

CoDeSys 3 – programowanie w języku CFC

Spis treści

1.	Wstęp	2
1.1.	Wymagania programowe	2
2.	Tworzenie projektu i dodawanie programu w języku CFC	3
3.	Organizacja okien dla języka CFC	5
4.	Składnia języka CFC – Toolbox	6
5.	Łączenie elementów programu.	7
6.	Tworzenie połączeń bez linii.	7
7.	Zmiana kolejności wykonywania bloków.	8
8.	Poprawianie kolizji linia – blok.....	10
9.	Przykład.....	11



Powering Business Worldwide

1. Wstęp

Aplikacja CoDeSys 3 jest nowoczesnym środowiskiem służącym do programowania sterowników firmy Eaton Electric.

Notatka ta ma na celu przedstawienie programowania w języku CFC jednym z 7 języków zgodnych z normą IEC-61131-3. Podstawy programowania przedstawiono w notatkach NA03001PL oraz NA03002PL.

1.1. Wymagania programowe

Przedstawiana notatka opiera się na oprogramowaniu narzędziowym CoDeSys V3.5.7 (Build 3152).

Oprogramowanie przetestowane jest dla systemu Windows XP/7/8 (32,64bit), pracuje również poprawnie w Windows 10, ale nie było to szczegółowo testowane i użytkownik robi to na własną odpowiedzialność.

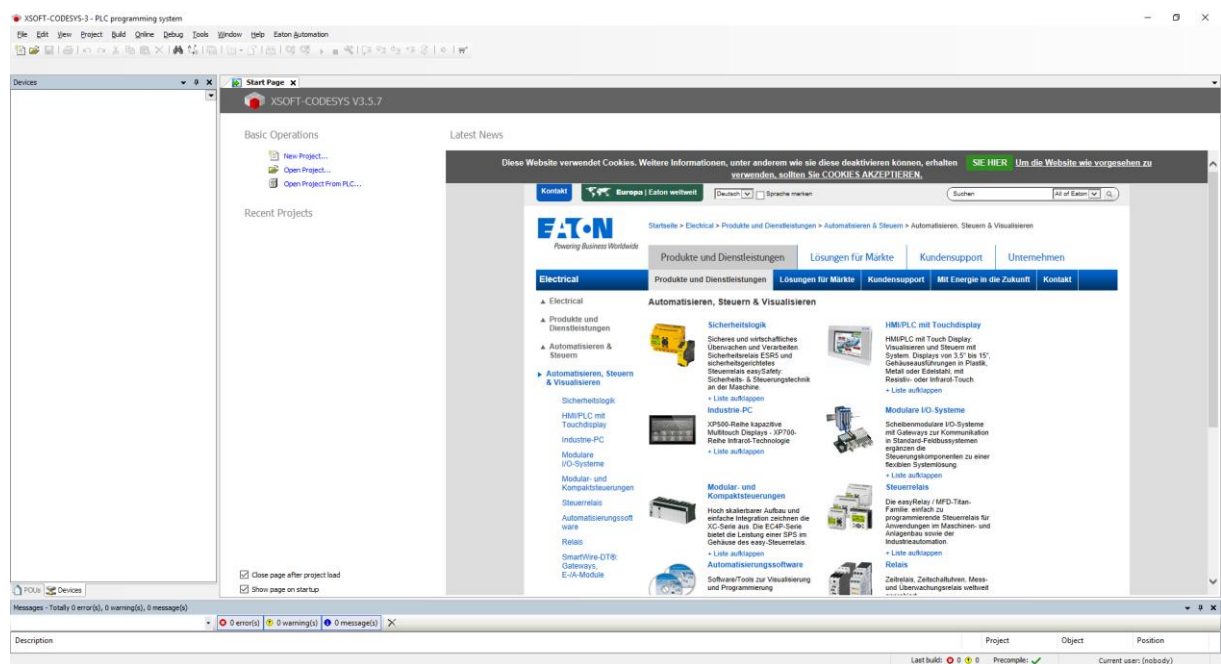
Aktualne wersje demonstracyjne oprogramowania Eaton można pobrać z Download Centre pod adresem:

<http://applications.eaton.eu/sdlc>

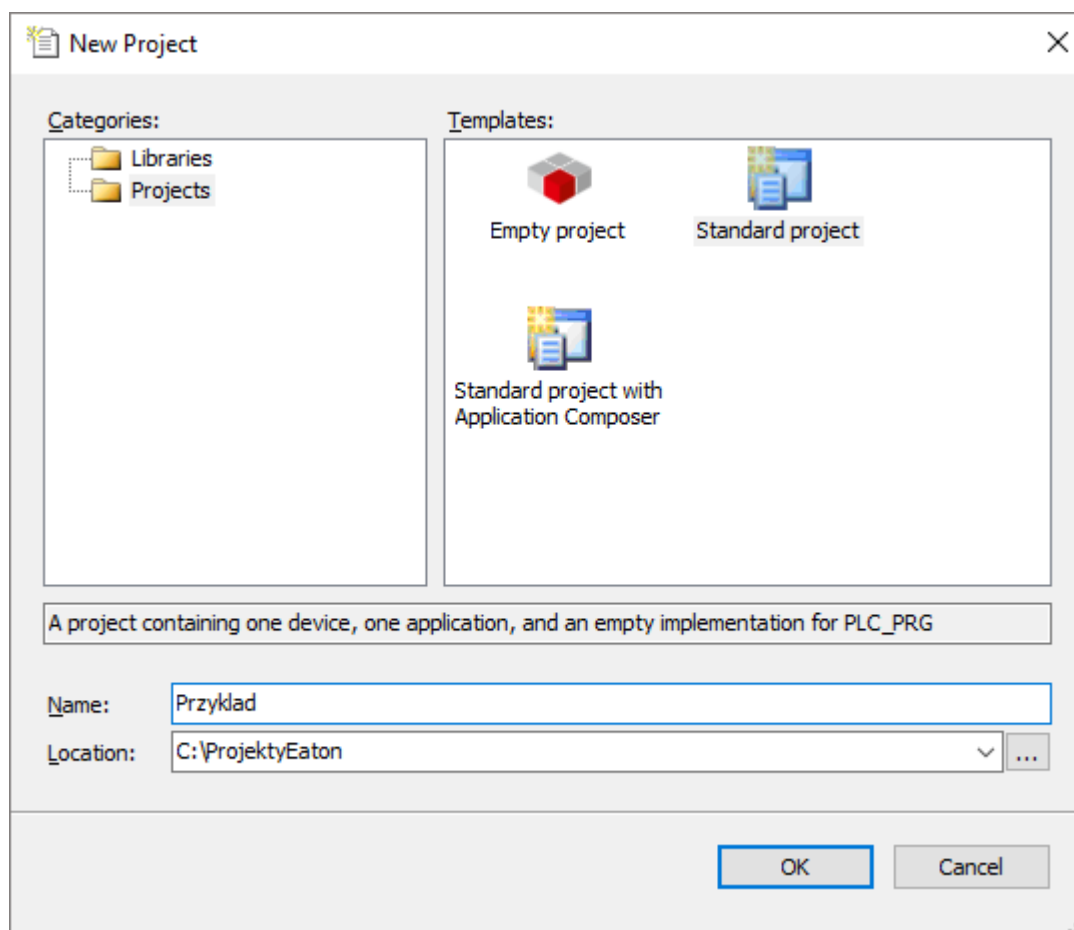
Uwaga. Z poziomu CoDeSys 3 nie ma możliwości zaprogramowania sterowników EC4P, XC-CPU101, XC-CPU201, XN-PLC, XC-CPU121. Jednostki te można programować jedynie z poziomu CoDeSys 2.

