

Politechnika Łódzka
Instytut Automatyki
Zakład Sterowania Robotów

Laboratorium sterowania robotów

Łódź 2009

Ćwiczenie A i B

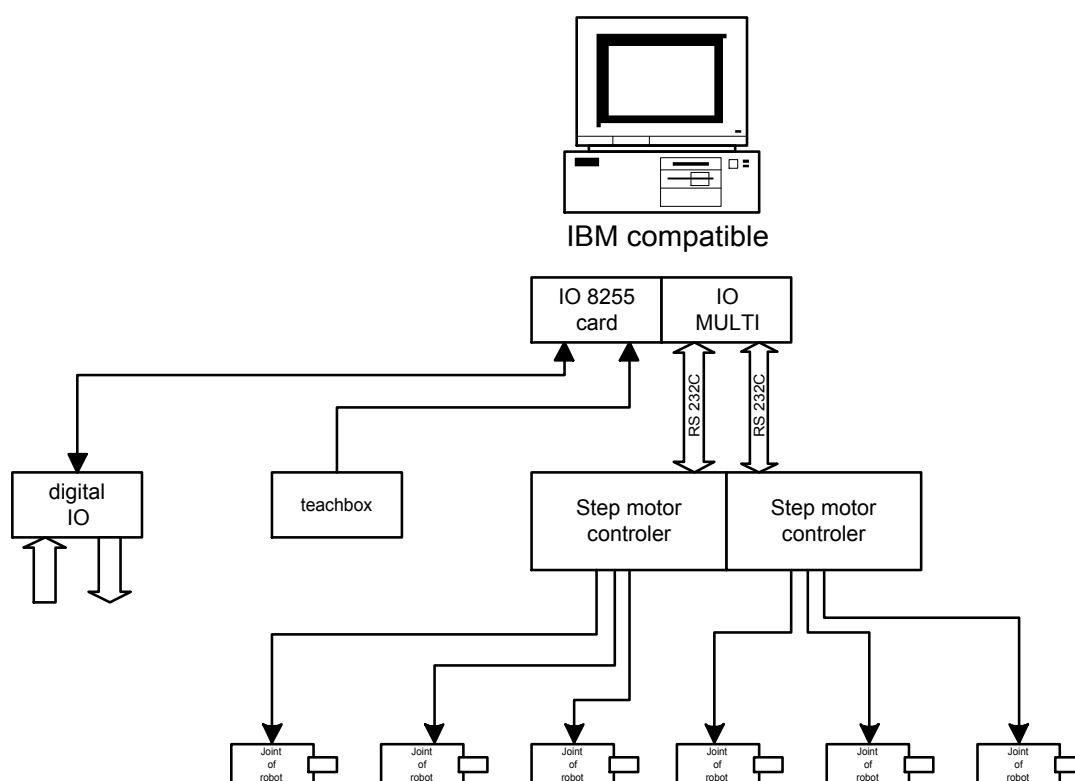
Cel ćwiczenia

Celem ćwiczenia jest pokazanie budowy, działania oraz zasad programowania prostych robotów i manipulatorów na przykładzie robotów dydaktycznych L1 i L2. Przed wykonaniem ćwiczenia należy zapoznać się z przeznaczeniem i danymi technicznymi tych robotów korzystając z niniejszego opracowania oraz instrukcji technicznych zawierających szczegółowy opis. Należy dokładnie sprawdzić możliwości ruchowe manipulatorów w trybie ręcznym (patrz akapit Software) ze szczególnym zwróceniem uwagi na zakresy ruchów i dozwolone prędkości.

Opis stanowiska

Układ napędowy i sterowania

Układ napędowy każdej osi zawiera sterownik, silnik krokowy oraz wyłącznik krańcowy. Układ pracuje w pętli otwartej, przez co wymaga każdorazowego bazowania (patrz Bazowanie robotów L1 i L2). Głównym elementem układu sterowania jest komputer klasy PC wyposażony w specjalizowane karty wejść/wyjść, klawiaturę oraz w przypadku robota L2 panel sterowania. Strukturę systemu sterowania robotów L1 i L2 pokazano na rys. 1.1.



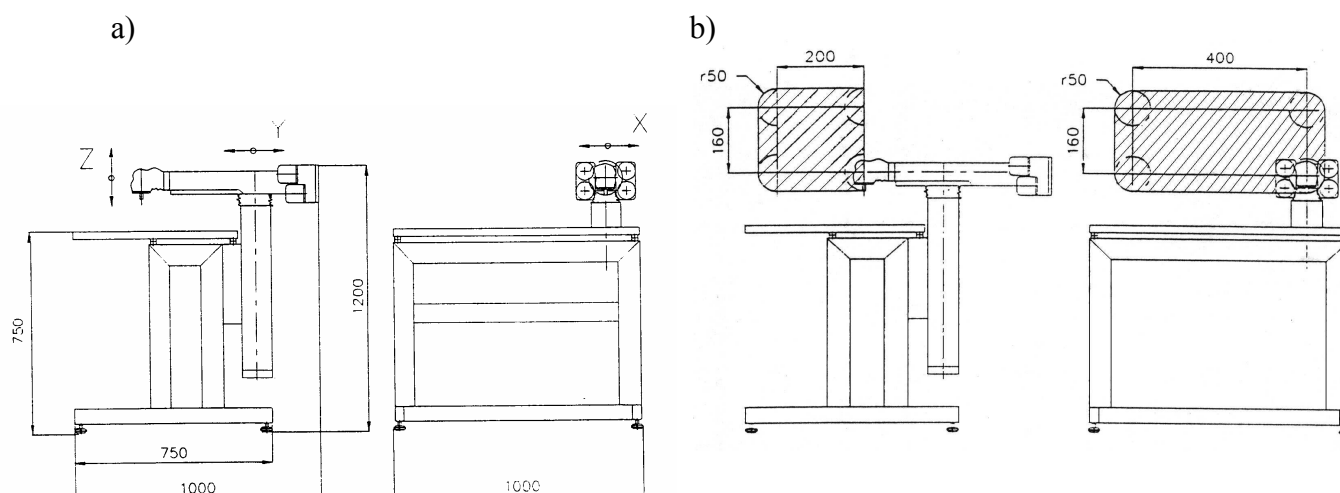
Rys. 1.1. System sterowania robotów przemysłowo-dydaktycznych

Część manipulacyjna

Robot L1 jest typowym robotem o kartezjańskim układzie kinematycznym ze sferycznym nadgarstkiem. Posiada 6 stopni swobody i elektryczny chwytak. Każda oś napędzana jest silnikiem krokowym o rozdzielczości 200 kroków/obróć, za pomocą układu przekazywania napędu.

Trzy pierwsze złącza liniowe tworzą strukturę pozycjonowania manipulatora. Ramiona robota są zorientowane prostopadle względem siebie. Dzięki temu istnieje możliwość niezależnego sterowania wzdłuż osi kartezjańskiego układu współrzędnych. Trzy pozostałe zespoły ruchu tworzą układ nadgarstkowy (osie α , β , γ) zapewniający ustawienie żądanej orientacji. Sprowadzenie napędów sterujących ruchem złącz obrotowych, do jednego miejsca, wymusiło zastosowanie sprzężenia mechanicznego na tych złączach. Jego wynikiem jest interakcja pomiędzy obrotami poszczególnych złącz. Sterowanie osią α powoduje jednoczesny obrót w pozostałych osiach β oraz γ . Uniknięcie niepożądanej interakcji możliwe jest poprzez jednoczesny ruch w dwóch osiach z uwzględnieniem kompensacji na sprzężenie mechaniczne.

Widok robota L1 oraz dane techniczne przedstawiono poniżej.



Rys. 1.2. Robot L: a) wymiary zewnętrzne, b) wymiary przestrzeni roboczej

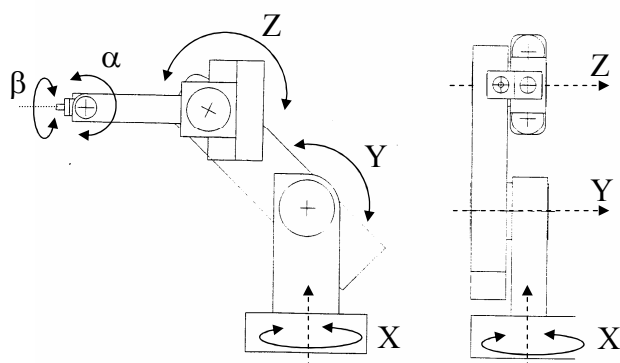
Tabela 1

Obciążenie maksymalne(podane przez producenta)	3.2 kg
Zakres złącza 1	400mm
Zakres złącza 2	300mm
Zakres złącza 3	160mm
Zakres złącza 4	$n \times 360^\circ$
Zakres złącza 5	180°
Zakres złącza 6	$n \times 360^\circ$
Prędkość max dla złączy liniowych	4 m/min
Prędkość max dla złączy obrotowych	0.5 rad/s
Błąd pozycjonowania (jw.).	$\pm 0.02\text{mm}$

Robot L2 zbudowany jest w oparciu o stawowy łańcuch kinematyczny typu Puma z 5 stopniami swobody. Efektor stanowi pudełko do odbierania detalu. Widok robota L2 przedstawiono na rys. 1.3, a dane techniczne zestawione są w tabeli 2.

Tabela 2

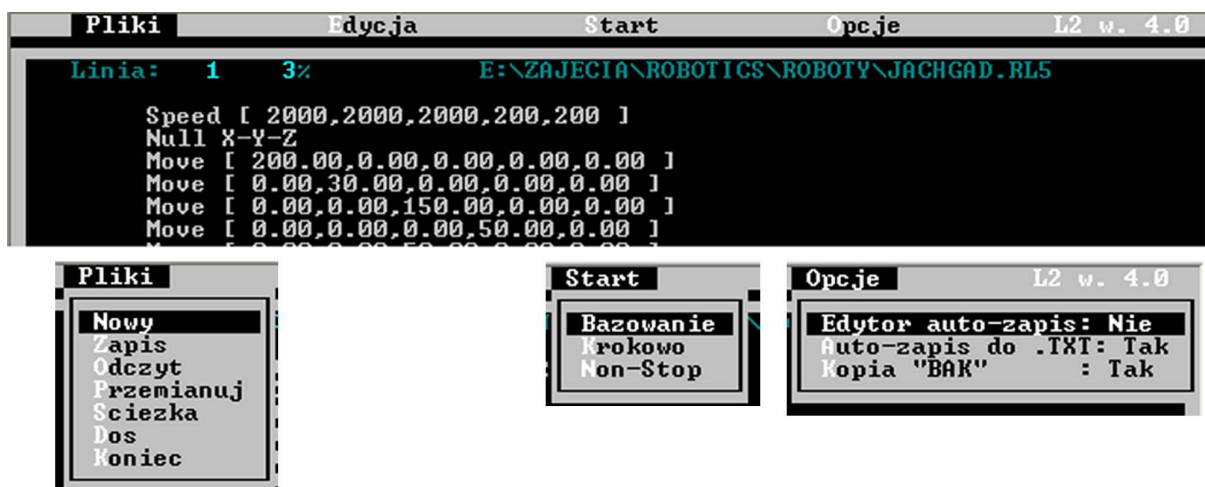
Obciążenie maksymalne (podane przez producenta)	3.2 kg
Zakres złącza 1	320°
Zakres złącza 2	100°
Zakres złącza 3	270°
Zakres złącza 4	170°
Zakres złącza 5	340°
Błąd bazowania	±0.02mm
Dokładność pozycjonowania(jw.)	±0.05mm.. ±0.15mm



Rys. 1.3. Robot L2 z zaznaczonymi nazwami złącz obrotowych.

Software

Program L1/L2 umożliwia tworzenie, edycję oraz uruchamianie programów sterowania robotów. Hierarchiczne menu oraz fragment okna edycyjnego prezentuje rys. 1.4.

