THE NATIONAL
ROBOTARIUM
PEOPLE CENTRED :: INTELLIGENCE DRIVEN

ABB YuMi Dual Arm Robot and ABB RobotStudio Hands-On Training Part 2

Provided By,

Rahul R. Ramachandran
Robotics Engineer
Manipulation team Lead
The National Robotarium (UK)

r.ramachandran@hw.ac.uk

Agenda

I. Getting Started with ABB YuMi Dual Arm

II. Jogging of the ABB YuMi Dual Arm

III. Online Programming of the ABB YuMi using FlexPendant

IV. Offline Programming of the ABB YuMi using RobotStudio

Online Programming of the ABB YuMi using FlexPendant

Topics Covered:

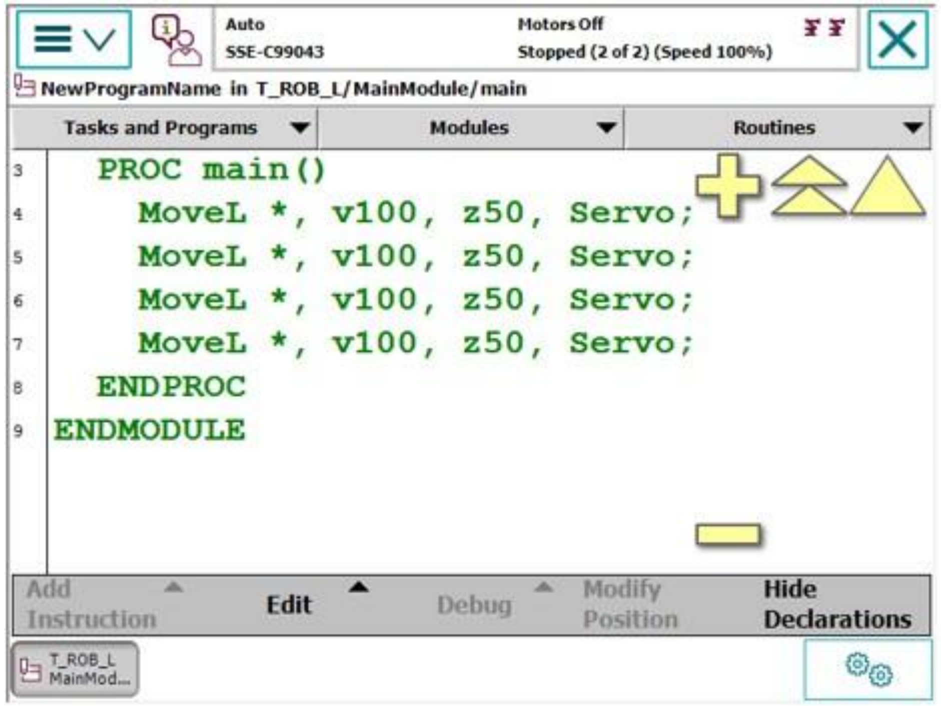
1. Introduction
2. Coding
3. Program editor
4. Program data screen
5. Production window screen

Online Programming of the ABB YuMi using FlexPendant

1. Introduction

Why code the ABB YuMi?

In jogging, we need to manually move the robot. Coding is helpful in setting up precise and automatic movements for the YuMi. The coding language used for the ABB YuMi® is RAPID.



```
  PROC main()
    MoveL *, v100, z50, Servo;
    MoveL *, v100, z50, Servo;
    MoveL *, v100, z50, Servo;
    MoveL *, v100, z50, Servo;
  ENDPROC
ENDMODULE
```

Online Programming of the ABB YuMi using FlexPendant

2. Coding

RAPID is the coding language used by ABB YuMi®.

The most common movement codes are MoveL, MoveJ, MoveAbsJ.

- MoveL is used to move the tool center point (TCP) linearly from one point to another.
- MoveJ is used to move the robot from one point to another when that movement does not have to be in a straight line.
- MoveAbsJ is used to move the robot and external axes to an absolute position defined in axes positions.

MoveAbsJ [target] , [velocity] , [zone] , [tool] , [workobject] , ;

Online Programming of the ABB YuMi using FlexPendant

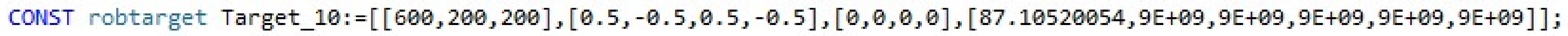
2. Coding

Target : The parameters of the target can be added into the move function directly or by creating a constant variable.

- MoveL and MoveJ use robtarget as their constant variable.

robtarget [target_name] := [[x, y, z],[q1, q2, q3, q4],[cf1, cf2, cf3, cf4],[eax_a, ...]];

This variable contains target position data such x, y, z, orientation, axis configuration, and position of external axis. For example;



```
CONST robtarget Target_10:=[[600,200,200],[0.5,-0.5,0.5,-0.5],[0,0,0,0],[87.10520054,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

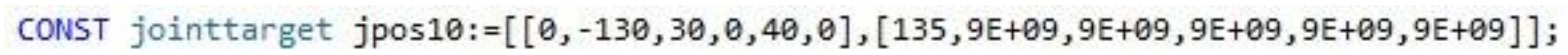
Online Programming of the ABB YuMi using FlexPendant

2. Coding

- MoveAbsJ uses jointtarget as its constant variable:

jointtarget [target_name] := [[j1, j2, j3, j4, j5, j6] , [j7, ...];

This variable contains the angle of each joint, there are only 7 joints, so the rest of values are infinity/ 9E+09. For example;



```
CONST jointtarget jpos10:=[[0,-130,30,0,40,0],[135,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

Velocity : The velocity data can be changed and is in mm/sec.

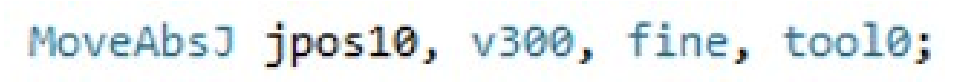
Zone : The zone data describes the size of the generated corner path, fine can be selected so that it goes to the exact point without cutting the corner.

Tool : The tool data stores any tools with parameters added. tool0 at default.

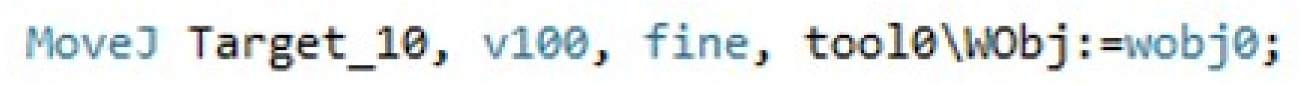
Online Programming of the ABB YuMi using FlexPendant

2. Coding

Workobject and Payload: The workobject and payload data are considered default when not added in the code. wobj0 and load0 at default. For example;



```
MoveAbsJ jpos10, v300, fine, tool0;
```



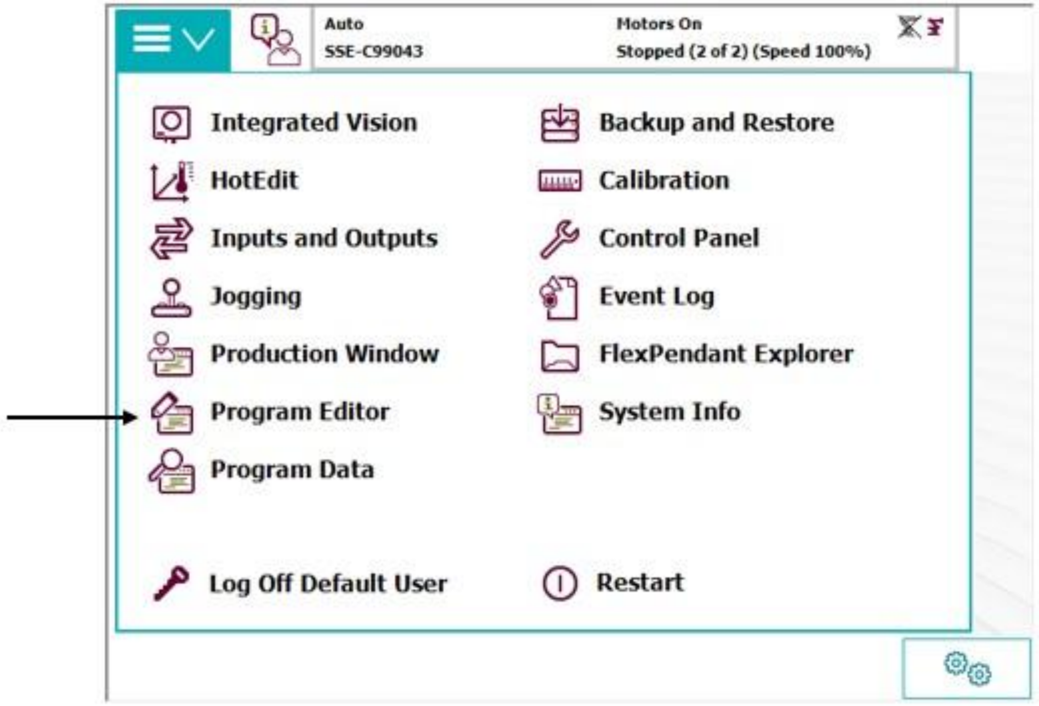
```
MoveJ Target_10, v100, fine, tool0\WObj:=wobj0;
```

Online Programming of the ABB YuMi using FlexPendant

3. Program Editor

The Program Editor screen is located in the Main Menu.

