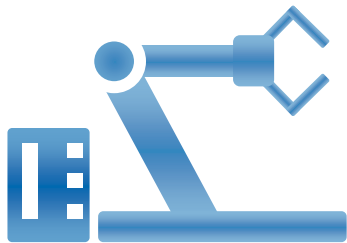


# Programowanie sterowników PLC w CodeSys



**Środowisko CoDeSys 2.3**

**Tworzenie programu w językach FBD i LD**

**Tworzenie wizualizacji**

**CoDeSys** (aktualnie CODESYS) – Controller Development System, zintegrowane środowisko do programowania sterowników zgodne z normą IEC 61131-3.

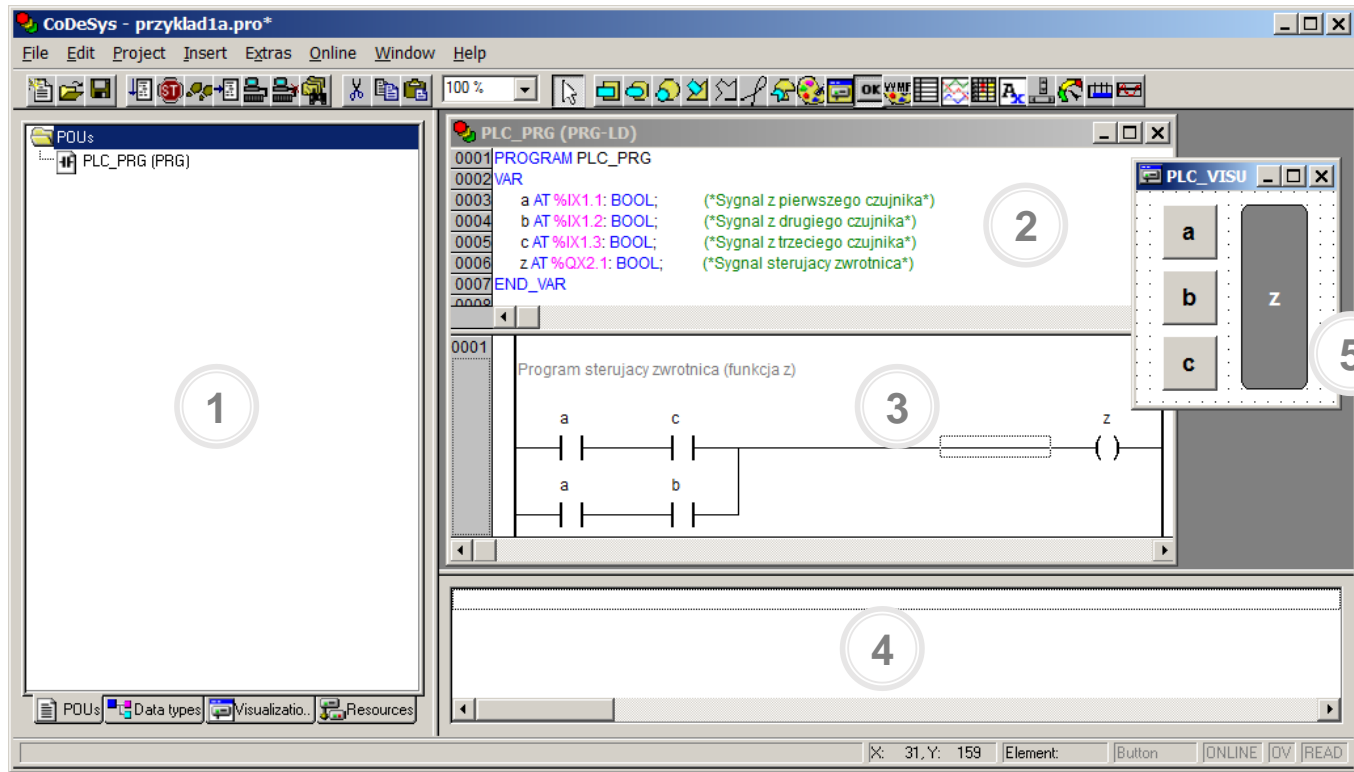
Producent: 3S-Smart Software Solutions, <https://www.codesys.com/>

Aktualne wersje: 2.3, 3.x

## Najważniejsze cechy

- Wsparcie dla języków IL, ST, LD, FBD, SFC,
- Dodatkowy edytor graficzny dla języka CFC (*ang. Continuous Function Chart*), odmiana języka FBD nie opisana w normie IEC 61131.
- Zintegrowany kompilator przekształcający program na kod binarny, który może być załadowany do sterownika
- Wsparcie dla wszystkich popularnych grup procesorów (C166, TriCore, 80x86, ARM/Cortex, Power Architecture, SH, MIPS, BlackFin)
- Wsparcie dla różnych standardów przemysłowych protokołów sieciowych używanych do komunikacji ze sterownikiem (PROFIBUS, CANopen, EtherCAT, PROFINET and EtherNet/IP)
- Zintegrowany edytor ułatwiający tworzenie rozbudowanych wizualizacji ilustrujących przebieg rzeczywistego procesu.

