

CIROS Programming Basics Melfa-Basic V [EN]

Version: v1.0 (26.01.2013)

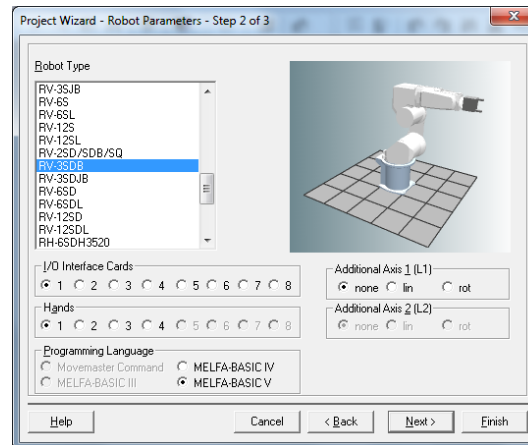
Daniel Bolla – FESTO DC-CC



Table of Contents

- 1. Project Wizard
- 2. Windows/Tools
- 3. Connection to the Robot
- 4. Melfa-Basic V Basics
- 5. Download and Upload
- 6. Further Presentations in this Topic

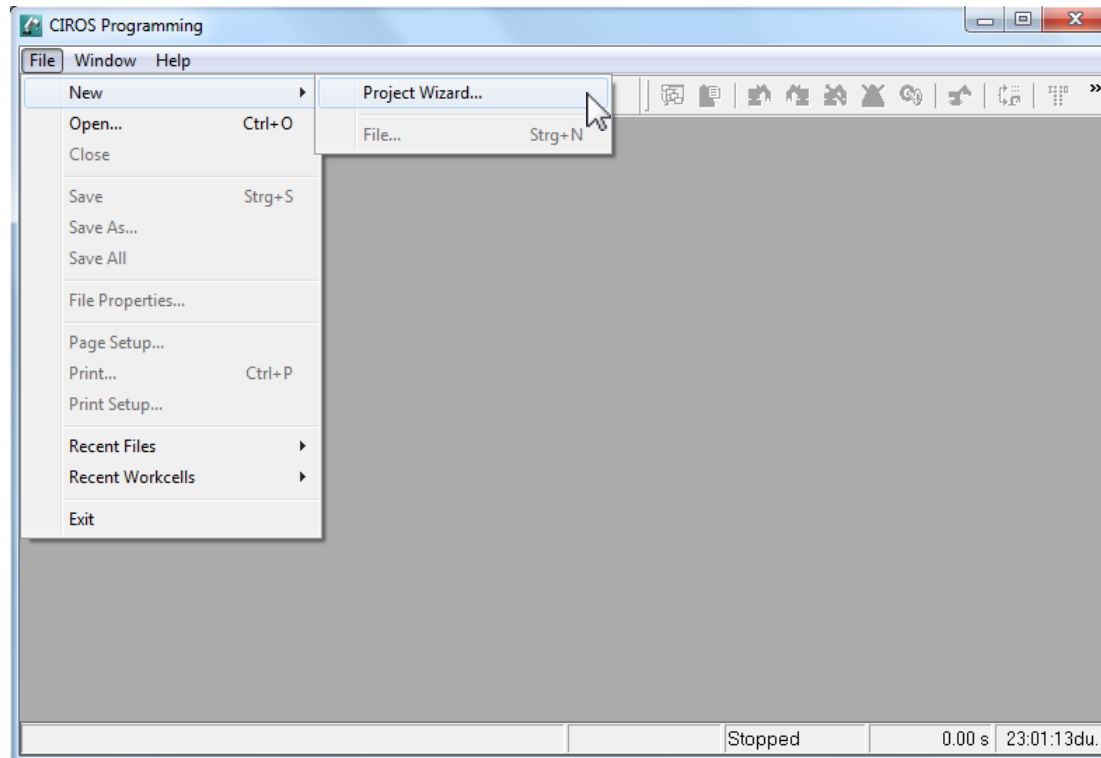
1. Project Wizard



➔ Skip over

Start the Project Wizard

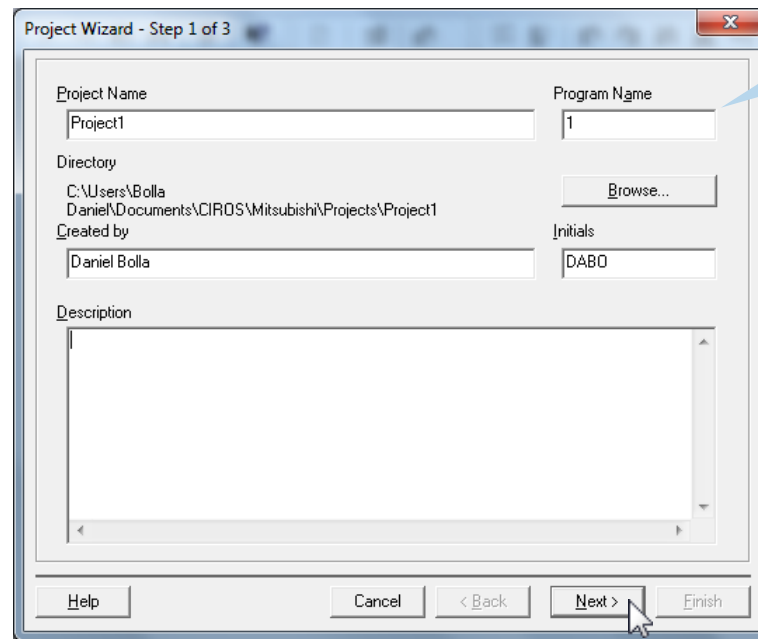
- File/New/Project Wizard...



Give a name to the project

- And then press the Next> button

The name of the robot program.
Should be maximum 3 characters long, because of the robot controllers display.



Select the robot type

- And than press the Next> button

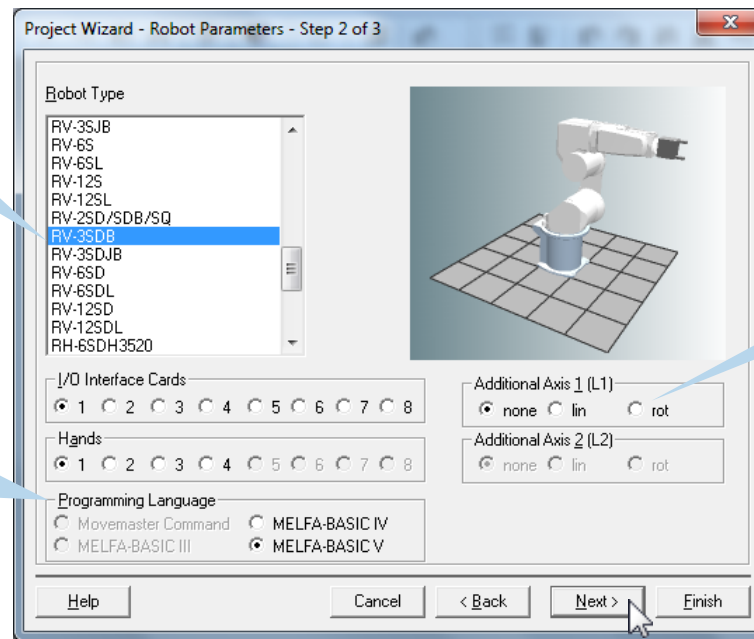
You can select the robot type here.