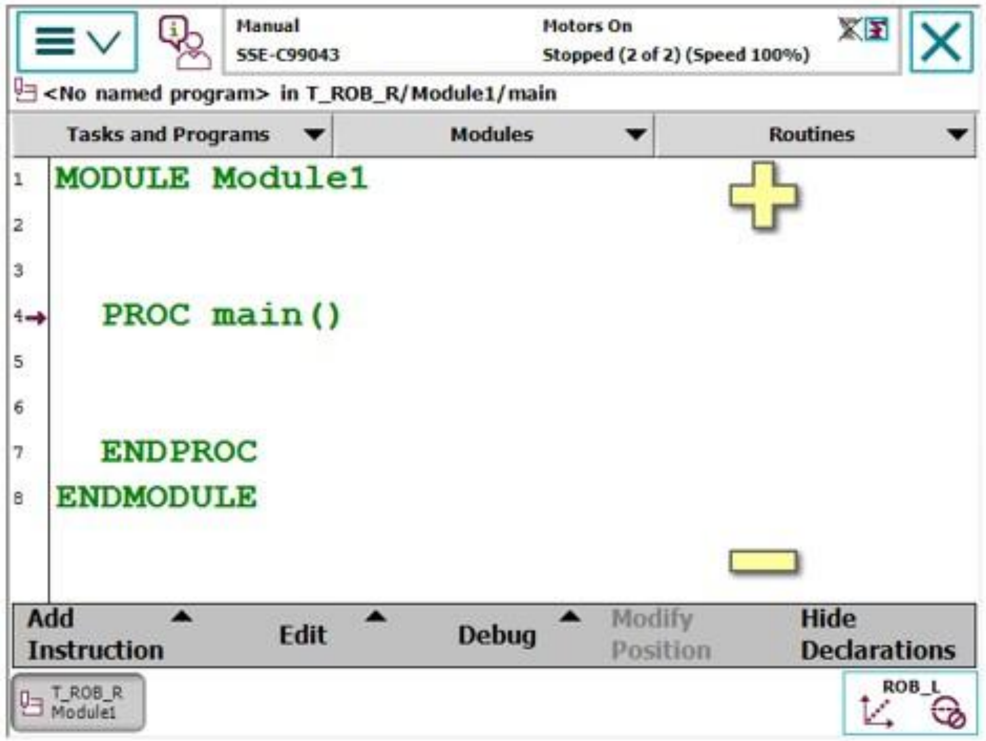
It is used to write and edit programs. There is a separate code for each arm. The programs can be saved into the system and loaded back at any time.



Online Programming of the ABB YuMi using FlexPendant

3. Program Editor

- Once you select any of the arms you will be able to see the programming window.
- Underneath the MODULE you can declare variables and underneath PROC main() (procedure main) you can write the code by using Add Instruction.
- Add Instruction is used to add any code/instruction in the program.
- Edit is used to edit any of the code lines or data set.
- Debug is used to move the pointer to specific lines of code and view the value of the selected data.



Manual
SSE-C99043
Motors On
Stopped (2 of 2) (Speed 100%)

<No named program> in T_ROB_R/Module1/main

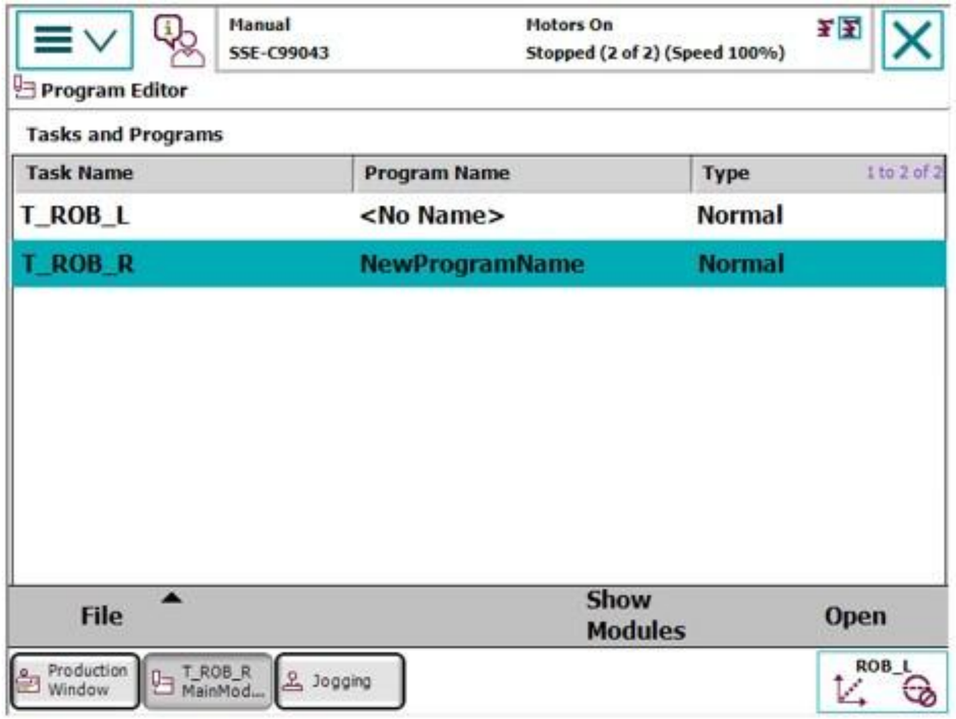
```
MODULE Module1


   PROC main()


   ENDPROC
ENDMODULE
```

THE NATIONAL ROBOTARIUM PEOPLE CENTRED :: INTELLIGENCE DRIVEN
HERIOT WATT UNIVERSITY
THE UNIVERSITY of EDINBURGH

Online Programming of the ABB YuMi using FlexPendant

3. Program Editor

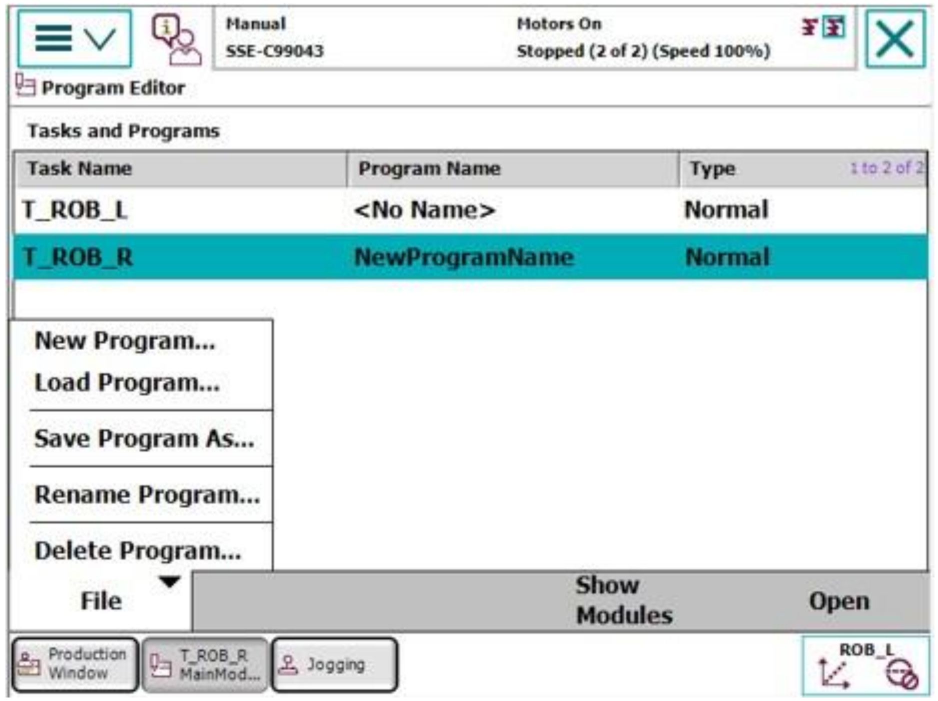
- Tapping Tasks and Programs will open the **Task Name** (right and left arm) and their **Program Name**.



Manual
SSE-C99043
Motors On
Stopped (2 of 2) (Speed 100%)
Program Editor
Tasks and Programs

Task Name	Program Name	Type
T_ROB_L	<No Name>	Normal
T_ROB_R	NewProgramName	Normal

- File is used to manage, load, and save programs.



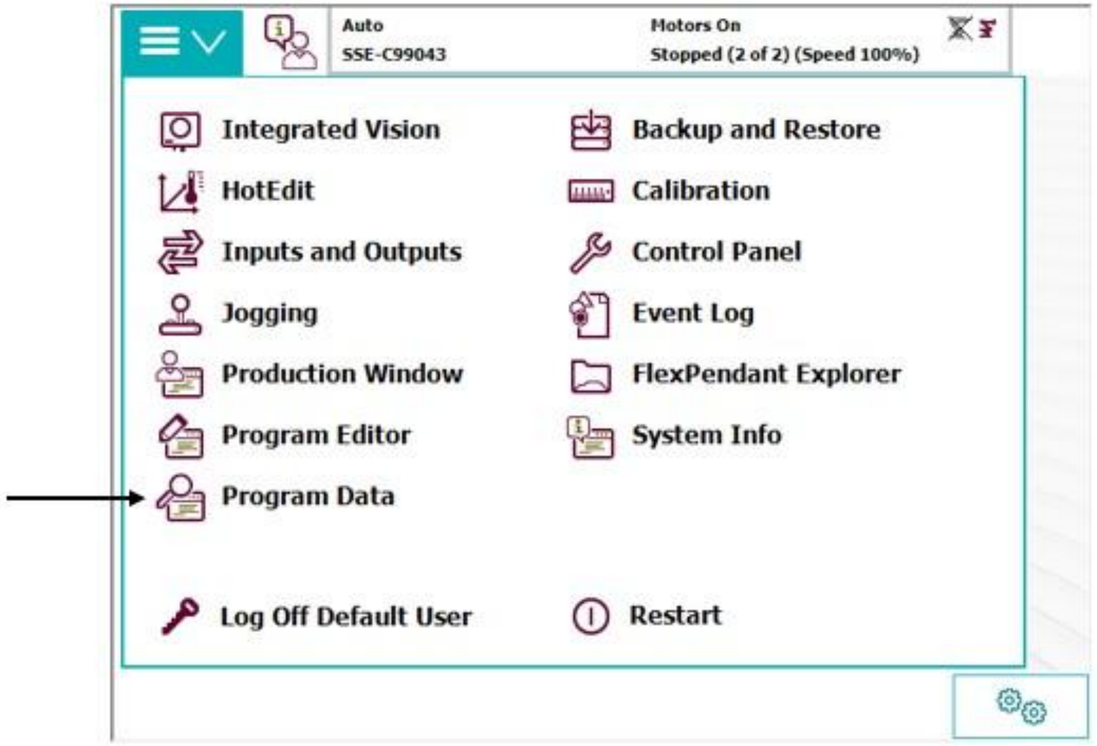
Manual
SSE-C99043
Motors On
Stopped (2 of 2) (Speed 100%)
Program Editor
Tasks and Programs

