

## Tworzenie prostego programu w językach ST i LD

### Tworzenie prostego programu w języku ST - wprowadzenie

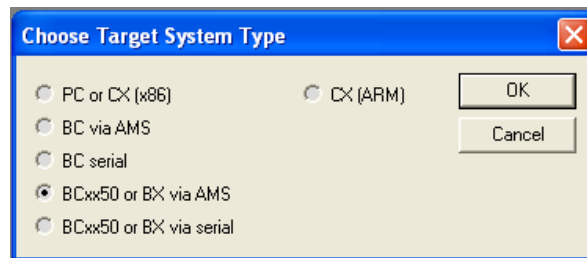
Aby utworzyć program należy uruchomić narzędzie *TwinCAT PLC Control*.



### Wybór obiektu docelowego

W pierwszym etapie należy określić sterownik, dla którego tworzony będzie program oraz sposób komunikacji z urządzeniem.

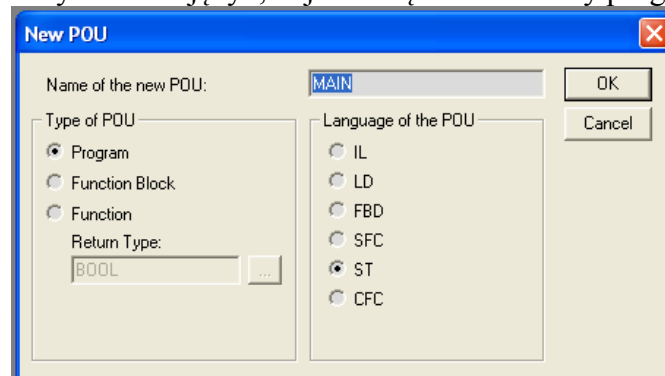
Z menu **File** wybrać opcję **New**, następnie określić docelową platformę w pojawiającym się oknie dialogowym.



W przypadku sterowników:

- BX9000 - jako platformę docelową należy wybrać *BCxx50 or BX via AMS*;
- CX1000, BK9050 - jako platformę docelową należy wybrać *PC or CX (x86)*;
- CX9000, CP6601 - jako platformę docelową należy wybrać *CX (ARM)*.

W kolejnym kroku należy określić język, w jakim będzie tworzony program.



Pakiet *TwinCAT PLC Control* umożliwia programowanie w pięciu językach zgodnych ze standardem IEC 61131-3 oraz w dodatkowym języku CFC. Na tym etapie możliwe jest określenie typu obiektu (*Type of POU – Program organization unit – jednostka organizacyjna*

programu) jaki będzie tworzony: program, blok funkcyjny lub funkcja. Dla potrzeb dalszych przykładów należy wybrać typ obiektu jako *Program* oraz język *ST*.

Dostępne języki programowania

**IL (Instruction List)** – tekstowy język programowania będący odpowiednikiem języka typu assembler, którego zbiór instrukcji obejmuje operacje logiczne, arytmetyczne, relacji, jak również funkcje przerzutników, czasomierzy, liczników itp..

**LD (Ladder Diagram)** – graficzny język programowania, który swoją strukturą przypomina stykowe obwody przekaźnikowe. Dopuszcza się w nim użycie funkcji arytmetycznych, logicznych, porównań i relacji jak również bloków funkcyjnych: przerzutników, czasomierzy, liczników, regulatora PID czy bloków programowych.

**FBD (Function Block Diagram)** – graficzny język programowania będący odpowiednikiem schematów przepływu sygnału dla obwodów logicznych przedstawionych w formie połączonych bramek logicznych oraz bloków funkcyjnych takich jak w języku *LD*.

**ST (Structured Text)** – tekstowy język programowania będący odpowiednikiem języka algorytmicznego wysokiego poziomu (np. C), zawierający struktury programowe takie jak np.

```
If ... then ... else ... end_if  
Case ... of ... end_case  
For ... to ... do ... end_for  
While ... do ... end_while  
Repeat ... until ... end_repeat
```

**SFC (Sequential Function Chart)** - graficzny język pozwalający na opisywanie zadań sterowania sekwencyjnego za pomocą grafów zawierających etapy (kroki) i warunki przejścia (tranzycje) między etapami. Grafy *SFC* obrazują strukturę programu, zaś poszczególne jego elementy są programowane w wybranych językach: *IL*, *LD*, *FBD* lub *ST*.

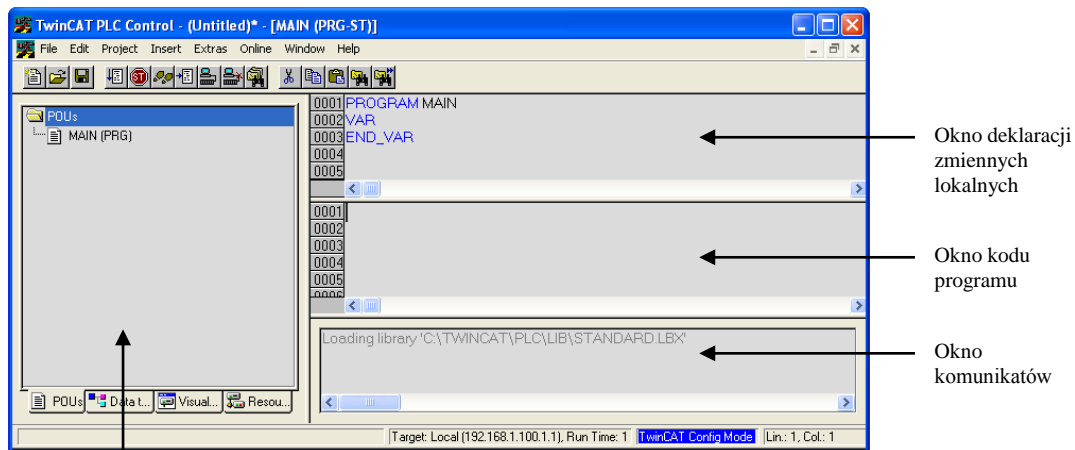
**CFC (Continuous Function Chart)** – graficzny język programowania zbliżony do *FBD*.

### Przykład prostego programu w języku *ST*

**ZADANIE.** Zrealizować w języku *ST* koniunkcję (AND) dwóch symbolicznych zmiennych binarnych o nazwach *input1* oraz *input2* zaś wynik operacji umieścić w symbolicznej zmiennej binarnej *output*.

$$output = input1 \wedge input2$$

Okno *TwinCAT PLC Control* dla języka *ST* pokazano na poniższym rysunku.



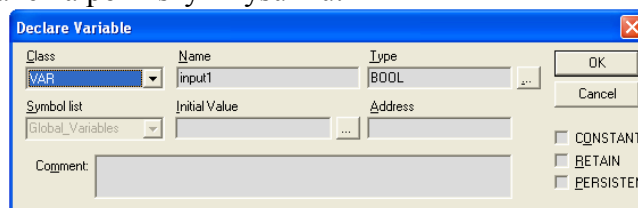
Okno organizacji projektu: programy, bloki funkcyjne, funkcje

W niniejszym przykładzie wykorzystana zostanie instrukcja *IF* języka *ST*, operator koniunkcji logicznej *AND*, operator porównania '=' oraz operator przypisania ':='.

