Politechnika Rzeszowska
Wydział Elektrotechniki i Informatyki
Katedra Informatyki i Automatyki

Laboratorium

Sterowanie procesami dyskretnymi

Stanowisko 2

Implementacja sieci Grafcet w sterowniku TSX Micro

Rzeszów 2012

W skład stanowiska laboratoryjnego wchodzą: obiekt sterownia (układ trzech wagoników), sterownik TSX Micro 37-22 grupy Schneider oraz komputer klasy PC z zainstalowanym oprogramowaniem PL7 Pro. Wygląd stanowiska przedstawiony jest na fotografii poniżej.

Fot. 1. Widok stanowiska laboratoryjnego

1. Obiekt sterowania

Obiektem sterownia są trzy modele lokomotyw spalinowych, które poruszają się po trzech niezależnych odcinkach torów kolejki elektrycznej umieszczonych w gablocie. Wykorzystano dwa modele lokomotyw spalinowych V36 oraz jeden model T334 produkcji niemieckiej firmy „TILLIG BAHN”. Po prawej stronie gabloty znajduje się konsola służąca do sterowania tymi wagonikami. Na konsoli umieszczono przełączniki służące do wyboru trybu pracy (automatyczny/ręczny), przełączniki służące do wyboru kierunku jazdy poszczególnych wagoników (prawo/lewo), pokrętła służące do zmiany prędkości

wagoników oraz zestaw diod informujących nas czy poszczególne wagoniki znajdują się w położeniach krańcowych torów. Każdy tor wyposażony jest w dwa czujniki krańcowe, po jednym na każdym końcu toru.

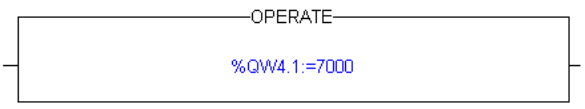
W tabeli 1.1 podano adresy wejść i wyjść sterownika oraz podłączone do tych wejść i wyjść sygnały z czujników i sygnały sterujące.

Tabela 1.1. Adresy wejść i wyjść wykorzystanych w sterowniku

Adres	Opis
%I1.0	Czujnik krańcowy toru 1 znajdujący się po lewej stronie
%I1.1	Czujnik krańcowy toru 1 znajdujący się po prawej stronie
%I1.2	Czujnik krańcowy toru 2 znajdujący się po lewej stronie
%I1.3	Czujnik krańcowy toru 2 znajdujący się po prawej stronie
%I1.4	Czujnik krańcowy toru 3 znajdujący się po lewej stronie
%I1.5	Czujnik krańcowy toru 3 znajdujący się po prawej stronie
%Q2.0	Zmiana kierunku jazdy wagonika nr 1
%Q2.1	Zmiana kierunku jazdy wagonika nr 2
%Q2.2	Zmiana kierunku jazdy wagonika nr 3
%QW4.0	Zmiana prędkości jazdy wagonika nr 1
%QW4.1	Zmiana prędkości jazdy wagonika nr 2
%QW4.2	Zmiana prędkości jazdy wagonika nr 3

Gdy wartość sygnału na wyjściu %Q2.x będzie 1 to wagonik będzie poruszał się w prawo, natomiast gdy będzie 0 wagonik będzie jechał w lewo.

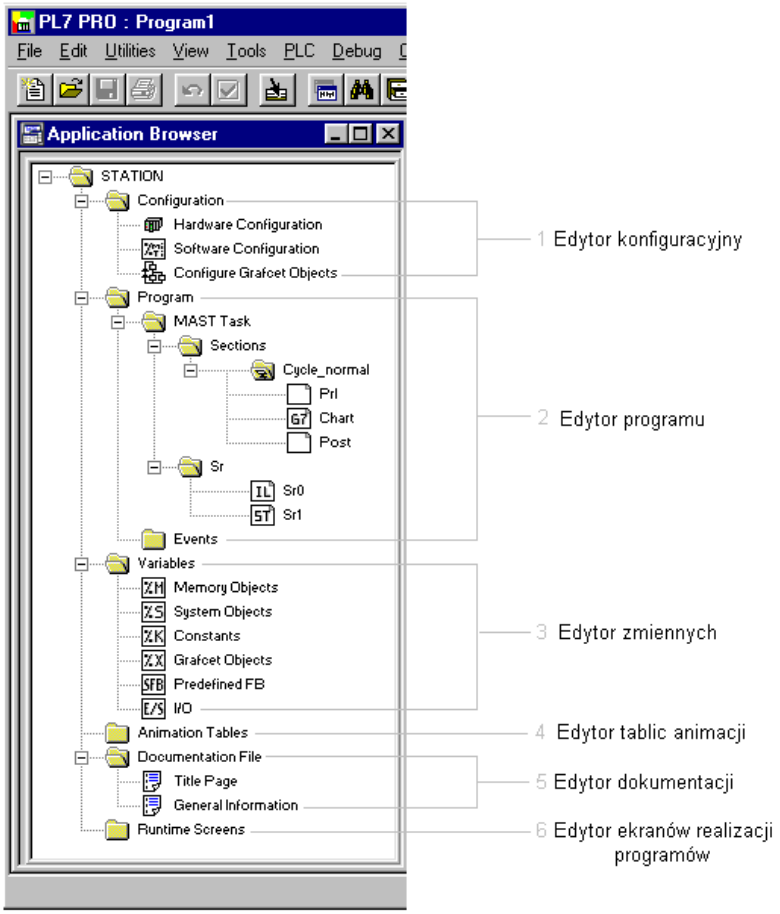
Prędkością wagonika steruje się za pomocą wyjścia %QW4.x poprzez wpisanie wartości od 0 do 10000 (w praktyce, aby wagonik ruszył należy podać wartość przynajmniej 4000, ze względu na duże opory toczenia wagonika). Aby zmienić prędkość wagonika należy użyć bloczku „OPERATE”. Sposób ten jest przedstawiony na rysunku 1.1.



Rys. 1.1. Nadawanie prędkości wagonikowi.

2. Oprogramowanie PL7

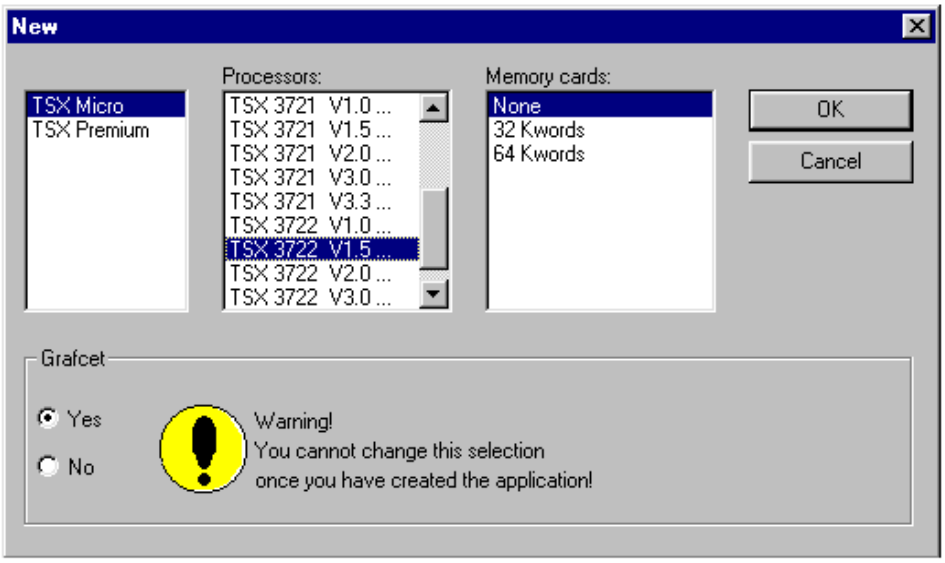
Oprogramowanie PL7 Pro jest przeznaczone dla sterowników Micro (TSX 37xx) i Premium (TSX 57xx). Dostęp do wszystkich narzędzi programowania i uruchamiania jest uzyskiwany przez tzw. przeglądarkę aplikacji (rys. 2.1).