Task Name	Program Name	Type
T_ROB_L	<No Name>	Normal
T_ROB_R	NewProgramName	Normal

Online Programming of the ABB YuMi using FlexPendant

3. Program Data Screen

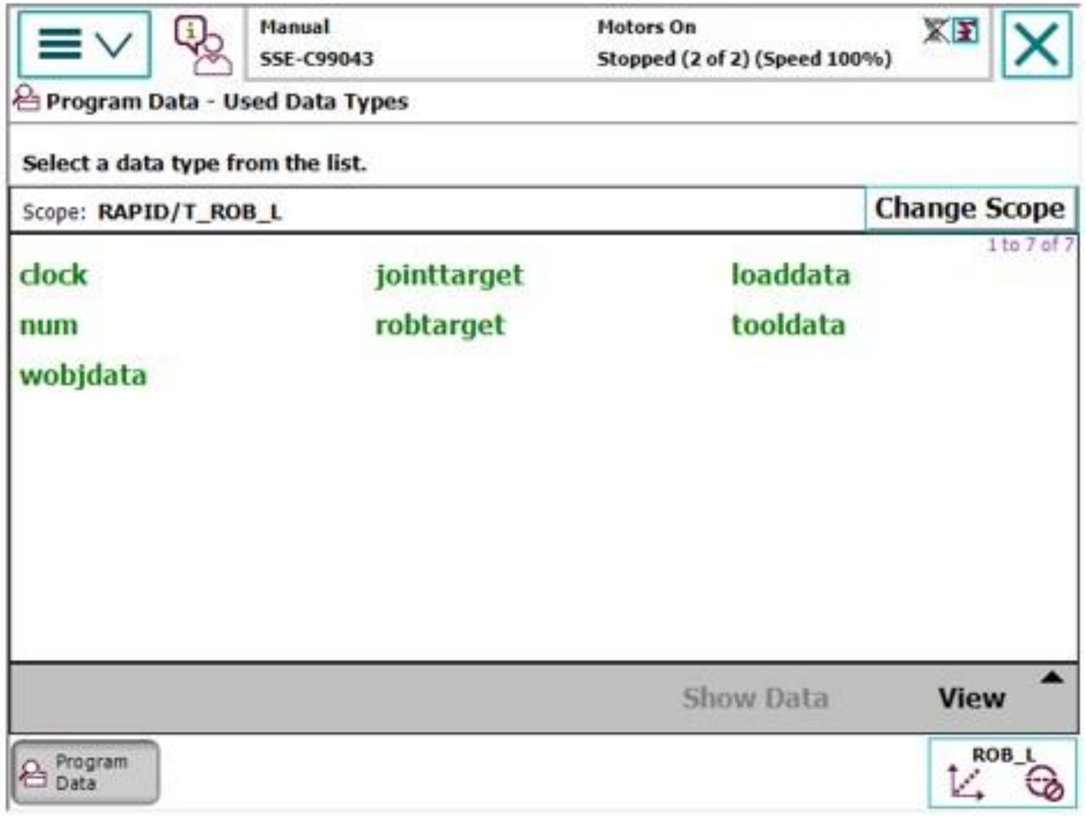
- The Program Data screen is located in the Main Menu.



Online Programming of the ABB YuMi using FlexPendant

3. Program Data Screen

- It contains functions for viewing and working with data types and instances. It can be used to create new instances for the selected data type by tapping **New** and the instances for the selected data type can be edited by tapping **Edit**.



Manual
SSE-C99043
Motors On
Stopped (2 of 2) (Speed 100%)

Program Data - Used Data Types

Select a data type from the list.

Scope: RAPID/T_ROB_L

1 to 7 of 7

clock
jointtarget
loaddata
num
robtarget
tooldata
wobjdata

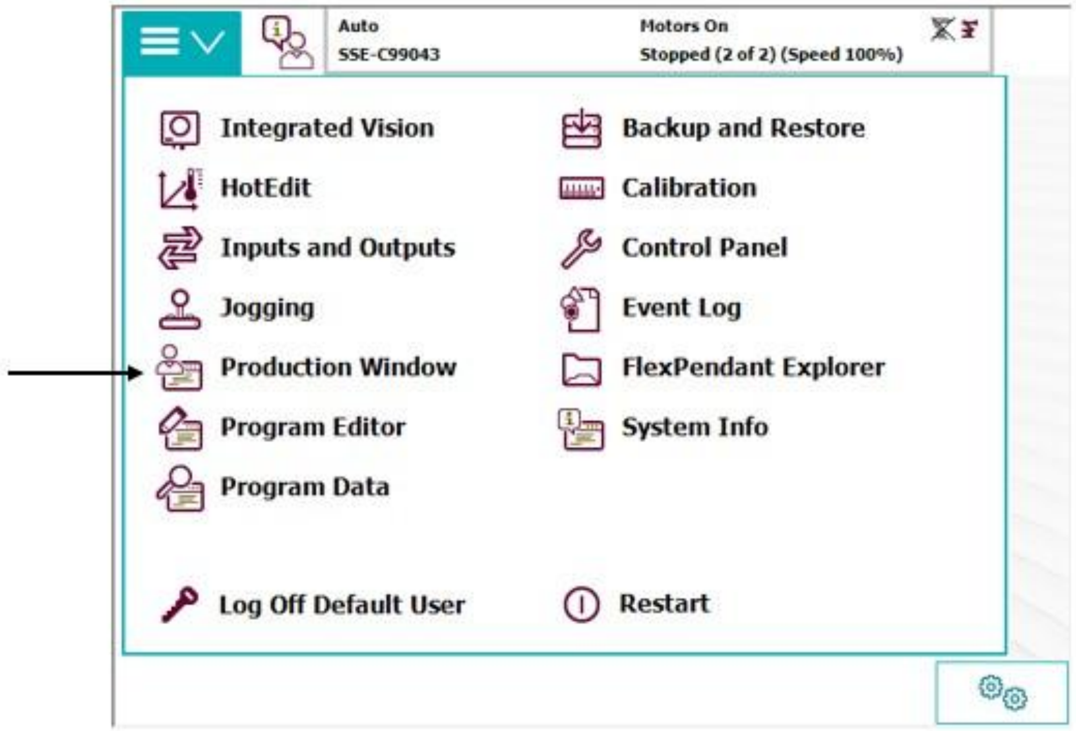
Online Programming of the ABB YuMi using FlexPendant

4. Production Window Screen

The Production Window screen is located on the Main Menu.

Production window is used to run the programs created.

- Load Program to load any programs saved in the system.
- PP to Main is used to move the code pointer to Proc Main and pull up the code that is in the program editor of each arm.

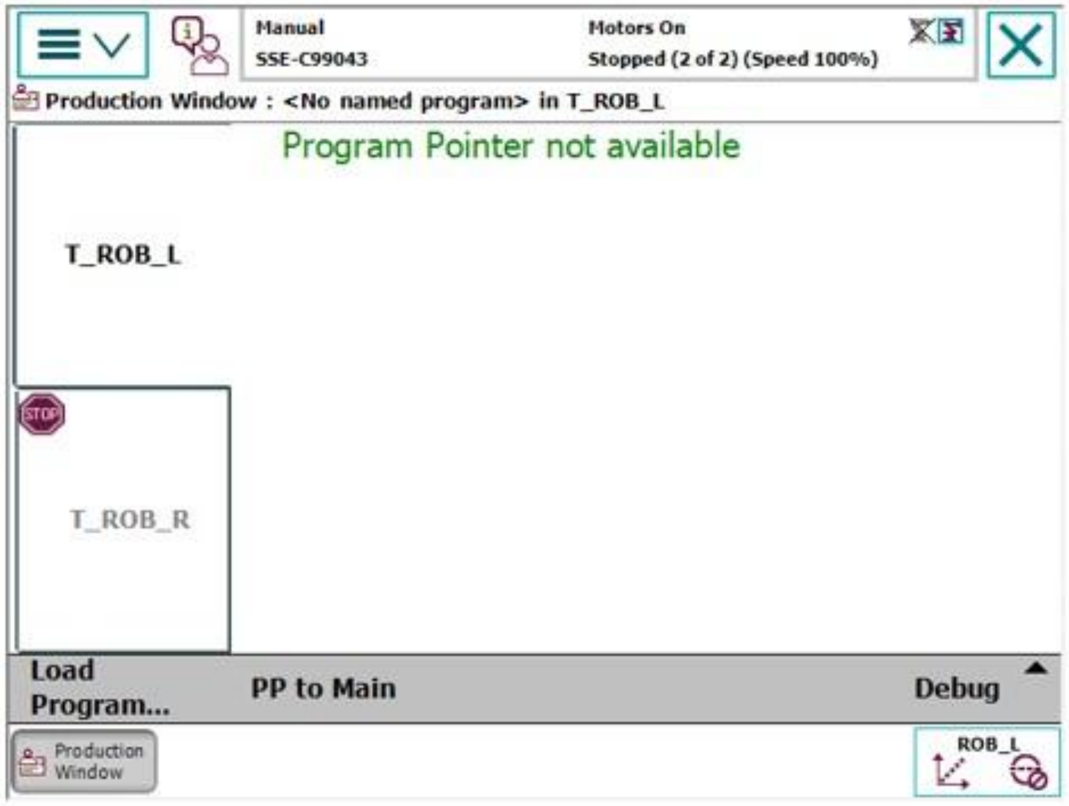


Online Programming of the ABB YuMi using FlexPendant

4. Production Window Screen

- After the code is loaded in for both arms, the code can be run by pressing the play button on the FlexPendant or run in steps by pressing the next step button for each step.

