**PDE4431 – Robot Manipulation**

Coursework – 01

Middlesex University Dubai

**N. G. Jayashanka Anushan,**

**MISIS – M01037028**

**MSc in Robotics (2024 / 2025 Year-01)**

# Table of Contents

[Table of Contents 2](#_Toc182242143)

[1). Introduction 5](#_Toc182242144)

[2). Task Explanation 7](#_Toc182242145)

[3). Achieve the task 8](#_Toc182242146)

[3.1). Planning and buildup the flow of task 8](#_Toc182242147)

[3.2). Define the base coordinates and measure the rest of the coordinates from base coordinates. 9](#_Toc182242148)

[3.3). Build on virtual environment 10](#_Toc182242149)

[3.4). Connect the Robot 14](#_Toc182242150)

[3.5). Setting up the tools 15](#_Toc182242151)

[3.6). Setting up the frame 17](#_Toc182242152)

[3.7). Initialize / Record the base coordinates 18](#_Toc182242153)

[3.8). Written functions 19](#_Toc182242154)

[3.8.1). Function for Letter “D” 19](#_Toc182242155)

[3.8.2). Function for Letter “X” 19](#_Toc182242156)

[3.8.3). Function for Letter “B” 20](#_Toc182242157)

[3.8.4). Function for the frame 20](#_Toc182242158)

[3.9). Used functions to develop the program 21](#_Toc182242159)

[4). Task preparation on the sheet in final 21](#_Toc182242160)

[5). Task Demonstration 22](#_Toc182242161)

[6). Conclusion 22](#_Toc182242162)

***List of Figures***

[Figure 1 Epson (Scara) T3 Robot Source: (Author Developed) 5](#_Toc182242163)

[Figure 2 Epson VT-6 Robot Source: (Author Developed) 6](#_Toc182242164)

[Figure 3 Epson RC 7.4.3 Programming Software Source:(Author Developed) 7](#_Toc182242165)

[Figure 4 Task Preparation - Base on coordinates Source: (Author Developed) 8](#_Toc182242166)

[Figure 5 Point define example Source: (Author Developed) 9](#_Toc182242167)

[Figure 6 Task preparation sheet with base coordinates Source: (Author developed) 10](#_Toc182242168)

[Figure 7 Setup connection Source: (Author developed) 11](#_Toc182242169)

[Figure 8 Added a robot to the environment Source: (Author developed) 11](#_Toc182242170)

[Figure 9 Robot simulator Source: (Author developed) 12](#_Toc182242171)

[Figure 10 Robot manager window Source: (Author developed) 12](#_Toc182242172)

[Figure 11 Points define Source: (Author developed) 13](#_Toc182242173)

[Figure 12 Jog and Teach the points Source: (Author developed) 13](#_Toc182242174)

[Figure 13 Programming Source: (Author developed) 14](#_Toc182242175)

[Figure 14 Custom defined functions call in main function Source: (Author developed) 14](#_Toc182242176)

[Figure 15 Connect the robot Source: (Author developed) 15](#_Toc182242177)

[Figure 16 Setup Tool Source: (Author developed) 16](#_Toc182242178)

[Figure 17 Teach the points to setup the tool Source: (Author developed) 16](#_Toc182242179)

[Figure 18 Teach origin, X and Y for local frame Source: (Author developed) 17](#_Toc182242180)

[Figure 19 Setup Local frame Source: (Author developed) 17](#_Toc182242181)

[Figure 20 Teach Origin, X and Y for local frame Source: (Author Developed) 18](#_Toc182242182)

[Figure 21 Teach the base points of each letter and the frame Source: (Author developed) 19](#_Toc182242183)

[Figure 22 Task preparation source: (Author developed) 21](#_Toc182242184)

***List of Tables***

[Table 1 Defined points to the task Source: (Author developed) 9](#_Toc182242185)

# Introduction

PDE4430 – Robot Manipulation module will be taught to use and program state-of-the-art robot arms and end effectors, such as those employed in car production lines, CNC machines, medical research, pick and place operations. Such machines are generally mounted in fixed positions and can move with speed or with load-carrying capability and precision. Programming an industrial robot is a challenge to bring the real-world purpose to real experience.

In this coursework we have been assigned to use one of following robots to complete the task as per the facilities provided by the university.

* Epson VT6 – (6 Axis)
* Epson T3 – (Scara)



Figure Epson (Scara) T3 Robot Source: (Author Developed)



Figure Epson VT-6 Robot Source: (Author Developed)

# Task Explanation

Task has been declared to move / draw the robot on given path. Path programming concept to be applied to the coursework aligned with tool setting and frame setting (if required). Randomly paper has been selected by the lecture and distributed to the student for implement the received design. Can be use all the lessons that taught throughout the module and demonstration required.

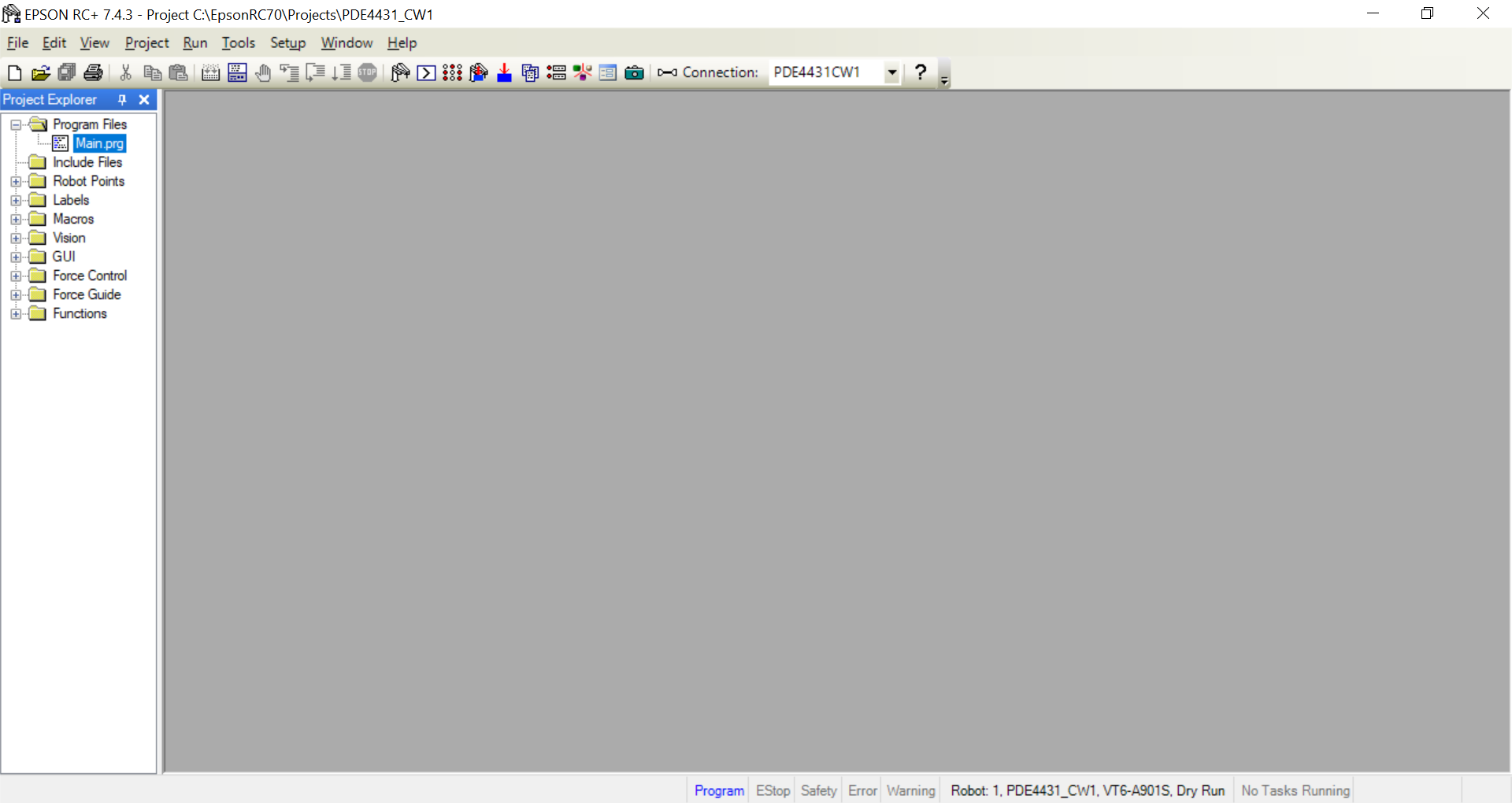


Figure Epson RC 7.4.3 Programming Software Source:(Author Developed)

# Achieve the task

Distributed papers contained printed designs of three letters and a frame as follows to do the path programming.

