



# TwinCAT System

## Pierwsze kroki w TwinCAT System Manager i TwinCAT PLC Control





# TwinCAT - Total Windows Control and Automation Technology

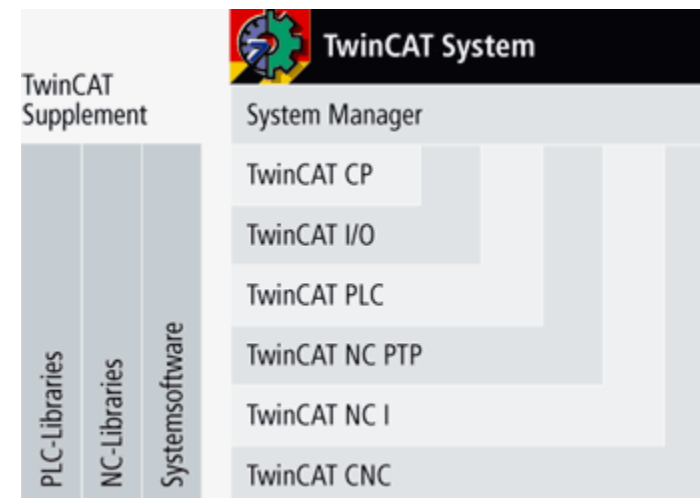
## TwinCAT System:

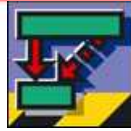
### ☐ TwinCAT System Manager

- o TwinCAT CP
- o TwinCAT I/O
- o TwinCAT PLC
- o TwinCAT NC PTP
- o TwinCAT NC I
- o TwinCAT CNC

### ☐ TwinCAT Supplement:

- o PLC – Libraries
- o NC – Libraries
- o Systemsoftware





# Konfiguracja urządzeń

## TwinCAT System Manager




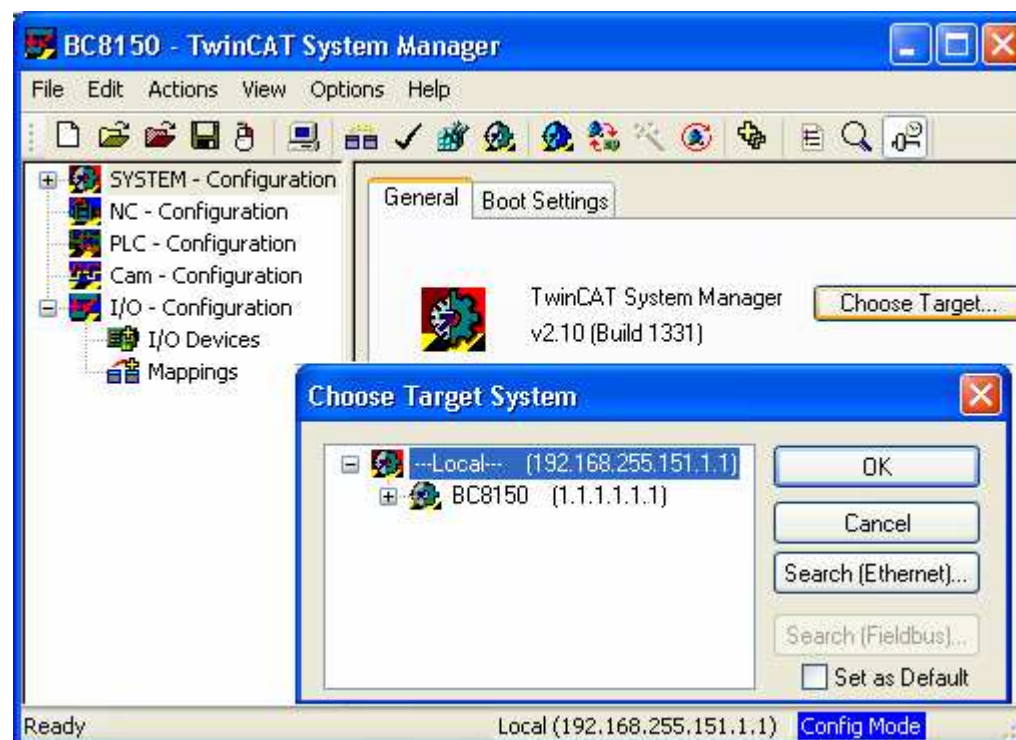


## Konfiguracja urządzenia – krok 1. Wybór obiektu

Wybór sterownika następuje w oknie Choose Target System

Dostęp do okna wyboru sterownika:

- ikona  na pasku narzędzi
- Actions\Choose Target System...
- klawisz F8
- SYSTEM-Configuration => General\Choose Target...
- Search (Ethernet) – jeżeli nie ma szukanego sterownika

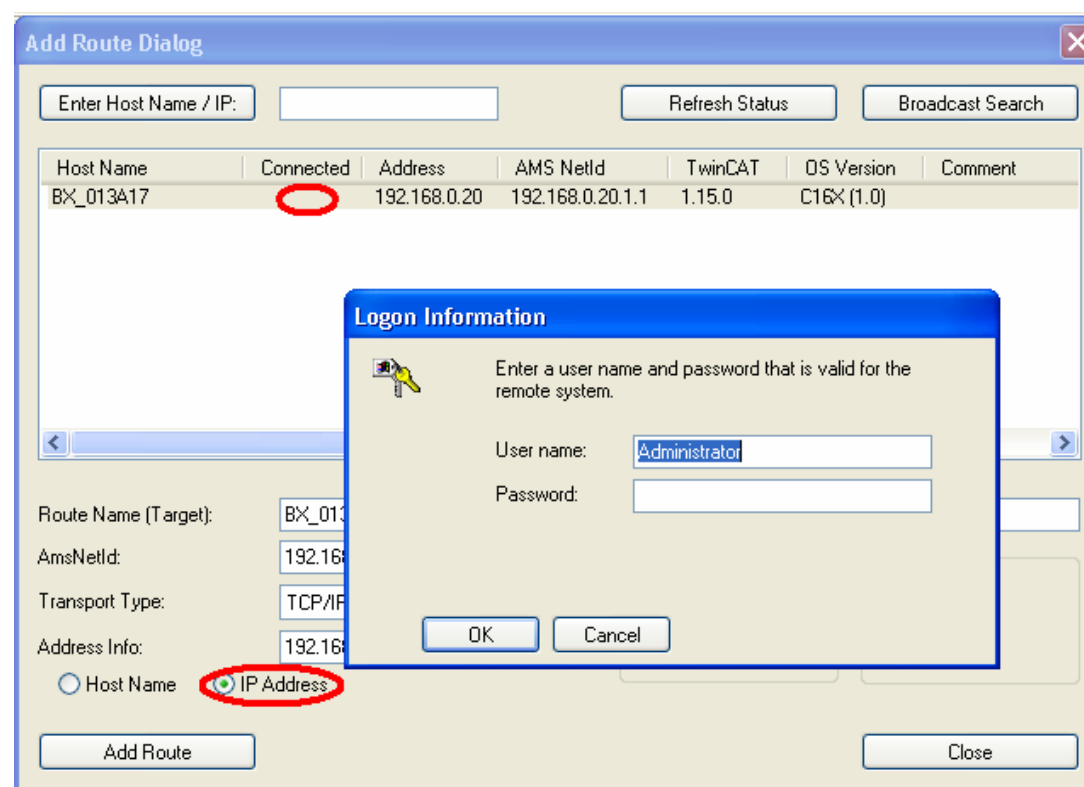




## Konfiguracja urządzenia – krok 1. Wybór obiektu – możliwe problemy

- Broadcast Search –  
znalezienie w sieci wszystkich  
obiektów z uruchomionym  
systemem TwinCAT.
- wybieramy sterownik,  
z którym chcemy się połączyć.
- połączenie nawiązujemy  
komendą Add Route (IP  
Address).
- nawiązanie połączenia  
sygnalizuje

Host Name	Connected
BX_013A17	X

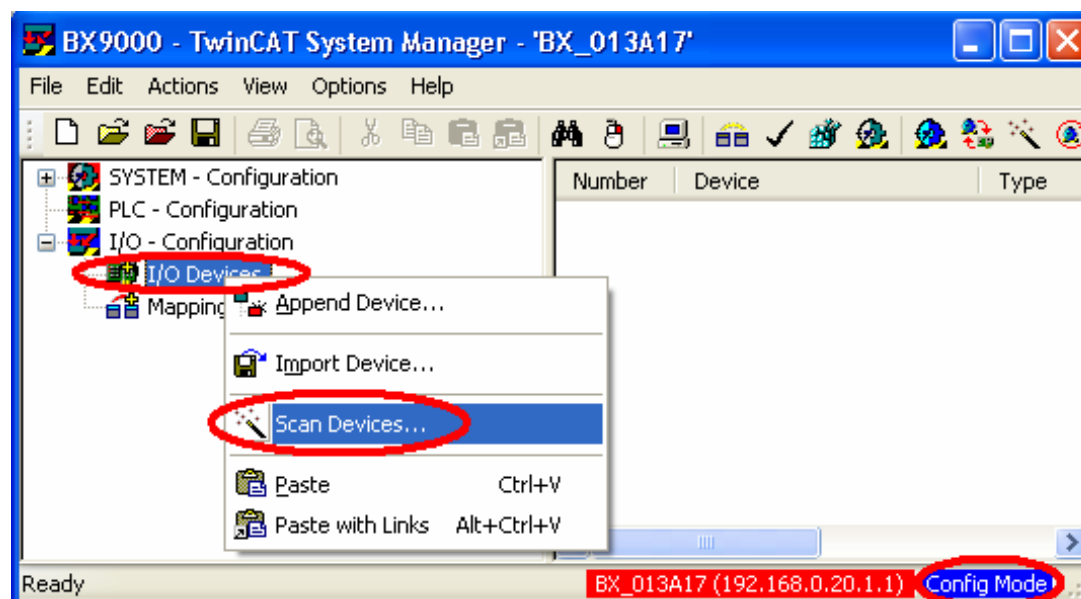




## Konfiguracja urządzenia – krok 2. Scan Devieces

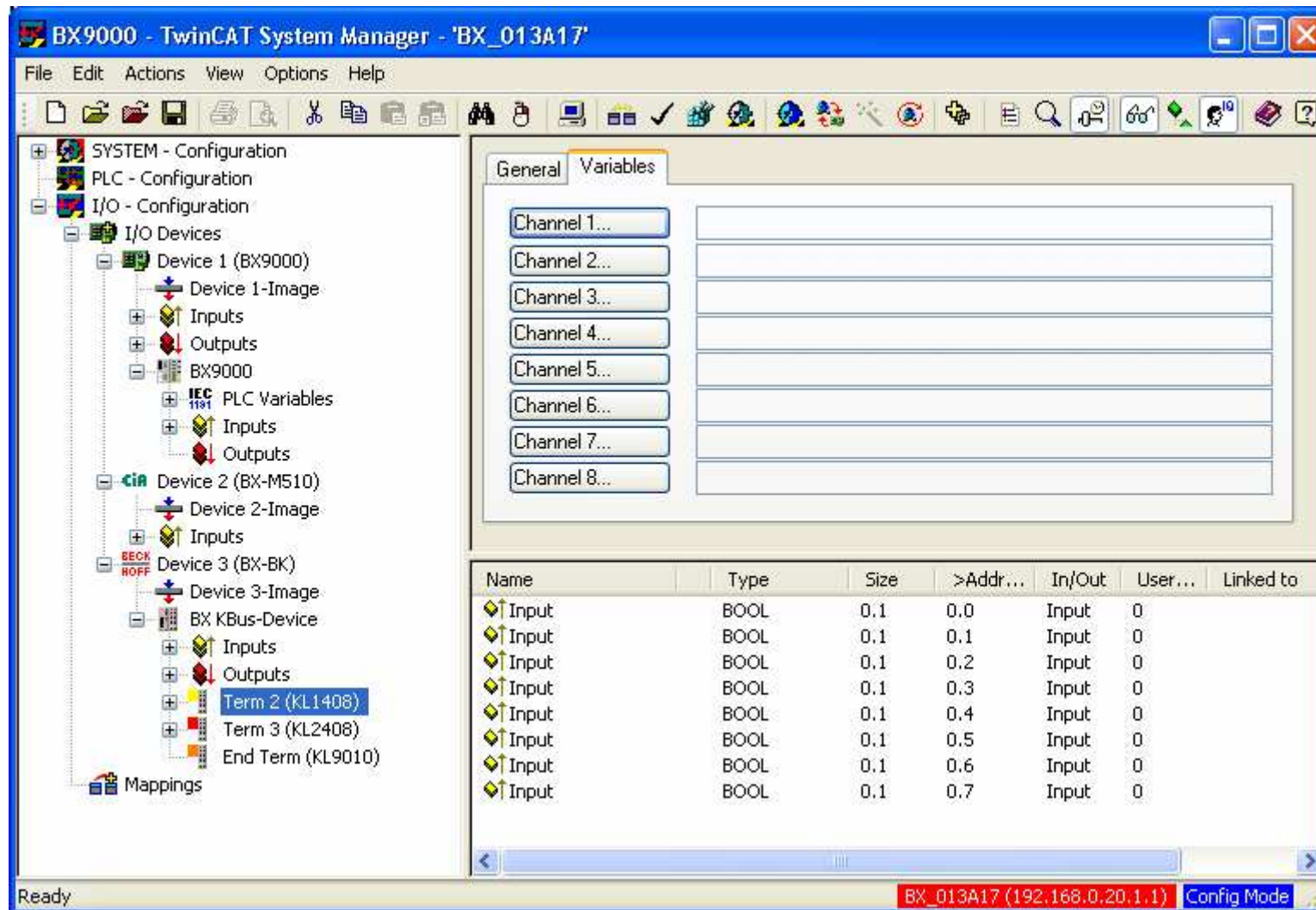
### Scan Devieces:

- wykrywa urządzenia połączone ze sterownikiem
- uruchamiana po kliknięciu prawym przyciskiem myszy na I/O Devieces
- aktywna w Config Mode





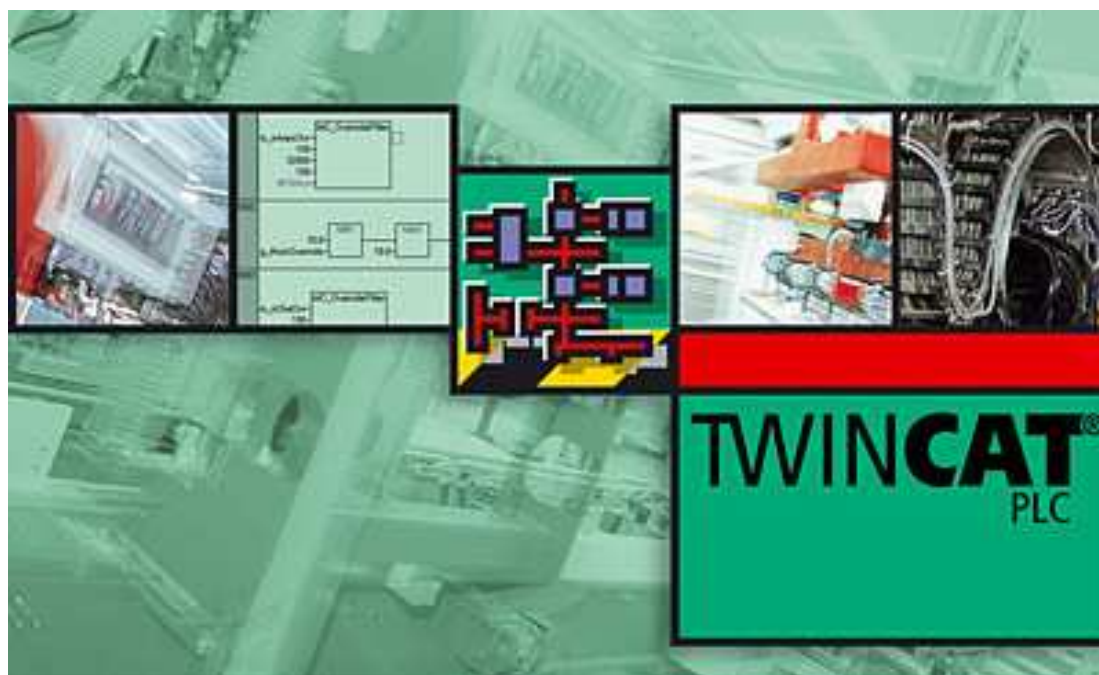
# Konfiguracja urządzenia - zakończenie





# TwinCAT PLC

## Tworzenie prostego programu



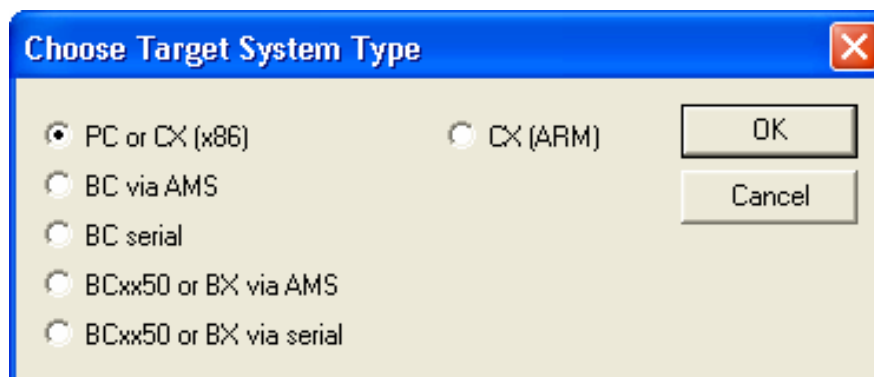




# Wybór obiektu docelowego

Krok 1.

Do wyboru jest sześć typów programów, zależnych od obiektu docelowego i sposobu komunikacji.

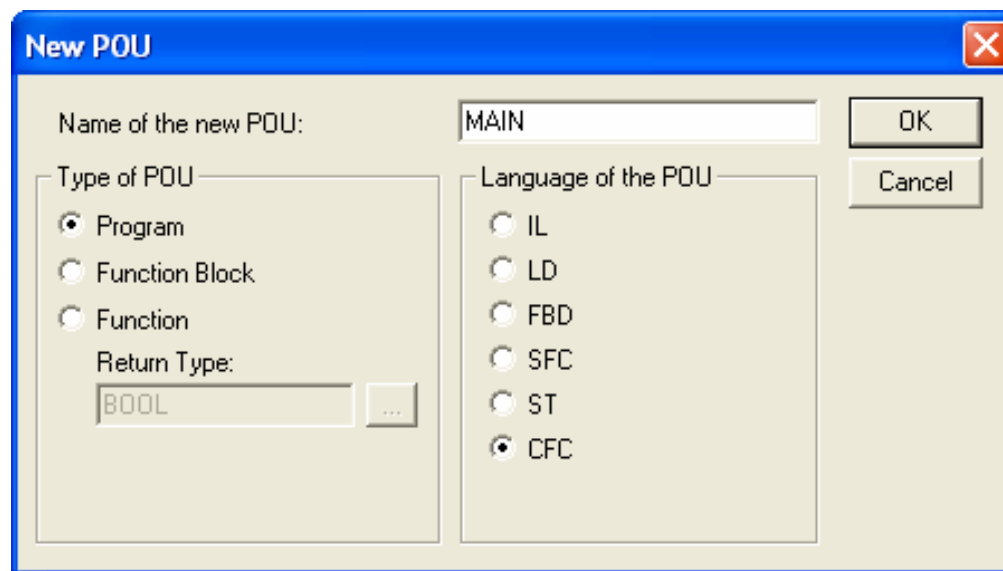


Podstawowe biblioteki zostaną dołączone automatycznie.



