

12. PIERWSZY PROGRAM W JĘZYKU SFC

12.1. CoDeSys a SFC

Kroki

W programie dostępne są dwa typy kroków: kroki uproszczone i kroki zgodne z normą IEC 61131-3 – kroki zgodne z normą będą w dalszej części opracowania nazywane krokami IEC. Korzystanie z kroków IEC jest uzależnione od ustawienia opcji „*Use IEC-Steps*” i dołączenia dodatkowej biblioteki: „*Iecsfc.lib*” – problem ten zostanie omówiony szczegółowo w punkcie 2.1.1.

Tranzycje

Warunki przejść w języku ST można w CoDeSys definiować bezpośrednio z poziomu edytora języka SFC obok symbolu tranzycji. Warunki w językach IL, LD, FBD muszą, a warunki w języku ST mogą, być zapisywane w postaci osobnych *funkcji* tworzonych w edytorze tranzycji.

Uwaga! Do zapisu koniunkcji nie można wykorzystywać symbolu „&” – dostępny jest tylko operator AND.

Skok do kroku

Edytor języka SFC programu CoDeSys nie pozwala na łączenie odległych kroków za pomocą linii łączących, takie kroki można ze sobą połączyć jedynie wykorzystując operację *skoku do kroku*.

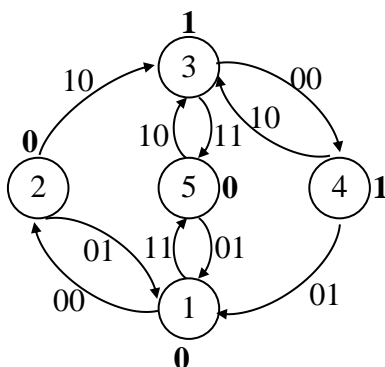
12.2. Przykład

Sposób wprowadzania grafu SFC w CoDeSys zostanie wyjaśniony w oparciu o przykład opisany w materiale poświęconym syntezie układów sekwencyjnych.

Zadanie polega na napisaniu programu sterującego pracą urządzenia sterowanego dwoma przyciskami: Z (*załącz*) i W (*wyłącz*). Naciśnięcie przycisku Z ($Z=1$) załącza urządzenie ($Y=1$), urządzenie pracuje również po zwolnieniu tego przycisku ($Z=0$). Naciśnięcie przycisku W ($W=1$) wyłącza urządzenie ($Y=0$). Urządzenie nie pracuje gdy wciśnięte są obydwa przyciski. Dodatkowo, nie jest możliwa jednoczesna zmiana stanu obydwu przycisków.

Przygotowując graf przejść tego układu wyodrębniono pięć stanów, w których może znajdować się obiekt sterowania:

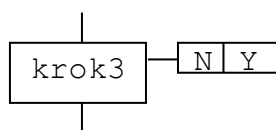
- 1 – przycisk Z zwolniony, przycisk W wciśnięty i urządzenie nie pracuje ($Z=0, W=1, Y=0$),
- 2 – przyciski Z i W zwolnione i urządzenie nie pracuje ($Z=0, W=0, Y=0$),
- 3 – przycisk Z wciśnięty, przycisk W zwolniony i urządzenie pracuje ($Z=1, W=0, Y=1$),
- 4 – przyciski Z i W zwolnione i urządzenie pracuje ($Z=0, W=0, Y=1$),
- 5 – przyciski Z i W wciśnięte i urządzenie nie pracuje ($Z=1, W=1, Y=0$).



Rys. 12.1. Graf przejść układu

Graf SFC

Graf powinien być zbudowany z pięciu kroków. W krokach 3 i 4 urządzenie powinno być włączone a w pozostałych wyłączone. Akcje związane z krokami 3 i 4 powinny urządzenie włączać a więc akcje te powinny zmiennej wyjściowej Y nadawać wartość *prawda* w czasie aktywności tego kroku. Rezultat taki można osiągnąć zapisując *akcję nieprzechowywaną* dla tych kroków.



Rys. 12.2. Skojarzenie kroku z akcją nieprzechowywaną w postaci zmiennej boolowskiej

Kroki 1, 2 i 5 nie muszą być połączone z żadną akcją. Ustalenie akcji dla kroków 3 i 4 powoduje, że w czasie kiedy kroki te są aktywne zmiennej Y przypisywana jest wartość *prawda*, a kiedy kroki te są nieaktywne, tzn. aktywne są kroki 1, 2 lub 5, zmiennej Y przypisywana jest wartość *falsz*.

Pomiędzy dwoma kolejnymi *krokami* grafu SFC musi znajdować się dokładnie jedna *tranzycja*, więc z *rys. 12.1* wynika, że niezbędnych jest 10 *tranzycji*. Warunek przejścia z kroku 1 do kroku 2 powinien być spełniony, gdy wymuszane jest przejście ze stanu 1 do stanu 2, czyli gdy wyłączone są przyciski *załłącz* i *wyłłącz*. Warunek taki można zapisać w postaci następującego wyrażenia języka ST:

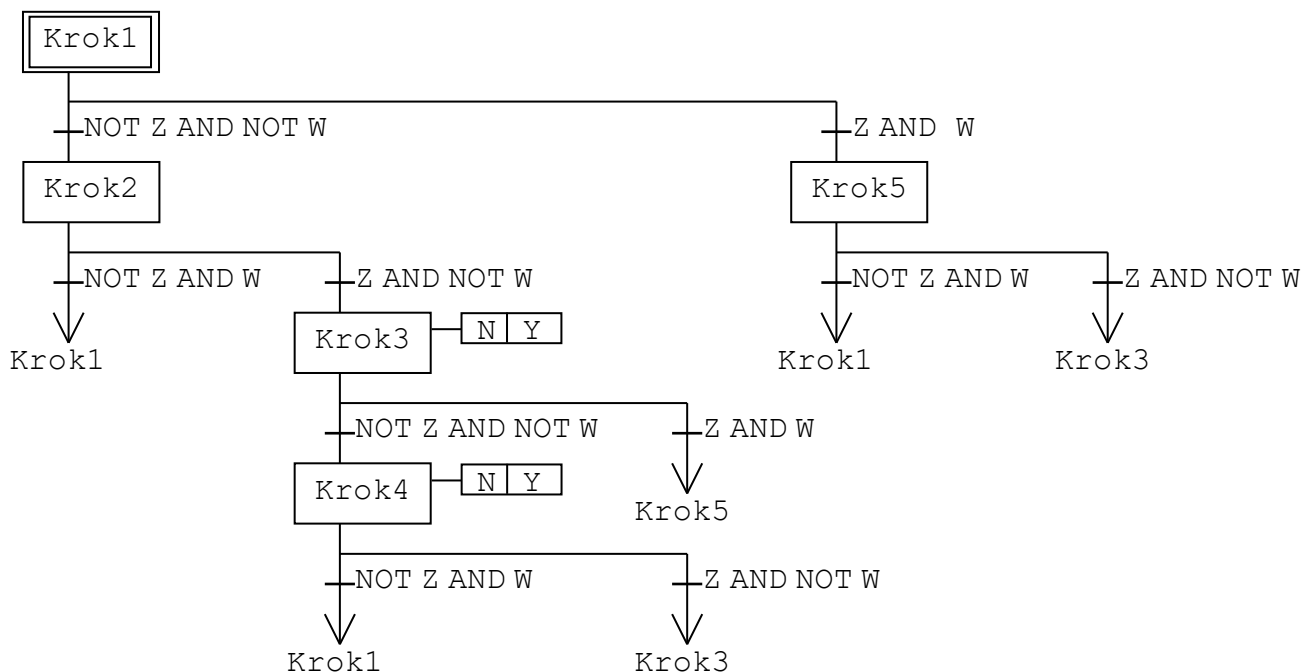
$$\text{NOT Z AND NOT W}$$

Warunek przejścia z kroku 2 do kroki 1 powinien być spełniony, gdy wymuszane jest przejście ze stanu 2 do stanu 1, czyli gdy wyłączony jest przycisk *załłącz* i włączony jest przycisk *wyłłącz*. Warunek ten można zapisać jako:

$$\text{NOT Z AND W}$$

Po zapisaniu wszystkich warunków przejść i zakładając, że po włączeniu sterownika urządzenie nie powinno być automatycznie załączane (ale dopiero po spełnieniu odpowiednich warunków) – tzn. zakładając, że *krokiem początkowym* jest np. krok pierwszy, graf programu w języku SFC można narysować w postaci *rys. 12.3*.