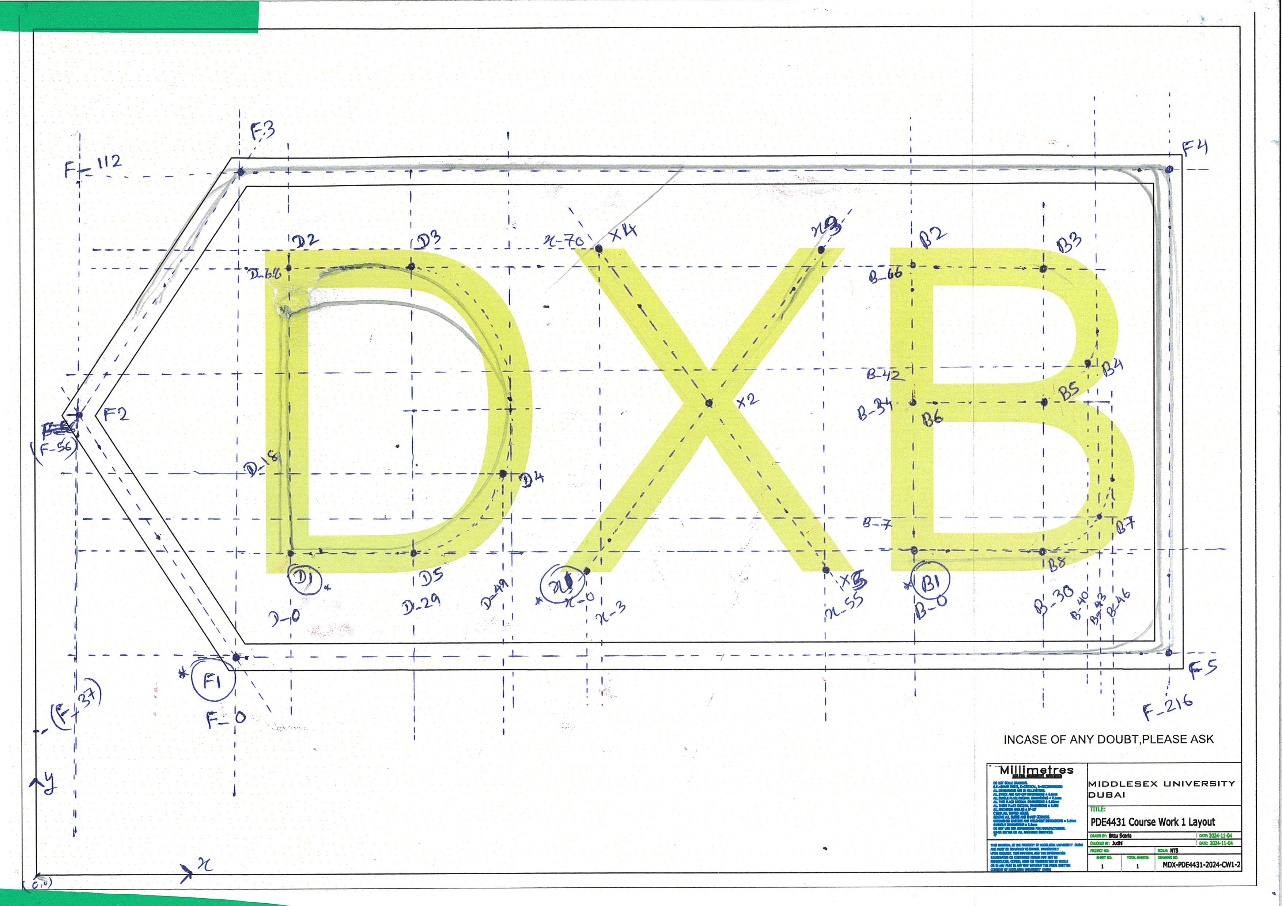


Figure Task Preparation - Base on coordinates Source: (Author Developed)

Required area has been divided into four functions and each function called one after another to complete the task. Each letter assigned as a function and frame of the letters has been a separate function. Every letter has been defined on a base point and rest of the coordinates defines base on one actual recorded point.

## Planning and buildup the flow of task

As per the given task, selected to use local frame and it will help to navigate the robot easily and complete the task with minimum errors. On the local frame, all the base points defined and take the measurements to rest of the coordinate to build the program. Each of the letter, there will be a base point and depend on the shape of the different parts in the letter, required total numbers of points can be different. Below is the example,

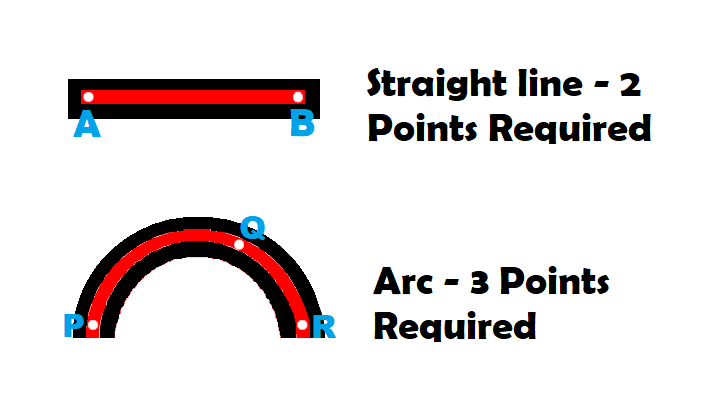


Figure Point define example Source: (Author Developed)

As per the above example, For the straight line, 2 points required and for an arc, 3 points required.

## Define the base coordinates and measure the rest of the coordinates from base coordinates.

As per below image points defined to be measure. Numbers of pints as below.

Table Defined points to the task Source: (Author developed)

|  |  |  |  |
| --- | --- | --- | --- |
| Description | Included shapes | Base Point | Naming pattern |
| Letter **D** | 3 - Straight lines and 1 - Arc | DP\_1 | (DP\_1 – DP\_2) = Straight line  (DP\_2 – DP\_3) = Straight line  (DP\_3 – DP\_4 – DP\_5) = Arc  (DP\_5 – DP\_1) = Straight line |
| Letter **X** | 2 - Straight lines | XP\_1 | (XP\_1 – XP\_3) = Straight line  (XP\_4 – XP\_5) = Straight line |
| Letter **B** | 4 - Straight lines and 2 - Arc | BP\_1 | (BP\_1 – BP\_2) = Straight line  (BP\_2 – BP\_3) = Straight line  (BP\_3 – BP\_4 – BP\_5) = Arc  (BP\_5 – BP\_6) = Straight line  (BP\_5 – BP\_7 – BP\_8) = Arc  (BP\_8 – BP\_1) = Straight line |
| Frame | 2 - Straight lines | FP\_1 | (FP\_1 – FP\_2) = Straight line  (FP\_2 – FP\_3) = Straight line  (FP\_3 – FP\_4) = Straight line  (FP\_4 – FP\_5) = Straight line  (FP\_5 – FP\_1) = Straight line |

As per described above below is the coordinates marking on the paper. Above base coordinates on the paper as below;

* DP\_1 as D\_1 on the paper.
* XP\_1 as X\_1 on the paper.
* BP\_1 as B\_1 on the paper.
* FP\_1 as F\_1 on the paper.