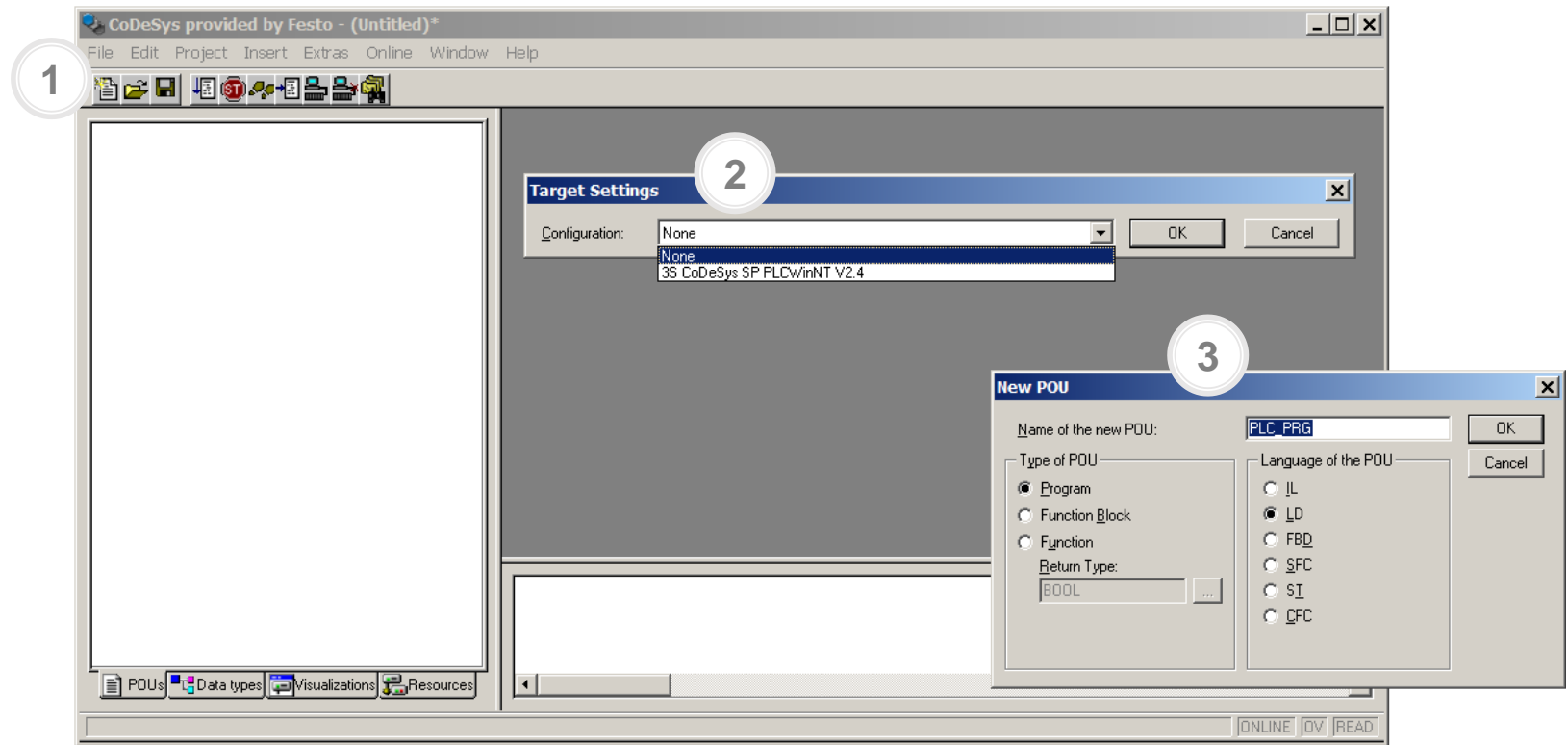
# Aplikacja CoDeSys, wersja 2.3



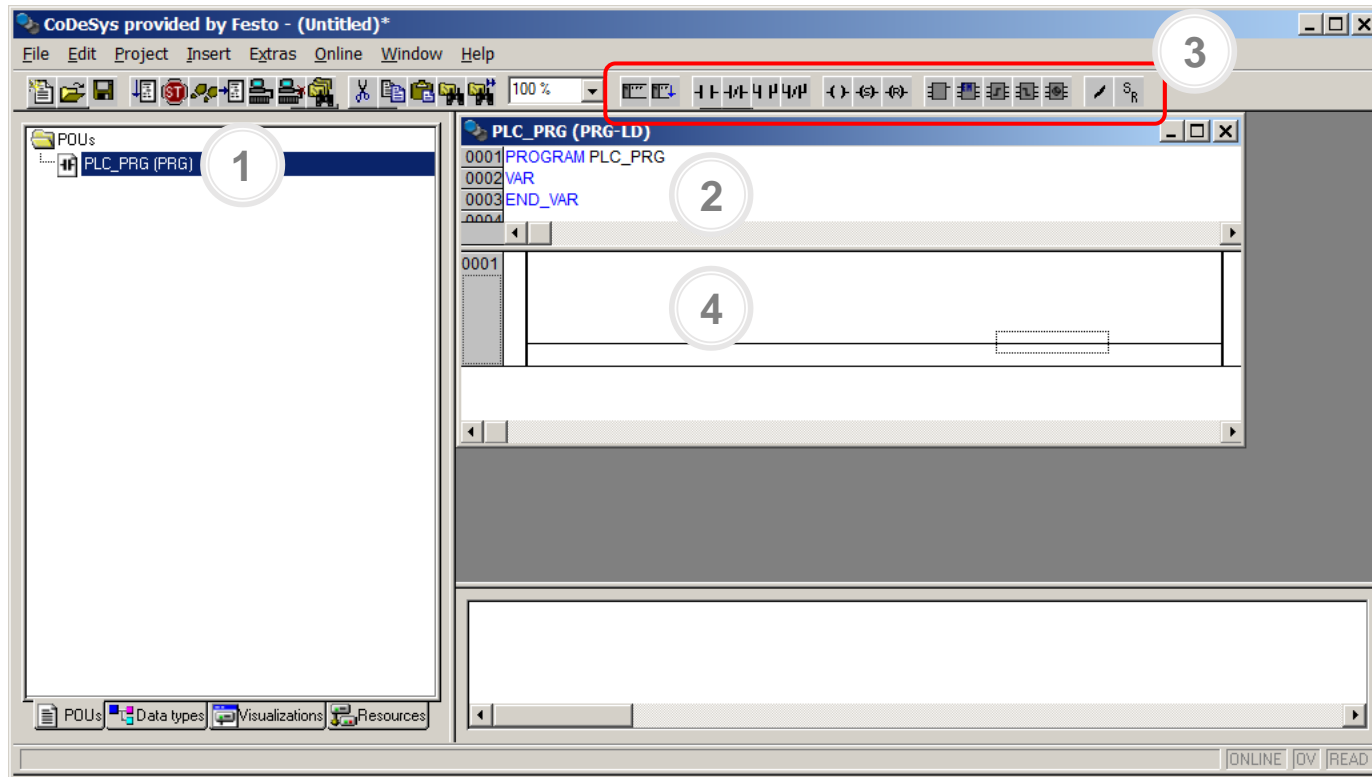
*Program sterownika z przykladu 1. z prostą wizualizacją*

- 1 – okno zasobów: moduły programów, typy danych, wizualizacje, zasoby sterownika (zależnie od wybranej zakładki), 2 – deklaracja zmiennych lokalnych, 3 – kod programu, 4 – okno komunikatów, 5 – wizualizacja procesu.

# Tworzenie nowego projektu



1. Nowy projekt: przycisk New lub opcja File->New
2. Okno Target settings: wybór sterownika lub none (wyłącznie symulacja)
3. Okno New POU: rodzaj jednostki organizacyjnej (Program), nazwa programu (koniecznie PLC\_PRG – nazwa programu głównego) i wybór języka (LD)

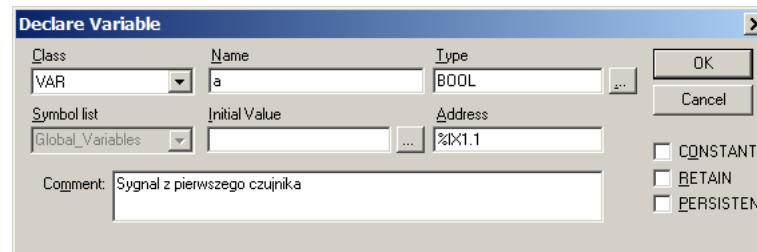


1. Nazwa wybranej jednostki POU.
2. Nagłówek programu i deklaracja zmiennych lokalnych.
3. Pasek narzędziowy edytora LD.
4. Edytor LD z pojedynczym, pustym obwodem.

Deklaracja zmiennych lokalnych aktualnie wybranej jednostki POU może być wykonana ręcznie poprzez wprowadzenie odpowiedniego kodu (w.2,s.11) lub z użyciem okna deklaracji zmiennych.

## Okno deklaracji zmiennych (Declare Variable)

Menu: Edit->Auto declare, klawiatura: Shift+F2



Class klasa zmiennej (VAR – lokalna; , VAR\_GLOBAL – globalna; VAR\_INPUT, VAR\_OUTPUT, VAR\_IN\_OUT – wejście lub wyjście bloków funkcyjnych),

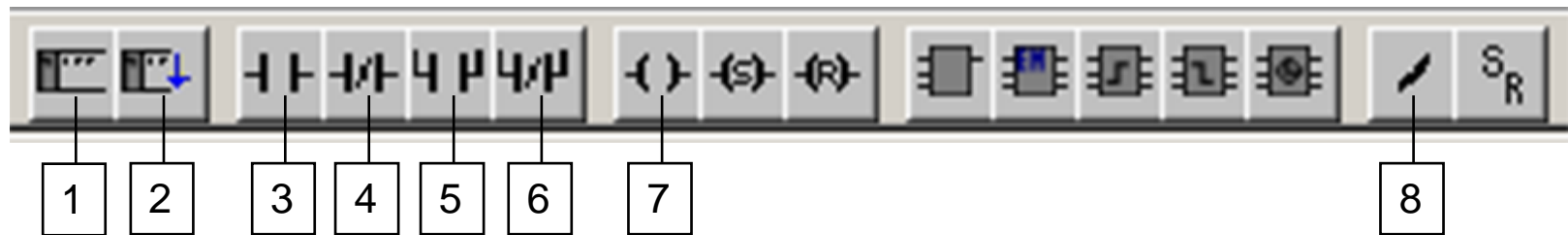
Name nazwa zmiennej,

Type typ zmiennej (w.2,s.12),

Initial Value wartość początkowa zmiennej,

Address adres (lokalizacja) zmiennej jest przypisana (w.2,s.13),

Comment komentarz.



1. Nowy obwód przed aktualnym, Insert->Network(before)
2. Nowy obwód za aktualnym, Insert->Network(after), Ctrl+T
3. Nowy styk zwierny, Insert->Contact, Ctrl+K
4. Nowy styk rozwierny, Insert->Contact(negated), Ctrl+G
5. Styk zwierny połączony równolegle, Insert->Paralel Contact, Ctrl+R
6. Styk rozwierny połączony równolegle,  
Insert->Paralel Contact(negated), Ctrl+R
7. Nowa cewka, Insert->Coil, Ctrl+L
8. Negacja styku (zamienia styk zwierny na rozwierny i odwrotnie),  
menu kontekstowe->Negate, Ctrl+N

Do każdego obwodu można dodać komentarz wybierając Insert->Comment

Kasowanie elementu: Edit->Del, Del

# Przykład 1. Kod programu w języku LD

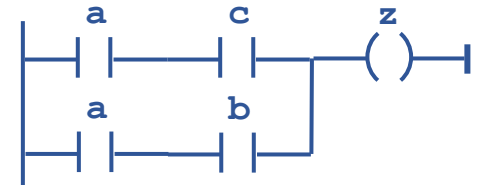
## Minimalna postać dysjunkcyjna

Address	Code	Description
0001	PROGRAM PLC_PRG	
0002	VAR	
0003	a AT %IX1.1: BOOL;	(*Sygnał z pierwszego czujnika*)
0004	b AT %IX1.2: BOOL;	(*Sygnał z drugiego czujnika*)
0005	c AT %IX1.3: BOOL;	(*Sygnał z trzeciego czujnika*)
0006	z AT %QX2.1: BOOL;	(*Sygnał sterujący zwrotnica*)
0007	END_VAR	

The screenshot shows five steps of ladder logic construction:

