A Guide for ABB RobotStudio

This guide/manual explains step-by-step the most common functionalities of RobotStudio.

by

Pedro Neto
Department of Mechanical Engineering (POLO II), University of Coimbra
3030-788 Coimbra, Portugal

Email: pedro.neto@dem.uc.pt

January 2014

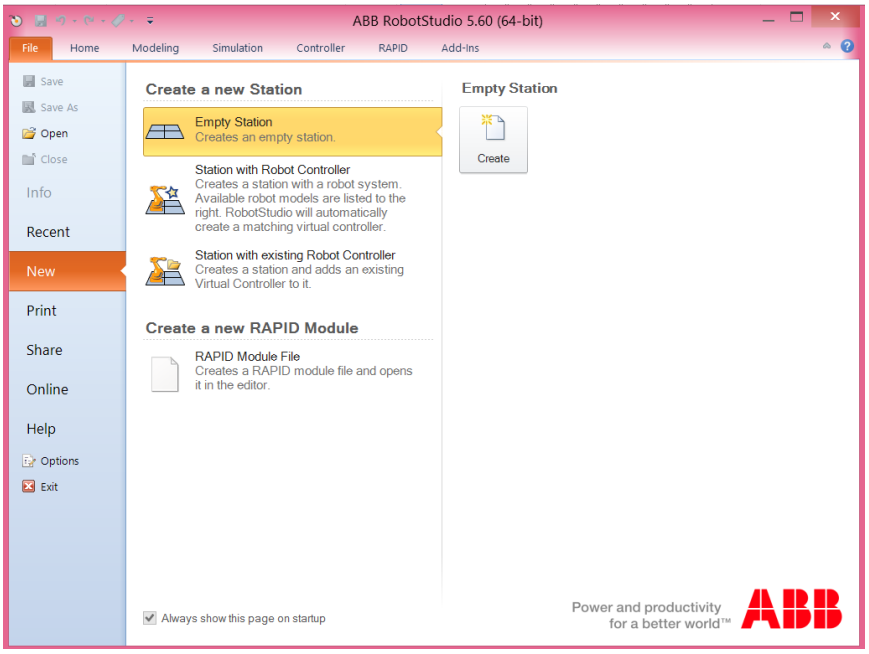
Contents

1. INTRODUCTION

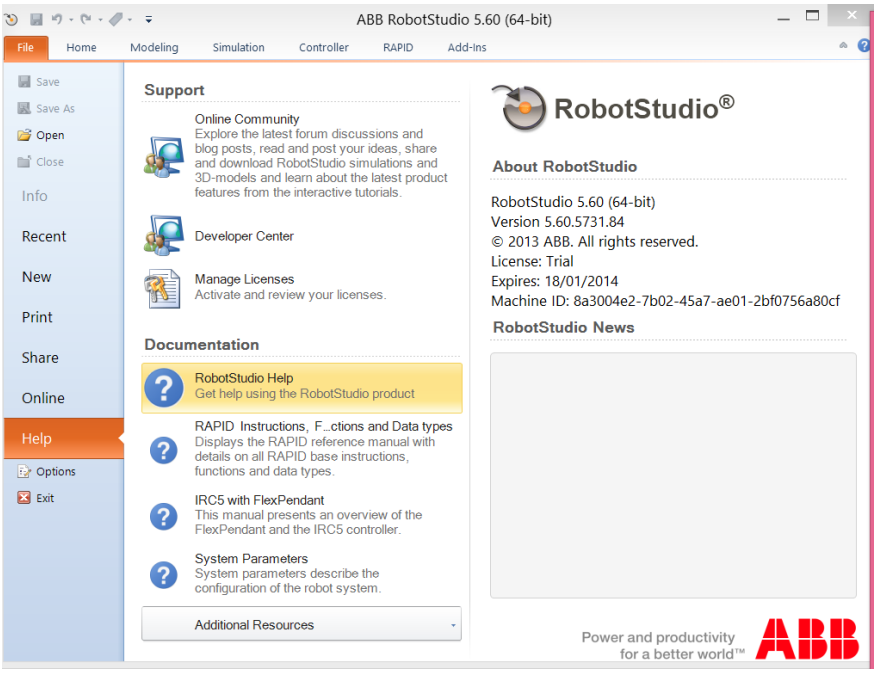
This guide contains the basic commands to start using ABB RobotStudio. It is divided in 4 modules in which each module is a different exercise. All the necessary procedures to create a given robot application are described, step-by-step. Several print screens from RobotStudio help to understand in a better way the process to operate RobotStudio and create robot programs off-line.

2. MODULE I

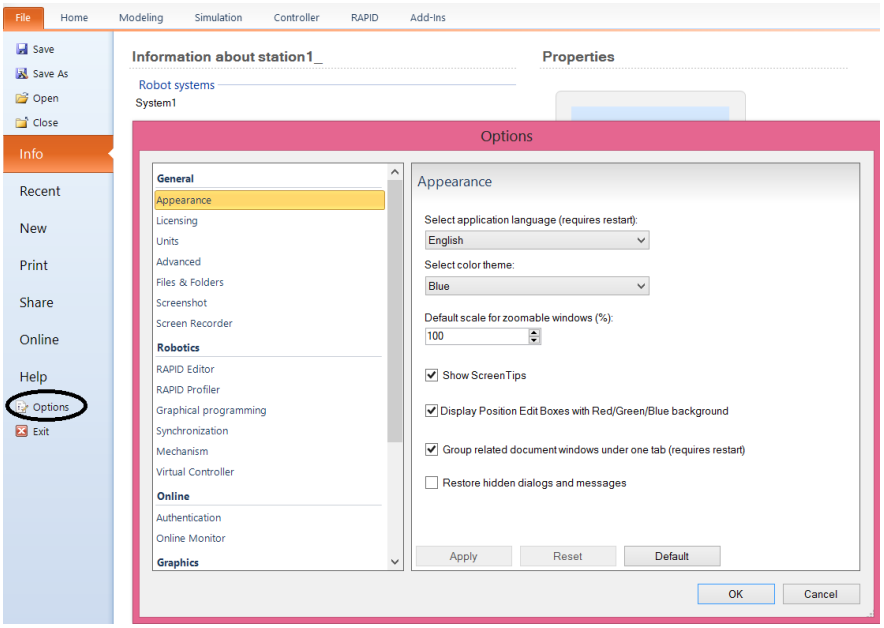
This is hoe RobotStudio looks like:



Here we can see the Help tab with all the necessary information about RobotStudio, RAPID language and the ABB teach pendant (FlexPendant).

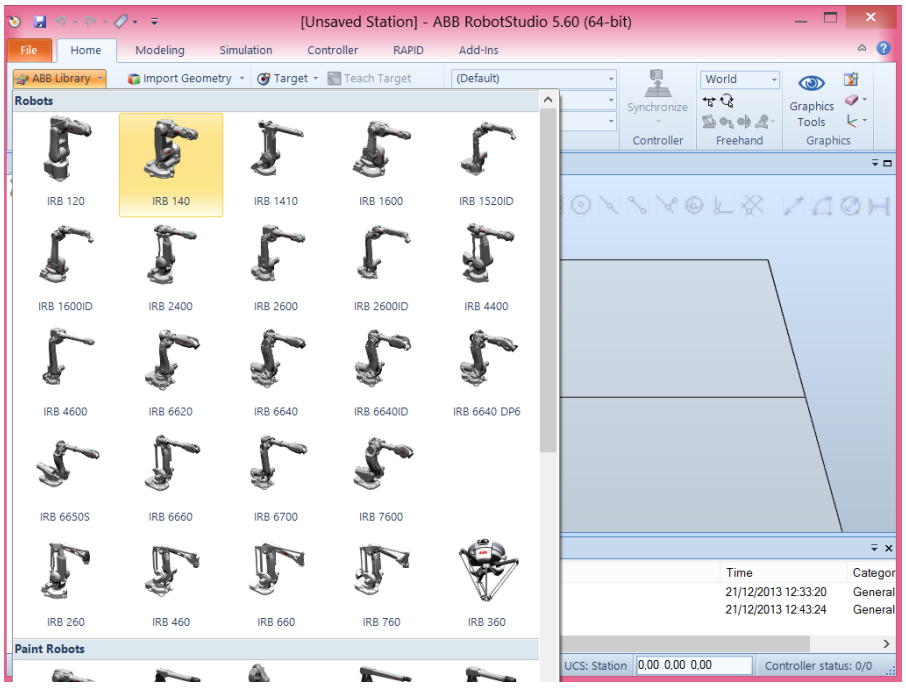


The software options can be changed in options section:

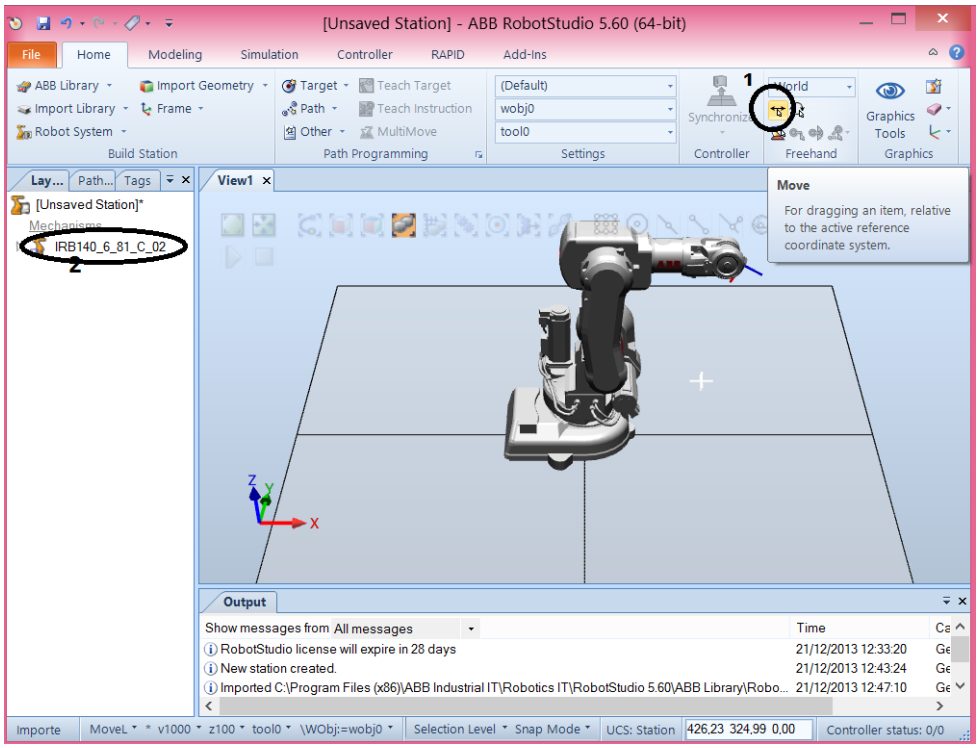


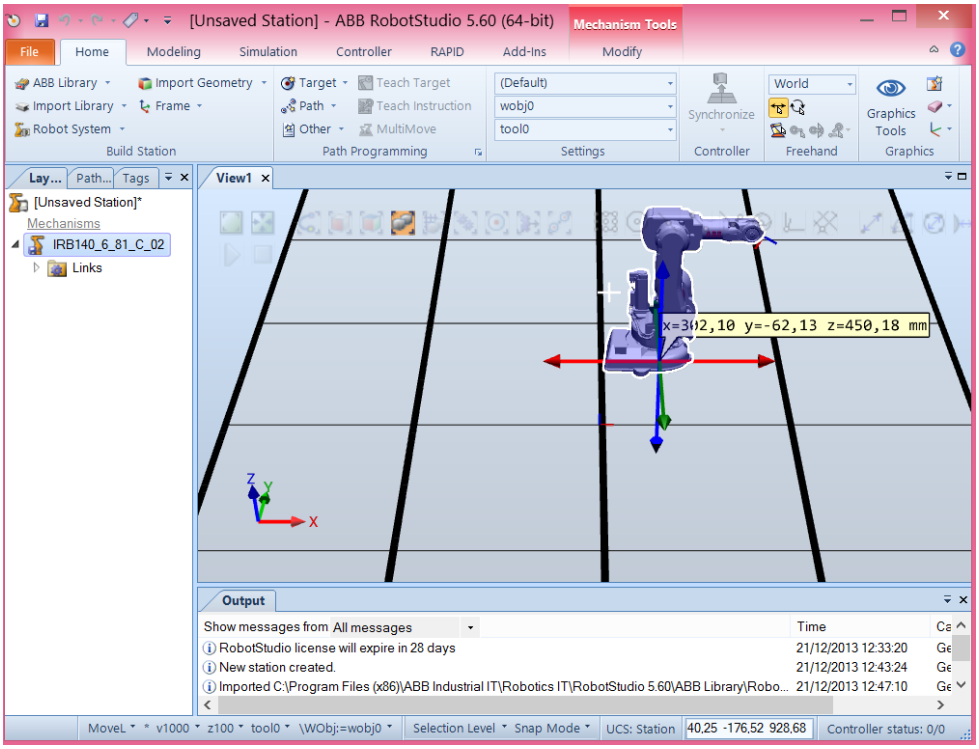
2.1. Robotic arms and positioning

The first think to do is to create an empty station. This empty station only contains the working plane with a reference system (world reference system). So, we need to include a robotic arm in the station. Just go to ABB Library and select the desired robotic arm.

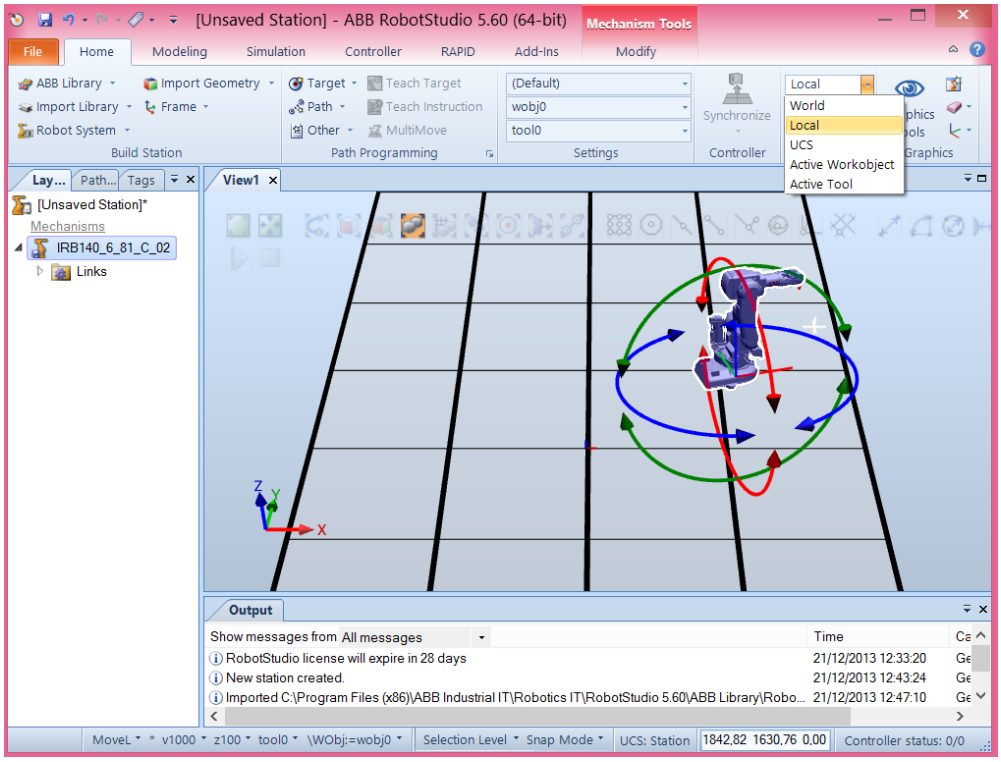


It is now possible to move (translations + rotations) the robot in the working space. In addition we can change the angle of each robot joint. To do that, just click on the robot model (see figure below the highlighted area on the left) and by clicking on the button move (highlighted in figure below) we can move the robot base with the mouse.

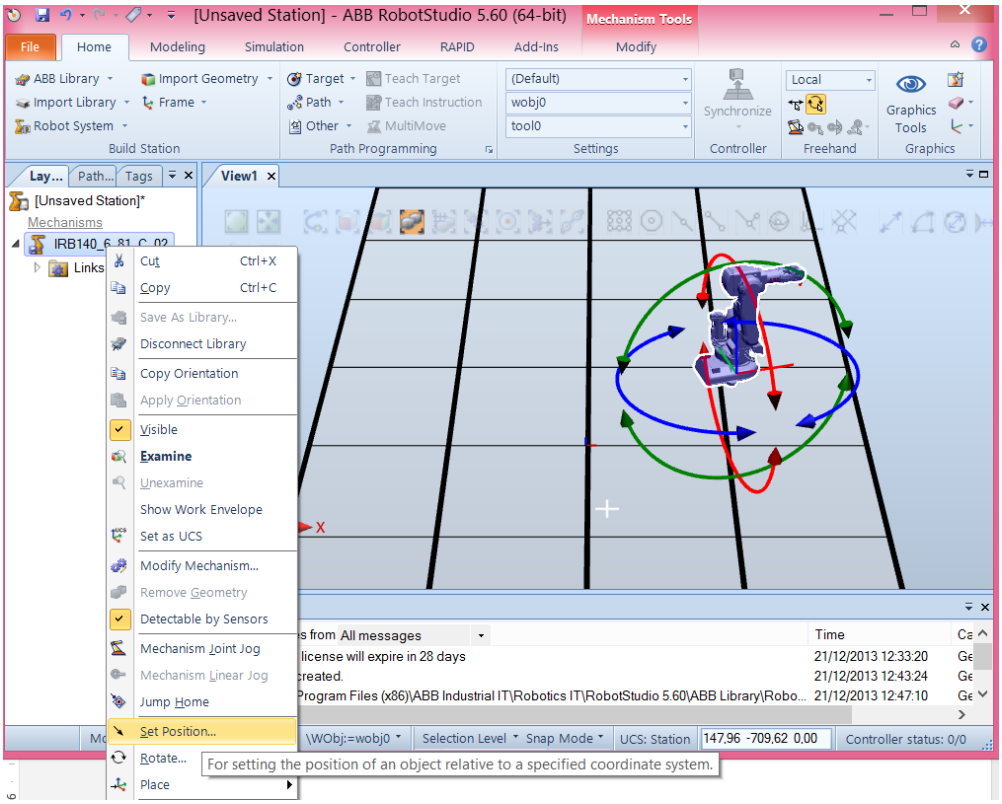


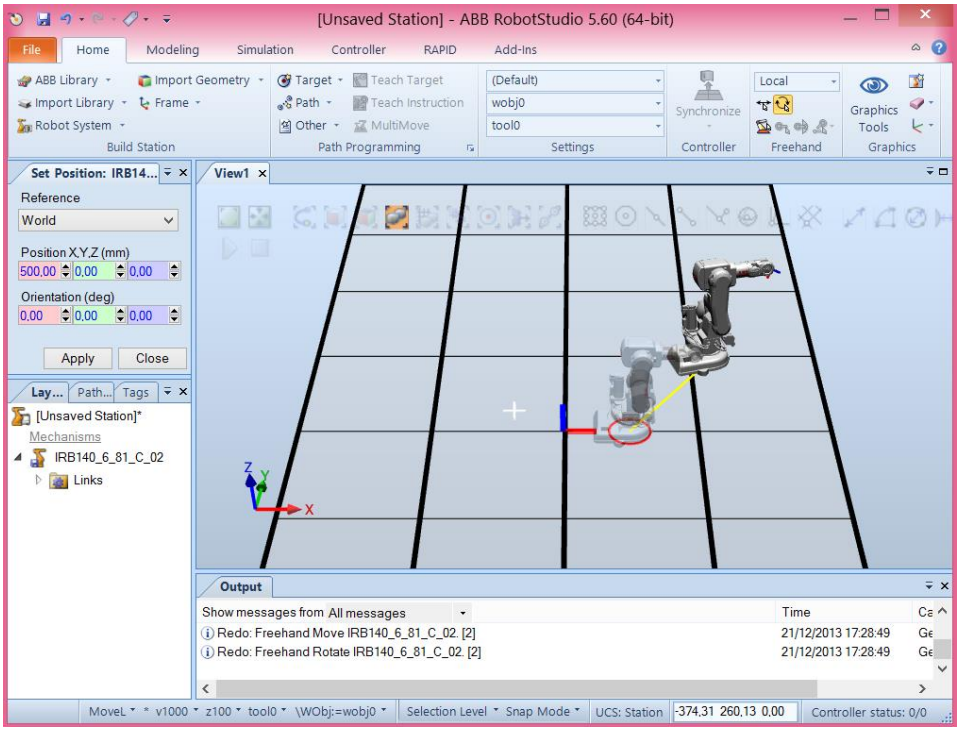


Moreover, we can also rotate the robot in relation to a selected reference system.



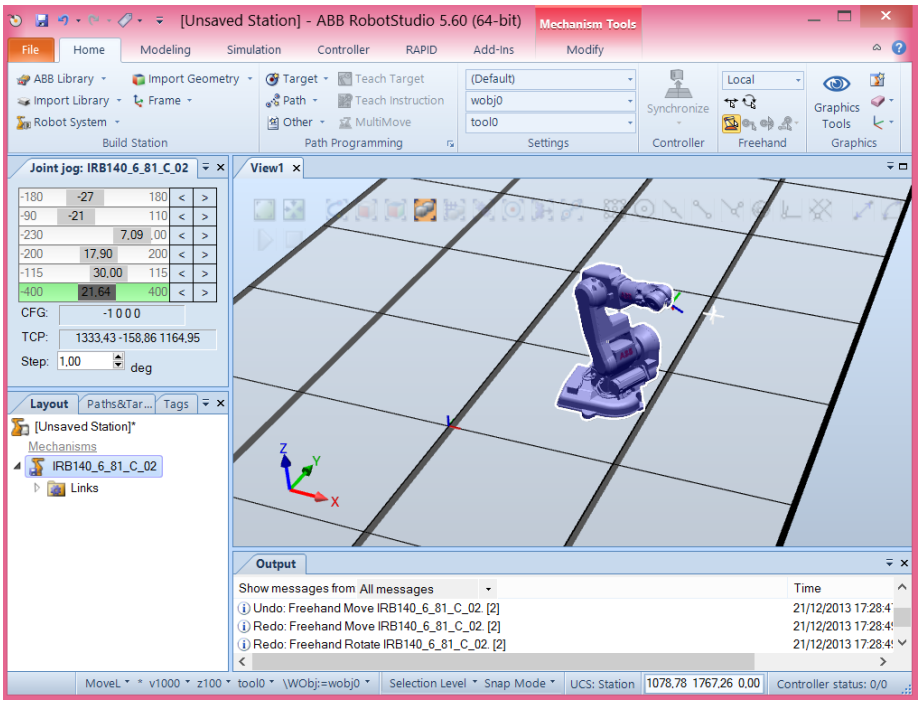
Ok, this is useful but we need accuracy when moving or rotating the robot:





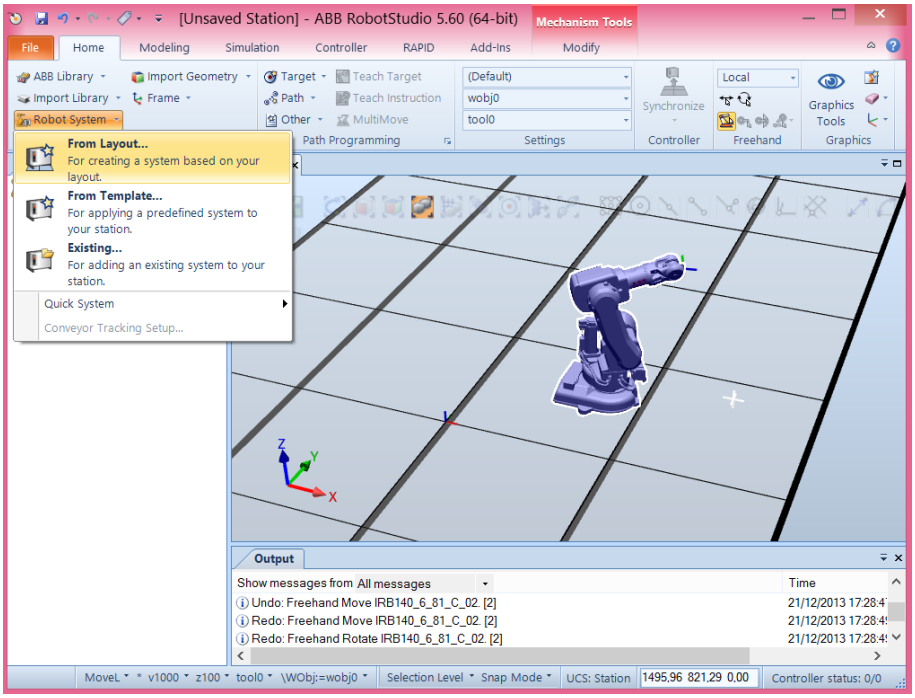
You can change the zoom of the working environment with the roller of the mouse. By pressing CTRL and left click with the mouse you can move the scene, and by pressing CTRL and SHIFT and left click with the mouse you can rotate the scene.

There follows an explanation on how you can change the robot joint angles, just right click on the robot model (Layout tab) and select Mechanism Joint Jog.

