

# BỘ GIÁO DỤC VÀ ĐÀO TẠO



## THỰC HÀNH ROBOT TRONG CÔNG NGHIỆP

ROBOT MELFA - MITSUBISHI

**Biên soạn:** ThS. Phạm Quốc Phương

---

**THỰC HÀNH ROBOT**

Ấn bản 2022

**PHƯƠNG PHÁP ĐÁNH GIÁ MÔN HỌC**

---

Môn học được đánh giá gồm:

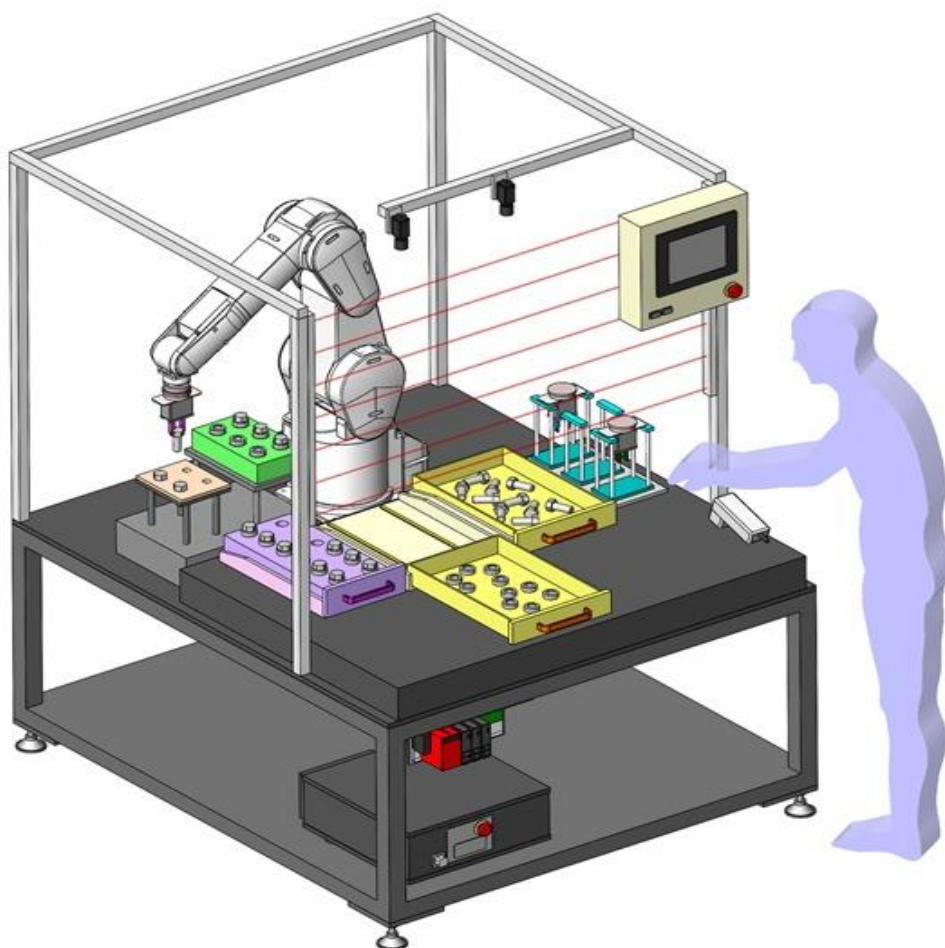
- Điểm chuyên cần: 30%.
- Điểm chuẩn bị bài ở nhà và thực hành tại lớp: 30%.
- Điểm kiểm tra kết thúc học phần và báo cáo thí nghiệm: 40%.

---

# GIỚI THIỆU

## BÀI 1

### GIỚI THIỆU HỆ THỐNG THỰC HÀNH ROBOT



*Hình 1:* Hệ thống thực hành Robot









## 1. Vertical, multiple-joint type (RV) series

### 1.1 Hình dáng



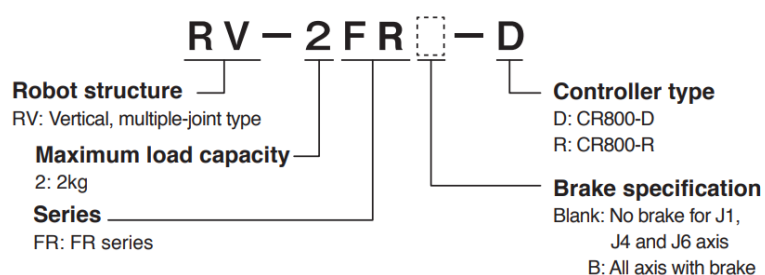
Hình 2: RV-FR SERIES

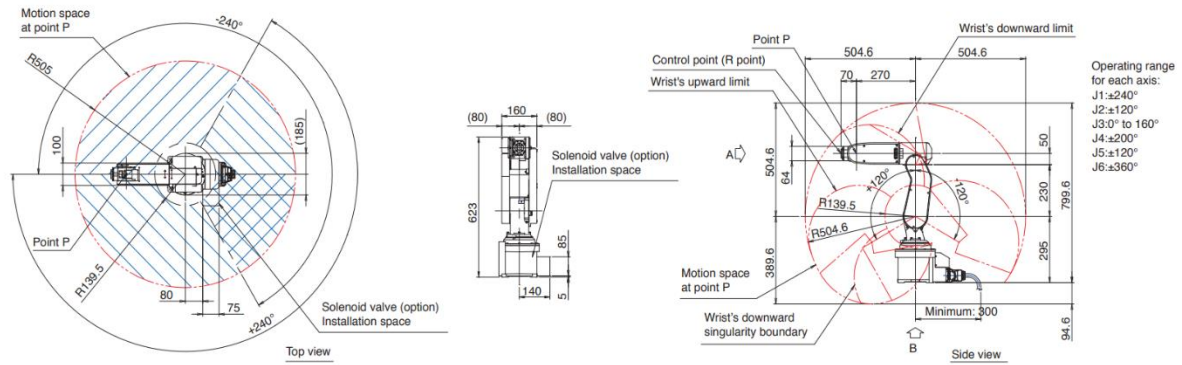
### 1.2 Phân loại

										
Type	RV-2FR	RV-2FRL	RV-4FR	RV-4FRL	RV-7FR	RV-7FRL	RV-7FRLL	RV-13FR	RV-13FRL	RV-20FR
Maximum load capacity	3kg		4kg		7kg			13kg		20kg
Maximum reach radius	504mm		515mm							
		649mm		649mm	713mm	908mm	1503mm	1094mm	1388mm	1094mm

Hình 3: Các loại Robot RV-FR Series

### 1.3 Thông số kỹ thuật





Hình 4: Vùng hoạt động – kích thước Robot RV-2FR

## 2. Horizontal, multiple-joint type (RH) series

### 2.1 Hình dáng



Hình 5: RH-3FRH Series

### 1.2 Phân loại

Type	RH-3FRH	RH-6FRH	RH-12FRH	RH-20FRH	RH-3FRHR
Maximum load capacity	3kg	6kg	12kg	20kg	3kg
Maximum reach radius	350mm 450mm 550mm	350mm 450mm 550mm	700mm 850mm 1000mm		350mm
Z stroke	150mm <sup>*1</sup>	200mm 340mm	350mm 450mm		150mm <sup>*2</sup>

Hình 6: Các loại Robot RH

### 1.3 Thông số kỹ thuật



Q Parameter Setting

PLC Name | PLC System | PLC File | PLC RAS | Boot File | Program | SFC | Device | I/O Assignment | Multiple CPU Setting | Built-in Ethernet Port Setting

I/O Assignment(\*1)

No.	Slot	Type	Model Name	Points	Start XY
0	PLC	PLC No.1			3E00
1	PLC	PLC No.2			3E10
2	1(0-1)	Input		64Points	
3	2(0-2)	Output		64Points	
4	3(0-3)				
5	4(0-4)				
6	5(0-5)				
7	6(0-6)				

Assigning the I/O address is not necessary as the CPU does it automatically.  
Leaving this setting blank will not cause an error to occur.

Base Setting(\*1)

	Base Model Name	Power Model Name	Extension Cable	Slots
Main				8
Ext.Base1				
Ext.Base2				
Ext.Base3				
Ext.Base4				
Ext.Base5				
Ext.Base6				
Ext.Base7				

Base Mode  
☐ Auto  
☒ Detail  
 8 Slot Default  
 12 Slot Default  
 Select module name

Export to CSV File | Import Multiple CPU Parameter | Read PLC Data

(\*1)Setting should be set as same when using multiple CPU.

Q Parameter Setting

PLC Name | PLC System | PLC File | PLC RAS | Boot File | Program | SFC | Device | I/O Assignment | Multiple CPU Setting | Built-in Ethernet Port Setting

No. of PLC (\*1)  
 2 Count

Host Station  
 No Specification

Operation Mode (\*1)  
 Error Operation Mode at the Stop of PLC  
☒ All station stop by stop error of PLC1  
☒ All station stop by stop error of PLC2  
☒ All station stop by stop error of PLC3  
☒ All station stop by stop error of PLC4

Multiple CPU Synchronous Startup Setting(\*1)  
 Target PLC  
☒ No.1  
☒ No.2  
☒ No.3  
☒ No.4

Online Module Change(\*1)  
☐ Enable Online Module Change with Another PLC.  
 When the online module change is enabled with another PLC,  
 I/O status outside the group cannot be taken.

I/O Sharing When Using Multiple CPUs (\*1)  
☒ All CPUs Can Read All Inputs  
☒ All CPUs Can Read All Outputs

Multiple CPU High-Speed Transmission Area Setting | Communication Area Setting (Refresh Setting)

☒ Use Multiple CPU High-Speed Transmission

PLC	CPU Specific Send Range (*1)					
	Points(K)	I/O No.	User Setting Area			Auto Refresh
			Points	Start	End	Points
PLC No.1	1	U3E0	1024	G10000	G11023	0
PLC No.2	1	U3E1	1024	G10000	G11023	0
PLC No.3						
PLC No.4						

Set auto refresh setting if it is needed( No Setting / Already Set )  
 Total 2K Points ☐ Advanced Setting(\*1) Assignment Confirmation

The total number of points is up to 14K.



- Cấu hình PLC Cooperation Parametter(RT-Toolbox3)

No. of CPU(QMLTCPUN)

No. of CPU 2

Please set the number of Multiple CPU.

Multiple CPU Synchronous Startup

☒ CPU1

☒ CPU2

☐ CPU3

☐ CPU4

Robot Input Offset (QMLTCPUS)

☐ Manual Setting

0-14(K word)

-1

Multiple CPU High Speed Transmission Area Setting (QMLTCPU1 - QMLTCPU4)

CPU	Send Range for Each CPU					
	Point(K)	I/O No.	User Setting Area			Automatic Refresh
			Point	Start	End	Point
No.1	<span>7</span>	U3E0	7168	G10000	G17167	<span>0</span>
No.2	<span>7</span>	U3E1	7168	G10000	G17167	<span>0</span>
No.3	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
No.4	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Total 14K Point ☐ Advanced Setting

The total is up to 14K points.

