AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Industrial Robots

Robot programming I

Programming of Mitsubishi robots in MELFA BASIC IV language – part I

Dr inż. Jarosław Bednarz

Faculty of Mechanical Engineering and Robotics
Department of Robotics and Mechatronics

Topics

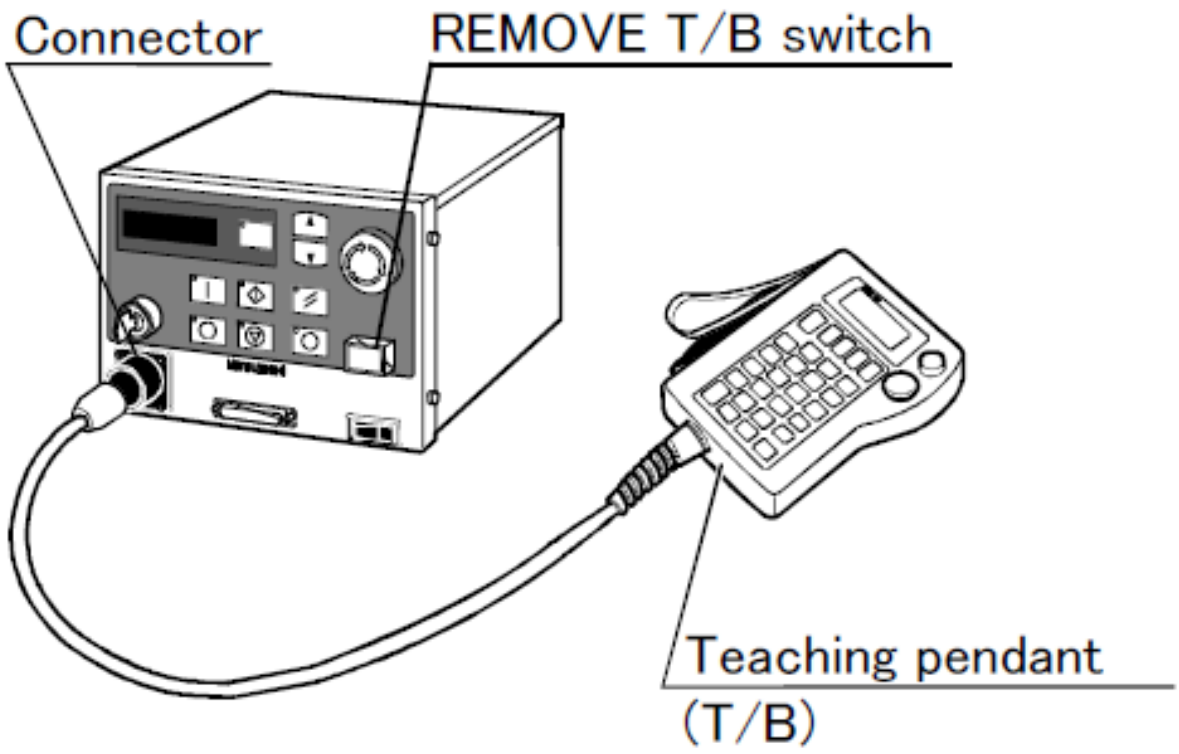
1. Technical data of Mitsubishi Melfa RV-2AJ robot
2. Working with robot teaching pendant
3. Programming of robot in Melfa Basic IV language

Technical data of Mitsubishi Melfa RV-2AJ robot

Model			RV-2AJ
No of axes			5
Installation posture			On floor, Hanging
Structure			Vertical, Multiple-joint type
Drive system			AC servo motor
Motion range	J1	Degree	300 (-150 to +150)
	J2		180 (-60 to +120)
	J3		230 (-110 to +120)
	J4		-
	J5		180 (-90 to +90)
	J6		400 (-200 to +200)
Speed of motion	J1	Degree/s	180
	J2		90
	J3		135
	J4		-
	J5		180
	J6		210
Maximum linear velocity		mm/s	2100
Load capacity	Maximum	kg	2
	Rating		1.5
Position repeatability		mm	±0.02
Mass		kg	17
Arm reachable radius		mm	410

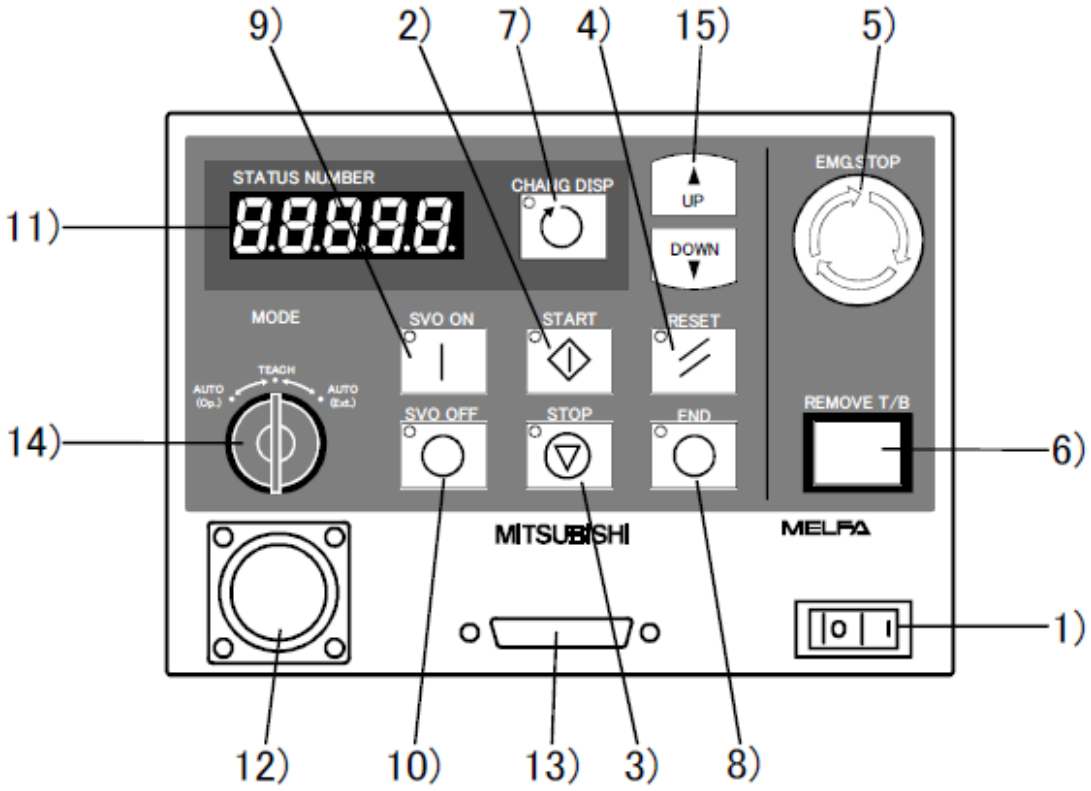
Technical data of Mitsubishi Melfa RV-2AJ robot