Rys. 2.1. Przeglądarka aplikacji.

Pozwala ona na ogólny przegląd programu i daje możliwość natychmiastowego dostępu do wszystkich elementów aplikacji, a mianowicie: konfiguracji, programu, zmiennych, dokumentacji, tablic animacji itp.

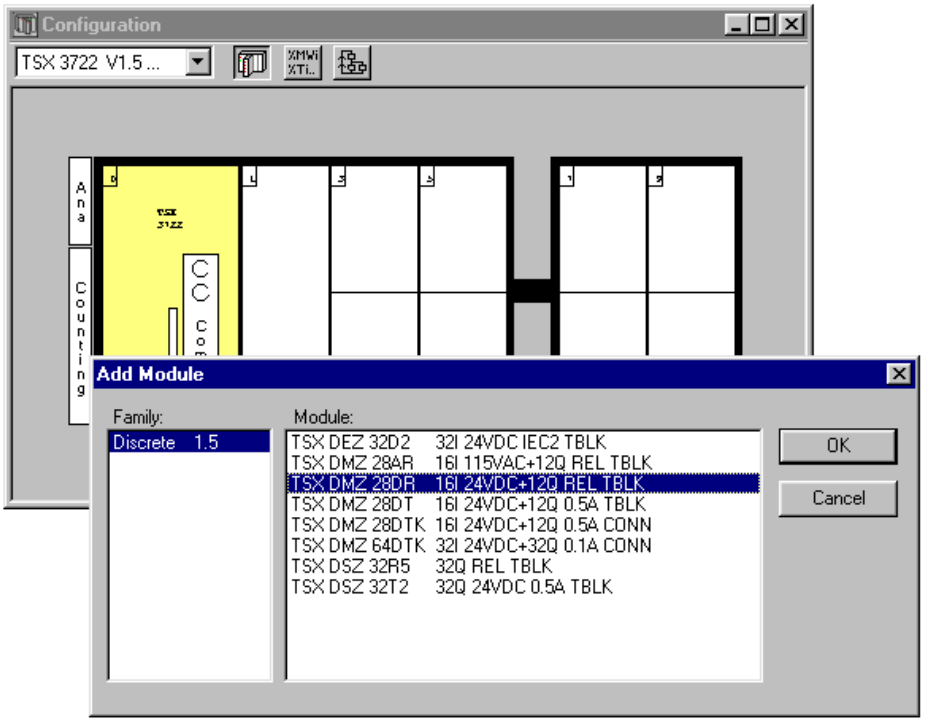
Tworząc nowe oprogramowanie należy wybrać sterownik, który chcemy zaprogramować. W naszym przypadku jest to TSX 3722 (Rys.2.2).



Rys. 2.2. Wybór sterownika

Edytor konfiguracyjny składa się z trzech pozycji (rys. 2.3): konfiguracji sprzętowej, oprogramowania i obiektów języka *Grafcet*. Pozwala on na intuicyjne i graficzne deklarowanie różnorodnych składników aplikacji sterowników Micro/Premium.

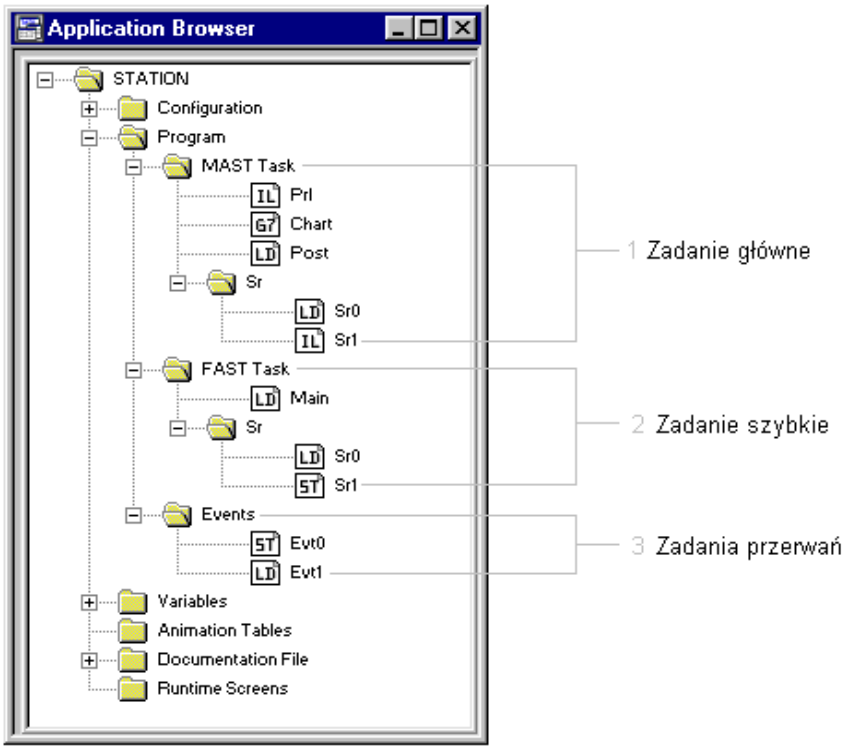
Dla konfiguracji sprzętowej wystarczy kliknąć na nieskonfigurowaną pozycję, a wyświetlacz bloku dialogowego pokaże dostępne moduły we/wy, klasyfikowane według rodzin.



Rys. 2.3. Konfiguracja sprzętowa – dobór we/wy

W tym przypadku jest to moduł wejść/wyjść dyskretnych *TSX DMZ 28DR* (16 wejść, 12 wyjść). W konfiguracji oprogramowania można ustawiać parametry oprogramowania aplikacji: wybór liczby stałych, liczby słów wewnętrznych oraz liczby bloków każdego typu. Natomiast konfiguracja obiektów języka *Grafcet* polega na zdefiniowaniu obiektów takich jak: kroki, makrokroki itp. oraz parametrów wykonawczych (np. liczbę kroków).

Edytor programu pozwala na zorganizowanie programu w formie struktury jedno- lub wielozadaniowej. Struktura jednozadaniowa jest to uproszczona struktura oferowana standardowo, gdzie wykonywane jest zadanie podstawowe zawierające program główny składający się z kilku segmentów i podprogramów. Struktura wielozadaniowa jest przeznaczona dla aplikacji czasu rzeczywistego o dużej wydajności. Składa się z zadanie główne (*MAST Task*), zadanie szybkie (*FAST Task*) oraz zadania przerwań alarmowych (*Events*), które mają najwyższy priorytet (rys. 2.4).



Rys. 2.4. Struktura oprogramowania PL7 Pro

Zadania główne i szybkie podzielone są na segmenty. Taki podział na segmenty pozwala na tworzenie programu strukturalnego oraz na proste tworzenie lub dodawanie nowych modułów programu.

Zadanie główne jest cykliczne lub okresowe (czas wykonania cyklu musi być krótszy niż zadeklarowany okres od 1 do 225 ms). Program główny jest wykonywany i aktywowany systematycznie. Przeznaczony jest do przetwarzania sekwencyjnego. Każdy segment może być programowany z wykorzystaniem schematu drabinkowego, listy rozkazów i tekstu strukturalnego. Tylko jeden segment jest przeznaczony na wykorzystanie języka *Grafcet*, ale oferowane są trzy operacje przetwarzania: przetwarzanie wstępne (*PRL*), przetwarzane sekwencyjne (*CHART*) oraz przetwarzanie końcowe (*POST*).