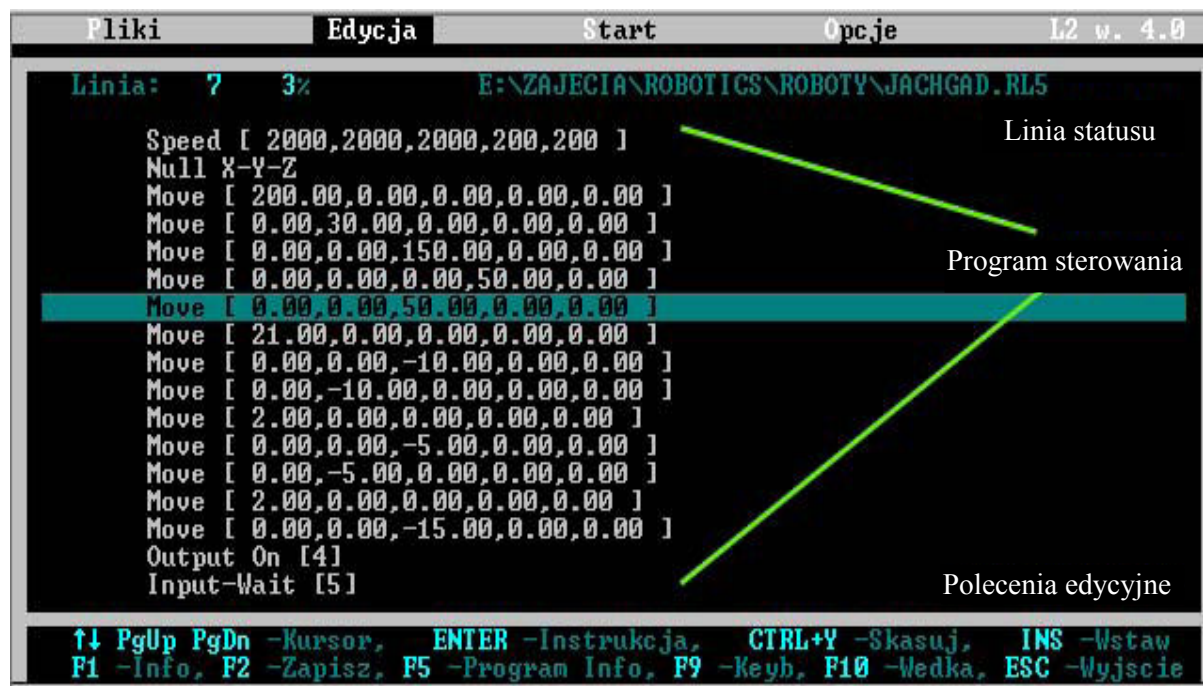


Rys. 1.4. Menu programu L2

Tabela 3 - opcje menu:

Menu	Funkcja
File	Operacje plikowe i dyskowe
Edit	Otwarcie okna edycyjnego. Wejście do sterowania ręcznego.
Start	Bazowanie oraz uruchamianie programu. Dodatkowo w L1 uproszczone sterowanie ręczne manipulatorem
Options	Funkcje dodatkowe

Wybranie menu Edit spowoduje otwarcie okna edycyjnego jak na rys. 1.5.

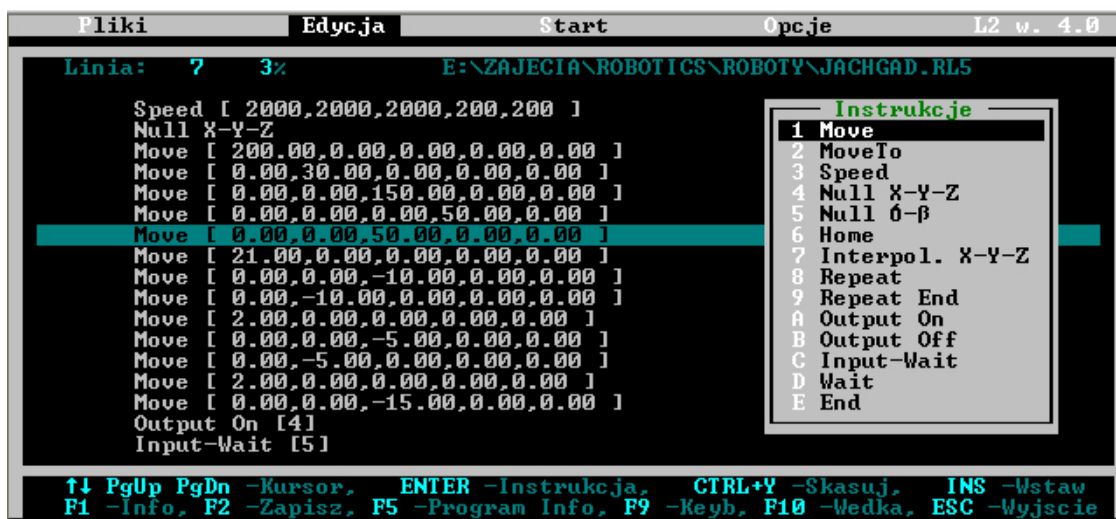


Rys. 1.5 Okno edycji

Polecenia edytora:

PgUp, PgDn, ↑↓	poruszanie kursorem
Enter	wywołanie menu instrukcji
Ctrl-Y	skasowanie podświetlonej linii
Ins	wstawienie linii
F1	informacja o edytowanym programie
F2	zapis
ESC	wyjście

Program zbudowany jest z makroinstrukcji i ich parametrów wstawianych z menu instrukcji (wywoływanego za pomocą ENTER). Widok okna z instrukcjami przedstawia rys. 1.6., zaś opis instrukcji zawarto w tabeli 4.



Rys. 1.6. Menu instrukcji

Tabela 4

Mnemonik	Parametry	Opis
MOVE	dx, dy, dz [mm or °] dα, dβ, dγ [°]	przesunięcie względne o wektor podany jako parametr
MOVE TO	j.w.	przesunięcie do pozycji określonej w ciele instrukcji, pozycja zerowa określona przez NULL
SPEED	sx, sy, sz, sα, sβ, sγ {21..4000}	ustawienie prędkości
NULL XYZ	-	ustalenie pozycji zerowej
NULL αβγ	-	ustalenie orientacji zerowej
INTERPOL XYZ	i {0,1,2} 0 - XY, 1 - XZ, 2 - YZ	ustalenie pary osi wykonujących ruch jednocześnie
INTERPOL αβγ	i {0,1,2} 0 - αβ, 1 - βγ, 2 - αγ	ustalenie pary osi wykonujących ruch jednocześnie
REPEAT	n	otwarcie pętli n powtórzeń
REPEAT END	-	zamknięcie pętli
OUTPUT ON	nr	ustawienie wyjścia nr
OUTPUT OFF	nr	wyzerowanie wyjścia nr
WAIT	t [sec]	pauza na t sekund
END	-	koniec programu

Przebieg ćwiczenia:

W celu rozpoczęcia ćwiczenia należy:

- uruchomić komputer,
- włączyć sterownik silników krokowych (pod komputerem),
- uruchomić program L1.exe (lub L2.exe) z katalogu C:\ROBOTY\L1.V4 (C:\ROBOTY\L2.V4),
- wybrać opcję sterowania przewodowego.

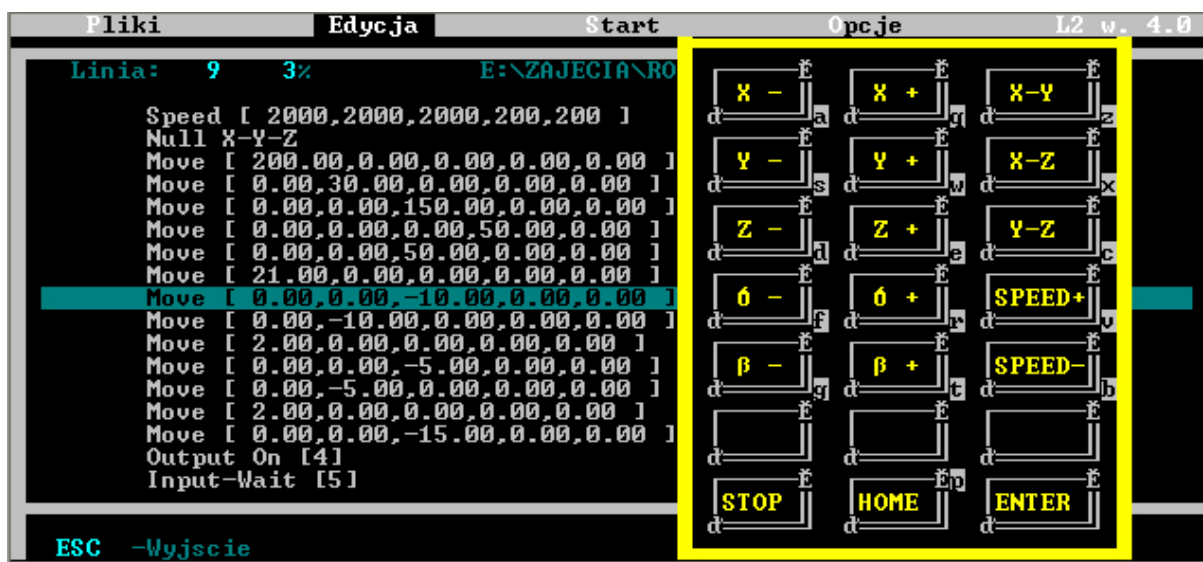
Bazowanie robotów L1 i L2:

Uwaga:

Przed synchronizacją robota należy upewnić się, że znajduje on się w przestrzeni roboczej, a w szczególności nie świeci się żadna kontrolka HOME na sterowniku silników krokowych. W przeciwnym przypadku należy sprowadzić robota do przestrzeni roboczej przy użyciu:

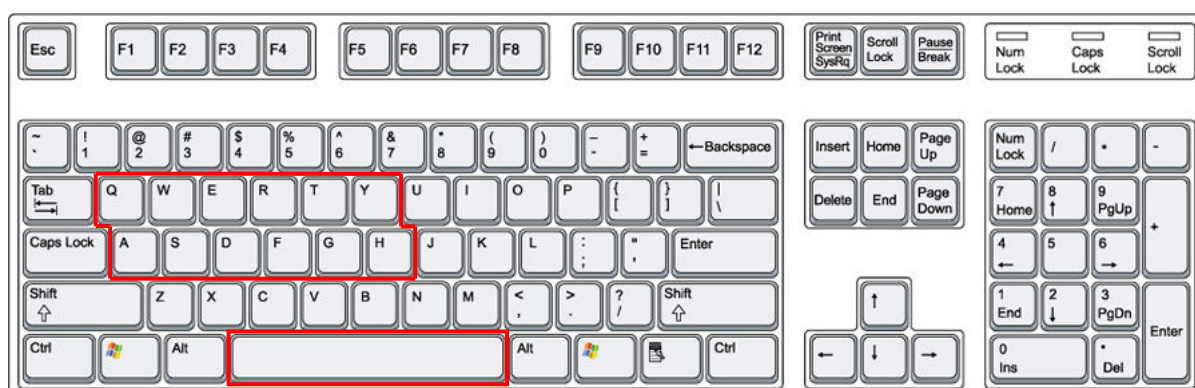
- teachboxu (dla robota L1 oraz L2).

Panel sterujący wywołuje się poprzez: Menu Edycja-->F9 (Rys 1.7).
















Rys 1.8. Widok teachboxu do ręcznego sterowania robotem.

Następnie naciskając odpowiednie klawisze dokonuje się sterowania ruchem w poszczególnych osiach Rysunek 1.8.



Rys 1.8. Klawisze używane w procedurze bazowania ręcznego dla robotów L1 i L2.