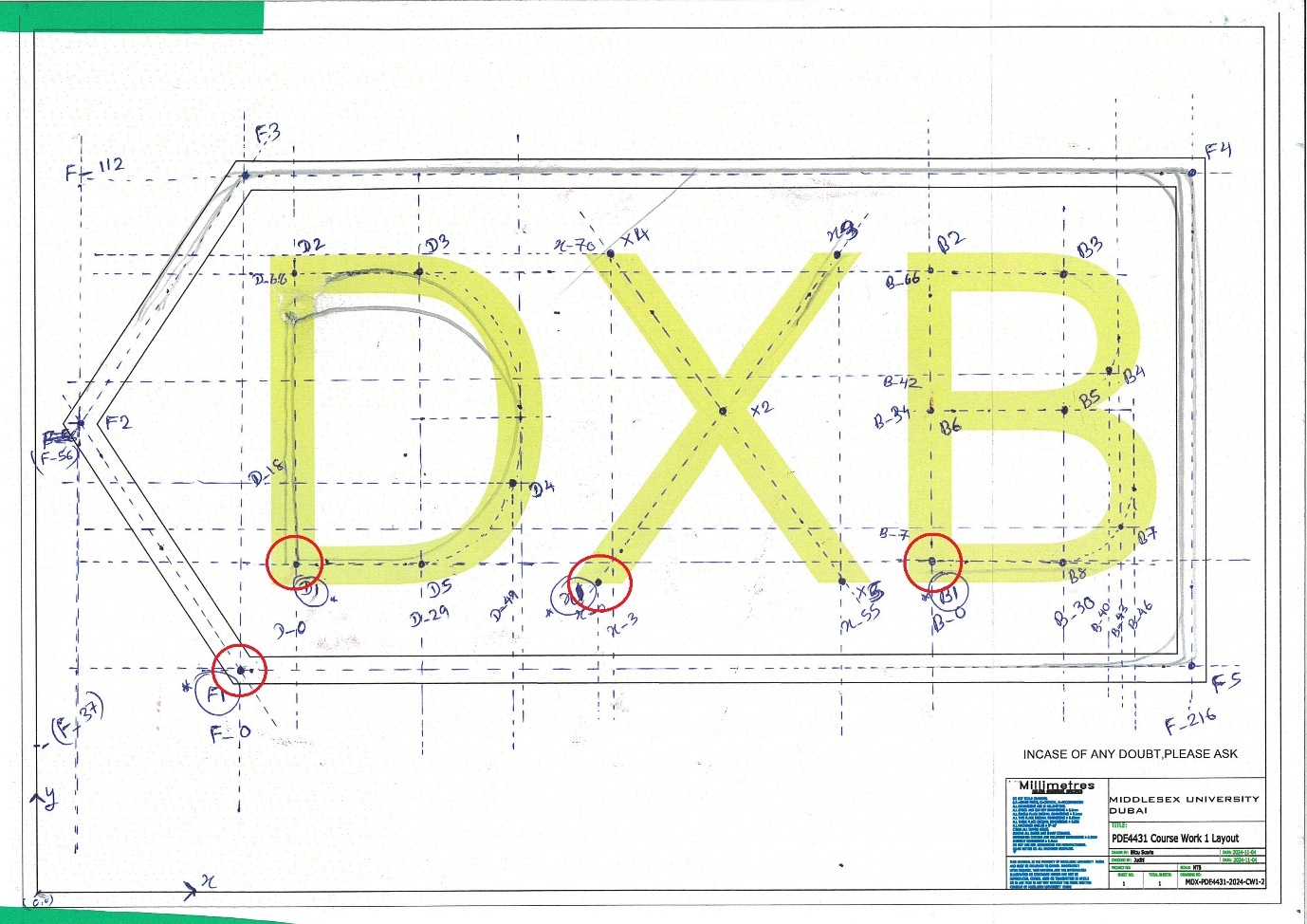


Figure Task preparation sheet with base coordinates Source: (Author developed)

## Build on virtual environment

Once finish the coordination in paper works, need to develop the program in the virtual environment. For that;

1. Open Epson RC software and create a new project
2. Create a connection (Setup > PC to controller). Then click add > Select ‘Connection to new virtual controller’. Finally click the new virtual controller and click connect. (*Note: Virtual controller name changes to “****PDE4431CW1****”)*

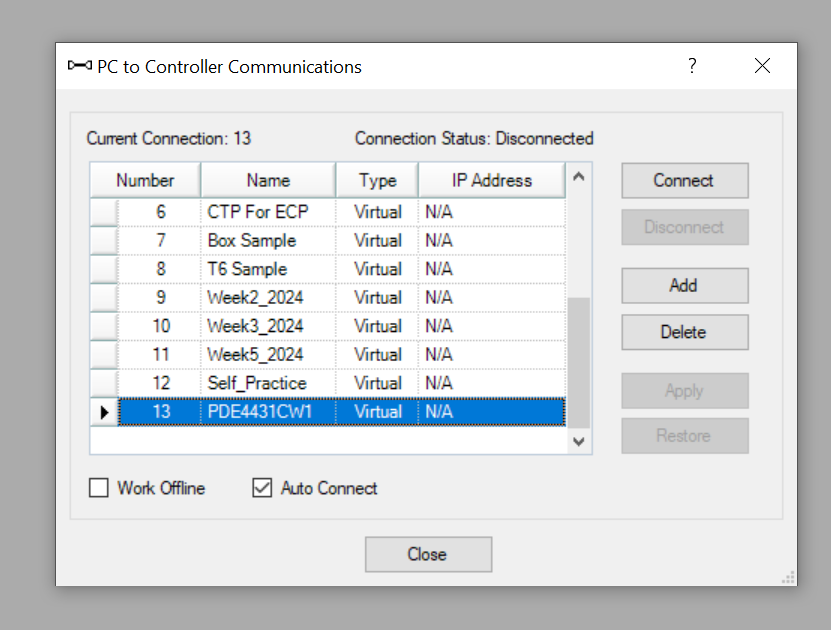


Figure Setup connection Source: (Author developed)

1. Create a robot in the environment. Go to Setup > System configuration > Under controller menu select robot and added a new 6-axis robot. (*Note: Selected same kind of robot of VT6-A901S)*

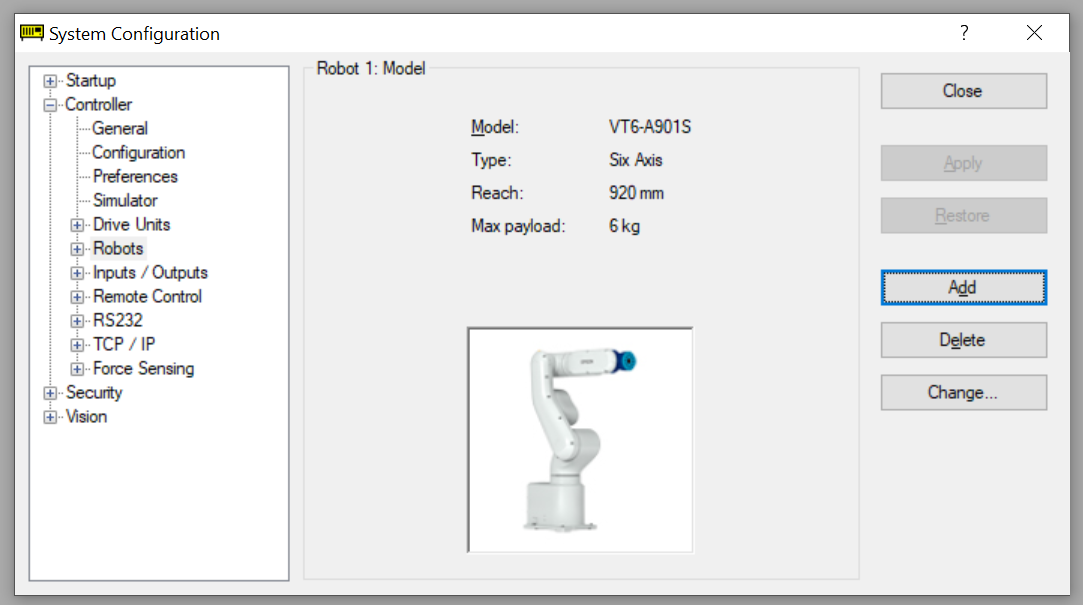


Figure Added a robot to the environment Source: (Author developed)

