystąpieniem przekaźnika, bloku funkcyjnego lub połączenia pionowego, nie może wystąpić rozgałęzienie mające początek lub koniec wewnątrz innego odgałęzienia.

W przykładzie na rys. 4.3.3.a rozgałęzienie (linia zawierająca styk I0.5) bierze początek w niewłaściwym miejscu szczebla (wewnątrz innego odgałęzienia) a na rys 4.3.3.b styk I0.5 jest nieprawidłowo połączony z wnętrzem odgałęzienia zawierającego styki I0.2 i I0.3.



Rys. 4.3.3. Przykład błędnie napisanego programu

Przy tworzeniu programu sterującego posługujemy się identyfikatorami dla określenia zmiennych występujących w programie. W poniższej tabeli 4.3.1 podane są identyfikatory powszechnie używane w pisaniu programów, czy to w języku LAD, czy to w języku STL, czy FBD:

	Identyfikator
Oznaczenie	Nazwa
Ι	zmienna wejściowa
Q	zmienna wyjściowa
М	wewnętrzna zmienna dyskretna
SM	wewnętrzna zmienna specjalna
V	zmienne pamięciowe
Т	timer
С	licznik
AI	zmienna wejściowa analogowa
AQ	zmienna wyjściowa analogowa
AC	akumulator
HC	szybki licznik
K	stała

Tabela 4.3.1. Identyfikatory używane w pisaniu programów

Przy opracowywaniu programów sterujących wykorzystuje się zmienne dyskretne (bitowe) oraz rejestrowe (wielobitowe). Ma to również odzwierciedlenie w sposobie oznaczania identyfikatorów. W poniższej tabeli 4.3.2 przedstawiono zmienne rejestrowe oraz ich reprezentację w systemie dziesiętnym i heksadecymalnym (szesnastkowym):

	Liczba całkowita bez znaku		Liczba całkowita ze znakiem		
Zmienna rejestrowa	#10	#16	#10	#16	
B (bajt – zawiera 8 bitów)	0 do 255	0 do FF	-128 do +127	80 do 7F	
W (słowo – zawiera 16 bitów)	0 do 65535	0 do FFFF	-32.768 do +32.767	8000 do 7FFF	
D (podwójne słowo – zawiera 32 bity	0 do 4.294.967.295	0 do FFFF FFFF	-2.147.483.648 do +2.147.483.647	8000 0000 do 7FFF FFFF	

Tabela 4.3.2. Zmienne rejestrowane i ich reprezentacje w systemie dziesiętnym i szesnastkowym

Przy pisaniu programów obok oznaczenia literowego identyfikatora należy podać odpowiednią cyfrę (liczbę) określającą miejsce (adres) pamięci CPU, w którym przypisana mu zmienna będzie umieszczona. W przypadku określania miejsca w przestrzeni adresowej dla zmiennej bitowej (dyskretnej) podaje się najpierw numer (adres) bajtu, a następnie po kropce numer wybranego bitu, rys. 4.3.4:



13.4 zmienna wejściowa bitowa o numerze 4 wybrana z bajtu o adresie 3, czyli pierwsza cyfra po identyfikatorze oznacza numer bajtu, a druga numer bitu w bajcie.

MSB – bit najbardziej znaczący

LSB – bit najmniej znaczący

Rys. 4.3.4. Przestrzeń adresowa dla zmiennych wejściowych

W przypadku określania miejsca w przestrzeni adresowej dla zmiennych rejestrowych (bajt – B, słowo – W, podwójne słowo – D) obok identyfikatora podaje się adres najbardziej znaczącego bajtu (danej 8 – bitowej). Mniej znaczące bajty zajmują kolejne bajty przestrzeni adresowej. Poniższe przykłady ilustrują rozmieszczenie zmiennych w przestrzeni adresowej CPU w przypadku tego samego adresu:



VB 100 – zmienna 8 bitowa [VB] o adresie 100.

	MSB			LSB
	15	8	7	0
VW100	VB100		VB101	
	bajt najbardziej znaczący		bajt najmniej znaczący	

VW 100 – zmienna 16 bitowa [VW] o adresie 100.

	MSB				LSB
	31	24 23	16 15	8 7	0
VD100	VB100	VB101	VB102	VB103	
	bait naibar	dziei znaczacy	bait naimnie	ei znaczacy	

VD 100 – zmienna 32 bitowa [VD] o adresie 100.

Język programowania STL.

Jest to zapis symboliczny stanowiący ciąg kolejno ułożonych jedna pod drugą instrukcji (rozkazów). Jest to zapis bardziej zwarty operujący na skrótach literowych symbolizujących np.: wejścia/wyjścia sterownika, zmienne wewnętrzne, operacje logiczne i matematyczne oraz inne bloki funkcyjne. Rozkaz jest najmniejszą jednostką programu i składa się z operacji i operandu. Operacja określa rodzaj czynności jaka w danym rozkazie ma być wykonana, natomiast operand składa się z oznaczenia i parametru. Oznaczeniem jest skrót literowy określonej zmiennej binarnej, rejestru lub innego bloku funkcyjnego, a parametrem jest adres zmiennej (zmiennych) podlegającej danej operacji w bieżącym rozkazie, wykonywanym przez sterownik, programu sterującego. Na rys. 4.3.5. pokazano budowę i sposób zapisu instrukcji w STL.



Rys. 4.3.5. Budowa i zapis rozkazów w STL

Sterownik mikroprocesorowy wykonując dany program analizuje każdy pojedynczy rozkaz osobno krok po kroku w kolejności ich występowania począwszy od rozkazu pierwszego, a skończywszy na ostatnim (z góry na dół). Po realizacji ostatniego rozkazu mikroprocesor rozpoczyna proces analizy od początku. Proces ten nazywa się cyklicznym wykonywaniem programu, a czas potrzebny na przeanalizowanie wszystkich rozkazów danego programu nazywa się czasem jednego cyklu. Na rys. 4.3.6. pokazano program zapisany w języku STL, a obok ten sam program zapisany w języku drabinkowym LAD.



Rys. 4.3.6. Program sterujący zapisany w języku STL i LAD

Język schematów blokowych FBD

Rozwiązanie zadania automatyzacji w formie schematu układu logicznego zbudowanego z funktorów logicznych nosi nazwę schematu funkcyjnego (ang.: FBD Function Block Diagram). Na rys. 4.3.7. pokazano program sterujący napisany w języku FBD. Jest to ten sam program w wersji STL i LAD pokazany na rys. 4.3.6.



Rys. 4.3.7. Program sterujący zapisany w języku FBD

Wybór języka, w którym programista będzie pisał program sterujący zależy od niego samego. Należy podkreślić, że program tworzony w formie schematu drabinkowego LAD lub schematu funkcyjnego FBD jest natychmiast przetwarzany przez programator na listę instrukcji STL.

Budowa programu

Każde zadanie związane z automatyzacją traktowane jest jako projekt. Pierwszym zadaniem jest wybór sprzętu, jego kompletacja i konfiguracja. W przypadku sterownika w wykonaniu kompaktowym wybieramy sterownik odpowiedni do projektu; w przypadku sterownika o budowie modułowej należy wybrać zasilacz, moduł jednostki centralnej oraz moduły dyskretnych wejść i wyjść, moduły wejść/wyjść analogowych itd. składające się na

sterownik PLC (rys. 4.3.8). Skonfigurowanie sprzętu zamyka etap tworzenia struktury sprzętowej układu sterowania. Kolejnym etapem jest wybranie sposobu pisania programu użytkowego. Korzystne dla użytkownika jest adresowanie symboliczne zmiennych (rys. 4.3.9), gdyż łatwiej jest rozpoznać konkretne nazwy niż podobne do siebie ciągi cyfr adresowania absolutnego, np. I 0.0 oraz I 0.4.



Rys. 4.3.8. Przykład konfigurowania sprzętu sterownika o budowie modułowej

1	- Zbiornik - [Symbol Table]					
.	File Edit View PLC Deb	ug Tools	Windows Help			
]] *C) 🛎 🕼 🖨 🕼 🐇	ĥ ∎ ∽	- 🗹 🔯 📥 ᆂ 24 21 🔍 (#) 📗 🕨 = 🎘 🎘 💭			
	Name	Address	Comment			
1	Start_silnika_pompy_1	10.0	Silnik pompy napełniającej zbiornik			
2	Otwarcie_zaworu_nap	Q0.0	Napełnianie zbiornika			
3	3 Czujnik_maks_poziomu 10.2		Zbiornik napełniony			
4	Otwarcie_zaworu_spust	Q0.2	Opróznianie zbiornika			
5	Czujnik_min_poz	10.5	Zbiornik opróżniony			

Rys. 4.3.9. Tworzenie nazw symbolicznych - przykład

Zanim przejdziemy do programowania podstawowych funkcji logicznych tytułem wprowadzenia w programowanie zostanie podane kilka prostych przykładów.

Przyjmujemy, że jeżeli element układu jest w stanie pasywnym (przycisk jest nie wciśnięty, lampka nie świeci, stycznik nie działa itp.), to przypisujemy mu wartość logiczną "0", natomiast jeżeli element układu jest w stanie aktywnym (przycisk jest wciśnięty, lampka świeci itp.) to przypisujemy mu wartość logiczną "1".

Przykład 1

Lampka sygnalizacyjna H1 powinna zaświecić, jeżeli przycisk zwierny S1 zostanie wciśnięty i świecić tak długo, jak długo przycisk ten jest wciśnięty.

