Introduction

Abb one of the leading suppliers of industrial robots installed more than 200,000 robots worldwide. Offline programming is the best way to training, programming, modeling and simulation of robot cells can be done in safety. It allows robot programming to be done in the office without shutting down production. ABB virtual controllers in the robot studio work as like the same of the real software that runs on the real robot. This software provides many tools to the better performance of your robot systems tasks such as training and optimization. Robot studio simulation performance is very realistic, using program and configuration files to identical real robots those used on the shop floor. When a robot studio is not connected to a real controller, or while it is connected to a virtual controller, this is called offline mode/ manual mode. Robot studio allowing custom and user-customized contents and paths to make a better user interface. ABB industrial robots for controlling used RAPID high-level programming language. this assignment will explain all the necessary procedures step-by-step to create a robot application and most useful tools. There are many benefits included risk reduction, quicker start-up, shorter change-over, increase productivity.

Explanation all parts program

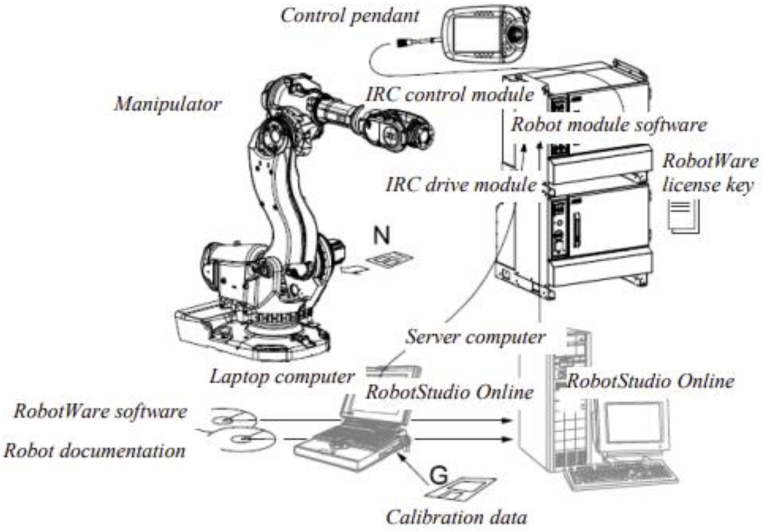


Figure 1.1 Industrial robot IRB 1600 of ABB and its components

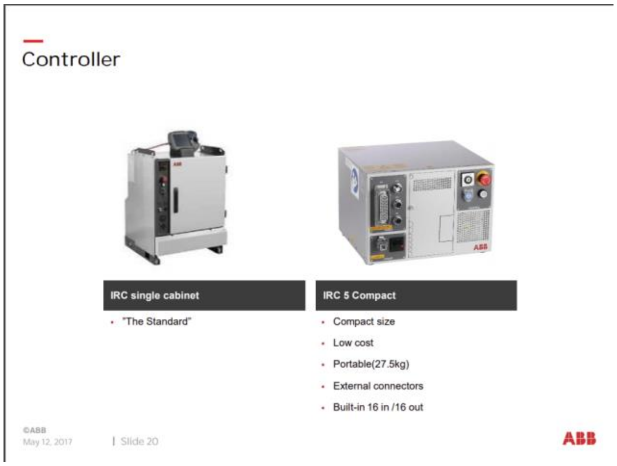
Robot Manipulator:

We have application for this assignment ABB IRB 1200 has 6 axes which is a fast, compact, flexible, and functional little industrial robot. ABB IRB1200 allows doing 10% faster work and 15% smaller than other robots. It is usable to Material handling and machine tending industries for flexibility to work short cycle times. Food Grade lubrication, SafeMove2, Foundry Plus 2 and clean room option are available to use with the robot. The longer reaches 900 mm different can bear up to 5kg of payload, this one we have used in our project.

Features and benefits of this robot:

1. It 15% of smaller cells that have 10% shorter cycle times
2. It is usable with Food Grade Lubrication
3. Deliberately with SafeMove2 Functionality
4. Available with foundry Plus 2 protection for harsh environments
5. 4 air ducts, 10 customer signals, and Ethernet routed internally
6. Mountable at any angle

Control Module:



Controller

IRC single cabinet

- "The Standard"

IRC 5 Compact

- Compact size
- Low cost
- Portable(27.5kg)
- External connectors
- Built-in 16 in /16 out

©ABB
May 12, 2017 | Slide 20

ABB

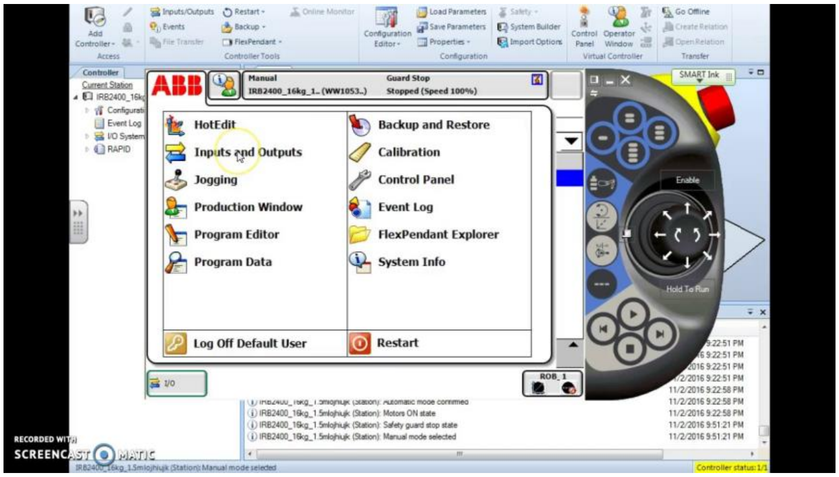
The control module contains all electronic control devices to maintain the robot system containing the main control computer of a robot or a central controller. One control module can be connected to 1-4 drive modules.