# Wybór języka programowania

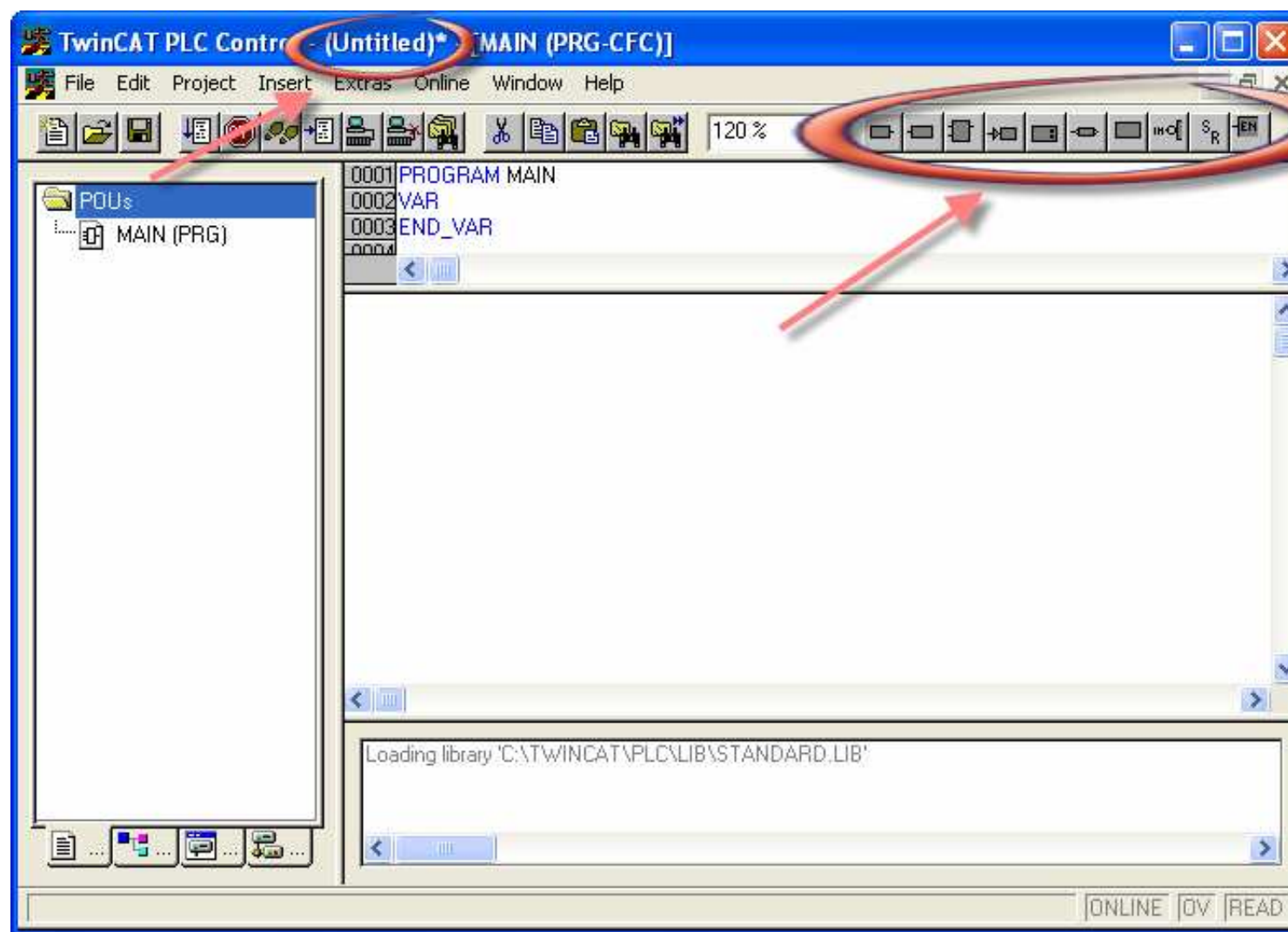
Krok 2.



Do wyboru jest sześć języków programowania, zgodnych ze standardem IEC 61131-3.

W oknie tym można wybrać również typ obiektu jaki zostanie stworzony: program, blok funkcyjny lub funkcja.

# Okno główne



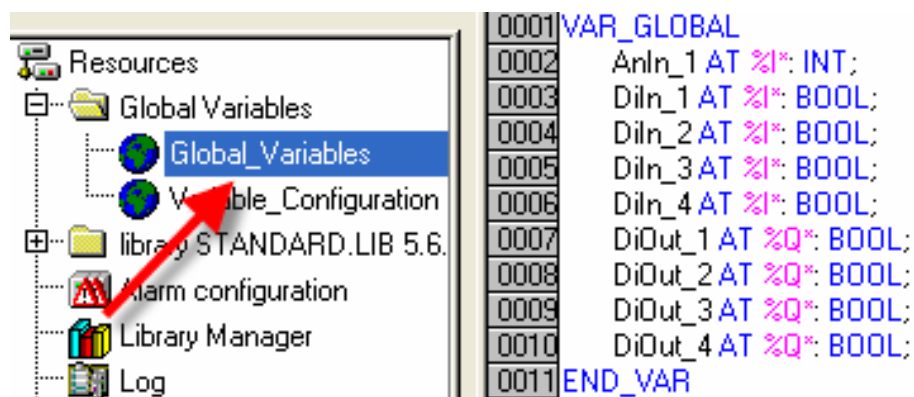


# Deklaracja zmiennych

Krok 3. Wybór zakładki Resources



Krok 4. Przejście do Global\_Variables



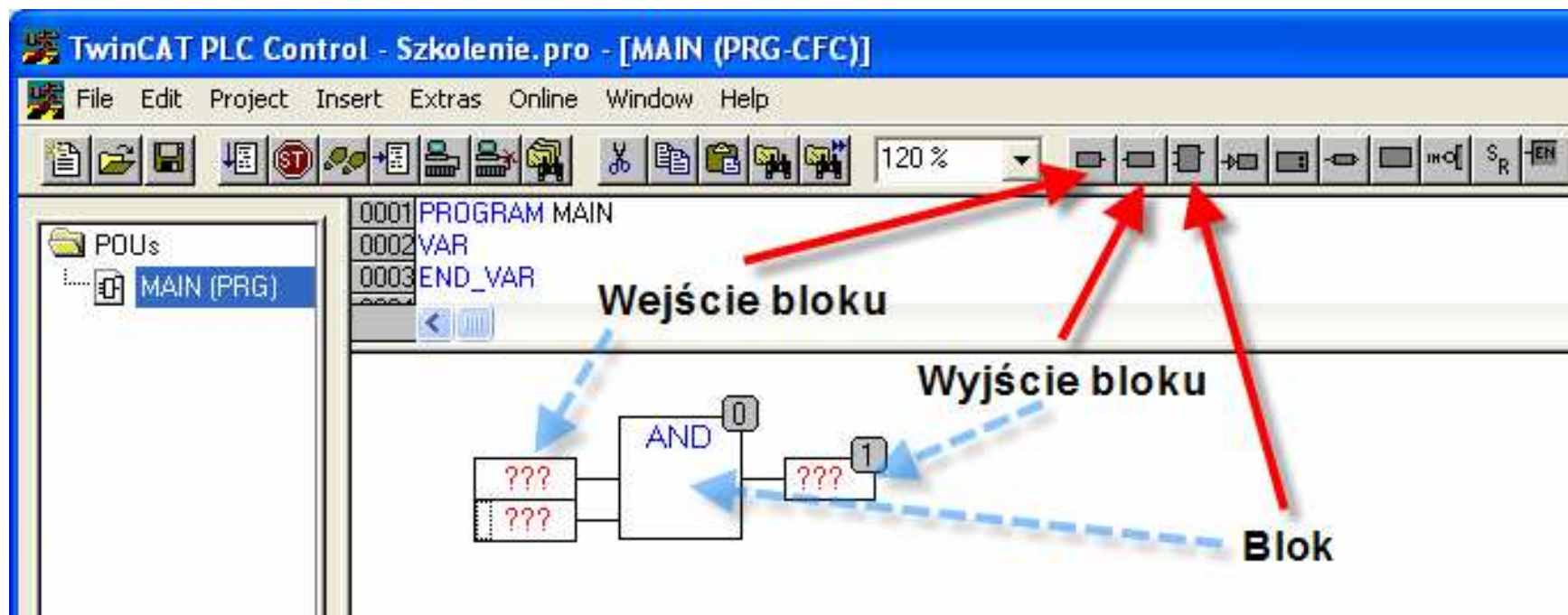
Krok 5. Deklaracja zmiennych:

Nazwa AT %I\*:Typ\_Zmiennej; - zmienna wejściowa

Nazwa AT %Q\*:Typ\_Zmiennej; - zmienna wyjściowa

# Dodanie bloczka

Krok 6.



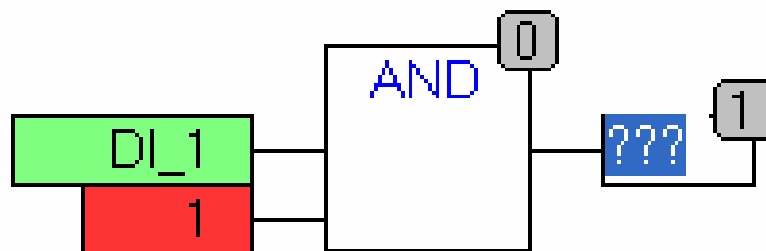
Dodawanie elementów metodą przeciągnij -> upuść  
 Domyślny typ bloczka to bramka AND  
 Widoczne wejścia/wyjścia bloczka  
 Łatwe dodawanie wejść/wyjść





## Nazwanie zmiennych we/wy

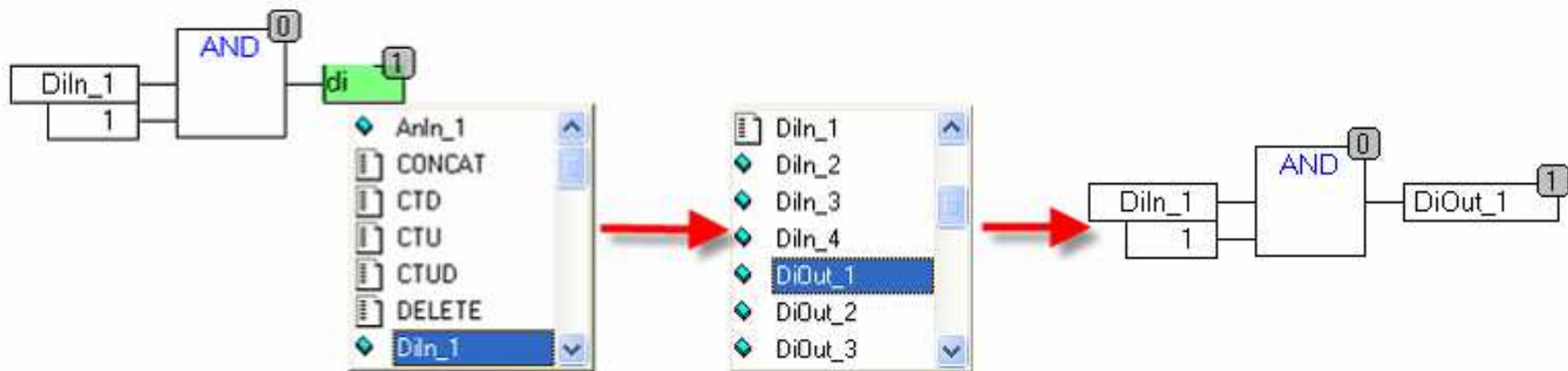
Krok 7.



Na wejściu/wyjściu można wpisać:

- Zmienną globalną
- Zmienną lokalną
- Zmienną wcześniej niezadeklarowaną (domyślna autodeklaracja)
- Wartość stałą (nie da się jej zmienić podczas pracy programu!)
- Wyjście z innego bloku (np. Timer1.Q)
- Wyrażenie (np. WORD\_TO\_BOOL(Dane))

# Ctrl + Spacja



Skrót klawiszowy Ctrl + Spacja otwiera okno z zadeklarowanymi Zmiennymi, blokami funkcyjnymi i funkcjami



# Wybór obiektu docelowego

Krok 8.

Online Window Help	
Login	F11
Logout	F12
<b>Choose Run-Time System...</b>	
Create Bootproject	
Create Bootproject (offline)	
Delete Bootproject	

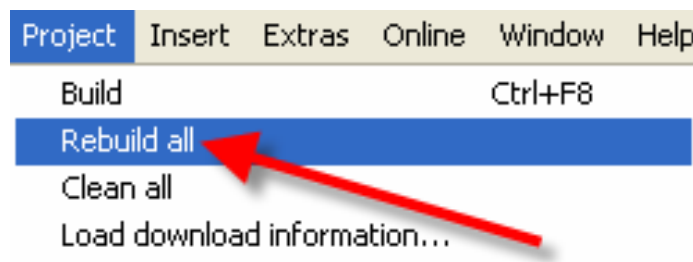
---Local---	(192.168.255.151.1.1)
<b>IEC 1101</b>	<b>Run-Time 1 (Port 801)</b>
+ <Default>	(255.255.255.255.255.255)
+ BX8000	(1.1.1.1.1.1)
+ CX1000	(192.168.10.24.1.1)

Local – program zostanie wgrany na lokalny komputer  
 Pozostałe obiekty to nazwy sterowników wraz z AMS Net Id



# Kompilacja projektu

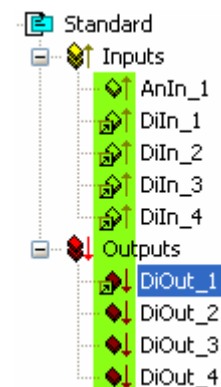
Krok 9.



Warning 1990 -> brak obiektu docelowego dla zmiennej zaadresowanej

```
Warning 1990: No 'VAR_CONFIG' for 'DiIn_2'
Warning 1990: No 'VAR_CONFIG' for 'DiIn_3'
Warning 1990: No 'VAR_CONFIG' for 'DiIn_4'
Warning 1990: No 'VAR_CONFIG' for 'DiOut_1'
Warning 1990: No 'VAR_CONFIG' for 'DiOut_2'
Warning 1990: No 'VAR_CONFIG' for 'DiOut_3'
Warning 1990: No 'VAR_CONFIG' for 'DiOut_4'
POU indices: 51 (2%)
Size of used data: 49 of 1048576 bytes (0.00%)
Size of used retain data: 0 of 32768 bytes (0.00%)
0 Error(s), 9 Warning(s).

↓
0 Error(s), 0 Warning(s).
```





Połączenie programu z urządzeniami

## TwinCAT System Manager



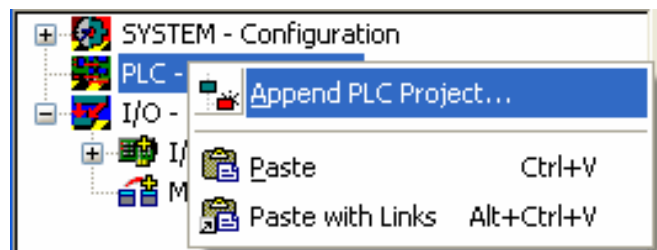




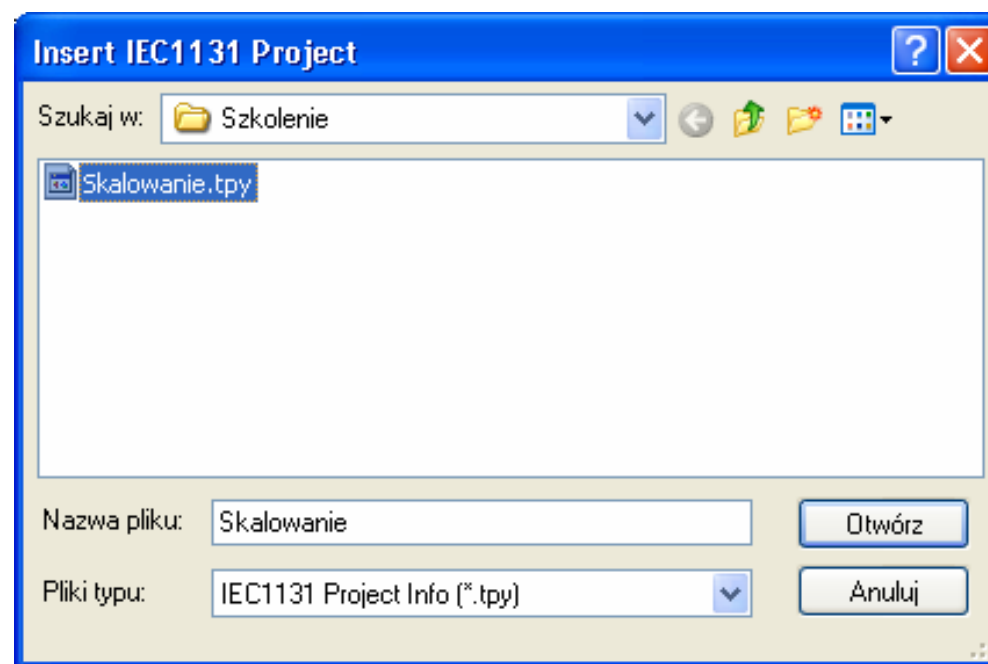
## Konfiguracja urządzenia – krok 3. Dodanie projektu PLC