Napiszemy listę przyporządkowującą (adresowanie symboliczne).

Przycisk zwierny S1 – I0.0 Lampka sygnalizacyjna H1 –Q0.0 Schemat sterowania stykowego pokazuje rys. 4.3.10, a schemat połączeń zacisków sterownika rys. 4.3.11.





Rys. 4.3.10. Schemat sterowania stykowego

Rys. 4.3.11. Schemat połączeń zacisków sterownika

Program sterujący pokazano na rys. 4.3.12.

1.10.0	Q	0.0	τD	το ο		Q0.0
LI L		<u>٦</u>	- 10	10.0	10.0	
пг	、		=	00.0	10.0-	=
1	•	•		20.0	-	

Rys. 4.3.12. Schemat LAD, STL i FBD

Jeżeli S1= 0 (przycisk S1 jest nie wciśnięty), I0.0 = 0 (brak sygnału na wejściu I0.0), to Q0.0 = 0 i lampka H1 nie świeci (H1=0); jeżeli S1 =1 (przycisk S1 jest wciśnięty) na wejściu I0.0 pojawia się 1, to I0.0 =1, a w następstwie tego Q0.0 = 1; gdy wyjście sterownika jest ustawione na "1" zapala się lampka H1 (H1 = 1).

Przykład 2

Lampka sygnalizacyjna powinna świecić przy nie wciśniętym przycisku zwiernym S1.

Napiszemy listę przyporządkowującą (adresowanie symboliczne).

Przycisk zwierny S1 – I0.0

Lampka sygnalizacyjna H1 – Q0.0

Schemat sterowania stycznikowego pokazuje rys. 4.3.13, a schemat połączeń zacisków sterownika rys. 4.3.14.





Rys. 4.3.13. Schemat sterowania stycznikowego

Rys. 4.3.14. Schemat połączeń zacisków sterownika

Program sterujący pokazano na rys. 4.3.15.

"Projekt współfinansowany ze środków Europejskiego Funduszu Społecznego"

$$\begin{array}{c} & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & &$$

Rys. 4.3.15. Schemat LAD, STL i FBD

Jeżeli S1= 0 (przycisk S1 jest nie wciśnięty), I0.0 = 0 (brak sygnału na wejściu I0.0), to Q0.0 = 0 i lampka H1 świeci (H1=1); jeżeli S1 =1 (przycisk S1 jest wciśnięty) na wejściu I0.0 pojawia się 1, to I0.0 =1, a w następstwie tego Q0.0 = 1; gdy wyjście sterownika jest ustawione na "1" lampka H1 gaśnie (H1 = 0).

Przykład 3

Lampka sygnalizacyjna H1 powinna świecić przy nie wciśniętym przycisku rozwiernym.

Napiszemy listę przyporządkowującą (adresowanie symboliczne). Przycisk rozwierny S1 – I0.0 Lampka sygnalizacyjna H1 – Q0.0

Schemat sterowania stykowego pokazuje rys. 4.3.16, a schemat połączeń zacisków sterownika rys. 4.3.17.





Rys. 4.3.16. Schemat sterowania stykowego



Program sterujący pokazano na rys. 4.3.18.

Rys. 4.3.18. Schemat LAD, STL i FBD

Jeżeli S1= 0 (przycisk S1 jest nie wciśnięty), I0.0 = 1 (jest sygnał na wejściu I0.0), to Q0.0 = 1 i lampka H1 świeci (H1=1); jeżeli S1 =1 (przycisk S1 jest wciśnięty) na wejściu I0.0 pojawia się 0, to I0.0 =0, a w następstwie tego Q0.0 = 0; gdy wyjście sterownika jest ustawione na "0" lampka H1 gaśnie (H1 = 0).

Przykład 4

Lampka sygnalizacyjna H1 powinna zaświecić po wciśnięciu przycisku rozwiernego S1.

Napiszemy listę przyporządkowującą (adresowanie symboliczne). Przycisk rozwierny S1 – I0.0 Lampka sygnalizacyjna H1 – Q0.0

Schemat sterowania stycznikowego pokazuje rys. 4.3.19, a schemat połączeń zacisków sterownika rys. 4.3.20.



S1E-7 I0.0 Wejścia +24 V Wyjścia L1 Q0.0 H1 × N

Rys. 4.3.19. Schemat sterowania stycznikowego

Rys. 4.3.20. Schemat połączeń zacisków sterownika

Program sterujący pokazano na rys. 4.3.21.

$$\begin{array}{c} & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & &$$

Rys. 4.3.21. Schemat LAD, STL i FBD

Jeżeli S1= 0 (przycisk S1 jest nie wciśnięty), I0.0 = 1 (jest sygnał na wejściu I0.0), to Q0.0 = 1 i lampka H1 świeci (H1=1); jeżeli S1 =1 (przycisk S1 jest wciśnięty) na wejściu I0.0 pojawia się 0, to I0.0 =0, a w następstwie tego Q0.0 = 0; gdy wyjście sterownika jest ustawione na "0" lampka H1 gaśnie (H1 = 0).

Przykład 5

Po wciśnięciu przycisku zwiernego S1 powinny się zaświecić jednocześnie trzy lampki H1, H2, H3 i świecić tak długo, jak długo przycisk ten jest wciśnięty.

Napiszemy listę przyporządkowującą (adresowanie symboliczne). Przycisk zwierny S1 – I0.0 Lampka sygnalizacyjna H1 – Q0.0 Lampka sygnalizacyjna H2 – Q0.2 Lampka sygnalizacyjna H3 – Q0.3

Schemat sterowania stykowego pokazuje rys. 4.3.22, a schemat połączeń zacisków sterownika rys. 4.3.23.



Rys. 4.3.22. Schemat sterowania stykowego



Rys. 4.3.23. Schemat połączeń zacisków sterownika

Program sterujący pokazano na rys. 4.3.24.



Rys. 4.3.24. Schemat LAD, STL i FBD

Jeżeli S1= 0 (przycisk S1 jest nie wciśnięty), I0.0 = 0 (brak sygnału na wejściu I0.0), to Q0.0 = 0, Q0.1=0, Q0.2=0 i lampki H1, H2, H3 nie świecą (H1=0, H2=0, H3=0); jeżeli S1 = 1 (przycisk S1 jest wciśnięty) na wejściu I0.0 pojawia się 1, to I0.0 = 1, a w następstwie tego Q0.0 = 1, Q0.1=1, Q0.2=1 (wyjścia sterownika są ustawione na 1), a więc lampki H1,H2, H3 świecą (H1=1, H2=1, H3=1) dopóki S2 jest wciśnięty.

W przedstawionych przykładach sterowniczych wyjścia sterownika były włączone bez samopodtrzymania (wyjście było włączone dopóty, dopóki był spełniony warunek włączenia). Jednak nieraz zachodzi potrzeba, aby wyjście sterownika było włączone pomimo zmiany warunku włączenia. Wyjścia tego typu są wyjściami bistabilnymi, charakteryzującymi się dwoma stabilnymi stanami spoczynkowymi. Przechodzenie z jednego stanu spoczynkowego do drugiego i odwrotnie odbywa się za pomocą dwóch rozkazów: Set-zapisz, ustaw i Reset-skasuj, wyzeruj. Działanie tych instrukcji wyjaśnimy na przykładzie.

Przykład 6

Lampka sygnalizacyjna H1 powinna zaświecić, jeżeli przycisk zwierny S1 zostanie wciśnięty i świecić po jego zwolnieniu. Zgaszenie lampki powinno nastąpić tylko wtedy, gdy zostanie wciśnięty przycisk zwierny S0.

Napiszemy listę przyporządkowującą (adresowanie symboliczne).

Przycisk rozwierny S0 – I0.0

Przycisk zwierny S1 – I0.1

Lampka sygnalizacyjna H1 – Q0.0

Schemat sterowania stycznikowego pokazuje rys. 4.3.25, a schemat połączeń zacisków sterownika rys. 4.3.26.



Rys. 4.3.25. Schemat sterowania stycznikowego

Rys. 4.3.26. Schemat połączeń zacisków sterownika

Program sterujący z instrukcjami Set i Reset pokazano na rys. 4.3.27.



Rys. 4.3.27. Schemat LAD, STL i FBD, sposób 1

Ten program można zapisać w inny sposób, bez instrukcji Set i Reset (rys. 4.3.28).



Rys. 4.3.28. Schemat LAD, STL i FBD, sposób 2

Programowanie podstawowych funkcji logicznych

Prawie wszystkie zadania sterownicze można zrealizować za pomocą podstawowych funkcji AND, OR, NAND, NOR, NOT, EX-OR. Oprócz wymienionych funkcji omówione zostaną moduły czasowe i liczniki. Założono, że znane są symbole i tabele prawdy wymienionych elementów. Programowanie tych funkcje omówimy przy pomocy przykładów.

Funkcja logiczna AND- koniunkcja, iloczyn logiczny Przykład

Lampka sygnalizacyjna H1 powinna zaświecić się po naciśnięciu przycisków zwiernych S1 i S2.

Napiszemy listę przyporządkowującą (adresowanie symboliczne).

Przycisk zwierny S1 – I0.0 Przycisk zwierny S2 – I0.1 Lampka sygnalizacyjna H1 – Q0.0 Schemat sterowania stykowego pokazuje rys. 4.3.29, a schemat połączeń zacisków sterownika rys. 4.3.30.



