

PLC

Podstawy GX Works2

Niniejszy kurs szkoleniowy (e-learning) został opracowany z myślą o osobach pierwszy raz korzystających z oprogramowania GX Works2 w celu tworzenia programów sekwencyjnych.

Niniejszy kurs zapewnia podstawową wiedzę z zakresu korzystania z oprogramowania GX Works2 w celu programowania, wyszukiwania i usuwania błędów i sprawdzania operacji sterownika programowalnego (PLC). Kurs jest przeznaczony dla osób, które chcą tworzyć programy sekwencyjne dla sterowników serii MELSEC-Q, MELSEC-L oraz MELSEC-F.

Treść tego kursu posiada następującą strukturę.
Zalecamy rozpoczęcie od Rozdziału 1.

Rozdział 1 – Metoda sterowania systemu PLC

W niniejszym rozdziale został opisany język programowania i oprogramowanie stosowane podczas programowania.

Rozdział 2 – Projektowanie programu

Z rozdziału tego można dowiedzieć się, w jaki sposób zaprojektować program w oparciu o elementy sterowania i konfigurację sprzętu.

Rozdział 3 – Programowanie

W rozdziale tym zawarte zostały informacje o sposobie korzystania z programu dedykowanego oprogramowania GX Works2.

Rozdział 4 – Wyszukiwanie i usuwanie błędów

W rozdziale tym zawarte zostały informacje o sposobie zapisywania programów sekwencyjnych w module CPU oraz ich przeszukiwania w celu wyszukania i usunięcia błędów.

Rozdział 5 – Test końcowy

Ocena zaliczająca: 60% lub wyższa.

Przejdź do następnej strony		Przejdź do następnej strony.
Wróć do poprzedniej strony		Wróć do poprzedniej strony.
Przejdź do żądanej strony		Wyświetli się „Spis treści”, umożliwiający przejście do żądanej strony.
Zakończ naukę		Zakończ naukę. Okna takie jak okno „Zawartość” zostaną zamknięte i nauka zostanie zakończona.

Zalecenia dotyczące bezpieczeństwa

Jeśli uczysz się, korzystając z rzeczywistych produktów, prosimy o dokładne przeczytanie zasad bezpieczeństwa zawartych w odpowiednich instrukcjach obsługi.

Środki ostrożności dla tego kursu

- Ekrany wyświetlane dla wersji oprogramowania, którego używasz, mogą się różnić od przedstawionych w tym kursie.

Rozdział 1 Metoda sterowania systemu PLC

Kurs ten jest przeznaczony dla osób pracujących z oprogramowaniem inżynierskim. Obejmuje on podstawowe założenia zarządzania systemami serii MELSEC-Q, L i F.

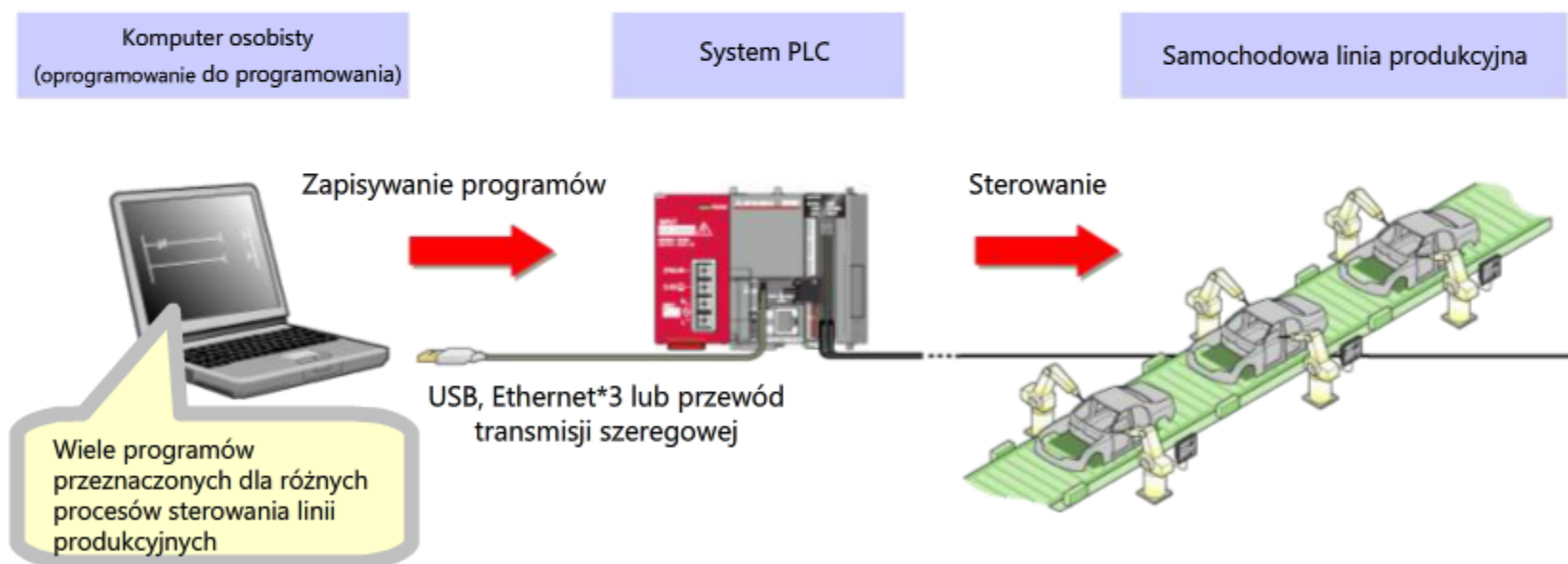
GX Works 2 (GXW2) wykorzystuje międzynarodowe ustandaryzowane języki programowania, włączając sekwencyjny język graficzny Sequential Function Chart (SFC), język listy instrukcji Instruction List (IL)*1, język logiki drabinkowej Ladder Logic, język schematów bloków funkcyjnych Function Block Diagram (FBD)*2 oraz język tekstu strukturalnego Structured Text (ST).

Programy są opracowywane przy użyciu komputera osobistego z „opracowaniem inżynierskim” GX works2 i zazwyczaj są zapisywane na sterowniku programowalnym CPU przy użyciu portu USB, przewodu Ethernet*3 lub przewodu do transmisji szeregowej. Moduł CPU może być programowany wielokrotnie, zgodnie z potrzebą, w celu dopasowania do dowolnie wymaganych zmian w danym sterowaniu.

*1 Przyszłe plany GX works2.

*2 Obecnie nazywany w GX works2 Structured Ladder (Drabinka struktury), planowane zapewnienie zgodności z IEC.

*3 Ethernet jest zarejestrowanym znakiem towarowym Xerox Corp.



W ramach niniejszego kursu w przykładowym programie użyto języka logiki drabinkowej (jednego z najpopularniejszych języków programowania PLC). W przykładzie tym zastosowano sterownik PLC serii L, jednakże zakres kursu w równym stopniu odnosi się również do systemów serii Q.

Podstawowa metoda sterowania jest taka sama również w przypadku serii MELSEC-F, jednak niektóre operacje i funkcje różnią się.

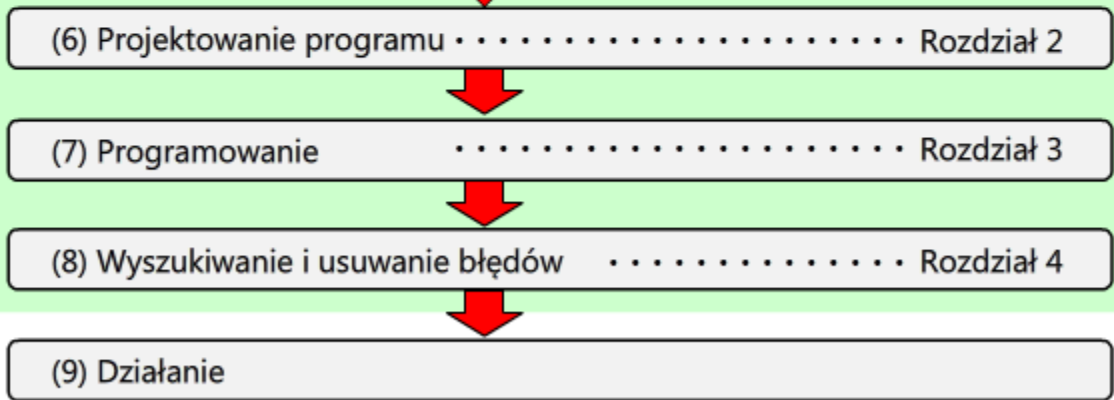
1.1 Procedura dotycząca konstrukcji systemu PLC

Niniejszy kurs e-learningowy obejmuje poszczególne kroki projektowania oprogramowania konieczne do wdrażania systemu sterownika programowalnego.

Projektowanie sprzętu



Projektowanie oprogramowania



**Zakres
niniejszego
kursu**

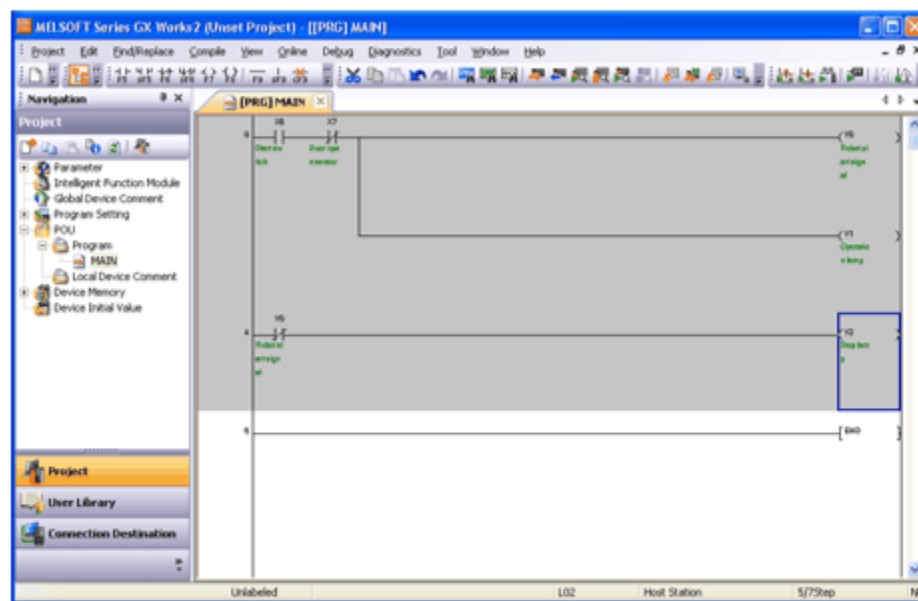
1.2

Wymagania dotyczące programowania

W ramach niniejszego kursu skupimy się na sposobie zastosowania oprogramowania inżynierskiego GX Works2 sterownika programowalnego w celu opracowania przykładowego programu.

Poniżej wymieniono kilka podstawowych funkcji GX Works2.

