

PIERWSZY PROGRAM W JĘZYKU LD

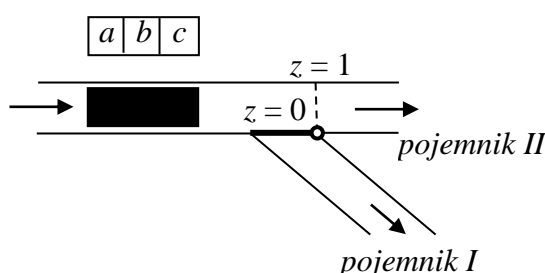
Materiał ten stanowi skróconą wersję opracowania omawiającego pracę w środowisku CoDeSys (plik CoDeSys na stronie przedmiotu). Poniżej przedstawiona została instrukcja przygotowania programu dla zadania sterowania zwrotnicą omówionego w punkcie 1.

1. Zadanie

Układ powinien sterować zwrotnicą z urządzenia sortującego kierującego produkowane detale do jednego z dwóch pojemników. Przed przesunięciem detalu do odpowiedniego pojemnika, badane są przy pomocy odpowiednich czujników trzy cechy (a , b , c) każdego z nich.

Każdy z czujników sygnalizuje zbadaną przez siebie cechę jedną z dwóch wartości (1 – wartość prawidłowa, 0 – wartość nieprawidłowa). Sygnały z czujników monitorujących cechy a , b i c odbierane są kolejno na pierwszym, drugim i trzecim wejściu modułu wejść cyfrowych, który jest włączony do gniazda sterownika o numerze 1.

Detale o co najmniej dwóch prawidłowych cechach, przy dodatkowym założeniu że cecha a jest prawidłowa, powinny być kierowane do *pojemnika I* ($z = 1$), pozostałe detale do *pojemnika II* ($z = 0$). Sygnałem sterującym położeniem zwrotnicy jest sygnał z pierwszego wyjścia modułu wyjść cyfrowych, który jest włączony do gniazda o numerze 2.



Wartość sygnału sterującego dla zwrotnicy można opisać na przykład funkcją logiczną w postaci kanonicznej dysjunkcyjnej:

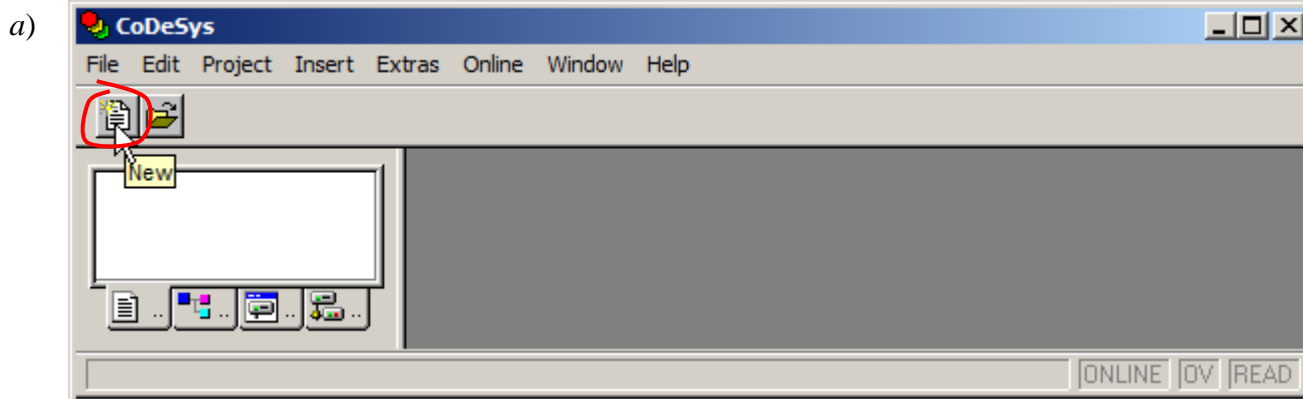
$$z = abc + a\bar{b}\bar{c} + \bar{a}\bar{b}c,$$

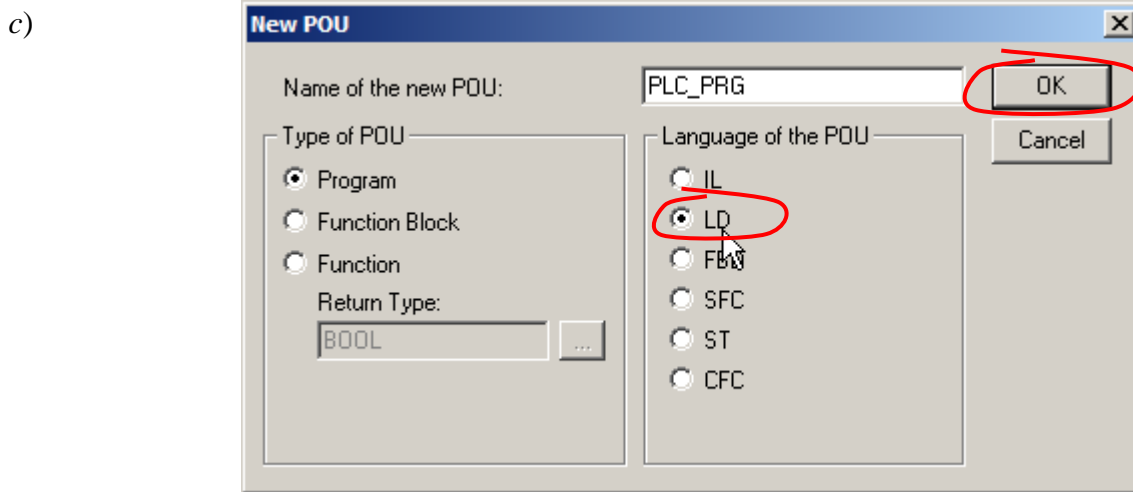
lub w postaci koniunkcyjnej:

$$z = (a + b) \cdot (a + \bar{b}) \cdot (b + c).$$

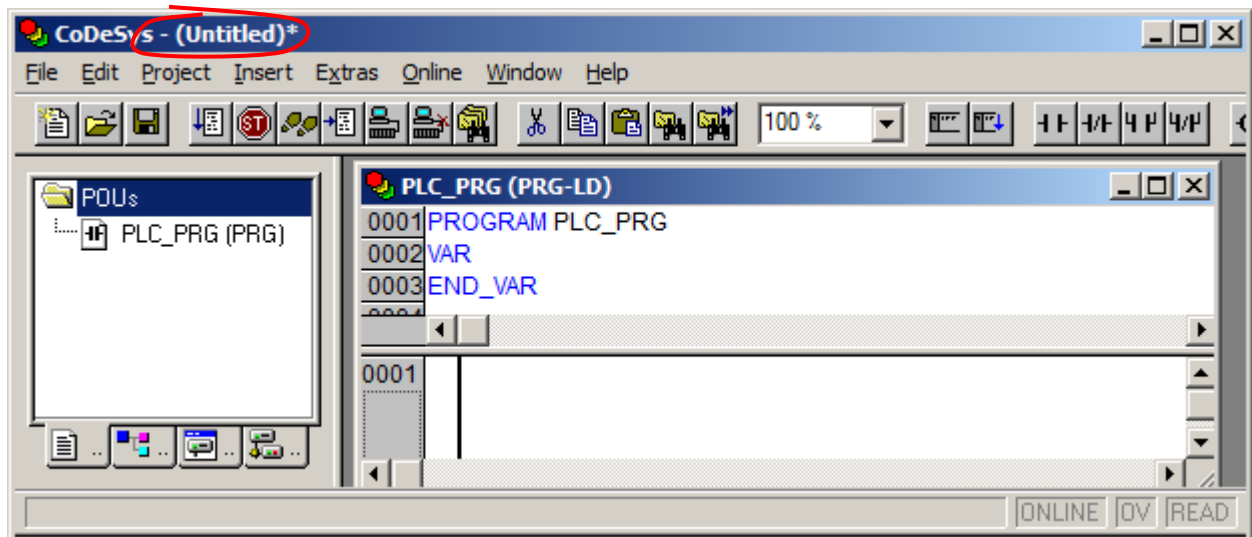
2. Tworzenie nowego projektu

Przygotowanie projektu z programem głównym w języku LD wymaga wykonania kroków przedstawionych na kolejnych rysunkach ([szczegółowe omówienie zagadnienia można znaleźć w materiale CoDeSys zamieszczonym na stronie przedmiotu](#)):

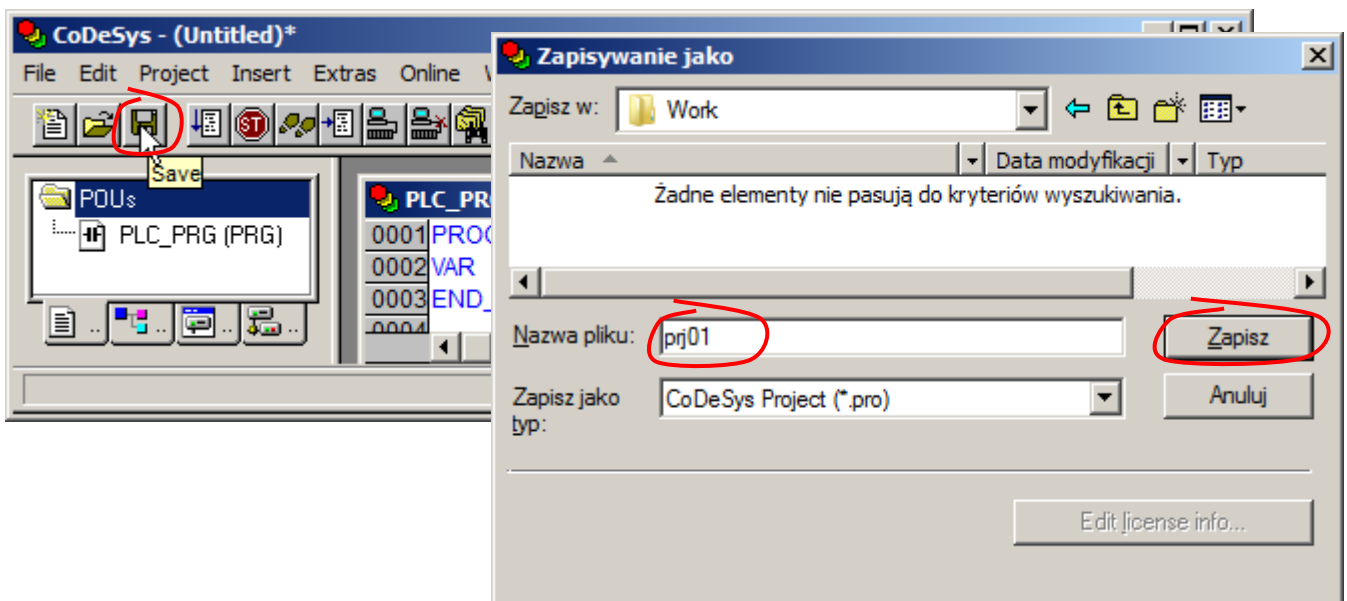




Po utworzeniu, nowy projekt otrzymuje nazwę **Untitled** i jest automatycznie otwierany w środowisku.



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



3. Tworzenie programu

3.1. Zasady ogólne

Język *schematów drabinkowych* LD jest odpowiednikiem schematów (obwodowych, drabinkowych) układów cyfrowych realizowanych w technologii stykowo – przekaźnikowej, dodatkowo język pozwala na wykorzystywanie funkcji realizujących operacje arytmetyczne, logiczne, porównania oraz bloków funkcjonalnych realizujących zadania przerzutników, timerów, liczników.

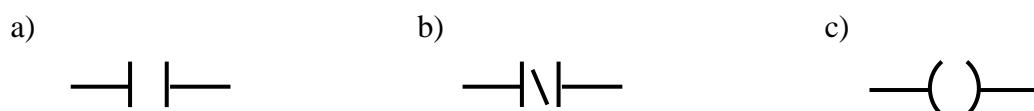
Schemat drabinkowy jest ograniczony z lewej i prawej strony „szynami prądowymi”:



Szyny prądowe a) lewa, b) prawa.

Elementy położone pomiędzy lewą i prawą szyną tworzą tzw. „szczebel”. Pojedynczy szczebel odpowiada odrębnemu obwodowi prądowemu, w którym przepływ prądu zależy od stanu elementów tego obwodu. Upraszczając można stwierdzić, że każdy ze szczebli schematu definiuje jedną instrukcję programu.

Podstawowymi elementami używanymi podczas definiowania szczebla są: *zestyki zwierne* i *rozwiernie* oraz *cewki*. *Zestyki* łączone są w programie ze zmiennymi wejściowymi, wyjściowymi lub wewnętrznymi, *cewki* mogą być łączone tylko ze zmiennymi wyjściowymi i wewnętrznymi. Symbol zmiennej skojarzonej z określonym elementem pokazywany jest nad symbolem elementu.



a) Zestyk zwierny, b) zestyk rozwierny, c) cewka.

Zestyk zwierny (normalnie otwarty) jest elementem, który pozwala na przepływ prądu (z jego lewej strony na prawą) jeżeli skojarzona z nim zmienna ma wartość *prawda*, jeżeli zmienna ma wartość *fałsz* element nie przewodzi prądu.

Zestyk rozwierny (normalnie zamknięty) pozwala na przepływ prądu jeżeli skojarzona z nim zmienna ma wartość *fałsz*.

Cewka jest elementem wykorzystywanym do nadawania wartości zmiennym wyjściowym lub wewnętrznym, jeżeli linia z lewej strony cewki jest „pod prądem” to skojarzona z cewką zmienna otrzymuje wartość *prawda*, w przeciwnym przypadku wartość *fałsz*.

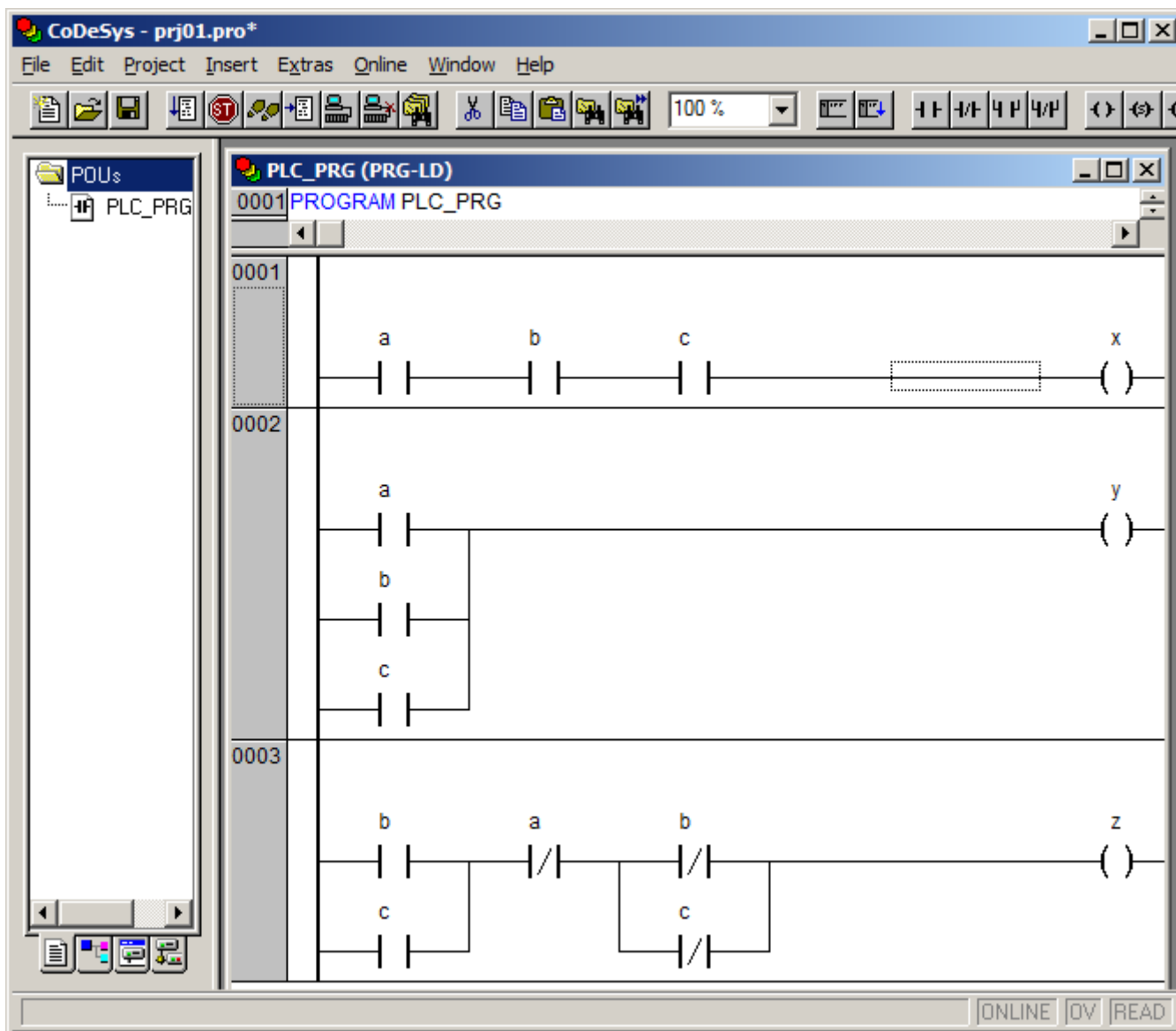
Typowa definicja prostego szczebla, zaczynając od jego lewej strony, składa się z:

- symbolu lewej szyny prądowej,
- połączonych na różne sposoby symboli zestyków odpowiadających wybranemu wyrażeniu logicznemu,
- symbolu cewki, cewka nadaje wartość określonej zmiennej przypisując jej wartość wynikającą z wartości wyrażenia logicznego wynikającego z postaci szczebla,
- symbolu prawej szyny prądowej.

Przy budowie wyrażenia logicznego z symboli zestyków należy kierować się zasadami:

- **połączenie szeregowe** zestyków **odpowiada koniunkcji** zmiennych skojarzonych z tymi zestykami,
- **połączenie równoległe** zestyków **odpowiada alternatywie** skojarzonych z nimi zmiennych,
- zestykowi rozwiernemu odpowiada negacja połączonej z nim zmiennej.

3.2. Przykład



Program, którego schemat przedstawiony został na powyższym rysunku budowany jest z trzech instrukcji (na schemacie widoczne są trzy szczeble). Instrukcje te przypisują wartości kolejno zmiennym: x , y i z (przed symbolem prawej szyny prądowej na każdym ze szczebli umieszczony jest symbol cewki, na pierwszym szczeblu cewka skojarzona jest ze zmienną x , na drugim ze zmienną y , ...).

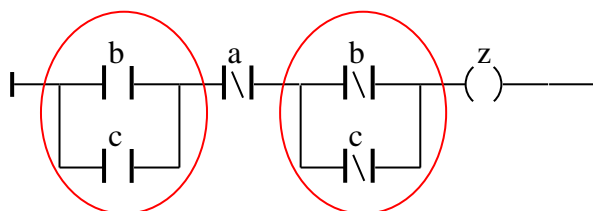
Zestyki szczebla pierwszego połączone są *szeregowo*, więc zmienna x otrzymuje wartość wynikającą z wartości wyrażenia:

$$a \cdot b \cdot c.$$

Zestyki szczebla drugiego połączone są *równolegle*, więc zmienna y otrzymuje wartość:

$$a + b + c.$$

Budowa trzeciego szczebla jest bardziej złożona. Na poniższym rysunku szczebel ten został przerysowany z właściwego schematu.



