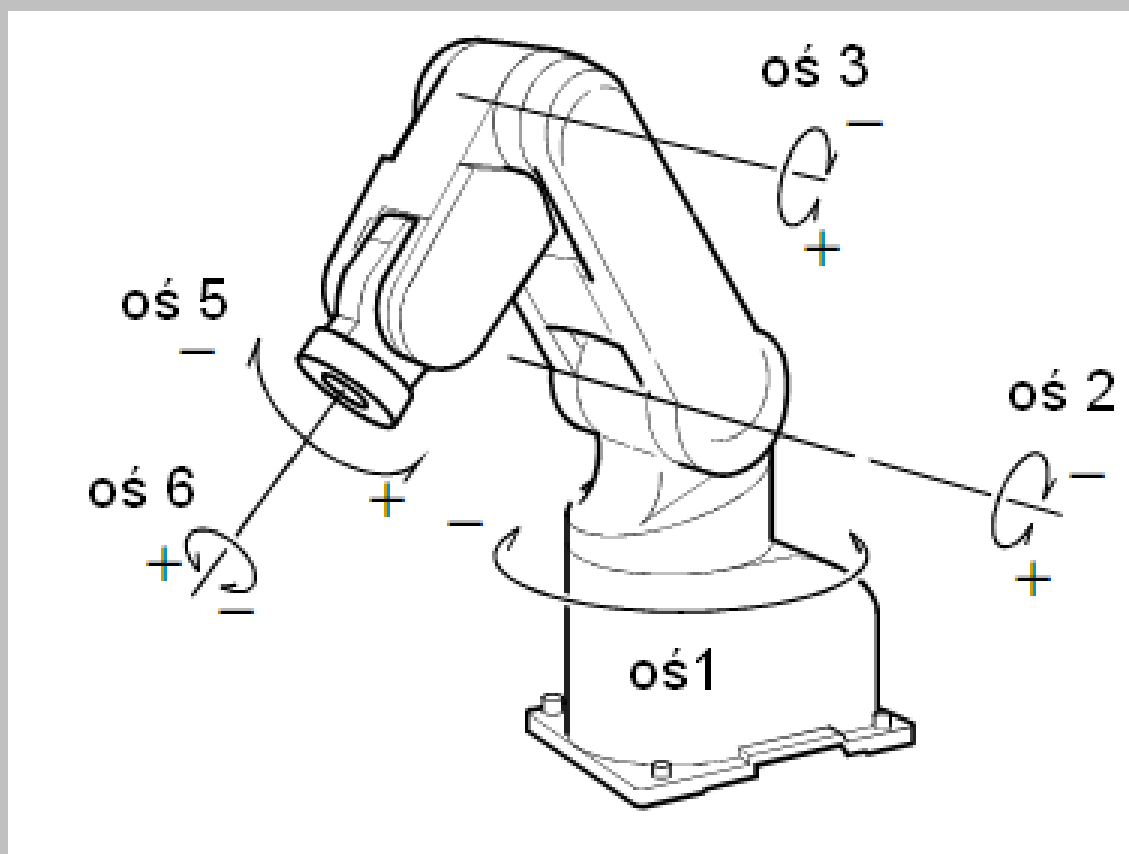


PODSTAWY PROGRAMOWANIA

Robot Mitsubishi RV-2AJ

Model robota Mitsubishi RV-2AJ posiada pięć przegubów cylindrycznych napędzanych serwo-napędami prądu przemiennego. Moment napędowy z silników przenoszony jest za pomocą pasków zębatych i przekładni zębatych. Robot może być mocowany w położeniu „pionowym do góry”, czyli ogniwa znajdują się powyżej podstawy, lub może być podwieszany.