Drive Module: including power electronics of electrical drives. For the purpose of a multi-manipulator or so-called Multi-Move system, will be used various drive modules. It supplies power to the motors of a manipulator. This module can include up to 9 drive units, each controlling one manipulator joint. Since the quality manipulator has 6 joints, normally uses one drive module per robot controller.

Flex Controller:

The Controller cupboard for storage for the IRC5 robots. It included one control module and one drive module for a single robot manipulator in the system. It situated application interfaces, flexibility, safety, modularity, multi-robot control and computer support. The IRC5 takes various variants to supply cost-effective and optimized solutions for every need. All state-of-the-art support the majority of providing an industrial network for I/O. High-level programming language and ABB's flexibility of all ABB robot systems are programmed with RAPID Programming language.

Flex pendant:

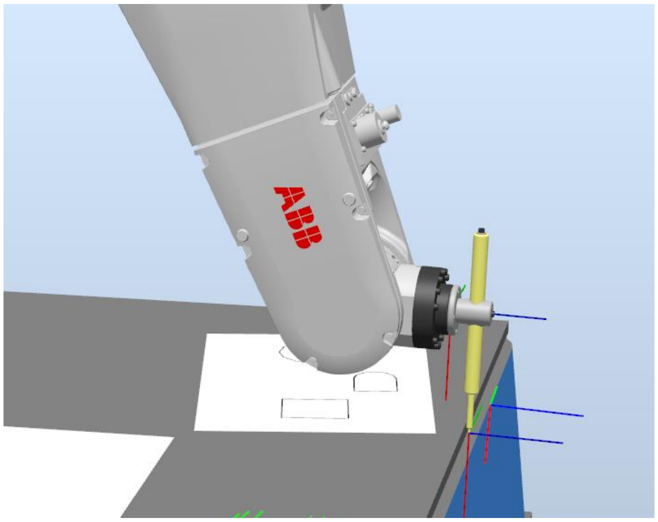


```
Manual                           Guard Stop
IRB2400_16kg_1..(WW1053..)       Stopped (Speed 100%)
```

```
                                                                   9:22:51 PM
                                                                  6 9:22:51 PM
                                                               2016 9:22:51 PM
                                                             /2/2016 9:22:51 PM
IRB2400_16kg_1.5mlojhiujk (Station): Automatic mode confirmed    11/2/2016 9:22:58 PM
IRB2400_16kg_1.5mlojhiujk (Station): Motors ON state             11/2/2016 9:22:58 PM
IRB2400_16kg_1.5mlojhiujk (Station): Safety guard stop state     11/2/2016 9:22:58 PM
IRB2400_16kg_1.5mlojhiujk (Station): Manual mode selected        11/2/2016 9:51:21 PM
                                                                 11/2/2016 9:51:21 PM
IRB2400_16kg_1.5mlojhiujk (Station): Manual mode selected
```

Flex pendant is connected to a robot Controller and usable to robot manual programming and control. Flex Pendant has a touch screen facility, a joystick and only 8 hardware push buttons for robot control. Programming on the Flex Pendant is referred to as "online programming". Flex pendant can help to write the program without coding knowledge and create, edit, update, delete variable to command the robot for work. There we can use a manual function where can customize the full program.

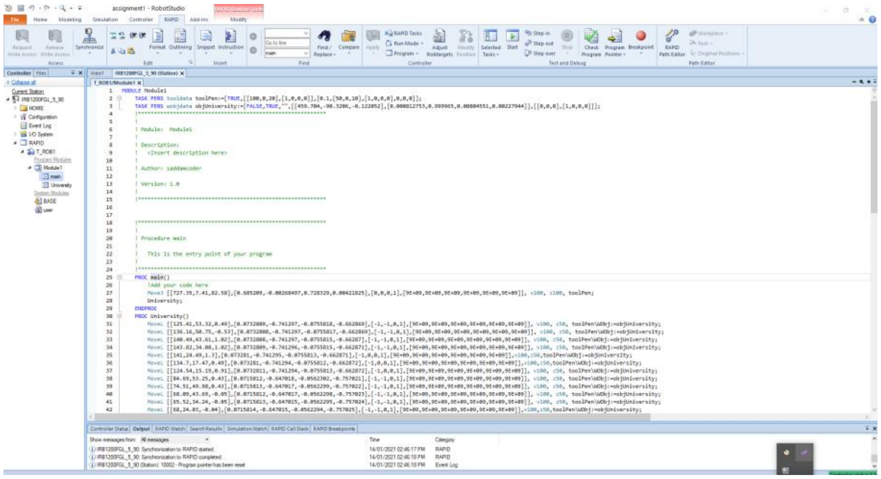
Tool:



A device that normally added to the robot manipulator to allow it to perform specific tasks, such as gripping, cutting or welding. This tool also changeable on a work purpose base, it is not mounted on the robot all time that's why it's called an external tool.