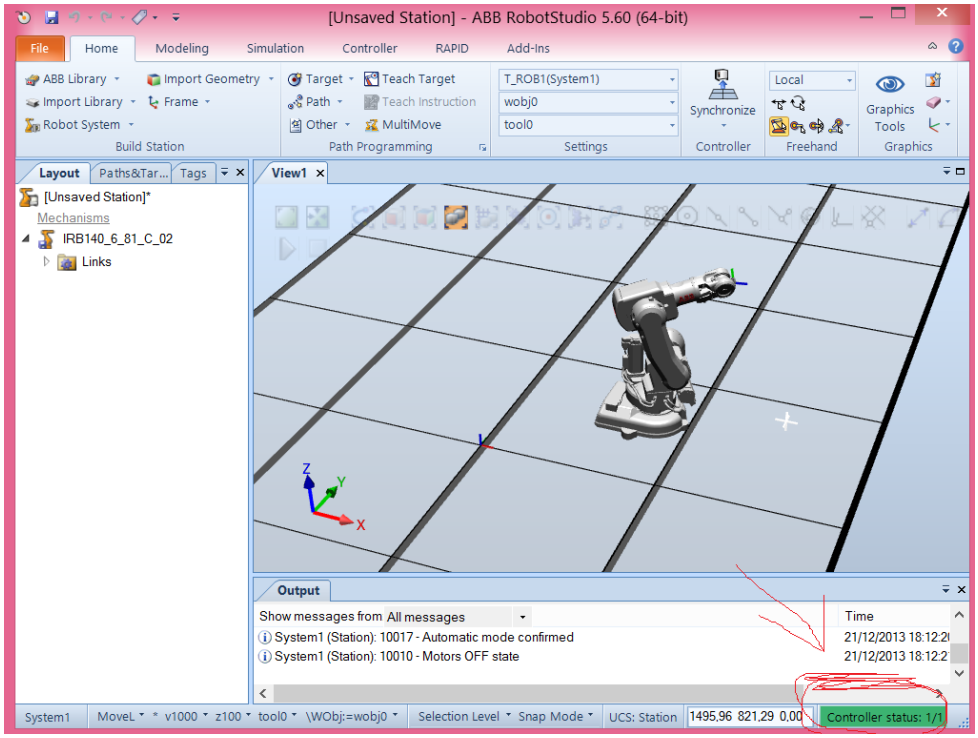


2.2. Controller

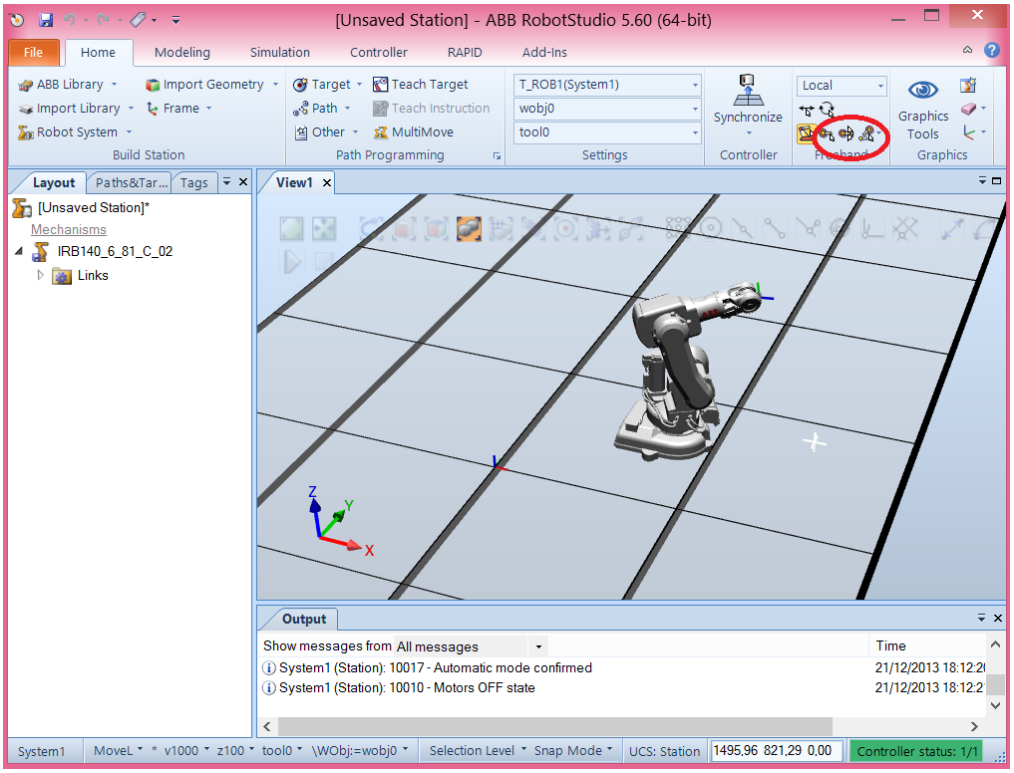
It is time to include a controller into the system.



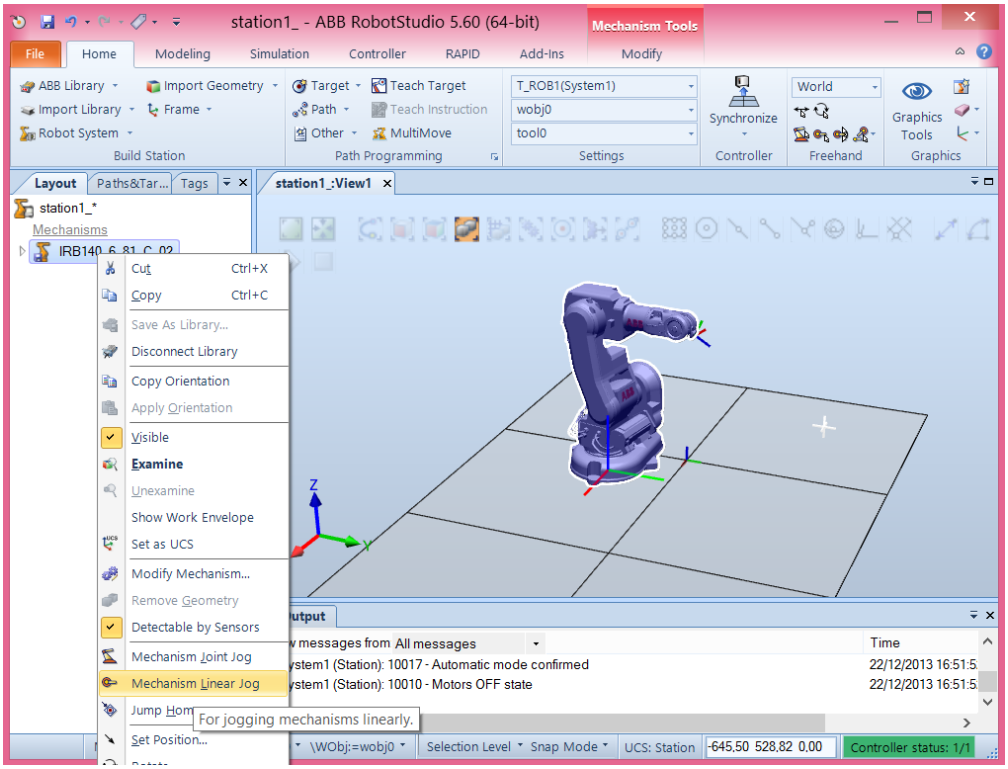
You can select a number of options for the controller, but for now, the default configuration is ok. So, just press, next, next and finish. Remember that this process may takes a little time until to obtain the green light (see figure below).



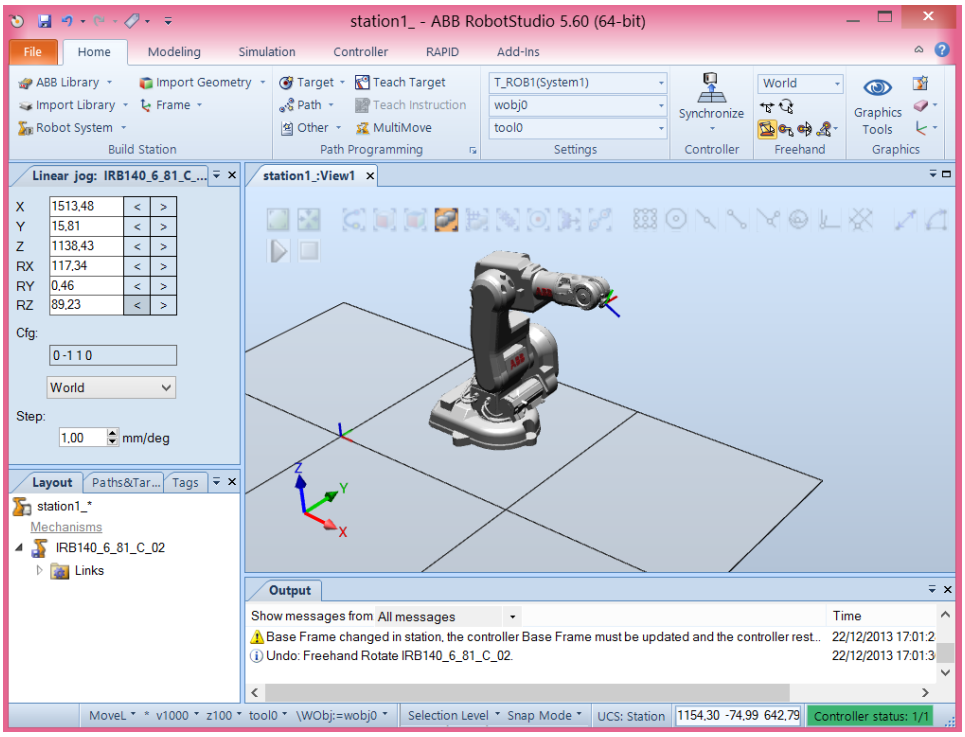
The 3 buttons highlighted in figure below are now active. This means that at this moment the virtual robot controller is ready to apply motion to the robotic arm.



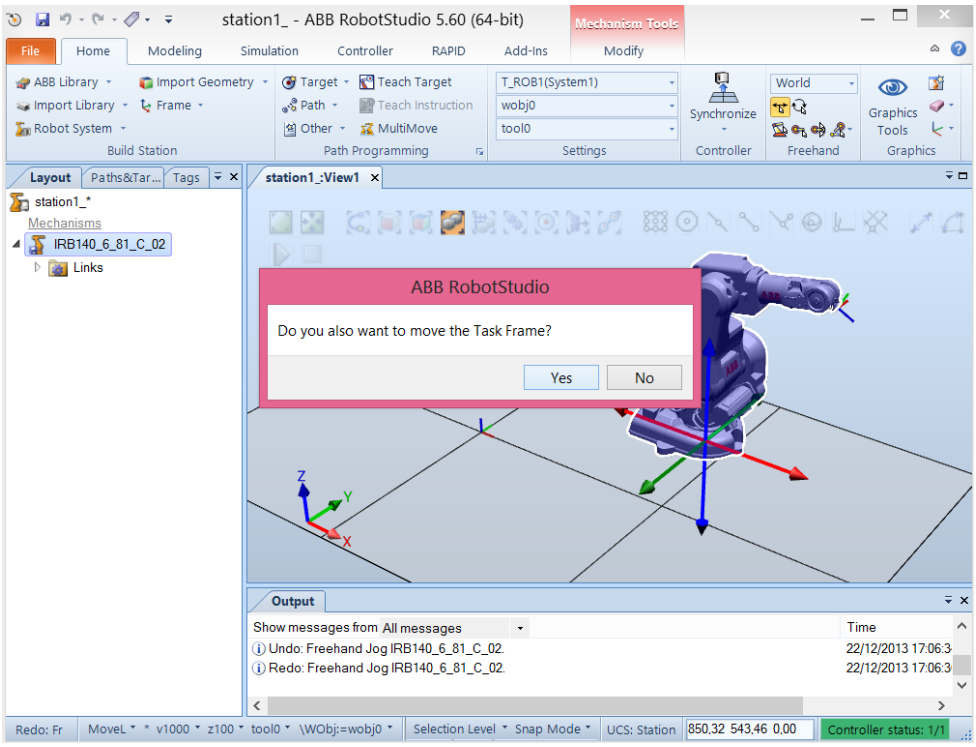
We can call these functionalities in a different way:



And this is the result:

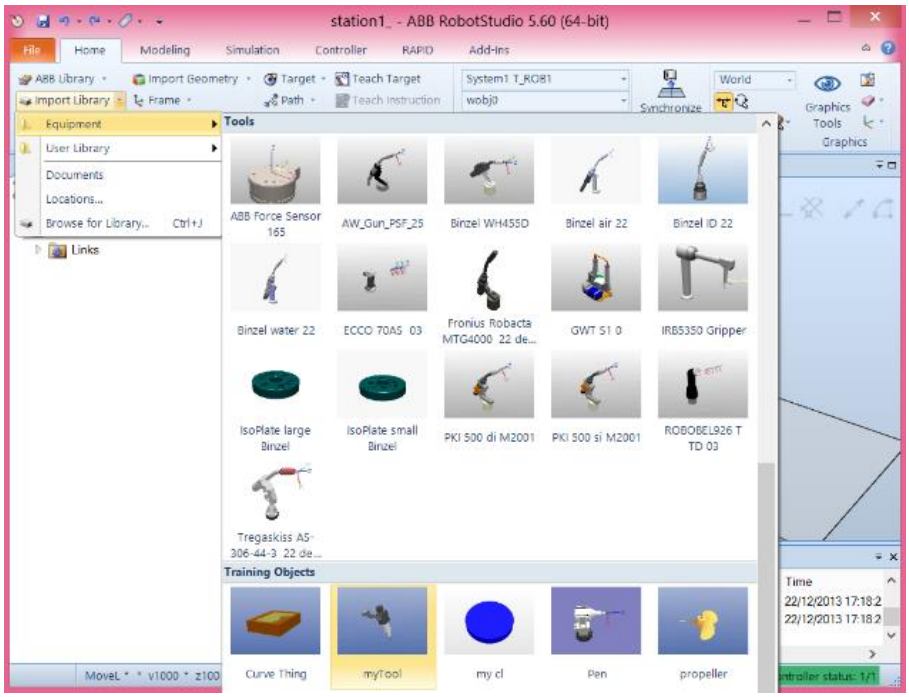


Now, since we have a controller associated to the arm, if we want to move the robot base the software asks if we want to move the task frame associated to the base of the robot. The answer is yes.

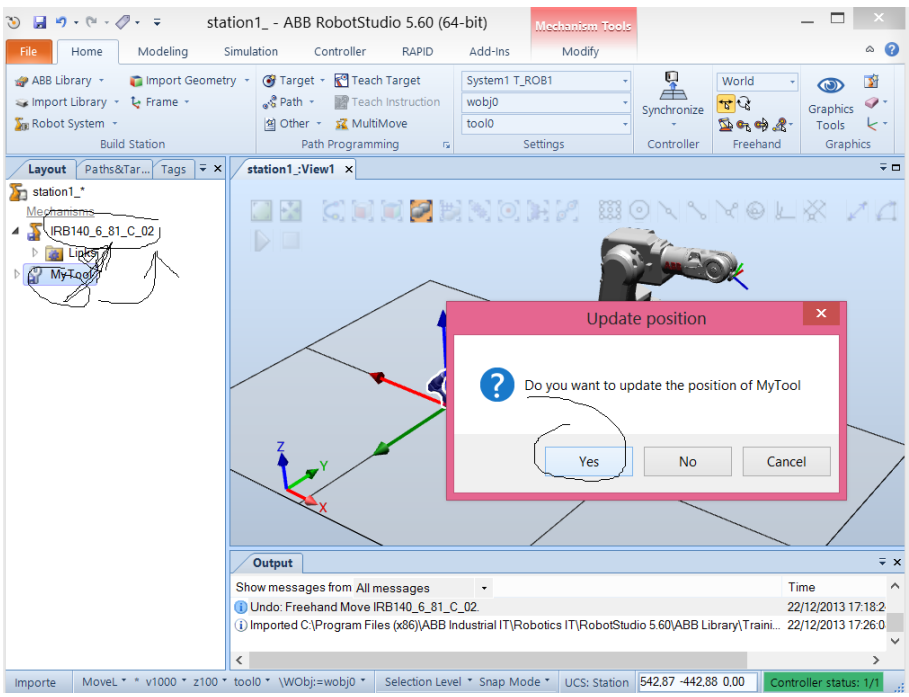


2.3. Tools

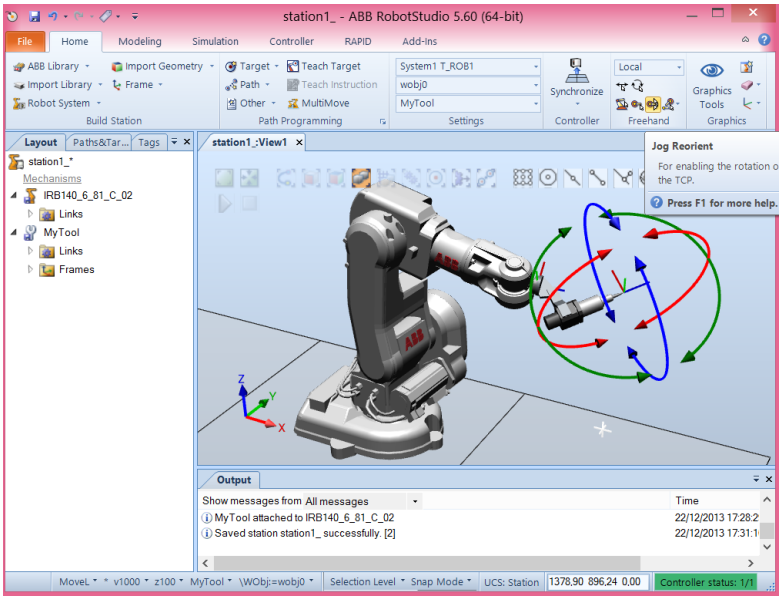
Next step is to attach a tool to the robot wrist. It can be imported from the library, select for example a pre-defined tool named MyTool.



After selecting a tool, we have to attach that tool to the robot, just drag it to inside the robot.

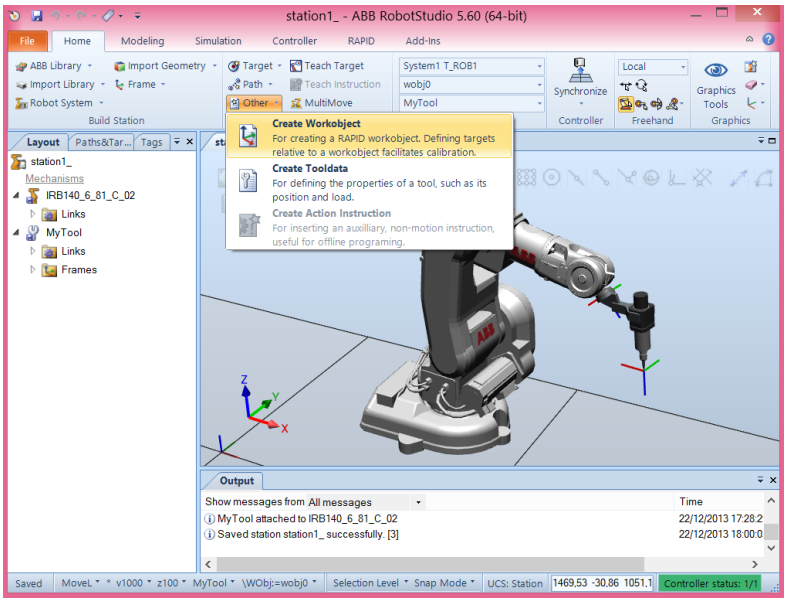


Now, it is possible to change the orientation of the tool but keeping the tool center point (TCP).

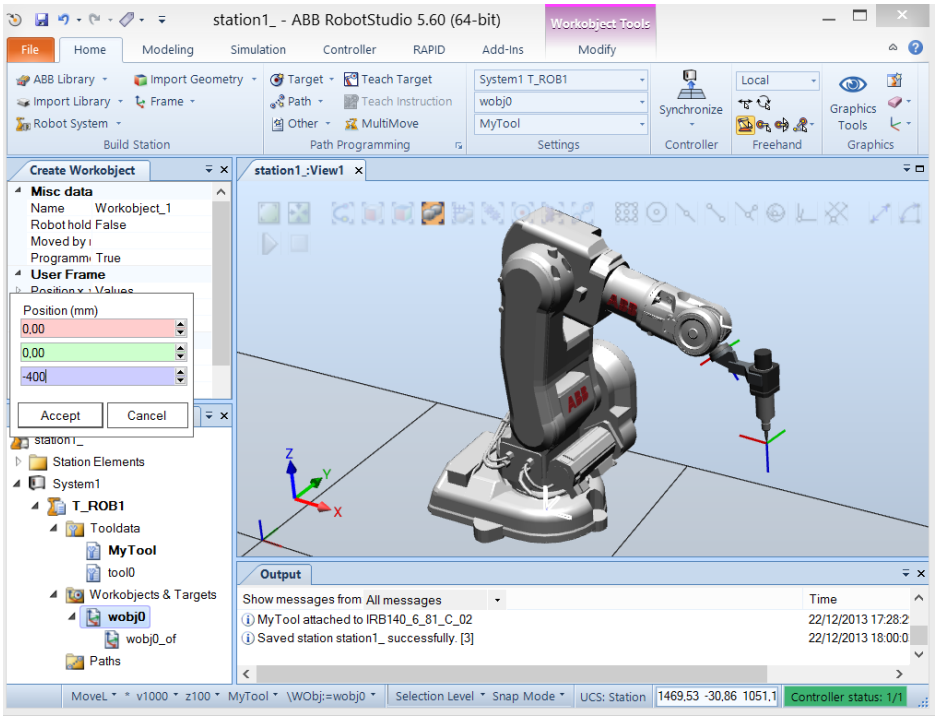


2.4. Workobjects

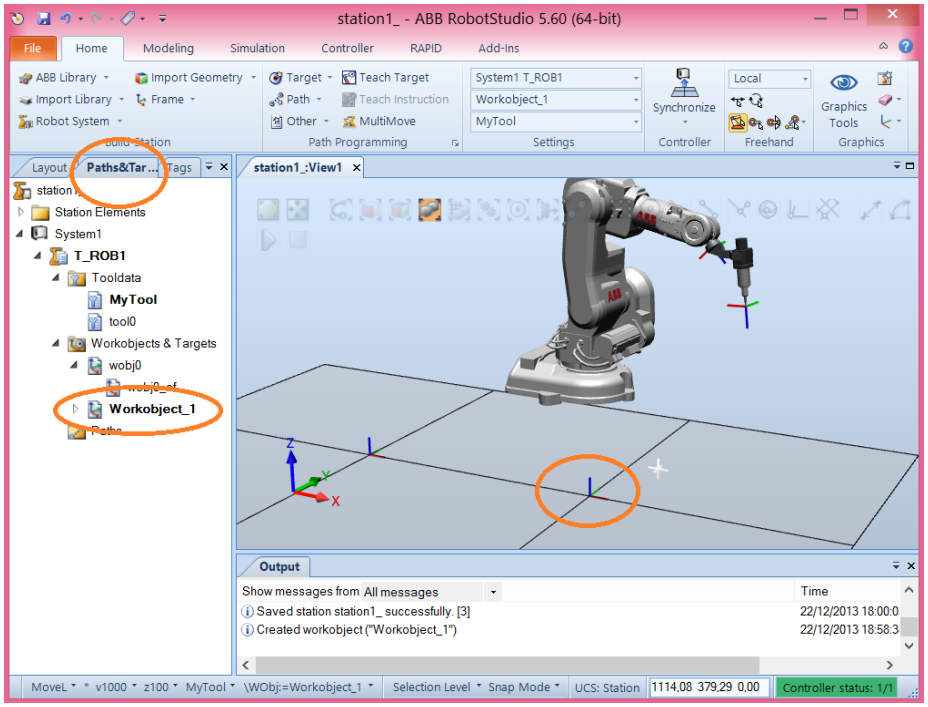
When we talk in robot programming the definition of a workobject is a subject of major importance. What is a workobject? Well, it is no more than a reference system and we define robot targets in relation to this reference system. In practice, this is very important for the calibration process between the virtual environment and the real environment with a real robot. Usually, the origin of a workobject is a point that can be easily defined, for example the corner of a table. Let's create a workobject:



By default, the workobject we created named workobject_1 is coincident with wobj0 (in the base of the robot). In this case, imagine that we want to have it defined at the level of the floor and imagine also that the robot is over the floor 400 mm, so we have to define the workobject with -400 along z axis.

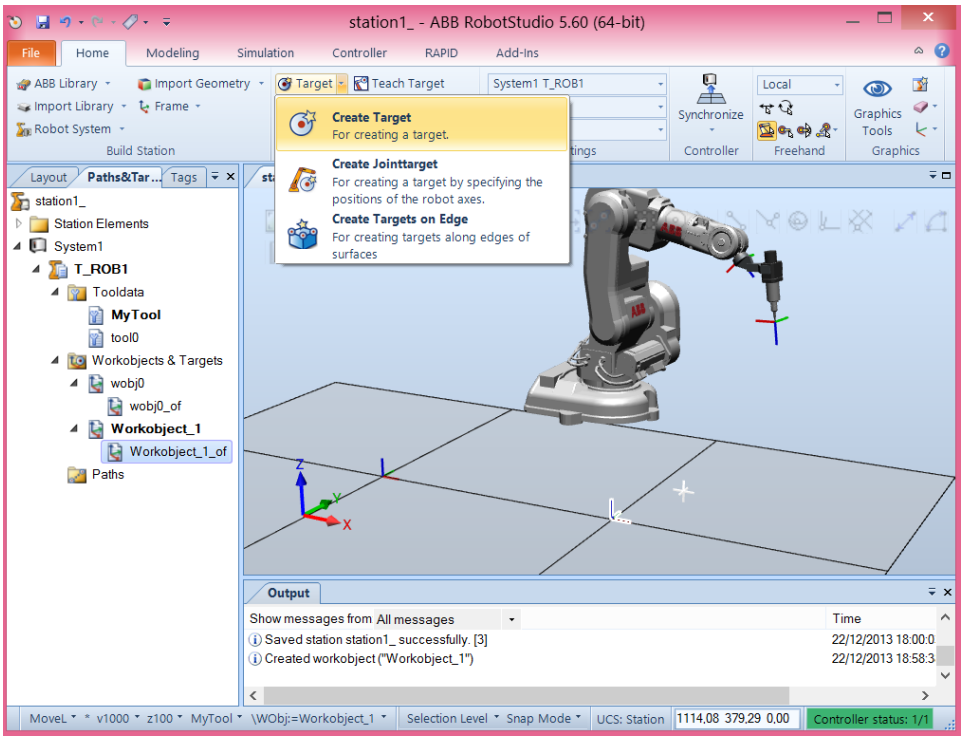


And we have the workobject:

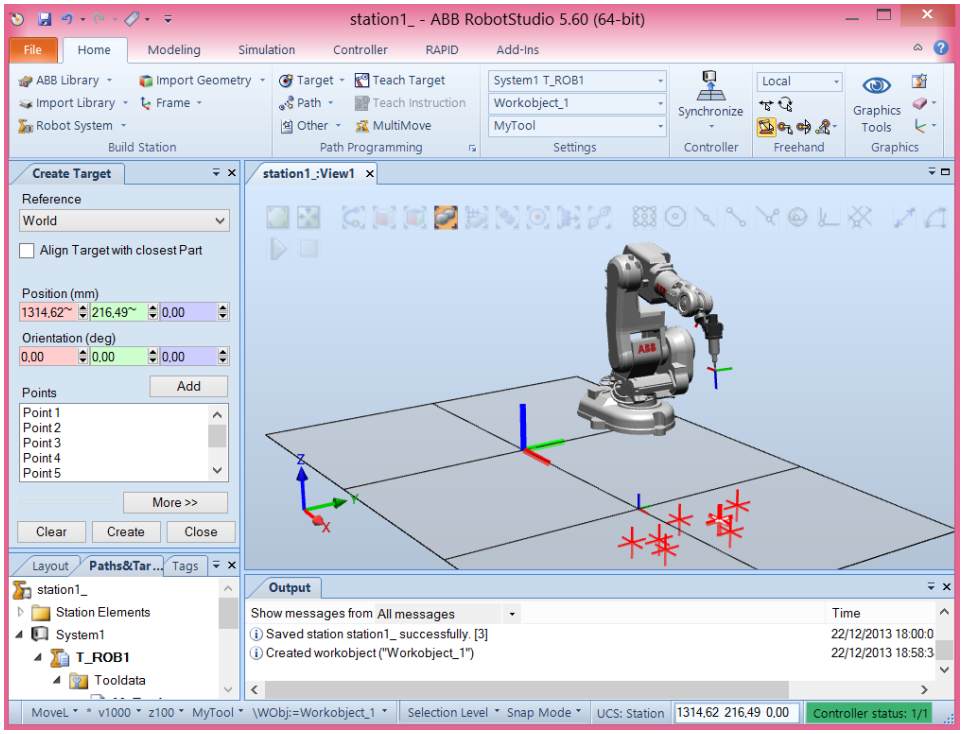


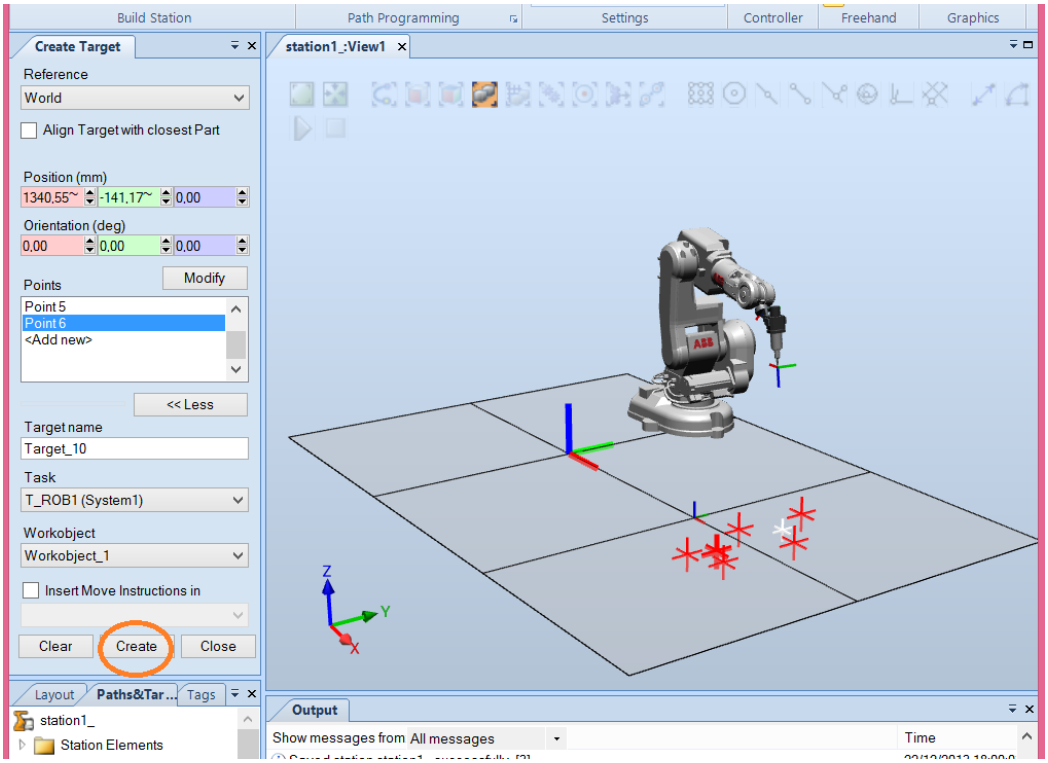
2.5. Targets

I tis now the time to define the target points that will be the base for the robot paths.

