



Szkolenie PLC cz. 1

Wychodząc naprzeciw oczekiwaniom naszych klientów, rozpoczynamy nowy cykl szkoleń, dotyczący samodzielnego programowania sterowników logicznych PLC. Przez kolejne miesiące, nasi czytelnicy będą mieli okazję poznać wszystkie tajniki programowania z wykorzystaniem popularnego języka drabinkowego (z ang. LAD – Ladder Diagram) oraz diagramu funkcyjnego (z ang. FBD – Function Block Diagram). Rozpoczynając od najprostszych, podstawowych instrukcji, realizując szereg ćwiczeń oraz analizując zamieszczone przykłady, użytkownicy będą w stanie tworzyć coraz bardziej zaawansowane algorytmy sterowania.

CHARAKTERYSTYKA STEROWNIKA.

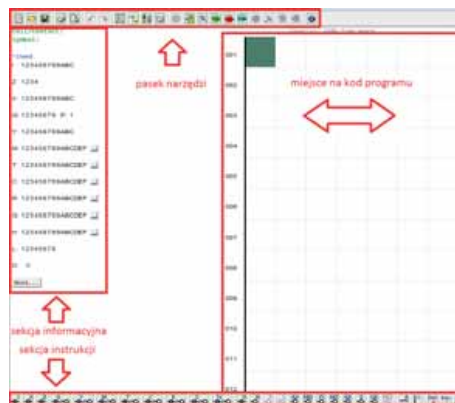
W pierwszej odsłonie szkolenia zajmiemy się omówieniem środowiska programistycznego, oraz najważniejszych instrukcji języka drabinkowego. Zanim jednak przejdziemy do zasadniczej części, warto poświęcić kilka słów na temat sterownika TECO SG2. Jest to kompaktowe urządzenie, znajdujące zastosowanie zarówno w prostych aplikacjach domowych, jak i bardziej zaawansowanych wdrożeniach prze-



mysłowych. Ze względu na bardzo dobry stosunek jakości do ceny, sterownik ten stanowi silną konkurencję dla bardziej znanych oraz zdecydowanie droższych zamienników. Urządzenie posiada kilka modeli, różniących się parametrami takimi jak napięcie zasilania, a także rodzaj i ilość sygnałów I/O. Jednostka bazowa może pracować samodzielnie, lub w razie potrzeby może być rozbudowana o konkretne moduły rozszerzeniowe (cyfrowe sygnały I/O, moduły przetworników A/C oraz moduły komunikacyjne). W najbardziej zaawansowanej konfiguracji, otrzymujemy sterownik z trzema modułami cyfrowymi, jednym modułem analogowym oraz jedną jednostką komunikacyjną, co gwarantuje wystarczające możliwości aplikacyjne w swojej dziedzinie zastosowań.

OPROGRAMOWANIE

Darmowa platforma programistyczna SG2client (do pobrania ze strony www.intrrol.pl), jest środowiskiem intuicyjnym oraz prostym w obsłudze. Programowanie sterownika odbywa się poprzez podłączenie go do interfejsu szeregowego komputera za pomocą specjalnego kabla komunikacyjnego, a następnie załadowanie do pamięci EEPROM wcześniej stworzonego programu. Możliwe jest także tworzenie programu „ręcznie”, wykorzystując przyciski na klawiaturze jednostki centralnej, jednakże sposób ten jest zdecydowanie bardziej uciążliwy i nie zaleca się go początkującym użytkownikom. Poniżej zamieszczono zrzut ekranu przedstawiający środowisko programistyczne SG2client.



Okno główne składa się z czterech podstawowych sekcji: paska narzędzi, sekcji informacyjnej, sekcji wykorzystywanych instrukcji oraz właściwego kodu programu. Funkcje paska narzędzi, oprócz standardowych funkcji zapisu i odczytu pliku projektu, sprowadzają się także do obsługi symulatora programu oraz transmisji skompilowanego kodu z i do sterownika. Sekcja Informacyjna odpowiedzialna jest za monitorowanie wykorzystania w programie odpowiednich wejść i wyjść cyfrowych, liczników czy timerów. Sekcja instrukcji zawiera wszystkie funkcje logiczne oraz arytmetyczne, obsługiwane przez dany sterownik. Ostatnia sekcja stanowi miejsce tworzenia przez użytkownika kodu drabinkowego.