2. Tworzenie projektu i dodawanie programu w języku CFC

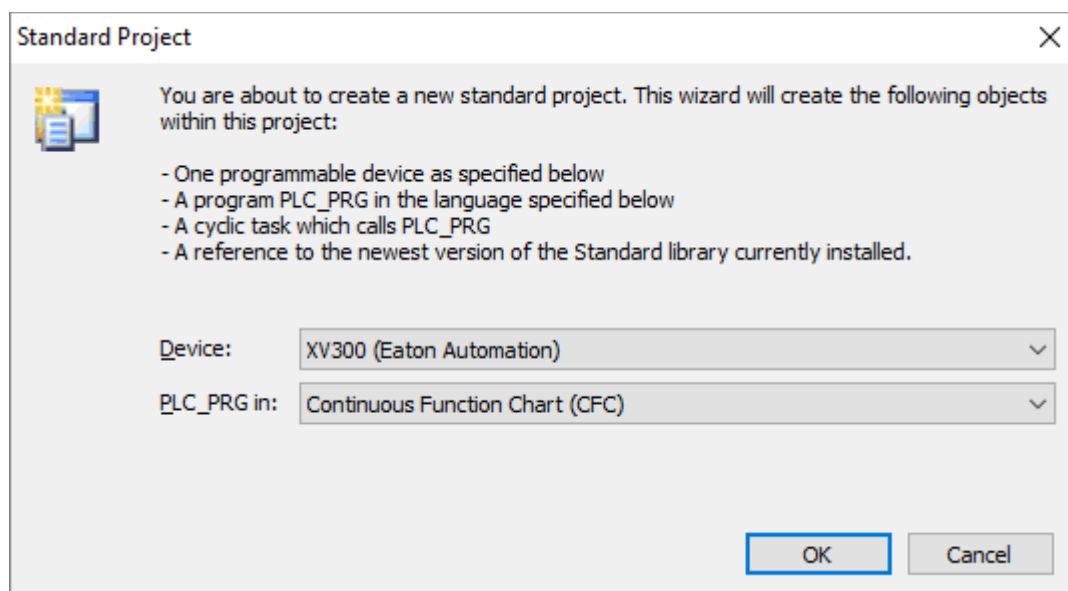
Uruchamiamy CoDeSys 3 i w oknie **Start Page** wybieramy opcję **New Project**:



W okienku **New Project** wybieramy **Standard project** – wtedy



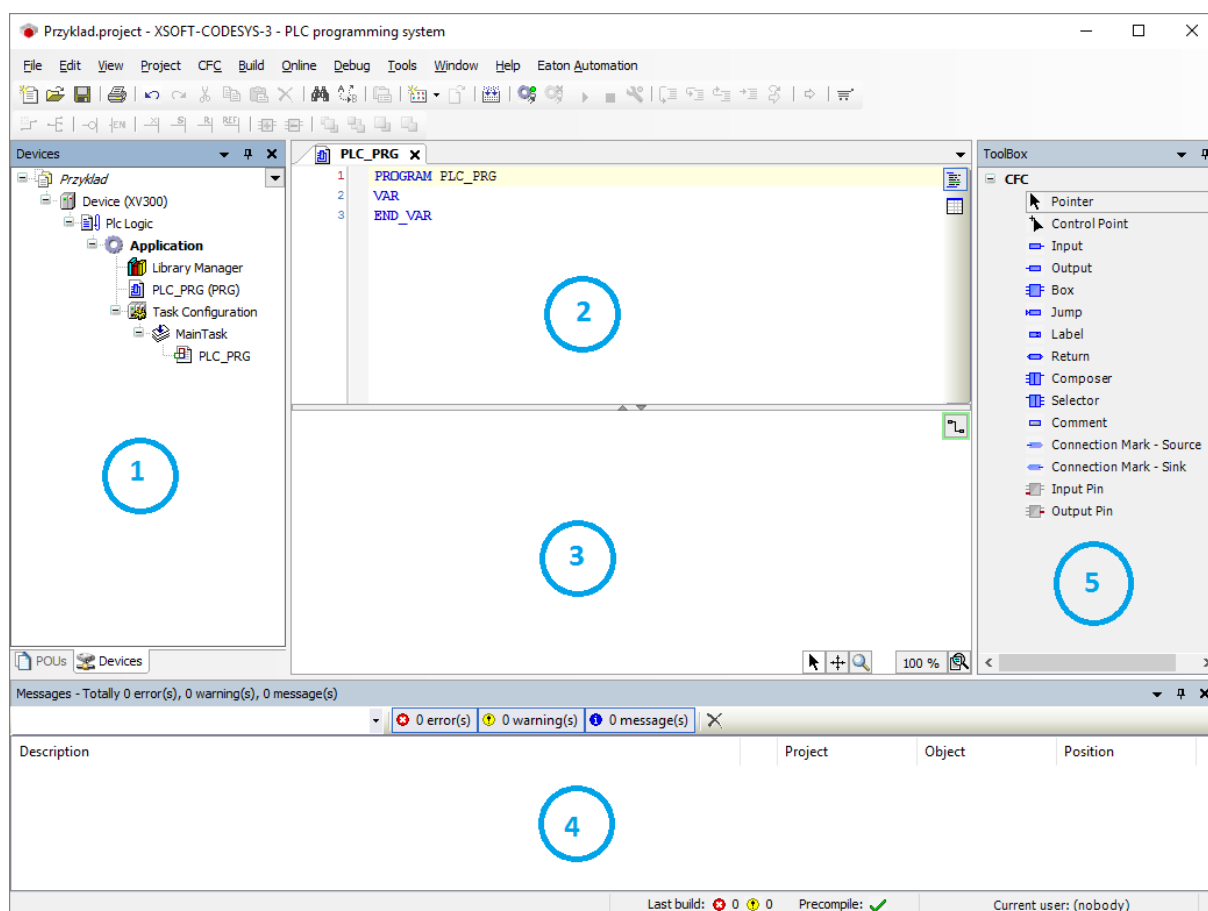
W kolejnym oknie **Standard Project** wybieramy typ sterownika oraz język programowania programu głównego. Mamy do dyspozycji 7 języków programowania zgodnych z IEC. W obrębie projektu programy mogą być napisane w różnych językach. Na potrzeby tej notatki wybieramy **XV300** oraz język najwyższego poziomu **CFC**.