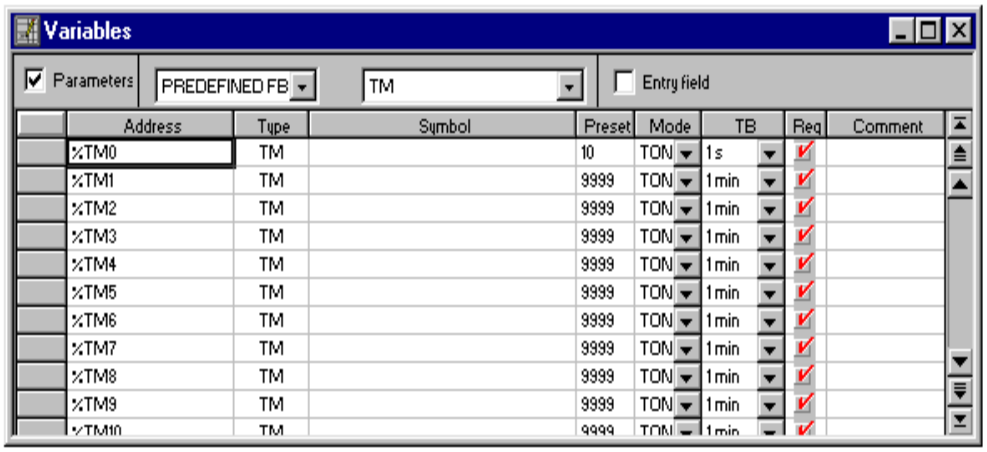
Zadanie szybkie jest o wyższym priorytecie niż zadanie główne. Jest okresowe w celu pozostawienia czasu na wykonanie zadań o niższym priorytecie. Operacje przetwarzania w tym zadaniu muszą być tak krótkie jak to tylko możliwe, aby nie wpływać negatywnie na zadanie główne. Jest ono użyteczne, kiedy wymagane jest monitorowanie szybkich okresowych zmian wejść cyfrowych.

Zadania przerwań w przeciwieństwie do zadania głównego i szybkiego, nie są związane z okresem. Ich wykonywanie jest przerywane przez zdarzenie występujące w określonym module aplikacji (tzn. przekroczenie progu licznika, zmiana stanu wejścia cyfrowego). Zadania te mają wyższy priorytet niż wszystkie inne zadania i dlatego są odpowiednie do operacji przetwarzania, wymagających bardzo krótkich czasów odpowiedzi na występujące zdarzenia.

Edytor zmiennych (rys. 2.5) jest używany do:

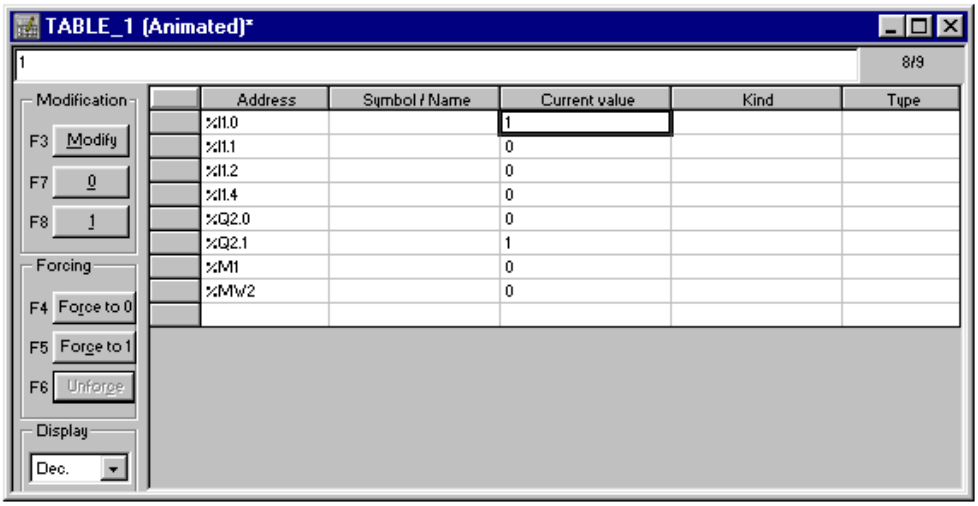
- przypisywania symboli zmiennym obiektom aplikacji (bitów, słów, bloków funkcji, we/wy itp.),
- definiowania parametrów standardowych bloków funkcji (bloki czasowe, liczniki, rejestry itp.),
- wprowadzania wartości stałych i określania sposobu wyświetlania (dziesiętny, heksadecymalny, zmiennoprzecinkowy, komunikat),
- wprowadzenia parametrów bloku funkcji użytkownika *DFB*.

To w edytorze zmiennych nadaje się wartość parametrom dla poszczególnych bloków funkcyjnych. Jest to ważne, ponieważ nie da się edytować danego bloku w głównym ekranie oprogramowania PL7 Pro.



Rys. 2.5. Nadanie wartości parametrom dla timera TM0

Edytor tablic animacji pozwala na monitorowane i modyfikowane zmiennych aplikacji (należy pracować w trybie „online”). Przykładowa tablica przedstawiona jest na rys. 2.6 Zmienne mogą być modyfikowane oraz wymuszane na 0 lub 1 dla obiektów bitowych. Dla każdej zmiennej numerycznej istnieje możliwość wyboru sposobu wyświetlania (dziesiętny, binarny, heksadecymalny, zmiennoprzecinkowy, komunikat ASCII).



Rys. 2.6. Tablica animacji

Edytor dokumentacji jest tworzony na bazie przeglądarki, która wyświetla zawartość pliku dokumentacji w strukturze drzewa. Może być stosowany do drukowania całości lub części pliku dokumentacji aplikacji na dowolnej drukarce graficznej, dostępnej w systemie Windows, używającej technologii *True Type* (format druku A4 lub wymiar calowy *US*). W edytorze

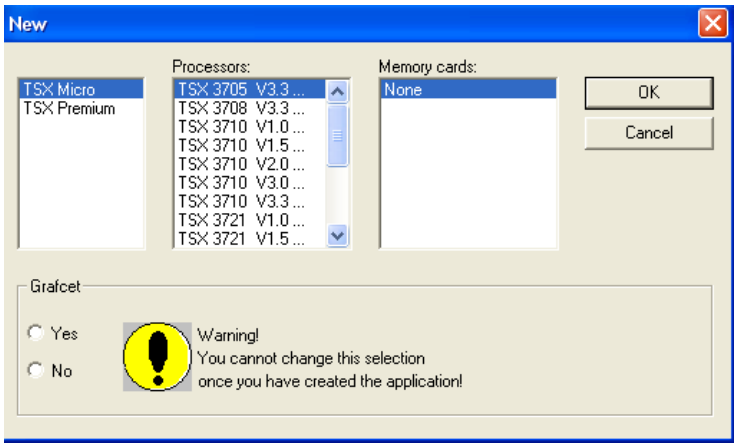
dokumentacji możemy zdefiniować stronę tytułową zawierającej nazwę projektanta i projektu, strony informacji ogólnych i stopki. Dokumentacja edytora generuje automatycznie: pliki dokumentacji aplikacji (program, sprzęt, konfigurację oprogramowania), listę zmiennych (sortowanych według adresów lub symboli) i ich powiązania z modułami programu.

Program narzędziowy do tworzenia wizualizacji jest zintegrowany z oprogramowaniem PL7 Pro (lub z oprogramowaniem PL7 Pro-Dyn). Jest szczególnie przydatny przy uruchamianiu programu, podczas rozruchu instalacji dla diagnostyki błędów i uszkodzeń. Ekrany wykonywania programu, odwzorowują i animują wszystkie zmienne. Tworzone są z wykorzystaniem następujących narzędzi:

- linii, trójkątów, elips, krzywych, wieloboków, obrazów (bit-mapy) i tekstów, które kształtują statyczne części ekranu (tło, tytuł itp.),
- animowanych obiektów graficznych, które odzwierciedlają stan procesu. Dla dynamicznej animacji obiektów, użytkownik przypisuje zmienne językowe (bit, bajt, słowa od długości pojedynczej/podwójnej i zmiennoprzecinkowe), wyświetlanie warunków (trwałych lub w zależności od typu zmiennej),
- określonych obiektów do sterowania procesu,
- biblioteki predefiniowanych obiektów graficznych (pompy, cylindry, przyciski, panel czołowy sterownika itp.). Użytkownik może modyfikować tę bibliotekę zgodnie ze swoimi wymaganiami,
- biblioteki komunikatów z wyświetlaniem warunkowym.

3. Grafcet w PL7

Chcąc wykorzystać metodę Grafcet podczas pisania nowego programu za pomocą oprogramowania PL7 należy zaznaczyć ją podczas wyboru rodzaju sterownika, co przedstawia rysunek 3.1.