PIERWSZY PROGRAM W JĘZYKU DRABINKOWYM STEROWNIKA SG2

Programowanie z wykorzystaniem kodu drabinkowego przypomina tworzenie schematu elektrycznego. Kod jest interpretowany (skanowany) od lewej do prawej, z góry na dół. W momencie osiągnięcia ostatniego „szczebla drabiny”, czyli ostatniej linii instrukcji, skanowanie rozpoczyna się od początku. W przypadku środowiska SG2client, rozróżniamy dwa podstawowe typy instrukcji: instrukcje wejściowe (styki I, Z, X) oraz instrukcje wyjściowe. Te ostatnie można podzielić na wyjścia logiczne (cewki, przekaźniki Q, Y) oraz wyjścia funkcyjne (np. liczniki C, komparatory G, itd), realizujące przypisane im zadania. W zależności od kolejności, rodzaju oraz parametryzacji wykorzystanych instrukcji, programista steruje przepływem informacji cyfrowej co z kolei definiuje działanie programu wynikowego.

PROGRAM 1

Funkcjonalność: Program powinien zwiierać przekaźnik wyjściowy po aktywacji styku wejściowego. Przekaźnik powinien zostać rozarty po aktywacji drugiego z wejść.

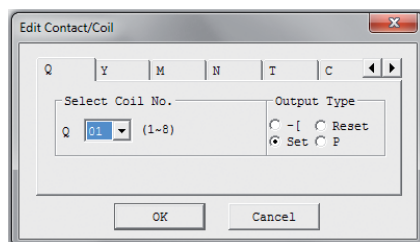


Przykład:

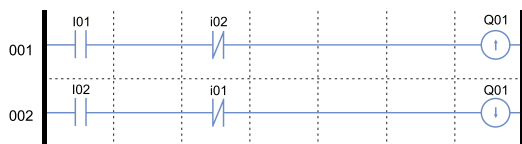
Załączanie/wyłączanie oświetlenia dwoma różnymi przyciskami.



Jak widać na załączonej ilustracji, do realizacji zadania wykorzystane zostały cztery elementy logiczne, dwa wejściowe oraz dwa wyjściowe. Jak zostało wspomniane wcześniej, program interpretowany jest od lewej do prawej, z góry na dół. Lewą pogrubioną linię można interpretować jako potencjał dodatni (+), prawą linię jako masę (-), a wykonywanie programu jako przepływ prądu pomiędzy tymi dwoma liniami. Styki I01 oraz I02 reprezentują pierwsze dwa wejścia cyfrowe sterownika. Przekaznik wyjściowy Q01 reprezentuje pierwsze wyjście cyfrowe urządzenia. Wybór charakteru przekaznika odbywa się podczas umieszczania symbolu na schemacie. Dostępne są cztery funkcje widoczne na ilustracji poniżej – cewka zwykła, cewka załączająca (SET), wyłączająca (RESET) oraz cewka typu przerytnik (z pamięcią).

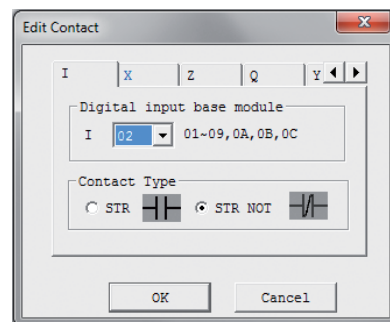


Opis działania programu: W momencie aktywacji (zwarcia) styku I01, cewka Q01 przechodzi w tryb SET, czyli przyjmuje stan logiczny „1”, a rzeczywisty przekaznik wyjściowy sterownika zostaje zwarty. Drugi ze styków jest nieaktywny (rozarty). W momencie aktywacji styku I02, cewka Q01 przechodzi w tryb RESET, przyjmując stan logiczny „0”, a przekaznik zostaje rozarty. Jednoczesna aktywacja obu styków wejściowych może skutkować uszkodzeniem sterownika, gdyż przekaznik wyjściowy Q01 będzie zwierany i rozwierany zgodnie z częstotliwością skanowania. Poniżej przedstawiona została modyfikacja kodu, uwzględniająca zabezpieczenie chroniące przed takim stanem.