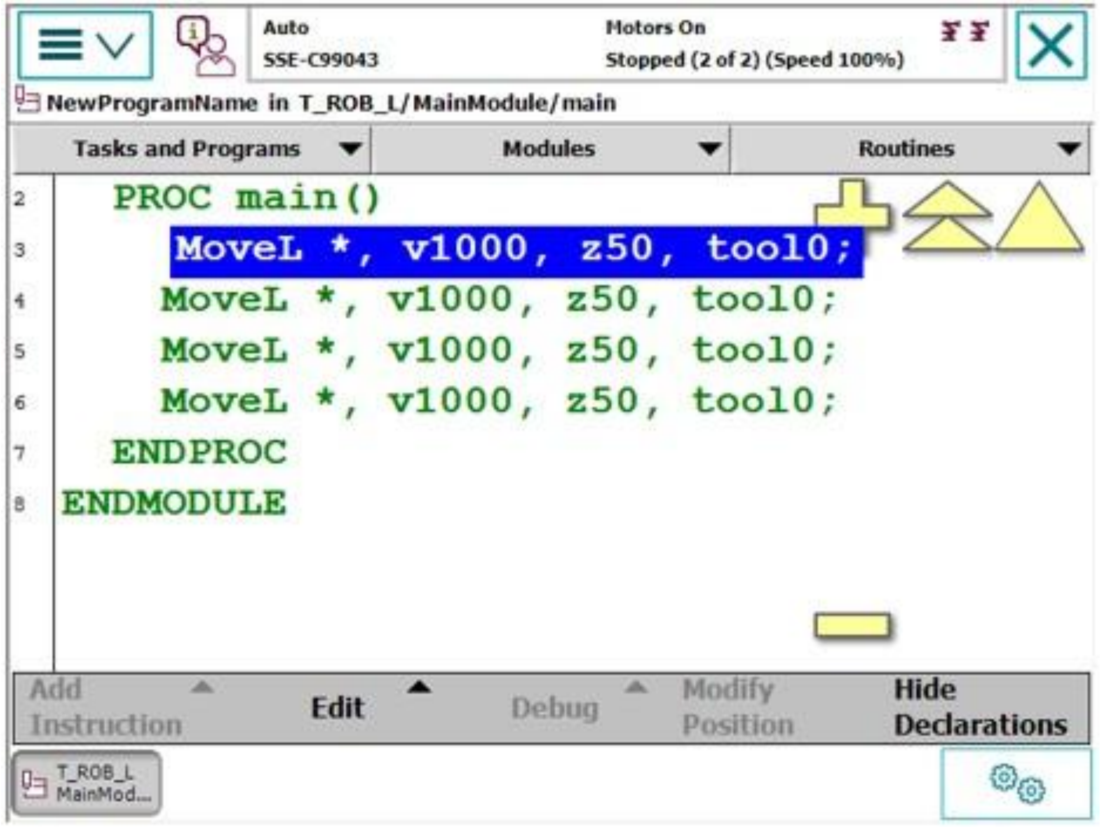
- The code can be stopped at any moment by pressing the stop button or the pushing in the E-stop at the top of the FlexPendant.


Manual
SSE-C99043
Motors On
Stopped (2 of 2) (Speed 100%)
Production Window : <No named program> in T_ROB_L
Program Pointer not available

Online Programming of the ABB YuMi using FlexPendant

Hands-On Exercise

1. Tap on **Add Instructions**.
2. Tap on **MoveL**.
3. Tap on **MoveL** again > tap **Below**.
4. Repeat the step above until you have 4 MoveL functions.
5. Tap on the *.
6. Tap on **Debug** > **View Value**.
7. Change the preset values from the following table for each of MoveL functions:



```
  PROC main()
    MoveL *, v1000, z50, tool0;
    MoveL *, v1000, z50, tool0;
    MoveL *, v1000, z50, tool0;
    MoveL *, v1000, z50, tool0;
  ENDPROC
ENDMODULE
```

Online Programming of the ABB YuMi using FlexPendant

Hands-On Exercise

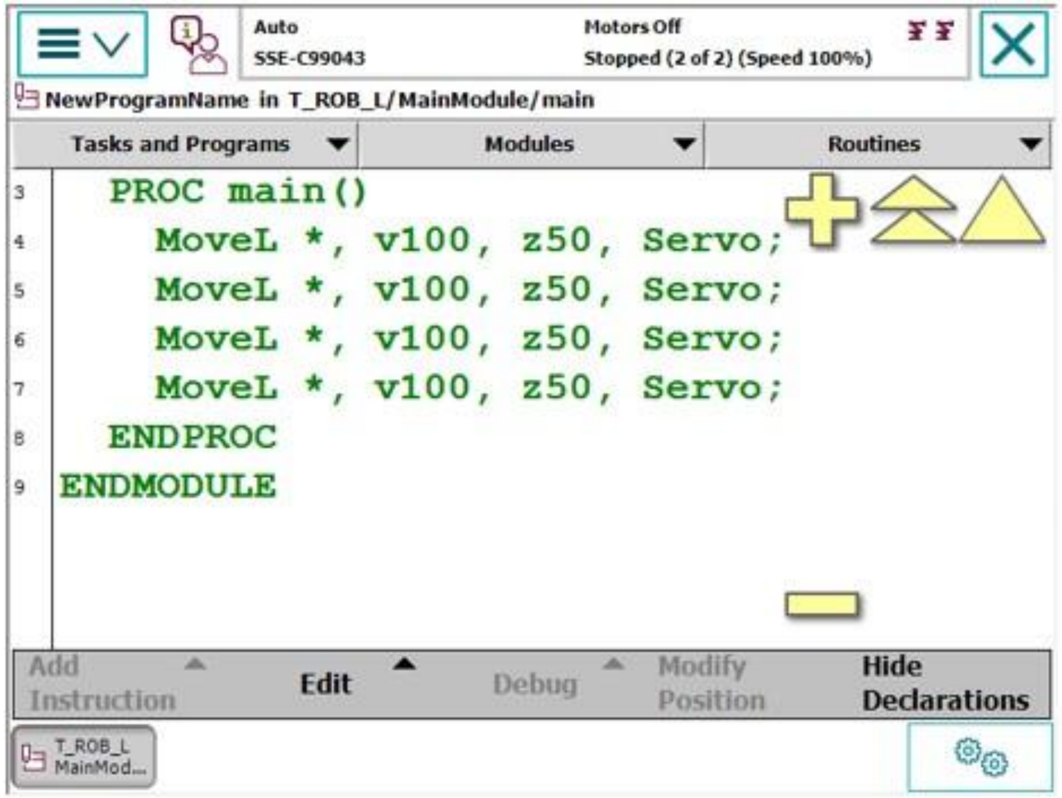
	x	y	z	q1	q2	q3	q4	cf1	cf4	cf6	cfx	eax_a
Move L 1	600	200	200	0.5	-0.5	0.5	-0.5	-1	1	0	5	120
Move L 2	600	200	150	0.5	-0.5	0.5	-0.5	0	0	0	0	120
Move L 3	600	100	150	0.5	-0.5	0.5	-0.5	0	0	0	0	120
Move L 4	600	100	200	0.5	-0.5	0.5	-0.5	0	0	0	0	120

Online Programming of the ABB YuMi using FlexPendant

Hands-On Exercise

8. Tap on the **tool0** > **Edit** > **Change Selected** > select **Servo** for all the MoveL functions.

9. Tap on the **v1000** > **Edit** > **Change Selected** > select **v100** for all the MoveL functions.



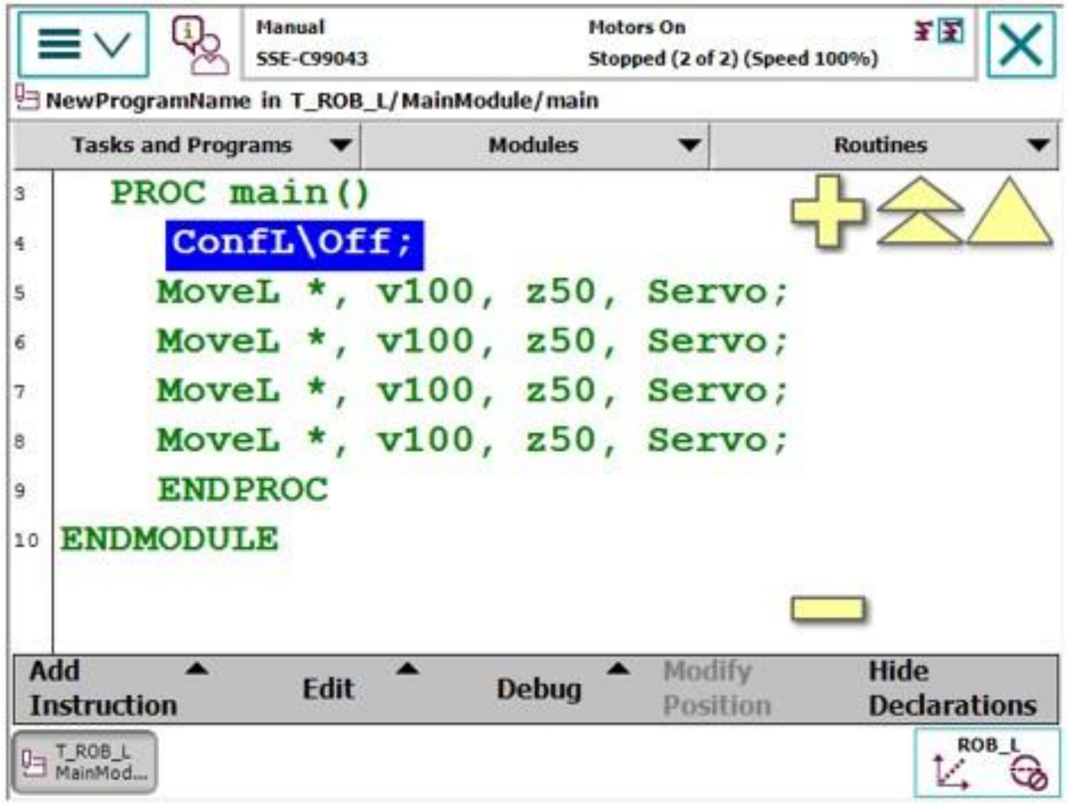
```
  PROC main()
    MoveL *, v100, z50, Servo;
    MoveL *, v100, z50, Servo;
    MoveL *, v100, z50, Servo;
    MoveL *, v100, z50, Servo;
  ENDPROC
ENDMODULE
```

Online Programming of the ABB YuMi using FlexPendant

Hands-On Exercise

10. Tap on the first MoveL and tap **Add Instruction** > **Common** > **Settings** > select **ConfL** > **Above**.

11. Tap on the **ConfL\On** > **Edit** > **Change Selected** > **Optional Argument** > tap on the line > tap **\Off** > **Use** > **Close** > **Close** > **Ok**.

