

OPIS KONTROLERA

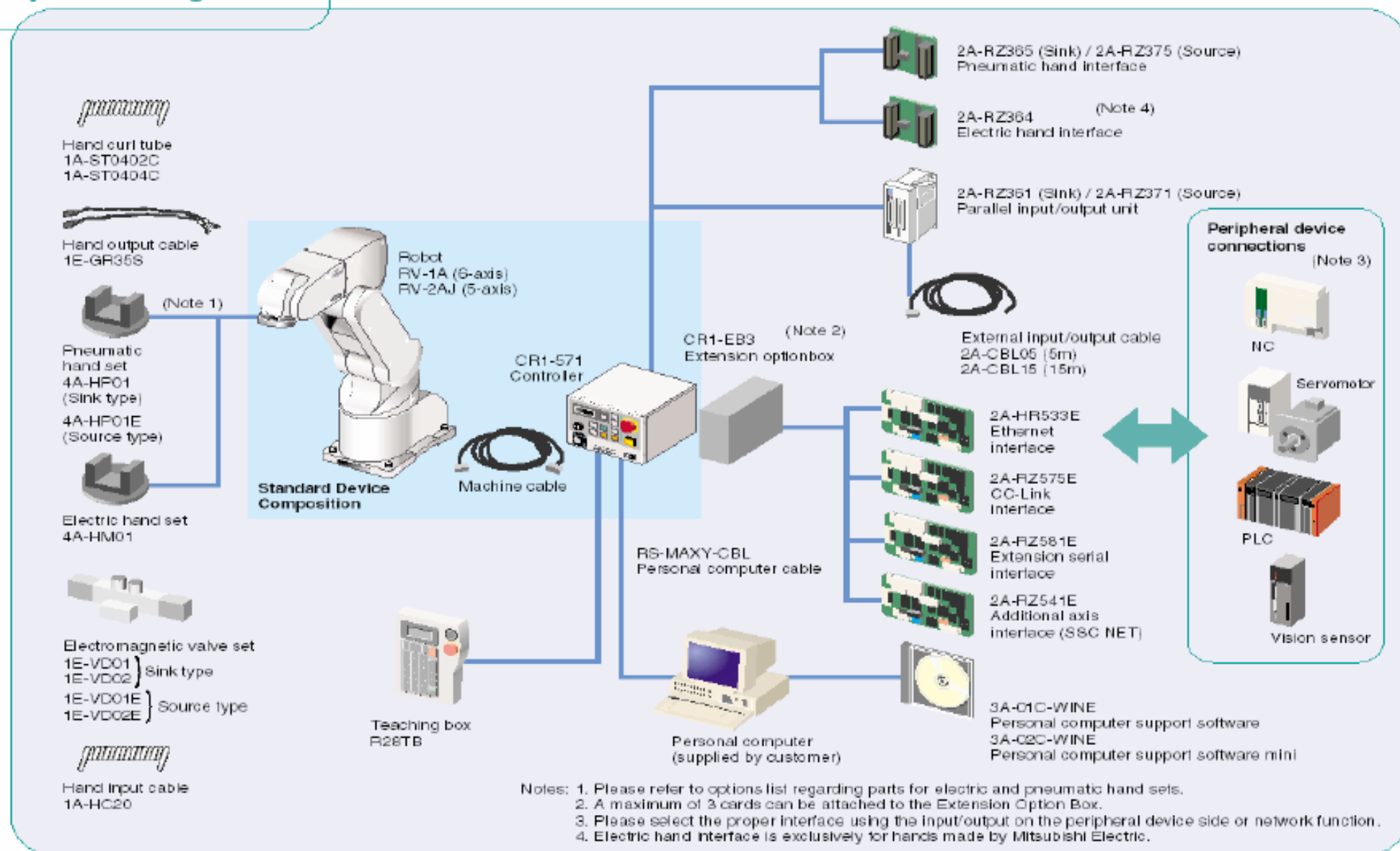
Mózgiem tego kontrolera jest 64 - bitowe CPU, które w trybie wielozdaniowym może wykonywać do 32 zadań jednocześnie.

Oznacza to, że gdy RV-2AJ wykonuje sekwencyjne ruchy, to może on jednocześnie, przez interfejs, odbierać dane o pozycji, włączać wejścia i wyjścia, dokonywać obliczeń a ponadto wykonywać jeszcze 28 innych zadań jednocześnie!

Kontroler robota może być wyposażony w dodatkowe karty, pozwalające na zwiększenie jego stopni swobody, co daje możliwość nieograniczonych zastosowań. Dla przykładu, przestrzeń robocza może być powiększona przez dodanie osi liniowej. Pozwala to efektywnie zrealizować automatyzację w kilku maszynach, liniach produkcyjnych, przy wymianie narzędzi lub zrobotyzowanych laboratoriach.

KONFIGURACJA ROBOTA

System Configuration



PANEL OPERATORSKI ROBOTA

Do sterowania, programowania, bądź implementacji sekwencji nadzorczych robota służy panel operatorski.

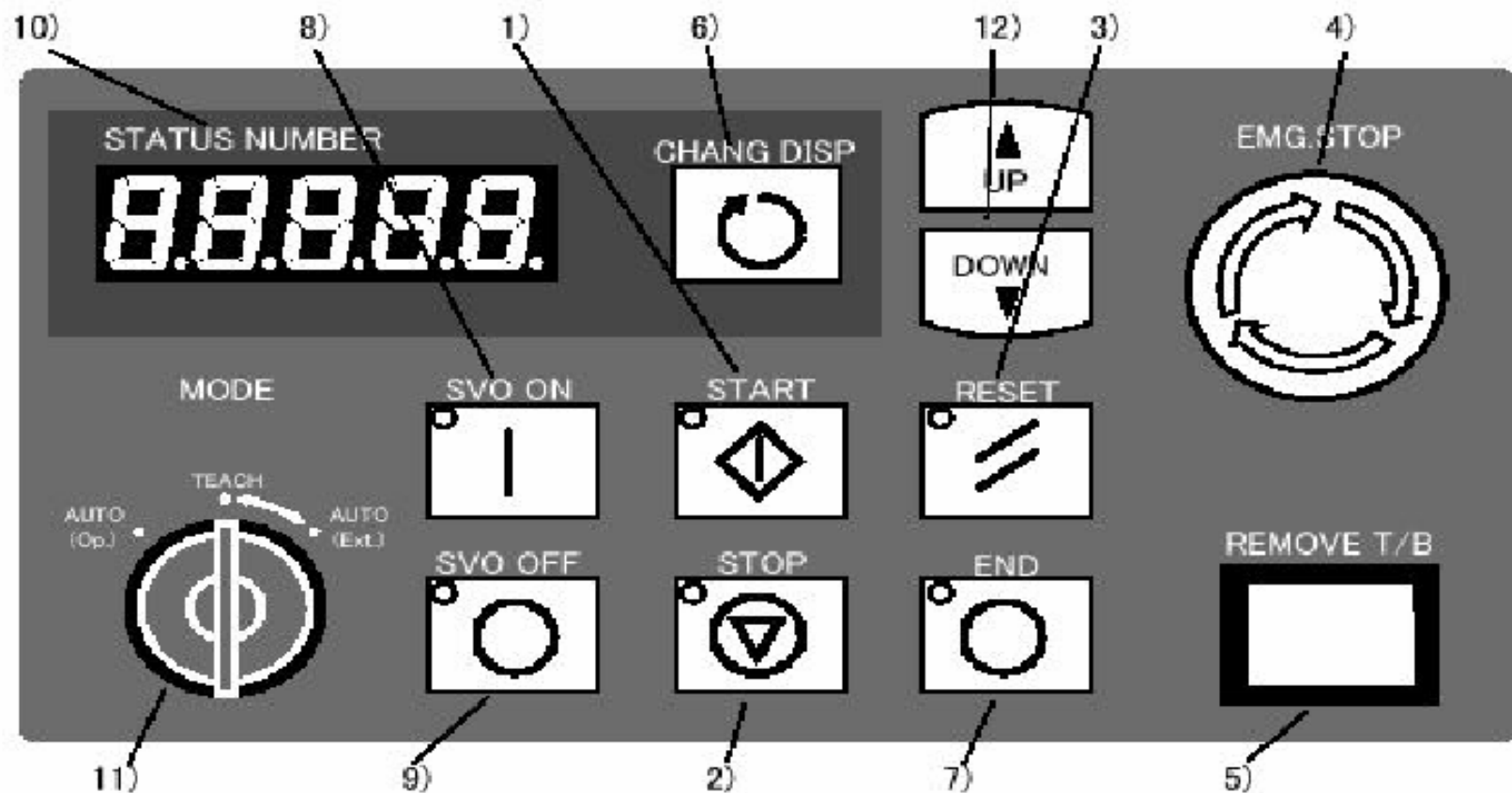
Opisywany model posiada dwojaki rodzaj takiego panelu, panel sterujący wbudowany i panel sterujący ręczny. Różnią się one pomiędzy sobą sposobem obsługi a także umiejscowieniem.

Pierwszy z nich umiejscowiony jest na samej jednostce robota, a drugi jest sprzężony z nią za pomocą odpowiedniego przewodu.

Takie rozwiązanie narzucone jest koniecznością sterowania zdalnego robota podczas pracy.

PANEL STERUJĄCY WBUDOWANY

ROZMIESZCZENIE PRZYCISKÓW I WYGLĄD ZEWNĘTRZNY PANELU



1→PRZYCISK STARTU - Powoduje uruchomienie programu robota i jego wykonywanie.

Operacje tą sygnalizuje zielona dioda LED, gdy program jest tylko sekwencją wejściową i nie jest umieszczony w pamięci dioda jest wyłączona.

2→PRZYCISK STOPU - Powoduje on natychmiastowe zakończenie pracy robota, bez wyłączenie wszystkich serwomotorów. Czerwona dioda LED sygnalizuje tą operacje.

3→PRZYCISK RESETU - Powoduje zresetowanie całego systemu, całego wykonywanego programu, a także wszystkich przerwań wywołanych w systemie. Przycisk ten służy także do wyczyszczenia zaistniałych błędów w Systemie.

4→PRZYCISK AWARYJNY - Sprawia natychmiastowe zakończenie pracy robota, z powrotem wszystkich przegubów do pozycji podstawowej. Powoduje także wyłączenie wszystkich serwomotorów.

5→PRZEŁĄCZNIK T/B - Przycisk służy do załączania i wyłączania panelu ręcznego. Panel ręczny powinien być od instalowany od jednostki w ciągu pięciu sekund po naciśnięciu switcha. Tak samo jest z jego instalacją.

6→PRZEŁĄCZNIK ZMIANY TRYBU WYŚWIETLANIA. Stosuje się go w celu zmiany rodzaju i sposobu wyświetlania komunikatów systemowych.

7→ PRZYCISK KOŃCA. Powoduje zatrzymanie podprogramu bądź programu w miejscu przetwarzania tego programu gdzie wystąpiła komenda END.

8→SVO.ON. Przełącznik powoduje włączenie zasilania serwomotorów i ich uruchomienie. Sygnalizowane jest to zaświeceniem zielonej diody LED.

9→ SVO.OF. Przełącznik powoduje odłączenie zasilania od serwomotorów i ich zatrzymanie. Operacja ta jest sygnalizowana zaświeceniem czerwonej diody LED.

10→WYŚWIETLACZ NUMERU STATUSU. Powoduje wyświetlanie znaku statusu informującego nas o zaistniałych zmianach w systemie np.: błędzie lub braku programu.

11→PRZEŁĄCZNIK TRYBU PRACY ROBOTA

Przełącznik ten powoduje zmianę rodzaju pracy robota, wyróżniamy trzy tryby pracy robota:

AUTO (op.): Operacje implementowane w systemie są tylko z wewnętrznego kontrolera, nie ma możliwości implementacji z panelu Ręcznego.

