

DYDAKTYKA ABB



TASK/PROGRAM/MODUŁ/PROCEDURA/INSTRUKCJA

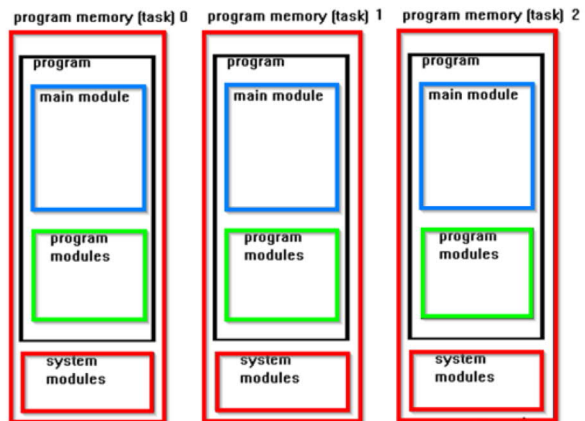
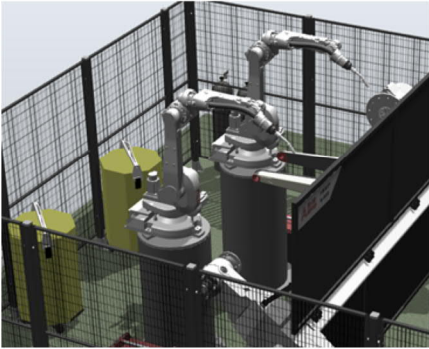
Struktura programu

Temat nr 5

Wersja 01/2019



Materiały są własnością ABB Sp. z o.o. Wszelkie prawa zastrzeżone.



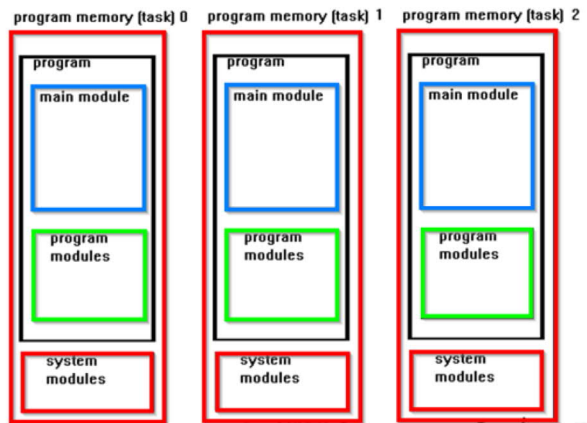
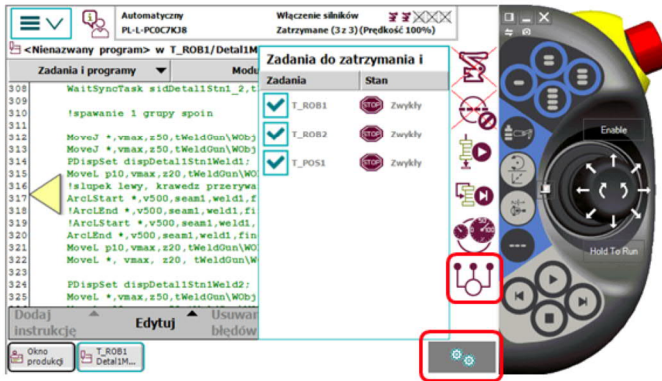
TASK > PROGRAM > MODUŁ > PROCEDURA

Struktura Multimove

Application manuals RobotWare options Multimove



Struktura systemu Multimove stworzona w przypadku gdy jeden kontroler steruje dwoma lub więcej robotami lub manipulatorami tworzy dla każdej jednostki mechanicznej własny Task (Zadanie), w którym znajduje się program, a w nim moduł główny (Main), moduły programowe oraz moduły systemowe. Dostęp do wszystkich tasków jest możliwy z panelu Flexpendant lub przez RobotStudio. Funkcja Multitasking jest niezbędna do tego aby uruchamiać kilka programów jednocześnie na kontrolerze.



TASK > PROGRAM > MODUŁ > PROCEDURA











Struktura Multimove

DVD > Application manuals > RobotWare options > Multimove

Dodatkowa płyta z opcjami systemowymi



W systemach Multimove istnieje możliwość aktywacji lub deaktywacji tasku poprzez menu Quickset. Należy zaznaczyć w nim, które jednostki mechaniczne mają być aktywne w trybie manualnym.