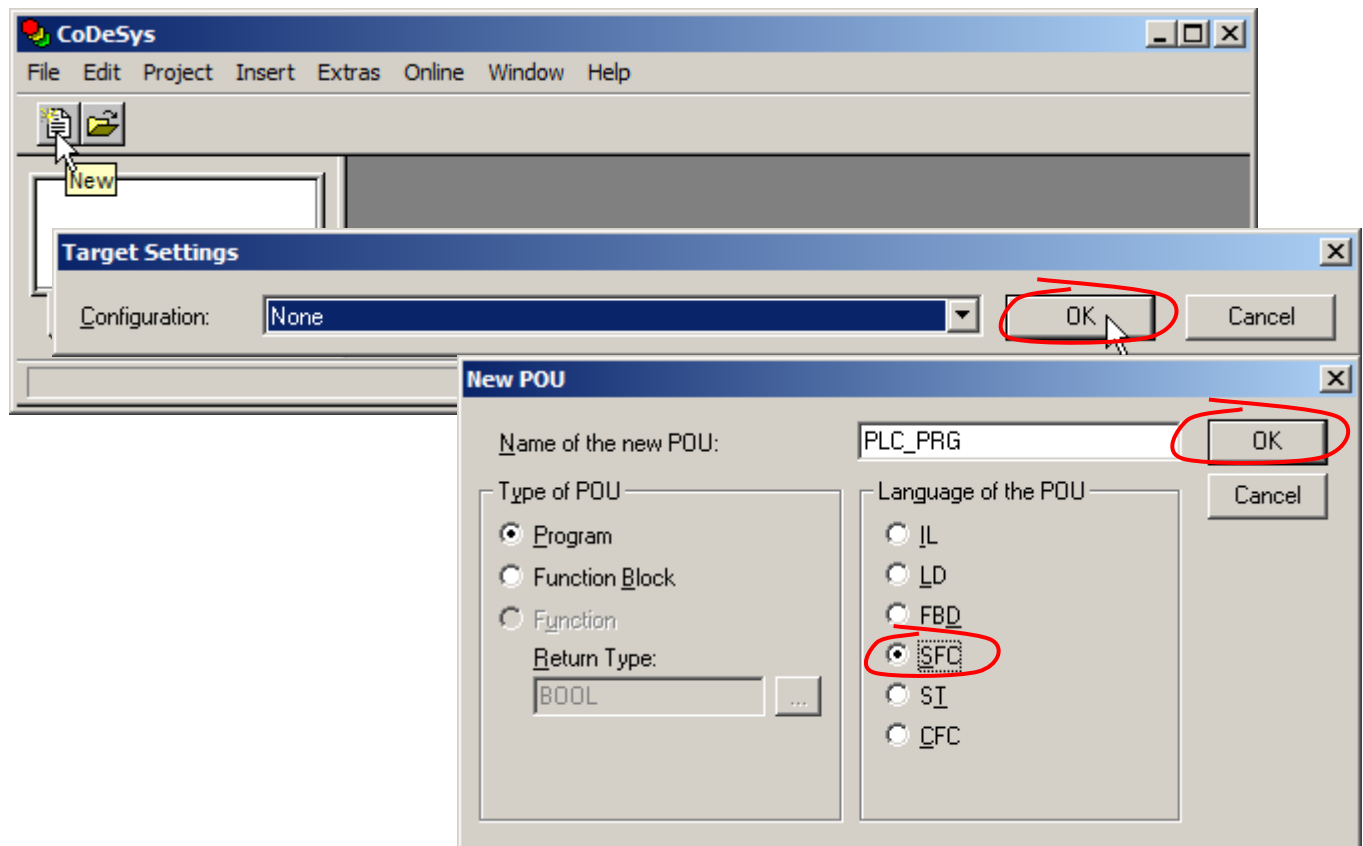
Program załączający i wyłączający urządzenie został zrealizowany w poprzednim punkcie w oparciu o graf przejść w wersji 2 (*rys. 12.1*). Gdyby redukcja stanów nie została wykonana program musiałby być narysowany w postaci przedstawionej na *rys. 12.1*.



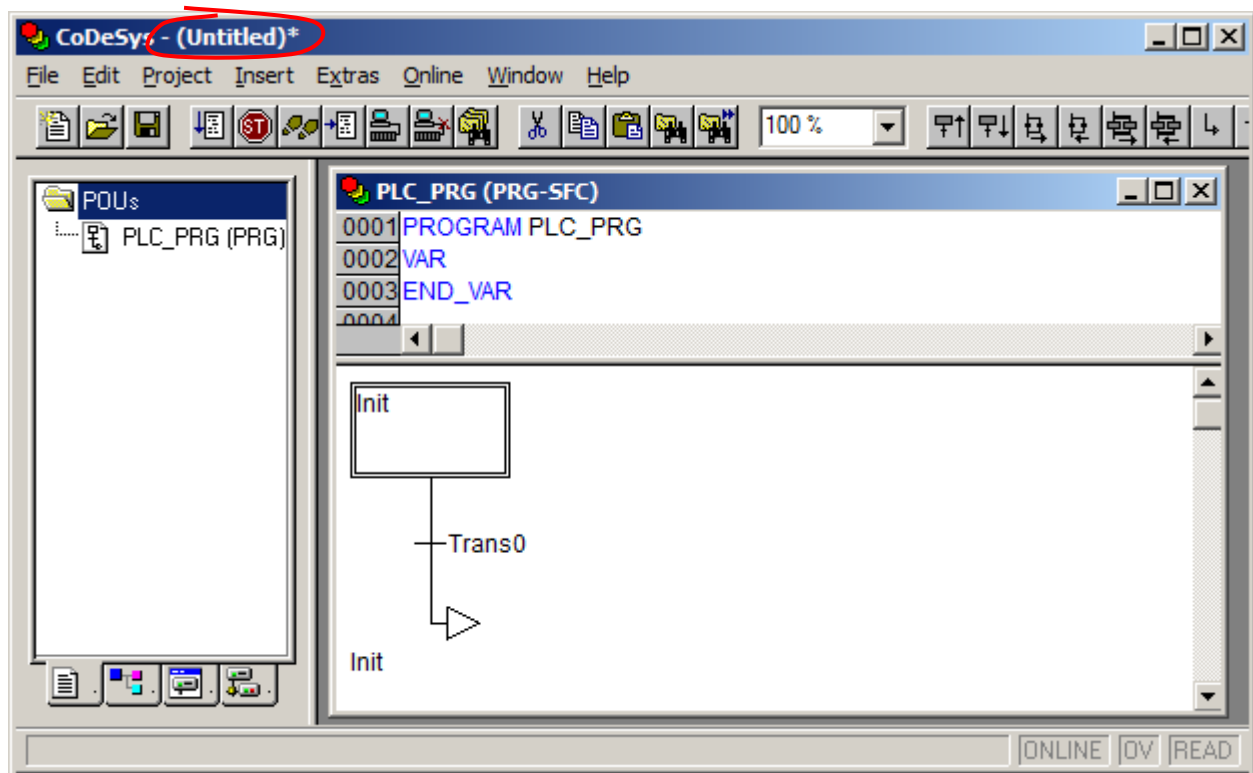
Rys. 12.3. Przykładowy program w języku SFC.

12.3. Nowy projekt

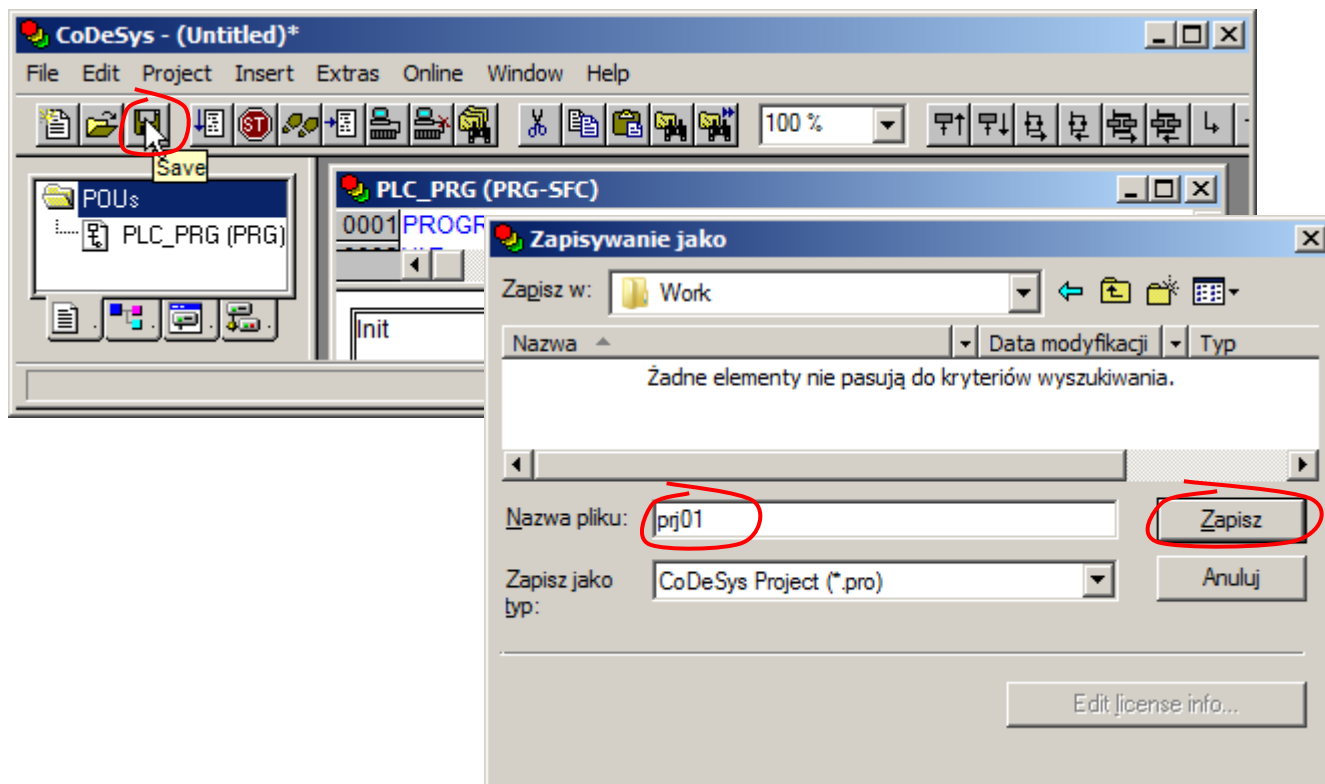
Przygotowanie projektu z programem głównym w języku SFC wymaga wykonania kroków przedstawionych na kolejnych rysunkach:



Po utworzeniu, nowy projekt otrzymuje nazwę **Untitled** i jest automatycznie otwierany w środowisku.



