POLITECHNIKA RZESZOWSKA
im. IGNACEGO ŁUKASIEWICZA

Katedra
Informatyki i Automatyki
Politechniki Rzeszowskiej

Sterowanie procesami dyskretnymi

Instrukcja I

„CODESYS Symulacja”

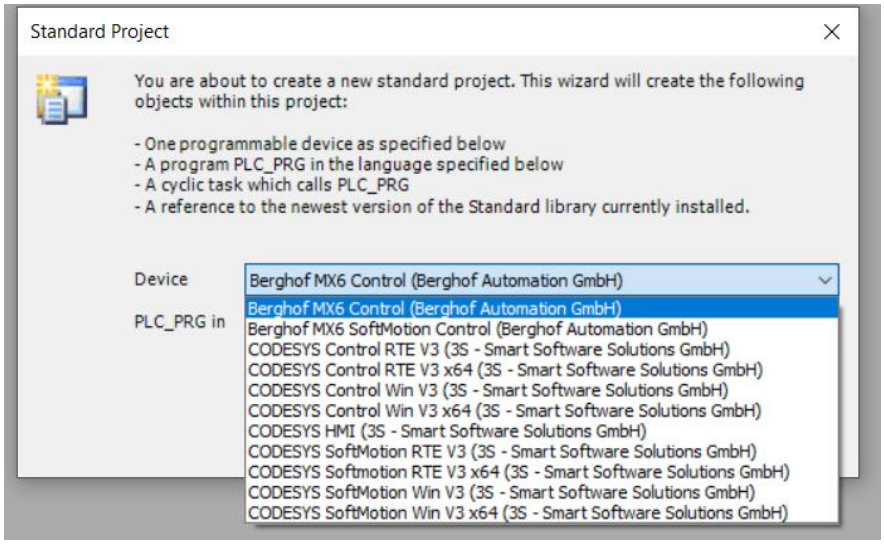
Cel ćwiczenia

Celem ćwiczenia jest zapoznanie studentów z podstawami programowania sterowników PLC w środowisku CODESYS. Studenci nauczą się tworzyć nowy projekt, dodawać elementy sterujące, programować w języku ST oraz tworzyć prostą wizualizację.

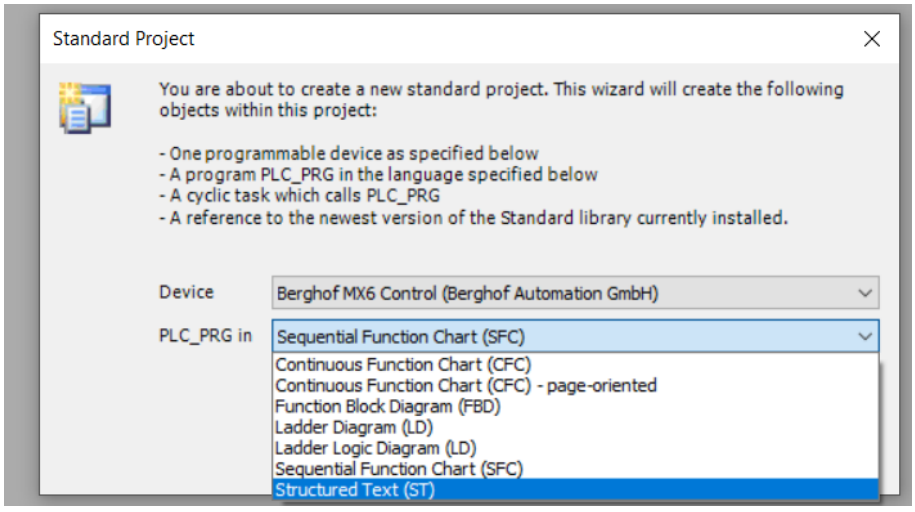
---

1. Tworzenie nowego projektu

1. Uruchamiamy program **CODESYS**.
2. Klikamy **„New Project”** (Nowy projekt).
3. Wybieramy **„Standard project”**.
4. Ustawiamy nazwę projektu oraz wybieramy lokalizację do zapisania pliku.
5. W oknie wyboru:
    - Wybieramy target: **Berghof MX6 Control**.