Oś	Przycisk	Funkcja po jednokrotnym naciśnięciu
X		Ruch w kierunku dodatnim dla osi X
		Ruch w kierunku ujemnym dla osi X
Y		Ruch w kierunku dodatnim dla osi Y
		Ruch w kierunku ujemnym dla osi Y
Z		Ruch w kierunku dodatnim dla osi Z
		Ruch w kierunku ujemnym dla osi X
α		Ruch w kierunku dodatnim dla osi α
		Ruch w kierunku ujemnym dla osi α
β		Ruch w kierunku dodatnim dla osi β
		Ruch w kierunku ujemnym dla osi β
γ		Ruch w kierunku dodatnim (oprócz L2)
		Ruch w kierunku ujemnym (oprócz L2)
X,Y,Z, α , β , γ		Zatrzymanie ruchu osi podczas sterowania ręcznego

Jeżeli na sterowniku nie pali się żadna kontrolka Home oznacza to, że robot znajduje się w dodatniej przestrzeni roboczej. Aby sprowadzić go do punktu bazowego - [0,0,0,0,0,0], należy dla każdej osi wymusić ruch w kierunku ujemnym. Robot będzie poruszał się do momentu najechania na czujnik krańcowy. Po osiągnięciu tego stanu, zatrzyma się, a na sterowniku zostanie zapalona dioda Home. Następnie należy wymusić ruch w kierunku dodatnim, do momentu ponownego zatrzymania (zjazdu z czujnika krańcowego).

W przypadku robota L2 niewłaściwa kolejność bazowania osi może powodować kolizję z elementami otoczenia, bądź stołem.

Podczas procedury ręcznego sprowadzania robota do punktu bazowego istnieje możliwość zatrzymania ruchu w osi poprzez naciśnięcie przycisku spacji.

b) opcji sterowania ręcznego z menu START i strzałek (dla L1).

!!

Ze względu na możliwość wystąpienia kolizji w trakcie bazowania automatycznego w robocie L2 oraz robocie L1, zabrania się używania polecenia: Menu **START --> Bazowanie**

Programowanie robotów L1 i L2:

Po prawidłowym ustawieniu robota w pozycji bazowej należy stworzyć program realizujący podane przez prowadzącego zadania. Makroinstrukcje tworzące strukturę programu wywołuje się za pomocą przycisku ENTER w menu Edycja. Na samym początku należy zdefiniować prędkość dla każdej z osi. W przypadku XYZ nie powinna ona przekraczać 2000 mm/min. Dla osi $\alpha, \beta, (\gamma)$ powinna być mniejsza od 150.

Kolejne dwie instrukcje Null XYZ oraz Null $\alpha\beta\gamma$ ustawiają pozycję zerową. Przesunięcie liniowe bądź kąt obrotu będą liczone względem tego punktu. Dalej w kolejnych liniijkach następuje definiowanie wartości przemieszczenia robota, sterowania wyjściami cyfrowymi lub oczekiwanie na pojawienie się sygnału na wejściach cyfrowych z zakresu [4-7].

Przed uruchomieniem realizacji programu należy sprowadzić robota do wcześniej zdefiniowanej pozycji zerowej. Pierwsze uruchomienie programu powinno być wykonywane w trybie pracy krokowej. Pozwala to na wychwycenie wszystkich błędów, które mogły pojawić się w czasie tworzenia programu.

W przypadku zaobserwowania sytuacji wykonywania przez robota błędnego ruchu, prowadzącego do kolizji z elementami otoczenia, należy zatrzymać pracę robota poprzez naciśnięcie przycisku STOP na sterowniku. W przypadku, gdy robot nie będzie reagował na sygnał zatrzymania, należy odłączyć zasilanie od sterowników przyciskami, które znajdują się na tylnej ścianie sterowników, obok przewodów zasilających.

Zadania do wykonania

1. Pierwszym zadaniem studentów jest zapoznanie się ze wszystkimi możliwościami manipulacyjnymi i komunikacyjnymi robotów L1 i L2 oraz sposobem tworzenia algorytmu ich działania. Dodatkowo należy poprawnie sprowadzić robota do konfiguracji bazowej.
2. Za pomocą wskaźnika umieszczonego w efektorze należy obrysować kształt wskazany przez prowadzącego.
3. Wynikiem ćwiczeń A i B powinny być dwa programy dla robotów L1 i L2. Oba roboty muszą ze sobą współpracować w następujący sposób: robot L1 przekazuje styropianowy detal do robota L2, który wrzuca otrzymany element do przygotowanego pudełka. Roboty powinny informować się wzajemnie o fazie pracy w jakiej się znajdują, by programy mogły działać zarówno w trybie pracy krokowej jak i automatycznej.

Ćwiczenie D

Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się ze sposobem programowania robotów przemysłowych za pomocą panelu sterowania.

Zadania do wykonania

W trakcie ćwiczenia należy uruchomić i zsynchronizować robota IRp-6, a następnie zapoznać się z obsługą panelu programowania. Pierwsze zadanie do wykonania polega na „obrysowaniu” w przestrzeni kształtki styropianowej wskazanej przez prowadzącego. Główna część ćwiczenia polega na ułożeniu i uruchomieniu programów przenoszenia niewielkich przedmiotów w przestrzeni roboczej manipulatora.

Opis stanowiska

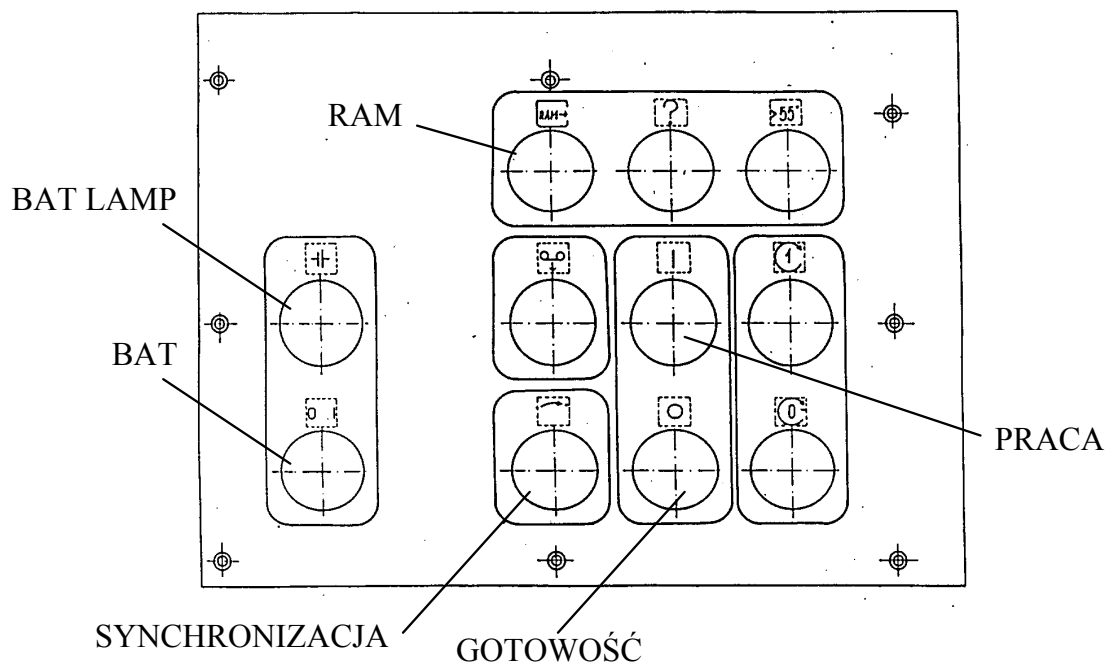


Rys 4.1. Stanowisko Programowania robota IRP-6

Uruchomienie i synchronizacja robota

Panel sterowania robotem (rys. 4.2.) wmontowany jest w szafę sterowniczą. Uruchomienie robota dokonywane jest przez włączenie stacyjki. Po jej włączeniu świecą się lampki sygnalizacyjne „SIEĆ” oraz „STOP AWARYJNY”. Wciśnięcie przycisku

„GOTOWOŚĆ” włącza zasilanie układu elektronicznego i jest sygnalizowane zapaleniem lampek „GOTOWOŚĆ” i „UTRATA PROGRAMU” oraz zgaśnięciem lampki „STOP AWARYJNY”. W tym stanie pracy silniki pozostają bez zasilania z brak synchronizacji robota jest sygnalizowany systematycznym zapalaniem i gaśnięciem lampki „SYNCHRONIZACJA”. Wciśnięcie klawisza praca powoduje zasilenie silników manipulatora i jest sygnalizowane zapaleniem lampki PRACA i zgaśnięciem lampki GOTOWOŚĆ.



Rys. 4.2. Panel sterowania

Przed przystąpieniem do dalszych czynności należy zsynchronizować manipulator. W tym celu, używając joystick'a panelu programowania, należy sprowadzić manipulator do następującego położenia:

- ramię dolne powinno być pochylone do przodu o ok. 15° od pionu,
- ramię górne powinno być pochylone w dół o ok. 30° od poziomu,
- kolumna powinna być obrócona o ok. 30° w lewo (patrząc od przodu) od położenia środkowego,
- przegub powinien być ustawiony na przedłużeniu ramienia górnego,
- otwór ustalający w końcówce kołnierzej powinien być skierowany do góry.

Po sprowadzeniu manipulatora do położenia wyjściowego należy wcisnąć przycisk SYNCHRONIZACJA. Zostaje wtedy zgaszona lampka UTRATA PROGRAMU i manipulator jest automatycznie sprowadzany do położenia podstawowego. Zakończenie synchronizacji sygnalizowane jest zgaśnięciem lampki SYNCHRONIZACJA i wyświetleniem wstępnego komunikatu na panelu programowania.

Ręczne sterowanie manipulatorem

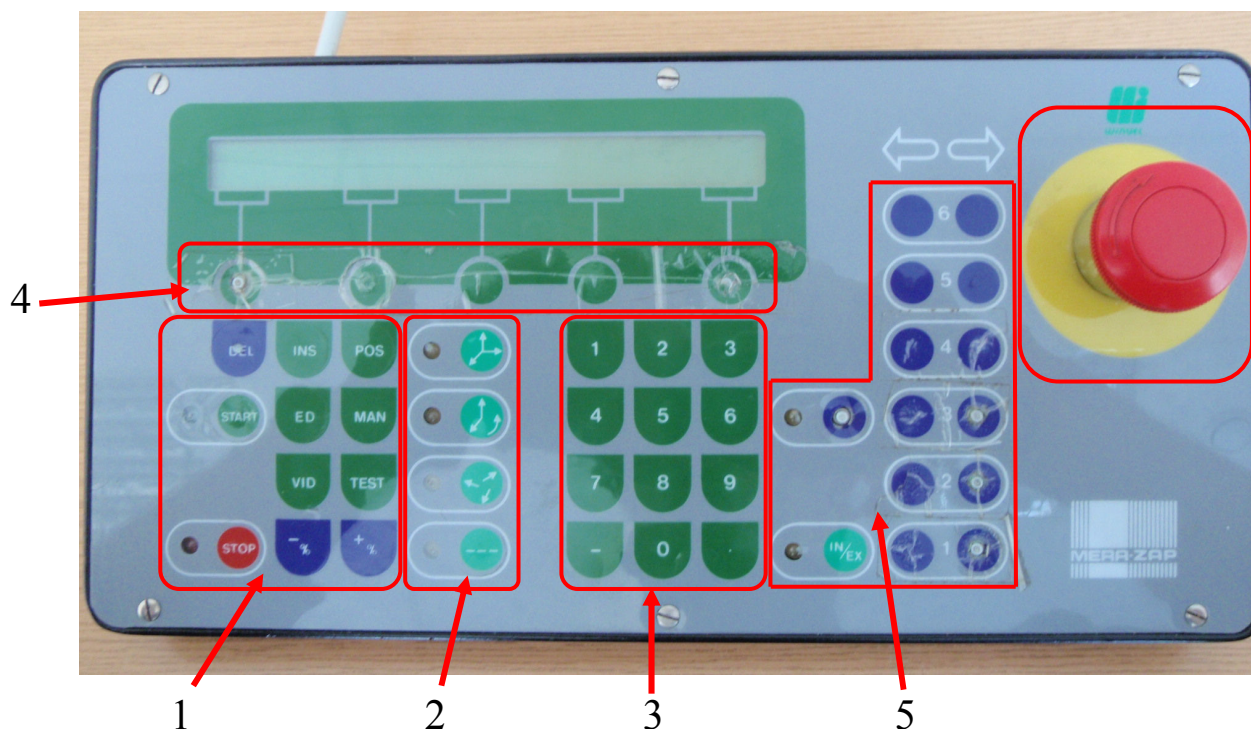
Panel programowania robota umożliwia komunikację między człowiekiem - operatorem a robotem. Za ich pośrednictwem możliwe jest programowanie robota przez "uczenie" oraz sterowanie wykonywaniem wyuczonych czynności.

