



# Podstawy obsługi programu TwinCAT PLC Control

Część 2

Wersja dokumentacji 1.1

Warszawa 2011

<b>WSTĘP</b> .....	<b>3</b>
<b>1 TWORZENIE NOWEGO PROJEKTU</b> .....	<b>4</b>
1.1 TYP PROJEKTU .....	4
1.2 TYP OBIEKTU .....	5
1.3 JĘZYKI PROGRAMOWANIA .....	6
1.3.1 <i>IL – Instruction List</i> .....	6
1.3.2 <i>ST – Structured Text</i> .....	7
1.3.3 <i>LD – Ladder Diagram</i> .....	8
1.3.4 <i>FBD - Function Block Diagram</i> .....	9
1.3.5 <i>SFC – Sequential Function Chart</i> .....	10
1.3.6 <i>CFC – Continous Function Chart</i> .....	11
<b>2 GŁÓWNE OKNO PROGRAMU</b> .....	<b>12</b>
2.1 PASEK ZADAŃ, IKONY SKRÓTÓW. ....	12
2.2 MENAGER ORGANIZACJI PROJEKTU .....	14
2.3 OKNO DEKLARACJI ZMIENNYCH .....	20
2.4 OKNO KODU PROGRAMU .....	20
2.5 OKNO INFORMACYJNE .....	21
<b>3 TASK(WĄTEK)</b> .....	<b>22</b>
<b>4 EDYCJA PROJEKTU</b> .....	<b>24</b>
4.1 TWORZENIE NOWEGO PROJEKTU .....	24
4.2 OTWIERANIE PROJEKTU .....	24
4.3 ZAPISYWANIE PROJEKTU .....	25
<b>5 DEKLARACJA ZMIENNYCH</b> .....	<b>26</b>
5.1 WSTĘP .....	26
5.2 STRUKTURA DEKLARACJI .....	27
5.3 KREATOR DEKLARACJI ZMIENNEJ .....	29
5.4 TYPY STANDARDOWE .....	31
5.5 TYPY DEKLAROWANE PRZEZ UŻYTKOWNIKA .....	31
5.5.1 <i>Deklaracja struktury</i> .....	32
5.5.2 <i>Deklaracja typu ENUM</i> .....	33
5.5.3 <i>Deklaracja tablicy</i> .....	34
<b>6 URUCHOMIENIE PROGRAMU, LINKOWANIE ZMIENNYCH</b> .....	<b>36</b>
6.1 KOMPILACJA PROGRAMU .....	36
6.2 LINKOWANIE ZMIENNYCH .....	36
6.3 URUCHAMIANIE PROGRAMU .....	37
<b>7 DODATEK</b> .....	<b>39</b>
7.1 DODATKOWA POMOC .....	39
7.2 AKCJE (ACTION) .....	39
7.3 OPERACJE NA STRINGACH .....	39
7.4 ZNAKI SPECJALNE .....	40
7.5 SKRÓTY KLAWISZOWE .....	41
<b>8 PRZEGLĄD NAJWAŻNIEJSZYCH OPCJI PROGRAMU TWINCAT PLC CONTROL</b> .....	<b>42</b>
8.1 MENU FILE .....	42
8.2 MENU EDIT .....	42
8.3 MENU PROJECT .....	43
8.4 MENU ONLINE .....	47

## Wstęp

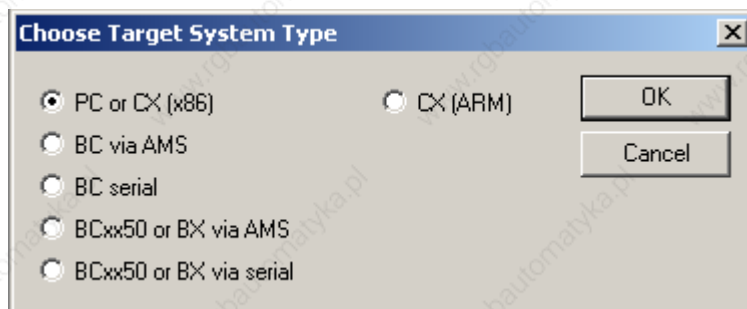
Niniejsza dokumentacja to jedynie krótki opis najważniejszych funkcji programu TwinCAT PLC Control. Niektóre zagadnienia zostały wyjaśnione skrótowo, a niektóre problemy znacznie uproszczone. Pełna dokumentacja dostępna jest w Beckhoff Information System oraz na [infosys.beckhoff.com](http://infosys.beckhoff.com).

TwinCAT PLC Control jest narzędziem programistycznym pracującym w środowisku Windows NT/2000/XP/Vista/7 umożliwiającym tworzenie, zarządzanie oraz edytowanie programów PLC w językach normy IEC 61131-3: IL, LD, FBD, CFC, SFC, ST. Pozwala on na tworzenie funkcji, bloków funkcyjnych i programów, które można zapisać w formie projektu lub biblioteki. Poprawność działania algorytmów można sprawdzić dzięki wbudowanej opcji symulacji i wielu funkcjom diagnostycznym (debugger). TwinCAT PLC Control pozwala również na utworzenie wizualizacji wyświetlanej lokalnie lub poprzez www. Konfigurację sprzętową realizuje się za pomocą oprogramowania TwinCAT System Manager, które zostało opisane w oddzielnej dokumentacji.

# 1 Tworzenie nowego projektu

## 1.1 Typ projektu

Nowy projekt tworzymy wybierając z menu **File** → **New**. W oknie **Choose Target System Type**, wskazujemy typ sterownika, na który będzie wysłany program.

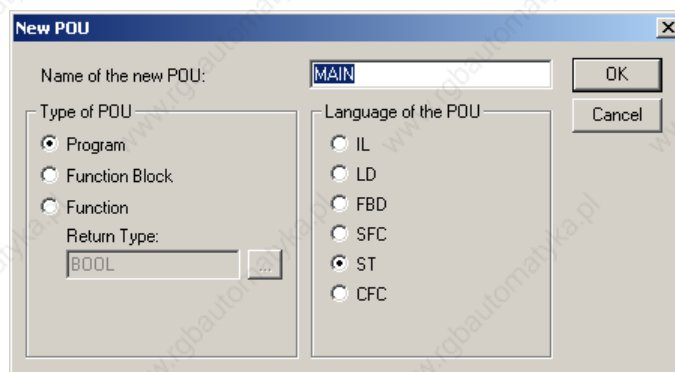


- PC or CX (x86) – opcja używana do połączenia wszystkich komputerów klasy PC oraz sterowników z architekturą procesora x86 np. sterowniki z serii CX10xx, panele z serii CP67xx, wszystkie komputery.
- CX (ARM) – opcja używana do połączenia sterowników z architekturą procesora ARM np. z serii CX90xx lub panele CP66xx.
- BCxx50 or BX via AMS – do połączenia wykorzystywany jest protokół komunikacyjny ADS. Pozwala on nawiązać połączenie ze sterownikiem wykorzystując interfejsy: Ethernet lub port szeregowy. Opcja może być używana do sterowników z serii BCxx50, BCxx20 oraz BX.
- BCxx50 or BX via serial – opcja używana do łączenia ze sterownikami przez port szeregowy. Sposób połączenia wykorzystywany dawniej, wymagał określenia adresów wejść/wyjść. Niezalecany w nowych aplikacjach. Opcja może być używana do sterowników z serii BCxx50, BCxx20 oraz BX.
- BC via serial i BC via AMS analogicznie jak wyżej dla sterowników BCxx00

Uwaga! Opcję wyboru typu sterownika można zmienić również po utworzeniu projektu w zakładce **Resources** → **PLC Configuration**.

## 1.2 Typ obiektu

Po potwierdzeniu wyboru typu sterownika pojawia się okno **New POU** (Program Organisation Unit). Okno pozwala utworzyć program główny oraz wybrać dla niego język programowania.



Jednostką organizacyjną może być funkcja, blok funkcyjny lub program.

- **Program** może być wywoływany bezpośrednio przez task (opisany w rozdziale 4. **Task** ) lub w innym programie. Nie wymaga deklaracji w polu zmiennych – zawiera jedną instancję (każde wywołanie alokuje ten sam obszar pamięci), dlatego powinien być wywołany raz w cyklu i korzystać tylko ze zmiennych lokalnych bądź globalnych. W programie mogą być wywoływane funkcje, bloki funkcyjne i inne programy. Wartości zmiennych wewnętrznych widoczne są w trybie online.
- **Function block (blok funkcyjny)** może być wywołany w innym bloku lub programie. Sam może wywołać blok funkcyjny bądź funkcję. Każda instancja alokuje nowy obszar pamięci, w związku z tym wartości zmiennych przechowywane są do następnego wywołania bloku funkcyjnego i można podejrzeć je w trybie online. Blok funkcyjny może posiadać dowolną liczbę wejść i wyjść. Przykłady: R\_Trig (wykrycie zbocza narastającego), FB\_BasicPID (regulator PID).
- **Function (funkcja)** może wywoływać tylko inne funkcje, a sama może być wywołana wszędzie. Może mieć wiele wejść, a tylko jedno wyjście – zwracające jej wartość. Nie wymaga deklaracji, ponieważ posiada jedną instancję - alokuje zawsze ten sam obszar pamięci. W związku z tym nie przechowuje wartości zmiennych wewnętrznych i nie możliwe jest ich podejrze w trybie online. Przykłady: AND (iloczyn logiczny), REAL\_TO\_INT (funkcja konwersji typów), SIN (sinus).

### 1.3 Języki programowania

Języki programowania dostępne w TwinCAT są zgodne ze standardem IEC 61131-3, który jest międzynarodową normą dotyczącą programowania sterowników PLC. Zaletą tego standardu jest jego uniwersalność. Ta sama struktura tworzenia programów stosowana jest przez wielu producentów. Dostępne są dwa tekstowe oraz cztery graficzne języki programowania.

#### - Tekstowe:

- IL – Instruction List
- ST – Structured Text

#### - Graficzne

- LD – Ladder Diagram
- FBD – Function Block Diagram
- SFC – Sequential Function Chart
- CFC – Continous Function Chart

#### 1.3.1 IL – Instruction List

Jest to język niskiego poziomu, strukturą języka podobny do assemblera. Kod składa się z szeregu instrukcji, z których każda rozpoczyna się nową liniijką i w zależności od rodzaju operacji zawiera jeden lub więcej operatorów oddzielonych przecinkami. Przed instrukcją może znajdować się nazwa etykiety zakończona dwukropkiem. Linie mogą być oddzielone pustymi wierszami a komentarze mogą znajdować się w linii za instrukcją.

Przykład:

```
LD 17
ST lint (* komentarz *)
GE 5
JMPC next
LD idword
EQ instruct.sdword
STN test
next:
```

### 1.3.2 ST – Structured Text

Język ten jest podobny do języków wyższego rzędu (np. C, Pascal). Możliwe jest stosowanie instrukcji warunkowych oraz pętli (IF..THEN..ELSE; WHILE..DO; FOR itp.) Język ST pozwala na konstruowanie złożonych wyrażeń. Wyrażenia składają się z operatorów i argumentów. Wyrażenie zwraca wartość po zakończeniu obliczeń. Kolejność obliczania jest realizowana na podstawie ważności operatorów. Najpierw wykona się operacja najsilniejszego operatora a na końcu operatora najsłabszego. W przypadku wystąpienia kilku równoważnych operatorów, operacje zostaną wykonane od lewej do prawej. Blok funkcyjny wywoływany jest poprzez napisanie nazwy zmiennej a w nawiasie wypisanie jego zmiennych wejściowych lub wyjściowych. Poniżej zostało przedstawiona przykładowa deklaracja oraz wywołanie bloku funkcyjnego.

```

0002 VAR
0003     tonTimer: TON;
0004     bStart: BOOL;
0005     bOut AT %Q*: BOOL;
0001 tonTimer(IN:=bStart, PT:=#10s, Q=>bOut, ET=>timET);
0002
0003

```

Kod wykonywany jest od góry do dołu, od lewej do prawej, zachowując ważność operatorów.

przypisanie

```

0001 rTemp:= rDane;
0002 iLicznik := iLicznik + 1;
0003 rWynik:=SIN(rSygnal);
0004

```

IF

```

0001 IF rSygnal > 100 THEN
0002     rWyjscie := rSygnal;      (*kod jest wykonywany *)
0003                             (* jeśli warunek rSygnal > 100 jest spełniony *)
0004
0005 ELSIF rSygnal < 10 THEN      (* jeśli warunek rSygnal > 100 nie jest spełniony *)
0006     rWyjscie := 100;         (* sprawdzany jest warunek rSygnal < 10 *)
0007                             (* jeśli jest spełniony wykonywany jest kod po then *)
0008 ELSE
0009     rWyjscie := rWyjscie + 1; (* w pozostałych przypadkach wykonywany *)
0010                             (* jest kod po else *)
0011 END_IF

```

## CASE

```

0001 (* instrukcje będą wykonywane *)
0002 (* w zależności od stanu zmiennej iStep *)
0003 CASE iStep OF
0004 1: bStart := TRUE;
0005 2: bStart := FALSE;
0006    bEnd := TRUE;
0007 3: iTemp := 30;
0008    bStart := FALSE;
0009 ELSE
0010    iTemp := 0;
0011    bStart := FALSE;
0012    bEnd := FALSE;
0013 END_IF

```

## FOR

```

0001 FOR iIndex:=0 TO 10 DO
0002   IF aiTablica[iIndex] = 70 THEN
0003     aiTablica[iIndex] := aiTablica[iIndex] - 20;
0004   END_IF;
0005 END_FOR;
0006

```

## WHILE

```

0001 (* jeśli warunek iCounter <= 100 jest spełniony *)
0002 (* będzie ciągle wykonywany kod znajdujący się *)
0003 (* w pętli while *)
0004 WHILE iCounter <= 100 DO
0005   iCounter := iCounter+2;
0006 END_WHILE;

```

## REPEAT

```

0001 (* jeśli warunek iCounter <= 100 OR NOT bEnd *)
0002 (* jest spełniony, kod znajdujący się *)
0003 (* w pętli REPEAT będzie wykonywany *)
0004 REPEAT
0005   iCounter:=iCounter+2;
0006 UNTIL iCounter <= 100 OR NOT bEnd
0007 END_REPEAT;

```

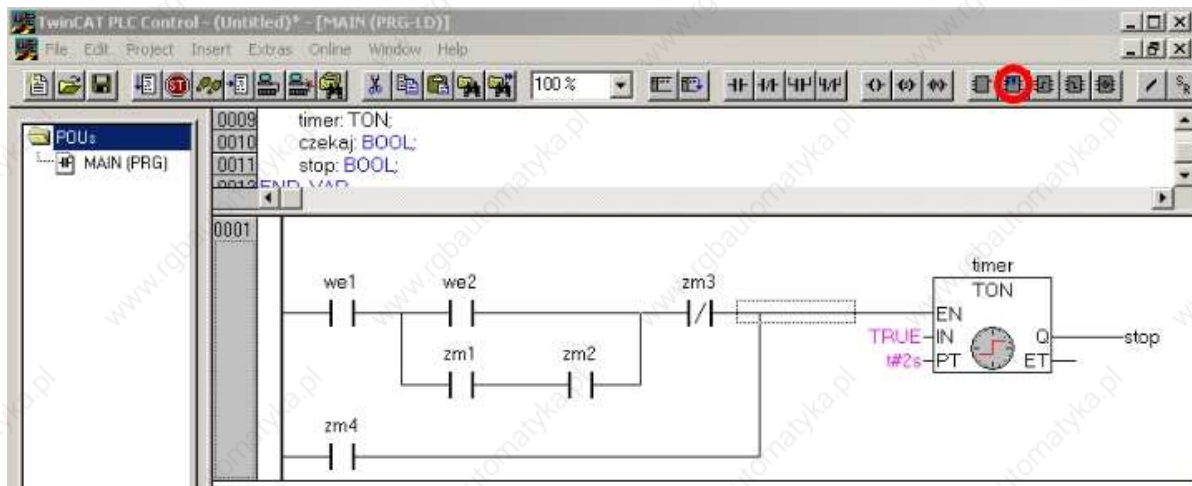
Język ST bardzo często stosowany jest do tworzenia programów, bloków funkcyjnych wymagających zastosowania obliczeń matematycznych.

