

POLITECHNIKA ŚWIĘTOKRZYSKA

Wydział Elektrotechniki, Automatyki i
Informatyki

Katedra Elektrotechniki Przemysłowej i
Automatyki

Zakład Urządzeń i Systemów Automatyki

Urządzenia i Systemy Automatyki

Temat: **Programowanie sterownika PLC firmy
Fatek.**

Instrukcja laboratoryjna

1. Sterownik PLC

Programowalny sterownik logiczny PLC (*ang. programmable logic controller*) to urządzenie pozwalające na sterowanie pracą maszyn i procesów technologicznych według określonego algorytmu. Sterownik programowalny składa się z jednostki centralnej (CPU), modułów wejść oraz wyjść. Sterownik PLC cechuje praca cykliczna. Na początku cyklu pracy sprawdzany jest stan wejść sterownika następnie przetwarzane są kolejne instrukcje programu. Na końcu obiegu pętli w wyniku przetwarzania programu aktualizowany zostaje stan wyjść. Ze względu na złożoność budowy sterowników programowalnych wyróżnić można różne grupy tych urządzeń m.in. najprostsze przekaźniki programowalne, sterowniki kompaktowe oraz modułowe. Każdy z producentów oferujących sterowniki programowalne posiada w swej ofercie kilka różnych rodzin sterowników do różnych zastosowań.



Rysunek 1.1. Sterowniki PLC różnych producentów

1.1. Parametry sterownika firmy Fatek FBs-60MCJ2-D24

Sterownik FBs-60MCJ2-D24 to zaawansowane urządzenie programowalne firmy Fatek, które znajduje zastosowanie w różnorodnych aplikacjach automatyki przemysłowej. Dzięki swojej wszechstronności, wysokiej wydajności oraz rozbudowanym możliwościom komunikacyjnym, sterownik ten pozwala na precyzyjne sterowanie procesami

technologicznymi i urządzeniami. Poniżej przedstawiono kluczowe parametry techniczne tego modelu, które należy uwzględnić w projektowaniu i implementacji systemów sterowania.

Tabela 1.1 Parametry jednostki centralnej Fatek FBs-60MCJ2-D24

Dane podstawowe	
Seria	FBs-MC
Zasilanie	24VDC
Rozszerzalność o moduły prawostronne	Tak
Komunikacja	
Ilość portów komunikacyjnych	1
Protokoły komunikacyjne	RS232
Protokół	Fatek (slave i master)
Max portów komunikacyjnych	6
I/O (wejścia/wyjścia) cyfrowe	
Ilość I/O cyfrowych	36/24
Typ wyjść	Tranzystorowe PNP
Max szybkość I/O cyfrowych	200kHz
Ilość obsługiwanych osi (impulsowo)	4
Ilość szybkich liczników sprzętowych jednofazowych UP lub dwufazowych A/B	4 lub 4
Ilość I/O (wejść/wyjść) tranzystorowych	
200kHz	8/8
20kHz	0/0
5kHz (łącznie)	8/0
Małej szybkości	20/16
Maksymalna ilość	
I/O cyfrowych	256/256
I/O analogowych	66/68
Dane dodatkowe	
Wymiary	175x90x80mm
Mechanizm okablowania	Wymienna listwa zaciskowa 7,62
Zegar czasu rzeczywistego	Tak

1.2. Obwody zasilania sterowników Fatek

Sterowniki PLC firmy Fatek, takie jak modele z serii FBs, wymagają prawidłowego podłączenia obwodów zasilania, aby zapewnić niezawodne działanie jednostki głównej oraz podłączonych urządzeń. Zasilanie sterownika nie tylko umożliwia jego pracę, ale także zasila wejścia, wyjścia i ewentualne moduły rozszerzeń. Prawidłowe zaprojektowanie i wykonanie obwodów zasilających jest kluczowe dla stabilności całego systemu automatyki.

Zasada podłączenia obwodów zasilania sterownika:

1. Zasilanie główne:

- Sterownik wymaga napięcia zasilania stałego (DC), zwykle 24VDC, podłączanego do dedykowanych terminali (+24V i GND).
- Zasilanie może pochodzić z zewnętrznego zasilacza stabilizowanego, który powinien być dobrany odpowiednio do mocy wymagań sterownika i urządzeń współpracujących.

2. Zasilanie czujników i modułów rozszerzeń:

- Sterownik udostępnia wyjścia 24VDC przeznaczone do zasilania podłączonych czujników (np. PNP, NPN) lub innych urządzeń peryferyjnych.
- W przypadku rozbudowy systemu można korzystać z wyjść zasilania dostępnych na modułach rozszerzeń.

3. Ochrona przed przeciążeniami:

- Należy zadbać o właściwe zabezpieczenia, takie jak bezpieczniki lub ograniczniki prądowe, które ochronią sterownik przed uszkodzeniem w przypadku zwarcia lub przeciążenia w obwodach.

4. Uziemienie:

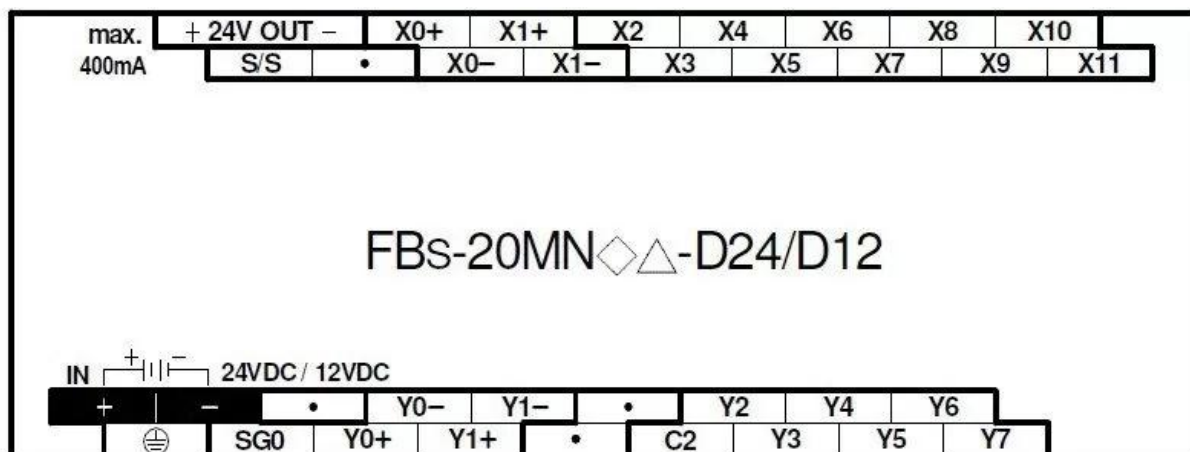
- Sterownik oraz wszystkie moduły powinny być prawidłowo uziemione (terminal PE), aby zapewnić ochronę przed zakłóceniami elektromagnetycznymi i przepięciami.