### Append PLC Project:

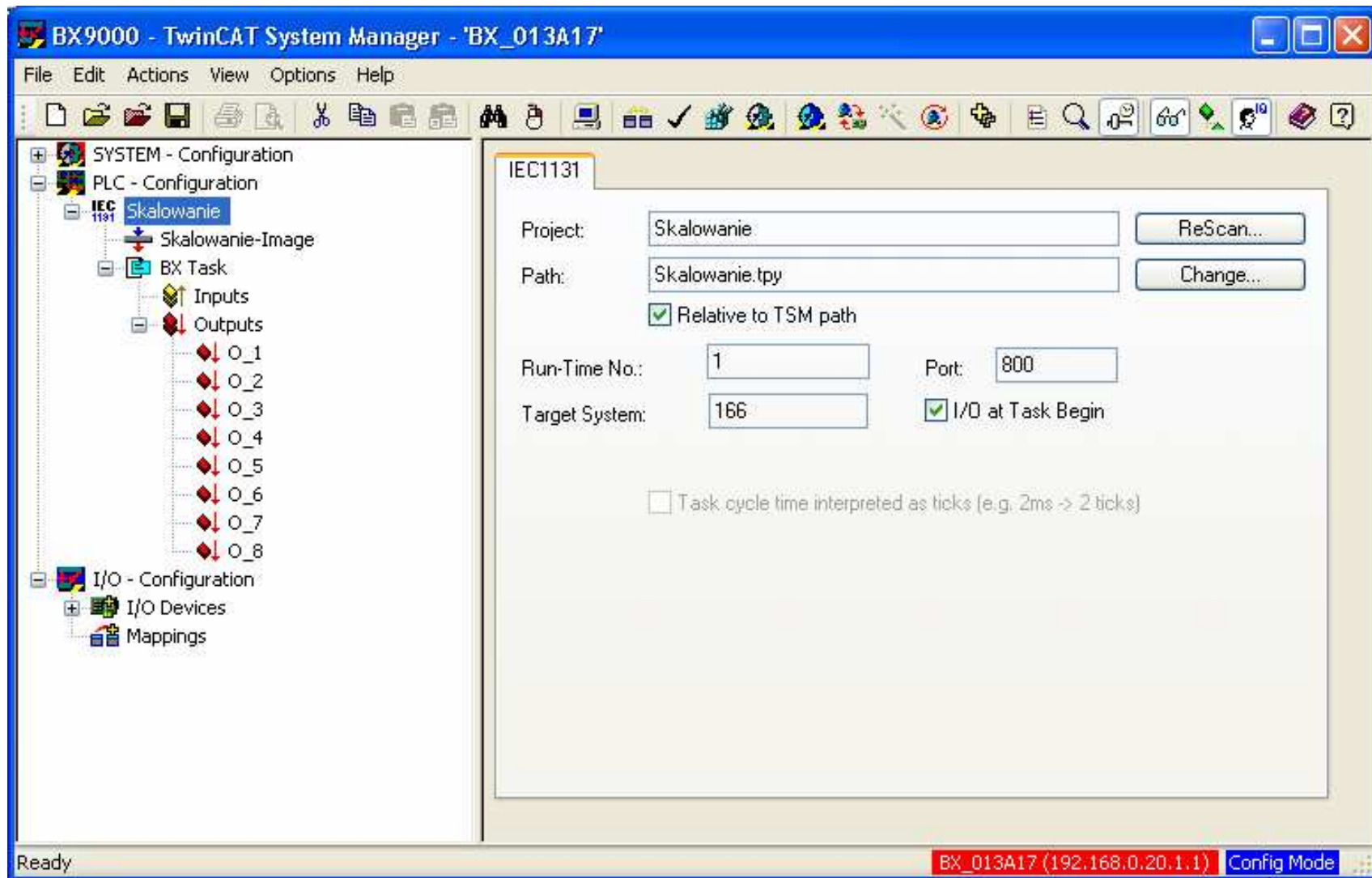
- wywołanie kliknięciem prawym przyciskiem myszy na PLC – Configuration



- otwieramy plik z rozszerzeniem \*.tpy



# Konfiguracja urządzenia sterownik + program PLC



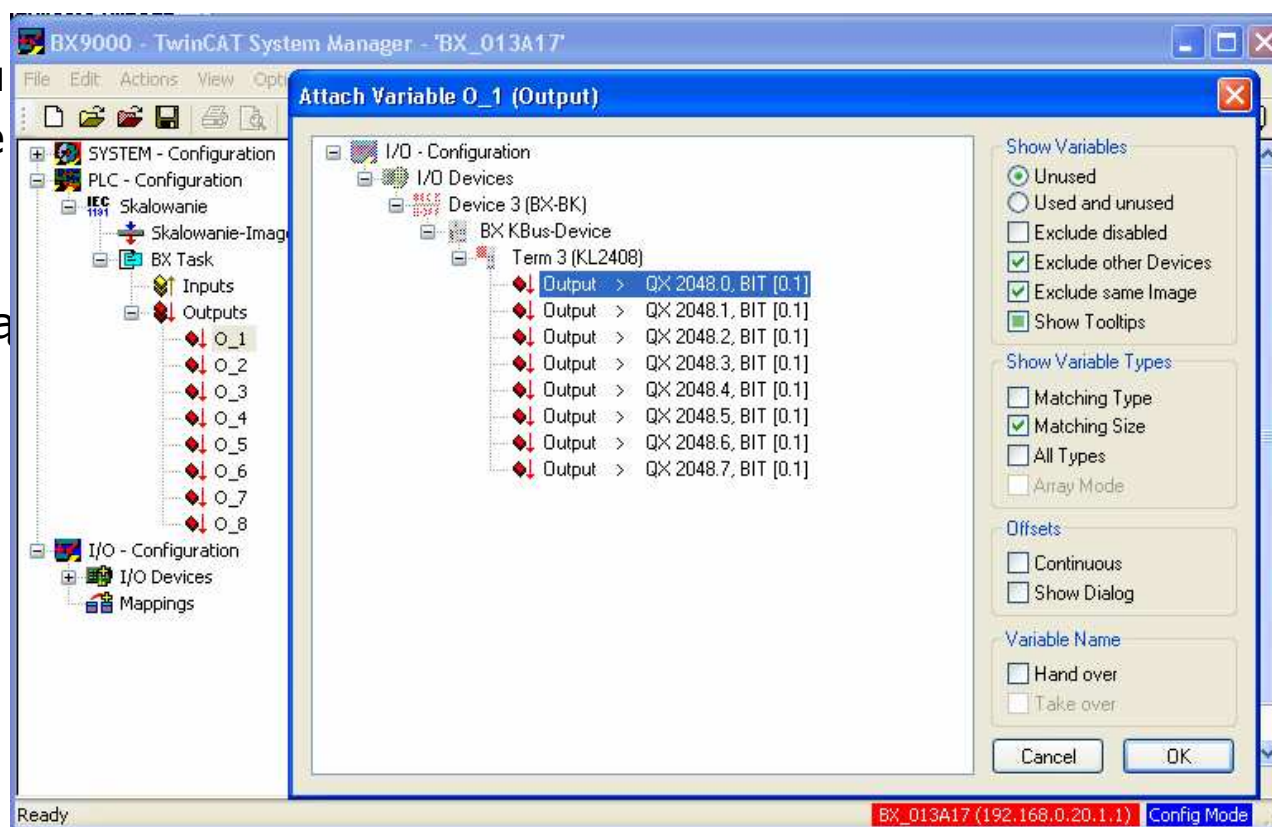


## Konfiguracja urządzenia – krok 4. linkowanie zmiennych

Linkowanie zmiennych:

- wywołanie 2-krotnym kliknięciem na zmienną we/wy w programie PLC lub kliknięcie prawym przyciskiem i wybranie opcji Change Link...
- linkowane mogą być zmienne tego samego typu
- linkowanie łączy zmienne programu PLC ze zmiennymi sprzętowymi
- zmienne zlinkowane mają symbol strzałki i są oznaczone „x”



Name	Type
↓ O_1	X BOOL
↓ O_2	X BOOL
↓ O_3	X BOOL
↓ O_4	X BOOL
↓ O_5	X BOOL
↓ O_6	BOOL
↓ O_7	Variable
↓ O_8	BOOL





## Konfiguracja urządzenia – krok 5. aktywacja konfiguracji

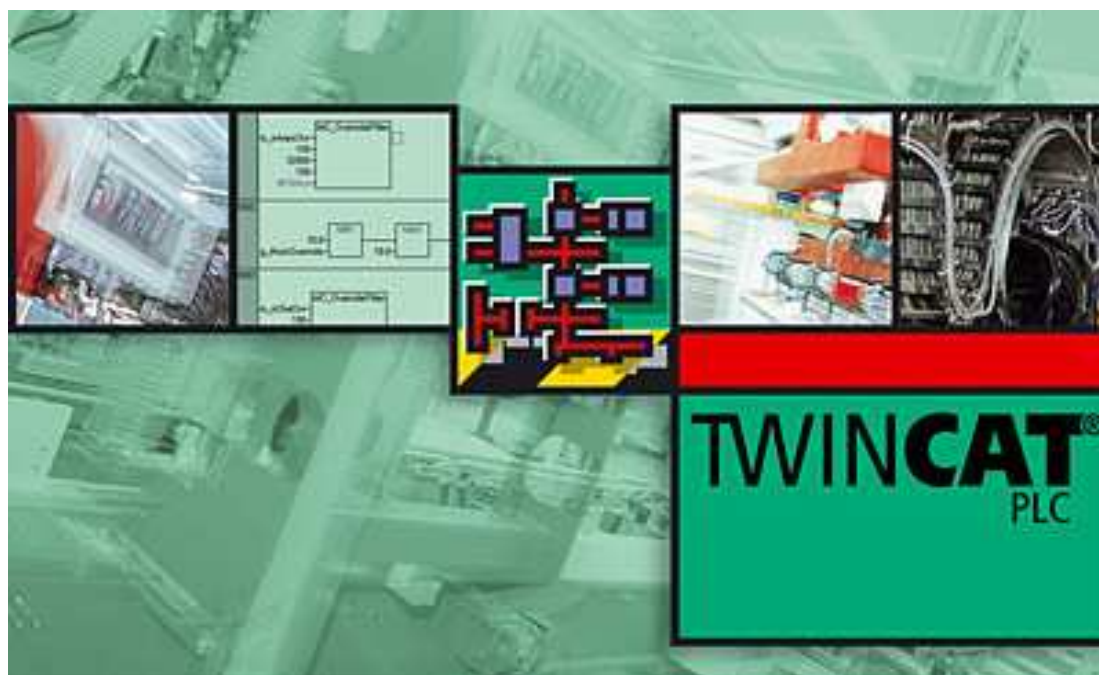
Zakończenie konfiguracji sterownika:

- aktywowanie konfiguracji (Active configuration) 
- przejście w tryb pracy (Run Mode) 



Wgranie programu na sterownik

# TwinCAT PLC







# Logowanie/wylogowanie się

Krok 10.

Zalogowanie się:



lub

Online	Window
Login	F11

Podczas logowania się w razie potrzeby wgrywany jest program PLC.

Wylogowanie się:



lub

Online	Window
Login	F11
Logout	F12

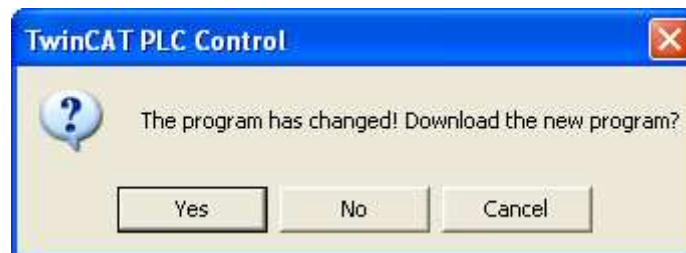


## Wgranie projektu na obiekt docelowy - komunikaty

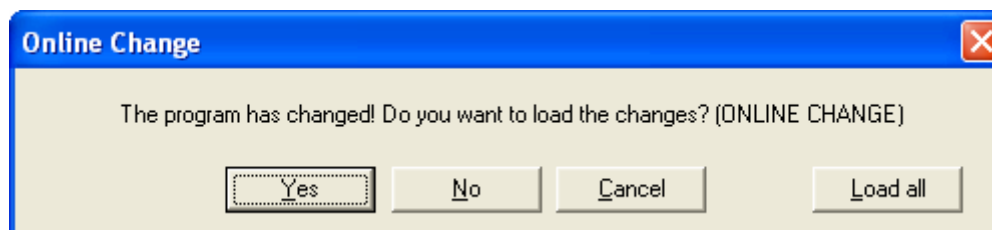
Na sterowniku nie ma programu:



Na sterowniku jest już program, jednak znacznie różni się od obecnego:



Na sterowniku jest już program, nieznacznie różniący się od obecnego, możliwa jest zmiana w trybie on-line (program PLC się nie zatrzymuje):





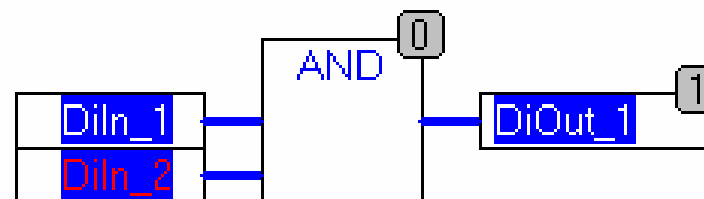
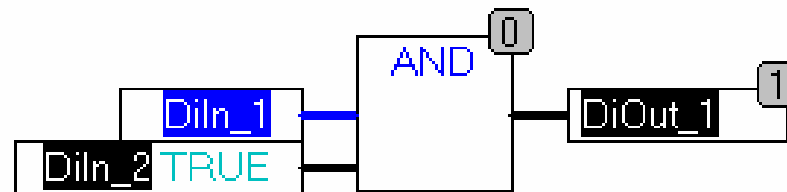
# Praca w trybie on-line

Uruchomienie programu (f5) lub



Target: CX\_00F436 (5.0.244.54.1.1), Run Time: 1 | ONLINE: | SIM | RUN

Wpisywanie zmiennych (Ctrl + f7)





## Praca w trybie on-line

Podgląd wartości zmiennych adresowanych typu bool):

0001	AnIn_1 = 0
0002	DiIn_1 (%IX0.1) = TRUE
0003	DiIn_2 (%IX0.2) = TRUE
0004	DiIn_3 (%IX0.3) = FALSE
0005	DiIn_4 (%IX0.4) = FALSE
0006	DiOut_1 (%QX0.0) = TRUE
0007	DiOut_2 (%QX0.1) = FALSE
0008	DiOut_3 (%QX0.2) = FALSE
0009	DiOut_4 (%QX0.3) = FALSE

Nieadresowane, typ Int oraz String:

Cz\_zapisu = 1

Cz\_pliku = 0

Godzina = 15

Sciezka = 'c:\'

Naglowek = 'Kolumna 1'



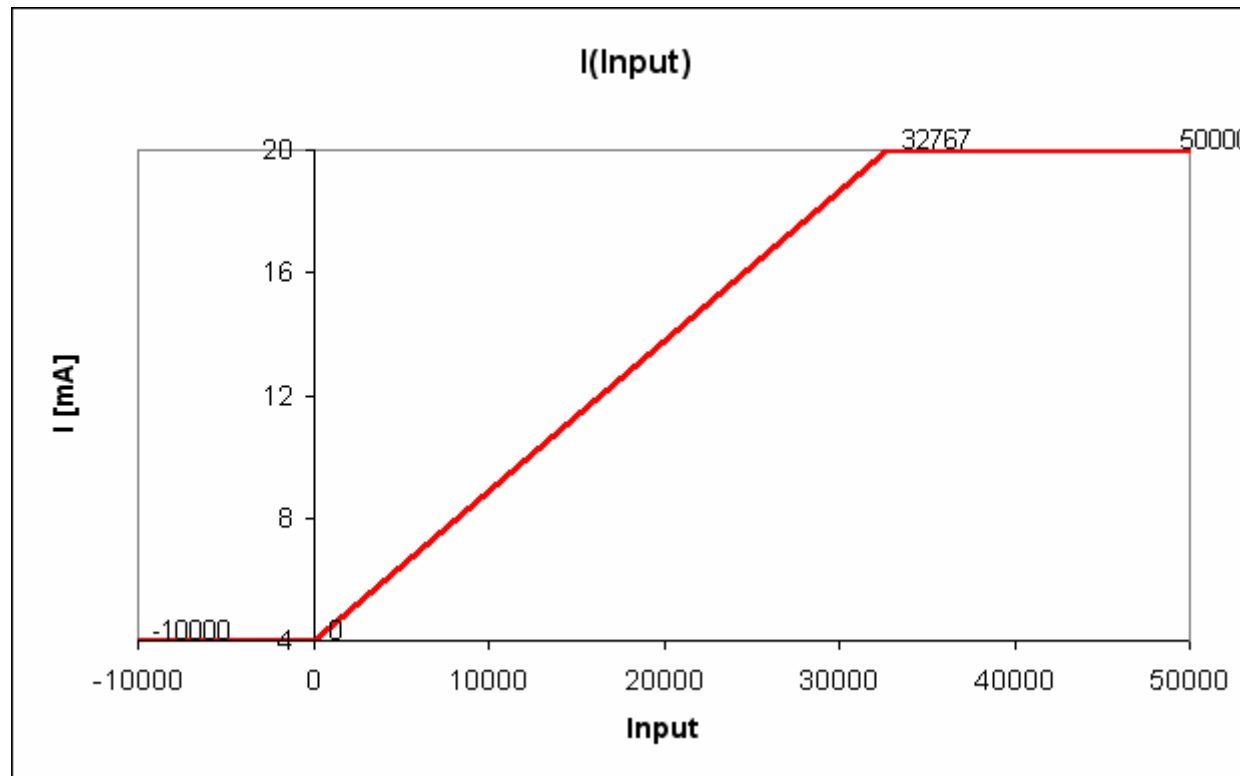
# Tworzenie bloku funkcyjnego







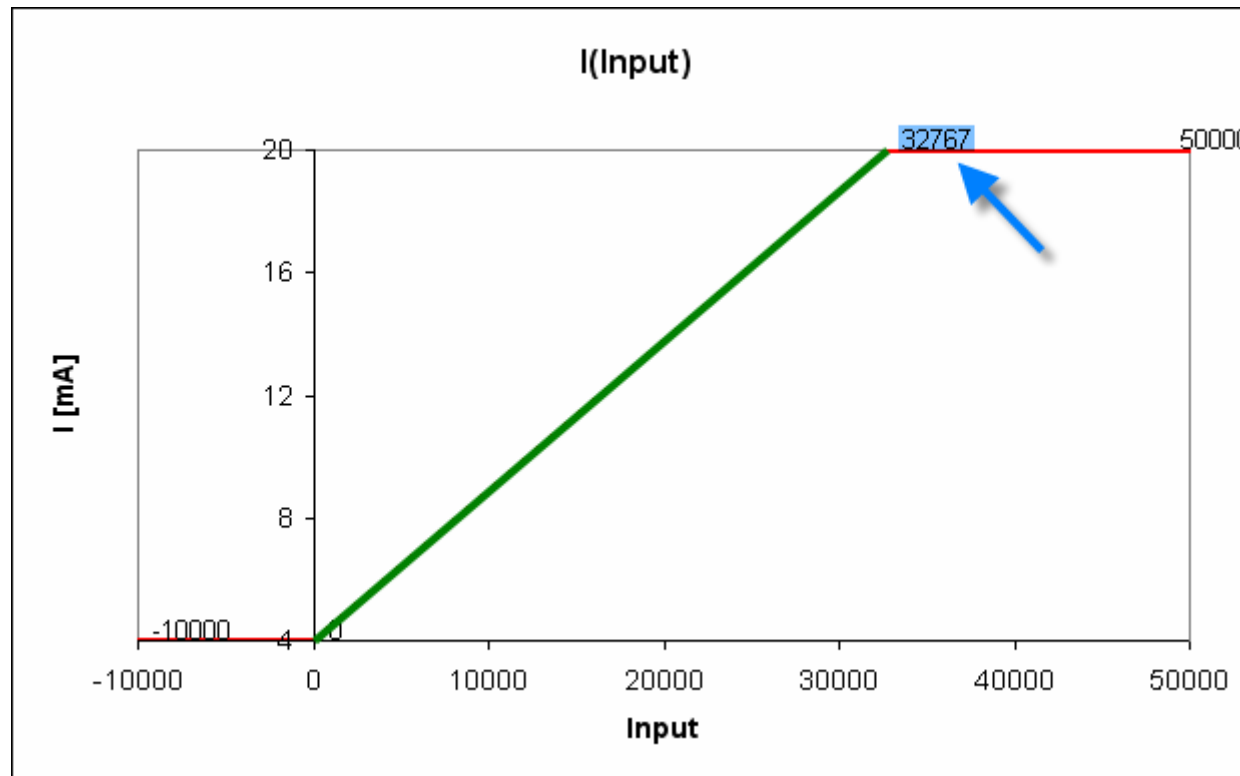
# Problem do rozwiązania



Przeskalowanie wartości z zakresu przetwornika (0-32767)  
na wyjście modułu (4 - 20 mA)



