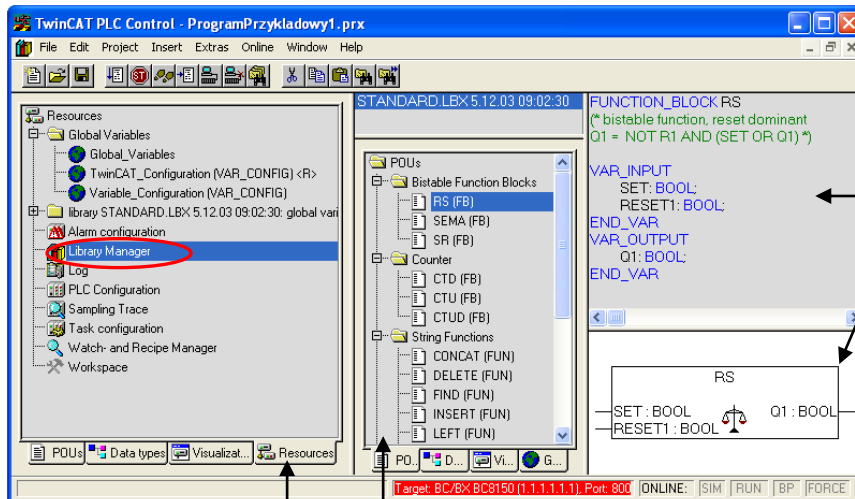


Standardowe bloki funkcjonalne

Wykorzystując języki *ST* i *LD* należy zapoznać się z działaniem standardowych bloków funkcjonalnych (elementy dwustanowe (bistabilne), elementy detekcji zbocza, liczniki, czasomierze) zdefiniowanych w normie IEC-61131-3.

Lista standardowych funkcji dołączonych do projektu, wraz z ich opisem, jest umieszczona w *Library Manager* znajdującym się w zasobach projektu (*Resources*), co pokazano na poniższym rysunku.



Opis bloków funkcyjnych

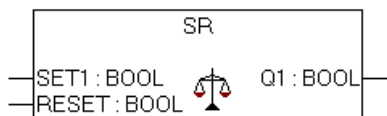
Zasoby Lista bloków funkcyjnych

1. Elementy dwustanowe (bistabilne – *bistable elements*)

Elementy dwustanowe zdefiniowane w normie IEC-61131-3 to przerzutniki SR i RS oraz Semafor.

Przerzutnik SR

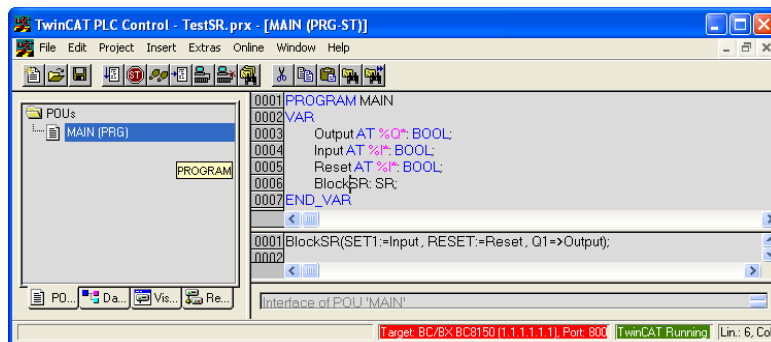
Zadanie Napisać program w języku *ST* umożliwiający testowanie działania przerzutnika SR przy pomocy dwóch przełączników i jednej diody podłączonych do sterownika.



Opis:

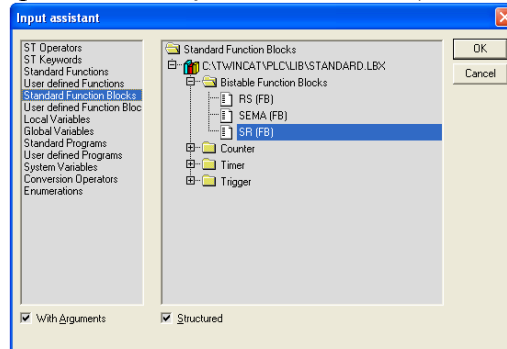
$$Q1 = SET1 \text{ OR } (\text{NOT } RESET \text{ AND } Q1)$$

Program



Ponieważ SR jest blokiem funkcjonalnym, dlatego przed użyciem musi zostać zadeklarowany np. *BlockSR:SR*; gdzie *BlockSR* jest dowolną nazwą. Przed uruchomieniem programu w sterowniku należy pamiętać o powiązaniu zmiennych *Output*, *Input* oraz *Reset* z fizycznymi wyjściami/wejściami.

Aby ułatwić programistom tworzenie kodu, pakiet *TwinCAT PLC Control* wyposażono w okno pomocy wywoływane przez naciśnięcie klawisza **F2** (zobacz rysunek poniżej).