---

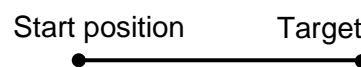
## BÀI 2

### ĐIỀU KHIỂN ROBOT MELFA RV-2FR-Q



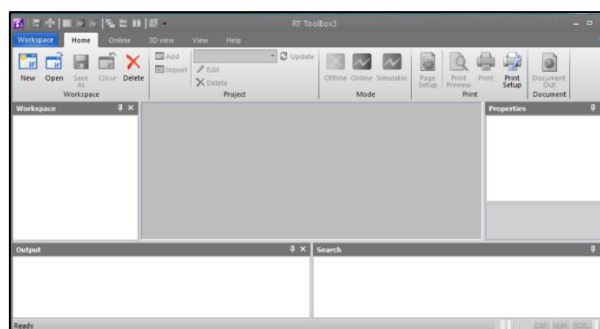
#### A. THỰC HÀNH

**Bài 2.1** Thực hiện thao tác di chuyển Robot từ điểm Start position đến Target position theo panel điều khiển



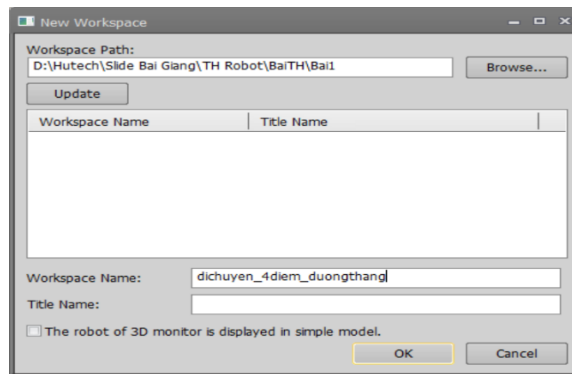
#### Hướng dẫn:

1. Chạy phần mềm RT-Toolbox 3: Chọn New

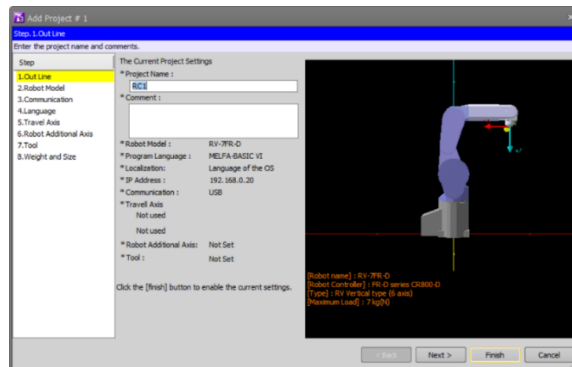


2. Đặt tên

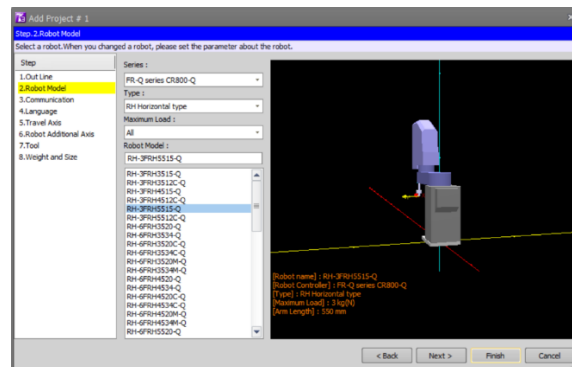
**Workspace Name:** Đặt tên (chú ý không có khoảng cách)



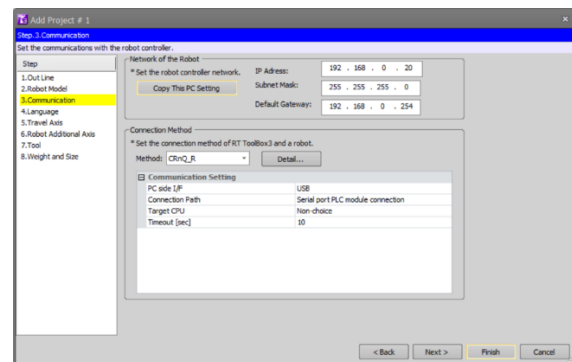
### 3. Đặt tên project (mặc định RC1)



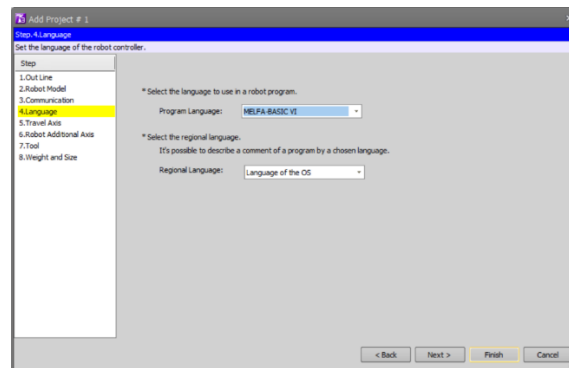
### 4. Chọn Controller và loại robot: Series:CR800-Q; Robot model:RV-2FR



### 5. Chọn kết nối RT Toolbox 3 – Robot (mặc định)

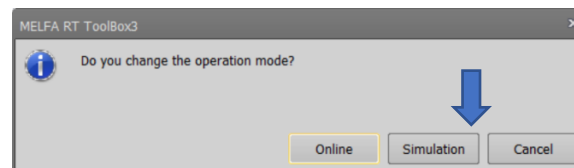


### 6. Chọn ngôn ngữ lập trình



7. Chọn Finish

8. Chọn **Simulation**

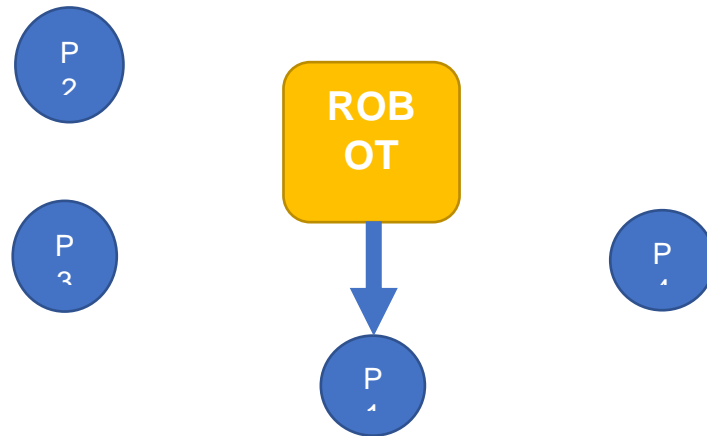


9. Mở cửa sổ **Operation Panel** → di chuyển J1 J2 J3 J4 J5 J6 để tạo các điểm: **Start position, Target position**



---

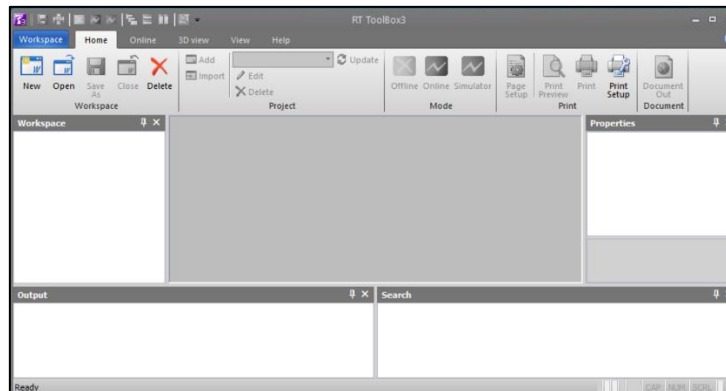
**Bài 2.2:** Điều khiển Robot RV-2FR di chuyển đến các điểm P1, P2, P3, P4.



---

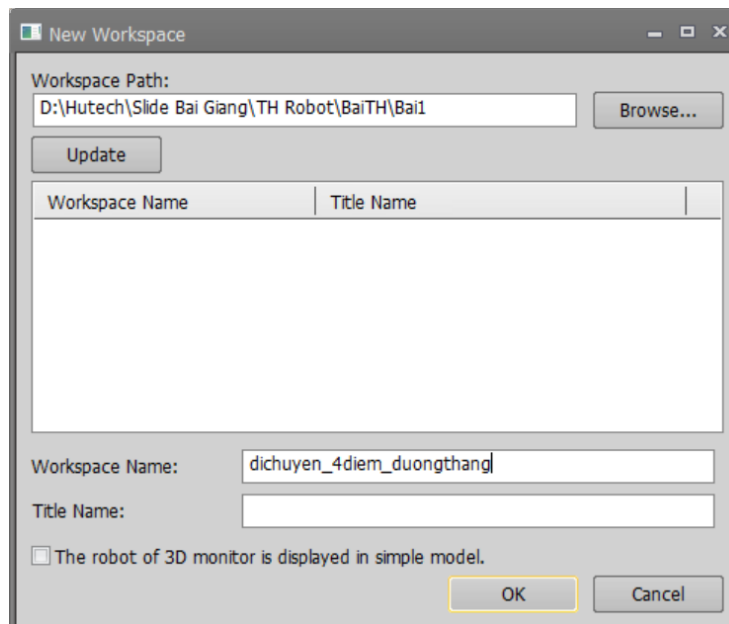
## Hướng dẫn:

### 1. Chạy phần mềm RT-Toolbox 3: Chọn **New**

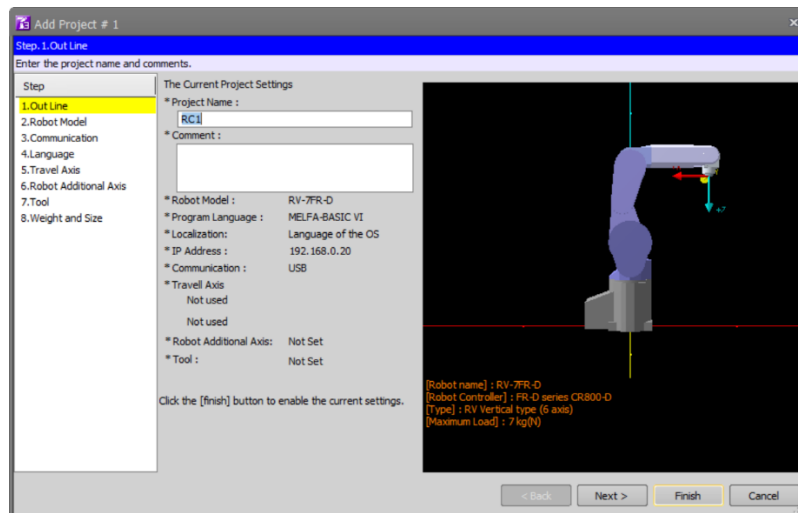


### 2. Đặt tên

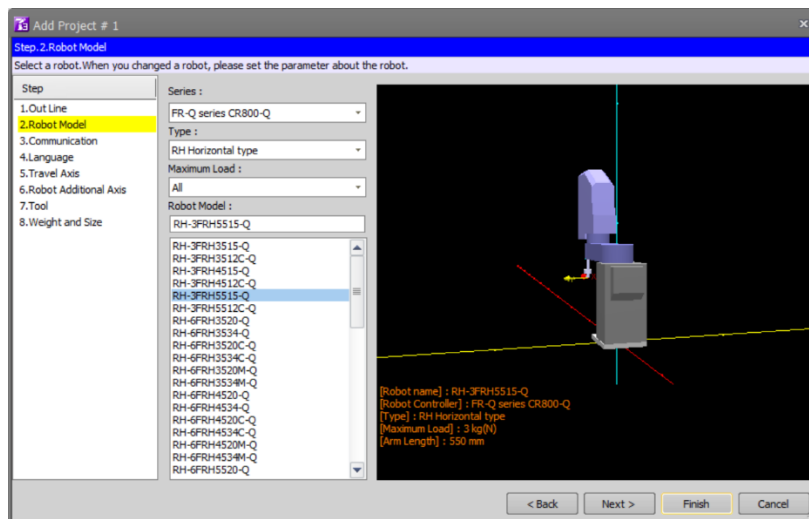
**Workspace Name:** Name



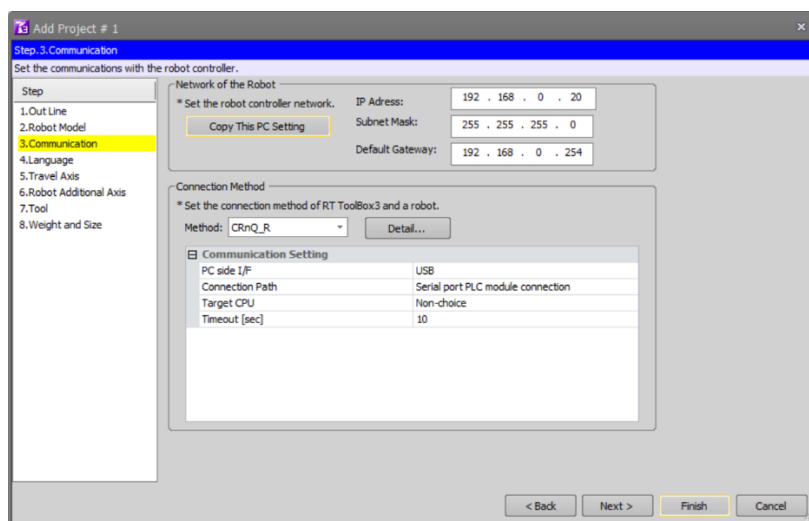
### 3. Đặt tên project (mặc định RC1)



