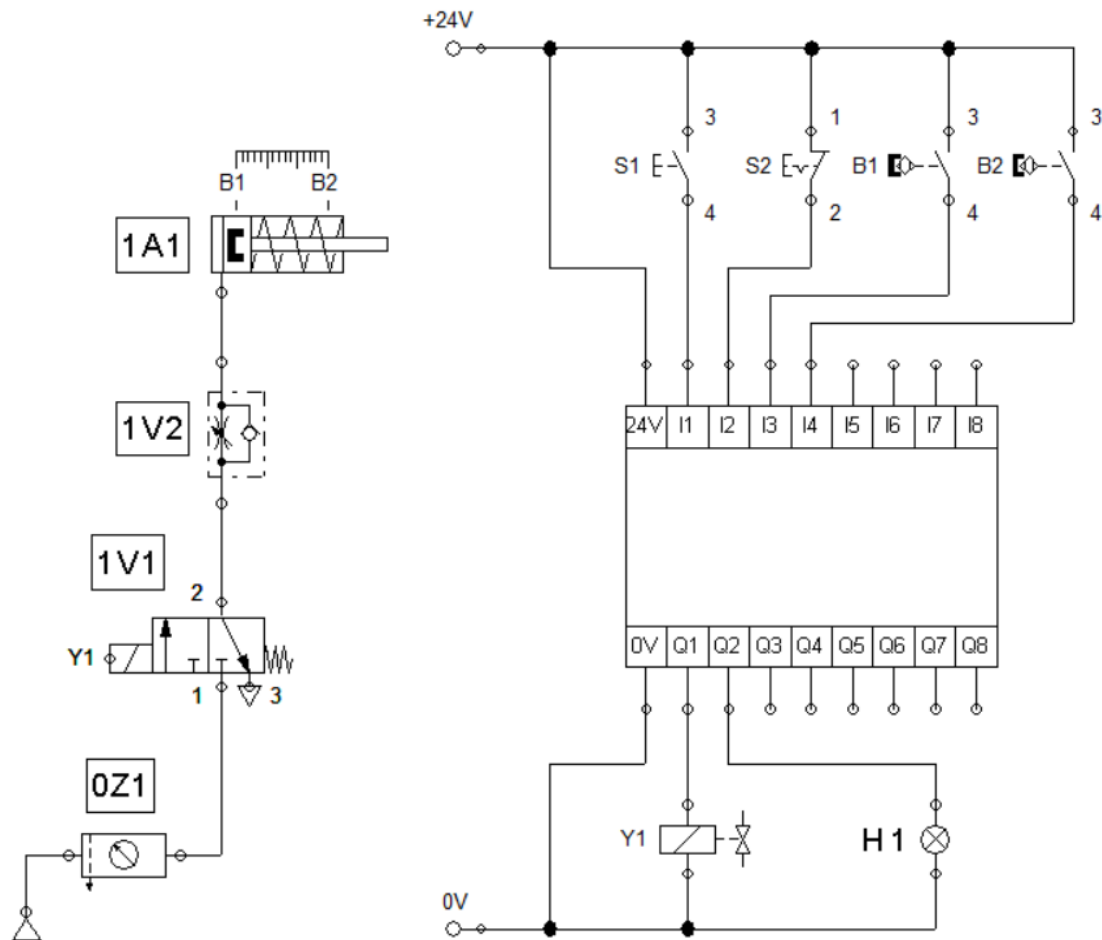


KONWERSJA ALGORYTMU SFC NA PROGRAM W JĘZYKU LD

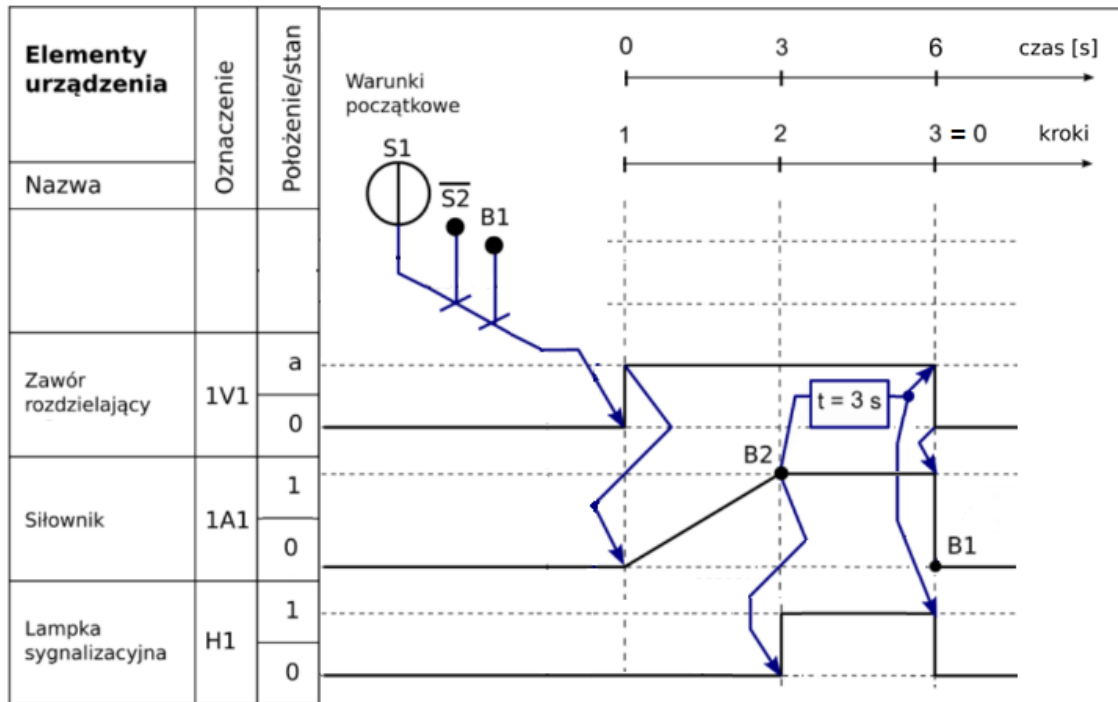
Schemat części pneumatycznej i elektrycznej układu mechatronicznego



Tekstowy opis pracy układu mechatronicznego

Po włączeniu zasilania tłoczek siłownika 1A1 pozostaje wsunięty, a lampka H1 nie świeci. Gdy tłoczek siłownika jest wsunięty i przycisk S2 nie jest wciśnięty, to naciśnięcie przycisku S1 powoduje rozpoczęcie wysuwania tłoczka siłownika. Po całkowitym wysunięciu tłoczka siłownika załącza się lampka H1. Po czasie 3s lampka gaśnie i rozpoczyna się wsuwanie tłoczka siłownika. Wsuwanie tłoczka i zgaszenie świecącej lampki można również wywołać w dowolnym momencie pracy układu przez