



AKADEMIA GÓRNICZO-
HUTNICZA
IM. STANISŁAWA STASZICA
W KRAKOWIE

LABORATORIUM PRZEMYSŁOWYCH
SYSTEMÓW STEROWANIA



Wydział Inżynierii Mechanicznej i Robotyki



Katedra Automatykacji Procesów

Przedmiot:

Przemysłowe Systemy Sterowania (PSS)

Laboratorium:

Laboratorium Industry 4.0 – język GRAPH (SFC) dla sterowników
S7-1500

Kraków

Wszelkie prawa zastrzeżone dla KAP, WIMiR, AGH
Jakiegolwiek kopiowanie materiałów (w tym na potrzeby prac dyplomowych) bez zgody autorów niedozwolone.
(wersja kwiecień 2022)

Spis treści

| | | |
|-----|--|----|
| 1 | Przygotowanie sterownika..... | 3 |
| 2 | ProfiSAFE..... | 5 |
| 3 | Opis wejść..... | 7 |
| 4 | Podstawy SFC..... | 8 |
| 5 | Przygotowanie „szablonu” FB w języku GRAPH do realizacji przykładów..... | 11 |
| 5.1 | Przygotowanie panelu HMI do wyboru jednego z oprogramowanych przykładów..... | 11 |
| 5.2 | Przygotowanie szablonu bloku funkcyjnego FB w języku Graph..... | 14 |
| 5.3 | Sześciokrotne skopiowanie FB PRZYKLAD pod nazwami PRZYKLAD_1 do PRZYKLAD_6..... | 18 |
| 5.4 | Wywołanie skopiowanych bloków w OB1..... | 19 |
| 5.5 | Mapowanie na wyjścia fizyczne wyjść z sześciu FB przypisanych do elementów tablic..... | 23 |
| 6 | Przykład rozwiązania – generator w Graph czyli przerost formy nad treścią..... | 24 |
| 7 | Zadania do samodzielnej realizacji..... | 30 |
| 7.1 | Przykład 1 – sekwencja podstawowa..... | 30 |
| 7.2 | Przykład 2 – obsługa jednego siłownika..... | 30 |
| 7.3 | Przykład 3 – obsługa czasu zadziałania..... | 30 |
| 7.4 | Przykład 4 – obsługa wszystkich aktuatorów..... | 31 |
| 7.5 | Przykład 5 - alternatywa..... | 31 |
| 7.6 | Przykład 6 - współbieżność..... | 32 |

1 Przygotowanie sterownika

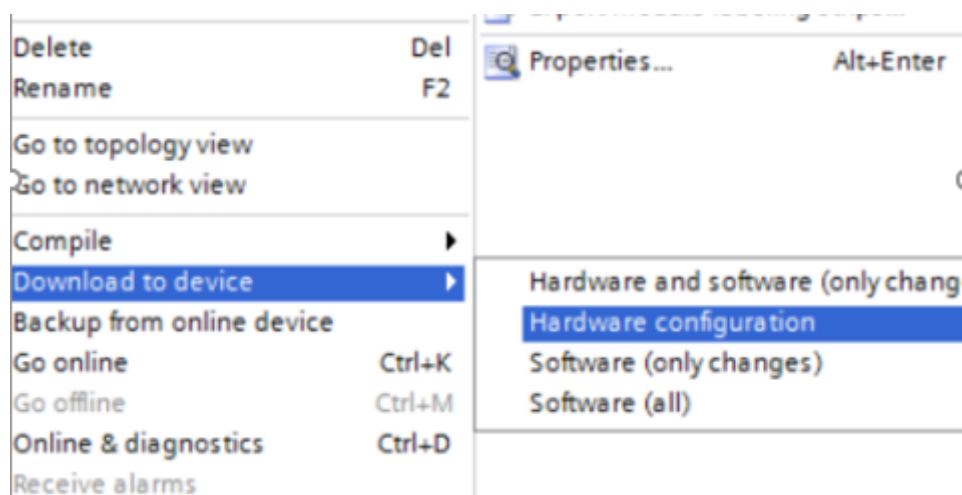
W lokalizacji: **D:\ProgramyBazowe\V17** znajduje się przygotowany projekt bazowy, który zawiera:

- pełną konfigurację sterownika Simatic S7-ET200,
- konfigurację panelu HMI Simatic TP700 comfort,
- obsługę powiązanych urządzeń peryferyjnych,
- zadeklarowaną tablicę tagów określających fizyczne wejścia oraz fizyczne wyjścia.

Krok 01: Uruchomić program TIA Portal v17 oraz otworzyć z widoku **Portal View** otworzyć plik z wymienionej wcześniej lokalizacji.

Krok 02: Wgrać hardware do sterownika według podanej poniżej instrukcji.

1. Na ekranie *Devices & Networks* należy kliknąć prawym przyciskiem myszy na sterownik PLC i wybrać opcję *Download to device* -> *Hardware configuration* (Rys. 1)

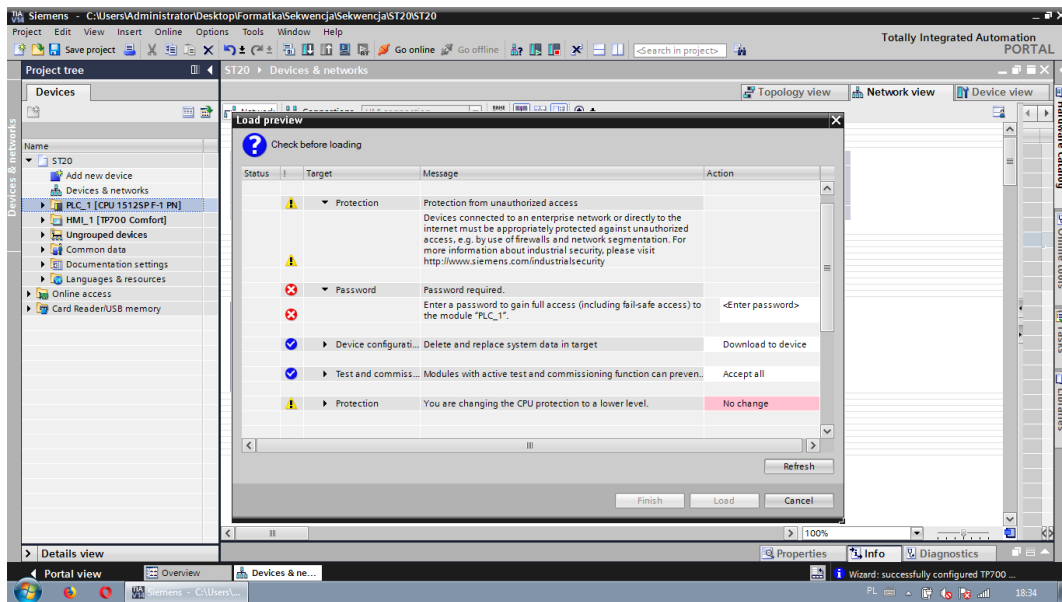


Rys. 1. Rozwinięta lista po kliknięciu PPM na sterownik PLC

2. Po pojawieniu się okna *Load preview* wybrać *Load*. Jeśli trzeba będzie wprowadzić hasło, wpisz "0" i zatwierdź przyciskiem Enter.

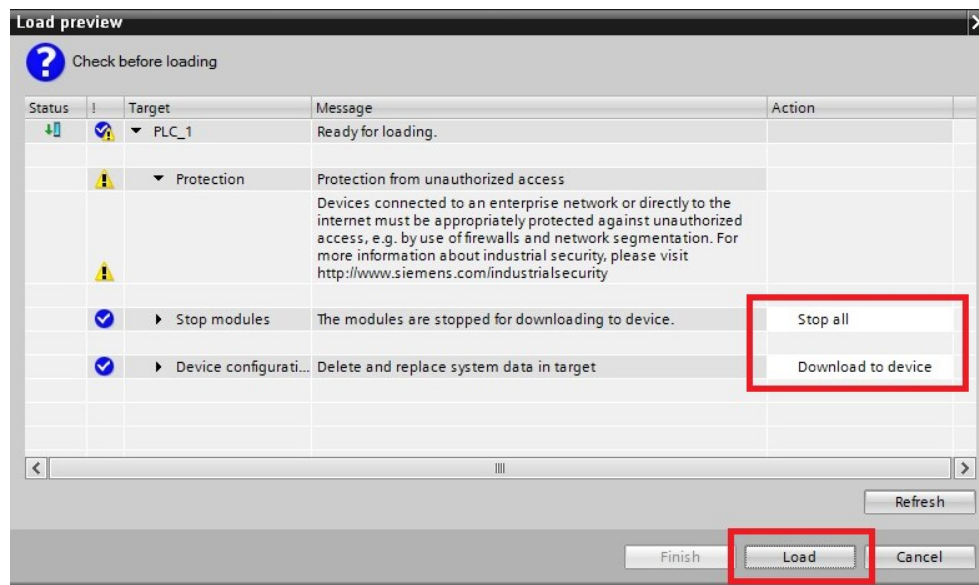
UWAGA!! W laboratorium stosujemy hasło: „0” a więc 1 znak, po prostu cyfra zero, które należy wpisać i potwierdzić (Rys. 2).

NIE NALEŻY USTAWIAĆ POD ŻADNYM POZOREM INNEGO HASŁA!!!



Rys. 2. Okno wyświetlone podczas wgrwania programu z hasłem

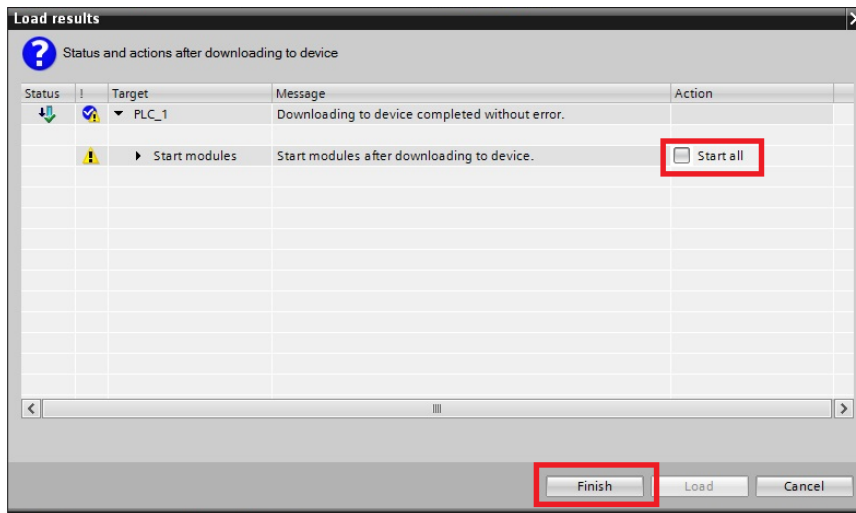
3. W przypadku braku hasła podczas wgrwania programu pojawi się okno, w którym należy zaznaczyć odpowiednie opcje, a następnie zatwierdzić przyciskiem "Load". (Rys. 3)



Rys. 3. Okno wyświetlone podczas wgrwania programu

4. W następnym oknie należy odznaczyć opcję "Start modules" i zatwierdzić przyciskiem "Finish". (Rys. 4)

UWAGA!! W przypadku sterowników w żółtej technologii FAIL-SAFE (inaczej niż w zwykłych sterownikach) porównywany jest znacznik sprzętowy oraz softwarowy. Stąd, jeśli przed wgraniem programu wystartuje się moduły, to nie będzie możliwa poprawna praca sterownika PLC.



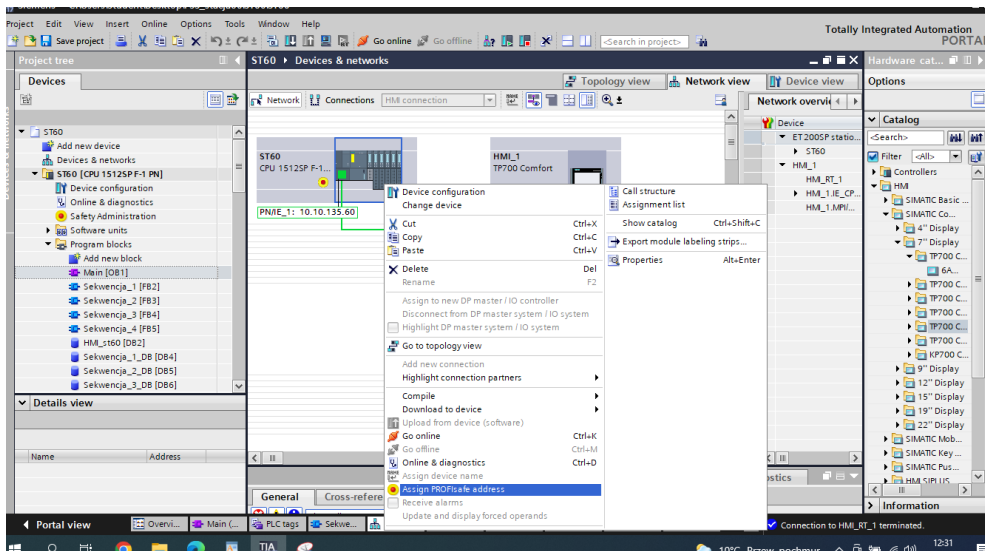
Rys. 4. Okno wyświetlone po wgraniu programu

5. Na ekranie *Devices & Networks* należy kliknąć prawym przyciskiem myszy na sterownik PLC i wybrać opcję *Download to device -> Software (All)*
6. Po wgraniu software należy wystartować moduły albo przez zaznaczenie przed kliknięciem *Finish -> Start modules*, bądź też poprzez kliknięcie RUN po wgraniu. Trzecią opcją jest przełożenie przełącznika znajdującego się na fizycznym sterowniku w pozycję środkową (Stop), a następnie w pozycję górną (RUN)

UWAGA!! W przypadku sterowników s7-et200SP przełącznik sprzętowy ma nadrzędną funkcjonalność nad ikonami RUN i STOP w Tia Portal. Jeśli fizycznie przełącznik jest w środkowej pozycji, to sterownik pozostanie w trybie pracy STOP, niezależnie od kliknięcia w TIA PORTAL. Jeśli sterownik ma fizyczny przełącznik trybu w pozycji górnej, to wtedy można przejść w tryb pracy stop z poziomu TIA PORTAL.