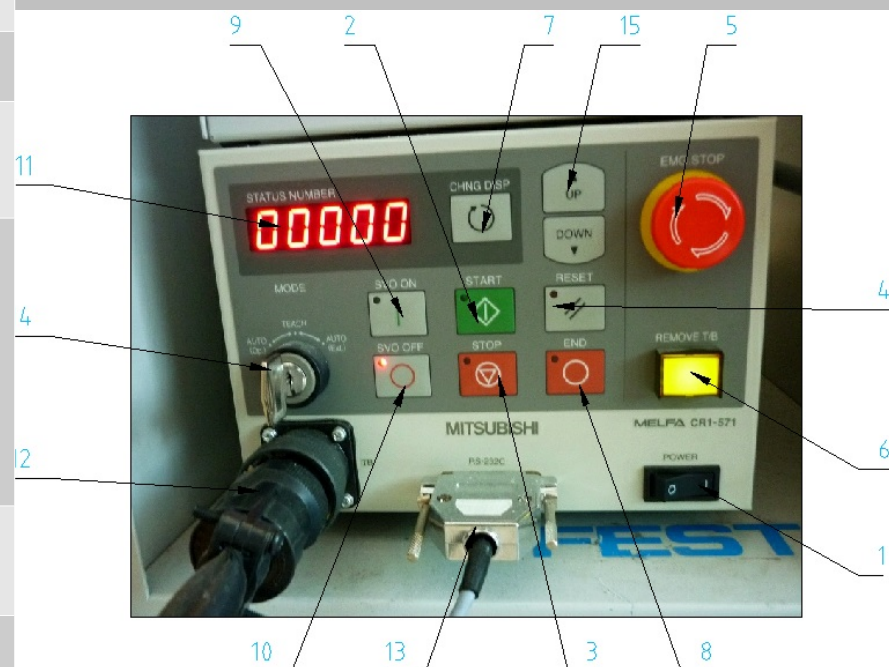
Parametry robota Mitsubishi RV-2AJ

Parametr		Jednostki	Wartości
Liczba stopni swobody			5
Napęd			serwo-napęd AC (J1÷J3:50W, J5 i J6:15W)
Detekcja pozycji			enkodery absolutne
Zakresy obrotów przegubów	J1	stopnie	300 (od -150 do 150)
	J2		180 (od -60 do 120)
	J3		230 (od -110 do 120)
	J5		180 (od -90 do 90)
	J6		400 (od -200 do 200)
Prędkość kątowna przegubów	J1	stopnie/s	180
	J2		90
	J3		135
	J5		180
	J6		210
Powtarzalność		mm	0,02
Maksymalne obciążenie		kg	1,5
Masa całkowita		kg	ok. 17

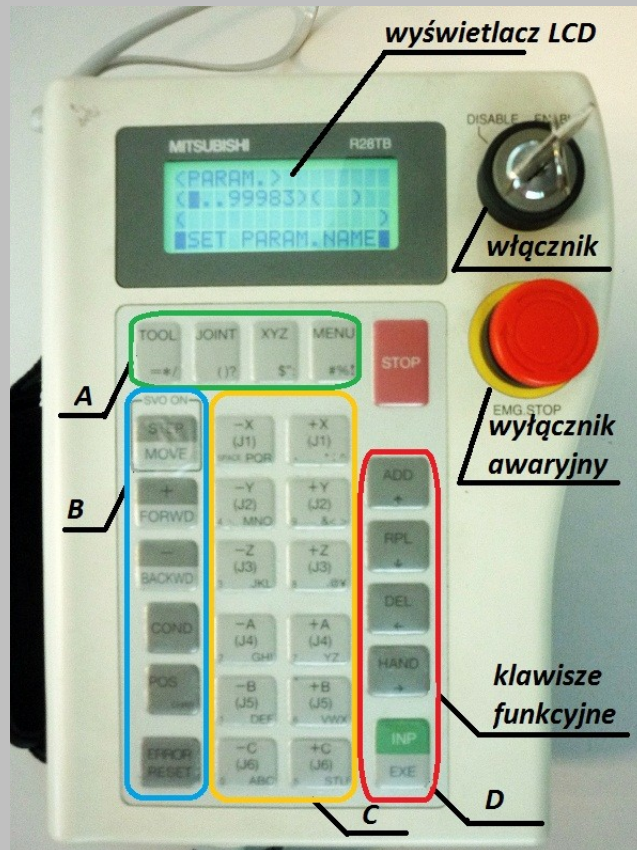
Jednostka sterująca CR1

Edycję programów można z wykorzystaniem wykonywać teach box-a lub na komputerze PC z zainstalowanym odpowiednim programem. Komunikacja jednostki sterującej z urządzeniami odbywa się za pomocą interfejsów znajdujących się na panelu czołowym. Jednostka dodatkowo jest wyposażona w wyjść cyfrowych ogólnego przeznaczenia 16 wejść .

1		Włącznik zasilania jednostki sterującej
2	START	Przycisk startu programu
3	STOP	Natychmiastowe zatrzymanie robota (nie powoduje wyłączenia serwo-napędów)
4	RESET	Kasowanie statusu występowania błędu
5	EMG. STOP	Awaryjne zatrzymanie pracy robota (powoduje wyłączenie serwonapędów)
6	REMOVE T/B	Przycisk wykorzystywany przy podłączaniu i rozłączaniu teach box-a bez wyłączania zasilania Stosuje się go w celu zmiany rodzaju i sposobu wyświetlania komunikatów systemowych
7	CHNGDIS P	
8	END	Zatrzymuje wykonywany program po wykonaniu ostatniej linii albo napotkaniu funkcji END
9	SVO. ON	Załącza zasilanie do serwo-napędów
10	SVO.OFF	Wyłącza zasilanie serwo-napędów



Teach box

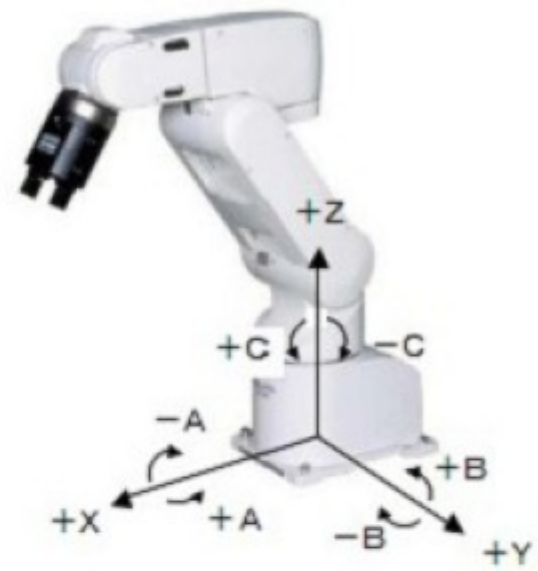


Podczas edycji programu, prócz algorytmu wykonywanych operacji, należy stworzyć pozycji listę w których będzie znajdował się robot. Przebywanie w obszarze roboczej przestrzeni robota jest groźne dla zdrowia człowieka. Dlatego też w teach box-ie się znajduje trójpozycyjny przycisk dead man, które zadaniem jest zwiększenie bezpieczeństwa osób tworzących program. Sterowanie robotem w trybie ręcznym możliwe jest tylko i wyłącznie wtedy gdy klawisz dead man jest delikatnie wciśnięty. W sytuacji gdy przycisk jest w pozycji wciśnięty lub swobodnej z dużą siłą, robot zatrzymuje się, sterowanie nim jest niemożliwe. Wykonywanie ruchów ręcznych robotem przy użyciu teach box-a wymaga wciśnięcia połączenie dwóch klawiszy: STEP/MOVE oraz klawisza z grupy C.