- Zarządzanie pamięcią i plikami
- Opracowywanie programów sterownika programowalnego
- Zarządzanie dokumentacją programu (komentarze itp.)
- Odczytywanie/zapisywanie danych (w szczególności programów) z/do modułu CPU
- Weryfikacja działania programu
 - Symulacja oprogramowania sprzętu PLC
 - Wymuszanie wł. lub wył. we/wy
 - Monitorowanie wejść/wyjść i statusu pamięci
- Przeprowadzanie obowiązkowych czynności konserwacji i rozwiązywania problemów

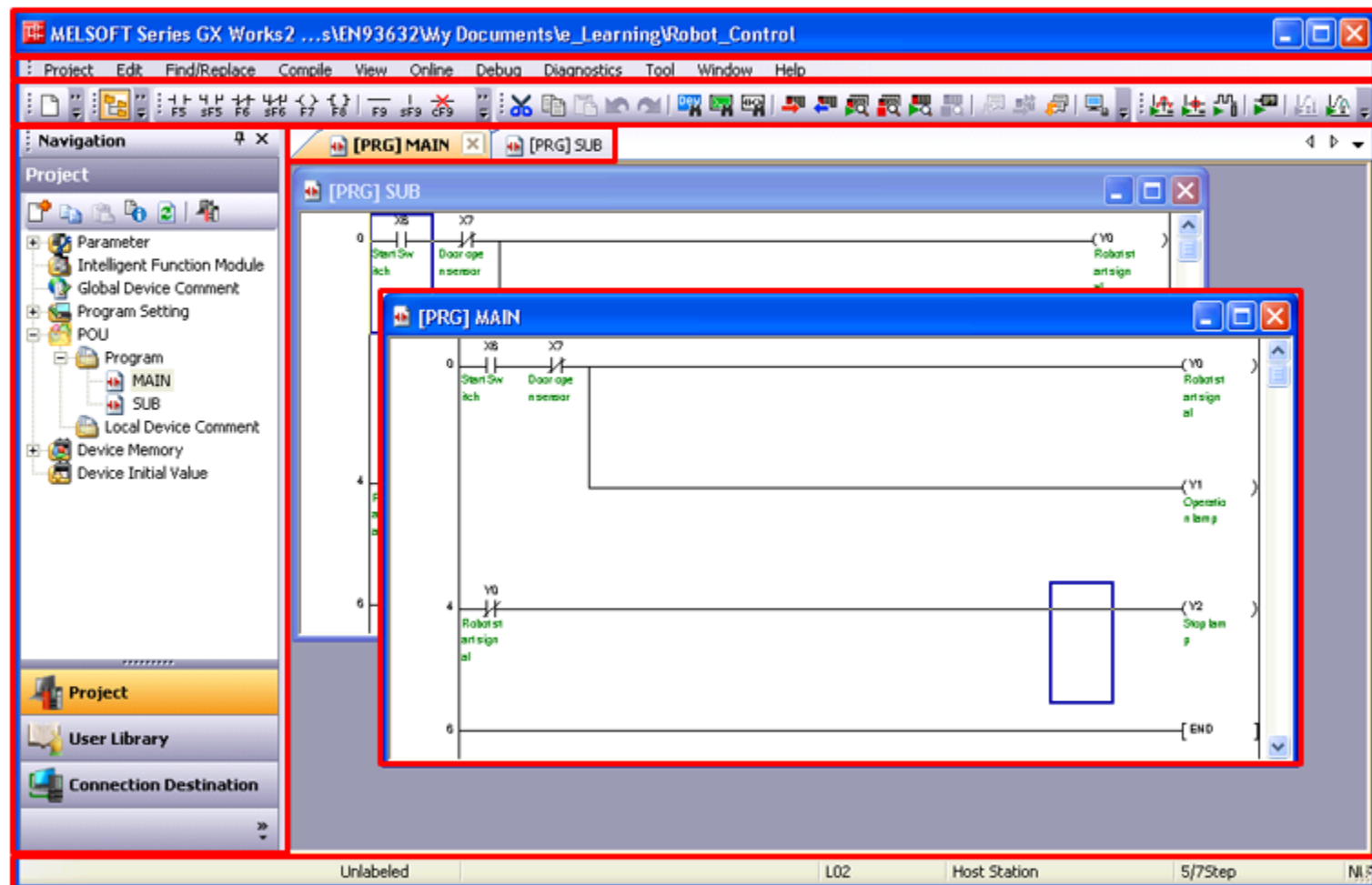


1.3

Konfiguracja ekranu GX Works2

Poniżej przedstawiono konfigurację ekranu GX Works2.

Umieść kursor myszki w czerwonej ramce, aby wyświetlić odpowiednią funkcję.



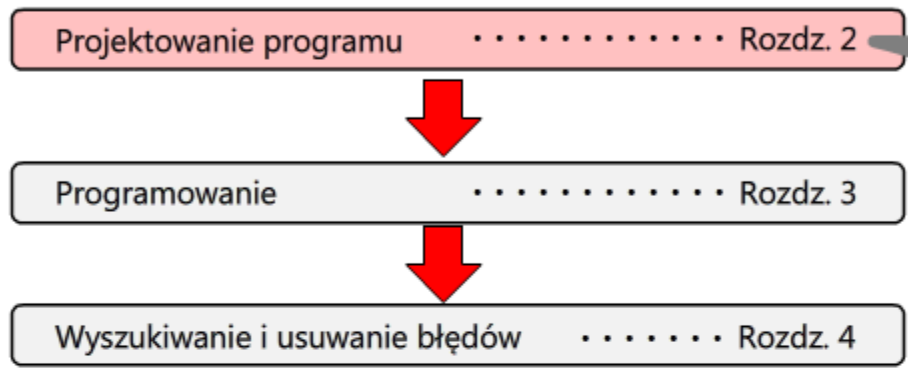
1.4**Procedura tworzenia programu sekwencyjnego**

Utwórz program sekwencyjny zgodnie z następującą procedurą.



Rozdział 2 Projektowanie programu

W Rozdziale 2 opisano, w jaki sposób projektować programy, włączając definiowanie zawartości elementów sterowania i konwertowania jej do programu.



Kroki procesu uczenia w Rozdziale 2

- 2.1 Konfiguracja sprzętu w ramach przykładowego systemu zastosowanego do nauki
- 2.2 Definiowanie elementów sterowania
- 2.3 Tworzenie tabeli zbiorczej urządzeń we/wy i numerów urządzeń
- 2.4 Projektowanie programowania

>>

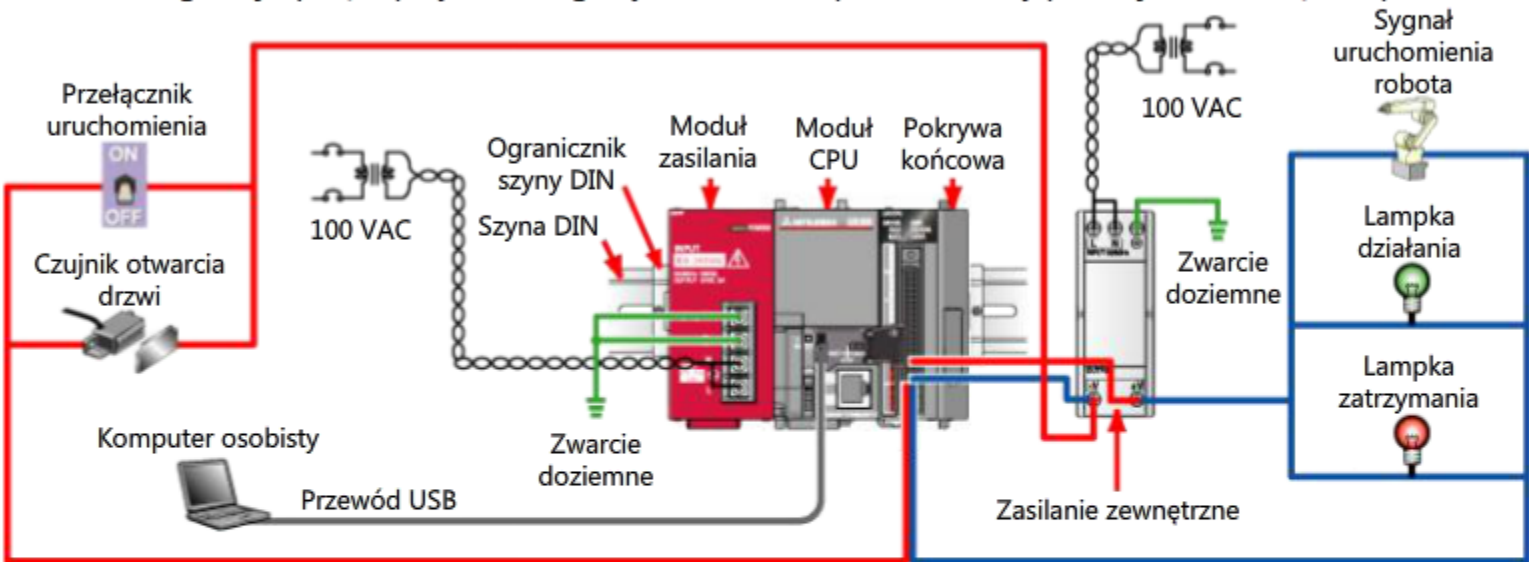
2.1

Konfiguracja sprzętu dla przykładowego systemu zastosowanego do nauki

TOC

W ramach tego kursu uczestnicy będą projektować system PLC (zwany dalej „przykładowym systemem”), który uruchamia robota zgodnie z procedurą.

Schemat konfiguracji sprzętu przykładowego systemu został przedstawiony poniżej wraz z listą komponentów sprzętowych.



Pozycja	Komponent	Model	Opis
System PLC	Moduł zasilania	L61P	Doprowadza zasilanie do modułów, włączając moduł CPU i moduł we/wy.
	Moduł CPU	L02CPU	Steruje systemem PLC.
	Pokrywa końcowa	L6EC	Zamocowana z prawej strony bloku systemu.
	Przewód USB	MR-J3USBCBL3M	Łączy komputer osobisty, na którym zainstalowano GX Works2, z modułem CPU.
	Komputer osobisty	-	Zapewnia pracę na zainstalowanym oprogramowaniu GX Works2.
Zasilanie zewnętrzne	-	-	Doprowadza zasilanie do zewnętrznych urządzeń we/wy.
Zewnętrzny sprzęt we/wy	Przełącznik	-	Umożliwia ustawienie w położeniu ON (WŁ.) w celu uruchomienia systemu sterowania.
	Czujnik	-	Wykrywa, czy drzwi są otwarte, czy zamknięte.
	Robot	-	Pracuje zgodnie z otrzymywanymi sygnałami sterującymi.
	Dwie lampki	-	Świecą się w zależności od statusu operacji.

2.2

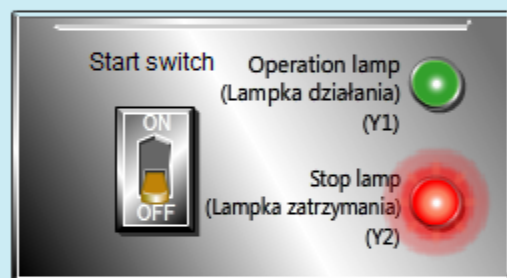
Definiowanie elementów sterowania

Pierwszym krokiem w projektowaniu programu jest zidentyfikowanie urządzeń, które mają podlegać sterowaniu oraz urządzeń we/wy wymaganych do danego sterowania. W przykładowym systemie sterowanie będzie polegać na uruchamianiu i zatrzymywaniu operacji robota. Robot będzie blokowany przed uruchomieniem, jeśli drzwi ogrodzenia zabezpieczającego będą otwarte, oraz będzie zatrzymywany w przypadku otwarcia drzwi podczas wykonywania operacji. Patrz animacja poniżej w celu lepszego zrozumienia sposobu pracy przykładowego systemu.

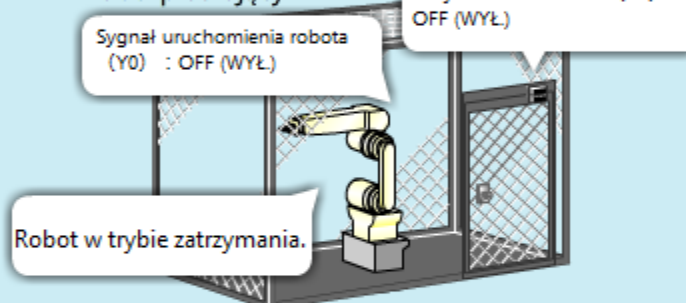
Praca przykładowego systemu

 Kliknij wewnątrz czerwonego okręgu

Panel sterowniczy robota



Robot w ogrodzeniu zabezpieczającym



Gdy **start switch (X6)** (przełącznik uruchomienia (X6)) zostanie ustawiony w położeniu OFF (WYŁ.), **robot start signal (Y0)** (sygnał uruchomienia robota (Y0)) zostanie wyłączony, a robot zatrzyma się. Jednocześnie **operation lamp (Y1)** (lampka działania (Y1)) na panelu sterowniczym wyłączy się, a **stop lamp (Y2)** (lampka zatrzymania (Y2)) zaświeci się.

[Powtórz](#)[Wstecz](#)

2.3

Tworzenie tabeli zbiorczej urządzeń we/wy i numerów urządzeń

Dobrym rozwiązaniem jest utworzenie tabeli zawierającej zestawienie wszystkich urządzeń we/wy i rejestrów zastosowanych w PLC oraz odpowiadających im informacji dla każdego utworzonego programu. Taka tabela zmniejsza ryzyko popełnienia błędów podczas procesów projektowania i programowania oraz umożliwia zwiększenie wydajności programowania. Jeśli tabela zbiorcza już istnieje dla danego systemu, na przykład utworzona przez osobę, która skonfigurowała sprzęt, wystarczy z niej skorzystać.