Robot controler CR – 1 – teaching pendant (T/B)



Technical data of Mitsubishi Melfa RV-2AJ robot

Robot controler CR - 1



1) POWER switch

2) START button

3) STOP button

4) RESET

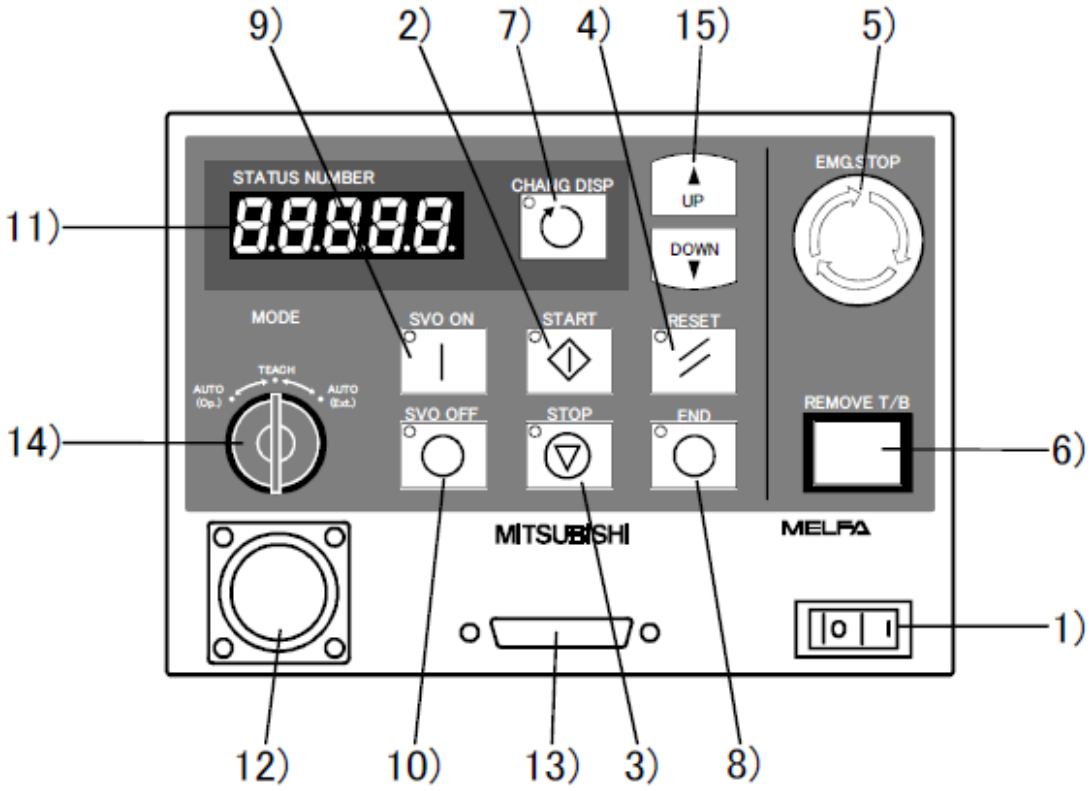
5) Emergency stop switch

6) T/B remove switch

7) CHNGDISP button

Technical data of Mitsubishi Melfa RV-2AJ robot

Robot controler CR - 1



8) END button.

9) SVO.ON button

10) SVO.OFF button

11) STATUS.NUMBER display

12) T/B connection

13) Personal computer connection connector (RS232)

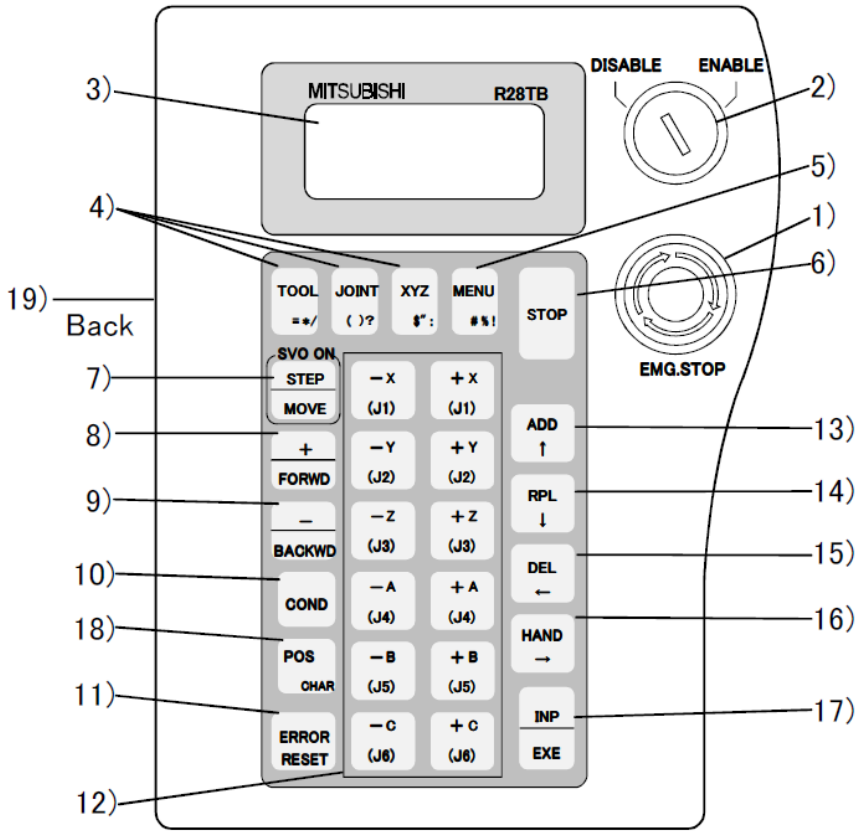
14) MODE changeover

- •AUTO(Op.)
- •TEACH
- •AUTO (Ext.)