- Step 1: A normally open contact labeled 'a' is connected to a normally closed contact labeled 'c'.
- Step 2: A second normally open contact labeled 'a' is added in parallel to the first 'a' contact.
- Step 3: A second normally open contact labeled 'b' is added in parallel to the first 'b' contact.
- Step 4: The two parallel branches are connected to a coil (represented by a rectangle) on the right.
- Step 5: The coil is labeled with the variable 'z'.

$$z = ac + ab$$



Deklaracja zmiennych

Dwa styki zwarte [3], nazwy zmiennych

Zaznaczenie dwóch styków  
(strzałki + Shift, lewy przycisk myszy + Shift)

Styk zwiczny połączony równolegle [5]

Drugi styk zwiczny na połączeniu równol. [3],  
nazwy zmiennych

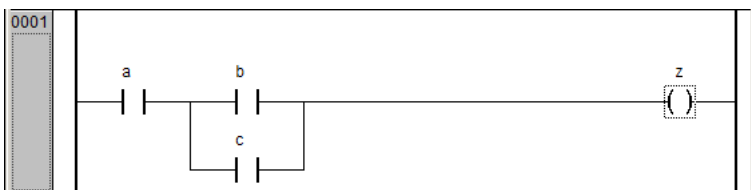
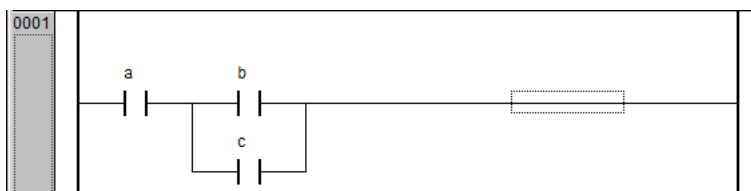
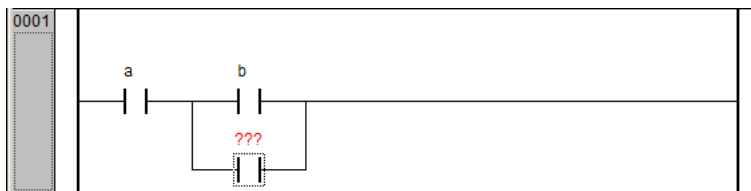
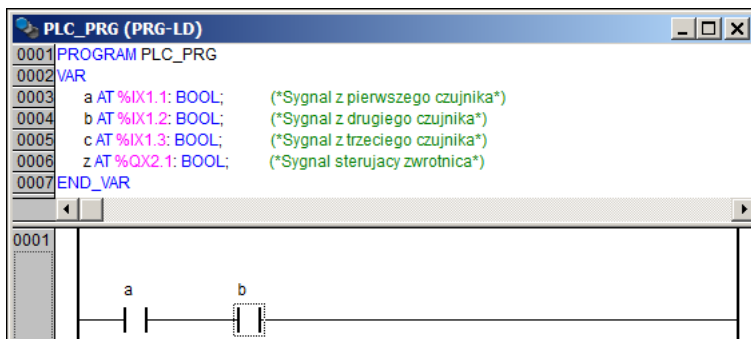
Zaznaczenie z prawej strony obwodu (patrz  
powyżej), cewka [7], nazwa zmiennej

Numery w nawiasach określają przyciski na pasku  
narzędziowym (s.7)

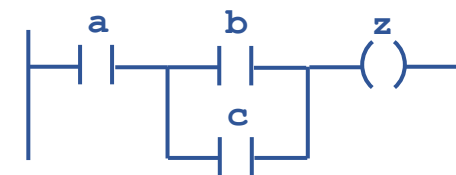


# Przykład 1. Kod programu w języku LD

## Minimalna postać koniunkcyjna



$$z = a(b + c)$$



Deklaracja zmiennych

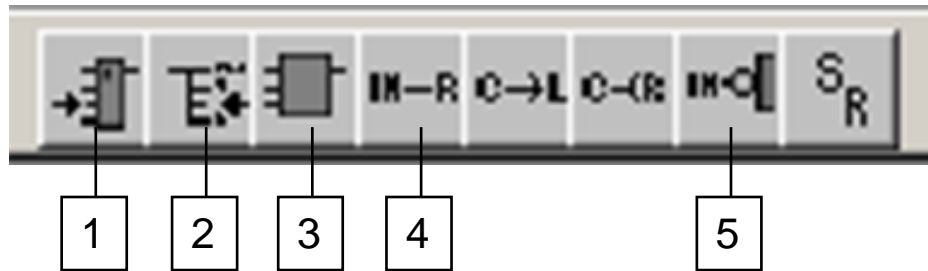
Dwa styki zwarte [3], nazwy zmiennych

Zaznaczony styk "b" (patrz powyżej), styk zwirny połączony równolegle [5]

Przypisanie nazwy zmiennej

Zaznaczenie z prawej strony obwodu (patrz powyżej), cewka [7], nazwa zmiennej

Numery w nawiasach określają przyciski na pasku narzędziowym (s.7)



1. Nowe wejście bloku, Insert->Input, Ctrl+U
2. Nowe wyjście bloku, Insert->Output
3. Nowy blok, Insert->Box, Ctrl+B
4. Połączenie ze zmienną, Insert->Assign, Ctrl+A
5. Negacja sygnału (dodaje/usuwa negację do połączenia),  
menu kontekstowe->Negate, Ctrl+N

Nowy obwód przed aktualnym Insert->Network(before)

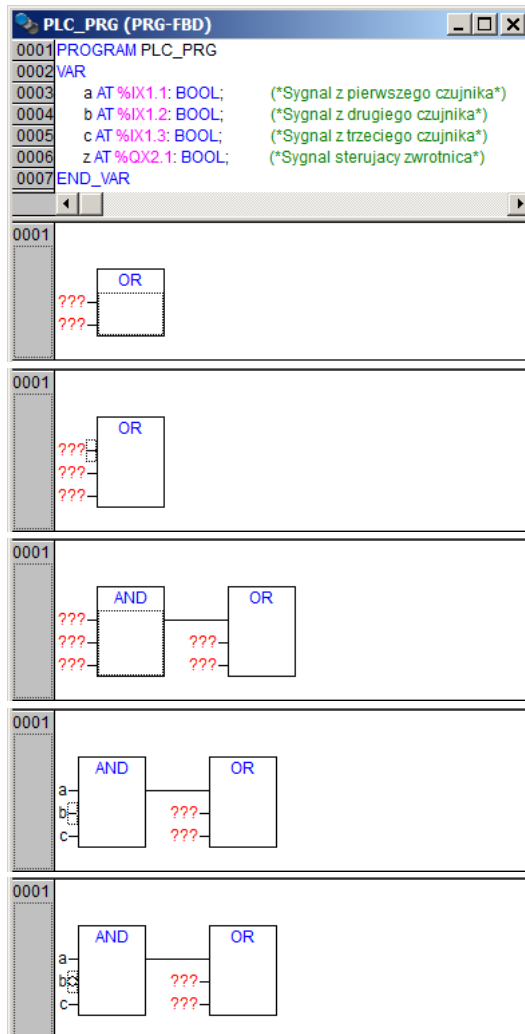
Nowy obwód za aktualnym Insert->Network(after)

Do każdego obwodu można dodać komentarz wybierając Insert->Comment

Kasowanie elementu: Edit->Del, Del

# Przykład 1. Kod programu w języku FBD (1)

Kanoniczna postać dysjunkcyjna  $z = \bar{a}bc + ab\bar{c} + abc$



Deklaracja zmiennych

Nowy blok [3] (domyślnie AND), zmiana typu AND->OR

Zaznaczony blok (patrz powyżej), trzecie wejście [1]

Zaznaczone pierwsze wejście (patrz powyżej),  
nowy blok [3], trzecie wejście [1]

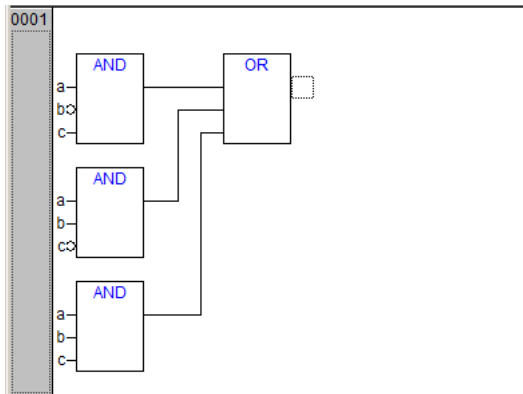
Przypisanie nazw zmiennych

Zaznaczone wejście "b" (patrz powyżej), negacja [5]

