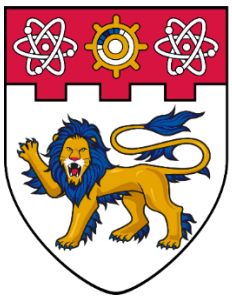

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

MA4825 Robotics Assignment 1

(RobotStudio Simulation)

Lee Yong Ern, Gabriel (U2121331L)

Submission Date: 19 November 2024

Tutor: Dr. Ahmad K. Bin Ratmin

1. List of Models Used

Functional Components

Component Name	Filename	Source
Raspberry Pi 4	Raspberry Pi 4 Model B.SAT	GrabCAD
Raspberry Pi 4 USB2.0 Ports	RPi4 2X USB2.0 PORTS.SAT	GrabCAD
Raspberry Pi 4 GPIO Header Pins	RPi4 GPIO.SAT	GrabCAD

Auxiliary Models (for modelling pick-and-place)

Component Name	Filename	Source
Conveyor	Static_Conveyor.sat	NTULearn / MA4825 Course Site
Robot Base Platform	Static_Platform.sat	Self-Modelled on SolidWorks
Workbench for Loose Components	Static_Table.sat	Self-Modelled on SolidWorks
Raspberry Pi 4 without GPIO and USB	Raspberry Pi 4 Model B_both missing.SAT	GrabCAD; Edited on SolidWorks
Raspberry Pi 4 without GPIO	Raspberry Pi 4 Model B_GPIO missing.SAT	GrabCAD; Edited on SolidWorks

2. Robot Selection and Justification

Figure 1: ABB IRB 120 Model

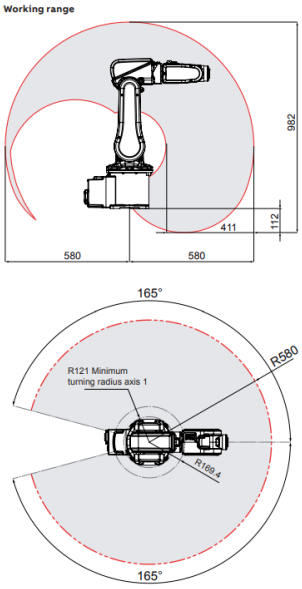

Figure 2: ABB IRB 120 Working Range (Source: ABB Library, 2019)

The **ABB IRB 120** is a compact, 6-DoF robotic arm known for its precision, speed, and versatility, making it suitable for applications like electronics assembly, machine tending, and laboratory automation.

- **Repeatability**: The IRB 120 has a repeatability of ±0.01 mm (10 microns), which is excellent for high-precision tasks like placing small electronic components on a PCB, which has tolerances of about 0.5-1 mm.
- **Payload**: It can handle payloads up to 3 kg (or 4 kg with the wrist down), which is more than enough for handling small electronic components.
- **Reach**: It has a reach of 580 mm, which is appropriate for small workspace environments typical in electronics assembly.

End Effector (Gripper):

For electronics assembly, including tasks like handling small components for Raspberry Pi boards, choosing the right gripper is crucial. The requirements for a suitable gripper include precision, gentle handling to avoid damaging delicate components, and sometimes anti-static properties to protect sensitive electronics.

To meet the requirements of gripping both a GPIO Header Pin and USB Connector, a Two-Finger gripper is most appropriate for this application. The ABB RobotStudio's SmartGripper provides a simulated gripper tool used for digital twins and virtual prototyping, and is useful for testing and validating pick-and-place sequences. However, it does not fully represent the hardware capabilities needed for real-world electronics handling. The SmartGripper provides a basic simulation of grip and release functionality but lacks features necessary for delicate electronic assembly, such as Fine Control of Grip Force and ESD (Electrostatic Discharge) Protection.

In reality, other feasible options include Vacuum Grippers for flat, smooth components, or Electrostatic Grippers for sensitive electronics.

