

Urządzenie służy do napełniania małych zbiorników- naboju (400g) smarem z beczki (50 / 180 kg).
Realizowane są dwie zasadnicze funkcje:

- 1- Obcinanie – siłownik z ostrzem, sterowany z przycisku, obcina zaślepkę na naboju (fabrycznie naboje dostarczane są zaślepione)
- 2- Napełnianie – nabój już bez zaślepki umieszczany jest w uchwycie, dociskany przez siłownik, wchodzi w gniazdo pompy która napełnia go smarem.

Dane są następujące sygnały:

In / out	Nazwa	Funkcja	Numer wejścia / wyjścia
I	NAP_OBEC	Czujnik obecności naboju w gnieździe napełniacza	1
I	NAP_PELNY	Czujnik pełnego naboju	3
I	NAP_MAX	Czujnik max wysunięcia SIŁ_1	2
I	CIECIE_OBEC	Czujnik obecności naboju w gnieździe obcinacza	4
I	SAFETY_1	Przycisk bezpieczeństwa (grzyb)	9
I	NAP_START_1	Przycisk startu napełniania 1	5
I	NAP_START_2	...	7
I	CIECIE_START_1	...	6
I	CIECIE_START_2	...	8
Q	POMP_L	Pompa obroty lewe (wstecz)	1
Q	POMP_P	Pompa obroty prawe (kierunek przepływu)	2
Q	SIŁ_1_IN	SIŁ_1 Wsuniecie	3
Q	SIŁ_1_OUT	SIŁ_1 Wysunięcie	4
Q	SIŁ_2_CIECIE	Cięcie	5
Q	SIŁ_3_DOCISK	Docisk przy cięciu	6

Algorytm działania sterownika urządzenia

Program realizuje równoległe 4 funkcje:

- 1- Funkcje bezpieczeństwa: Jakiegokolwiek działania efektorów maszyny, muszą zostać przerwane jeśli przerwany zostanie sygnał bezpieczeństwa SAFETY_1 (stan wysoki – dobrze, stan niski- zatrzymanie awaryjne)
- 2- Obcinanie: Obcinanie jest realizowane kiedy przyciski cięcie_start_1 i cięcie_start_2 są wciśnięte, jeśli przestaną być wciśnięte, proces jest przerywany. Po wciśnięciu przycisków, wysuwa się siłownik 3 – dociskający nabój w gnieździe, siłownik wchodząc w gniazdo, aktywuje czujnik obecności naboju CIECIE_OBEC (zasymulować to ustawiając sygnał ręcznie). Z pewnym opóźnieniem (500ms) wsuwa się siłownik 2 który realizuje obcinanie. Jeśli czujnik obecności naboju nie da stanu wysokiego to siłownik 3 nie wykona cięcia. Po zakończeniu obcinania operator puszcza przyciski i oba siłowniki powracają do pozycji początkowych.
- 3- Napełnianie: Napełnianie jest realizowane kiedy przyciski nap_start_1 i nap_start_2 zostaną wciśnięte, jeśli przestaną być wciśnięte, proces jest kontynuowany. Po wciśnięciu przycisków, wysuwa się siłownik 1 (SIŁ_1_out) – dociskający nabój w gnieździe, siłownik wchodząc w

gniazdo, aktywuje czujnik obecności naboju NAP_OBEC (zasymulować to ustawiając sygnał ręcznie). Kolejny etap następuje tylko wtedy kiedy siłownik _1 osiągnie pozycję maksymalnie wysuniętą – nabój zostanie do końca wsunięty w gniazdo, co sygnalizowane jest czujnikiem NAP_MAX (zasymulować to ustawiając sygnał ręcznie). Jeśli NAP_MAX i NAP_OBEC są w stanie wysokim, następuje pompowanie – sterowane jest wyjście POMP_P a Siłownik_1 jest nie sterowany (oba sygnały sterujące=0). Kiedy krańcówka NAP_PELNY da stan wysoki (zasymulować to ustawiając sygnał ręcznie), wiadomo że nabój jest pełen. Pompa przerywa pompowanie, następnie pompuje przez 500ms w kierunku przeciwnym aby cofnąć nieznacznie smar (POMP_L). Po tym, wsuwany jest siłownik 1 (SIŁ_1_IN) aby uwolnić pełny nabój z uchwytu.

- 4- Funkcja komunikacyjna: Sterownik LOGO posiada ekran. Powinien on komunikować stan maszyny zgodnie z tabelą:

Stan maszyny	Sygnalizacja
Zatrzymanie awaryjne	Czerwony ekran i stosowna wiadomość
Gotowość do pracy	Zielony ekran i stosowna wiadomość
Prowadzenie cyklu napełniania	Żółty ekran i stosowna wiadomość

Ponadto należy zaimplementować licznik napełnionych nabojów i jego stan komunikować na ekranie gotowości do pracy.

Sposób przygotowania zadania:

Proszę wszelkie wejścia i wyjścia komentować zgodnie z konwencją przedstawioną w zadaniu np.: Q2 (POMP_P). Bloki wejść i wyjść (I & Q) powinny znajdować się w miarę możliwości w lewym górnym rogu pola roboczego tak aby możliwe było komfortowe objęcie wszystkich tych bloków na ekranie bez przesadnego oddalania widoku. Wszystkie wejścia w trybie (switch). Podane parametry: częstotliwości i okresy oczekiwania- są istotne i proszę się ich trzymać.

Rada: Części programu odpowiedzialne za obcinanie, napełnianie, bezpieczeństwo i ekran mogą działać poniekąd jako osobne bloki wymieniające pomiędzy sobą nieliczne sygnały – warto je tak napisać, zwiększy to czytelność układu i ułatwi podział pracy.

Sekwencję zdarzeń można zrealizować uzależniając włączenie danej operacji od zakończenia wszystkich poprzednich operacji. Czy przydatny będzie 'latching relay'? może... :d

LAD jest dozwolony ale lepiej napisać programy w FBD.

Sygnał bezpieczeństwa powinien być zrealizowany:

stan wysoki -> praca normalna

stan niski-> awaria