1. Open robot simulator.

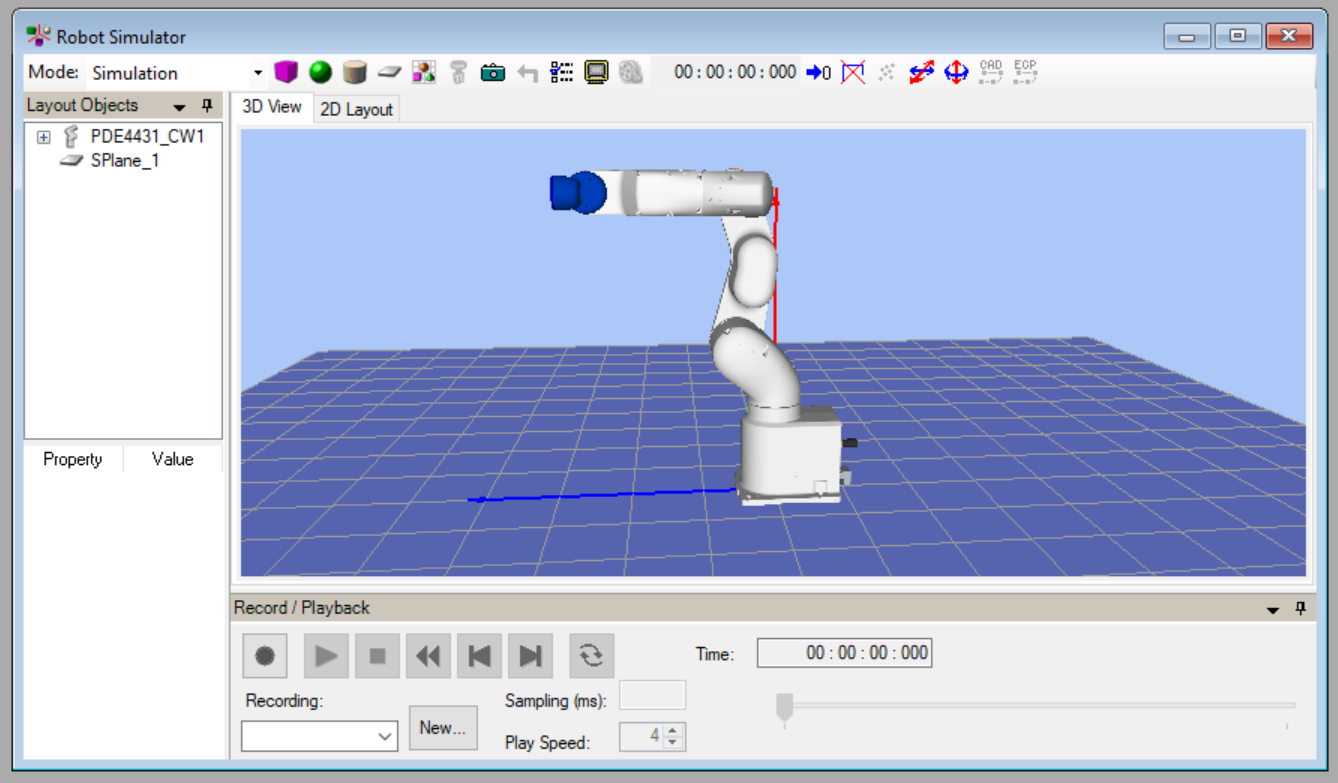


Figure Robot simulator Source: (Author developed)

1. Start the robot manager for enter the points and test the program. Go to Tools > Robot Manager.

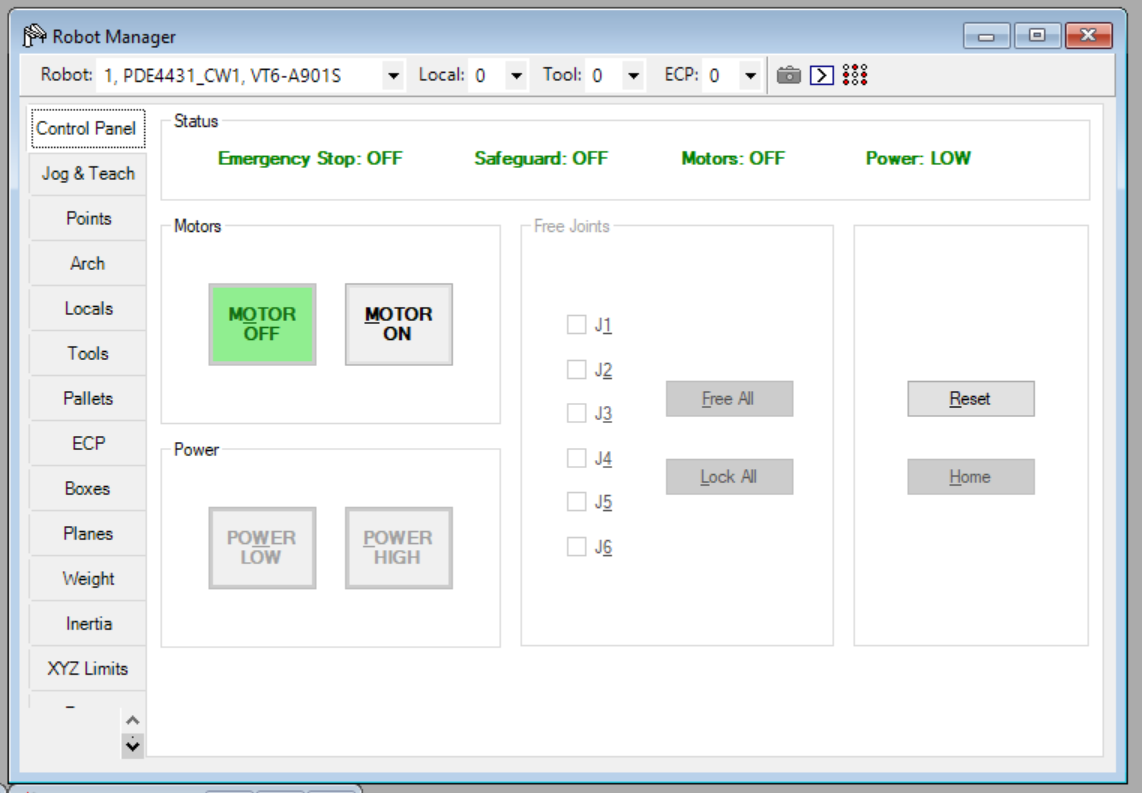


Figure Robot manager window Source: (Author developed)

1. Click the “Motor On” > “Power high” to start the robot.
2. Go to “Points” and enter points to the program. Then Click save.

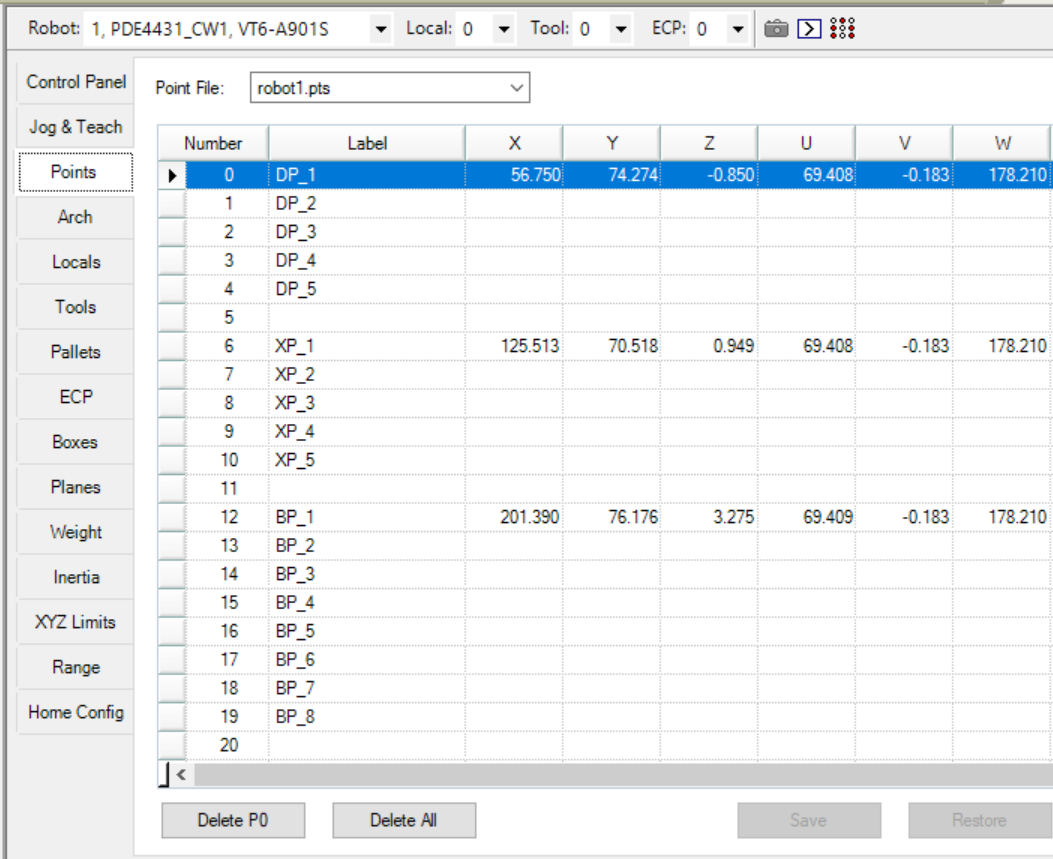


Figure Points define Source: (Author developed)

1. Then go to “jog and Teach”. Using “Teaching Points” in bottom tab, teach the base point as mentioned above table. (*Note: Simulation will be run on the world frame and real-world application will be test and run on local frame*)