5. Konfiguracja wejść i wyjść:

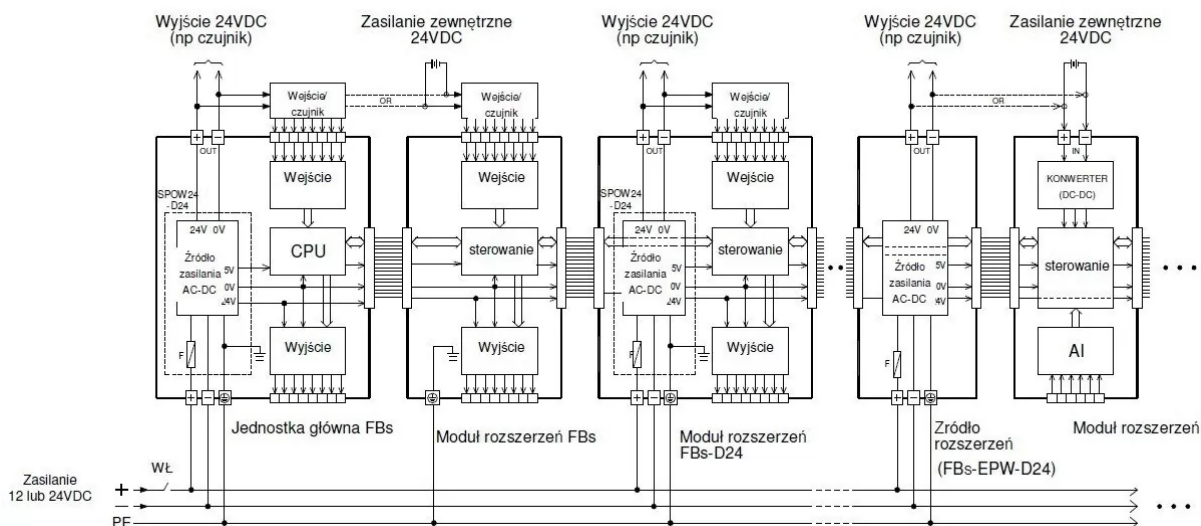
- Przy projektowaniu należy uwzględnić logikę podłączanych sygnałów wejściowych (PNP lub NPN) oraz wyjść (tranzystorowe, przekaźnikowe).
- Wspólne punkty odniesienia (S/S) należy prawidłowo połączyć z odpowiednim potencjałem masy systemu.

Uwagi praktyczne:

- Zawsze sprawdzaj zgodność napięcia zasilania ze specyfikacją sterownika, aby uniknąć uszkodzenia urządzenia.
- W przypadku dużej liczby podłączonych czujników i urządzeń wykonawczych należy upewnić się, że źródło zasilania ma wystarczającą wydajność prądową.
- Wszystkie połączenia powinny być wykonane zgodnie z załączonym schematem elektrycznym oraz instrukcją obsługi dostarczoną przez producenta sterownika.



Rysunek 1.2. Schemat wejść/wyjść sterownika[1].

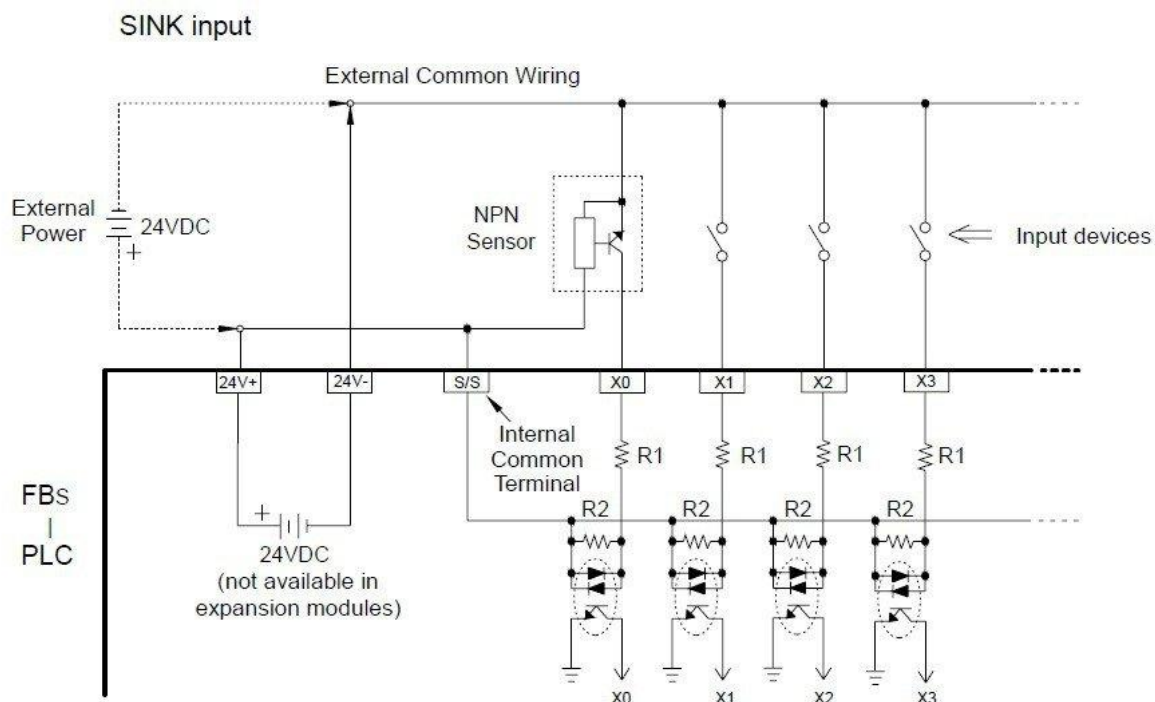


Rysunek 1.2. Schemat blokowy zasilania i rozbudowy sterownika[1].

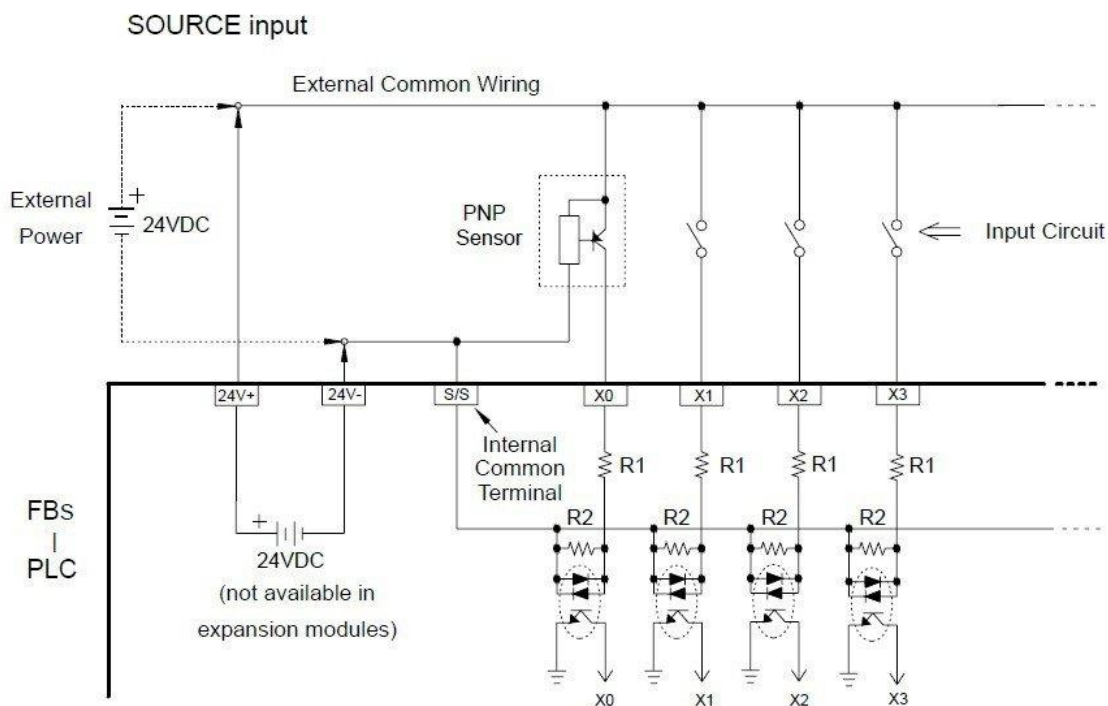
1.3. Obwody wejść/wyjść cyfrowych

Sterownik Fatek FBS-60MCJ2-D24 wyposażony jest w 36 wejść cyfrowych, które mogą być skonfigurowane zarówno w logice PNP (sourcing), jak i NPN (sinking). Oznacza to, że

użytkownik ma możliwość dostosowania konfiguracji wejść do specyficznych wymagań aplikacji, wybierając odpowiedni sposób podłączenia sygnałów wejściowych.

