ROBOT MANIPULATION CW1-EPSON RC 8.0

Robot Programming Report CW1
Student Name: Arman Shaikh
Student ID: M01096196
Course: PDE4431 – Fall 2025
Institution: Middlesex University Dubai

1. Introduction & Task Overview:

- Robot programming is a cornerstone of modern industrial automation. It enables robots to perform repetitive, precise, and high-speed tasks with minimal human intervention. In manufacturing environments, robot programming is used for:

- ✓ Assembly operations, material handling, welding and painting, inspection and quality control, Packaging and palletizing.

- By mastering robot programming, engineers can design work-flows that are scalable, safe, and adaptable to changing production needs. This coursework simulates such industrial applications by requiring students to configure tools, teach points, and execute motion sequences that reflect real-world robotic tasks.

- This project involved programming an industrial robot to trace a defined path using motion commands such as GO, MOVE, ARC, and JUMP.The robot was equipped with a pointer tool mounted on its end flange, and the task required precise path execution within a specified tolerance. The robot was programmed to start and end at a Home/Reference position, incorporating approach and retract points. Tool offsets and coordinate shifts were applied to ensure accurate Z-height alignment. Coordinate shifts were also applied to move on linear path in some parts of the project.

2.Robot Setup and Tool Configuration

2.1 Robot and Software Used

- Robot Model: EPSON ARTICULATED VT6 910S
- Connection Type: USB
- Programming Software: EPSON RC+ 8.0

ROBOT MANIPULATION CW1-EPSON RC 8.0

2.2 Tool Description and Mounting Procedure

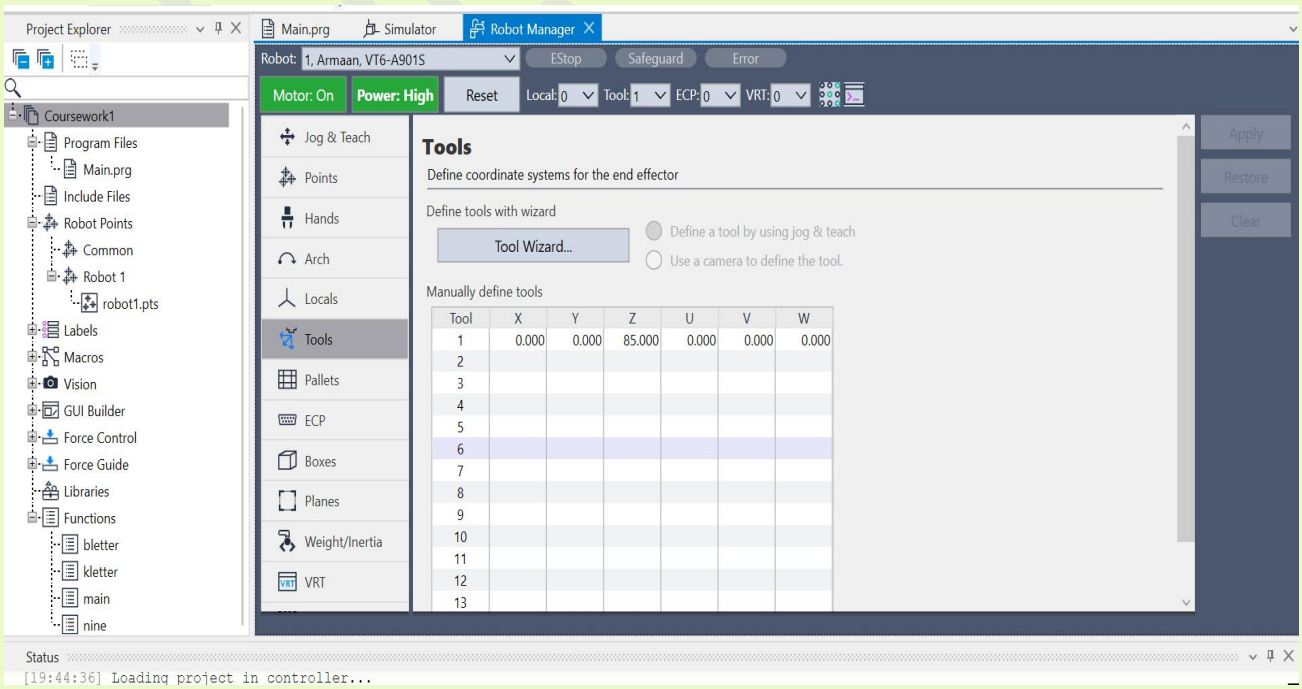
- A lightweight PENCIL pointer tool was mounted on the robot's EOAT using a holder. The bottom of the pencil was mounted on a small 3D printed holder, which was then attached to the main fixture. The holder was designed to align the pointer tip precisely with the robot's Z-axis. The pointer tip was designated as the Tool Center Point (TCP), and its position was calibrated manually.