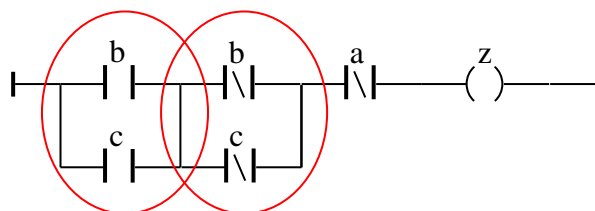
Czerwonymi elipsami zaznaczone zostały *równoległe* połączenia zestyków zwiernych b i c (pierwsza elipsa) i *równoległe* połączenia zestyków rozwiernych b i c (druga elipsa). Wyodrębnione fragmenty szczebla tworzą połączenie szeregowe z zestykiem rozwiernym a . Zmienna z otrzymuje więc wartość wynikającą z wartości wyrażenia:

$$(b + c) \cdot \bar{a} \cdot (\bar{b} + \bar{c}).$$

Wyrażenie to jest równoważne np. wyrażeniu:

$$(b + c) \cdot (\bar{b} + \bar{c}) \cdot \bar{a}.$$

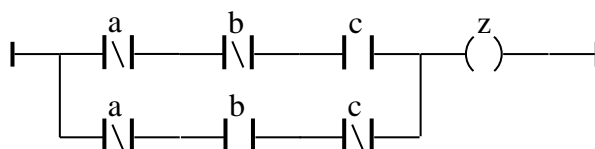
Na poniższym rysunku przedstawiony został szczebel nadający wartość zmiennej z na podstawie nowej postaci wyrażenia:



Wykorzystując prawa algebry Boole'a powyższe wyrażenie można również przekształcić do postaci:

$$\bar{a}\bar{b}c + \bar{a}b\bar{c}.$$

Definicja pierwszego składnika wyrażenia sprowadza się w języku LD do szeregowego połączenia trzech zestyków skojarzonych ze zmiennymi a , b i c – zestyki a i b muszą być zestykami rozwiernymi, a zestyk c musi być zestykiem zwiernym. Podobnie wygląda definicja drugiego składnika wyrażenia. Składnik ten odpowiada szeregowemu połączeniu zestyków: rozwiernego a , zwiernego b i rozwiernego c . Sumowanie składników sprowadza się do *równoległego* połączenia odpowiednich fragmentów szczebla. Wartość zmiennej z można więc również wyznaczyć za pomocą szczebla:

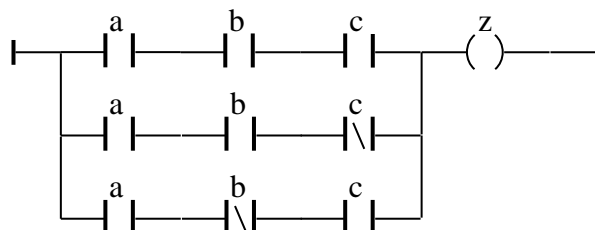


3.3. Zadanie

W punkcie 1. zaproponowane zostały dwie równoważne funkcje opisujące wartość sygnału sterującego dla zwrotnicy. Pierwsza postać:

$$z = abc + \bar{a}b\bar{c} + \bar{a}\bar{b}c,$$

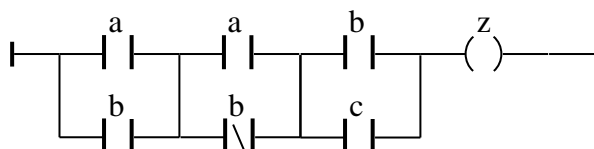
jest sumą trzech iloczynów, w języku LD należy ją więc narysować jako:



Druga postać:

$$z = (a + b) \cdot (a + \bar{b}) \cdot (b + c),$$

to iloczyn trzech sum, w języku LD odpowiada jej następujący schemat:

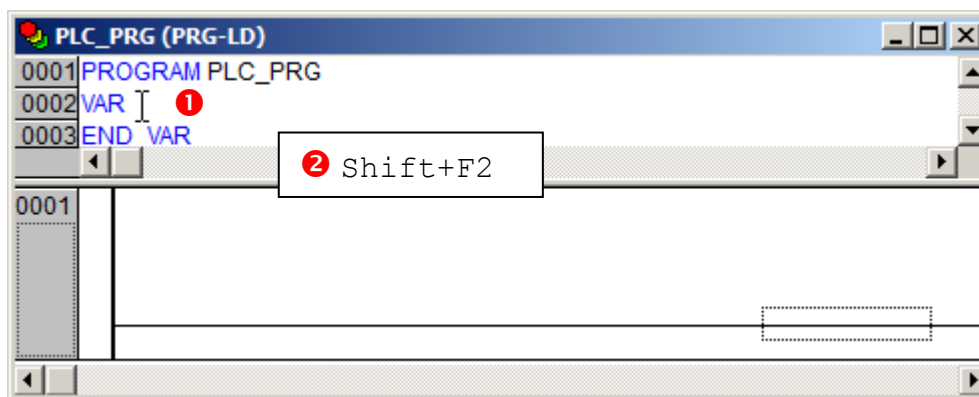


Program realizujący przykładowe zadanie można wprowadzić w CoDeSys na wiele różnych sposobów (szczegółowe omówienie zagadnienia można znaleźć w materiale CoDeSys – plik na stronie przedmiotu). W omówionym poniżej podejściu najpierw zdefiniowane zostaną zmienne, później krok po kroku zostanie narysowany diagram LD dla pierwszej i drugiej postaci funkcji

Deklaracja zmiennych

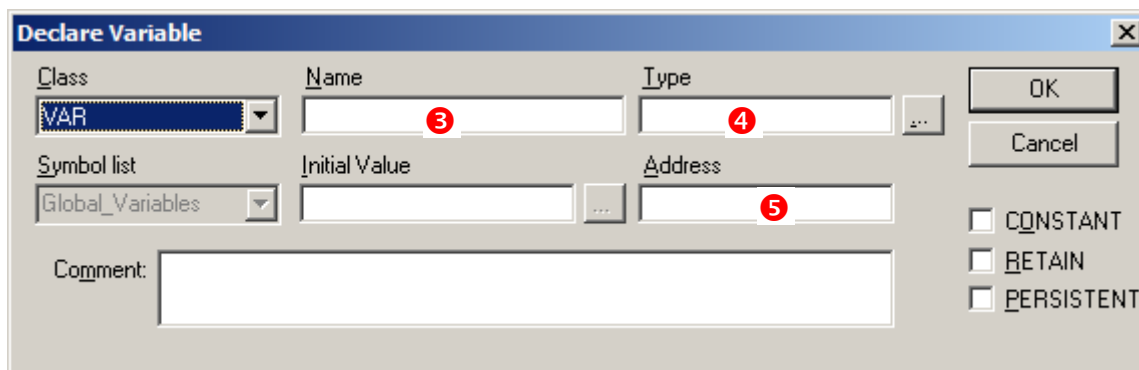
W celu zdefiniowania zmiennych:


- 1) ustawić kursor w górnej części edytora w linii zawierającej słowo **VAR**,
- 2) nacisnąć z klawiatury kombinację klawiszy: **Shift+F2**,

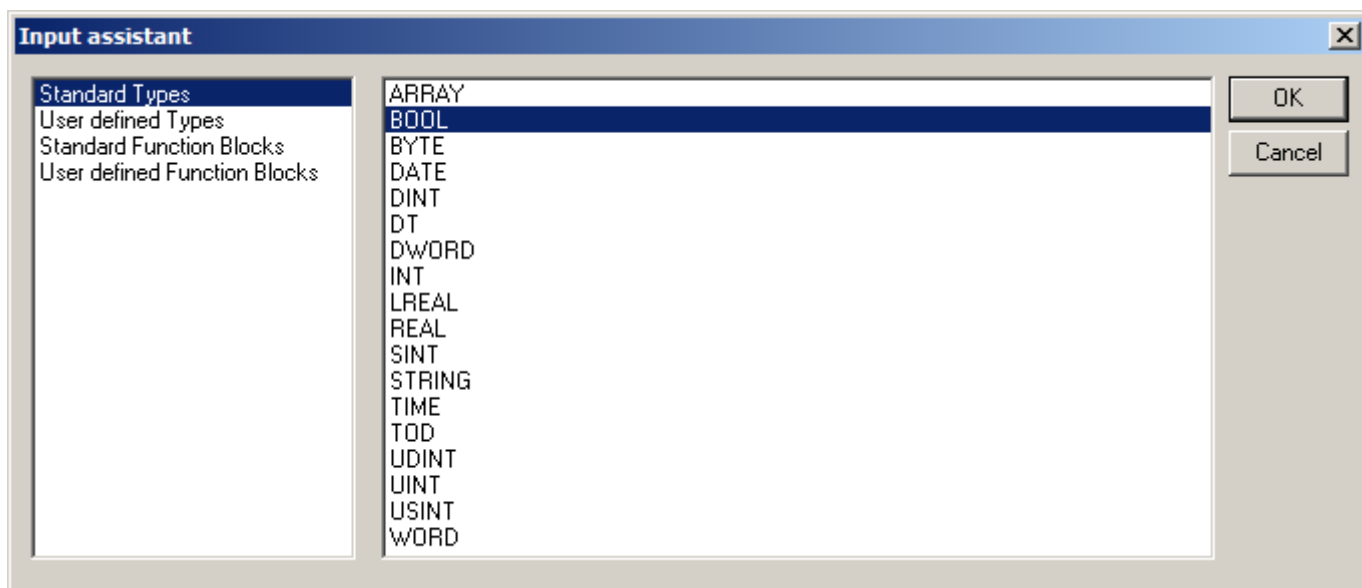


wprowadzić parametry deklarowanych zmiennych.

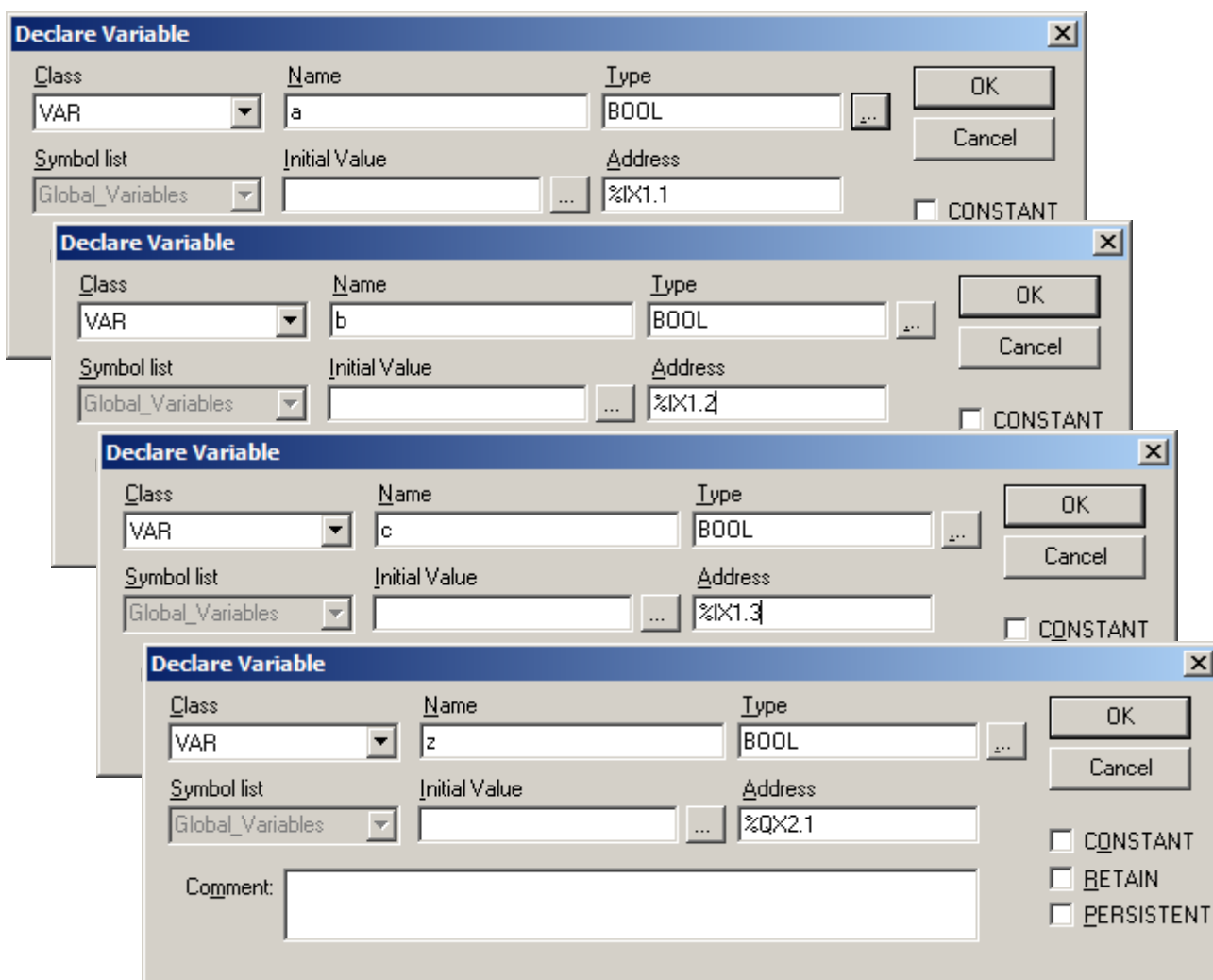
- 3) nazwę,
- 4) typ,
- 5) adres w kartach We/Wy sterownika.



Wartości parametrów można wprowadzić wypełniając odpowiednie pola tekstowe okna deklaracji zmiennej, w przypadku wypełniania typu można skorzystać z pomocniczego okna z listą dostępnych typów (dostępnego pod przyciskiem )



godnie z treścią zadania w programie należy zdefiniować 4 zmienne o rozmiarze 1 bitu – zmienne takie muszą być zadeklarowane jako zmienne boolowskie, tzn. zmienne typu BOOL. Na poniższych rysunkach pokazane zostały deklaracje zmiennych: *a*, *b*, *c* i *z*, ich adresy zostały ustawione na podstawie opisu w punkcie 1. (zmienne *a*, *b* i *c* odpowiednio: %IX1.1, %IX1.2, %IX1.3, zmienna *z*: %QX2.1).



Po wykonaniu powyższych kroków, deklaracje zmiennych widoczne są w oknie edytora.

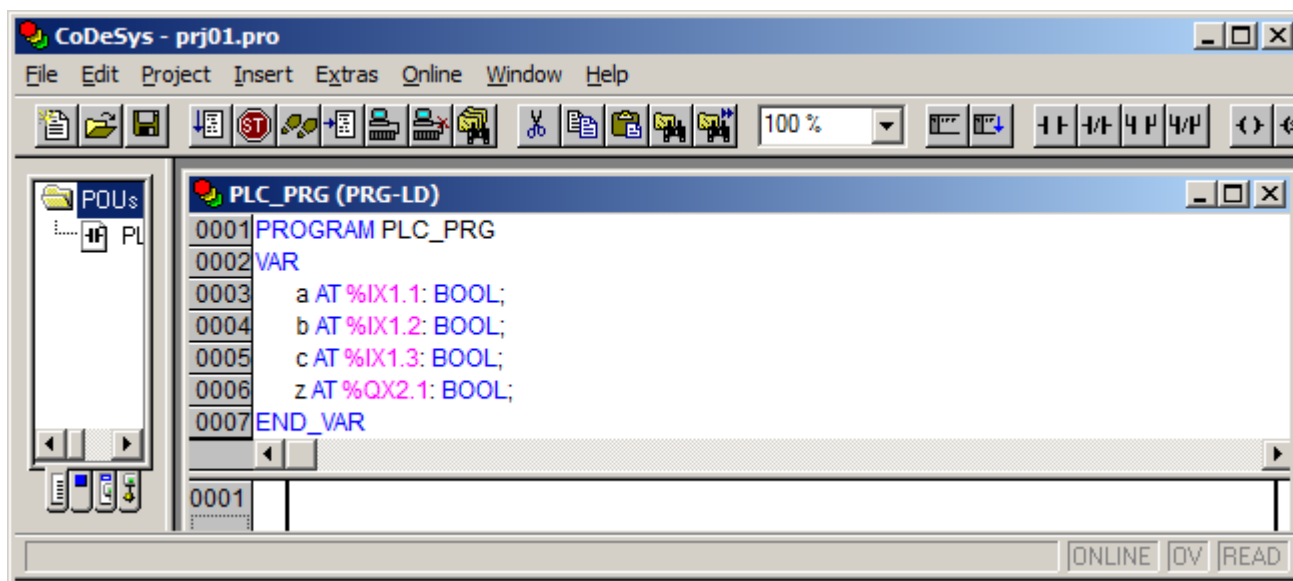


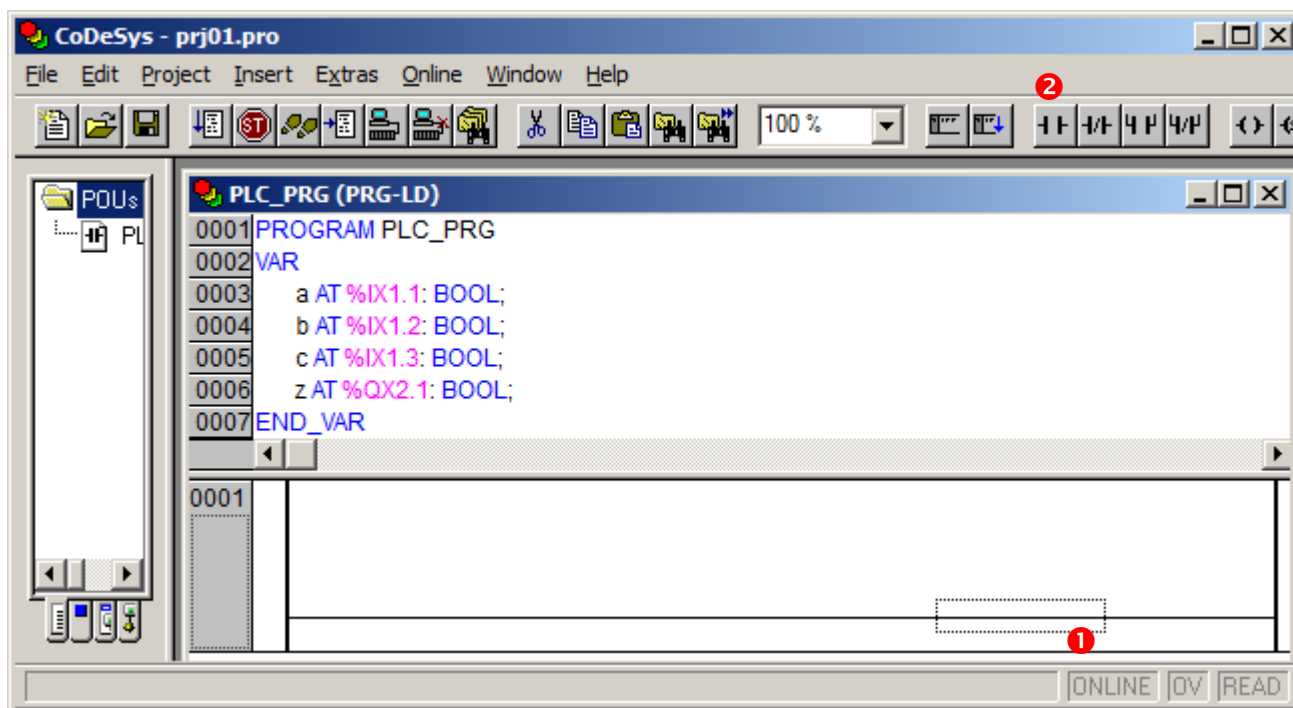
Diagram LD – funkcja 1

Narysowanie diagramu LD dla pierwszej z zaproponowanych postaci funkcji sprowadza się do narysowania trzech równoległych gałęzi z których każda zawiera po trzy szeregowo połączone zestyki. W przedstawionym poniżej sposobie postępowania:

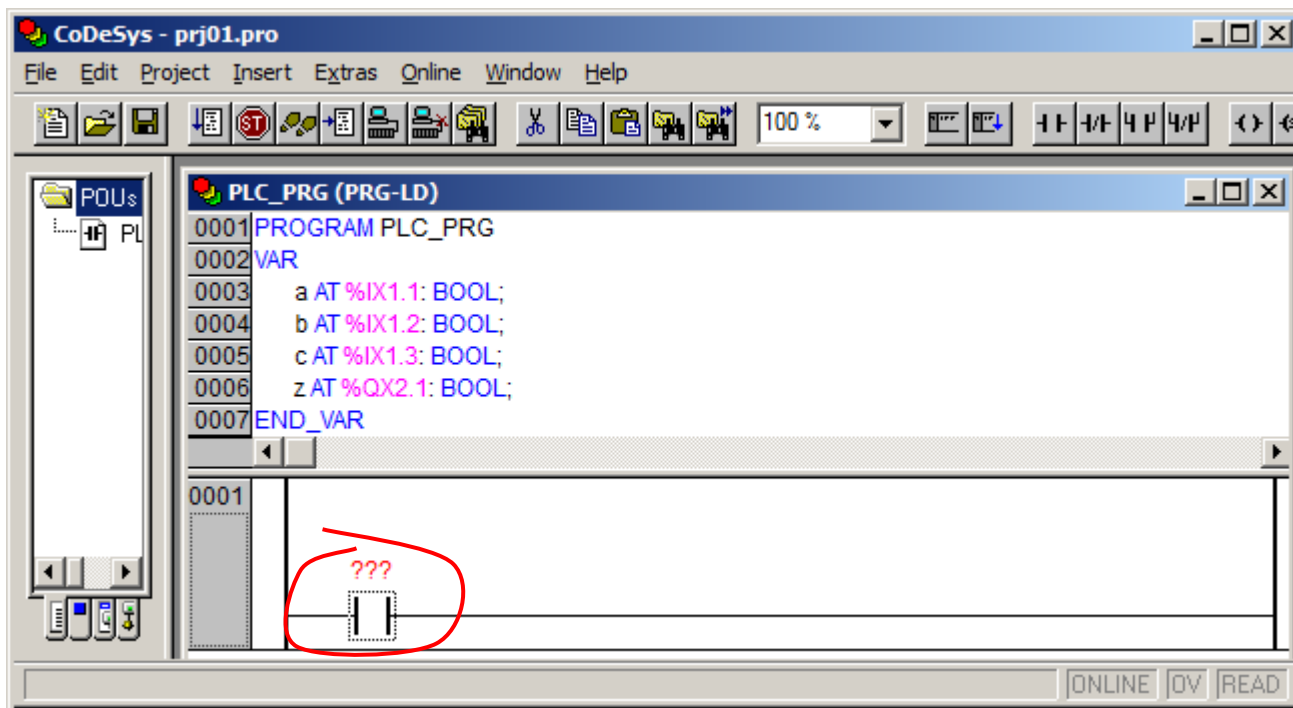
- najpierw wstawiony zostanie zestyk w pierwszej gałęzi,
- następnie zostanie utworzona gałąź równoległa zawierająca pojedynczy zestyk,
- kolejno zostanie utworzona trzecia gałąź równoległa z pojedynczym zestykiem,
- na koniec, w każdej gałęzi zostaną wstawione brakujące, połączone szeregowo, dwa zestyki.