TEACH: Operacje aplikowane w urządzeniu są ważne tylko z zewnętrznego panelu ręcznego.

AUTO (Ext.): System przyjmuje operacje tylko z interfejsów komunikacyjnych zewnętrznych za wyjątkiem panelu ręcznego.

11→PRZEŁĄCZNIK UP/DOWN

Przełącznik ten pozwala na przewijanie w górę i w dół komunikatów pojawiających się na wyświetlaczu.

UWAGI:

podczas zmiany trybu pracy robota serwomotory zostają wyłączone podczas uruchamiania panelu ręcznego bądź wbudowanego serwomotory także zostają wyłączone.

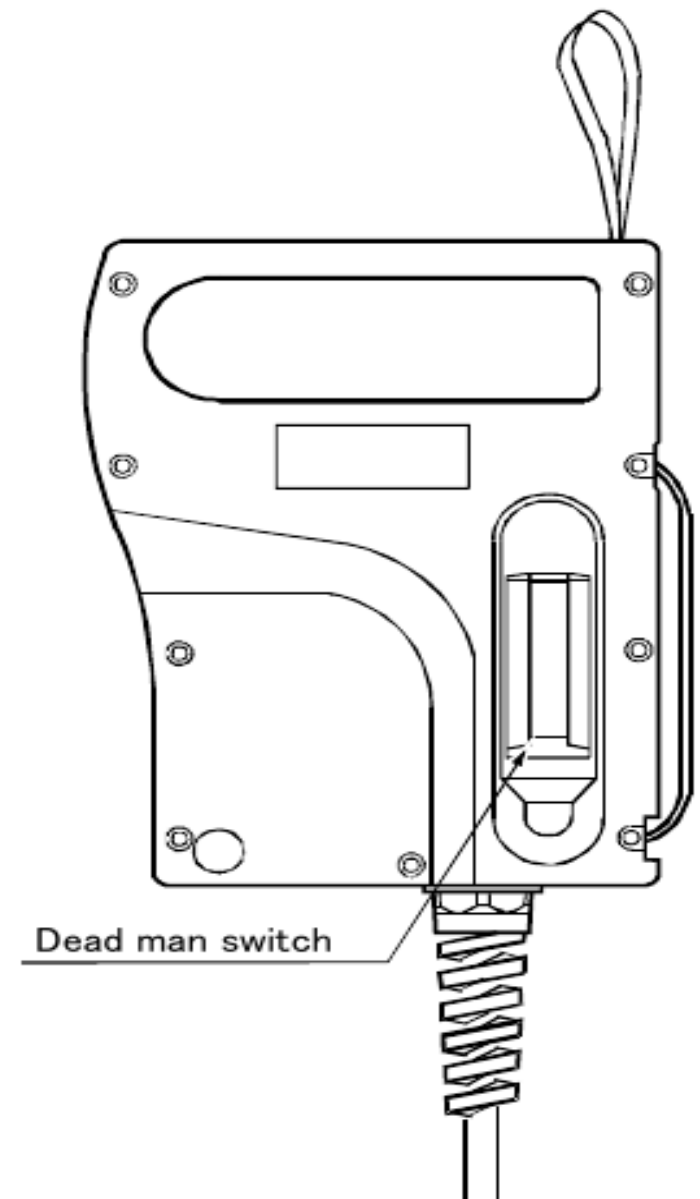
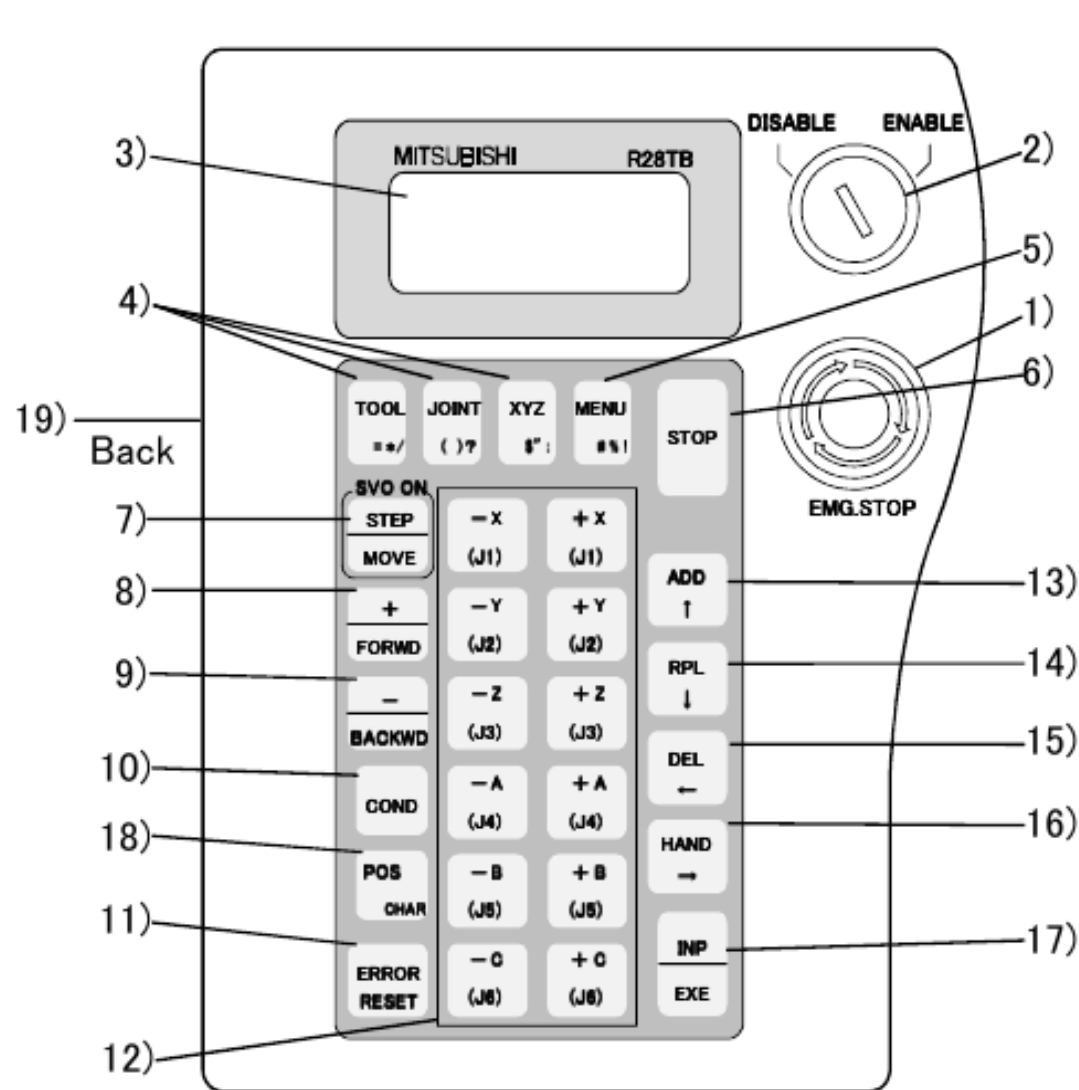
RODZAJE ZNAKÓW UŻYTYCH DO WYŚWIETLANIA KOMUNIKATÓW

Na panelu mamy do dyspozycji pięć wyświetlaczy siedmiosegmentowych na których są wyświetlane nazwy pracujących programów bądź inne zmienne systemowe.

Dopuszczalne jest tworzenie nazw programów mających długość maksymalną równą czterem pozycjom. Zestaw tych znaków przedstawiono poniżej.

A	A	B	b	C	c	D	d	E	E	F	F	G	G	H	H	I	I
J	J	K	K	L	L	M	M	N	N	O	O	P	P	Q	Q	R	R
S	S	T	T	U	U	V	V	W	W	X	X	Y	Y	Z	Z		

ROZMIESZCZENIE PRZYCISKÓW I WYGLĄD ZEWNĘTRZNY



FUNKCJE I NAZWY PRZYCISKÓW PANELA RĘCZNEGO

1→PRZYCISK AWARYJNY. Przycisk awaryjny z funkcją blokowania, po przyściśnięciu powoduje zatrzymanie serwomotorów natychmiastowo, nie zależnie od przełącznika ENABLED/DISABLED.

2→Przełącznik ENABLE / DISABLE. Zmiana tego przełącznika powoduje uaktywnienie bądź wyłączenie zewnętrznego panelu operatorskiego. Aby przeprowadzać operacje sterowania z panelu zewnętrznego przełącznik musi być ustawiony na pozycji ENABLE, wówczas panel ten jest aktywny a funkcjonowanie panelu wbudowanego i urządzeń zewnętrznych jest niemożliwe. Funkcjonowanie panelu na kontrolerze oraz sterowanie z urządzeń zewnętrznych jest możliwe po przełączeniu przełącznika na pozycje DISABLE. W tej pozycji musi być przełącznik także podczas edytowania programu w celu jego zapisania i zachowania w pamięci.

3→ Wyświetlacz LCD. Powoduje wyświetlanie zawartości programu oraz stanu robota wraz z operacjami wprowadzanymi z klawiatury.

FUNKCJE I NAZWY PRZYCISKÓW PANELA RĘCZNEGO

- 4→ **Klawisz narzędziowy [TOOL]** Klawisz wyboru rodzaju używanych narzędzi.
- 5→ **Klawisz przyłączania[JOINT]** Klawisz powoduje uaktywnienie funkcji axis jog mode.
- 6→ **Klawisz wyboru osi układu współrzędnych XYZ** Powoduje wybór i ustawienie osi układu współrzędnych pracy ramion robota.
- 7→ **Klawisz MENU** Powoduje zwrócenie zawartości menu na wyświetlany ekran. Gdy przycisk ten jest wciśnięty podczas edytowania właściwy program jest zachowany.
- 8→ **Klawisz stopu** Klawisz służy do zatrzymania pracy programu i funkcjonowania samego robota. Pełni on taką samą funkcję jak przycisk stopu na panelu wbudowanym może być używany gdy przełącznik ENABLED/DISABLED jest ustawiony na disable (jest nieaktywny).

FUNKCJE I NAZWY PRZYCISKÓW PANELA RĘCZNEGO

10→Klawisz STEP / MOVE. Kiedy klawisz ten jest wciśnięty wraz z klawiszem nr 12 jednocześnie możliwe są do wykonania tzw. jog operations. Kiedy klawisz ten jest wciśnięty jednocześnie z klawiszem nr 17 możliwa jest operacja krokowego skoku ramiona robota. Klawisz ten może też służyć do włączania serwomotorów.

11→Klawisz [+ /FORWD]. Operacja przemieszczenia krokowego w przód jest możliwa kiedy klawisz jest wciśnięty jednocześnie z klawiszem nr 17. Na edytorskim ekranie wyświetlana jest następna linijka programu. Wciśnięcie tego klawisza wraz z klawiszem nr 7 podczas trwania operacji robota powoduje wzrost prędkości ramiona robota. Możliwa ta operacja jest do wykonania nawet gdy panel ręczny jest nieaktywny.

12→ Klawisz [- /BACKWD]. Gdy klawisz jest wciśnięty na wyświetlaczu wyświetlana jest poprzednia linijka programu. Równoczesne wciśnięcie tego klawisza z klawiszem 17 powoduje powrót ramion robota według poprzednio określonej trajektorii (jakby wstecz) i według poprzednio przyjętych osi układu współrzędnych. Wciśnięcie tego klawisza wraz z klawiszem nr 7 podczas trwania operacji robota powoduje zmniejszenie prędkości ramienia robota. Operacja ta możliwa jest do wykonania nawet gdy panel ręczny jest nieaktywny.

