

5

Podstawy programowania sterowników PLC

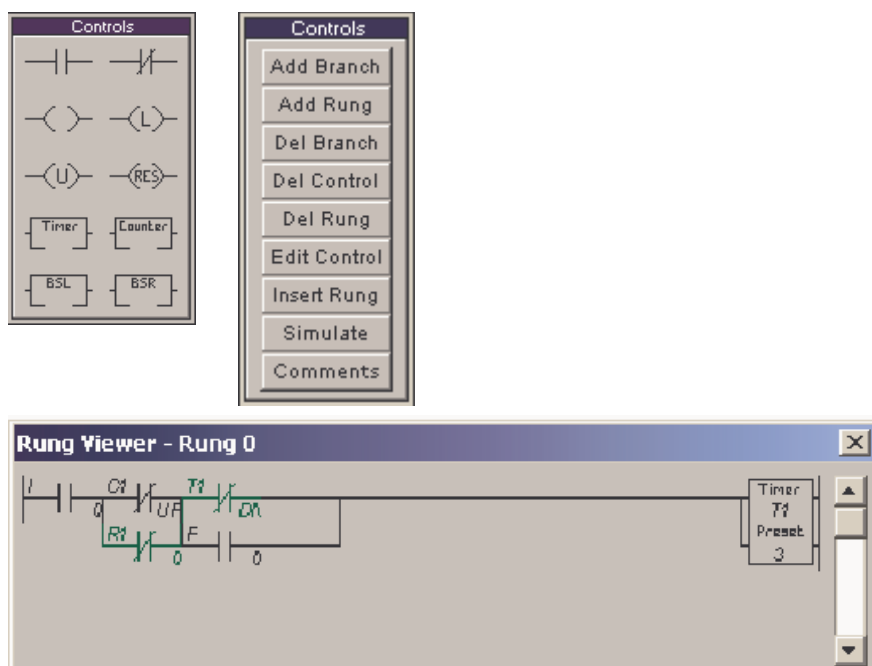
Cele ćwiczenia

1. Zrozumienie istoty programowania sterowników PLC
2. Zapoznanie się z podstawowymi metodami programowania

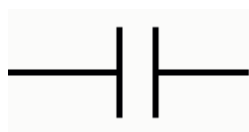
Opis programu LADSIM.

Pakiet LADSIM jest w pełni funkcjonalnym edytorem programów w formie drabinkowej, (LAD) symulując jednocześnie działanie sterownika programowalnego PLC. Zawiera on wszystkie podstawowe funkcje wykorzystywane podczas tworzenia oprogramowania sterującego sterowników PLC. Przy pomocy tzw. styków i cewek oraz timerów, liczników, flag i rejestrów przesuwanych można łatwo i szybko stworzyć programy drabinkowe.

Po uruchomieniu programu powinny być widoczne 3 okienka, dwa z nich nazywają się CONTROLS, jedno ma nazwę LADDER LOGIC DIAGRAM.



LADDER LOGIC DIAGRAM to główne okno programu, w nim ustawiamy szczeble diagramu, a na tych szczeblach kontrolki logiczne. Po lewej stronie szczebla znajdują się kontrolki wejścia (źródła sygnałów), po prawej wyjścia (czyli efekty pojawienia się (lub nie) sygnału na wejściu). Żeby wstawić na szczebel kontrolki, musimy posłużyć się oknem CONTROLS z 10 symbolami graficznymi przedstawiającymi dostępne w programie kontrolki. Poniżej kolejno wszystkie wraz z opisem:



Jest to czujnik wejścia, nazywa się NOC (notify on contact czyli powiadom w przypadku aktywacji). Kontrolki wstawiamy techniką przeciągnij-i-upuść. Po przeciągnięciu na RUNG0 (szczebel zerowy który dla nas jest pierwszy), pojawia się nowe okno gdzie mamy do wyboru źródło sygnału. Naszym źródłem mogą być wszystkie symulowane przez LADSIM urządzenia, czyli:

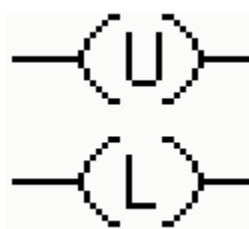
- **Inputs**, czyli czujniki które aktywowane są poprzez pewne zdarzenia, np: fotokomórka, wciśnięcie przycisku itp.
- **Outputs**, czyli urządzenia. Włączamy je sami, lub wykonuje się to automatycznie, po zaistnieniu pewnych okoliczności, np. przesuwająca się linia produkcyjna będzie miała aktywny sygnał pracy, czyli może być użyta przez inne urządzenie (np. podajnik kartonów dostarcza je na linię tylko gdy ten sygnał jest aktywny). Może to też być jakaś blokada, np. zatrzymująca karton przed urządzeniem zaklejającym je, do czasu gdy urządzenie zaklei poprzedni karton. W tym przypadku sygnał będzie aktywny w stanie wychylenia blokady.
- **Counters**, czyli liczniki, w tym wypadku mamy do czynienia z sygnałami DN i UP. Liczniki w tym programie mogą zliczać w górę do pewnej wartości, lub w dół do osiągnięcia zera. Sygnał DN oznacza zliczenie do zera, a UP osiągnięcie zadanej wartości. Szerzej o licznikach jeszcze powiemy.
- **Timers**, czyli coś w rodzaju zegarka. Źródłem wejścia może być tu sygnał DN, czyli coś w stylu 'alarmu' po upływie określonego czasu. Szerzej w dalszej części.
- **Flags**, czyli znaczniki lub coś w rodzaju zmiennych logicznych. Jeśli są aktywne, czyli ustawione, uaktywnią nasze wejście.
- **Register Bits**, coś w stylu bardziej rozbudowanej flagi, tyle że z szesnastoma miejscami na wartość logiczną. Szerzej w dalszej części.



Jest to identyczny jak poprzedni czujnik wejścia, tyle że jest on aktywowany gdy NIE są aktywne wymienione powyżej sygnały.



Urządzenie wyjściowe OP, czyli w praktyce co ma się stać gdy czujnik wejścia będzie aktywowany. Po przeciągnięciu na szczebel, pojawia się okno tak jak w przypadku czujnika wejścia, tyle że do wyboru mamy tylko OUTPUTS, FLAGS i REGISTER BITS. Zatem możemy aktywować urządzenie, ustawić flagę albo dowolny bit rejestru.



Te dwa urządzenia wyjściowe łączą się ze sobą, dlatego omówimy je razem. Ten z literką L w środku to OPL (output locked) czyli urządzenie zablokowane, natomiast ten z literką U to OPU (output unlocked) czyli odblokowanie urządzenia. Różnicę między zwykłym urządzeniem wyjściowym a tym blokującym najlepiej wyjaśnić na przykładzie: mamy wyjazd z parkingu, 10 metrów przed barierką jest fotokomórka która aktywuje podnoszenie barierki, zatem musimy mieć

podniesioną barierkę nie w TRAKCIE aktywowania fotokomórki (czyli gdy przejeżdżamy koło niej samochodem) a po tym fakcie, zatem po aktywacji fotokomórki podnosimy (blokujemy) barierkę i potem nie interesuje nas że wyjedziemy poza obszar fotokomórki i będzie ona nieaktywna, bo barierka jest podniesiona. Czyli używamy OPL, gdybyśmy używali OP barierka byłaby podniesiona tylko gdy będziemy 10 metrów od niej co nie ma sensu. Wyjaśniłem to tak dokładnie bo to dość istotne przy rozwiązywaniu zadań w tym programie. Jeśli natomiast chcemy odblokować urządzenie, np. wyjechaliśmy z parkingu i 3 metry dalej aktywowaliśmy inną fotokomórkę, to używamy kontrolki OPU i barierka opadnie jeśli była uniesiona lub nie stanie się nic jeśli była opuszczona (to duże uproszczenie ale nie w tym rzecz, chodzi o pokazanie różnicy między dwoma typami wyjść).