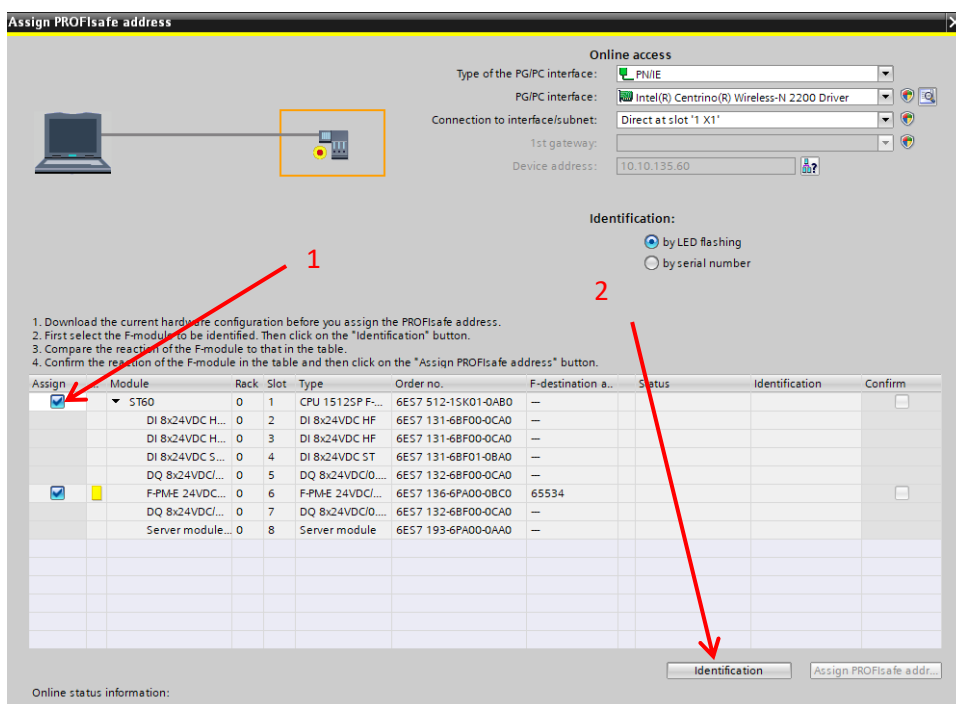
2 ProfiSAFE

Jeśli po wgraniu ustawień sprzętowych zauważysz, że na **ŻÓŁTYM MODULE** większość diod świeci się na czerwono, to oznacza, że należy przypisać sterownikowi adres *ProfiSAFE*. W tym celu w drzewie projektu kliknij *Devices and Networks*, a następnie prawym przyciskiem myszy kliknij w sterownik (Rys. 5). Uwaga – kliknij w sterownik, a nie w szare pole wokół sterownika. Następnie z menu wybierz opcję *Assign ProfiSafe address*.



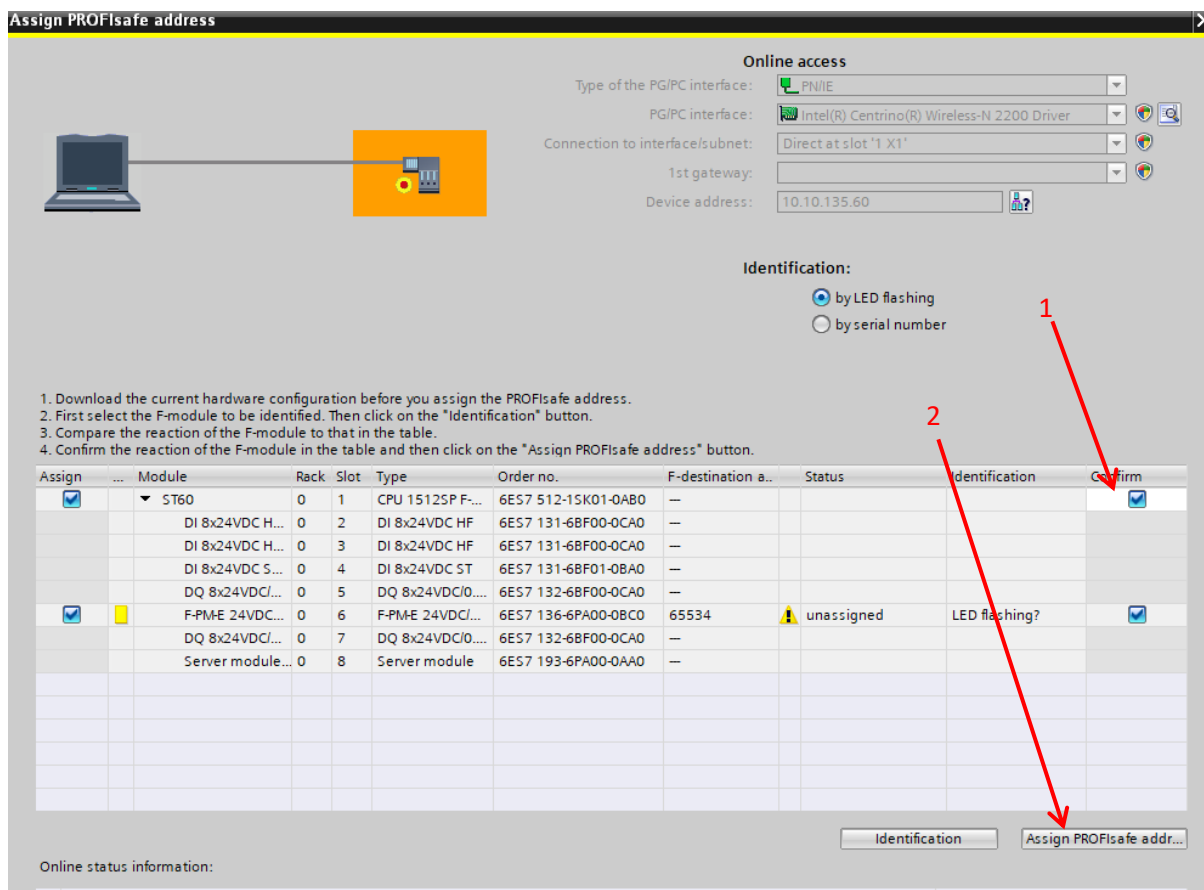
Rys. 5. Przypisywanie adresu Profisafe – początek

W wywołanym oknie (Rys. 6) zaznacz pole *Assign* (1), a następnie kliknij *Identyfikacja*(2)



Rys. 6. Okno identyfikacji urządzenia

W ostatniej kolejności pojawi się okno zatwierdzenia (Rys. 7). Należy zaznaczyć *Confirm* (1) oraz przycisk *Assign ProfiSafe Address* (2). Po tym pojawi się okno, które w 60 sekund należy zatwierdzić przyciskiem YES (to jest kolejne zapytanie, czy jesteś pewien/pewna, że jesteś pewien/pewna, że chcesz przypisać *ProfiSafe*).

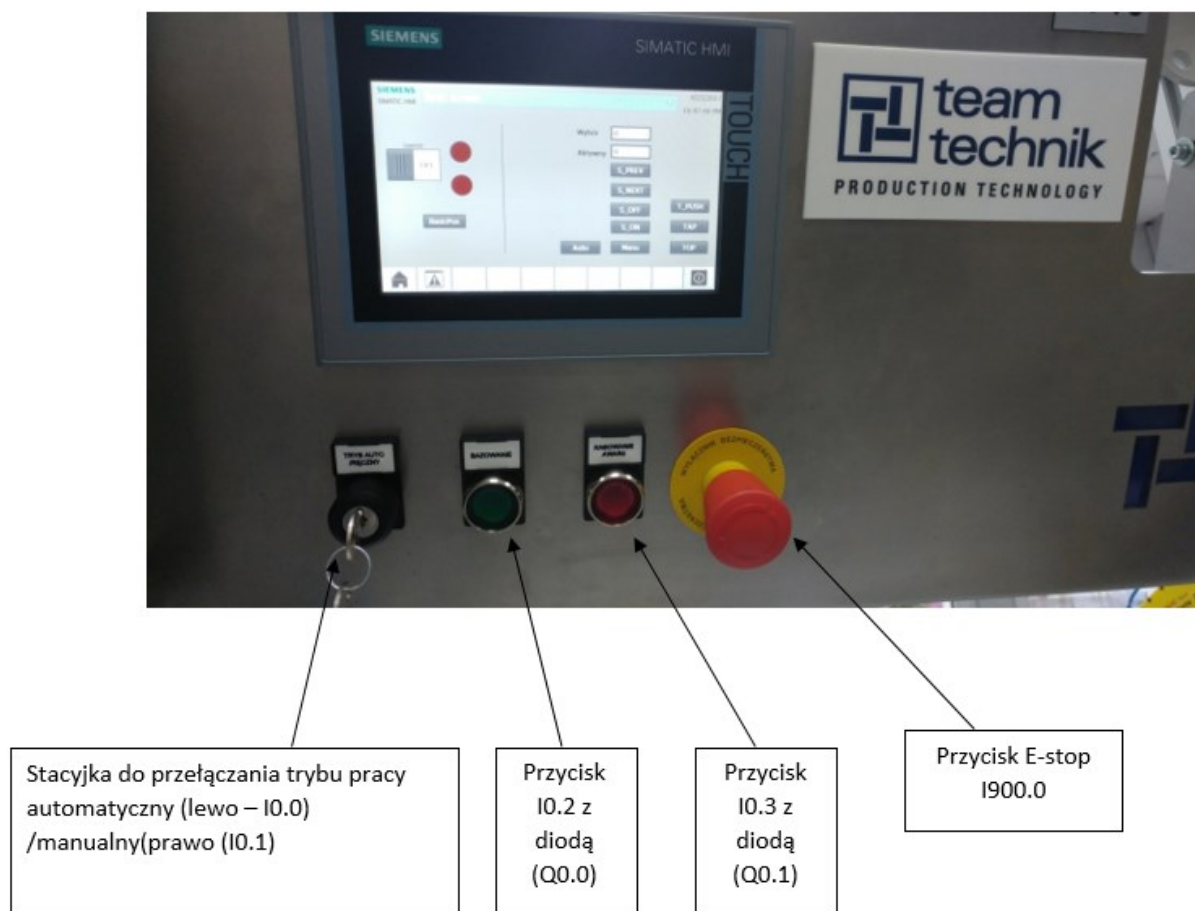


Rys. 7. Końcowe zatwierdzenie ProfiSafe

UWAGA!! W przypadku, gdy na module po wgraniu hardware'u nie świecą się diody na czerwono, to należy pominąć ProfiSafe

3 Opis wejść

Każde stanowisko wyposażone jest w **4 identyczne fizyczne wejścia** oraz **2 fizyczne cyfrowe wyjścia** (Rys. 8). Dodatkowo każde stanowisko wyposażone jest w minimum **dwa siłowniki dwustronnego działania** z obsługiwane przez zawory: wysuwający i wsuwający oraz dwa cyfrowe **czujniki położenia krańcowych tłoka**. Ich adresy fizyczne są dostępne w tabeli tagów załadowanego projektu w TIA Portal



Rys. 8. Panel operatora z panelem HMI, przyciskami i sygnalizatorami

UWAGA!! Upewnij się przy obsłudze siłowników, że przycisk E-STOP jest wyciśnięty i zatwierdziłeś ruch siłowników przyciskiem czerwonym

4 Podstawy SFC

Sekwencyjny schemat funkcjonalny, w skrócie SFC, to jeden z pięciu języków programowania według postanowień normy IEC 61131-3. Jako język graficzny opiera się na metodzie GRAFCET, której podstawą jest teoria sieci Petriego. Jest stosowany w złożonych przypadkach sterowania, to znaczy wtedy, gdy proces technologiczny wymaga ściśle określonej kolejności wykonywania operacji lub zadanego czasu ich realizacji. Jako jedyny z języków określonych w trzeciej części normy IEC-61131 nie jest samodzielny. Do realizacji swoich zadań wykorzystuje jeden z pozostałych.