wciśnięcie przycisku S2. Tłoczek siłownika można ponownie wysunąć dopiero po ponownym spełnieniu warunków początkowych. Poziom dławienia zaworu 1V2 nastawiony jest tak, że tłoczek siłownika 1A1 osiąga stan pełnego wysunięcia po 3s od rozpoczęcia wysuwania.

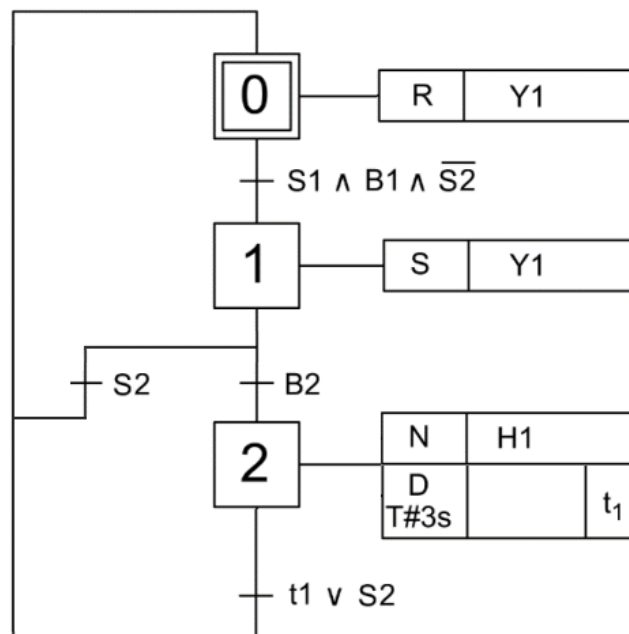
Graficzny opis pracy układu mechatronicznego za pomocą **cyklogramu**



*warunki w cyklogramie zostały zapisane z zastosowaniem logiki, w której np. S2 oznacza przycisk naciśnięty, a $\overline{S2}$ oznacza przycisk nienaciśnięty, niezależnie od tego, czy posiada on zestyk NO czy NC

UWAGA: Naciśnięcie przycisku S2 w dowolnym momencie pracy układu powoduje natychmiastowe wsuwanie tłoczyska siłownika 1A1 i zgaszenie świecącej lampki H1.