 BMW_FBK_G01_02.mod	2018-01-23 10:50	RAPID module file	40 KB
 BMW_FBK_G05.mod	2018-01-23 10:50	RAPID module file	106 KB
 BMW_FBK_G06.mod	2018-01-23 10:50	RAPID module file	41 KB
 BMW_FBK_G21.mod	2018-01-23 10:50	RAPID module file	18 KB
 JLR_405.mod	2018-01-23 10:50	RAPID module file	151 KB
 MainModule.mod	2018-01-23 10:50	RAPID module file	6 KB
 MFA_2.mod	2018-01-23 10:50	RAPID module file	60 KB
 testR1.mod	2018-01-23 10:50	RAPID module file	6 KB
 VS30_DFNS.mod	2018-01-23 10:50	RAPID module file	16 KB
 Welding.mod	2018-01-23 10:50	RAPID module file	8 KB

PROGRAM > TASK > MODUŁ > PROCEDURA

Struktura programu - moduły

Operating manuals Integrator's guide

Dodatkowa płyta z opcjami systemowymi



Fizycznie moduł jest jednym plikiem o wielkości zwykle do kilkuset KB. Na rysunku widok modułów programowych systemie Windows.

```

1  MODULE·Module1
2  →CONST·robtarget·Target_10:=[800,0,300],[0.000000001,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09];
3  →CONST·robtarget·Target_10_2:=[799.9999998,0,400],[0.000000001,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09];
4  →CONST·robtarget·Target_20_2:=[-499.999999815,-499.999999923,110],[0,0.923879533,0.382683431,-0.000000001],[-2,-1,0,0];
5  →CONST·robtarget·Target_20:=[-500,-500,10],[0,0.923879533,0.382683431,-0.000000001],[-2,-1,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09];
6  →TASK·PERS·tooldata·tool1:=[TRUE,[[0,0,0],[1,0,0,0]],[1,[0,0,0],[1,0,0,0],[1,0,0,0],[1,0,0,0],[1,0,0,0],[1,0,0,0],[1,0,0,0]]];
7  →TASK·PERS·wobjdata·wobj1:=[FALSE,TRUE,"",[[0,0,0],[1,0,0,0]],[[0,0,0],[1,0,0,0]],[[0,0,0],[1,0,0,0]]];
8  .....!*****
9  |
10 →PROC·Path_10()
11 →.....MoveL·Target_10,vmax,fine,tool0\Wobj:=wobj0;
12 →.....MoveL·Target_10_2,vmax,z100,tool0\Wobj:=wobj0;
13 →.....MoveJ·Target_20_2,vmax,z100,tool0\Wobj:=wobj0;
14 →.....MoveL·Target_20,vmax,fine,tool0\Wobj:=wobj0;
15 →.....MoveL·Target_20_2,vmax,z100,tool0\Wobj:=wobj0;
16 →.....MoveJ·Target_10_2,vmax,z100,tool0\Wobj:=wobj0;
17 →.....MoveL·Target_10,vmax,fine,tool0\Wobj:=wobj0;
18 →ENDPROC
19 ENDMODULE

```

PROGRAM > TASK > MODUŁ > PROCEDURA

Struktura programu – moduły – dane programu

Operating manuals Integrator's guide

Dodatkowa płyta z opcjami systemowymi



Moduły można otwierać w programach typu notatnik lub RobotStudio. Moduł może zawierać definicje danych typu tooldata, wobjdata, jointtarget itd. oraz procedury wykonawcze utworzone przez użytkownika. Na rysunku widok danych w module o nazwie Module1 a pod nimi procedura (routine) z instrukcjami ruchu – widok z programu RobotStudio.

```
PROC DetaliStn1()
```

```
MoveJ [[-216.21,-439.04,271.09],[0.0519619,0.559271,0.705551,-0.432105],[0,0,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vmax,z50,tWeldGun\WObj:=wobjDetaliStn1;  
MoveJ [[-471.80,110.53,326.34],[0.252129,-0.910676,-0.265313,0.191598],[0,1,-2,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vmax,z50,tWeldGun\WObj:=wobjDetaliStn1;  
PDispSet dispDetaliStn1Weld1;  
MoveL p10,vmax,z20,tWeldGun\WObj:=wobjDetaliStn1;  
!slupek lewy, krawedz przerywana  
ArcLStart [[-684.01,142.28,-2.93],[0.254283,-0.850762,-0.347326,0.301512],[0,0,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,seam1,weld1,fine,tWeldGun\WObj:=wobjDetaliStn1;  
!ArcLEnd [[-684.06,147.63,-2.92],[0.254285,-0.850767,-0.347316,0.301507],[0,0,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,seam1,weld1,fine,tWeldGun\WObj:=wobjDetaliStn1;  
!ArcLStart [[-685.11,160.50,-2.90],[0.254282,-0.850743,-0.347337,0.301553],[0,0,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,seam1,weld1,fine,tWeldGun\WObj:=wobjDetaliStn1;  
!ArcLEnd [[-685.09,166.32,-3.95],[0.278866,-0.86709,-0.329222,0.249002],[0,0,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,seam1,weld1,fine,tWeldGun\WObj:=wobjDetaliStn1;  
MoveL p10,vmax,z20,tWeldGun\WObj:=wobjDetaliStn1;  
MoveL [[-635.86,66.01,157.67],[0.252141,-0.910688,-0.26527,0.191586],[0,1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vmax,z20,tWeldGun\WObj:=wobjDetaliStn1;
```

```
ENDPROC
```