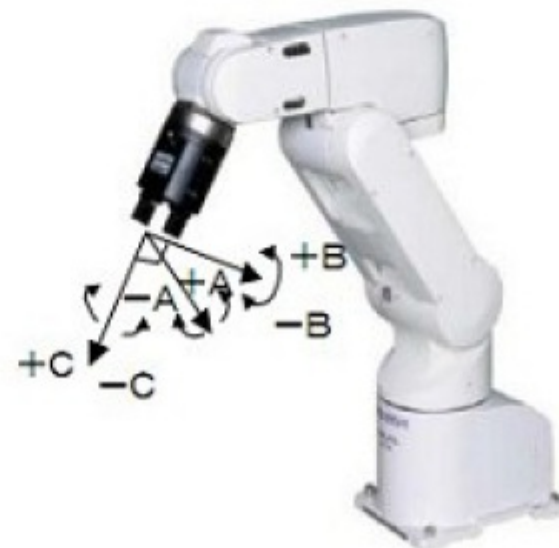
JOINT



XYZ



TÓOL



Język programowania MELFA BASIC IV

Roboty Mitsubishi są sterowane za pomocą języka programowania MELFA BASIC. Język ten jest oparty na standardzie BASIC, dzięki czemu jest bardzo łatwy do nauki. Oprócz instrukcji i konstrukcji, wywodzących się ze standardu BASIC, takich jak FOR...NEXT, GOTO, dostępne są instrukcje specjalizowane związane z:

- obsługą dodatkowych typów danych,
- wykonywaniem sekwencji ruchów manipulatora
- obsługą chwytaka
- obsługą wejść i wyjść cyfrowych

Język programowania MELFA BASIC IV

... jest specjalizowanym narzędziem służącym do programowania robotów i jego zasadniczą część stanowią funkcje definiujące trajektorie ruchu chwytaka. Zbiór tych funkcji można podzielić na kategorie związane z:

- ruchem po krzywej,
- ruchem po linii prostej,
- ruchem po okręgu,
- ciągłym lub krokowym przemieszczeniem,
- definiowaniem parametrów ruchu (prędkość, przyspieszenie, dokładnością z jaką robot osiąga zdefiniowane punkty),
- potwierdzeniem osiągnięcia zadanego położenia,
- operacją paletyzacji.

W języku MELFA-BASIC IV programista może korzystać z instrukcji skoku bezwarunkowego (GOTO) lub warunkowego (ON GOTO), funkcji warunkowych (IF THEN ELSE, SELECT CASE) oraz stosować pętle (FOR NEXT, WHILE WEND).

Kolejna grupa funkcji jest związana z wielozadaniowym trybem pracy kontrolera. Polega on na tym, że jednocześnie można uruchomić kilka programów (do 8 w wersji standardowej) a szybkość ich wykonywania zależy od przydzielonego im priorytetu. Główną zaletą wielozadaniowej pracy jest skrócenie czasu wykonania zaprogramowanego zadania dzięki temu że instrukcje sterujące robotem i urządzeniami zewnętrznymi wykonywane są jednocześnie.

W języku MELFA-BASIC program to zbiór linii o określonej strukturze. Każda linia rozpoczyna się od numeru w przedziale od 1 do 32676.

Linie muszą być numerowane w kolejności rosnącej z dowolnym krokiem. Nie zachowanie tej zasady spowoduje błąd podczas kompilacji programu.

W każdej linii może znajdować się tylko jedna:

- komenda'
- komenda rozszerzona
- etykieta.

10 <i>nr linii</i>	MOV <i>funkcja ruchu chwytaka z aktualnej pozycji do punktu określonego w argumencie</i>	P1 <i>argument funkcji (współrzędne punktu)</i>
-----------------------	-----------------------------------------------------------------------------------------------------	------------------------------------------------------------

Komenda rozszerzona daje możliwość wykonania dodatkowej czynności w tym samym czasie gdy robot wykonuje ruch.

10 <i>nr linii</i>	MOV P1 <i>komenda</i>	WTH <i>funkcja tworząca komendy rozszerzone</i>	M_OUT(5)=1 <i>operacja przypisania-wyjście cyfrowe kontrolera będzie w stanie wysokim tak długo jak będzie trwał ruch robota</i>
-----------------------	--------------------------	--------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------

10 <i>nr linii</i>	MOV P1 <i>komenda</i>	WTHIF <i>funkcja tworząca komendy rozszerzone warunkowe</i>	M_IN(5)=1, HLT <i>gdy wejście cyfrowe nr 5 znajdzie się w stanie wysokim, nastąpi zatrzymanie ruchu robota</i>
-----------------------	--------------------------	------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------

Język MELFA-BASIC IV umożliwia wykonanie operacji skoków bezwarunkowych, warunkowych oraz pętli. W tym celu w programie należy umieścić informacje o miejscu do którego zostanie wykonany skok.