2.3 Tool Offset Verification – Direct Entry Method

- To configure the tool offset, the Direct Entry Method was used within the EPSON RC+ software. This method involves manually inputting the tool's spatial parameters into the robot's tool configuration table.

❖ Steps Followed:

1. Measurement using a measuring tape to determine Z-offset (+85 mm).
2. Manual entry of X, Y, Z values into EPSON RC+8.0 tool table; W, P, R set to 0.0°.
3. Apply the value entered and select the tool from TOOL DROP DOWN LIST on the same page which will activate the tool for program reference.
4. Below is the image for the TCP setting window and TCP value for the CW1.



ROBOT MANIPULATION CW1-EPSON RC 8.0

2.4 Teaching and Saving Robot Points

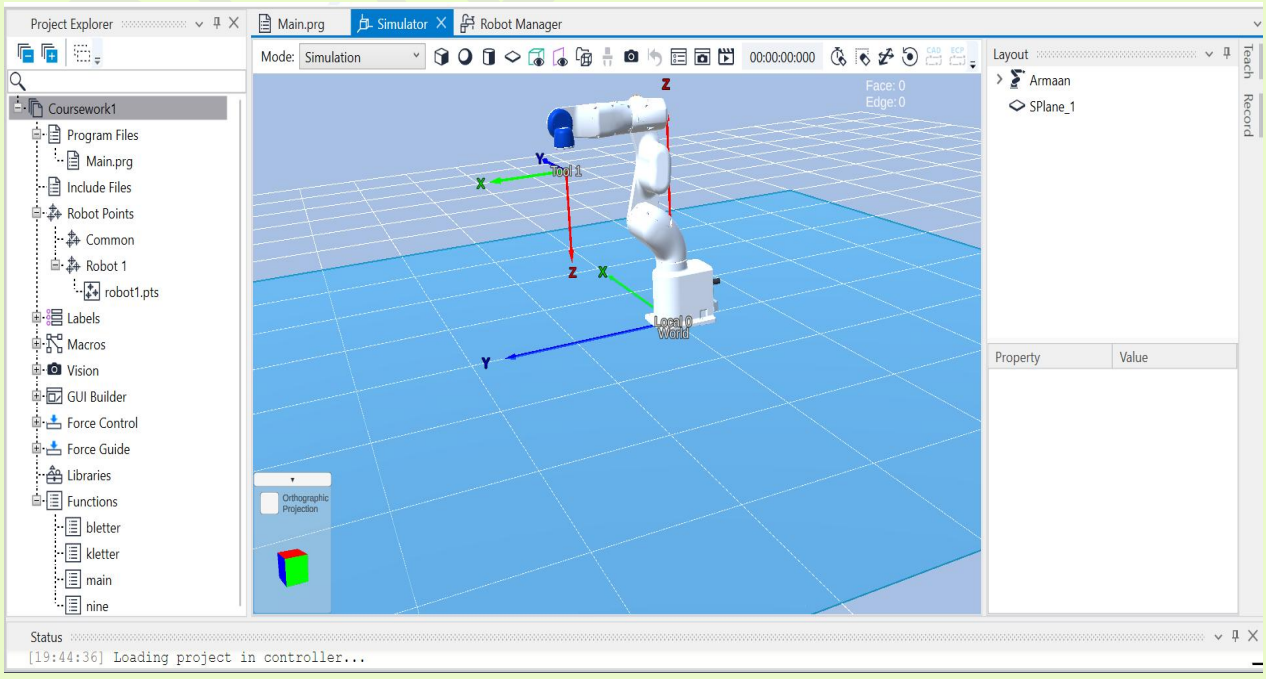
➢ To execute precise movements, the robot must be taught specific positions called POINTS which define the coordinates and orientation of the end effector.