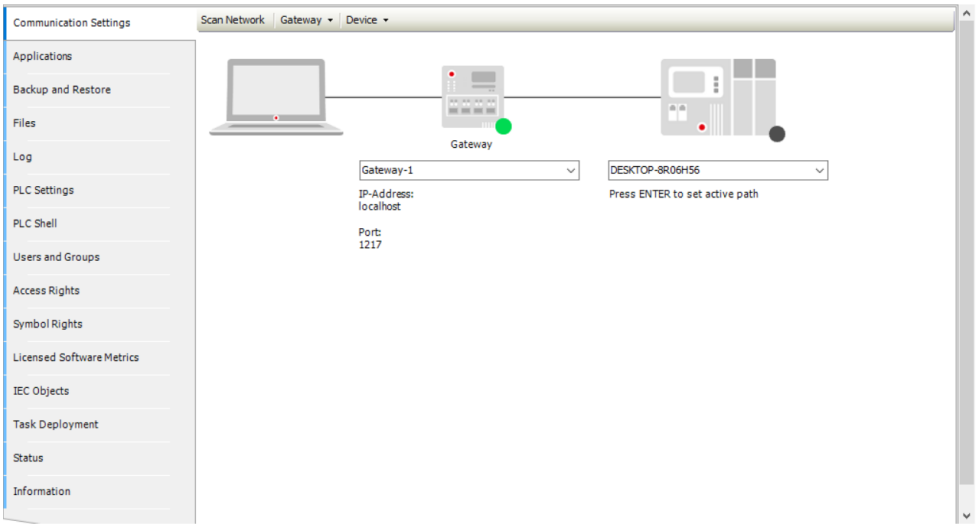


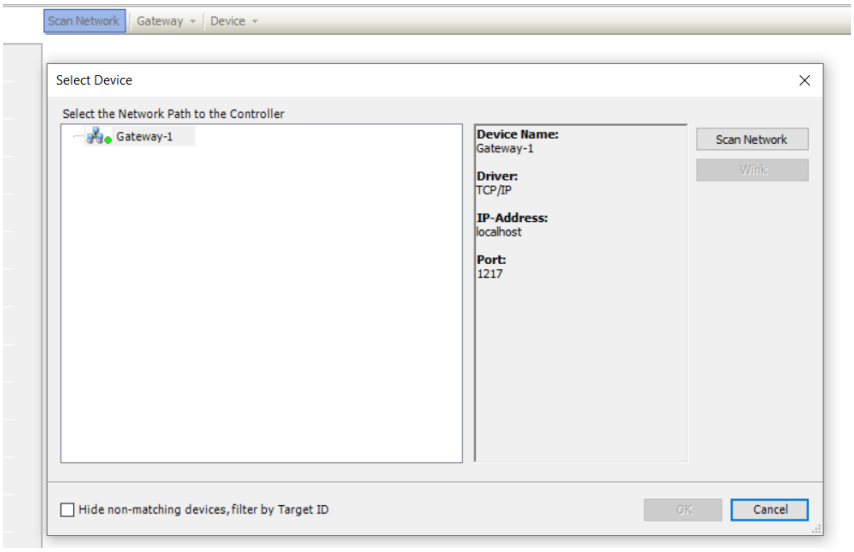
    - Wybieramy język programowania: **SFC (Sequential Function Chart)**.



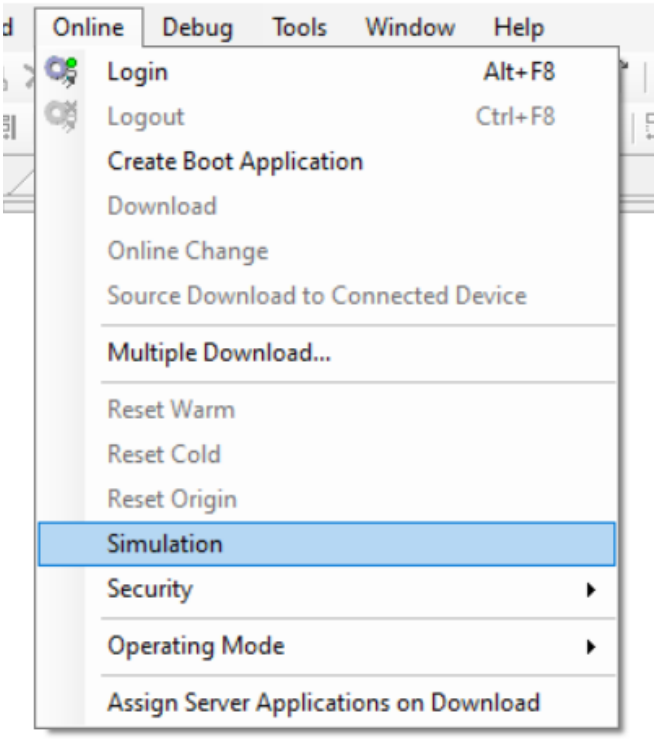
6. Po dokonaniu wyboru klikamy **„OK”**, co spowoduje utworzenie nowego projektu.