There is no RV-2SDB. If you have RV-2SDB, select RV-3SDB.

The newer robots use the MELFA-BASIC V.

The older ones use just the MELFA-BASIC IV

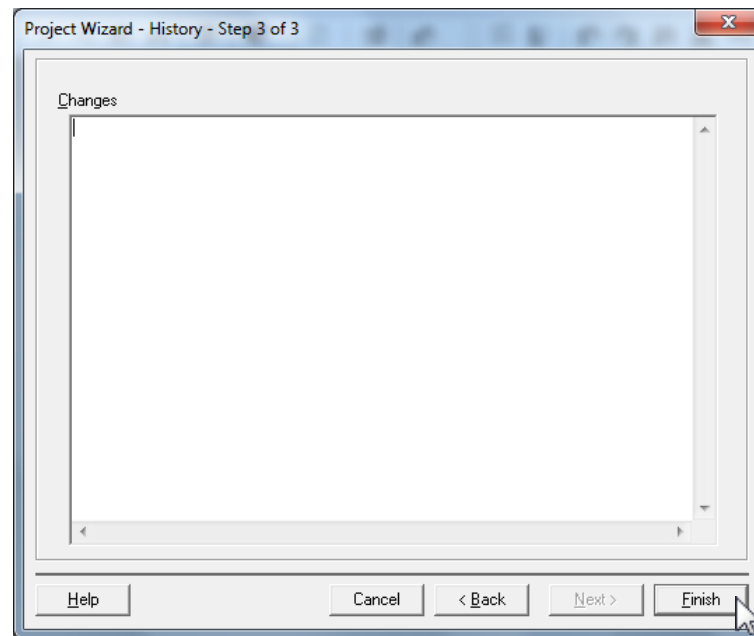
If it is possible for your robot, select the MELFA-BASIC V.



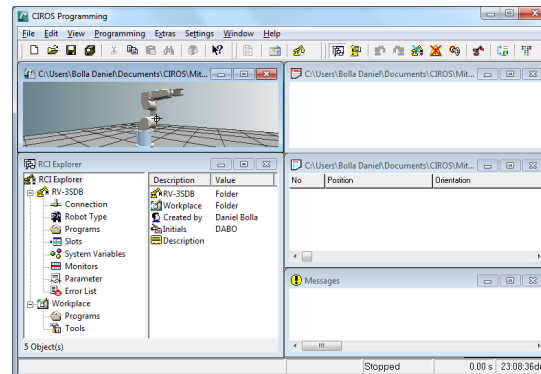
If you have an additional axis. For example at FMS.

Write down the changes here

- And than press the Finish button



2. Windows/Tools

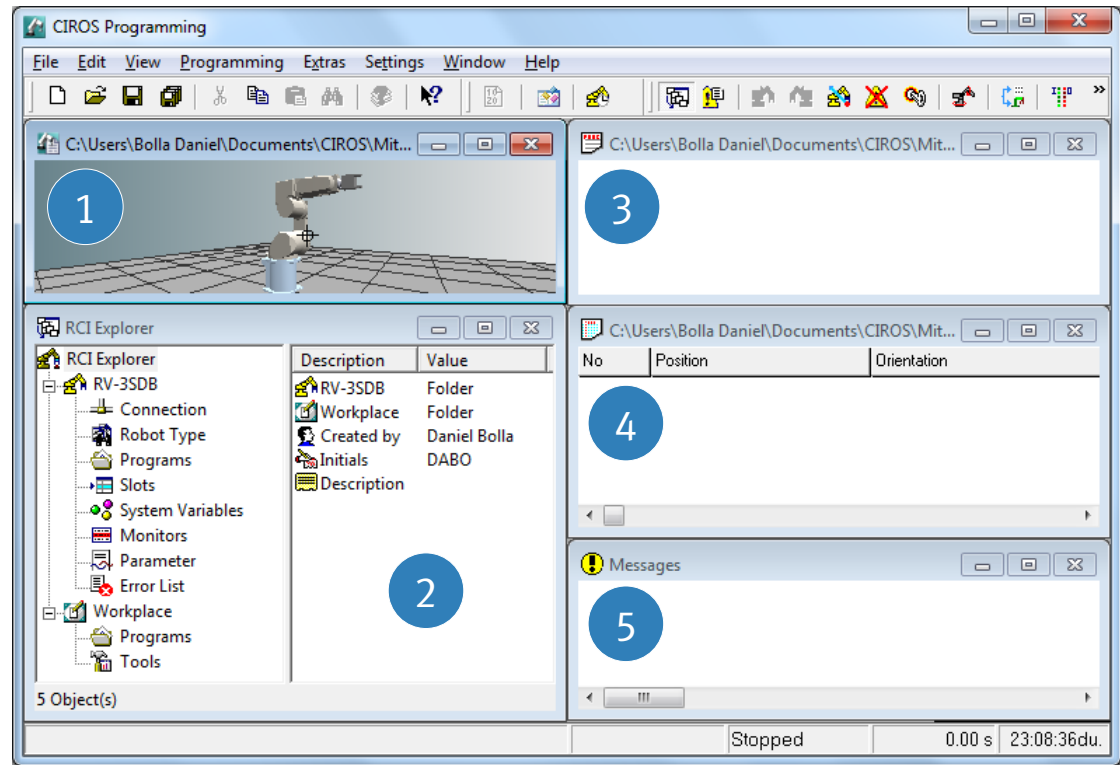


Virtual Robot Window (1)

- Check the positions on a virtual robot

RCI Explorer (2)

- Project tree
- Contents of the robot
 - Programs
 - Program slots
 - Variables
 - Parameters
 - ...
- Contents of the Workplace
 - Programs
 - Position lists



Program Editor (3)