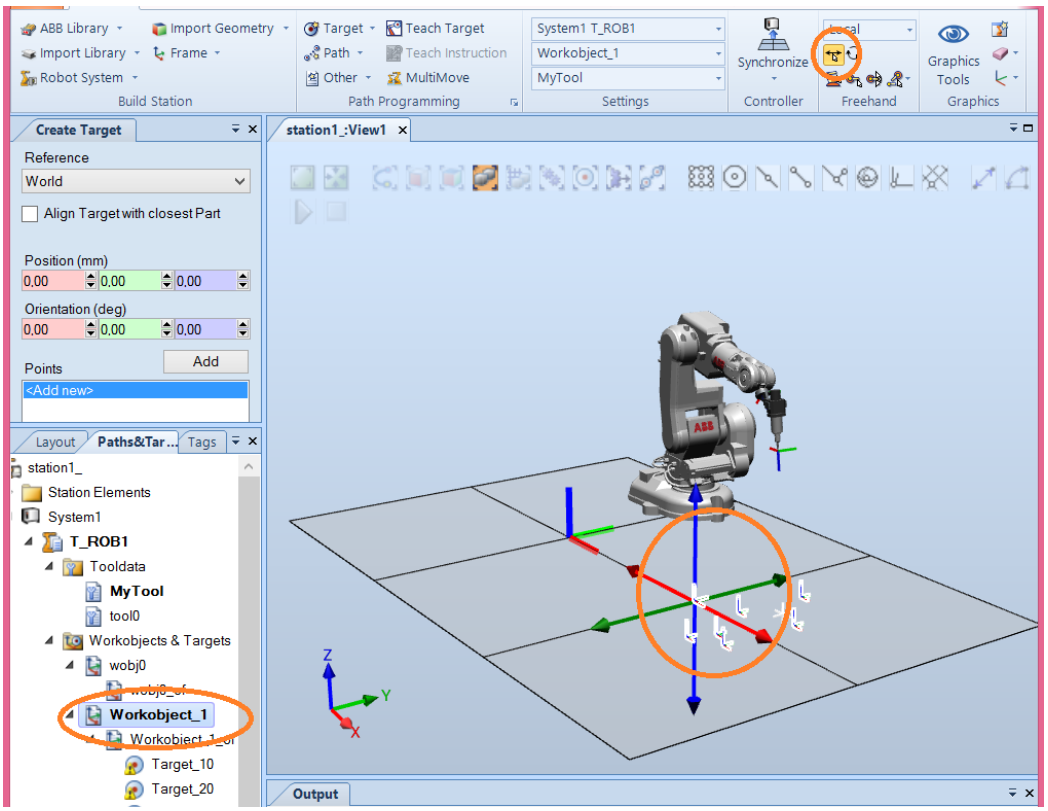


It is possible to use the mouse to create the target points. In this specific case, we are creating those points in relation to workobject_1.

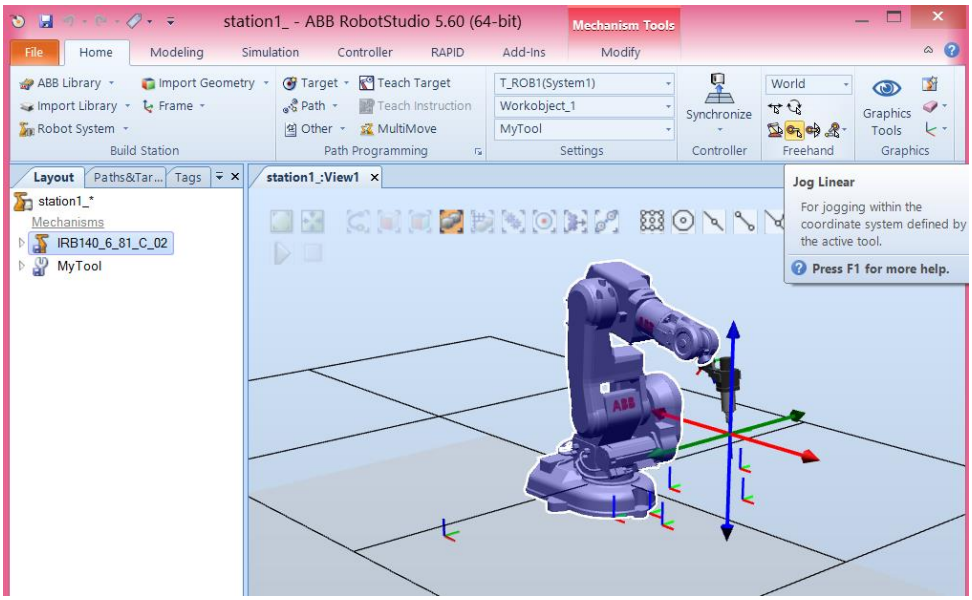




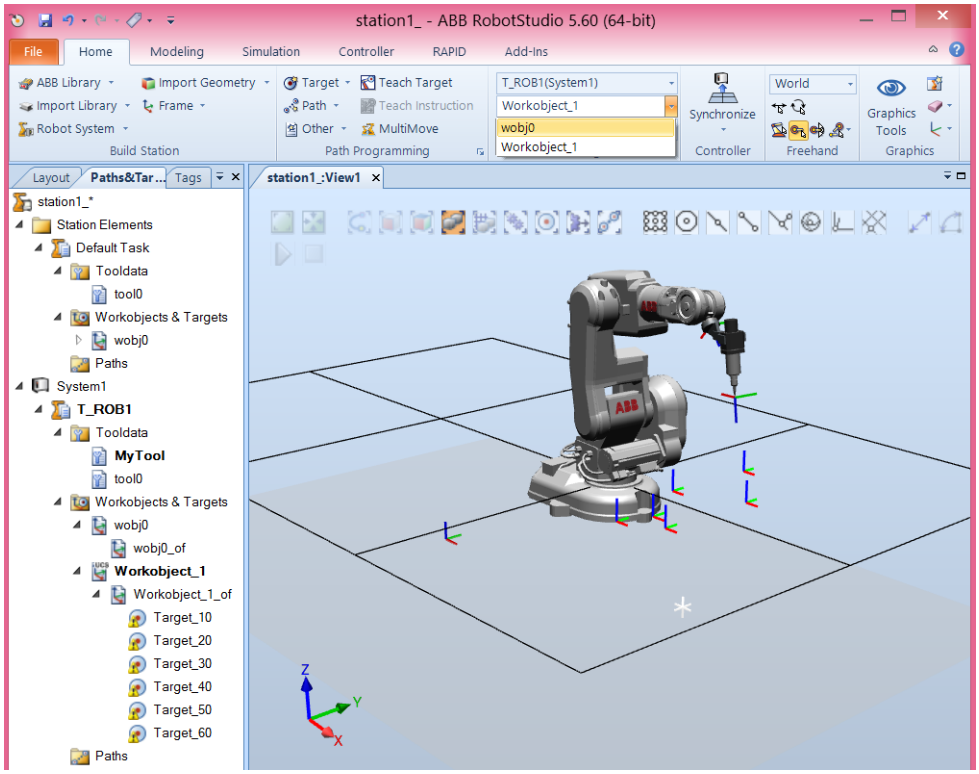
Now, it becomes easy to move the workobject_1 with the associated target points.



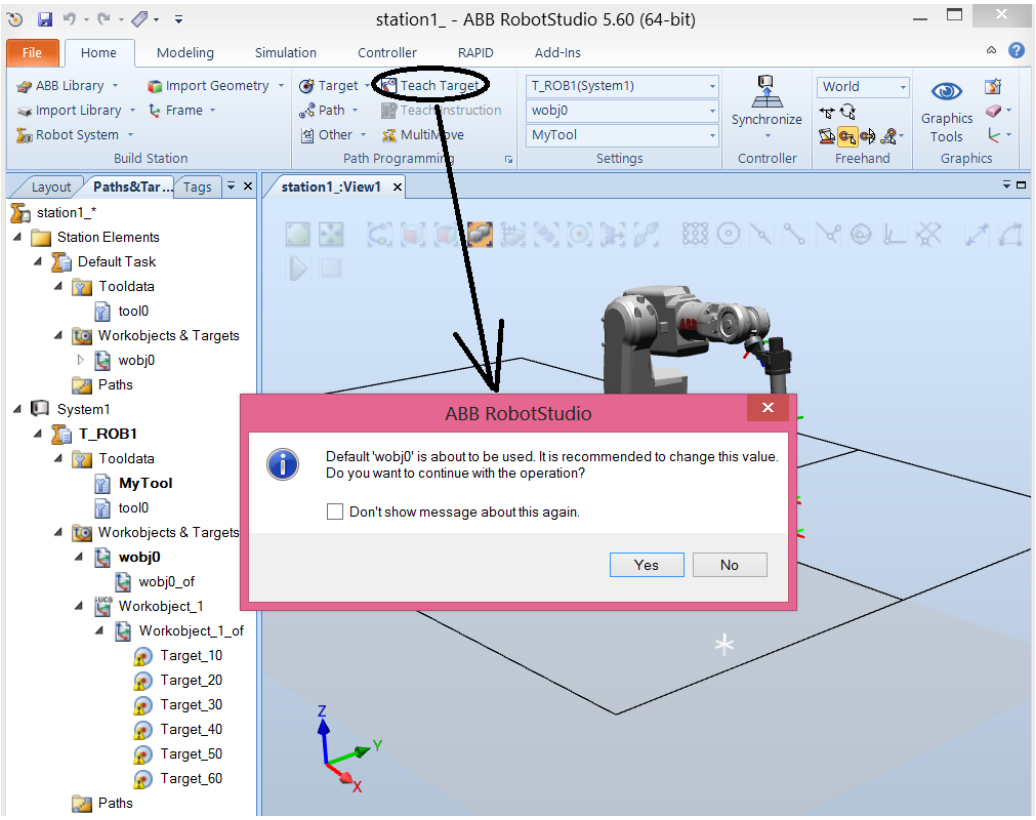
At this moment we don't have information if the robot effectively can reaches the targets defined or not. Anyway, in most of robot applications/programs we usually define a home position for the robot in relation to the base of the robot, in this case, wobj0. So, you can move the robot to a desired home position:



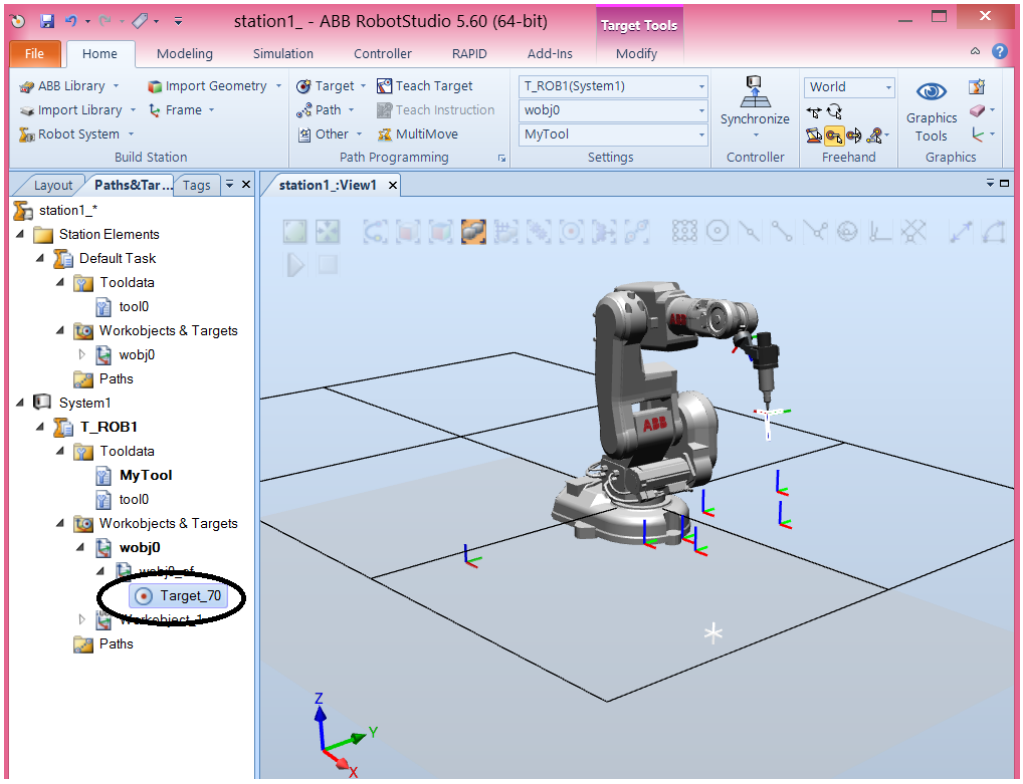
And create another type of target, a Teach Target in relation to wobj0. This creates a target according to the current position of the robot.



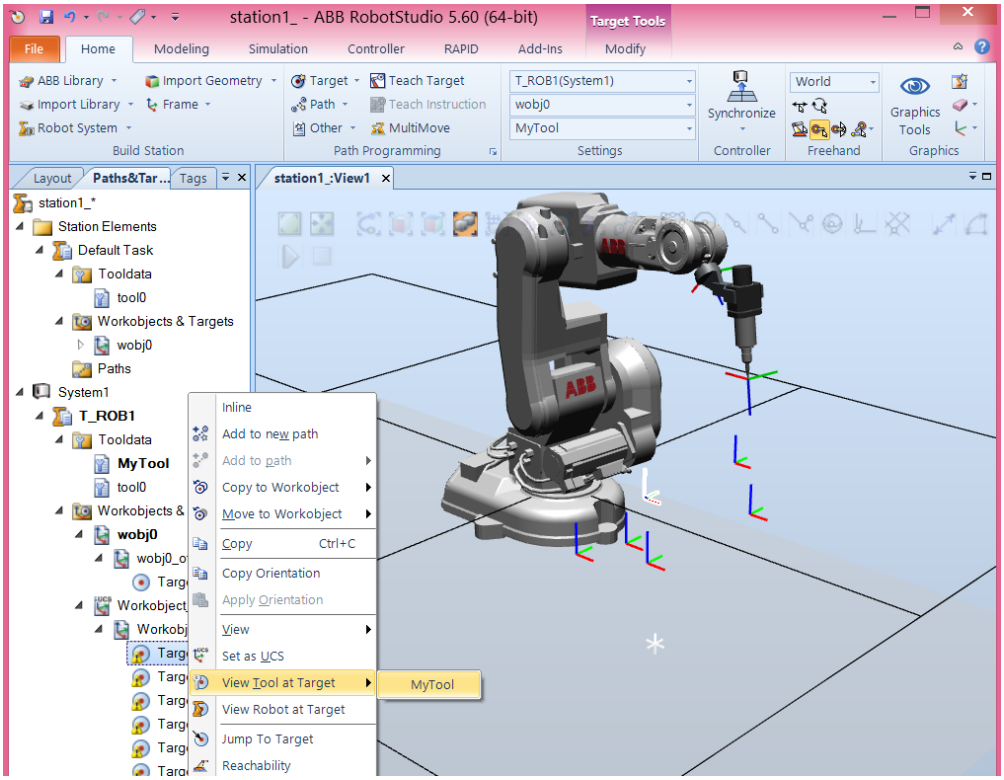
Answer yes (see figure below):

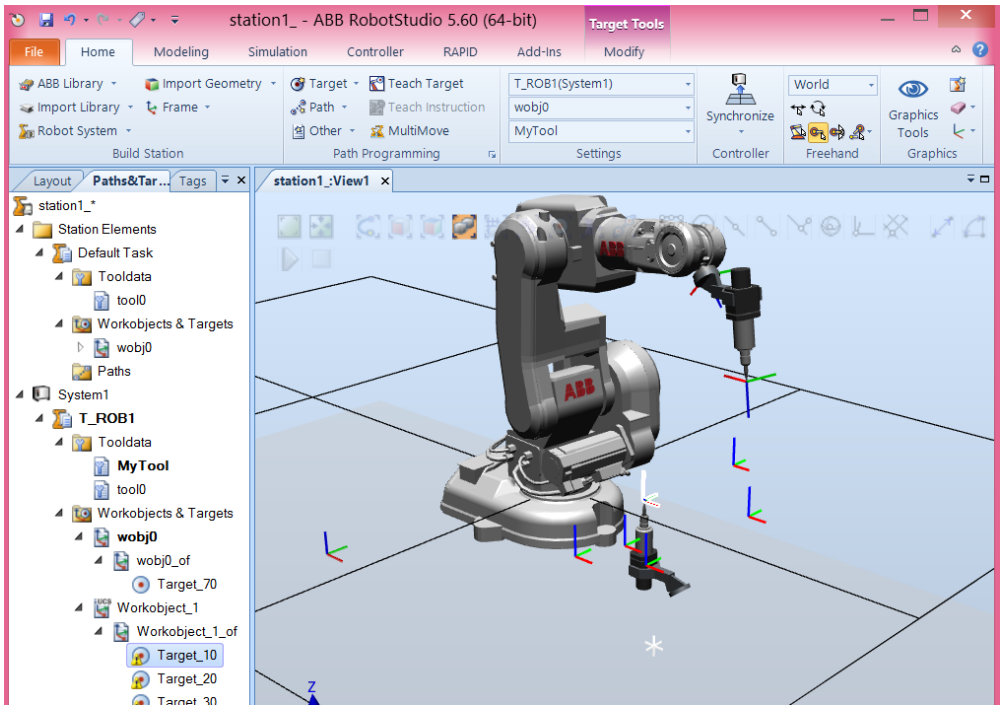


And the target point was created:

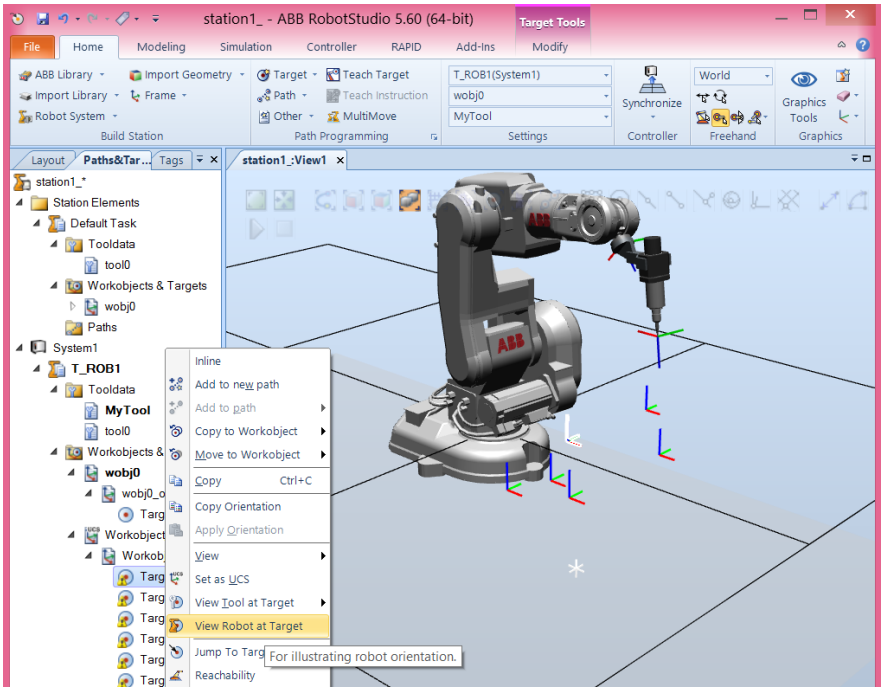


After this, we are going to check if the robot reaches of not the previously defined target points. Starting by the tool:

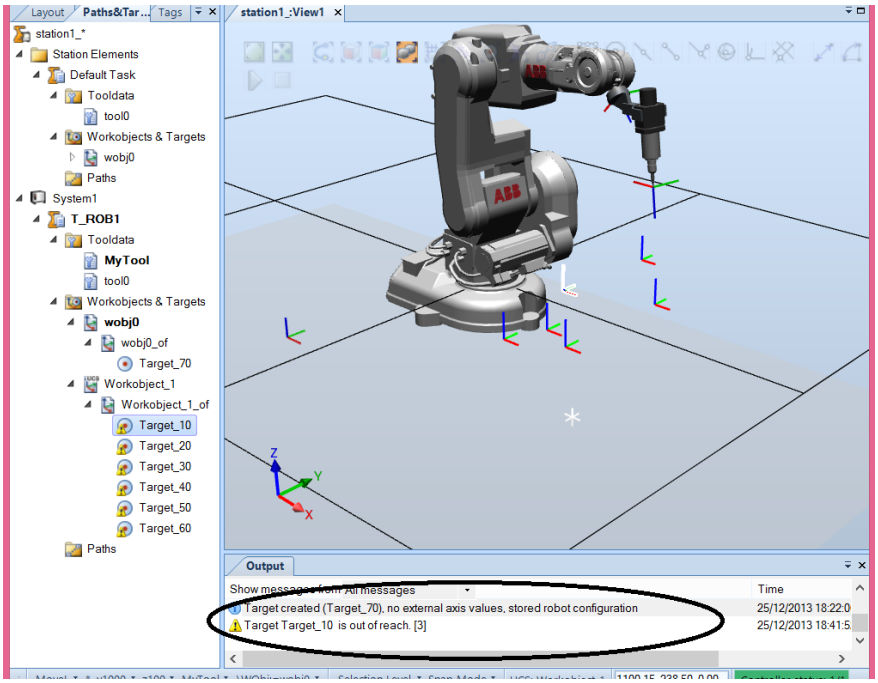




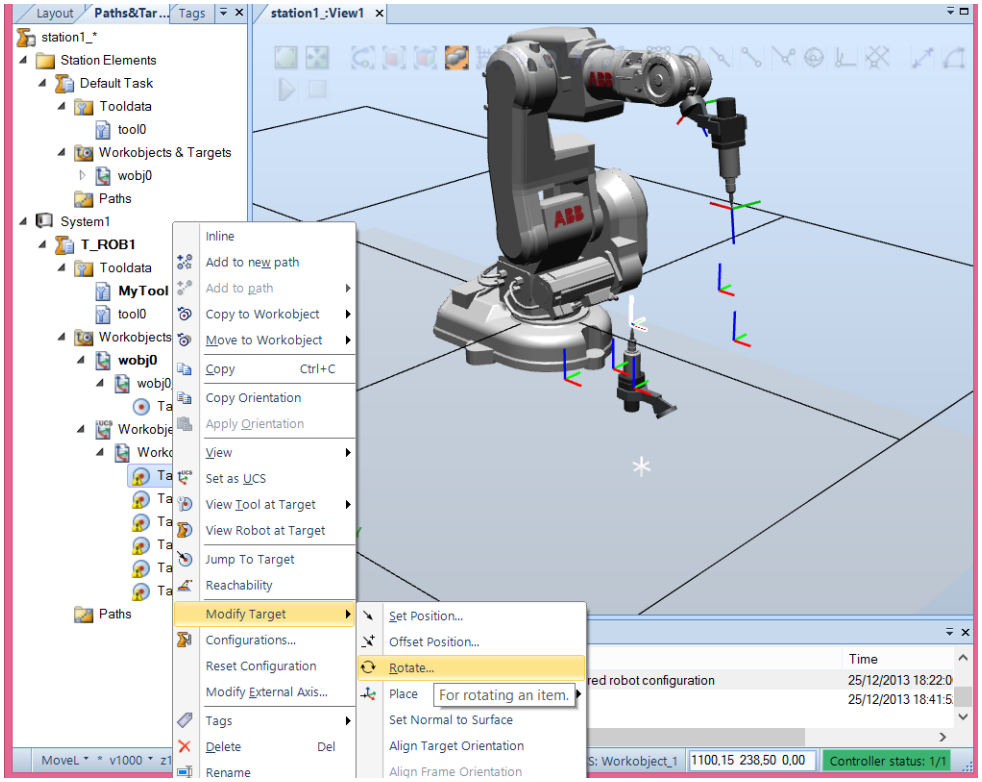
Well, we can see that the tool is with a wrong orientation. But, anyway, we can try to see the robot:

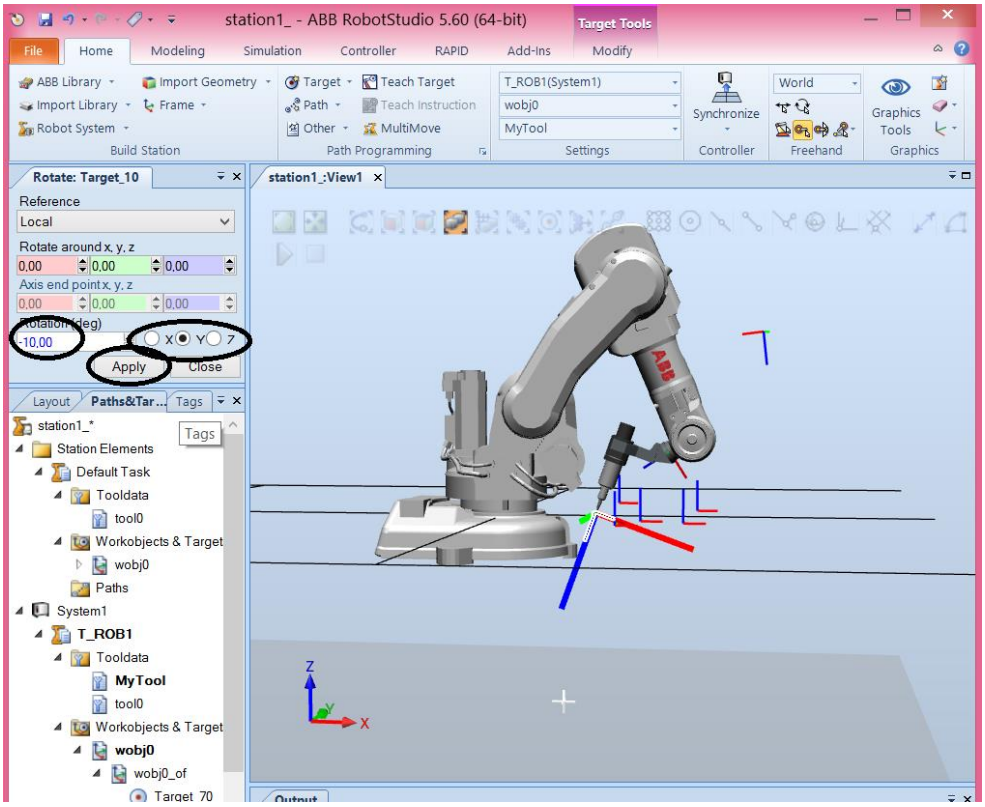


Nothing happens because the robot is not able to reach such position, RobotStudio gives us a warning.