Tabela poniżej jest tabelą zbiorczą przykładowego systemu zastosowanego w ramach niniejszego kursu

Nazwa urządzenia we/wy	Urządzenie nr	Typ we/wy	Typ urządzenia	Opis
Przełącznik uruchomienia	X6	Wejście	Bitowe	Przełącznik ten uruchamia lub zatrzymuje operację robota.
Czujnik otwarcia drzwi	X7	Wejście	Bitowe	Czujnik ten sprawdza, czy drzwi ogrodzenia zabezpieczającego robota są otwarte. Gdy drzwi zostaną otwarte, czujnik włączy się. Gdy drzwi zostaną zamknięte, czujnik wyłączy się.
Sygnał uruchomienia robota	Y0	Wyjście	Bitowe	Jeśli sygnał włączy się, operacja robota zostanie uruchomiona.
Lampka działania	Y1	Wyjście	Bitowe	Lampka świeci się, gdy robot znajduje się w trybie pracy.
Lampka zatrzymania	Y2	Wyjście	Bitowe	Lampka świeci się, gdy robot znajduje się w trybie zatrzymania.

* W przypadku użycia słowa danych wartość początkowa (górne i dolne wartości graniczne), typ danej (przypisana, rzeczywista itp.) oraz komentarz należy uwzględnić w tabeli.

Informacje te będą użyteczne podczas projektowania i modyfikowania programów.

2.4 Projektowanie programów

Zaprojektuj program przy użyciu języka logiki drabinkowej w oparciu o elementy sterowania i tabelę zbiorczą wejść/wyjść. Poniżej przedstawiono program drabinkowy i tabelę zbiorczą wejść/wyjść zaprojektowane dla przykładowego systemu.

Program drabinkowy

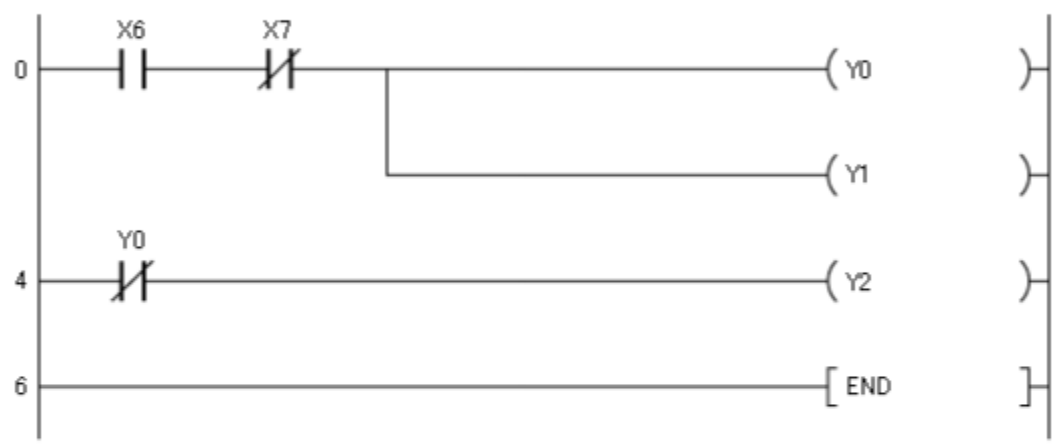
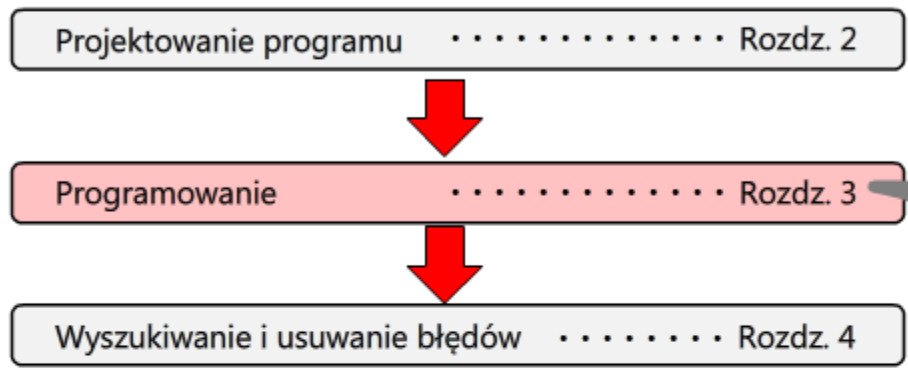


Tabela zbiorcza we/wy

Nazwa urządzenia we/wy	Typ	Numer urządzenia
Przełącznik uruchomienia	Wejście	X6
Czujnik otwarcia drzwi	Wejście	X7
Sygnał uruchomienia robota	Wyjście	Y0
Lampka działania	Wyjście	Y1
Lampka zatrzymania	Wyjście	Y2

Rozdział 3 Programowanie

W Rozdziale 3 dowiesz się, jak zaprogramować zaprojektowany program przy użyciu GX Works2.



Kroki procesu uczenia w Rozdziale 3

- 3.1 Tworzenie projektów
- 3.2 Tworzenie programów
- 3.3 Tworzenie programów łatwych do zrozumienia
- 3.4 Konwertowanie programów do postaci możliwej do wykonania
- 3.5 Zapisywanie projektów

3.1

Tworzenie projektów

Pierwszym krokiem zapisania programu jest utworzenie projektu. Projekt jest zbiorem danych, które GX Works2 wykorzystuje do zarządzania programami. Poniższa tabela zawiera listę podstawowych komponentów składowych projektu.

Typ danej	Opis
Program	Kod źródła i skompilowany kod sekwencji operacji CPU.
Komentarz	Typ dokumentacji wyświetlonej w programie. Patrz punkt 3.3 „Tworzenie programów łatwych do zrozumienia” w celu uzyskania dodatkowych informacji.
Parametr	Zawiera większość lub wszystkie informacje na temat ustawienia i konfiguracji systemu.
Konfiguracja transmisji	Informacje dotyczące przebiegu połączenia wymagane w celu ustanowienia komunikacji pomiędzy oprogramowaniem GX Works2 systemu a modulem CPU.

Program drabinkowy

GX Works2 umożliwia wybranie następujących dwóch typów projektów. Przykładowy program w ramach niniejszego kursu utworzony został w oparciu o typ „**simple project**” (prosty projekt).

Typ projektu	Opis
Prosty projekt	Ten typ projektu jest wstecznie kompatybilny z projektami GX Developer. Proste projekty mogą być konwertowane w późniejszym czasie do strukturalnych projektów, ale działanie odwrotne nie jest możliwe.
Strukturalny projekt	W projektach tych możliwe jest stosowanie dodatkowego języka programowania zwanego drabinką struktury. Ponadto programy mogą zostać podzielone na wiele mniejszych części i często używanych fragmentów kodów, które z łatwością mogą być zmodularyzowane i ponownie użyte przy wykorzystaniu biblioteki użytkownika. Istnieje również możliwość łatwego zmodularyzowania etykiet w celu ich ponownego użycia. Takie rozwiązanie zwiększa wydajność programowania oraz wyszukiwania i usuwania błędów, w szczególności w przypadku dużych projektów.

Etykiety

Etykiety są nazwami utworzonymi przez użytkownika, które stają się aliasami adresów urządzeń. Mogą być one używane globalnie, lokalnie lub w ramach systemu podczas jego wdrażania w połączeniu z MELSOFT Navigator. Proste projekty mogą być tworzone z możliwością lub bez możliwości użycia etykiet. W ramach przykładowego projektu etykiety nie będą używane.

>>

3.1 Tworzenie projektów

Navigation icons: back, forward, search, etc.

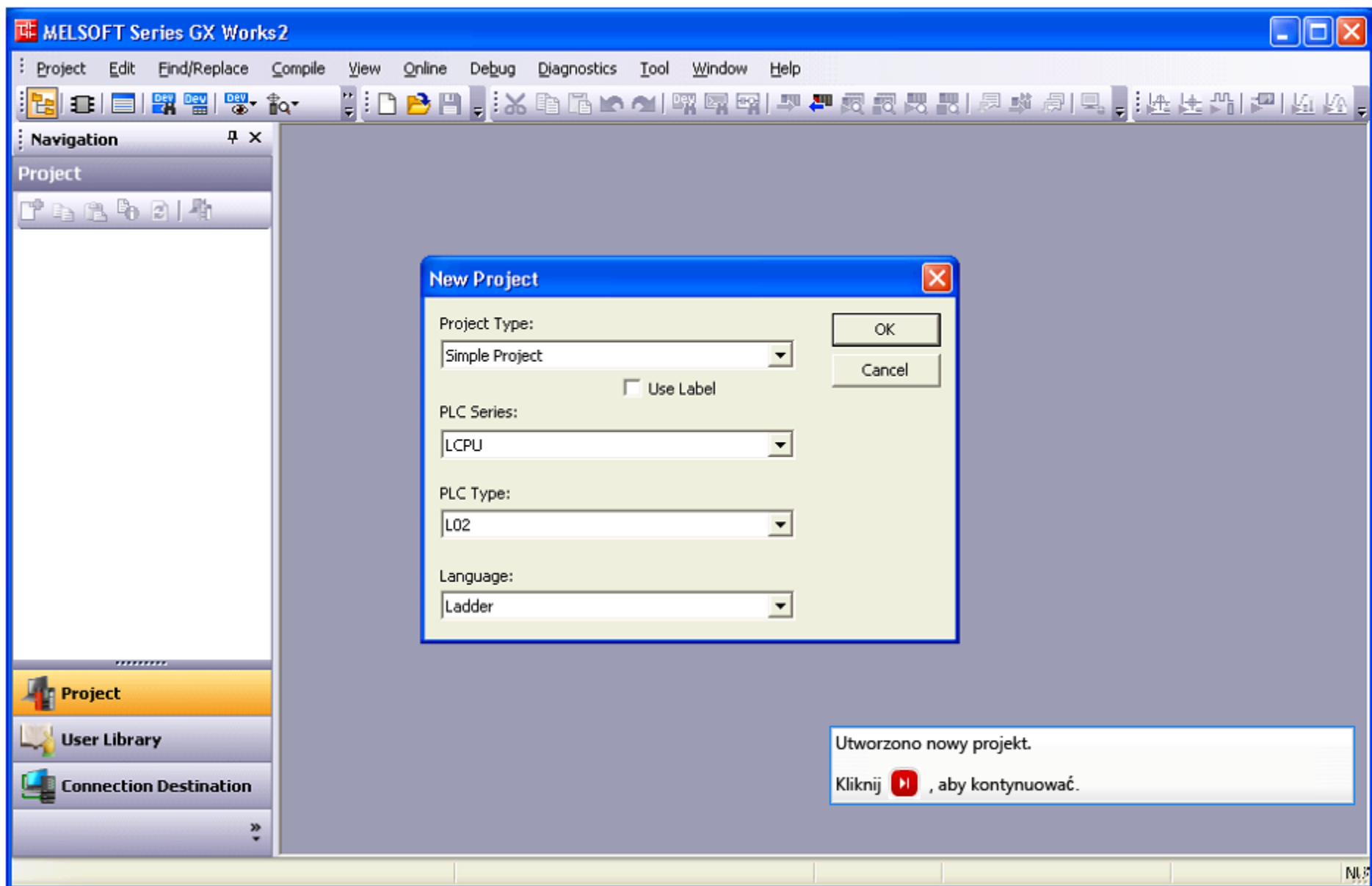
Aby przystąpić do tworzenia przykładowego projektu, należy wprowadzić następujące ustawienia. Przy utworzeniu projektu należy znać nazwę modelu i serię sterownika programowalnego, jak również typ projektu.