15) UP/DOWN

Technical data of Mitsubishi Melfa RV-2AJ robot

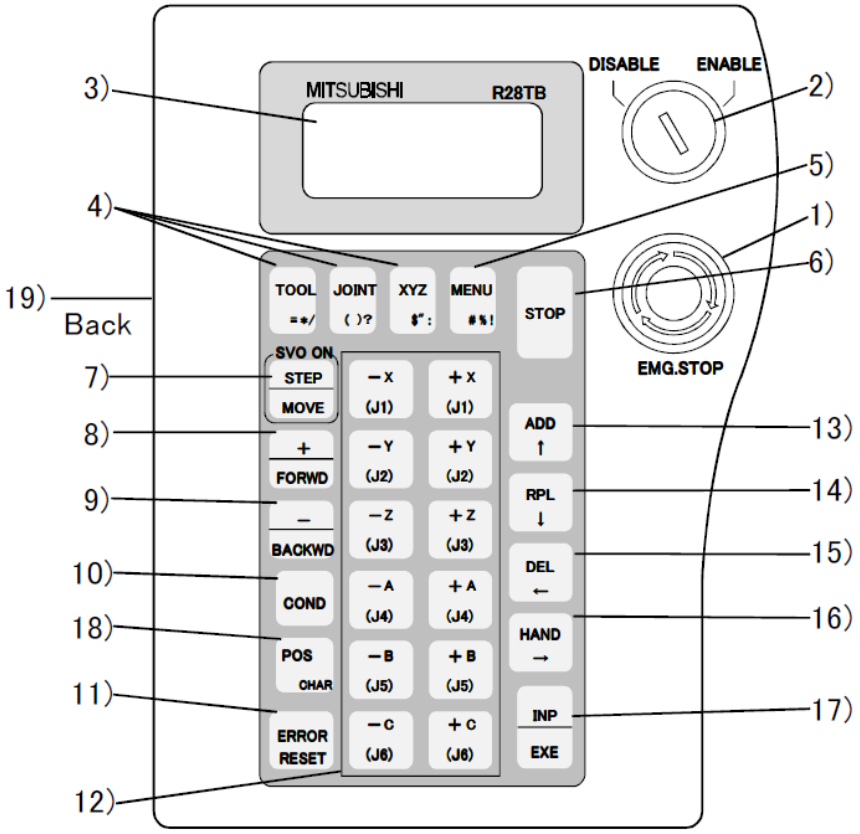
Robot controler CR – 1 – teaching pendant



1) [EMG. STOP] switch
2) [ENABLE/DISABLE] switch
3) Display LCD
4) [TOOL] key
4) [JOINT] key
4) [XYZ] key
5) [MENU] key
6) [STOP] key
7) [STEP/MOVE] key
8) [+/FORWD] key
9) [-/BACKWD] key

Technical data of Mitsubishi Melfa RV-2AJ robot

Robot controler CR – 1 – teaching pendant



10) [COND] key

11) [ERROR RESET]

12) [Jog operation] key - 12 keys from [-X(J1)] to [+C (J6)]

13) [ADD/ ↑] key

14) [RPL/ ↓] key

15) [DEL/ ←] key

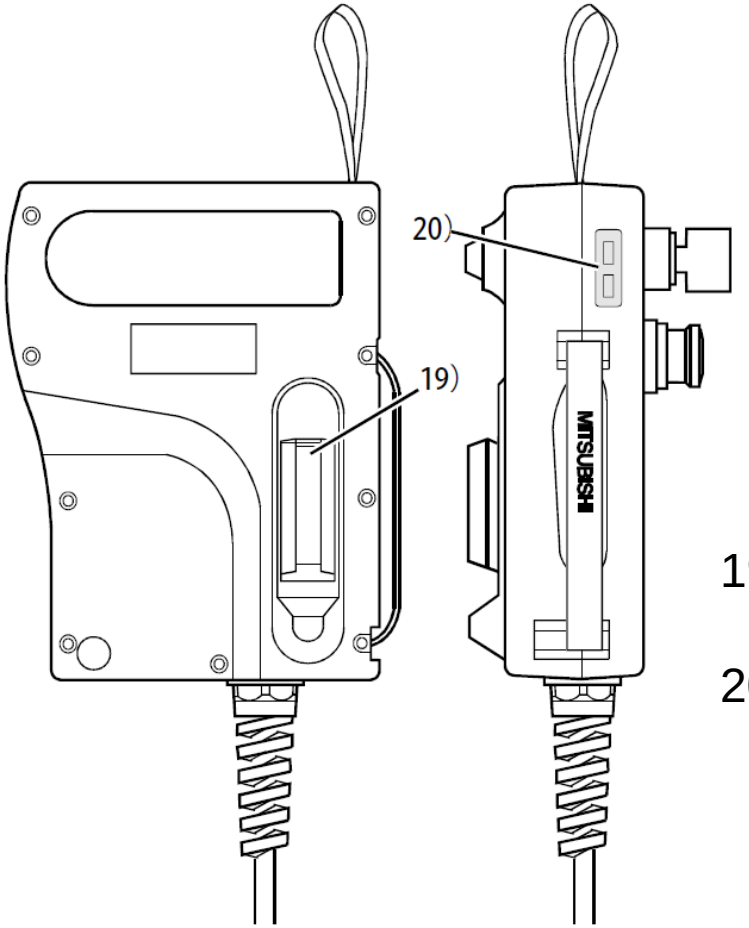
16) [HAND/ →] key

17) [INP/EXE] key

18) [POS CHAR]

Technical data of Mitsubishi Melfa RV-2AJ robot

Robot controler CR – 1 – teaching pendant



19) Deadman switch

20) Contrast setting switch

Working with robot teaching pendant

Moving the robot manually is called "jog operation".

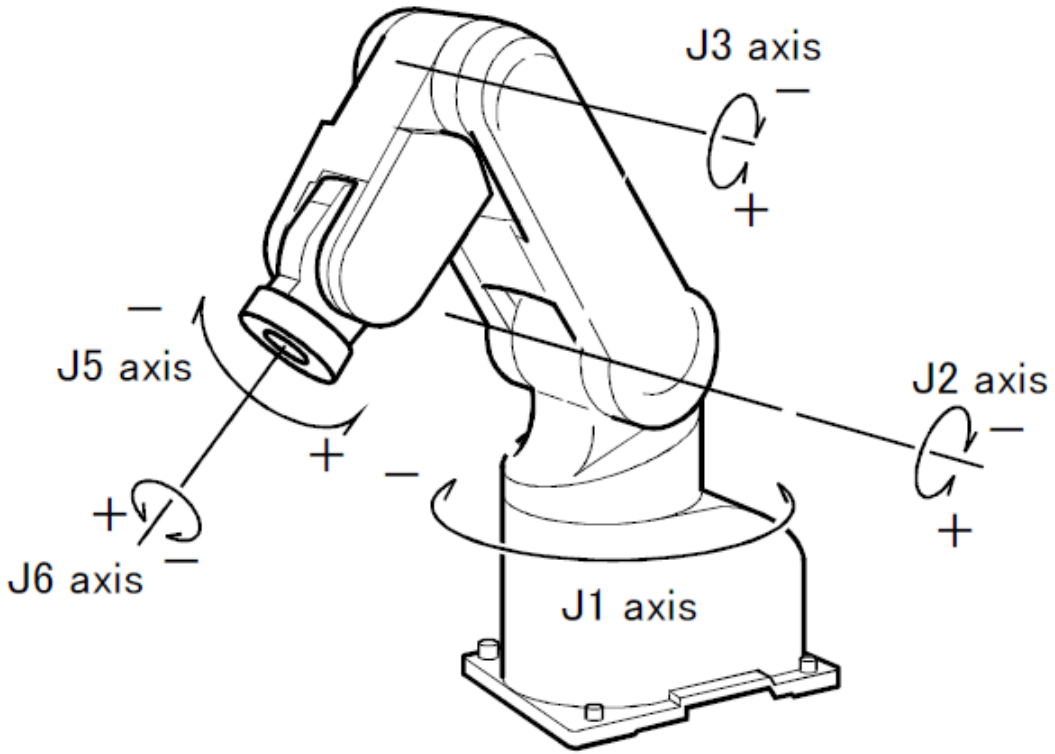
This operation includes:

1. JOINT jog that moves each axis independently,
2. XYZ jog that moves along the base coordinate system,
3. TOOL jog that moves along the tool coordinate system,
4. CYLINDER jog that moves along the circular arc.

This operation is carried out while pressing the dead-man switch on the back of the T/B.

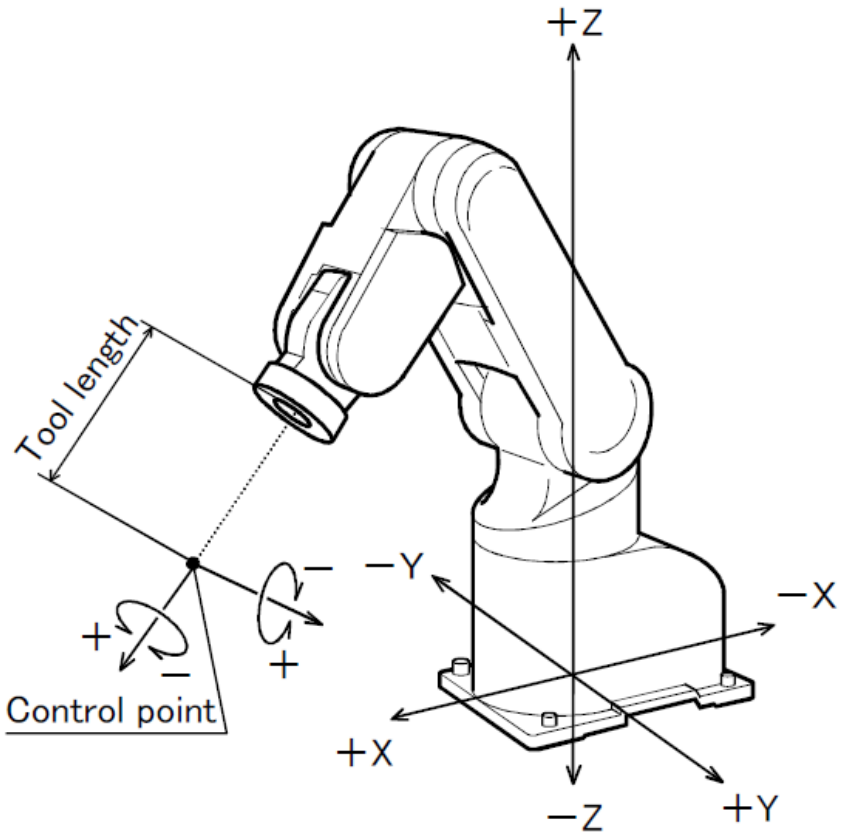
Working with robot teaching pendant

JOINT jog operation



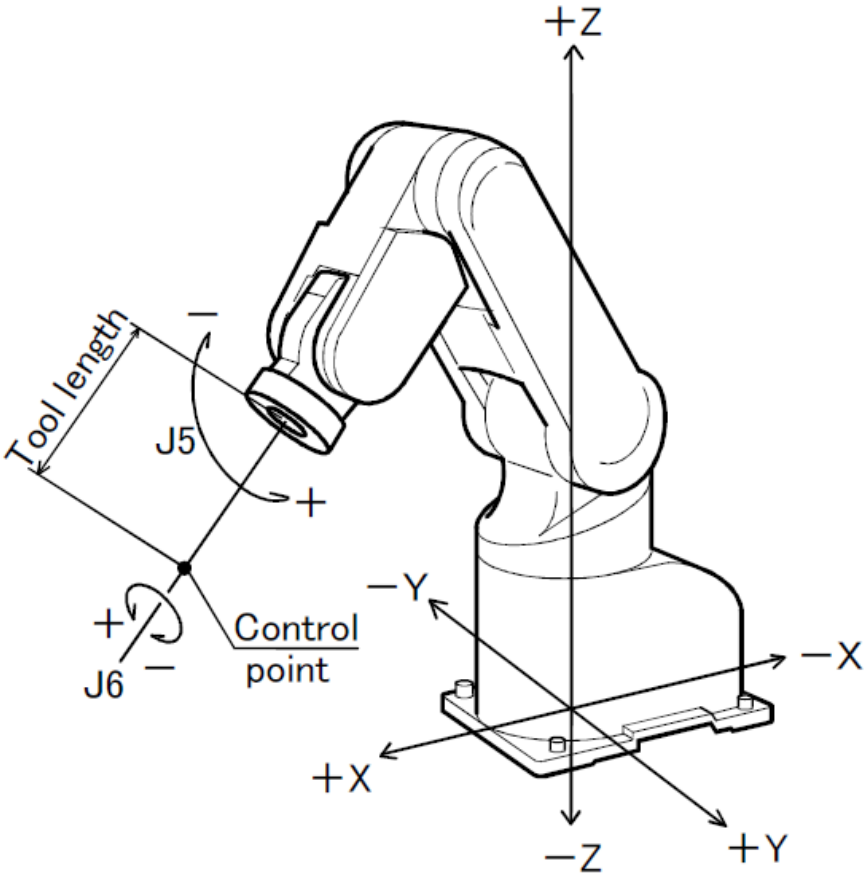
Working with robot teaching pendant

XYZ jog operation



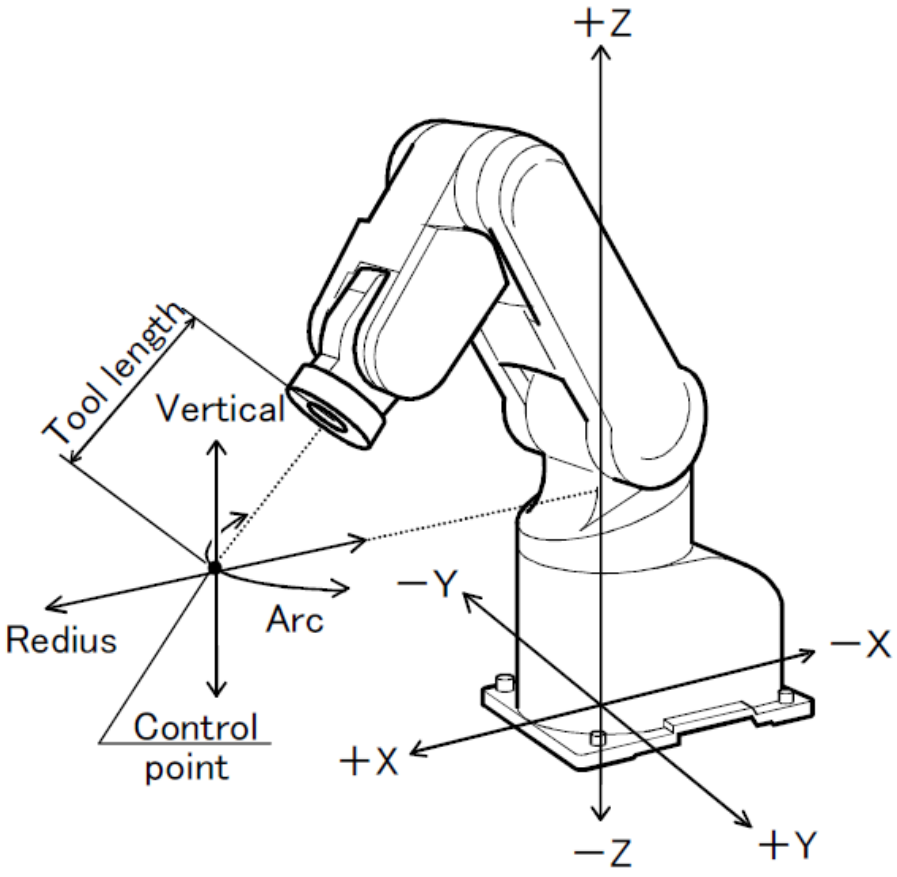
Working with robot teaching pendant

3-axis XYZ jog operation



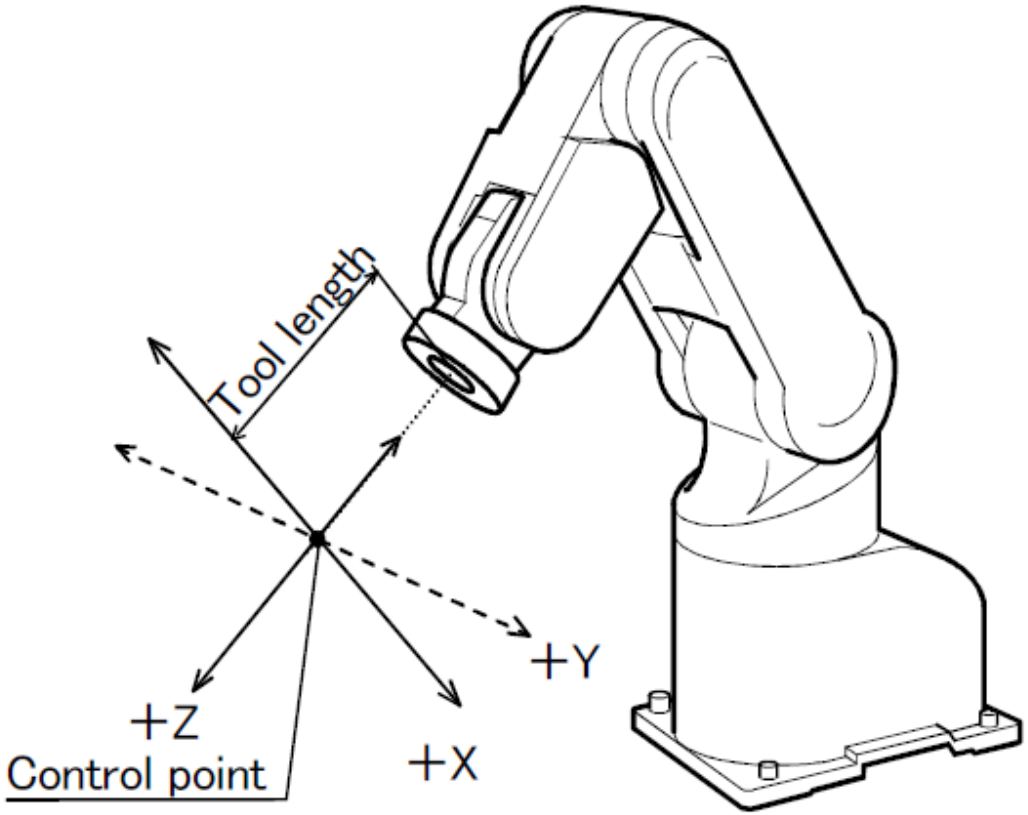
Working with robot teaching pendant

CYLINDER jog operation



Working with robot teaching pendant

TOOL jog operation



MELFA BASIC IV programming language

1. Melfa Basic IV is a programming language designed specifically for Mitsubishi robots.
2. Using this language, you can directly program the robot motion, as well as perform many special features such as he calculation.
3. The structure of Melfa Basic IV is very similar to the well-known for many years, a simple programming language that is BASIC.