3. Organizacja okien dla języka CFC















Po kliknięciu na program CFC np. **PLC_PRG** ekran wyświetla się jak poniżej. Jest podzielony na pięć części:

- 1 – Struktura projektu;
- 2 – Okno zmiennych lokalnych;
- 3 - Okno w którym piszemy program;
- 4 – Okno przebiegu kompilacji projektu;
- 5 – Toolbox - elementy programowania dla CFC.



4. Składnia języka CFC – Toolbox

Gdy wybrany jest język CFC w narzędziu toolbox znajdują się następujące elementy tego języka:

-  - dodawanie punktu kontrolnego na linii łączącej elementy. Umożliwia ręczne kształtowanie linii
-  - wejście programowe – wczytujemy zmienną i przesyłamy do podłączonego bloku.
-  - wyjście programowe – wynik operacji z podłączonego bloku przesyłamy do zmiennej.
-  - blok – wewnątrz wykonywane są operacje na zmiennych.
-  - jeżeli na wejściu tego elementu jest stan wysoki to następuje pominięcie kodu programu aż do miejsca w którym jest odpowiadający mu Label.
-  - miejsce do którego ma być wykonany skok od przypisanego elementu Jump
-  - jeżeli na wejściu tego elementu jest stan wysoki to na nim kończy się wykonywanie danego programu.
-  - przepisanie zmiennych do elementów struktury.
-  - przepisanie elementów struktury do zmiennych
-  - komentarz do programu
-  - wraz z Connection Mark – Sink tworzą połączenie równoznaczne z łączeniem linią.
-  - jak wyżej.
-  - dodawanie wejścia do bloku o ile ma on taką możliwość.
-  - dodawanie wyjścia do bloku o ile ma on taką możliwość.

5. Łączenie elementów programu.

