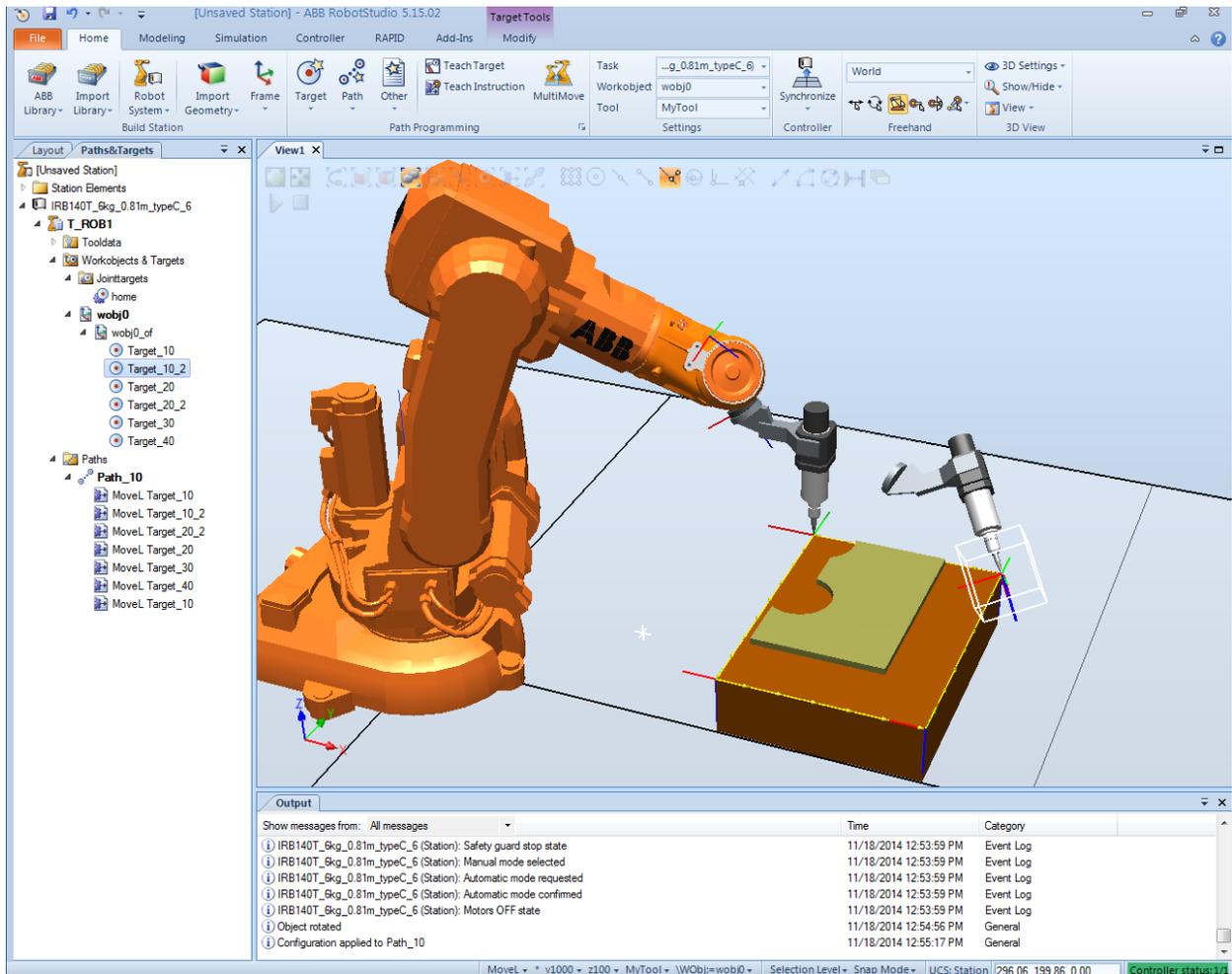


## RBE/ME 4815 In-Class Assignment #1

In this assignment you will use RobotStudio to create a 'station', populate it with various items, and then generate a program to control the robot to accomplish certain tasks.