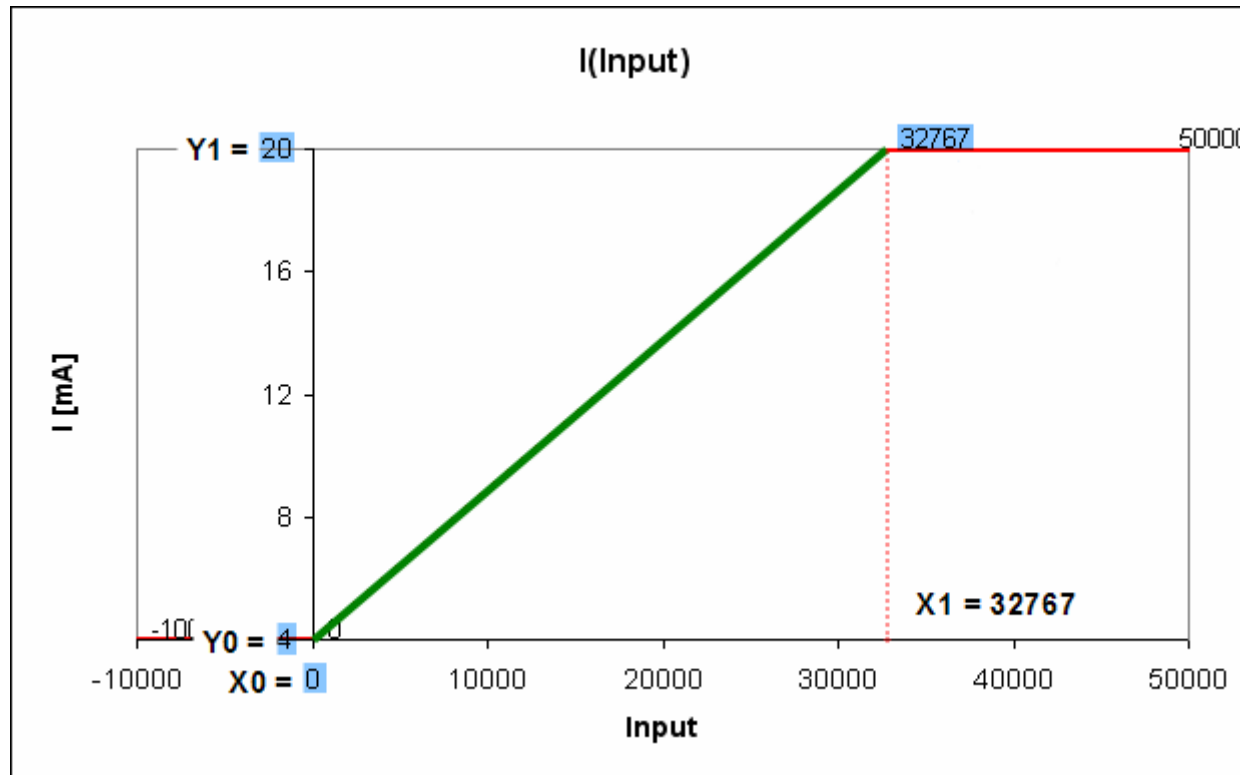
# Zakres pracy przetwornika



Rozdzielczość 15 bitów => 0 - 32767



# Współczynniki



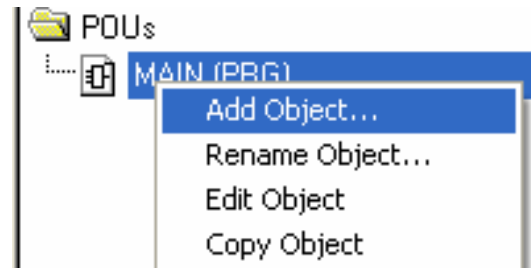
$$fA := (Y1 - Y0) / (X1 - X0);$$

$$fB := Y1 - fA * X1;$$

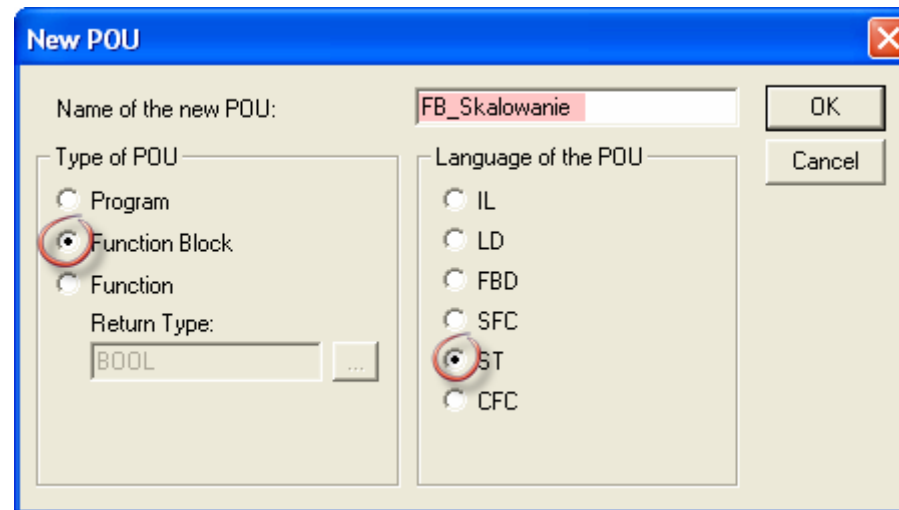


# Dodanie bloku

Krok 1.



Wybór typu obiektu i języka programowania





# Deklaracja zmiennych

Krok 2.

Var\_Input – zmienne wejściowe bloczka

Var\_Output – zmienne wyjściowe bloczka

Var – zmienne wewnętrzne bloczka

```
0001 FUNCTION_BLOCK FB_Skalowanie
0002 VAR_INPUT
0003     Value : REAL;
0004     X0 : REAL;
0005     X1 : REAL;
0006     Y0 : REAL;
0007     Y1 : REAL;
0008 END_VAR
0009 VAR_OUTPUT
0010     OUT: REAL;
0011 END_VAR
0012 VAR
0013     fA : REAL;
0014     fB : REAL;
0015 END_VAR
```

Deklaracja zmiennych może odbywać się na bieżąco podczas pojawiania się nowych zmiennych w programie PLC

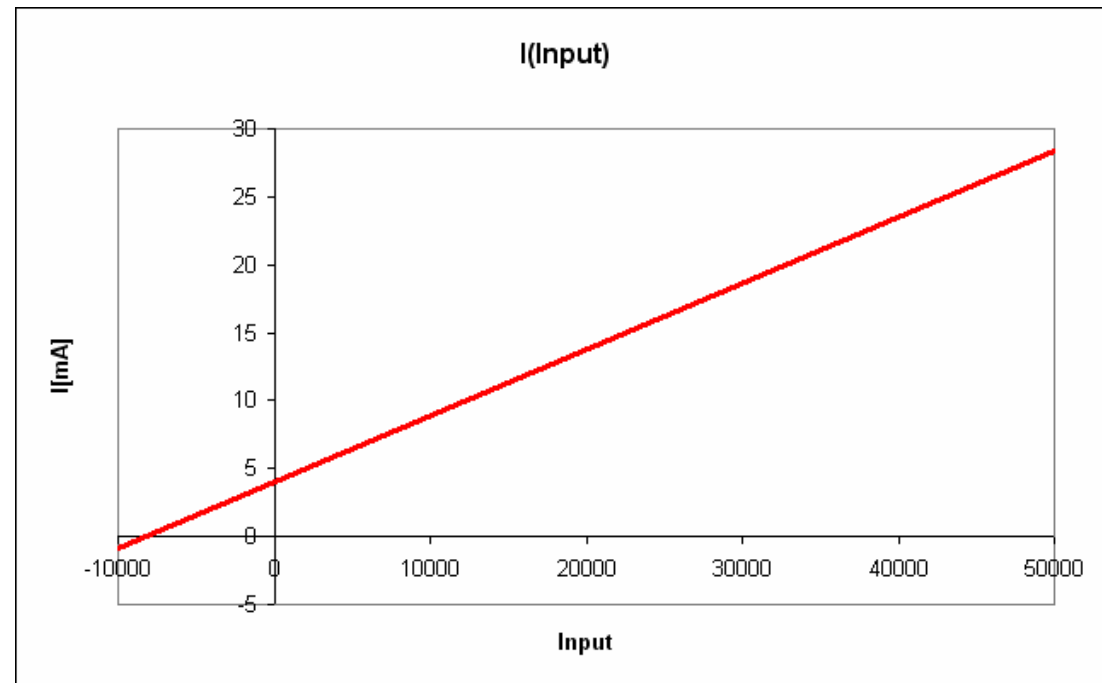


# Pisanie programu (język ST)

### Krok 3.

Kod:

```
fA := (Y1 - Y0) / (X1 - X0);
fB := Y1 - fA * X1;
OUT := fA * Value + fB;
```



## WADA: Brak ograniczeń zakresu!





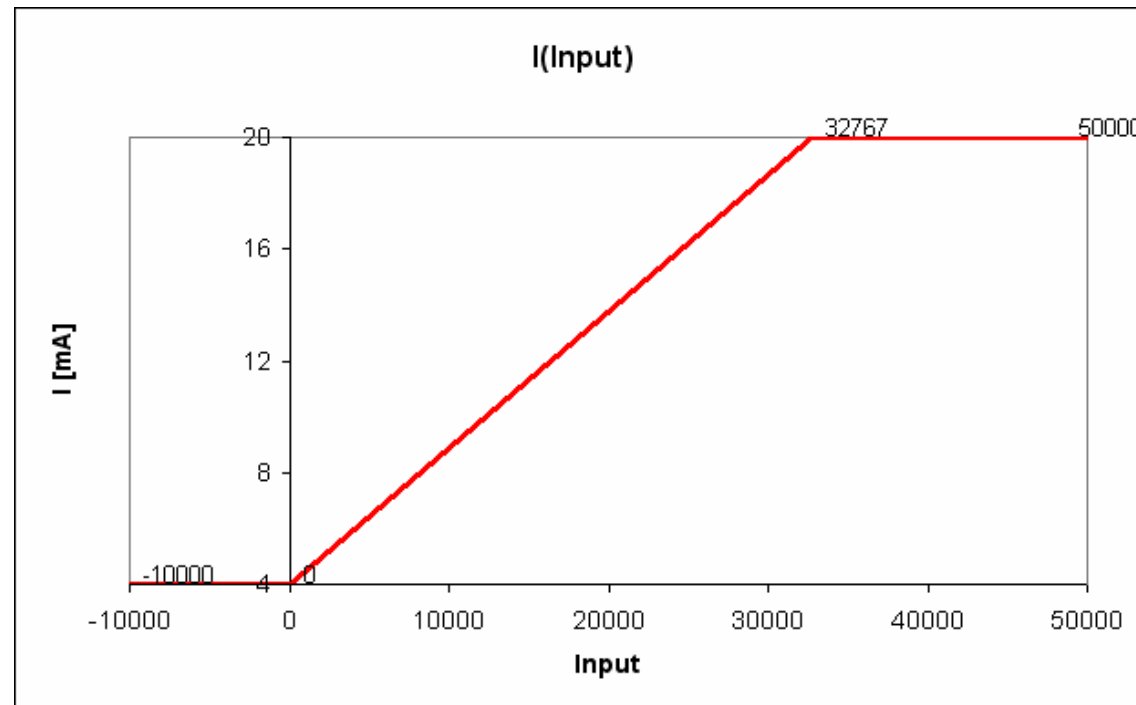
# Pisanie programu (język ST)

Krok 4.

Kod:

```

0001 IF (Value > X1) THEN
0002   OUT := Y1;
0003 ELSIF (Value <= X0) THEN
0004   OUT := Y0;
0005 ELSE
0006   fA := (Y1 - Y0) / (X1 - X0);
0007   fB := Y1 - fA * X1;
0008   OUT := fA * Value + fB;
0009 END_IF
    
```



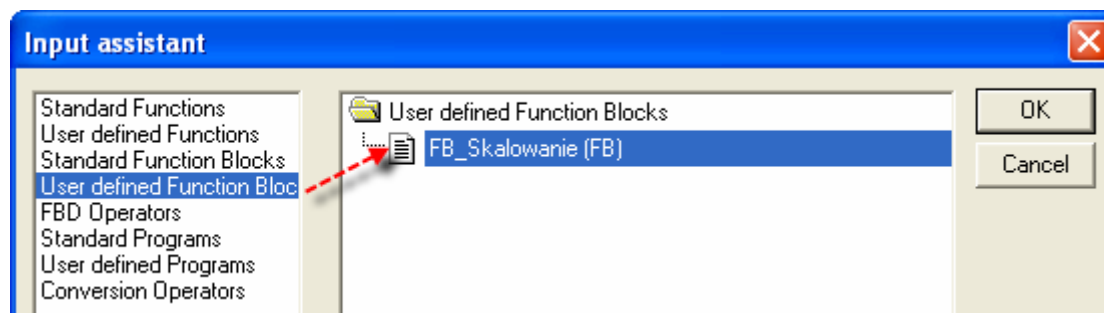
WADA: Brak odporności na  $X1 = X2$ !



# Wywołanie bloczka

Krok 5.

1. Przechodzimy do programu głównego
2. Wstawiamy bloczek (domyślnie AND)
3. Zaznaczamy nazwę bloczka i:
  - A) Wciskamy F2 i wybieramy interesujący nas bloczek



- B) Wpisujemy nazwę bloku funkcyjnego (FB\_Skalowanie)



Wywołany jest dopiero blok funkcyjny, należy przypisać mu jeszcze nazwę

# Przypisanie zmiennych we/wy

Krok 6.

Okno autodeklaracji

**Declare Variable**

Class	Name	Type
VAR	Wejscie	BOOL

Symbol list: Global\_Variables

Initial Value:

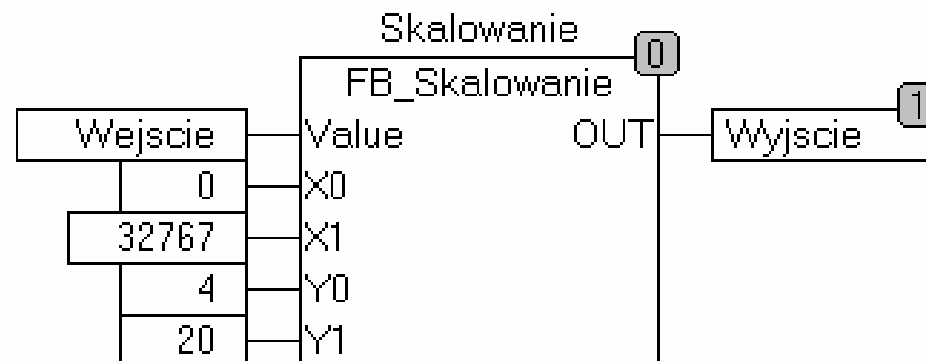
Address:

Comment:

OK Cancel

☐ CONSTANT ☐ RETAIN ☐ PERSISTENT

Gotowy blok skalujący zakres (0 - 32767) na (4 - 20)

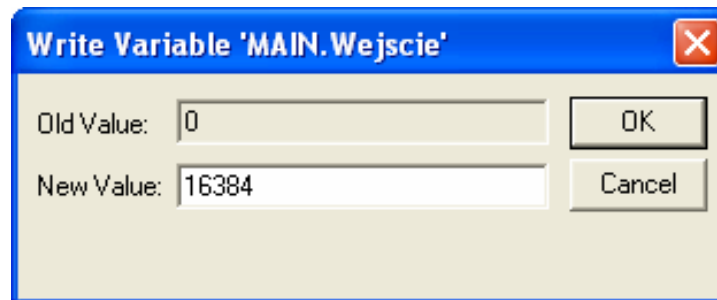




# Uruchomienie programu

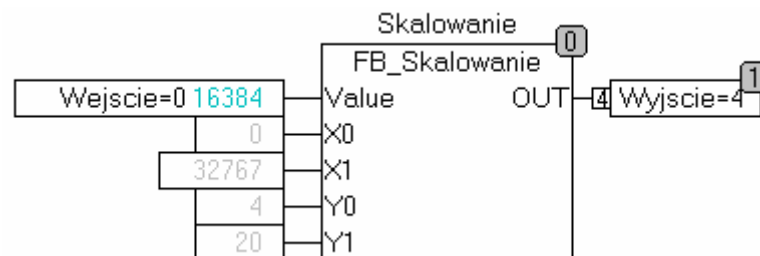
Krok 7.

Wpisanie nowej wartości



Wpisanie wartości – Ctrl + f7

Forsowanie wartości – f7

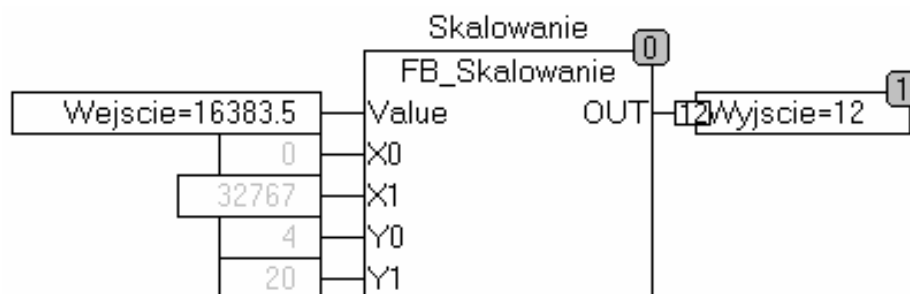




# Praca bloczka w trybie on-line

Krok 8.

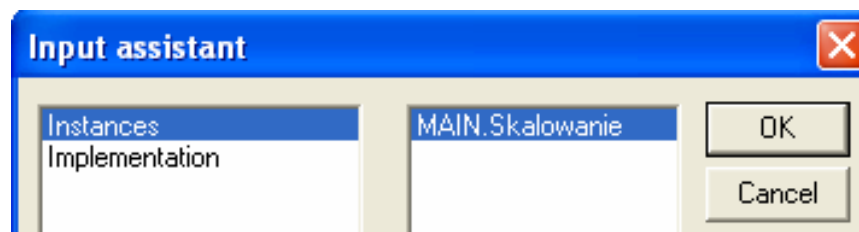
Praca w trybie on-line



Wejście do bloczka:

Instances – podgląd wartości zmiennych w konkretnym bloczku

Implementation – podgląd kodu bloczka





## Podgląd wartości zmiennych w bločku

Krok 9.

Wejście do bločku pozwala na sprawdzenie aktualnych wartości zmiennych lokalnych w bločku.

Wyświetlany jest również kod źródłowy bločku.