3. Description of Pick and Place Task

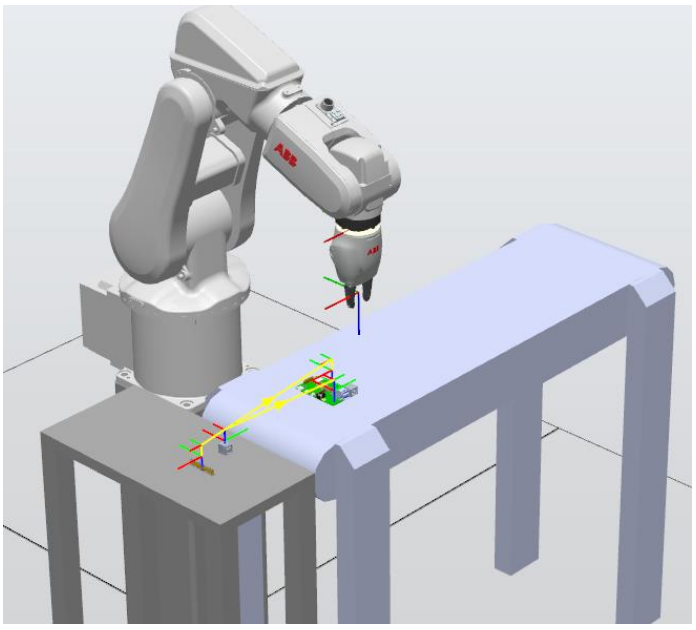

Figure 3: RobotStudio Simulation (in Home Position)

The following targets were created:

- **HomePos**: Starting position of the robot
- **USB**: Position of the loose USB component on the workbench
- **USB_onboard:** Final position of the USB component onboard the RPI4
- **GPIO**: Position of the loose GPIO connector on the workbench
- **GPIO_onboard:** Final position of the GPIO connector onboard the RPI4

The robot follows the following sequence of steps, which have also been grouped into functions, as they perform similar steps for the respective components:

Function Name	**Steps**
Main	1. Go to HomePos and ensure gripper is open. 2. Execute Path_USB function 3. Execute Path_GPIO function 4. Go to HomePos and ensure gripper is open.
Path_USB	1. Go to 20mm offset above the USB component. 2. At a slower speed, approach the USB component, then close the gripper. 3. Move 20mm up, then go to 20mm offset above the desired position (USB_onboard). 4. At a slower speed, approach the final position (USB_onboard), then open the gripper. 5. Move 20mm up.
Path_GPIO	1. Go to 20mm offset above the GPIO component. 2. At a slower speed, approach the GPIO component, then close the gripper. 3. Move 20mm up, then go to 20mm offset above the desired position (GPIO_onboard). 4. At a slower speed, approach the final position (GPIO_onboard), then open the gripper. 5. Move 20mm up.

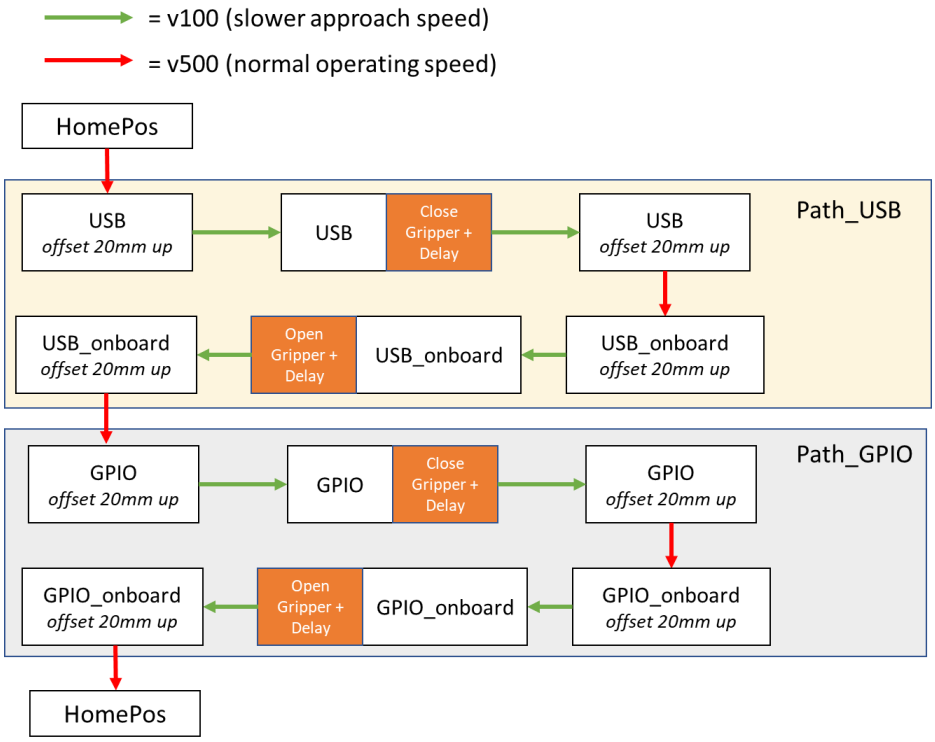

Figure 4: Flowchart of Robot's Instructions

To optimize for speed and accuracy, 0.5-second delays were introduced at specific points in the robot's motion, such as when the component is being picked and placed. Additionally, the robot's movement speed is significantly reduced (from 500 to 100) in the 2cm proximity around the components, which ensures the precise placement of components without risking interference with other components.