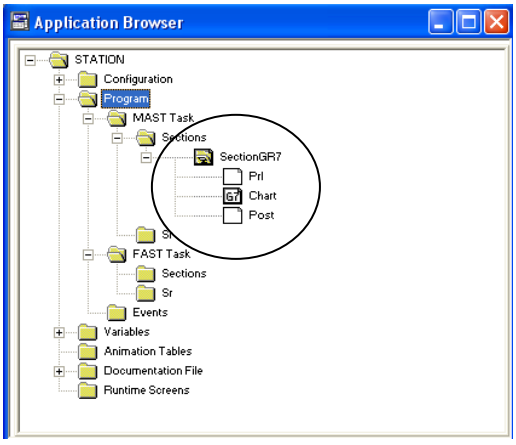


Rys. 3.1. Okno wyboru metody Grafcet.

3.1. Organizacja sekcji Grafcet

Program napisany w języku Grafcet można podzielić na trzy kolejne sekcje:

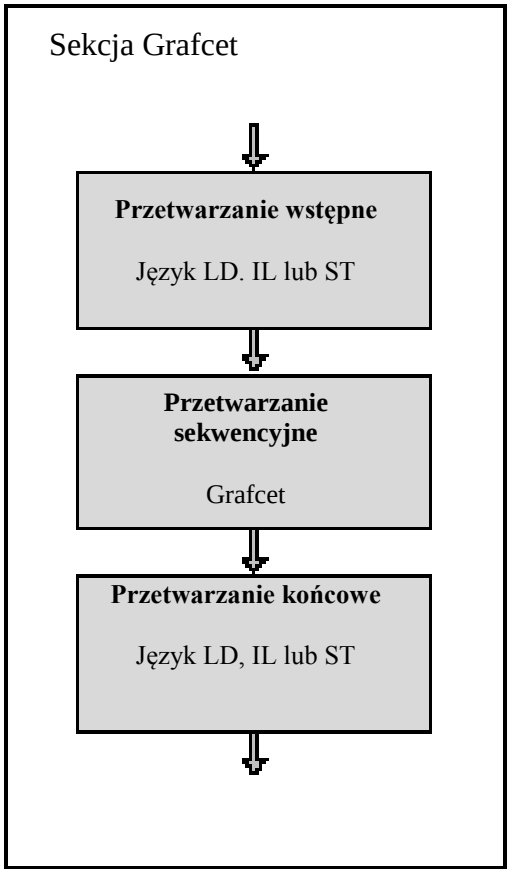
- przetwarzanie wstępne *PRL*,
- przetwarzanie sekwencyjne *Chart*,
- przetwarzanie końcowe *Post*.



Rys. 3.2. Umiejscowienie sekcji Grafcet.

Sekcja języka Grafcet jest umieszczana w zadaniu głównym MAST i wczytywana w następujący porządku:

- **Przetwarzanie wstępne,** które umożliwia ponowne rozpoczęcie działania programu w razie zaniku zasilania, przygotowanie diagramu Grafcet oraz odczytanie stanu wejść logicznych.
- **Przetwarzanie sekwencyjne,** służące do przetwarzania struktury sekwencyjnej aplikacji i umożliwiające przetwarzania warunków przejścia oraz akcji powiązanych bezpośrednio z krokami.
- **Przetwarzanie końcowe,** umożliwiające ustawienie logicznych wyjść oraz monitorowanie i zabezpieczanie tych wyjść.



Rys. 3.3. Sekcja Grafcet.

Przetwarzanie wstępne (*PRL*)

Przetwarzanie wstępne programuje się w języku *LD*, IL lub *ST* i jest ono wykonywane w całości od góry do dołu. Przetwarzanie wstępne wykonywane jest przed przetwarzaniem sekwencyjnym i przetwarzaniem końcowym i służy ono do przetwarzania wszystkich zdarzeń mogących mieć wpływ na:

- zarządzanie systemem w razie zaniku zasilania oraz podczas reinicjacji,
- kasowanie lub wstępne zadawanie parametrów diagramów *Grafcet*.

Stąd też tylko w przypadku przetwarzania wstępnego powyższych zdarzeń wykorzystane zostaną bity sprzężone z krokami (nadanie wartości 0 lub 1 bitom kroków %Xi lub %Xi.j za pośrednictwem instrukcji *Set* i *Reset*).

Czasami konieczne jest przygotowanie diagramu *Grafcet*, gdy ma nastąpić zmiana trybu pracy z pracy normalnej na tryb pracy specjalnej lub w razie wystąpienia jakiegoś zdarzenia (przykład: błąd powodujący zakłócenie pracy). Operacje te powodują zakłócenie normalnej pracy programu, dlatego też powinny być stosowane z dużą ostrożnością. Przetwarzanie przygotowujące może odnosić się do całości lub części przetwarzania sekwencyjnego:

- za pośrednictwem instrukcji SET i RESET,
- poprzez zresetowanie całego systemu (%S22) i nadanie krokom, przy następnym przejściu programu, wartości 1.

Ponieważ bity systemowe związane z diagramem *Grafcet* są ponumerowane w porządku ważności (%S21 do %S24), jednoczesne nadanie im wartości 1 podczas przetwarzania wstępnego powoduje, że są one przetwarzane jeden po drugim w kolejności rosnących numerów (w jednym "przejściu" programu przetwarzany jest tylko jeden bit). Bity te są uwzględniane na początku przetwarzania sekwencyjnego.

Bit %S21 służy do inicjacji diagramu *Grafcet* i ma normalnie wartość 0. Nadanie mu wartości 1 powoduje dezaktywację aktywnych kroków i uaktywnienie kroków inicjujących.

Nadanie wartość 1	Skasowanie do 0
• przez nadanie %S0 wartości 1 • za pomocą programu • za pomocą terminala	• przez system, na początku przetwarzania sekwencyjnego • za pomocą programu • za pomocą terminala

Bit %S22 służy do skasowania diagramu *Grafcet* do zera i ma normalnie wartość 0. Nadanie mu wartości 1 powoduje dezaktywację wszystkich aktywnych kroków przetwarzania sekwencyjnego. Aby w danej sytuacji możliwe było ponowne wystartowanie przetwarzania sekwencyjnego, aplikacja musi zawierać procedurę inicjacji lub procedurę przygotowawczą.

Nadanie wartość 1	Skasowanie do 0
• za pomocą programu • za pomocą terminala	• przez system, na końcu przetwarzania końcowego

Bit %S23 służy do zamrażania diagramu i ma normalnie wartość 0, a zmiana wartości na 1 powoduje zatrzymanie wykonywania diagramu. Niezależnie od wartości warunków przejścia następujących po aktywnych krokach stan diagramu się nie zmienia. Taki stan zamrożenia trwa dopóty, dopóki bit %S23 ma wartość 1.

Nadanie wartość 1	Skasowanie do 0
• za pomocą programu • za pomocą terminala	• za pomocą programu • za pomocą terminala

Przy uruchamianiu nowej aplikacji lub w przypadku utraty kontekstu, system inicjuje zimny start. System nadaje bitowi %S21 wartość 1 jeszcze przed uruchomieniem przetwarzania wstępnego tak, że diagram jest przygotowany do wykonywania z uwzględnieniem kroków inicjujących. Jeżeli użytkownik chce, aby aplikacja była po zimnym starcie wykonywana w jakiś szczególny sposób, to może on tego dokonać wykorzystując wartość bitu %S20, który ma stan 1 przez cały czas trwania pierwszego "przejścia" zadania głównego MAST.

W przypadku zaniku zasilania bez utraty kontekstu aplikacji, system inicjuje gorący start rozpoczynając wykonywanie aplikacji od miejsca, w którym wystąpił zanik. Jeżeli użytkownik chce, aby aplikacja była po gorącym starcie wykonywana w jakiś szczególny sposób, to może on tego dokonać wykorzystując, przy przetwarzaniu przygotowującym, wartość bitu %S1 i wywołując odpowiedni program.