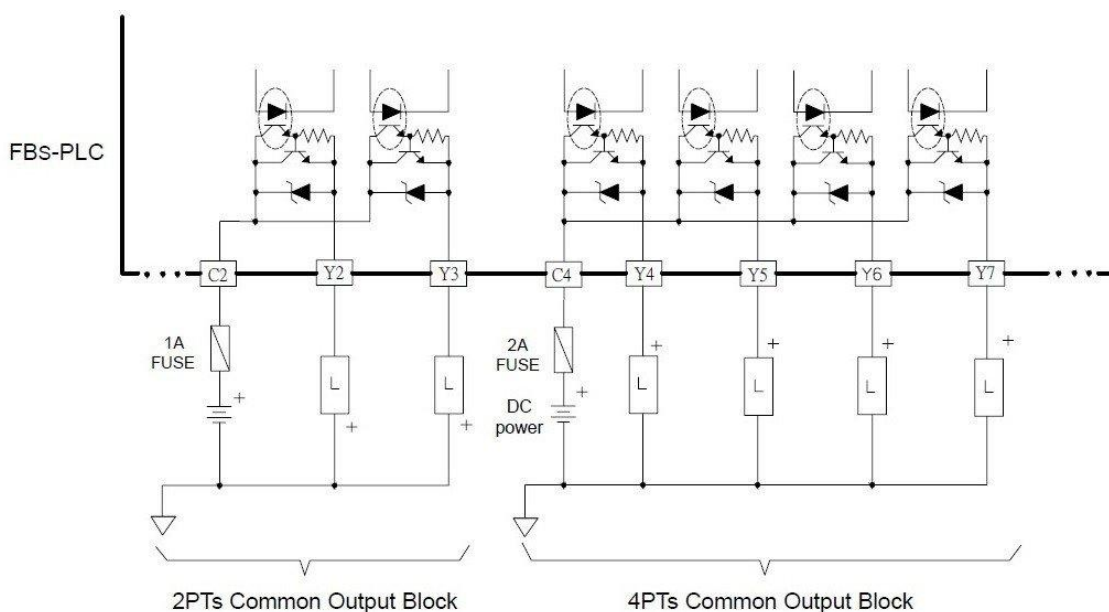


Rysunek 1.4. Wejścia 24VDC typu NPN (sink)[1].



Rysunek 1.5. Wejścia 24VDC typu PNP (source)[1].

Warto jednak zauważyć, że wyjścia cyfrowe w tym modelu są typu tranzystorowego PNP, co determinuje sposób ich podłączenia w układzie sterowania.



Rysunek 1.6. Wyjścia tranzystorowe PNP (source)[1].

2. Alokacja pamięci w sterownikach Fatek FBS-PLC

Sterowniki PLC serii FBs firmy Fatek oferują elastyczną i rozbudowaną strukturę pamięci, umożliwiającą efektywne zarządzanie danymi oraz precyzyjne sterowanie procesami automatyki. Zrozumienie alokacji pamięci w tych sterownikach jest kluczowe dla optymalnego projektowania aplikacji i pełnego wykorzystania ich możliwości.

Struktura pamięci:

Pamięć sterowników FBS-PLC podzielona jest na różne obszary, z których każdy pełni określoną funkcję:

Obszar dyskretnych stanów (bitowych):

- **Wejścia cyfrowe (X):** Adresy od X0 do X255, odpowiadające fizycznym wejściom cyfrowym.
- **Wyjścia cyfrowe (Y):** Adresy od Y0 do Y255, odpowiadające fizycznym wyjściom cyfrowym.
- **Przełączniki wewnętrzne (M):** Adresy od M0 do M2001, używane jako wewnętrzne markery do przechowywania stanów logicznych.
- **Przełączniki krokowe (S):** Adresy od S0 do S999, wykorzystywane w programowaniu sekwencyjnym.

- **Tymczasowe przekaźniki (TR):** Adresy od TR0 do TR39, stosowane w operacjach tymczasowych.
- **Kontakty statusowe timerów (T):** Adresy od T0 do T255, wskazujące status liczników czasowych.
- **Kontakty statusowe liczników (C):** Adresy od C0 do C255, wskazujące status liczników zdarzeń.

Obszar rejestrów danych (słowowych):

- **Rejestry danych (R):** Adresy od R0 do R3839, służące do przechowywania danych liczbowych i innych informacji.
- **Rejestry danych (D):** Adresy od D0 do D4095, pełniące podobną funkcję jak rejestry R.
- **Rejestry specjalne (SR):** Adresy od R3968 do R4167, przeznaczone do przechowywania specjalnych parametrów systemowych.
- **Rejestry tylko do odczytu (ROR):** Adresy od R5000 do R8071, zawierające dane konfiguracyjne ładowane przy starcie systemu.
- **Rejestry plikowe (F):** Adresy od F0 do F8191, używane do przechowywania większych bloków danych.
- **Rejestry wejściowe (IR):** Adresy od R3840 do R3903, odpowiadające zewnętrznym wejściom analogowym.
- **Rejestry wyjściowe (OR):** Adresy od R3904 do R3967, odpowiadające zewnętrznym wyjściom analogowym.

Podtrzymywanie danych:

Sterowniki FBs-PLC umożliwiają konfigurację podtrzymywania danych w przypadku zaniku zasilania:

- **Przekaźniki wewnętrzne (M):** Domyślnie M0–M799 są niepodtrzymywane, a M800–M1399 są podtrzymywane. Użytkownik może dostosować te zakresy według potrzeb.
- **Przekaźniki krokowe (S):** Domyślnie S0–S499 są niepodtrzymywane, a S500–S999 są podtrzymywane. Zakresy te również można konfigurować.
- **Rejestry danych (R):** Domyślnie R0–R2999 są podtrzymywane, a R3000–R3839 są niepodtrzymywane. Użytkownik ma możliwość zmiany tych ustawień.

Konfiguracja timerów i liczników:

- **Timery (T):** Dostępne są 256 timerów (T0–T255) z trzema podstawami czasu: 0,01 s, 0,1 s i 1 s. Użytkownik może przypisać odpowiednie timery do wybranej podstawy czasu w zależności od wymagań aplikacji.
- **Liczniki (C):** Sterownik oferuje 256 liczników (C0–C255), z możliwością wyboru między licznikami 16-bitowymi a 32-bitowymi.

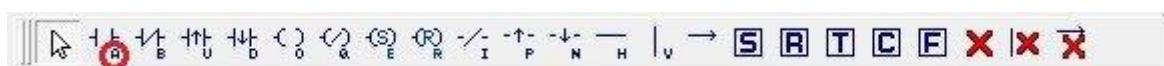
Uwagi praktyczne:

- **Rejestry tylko do odczytu (ROR):** Jeśli użytkownik skonfiguruje rejestry ROR, zawartość odpowiednich rejestrów (R5000–R8071) zostanie załadowana z ROR przy każdym uruchomieniu sterownika lub przejściu z trybu STOP do RUN.
- **Dostosowanie alokacji pamięci:** W oprogramowaniu WinProLadder użytkownik może konfigurować zakresy podtrzymywania dla przełączników i rejestrów, dostosowując je do specyficznych potrzeb aplikacji.