Otwieranie chwytaka realizowane jest sekwencją poleceń

CHWYTAK 1 CHWYĆ
CHWYTAK 2 ZWOLNIJ

natomiast zamykanie sekwencją

CHWYTAK 2 CHWYĆ
CHWYTAK 1 ZWOLNIJ



Rys. 4.3. Teachbox

Pierwszą grupę przycisków stanowią klawisze wyboru menu. Pozwalają na tworzenie, edycję programu za pomocą zaprogramowanych funkcji.

- POS – zestaw instrukcji pozycjonowania (Rys. 4.5.)
- INS – zestaw dodatkowych instrukcji (Rys. 4.6.)
- ED – zestaw funkcji edycyjnych programu. Pozwala m.in. na wstawianie linii, modyfikowanie jej zawartości, kasowanie fragmentu programu.
- MAN – służy do ręcznego sterowania systemem. Pozwala na ręczne operowanie chwytakiem, zapis i ładowanie stworzonego programu z zewnętrznego nośnika.
- TEST – wykorzystywany jest do testowania klawiatury i wyświetlacza
- VID – niewykorzystywany. Może być użyty do wprowadzania dodatkowych funkcji dla robota.
- START – wywołuje menu umożliwiające uruchomienie stworzonego programu. Ważne, by program po raz pierwszy uruchamiany był w trybie krokowym.
- STOP – zatrzymuje wykonywanie programu użytkownika
- % - (zmniejsz prędkość) - zmniejszenie prędkości ruchu punktu roboczego narzędzia przy pracy automatycznej lub przesunięcie tekstu górnej linii w prawo w trybie programowania i edycji.
- +% - (zwiększ prędkość) - zwiększenie prędkości ruchu punktu roboczego narzędzia przy pracy automatycznej lub przesunięcie tekstu górnej linii w lewo w trybie programowania i edycji.

DEL - pozwala na skasowanie wpisanego znaku numerycznego lub skasowanie komunikatu o błędzie

Przyciski drugiej grupy pozwalają na zmianę trybu poruszania robotem. Patrząc od góry mamy w kolejności:

- praca robota w układzie kartezjańskim,
- praca robota w układzie cylindrycznym
- praca robota w układzie zmiennych złączowych (wewnętrznych)
- sterowanie robotem w wybranym układzie z redukcją prędkości.

Trzecią grupę stanowią przyciski numeryczne. Wykorzystywane są do wprowadzania parametrów ruchu, prędkości, czasu opóźnienia, numeru wejść i wyjść cyfrowych.

Grupa 4

Przyciski funkcyjne. Pozwalają na wybór opcji, bądź parametrów funkcji. Aktywowane są po naciśnięciu przycisków z grupy 1. W przypadku gdy nad jednym z przycisków wyświetlony jest znak --->, wówczas wybranie tej opcji spowoduje przewinięcie ekranu do następnej strony wyboru i wyświetlenie dodatkowych opcji.

Przyciski z grupy piątej wykorzystywane są do ręcznego manipulowania robotem

ZEZW (zezwolenie na ręczne manipulowanie robotem) - uaktywnia przyciski ruchu poszczególnymi osiami, 'naciśnięcie Jakiegokolwiek innego przycisku spoza tej grupy oraz spoza grupy 2 ponownie blokuje przyciski manipulacji.

IN/EXT (osie zewnętrzne) przycisk ten pozwala na przełączenie sterowania pomiędzy wewnętrznymi i zewnętrznymi osiami robota. Na stanowisku laboratoryjnym, zewnętrzną osią robota jest oś stołu obrotowego. Celem zapamiętania pozycji dla osi zewnętrznej należy wybrać tryb pozycjonowania QLIN z czasem min. 5 sec. Dla kąta obrotu $> 90^{\circ}$. Informacje o aktualnie wybranym zestawie osi informuje dioda. Jest ona zapalona gdy wybrane są osie zewnętrzne. Sterowanie stołem obrotowym dokonywane jest za pomocą przycisków umieszczonych obok IN/EXT

OŚ1R - OŚ1L ręczne sterowanie ruchem osi pierwsze) w kierunku dodatnim i ujemnym.

OŚ2R - OŚ2L ręczne sterowanie ruchem osi drugiej w kierunku dodatnim i ujemnym.

OŚ3R - OŚ3L ręczne sterowanie ruchem osi trzeciej w kierunku dodatnim i ujemnym.

OŚ4R - OŚ4L ręczne sterowanie ruchem osi czwartej w kierunku dodatnim i ujemnym.

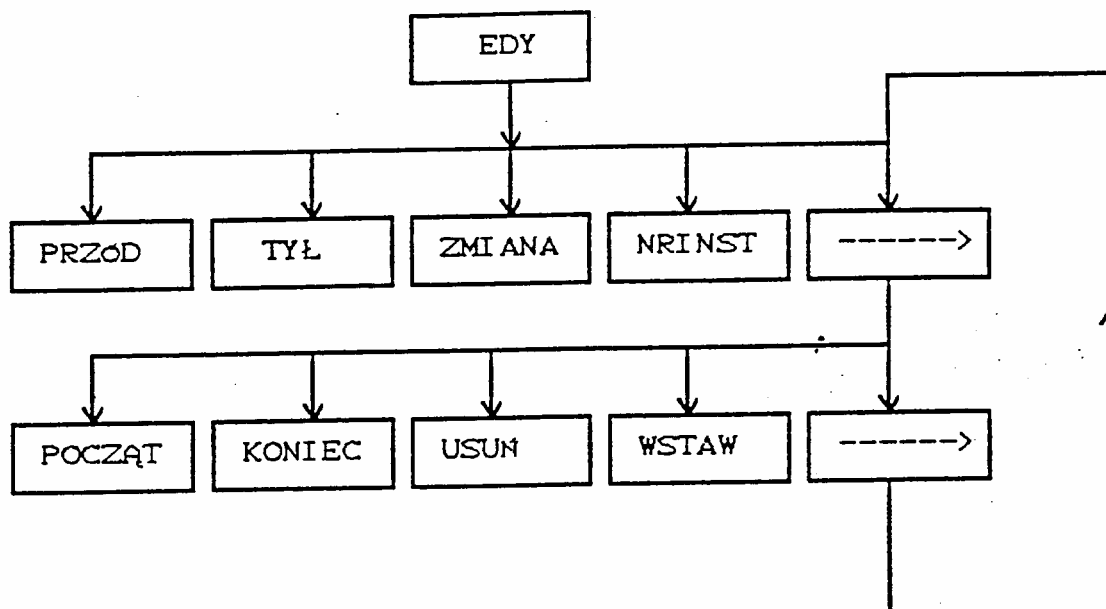
OŚ5R - OŚ5L ręczne sterowanie ruchem osi piątej w kierunku dodatnim i ujemnym.

OŚ6R - OŚ6L ręczne sterowanie ruchem osi szóstej w kierunku dodatnim i ujemnym.

Pełny opis obsługi panelu sterowania studenci znajdą w podręczniku programowania robota Irp-6 dostępnym w trakcie ćwiczeń.

Edycja programu

Dla sprawnego przebiegu ćwiczenia należy zapoznać się z podstawowymi możliwościami edytora programu: wyświetlenie instrukcji następnej / poprzedniej / pierwszej / ostatniej / określonej numerem, usunięcie instrukcji, wstawienie dodatkowej instrukcji oraz zmiana parametrów instrukcji (rys. 4.4.).



Rys. 4.4. Funkcje edycyjne

Uruchamianie programu

Istnieją trzy sposoby uruchomienia manipulatora w trybie pracy automatycznej:

- STPOCZ pozwala uruchomić program od jego pierwszej instrukcji,
- START pozwala uruchomić program od instrukcji pokazanej na wyświetlaczu,
- KROK pozwala wykonać jedną, aktualnie wyświetloną instrukcję.

W czasie uruchamiania programu korzystnie jest używać opcji KROK do testowania i korygowania każdej instrukcji osobno a dopiero po sprawdzeniu całego programu zaleca się jego uruchomienie od pierwszej instrukcji. Należy uruchomić krokowo, od pierwszej i od aktualnie wyświetlanej instrukcji program dostarczony przez prowadzącego ćwiczenie.

Programowanie podstawowych operacji robota

Każdy program działania robota powinien rozpoczynać się instrukcjami wyboru narzędzia i zmiany prędkości podstawowej. W początkowej fazie ćwiczenia mogą one mieć następującą postać:

10 NARZĘDZIE 1

20 PRĘDKOŚĆ = 100 MM/S, PRĘDKOŚĆ MAX = 200 MM/S

Narzędzie nr 1 jest w robocie zdefiniowane wstępnie i dlatego może być używane bez wcześniejszej definicji jawnej.

Należy pamiętać, że w przygotowywanym programie często występują błędy uniemożliwiające jego realizację. Najczęściej polegają one na przekroczeniu maksymalnej prędkości w osiach, nie włączeniu napędów lub niedozwolonej zmianie konfiguracji. Wystąpienie błędu jest zawsze sygnalizowane na panelu operacyjnym i na panelu programowania a lista sygnalizowanych błędów jest zamieszczona w instrukcji programowania robota na stronie 74. W razie wystąpienia błędu powinien on zostać "skasowany" przyciskiem KAS. Następnie należy dokonać odpowiedniej korekty programu i ponownie go uruchomić.

Programowanie podstawowych czynności manipulatora.

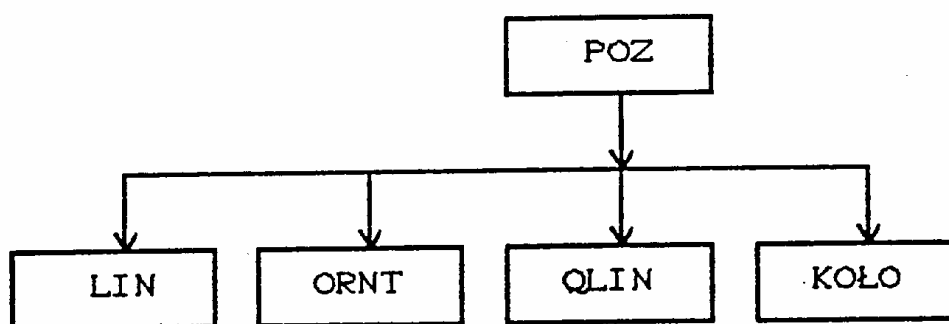
Instrukcja pozycjonowania.