W środowisku projektowym TIA Portal sekwencyjny schemat funkcjonalny występuje pod nazwą **S7-GRAPH** i jest tworzony jako graf w bloku funkcyjnym, w skrócie FB. Zestawienie parametrów wchodzących i wychodzących z takiego bloku zostało przedstawione w tabeli 1. Po utworzeniu takiego bloku automatycznie zostaje do niego przypisany instancyjny blok danych – DB. Dedykowany blok danych jest używany jedynie przez utworzony blok FB. Usuwanie, zmienianie lub dodawanie

zawartych w nim zmiennych zachodzi poprzez zmianę bloku funkcyjnego. Służy do przechowywania statycznych zmiennych lokalnych zdefiniowanych w bloku FB do momentu ich ponownego zapisu.

Sekwencja pozwala na realizowanie 4 trybów pracy. Są to „SW_AUTO”, „SW_TAP”, „SW_TOP” i „SW_MAN”. Wszystkie zostały opisane w tabeli 1.

Poza argumentami standardowymi możemy w interfejsie FB sekwensera (sekwenser - FB dla którego wybierzemy język S7-GRAPH) zdefiniować własne argumenty

Tabela 1. Parametry wejściowe i wyjściowe sekwencyjnego bloku funkcyjnego

| Nazwa | Typ danych | Opis |
|---------------------|------------|--|
| Parametry wejściowe | | |
| OFF_SEQ | Bool | Wyłączenie sekwencji. |
| INIT_SEQ | Bool | Załączenie kroku inicjującego. |
| ACK_EF | Bool | Potwierdzenie wszystkich błędów i ostrzeżeń sekwencji. |
| S_PREV | Bool | Tryb automatyczny: W przypadku aktywności więcej niż jednego kroku w danej chwili wyświetla numer poprzedniego kroku w parametrze S_NO Tryb manualny: Dekrementacja wartości parametru wyjściowego S_NO |
| S_NEXT | Bool | Tryb automatyczny: W przypadku aktywności więcej niż jednego kroku w danej chwili wyświetla numer kolejnego kroku w parametrze S_NO Tryb manualny: Inkrementacja wartości sygnału wyjściowego S_NO |
| SW_AUTO | Bool | Załączenie trybu automatycznego, który pozwala na przejście do kolejnego kroku tylko po spełnieniu warunków tranzycji. |
| SW_TAP | Bool | Załączenie trybu, który pozwala na przejście do kolejnego kroku po spełnieniu warunków tranzycji oraz po równoczesnym zarejestrowaniu zbocza narastającego na parametrze wejściowym T_PUSH. |
| SW_TOP | Bool | Załączenie trybu, który pozwala na przejście do kolejnego kroku po spełnieniu warunków tranzycji lub po zarejestrowaniu zbocza narastającego na parametrze wejściowym T_PUSH |
| SW_MAN | Bool | Załączenie trybu ręcznego. |
| S_SEL | Int | Wybór numeru kroku „S_NO” w trybie ręcznym, który zostanie wyłączony lub załączony za pomocą S_ON, S_OFF. |
| S_ON | Bool | Włączenie kroku o wybranym w S_SEL numerze w trybie ręcznym. |
| S_OFF | Bool | Wyłączenie kroku o wybranym w S_SEL numerze w trybie ręcznym.. |
| T_PUSH | Bool | Zezwolenie na przejście do kolejnego kroku w trybach TAP i TOP. |

| Parametry wyjściowe | | |
|---------------------|------|---|
| S_NO | Int | Wyświetlenie numeru wykonywanego kroku. |
| S_MORE | Bool | Informacja, że więcej niż jeden krok jest aktywny. |
| S_ACTIVE | Bool | Informacja, że wyświetlony numer kroku w S_NO jest aktywny. |
| ERR_FLT | Bool | Obecność błędu blokady lub nadzoru. |
| AUTO_ON | Bool | Informacja, że załączony jest tryb automatyczny. |
| TAP_ON | Bool | Informacja, że załączony jest tryb TAP. |
| TOP_ON | Bool | Informacja, że załączony jest tryb TOP. |
| MAN_ON | Bool | Informacja, że załączony jest tryb manualny. |

Na sekwencje składają się kroki oraz tranzycje (warunki przejścia do kolejnych kroków). Akcje do wykonania w poszczególnych krokach określone są za pomocą kwalifikatorów działania. Podstawowe kwalifikatory działania przedstawiono w tabeli 2.

Tabela 2. Podstawowe kwalifikatory w języku SFC

| Kwalifikator | Działanie |
|--------------|---|
| <i>N</i> | Akcja jest wykonywana tylko wtedy, gdy aktywny jest krok. |
| <i>S</i> | Ustaw (<i>set</i>) - akcja będzie wykonywana, nawet jeśli krok stanie się nieaktywny. |
| <i>R</i> | Akcja, która została ustawiona w innym kroku zostanie skasowana. |
| <i>L</i> | Ograniczony w czasie (<i>time Limited</i>) - akcja wykonywana jest przez, z góry zadany czas, chyba, że krok stanie się nieaktywny. |
| <i>D</i> | Opóźniony w czasie (<i>time Delayed</i>) - akcja wykonana jest po określonym czasie, chyba że krok stanie się nieaktywny. |

5 Przygotowanie „szablonu” FB w języku GRAPH do realizacji przykładów

Każda sekwencja może być wykonywana tylko i wyłącznie, gdy na wejściu do bloku funkcyjnego (EN) jest doprowadzony stan wysoki (TRUE).

Stąd przygotuj na panelu HMI 6 przycisków, które będą aktywowały jedną z sześciu sekwencji przykładowych.

Dla odmiany proponuję wykonanie zestawu „przycisków radiowych” poprzez wykorzystanie zmiennej typu Word (16 bitów, ustawienie jednego z 16 bitów poprzez wpisanie do zmiennej wagi na pozycji aktywowanego bitu) do której wpisywane będą wartości: 1, 2, 4,.. 32 (jeden z 6 młodszych bitów).

Dodaj DB i nazwij go "V", zdefiniuj tam zmienną:

| | | |
|-------------|------|------|
| Static | | |
| EN_FB_graph | Word | 16#0 |

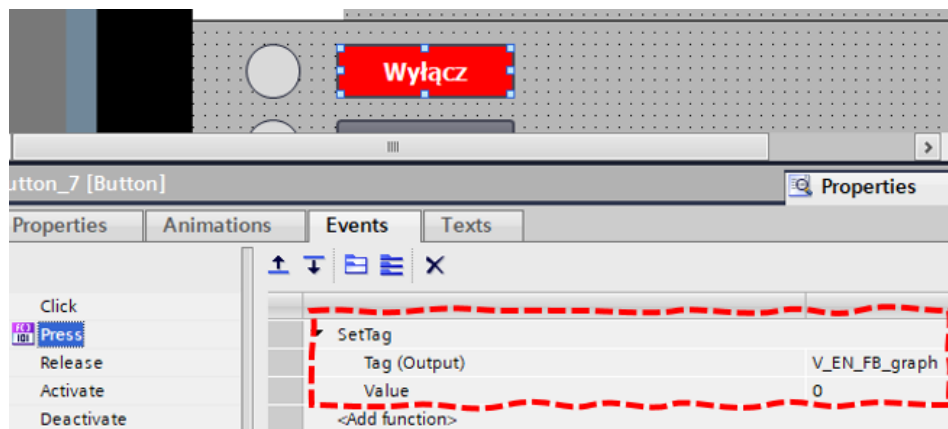
5.1 Przygotowanie panelu HMI do wyboru jednego z oprogramowanych przykładów

Na panelu HMI dodaj dwa przyciski i dwa sygnalizatory (Rys. 9):



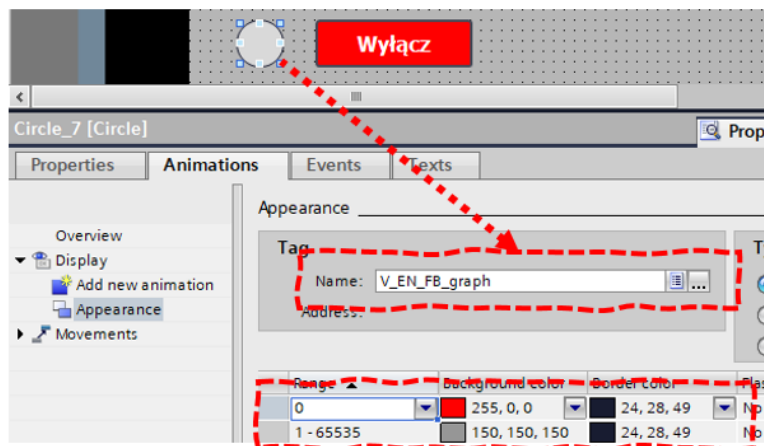
Rys. 9. Dwa przyciski i sygnalizatory

Konfiguracja przycisku „Wyłącz” (Rys. 10)



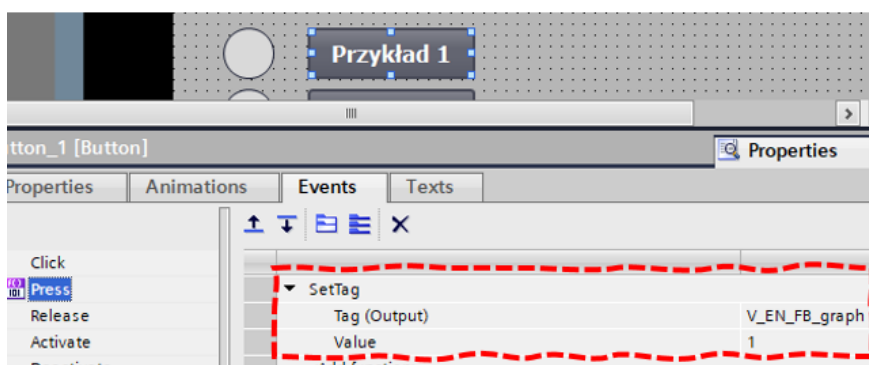
Rys. 10. Konfiguracja obsługi zdarzenia przycisku "Wyłącz"

Konfiguracja sygnalizatora na lewo od przycisku „Wyłącz” (Rys. 11)



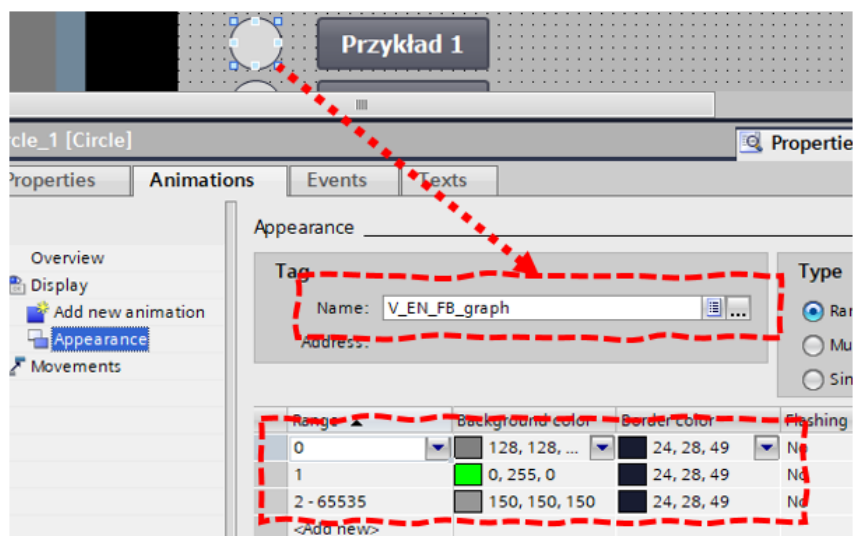
Rys. 11. Konfiguracja animacji pierwszego sygnalizatora

Konfiguracja przycisku „Przykład 1” (Rys. 12)



Rys. 12. Konfiguracja obsługi zdarzenia przycisku "Przykład 1"

Konfiguracja sygnalizatora na lewo od przycisku „Przykład 1” (Rys. 13)



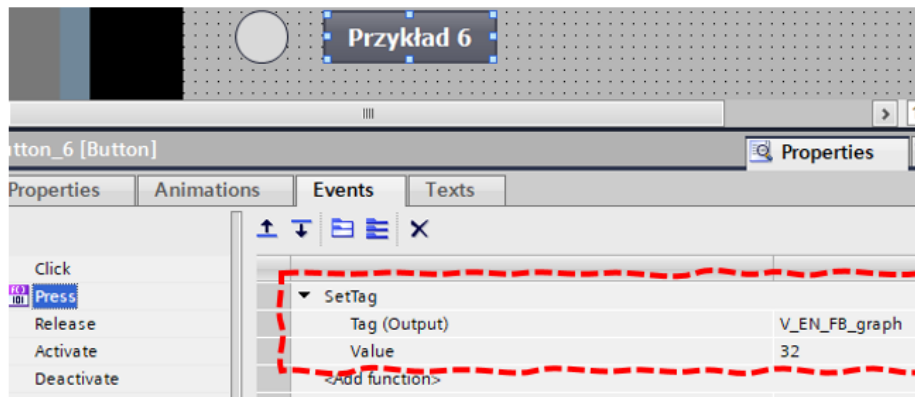
Rys. 13. Konfiguracja animacji sygnalizatora wyboru "Przykład 1"

Skopiuj przycisk "Przykład 1" wraz z sygnalizatorem tak, aby była możliwość wyboru jednego z sześciu przypadków (Rys. 14) – trzeba zmienić wartość zmiennej `V_EN_FB_graph` funkcją `SetTag()` przypisaną do obsługi zdarzenia „Press” przycisku.