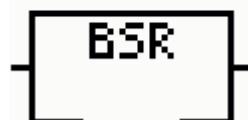
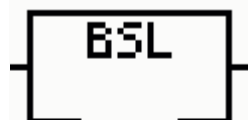
Jest to wyjście resetujące, mamy do wyboru ustawić TIME albo COUNTER w pozycji wyjściowej, stosuje się to po użyciu tych urządzeń. Możemy też wyzerować wybrany bit rejestru.



Zegar, stoper, licznik czasu. W tym programie jest traktowany jak urządzenie wyjściowe, czyli po prostu uruchamia się odliczanie. Czas po którym nastąpi 'alarm' czyli sygnał TIMER\DN, ustawia się w okienku które pojawi się po przeciągnięciu tej ikonki na szczebel. UWAGA: TIMER ustawiamy na osobnym szczeblu, tzn. musi być on jedynym urządzeniem wyjściowym na tym szczeblu.



Licznik, możemy w nim ustawić czy ma zliczać w dół czy w górę (UP\DN), oraz wpisać dwie wartości, PRESET czyli wartość do której będzie liczył jeśli wybraliśmy UP, oraz ACCUMULATOR czyli wartość początkowa. Jeśli wybraliśmy UP, od tej wartości zacznie się zliczanie w górę aż do osiągnięcia PERSET, albo jeśli wybraliśmy DN, od tej wartości zliczać będzie licznik do zera. To właśnie wartość ACCUMULATOR przyjmuje licznik po zresetowaniu. Również wymaga umieszczenia na osobnym szczeblu.



Przesunięcie bitów rejestru w lewo lub prawo. W praktyce stosuje się je w szybkich taśmach transportowych, np. do ustalania wadliwego produktu, albo niedokładnie wykonanej pracy, np. źle dokręcony śoik. Z uwagi na ograniczenia programu LADSIM jest to raczej bezużyteczne

Ostatnim oknem do omówienia jest drugie okno CONTROLS. Zawiera ono opcje, pomocne w edytowaniu szczebli diagramu. Są to kolejno:

- ADD BRANCH, dodaje rozgałęzienie, przydatne gdy np. kilka różnych stanów wejścia aktywuje to samo wyjście, lub odwrotnie, jedno wejście uruchamia kilka niezależnych wyjść. Dzieje się to równoległe, oczywiście w obrębie danego rozgałęzienia. W praktyce nieprzydatne w tym programie.
- ADD RUNG, dodaje nowy pusty szczebel.
- DEL BRANCH, kasuje rozgałęzienie.
- DEL CONTROL, kasuje wybraną kontrolkę.
- DEL RUNG, kasuje cały szczebel wraz z zawartością.

- EDIT CONTROL, przywołuje takie same okienko które pojawia się po upuszczeniu kontrolki na szczebel, można w nim zmienić źródło lub cel sygnału. UWAGA: to samo można uzyskać poprzez dwukrotne kliknięcie na kontrolce.
- INSERT RUNG, wstawia szczebel pomiędzy dowolne inne, przydatne przy np. dodawaniu pewnych dodatkowych opcji do poszczególnych części diagramu.
- SIMULATE, włącza okno debuga. Omówię tę opcję na końcu opisu programu.
- COMMENTS, pojawiają się pola tekstowe pod każdym szczeblem, gdzie możemy opisać co dany kawałek diagramu ma znaczyć w praktyce.

Po wstawieniu własnych sekwencji czas zobaczyć ich działanie w praktyce. W tym celu z menu SIMULATE wybieramy odpowiednie ćwiczenie. Jest ich osiem. Po wyborze ćwiczenia, ukazują się nowe okna, główne okno ćwiczenia, okno RUNG VIEWER oraz I/O BROWSER. Główne okno ćwiczenia to po prostu wizualizacja pewnych działań, na tym oknie zaczynamy pracę układu. W każdym z tych ćwiczeń znajduje się opis jak to zrobić, np. wrzucenie monety w ćwiczeniu DRINKS MACHINE czy kliknięcie na samochód w ćwiczeniu PARKING. Okno RUNG VIEWER pozwala obejrzeć każdy ze szczebli diagramu. Program podświetla na każdym szczeblu aktualnie aktywne elementy, co może być pomocne przy określaniu błędów programu.

Okno I/O BROWSER to klucz do działania z programem LADSIM. W nim mamy możliwość ręcznego ustawiania dowolnych czujników wejścia(IP), jako aktywnych. Jest to niezbędne w większości ćwiczeń, o czym szerzej przy wprowadzeniu do nich. Ponadto mamy możliwość poodglądu stanu wszystkich innych urządzeń (każde z nich ma swoją zakładkę). Dodatkowo na dole tego okna znajdują się przyciski sterujące pracą symulacji. Od lewej są to:

- przycisk RESTART SIMULATION (opis pokazuje się po najechaniu myszą nad dany przycisk) powoduje zresetowanie całej symulacji,
- przycisk PLAY SIMULATION wprawia w ruch symulację, aktywny dopiero po wybraniu przycisku PAUSE,
- przycisk STEP WHOLE DIAGRAM czyli przejście wszystkich szczebli diagramu. Jest to po prostu jeden cykl pracy układu,
- przycisk STEP SINGLE RUNG, czyli pozwolenie na zadziałanie tylko następnego szczebla diagramu, pomocne przy lokalizacji błędu, tzn wykrycia momentu jego powstawania,
- przycisk PAUSE, wstrzymuje symulację dzięki czemu możemy wybrać dwie poprzednie opcje (nie dostępne w trybie ciągłego działania),
- przycisk STOP, kończy symulację i przenosi nas do edytora szczebli diagramu.

