

STEROWANIE MASZYN I URZĄDZEŃ I

Laboratorium

1. Wprowadzenie

Opracował: dr hab. inż. Cezary Orlikowski

Instytut Politechniczny



Ogólny schemat stanowiska laboratoryjnego ze sterownikiem programowalnym przedstawiono na rys. 1, gdzie:

PLC – sterownik programowalny (*GE VersaMax Micro*),

JP – jednostka podstawowa,

MR – moduł rozszerzający;

PC – komputer (programator).

Komputer wyposażony jest w program narzędziowy *Cimplicity* do programowania sterowników (tworzenia programów sterujących) przy pomocy schematów drabinkowych.

STEROWNIK PROGRAMOWALNY

Sterownik programowalny jest urządzeniem, w którym mikroprocesor na podstawie programu zapisanego w pamięci steruje przebiegiem procesu technologicznego. Ogólny schemat funkcjonalny sterownika pokazano na rys.2. Sterownik odczytuje sygnały wejściowe pochodzące z czujników i przetworników pomiarowych, następnie wykonuje obliczenia zgodnie z programem sterującym, a potem na podstawie wyników ustawia sygnały wyjściowe sterując urządzeniami wykonawczymi. Odczyt wejść, obliczenia i ustawianie wyjść są powtarzane stale z okresem zwanym cyklem sterownika – rys. 3. Wejścia binarne pochodzą z przekaźników, czujników stykowych, wyłączników krańcowych, przycisków ręcznych itd. Wyjścia binarne za pośrednictwem przekaźników i styczników włączają silniki, zawory, grzejniki, elementy sygnalizacyjne itp.

Jednostka centralna steruje wymianą informacji z urządzeniami zewnętrznymi i z pamięcią oraz wykonuje kolejne kroki programu zapisane w pamięci. Pamięć w sterowniku, podobnie jak w każdym komputerze, służy do zapamiętania programu, który ma być wykonywany oraz do przechowywania informacji pośrednich powstających w trakcie wykonywania programu.

Do programowania sterowników najczęściej służą specjalne programatory lub komputery z odpowiednim oprogramowaniem (narzędziowym), połączone ze sterownikiem.

Dane sterownika w laboratorium:

wejścia logiczne:

- liczba: 16 (*JP*) i 8 (*MR*)

- poziom sygnału: 0V÷24V („0”= 0V÷5V; „1”=11,5V÷24V)

wyjścia logiczne:

- liczba: 8 (*JP*) i 4 (*MR*)

- poziom sygnału maksymalny: 5V÷24V.

Na rys. 4 pokazano sposób podłączenia wejść i wyjść logicznych, gdzie: *S* – element zadający sygnał wejściowy, *P* – element wyjściowy, *K* – przekaźnik.

PROGRAM NARZĘDZIOWY *CIMPLICITY*

Sterowniki *GE FANUC* mogą być programowane przy pomocy programu *CIMPLICITY*. Program ten bazuje na metodzie schematów stykowo – przekaźnikowych nazywanych także schematami drabinkowymi. Po zbudowaniu schematu regulatora przy pomocy programu należy zbudowany schemat przesłać, do ste-

rownika, który po uruchomieniu będzie działał (przetwarzał sygnały wejściowe na wyjściowe) zgodnie z tym schematem.

Ogólnie, układ stykowo – przekaźnikowy składa się z dwóch podstawowych rodzajów elementów: styków (elementów wejściowych) i przekaźników (elementów wyjściowych). W programie *CIMPLICITY* funkcje logiczne realizowane przez sterownik przedstawiane są właśnie w taki właśnie sposób. Programowanie sterownika polega zatem na ustawieniu na ekranie komputera konfiguracji styków i przekaźników odpowiadającej funkcjom logicznym, które mają być realizowane przez sterownik. Stosowane w programie *CIMPLICITY* symbole graficzne styków i przekaźników zostały pokazane na rys. 5.

Styki i przekaźniki nie są jedynymi elementami dostępnymi podczas programowania sterownika. Oprócz nich są także takie elementy (bloki funkcyjne – rys. 5) jak: przekaźniki czasowe, liczniki i inne.

Podstawową strukturą programu jest jednak schemat drabinkowy i dlatego wszystkie inne (oprócz styków i przekaźników) elementy są także w taki schemat włączane. Bloki funkcyjne przedstawione są jako duże prostokąty. Schemat drabinkowy posiada symboliczne źródło zasilania – zakłada się przepływ sygnału od „szyny zasilającej” umieszczonej po lewej stronie schematu, poprzez połączenia, do przekaźnika lub bloku funkcyjnego umieszczonego po prawej stronie. Cały program składa się ze „szczebli” drabiny. Szczebel drabiny logicznej musi posiadać odpowiedni format oraz składnię. Najważniejsze zasady tworzenia struktury drabiny logicznej są następujące:

- ostatnim elementem szeregowego połączenia elementów szczebla programu musi być przekaźnik lub blok funkcyjny
- jeśli w szczeblu jest przekaźnik uaktywniany zboczem sygnału sterującego to musi być on jedynym przekaźnikiem w tym szczeblu
- szczebel musi zawierać przynajmniej jeden styk przed przekaźnikiem, blokiem funkcyjnym lub połączeniem pionowym
- po bloku funkcyjnym w szczeblu nie mogą wystąpić żadne styki

Próba nieprawidłowego komponowania szczebla jest sygnalizowana przez program. Na rys. 6 pokazano przykładowy, prawidłowy, rozgałęziony szczebel drabiny logicznej, w którym występuje 3 styki, 2 przekaźniki i jeden blok funkcyjny.