Pozycja	Opis
Project type (Typ projektu)	Typ projektu określa zakres dostępnych funkcji podczas zapisywania programów. W ramach tego przykładu należy wybrać „simple project” (prosty projekt).
Use label (Użyj etykiety)	Jeśli podczas zapisywania programu wymagane jest użycie etykiet, należy zaznaczyć tę pozycję. W ramach przykładowego programu etykiety nie będą używane. W związku z powyższym pozostaw to pole odznaczone.
PLC series (Seria PLC)	Seria PLC określa modele dostępne dla danego wyboru na liście rozwijanej typu PLC. W ramach tego przykładu należy wybrać „LCPU”.
PLC type (Typ PLC)	Typ PLC określa, w jaki sposób kompilator skonwertuje programy użytkownika na kody maszyny. Wybierz model PLC, który zostanie zaprogramowany, w tym przypadku „L02”.
Programming language (Język programowania)	Język programowania określa typ programu pierwszego automatycznie utworzonego programu (MAIN). Ponadto programy wykorzystujące różne języki mogą zostać dodane w późniejszym czasie. W ramach tego przykładu należy wybrać „Ladder” (Drabinka).

Przejrzyj następną stronę, na której przedstawiono symulację procesu tworzenia nowego projektu.

3.1

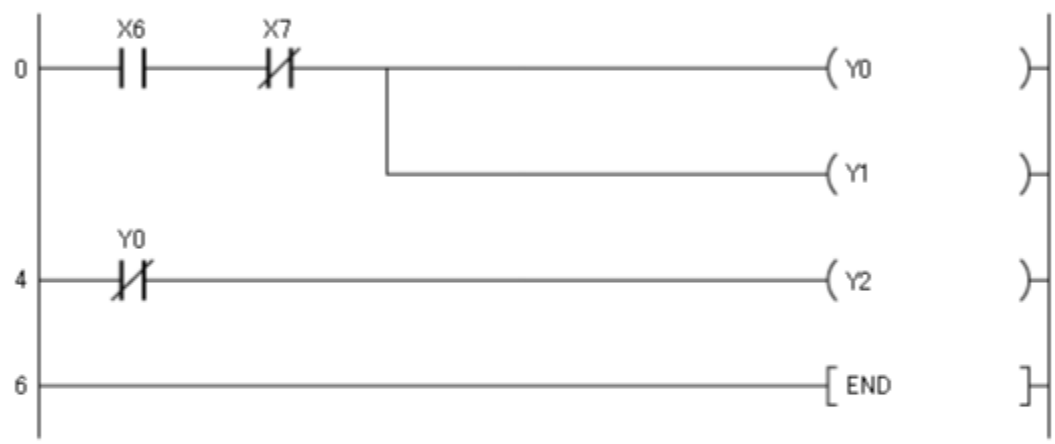
Tworzenie projektów



3.2 Tworzenie programów

Po utworzeniu projektu należy utworzyć program.
Utwórz poniższy program i poznaj podstawowe operacje (wprowadzenie instrukcji, zmiana, usuwanie, kopiowanie i wklejanie oraz wprowadzanie/usuwanie linii prostej).
Poniżej został przedstawiony program zaprojektowany dla przykładowego systemu w Rozdziale 2.

Program dla przykładowego systemu



Na następnej stronie spróbuj utworzyć program przy użyciu symulowanego okna.

3.2

Tworzenie programów

MELSOFT Series GX Works2 (Unset Project) - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
 - Program
 - MAIN
 - Local Device Comment
- Device Memory
- Device Initial Value

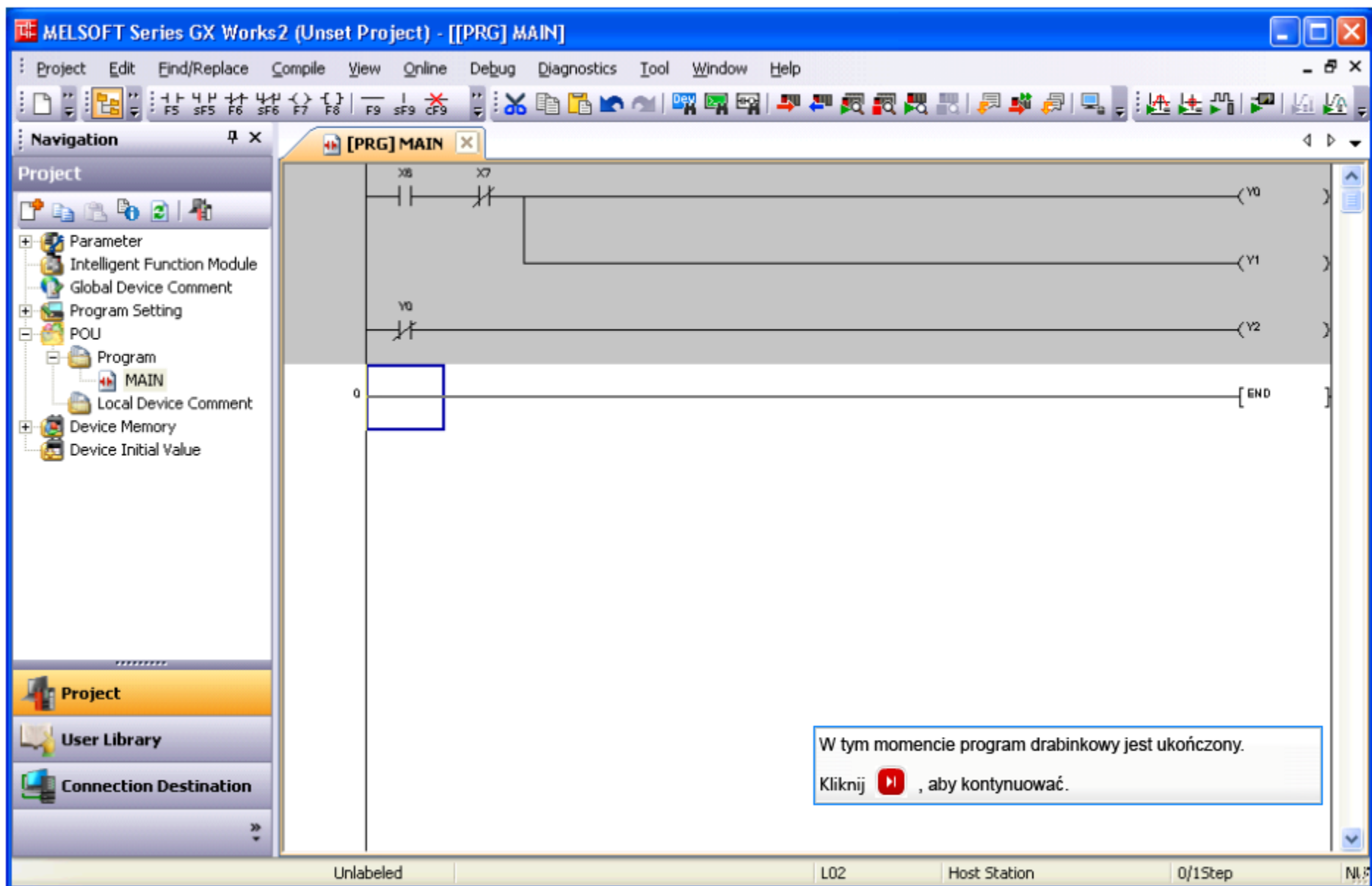
Unlabeled

L02

Host Station

0/15Step

W tym momencie program drabinkowy jest ukończony.
Kliknij , aby kontynuować.



3.3

Tworzenie programów łatwych do zrozumienia

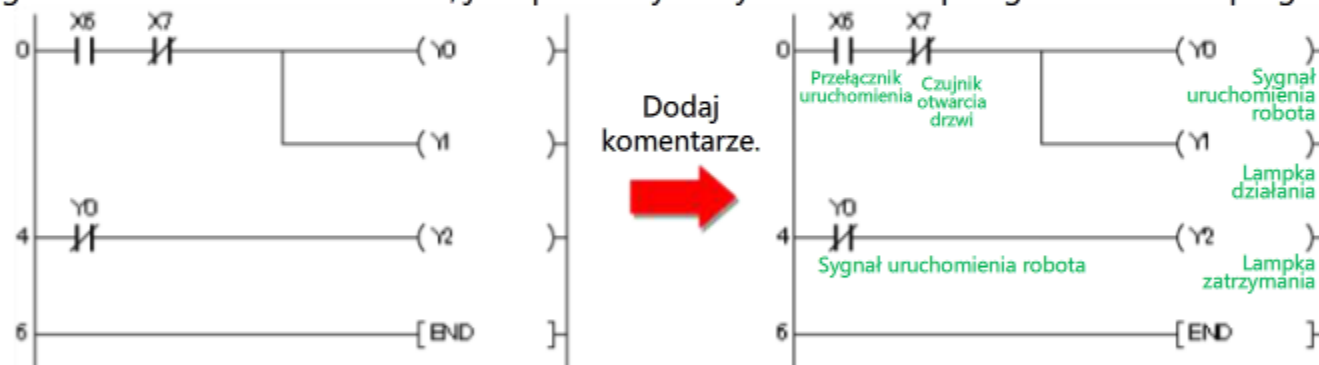
Na tym etapie wizualna prezentacja programu zawiera wyłącznie urządzenia, instrukcje, linie i numery kroków. Przyglądając się złożonemu programowi trudno określić czym steruje program.

- Trudno znaleźć błędy programowania, takie jak niewłaściwe numery urządzenia lub instrukcje.
- Podsumowując, trudno przeprowadzić analizę operacyjną, wyszukiwanie i usuwanie błędów oraz rozbudowę programu.
- Jeśli programista oryginalnego programu nie może dłużej utrzymywać programu, zadanie dotyczące uczenia się działania programu może dla większości osób być zniechęcające, a nawet niemożliwe.

Środki zaradcze

Dołącz **dokumentację** do programu w celu umożliwienia użytkownikowi szybkiego zapoznania się z zasadą działania programu.

Do dobrych praktyk zalicza się dodawanie przez wszystkich programistów szczegółowych komentarzy dotyczących ich programów w celu umożliwienia im, jak i pozostałym użytkownikom lepszego zrozumienia programu.



W oprogramowaniu GX Works2 istnieje możliwość wprowadzania trzech różnych typów komentarzy. W celu uzyskania szczegółowych informacji patrz podręcznik Proste projekty GX Works2.

Typ komentarza	Zakres komentarza
Device comment (Komentarz do urządzenia)	Wprowadź tekst o długości maksymalnie 32 znaków, który będzie wyświetlany pod wybranym urządzeniem (we/wy lub inny adres pamięci).
Statement (Wyjaśnienie)	Wprowadź wyjaśnienie o długości maksymalnie 64 znaków, które zostanie dodane na górze wybranego bloku drabinki (powyżej numeru kroku). Każdy blok drabinki może mieć przypisanych wiele wyjaśnień.
Note (Uwaga)	Wprowadź tekst o długości maksymalnie 32 znaków, który będzie wyświetlany nad wybraną cewką lub instrukcją.

Na następnej stronie została przedstawiona symulacja procesu dodawania komentarzy do urządzenia do przykładowego programu.

3.3

Tworzenie programów łatwych do zrozumienia

MELSOFT Series GX Works2 (Unset Project) - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
 - Program
 - MAIN
 - Local Device Comment
- Device Memory
- Device Initial Value

[[PRG] MAIN]

0 X6 Start switch X7 Door open sensor Y0 Robot start signal Y1 Operation lamp Y2 Stop lamp

4 Y0 Robot start signal

6 [END]

Unlabeled L02 Host Station 5/75Step

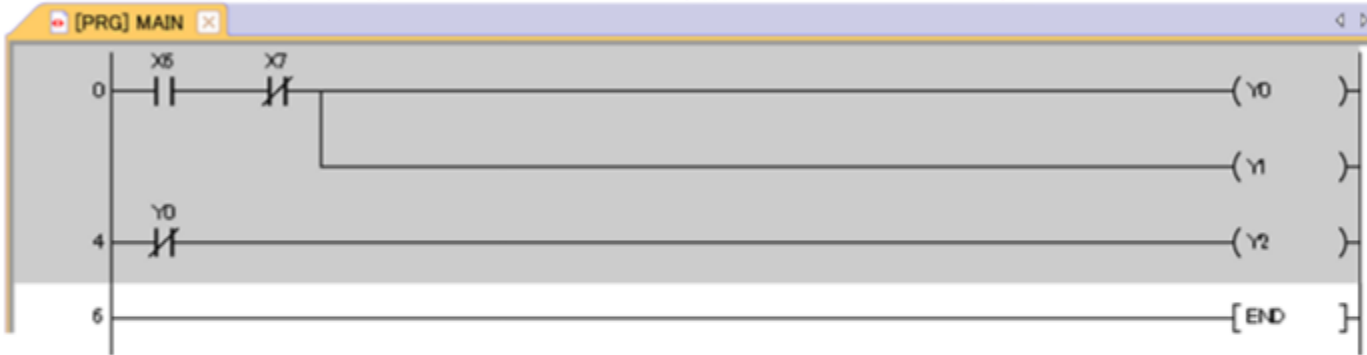
Wprowadzanie komentarza dla urządzenia zostało ukończone.
Kliknij , aby kontynuować.