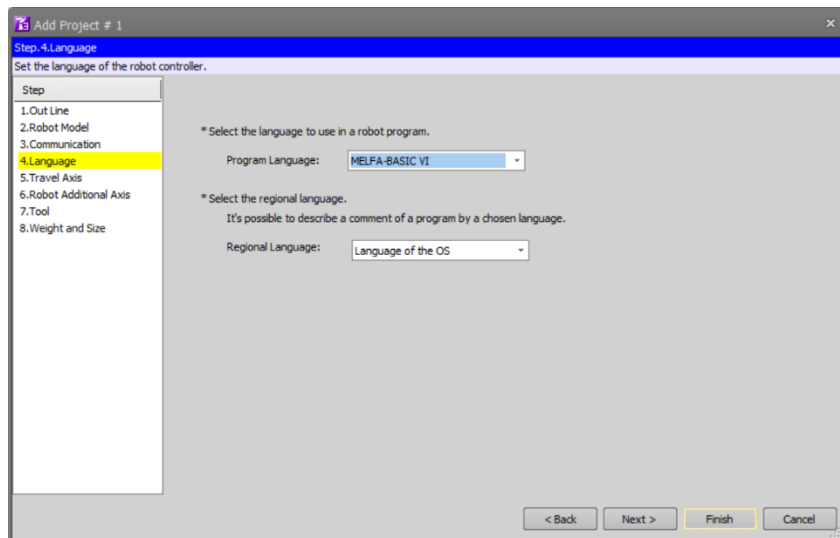
#### 4. Chọn Controller và loại robot: Theo hình



#### 5. Chọn kết nối RT Toolbox 3 – Robot (mặc định)→ Chọn **Next**

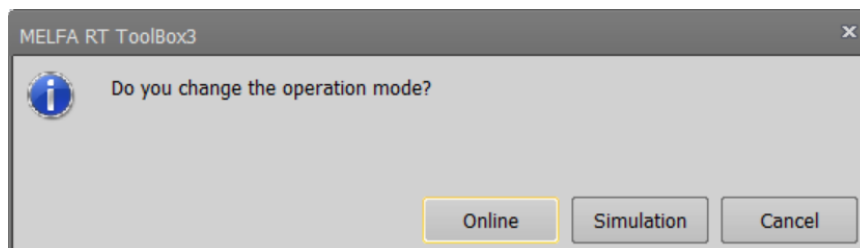


#### 6. Chọn ngôn ngữ lập trình



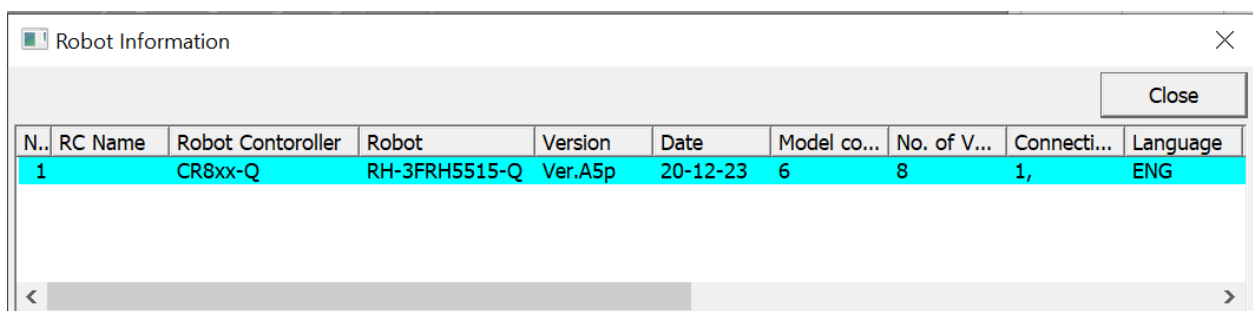
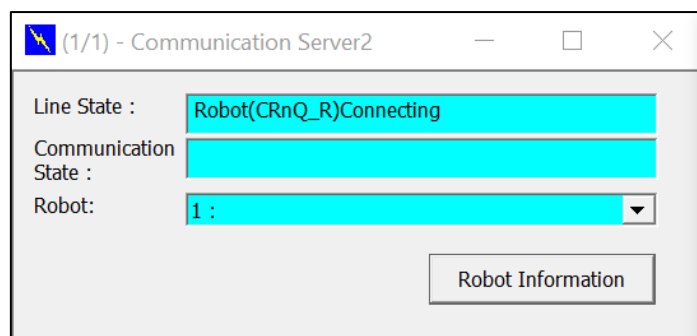
7. Chọn Finish

8. Chọn **Simulation**



9. Trạng thái kết nối RT Toolbox 3 ⇔ Robot

10. Kết nối thành công





---

**Chú ý:** Không để các vật cản trên đường di chuyển của tay Robot

11. Di chuyển J1, J2, J3, J4 đến vị trí điểm P1 theo bảng giá trị

J1:	-86.488	-	+
J2:	-1.595	-	+
J3:	369.495	-	+
J4:	359.381	-	+

- Nhấn + hoặc - để thay đổi

12. Di chuyển J1, J2, J3, J4 đến vị trí điểm P2 theo bảng giá trị

J1:	-51.199	-	+
J2:	-35.101	-	+
J3:	369.495	-	+
J4:	359.381	-	+

- Nhấn + hoặc - để thay đổi

13. Di chuyển J1, J2, J3, J4 đến vị trí điểm P3 theo bảng giá trị

J1:	46.093	-	+
J2:	28.497	-	+
J3:	369.495	-	+
J4:	359.381	-	+

- Nhấn + hoặc - để thay đổi

14. Di chuyển J1, J2, J3, J4 đến vị trí điểm P4 theo bảng giá trị

J1:	83.940	-	+
J2:	57.116	-	+
J3:	369.495	-	+
J4:	359.381	-	+

- Nhấn + hoặc - để thay đổi

---

## Bài 2.3: Viết chương trình đóng mở tay kẹp lặp lại liên tục

### Lệnh thực hiện:

M\_UDevW: Reads/ Writes per word. (16 bits)

M\_UDevD: Reads/ Writes per double word. (32 bits)

### Ví dụ:

' The &HFFFF (hexadecimal number) is written to the shared memory address 10010 of No. 2 CPU (host CPU).

' Calculate the X coordinate value of position variable P1 by 1000. And write the result value to shared memory addresses of 10011/10012 (two word) on No. 2 CPU (host CPU).

' The value of 3-bit width from 10001 of shared memory address of No. 3 CPU is substituted to M1.

1 M\_UDevW(&H3E1, 10010)=&HFFFF

2 M\_UDevD(&H3E1, 10011)=P1.X \* 1000

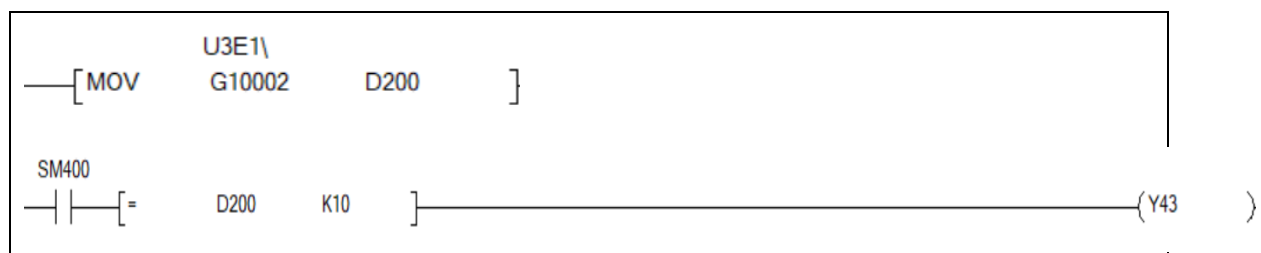
3 M1%=M\_UDevW(&H3E2, 10001) And &H7

### Chương trình tham khảo

- RT Toolbox3

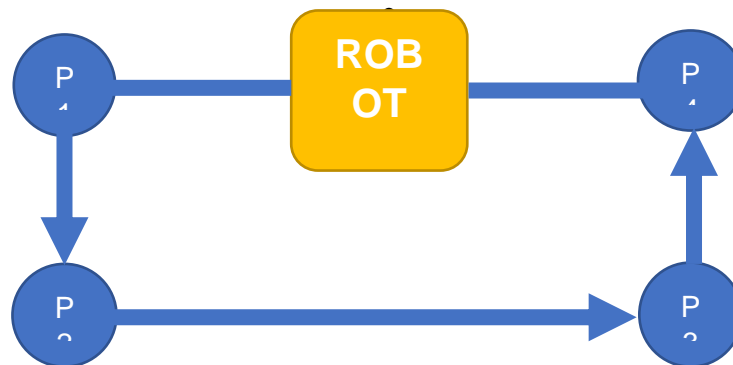
```
*L1
` lệnh mở tay kẹp
M_UDevW(&H3E1,10002) = &HA
Dly 5
` lệnh đóng tay kẹp
M_UDevW(&H3E1,10002) = &HB
Dly 10
GoTo *L1
```

- Chương trình PLC



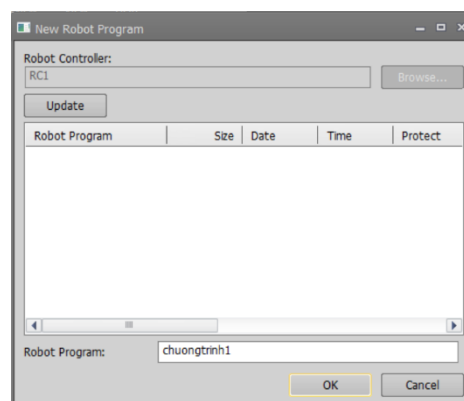
---

**Bài 2.4: Viết chương trình** điều khiển Robot RH-3FRH di chuyển theo ***đường thẳng*** từ điểm P1 → P2 → P3 → P4



**Hướng dẫn:**

**Bước 1:** Tạo New Robot Program → đặt tên "chuongtrinh1" → OK



## Bước 2: Thêm 4 điểm P1, P2, P3, P4 vào cửa sổ Position

Name	J1	J2	J3	J4	J5	J6	J7	J8
P1	-86.488	-1.595	369.495	359.381	X	X	X	X
P2	-51.199	-35.101	369.495	359.381	X	X	X	X
P3	46.093	28.497	369.495	359.381	X	X	X	X
P4	83.940	57.116	369.495	359.381	X	X	X	X

XYZ Alt+X   Joint Alt+J   Work Coordinate Alt+W

## Bước 3: Cập nhật vị trí J1, J2, J3, J4 theo 4 điểm P1, P2, P3, P4

Position Jump

Robot Coordinates in Selection   Offset:

J1: -51.199   0.000

J2: -35.101   0.000

J3: 369.495   0.000

J4: 359.381   0.000

Offset Reset

Interpolation: Joint (Mov)