0001	fA = 4.882962e-004				
0002	fB = 4				
0003	Value = 16383.5				
0004	X0 = 0				
0005	X1 = 32767				
0006	Y0 = 4				
0007	Y1 = 20				
0008	OUT = 12				
0009					
0001	IF (Value > X1) THEN	Value = 16383.5	X1 = 32767		
0002	OUT := Y1;	OUT = 12	Y1 = 20		
0003	ELSIF (Value <= X0) THEN	Value = 16383.5	X0 = 0		
0004	OUT := Y0;	OUT = 12	Y0 = 4		
0005	ELSE				
0006	fA := (Y1 - Y0) / (X1 - X0);	fA = 4.882962e-004	Y1 = 20	Y0 = 4	X1 = 32767
0007	fB := Y1 - fA * X1;	fB = 4	Y1 = 20	fA = 4.882962e-004	X1 = 32767
0008	OUT := fA * Value + fB;	OUT = 12	fA = 4.882962e-004	Value = 16383.5	fB = 4
0009	END_IF				

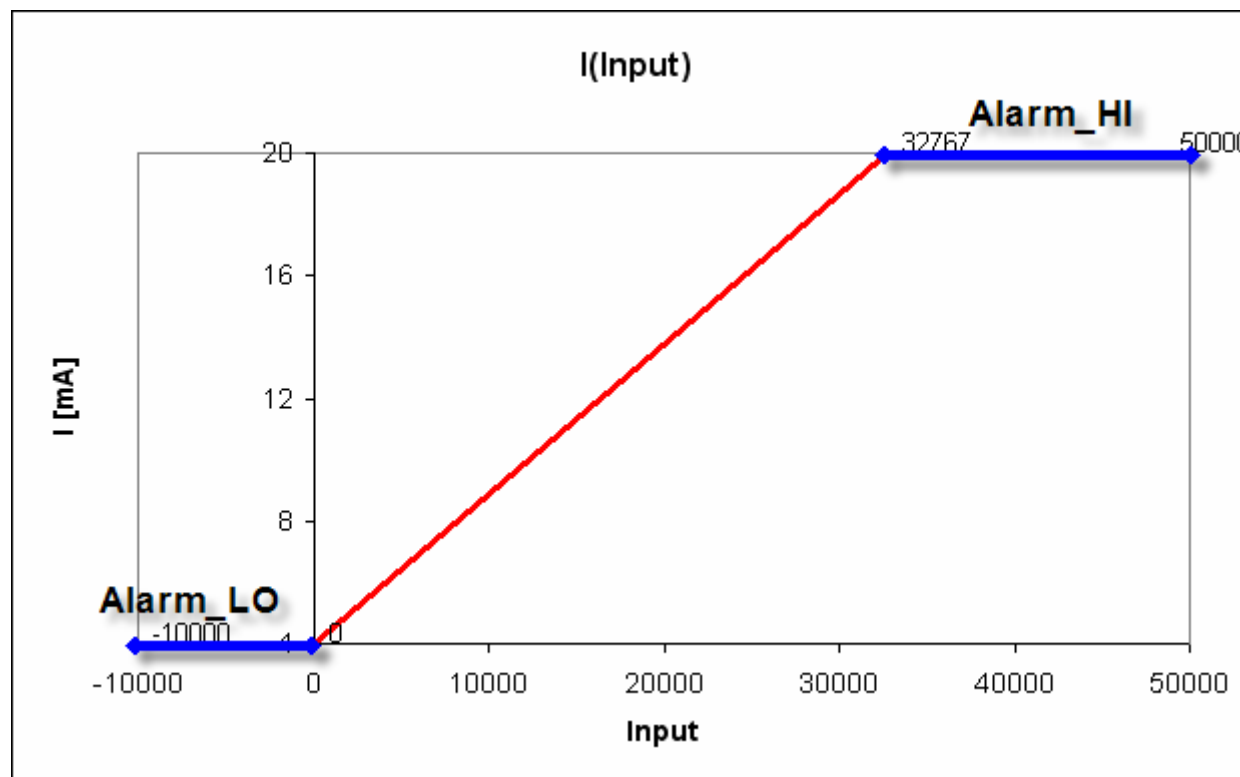




# Alarmy

Krok 10.

Dodanie alarmu o przekroczeniu zakresu górnego lub dolnego.





# Alarms

Kod:

Deklaracja zmiennych:

Program:

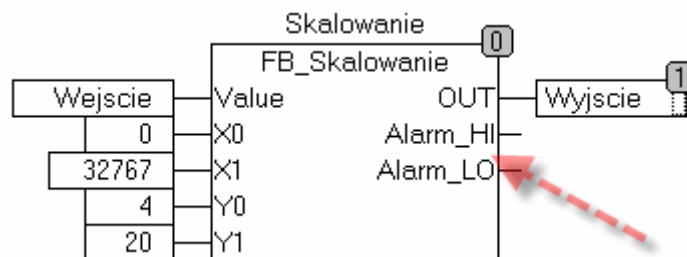
```

0001 FUNCTION_BLOCK FB_Skalowanie
0002 VAR_INPUT
0003     Value : REAL;
0004     X0 : REAL;
0005     X1 : REAL;
0006     Y0 : REAL;
0007     Y1 : REAL;
0008 END_VAR
0009 VAR_OUTPUT
0010     OUT : REAL;
0011     Alarm_HI : BOOL;
0012     Alarm_LO : BOOL;
0013 END_VAR
0014 VAR
0015     fA : REAL;
0016     fB : REAL;
0017 END_VAR
    
```

```

0001 IF (Value > X1) THEN
0002     OUT := Y1;
0003     Alarm_HI := TRUE;
0004     Alarm_LO := FALSE;
0005 ELSIF (Value <= X0) THEN
0006     OUT := Y0;
0007     Alarm_LO := TRUE;
0008     Alarm_HI := FALSE;
0009 ELSE
0010     fA := (Y1 - Y0) / (X1 - X0);
0011     fB := Y1 - fA * X1;
0012     OUT := fA * Value + fB;
0013     Alarm_LO := FALSE;
0014     Alarm_HI := FALSE;
0015 END_IF
    
```

Zmiany w bločku w programie głównym:





# Alarms

Kod:

Deklaracja zmiennych:

```

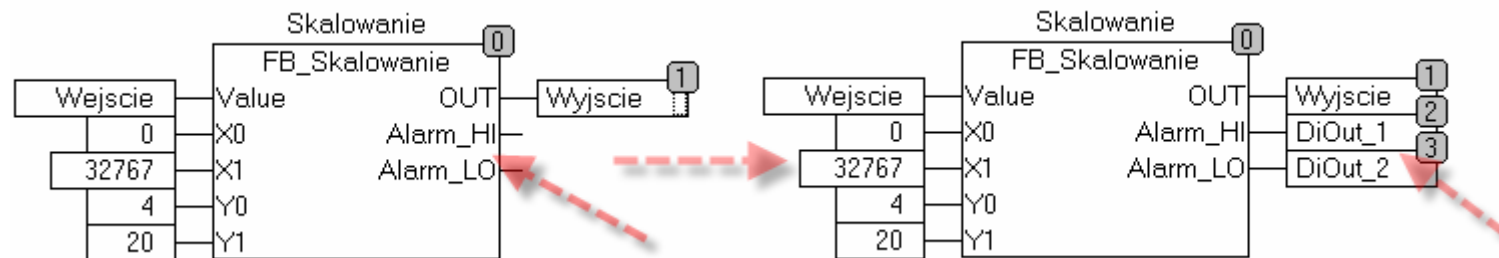
0001 FUNCTION_BLOCK FB_Skalowanie
0002 VAR_INPUT
0003     Value : REAL;
0004     X0 : REAL;
0005     X1 : REAL;
0006     Y0 : REAL;
0007     Y1 : REAL;
0008 END_VAR
0009 VAR_OUTPUT
0010     OUT : REAL;
0011     Alarm_HI : BOOL;
0012     Alarm_LO : BOOL;
0013 END_VAR
0014 VAR
0015     fA : REAL;
0016     fB : REAL;
0017 END_VAR
    
```

Program:

```

0001 IF (Value > X1) THEN
0002     OUT := Y1;
0003     Alarm_HI := TRUE;
0004     Alarm_LO := FALSE;
0005 ELSIF (Value <= X0) THEN
0006     OUT := Y0;
0007     Alarm_LO := TRUE;
0008     Alarm_HI := FALSE;
0009 ELSE
0010     fA := (Y1 - Y0) / (X1 - X0);
0011     fB := Y1 - fA * X1;
0012     OUT := fA * Value + fB;
0013     Alarm_LO := FALSE;
0014     Alarm_HI := FALSE;
0015 END_IF
    
```

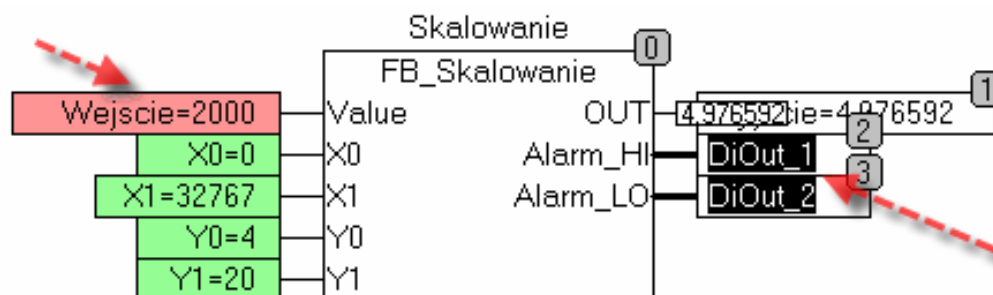
Zmiany w bloku w programie głównym (sygnalizacja diodowa):



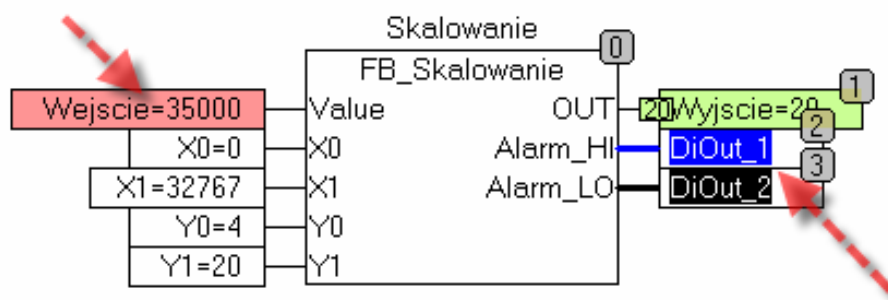


# Praca w trybie on-line

Wartość z zakresu (alarmy wyłączone):



Wartość spoza zakresu (uruchomiony alarm):



Podane jest też napięcie na odpowiednie wyjście cyfrowe (sygnalizowane jest to zapaleniem się diody)

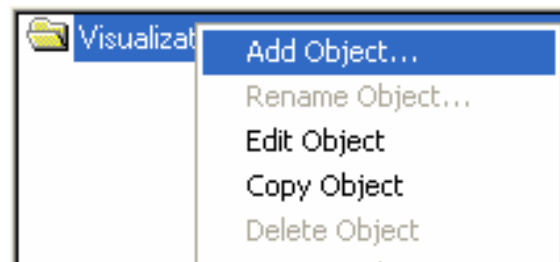


# Wizualizacja

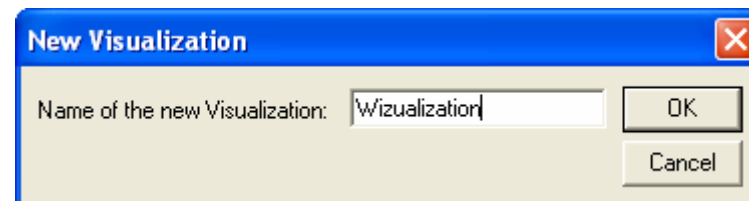
Obiekty wizualizacji znajdują się na zakładce Visualizations



Dodanie nowego obiektu



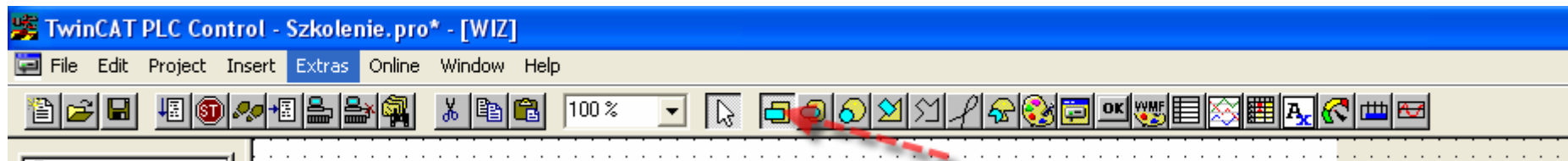
Nadanie nazwy



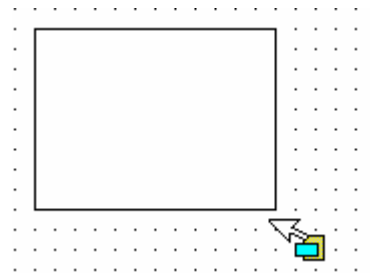


# Element wyświetlający wartość

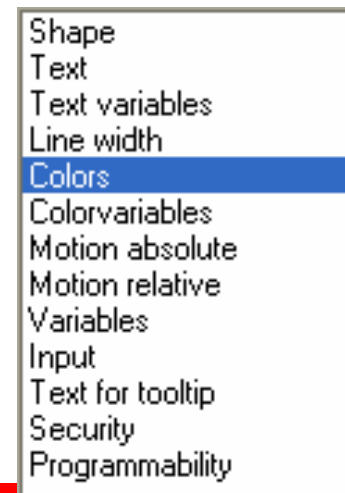
## Wybór elementu



## Wstawienie elementu



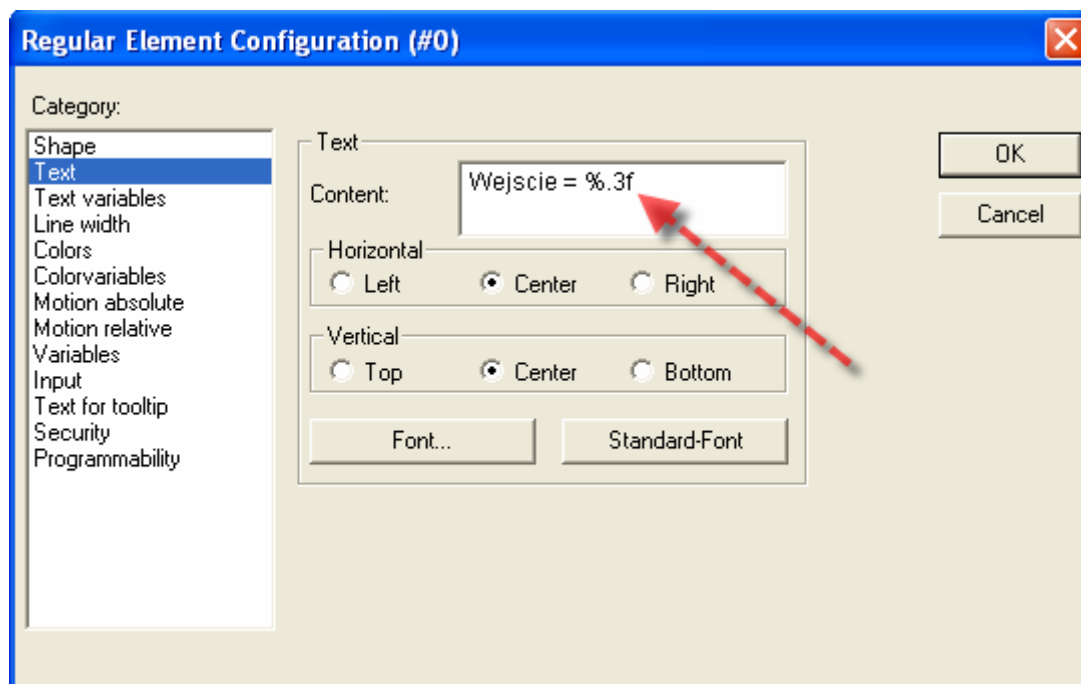
## Ustawienie parametrów elementu





# Wstawienie tekstu

Tekst wyświetlany na elemencie (przycisku, polu, itp.)