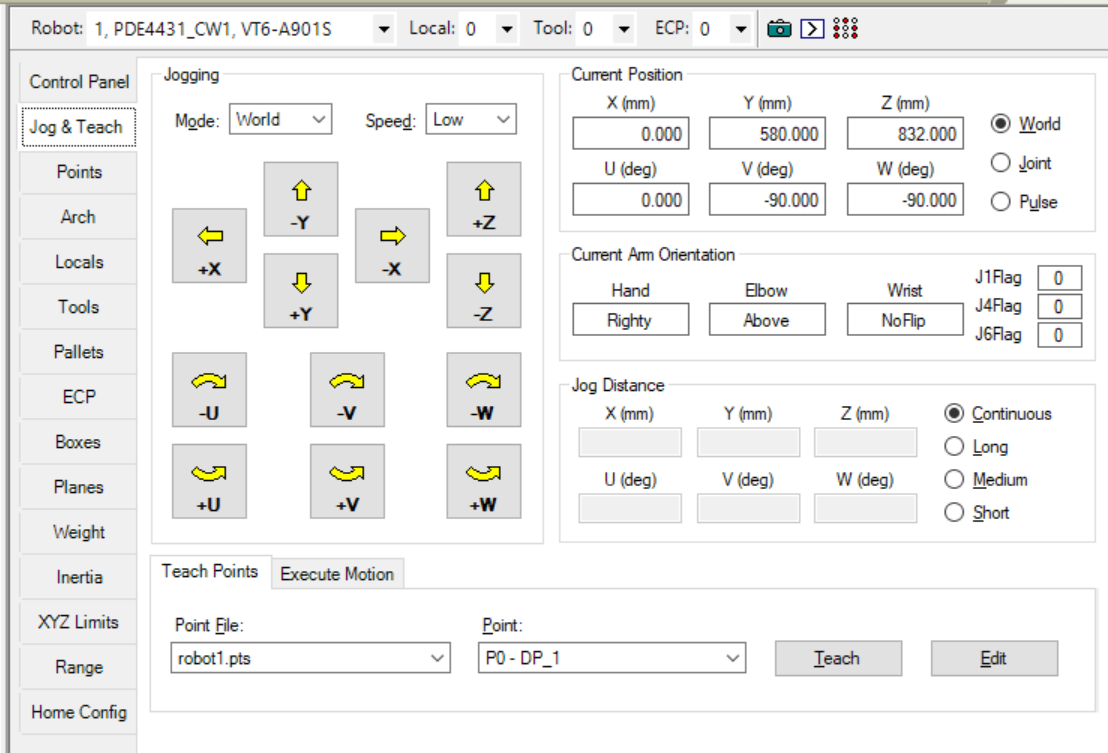


Figure Jog and Teach the points Source: (Author developed)

1. Do the programming.

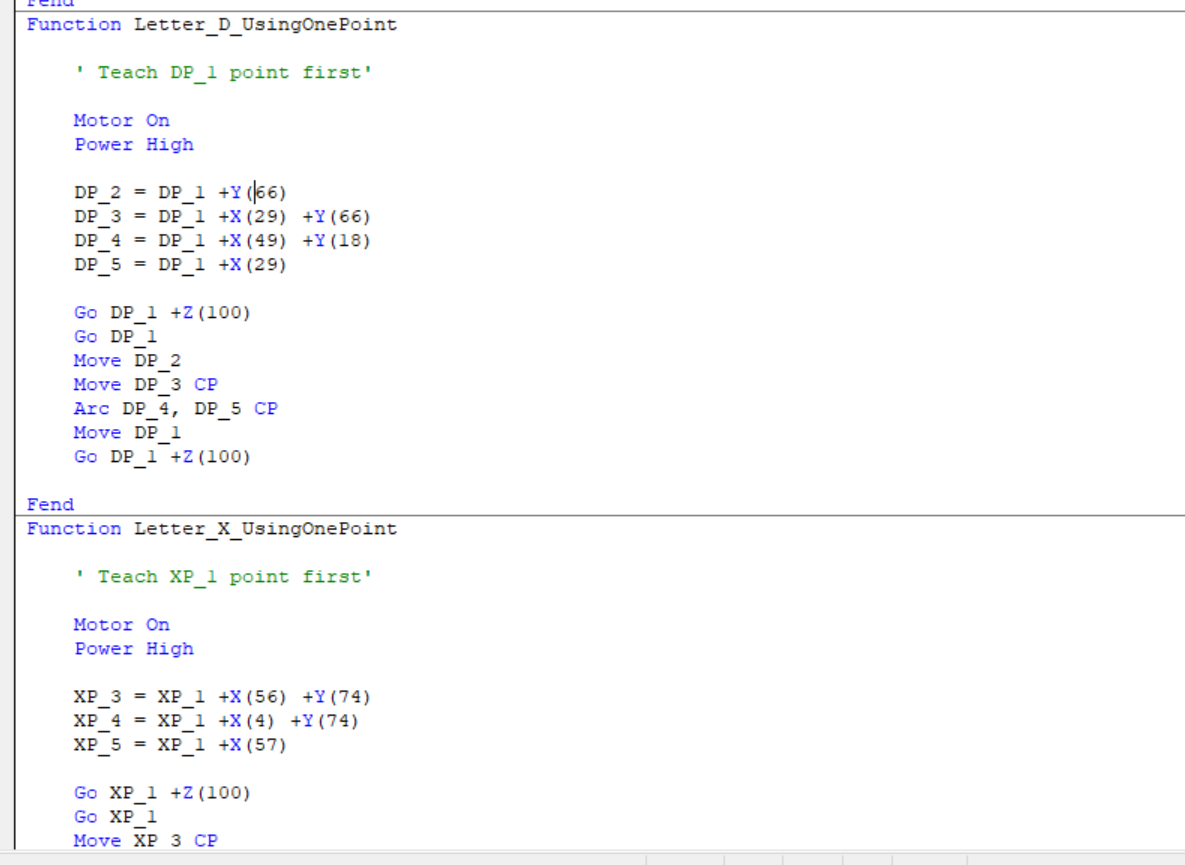


Figure Programming Source: (Author developed)

1. Call by function names and run the program.

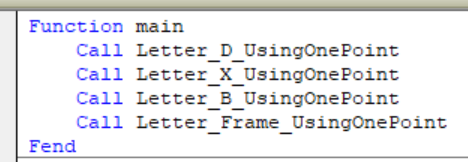


Figure Custom defined functions call in main function Source: (Author developed)

## Connect the Robot

Once successfully run the program in simulator, need to test the program before final demonstration. For that, Go to Setup > PC to controller and then select USB, then click connect.

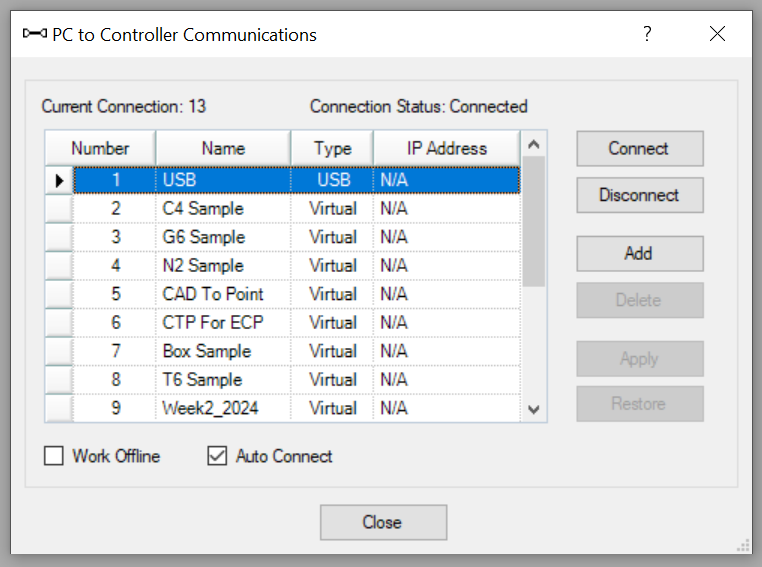


Figure Connect the robot Source: (Author developed)

## Setting up the tools

For set the tools,

1. Install the tool to J6 joint.
2. Go to Tools > Robot Manager > Tools. Click Tool wizard and follow the instruction to setup the tool.