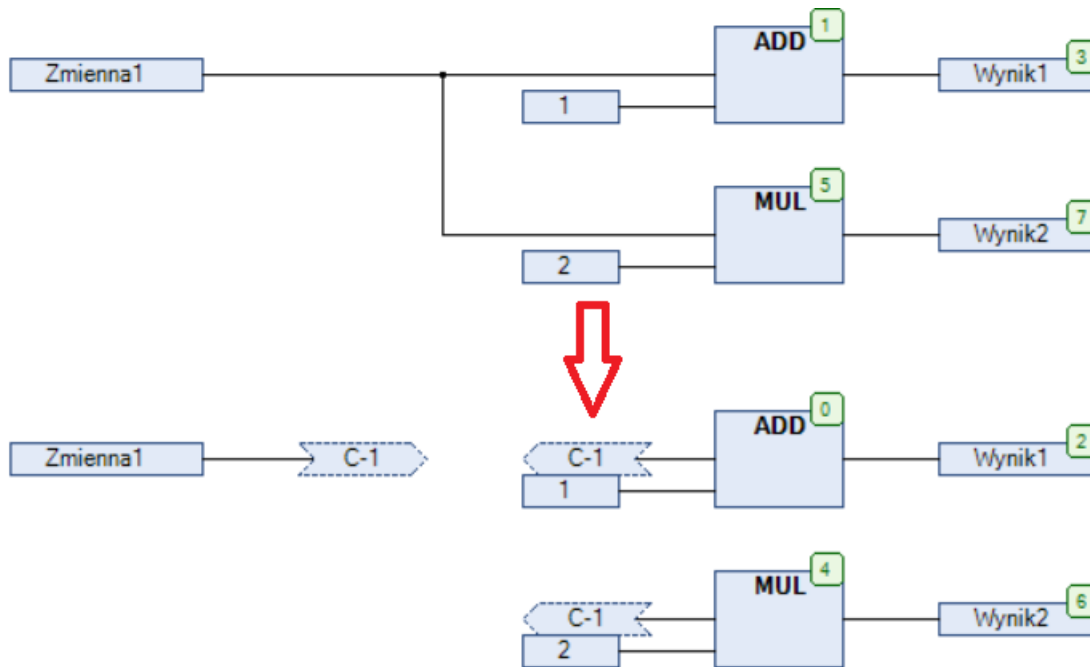
1. Przeciągamy element **Box** oraz **Output** do edytora.
2. Klikamy lewym klawiszem na wyjściu bloku **Box**. Wyjście zostaje podświetlone na czerwono.
3. Przeciągamy linię od wyjścia bloku **Box** do wejścia bloku **Output**. Gdy jesteśmy w odpowiedniej pozycji pojawi się strzałka kursora.
4. Puszczamy przycisk myszy. Elementy zostają połączone linią.



Możemy również zaznaczyć oba piny przytrzymując klawisz CTRL a następnie wybrać **Connect Selected Pins** w menu **CFC**.

6. Tworzenie połączeń bez linii.

1. Zaznaczamy linię pomiędzy dwoma elementami. Linia zostaje pogrubiona a punkty połączenia podświetlone na czerwono.
2. Wybieramy z menu CFC -> Connection Mark. Połączenie liniowe zastępowane jest znacznikami połączeń.
3. Klikamy Connection Mark Source i możemy zmienić nazwę.
4. Taką samą nazwę musimy wpisać do Connection Mark Sink.



7. Zmiana kolejności wykonywania bloków.

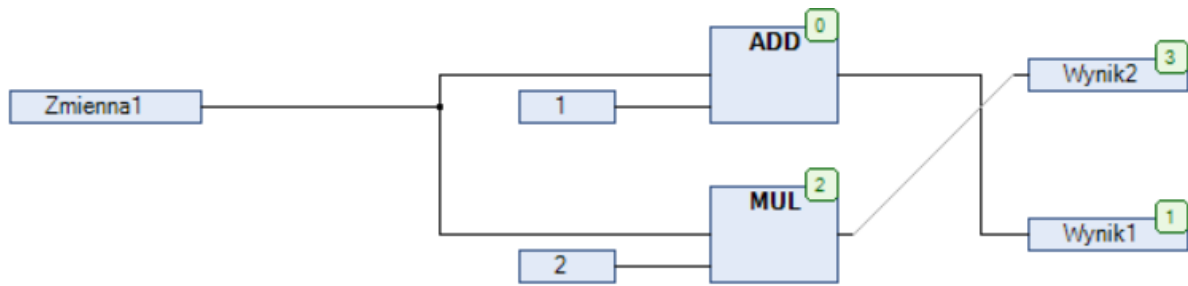
Program w języku CFC wykonywany jest zgodnie z numeracją elementów umieszczoną w ich prawym górnym narożniku od wartości najmniejszej do największej. Jeżeli wstawiamy nowe elementy to numer jest nadawany automatycznie zgodnie z kolejnością – od lewej do prawej i od góry do dołu. Jeżeli przesuwamy elementy wtedy numeracja nie jest aktualizowana. Numerację elementów możemy zmienić na kilka sposobów.