❖ **Accessing the Points Menu:**

- Open EPSON RC+ and make connection via USB.
- Navigate to Robot Manager > Points tab.

❖ **Setting up the J5 axis:** The First step after setting up TCP is to set the J5 axis before teaching points.

- Setting the J5 axis (The wrist rotation) of a robot to -90 degrees is often done to achieve a specific tool orientation relative to the work surface.
- It is important to set J5 axis to -90 degrees to avoid COLLISION AVOIDANCE and maintain TOOL PERPENDICULARITY, CONSISTENT CONTACT ANGLE.
- In the Jog & Teach menu select the joint mode and rotate J5 to -90 degrees.
- Below is the image from simulation window where you can observe the TCP and the J5 at -90 degree.



ROBOT MANIPULATION CW1-EPSON RC 8.0

❖ **Teaching Points:** For recording points follow the steps below:

- Jog the robot to the desired location.
- Here in this project we have used World frame as reference for jogging and kept power and speed HIGH.
- Used Continuous and medium position changing options to achieve precise and accurate position.
- Click “Teach Point” to record the position.
- Assign labels and confirm configuration.

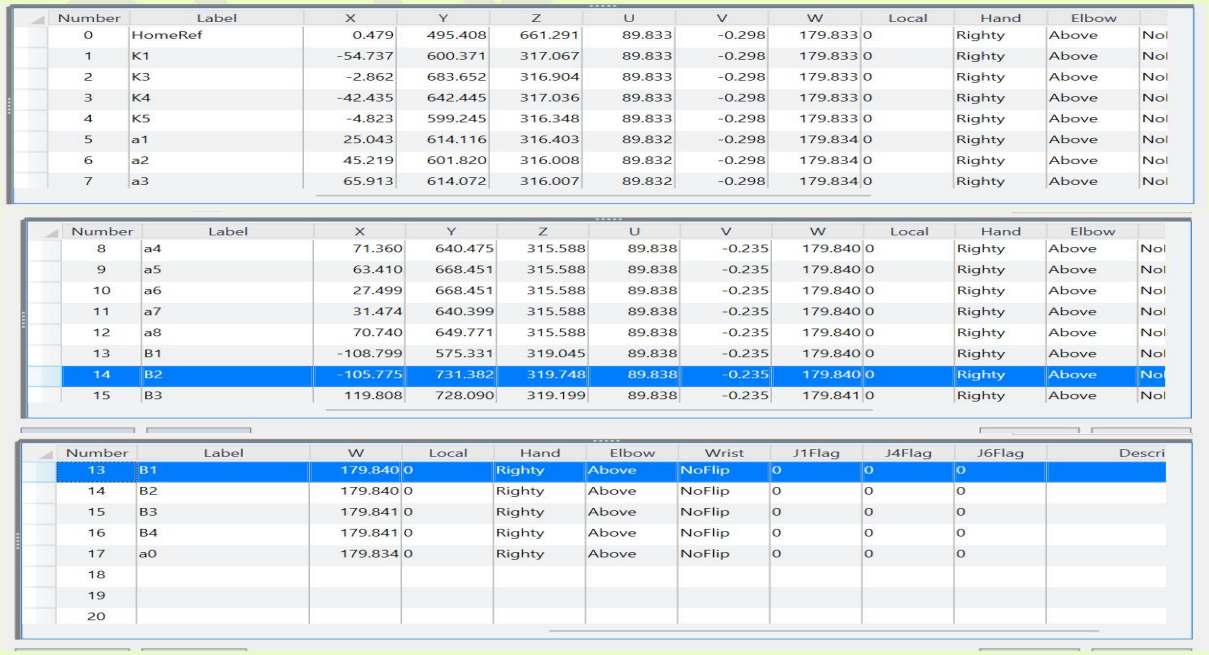
❖ **Saving Points:**

- Points are auto-saved in the project.
- Click the save button on the Jog & Teach menu or go to points menu and do the same.
- Use File > Save Project to ensure persistence.

