

SPIS TREŚCI

1	JĘZYK AS.....	2
1.1	WSTĘP.....	2
2	CHARAKTERYSTYKA JĘZYKA AS.....	3
3	WYRAŻENIA W JĘZYKU AS	3
3.1	NOTACJA I KONWENCJE	3
4	DANE POZYCJI, DANE NUMERYCZNE I TEKSTOWE	5
4.1	DANE POZYCJI.....	5
4.2	DANE NUMERYCZNE.....	7
5	ZMIENNE.....	8
5.1	ZMIENNE.....	8
5.2	NAZWY ZMIENNYCH.....	8
5.3	DEFINIOWANIE ZMIENNYCH POZYCJI	8
5.4	DEFINIOWANIE ZA POMOCĄ POLECEŃ EKRAŃOWYCH	9
6	WYRAŻENIA NUMERYCZNE	9
6.1	OPERATORY	9
7	RUCH ROBOTA.....	10
8	FUNKCJE JĘZYKA AS DOSTĘPNE W ASTORINO	10
8.1	POLECENIA DO STEROWANIA PROGRAMEM	10
8.2	POLECENIA OBSŁUGI ZMIENNYCH POZYCJI.....	13
8.3	POLECENIA DO STEROWANIA SYSTEMEM	14
8.4	POLECENIA DLA SYGNAŁÓW BINARNYCH.....	15
8.5	POLECENIA DO WYŚWIETLANIA KOMUNIKATÓW	16
9	INSTRUKCJE PROGRAMU.....	17
9.1	INSTRUKCJE RUCHU.....	18
9.2	INSTRUKCJE STERUJĄCE PRĘDKOŚCIĄ I DOKŁADNOŚCIĄ	23
9.3	INSTRUKCJE STERUJĄCE PROGRAMEM	25
9.4	INSTRUKCJE STRUKTURY PROGRAMU.....	25
9.5	INSTRUKCJE DLA SYGNAŁÓW BINARNYCH.....	31
10	FUNKCJE.....	32
10.1	FUNKCJE OPERUJĄCE NA WARTOŚCIACH RZECZYWISTYCH.....	32
10.2	FUNKCJE POZYCJI.....	33

10.3	FUNKCJE MATEMATYCZNE	35
11	PRZYKŁADOWE PROGRAMY	35
11.1	KONFIGURACJA POCZĄTKOWA PROGRAMÓW	35
11.2	PALETYZACJA	36

1 JĘZYK AS

1.1 WSTĘP

Niniejsza instrukcja opisuje język AS* stosowany w kontrolerach robotów ASTORINO. Celem niniejszej instrukcji jest dostarczenie szczegółowych informacji o całym języku AS, jego podstawowych zastosowaniach, typach danych, sterowaniu trajektorią robota oraz wszystkich poleceniach i instrukcjach pozwalających na efektywne korzystanie z tego języka. Instrukcja ta nie zawiera procedur obsługi robota, które zamieszczone są w Instrukcji obsługi. Należy dokładnie zapoznać się z zawartością niniejszej instrukcji jak i pozostałych instrukcji wymienionych poniżej.

Korzystanie z robota dozwolone jest wyłącznie po dokładnym przeczytaniu i zrozumieniu instrukcji.

W treści niniejszej instrukcji założono, że robot został zamontowany oraz podłączony zgodnie z wymogami zamieszczonymi w wymienionych w instrukcji obsługi. W przypadku jakichkolwiek pytań lub wątpliwości dotyczących eksploatacji robota, prosimy skontaktować się z firmą Astor.

Wersja języka AS stosowana w robotach ASTORINO jest wersją uproszczoną pełnego języka AS stosowanego w robotach Kawasaki Robotics firmy Kawasaki Heavy Industries Ltd.

Podstawowe różnice to:

- brak opcjonalnych wyrażeń w funkcjach
- zmienne liczbowe mogą składać się tylko z liter (A-Z)
- brak zmiennych tablicowych
- brak zmiennych typu string

Uwaga* AS należy wymawiać jako [az].

1. Niniejsza instrukcja nie może być traktowana jako gwarancja na systemy, w których używane są roboty. Firma Astor nie ponosi również odpowiedzialności za jakiegokolwiek wypadki, szkody i/lub naruszenia praw autorskich wynikające z korzystania z takich systemów.

2. Zaleca się, aby wszyscy pracownicy odpowiedzialni za uruchamianie, programowanie, serwisowanie lub kontrolę robota, zostali wcześniej przeszkoleni na kursach oferowanych przez firmę Astor.

3. Firma Astor zastrzega sobie prawo do wprowadzania zmian, poprawek i aktualizacji niniejszej instrukcji, bez uprzedniego powiadomienia.

4. Instrukcję należy zachować i przetrzymywać w łatwo dostępnym miejscu. W przypadku zmiany miejsca pracy robota, przenoszenia do innego oddziału lub sprzedaży, należy obowiązkowo dołączyć

do niego niniejszą instrukcję. W przypadku utraty lub zniszczenia niniejszej instrukcji, prosimy skontaktować się z firmą Astor.

Wszelkie prawa zastrzeżone. Copyright © 2022 Astor Sp. z o.o.

2 CHARAKTERYSTYKA JĘZYKA AS

W systemie AS robot jest sterowany i pracuje zgodnie z programem. Program jest wcześniej przygotowywany i opisuje zadania, jakie mają być wykonane. (Metoda polegająca na odtwarzaniu nauczonego programu).

Instrukcje dostępne w języku AS można podzielić na dwie grupy: polecenia konsoli i instrukcje programu.

Polecenia konsoli: służą do zapisu, edycji i wykonywania programów. Są one wprowadzane w terminalu w oprogramowaniu Astorino-RobotController po znaku (>) i natychmiastowo wykonywane.

Niektóre z tych poleceń są również używane w programach jako instrukcje programu.

Instrukcje programu: wykorzystywane w programach do definiowania ruchów robota, monitorowania i sterowania sygnałami zewnętrznymi, itp. Program to zbiór instrukcji programu.

W niniejszej instrukcji polecenia wprowadzane na ekranie są nazywane poleceniami, a instrukcje programu są nazywane instrukcjami.

Uproszczony język AS posiada szereg unikatowych cech:

1. Dwa układy współrzędnych: układ współrzędnych globalnych na podstawie robota oraz układ współrzędnych narzędzia związany z narzędziem zamocowanym na końcu ramienia.

Robot można programować w dowolnym z tych układów współrzędnych.

2. W trybie uczenia lub odtwarzania można przemieszczać robota po torze liniowym. W trybie uczenia może być jednocześnie zachowywana orientacja narzędzia.

3. Nazwy programów mogą być dowolne, a ich liczba jest ograniczona jedynie przez dostępną pamięć robota.

4. Każda sekwencja ruchu może być zdefiniowana jako program.

5. Robota można programować za pomocą komputera osobistego z uruchomionym oprogramowaniem Astorino-RobotController.

3 WYRAŻENIA W JĘZYKU AS

W niniejszym rozdziale opisano rodzaje danych i zmienne używane w uproszczonym języku AS.

3.1 NOTACJA I KONWENCJE

1. Małe i duże znaki

W celu ułatwienia zrozumienia tekstu, w niniejszej instrukcji przyjęto opisane poniżej zasady stosowania małych i dużych liter. Wszystkie słowa kluczowe języka AS (polecenia, instrukcje, itp.) są pisane dużymi literami. Zmienne oraz inne wprowadzane dane są pisane małymi literami. Nie mniej jednak, w czasie wprowadzania z terminala AS wielkość liter nie ma znaczenia.

2. Spacja, tabulator

Pomiędzy parametrem (lub instrukcją) a parametrem musi znajdować się co najmniej jedna spacja lub tabulator*. Spację lub tabulator należy również wstawiać pomiędzy parametry nie oddzielone za pomocą przecinka lub innego znaku ograniczającego. Nadmiarowe spacje lub tabulatory są ignorowane przez system.

Uwaga* Parametr to dane wymagane do prawidłowego wykonania polecenia lub innej funkcji. Przykładowo, dla polecenia SPEED należy podać parametr określający prędkość robota. Jeżeli polecenia lub funkcja korzystają z wielu parametrów, muszą one być oddzielone za pomocą przecinka lub spacji.

Przykład SPEED 50

3. Klawisz ENTER

Większość poleceń lub instrukcji programu jest wykonywanych po wciśnięciu klawisza ENTER. W niniejszej instrukcji klawisz ENTER jest oznaczany symbolem ↵.

Wartości numeryczne

4. Wartości podawane są w systemie dziesiętnym, o ile nie zaznaczono inaczej. Wyrażenia matematyczne służą do wyznaczania wartości parametrów w poleceniach ekranowych oraz instrukcjach programów AS. Nie mniej jednak należy pamiętać o ograniczeniach dotyczących tych wartości. Poniżej zamieszczono zasady interpretacji wartości w różnych kontekstach.

- Odległość

Służy do definiowania długości ruchu robota pomiędzy dwoma punktami. Jednostka odległości to milimetr (mm); jest ona pomijana w czasie wprowadzania. Dozwolone są zarówno wartości ujemne jak i dodatnie.

- Kąty

Opisują orientację osi narzędzia oraz położenie katowe za pomocą odpowiednio 3 kątów obrotu Eulera (Rx – Ry – Rz). Dozwolone są zarówno wartości ujemne jak i dodatnie, a dopuszczalna wartość maksymalna kąta to 180

- Numer osi

Numer osi to wartość całkowita od 1 do liczby dostępnych osi (standardowe robot posiada 6 osi). Osie są numerowane kolejno, począwszy od osi podstawy. (Zwykle są określane jako JT1, JT2,...).

- Numery sygnałów

Numery sygnałów służą do identyfikacji sygnałów binarnych (ON/OFF). Są to wartości całkowite. Dopuszczalne zakresy wartości podaje zamieszczona poniżej tabela.