FUNKCJE I NAZWY PRZYCISKÓW PANELA RĘCZNEGO

13→ [COND]. Klawisz służy do przywoływania ekranu instrukcji programu na wyświetlaczu.

14→ Klawisz [ERROR RESET]. Klawisz powoduje resetowanie wszystkich błędów które zaistniały w systemie. Naciśnięcie go wraz z klawiszem nr 17 powoduje reset programowy.

15→ Klawisze[Jog operations]. Klawisz ten służy do definiowania rodzaju układu współrzędnych robota oraz jego parametrów, umożliwia on np.: rotacje osi układu współrzędnych. Wciśnięty wraz z klawiszem XYZ jog mode powoduje przejęcie kontroli nad ramionami przez system – możemy ze sterownika poruszać ramionami. Klawisz ten służy też do wprowadzania numerycznych zewnętrznych sekwencji sterujących.

16→ Klawisz [ADD1]. Służy do przewijania kursora w menu do góry. Można także dodawać bądź korygować zawartość danych na wyświetlaczu przez jednoczesne wciśnięcie go z przyciskiem [STEP] key na wybranej edytowanej pozycji.

FUNKCJE I NAZWY PRZYCISKÓW PANELA RĘCZNEGO

17→Klawisz [RPL/1]. Przycisk służy do przewijania kursora w dół. Wciśnięty jednocześnie z klawiszem [STEP] key powoduje wyświetlanie następnej pozycji menu po tej właściwej która jest wyświetlana bądź edytowana.

18→Klawisz [DEL]. Powoduje wykasowanie danych stojących na lewo od kursora. Sprawia także przesuwanie kursora w lewo.

19→Klawisz [HAND/-]. Wciskając ten klawisz jednocześnie z odpowiednimi kombinacjami klawiszy funkcyjnych możemy uzyskać odpowiednie kombinacje podstawowych ruchów ramion robota- ruch w górę, w dół. Klawisz ten powoduje przemieszczanie się także kursora w prawo na ekranie wyświetlacza.

20→Klawisz [INP / EXE]. Klawisz powoduje uruchomienie zaimplementowanych programów i algorytmów robota, odpowiednia kombinacja tego klawisza z innymi umożliwia uruchomienie specyficznych procedur robota opisanych wcześniej.

21→Klawisz [POS CHAR]. Umożliwia wyświetlenie aktualnej pozycji wykonywanego bądź edytowanego programu. Umożliwia wprowadzenie do edytowanego programu etykiet bądź oznaczeń i opisów.

FUNKCJE I NAZWY PRZYCISKÓW PANELA RĘCZNEGO

22→Przełącznik DEADMAN. Kiedy przełącznik [ENABLE / DISABLE] jest ustawiony na pozycji enable i przełącznik jest zwolniony bądź wciśnięty szybko serwomotory zostaną wyłączone. Po delikatnym wciśnięciu przycisku i odczekaniu pewnej chwili serwomotory zostają włączone.

23→Przełącznik ustawiania kontrastu. Przełącznik ten powoduje ustawienie parametrów wyświetlacza LCD takich jak kontrast i jasność.

UWAGA:

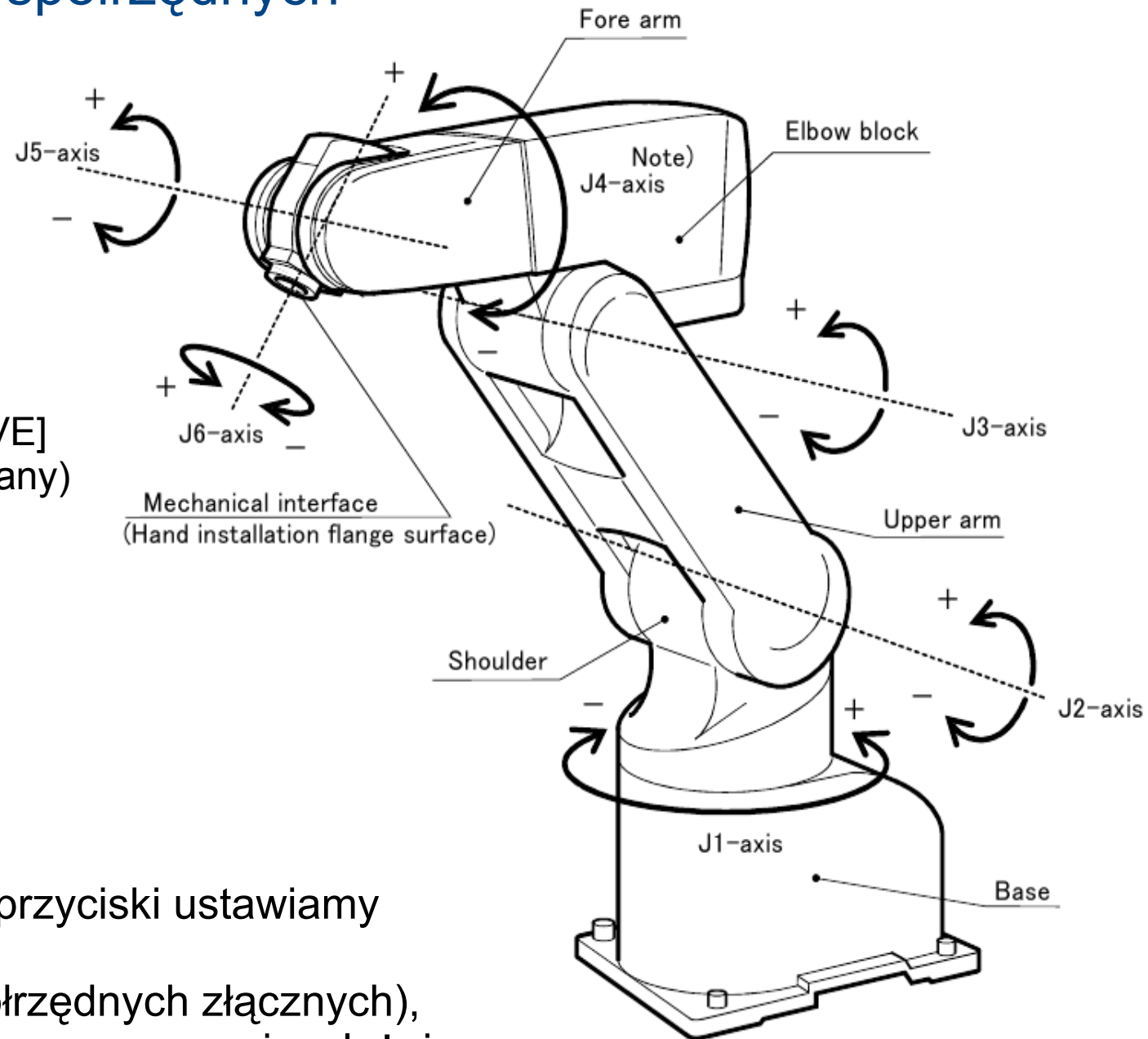
1.Kiedy do gniazd wejściowych jednostki centralnej robota są podłączone jednocześnie komputer PC bądź inny interfejs sterujący wówczas priorytet sterowań posiada urządzenie zewnętrzne, sterowanie z komputera jest niemożliwe.

2.Kiedy przełącznik [MODE] jest ustawiony na którejś z pozycji auto a panel zewnętrzny jest aktywny – pozycja enable w takim przypadku wyświetlany jest błąd systemowy.

Ustawienia osi współrzędnych

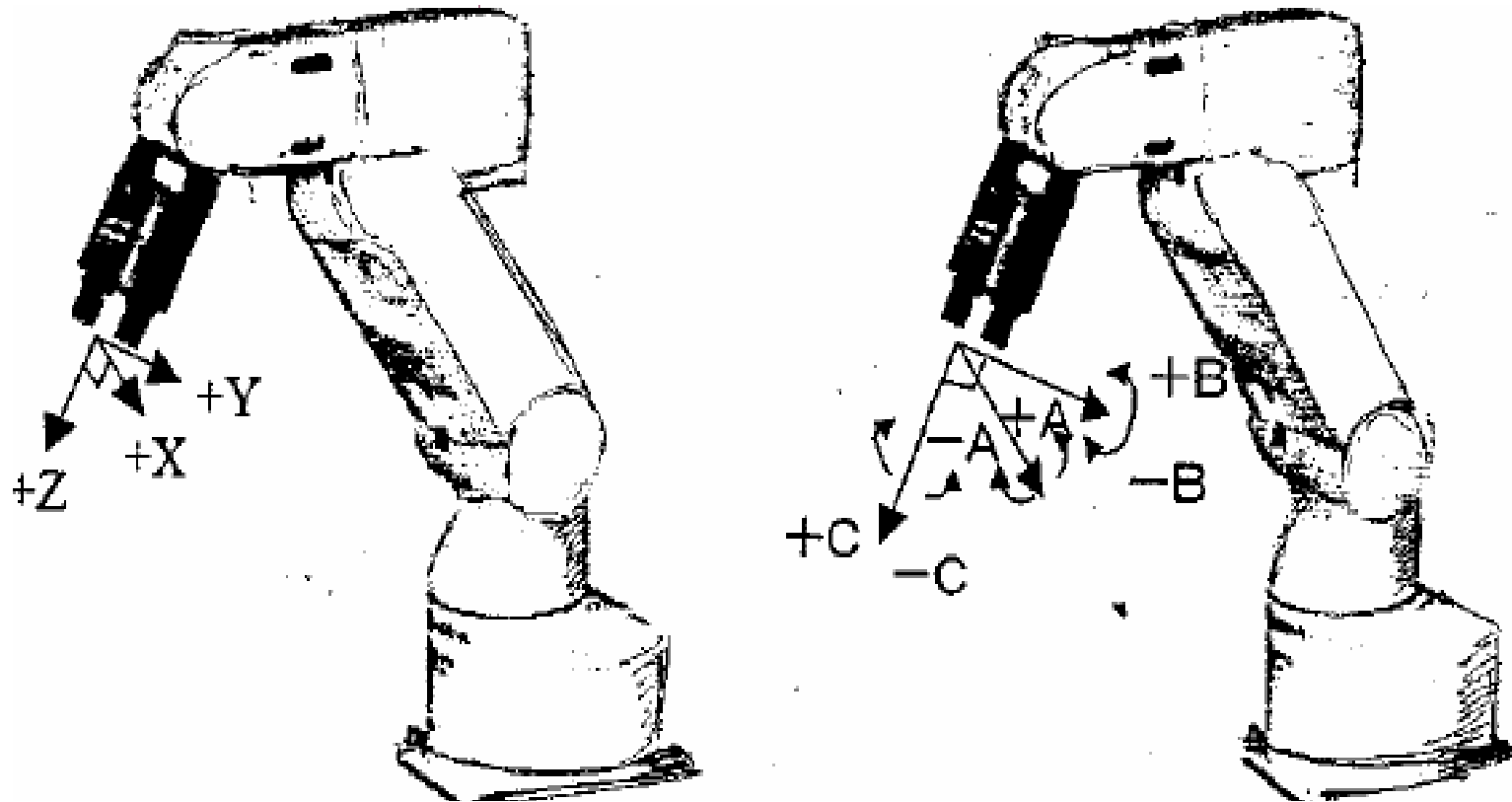
Ustaw przełącznik w pozycji
[ENABLE]
Naciśnij przycisk [STEP/MOVE]
(serwomechanizm jest włączany)
Naciśnij przycisk [JOINT]
(zostanie ustawiony tryb
współrzędnych złącznych)

- Naciskając odpowiednio przyciski ustawiamy kolejno osie od J1 do J6
- W tym trybie (trybie współrzędnych złącznych), każda z osi może być dostosowywane niezależnie.



Note) J4-axis doesn't exist for 5-axis type.