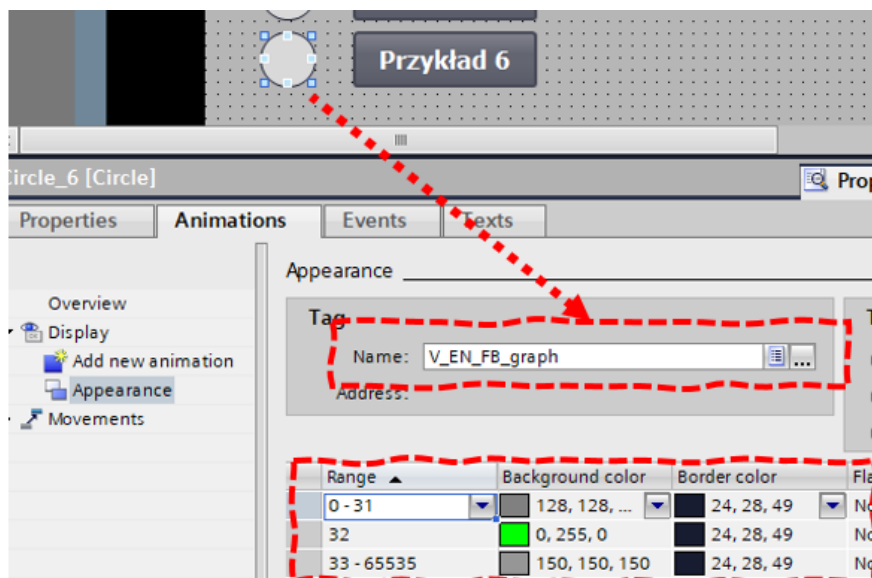
Rys. 14. Przycisk "Przykład 1" skopiowany na 6 przykładów

Dla przykładu konfiguracja przycisku „Przykład 6” (Rys. 15)



Rys. 15. Konfiguracja animacji sygnalizatora wyboru "Przykład 6"

... i konfiguracja sygnalizatora na lewo od przycisku „Przykład 6” (Rys. 16)



Rys. 16. Konfiguracja animacji sygnalizatora wyboru "Przykład 6"

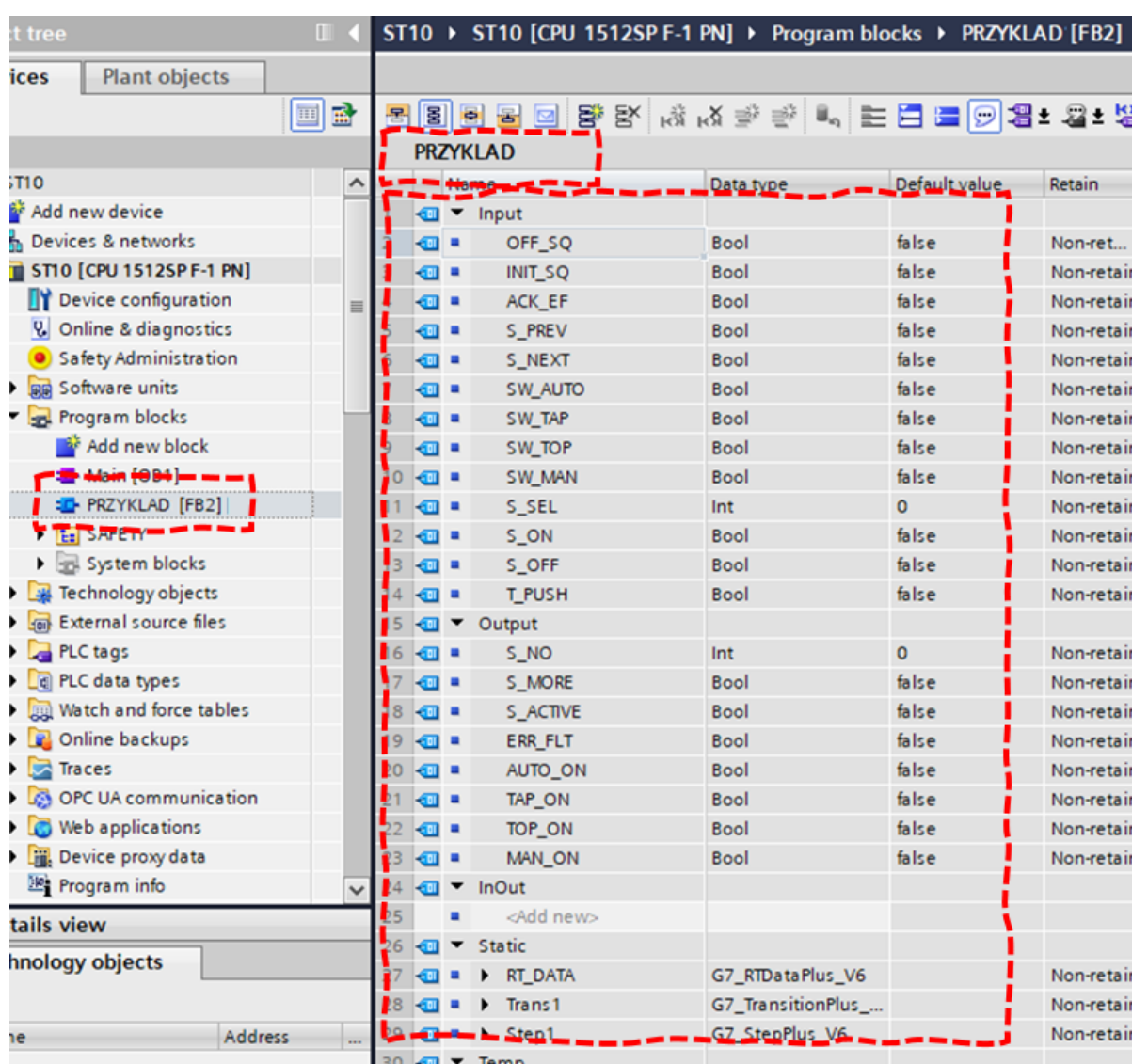
5.2 Przygotowanie szablonu bloku funkcyjnego FB w języku Graph

UWAGA!! W laboratorium jest ogólny dostęp do plików roboczych. Istnieje zatem szansa, że w wyniku działania „sabotażystów” odwrócone zostaną adresy krańcówek (krańcówka dolna ma adres górnej i vice versa), lub zaworów (wysunięcia z wsunięciem). Zatem – ZACHOWAJ CZUJNOŚĆ!! :P

Przygotowanie do wykonania samodzielnych przykładów.

Każdy przykład należy wykonać w innym FB zaprogramowanym w GRAPH

W FB w języku GRAPH podobnie jak w innych FB **NIE UZYWAMY** zmiennych globalnych. Przygotuj sobie FB wzorcowy – potem skopiujesz go 6 razy i będziesz miał miejsce na 6 przykładów których treść zapisano w ostatnim rozdziale instrukcji (Rys. 17).



| Name | Data type | Default value | Retain |
|-----------|-----------------------|---------------|------------|
| Input | | | |
| OFF_SQ | Bool | false | Non-ret... |
| INIT_SQ | Bool | false | Non-retain |
| ACK_EF | Bool | false | Non-retain |
| S_PREV | Bool | false | Non-retain |
| S_NEXT | Bool | false | Non-retain |
| SW_AUTO | Bool | false | Non-retain |
| SW_TAP | Bool | false | Non-retain |
| SW_TOP | Bool | false | Non-retain |
| SW_MAN | Bool | false | Non-retain |
| S_SEL | Int | 0 | Non-retain |
| S_ON | Bool | false | Non-retain |
| S_OFF | Bool | false | Non-retain |
| T_PUSH | Bool | false | Non-retain |
| Output | | | |
| S_NO | Int | 0 | Non-retain |
| S_MORE | Bool | false | Non-retain |
| S_ACTIVE | Bool | false | Non-retain |
| ERR_FLT | Bool | false | Non-retain |
| AUTO_ON | Bool | false | Non-retain |
| TAP_ON | Bool | false | Non-retain |
| TOP_ON | Bool | false | Non-retain |
| MAN_ON | Bool | false | Non-retain |
| InOut | | | |
| <Add new> | | | |
| Static | | | |
| RT_DATA | G7_RTDataPlus_V6 | | Non-retain |
| Trans1 | G7_TransitionPlus_... | | Non-retain |
| Step1 | G7_StepPlus_V6 | | Non-retain |
| Temp | | | |

Rys. 17. Standardowy interfejs FB programowanego w GRAPH (patrz Tabela 1)

Na potrzeby przykładów nie potrzebujesz wszystkich parametrów formalnych (Rys. 18, Rys. 19). Potrzebujesz za to wprowadzić do FB argumenty związane np. z wejściami fizycznymi i/lub zmiennymi

ustawianymi na HMI oraz wyprowadzić wyliczone w FB stany przypisane później do wyjść fizycznych lub sterujących kontrolkami na HMI.

| | Name | Data type | Default value | Retain |
|----|----------|-----------|---------------|------------|
| 1 | Input | | | |
| 2 | OFF_SQ | Bool | false | Non-retain |
| 3 | INIT_SQ | Bool | false | Non-retain |
| 4 | ACK_EF | Bool | false | Non-retain |
| 5 | S_PREV | Bool | false | Non-retain |
| 6 | S_NEXT | Bool | false | Non-retain |
| 7 | SW_AUTO | Bool | false | Non-retain |
| 8 | SW_TAP | Bool | false | Non-retain |
| 9 | SW_TOP | Bool | false | Non-retain |
| 10 | SW_MAN | Bool | false | Non-retain |
| 11 | S_SEL | Int | 0 | Non-retain |
| 12 | S_ON | Bool | false | Non-retain |
| 13 | S_OFF | Bool | false | Non-retain |
| 14 | T_PUSH | Bool | false | Non-retain |
| 15 | Output | | | |
| 16 | S_NO | Int | 0 | Non-retain |
| 17 | S_MORE | Bool | false | Non-retain |
| 18 | S_ACTIVE | Bool | false | Non-retain |
| 19 | ERR_FLT | Bool | false | Non-retain |
| 20 | AUTO_ON | Bool | false | Non-retain |
| 21 | TAP_ON | Bool | false | Non-retain |
| 22 | TOP_ON | Bool | false | Non-retain |
| 23 | MAN_ON | Bool | false | Non-retain |
| 24 | InOut | | | |

Rys. 18. Modyfikacja interfejsu FB napisanego w Graph

Po tej akcji interfejs będzie wyglądał jak poniżej

| | Name | Data type | Default value | Retain |
|----|-----------|-----------------------|---------------|------------|
| 1 | Input | | | |
| 2 | OFF_SQ | Bool | false | Non-retain |
| 3 | INIT_SQ | Bool | false | Non-retain |
| 4 | ACK_EF | Bool | false | Non-retain |
| 5 | Output | | | |
| 6 | S_NO | Int | 0 | Non-retain |
| 7 | S_ACTIVE | Bool | false | Non-retain |
| 8 | ERR_FLT | Bool | false | Non-retain |
| 9 | InOut | | | |
| 10 | <Add new> | | | |
| 11 | Static | | | |
| 12 | RT_DATA | G7_RTDataPlus_V6 | | Non-retain |
| 13 | Trans1 | G7_TransitionPlus_... | | Non-retain |
| 14 | Step1 | G7_StepPlus_V6 | | Non-retain |
| 15 | Temp | | | |
| 16 | <Add new> | | | |

Rys. 19. Interfejs FB w Graph po pierwszej modyfikacji

Teraz zaglądaj do swojej tablicy tagów i na jej podstawie uzupełnij listę argumentów w interfejsie wzorcowego FB – tutaj na przykładzie stanowiska 10 na każdym stanowisku będzie trochę inaczej.

W jaki sposób szybko uzupełnić argumenty w interfejsie FB na podstawie tablicy tagów?