Bit %S24 służy kasowania makrodefinicji do zera i ma normalnie stan 0, przestawienie go na 1 powoduje skasowanie do zera makr wskazanych w tablicy mieszczącej 4 słowa systemowe (%SW22 do %SW25).

Nadanie wartość 1	Skasowanie do 0
• za pomocą programu	• przez system, na początku przetwarzania sekwencyjnego

Wartość tego bitu powinna być zmieniana na 1 tylko podczas przetwarzania wstępnego. Kasowanie %S24 do 0 realizuje system; stąd nie ma potrzeby kasowania za pomocą programu lub terminala.

Przetwarzanie sekwencyjne (CHART)

Ta część programu stosowana jest do programowania sekwencyjnej struktury programu. Przetwarzanie takie składa się z diagramu głównego zapisanego na 8 stronach. W diagramie głównym można zamieścić kilka autonomicznych diagramów *Grafcet,* które mogą być programowane i wykonywane jednocześnie.

Diagram *Grafcet* powinien być programowany w następujący sposób:

Faza 1:

1) oszacowanie warunków aktywnych bramek,
2) żądanie dezaktywacji, połączonych z bramką, kroków poprzedzających,
3) żądanie uaktywnienia, połączonych z bramką, kroków następnych.

Faza 2:

Zaktualizowanie stanu diagramu z uwzględnieniem otwartych bramek:

1) dezaktywacja kroków poprzedzających otwarte bramki,

2) uaktywnienie kroków następujących po otwartych bramkach,
3) zablokowanie otwartych bramek,
4) odblokowanie bramek następujących po nowo uaktywnionych krokach.

System aktualizuje dwie dedykowane tablice zwierające informacje o aktywności kroków i odblokowaniu bramek:

- **tablica aktywności kroków** przechowuje aktualne (dla bieżącego "przejścia" programu) informacje o krokach aktywnych, krokach, które mają być uaktywnione oraz o krokach, które mają być zdezaktywowane,
- **tablica odblokowania bramek** przechowuje aktualne (dla bieżącego "przejścia" programu) informacje o bramkach następujących po krokach uwzględnionych w poprzedniej tablicy,

Faza 3:

Akcje przyporządkowane aktywnym krokom są wykonywane w następujący sposób:

1) akcje związane z dezaktywacją kroków, które są dezaktywowane,
2) akcje związane z uaktywnianiem kroków, które są uaktywniane,
3) akcje związane z aktywnymi krokami, wykonywane w sposób ciągły.

Liczba elementów w tablicy aktywności kroków i tablicy odblokowania bramek jest konfigurowalna. Przekroczenie pojemności tych tablic powoduje:

- zatrzymanie pracy sterownika (przerwanie wykonywania aplikacji),
- zmianę stanu bitu %S26 na 1 (przekroczenie pojemności jednej z tablic),
- zapalenie się kontrolki ERR na sterowniku.

Przetwarzanie końcowe (POST)

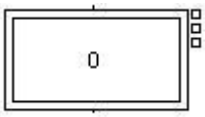
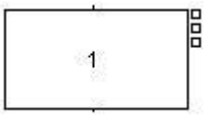
Przetwarzanie końcowe programuje się w języku LD, IL lub *ST*. Jest to przetwarzanie wykonywane jako ostatnie, przed zaktualizowaniem stanów wyjść, może być wykorzystane do programowania logiki wyjść. Przetwarzanie to umożliwia wykonanie akcji wygenerowanych w czasie przetwarzania sekwencyjnego poprzez zintegrowanie trybu pracy i zatrzymania oraz uwzględnienie zabezpieczeń (*safety interlocks*) przypisanych do akcji podczas obliczania stanów wyjść To przetwarzanie może być również wykorzystane do

przetworzenia wyjść kilkukrotnie uaktywnianych podczas przetwarzania sekwencyjnego. Przetwarzanie końcowe jest również wykorzystywane do programowania wyjść, których stany mają być niezależne od przetwarzania sekwencyjnego.

3.2. Zasady tworzenia diagramu

Elementami grafu sekwencji są połączone ze sobą za pomocą linii kroki (STEP), nazywane też w języku polskim etapami, i tranzycje (TRANSITION), nazywane też przejściami. Symbolem graficznym kroku jest prostokąt z wpisanym numerem kroku. Z każdym krokiem jest skojarzony zestaw działań wykonywanych przez sterownik w ramach danego kroku, zapisanych za pomocą jednego z wcześniej wymienionych języków. Kolejne kroki w grafie sekwencji są łączone za pomocą pionowych linii. Przy tym wejście do kroku jest u góry jego symbolu, a wyjście u dołu. Przejście reprezentuje warunki, jakie muszą być spełnione, aby mogły być wykonywane kroki następne, zgodnie z liniami łączącymi kroki. Symbolem graficznym przejścia jest pozioma kreska na linii łączącej kroki. Z każdym przejściem są skojarzone określone warunki przejścia, które również są zapisane z wykorzystaniem wcześniej opisanych języków programowania. Wygląd oraz opis elementów wykorzystywanych do tworzenia diagramu Grafcet przedstawia tabela 3.1.

Tabela 3.1. Elementy języka Grafcet

Symbol	Oznaczenie	Opis
	Krok inicjujący	Oznacza kroki aktywne na początku cyklu po inicjacji lub zimnym starcie.
	Krok pojedynczy	Oznacza, że system sterujący jest stabilny.

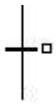
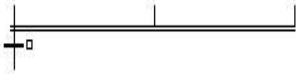
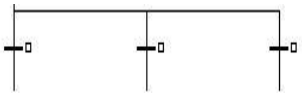
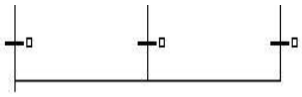
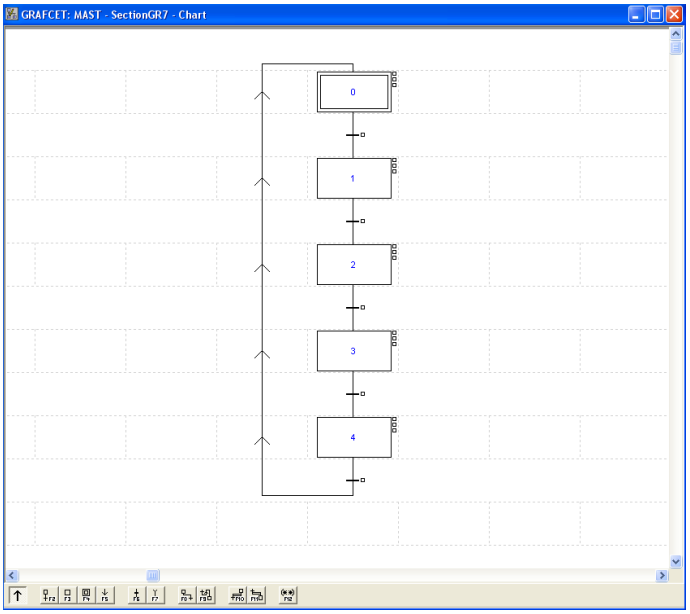
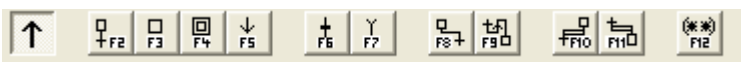
	Bramka (tranzycja)	Bramka realizuje przejście od jednego kroku, do drugiego. Związany z bramką warunek jest niezbędny do przejścia do następnego kroku.
	Początek koniunkcji	Jest to przejście od jednego kroku do kilku kroków. Umożliwia jednoczesne uaktywnienie maksymalnie 11 kroków.
	Koniec koniunkcji	Przejście od kilku jednocześnie otwartych kroków do jednego kroku. Umożliwia jednoczesne zamknięcie maksymalnie 11 kroków.
	Początek alternatywy	Przejście od jednego kroku do kilku innych. Umożliwia wybór maksymalnie 11 sekwencji kroków.
	Koniec alternatywy	Przejście od kilku kroków do jednego. Zamyka jedną z maksymalnie 11 sekwencji kroków.
	Łącznik źródłowy	'n' jest numerem kroku, od którego przychodzi połączenie (krok źródłowy).
	Łącznik docelowy	'n' jest numerem kroku, do którego ma nastąpić przejście (krok docelowy).
	Łączniki bezpośrednie	Te łączniki stosowane są do wyboru sekwencji, do przeskoczenia jednego lub kilku kroków, do powtórzenia kroków (sekwencji).