Zrozumienie i właściwe zarządzanie alokacją pamięci w sterownikach Fatek FBs-PLC pozwala na efektywne projektowanie aplikacji, optymalne wykorzystanie zasobów sprzętowych oraz zapewnienie niezawodnej pracy systemu automatyki.

3. Podstawowe instrukcje bitowe języka drabinkowego

Język drabinkowy (LD) jest jednym z najpopularniejszych języków programowania sterowników PLC, ze względu na swoją czytelność i analogię do klasycznych schematów przełącznikowych. Podstawowe instrukcje bitowe w tym języku umożliwiają realizację operacji logicznych na sygnałach dwustanowych, co stanowi fundament w projektowaniu układów sterowania.














Rysunek 3.1. Podstawowe instrukcje dostępne w programie WinProLadder.

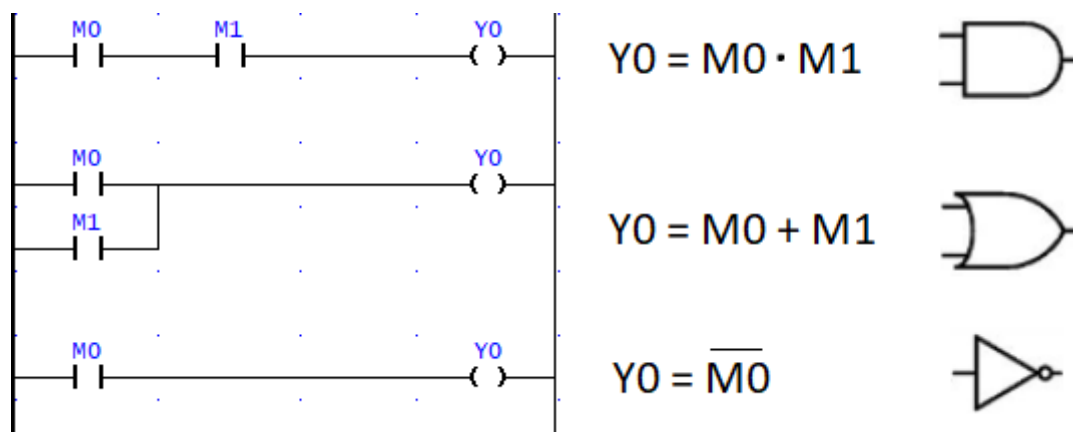
Aby wprowadzić dowolny element programu w oprogramowaniu WinProLadder, można skorzystać z paska narzędzi lub użyć odpowiedniego skrótu klawiaturowego. Każdy element posiada przypisaną literę, która jest wyświetlana obok jego ikony (np. litera "A" dla styku normalnie otwartego). Klikając w wybrane miejsce w oknie programu i naciskając odpowiednią literę na klawiaturze, wstawisz żądany element do schematu drabinkowego.

Opis palety:

	Styk normalnie otwarty (A)
	Styk normalnie zamknięty (B)

	Styk reagujący na zbocze narastające sygnału (U)
	Styk reagujący na zbocze opadające sygnału (D)
	Cewka normalnie otwarta (O)
	Cewka normalnie zamknięta (Q)
	Set (S)
	Reset (R)
	Zanegowanie linii (I)
	Stan narastający na linii (P)
	Stan opadający na linii (N)
	Linia pozioma (H)
	Linia pionowa (V)

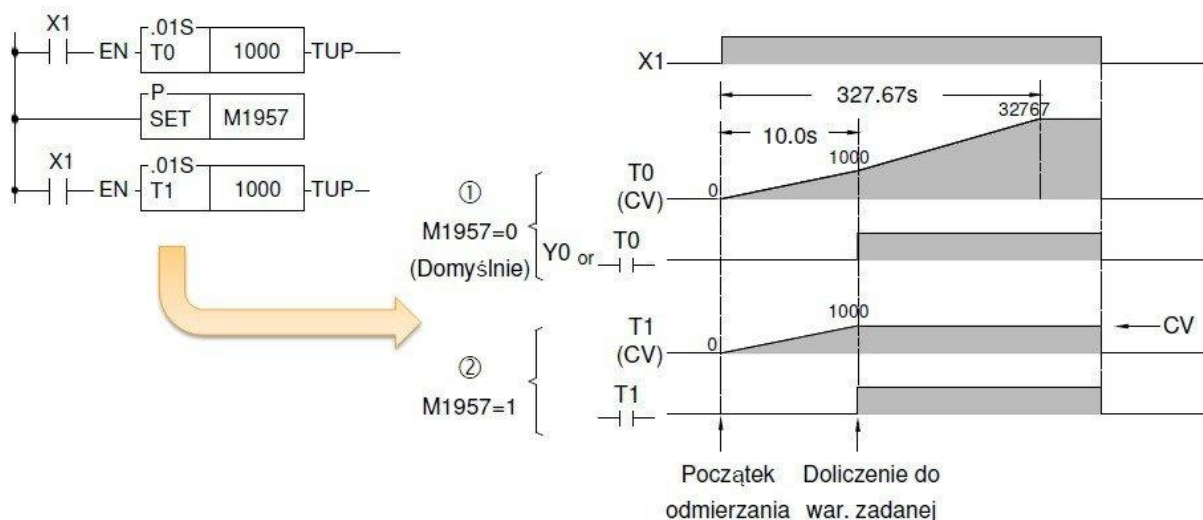
Program w języku drabinkowym wykonuje się szczebel po szczeblu. Symbole umieszczane w gałęziach odpowiadają elementom z układów przekaźnikowych np. wejście sterownika odpowiada stykowi przekaźnika natomiast wyjście sterownika cewce. Styk realizuje operacje mnożenia logicznego AND stanu z lewej strony styku oraz stanu zmiennej (w zależności od rodzaju styku z negacją lub bez) przypisanej do styku. Wynik tej operacji jest przekazywany do prawej strony styku. Cewka powoduje przypisanie stanu z jej lewej strony do zmiennej jej przypisanej np. bity wyjściowego lub markera. Na rysunku poniżej przedstawiono realizację prostych operacji logicznych z wykorzystaniem podstawowych elementów języka LAD.



Rysunek 3.2. Proste operacje logiczne w języku drabinkowym.

4. Układy czasowe – timery

Jedną z najczęściej wykorzystywanych poza elementarnymi operacjami bitowymi grupą instrukcji w języku LAD są timery. Pozwalają one na wprowadzanie w programie opóźnień oraz różnego rodzaju zależności czasowych. Timery służą do odliczania czasu, np. do pomiaru czasu, jaki upłynął od uruchomienia maszyny. Timer, skonfigurowany z odpowiednią podstawą czasu, rozpoczyna odliczanie po otrzymaniu sygnału o stanie wysokim na wejściu. Po osiągnięciu zadanej wartości PV (może to być stała lub zmienna), na jego wyjściu TUP pojawia się sygnał wysoki. Sygnał z wyjścia TUP można wykorzystać w dowolnym miejscu programu, stosując np. styk normalnie otwarty, podając jako zmienną numer timera, np. T10. Jeśli na wejściu timera zostanie podany sygnał niski, zliczana wartość CW zostanie automatycznie wyzerowana, a timer przygotuje się do kolejnego odliczania.



Rysunek 4.1. Przykładowe działanie timera[1].