Szczególną uwagę w trakcie wykonywania ćwiczenia należy poświęcić instrukcji pozycjonowania (rys. 4.5.). W przestrzeni roboczej **P** manipulatora należy wybrać dwa różne punkty **A** i **B** (odległe od siebie o ok. 50-100cm) takie, że odcinek **AB** jest całkowicie zawarty w przestrzeni **P**. Należy zaprogramować a następnie wykonać w trybie pracy automatycznej przemieszczenie manipulatora z punktu **A** do **B** i z powrotem. Przemieszczenie takie programuje się umieszczając w programie instrukcje pozycjonowania w odpowiednich punktach przestrzeni roboczej. Należy zastosować pozycjonowanie dokładne, bezwzględne z zadaną prędkością a ruch manipulatora powinien być wykonywany kolejno przy pozycjonowaniu:

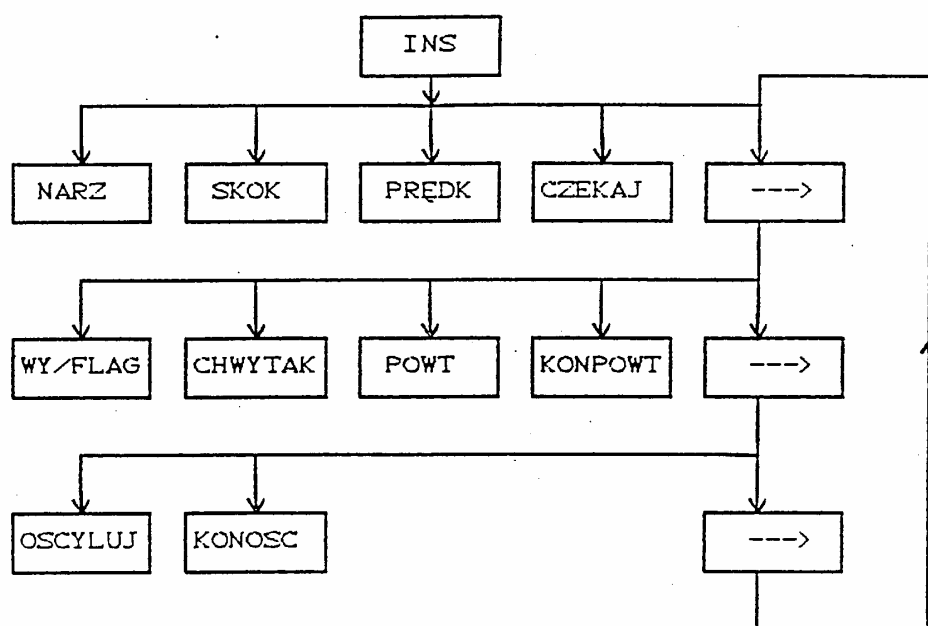
- a) quasiliniowym,
- b) liniowym,
- c) kołowym (punkt pośredni należy wybrać samodzielnie).

Należy zaobserwować na czym polegają różnice pomiędzy poszczególnymi rodzajami pozycjonowania (a, b, c), a następnie wykonać te same ruchy przy pozycjonowaniu zgrubnym oraz przy pozycjonowaniu z zadanym czasem. Po zapoznaniu się z poszczególnymi rodzajami ruchów manipulatora należy ułożyć program demonstrujący kolejno wszystkie jego proste ruchy.

W praktyce często wykorzystuje się również pozycjonowanie względne manipulatora. W czasie wykonywania ćwiczenia należy zapoznać się z tym rodzajem pozycjonowania i ułożyć krótki program z jego wykorzystaniem.



Rys. 4.5. Instrukcje pozycjonowania



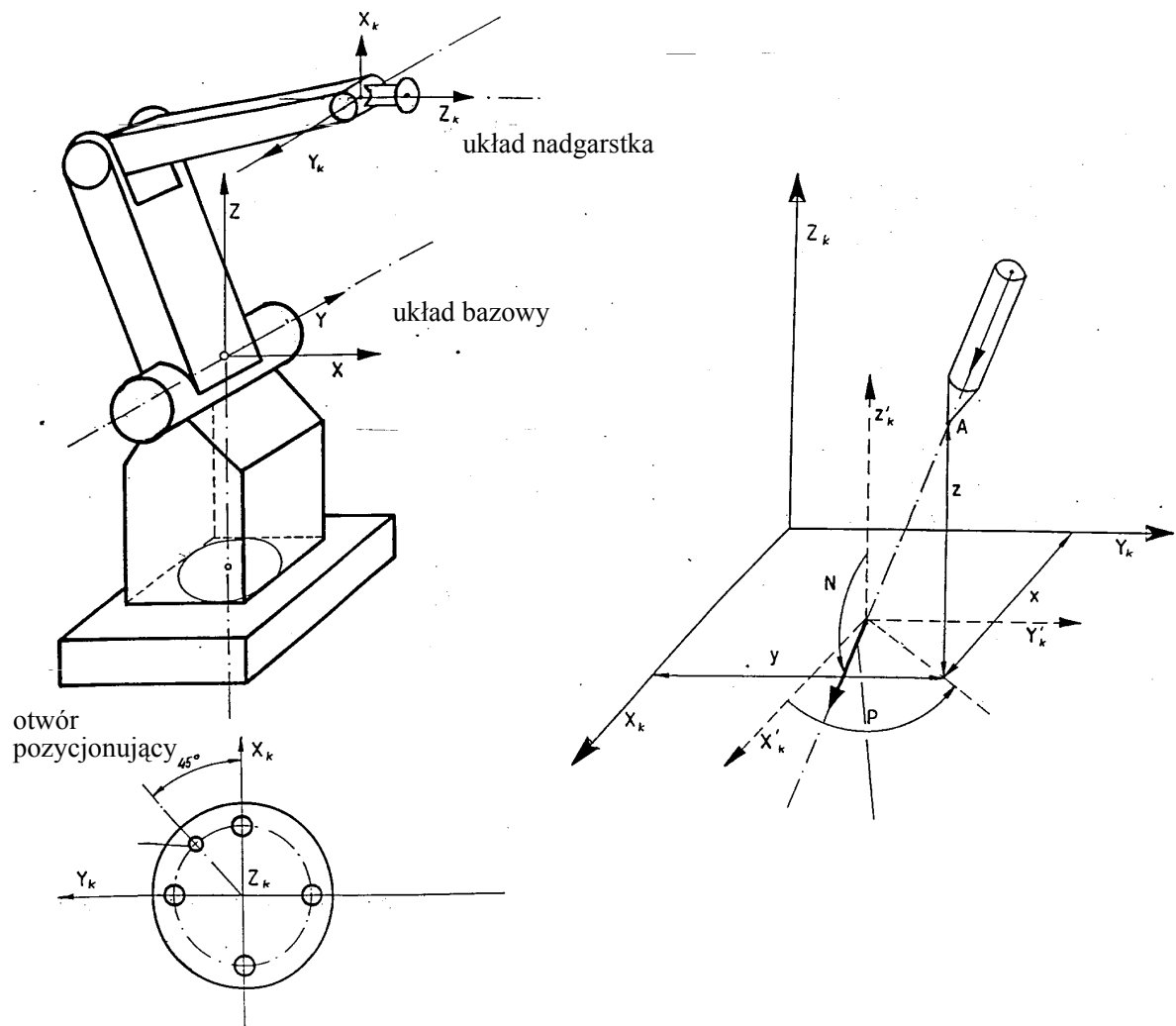
Rys. 4.6. Instrukcje pomocnicze

Inne instrukcje robota.

Oprócz instrukcji pozycjonowania robot IRp-6 pozwala używać 10 instrukcji pomocniczych (rys. 4.6.). Należy zapoznać się z tymi instrukcjami a następnie ułożyć program chwytaka i przenoszenia niewielkiego przedmiotu (na przykład pudełka zapalek) wzdłuż obwodu prostokąta **PQXY**. Prostokąt ten powinien leżeć w jednej z płaszczyzn kartezjańskiego układu współrzędnych związanego z podstawą manipulatora. W każdym z wierzchołków prostokąta manipulator powinien pozostawać nieruchomy przez czas 2 sekund a prędkości poruszania się manipulatora wzdłuż każdego z boków prostokąta powinny być różne. Wprowadzając instrukcję powtarzania należy otrzymany program zmodyfikować tak aby końcówka manipulatora kreśliła najpierw kilkakrotnie trójkąt **PQX** a następnie prostokąt **PQXY**, a dodatkowo otrzymany program należy uzupełnić instrukcjami początku oraz końca oscylacji.

W końcowej fazie ćwiczenia należy zmienić definicję narzędzia i zaobserwować wpływ tej zmiany na działanie manipulatora. Szczególną uwagę należy zwrócić na wpływ definicji narzędzia, a zwłaszcza jego orientacji, na możliwość wykonania ruchu kołowego.

Zdefiniowanie narzędzia polega na określeniu położenia punktu roboczego narzędzia i orientacji jego osi względem układu nadgarstka (rys. 20). Położenie określone jest przez trzy współrzędne kartezjańskie odczytane w układzie nadgarstka (indeks k na rys. 20). Orientację narzędzia określają dwa kąty: nutacji (N) i precesji (P) odczytane również względem osi układu nadgarstka.



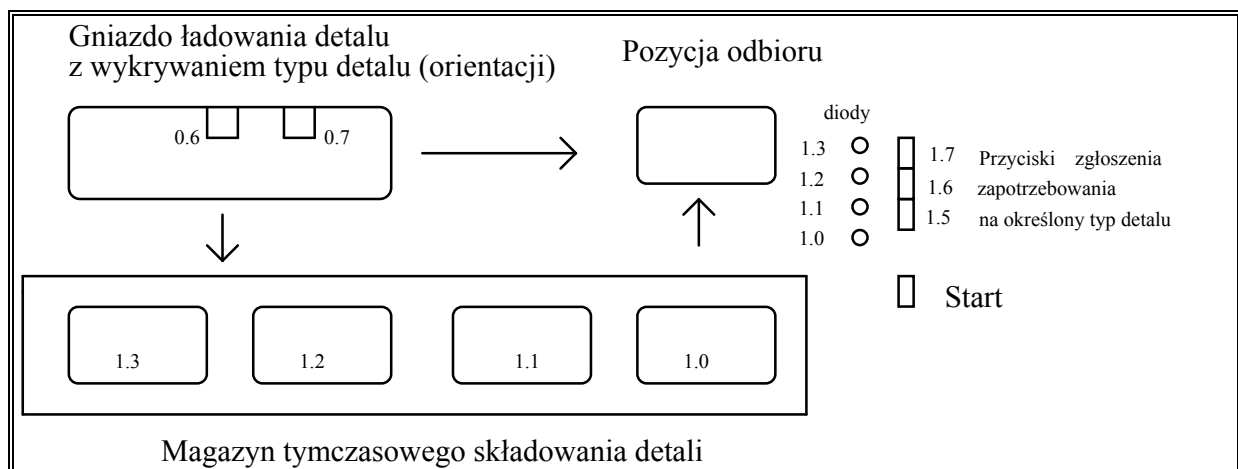
Rys. 4.7. Definiowanie narzędzia

Ćwiczenie F

Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z budową, działaniem i programowaniem robota dydaktycznego sterowanego za pomocą przemysłowego sterownika programowalnego. Zadanie do wykonania polega na zaprojektowaniu działania manipulatora w środowisku w którym dużą rolę odgrywają zależności logiczno-relacyjne.

Zadania do wykonania



Zadaniem do wykonania jest napisanie programu, który uwzględniając powyższy rysunek powinien spełniać następujące warunki:

1. Detal jest ładowany do gniazda ręcznie w dozwolonych 3 orientacjach. Tam przebywa co najmniej przez zadany czas (np. 5s), kiedy to w warunkach przemysłowych poddawany byłby obróbce (np.: frezowanie ścian). W tym czasie czerwona dioda powinna pulsować. Po upływie zadanego czasu dioda pali się w sposób ciągły, aż do odbioru detalu. Wygaszona dioda winna sygnalizować puste gniazdo.
2. Użytkownik ma prawo zgłosić zapotrzebowanie na określony typ detalu za pomocą jednego z 3 przycisków na ścianie bocznej rozgałęźnika, co powinno być sygnalizowane jedną z zielonych diod
3. Detal z gniazda powinien zostać przeniesiony przez manipulator według następujących kryteriów:
 - jeżeli jest zgłoszone zapotrzebowanie na ten typ detalu jaki jest w gnieździe manipulator przenosi go bezpośrednio na pozycję końcową;
 - jeżeli w danym momencie nie ma zapotrzebowania składowuje w magazynie tymczasowym na pozycję przyporządkowaną określonej orientacji (segregacja detali);
 - jeżeli nie ma zapotrzebowania i magazyn tymczasowy jest wypełniony (pozycja dla detalu o określonej orientacji jest zajęta) detal pozostaje w gnieździe. Zatem wyższy priorytet ma tu zgłoszenie odbioru detalu w pozycji końcowej od składowania w magazynie.