Robot Ware: Robot ware is used for robot adjustment, control and maintenance. Robot Ware is full of controller software designed to make work more productively and cost efficient of owning and operating a robot. It is a set of software files that, when loaded into a controller, able to get all functions, configurations, data and programs controlling the robot system. Robot Ware confirms optimal process performance through ABB's aid motion technology. With Robot Ware, it is possible to do exactly what you want.

RAPID Editor:



```
MODULE Module1
    TASK PERS tooldata toolPen:=[TRUE,[[100,0,20],[1,0,0,0]],[0.1,[50,0,10],[1,0,0,0],0,0,0]];
    TASK PERS wobjdata objUniversity:=[FALSE,TRUE,"",[[459.704,-90.3206,-0.122052],[0.000812753,0.999965,0.00804551,0.00227944]],[[0,0,0],[1,0,0,0]]];
    !***********************************************************
    !
    ! Module:  Module1
    !
    ! Description:
    !   <Insert description here>
    !
    ! Author: saddamcoder
    !
    ! Version: 1.0
    !
    !***********************************************************


    !***********************************************************
    !
    ! Procedure main
    !
    !   This is the entry point of your program
    !
    !***********************************************************
    PROC main()
        !Add your code here
        MoveJ [[727.35,7.41,82.58],[0.685209,-0.00268497,0.728329,0.00421825],[0,0,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z100, toolPen;
        University;
    ENDPROC
    PROC University()
        MoveL [[125.41,53.32,0.49],[0.0732809,-0.741297,-0.0755818,-0.662869],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen\WObj:=objUniversity;
        MoveL [[136.16,50.75,-0.53],[0.0732808,-0.741297,-0.0755817,-0.662869],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen\WObj:=objUniversity;
        MoveL [[140.49,43.61,1.02],[0.0732808,-0.741297,-0.0755815,-0.66287],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen\WObj:=objUniversity;
        MoveL [[143.02,34.08,1.02],[0.0732809,-0.741296,-0.0755815,-0.662871],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen\WObj:=objUniversity;
        MoveL [[141,24.49,1.3],[0.073281,-0.741295,-0.0755813,-0.662871],[-1,0,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v100,z50,toolPen\WObj:=objUniversity;
        MoveL [[134.7,17.47,0.49],[0.073281,-0.741294,-0.0755812,-0.662872],[-1,0,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v100,z50,toolPen\WObj:=objUniversity;
        MoveL [[124.54,15.19,0.91],[0.0732811,-0.741294,-0.0755813,-0.662872],[-1,0,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen\WObj:=objUniversity;
        MoveL [[84.69,53.25,0.43],[0.0715812,-0.647018,-0.0562302,-0.757021],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen\WObj:=objUniversity;
        MoveL [[74.51,49.98,0.43],[0.0715813,-0.647017,-0.0562299,-0.757022],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen\WObj:=objUniversity;
        MoveL [[68.09,43.69,-0.05],[0.0715812,-0.647017,-0.0562298,-0.757023],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen\WObj:=objUniversity;
        MoveL [[65.52,34.24,-0.05],[0.0715813,-0.647015,-0.0562295,-0.757024],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen\WObj:=objUniversity;
        MoveL [[68,24.03,-0.04],[0.0715814,-0.647015,-0.0562294,-0.757025],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v100,z50,toolPen\WObj:=objUniversity;
```

```
Show messages from:  All messages                                   Time                     Category
IRB1200FGL_5_90: Synchronization to RAPID started.                  14/01/2021 02:46:17 PM   RAPID
IRB1200FGL_5_90: Synchronization to RAPID completed.                14/01/2021 02:46:18 PM   RAPID
IRB1200FGL_5_90 (Station): 10002 - Program pointer has been reset   14/01/2021 02:46:18 PM   Event Log
```

ABB robot and use software to program it, as in all applications of industrial or automation equipment, must be known a programming language. In this case, ABB robot used the RAPID programming language. It is an integrated RAPID editor to view and edit the program loaded both real and virtual into a controller. With this RAPID code, you can edit program modules and system modules.

There is so many features's included in RAPID editors:

1. It can highlight the syntax and error
2. There has a facility to find and replace a program
3. Tooltips help to know symbols, arguments, function calls
4. Context-sensitive help for RAPID code
5. Indicate the number of lines in the RAPID code
6. Editing RAPID modules in controller memory and in files.

Appropriate use of movement types

Move C: The move C instruction habituated to move the Tool Centre point (TCP) circularly to a given point, while the movement orientation usually unchangeable relative to the circle. When want to make a curved path this movement is very important to be carried out by the robot. Other, movement types take one position, but this movement takes two positional arguments, the first position arguments on the circle between the start point and semi-circle point and the second position end of the point of the semi-circle. The orientation in the circle points complete (360 degrees) circular path.

Example: MoveC p1, p2, v100, z30, tool2;

Move L: Move L is frequently used when designing a model of the path operation of the robot. Move L is used to Tool Centre Point (TCP) moves linearly straight line from one point to next particularized point. This move instruction can only be used in the main task T-ROB1.