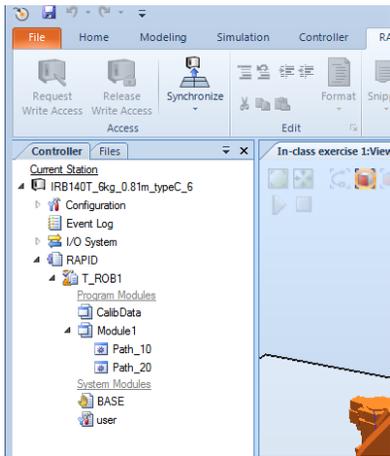
1. Create a station with a controller. Pick any robot to use in the station. [ The 'IRB140T 6kg 0.81m type C' robot is what I have been using in class. You may be prompted about multiple libraries being available. Any library is fine for the purposes of this assignment. ]
2. Go to the Home tab. Import the tool called 'myTool'. Attach it to the end of the robot arm.
3. Import the 'curve thing'. You will need to move it closer to the robot and orient it so that you can accomplish the task. Consider rotating it 90° so it presents the wider side of the base to the robot.
4. Create the 'home' joint target. You could set this up such that all of the joints are at their zero values. Then create a joint target using those values. Note that this is actually a singular condition, so you alternatively might want to set joint 5 to be some non-zero value.
5. Create a path around the upper surface of the base of the curve thing. Use the AutoPath tool to help with that.
  - a. Drill down so you can see the targets that were created. Using one of the targets, turn on 'View Tool At Target'. Adjust the orientation of the tool so that it is perpendicular to the surface and facing in the correct direction. Propagate that tool orientation to the other targets in the path. [ Remember that you can Copy/Paste orientations from one target to another. ]
  - b. Select a configuration for the first target and then apply that configuration to the other targets (you may use the AutoConfigure function to do that). [ The first configuration in the list of options is a good choice. ]
  - c. Depending on how you chose the edges, the AutoPath function may create an extra target (a duplicate of the first target in the path). If this happens, delete the duplicated target; then add an additional MoveL instruction to the end of the path so that the path is closed.
  - d. Test your path by right-clicking on the path and selecting 'Move Along Path'.
6. Part of the path around the upper surface of the base goes along an edge that is coincident with the raised surface. To avoid a possible tool collision in that area, we want to tilt the tool away while traversing that edge. To do this we will create two additional targets. These targets should be coincident with the existing corner targets, but rotated out 45°. [ You may rotate them around multiple axes if you wish. ] Create these targets and add them to the existing path. Add MoveL instructions at the appropriate points of the path to cause the tool to be rotated once it reaches the corner. It will then stay rotated out as the tool traverses that edge – then become perpendicular to the surface again after it reaches the end of the edge.

- The following figure shows how the tool could be rotated at the start of the far edge so that a possible collision with the raised surface is avoided. The robot is shown with the tool positioned at the start of the path. The goal is to trace a path counter-clockwise around the edges. The 'detached' tool shows how it should be oriented for traversing the last (farthest) edge.