USTAWIENIA OSI WSPÓŁRZĘDNYCH NARZĘDZIA



- Ustaw przełącznik w pozycji [ENABLE]
- Naciśnij przycisk [STEP/MOYE] (serwomechanizm jest włączany)
- Naciśnij przycisk [TOOL] aby przejść do trybu współrzędnych narzędzia
- Naciśnij odpowiedni przycisk należy ustawić odpowiednie osie: XY,Z,A,B,C

Pozycja może być dostosowywana: w przód / w tył, w lewo / w prawo albo w górę / na dół względem kierunku końca ręki robota (" TOOL " ustawia współrzędnego systemu).

Koniec ręki robota porusza się linearnie (interpolacja liniowa) . Podstawa może się obracać dookoła osi X, Y i Z we współrzędnych narzędzia. Naciskając przyciski A, B i C pozycja końca ręki pozostanie niezmienna.

Konieczne jest wyszczególnienie długość narzędzia przed wykonaniem ruchu ustawiając parametry przyciskiem [MEXTL].

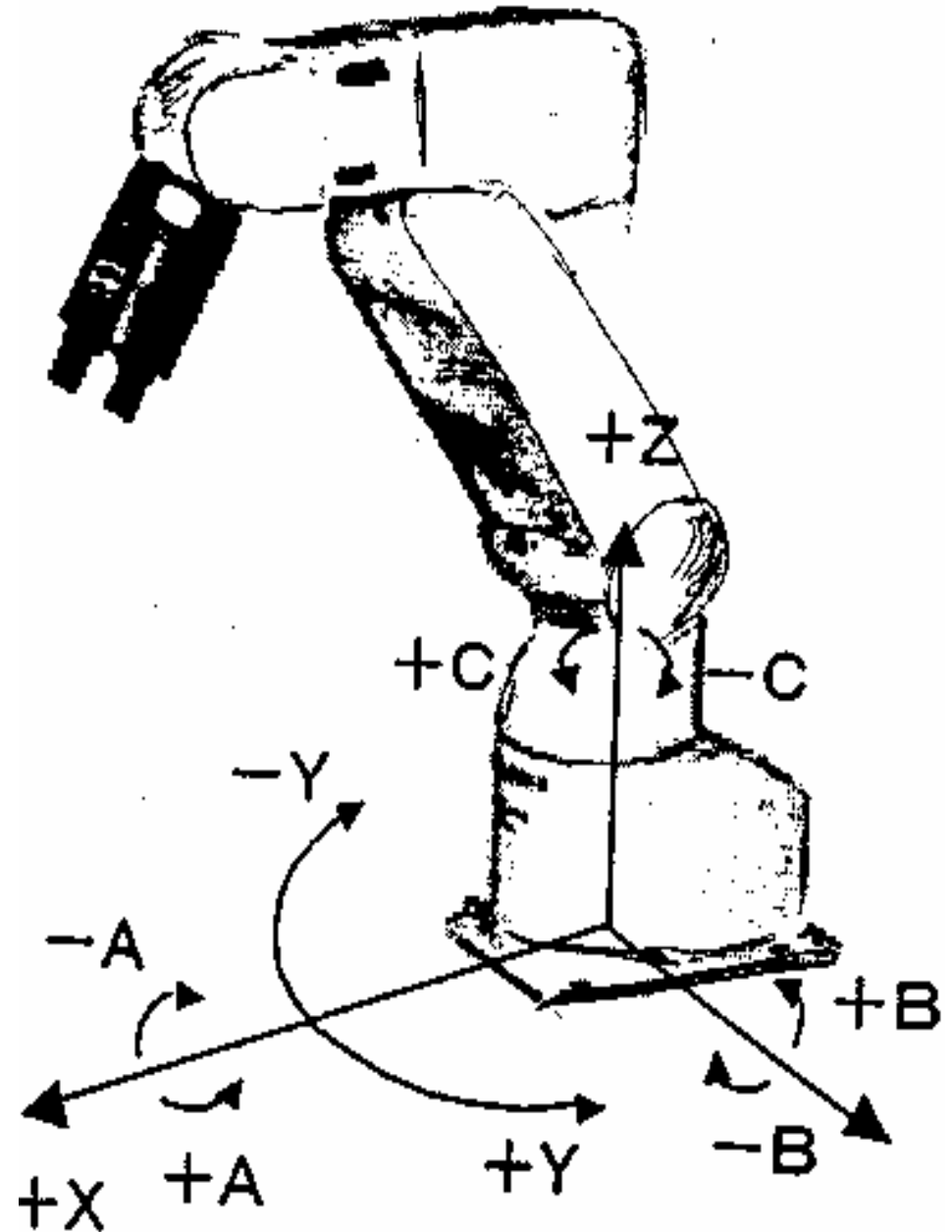
Współrzędne narzędzia systemu, w których jest definiowane położenie końca ręki zależne jest od typu robota.

W przypadku pionowego multi- połączenia kierunek : od mechanicznego interfejsu płaszczyzny do końca ręki, zapisywany jest jako +Z.

W przypadku poziomego multi- połączenia robota, ruch od mechanicznego interfejsu płaszczyzny ku górze, zapisywany jest jako + Z

WSPÓŁRZĘDNE CYLINDRYCZNE

- Ustaw przełącznik w pozycji [ENABLE]
- Naciśnij przycisk [STEP/MOVE]
- (serwomechanizm jest włączany)
- Naciskając przycisk [XYZ] dwukrotnie
- ustawiony zostaje tryb współrzędnych
- cylindrycznych



SPECYFIKACJA JĘZYKA MELFA BASIC IV

Nazwa programu

Nazwa programu może zawierać do 12 znaków, jednak panel sterujący wyświetla do 4 znaków i dlatego zaleca się nazwy nie dłuższe niż 4 znaki. Mogą być używane duże znaki alfabetu oraz cyfry. Jeżeli nazwa programu ma więcej niż 4 znaki to nie można go wybrać z panelu sterowania.

Dodatkowo jeżeli pożądanym jest zewnętrzny wyjściowy sygnał do wybrania programu do wykonania program powinien mieć jako nazwę numer.

Podprogram może mieć nazwę składającą się z 12 znaków.

STRUKTURA POLECENIA

10	MOV	P1	WITH M OUT(17)=1
(1)	(2)	(3)	(4)

1) Numer linii

2) Komenda

3) Dany punkt

4) Dołożona na końcu dyrektywa. Słowa komend ruchu: MVS, MVR, MVR2, MVR3, MVC, MOV mogą być połączone z dodawanym na końcu stwierdzeniem (tylko komendy ruchu).

Pozwala to na wykonanie kilku komend równolegle z komendą ruchu

STRUKTURA POLECENIA

Instrukcja- jest minimalną jednostką konfigurującą program, np. **MOV P1**

Linia - składa się z numeru i instrukcji; zawiera do 127 znaków !

Tylko 1 instrukcja na linię!- nie można np. oddzielać kolejnych instrukcji średnikiem.

Numery linii powinny być w porządku rosnącym od pierwszej linii programu.

Numery linii to liczby całkowite 1- 32767. Jeśli dana instrukcja nie ma przypisanego numeru linii na ekranie ręcznego panelu sterującego to jest ona wykonywana zaraz po pojawieniu się na wejściu - jest to bezpośrednie wykonanie.

W takim przypadku komenda nie zostanie zapisana w pamięci ale jej wynik zostanie zapisany w zmiennych.

STRUKTURA POLECENIA

Etykieta - jest nazwą zdefiniowaną przez użytkownika; służy do oznaczania rozgałęzień.

Tworzy się je przez wstawienie znaku gwiazdki „*” ze znakami alfabetu zaraz po numerze linii.

Jeśli etykieta zaczyna się od litery „L” zaraz po gwiazdce to po tej literze może być użyty znak podkreślenia „_”

Znaki których nie można używać : Zarezerwowane słowa (DLY, HOPEN itd.)

Symbole i cyfry.

Każda nazwa już wykorzystana jako np. zmienna czy nazwa funkcji

Przykład etykiety:

```
10 GOTO *LBL
```

```
.....  
100 *LBL
```

Znaki mające specjalne znaczenie

1) Identyfikacja znaków pisanych małymi i dużymi literami

Małe litery zostaną pozostawione jako małe tylko jeśli są używane w komentarzach lub są to dane typu „Słowo”, w przeciwnym razie zostaną zamienione na duże w czasie wczytywania programu.

2) Podkreślenie „_”

Znak podkreślenia jest używany jako drugi znak nazwy zmiennej aby określić, że jest to zmienna zewnętrzna (dla różnych programów). Np. P_CURR , M_01 , M_ABC

3) Apostrof „'”

Jest używany do oznaczenia komentarza. Jeśli jest to pierwszy znak w linii to zastępuje on komendę REM. Np.

```
100 MOV P1 'GET
```

```
150 'GET PARTS
```


Znaki mające specjalne znaczenie

1) Identyfikacja znaków pisanych małymi i dużymi literami

Małe litery zostaną pozostawione jako małe tylko jeśli są używane w komentarzach lub są to dane typu „Słowo”, w przeciwnym razie zostaną zamienione na duże w czasie wczytywania programu.

2) Podkreślenie „_”

Znak podkreślenia jest używany jako drugi znak nazwy zmiennej aby określić, że jest to zmienna zewnętrzna (dla różnych programów). Np. P_CURR , M_01 , M_ABC

3) Apostrof „'”

Jest używany do oznaczenia komentarza. Jeśli jest to pierwszy znak w linii to zastępuje on komendę REM. Np.

```
100 MOV P1 'GET
```

```
150 'GET PARTS
```

4) Gwiazdka „ * ”

Jest umieszczana przed nazwą etykiety służącej do określenia miejsca docelowego skoku. Np. **200 *CHECK**

5) Przecinek „ , ”

Służy do oddzielania kilku parametrów lub przyrostków. Np.

P1 = (200, 150, ...)

6) Kropka „ . ”

Jest używana aby otrzymać pewne elementy złożonych danych jak kropka dziesiętna, parametry pozycjonowania lub parametry złączne. Np.

M1=P2.X ‘Podstawia pozycje zmiennej **P2.X** współrzędnej elementu do zmiennej numerycznej **M1**

7) Spacja „ ”

Znak odstępu jest wymagany jako separator bezpośrednio po numerze linii albo po komendzie i pomiędzy pozycjami danych.

TYPY DANYCH

W MELFA BASIC IV możliwe jest użycie 4 typów danych:

- wielkości liczbowe,
- pozycyjne,
- Złączne
- Ciągi znaków.

Wielkości liczbowe są dodatkowo podzielone na rzeczywiste i całkowite.

Wszystkie typy danych mogą być zmiennymi i stałymi.

STAŁE

Stałymi mogą być wartości:

-Liczbowe:

- dziesiętne, np. 1, 1.7, -10.5, +1.2E+5
- szesnastkowe, np. &H0001, &HFFFF
- dwójkowe, np. &B0010, &B1111

-ciągi znaków

Napisy są ciągami znaków ujętymi w cudzysłów, np.

„ABCDEF” „123”

-Pozycyjne

Np. $P1 = (300, 100, 400, 180, 0, 180, 0, 0) (7, 0)$

Liczby odpowiadają współrzędnym na odpowiednich osiach.

-Złączne

Np. $J1 = (0, 10, 80, 10, 90, 0)$

Liczby odpowiadają współrzędnym na odpowiednich osiach.

-Kątowe

Stałe kątowe muszą być wyrażone w stopniach a nie w radianach.

Jeśli zapiszemy 100DEG to ta stała staje się kątem i możemy jej użyć w funkcjach trygonometrycznych, np. $SIN(90DEG)$

Zmienne

Nazwy zmiennych powinny mieć **do 8 znaków**. Typ zmiennej jest określony przez pierwszy znak w nazwie zmiennej.