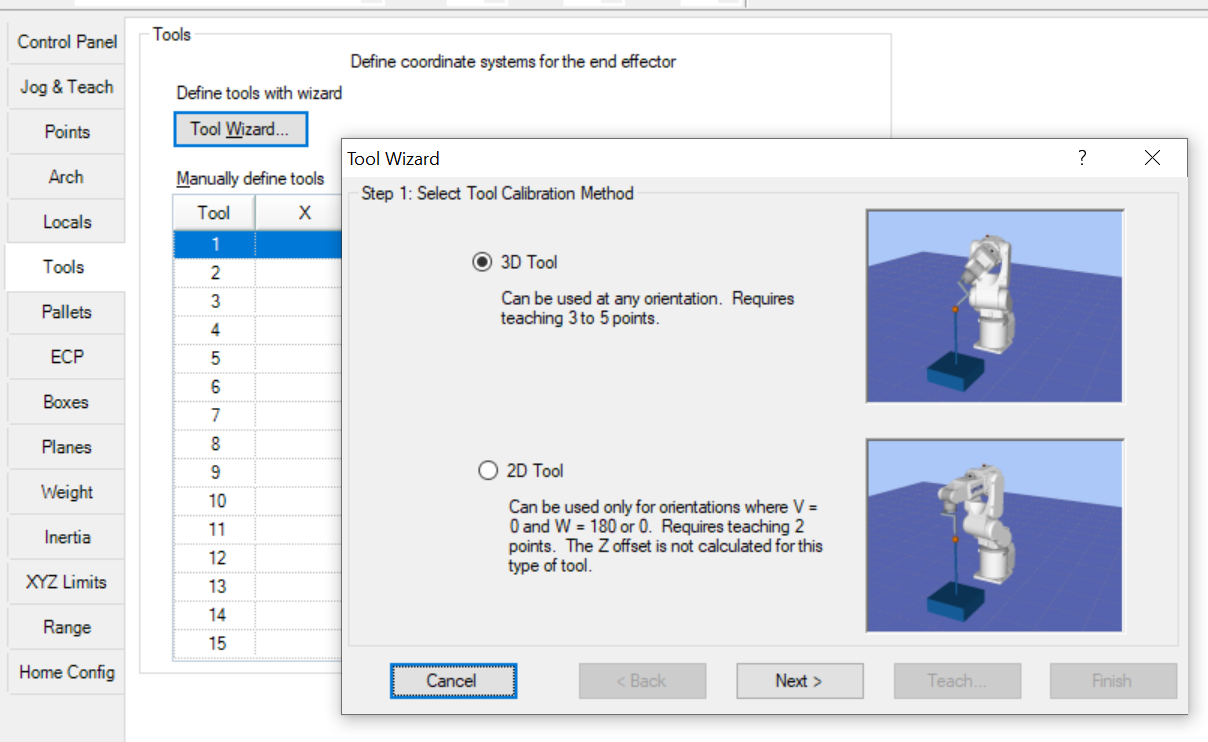


Figure Setup Tool Source: (Author developed)

1. For more accuracy, 3D tools selected and selected 4 points to teach for the tool.

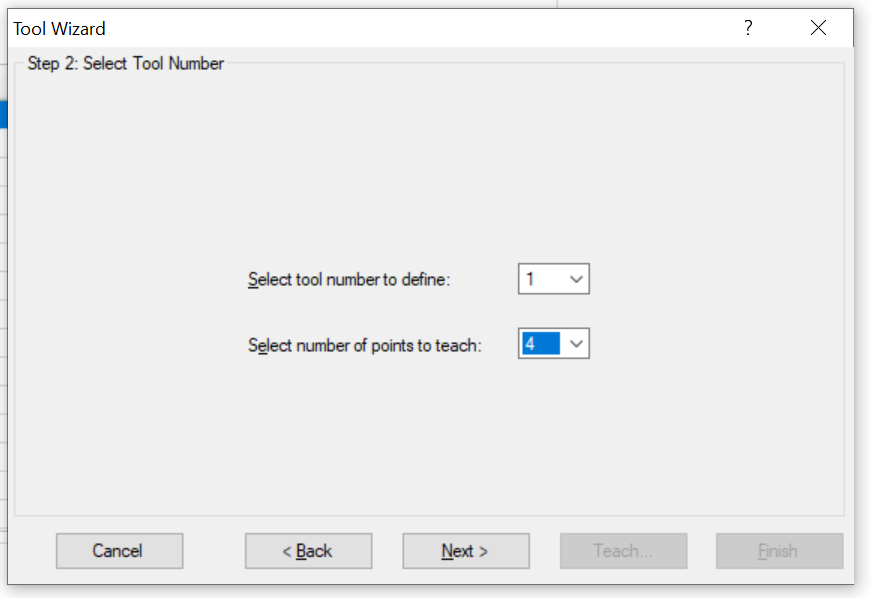


Figure Teach the points to setup the tool Source: (Author developed)

1. Followed the instruction to teach the points and save the tool. Tool will be Tool-1.

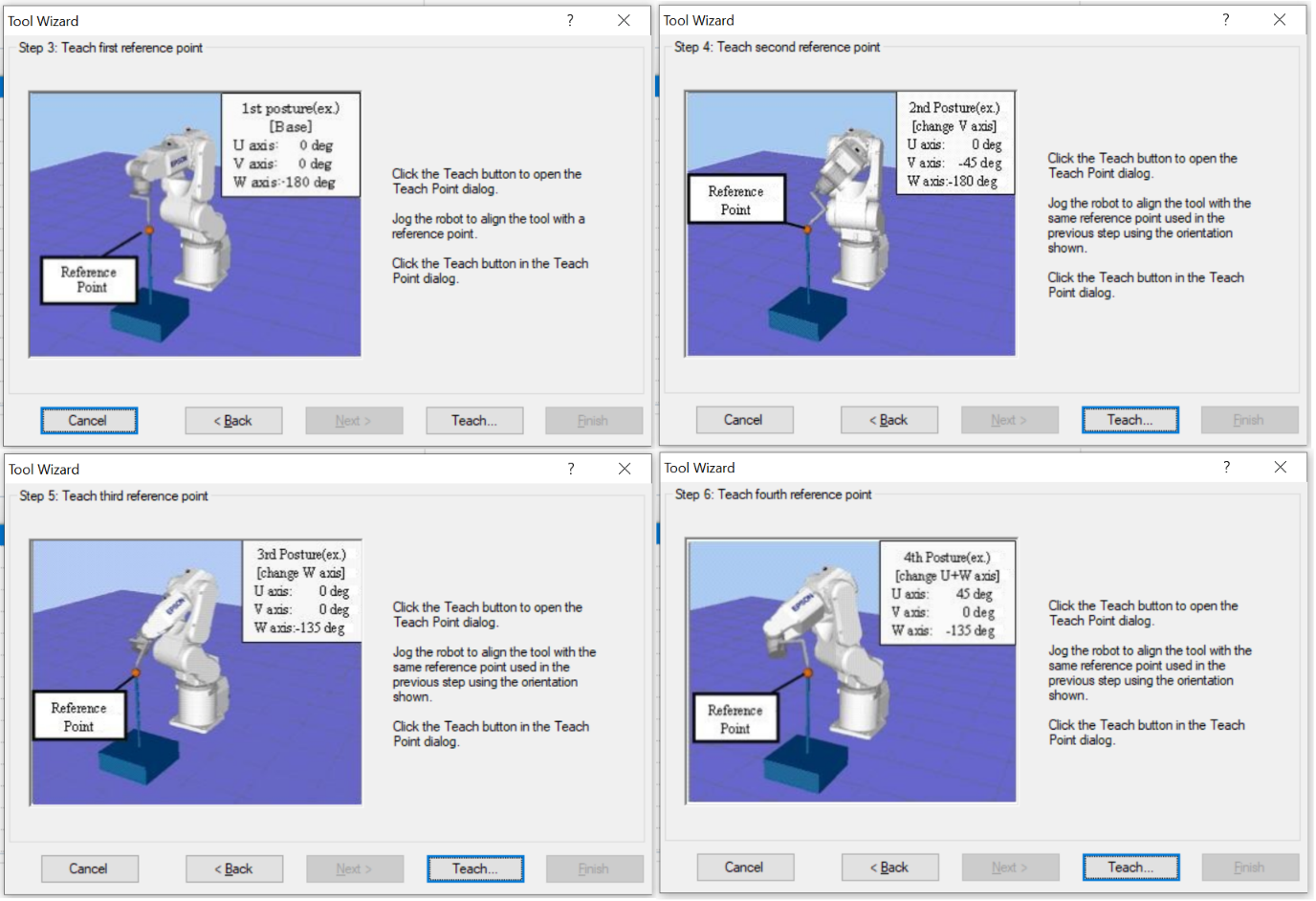


Figure Teach origin, X and Y for local frame Source: (Author developed)

## Setting up the frame

1. Go to Tools > Robot Manager > Locals. Click Local wizard and follow the instruction to setup the tool. Used 3 origin to set the local frame.