Wstawienie pierwszego zestyku wymaga:

- 1) kliknięcia w edytorze programu na pustym polu,
- 2) wstawienia elementu.

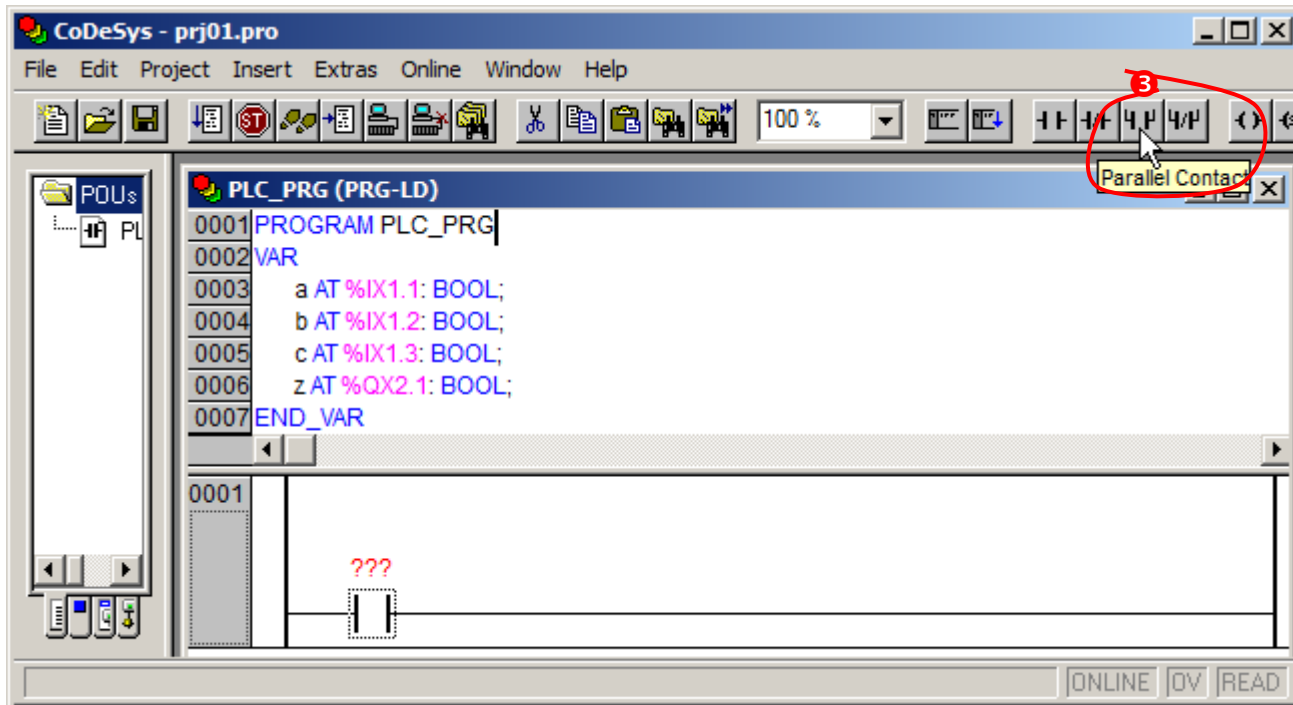


Po wstawieniu elementu jest on automatycznie zaznaczany:

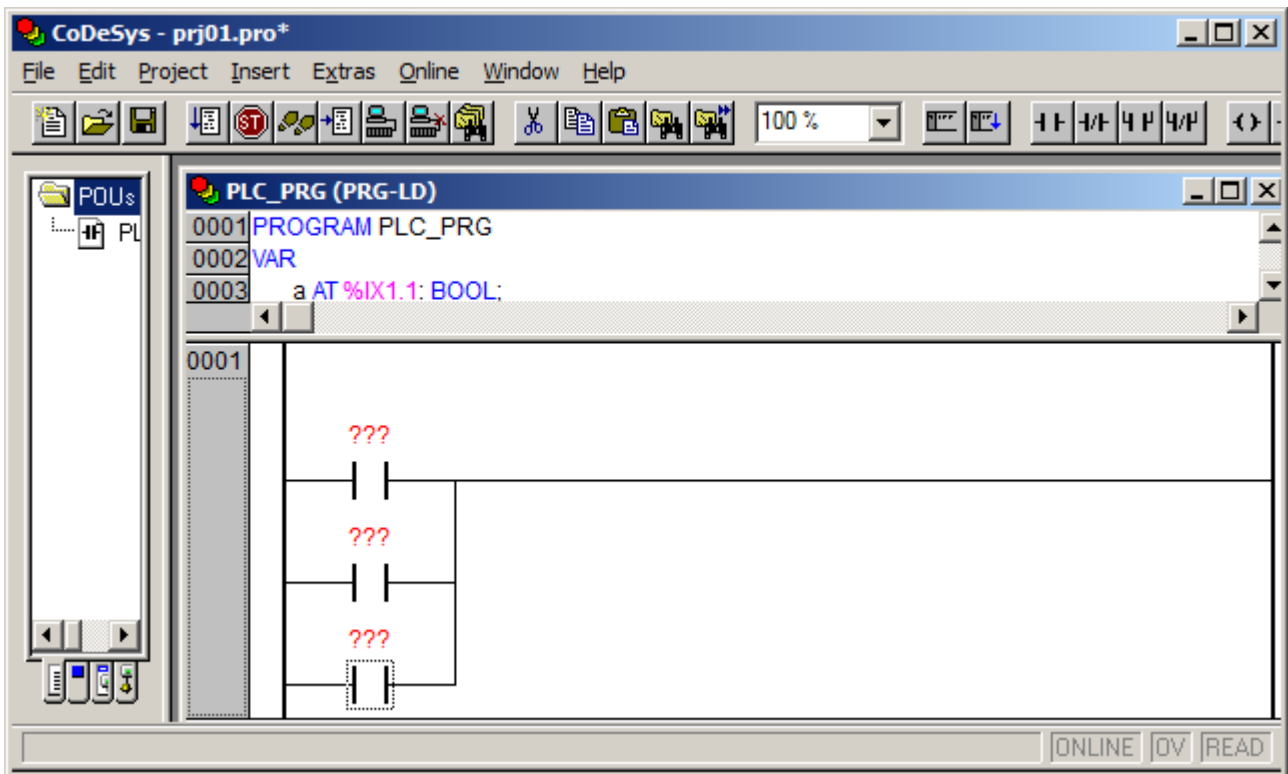


W kolejnym kroku diagram zostanie uzupełniony o zestyk połączony równoległe, w tym celu należy:

- 3) z paska narzędziowego wybrać zestyk połączony równoległe.

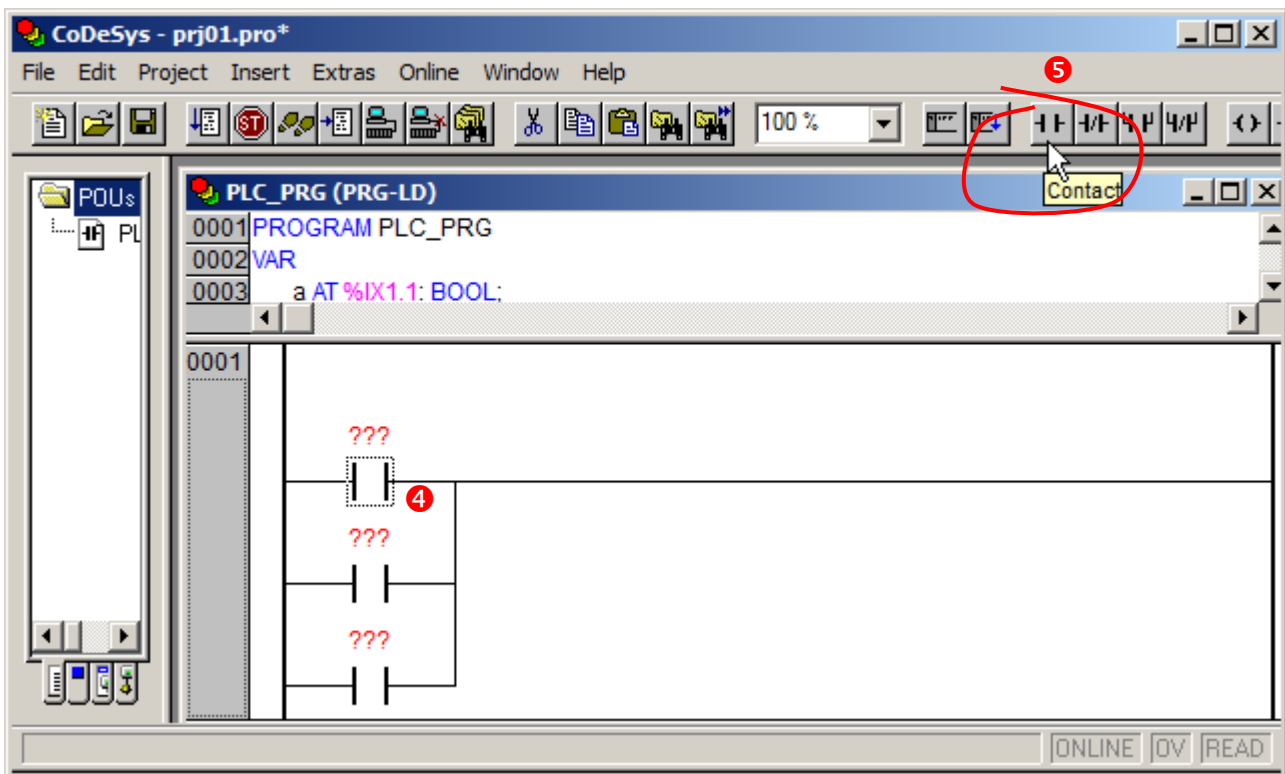


Po wstawieniu elementu jest on automatycznie zaznaczany. Gotowy diagram LD musi mieć trzy gałęzie równoległe – krok 3. należy więc powtórzyć. Stan edytora po wstawieniu trzech równoległe połączonych zestyków przedstawiony został na poniższym rysunku.

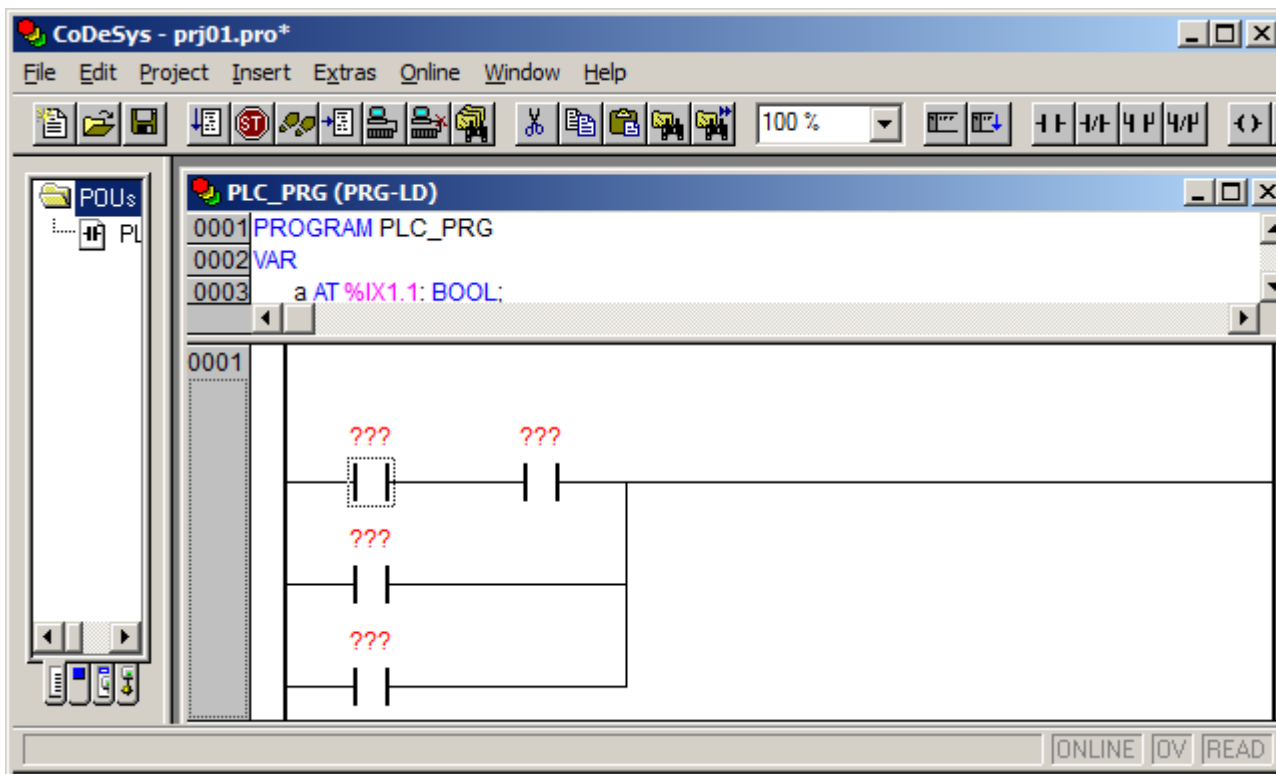


Na koniec, wszystkie gałęzie diagramu należy uzupełnić wstawiając po dwa zestyki połączone szeregowo z wcześniej dodanym elementem. W celu dodania elementu połączonego z wybranym należy:

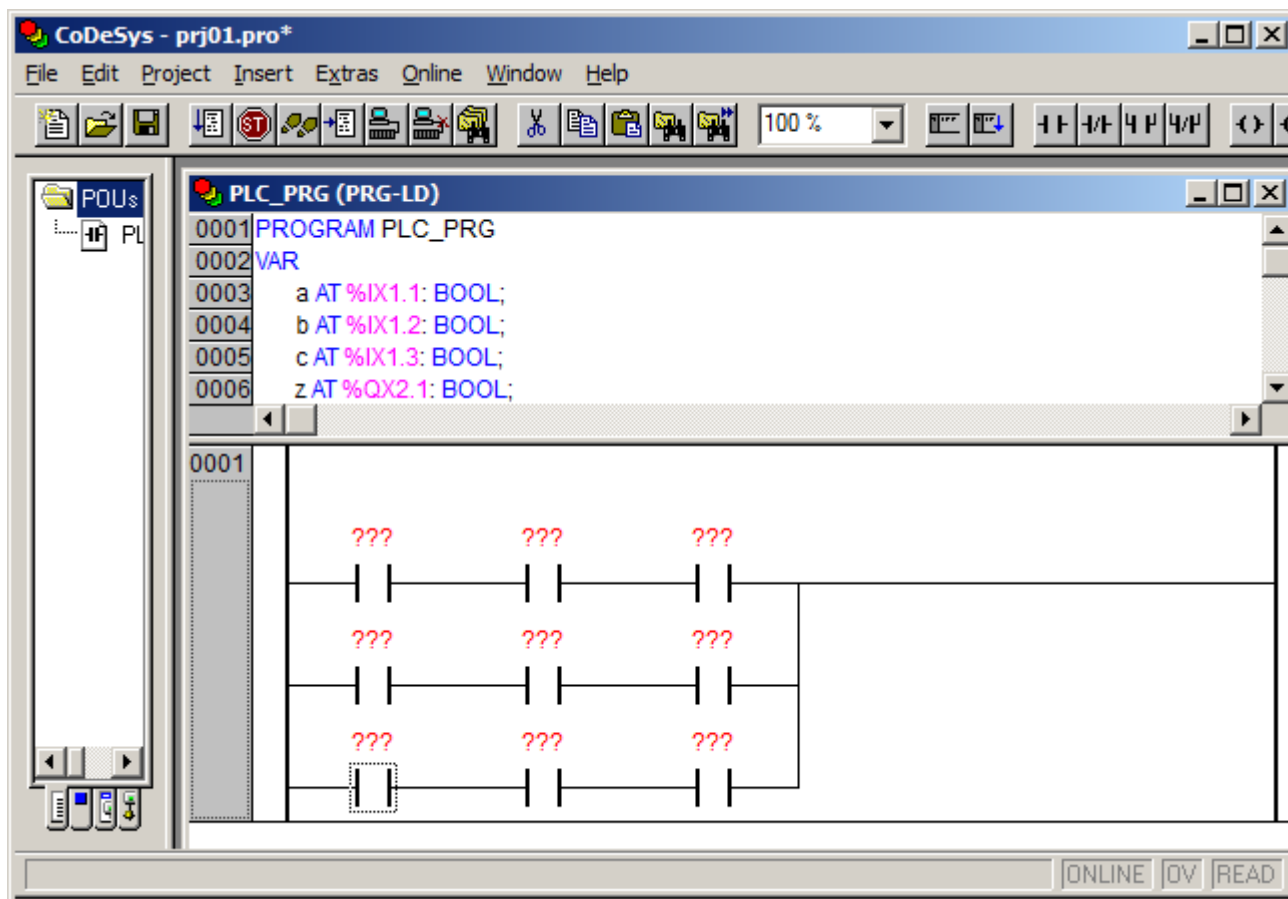
- 4) kliknąć na elemencie przed którym zostanie dodany wstawiany zestyk,
- 5) z paska narzędziowego wybrać zestyk połączony szeregowo.



