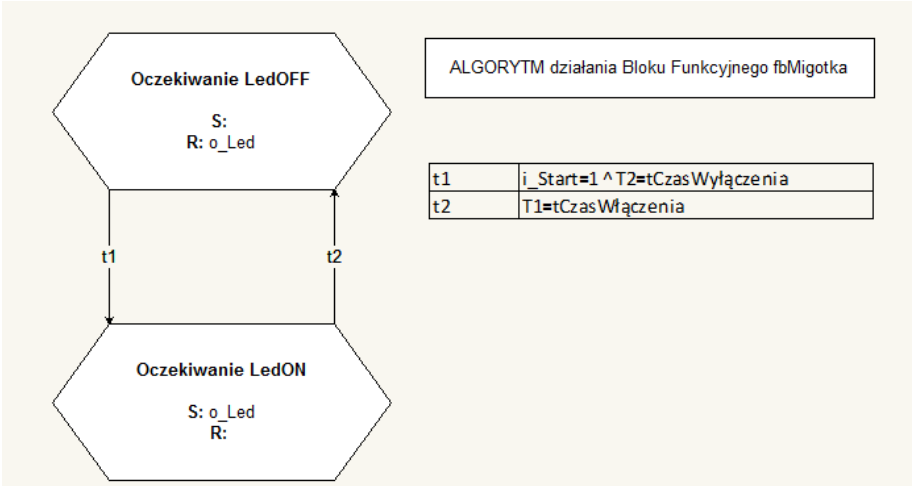
GX Works3 – Bloki funkcyjne i struktury typów danych.

Wprowadzenie:

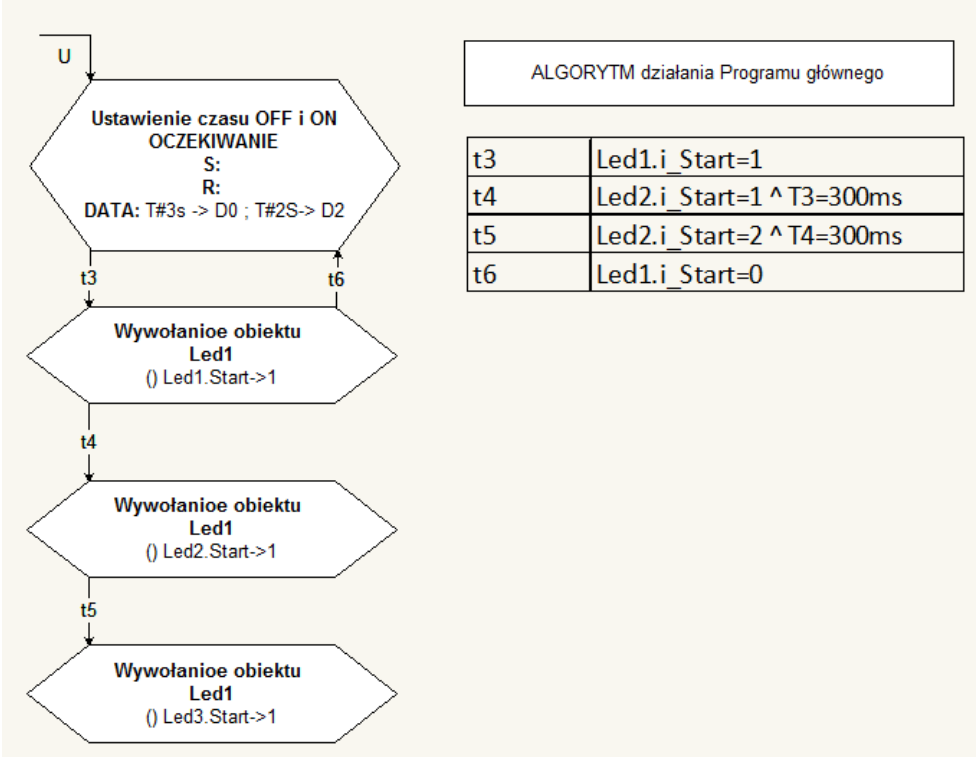
Lekcja oparta będzie o konkretne zadanie: Do zaprojektowania mamy system składający się z 3 lampek Led1, Led2, Led3, które mają migać w konkretnej sekwencji. Wszystkie lampki będą miały ustawione takie same czasy włączona 3s, wyłączona 2s co będzie stanowiło sekwencję migania. Każda lampka będzie posiadać wejście uruchamiające Start (1 – miga, 0 - zgaszona). System startował będzie po uruchomieniu pierwszej lampki z opóźnieniem 300ms po między startem kolejnej lampki.

Algorytmy do projektu:



ALGORYTM działania Bloku Funkcyjnego fbMigotka

t1	i_Start=1 ^ T2=tCzasWyłączenia
t2	T1=tCzasWłączenia

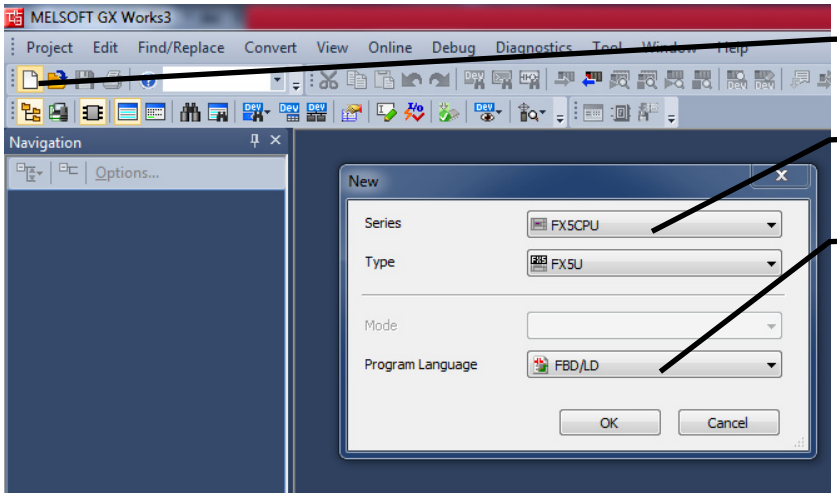


ALGORYTM działania Programu głównego

t3	Led1.i_Start=1
t4	Led2.i_Start=1 ^ T3=300ms
t5	Led2.i_Start=2 ^ T4=300ms
t6	Led1.i_Start=0

Implementacja:

Uruchom GX Works3 i stwórz nowy projekt.