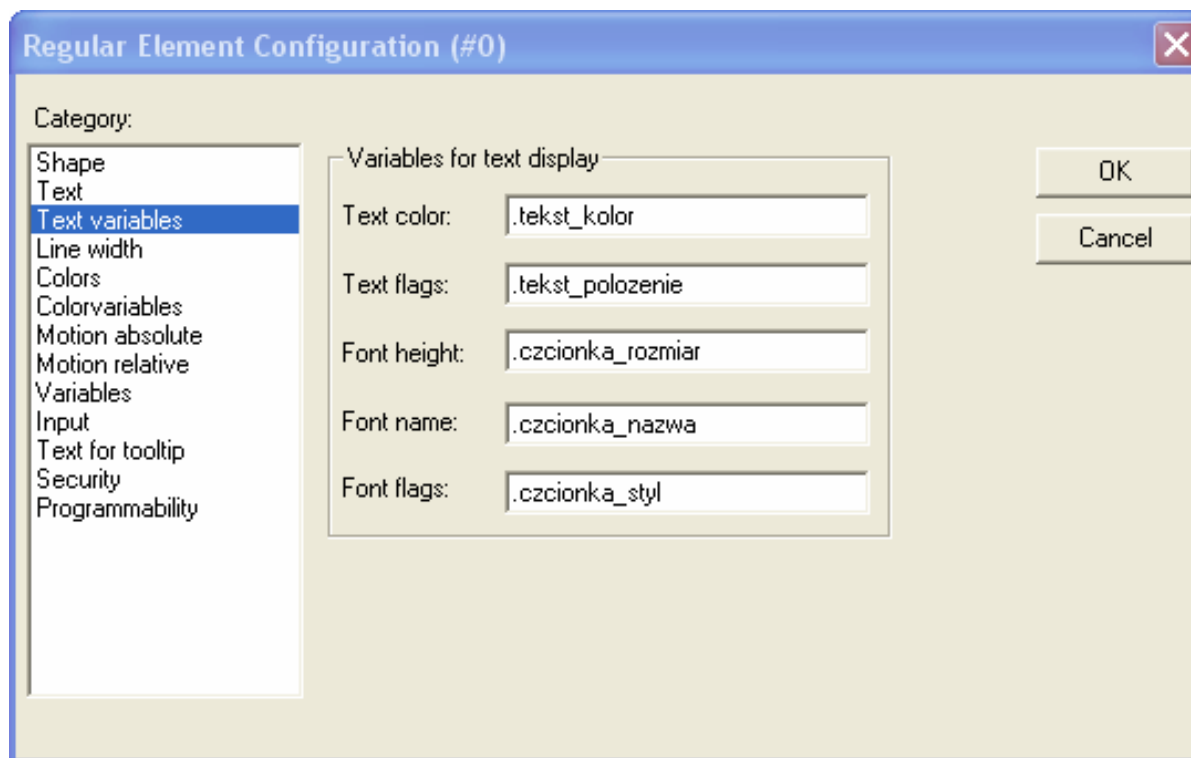


%.3f – wyświetlenie zmiennej typu Real z trzema miejscami po przecinku

%s – zmienna wyświetlana jako napis



# Edycja tekstu



## Formatowanie tekstu

*Czcionka: broadway*

*Wysokosc: 20*

*Polozenie tekstu: 4*

*Styl czcionki: 1*

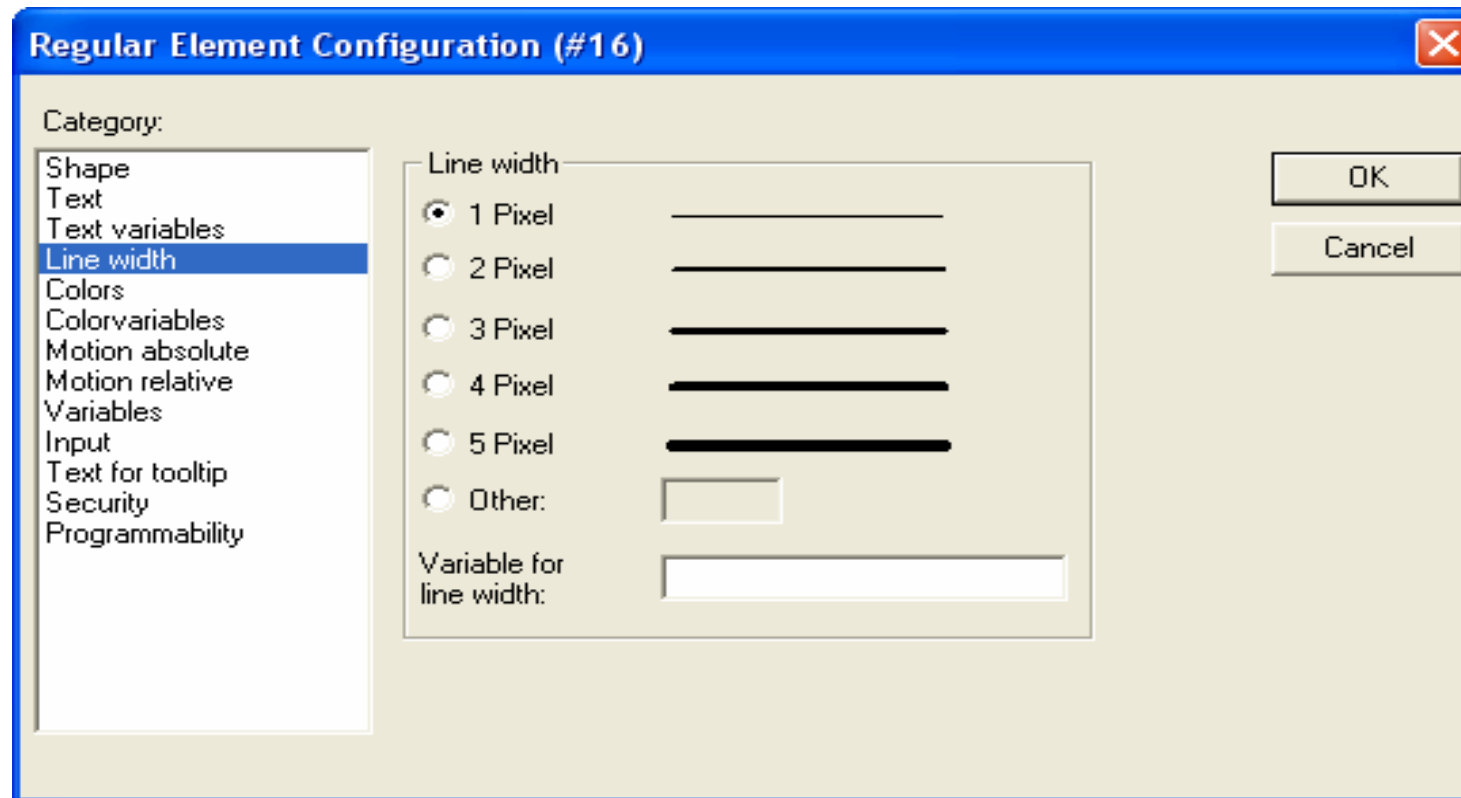
*Kolor: 499*

*Grubosc linii: 2*

Każda cecha wyświetlanego tekstu może być kontrolowana przez zmienną w programie

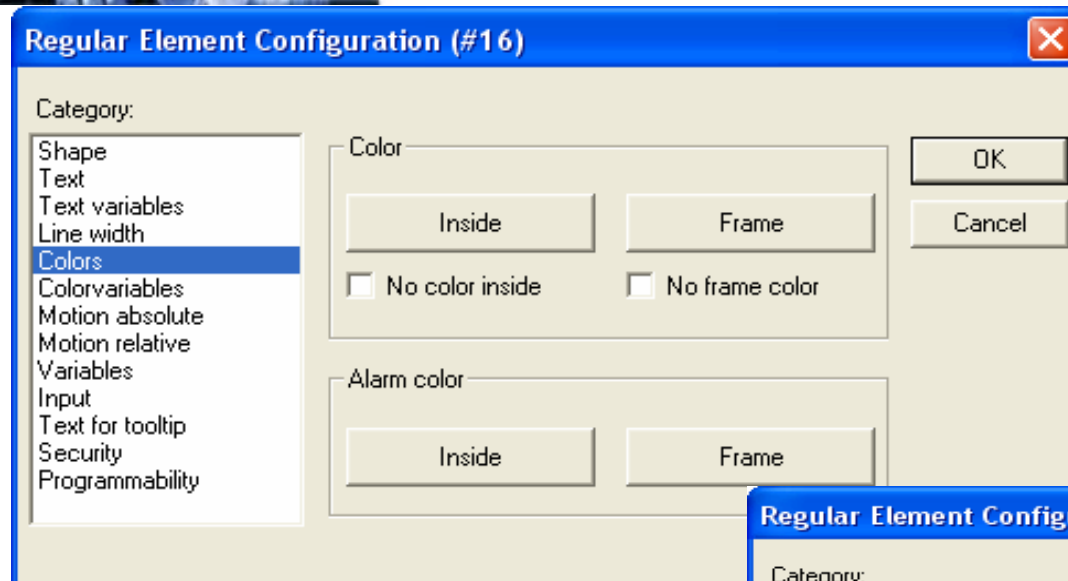


# Edycja linii



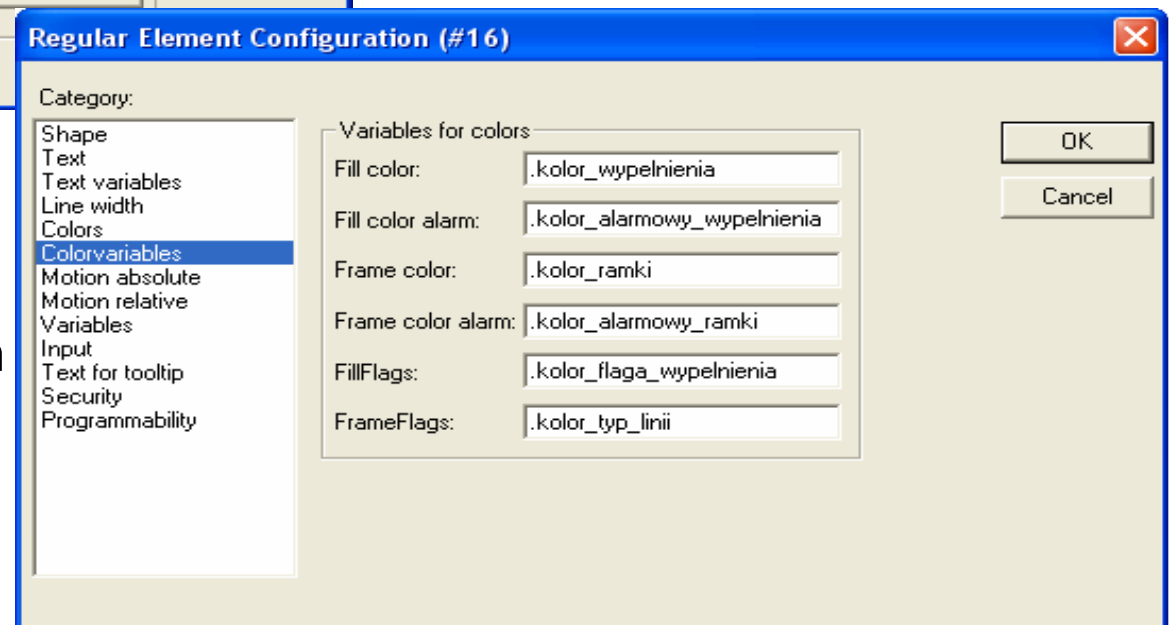
Każda cecha wyświetlanego tekstu może być kontrolowana przez zmienną w programie

# Kolory

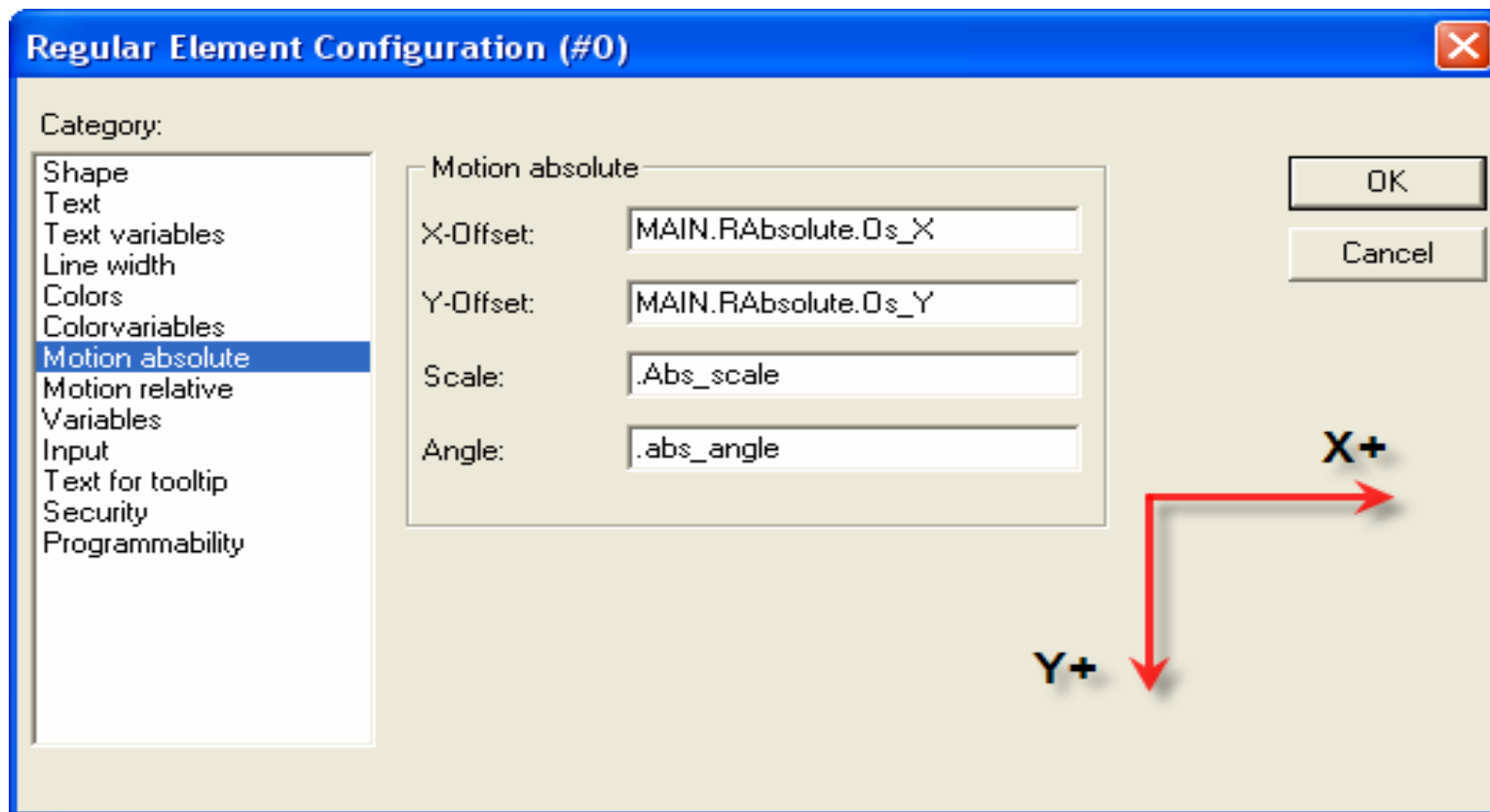


Kolory „statyczne” –  
nadawane tylko raz

Kolory dynamiczne –  
zależne od wartości zmiennych



# Ruch elementu

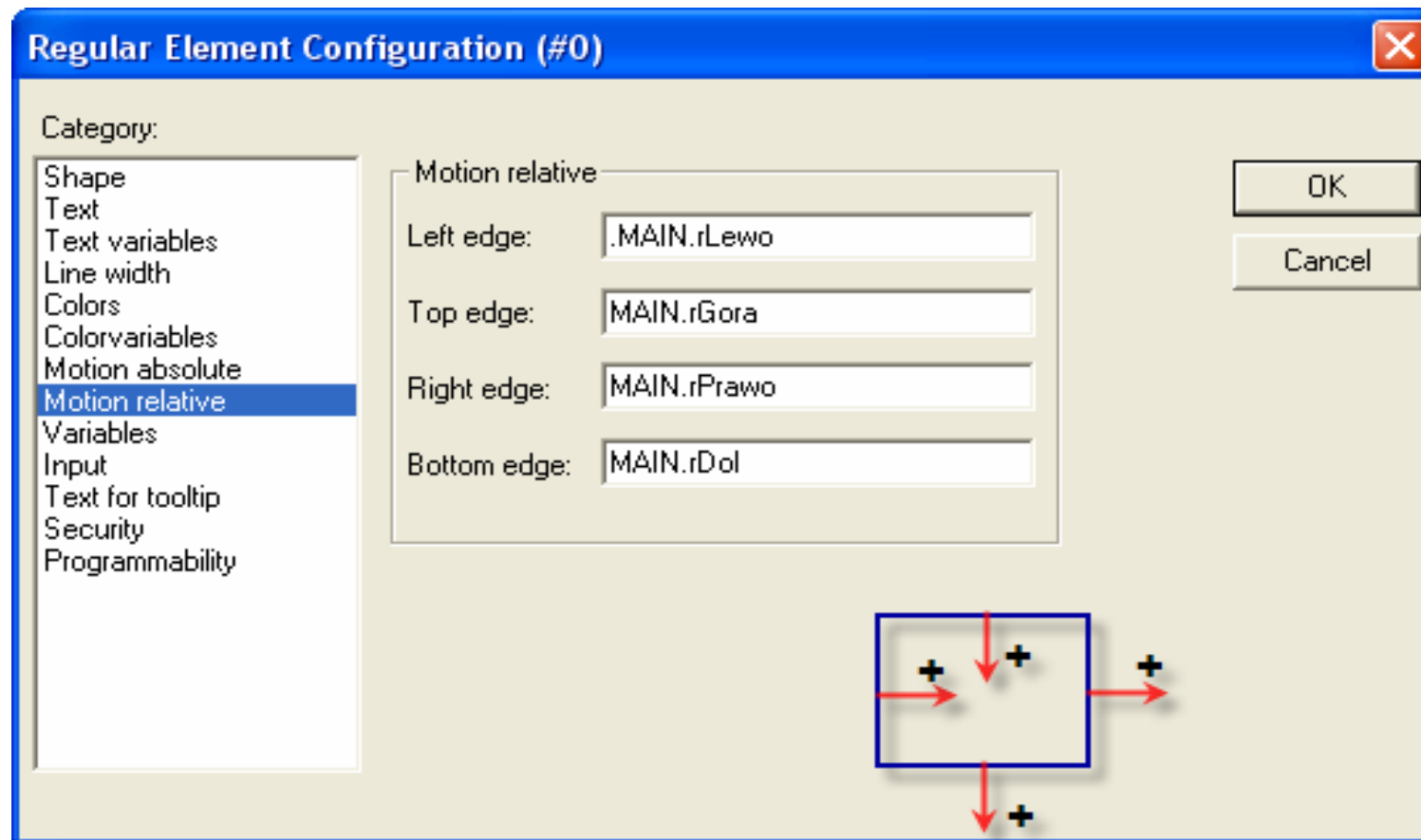


Wartość 0 to pozycja elementu w trybie offline.

Kontrola ruchu elementu może odbywać się poprzez zmienne typu Int

Dokładność 1 piksela.

# Ruch krawędzi elementu



Dynamiczna zmiana położenia krawędzi pozwala na kontrolę wymiarów elementów w wizualizacji za pomocą zmiennych.

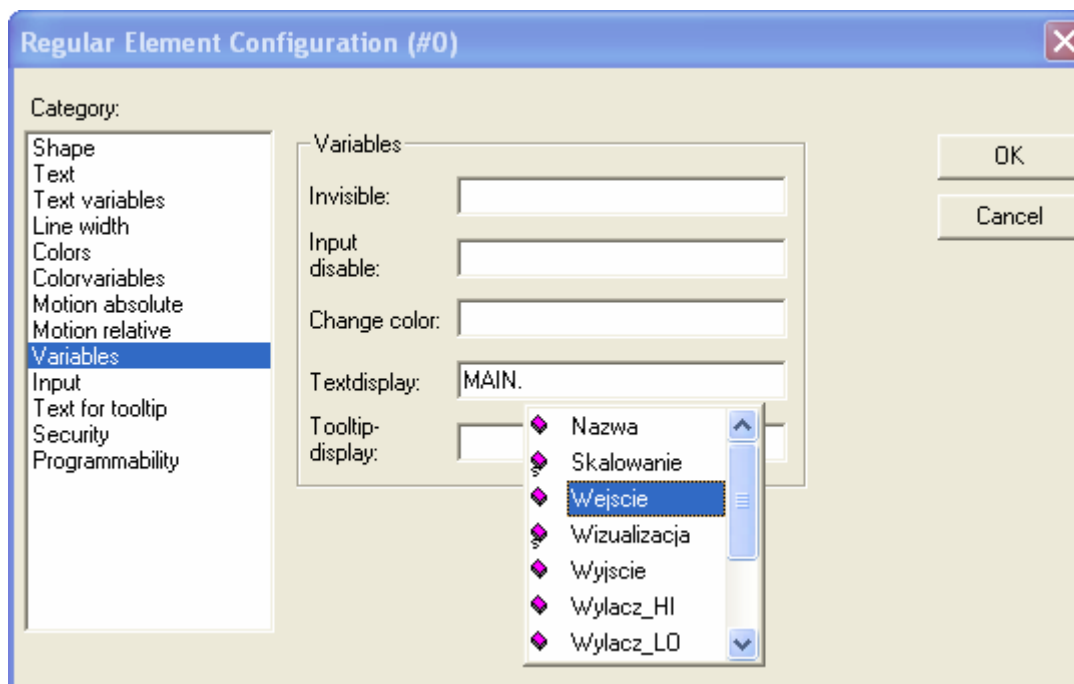




# Własności elementu

## Ustawienie:

- znikania,
- blokowania,
- zmiany koloru,
- wyświetlanego tekstu
- podpowiedzi w trybie on-line



Main.Wejscie  
.DiOut\_1

- odwołanie do zmiennej zadeklarowanej w programie głównym
- odwołanie do zmiennej globalnej

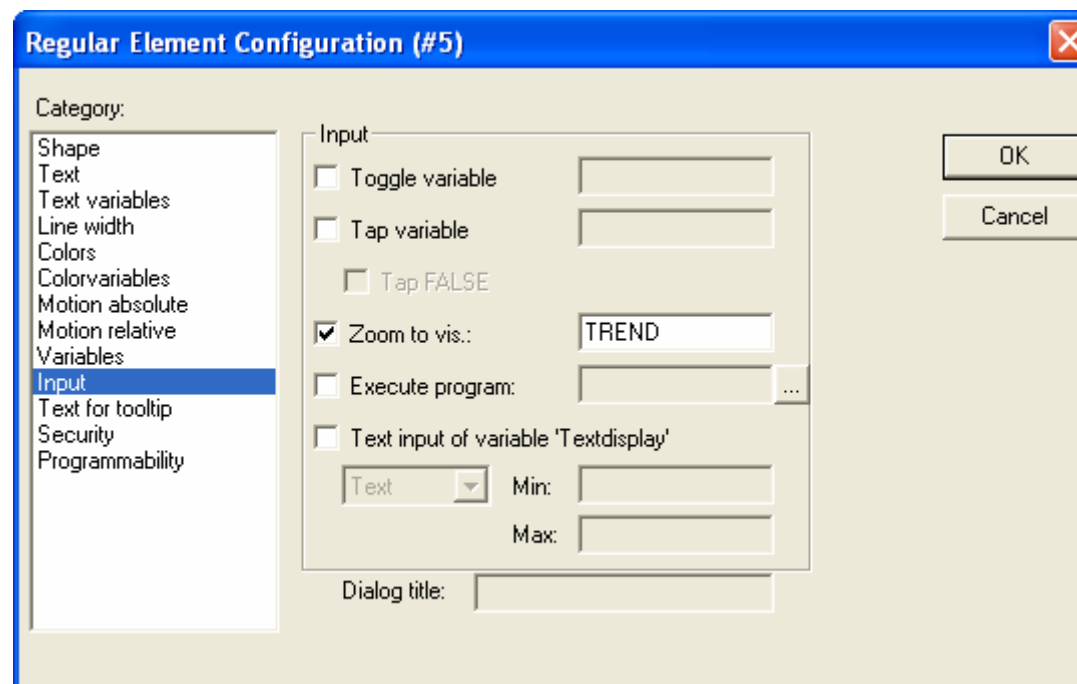


# Akcje

Akcja wykonuje się podczas naciśnięcia obiektu.