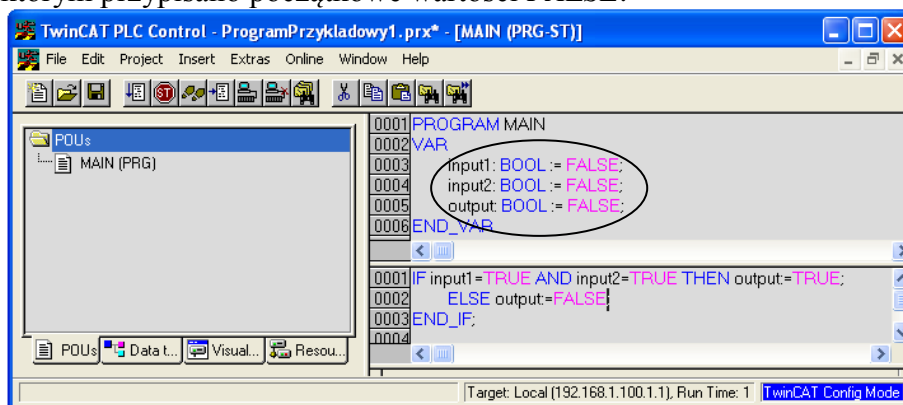
W oknie kodu programu należy wprowadzić program:

```
IF input1=TRUE AND input2=TRUE THEN output:=TRUE;
      ELSE output:=FALSE;
END_IF;
```

Ponieważ zmienne symboliczne *input1*, *input2* oraz *output* nie zostały wcześniej zadeklarowane, po wprowadzeniu pierwszej linii programu pojawi się okno deklaracji zmiennych pokazane na poniższym rysunku.

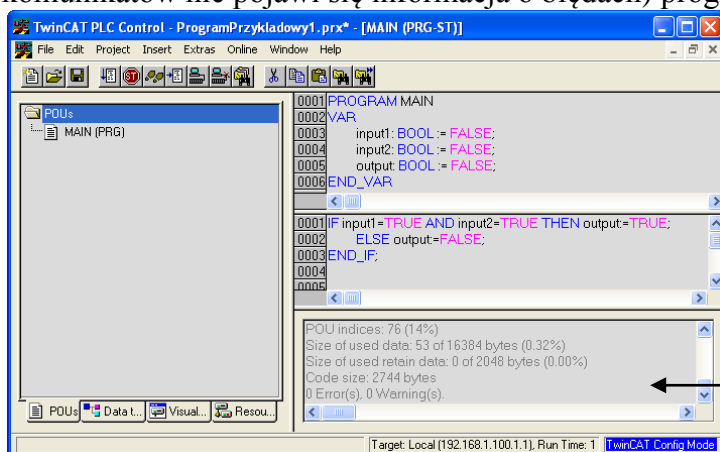


Dla potrzeb niniejszego przykładu dla wszystkich zmiennych należy wprowadzić *FALSE* jako wartość inicjalizującą (*Initial Value*) oraz zatwierdzić przyciskiem *OK*, pozostałe domyślne parametry. W obszarze deklaracji zmiennych lokalnych (rysunek poniżej) pojawi się zapis dotyczący trzech zmiennych binarnych (BOOL): *input1*, *input2* oraz *output*, którym przypisano początkowe wartości *FALSE*.



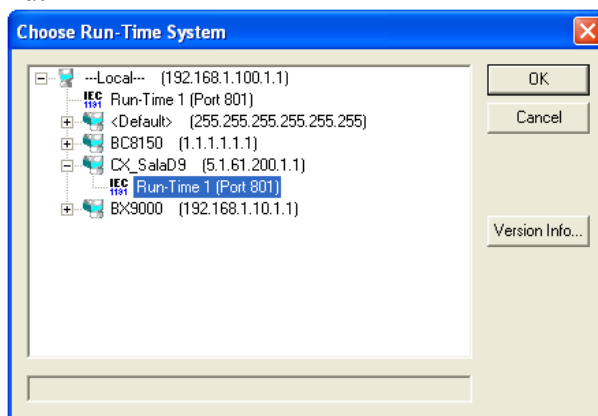
W kolejnym kroku należy zapisać program na dysku wykorzystując menu *File* i opcję *Save*. Po wprowadzeniu kodu programu należy przystąpić do jego kompilacji i konsolidacji

wybierając z menu **Project** opcję **Build**. Gdy proces zakończy się sukcesem (w oknie komunikatów nie pojawi się informacja o błędach) program jest gotowy do testów.

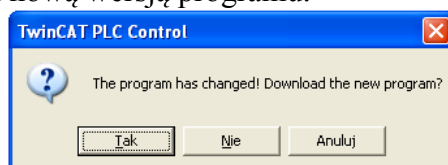


0 Error(s), 0 Warning(s)  
Operacja **Build** zakończona sukcesem

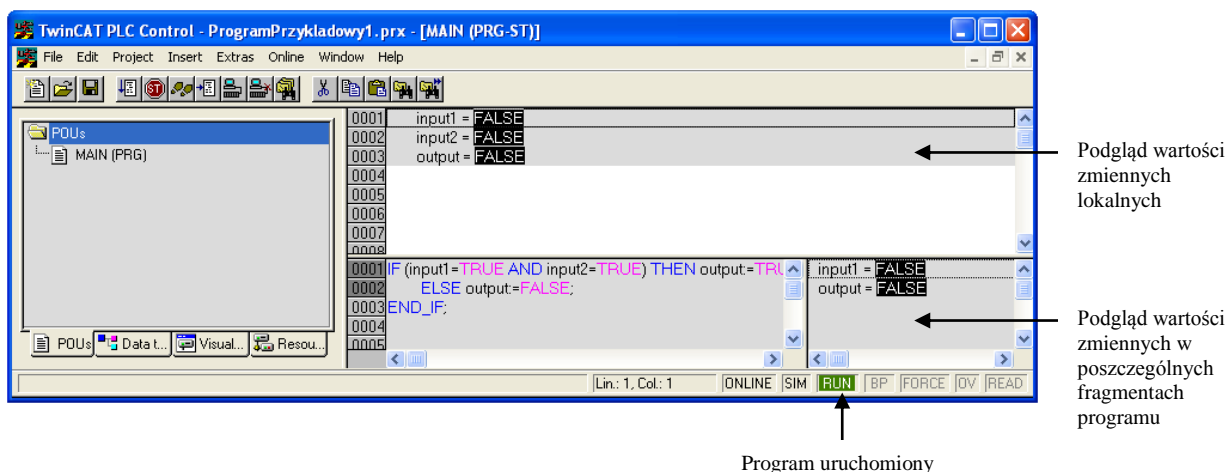
W celu uruchomienia program, z menu **Online** wybrać platformę docelową – opcja **Choose Run-Time System...** i określić odpowiednie urządzenie, a dla niego, po rozwinięciu, wybrać opcję **Run-Time...** Np. dla konfiguracji sterownika CX1000 wybór będzie wyglądał podobnie jak na poniższym rysunku.