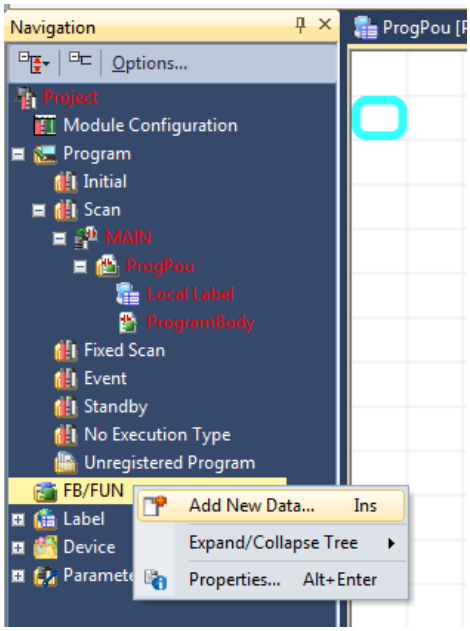


Utwórz nowy projekt

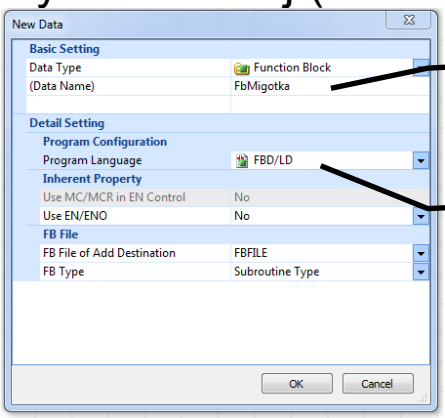
Wybierz sterownik

Wybierz język programowania

Aby pracować z **BF**(blokami funkcyjnymi) musimy stworzyć nowy blok funkcyjny, odpowiedzialny będzie za sposób działania jednej lampki. Mieścił będzie program i zmienne do obsługi wzorcowej lampki (**klasę**) z której następnie będziemy budowali (**obiekty**) Led1, Led2,…..LedX.



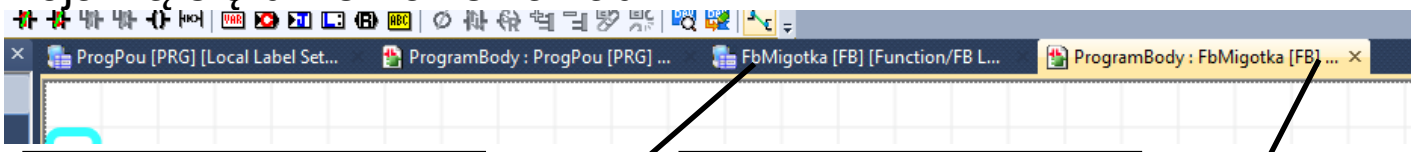
W drzewie projektu wybierz FB/FUN (Bloki Funkcyjne/ Funkcje) -> PPM(prawym przyciskiem myszki) z menu kontekstowego wybierz dodaj (Add New Data..)



Wpisz nazwę bloku: FbMigotka (nazwa może być dowolna)

Wybierz język w jakim będziesz programował nowy blok.

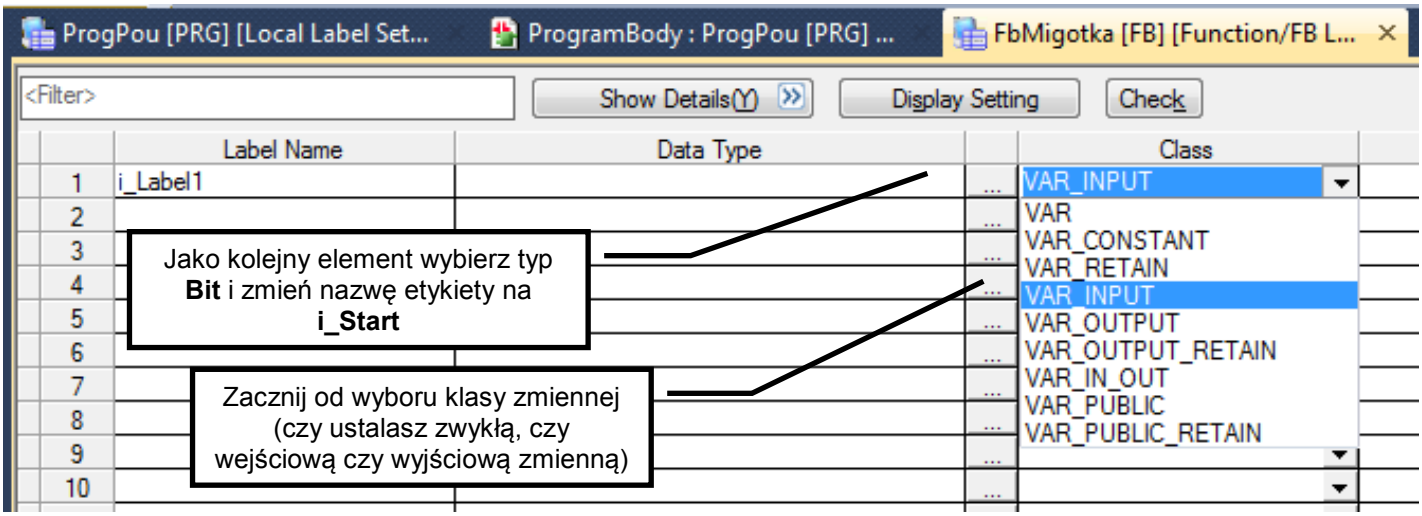
Pojawią się dwie nowe zakładki :