Typy zmiennych: **arytmetyczne**: (znaki inne od C, P, J, zalecany M), **słowa** (C), **pozycyjne** (P), **złączne** (J) i typu we/wy.

Zmienne arytmetyczne mogą być jeszcze podzielone na

- zmienne całkowite (np. **M1=10**),
- rzeczywiste o pojedynczej dokładności (np. **M2!** = -1.73E+10),
- rzeczywiste o podwójnej dokładności (np. **M3#** = 0.123).

Poniższe 2 typy zmiennych są szeroko stosowane:

Lokalne zmienne – ważne tylko w programie

Zmienne statusu robota – zewnętrzne zmienne definiowane przez użytkownika; mają one znak podkreślenia **_** jako drugi znak w nazwie zmiennej.

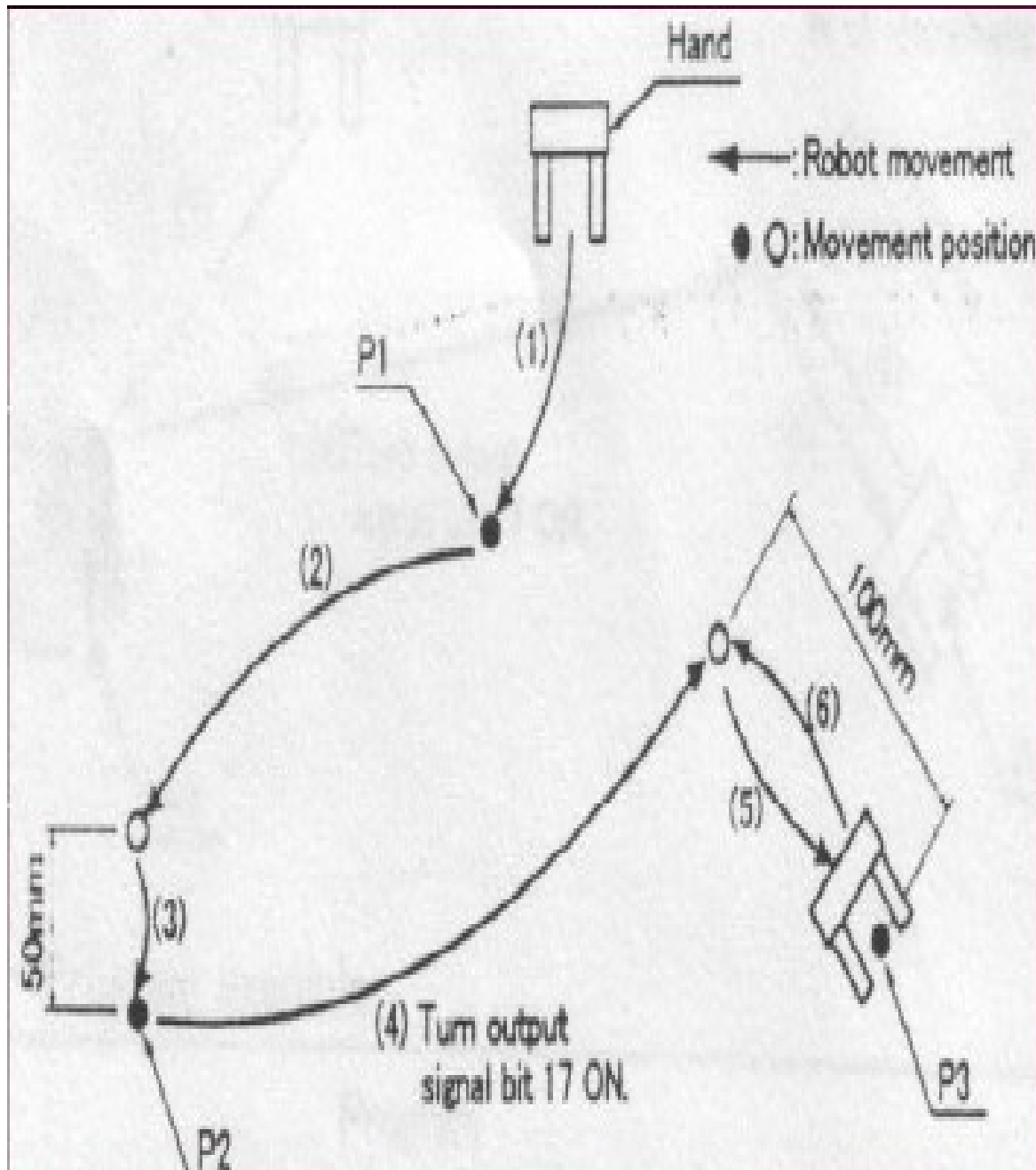
UWAGA: Zmienne nie są zerowane gdy zostaną wywołane, kiedy program jest ładowany albo resetowany.

KOMENDY DOTYCZĄCE RUCHU ROBOTA

MOV - robot porusza się do wyznaczonej pozycji z łączną interpolacją.

MOV P1 - Ruch z łączną interpolacją do punktu P1

MOV P1,-50 - Ruch z łączną interpolacją od punktu P1 do pozycji oddalonej 50 mm w kierunku ręki (cofa 50 mm).



PROGRAM

10 MOV P1

20 MOV P2, -50

30 MOV P2

40 MOV P3, -100 WTH M_OUT(17)=1

50 MOV P3

60 MOV P3, -100

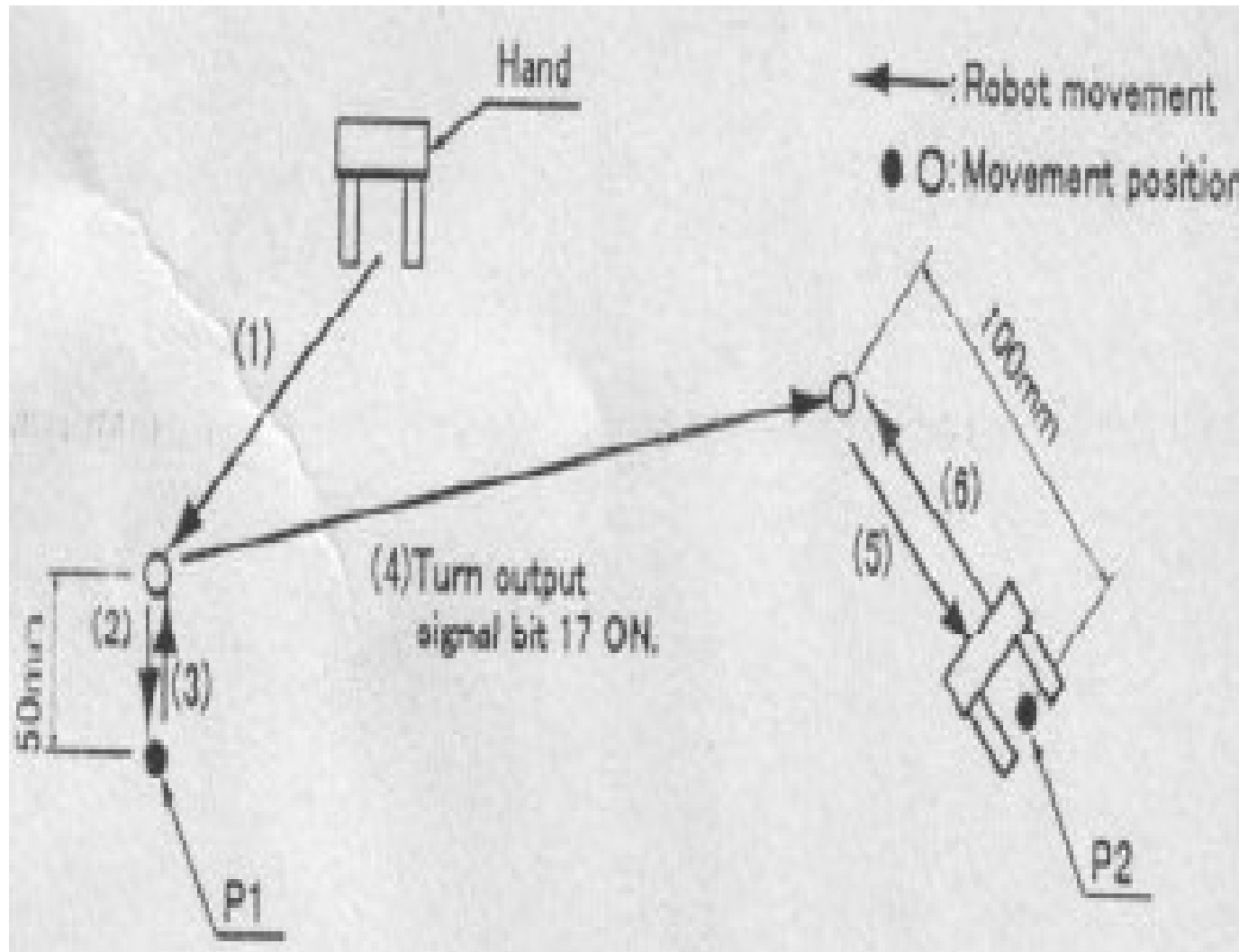
70 END

KOMENDY DOTYCZĄCE RUCHU ROBOTA

MVS - robot porusza się do wyznaczonej pozycji z interpolacją liniową.

MVS P1 - Ruch z interpolacją liniową do P1.

MVS P1, -100 - Ruch z interpolacją liniową od P1 do pozycji oddalonej o 100 mm w kierunku ręki.



PROGRAM

```
10 MVS P1, -50
20 MVS P1
30 MVS , -50
40 MVS P2, -100 WTH
M_OUT(17)=1
50 MVS P2
60 MVS , -100
70 END
```

KOMENDY DOTYCZĄCE RUCHU ROBOTA

MVR- ruch z interpolacją kołową. Wyznaczmy punkty:

początkowy ->przejściowy → końcowy

MVR P3, P4, P5 - Ruch z interpolacją kołową od P3 przez P4 do P5

MVR2 - wyznaczamy punkt początkowy, końcowy oraz odniesienia.

Robot porusza się kołową interpolacją w kolejności od początkowego punktu -> do końcowego punktu bez przechodzenia przez punkt odniesienia.

MVR2 P5, P7,P6 - Ruch z interpolacją kołową od P5 do P7 , a punktem odniesienia jest P6 i przez ten punkt ramię nie przechodzi.

KOMENDY DOTYCZĄCE RUCHU ROBOTA

MVR3 - ruch z interpolacją kołową w kolejności z wyznaczonymi punktami:

początkowy -> końcowy

Kąt tworzenia wynosi od 0 do 180 stopni.

MVR3 P7, P9, P8 - Ruch z interpolacją kołową od P7 do P9. P8 jest środkowym punktem . Ruch jest wykonywany pod jak najmniejszym kątem tworzenia.