W celu zmiany numeru pojedynczego elementu klikamy na nim prawym klawiszem i wybieramy Execution Order a następnie:

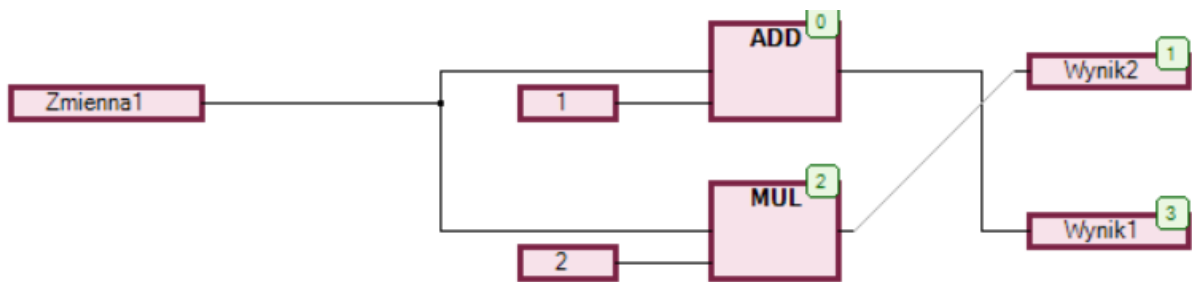
- Send to Front – otrzyma numer 0 i będzie wykonywany jako pierwszy
- Send to Back – otrzyma ostatni numer i będzie wykonywany jako ostatni
- Move Up - numer zmniejszy się o jeden
- Move Down – numer zwiększy się o jeden
- Set Execution Order – możemy wpisać wymagany numer

Jeżeli chcemy zaktualizować numerację całego programu to możemy wybrać

- Order by Data Flow – zostanie nadana chronologiczna numeracja dla wszystkich elementów zgodnie z kolejnością przepływu danych. Pozycja elementów nie zmieni się

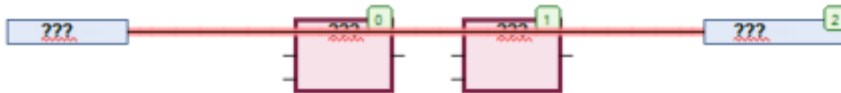


- Order by Topology – zostanie nadana chronologiczna numeracja zaznaczonych elementów zgodnie z ich położeniem

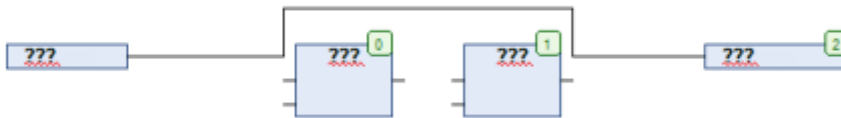


8. Poprawianie kolizji linia – blok

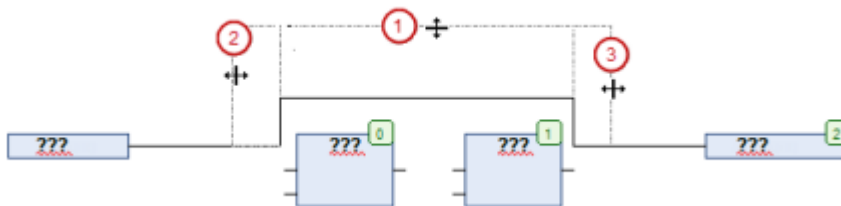
W przypadku gdy linia łącząca elementy przebiega pod innym elementem następuje kolizja i linia ta zostaje podświetlona na czerwono.