❖ **Best Practices:**

- Use meaningful labels.
- Teach approach/retract points above the surface.
- Verify each point before execution.

✧ After teaching and saving all points below is the image on how the GUI will appear and how the points look after naming and giving them positions.



Number	Label	X	Y	Z	U	V	W	Local	Hand	Elbow	
0	HomeRef	0.479	495.408	661.291	89.833	-0.298	179.833	0	Righty	Above	Nol
1	K1	-54.737	600.371	317.067	89.833	-0.298	179.833	0	Righty	Above	Nol
2	K3	-2.862	683.652	316.904	89.833	-0.298	179.833	0	Righty	Above	Nol
3	K4	-42.435	642.445	317.036	89.833	-0.298	179.833	0	Righty	Above	Nol
4	K5	-4.823	599.245	316.348	89.833	-0.298	179.833	0	Righty	Above	Nol
5	a1	25.043	614.116	316.403	89.832	-0.298	179.834	0	Righty	Above	Nol
6	a2	45.219	601.820	316.008	89.832	-0.298	179.834	0	Righty	Above	Nol
7	a3	65.913	614.072	316.007	89.832	-0.298	179.834	0	Righty	Above	Nol

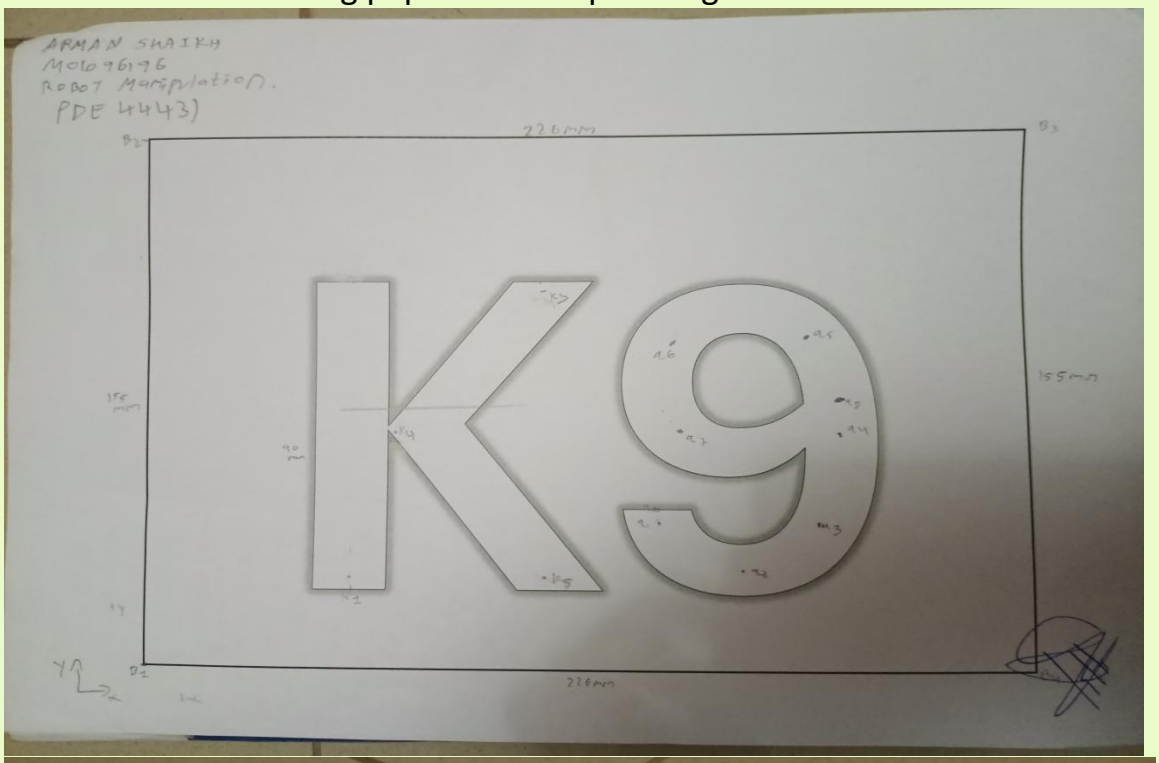
Number	Label	X	Y	Z	U	V	W	Local	Hand	Elbow	
8	a4	71.360	640.475	315.588	89.838	-0.235	179.840	0	Righty	Above	Nol
9	a5	63.410	668.451	315.588	89.838	-0.235	179.840	0	Righty	Above	Nol
10	a6	27.499	668.451	315.588	89.838	-0.235	179.840	0	Righty	Above	Nol
11	a7	31.474	640.399	315.588	89.838	-0.235	179.840	0	Righty	Above	Nol
12	a8	70.740	649.771	315.588	89.838	-0.235	179.840	0	Righty	Above	Nol
13	B1	-108.799	575.331	319.045	89.838	-0.235	179.840	0	Righty	Above	Nol
14	B2	-105.775	731.382	319.748	89.838	-0.235	179.840	0	Righty	Above	No
15	B3	119.808	728.090	319.199	89.838	-0.235	179.841	0	Righty	Above	Nol