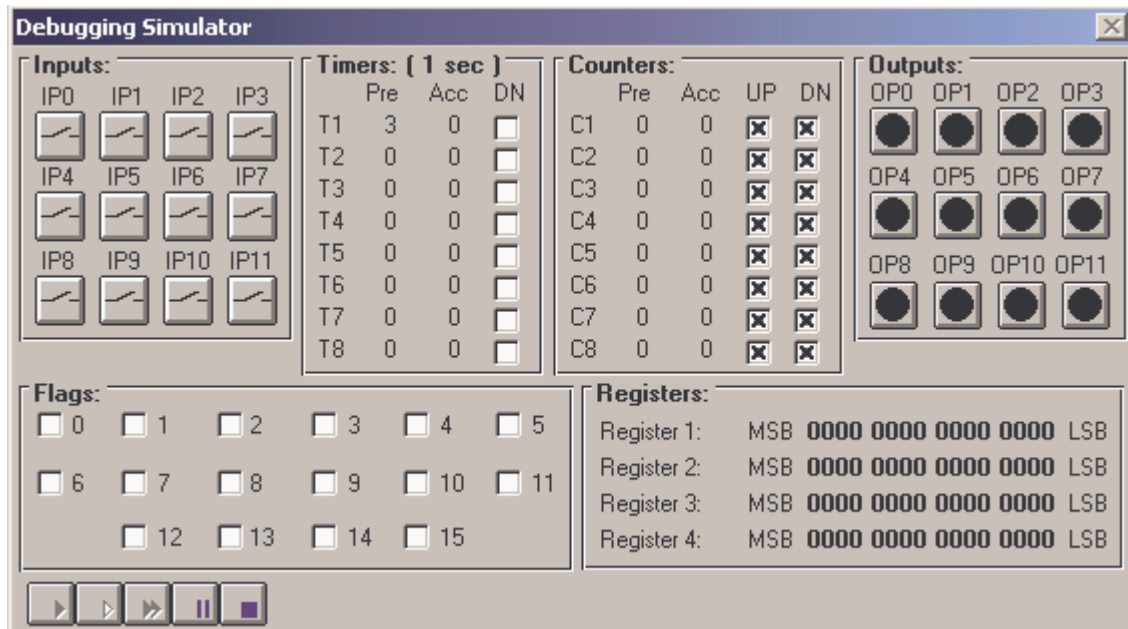
Ponadto w menu SIMULATE znajduje się opcja DEBUGGER. Teoretycznie jest to ułatwienie w znajdowaniu błędów diagramu, jednak przydatność tego modułu jest nikła. Najgorszą wadą tej opcji jest fakt iż nie widać ekranu symulacji przez co trzeba pamiętać do czego służą wszystkie czujniki i co robią wszystkie urządzenia. W zasadzie opcja DEBUGGER to to samo co okienko I/O BROWSER podczas symulacji tyle że wszystkie urządzenia rozmieszczone są w jednym oknie, przez co nie trzeba zmieniać zakładek, oprócz tego nie ma między nimi różnicy.

Pakiet LADSIM zawiera kilka przygotowanych symulacji rzeczywistych procesów, które pozwalają sprawdzić umiejętność programowania drabinkowego. W każdym przypadku należy tak oprogramować sterownik, by zapewnić pełną funkcjonalność prezentowanego układu. W pakiecie przygotowano następujące symulacje: Traffic Light – światła uliczne, Annunciator, Car Park – parkowanie pojazdów, Lift - winda, Drinks Machine – automat do wydawania napojów, Packing Line - sortowanie paczek, Bottling Plant – butelkowanie i paczkowanie oraz Industrial Control Trainer - linia technologiczna.

Testowanie programów

Pakiet LADSIM pozwala na sprawdzenie działania stworzonych programów. Służy do tego tryb debuggera, oraz tryb symulacji.

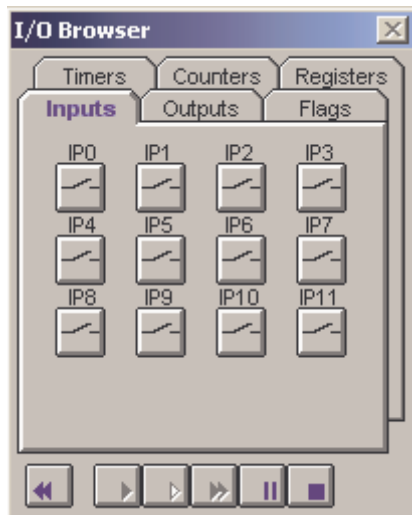
Tryb debuggera (menu: SIMULATE->DEBUGGER lub przycisk SIMULATE)



Tryb ten daje dostęp do wszystkich wejść wykorzystanych w programie – można dowolnie ustawiać ich stany, by sprawdzić reakcje zaprogramowanego sterownika. Uzyskane w ten sposób informacje pomogą wykryć ewentualne błędy programu. Dla ułatwienia pracy dostępny jest podgląd schematu.

Tryb symulacji (menu: SIMULATE->wybrany układ)

Do sprawdzenia działania napisanych programów służy również tryb symulacji. W tym trybie na ekranie zilustrowany jest oprogramowywany układ z wyszczególnionymi wejściami i wyjściami. Sprawdzenia funkcjonowania programu dokonuje się przez pobudzenie wejść na ilustracji. Sposób reakcji układu jest widoczny na ekranie. Dodatkowo jest możliwość podglądu stanów poszczególnych elementów wykorzystanych w programie. Służy do tego specjalne okno podzielone na zakładki odpowiadające rodzajom elementów.



Dla ułatwienia pracy dostępny jest także podgląd schematu.

W obu trybach w oknach podglądu stanów elementów jest umieszczony zestaw przycisków służący do kontroli przebiegu symulacji.



Funkcje przycisków w kolejności od lewej strony:

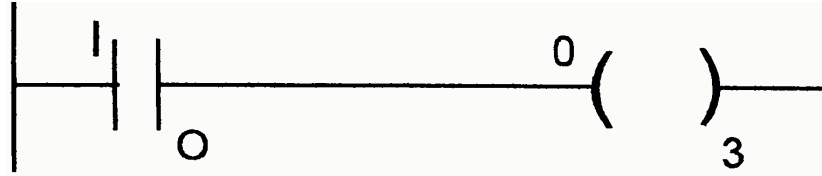
- cofnięcie do początku symulacji,
- uruchomienie symulacji (wejście do jednego z trybów testowania automatycznie powoduje uruchomienie symulacji, niezależnie od tego, czy został stworzony jakikolwiek program),
- symulacja „krok po kroku”,
- tryb krokowy – po jednym szczeblu,
- wstrzymanie symulacji,
- zatrzymanie symulacji – powrót do ekranu edycyjnego.

Kolejność czynności

Procedura tworzenia programu symulacji parkingu

1. Uruchom program.
2. Kliknij na opcji Simulate menu i wybierz Car Park. Otworzy się ekran przedstawiający parking dla którego będziesz tworzył program obsługi. Cel symulacji jak i źródła sygnałów w programie można odnaleźć w pliku pomocy programu LADSIM. Możesz się tam dostać naciskając klawisz F1. Po przeczytaniu pliku pomocy możesz powrócić do głównego programu. Na ekranie widzisz przeglądarkę sygnałów jak i okno z kodem programu w postaci drabinki.
3. Kliknij na samochód, wyświetla się dwie ikony rąk. Kliknij na lewą rękę a samochód zacznie jechać w lewo i będzie próbował się dostać na parking. Zostanie naruszony czujnik pierwszej bramki (IP0) ale sama bramka nie podniesie się. Tak więc w pierwszej części programu musimy otwierać bramkę czujnikiem. W dole ekranu kontrolującego mamy przyciski podobne do odtwarzacza wideo. Kontrolują one działaniem programu nawet kiedy jest on pusty. Kliknij na stop i powrócisz do diagramu programu.
4. Czujnik bramki (IP0) w tej symulacji jest przygotowany na kontakt (NOC). Jeśli będziesz poruszał się wolno wskaźnikiem myszy nad symbolami, w oknie **Controls** zobaczysz, że **NOC** jest przedstawiony po lewej u góry. Przyciśnij przycisk myszy i przenieś symbol na lewą stronę szczebla 0 i puść przycisk myszy. Zostaniesz zapytany przy pomocy dialogu

Inputs o źródło sygnału. Chcemy, by było to IP0, więc kliknij na IP0 i pokaże się to w diagramie programu. Kiedy IP0 jest zamknięte, chcemy, ażeby bramka się otwierała, więc przenieś symbol oznaczający otwarcie na prawą stronę szczebla 0 i określ go jako OP3, czyli bramkę, klikając na OP3 w oknie dialogowym **OUTPUTS** a potem na OK. Pierwszy szczebel drabiny programu powinien wyglądać tak:

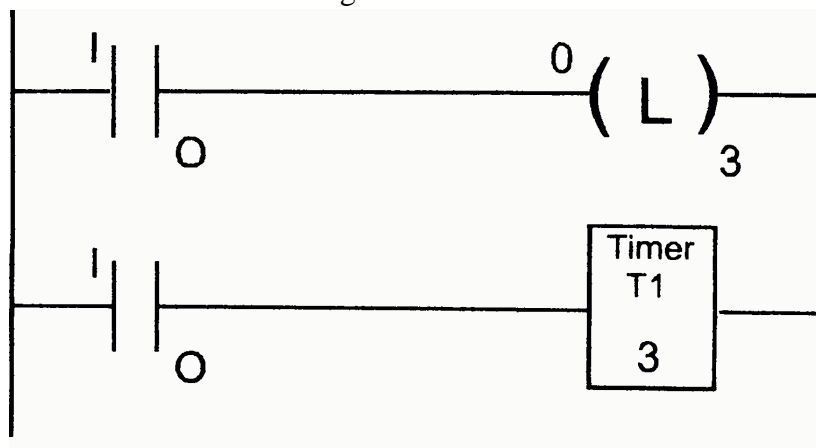


Przetestujmy program i przejdźmy do kroku 5.

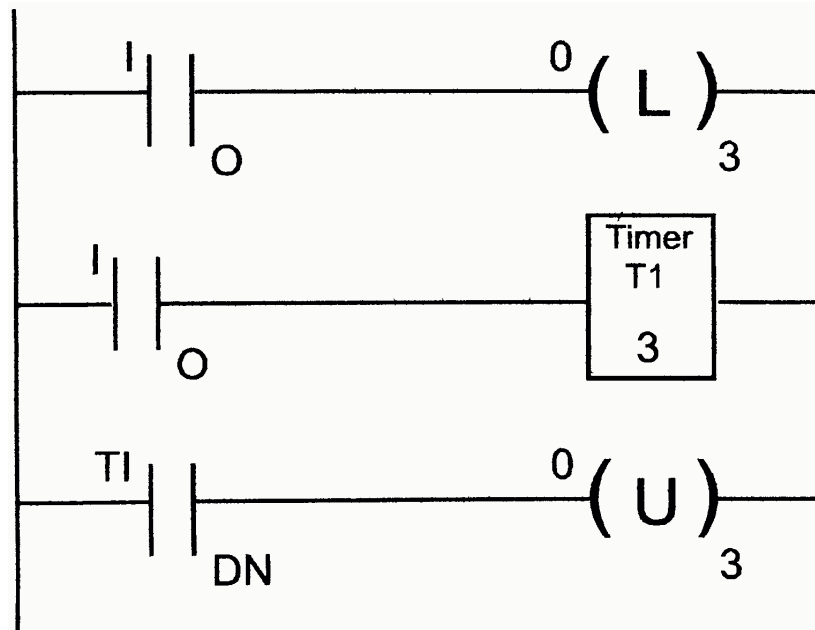
- Wybierz symulację Car Park ponownie i poślij samochód na parking. Wszystko dzieje się dobrze do momentu kiedy bariera zamyka się zbyt szybko i w ten sposób może uszkodzić samochód. Nie możemy dopuścić do takiej sytuacji, musimy zablokować zamykanie się bramki. Wracamy do edytora programu klikając na stop jak poprzednio. Z menu Controls wybierz Del Control. Zgodnie z instrukcją na dole usuń OP3. Zastąp elementem z blokadą (OPL). Diagram programu powinien wyglądać tak:



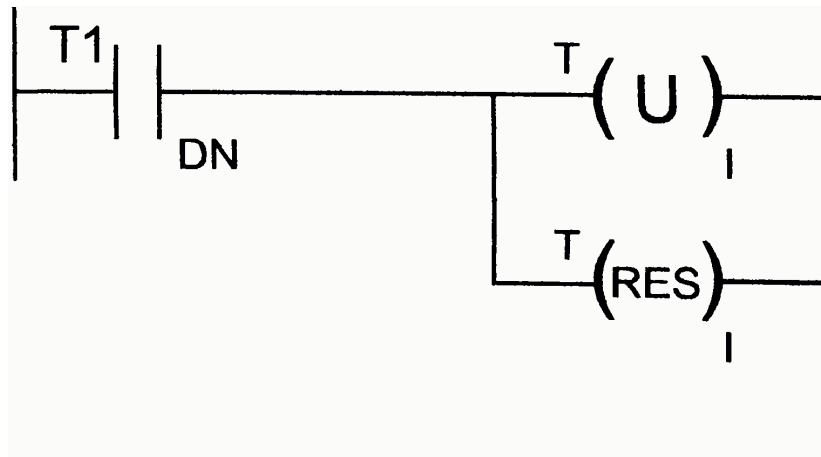
- Wypróbujmy tą wersję. Zobaczysz, że bramka jest otwarta i samochód nie zostaje uszkodzony. Jednak następne samochody mogą wjechać bez problemu i mogą też wyjechać poprzez wjazd. Wybierz samochód i sam zobacz co się stanie. Musimy dodać opcję zamykającą bramkę po pewnym czasie. W następnym kroku dodamy zegar kontrolujący zamykanie i zobaczymy jak działa ten element. (Wróć do edytora programu). Wybierz Add Rung by dodać kolejny szczebel diagramu. Chcemy, by zegar zaczął odmierzać w momencie, gdy czujnik bramki IP0 zostanie aktywowany, więc przenieś IP0 na następny szczebel po lewej stronie a po prawej dodaj element Timer. Wybierz T1 i ustaw opóźnienie na 3 sekundy w oknie dialogowym kiedy się wyświetli, kliknij na OK. Powinieneś mieć na ekranie coś takiego:



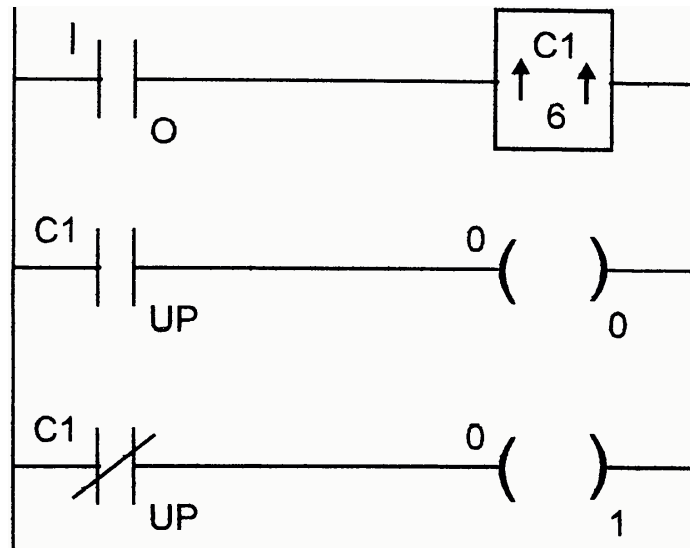
7. Uruchom symulację tak jak poprzednio ale uruchom najpierw Timer na oknie przeglądania sygnałów (I/O). Kiedy samochód aktywuje czujnik bramki wartość Accum rośnie od 0 do 3 i ustawiany jest bit DN. Mamy więc zegar odmierzający czas od aktywowania czujnika IP0 jaki ustawienie bitu T1/DN po 3 sekundach. Możemy dodać następny szczebel do diagramu programu który będzie zamykał bramkę, kiedy bit T1/DN zostanie ustawiony. Więc dodaj kolejny szczebel tak jak poprzednio, przenieś NOC na lewą stronę nadaj mu identyfikator T1/DN w oknie dialogowym. Z prawej strony dodaj OPU i nadaj mu identyfikator O/P3:



8. Przetestuj nowy program. Wszystko wydaje się być ok, ale działa tylko dla pierwszego samochodu, dlaczego? Możesz odgadnąć patrząc na diagram który jest wyświetlany podczas symulacji. Zauważysz, że wykonywane elementy są podświetlane w tym samym czasie, zauważysz że T1/DN nie jest zapalone, gdy pierwszy samochód wjeżdża na parking. Znacznik T1/DN zostaje aktywowany i żaden inny samochód nie może wjechać na parking. Możesz to zobaczyć przewijając symulację do tyłu i oglądając zdarzenie jeszcze raz. Rozwiązaniem problemu może być kasowanie T1/DN po zamknięciu bramki. Dodaj element Reset na szczebel 2 i nadaj mu identyfikator T1. Zobacz, że element Reset jest automatycznie łączony z szczeblem 2:

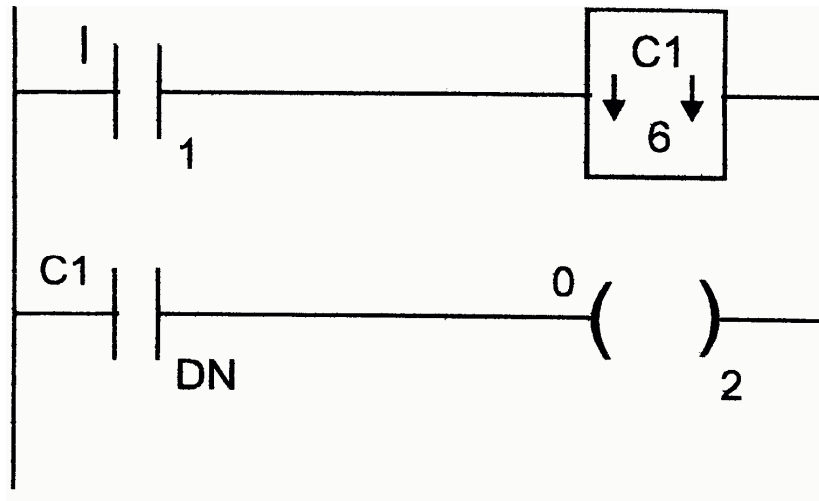


9. Wypróbuj program i zobaczysz, że wszystko działa dobrze. Jeśli nie – zmień czas opóźnienia dla T1. Gratulacje, rozwiązałeś problem wjazdu samochodu na parking. Teraz masz nowy problem: samochody są uwięzione na parkingu. Wybierz jeden z zaparkowanych samochodów i zobacz co się stanie. Musimy stworzyć podobny program otwierający bramkę wyjazdową. Jeśli się to uda mamy działający algorytm obsługi parkingu.
10. Powinieneś zauważyć, że parking ma wyświetlacz opisujący jego status. Następnym problemem jest kontrolowanie tego wyświetlacza. W programie LADSIM wartość licznika Counter 1 jest połączona bezpośrednio z wyświetlaczem. Jest to odzwierciedlenie często stosowanego w rzeczywistości łączenia licznika z elementem PLC.
11. Zatem jeśli użyjemy licznika Counter 1 (C1) do zliczania ile razy została otwarta bramka policzymy ile samochodów jest na parkingu. Dodatkowo jeśli ustalimy że w przypadku gdy licznik wskaże na 6 (bit C1/UP ma wartość 6) możemy zapalić sygnalizator pełny (Full OP0). W podobny sposób możemy kontrolować OP1, kiedy nie jest ustawiony sygnalizator SPACES (miejsca) jest zapełniony.

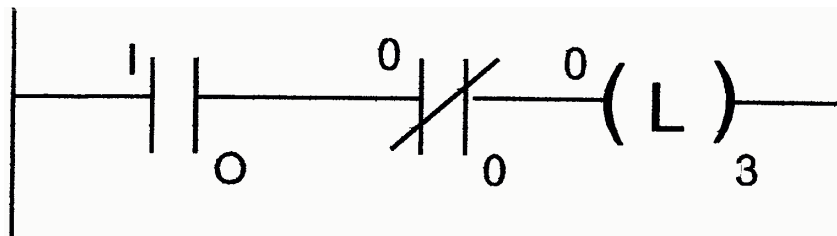


12. Żeby dodać powyższe funkcje musimy dodać kolejny szczebel, przenieść element IPO na lewo i licznik na prawo, nadać mu identyfikator C1. C1 powinien mieć wartość 6 i zliczać w górę (UP). Na następnym szczeblu powinniśmy dodać element NOC po lewej z identyfikatorem C1/UP i OP0 po prawej. Na koniec na kolejnym szczeblu NOC po lewej z identyfikatorem C1/UP a po prawej z identyfikatorem OP1:
13. Jeśli przetestujesz ten program zobaczysz że sygnalizatory FULL, SPACES i COUNT działają dobrze, ale tylko wtedy gdy samochody wjeżdżają na parking. Kiedy go

opuszczają wartości sygnalizatorów nie zmieniają się. Można to poprawić używając wartości sygnalizatorów przekazywanych przez czujnik bramki wyjazdowej, ale w odwrotną stronę. Możemy też użyć znacznika C1/DN do kontrolowania sygnalizatora pusty (EMPTY OP/2):

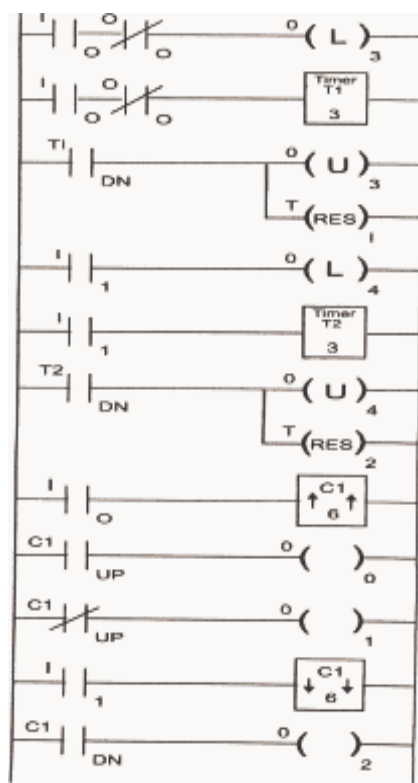


14. Kiedy wypróbujesz program zobaczysz, że wszystko działa dobrze do momentu kiedy pozwolimy wjechać na parking 7 samochodowi. Nie może on znaleźć miejsca więc wyjeżdża, a na wyświetlaczu pojawia się liczba 5. To jest oczywiście źle, na parkingu jest nadal 6 samochodów.
15. Żeby rozwiązać ten ostatni problem musimy zabronić wjazdu na parking samochodom kiedy jest on pełny. Musimy dodać warunek do pierwszego szczebla, kiedy wyświetlacz sygnalizuje pełny (FULL) by bramka się nie otwierała, przedstawione to jest poniżej:

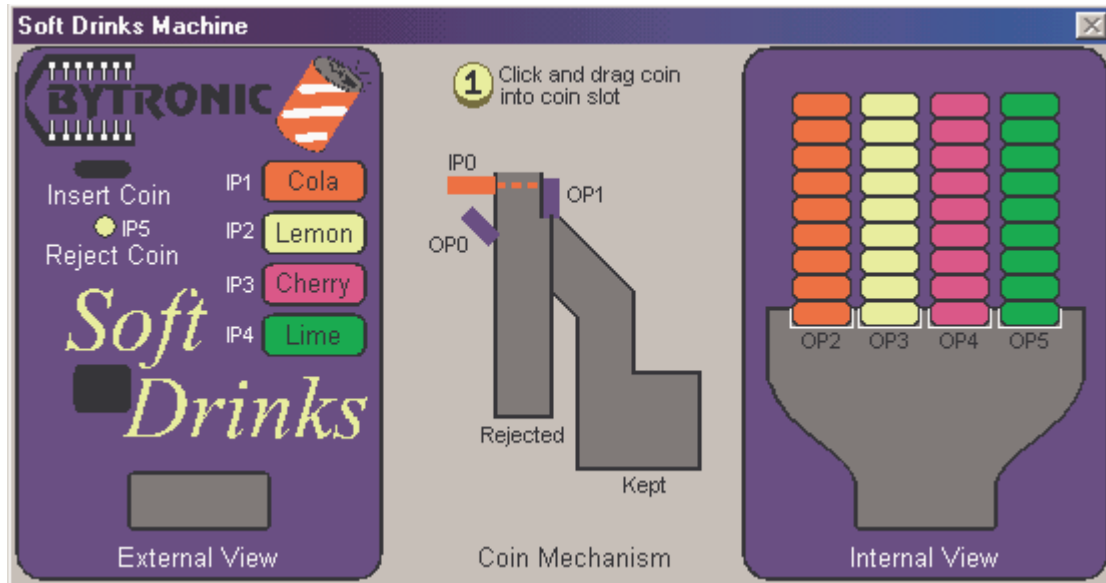


16. Musimy zmodyfikować drugi (rung 1) szczebel w podobny sposób.

Rozwiązanie problemu parkowania :



Opis ćwiczenia DRINKS MACHINE. Wygląd układu



Opis działania układu

Ćwiczenie to polega na obsłudze automatu do wydawania napojów. Klikamy na monecie (żółte kółeczko z jedynką) i przeciągamy ją do slotu INSERT COIN.

Opis oczyjnikowania i urządzeń wykonawczych :

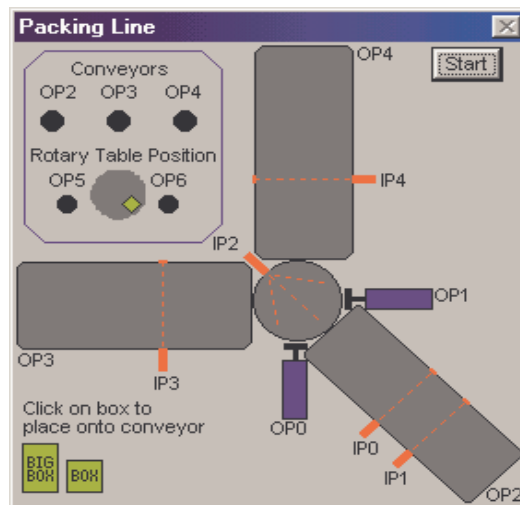
- IP0 - czujnik wrzutu monety,
- OP0 - blokada wylotu monety,
- OP1 - bramka blokująca wpadanie monet do 'kasy'. Potrzebne w przypadku anulowania wrzutu albo do wydania monety po dłuższym czasie nie wybrania napoju,
- OP2,OP3,OP4,OP5 - podajniki puszek z napojami,
- IP1,IP2,IP3,IP4 - czujniki wyboru odpowiedniego napoju,
- IP5 - czujnik rezygnacji z pobrania napoju.

Sekwencyjny opis działania maszyny :

- Wrzucić monetę,
- Czujnik IP0 zostaje aktywowany przez monetę co powoduje aktywację urządzeń OP0 i OP1 w celu przetrzymania monety aż do wyboru napoju lub rezygnacji,
- Jeśli aktywowany był czujnik IP5, odblokowujemy urządzenie OP0, koniec symulacji
- Jeśli aktywowany był któryś z czujników IP1,IP2,IP3,IP4, odblokowujemy urządzenie OP0 oraz odpowiedni podajnik napoju.
- Koniec.

Opis ćwiczenia PACKING LINE

Wygląd układu



Opis działania układu

Ćwiczenie polegające na sortowaniu paczek o różnych wymiarach i przesłaniu na odpowiednie dla nich taśmy produkcyjne. Przed przeciągnięciem pudełek na OP2 proponuje aktywować na stałe OP2, OP3, OP4, np. czujnikiem IP11. Inaczej kartony nie będą się poruszać.

Opis oczujnikowania i urządzeń wykonawczych :

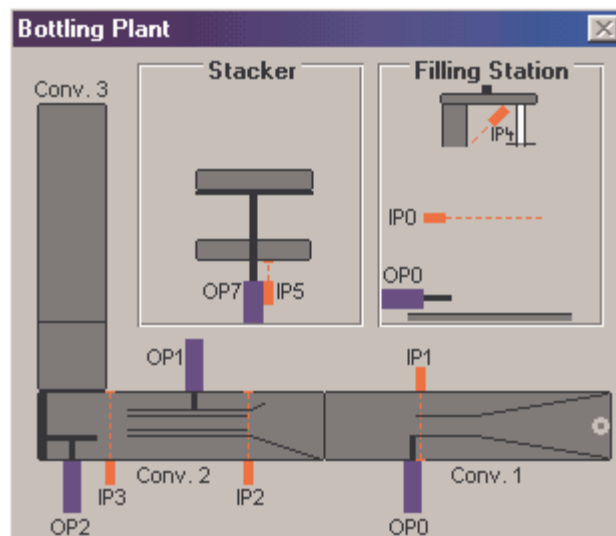
- OP2 - taśma podająca różne typy kartonów,
- OP3 - taśma na pudełka małe,
- OP4 - taśma na pudełka duże,
- OP0, IP1 - czujniki obecności kartonu na taśmie podającej,
- IP2 - czujnik obecności kartonu w punkcie skierowania na odpowiednią taśmę,
- OP5, OP6 - odpowiednio obrót w celu wypchnięcia kartonu małego(OP5) i dużego(OP6),
- OP0, OP1 - 'wypychacze' pudełek na odpowiednie taśmy,
- IP3, IP4 - czujnik obecności na właściwej sobie taśmie kartonu małego(IP3), oraz dużego(IP4).

Sekwencyjny opis działania urządzenia :

- pudełko poruszając się po linii OP2 aktywuje czujniki IP1 IP0,
- jeżeli czujniki IP0 i IP1 są aktywne równocześnie to mamy do czynienia z pudełkiem dużym, przeciwnym wypadku z małym,
- pudełko dotarło do urządzenia obracającego i aktywowało czujnik IP2,
- po aktywowaniu czujnika IP2 jeśli pudełko było duże aktywujemy urządzenie OP6, jeśli było małe aktywujemy urządzenie OP5,
- po zakończeniu obrotu aktywujemy urządzenie OP0 jeśli pudełko było duże, lub OP1 jeśli było małe.
- Po aktywacji IP3 (lub IP4) można zatrzymać taśmę OP3 (lub OP4), albo i nie, zależy to od nas.

Opis ćwiczenia BOTTLING PLANE 1.