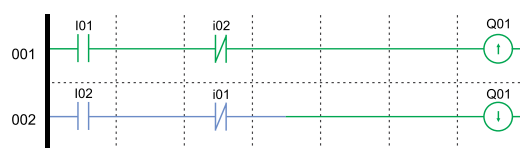


Podczas inicjacji programu, oba styki wejściowe są rozarte. W momencie aktywacji styku normalnie otwartego I01 (NO – z ang. normal opened) wyjście Q01 zostaje zwarte, jednocześnie druga linia kodu jest blokowana poprzez umieszczenie na niej styku normalnie zamkniętego i01 (NC – z ang. normal closed), który jest zanegowanym sygnałem I01. Wybór pomiędzy rodzajem styku dokonywany

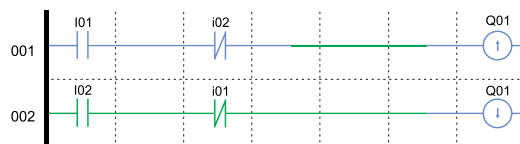
jest w momencie umieszczania elementu na schemacie.



Do momentu dezaktywacji styku wejściowego I01, styk i01 stanowi przerwę w obwodzie i niemożliwe jest rozwarcie wyjścia Q01 poprzez zwarcie drugiego z wejść. Sytuację taką obrazuje poniższa ilustracja.



Na zielono zaznaczono elementy stanowiące zwarcie. Widzimy, że styk i01 blokuje propagację ewentualnego sygnału rozłączającego przekaznik. Dopiero po rozłączeniu styku I01, jego zanegowany sygnał i01 przyjmuje wartość logiczną „1” i umożliwia wyłączenie przekazu wyjściowego Q01. Taki stan został przedstawiony na ilustracji poniżej.



W przypadku sytuacji odwrotnej, analiza jest analogiczna gdyż do momentu rozłączenia styku I02, niemożliwe jest ponowne zwarcie cewki wyjściowej.

PROGRAM 2

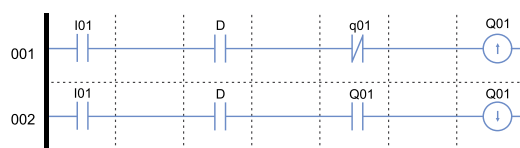
Funkcjonalność: Program powinien zewrzeć przekaznik wyjściowy po aktywacji styku wejściowego. Przekaznik powinien zostać rozarty po ponownej aktywacji tego samego styku wejściowego.

Zastosowanie w praktyce:

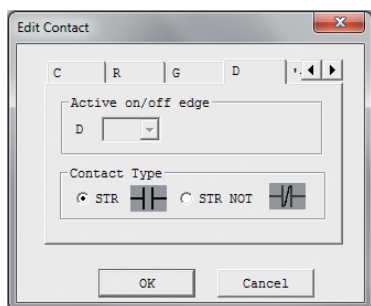
Załączanie/wyłączanie oświetlenia tym samym przyciskiem.

To zadanie można rozwiązać na 2 sposoby:

Sposób 1 (wykorzystujący standardowe elementy języka drabinkowego)

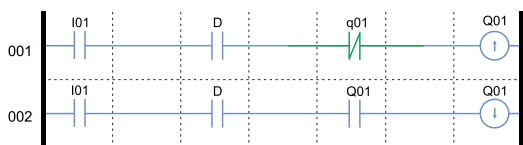


W powyższym przykładzie zastosowane zostały elementy wykrywające zmianę sygnału na przeciwny, czyli styki wykrywające zbocze narastające lub opadające. Wybór charakteru styku odbywa się w momencie umieszczania go na schemacie.