	Zakres standardowy
Zewnętrzne sygnały wyjściowe	1-8

Zewnętrzne sygnały wejściowe	1001-1008
Sygnały wewnętrzne	2001-2016

Numer sygnału o wartości ujemnej oznacza stan OFF.

- Słowa kluczowe

Ogólnie ujmując, można stosować dowolne nazwy zmiennych w systemie AS. Nie mniej jednak, pewne słowa są zastrzeżone, jak nazwy poleceń, instrukcji, itp. i nie można z nich korzystać do nazywania danych o pozycji, jako nazwy zmiennych, itp.

4 DANE POZYCJI, DANE NUMERYCZNE I TEKSTOWE

W uproszczonym systemie AS dostępne są dwa typy danych: Dane o pozycji oraz numeryczne.

4.1 DANE POZYCJI

Dane pozycji służą do określenia położenia i orientacji robota w przestrzeni roboczej. Położenie i orientacja robota odnoszą się do położenia punktu środkowego narzędzia (TCP) oraz orientacji narzędzia (współrzędne), o ile nie zaznaczono inaczej. Położenie i orientacja robota określają łącznie pozycję robota.

Pozycja określa miejsce w którym znajduje się robot oraz jak jest skierowany, a więc w przypadku podawania instrukcji ruchu określone są obydwie te dane.

1. Robot przemieszcza punkt centralny narzędzia (TCP) do określonego położenia.
2. Układ współrzędnych narzędzia robota jest obracany do określonej orientacji.

Dane o pozycji są określane poprzez podanie wartości przemieszczenia w osiach lub współrzędnych transformacji:

1. Wartości przemieszczeń osi

Pozycja robota jest określana poprzez podanie przemieszczenia liniowego lub kąтового w każdym z układów współrzędnych osi robota. Wartości dla osi kątowych są podawane w stopniach, a wartości dla osi liniowych są podawane w milimetrach. Po wykonaniu podanych ruchów w osiach, położenie i orientacja punktu środkowego narzędzia są jednoznacznie określone.

Przykład Numery osi są oznaczone jako JT1,..., JT6, a wartości przemieszeń podano pod numerami osi.

	JT1	JT2	JT3	JT4	JT5	JT6
#pose=	0.00,	33.00,	-15.00,	0.00,	-40.00,	30

2. Wartości transformacji (X,Y,Z,Rx,Ry,Rz)

Wartości te opisują transformacje współrzędnych względem układu współrzędnych odniesienia. O ile nie zaznaczono inaczej, podawane są wartości transformacji układu współrzędnych narzędzia względem bazowego układu współrzędnych robota. Położenie jest określane za pomocą wartości XYZ względem bazowego układu współrzędnych, a orientacja za pomocą kątów Eulera (Rx-Ry-Rz) narzędzia względem bazowego układu współrzędnych

Przykład X Y Z Rx Ry Rz

Pozycja = 0, 1434, 300, 0, 0, 0

Jeżeli robot posiada więcej niż sześć osi, wraz z wartościami transformacji podawane jest położenie dodatkowej osi.

Przykład X Y Z Rx Ry Rz JT7

Pozycja = 0, 1434, 300, 0, 0, 0, 1000

Korzystanie z wartości przemieszczeń osi oraz współrzędnych transformacji posiada wiąże się z pewnymi zaletami i wadami. Należy wybrać jedną z tych metod, stosownie do potrzeby.

	Wartości przemieszczeń osi	Współrzędne transformacji
Zalety	<ul style="list-style-type: none"> • Duża precyzja odtwarzania oraz brak niejednoznaczności konfiguracji robota w danej pozycji 	<ul style="list-style-type: none"> • Środek układu współrzędnych narzędzia stosowany w trybie odtwarzania nie zmienia się nawet po wymianie narzędzia. (Przesunięcie zerowego układu współrzędnych narzędzia). • Możliwość korzystania z współrzędnych względnych (np. współrzędnych w układzie przedmiotu). • Wygoda przetwarzania, ponieważ dane są podane jako wartości XYZRxRyRz.
Wady	<ul style="list-style-type: none"> • Punkt centralny narzędzia (TCP) ulega zmianie w przypadku zmiany narzędzia (zerowy ukł. wsp. narzędzia pozostaje taki sam). 	<ul style="list-style-type: none"> • Współrzędne ulegają zmianie stosownie do współrzędnych transformacji układu bazowego lub układu narzędzia, a więc wymagane jest pełne śledzenie w celu oceny wpływu

	<ul style="list-style-type: none"> • Brak możliwości korzystania ze współrzędnych względnych (np. współrzędnych przedmiotu) 	<p>jakiegokolwiek zmiany na bezpieczeństwo</p>
<p>Zalecane zastosowania</p>	<ul style="list-style-type: none"> • Definiowanie pozycji początkowej w programie • Ustawianie konfiguracji robota tuż przed lub w pozycji opisanej za pomocą współrzędnych transformacji • Stosowanie dla innych, powszechnie: używamy pozycji 	<ul style="list-style-type: none"> • Opisywanie współrzędnych względnych, przykładowo współrzędnych przedmiotu • Opisywanie pozycji, która ma być zmieniana za pomocą wartości numerycznych i funkcji SHIFT • Opisywanie pozycji, która ma być zmieniona w oparciu o informacje dostarczane z czujnika

4.2 DANE NUMERYCZNE

W systemie AS dozwolone jest stosowanie wartości i wyrażeń numerycznych. Wyrażenie numeryczne to wartość wyrażona za pomocą liczb, zmiennych oraz operatorów i funkcji. Wyrażenia numeryczne używane są nie tylko w obliczeniach matematycznych, ale również jako parametry poleceń konsoli i instrukcji programu.

Przykładowo, dla polecenia DRIVE wymagane jest podanie trzech parametrów, numeru osi, wielkości przemieszczenia oraz prędkości. Parametry te można podać w postaci wartości numerycznych lub wyrażeń, jak pokazano w przykładzie poniżej:

DRIVE 3,45,75 Przesunięcie osi nr 3 o kąt 45, z prędkością 75%

DRIVE joint, (start+30)/2, 75 Jeżeli joint=2, start=30, to oś 2 zostanie przesunięta o +30 z prędkością 75%.

Wartości numeryczne w systemie AS dzielą się na trzy typy:

1. Liczby rzeczywiste

Liczby rzeczywiste mogą być wartościami całkowitymi lub ułamkowymi. Dozwolone są wartości dodatnie i ujemne z przedziału $-3.4 \text{ E}+38$ do $3.4 \text{ E}+38$ (-3.4×10^{38} do 3.4×10^{38}) oraz zero. Liczby rzeczywiste można przedstawiać w notacji naukowej. Symbol E służy do oddzielenia mantysy od wykładnika. Wykładnik może być ujemny (potęga 1/10) lub dodatni (potęga 10).

Przykład $8.5\text{E}3 = 8500$

Wartości rzeczywiste bez części ułamkowej są to wartości całkowite. Dozwolony przedział wartości wynosi od $-16\ 777\ 216$ do $+16\ 777\ 215$. Wartości całkowite są wprowadzane w systemie dziesiętnym

2. Wartości logiczne

Wartości logiczne mogą posiadać tylko dwa stany TRUE i FALSE. Wartość 1.0 jest równoważna stanowi TRUE, a wartość 0 (lub 0.0) jest równoważna stanowi FALSE. TRUE i FALSE są słowami zastrzeżonymi w języku AS.

Wartość logiczna True = TRUE, 1.0

Wartość logiczna False = FALSE, 0.0

5 ZMIENNE

W uproszczonym języku AS można przypisywać nazwy do danych o pozycji, numerycznych. Są to tzw. zmienne.

5.1 ZMIENNE

Zmienne zawierające dane o pozycji, numeryczne są określane odpowiednio jako zmienne pozycji, rzeczywiste. Po zdefiniowaniu zmiennej jest ona zapisywana w pamięci wraz z przypisaną do niej wartością. Można wtedy z takiej zmiennej korzystać w programie.

5.2 NAZWY ZMIENNYCH

Nazwy zmiennych muszą rozpoczynać się od znaku alfabetu i mogą zawierać tylko litery. Dozwolone jest stosowanie zarówno małych jak i dużych znaków. Długość nazwy zmiennej nie może przekraczać piętnastu znaków. W przypadku dłuższych nazw, uwzględnianych jest jedynie pierwszych piętnaście znaków.

Poniżej podano przykłady niedozwolonych nazw zmiennych:

3p2a znak nie może być cyfrą

part#2 Znak "#" nie może być stosowany w środku nazwy zmiennej.

random Słowo zastrzeżone

5.3 DEFINIOWANIE ZMIENNYCH POZYCJI

Zmienne zawierające informacje o pozycji są nazywane zmiennymi pozycji. Zmienna pozycji jest zdefiniowana wyłącznie, jeżeli zostanie podana jej nazwa oraz wartość. Jeżeli nie zostanie podana jej wartość, pozostaje nadal niezdefiniowana, a wykonanie programu z niezdefiniowaną zmienną powoduje wygenerowanie błędu.

W uproszczonej wersji języka AS można stosować dwa typy zmiennych pozycji.

- Zmienne o nazwach P1..P40 oraz #P1..#P40 – zmienne zapisywane w pamięci trwałej kontrolera i nie usuwane po wyłączeniu zasilania
- Zmienne o dowolnej nazwie np. pobranie – zmienne nie zapisywane w pamięci trwałej kontrolera i usuwane po wyłączeniu zasilania.

Zmienne pozycji są bardzo wygodnym rozwiązaniem z wielu powodów:

1. Można wykorzystywać te same dane o pozycji wiele razy, bez uczenia pozycji za każdym razem od nowa.
2. Zdefiniowana zmienna pozycji może być używana w różnych programach.
3. Zdefiniowana zmienna pozycji może być użyta do definiowania innej pozycji.