Move   Cancel

- Điểm P1

J1: -86.488   -   +

J2: -1.595   -   +

J3: 369.495   -   +

J4: 359.381   -   +

- Điểm P2

J1: -51.199   -   +

J2: -35.101   -   +

J3: 369.495   -   +

J4: 359.381   -   +

- Điểm P3

J1: 46.093   -   +

J2: 28.497   -   +

J3: 369.495   -   +

J4: 359.381   -   +

- Điểm P4

J1:	83.940	-	+
J2:	57.116	-	+
J3:	369.495	-	+
J4:	359.381	-	+

Kết quả sau khi cập nhật vị trí các điểm P1, P2, P3, P4

Name	J1	J2	J3	J4	J5	J6	J7	J8
P1	-86.488	-1.595	369.495	359.381	X	X	X	X
P2	-51.199	-35.101	369.495	359.381	X	X	X	X
P3	46.093	28.497	369.495	359.381	X	X	X	X
P4	83.940	57.116	369.495	359.381	X	X	X	X

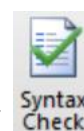
**Bước 4:** Viết chương trình cho Robot di chuyển theo đường thẳng từ điểm P1 → P2 → P3 → P4 và dừng lại.

```

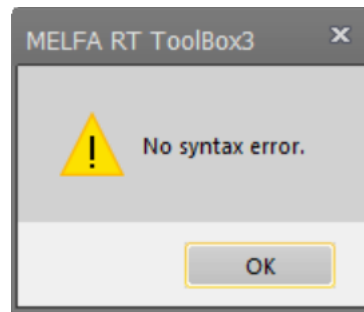
Program 1:RC1 CHUONGTRINH1... x
1 Servo On
2 Ovrd 50
3 Mvs P1
4 Dly 0.5
5 Mvs P2
6 Dly 0.5
7 Mvs P3
8 Dly 0.5
9 Mvs P4
10 Dly 0.5
11 Servo Off
12 Hlt
13

```

**Bước 5:** Kiểm tra lỗi

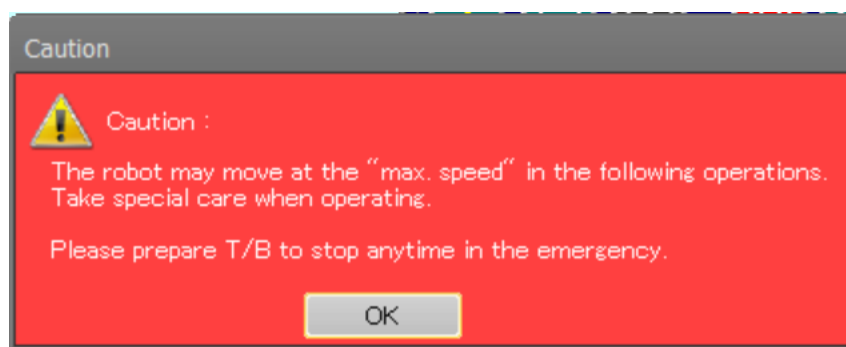


Sau khi viết code nhấn biểu tượng "Syntax Check" để kiểm tra lỗi. Nếu chương trình không có lỗi sẽ hiển thị cửa sổ như hình.



**Bước 6:** Chạy chương trình

Cửa sổ cảnh báo xuất hiện. Nhấn OK



**Chú ý:**

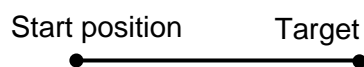
1. Kiểm tra khu vực chuyển động của Robot
2. Khi Robot hoạt động không đưa các vật cản vào khu vực chuyển động của Robot
3. Không để bất kỳ phần nào của cơ thể vào khu vực chuyển động của Robot để tránh các tai nạn.

---

# BÁO CÁO THỰC HÀNH

Ngày ..... tháng ..... năm .....

**Bài tập 1** Viết chương trình điều khiển Robot di chuyển theo yêu cầu sau:



## I. DÀNH CHO SINH VIÊN

### 1. Chương trình

.....

.....

.....

.....

.....

.....

.....

.....

### 2. Giải thích chương trình

.....

.....

.....

.....

.....

## II. DÀNH CHO CBHD THỰC HÀNH.

### 1. Nhận xét:



---

.....

.....

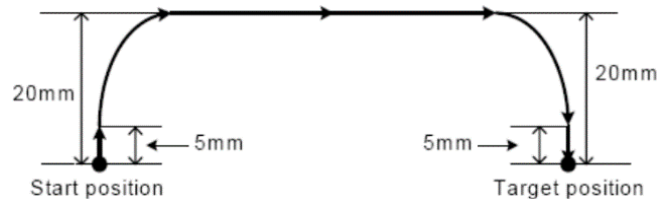
.....

**2. Điểm đánh giá:** .....

**3. Chữ ký của giảng viên:** .....

---

**Bài tập 2** Viết chương trình điều khiển Robot di chuyển theo yêu cầu sau:



## I. DÀNH CHO SINH VIÊN

### 1. Chương trình

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

### 2. Giải thích chương trình

.....

.....

.....

.....

.....

---

.....

.....

.....

.....

.....

**II. DÀNH CHO CBHD THỰC HÀNH.**

**1. Nhận xét:**

.....

.....

.....

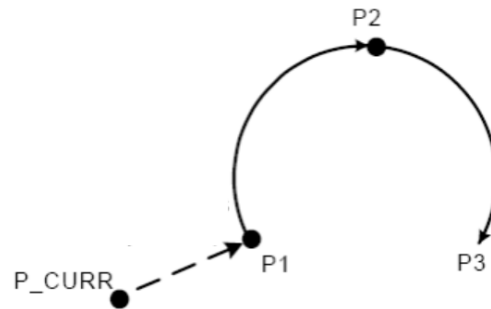
.....

**2. Điểm đánh giá: .....**

**3. Chữ ký của giảng viên: .....**

---

**Bài tập 3** Viết chương trình điều khiển Robot di chuyển theo yêu cầu sau:



## I. DÀNH CHO SINH VIÊN

### 1. Chương trình

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

### 2. Giải thích chương trình

.....

.....

.....

.....

.....

.....

---

.....

.....

.....

.....

**II. DÀNH CHO CBHD THỰC HÀNH.**

**1. Nhận xét:**

.....

.....

.....

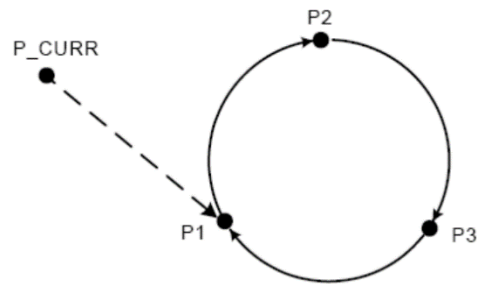
.....

**2. Điểm đánh giá:** .....

**3. Chữ ký của giảng viên:** .....

---

**Bài tập 4** Viết chương trình điều khiển Robot di chuyển theo yêu cầu sau:



## I. DÀNH CHO SINH VIÊN

### 1. Chương trình

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

### 2. Giải thích chương trình

.....

.....

.....

.....

.....

.....

---

.....

.....

.....

.....

**II. DÀNH CHO CBHD THỰC HÀNH.**

**1. Nhận xét:**

.....

.....

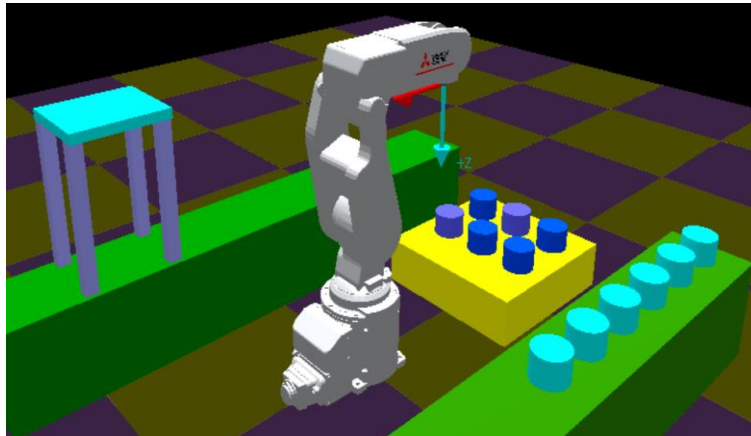
.....

.....

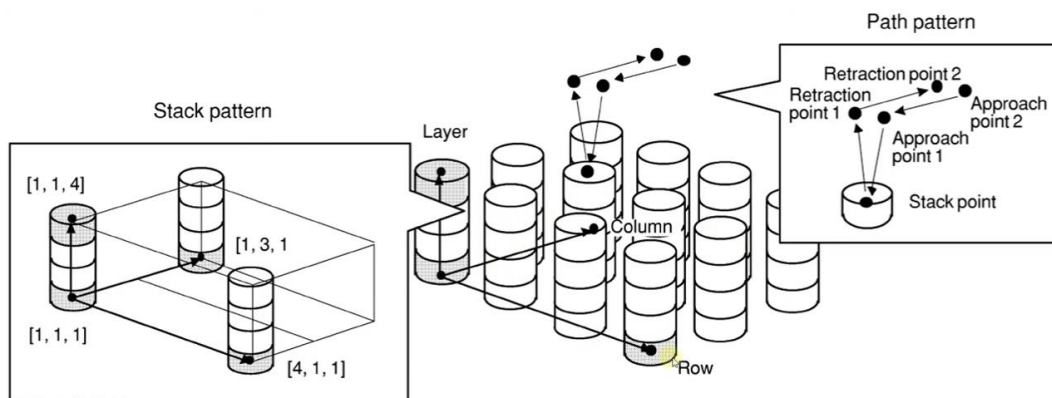
**2. Điểm đánh giá: .....**

**3. Chữ ký của giảng viên: .....**

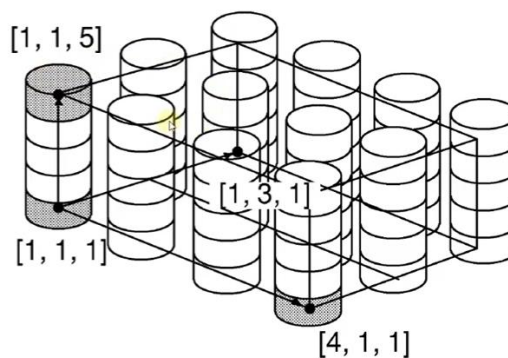
## Bài 5 Viết chương trình điều khiển Robot sắp 6 phôi lên băng tải



### Hướng dẫn:



Initial data	Points to be taught
ROWS = [4 FIX]	P [1, 1, 1]
COLUMNS = [3 FIX]	P [4, 1, 1]
LAYERS = [5 FIX]	P [1, 3, 1]
AUXILIARY POS = [YES]	P [1, 1, 5]



### 1. [Function]

Defines the pallet. (3-point pallet, 4-point pallet)



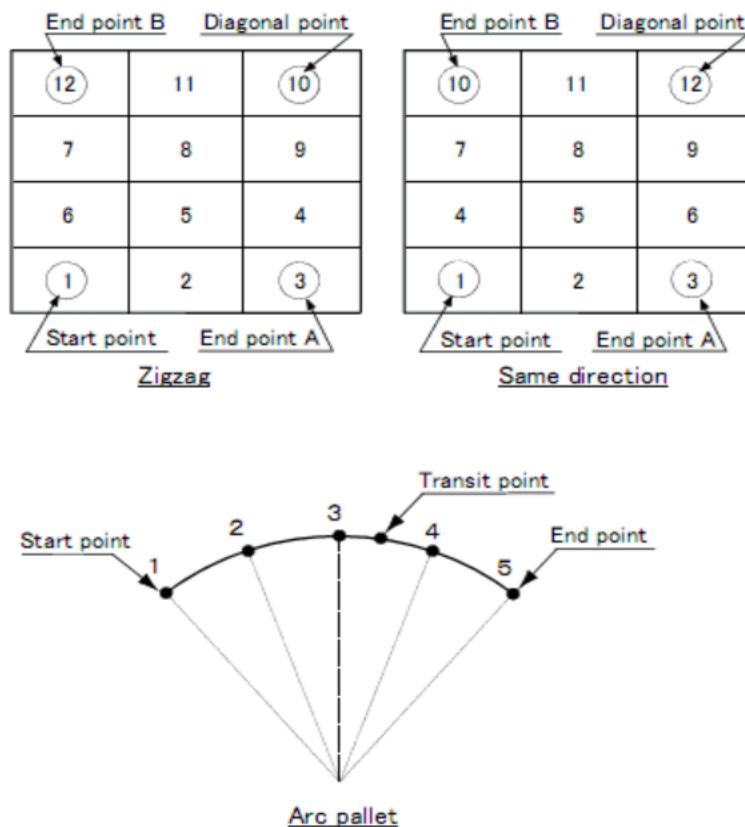
---

## 2. [Format]

Def[] Plt[] <Pallet No.>, <Start Point>, <End Point A>, <End Point B>, [<Diagonal Point>], <Quantity A>, <Quantity B>, <Pallet pattern>

### <Pallet Pattern>

- 1 : Zigzag(posture equal division)
- 2 : Same direction (posture equal division)
- 3 : Arc pallet(posture equal division)
- 11 : Zigzag(posture fixation)
- 12 : Same direction (posture fixation)
- 13 : Arc pallet(posture fixation)



---

## **I. DÀNH CHO SINH VIÊN**

### **1. Chương trình**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

### **2. Giải thích chương trình**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

## **II. DÀNH CHO CBHD THỰC HÀNH.**

### **1. Nhận xét:**

.....