Example: MoveL p1, v100, z30, tool2;

Move J: Move j can move the robot quickly from one point to another when the movement doesn't need to be in a straight line. Move j is used both linear and joint motion to jog the robot. This direction is assigned to move the robot's Tool Centre Point (TCP) by a joint movement to a position or location. When the programmer's specified point the robot can easily take any path. It can work external axes move a non-linear path from destination position and all axes move there at the same time. If the robot has Multi Move system or motion tasks MOVE J can only be used in the main task T-ROB1.

Example: MoveJ p1, v100 , z30, tool2;

Movement is deciding how to use a joystick. There is a big difference between moveJ all instruction path goes to a non-linear path. MoveL is mostly used in the work field to make a design path by linear model. MoveC is used to complete the circular path task. At the beginning of the program moveJ is used at the first move instruction, moveL and moveC familiar with the main tasks.

Appropriate use and explanation of coordinate systems

The Coordinate system influences the jogging directions and used to define position and orientations. There are many types of coordinate systems without difficulty position objects relative to each different. Offline programming can use coordinate systems. Presently, industrial robots use the “right hand” cartesian coordinate system.

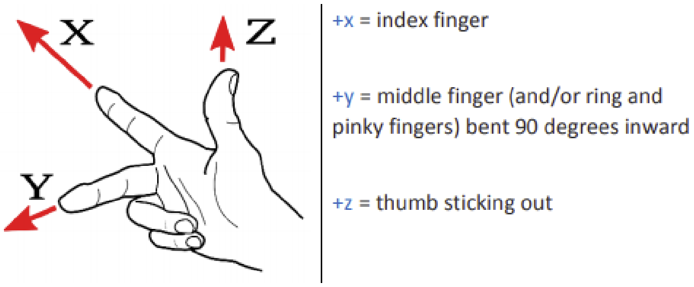
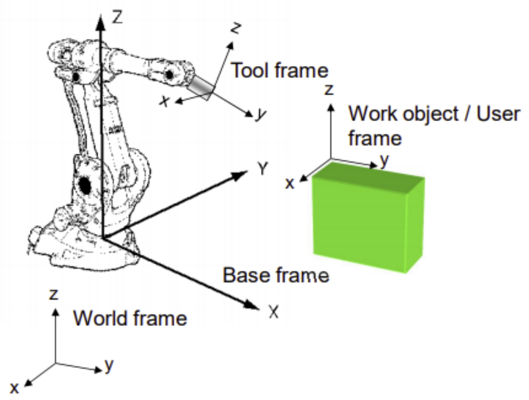

X
Z
Y
+x = index finger
+y = middle finger (and/or ring and pinky fingers) bent 90 degrees inward
+z = thumb sticking out

Figure 1.3. “Right hand” Cartesian coordinate system configuration



To help in the programming it is possible to define a number of coordinate systems. Some can be defined/calibrated using the robot.

The base Coordinate system: the base coordinate system is always located base of the robot, whether both in robot studio and the real world. It is always attached to the robot base mounting surface. There z-axis point coincident upwards and x axis, y axis plane is base of the same mounting surface. For instance, floor mounted robots are usually programmed in the base coordinate system.

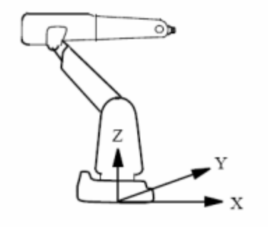
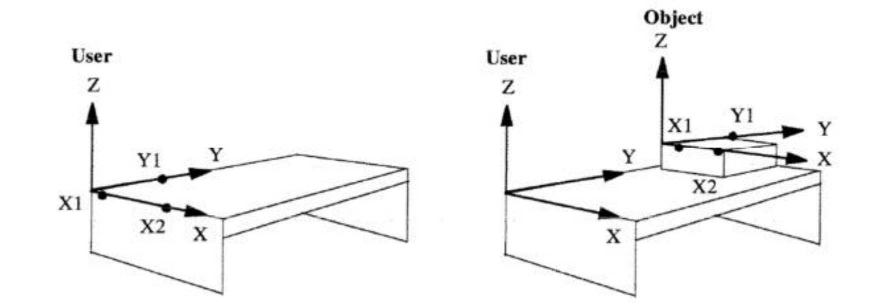


Figure 1.6 Base coordinate system of a manipulator

The Tool Coordinate system: the tool Centre point coordinate system also called TCP which is the Centre point of the tool. Different TCPs at the Robt's have one predefined tool, called tool0. When running the program, the robot moves to the programmed position. This system can get suitable motion direction when jogging the robot. When damaged and replaced the tool coordinate system must be redefined. when the robot is in motion and we do not accept to change the orientation of the tool that purpose used the tool coordinate system. the robotics programmed time moves the TCP from one point to another programmed point reason is this TCP must be defined before starting programming.

The work object Coordinate system: in the work object target is a point in space defined by its coordinates system position, orientation, configuration. The three-point method defined a work object coordinate system. This system programmed by specifying numeric value or jogging the robot through a few each position specified in object coordinates with following to the tool's position and orientation. when a tool is changed then it is damaged, for making a new definition of tool old programmed can be used. This coordinate system is helping to store the programmed position. The robot can work with multiple work object systems which is done in this system. User frame and object frame composed of two coordinates system of object coordinate when programmed a robot always all targets are involved in the object frame of a work object. In the work object target attached to the default wobj0, it always coincides with the base frame of the robot. To set the workpiece there used three points that define the position of that, two X- axis and one Y-axis point. For this project established three work objects objUni, objName, objShap.