Do tego celu służy etykieta *NAZWA

Przykład:\

```
10 GOTO *ETYKIETA  
200 *ETYKIETA
```

Znaki mające specjalne znaczenie

Podkreślenie (`_`) użyte jako drugi znak w nazwie zmiennej wskazuje, że zadeklarowana przez użytkownika zmienna ma zasięg globalny, czyli jej wartość jest dostępna w więcej niż jednym programie.

Apostrof (`'`) służy do wstawiania komentarzy do programu. Wszystkie znaki umieszczone za apostrofem nie są traktowane przez kompilator jako komendy, więc można wstawić tam dowolny ciąg znaków.

Gwiazdka (`*`) stosowana jest do wstawiania etykiet.

Typy zmiennych

Zmienne, w zależności od obszaru w jakim mogą być użyte, dzielą się na dwie kategorie:

- lokalne – można wykonywać na nich operacje w zakresie tylko tego programu w którym zostały zadeklarowane. W wielu programach mogą powtarzać się nazwy zmiennych lokalnych lecz odnoszą się one do innych komórek pamięci kontrolera,
- globalne (zewnętrzne) – definiowane są przez użycie znaku podkreślenia () jako drugiego znaku. Można się do nich odwoływać z dowolnego programu. Zmienne globalne pozwalają na przekazywanie danych między programami.

Zmienne globalne - programowe

Nazwa zmiennej globalnej składa się zawsze dużej litery odnoszącej się do typu danych oraz znaku podkreślenia na drugiej pozycji. Pula zmiennych programowych jest z góry określona i ograniczona do 20 elementów dla każdego typu.

Typ zmiennej	Nazwy	
położenia	P_00 ÷ P_20	
złączowa	J_00 ÷ J_20	
liczbowa	M_00 ÷ M_20 (liczby zmiennoprzecinkowe podwójnej precyzji)	
znakowa	C_00 ÷ C_20	

Zmienne typu wejścia/wyjścia

Jednostka sterująca robotów Mitsubishi wyposażona jest w moduł wejść i wyjść cyfrowych, które mogą realizować funkcje sterownika PLC oraz port RS-232 służący do komunikacji z komputerem PC. Obsługa tych urządzeń odbywa się za pomocą predefiniowanych systemowych zmiennych wejścia/wyjścia.

Nazwa zmiennej	Opis
M_IN(x)	Odczytanie bitu sygnału wejściowego o numerze x
M_INB(x)	odczytanie bajtu sygnałów wejściowych zaczynając od bitu x
M_INW(x)	odczytanie słowa (16 bitów) sygnałów wejściowych zaczynając od bitu x
M_OUT(x)	ustawienie bitu sygnału wyjściowego o numerze x
M_OUT(x)	ustawienie bajtu sygnałów wyjściowych zaczynając od bitu x
M_OUT(x)	ustawienie słowa sygnałów wyjściowych zaczynając od bitu x

Jednostka sterująca na bieżąco kontroluje parametry pacy robota i zapisuje wszystkie informacje o jego stanie w specjalnym obszarze pamięci zwanym zmiennymi systemowymi.

P_CURR	Aktualna pozycja	R	położenia
J_CURR	Aktualna pozycja	R	złączowy
P_FBC	Odczyt położenia ze sprzężenia zwrotnego od serwo-napędów	R	położenia
J_FBC	Odczyt kątów przegubów ze sprzężenia zwrotnego od serwo-napędów	R	złączowy
M_CMPDST	Różnica pomiędzy położeniem zadany programowo a aktualną pozycją	R	położenia
J_COLMXL	Różnica między szacowanym momentem napędowym a wartością rzeczywistą	R	złączowy [%]
M_COLSTS	Wykrycie kolizji (zderzenia)	R	liczbowy
P_COLDIR	Kierunek ruchu podczas wystąpienia kolizji	R	położenia
M_OPOVRD	Limit prędkości ustawiony na TB	R	liczbowy [%]
M_OVRD	Limit prędkości ustawiony w programie	R	liczbowy [%]
M_RSPD	Aktualna prędkość	R	liczbowy [mm/s]
M_ACLSTS	Wskaźnik przyspieszenia (0-stop, 1-przyspieszanie, 2 – ruch jednostajny, 3-hamowanie)	R	liczbowy
M_LDFACT	Współczynnik obciążenia serwo-napędu	R	liczbowy
C_PROG	Nazwa aktualnie wykonywanego programu	R	znakowy
M_LINE	Aktualnie wykonywana linia programu	R	liczbowy
M_ERR	Wystąpienie błędu	R	liczbowy
M_ERRLV	Odczytuje poziom błędu (1-ostrzeżenie, 2-niski, 3-wysoki1, 4-wysoki2)	R	liczbowy
M_OUT	Ustawienie bitu sygnału wyjściowego karty rozszerzeń	RW	liczbowy

Sterowanie stanowiskiem roboczym odbywa się z wykorzystaniem wejść cyfrowych kontrolera robota, które połączone są z wszystkimi aktywnymi elementami stanowiska.