PROGRAM > TASK > MODUŁ > PROCEDURA

Struktura programu – moduły – procedura

Operating manuals Integrator's guide



Widok procedury w RobotStudio, na zielono wyświetlane są znaki rozpoznawane jako tekst (ciąg znaków lub cyfr poprzedzony wykrzyknikiem jest rozpoznawany jako tekst).


```

MoveJ [[-216.21,-439.04,271.09],[0.0519619,0.559271,0.705551,-0.432105],[0,0,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vmax,z50,tWeldGun\Wobj:=wobjDetail1Stn1;
MoveJ [[-471.80,110.53,326.34],[0.252129,-0.910676,-0.265313,0.191598],[0,1,-2,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],vmax,z50,tWeldGun\Wobj:=wobjDetail1Stn1;
PDispSet dispDetail1Stn1Weld1;
MoveL p10,vmax,z20,tWeldGun\Wobj:=wobjDetail1Stn1;
!slupek lewy, krawedz przerywana
ArcLStart [[-684.01,142.28,-2.93],[0.254283,-0.850762,-0.347326,0.301512],[0,0,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,seam1,weld1,fine,tWeldGun\Wobj:=wobjDetail1Stn1;
!ArcLEnd [[-684.06,147.63,-2.92],[0.254285,-0.850767,-0.347316,0.301507],[0,0,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,seam1,weld1,fine,tWeldGun\Wobj:=wobjDetail1Stn1;
!ArcLStart [[-685.11,160.50,-2.90],[0.254282,-0.850743,-0.347337,0.301553],[0,0,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,seam1,weld1,fine,tWeldGun\Wobj:=wobjDetail1Stn1;
ArcLEnd [[-685.09,166.32,-3.95],[0.278866,-0.86709,-0.329222,0.249002],[0,0,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,seam1,weld1,fine,tWeldGun\Wobj:=wobjDetail1Stn1;
MoveL p10,vmax,z20,tWeldGun\Wobj:=wobjDetail1Stn1;
MoveL [[-635.86,66.01,157.67],[0.252141,-0.910688,-0.26527,0.191586]]

```



PROGRAM > TASK > MODUŁ > PROCEDURA

Struktura programu – moduły – procedura

Operating manuals Integrator's guide



Na rysunku przedstawiono fragment procedury zawierającej instrukcję ruchu i instrukcje spawalnicze w programie RobotStudio oraz na panelu Flexpendant jak widać w RobotStudio część kodu procedury jest w odmiennej kolorystyce aby zwiększyć czytelność kodu Rapid. Instrukcje na panelu Flexpendant są wyświetlane w kolorze zielonym.

Współrzędne punktów zapisywanych podczas tworzenia instrukcji ruchów RobotStudio widoczne są jako:

- współrzędne x,y,z punktu względem określonego układu współrzędnych,
- kwaterniony określające ustawienie narzędzia w przestrzeni,
- Kwaterniony 4 / 6 osi robota,
- Kątowe położenie osi zewnętrznych

W instrukcji ruchu występują po nich argumenty typu tool, wobj, speed itp.

Na panelu Flexpendant aby zmniejszyć liczbę wyświetlanych informacji dane typu robtarget reprezentowane są jako gwiazdka lub nazwa. Wartości ukryte pod gwiazdką można zobaczyć zaznaczając gwiazdkę a następnie naciskając przycisk Debug (usuwanie błędów) następnie View Value (Wyświetl wartość).