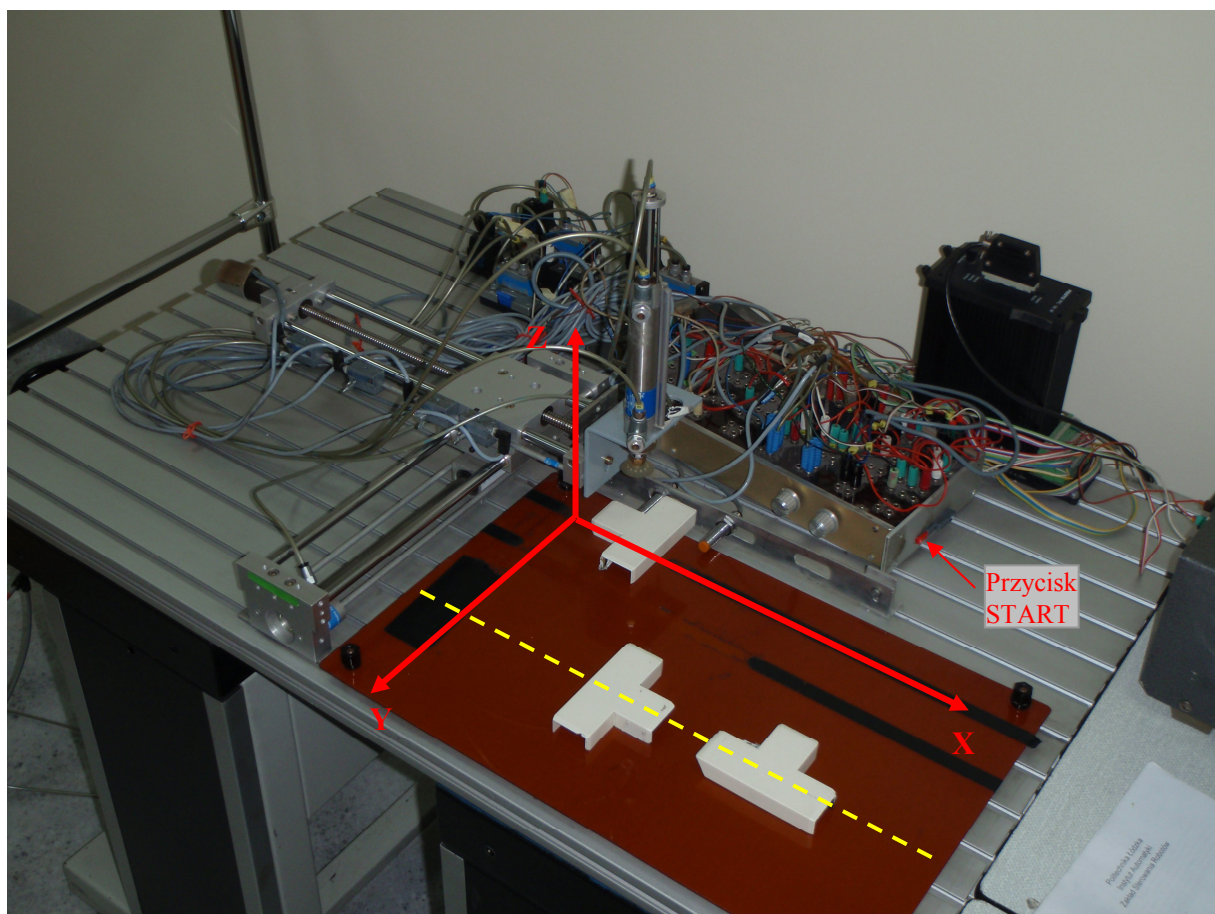
4. W przypadku gdy w gnieździe i w magazynie jest detal określonej orientacji i zostało zgłoszone zapotrzebowanie na detal innego typu, który aktualnie nie jest dostępny (nie ma go w magazynie) powinien zostać włączony alarm (np.: 3 diody zielone rytmicznie pulsują), gdyż taka sytuacja grozi zablokowaniem systemu i wymaga ingerencji operatora.

Przebieg ćwiczenia:

1. Zapoznać się ze strukturą i możliwościami języka programowania.
2. Załączyć komputer, zasilacz i przyłącze pneumatyczne.
3. Uruchomić program fst101.exe.
4. Otworzyć własny projekt w systemie FST101.
5. Przygotować program sterowania robota dla zadania opisanego w części „Zadania do wykonania”.

Opis stanowiska

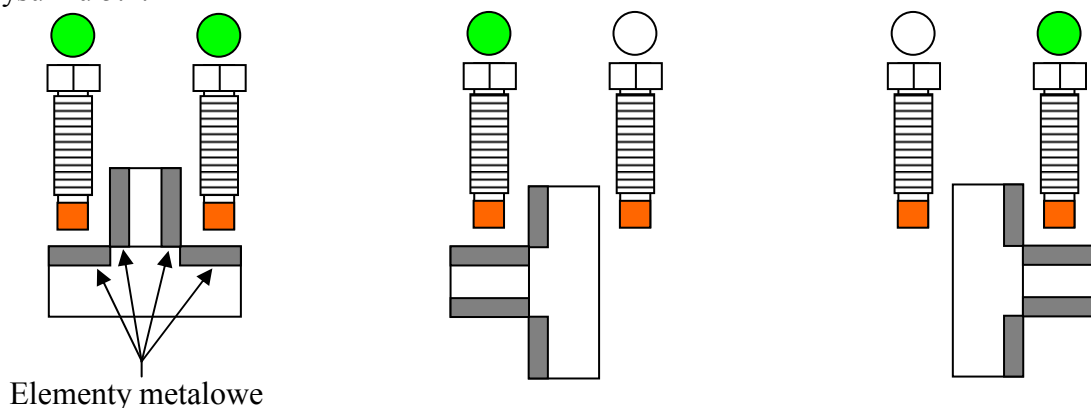
Manipulator o kartezjańskim łańcuchu kinematycznym posiada 3 stopnie swobody, przy czym jedna oś (X) napędzana jest silnikiem prądu stałego zaś dwie pozostałe (Y, Z) siłownikami pneumatycznymi. Robot dodatkowo wyposażony jest w przyssawkę podciśnieniową. Wszystkie elementy stanowiska są produkcji firmy Festo.



Rys 5.1. Stanowisko z manipulatorem firmy FESTO

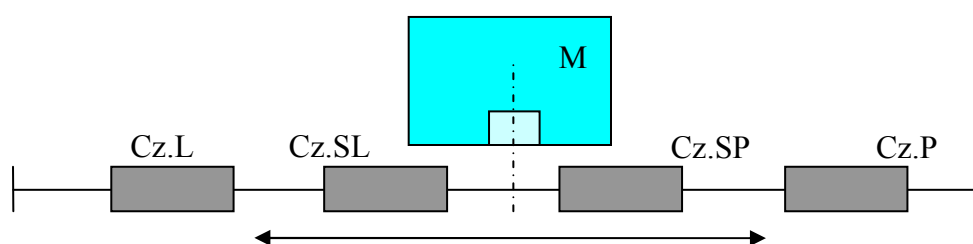
Wszystkie elementy sterujące pracą aktuatorów podłączone są do sterownika PLC firmy FESTOP. Posiada on wejścia i wyjścia cyfrowe, wejścia i wyjścia analogowe, oraz interfejs RS232 do komunikacji z komputerem. Do wejść cyfrowych przyłączone są czujniki informujące o orientacji podawanego elementu oraz o położeniu efektora manipulatora.

Każdy detal o kształcie litery „T” posiada elementy stalowe. Dzięki nim istnieje możliwość określenia jego orientacji za pomocą dwóch czujników indukcyjnych. Pokazane jest to na rysunku 5.2.



Rys 5.2. Konfiguracje elementu: a) działają dwa czujniki indukcyjne, b) działa lewy czujnik indukcyjny, c) działa prawy czujnik indukcyjny.

Dodatkowo na osi X rozmieszczone są 4 czujniki zbliżeniowe. Wyznaczają one charakterystyczne współrzędne, pomagające w odpowiednim rozmieszczeniu elementów na dolnej linii magazynowania, oznaczonej kolorem żółtym na Rys 5.1.. Zadziałanie czujnika zbliżeniowego na osi X sygnalizowane jest załączeniem diody na czujniku oraz wystawieniem stanu wysokiego na wejściu odpowiadającym danemu sensorowi. Element magnetyczny oddziałujący na czujniki jest nieruchomy a przesuwa się listwa z czujnikami oznaczonymi odpowiednio od strony lewej jako lewy, środkowo-lewy, środkowo-prawy, prawy (tabela 7). Ruch robota powinien tak być programowany, aby znajdował się w strefie między czujnikami skrajnymi.

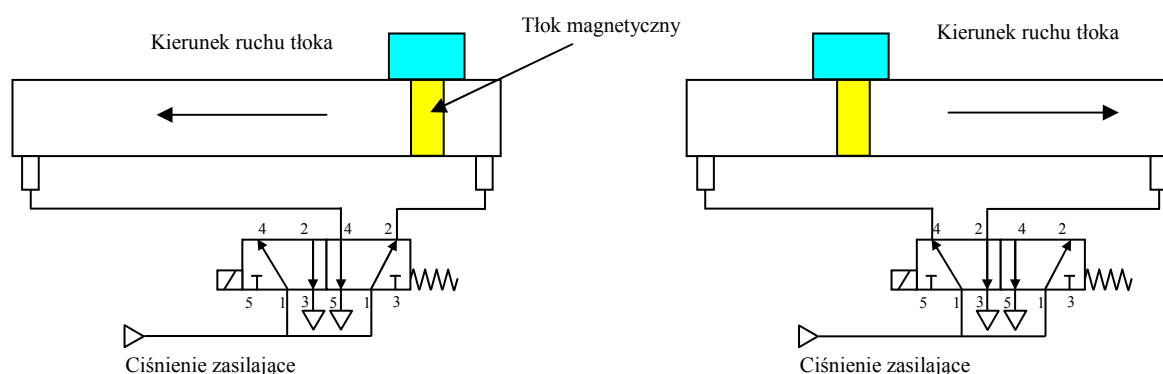


Rys. 5.3 Lista z czujnikami i element magnetyczny.

Zatrzymaniom napędu osi X w kolejnych pozycjach odpowiada rozmieszczenie elementów w obszarze umownie nazywanym magazynem (z reguły wykorzystuje się 3 wybrane takie pozycje).

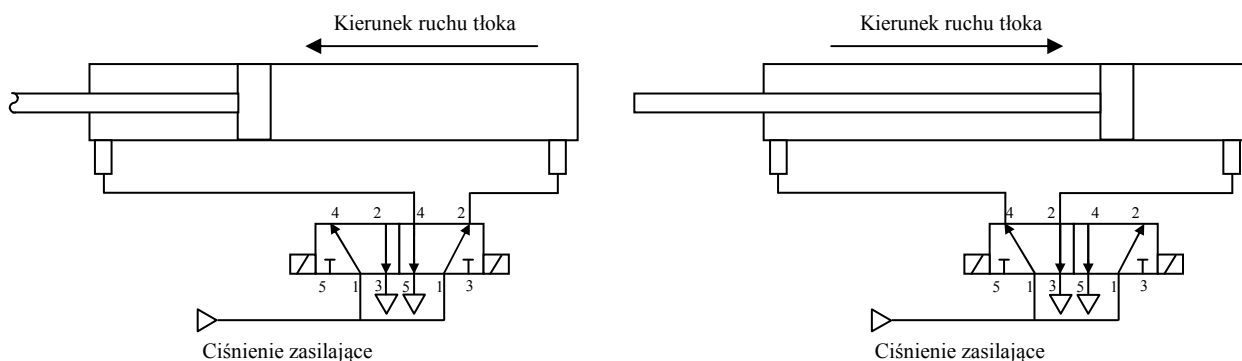
Osie Y, Z napędzane są siłownikami pneumatycznymi. Cechą charakterystyczną tego rodzaju napędów jest przyjmowanie dwóch krańcowych stanów ustalonych. Ze względu na zasadę działania siłowniki pneumatyczne można podzielić na dwie grupy: jednostronnego oraz dwustronnego działania.