4. Można wprowadzać bezpośrednio wartości numeryczne w celu określenia pozycji, w miejsce bardzo czasochłonnego uczenia pozycji za pomocą programatora ręcznego.

5. Zmienne pozycji mogą posiadać dowolne nazwy, co zwiększa czytelność programów.

Poniżej przedstawiono sposób definiowania zmiennych pozycji.

5.4 DEFINIOWANIE ZA POMOCĄ POLECEŃ EKRANOWYCH

1. Polecenie HERE zastosowane w terminalu pozwala zapisać aktualną pozycję robota w zmiennej pozycji o podanej nazwie.

Przykład 1 Korzystanie z wartości przemieszczeń osi

Nazwa zmiennej musi rozpoczynać się od znaku #, w celu odróżnienia jej od współrzędnych transformacji. Po poleceniu wyświetlane są wartości przemieszczeń w osiach dla aktualnej pozycji:

```
> HERE #pose ↵
```

Przykład 2 Korzystanie z współrzędnych transformacji

```
>HERE pose ↵
```

- Zastępowanie wartości już zdefiniowanej zmiennej

```
>POINT pose = P1 ↵
```

6 WYRAŻENIA NUMERYCZNE

Wyrażenia numeryczne mogą się składać z cyfr, zmiennych, funkcji oraz innych wyrażen numerycznych połączonych za pomocą operatorów. Wszystkie wyrażenia numeryczne wyznaczone przez system są wartościami rzeczywistymi. Wyrażenia numeryczne można podawać zawsze w miejsce wartości numerycznych. Można je stosować jako parametry poleceń konsoli lub instrukcji. Interpretacja wartości zależy od kontekstu, w którym dane wyrażenie jest używane. Przykładowo, Wyrażenie podane w miejsce wartości logicznej ma wartość logiczną False, jeżeli daje w wyniku 0 lub True, jeżeli daje w wyniku wartość 1.

6.1 OPERATORY

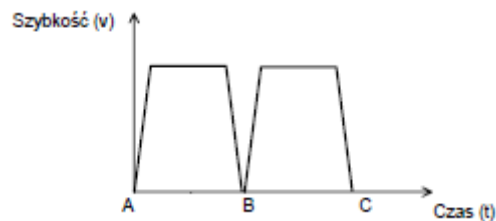
W wyrażeniach można korzystać z operatorów arytmetycznych i logicznych. Wszystkie operatory korzystają z dwóch wartości i zwracają jedną wartość wynikową. Zamieszczona poniżej tabela zawiera zestawienie operatorów.

Operatory arytmetyczne	+ - * / ^ MOD	Dodawanie Odejmowanie lub negacja Mnożenie Dzielenie Potęgowanie Reszta z dzielenia
Operatory porównania	< <=, =<	Mniejszy Mniejszy lub równy

	== <> >=, => >	Równy Różny Większy lub równy Większy
Operatory logiczne	AND NOT OR XOR	Logiczne AND dopełnienie logiczne Logiczne OR Koniunkcja logiczna OR

7 RUCH ROBOTA

Przyspieszenie dla drugiego segmentu rozpoczyna się po zakończeniu wykonywania pierwszego segmentu, gdy bieżąca pozycja znajdzie się w docelowym punkcie. Zbocze narastania prędkości określone jest parametrem ACCEL, a zbocze hamowania parametrem DECEL.



8 FUNKCJE JĘZYKA AS DOSTĘPNE W ASTORINO

8.1 POLECENIA DO STEROWANIA PROGRAMEM

SPEED	Ustawia prędkość monitorowania.
PRIME	Przygotowuje program do wykonania
EXECUTE	Wykonuje program
HOLD	Zatrzymuje wykonanie
DO	Wykonuje pojedynczą instrukcję programu

SPEED szybkość monitorowania

Funkcja

Ustawia prędkość monitorowania, wyrażaną w procentach.

Parametr

Szybkość monitorowania

Ustawia prędkość w procentach. Jeśli parametr ten ma wartość 100, prędkość będzie ustawiona na 100% prędkości maksymalnej. Jeśli wartość ta jest równa 50, prędkość będzie ustawiona na połowę prędkości maksymalnej.

Dodatkowe informacje

Prędkość ruchu robota to iloczyn prędkości monitorowania (ustawianej przy pomocy tego polecenia) i prędkości programu (ustawianej w programie przy pomocy instrukcji SPEED). Przykładowo, jeśli prędkość monitorowania jest ustawiona na 50, a prędkość w programie jest ustawiona na 60, maksymalna prędkość robota będzie wynosiła 30%.

UWAGA

Jeśli iloczyn prędkości monitorowania i prędkości programu ma wartość wyższą niż 100, maksymalna prędkość robota jest automatycznie ustawiana na 100%.

Domyślna wartość prędkości monitorowania to 80%. Polecenie to nie ma wpływu na prędkość ruchu, który jest aktualnie realizowany. Nowo wprowadzona prędkość jest stosowana po ukończeniu bieżącego ruchu i zaplanowanego ruchu.

Przykład

Jeśli prędkość programu jest ustawiona na 100%:

>SPEED 30 ← Prędkość ruchu robota jest ustawiona na 30 % prędkości maksym.

>SPEED 50 ← Prędkość ruchu robota jest ustawiona na 50 % prędkości maksym.

>SPEED 100 ← Prędkość ruchu robota jest ustawiona na 100 % prędkości maksym.

>SPEED 200 ← Prędkość ruchu robota jest ustawiona na 100 % prędkości maksym.

PRIME nazwa programu

Funkcja

Przygotowuje system w taki sposób, aby program mógł być wykonany przy pomocy przycisku CYCLE START. Polecenie to zastosowane samodzielnie nie powoduje wykonania programu.

Parametr

Nazwa programu

Wybiera program, który ma być przygotowany do wykonania. Jeśli nie podano tego parametru, wybierany jest ostatnio wykonany program lub też program ostatnio wykorzystany w poleceniu prime.

Dodatkowe informacje

Polecenie to przygotowuje do wykonywania programu tylko system. Nie powoduje ono wykonania programu. Po przygotowaniu systemu przez polecenia PRIME, program może być

wykonany przy pomocy polecenia EXECUTE. Program może być także wykonany przy pomocy przycisku CYCLE START.

EXECUTE nazwa programu

Funkcja

Wykonuje program robota.

Parametr

Nazwa programu

Wybiera program, który ma być wykonany. W przypadku nie podania tego parametru, wybierany jest ostatnio wykonany program (przy pomocy polecenia EXECUTE, PRIME).

Przykład

>EXECUTE test ↵ Wykonuje w sposób ciągły program o nazwie "test". (Program jest wykonywany ciągle, aż do zatrzymania, np. za pomocą polecenia HOLD lub do momentu wystąpienia błędu.)

>EXECUTE ↵ Wykonuje ostatnio wykonywany program.

HOLD

Funkcja

Natychmiastowo zatrzymuje wykonywanie programu robota.

Dodatkowe informacje

Ruch robota jest natychmiastowo zatrzymywany. W przeciwieństwie do zastosowania przełącznika EMERGENCY STOP, nie następuje nagłe zatrzymanie ruchu.

DO instrukcja programu

Funkcja

Wykonuje pojedynczą instrukcję programu. (Niektóre instrukcje programu nie mogą być używane wraz z tym poleceniem.)

Parametr

Instrukcja programu

Wykonuje podaną instrukcję programu. W przypadku nie podania tego parametru, wykonywana jest ponownie instrukcja podana przy ostatnim wywołaniu polecenia DO.

Dodatkowe informacje

Instrukcje programu są zazwyczaj wpisywane w programie i wykonywane jako kroki programu. Nie mniej jednak, polecenie DO umożliwia wykonanie pojedynczej instrukcji bez potrzeby tworzenia specjalnie w tym celu programu.

Przykład

>DO JMOVE safe ← Przejście robota do pozycji "safe" w ruchu z interpolacją osiową.

>DO HOME ← Przejście robota do pozycji domowej w ruchu z interpolacją osiową.

8.2 POLECENIA OBSŁUGI ZMIENNYCH POZYCJI

HERE Przepisuje aktualną pozycję do podanej zmiennej

HERE zmienna pozycji

Funkcja

Polecenie to przypisuje aktualną pozycję do podanej zmiennej pozycji. Pozycja może być wyrażana w wartościach transformacji, przesunięciach osi lub w złożonych wartościach transformacji.

Parametr

Zmienna pozycji

Wartość zmiennej może być podana za pomocą współrzędnych transformacji, przesunięć osi lub przy pomocy złożonych współrzędnych transformacji.

Dodatkowe informacje

Pozycja może być wyrażana w wartościach transformacji, przesunięciach osi lub w złożonych wartościach transformacji.

Uwaga

Określana jest tylko zmienna położona skrajnie po prawej stronie wartości transformacji. (Proszę porównać z przykładem poniżej). Jeśli pozostałe zmienne używane w wartościach transformacji złożonej nie są zdefiniowane, w czasie wykonywania polecenia generowany jest błąd.

Wartości zmiennej są wyświetlane na terminalu, a po nich pokazywany jest komunikat "Change?" Wartości też można zmienić poprzez wprowadzenie liczb oddzielanych przecinkiem.

Wartość, która nie jest zmieniana może być pominięta. Aby zakończyć edycję wartości, po komunikacie "Change?" należy wcisnąć klawisz ←.

Jeśli zmienna jest wprowadzana poprzez podanie przesunięć osi (nazwa zmiennej rozpoczyna się symbolem #), wyświetlane są wartości osi dla aktualnej pozycji. Jeśli zmienna zawiera współrzędne transformacji, wyświetlane są wartości XYZRxRyRz. Wartości XYZ określają położenie punktu początkowego układu współrzędnych względem bazowego układu współrzędnych. Wartości RxRyRz określają orientację układu współrzędnych narzędzia.