Zawartość procedury można przeglądać na Flexpendant przesuwając ekran żółtymi strzałkami, plusem i minusem można zmniejszać lub powiększać czcionkę

▪ MOVEJ ToPoint Speed Zone Tool [\WObj]

instrukcja do szybkiego poruszania robota, gdy nie jest wymagany dokładny ruch liniowy; robot porusza się z narzędziem **Tool** do punktu **ToPoint**, z prędkością **Speed**, a dokładność przejścia trajektorii przez zadany punkt określony jest przez strefę **Zone**,

[\WObj] – opcjonalny argument służący do określenia układu odniesienia, w jakim przemieszczane jest narzędzie; ustawienie domyślne to WObj0

MoveJ *, v1000, z50, tWeldgun\Wobj:=rama1

PROGRAM > TASK > MODUŁ > PROCEDURA

Struktura programu – instrukcje

Technical reference manuals Robot controllers Rapid instructions

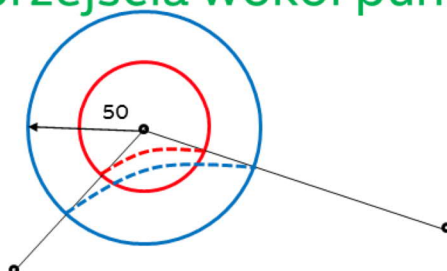


Instrukcja ruchu MoveJ umożliwia przemieszczenie robota z pozycji, w której aktualnie się znajduje do pozycji określonej współrzędnymi punktu robtarget (ToPoint). Robot obliczając ruch zastosuje taką trajektorię przejazdu aby jak najbardziej ograniczyć ruch osi ramienia. Podstawowe argumenty tej instrukcji to:

- Robtarget (ToPoint)
- Prędkość (Speed) v1000 (prędkość ruchu w [mm/s])
- Strefa (Zone) z50 (promień strefy w [mm])
- Narzędzie (Tool) (zdefiniowane narzędzie użytkownika)
- Wobj - argumentem opcjonalnym zawsze wyświetlanym po znaku backslash (\) jest układ współrzędnych, w którym zapisano współrzędne punktu

z50 – strefa przejścia wokół punktu

z0
fine



MoveJ *, v1000, z50, tWeldgun\Wobj:=rama1

PROGRAM > TASK > MODUŁ > PROCEDURA

Struktura programu – instrukcje

Technical reference manuals Robot controllers Rapid instructions

ABB

Strefa przejścia wokół punktu. Na rysunku przedstawiona jest zasada jej funkcjonowania. im większą strefę wokół punktu zdefiniujemy - wartość podawana jest jako promień sfery w milimetrach - tym wcześniej robot zacznie omijać punkt przechodząc do kolejnego. W przypadku strefy z = 0 robot musi dokładniej przejść przez punkt pośredni, w przypadku kiedy strefa Z zdefiniowana jest jako Fine robot musi przejść przez punkt i wyhamować w nim do prędkości 0 [mm/s].

▪ MOVEL ToPoint Speed Zone Tool [\WObj]

instrukcja do szybkiego poruszania robota, gdy jest wymagany dokładny ruch liniowy; robot porusza się z narzędziem **Tool** do punktu **ToPoint**, z prędkością **Speed**, a dokładność przejścia trajektorii przez zadany punkt określony jest przez strefę **Zone**,

[\WObj] – opcjonalny argument służący do określenia układu odniesienia, w jakim przemieszczane jest narzędzie; ustawienie domyślne to WObj0

MoveL *, v1000, z50, tWeldgun\Wobj:=rama1

PROGRAM > TASK > MODUŁ > PROCEDURA

Struktura programu – instrukcje

Technical reference manuals Robot controllers Rapid instructions



Instrukcja ruchu MoveL umożliwia przemieszczenie robota z pozycji, w której aktualnie się znajduje do pozycji określonej współrzędnymi punktu robtarget (ToPoint). Robot obliczając ruch wyznaczy linię prostą przejazdu między punktami i wzdłuż tej linii będzie przesuwiał TCP narzędzia. Ta instrukcja ruchu może okazać się niewłaściwa w przypadku dużych różnic położenia narzędzia.