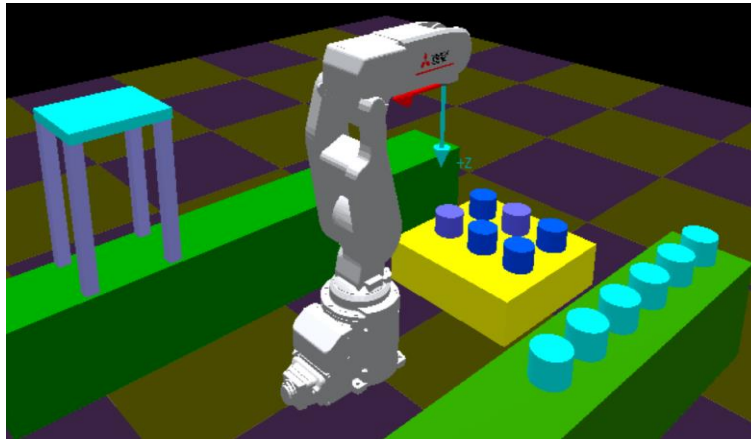
---

.....  
.....  
.....  
**2. Điểm đánh giá:** .....

**3. Chữ ký của giảng viên:** .....

---

**Bài 6** Viết chương trình điều khiển Robot sắp 6 phôi trên băng tải lên khay đựng phôi



## **I. DÀNH CHO SINH VIÊN**

### **1. Chương trình**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

### **2. Giải thích chương trình**

.....

.....

---

.....

.....

.....

.....

.....

.....

.....

.....

**II. DÀNH CHO CBHD THỰC HÀNH.**

**1. Nhận xét:**

.....

.....

.....

.....

**2. Điểm đánh giá: .....**

**3. Chữ ký của giảng viên: .....**

---

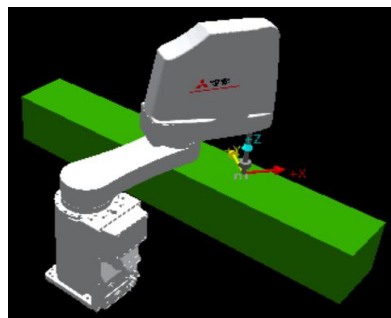
## BÀI 3

### ĐIỀU KHIỂN ROBOT MELFA RH-3FRH5515-Q



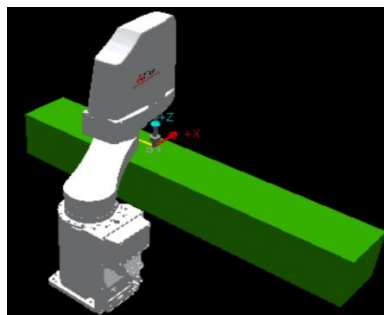
## B. THỰC HÀNH

**Bài 3.1** Thực hiện thao tác di chuyển Robot từ điểm **Start position** đến **Target position** trên panel điều khiển



X:	-179.626	0.000
Y:	-509.960	0.000
Z:	221.643	0.000
A:	0.000	0.000
B:	0.000	0.000
C:	-128.886	0.000

**Start position**



X:	179.626	0.000
Y:	-509.960	0.000
Z:	221.643	0.000
A:	0.000	0.000
B:	0.000	0.000
C:	-88.786	0.000

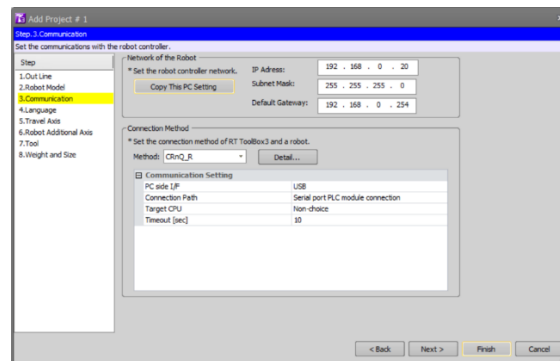
**Target position**

### Hướng dẫn:

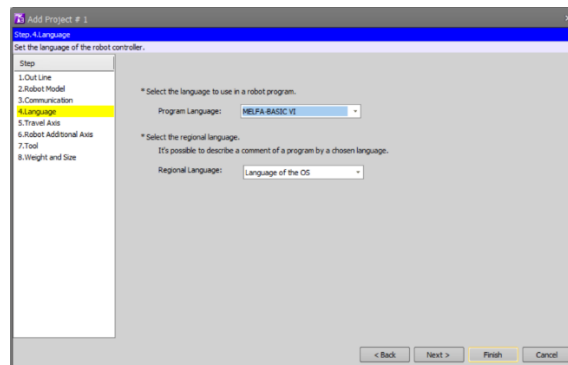
1. Chạy phần mềm RT-Toolbox 3: Chọn New



## 5. Chọn kết nối RT Toolbox 3 – Robot (mặc định)

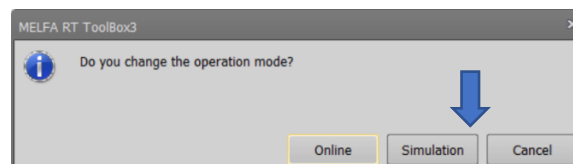


## 6. Chọn ngôn ngữ lập trình



## 7. Chọn Finish

## 8. Chọn **Simulation**



## 9. Mở cửa sổ **Operation Panel** → nhập các giá trị X,Y,Z để tạo các điểm: **Start position, Target position**



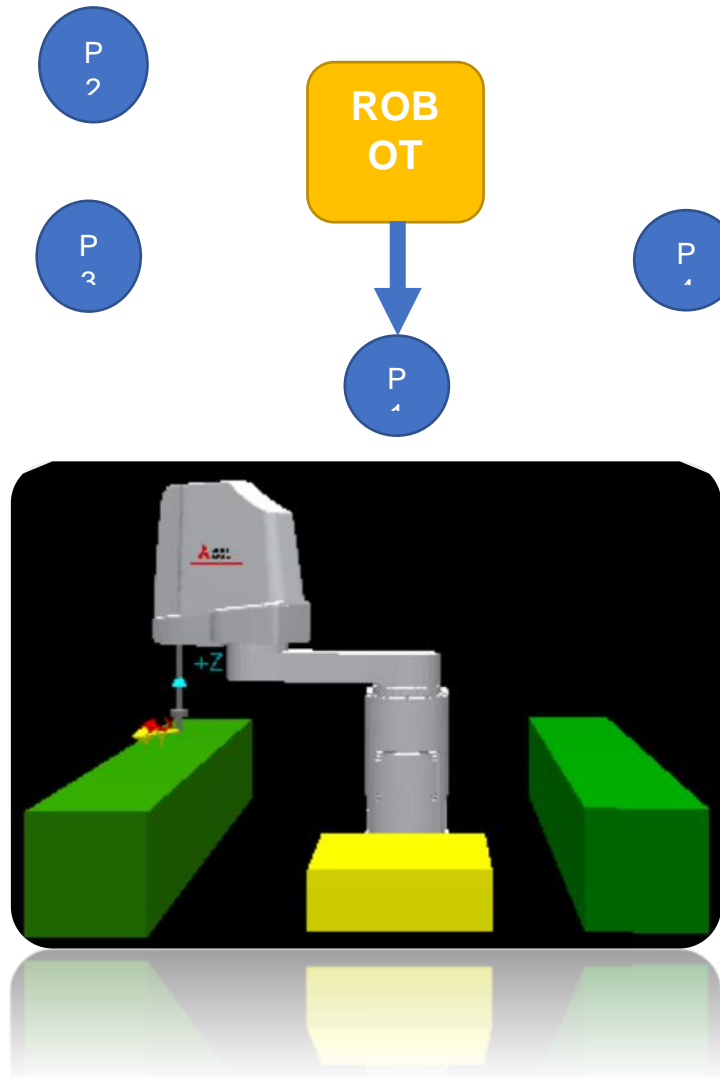


X:	-179.626	0.000
Y:	-509.960	0.000
Z:	221.643	0.000
A:	0.000	0.000
B:	0.000	0.000
C:	-128.886	0.000

X:	179.626	0.000
Y:	-509.960	0.000
Z:	221.643	0.000
A:	0.000	0.000
B:	0.000	0.000
C:	-88.786	0.000

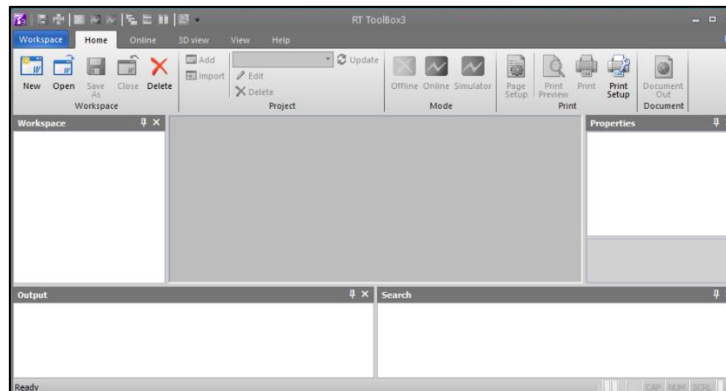
---

**Bài 3.2:** Điều khiển Robot RH-3FRH di chuyển đến các điểm P1(Home), P2, P3, P4

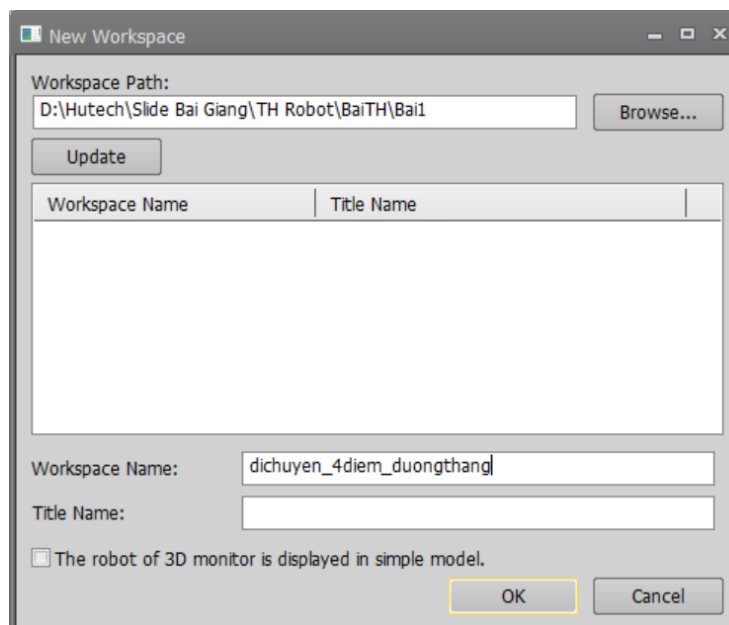


## Hướng dẫn:

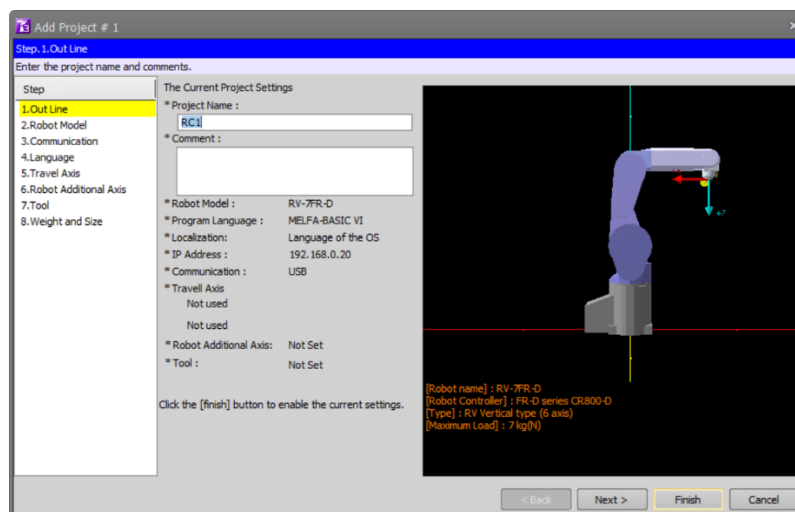
### 1. Chạy phần mềm RT-Toolbox 3: Chọn New



