

POLITECHNIKA ŚWIĘTOKRZYSKA
Wydział Elektrotechniki, Automatyki i Informatyki
Katedra Elektrotechniki Przemysłowej i Automatyki

Sterowniki PLC

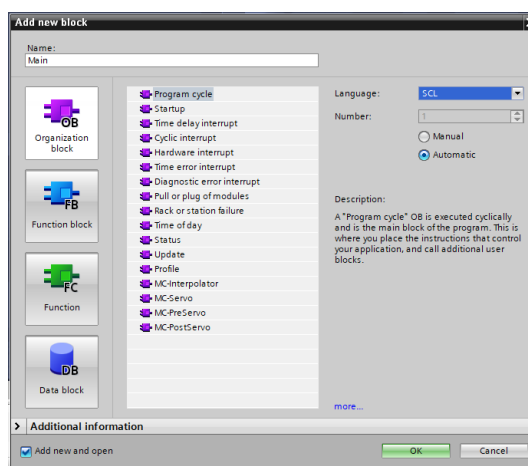
Programowanie sterowników S7-1200 w języku SCL

Instrukcja laboratoryjna
(wersja robocza)

Paweł Strączyński

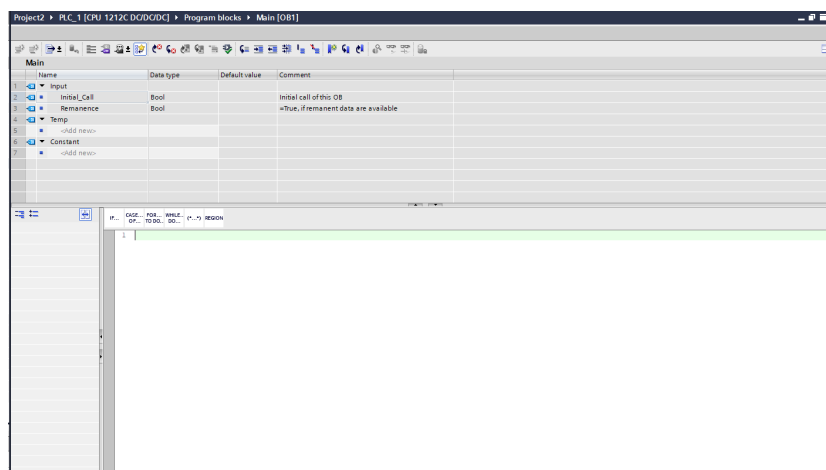
1 Język S7-SCL - konfiguracja języka w TIA Portal

Język SCL (*ang. Structured Control Language*) jest językiem tekstu strukturalnego (ST) wysokiego poziomu bazującym na Pascalu. SCL jest zgodny z normą IEC 61131-3 dotyczącą sposobów programowania sterowników PLC. Jako język wysokiego poziomu posiada typowe elementy języka takie jak możliwość definiowania zmiennych, instrukcje warunkowe, pętle, operacje porównania, operacje bitowe itd. Na rysunku 1.1 przedstawiono sposób wyboru języka SCL dla bloku organizacyjnego.



Rysunek 1.1: Konfiguracja języka SCL dla bloku programu OB1

Rysunek 1.2 przedstawia okno edytora programu w języku SCL w środowisku TIA Portal.



Rysunek 1.2: Okno edytora programu w języku SCL

2 Podstawowe instrukcje i operacje bitowe w języku SCL

Podstawowe operatory binarne w języku SCL przedstawiono na rysunku 2.1. Kolejno na rysunku przedstawiono operacje:

1. Przepisania wartości **We1** (I0.0) na wyjście **Q0.0** - **Wy1**,
2. Przepisania wyniku operacji iloczynu logicznego **We1** · **We1** na wyjście **Wy2**,
3. Przepisania wyniku operacji sumy logicznej **We1** + **We1** na wyjście **Wy3**,
4. Przepisania na wyjście **Wy4** negacji wejścia **We1**,
5. Przepisania wyniku operacji alternatywy wykluczającej **We1** oraz **We1** na **Wy5**,

```

1  "Wy1" := "We1";
2  "Wy2" := "We1" AND "We2";
3  "Wy3" := "We1" OR "We2";
4  "Wy4" := NOT "We1";
5  "Wy5" := "We1" XOR "We2";

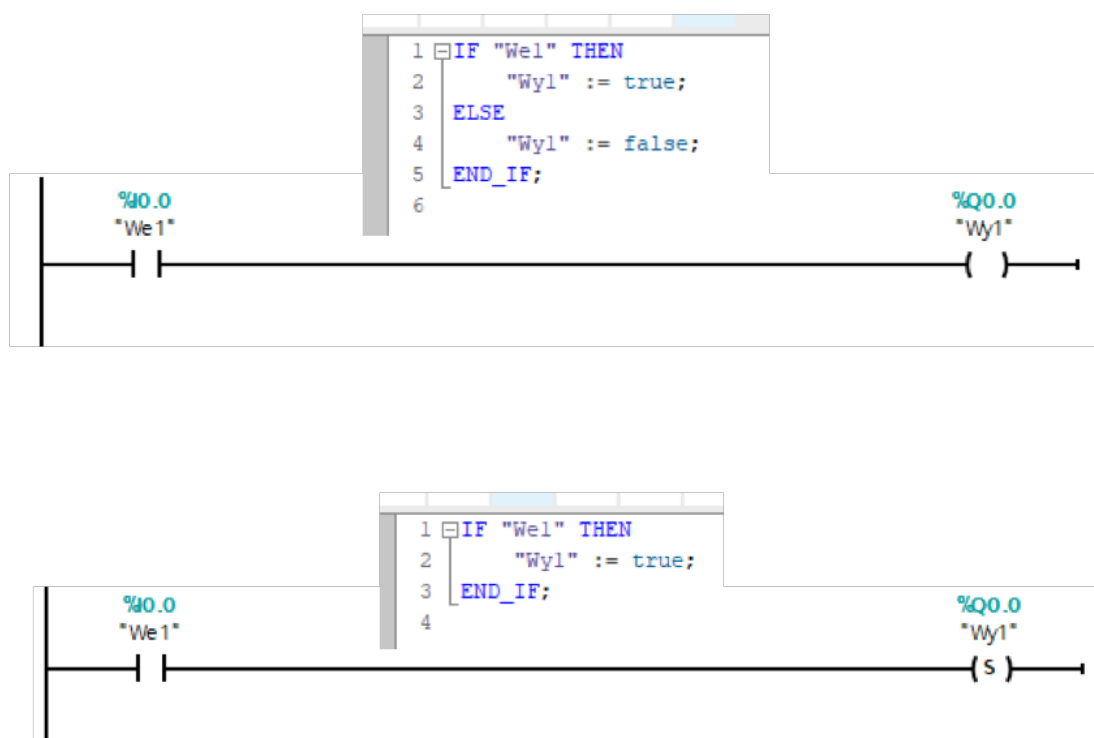
```

Rysunek 2.1: Podstawowe operacje bitowe w języku SCL

Symbol	Typ	Adres	Wartość	Wartość	Wartość	Wartość
We1	Bool	%I0.0	☐	☒	☒	☒
Wy1	Bool	%Q0.0	☐	☒	☒	☒
We2	Bool	%I0.1	☐	☒	☒	☒
Wy2	Bool	%Q0.1	☐	☒	☒	☒
Wy3	Bool	%Q0.2	☐	☒	☒	☒
Wy4	Bool	%Q0.3	☐	☒	☒	☒
Wy5	Bool	%Q0.4	☐	☒	☒	☒

Rysunek 2.2: Tablica tagów

Przepisanie wartości możliwe jest również z wykorzystaniem instrukcji warunkowej jak na rysunku 2.3



Rysunek 2.3: Przepisanie wartości z wykorzystaniem instrukcji warunkowej

3 Detekcja zbocza sygnału

W języku SCL podobnie jak w języku drabinkowym istnieje możliwość detekcji zbocza sygnału. W bibliotece funkcji dostępne są funkcje `R_TRIG` oraz `F_TRIG` odpowiedzialne za detekcję zbocza narastającego (*rising*) oraz opadającego (*faling*). Funkcje te powiązane są z blokiem danych, gdyż aby detekcja zbocza była możliwa musi być pamiętany poprzedni stan sygnału.

```

1 IF "F_TRIG_DB" (CLK:="We1",
2   [           Q=>"Var1");
3 IF "IEC_Timer_0_DB_1".TP(IN := "Var1",
4   [           PT := T#300ms,
5   [           Q => "Wy2");
6

```

Rysunek 3.1: Przykład użycia instrukcji `F_TRIG`

4 Timery i liczniki

W języku SCL znaleźć można wszystkie rodzaje układów do odmierzania czasu i liczniki jakie dostępne są w języku drabinkowym (LAD). Instrukcje timerów dostępne są w zakładce **Instrukction** – > **Basic Instructions** – > **Timer operations** jak na rysunku 4.1. Instrukcje liczników w zakładce **Counter operations**. Timer oraz Licznik można dodać do

▼ Timer operations		V1.0
TP	Generate pulse	V1.0
TON	Generate on-delay	V1.0
TOF	Generate off-delay	V1.0
TONR	Time accumulator	V1.0
RESET_TIMER	Reset timer	V1.0*
PRESET_TIMER	Load time duration	V1.0*
▼ Counter operations		V1.0