Zmienne programu sterującego

Podczas budowania drabiny logicznej programu sterującego elementom logicznym użytkownik (programista) przypisuje zmienne. Oprogramowanie *CIMPLICITY* posiada dwa typy zmiennych: zmienne dyskretne (jednobitowe – mogące przyjmować wartość logiczną 0 lub 1), oraz zmienne rejestrowe, które służą do przechowywania danych innych niż pojedyncze bity. Elementom takim jak przekaźniki i styki przypisuje się zmienne dyskretne, pozostałe elementy wymagają przypisania kilku zmiennych, często różnego typu, jako parametrów. W tabl. 1 – 3 dokonano skrótego zestawienia wybranych typów zmiennych i ich symboli.

Zmienne dyskretne

Tablica

1

Typ	Opis
%I	Zmienna reprezentująca fizyczne wejście dyskretne. Po symbolu następuje adres zmiennej w tablicy stanu wejść (np. %I00121).
%Q	Zmienna reprezentująca fizyczne wyjście dyskretne. Po symbolu następuje adres zmiennej w tablicy stanu wyjść (np. %Q00121).
%M	Symbol reprezentujący wewnętrzną zmienną dyskretną programu sterującego. Zmienna typu %M może posiadać pamięć stanu, jeśli przypisana jest przekąźnikowi z pamięcią. (po wyłączeniu sterownika i jego ponownym włączeniu wartość tej zmiennej jest taka jak przed wyłączeniem zasilania). W przypadku przypisania przekąźnikowi bez pamięci zmienna nie posiada pamięci stanu.
%T	Symbol reprezentujący chwilową zmienną dyskretną, bez możliwości wykorzystania pamięci stanu..

Zmienne rejestrowe

Tablica

2

Typ	Opis
%R	Zmienna 16-bitowa, oznaczająca rejestr, w którym można przechowywać dane programu sterującego (np. wyniki obliczeń). Symbolowi powinien towarzyszyć adres rejestru (np. %R00201).

Zmienne systemowe

Tablica

3

Zmienne	Nazwa pomocnicza	Definicja
%S0001	FST_SCN	Ustawiana jest wartość 1 jeżeli bieżący cykl jest pierwszym cyklem pracy sterownika
%S0002	LST_SCN	Ustawiana jest wartość 1 jeżeli bieżący cykl jest ostatnim cyklem pracy sterownika
%S0003	T_10MS	Podstawa czasu generatora sygnału prostokątnego: 0.01 s.
%S0004	T_100MS	Podstawa czasu generatora sygnału prostokątnego: 0.1 s.
%S0005	T_SEC	Podstawa czasu generatora sygnału prostokątnego: 1 s.
%S0006	T_MIN	Podstawa czasu generatora sygnału prostokątnego: 1 min.
%S0007	ALW_ON	Styk zwarty na stałe.
%S0008	ALW_OFF	Styk otwarty na stałe.

Przy pomocy zmiennych opisanych w tabl. 1 –3, można określać w programie sterującym różne typy danych przedstawione w tabl. 4.

Typy danych

Tablica 4

Typ	Nazwa	Opis
<i>INT</i>	Signed Integer – liczby całkowite ze znakiem	Liczby całkowite ze znakiem zajmują 16 bitów pamięci i są zapisywane w formie dopełnienia do dwóch dla liczb ujemnych. Zakres: od –32768 do +32767
<i>DINT</i>	Double Precision Signed Integer – liczby całkowite podwójnej precyzji ze znakiem	Liczby całkowite podwójnej precyzji ze znakiem są przechowywane w 32 bitach pamięci i są zapisywane w formie dopełnienia do dwóch dla liczb ujemnych. Zakres: od –2147483648 do +2147483647
<i>BIT</i>	Bit	Dana zajmująca najmniejszą komórkę pamięci, mogąca przyjmować wartość 0 lub 1. Słowo bitowe może mieć długość N.
<i>BYTE</i>	Byte – bajt	Dana zawierająca 8 bitów
<i>WORD</i>	Word – słowo	Słowo – wykorzystuje 16 kolejnych bitów pamięci sterownika, ale, w przeciwieństwie do ciągu bitów reprezentującego w pamięci sterownika liczbę, bity mogą być niezależne od siebie. Każdy bit posiada swój własny stan logiczny: 0 lub 1.

Zestaw instrukcji

Program narzędziowy *CIMPLICITY* udostępnia szeroki zestaw instrukcji (elementów) do tworzenia programów sterujących. Celem przybliżenia możliwości programowania sterowników poniżej zestawiono wybrane elementy zestawu instrukcji. Bardziej szczegółowo będą one omówione w kolejnych ćwiczeniach. Tablica 5 zawiera wybrane grupy elementów, a tablice 6 –10 wybrane elementy w poszczególnych grupach.

Wybrane grupy elementów programu

Tablica 5

Grupa funkcji	Opis
<i>BIT OPERATIONS</i>	Bloki działań na ciągach bitów
<i>COILS</i>	Przekaźniki
<i>CONTACTS</i>	Styki
<i>COUNTERS</i>	Liczniki
<i>TIMERS</i>	Przekaźniki czasowe

Bloki działań na ciągach bitów

Tablica 6

Symbol	Opis funkcji
<i>AND</i>	Operacja logiczna <i>AND</i> na dwóch ciągach bitów
<i>OR</i>	Operacja logiczna <i>OR</i> na dwóch ciągach bitów
<i>XOR</i>	Operacja logiczna <i>XOR</i> na dwóch ciągach bitów
<i>NOT</i>	Operacja logiczna <i>NOT</i> na ciągu bitów
<i>SHL</i>	Przesunięcie ciągu bitów w lewo
<i>SHR</i>	Przesunięcie ciągu bitów w prawo
<i>ROL</i>	Przesunięcie ciągu bitów w lewo w obiegu zamkniętym
<i>ROR</i>	Przesunięcie ciągu bitów w prawo w obiegu zamkniętym
<i>BIT-SEQ</i>	Przesuwanie „1” w obiegu zamkniętym