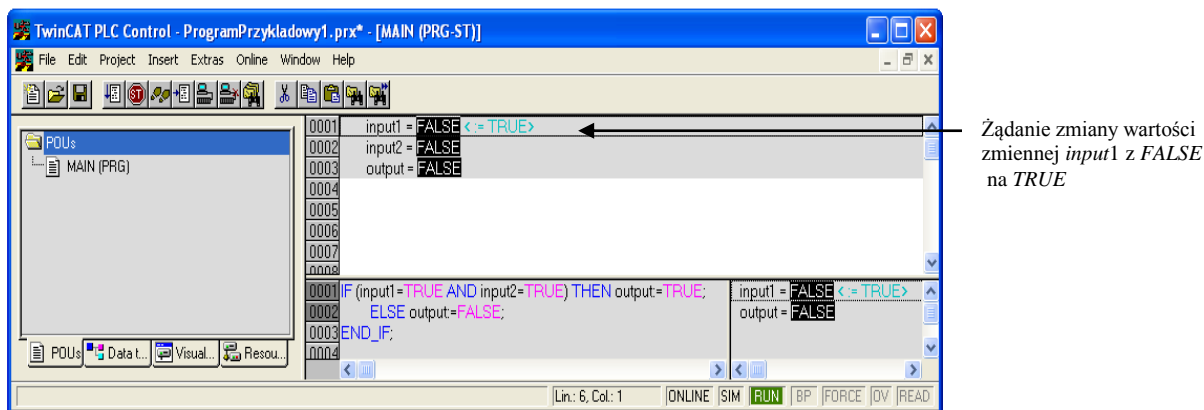
Jeżeli fizyczny sterownik nie jest dostępny, utworzony program może zostać przetestowany w trybie symulacji poprzez wybór z menu **Online** opcji **Simulation Mode** (za wyjątkiem programów przygotowanych dla sterowników z procesorem x86, które należy uruchomić w lokalnym podsystemie **Run-Time**, w trybie soft-PLC). W kolejnym kroku należy zalogować się do sterownika używając opcji **Login** z menu **Online**. Gdy program w sterowniku różni się od aktualnie uruchamianego, system wyświetli okno komunikatu umożliwiające zaprogramowanie sterownika nową wersją programu.



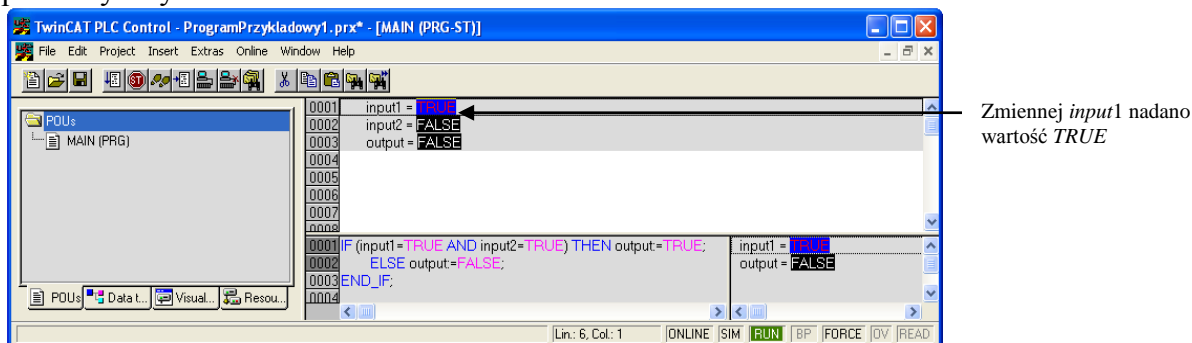
W wyniku naciśnięcia przycisku **Tak** nowy program zostanie przesłany do sterownika. W przypadku, gdy logowanie oraz przesłanie programu zakończy się sukcesem (dotyczy to zarówno pracy bezpośrednio ze sterownikiem jak i trybu symulacji) program **TwinCAT PLC Control** przechodzi do trybu podglądu działania programu. W menu **Online** uaktywniają się opcje takie jak np. **Logout** (pozwala wrócić do trybu edycji programu) oraz **Run** (uruchamia program). Aby obserwować działanie programu należy go uruchomić przy pomocy opcji **Run**, powodzenie operacji jest sygnalizowane pojawieniem się napisu **RUN** na zielonym tle w pasku statusu głównego okna **TwinCAT PLC Control**.



W celu testowania programu, wartości zmiennych mogą być modyfikowane podczas jego działania. Zmianę wartości zmiennej można zrealizować między innymi „klikając” dwukrotnie lewym przyciskiem myszki na nazwie zmiennej zarówno w oknie podglądu wartości zmiennych lokalnych, jak i w oknie podglądu wartości zmiennych w poszczególnych fragmentach kodu programu. W przypadku zmiennych logicznych (BOOL) zmiana wartości pomiędzy TRUE albo FALSE jest podpowiadana przez system automatycznie, w przypadku zmiennych innych typów pojawia się okno dialogowe umożliwiające wprowadzenie danych. Żądanie zmiany wartości zmiennej sygnalizowane jest ciągiem znaków <:=nowa wartość zmiennej> wyświetlanym obok jej nazwy. Aby zmiana nastąpiła należy uruchomić opcję *Write Values* albo *Force Values* z menu *Online*.

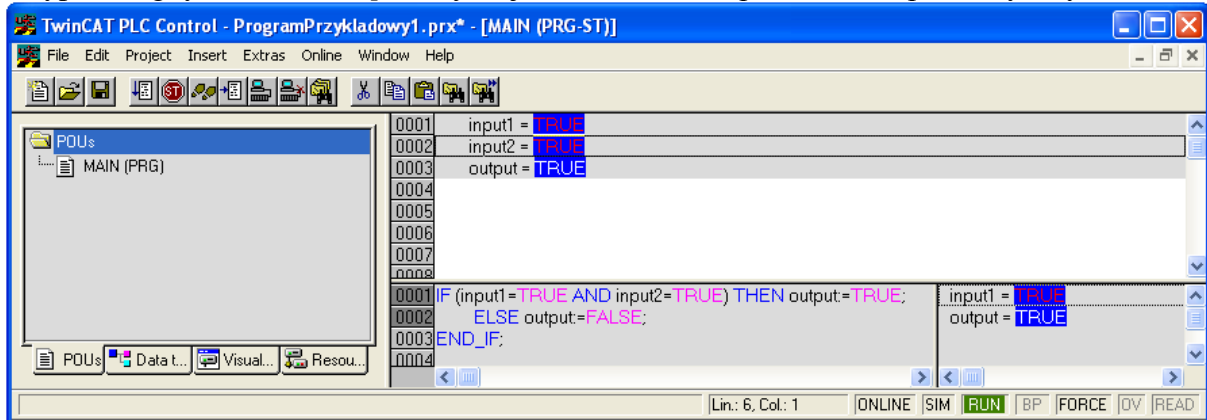


Po zatwierdzeniu zmiany (wykonanie *Write Values* albo *Force Values*) aktualna wartość zmiennej wyświetlana jest w kolorze czerwonym na niebieskim tle jak pokazano na poniższym rysunku.