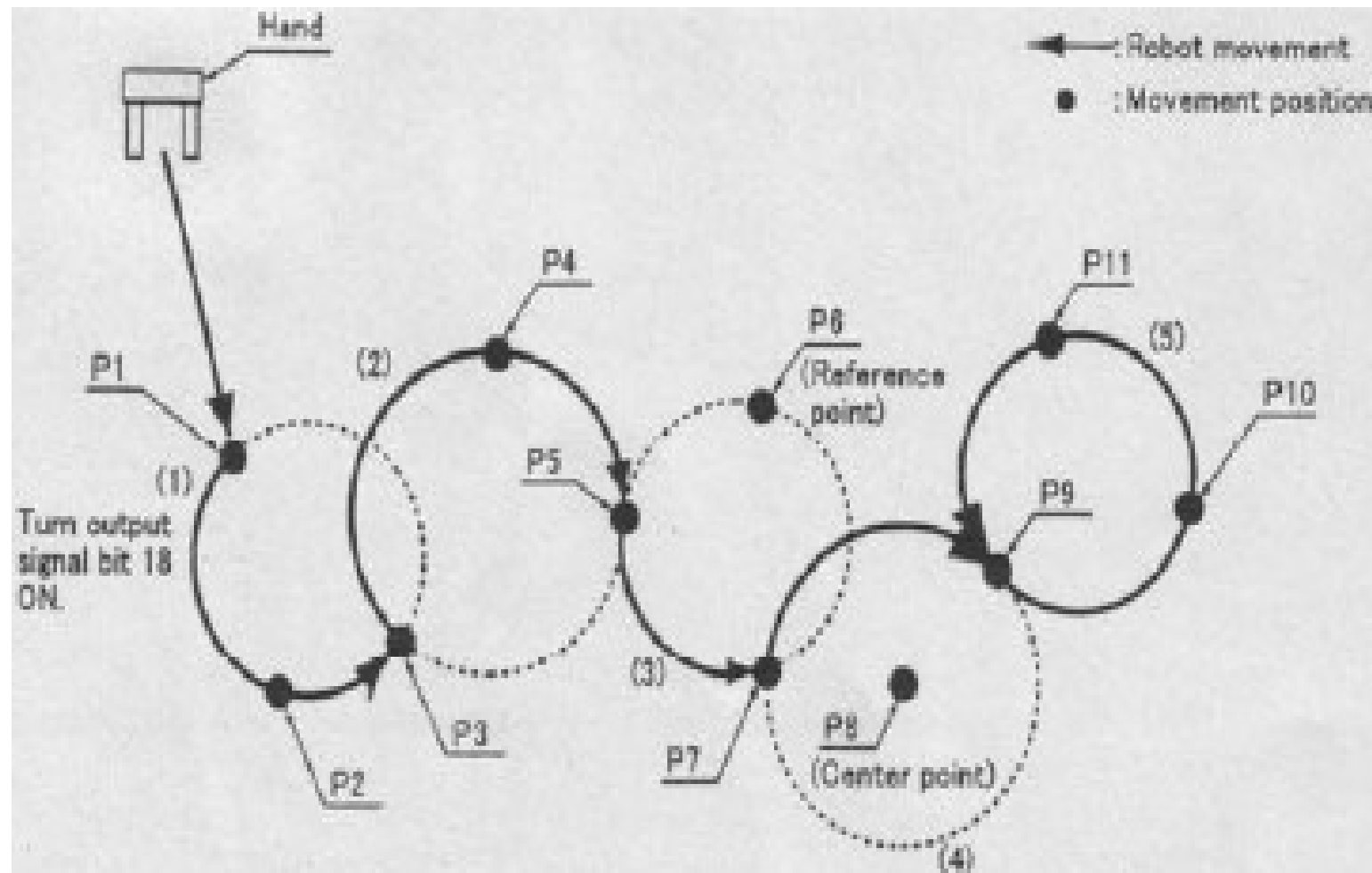
MVC- ruch z interpolacją kołową. Wyznaczamy punkty:

Początkowy -> przejściowy 1 → przejściowy 2 -> końcowy

MVC P1, P2, P3

- Robot porusza się z interpolacją kołową od P1 przez P2 i P3 do P1.

Zamknięte koło.



PROGRAM

```

10 MVR P1, P2, P3, WTH
M_OUT(18)=1
20 MVR P3, P4, P5
30 MVR2 P5, P7, P6
40 MVR3 P7, P9, P8
50 MVC P9, P10, P11
60 END

```

KOMENDY ODNOSZĄCE SIĘ DO PRĘDKOŚCI RUCHU ROBOTA

OVRD - wyznacza szybkość ruchu zastosowaną na niepodzielnym programie jako procentowość (%) w stosunku do prędkości maksymalnej.

OVRD 50 - Ustawiona szybkość ruchu wynosi 50% maksymalnej prędkości.

JOVRD - wyznacza łączną szybkość interpolacji jako procentowość (%) w stosunku do prędkości maksymalnej

JOVRD 70 - Ustawiona szybkość ruchu interpolacji na 70% szybkości maksymalnej

KOMENDY ODNOSZĄCE SIĘ DO PRĘDKOŚCI RUCHU ROBOTA

SPD - wyznacza linearną i kołową szybkość interpolacji ruchu końca ręki (mm / s)

SPD 120 -Szybkość końcówki ramienia wynosi 120 mm/s

KOMENDY ODNOSZĄCE SIĘ DO PRZYSPIESZENIA RUCHU ROBOTA

OADL - ta instrukcja wyszczególnia, czy optymalne przyspieszanie / opóźnienie funkcji powinno zostać włączone albo wyłączone

OADL ON/OFF -Włącza/Wyłącza optymalne przyspieszenie

KOMENDY ODNOSZĄCE SIĘ DO PRZYSPIESZENIA RUCHU ROBOTA

ACCEL - wyznacza przyspieszanie podczas ruchu i zmniejsza jako procentowość (%) w stosunku do maksymalnego przyspieszanie / zmniejszenia szybkości. Inaczej mówiąc ustawia przyśpieszenie i zmniejszenie prędkości do 100%.

ACCEL 60, 80 - Ustawia przyśpieszenie do 60% i zmniejszenie prędkości do 80%. (Dla maksymalnego przyśpieszenia/zmniejszenia prędkości jest 0,2 sekundy przyśpieszenia $0,2-0,6=0,33\text{sek.}$ Zmniejszenie prędkości $0,2-0,8=0.25\text{ sek.}$)

KOMENDY, KTÓRE OKREŚLA Z JAKĄ DOKŁADNOŚCIĄ ROBOT WYKONA DANE ZADANIE

PREC - ta instrukcja wyszczególnia, czy rodzaj dokładności (dokładności wykonania zadania) ścieżki dostępu powinien zostać włączony albo Wyłączony

PREC ON - umożliwia włączenie trybu dokładnego

PREC OFF - umożliwia wyłączenie trybu dokładnego

KOMENDY, KTÓRE STERUJĄ CHWYTAKIEM

HOPEN - wyznaczony chwytak otwiera się

HCLOSE - wyznaczony chwytak zamyka się

DLY - powoduje zatrzymanie ruchu

DLY 0.5 - czekaj 0.5 sek.

PALETYZACJA

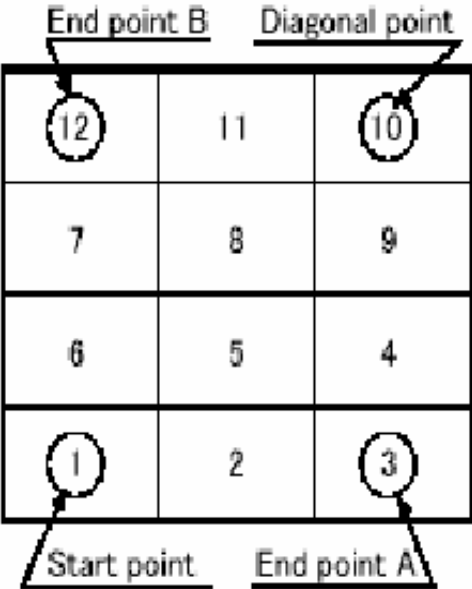
Przykładowe polecenia

Def Plt 1 , P1 , P2 , P3 , P4 , 4 , 3 , 1 -definiuje działanie paletyzacji No . 1
z początkiem w punkcie P1 , A=P2 i punkcie B=P3 i punkcie przekątnej =P4

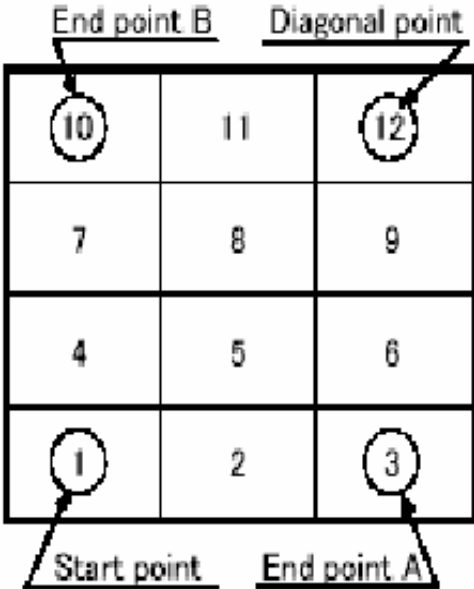
Def Plt 2 , P1 , P2 , P3 , P4 , 8 , 5 , 2 -definiuje działanie paletyzacji No. 2
z początkiem w punkcie =P1 , A=P2 i punkcie B=P3.

Def Plt 3 , P1 , P2 , P3 , P4 , 5 , 1 , 3 -definiuje działanie paletyzacji No. 3
stanowiącej łuk .Posiada pięć stanów pracy pozycja początkowa w punkcie =P1 .

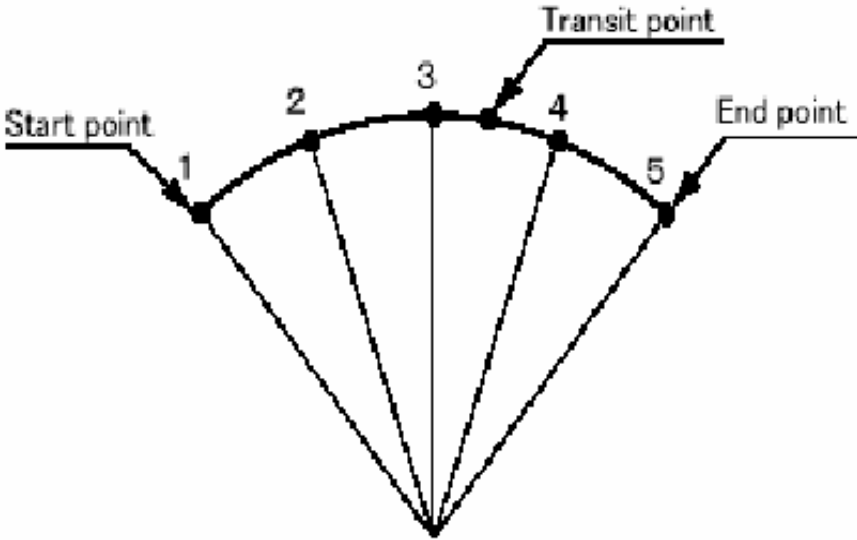
OBRAZOWE PRZEDSTAWIENIE POSZCZEGÓLNYCH FUNKCJI



Assignment direction = 1 (zigzag)