Przełączniki

Tablica 7

Symbol	Opis funkcji
–()–	Przełącznik o stykach otwartych, zwieranych w momencie dotarcia sygnału do przełącznika. Wartość logiczna przypisanej zmiennej jest wtedy ustawiana na 1
–(/)–	Przełącznik o stykach zamkniętych, rozwieranych w momencie dotarcia sygnału do przełącznika. Wartość logiczna przypisanej zmiennej jest wtedy ustawiana na 0
–(S)–	Przełącznik o stykach zwieranych w momencie dotarcia sygnału (wartość logiczna przypisanej zmiennej ustawiana jest wtedy na 1). Styki pozostają zwarte, a wartość zmiennej pozostaje 1 do czasu zadziałania sprzężonego przełącznika –(R)–
–(R)–	Przełącznik o stykach rozwieranych w momencie dotarcia sygnału. Styki pozostają rozwarte, a wartość zmiennej pozostaje 0 do czasu zadziałania sprzężonego przełącznika –(S)–
–(↑)–	Przełącznik uaktywniany zboczem narastającym sygnału
–(↓)–	Przełącznik uaktywniany zboczem opadającym sygnału

Styki

Tablica 8

Sym- bol	Opis funkcji
– –	Styk otwarty – przewodzi sygnał, gdy wartość logiczna przypisanej zmiennej jest 1
– /–	Styk zamknięty – przewodzi sygnał, gdy wartość logiczna przypisanej zmiennej jest 0

Liczniki

Tablica 9

Symbol	Opis funkcji
<i>UPCT</i> <i>R</i>	Licznik zliczający w górę – jego zawartość jest zwiększana o 1 za każdym razem, gdy wartość logiczna dopływającego sygnału zmienia się z 0 na 1. Licznik przesyła sygnał wyjściowy, gdy zliczana wartość osiągnie wartość zadaną. Podanie sygnału na wejście zerujące licznika powoduje wyzerowanie jego zawartości.
<i>DNCT</i> <i>R</i>	Licznik zliczający w dół (od wartości zadanej). Jego zawartość jest zmniejszana o 1 za każdym razem, gdy wartość logiczna dopływającego sygnału zmienia się z 0 na 1. Licznik przesyła sygnał wyjściowy, gdy zliczana wartość osiągnie zero. Podanie sygnału na wejście zerujące licznika powoduje ustawienie jego zawartości na wartość zadaną.

Przełączniki czasowe

Tablica 10

Symbol	Opis funkcji
<i>ONDTR</i>	Przełącznik czasowy z pamięcią zliczający czas, gdy dopływa do niego sygnał. Gdy sygnał nie dopływa, wartość zliczona jest przechowywana. Przesłanie sygnału wyjściowego następuje, gdy zliczona wartość osiągnie wartość zadaną. Bieżąca zawartość pamięci przełącznika może zostać wyzerowana poprzez przesłanie sygnału na wejście zerujące przełącznika.
<i>TMR</i>	Przełącznik czasowy bez pamięci zliczający czas, gdy dopływa do niego sygnał. Gdy sygnał przestaje dopływać, wartość zliczona jest zerowana. Przesłanie sygnału wyjściowego następuje, gdy zliczona wartość osiągnie wartość zadaną.

Menu główne programu narzędziowego

Po uruchomieniu programu *CIMPLICITY* wyświetlony zostaje ekran główny programu oraz pojawia się okno wyboru istniejącego (lub rozpoczęcia nowego) programu sterującego (rys. 7). W przypadku wyboru opcji „nowy program”

(empty project) ukazuje się okno (rys. 8) z polem do wpisania nazwy nowego programu.

Po wybraniu istniejącego programu lub wpisaniu nazwy nowego i jej zatwierdzeniu ekran programu Cimplicity ma postać jak na rys. 9 z aktywnym oknem *Nawigator* zawierającym drzewo katalogowe realizowanego projektu.

Tworzenie programów sterujących

Ekran programu narzędziowego z rys.9 zawiera paski narzędziowe znajdujące się w górnej jego części. Jeden z pasków narzędziowych (pokazany oddzielnie na rys.10) zawiera ikonę okna *Nawigator* umożliwiającą zamykanie i otwieranie tego okna. Ten sam pasek narzędziowy zawiera także ikony dwóch innych okien: *Biblioteka elementów*, *Inspektor*. Rozpoczęcie edycji programu sterującego wymaga uaktywnienia tych okien (pomocniczych), a także okna podstawowego edycji programu sterującego. Okno edycji programu sterującego otwiera się przez dwukrotne kliknięcie katalogu *Main* w drzewie katalogowym projektu w oknie *Nawigatora*. Na rys. 11 pokazano widok ekranu programu *CIMPLICITY* z oknem edycji programu sterującego (*Main*) i oknami pomocniczymi *Nawigator*, pokazanymi oddzielnie na rys. 12 – 13. Trzy okna pomocnicze są rozwijane (lub zwijane) przez wskazywanie odpowiednich ikon w jednym z pasków narzędziowych widocznym na rys. 11 oraz oddzielnie pokazanym na rys. 10.

Okno *Biblioteka elementów* (pokazane oddzielnie na rys.12) zawiera dostępne elementy (funkcje) do tworzenia programu sterującego. Wybrane z nich są przedstawione w tablicach 5 – 10. Niektóre z tych funkcji dostępne są na pasku narzędziowym widocznym na ekranie roboczym przedstawionym na rys. 11, a także pokazanym oddzielnie na rys.14. Ten pasek narzędziowy zawiera także narzędzia, pomocne w konstruowaniu schematów drabinkowych, takie jak: wskaźnik, rysowanie linii poziomych i pionowych, wstawianie nowego szczebla schematu drabinkowego, wpisywanie komentarza.

Okno *Inspektora* (pokazane oddzielnie na rys. 13) służy, między innymi, do wpisywania adresów zmiennych (sygnałów) przypisanych elementom takim jak styki, przekaźniki i bloki funkcyjne.