### 1.3.3 LD – Ladder Diagram

Graficzny język programowania. Kod napisany w języku LD zbliżony jest do struktury obwodu elektrycznego zawierającego połączone szeregowo lub równolegle elementy takie jak: styki, cewki, liczniki, przekaźniki czasowe itp.. Kod podzielony jest na linie (tzw. Network). Linie wykonywane są od góry do dołu. W języku drabinkowym najczęściej używa się zmiennych typu BOOL, tworząc rozbudowane warunki logiczne. Jeżeli warunek jest prawdziwy to sygnał przekazywany jest do kolejnego elementu. Operacje na innych typach zmiennych, np. INT, REAL, wykonane mogą być przez specjalnie wywołane bloki funkcyjne i funkcje - „box with EN” (kliknięcie PPM na linii bądź wybranie z górnego menu). Wybrany blok pojawi się w oknie kodu

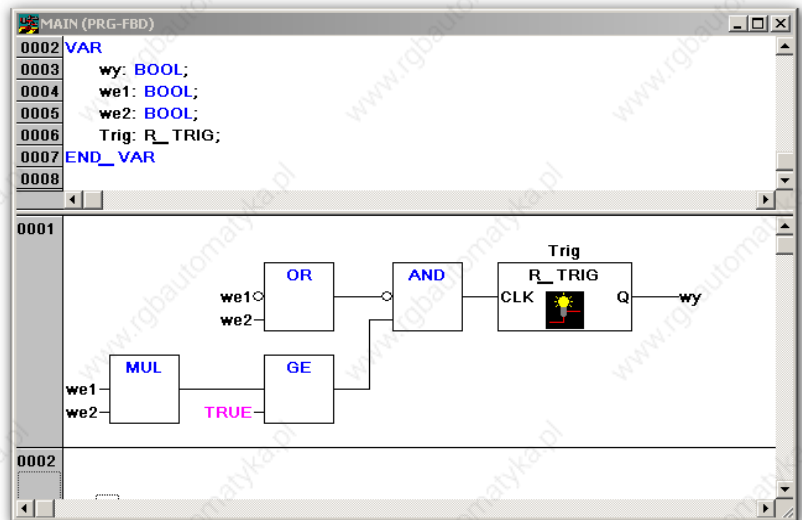


programu w zaznaczonej linii (networku). Wstawiony blok funkcyjny jest blokiem typu AND. Chcąc wybrać inny typ bloczka funkcyjnego należy kliknąć na jego nazwę, ręcznie wpisać typ bloku funkcyjnego lub wybrać klawisz F2. W oknie Input assistant należy wybrać określony typ bloku funkcyjnego. Zaletą programów napisanych w języku drabinkowym jest łatwość analizy i czytelność kodu.

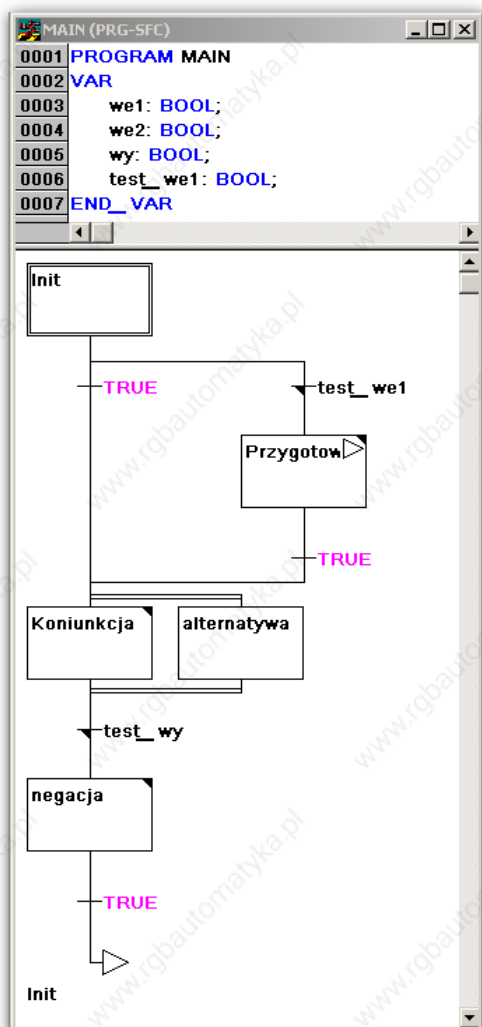


### 1.3.4 FBD - Function Block Diagram

FBD jest językiem graficznym. Kod tworzą linie wykonywane w określonej kolejności – linie wykonywane są od góry do dołu a ich zawartość od lewej do prawej. Linia zawierać może połączenie różnego typu elementów i dowolnych typów zmiennych – to daje większą swobodę niż język LD.

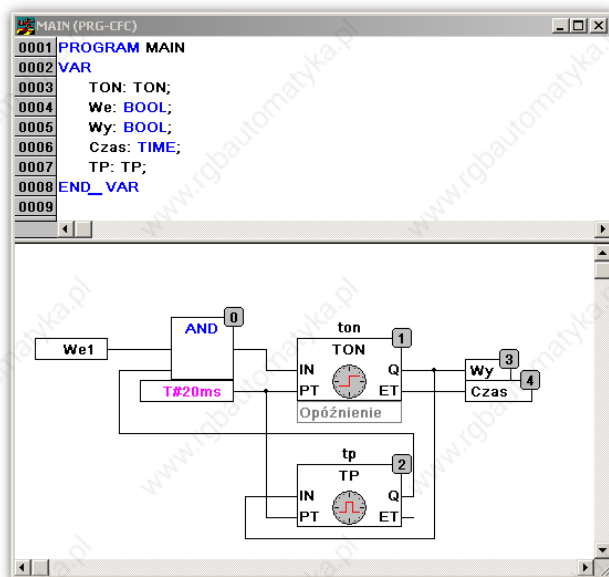


### 1.3.5 SFC – Sequential Function Chart



Język programowania, który graficznie przedstawia kolejność poszczególnych działań. Kod ujęty jest w blokach (**krokach**), które definiowane są przez użytkownika. Podczas tworzenia danego kroku programista może korzystać z dowolnego języka. Możliwe jest określanie warunku przejścia do następnego kroku (bloku), przy czym wartość wyjściowa warunku musi być TRUE lub FALSE. W języku SCF takie przejścia nazywane są tranzycjami. Kod wykonywany jest od góry do dołu, przy czym możliwe jest równoległe łączenie kroków. Wtedy dany poziom wykonywany jest kolejno od lewej do prawej. Język wykorzystywany jest, gdy należy zapewnić ścisłą kolejność wykonywania danego algorytmu. SFC stosowany jest np. do tworzenia programów sterujących sygnalizacją świetlną, procesów sekwencyjnej obróbki.

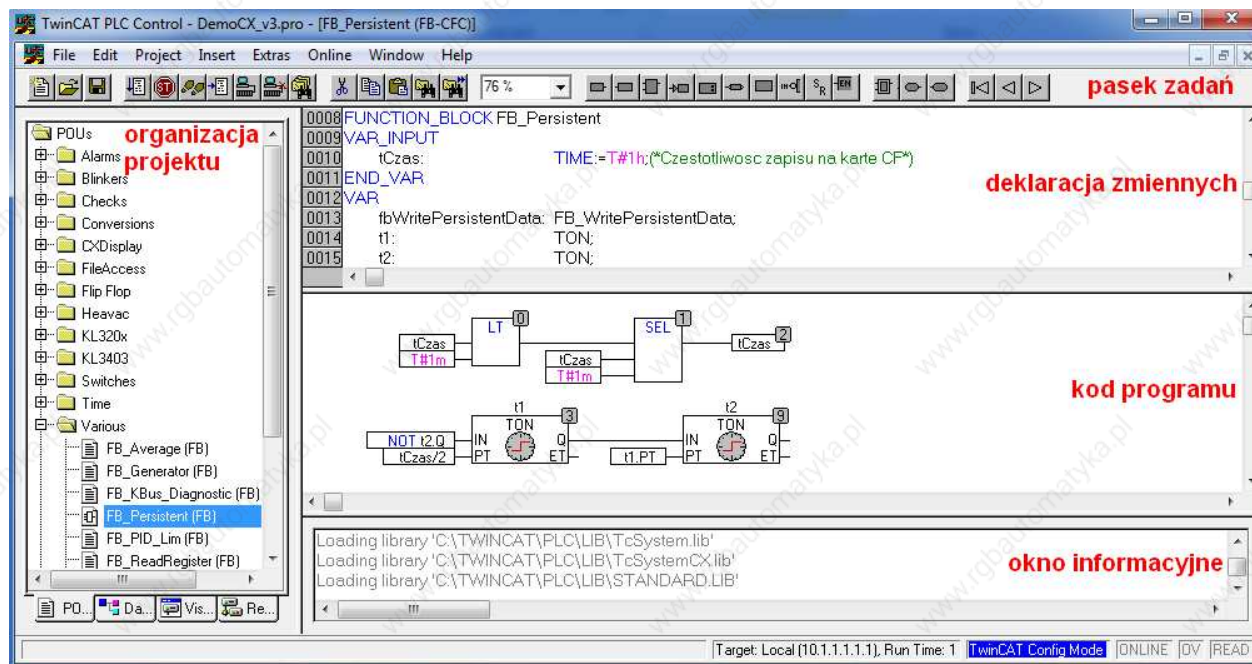
### 1.3.6 CFC – Continuous Function Chart



Kod tworzony jest z elementów, które mogą być umieszczane w dowolnym miejscu na ekranie. Wejścia i wyjścia można dowolnie łączyć, a połączenia te są obrazowane liniami. Można wielokrotnie odwoływać się do tych samych zmiennych, wejść i wyjść. Bardzo łatwo tworzy się sprzężenia zwrotne. Kolejność wykonywania elementów kodu jest zgodna z przypisaną im numeracją niezależnie od ich umiejscowienia. Istnieje możliwość automatycznego i ręcznego numerowania bloków.

## 2 Główne okno programu

Edytor programu podzielony jest na 5 części: pasek zadań, organizację projektu, deklarację zmiennych, kod programu i okno informacyjne. Wyszczególnione części zostały opisane i pokazane na rysunku poniżej.



### 2.1 Pasek zadań, ikony skrótów.



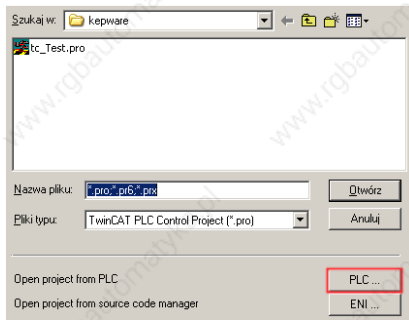
**Pierwsza grupa zawiera przyciski odpowiadające za tworzenie, otwieranie i zapisywanie programu:**



“New” – tworzenie nowego projektu



“Open” (Ctrl+O) – Otwieranie projektu, po wybraniu ikony należy podać ścieżkę do pliku zapisanego na dysku lub wybrać opcję PLC odczytującą program ze stownika PLC



PLC.. – otwórz projekt z PLC, projekt zostanie otwarty tylko wtedy, gdy na sterownik został wysłany cały projekt wraz z kodem. Standardowo podczas wgrывania programu na PLC przesyłany jest tylko kod binarny - efekt pracy kompilatora.



„Save” (Ctrl+S) - zapisywanie projektu

### **Grupa przycisków odpowiadająca za start/stop programu oraz za debugger**



„Run” (F5) - uruchomienie aplikacji w sterowniku



„Stop” (Shift+F8) - zatrzymanie aplikacji w sterowniku



“Toggle Breakpoint” (F9) - opcja debuggera, ustawia punkt w którym działanie punktu zostanie zatrzymane. W tym punkcie można podejrzeć wartości zmiennych oraz rozpocząć pracę krokową programu



“Step Over” (F10) - krokowe wykonanie programu, powoduje wykonanie program linijka po linijce, nie wchodząc do bloków funkcyjnych

### **Opcje logowania się na wybrany sterownik**



“Login” (F11) – logowanie się na wybrany sterownik, w momencie logowania istnieje możliwość wgrania lub aktualizowania programu na sterowniku. TwinCAT pyta użytkownika jaką operację wykonać



“Logout” (F12) – wylogowanie ze sterownika

### **Grupa przycisków służąca do edycji programu**



“Global Search” – wyszukiwanie w całym projekcie wpisanego ciągu znaków



“Cut” (Ctrl+X) – wycinanie zaznaczonego tekstu



“Copy” (Ctrl+C) – kopiowanie zaznaczonego tekstu



“Paste” (Ctrl+V) – wklejanie skopiowanego tekstu

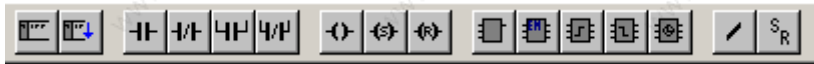


“Find” (Ctrl+F) – znajdowanie podanego ciągu znaków

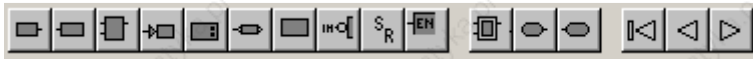


“Find Next” (F3) – znalezienie następnego dopasowania

**Pasek bloków funkcyjnych, wyświetlany tylko dla języków graficznych (inny dla każdego z nich)**



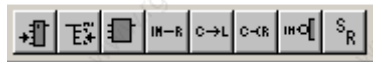
dla języka LD



dla języka CFC



dla języka SFC

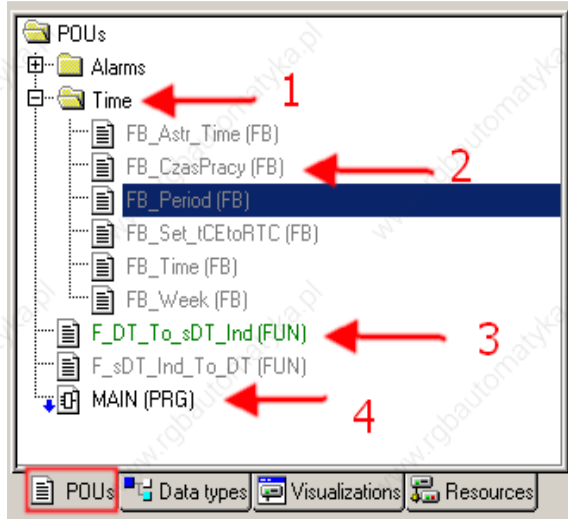


dla języka FBD

## 2.2 Menager organizacji projektu

Manager organizacji projektu jest narzędziem, które pomaga w zarządzaniu projektem. Menager wyświetla definiowane przez użytkownika programy, funkcje, bloki funkcyjne, wizualizacje, definiowane typy danych. Poniżej przedstawione są najważniejsze opcje menagera.

### Zakładka POUs (Program Organization Unit)

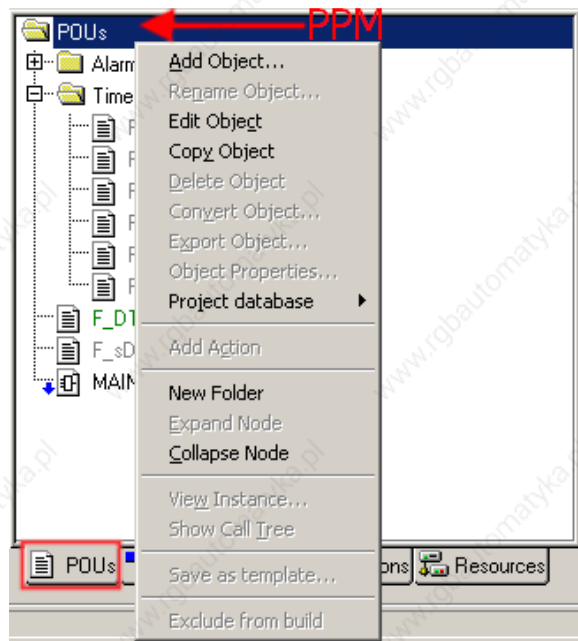


Zakładka przechowuje programy, funkcje i bloki funkcyjne. Elementy POUs można grupować w katalogi w celu lepszej organizacji pracy (nr 1).

Szara nazwa obiektu informuje programistę, że dany obiekt nie jest wywołany (wykorzystany) w programie (nr 2).

Nazwa POUs zaznaczona na zielono oznacza, że programista, wykluczył dany element kompilacji i jest sprawdzany pod kątem poprawności, wszystkie błędy w nim występujące są przez kompilator ignorowane (nr 3).

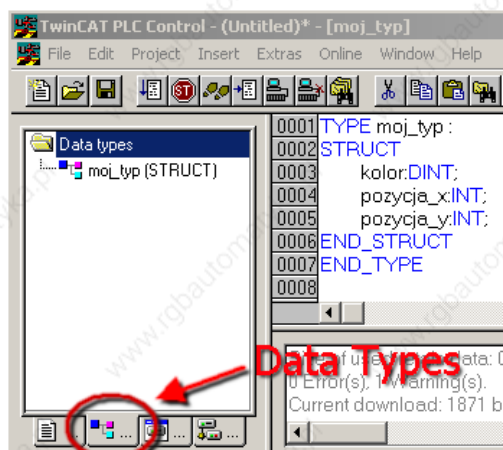
Kompilowane i używane POUs zaznaczane są kolorem czarnym.



Niebieska strzałka przy nazwie POU's informuje, że dany obiekt będący w pamięci sterownika różni się od tego w projekcie (nr 4) i różnice zostaną wgrane podczas następnego logowania.