Na koniec, projekt należy zapisać na dysku nadając mu wybraną nazwę.



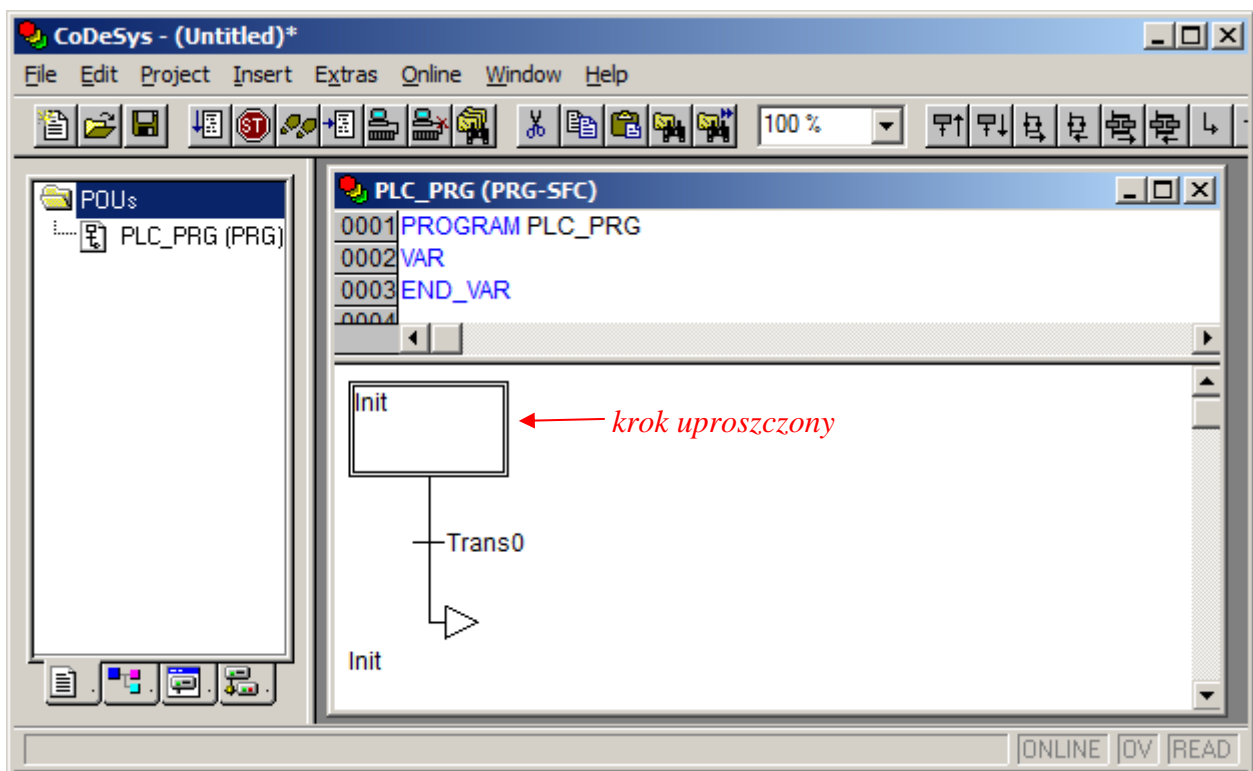
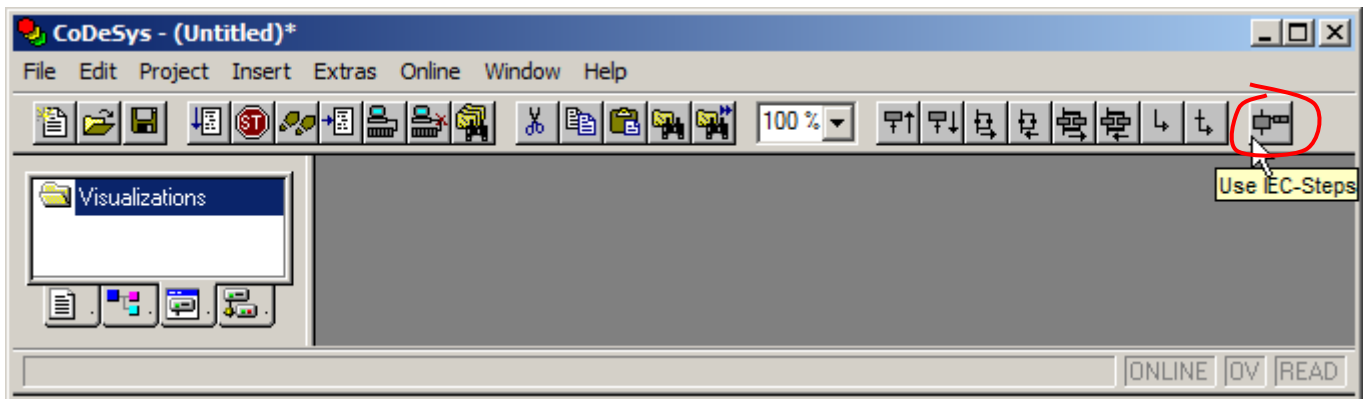
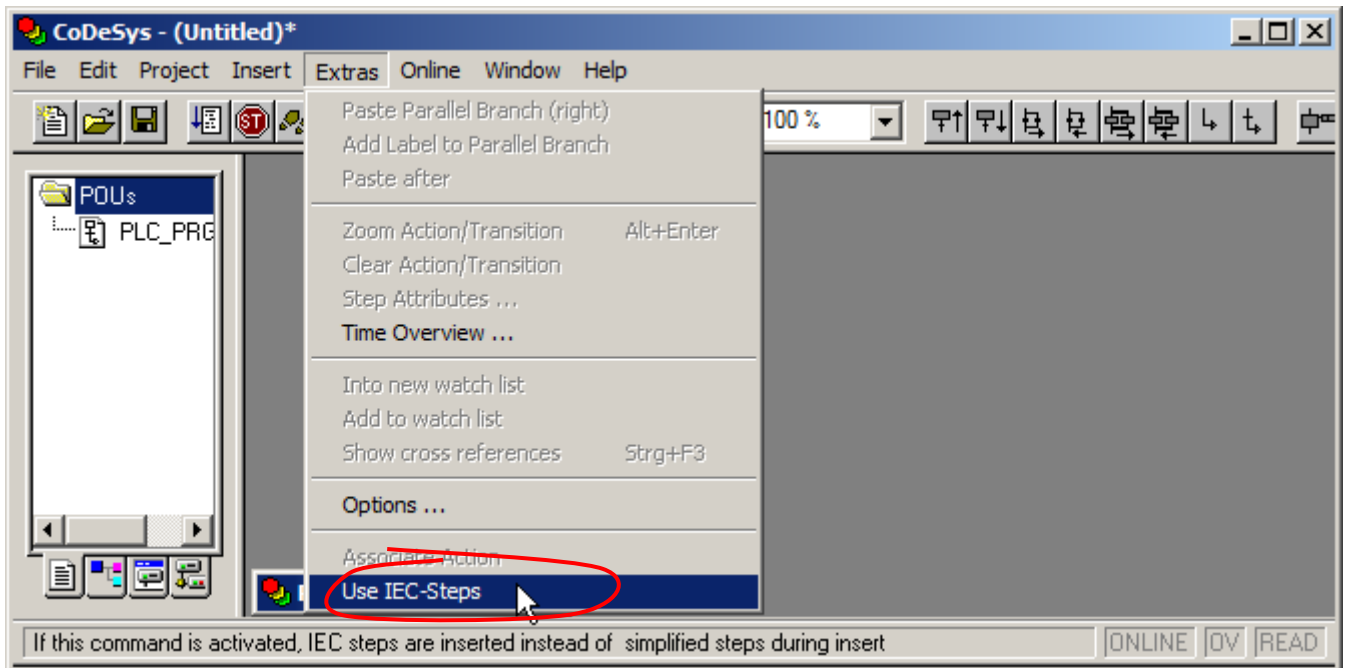
W kolejnym kroku użytkownik może zdefiniować zmienne niezbędne do realizacji programu: 2 zmienne wejściowe: Z i W oraz 1 zmienną wyjściową Y. Sposób wprowadzania zmiennych zostanie pominięty (został on szczegółowo omówiony przy okazji omawiania edycji programów w językach FBD i LD).