Konstruowanie schematu drabinkowego polega na przeciągnięciu myszką nazwy elementu wybranego z okna *Biblioteka elementów* do okna edycji programu sterującego. W przypadku stosowania elementów dostępnych na pasku narzędziowym należy wybrany element wskazać kursorem, a następnie kliknąć w odpowiednim miejscu ekranu edycji programu sterującego.

Po narysowaniu schematu programu sterującego należy nadać nazwy (dowolne) wszystkim występującym w nim sygnałom. Dwukrotne kliknięcie na styk, przekaźnik lub linię sygnału wejściowego/wyjściowego bloku funkcyjnego powoduje otwarcie okna edycji nazwy sygnału (rys. 15). Nazwę należy wpisać w odpowiednim polu i zatwierdzić.

Następnym etapem jest przypisanie wszystkim sygnałom odpowiednich adresów zgodnych z typem danej zmiennej. Po wskazaniu nazwy sygnału odpowiedni adres wpisuje się w polu *Ref.Address* w oknie *Inspektora* (rys. 11 i 13).

Uruchamianie i testowanie programów sterujących

Poprawnie skonstruowany program sterujący (schemat drabinkowy) może być przesłany do sterownika i uruchomiony. Na rys. 16 pokazano pasek narzędziowy (por. także rys. 11) z poleceniami związanymi z uruchamianiem programu sterującego.

Polecenie *online/offline* służy do przełączania trybu połączenia pomiędzy sterownikiem i komputerem. Tworzenie programu sterującego (lub jego modyfikacja) powinno odbywać się w trybie *offline* natomiast przesłanie programu sterującego do sterownika oraz testowanie programu musi być realizowane w trybie *online*. Tryb *online* jest sygnalizowany zielonymi znacznikami „przy szynie zasilającej” schematu drabinkowego.

Przesłanie programu do sterownika (oraz jego uruchomienie) nastąpi wtedy gdy w trybie *online* zostanie wskazana ikona *ładowanie i start programu* (rys. 16). Uruchamiania programu, który znajduje się już w pamięci sterownika dokonuje się przy pomocy polecenia *start programu*. Zatrzymanie wykonywania programu przez sterownik może być dokonane przy pomocy polecenia *stop programu*.

Uruchomiony program sterujący przetwarza sygnały wejściowe (wyjściowe obiektu) na wyjściowe (wejściowe obiektu) zgodnie ze strukturą programu. Działanie programu (zmianę wartości sygnałów w czasie rzeczywistym) można obserwować na ekranie komputera programatora.

Program *CIMPLICITY* umożliwia także przetestowanie skonstruowanego programu bez konieczności podłączania sygnałów z(do) obiektu do(ze) sterownika. Możliwe jest bowiem programowe wymuszanie wartości sygnałów dyskretnych i sprawdzenie czy sygnały wyjściowe przyjmują wartości zgodne założeniami autora programu sterującego.

Programowe wymuszenie wartości sygnału wejściowego (dyskretnego) wykonywane jest przez wskazanie prawym klawiszem myszki danego elementu (rys. 17), a następnie wybranie polecenia *force on* („1”) lub *force off* („0”). Nazwa sygnału z wymuszoną w ten sposób wartością wyświetlana jest w kolorze czerwonym. Likwidacji wymuszenia wartości sygnału można dokonać przez wybór polecenia *remove force* (rys.17).

CEL I PRZEBIEG ĆWICZENIA

Cel

Zapoznanie się ze sterownikiem programowalnym *GE VersaMax Micro* i programem narzędziowym *Cimplicity*.

Zakres

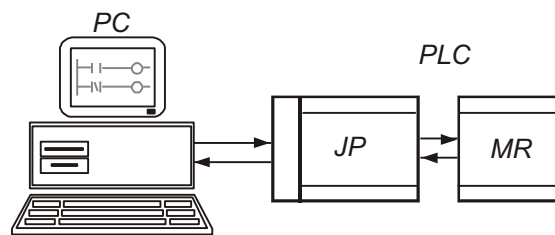
Uruchamianie sterownika.

Uruchamianie i obsługa programu narzędziowego.

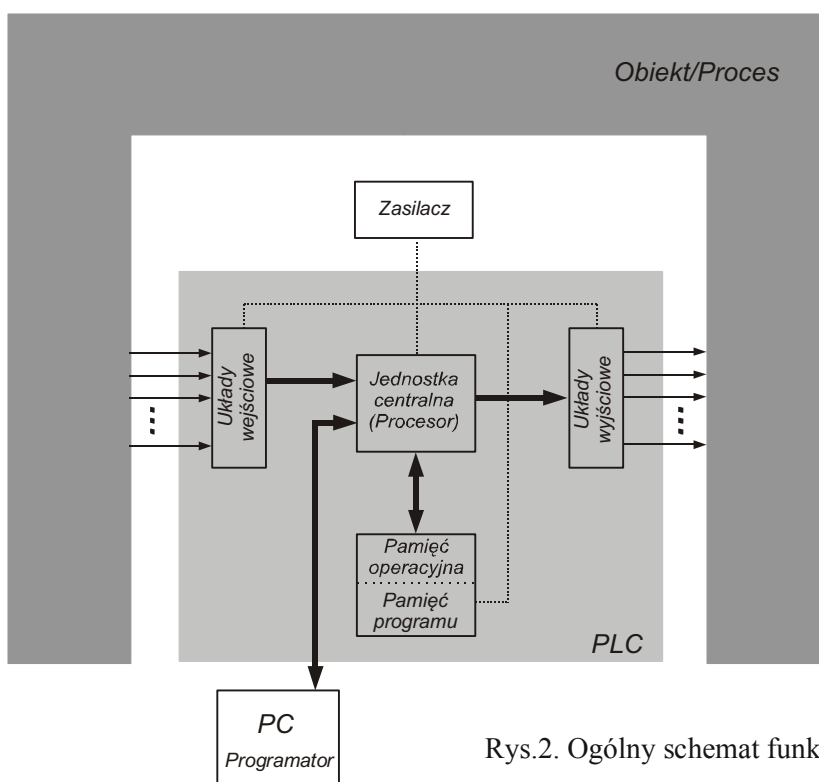
Poznanie zasad tworzenia programu sterującego.