Wejść do tablicy tagów i kliknij na nagłówku kolumny *Address* – posortujesz tagi po adresie (Rys. 20, obszary *I* i *Q* będą odseparowane).

| Name | Data type | Address | Retain | Action |
|---------------|-----------|---------|--------------------------|--------|
| StacyjkaPrawo | Bool | %I0.0 | <input type="checkbox"/> | |

Rys. 20. Sortowanie tablicy tagów po kolumnie *Address*

Zaznacz wartości w kolumnach *Name* i *Data type* w obszarze *I*. Skopiuj zaznaczone do schowka (Rys. 21)

| Name | Data type | Address |
|------------------|-----------|---------|
| StacyjkaPrawo | Bool | %I0.0 |
| StacyjkaLewo | Bool | %I0.0 |
| PrzyciskZielony | Bool | %I0.0 |
| PrzyciskCzerwony | Bool | %I0.0 |
| isStopDown | Bool | %I1.0 |
| isHubPosDown | Bool | %I1.0 |
| isHubPosUp | Bool | %I1.0 |
| isInPosition | Bool | %I1.0 |
| ESTOP | Bool | %I90.0 |

Rys. 21. Zaznaczenie tagów z obszaru *I*

Idź do interfejsu FB i dodaj miejsce na parametr w sekcji *Input* (Rys. 22)

| Name | Data type | Default value | Retain |
|-----------|-----------|---------------|--------|
| Input | | | |
| OFF_SQ | Bool | false | No |
| INIT_SQ | Bool | false | No |
| ACK_EF | Bool | false | No |
| <Add new> | | | |

Rys. 22. Interfejs FB przed wklejeniem tagów skopiowanych z tablicy tagów

Wklej nazwy i typy skopiowane z tablicy tagów (Rys. 23). Będziemy mieć takie same nazwy parametrów formalnych jak i argumentów wywołania FB. Nie ma tu jednak żadnej kolizji – jak wiemy to są dwie różne rzeczy.

| PRZYKŁAD | | | |
|----------|------------------|-----------|---------------|
| | Name | Data type | Default value |
| | ▼ Input | | |
| | OFF_SQ | Bool | false |
| | INIT_SQ | Bool | false |
| | ACK_EF | Bool | false |
| | StacyjkaPrawo | Bool | false |
| | StacyjkaLewo | Bool | false |
| | PrzyciskZielony | Bool | false |
| | PrzyciskCzerwony | Bool | false |
| | isStopDown | Bool | false |
| | isHubPosDown | Bool | false |
| | isHubPosUp | Bool | false |
| | isInPosition | Bool | false |

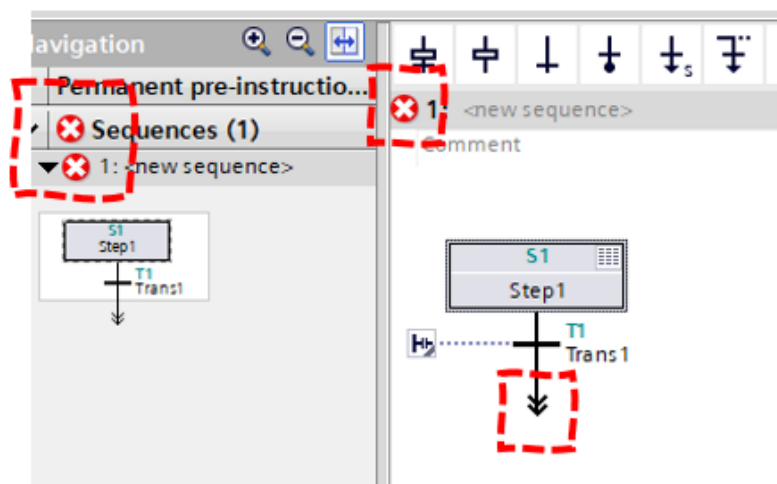
Rys. 23. Interfejs FB po wklejeniu tagów z obszaru I tablicy tagów

Zrób to samo dla sekcji Output (Rys. 24)

| PRZYKŁAD | | | | |
|----------|-----------------|-----------|---------------|------------|
| | Name | Data type | Default value | Retain |
| | ▶ Input | | | |
| | ▼ Output | | | |
| | S_NO | Int | 0 | Non-retain |
| | S_ACTIVE | Bool | false | Non-retain |
| | ERR_FLT | Bool | false | Non-retain |
| | GreenLED | Bool | false | Non-ret... |
| | RedLED | Bool | false | Non-retain |
| | PrzenosnikStart | Bool | false | Non-retain |
| | PrzenosnikSTOP | Bool | false | Non-retain |
| 0 | StopDown | Bool | false | Non-retain |
| 1 | HubPosDown | Bool | false | Non-retain |
| 2 | HubPosUp | Bool | false | Non-retain |
| 3 | ▼ InOut | | | |

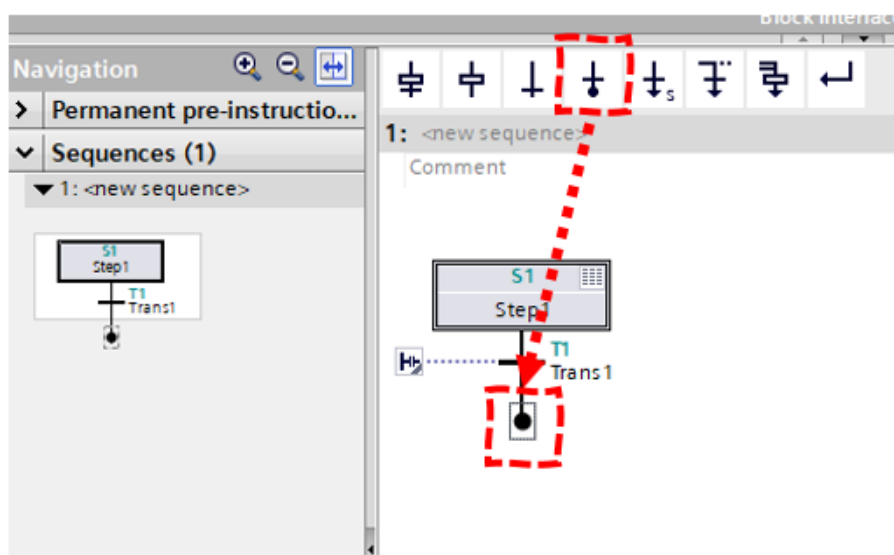
Rys. 24. Interfejs FB po wklejeniu tagów z obszaru Q tablicy tagów

Kod przygotowanego bloku jest niezakończony – błędy składniowe (Rys. 25)



Rys. 25. Standardowa zawartość nowo utworzonego bloku FB w języku Graph – zawiera błędy

Najprostszym sposobem usunięcia błędu składniowego (sekwencja nie będzie wykonywała żadnych akcji) jest dołożenie na końcu grafu symbolu zakończenia sekwencji (Rys. 26).



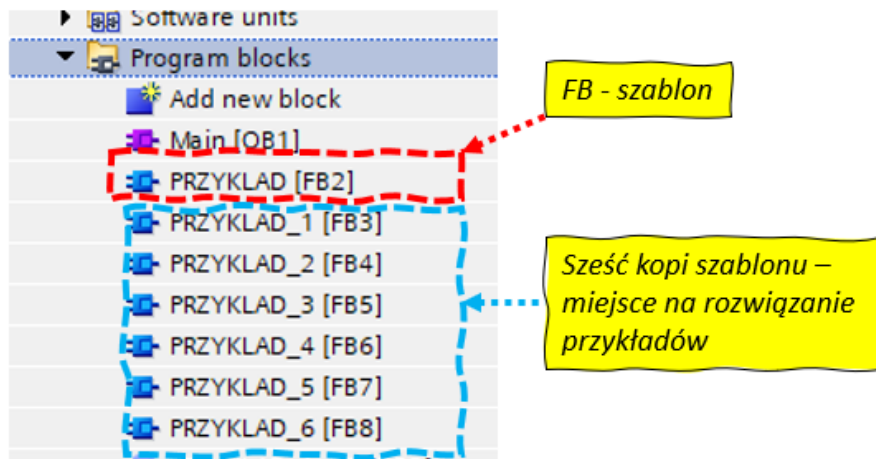
Rys. 26. Wstawienie symbolu zakończenia sekwencji - tylko w celu zachowania poprawności składniowej

5.3 Sześciokrotne skopiowanie FB PRZYKLAD pod nazwami PRZYKLAD_1 do PRZYKLAD_6

Przygotowany FB posłuży jako szablon do samodzielnego rozwiązania sześciu przykładów. Dlatego sześciokrotnie skopiuj przykład (Rys. 27).

Przy kopiowaniu zwracaj uwagę na to aby nie powtórzyły się numery (adresy) FB – każdy FB powinien mieć inny adres (numer).

UWAGA – Twoje numery FB nie muszą być takie same jak na przykładowym rysunku

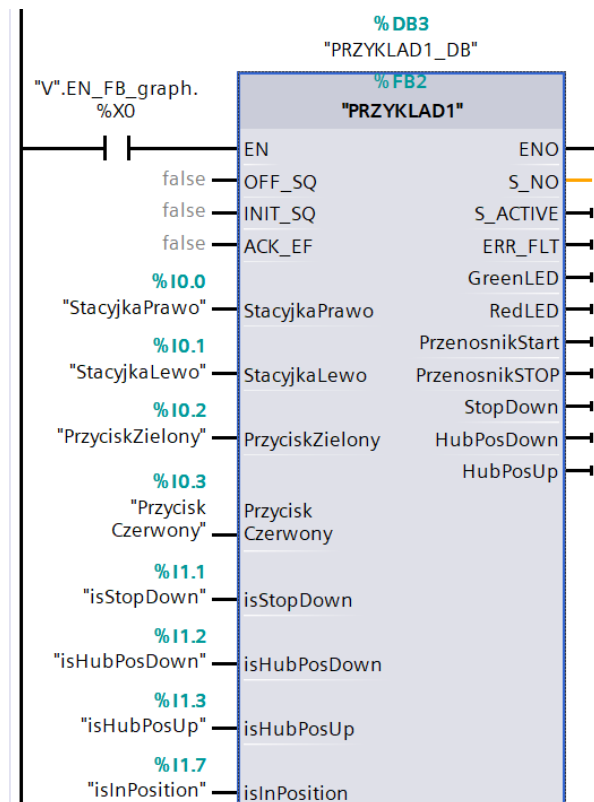


Rys. 27. Skopiowane sześciokrotnie FB

5.4 Wywołanie skopiowanych bloków w OB1

Zamiast kopiować FB „PRZYKLAD” można go sześciokrotnie wywołać – tutaj będziemy jednak zmieniać interfejs FB i dlatego będziemy potrzebować osobną kopię dla każdego przykładu.

Aby ułatwić sobie wywołanie w OB1 wszystkich sześciu FB wywołaj najpierw FB „PRZYKLAD_1” i uzupełnij argumenty po stronie wejść – przykład poniżej (Rys. 28).



Rys. 28. Wywołanie FB, uzupełnienie warunku wykonywalności (EN) i argumentów wejściowych bloku

Wielokrotne przypisanie wejść fizycznych do wejść FB (6 przykładów) nie będzie stanowiło problemu. Nie można natomiast (nie jest to sygnalizowane jako błąd) w standardowym programie wielokrotnie przypisać wartości tej samej zmiennej.

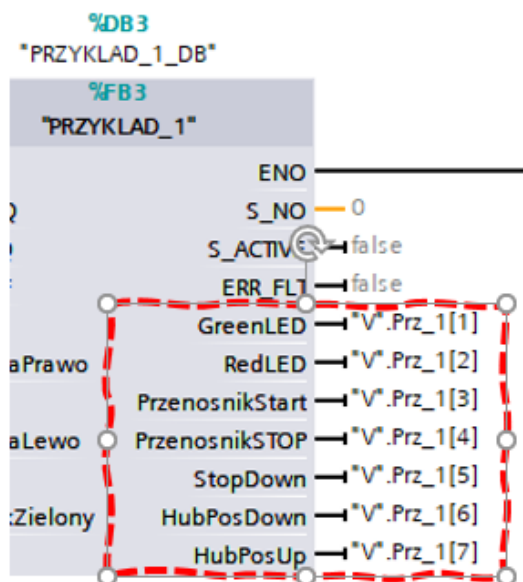
Ostatecznie w takim przypadku zostanie ustawiona ostatnio przypisana wartość (wartość z „network” o największym numerze). Dlatego dla każdego z 6 przykładów zadeklarowano w DB „V” tablicę zmiennych typu bool (Rys. 29).

| Prz_1 | | Array[0..7] of Bool | |
|----------|---------------------|---------------------|--|
| Prz_1[0] | Bool | false | |
| Prz_1[1] | Bool | false | |
| Prz_1[2] | Bool | false | |
| Prz_1[3] | Bool | false | |
| Prz_1[4] | Bool | false | |
| Prz_1[5] | Bool | false | |
| Prz_1[6] | Bool | false | |
| Prz_1[7] | Bool | false | |
| Prz_2 | Array[0..7] of Bool | | |
| Prz_3 | Array[0..7] of Bool | | |
| Prz_4 | Array[0..7] of Bool | | |
| Prz_5 | Array[0..7] of Bool | | |
| Prz_6 | Array[0..7] of Bool | | |

Rys. 29. Deklaracja ośmioelementowych tablic zmiennych typu bool

I tak dla „PRZYKLAD_1” wykorzystano elementy „V”.Prz_1[1] ... „V”.Prz_1[7] (Rys. 30).

Dla kolejnych przykładów wykorzystaj kolejne elementy tablic.