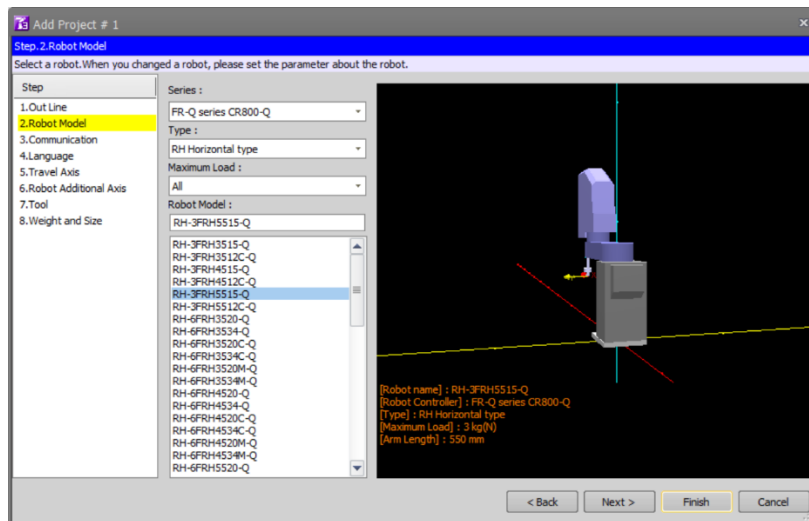
### 2. Đặt tên **Workspace Name:** Name



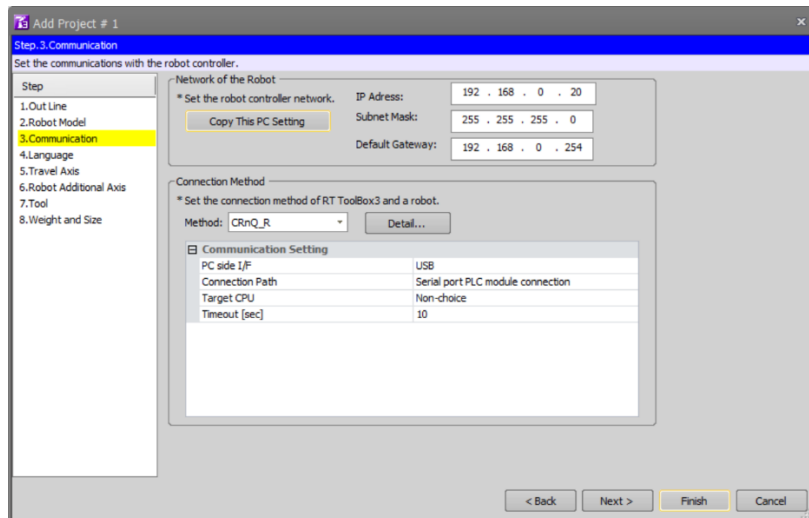
### 3. Đặt tên project (mặc định RC1)



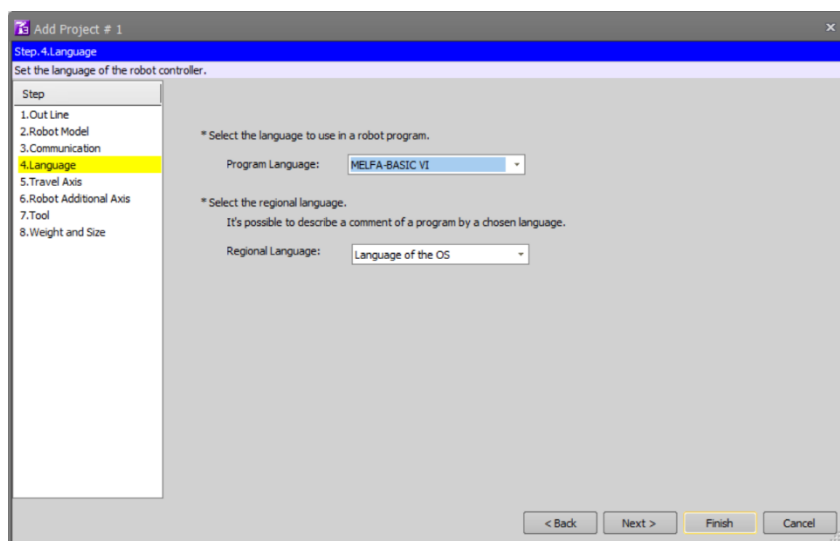
#### 4. Chọn Controller và loại robot: Theo hình



#### 5. Chọn kết nối RT Toolbox 3 – Robot (mặc định) → Chọn **Next**

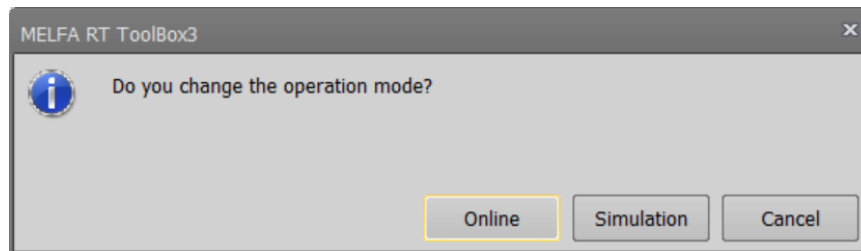


#### 6. Chọn ngôn ngữ lập trình



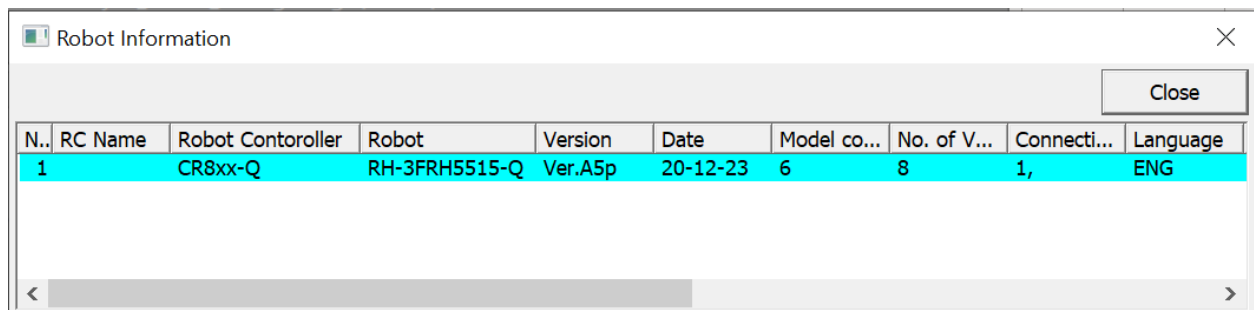
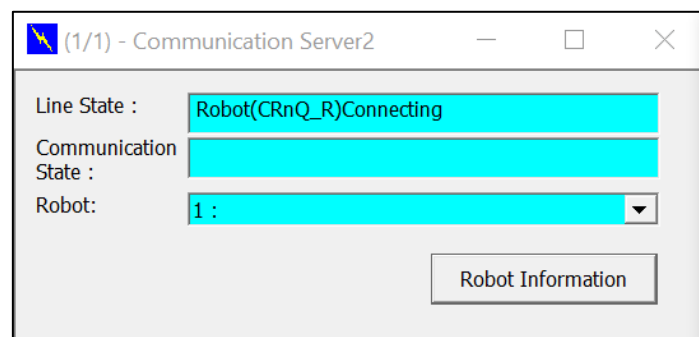
7. Chọn Finish

8. Chọn **Simulation**



9. Trạng thái kết nối RT Toolbox 3 ⇔ Robot

10. Kết nối thành công



**Chú ý:** Không để các vật cản trên đường di chuyển của tay Robot

11. Di chuyển J1, J2, J3, J4 đến vị trí điểm P1 theo bảng giá trị

J1 :	-86.488	-	+
J2 :	-1.595	-	+
J3 :	369.495	-	+
J4 :	359.381	-	+

- Nhấn + hoặc - để thay đổi

12. Di chuyển J1, J2, J3, J4 đến vị trí điểm P2 theo bảng giá trị

---

J1:	-51.199	-	+
J2:	-35.101	-	+
J3:	359.495	-	+
J4:	359.381	-	+

- Nhấn + hoặc - để thay đổi

13. Di chuyển J1, J2, J3, J4 đến vị trí điểm P3 theo bảng giá trị

J1:	46.093	-	+
J2:	28.497	-	+
J3:	359.495	-	+
J4:	359.381	-	+

- Nhấn + hoặc - để thay đổi

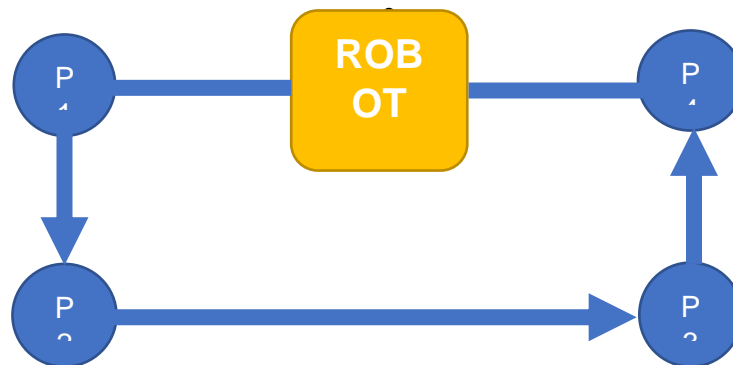
14. Di chuyển J1, J2, J3, J4 đến vị trí điểm P4 theo bảng giá trị

J1:	83.940	-	+
J2:	57.116	-	+
J3:	359.495	-	+
J4:	359.381	-	+

- Nhấn + hoặc - để thay đổi

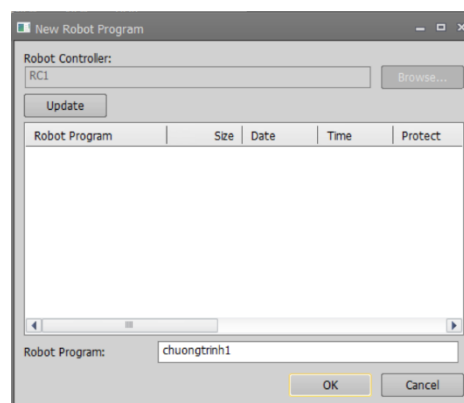
---

**Bài 3.3:** Điều khiển Robot RH-3FRH di chuyển theo ***đường thẳng*** từ điểm P1 → P2 → P3 → P4



**Hướng dẫn:**

**Bước 1:** Tạo New Robot Program → đặt tên "chuongtrinh1" → OK



## Bước 2: Thêm 4 điểm P1, P2, P3, P4 vào cửa sổ Position

Name	J1	J2	J3	J4	J5	J6	J7	J8
P1	-86.488	-1.595	369.495	359.381	X	X	X	X
P2	-51.199	-35.101	369.495	359.381	X	X	X	X
P3	46.093	28.497	369.495	359.381	X	X	X	X
P4	83.940	57.116	369.495	359.381	X	X	X	X