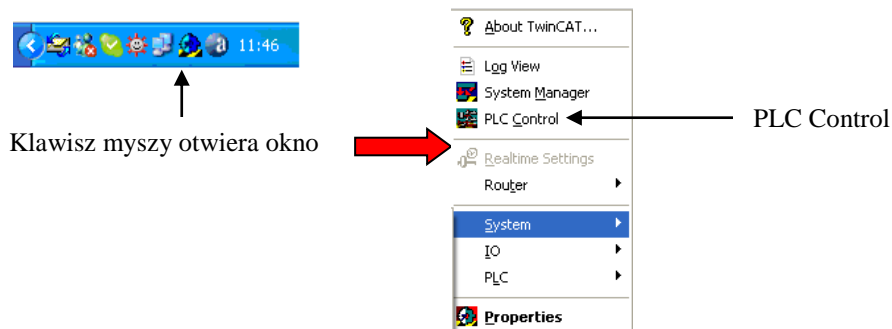
Aby sprawdzić działanie powyższego programu, wartości zmiennych *input1* i *input2* należy modyfikować zgodnie z tablicą prawdy dla koniunkcji logicznej i obserwować wartość zmiennej *output*.

Przypadek, gdy zmienna, *output* uzyskuje wartość *TRUE* pokazano na poniższym rysunku.



## Tworzenie prostego programu w języku LD

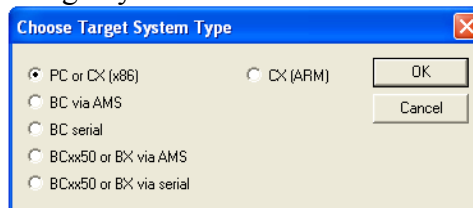
Aby utworzyć program należy uruchomić narzędzie *TwinCAT PLC Control*.



## Wybór obiektu docelowego

W pierwszym etapie należy określić sterownik, dla którego tworzony będzie program oraz sposób komunikacji z urządzeniem.

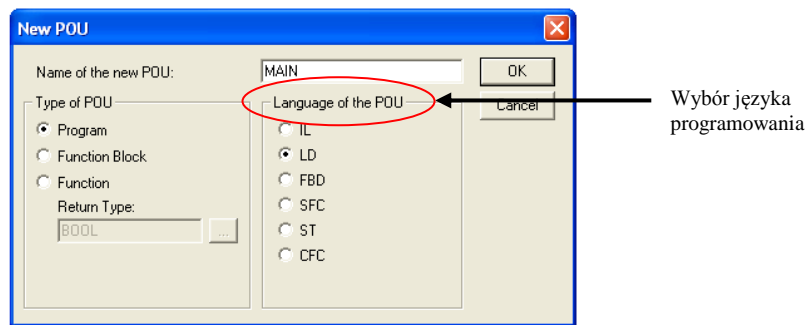
Z menu **File** wybrać opcję **New**, następnie określić docelową platformę w pojawiającym się oknie dialogowym.



W przypadku sterowników:

- BX9000 - jako platformę docelową należy wybrać *BCxx50 or BX via AMS*;
- CX1000, BK9050 - jako platformę docelową należy wybrać *PC or CX (x86)*;
- CX9000, CP6601 - jako platformę docelową należy wybrać *CX (ARM)*.

Podstawowe biblioteki (standard.lbx) związane z wybraną platformą zostaną automatycznie dołączone do projektu. Następnie należy określić język, w jakim będzie tworzony program.



Pakiet *TwinCAT PLC Control* umożliwia programowanie w pięciu językach zgodnych ze standardem IEC 61131-3 oraz dodatkowym językiem CFC. Na tym etapie możliwe jest również określenie typu obiektu (*Type of POU* – *Program Organization Unit* – jednostka organizacyjna programu), jaki będzie tworzony: program (*Program*), blok funkcyjny (*Function Block*) lub funkcja (*Function*). Dla potrzeb dalszych przykładów należy wybrać typ obiektu jako *Program* oraz język *LD*.

#### Dostępne języki programowania

**IL (*Instruction List*)** – tekstowy język programowania będący odpowiednikiem języka typu assembler, którego zbiór instrukcji obejmuje operacje logiczne, arytmetyczne, relacji, jak również funkcje przerzutników, czasomierzy, liczników itp..

**LD (*Ladder Diagram*)** – graficzny język programowania, który swoją strukturą przypomina stykowe obwody przekaźnikowe. Dopuszcza się w nim użycie funkcji arytmetycznych, logicznych, porównań i relacji jak również bloków funkcyjnych: przerzutników, czasomierzy, liczników, regulatora PID czy bloków programowych.

**FBD (*Function Block Diagram*)** – graficzny język programowania będący odpowiednikiem schematów przepływu sygnału dla obwodów logicznych przedstawionych w formie połączonych bramek logicznych oraz bloków funkcyjnych takich jak w języku *LD*.

**ST (*Structured Text*)** – tekstowy język programowania będący odpowiednikiem języka algorytmicznego wysokiego poziomu (np. C), zawierający struktury programowe takie jak np.

```
If ... then ... else ... end_if  
Case ... of ... end_case  
For ... to ... do ... end_for  
While ... do ... end_while  
Repeat ... until ... end_repeat
```

**SFC (*Sequential Function Chart*)** - graficzny język pozwalający na opisywanie zadań sterowania sekwencyjnego za pomocą grafów zawierających etapy (kroki) i warunki przejścia (tranzycje) między etapami. Grafy *SFC* obrazują strukturę programu, zaś poszczególne jego elementy są programowane w wybranych językach: *IL*, *LD*, *FBD* lub *ST*.

**CFC (*Continuous Function Chart*)** – graficzny język programowania zbliżony do *FBD*.

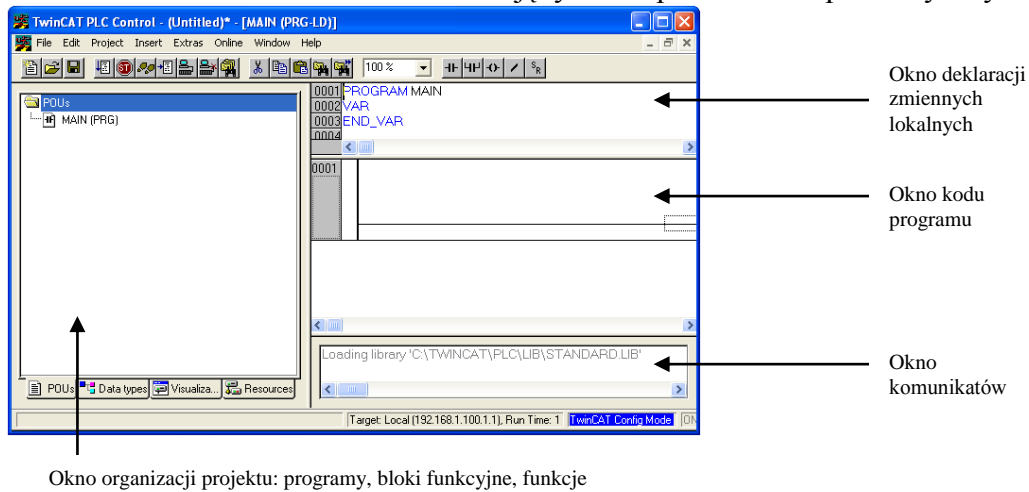
#### Przykład prostego programu w języku *LD*

**ZADANIE.** Zrealizować w języku *LD* koniunkcję (AND) dwóch symbolicznych zmiennych binarnych o nazwach *Input1* oraz *Input2* zaś wynik operacji umieścić w

symbolicznej zmiennej binarnej *Output*. Należy wykorzystać dwa styki zwierne oraz jedną cewkę zwykłą.