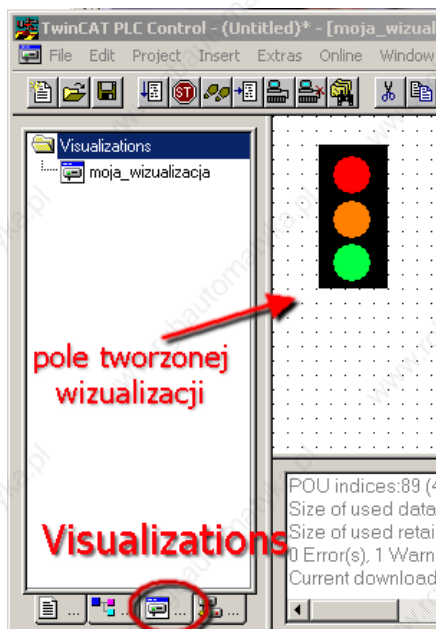
Klikając prawym przyciskiem myszy na folder POU's pojawia się menu kontekstowe, zawierające wiele opcji dotyczących obiektów. Najważniejsze z nich to tworzenie, edytowanie, usuwanie i eksportowanie obiektów.

### Zakładka Data Types



Zakładka Data Types pozwala na tworzenie własnych typów zmiennych. Tworzone typy to najczęściej struktury i typy wyliczeniowe ENUM (Enumeration). Polami struktur mogą być zmienne typów standardowych lub stworzonych przez użytkownika.

### Zakładka Visualizations

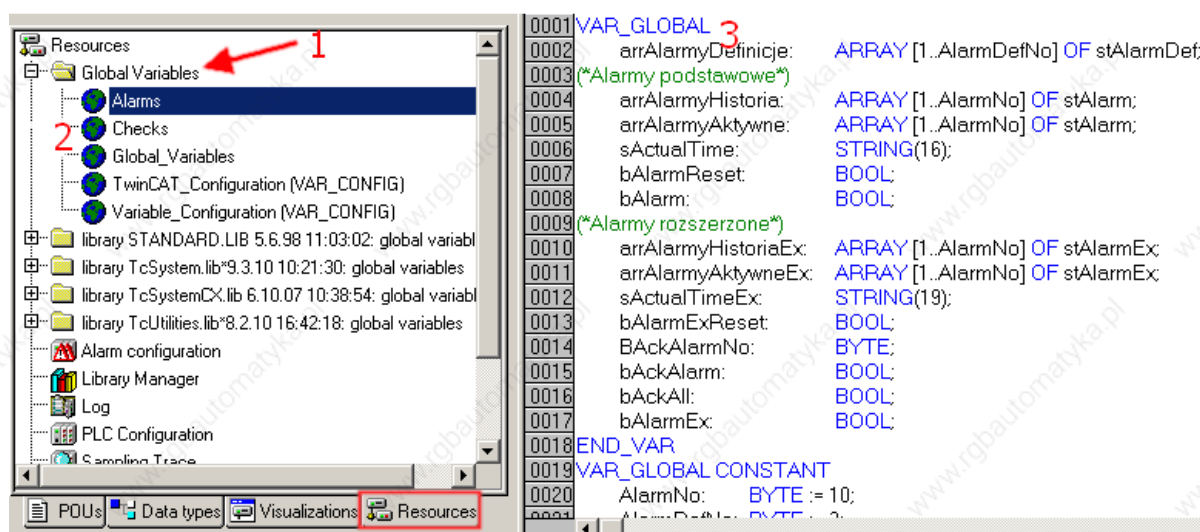


Zakładka Visualizations pozwala na tworzenie własnych wizualizacji. Wizualizacja może zawierać obiekty dynamiczne, których stan zależy od wartości zmiennych przypisanych w projekcie PLC.

Wizualizacje można przesłać do sterowników z serii CX, paneli operatorskich oraz komputerów przemysłowych, na którym zainstalowana jest biblioteka TwinCAT PLC HMI lub TwinCAT PLC HMI WEB. Nie ma ograniczeń typów oraz liczby przesyłanych zmiennych z PLC do wizualizacji. Biblioteka PLC HMI pozwala wyświetlać wizualizację na Panelu (PLC z serii CX przesyłają wizualizację poprzez złącze DVI). Biblioteka PLC HMI WEB pozwala wyświetlić wizualizację w przeglądarce internetowej.

### Zakładka Resources

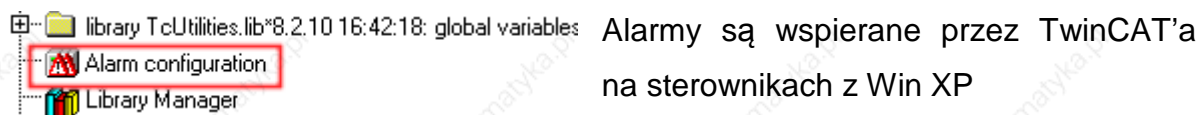
#### Zmienne Globalne (Global Variables)



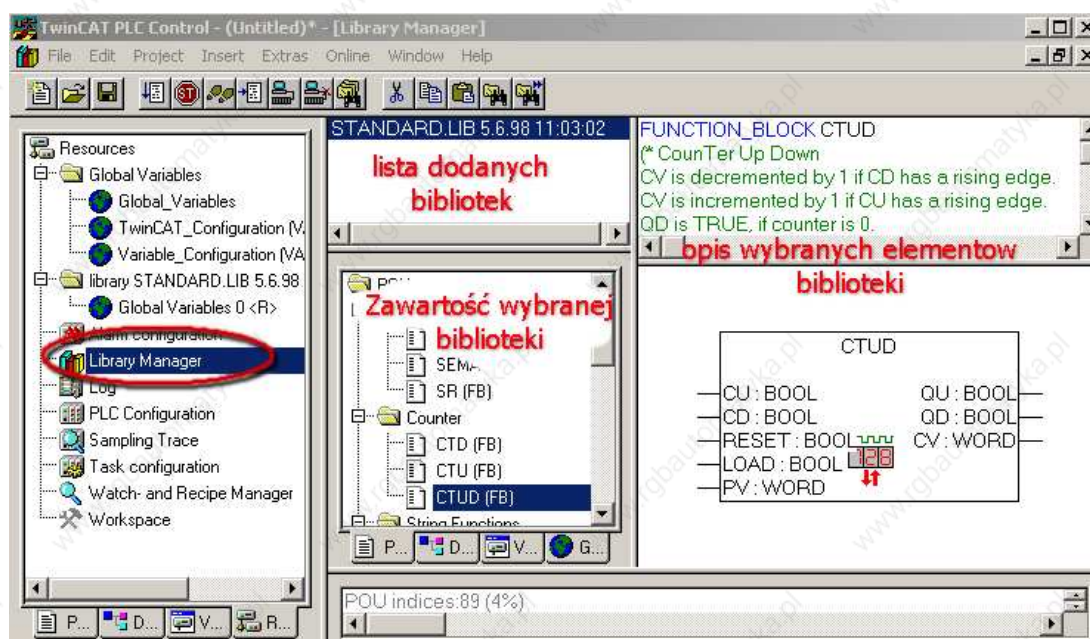


Pozycja Global Variables służy do deklaracji zmiennych globalnych. W folderze Global Variables oznaczonego nr 1 przechowywane są wszystkie zmienne globalne występujące w projekcie. Zmienne globalne można grupować tworząc nowe tzw. obiekty/foldery zmiennych (obiekty/foldery oznaczono na rysunku nr 2). Deklaracja zmiennych przebiega w oknie deklaracji oznaczonym nr 3. Składnia deklaracji jest taka sama jak zmiennych lokalnych.

### Alarm Configuration

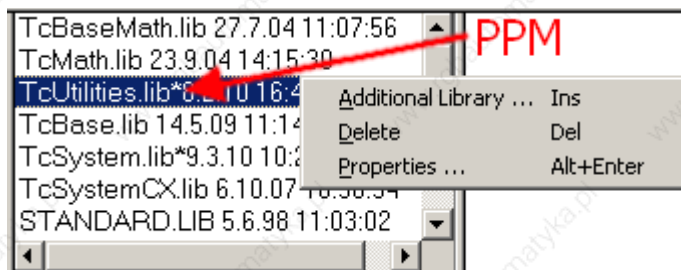


### Library Manager



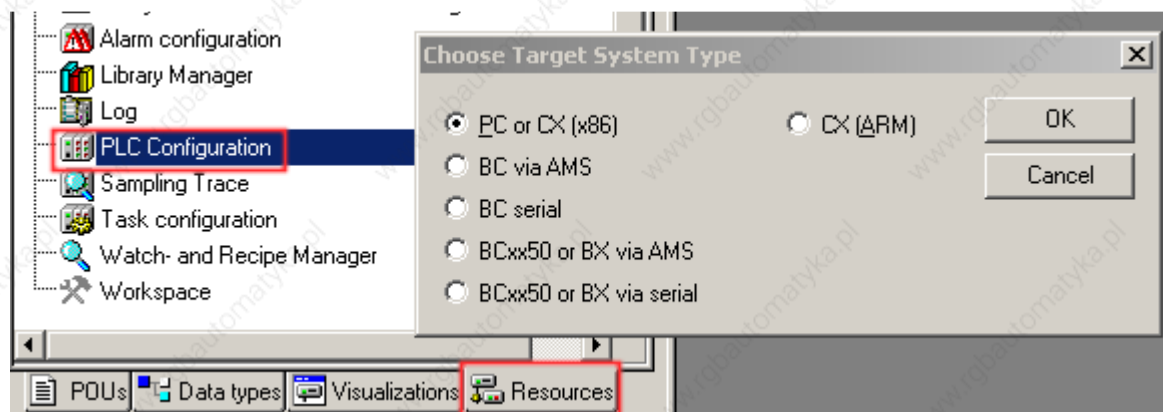
Library Manager pozwala efektywnie zarządzać bibliotekami dołączanymi do projektu. Zawiera informacje o aktualnie załączonych bibliotekach jak również pozwala na dołączanie kolejnych lub usuwanie tych nieużywanych.

Po utworzeniu programu dołączona jest do niego tylko standardowa biblioteka zawierająca podstawowe elementy. Dzięki temu program szybko się kompiluje i zajmuje mało miejsca w pamięci. Użytkownik może dodać do projektu inne biblioteki klikając prawym przyciskiem myszy na okno oznaczone jako „**lista dodanych bibliotek**”. W managerze można przeglądać zawartość danej biblioteki oraz opis poszczególnych jej elementów.



Dodanie biblioteki jest możliwe po wybraniu funkcji **Additional Library...** z menu rozwiniętego po kliknięciu PPM w oknie z listą dodanych bibliotek.

### PLC Configuration



PLC Configuration pozwala wybrać typ sterownika, na który jest pisany program. Jest to to samo okno, które występuje podczas tworzenia projektu.

### Sampling Trace

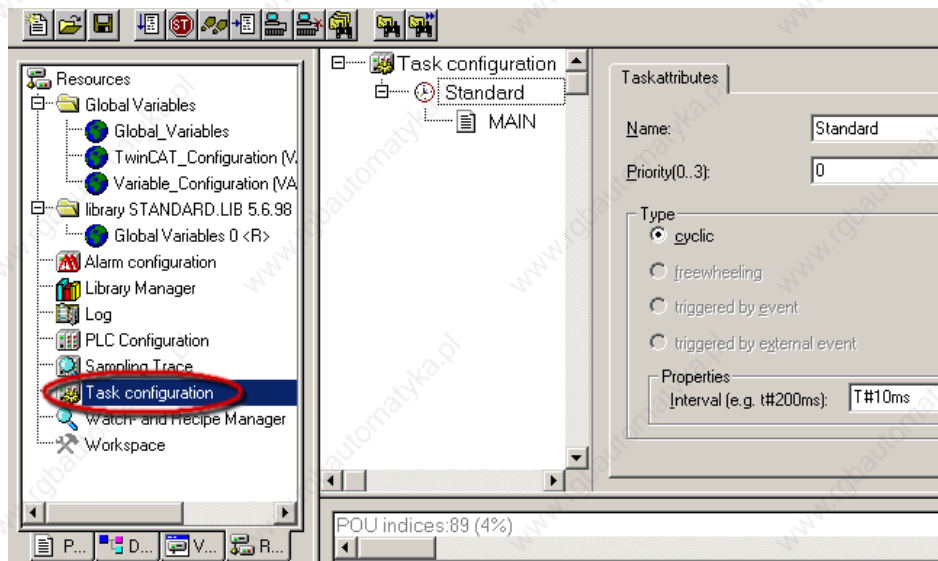
Narzędzie za pomocą którego można rysować wartości zmiennych na wykresie, obecnie zastąpione darmowym programem Scope View. Program ten jest instalowany razem z TwinCAT'em.

### Workspace



Workspace służy do ustawiania opcji PLC Control – opcje zostały opisane w rozdziale **10.3 Menu Project**

### Taks Configuration



Task configuration służy do konfiguracji tasków. Pozwala dodawanie, edytowanie, usuwanie, ustawianie priorytetu oraz ustawianie czasu cyklu tasku. Konfiguracja tasku jest opisana w rozdziale 4. **Task(Wątek)**.

### Watch and Recipe Manager



Watch and Recepte Manager pozwala na podgląd wybranych zmiennych. Zmienne możemy podzielić na grupy. Grupy dodaje się w polu oznaczonym na rysunku nr 1 klikając **PPM**, wybierając menu kontekstowego **New Watch List**. Zmienne dodajemy w polu oznaczonym nr 2 przez wybranie **F2** i uruchomienie **Input Assistant**.

Inną przydatną funkcją tego okna jest możliwość przechowywania aktualnych wartości zmiennych po wylogowaniu się. Pozwala to np. wpisać te wartości jako startowe do innego sterownika. W ten celu zmiennym należy nadać wartości startowe i będąc zalogowanym na sterowniku wybrać z menu **Extras** → **Read Recipe**. Po wylogowaniu wartości startowe będą zawierały odczytane wartości zmiennych (będą zapisane w projekcie, widoczne offline). W celu wpisania tych wartości do zmiennych na sterowniku należy w trybie online wybierać opcję **Extras** → **Write Recipe**. Wartości widziane w trybie offline będą teraz aktualnymi tych zmiennych.

## 2.3 Okno deklaracji zmiennych

Zawiera deklaracje wszystkich zmiennych lokalnych danego elementu. Zmienne deklarowane są pomiędzy znacznikami np.: `VAR END_VAR`. Wszystkie znaczniki opisane zostały w rozdziale 5. **Deklaracja zmiennych**.

Przykładowe okno deklaracji zmiennych przedstawiono na rysunku poniżej.

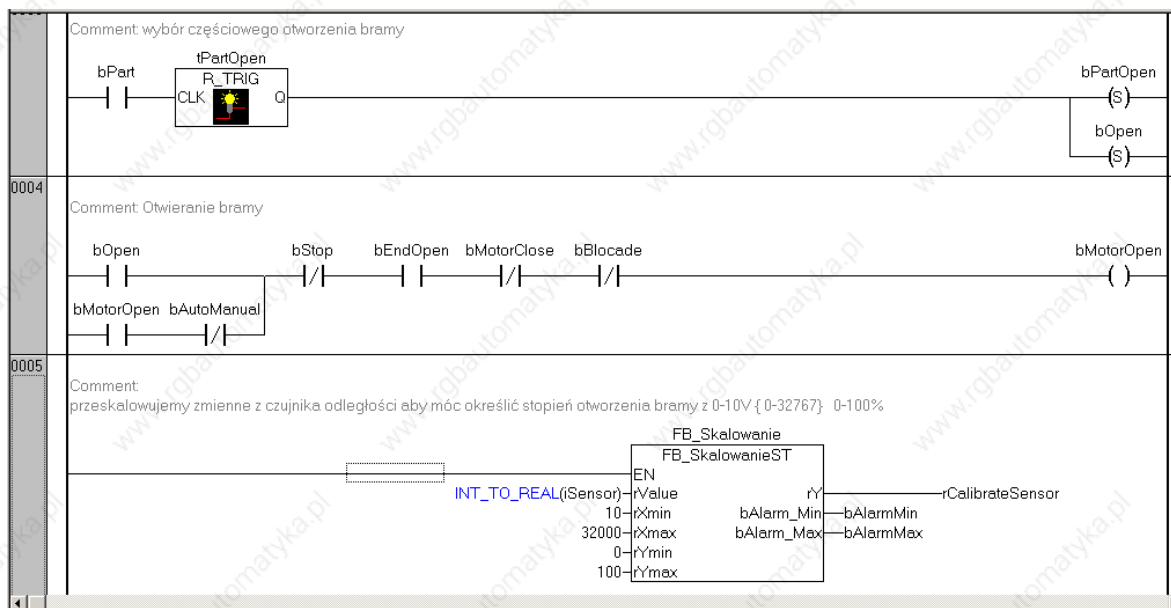
```

0001 FUNCTION_BLOCK FB_BlokFunkcyjny
0002 VAR_INPUT
0003     bInput: BOOL;                (* zmienne wejściowe *)
0004     wStatus: WORD;
0005 END_VAR
0006 VAR_OUTPUT
0007     bOutput: BOOL;              (* zmienne wyjściowe *)
0008     byControl: BYTE;
0009 END_VAR
0010 VAR
0011     iZwyklaZmienna: INT;
0012     arrZawory: ARRAY [1..10] OF INT; (* zmienna tablicowa *)
0013     byMemory AT : BYTE;         (* zmienna odwołująca się do *)
0014                                     (* przestrzeni adresowej %MB0 *)
0015 END_VAR

```

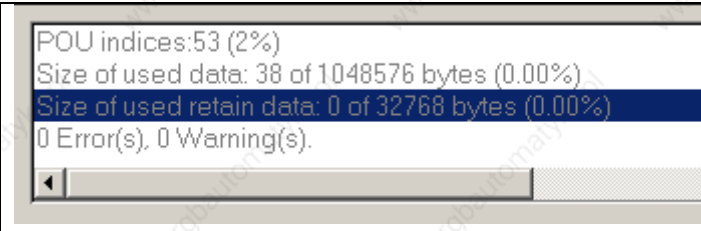
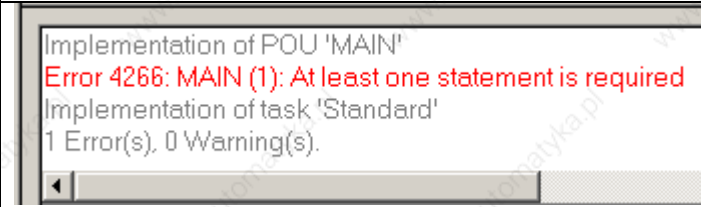
## 2.4 Okno kodu programu

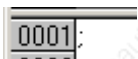
Zawiera kod programu. Do pisania programu można wykorzystać 6 języków programowania opisanych w rozdziale 1.3 **Języki programowania**. Poniżej pokazano przykładowy program napisany w języku drabinkowym LD.