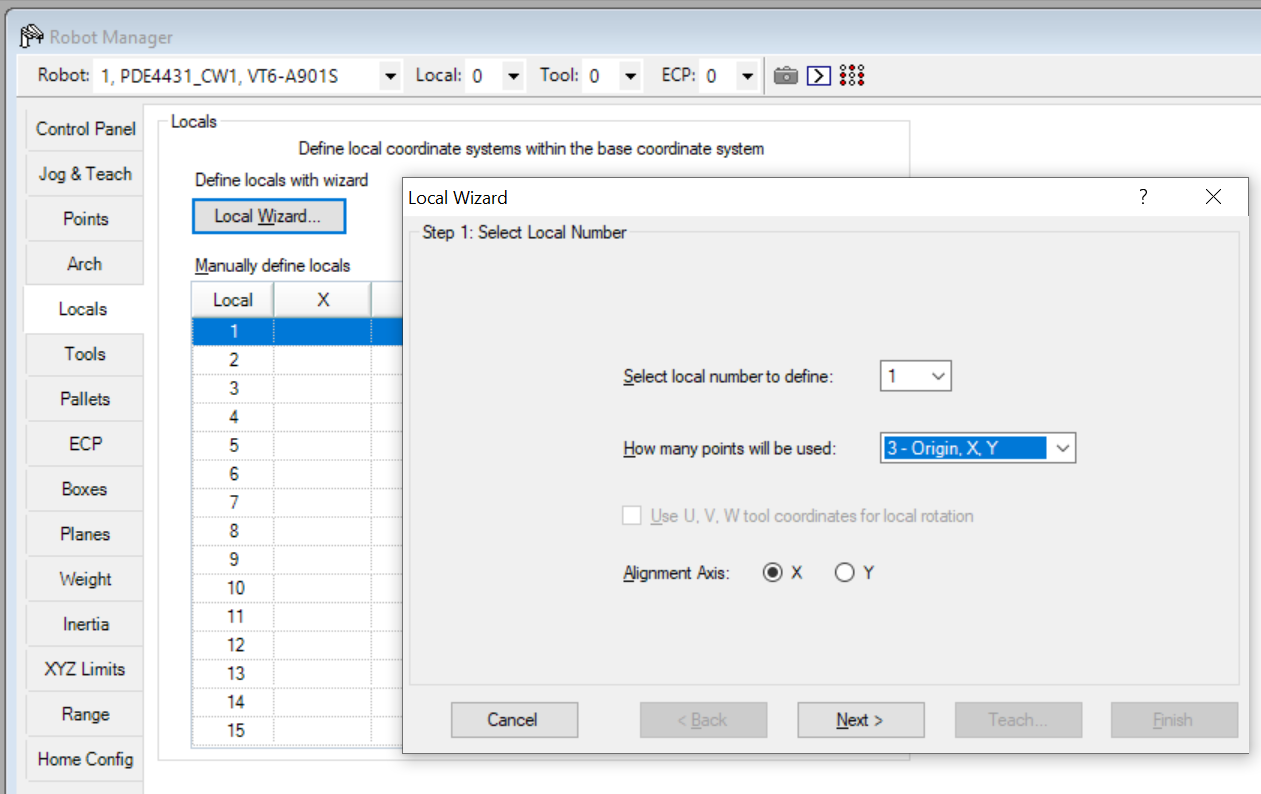


Figure Setup Local frame Source: (Author developed)

1. Then follow the instructions and record the frame using base, X and Y axis.

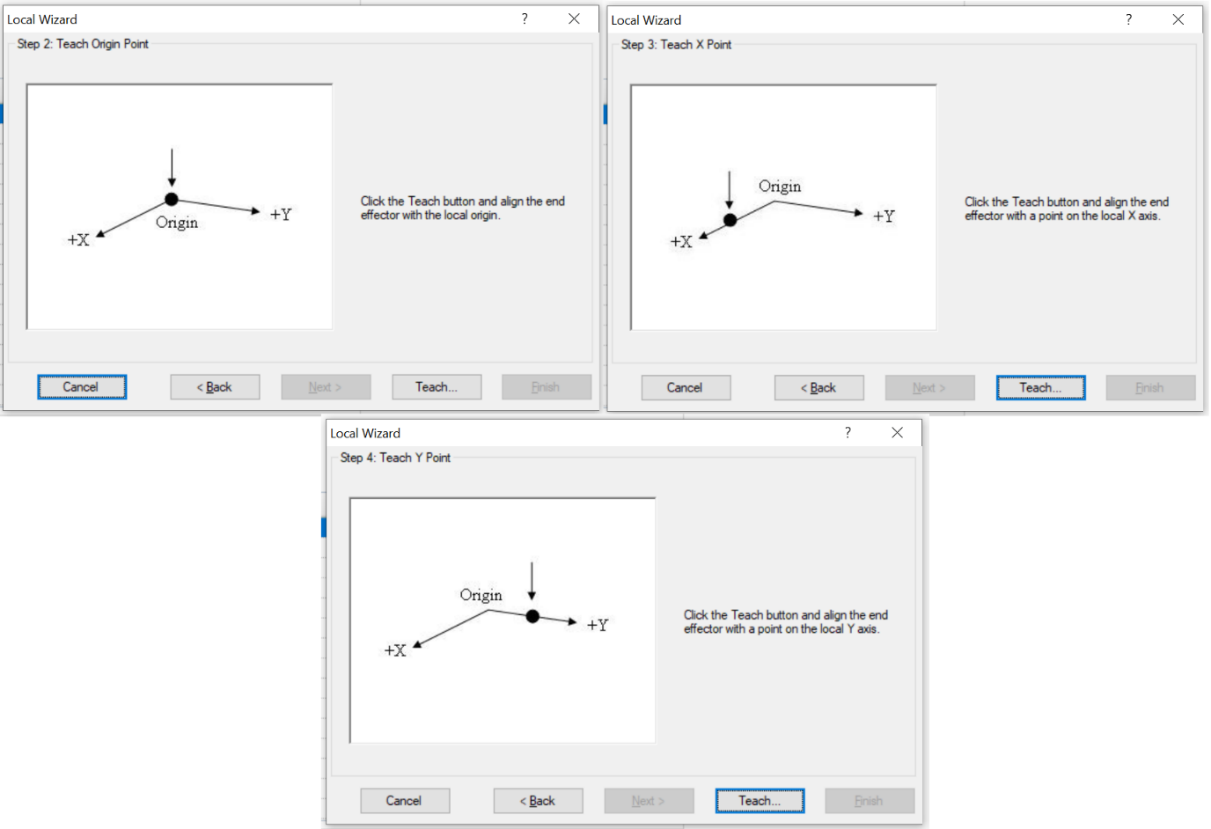


Figure Teach Origin, X and Y for local frame Source: (Author Developed)

1. At last need to save and record the base points.

## Initialize / Record the base coordinates

Once the tool and frame setup complete need to record the base points. First select the local frame (Local – 1) and Tool-1. And open Jog and teach mode. Then using Teach points tab, taught below points.

* **DP\_1 = Base point of Letter D**
* **XP\_1 = Base point of Letter X**
* **BP\_1 = Base point of Letter B**
* **FP\_1 = Base point of Frame**