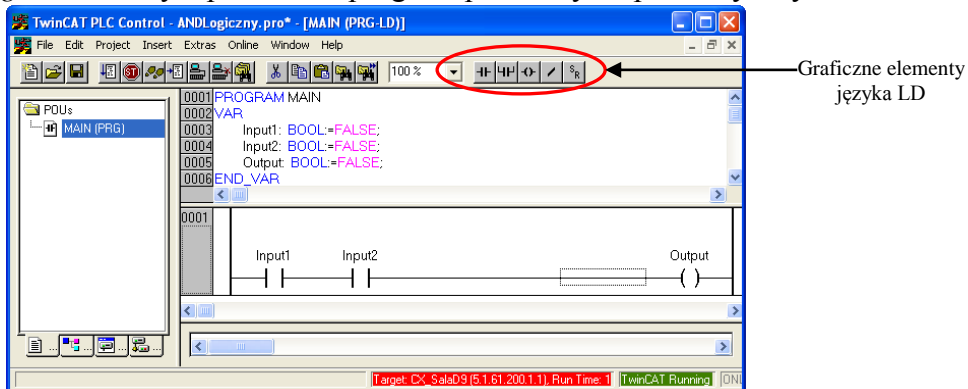
$$Output = Input1 \wedge Input2$$

Okno *TwinCAT PLC Control* dla języka *LD* pokazano na poniższym rysunku.



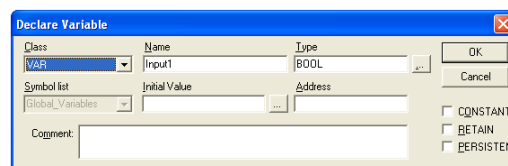
Okno organizacji projektu: programy, bloki funkcyjne, funkcje

W niniejszym przykładzie wykorzystana zostanie funkcja *AND*. W *oknie kodu programu* należy wprowadzić program pokazany na poniższym rysunku.



Aby wstawić do programu graficzne elementy języka *LD*, można wykorzystać pasek narzędzi zaznaczony na powyższym rysunku czerwoną elipsą, bądź skorzystać z menu kontekstowego prawego klawisza myszy dla okna kodu programu.

Po wstawieniu do programu styku (*Contact*) zwiernego  $\text{--|T}$  nazwę zmiennej z nim związanej należy wprowadzić z klawiatury, zastępując ciąg znaków „???” (umiejscowiony powyżej symbolu graficznego styku) nazwą zmiennej. Ponieważ zmienne symboliczne *Input1*, *Input2* oraz *Output* nie zostały wcześniej zadeklarowane, po wprowadzeniu nazwy zmiennej pojawi się okno deklaracji zmiennych pokazane na poniższym rysunku.

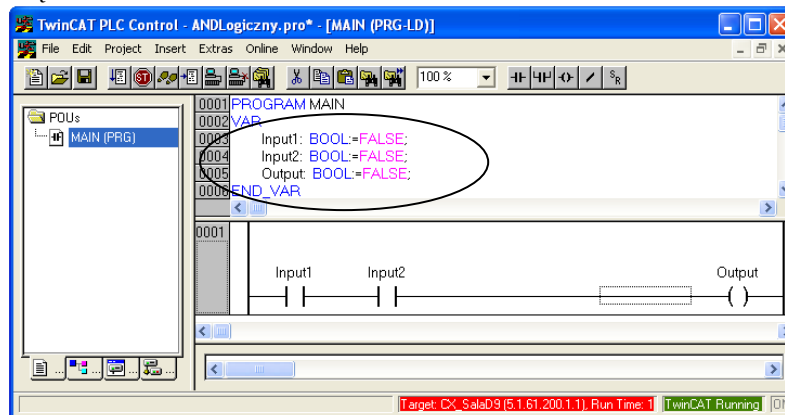


Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Delete	Del
Network (before)	
Network (after)	Ctrl+T
Contact	Ctrl+K
Parallel Contact	Ctrl+R
Function Block ...	Ctrl+B
Coil	Ctrl+L
Box with EN	
Insert at blocks	
Jump	
Return	
Comment	
Negate	Ctrl+N
Set/Reset	
Zoom	Alt+Enter
Open instance	

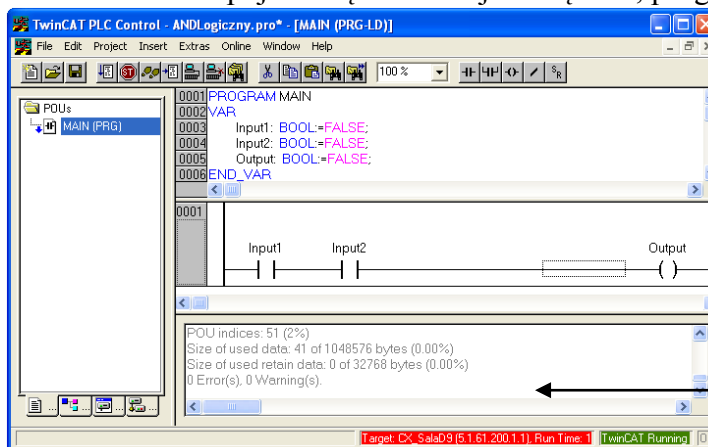
Dla potrzeb niniejszego przykładu dla wszystkich zmiennych należy wprowadzić *FALSE* jako wartość inicjalizującą (*Initial Value*) oraz zatwierdzić (przyciskiem *OK*) pozostałe domyślne parametry. W obszarze deklaracji zmiennych lokalnych pojawi się zapis



dotyczący trzech zmiennych binarnych (BOOL): *Input1*, *Input2* oraz *Output*, którym przypisano początkowe wartości *FALSE*.

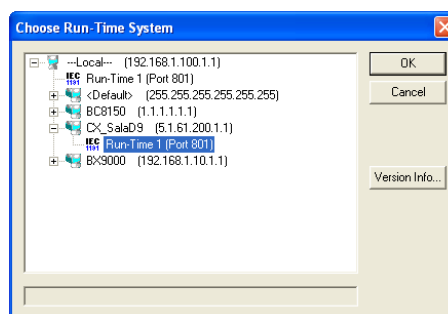


W dalszej kolejności należy zapisać program na dysku wykorzystując menu *File* i opcję *Save*. Po wprowadzeniu kodu programu można przystąpić do jego kompilacji i konsolidacji wybierając z menu *Project* opcję *Build*. Gdy proces zakończy się sukcesem, w oknie komunikatów nie pojawi się informacja o błędach, program jest gotowy do testów.