Rysunek 4.1: Instrukcje timerów

programu metodą przeciągnij i upuść - tak samo każdy inny blok programu. Przykład użycia timera przedstawiono na rysunku 4.2.

```

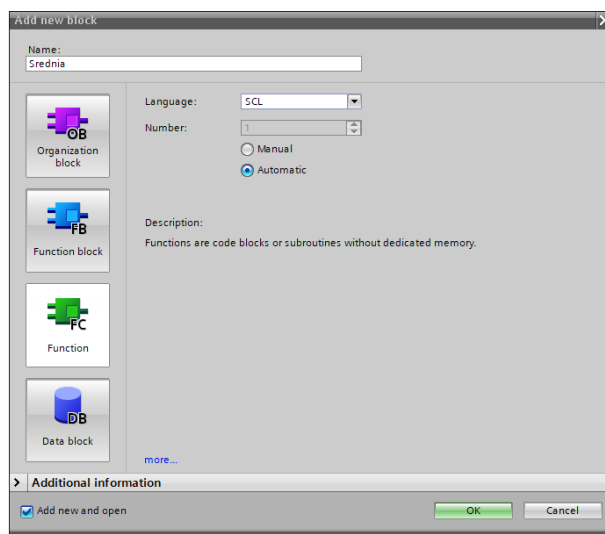
1  ▢ "IEC_Timer_0_DB".TON(IN:="We1",
2    PT:=T#2s,
3    Q=>"Wyl",
4    ET=>"Czas");
5
6  ▢ IF "IEC_Timer_0_DB".ET >= T#1s THEN
7    "Wy2" := true;
8  ELSE
9    "Wy2" := false;
10 END_IF;
11 ▢ IF "IEC_Timer_0_DB".Q THEN
12   RESET_TIMER("IEC_Timer_0_DB");
13 END_IF;
..

```

Rysunek 4.2: Generator fali prostokątnej w język SCL

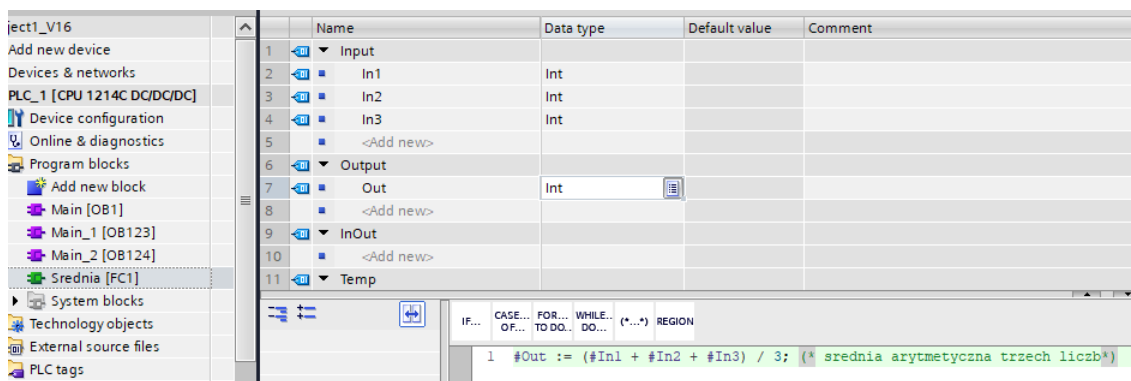
5 Funkcje i bloki funkcyjne

Język SCL podobnie jak LAD i FBD umożliwia tworzenie funkcji i bloków funkcyjnych (funkcji zawierających blok danych). Funkcje i bloki funkcyjne pisane w języku SCL mogą być wywoływane w programie głównym pisanym zarówno w SCL-u jak i języku drabinkowym czy języku bloków funkcyjnych. Na rysunku 5.1 przedstawiono sposób tworzenia nowej funkcji.



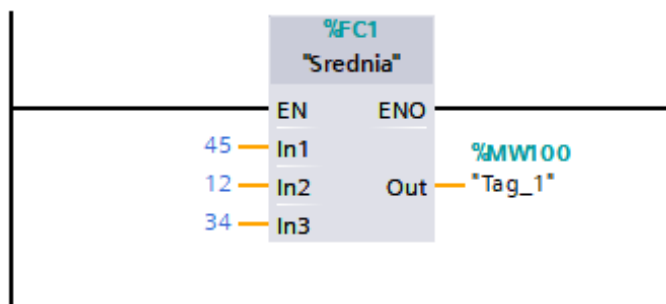
Rysunek 5.1: Tworzenie funkcji

Rysunek 5.2 przedstawia implementację funkcji wyznaczającą średnią arytmetyczną trzech liczb.



Rysunek 5.2: Funkcja w języku SCL wyznaczająca średnią arytmetyczną

Funkcja taka może zostać wywołana w programie w języku drabinkowym - rysunek 5.3.



Rysunek 5.3: Użycie funkcji w stworzonej w języku SCL

6 Zadania do wykonania

Zadanie 1 Napisać program w języku SCL realizujący następującą operację:

$$Q0.0 = I0.0 + I0.1 \cdot \overline{I0.3} \quad (1)$$

$$Q0.1 = \overline{I0.1 + I0.2} \quad (2)$$

Zadanie 2 Zrealizuj dekodery naturalnego kodu binarnego na kod 1 z 4 w języku SCL.

Zadanie 3 Zrealizuj układ sterowania oświetleniem klatki schodowej. Po wciśnięciu przycisku oświetlenie świeci przez 5s. Każde kolejne wciśnięcie powoduje wydłużenie czasu świecenia o kolejne 5s. Program napisać w języku SCL.

Zadanie 4 Utworzyć funkcję w języku SCL która wyznaczy wartość maksymalną i minimalną z pięciu liczb. Funkcja powinna mieć 5 wejść i dwa wyjścia. Program główny utworzyć w języku LAD. Zaprezentować działanie funkcji.