## 2.5 Okno informacyjne

Okno można zamknąć i otworzyć wciskając kombinację klawiszy Shift+Esc. Pokazuje ono komunikaty, ostrzeżenia lub błędy wygenerowane przez kompilator podczas kompilacji programu oraz informacje dodatkowe np. o dodanych bibliotekach, zaimportowanych blokach - po ich dodaniu. Po zakończeniu kompilacji na samym dole okna dostępne jest podsumowanie zawierające dane na temat wielkości skompilowanego programu, wielkości zadeklarowanych zmiennych przechowywanych po zaniku zasilania oraz liczba znalezionych błędów i ostrzeżeń. W oknie informacji pokazują się również treść ostrzeżenia (warning:) oraz treść i numer błędu (error:). Dwukrotnie klikając na linijkę informującą o błędzie kursor zostanie przeniesiony w miejsce błędu w oknie kodu programu. Błędy należy usuwać od pierwszego znajdującego się na liście, ponieważ jeden błąd może generować kolejne. Program będzie skompilowany jeśli nie będzie zawierał żadnego błędu (0 Errors), może jednak zawierać ostrzeżenia (Warnings). Poniżej przedstawiono podsumowanie kompilacji, oraz wybrane komunikaty błędów i ostrzeżeń.

 <p>POU indices:53 (2%)          Size of used data: 38 of 1048576 bytes (0.00%)          Size of used retain data: 0 of 32768 bytes (0.00%)          0 Error(s), 0 Warning(s).</p>	<p>Poprawna kompilacja programu,          0 błędów, 0 ostrzeżeń</p>
 <p>Implementation of POU 'MAIN'  <b>Error 4266: MAIN (1): At least one statement is required</b>          Implementation of task 'Standard'          1 Error(s), 0 Warning(s).</p>	<p>Niepoprawna kompilacja programu, 1 błąd, 0 ostrzeżeń.</p>
<p><b>Wybrane ostrzeżenia</b></p>	
<p>Warning: failed to create symbol information file</p>	<p>Tworzony projekt nie został zapisany na dysku, kompilator nie mógł stworzyć pliku „*.tpy”. Plik z rozszerzeniem tpy tworzony jest w katalogu projektu i przechowuje informację o deklarowanych zmiennych</p>

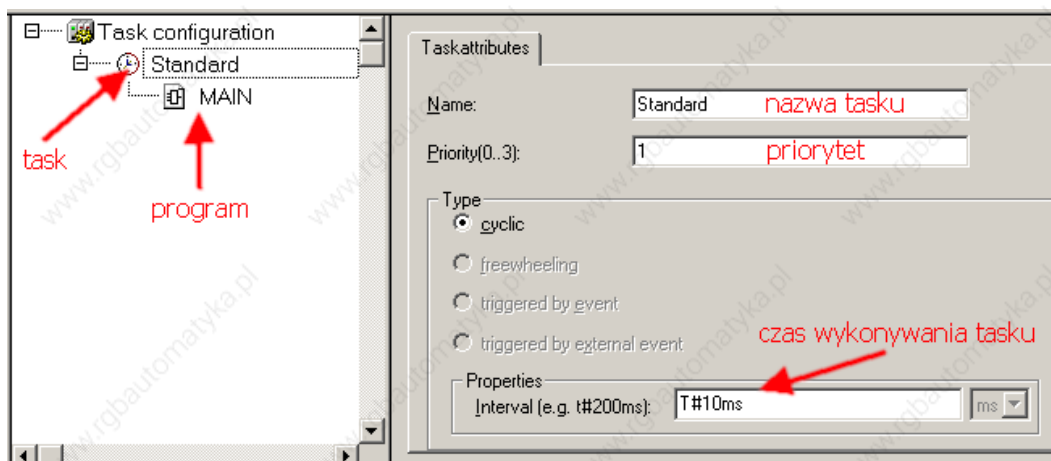
Warning 1990: No 'VAR_CONFIG' for 'MAIN.bInput	Została zadeklarowana zmienna, która może łączyć się z fizycznym urządzeniem (np. D/A input/output lub port szeregowy), lecz zmienna ta nie została zlinkowana w System Managerze z żadnym urządzeniem.
<b>Wybrane błędy</b>	
Error 4266: MAIN (1): At least one statement is required	Wymagana chociaż jedna instrukcja wywołana w oknie programu napisanego w języku ST. Język ST wymaga użycia co najmniej jednej instrukcji, może to być instrukcja pusta zakończona średnikiem 
Error 4024: MAIN (5): Expecting '!' or '!=' before 'END_VAR' Error 3782: MAIN (5): Unexpected end	Błędna deklaracja zmiennej, brak średnika na końcu instrukcji lub błędna inicjalizacja początkowej wartości
Error 4001: MAIN (2): Identifier 'BZMIENNA' not defined	Brak deklaracji zmiennej o nazwie bZmienna.
Error 4024: MAIN (3): Expecting '!', '!=' or '(' before "	Brak średnika kończącego instrukcję lub niepoprawne przypisanie.

### 3 Task(Wątek)

Aplikacje pisane w PLC Control na sterowniki z serii CX mogą mieć zdefiniowane maksymalnie 4 taski. Projekty przeznaczone dla sterowników z serii BX i BC mają zdefiniowany tylko jeden wątek. W każdym zdefiniowanym wątku należy wywołać przynajmniej jeden program. W jednym tasku można wywołać kilka programów. W tasku wywoływany może być tylko program. Task wykonywany jest cyklicznie, musi mieć on przypisany konkretny interwał czasowy oraz określony priorytet. Priorytety

przyjmują wartości od 0 do 4, przy czym wątek z priorytetem 0 jest najważniejszy i powinien mieć ustawiony najmniejszy czas cyklu. Wątki nie mogą mieć tego samego priorytetu. Każdy task może mieć przypisaną inną wartość interwału czasowego.

Domyślnie podczas tworzenia nowego projektu, pierwszy zdefiniowany program automatycznie jest przypisywany do pierwszego wątku. Domyślny czas cyklu w sterownikach CX to 10 ms a w sterownikach BC/BX to 20 ms. Wszystkie ustawienia związane z konfiguracją wątków znajdują się w PLC Control w zakładce **Resources** w pozycji **Task Configuration**.




Dodając nowy wątek należy kliknąć PPM na drzewo Task configuration i z menu kontekstowego wybrać spośród dostępnych opcji **Append Task**.


Analogicznie należy postąpić w przypadku przypisania programu do wątku, wybierając **Append Program Call**. Standardowo wywołany jest już pierwszy task z dołączonym programem głównym. W przypadku usuwania programów z kolejki lub tasków – należy posłużyć się klawiszem **delete** lub klikając PPM na nazwę programu wybrać opcję **Delete**.

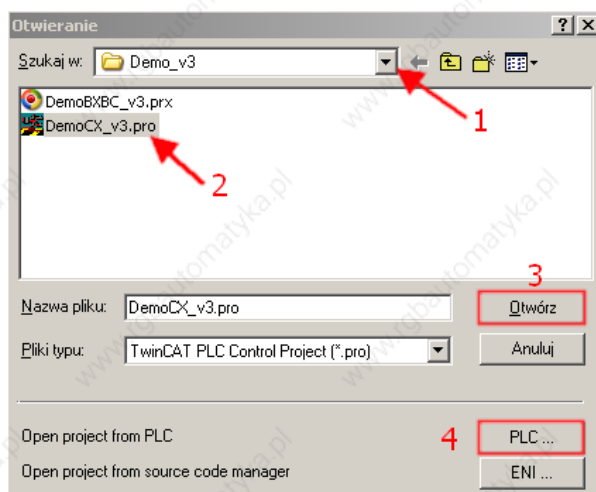
## 4 Edycja projektu

### 4.1 Tworzenie nowego projektu

Podczas uruchomienia TwinCAT PLC Control automatycznie otwierany jest ostatnio modyfikowany i zapisany projekt. W każdej chwili można stworzyć nowy projekt klikając w ikonę  lub wywołując polecenie z menu: **File** → **New**

### 4.2 Otwieranie projektu

Chcąc otworzyć zapisany na dysku projekt należy wybrać z paska zadań ikonę folderu  lub wybrać z menu **File** → **Open** (Ctrl+O). Wybranie opcji spowoduje uruchomienie się okna pokazanego na rysunku poniżej.




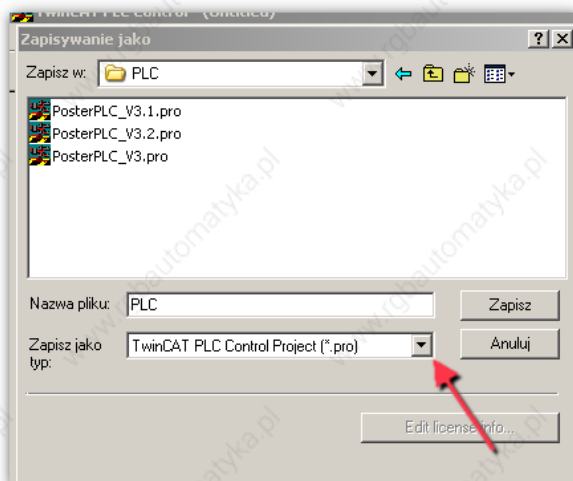
Wybierając ścieżkę do stworzonego projektu (nr 1) i odpowiedni plik z rozszerzeniem \*.pro, \*.prx lub \*.pr6 (nr 2). Projekt zostanie otwarty po wybraniu przycisku **Otwórz** (nr 3) można otworzyć program zapisany na komputerze.

Wybierając opcję **Open project from PLC** – przycisk **PLC** (nr 4) można otworzyć program zapisany na sterowniku, jeżeli został on wcześniej tam wysłany (aby przesłać kod źródłowy na sterownik, po zalogowaniu się należy wybrać z menu **Online** → **SourceCode download** – jest to opisane w rozdziale **9.4 Menu online**).



### 4.3 Zapisywanie projektu

Po kliknięciu w ikonę  (Ctrl+S) można zapisać wprowadzone zmiany w projekcie. Zapisanie projektu pod nową nazwą następuje po wybraniu polecenia z menu **File** → **Save As**. Pojawia się okno dialogowe, w którym podajemy nazwę i nową lokalizację projektu:



Projekty zapisywane są w domyślnym formacie zależnym od sterownika, dla którego program jest przeznaczony (\*.pro – PC i CX; \*.pr6 – BC; \*.prx – BC xx50 i BX). Odpowiedni typ pliku można wybrać z listy rozwijanej. Zmieniając rozszerzenie na \*.lib projekt zapisany zostanie jako biblioteka. Biblioteka może podobnie jak i projekt zawierać typy zmiennych, funkcje i bloki funkcyjne definiowane w zapisywanym projekcie. Nie może zawierać tasków.

## 5 Deklaracja zmiennych

### 5.1 Wstęp

TwinCAT pozwala na tworzenie nazw zmiennych, które:

- Nie są słowami kluczowymi wykorzystywanymi przez kompilator np. VAR, END\_VAR, TRUE, FUNCTION
- Nie są nazwami zmiennych, bloków funkcyjnych, np. REAL, TON
- Nie zawierają polskich znaków
- Nie zawierają znaków specjalnych jak: !, @, #, \$, ^, &, \*
- Nie zawierają pustych znaków, tj. spacja ' '
- Nie zaczynają się od cyfr
- Nie mogą różnić się jedynie wielkością znaków

### Pole deklaracji zmiennych

Deklaracje zmiennych umieszczane są pomiędzy słowami kluczowymi przedstawionymi poniżej:

VAR END_VAR	zmienne lokalne
VAR_INPUT END_VAR	zmienne wejściowe
VAR_OUTPUT END_VAR	zmienne wyjściowe
VAR_IN_OUT END_VAR	zmienne wejściowo – wyjściowe
VAR GLOBAL END_VAR	zmienne globalne

### Komentarze

Komentarze zmiennych ograniczone są koloru zielonego i ograniczone znacznikami „(“ i „\*)”, np. (\* treść komentarza \*). Komentarze można wstawiać w każde pole przyjmujące znaki. Komentarzy nie można stosować wewnątrz łańcucha znaków słów kluczowych, nazw zmiennych, nazw bloków funkcyjnych itp..

```

0029 PROGRAM MAIN
0030 VAR
0031 (**Wejścia cyfrowe*)
0032   bStart AT%IX0.0:BOOL;(*Start Maszyny*)
0033 (**Wejścia Analogowe*)
0034   wTErmopara AT%IW10(*Byte 10-11*):WORD;
0035 END_VAR
    
```

Poprawny komentarz

```

0029 PROGRAM MAIN
0030 VAR
0031   bStart AT%IX0.0:BOOL;
0032   wTErmopara AT%(*Byte 10-11*)IW10:WORD;
0033 END_(**koniec deklaracji**)VAR
    
```

Błędny komentarz

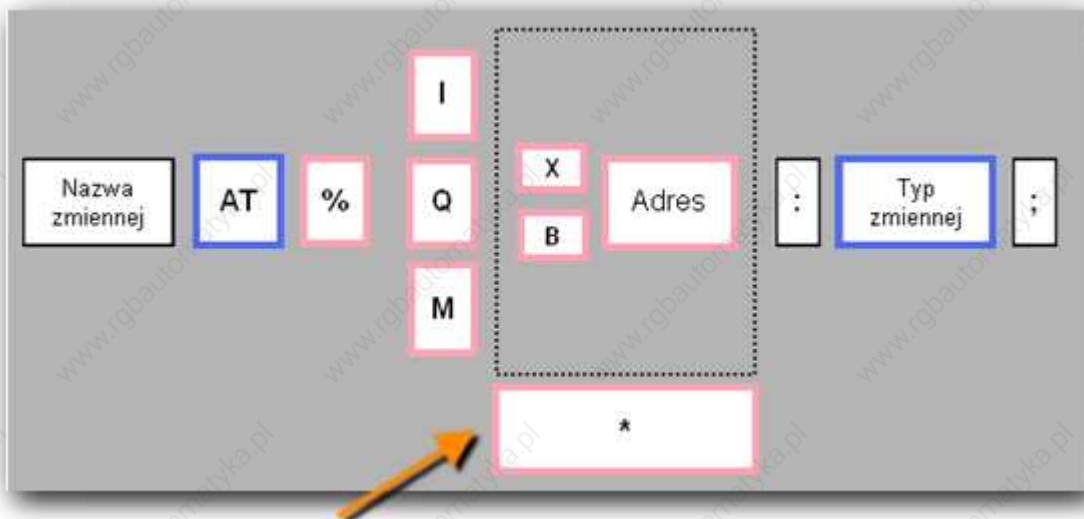
**Prefix**

TwinCAT pozwala na dowolność w tworzeniu nazw zmiennych. Dobrym nawykiem programistycznym jest stosowanie prefiksów. Poniżej przedstawiono przykłady tworzenia nazw zmiennych wykorzystujących Notację Węgierską.