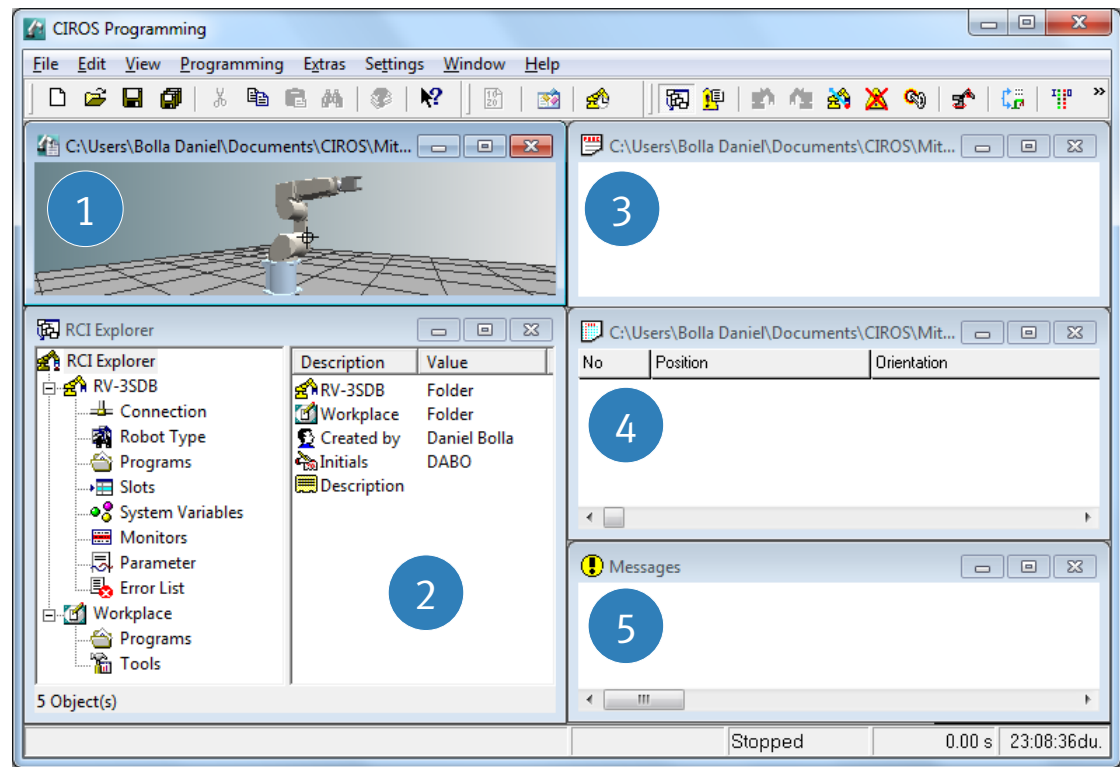
- You can write the program here

Position List (4)

- You can edit the positions here
 - add new positions
 - modify positions
 - remove positions

Messages (5)

- Masseges
- Warnings
- Errors

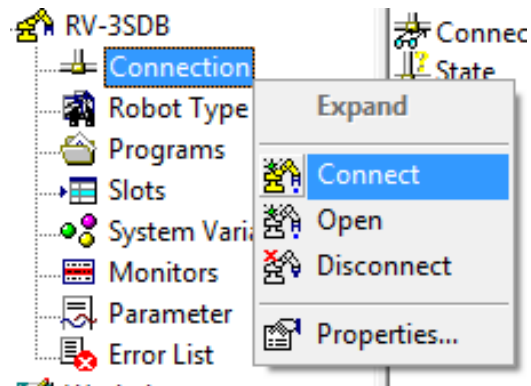


Tools

- Show/Hide RCI Explorer (1)
- Display the current error message of the robot (2)
- Download the program/position list to the robot (3)
- Upload the program and the position list from the robot (4)
- Read the actual position from the real robot to the virtual robot (5)
- Reset the errors (6)
- Initialize the connection to the robot (7)
- Open the JOG operation panel to move the robot from PC (8)
- Open the Command Tool to send commands directly to the robot (9)
- Open the I/O Monitor the check the I/O-s, or switch the outputs manually (10)

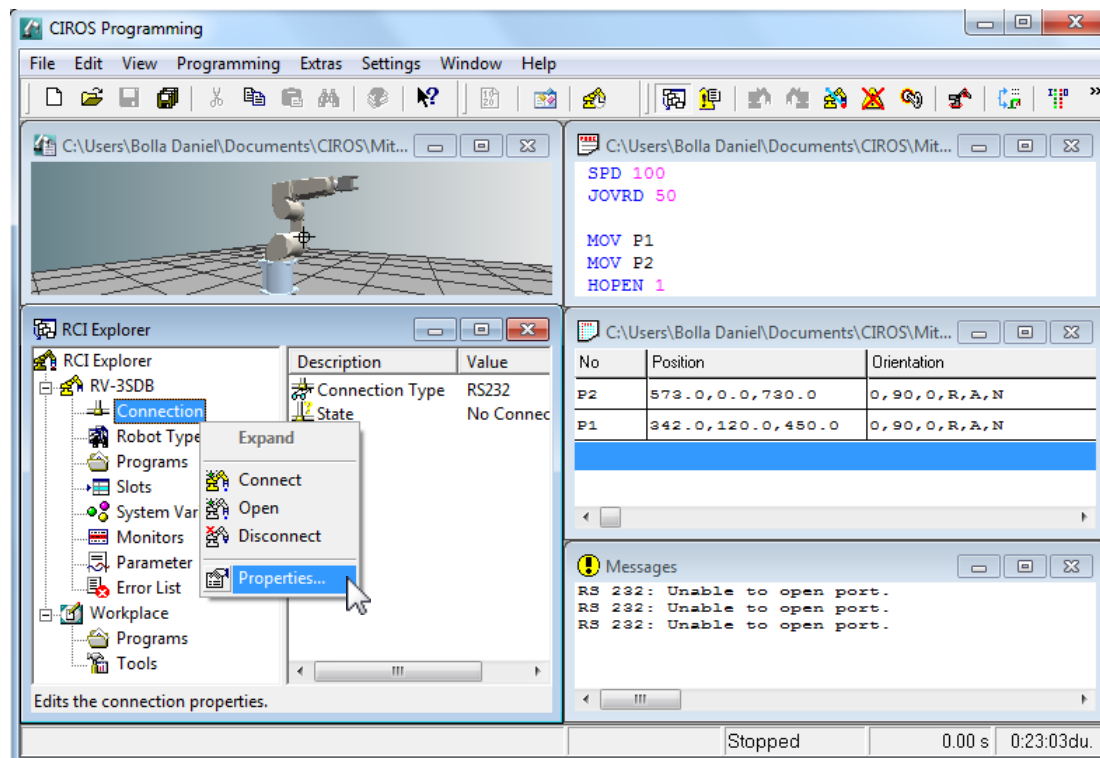


3. Connection to the Robot



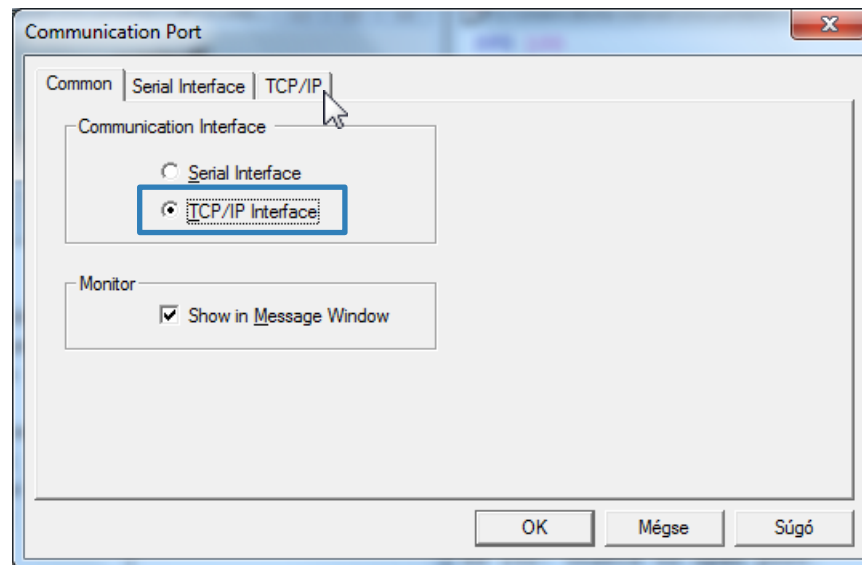
Open the Connection Parameters

- Right-click on the Connection in the RCI Explorer (in the robot section)
- And select Properties...