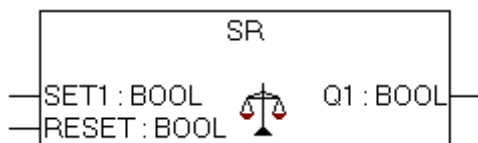


Wybierając odpowiedni element programowy i naciskając przycisk OK uzyskuje się, w oknie programu, wpis podpowiadający jak wywołać dany element. Dla bloku funkcyjnego SR jest to np.

SR(SET1:= , RESET:= , Q1=>);

Zapisy 'SET1:=' oraz 'RESET:=' podpowiadają nazwy wejściowych parametrów bloku zaś 'Q1=>' parametru wyjściowego. Aby w programie użyć bloku funkcjonalnego, jego nazwę (w linii uzyskanej z podpowiedzi) należy zastąpić własną nazwą oraz przypisać parametrom odpowiednie argumenty np. *BlockSR(SET1:=Input , RESET:=Reset , Q1=>Output)*. Tak, więc zmienne wewnętrzne bloku funkcjonalnego *SET1* i *RESET* uzyskują wartości równe *Input* i *Reset*, zaś zmienna *Output* wartość równą zmiennej wewnętrznej *Q1*.

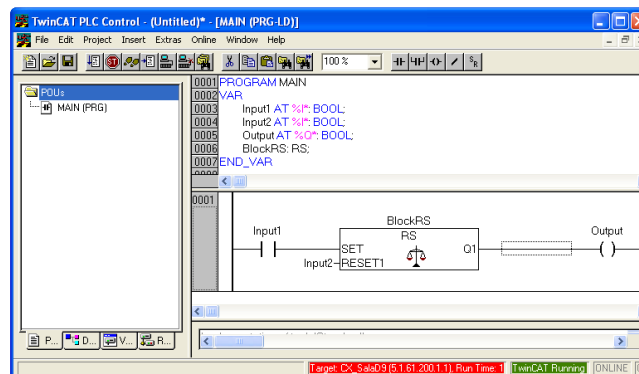
Zadanie. Napisać program w języku *LD* umożliwiający testowanie działania przerzutnika SR przy pomocy dwóch przełączników i jednej diody podłączonych do sterownika.



Opis:

$Q1 = SET1 \text{ OR } (\text{NOT } RESET \text{ AND } Q1)$

Program

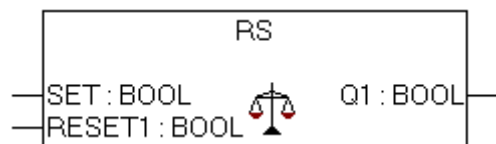


Blok funkcyjny RS można wstawić do programu wykorzystując opcję *Function Block* ... menu kontekstowego prawego klawisza myszy.

Przed uruchomieniem programu w sterowniku należy pamiętać o powiązaniu zmiennych *Output*, *Input* oraz *Reset* z fizycznymi wyjściami/wejściami.

Przerzutnik RS

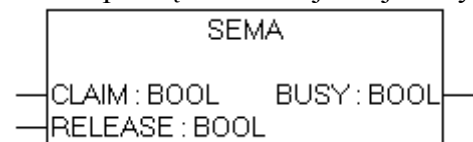
Zadanie. Napisać program umożliwiający testowanie działania przerzutnika RS przy pomocy dwóch przełączników i jednej diody podłączonych do sterownika.



Opis:
 $Q1 = \text{NOT RESET1 AND (SET OR Q1)}$

Semafor SEMA

Zadanie. Napisać program umożliwiający testowanie działania Semafora przy pomocy dwóch przełączników i jednej diody podłączonych do sterownika.



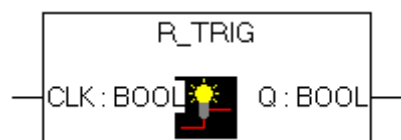
Opis: CLAIM=TRUE ustawia semafor (BUSY=TRUE),
 RELEASE=TRUE gdy CLAIM=FALSE
 zwalnia semafor (BUSY=FALSE)

2. Elementy detekcji zbocza

Elementy detekcji zbocza zdefiniowane w normie IEC-61131-3 umożliwiają wykrycie zbocza narastającego R_TRIG oraz zbocza opadającego F_TRIG.

Detektor zbocza narastającego (*rising edge*) R_TRIG

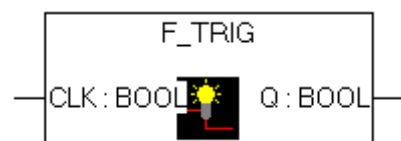
Zadanie. Napisać program umożliwiający testowanie działania bloku funkcyjnego R_TRIG przy pomocy jednego przełącznika i jednej diody podłączonych do sterownika.