Prefiks	Nazwa zmiennej
b – Boolean	bStart
r – Real	rAktualnaPozycja
s - String	sWiadomosc
ST_ - Definicja struktury	ST_Zawor (* Definicja *)
st – Deklaracja struktury	stZawrorPompy (* Deklaracja *)
FB_ - Definicja bloku funkcyjnego	FB_Skalowanie (* Definicja *)
fb	fbSkalTemp (* Deklaracja *)

**5.2 Struktura Deklaracji**

Ogólna struktura deklaracji zmiennych przedstawiona jest poniżej:



TwinCAT pozwala na zadeklarowanie dwóch grup zmiennych.

- **Zmienne wewnętrzne** – używane w obrębie sterownika. Ich deklaracja nie wymaga podawania adresu a tym samym nie są linkowane. Zmienne te nie mogą odwoływać się do urządzeń zewnętrznych. Deklaracja takiej zmiennej składa się z nazwy zmiennej i typu zmiennej. Przykładowa deklaracja:

***Nazwa\_zmiennej : typ zmiennej;***

***bStart : BOOL;***

- **Zmienne zewnętrzne** – łączone są z fizycznymi urządzeniami w procesie linkowania np. zmienne odwołujące się do konkretnego fizycznego wejścia/wyjścia. Deklaracja takiej zmiennej składa się z dodatkowego słowa kluczowego **AT** oraz określenia rodzaju zmiennej i podania adresu zmiennej.

Przedrostek zmiennej I, Q, M występujący po znaku % wskazuje obszar do jakiego zmienna będzie się odwoływać.

- **I** – obszar zm. wejściowej (Input);
- **Q** – obszar zm. wyjściowej (Output);
- **M** – pamięć flag (Memory);

Kolejnym elementem deklaracji jest określenie adresu zmiennej:

- **X** – zmienna bitowa (logiczna) – wyrażona przez pojedynczy bit;
- **B** – zmienna bajtowa (Byte) – wyrażona przez 8 bitów;
- **\*** – informuje kompilator, że adres będzie przydzielony podczas procesu linkowania

***Nazwa\_zm1 AT %I\* : BOOL;***

***Nazwa\_zm2 AT %MB0 : INT;***

Zmienne zadeklarowane jako wejścia lub wyjścia powinny być zlinkowane do odpowiadających im wejść i wyjść fizycznych w programie TwinCAT System Manager (program opisany w oddzielnym dokumencie dostępnym na [ftp.beckhoff.com/poland/Pomoc/](http://ftp.beckhoff.com/poland/Pomoc/)).

Podczas deklaracji zmiennej można przypisać jej **wartość początkową**. Wartość początkową można przypisać ręcznie poprzez dodanie na końcu deklaracji:

***bStart : BOOL := TRUE;***

***iOut AT %Q\* :INT := 3300;***

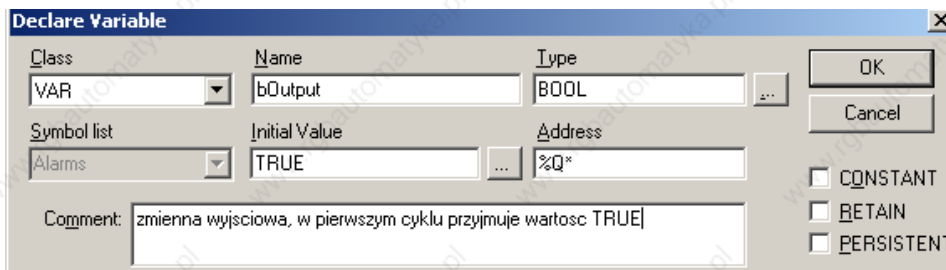
lub w polu Initial Value w oknie deklaracji zmiennych. Kreator deklaracji zmiennych opisany jest w rozdziale **6.3 Kreator deklaracji zmiennej**.

Przykładowe deklaracje zmiennych:

<pre> 0002 VAR 0003 (*zmienne wewnętrzne *) 0004   bBool: BOOL; 0005   wWord : WORD:= 320; 0006   rReal: REAL; 0007 END_VAR         </pre>	<p>Zmienne lokalne wewnętrzne, których nie można łączyć z zewnętrznymi urządzeniami. Zmienna wWord zostanie zainicjalizowana wartością 320.</p>
<pre> 0001 VAR_GLOBAL 0002 (*Zmienne Globalne*) 0003   bBoolGlobalna :BOOL; 0004   bDigitalOutputGlobalna AT %Q*: BOOL := TRUE; 0005 END_VAR         </pre>	<p>Zmienne Globalne - deklarowane są w zakładce <b>Resources → Global Variables</b>. Zmienne dostępne są we wszystkich POU,</p>


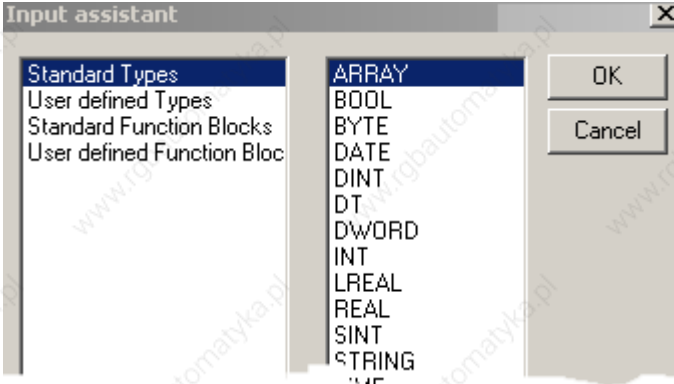




### 5.3 Kreator deklaracji zmiennej

Każda zmienna używana w programie musi być zadeklarowana. W przypadku użycia nowej nieznannej zmiennej w programie, edytor zapyta o jej deklarację. Automatycznie pojawia się okno deklaracji zmiennej **Declare Variable**.



Poniżej przedstawione są dostępne w polu deklaracji opcje:

	<p>Wybór klasy zmiennej:          VAR – lokalna,          VAR_INPUT – wejściowa,          VAR_OUTPUT – wyjściowa,          VAR_IN_OUT – wejściowo-wyjściowa,          VAR_GLOBAL – globalna</p>
	<p>Nazwa zmiennej – nazwę należy wpisać ręcznie lub zatwierdzić istniejącą</p>

	<p>Typ zmiennej można wpisać ręcznie lub wybrać przycisk, który otworzy okno <b>Input Assistant</b>. Pozwala ono na wybór typów standardowych lub własnych, zdefiniowany przez programistę.</p> 
	<p>Pole <b>Symbol List</b> (nr 2) pozwala wybrać grupę zmiennych, do której można dołączyć deklarowaną zmienną. Pole dostępne jest tylko przy deklaracji zmiennej globalnej wybrana klasa zmiennej VAR_GLOBAL (nr 1).</p>
	<p>W polu można nadawać wartość inicjalizacyjną – zmienna przyjmie ją w pierwszym cyklu programu</p>
	<p>Jeśli zmienna ma być linkowana z zewnętrznym wejściem/wyjściem to wpisujemy adres. Dla zmiennej wewnętrznej pole zostaje puste.</p>
	<p>Komentarz wyświetlany w linii deklaracji zmiennej,</p>
<input type="checkbox"/> CONSTANT	<p>Deklaracja zmiennej stałej – nie można zmienić jej wartości w programie. Zmiennej można tylko inicjalizować wartość.</p>
<input type="checkbox"/> RETAIN	<p>Deklaracja zmiennej nieulotnej typu RETAIN<sup>1</sup></p>
<input type="checkbox"/> PERSISTENT	<p>Deklaracja zmiennej nieulotnej typu PERSISTENT<sup>2</sup></p>

W celu modyfikacji zmiennej istnieje możliwość samodzielnego wywołanie okna **Declare Variable** poprzez kliknięcie PPM w polu deklaracji zmiennej na jej nazwę i z rozwiniętego menu wybranie opcji **Auto Declare (Shift+F2)**.

<sup>1</sup> Zmienne RETAIN opisane są w osobnej dokumentacji dostępnej na <ftp.beckhoff.com/poland/pomoc>

<sup>2</sup> Zmienne PERSISTENT opisane są w osobnej dokumentacji dostępnej na <ftp.beckhoff.com/poland/pomoc>

## 5.4 Typy Standardowe

W bibliotece standardowej zdefiniowane zostały podstawowe typy zmiennych.

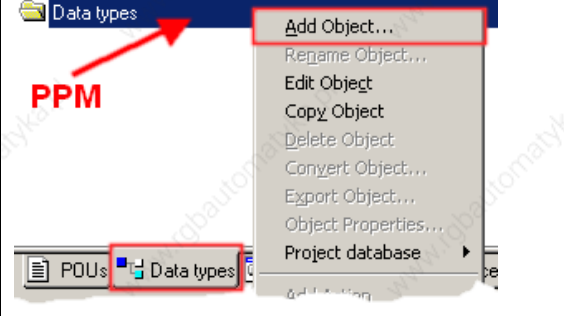
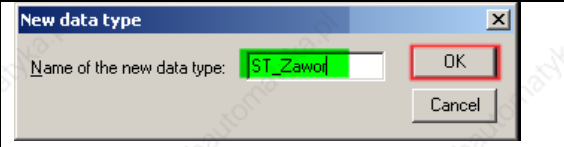
Typ zmiennej	Wartość min.	Wartość max.	Wielkość	prefix
BOOL	0	1	8 bit	b
BYTE	0	255	8 bit	by
WORD	0	65535	16	w
DWORD	0	4294967295	32	dw
SINT	-128	127	8	si
USINT	0	255	8	usi
INT	-32768	32767	16	i
UINT	0	65535	16	ui
DINT	-2147483648	2147483647	32	di
UDINT	0	4294967295	32	udi
REAL	$\sim -3.402823 \times 10^{38}$	$\sim 3.402823 \times 10^{38}$	32	r
LREAL	$\sim -1.797693134862 \times 10^{308}$	$\sim 1.797693134862 \times 10^{308}$	64 Bit	lr
STRING	-	-	Domyślnie 81 Bajtów	s
TIME	T#0ms	T#71582m47s295ms	32 Bit	t
TIME_OF_DAY (TOD)	TOD#00:00	TOD#1193:02:47.295	32	tod
DATE	D#1970-01-01	D#2106-02-06	32	date
DATE_AND_TIME (DT)	DT#1970-01-01-00:00	DT#2106-02-06-06:28:15	32	dt

## 5.5 Typy Deklarowane przez użytkownika

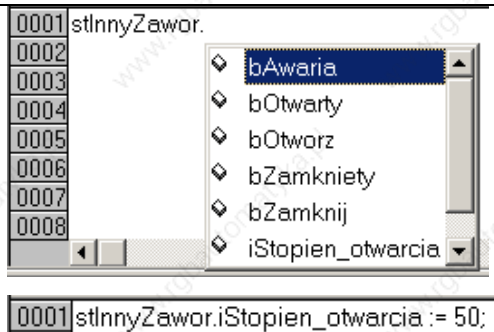
W sytuacji, gdy standardowe typy są niewystarczające można zadeklarować własne typy. Użytkownik może tworzyć własne struktury, typy wybieralne (ENUM), definiować tablice typów.

### 5.5.1 Deklaracja struktury

TwinCAT pozwala na tworzenie własnych struktur. Polami struktur mogą być zmienne typów standardowych lub stworzonych przez użytkownika. Struktura jest dobrym sposobem na grupowanie zmiennych odnoszących się do jednego obiektu np. zmienne służące do sterowania zaworu.

<b>Tworzenie struktury</b>	
	<p>Tworzenie struktury odbywa się w Menagerze organizacji projektu w zakładce <b>Data Types</b> poprzez wybranie z menu kontekstowego opcji <b>Add Object...</b></p>
	<p>W oknie <b>New data type</b> należy wpisać nazwę struktury</p>
<b>Definicja struktury</b>	
<pre> 0001 TYPE ST_Zawor : 0002 STRUCT 0003     sNazwa : STRING; 0004     iStopien_otwarcia : INT; 0005     bOtwarty : BOOL; 0006     bZamkniety : BOOL; 0007     bOtworz : BOOL; 0008     bZamknij : BOOL; 0009     bAwaria : BOOL; 0010 END_STRUCT 0011 END_TYPE </pre>	<p>Po wpisaniu nazwy utworzy się okno definicji struktury. Pomiędzy słowa kluczowe STRUCT END_STRUCT należy wpisać wszystkie pola struktury.</p>
<b>Deklaracja struktury</b>	
<pre> 0002 VAR 0003 (* deklaracja zmiennej typu ST_Zawor *) 0004     stZawor_1 : ST_Zawor; 0005     stInnyZawor : ST_Zawor := (sNazwa:='Z1', 0006                               iStopien_otwarcia:=50, 0007                               bAwaria:=FALSE); 0008 END_VAR </pre>	<p>Deklaracja struktury odbywa się tak jak deklaracja zmiennej typu standardowego, czyli najpierw nazwa zmiennej a potem nazwa typu.</p> <p>Strukturę można inicjalizować wartościami początkowymi wpisując w deklaracji w nawiasach nazwy i wartości pól oddzielone przecinkami. Pola pominięte będą inicjalizowane wartościami domyślnymi.</p>



<b>Odwołanie się do pola struktury</b>	
	<p>Do konkretnego pola struktury można się odwołać wpisując nazwę zmiennej kropkę i nazwę wybranego pola. Po wpisaniu kropki wyświetlana jest lista, z której można wybrać odpowiednie pole.</p>

### 5.5.2 Deklaracja typu ENUM

Typ wyliczeniowy (Enumeration) służy do ograniczenia możliwości wyboru wartości. Zmienna ENUM ogranicza wybór do kilku/kilkunastu ściśle określonych przez programistę elementów oraz do odwoływania się do tych wartości za pomocą zdefiniowanej nazwy. Wewnątrz definicji typu wyliczeniowego umieszczona jest lista stałych zmiennych. Tylko te zmienne będzie można przechowywać w zdefiniowanym typie enum. Warto wiedzieć, że stałe, które wprowadzono w definicji typu wyliczeniowego reprezentują liczby całkowite. Każda nazwa posiada kolejną wartość. Domyślnie pierwszy element przyjmuje wartość zero a kolejne nazwy wartości o jedną większą. Programista może nadać również wartości nazwie, każda kolejna będzie miała wartość o jeden większą. Poniżej została przedstawiona definicja i deklaracja zmiennej typu enum.