Uwaga: Jeśli pracujemy na rzeczywistym sterowniku PLC, należy kliknąć **„Device”**, wybrać **Gateway** oraz wyszukać sterownik, klikając **„Scan Network”**.





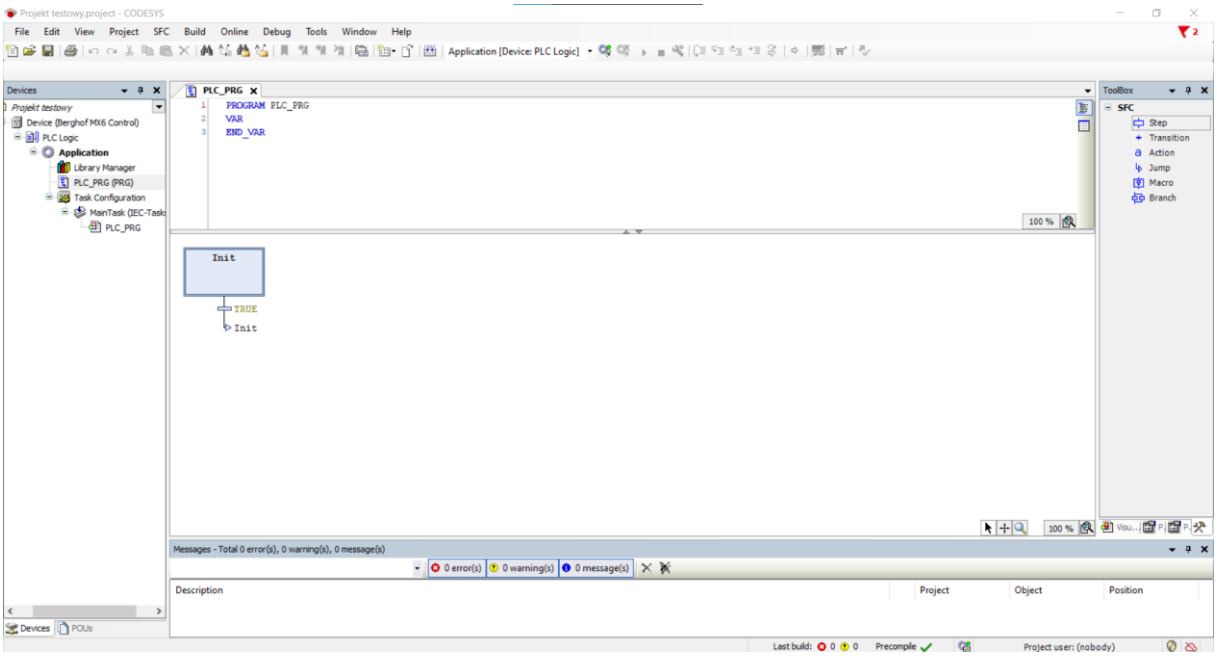
Jeśli korzystamy z symulacji, pomijamy ten krok i aktywujemy tryb symulacji: klikamy **„Online”**, a następnie zaznaczamy **„Simulation”**.