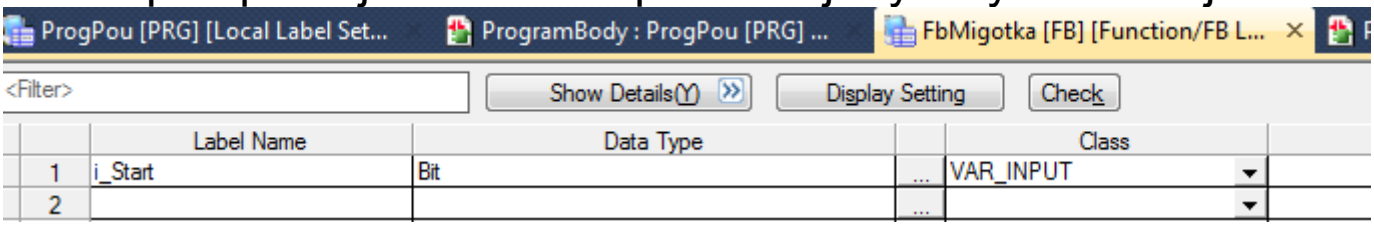
Tablica etykiet zmiennych lokalnych dla nowego bloku

Obszar programu głównego nowego bloku

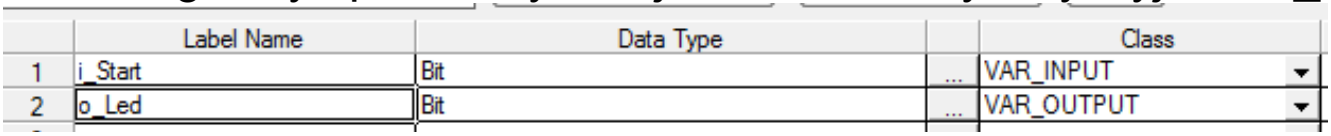
Zaczniemy od ustalenia nazw zmiennych lokalnych dla naszego nowego bloku funkcyjnego FbMigotka.
Jako pierwszą ustawimy zmienną o nazwie **Start**, która jest elementem **wejściowym** i odpowiedzialna jest za włączanie migania lampki jak ma wartość **1** lub wyłączanie procesy przy wartości **0**.
Zaczynamy od wyboru klasy typu zmiennej. Wybur ten spowoduje umieszczenie naszej zmiennej w odpowiednim miejscu FB.



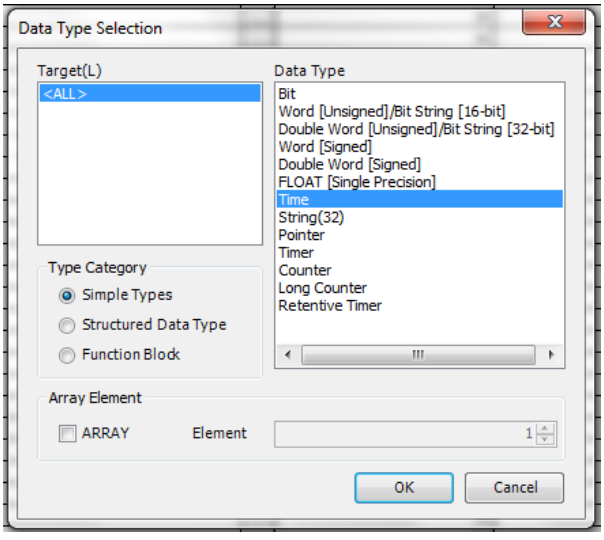
Widok po operacji ustawienia pierwszej etykiety zmiennej:



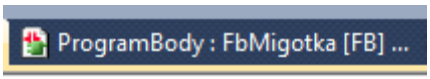
W analogiczny sposób wykonaj ustawienie etykiety wyjścia **o_Led.**



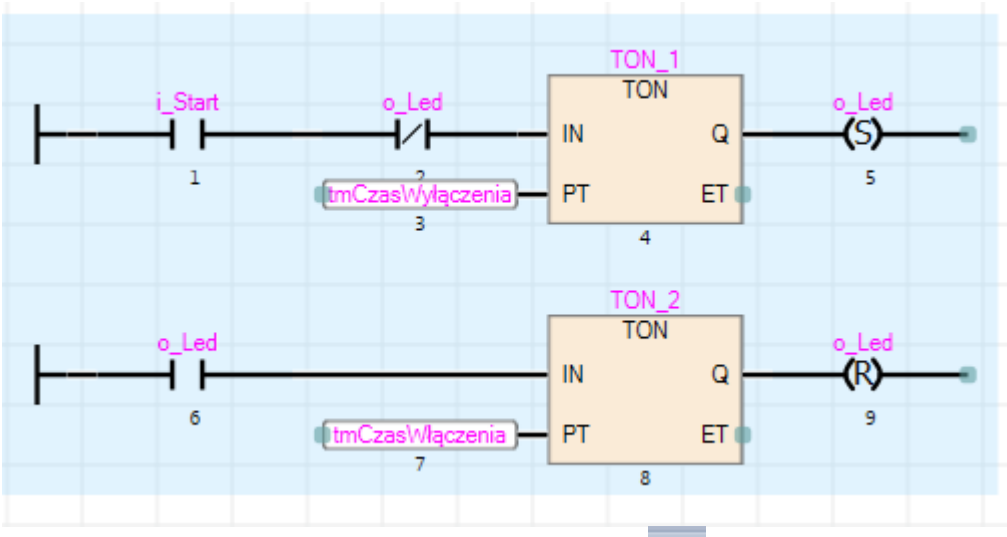
Ostanie dwa parametry potrzebne do działania naszego Bloku Funkcyjnego to zmienne pozwalające regulować czas gdy Led jest zapalony (tmCzasWłączenia) i gdzy się nie świeci (tmCzasWyłączenia).
Użyj typu Time:



	Label Name	Data Type		Class
1	i_Start	Bit	...	VAR_INPUT
2	o_Led	Bit	...	VAR_OUTPUT
3	tmCzasWłączenia	Time	...	VAR_INPUT
4	tmCzasWyłączenia	Time	...	VAR_INPUT

Przejdź do zakładki 

i narysuj program zgodnie z algorytmem **fbMigotka**.