4. The number of features of both languages is similar.
5. RT Toolbox is a programming environment for all Mitsubishi robots.

MELFA BASIC IV programming language

1. The program name can contain up to 12 characters. It is recommended to use the name up to 4 characters, due to the limitations of manual control panel (the program with the name longer than 4 characters can not be started manually via the control panel).
2. If the program is invoked using CALLP, his name may consist of more than 4 characters.
3. The program name can consist of uppercase letters and numbers.
4. If you use an external output signal for selecting the program, the program name must consist only of a numbers.

MELFA BASIC IV programming language

The following types of variables can be used in a program:

1. Position type variable - the robot's orthogonal coordinate value is saved. The variable name starts with "P“. (Example - MOV P1: The robot moves to the position saved in variable name P1)

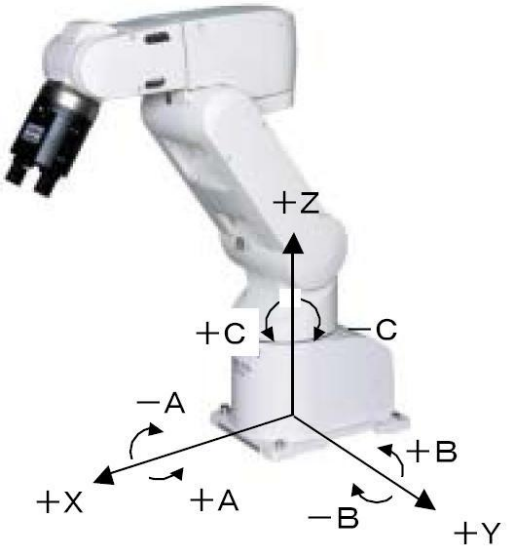
2. Joint type variable - the robot's joint angle is saved. The variable name starts with "J“. (Example - MOV J1: The robot moves to the position saved in variable name J1)