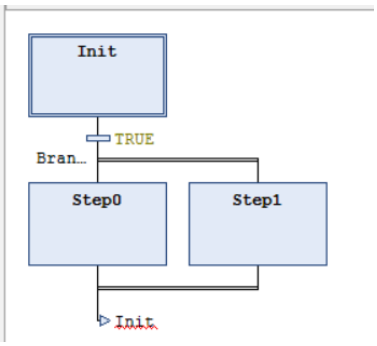
---

2. Dodawanie elementów sterowania w programie

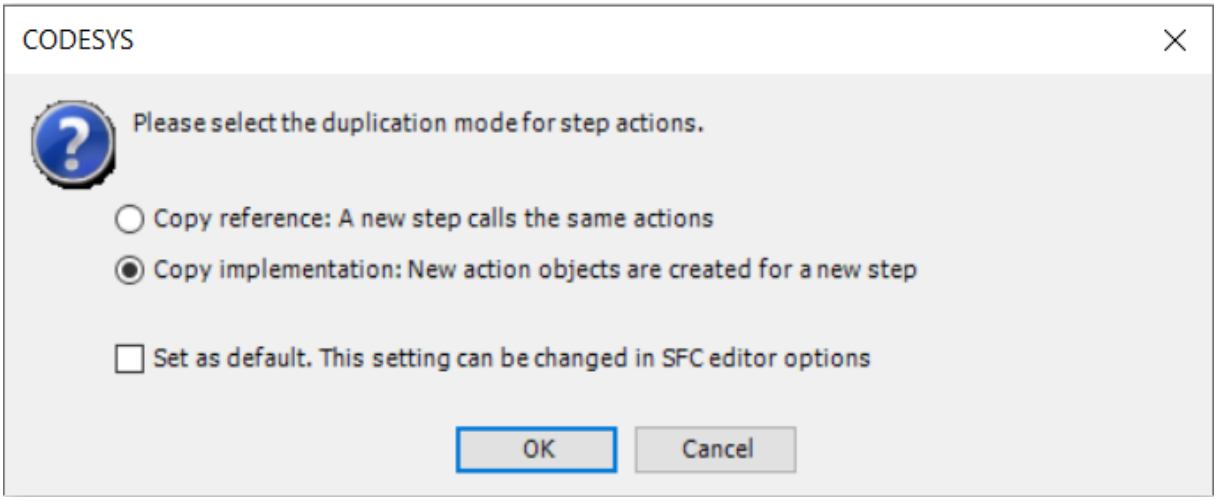
1. Po prawej stronie znajduje się panel **„Toolbox”**, skąd wybieramy elementy do programu.