Na powyższym rysunku zobrazowano sposób działania timerów w zależności od stanu zmiennej systemowej **M1957**, która kontroluje zachowanie wartości bieżącej (CV) timerów po osiągnięciu przez nie wartości zadanej:

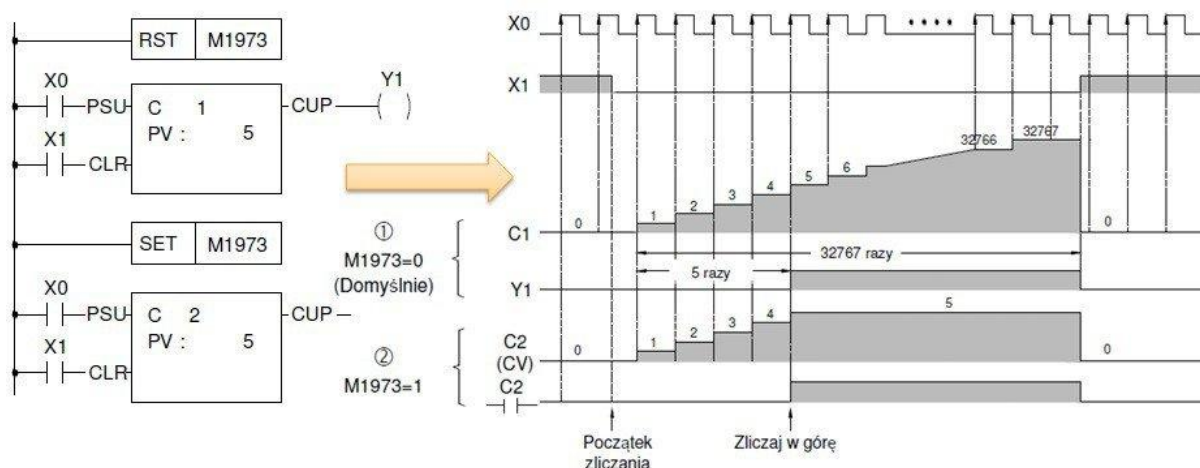
- **M1957 = 0:** Timer kontynuuje narastanie wartości bieżącej (CV) po odliczeniu czasu zadanego.
- **M1957 = 1:** Timer zatrzymuje narastanie wartości bieżącej (CV) po odliczeniu czasu zadanego.

Warto również zaznaczyć, że wartość bieżącą timera można wykorzystać w dowolnym miejscu programu, traktując ją jak rejestr. Odwołanie następuje poprzez podanie numeru timera, np. **T10**, co pozwala na elastyczne wykorzystanie informacji o stanie odliczania w logice sterowania.

5. Układy zliczające – liczniki

Instrukcje dotyczące liczników są stosowane do zliczania wewnętrznych oraz zewnętrznych zdarzeń. Liczniki służą do zliczania impulsów, co pozwala na monitorowanie np.

liczby wykonanych elementów. Licznik rejestruje impulsy na swoim wejściu **PLS**, które reaguje wyłącznie na zbocza narastające sygnału. Aby zliczyć impuls, na wejściu PLS musi pojawić się sygnał o stanie wysokim, po czym nastąpi przejście do stanu niskiego i ponownie wysokiego, co zarejestruje kolejny impuls. Po osiągnięciu wartości zadanej PV (może to być stała lub zmienna), licznik aktywuje sygnał wysoki na swoim wyjściu **CUP**. Sygnał ten można wykorzystać w dowolnym miejscu programu, stosując np. styk normalnie otwarty i podając jako zmienną numer licznika, np. **C10**. Dodatkowo, licznik posiada wejście **CLR**, które umożliwia resetowanie licznika. Po podaniu sygnału wysokiego na to wejście, wartość zliczona **CW** zostaje wyzerowana, a licznik jest gotowy do rozpoczęcia nowego cyklu zliczania.



Rysunek 5.1. Przykładowe działanie licznika[1].

Na powyższym rysunku zaprezentowano sposób działania liczników w zależności od stanu zmiennej systemowej **M1973**, która kontroluje zachowanie wartości bieżącej (CV) po osiągnięciu przez licznik wartości zadanej:

- **M1973 = 0:** Licznik kontynuuje narastanie wartości bieżącej (CV) po zliczeniu do wartości zadanej.
- **M1973 = 1:** Licznik zatrzymuje narastanie wartości bieżącej (CV) po osiągnięciu wartości zadanej.

Warto również zaznaczyć, że bieżącą wartość licznika można wykorzystać w dowolnym miejscu programu, traktując ją jako rejestr. Odwołanie do wartości bieżącej odbywa się przez podanie numeru licznika, np. **C10**, co pozwala na elastyczne wykorzystanie informacji o stanie licznika w logice sterowania.

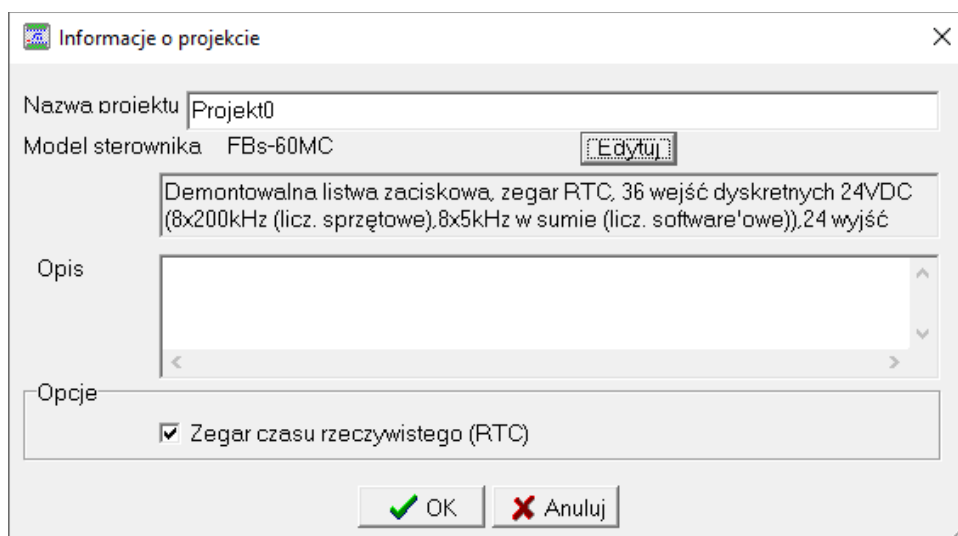
6. Tworzenie projektu w środowisku WinProLadder

WinProLadder to dedykowane środowisko programistyczne dla sterowników PLC firmy Fatek. Umożliwia ono tworzenie, edytowanie, symulację i wdrażanie programów w języku drabinkowym (Ladder Diagram). Dzięki intuicyjnemu interfejsowi, bogatemu zestawowi narzędzi i funkcjonalności, WinProLadder pozwala na efektywne projektowanie systemów automatyki przemysłowej. Proces tworzenia projektu w WinProLadder obejmuje

skonfigurowanie sterownika, definiowanie zasobów, takich jak wejścia, wyjścia i rejestry, oraz implementację logiki sterowania w postaci diagramu drabinkowego. W niniejszym punkcie omówione zostaną podstawowe kroki i zasady niezbędne do stworzenia projektu w tym środowisku.