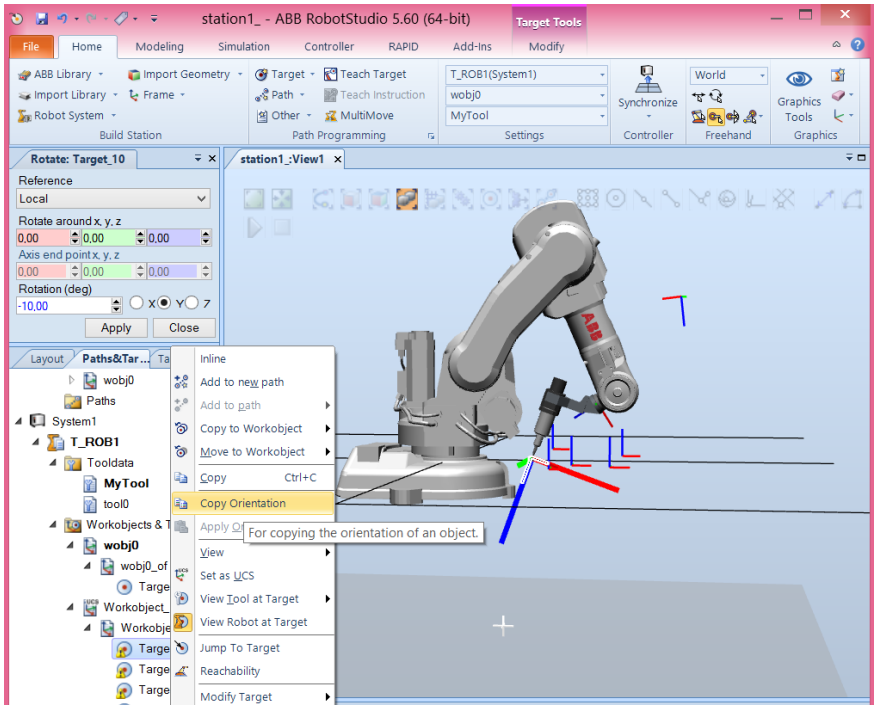


So, to fix this problem, we have to change the orientation of the tool:

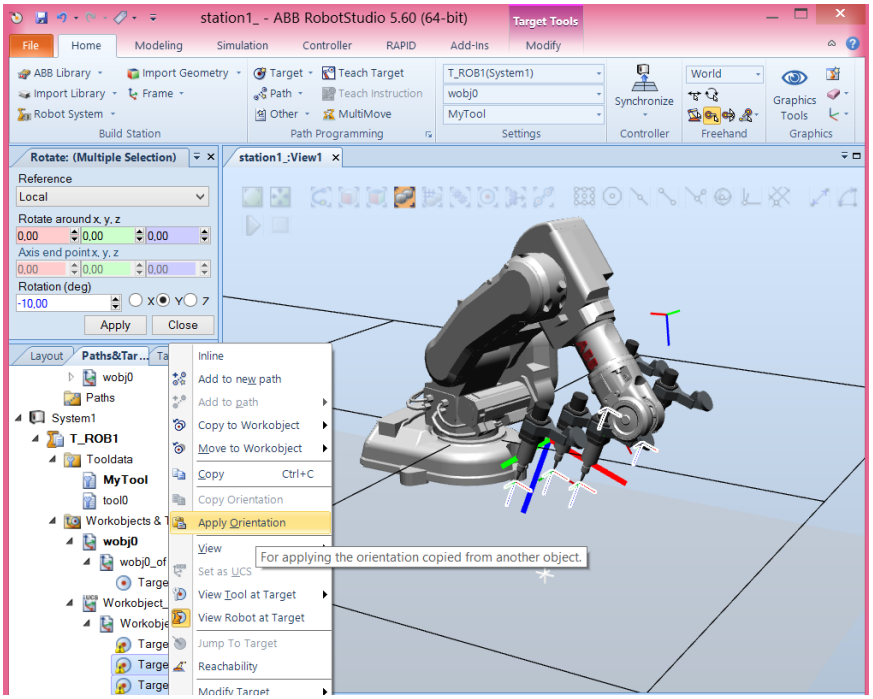




I did this to Target_10, but the other targets suffer the same problem. So, we can copy the orientation of this target and apply that orientation to all the other targets.



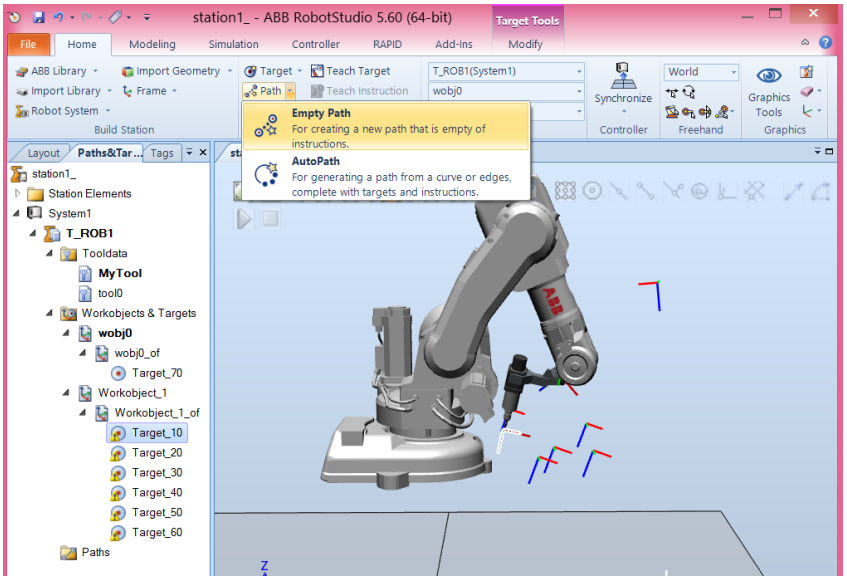
Select the other target points and apply orientation:



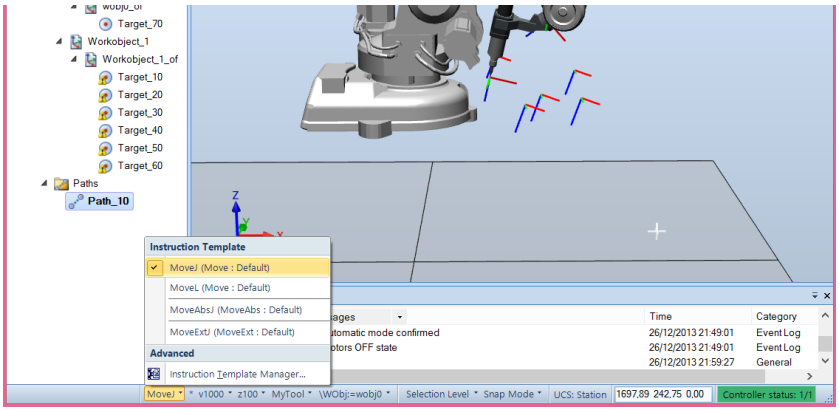
Ok, now, by clicking on the targets we can see if the robot reaches or not that targets. If the robot do not reaches a target we have to change that target point (position or orientation) using the command Set Position or Rotate.

2.6. Paths

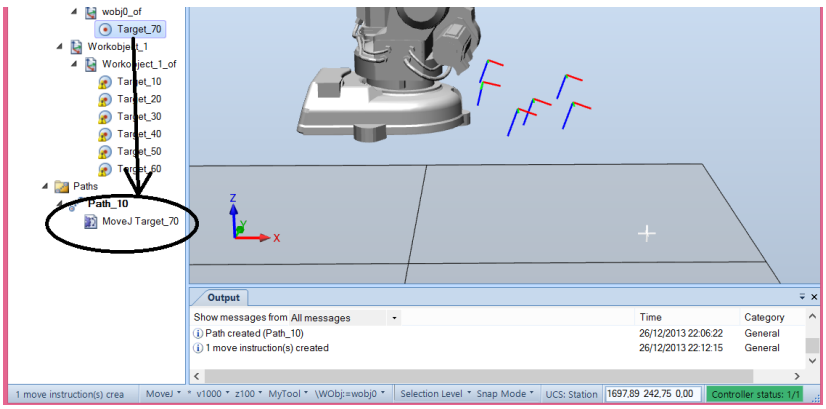
We can now connect the target points to create a working path for the robot.

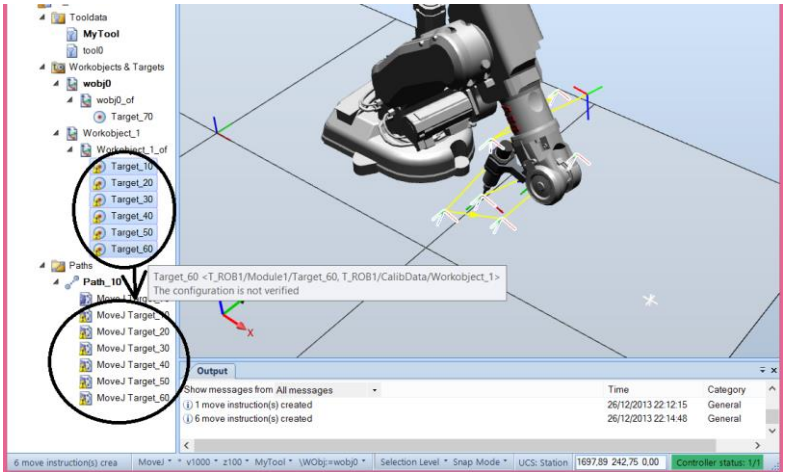


And select MOVEL (straight line motion between targets)



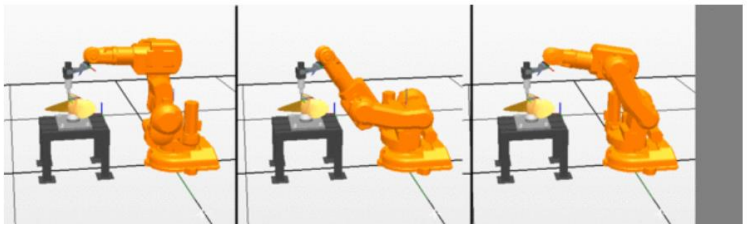
After having created Path_10, drag the targets to Path_10 in a desired order.



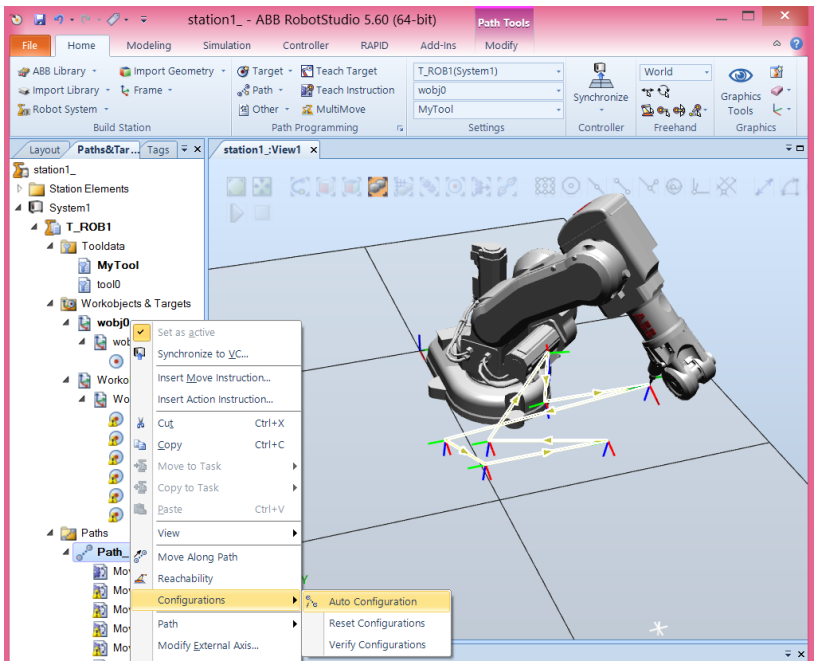


OK, we have the targets connected, but to eliminate the warnings we have to define the robot configuration for each one. This is because, as you can see in figure below, there are different ways to achieve the same robot tool position and orientation.

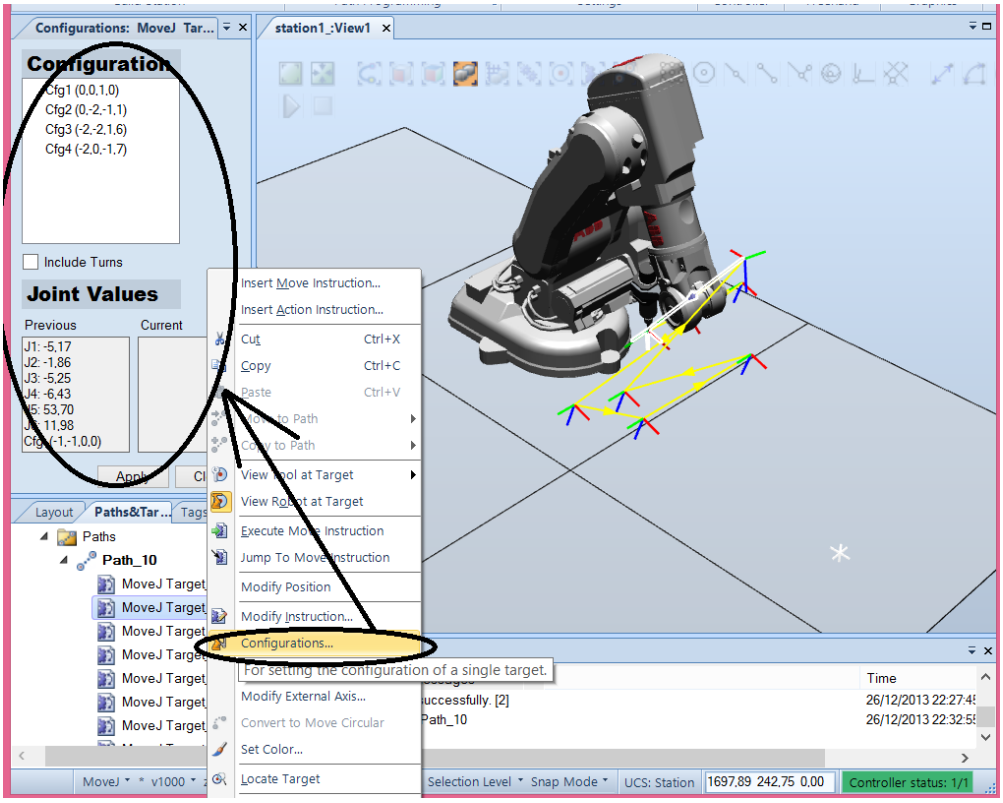
Axis configurations Targets are defined and stored as coordinates in a WorkObject coordinate system. When the controller calculates the position of the robot axes for reaching the target, it will often find more than one possible solution to configuring the robot axes.



The software is able to auto-configure this:

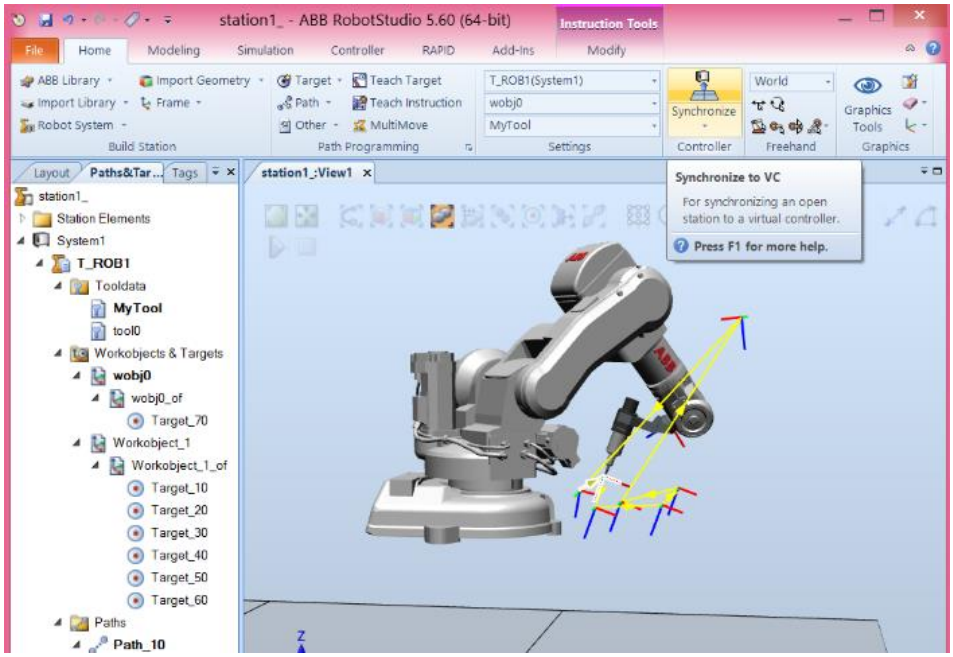


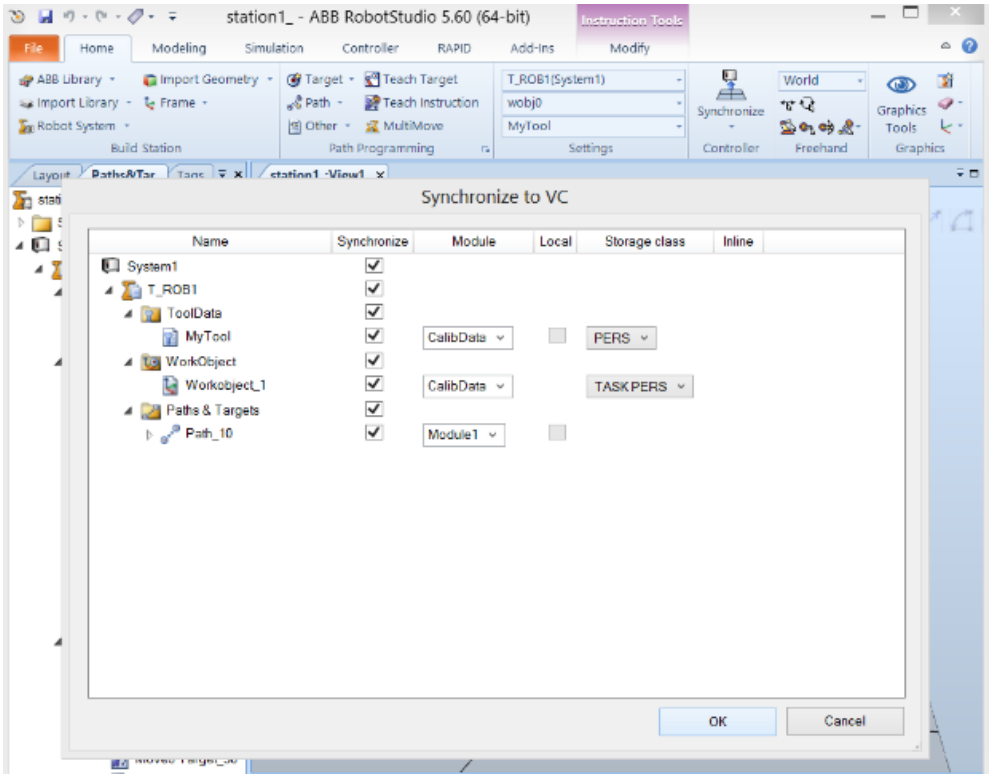
Or we can check and define the configuration for each target:



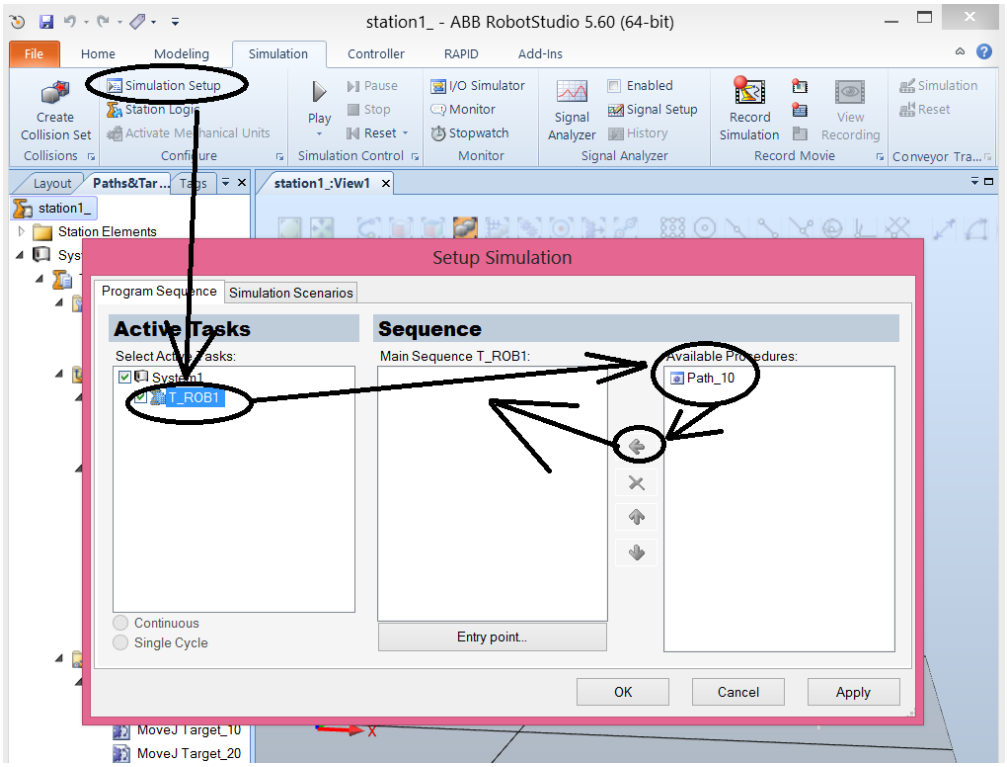
2.7. Simulation

It is time to simulate the robot program we create. First, synchronize with the robot controller (it is like to send all data to the robot controller).

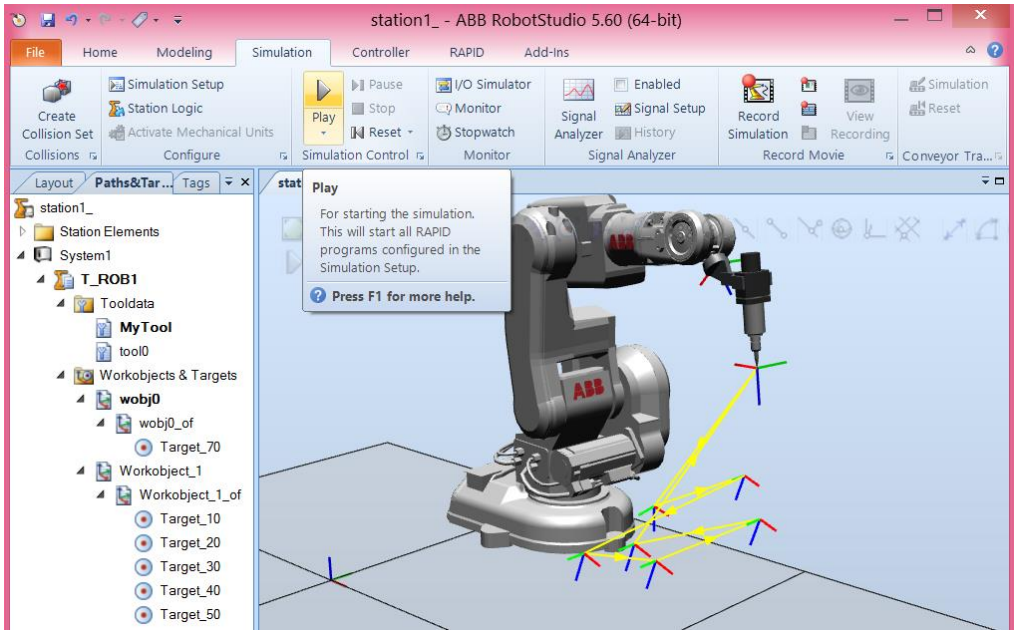




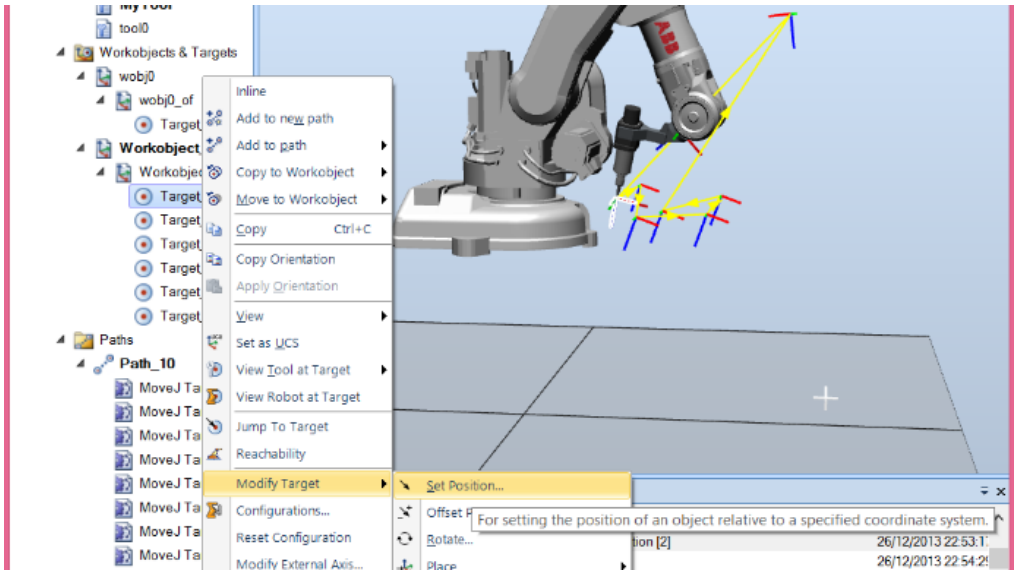
Next, select what paths to simulate, in this case we only have Path_10.