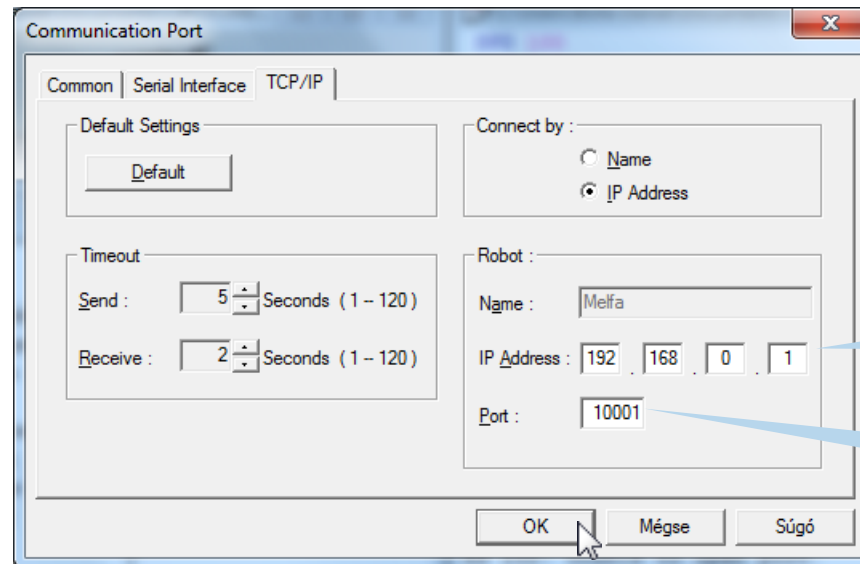
Select the TCP/IP Interface at Connection Interface

- And change to the TCP/IP tab



Set the IP address and the Port

- You can find it in the documentation
- And then press the OK button

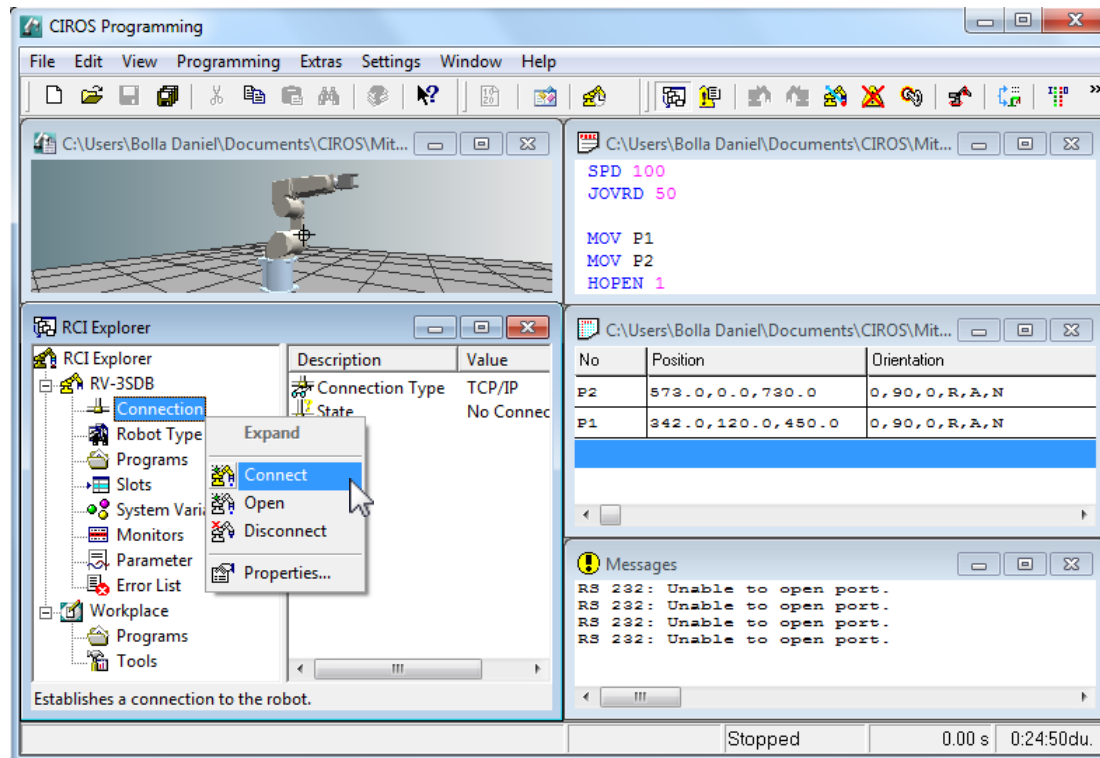


The IP address of the robot

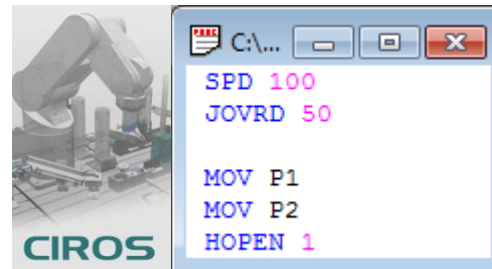
The port number of the robot

Connect to the Robot

- Right-click on the Connection in the RCI Explorer (in the robot section)
- And select Connect



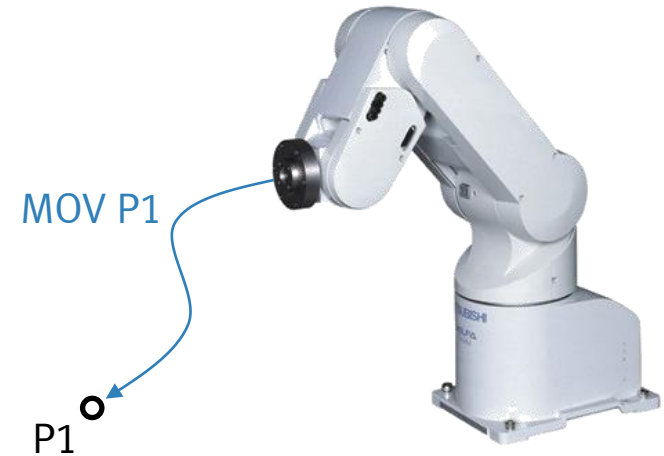
4. Melfa-Basic V Basics



Move Commands

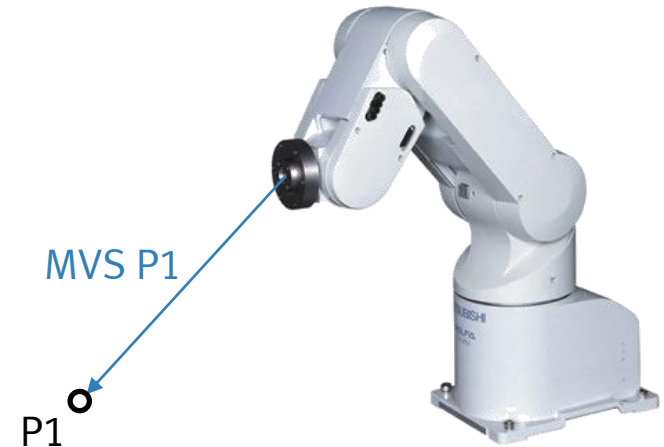
- **MOV <Position>**

- Movement with joint interpolation.
- You can reach all positions with this movement.
- But you do not know the path.
- We use it for movements far from the objects
- E.g.: **MOV P1** ' Move to P1 with joint interpolation