Kroki tworzenia programu:

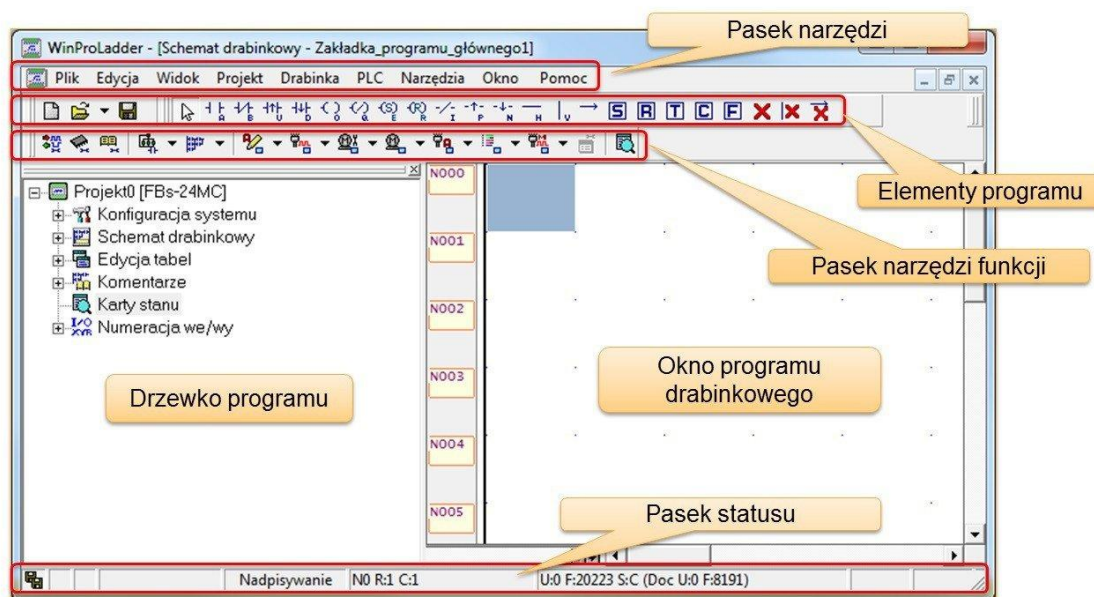
- 6.1. W zakładce **Plik** należy wybrać opcję **Nowy projekt**, a następnie kliknąć przycisk **[Edytuj]**, aby wybrać model sterownika, dla którego będzie tworzony projekt. W tym przypadku należy wybrać model **FBs-60MC**. Na koniec należy kliknąć przycisk OK.



Rysunek 6.1. Informacje o projekcie.

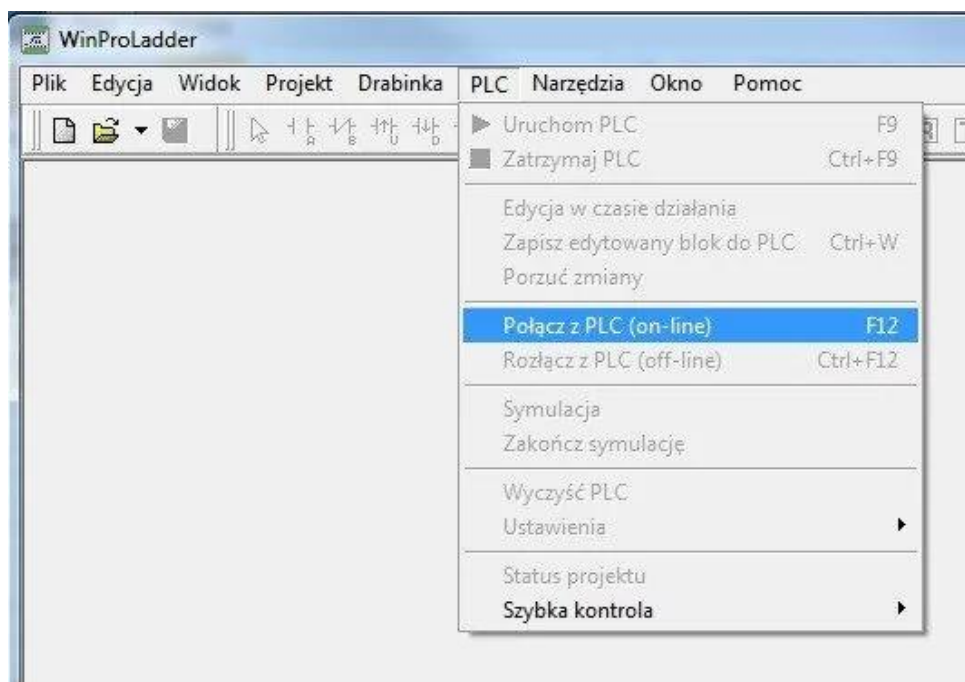
- 6.2. Okno główne programu WinProLadder stanowi centralne miejsce pracy, gdzie użytkownik może tworzyć, edytować i testować programy dla sterowników PLC. Główne elementy interfejsu to:
 - **Pasek menu:** Zawiera wszystkie dostępne funkcje programu, takie jak zarządzanie projektami, konfiguracja sterownika, narzędzia diagnostyczne i symulacja.
 - **Pasek narzędzi:** Umożliwia szybki dostęp do najczęściej używanych funkcji, takich jak wstawianie elementów drabinkowych, kompilacja czy przesyłanie programu do sterownika.
 - **Obszar edycji programu:** Centralna część okna, gdzie użytkownik tworzy schematy drabinkowe. Widok ten przedstawia logiczną strukturę programu w postaci graficznej.
 - **Zakładka stanu projektu:** Znajduje się na dole okna i wyświetla informacje o błędach, ostrzeżeniach oraz wynikach kompilacji programu.

- **Okna podrzędne:** Można w nich podglądać różne aspekty projektu, takie jak tabela zmiennych, konfiguracja zasobów czy parametry sterownika.



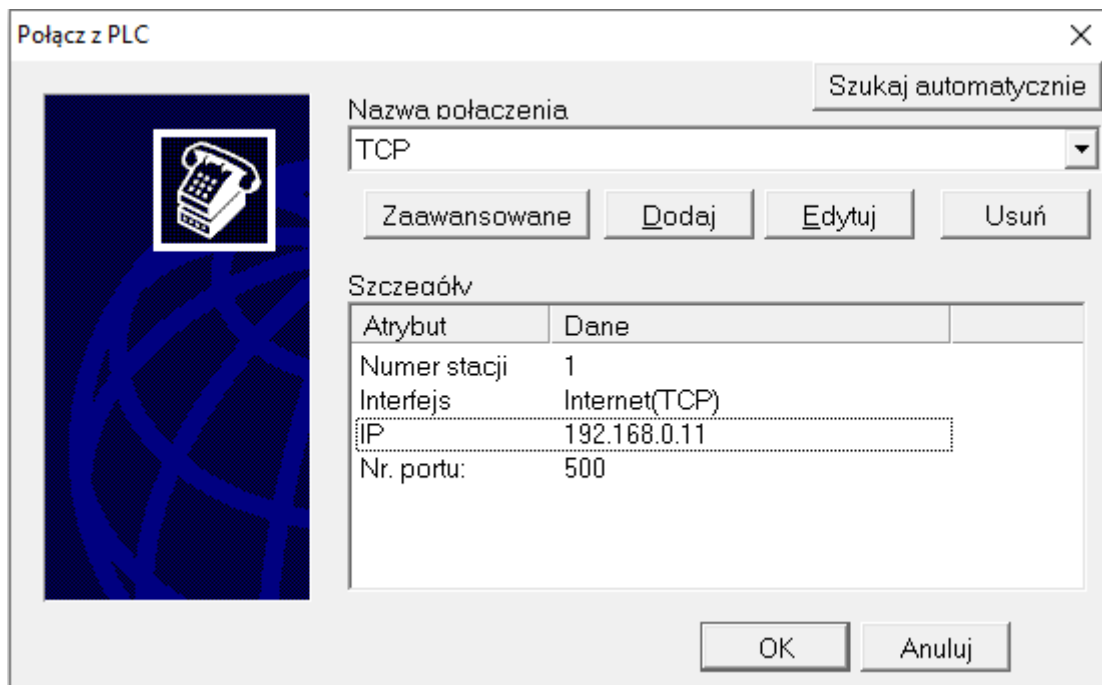
Rysunek 6.2. Okno główne programu.