Po wstawieniu elementu wszystkie gałęzie szczebla są rozszerzane, a element jest automatycznie zaznaczany:

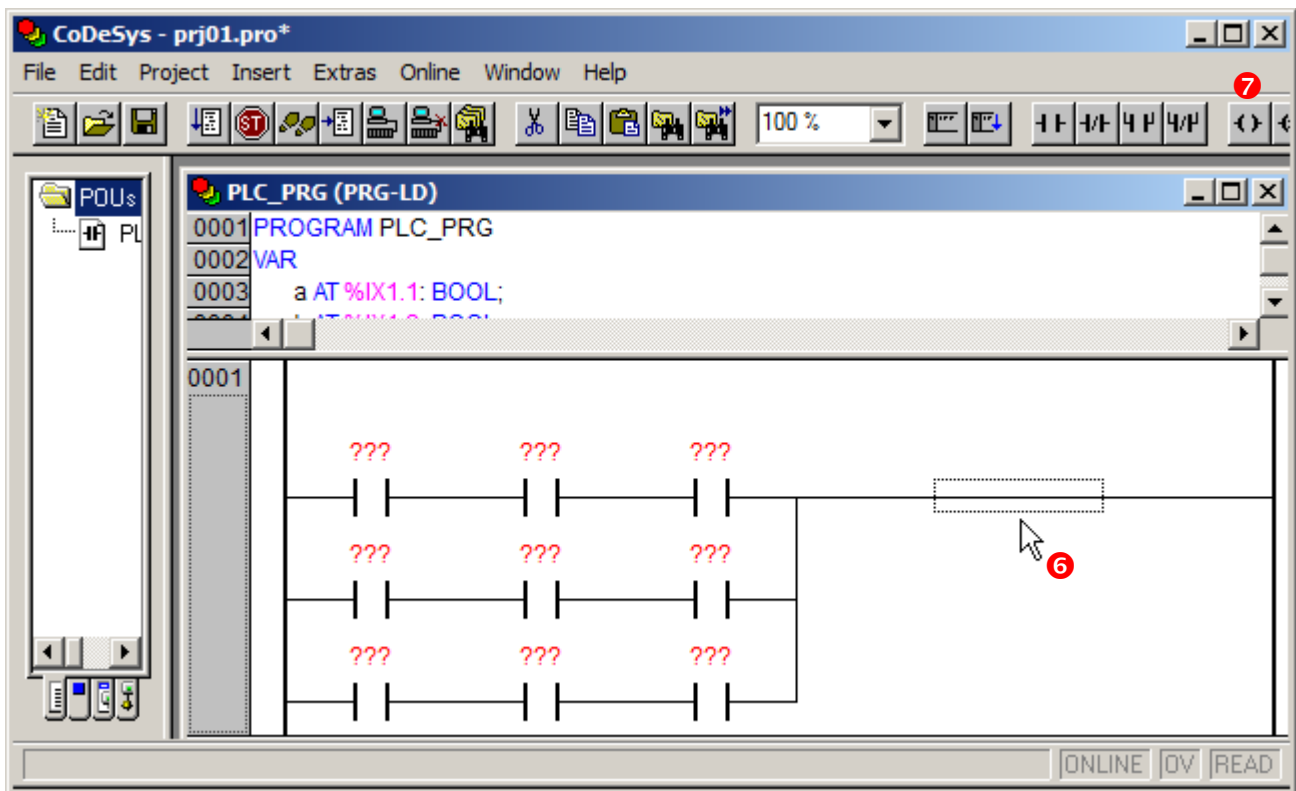


Gotowy diagram LD musi mieć po trzy szeregowo trzy gałęzie równoległe – krok 3. należy więc powtórzyć. Stan edytora po uzupełnieniu diagramu został przedstawiony na poniższym rysunku.

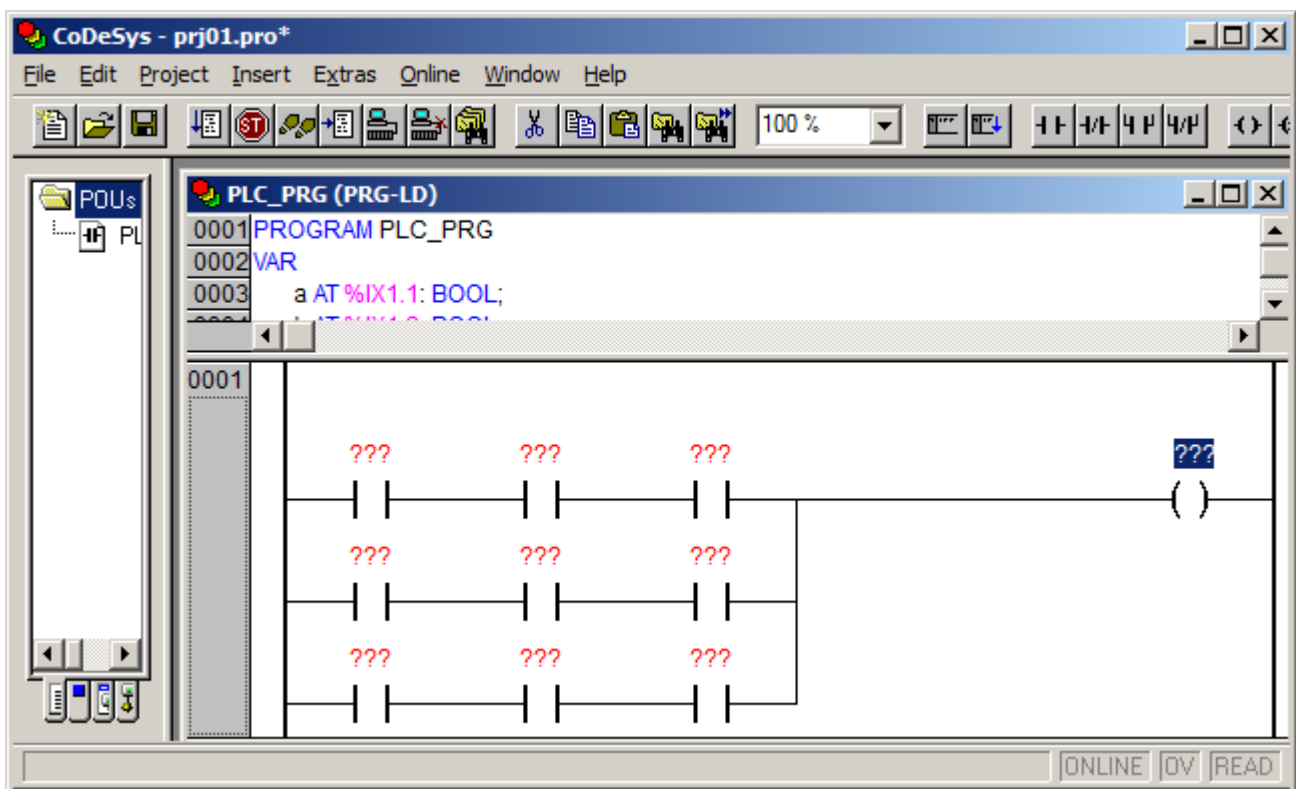


Na koniec należy jeszcze uzupełnić diagram wstawiając cewkę ustawiającą wartość zmiennej wyjściowej. W tym celu należy:

- 6) kliknąć po prawej stronie szczebla,
- 7) wstawić cewkę.

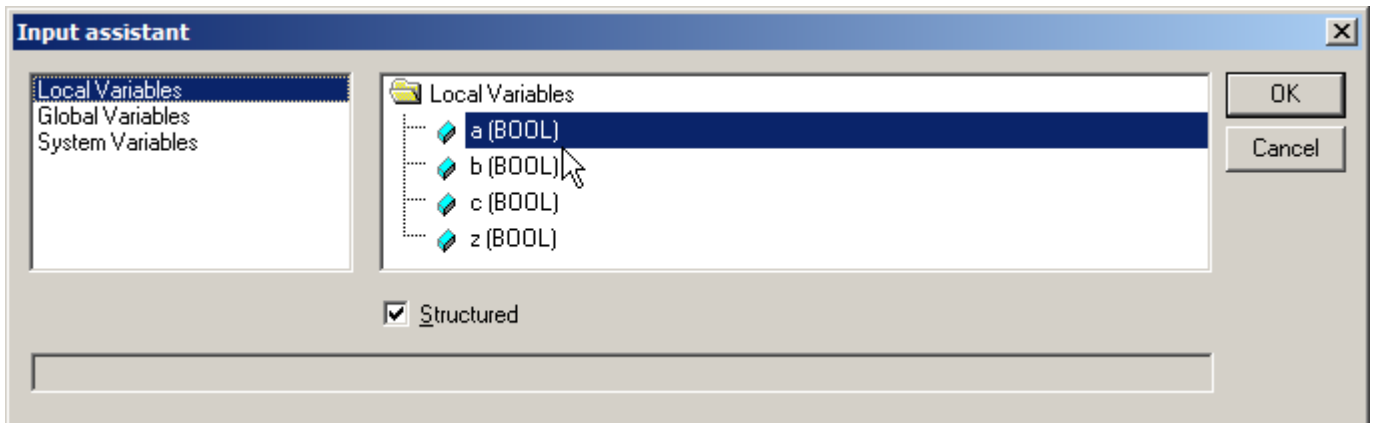


Po dodaniu cewki

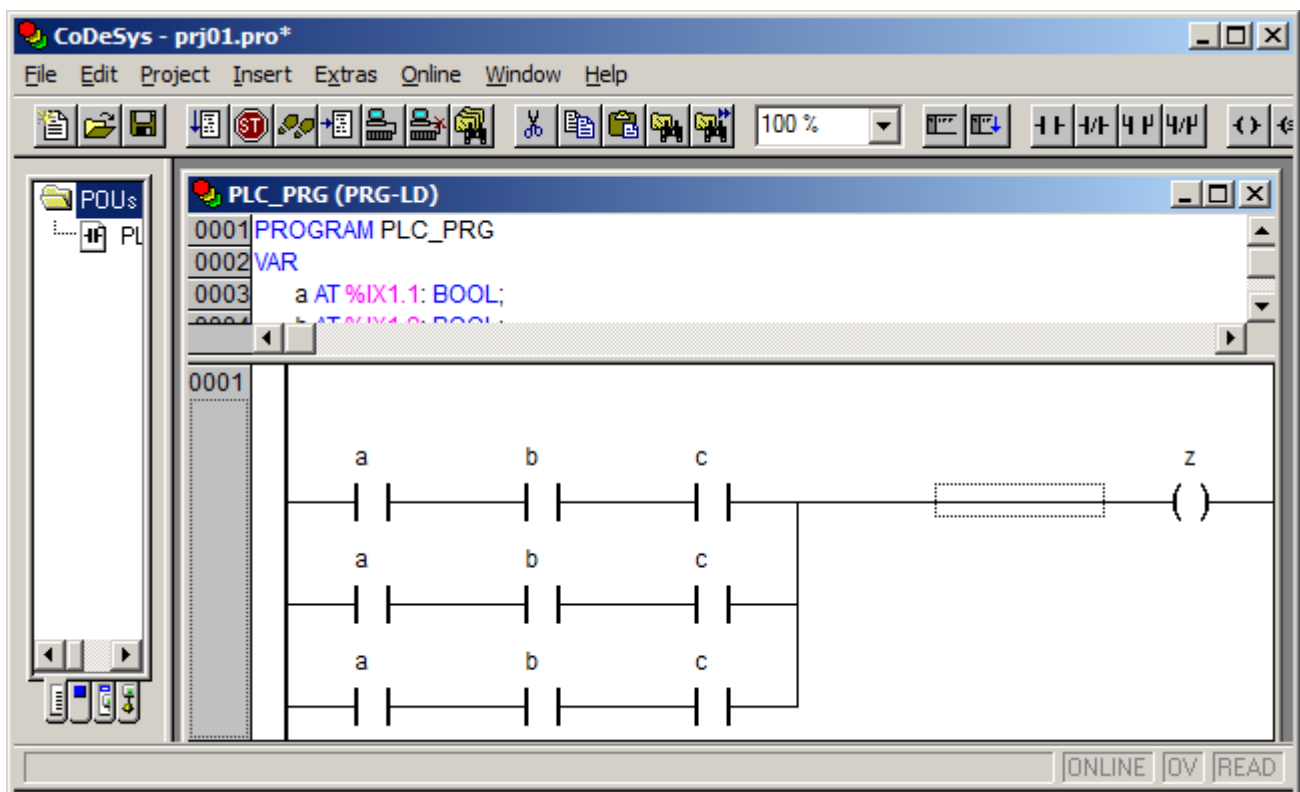


należy jeszcze skojarzyć elementy diagramu (zestyki i cewkę) ze zmiennymi. W tym celu należy zastąpić ??? nazwami odpowiednich zmiennych. Po zaznaczeniu zastępowanej nazwy, nazwę właściwej zmiennej można:

- wpisać z klawiatury,
- wybrać zmienną z okna wyboru dostępnego pod przyciskiem F2.

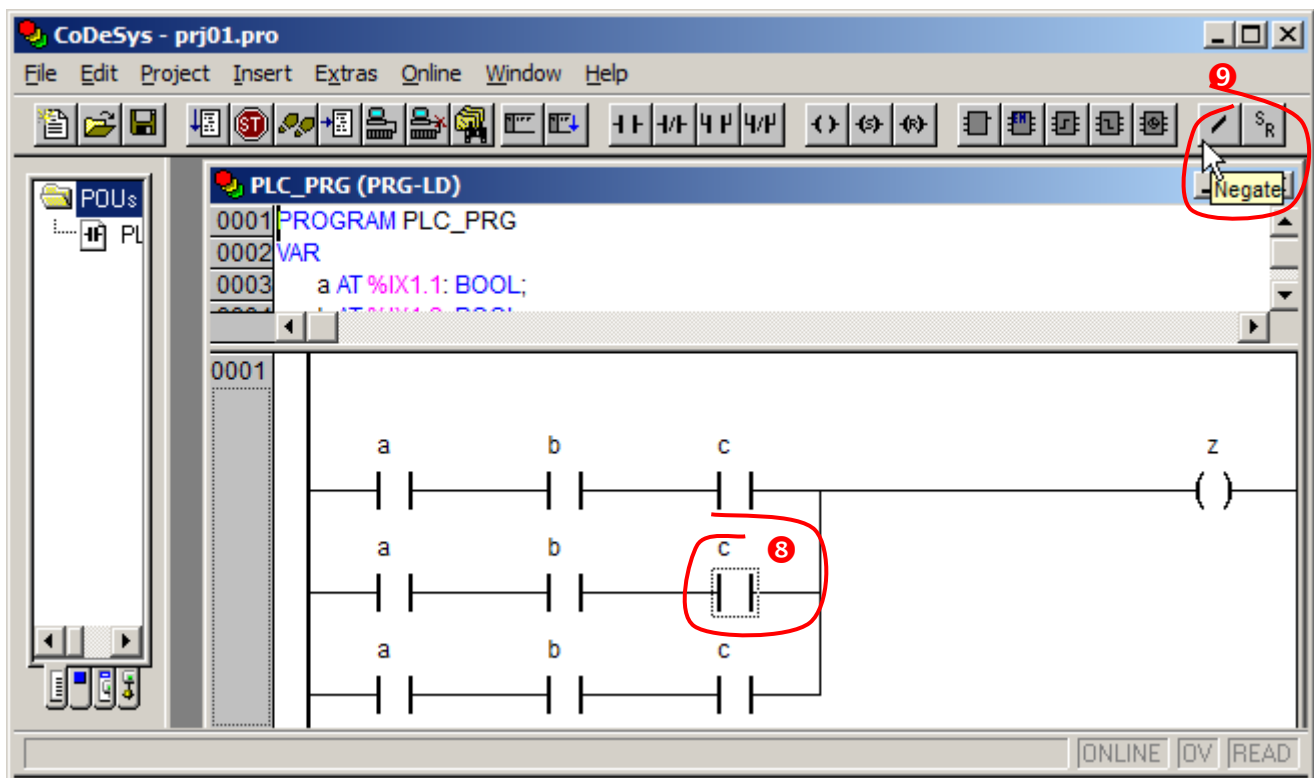


Po skojarzeniu zmiennych



należy jeszcze zamienić dwa zestyki ze *zwyrcnych* na *rozwiernie*, w tym celu należy:

- 8) wybrać modyfikowany element,
- 9) zmienić jego rodzaj.



Gotowy diagram LD, po powtórzeniu kroków 7) i 8) dla drugiego z modyfikowanych zestyków został przedstawiony na rysunku.

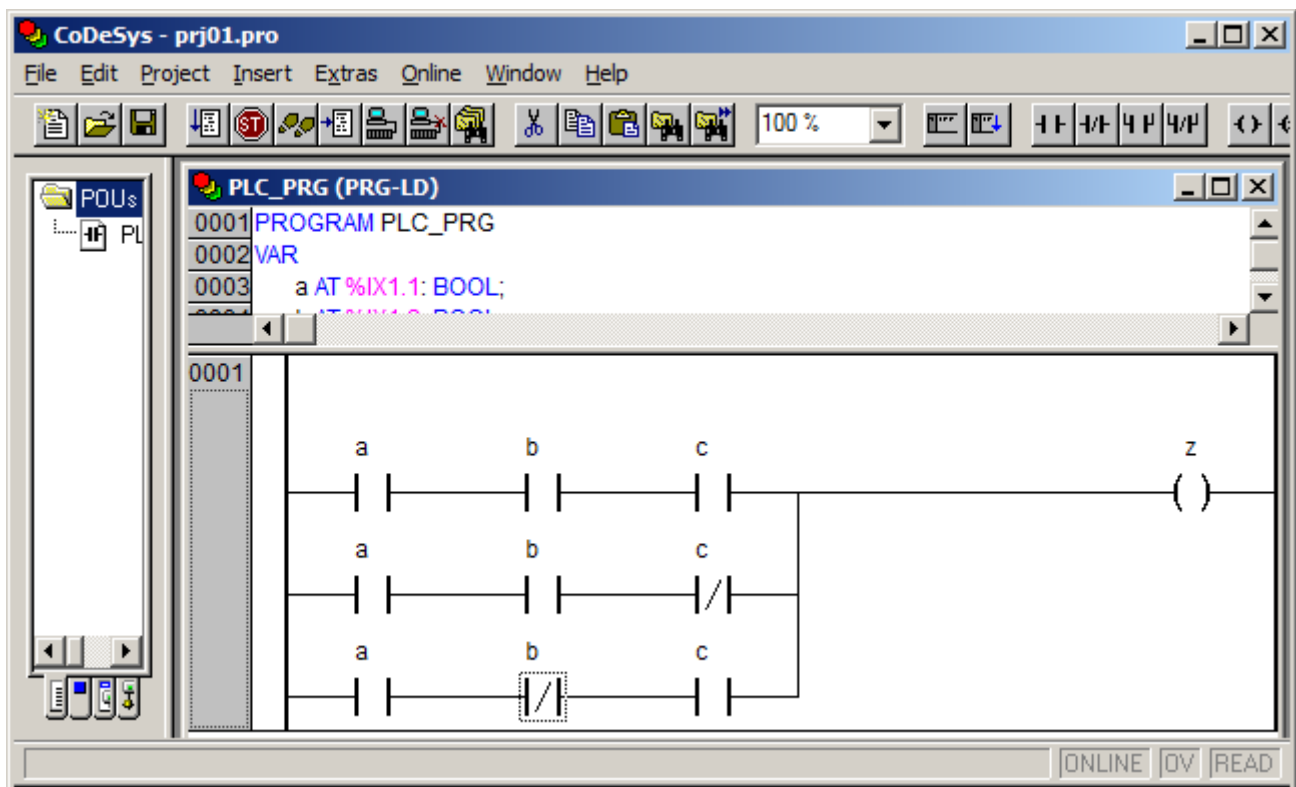


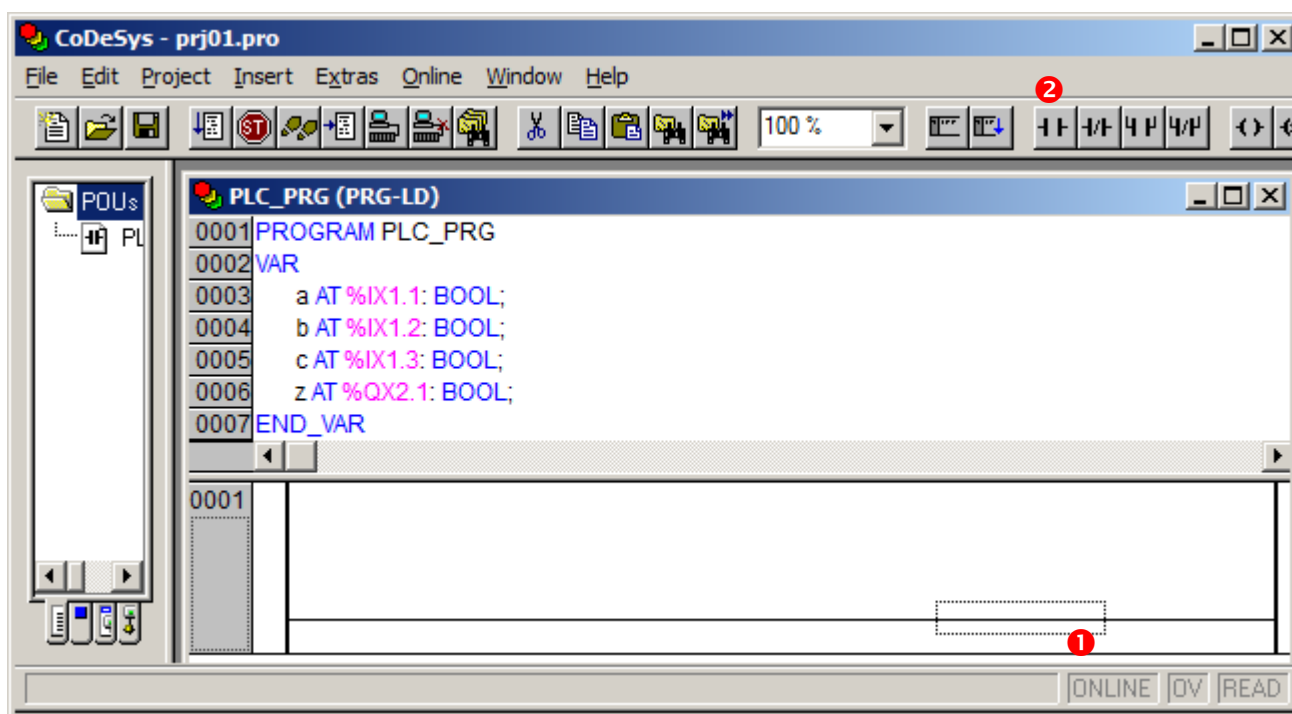
Diagram LD – funkcja 2

Narysowanie diagramu LD dla drugiej z zaproponowanych postaci funkcji sprowadza się do narysowania trzech szeregowo obwodów z których każda składa się z trzech równoległych gałęzi zawierających po jednym zestyku. W przedstawionym poniżej sposobie postępowania:

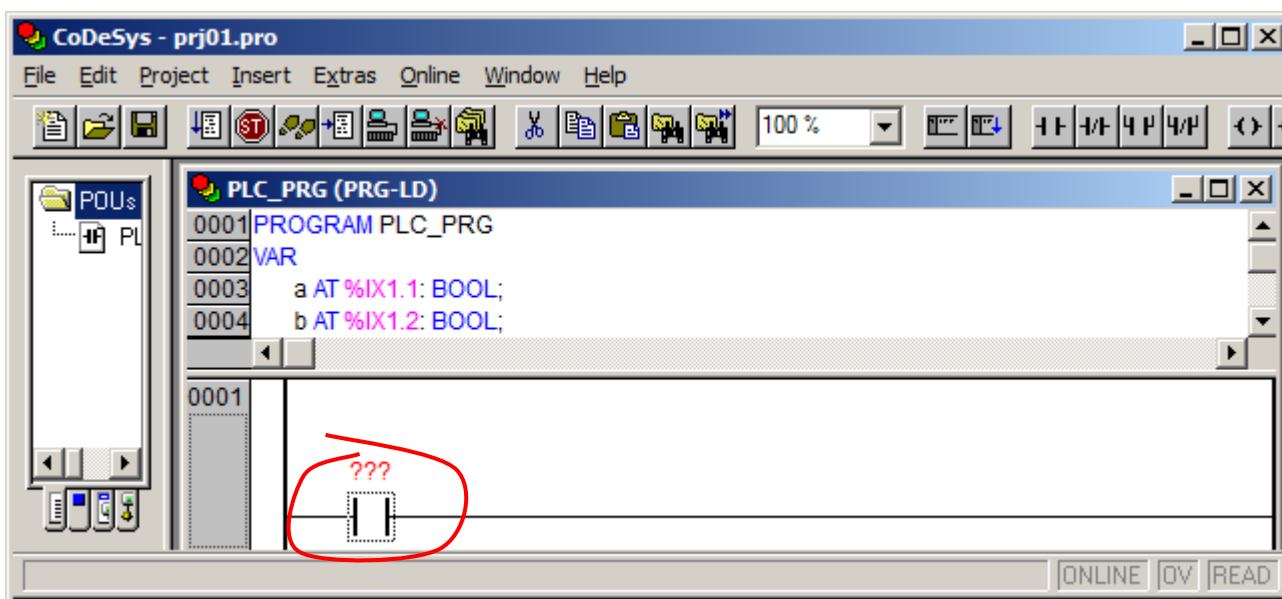
- najpierw wstawione zostaną trzy szeregowo połączone zestyki,
- następnie zostaną utworzone obwody składające się z gałęzi równoległych połączonych z kolejno: pierwszym, drugim i trzecim zestykiem.

Wstawienie pierwszego zestyku wymaga:

- 1) kliknięcia w edytorze programu na pustym polu,
- 2) wstawienia elementu.

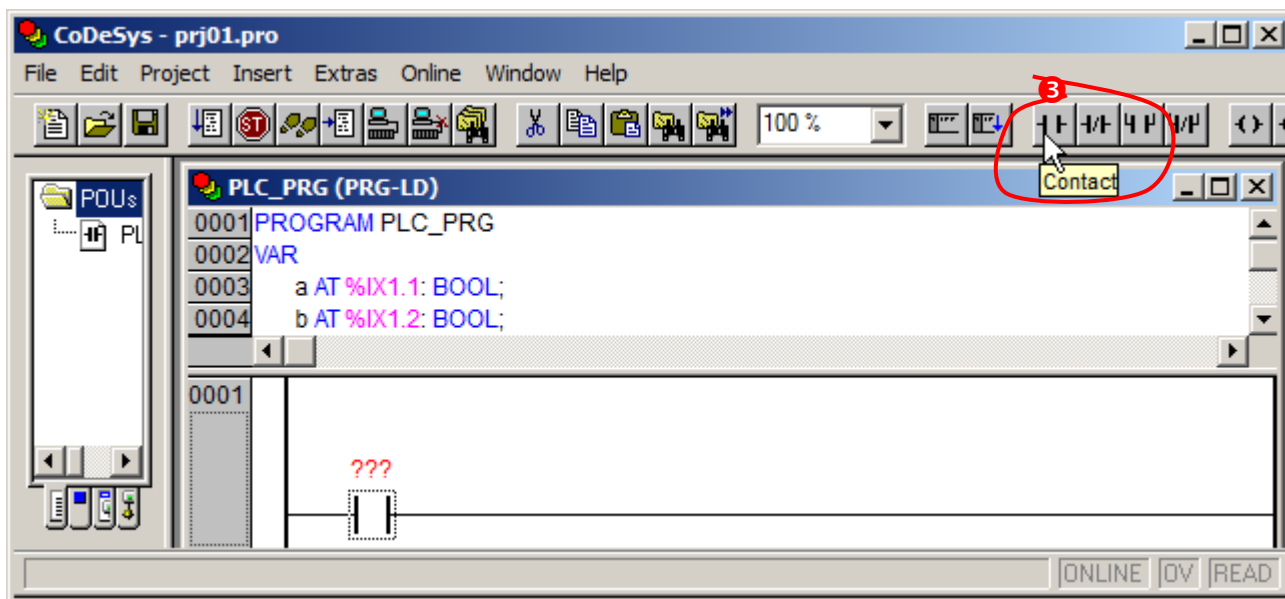


Po wstawieniu elementu jest on automatycznie zaznaczany:

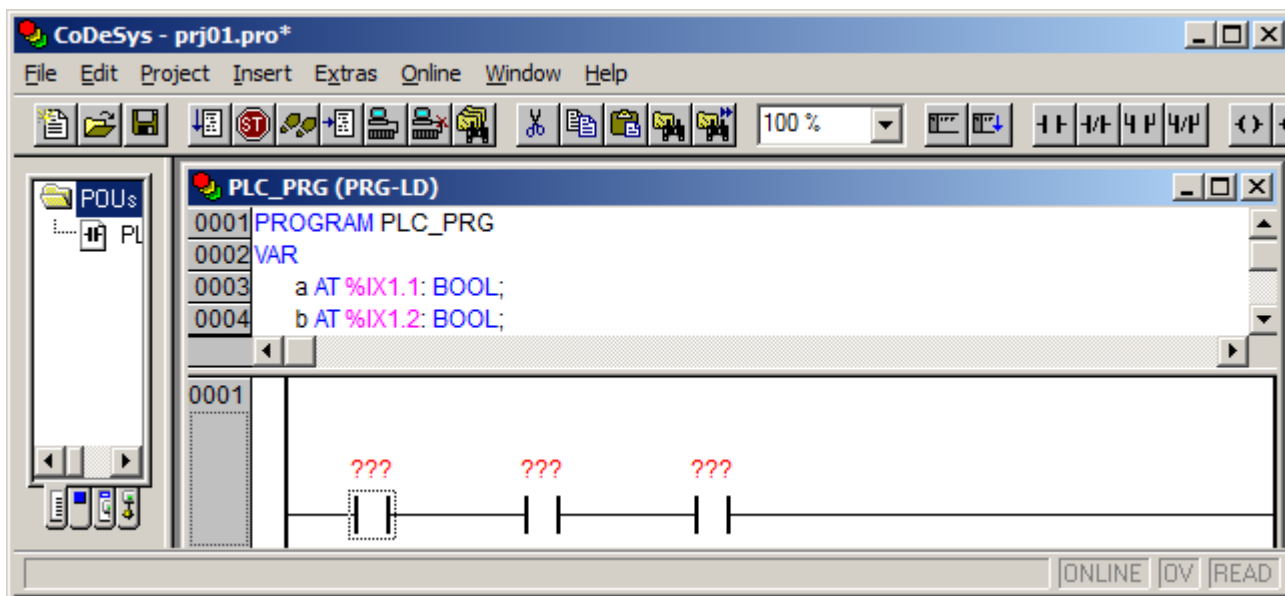


W kolejnym kroku diagram zostanie uzupełniony o dwa zestyki połączone szeregowo z dodanym wcześniej elementem, w tym celu:

- 3) z paska narzędziowego należy wybrać zestyk połączony szeregowo.



Stan edytora po wstawieniu trzech szeregowo połączonych zestyków przedstawiony został na poniższym rysunku.



Teraz każdy z zestyków zostanie uzupełniony o zestyk połączony z nim równolegle. W celu dodania elementu połączonego równolegle z wybranym należy:

- 4) kliknąć na elemencie pod którym zostanie dodany wstawiany zestyk,
- 5) z paska narzędziowego wybrać zestyk połączony równolegle.

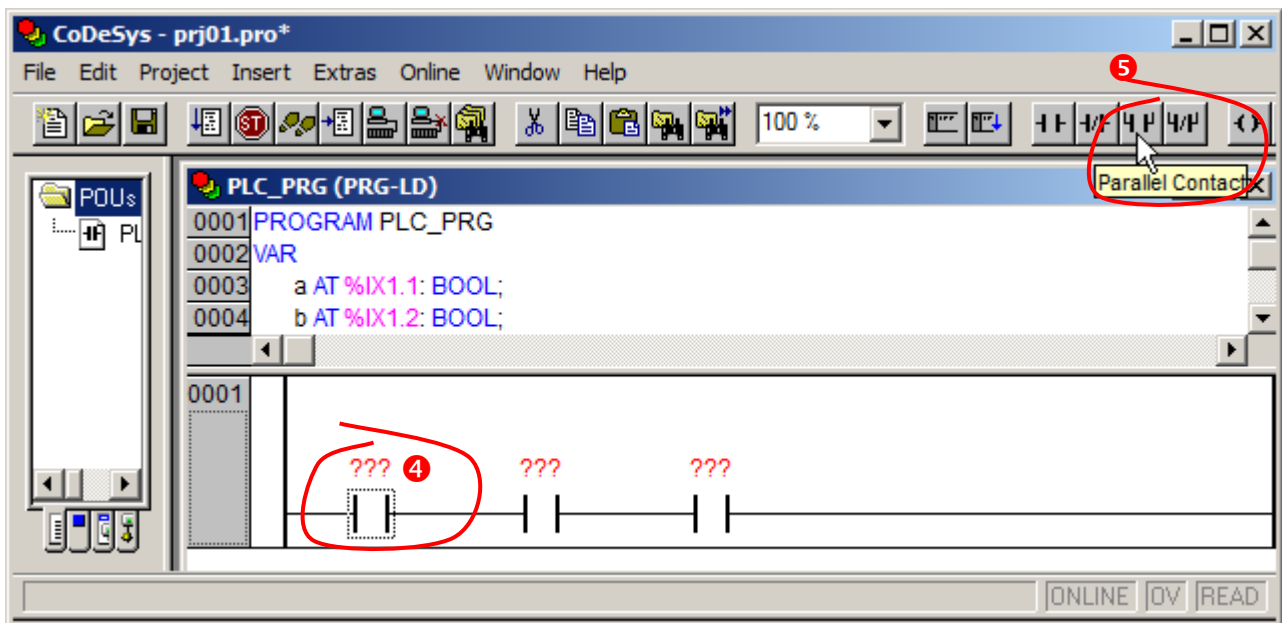
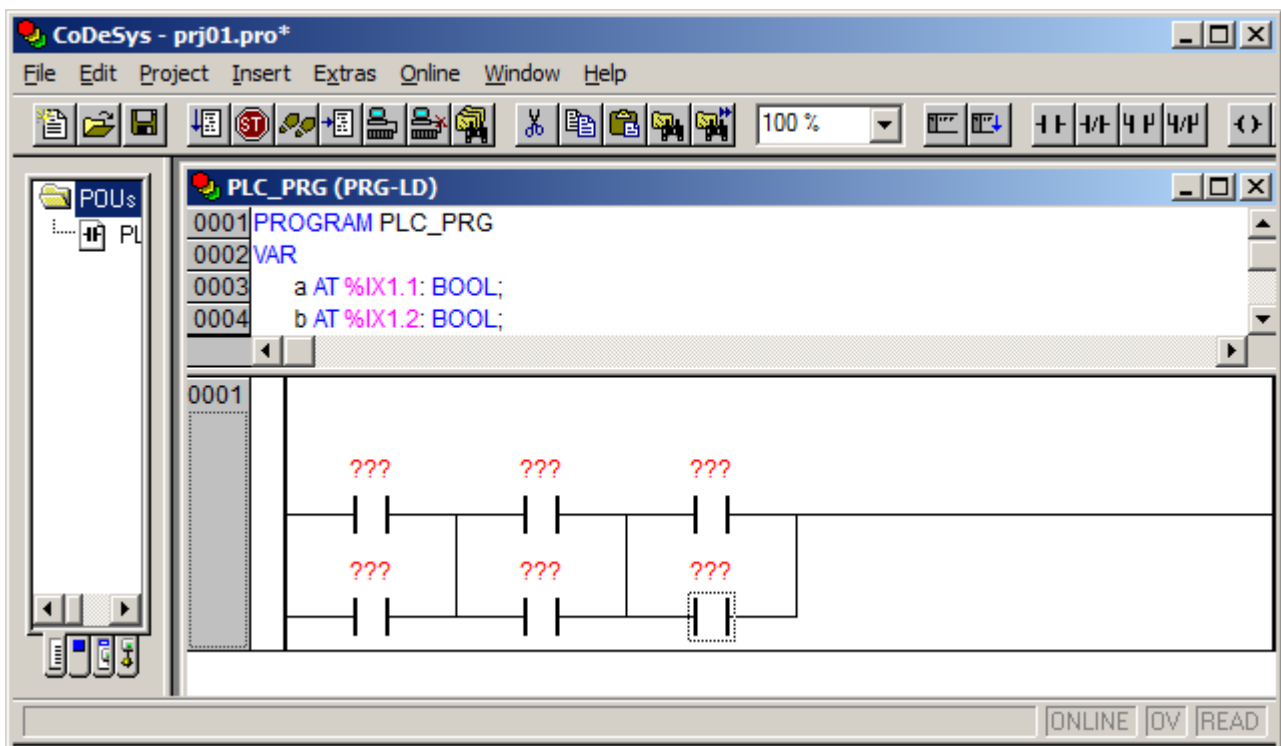
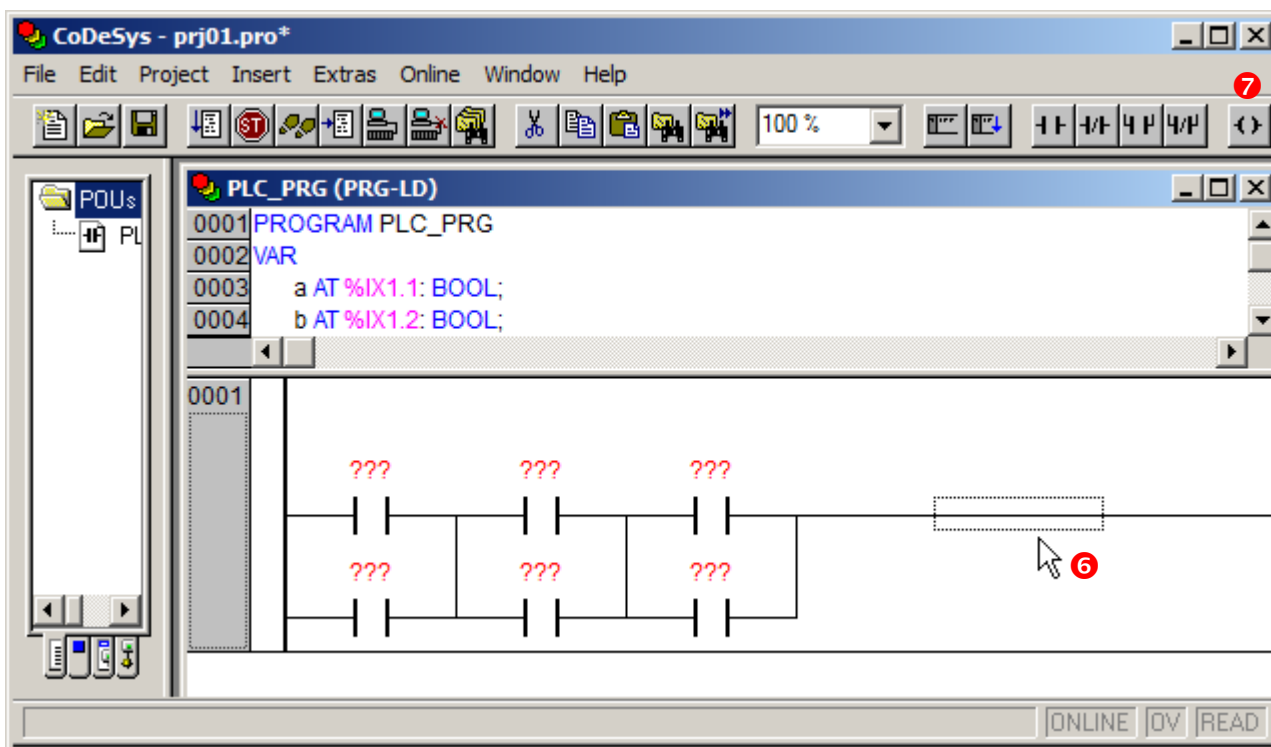


Diagram LD, po powtórzeniu kroków 4) i 5) dla drugiego i trzeciego zestyku, został przedstawiony na rysunku.