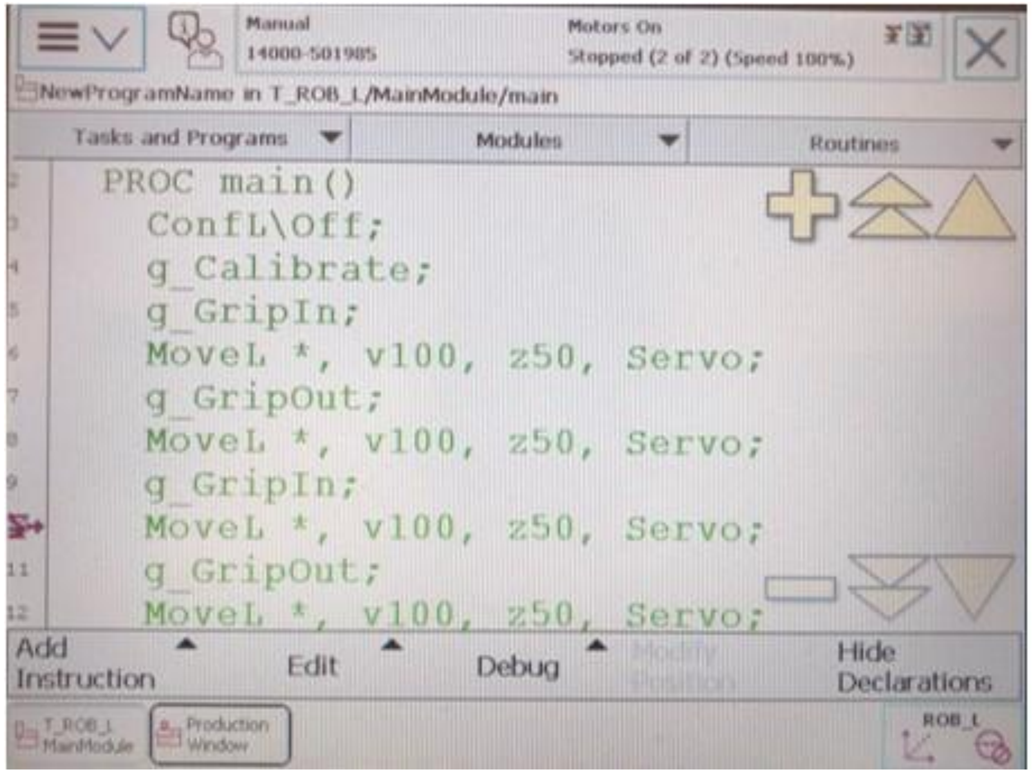


```
  PROC main()
    ConfL\Off;
    MoveL *, v100, z50, Servo;
    MoveL *, v100, z50, Servo;
    MoveL *, v100, z50, Servo;
    MoveL *, v100, z50, Servo;
    ENDPROC
ENDMODULE
```

Online Programming of the ABB YuMi using FlexPendant

Hands-On Exercise

1. Tap on the **ConfL\Off** function.
2. Tap **Add Instruction** > **Common** > **Smart Gripper** > **g_Calibrate** > tap **Below**.
3. Tap on the **g_Calibrate** function.
4. Tap **Add Instruction** > **g_GripIn** > tap **Below**.
5. Tap on the First MoveL function.
6. Tap **Add Instruction** > **g_GripOut** > tap **Below**.
7. Tap on the Second MoveL function.
8. Tap **Add Instruction** > **g_GripIn** > tap **Below**.
9. Tap on the Third MoveL function.
10. Tap **Add Instruction** > **g_GripOut** > tap **Below.**



```
PROC main()
  ConfL\Off;
  g_Calibrate;
  g_GripIn;
  MoveL *, v100, z50, Servo;
  g_GripOut;
  MoveL *, v100, z50, Servo;
  g_GripIn;
  MoveL *, v100, z50, Servo;
  g_GripOut;
  MoveL *, v100, z50, Servo;
```

Offline Programming of the ABB YuMi using RobotStudio

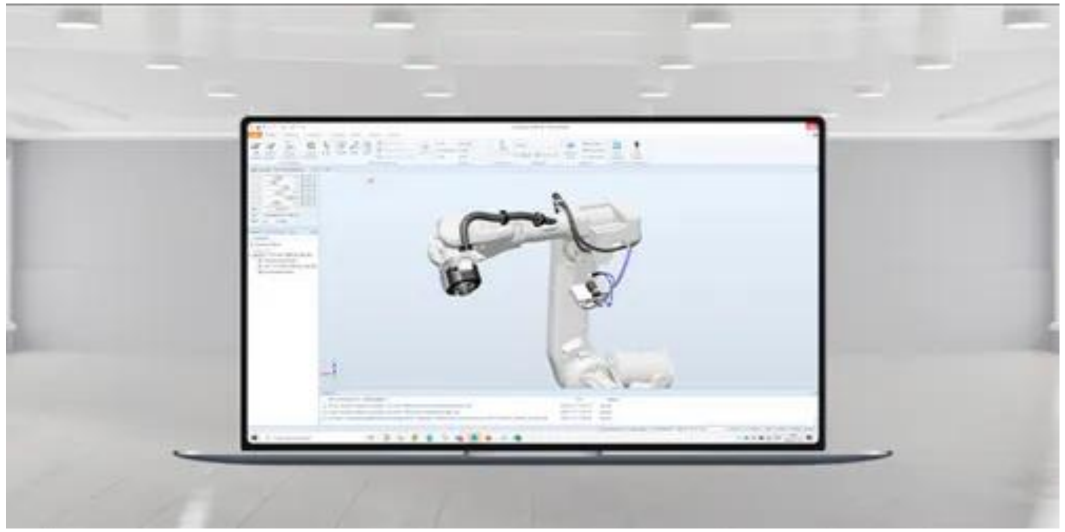
Topics Covered:

1. Introduction to RobotStudio
2. Building station and Creating robot controller
3. Programming robots in the 3D environment
4. Simulating programs
5. Deploying and Sharing

Offline Programming of the ABB YuMi using RobotStudio

1. Introduction to RobotStudio

- RobotStudio is an engineering tool for configuring and programming ABB robots.
- This application is used for modeling, offline programming, and simulation of robot cells.
- Its advanced modeling and simulation features help in visualizing multi-robot control, safety features, 3D vision, and remote robot supervision.



Offline Programming of the ABB YuMi using RobotStudio

2. Building a station and Creating a robot controller

Starting a station

When RobotStudio opens there are three type of stations that can be opened:

- **Empty station:** saves station files.
- **Solution with empty station:** stores station and solution files.
- **Solution with station and virtual controller:** stores station, solution, and controller files.

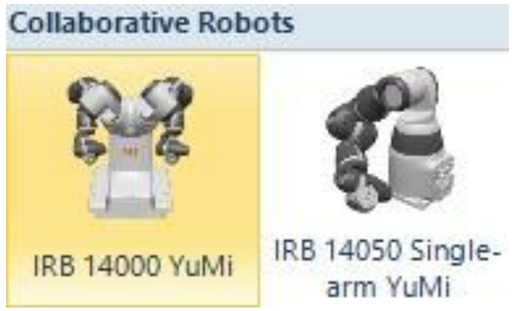
Offline Programming of the ABB YuMi using RobotStudio

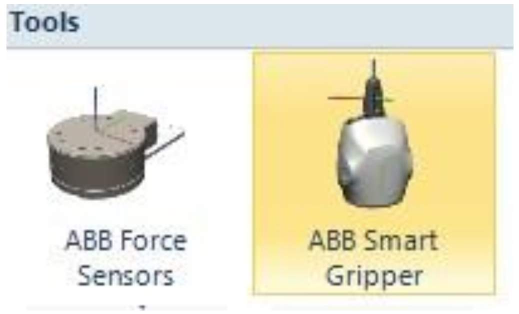
2. Building a station and Creating a robot controller

Virtual YuMi and Controller

- Adding the virtual YuMi in RobotStudio can be done by clicking the Home tab > ABB Library > IRB 14000 Yumi. The YuMi will then be displayed in the graphics window.

- Adding the virtual grippers can be done by clicking the Home tab > Import Library > Equipment > tools: ABB Smart Gripper > variant: servo, fingers, camera, suction > OK.