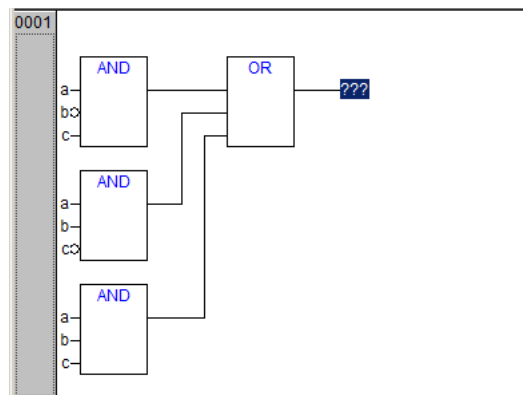
Numery w nawiasach określają przyciski na pasku narzędziowym (s.10)

# Przykład 1. Kod programu w języku FBD (2)

## Kanoniczna postać dysjunkcyjna c.d.

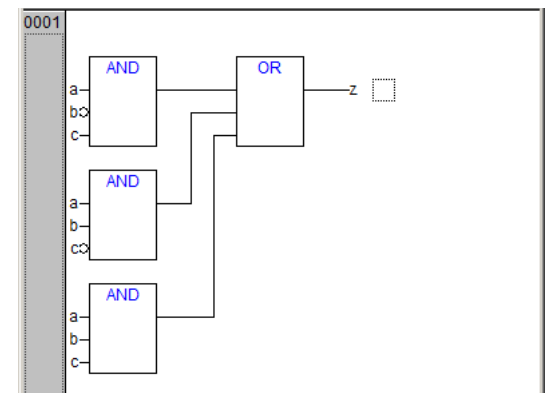


Dwa bloki [3] połączone z drugim i trzecim wejściem bloku OR



Zaznaczone wyjście bloku OR (patrz wyżej), połączenie ze zmienną [4]

Przypisanie nazwy zmiennej wyjściowej



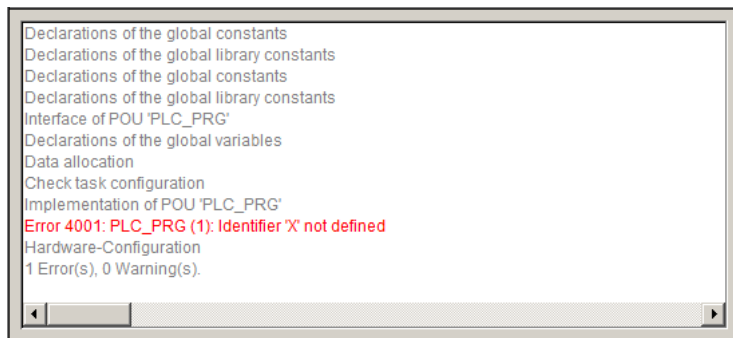
Numery w nawiasach określają przyciski na pasku narzędziowym (s.10)

Program napisany w dowolnym języku tekstowym lub graficznym wymaga **przetłumaczenia na kod binarny**, który może być wykonany przez mikroprocesor sterownika PLC (w.2,s.6). W celu przekształcenia programu na kod binarny należy wykonać kompilację projektu.

## Opcje kompilacji

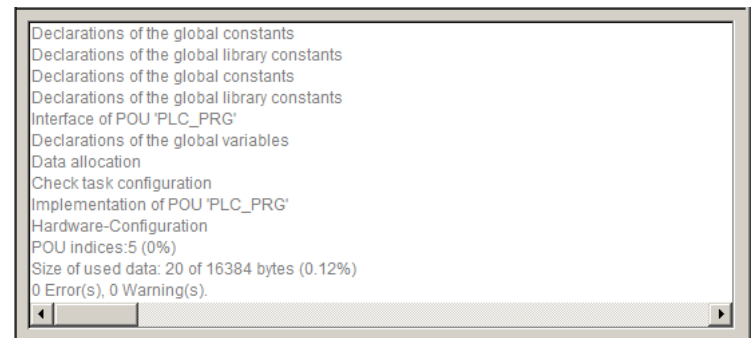
- Project->Build (F11)      kompiluje zmodyfikowane elementy projektu
- Project->Rebuild all      wykonuje pełną kompilację projektu
- Project->Clean all      usuwa wszystkie wyniki poprzednich kompilacji

Proces kompilacji nie będzie ukończony, jeżeli program zawiera błędy. Informacje o znalezionych błędach wyświetlane są w oknie komunikatów (element nr 4 na s.3).



```
Declarations of the global constants
Declarations of the global library constants
Declarations of the global constants
Declarations of the global library constants
Interface of POU 'PLC_PRG'
Declarations of the global variables
Data allocation
Check task configuration
Implementation of POU 'PLC_PRG'
Error 4001: PLC_PRG (1): Identifier 'X' not defined
Hardware-Configuration
1 Error(s), 0 Warning(s).
```

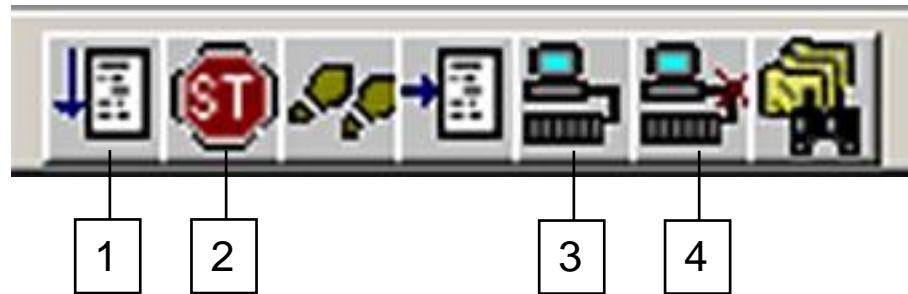
*Kompilacja zakończona błędem  
(niezdefiniowany identyfikator X)*



```
Declarations of the global constants
Declarations of the global library constants
Declarations of the global constants
Declarations of the global library constants
Interface of POU 'PLC_PRG'
Declarations of the global variables
Data allocation
Check task configuration
Implementation of POU 'PLC_PRG'
Hardware-Configuration
POU indices:5 (0%)
Size of used data: 20 of 16384 bytes (0.12%)
0 Error(s), 0 Warning(s).
```

*Kompilacja prawidłowa*

# Uruchomienie programu



1. Uruchomienie programu (tryb RUN), Online->Run, F5
2. Zatrzymanie programu (tryb STOP), Online->Stop, Shift+F8
3. Połączenie ze sterownikiem i wgranie kodu, Online->Login, Alt+F8
4. Zakończenie połączenia ze sterownikiem, Online->Logout, Ctrl+F8

W celu **uruchomienia** programu należy: nawiązać połączenie ze sterownikiem i załadować kod binarny programu [3], uruchomić program [1]. Jeżeli program nie jest skompilowany wybór opcji `Login` automatycznie uruchomi proces kompilacji.

W celu **zakończenia** pracy programu i powrotu do projektu należy zakończyć połączenie ze sterownikiem [4]. Opcja `Stop` [2] zatrzymuje program, ale nie przerywa połączenia, program zatrzymany może być uruchomiony ponownie.

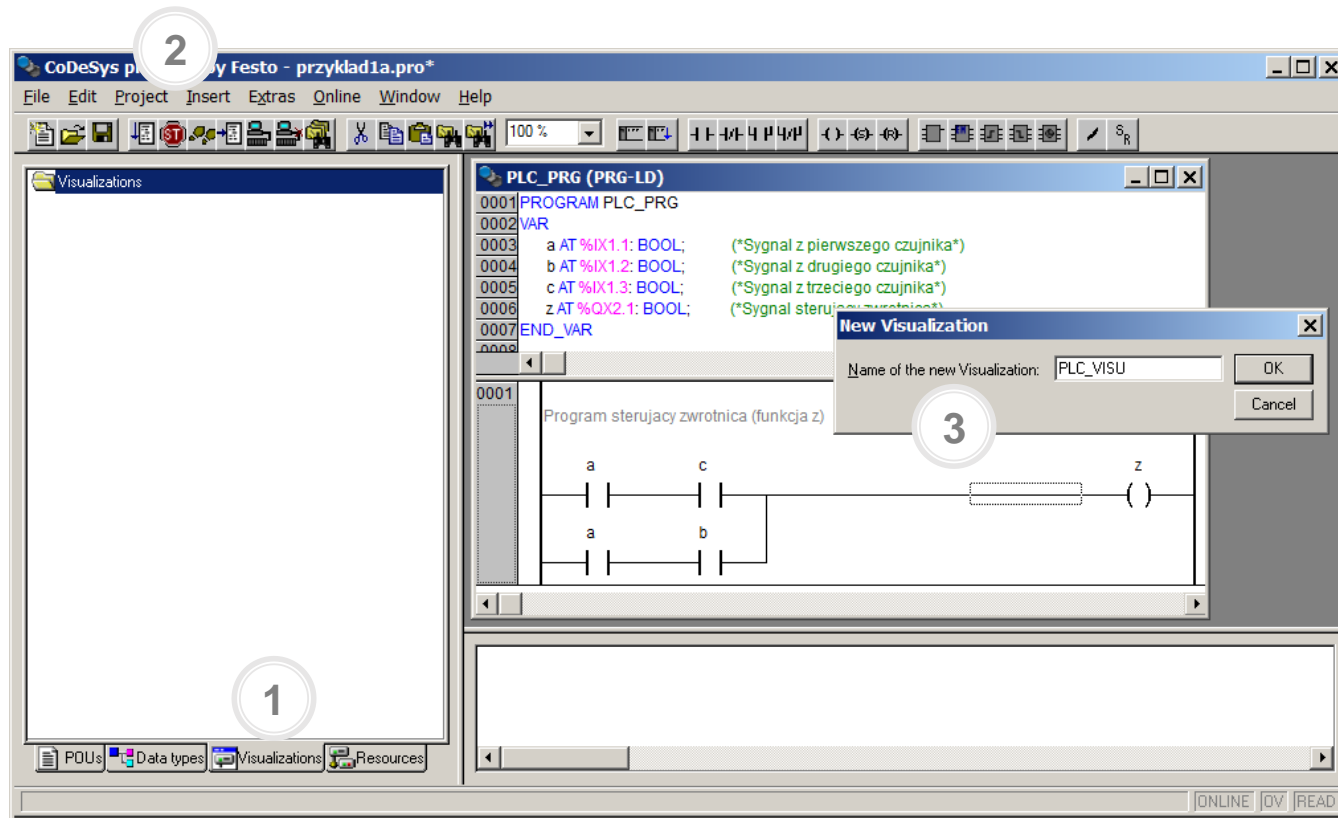
W przypadku **symulacji** należy zaznaczyć opcję `Online->SimulationMode`, proces uruchamiania i zatrzymywania programu przebiega identycznie. Tryb symulacji jest wybrany domyślnie jeżeli jako `Target` jest wskazane `none` (s.4).