- **MVS <Position>**


- Movement with linear interpolation.
- Use to avoid collisions. (You know the path.)
- But you can not reach all positions with this movement.
- We use it for movements near to the objects
- E.g.: **MVS P1** ' Move to P1 with linear interpolation



Gripper Commands

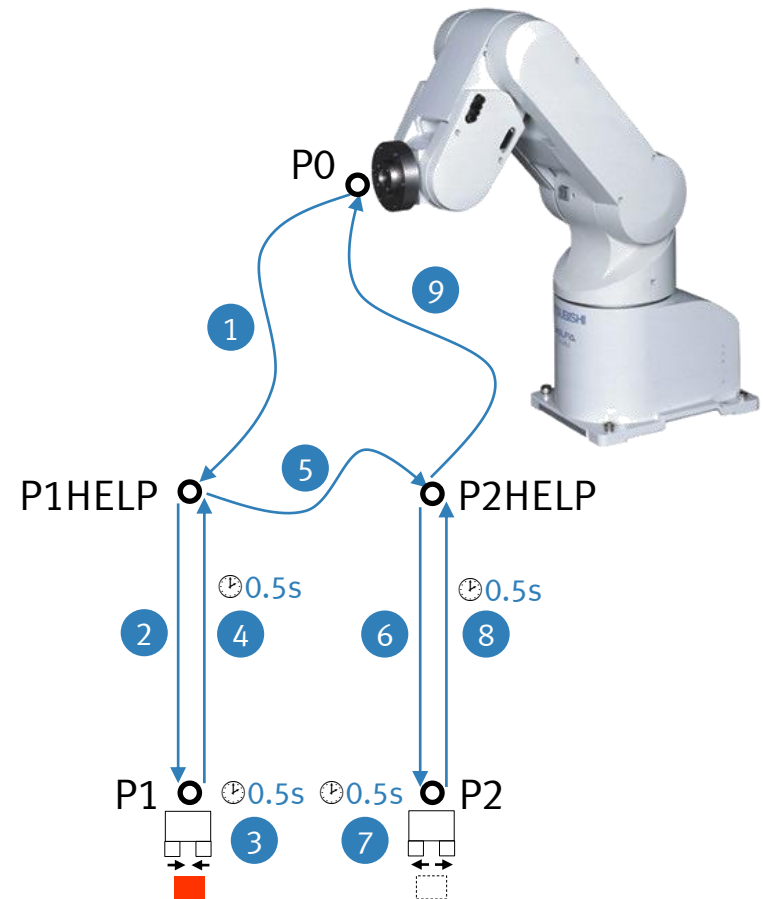
- **HOPEN <Index> / HCLOSE <index>**
 - Open / Close the gripper.
 - Robot can have several grippers, so you could specify the index of the gripper. (The index starts from 1)
 - E.g.: **HOPEN 1** ' Open the gripper 1 
 - HCLOSE 1** ' Close the gripper 1 

Time Commands

- **DLY <Seconds>**
 - Wait some minutes (Delay).
 - The time is in seconds with a point separator (e.g.: 0.5).
 - It is good to avoid an early gripping of work pieces.
 - E.g.: **DLY 0.5** ' Wait 0.5 seconds  0.5s

Exercise 1 – Basic Movements

1. Move in any way to P1HELP position.
2. Move to P1 position with linear movement.
3. After 0.5s delay, grip the work piece.
4. After 0.5s delay, move back to P1HELP with linear interpolation.
5. Move in any way to P2HELP position.
6. Move to P2 with linear interpolation.
7. After 0.5s delay, drop the work piece.
8. After 0.5s, move back to P2HELP with linear interpolation.
9. Move in any way to P0.

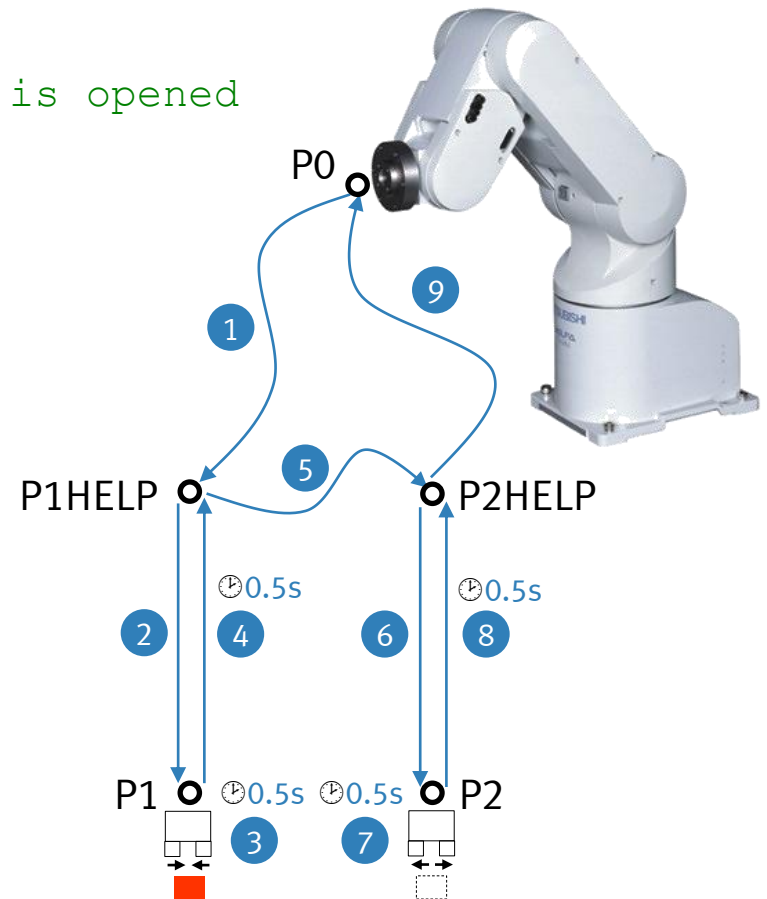


Exercise 1 – Basic Movements

- The solution:

```

HOPEN 1      'To be sure, the gripper is opened
MOV P1HELP  ' (1)
MVS P1       ' (2)
DLY 0.5     ' (3)
HCLOSE 1    ' (4)
DLY 0.5     ' (4)
MVS P1HELP
MOV P2HELP  ' (5)
MVS P2      ' (6)
DLY 0.5     ' (7)
HOPEN 1
DLY 0.5     ' (8)
MVS P2HELP
MOV P0      ' (9)
    
```