Napęd **osi Y** manipulatora FESTO zbudowany jest w oparciu o siłownik bez tłoczyskowy dwustronnego działania ze sprzężeniem magnetycznym tłoka. Tego rodzaju aktuator posiada dwa przyłącza służące do doprowadzania i odprowadzania sprężonego powietrza. Otwarcie zaworu elektromagnetycznego i podanie sprężonego powietrza do komory siłownika powoduje przesunięcie tłoka do drugiego skrajnego położenia przy jednoczesnym opróżnieniu drugiej komory. Operacja ta zajmuje pewien czas, dlatego po każdej zmianie stanu zaworu sterującego dopływem powietrza wskazane jest odczekanie, aby akcja mogła być zakończona. Do sterowania wykorzystano zawór typu MEH 5/2 (prod. FESTO) z *pojedynczą* cewką sterującą i *sprężyną zwrotną* (wystarczający jest zatem *pojedynczy* sygnał sterujący) a przepływ powietrza w układzie oraz kierunek ruchu mechanicznego przedstawiono na rysunku poniżej. Cechą używanego zaworu jest, że przy braku sygnału sterującego jest w określonym stanie, co wymusza spoczynkowy stan napędu osi.



Rys 5.4. Przykład sterowania siłownikiem dwustronnego działania ze sprzężeniem magnetycznym.

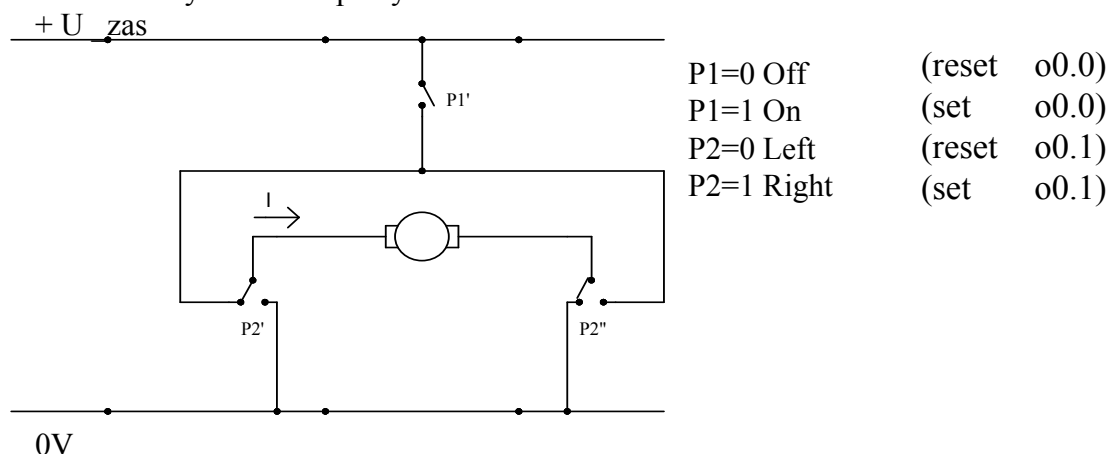
Do podnoszenia i opuszczania przyssawki podciśnieniowej w manipulatorze dydaktycznym FESTO (**oś Z**) wykorzystano siłownik dwustronnego działania z wyprowadzonym tłokiem. Ruch tłoka wynika, podobnie jak poprzednio, z różnicy ciśnień, jaka panuje w komorach siłownika. Zastosowano dwustanowe sterowanie takim siłownikiem z wykorzystaniem zaworu typu JMEH 5/2 sterowanego z *dwóch* cewek zasilających, co wymaga odpowiedniego podania *dwóch* sygnałów sterujących. Cechą używanego zaworu jest to, że przy braku sygnałów sterujących zawór nie zmienia stanu pozostawiając napęd w pozycji, w jakiej się znalazł po ostatnimysterowaniu. Przepływ powietrza pokazany jest na rysunku 5.5.



Rys 5.5. Przykład sterowania siłownikiem dwustronnego działania.

Schemat sterowania silnikiem prądu stałego, wykorzystywanego do napędzania osi X przedstawiony jest na rysunku 5.5. Wykorzystane są tutaj dwa przekaźniki elektromagnetyczne. Przekaźnik P1 służy do załączenia bądź odłączenia napięcia zasilania od obwodu silnika. Przekaźnik P2 ma za zadanie zmienić polaryzację napięcia na silniku. Silnik połączony jest sprzęgłem z przekładnią śrubowo-toczną zamieniającą ruch obrotowy na przesunięcie linowe. Dzięki temu możliwe są do uzyskania 3 stany pracy układu. Ruch w osi X w kierunku lewym, w kierunku prawym oraz zatrzymanie ruchu.

Przekaźniki P1 oraz P2 podłączone są do wyjść cyfrowych sterownika (o0.0, o0.1). Odpowiednia kombinacja stanów na tych wyjściach zgodnie z informacją na rysunku pozwala na uzyskanie możliwych stanów pracy.



P1, P2 - przekaźniki,
P1 - załączenie układu,
P2 - kierunek przesuwu silnika.

Rys. 5.6. Obwód sterowania silnika DC

Struktura języka Festo-INTERPRETER

Język Festo-INTERPRETER jest problemowym językiem programowania zadań sterowania procesami przemysłowymi o dowolnym stopniu złożoności. Podstawowym elementem strukturalnym języka jest zdanie logiczne postaci:

IF <warunek> THEN <rozказы części wykonawczej 1>
OTHRW <rozказы części wykonawczej 2>

W części warunkowej następuje sprawdzenie zgodności rzeczywistego stanu zmiennych ze stanem zaprogramowanym. Wynik sprawdzenia decyduje o realizacji określonej części wykonawczej.

Przykład:

IF	<i>i1.0</i>	<i>'jeśli bit wejściowy i1.0=1</i>
THEN	<i>SET o1.0</i>	<i>'to ustaw bit wyjściowy o1.0=1</i>
OTHRW	<i>RESET o1.0</i>	<i>'w przeciwnym razie wyzeruj bit wyjściowy o1.0=0</i>

Część warunkowa może zawierać złożoną funkcję logiczną wielu zmiennych, zaś część wykonawcza może składać się z wielu rozkazów. Maksymalna liczba instrukcji w jednym zdaniu logicznym wynosi 100.

Składnia Festo-INTERPRETER dopuszcza uproszczone struktury zdania logicznego:

- bez części warunkowej:
 THEN <rozказы części wykonawczej 1>
- bez części wykonawczej drugiej:
 IF <warunek> THEN <rozказы części wykonawczej 1>
- z częścią wykonawczą 1 lub 2 zawierającą instrukcje pustą NOP:
 IF <warunek> THEN NOP
 ELSE NOP

Kolejnym elementem struktury oprogramowania jest KROK będący zespołem zdań logicznych. Sekwencja kroków tworzy PROGRAM. W obrębie programu kroki wykonywane są kolejno zgodnie z ich numeracją. Maksymalna liczba kroków wynosi 255. Przejście do kolejnego kroku uwarunkowane jest przez strukturę zdań logicznych np.:

```
STEP 1
  THEN <rozказы części wykonawczej. 1>
STEP 2
  IF <warunek.>
  THEN <rozказы części wykonawczej. 1>
STEP 3
  IF <warunek.>
  THEN <rozказы części wykonawczej. 1>
  OTHRW <rozказы części wykonawczej. 2>
STEP 4
  IF <warunek.>
```

THEN < rozkazy części wykonawczej. 1>
OTHRW < rozkazy części wykonawczej. 2> 'zdanie 4.1
IF < warunek.>
THEN < rozkazy części wykonawczej. 1>
OTHRW < rozkazy części wykonawczej. 2> 'zdanie 4.2
IF < warunek.>
THEN < rozkazy części wykonawczej. 1> 'zdanie 4.3
STEP 5
THEN < rozkazy części wykonawczej. 1>

Przejście 1-2 zostanie zrealizowane bezwarunkowo z jednoczesnym wykonaniem rozkazów części wykonawczej. Przejście 2-3 nastąpi wyłącznie po spełnieniu warunku (i wykonaniu rozkazów). Przejście 3-4 nastąpi zawsze zaś wykonane rozkazy będą zależeć od spełnia warunku zdania. Zdania logiczne w kroku 4 tworzą zestaw niezależnie realizowanych zdań. Ich wielokrotna realizacja trwa do momentu spełnienia warunku w ostatnim (4.3) zdaniu. Nastąpi wtedy sekwencyjne przejście do 5 kroku. Gdyby zdanie 4.3 miało strukturę zdania z kroku 1 lub 3 wówczas przejście do kroku 5 nastąpiłoby natychmiast. Przejście do dowolnego kroku umożliwia instrukcja skoku.

Zmienne systemowe

Tabela 5

symbol	ilość bitów	opis	możliwe działania
i<w>.	1	one bit input <w> input word [0..2] bit [0..7]	AND, OR, EXOR, N
iw<n>	8	input word <n> word number [0..2]	LOAD, mathematical relations
o<w>.	1	one bit output <w> output word [0..1] bit [0..7]	AND, OR, EXOR, N, SET, RESET
ow<n>	8	output word <n> word number [0..1]	LOAD, mathematical relations
f<w>.	1	one bit of memory <w> memory word [0..15] bit [0..7]	AND, OR, EXOR, N, SET, RESET
fw<n>	16	memory word <n> word number [0..15]	LOAD, mathematical relations
t<n>	1	timer flag <n> number of timer [0..31]	AND, OR, EXOR, N, SET, RESET
tp<n>	16	timer set value <n> number of timer [0..31]	LOAD
tw<n>	16	timer actual value <n> number of timer [0..31]	LOAD, mathematical relations
c<n>	1	counter flag <n> number of counter [0..15]	AND, OR, EXOR, N, SET, RESET

cp<n>	16	counter set value <n> number of timer [0..15]	LOAD
cw<n>	16	counter actual value <n> number of timer [0..15]	LOAD, mathematical relations
r<n>	16	register <n> number of register [0..63]	LOAD, mathematical relations
v	--	direct argument	LOAD

System umożliwia użycie własnych nazw zmiennych systemowych. Wymaga to zapisu powiązań w tablicy alokacji.

Instrukcje

Instrukcje Festo-INTERPRETER można podzielić na następujące grupy:

1. Instrukcje tworzące strukturę programu.
2. Instrukcje logiczne tworzące część warunkową zdania logicznego, których argumentami mogą być zmienne binarne jak i wielobitowe.
3. Instrukcje porównania zmiennych wielobitowych tworzące również część warunkową.
4. Rozkazy zmiany stanu zmiennych binarnych lub wartości zmiennych wielobitowych, występujące w części wykonawczej zdania logicznego. Zmiana stanu bitu następuje przy użyciu jednej instrukcji, zmiana argumentu wielobitowego wymaga dwóch instrukcji. W skład złożonych rozkazów zmiany wartości mogą wchodzić operacje matematyczne lub instrukcje konwersji.
5. Rozkazy obsługi procedury - maksymalnie może być 8 procedur numerowanych 0-7
6. Rozkaz skoku.

Szczegółowy opis instrukcji zawiera tabela 6.