**Wizualizacja** jest graficznym interfejsem, który umożliwia obsługę danych oraz obserwację pracy sterownika i/lub sterowanego procesu.

CoDeSys posiada zintegrowany edytor umożliwiający tworzenie **masek** (okien) wizualizacji powiązanych bezpośrednio ze zmiennymi sterownika. Maski utworzone w systemie CoDeSys można bez żadnych modyfikacji wykorzystać w czterech wariantach przebiegu (również równolegle):

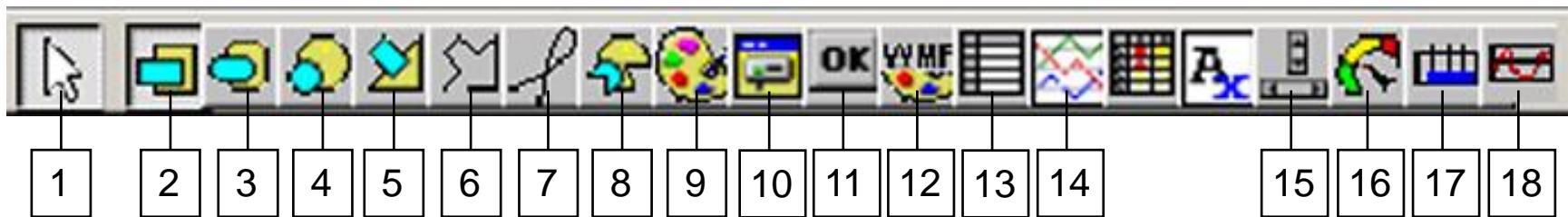
1. Bezpośrednio w systemie programowania – maska jest wyświetlana w systemie CoDeSys, programista otrzymuje obraz rzeczywistej wizualizacji taki jak inni użytkownicy (np. operator panelu HMI).
2. Wizualizacja dla Windows – maska jest wyświetlana na komputerze bez kompletnego interfejsu programistycznego (wymagane zainstalowanie aplikacji CoDeSys HMI).
3. Wizualizacja sieciowa – pliki wizualizacji (plik XML + aplet Java) są przechowywane w pamięci sterownika i wyświetlane w oknie przeglądarki.
4. Wizualizacja elementu docelowego – wizualizacja jest wyświetlana na specjalizowanych panelach zintegrowanych lub połączonych ze sterownikiem PLC (np. panele HMI).

# Tworzenie nowej wizualizacji



1. Zakładka Visualizations w oknie zasobów
2. Opcja Project->Object->Add lub menu kontekstowe->Add Object
3. Okno New Visualization, nazwa wizualizacji (PLC\_VISU jest domyślną nazwą wizualizacji startowej, może być zmieniona)





1. Selektor obiektów
- 2-7. Prostokąt, zaokrąglony prostokąt, elipsa, wielokąt, łamana, krzywa
8. Wycinek koła
9. Mapa bitowa – grafika bitmapowa w formacie bmp, tif, jpg
10. Wizualizacja – obiekt wizualizacji osadzony wewnątrz bieżącej
11. Przycisk
12. Plik WMF – grafika wektorowa w formacie wmf
13. Tabela – prezentacja wartości zbioru danych
14. Trend – graficzna prezentacja zmiennych w pewnym horyzoncie czasowym
15. Pasek przewijania
- 16-17. Wskaźniki (wskazówkowy i słupkowy)
18. Histogram – graficzna prezentacja zbioru danych w formie wykresu

Wszystkie elementy mają swoje odpowiedniki w menu `Insert`.

W celu **wstawienia nowego elementu** wizualizacji należy wskazać go w pasku narzędziowym (wybrać opcję w menu `Insert`) i zaznaczyć prostokątny obszar na masce wizualizacji. W przypadku wybranych elementów przed wstawieniem pojawia się okno dialogowe z pytaniem o nazwę pliku ([9], [12]), nazwę wizualizacji ([10]), lub okno konfiguracyjne ([13]-[18]).

## Edycja elementów

1. Zaznaczenie: selektor obiektów [1] i kliknięcie na elemencie;
2. Zaznaczenie kilku elementów: `Shift+kliknięcie`, zaznaczenie obszaru;
3. Przesuwanie, zmiana rozmiarów: zaznaczenie i przeciąganie;
4. Kopiowanie: opcja `Edit->Copy`, `Edit->Paste`;
5. Usuwanie: opcja `Edit->Del`, klawiatura: `Del`;
6. Przenoszenie do przodu/tyłu: `Extras->Bring to front`, `Send to back`;
7. Wyrównanie: opcja `Extras->Align (Left, Right, Top, ...)`;
8. Grupowanie: opcja `Extras->Group`, `Extras->Ungroup`;
9. Cofanie zmian: opcja `Edit->Undelete`.

*Uwaga: Zgrupowane elementy mają wspólne właściwości, każda operacja (przenoszenie zmiana rozmiarów) dotyczy całej grupy, rozgrupowanie usuwa ustawione parametry.*

# Konfigurowanie elementów wizualizacji

Każdy element wizualizacji może być **konfigurowany** przy pomocy okna dialogowego Element Configuration (opcja Extras->Configure, menu kontekstowe, podwójne kliknięcie).

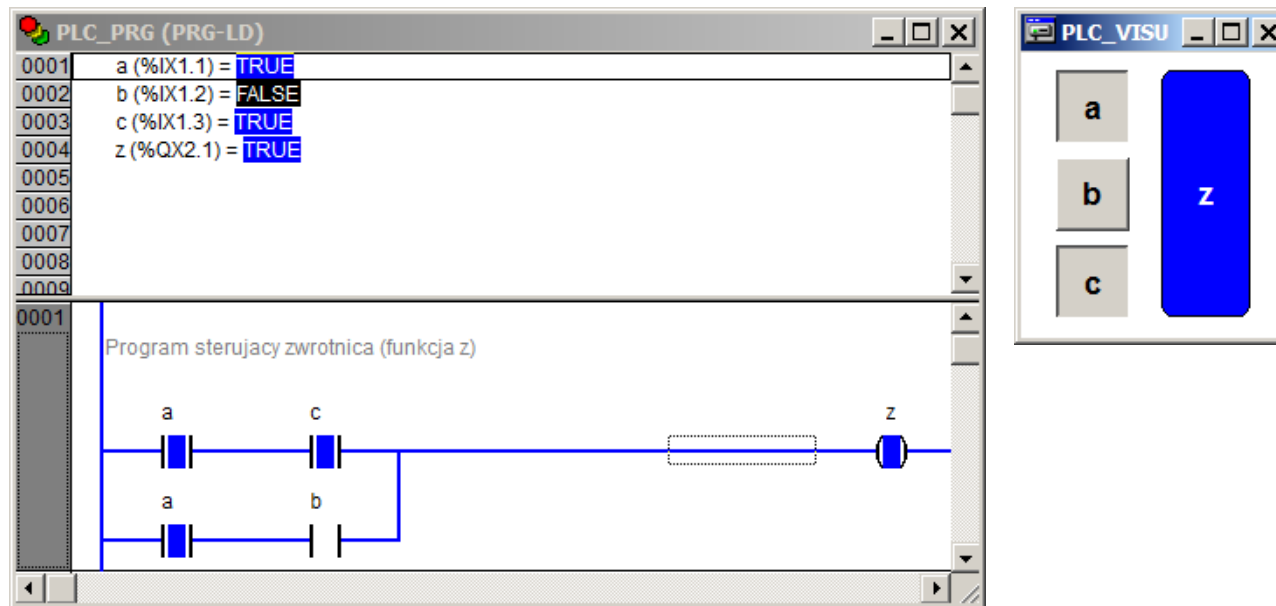
## Najważniejsze kategorie opcji konfiguracyjnych