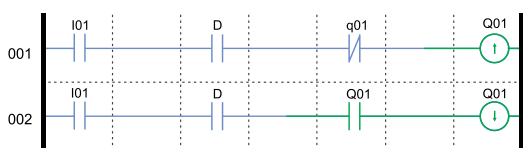


Zasada działania styku polega na wykrywaniu zmiany sygnału na przeciwny i aktywację samego siebie na jeden cykl skanowania programu. Do momentu kolejnej zmiany sygnału, styk taki stanowi przerwę w obwodzie. Dzięki temu, programista może zdefiniować akcję wykonywaną tylko raz, dokładnie w momencie modyfikacji sygnałów wejściowych.

W momencie inicjacji programu, stan styków jest taki jak na rysunku poniżej.



W tej sytuacji możliwe jest zwarcie cewki Q01 poprzez aktywację styku wejściowego I01. Cewka wykrywająca zbocze narastające D na szczeblu 001, pozwoli na propagację sygnału tylko przez jeden cykl skanowania programu. Po rozwarciu styku wejściowego I01, stan pozostałych styków będzie następujący:



Zwarcie przekaźnika Q01, spowodowało rozłączenie styku q01, gdyż jest to jego negacja. Jednocześnie w drugim szczeblu kodu, styk Q01 zostanie zwarty i umożliwi w przyszłości wyłączenie przekaźnika. Cewka D została zresetowana i jest gotowa na wykrycie kolejnego impulsu narastającego na wejściu I01. Druga

aktywacja tego styku, spowoduje propagację sygnału w linii 002 i ostatecznie rozłączenie cewki Q01. Stan ten jest analogiczny do stanu początkowego.

W powyższym przykładzie wykorzystaliśmy informacje dotyczące aktualnego stanu cewek wyjściowych. Na podstawie tej informacji, programista może zaprogramować odpowiednie działanie w zależności od konkretnego stanu wyjść – w naszym przypadku zmianę na stan przeciwny. Rozłączenie następuje tylko gdy cewka jest zwarta i odwrotnie. W ten sposób otrzymaliśmy program posiadający pamięć. Stan przekaźników nie jest zależny tylko od stanu sygnałów wejściowych ale także od poprzedniego stanu wyjść cyfrowych. Urządzenie takie nosi nazwę automatu sekwencyjnego.

Sposób 2 (wykorzystujący dodatkowe elementy środowiska SG2client)



W tym rozwiązaniu zastosowaliśmy wyżej wspomnianą cewkę wyjściową typu przerzutnik, czyli cewkę „z pamięcią”. Cewka P Q01 zmienia stan na przeciwny za każdym razem gdy dojdzie do niej sygnał z lewej strony. Pierwsza aktywacja styku wejściowego:



Druga aktywacja styku wejściowego:



Zastosowanie cewki wykrywającej zbocze narastające D, gwarantuje jednorazowe zwarcie styku wyjściowego Q01. Brak tej cewki, skutkowałby ciągłym przelączaniem przekaźnika z częstotliwością równą częstotliwości skanowania.

Drugi ze sposobów jest zdecydowanie prostszy i krótszy, jednakże jest mniej dydaktyczny niż pierwszy, który przybliży więcej cech programowania w języku drabinkowym.

W kolejnej części szkolenia: zastosowanie cewek pomocniczych, obsługa liczników oraz timerów, komparatory analogowe. Zapraszamy do lektury „Pod kontrolą” 2/2012

Szkolenie prowadzi:
Dominik Szewczyk



Zapraszamy na targi **Automaticon 2012**



WARSZAWSKIE CENTRUM **EXPO XXI**
Warszawa - Wola, ul. Prądzyńskiego 12
20-23 marca 2012

Jesteśmy tam gdzie zawsze:

Hala **1** Stoisko **A31**