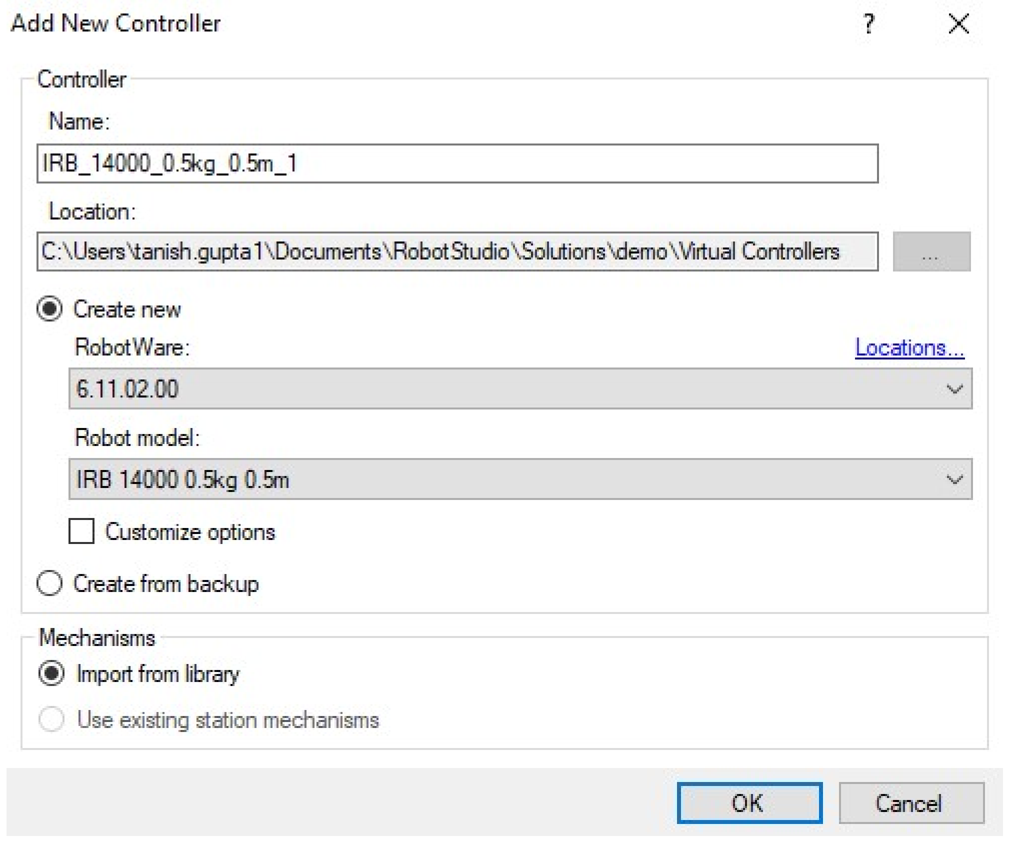


Offline Programming of the ABB YuMi using RobotStudio

2. Building a station and Creating a robot controller

Virtual YuMi and Controller

- This will add a smart gripper in the graphics window. To attach the gripper, drag the "Smart_Gripper_Servo_Fingers" to the "IRB14000_0.5_0.5_R". Add another ABB smart Gripper and drag it to the "IRB 14000_0.5_0.5_L".
- To access many of the features for the virtual YuMi you will need to add a virtual controller. To do this, click home tab > virtual controller > new controller > change the robot model to IRB 14000 > select use existing station mechanisms > OK.


Add New Controller

Controller

Name: IRB_14000_0.5kg_0.5m_1

Location: C:\Users\tanish.gupta1\Documents\RobotStudio\Solutions\demo\Virtual Controllers

(●) Create new

RobotWare: 6.11.02.00

Robot model: IRB 14000 0.5kg 0.5m

[ ] Customize options

( ) Create from backup

Mechanisms

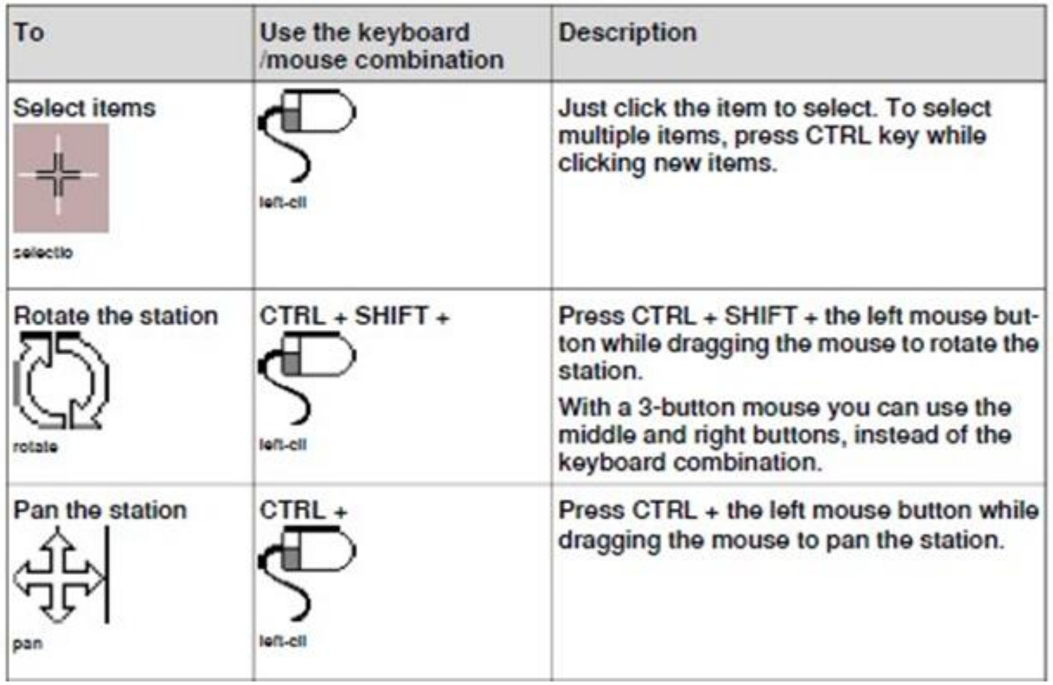
(●) Import from library

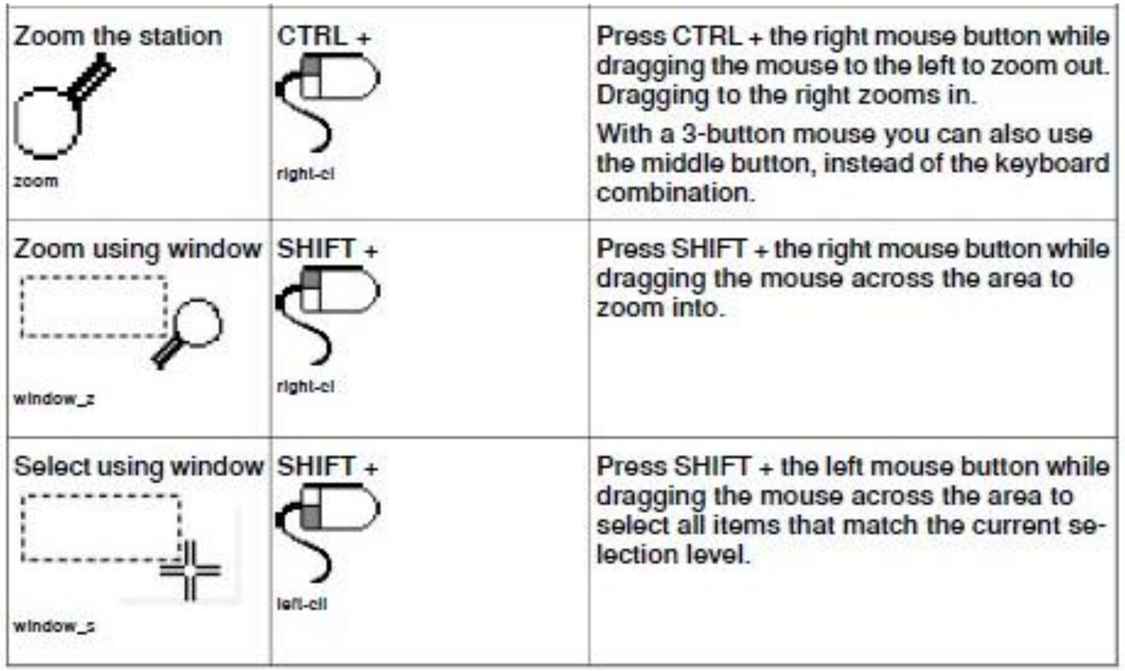
( ) Use existing station mechanisms

Offline Programming of the ABB YuMi using RobotStudio

2. Building a station and Creating a robot controller

<u>Navigating in the graphics window</u>





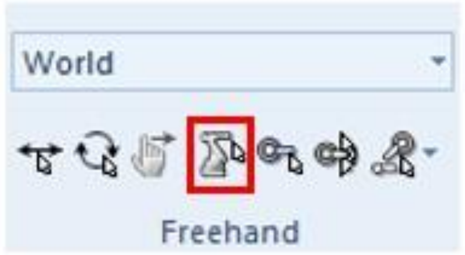
To	Use the keyboard /mouse combination	Description
Select items selectio	left-cli	Just click the item to select. To select multiple items, press CTRL key while clicking new items.
Rotate the station rotate	CTRL + SHIFT + left-cli	Press CTRL + SHIFT + the left mouse button while dragging the mouse to rotate the station. With a 3-button mouse you can use the middle and right buttons, instead of the keyboard combination.
Pan the station pan	CTRL + left-cli	Press CTRL + the left mouse button while dragging the mouse to pan the station.
Zoom the station zoom	CTRL + right-cl	Press CTRL + the right mouse button while dragging the mouse to the left to zoom out. Dragging to the right zooms in. With a 3-button mouse you can also use the middle button, instead of the keyboard combination.
Zoom using window window_z	SHIFT + right-cl	Press SHIFT + the right mouse button while dragging the mouse across the area to zoom into.
Select using window window_s	SHIFT + left-cli	Press SHIFT + the left mouse button while dragging the mouse across the area to select all items that match the current selection level.

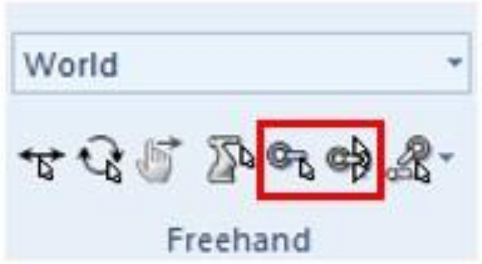
Offline Programming of the ABB YuMi using RobotStudio

2. Building a station and Creating a robot controller

<u>Moving the Virtual YuMi</u>

- To move the YuMi click on the Home tab, in the freehand section (top bar) click the jog button and click on any joint of the virtual YuMi to jog it.
- To move an arm linearly or reorient click the jog linear or jog reorient.