Przekonwertuj program funkcji  i popraw ewentualne błędy.

Output

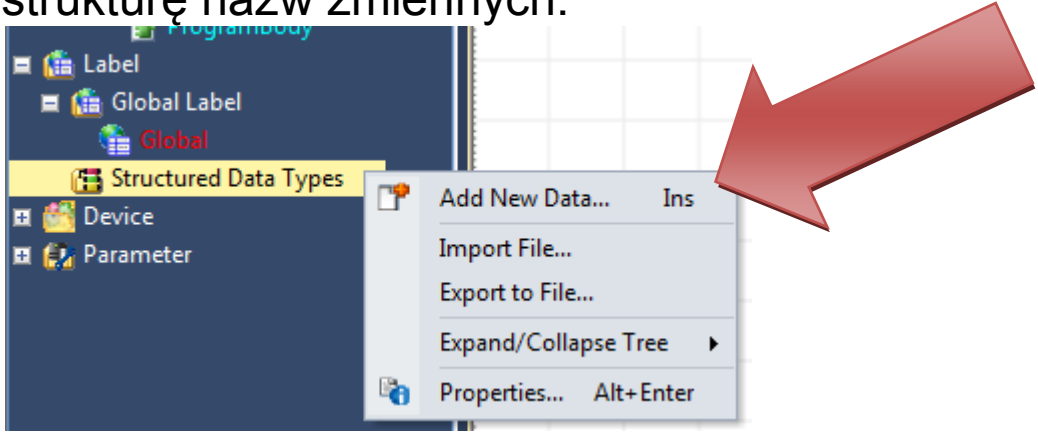
Rebuild All (Reassignment) | Error:0 | Warning:0 | Information

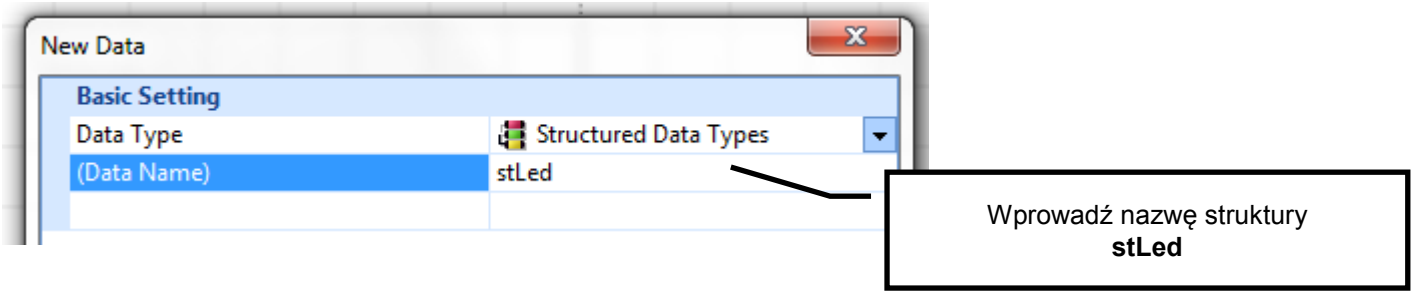
No.	Result	Data Name	Category	Content
1	Information	Number of Steps	Free Volume	99.18[%] (63475 [Step] = 64000 [Step] - 525 [Step])
2	Information	Label	Free Volume	100.00[%] (12288 [Word] = 12288 [Word] - (Global: 0 [Word] + Local: 0 [Word]))
3	Information	Latch Label	Free Volume	100.00[%] (1024 [Word] = 1024 [Word] - (Global: 0 [Word] + Local: 0 [Word]))

Tworzenie struktury typów zmiennych dla funkcji fbMigotka:

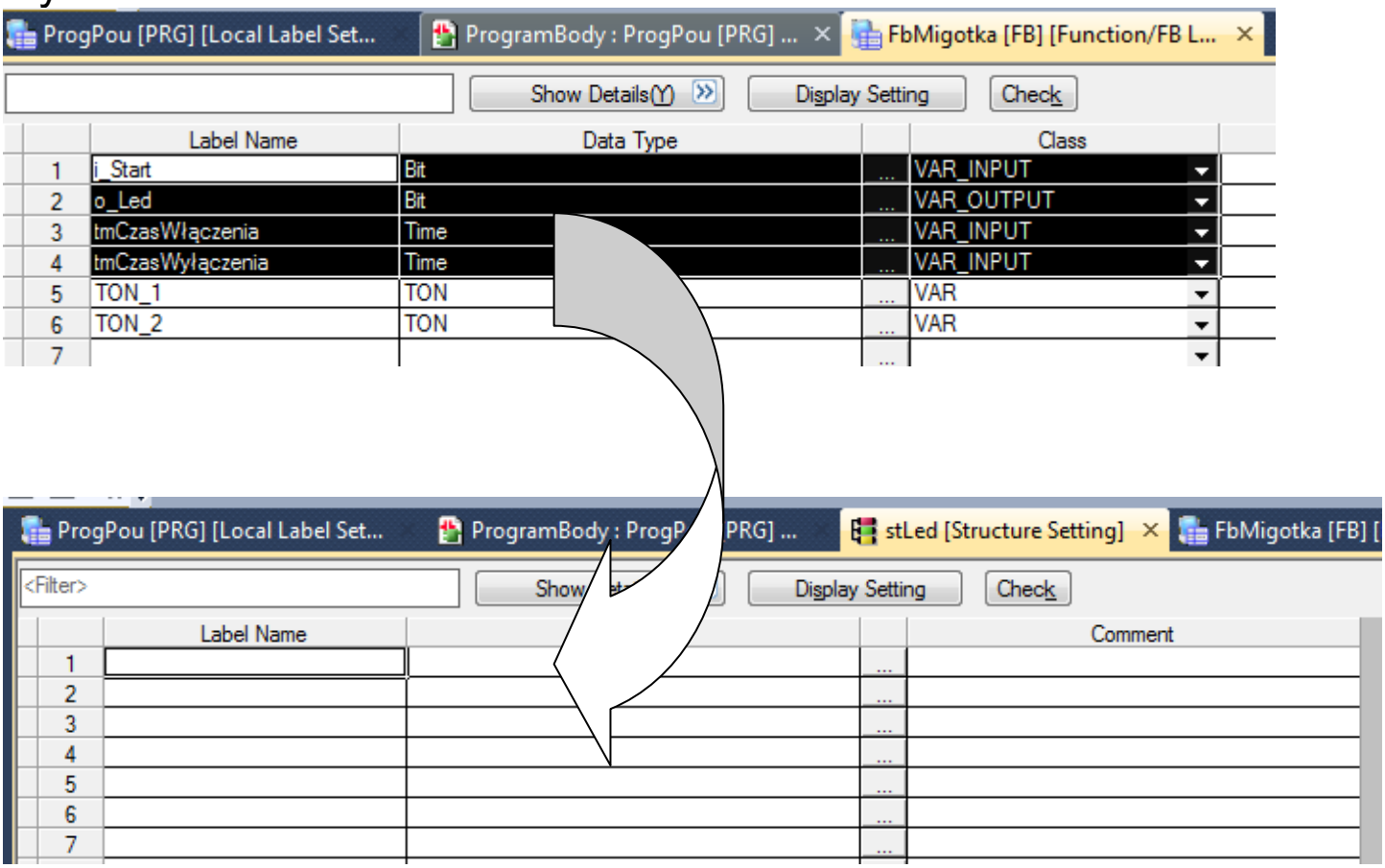
W celu łatwiejszego zarządzania zmiennymi w programie głównym stworzymy strukturę zmiennych, która pozwoli w łatwy sposób dodawać do zmiennych globalnych dowolną ilość zmiennych związanych z poszczególnymi Lamkami Led1, Led2 .. itd.

W drzewie programu rozwiń gałąź **Label** i PPM dodaj nową strukturę nazw zmiennych.

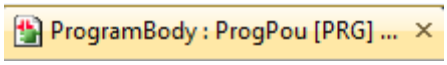




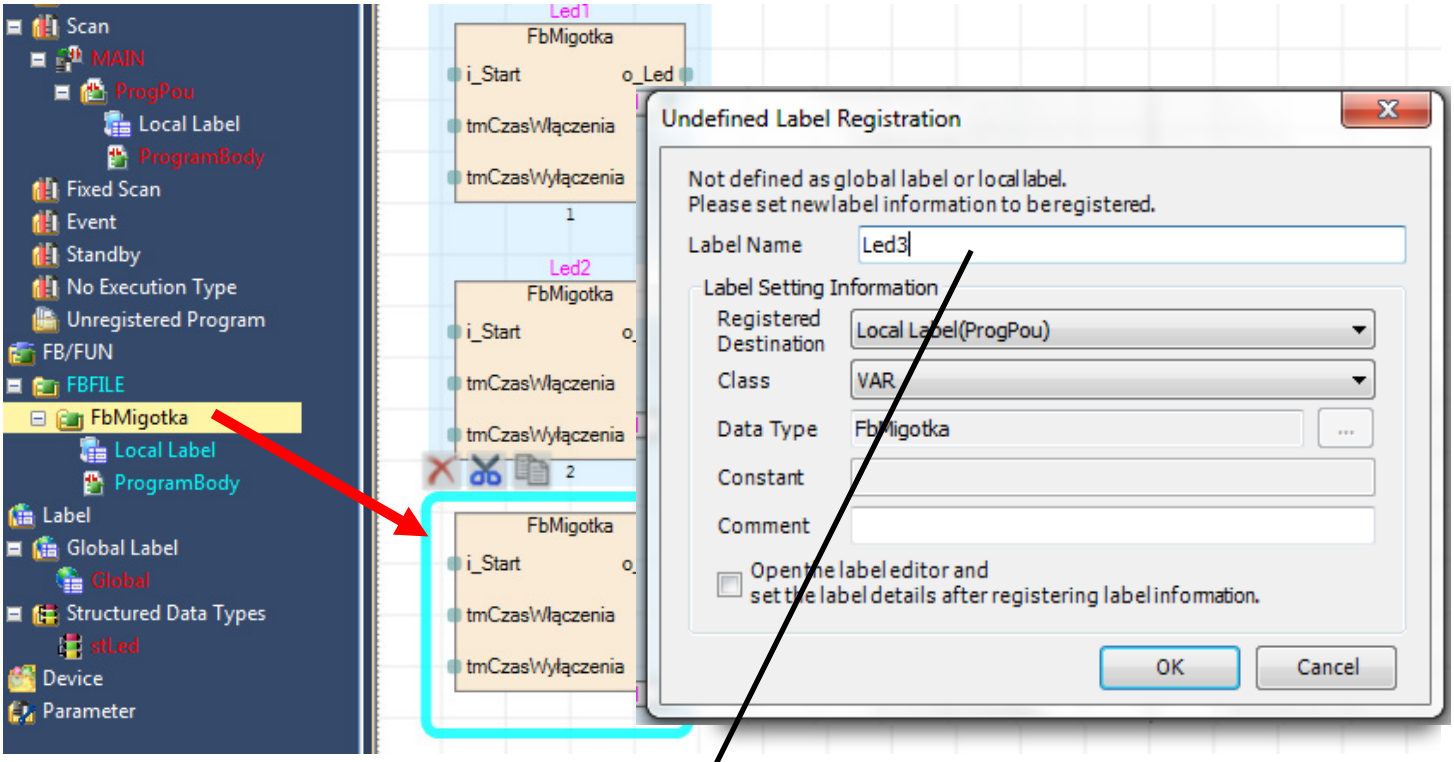
W tablicy etykiet **fbMigotka** zaznacz zmienne które stworzyłeś (bez Timerów i liczników) skopiuj je do schowka **Ctrl+C** i wklej w tablicy etykiet **stLed**.