Rys. 30. Przypisanie elementów tablicy do wyjść FB na przykładzie „PRZYKLAD_1”

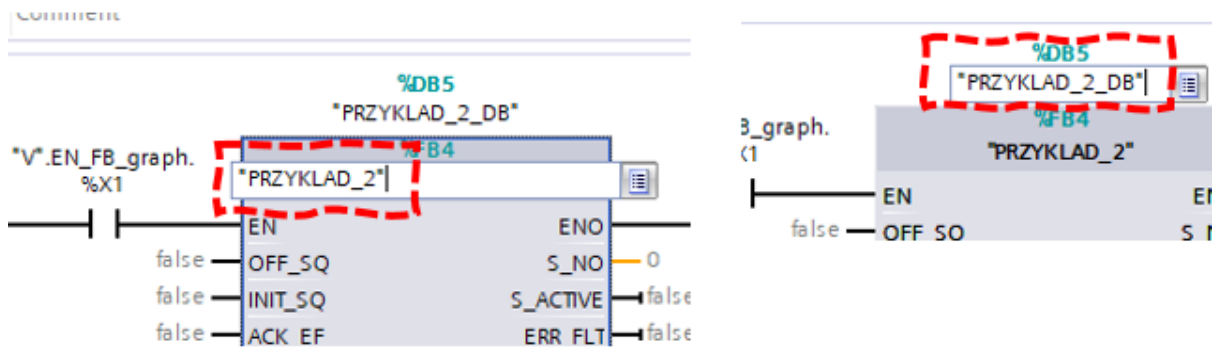
Skopiuj cały „network” z wywołaniem bloku „PRZYKLAD_1” zmieniając:

- adres FB i nazwę zmiennej instancyjnej (Rys. 31),
- elementy tablic przypisanych do wyjść FB (Rys. 32).

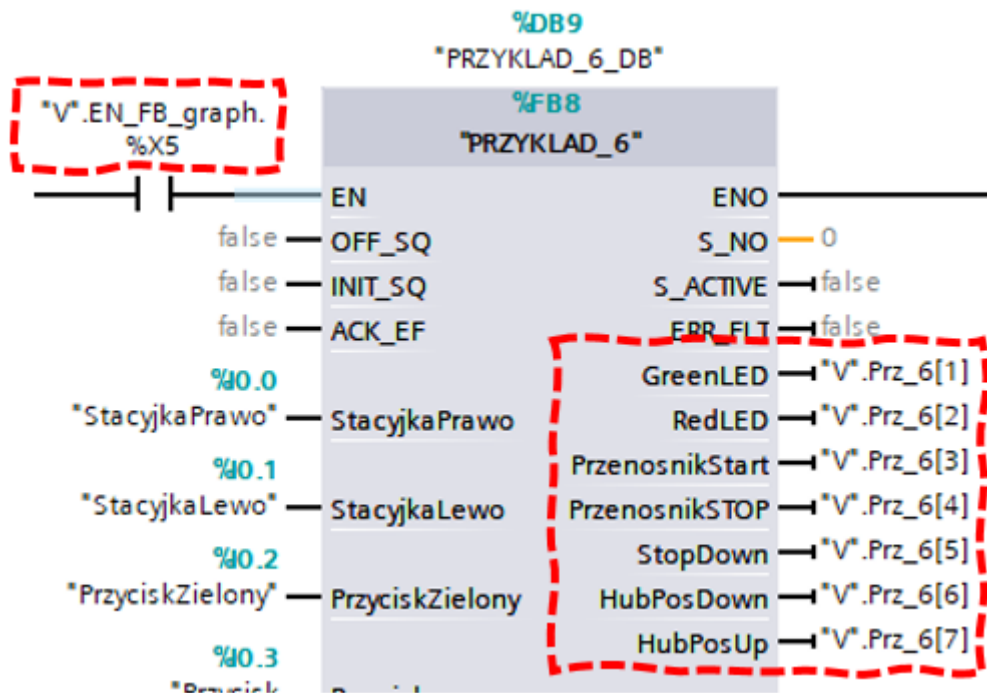
UWAGA!!!

Po skopiowaniu wywołania FB (networki dla kolejnych przykładów) należy zmienić nazwę instancyjnego DB – zostanie on wtedy utworzony.

Rys. 33 - struktura programu po sześciokrotnym skopiowaniu

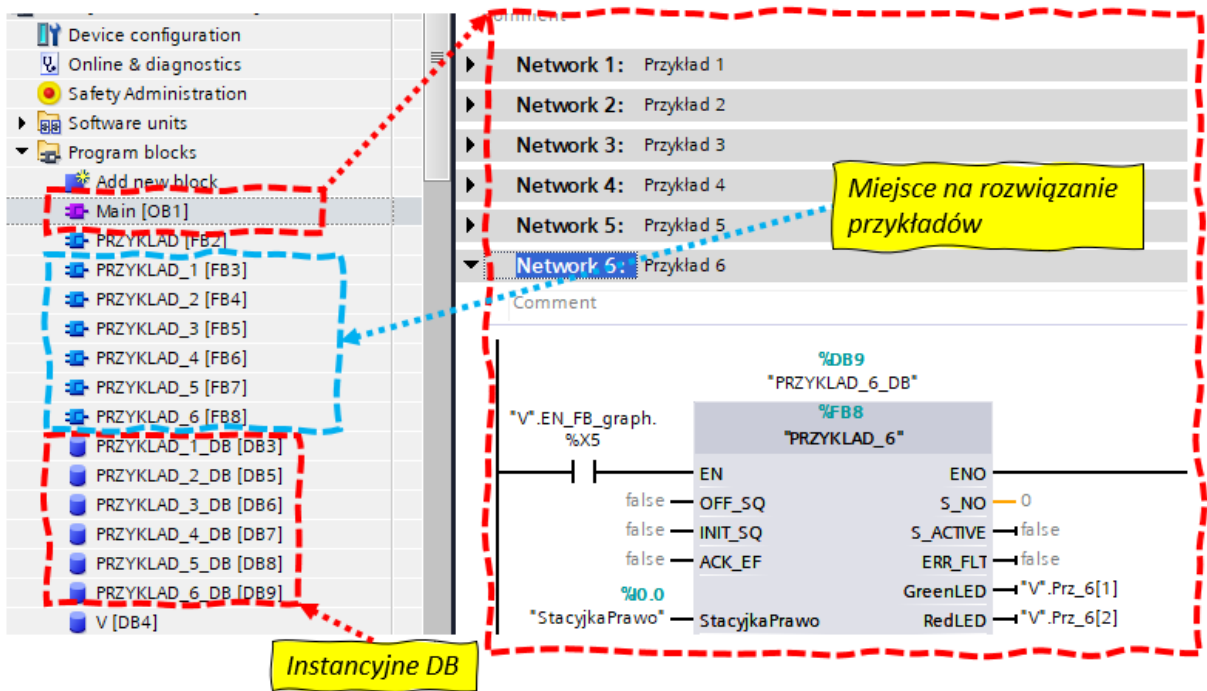


Rys. 31. Przy kopiowaniu FB zmieniaj nazwy FB i instancyjnego DB - tutaj na przykładzie 2



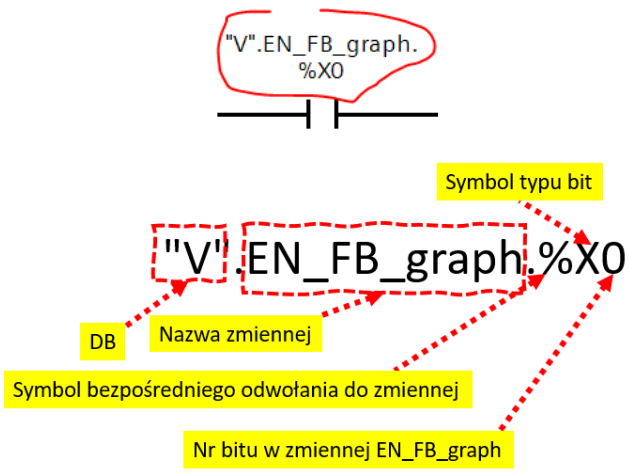
Rys. 32. Modyfikacja elementów tablic przypisanych do wyjść FB i warunku wykonywalności

Struktura programu po sześciokrotnym skopiowaniu FB i utworzeniu instancyjnych DB w czasie ich wywołania w OB1 (Rys. 33).



Rys. 33. Struktura programu po sześciokrotnym wywołaniu bloku FB napisanego w Graph - gotowość do wykonania sześciu samodzielnych przykładów – rozwinięty „PRZYKLAD6”

Co oznacza argument styku z poniższego rysunku – warunek wykonywalności w tym przypadku FB z „Przykład_1” (Rys. 34)



Rys. 34. Odwołanie do zmiennej typu bool przez „slice access”

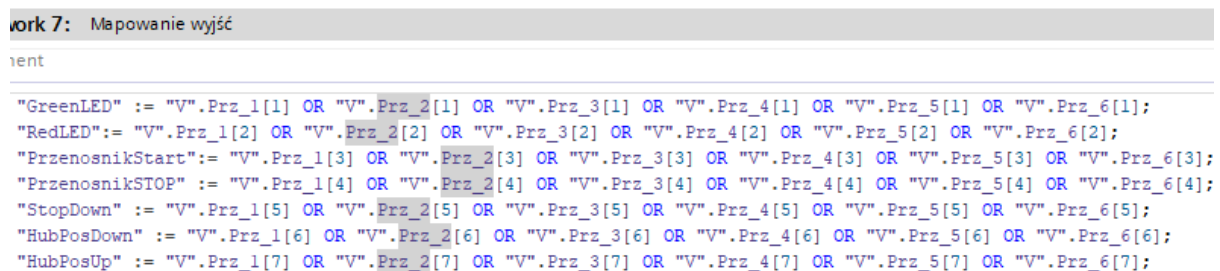
5.5 Mapowanie na wyjścia fizyczne wyjść z sześciu FB przypisanych do elementów tablic

W bloku OB1 dodaj network w języku SCL (tekst strukturalny, Rys. 35)



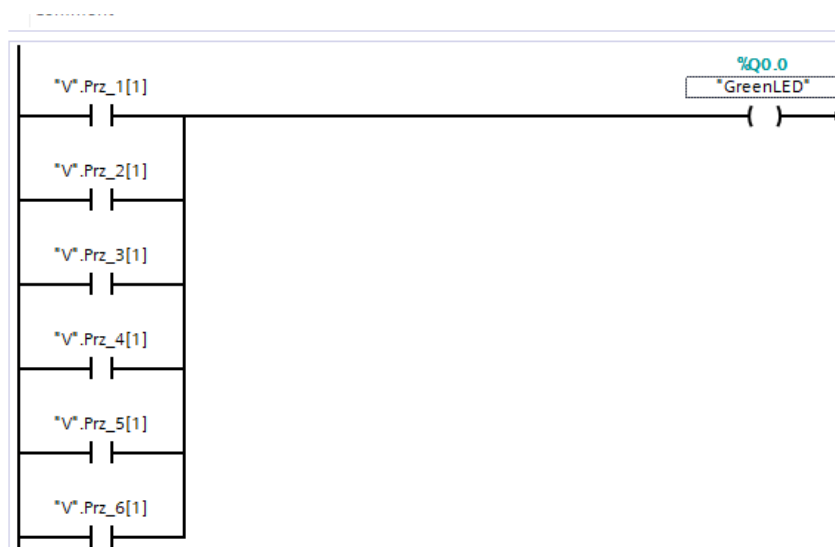
Rys. 35. Dodawanie linii (network) w języku SCL

Aby brać pod uwagę element wybranej tablicy przypisany do wyjścia fizycznego napisz w SCL network mapujący zmienne (elementy tablicy) na wyjścia fizyczne (Rys. 36).



Rys. 36. Mapowanie wyjść fizycznych w języku SCL

Gdyby mapowanie zmiennych na wyjścia fizyczne napisać w języku drabinkowym to należałoby powtórzyć 6 razy konstrukcję z poniższego rysunku (Rys. 37).



Rys. 37. Przykład mapowania jednej linii w LAD - tutaj mapowanie w SCL

6 Przykład rozwiązania – generatory w Graph – niekoniecznie stosuj, ale znaj zasady.

UWAGA: pokazany w tej instrukcji sposób aktywowania sekwensera przez ustawienie TRUE na EN FB jest najprostszym rozwiązaniem. W jego wyniku FALSE na EN powoduje zatrzymanie wykonywania sekwencji, ale nie dezaktywuje bieżącego kroku. Ma to swoje wady i zalety.