Tabela 6

mnemonik	zmienna	część zdania	działanie
STEP IF, THEN, OTHRW	number		fundamental instructions of AWL language syntax
AND OR N EXOR ()	i, o, f, t, c, r	condition	logical and logical or logical not logical exclusive or
>, <, >=, <=, =, ◇	r, cw, cp, tw, tp, iw, ow, fw, v	condition	mathematical relations
SET RESET	o, f, t, c	execution	set bit reset bit
LOAD TO	r, v, cp, cw, tp, tw, fw, iw, ow	execution	load multibit value to destination variable
+, -, *, /, (,)	r, cw, cp, v, tw, tp, fw, iw, ow	execution	mathematical operations

DEC INC ROR ROL SHR SHL BID DEB SWAP CPL INV	iw, ow, fw, tw, tp, cw, cp, r	execution	decrement increment rotation right rotation left shift right shift left transcoding binary to BCD trancoding BCD to binary byte changing $x = \text{not } x + 1$ inversion
CMP WITH	no of procedure parameters	execution	jump to user procedure [0..7] with parameters [p0..p15]
NOP		condition/exec.	no operating
JMP TO	step no	execution	jump to program step

Operacje timer-licznik

W systemie FPC 101 istnieje możliwość odmierzania jednocześnie i niezależnie 32 czasów. Timery działają z podstawą czasu 0.01 sek. i umożliwiają odmierzanie czasu do 655.32 sek. każdy. W systemie wyróżniono 16 rejestrów do wykorzystania jako liczniki liczące w przód lub w tył. Zarówno timery jak i liczniki zawierają rejestr wartości zadanej do inicjacji układu, rejestr wartości bieżącej oraz flagę informującą o stanie układu.

Przykład:

Odmierzenie 1 sek opóźnienia:

STEP 10

THEN LOAD V100 'load the value 100
 TO TP12 'to set value register of 12th timer
 SET T12 'start timer 12

STEP 20

IF N T12 'if T12=0 timer counted down
THEN NOP

Powtórzenie pętli programu 3 razy (zliczanie wstecz):

THEN LOAD V3 'load the value 3
 TO CW3 'to actual value register of 3rd counter
..... 'body of loop
THEN DEC CW3 'decrement value
IF N C3 'if counter finished
THEN NOP

Powtórzenie pętli 3 razy (zliczanie w przód):

THEN	LOAD V3	<i>'load the value 3</i>
	TO CP3	<i>'to set value register of 3rd counter</i>
	LOAD V0	<i>'load the value 0</i>
	TO CW3	<i>'to actual value register</i>
.....		<i>'body of loop</i>
THEN	INC CW3	<i>'increment value</i>
IF	N C3	<i>'if counter finished</i>
THEN	NOP	

Procedury użytkownika

System umożliwia zdefiniowanie max. 8 procedur użytkownika. Uaktywnienie procedury powoduje zatrzymanie wykonania programu, wykonanie instrukcji zapisanych w module i wznowienie programu z miejsca zatrzymania. Nie ma możliwości wywołania procedury z innej procedury ani z tej samej. Procedury mogą być parametryzowane (max. 16 argumentów) lub nie. W celu zdefiniowania procedury należy wywołać edytor z następującymi ustawieniami (program/module - B, program number - 0..7). Każdy moduł ma niezależną numerację kroków a w przypadku procedur parametryzowanych zmienne zapisywane są w komórkach: FU32 - FU47. Na przykład procedura numer 0 odmierzenia zadanego odcinka czasu ma postać:

STEP 10

THEN	LOAD FU32	<i>'load first parameter value</i>
	TO TP0	<i>'to set value register of 0 timer</i>
	SET T0	<i>'start timer 0</i>

STEP 20

IF	N T0	<i>'if T0=0 timer counted down</i>
THEN	NOP	

Wywołanie tej procedury dla czasu 1 sekunda ma postać:

THEN	CMP 0	<i>'call user defined module 0</i>
	WITH V100	<i>'with first argument 100</i>

System manipulatora Festo

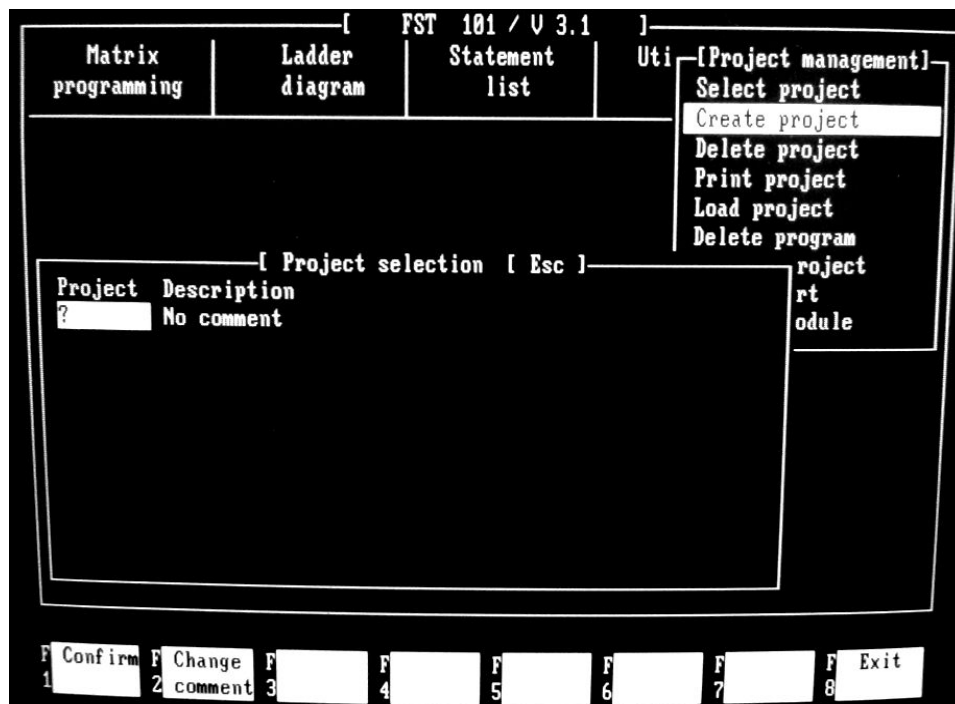
Stanowisko laboratoryjne złożone jest z komputera klasy PC jako konsoli operatorskiej, programowalnego sterownika przemysłowego, części mechanicznej manipulatora, zespołu zaworów i zasilaczy. Opis połączeń układu manipulatora ze sterownikiem przedstawia tabela 7.

Tabela 7

<i>Element układu</i>	<i>Adres sterownika</i>	<i>Uwagi</i>
silnik DC- przekaźnik P1 (oś X)	o0.0	załączenie/wyłączenie napięcia
silnik DC - przekaźnikP2 (oś X)	o0.1	wybór polaryzacji napięcia (kierunku ruchu w osi X)
zawór 1 siłownika pionowego (oś Z) - komora górna	o0.2	rozkaz set wypełnia komorę
zawór 2 siłownika pionowego (oś Z)- komora dolna	o0.3	rozkaz reset opróżnia komorę
zawór siłownika poziomego (oś Y)	o0.4	reset/set - dwie pozycje skrajne
zawór przyssawki	o0.5	rozkaz set uaktywnia przyssawkę
diody sygnalizacyjne (cz. - ziel.)	o1.0 - o1.4	rozkaz set załącza
czujnik prawy na osi X	i1.0	do wykorzystania z instrukcją typu IF dla wygenerowania akcji typu STOP dla silnika DC
czujnik środkowy-prawy na osi X	i1.1	
czujnik środkowy-lewy na osi X	i1.2	
czujnik lewy na osi X	i1.3	
czujnik osi Y	i1.4	
Czujniki indukcyjne (lewy - prawy)	i0.6 - i0.7	detekcja elementu
Przyciski (dolny, środkowy, górny)	i1.5 - i1.7	zgłoszenie operatora

Rozpoczęcie pisania programu dla manipulatora FESTO.

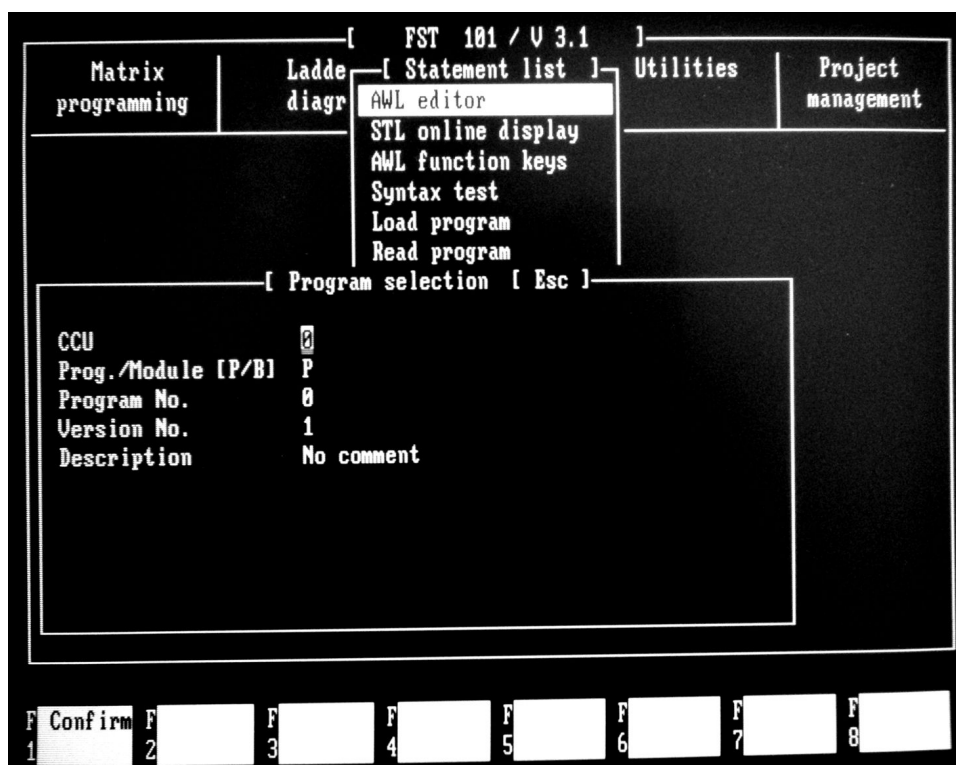
Przy pierwszym uruchomieniu programu należy wybrać z górnego menu „**Project management**”. Zawarte w nim opcje pozwalają m.in. na otwarcie istniejącego projektu, tworzenie nowego projektu, kasowanie lub drukowanie aktualnie otwartego projektu. Następnie po wybraniu „**Create project**” ukazuje się okno jak na rys 5.6. Jako nazwę wpisuje się identyfikator grupy, a komentarz mogą stanowić nazwiska członków grupy.



Rys. 5.6. Tworzenie nowego projektu.

W kolejnym kroku z menu „**Statement list**” należy wybrać polecenie „**AWL editor**”. Powoduje to otwarcie okna opcji edytora Rys 5.7. Festo-INTERPRETER pozwala na napisanie programu [P] lub procedury (modułu) [B]. Po wybraniu opcji procedury [B], jej numer ustawia się w polu „**Program No.**”. W przypadku wybrania opcji Programu [P], jego kolejną wersję można stworzyć ustawiając pole „**Version No.**”.

Wszystkie ustawienia zatwierdza się poprzez naciśnięcie przycisku F1.



Rys. 5.7. Opcje edytora.

Podczas pisania programu istnieje możliwość używania komend edycyjnych dotyczących tworzonego tekstu. Operacje kopiowania, przesuwania, kasowania tekstu należy rozpocząć od zaznaczenia fragmentu programu poddawanego edycji. Początek bloku programu zaznacza się kombinacją klawiszy Ctrl+K+B, koniec bloku tekstu Ctrl+K+K. Zaznaczony obszar można kopiować naciskając klawisze Ctrl+K+C, przesuwać Ctrl+K+V, odznaczyć Ctrl+K+H. Kasowanie dokonywane jest za pomocą kombinacji Ctrl+Y.

Po stworzeniu procedury bądź programu głównego istnieje możliwość sprawdzenia poprawności jego składni. W tym celu wybiera się **Statement list--> Syntax test**. Lista ewentualnych błędów wraz z ich typem i miejscem w programie, dostępna jest po wybraniu opcji **Statement list--> Error list**.

Łaďadowanie poprawnie napisanego programu do sterownika odbywa się poprzez wybranie opcji **Statement list--> Load program**. Należy pamiętać, by podczas ładowania, czerwony przycisk uruchamiający program, był wyłaczony. W przeciwnym przypadku program nie zostanie załadowany. Do sterownika należy załadować program główny oraz wszystkie wykorzystywane w nim procedury.