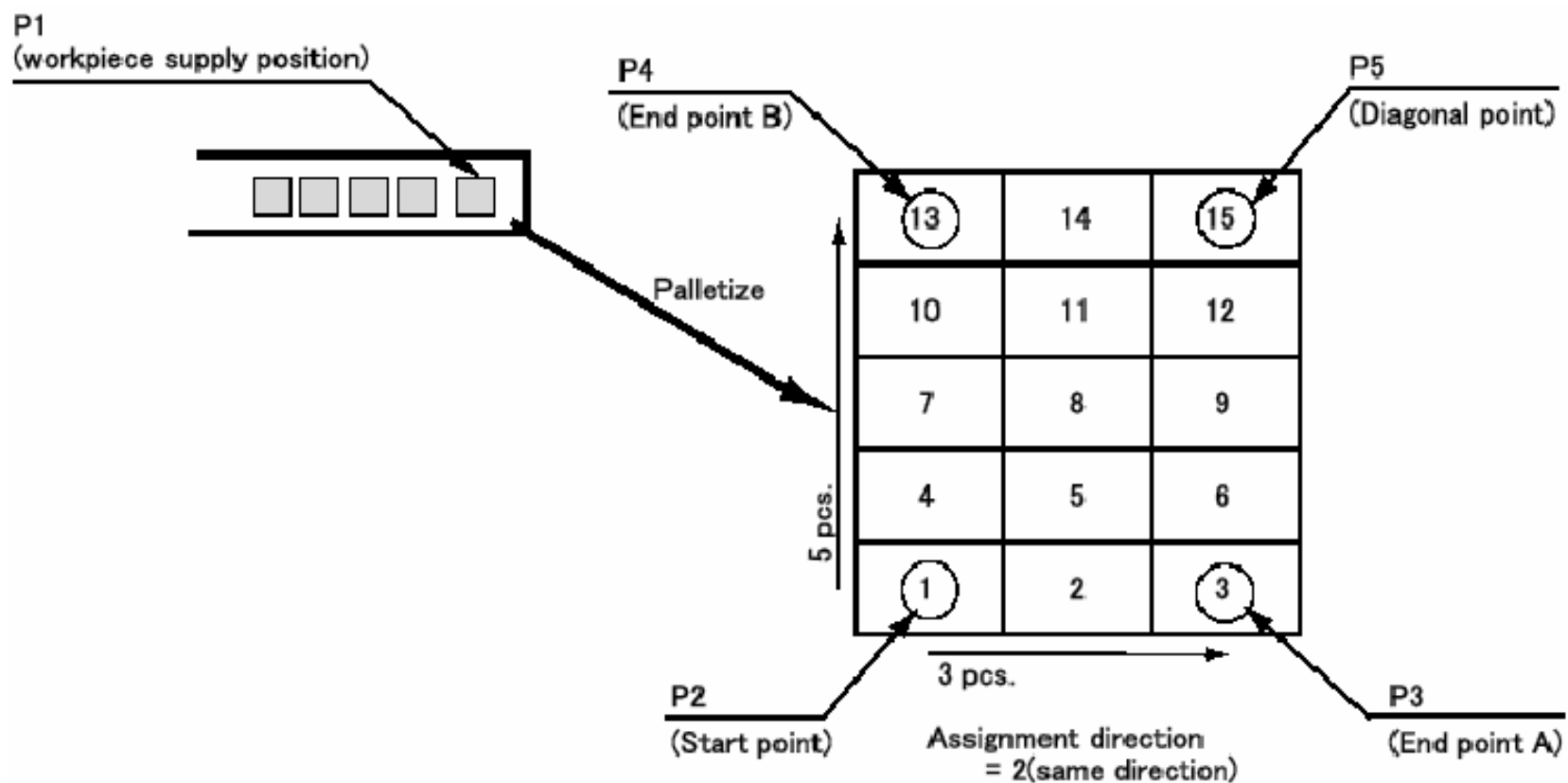
Assignment direction = 2 (same direction)



Assignment direction = 3 (arc pallet)

PALETYZACJA

Przykładowy program:



Przykładowy program

10 DEF PLT 1, P2, P3, P4, 3,5,2	'definicja paletyzacji
	'Paletyzacja No.1, punkt startowy -P2 , punkt
	'końcowy B=P4 , punkt przekątny point=P5
20 M1=1	'zastępuje wartość 1 zmienną liczbową M1
30 *LOOP	'definiuje polecenie pętli
40 MOV P1, -50	'ruch ze wspólną
	'interpolacją od P1 do pozycji odległej o 50 mm
50 OVERD 50	'nadanie prędkości robotowi
	'równej 50% z max. Prędkości jaka może rozwinąć
60 MVS P1	'przesunięcie linearne do pozycji P1
70 HCLOSE 1	'zamknięcie ręki 1

Przykładowy program c.d.

80 DLY 0.5	‘czekanie 0.5 sekundy
90 OVRD 100	‘nadanie maksymalnej prędkości).
100 MVS,- 50	‘przesunięcie linearne od ‘aktualnej pozycji (P1) do pozycji odległej o 50mm
110 P10=(PLT 1,M1)	‘działanie pozycji w paletyzacji No. 1 wskazywanej przez ‘zmienną liczbową M1
120 MOV P10, -50	‘przesunięcie od punktu ‘P10 do punktu odległego o 50mm
130 OVRD 50	‘nadanie prędkości robotowi ‘równej 50% z max. Prędkości jaka może rozwinąć
140 MVS P10	‘przesunięcie linearne do P10

Przykładowy program c.d

150 HOPEN 1	‘otwarcie ręki 1
160 DLY 0.5	‘czekanie 0.5 sekundy
170 OVRD 100	‘nadanie maksymalnej prędkości
180 MVS, -50	‘przesunięcie linearne od ‘aktualnej pozycji (P10) do pozycji odległej o 50mm
190 M1 =M1+1	‘zwiększenie ‘zmiennej liczbowej M1 o 1
200 IF M1<+15 THEN *LOOP	‘jeżeli wartość ‘zmiennej liczbowej M1 jest mniejsza niż15 , wtedy ‘przeskocz do początku i powtórz proces. ‘Jeżeli więcej niż 15 , idzie do następnego kroku
210 END	‘koniec programu

SYGNAŁY I INSTRUKCJE WEJŚCIA I WYJŚCIA

Robot jako jednostka wykonująca zadanie w pewnym otoczeniu, może być nadzorowana i sterowana podobnie jak sterownik PLC.

Jednostkę centralną robota stanowi pewien system mikroprocesorowy wyposażony w odpowiednie interfejsy we/wy zdolne do pozyskiwania i wysyłania sygnałów sterujących (bądź pomiarowych). Odpowiednim przepływem danych we/wy sterują zmienne systemowe robota.

Przykładowy rozkaz wejścia:

WAIT – oczekiwanie na podanie sygnału wejściowego

przykładowe zmienne wejściowe

M_IN, M_INB, M_INW, M_DIN,

SYGNAŁY I INSTRUKCJE WEJŚCIA I WYJŚCIA

Przykładowa składnia instrukcji

Statement example	Explanation
WAIT M_IN (1) = 1	Waits for the input signal bit 1 to turn ON.
M1 = M_INB (20)	Substitutes the input signal bit 20 to 27, as an 8-bit state, in numeric variable M1.
M1 = M_INW (5)	Substitutes the input signal bit 5 to 20, as an 16-bit state, in numeric variable M1.

Przykładowy rozkaz wyjścia:

CLR – „czyści” wszystkie sygnały wyjściowy wystawione w poprzednich operacjach

Przykładowe zmienne wyjściowe:

M_OUT, M_OUTB, M_OUTW, M_DOUT,

ZAGNIEŹDZENIE STRUKTUR PROGRAMOWYCH

Podczas tworzenia programu robota musimy pamiętać, że głębokość zagnieżdżenia struktur sterujących programem jest ograniczona. Wynika to ze specyfiki języka danego robota, i tak głębokość ta może wynosić 8, 16 bądź 800 poziomów maksymalnie w zależności od rodzaju wykonywanego programu.

Przekroczenie danego poziomu przez daną strukturę programową równoważne jest z wadliwym działaniem programu.

INSTRUKCJE STERUJĄCE PROGRAMEM (WARUNKOWE,BEZWARUNKOWE, CZEKANIE)

GOTO -skacze bezwarunkowo do wskazanej linii

np. **GOTO 200** ‘ skacze do linii 200

ON GOTO -skacze zgodnie z wartością wskazanej zmiennej, wartość warunku to całkowita część wartości polecenia

np. **ON M1 GOTO 100,200,300**

‘ jeśli wartość M1=1,to skocz do linii 100

‘ jeśli M1=2, skocz do 200,a jeśli M1=3,to do 300,

‘ jeśli wartości są inne,przejdź do kolejnego kroku programu.

INSTRUKCJE STERUJĄCE PROGRAMEM

IF... THEN.. ELSE.. - jeśli podany po „ IF „ warunek jest spełniony, to instrukcja wykonuje polecenie zapisane po „THEN „, jeśli zaś warunek ten nie jest spełniony, to wykonuje to, co zapisano po „ELSE „

np. **IF M1=1 THEN 100 ELSE 200** ‘jeśli M1=1, to program skacze do linii 100, w przeciwnym razie „skacze” do linii 200,

!!! cała instrukcja musi być zapisana w jednej linii

INSTRUKCJE STERUJĄCE PROGRAMEM

CASE 6 TO 9 'wartości od 6 do 9, więc wykonywane instrukcje 'CASE 6 TO 9

.....

BREAK

DEFAULT 'gdy wartość nieznana , to wykonanie tylko DEFAULT

.....

BREAK

END SELECT 'kończy polecenie SELECT CASE

WAIT - czeka , aż zmienna osiągnie wskazaną wartość

np.

WAIT M_IN (1) = 1 'czeka,aż na wejściu będzie sygnał bit 1' i zaczyna działać

Instrukcje sterujące programem

IF...THEN - jeśli podany po „ IF „ warunek jest spełniony, to

.....

jest to,co zapisane w liniach między IF...THEN a ELSE, jeśli zaś ELSE warunek ten nie jest spełniony,to program wykonuje to,co zapisano

.....

w liniach między komendami ELSE i END IF. !!!Instrukcja ta END IF jest zapisywana w kilku liniach!!!

np. IF M1=1 THEN

‘jeśli M1= 1, to wykonywane są instrukcje :

M2=1

‘ M2 = 1 i M3 = 2 , jeśli natomiast M1

M3=2

‘jest różne od podanej w warunku wartości 1,

ELSE

‘to wykonywane są instrukcje : M2 = - 1

M2= - 1

‘oraz M3 = - 2 .

M3= - 2

END IF

Instrukcje sterujące programem

SELECT

- ta instrukcja skacze zgodnie ze wskazaną zmienną

CASE

i warunkiem , wartość warunku może być wskazana

END SELECT przypadkowo .

np. SELECT M1

//wybór zmiennej M1,której będą

CASE 10

//przyporządkowywane różne wartości.

.....

//jeśli wartość 10 , to wykonywana jest tylko

BREAK

//instrukcja CASE 10

CASE IS 11

// wartość 11,to wykonanie tylko w środku

.....

// instrukcji CASE 11

BREAK

CASE IS <5

// wartości mniejsze od 5, to wykonanie

.....

// instrukcji tylko CASE <5

BREAK

POWTÓRZENIA

FOR ... NEXT - powtórz wszystko między komendami FOR i NEXT,
aż

wskazany warunek będzie satysfakcjonujący

np. FOR M 1 = 1 TO 10

‘ powtarzaj między komendami FOR i NEXT

.....

‘ wszystko 10 razy od wartości początkowej M1=1

NEXT

‘ za każdym razem dodając 1

WHILE ... WEND - powtórz to,co między komendami,gdy warunki są
spełnione

np. WHILE (M1>=1) AND (M1<=10) ‘powtarzaj to,co zapisane
między

.....

‘komendami WHILE i WEND wtedy,

WEND

‘gdy M1 należy do przedziału <1,10>.

Instrukcje sterujące programem

PRZERWANIA

DEF ACT - definiuje przerwanie warunku i proces generacji przerwania

np. DEF ACT 3. M_IN (12) = 1 GOSUB 300. S

‘ jeśli wejściowy sygnał- bit 12 jest włączony dla przerwania numer 3,

‘ to podprogram na linii 300 jest definiowany, robot zmniejsza prędkość
i

‘ zatrzymuje się w najkrótszym czasie i najbliższej odległości jak to
tylko

‘ możliwe.

ACT - wyznacza priorytet przerwania

np. ACT 1= 1

‘ możliwe pierwszeństwo przerwania nr 1

ACT 2= 0

‘ niemożliwe pierwszeństwo przerwania nr 2

RETURN - jeśli program wywołuje przerwanie procesu , to zwraca
przerwanie linii źródłowej

np. RETURN 0 ‘ wraca do linii, gdzie zdarzyło się przerwanie

RETURN 1 ‘ wraca do linii następnej po tej z przerwaniem

Instrukcje sterujące programem

PODPROGRAM

GOSUB - wywołuje podprogram ze wskazanej linii lub wskazanego poziomu
np. GOSUB 100

‘wywołuje procedurę z linii 100

GOSUB *GET ‘ wywołuje procedurę z poziomu GET

ON ... GOSUB - wywołuje podprogram zgodnie ze wskazaniem numeru zmiennej, wartość warunku stanowi część całkowita wartości polecenia.
np.