Przesyłanie programu sterującego do sterownika, uruchamianie i testowanie.

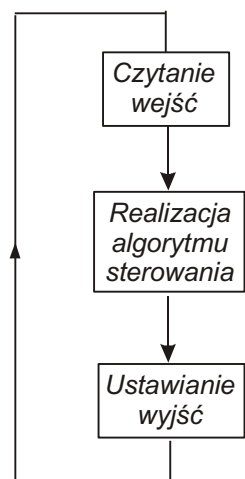
Demonstracja połączenia sterownika z obiektem sterowania.



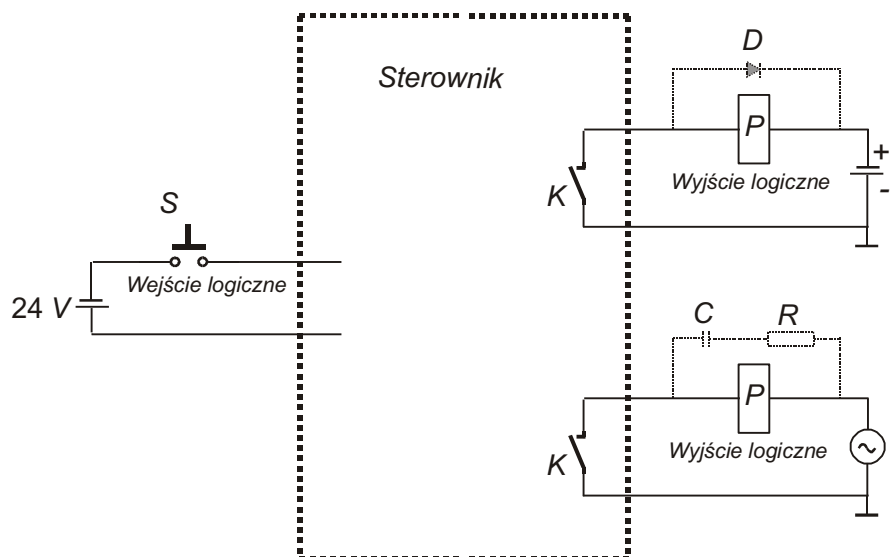
Rys. 1. Schemat ogólny stanowiska laboratoryjnego



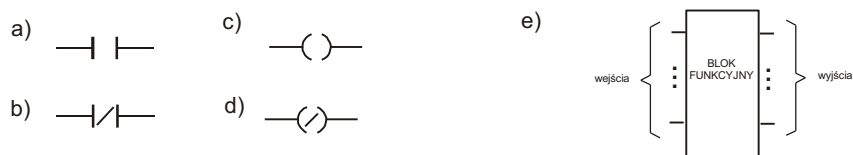
Rys.2. Ogólny schemat funkcjonalny sterownika



Rys. 3. Cykl pracy sterownika



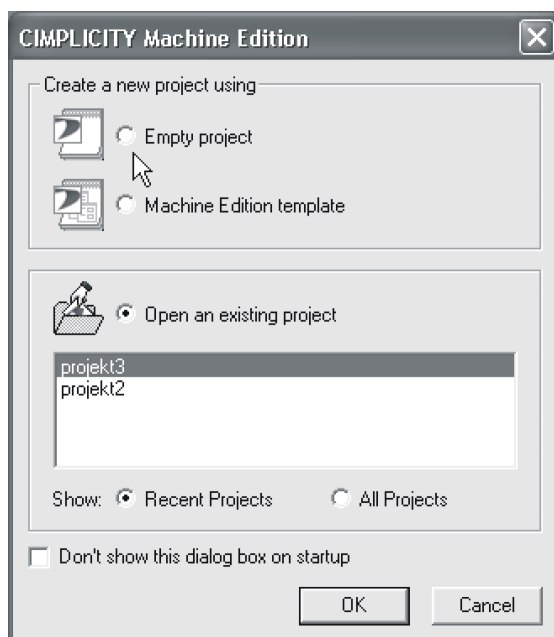
Rys. 4. Sposób podłączenia wejść i wyjść logicznych



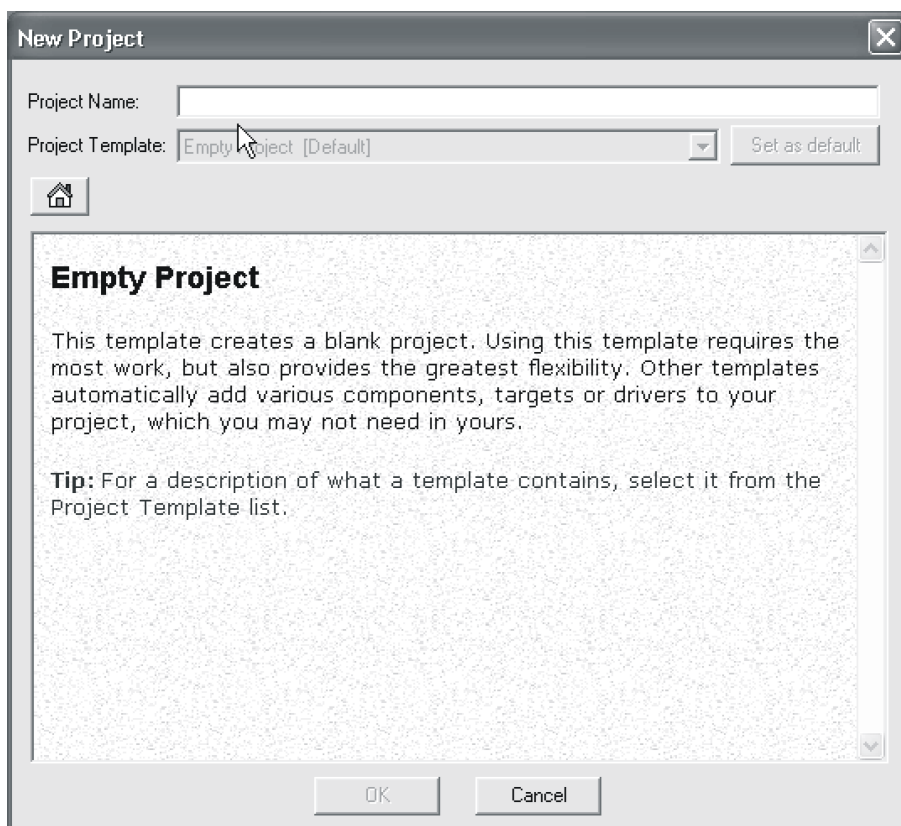
Rys. 5. Sposób przedstawiania styków (a - styk normalnie otwarty, b - styk normalnie zamknięty), przekaźników (c, d) i bloków funkcyjnych (e) w programie narzędziowym *Cimplicity*