3.4 Konwertowanie programów do postaci możliwej do wykonania

Po zakończeniu tworzenia programu konieczne jest jego skonwertowanie do formatu, który może zostać wykonany przez moduł CPU.

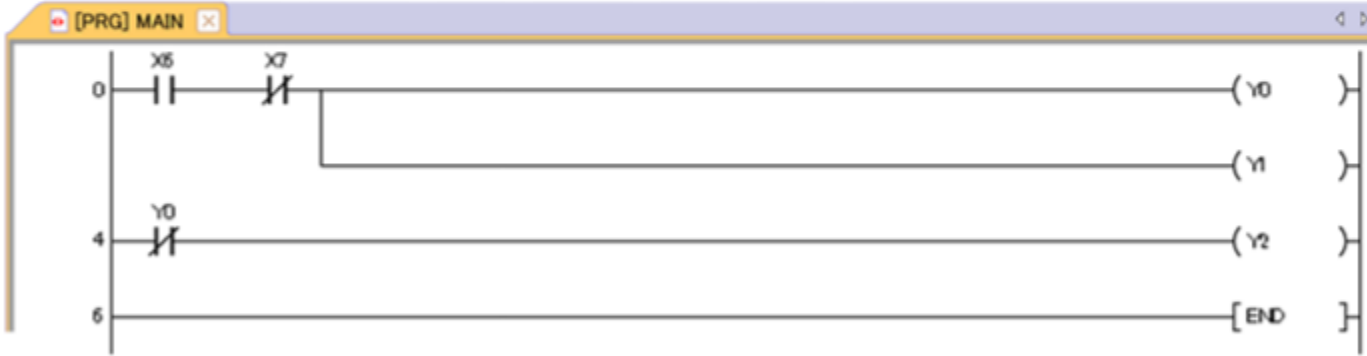
Nieskonwertowane programy nie mogą być wykonywane lub zapisywane.

Tło nieskonwertowanych programów ma kolor szary, jak pokazano poniżej.



Convert (Konwertuj)

Po konwersji kolor tła zmienia się na biały, jak pokazano poniżej.



Na następnej stronie spróbuj skonwertować program przy użyciu symulowanego okna.

3.4 Konwertowanie programów do postaci możliwej do wykonania

MELSOFT Series GX Works2 (Unset Project) - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
 - Program
 - MAIN
 - Local Device Comment
- Device Memory
- Device Initial Value

[[PRG] MAIN

0 X6 X7 Y0 Y1 Y2

Start switch Door open sensor Robot start signal Operation lamp Stop lamp

Po skonwertowaniu programu kolor tła ulega zmianie z szarego na biały.

4 Y0 Robot start signal

6 [END]

Program został skonwertowany.
Kliknij , aby kontynuować.

Unlabeled L02 Host Station 5/75Step

>>

3.5

Zapisywanie projektów

Navigation icons: back, forward, search, etc.

Po zakończeniu konwertowania programu zapisz projekt obejmujący programy. Jeśli oprogramowanie GX Works2 zostanie zamknięte bez zapisywania projektu, powiązane programy zostaną odrzucone, dlatego też należy bezwzględnie regularnie zapisywać swój projekt.

Podczas zapisywania nowego projektu określ następujące typy informacji projektu. (Ta czynność nie jest wymagana w przypadku nadpisywania-zapisywania.)

Należy dołączyć informacje ułatwiające pozostałym użytkownikom zapoznanie się z zawartością elementów sterowania programu, nazwą systemu itp.

Pozycja	Wymagane	Opis
Save destination path(Zapisz ścieżkę miejsca przeznaczenia)	✓	Określ katalog, w którym zostanie zapisana przestrzeń robocza.
Workspace/project list (Lista przestrzeni roboczej/projektu)		Jeśli istnieje już jedna lub więcej przestrzeni roboczych w katalogu określone jako „Save destination path” (Ścieżka miejsca przeznaczenia), istniejące przestrzenie robocze zostaną wymienione na liście.
Workspace name (Nazwa przestrzeni roboczej)	✓	Nadaj nazwę przestrzeni roboczej. Długość nazwy nie może przekraczać 128 znaków.
Project name (Nazwa projektu)	✓	Nadaj nazwę projektowi. Długość nazwy nie może przekraczać 128 znaków.
Title (Tytuł)		Nadaj tytuł projektowi. Długość tytułu nie może przekraczać 128 znaków. Parametr ten jest użyteczny w przypadku konieczności przypisania długiej nazwy, która przekracza dozwoloną liczbę znaków pola „Project name” (Nazwa projektu).

Workspace (przestrzeń robocza) jest katalogiem umożliwiającym zarządzanie wieloma projektami.

Przykład użycia przestrzeni roboczej został przedstawiony poniżej. (Projekty są zarządzane w ramach każdego typu pojazdu na samochodowej linii produkcyjnej.)

Nazwa przestrzeni roboczej	Nazwa projektu	Tytuł
Samochodowa linia produkcyjna	Linia produkcyjna typu A	Normalny program pracy do sterowania linią produkcyjną typu A
	Linia produkcyjna typu B	Normalny program pracy do sterowania linią produkcyjną typu B
	Linia produkcyjna typu C	Normalny program pracy do sterowania linią produkcyjną typu C

- Uwaga:**
- Jeśli projekt zawierający nieskonwertowany program nie zostanie zapisany, odrzucony zostanie wyłącznie nieskonwertowany program. Przed zapisaniem projektu przeprowadź konwersję programu zgodnie z opisem w punkcie 3.4.
 - Wybierz ścieżkę zapisania miejsca przeznaczenia, nazwę przestrzeni roboczej oraz nazwę projektu, tak aby maksymalna długość nie przekraczała 150 znaków.

Na następnej stronie spróbuj zapisać projekt przy użyciu symulowanego okna.

3.5

Zapisywanie projektów

MELSOFT Series GX Works2 C:\SequenceProgram\Learning\Robot_Control - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
 - Program
 - MAIN
 - Local Device Comment
- Device Memory
- Device Initial Value

[[PRG] MAIN

0 X6 Start Switch X7 Door open sensor Y0 Robot start signal Y1 Operation lamp Y2 Stop lamp

4 Y0 Robot start signal

6 END

Unlabeled L02 Host Station 6/75Step

Projekt został zapisany.
Kliknij , aby kontynuować.

Rozdział 4 Wyszukiwanie i usuwanie błędów

W Rozdziale 4 zawarte zostały informacje o sposobie zapisywania programów sekwencyjnych w module CPU oraz ich przeszukiwania w celu wyszukania i usunięcia błędów.

Projektowanie programu Rozdz. 2



Programowanie Rozdz. 3



Wyszukiwanie i usuwanie błędów Rozdz. 4

Kroki procesu uczenia w Rozdziale 4

4.1 Wyszukiwanie i usuwanie błędów

4.1.1 Wyszukiwanie i usuwanie błędów
w programie bez użycia modułu CPU

4.1.2 Zmienianie statusu urządzenia we/wy

4.1.3 Monitorowanie statusu urządzenia

4.2 Zapisywanie programów do modułu CPU

4.3 Włączanie zapisanych programów

4.4 Uruchamianie programów

4.5 Wykrywanie i usuwanie błędów programów

4.6 Sprawdzanie działania układu PLC

4.7 Obsługa układu PLC

>>

4.1

Jak działa funkcja „wykrywanie i usuwanie błędów”

◀

▶

TOC

Po zapisaniu programu lub części programu należy przetestować kod w celu sprawdzenia, czy działa zgodnie z oczekiwaniami. Usterki oprogramowania (gdy zapisany kod nie przeprowadza działań zgodnie z jego przeznaczeniem) nazywane są „**błędami**”, a proces wyszukiwania przyczyny niezamierzonych działań i ich naprawa nazywany jest „**wyszukiwaniem i usuwaniem błędów**”.

Testowanie oraz wyszukiwanie i usuwanie błędów są najważniejszymi krokami w procesie tworzenia programów. W szczególności w przypadku sterowników programowalnych, ponieważ wystąpienie błędów może prowadzić do zatrzymania systemu, uszkodzenia sprzętu lub innych wypadków.

Poniższa tabela zawiera listę kilku funkcji oprogramowania GX Works2, które mogą pomóc podczas przeprowadzania procesu wyszukiwania i usuwania błędów.

Nazwa funkcji	Opis
Simulator (Symulator)	Funkcja ta umożliwia przeprowadzenie symulacji wykonania programu nawet bez modułu CPU. Funkcja ta może być stosowana do wyszukiwania i usuwania błędów w środowisku, w którym moduł CPU jest niedostępny.
Monitor (Monitor)	Funkcja ta umożliwia monitorowanie statusu wykonania i statusu każdego urządzenia podczas wykonywania programu przez moduł CPU. Monitor oferuje wiele funkcji w zależności od zastosowania, takie jak monitorowanie na drabince, monitorowanie wyłącznie zarejestrowanych urządzeń oraz monitorowanie wszystkich urządzeń w serii.
Change current value (Zmień bieżącą wartość)	Funkcja ta może wymusić zmianę statusu urządzenia (bit (bit); ON ↔ OFF (WŁ. ↔ WYŁ.), word: current value (słowo: bieżąca wartość)) podczas wykonywania programu przez moduł CPU. Funkcja ta jest użyteczna przy zmianie bieżącej wartości pamięci podzielonej na słowa danych lub statusu przekaźnika wewnętrznego.
Forced input output registration/cancellation (Rejestracja/anulowanie wymuszonego sygnału wejściowego/wyjściowego)	Funkcja ta może wymusić zmianę statusu (ON ↔ OFF (WŁ. ↔ WYŁ.)) zarejestrowanego urządzenia we/wy podczas wykonywania programu przez moduł CPU. W przypadku wyszukiwania i usuwania błędów lub weryfikacji działania z oddzielnym modułem CPU funkcja ta może być stosowana jako zamiennik przełącznika.

Funkcje te zostały wyjaśnione bardziej szczegółowo w odniesieniu do procesu wyszukiwania i usuwania błędów w dalszej części niniejszego rozdziału.

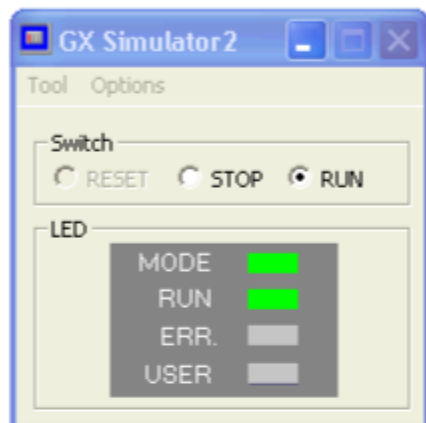
Uwagi dotyczące wyszukiwania i usuwania błędów

Nie przeprowadzaj zadań związanych z wyszukiwaniem i usuwaniem błędów, gdy sterownik programowalny jest podłączony fizycznie do urządzeń we/wy. Błędy w programie, wymuszone urządzenia we/wy lub zmiany wartości słowa mogą prowadzić do uszkodzenia sprzętu zewnętrznego lub gorszych skutków. Jeśli odłączony system PLC jest niedostępny, użyj funkcji symulator (symulator).

4.1.1

Wykrywanie i usuwanie błędów programu bez używania jednostki CPU

Jeśli moduł CPU jest niedostępny dla wyszukiwania i usuwania błędów, użyj **funkcji simulator (symulator)**. Program może pracować na wirtualnej jednostce centralnej zapewnionej przez oprogramowanie bez użycia rzeczywistego modułu CPU.



 On (Wł.)

 Off (Wył.)