Rys. 4.3.30. Schemat połączeń zacisków sterownika

-Q0.0

Program sterujący realizujący funkcję AND przedstawia rys. 4.3.31.



Rys. 4.3.31. Schemat LAD, STL i FBD realizujący funkcję AND

10.1

Funkcja logiczna OR-lub, alternatywa Przykład

Lampka sygnalizacyjna H1 powinna zaświecić się po naciśnięciu przycisku zwiernego S1 lub przycisku zwiernego S2.

LD IO.O

Napiszemy listę przyporządkowującą (adresowanie symboliczne).

Przycisk zwierny S1 – I0.0 Przycisk zwierny S2 – I0.1 Lampka sygnalizacyjna H1 – Q0.0

Schemat sterowania stykowego pokazuje rys. 4.3.32, a schemat połączeń zacisków sterownika rys. 4.3.33.



S1 8

S2 E

L1









Program sterujący realizujący funkcję OR przedstawia rys. 4.3.34.



Rys. 4.3.34. Schemat LAD, STL i FBD realizujący funkcję OR

Funkcja logiczna NAND- negacja iloczynu NIE-I. Przykład.

Lampka sygnalizacyjna H1 powinna zgasnąć, przy wciśniętych przyciskach rozwiernych S1 i S2.

Napiszemy listę przyporządkowującą (adresowanie symboliczne).

Przycisk rozwierny S1 – I0.0 Przycisk rozwierny S2 – I0.1 Lampka sygnalizacyjna H1 – Q0.0

Schemat sterowania stycznikowego pokazuje rys. 4.3.35, a schemat połączeń zacisków sterownika rys. 4.3.36.





Rys. 4.3.35. Schemat sterowania stycznikowego

Rys. 4.3.36. Schemat połączeń zacisków sterownika

Program sterujący realizujący funkcję NAND przedstawia rys. 4.3.37.



Rys. 4.3.37. Schemat LAD, STL i FBD realizujący funkcję NAND

Funkcja logiczna NOR-NIE-LUB Przykład

Lampka sygnalizacyjna H1 powinna świecić przy nie wciśniętych przyciskach rozwiernych S1 i S2.

Napiszemy listę przyporządkowującą (adresowanie symboliczne).

Przycisk rozwierny S1	– I0.0
Przycisk rozwierny S2	- I0.1
Lampka sygnalizacyjna H	1 – Q0.0

Schemat sterowania stykowego pokazuje rys 4.3.38, a schemat połączeń zacisków sterownika rys. 4.3.39.





Rys. 4.3.38. Schemat sterowania stykowego

Rys. 4.3.39. Schemat połączeń zacisków sterownika

Program sterujący realizujący funkcję NOR przedstawia rys. 4.3.40.



Rys. 4.3.40. Schemat LAD, STL i FBD realizujący funkcję NOR

Funkcja logiczna X-OR, z wyłączeniem OR (LUB) Przykład

Lampka sygnalizacyjna H1 powinna świecić przy wciśniętym przycisku zwiernym S1 lub przy wciśniętym przycisku zwiernym S2.

Napiszemy listę przyporządkowującą (adresowanie symboliczne).

Przycisk zwierny S1 – I0.0 Przycisk zwierny S2 – I0.1 Lampka sygnalizacyjna H1 – Q0.0

Schemat sterowania stycznikowego pokazuje rys 4.3.41, a schemat połączeń zacisków sterownika rys. 4.3.42.





Rys. 4.3.41. Schemat sterowania stycznikowego

Rys. 4.3.42. Schemat połączeń zacisków sterownika

Program sterujący realizujący funkcję NOR przedstawia rys. 4.3.43.



Rys. 4.3.43. Schemat LAD, STL i FBD realizujący funkcję X-OR

Funkcja logiczna NOT- negacja, inwerter Przykład

Lampka sygnalizacyjna H1 powinna zgasnąć przy wciśniętym przycisku rozwiernym S1.

Napiszemy listę przyporządkowującą (adresowanie symboliczne).

Przycisk rozwierny S1 – I0.0 Lampka sygnalizacyjna H1 – Q0.0

Schemat sterowania stykowego pokazuje rys 4.3.44, a schemat połączeń zacisków sterownika rys. 4.3.45.



Rys. 4.3.44. Schemat sterowania stykowego



Program sterujący realizujący funkcję NOT przedstawia rys. 4.3.46.



Rys. 4.3.46. Schemat LAD, STL i FBD realizujący funkcję NOT

Timery (moduły czasowe)

- Timery sterowników PLC są programowane do:
- tworzenia impulsu,
- tworzenia impulsu z przedłużeniem,
- załączania z opóźnieniem TON (On-Delay Timer),
- załączania z opóźnieniem z podtrzymaniem TONR (On-Delay Timer Retentive),
- wyłączania z opóźnieniem TOF (Off-Delay Timer).

Timer TOF zostaje uruchomiony, jeżeli na jego wejściu (IN) wynik operacji logicznej zmienia się z 1 na 0 (tylne zbocze, opadające zbocze); pozostałe timery, TON i TONR są

uruchamiane, jeżeli na wejściu (IN) wynik operacji logicznej zmienia się z 0 na 1 (przednie zbocze, zbocze narastające).

Element czasowy TON (Timer On–Delay)



Opóźnione załączenie. Timer zlicza jednostki czasu, gdy do jego wejścia IN zostanie doprowadzony sygnał IN = 1, a jest zerowany wtedy, gdy sygnał IN = 0. Po ponownym pojawieniu się sygnału IN = 1 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, co przy rozdzielczości odmierzanego czasu 1ms, 10ms, 100ms daje przedział czasu od 0-3276,7s, czyli około 1 godziny.

Przykład

Lampka sygnalizacyjna powinna zapalić się po czasie t = 1s od naciśnięcia przycisku bistabilnego dołączonego do wejścia I0.0.

Rysunek 4.3.46 przedstawia program w języku LAD i STL, a rys.4.3.47 wykres czasowy działania timera TON.



Rys. 4.3.46. Załączenie wyjścia sterownika z opóźnieniem 1s przez timer TON



Rys. 4.3.47. Wykres czasowy działania timera TON

Element czasowy TONR (Timer Retentive On–Delay)



Opóźnione załączenie z podtrzymaniem. Timer z podtrzymaniem różni się od poprzedniego tym, że sygnał wejściowy IN = 0 nie zeruje zawartości licznika, tylko zawiesza zliczanie, które jest kontynuowane w chwili ponownego ustawienia sygnału IN = 1. Timer mierzy więc sumaryczny czas trwania sygnału IN = 1. Timer z podtrzymaniem można wyzerować za pomocą instrukcji 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.

Przykład

Lampka sygnalizacyjna powinna zapalić się, jeżeli czas pracy urządzenia wyniesie t = 10s.

Rysunek 4.3.48 przedstawia program w języku LAD i STL, a rys. 4.3.49 wykres czasowy działania timera TONR.



Rys. 4.3.48. Pomiar łącznego czasu pracy urządzenia przez timer TONR



Rys. 4.3.49. Wykres czasowy działania timera TONR

Element czasowy TOF (Timer Off–Delay)



Timer zwiera swój styk wyjściowy, oznaczony tą samą nazwą co nazwa timera, jeżeli do wejścia IN zostanie doprowadzony sygnał IN = 1. Gdy sygnał na wejściu IN=0 timer zaczyna zliczać jednostki czasu do wartości określonej przez stałą podaną na wejście PT. Maksymalny zakres zliczania wynosi 32767 jednostek czasu, co przy rozdzielczości odmierzanego czasu 1ms, 10ms, 100ms daje przedział czasu od 0–3276,7s, czyli około 1 godziny.

Przykład

Lampka sygnalizacyjna powinna zaświecić się po naciśnięciu przycisku zwiernego i zgasnąć po 10s od zwolnienia przycisku.

Rysunek 4.3.48 przedstawia program w języku LAD i STL, a rys.4.3.49 wykres czasowy działania timera TOF.





Rys. 4.3.51. Wykres czasowy działania timera TOF

Liczniki (ang. counters)

Są używane do zliczania impulsów podawanych na ich wejścia. Jeżeli impuls podawany na wejście zwiększa stan licznika o 1, to licznik nazywamy licznikiem zliczającym w górę i jego symbol w LAD to CTU; jeżeli impuls podawany na wejście licznika zmniejsza stan licznika o 1, to licznik nazywamy licznikiem zliczającym w dół i jego symbol w LAD to CTD; są jeszcze liczniki, które zliczają impulsy w górę i w dół, oznaczamy je symbolem CTUD.



Licznik zliczający w górę CTU. Zlicza zmiany wartości z 0 na 1 sygnału podanego na wejście CU. Licznik jest zerowany, gdy na wejście kasujące R zostanie podany sygnał o wartości 1. Po doliczeniu do wartości równej stałej podanej na wejściu PV, licznik zwiera swój styk wyjściowy, oznaczony tą samą nazwą, co jego nazwa (Cxx =1). Zakres zliczania: $0 \div 32767$.

Przykład

Lampka sygnalizacyjna powinna zapalić się po zliczeniu 3 impulsów. Rysunek 4.3.52 przedstawia program w języku LAD i STL dla licznika CTU, a rys.4.3.53 wykres czasowy działania licznika CTU.