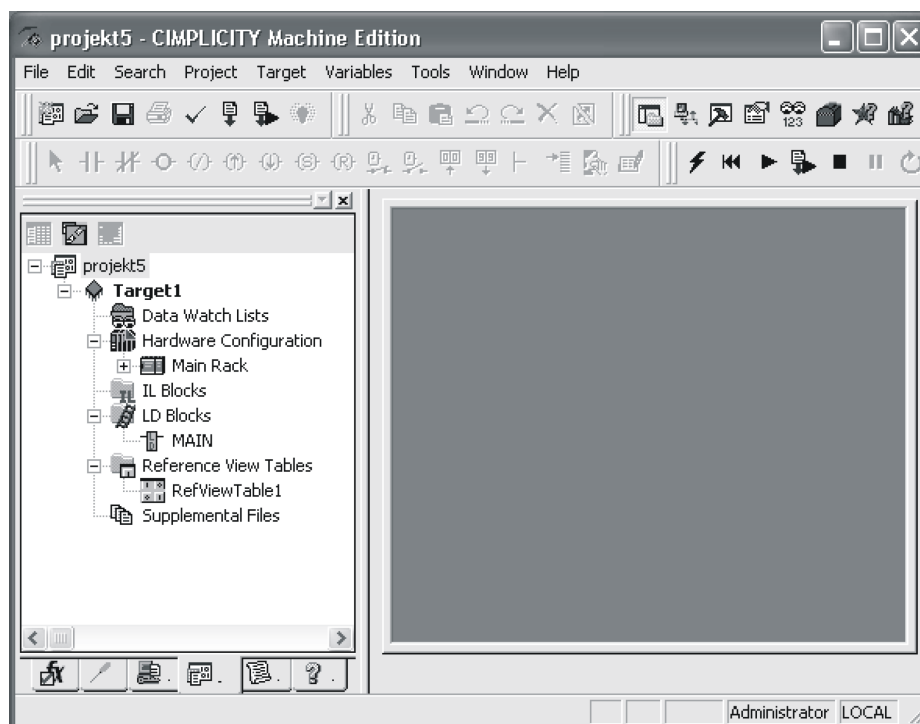
Rys. 6. Przykład szczebla drabiny logicznej



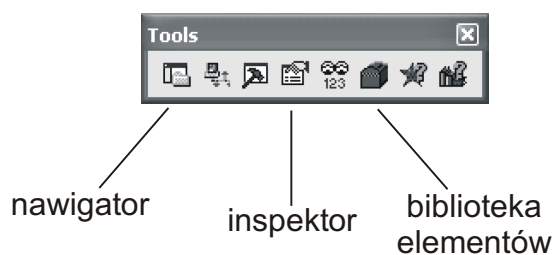
Rys.7. Okno wyboru istniejącego (lub rozpoczęcia nowego) programu sterującego