Dla pełniejszej kontroli należałoby FB z sekwenserami wywoływać bezwarunkowo, a sekwencją sterować np. przez wejścia OFF_SQ i INIT_SQ – nie są to jedyne wejścia mające wpływ na sterowanie sekwencją

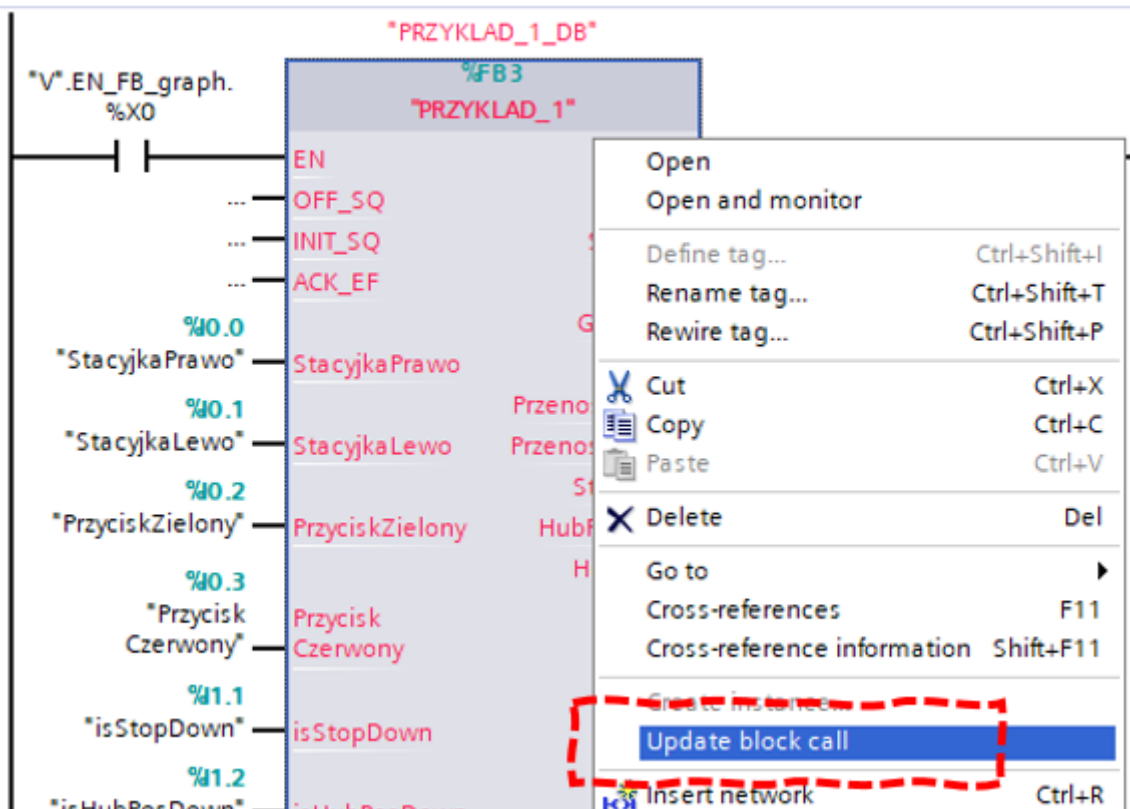
Zadanie: napisz FB który będzie realizował funkcjonalność 3 generatorów o różnych częstotliwościach:

- wybór jednego z 3 wejść FB: StacyjkaPrawo, StacyjkaLewo lub PrzyciskZielony wybiera jeden z generatorów, stany niskie na wszystkich 3 wejściach zeruje wyjścia generatorów,
- czasy połowy okresu poszczególnych generatorów przekaż jako parametry wywołania FB (Rys. 38),
- wynik działania generatorów pokaż na wyjściach FB: GreenLED, RedLED, stopDown,
- każdy generator wykonaj innym sposobem.

| PRZYKLAD_1 | | | | |
|------------|------------------|-----------|---------------|---|
| | Name | Data type | Default value | Retain |
| | Input | | | |
| | OFF_SQ | Bool | false | Non-retain |
| | INIT_SQ | Bool | false | Non-retain |
| | ACK_EF | Bool | false | Non-retain |
| | generator_1 | Bool | false | Non-retain |
| | generator_2 | Bool | false | Non-retain |
| | generator_3 | Bool | false | Non-retain |
| | PrzyciskCzerwony | Bool | false | Non-retain |
| | isStopDown | Bool | false | Non-retain |
| 0 | isHubPosDown | Bool | false | Non-retain |
| 1 | isHubPosUp | Bool | false | Non-retain |
| 2 | isInPosition | Bool | false | Non-ret... <input type="button" value="⌵"/> |
| 3 | czas_1_2_gen_1 | Time | T#0ms | Non-retain |
| 4 | czas_1_2_gen_2 | Time | T#0ms | Non-retain |
| 5 | czas_1_2_gen_3 | Time | T#0ms | Non-retain |

Rys. 38. Modyfikacja interfejsu "PRZYKLAD_1"

Zmiany w interfejsie powodują konieczność uaktualnienia wywołania bloku FB (Rys. 39). Takie uaktualnienie będzie potrzebne zawsze jeżeli zobaczymy czerwoną czcionkę w miejscu wywołania FB.



Rys. 39. Aktualizacja wywołania bloku

Generatory w tym przykładzie postanowiono wykonać jako alternatywne drogi algorytmu.

Dlatego po kroku S1 dodane zostaną trzy gałęzi alternatywne pozwalające na wybór jednej z realizacji (dbajmy o to aby tylko jedna gałąź rozpoczynała się tranzycją w stanie TRUE).

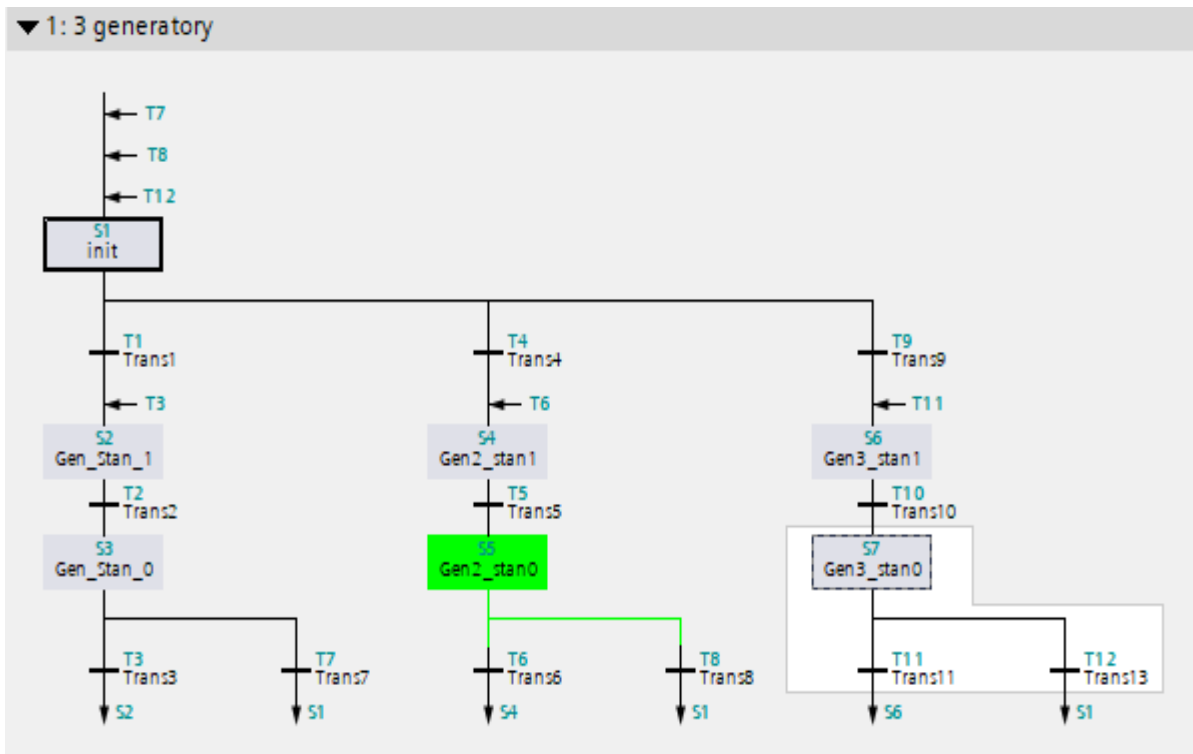
Dwa pierwsze rozwiązania generatorów będą wymagały zastosowania zależności czasowych – akcje: D i L. dlatego w sekcji static interfejsu zdefiniowano zmienne typu bool sterowane tymi akcjami (Rys. 40). Zmienne z 1 na końcu dotyczą odmierzenia czasu stanu wysokiego, a z 0 stanu niskiego.

| Static | |
|---------|------------------|
| T1_1 | Bool |
| T1_0 | Bool |
| T2_1 | Bool |
| T2_0 | Bool |
| RT DATA | G7 RTDataPlus V6 |

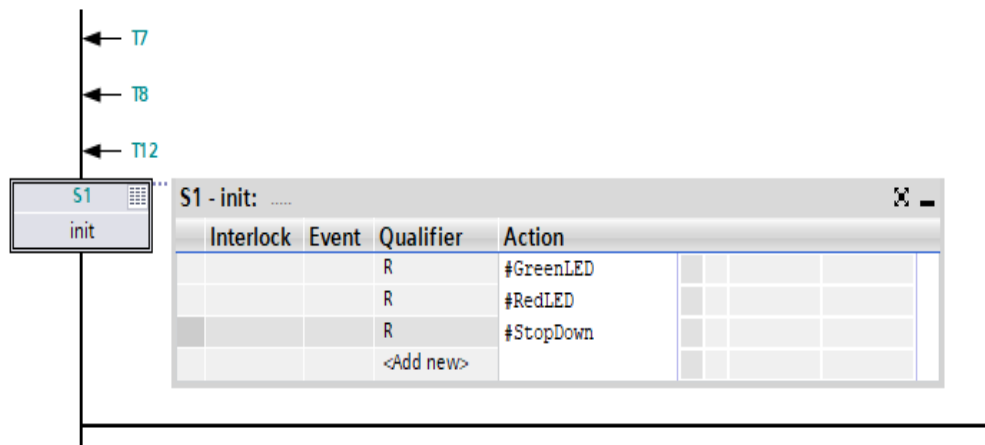
Rys. 40. Zmienne typu bool dla akcji: D i L

Rys. 41 pokazuje całą postać grafu SFC. Szczegóły części wspólnej zobacz na rysunku (Rys. 42). Natomiast poszczególne gałęzi to:

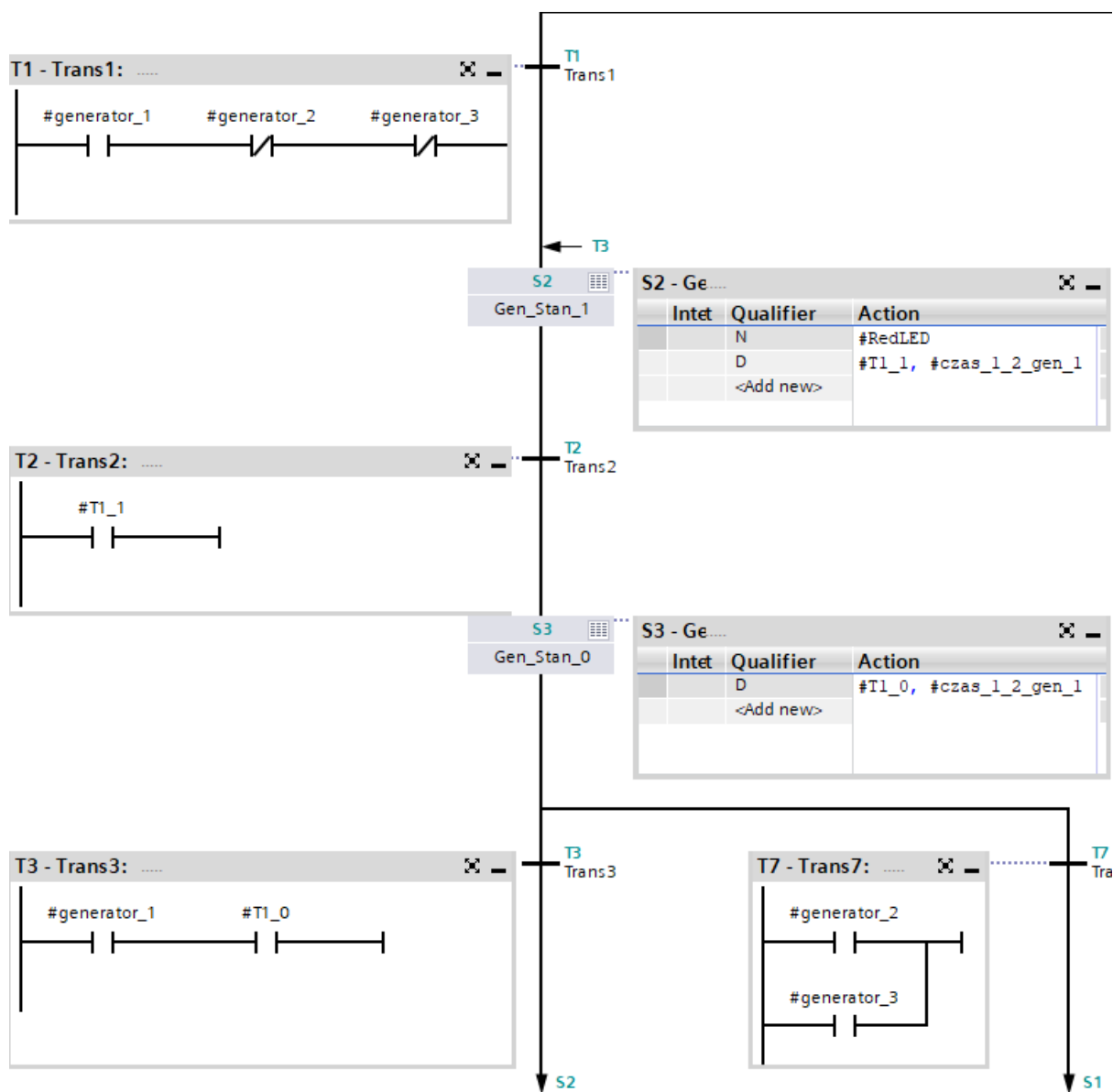
- wykorzystanie akcji N i D (Rys. 43),
- wykorzystanie akcji N i L (Rys. 44),
- wykorzystanie akcji N i odczyt czasu aktywności korku () – zachęcam do dokładniejszego oglądnięcia interfejsu FB programowanego w GRAPH.