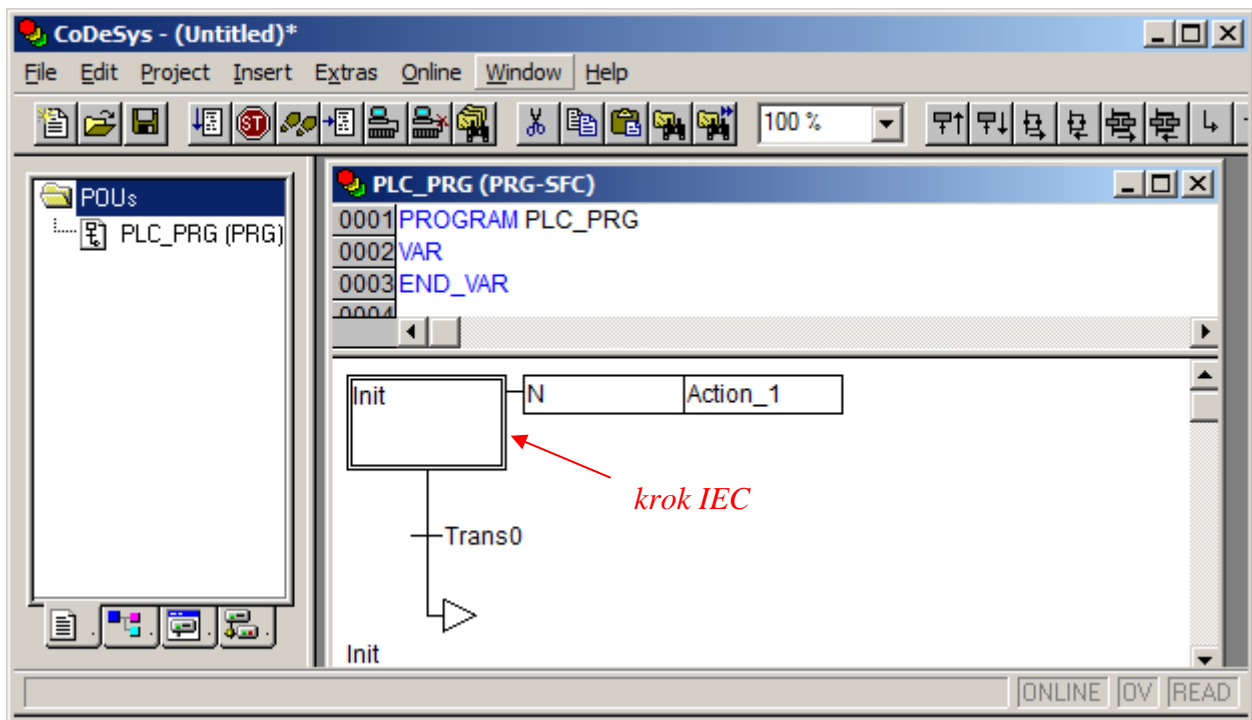
Program może napisany z użyciem kroków uproszczonych i/lub kroków IEC, obydwa typy kroków mogą występować w jednym programie. W materiale omówione zostanie wykorzystanie wyłącznie kroków IEC.

Korzystanie z kroków IEC

Korzystanie z kroków IEC jest uzależnione od ustawienia opcji „Use IEC-Steps” – opcja ta jest dostępna jedynie w przypadku edycji programu w języku SFC.

Typ kroku początkowego, który jest automatycznie wstawiany podczas tworzenia programu, jest uzależniony od ustawienia opcji podczas edycji poprzedniego programu w SFC. Domyślnie opcja ta jest wyłączona więc podczas tworzenia pierwszego programu krok początkowy jest krokiem uproszczonym. Zmiana typu kroku już wstawionego nie jest możliwa więc jeżeli pierwszy krok ma być krokiem IEC należy utworzyć ponownie projekt z programem w języku SFC.

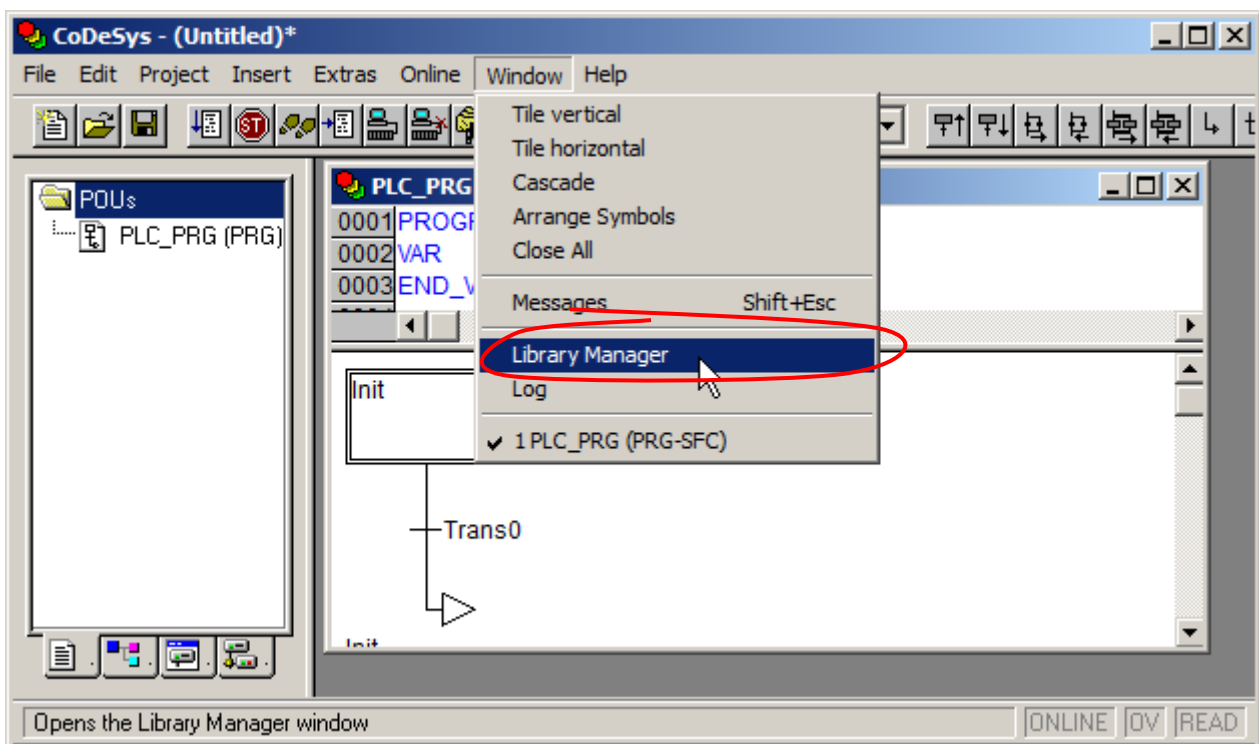




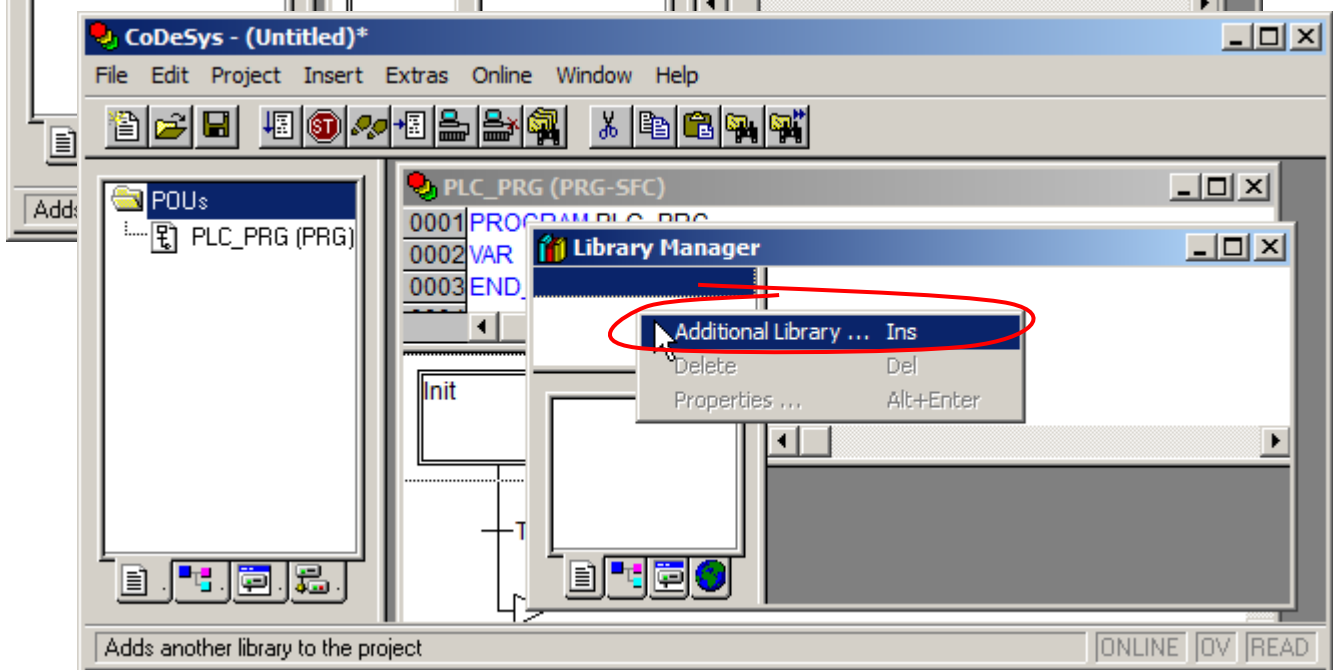
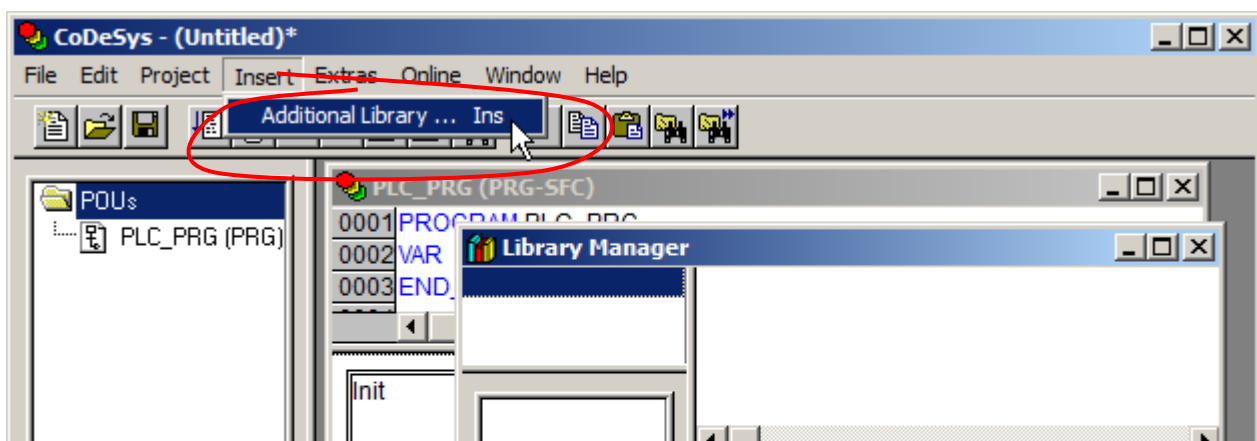
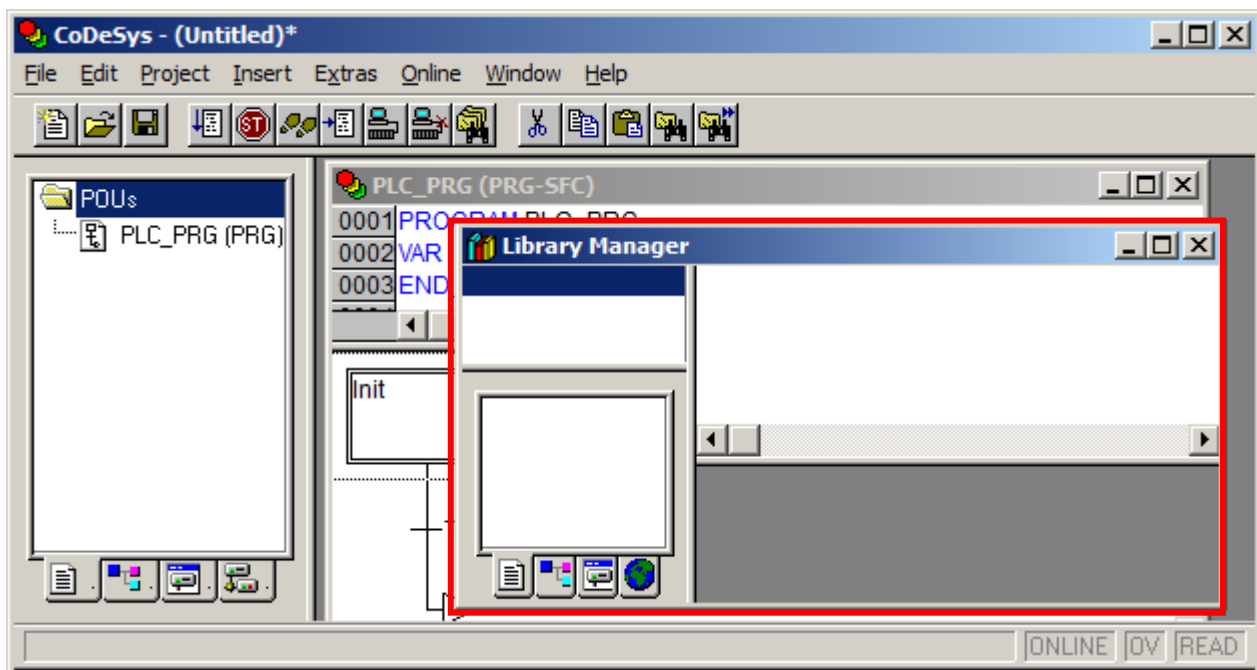
Porównując rysunki przedstawiające projekty z programami w SFC można zauważyć różnicę pomiędzy krokiem uproszczonym a krokiem IEC – po prawej stronie kroku IEC wyświetlane są informacje o zmiennych i akcjach skojarzonych z krokiem (domyślnie z krokiem kojarzona jest akcja lub zmienna nieprzechowywana (N) o nazwie Action_1.