Pozycja	Stan	Opis
Switch (Przełącznik)	RUN (PRACA)	Uruchamia wirtualną jednostkę centralną.
	STOP (ZATRZYMANIE)	Zatrzymuje wirtualną jednostkę centralną.
	RESET (RESETOWANIE)	Resetuje wirtualną jednostkę centralną. (opcja aktywna wyłącznie w stanie STOP (ZATRZYMANIE))
LED (LED)	MODE (TRYB)	Wskazuje status TRYBU wirtualnej jednostki centralnej.
	RUN (PRACA)	Wskazuje status pracy wirtualnej jednostki centralnej. •On (Wł.): stan RUN (PRACY) •Off (Wył.): stan STOP (ZATRZYMANIE)
	ERR (BŁĄD)	Wskazuje status błędu wirtualnej jednostki centralnej. W przypadku wystąpienia błędu dioda LED świeci się lub miga.
	USER (UŻYTKOWNIK)	Wskazuje, czy wystąpił błąd użytkownika w wirtualnej jednostce centralnej. Świeci się lub miga w przypadku wystąpienia błędu.

Uwagi dotyczące korzystania z funkcji simulator (symulator)

- Wyszukiwanie i usuwanie błędów za pomocą funkcji simulator (symulator) nie gwarantuje, że program sekwencyjny będzie działał prawidłowo po przeprowadzeniu procesu wyszukiwania i usuwania błędów.
- Funkcja simulator (symulator) wykonuje operację wprowadzania/wysyłania danych za pomocą modułów wejść/wyjść przy użyciu pamięci symulacji.
Funkcja ta nie obsługuje niektórych instrukcji, funkcji i pamięci urządzenia. W związku z tym wyniki operacji za pomocą funkcji simulator (symulator) mogą różnić się od tych z rzeczywistym modułem CPU.

Na następnej stronie spróbuj użyć funkcji simulator (symulator) przy użyciu symulowanego okna.

>>

4.1.1

Wykrywanie i usuwanie błędów programu bez używania jednostki CPU

◀

▶

TOC

MELSOFT Series GX Works2 C:\SequenceProgram\LearningRobot_Control - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation

Project

Parameter

Intelligent Function Module

Global Device Comment

Program Setting

POU

Program

MAIN

Local Device Comment

Device Memory

Device Initial Value

Project

User Library

Connection Destination

[PRG] MAIN

0

X6 Start Switch

X7 Door open sensor

Y0 Robot start signal

Y1 Operation lamp

4

Y0 Robot start signal

Y2 Stop lamp

6

END

Unlabeled

L02

Host Station

6/75Step

NL

Właśnie poznałeś/-aś sposób korzystania z funkcji symulacja.

Kliknij  , aby kontynuować.

4.1.2

Zmienianie statusu urządzenia wejść/wyjść

Podczas wyszukiwania i usuwania błędów w programie sekwencyjnym z modułem CPU, do którego nie podłączono żadnego urządzenia wejść/wyjść lub korzystania z simulator function (funkcji simulator) użyj funkcji **Forced Input Output Registration/Cancellation** (Rejestracja/anulowanie wymuszonego sygnału wejściowego/wyjściowego), aby zmienić stan ON/OFF (Wł./WYł.) urządzenia wejść/wyjść.

Zmianę statusu z ON (Wł.) lub OFF (WYł.) zarejestrowanych urządzeń we/wy można wymusić za pomocą oprogramowania.

(Seria MELSEC-Q i MELSEC-L): Z poziomu ekranu „Forced Input Output Registration/Cancellation” (Rejestracja/anulowanie wymuszonego sygnału wejściowego/wyjściowego)

(Seria ELSEC-F): Z poziomu ekranu „Modify Value” (Zmodyfikuj wartość)

No.	Device	ON/OFF	No.	Device	ON/OFF
1	X6	ON	17		
2	X7	OFF	18		
3			19		
4			20		
5			21		
6			22		
7			23		
8			24		
9			25		
10			26		
11			27		
12			28		
13			29		
14			30		
15			31		
16			32		

Ekran Forced Input Output Registration/Cancellation
(Rejestracja/anulowanie wymuszenia sygnału wejściowego/wyjściowego)
(seria MELSEC-Q i MELSEC-L)

Device/Label	Data Type	Setting Value
X6	Bit	ON

Ekran Modify Value (Zmodyfikuj wartość)
(seria MELSEC-F)

Aby zmienić stany pozostałych urządzeń

Aby zmienić bieżącą wartość pamięci podzielonej na słowa danych lub stan ON/OFF (Wł./WYł.) przekaźnika wewnętrznego, użyj funkcji **zmiana bieżącej wartości**.

W celu uzyskania szczegółowych informacji patrz podręcznik.

4.1.3

Monitorowanie statusu urządzenia

Po uruchomieniu symulacji automatycznie uruchamiane jest monitorowanie. Aby wprowadzić tryb monitorowania w przypadku połączenia do bieżącego CPU sterownika programowalnego, wystarczy kliknąć opcję Online (Online), Monitor (Monitor), a następnie Start Monitoring (Uruchom monitorowanie). Lub użyj skrótu klawiaturowego F3.

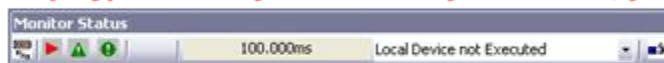
W trybie monitorowania wartości i statusy wszystkich urządzeń wykorzystanych w programie są widoczne powyżej kodu programu. Umożliwia to użytkownikowi przyjrzenie się zmianie wartości, włączając wpływ funkcji „Forced input output registration/cancellation” (rejestracja/anulowanie wymuszenia sygnału wejściowego/wyjściowego).

Ponadto zostaje wyświetlony pasek **Monitor Status (Monitoruj status)** zawierający podstawowe informacje umożliwiające określenie statusu CPU lub wirtualnej jednostki centralnej. Patrz tabela poniżej w celu zapoznania się z informacjami udostępnionymi przez pasek **Monitor Status (Monitoruj status)**.

W przypadku połączenia z modułem CPU



W przypadku korzystania z funkcji simulator (symulator)



Stan	Ikona/wskazanie	Opis
Status połączenia	W przypadku połączenia z modułem CPU	Wyświetla status połączenia z modułem CPU lub funkcji simulator (symulator).
	W przypadku korzystania z funkcji simulator (symulator)	
Status RUN/STOP (PRACA/ZATRZYMANIE)	RUN (PRACA)	Wyświetla status pracy CPU (RUN (PRACA) LUB STOP (ZATRZYMANIE)).
	STOP (ZATRZYMANIE)	
ERR. status (Status BŁĘDU)	ERR. off (BŁĄD wył.)	Wyświetla status błędu modułu CPU.
	ERR. on (BŁĄD wł.)	
	ERR blinking (Miganie kontrolki BŁĘDU)	
USER status (Status UŻYTKOWNIKA)	USER off (UŻYTKOWNIK wył.)	Wyświetla status błędu użytkownika modułu CPU.
	USER on (UŻYTKOWNIK wł.)	
	USER blinking (Miganie kontrolki UŻYTKOWNIKA)	
Scan time (Czas skanowania)	0.000ms	Wyświetla maksymalny czas skanowania monitorowanego modułu CPU.
Unsupported instruction presence/absence status (status obecności/braku obecności nieobsługiwanej instrukcji)	Obecność nieobsługiwanej instrukcji.	Wyświetla, czy jest obecna nieobsługiwana instrukcja podczas wykonywania funkcji simulator (symulator). Kliknięcie ikony otwiera okno Unsupported Instruction/Device (Nieobsługiwana instrukcja/urządzenie).
	Brak obecności nieobsługiwanej instrukcji.	

4.1.3 Monitorowanie statusu urządzenia

W trybie monitorowania bieżący status wszystkich urządzeń w programie jest widoczny.

Wyświetlanie statusu urządzenia bitowego (ON/OFF (WŁ./WYŁ.))

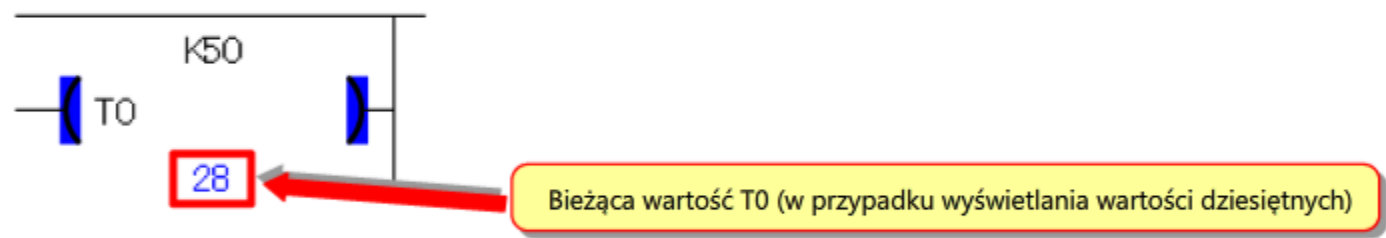
Status ON/OFF (WŁ./WYŁ.) jest wyświetlany podczas monitorowania, jak pokazano poniżej.



* Ten sposób wyświetlania dotyczy wyłącznie SET, RST, PLS, PLF, SFT, SFTP, MC oraz instrukcji porównania typów styków. W przypadku instrukcji RST wyświetlany jest wyłącznie status ON/OFF (WŁ./WYŁ.).

Wyświetlana bieżąca wartość pamięci podzielonej na słowa danych (wyświetlanie wartości dziesiętnej/szesnastkowej)

Bieżąca wartość podczas monitorowania jest wyświetlana, jak pokazano poniżej.



Monitorowanie tylko określonych urządzeń

Podczas monitorowania bardzo rozbudowanego lub złożonego programu ważne może okazać się monitorowanie wyłącznie określonych urządzeń w ramach danego problemu. Aby tego dokonać, oprogramowanie GX Works2 zawiera okna obserwacji, umożliwiające użytkownikowi na łatwe dodanie urządzeń, których dotyczy dana kwestia. Umożliwiają one również sprawdzenie bieżącego statusu i modyfikowanie wartości podczas monitorowania W celu uzyskania szczegółowych informacji patrz podręcznik obsługi GX Works2 (wspólny).

Watch 1					
Device/Label	Current Value	Data Type	Class	Device	Comment
X7	--	Bit		X7	Door open sensor
Y0	--	Bit		Y0	Robot start signal
Y1	--	Bit		Y1	Operation lamp
Y0	--	Bit		Y0	Robot start signal
Y2	--	Bit		Y2	Stop lamp
Y0	--	Bit		Y0	Robot start signal

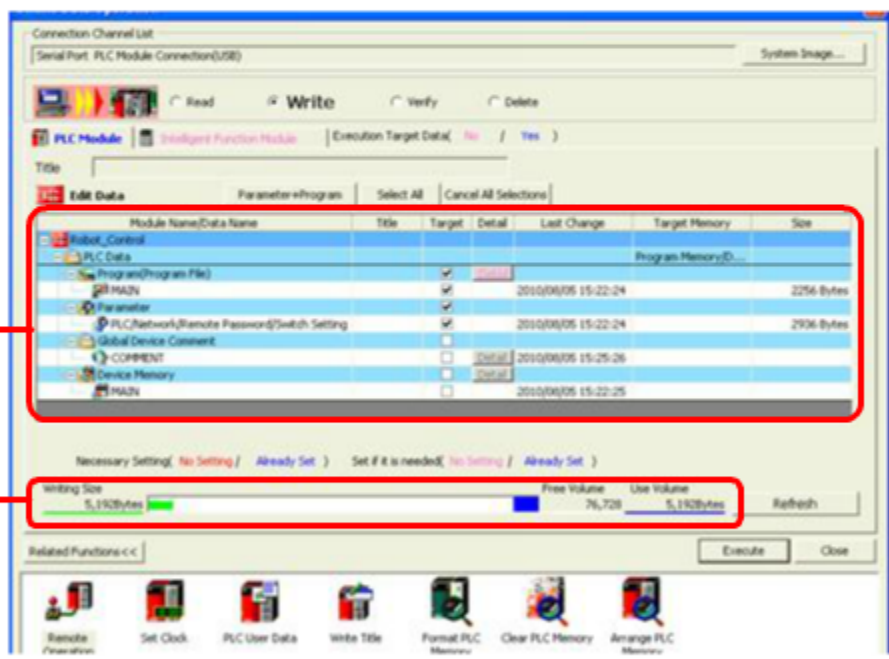
4.2 Zapisywanie programów do modułu CPU

Przed przeprowadzeniem procesu wyszukiwania i usuwania błędów za pomocą rzeczywistego modułu CPU, ustawić CPU w **trybie STOP (ZATRZYMANIE)**, upewnij się, że połączenie z CPU zostało nawiązane, a następnie zapisz programy i parametry w pamięci programu.