3. Numeric value type variable - a numeric value (integer, real value, etc.) is saved. The variable name starts with "M". (Example - M1 = 1: The value 1 is substituted in variable name M1)

4. Character type variable - a character string is saved. A "$" is added to the end of the variable name. (Example - C1$ = "ERROR": the character string "ERROR" is substituted in variable name C1$)

MELFA BASIC IV programming language

The syntax for position constants is as shown below.
(X, Y, Z, A, B, C, L1,L2)



(100, 100, 300, 180, 0, 180, 0, 0) (7, 0)

- structure flag 2 (multi-rotation data)
- structure flag 1 (posture data)
- L2 axis (additional axis 2)
- L1 axis (additional axis 1)
- C axis
- B axis — Posture axes of the robot (degree)
- A axis
- Z axis
- Y axis — Coordinate values of the hand tip (mm)
- X axis

MELFA BASIC IV programming language

Coordinate, posture and additional axis data types and meanings:

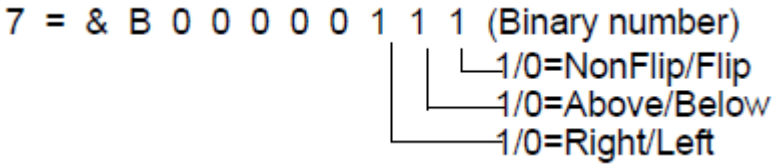
1. Format - X, Y, Z, A, B, C , L1, L2
2. Meaning:
   - X, Y, Z: coordinate data - the position of the tip of the robot's hand in the XYZ coordinates. (The unit is mm.)
   - A, B, C: posture data - this is the angle of the posture. (The unit is deg.)
   - L1, L2: additional axis data - here are the coordinates for additional axis 1 and additional axis 2, respectively. (The unit is mm or deg.)

MELFA BASIC IV programming language

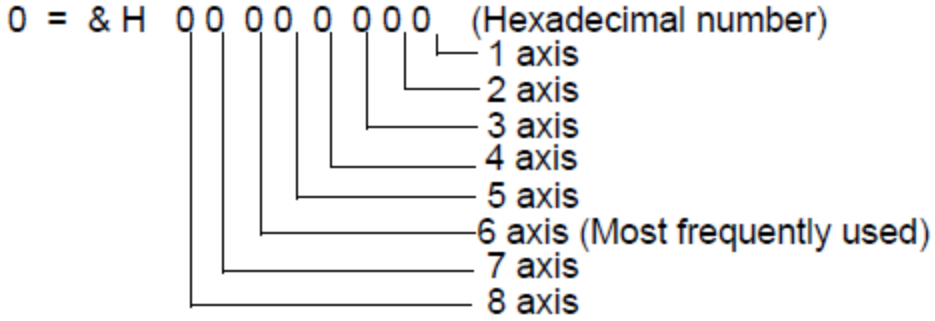
Meaning of structure flag data type:

1. Format - FL1, FL2
2. Meaning:

FL1: Posture data. It indicates the robot arm posture in the XYZ coordinates.

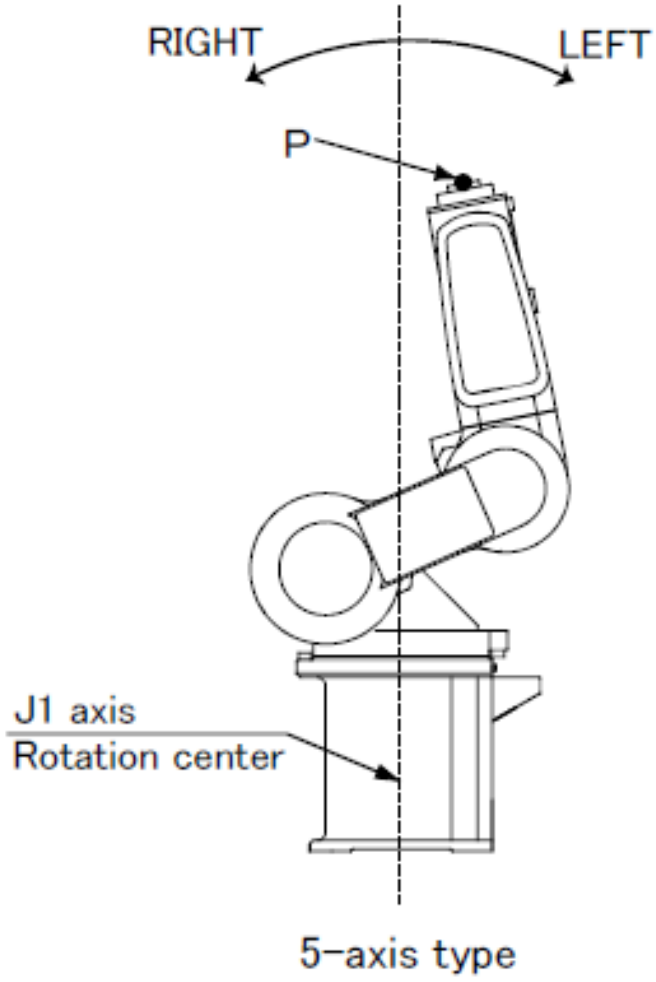


FL2: Multiple rotation data. It includes information of the rotational angle of each joint axis at the position (XYZ) and posture (ABC) expressed as XYZ coordinates.