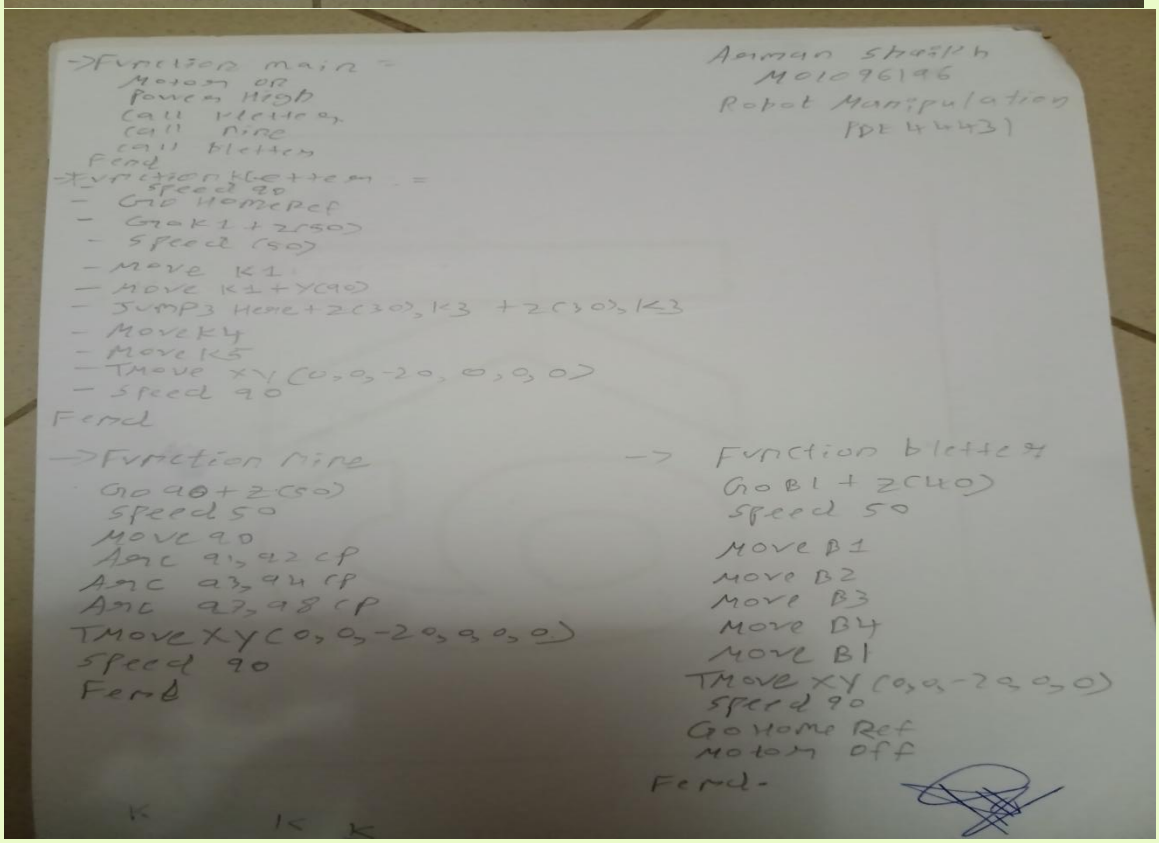
Number	Label	W	Local	Hand	Elbow	Wrist	J1Flag	J4Flag	J6Flag	Descri
13	B1	179.840	0	Righty	Above	NoFlip	0	0	0	
14	B2	179.840	0	Righty	Above	NoFlip	0	0	0	
15	B3	179.841	0	Righty	Above	NoFlip	0	0	0	
16	B4	179.841	0	Righty	Above	NoFlip	0	0	0	
17	a0	179.834	0	Righty	Above	NoFlip	0	0	0	
18										
19										
20										