Opis: Wyjście Q utrzymuje wartość TRUE pomiędzy dwoma kolejnymi wywołaniami bloku w sytuacji, gdy nastąpiła zmiana wartości wejścia CLK z FALSE na TRUE. W przeciwnym wypadku Q utrzymuje wartość FALSE. Dla trzech kolejnych wywołań bloku, gdy pomiędzy dwoma pierwszymi wywołaniami nastąpiła odpowiednia zmiana CLK, wyjście Q utrzymuje wartość TRUE pomiędzy drugim i trzecim wywołaniem bloku.

Detektor zbocza opadającego (*falling edge*) F_TRIG

Zadanie. Napisać program w języku ST umożliwiający testowanie działania bloku funkcyjnego F_TRIG przy pomocy jednego przełącznika i jednej diody podłączonych do sterownika.



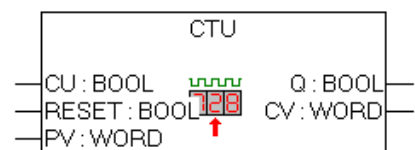
Opis: Wyjście Q utrzymuje wartość TRUE pomiędzy dwoma kolejnymi wywołaniami bloku w sytuacji, gdy nastąpiła zmiana wartości wejścia CLK z TRUE na FALSE. W przeciwnym wypadku Q utrzymuje wartość FALSE. Dla trzech kolejnych wywołań bloku, gdy pomiędzy dwoma pierwszymi wywołaniami nastąpiła odpowiednia zmiana CLK, wyjście Q utrzymuje wartość TRUE pomiędzy drugim i trzecim wywołaniem bloku.

3. Liczniki

Liczniki zdefiniowane w normie IEC-61131-3 to licznik dodający, odejmujący oraz dodająco-odejmujący.

Licznik dodający (*up-counter*) CTU

Zadanie. Napisać program umożliwiający zliczanie liczby wykrytych zboczy narastających na wybranym wejściu binarnym. Liczbę zliczonych zboczy należy wyświetlać w systemie binarnym przy pomocy trzech kolejnych diod. Gdy liczba impulsów przekroczy wartość możliwą do wyświetlenia przy pomocy trzech diod, zliczanie należy rozpocząć ponownie od wartości zero.



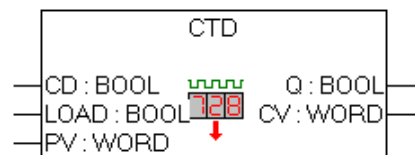
Opis:

- CU – wejście, którego zmiany z wartości FALSE na TRUE są zliczane
- RESET – wejście zerujące licznik
- PV – zadana liczba impulsów
- CV – liczba zliczonych impulsów
- Q – wyjście załączane, gdy CV osiągnie wartość PV

Liczbę zliczonych zboczy należy obserwować w trybie podglądu działania programu. Aby zrealizować wyświetlanie binarne należy wykorzystać dostęp do poszczególnych bitów przy pomocy operatora ‘.’ np. bit o indeksie 0 zmiennej N to N.0 itd..

Licznik odejmujący (*down-counter*) CTD

Zadanie. Napisać program umożliwiający zliczanie „w dół” liczby wykrytych zboczy opadających na wybranym wejściu binarnym. Odliczaną „w dół” liczbę zboczy opadających należy wyświetlać w systemie binarnym przy pomocy trzech kolejnych diod. Gdy liczba impulsów osiągnie wartość 0 zliczanie „w dół” należy rozpocząć ponownie.



Opis:

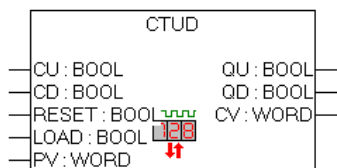
- CD – wejście, którego zmiany z wartości FALSE TRUE są zliczane
- LOAD – wejście ustawiające CV na wartość PV
- PV – zadana liczba impulsów
- CV – liczba zliczonych impulsów
-Q – wyjście załączane, gdy CV osiągnie wartość 0

Liczbę zliczonych zboczy należy obserwować w trybie podglądu działania programu. Aby zrealizować wyświetlanie binarne należy wykorzystać dostęp do poszczególnych bitów przy pomocy operatora ‘.’ np. bit o indeksie 1 zmiennej N to N.1 itd.. Do wykrywania zbocza opadającego należy zastosować blok funkcyjny F_TRIG.

W pewnych sytuacjach, gdy nastąpiła zmiana programu, należy wykonać restart sterownika (menu **Online** opcja **Reset** po uprzednim zalogowaniu się opcją **Login**, przed wykonaniem restartu należy zatrzymać działanie programu – opcja **Stop** menu **Online** a następnie wykonać **Reset** i ponownie uruchomić sterownik opcją **Run**).