Diagram główny może być zaprogramowany na 8 stronach (strony od 0 do 7). Każda strona zawiera 14 linii i 11 kolumn, co w efekcie daje 154 komórki. W jednej komórce może być umieszczony jeden element graficzny. Podczas tworzenia diagramu obowiązują następujące reguły:

- pierwsza linia służy do wprowadzania łączników źródłowych,
- ostatnia linia służy do wyprowadzania łączników docelowych,
- linie parzyste (od 2 do 12) są liniami kroków (dla kroków i łączników źródłowych),
- linie nieparzyste (od 3 do 13) są liniami bramek (dla bramek i łączników docelowych),
- numery kroków (od 0 do 127) definiuje się w dowolnym porządku,
- na jednej stronie można umieścić kilka diagramów.



Rys. 3.4. Wygląd okna do tworzenia diagramu Grafcet

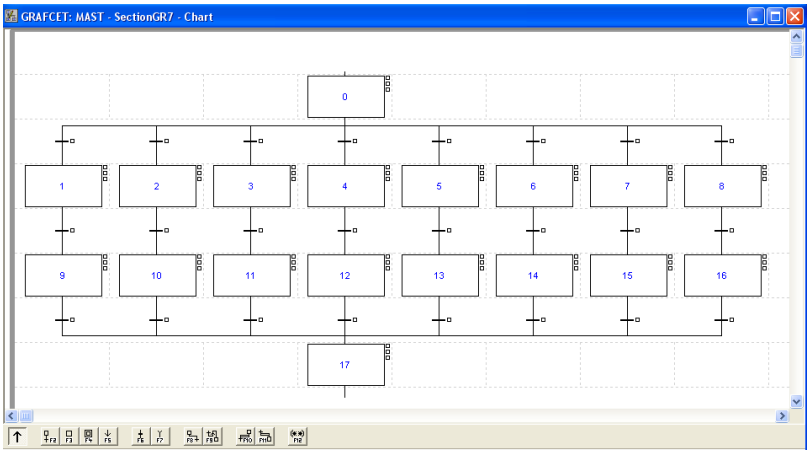


Rys. 3.5. Wygląd klawiszy służących do wstawiania elementów języka Grafcet

Wybór sekwencji i zakończenie wyboru (alternatywa)

Podczas tworzenia alternatywy (rys. 16): należy zachować następujące reguły

- liczba przejść górę w stosunku do końca wyboru sekwencji (zamknięcie alternatywy) i przejść w dół w stosunku do wyboru sekwencji (początek alternatywy) nie może przekraczać 11,
- wybór sekwencji rysuje się przy pomocy łącznika poziomego,
- otwarta alternatywa musi być zamknięta,
- w celu zabezpieczenia się przed jednoczesnym otwarciem kilku bramek należy tak dobierać warunki, żeby się one wzajemnie wykluczały.

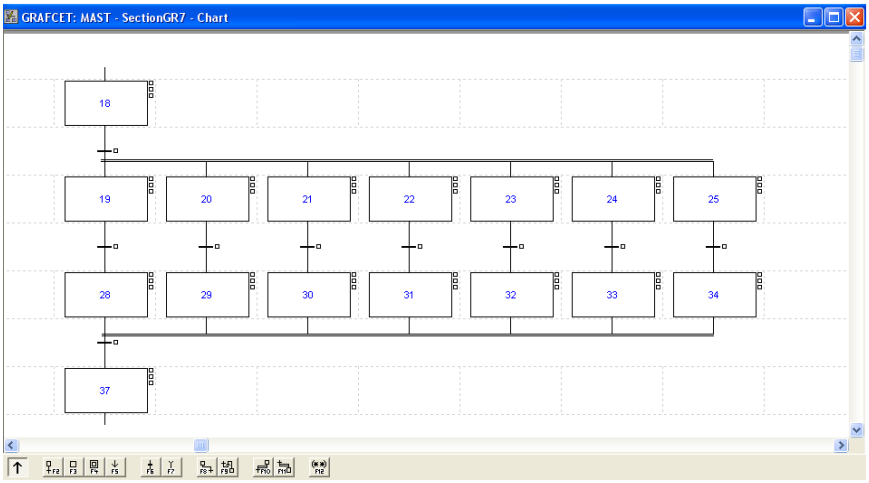


Rys. 3.6. Wybór sekwencji i zakończenie wyboru (alternatywa).

Jednoczesne uaktywnienie i zamknięcie kroków (koniunkcja)

Podczas tworzenia koniunkcji (rys. 17): należy zachować następujące reguły

- liczba kroków w dół w stosunku do początku koniunkcji (jednoczesne uaktywnienie kroków) i w górę w stosunku do zamknięcia koniunkcji (jednoczesne zamknięcie kroków) nie może przekraczać 11,
- otwarta koniunkcja musi być zamknięta,
- jednoczesna aktywacja kroków zawsze odbywa się od lewej do prawej,
- jednoczesne zamknięcie kroków zawsze odbywa się od prawej do lewej.

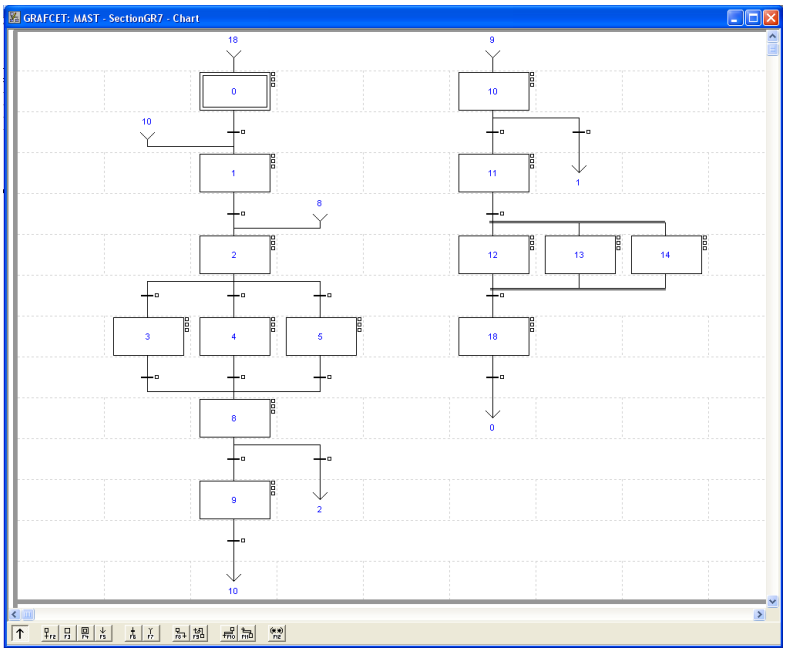


Rys. 3.7. Jednoczesne uaktywnienie i zamknięcie kroków (koniunkcja).

Stosowanie łączników

Zastosowanie łączników źródłowych i docelowych (rys. 18) pozwala na zapewnienie ciągłości diagramu wtedy, kiedy łącze bezpośrednie nie może być narysowane w obrębie danej strony lub dwu kolejnych stron. Tę ciągłość zapewnia zastosowanie łącznika docelowego, który występuje w parze z łącznikiem źródłowym.

