Robot Studio, using a pen tool

"An introduction to designing, building, and testing robotic applications"

Student: **R. Eidelman**
eidelmanrv@gmail.com

Date: 21.11.2024

1 Introduction

The aim of this assignment is to familiarize students with RobotStudio and the general workflow to create a workstation, define actions such as move, instructions, make a simulation, edit a robot program (RAPID code) and upload and run the generated robot code on the real target robot.

Task for the manipulator: draw initials using a special module with a pen.

2 Methods

Before performing work on a real robot, it is necessary to create a model in simulation, create a program to perform the task, test and transfer the RAPID code to a real controller.

Robot Studio is a program developed by ABB for simulating and debugging software. A new Solution "Lab2" was created. Figure 1 shows the created environment for testing tasks on the IRB 1600 robot.

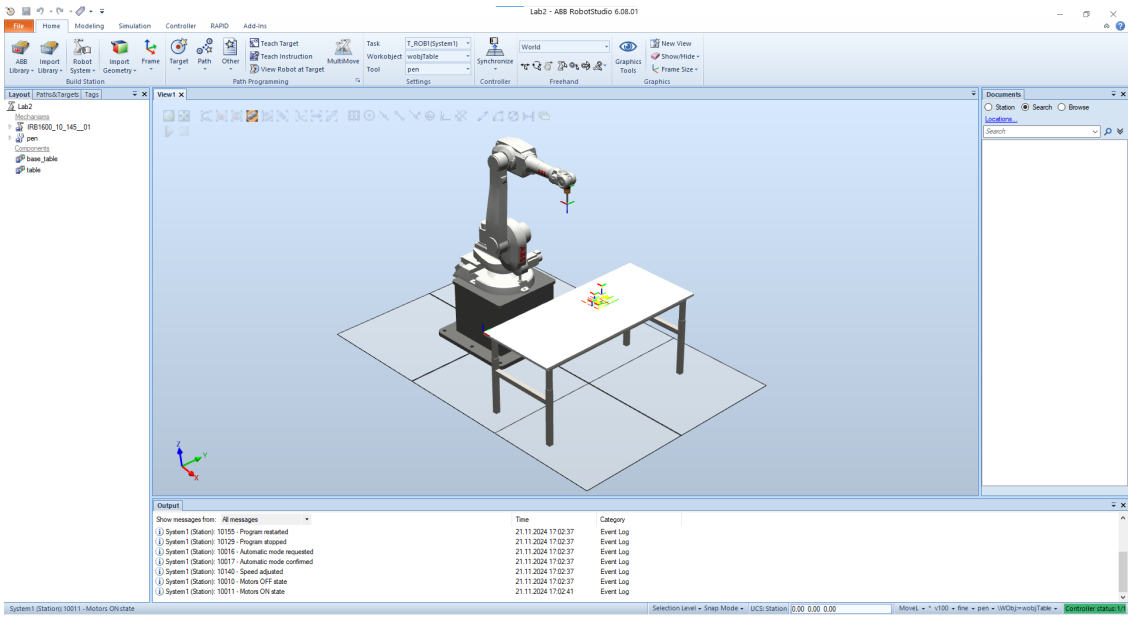

Figure 1: Simulation of IRB 1600

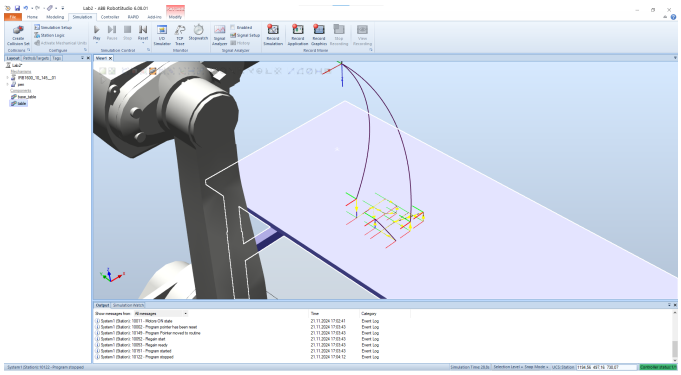

Figure 2: Desired trajectory

We need to draw "R E". Using the built-in tools, move the TCP and draw the desired trajectory of movement. We form paths and run the simulation. The trajectory is shown in Figure 2.