Przykład

>HERE #pick ↵ Przepisuje bieżącą pozycję robota do zmiennej "#pick" (wartości przesunięcia osi).

>HERE place ↵ Przepisuje bieżącą pozycję robota do zmiennej "place" (współrzędne transformacji).

8.3 POLECENIA DO STEROWANIA SYSTEMEM

TOOL	Definiuje współrzędne transformacji dla współrzędnych narzędzia
SETHOME	Konfiguruje pozycję domową.
ERESET	Kasuje błąd

TOOL numer układu tool

Funkcja

Polecenie to definiuje współrzędne transformacji dla narzędzia, określające pozycję układu współrzędnych narzędzia względem zerowego układu współrzędnych.

Parametr

Numer układu tool

Jeden z 3 zapisanych układów tool.

Wartość

1

2

3

Dodatkowe informacje

Jeżeli jako parametr zostanie podana wartość od 1 do 3, współrzędne transformacji układu współrzędnych narzędzia są ustawione na wartości zapisane w pamięci robota. Zerowy układ współrzędnych narzędzia posiada środek na kołnierzu montażowym narzędzia a osie są równoległe do osi ostatniej pary kinematycznej robota. Podczas inicjacji systemu, współrzędne transformacji narzędzia są automatycznie ustawiane na zerowe.

Po przejściu robota się do ustawienia zdefiniowanego przy pomocy współrzędnych transformacji lub przez sterowanie ręczne w układzie bazowym lub układzie współrzędnych narzędzi, system automatycznie oblicza pozycję robota uwzględniając wprowadzone współrzędne transformacji narzędzia.

Przykład

>TOOL 1 ← Zmienia pozycję układu współrzędnych narzędzia na pozycję określoną w zakładce TOOL.

SETHOME HERE

Funkcja

Polecenie to ustawia i wyświetla pozycję HOME.

Parametr

HERE

Definiuje aktualną pozycję jako pozycję HOME.

Przykład

>SETHOME HERE ← Konfiguruje aktualną pozycję jako pozycję HOME.

ERESET

Funkcja

Kasuje błąd. Działanie tego polecenia jest identyczne z wciśnięciem przycisku RESET w oprogramowaniu RobotController.

Dodatkowe informacje

W momencie wykonania polecenia ERESET wysyłany jest sygnał RESET. Nie mniej jednak, polecenie to nie ma żadnego efektu, jeżeli jednocześnie wystąpi błąd.

8.4 POLECENIA DLA SYGNAŁÓW BINARNYCH

RESET Wyłącza wszystkie zewnętrzne sygnały We/Wy

SIGNAL Włącza (lub wyłącza) sygnały wyjściowe

RESET

Funkcja

Wyłącza wszystkie zewnętrzne sygnały wyjściowe. Polecenie to nie ma wpływu na sygnały dedykowane.

Uwaga!

Należy zwrócić uwagę, że polecenie to wyłącza wszystkie sygnały za wyjątkiem podanych powyżej, nawet w trybie odtwarzania.

SIGNAL numer sygnału

Funkcja

Włącza lub wyłącza określone zewnętrzne lub wewnętrzne sygnały Wy.

Parametr

Numer sygnału

Numer zewnętrznego sygnału wyjściowego lub sygnału wewnętrznego. Liczba dodatnia włącza sygnał, a ujemna wyłącza ten sygnał.

Dodatkowe informacje

Numer sygnału określa, czy jest on zewnętrzny, czy też wewnętrzny.

Dopuszczalne numery sygnałów

Zewnętrzne sygnały wyjściowe 1 – 8

Sygnały wewnętrzne 2001–2016

Zewnętrzne sygnały wejściowe nie może podawać

Jeśli numer sygnału jest liczbą dodatnią, sygnał jest włączany; jeśli jest liczbą ujemną, sygnał jest wyłączany.

Przykład

>SIGNAL -1 ← Wyłączenie zewnętrznego sygnału wyjściowego 1,

>SIGNAL 2005 ← Włączenie wewnętrznego sygnału 2005,

>SIGNAL -reset ← Jeśli zmienna "reset" ma wartość dodatnią, wyłączany jest sygnał wyjściowych określany przez tę wartość oraz włączany jest sygnał wyjściowy 4.

8.5 POLECENIA DO WYŚWIETLANIA KOMUNIKATÓW

PRINT Wyświetla dane

PRINT dane do wyświetlenia

Funkcja

Wyświetla na terminalu zadane dane.

Parametr

Dane do wyświetlenia

Wybrać jedną lub więcej z poniższych opcji.

(1) Ciąg znaków np. PRINT TST

(2) Wyrażenie typu rzeczywistego (wartość jest najpierw obliczana, a następnie wyświetlana, np. PRINT SIN(30))

Przykład

W niniejszym przykładzie wartość zmiennej rzeczywistej "i" równa się 5

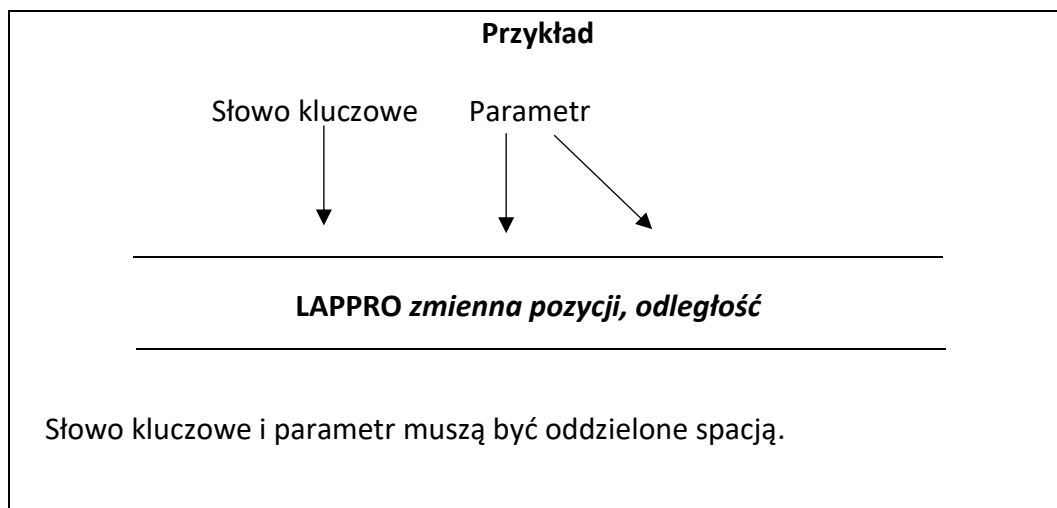
```
>PRINT i ↵
```

Wyświetlacz powinien wyglądać jak poniżej:

```
>5.00
```

9 INSTRUKCJE PROGRAMU

Instrukcja programu składa się ze słowa kluczowego wyrażającego polecenie i parametru (parametrów), jak pokazano w przykładzie zamieszczonym poniżej.



9.1 INSTRUKCJE RUCHU

JMOVE	Ruch robota z interpolacją osiową
LMOVE	Ruch robota z interpolacją liniową
DELAY	Zatrzymuje ruch robota na określony czas
JAPPRO	Zbliża robota do punktu docelowego ruchem z interpolacją osiową
LAPPRO	Zbliża robota do punktu docelowego z interpolacją liniową
JDEPART	Opuszcza bieżącą pozycję ruchem z interpolacją osiową
LDEPART	Opuszcza bieżącą pozycję ruchem z interpolacją liniową
HOME	Ruch robota do pozycji domowej
DRIVE	Ruch w kierunku pojedynczej osi
DRAW	Przejście o określoną odległość w kierunku osi X, Y, Z globalnego układu współrzędnych
TDRAW	Przejście o określoną odległość w kierunku osi X, Y, Z układu współrzędnych narzędzia
C1MOVE	Ruch z interpolacją kołową
C2MOVE	Ruch z interpolacją kołową

JMOVE zmienna pozycji

LMOVE zmienna pozycji

Funkcja

Ruch robota do określonej pozycji.

JMOVE: Ruch z interpolacją osiową.

LMOVE: Ruch z interpolacją liniową.

Parametr

Zmienna pozycji

Określa pozycję docelową robota.

Dodatkowe informacje

Instrukcja JMOVE powoduje ruch robota z interpolacją osiową. Robot porusza się w taki sposób, że stosunek przebytej odległości do całkowitej odległości pozostaje równy dla wszystkich osi, przez cały czas trwania ruchu, od pozycji początkowej do pozycji końcowej. Podczas wykonywania instrukcji LMOVE robot porusza się ruchem z interpolacją liniową. Punkt początkowy układu współrzędnych narzędzia (TCP) przesuwa się wzdłuż toru liniowego.

Przykład

JMOVE #pick - Ruch z interpolacją osiową do pozycji określonej za pomocą wartości przemieszczeń osi "#pick".

LMOVE P1 Ruch z interpolacją liniową do pozycji określonej za pomocą współrzędnych transformacji P1.

LMOVE #pick Ruch z interpolacją liniową do pozycji określonej przez wartości przesunięć osi "#pick".

DELAY czas

Funkcja

Zatrzymuje ruch robota na określony czas.

Parametr

Czas

Określa w sekundach czas zatrzymania ruchu robota.

Dodatkowe informacje

W systemie AS, instrukcja DELAY jest traktowana jak instrukcja ruchu, która "nie zmienia pozycji".

Nawet jeśli ruch robota jest zatrzymywany przez instrukcję DELAY, przed zatrzymaniem wykonywane są wszystkie kroki programu znajdujące się przed kolejną instrukcją ruchu.

Przykład

DELAY 2.5 Zatrzymuje ruch robota na 2,5 sekundy.

JAPPRO zmienna pozycji , odległość

LAPPRO zmienna pozycji , odległość

Funkcja

Ruch w kierunku osi Z narzędzia na zadaną odległość od pozycji wyuczonej.