IN 1	Panel – przycisk START
IN 2	Panel – przycisk STOP
IN 3	Panel – przycisk RESET
IN 5	Czujnik optyczny 22
IN 7	Czujnik optyczny 21
IN 8	Czujnik optyczny 11

OUT 0	Lampka Start (panel)
OUT 4	Załączenie transportera nr 1
OUT 5	Załączenie transportera nr 2

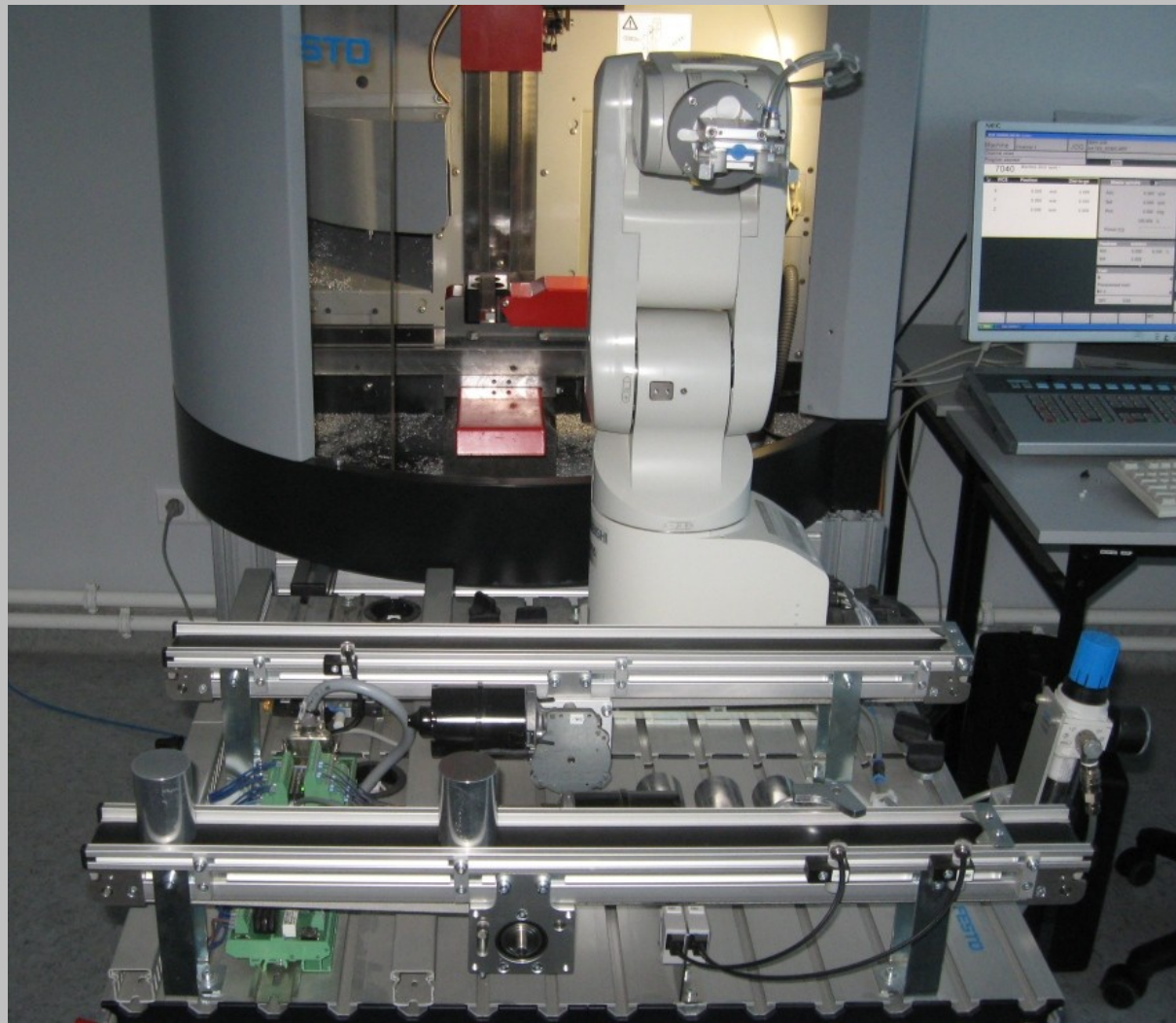
Algorytm programu sterującego

Sterowanie stanowiskiem roboczym odbywa się z wykorzystaniem wejść cyfrowych kontrolera robota, które połączone są z wszystkimi aktywnymi elementami stanowiska. Dla jednoznaczności opisu wyposażenia stanowiska przyjęto następujące nazewnictwo:

- podajnik taśmowy 1 – umieszczony jest w głębi stanowiska
- podajnik taśmowy 2 – umieszczony jest blisko krawędzi stanowiska
- czujnik optyczny XY – indeks X odnosi się do transportera na którym jest zamocowany, Y oznacza kolejny czujnik na tym samym transporterze (licząc zgodnie kierunkiem ruchu taśmy transportującej)

Algorytm programu sterującego

1. sprawdzenie czy obiekt manipulacji znajduje się na podajniku 2 na wysokości czujnika 22
`IF M_IN(5)=1 THEN "FLAGA1"`
1. jeśli czujnik nie wskazuje obecności przedmiotu zostanie załączony podajnik taśmowy
`ELSE M_OUT(5)=1`