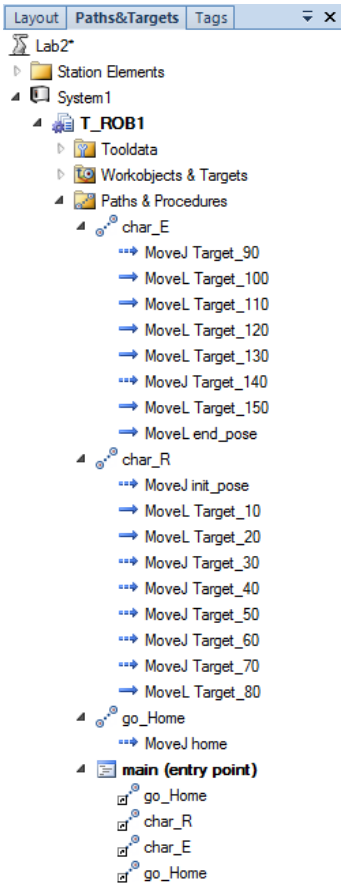

Figure 3: Structure of "Paths&Targets"

Structure of "Paths&Targets" is shown in Figure 3. Three paths were created: go_Home is for going to start position, char_R is for drawing "R" and "char_E is for drawing "E".

Next, we need to generate the RAPID code. This is done by pressing the "Synchronize to RAPID..." button. Structure of controller is presented on Figure 4 and codes of "Module 1" and "CalibData" are shown below:

```
MODULE Module1
    CONST robtarget
       init_pose:=[[216.461714459,1041.265991727,81.999249998],
    [-0.000000071,-0.000000006,1,-0.000000022],[0,-1,0,0],
    [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
       Target_10:=[[216.461718177,1041.265992381,-0.039384878],
    [-0.000000071,-0.000000002,1,-0.000000022],[0,-1,0,0],
    [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
       Target_20:=[[357.448040772,1041.266003704,-0.039371256],
    [-0.000000072,-0.000000006,1,-0.000000022],[0,-1,0,0],[
    9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

```
    CONST robtarget
       Target_30:=[[357.448046402,984.74128717,-0.039363275],
    [-0.000000057,-0.000000003,1,-0.000000021],[0,-1,0,0],
    [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
       Target_40:=[[327.012967805,958.724705453,-0.03934658],
    [-0.000000036,-0.000000006,1,-0.000000019],[0,-1,0,0],
    [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
       Target_50:=[[297.113959914,985.511099362,-0.03927025],
    [-0.000000067,-0.000000005,1,-0.000000021],[0,-1,0,0],
    [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
       Target_60:=[[297.113959919,1042.1893038,-0.03926682],
    [-0.000000055,-0.000000007,1,-0.000000019],[0,-1,0,0],
    [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
       Target_70:=[[220.193685145,958.869251733,-0.039269098],
    [-0.000000043,-0.000000008,1,-0.000000018],[0,-1,0,0],
    [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
       Target_80:=[[220.193690021,958.869252172,39.627339114],
    [-0.000000035,-0.000000009,1,-0.000000018],[0,-1,0,0],
    [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
       Target_90:=[[220.19369359,863.027556433,0.148656861],
    [-0.000000022,-0.00000001,1,-0.000000018],[-1,-1,-1,0],
    [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
       Target_100:=[[220.193689256,932.458171989,0.148743709],
    [-0.000000013,-0.000000012,1,-0.000000017],[0,-1,0,0],
    [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
       Target_110:=[[356.895981016,932.45816693,0.148727049],
    [-0.000000021,-0.000000011,1,-0.000000018],[0,-1,0,0],
    [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
       Target_120:=[[356.895968626,864.911696335,0.148697911],
    [-0.000000052,-0.000000011,1,-0.000000018],[-1,-1,-1,0],
    [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
       Target_130:=[[356.895982834,865.541145089,39.914148887],
    [-0.000000018,-0.000000011,1,-0.000000018],[-1,-1,-1,0],
    [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
       Target_140:=[[289.395030822,932.471849734,0.117301704],
    [-0.000000054,-0.000000011,1,-0.000000019],[0,-1,0,0],
    [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
       Target_150:=[[289.395036619,862.943540985,0.117327628],
    [-0.000000048,-0.000000011,1,-0.000000019],[-1,-1,-1,0],
    [9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
```