XYZ Alt+X   Joint Alt+J   Work Coordinate Alt+W

## Bước 3: Cập nhật vị trí J1, J2, J3, J4 theo 4 điểm P1, P2, P3, P4

Position Jump

Robot Coordinates in Selection   Offset:

J1: -51.199   0.000

J2: -35.101   0.000

J3: 369.495   0.000

J4: 359.381   0.000

Offset Reset

Interpolation: Joint (Mov)

Move   Cancel

- Điểm P1

J1: -86.488   -   +

J2: -1.595   -   +

J3: 369.495   -   +

J4: 359.381   -   +

- Điểm P2

J1: -51.199   -   +

J2: -35.101   -   +

J3: 369.495   -   +

J4: 359.381   -   +

- Điểm P3

J1: 46.093   -   +

J2: 28.497   -   +

J3: 369.495   -   +

J4: 359.381   -   +



- Điểm P4

J1:	83.940	-	+
J2:	57.116	-	+
J3:	369.495	-	+
J4:	359.381	-	+

Kết quả sau khi cập nhật vị trí các điểm P1, P2, P3, P4

Name	J1	J2	J3	J4	J5	J6	J7	J8
P1	-86.488	-1.595	369.495	359.381	X	X	X	X
P2	-51.199	-35.101	369.495	359.381	X	X	X	X
P3	46.093	28.497	369.495	359.381	X	X	X	X
P4	83.940	57.116	369.495	359.381	X	X	X	X

XYZ Alt+X Joint Alt+J Work Coordinate Alt+W

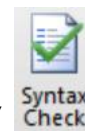
**Bước 4:** Viết chương trình cho Robot di chuyển theo đường thẳng từ điểm P1 → P2 → P3 → P4 và dừng lại.

```

Program 1:RC1 CHUONGTRINH1... x
1 Servo On
2 Ovrd 50
3 Mvs P1
4 Dly 0.5
5 Mvs P2
6 Dly 0.5
7 Mvs P3
8 Dly 0.5
9 Mvs P4
10 Dly 0.5
11 Servo Off
12 Hlt
13

```

**Bước 5:** Kiểm tra lỗi

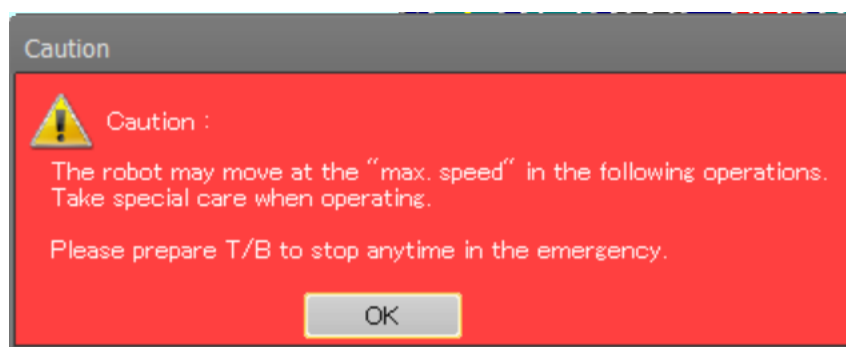


Sau khi viết code nhấn biểu tượng "Syntax Check" để kiểm tra lỗi. Nếu chương trình không có lỗi sẽ hiển thị cửa sổ như hình.



**Bước 6:** Chạy chương trình

Cửa sổ cảnh báo xuất hiện. Nhấn OK



**Chú ý:**

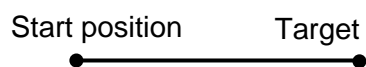
1. Kiểm tra khu vực chuyển động của Robot
2. Khi Robot hoạt động không đưa các vật cản vào khu vực chuyển động của Robot
3. Không để bất kỳ phần nào của cơ thể vào khu vực chuyển động của Robot để tránh các tai nạn.

---

# BÁO CÁO THỰC HÀNH

Ngày ..... tháng ..... năm .....

**Bài tập 1** Viết chương trình điều khiển Robot di chuyển theo yêu cầu sau:



## I. DÀNH CHO SINH VIÊN

### 3. Chương trình

.....

.....

.....

.....

.....

.....

.....

.....

### 4. Giải thích chương trình

.....

.....

.....

.....

.....

## II. DÀNH CHO CBHD THỰC HÀNH.

### 4. Nhận xét:

---

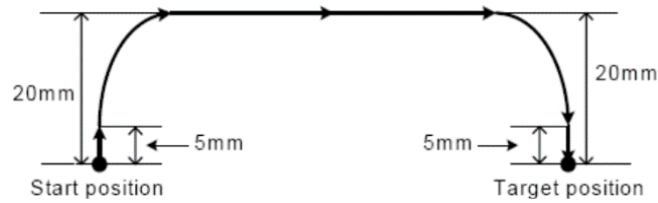
.....  
.....  
.....

**5. Điểm đánh giá:** .....

**6. Chữ ký của giảng viên:** .....

---

**Bài tập 2** Viết chương trình điều khiển Robot di chuyển theo yêu cầu sau:



## I. DÀNH CHO SINH VIÊN

### 5. Chương trình

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

### 6. Giải thích chương trình

.....

.....

.....

.....

.....

---

.....

.....

.....

.....

.....

**II. DÀNH CHO CBHD THỰC HÀNH.**

**4. Nhận xét:**

.....

.....

.....

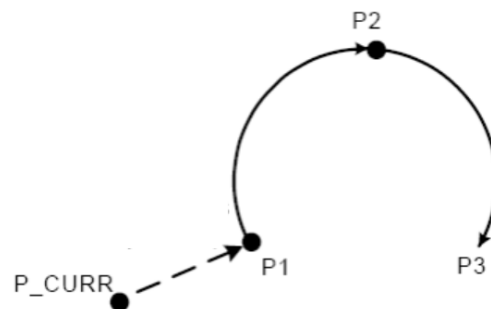
.....

**5. Điểm đánh giá:** .....

**6. Chữ ký của giảng viên:** .....

---

**Bài tập 3** Viết chương trình điều khiển Robot di chuyển theo yêu cầu sau:



**I. DÀNH CHO SINH VIÊN**

**3. Chương trình**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

**4. Giải thích chương trình**

.....

.....

.....

.....

.....

.....

---

.....

.....

.....

.....

**II. DÀNH CHO CBHD THỰC HÀNH.**

**4. Nhận xét:**

.....

.....

.....

.....

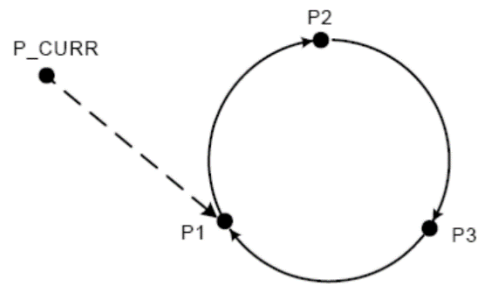
**5. Điểm đánh giá: .....**

**6. Chữ ký của giảng viên: .....**



---

**Bài tập 4** Viết chương trình điều khiển Robot di chuyển theo yêu cầu sau:



## I. DÀNH CHO SINH VIÊN

### 3.Chương trình

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

### 4. Giải thích chương trình

.....

.....

.....

.....

.....

.....

---

.....

.....

.....

.....

**II. DÀNH CHO CBHD THỰC HÀNH.**

**1. Nhận xét:**

.....

.....

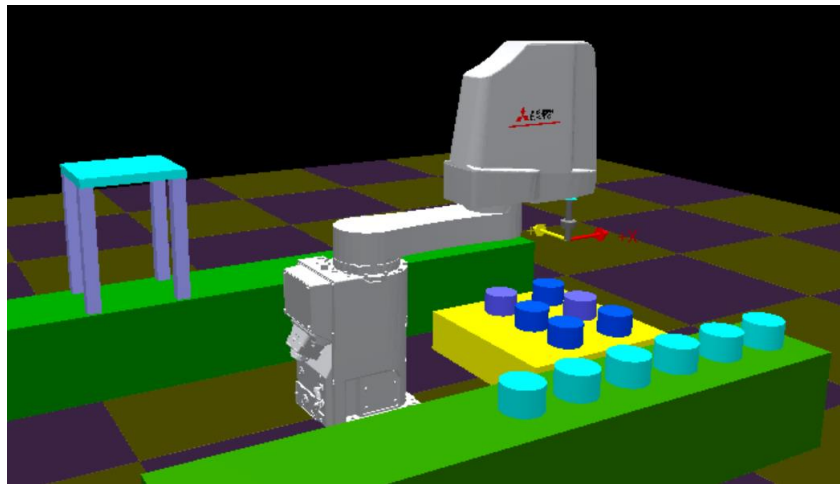
.....

.....