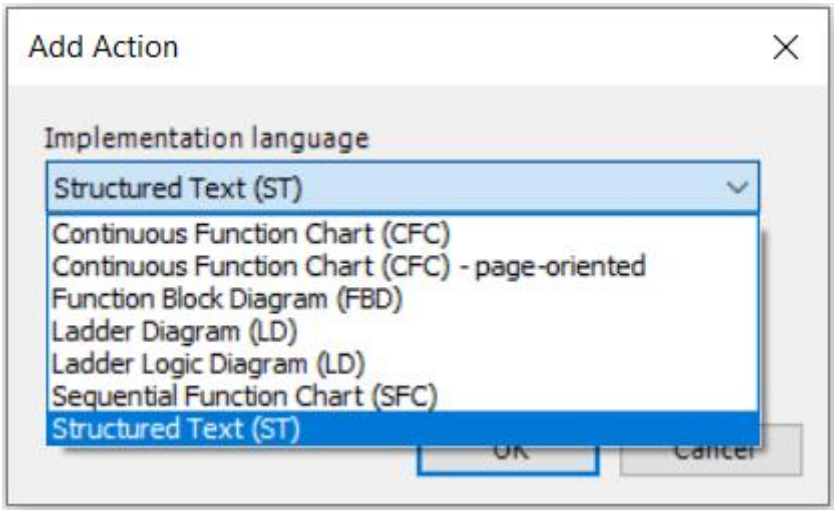


2. Dodajemy kolejne kroki do programu. Przykładowo tworząc graf jak na rysuku poniżej.



3. Po kliknięciu na dodany krok, wybieramy opcję **„Copy implementation”**, a następnie wybieramy język ST.

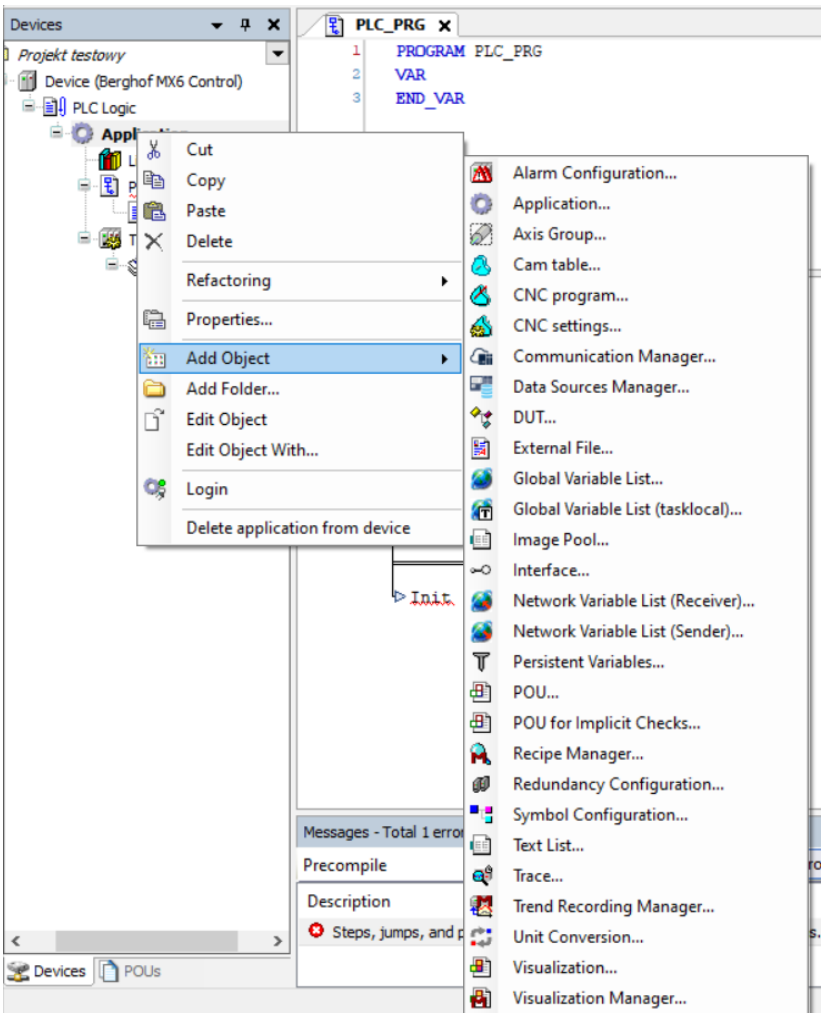




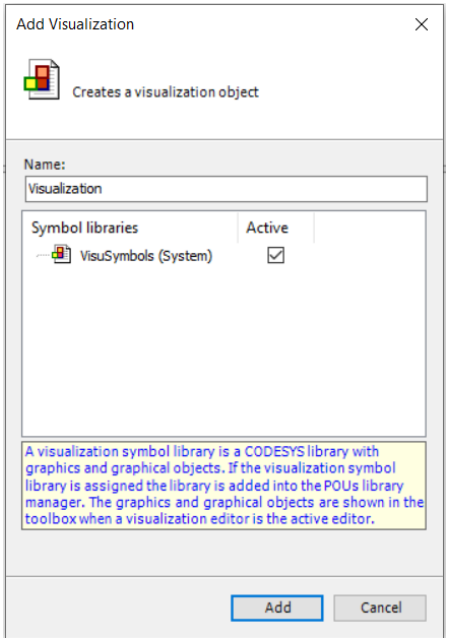
4. Powtarzamy proces dla kolejnych kroków.