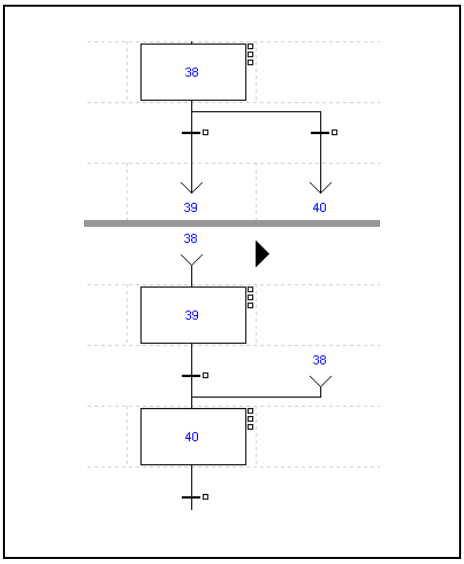
- Przy pomocy łączników można zrealizować pętle (np. zapętlić krok 18 z 0).
- Za pomocą łączników można restartować sekwencje (np. zastosować połączenie kroku 10 z krokiem 1 lub kroku 8 z krokiem 2).
- Zastosowanie łączników jest niezbędne, gdy rozgałęzienie programu wykracza poza stronę (np. połączenie kroku 9 z krokiem 10).



Rys. 3.8. Stosowanie łączników.

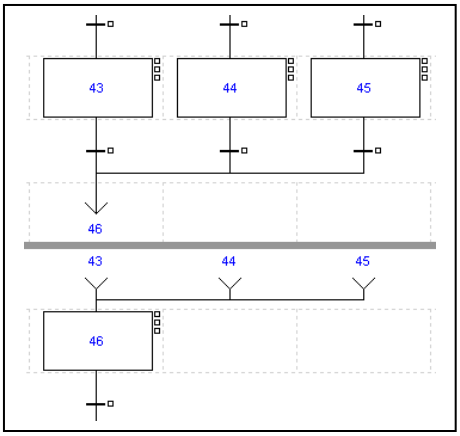
Łączniki wyboru sekwencji i zamknięcia wyboru sekwencji

- W przypadku wyboru sekwencji, bramki i włączniki docelowe muszą być umieszczone na tej samej stronie.

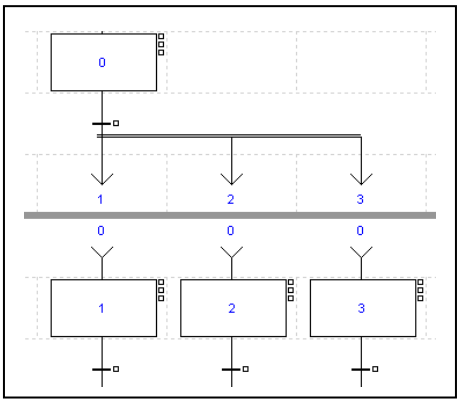


- W przypadku zamknięcia wyboru sekwencji, łącznik źródłowy musi być umieszczony na tej samej stronie, co krok docelowy.

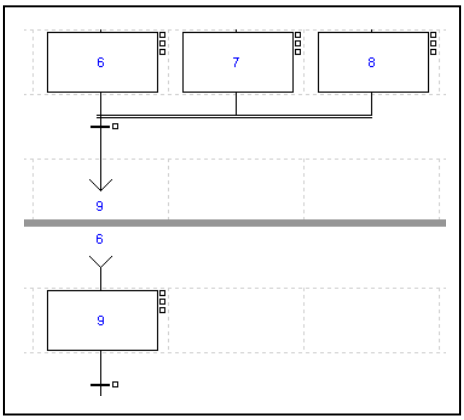
- W przypadku, gdy zamknięcie wyboru sekwencji występuje przed łącznikiem docelowym, to należy zdefiniować tyle samo łączników źródłowych ile jest kroków przed zamknięciem wyboru sekwencji.



Łączniki jednoczesnego uaktywnienia kroków



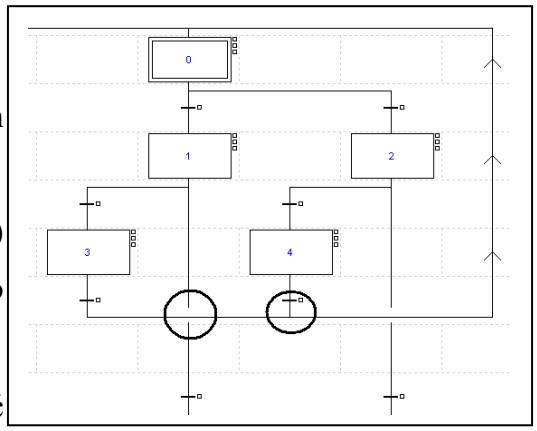
- W przypadku jednoczesnego uaktywnienia kroków należy wstawić łączniki docelowe na tej samej stronie, co krok i bramka.



- W przypadku jednoczesnego zamknięcia, kroki i bramka zamknięcia, muszą znajdować się na tej samej stronie łącznik docelowy.
  Kiedy kilka kroków działa na jedną bramkę, łącznik źródłowy przyjmuje numer pierwszego kroku od lewej strony licząc (patrząc w górę w stosunku do łącznika).

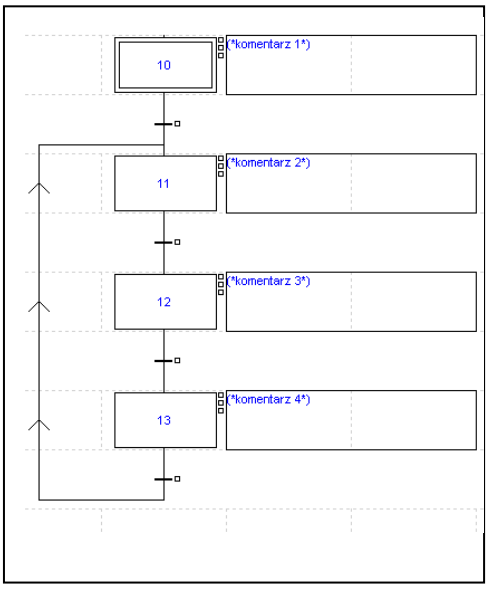
Łączniki bezpośrednie

- Łączniki bezpośrednie łączą krok z bramką albo bramkę z krokiem. Mogą być poziome albo pionowe.
- Łącza bezpośrednie mogą:
  - przecinać łączniki innych rodzajów,
  - łączyć się (tworzyć węzły) z łącznikami tego samego rodzaju.
- Łącznik bezpośredni nie może być przecinany przez jednoczesne uaktywnienie kroków oraz przez zamknięcie jednoczesnego uaktywnienia.



Komentarze

- Do każdej komórki diagramu Grafcet można dopisać komentarz. Komentarz zaczyna się znakiem (* i kończy się znakiem *). Może on mieć maksymalnie 64 znaki.
- Komentarz zajmuje obszar dwóch kolejnych komórek i może mieć szerokość dwóch linii. Jeżeli komentarz się nie mieści w okienku, to następuje jego obcięcie. Podczas drukowania dokumentu komentarz jest drukowany w całości.
- Komentarze umieszczane na stronach diagramu Grafcet są zapamiętywane w sterowniku jako dane graficzne.



Akcje przyporządkowane krokom

Każdemu krokowi przyporządkowuje się akcje (operacje) zaprogramowane w języku drabinkowym LD, języku instrukcji IL lub języku strukturalnym ST. Akcje te wykonywane są tylko wtedy, gdy sprzężony z nimi krok jest aktywny. W programie PL7, za pomocą którego tworzy się sterowanie sekwencyjne (strukturę Grafcet), wyróżnia się trzy rodzaje akcji:

- **akcje na skutek uaktywnienia** – akcje są wykonywane w momencie, gdy następuje uaktywnienie kroku,
- **akcje na skutek dezaktywacji** – akcje są wykonywane w momencie dezaktywacji kroku,
- **akcje ciągłe** – akcje są wykonywane w sposób ciągły przez cały czas aktywności sprzężonego z nimi kroku.