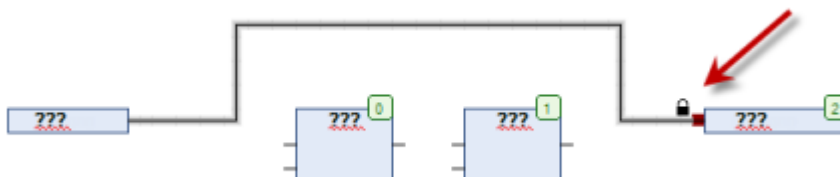
Możemy wtedy użyć funkcji automatycznego poprawiania nazywanej w Codesysie routowaniem. Możemy to zrobić klikając prawym przyciskiem na edytorze i wybierając Routing -> Route All Connections



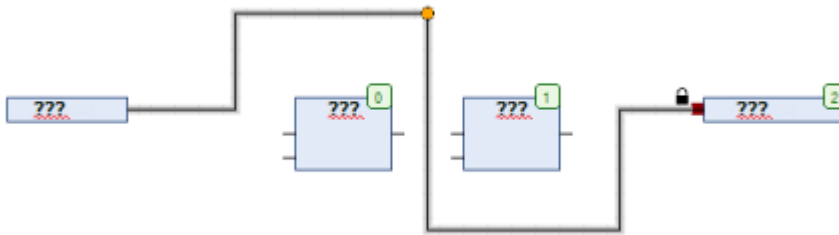
lub ręcznie prowadzić linię łączącą



tak zmieniona linia nie podlega już funkcji automatycznego routingu a sygnalizowane jest to znakiem kłódki



Możemy jeszcze dokładniej prowadzić linię dodając dowolną liczbę punktów kontrolnych. Element Control Point z toolboxa lub prawym przyciskiem myszy Routing -> Create Controlpoint



Linię możemy odblokować do automatycznego routingu klikając prawy a następnie Routing -> Unlock Connection lub w menu CFC

9. Przykład

Przedstawimy tutaj bardzo prosty przykład opóźnionego załączania wyjścia.

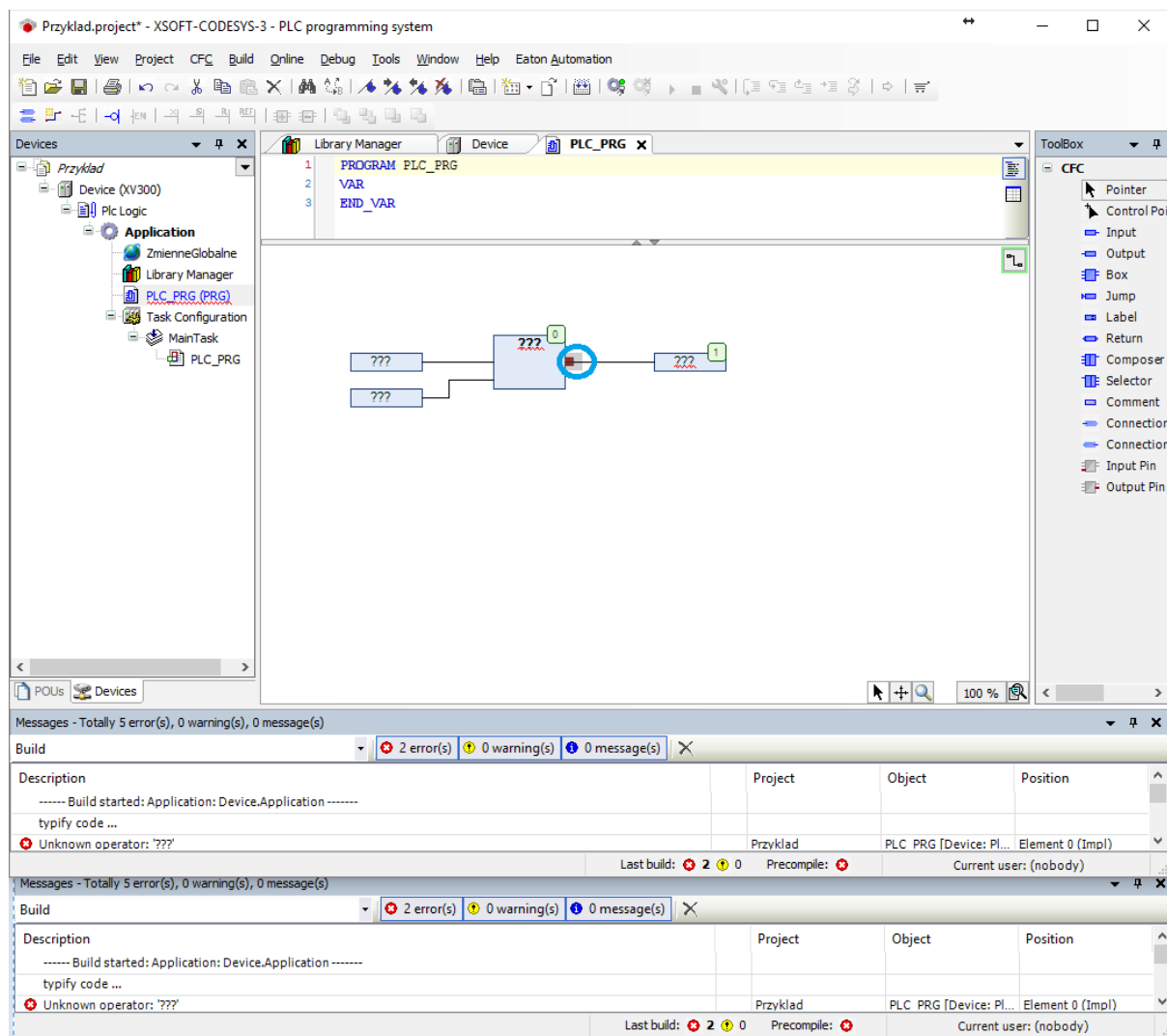
W tym celu do okna programowanie przeciągamy dwa Inputy, jeden Output i jeden Box