<p>Data types</p> <ul style="list-style-type: none"> <li>E_Operacja (ENUM)</li> </ul>	<pre> 0001 TYPE E_Operacja : (Szlifowanie, Malowanie, Wiercenie:=10, Obrót, Toczanie:=45, Polerowanie ); 0002 END_TYPE 0003 0004 0005 0006 0007 0008 0009 0010 0011 0012 0013 0014 0015 0016 0017 0018 0019 0020 0021 0022 0023 0024 0025 0026 0027 0028 0029 0030 0031 0032 0033 0034 0035 0036 0037 0038 0039 0040 0041 0042 0043 0044 0045 0046 0047 0048 0049 0050 0051 0052 0053 0054 0055 0056 0057 0058 0059 0060 0061 0062 0063 0064 0065 0066 0067 0068 0069 0070 0071 0072 0073 0074 0075 0076 0077 0078 0079 0080 0081 0082 0083 0084 0085 0086 0087 0088 0089 0090 0091 0092 0093 0094 0095 0096 0097 0098 0099 0100 0101 0102 0103 0104 0105 0106 0107 0108 0109 0110 0111 0112 0113 0114 0115 0116 0117 0118 0119 0120 0121 0122 0123 0124 0125 0126 0127 0128 0129 0130 0131 0132 0133 0134 0135 0136 0137 0138 0139 0140 0141 0142 0143 0144 0145 0146 0147 0148 0149 0150 0151 0152 0153 0154 0155 0156 0157 0158 0159 0160 0161 0162 0163 0164 0165 0166 0167 0168 0169 0170 0171 0172 0173 0174 0175 0176 0177 0178 0179 0180 0181 0182 0183 0184 0185 0186 0187 0188 0189 0190 0191 0192 0193 0194 0195 0196 0197 0198 0199 0200 0201 0202 0203 0204 0205 0206 0207 0208 0209 0210 0211 0212 0213 0214 0215 0216 0217 0218 0219 0220 0221 0222 0223 0224 0225 0226 0227 0228 0229 0230 0231 0232 0233 0234 0235 0236 0237 0238 0239 0240 0241 0242 0243 0244 0245 0246 0247 0248 0249 0250 0251 0252 0253 0254 0255 0256 0257 0258 0259 0260 0261 0262 0263 0264 0265 0266 0267 0268 0269 0270 0271 0272 0273 0274 0275 0276 0277 0278 0279 0280 0281 0282 0283 0284 0285 0286 0287 0288 0289 0290 0291 0292 0293 0294 0295 0296 0297 0298 0299 0300 0301 0302 0303 0304 0305 0306 0307 0308 0309 0310 0311 0312 0313 0314 0315 0316 0317 0318 0319 0320 0321 0322 0323 0324 0325 0326 0327 0328 0329 0330 0331 0332 0333 0334 0335 0336 0337 0338 0339 0340 0341 0342 0343 0344 0345 0346 0347 0348 0349 0350 0351 0352 0353 0354 0355 0356 0357 0358 0359 0360 0361 0362 0363 0364 0365 0366 0367 0368 0369 0370 0371 0372 0373 0374 0375 0376 0377 0378 0379 0380 0381 0382 0383 0384 0385 0386 0387 0388 0389 0390 0391 0392 0393 0394 0395 0396 0397 0398 0399 0400 0401 0402 0403 0404 0405 0406 0407 0408 0409 0410 0411 0412 0413 0414 0415 0416 0417 0418 0419 0420 0421 0422 0423 0424 0425 0426 0427 0428 0429 0430 0431 0432 0433 0434 0435 0436 0437 0438 0439 0440 0441 0442 0443 0444 0445 0446 0447 0448 0449 0450 0451 0452 0453 0454 0455 0456 0457 0458 0459 0460 0461 0462 0463 0464 0465 0466 0467 0468 0469 0470 0471 0472 0473 0474 0475 0476 0477 0478 0479 0480 0481 0482 0483 0484 0485 0486 0487 0488 0489 0490 0491 0492 0493 0494 0495 0496 0497 0498 0499 0500 0501 0502 0503 0504 0505 0506 0507 0508 0509 0510 0511 0512 0513 0514 0515 0516 0517 0518 0519 0520 0521 0522 0523 0524 0525 0526 0527 0528 0529 0530 0531 0532 0533 0534 0535 0536 0537 0538 0539 0540 0541 0542 0543 0544 0545 0546 0547 0548 0549 0550 0551 0552 0553 0554 0555 0556 0557 0558 0559 0560 0561 0562 0563 0564 0565 0566 0567 0568 0569 0570 0571 0572 0573 0574 0575 0576 0577 0578 0579 0580 0581 0582 0583 0584 0585 0586 0587 0588 0589 0590 0591 0592 0593 0594 0595 0596 0597 0598 0599 0600 0601 0602 0603 0604 0605 0606 0607 0608 0609 0610 0611 0612 0613 0614 0615 0616 0617 0618 0619 0620 0621 0622 0623 0624 0625 0626 0627 0628 0629 0630 0631 0632 0633 0634 0635 0636 0637 0638 0639 0640 0641 0642 0643 0644 0645 0646 0647 0648 0649 0650 0651 0652 0653 0654 0655 0656 0657 0658 0659 0660 0661 0662 0663 0664 0665 0666 0667 0668 0669 0670 0671 0672 0673 0674 0675 0676 0677 0678 0679 0680 0681 0682 0683 0684 0685 0686 0687 0688 0689 0690 0691 0692 0693 0694 0695 0696 0697 0698 0699 0700 0701 0702 0703 0704 0705 0706 0707 0708 0709 0710 0711 0712 0713 0714 0715 0716 0717 0718 0719 0720 0721 0722 0723 0724 0725 0726 0727 0728 0729 0730 0731 0732 0733 0734 0735 0736 0737 0738 0739 0740 0741 0742 0743 0744 0745 0746 0747 0748 0749 0750 0751 0752 0753 0754 0755 0756 0757 0758 0759 0760 0761 0762 0763 0764 0765 0766 0767 0768 0769 0770 0771 0772 0773 0774 0775 0776 0777 0778 0779 0780 0781 0782 0783 0784 0785 0786 0787 0788 0789 0790 0791 0792 0793 0794 0795 0796 0797 0798 0799 0800 0801 0802 0803 0804 0805 0806 0807 0808 0809 0810 0811 0812 0813 0814 0815 0816 0817 0818 0819 0820 0821 0822 0823 0824 0825 0826 0827 0828 0829 0830 0831 0832 0833 0834 0835 0836 0837 0838 0839 0840 0841 0842 0843 0844 0845 0846 0847 0848 0849 0850 0851 0852 0853 0854 0855 0856 0857 0858 0859 0860 0861 0862 0863 0864 0865 0866 0867 0868 0869 0870 0871 0872 0873 0874 0875 0876 0877 0878 0879 0880 0881 0882 0883 0884 0885 0886 0887 0888 0889 0890 0891 0892 0893 0894 0895 0896 0897 0898 0899 0900 0901 0902 0903 0904 0905 0906 0907 0908 0909 0910 0911 0912 0913 0914 0915 0916 0917 0918 0919 0920 0921 0922 0923 0924 0925 0926 0927 0928 0929 0930 0931 0932 0933 0934 0935 0936 0937 0938 0939 0940 0941 0942 0943 0944 0945 0946 0947 0948 0949 0950 0951 0952 0953 0954 0955 0956 0957 0958 0959 0960 0961 0962 0963 0964 0965 0966 0967 0968 0969 0970 0971 0972 0973 0974 0975 0976 0977 0978 0979 0980 0981 0982 0983 0984 0985 0986 0987 0988 0989 0990 0991 0992 0993 0994 0995 0996 0997 0998 0999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1330 1331 1332 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 1343 1344 1345 1346 1347 1348 1349 1350 1351 1352 1353 1354 1355 1356 1357 1358 1359 1360 1361 1362 1363 1364 1365 1366 1367 1368 1369 1370 1371 1372 1373 1374 1375 1376 1377 1378 1379 1380 1381 1382 1383 1384 1385 1386 1387 1388 1389 1390 1391 1392 1393 1394 1395 1396 1397 1398 1399 1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410 1411 1412 1413 1414 1415 1416 1417 1418 1419 1420 1421 1422 1423 1424 1425 1426 1427 1428 1429 1430 1431 1432 1433 1434 1435 1436 1437 1438 1439 1440 1441 1442 1443 1444 1445 1446 1447 1448 1449 1450 1451 1452 1453 1454 1455 1456 1457 1458 1459 1460 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 1471 1472 1473 1474 1475 1476 1477 1478 1479 1480 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490 1491 1492 1493 1494 1495 1496 1497 1498 1499 1500 1501 1502 1503 1504 1505 1506 1507 1508 1509 1510 1511 1512 1513 1514 1515 1516 1517 1518 1519 1520 1521 1522 1523 1524 1525 1526 1527 1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544 1545 1546 1547 1548 1549 1550 1551 1552 1553 1554 1555 1556 1557 1558 1559 1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1573 1574 1575 1576 1577 1578 1579 1580 1581 1582 1583 1584 1585 1586 1587 1588 1589 1590 1591 1592 1593 1594 1595 1596 1597 1598 1599 1600 1601 1602 1603 1604 1605 1606 1607 1608 1609 1610 1611 1612 1613 1614 1615 1616 1617 1618 1619 1620 1621 1622 1623 1624 1625 1626 1627 1628 1629 1630 1631 1632 1633 1634 1635 1636 1637 1638 1639 1640 1641 1642 1643 1644 1645 1646 1647 1648 1649 1650 1651 1652 1653 1654 1655 1656 1657 1658 1659 1660 1661 1662 1663 1664 1665 1666 1667 1668 1669 1670 1671 1672 1673 1674 1675 1676 1677 1678 1679 1680 1681 1682 1683 1684 1685 1686 1687 1688 1689 1690 1691 1692 1693 1694 1695 1696 1697 1698 1699 1700 1701 1702 1703 1704 1705 1706 1707 1708 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1720 1721 1722 1723 1724 1725 1726 1727 1728 1729 1730 1731 1732 1733 1734 1735 1736 1737 1738 1739 1740 1741 1742 1743 1744 1745 1746 1747 1748 1749 1750 1751 1752 1753 1754 1755 1756 1757 1758 1759 1760 1761 1762 1763 1764 1765 1766 1767 1768 1769 1770 1771 1772 1773 1774 1775 1776 1777 1778 1779 1780 1781 1782 1783 1784 1785 1786 1787 1788 1789 1790 1791 1792 1793 1794 1795 1796 1797 1798 1799 1800 1801 1802 1803 1804 1805 1806 1807 1808 1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823 1824 1825 1826 1827 1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1839 1840 1841 1842 1843 1844 1845 1846 1847 1848 1849 1850 1851 1852 1853 1854 1855 1856 1857 1858 1859 1860 1861 1862 1863 1864 1865 1866 1867 1868 1869 1870 1871 1872 1873 1874 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889 1890 1891 1892 1893 1894 1895 1896 1897 1898 1899 1900 1901 1902 1903 1904 1905 1906 1907 1908 1909 1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 1930 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944 1945 1946 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196 2197 2198 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 2231 2232 2233 2234 2235 2236 2237 2238 2239 2240 2241 2242 2243 2244 2245 2246 2247 2248 2249 2250 2251 2252 2253 2254 2255 2256 2257 2258 2259 2260 2261 2262 2263 2264 2265 2266 2267 2268 2269 2270 2271 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2287 2288 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298 2299 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 231</pre>
---	--

### 5.5.3 Deklaracja tablicy

Tablica przechowuje wartości kilku zmiennych tego samego typu. Można zadeklarować tablice jedno-, dwu-, trójwymiarowe. Tablice mogą zawierać elementy dowolnego typu. Przykład deklaracji tablicy przedstawiono poniżej:

<b>Przykładowe deklaracje tablic</b>	
(* tablica jednowymiarowa *) (*nazwa : słowo kluczowe [rozmiar: low.. max] of typ.*) aiTablica: <b>ARRAY</b> [1.. 10] <b>OF INT</b> ;	deklaracja tablicy jednowymiarowej typu INT
(* tablica dwuwymiarowa *) abyTabDwuWym: <b>ARRAY</b> [1.. 10, 2.. 5] <b>OF BYTE</b> ;	deklaracja tablicy dwuwymiarowej typu BYTE
(* tablica trójwymiarowa *) awTabTrzyWym: <b>ARRAY</b> [1.. 10, 2.. 5, 0.. 10] <b>OF WORD</b> ;	deklaracja tablicy trójwymiarowej typu INT
(* tablica struktur *) astTabOfStruct: <b>ARRAY</b> [1.. 10] <b>OF ST_Zawor</b> ;	deklaracja tablicy jednowymiarowej struktur typu ST_Zawor

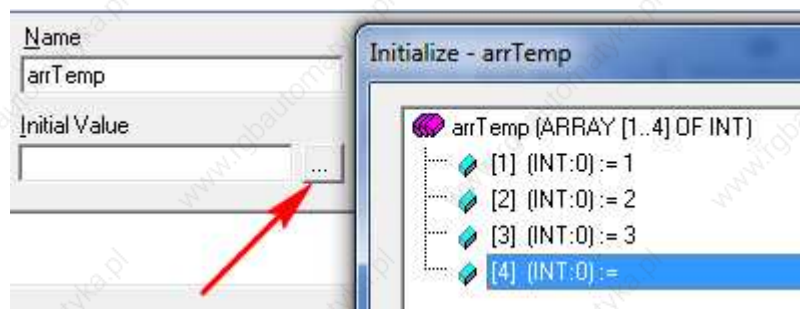
#### **Inicjalizacja wartości początkowych tablicy**

Wartości początkowe można wprowadzić:

- W polu deklaracji zmiennych – wpisując ręcznie wartości startowe:

(*inicjalizacja tablicy*) adwTab: <b>ARRAY</b> [0.. 3] <b>OF DWORD</b> := 1,2,3,4;	inicjalizacja elementów tablicy wartościami 1,2, 3,4.
astZawory: <b>ARRAY</b> [1..10] <b>OF ST_Zawor</b> := (sNazwa:='Z1',iStopien_otwarcia:=50,bAwaria:= <b>FALSE</b> ), (sNazwa:='Z2',iStopien_otwarcia:=34), (sNazwa:='Z3',iStopien_otwarcia:=100), (sNazwa:='Z4',iStopien_otwarcia:=0), (sNazwa:='Z5',iStopien_otwarcia:=56,bAwaria:= <b>TRUE</b> ), (sNazwa:='Z6',iStopien_otwarcia:=63);	

- W oknie deklaracji zmiennej – wciskając symbol „...” obok pola Initial Value i uzupełniając odpowiednie pola:



**Odwołanie się do elementu tablicy**

Do tablicy można odwołać się podając za nazwą tablicy w nawiasach kwadratowych nr pola tablicy. Przykładowe odwołanie się do elementu tablicy przedstawiono poniżej

<code>aiTablica[1] := 4;</code>	wpisanie do pierwszego elementu tablicy jednowymiarowej wartości 4
<code>byTemp := abyTabDwuWym[1,2];</code>	przepisanie elementu [1,2] tablicy dwuwymiarowej do zmiennej byTemp
<code>astZawory[3].sNazwa := 'Zawor 1';</code>	odwołanie się do pola struktury umieszczonej w 3 elemencie tablicy

**Przykład wykorzystania tablicy**

W tablicy umieszcza się wiele elementów tego samego typu. Tablice doskonale nadają się zarządzania wieloma elementami. **Wykorzystując pętle for można szybko i wygodnie zarządzać wszystkimi wartościami w tablicy.** Instrukcja często wykorzystywana jest do przeglądania, sortowania, wyszukiwania, zmieniania elementów tablicy. Poniżej pokazany jest sposób ustawiania wartości zmiennej we wszystkich elementach tablicy:

```

008 FOR i:= 1 TO 10 DO
009     astZawory[i].bZamknij := TRUE;
010 END_FOR

lub

008 FOR i:= 1 TO 10 DO
009     astZawory[i].bZamknij := bPolecenie;
010 END_FOR

```

Stworzony skrypt ma za zadanie sterować dziesięcioma zaworami. Każdy zawór reprezentowany przez strukturę ST\_Zawor przechowującą wszystkie zmienne sterujące zaworami. Wszystkie struktury zgromadzone są w tablicy struktur astZawory. Jeden element tablicy dotyczy jednego zaworu. Zadaniem przedstawionego fragmentu kodu jest przesłać sygnał zamknięcia do wszystkich zaworów. Pętla przechodzi zatem po wszystkich elementach tablicy i ustawia wartość TRUE w polu struktury bZamknij. Zmienna „i” zwana jest iteratorem pętli FOR.

## 6 Uruchomienie programu, linkowanie zmiennych

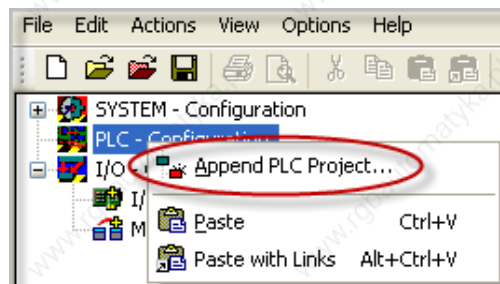
### 6.1 Kompilacja programu

Przed kompilacją programu, projekt należy zapisać na dysku. Najlepiej stworzyć sobie osobny katalog do każdego projektu, ponieważ w miejscu zapisu projektu kompilator utworzy pliki pomocnicze. Kompilacja rozpocznie się po wybraniu z menu opcji **Project → Rebuild**. Po kompilacji w oknie informacyjnym zostanie wyświetlone podsumowanie procesu kompilacji. Jeśli kompilator nie zgłasza żadnych błędów program może zostać wgrany na sterownik. Po poprawnej kompilacji automatycznie zostanie utworzony plik z rozszerzeniem **\*.tpy**. Zostaje on zapisany w tej samej lokalizacji co projekt. Plik \*.tpy zawiera różnorodne informacje, przechowuje również wszystkie szczegóły dotyczące zmiennych zewnętrznych (zmienne te w deklaracji zawierają słowo kluczowe AT). Chcąc połączyć zmienne zewnętrzne programu PLC z fizycznymi urządzeniami należy dodać plik \*.tpy w TwinCAT System Manager a następnie zlinkować dodane zmienne<sup>3</sup>.

Do kompilacji służy również polecenie **Project → Build (Ctrl+F8)**. Polecenie **Build** kompiluje tylko nowo wprowadzone lub zmodyfikowane fragmenty programu. Dzięki temu kompilacja trwa krócej. Czasem po kompilacji nie wszystkie wprowadzone zmiany będą działały. Należy zatem co pewien czas przebudować cały projekt kompilując go poleceniem **Rebuild all**.