3. Tworzenie wizualizacji

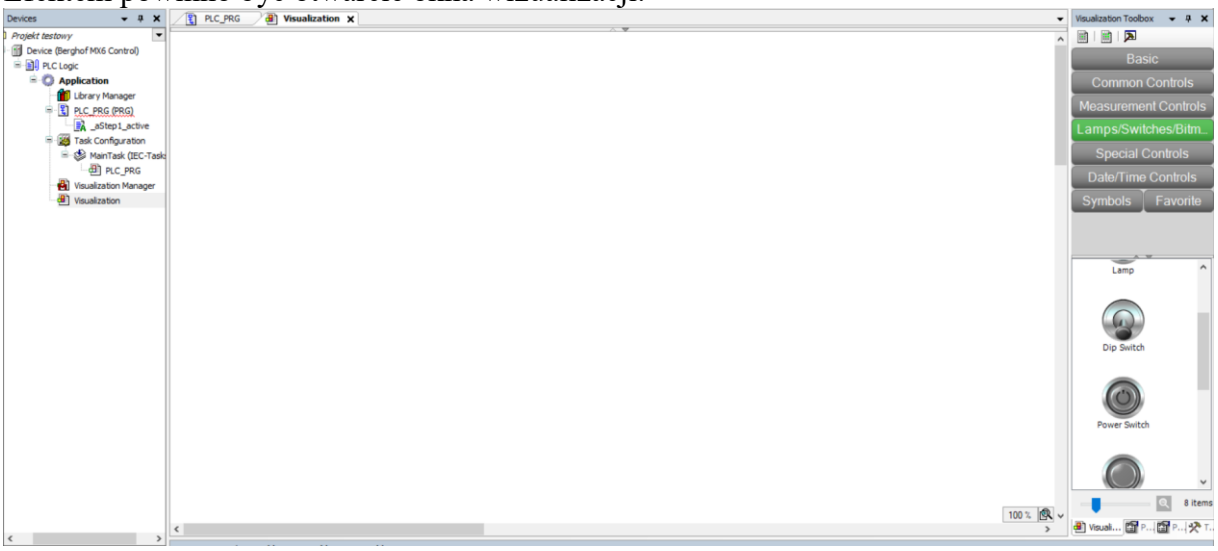
1. W drzewie projektu znajdujemy **„Application”**.
2. Klikamy na nią prawym przyciskiem myszy i wybieramy **„Add Object”**, a następnie **„Visualization”**.



3. W nowo otwartym oknie zaznaczamy **„VisuSymbols”** i klikamy **OK** – dodaje to obiekt wizualizacji do projektu.

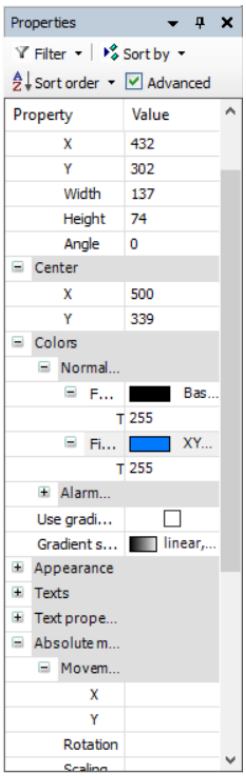


4. Efektem powinno być otwarcie okna wizualizacji.



5. Po prawej stronie pojawi się **„Visualization Toolbox”**, z którego wybieramy potrzebne elementy wizualizacji.

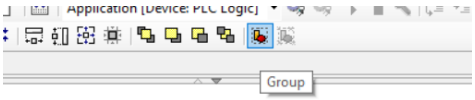
6. Aby dodać element do wizualizacji, przeciągamy go do okna projektu za pomocą myszy.
7. Po dodaniu elementu klikamy na niego, a po prawej stronie pojawi się okno parametrów, w którym możemy zmieniać właściwości (np. kolor, rozmiar).

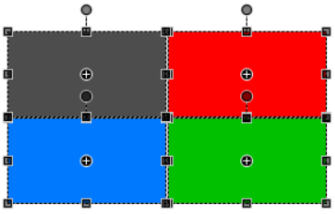


Grupowanie elementów

Jeśli chcemy, aby kilka elementów było traktowanych jako jedna całość:

1. Zaznaczamy wszystkie elementy.
2. Klikamy opcję **„Group”**.

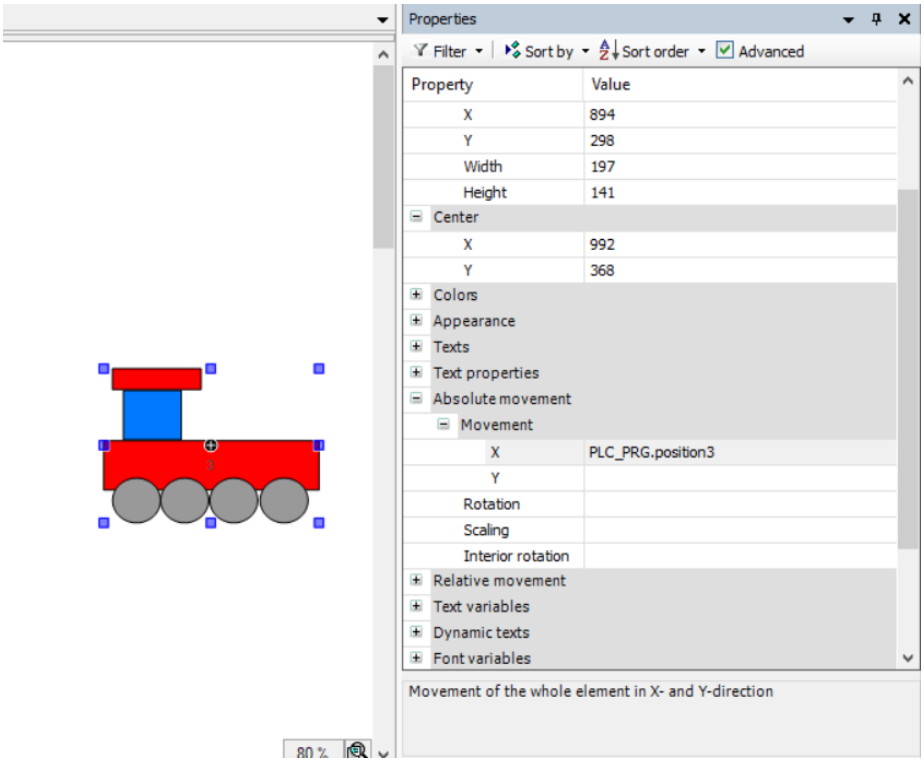




Animacja ruchu elementów

Aby elementy poruszały się w wizualizacji:

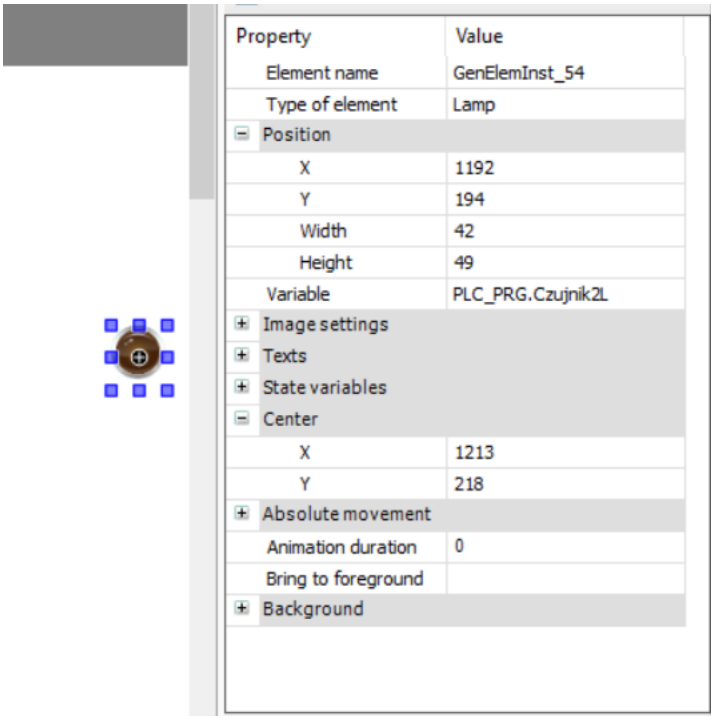
1. Przypisujemy do nich odpowiednie zmienne, np. **Movement, X**.
2. Wprowadzamy odpowiednie wartości sterujące ruchem.