Biblioteka lecsfc.lib

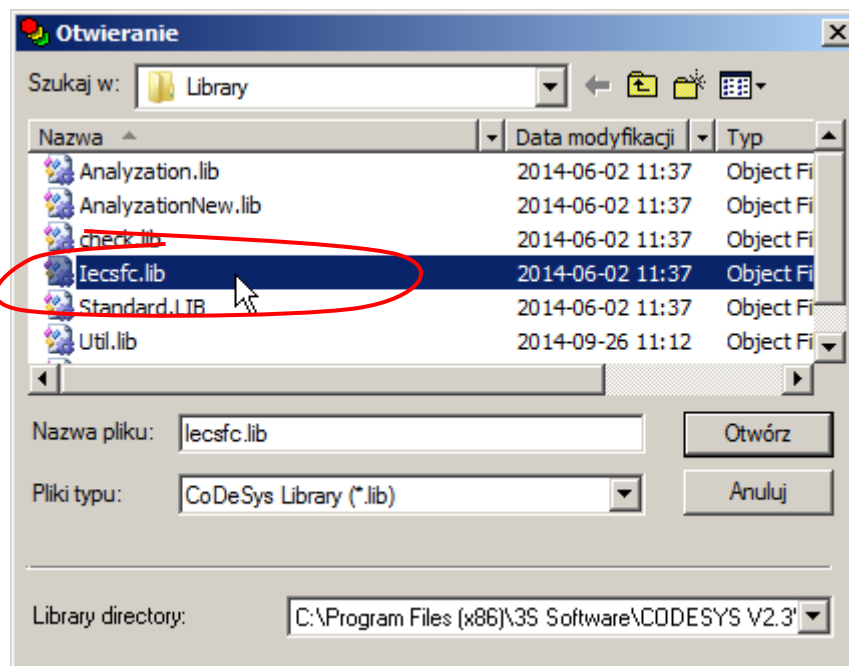
Korzystanie z kroków IEC wiąże się w CoDeSys z koniecznością dołączenia biblioteki „Iecsfc.lib”. Dodatkowe biblioteki mogą być dołączane do projektu z wykorzystaniem menadżera bibliotek:



Po uaktywnieniu menadżera, zarówno w menu głównym programu jak i w menu podręcznym menadżera staje się dostępna opcja umożliwiająca dodanie nowej biblioteki.

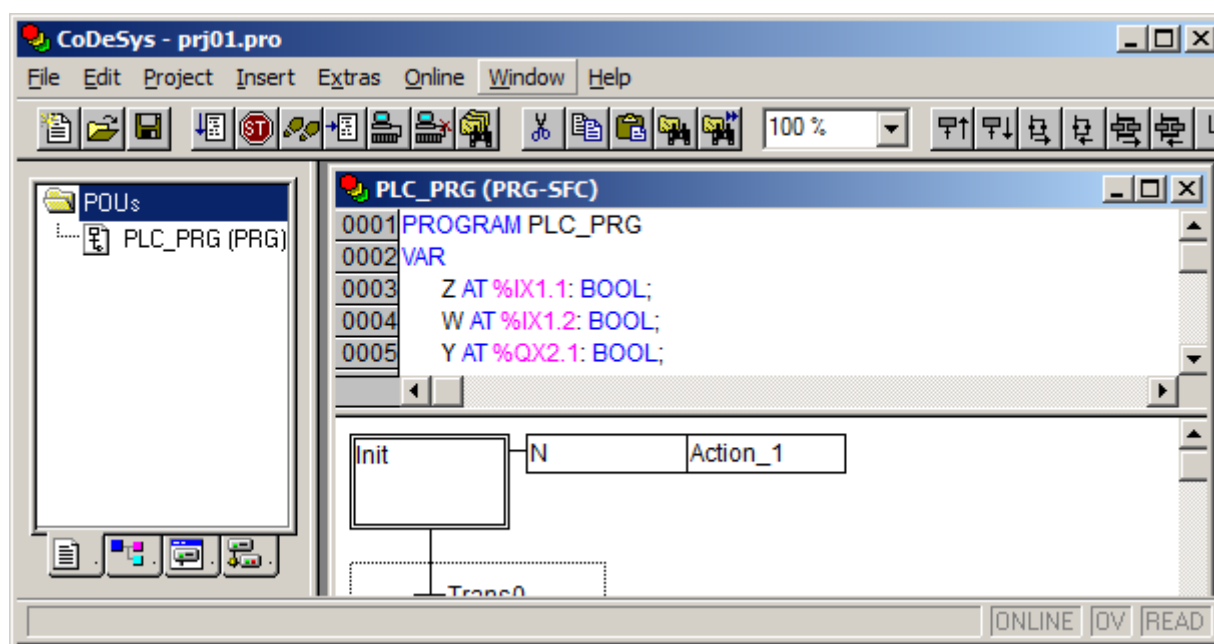


Po wskazaniu właściwej biblioteki okno menadżera może zostać zamknięte.



Edycja programu

Po zdefiniowaniu zmiennych należy uzupełnić program



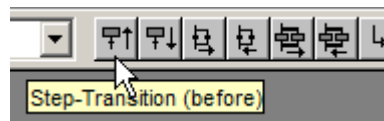
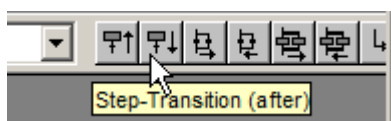
dodając:

- krok i tranzycie,
- warunki przejścia dla tranzycji,
- kojarząc z krokami zmienne.

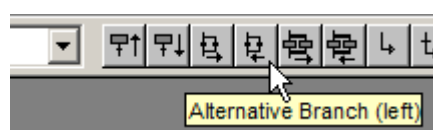


Podstawowe narzędzia

Na kolejnych rysunkach pokazane zostały podstawowe narzędzia edycyjne. Pierwsze dwa przyciski paska narzędziowego pozwalają na dodanie pary „*krok – tranzycja*” przed lub po bieżącym kroku lub bieżącej tranzycji.