```
    end_pose:=[[289.395029601,862.943541045,40.413119991],
[-0.000000056,-0.000000011,1,-0.000000019],[-1,-1,-1,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget home:=[[750,0,982.5],[0,0,1,0],[0,0,0,0],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
!*********************************************************
!
! Module:  Module1
!
! Description:
!   <Insert description here>
!
! Author: R. Eidelman
!
! Version: 1.0
!
!*********************************************************


!*********************************************************
!
! Procedure main
!
!   This is the entry point of your program
!
!*********************************************************
PROC main()
    go_Home;
    char_R;
    char_E;
    go_Home;
    !Add your code here
ENDPROC
PROC char_R()
    MoveJ init_pose,v100,fine,pen\WObj:=wobjTable;
    MoveL Target_10,v100,fine,pen\WObj:=wobjTable;
    MoveL Target_20,v100,fine,pen\WObj:=wobjTable;
    MoveJ Target_30,v100,z30,pen\WObj:=wobjTable;
    MoveJ Target_40,v100,z100,pen\WObj:=wobjTable;
    MoveJ Target_50,v100,z40,pen\WObj:=wobjTable;
    MoveJ Target_60,v100,fine,pen\WObj:=wobjTable;
    MoveJ Target_70,v100,fine,pen\WObj:=wobjTable;
    MoveL Target_80,v100,fine,pen\WObj:=wobjTable;
ENDPROC
PROC char_E()
    MoveJ Target_90,v100,fine,pen\WObj:=wobjTable;
    MoveL Target_100,v100,fine,pen\WObj:=wobjTable;
    MoveL Target_110,v100,fine,pen\WObj:=wobjTable;
    MoveL Target_120,v100,fine,pen\WObj:=wobjTable;
    MoveL Target_130,v100,fine,pen\WObj:=wobjTable;
    MoveJ Target_140,v100,fine,pen\WObj:=wobjTable;
    MoveL Target_150,v100,fine,pen\WObj:=wobjTable;
    MoveL end_pose,v100,fine,pen\WObj:=wobjTable;
```

```
    ENDPROC
    PROC go_Home()
        MoveJ home,v100,fine,pen\WObj:=wobj0;
    ENDPROC
ENDMODULE
```

```
MODULE CalibData
    PERS tooldata pen:=[TRUE,[[0,0,139],[1,0,0,0]],
    [0.5,[0,0,20],[1,0,0,0],0,0,0]];
    TASK PERS wobjdata wobjTable:=[FALSE,TRUE,"",
    [[747.33,-871.732,210.071],[1,0,0,0]],[[0,0,0],[1,0,0,0]]];
ENDMODULE
```

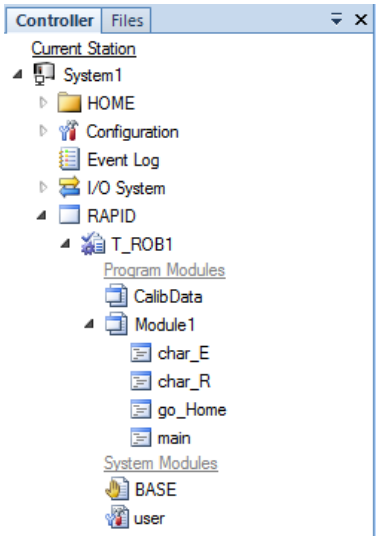

Figure 4: Structure of controller

We write main code for some moves into "Module1". There are special frames in "CalibData". "Lab2_controller" is for connecting all files in controller.

After showing and demonstrating robot simulation to teacher, we generate files for a real controller by pushing button "T_ROB1" with right-click and save program as folder with name "Lab2_controller".

Before the experiment, it is necessary to calibrate the workspace (in our case, the plane of the table on which the robot will draw). We need to change variables in "wobjTable" frame. Process of calibrating is shown in Figure 5. We measure the distance by which we want to move the TCP along the x, y, z axis. This will work because all the points of the trajectory are tied to the frame "wobjTable". By changing it we change all the points of our trajectory. New "CalibData" file with new workspace frame is shown below:

```
MODULE CalibData
    PERS tooldata pen:=[TRUE,[[0,0,139],[1,0,0,0]],
    [0.5,[0,0,20],[1,0,0,0],0,0,0]];
    TASK PERS wobjdata wobjTable:=[FALSE,TRUE,"",
    [[1047.33,-1171.73,298],[1,0,0,0]],[[0,0,0],[1,0,0,0]]];
ENDMODULE
```

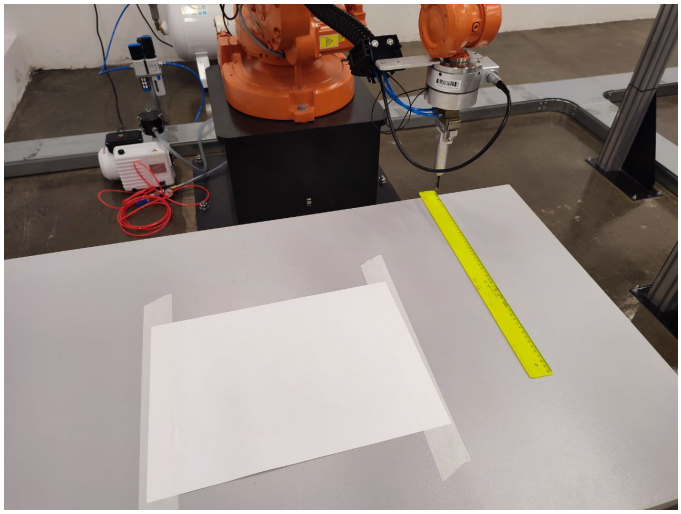

Figure 5: Calibration of workspace

A video recording of the experiments can be viewed at the link.

3 Results

Two experiments were conducted. In the video of the first experiment, it can be seen that the robot did not draw the letter 'E'. This is because the workspace frame is positioned too high. The z-axis was lowered by 3 mm. In the video of the second experiment, the robot clearly draws the letters. Figure 6 shows the result of the robot's work.

The modified RAPID code, updated using the real controller panel, was exported to a USB drive for further visualization in RobotStudio. The new controller code is either uploaded or the 'CalibData' is manually adjusted (the code was provided earlier). Using the 'Synchronize to Station...' button, the changes are transferred to the model. Figure 7 shows that the workspace frame is shifted compared to the initial version (Figure 1)