Obsługa lampek kontrolnych

Aby elementy poruszały się w wizualizacji:

1. Do lampek należy przypisać odpowiednie zmienne.



---

4. Programowanie w języku ST

1. W oknie edytora kodu deklarujemy zmienne sterujące.

```
Wagonik1Vel: INT; // Prędkość wagonika 1
Wagonik2Vel: INT; // Prędkość wagonika 2
Wagonik3Vel: INT; // Prędkość wagonika 3
Czujnik1L, Czujnik1P: BOOL; // Czujniki krańcowe wagonika 1 (L - lewa, P - prawa)
Czujnik2L, Czujnik2P: BOOL; // Czujniki krańcowe wagonika 2
Czujnik3L, Czujnik3P: BOOL; // Czujniki krańcowe wagonika 3
position1,position2, position3 : REAL;
```

2. Należy w kodzie kroku 0 ustawić początkowe wartości zmiennych zdefiniowanych podczas budowy wizualizacji.

```
position1 := 992;
position2 := 992;
position3 := 992;

 Czujnik1P := TRUE;
 Czujnik2P := TRUE;
 Czujnik3P := TRUE;
 Czujnik1L := FALSE;
 Czujnik2L := FALSE;
 Czujnik3L := FALSE;
```

3. Przykładowe linijki kodu w języku ST ustawiające prędkość wagonu oraz zmieniające czujniki krańcowe w zależności od położenia wagonika

```
Wagonik1Vel:=1000;

position1 := position1 - 0.001*Wagonik1Vel;

IF position1 <= 12 THEN position1:= 12; czujnik1L:= TRUE;
END_IF
IF position1 <992 THEN czujnik1P := FALSE;
END_IF
```

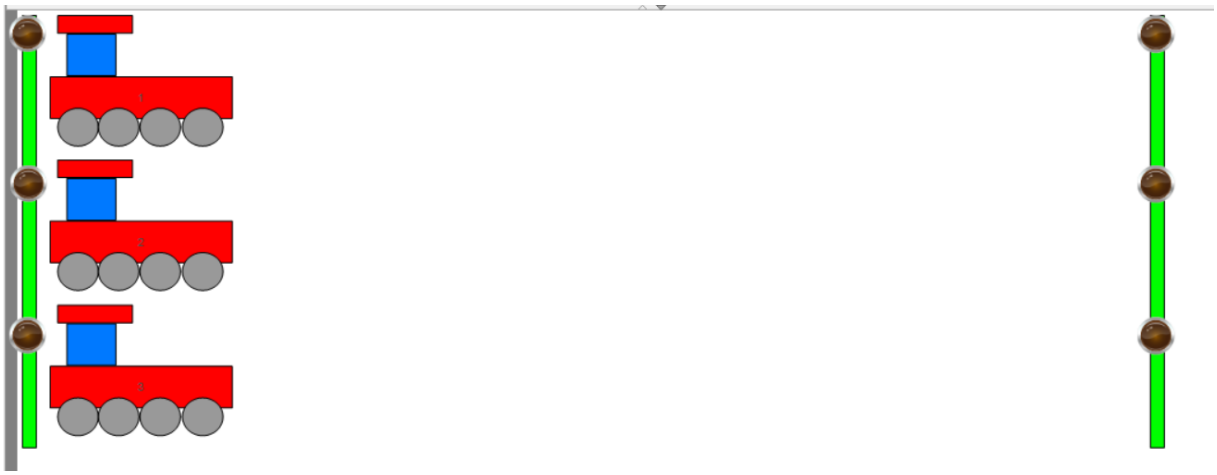
---

5. Testowanie symulacji

1. Klikamy **„Online” → „Login”**, aby połączyć się z symulatorem.
2. Uruchamiamy program za pomocą **„Start”**.
3. Obserwujemy działanie programu i wizualizacji.
4. W razie potrzeby wprowadzamy poprawki do kodu lub elementów wizualizacji.
5. Po zakończeniu testów klikamy **„Stop”**, a następnie **„Logout”**.

Zadania

1. Zadaniem zespołu studentów jest opracowanie wizualizacji zawierającej 3 wagoniki poruszające się w prostej linii wraz z kontrolkami sygnalizującymi osiągnięcie stanów skrajnych położenia wagoników. Przykładowy wygląd wizualizacji:



2. Należy utworzyć sterowanie, które z różnymi prędkościami wystartuje wagoniki z lewej strony do prawej, a następnie każdy z nich niezależnie po osiągnięciu skrajnej pozycji po prawej stronie wróci z tą samą prędkością na stronę lewą, gdzie poczeka aż wszystkie wagoniki znajdą się po lewej stronie. Wtedy program wykona się ponownie.