Tworzenie programu głównego:

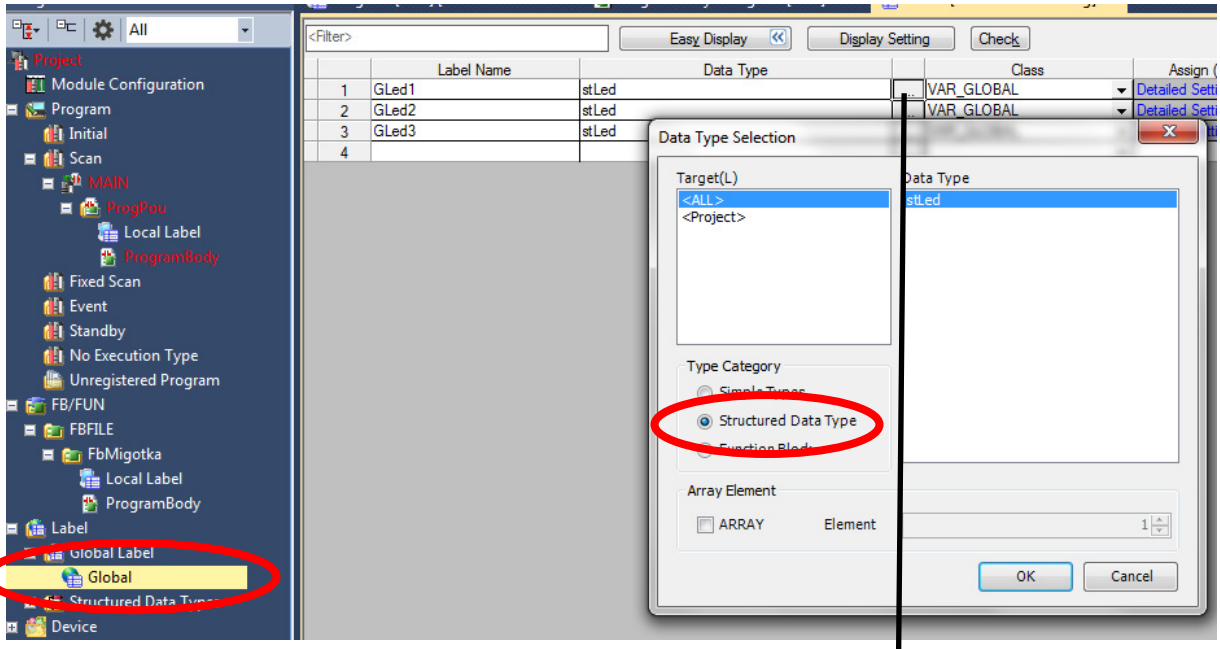
Przejdź na zakładkę 

i przeciągając z drzewa programu umieść 3 funkcje po jednej do każdej z lampek.

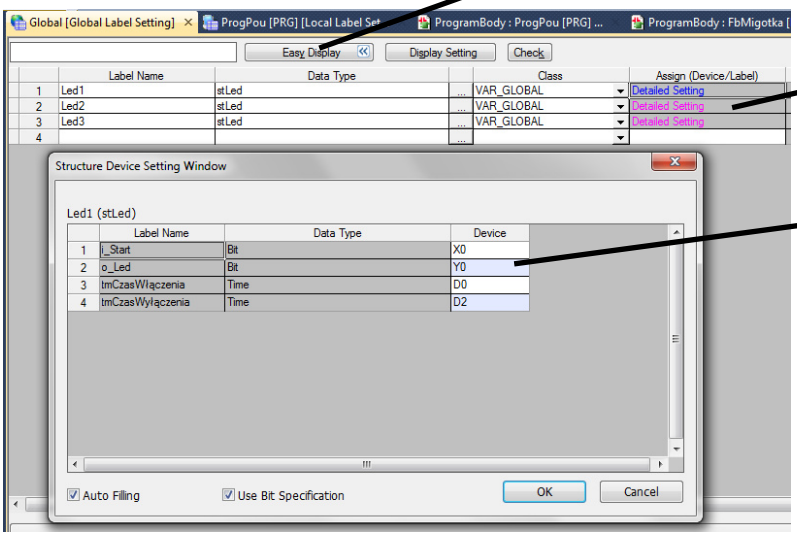


Zmień nazwę obiektu na LedNr

Przejdź do Tablicy etykiet globalnych: Global [Global Label Setting]
Wprowadź 3 etykiety dla 3 lampek. **G** oznacza zmienną globalną: **GLed1, GLed2, GLed3.**



Zmień typ danych na strukturę **stLed**

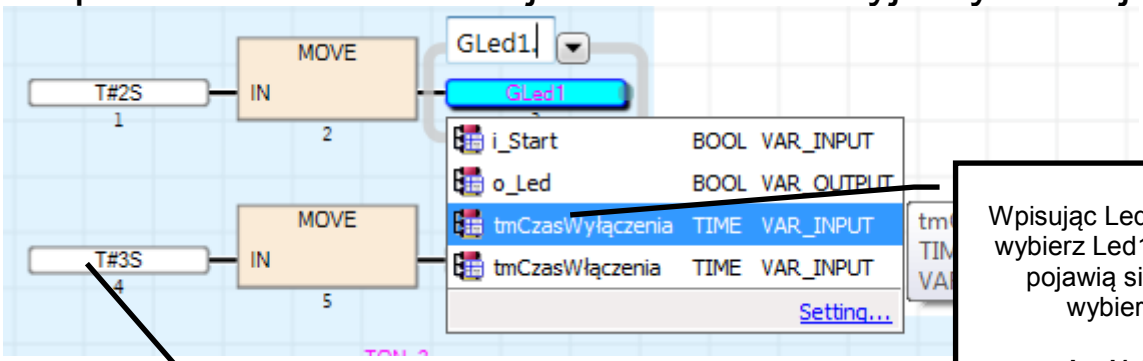


Rozwiń szczegóły tablicy

Otwórz tablicę przypisywania rzeczywistych wejść i wyjść

Uzupełnij wszystkie tablice pamiętając by nie pokrywały się.
Wejścia X0, X1, X2 (do i_ Start)
Wyjścia Y0, Y1, Y2 (do o_Led)
Komórki pamięci co 2 liczby DWord
D0, D2, D4, D6, D8, D10