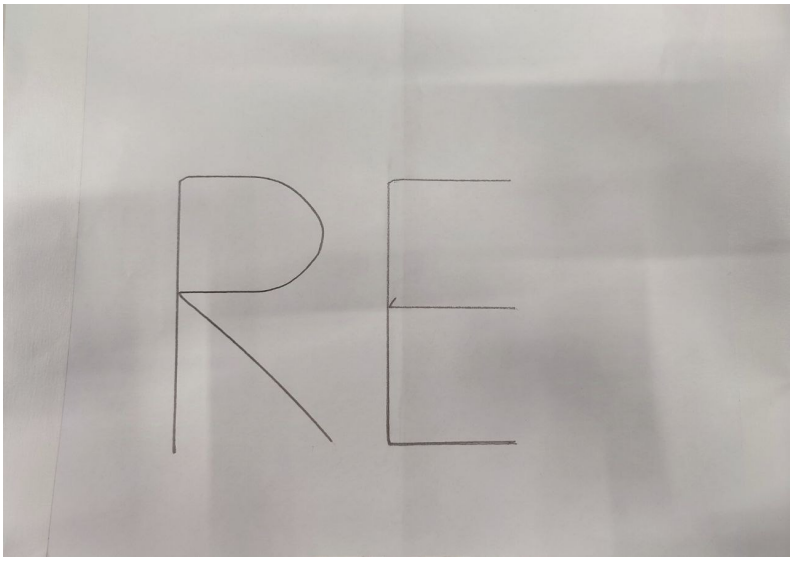

Figure 6: Results

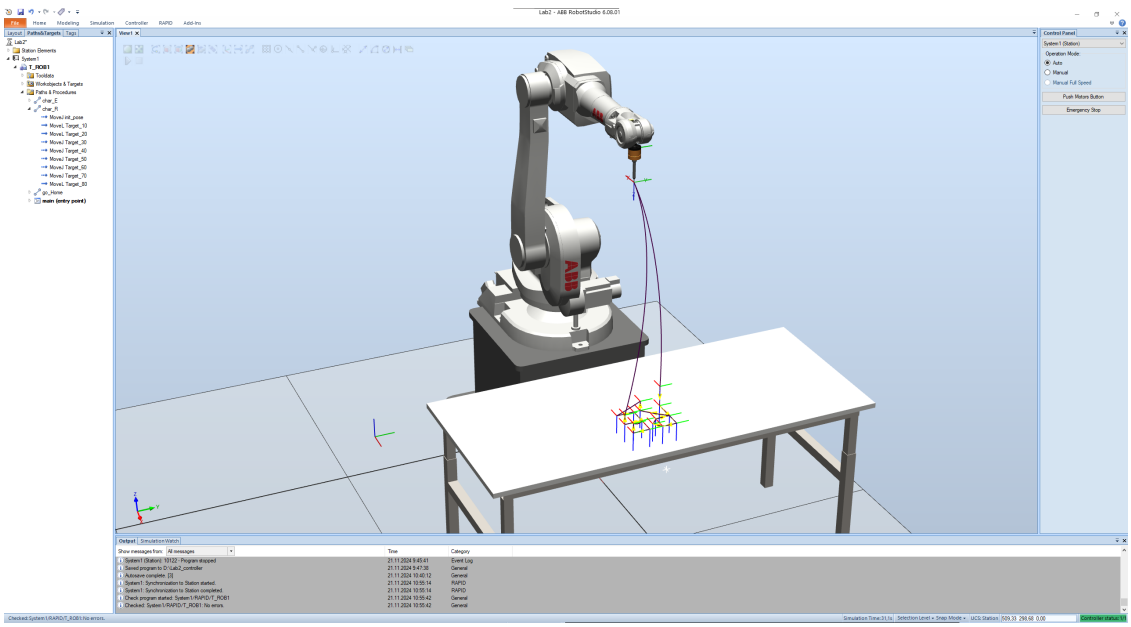

Figure 7: "Real" workspace frame

4 Discussion

In this laboratory assignment, the RobotStudio program was studied. A model of the manipulator was constructed. It was programmed and operated in the simulation. Additionally, this code was transferred to a real robot, and an experiment was conducted.

The experiment involved a TCP module equipped with a pen and a spring. The spring was needed to smooth out irregularities in the table surface. It was also necessary due to inaccuracies in modeling the contact point: for trajectory construction, "sliders" were used to move the TCP. For precise positioning, points should be set using coordinates, which will result in a more accurate trajectory.

Additionally, the workspace calibration process was very useful. It allows adjusting the robot to the surrounding environment rather than adapting the environment to the robot.