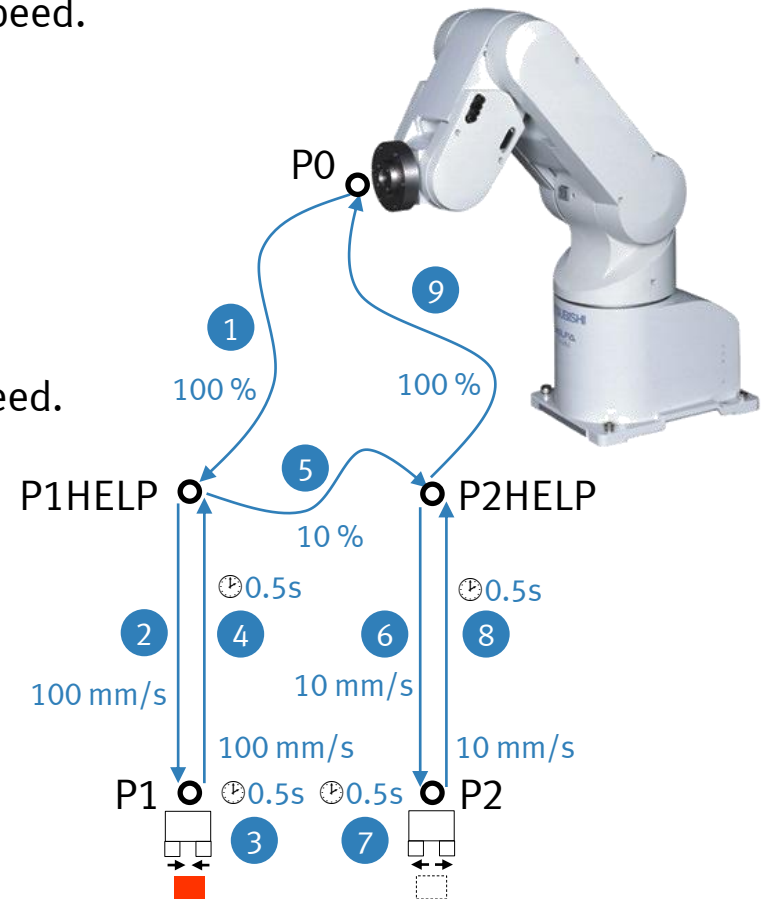


Speed/Acceleration Commands

- **OVRD <%>**
 - Override of all speeds (independently from movement type) (0-100%).
 - E.g.: **OVRD 50** ' Decrease the speed to 50%
- **SPD <mm/s>**
 - Change the speed of the linear movement (MVS).
 - It is in mm/s
 - E.g.: **SPD 100** ' Change the linear speed to 100 mm/s
- **JOVRD <%>**
 - Change the speed of the joint interpolated movement (MOV) (0-100%).
 - E.g.: **JOVRD 50** ' Change the speed of joint interpolated movements to 50%
- **ACCEL <accel %>, <decel %>**
 - Change the acceleration and deceleration of the movements (0-100%).
 - E.g.: **ACCEL 80, 50** ' Change the acceleration to 80%, and the deceleration to 50%.

Exercise 2 – Speeds

1. Move to P1HELP position with 100% of maximum speed.
2. Move to P1 position with 100 mm/s.
3. After 0.5s delay, grip the work piece.
4. After 0.5s delay, move back to P1HELP with 100 mm/s.
5. Move to P2HELP position with 10% of maximum speed.
6. Move to P2 with 10 mm/s.
7. After 0.5s delay, drop the work piece.
8. After 0.5s, move back to P2HELP with 10 mm/s.
9. Move to P0 with maximum speed.



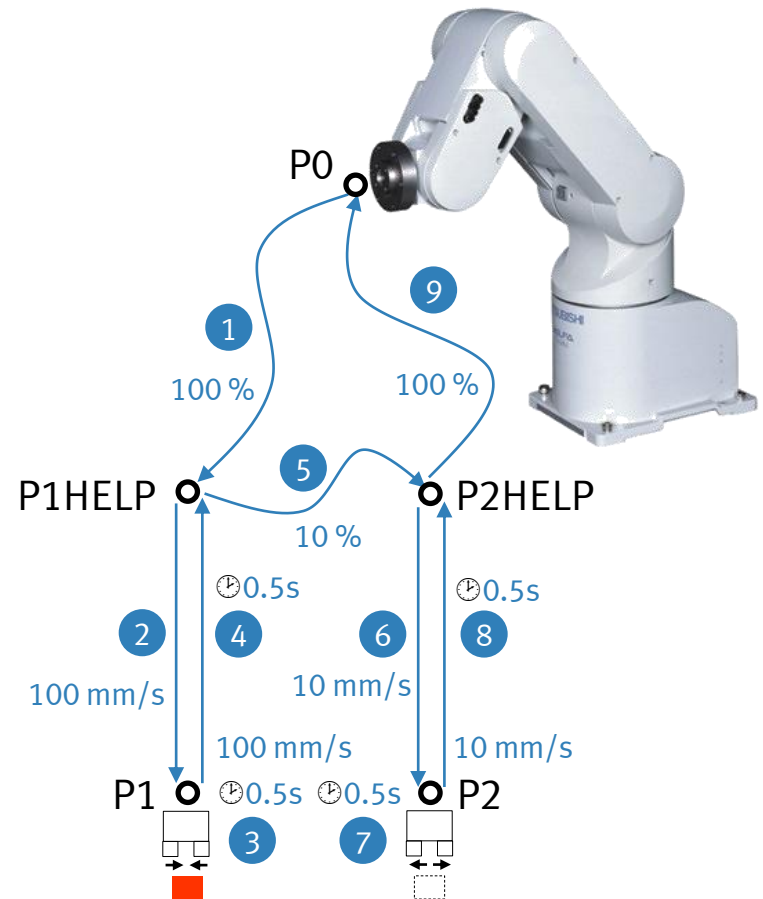
Exercise 2 – Speeds

- The solution:

```

HOPEN 1
+ JOVRD 100 ' (1)
MOV P1HELP
+ SPD 100 ' (2)
MVS P1
DLY 0.5 ' (3)
HCLOSE 1
DLY 0.5 ' (4)
MVS P1HELP
+ JOVRD 10 ' (5)
MOV P2HELP
+ SPD 10 ' (6)
MVS P2
DLY 0.5 ' (7)
HOPEN 1
DLY 0.5 ' (8)

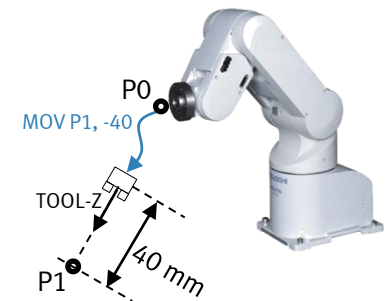
MVS P2HELP
+ JOVRD 100 ' (9)
MOV P0
    
```



Relative Movements – Less Position to Teach

- **MOV <Position>, <Shift> / MVS <Position>, <Shift>**

- Move to the <Position> shifted with <Shift> mm to TOOL-Z direction.
- E.g.: **MOV P1, -40** ' Move to P1 shifted with 40 mm to TOOL-Z minus
- MVS P1, -40** ' Move to P1 shifted with 40 mm to TOOL-Z minus



- **DEF POS <Position name>**

- Define a new position to use as relative position.