1. `Text` – opis elementu (etykieta), sposób jego wyrównania i rodzaj czcionki;
2. `Colors` – kolor normalny i kolor alarmu, element jest wyświetlany w kolorze normalnym, jeżeli skojarzona zmienna projektowa (patrz p.3.) ma wartość `False` i w kolorze alarmu jeżeli zmienna ma wartość `True`;
3. `Variables` – skojarzone zmienne, które wpływają na stan elementu:
  - `Change color` – zmiana koloru (element w kolorze normalnym jeżeli zmienna ma wartość `TRUE` w kolorze alarmu jeżeli `FALSE`);
4. `Input` – reakcja na wejście (kliknięcie myszą, dotknięcie ekranu, itp.):
  - `Toggle variable` – kliknięcie przełącza wartość wskazanej zmiennej (`TRUE->FALSE`, `FALSE->TRUE`).

*Uwaga I:* Odwołanie do zmiennej projektowej: `nazwa_POU.nazwa_zmiennej`

*Uwaga II:* W każdym polu wyboru zmiennej klawisz `F2` otwiera ono dialogowe z listą dostępnych zmiennych projektowych.

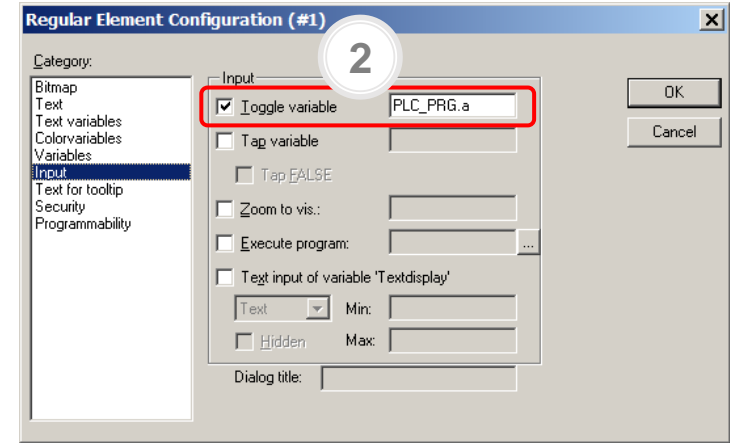
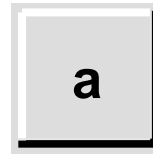
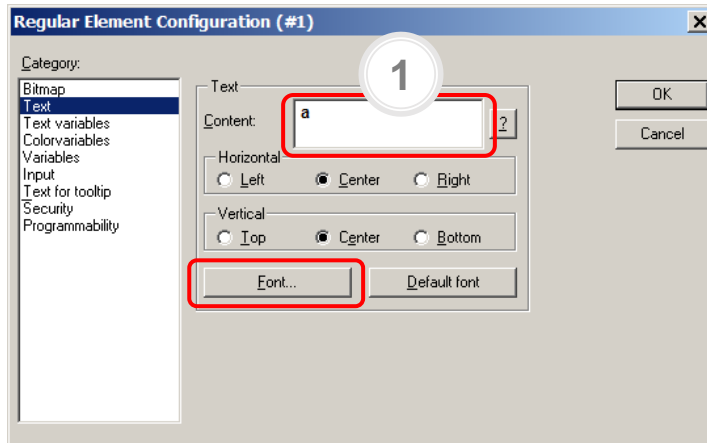
# Przykład 1. – podstawowa wizualizacja



Zmienne projektowe odpowiadające sygnałom wejściowym a, b, c oraz sygnałowi wyjściowemu z powiązane z elementami wizualizacji opisanymi zgodnie z nazwami zmiennych. Wartości FALSE odpowiada kolor szary, wartości TRUE kolor niebieski. Kliknięcie na elementach reprezentujących sygnały wejściowe przełącza wartość powiązanej zmiennej (możliwość symulacji różnych stanów wejściowych). Element reprezentujący wyjście sygnalizuje stan zwrotnicy (nie reaguje na kliknięcie).

*Projekt dostępny na stronie przedmiotu jako "Przykład1a"*

# Przykład 1. – konfiguracja wejść

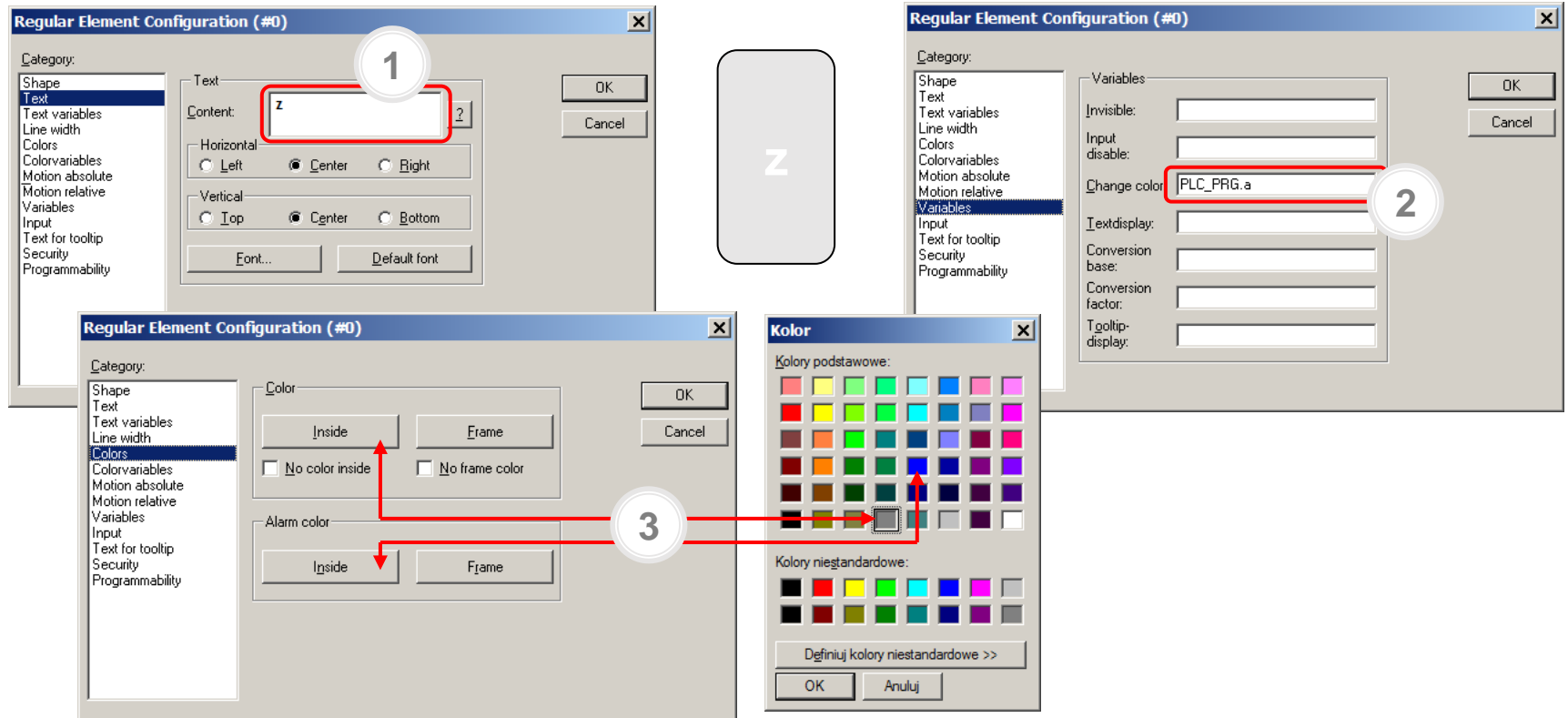


## Element "a" (Button)

1. Opis przycisku (Text): etykieta "a", font Arial, 12pt, pogrubiony.
2. Wejście (Input): po kliknięciu przełącza wartość zmiennej PLC\_PRG.a (opcja Toggle variable), stan przycisku (włączony/wyłączony) odpowiada wartości zmiennej.

Wejścia (przyciski) "b" i "c" analogicznie.

# Przykład 1. – konfiguracja wyjść



Element "z" (Rounded rect)

1. Opis (Text): etykieta "z", font Arial, 12pt, pogrubiony.
2. Zmienna (Variables): PLC\_PRG.z (stan elementu zależy od wartości tej zmiennej).
3. Kolory (Colors): szary gdy zmienna ma wartość FALSE, niebieski gdy TRUE (alarm).