Widoczne na zrzucie z ekranu poniżej główne funkcje okna **Write to PLC (Zapisz do PLC)** umożliwiają użytkownikowi wybranie wymaganych plików od zapisania, wybranie ich lokalizacji i potwierdzenie wielkości pamięci CPU. Trzy przyciski powyżej listy plików umożliwiają użytkownikowi szybkie wybranie wymaganych plików do zapisania. Najczęściej stosowaną opcją w poniższej symulacji jest „**Parameter+Program**” (Parametr+Program).

Lista plików

Wielkość pamięci



Na następnej stronie spróbuj zapisać daną operację w module CPU przy użyciu symulowanego okna.

4.2

Zapisywanie programów do modułu CPU

MELSOFT Series GX Works2 C:\SequenceProgram\Learning\Robot_Control - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
 - Program
 - MAIN
 - Local Device Comment
- Device Memory
- Device Initial Value

[[PRG] MAIN

0 X6 Start Switch X7 Door open sensor

Y0 Robot start signal

Y1 Operation lamp

4 Y0 Robot start signal

Y2 Stop lamp

6 END

Unlabeled L02 Host Station 6/75Step

Program został zapisany w module PLC.
Kliknij  , aby kontynuować.

4.3

Włączanie zapisanych programów

(Seria MELSEC-F): Poniższa operacja nie jest wymagana.

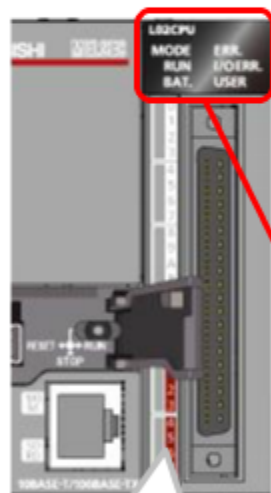
(Seria MELSEC-Q i MELSEC-L): Poniższa operacja jest wymagana.

Po zapisaniu programu w module CPU **zresetuj** moduł CPU.

Nie ma możliwość włączenia zapisanych programów dopóki moduł CPU nie zostanie zresetowany.

* Operacja ta nie jest wymagana, jeśli funkcja symulatora jest stosowana do przeprowadzania operacji wyszukiwania i usuwania błędów.

Zresetuj moduł CPU zgodnie z poniższą procedurą:



- (1) Naciśnij i przytrzymaj przełącznik RESET/STOP/RUN (RESETOWANIE/ZATRZYMANIE/PRACA) znajdujący się na przednim panelu modułu CPU w celu RESET (ZRESETOWANIA) pozycji (przez 1 sekundę lub dłużej).
[Resetowanie w toku]

L02CPU	
MODE ■ ERR.	MODE (TRYB): wł. – kolor zielony
RUN I/OERR.	RUN(PRACA): wył.
BAT. USER	ERR(BŁĄD) : miganie

Nacisnąć i przytrzymać przez 1 sekundę lub dłużej.



- (2) Zwolnij przełącznik po zaświeceniu się diody LED MODE (TRYB) i miganiu ERR. (BŁĄD). Obydwie diody LED wyłączają się.
[Resetowanie ukończone]

L02CPU	
MODE ■ ERR.	MODE (TRYB): wł. – kolor zielony
RUN I/OERR.	RUN(PRACA): wył.
BAT. USER	ERR(BŁĄD) : wył.

Przełącznik
RESET/STOP/RUN
(RESETOWANIE/ZATRZYMANIE/PRACA)

- (3) Przełącznik powraca do ustawienia STOP (ZATRZYMANIE) w celu ukończenia operacji resetowania.

4.4

Uruchamianie programów

Seria MELSEC-Q i MELSEC-L

Po ukończeniu operacji resetowania uruchom program.

Ustaw moduł CPU na **status RUN (PRACA)** zgodnie z poniższą procedurą, aby uruchomić program.

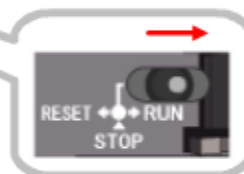
* Operacja ta nie jest wymagana, jeśli simulator function (funkcja simulator) jest stosowana do przeprowadzania operacji wyszukiwania i usuwania błędów.

- (1) Ustaw przełącznik RESET/STOP/RUN (RESETOWANIE/ZATRZYMANIE/PRACA) znajdujący się na przednim panelu modułu CPU w położeniu RUN (PRACA).
Dioda LED wyświetla status STOP (ZATRZYMANIE)



L02CPU	
MODE	ERR.
RUN	I/OERR.
BAT.	USER

MODE (TRYB):
wł. – kolor zielony
RUN(PRACA): wyt.



- (2) Jeśli dioda LED RUN (PRACA) świeci się na zielono, program pracuje w trybie normalnym.

Dioda LED wyświetla status RUN

L02CPU	
MODE	ERR.
RUN	I/OERR.
BAT.	USER

MODE (TRYB):
wł. – kolor zielony
RUN(PRACA):
wł. – kolor zielony

Seria MELSEC-F

Po zapisaniu programu w jednostce głównej przełącz jednostkę główną do stanu RUN (PRACA) zgodnie z poniższą procedurą, aby uruchomić program. (Operacja resetowania nie jest wymagana.)

- (1) Ustaw przełącznik RUN/STOP (PRACA/ZATRZYMANIE) znajdujący się na przednim panelu jednostki głównej w położeniu RUN (PRACA).



POWER
RUN
BATT
ERROR

Dioda LED wyświetla status STOP (ZATRZYMANIE)

- (2) Jeśli dioda LED RUN (PRACA) świeci się, program pracuje w trybie normalnym.

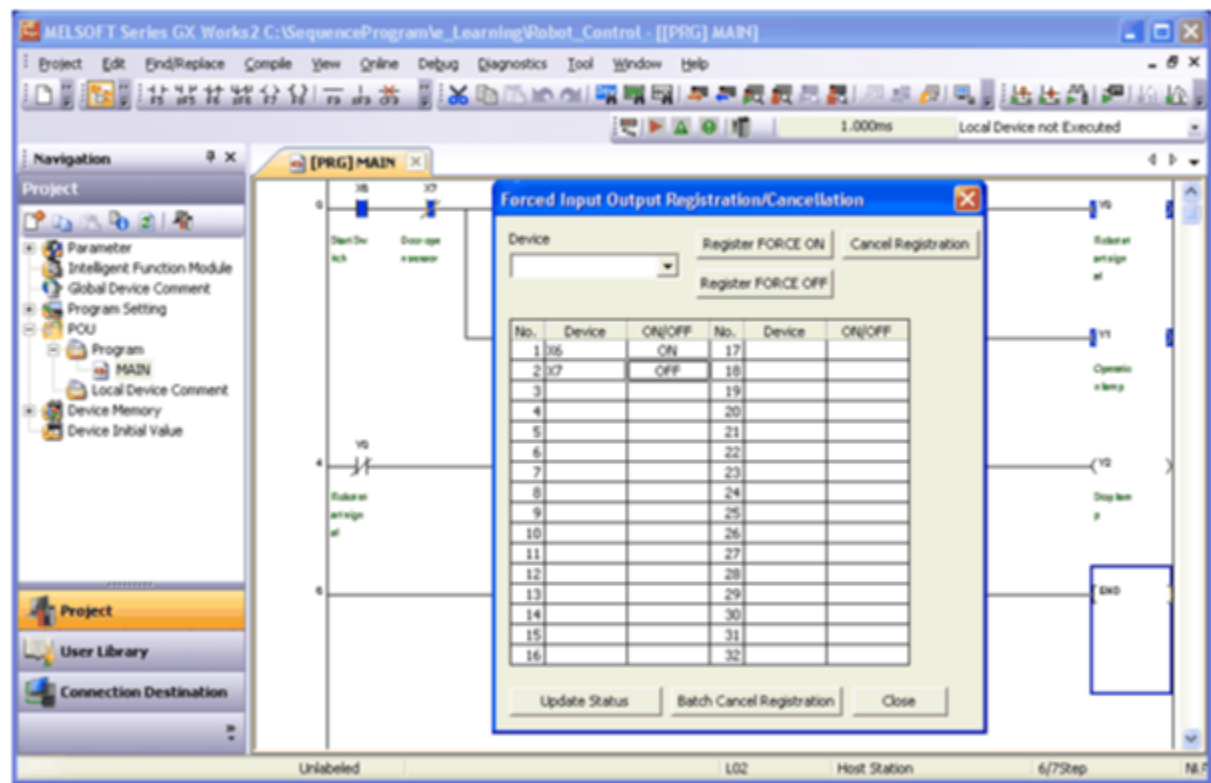
POWER
RUN
BATT
ERROR

Dioda LED wyświetla status RUN (PRACA)

4.5 Wykrywanie i usuwanie błędów programów

Po uruchomieniu modułu CPU użyj funkcji forced input output registration/cancellation (rejestracja/anulowanie wymuszonego sygnału wejściowego/wyjściowego), aby zmienić status każdego urządzenia i monitorować wyniki (sygnał wyjściowy) na drabince.

(Przykładowy ekran serii MELSEC-Q i MELSEC-L)



Na następnej stronie spróbuj wyszukać i usunąć usterki programu przy użyciu symulowanego okna.

4.5 Wykrywanie i usuwanie błędów programów

MELSOFT Series GX Works2 C:\SequenceProgram\Learning\Robot_Control - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
 - Program
 - MAIN
- Local Device Comment
- Device Memory
- Device Initial Value

Project

User Library

Connection Destination

[PRG] MAIN

0

X6 Start Switch

X7 Door open sensor

Y0 Robot start signal

Y1 Operation lamp

4

Y0 Robot start signal

Y2 Stop lamp

6

END

Unlabeled

L02

Host Station

6/75Step

NL

Wyszukiwanie i usuwanie błędów programu zostało zakończone

Kliknij  , aby kontynuować.

4.6

Sprawdzanie działania układu PLC

Po zakończeniu operacji wyszukiwania i usuwania błędów zapisz program w rzeczywistym systemie PLC, aby sprawdzić ostatecznie operację.

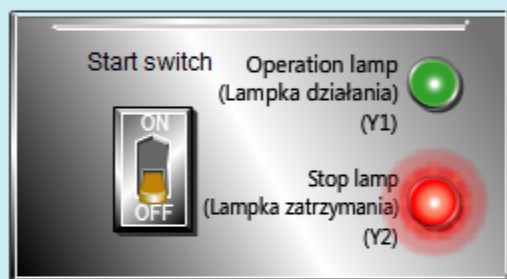
Obsłuż rzeczywiste urządzenie we/wy, aby potwierdzić, czy działa ono prawidłowo zgodnie z projektem.

Nawet gdy urządzenie we/wy działa, status każdego urządzenia można sprawdzić za pomocą funkcji monitor (monitor) oprogramowania GX Works2.