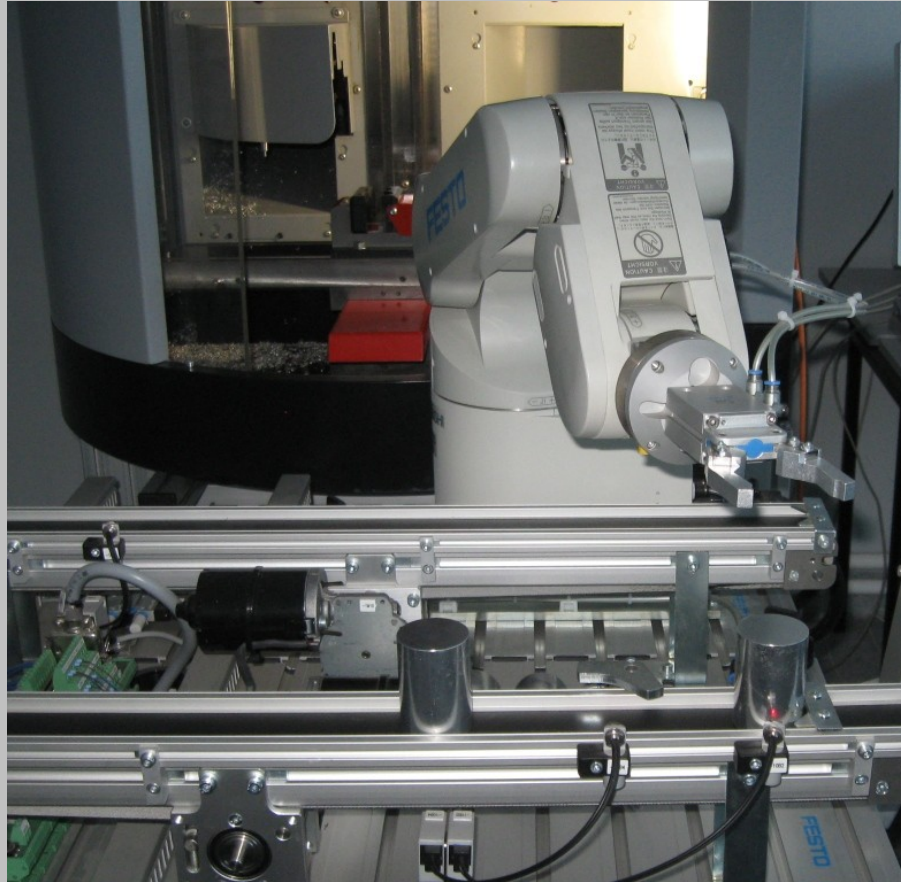
Algorytm programu sterującego

1. ruch podajnika do momentu dojechania przedmiotu do czujnika 22

M_OUT(5)=1

WAIT M_IN(5)=1

M_OUT(5)=0



1. szybki ruch robota w kierunku przedmiotu wskazywanego przez czujnik 22

*OVRD 90

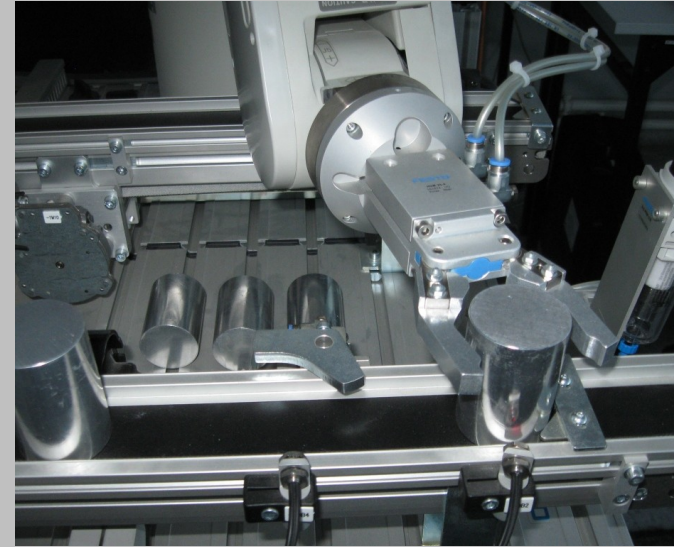
CNT 1, 30

MOV P5

Algorytm programu sterującego

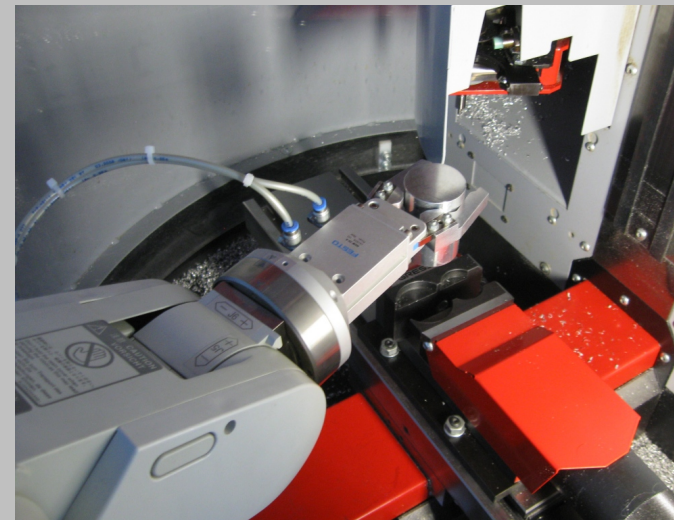
1. powolny ruch robota do przedmiotu manipulacji i zamknięcie chwytaka