JAPPRO: Ruch z interpolacją osiową.

LAPPRO: Ruch z interpolacją liniową.

Parametr

Zmienna pozycji

Określa pozycję końcową (w wartościach transformacji lub wartościach przesunięć osi).

Odległość

Określa odległość (w milimetrach) pomiędzy pozycją końcową, a pozycją faktyczną uzyskaną przez robota w kierunku osi Z układu współrzędnych narzędzia. Jeżeli podana odległość jest dodatnia, robot porusza się w kierunku ujemnym osi Z. Jeżeli podana odległość jest ujemna, robot porusza się w kierunku dodatnim osi Z.

Dodatkowe informacje

W poleceniach tych, narzędzia jest orientowane w określonej pozycji, a pozycja jest ustawiana w określonej odległości od zadanej pozycji w kierunku osi Z układu współrzędnych narzędzia.

Przykład

JAPPRO place,100 Ruch z interpolacją osiową do pozycji 100 mm od pozycji "place", w kierunku osi Z układu współrzędnych narzędzia. Pozycja "place" jest podana za pomocą współrzędnych transformacji.

LAPPRO place, offset Ruch z interpolacją liniową do pozycji oddalonej od pozycji "place", podanej za pomocą współrzędnych transformacji, o odległość określoną przez zmienną "offset", w kierunku osi Z układu współrzędnych narzędzia.

JDEPART odległość

LDEPART odległość

Funkcja

Ruch robota do pozycji oddalonej od bieżącej pozycji o zadaną określoną odległość, wzdłuż osi Z układu współrzędnych narzędzia.

JDEPART: Ruch z interpolacją osiową.

LDEPART : Ruch z interpolacją liniową.

Parametr

Odległość

Określa w milimetrach odległość pomiędzy aktualną pozycją, a pozycją docelową w kierunku osi Z układu współrzędnych narzędzia. Jeżeli podana odległość jest dodatnia, robot porusza się "wstecz", lub inaczej, w kierunku wartości ujemnych osi Z. Jeżeli podana odległość jest ujemna, robot porusza się "w przód", lub inaczej, w kierunku wartości dodatnich osi Z.

Przykład

JDEPART 80 Ruch robota z interpolacją osiową wstecz o 80 mm w kierunku osi -Z układu współrzędnych narzędzia.

LDEPART 2*offset Ruch robota z interpolacją liniową wstecz, na odległość 2*offset (200 mm jeśli offset = 100) w kierunku osi - Z układu współrzędnych narzędzia.

HOME

Funkcja

Ruch z interpolacją osiową do pozycji zdefiniowanej jako HOME.

Przykład

HOME Ruch z interpolacją osiową do pozycji domowej określonej za pomocą polecenia/instrukcji SETHOME lub w oprogramowaniu Astorino- RobotController

DRIVE numer osi, przesunięcie, prędkość

Funkcja

Ruch jednej osi robota.

Parametr

Numer osi

Numer osi, która ma być przesunięta. (W przypadku robotów wyposażonych w sześć osi, są one oznaczone numerami od 1 do 6, zaczynając od osi najbardziej oddalanej od kołnierza montażowego narzędzia.)

Przesunięcie

Wielkość przesunięć osi, podawana jako wartość dodatnia lub ujemna.

Wartość ta jest wyrażana w tych samych jednostkach, które używane są do opisu pozycji osi, tzn. jeśli oś jest osią obrotową, wartość ta jest wyrażana w stopniach ($^{\circ}$), a jeśli oś jest osią przesuwną, wartość ta jest wyrażana w jednostce odległości (mm).

Prędkość

Prędkość ruchu. Podobnie jak w przypadku standardowej prędkości ruchu, jest ona wyrażana w procentach prędkości monitorowania. W przypadku nie podania tego parametru, przyjmuje się 100% prędkości monitorowania.

Dodatkowe informacje

Omawiana instrukcja przesuwa tylko jedną oś.

Prędkość monitorowania tej instrukcji to kombinacja prędkości podanej w tej instrukcji oraz prędkości monitorowania. Prędkość programu ustawiona w programie nie ma wpływu na tę instrukcję.

Przykład

DRIVE 2,-10,75 Przesunięcie osi 2 (JT2) z bieżącej pozycji o -10° . Prędkość ta stanowi 75% prędkości monitorowania.

DRAW przesuw X, przesuw Y, przesuw Z, obrót wokół X, obrót wokół Y, obrót wokół Z

TDRAW przesuw X, przesuw Y, przesuw Z, obrót wokół X, obrót wokół Y, obrót wokół Z

UWAGA! W obecnej wersji języka funkcje DRAW oraz TDRAW przyjmują tylko argumenty X,Y,Z obroty nie są wspierane i są w fazie opracowania

Funkcja

Ruch robota z interpolacją liniową, z określoną prędkością, z bieżącej pozycji na zadaną odległość w kierunku osi X, Y, Z i obrót o zadany kąt wokół każdej z osi. Instrukcja DRAW

przesuwa robot w globalnym układzie współrzędnych, a instrukcja TDRAW przesuw robota w układzie współrzędnych narzędzia.

Parametr

Przesuw X

Wielkość przesunięcia w osi X wyrażona w mm. W przypadku nie podania tego parametru, wprowadzana jest wartość 0 mm.

Przesuw Y

Wielkość przesunięcia w osi Y wyrażona w mm. W przypadku nie podania tego parametru, wprowadzana jest wartość 0 mm.

Przesuw Z

Wielkość przesunięcia w osi Z wyrażona w mm. W przypadku nie podania tego parametru, wprowadzana jest wartość 0 mm.

Obrót wokół X

Kąt obrotu wokół osi X wyrażony w stopniach. Dopuszczalny zakres wartości jest mniejszy od $\pm 180^\circ$. W przypadku nie podania tego parametru, przyjmowana jest wartość 0 stopni.

Obrót wokół Y

Kąt obrotu wokół osi Y wyrażony w stopniach. Dopuszczalny zakres wartości jest mniejszy od $\pm 180^\circ$. W przypadku nie podania tego parametru, przyjmowana jest wartość 0 stopni.

Obrót wokół Z

Kąt obrotu wokół osi Z wyrażony w stopniach. Dopuszczalny zakres wartości jest mniejszy od $\pm 180^\circ$. W przypadku nie podania tego parametru, przyjmowana jest wartość 0 stopni.

Dodatkowe informacje

Robot przesuwa się ruchem liniowym z bieżącej pozycji do pozycji zadanej.

Przykład

DRAW 50,-30 Ruch robota z interpolacją liniową o 50 mm w kierunku osi X i o -30 mm w kierunku osi Z globalnego układu współrzędnych.

C1MOVE zmienna pozycji

C2MOVE zmienna pozycji

Funkcja

Ruch robota do określonej pozycji z interpolacją kołową.

Parametr

Zmienna pozycji

Określa docelową pozycję ruchu robota. (Może być wyrażona za pomocą współrzędnych transformacji, wartości złożonej transformacji, wartości przesunięć osi lub funkcje z informacją o pozycji.)

Dodatkowe informacje

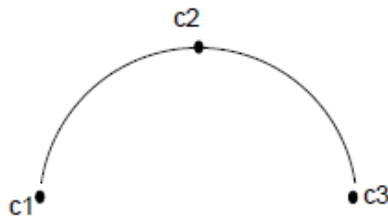
Instrukcja C1MOVE przemieszcza robota do punktu znajdującego się w połowie drogi na torze kołowym, a instrukcja C2MOVE przesuwa do końca toru kołowego.

Aby przesuwać robota ruchem z interpolacją kołową, niezbędne jest wyuczenie trzech pozycji. Te trzy pozycje są różne dla instrukcji C1MOVE i C2MOVE.

1. Pozycja z ostatniej instrukcji ruchu lub obecna pozycja.
2. Pozycja, które będzie użyta jako parametr instrukcji C1MOVE.
3. Pozycja następnej instrukcji ruchu C2MOVE.

Przykład

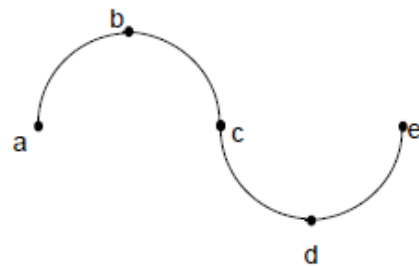
```
JMOVE c1  
C1MOVE c2  
C2MOVE c3
```



Robot przesuwa się ruchem z interpolacją osiową do c1, a następnie przesuwa się ruchem z interpolacją kołową po łuku utworzonym przez c1, c2 i c3.

```
JMOVE #a  
C1MOVE #b  
C2MOVE #c  
C1MOVE #d  
C2MOVE #e
```

} łuk a,b,c
} łuk c,d,e



9.2 INSTRUKCJE STERUJĄCE PRĘDKOŚCIĄ I DOKŁADNOŚCIĄ

SPEED Ustawia prędkości ruchu (prędkość programu)
ACCEL Ustawia przyspieszenie
DECEL Ustawia hamowanie

SPEED prędkość, prędkość kątowna

Funkcja

Określa prędkość ruchu robota.

Parametr

Prędkość

Określa prędkość programu. Zwykle jest ona podawana jako wartość procentowa z zakresu 0.01 do 100 (%). Można również podać wartość bezwzględną, dopisując jednostki: MM/S

Dodatkowe informacje

Faktyczna prędkość ruchu robota to iloczyn prędkości monitorowania oraz prędkości ruchu ustawionej za pomocą tej instrukcji (prędkość monitorowania x prędkość programu). Nie mniej jednak, nie można zagwarantować uzyskania pełnej prędkości w następujących przypadkach:

1. jeżeli odległość pomiędzy dwoma wyuczonymi pozycjami jest za mała

2. jeżeli wyuczono ruch z interpolacją liniową, który przekracza maks. prędkość kątową osi.