ON M1 GOSUB 100 , 200 , 300

‘jeśli wartość liczbową zmiennej M1 jest 1, to wywołuje podprogram z linii 100,

‘jeśli ta wartość wynosi 2, to podprogram z linii 200, a jeśli wartością jest 3, to

‘ podprogram z linii 300, gdy każda inna wartość , to program przeskakuje do

‘ następnego kroku

Instrukcje sterujące programem

RETURN - wraca do następnej linii po linii wywołanej komendą

GOSUB

CALLP -wywołuje wskazany program,następna linia w programie źródłowym jest zwracana poleceniem END w wywołanym programie np.

CALLP " 10"

‘wywołuje program nr 10

CALLP " 20", M1, P1

‘przenosi zmienną liczbową M1 i zmienną pozycji P1

‘do programu nr 20 i wywołuje ten program

FPRM - argument jest przenoszony przez program wywołany komendą

CALLP

np.

FPRM M 10 , P 10 ‘przyjmuje liczbową zmienną w M 10 przenoszoną

‘przez CALLP podprogramu i zmienną pozycji w P 10

Instrukcje sterujące programem

ZEGAR

DLY - funkcjonuje jako wskazany czas zegara

np. DLY 0.05

‘ czeka tylko przez 0.05 sekundy

M_ OUT (10) = 1 DLY 0.5 ‘włącza wyjściowy sygnał bit
10

tylko przez 0.5 sek.

Instrukcje sterujące programem

ZATRZYMANIE

HLT - zatrzymuje robota i wstrzymuje wykonywanie programu, gdy program włączymy, to wykonuje od następnej linii

END - definiuje koniec jednego cyklu programu. W nieprzerwanej operacji, program jest wykonywany znowu z linii startowej, dalej wykonywanie instrukcji END. W cyklu operacji, program kończy dalej

wykonywanie instrukcji END , gdy cykl jest zatrzymany .

np. END

‘kończy program nawet w środku wykonywania

IF M_IN(20) =1 THEN HLT‘zatrzymuje program gdy wejściowy sygnał bit 20 jest włączony

MOV P1 WTHIF M_IN(18) =1 HLT ‘zatrzymuje program ,gdy wejściowy sygnał bit 18 jest włączony,
gdy ruch w kierunku P1

Komunikacja

COM ON - możliwe przerwanie procesu z linii komunikacyjnej

np.COM (1) ON 'możliwe przerwanie z komunikacyjnej linii COM1

COM OFF - niemożliwe przerwanie procesu z linii komunikacyjnej ,
przerwanie będzie unieważnione nawet , jeśli się ono wydarzy

np. COM (2) OFF 'niemożliwe (zabronione) przerwanie z linii
komunikacyjnej COM 2

COM STOP - wstrzymuje przerwanie procesu z linii komunikacyjnej
jeśli tam jest przerwanie, jest ono zapamiętane , i jest
możliwe dalsze wykonanie

np.COM (1) STOP 'zatrzymuje przerwanie z linii komunikacyjnej COM
1

FUNKCJE

VAL - zamienia rodzaj w wartość liczbowa

ATN (<liczba>) - liczy arc tangens wartości od $-\pi / 2$ do $+\pi / 2$

ATN 2 (<liczba> <liczba>) - liczy arc tangens (z przyrostów ΔX , ΔY)
wartości od $-\pi$ do $+\pi$

COS (<liczba>) - liczy cosinus

SIN (<liczba>) - liczy sinus

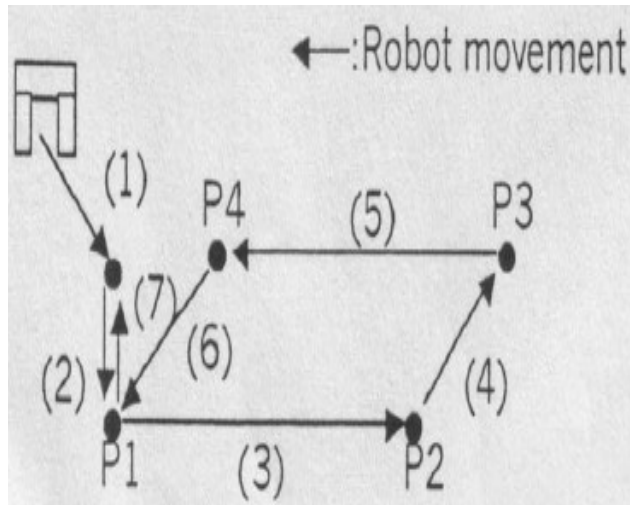
TAN (<liczba>) - liczy tangens

BIN\$ (<liczba>) - zamienia liczbowa wartość na system binarny

HEX\$(<liczba>)- zamienia liczbę na system szesnastkowy

MIRROR\$ - realizowane jest lustro przeciwne dwójkowego systemu bitu

MKI\$ (<liczba>) - zamienia liczbowa wartość wyrażenia na rodzaj 2-bajtowy



PROGRAM

10 MOV P1, -50

20 OVRD 50

30 MVS P1

40 PREC ON

50 MVS P2

60 MVS P3

70 MVS P4

80 MVS P1

90 PREC OFF

100 MVS P1,
-50

1

10 END

OBJAŚNIENIE

'(1)Ruch z łączną interpolacją do punktu oddalonego o 50 mm od punktu P1 w kierunku ręki. Ustawiona szybkość ruchu wynosi 50% max.

'(2)Ruch z interpolacją liniową do P1. Włącza tryb dokładny wykonania zadania.

'(3)Ruch z interpolacją liniową do P1

'(4)Ruch z interpolacją liniową do P1

'(5)Ruch z interpolacją liniową do P1

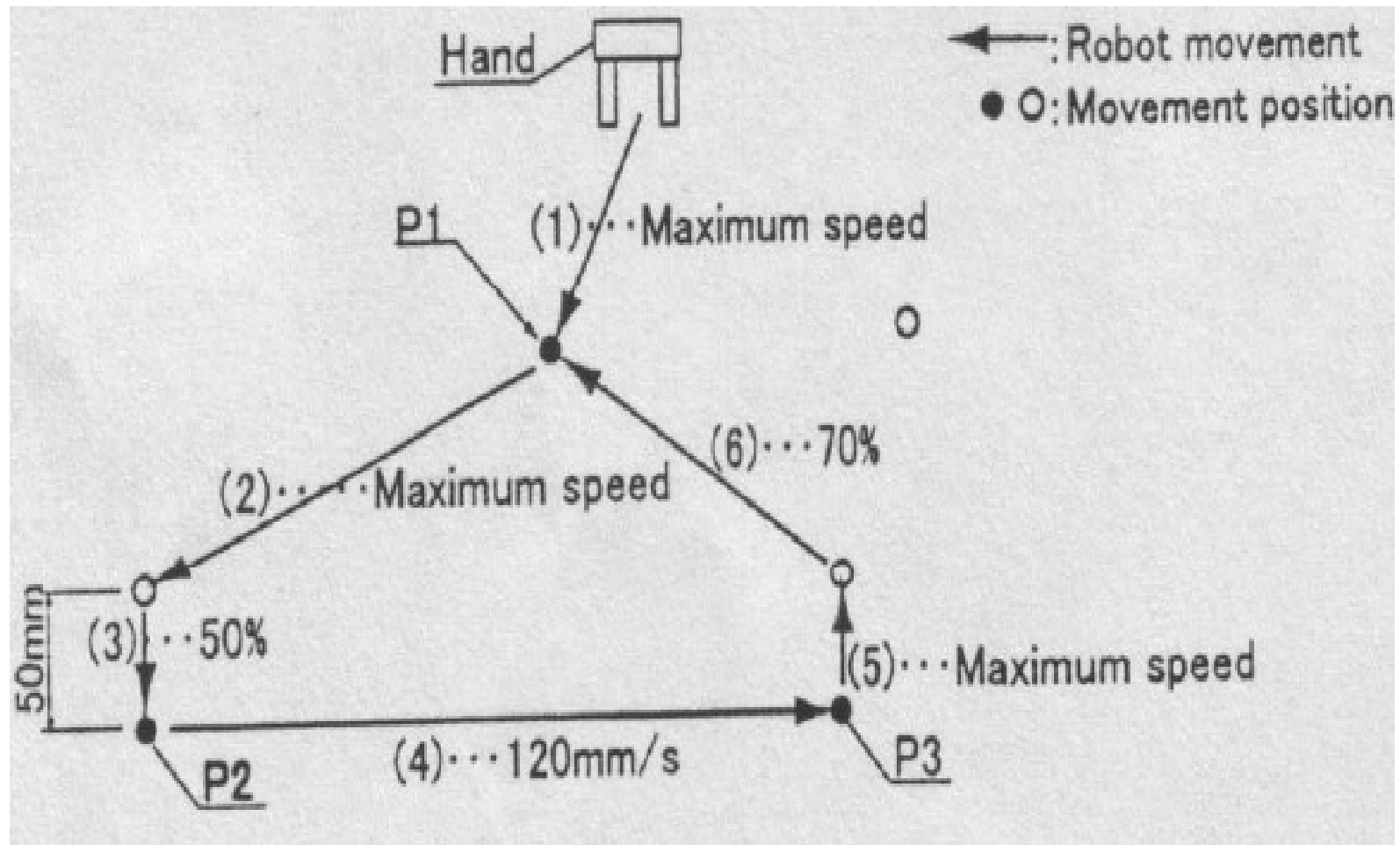
'(6)Ruch z interpolacją liniową do P1

Wyłącza tryb dokładny wykonania zadania.

'(7)Ruch z łączną interpolacją od punktu P1 do pozycji oddalonej 50 mm w kierunku ręki (cofa 50 mm).

Koniec programu

10 OVRD 100	Ustawiona szybkość ruchu wynosi 100% maksymalnej prędkości.
20 MVS P1	'(1)Ruch z interpolacją liniową do P1 z maksymalną szybkością.
30 MVS P2, -50	'(2))Ruch z liniową interpolacją do punktu oddalonego o 50 mm od punktu P2 w kierunku ręki.
40 OVRD 50	Ustawiona szybkość ruchu wynosi 50% maksymalnej prędkości.
50 MVS P2	'(3)Ruch z interpolacją liniową do P2 z prędkością równą połowie maksymalnej.
60 SPD 120	Szybkość końcówki ramienia wynosi 120 mm/s
70 OVRD 100	Ustawia w procentach szybkości ruchu do 100% by otrzymać bezwzględną szybkość końca 120 mm / s
80 ACCEL 70, 70	Ustawia przyśpieszenie do 70% i zmniejszenie prędkości do 70% maksymalnej szybkości.
90 MVS P3	'(4)Ruch z interpolacją liniową do P3 z prędkością równą 120 mm/s.
100 SPD M_NSPD	Powrót szybkości końca do wartości standardowej Ustawiona szybkość ruchu interpolacji na 70% szybkości maksymalnej
110 JOVRD 70	Powrót do maksymalnej szybkości
120 ACCEL	'(5) Ruch z interpolacją liniową do pozycji oddalonej 50 mm w kierunku ręki (cofa 50 mm) z maksymalną prędkością.
130 MVS, -50	'(6)Ruch z interpolacją liniową do P3 z prędkością równą 70% maksymalnej .
140 MVS P1	Koniec programu
150 END	



KONIEC