- 6.3. Aby nawiązać połączenie ze sterownikiem Fatek przy użyciu programu WinProLadder, należy wybrać z zakładki PLC → Połącz z PLC (on-line).



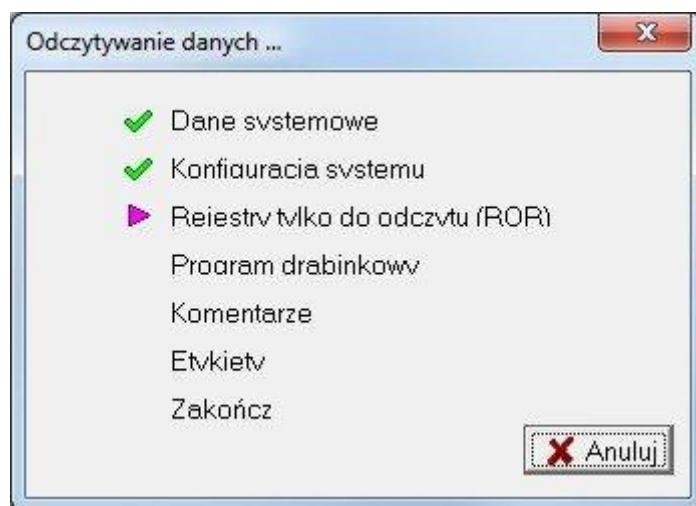
Rysunek 6.3. Połącz ze sterownikiem.

- 6.4. W sekcji **Nazwa połączenia** wybrać **TCP/IP**, wprowadzić adres IP sterownika Fatek (adres domyślny sterownika (np. 192.168.1.13), numer portu (np. 502) i zatwierdzić przyciskiem OK.



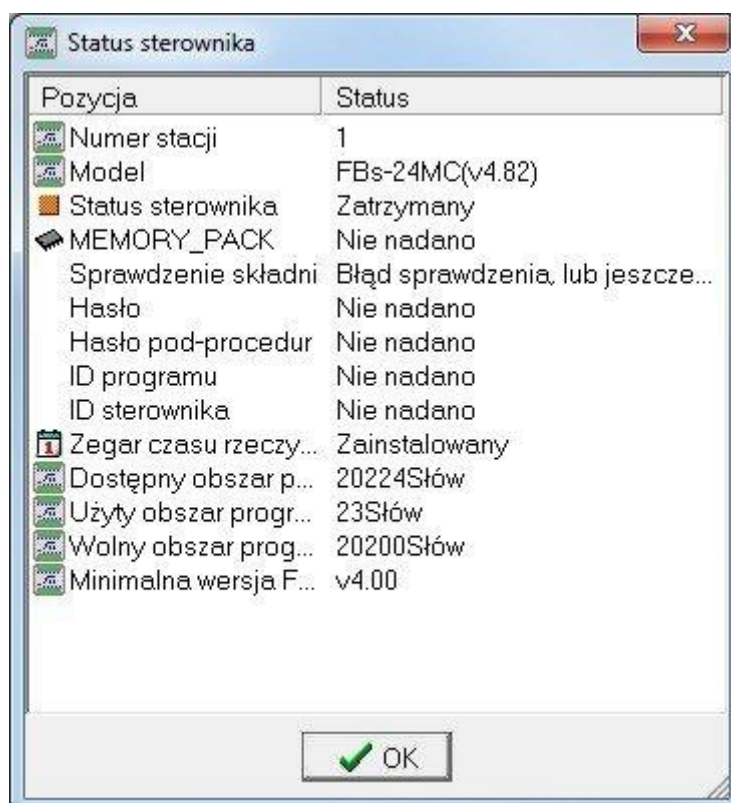
Rysunek 6.4. Wybór połączenia.

- 6.6. Okno dialogowe pokazujące postęp odczytywania, lub w przypadku wybrania opcji wgrywania postępu zapisu danych do sterownika:



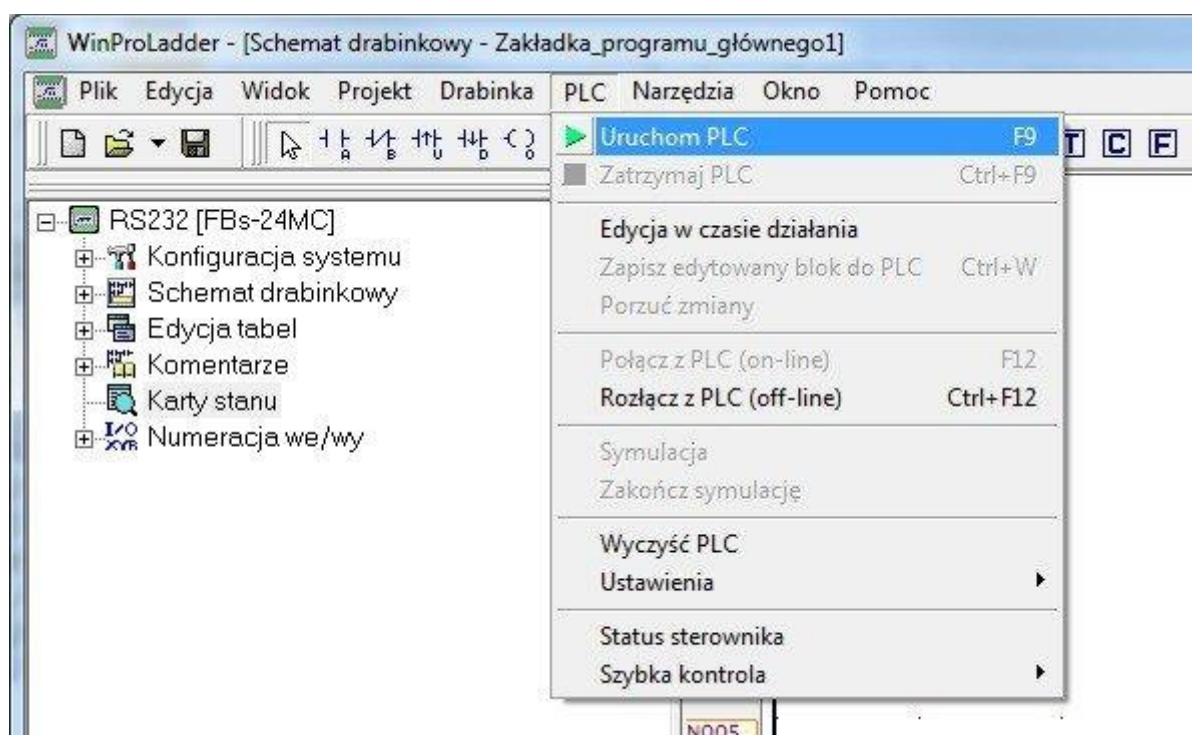
Rysunek 6.5. Okno odczytywanie danych.

- 6.7. Po zakończeniu procedury pojawi się kolejne okno pokazujące status sterownika, w tym miejscu kończy się procedura:



Rysunek 6.6. Status sterownika.

- 6.7. Aby program rozpoczął działanie, należy wejść w zakładkę PLC i wybrać opcję Uruchom PLC:



Rysunek 6.7. Uruchom PLC.