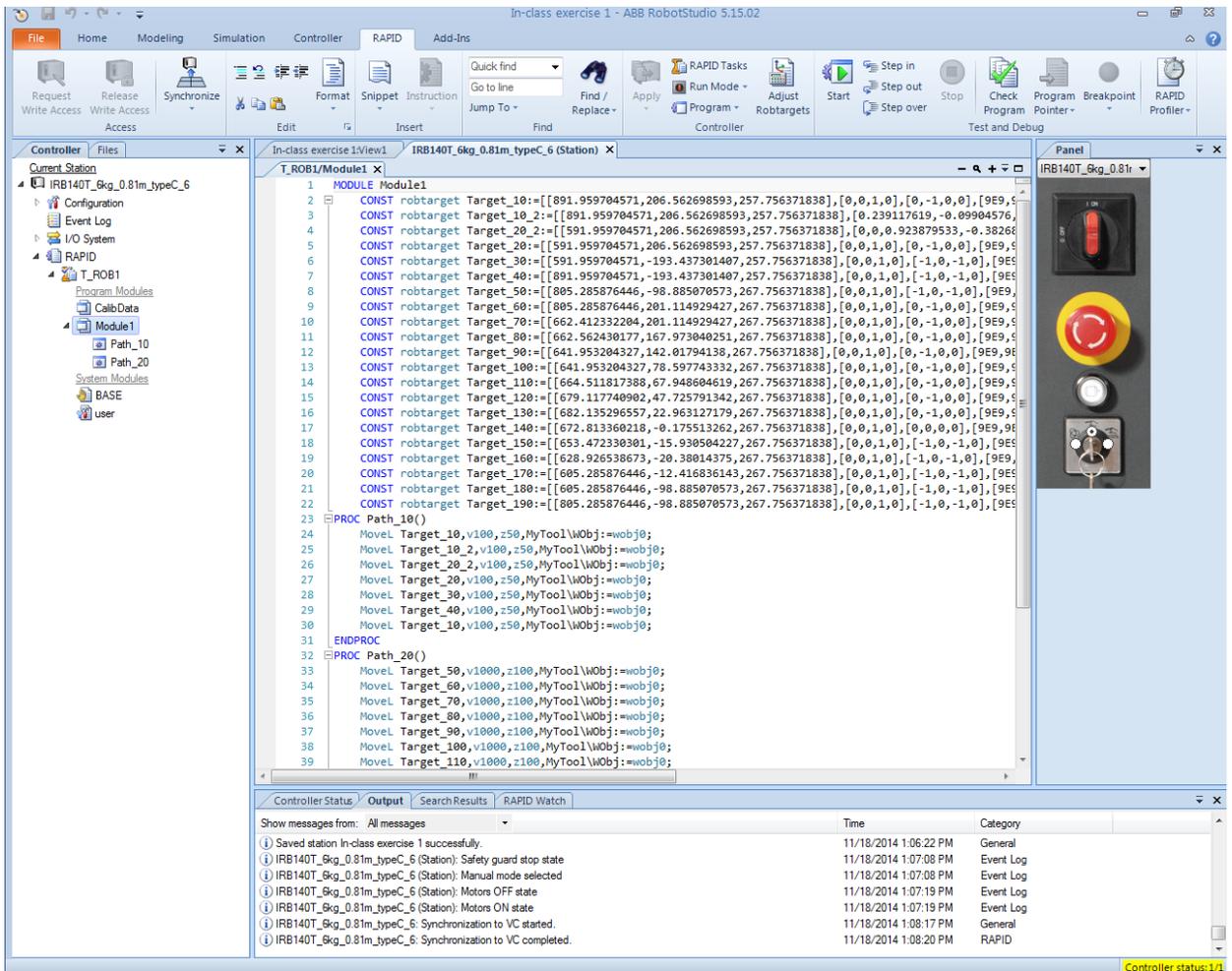


- Create a path around the raised and curved surface of the curve thing. Repeat the process steps (a – c) in step 5 above to ready the path for use.
- Go to the Offline Controller tab. Bring up the Control Panel. If it is not already in 'Manual Reduced Speed' mode then set it to that (i.e., click on the white dot at the 12 o'clock position above the key). Notice that the motor status button will start flashing. This means that the motors are off. Turn the motors on (press the flashing white motor status button so that it is illuminated continuously). You should see the motor state change to ON in the Output status window.
- Now go to the RAPID tab and then click the 'Synchronize to VC' button. VC is short for Virtual Controller. Select everything in the window that pops up and click OK.

11. Drill down in the leftmost pane on the RAPID tab. You should see that a RAPID code source file named 'Module1' has been created. It should include the two paths that were created.



12. Double click on Module1. This will open up a text editor with the RAPID code that was created.

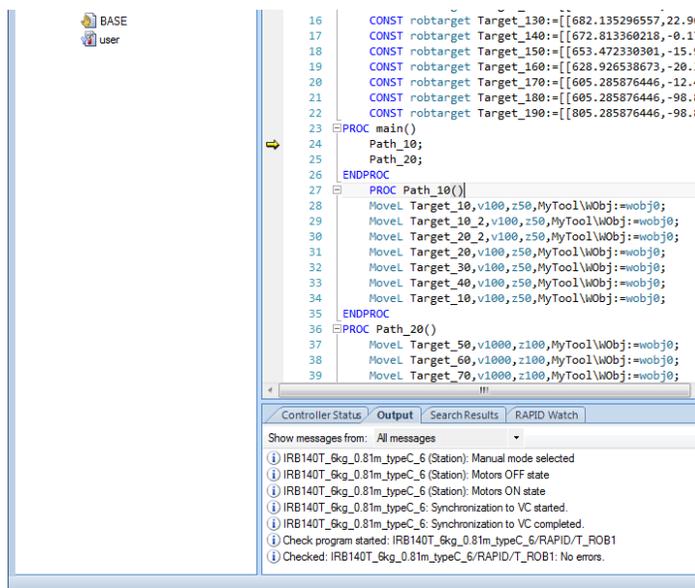


You can see the details of the robtargets that were created. You can also see the MoveL instructions for the two different paths.

13. You need to add a 'main' procedure of the module called MainModule. Do this by editing the module to include the following (place it prior to the Path\_10 procedure).

```
PROC main()  
  Path_10;  
  Path_20;  
ENDPROC
```

14. Click on the 'Apply Changes' icon on the toolbar. Make sure that no errors are found (check the results in the output status window). There should be no syntax or semantic errors.
15. You are now ready to run the program. But first you need to set the program pointer. Right-click in the editing window and select 'Set Program Pointer to Main in all tasks'. A yellow arrow should now be pointing to the call to Path\_10 (the first executable instruction in the main() routine).



The last thing you may want to do (it's optional – your call) is to click on the down arrow button below the Program Pointer icon in the toolbar and then uncheck Follow Program Pointer. [ If you don't uncheck this, the view will keep jumping from the view of the robot moving to the code window as the program executes. ]

16. Now click on the green arrow icon on the editor toolbar. This will start the program executing. Click on the View1 tab to see what's going on.



them back ON by pushing the white button again. Re-establish your program pointer and click the start button.

23. If the program only executes once and you want it to run continuously, click on the Run Mode button on the RAPID toolbar. Change it to 'Continuous' and then restart your RAPID program.