### 6.2 Linkowanie zmiennych<sup>4</sup>

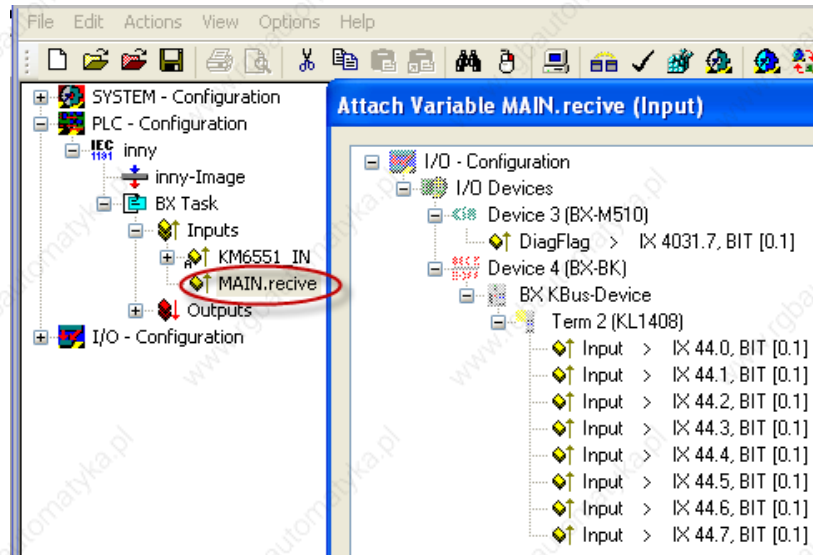
Plik \*.tpy należy dodać w programie **TwinCAT System Manager** do przygotowanej wcześniej konfiguracji sprzętowej. Klikając PPM na zakładkę PLC – Configuration z menu kontekstowego należy wybrać opcję **Append PLC Project...**




<sup>3</sup> Proces linkowania zmiennych opisany jest szczegółowo w oddzielnej dokumentacji do programu TwinCAT System Manager dostępnej na <ftp.beckhoff.com/poland/pomoc>

<sup>4</sup> jw

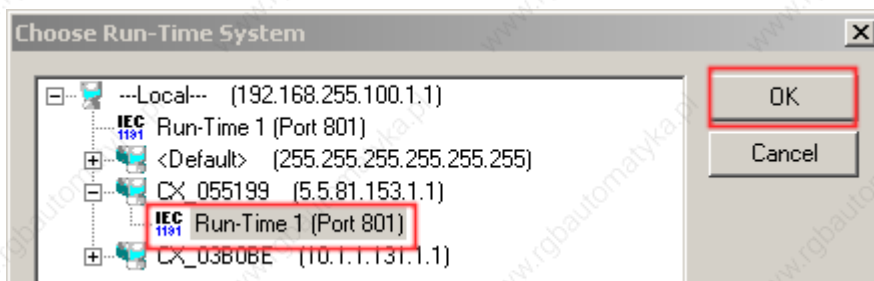
Po dodaniu pliku \*.tpy oraz po rozwinięciu zakładki PLC Configuration można zobaczyć zmienne zadeklarowane w programie. Zmienne te należy **zlinkować** z fizycznymi wejściami i wyjściami. W tym celu wystarczy dwukrotnie kliknąć na daną zmienną i z okna **Attach Variable** wybrać dostępne urządzenie np. wejście/wyjście cyfrowe.



Na koniec należy ponownie wgrać konfigurację, wybierając opcję **Activate Configuration** .

### 6.3 Uruchamianie programu

Gdy program jest poprawnie skompilowany, zmienne zlinkowane i wgrana konfiguracja można przystąpić do testowania programu. W pierwszej kolejności należy wybrać sterownik, na którym chcemy uruchomić program tzw. środowisko uruchomieniowe (run-time). W tym celu należy wybrać polecenie **Online → Choose Run-Time System...** Domyślnie wybrany jest Run-Time lokalny. Chcąc wgrać program na sterownik należy wybrać Run-Time właściwego sterownika.

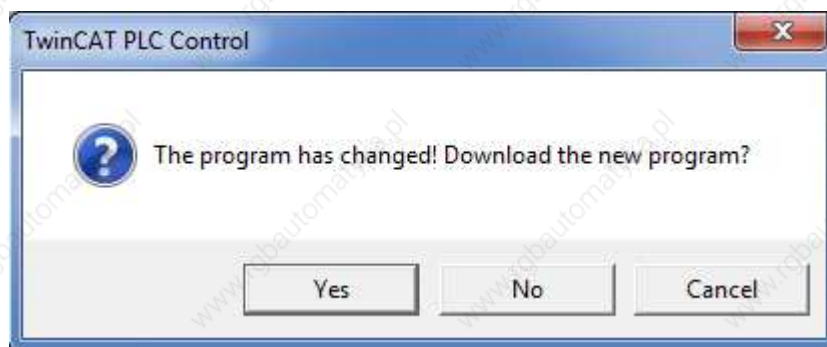


Lista sterowników wyświetlona w oknie **Choose Run-Time System** zawiera historię dotychczasowych połączeń ze sterownikami. Wybieramy właściwy Run-Time i

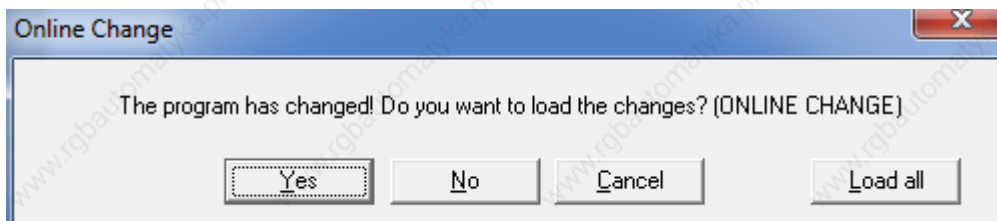
klikamy na nim dwukrotnie **LPM**. Od tej pory w prawym dolnym rogu okna głównego wyświetlana jest nazwa sterownika.

Target: CX 0939B4 (5.9.57.180.1.1), Run Time: 1 | TwinCAT Running

W dalszej kolejności za pomocą opcji **Online** → **Login (F11)** należy podjąć próbę zalogowania się na sterownik. W tym wypadku może wyświetlić się jeden z komunikatów:



Gdy nie ma programu na sterowniku lub gdy różni się on całkowicie od obecnego, pojawia się pytanie czy wgrać nowy program. Po potwierdzeniu program zostanie przesłany na sterownik automatycznie, ale nie zostanie on uruchomiony. Aby uruchomić program trzeba wykonać komendę **Online** → **Run (F5)**.



Podczas aktualizacji programu zostanie wyświetlone okno z propozycją zmiany online. W takim wypadku program znajdujący na sterowniku nie zatrzyma się. Możliwe jest wgranie programu od nowa za pomocą komendy **Load All** – zatrzyma to działanie programu. Zaleca się raz na kilka zmian online wykorzystywać rozkaz Load All.

Wgrany projekt jest przechowywany w pamięci ulotnej RAM. Jeżeli chcemy, aby był on projektem bootowalnym (uruchamiał się automatycznie po starcie sterownika) należy wybrać opcję **Online** → **Create Bootproject**.

## 7 Dodatek

### 7.1 Dodatkowa pomoc

<ftp://ftp.beckhoff.com/poland/Pomoc/> - adres serwera ftp na którym umieszczane są pomoce, programy, materiały szkoleniowe oraz funkcje diagnostyczne.

### 7.2 Akcje (Action)

Akcja jest to wyodrębniony fragment kodu programu, który może być wywoływany w dowolnym miejscu. Akcja może być napisana w innym języku niż język danego POU'sa. Stosowanie akcji pozwala uporządkować kod, ułatwia przez to debugowanie programu i pozwala szybciej zrozumieć algorytm osobie, która korzysta z kodu napisanego przez innego programistę. Akcja nie posiada własnych zmiennych, korzysta ze zmiennych danego POU'sa. Nie można utworzyć akcji w akcji.

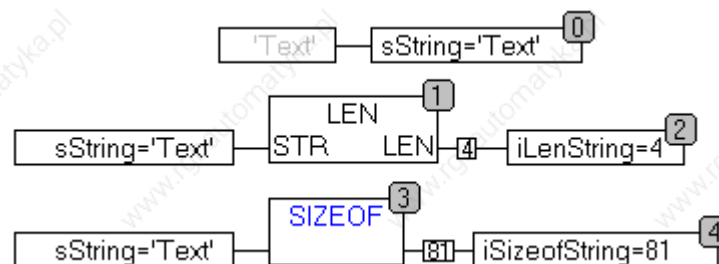
### 7.3 Operacje na Stringach

Zmienna typu STRING (łańcuch znaków) może mieć maksymalnie 255 znaków. Deklaracja może zawierać rozmiar zmiennej podany w nawiasach za typem, np. STRING(25), jeżeli rozmiar nie jest podany, to zmienna alokuje 80 bajtów na znaki. Dodatkowo jeden bajt jest zajmowany przez znak końca STRINGa.

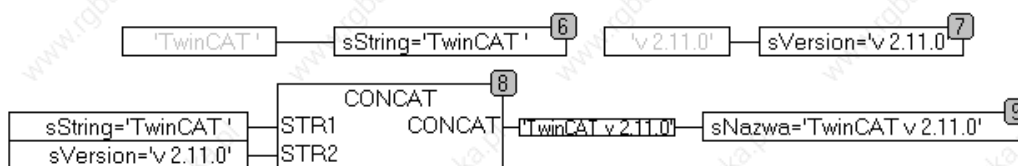
W bibliotece standardowej znajdują się funkcje, które pozwalają na wykonywanie operacji na STRINGach.

- **LEN** - zwraca liczbę znaków występujących w stringu. Funkcja oblicza długość aktualnego łańcucha znaków, bez znaku końca STRINGa.
- **SIZEOF** - zwraca ilość pamięci, którą kompilator zarezerwował dla zmiennej typu STRING. Funkcja zawsze zwraca maksymalną liczbę znaków, która można umieścić w STRINGu + 1 (znak końca STRINGa).

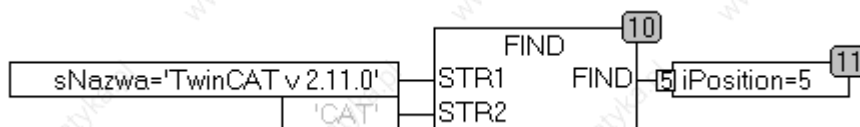
Rezultaty działania funkcji przedstawione są na rysunku poniżej.



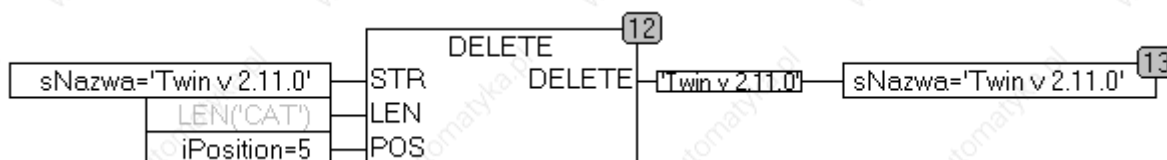
- **CONCAT** – służy do łączenia dwóch łańcuchów znaków



- **FIND** – znajduje jeden ciąg znaków w drugim ciągu znaków, zwraca pozycję pierwszego znaku znalezionej ciągu.



- **DELETE** – funkcja usuwa podaną liczbę znaków zaczynając od określonej pozycji



## 7.4 Znaki specjalne

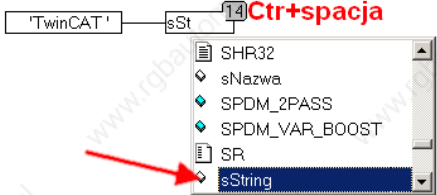
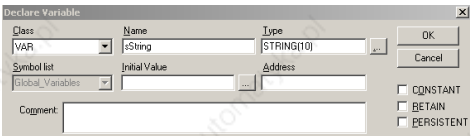
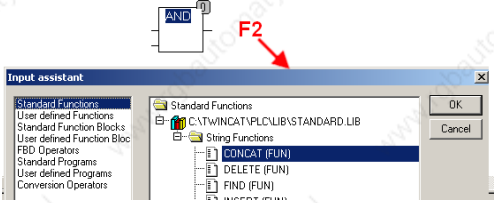
Wprowadzanie znaków specjalnych (takich, które mają przypisane określone funkcje) odbywa się przez wpisanie znaku \$ przed znakiem specjalnym. Poniżej przedstawiono przykłady użycia znaków specjalnych.

znak	opis
\$\$	znak dolara \$
\$'	apostrof
\$L lub \$l	nowy wiersz
\$N lub \$n	znak końca linii
\$P lub \$p	znak nowej strony
\$R lub \$r	znak przerwy w linii
\$T lub \$t	znak tabulatora



## 7.5 Skróty klawiszowe

Poniżej znajdują się wybrane skróty, które często wykorzystywane są podczas tworzenia programu:

<p>Ctrl + spacja</p> 	<p>Wywołuje menu kontekstowe, wyświetlające nazwy zmiennych, funkcji, bloków funkcyjnych, programu rozpoczynających się od danego ciągu znaków. Skróót szczególnie przydatny przy wywoływaniu zmiennych – zapobiega tzw. literówkom.</p>
<p>Shift + Esc</p>	<p>Otwieranie i zamykanie okna informacyjnego w TwinCAT PLC Control (na dole ekranu).</p>
<p>Shift + F2</p> 	<p>Wyświetlenie okna deklaracji zmiennych. Jeśli jakaś zadeklarowana zmienna została zaznaczona a skrót Shift +F2 został wywołany w oknie Declare Variable pola będą wypełnione – w ten sposób można edytować deklarację zmiennej</p>
<p>F2</p> 	<p>Wybrany przy zaznaczonej nazwie elementu wywołuje okno <b>Input assistant</b> wspomagającego wybór odpowiedniej nazwy bloku funkcyjnego, funkcji.</p>
<p>F4</p>	<p>Przenosi do następnego błędu wyświetlonego podczas kompilacji</p>
<p>Shift + F4</p>	<p>Przenosi do wcześniejszego błędu</p>

## 8 Przegląd najważniejszych opcji programu TwinCAT PLC Control




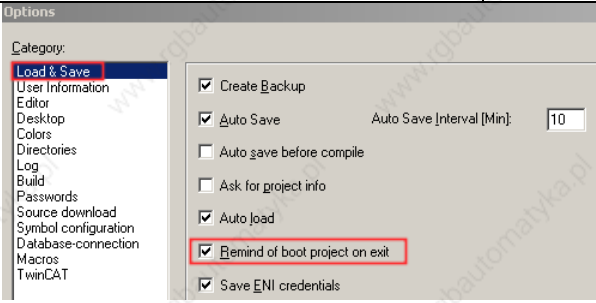
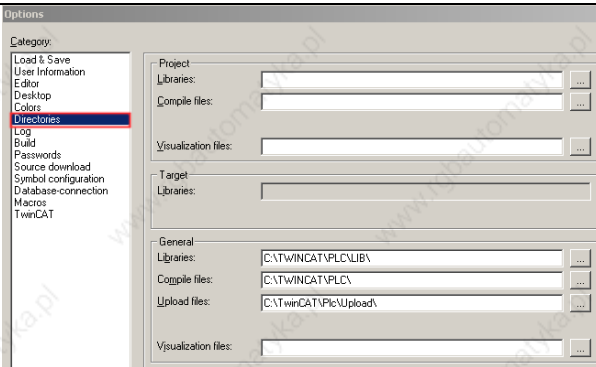
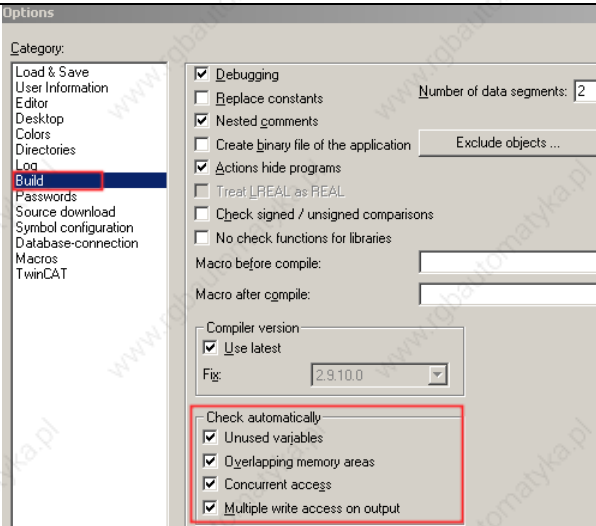
### 8.1 Menu File