7. Zadania do wykonania

Zadanie 1: Sterowanie diodą LED

Opis: Napisz program, który będzie sterował diodą LED w zależności od stanu przycisku. Jeśli przycisk jest wciśnięty (stan wysoki), dioda LED świeci, jeśli przycisk jest puszczony (stan niski), dioda gaśnie.

Wskazówki:

- Wejście: przycisk (X0)
- Wyjście: dioda LED (Y0)
- Wykorzystaj prostą logikę AND lub IF.

Zadanie 2: Włączanie silnika na określony czas

Opis: Napisz program, który uruchomi silnik po wciśnięciu przycisku, a silnik będzie pracował przez 10 sekund po puszczeniu przycisku. Użyj timera.

Wskazówki:

- Wejście: przycisk (X0)
- Wyjście: silnik (Y0)
- Timer (czas opóźnienia wyłączenia silnika).

Zadanie 3: Sterowanie silnikiem z podtrzymaniem

Opis: Napisz program, który pozwala na włączenie silnika po jednokrotnym naciśnięciu przycisku START i jego wyłączenie po naciśnięciu przycisku STOP.

Wskazówki:

- Wejście: przycisk START (X0), przycisk STOP (X1)
- Wyjście: silnik (Y0)
- Wykorzystaj cewkę set oraz reset.

Zadanie 4: Sterowanie sekwencyjne trzech lampek

Opis: Napisz program, który włącza trzy lampki w sekwencji, każda na 2 sekundy, a następnie wszystkie gasną. Proces powtarza się po 10 sekundach przerwy.

Wskazówki:

- Wyjścia: lampka 1 (Y0), lampka 2 (Y1), lampka 3 (Y2)
- Użyj timerów, by kontrolować czas świecenia każdej lampki.

Zdanie 5: Sterowanie windą na dwa piętra

Opis: Zaprojektuj prosty system sterowania windą, która porusza się między dwoma piętrami. Jeśli wciśnięto przycisk na pierwszym piętrze, winda zjeżdża na dół. Jeśli wciśnięto przycisk na drugim piętrze, winda wjeżdża do góry.

Wskazówki:

- Wejścia: przycisk na pierwszym piętrze (X0), przycisk na drugim piętrze (X1)
- Wyjścia: silnik winda w dół (Y0), silnik winda w górę (Y1)
- Dodaj czujniki krańcowe, które zatrzymają windę na odpowiednim piętrze (czujnik krańcowy pierwsze piętro X2, czujnik krańcowy drugie piętro X3).

Zadanie 6: Sterowanie taśmociągiem

Opis: Napisz program, który steruje pracą taśmociągu w taki sposób, że jeśli wciśnięty zostanie przycisk START, taśmociąg zaczyna pracować. Kiedy wciśnięty zostanie przycisk STOP, taśmociąg zatrzymuje się.

Wskazówki:

- Wejścia: przycisk START (X0), przycisk STOP (X1)
- Wyjście: silnik taśmociągu (Y0)
- Użyj cewki set oraz reset.

Zadanie 7: Liczenie przedmiotów na taśmociągu (rozszerzenie zadania 8)

Opis: Napisz program, który liczy liczbę przedmiotów przechodzących przez czujnik na taśmociągu.

Wskazówki:

- Wejście: czujnik (X2)
- Użyj licznika.

Zadanie 8: Rozruch silnika gwiazda trójkąt

Opis: Napisz program sterujący rozruchem silnika trójfazowego w układzie gwiazda-trójkąt. Układ ten działa w ten sposób, że na początku silnik jest zasilany w konfiguracji gwiazdy, co zmniejsza prąd rozruchowy, a po określonym czasie przełącza się na trójkąt, aby silnik mógł pracować z pełną mocą.

Wskazówki:

Wejścia:

- Przyciski:
 - Start (X0)
 - Stop (X1)

Wyjścia:

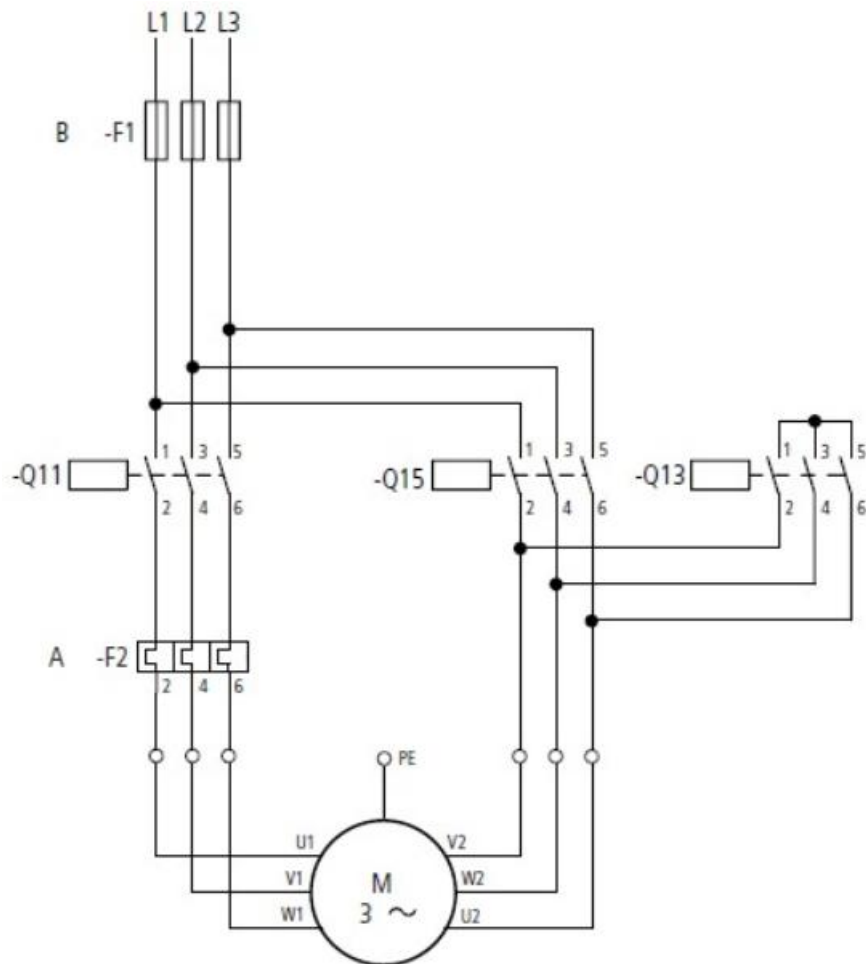
- Przekaznik gwiazda (Y0) – załącza stycznik w konfiguracji gwiazdy
- Przekaznik trójkąt (Y1) – załącza stycznik w konfiguracji trójkąta
- Przekaznik główny (Y2) – załącza główny stycznik zasilania silnika

Działanie:

1. Po naciśnięciu przycisku Start:
 - Główny stycznik (Y2) zostaje załączony (podanie zasilania na silnik).
 - Stycznik gwiazda (Y0) załącza silnik w układzie gwiazdy.
 - Timer 1 liczy czas rozruchu (np. 5-10 sekund).
2. Po zakończeniu odliczania przez Timer 1:
 - Stycznik gwiazda (Y0) zostaje wyłączony.
 - Timer 2 uruchamia przerwę między zmianą gwiazda a trójkąt (np. 1-2 sekundy).
3. Po zakończeniu Timer 2:

- Stycznik trójkąt (Y1) zostaje załączony, a silnik pracuje w konfiguracji trójkąta.
4. Naciśnięcie przycisku Stop:
- Wyłącza wszystkie styczniki (Y0, Y1, Y2) i zatrzymuje silnik.

Schemat poglądowy:



Bibliografia:

5. Fatek bez tajemnic [www.multiprojekt.pl]