OVRD 20
MOV P6
HCLOSE 1



1. umieszczenie przedmiotu w obrabiarce

OVRD 90
CNT 1,30
MOV P7
OVRD 20
MOV P8
CNT 0
HOPEN 1



Algorytm programu sterującego

1. wyjęcie przedmiotu z frezarki
i umieszczenie na podajniku nr 1

OVRD 50

CNT 1,30

MOV P10

OVRD 20

MOV P11

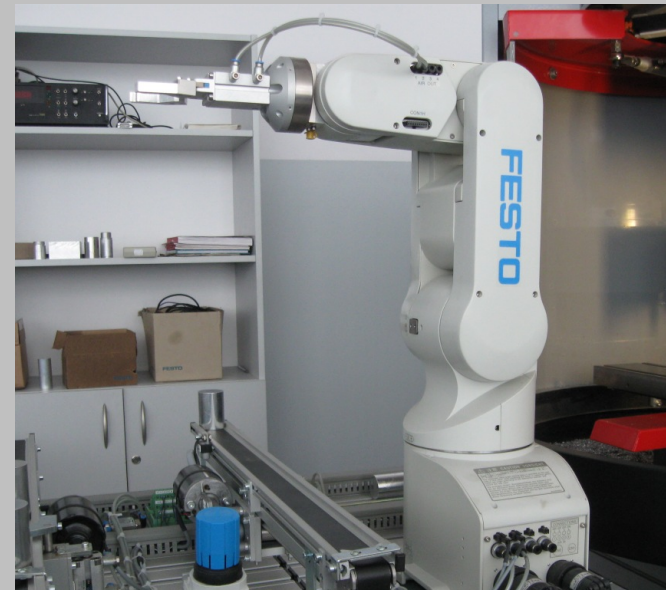
HOPEN 1



1. odjazd robota do pozycji neutralnej

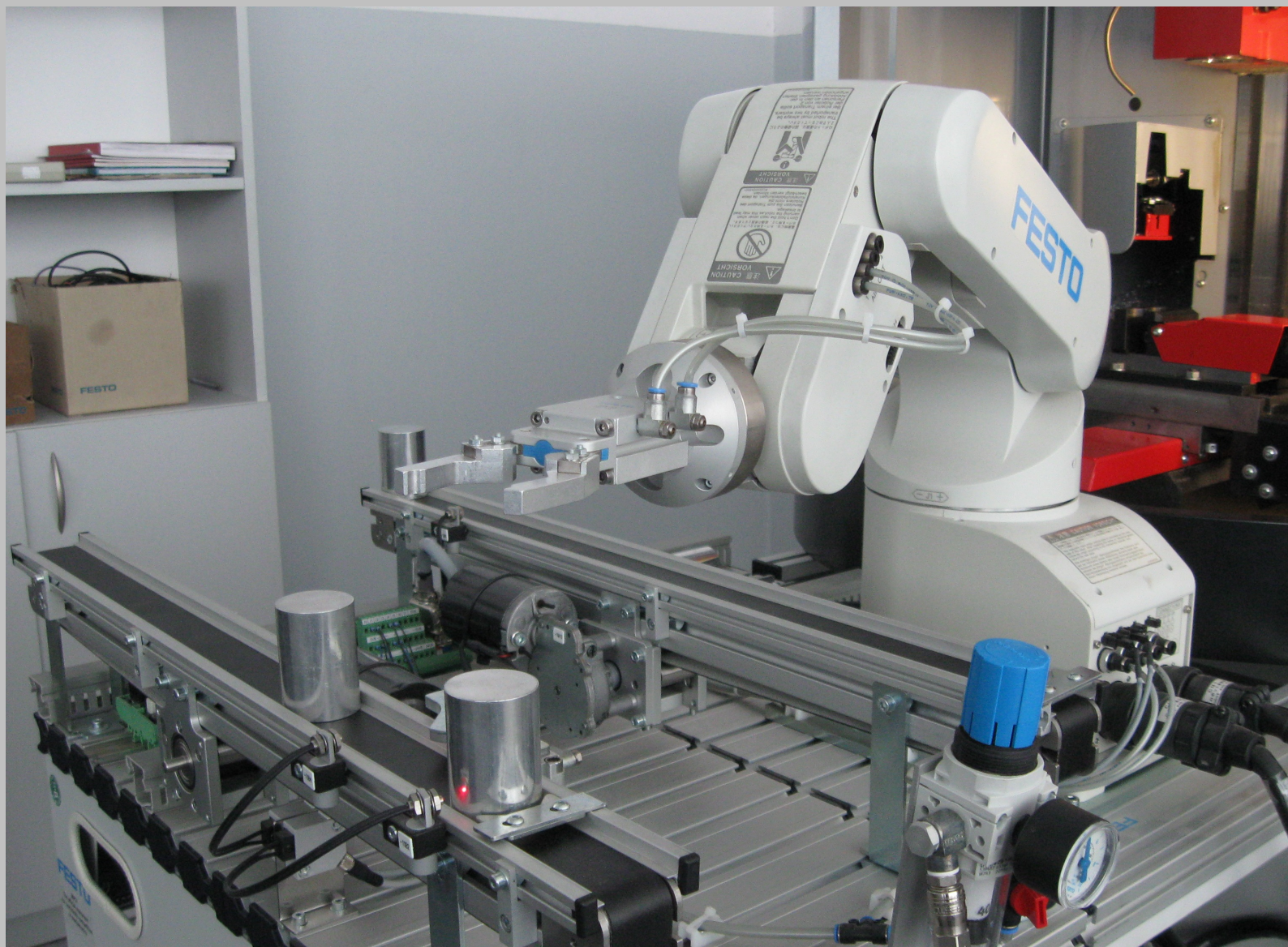
OVRD 90

MOV P9

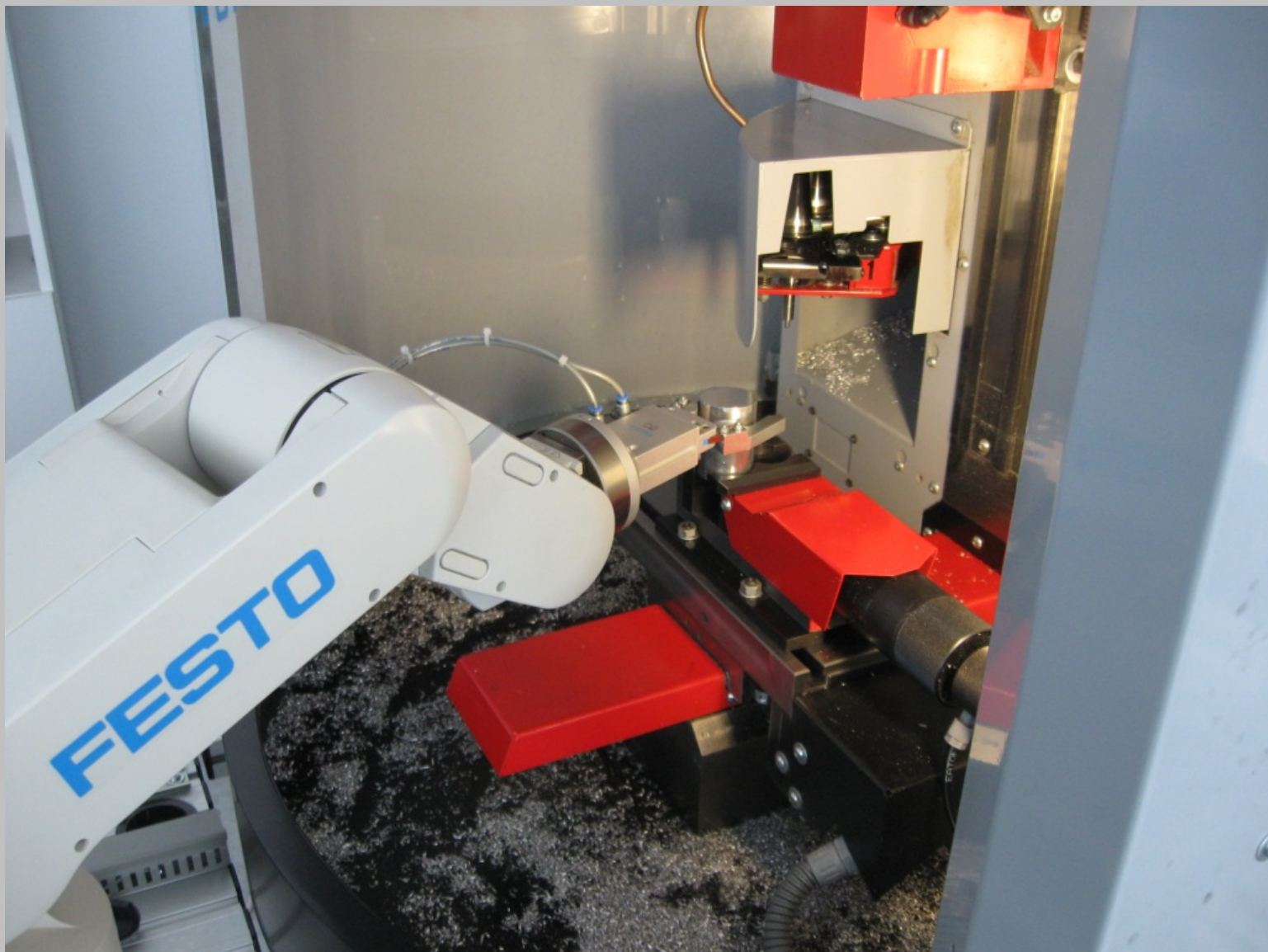


Algorytm programu sterującego

1. załączenie transportera 1 i oczekiwanie na sygnał z czujnika 11, który będzie oznaczał, że przedmiot po obróbce opuszcza stanowisko
 M_OUT(4)=1
 WAIT M_IN(8)=1
1. wyłączenie transportera 1 po upływie 5 sekund od zadziałania czujnika 11
 DLY 5
 M_OUT(4)=0
1. zwiększenie o 1 zmiennej globalnej M_01 określającej ilość wykonanych cykli obróbki
 M_01=M_01+1
1. start kolejnego cyklu pracy stanowiska jeśli liczba wykonanych operacji jest mniejsza od określonej w programie
 IF M_01<10 THEN GOSUB *START



Dojazd robota do przedmiotu przeznaczonych do obróbki



Umieszczenie przedmiotu w obrabiarce