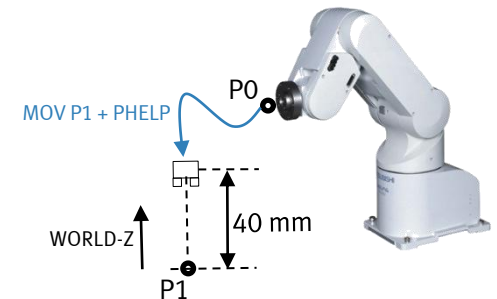
- **<Position> = (<X>, <Y>, <Z>, <A>, , <C>)**

- Assigne the position with WORLD coordinates.

- **MOV <Position1> + <Position2> / MVS <Position1> + <Position2>**

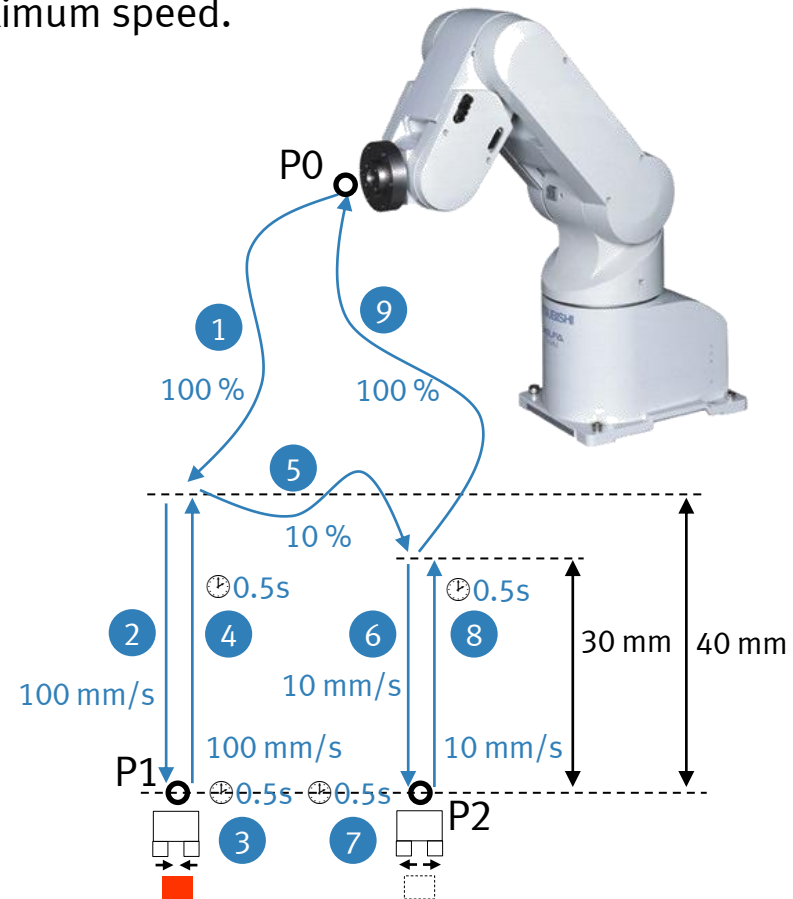
- Move to the <Position1> shifted with <Position2> in WORLD coordinates.

- E.g.: **DEF POS PHELP** ' Define the PHELP position
- PHELP = (0, 0, 40, 0, 0,0)** ' Assign the PHELP with a 40 mm WORLD-Z positive shift
- MOV P1 + PHELP** ' Move to P1 shifted with PHELP in WORLD cooednates
- MVS P1 + PHELP** ' Move to P1 shifted with PHELP in WORLD cooednates



Exercise 3 – Relative Movements

1. Move to 40 mm above P1 position with 100% of maximum speed.
2. Move to P1 position with 100 mm/s.
3. After 0.5s delay, grip the work piece.
4. After 0.5s delay, move back to 40 mm above P1 with 100 mm/s.
5. Move to 30 mm above P2 position with 10% of maximum speed.
6. Move to P2 with 10 mm/s.
7. After 0.5s delay, drop the work piece.
8. After 0.5s, move back to 30 mm above P2 with 100 mm/s.
9. Move to P0 with maximum speed.

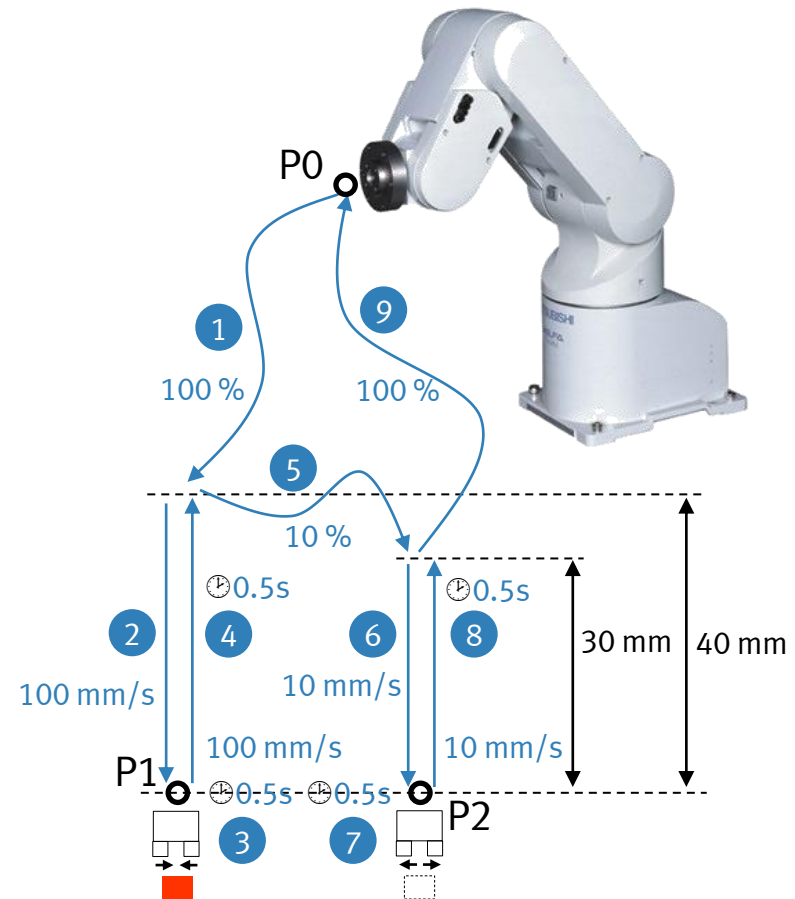


Exercise 3 – Relative Movements

- 1st solution:

```

HOPEN 1
JOVRD 100 ' (1)
→ MOV P1, -40
SPD 100 ' (2)
MVS P1
DLY 0.5 ' (3)
HCLOSE 1
DLY 0.5 ' (4)
→ MVS P1, -40
JOVRD 10 ' (5)
→ MOV P2, -30
SPD 10 ' (6)
MVS P2
DLY 0.5 ' (7)
HOPEN 1
DLY 0.5 ' (8)
→ MVS P2, -30
JOVRD 100 ' (9)
MOV P0
    
```