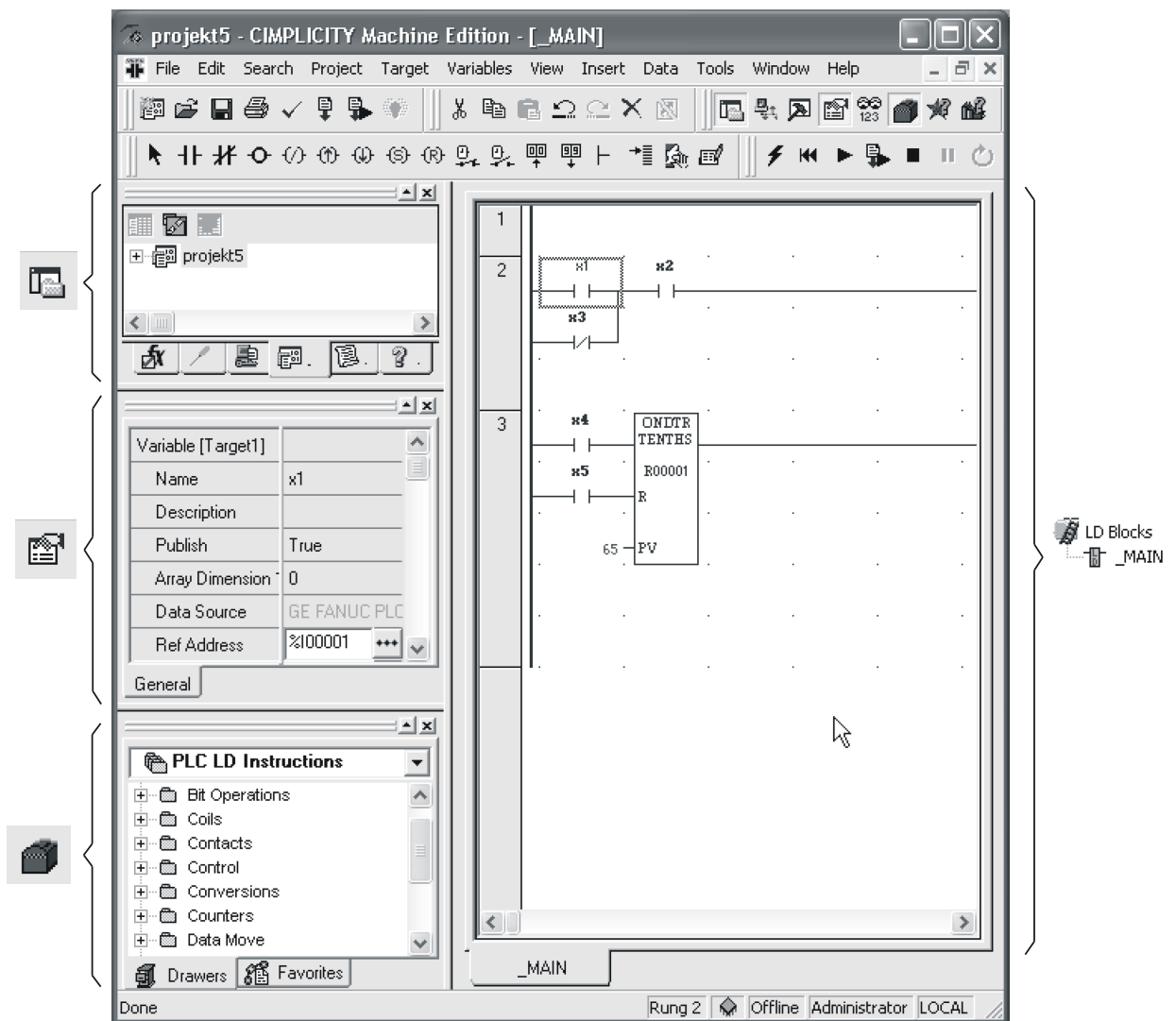
Rys. 8. Okno wpisywania nazwy nowego programu



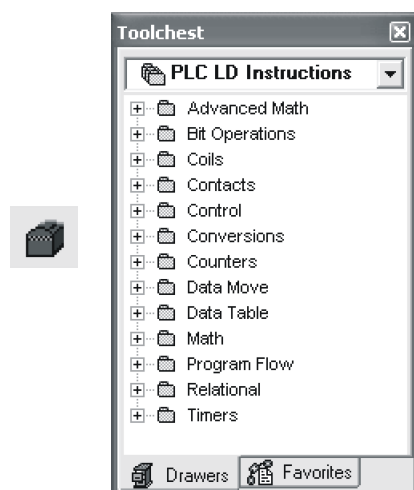
Rys. 9. Ekran roboczy programu *CIMPPLICITY* z aktywnym oknem *Nawigatora*



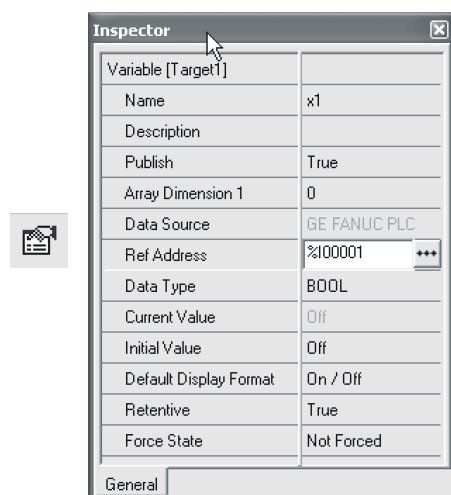
Rys. 10. Pasek narzędziowy okien pomocniczych



Rys. 11. Ekran roboczy programu *Cimplicity* (aktywne okno edycji programu sterującego oraz okna pomocnicze: *Nawigator*, *Biblioteka elementów*, *Inspektor*)



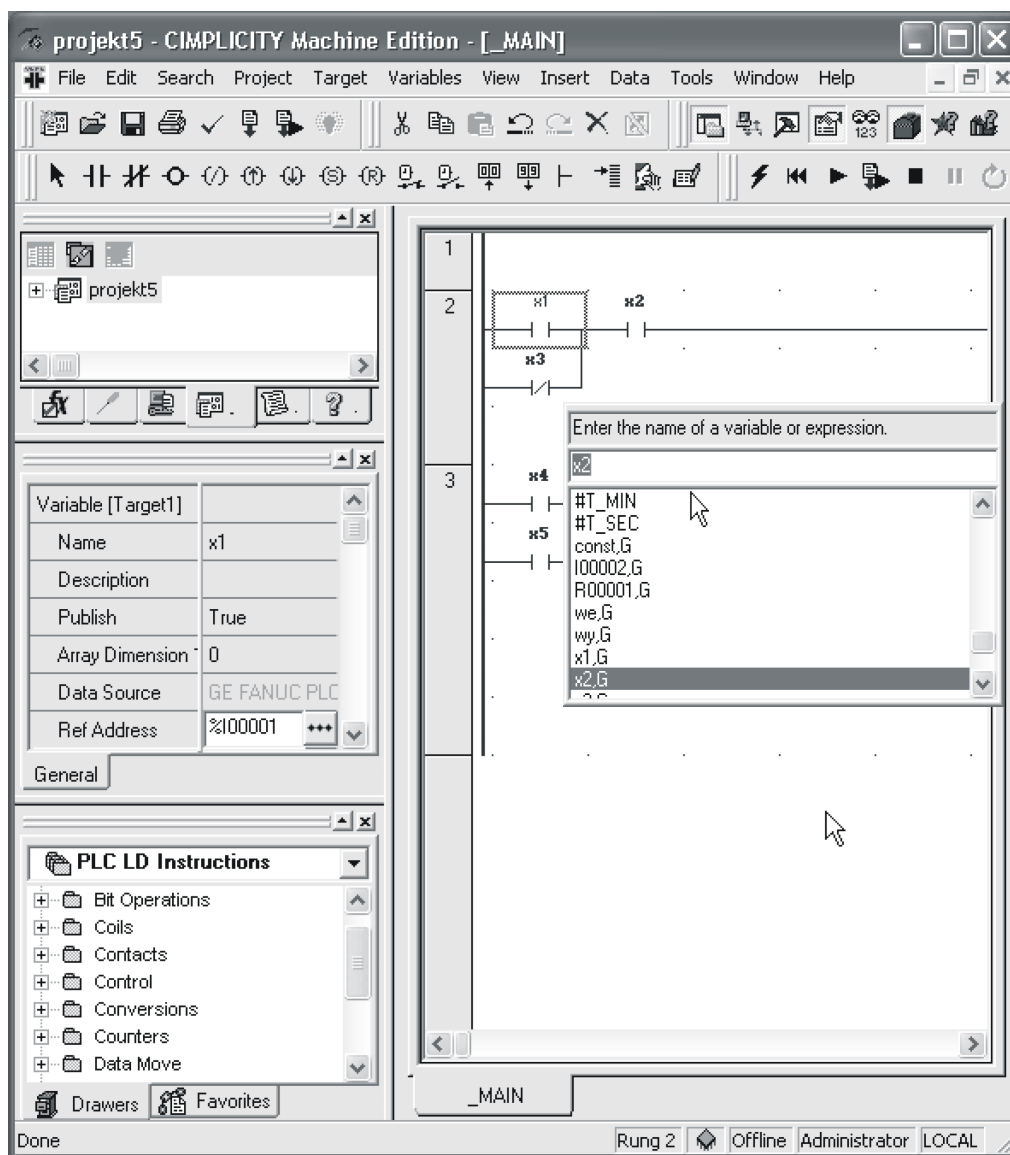
Rys. 12. Ekran pomocniczy *Biblioteka elementów*