Rys. 4.3.52. Licznik zliczający w górę CTU



Rys. 4.3.53. Wykres czasowy działania licznika CTU



Licznik zliczający w dół CTD. Zlicza zmiany wartości z 0 na 1 sygnału podanego na wejście CD, zmniejszając stan licznika o jeden (warunkiem rozpoczęcia zliczania jest podanie impulsu na wejście LD). Kiedy stan licznika jest równy zero, licznik przestaje zliczać impulsy na wejściu CD i ustawia swój bit na 1 (zwiera swój styk wyjściowy oznaczony tą samą nazwą, co jego nazwa (Cxx =1)). Zerowanie licznika i ustawienie liczby zliczanych impulsów zgodnej z wartością stałej podanej na wejściu PV, zakres zliczania: 0 \div 32767. Na rys. 4.3.54 podano przykład programu dla licznika CTD, a na rys. 4.3.55 wykres czasowy działania tego licznika.



Rys. 4.3.54. Przykład użycia licznika CD



Rys. 4.3.55. Wykres czasowy działania licznika CTD



Licznik dwukierunkowy. Jego zawartość może zarówno rosnąć, jak i maleć, wskutek zliczania impulsów na jego wejściach. Każda zmiana z 0 na 1 wartości sygnału podanego na wejście CU powoduje zwiększenie zawartości licznika o 1, natomiast zmiana z 0 na 1 sygnału na wejściu CD powoduje zmniejszenie tej zawartości o 1. 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). Licznik można wykorzystać do kontroli pojazdów na parkingu. Na rys. 4.3.56 pokazano przykład programu dla licznika CTUD, a na rys. 4.3.57 wykres czasowy działania tego licznika.



Rys. 4.3.56. Przykład użycia licznika CTUD



Rys. 4.3.57. Wykres czasowy działania licznika CTUD

Programowanie sterowania sekwencyjnego w oparciu o metodę Grafcet

Dla przejrzystości programu PLC i jego rozumienia nie tylko przez autora, opis działania układu sterowania powinien charakteryzować się strukturą niezależną od realizacji sprzętowej i programowej. Dlatego programy sterowania sekwencyjnego dzieli się bloki– etapy, te zaś z kolei na pojedyncze stany, akcje lub obwody – ogólnie działania, pomiędzy którymi określone zostały warunki przejść. Działania stowarzyszone z sobą mogą być łączone w większe jednostki – funkcje programowe. W bloku organizacyjnym określa się kolejność wywoływania poszczególnych bloków i funkcji. Rozróżnia się je przypisując im nazwy tekstowe lub kolejne numery. Dla działań szczególnie skomplikowanych, o charakterze uniwersalnym, producenci dostarczają już gotowe bloki i funkcje – ogólnie podprogramy, które użytkownik tylko wyposaża w odpowiednie parametry i ich wartości. Dla przeprowadzenia testów programów lub ich części producenci dostarczają także sterowniki programowe. Będąc składnikiem oprogramowania, zachowują się –wirtualnie – jak prawdziwe sterowniki. Pozwala to na przeprowadzenie testu programu sterowania on–line na takim sterowniku, zanim jeszcze zostanie skompletowany i skonfigurowany odpowiedni, rzeczywiście istniejący sprzęt.

Blok sekwencji działań zawiera stany (kroki) oznaczone znacznikami ze skojarzonymi z nimi warunkami przejść do następnego stanu (kroku). Te warunki przejść nazywa się tranzycjami (łac. transitus = przejście).

Blok rozkazów zawiera rozkazy – polecenia wysterowania urządzeń wykonawczych sterowanego procesu. Rozkazy te towarzyszą poszczególnym działaniom w programie sterowania. W bloku diagnostyki zdefiniowane są niesekwencyjne zdarzenia i stany awaryjne.

Odpowiednio do wykorzystywanej metody opis sterowanego procesu zorganizowany jest w postaci zorientowanego grafu, w którym występują jako wierzchołki etapy-kroki i przejścia-tranzycje (ang. Transition). Etap przedstawiany jest w postaci kwadratu (rys. 4.3 58). Etap początkowy zaznaczony jest podwójna ramka. W programach, w odpowiednim bloku, programuje się reguły dwa tryby pracy: Ζ prace automatyczna i prace krokowa. Ten jest bardzo przydatny ostatni w sprawdzaniu poprawności programu, zarówno w badaniach wirtualnych, jak i uruchomieniowych rzeczywistego sterownika.



Rys. 4.3.58. Opis sekwencyjnego sterowania metodą Grafcet

Poszczególne kwadraty-etapy połączone

poziomą kreskę, dwoma pionowymi odcinkami: wchodzącym, związanym z poprzednim etapem i wychodzącym, związanym z następnym etapem. Obok symbolu etapu, z reguły w oddzielnym prostokątnym polu, podany jest słowny lub sformalizowany opis działań dotyczących tego etapu. Bezpośrednio przy symbolu tranzycji podany jest także jej opis słowny.

Każdy etap ma swój kolejny numer, etap początkowy oznaczony jest numerem zero. Etap może być aktywny lub nieaktywny – stan ten określa wskaźnik etapu reprezentowany przez zmienną logiczną przyjmującą wartość 1 dla etapu aktywnego lub 0 dla etapu nieaktywnego.

Informacja o stanie etapu przechowywana jest w pamięci.

Etap może być uaktywniony tylko wtedy, gdy etap poprzedni jest aktywny i gdy spełniony jest warunek przejścia-tranzycji (koniunkcja tych dwóch zdarzeń jest równa 1).

Warunki przejść-tranzycji dają się określić na podstawie zależności procesowych, np. został osiągnięty pewien poziom w zbiorniku, lub na podstawie aktualnego trybu pracy sterownika. W trybie pracy automatycznej następuje przechodzenie od etapu do etapu po spełnieniu warunków tranzycji. W trybie pracy krokowej przejście musi zostać dodatkowo potwierdzone przyciskiem kroku panelu sterującego. Etap kolejny staje się nieaktywny, gdy uaktywni się następny w kolejności etap lub wyłączone zostanie przetwarzanie automatyczne programu z wymuszeniem skoku do etapu początkowego lub do innego etapu.

Przykład sterowania krokowego

Program sterowniczy składa się z kroków, które są wykonywane jeden po drugim. W sterowaniu krokowym obowiązują następujące zasady:

- w danej chwili może być włączony tylko jeden krok programowy,
- dany krok włącza się po spełnieniu warunku przełączenia tego kroku przy włączonym kroku poprzednim,
- w zwykłym trybie pracy włączenie danego kroku powoduje automatyczne wyłączenie kroku
- uruchomienie sterowania krokowego (włączenie kroku 1) jest możliwe po spełnieniu warunków startu.

Zadanie

Trzy silniki powinny być włączane kolejno jeden po drugim, przy czym w danej chwili może pracować tylko jeden silnik. Oto poszczególne kroki programu sterowniczego:

Krok 1:

Po spełnieniu warunków startu (wszystkie silniki są wyłączone) i wciśnięciu przycisku S1 (I 0.1) włącza się krok 1 i stycznik K1 (a tym samym silnik M1).

Krok 2:

Po włączeniu silnika M1 i wciśnięciu przycisku S2 (I0.2) (warunek przełączenia kroku 2) włącza się krok 2 i stycznik K2 (a tym samym silnik M2) co powoduje wyłączenie kroku 1 i stycznika K1 (silnika M1).

Krok 3:

Po włączeniu silnika M2 i wciśnięciu przycisku S3 (I 0.3) (warunek przełączenia kroku 3) włącza się krok 3 i stycznik K3 (a tym samym silnik M3), co powoduje wyłączenie kroku 2 i stycznika K2.

Wciśnięcie przycisku S4 "Wyłącz" (I 0.0) zeruje wszystkie kroki programowe, aby silniki znalazły się w sytuacji wyjściowej.

Przebieg programu sterowania został opisany zgodnie z metodą Grafcet i przedstawiony jest na rys. 4.3.59.



Rys. 4.3.59. Programu sterowania opisany zgodnie z metodą Grafcet

[&]quot;Projekt współfinansowany ze środków Europejskiego Funduszu Społecznego"

Po sporządzeniu opisu procesu sterowania powinna zostać sporządzona tabela nazw symbolicznych (lista przyporządkowania), określająca przyporządkowanie wejść poszczególnym łącznikom oraz wyjść – aktuatorom (urządzeniom wykonawczym) (tabela 4.3.3 i rys.4.3.60).



Rys. 4.3.60. Układ połączeń elektrycznych

 Tabela 4.3.3.
 Tabela nazw symbolicznych

Tabela nazw symbolicznych						
Symbol	Adres	Typ d.	Komentarz			
Przycisk załączenie S1 Przycisk załączenie S2 Przycisk załączenie S3 Przeciążenie silnika M1 Przeciążenie silnika M2 Przeciążenie silnika M3 Przycisk wyłączenia S0 Krok 1	IO.1 IO.2 IO.3 IO.4 IO.5 IO.6 IO.0 M1.0 M2.0	BOOL BOOL BOOL BOOL BOOL BOOL BOOL BOOL	Normalnie otwarty Normalnie otwarty Normalnie otwarty Normalnie zamknięty Normalnie zamknięty Normalnie otwarty Silnik1 pracuje			
Krok 2 Krok 3 Stycznik silnika M1(K1) Stycznik silnika M2(K2) Stycznik silnika M3(K3)	M2.0 M3.0 Q0.1 Q0.2 Q0.3	BOOL BOOL BOOL BOOL BOOL	Sılnık 2 pracuje Silnik 3 pracuje Załącza 1 silnik Załącza 2 silnik Załącza3 silnik			

Na podstawie sporządzonego opisu można napisać program sterowania w dowolnym języku. Na rys. 4.3.61 przedstawiony jest program w języku STL ze sposobem adresowania symbolicznego i absolutnego.