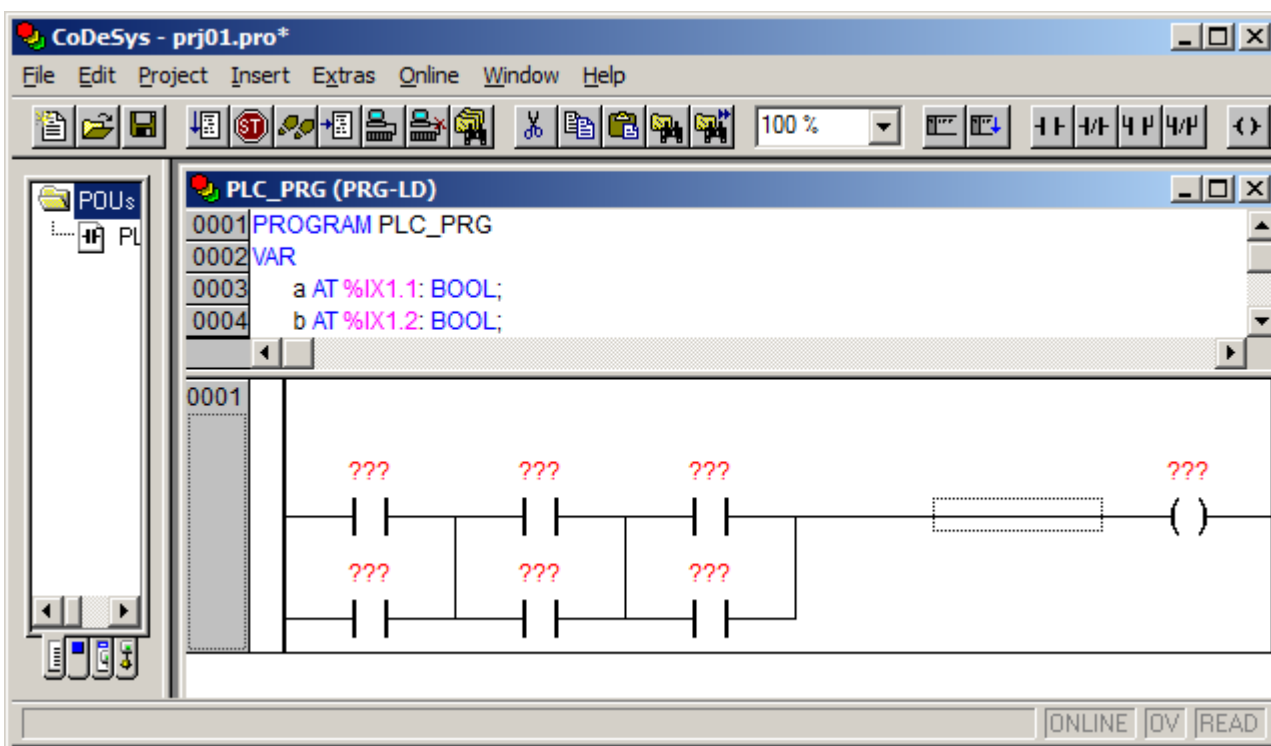


Na koniec należy jeszcze uzupełnić program wstawiając cewkę ustawiającą wartość zmiennej wyjściowej. W tym celu należy:

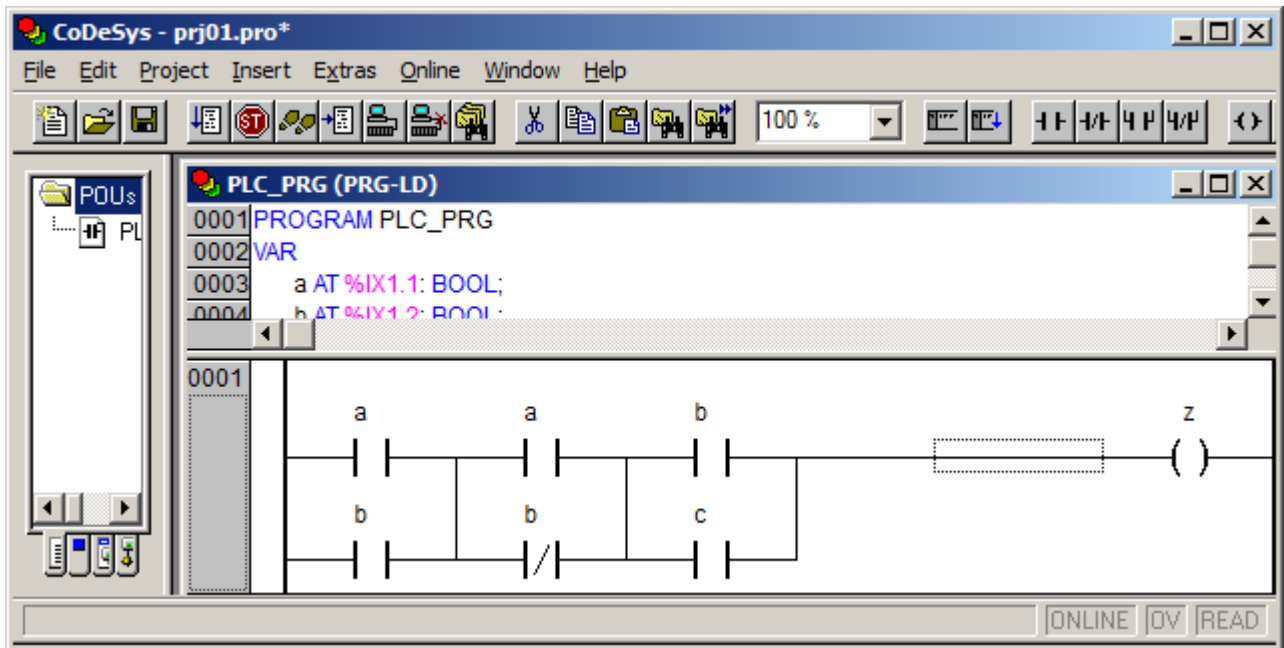
- 6) kliknąć po prawej stronie szczebla,
- 7) wstawić cewkę.



Po dodaniu cewki



należy, podobnie jak w przypadku pierwszej funkcji (punkt Diagram LD – funkcja 1), skojarzyć zmienne z elementami diagramu i wprowadzić w odpowiednich miejscach zestyki rozwiernie. Gotowy diagram LD dla drugiej z rozważanych postaci funkcji, przedstawia poniższy rysunek.



4. Symulacja działania programu

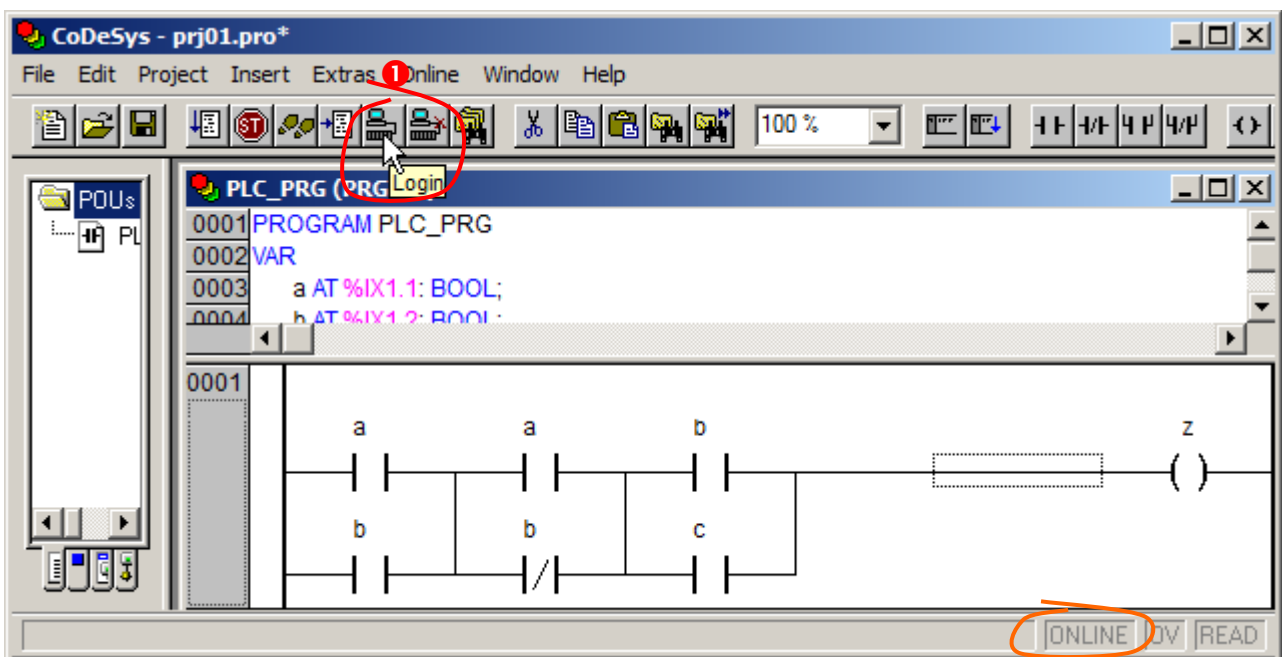
Działanie aplikacji można przetestować uruchamiając projekt na symulatorze sterownika ([szczegółowe omówienie zagadnienia można znaleźć w materiale CoDeSys zamieszczonym na stronie przedmiotu](#)).

W tym celu należy:

- 0) skompilować projekt,
- 1) nawiązać połączenie ze sterownikiem,
- 2) uruchomić program na sterowniku.

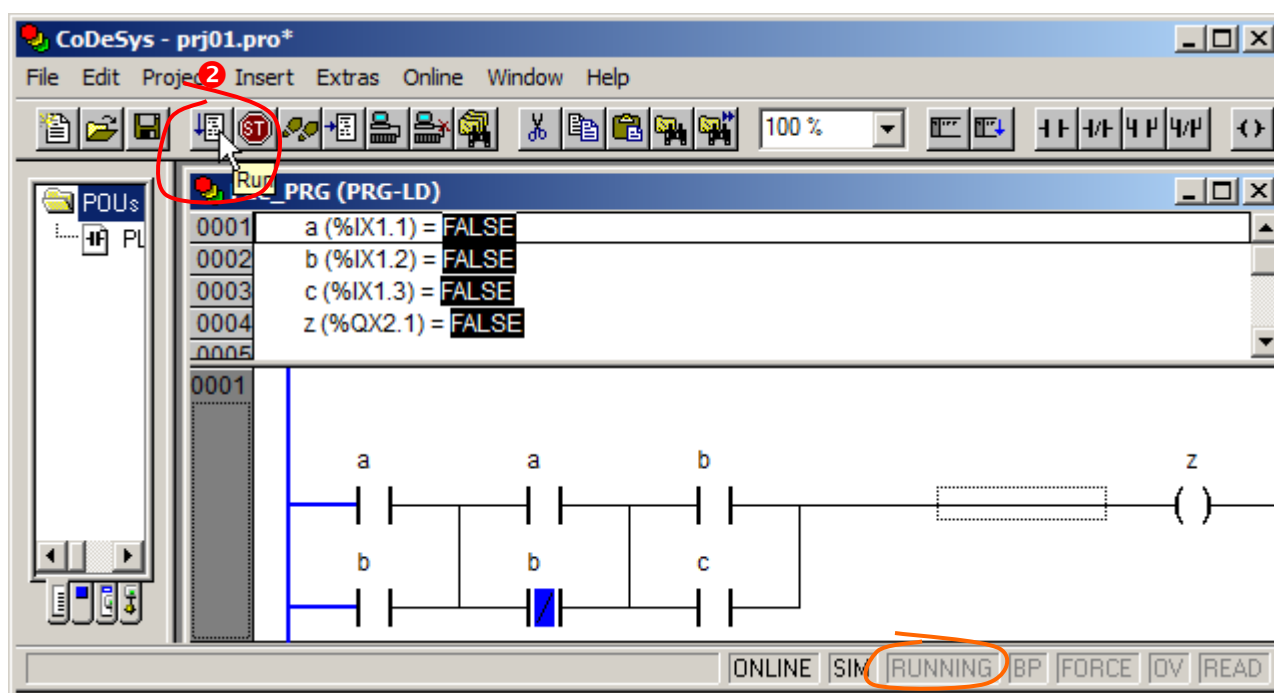
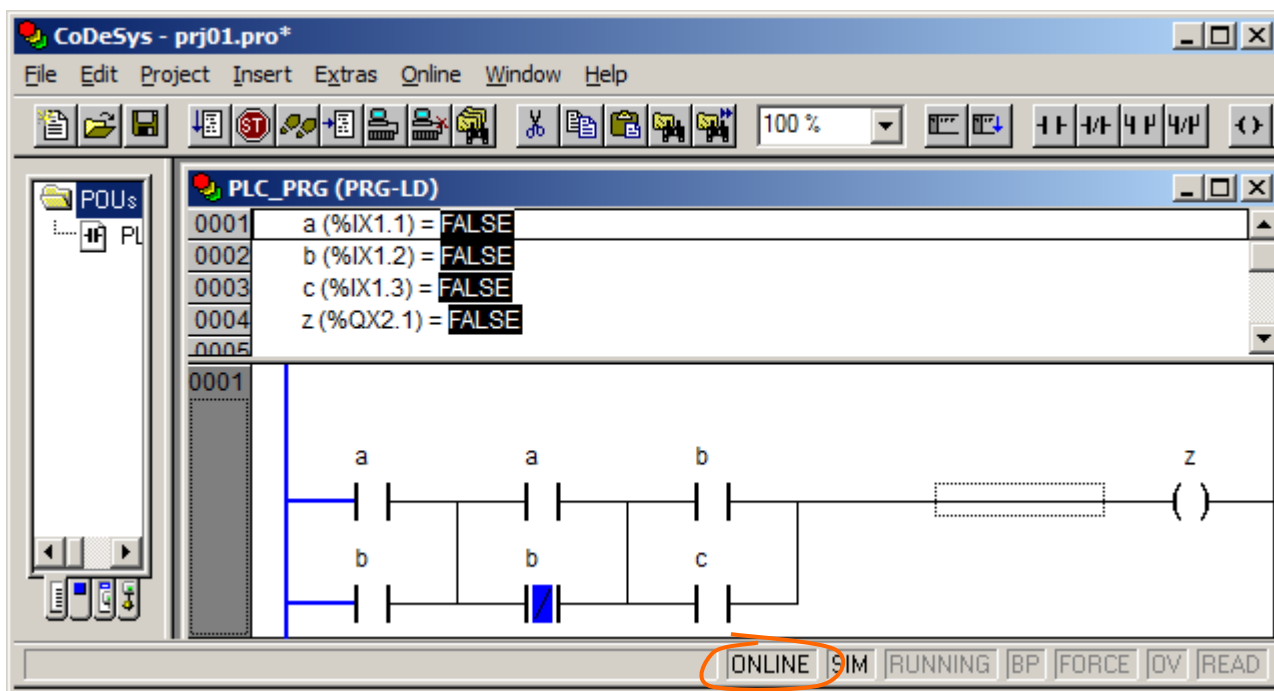
Ze względu na to, że środowisko rejestruje zmiany projektu i wykonuje jego automatyczną kompilację w przypadku próby nawiązywania połączenia ze sterownikiem ze zmodyfikowaną wersją projektu, krok 0. można pominąć wykonując od razu krok 1.

Kroki 1) i 2) zostały przedstawione na poniższych rysunkach.

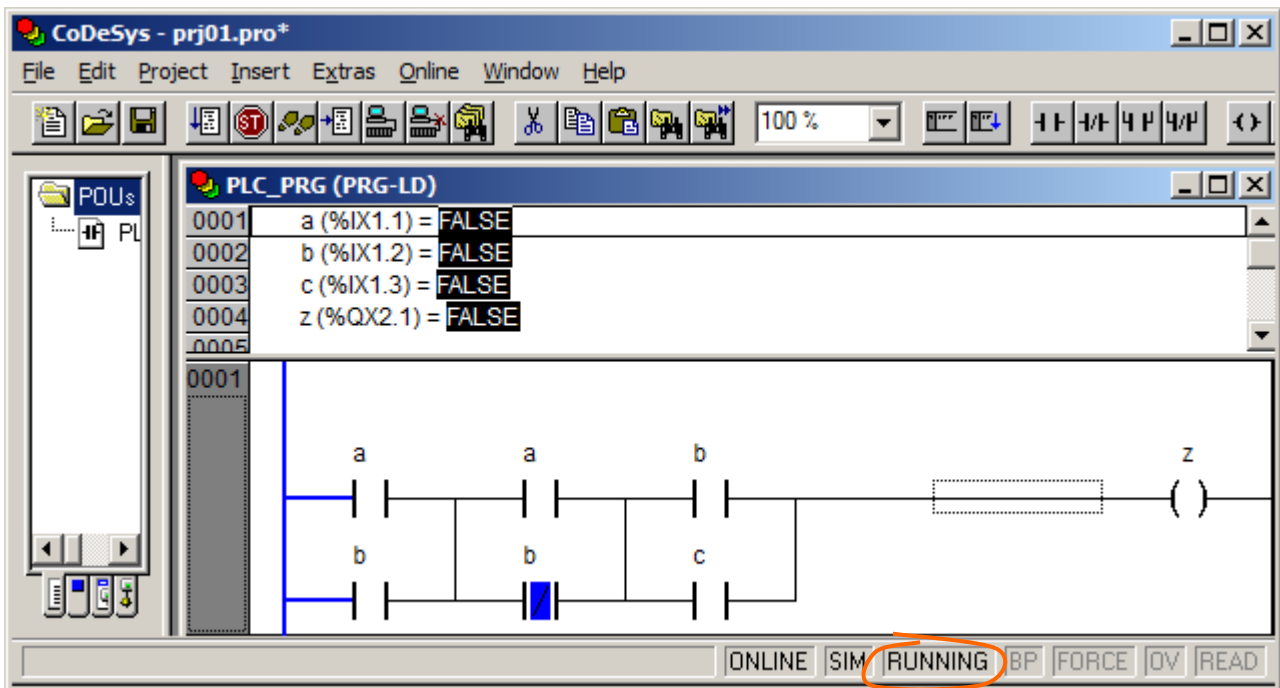


Po wykonaniu kroku 1., w przypadku braku błędów kompilacji, CoDeSys przechodzi w tryb pracy ONLINE.

Linia statusu wyświetla aktualny stan pracy programu: na rysunku powyżej nie nawiązano jeszcze połączenia ze sterownikiem (napis ONLINE jest szary), na rysunku poniżej połączenie zostało nawiązane (napis ONLINE jest czarny).



Po wykonaniu kroku 2. CoDeSys uruchamia program na sterowniku. Po nawiązaniu połączenia, tzn. w trybie ONLINE, linia statusu wyświetla również tryb pracy sterownika: na rysunku powyżej sterownik pracuje w trybie STOP (napis RUNNING jest szary), na rysunku poniżej sterownik pracuje w trybie RUN (napis RUNNING jest czarny).



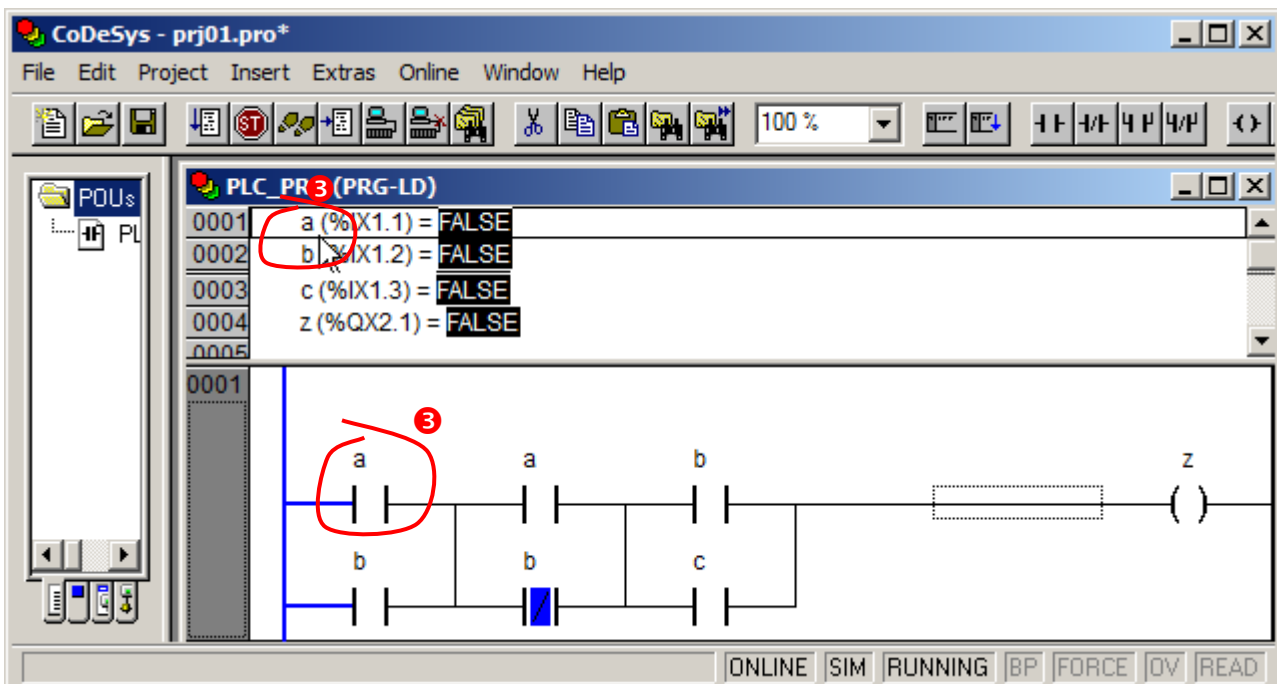
Testowanie programu sprowadza się do sprawdzania wartości zmiennych wyjściowych dla różnych kombinacji zmiennych wejściowych. Na powyższym rysunku widać, że na starcie wszystkie zmienne wejściowe otrzymały wartość FAŁSZ co odpowiada nieprawidłowym wartościom odpowiednich cech badanego detalu. W takim przypadku zmienna z otrzymuje również wartość FAŁSZ co prowadzi do umieszczenia nieprawidłowego detalu w *pojemniku II*.