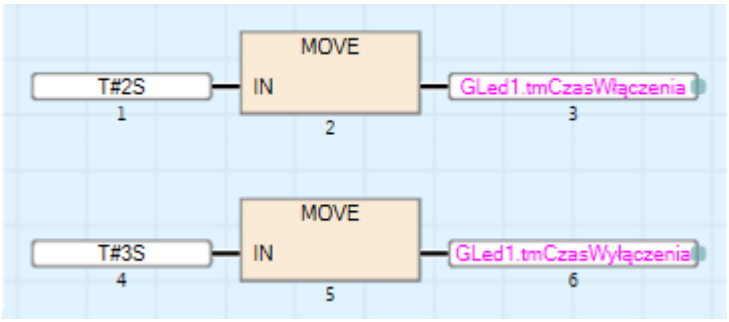
W naszym programie wszystkie lampki będą miały taki sam czas LedOn (2s) i LedOff(3s). Aby ustawić te czasy będziemy ich wartość przechowywać w komórkach pamięci sterownika. Aby zapamiętać zmienną typu Time potrzebujemy 64 bitów pamięci. Potrzebne nam będą 2 sktory pamięci DWord (32bity).
Do przekazania zmiennej do komórek użyjemy funkcji **MOVE**.



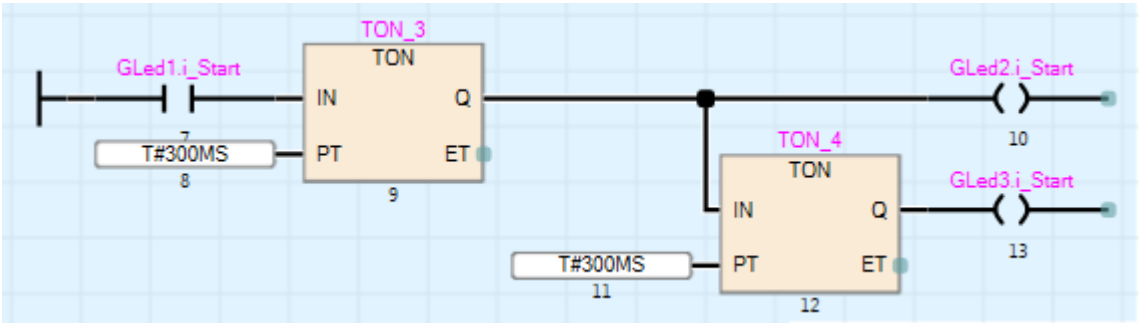
Wpisując Led pojawią się podpowiedzi wybierz Led1 i dodaj separator kropki pojawią się kolejne podpowiedzi wybierz tmCasWłączenia

Led1.tmCzasWłączenia

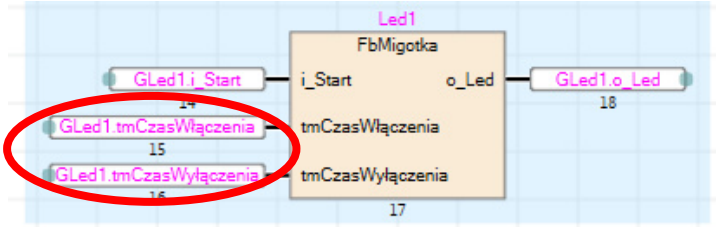
Sposób wprowadzania danych typu Time
T#3S
T#3000MS

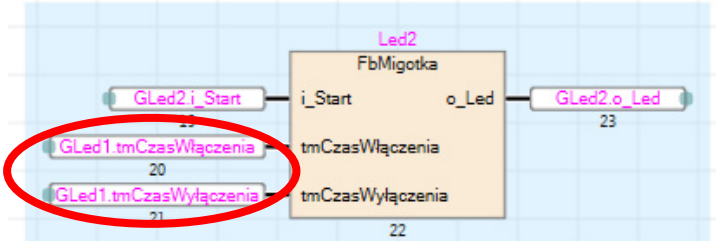


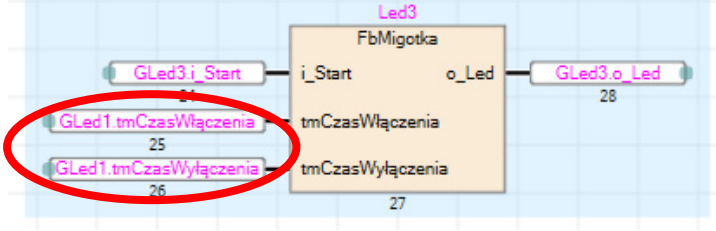
Dodajmy sekwencję opóźnienia startu poszczególnych lampek wynikającą z algorytmu głównego:



W analogiczny sposób uzupełnij wejścia i wyjścia w blokach funkcyjnych, zwróć uwagę że **tmCzasWłączenia** i **tmCzasWyłączenia** odnoszą się tylko do **LED1**. Można je wprowadzać lub skopiować z pierwszego bloku.