Graficzny opis pracy układu mechatronicznego za pomocą **algorytmu SFC**



*tranzycje w algorytmie zostały zapisane z zastosowaniem logiki, w której np. S2 oznacza przycisk naciśnięty, a $\overline{S2}$ oznacza przycisk nienaciśnięty, niezależnie od tego, czy posiada on zestyk NO czy NC

Etapy tworzenia programu w języku LD na podstawie algorytmu SFC

Sekcja kroków:

1. Wstaw tyle bloków RS (Latching Relay) ile jest kroków w algorytmie SFC.
2. Nazwij każdy blok RS tak, aby kojarzył się z danym krokiem algorytmu, np. KROK 0, KROK 1 itd. Dla ułatwienia identyfikacji bloków RS w dalszej części pracy, blok RS o oznaczeniu SF001 nazwij KROK 1, blok RS o oznaczeniu SF002 nazwij KROK 2 itd. Jeśli krok początkowy w algorytmie SFC ma nadany numer 0, to nazwę KROK 0 nadaj blokowi RS o najwyższym oznaczeniu SF, czyli temu, który wstawiłeś do programu jako ostatni blok RS.
3. Rozmieść bloki RS w programie jeden pod drugim w pobliżu prawej krawędzi strony w kolejności od kroku o najmniejszym numerze do kroku o największym numerze, zostawiając między nimi od 3 do 5 linii odstępu, w zależności od stopnia skomplikowania algorytmu. W te puste linie będą później wstawiane kolejne elementy programu.
4. Pod każdym blokiem RS wstaw symbol wyjścia Q (Relay coil). Umieść go w przedostatniej wolnej linii po danym bloku RS.
5. Zmień funkcję wszystkich wstawionych wyjść Q na flagi M. Dla ułatwienia identyfikacji flag, oznaczenia dobierz tak, aby pod blokiem RS nazwanym KROK 1 znalazła się flaga M1, pod blokiem RS oznaczonym KROK 2 znalazła się flaga M2 itd. Jeśli krok początkowy to KROK 0, umieść pod nim flagę o najwyższym numerze. Pamiętaj, że flagi M8, M25, M26, M27 mają przypisane specjalne funkcje i nie wolno ich używać do opisywanych w tym punkcie działań.
6. Nazwij każdą flagę M tak, aby kojarzyła się z danym blokiem RS (krokiem), np. fladze M pod blokiem RS KROK 0 nadaj nazwę FLAGA K0, fladze M pod blokiem RS KROK 1 nadaj nazwę FLAGA K1 itd.
7. Na początku każdej linii, w której wstawiłeś flagę M wstaw symbol wejścia prostego I (Make contact) i powiąż go z blokiem RS danego kroku, np. dla flagi M1 z kroku 1 będzie to powiązanie z blokiem RS o oznaczeniu SF001, nazwanym wcześniej jako KROK 1. Następnie połącz wstawiony symbol wejścia z linią zasilającą znajdującą się po lewej stronie programu i z symbolem flagi znajdującym się na końcu tej samej linii programu. W ten sposób flaga M będzie sygnalizować stan bloku RS. W teoretycznych rozważaniach flagi M nie są potrzebne, ale w praktyce oprogramowanie LOGO!Soft Comfort będzie wymuszało użycie flagi w przypadku występowania w programie rekurencji, tzn. sytuacji, gdy np. stan bloku RS KROK 0 zależy od stanu bloku RS KROK 2, a stan bloku RS KROK 2 zależy od stanu bloku RS KROK 0.
8. Określ warunki aktywacji poszczególnych kroków algorytmu SFC zaczynając od kroku początkowego. Główny warunek aktywacji kroku początkowego algorytmu (aktywacja po przełączeniu sterownika PLC w tryb RUN) jest zawsze taki sam, bez względu na stopień skomplikowania algorytmu SFC: krok początkowy jest aktywny wtedy, gdy wszystkie pozostałe kroki algorytmu nie są aktywne. Ten warunek zapisz w programie wstawiając między linię zasilającą a wejście S bloku RS KROK 0 szeregowo połączone ze sobą symbole wejść zanegowanych I (Break contact) w ilości równej liczbie wszystkich kroków algorytmu pomniejszonej o 1. Wstawione wejścia I powiąż z blokami RS poszczególnych kroków, innych niż krok początkowy, czyli KROK 1, KROK 2 itd.
9. Określ warunki aktywacji pozostałych kroków na podstawie algorytmu SFC i zapisz je w programie. Pamiętaj, że aktywacja danego kroku następuje w momencie, gdy aktywny jest krok poprzedzający go i spełniony jest warunek przejścia (tranzycja) znajdujący się między tymi dwoma krokami w algorytmie. Dlatego też zapis warunku aktywacji danego kroku w programie zacznij od wstawienia na początku linii znajdującej się na wysokości wejścia S bloku RS danego kroku, symbolu wejścia prostego I (Make contact) i powiąż go z flagą M sygnalizującą stan bloku RS kroku poprzedzającego dany krok. Przykładowo na wysokości wejścia S bloku RS KROK 1 wstaw symbol