Reakcję programu na inną kombinację sygnałów wejściowych można sprawdzić:

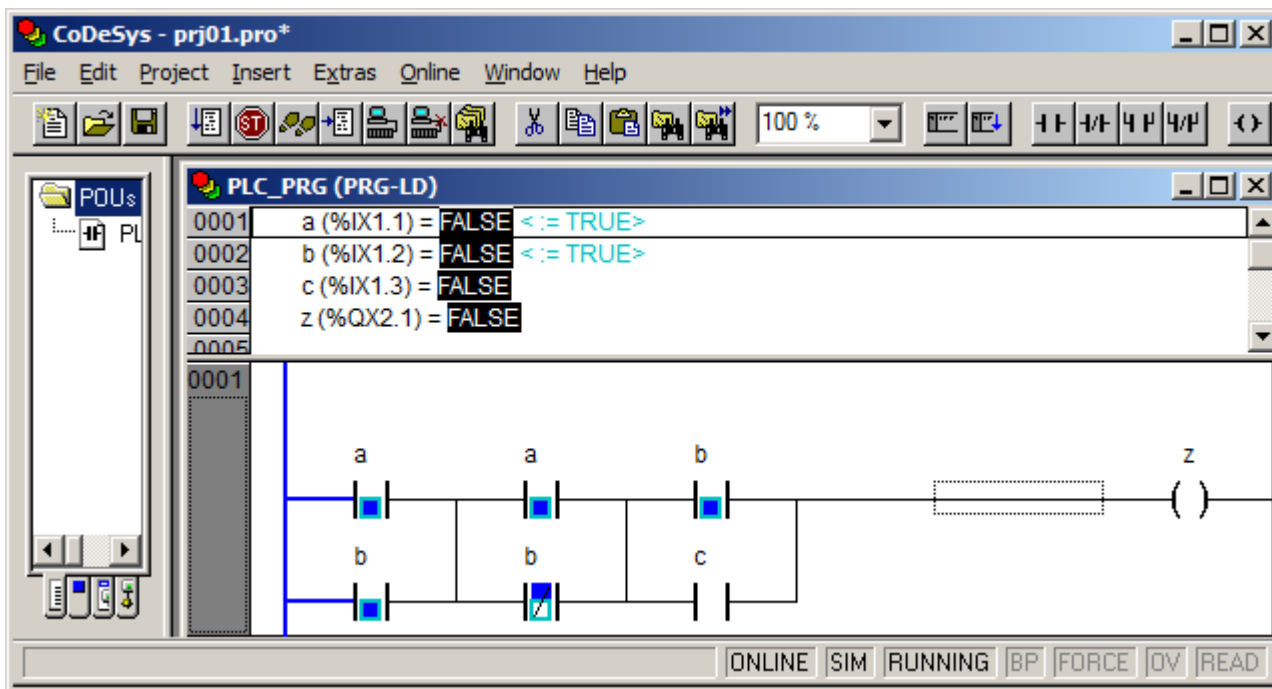
- 3) ustawiając wartości odpowiednich zmiennych wejściowych (wykonując podwójne kliknięcie na nazwie zmiennej)
- 4) przekazując nowe wartości zmiennych do sterownika (Ctrl+F7).

Na poniższych rysunkach pokazany został stan programu

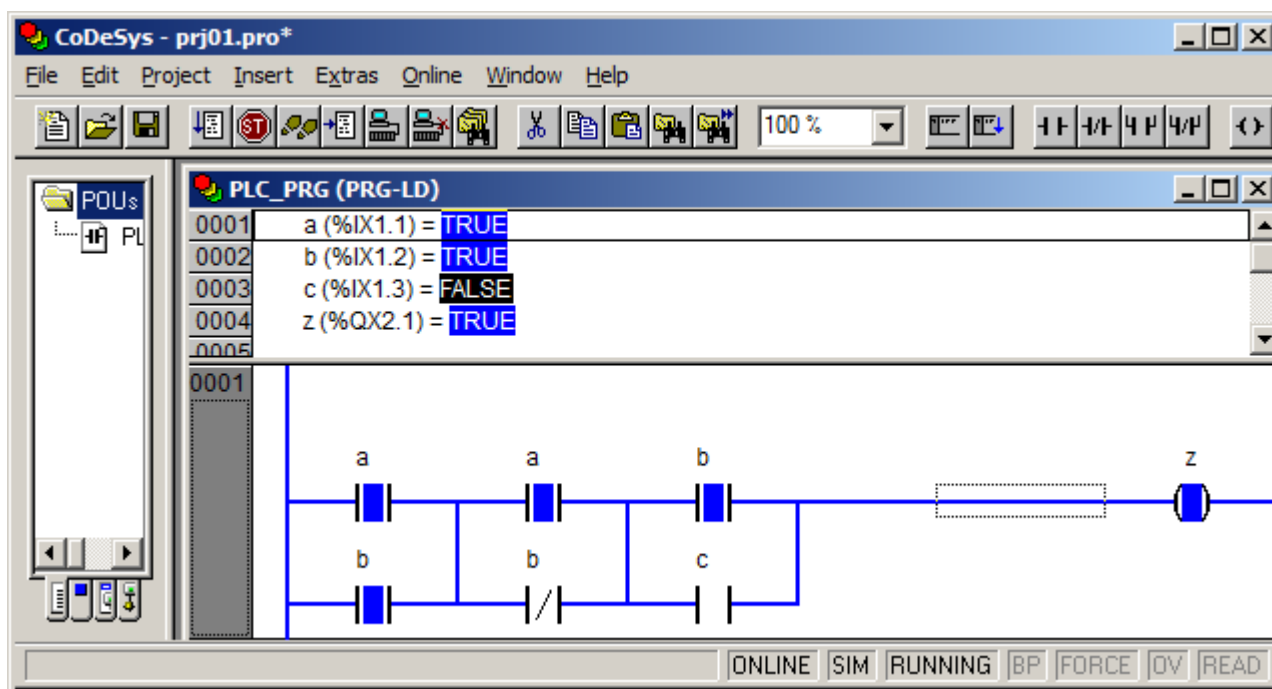
- przed zamianą wartości zmiennych:



po ustawieniu cech a i b na prawidłowe (zmienna a i b ustawione na TRUE) :



po przekazaniu zmiennych do sterownika:



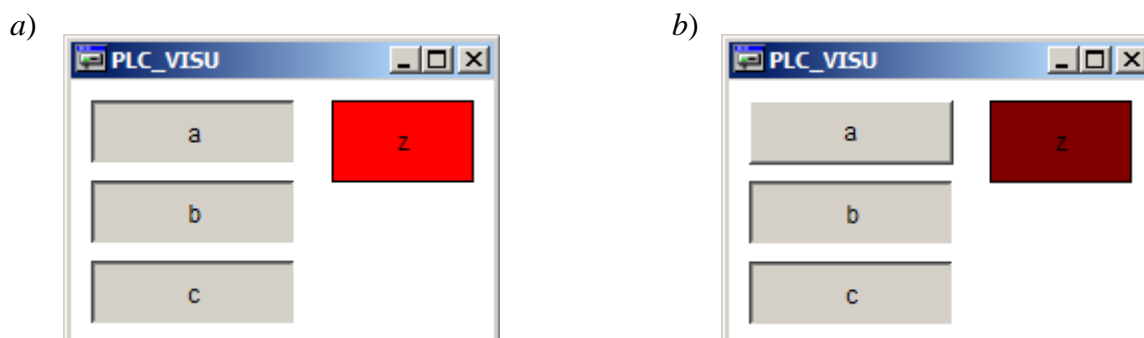
Detal o prawidłowych cechach a i b jest więc kierowany, zgodnie z opisem w punkcie 1., do pojemnika I. Symulację można przerwać zrywając połączenie ze sterownikiem tak jak to zostało pokazane na poniższym rysunku.



5. Wizualizacja

Testowanie działania aplikacji ułatwiają wizualizacje ([szczegółowe omówienie zagadnienia można znaleźć w materiale CoDeSys zamieszczonym na stronie przedmiotu](#)). Poniżej zostanie omówiona najprostsza z możliwych:

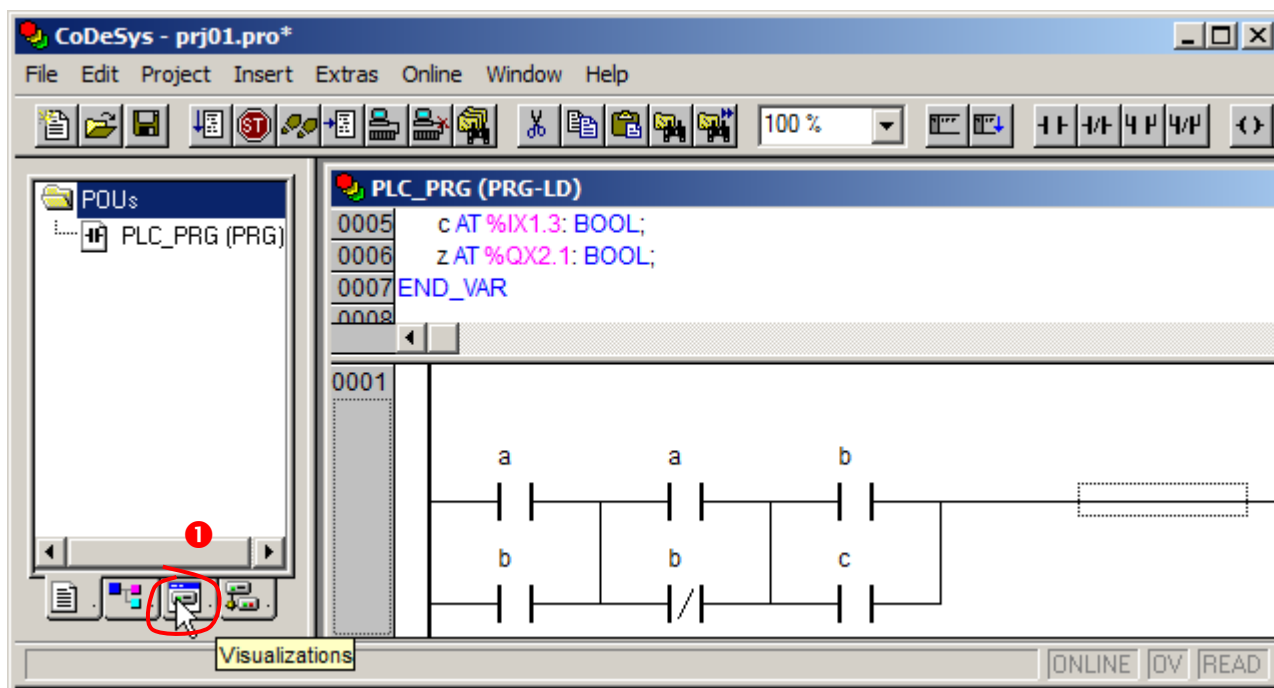
- pozwalająca przyciskami przełączającymi *a*, *b* i *c* zmieniać wartości sygnałów z czujników monitorujących odpowiednie cechy detalu,
- wyświetlająca stan sygnału sterującego zwrotnicą przy pomocy kolorowego prostokąta rozjaśnianego z chwilą otrzymania przez sygnał wartości „1” (TRUE) i przygaszanego gdy sygnał otrzyma wartość „0” (FALSE).

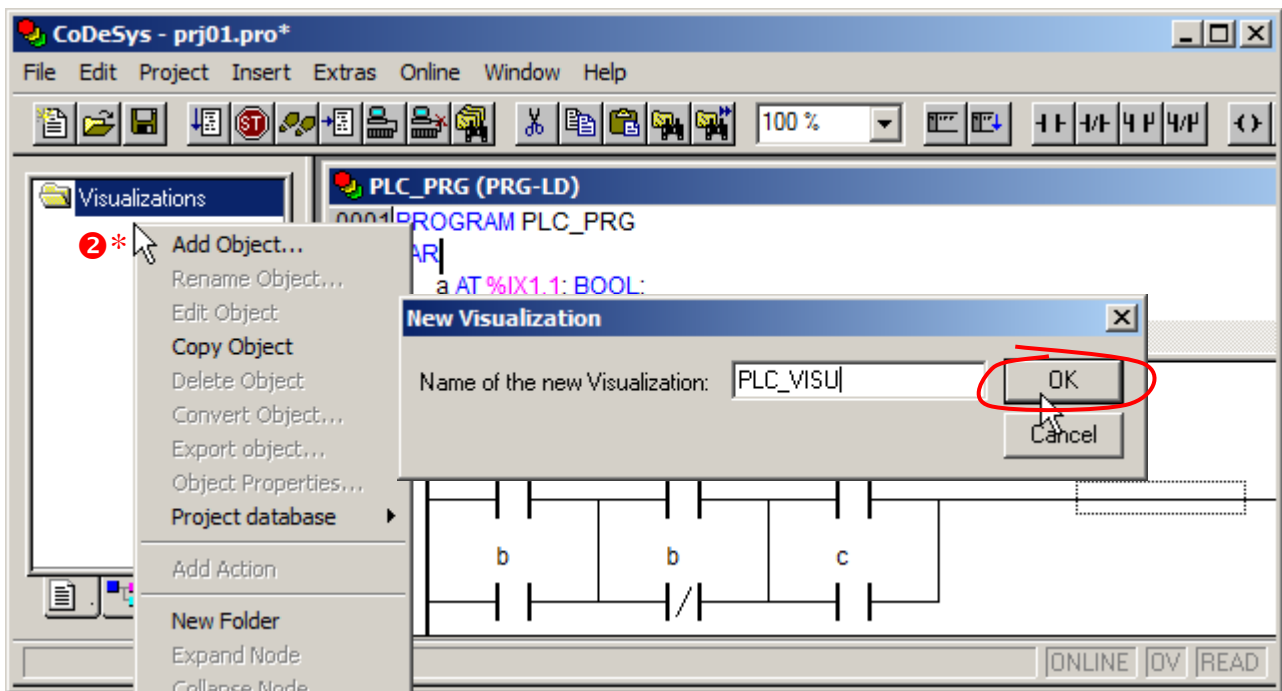


*Wizualizacja a) wszystkie cechy detalu prawidłowe – zwrotnica kieruje detal do pojemnika I
b) prawidłowe tylko cechy b i c – zwrotnica kieruje detal do pojemnika II.*

Do stworzenia przykładowej wizualizacji niezbędne jest rozbudowanie projektu. W tym celu w oknie zasobów należy:

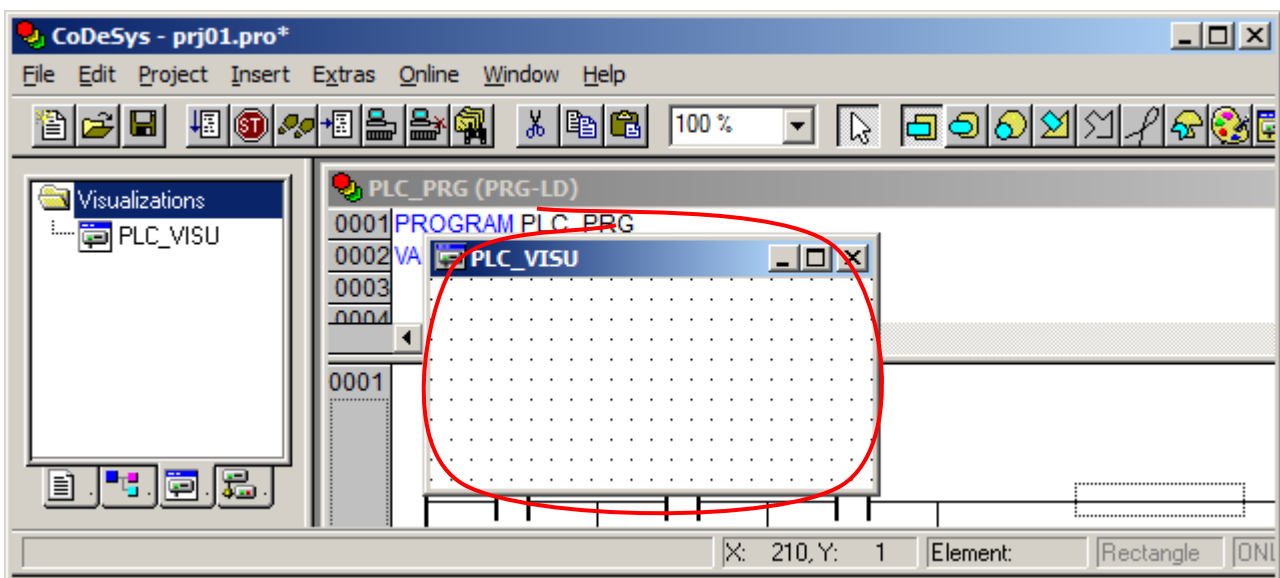
- 1) przełączyć się na zakładkę z wizualizacjami,
- 2) dodać nową wizualizację nadając jej nazwę PLC_VISU (domyślna nazwa dla wizualizacji startowej).





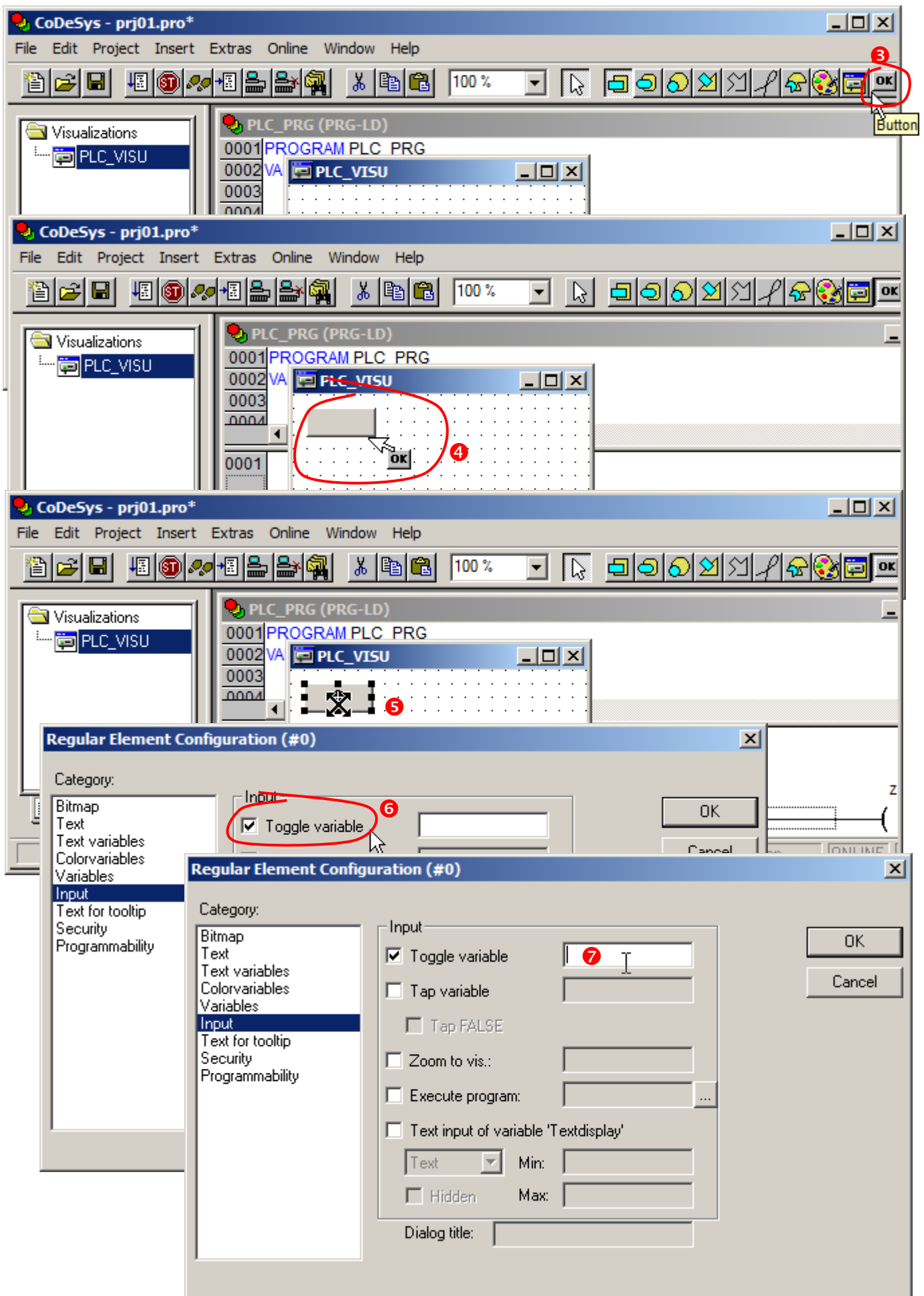
2* - kliknięcie wykonać prawym przyciskiem myszy.