Kolejne dwa przyciski pozwalają na użycie *symboli rozbieżności* – gałąź alternatywna może być dodana z prawej lub lewej strony bieżącej tranzycji.



Dodanie gałęzi alternatywnej powoduje automatyczne uzupełnienie programu o symbole rozbieżności i zbieżności. Łączenie gałęzi alternatywnych jest przerywane po wstawieniu *symbolu skoku do kroku*. Symbol ten może być wstawiony przy pomocy przedostatniego przycisku, który umożliwia dodanie *skoku do kroku* po bieżącej tranzycji znajdującej się w jednej z gałęzi alternatywnych.



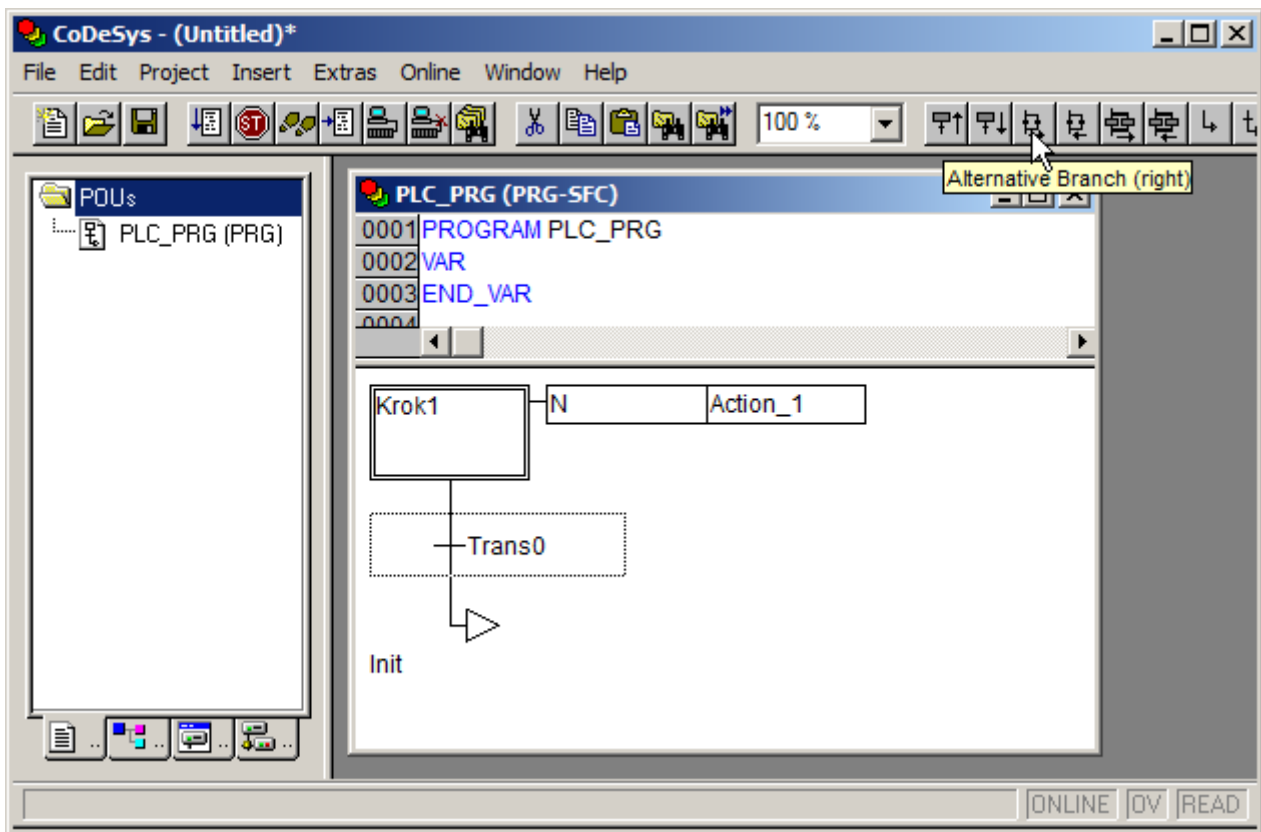
Ostatni przycisk paska narzędzi wstawia tranzycję i skok do kroku po bieżącej tranzycji znajdującej się w jednej z gałęzi realizowanych współbieżnie.

Uwaga!

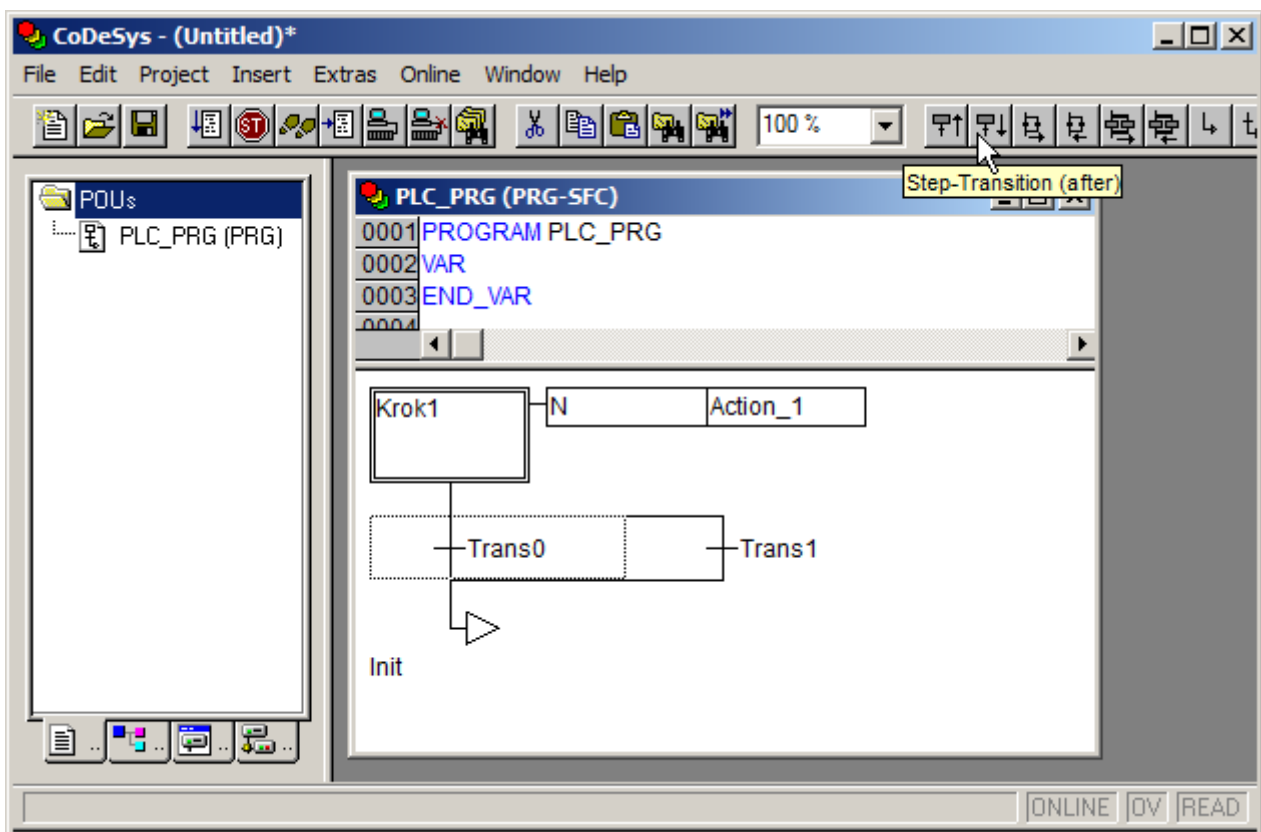
W przypadku błędnego wstawienia kroku lub tranzycji, elementy te mogą zostać usunięte jedynie po uprzednim zaznaczeniu pary: *krok –tranzycja* (nie jest możliwe usunięcie wyłącznie jednego elementu).