After optimizing for speed and accuracy, the current sequence of tasks has a cycle time of 7.656 seconds.

References

Free CAD designs, Files & 3D models: The grabcad community library. GrabCAD Community Library. (n.d.). https://grabcad.com/library/raspberry-pi-4-model-b-1

Product specification - IRB 120. ABB Library. (2019). https://library.e.abb.com/public/6aed5e91083f4fceb358eea2fe4c1bab/3HAC035960%20PS%20IRB%20120-en.pdf?x-sign=edIex5StIjpgmJBJJ95tak9NHdyuuut6mzzJHESGKt5i1JG8dhLRBdvggKptgBJn

Appendix I: RAPID Code

```
MODULE Module1
    CONST robtarget Home:=[[395.929,-2.572,306.197],[0,-0.707106781,0.707106781,0],[-
1,0,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget GPIO:=[[349.735,-301.105,125],[0,-0.707106781,0.707106781,0],[-1,0,-
2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget USB:=[[341.269,-243.994,123.933],[0,0,1,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget USB_on_board:=[[357.259,-19.318,123.7],[0,0,1,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget GPIO_on_board:=[[310.452,24.2,125.2],[0,-
0.707106781,0.707106781,0],[0,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];


    PROC main()
        SetDO Gripper_USB,0;
        SetDO Gripper_GPIO,0;
        SetDO RPI_Empty,1;
        SetDO RPI_GPIOMissing,0;
        SetDO RPI_Full,0;
        SetDO Comp_GPIO,1;
        SetDO Comp_USB,1;
        SetDO GripClose_GPIO, 0;
        SetDO GripClose_USB, 0;
        SetDO GripOpen, 1;

        MoveJ Home,v200,z100,Servo\WObj:=wobj0;

        Path_USB;
        Path_GPIO;
        MoveJ Home,v500,z100,Servo\WObj:=wobj0;

    ENDPROC

    PROC Path_USB()
        MoveL offs(USB,0,0,20),v500,fine,Servo\WObj:=wobj0;
        MoveJ USB,v100,z100,Servo\WObj:=wobj0;
        WaitTime 0.7;
        SetDO GripClose_USB, 1;
        ! USB Transfer from Table to Gripper
        SetDO Gripper_USB,1;
        SetDO Comp_USB,0;
        WaitTime 0.2;
        MoveL offs(USB,0,0,20),v200,fine,Servo\WObj:=wobj0;
        MoveL offs(USB_on_board,0,0,20),v500,fine,Servo\WObj:=wobj0;
        MoveJ USB_on_board,v200,z100,Servo\WObj:=wobj0;
        WaitTime 0.5;
        ! USB Transfer from Gripper to Board
```

```
    SetDO Gripper_USB,0;
    SetDO RPI_GPIOMissing,1;
    WaitTime 0.2;
    SetDO GripClose_GPIO, 0;
    SetDO GripClose_USB, 0;
    SetDO GripOpen, 1;
    WaitTime 0.2;
    MoveL offs(USB_on_board,0,0,20),v200,fine,Servo\WObj:=wobj0;
  ENDPROC

  PROC Path_GPIO()
    MoveL offs(GPIO,0,0,20),v500,fine,Servo\WObj:=wobj0;
    MoveJ GPIO,v100,z100,Servo\WObj:=wobj0;
    WaitTime 0.5;
    SetDO GripClose_GPIO, 1;
    ! GPIO Transfer from Table to Gripper
    SetDO Gripper_GPIO,1;
    SetDO Comp_GPIO,0;
    SetDO RPI_Empty,1;
    WaitTime 0.2;
    MoveL offs(GPIO,0,0,20),v200,fine,Servo\WObj:=wobj0;
    MoveL offs(GPIO_on_board,0,0,20),v500,fine,Servo\WObj:=wobj0;
    MoveJ GPIO_on_board,v200,z100,Servo\WObj:=wobj0;
    WaitTime 0.5;
    ! GPIO Transfer from Gripper to Board
    SetDO Gripper_GPIO,0;
    SetDO RPI_Full,1;
    WaitTime 0.2;
    SetDO GripClose_GPIO, 0;
    SetDO GripClose_USB, 0;
    WaitTime 0.2;
    SetDO GripOpen, 1;
    MoveL offs(GPIO_on_board,0,0,20),v200,fine,Servo\WObj:=wobj0;
  ENDPROC


ENDMODULE
```