The world Coordinate system: a reference to the floor defines the world coordinate system. It helps to robot position of a fixed point in the shop floor. This robot helps two robots work together with a robot carrier. This coordinate system is referred to as the other coordinates system when the same working space several robots work at a plant. It this advantageous to apply this type of system when the related position is a fixed point in the workshop.

Appropriate use of velocity data

Each move instruction contains a velocity value (mm/s) e.g., v1000= 1m/s. When set the highest velocities do not seem to make the robot any faster. The velocity of the TCP in mm/s directly in the instruction. It will be substituted for the corresponding speed data. As defining the velocity of the robot and external axes, it can set the time speed for the movement can be programmed. For instance, v100 is the speed of the robot's velocity to move with. The user defined this variable to get the value in a programmed position. In the flex pendant, there have a value Manu list range from v5 m/s to above 1000 m/s. The velocity data always default speed is v 1000 m/s, but for this project, we need 100 m/s.

Appropriate use of Tool data

Tool data is familiar with explained the characteristics of a tool. If the tool is fixed with a robot, the tool data is defined tool and the work object of gripper holding. Tool data is the need for the tool to simulate the robot tool. Different tools that may need to use with appreciation from the tool data clarify the programming work. For various tools defined the isolated sets of tool data make it achievable to run a similar robot program with different tools. When making a new tool data must be defined. The tool data include the information needed for moving and simulating the tool. this project has used a tool to draw the shape it is imported from browse geometry then make a tool data which name is tool pen. There defined the position of the tool data and rotation of the tool data than defined the weight and Center of gravity of the tool with inertia.

Appropriate use of zone data

How near to the destination position, zone data describes the axes must be before to next target point can be executed. When the robot is programmed to move to a point, where that doesn't need to arrive at the specified position. Zone data is a variable that helps to the robot to be closed the programmed position before moving towards the next target point. In most industries need to be robots worked with exact position programmed to use "fine" zone data. The fine criteria, that informed when the robot is considered to have arrived at the point, it is manipulated. The value of the zone data RAPID rand from z5 to z200.

Appropriate use of variable names(10%)

A variable is a defined quantity by the user in the robot control system. The start of the program needs to declare the robot about a variable. It sends the controller whether it is an Integer, a real, a Boolean or a string. If you try to put text in an integer, it will not work. The program can take the decision of type and save the variable value. Suddenly, the program is stopped and begins the variable keeps its value, when variable data moved from the program pointer the data value is lost. There is showing the variable assigning value it means the right is passed the variable on the left. We have used in this assignment the variable to show the counting increment data.

```
Example: VAR num length: = 10;
  VAR String name:= saddam;
  VAR bool finished: = True;
```

There are listing some feature of the variables:

1. During the program execution a variable can declare a new value
2. Variable has Boolean (true or false)
3. Real variables have decimal points
4. A string variable can be a single character or line of text
5. Variable store the data value
6. Declare a variable which data type it should have.
7. A constant present a static value and not accredit a new value.

Inclusion of a user interface

this project has used many functions to show a better user interface. There is describing the interface list that have used in this project:

TPErase: TPErase is used to clear the display of the FlexPendant.The next time is written it will be entered on the uppermost line of the display. For instance, in flexPendant then program editor to click add instructions and from the communicate list select TPErase.

TPWRITE: In FlexPendant to write the text is used TPWrite. There the value of specified data can be written as well as text. When the robot is working, we often want to send messages to the user or ask for input from the user. This is achieved using the TPWRITE. For instance, in flexPendant then program editor to click add instructions and from the communicate list select TPWrite.

TPreadFK: Flexpendant read function key is used to write text on the functions keys and to find out which key is depressed.

TEST: A test is used when different instruction needs to executed depending on the value of an expression or data.

Proc.call: It is used to set the routing in the test case then the user can select which one they want to see.

Incr: incr is used to increase 1 numeric variable. It used to be counted how many times draw the robot on a different shapes.