Akcje tych trzech rodzajów można stosować dla każdego. Akcje wykonywane przy uaktywnieniu i dezaktywacji kroku są realizowane tylko jeden raz po wywołaniu. Służą one wywoływania procedur, zwiększania wartości licznika itd. Pojedyncza akcja składa się kilku elementów programowych (sekwencje, wyrażenia, labelki). Wymienione powyżej akcje opisuje się w następujący sposób:

MAST - <nazwa sekcji Grafcet> - DIAGRAM (lub MAKROk) - STRONA n %Xi x

gdzie: x = P1 Uaktywnienie

=N1 Ciągłe wykonywanie akcji

=P0 Dezaktywacja

n =numer strony

i =numer kroku

Zasady dotyczące akcji

- Wszystkie akcje są zapamiętywane, co w konsekwencji powoduje, że:
  - akcja uruchamiana na czas aktywności kroku Xn musi być skasowana (reset) w momencie dezaktywacji kroku Xn lub w momencie uaktywnienia kroku Xn+1,
  - akcja, której efekty wpływają na kilka kroków, przyjmuje wartość 1 przy uaktywnieniu kroku Xn i jest resetowana przy uaktywnieniu kroku Xn+m.
- Wszystkim akcją można przypisywać warunki logiczne (wykonywanie warunkowe).
- Akcje, które są wykonywane z zastosowaniem zabezpieczeń (*safety interlocks*) umieszcza się w przetwarzaniu końcowym (*post-processing*). Jest to przetwarzanie wykonywane na końcu każdego „przebiegu" programu.

Kolejność wykonywania akcji

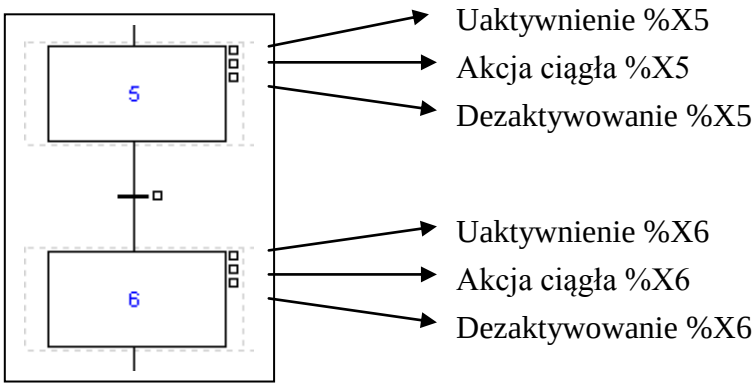
W przykładzie zamieszczonym poniżej, kolejność wykonywania akcji (w trakcie jednego "przejścia") jest następująca:

Uaktywnienie kroku 6 powoduje uaktywnienie akcji w następującym porządku:

1 - akcje związane z dezaktywacją kroku 5,

2 - akcje związane z uaktywnieniem kroku 6,

3 - ciągłe wykonywanie akcji związanej z krokiem 6.



Dezaktywacja kroku 6 powoduje przerwanie wykonywania akcji ciągłej.

Warunki przejścia bramki

Każdej bramce przyporządkowany jest warunek zaprogramowany w języku drabinkowym LD, języku instrukcji IL lub w języku ST. Warunek przejścia bramki jest uwzględniany tylko wtedy, gdy bramka jest aktywna. Warunek stanowi labelka, lista instrukcji lub wyrażenie języka ST składające się z szeregu odczytów (testów) stanu bitów lub słów. Warunek przejścia, któremu nie przypisano żadnego programu jest zawsze traktowany jako warunek typu fałsz.

Warunki przejścia opisane są w następujący sposób:

MAST-<nazwa sekcji Grafcet>-DIAGRAM (lub MAKROk) - STRONA n %X(i) --> % X(j)

gdzie: n = numer strony, i =numer kroku poprzedzającego, j =numer kroku następnego

W przypadku jednoczesnego uaktywniania lub dezaktywacji kroków wpisuje się adres kroku w pierwszej kolumnie od lewej strony.

Użytkownik ma do dyspozycji bity obiektowe związane z krokami, bity systemowe języka *Grafcet,* słowa obiektowe sygnalizujące aktywność kroków i słów systemowych języka *Grafcet*. Obiekty języka *Grafcet* zebrano w tabeli 3.2.

Tabela 3.2

Oznaczenie	**Adres**	**Opis**
Bity kroków (1=krok aktywny)	**%Xi**	Status kroku *i* głównego diagramu (*i* od 0 do *n*, gdzie *n* zależy od procesora).
	%XMj	Stan makra j (*j* od 0 do 63 dla TSX/PMX/PCX 57)
	%Xj.i	Status kroków *i* makra *j*
	%Xj.IN	Status kroku wejściowego makra *j*
	%Xj.OUT	Status kroku wyjściowego makra *j*
Bity systemowe *Grafcet* (1)	**%S21**	Inicjacja diagramu
	%S22	Skasowanie wszystkich diagramów
	%S23	„Zamrożenie” diagramu
	%S24	Skasowanie makr do 0 z uwzględnieniem słów systemowych %SW22 do %SW25
	%S26	Przyjmuje wartość 1 w przypadku: -przepełnienia tablicy (kroki/bramki), -wykonywania niewłaściwego diagramu (połączenie z krokiem nie należącym do diagramu)
Słowa kroków	**%Xi.T**	Czas aktywności kroku *i* diagramu głównego
	%Xj.i.T	Czas aktywności kroku *i* makra *j*
	%Xj.IN.T	Czas aktywności kroku wejścia makra *j*
	%Xj.OUT.T	Czas aktywności kroku wyjścia makra *j*

Tabela 3.2 c.d.

Słowa systemowe **Grafcet-u**	**%SW20**	Słowo określające dla bieżącego cyklu liczbę kroków aktywnych oraz kroków, które mają być uaktywnione lub zakończone
	%SW21	Słowo określające, dla bieżącego cyklu liczbę bramek aktywnych oraz bramek, które mają być uaktywnione lub zamknięte
	%SW22 do %SW25	Wartość 4 słów powodujących skasowanie makr do 0, kiedy bit %S24 ma wartość 1

Podsumowanie właściwości grafu sekwencji

1) Jednostka organizacyjna oprogramowania napisana za pomocą grafu sekwencji składa się z jednej lub kilku sieci, w których występują kroki, przejścia i połączenia między nimi:
   - Kroki i przejścia muszą występować naprzemiennie, nie można łączyć kroku z krokiem lub przejścia z przejściem.
   - Z każdym przejściem są związane warunki przejścia.
   - Z krokami są kojarzone odpowiednie akcje. Są to akcje związane z aktywacją kroku, z działaniem ciągłym kroku oraz akcje związane z dezaktywacją kroku.
2) Z chwilą wywołania programu zawierającego strukturę SFC jest aktywny krok początkowy sieci. W każdej sieci może znajdować się tylko jeden krok początkowy. Dalszy rozwój sieci polega na przechodzeniu między kolejnymi krokami według następujących zasad:
   - Dla każdego aktywnego kroku, po wykonaniu przypisanych mu akcji, następuje sprawdzenie warunku przejścia znajdującego się bezpośrednio po kroku. Przejście występujące za krokiem aktywnym nosi nazwę przejścia dozwolonego.

- Jeżeli warunki przejścia dla przejścia dozwolonego są spełnione, to krok poprzedzający przejście staje się nieaktywny, przejście jest kasowane, a krok występujący za przejściem jest aktywowany.
- Jeżeli warunki przejścia są niespełnione, to krok poprzedzający pozostaje aktywny.

3) Wykonywanie kroków aktywnych podlega następującym regułom:
   - Po aktywacji krok jest wykonywany, co najmniej raz, włączając w to wszystkie skojarzone z nim akcje.

Po dezaktywacji kroku wszystkie skojarzone z nim akcje są wykonywane ostatni raz w celu zapewnienia właściwego zakończenia akcji.

4. Zadania do samodzielnej realizacji

1. Zapoznać się z obiektem laboratoryjnym „Układ 3 wagoników" i wypróbować sterowanie ręczne.
2. W środowisku PL7 zainstalować **wcześniej przygotowany** WŁASNY algorytm sterowania w postaci diagramu Grafcet.
3. Uruchomić przygotowany algorytm w sterowniku TSX Micro i przetestować jego działanie na obiekcie laboratoryjnym.
4. Eksperymentalnie dobrać optymalne wartości sterowania – prędkości wagoników.

W sprawozdaniu należy udokumentować wszystkie etapy realizacji ćwiczenia, przedstawić wyniki, sformułować wnioski.