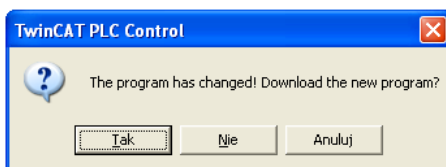


0 Error(s), 0 Warning(s)  
Operacja Build  
zakończona sukcesem

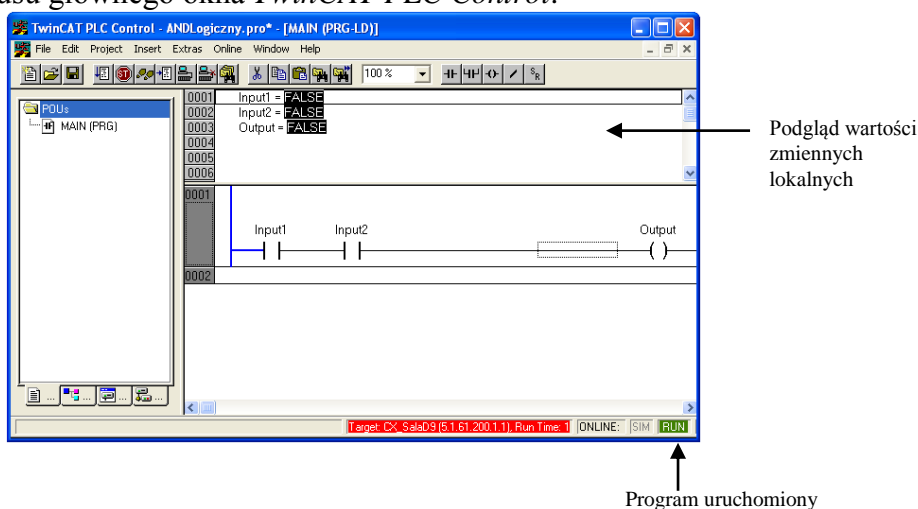
W celu uruchomienia program, w menu *Online* należy wybrać platformę docelową – opcja *Choose Run-Time System...* i określić odpowiednie urządzenie - opcja odpowiadająca danemu sterownikowi (nazwa i adres).



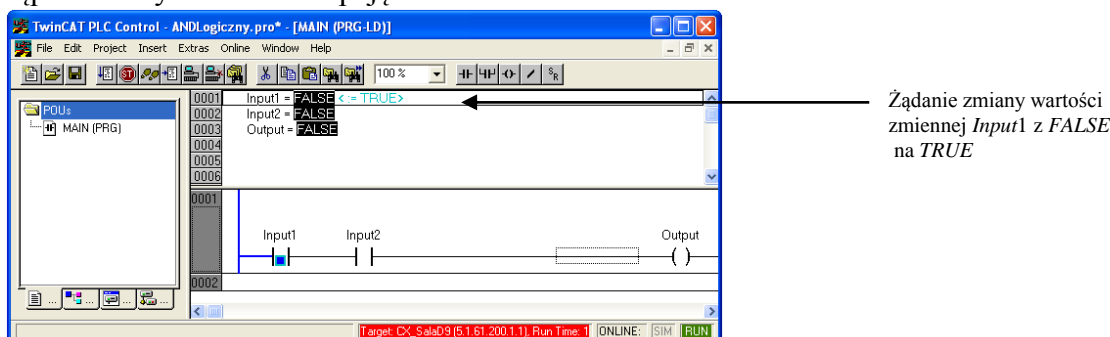
Jeżeli fizyczny sterownik nie jest dostępny, utworzony program można zostać przetestowany w trybie symulacji poprzez wybór z menu *Online* opcji *Simulation Mode* (za wyjątkiem programów przygotowanych dla sterowników z procesorem x86, które należy uruchomić w lokalnym podsystemie *Run-Time*, w trybie soft-PLC). W kolejnym kroku należy zalogować się do sterownika używając opcji *Login* z menu *Online*. Gdy program w sterowniku różni się od aktualnie uruchamianego, system wyświetli okno komunikatu umożliwiające zaprogramowanie sterownika nową wersją programu.



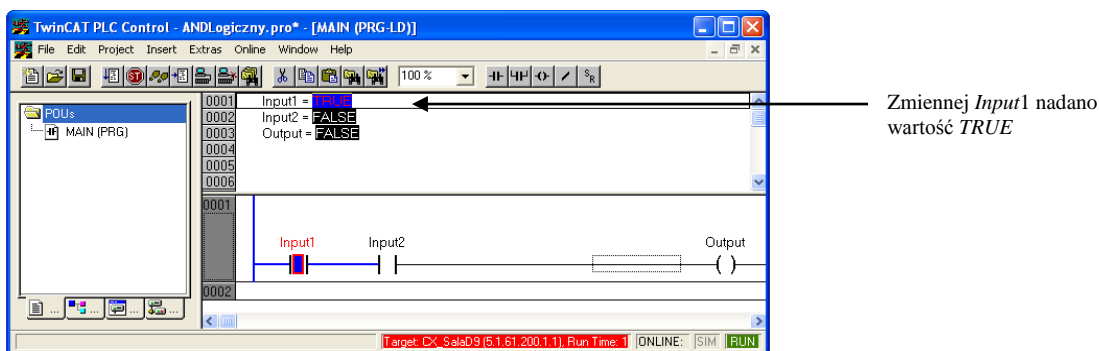
W wyniku naciśnięcia przycisku **Tak** nowy program zostanie przesłany do sterownika. W przypadku, gdy logowanie oraz przesłanie programu zakończy się sukcesem (dotyczy to zarówno pracy bezpośrednio ze sterownikiem jak i trybu symulacji) program *TwinCAT PLC Control* przechodzi do trybu podglądu działania programu. W menu **Online** uaktywniają się opcje takie jak np. **Logout** (pozwala wrócić do trybu edycji programu) oraz **Run** (uruchamia program). Aby obserwować działanie programu należy go uruchomić przy pomocy opcji **Run**, powodzenie operacji jest sygnalizowane pojawieniem się napisu *RUN* na zielonym tle w pasku statusu głównego okna *TwinCAT PLC Control*.



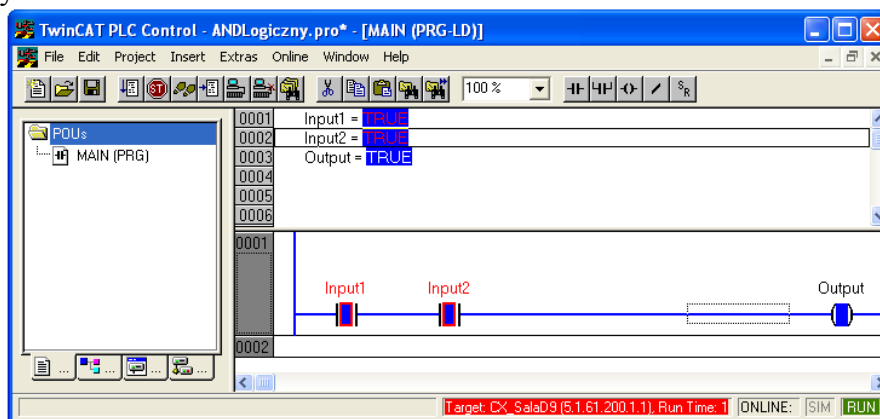
W celu testowania programu, wartości zmiennych mogą być modyfikowane podczas jego działania. Zmianę wartości zmiennej można zrealizować między innymi „klikając” dwukrotnie lewym przyciskiem myszki na nazwie zmiennej zarówno w oknie podglądu wartości zmiennych lokalnych jak i w oknie programu. W przypadku zmiennych logicznych (BOOL) zmiana wartości pomiędzy TRUE albo FALSE jest podpowiadana przez system automatycznie, w przypadku zmiennych innych typów pojawia się okno dialogowe umożliwiające wprowadzenie danych. Żądanie zmiany wartości zmiennej sygnalizowane jest ciągiem znaków <:=nowa wartość zmiennej> wyświetlanym obok jej nazwy. Aby zmiana nastąpiła należy uruchomić opcję **Write Values** albo **Force Values** z menu **Online**.