Rapid Programming Code

```
MODULE Module1

TASK PERS tooldata toolPen:=[TRUE,[[100,0,20],[1,0,0,0]],[0.1,[50,0,10],[1,0,0,0],0,0,0]];

TASK PERS wobjdata objUniversity:=[FALSE,TRUE,"",[[459.76,-90.1561,0.0700791],[0.00418974,0.999951,0.00823316,0.00349794]],[[0,0,0],[1,0,0,0]]];

TASK PERS wobjdata objName:=[FALSE,TRUE,"",[[234.894,103.72,2.90935],[0.00198115,0.999982,0.00391042,0.00411121]],[[0,0,0],[1,0,0,0]]];

TASK PERS wobjdata objShape:=[FALSE,TRUE,"",[[403.819,428.003,0.0320855],[0.000825506,0.996191,0.0871904,0.00115584]],[[0,0,0],[1,0,0,0]]];

CONST robtarget U1:=[[567.80,-371.90,0.24],[0.759116,0.0705116,0.643524,-0.0681693],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Shape1:=[[75.25,204.3,0.61],[0.0797013,0.751873,-0.115924,0.644125],[0,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Shape11:=[[134.52,204.3,0.61],[0.0797013,0.751873,-0.115924,0.644125],[0,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Shape21:=[[134.52,258.07,0.61],[0.0797013,0.751873,-0.115924,0.644124],[0,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Shape31:=[[76.07,258.07,0.61],[0.0797013,0.751873,-0.115924,0.644124],[0,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Shape41:=[[76.07,203.99,0.61],[0.0797013,0.751873,-0.115924,0.644124],[0,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Shape51:=[[140.55,177.88,0.61],[0.0797013,0.751873,-0.115924,0.644124],[0,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Shape61:=[[179.51,177.88,0.61],[0.0797013,0.751873,-0.115924,0.644124],[0,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Shape71:=[[179.51,137.89,0.61],[0.0797013,0.751874,-0.115924,0.644124],[0,0,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Shape91:=[[140.74,138.08,0.62],[0.0797018,0.75187,-0.115923,0.644128],[0,0,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Shape81:=[[159.85,119.43,0.61],[0.0797008,0.751874,-0.115924,0.644123],[0,0,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Shape101:=[[140.74,177.91,0.62],[0.0797018,0.751869,-0.115923,0.644129],[0,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Shape111:=[[104.54,93.24,0.62],[0.0797017,0.751869,-0.115923,0.644129],[0,0,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Shape131:=[[104.76,24.12,-0.18],[0.0797004,0.751873,-0.115924,0.644125],[0,0,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Shape121:=[[139.17,58.59,0.61],[0.0797008,0.751873,-0.115924,0.644124],[0,0,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Shape151:=[[107.28,92.33,-0.38],[0.0797034,0.751867,-0.115923,0.644132],[0,0,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Shape141:=[[72.42,58.87,-0.38],[0.0797025,0.751867,-0.115922,0.644131],[0,0,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget PHome:=[[656.76,-50.64,169.64],[0.606705,-0.0115484,0.794713,-0.0143731],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Us10:=[[586.44,-141.41,0.2],[0.761056,0.0207771,0.641038,-0.0971184],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Us20:=[[604.42,-121.6,0.2],[0.761056,0.0207772,0.641039,-0.0971184],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Us30:=[[584.82,-103.02,0.27],[0.781259,0.0638459,0.617543,-0.064799],[-1,0,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Us40:=[[544.61,-141.98,0.27],[0.78126,0.0638457,0.617541,-0.0647992],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Us50:=[[525.71,-123.01,0.27],[0.781259,0.0638459,0.617543,-0.0647991],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Us60:=[[544.79,-103.36,0.27],[0.78126,0.0638458,0.617542,-0.0647992],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Names10:=[[169.06,61.25,0.08],[0.0538504,0.619983,-0.0626806,0.780252],[0,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Names20:=[[187.05,38.51,0.08],[0.0538504,0.619982,-0.0626807,0.780252],[0,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Names30:=[[153.74,34.70,0.08],[0.0538504,0.619983,-0.0626807,0.780252],[0,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Names40:=[[102.36,67.48,-0.25],[0.0538501,0.619984,-0.0626807,0.780251],[0,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Names50:=[[74.08,53.58,-0.25],[0.0538502,0.619984,-0.0626808,0.78025],[0,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Names60:=[[102.02,22.46,-0.25],[0.0538504,0.619984,-0.0626808,0.78025],[0,0,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name10:=[[170.04,61.2,-0.25],[0.0538501,0.619984,-0.0626806,0.780251],[0,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Name30:=[[157.69,34.25,-0.25],[0.05385,0.619984,-0.0626808,0.780251],[0,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name20:=[[187.31,38.48,-0.25],[0.0538501,0.619983,-0.0626808,0.780252],[0,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name40:=[[102.94,67.27,-0.25],[0.05385,0.619987,-0.0626809,0.780249],[0,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name60:=[[103.65,22.99,-0.25],[0.0538502,0.619985,-0.0626808,0.78025],[0,0,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name50:=[[66.07,49.64,-0.25],[0.05385,0.619986,-0.062681,0.780249],[0,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name70:=[[68.83,79.77,-0.25],[0.0538502,0.619985,-0.0626807,0.78025],[0,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name80:=[[172.22,105.02,-0.25],[0.0538499,0.619983,-0.0626809,0.780251],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name90:=[[78.53,140.59,-0.25],[0.0538496,0.619983,-0.062681,0.780251],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name