Lancu	uch krokowy	Łańcuch	krokowy	
LDN	Silnik_M1	LDN	Q0.1	
AN	Silnik_M2 0	AN	Q0.2	
AN	Silnik_M3	AN	Q0.3	
A	Przycisk_wyłączenia_SO	A	IO.O	
A	Przycisk_załączenia_S1	A	IO.1	
S	Krok_1, 1	s	M1.O,	1
LD	Krok_2	LD	M2.0	
ON	Przycisk_wyłączenia_SO	ON	IO.O	
R	Krok_1, 1	R	M1.O,	1
LD	Krok_1	LD	M1.0	
A	Przycisk_załączenia_S2	A	IO.2	
S	Krok_2, 1	s	M2.0,	1
LD	Krok_3	LD	M3.O	
ON	Przycisk_wyłączenia_S0	ON	IO.O	
R	Krok_2, 1	R	M2.0,	1

LD	Krok 2	LD	M2.0
A	Przycisk załaczenia S3	A	IO.3
s	Krok 3, 1	S	M3.0, 1
LDN	Przycisk wyłaczenia SO	LDN	10.0
R	Krok 3, 1	R	M3.0, 1
Roz	kazv	Rozk	azy
LD	Przeciażenie silnika M1	LD	IO.4
A	Krok 1	A	M1.0
=	Silnik M1	=	Q0.1
LD	Przeciażenie silnika M2	LD	IO.5
A	Krok 2	A	M2.0
LD	Przeciażenie silnika M3	LD	IO.6
A	Krok 3	A	M3.O
=	Silnik M3	=	Q0.3
adres	sowanie symboliczne	adresowa	nie absolutne

Rys. 4.3.61. Program sterowania sekwencyjnego w języku STL

4.3.2. Pytania sprawdzające

Odpowiadając na pytania, sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

- 1. Jakie urządzenia służą do programowania sterowników PLC?
- 2. Jakie są podstawowe grupy języków programowania? Jakie języki są powszechnie stosowane?
- 3. Jakie są podstawowe symbole programowania drabinkowego LAD?
- 4. Jakie są oznaczenia i nazwy zmiennych używanych w pisaniu programów sterujących?
- 5. Co to są zmienne bitowe i wielobitowe, podaj przykłady?
- 6. Jakie liczby można zapisać przy pomocy zmiennych rejestrowych? Podaj przykłady w zapisie dziesiętnym i heksadecymalnym.
- 7. Co oznaczają zapisy: I0.5, T33, C124, VB0, MW100, VD100?
- 8. Czym się różni język programowania STL od języka LAD?
- 9. Jak piszemy program w języku FBD?
- 10. Jakie są etapy tworzenia projektu sterowniczego?
- 11. Jaka jest różnica między adresowaniem absolutnym a adresowaniem symbolicznym sterowników?
- 12. Co to są timery, jakie znasz typy timerów i jak one działają?
- 13. Co to są liczniki zdarzeń, jakie znasz rodzaje liczników, jak one działają?
- 14. Jakie znasz podstawowe funkcje logiczne i jak są one realizowane w językach programowania LAD, STL i FBD?
- 15. Jakie są zasady programowanie sterowania sekwencyjnego w oparciu o metodę Grafcet?

4.3.3. Ćwiczenia

Ćwiczenie 1

Przygotuj stanowisko do programowania sterowników PLC.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) zapoznać się z kartą katalogową sterownika PLC,
- 2) umieć obsługiwać programy komputerowe do programowania sterowników PLC,
- 3) umieć obsługiwać urządzenia zasilające sterowniki PLC.

Wyposażenie stanowiska pracy:

- karty katalogowe sterowników,
- komputer klasy PC lub programator,
- kabel programatora,
- zasilacz stabilizowany napięcia stałego,
- literatura zgodna z punktem 6 Poradnika dla ucznia.

Ćwiczenie 2

Podaj oznaczenia i nazwy zmiennych używanych w pisaniu programów sterujących.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) wypisać oznaczenia literowe zmiennych używanych w programowaniu sterowników,
- 2) napisać obok wypisanych oznaczeń literowych zmiennych ich nazwy,
- 3) podać, które zmienne są wewnętrzne.

Wyposażenie stanowiska pracy:

- karty katalogowe sterowników,
- literatura zgodna z punktem 6 Poradnika dla ucznia.

Ćwiczenie 3

Omów i porównaj sposoby adresowania zmiennych bitowych i rejestrowych.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) omówić i podać przykłady adresowania absolutnego zmiennych procesowych,
- 2) omówić i podać przykłady adresowania symbolicznego zmiennych procesowych,

Wyposażenie stanowiska pracy:

- karty katalogowe sterowników,
- literatura zgodna z punktem 6 Poradnika dla ucznia.

Ćwiczenie 4

Podaj zasady pisania programów sterujących w języku LAD.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) omówić podstawowe elementy programowania drabinkowego LAD,
- 2) narysować symbole podstawowych elementów programowania drabinkowego LAD,
- 3) podać sposoby adresowania zmiennych przypisanych do tych elementów,
- 4) napisać na kartce kilka przykładów pisania programu w języku LAD.

Wyposażenie stanowiska pracy:

- karty katalogowe sterowników,
- literatura zgodna z punktem 6 Poradnika dla ucznia.

Ćwiczenie 5

Podaj zasady pisania programów sterujących w języku STL i napisz program w tym języku.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) wyjaśnić co to jest rozkaz,
- 2) podać z jakich części składa się rozkaz,
- 3) podać przykład rozkazu zapisanego w języku STL i omówić jego budowę,
- 4) wskazać, który z zapisów, LAD czy STL jest dla niego łatwiejszy.

Wyposażenie stanowiska pracy:

- karty katalogowe sterowników,
- literatura zgodna z punktem 6 Poradnika dla ucznia.

Ćwiczenie 6

Podaj zasady pisania programów sterujących w języku FBD i napisz prosty przykład rogramu.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) podać przykłady symboli operacji używanych w języku FBD,
- 2) napisać prosty przykład programu w tym języku,
- 3) porównać ze sobą języki programowania LAD, STL, FBD i powiedzieć, który z nich jest dla niego bardziej przystępny.

Wyposażenie stanowiska pracy:

- karty katalogowe sterowników,
- literatura zgodna z punktem 6 Poradnika dla ucznia.

Ćwiczenie 7

Podaj podstawowe funkcje logiczne i zapisz je w wybranym przez siebie języku programowania.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) narysować schematy elektryczne realizujące podstawowe funkcje logiczne,
- 2) w oparciu o schematy z punktu 1 napisać programy realizujące te funkcje w wybranym języku programowania,
- 3) porównać elementy schematów elektrycznych z elementami programu sterowniczego.

Wyposażenie stanowiska pracy:

- karty katalogowe sterowników,
- literatura zgodna z punktem 6 Poradnika dla ucznia.

Ćwiczenie 8

Omów timery TON, TONR, TOF. Napisz proste programy wykorzystujące te timery. Wskaż zastosowania timerów w programach sterujących.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) narysować symbol timera TON, omówić zasadę działania, narysować przebiegi czasowe, wskazać zastosowanie w programach sterujących,
- narysować symbol timera TONR, omówić zasadę działania, narysować przebiegi czasowe, podać różnicę w działaniu tego timera a timera TON, wskazać zastosowanie w programach sterujących,
- 3) narysować symbol timera TOF, omówić zasadę działania, narysować przebiegi czasowe, wskazać zastosowanie w programach sterujących.

Wyposażenie stanowiska pracy:

- karty katalogowe sterowników,

- literatura zgodna z punktem 6 Poradnika dla ucznia.

Ćwiczenie 9

Wymień i omów liczniki CTU, CTD, CTUD i ich zastosowanie w programach sterujących.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) narysować symbol licznika CTU, omówić zasadę działania, narysować przebiegi czasowe, napisać prosty program wykorzystujący ten licznik,
- narysować symbol licznika CTD, omówić zasadę działania, narysować przebiegi czasowe, podać różnicę w działaniu tego licznika a licznika CTU, napisać prosty program wykorzystujący ten licznik,
- 3) narysować symbol licznika CTUD, omówić zasadę działania, narysować przebiegi czasowe, napisać prosty program wykorzystujący ten timer.

Wyposażenie stanowiska pracy:

- karty katalogowe sterowników,
- literatura zgodna z punktem 6 Poradnika dla ucznia.

Ćwiczenie 10

Napisz prosty program sterowania sekwencyjnego w oparciu o metodę Grafcet.

Sposób wykonania ćwiczenia

Aby wykonać ćwiczenie powinieneś:

- 1) omówić istotę programowania sterowników przy pomocy metody Grafcet,
- 2) podać etapy pisania programu sterującego tą metodą,
- 3) napisać prosty program sterowania sekwencyjnego w oparciu o tą metodę.

Wyposażenie stanowiska pracy:

- karty katalogowe sterowników,
- literatura zgodna z punktem 6 Poradnika dla ucznia.