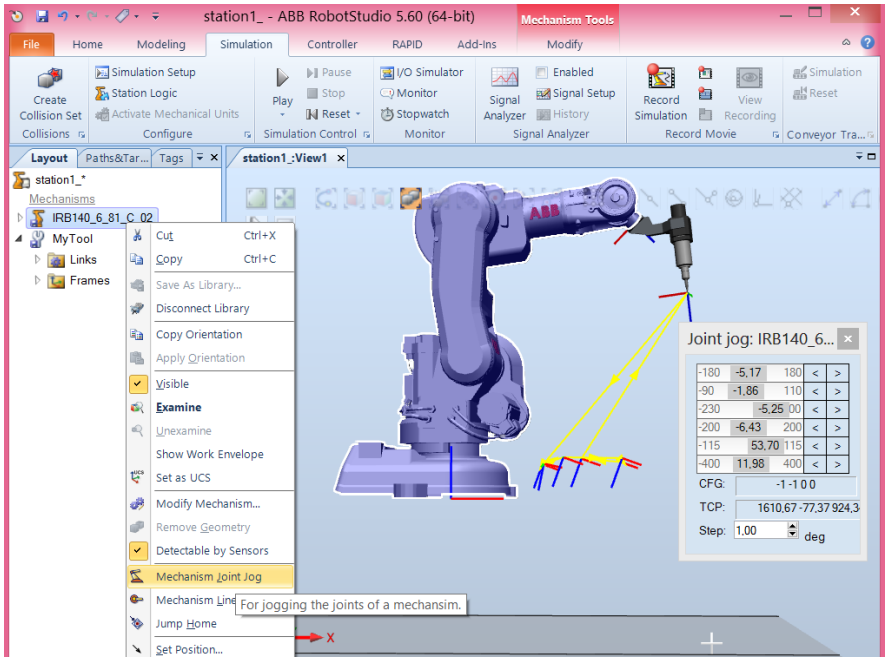
And play:



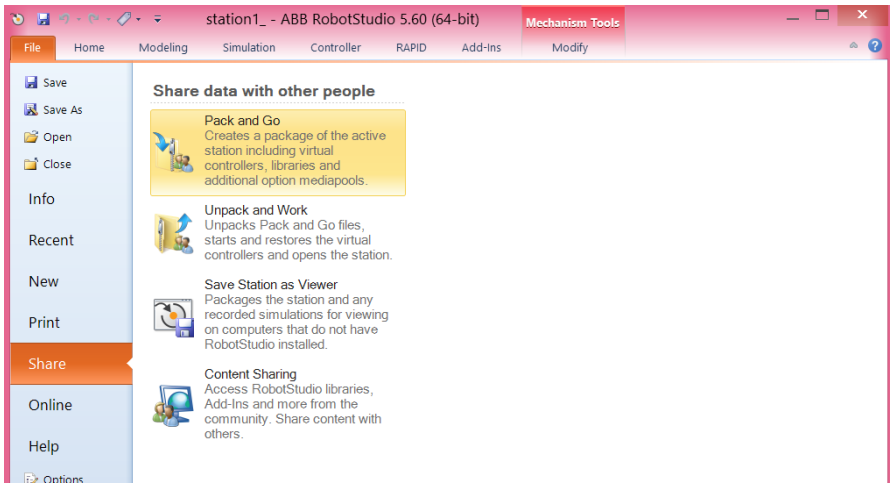
After simulation, if we see that it is necessary to change something, a target for example, we can do that:



But after this, we have to synchronize again. Sometimes, during simulation is useful to see the angles of each joint of the robot:



Save the station in a normal way, or using the pack and go. This last option saves all the project in a folder, so that you can open it in another computer.

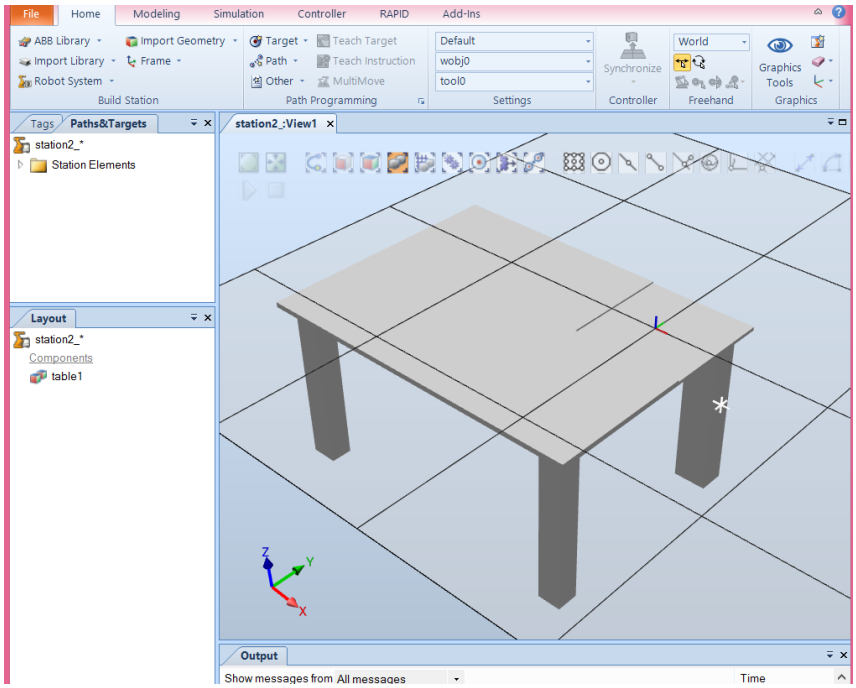


3. MODULE II

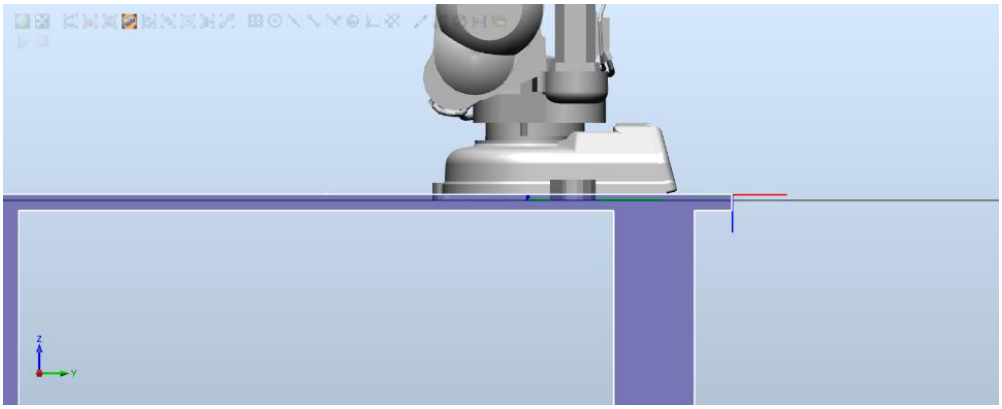
This module complements MODULE I with more functionalities provided by RobotStudio.

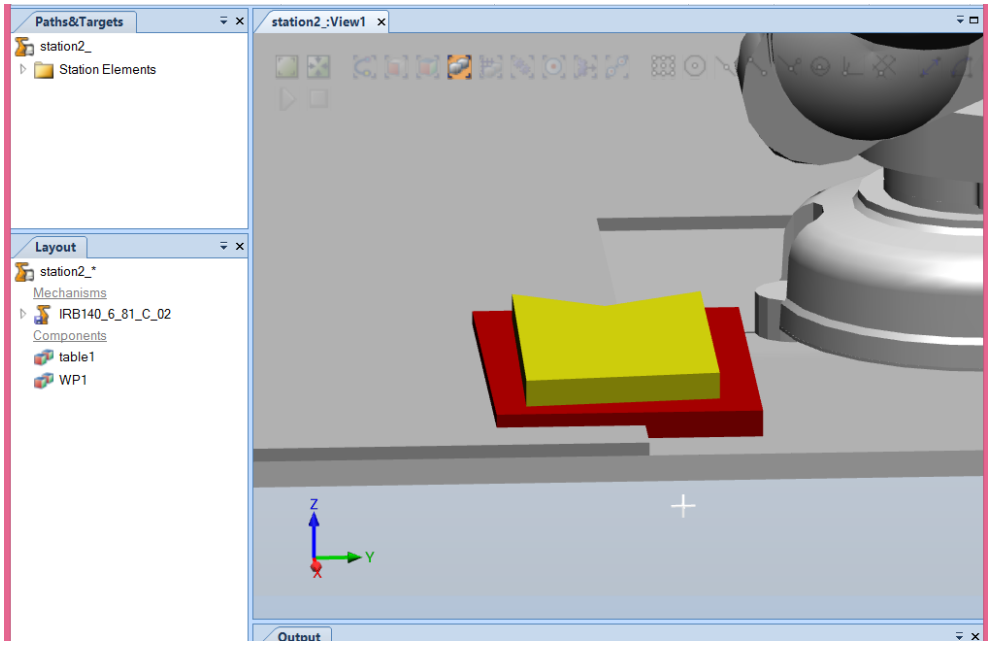
3.1. Importing geometries

In this case we will use not only a robot but also a table and a workpiece. So, RobotStudio allows to import CAD drawings in most of the CAD file formats. Having a drawing of a table, we can insert it into the RobotStudio simulation environment by pressing the button Import Geometry (note that the geometries have to be located in the folder C:\Users\xxxxx\Documents\RobotStudio\Geometry). Using the set position and rotate commands we can place the table in the scenario.

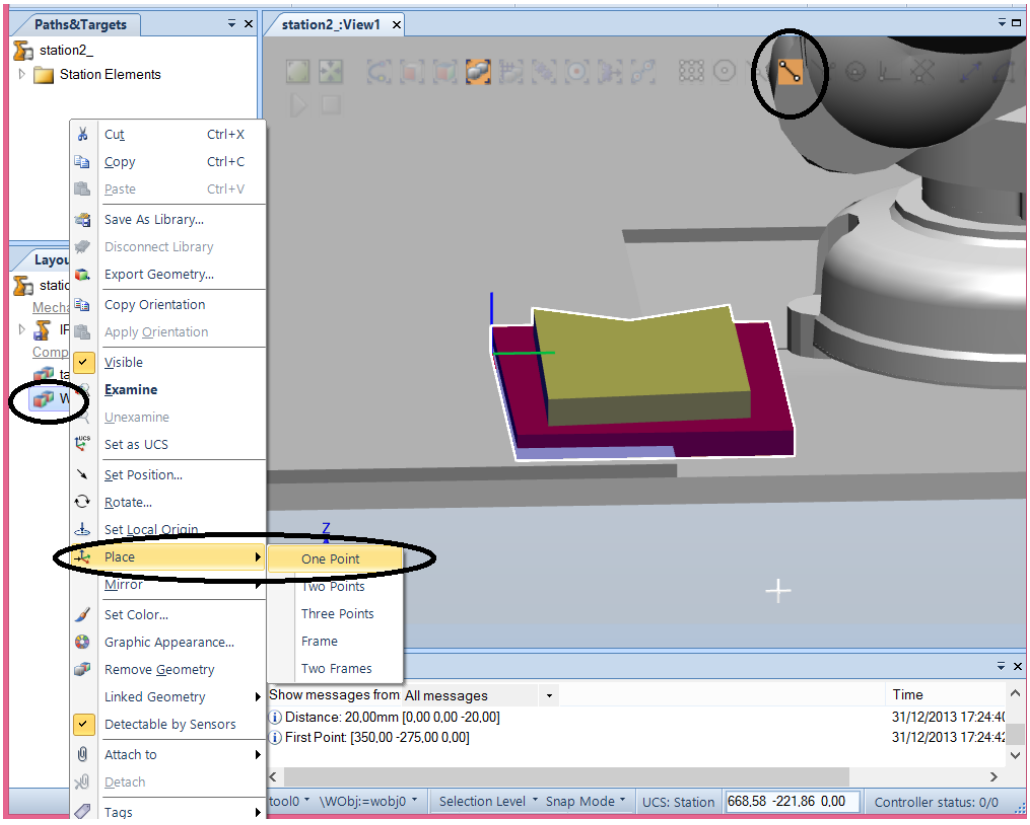


Ok, now we can insert the robotic arm and the workpiece:

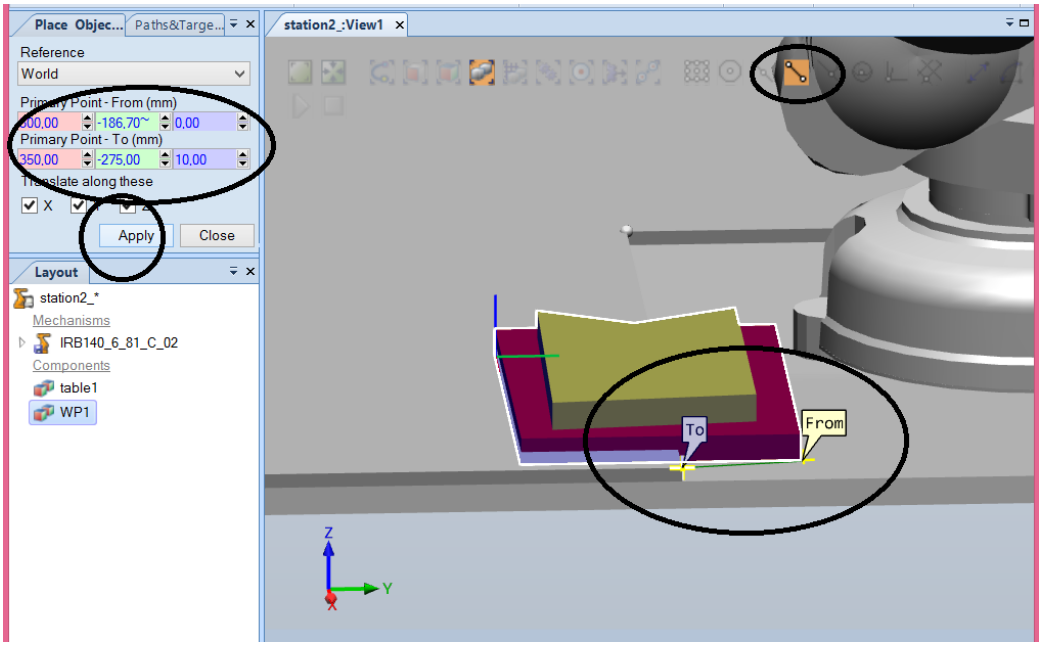




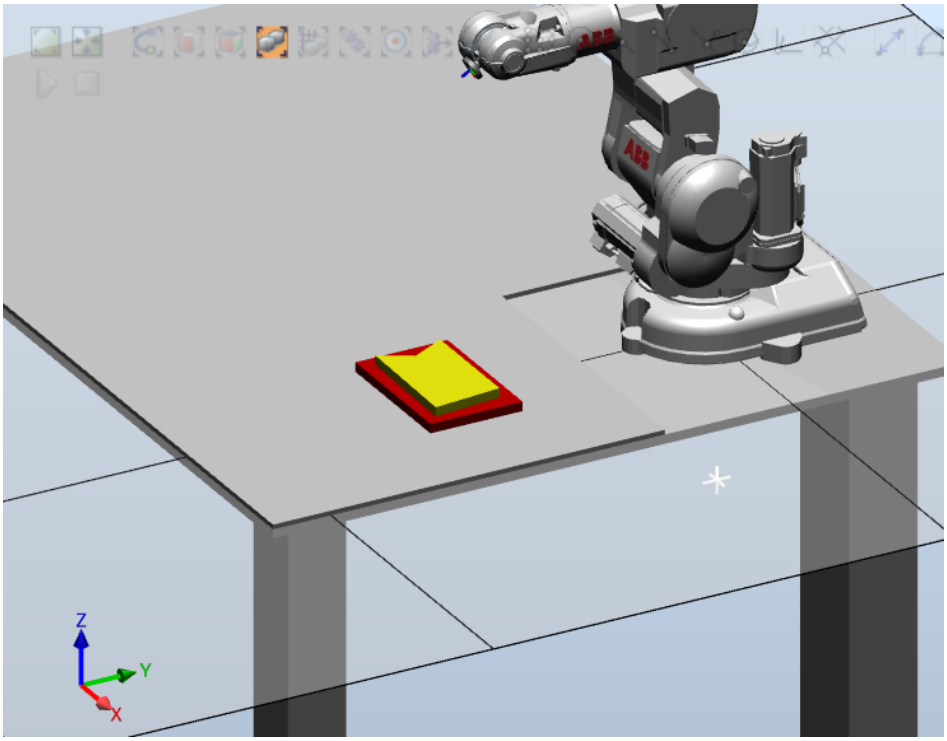
Now, we want to place the workpiece over the table. This can be achieved by selecting points:



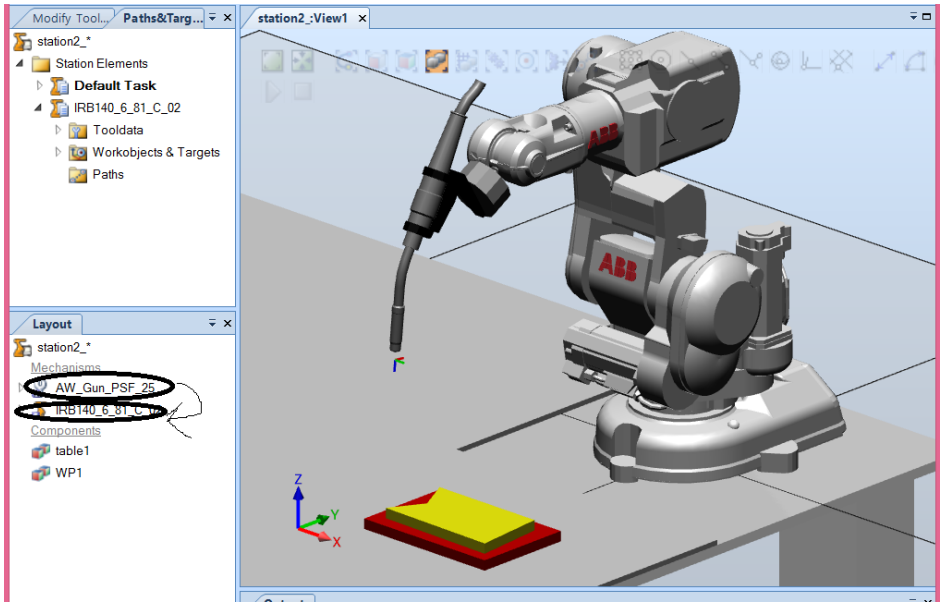
And just click on the desired points:



And by moving along x and y we have this:

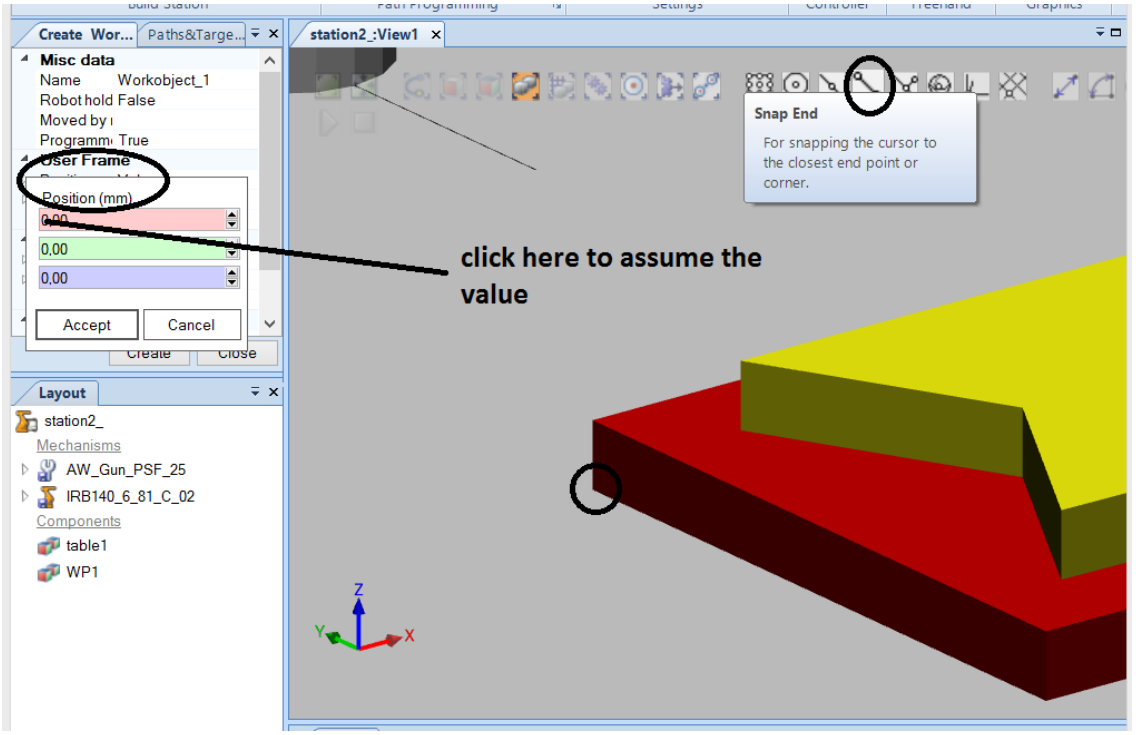


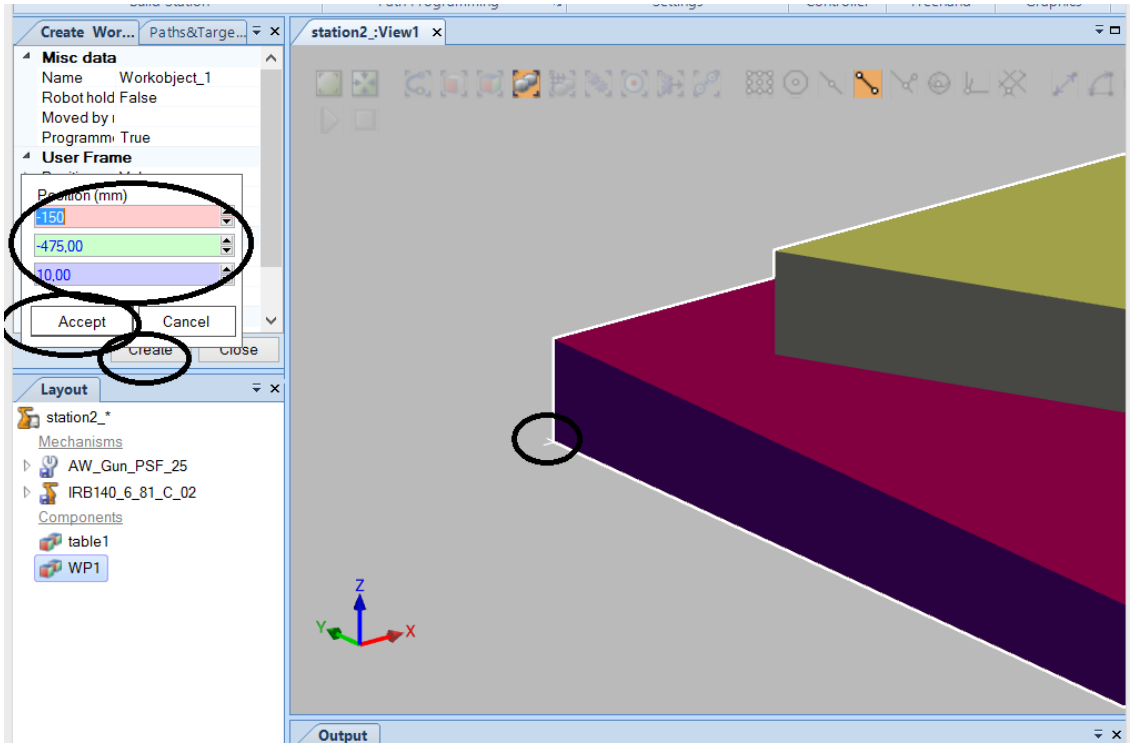
It is time to insert a tool and attach that tool to the robotic arm:



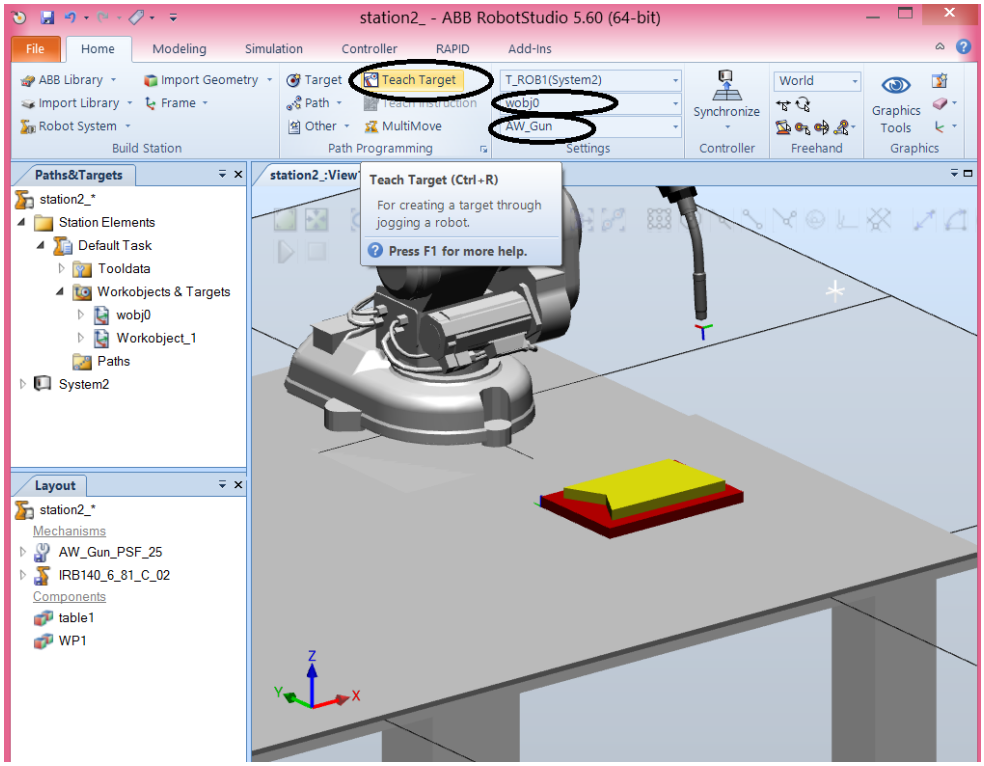
3.2. Workobjects and targets

Create a workobject by selecting its origin with the mouse:

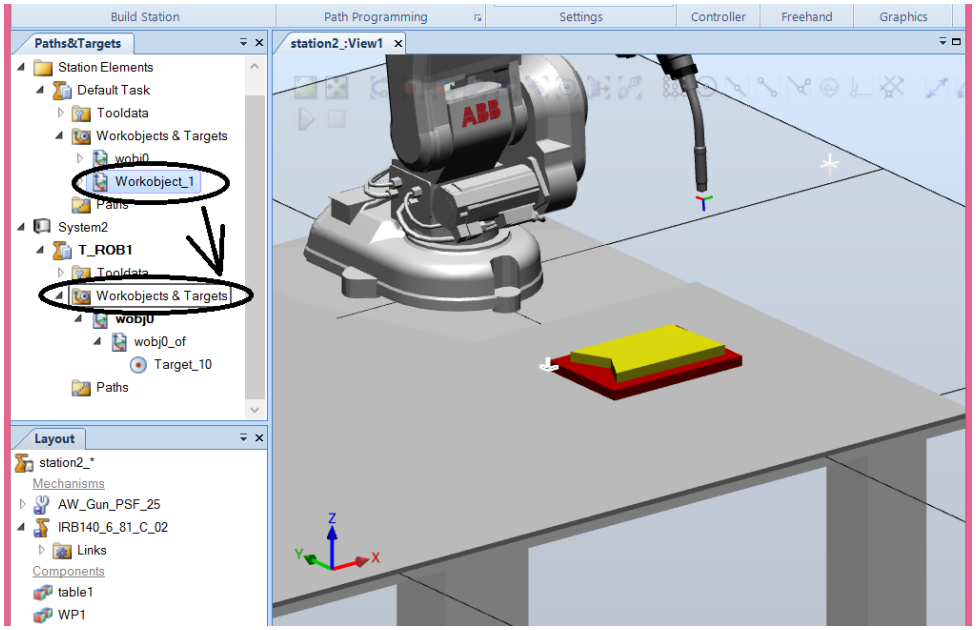




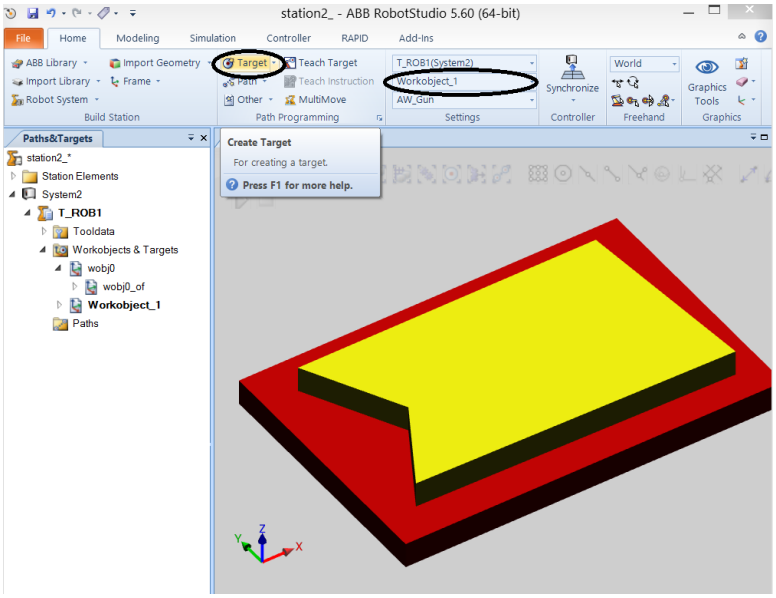
Create a home position for the robot like in MODULE I.