Licznik dodająco-odejmujący (*down-counter*) CTUD

Zadanie. Napisać program umożliwiający zliczanie wykrytych zboczy narastających dla dwóch wejść binarnych. Zbocze narastające na wejściu pierwszym powoduje zwiększanie liczby zliczonych impulsów, zaś zbocze narastające na wejściu drugim zmniejszanie tej liczby. Liczbę impulsów należy wyświetlać w systemie binarnym przy pomocy trzech kolejnych diod. Gdy liczba impulsów przekroczy wartość zero lub wartość możliwą do wyświetlenia przy pomocy trzech diod, zliczanie należy rozpocząć od zera.



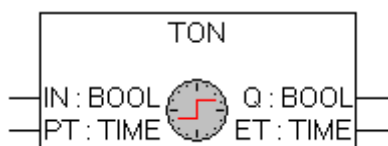
Opis: CU – wejście, którego zmiany z wartości FALSE na TRUE są zliczane „w górę”
 CD – wejście, którego zmiany z wartości FALSE na TRUE są zliczane „w dół”
 RESET – wejście zerujące licznik
 LOAD – wejście ustawiające CV na wartość PV
 PV – wartość zadana dla zliczania „w dół”
 CV – liczba zliczonych impulsów
 QU – wyjście załączane, gdy CV osiągnie wartość PV
 QD – wyjście załączane, gdy CV osiągnie wartość 0

4. Czasomierze

Czasomierze (*timery*) zdefiniowane w normie IEC-61131-3 to czasomierz załączający z opóźnieniem, wyłączający z opóźnieniem, generator impulsu o zadanym czasie trwania oraz zegar czasu rzeczywistego. W pakiecie *TwinCAT* zegar czasu rzeczywistego nie znajduje się w standardowej bibliotece *Standard.Lib*, aby go użyć należy dołączyć do projektu bibliotekę *TcUtilities.Lib*.

Czasomierz załączający (*on-delay*) TON

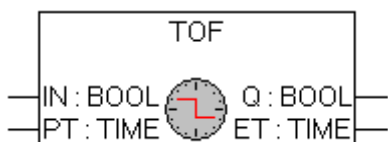
Zadanie. Napisać program załączający wybrane wyjście sterownika z opóźnieniem 10 sekund w stosunku do chwili wykrycia zbocza narastającego na wybranym wejściu sterownika.



Opis: IN – wejście uruchamiające czasomierz (zbocze narastające uruchamia czasomierz, zbocze opadające zeruje odliczany czas)
 PT – wartość zadana czasu do odliczenia
 Q – wyjście załączane, gdy upłynie założony czas opóźnienia
 ET – aktualna wartość mierzonego czasu

Czasomierz wyłączający (*off-delay*) TOF

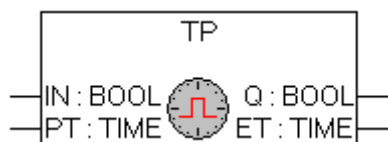
Zadanie. Napisać program wyłączający wybrane wyjście sterownika z opóźnieniem 10 sekund w stosunku do chwili wykrycia zbocza opadającego na wybranym wejściu sterownika.



Opis: IN – wejście uruchamiające czasomierz (zbocze opadające uruchamia czasomierz, zbocze narastające zeruje odliczany czas)
 PT – wartość zadana czasu do odliczenia
 Q – wyjście wyłączane, gdy upłynie założony czas opóźnienia
 ET – aktualna wartość mierzonego czasu

Generator impulsu (*timer-pulse*) TP

Zadanie. Napisać program załączający wybrane wyjście sterownika na czas równy 10 sekund do chwili wykrycia zbocza narastającego na wybranym wejściu sterownika.



Opis: IN – wejście uruchamiające generator (poziom wysoki uruchamia generator, poziomy niski zeruje licznik gdy czas impulsu osiągnął lub przekroczył wartość zadaną PT)
 PT – wartość zadana czasu trwania impulsu
 Q – wyjście załączane na czas PT, od momentu wykrycia zbocza narastającego na wejściu IN
 ET – aktualna wartość mierzonego czasu

Zadanie. Napisać program sygnalizujący przy pomocy jednej diody wystąpienie zbocza narastającego i opadającego na wybranym wejściu binarnym. Informacja o

wystąpieniu odpowiedniego zbocza jest do czasu wystąpienia kolejnego zbocza utrzymywana (dioda świeci – zbocze narastające, dioda nie świeci – zbocze opadające). W programie należy wykorzystać bloki R_TRIG, F_TRIG oraz odpowiedni blok dwustanowy.

Zadanie. Napisać program generujący na wybranym wyjściu sterownika przebieg prostokątny o czasie trwania stanu wysokiego 6 sekund i stanu niskiego 3 sekund. Należy wykorzystać jedno wyjście modułu KL2408.