4.3.4. Sprawdzian postępów

		Tak	Nie
Cz	zy potrafisz:		
1)	przygotować stanowisko do programowania sterowników?		
2)	adresować zmienne w pisanym programie w sposób absolutny i symboliczny?		
3)	pisać programy sterujące w języku LAD, STL, FBD?		
4)	programować timery i liczniki?		
5)	programować podstawowe funkcje logiczne AND, NAND, OR, NOR, X-OR, NOT?		
6)	wykorzystać metodę Grafcet do pisania programów sterujących?		

4.4. Obsługa programów komputerowych do programowania sterowników PLC, uruchamianie i testowanie sterowników PLC

4.4.1. Materiał nauczania

Edytor kodu źródłowego do programowania sterowników.

Programowanie sterowników wykonuje się w języku programowania pracującym w środowisku Windows. Edytor umożliwia programowanie w formie schematu drabinkowego LAD, w formie listy instrukcji STL, w formie bloków funkcjonalnych FBD. Edytor pozwala na:

– wybór typu jednostki centralnej CPU,

- ustalenie parametrów i sprawdzenie komunikacji ze sterownikiem,

- tworzenie programu w wybranej formie (LAD, STL, FBD),

- kompilacje i ładowanie programu do sterownika,

- sprawdzenie działania programu przy pomocy narzędzi edytora,

- zapisanie opracowanego programu.

Wymienione punkty tworzenia programu zostaną omówione kolejno. Korzystanie z edytora jest podobne do pracy z innymi programami. Otwieranie plików, zapisywanie plików, zmiana nazwy, drukowanie plików i inne funkcje ogólne wykonuje się korzystając z opcji "File" (rys. 4.4.1). Typowe działania w zakresie edycji programu: kopiowanie, wklejanie i usuwanie fragmentów programu, powrót do stanu sprzed ostatniej operacji wykonuje się korzystając z opcji "Edit". Położenie pasków narzędzi i środków tworzenia schematów podaje rysunek 4.4.1. Szybszy dostęp do opcji edytora zapewniają ikony tworzące pasek narzędzi edytora. Objaśnienie ich podaje rysunek 4.4.2.

Uruchomienie edytora i wybór typu CPU.

Po zainstalowaniu na komputerze oprogramowania do programowania sterowników należy

utworzyć na pulpicie skrót dostępu do edytora, np. Dozwala to na szybkie uruchomienie edytora. Po uruchomieniu edytora możliwe jest przystąpienie do tworzenia nowego projektu po wybraniu opcji File/New lub do pracy z zapisanym w pamięci projektem (wybieramy wówczas opcję File/Open). Rozpoczęcie nowego projektu rozpoczynamy od wyboru typu CPU, czyli sterownika, który chcemy oprogramować.



Rys. 4.4.1. Obraz edytora oprogramowania do programowania sterowników

Sta	ndardowy pasek narzędzi		
Stan	dard Carl A B C A B C	3 N	
۳	Otwórz nowy projekt	3	Kompiluj program
õ	Otwórz istniejąccy projekt	3	Kompiluj wszystko
Ø	Zapisz bieżący projekt		Pobierz projekt z PLC do programatora
5	Drukuj	_	Prześlij projekt z programatora do PLC
A	Podgląd wydruku	₿↓	Zapisz alfabetycznie od A-Z
*	Wytnij	<u></u> <u></u>	Zapisz alfabetycznie od Z-A
8	Kopiuj	Q	Pwiększ
B	Wklej	{#}	Załącz/wyłącz format stałej
ŝ	Cofnij		

Rys. 4.4.2. Typowy pasek narzędzi oprogramowania do programowania sterowników PLC

Wybór typu CPU i ustalenie parametrów komunikacji.

Po uruchomieniu edytora na ekranie pojawia się menu rozwijalne edytora, klikamy przycisk rozwijalny PLC, (pojawia się ikona pokazana na rys. 4.4.3), a następnie Type (pojawia się ikona pokazana na rys. 4.4.4) dla dokonania wyboru typu CPU.

View	
<u>S</u> TL Ladder	PLC Type
EBD	Select or read the PLC type from the PLC if you would like the software to range check parameters to the PLC's allowable memory ranges.
Symbol <u>T</u> able Status <u>C</u> hart	Odczytanie typu PLC
<u>D</u> ata Block System Block	PLC Type CPU 221 Read PLC Read PLC
Cross <u>R</u> eference	Wybór typu sterownika Communications
Symbolic Addressing	OK Cancel
	⊻iew STL Ladder FBD Symbol Table Status Chart Data Block System Block Cross Beference Communications Symbolic Addressing

Rys. 4.4.3. Pasek menu PLC i View



Mamy dwie możliwości wyboru typu CPU: rozwijamy zawartość paska PLC Type i dokonujemy wyboru, albo naciskamy przycisk Read PLC co spowoduje odczytanie typu CPU sterownika podłączonego do programatora lub komputera stosowanego jako programator. Dostępna z tego poziomu opcja PLC/Type/Communications pozwala również na ustalenie parametrów komunikacji programator – sterownik (rys. 4.4.5).

Communicatio	ons Links		×
	Communications	Setup	
		PC/PPI cable(PPI) Address: 0	•
Double click the with.	icon representing the PLC to communicate	>→ C Double-Click to Refresh	
Double click the parameters.	interface icon to change communication		
Double click the parameters or dia	modem icon to setup the modem al to start modem communications.		
Communicatio	n Parameters	– Parametry komunikacji	
Remote Address	2	– Adres sterownika	
Local Address	0	– Adres programatora	
Module	PC/PPI cable (COM 2)	– Podłączenie kabla programatora	
Protocol	PPI	– Protokół komunikacyjny	
Transmission Rat	e 9.6 kbps	– Szybkość transmisji	
Mode	11-bit •	– Tryb transmisji	
			-

Rys. 4.4.5. Ustalenie parametrów komunikacji

Warunkiem poprawnej komunikacji jest zgodność wybranych programowo parametrów z opcjami ustalonymi sprzętowo. Przed podjęciem pracy należy sprawdzić:- do którego szeregowego portu komputera lub programatora podłaczony jest programowany sterownik; - jaką szybkość transmisji ustalono na przełączniku znajdującym się na kablu PPI łaczącym programator i sterownik. Możliwe jest tworzenie programu bez podłączenia sterownika do programatora i późniejsze załadowanie programu do sterownika. Należy zwrócić tym przypadku uwagę, W że pozytywny wynik kompilacji nie jest jednoznaczny z poprawnością programu. Może się zdarzyć, że błędy programu zostaną

wykryte podczas ładowania programu mimo wcześniejszej pozytywnej kompilacji, a to spowoduje przerwanie ładowania programu do sterownika PLC.

Tworzenie schematu drabinkowego.

Po wybraniu typu CPU i ustaleniu parametrów komunikacji sterownika z programatorem,

🚉 F	rojec	t1						_
<u>F</u> ile	<u>E</u> dit	⊻iew	<u>P</u> LC	<u>D</u> ebug	<u>T</u> ools	<u>W</u> indows	<u>H</u> elp	

przystępujemy do wyboru języka programowania. Dokonujemy w menu View (rys.4.4.3),

z którego wybieramy opcję Ladder. Można też w ustawieniach oprogramowania ustawić domyślne otwieranie edytora w języku LAD. Do pisania programu mamy do wyboru przyciski z paska narzędzi dla języka LAD (rys. 4.4.6) lub drzewo instrukcji (ang. Instruction Tree) rys.4.4.7.



Rys. 4.4.6. Pasek narzędzi LAD

Rys. 4.4.7. Lista instrukcji LAD

W rozdziale 4.3 opisane są zasady tworzenia schematu drabinkowego. Tutaj dodamy to, że każdy szczebel drabiny logicznej (oznaczony w programie jako Network) może zawierać tylko jedną linię "prądową". Rys. 4.4.8 pokazuje błędny zapis programu, zaś rys. 4.4.9 poprawny zapis programu.





Rys. 4.4.9. Poprawny zapis programu

Tworzenie szczebla drabinki przy pomocy paska narzędzi (rys. 4.4.6).

- W celu tworzenie szczebla drabinki przy pomocy paska narzędzi należy:
- umieścić kursor w miejscu, gdzie ma być ulokowany element schematu,



- kliknąć lewym przyciskiem myszy; w wybranym miejscu pojawi się ramka,



 – kliknąć lewym przyciskiem myszy przycisk, np. styku z paska narzędzi LAD (rys. 4.4.6); w miejscu ramki pojawi się symbol styku z rozwiniętym menu styków,



 kursorem wybrać rodzaj styku i klikając powtórnie umieszczamy go w ramce lokalizacji; jednocześnie ramka przesuwa się o jedną pozycję w prawo,



- następnie klikamy na kolejny przycisk (np. cewki, bloku, rys. 4.4.6), wybieramy potrzebny symbol, klikamy i wybrany symbol lokuje się w ramce,
- czynności te powtarzamy dotąd, aż ukończymy cały zapis programu w danym szczeblu.
 Jeżeli ostatnim symbolem jest cewka lub blok ramka pozostaje na symbolu cewki (bloku),
- w miejscu, gdzie pojawiły się trzy znaki zapytania należy wpisać adres danego rozkazu (rys. 4.4.9).

Tworzenie szczebla drabiny programowej przy pomocy listy instrukcji (rys. 4.4.7).