Edycja

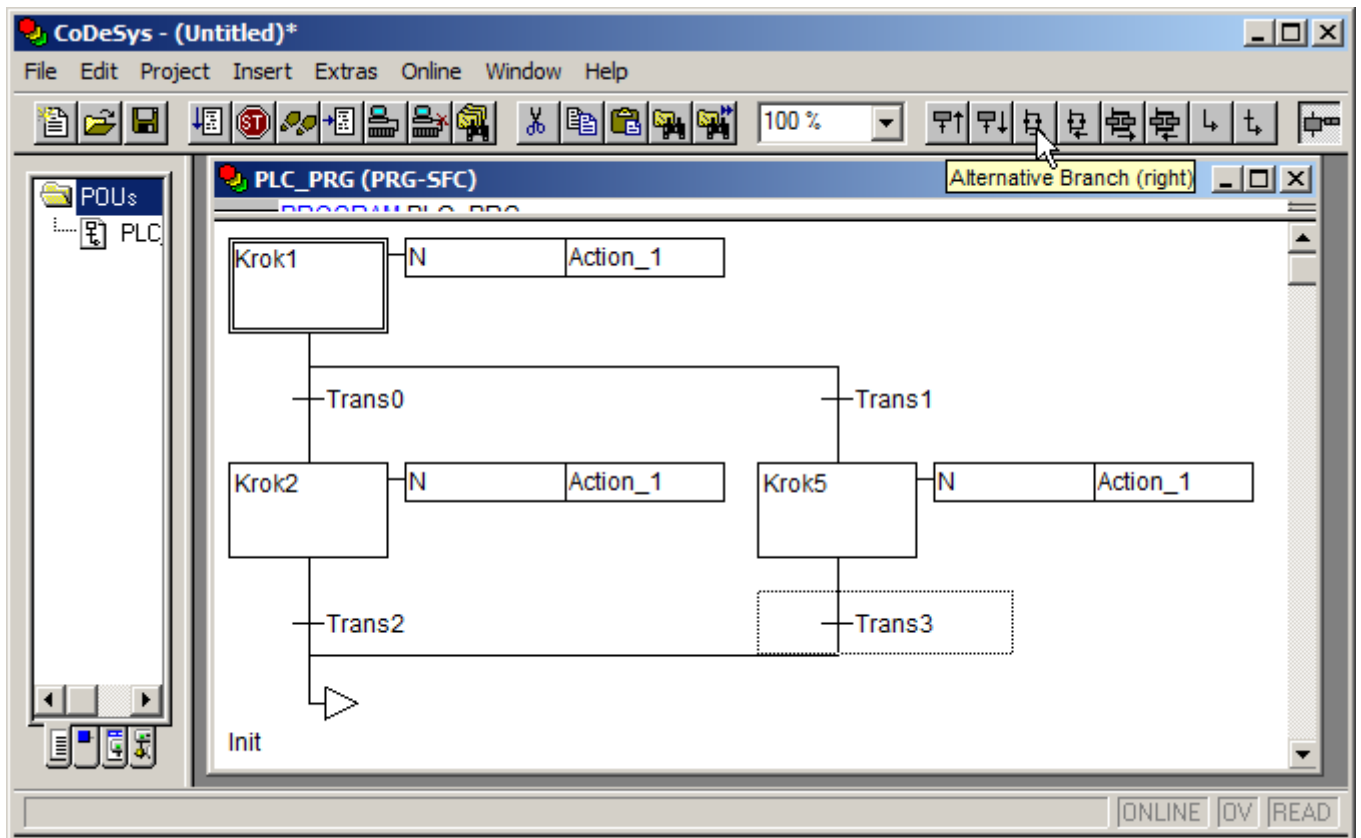
W rozważanym programie, w każdym przypadku, po kroku występują dwie alternatywne gałęzie. Na początek zostanie dodana gałąź po kroku pierwszym.



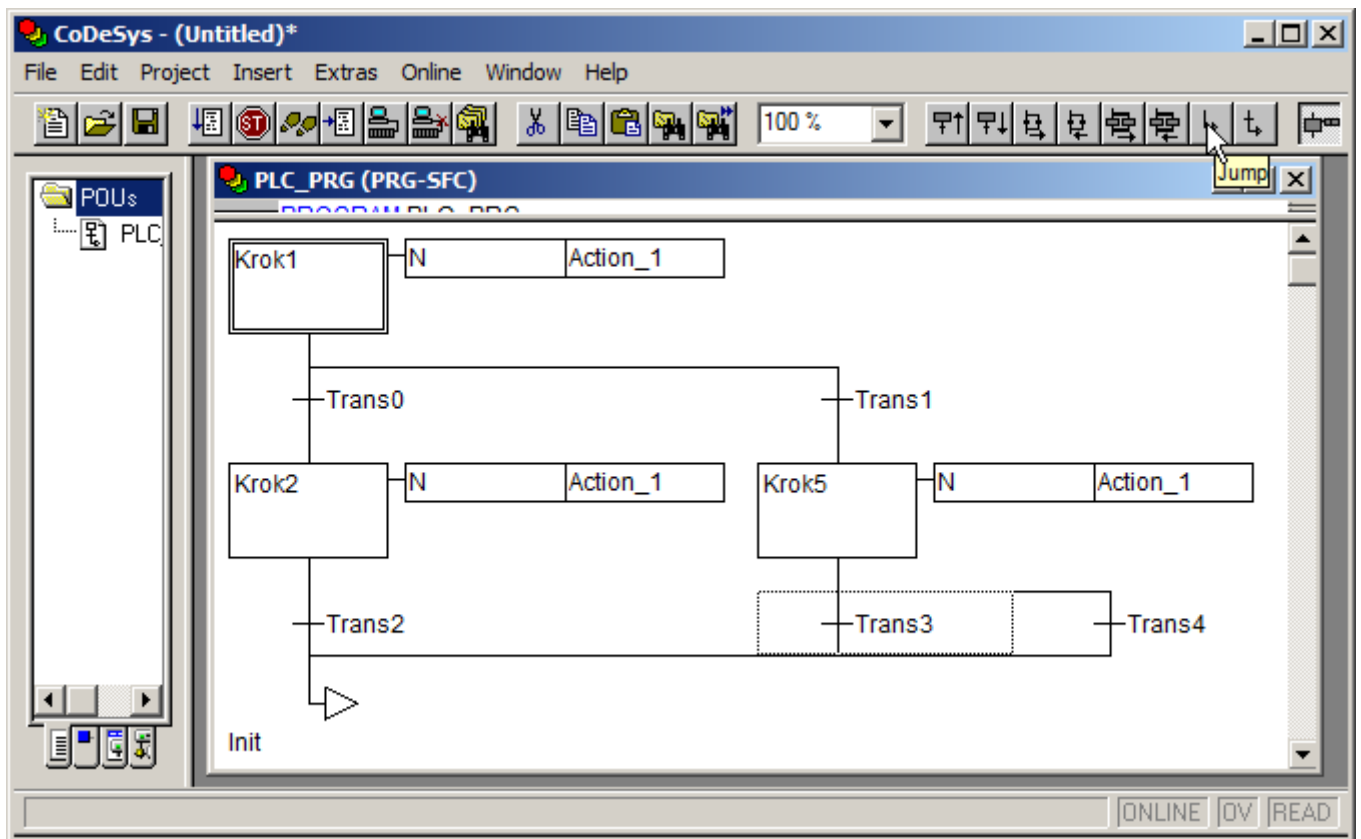
Po dodaniu gałęzi alternatywnej można uzupełnić program o Krok2 i Krok5, które są aktywowane po spełnieniu warunków tradycji w gałęziach alternatywnych.

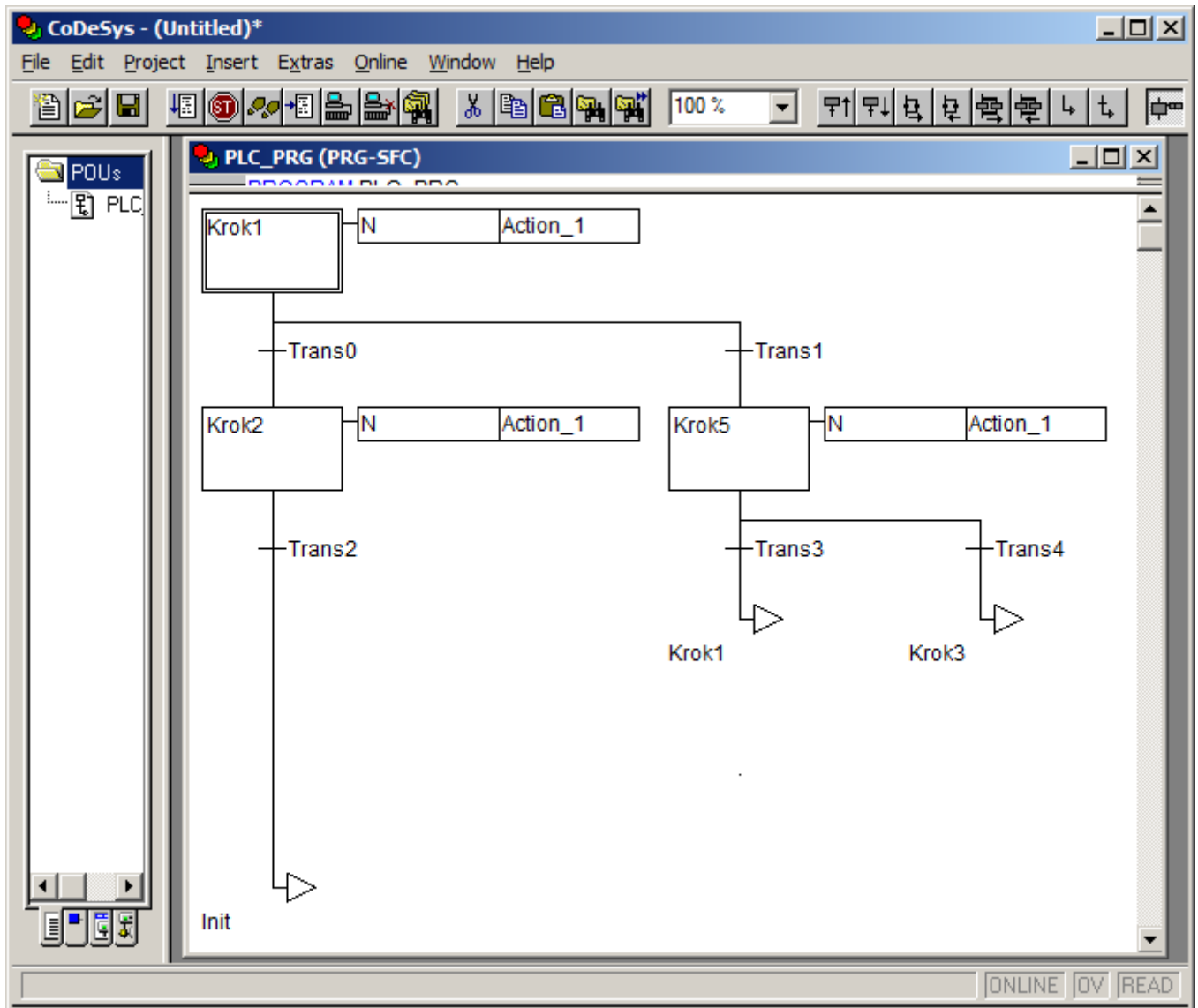


Po dodaniu kroków drugiego i piątego można uzupełnić graf programu o gałęzie alternatywne po kroku piątym.