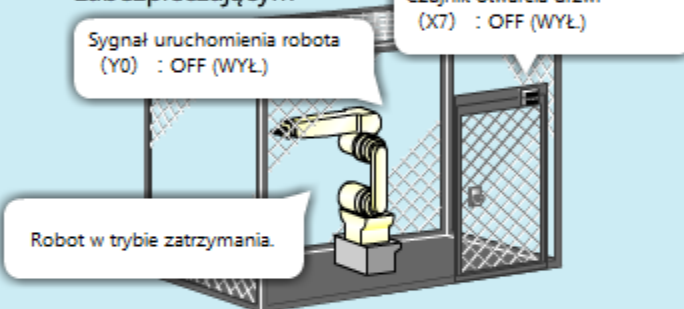
Praca przykładowego systemu

☐ Kliknij wewnątrz czerwonego okręgu

Panel sterowniczy robota



Robot w ogrodzeniu zabezpieczającym

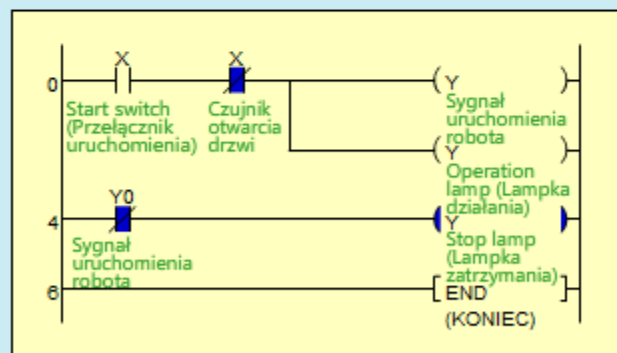


Gdy **start switch (X6)** (przełącznik uruchomienia (X6)) zostanie ustawiony w położeniu OFF (WYŁ.), **robot start signal (Y0)** (sygnał uruchomienia robota (Y0)) zostanie wyłączony, a robot zatrzyma się. Jednocześnie **operation lamp (Y1)** (lampka działania (Y1)) na panelu sterowniczym wyłączy się, a **stop lamp (Y2)** (lampka zatrzymania (Y2)) zaświeci się.

Powtór



Wstecz



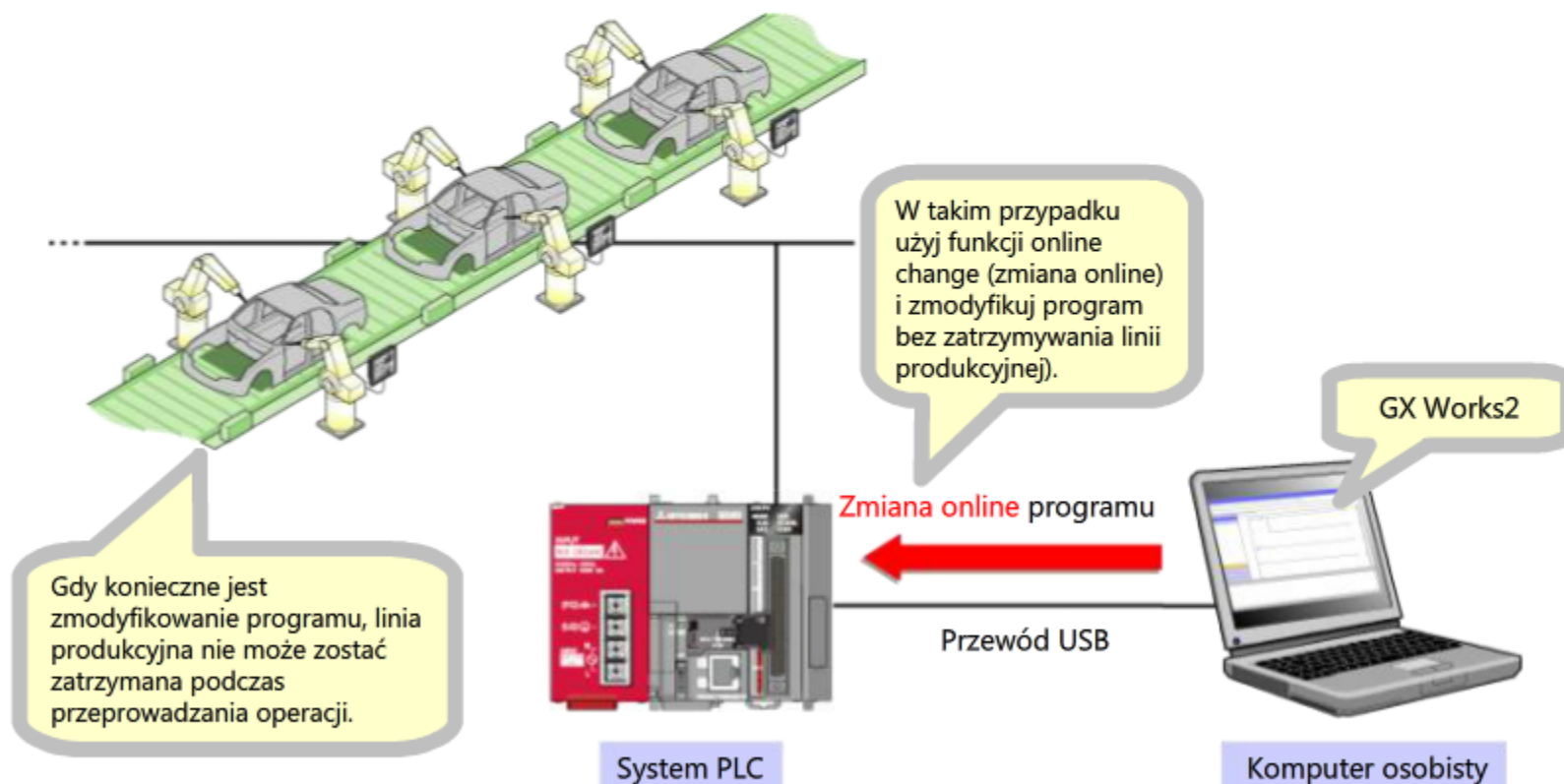
4.7

Obsługa układu PLC

Po zakończeniu weryfikacji operacji uruchom system PLC, aby rozpocząć operację.

Jeśli program wymaga zmodyfikowania w pracującym systemie

Modyfikacja programu, taka jak korekta błędu lub rozbudowa systemu, może być wymagana po uruchomieniu operacji systemu. Standardowo system (moduł CPU) musi zostać zatrzymany w celu zapisania zmodyfikowanego programu, jednak taka sytuacja nie zawsze jest możliwa. Aby rozwiązać ten problem, GX Works udostępnił funkcję online change (zmiana online), która jest stosowana w celu zapisania programów bez zatrzymywania uruchomionego modułu CPU.

Przykład: Samochodowa linia produkcyjna pracująca 24 godziny na dobę

Na następnej stronie spróbuj użyć funkcji online change (zmiana online) przy użyciu symulowanego okna.

4.7

Obsługa układu PLC

MELSOFT Series GX Works2 C:\SequenceProgram\Learning\Robot_Control - [[PRG] MAIN]

Project Edit Find/Replace Compile View Online Debug Diagnostics Tool Window Help

Navigation

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
 - Program
 - MAIN
- Local Device Comment
- Device Memory
- Device Initial Value

Unlabeled

Y0 Robot start signal

Y1 Operation lamp

Y2 Stop lamp

END

1.000ms Local Device not Executed

Zmiana online zmodyfikowanego programu została zakończona.
Kliknij , aby kontynuować.

Unlabeled L02 Host Station 2/75Step

4.8**Podsumowanie**

Powyższe informacje zawierają wyjaśnienie podstawowych danych z zakresu budowy oprogramowania sterownika programowalnego.

W ramach niniejszego kursu omówiono:

- Wymagane elementy z zakresu programowania systemu PLC
- Podstawowe wytyczne dotyczące projektowania programu, włączając użycie komentarzy
- Sposób stosowania oprogramowania GX works2 do przeprowadzania podstawowych zadań z zakresu programowania PC
- Kilka technik stosowanych do przeprowadzania operacji wyszukiwania i usuwania błędów programów PLC

Test**Test końcowy**

Po zakończeniu wszystkich etapów kursu **PLC GX Works2 Basics** możesz teraz przystąpić do testu końcowego. W razie niejasności w zakresie któregoś z tematów, wykorzystaj tę możliwość do ponownego zapoznania się z tymi zagadnieniami.

Test końcowy składa się z 5 pytań (15 elementów).

Możesz zdawać test końcowy dowolną ilość razy.

Jak rozwiązywać test

Po wybraniu odpowiedzi upewnij się, że przycisk **Odpowiedź** został kliknięty. Twoja odpowiedź zostanie utracona, jeśli będziesz kontynuować bez kliknięcia przycisku Odpowiedź. (Zostanie potraktowana jako pytanie, na które nie udzielono odpowiedzi.)

Punktacja końcowa

Liczba prawidłowych odpowiedzi, liczba pytań, procent prawidłowych odpowiedzi i wynik zaliczony/niezaliczony pojawią się na stronie wyniku.

Prawidłowe odpowiedzi: 4

Wszystkie pytania: 4

Procent prawidłowych odpowiedzi: 100%

Aby zaliczyć test musisz odpowiedzieć poprawnie na **60%** pytań.

Kontynuuj**Przeglądaj**

- Kliknij przycisk **Kontynuuj**, aby zakończyć test.
- Kliknij przycisk **Przeglądaj**, aby przeglądać test. (Sprawdzenie prawidłowych odpowiedzi)
- Kliknij przycisk **Spróbuj ponownie**, aby powtórzyć test.

Test

Test końcowy 1



Program, za który ponosiłeś/-aś odpowiedzialność został przejęty przez inną osobę, która stwierdziła trudności ze zrozumieniem elementów sterowania programu. Jakie podjąć właściwe środki zaradcze, aby nie dopuścić do takiego problemu?

- ☐ Przy użyciu funkcji comment (komentarz) GX Works2 podać w programie właściwy tytuł i wyjaśnienie.
- ☐ Ustnie wyjaśnij elementy sterowania nowej osobie.
- ☐ Unikaj zbyt złożonego, obszernego programu.
- ☐ Przekaż tabelę zbiorczą urządzeń we/wy i numerów urządzeń wraz z programem.

Test**Test końcowy 2**

Ukończ prawidłową procedurę programowania.

Krok 1 Projektowanie programu

Krok 2 (Q1)

Krok 3 (Q2)

Krok 4 konwertowanie programów

Krok 5 Zapisywanie projektów

Krok 6 (Q3)

Krok 7 (Q4)

Krok 8 Uruchamianie modułu CPU (RUN (PRACA))

Krok 9 (Q5)

Krok 10 Sprawdzanie działania systemu PLC

Odpowiedź

Wstecz

Test

Test końcowy 3



Wypełnij puste pola, aby zakończyć wyjaśnienia dotyczące wymagań odnośnie do czynności, jakie należy wykonać po zakończeniu programu.

Po zapisaniu programu należy go przetestować w celu sprawdzenia, czy działa zgodnie z oczekiwaniami.

Błąd () (gdy zapisany kod nie przeprowadza czynności zgodnie z oczekiwaniem) jest nazywany () co powoduje wywołanie procesu wyszukiwania przyczyny i naprawy ().

Proces ten jest zasadniczym krokiem podczas tworzenia programów.

Test**Test końcowy 4**

Wybierz właściwe zastosowanie każdej funkcji GX Works2.

Funkcja	Zastosowanie
Simulation (Symulacja)	--Select--
Forced input output registration/cancellation (Rejestracja/anulowanie wymuszonego sygnału wejściowego/wyjściowego)	--Select--
Change current value (Zmień bieżącą wartość)	--Select--
Ladder monitor (Monitorowanie drabinki)	--Select--
Watch (Obserwacja)	--Select--

Odpowiedź

Wstecz

Test**Test końcowy 5**

Wybierz właściwy opis funkcji online change (zmiana online).

- ☐ Funkcja automatycznie zatrzymuje CPU, zapisuje program w CPU, a następnie automatycznie uruchamia CPU.
- ☐ Funkcja porównuje program w działającym module CPU z programem otwartym przez GX Works2.
- ☐ Funkcja może zapisać program w module CPU po bezpiecznym zatrzymaniu działającego modułu CPU.
- ☐ Funkcja może zapisać program w działającym module CPU bez jego wyłączenia.

Odpowiedź

Wstecz

Test**Wynik testu**

Test końcowy został zakończony. Twoje wyniki są przedstawione poniżej.
Aby zakończyć test końcowy, przejdź do następnej strony.

Prawidłowe
odpowiedzi: **5**

Wszystkie pytania: **5**

Procent
prawidłowych **100%**

[Kontynuuj](#)[Przeglądaj](#)

Gratulacje. Test został zaliczony.

Kurs Podstawy **GX Works2** został ukończony.

Dziękujemy za wzięcie udziału w kursie.

Mamy nadzieję, że poruszone tematy były interesujące, a informacje uzyskane w trakcie tego kursu będą przydatne w przyszłości.

Możesz przeglądać kurs dowolną ilość razy.

Przeglądaj

Zamknij