Prędkość ruchu jest określana inaczej w ruchu z interpolacją osiową i ruchu z interpolacją liniową. W przypadku interpolacji osiowej, prędkość ruchu jest określana jako procent maksymalnej prędkości każdej z osi. W ruchu z interpolacją liniową, prędkość ruchu jest określana jako procent prędkości maksymalnej w punkcie początkowym układu współrzędnych narzędzia. Jeżeli prędkość jest podawana w jednostce odległości na jednostkę czasu, konfigurowana jest prędkość ruchu liniowego w punkcie początkowym układu współrzędnych narzędzia. Podczas ruchu z interpolacją osiową, prędkość jest ustawiana w procentach. (Nawet jeśli prędkość ustawiono jako wartość absolutną lub czas ruchu, robot nie będzie się poruszał z tak ustawioną prędkością. Prędkość natomiast będzie przetwarzana jako procent, jaki stanowi dana wartość, prędkości maksymalnej.)

Prędkość absolutna jest wyrażana w mm/s. Zmniejszenie prędkości monitorowania powoduje proporcjonalne zmniejszenie tych prędkości.

Uwaga!

Nawet, jeżeli iloczyn prędkości programu oraz prędkości ustawionej za pomocą polecenia SPEED (prędkość programu) przekracza 100%, faktyczna prędkość ruchu nie przekroczy 100%.

Przykład

Jeśli prędkość jest ustawiona jako podano poniżej, a prędkość monitorowania wynosi 100%:

SPEED 50 Ustawia prędkość kolejnego ruchu na 50 % prędkości maksymalnej.

SPEED 100 Ustawia prędkość kolejnego ruchu na 100 % prędkości maksymalnej.

SPEED 200 Ustawia prędkość kolejnego ruchu na 100 % prędkości maksymalnej (prędkość przekraczająca 100 % jest odczytywana jako 100 %).

SPEED 20MM/S Prędkość punktu początkowego układu współrzędnych narzędzia (TCP) jest ustawiona na 20 mm/sek (jeśli prędkości monitorowania wynosi 100 %).

(Prędkość ruchu liniowego punktu początkowego układu współrzędnych narzędzia).

ACCEL przyśpieszenie

DECEL hamowanie

Funkcja

Ustawia przyspieszenie (lub hamowanie) ruchu robota.

Parametr

Przyspieszenie / Hamowanie

Ustawia przyspieszenie lub hamowanie ruchu robota w procentach maksymalnego przyspieszenia (hamowania). Dopuszczalny zakres wartości wynosi od 0,01 do 100. Wartości wykraczające poza ten zakres są traktowane jako 100, a wartości poniżej zakresu są traktowane jako 0,01.

Dodatkowe informacje

Instrukcja ACCEL ustawia przyspieszenie w momencie rozpoczynania ruchu robota jako procent maksymalnego przyspieszenia. Instrukcja DECEL ustawia hamowanie w momencie kończenia ruchu robota jako procent maksymalnego hamowania.

Przykład

ACCEL 80 Przyspieszenie jest ustawione na 80% dla wszystkich ruchów następujących po tej instrukcji.

DECEL 50 Hamowanie dla kolejnych instrukcji ruchu zostaje ustawione na 50%.

9.3 INSTRUKCJE STERUJĄCE PROGRAMEM

TWAIT Wstrzymuje wykonywanie programu do momentu upływu określonego czasu

TWAIT czas

Funkcja

Wstrzymuje wykonywanie programu do momentu upływu określonego czasu.

Parametr

Czas

Określa w sekundach czas wstrzymania wykonywania programu.

Dodatkowe informacje

Instrukcja ta wstrzymuje wykonywanie programu do momentu upływu określonego czasu.

Przykład

TWAIT 0.5 Oczekiwanie przez 0,5 sekundy.

TWAIT deltat Oczekiwanie do momentu upływu czasu określonego przez zmienną "deltat".

9.4 INSTRUKCJE STRUKTURY PROGRAMU

IF.....THEN...ELSE.....END

WHILE.....DO.....END

DO.....UNTIL

FOR.....END

IF wyrażenie logiczne THEN

instrukcje programu(1)

ELSE

instrukcje programu(2)

END

Funkcja

Wykonuje jedną z grup kroków programu w zależności od wartości wyrażenia logicznego.

Parametr

Wyrażenie logiczne

Wyrażenie logiczne lub wyrażenie na wartościach rzeczywistych. Wyrażenie to zwraca wartość True (wartość różna od 0) lub False (0).

Instrukcje programu (1)

Instrukcje programu wykonywane, jeśli wyrażenie logiczne ma wartość logiczną TRUE.

Instrukcje programu (2)

Instrukcje programu wykonywane, jeśli wyrażenie logiczne ma wartość logiczną FALSE.

Dodatkowe informacje

Omawiana instrukcja wykonuje jedną z dwóch grup instrukcji, zależnie od wartości wyrażenia logicznego. Poniżej przedstawiono procedurę wykonywania:

1. Oblicz wartość wyrażenia logicznego i przejdź do kroku 4, jeśli wyrażenie to ma wartość 0 (FALSE).
2. Oblicz wartość wyrażenia logicznego i przejdź do wykonywania instrukcji programu (1), jeśli wyrażenie to ma wartość 1 (TRUE).
3. Przejdź do 5.
4. Jeśli istnieje instrukcja ELSE, wykonaj instrukcje programu (2).
5. Kontynuuj wykonywanie programu od kroku umieszczonego za słowem kluczowym END.

UWAGA

1. Instrukcje ELSE i END muszą być jedynymi słowami kluczowymi w liniach, w których się znajdują.
2. Instrukcja IF...THEN musi się kończyć słowem kluczowym END.

Przykład

W przykładowym programie poniżej, jeśli n jest większe od 5, prędkość programu zostaje ustawiana na 10%, a jeżeli nie, to na 20 %.

```
21 IF n>5 THEN
```

```
22     sp=10
```

```
23 ELSE
24     sp=20
25 END
26 SPEED sp
```

Zamieszczony poniżej program sprawdza najpierw wartość zmiennej "m". Jeśli zmienna "m" jest różna od 0, program sprawdza zewnętrzny sygnał wejściowy 1001 i wyświetla odpowiedni komunikat, w zależności od statusu tego sygnału. W przykładzie tym, zewnętrzna struktura IF nie posiada instrukcji ELSE.

```
71 IF m THEN
72     IF SIG(1001) THEN
73         PRINT 1.0
74     ELSE
75         PRINT 0.0
76     END
77 END
```

WHILE warunek DO **instrukcje programu** **END**

Funkcja

Jeśli podany warunek ma wartość TRUE, wykonywane są instrukcje programu. Jeśli warunek ma wartość FALSE, instrukcja WHILE jest pomijana.

Parametr

Warunek

Wyrażenie logiczne lub wyrażenie na wartościach rzeczywistych. Sprawdza jaką wartość ma wyrażenie: TRUE (różna od 0), czy FALSE (0).

Instrukcje programu

Grupa instrukcji, które mają być wykonane, jeśli warunek ma wartość TRUE.

Dodatkowe informacje

Omawiana struktura powtarza określone kroki programu przez cały czas, kiedy warunek ma wartość TRUE. Poniżej przedstawiono procedurę wykonania:

1. Oblicz wartość wyrażenia logicznego i przejdź do kroku 4, jeśli wyrażenie to ma wartość 0 (FALSE).
2. Oblicz wyrażenie logiczne i wykonuj instrukcje programu, jeśli wyrażenie to ma wartość 1 (TRUE).
3. Przejdź do 1.
4. Kontynuuj wykonywanie programu od kroku umieszczonego za słowem kluczowym END.

UWAGA

W odróżnieniu od instrukcji DO, jeżeli wyrażenie ma wartość FALSE, kroki wewnątrz konstrukcji WHILE nie są wykonywane. W czasie korzystania z tej instrukcji, warunek może się zmienić z TRUE na FALSE.

Przykład

W zamieszczonym poniżej przykładzie, monitorowany jest sygnał wejściowy 1001 oraz zatrzymywany jest ruch robota w zależności od jego stanu. Jeśli sygnał pochodzący z podajnika części zmienia swoją wartość na 0 (podajnik jest pusty), robot zatrzymuje się, a wykonanie jest kontynuowane od kroku po instrukcji END (krok 27 w niniejszym przykładzie).

Jeśli jeden z podajników jest pusty w czasie uruchomienia struktury WHILE (zewnątrzny sygnał wejściowy = 0), nie jest wykonywany żaden krok struktury, a przetwarzanie jest kontynuowane od kroku 27.

```
20 .
21 .
22 .
23 WHILE SIG(1001) DO
24     LMOVE P1
25     TWAIT 1
26     LMOVE part
27 END
28 .
29 .
30 .
```

DO

instrukcje programu

UNTIL wyrażenie logiczne

Funkcja

Tworzy pętlę iteracyjną DO

Parametr

Instrukcje programu

Instrukcje te są powtarzane przez cały czas, kiedy wyrażenie logiczne ma wartość FALSE.

Wyrażenie logiczne

Wyrażenie logiczne lub wyrażenie na wartościach rzeczywistych. Jeżeli to wyrażenie logiczne zmieni wartość na TRUE, instrukcje wewnątrz pętli nie są już dalej wykonywane.

Dodatkowe informacje

Omawiana struktura wykonuje grupę instrukcji programu, jeśli podany warunek (wyrażenie logiczne) ma wartość FALSE.

Poniżej przedstawiono procedurę wykonywania:

1. Wykonaj instrukcje programu.
2. Sprawdź wartość wyrażenia logicznego i jeżeli jest to wartość FALSE powtórz krok 1. Jeżeli wyrażenie logiczne ma wartość TRUE, przejdź do kroku 3.
3. Kontynuuj wykonywanie programu od kroku umieszczonego po instrukcji UNTIL.