100:=[[128.11,126.59,-0.25],[0.0538498,0.619983,-0.0626812,0.780251],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name110:=[[128.11,89.49,-0.25],[0.0538498,0.619983,-0.0626812,0.780252],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name120:=[[80.1,150.53,-0.25],[0.0538495,0.619982,-0.0626814,0.780252],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name130:=[[179.87,148.38,-0.25],[0.0538495,0.619982,-0.0626815,0.780252],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name140:=[[179.87,175.03,-0.25],[0.0538495,0.619982,-0.0626815,0.780252],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name160:=[[83.29,150.4,-0.25],[0.0538502,0.619982,-0.062681,0.780252],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name150:=[[132.64,187.79,-0.25],[0.0538496,0.619982,-0.0626813,0.780252],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name170:=[[83.29,198.88,-0.25],[0.0538503,0.619982,-0.0626808,0.780252],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name180:=[[171.29,198.88,-0.25],[0.0538503,0.619983,-0.0626808,0.780252],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name190:=[[171.29,218.61,-0.25],[0.0538503,0.619983,-0.0626808,0.780252],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name210:=[[72.7,198.46,-0.32],[0.0538505,0.619983,-0.0626802,0.780251],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST robtarget Name200:=[[135.69,223.94,-0.25],[0.0538499,0.619983,-0.0626806,0.780252],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name220:=[[72.7,227.21,-0.32],[0.0538505,0.619984,-0.0626801,0.780251],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name230:=[[174.16,244.75,-0.32],[0.0538505,0.619984,-0.0626802,0.780251],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name240:=[[48.98,255.47,-0.32],[0.05385,0.619983,-0.0626803,0.780251],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name250:=[[123.11,255.47,-0.32],[0.0538499,0.619983,-0.0626804,0.780251],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name260:=[[123.11,227.49,-0.32],[0.0538497,0.619983,-0.0626805,0.780251],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name270:=[[59.41,266.83,-0.32],[0.0538496,0.619983,-0.0626805,0.780251],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name280:=[[179.39,266.83,-0.32],[0.0538496,0.619983,-0.0626805,0.780251],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name290:=[[147.31,279.78,-0.32],[0.0538496,0.619983,-0.0626805,0.780251],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name300:=[[185.43,292.67,-0.32],[0.0538498,0.619983,-0.0626803,0.780251],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Name310:=[[58.43,292.67,-0.32],[0.0538497,0.619983,-0.0626802,0.780251],[-1,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR num MyChoice:=0;

VAR num NumUni:=0;

VAR num NumShape:=0;

VAR num NumName:=0;

   !***********************************************************

   !

   ! Module:  Module1

   !

   ! Description:

   !   <Insert description here>

   !

   ! Author: saddamcoder

   !
```

```
    ! Version: 1.0
    !
    !***********************************************************


    !***********************************************************
    !
    ! Procedure main
    !
    !   This is the entry point of your program
    !
    !***********************************************************
    PROC main()
        !Add your code here
        MoveJ PHome, v100, z50, toolPen;
        TPErase;
        TPWrite "Drawing Demo work";
        TPWrite "Number of university name="\Num:=NumUni;
        TPWrite "Number of shape="\Num:=NumShape;
        TPWrite "Number of Name="\Num:=NumName;
        TPReadFK MyChoice, "Which want to draw", "University Name", "Shapes", "My Name", stEmpty, stEmpty;
        TEST Mychoice
        CASE 1:
            University;
        CASE 2:
            Shape;
        CASE 3:
            Name;
        ENDTEST
    ENDPROC
PROC University()
```

```
MoveL Us10, v100, z50, toolPen;

MoveC Us20, Us30, v100, fine, toolPen;

MoveL Us40, v100, fine, toolPen;

MoveC Us50, Us60, v100, fine, toolPen;

MoveL [[525.59,-148.44,0.08],[0.759116,0.0705105,0.643524,-0.068173],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen;

MoveL [[604.83,-167.01,0.08],[0.759121,0.0705099,0.643519,-0.0681739],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen;

MoveL [[526.16,-187.2,0.08],[0.759122,0.0705097,0.643517,-0.0681741],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v100,z50,toolPen;

MoveL [[556.75,-179.33,0.25],[0.759121,0.0705099,0.643518,-0.068174],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen;

MoveL [[556.75,-155.74,0.25],[0.759121,0.0705099,0.643519,-0.0681739],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen;

MoveL [[605.68,-192.57,0.25],[0.759122,0.0705096,0.643517,-0.0681742],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, fine, toolPen;

MoveL [[526.95,-193.33,0.25],[0.759122,0.0705096,0.643517,-0.0681742],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen;

MoveL [[526.95,-222.68,0.25],[0.759122,0.0705095,0.643517,-0.0681743],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen;

MoveL [[526.95,-228.34,0.25],[0.759123,0.0705095,0.643517,-0.0681744],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen;

MoveL [[606.19,-227.53,0.25],[0.759122,0.0705095,0.643517,-0.0681744],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen;

MoveL [[606.19,-256.81,0.25],[0.759123,0.0705095,0.643517,-0.0681744],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, fine, toolPen;