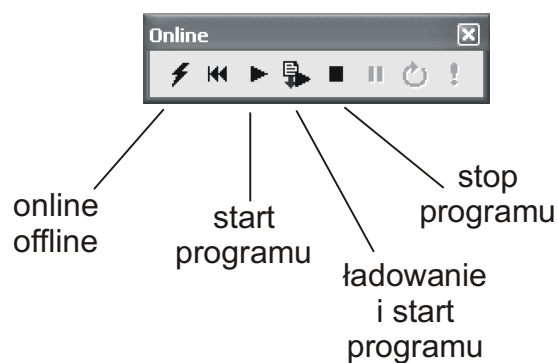
Rys. 13. Ekran pomocniczy *Inspektor*



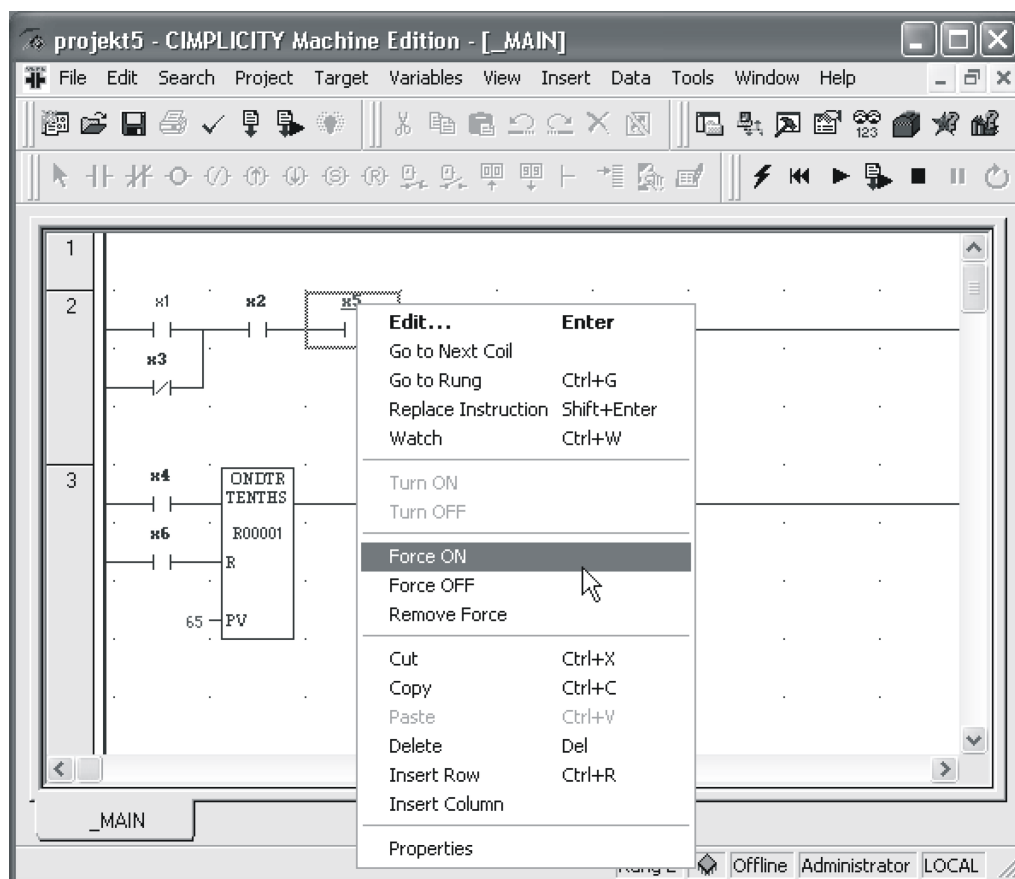
Rys.14. Pasek narzędziowy z wybranymi funkcjami oraz narzędziami budowy schematów drabinkowych



Rys. 15. Wpisywanie nazwy sygnału



Rys. 16. Pasek narzędziowy z poleceniami do uruchamiania programu sterującego



Rys. 17. Testowanie programu sterującego