Po zatwierdzeniu zmiany (**Write Values** albo **Force Values**) aktualna wartość zmiennej w oknie deklaracji zmiennych jest sygnalizowana czerwonym kolorem na niebieskim tle (jak pokazano na poniższym rysunku) bądź odpowiednim symbolem graficznym w oknie programu.

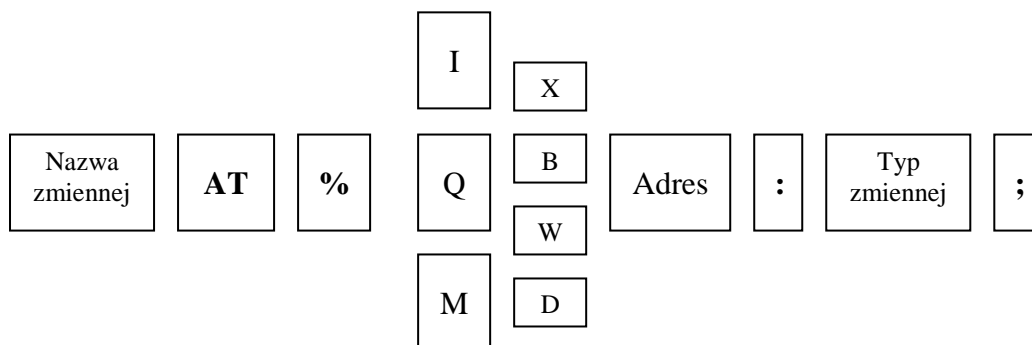


Aby sprawdzić działanie powyższego programu, wartości zmiennych *Input1* i *Input2* należy modyfikować zgodnie z tablicą prawdy dla koniunkcji logicznej i obserwować wartość zmiennej *output*. Przypadek, gdy zmienna, *Output* uzyskuje wartość *TRUE* pokazano na poniższym rysunku.



### Powiązanie symbolicznych zmiennych z fizycznym obszarem pamięci urządzenia

Aby symboliczna zmienna z programu PLC mogła być powiązana z fizycznym wejściem, wyjściem bądź obszarem pamięci sterownika musi być zadeklarowana jako tzw. *zmienna adresowana*. W celu ulokowania zmiennej w odpowiednim obszarze pamięci (obszar wejść, wyjść, przestrzeń flag) w jej deklaracji należy użyć słowa kluczowego **AT**. Sposób deklarowania *zmiennej adresowanej* pokazano na poniższym schemacie



gdzie symbole I ,Q, M określają obszar pamięci w jakim ma być ulokowana zmienna:

- I – obszar zmiennych wejściowych,
- Q – obszar zmiennych wyjściowych,
- M – obszar przestrzeni flag (fizyczny obszar pamięci sterownika o podanym adresie).

Modyfikatory X, B, W, D określają rozmiar zmiennej:

- X - bit (bit),
- B - bajt (byte - 8 bitów),
- W - słowo (word - 16 bitów),
- D - podwójne słowo (double word - 32 bity).

**Adres** określa fizyczną lokalizację zmiennej adresowanej w danym obszarze pamięci. W przypadku zmiennych bitowych adres jest tworzony jako ciąg liczb całkowitych bez znaku, oddzielonych kropkami np. 1.3 co oznacza bit o indeksie numer 3 (liczone od 0) w bajcie 1. Adresy zmiennych bitowych ulokowanych w przestrzeniach wejść (I) i wyjść (Q) mogą być identyczne, gdyż są przechowywane w różnych obszarach pamięci. Dla zmiennych B, W, D - **Adres** jest reprezentowany przez liczbę całkowitą bez znaku.

**Typ zmiennej** – określa typ zmiennej np. BOOL, BYTE, INT, REAL, ARRAY pozwalający na prawidłową jej interpretację przez operatory danego języka programowania.

W typowych przypadkach, zmienne reprezentujące fizyczne wejścia albo wyjścia urządzenia powinny być ulokowane w odpowiednim obszarze pamięci bez dokładnego wyszczególnienia **Adresu**. Powiązanie danej zmiennej z fizycznym wyjściem/wyjściem jest realizowane w pakiecie *TwinCAT System Manager*. W omawianym przypadku **Adres** w deklaracji zmiennej zastępowany jest znakiem ‘\*’ np. `output AT %Q*:BOOL`; co oznacza, że zmienna *output* jest ulokowana w obszarze wyjść zaś jej szczegółowy adres zostanie określony w *TwinCAT System Manager*.

Aby powiązać zmienną *output* z fizycznym wyjściem urządzenia należy zmienić jej deklarację w następujący sposób

```
output AT %Q* : BOOL := FALSE;
```

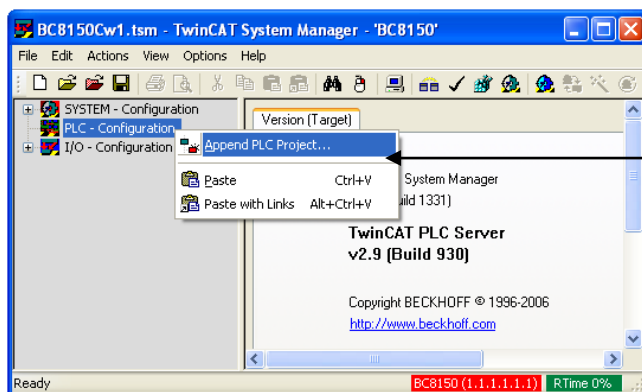
aby powiązać zmienne *input1* i *input2* z fizycznymi wejściami sterownika należy zmienić ich deklarację w następujący sposób

```
input1 AT %I* : BOOL := FALSE;
```

```
input2 AT %I* : BOOL := FALSE;
```

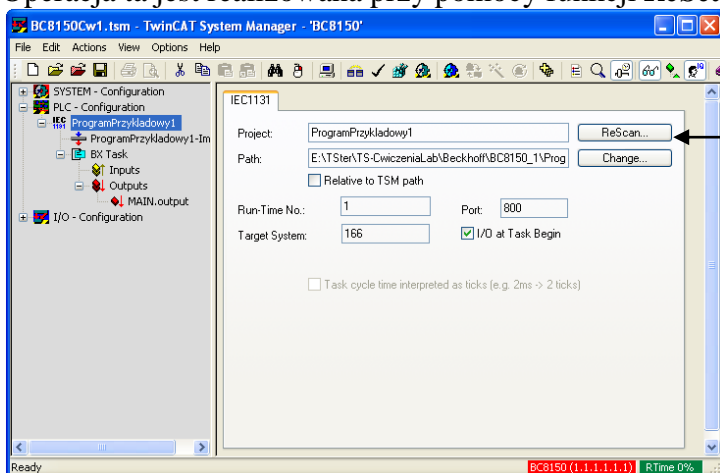
W kolejnym kroku należy dokonać ponownej kompilacji programu (menu **Project** opcja **Rebuild all**). W wyniku poprawnego utworzenia programu (w lokalizacji, w której zapisano projekt na dysku) zostanie utworzony plik z rozszerzeniem *tpy*.