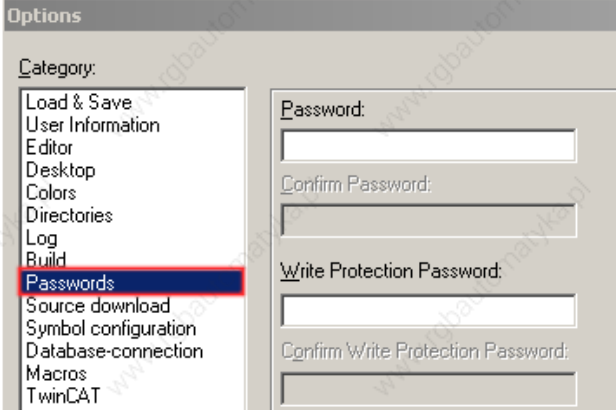
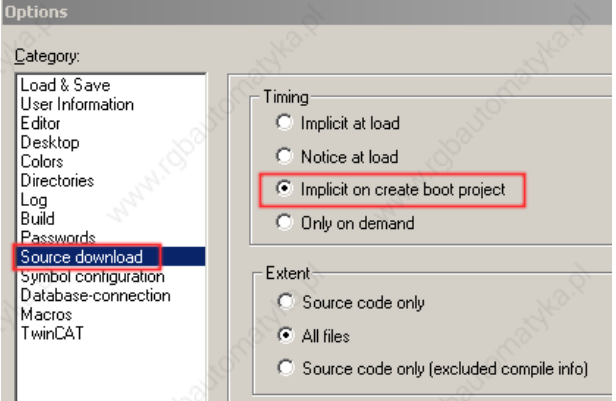
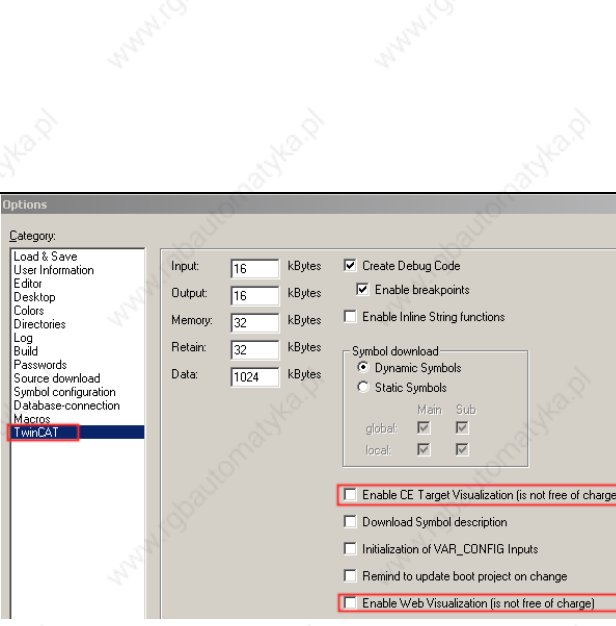
<b>N</b> ew	tworzenie nowego projektu
<b>O</b> pen...	(Ctrl+O) otwieranie projektu
<b>C</b> lose	zamykanie okna projektu
<b>S</b> ave	(Ctrl+S) zapisywanie projektu
<b>S</b> ave as...	zapisywanie projektu pod inną nazwą
<b>S</b> ave/Mail Archive...	archiwizowanie całego projektu, możliwość wysyłania projektu za pomocą maila
<b>P</b> rint	(Ctrl+P) drukowanie aktywnego okna programu
<b>P</b> rinter Setup...	Ustawienia drukowania
<b>E</b> xit	(Alt+F4) Zamknięcie programu TwinCAT PLC Control
<ul style="list-style-type: none"> <li>1 G:\pdf\AppNotes_Hayes\StepperMotor\Sample.pro</li> <li>2 C:\Documents and Settings\bober\Pulpit\Testowy\he</li> </ul>	Lista ostatnio otwartych projektów

### 8.2 Menu Edit

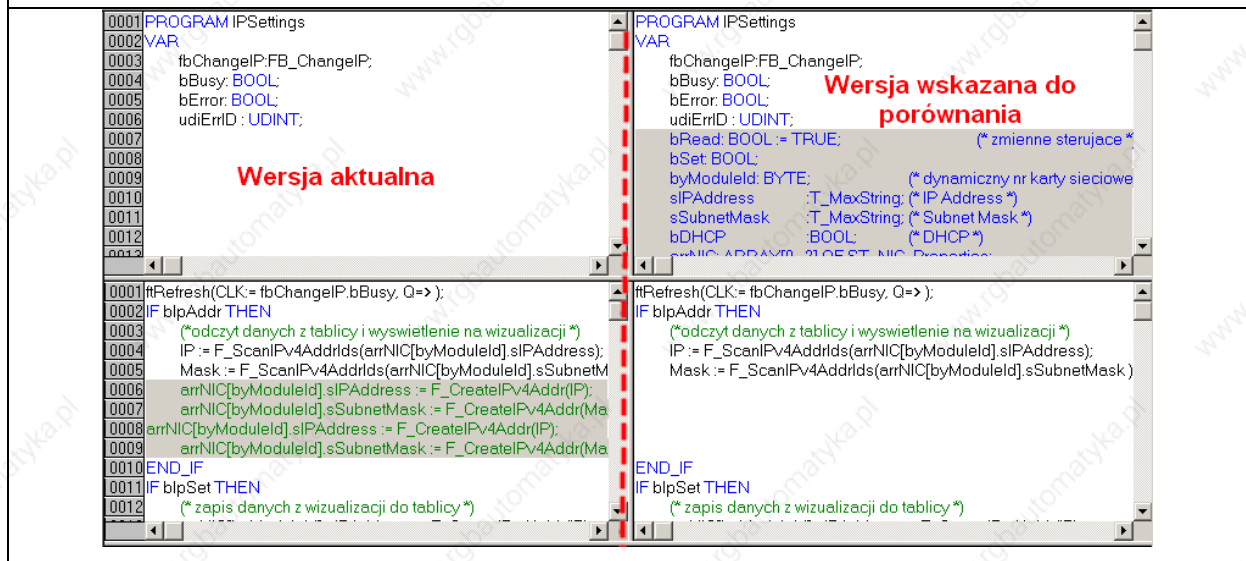
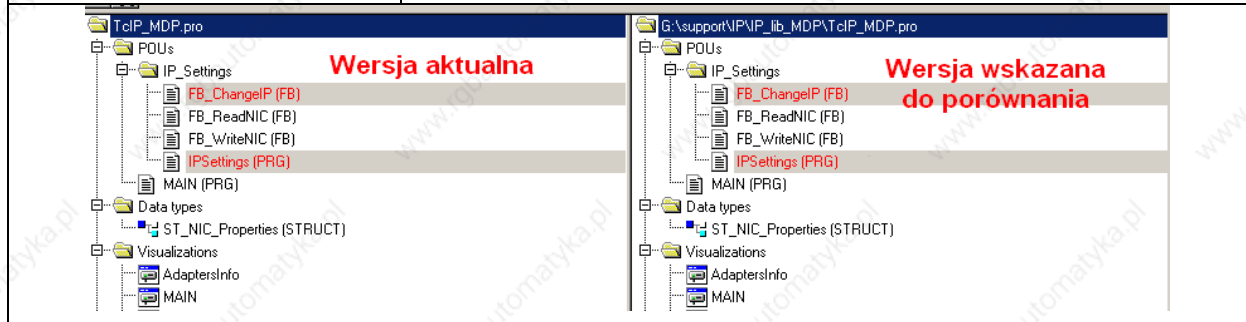
<ul style="list-style-type: none"> <li><b>U</b>ndo      Ctrl+Z</li> <li><b>R</b>edo      Ctrl+Y</li> </ul>	<p>Cofanie ostatniego kroku</p> <p>Ponowne wykonanie cofniętej funkcji</p>
<ul style="list-style-type: none"> <li><b>C</b>ut      Ctrl+X</li> <li><b>C</b>opy      Ctrl+C</li> <li><b>P</b>aste      Ctrl+V</li> <li><b>D</b>ele<del>t</del>e      Del</li> </ul>	<p>Wycinanie, kopiowanie, wklejanie, usuwanie zaznaczonego fragmentu programu</p>
<ul style="list-style-type: none"> <li><b>F</b>ind...      Ctrl+F</li> <li><b>F</b>ind next      F3</li> <li><b>R</b>eplace...      Ctrl+H</li> </ul>	<p>Szukanie w bieżącym oknie pierwszego pasującego ciągu znaków, szukanie kolejnego wystąpienia ciągu znaków, zastępowanie znalezionej ciągu znaków ciągiem podanym</p>
<ul style="list-style-type: none"> <li><b>I</b>ntput Assistant...      F2</li> <li><b>A</b>uto Declare...      Shift+F2</li> </ul>	<p>Wywołanie okna Input Assistant (asystent wyboru nazw i obiektów)</p> <p>Wywołanie okna Auto Declare (automatyczna deklaracja zmiennych).</p>
<ul style="list-style-type: none"> <li><b>N</b>ext Error      F4</li> <li><b>P</b>revious Error      Shift+F4</li> </ul>	<p>Przeniesienie w miejsce kolejnego błędu, wykrytego podczas kompilacji,</p> <p>Cofnięcie się do błędu wcześniejszego</p>

### 8.3 Menu Project

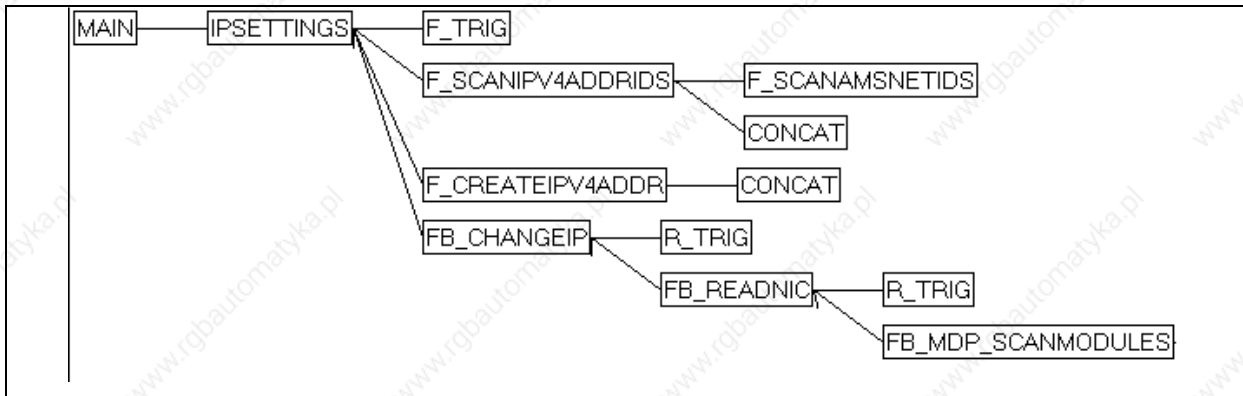
	<p><b>Build</b> – kompiluje tylko modyfikacje wprowadzone do projektu, natomiast <b>Rebuild all</b> kompiluje cały program od nowa</p>
	<p>Czyści pamięć i wszystkie pliki wykorzystywane do kompilacji oraz adresy przyznawane dynamicznie</p>
	<p>Opcje środowiska programistycznego</p>
	<p>Opcje dotyczących automatycznego zapisywania projektu oraz automatycznego wczytywania ostatniego edytowanego projektu po uruchomieniu TwinCAT. Zaleca się włączenie opcji Remind of boot Project on exit.</p>
	<p>Ustawienie ścieżek dostępu do katalogów przechowujących biblioteki. Domyślnie wybrane są katalogi instalacyjne TwinCAT'a. Domyślnie do tych katalogów instalowane są nowe biblioteki</p>
	<p>Ustawienia opcji kompilatora. Zaleca się włączenie opcji Check automatically:</p> <ul style="list-style-type: none"> <li>• Nieużywane zmienne</li> <li>• Nakładające się obszary pamięci</li> <li>• Jednoczesny dostęp</li> <li>• Nadpisywanie wyjść</li> </ul>

	<p>Możliwość zabezpieczania projektu:</p> <ul style="list-style-type: none"> <li>→ brak dostępu do projektu bez podania hasła</li> <li>→ dostęp tylko do odczytu, możliwość dokonania modyfikacji po podaniu hasła</li> </ul>
	<p>Ustawianie opcji wysyłania na sterownik całego projektu, wraz z kodem źródłowym i komentarzami.</p> <p>Domyślnie ustawiona jest opcja wgrzywania na żądanie, czyli po wybraniu opcji <b>Online</b> → <b>Sourcecode download</b>.</p> <p>Zaleca się ustawienie opcji <b>Implicite on create boot project</b> – wysyłania kodu programu podczas tworzenia projektu bootowalnego</p>
	<p>Zaznaczone opcje odpowiadają za wysyłanie wizualizacji do sterownika. Pierwsza opcja wspiera bibliotekę TargetVisu HMI – pozwala na wyświetlanie wizualizacji lokalnie. Druga opcja wspiera bibliotekę TargetVisu Web – pozwala wyświetlać wizualizację w przeglądarce internetowej.</p>
<p><b>Document...</b></p>	<p>Tworzenie dokumentacji dotyczącej programu. Opcja pozwala na wydrukowanie wybranych elementów projektu tj. kody programów, zmienne globalne, wizualizacje itp.</p>
<p><b>Export...</b> <b>Import...</b></p>	<p>Exportowanie służy do zapisania wybranych części</p>

	projektów do jednego pliku. Importowanie jest operacją odwrotną, pozwala na wczytanie wyexportowanego pliku.
<b>Merge...</b>	Służy do łączenia dwóch projektów. Po wybraniu opcji należy wskazać projekt, który będzie dołączony do aktualnego.
<b>Compare...</b>	Polecenie porównuje dwa projekty, obecny ze wskazanym poprzez podanie ścieżki. Opcja zaznacza kolorami różniące się części projektu. Poniżej pokazano porównanie dwóch projektów



<b>Global Search...</b> <span style="float: right;">Ctrl+Alt+S</span>	Wyszukiwanie ciągów znaków w całym projekcie np. zmiennych,
<b>Global Replace...</b>	
<b>Show Call Tree</b>	Call Tree przedstawia strukturę programu. Opcja staje się aktywna dopiero po poprawnym skompilowaniu programu i wybraniu programu.



Show Cross Reference...

Pokazuje miejsca wystąpienia oraz prawo dostępu podanej zmiennej, adresu, POU's'a. Cross references pozwala na przeniesienie się do miejsca występowania zmiennej.

POU	Variable	Address	Scope	Access
FB_CHANGEIP (38)	bBusy		Local	Read
FB_READNIC (8)	bBusy		Local	Write
FB_READNIC (13)	bBusy		Local	Write
FB_CHANGEIP (28)	bBusy		Local	Read
FB_CHANGEIP (52)	bBusy		Local	Read

Check ▸ Unused Variables  
 Overlapping Memory Areas  
 Concurrent Access  
 Multiple Write Access on Output

wyszukuje nieużywane zmienne

Check ▸ Unused Variables  
 Overlapping Memory Areas  
 Concurrent Access  
 Multiple Write Access on Output

wyszukuje nakładające się obszary pamięci

Check ▸ Unused Variables  
 Overlapping Memory Areas  
 Concurrent Access  
 Multiple Write Access on Output

sprawdza bieżący dostęp do zmiennej

Check ▸ Unused Variables  
 Overlapping Memory Areas  
 Concurrent Access  
 Multiple Write Access on Output

wykrywa nadpisywanie wyjść

Add Action

dodaje akcję do wybranego POU's'a

## 8.4 Menu Online

Login	(F11) logowanie się do wybranego sterownika
Logout	(F12) wylogowanie się ze sterownika
Run	(F5) uruchomienie programu na sterowniku
Stop	(Shift + F8) zatrzymanie działania programu
Reset	przywracanie programu do ustawień inicjalizacyjnych, zachowanie zmiennych nieulotnych
Reset All	przywracanie programu do ustawień inicjalizacyjnych, zerowanie również zmiennych nieulotnych
Toggle Breakpoint	(F9) ustawienie punktu zatrzymania programu, włączenie trybu krokowego wykonywania programu
Breakpoint Dialog	wywołanie okna zarządzania pracą debugera
Step over	(F10) praca krokowa, przejście do następnej wykonywanej instrukcji programu, nie wchodzimy do wnętrza elementów.
Step in	(F8) praca krokowa, przejście do następnej wykonywanej instrukcji programu, wchodzimy do wnętrza elementów
Single Cycle	(Ctrl + F5) wykonanie pojedynczego cyklu programu
Write Values	(Ctrl+ F7) wpisanie wartości do wybranej zmiennej, wartość zmiennej może być zmieniona w kolejnym cyklu
Force Values	(F7) forsowanie wartości wybranej zmiennej tzn. wpisanie wartości do zmiennej w każdym cyklu programu
Release Force	(Shift + F7) usuwanie wszystkich forsowań zmiennych
Show Call Stack...	wyświetlanie informacji o miejscu zatrzymania programu
Display Flow Control	(Shift + F11) podświetlenie linii wykonywanych w danej chwili
Simulation Mode	Tryb symulatora, opcja aktywna dla sterowników serii BC, BX
Sourcecode download	wgranie kodu źródłowego programu do wybranego sterownika
Choose Run-Time System...	wybór sterownika, do którego będzie można się zalogować
Create Bootproject	Tworzenie bootprojectu - po włączeniu zasilania sterownika, zostanie uruchomiony program wgrany na PLC
Delete Bootproject	Usunięcie bootprojectu, po włączeniu sterownika program wgrany do jego pamięci nie wystartuje automatycznie