Możliwe akcje:

- Przełączenie zmiennej
- Chwilowe włączenie zmiennej
- Zmiana wizualizacji
- Wykonanie programu
- Zmiana wartości



Main.Wejscie – odwołanie do zmiennej zadeklarowanej  
w programie głównym

.DiOut\_1 – odwołanie do zmiennej globalnej

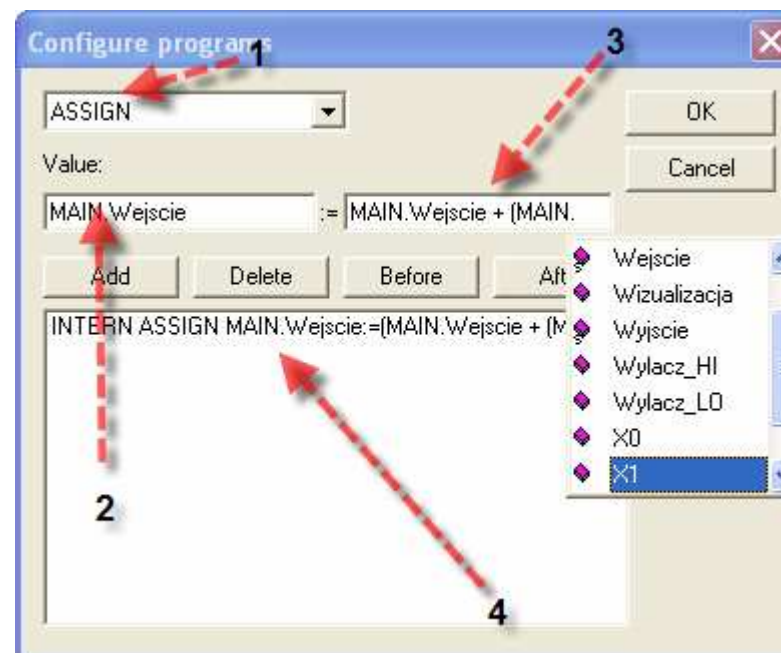
Możliwe jest ograniczenie zakresu wpisywanych wartości i wybór sposobu ich wpisywania.



# Wykonanie programu

W oknie wykonania programu:

- 1: wybieramy typ programu
- 2: wybieramy zmienną,  
na której operacji dokonujemy
- 3: wpisujemy działanie jakie wykonujemy
- 4: dodajemy operację
- 5: wybieramy kolejną operację  
lub wracamy do okna głównego



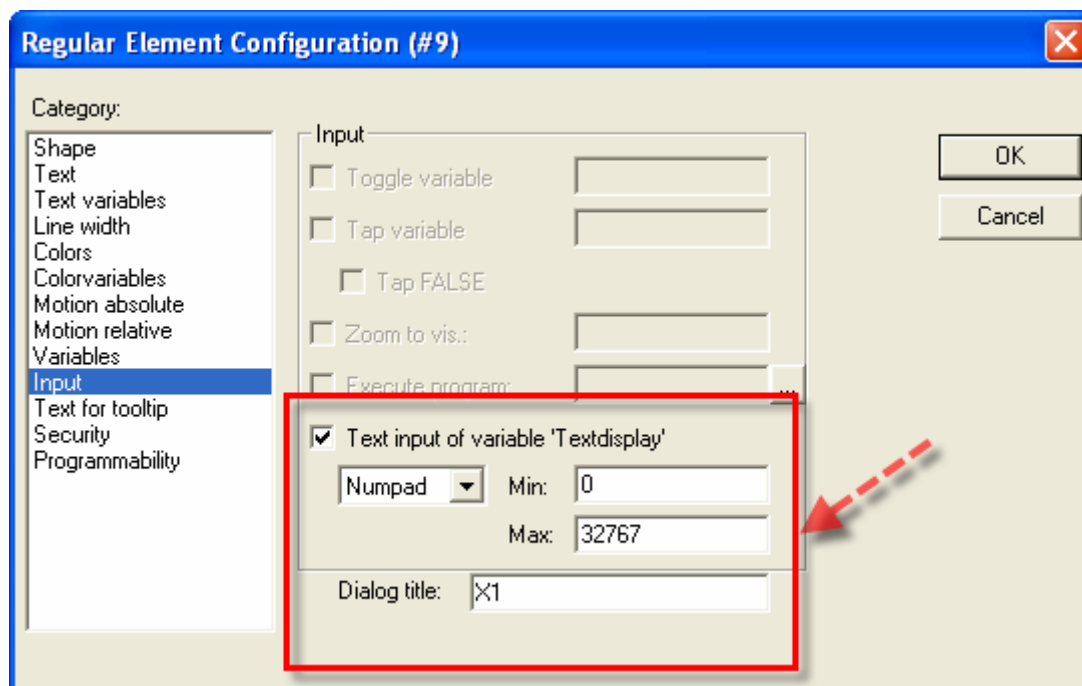
W programie możemy wykorzystywać zmienne wcześniej zdefiniowane, wartości stałe oraz wszystkie operacje matematyczne.



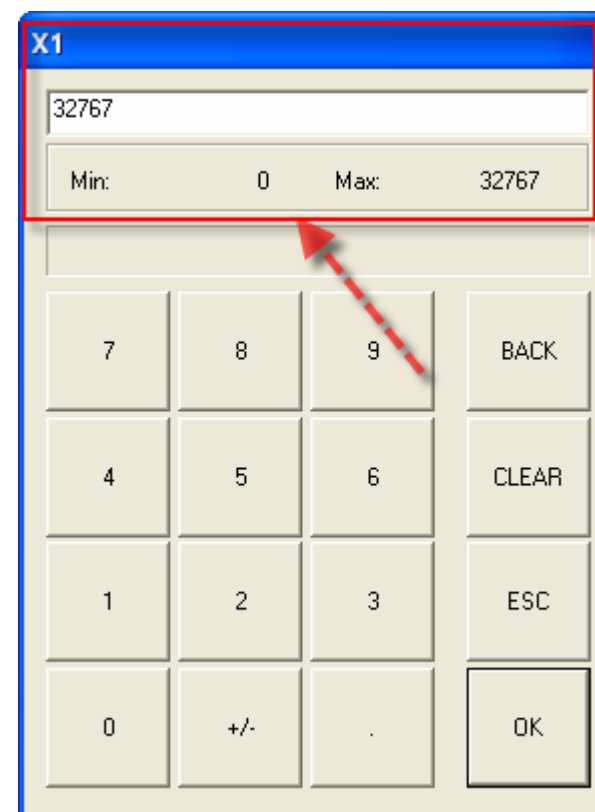
# Wprowadzenie wartości

Możliwy wybór:

- sposobu wprowadzenia danych,
- zakresów,
- nazwy dla pojawiającego się obiektu.



Widok on-line

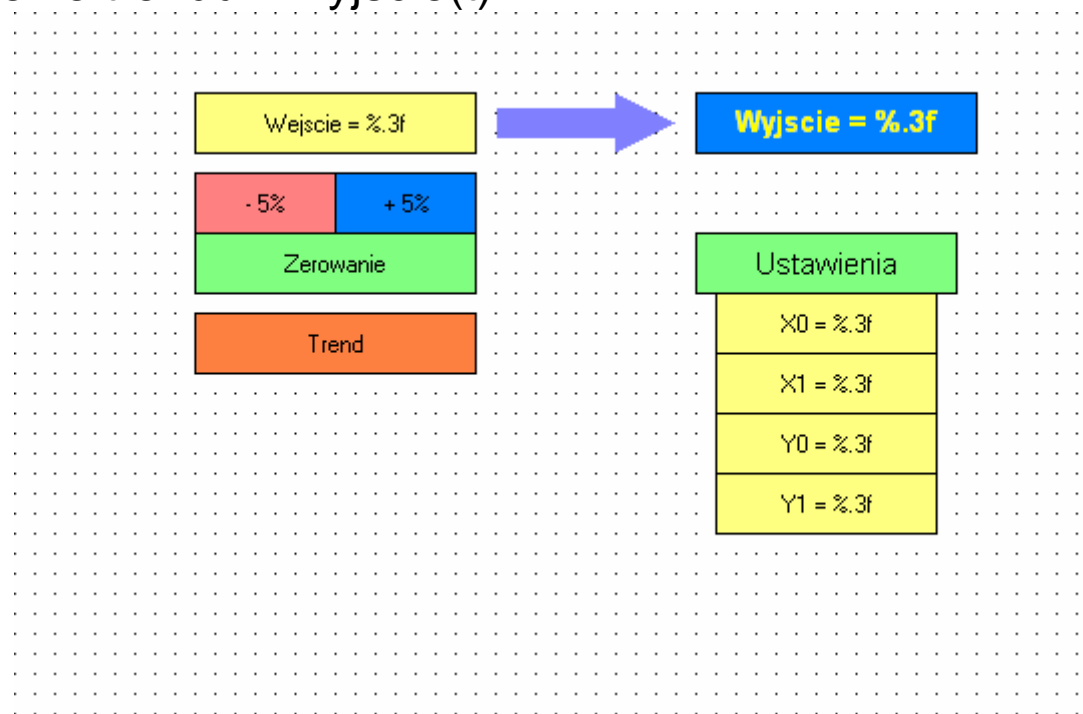




# Wiz – strona 1 (off-line)

Wejscie -> Wyjscie:

- Dodanie / odjęcie 5% wartości zakresu
- Zerowanie wejścia,
- Zmiana zakresów
- Wyświetlenie trendu - Wyjscie(t)

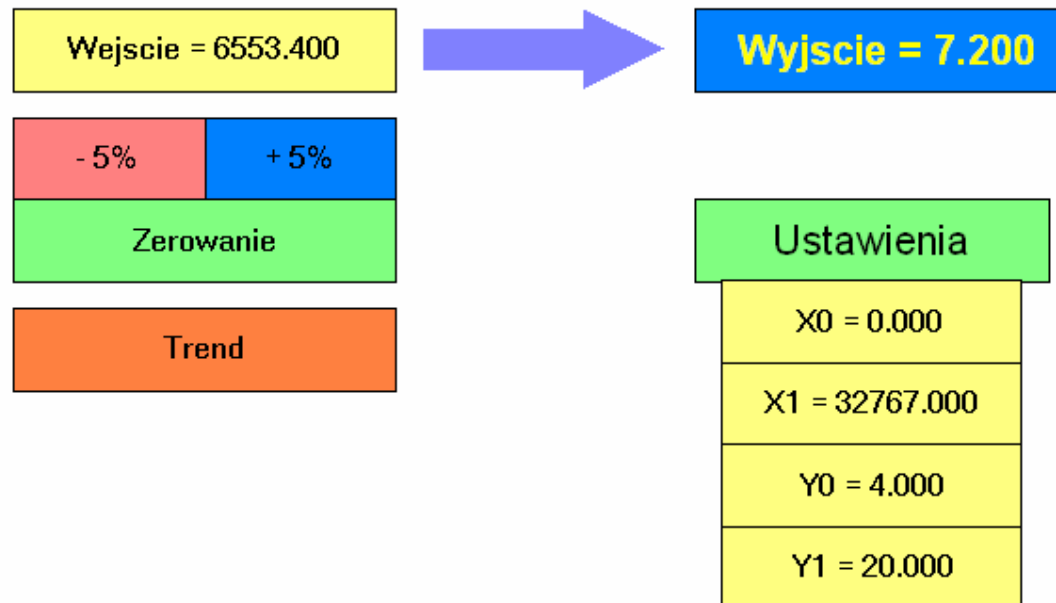




# Wiz – strona 1 (on-line)

Wejscie -> Wyjscie:

- Dodanie / odjęcie 5% wartości zakresu
- Zerowanie wejścia,
- Zmiana zakresów
- Wyświetlenie trendu - Wyjscie(t)





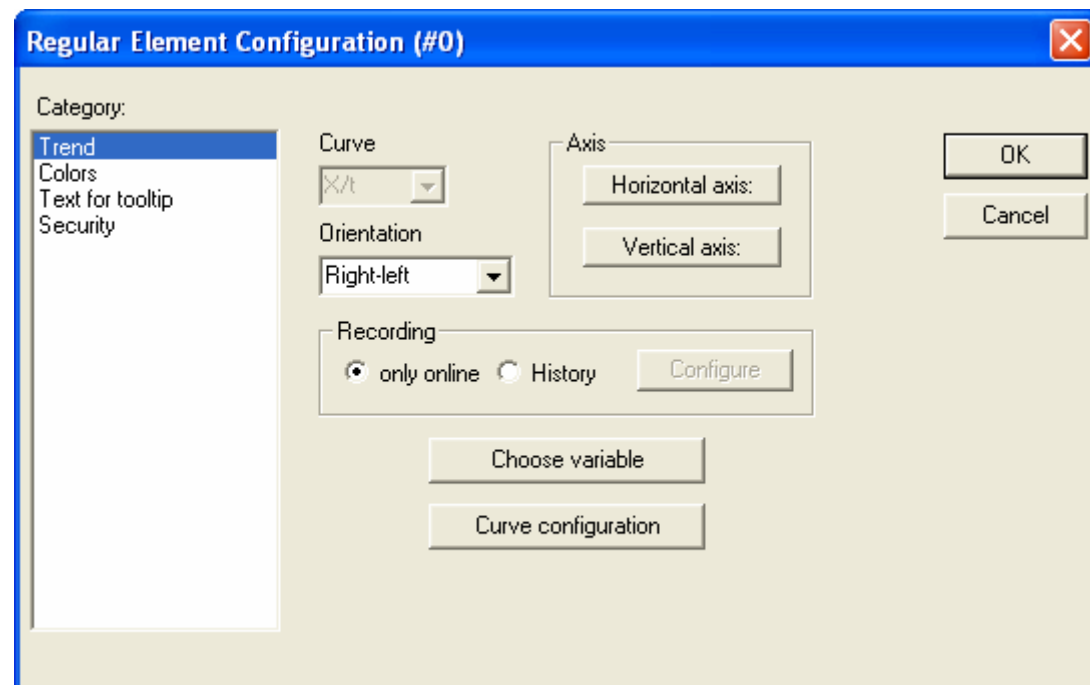
## Trend – strona 2

Dodanie elementu:



Trend:

- Kierunek zmian,
- Konfiguracja obu osi,
- Wybór zmiennej
- Konfiguracja krzywej









## Trend – strona 2

Oś pozioma:

- Jednostki - czas,
- Linie podziałowe,
- Skala,
- Odświeżanie,
- Opis osi.

**Horizontal axis**

<p>Division lines</p> <p><input checked="" type="checkbox"/> visible</p> <p>Scale <input type="text" value="T#20s0ms"/></p> <p></p> <p></p>	<p>Legend</p> <p>Font <input type="button" value="Font"/></p> <p>Scaling <input type="text" value="T#10s0ms"/></p> <p><input checked="" type="checkbox"/> Time Format <input type="text" value="hh':'mm':'ss"/></p> <p><input type="checkbox"/> Date Format <input type="text" value="dd':'MM':'yyyy"/></p>	<p><input type="button" value="OK"/></p> <p><input type="button" value="Cancel"/></p>
<p>Scale</p> <p>Duration <input type="text" value="T#1m0s0ms"/></p> <p>Main <input type="text" value="T#20s0ms"/></p> <p>Sub scale <input type="text" value="T#5s0ms"/></p>	<p>Variables</p> <p>Zoom <input type="text"/></p> <p>Offset <input type="text"/></p>	
<p>Degree of accuracy</p> <p><input type="text" value="T#200ms"/></p>	<p>Symbol bar</p> <p><input type="checkbox"/> use</p>	



## Trend – strona 2

Oś pionowa:

- Jednostki - dowolne,
- Linie podziałowe,
- Skala,
- Opis osi.