Wykonywanie struktury DO jest kończone, gdy wartość wyrażenia logicznego zmieni się z FALSE na TRUE.

UWAGA

W odróżnieniu od konstrukcji WHILE, instrukcje konstrukcji DO wykonywane są zawsze co najmniej jeden raz. Instrukcje programu umieszczone pomiędzy słowami DO i UNTIL można pominąć. Jeżeli nie ma wstawionych żadnych instrukcji, przez cały czas jest wyznaczana wartość wyrażenia logicznego umieszczonego po słowie UNTIL. W momencie, gdy wartość tego wyrażenia zmieni się na TRUE, następuje wyjście z pętli i przejście do kroku umieszczonego po konstrukcji DO.

Konstrukcja DO musi zawsze kończyć się słowem kluczowym UNTIL.

Przykład

W przykładzie poniżej, konstrukcja DO realizuje następujące zadanie: część jest podnoszona i przenoszona do bufora. Po wypełnieniu bufora następuje załączenie binarnego sygnału wejściowego "buffer". Załączenie tego sygnału wymusza zatrzymanie się robota i przejście do realizacji innego zadania.

```
10 .
11 .
12 .
13 DO
14   JMOVE get
15   JMOVE put
16 UNTIL (SIG(buffer))
17 .
18 .
19 .
```

FOR zmienna sterująca = wart. początkowa TO wart. końcowa

instrukcje programu

END

Funkcja

Powtarzanie wykonywania instrukcji programu.

Parametr

Zmienna sterująca

Zmienna lub wartość rzeczywista. Zmienna ta na samym początku jest ustawiana na wartość początkową i jest następnie zwiększana o 1 po każdym wykonaniu pętli.

Wartość początkowa

Wartość lub wyrażenie typu rzeczywistego Parametr ten służy do konfigurowania wartości początkowej zmiennej sterującej pętlą.

Wartość końcowa

Wartość lub wyrażenie typu rzeczywistego Wartość ta jest porównywana z aktualną wartością zmiennej sterującej pętlą, a w momencie dojścia do tej wartości następuje wyjście z pętli.

Dodatkowe informacje

Omawiana konstrukcja powtarza wykonanie instrukcji programu pomiędzy słowami kluczowym FOR i END. Zmienna sterująca pętli jest powiększana o określoną wartość kroku po każdym wykonaniu pętli.

Poniżej przedstawiono procedurę wykonywania:

1. Przypisz wartość początkową do zmiennej sterującej pętli.
2. Oblicz wartość końcową oraz wartość kroku.
3. Porównaj wartość zmiennej sterującej pętli z wartością końcową.
 - a. Jeśli wartość kroku jest dodatnia, a zmienna sterująca pętli jest większa niż wartość końcowa, przejdź do kroku 7.
 - b. Jeśli wartość kroku jest ujemna, a zmienna sterująca pętli jest mniejsza od wartości końcowej, przejdź do kroku 7.

W pozostałych przypadkach przejdź do kroku 4.

4. Wykonaj instrukcje programu umieszczone po słowie kluczowym FOR.

5. Po dojściu do instrukcji END, dodaj wartość kroku do zmiennej sterującej pętli.

6. Wróć do kroku 3.

7. Wykonaj instrukcje programu umieszczone po instrukcji END. (Wartość zmiennej sterującej pętli w momencie porównywania w kroku 3 nie zmienia się.)

UWAGA

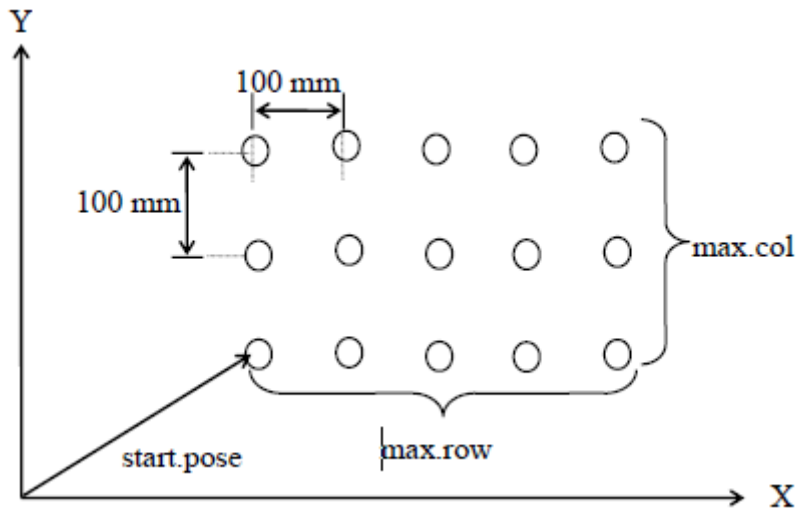
Każde słowo kluczowe FOR musi posiadać odpowiadające mu słowo kluczowe END. Zwróć uwagę, że jeżeli zmienna sterująca pętli jest większa od wartości końcowej (lub mniejsza dla ujemnej wartości kroku) w czasie pierwszego porównywania wartości, żadna z instrukcji programu pomiędzy słowami kluczowymi FOR i END nie zostanie wykonana.

Nie wolno zmieniać wartości sterującej pętli w obrębie pętli FOR poprzez programowanie (stosowanie operatorów, wyrażeń, itp.).

Przykład

Program "pick" podnosi część i umieszcza ją w pozycji "hole". Części są umieszczane w sposób przedstawiony na rysunku poniżej. (Paleta jest umieszczona równolegle do osi X i Y globalnego układu współrzędnych, a odległość pomiędzy częściami wynosi 100 mm).

```
FOR row = 1 TO maxrow
  POINT hole = SHIFT (startpose BY (row-1) * 100, 0, 0)
  FOR col = 1 TO maxcol
    JMOVE pick
    POINT hole = SHIFT ( hole BY 0, 100, 0)
  END
END
```



9.5 INSTRUKCJE DLA SYGNAŁÓW BINARNYCH

SWAIT Zawiesza program do momentu ustawienia określonego warunku

SWAIT numer sygnału

Funkcja

Oczekuje do momentu, kiedy określony sygnał zewnętrzny wejściowy lub sygnał wewnętrzny będzie miał żądany stan.

Parametr

Numer sygnału

Numer zewnętrznego sygnału wejściowy lub sygnału wewnętrznego, który ma być monitorowany.

Numer ujemny oznacza, że warunek jest spełniony, jeżeli sygnał jest wyłączony.

Dopuszczalne numery sygnałów

Zewnętrzne sygnały wejściowe 1001-1008

Sygnały wewnętrzne 2001–2016

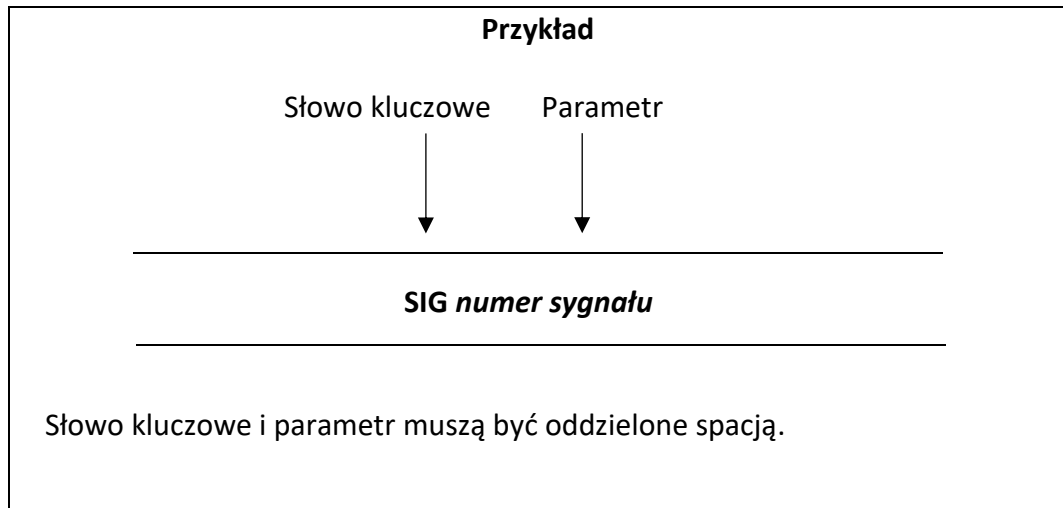
Przykład

SWAIT 1001 Oczekuje do włączenia zewnętrznego sygnału wejściowego 1001.

SWAIT -2001 Oczekuje do wyłączenia sygnału wewnętrznego 2001.

10 FUNKCJE

W niniejszym rozdziale przedstawiono funkcje używane w systemie AS. Funkcje te są zazwyczaj używane w połączeniu z poleceniami ekranowymi oraz instrukcjami programu. Poniżej przedstawiono format funkcji. Słowo kluczowe określa funkcję, a parametry wprowadzone w nawiasach określają wartości na których operuje funkcja.



10.1 FUNKCJE OPERUJĄCE NA WARTOŚCIACH RZECZYWISTYCH

SIG Wykonuje operację logiczną AND na określonym sygnale

SIG (numer sygnału)

Funkcja

Wykonuje operację logiczną AND na podanym sygnale.

Parametr

Numer sygnału

Numerzy zewnętrznych lub wewnętrznych sygnałów wejściowych.

Dodatkowe informacje

Wykonuje operacje logiczną AND na podanym sygnale i zwraca wartość wynikową. Jeśli sygnał ma wartość logiczną TRUE, zwraca jest wartość logiczna 1 (True). W przeciwnym wypadku zwracana jest wartość 0 (FALSE). Zewnętrzne lub wewnętrzne sygnały wejściowe są określane za pomocą odpowiednich numerów, jak pokazano poniżej.