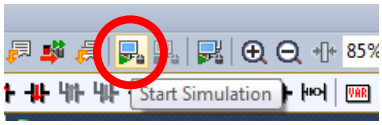




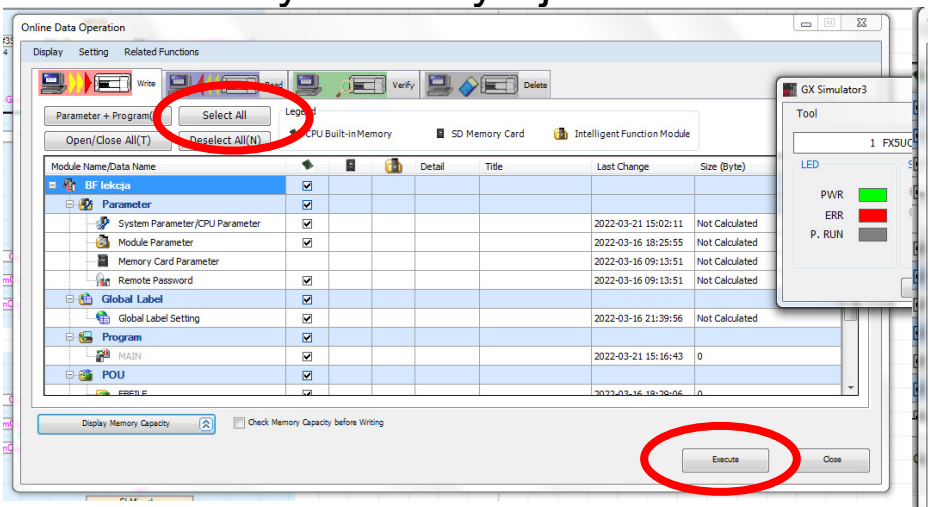


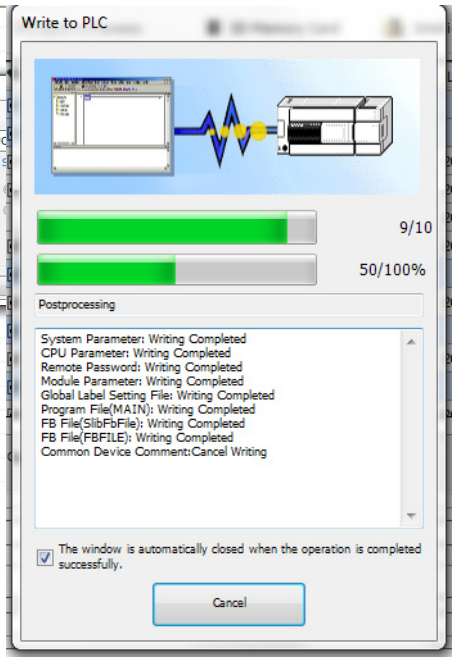
Przekonwertuj program główny i popraw błędy jeżeli wystąpiły.

Testowanie programu:

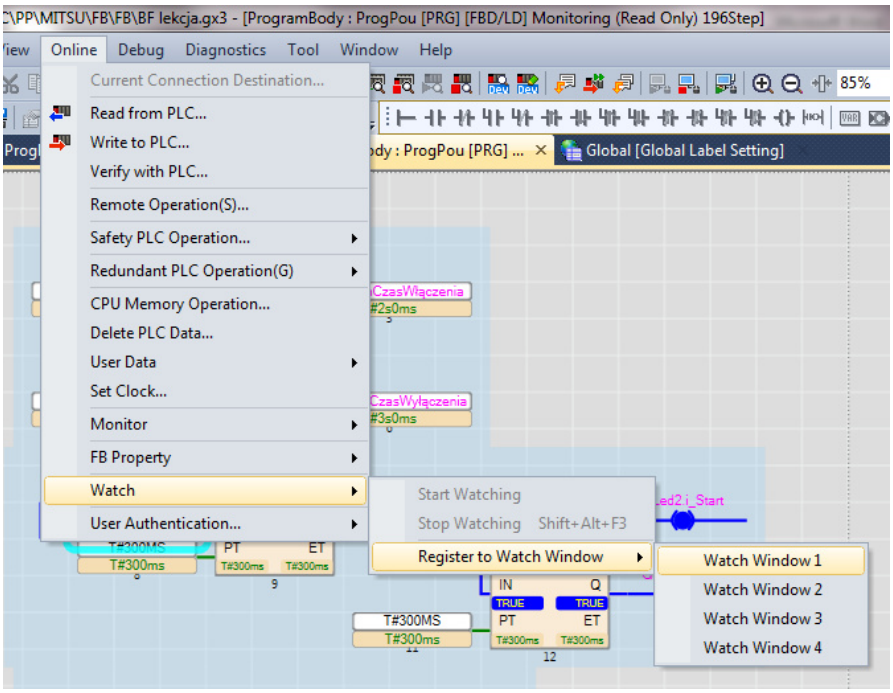
Przejdź do uruchomienia symulacji: 

Zaznacz wszystkie i wyślij na sterownik:

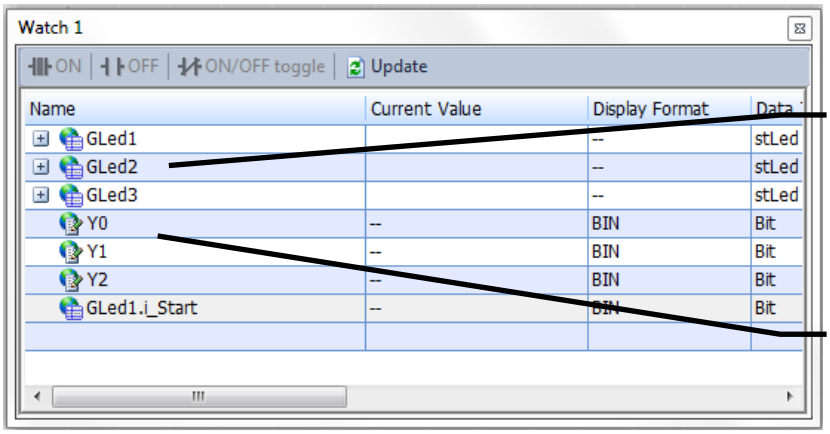




Otwórz okno WATCH do wyświetlenia danych symulacji:



Uzupełnij w tabeli etykiety danych, które chcesz kontrolować.



Sprawdź działanie programu.