Aby powiązać zmienną *output* z fizycznym wyjściem sterownika należy uruchomić projekt *TwinCAT System Manager* utworzony dla aktualnej konfiguracji sprzętowej urządzenia i połączyć z nim utworzony projekt PLC. Operacja ta jest realizowana przy pomocy opcji **Append PLC Project...**, dostępnej w *TwinCAT System Manager* dla gałęzi **PLC – Configuration** w menu pomocniczym wywoływanym przy pomocy prawego klawisza myszy.



Połączenie z projektem PLC

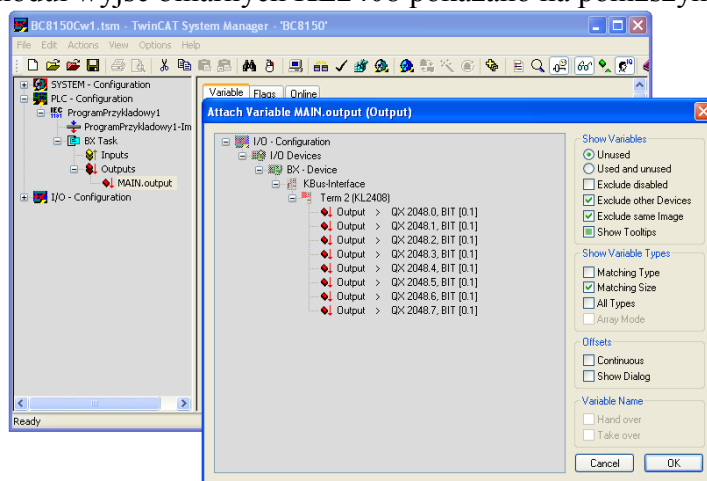
Wykonanie powyższej operacji umożliwia programowi *TwinCAT System Manager* dostęp do zmiennych adresowanych zadeklarowanych w projekcie PLC. Jak pokazano na poniższym rysunku dla rozważanego programu dostępna jest jedna zmienna o nazwie *MAIN.output*, gdzie *MAIN* określa nazwę programu w którym zmienna *output* została zadeklarowana. W przypadku modyfikacji deklaracji zmiennych adresowanych w projekcie PLC, należy w pakiecie *TwinCAT System Manager* zaktualizować dane dotyczące danego projektu PLC. Operacja ta jest realizowana przy pomocy funkcji *ReScan....*



Aktualizacja projektu PLC

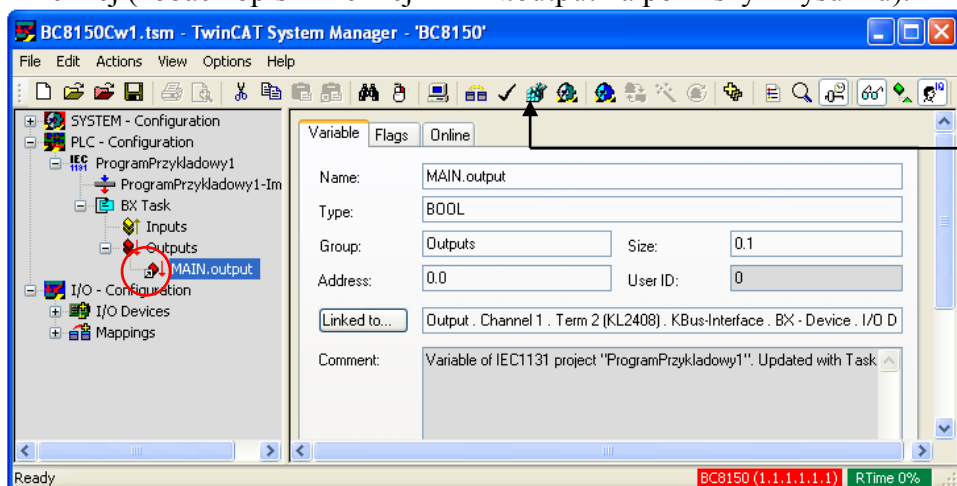
W wyniku dwukrotnego kliknięcia lewym klawiszem myszy na wybranej nazwie zmiennej, uzyskuje się dostęp do fizycznych wejść/wyjść urządzenia, które mogą być z daną zmienną skojarzone.

Przykładową możliwość dowiązania zmiennej *output* do fizycznych wyjść urządzenia wyposażonego w moduł wyjść binarnych KL2408 pokazano na poniższym rysunku.




Klikając dwukrotnie lewym klawiszem myszy na wybranym wyjściu modułu KL2408 dokonujemy jego powiązania ze zmienną *output*. Informacja o powiązaniu zmiennej z

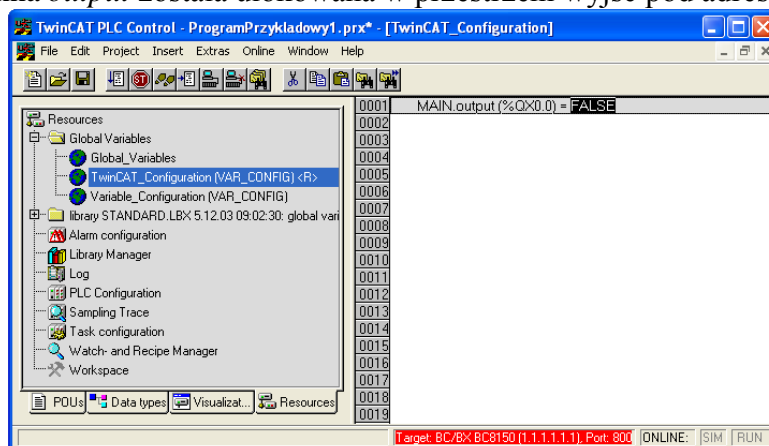
fizycznym wyjściem jest sygnalizowana, między innymi, znakiem strzałki przy nazwie zmiennej (zobacz opis zmiennej MAIN.output na poniższym rysunku).



Aktualizacja konfiguracji w sterowniku

W kolejnym kroku utworzone powiązania należy przesłać do sterownika przy pomocy przycisku **Activate configuration** .

Po wykonaniu powyższych czynności należy powrócić do pakietu *TwinCAT PLC Control* i uruchomić program. Operacja ta jest realizowana identycznie jak w przypadku trybu symulacji. Należy pamiętać, aby poprawnie określić system docelowy (**Choose Run-Time System...**) oraz wyłączyć tryb symulacji (**Simulation Mode**) (opcje menu **Online**). Po poprawnym przesłaniu programu do sterownika w zakładce *Resources* w folderze *Global Variables* pojawi się pole *TwinCAT\_Configuration*, które zawiera informacje o powiązaniach zmiennych adresowanych z fizyczną pamięcią urządzenia. W niniejszym przypadku zmienna *output* została ulokowana w przestrzeni wyjść pod adresem QX0.0.



Uruchomienie i testowanie programu może być realizowane dla programu działającego w sterowniku w analogiczny sposób jak dla trybu symulacji.