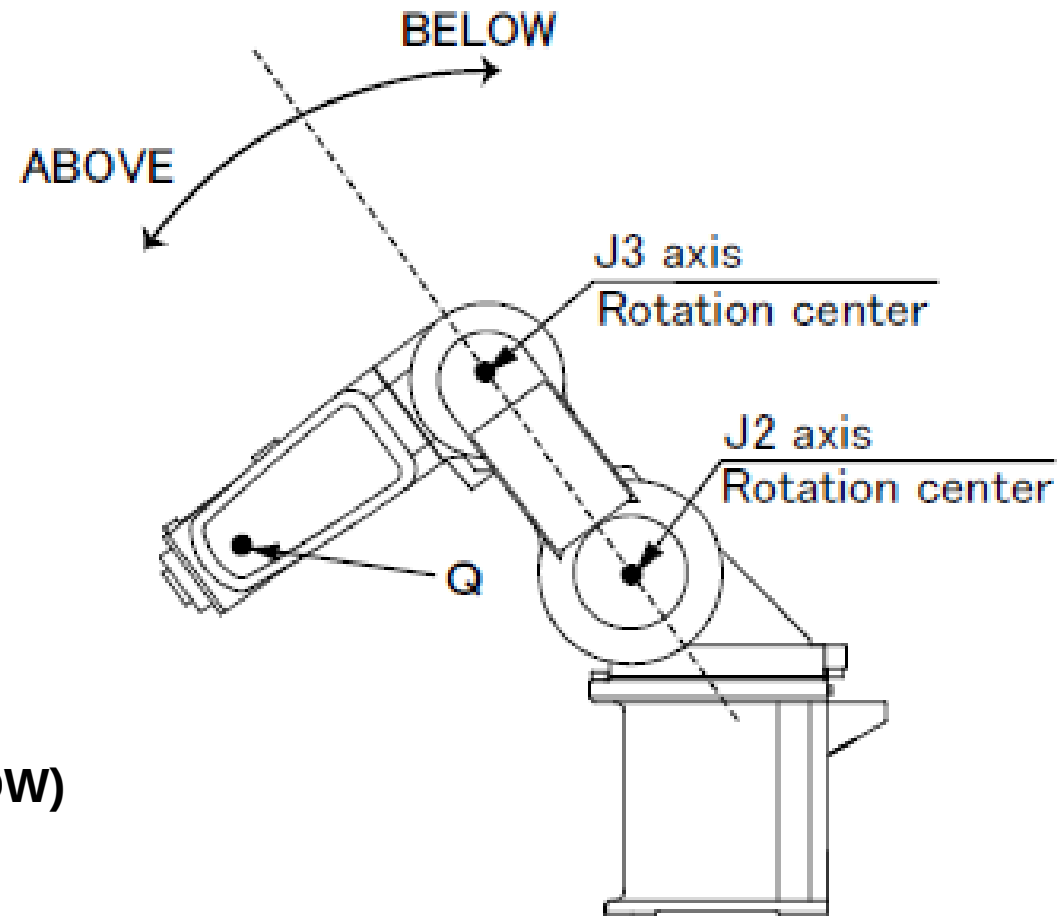


MELFA BASIC IV programming language

Configuration flag (RIGHT/LEFT)

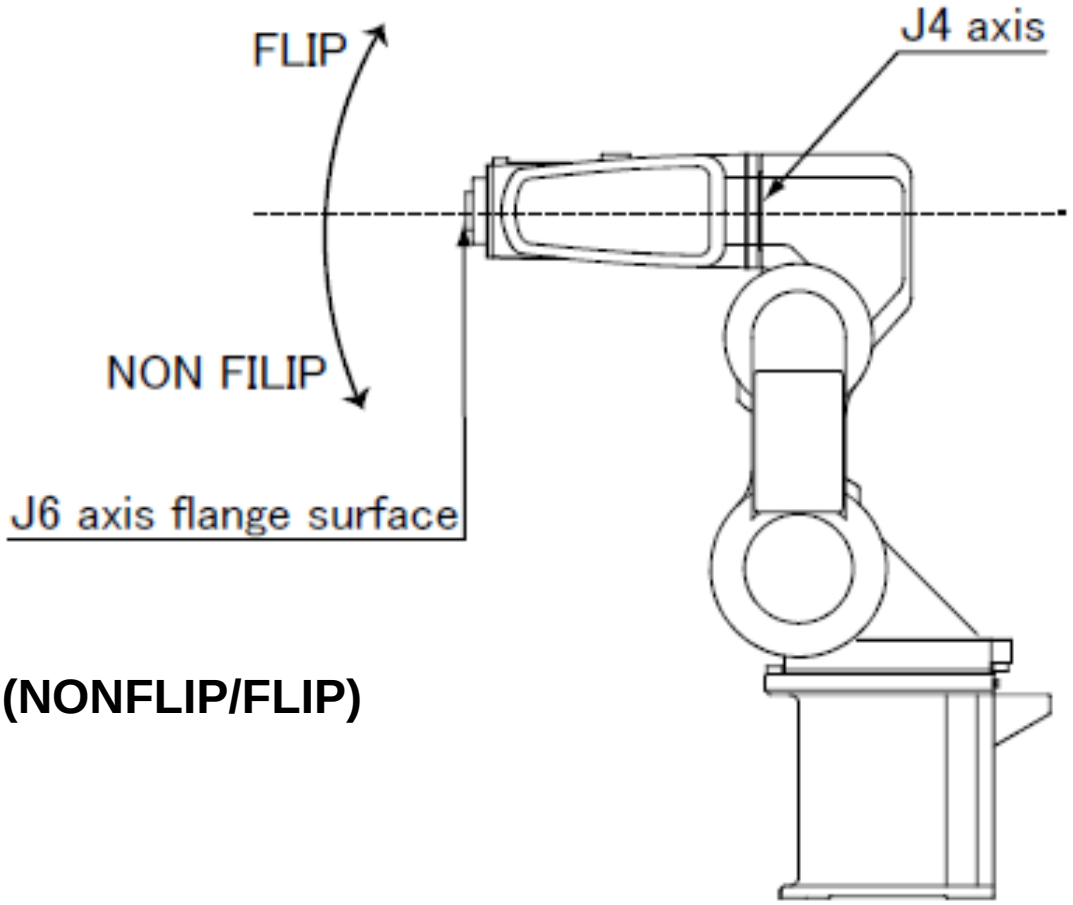


MELFA BASIC IV programming language



Configuration flag (ABOVE/BELOW)

MELFA BASIC IV programming language



Configuration flag (NONFLIP/FLIP)
(6-axis robot only.)

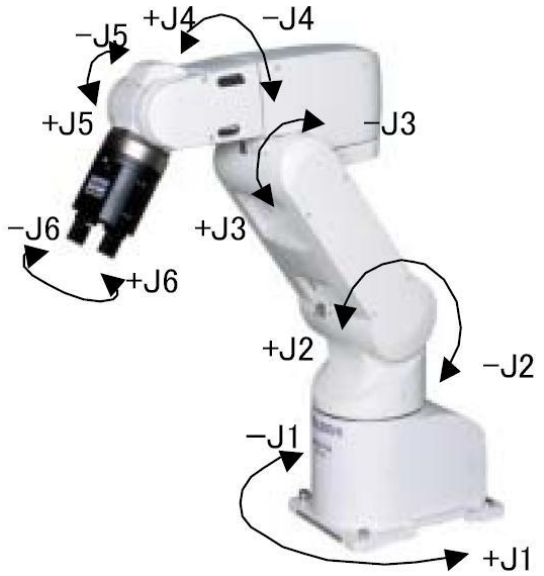
MELFA BASIC IV programming language

Position variables:

1. Variables whose names begin with character P are considered position variables. If it is defined by the DEF POS instruction, it is possible to specify a name beginning with a character other than P.
2. It is possible to reference individual coordinate data of position variables. In this case, add "." and the name of a coordinate axis, e.g. "X," after the variable name - P1.X, P1.Y, P1.Z, P1.A, P1.B P1.C, P1.L1, P1.L2
3. The unit of the angular coordinate axes A, B, and C is radians. Use the DEG function to convert it to degrees (M2 = DEG(P1. A) (Unit : degree))

MELFA BASIC IV programming language

The syntax for the joint constants is as shown below



(10, −20, 90, 0, 90,. 0, 0, 0)

- J8 axis (additional axis 2)
- J7 axis (additional axis 1)
- J6 axis
- J5 axis
- J4 axis
- J3 axis
- J2 axis
- J1 axis

MELFA BASIC IV programming language

Joint variables:

1. A character string variable should start with J. If it is defined by the DEF JNT instruction, it is possible to specify a name beginning with a character other than J.
2. It is possible to reference individual coordinate data of joint variables. In this case, add "." and the name of a coordinate axis, e.g. "J1," after the variable name - JDATA.J1, JDATA.J2, JDATA.J3, JDATA.J4, JDATA.J5, JDATA.J6, JDATA.J7, JDATA.J8
3. The unit of the angular coordinate axes A, B, and C is radians. Use the DEG function to convert it to degrees.

MELFA BASIC IV programming language

Example of constructing a command statement:

10 MOV P1 WTH M_OUT(17)=1

1. **Line No - Numbers for determining the order of execution within the program.**

   Lines are executed in ascending order.

2. **Command word - Instructions for specifying the robot's movement and tasks.**

3. **Data - Variables and numerical data necessary for each instruction.**

4. **Appended statement - Specify these as necessary when adding robot tasks.**

MOVEMENT COMMANDS

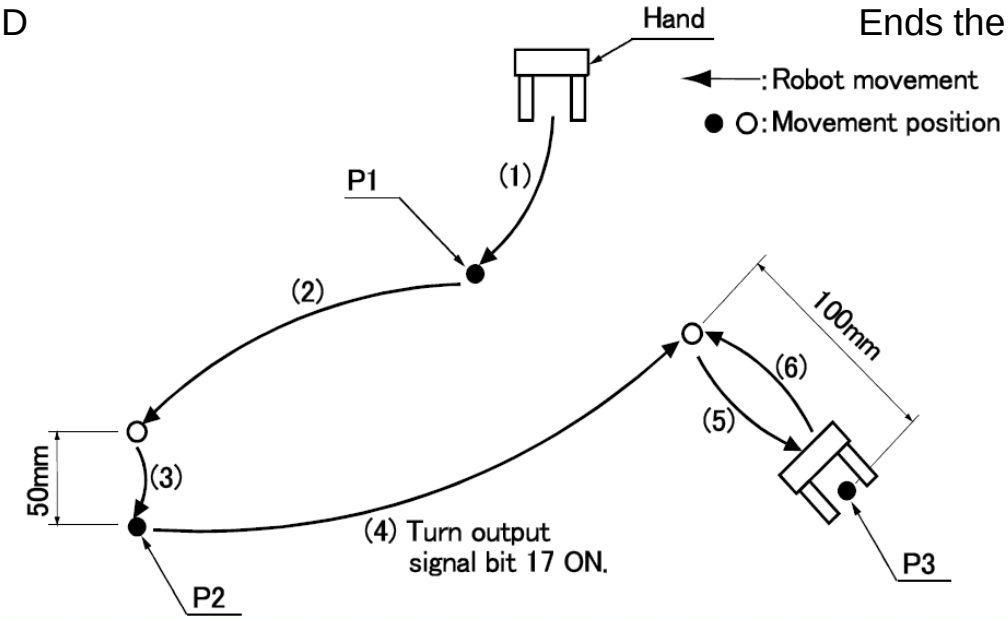
MELFA BASIC IV programming language

Joint interpolation movement - MOV - The robot moves to the designated position with joint interpolation. An appended statement WTH or WTHIF can be designated.

1. **MOV P1** - Moves to P1.
2. **MOV P1+P2** - Moves to the position obtained by adding the P1 and P2 coordinate elements.
3. **MOV P1*P2** - Moves to the position relatively converted from P1 to P2.
4. **MOV P1,-50** - Moves from P1 to a position retracted 50mm in the hand direction.
5. **MOV P1 WTH M_OUT(17)=1** - Starts movement toward P1, and simultaneously turns output signal bit 17 ON.
6. **MOV P1 WTHIF M_IN(20)=1, SKIP** - If the input signal bit 20 turns ON during movement to P1, the movement to P1 is stopped, and the program proceeds to the next stop.

MELFA BASIC IV programming language

10 MOV P1	(1) Moves to P1.
20 MOV P2, -50	(2) Moves from P2 to a position retracted 50mm in the hand direction.
30 MOV P2	(3) Moves to P2
40 MOV P3, -100 WTH M_OUT (17) = 1	(4) Starts movement from P3 to a position retracted 100mm in the hand direction, and turns ON output signal bit 17.
50 MOV P3	(5) Moves to P3
60 MOV P3, -100	(6) Returns from P3 to a position retracted 100mm in the hand direction.
70 END	Ends the program.



MELFA BASIC IV programming language

Linear interpolation movement - MVS - the robot moves to the designated position with linear interpolation. It is possible to specify the interpolation form using the TYPE instruction. An appended statement WTH or WTHIF can be designated.

1. **MVS P1** - Moves to P1
2. **MVS P1+P2** - Moves to the position obtained by adding the P1 and P2 coordinate elements.
3. **MVS P1*P2** - Moves to the position relatively converted from P1 to P2.
4. **MVS P1, -50** - Moves from P1 to a position retracted 50mm in the hand direction.
5. **MVS ,-50** - Moves from the current position to a position retracted 50mm in the hand direction.
6. **MVS P1 WTH M_OUT(17)=1** - Starts movement toward P1, and simultaneously turns output signal bit 17 ON.
7. **MVS P1 WTHIF M_IN(20)=1, SKIP** - If the input signal bit 20 turns ON during movement to P1, the movement to P1 is stopped, and the program proceeds to the next stop.

MELFA BASIC IV programming language

10 MVS P1, -50	(1) Moves with linear interpolation from P1 to a position retracted 50mm in the hand direction.
20 MVS P1	(2) Moves to P1 with linear interpolation.
30 MVS ,-50	(3) Moves with linear interpolation from the current position (P1) to a position retracted 50mm in the hand direction.
40 MVS P2, -100 WTH M_OUT(17)=1	(4) Output signal bit 17 is turned on at the same time as the robot starts moving.
50 MVS P2	(5) Moves with linear interpolation to P2.
60 MVS , -100	(6) Moves with linear interpolation from the current position (P2) to a position retracted 50mm in the hand direction.