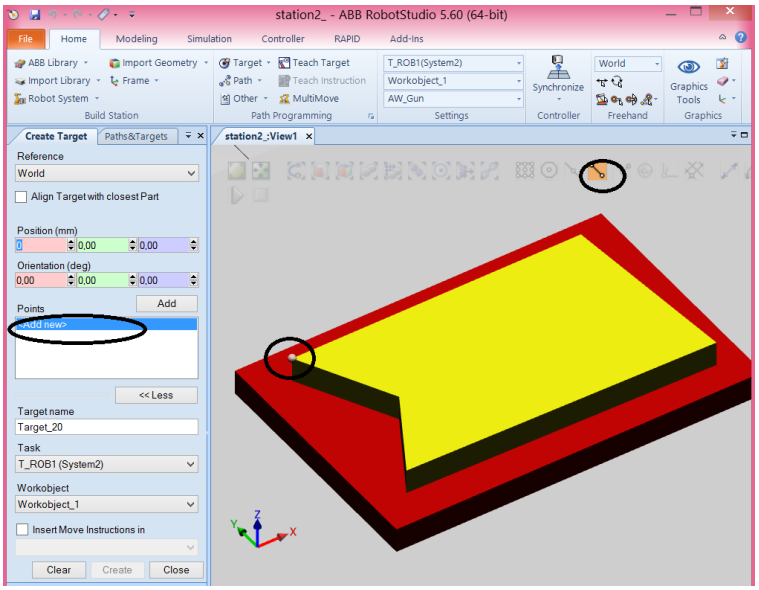


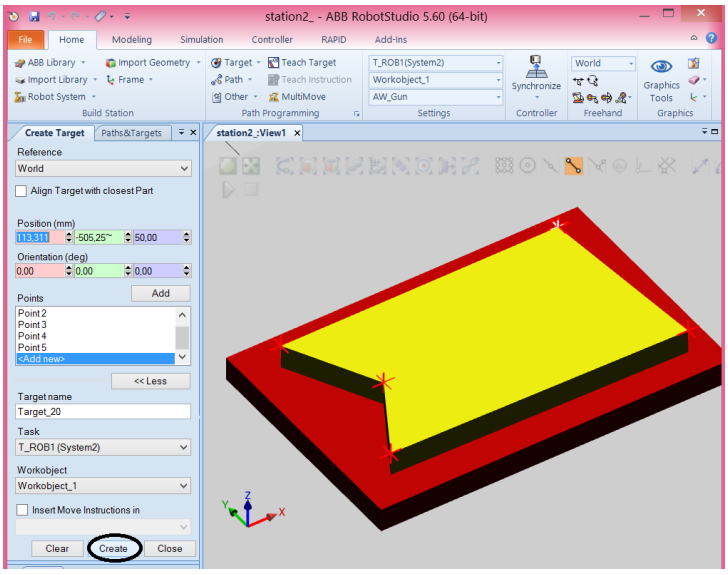
I forgot to create a robot controller, so, we have to do this. The problem is that in the moment we created the workobject the controller was not defined, we have to associate such workobject to the controller. Just drag the workobject to inside the controller and answer yes.



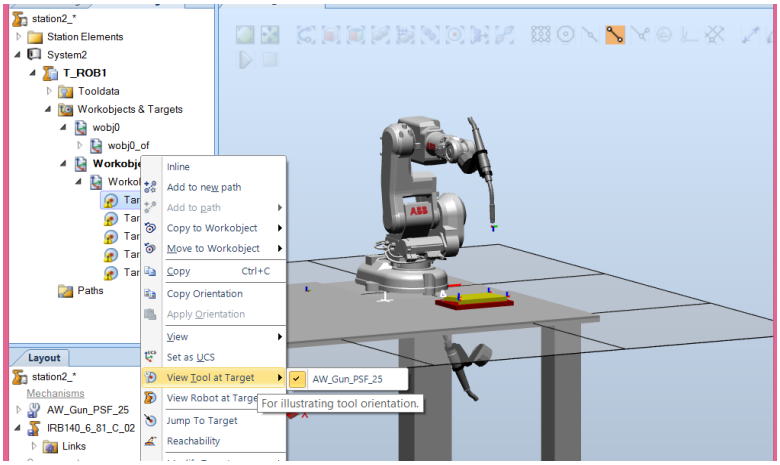
Create target points with reference to workobject:


station2_ - ABB RobotStudio 5.60 (64-bit)
File
Home
Modeling
Simulation
Controller
RAPID
Add-Ins
Target
Teach Target
Workobject_1
Synchronize
Graphics Tools
Paths&Targets
station2_*
Station Elements
System2
T_ROB1
Tooldata
Workobjects & Targets
wobj0
wobj0_of
Workobject_1
Paths
Create Target
For creating a target.
Press F1 for more help.

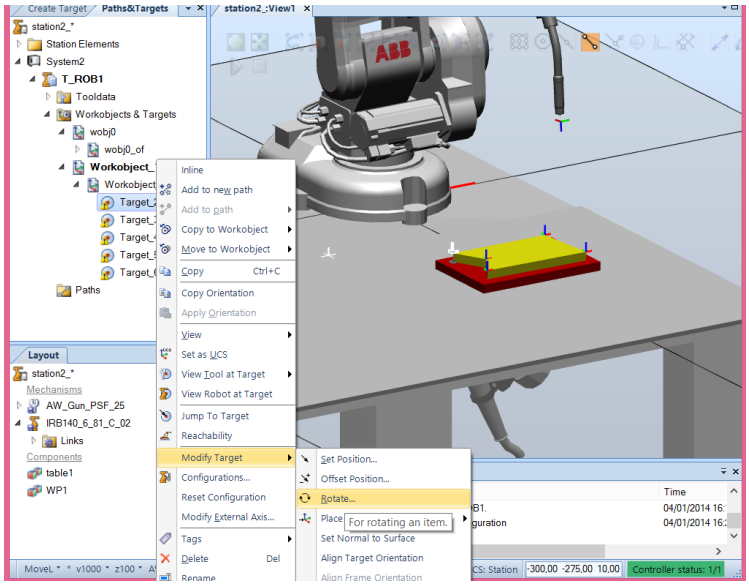

station2_ - ABB RobotStudio 5.60 (64-bit)
Create Target
Paths&Targets
station2_:View1
Reference
World
Align Target with closest Part
Position (mm)
Orientation (deg)
Points
Add
<Add new>
<< Less
Target name
Target_20
Task
T_ROB1 (System2)
Workobject
Workobject_1
Insert Move Instructions in
Clear
Create
Close

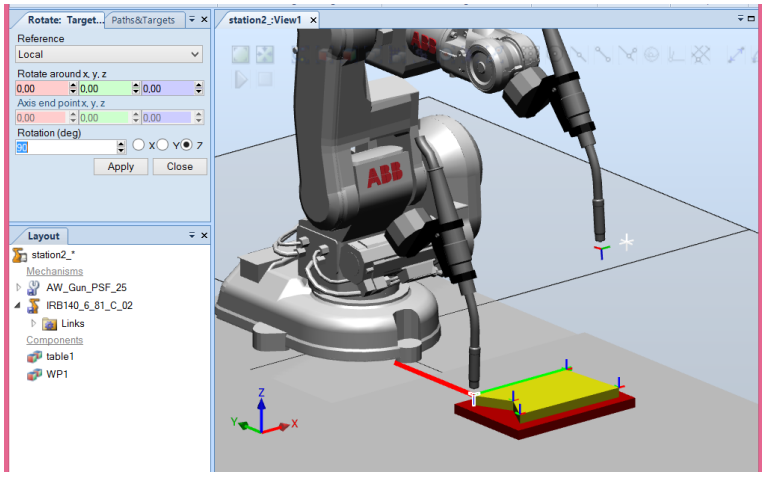

station2_ - ABB RobotStudio 5.60 (64-bit)
Create Target
Paths&Targets
station2_:View1
Reference
World
Align Target with closest Part
Position (mm)
Orientation (deg)
Points
Add
Point 2
Point 3
Point 4
Point 5
<Add new>
<< Less
Target name
Target_20
Task
T_ROB1 (System2)
Workobject
Workobject_1
Insert Move Instructions in
Clear
Create
Close

Now, we can see the tool placed on that target points:

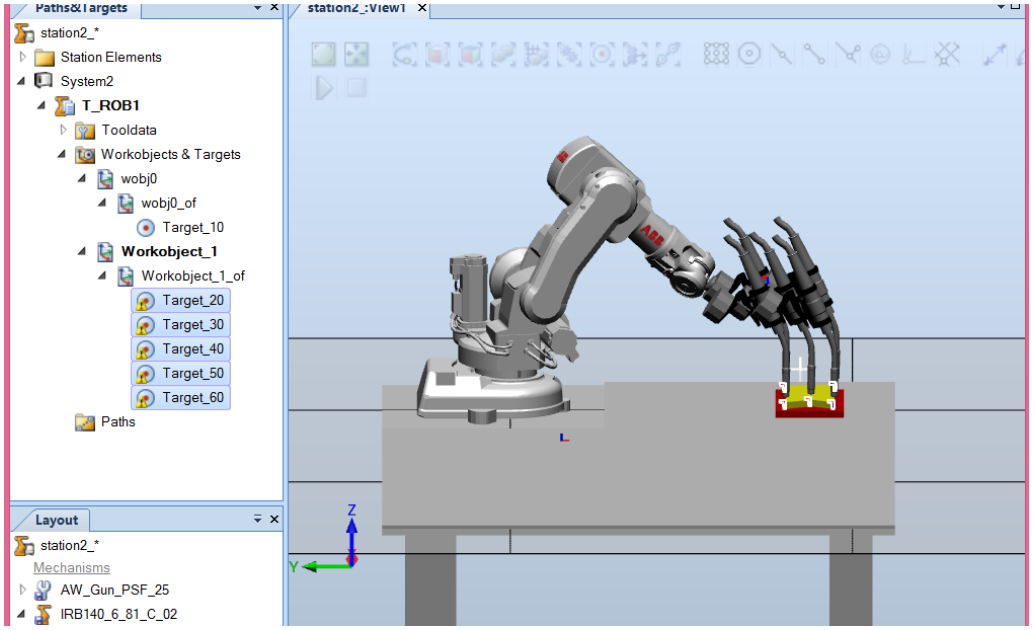


We have to rotate it:

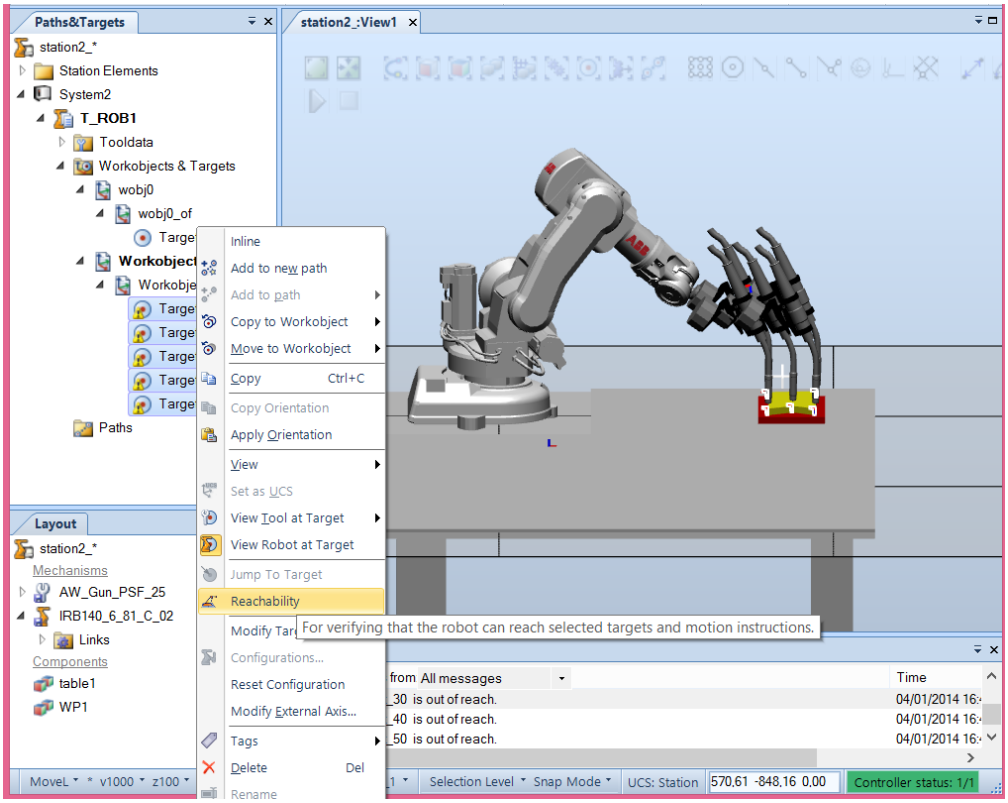


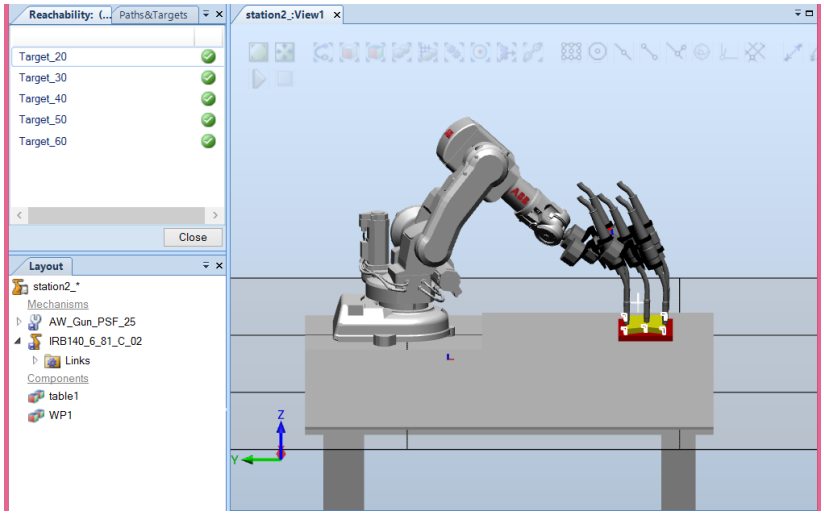


Like in MODULE I we can copy the orientations:

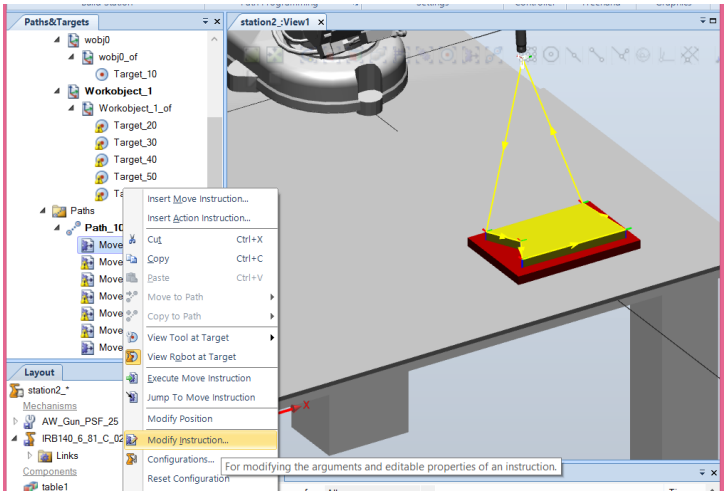


Check the reachability, in other words, check if the robot can reaches the selected points:

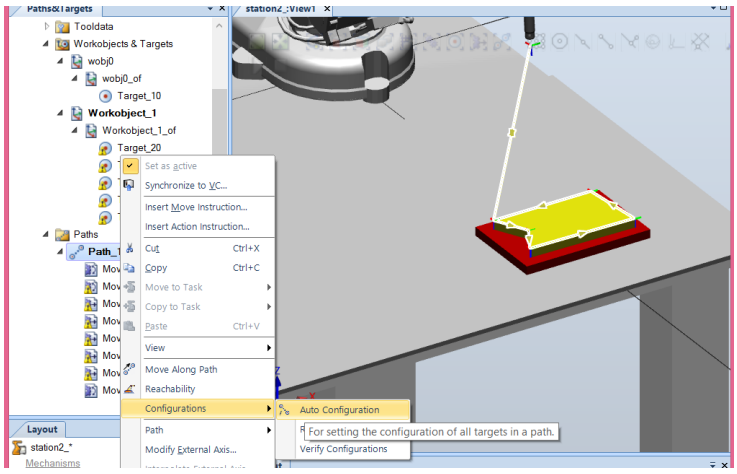




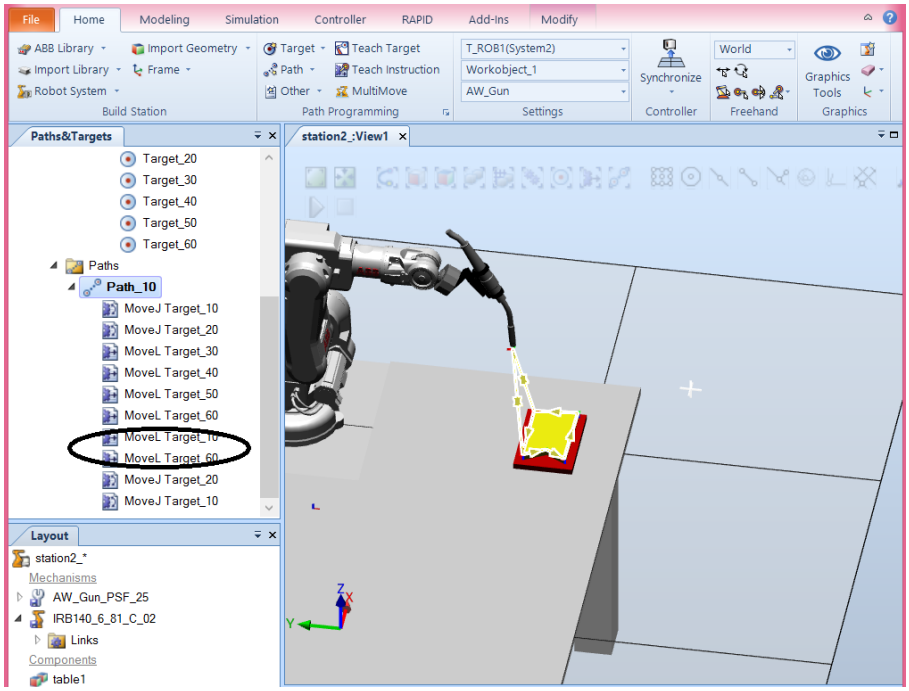
If they are green it is time to create the paths like in MODULE I. After this the instructions can be modified if necessary:



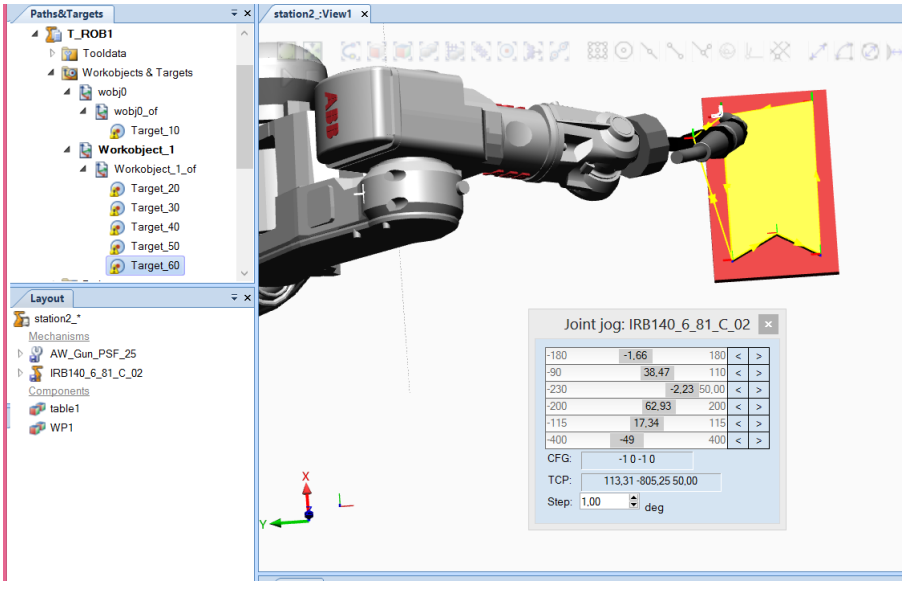
And auto configurate:



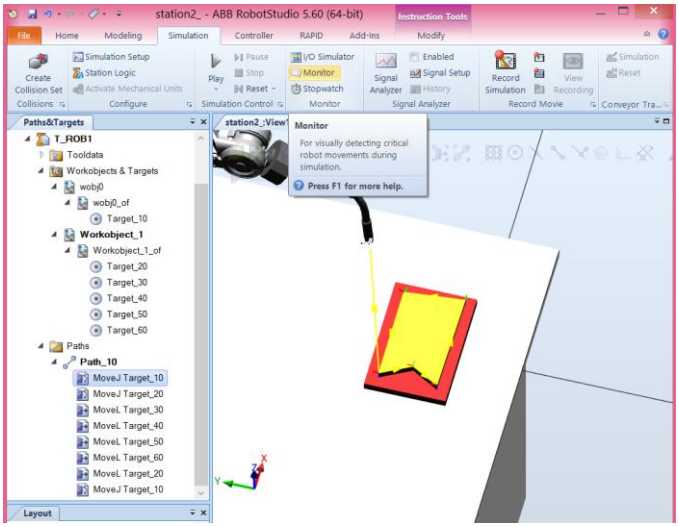
Here a boring situation can happen. You can have some warnings on it, so that you have to change the orientation of each target until to achieve a good solution.

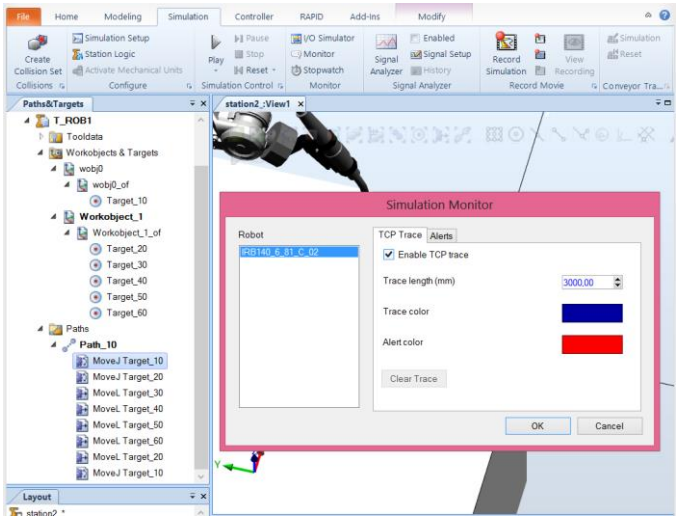


We can look to the robot axis to see what is the robot joint near the limit:

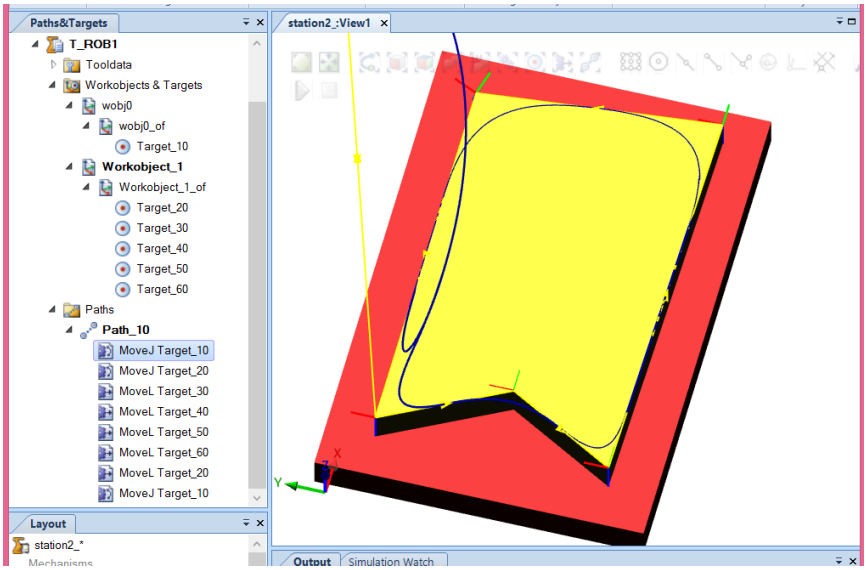


Synchronize, add Path_10 in simulation setup and simulate. To see the path line select the monitor functionality:





And simulate:



You can see that the TCP do not reaches the target points. This is because we are using a zone of 100. Select fine (synchronize again) and the robot will reaches that target points.

Data types

zonedata - Zone data

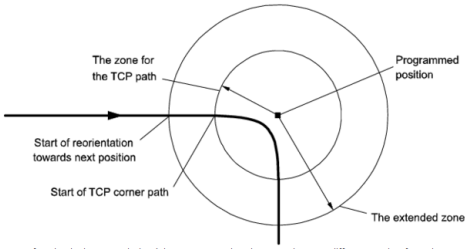
Usage `zonedata` is used to specify how a position is to be terminated, i.e. how close to the programmed position the axes must be before moving towards the next position.

Description A position can be terminated either in the form of a stop point or a fly-by point.

A stop point means that the robot and external axes must reach the specified position (stand still) before program execution continues with the next instruction. It is also possible to define stop points other than the predefined `fine`. The stop criteria, that tells if the robot is considered to have reached the point, can be manipulated using the `stoppointdata`.