Po spełnieniu warunków po tranzycjach w kroku piątym następują skoki do kroków pierwszego i trzeciego.



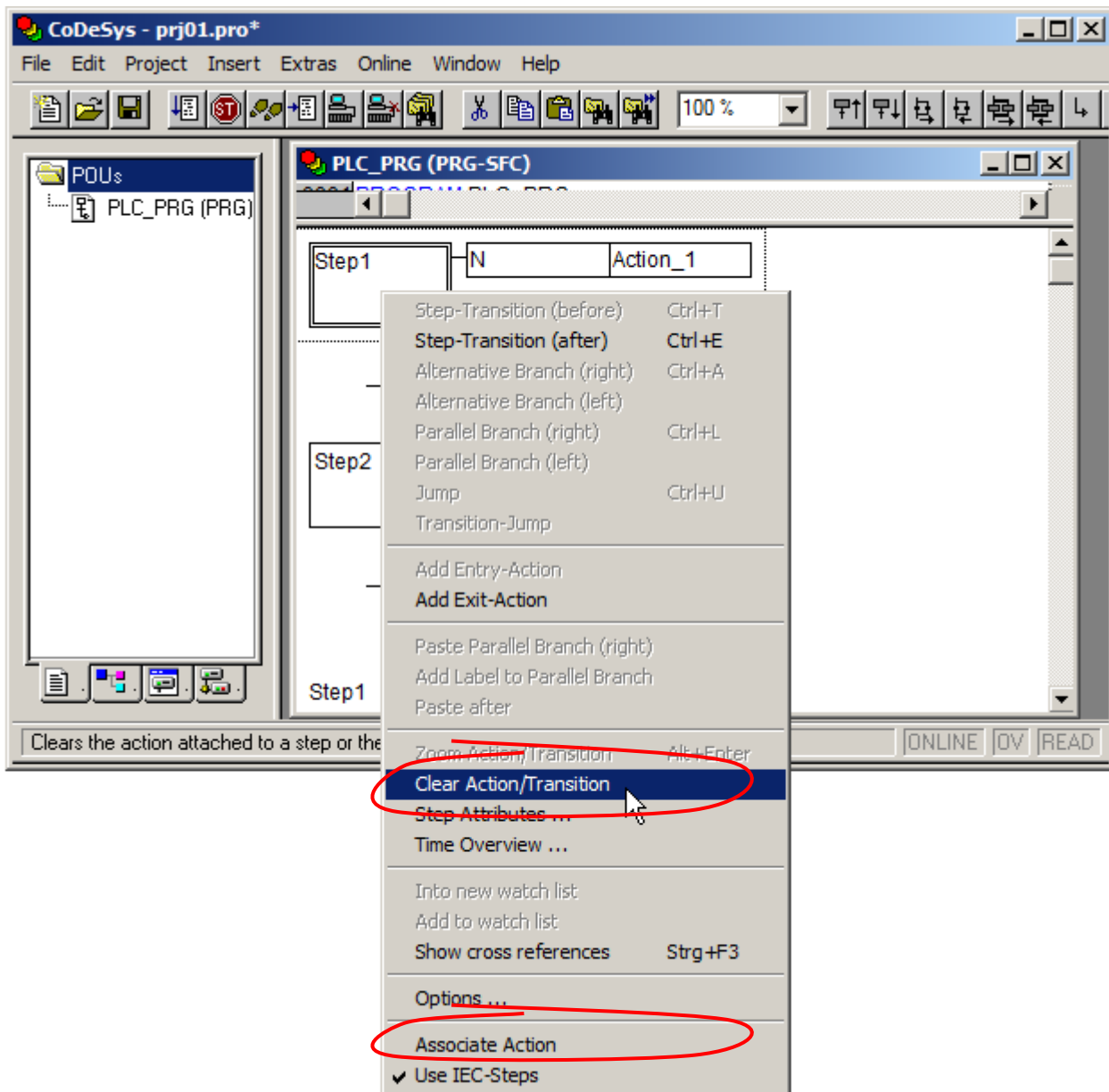


Po dodaniu rozgałęzienia za krokiem drugim, dodaniu kroku trzeciego i rozgałęzienia za krokiem trzecim, następnie kroku czwartego i rozgałęzienia za czwartym krokiem wystarczy uzupełnić już tylko graf programu o trzy symbole skoku do kroku.

Po narysowaniu grafu należy jeszcze wykorzystując *rys. 12.3.* uzupełnić:

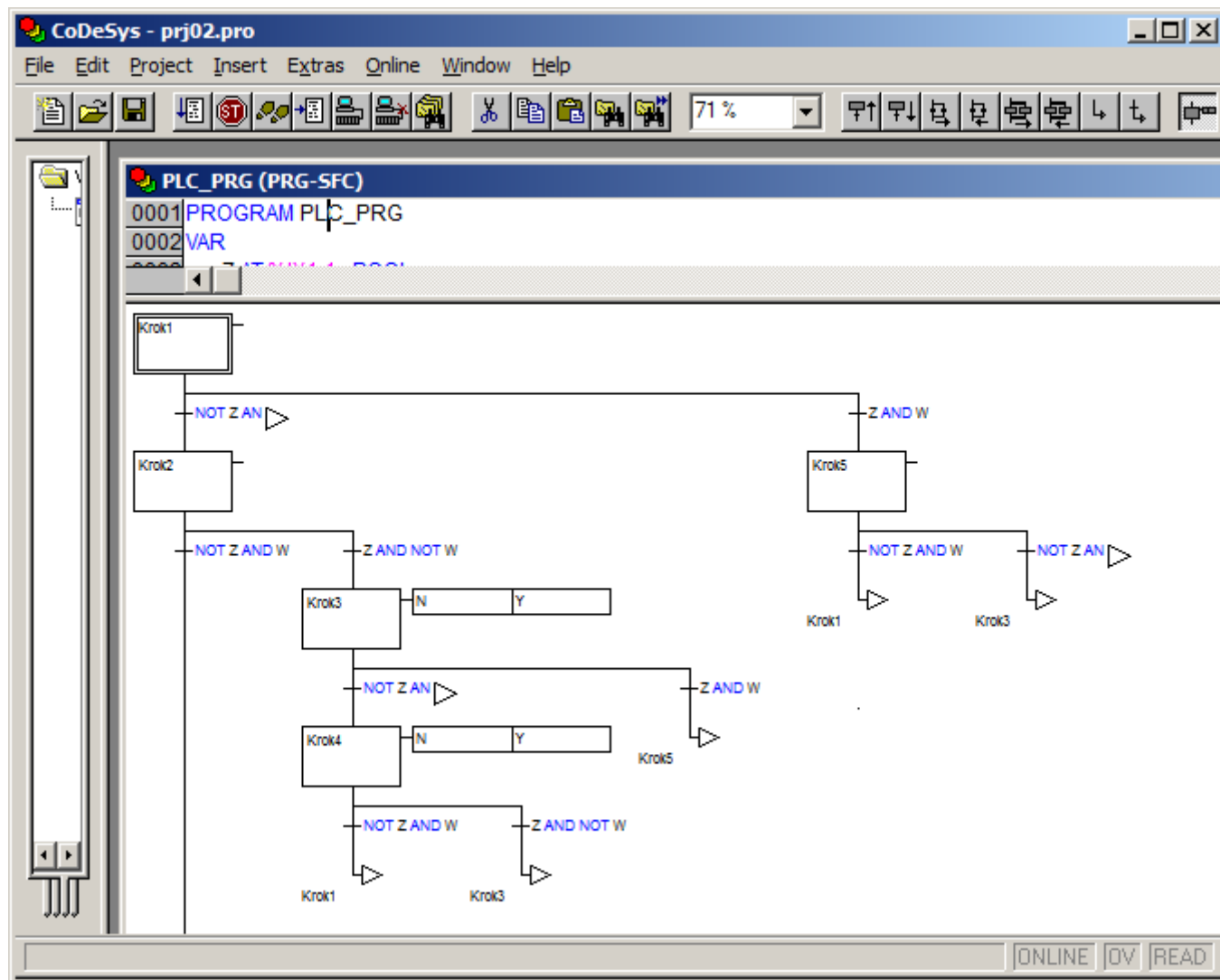
- warunki przejść tranzycji,
- skojarzyć zmienną Y z krokami trzecim i czwartym,
- usunąć akcje domyślnie kojarzone z krokami pierwszym, drugim i piątym (z krokami nie muszą być kojarzone zmienne – skojarzenie zmiennej Y z kwalifikatorem N w kroku nr 3 i nr 4 nadaje zmiennej Y w tych krokach wartość 1 a w pozostałych wartość 0).

Kroki IEC mogą być w CodeSys kojarzone z maksymalnie dziewięcioma akcjami lub zmiennymi. Usunięcie skojarzenia umożliwia opcja: „Clear Action/Transition”, dodanie skojarzenia opcja: „Associate Action”. Operacje te są przedstawione na kolejnym rysunku.

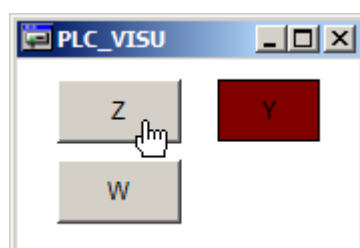


Gotowy program został przedstawiony na kolejnym rysunku.





W celu ułatwienia jego testowania została jeszcze przygotowana wizualizacja:



Po uruchomieniu, aktywny krok jest w oknie edytora programu wyróżniony na niebiesko.