A screenshot of the CoDeSys 3 software interface. The main window shows a CFC (Control Flow Chart) program. The program code is:

```
1 PROGRAM PLC_PRG
2 VAR
3 END_VAR
```

 The CFC diagram shows two input boxes labeled '???' on the left, one output box labeled '???' in the middle, and one box labeled '???' on the right. Blue arrows point from the 'Input' and 'Output' icons in the toolbox to the corresponding boxes in the diagram. The toolbox on the right is titled 'CFC' and contains various icons like Pointer, Control Point, Input, Output, Box, Jump, Label, Return, Composer, Selector, Comment, Connector, Input Pin, and Output Pin. The bottom status bar shows 'Messages - Totally 5 error(s), 0 warning(s), 0 message(s)' and 'Build' information.

Łączymy je ze sobą klikając lewym klawiszem myszy na „ogonek”, nie puszczać najeżdżamy na „ogonek” wejściowy i wtedy puszczaemy. Można też w drugą stronę.



Deklarujemy zmienne korzystając z Automatycznej Deklaracji **Błąd! Nie można odnaleźć źródła odwołania.** i wybieramy blok korzystając z Asystenta Deklaracji **Błąd! Nie można odnaleźć źródła odwołania..**

The screenshot displays the CoDeSys 3 PLC programming system interface. The main window shows a variable declaration in the CFC editor:


```
1 PROGRAM PLC_PRG
2 VAR
3   Wejscie1 AT %IX0.0: BOOL;
4   CzasOpoznienia: TIME := t#5s;
5   Wyjscie1 AT %QX0.0: BOOL;
6   Czasowka1: TON;
7 END_VAR
```

The variable declaration is circled in blue. Below the code, a ladder logic diagram is shown, featuring a TON timer block labeled 'Czasowka1'. The 'IN' and 'PT' inputs are connected to 'Wejscie1' and 'CzasOpoznienia' respectively. The 'Q' output is connected to 'Wyjscie1'.

The bottom status bar shows the build results:

Description	Project	Object	Position
typify code ...			
Compile complete -- 0 errors, 0 warnings			

The 'Compile complete' message is circled in blue. The status bar also indicates 'Last build: 0 errors, 0 warnings, Precompile: ✓, Current user: (nobody)'.

Po poprawnej kompilacji (przycisk ) , program możemy wgrać do sterownika.