Podstawowe argumenty tej instrukcji to:

- Robtarget (ToPoint)
- Prędkość (Speed) v1000 (prędkość ruchu w [mm/s])
- Strefa (Zone) z50 (promień strefy w [mm])
- Narzędzie (Tool) (zdefiniowane narzędzie użytkownika)
- Wobj - argumentem opcjonalnym zawsze wyświetlanym po znaku backslash (\) jest układ współrzędnych, w którym zapisano współrzędne punktu

- **MOVEABSJ ToJointPos Speed Zone Tool [\WObj]**

instrukcja do przemieszczenia robota do pozycji **ToJointPos** zdefiniowanej kątami ustawienia osi, z prędkością **Speed** i dokładnością przejścia trajektorii przez zadany punkt określony przez strefę **Zone**;

aktywne narzędzie **Tool** i układ odniesienia **WObj** nie mają wpływu na wykonanie instrukcji, a służą jedynie do obliczania bezwładności, prędkości TCP, warunków hamowania oraz trajektorii narożnikowych;

instrukcja stosowana do wyjątkowych punktów np.: pozycja home czy punkt końcowy, będący punktem osobliwym, itp.

- np.: `MOVEABSJ p10, v100, z10, tool1 \ WObj;`

TASK > PROGRAM > MODUŁ > PROCEDURA

Struktura programu – instrukcje

Technical reference manuals Robot controllers Rapid instructions



MoveAbsJ to instrukcja, w której występuje argument jointtarget - przechowuje on dane o położeniu kątowym osi robota.

Uwaga: robtargt przechowuje współrzędne punktu, a jointtarget wartości kątowe osi.

- **MOVEC CirPoint ToPoint Speed Zone Tool[\WObj]**

instrukcja do poruszania robota ruchem po łuku; argumenty j.w., przy czym od razu programowane są dwa punkty łuku – pośredni i końcowy

MoveC **, v1000, z50, tWeldgun\Wobj:=rama1

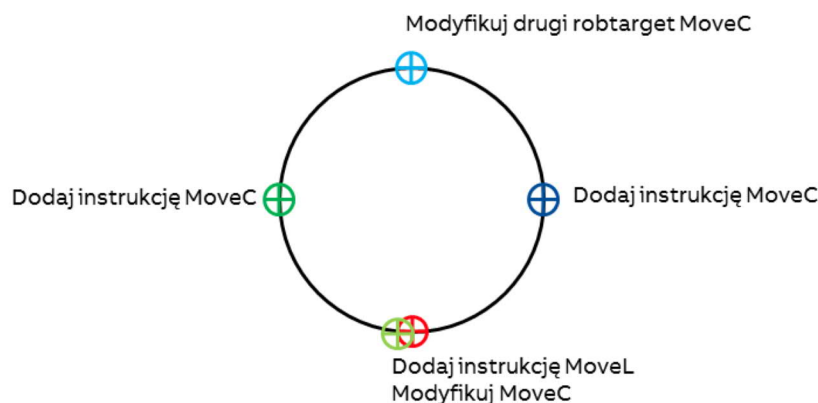
PROGRAM > TASK > MODUŁ > PROCEDURA

Struktura programu – instrukcje

Technical reference manuals Robot controllers Rapid instructions



Instrukcja MoveC umożliwia robotowi poruszanie się po łuku. Instrukcja ta posiada dwa argumenty robtarget. Istotne jest, że tylko drugi punkt przechowuje dane o kącie położenia narzędzia.



MoveC **, v1000, z50, tWeldgun\Wobj:=rama1

PROGRAM > TASK > MODUŁ > PROCEDURA

Struktura programu – instrukcje

Technical reference manuals Robot controllers Rapid instructions



Przykład opisuje jak utworzyć instrukcje w których narzędzie poruszać się będzie po okręgu.

- Ustaw TCP narzędzia w miejscu czerwonego znacznika i dodaj instrukcję MoveL.
- Przesuń TCP narzędzia robota do znacznika ciemnoniebieskiego i dodaj instrukcję MoveC – pierwszy robtarget zostanie zapisany i kursor podświetli drugą gwiazdkę.
- Przesuń TCP do znacznika jasnoniebieskiego i naciśnij przycisk Modify Position (Modyfikuj pozycję) potwierdź OK – w ten sposób zapisano pozycję drugiej gwiazdki.
- Przesuń TCP robota do znacznika ciemnozielonego i dodaj instrukcję MoveC – pierwszy robtarget zostanie zapisany i kursor podświetli drugą gwiazdkę.
- Przesuń TCP robota do znacznika jasnozielonego i naciśnij przycisk Modify Position (Modyfikuj pozycję) potwierdź OK – w ten sposób zapisano pozycję drugiej gwiazdki a robot zamknie koło.