ROBOT MANIPULATION CW1-EPSON RC 8.0

3. Programming Strategy:

- The task was to trace one alphabet and one letter and the outer box “K” & one number “9”. So at first we have to make en execution plan on a paper sheet and then work on the real robot.
- Below is the tracing paper and the planning sheet.





ROBOT MANIPULATION CW1-EPSON RC 8.0

3.1 Motion Commands Used: The planning and execution of tracing the path has been done by using the following commands:

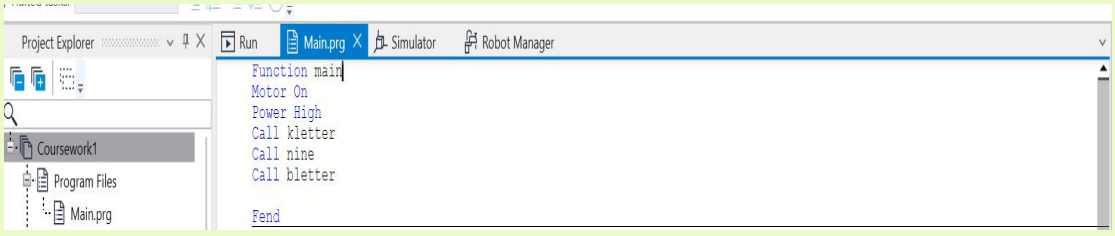
- Important commands such as MOTOR ON,OFF ,POWER HIGH & SPEED have been used to configure the setup.
- **GO**: Rapid, non-interpolated movement to predefined positions.
- **MOVE**: Linear interpolation for precise tracing.
- **ARC**: Circular interpolation using three points.
- **JUMP3**: Lifted movement between non-continuous segments.
- **TMove:** It performs a linear interpolation similar to MOVE, but with added control over the tool's position and W.R.T TCP.

3.2 Path Planning: The path planning includes important steps which are as follows:

- Set a Home Reference point.
- Set an approach and retract points for both the letters and the box.
- Give motion commands to trace both letters and the box.
- Points have been taught via jog-and-teach method & coordinate shift.
- Coordinate shift method has been used to fast the process and demonstrate the use of command.
- Tool offset method which is Tmove is used to ensure correct Z-height contact.(Retract position).