Wygląd układu



Opis działania układu

Ćwiczenie polega na napełnieniu butelki mlekiem, nasadzeniu kapsla, dociśnięciu kapsla, wypchnięciu 3 razy po 3 butelki na platformę, opuszczenie tej platformy, oczekiwanie aż butelki zjadą z niej, a potem podniesienie jej by mogła przyjąć kolejne butelki. Na początek należy aktywować taśmy OP5 i OP6 czyli odpowiednio taśma CONV. 1 i CONV. 2.

Opis czujnikowania i urządzeń wykonawczych :

- IP1 - czujnik obecności butelki przy punkcie napełniania,
- OP0 - blokada przytrzymująca butelkę przy punkcie napełniania,
- IP0 - czujnik poziomu nalanego mleka w butelce,
- OP4, OP3 - UWAGA: nie widać tych urządzeń na obrazu symulacji, ale są to odpowiednio OP4 - napełnianie mlekiem butelki oraz OP3 - nałożenie kapsla,
- IP2 - czujnik obecności butelki przy punkcie wciskania kapsla,
- OP1 - urządzenie dociskające kapsel,
- IP3 - czujnik obecności butelki tuż przed 'wypychaczem' OP2 na platformę,
- OP2 - 'wypychacz' butelek na platformę,
- OP5 - platforma którą trzeba obniżyć żeby butelki mogły iść dalej,
- IP5 - czujnik obniżenia platformy OP5.

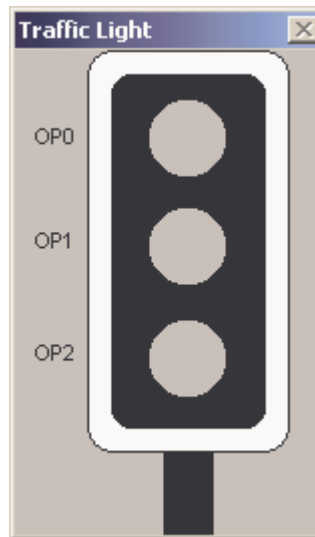
Sekwencyjny opis ćwiczenia:

- pusta butelka poruszając się po taśmie dociera do czujnika IP1, aktywując go,
- po aktywacji czujnika IP1 należy zablokować urządzenie OP1 przytrzymujące butelkę w punkcie napełniania (FILLING STATION)
- zablokowanie urządzenia OP1 jest sygnałem do rozpoczęcia napełniania butelki mlekiem (urządzenie OP4),
- po aktywacji czujnika IP0 należy przestać napełniać butelkę, oraz nałożyć kapsel (urządzenie OP3),
- aktywacja czujnika IP4 oznajmia że kapsel został nałożony, oraz że można zwolnić blokadę OP1, butelka z mlekiem posuwa się dalej,
- butelka dociera do czujnika IP2 i aktywuje go,
- aktywacja IP2 jest sygnałem inicjującym dla urządzenia OP1 (dociśnięcie kapsla), butelka posuwa się dalej,

- tuż przed dotarciem na ‘wypychacz’ (OP2) butelka aktywuje czujnik IP3,
- po każdym trzykrotnym aktywowaniu czujnika IP3 mamy pełen ‘wypychacz’, możemy wypchnąć butelki na platformę (tu uwaga, nie należy wypychać od razu po trzykrotnym aktywowaniu czujnika IP3, gdyż trzecia butelka nie zdąży wjechać na platformę),
- po trzykrotnym wypchnięciu butelek przez urządzenie OP2, możemy opuścić platformę OP7,
- opuszczona platforma aktywuje czujnik IP5, co jest sygnałem że butelki zaczynają zjeżdżać z platformy na końcową taśmę(CONV. 3),
- po zjechaniu wszystkich butelek musimy znów podnieść platformę OP7,
- koniec.

Opis ćwiczenia pt. TRAFFIC LIGHTS – Sygnalizacja uliczna

Wygląd układu



Opis działania układu

Układ jest typowym sygnalizatorem świetlnym. Przez 5 sekund świeci się światło czerwone. Przez następnie 5 sekund świecą się światło żółte i czerwone. Kolejne 5 sekund to światło zielone. Ostatnie 5 sekund to światło żółte. Potem cykl się powtarza.

Opis oczujnikowania i urządzeń wykonawczych :

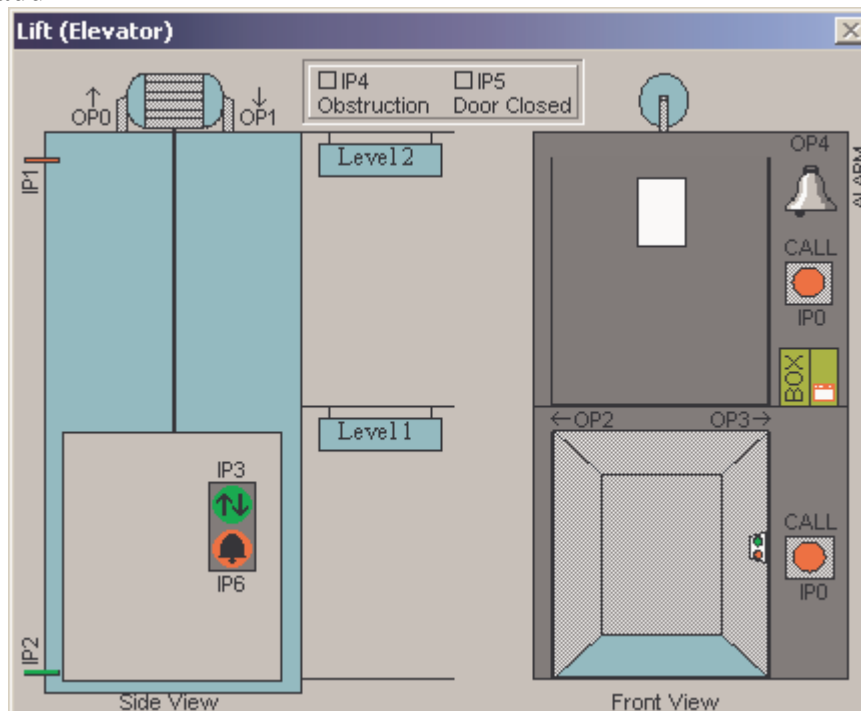
- IP1 -
- OP0 -
- IP0 -
- OP4,OP3 -
- IP2 -
- OP1 -
- IP3 -
- OP2 -
- OP5 -
- IP5 -

Sekwencyjny opis ćwiczenia:

- pusta butelka poruszając się po taśmie dociera do czujnika IP1, aktywując go,
- po aktywacji czujnika IP1 należy zablokować urządzenie OP1 przytrzymujące butelkę w punkcie napełniania (FILLING STATION)
- zablokowanie urządzenia OP1 jest sygnałem do rozpoczęcia napełniania butelki mlekiem (urządzenie OP4),
- po aktywacji czujnika IP0 należy przestać napełniać butelkę, oraz nałożyć kapsel (urządzenie OP3),
- aktywacja czujnika IP4 oznajmia że kapsel został nałożony, oraz że można zwolnić blokadę OP1, butelka z mlekiem posuwa się dalej,
- butelka dociera do czujnika IP2 i aktywuje go,
- aktywacja IP2 jest sygnałem inicjującym dla urządzenia OP1 (dociśnięcie kapsla), butelka posuwa się dalej,

Opis ćwiczenia pt. LIFT – WINDA

Wygląd układu



Opis działania układu

Winda po zgłoszeniu żądania (IP3 lub IP0) zamyka drzwi, jedzie na przeciwny poziom i otwiera drzwi. Winda posiada zabezpieczenie, które nie pozwala na ruszenie przy otwartych drzwiach (IP5 – drzwi zamknięte), zaś drzwi nie będą się zamykać, jeśli będą zablokowane pudełkiem (IP4). Do wezwania pomocy służy przycisk wewnątrz kabiny (IP6) – wciśnięty włącza alarm (OP4).