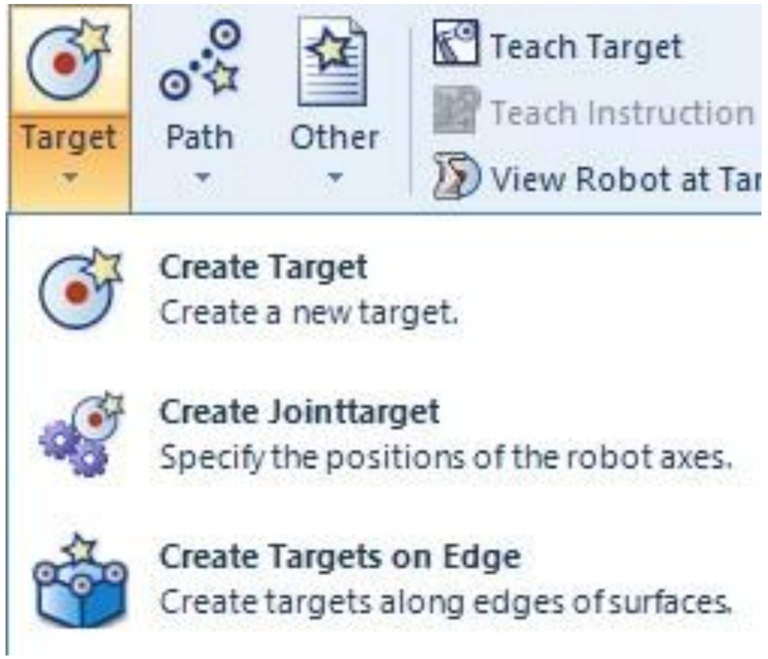
Offline Programming of the ABB YuMi using RobotStudio

3. Programming Robots in 3D Environment

Targets

There are two types of targets: position targets and joint targets.

- **Create Target** creates position targets which store the x, y, z and the orientation of the target for MoveL or MoveJ usage.
- **Create Jointtarget** creates joint targets which store the angle for each joint for using MoveAbsJ function.
- **Create Targets on Edge** automatically creates targets (which can either be position targets or joint targets) around an edge of a 3D object.

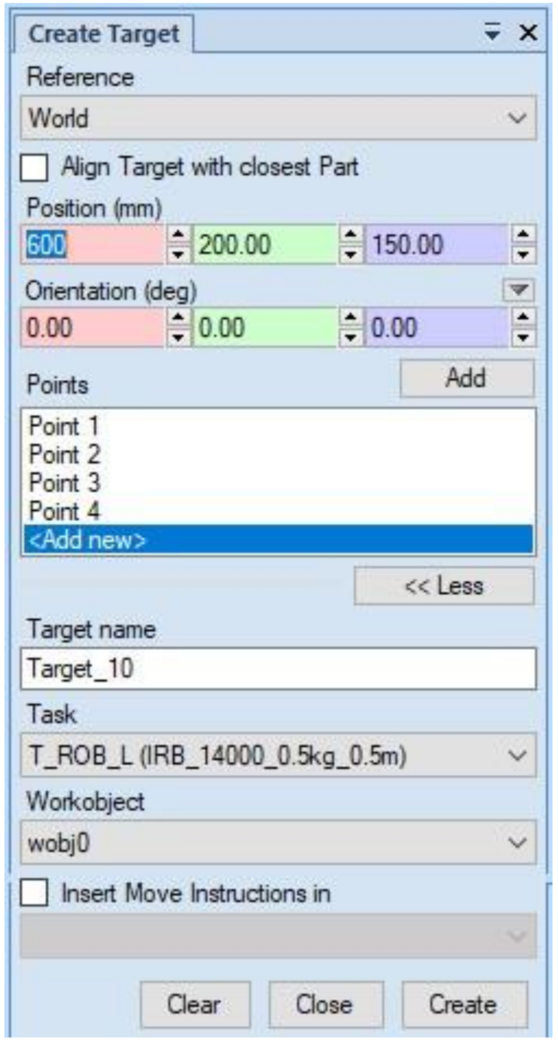


Offline Programming of the ABB YuMi using RobotStudio

3. Programming Robots in 3D Environment

Targets

Position targets can be made in the home tab by clicking Target > Create Target and placing them on the virtual area. The position and orientation of each target can be selected. The Arm and the work object that will be associated with the targets can be selected by clicking more.


Create Target

Reference: World

☐ Align Target with closest Part

Position (mm): 600 | 200.00 | 150.00

Orientation (deg): 0.00 | 0.00 | 0.00

Points: Point 1, Point 2, Point 3, Point 4, <Add new>

Target name: Target_10

Task: T_ROB_L (IRB_14000_0.5kg_0.5m)

Workobject: wobj0

☐ Insert Move Instructions in

THE NATIONAL ROBOTARIUM
PEOPLE CENTRED :: INTELLIGENCE DRIVEN

HERIOT WATT UNIVERSITY

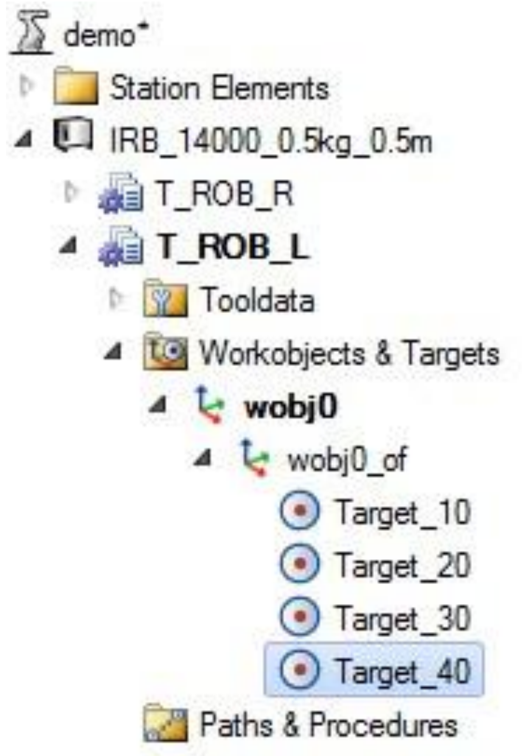
THE UNIVERSITY of EDINBURGH

Offline Programming of the ABB YuMi using RobotStudio

3. Programming Robots in 3D Environment

Targets

- The targets will show up on the left side under the "Workobjects & Targets".
- The position and orientation can be edited at any time by right clicking on the target > modify position > set target. The reference point can also be selected.
- The orientation of multiple targets can be aligned with one target being the reference. This helps aligning multiple targets without changing them individually. To do this change the orientation of one of the targets, shift click to select the other targets all at once > right click > modify target > align target orientation > select the first target as reference > align the axis that need to be aligned.



THE NATIONAL ROBOTARIUM PEOPLE CENTRED :: INTELLIGENCE DRIVEN

HERIOT WATT UNIVERSITY

THE UNIVERSITY of EDINBURGH

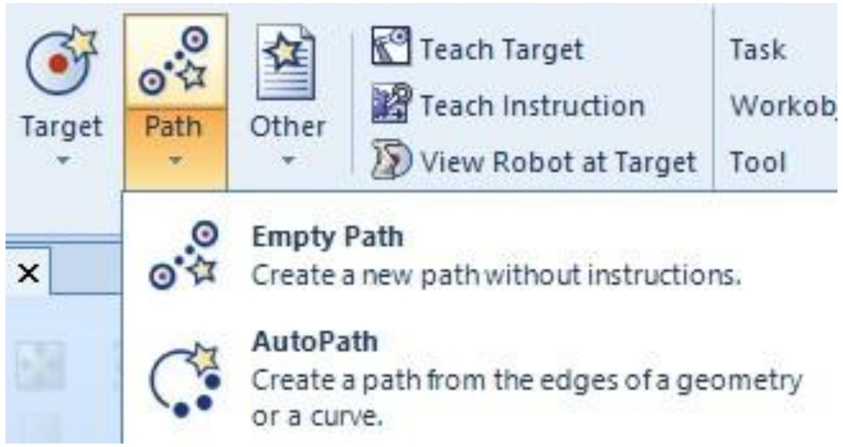
Offline Programming of the ABB YuMi using RobotStudio

3. Programming Robots in 3D Environment

<u>**Paths**</u>

There are two types of paths: empty path and auto path.

- **Empty path** creates a path by assigning targets to it.
- **Auto path** creates an automatic path around an edge of a 3D object.
- To create an empty path, drag the targets made to the path. The targets will be assigned MoveJ or MoveL. The move target functions will show up on the left side under the "Paths & Procedure".

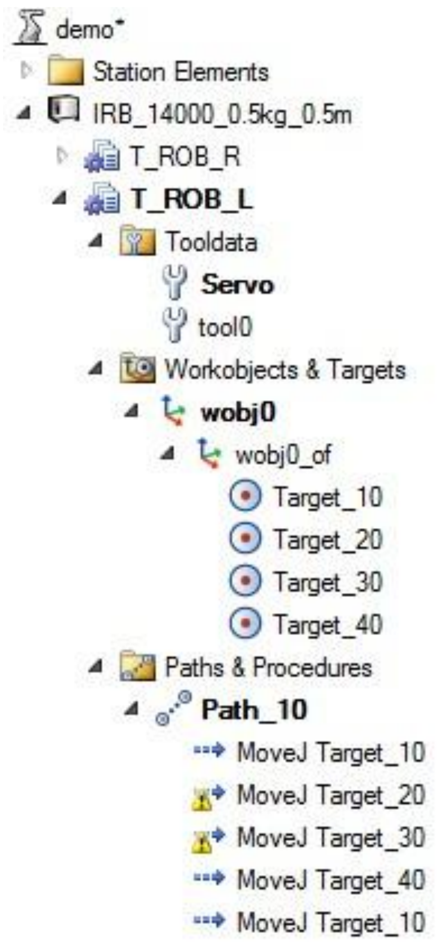


Offline Programming of the ABB YuMi using RobotStudio

3. Programming Robots in 3D Environment

Paths

- To create auto paths, draw a 3D object or import it into RobotStudio in the modeling tab and position the object in the range of the YuMi. Click paths > auto path > select the edges you want the path to be along > select the parameters > create, the new path will be created along with new targets.