➢ Path planning has been divided into 3 functions as seen in the planning sheet.
➢ The whole program has been divided into 4 Functions:

✓ **Function main:**Initializes robot and calls subroutines. This function is used to call the rest of the functions and to give the instruction commands such as motor on/off, power and speed.
✓ Below is the image:

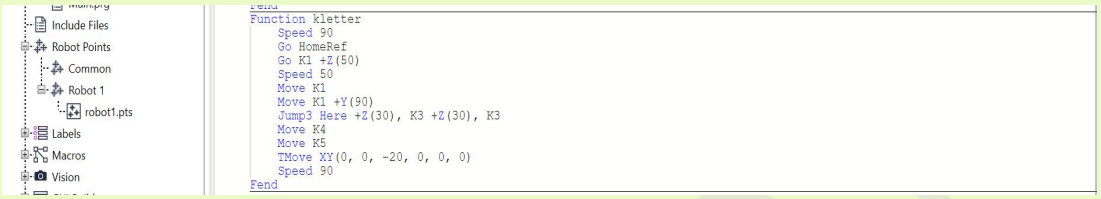


ROBOT MANIPULATION CW1-EPSON RC 8.0

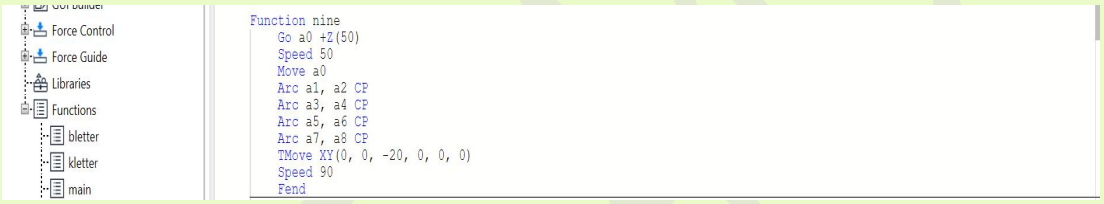
- Main function calls the rest 3 functions required for path tracing.

1. Function kletter= “k”, Function nine=”9”, & Function bletter=”Box”

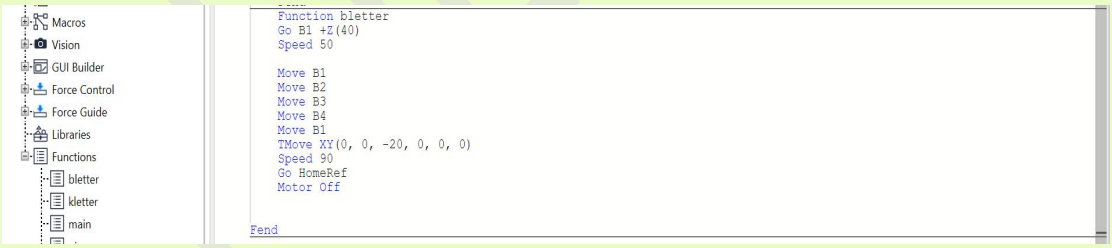
✓ **Function k letter:** Traces “k”. This function when called will demonstrate the motion commands and methods responsible for tracing “k”.



✓ **Function nine:** Traces “9”. This function when called will demonstrate the motion commands and methods responsible for tracing “9”.



✓ **Function bletter:** Traces “Box”. This function when called will demonstrate the motion commands and methods responsible for tracing “Box”.



1. Notes:

i. For Home reference Approach position in every function(letter) **GO** command has been used.
ii. For Retract position in every function(letter) **TMove** command has been used.
iii. To avoid overwriting **JUMP3** command has been used in necessary function.
iv. **CP** command is used as continuous path to provide smooth motion.

ROBOT MANIPULATION CW1-EPSON RC 8.0

4. Video Demonstration:

YouTube Link: Both videos can be accessed by following link.

https://youtube.com/@armaanshaikh6353?si=LSZ26K_MrYe13yNG

■ SUMMARY:

- Use the selected robot programming software.
- Connect the system with the physical robot.
- Set the tool frame accurately.
- Teach points using the most appropriate method.
- Apply various motion commands, speed commands, and other relevant programming instructions.
- Implement offsetting and coordinate shift methods in the program.
- Formulate and execute functions effectively.