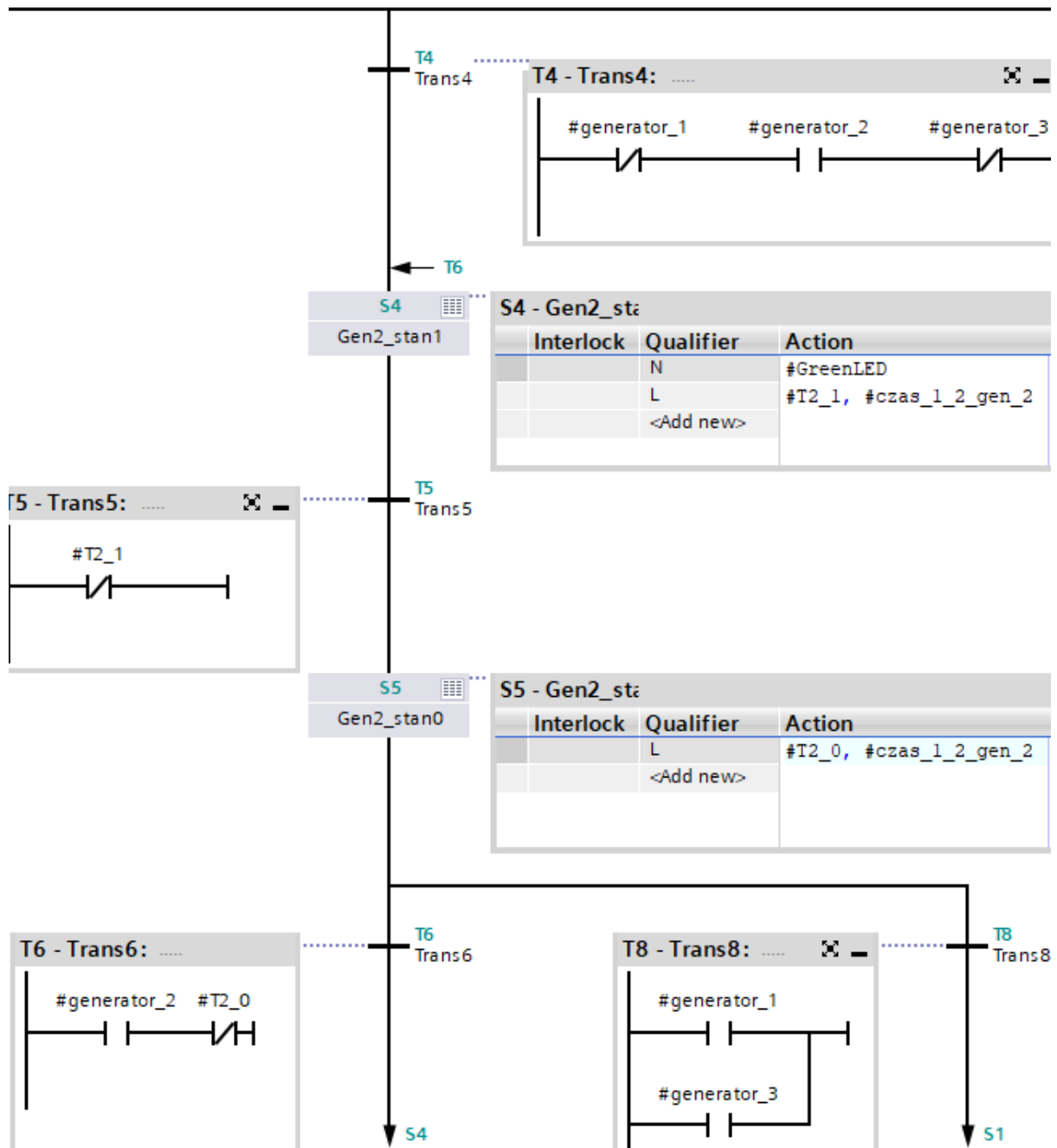
Rys. 41. Graf z trzema alternatywnymi realizacjami generatorów



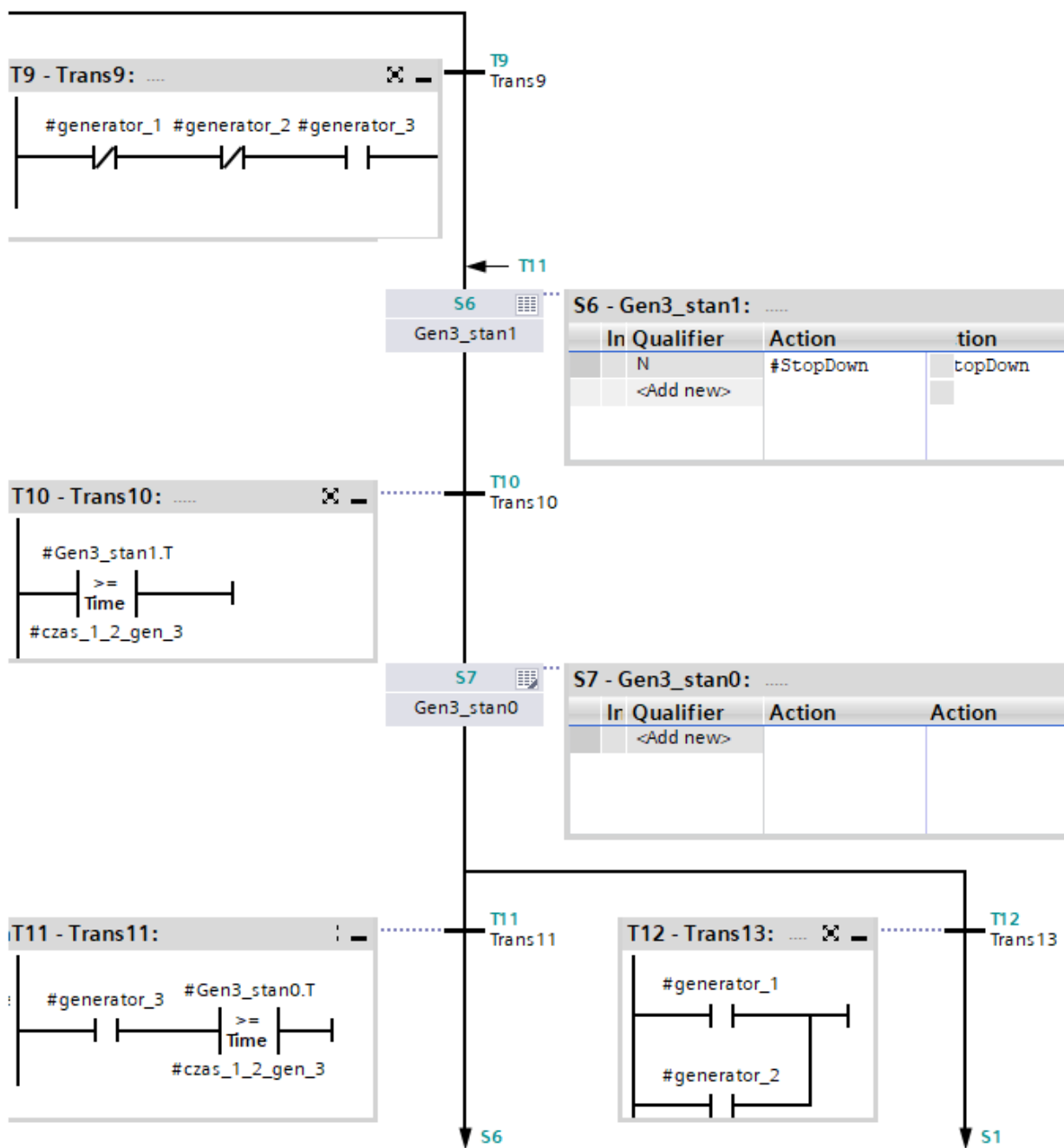
Rys. 42. Szczegóły części wspólnej



Rys. 43. Generator z pierwszej gałęzi



Rys. 44. Generator z drugiej gałęzi



Rys. 45. Generator z gałęzi trzeciej - zwróć szczególną uwagę na tranzycje T10 i T11

7 Zadania do samodzielnej realizacji

Bardzo korzystnym jest podglądanie wykonania sekwensera, poprzez dwukrotne kliknięcie w blok FB danego sekwensera i załączenie ikony monitoringu programu. Aktywne kroki sekwencji podświetlane są na zielono. Możliwe jest też sprawdzenie warunku tranzycji.

7.1 Przykład 1 – sekwencja podstawowa

Przy użyciu stacyjki, przycisków, diod oraz panelu HMI dostępnych na stanowisku zrealizuj sekwencję:

start - wciśnij przycisk zielony,

krok 1 - zaświeć diodę zieloną,

warunek 1 - wciśnij przycisk czerwony,

krok 2 - zaświeć diodę czerwoną, zgaś zieloną

warunek 3 – stacyjka lewo,

krok 3 -zaświeć diodę zieloną, zgaś czerwoną

warunek 3 – stacyjka prawo,

krok 4 - zgaś diodę zieloną,

7.2 Przykład 2 – obsługa jednego siłownika

Przy użyciu stacyjki, przycisków, diod oraz panelu HMI dostępnych na stanowisku zrealizuj sekwencję podaną w sposób lingwistyczny:

Po wciśnięciu przycisku zielonego siłownik się wysuwa, a jak się wysunie, to niech uruchomi diodę i od razu się wsunie. Po wsunięciu dioda gaśnie.

7.3 Przykład 3 – obsługa czasu zadziałania

Przy użyciu stacyjki, przycisków, diod oraz panelu HMI dostępnych na stanowisku zrealizuj sekwencję:

start – stacyjka prawo,

krok 1 – wysuń siłownik,

warunek 1 – siłownik wysunięty,

krok 2 - zaświeć diodę czerwoną

warunek 3 – dioda świeci się przez 5 sekund,

krok 3 -dioda gaśnie, a siłownik się wsuwa

warunek 3 – siłownik się wsunął,

krok 4 - zapal diodę zieloną na 5 sekund,

7.4 Przykład 4 – obsługa wszystkich aktuatorów

Przy użyciu stacyjki, przycisków, diod oraz panelu HMI dostępnych na stanowisku zrealizuj sekwencję:

Po przekręceniu stacyjki w lewo pierwszy siłownik zaczyna się wysuwać. Jak osiągnie pozycję końcową uruchomi się urządzenie wykonawcze reprezentowane przez zieloną diodę (na 5 sekund). Po zgaśnięciu wysuwa się drugi siłownik i po wysunięciu, na 3 sekundy uruchamia się dzida laserowa (dioda czerwona). Po ugaszeniu laserowego gniewu wsuwa się siłownik drugi, a następnie siłownik pierwszy. Po schowaniu się wszystkich siłowników, na panelu hmi pojawia się na 5 sekund rdost z rysunku 9. Można rozszerzyć swoją sekwencję na więcej siłowników.



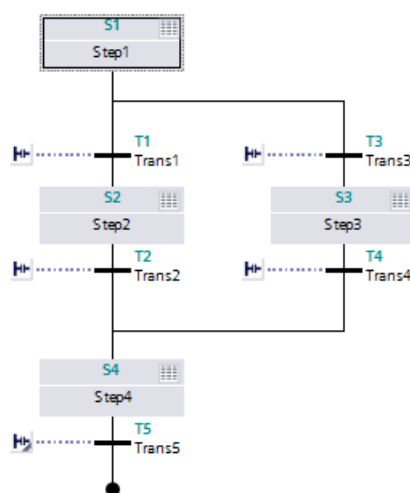
Rys 9. Rdest – wszystko można zrobić z takiego rdestu – wąsy, czapeczkę, pingle :P

7.5 Przykład 5- alternatywa

Przy użyciu stacyjki, przycisków, diod oraz panelu HMI dostępnych na stanowisku zrealizuj sekwencję:

Start – dodatkowy przycisk na hmi
krok 1 - zaświeć diodę czerwoną i zieloną,
warunek 1 - wciśnij przycisk zielony,
warunek 3 - wciśnij przycisk czerwony,
krok 2 - zgaś diodę zieloną po 3 sekundach,
wysuń siłownik 1
krok 3 - zgaś diodę czerwoną na 2 sekundy, ,
wysuń siłownik 1
warunek 2 - przekręć stacyjkę w prawo,
warunek 4 - przekręć stacyjkę w lewo,
krok 4 – wsuń siłowniki,
warunek 5 - wsunięte.
koniec - zgaś diody,

Uwaga – rysunek może różnić się od wynikowej sekwencji



7.6 Przykład 6- współbieżność

Przy użyciu stacyjki, przycisków, diod oraz panelu HMI dostępnych na stanowisku zrealizuj współbieżną sekwencję według pomysłu własnego lub prowadzącego.