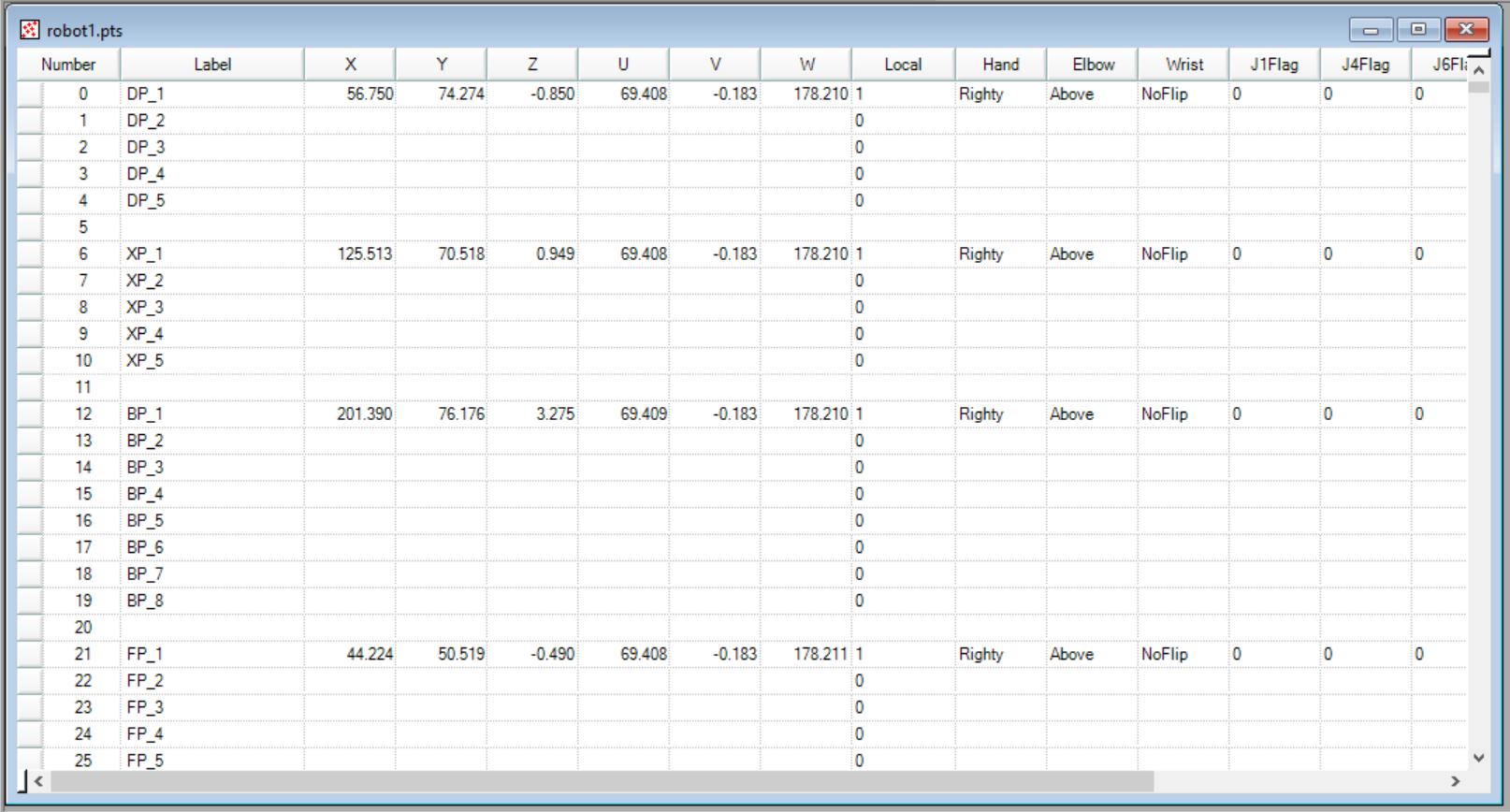


Figure Teach the base points of each letter and the frame Source: (Author developed)

## Written functions

### Function for Letter “D”

|  |
| --- |
| Function Letter\_D\_UsingOnePoint    ***' Teach DP\_1 point first'***    Motor On  Power High    DP\_2 = DP\_1 +Y(66)  DP\_3 = DP\_1 +X(29) +Y(66)  DP\_4 = DP\_1 +X(49) +Y(18)  DP\_5 = DP\_1 +X(29)    Go DP\_1 +Z(100)  Go DP\_1  Move DP\_2  Move DP\_3 CP  Arc DP\_4, DP\_5 CP  Move DP\_1  Go DP\_1 +Z(100)    Fend |

### Function for Letter “X”

|  |
| --- |
| Function Letter\_X\_UsingOnePoint    ***' Teach XP\_1 point first'***    Motor On  Power High    XP\_3 = XP\_1 +X(56) +Y(74)  XP\_4 = XP\_1 +X(4) +Y(74)  XP\_5 = XP\_1 +X(57)    Go XP\_1 +Z(100)  Go XP\_1  Move XP\_3 CP  Jump3 XP\_3, XP\_3 -X(26) +Z(50), XP\_4 CP  Move XP\_5  Go XP\_5 +Z(100)    Fend |

### Function for Letter “B”

|  |
| --- |
| Function Letter\_B\_UsingOnePoint    ***' Teach BP\_1 point first'***    Motor On  Power High    BP\_2 = BP\_1 +Y(66)  BP\_3 = BP\_1 +X(30) +Y(66)  BP\_4 = BP\_1 +X(40) +Y(42)  BP\_5 = BP\_1 +X(30) +Y(34)  BP\_6 = BP\_1 +Y(34)  BP\_7 = BP\_1 +X(43) +Y(7)  BP\_8 = BP\_1 +X(30)    Go BP\_1 +Z(100)  Go BP\_1  Move BP\_2 CP  Move BP\_3 CP  Arc BP\_4, BP\_5 CP  Move BP\_6  Jump3 BP\_6, BP\_6 +X(15) +Z(50), BP\_5 CP  Arc BP\_7, BP\_8 CP  Move BP\_1  Go BP\_1 +Z(100)    Fend |

### Function for the frame

|  |
| --- |
| Function Letter\_Frame\_UsingOnePoint    ***' Teach FP\_1 point first'***    Motor On  Power High    FP\_2 = FP\_1 -X(35) +Y(55)  FP\_3 = FP\_1 +Y(112) +X(4)  FP\_4 = FP\_1 +X(218.5) +Y(112)  FP\_5 = FP\_1 +X(218.5)    Go FP\_1 +Z(100)  Go FP\_1  Move FP\_2  Move FP\_3  Move FP\_4  Move FP\_5  Move FP\_1  Go FP\_1 +Z(100)    Fend |

## Used functions to develop the program

1. **Go –** To reach the point which is even not is straight line.
2. **Move –** To reach a point in linier angle.
3. **Arc** – This function required 2 points to complete the task and can be added CP function with this.
4. **Jump3** – This function will use to jump from one point to defined point.
5. **CP** **(Continues Path)** – This function will guide the robot to move with any deacceleration.

# Task preparation on the sheet in final

As described above, below is the final preparation of the task to be program in VT-6 robot.

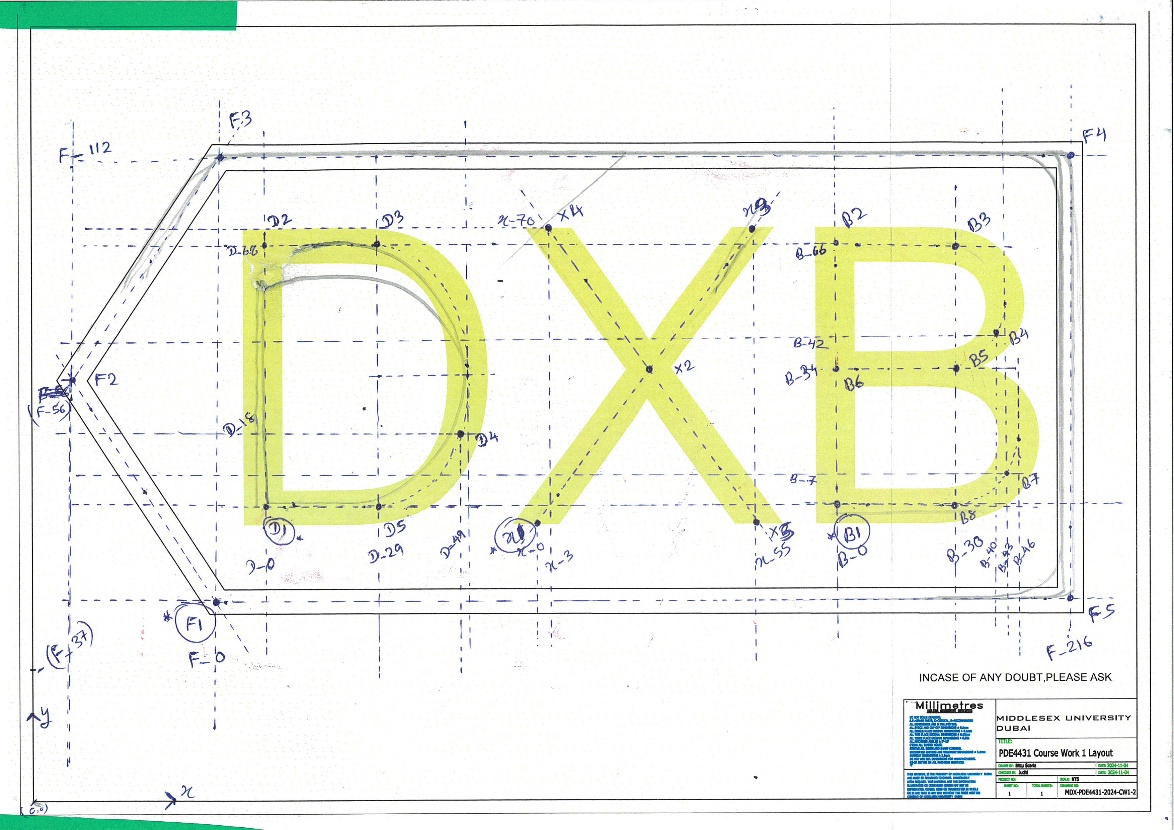


Figure Task preparation source: (Author developed)

# Task Demonstration

Task demonstration completed and below link can be connected to the demonstration video.

# Conclusion

- What will be the file size of the report?

-- 3-5 pages of your write up with supporting pages (trace sheet and preparation sheet)

- What will be the marking criteria of the report or is it just an explanation about the coursework?

-- It is the reflection of your task, so explain the task, outcome and improvement shortfalls in the report

- Need to explain in deep how the step carried out (Example: Step 1 - Define the base coordinate of the letter to calculate other points <<Explain how other coordinates aligned with base coordinates>>)?

-- yes

- Finally - My video is clearly showing that the tool is not touching the paper, is that will be an effect to the marks?

-- NOT to worry.