# Wyświetlanie wartości zmiennych

Etykieta (Element Configuration->Text->Content) może wyświetlać tekst statyczny lub wartość zmiennej (projektowej i systemowej). Zmienna projektowa, której wartość będzie wyświetlana jest określana w kategorii Variables->Text display.

Miejsce i sposób wyświetlenia zmiennej definiuje **znacznik formatu**:

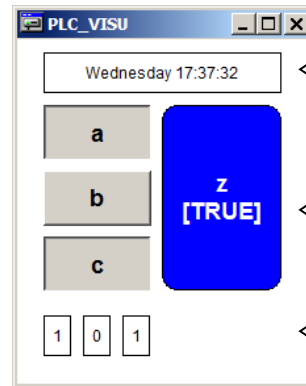
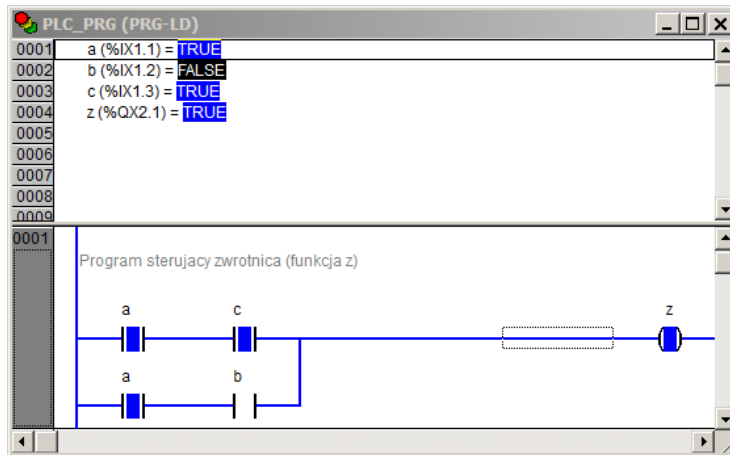
- %s – wartość wyświetlana jako ciąg znaków;
- %d, %o, %x – kolejno: liczba dziesiętna, ósemkowa, szesnastkowa;
- %f – liczba rzeczywista;
- %% – znak "%".

Symbol %t oznacza wyświetlanie **czasu systemowego**, dostępne znaczniki:

- %x, %X – kolejno data w formacie m/d/r i godzina w formacie g:m:s;
- %d, %m, %y, %Y – kolejno: dzień, miesiąc, rok w skrócie, rok pełny;
- %H, %I, %M, %S – kolejno: godzina (24), godzina (12), minuta, sekunda;
- %B, %b – kolejno: pełna i skrócona nazwa miesiąca;
- %A, %a – kolejno: pełna i skrócona nazwa dnia tygodnia.

Kombinacja `Ctrl+ENTER` wprowadza znak przejścia do nowego wiersza.

# Przykład 1. – wartości zmiennych



czas systemowy (1)

wartość zmiennej z w formie tekstowej (2)

wartości zmiennych a,b,c numerycznie (3)

Zmiany w wizualizacji:

1. Czas systemowy:

nowy element Rectangle, Text->Content: %t%A %X

2. Zmiany w konfiguracji wyjścia:

Text->Content: z [%s], Variables->Text display: PLC\_PRG.z

3. Wartości zmiennych wejściowych

trzy nowe elementy Rectangle, konfiguracja:

Text->Content: %d,

Variables->Text display: PLC\_PRG.a, PLC\_PRG.b, PLC\_PRG.c



# Ukrywanie elementów wizualizacji

---

Ukrywanie elementów wizualizacji pozwala na tworzenie dynamicznych masek, które zmieniają wygląd zależnie od stanu wejść i wyjść sterownika.

## Przykłady zastosowań

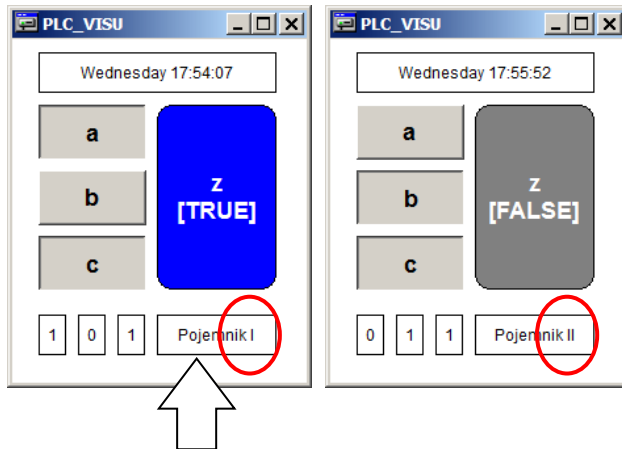
- Tekstowe opisy stanu procesu (odpowiednio ukrywane/pokazywane elementy z różnymi etykietami),
- Elementy graficzne wizualizujące stan procesu (rozmiar lub położenie elementu, poziom cieczy w zbiorniku, stan zaworu, załączony alarm, itp.),
- Strzałki pokazujące drogę transportowanego surowca lub detalu.

**Widoczność elementu wizualizacji** określa zmienna ustawiona w oknie konfiguracji w polu `Variables->Invisible`. Element jest niewidoczny, jeżeli zmienna ma wartość `TRUE`, widoczny dla wartości `FALSE`.

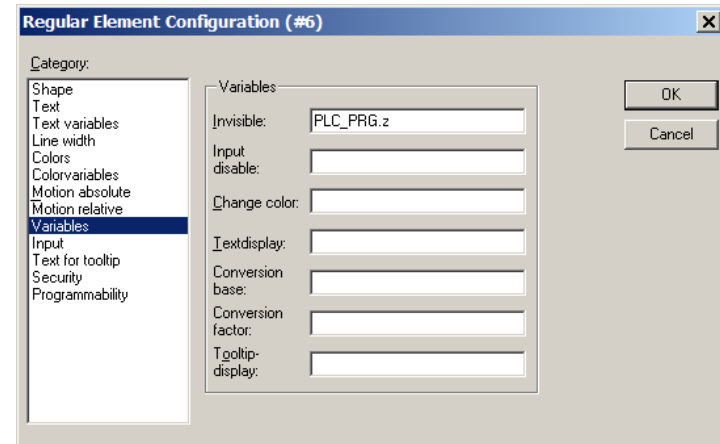
*Uwaga I:* W każdym polu zmiennej można wprowadzić wyrażenie. W przypadku wyrażeń logicznych dostępne są operatory `NOT`, `AND`, `OR` operator porównania `"=`" oraz nawiasy zmieniające kolejność działań.

*Uwaga II:* Lista wszystkich elementów wizualizacji dostępna jest w oknie dialogowym `Element list` (menu `Extras` lub menu kontekstowe). Z jego poziomu można uzyskać dostęp do przysłoniętych elementów wizualizacji.

# Przykład 1. – opis stanu procesu



Dwa elementy Rectangle  
położone jeden nad drugim



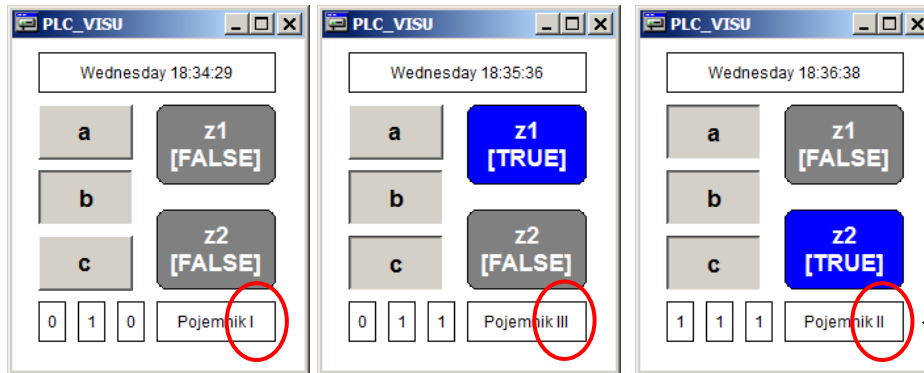
Konfiguracja elementu Rectangle #6

## Konfiguracja:

- Rectangle #5:  
Text->Content: **Pojemnik I**  
Variables->Invisible: NOT PLC\_PRG.z
- Rectangle #6:  
Text->Content: **Pojemnik II**  
Variables->Invisible: PLC\_PRG.z

Projekt dostępny na stronie przedmiotu jako "Przykład1b"