Dopuszczalne numery sygnałów

Zewnętrzne sygnały wejściowe 1001 – 1008

Sygnały wewnętrzne 2001–2016

Zakłada się, iż sygnały określone liczbami dodatnimi przyjmują wartość TRUE (PRAWDA), gdy są włączone (ON), natomiast sygnały określone liczbami ujemnymi przyjmują wartość TRUE (PRAWDA), gdy są wyłączone (OFF). Numerowi sygnału "0" nie odpowiada żaden sygnał, więc przyjmuje się, iż zawsze ma on wartość TRUE.

Przykład

Jeśli binarny sygnał wejściowy 1001=ON, 1004=ON, 2004=OFF, to

Wyniki

SIG(1001) == TRUE

SIG(2004) == FALSE

SIG(-1004) == FALSE

10.2 FUNKCJE POZYCJI

TRANS	Zwraca współrzędne transformacji, obliczone na podstawie podanych składowych
#PPOINT	Zwraca wartości przemieszczeń osi, obliczone na podstawie podanych elementów
SHIFT	Zwraca współrzędną transformacji uzyskaną poprzez przesunięcie oryginalnej pozycji

TRANS (komponent X, komponent Y, komponent Z, komponent O, komponent A, komponent T, JT7)

Funkcja

Zwraca współrzędne transformacji, dla podanych elementów składowych określających przesunięcia i obroty.

Parametr

Komponent X, Komponent Y, Komponent Z

Składowe przesunięcia X, Y, Z. W przypadku nie podania tych parametrów, przyjmowane są wartości 0.

Komponent O, Komponent A, Komponent T

Określa składowe obroty O, A, T. W przypadku nie podania tych parametrów, przyjmowane są wartości 0.

Komponent JT7

Określa składową przesunięcia osi JT7 w mm, w przypadku nie podania tego argumentu przyjmowane są wartości 0.

Dodatkowe informacje

Instrukcja ta oblicza współrzędne transformacji na podstawie wartości podanych za pomocą parametrów. Nowe współrzędne transformacji można użyć do definiowania zmiennych pozycji, współrzędnych transformacji złożonej lub w instrukcjach ruchu. Omawiana funkcja jest bardzo przydatna w połączeniu z instrukcją DECOMPOSE (przykład podano przy opisie funkcji #PPOINT).

Przykład

POINT temppos = TRANS(a, b+100,c, d, e, f)

#PPOINT (jt1, jt2, jt3, jt4, jt5, jt6, jt7)

Funkcja

Zwraca wartości przesunięcia osi.

Parametr

Określa wartość każdego kąta (w stopniach dla osi obrotowych).

Dodatkowe informacje

Funkcja ta zwraca wartości przesunięcia osi. Wartości te reprezentują przemieszczenia w każdej z osi, od osi 1 do ostatniej osi (niekoniecznie szóstej).

UWAGA

Funkcja #PPOINT korzysta wyłącznie z wartości przesunięć osi. Z tego powodu, zawsze na początku tej funkcji należy podać przedrostek "#".

Przykład

POINT #temp = #PPOINT (0, a, a/2, 0, 0,0)

SHIFT(zmienna współrzędnych transformacji BY przesunięcie X, przesunięcie Y, przesunięcie Z)

Funkcja

Zwraca pozycję otrzymaną w wyniku przesunięcia zmiennej określonej za pomocą współrzędnych transformacji przesuniętej o zadaną odległość względem każdej osi bazowego układu współrzędnych (X,Y,Z).

Parametr

Zmienna współrzędnych transformacji

Zmienna współrzędnych transformacji, określająca pozycję, która ma być przesunięta.

Przesunięcie X, Przesunięcie Y, Przesunięcie Z

Wartości przesunięcia w osiach X, Y, Z bazowego układu współrzędnych.

Dodatkowe informacje

Wartości przesunięcia w osiach X, Y i Z są dodawanego do składowych przesunięcia X, Y, Z podanej zmiennej określonej za pomocą współrzędnych transformacji. Wynik jest zwracany jako współrzędne transformacji.

Przykład

Jeżeli zmienna współrzędnych transformacji ma wartości (200,150,100,10,20,30), to po wykonaniu instrukcji
POINT y=SHIFT(x BY 5, -5, 10)
pozycja "x" jest przesuwana o określone wartości do pozycji (205,145,110,10,20,30), która następnie jest przypisywana do zmiennej "y".

10.3 FUNKCJE MATEMATYCZNE

FABS	Zwraca wartość bezwzględną z wyrażenia numerycznego.
SIN	Zwraca wartość sinus.
COS	Zwraca wartość cosinus
ATAN	Zwraca wartość arcustangens.

FABS (wartość rzeczywista)

SIN (wartość rzeczywista)

COS (wartość rzeczywista)

ATAN (wartość rzeczywista1, wartość rzeczywista2)

11 PRZYKŁADOWE PROGRAMY

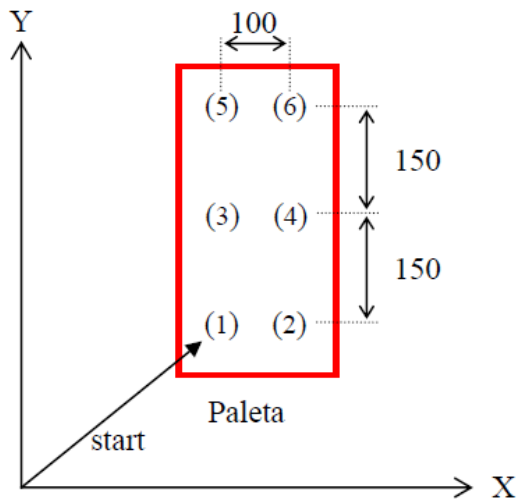
W niniejszym rozdziale przedstawiono przykładowe programy w języku AS.

11.1 KONFIGURACJA POCZĄTKOWA PROGRAMÓW

Wykonanie podanych poniżej czynności przed korzystaniem z jakichkolwiek funkcji robota ułatwia programowanie.

- Przesunąć robota do pozycji domowej.
- Zdefiniować potrzebne zmienne dla każdego zadania. (np. do paletyzacji, wprowadzić liczbę przedmiotów na palecie)
- Zainicjalizować licznik, flagę, itp.
- Ustawić układ współrzędnych narzędzia dla realizowanego zadania.
- Ustawić układ współrzędnych globalnych dla zadania.

Poniżej podano przykład podprogramu do inicjalizacji ustawień dla zadania paletyzacji przedstawionego na rysunku.

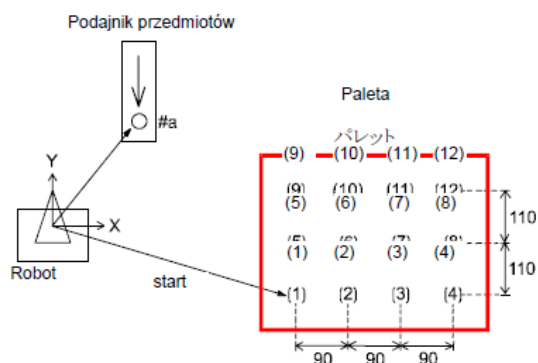


W zamieszczonym przykładzie przedmioty są paletyzowane w kolejności od (1) do (6). Należy wprowadzić podanej poniżej ustawienia początkowe. Paleta jest ustawiona równoległe do układu współrzędnych globalnych robota.

- 1 **TOOL 1** ;transformacja narzędzia (1)
- 2 **rowmax = 3** ; 3 wiersze
- 3 **colmax = 2** ; 2 kolumny
- 4 **xs = 100** ; odległość rozmieszczania w kierunku osi X ($\Delta X=100\text{mm}$).
- 5 **ys = 150** ; odległość rozmieszczenie w kierunku osi Y ($\Delta Y=150\text{mm}$).
- 6 **POINT put = start** ; przypisanie pozycji pose do pozycji zmiennej put.
- 7 **HOME** ; przejście do pozycji domowej

11.2 PALETYZACJA

W zamieszczonym poniżej przykładzie przedmioty są pobierane z podajnika i umieszczane na paletach w trzech rzędach (w odległości 110 mm) i czterech kolumnach (w odległości 90 mm). W celu uproszczenia przykładu, zarówno paleta, jak i przedmioty na palecie są ustawione równoległe do płaszczyzny XY w układzie współrzędnych globalnych robota. Pominięto także procedurę synchronizacji podajnika i robota za pomocą zewnętrznych sygnałów We/Wy (instrukcje SWAIT, SIGNAL, itp.).



- Paleta jest ustawiona równoległe do płaszczyzny XY globalnego układu współrzędnych.

- Pozycja #P0 (Podajnik przedmiotów) i pozycja "P0" (w której umieszczony jest pierwszy przedmiot) muszą być zdefiniowane przed rozpoczęciem wykonywania programu.

Przykładowy program

```
.PROGRAM palletize ()
; ustawienia początkowe (3 rzędy, 4 kolumny, ΔX=90, ΔY=110, itp.)
rowmax = 3
colmax = 4
xs = 90
ys = 110
gripper = 1 ; sygnał 1 sterujący chwytakiem
SPEED 100
POINT put = P0
SIGNAL -gripper ; otwarcie chwytaka
; Rozpoczęcie paletyzacji
FOR row = 1 TO rowmax
  FOR col = 1 TO colmax
    JAPPRO #P0, 100
    SPEED 30
    LMOVE #P0
    LDEPART 200
    JAPPRO put, 200
    SPEED 30
    LMOVE put
    SIGNAL -gripper
    LDEPART 200
    ; Obliczenie pozycji przedmiotu w następnym rzędzie.
    POINT put=SHIFT (put BY xs, 0, 0)
  END
  ; Obliczenie pozycji przedmiotu w następnej kolumnie.
  POINT put = SHIFT (P0 by 0, ys * row, 0)
END
.END
```