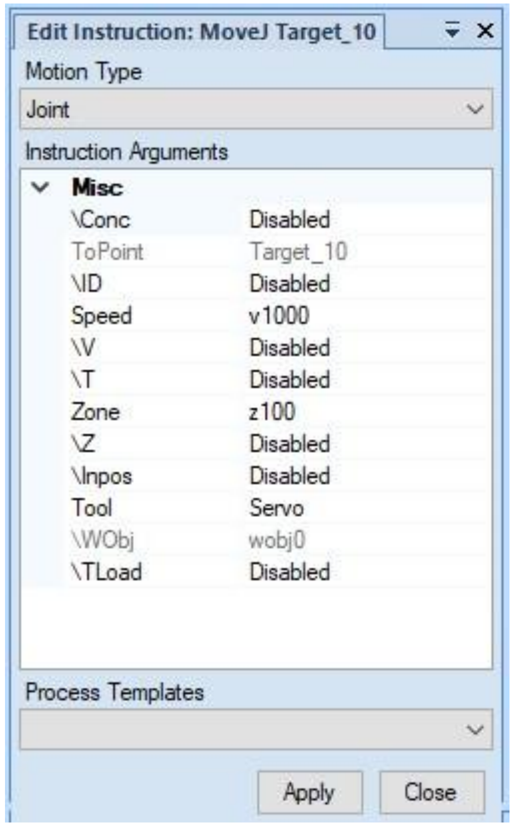
```
demo*
  Station Elements
  IRB_14000_0.5kg_0.5m
    T_ROB_R
    T_ROB_L
      Tooldata
        Servo
        tool0
      Workobjects & Targets
        wobj0
          wobj0_of
            Target_10
            Target_20
            Target_30
            Target_40
      Paths & Procedures
        Path_10
          MoveJ Target_10
          MoveJ Target_20
          MoveJ Target_30
          MoveJ Target_40
          MoveJ Target_10
```

Offline Programming of the ABB YuMi using RobotStudio

3. Programming Robots in 3D Environment

Paths

- The parameters of any of the move targets can be changed by right clicking on them > edit instruction. The motion type can be change between Move J and MoveL, speed, zone, and few other things.


Edit Instruction: MoveJ Target_10

Motion Type: Joint

Instruction Arguments

Misc

\Conc	Disabled
ToPoint	Target_10
\ID	Disabled
Speed	v1000
\V	Disabled
\T	Disabled
Zone	z100
\Z	Disabled
\Inpos	Disabled
Tool	Servo
\WObj	wobj0
\TLoad	Disabled

Process Templates

Offline Programming of the ABB YuMi using RobotStudio

3. Programming Robots in 3D Environment

Paths

After a path has been created, the YuMi can be moved along the path by right clicking on the path > Move Along Path.

The YuMi might show errors which can be avoided by doing the following:

- Adding ConfL Off and ConfJ Off instruction: to do this, right click on the path > Insert Action Instruction > dropdown by Instruction Templates > select ConfL Off > Create, do the same for ConfJ Off.
    - ✓ ConfL Off turns the configuration off for moveL motions and ConfJ Off turns the configuration off for moveJ motions. This helps the Yumi decide its own configurations and avoid errors.

Offline Programming of the ABB YuMi using RobotStudio

3. Programming Robots in 3D Environment

Paths

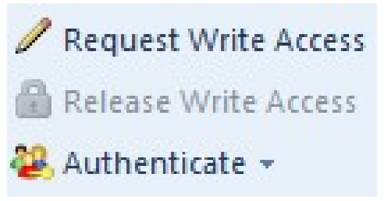
- Changing configuration of the first target, to do this right click on the path > Auto Configuration > All Move Instructions > select a configuration in the dropdown > Apply. Try until one of the configuration works.
- Make additional targets. Making targets a bit away from the path can help the YuMi approach the path easier and with the correct configuration. Adding multiple targets on the path can make it easier for the YuMi to move along the path.
- If the robot moves along the path without giving an error, it will work fine in code.
- The targets and paths can be synchronized to the code by pressing Home tab > Synchronize > Synchronize to RAPID > select the paths and targets box. The code should now be in the RAPID code of the virtual YuMi.
- To run the paths, write the path name under "PROC main()"
- To run the code on the actual YuMi, the code can be copy pasted into the RAPID for the connected YuMi.

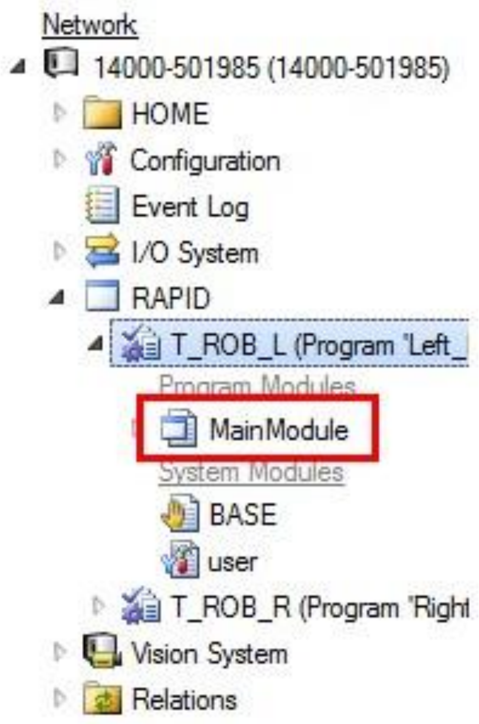
Offline Programming of the ABB YuMi using RobotStudio

3. Programming Robots in 3D Environment

Coding on connected YuMi

- Before editing or writing the code you will need write access, which can be done by clicking Request Write Access on the top left in the tab. The message will pop up on the FlexPendant asking for granting or revoking write access. Granting access will allow you to edit or write the code.
- You can find the coding tab for the ABB YuMi by clicking on the RAPID tab > dropdown arrow beside the 14000-501985 (ABB Yumi) > dropdown arrow beside rapid > dropdown arrow beside either arm > MainModule. This will open up a tab of the rapid code for that arm, the program that is loaded currently will show up.



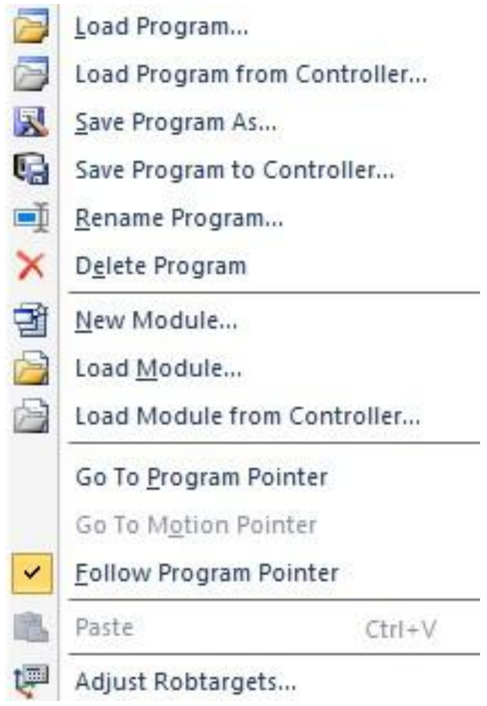


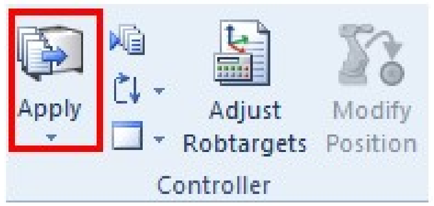
Offline Programming of the ABB YuMi using RobotStudio

3. Programming Robots in 3D Environment

Coding on connected YuMi

- Programs can be saved or loaded onto the ABB Yumi by right clicking the main module and clicking **load program to controller** or Save Program to Controller. This is not necessary for just running programs.
- Just like coding on the FlexPendant; any constant variables are written underneath the module, and any other codes are written under proc main.
- Before using the code on the FlexPendant click to apply changes and then revoke access on the FlexPendant. The code will now be changed in the program editor on the FlexPendant.



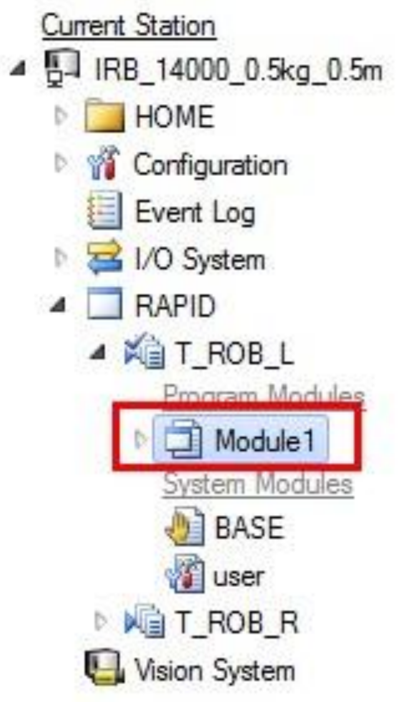


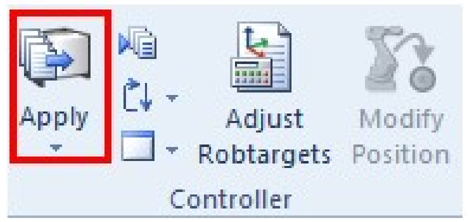
Offline Programming of the ABB YuMi using RobotStudio

3. Programming Robots in 3D Environment

Coding on Virtual YuMi

- Access the coding tab for the virtual Yumi by clicking on the RAPID tab > dropdown arrow beside the "IRB_14000_0.5kg_0.5m (Virtual Yumi) > dropdown arrow beside rapid > dropdown arrow beside either arm > Module 1. This will open up a tab of the rapid code for that arm. The same can be done for the other arm.
- Click Apply after done.





Thank You

Looking to Collaborate, let's connect

The National Robotarium

Rahul R. Ramachandran

THE NATIONAL ROBOTARIUM
PEOPLE CENTRED :: INTELLIGENCE DRIVEN

HERIOT WATT UNIVERSITY

THE UNIVERSITY of EDINBURGH