UWAGA:

- w układzie nie ma czujnika, który informowałby, że drzwi są całkowicie otwarte
- do jazdy oraz otwierania i zamykania drzwi zalecane jest wykorzystanie wyjścia z podtrzymaniem

Opis czujnikowania i urządzeń wykonawczych :

- IP1 -
- OP0 -
- IP0 -
- OP4,OP3 -
- IP2 -
- OP1 -
- IP3 -
- OP2 -
- OP5 -
- IP5 -

Sekwencyjny opis ćwiczenia:

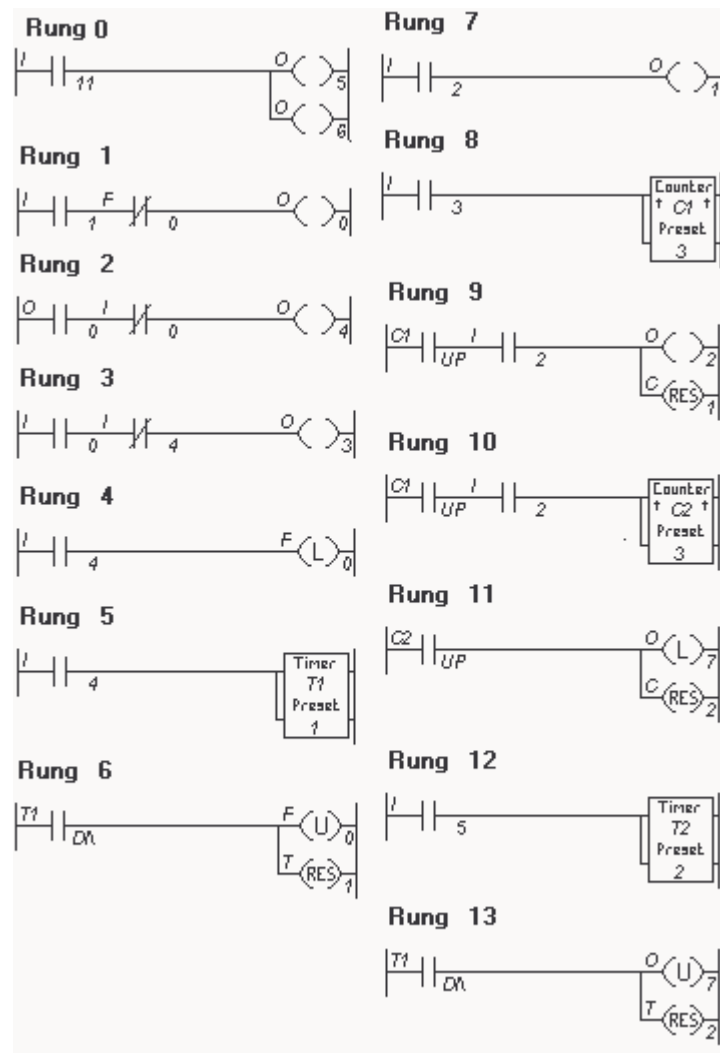
- pusta butelka poruszając się po taśmie dociera do czujnika IP1, aktywując go,
- po aktywacji czujnika IP1 należy zablokować urządzenie OP1 przytrzymując butelkę w punkcie napełniania (FILLING STATION)

- zablokowanie urządzenia OP1 jest sygnałem do rozpoczęcia napełniania butelki mlekiem (urządzenie OP4),
- po aktywacji czujnika IP0 należy przestać napełniać butelkę, oraz nałożyć kapsel (urządzenie OP3),
- aktywacja czujnika IP4 oznajmia że kapsel został nałożony, oraz że można zwolnić blokadę OP1, butelka z mlekiem posuwa się dalej,
- butelka dociera do czujnika IP2 i aktywuje go,
- aktywacja IP2 jest sygnałem inicjującym dla urządzenia OP1 (dociśnięcie kapsla), butelka posuwa się dalej,

4. Teraz tylko zatrzymujemy taśmę gdy kartony trochę się w niej przemieszczą, tzn gdy duży karton aktywuje czujnik IP4 a mały IP3. Kartony będą już dostatecznie daleko żeby zmieściły się następne kartony, a zarazem będą blisko siebie, co pozwoli przesunąć się taśmę produkcyjnej tylko gdy będą na niej kartony.

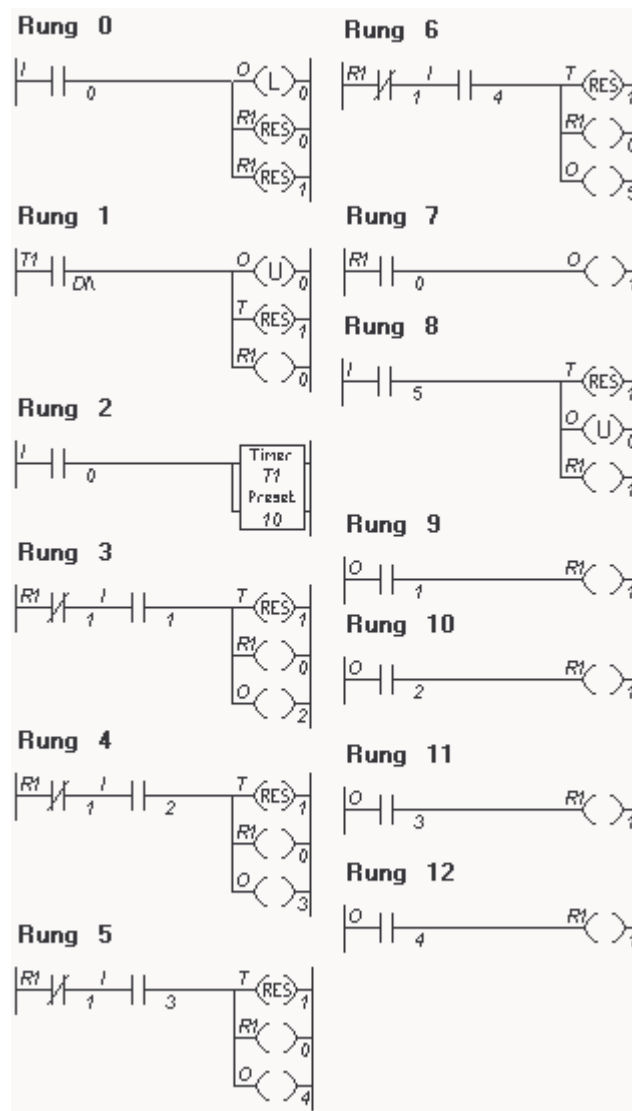


Przykładowe rozwiązanie ćwiczenia BOTTLING PLANE



Rozwiązanie to jest zgodne z sekwencyjnym opisem tego ćwiczenia.

Przykładowe rozwiązanie ćwiczenia DRINKS MACHINE



Rozwiązanie zgodne z opisem ćwiczenia, dodany 10 sekundowy czas oczekiwania, po którym przy braku wyboru ze strony użytkownika, automat zwraca monetę.