MoveL [[566.24,-228.14,0.25],[0.759122,0.0705095,0.643517,-0.0681744],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, fine, toolPen;

MoveL [[566.24,-252.21,0.25],[0.759123,0.0705094,0.643516,-0.0681744],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen;

MoveL [[541.86,-263.05,0.25],[0.759124,0.0705092,0.643515,-0.0681747],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen;

MoveL [[591.58,-262.53,0.25],[0.759124,0.0705093,0.643515,-0.0681747],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen;

MoveC [[606.54,-276.92,0.25],[0.759126,0.0705084,0.643512,-0.0681753],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[[592,-291.62,0.25],[0.759127,0.070508,0.643511,-0.0681753],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v100,z10,toolPen;
```

```
MoveL [[542.49,-292.25,0.25],[0.759127,0.0705079,0.643511,-0.0681751],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen;

MoveC [[527.57,-278.08,0.25],[0.759124,0.0705092,0.643515,-0.0681743],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], [[542.03,-263.15,0.25],[0.759123,0.0705098,0.643515,-0.0681748],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z10, toolPen;

MoveL [[527.94,-298.44,0.25],[0.759124,0.0705097,0.643515,-0.0681749],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, fine, toolPen;

MoveL [[606.88,-297.52,0.25],[0.759124,0.0705096,0.643515,-0.0681753],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen;

MoveC [[587.64,-326.67,0.25],[0.759125,0.0705092,0.643514,-0.0681751],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], [[567.38,-297.98,0.25],[0.759122,0.0705104,0.643517,-0.0681743],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, fine, toolPen;

MoveL [[527.81,-327.13,0.25],[0.759122,0.0705096,0.643517,-0.0681729],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, fine, toolPen;

MoveL [[528.73,-333.39,0.25],[0.759123,0.0705093,0.643517,-0.0681731],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen;

MoveL [[606.52,-332.55,0.24],[0.759116,0.0705123,0.643525,-0.06817],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen;

MoveL [[602.16,-352.31,0.24],[0.759116,0.070512,0.643524,-0.06817],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v100, z50, toolPen;

MoveC [[567.8,-371.9,0.24],[0.759116,0.0705116,0.643524,-0.0681693],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[[528.16,-333.29,0.24],[0.759115,0.0705127,0.643526,-0.0681699],[-1,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v100,z10,toolPen;

Incr NumUni;

ENDPROC

PROC Name()

    !Add your code here

    MoveL Name10, v100, fine, toolPen\WObj:=objName;

    MoveC Name20, Name30, v100, fine, toolPen\WObj:=objName;

    MoveL Name40, v100, fine, toolPen\WObj:=objName;

    MoveC Name50, Name60, v100, fine, toolPen\WObj:=objName;

    MoveL Name70, v100, fine, toolPen\WObj:=objName;

    MoveL Name80, v100, fine, toolPen\WObj:=objName;

    MoveL Name90, v100, fine, toolPen\WObj:=objName;

    MoveL Name100, v100, fine, toolPen\WObj:=objName;

    MoveL Name110, v100, fine, toolPen\WObj:=objName;
    MoveL Name120, v100, fine, toolPen\WObj:=objName;
    MoveL Name130, v100, fine, toolPen\WObj:=objName;
    MoveL Name140, v100, fine, toolPen\WObj:=objName;
    MoveC Name150, Name160, v100, fine, toolPen\WObj:=objName;
    MoveL Name170, v100, fine, toolPen\WObj:=objName;
    MoveL Name180, v100, fine, toolPen\WObj:=objName;
    MoveL Name190, v100, fine, toolPen\WObj:=objName;
    MoveC Name200, Name210, v100, fine, toolPen\WObj:=objName;
    MoveL Name220, v100, fine, toolPen\WObj:=objName;
    MoveL Name230, v100, fine, toolPen\WObj:=objName;
    MoveL Name240, v100, fine, toolPen\WObj:=objName;
    MoveL Name250, v100, z50, toolPen\WObj:=objName;
    MoveL Name260, v100, z50, toolPen\WObj:=objName;
    MoveL Name270, v100, z50, toolPen\WObj:=objName;
    MoveL Name280, v100, z50, toolPen\WObj:=objName;
    MoveL Name290, v100, z50, toolPen\WObj:=objName;
    MoveL Name300, v100, z50, toolPen\WObj:=objName;
    MoveL Name310, v100, z50, toolPen\WObj:=objName;
    Incr NumName;
ENDPROC
PROC Shape()
MoveL Shape1, v100, z50, toolPen\WObj:=objShape;
MoveL Shape11, v100, fine, toolPen\WObj:=objShape;
MoveL Shape21, v100, fine, toolPen\WObj:=objShape;
MoveL Shape31, v100, fine, toolPen\WObj:=objShape;
MoveL Shape41, v100, fine, toolPen\WObj:=objShape;
WaitTime 2;
MoveL Shape51, v100, z50, toolPen\WObj:=objShape;
MoveL Shape61, v100, z50, toolPen\WObj:=objShape;
MoveL Shape71, v100, z50, toolPen\WObj:=objShape;
MoveC Shape81, Shape91, v100, z10, toolPen\WObj:=objShape;
```

```
MoveL Shape101, v100, z50, toolPen\WObj:=objShape;

WaitTime 2;

MoveL Shape111, v100, z50, toolPen\WObj:=objShape;

MoveC Shape121, Shape131, v100, z10, toolPen\WObj:=objShape;

MoveC Shape141, Shape151, v100, z10, toolPen\WObj:=objShape;

Incr NumShape;

ENDPROC

ENDMODULE
```

Conclusion

Robot studio is offline program the robot software where we have learned like real by the virtual controller. It is an excellent tool to advance knowledge of the robot programming and operation of an ABB Robot.

Reference:

1. https://new.abb.com/products/robotics/robotstudio
2. A guide for abb robotstudio, Pedro Neto, 3030-788 coimbra, portugal
3. Technical reference manual RAPID Instructions, Robotware 5.13, Revesion: J
4. Operating manual robot studio, 5.13, Revesion: D
5. https://new.abb.com/products/robotics/industrial-robots/irb-1200
6. https://new.abb.com/products/robotics/controllers/irc5