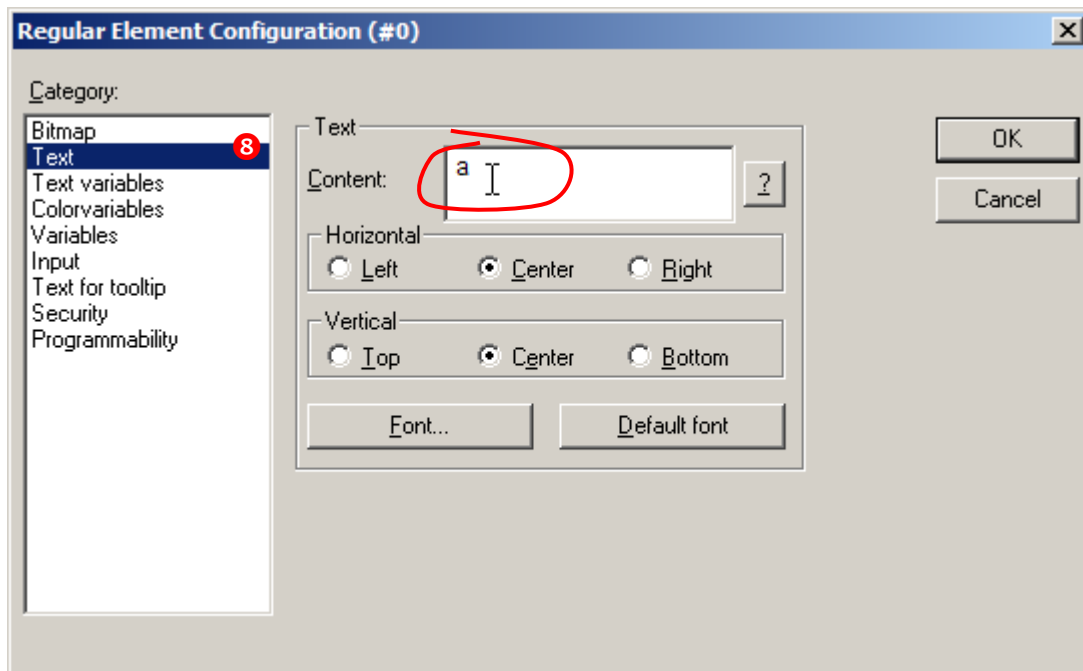
Po wykonaniu kroków 1) i 2) otwierane jest puste okienko projektu wizualizacji.



W przykładowej wizualizacji wartości sygnałów wejściowych zmieniane są przyciskami przełączającymi, w celu dodania przycisku należy:

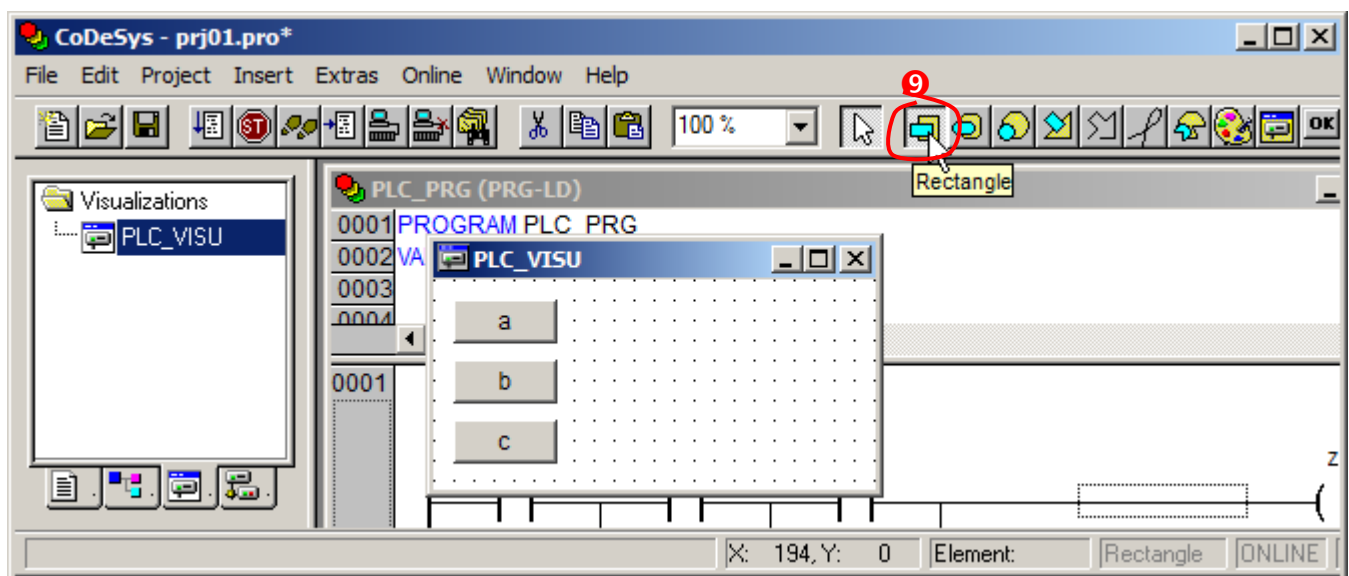
- 3) wskazać na pasku narzędziowym przycisk OK,
- 4) zaznaczyć w projekcie obszar przeznaczony na dodawany obiekt,
- 5) dwukrotnie kliknąć na przycisku,
- 6) w oknie konfiguracyjnym obiektu ustalić jego wejście na zmienną przełączającą – opcja **Toggle variable**,
- 7) w polu obok opcji wprowadzić nazwę zmiennej, najbezpieczniej wybierając ją z okna wywołanego przy pomocy przycisku F2,
- 8) bez wychodzenia z okna konfiguracyjnego ustawić odpowiednią etykietę przycisku.

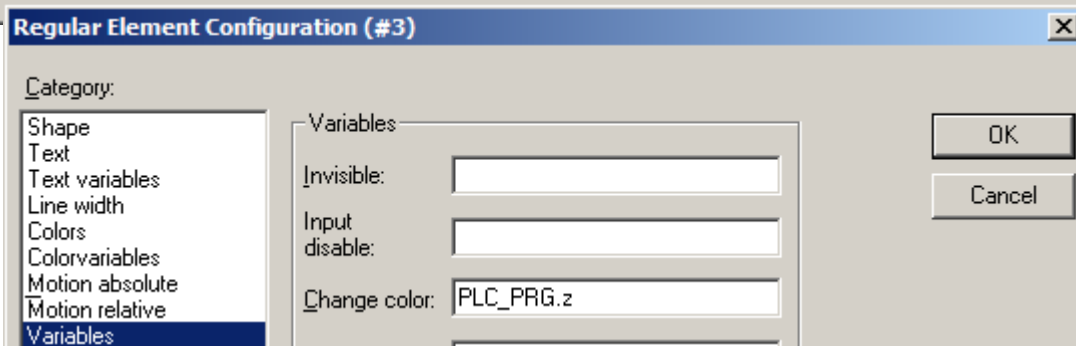
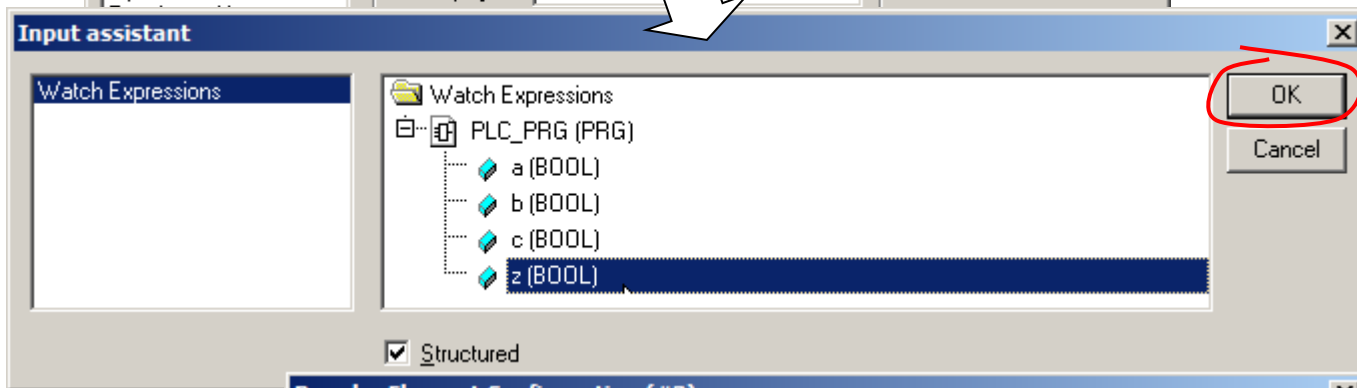
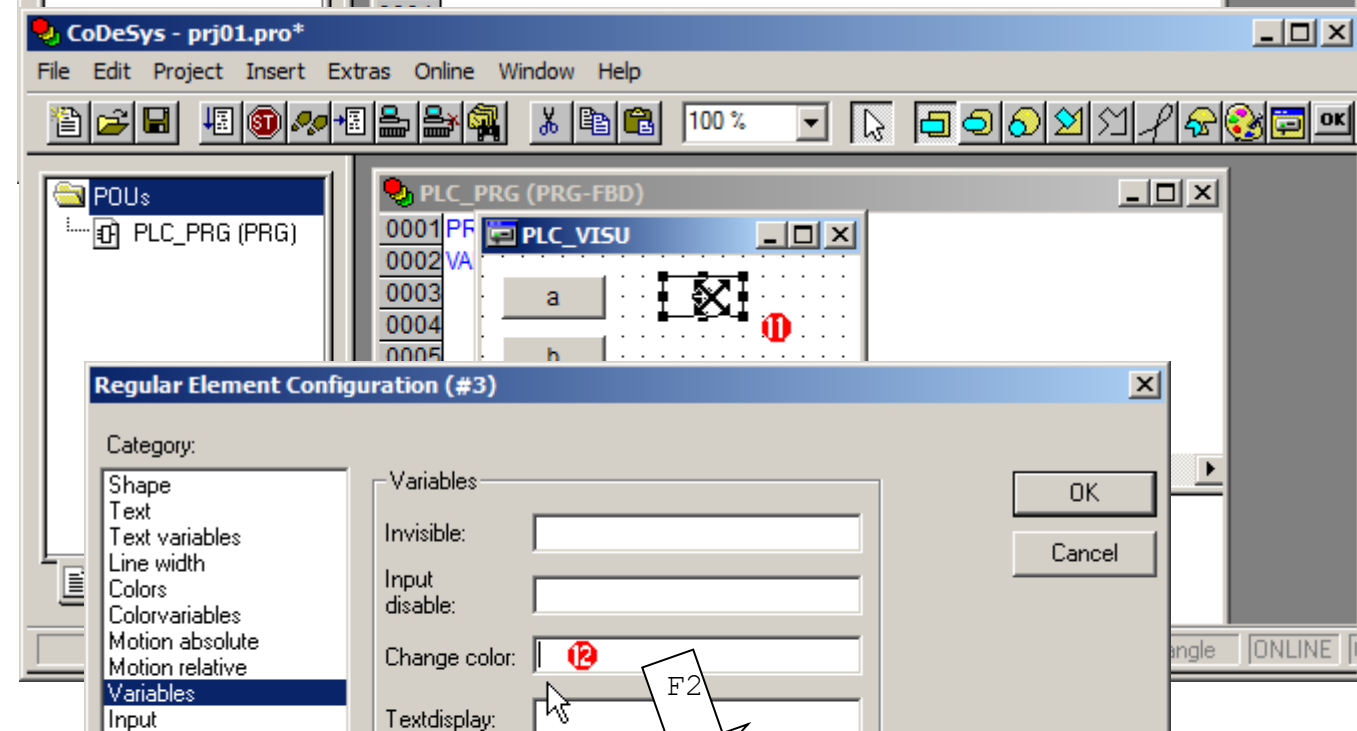
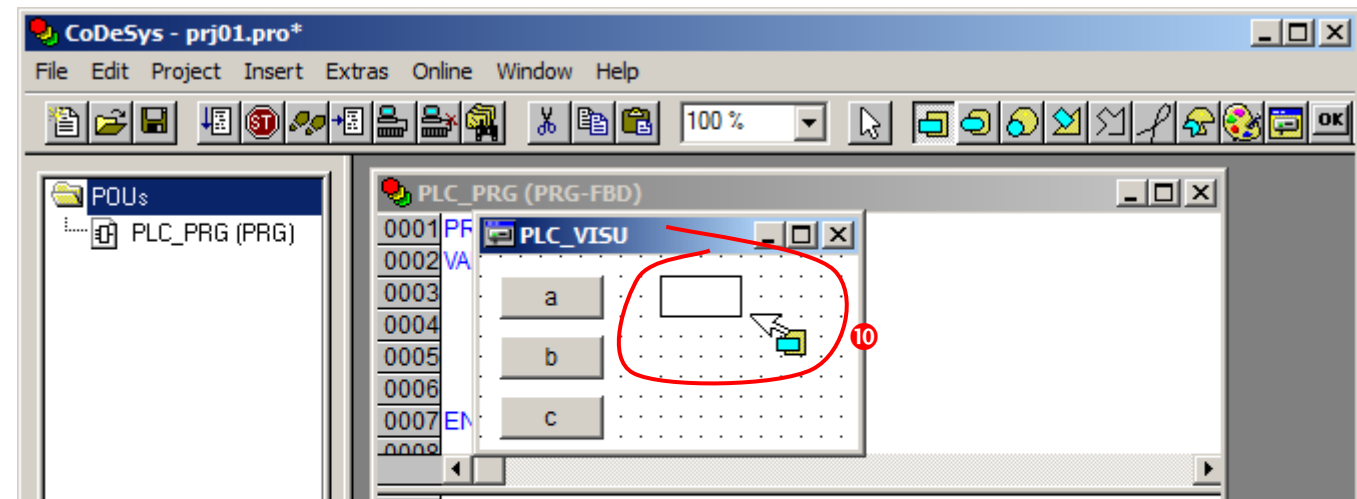


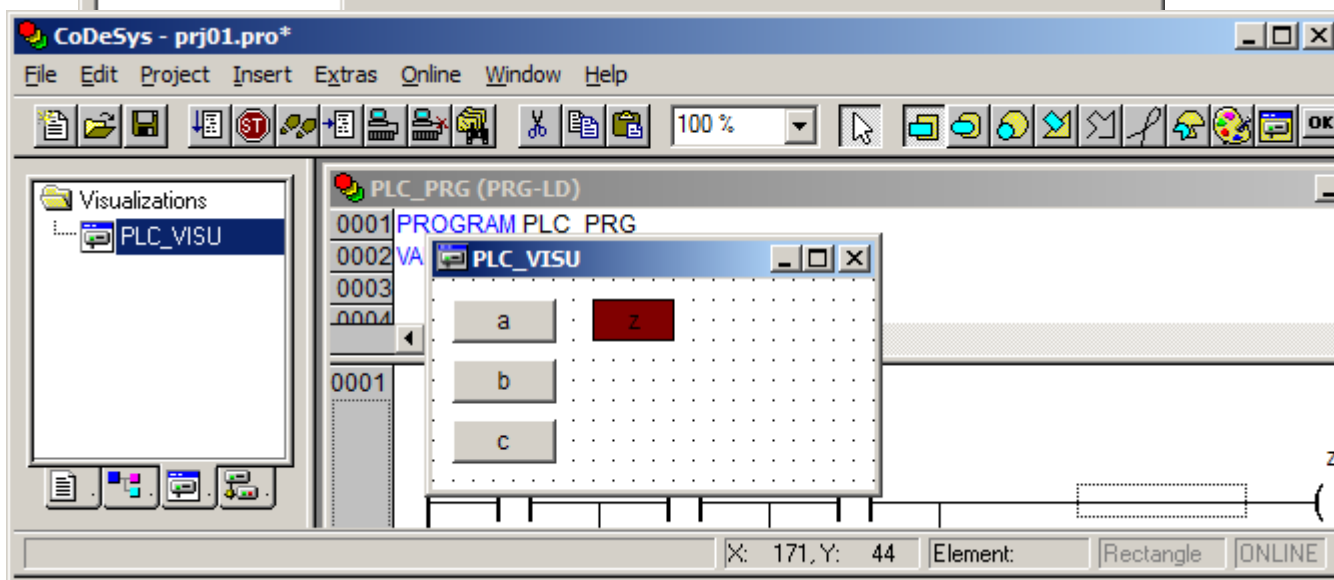
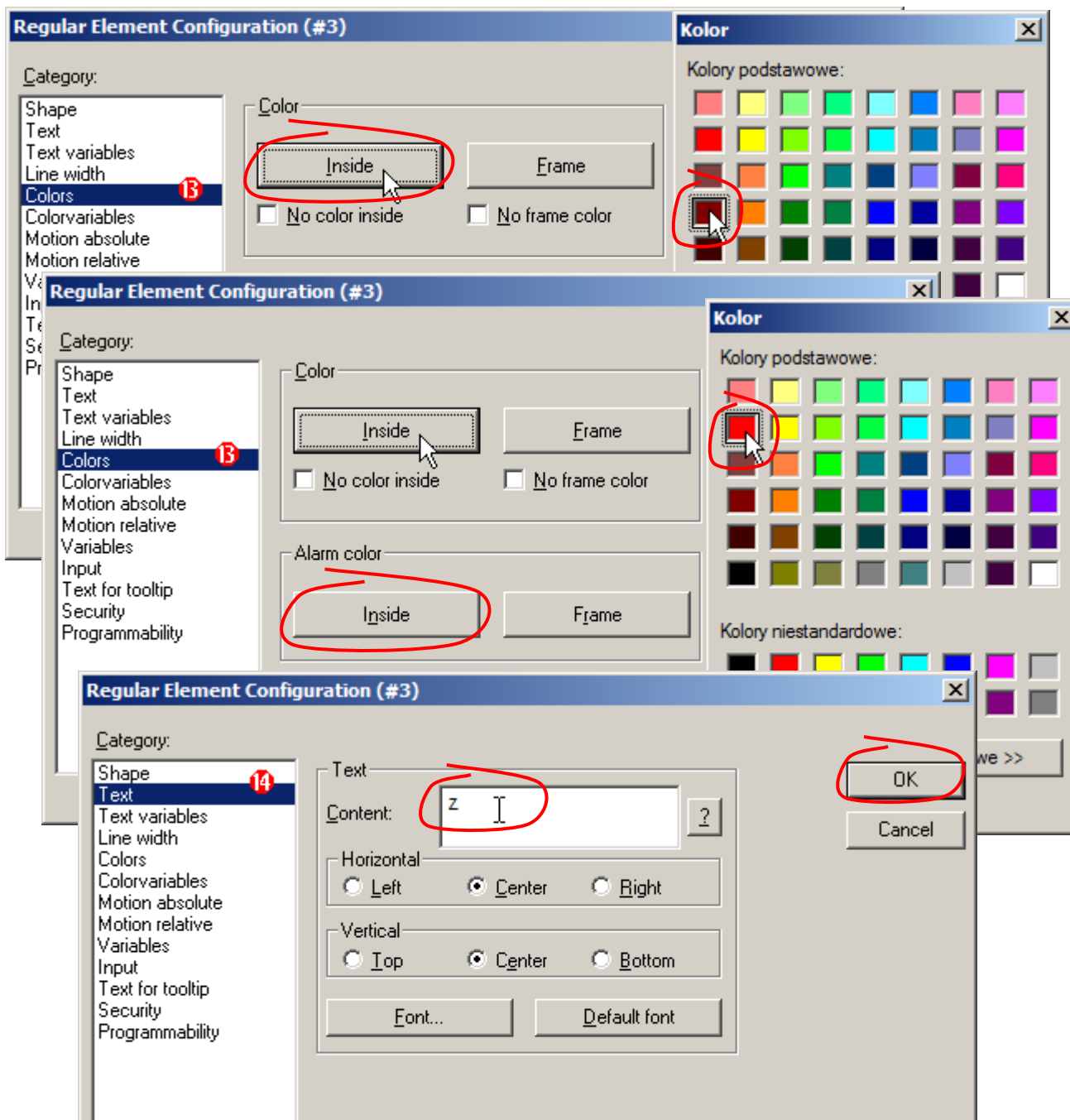


Po trzykrotnym powtórzeniu kroków 3) – 8) w wizualizacji brakuje jedynie elementu wyświetlającego stan sygnału sterującego zwrotnicą. przełączającymi – element ten dodawany jest w podobnie jak przyciski reprezentujące sygnały wejściowe, tzn. należy:

- 9) wskazać na pasku narzędziowym przycisk z symbolem prostokąta,
- 10) zaznaczyć w projekcie obszar przeznaczony na dodawany obiekt,
- 11) dwukrotnie kliknąć na elemencie,
- 12) w oknie konfiguracyjnym obiektu ustalić zmienną odpowiedzialną za zmianę koloru prostokąta – opcja **Change color** (wprowadzanie zmiennej z okna wywołanego przy pomocy F2),
- 13) bez wychodzenia z okna konfiguracyjnego ustawić kolory przygaszony i rozjaśniony (przygaszony jako kolor zwykły tzn. **Color** i rozjaśniony jako **Alarm color**),
- 14) oraz etykietę elementu.







Tak przygotowana wizualizacja jest już gotowa do użycia. Teraz po połączeniu ze sterownikiem i wystartowaniu programu można jej używać do przetestowania programu.