W celu tworzenie szczebla drabinki przy pomocy listy instrukcji należy:

- umieścić kursor w miejscu, gdzie mamy ulokować element schematu,



w liście instrukcji (rys. 4.4.7) umieść kursor na żądanej instrukcji (np. styk) i kliknąć dwukrotnie,



 po dwukrotnym kliknięciu wybrana instrukcja pojawia się w oknie edytora programu a ramka prostokątna przesuwa się o jedną pozycję w prawo,



- następnie klikamy kolejną instrukcję z listy instrukcji (np. cewki, bloku),
- czynności te powtarzamy dotąd, aż ukończymy cały zapis programu w danym szczeblu.
 Jeżeli ostatnim symbolem jest cewka lub blok ramka pozostaje na symbolu cewki (bloku),

 w miejscu, gdzie pojawiły się trzy znaki zapytania należy wpisać adres danego rozkazu (rys. 4.4.9).

Kompilacja i załadowanie programu

Po opracowaniu programu w jednej z omówionych form musi być on przetworzony na język maszynowy. Służy do tego kompilacja. Jest ona jednocześnie sprawdzeniem poprawności składniowej programu. Kompilacja może być wykonana z opcji PLC/Compile (rys. 4.4.3) lub przez wybranie ikony "Kompiluj program" z paska narzędziowego (rys. 4.4.2).

W przypadku błędów w programie są one podawane w postaci komunikatu, w którym podany jest numer błędu, jego opis i jego lokalizacja (rys.4.4.10).



Rys. 4.4.10. Wyświetlanie komunikatu o poprawności napisanego programu

Program z błędami nie może być przesłany do sterownika. Aby program załadować do sterownika, należy usunąć wszystkie błędy (w przykładzie na rys. 4.4.10 należy w miejsce znaków zapytania wpisać adres, np. T35) i ponownie dokonać kompilacji. Jeżeli nie będzie błędów to poprawny program możemy załadować do sterownika. Wykonujemy to przy pomocy opcji File/Download lub przy pomocy paska narzędzi (rys. 4.4.2) wybierając ikonę "Prześlij program do PLC". Po załadowaniu programu do sterownika w oknie edytora pojawia się komunikat, że ładowanie programu zakończyło się pomyślnie. Po potwierdzeniu tego komunikatu można przystąpić do wprowadzenia sterownika w tryb pracy (ang. RUN).

W tym celu korzystamy z opcji PLC/RUN (rys. 4.4.2) lub przez wybranie ikony "Ustaw PLC w tryb Praca (RUN)" z paska narzędziowego Debug (rys. 4.4.11).



Rys. 4.4.11. Pasek narzędziowy Debug

Aby zatrzymać sterownik korzystamy z opcji PLC/STOP (rys. 4.4.2) lub przez wybranie ikony "Ustaw PLC w tryb STOP" z paska narzędziowego Debug (rys. 4.4.11).

Testowanie programu (rys. 4.4.12)

Narzędzia ułatwiające testowanie obejmuje opcja edytora Debug, która umożliwia:

- obserwację aktualnych stanów elementów schematu drabinkowego (elementy programu wypełnione na niebiesko),
- obserwację aktualnych stanów zmiennych (tablica Status Chart),
- zmianę wartości zmiennych,
- powodować wymuszenie stanów wybranych zmiennych.

📻 sample					
<u>File E</u> dit <u>V</u> iew <u>F</u>	<u>PLC D</u> ebug <u>T</u> ools <u>W</u> indows	s <u>H</u> elp			
Þ = 🌠 📅	💭 Ar 💫 A 🖨 🖨	j 👘			
View	E LAD				
Program Block	Pump_1	ank with Paint Ingredi 0_1 High_Le	ent 1. vel Pump_1 (Debug Debug Ustaw PLC w tryb RUN (praca) Ustaw PLC w tryb STOP Rokaz (Urrai) status programu
Status Chart	De Chalue Chart				7 Jacz/Withors pauze
	Address	Format	Current Value	New Value	Pokaż/Ukryj tablicę statusu zmiennych
Data Block	1 Start_1 2 Start_2	Bit	2#0 2#1		ि Jednokrotne uaktualnienie wartości zmiennyc
	4 Stop_2	Bit	2#1 2#1 2#1		🐱 Wczytanie nowych wartości
System Block	6 Low_Level	Bit	2#0		🔁 Zablokuj dane PLC
	7 Reset	Bit Signed	2#0	_ _	🗄 Odblokuj dane PLC
Tools					🛅 Odblokuj wszystko
Ready	PC/PP	I cable 9.6 k	bps Local: 0, COM1	Remote: 2, //	🗂 Czytaj wszystkie zablokowane wartości

Rys. 4.4.12. Testowanie programu przy użyciu paska narzędziowego Debug

Opisy i komentarze

Poszczególne szczeble programu są oznaczone kolejnymi liczbami (Network 1, Network 2, Network 3 itd.). Obok numeracji obwodów można wpisać rodzaj operacji wykonywanej w danym szczeblu wraz z komentarzem, co czyni program bardziej czytelnym (rys. 4.4.10).

×
^
el
IC6

Rys. 4.4.13. Wpisywanie nazw symbolicznych oraz komentarzy programu w szczeblu 1

4.4.2. Pytania sprawdzające

Odpowiadając na pytania, sprawdzisz, czy jesteś przygotowany do wykonania ćwiczeń.

- 1. Jakie czynności należy wykonać po uruchomieniu oprogramowania specjalistycznego do programowania sterowników PLC?
- 2. Jak i po co dokonujemy wyboru jednostki centralnej CPU?
- 3. Jakie są podstawowe parametry komunikacyjne sterownika?
- 4. Co to jest adres sieciowy sterownika?
- 5. Co decyduje o wyborze języka programowania sterownika?
- 6. Jakie są zalety pisania programów sterujących z adresowaniem symbolicznym?
- 7. Co to jest kompilacja? Czy można przesłać program z programatora do sterownika bez kompilacji?

- 8. Co należy zrobić, jeżeli ładowanie programu zakończyło się niepowodzeniem?
- 9. Od czego zależy czas przesyłu programu z programatora do sterownika?
- 10. Który język programowania jest bardziej komunikatywny: LAD, STL, czy FBD?
- 11. Jakie instrukcje nie mogą rozpoczynać szczebla programu sterującego, a jakie go kończyć?
- 12. W jaki sposób przesyłamy program z programatora do sterownika, a w jaki ze sterownika do programatora?
- 13. Czy program przesłany ze sterownika do programatora ma taką samą postać jak program w programatorze?
- 14. Czy można programować sterownik, który nie jest podłączony do programatora?

4.4.3. Ćwiczenia

Ćwiczenie 1

Zainstaluj program komputerowy do programowania sterowników PLC do komputera PC, który będzie używany jako programator sterowników PLC.

Sposób wykonania ćwiczenia.

Aby wykonać ćwiczenie powinieneś:

- 1) zapoznać się z wymaganiami dotyczącymi instalacji oprogramowania specjalistycznego do programowania sterowników PLC zawartego na dysku instalacyjnym,
- 2) sprawdzić, czy komputer spełnia te wymagania,
- 3) zainstalować oprogramowanie na komputerze.

Wyposażenie stanowiska pracy:

- komputer PC,
- dysk instalacyjny z oprogramowaniem,
- literatura zgodna z punktem 6 Poradnika dla ucznia.

Ćwiczenie 2

Połącz komputer PC ze sterownikiem za pomocą kabla programatora i ustaw parametry komunikacyjne sterownika.

Sposób wykonania ćwiczenia.

Aby wykonać ćwiczenie powinieneś:

- 1) przygotować komputer PC do pracy,
- 2) uruchomić edytor do programowania sterowników,
- 3) zanotować parametry komunikacyjne, jakie należy ustawić,
- 4) załadować do sterownika dokonane ustawienia,
- 5) sprawdzić komunikację między sterownikiem a programatorem.

Wyposażenie stanowiska pracy:

- komputer PC,
- sterownik PLC,
- kabel programatora,
- literatura zgodna z punktem 6 Poradnika dla ucznia.

Ćwiczenie 3

Lampka H1 powinna się zapalić po naciśnięciu przycisków zwiernych S1 i S2 i zgasnąć po zwolnieniu jednego z nich. Napisz program sterujący załączaniem lampki i uruchom go.

Sposób wykonania ćwiczenia.

Aby wykonać ćwiczenie powinieneś:

- 1) narysować schemat ideowy elektryczny układu,
- 2) przeanalizować działanie układu,
- 3) narysować schemat połączeń zacisków sterownika,
- 4) napisać listę przyporządkowującą,
- 5) napisać program sterujący w wybranym przez siebie języku programowania,
- 6) przesłać program do sterownika,
- 7) sprawdzić działanie programu sterującego,
- 8) określić rodzaj funkcji logicznej realizowanej przez program.

Wyposażenie stanowiska pracy:

- komputer PC,
- sterownik PLC,
- kabel programatora,
- zasilacz prądu stałego,
- przybory do rysowania, kartka papieru,
- lampki
- literatura zgodna z punktem 6 Poradnika dla ucznia.

Ćwiczenie 4

Lampka H1 powinna się zapalić po naciśnięciu przycisku zwiernego S1 lub S2 i zgasnąć po zwolnieniu obydwu. Napisz program sterujący załączaniem i wyłączaniem lampki i uruchom go.

Sposób wykonania ćwiczenia.

Aby wykonać ćwiczenie powinieneś:

- 1) narysować schemat ideowy elektryczny układu,
- 2) przeanalizować działanie układu,
- 3) narysować schemat połączeń zacisków sterownika,
- 4) napisać listę przyporządkowującą,
- 5) napisać program sterujący w wybranym przez siebie języku programowania,
- 6) przesłać program do sterownika,
- 7) sprawdzić działanie programu sterującego,
- 8) określić rodzaj funkcji logicznej realizowanej przez program.