**Vertical axis**

Division lines

☒ visible

Scale

Legend

Font

Scaling

OK

Cancel

Scale

☒ left ☐ right

Start

End

Main

Sub scale

Variables

Zoom

Offset

Symbol bar

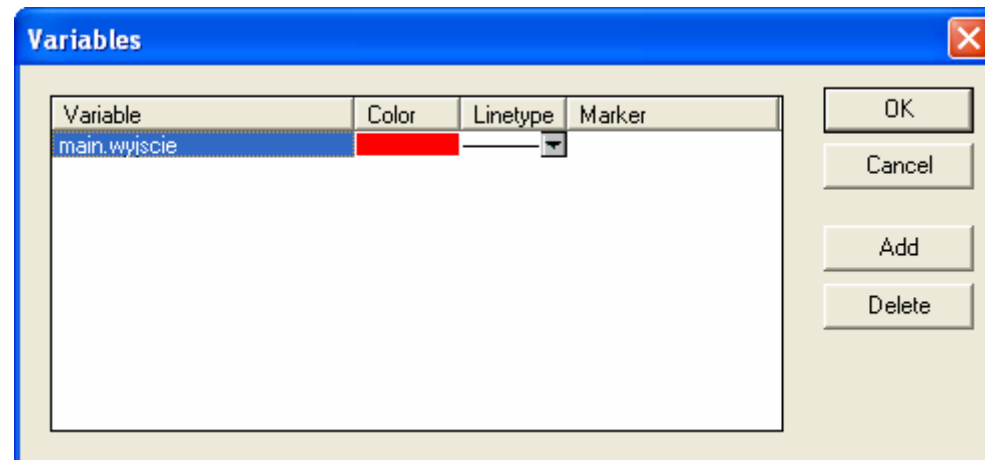
☐ use



## Trend – strona 2

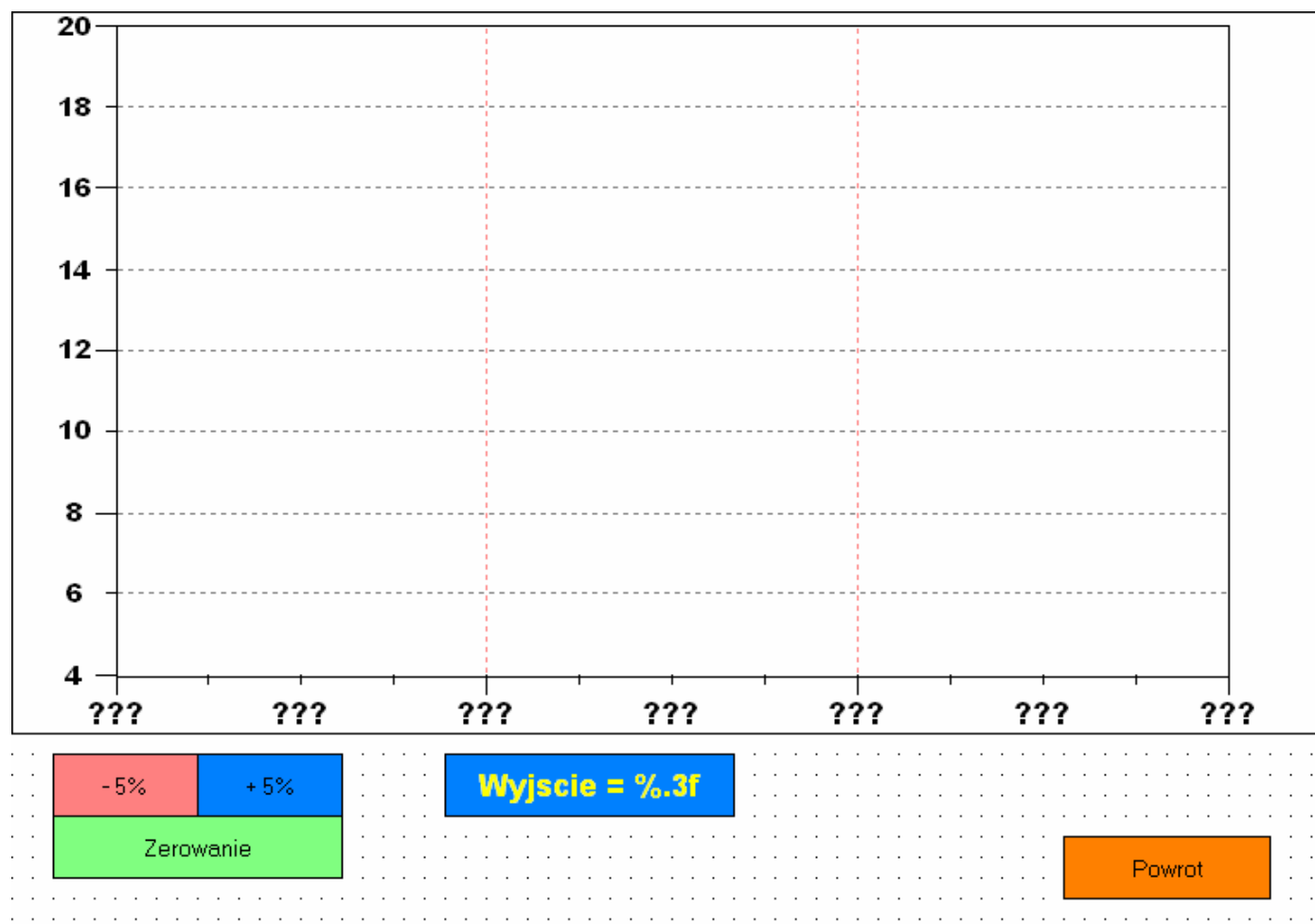
Zmienne:

- Wybór zmiennej,
- Kolor na wykresie,
- Styl linii,
- Znacznik.



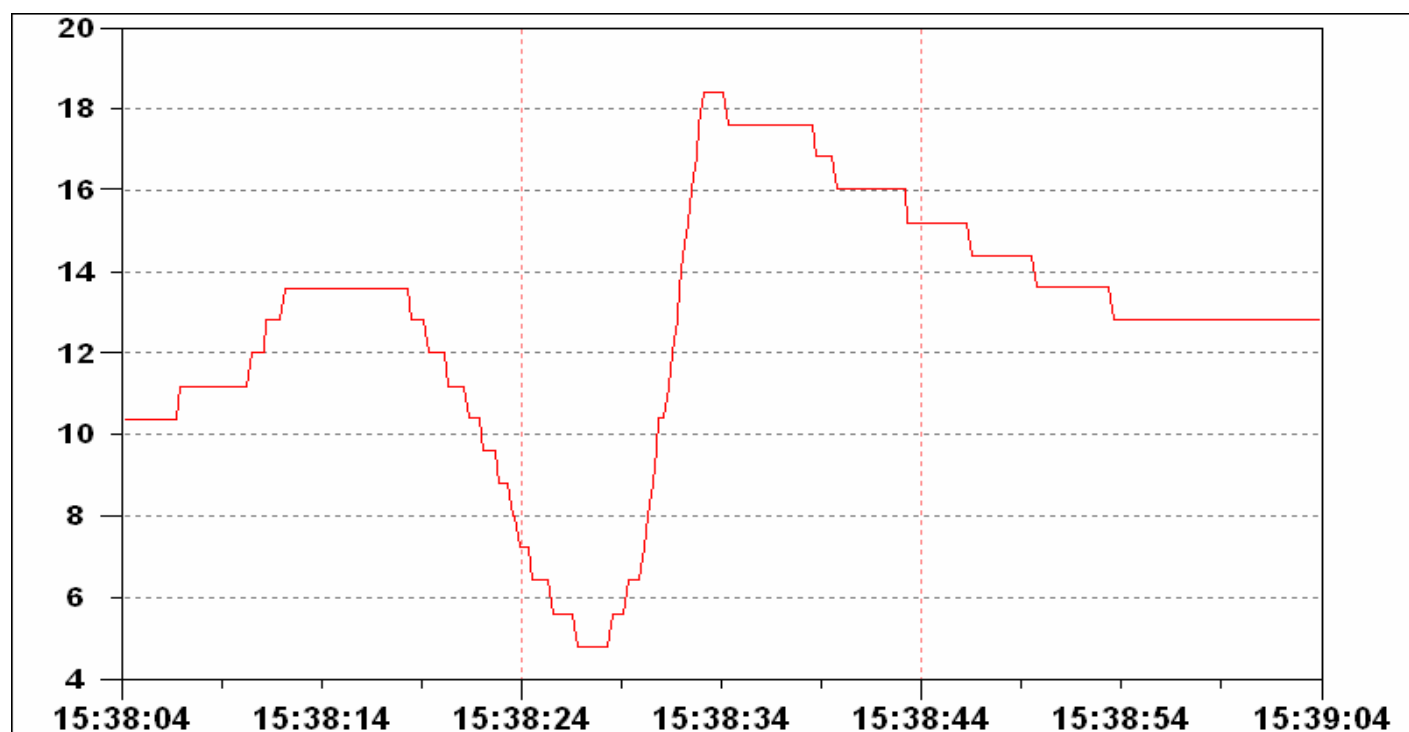


## Trend – strona 2 (off-line)





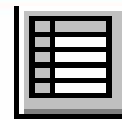
## Trend – strona 2 (on-line)



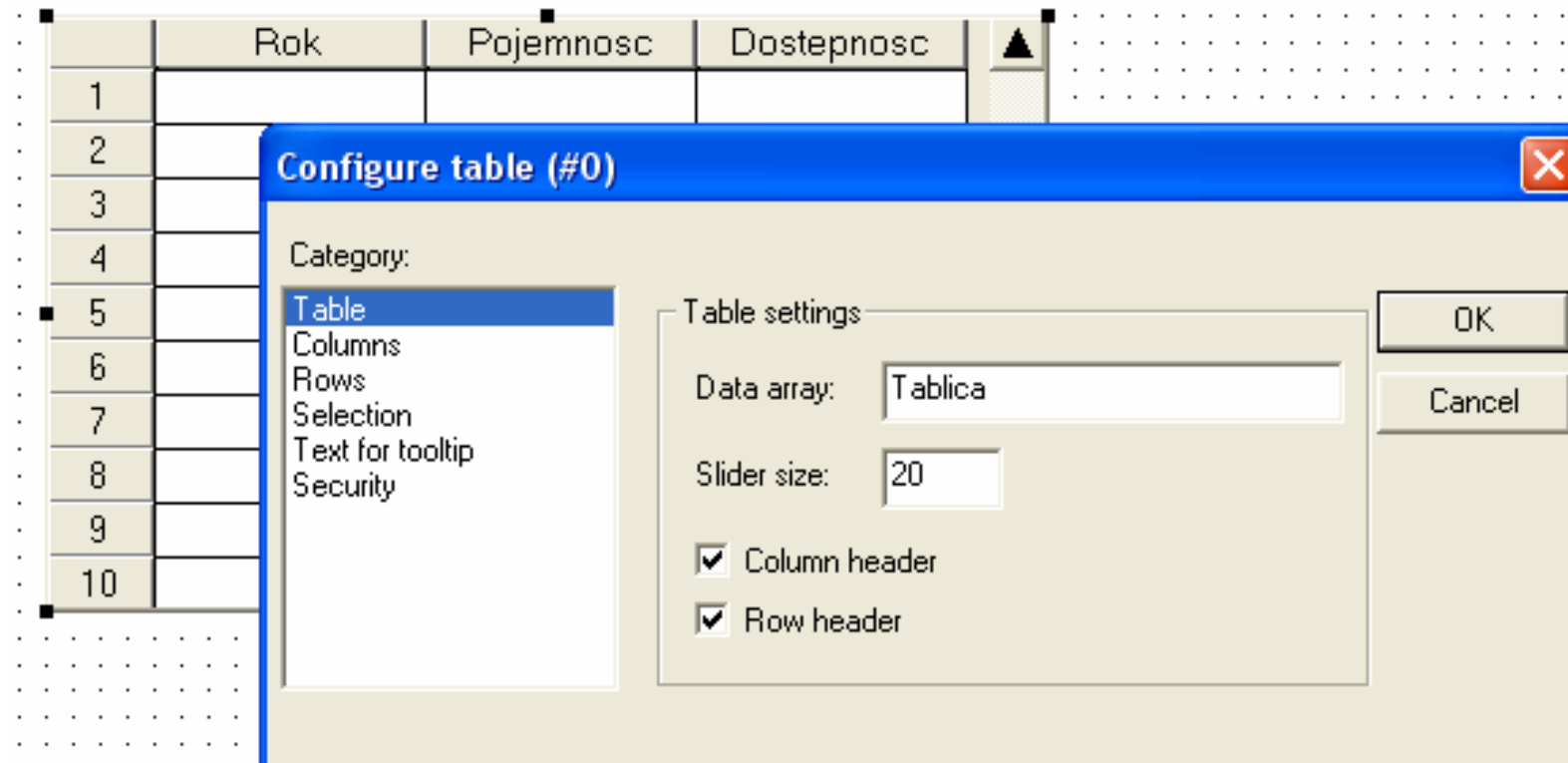
-5%	+5%
Zerowanie	

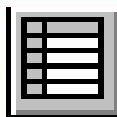
**Wyjscie = 12.800**

Powrot

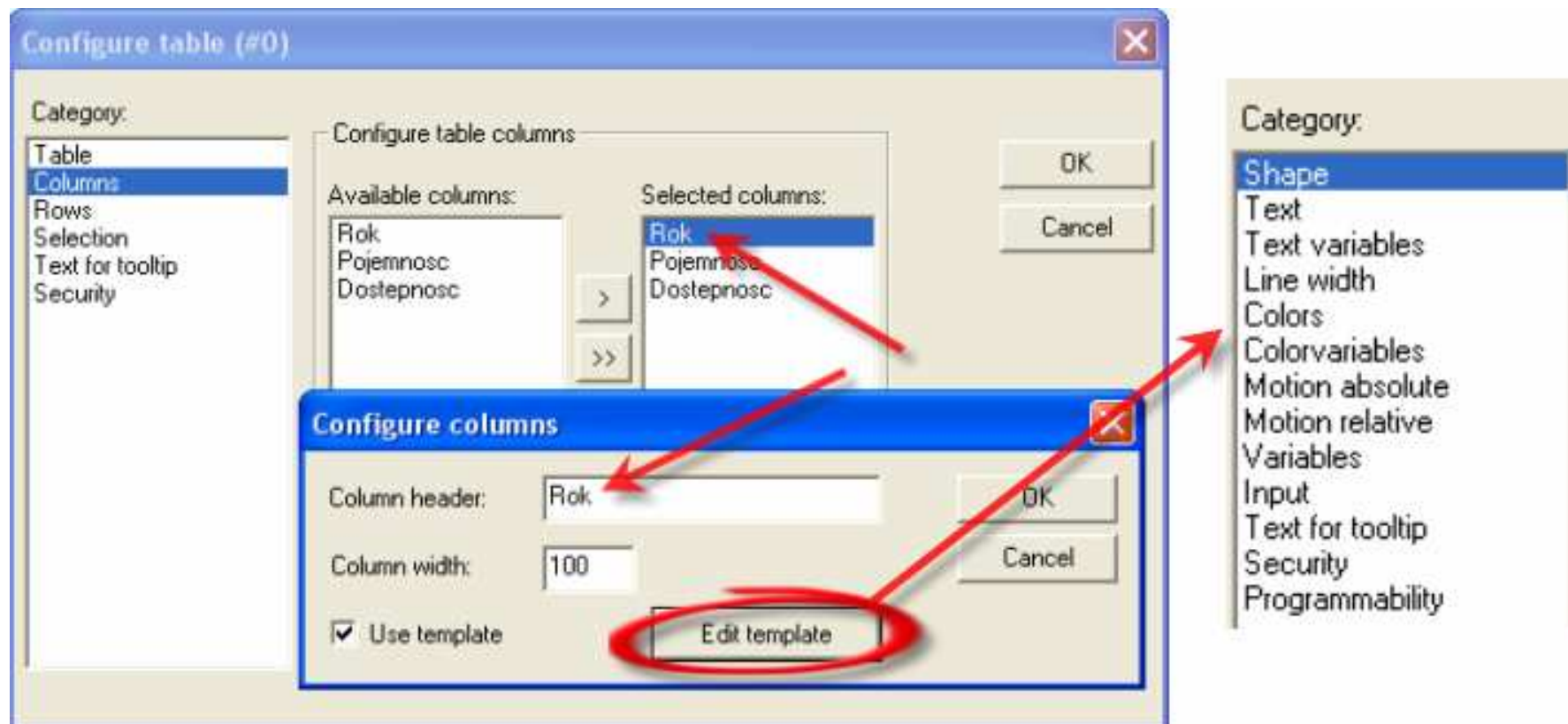


## Tabela – wstawienie elementu



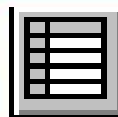


## Tabela – konfiguracja komórek

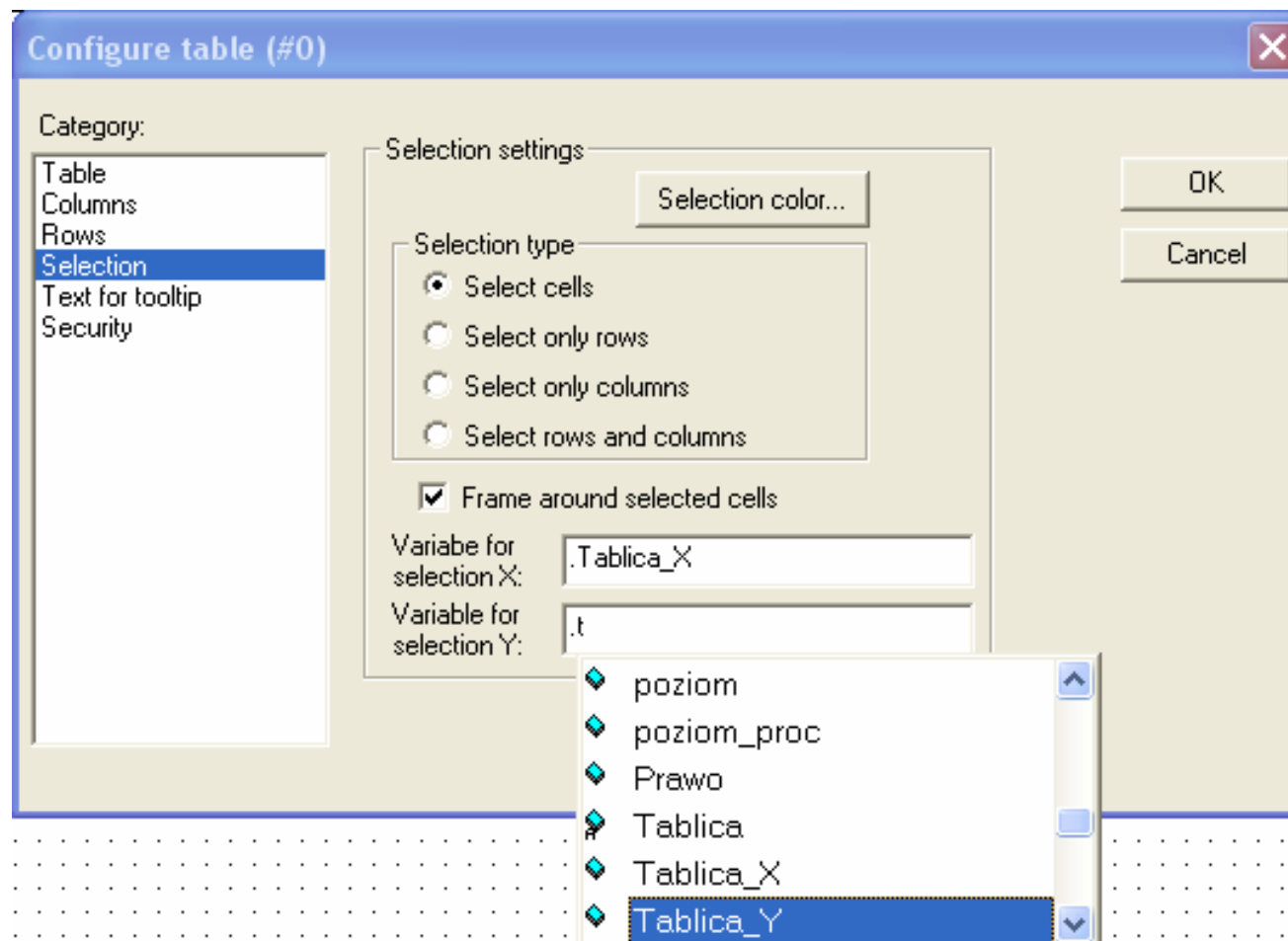


Parametry komórek mogą być konfigurowane tak samo jak każdego elementu w wizualizacji





## Tabela – zaznaczanie elementu



Zdefiniowanie zmiennych wskazujących na element tablicy pozwala np. na śledzenie obróbki