## Przykład 2. – wartości zmiennych i opis stanu procesu



$$p.I: \overline{z_1 z_2} = \overline{z_1} + \overline{z_2} = z_1 + z_2$$

$$p.II: \overline{z_1 z_2} = \overline{z_1} + \overline{z_2} = z_1 + \overline{z_2}$$

$$p.III: \overline{z_1}$$

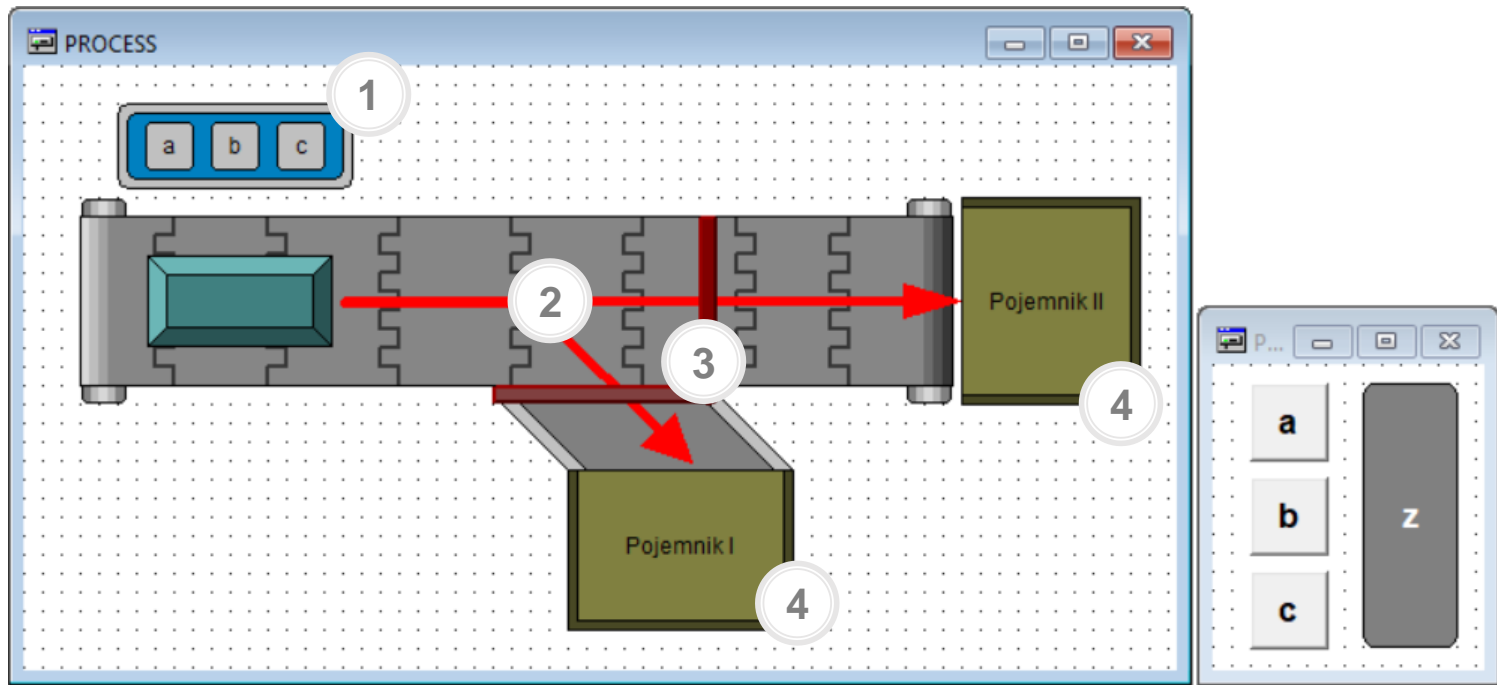
Trzy elementy Rectangle położone jeden nad drugim

### Konfiguracja:

- Rectangle #6:  
Text->Content: **Pojemnik I**  
Variables->Invisible: PLC\_PRG.z1 OR PLC\_PRG.z2
- Rectangle #7:  
Text->Content: **Pojemnik II**  
Variables->Invisible: PLC\_PRG.z1 OR NOT PLC\_PRG.z2
- Rectangle #8:  
Text->Content: **Pojemnik III**  
Variables->Invisible: NOT PLC\_PRG.z1

Projekt dostępny na stronie przedmiotu jako "Przykład2b"

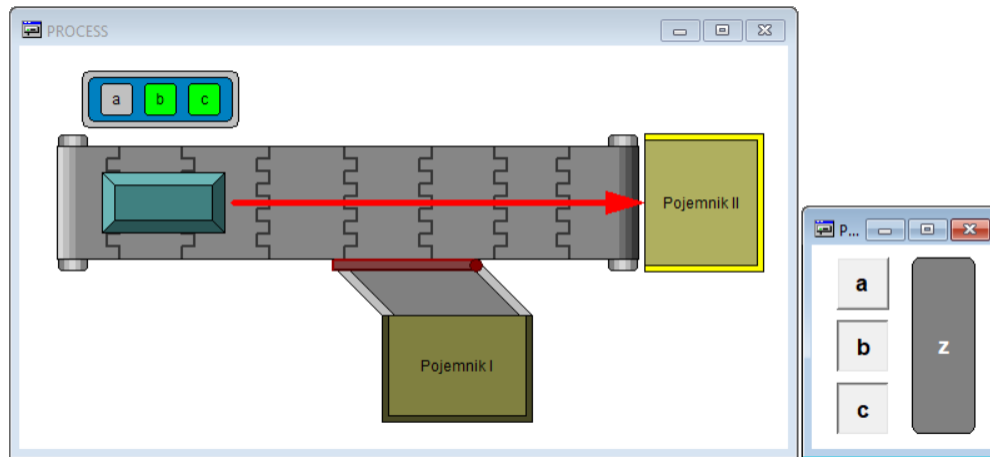
# Przykład 1. – wizualizacja procesu



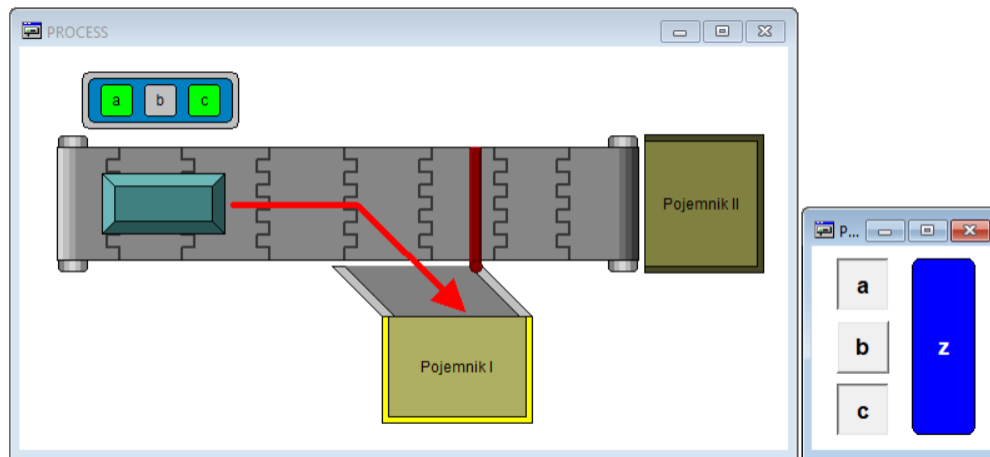
Aktywne elementy wizualizacji:

1. Kontrolki czujników: Rounded Rectangle
2. Strzałki: Polyline + Polygon
3. Zwrotnice: Rectangle + Polyline
4. Pojemniki: 2 x Rectangle

# Przykład 1. – wizualizacja procesu



Prawidłowe cechy "b" i "c",  
nieprawidłowa cech "a"  
Detal skierowany do pojemnika  
drugiego



Prawidłowe cechy "a" i "c",  
nieprawidłowa cech "b"  
Detal skierowany do pojemnika  
pierwszego

# Przykład 1. – wizualizacja procesu

---

## Konfiguracja elementów wizualizacji

### 1. Kontrolki czujników (Rectangle):

- Colors->Color **biały**, Alarm Color **zielony**,
- Variables->Change Color **odpowiednio** PLC\_PRG.a, PLC\_PRG.b, PLC\_PRG.c.

### 2. Strzałki (grupa Polyline+Polygon):

- kierunek do Pojemnika I Variables->Invisible: NOT PLC\_PRG.z,
- kierunek do Pojemnika II Variables->Invisible: PLC\_PRG.z.

### 3. Zwrotnica (Rectangle+Polyline):

- położenie poziome Variables->Invisible: PLC\_PRG.z,
- położenie pionowe Variables->Invisible: NOT PLC\_PRG.z.

### 4. Pojemniki (2 x Rectangle):

- Colors->Color (72,72,36), (128,128,64), Alarm Color (255,255,0), (175,175,97)
- Pojemnik I Variables->Invisible: PLC\_PRG.z,
- Pojemnik II Variables->Invisible: NOT PLC\_PRG.z.

*Projekt dostępny na stronie przedmiotu jako "Przykład1c"*