Wyposażenie stanowiska pracy:

- komputer PC,
- sterownik PLC,
- kabel programatora,
- zasilacz prądu stałego,
- przybory do rysowania, kartka papieru,
- lampki,
- literatura zgodna z punktem 6 Poradnika dla ucznia.

Ćwiczenie 5

Lampka H1 powinna zapalić się po 15s od naciśnięcia przycisku zwiernego S1. Lampka ma zostać wyłączona po naciśnięciu przycisku zwiernego S2. Napisz program sterujący załączaniem i wyłączaniem lampki i uruchom go.

Sposób wykonania ćwiczenia.

Aby wykonać ćwiczenie powinieneś:

- 1) narysować schemat ideowy elektryczny układu,
- 2) przeanalizować działanie układu,
- 3) narysować schemat połączeń zacisków sterownika,
- 4) napisać listę przyporządkowującą,
- 5) napisać program sterujący w wybranym przez siebie języku programowania,
- 6) przesłać program do sterownika,
- 7) sprawdzić działanie programu sterującego.

Wyposażenie stanowiska pracy:

- komputer PC,
- sterownik PLC,
- kabel programatora,
- zasilacz prądu stałego,
- przybory do rysowania, kartka papieru,
- lampki
- literatura zgodna z punktem 6 Poradnika dla ucznia.

Ćwiczenie 6

Lampka H1 powinna się zapalić po pięciokrotnym naciśnięciu przycisku zwiernego S1 a zgasnąć po naciśnięciu przycisku zwiernego S2. Napisz program sterujący załączaniem i wyłączaniem lampki i uruchom go.

Sposób wykonania ćwiczenia.

Aby wykonać ćwiczenie powinieneś:

- 1) narysować schemat ideowy elektryczny układu,
- 2) przeanalizować działanie układu,
- 3) narysować schemat połączeń zacisków sterownika,
- 4) napisać listę przyporządkowującą,
- 5) napisać program sterujący w wybranym przez siebie języku programowania,
- 6) przesłać program do sterownika,
- 7) sprawdzić działanie programu sterującego.

Wyposażenie stanowiska pracy:

- komputer PC,
- sterownik PLC,
- kabel programatora,
- zasilacz prądu stałego,
- przybory do rysowania, kartka papieru,
- lampki,
- literatura zgodna z punktem 6 Poradnika dla ucznia.

4.4.4. Sprawdzian postępów

	Tak	Nie
Czy potrafisz:		
1) zainstalować na komputerze PC oprogramowanie do programowania sterowników PLC?		
2) ustawić parametry komunikacyjne sterownika?		
3) sprawdzić poprawność napisanego programu?		
4) znaleźć przyczyny i usunąć je w przypadku, gdy przesyłanie programu do sterownika zakończyło się niepowodzeniem?		
5) określić, czy sterownik, który programujesz jest odpowiedni do procesu którym ma sterować?	l, 🗌	

5. SPRAWDZIAN OSIĄGNIĘĆ

TEST 1

INSTRUKCJA DLA UCZNIA

- 1. Przeczytaj uważnie instrukcję.
- 2. Podpisz imieniem i nazwiskiem kartę odpowiedzi.
- 3. Zapoznaj się z zestawem pytań testowych.
- 4. Test zawiera 10 pytań. Do każdego pytania dołączone są 4 możliwości odpowiedzi. Tylko jedna jest prawidłowa.
- 5. Udzielaj odpowiedzi tylko na załączonej karcie odpowiedzi, stawiając w odpowiedniej rubryce znak X. W przypadku pomyłki należy błędną odpowiedź zaznaczyć kółkiem, a następnie ponownie zakreślić odpowiedź prawidłową.
- 6. Pracuj samodzielnie, bo tylko wtedy będziesz miał satysfakcję z wykonanego zadania.
- 7. Kiedy udzielenie odpowiedzi będzie Ci sprawiało trudność, wtedy odłóż jego rozwiązanie na później i wróć do niego, gdy zostanie Ci wolny czas.
- 8. Na rozwiązanie testu masz 30 min. Powodzenia!

Zestaw zadań testowych

- 1. Podstawowa struktura sterownika PLC to: centralna jednostka przetwarzająca CPU oraz
 - a) zasilacz, układ programujący, sensory, interfejs komunikacyjny,
 - b) wskaźniki stanu, aktuatory, , układy wejścia i wyjścia, przetworniki analogowocyfrowe,
 - c) zasilacz, interfejs komunikacyjny, układy wejścia i wyjścia, wskaźniki stanu,
 - d) moduł rozszerzający, zasilacz, pamięć EEPROM, wskaźniki stanu.
- 2. Sensory to:
 - a) styczniki, lampki sygnalizacyjne.
 - b) przekaźniki, siłowniki.
 - c) elektrozawory, silniki.
 - d) fotokomórki, czujniki temperatury.
- 3. Aktuatory to:
 - a) przyciski, czujniki optyczne.
 - b) fotokomórki, łączniki.
 - c) silniki, styczniki.
 - d) czujniki ciśnienia, przetworniki analogowo-cyfrowe.
- 4. Interfejs komunikacyjny sterownika służy do podłączenia
 - a) sensorów.
 - b) aktuatorów.
 - c) sterownika z programatorem.
 - d) modułów rozszerzających.
- 5. Czas ładowania programu sterującego z programatora do sterownika zależy od:
 - a) języka programowania.
 - b) długości kabla łączącego programator ze sterownikiem.
 - c) ilości sygnałów wejściowych i wyjściowych.
 - d) liczby instrukcji programu sterującego.

[&]quot;Projekt współfinansowany ze środków Europejskiego Funduszu Społecznego"

- 6. Język programowania drabinkowego to:
 - a) STL.
 - b) Grafcet.
 - c) FBD.
 - d) LAD.
- 7. Zmienne wyjściowe sterownika oznaczane są w programie przez:
 - a) T.
 - b) C.
 - c) Q.
 - d) I.
- 8. Adresowanie bitowe zmiennych w programie ma postać:
 - a) MW12.
 - b) Q5.7.
 - c) IB5.
 - d) VD32.
- 9. Licznik CTU to licznik zliczający
 - a) w dół.
 - b) w górę i w dół.
 - c) w górę.
 - d) liczbę cyklów pracy sterownika.
- 10. Timer TON służy do
 - a) zliczania czasu pracy urządzenia.
 - b) regulacji czasu pracy urządzenia.
 - c) załączenia urządzenia z opóźnieniem.
 - d) wyłączenia urządzenia z opóźnieniem.

KARTA ODPOWIEDZI

Imię i nazwisko.....

Programowanie sterowników PLC

Zakreśl poprawną odpowiedź.

Nr zadania	Odpowiedź				Punkty
1	a	b	с	d	
2	a	b	с	d	
3	а	b	с	d	
4	а	b	с	d	
5	а	b	с	d	
6	а	b	с	d	
7	а	b	с	d	
8	а	b	с	d	
9	а	b	с	d	
10	a	b	с	d	
				Razem:	

6. LITERATURA

- 1. Ackerman R., Franz J., Hartma T., Hopf A.: Programowalne systemy sterowania. Stopień podstawowy. Festo Didactic GmbH–Esslingen–1987
- 2. Borelbach K. H., Kraemer G., Mock W., Nows E.: Technika sterowników z programowalną pamięcią. Tłumaczenie A. Rodak. WSiP, Warszawa 1998
- 3. Chwaleba A., Moeschke B., Płoszajski G.: Elektronika. WSiP, Warszawa 1985
- 4. Głocki W.: Układy cyfrowe. WKŁ, Warszawa1996
- 5. Heimann B.: Mechatronika. PWN, Warszawa 2001
- 6. Hörnemann E., Hübscher H., Klause J., Schierack K., Stolzenburg R.: Elektrotechnika. Instalacje elektryczne i elektronika przemysłowa. WSiP, Warszawa 1998
- 7. Instrukcja obsługi i programowania S7–200 (STEP 7–MicroWin User Manual)
- 8. Jeruszko U., Niemierko B.: Pomiary sprawdzające w kształceniu zawodowym. MEN, Warszawa 1997
- 9. Kostro J.: Elementy, urządzenia i układy automatyki. WSiP, Warszawa 1988
- 10. Kostro J.: Urządzenia automatyki. WSiP, Warszawa 1985
- 11. Kręglewska U., Sacha K., Gustowski J.: Podstawy sterowania. Oficyna Wydawnicza Politechniki Warszawskiej 2003
- 12. Money S.A.: Mikroprocesory, poradnik. WKŁ, Warszawa1996
- 13. Niemierko B.: Pomiar wyników kształcenia zawodowego. Centrum Informacji i Dokumentacji, Warszawa 1997
- 14. Norma IEC 1131–3
- 15. Norma europejska EN 61131
- 16. Praca zbiorowa: Praktyczne zastosowania elektrotechniki. Wydawnictwo REA, Warszawa 2003
- 17. Schmid D., Bauman A., Kaufmann H.: Mechatronika. REA, Warszawa 2002

Internet

- 1. www.automatyka.vector.com.pl
- 2. www.europe.omron.pl
- 3. www.festo.pl
- 4. www.moeller.pl
- 5. www.plcs.pl
- 6. www.siemens.pl/simatic
- 7. www.sterowniki-plc.xo.pl