- wejścia prostego I powiązanego z flagą M sygnalizującą stan bloku RS KROK 0. Następnie szeregowo z tym symbolem połącz następne symbole wejść opisujące odpowiednio warunek przejścia do danego kroku, a na koniec dołącz utworzoną linię do wejścia S bloku RS danego kroku.
10. Pamiętaj, że również krok początkowy może być aktywowany w sposób opisany w poprzednim punkcie. Jest to działanie alternatywne w stosunku do działania opisanego wcześniej jako główny warunek aktywacji kroku początkowego i powinieneś je zapisać równoległe względem warunku głównego, tzn. w linii pod warunkiem głównym.
 11. Jeżeli w warunkach przejść występują odwołania do stanu timerów lub liczników, to przed zapisaniem takiego warunku w programie należy wstawić blok odpowiedniego timera lub licznika na końcu utworzonego dotychczas programu, tzn. pod flagą ostatniego kroku, w tzw. sekcji akcji. Dobierz odpowiednie nastawy timerów i liczników zgodnie z algorytmem SFC. Na tym etapie nie podłączaj jeszcze tych bloków do linii zasilającej (zostaw puste miejsce po lewej stronie tych bloków).
 12. Określ warunki dezaktywacji poszczególnych kroków algorytmu SFC i zapisz je w programie. Pamiętaj, że dezaktywacja danego kroku następuje w momencie, gdy aktywny staje się krok występujący w algorytmie bezpośrednio po tym kroku. Dlatego też zapis warunku dezaktywacji danego kroku w programie zacznij od wstawienia na początku pierwszej wolnej linii znajdującej się w pobliżu wejścia R bloku RS danego kroku, symbolu wejścia prostego I (Make contact). Następnie powiąż go z krokiem występującym bezpośrednio po tym kroku w algorytmie SFC lub z flagą M tego kroku, jeżeli LOGO!Soft Comfort tego wymaga. Tak wstawiony i powiązany symbol wejścia połącz z linią zasilającą i wejściem R bloku RS danego kroku.
 13. Jeżeli w algorytmie SFC występuje możliwość natychmiastowego zatrzymania pracy urządzenia i powrót do kroku początkowego np. przez naciśnięcie przycisku pełniącego funkcję STOP, to uwzględnij to w programie. Do wejścia R każdego bloku RS, za wyjątkiem bloku RS kroku początkowego, dołącz symbol odpowiedniego wejścia I powiązanego z przyciskiem funkcji STOP jako alternatywny sposób natychmiastowej dezaktywacji danego kroku.

Sekcja akcji:

14. Na końcu utworzonego dotychczas programu wstaw symbole wyjść Q (Relay coil) w ilości odpowiadającej liczbie używanych w zadaniu wyjść sterownika PLC. Umieść je jeden pod drugim, bliżej prawej krawędzi strony. Nazwij każdy wstawiony symbol wyjścia oznaczeniem elementu dołączonego do danego wyjścia sterownika PLC, np. symbol Q1 nazwij Y1, a Q2 nazwij H1.
15. Określ, na podstawie algorytmu SFC, w których krokach aktywne są poszczególne wyjścia Q. Następnie wstaw między linię zasilającą a symbol danego wyjścia Q symbol wejścia prostego I (Make contact) i powiąż go z blokiem RS kroku, w którym dane wyjście Q jest aktywne. Jeżeli dane wyjście Q jest aktywne w kilku krokach, to wstaw poniżej kolejny symbol wejścia prostego I i powiąż go z blokiem RS kolejnego kroku, w którym wyjście Q jest aktywne itd. Na koniec połącz wstawione symbole wejść prostych I z linią zasilającą i danym wyjściem Q. Pamiętaj, że w przypadku timera Asynchronous Pulse Generator, używanego do realizacji migania lampek H, symbol wyjścia Q odpowiadającego lampce H należy połączyć z symbolem wejścia prostego I powiązanego z timerem Asynchronous Pulse Generator, a nie z krokiem, w którym ten timer jest aktywowany.
16. Jeżeli wcześniej wstawiłeś do programu timery lub liczniki, to na podstawie algorytmu SFC określ, w których krokach aktywowane są poszczególne timery, a w których krokach liczniki zliczają zdarzenia. Następnie postępuj identycznie jak powyżej w przypadku wyjść Q. Pamiętaj, aby w przypadku licznika określić też krok, w którym resetowany jest stan licznika i uwzględnić to w programie.

Gotowy program w języku LD