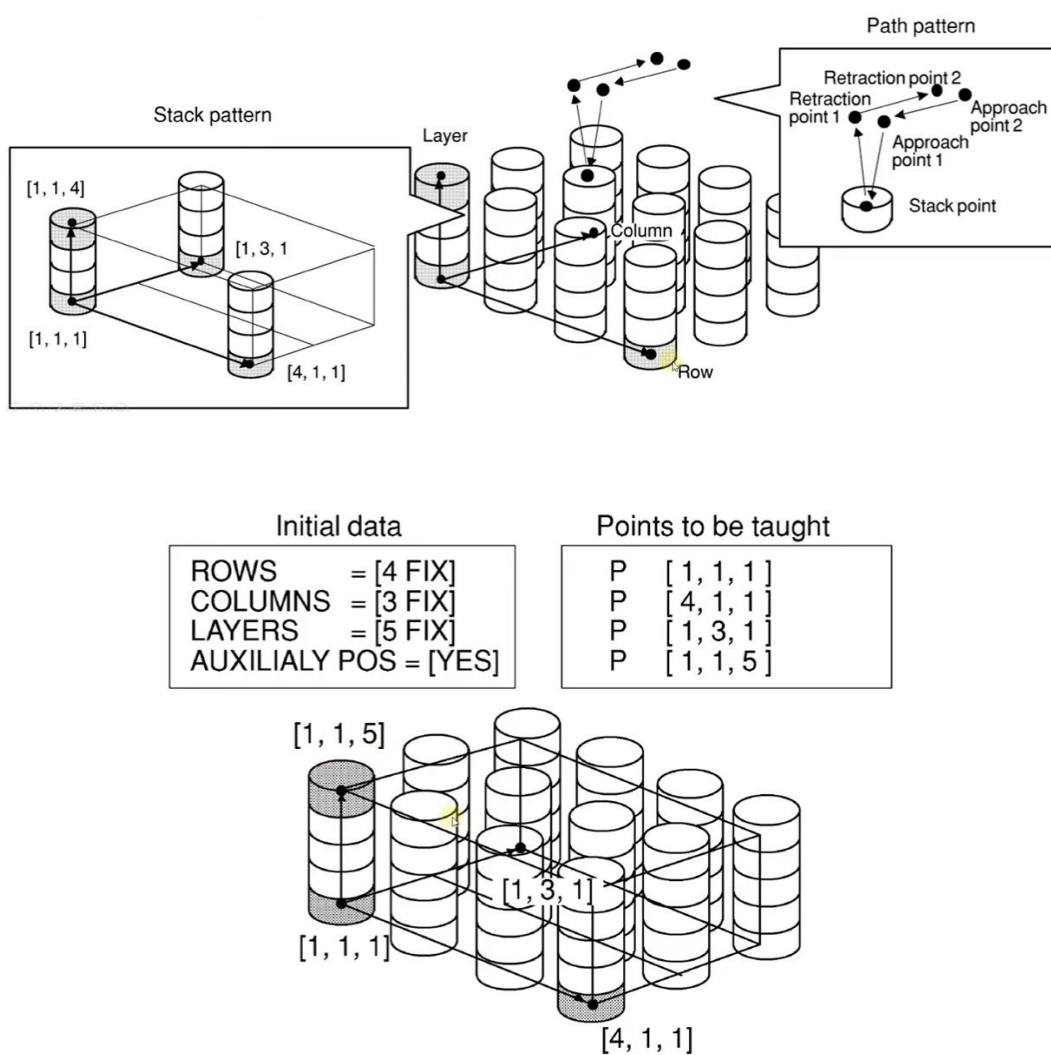
**2. Điểm đánh giá: .....**

**3. Chữ ký của giảng viên: .....**

## Bài 5 Viết chương trình điều khiển Robot sắp 6 phôi lên băng tải



### Hướng dẫn:



### 3. [Function]

---

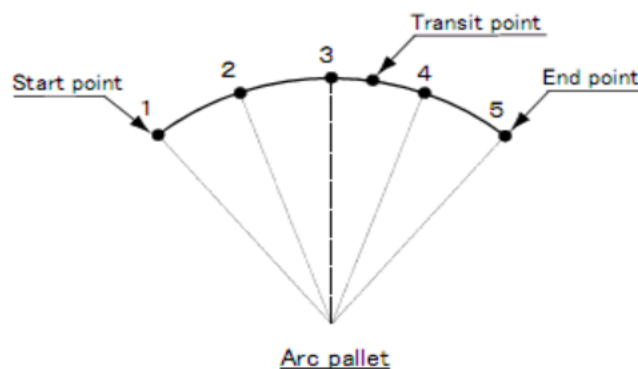
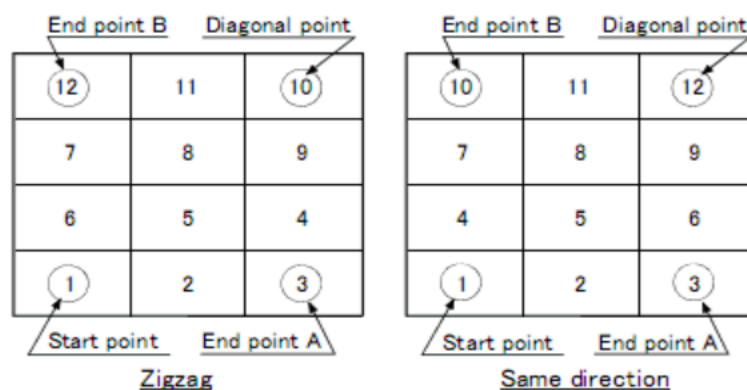
Defines the pallet. (3-point pallet, 4-point pallet)

#### 4. [Format]

Def[] Plt[] <Pallet No.>, <Start Point>, <End Point A>, <End Point B>, [<Diagonal Point>], <Quantity A>, <Quantity B>, <Pallet pattern>

##### <Pallet Pattern>

- 1 : Zigzag(posture equal division)
- 2 : Same direction (posture equal division)
- 3 : Arc pallet(posture equal division)
- 11 : Zigzag(posture fixation)
- 12 : Same direction (posture fixation)
- 13 : Arc pallet(posture fixation)



---

## **I. DÀNH CHO SINH VIÊN**

### **1. Chương trình**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

### **2. Giải thích chương trình**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

## **II. DÀNH CHO CBHD THỰC HÀNH.**

### **1. Nhận xét:**

.....

---

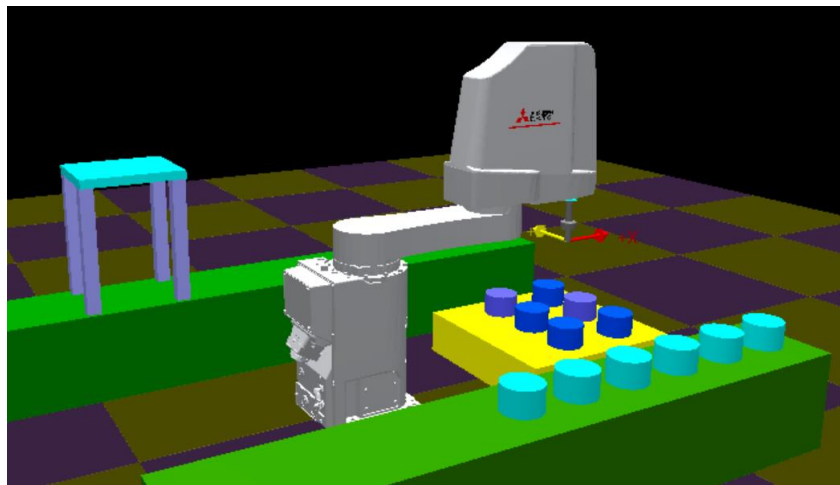
.....  
.....  
.....

**2. Điểm đánh giá: .....**

**3. Chữ ký của giảng viên: .....**

---

**Bài 6** Viết chương trình điều khiển Robot sắp 6 phôi trên băng tải lên khay đựng phôi



## **I. DÀNH CHO SINH VIÊN**

### **1. Chương trình**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

### **2. Giải thích chương trình**

.....

---

.....

.....

.....

.....

.....

.....

.....

.....

.....

**II. DÀNH CHO CBHD THỰC HÀNH.**

**1. Nhận xét:**

.....

.....

.....

.....

**2. Điểm đánh giá: .....**

**3. Chữ ký của giảng viên: .....**

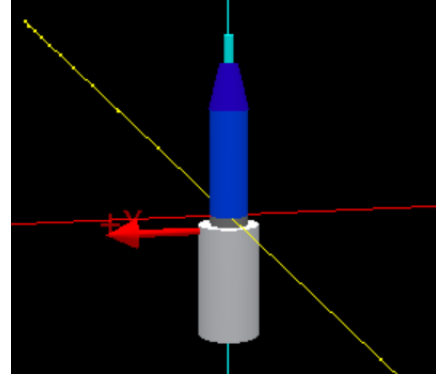
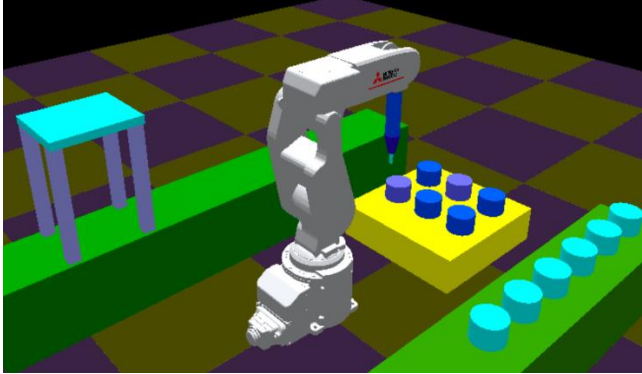


---

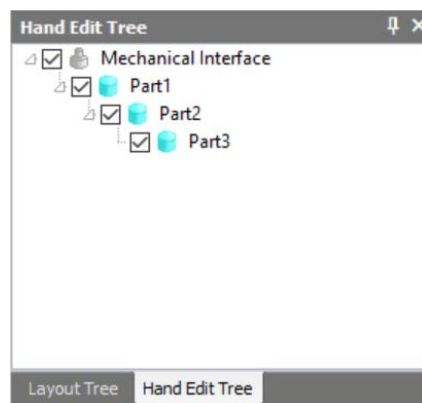
## BÀI 4

### TẠO HAND - TOOL

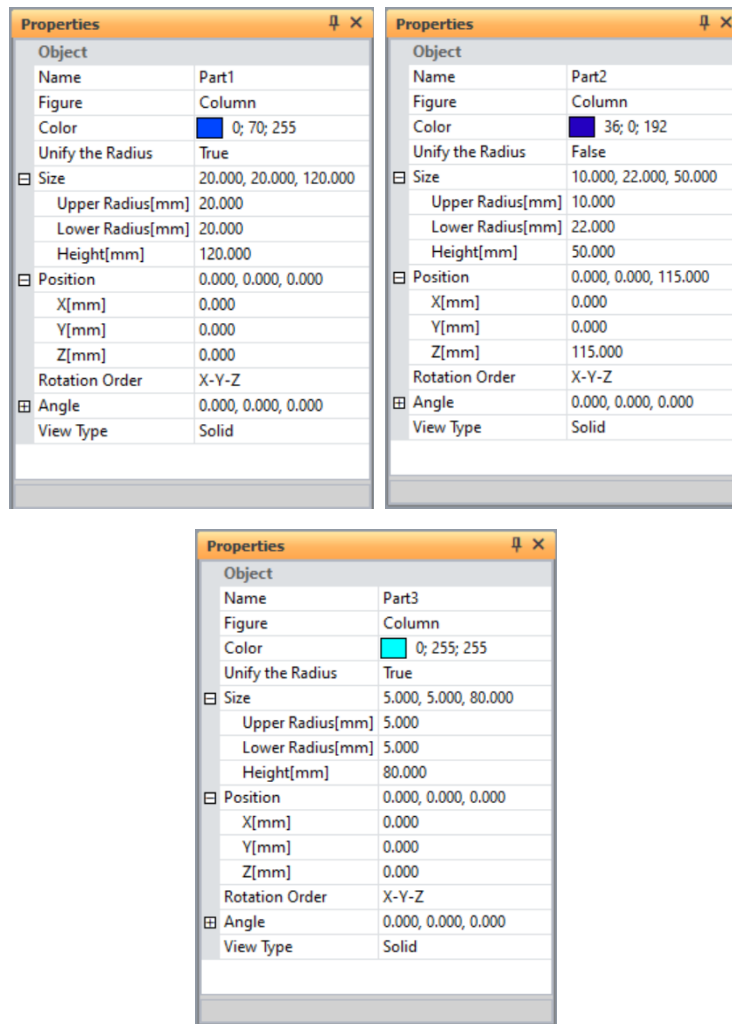
#### 4.1 Tạo Tool (bút vẽ) như hình



#### 1. Tạo 3 Part

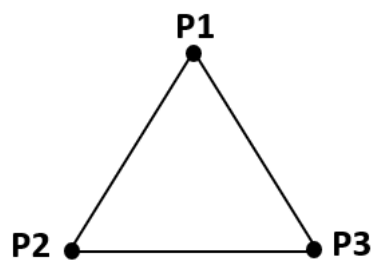


#### 2. Thông số Part

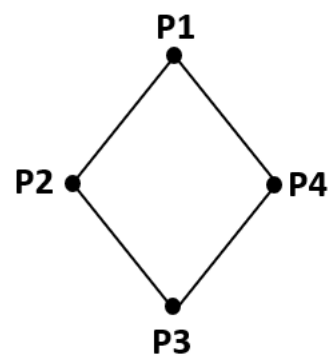
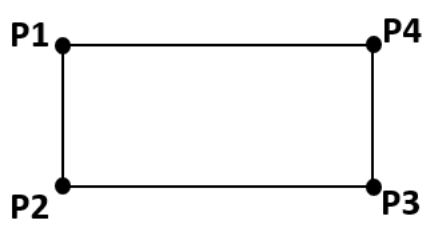
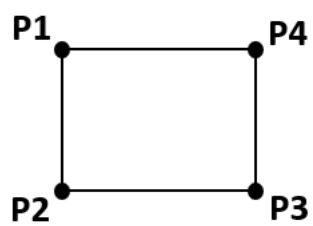


## 4.2 Viết chương trình di chuyển theo yêu cầu sau

a.  $P1 \rightarrow P2 \rightarrow P3$



b.  $P1 \rightarrow P2 \rightarrow P3 \rightarrow P4$



c. Điều khiển Robot vẽ logo Mitsubishi