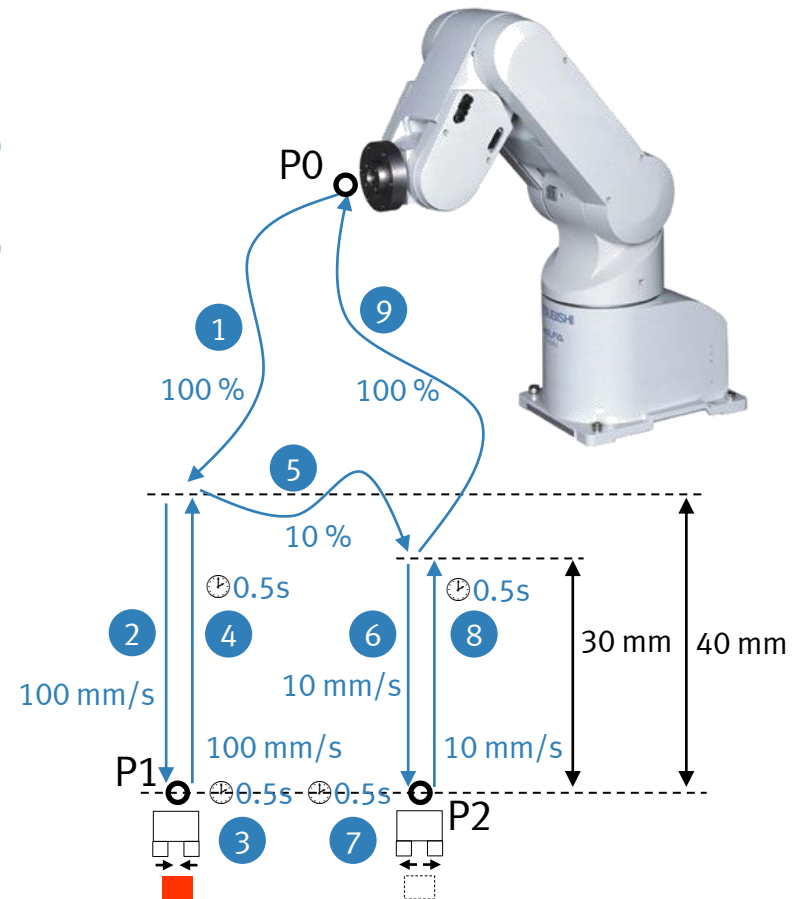


Exercise 3 – Relative Movements

- 2nd solution:

```

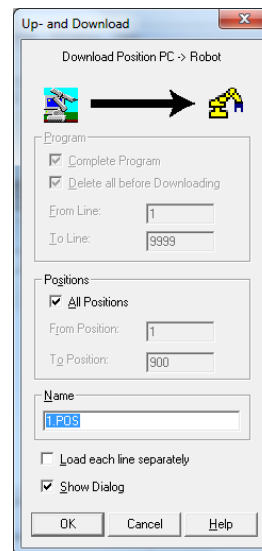
+ DEF POS PZ40          MVS P2
+ PZ40=( 0 , 0 , 40 , 0 , 0 , 0 )  DLY 0.5      ' (7)
+ DEF POS PZ30          HOPEN 1
+ PZ30=( 0 , 0 , 30 , 0 , 0 , 0 )  DLY 0.5      ' (8)
                                → MVS P2 + PZ30
                                JOVRD 100      ' (9)
                                MOV P0
→ MOV P1 + PZ40
  SPD 100                ' (2)
  MVS P1
  DLY 0.5                ' (3)
  HCLOSE 1
  DLY 0.5                ' (4)
→ MVS P1 + PZ40
  JOVRD 10               ' (5)
→ MOV P2 + PZ30
  SPD 10                 ' (6)
  
```



I/O handling

- **DEF IO <name> = Bit,<bbitnumber>**
 - Define an input or an output (depends on usage)
 - E.g.: **DEF IO StartButton, 1** ' Define Start Button to input 1
DEF IO StartLamp, 1 ' Define Start Lamp to output 1
- **WAIT <input/variable> = <value>**
 - Wait until the condition is TRUE
 - E.g.: **WAIT StartLamp = 1** ' Wait for Start Button
- **<output/variable> = <value>**
 - Assign an output or a variable
 - E.g.: **StartLamp = 1** ' Switch on the Start Lamp
- **IF <condition> THEN <instructions> ELSE <instructions> ENDIF**
 - Conditional branching

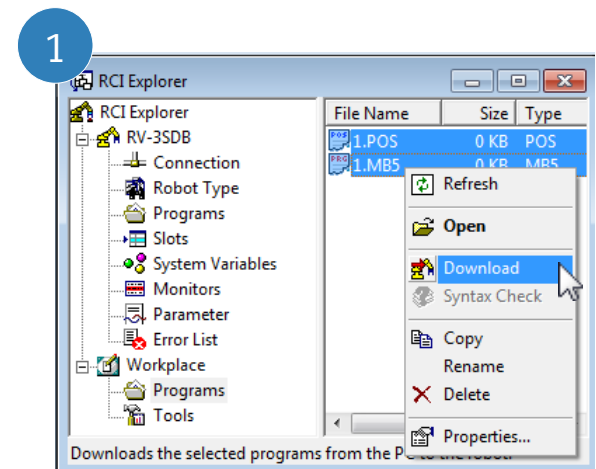
5. Download and Upload



➔ Skip over

Download

- Download from the PC to the robot.
- You can download the programs and the position lists separately.
- Two different ways:
 1. Select the programs and/or position lists in RCI Explorer (in workplace section) Right-click and select Download.
 2. Use the download tool.
Then you will download the content of the active window.
E.g.: If the program editor window is active, than the program will be downloaded.



Upload

- Upload from the robot to the PC.
- If you upload a program, the position list of this program will automatically be uploaded.
(You cannot upload the program and the position list separately.)
- Two different ways:
 1. Select the programs in RCI Explorer (in robot section)
Right-click and select Upload.
 2. Use the upload tool.
Then you will upload ???



6. Further presentations in this topic

→ Robot Basics [EN]

Robot Usage [EN] :

→ Mitsubishi Robot Usage [EN]

Robot Programing [EN] :

→ CIROS Programming Basics Melfa-Basic V [EN]

Robot Maintanace [EN] :

→ Mitsubishi Robot Maintanace [EN]