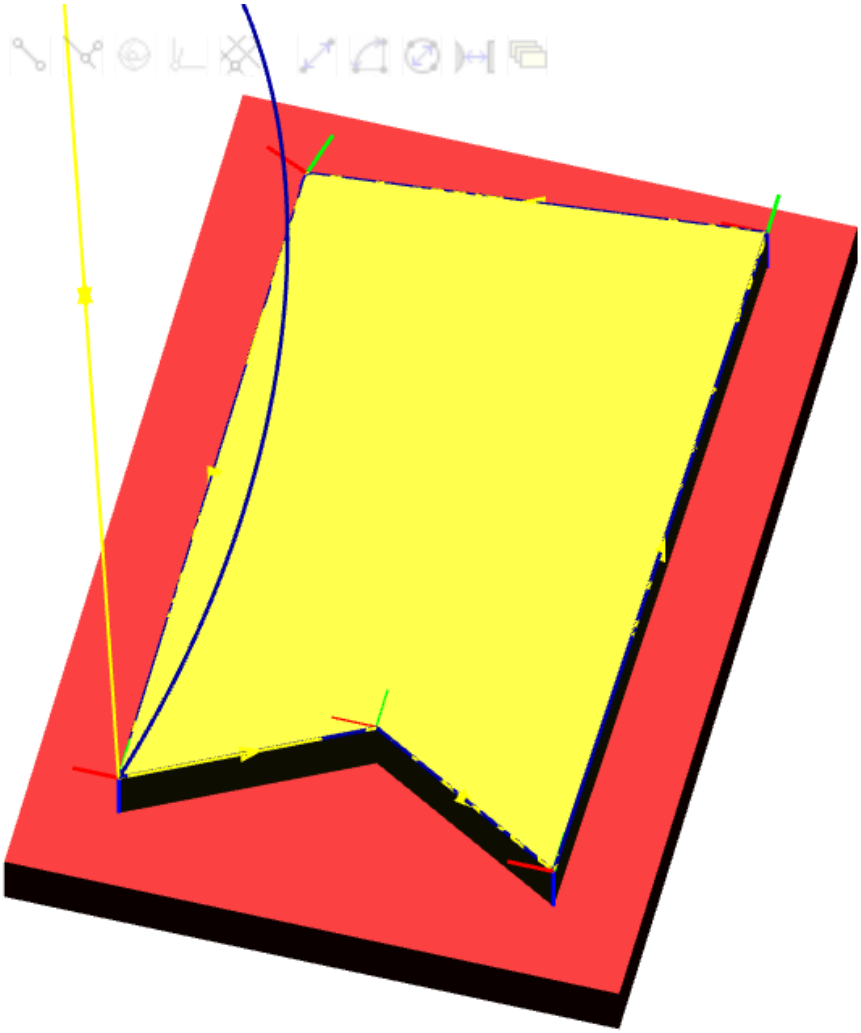
A fly-by point means that the programmed position is never attained. Instead, the direction of motion is changed before the position is reached. Two different zones (ranges) can be defined for each position:

- The zone for the TCP path.
- The extended zone for reorientation of the tool and for external axes.

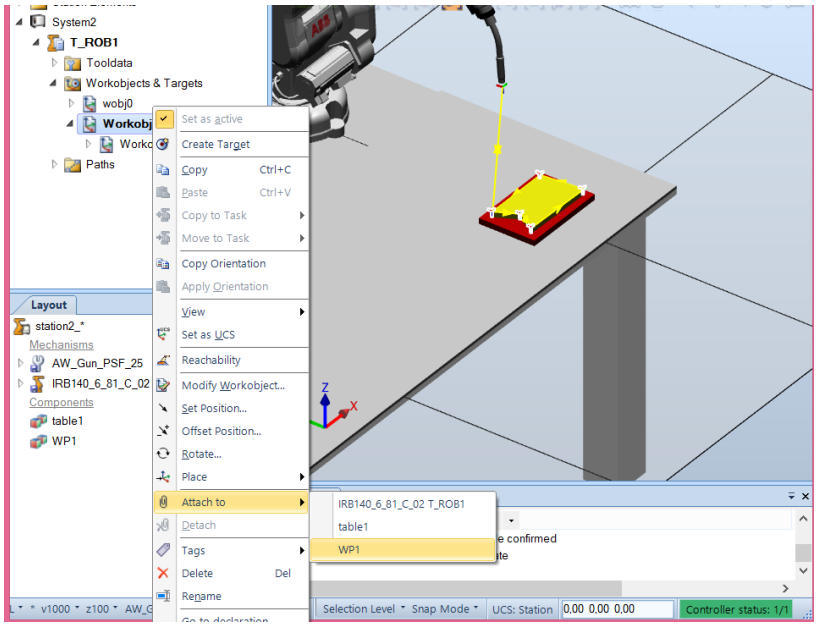


Zones function is the same during joint movement, but the zone size may differ somewhat from the one programmed.

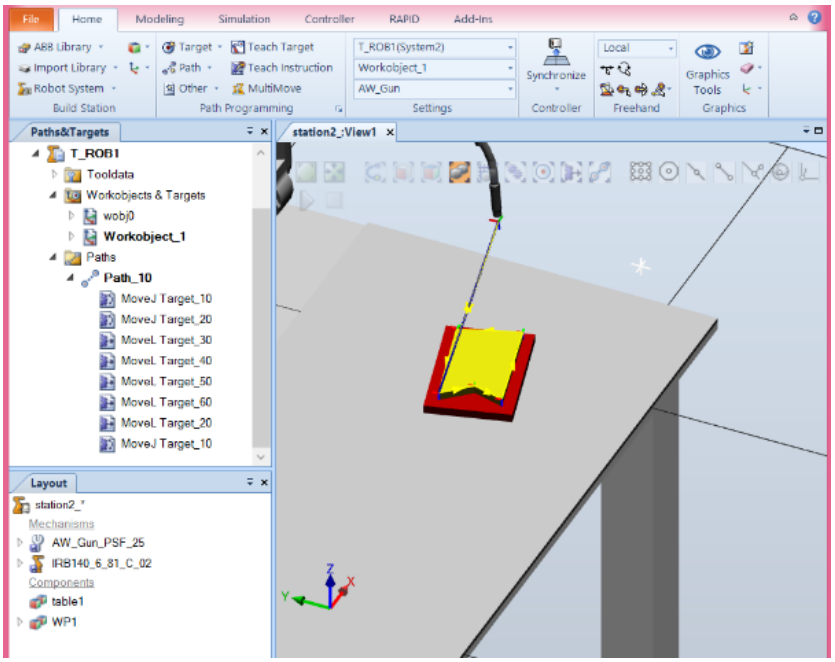
The zone size cannot be larger than half the distance to the closest position (forwards or backwards). If a larger zone is specified, the robot automatically reduces it.



After this, we may need to change the workpiece pose with the attached points:

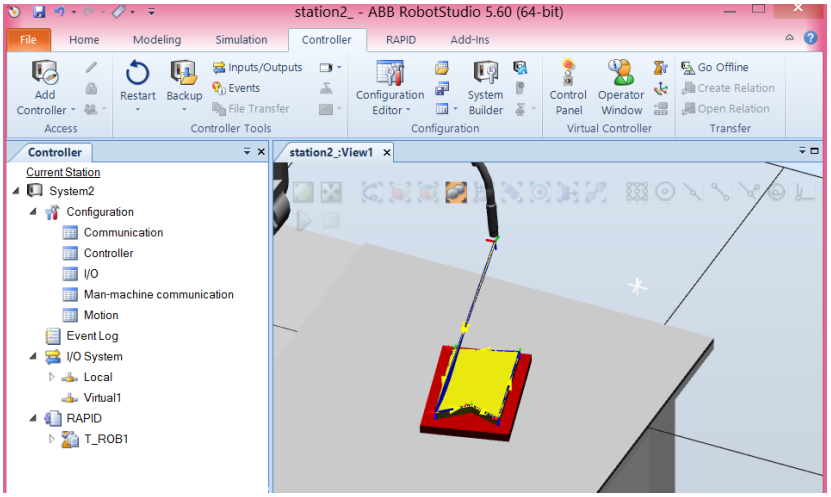


And we can now move the workpiece with associated target points:

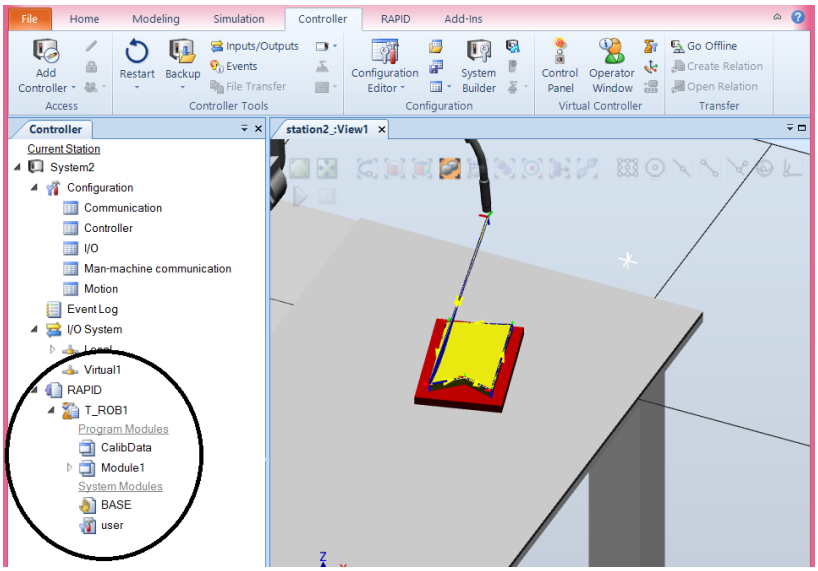


3.3. RAPID

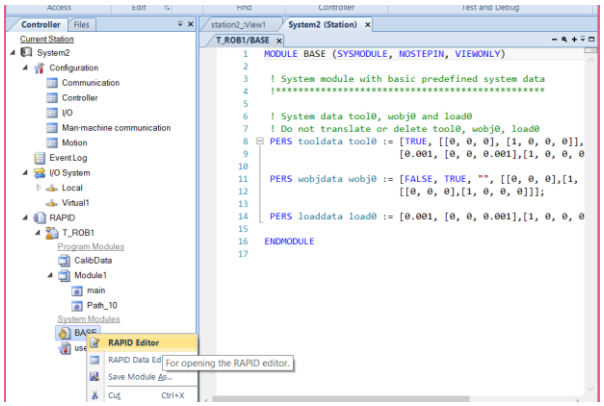
Go to the tab controller:



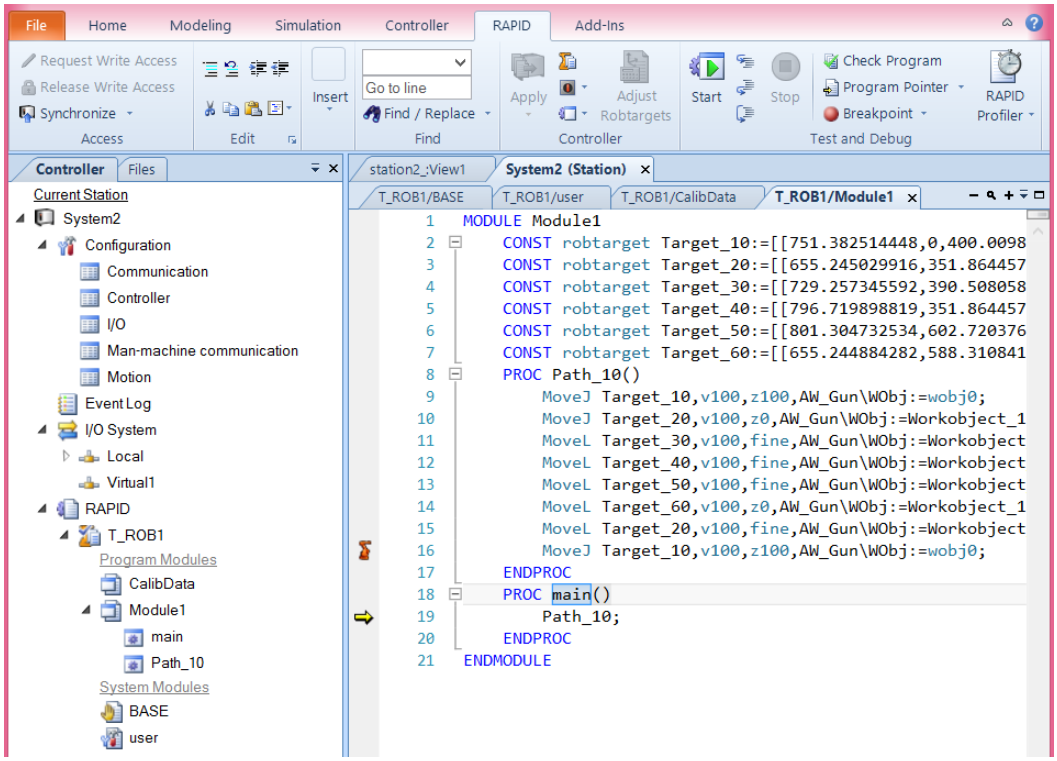
And in RAPID section we have the generated programs (see the presentation about RAPID):



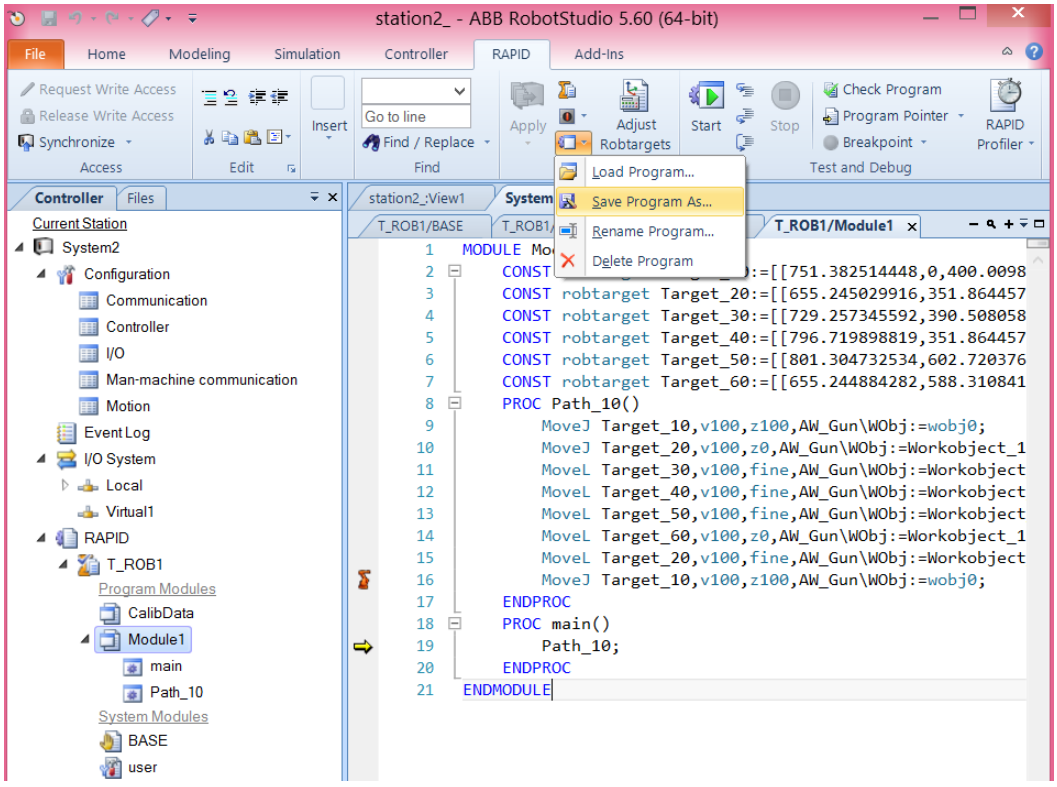
We can see what is inside these modules:



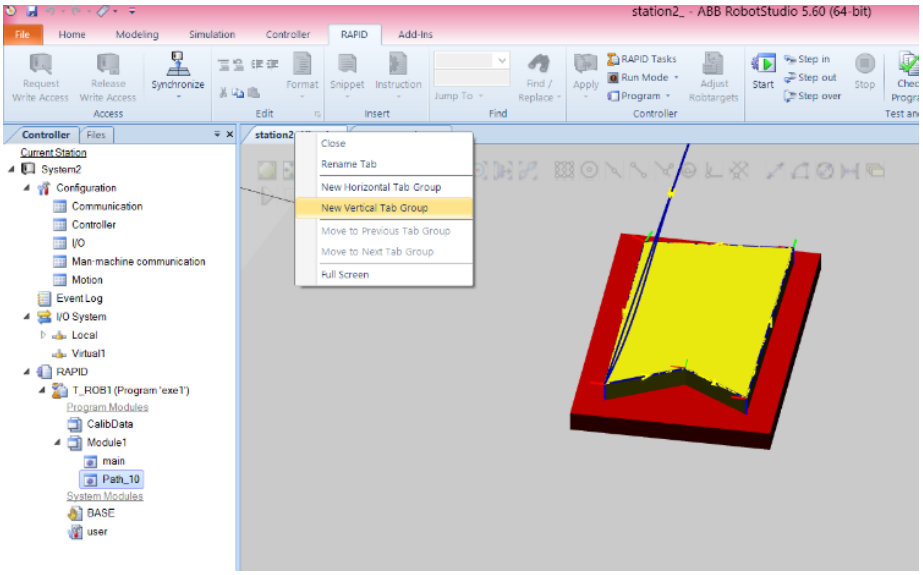
But the important is Module 1:

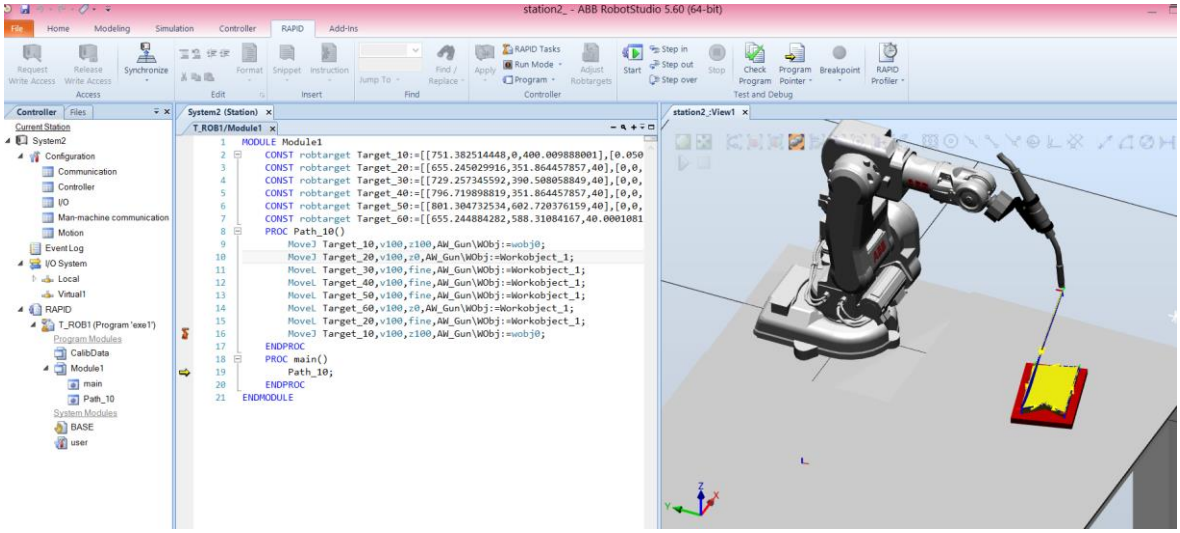


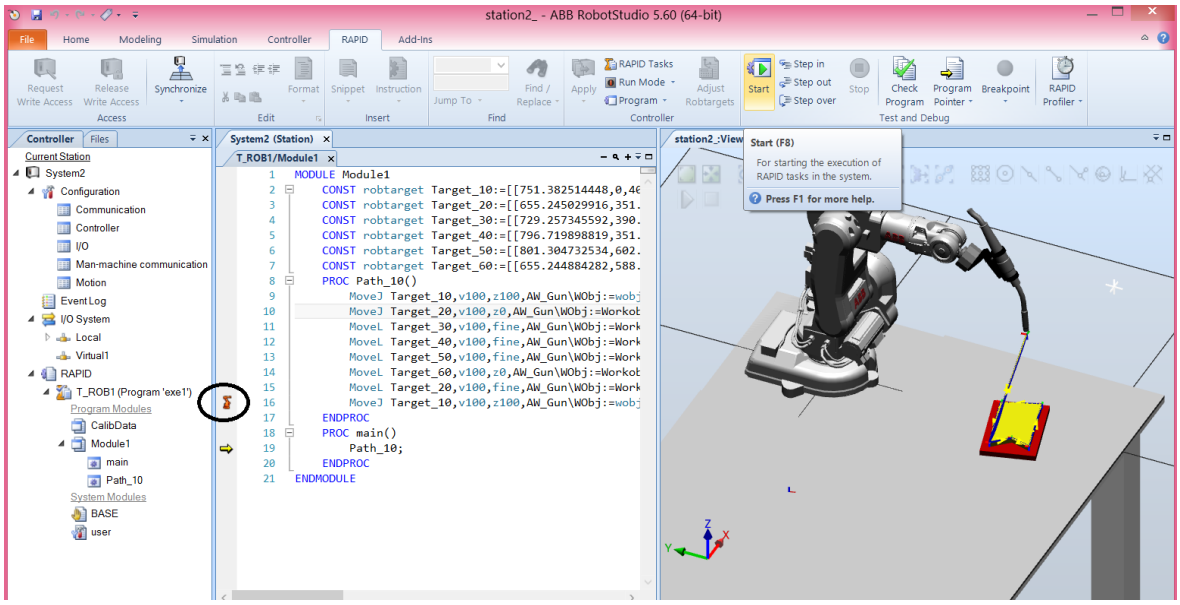
It is possible to save the program:



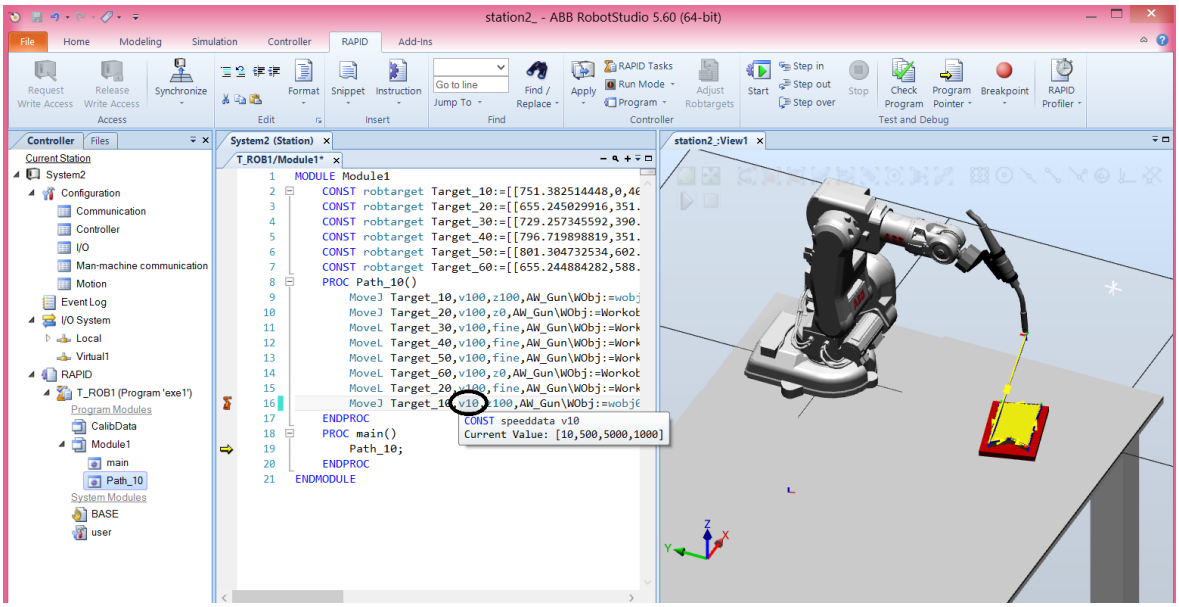
And see the robot simulation and running code at the same time:



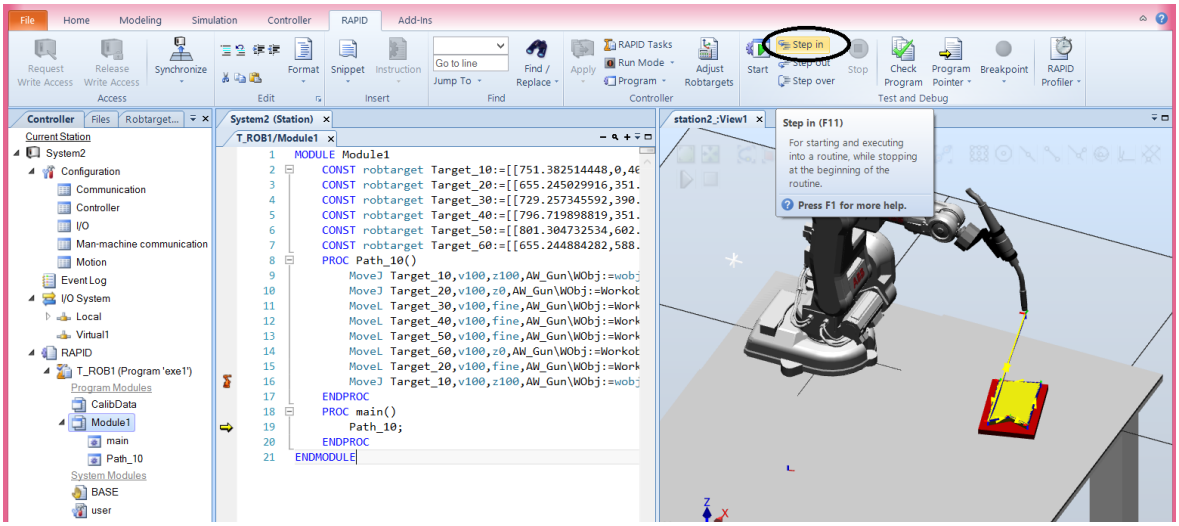




Now, we can edit the program, for example changing speed:

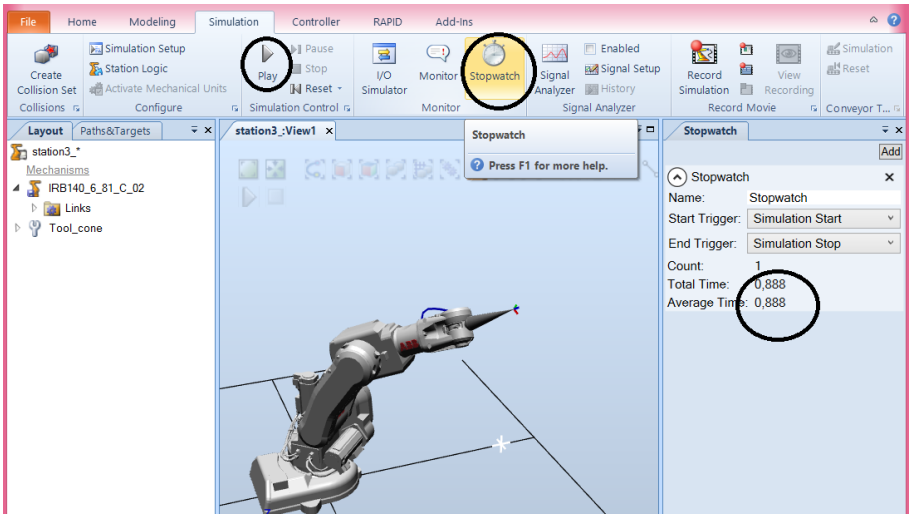


And we can simulate it step-by-step:



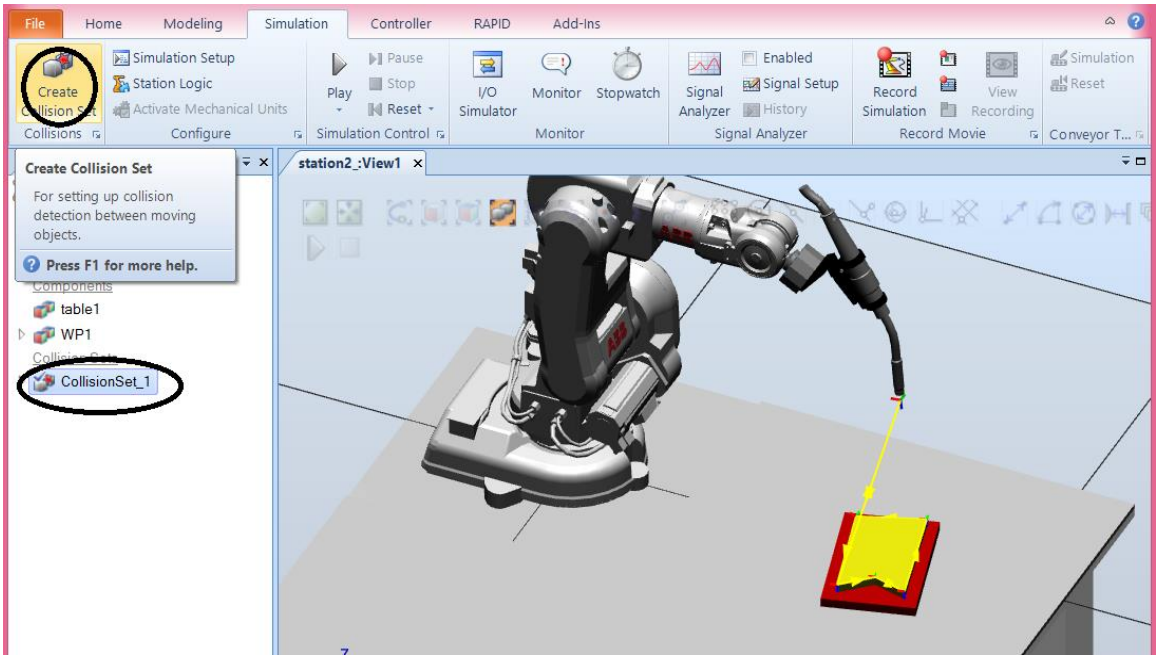
3.4. Time

In simulation tab select the watch to see how much time a robotic process takes in simulation:

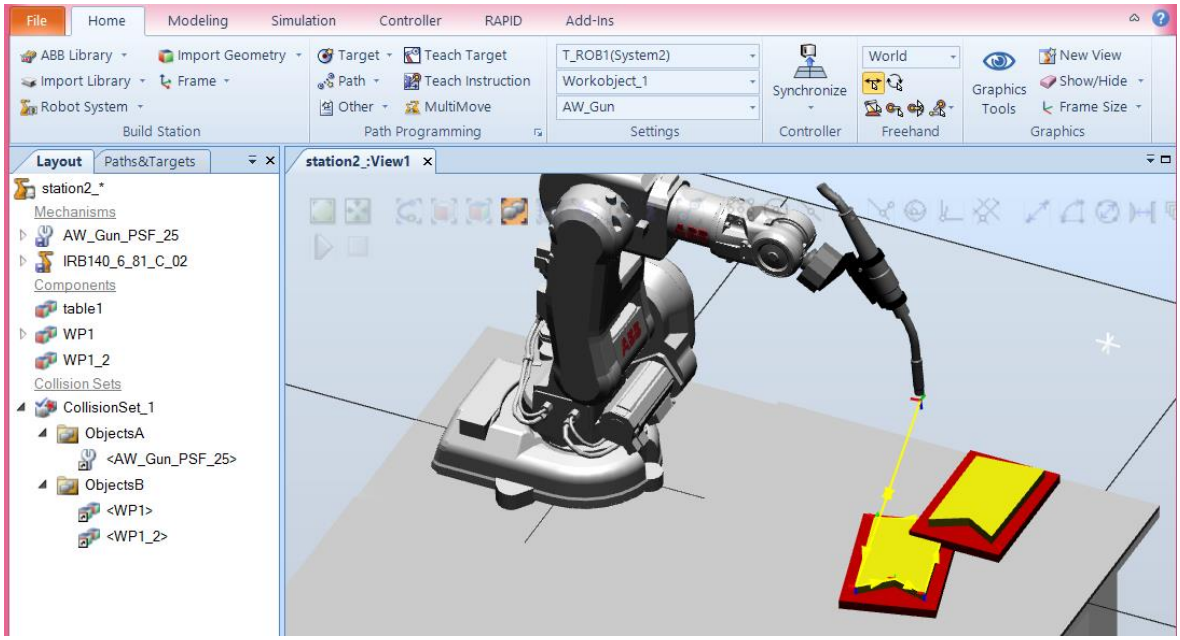


3.5. Collisions

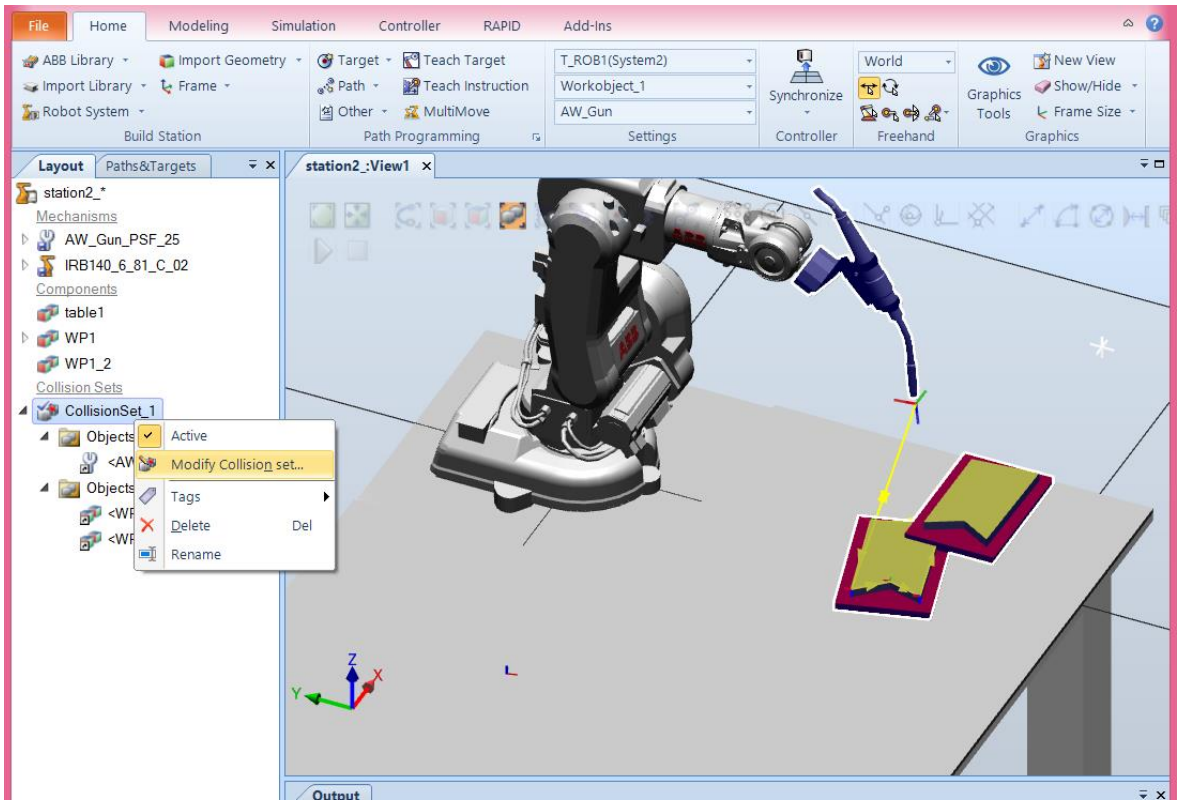
In the simulation tab press the button create collision:

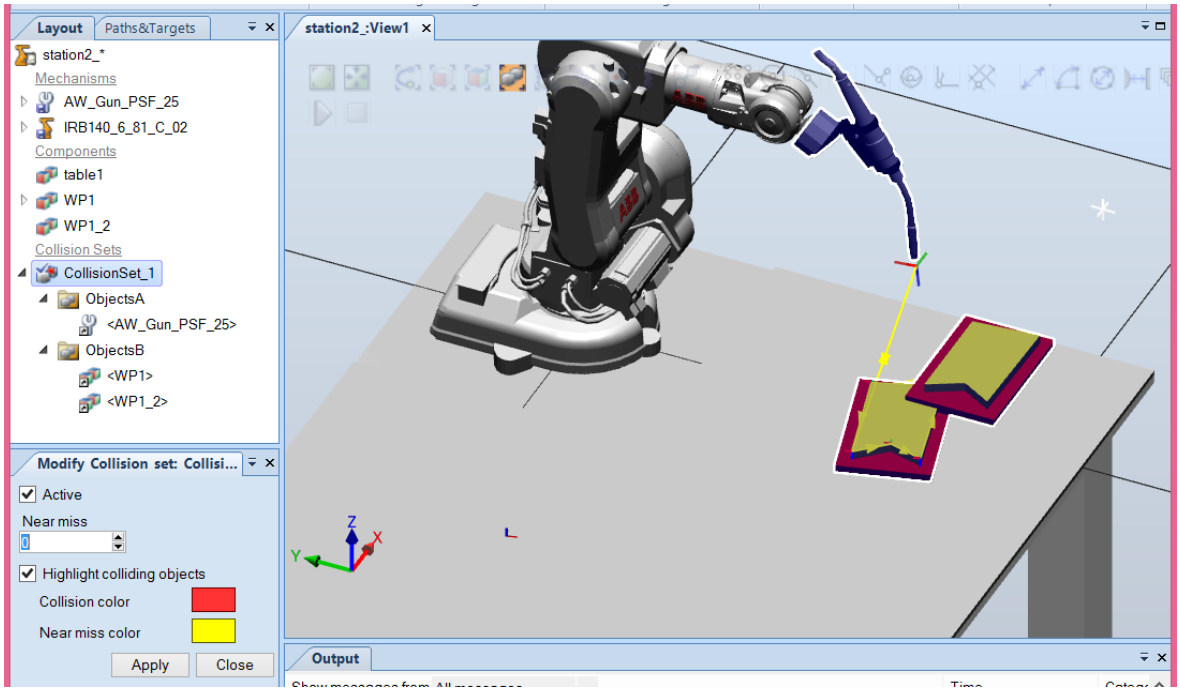


Open the collision object and drag the tool to ObjectsA and the workpieces to ObjectsB. The software analyses collisions between objects type A and B.

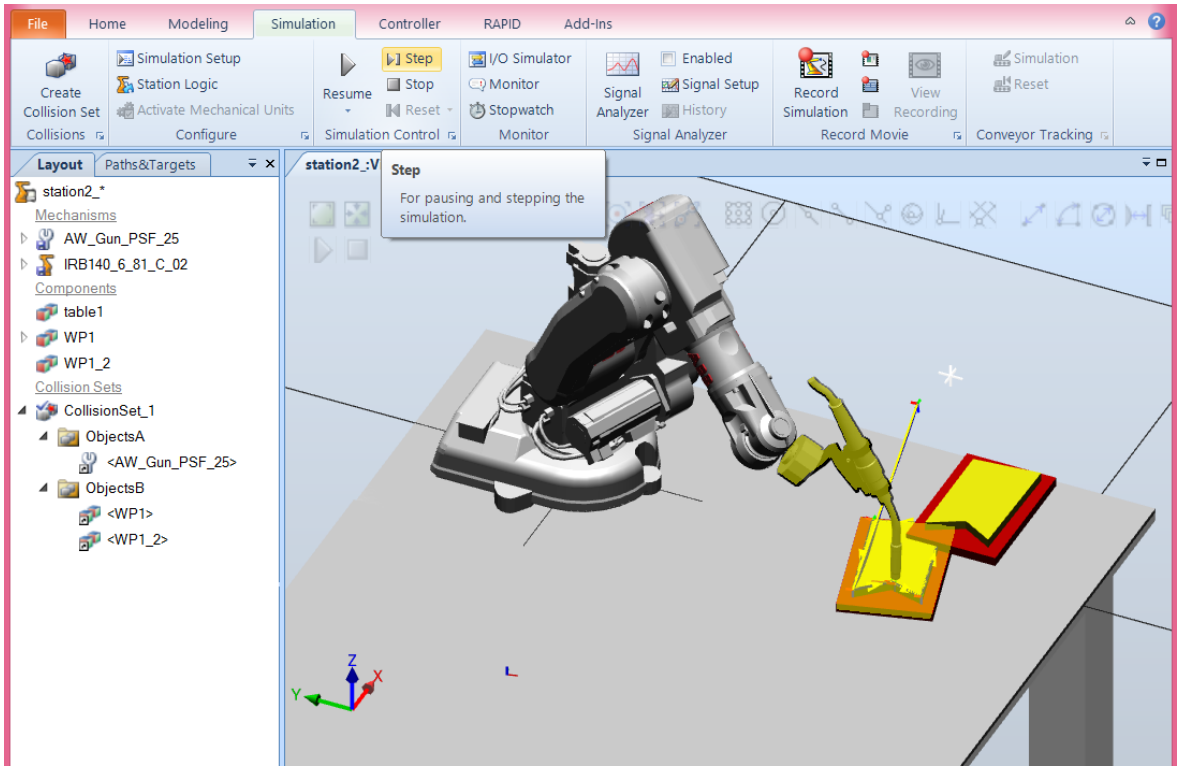


You can also define a near collision distance:



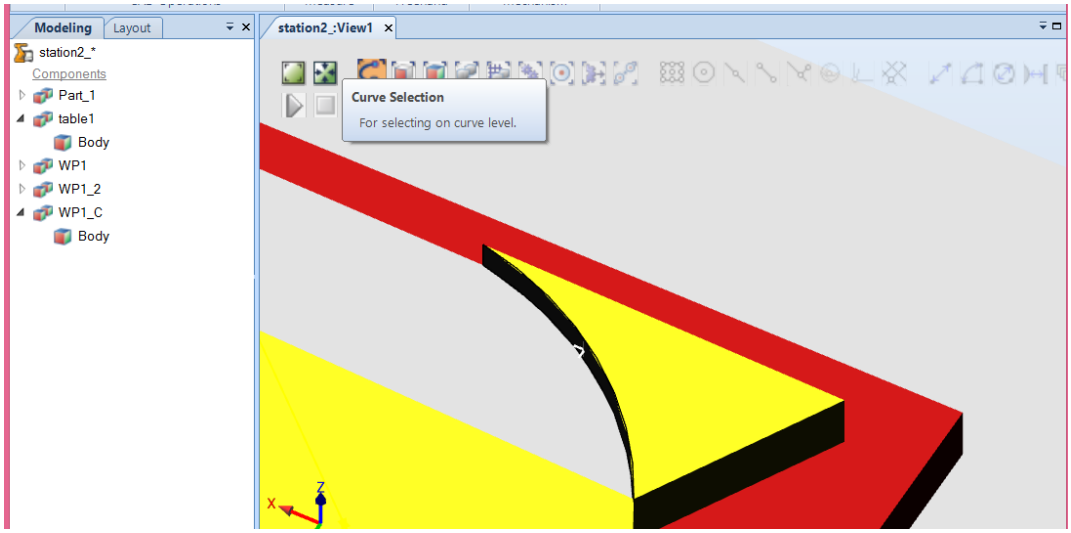


And simulate:

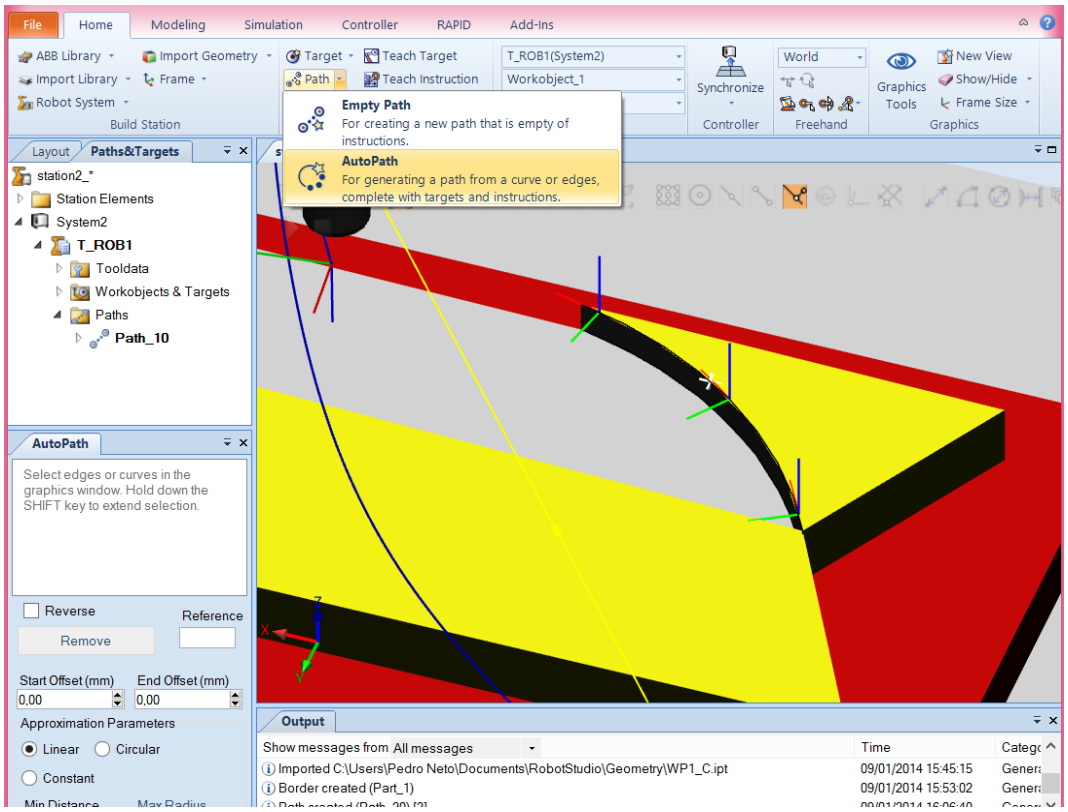


3.6. Curves

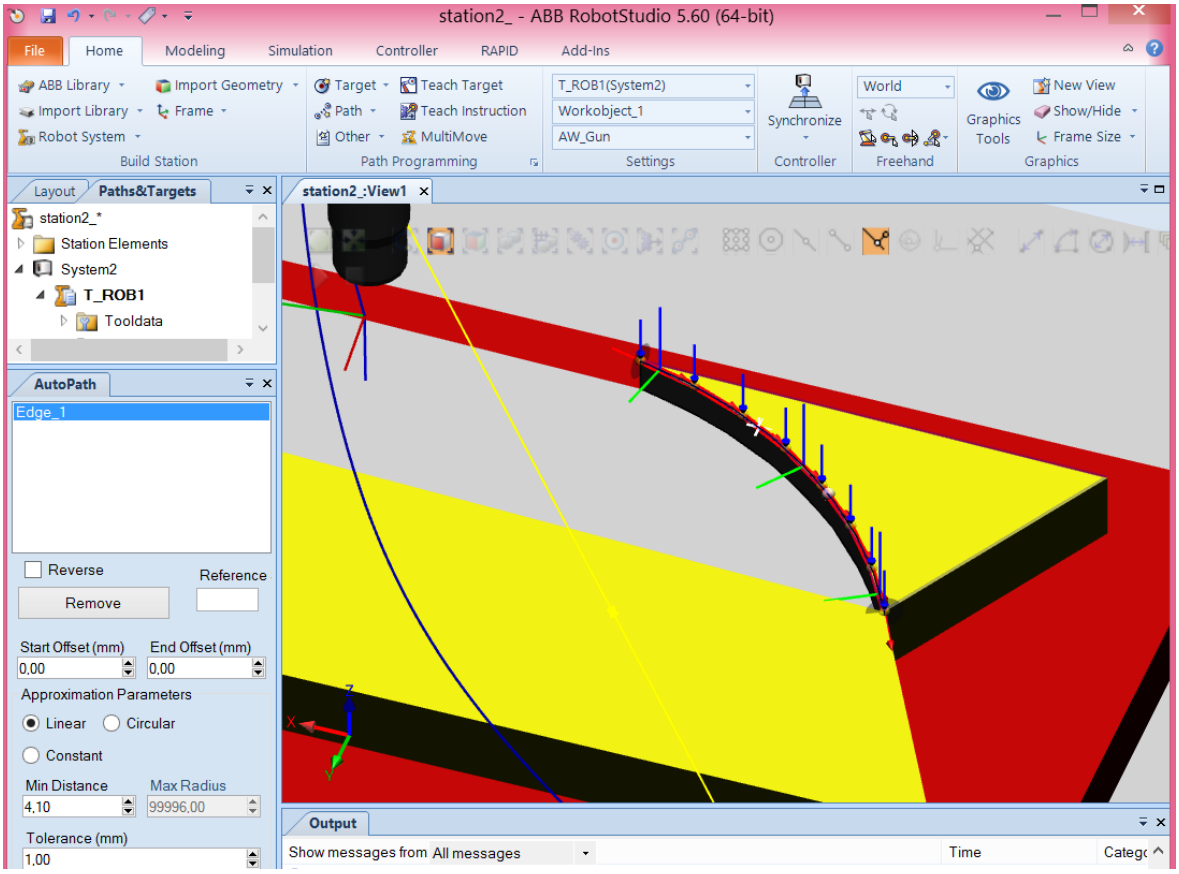
Include a new workpiece with a curve. Select Curve Selection:

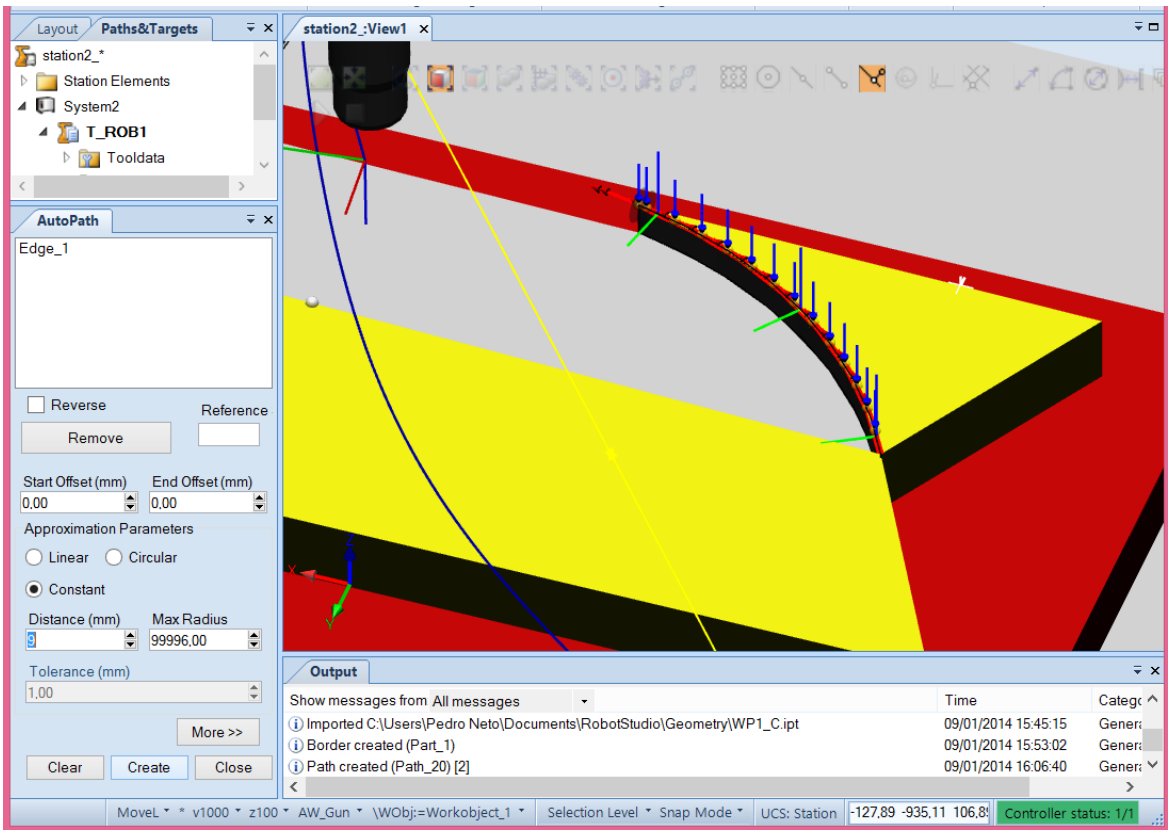


Select AutoPath:

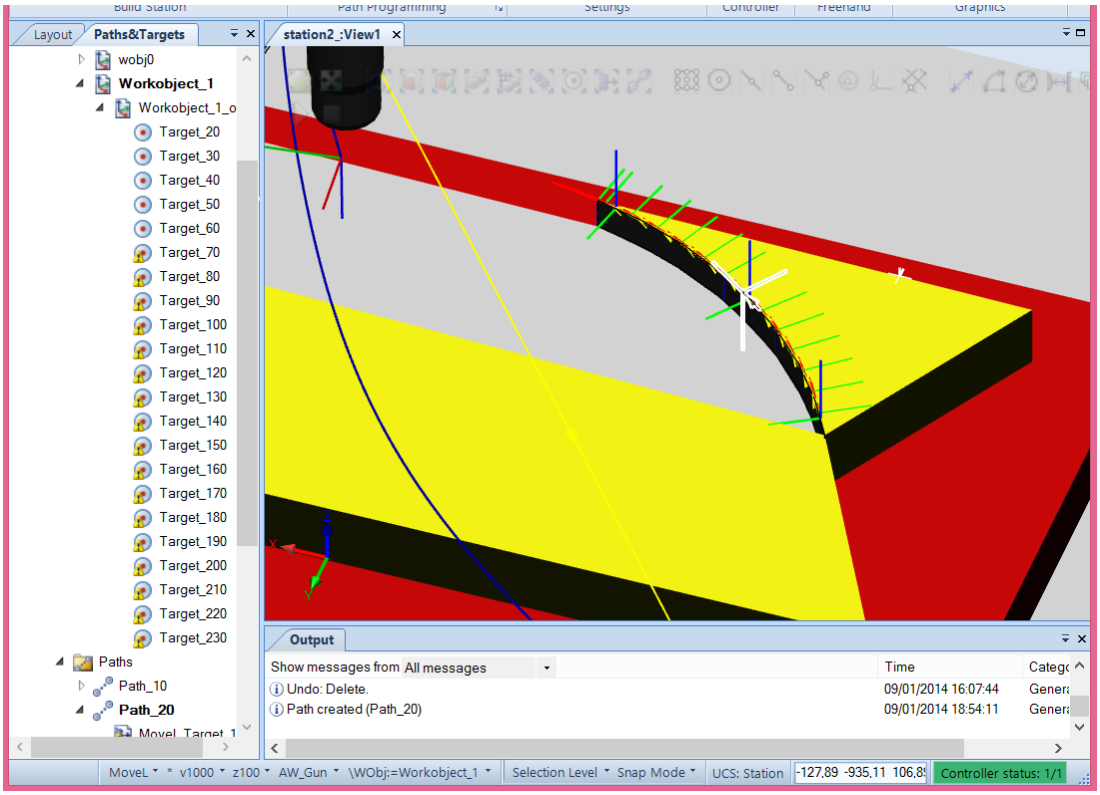


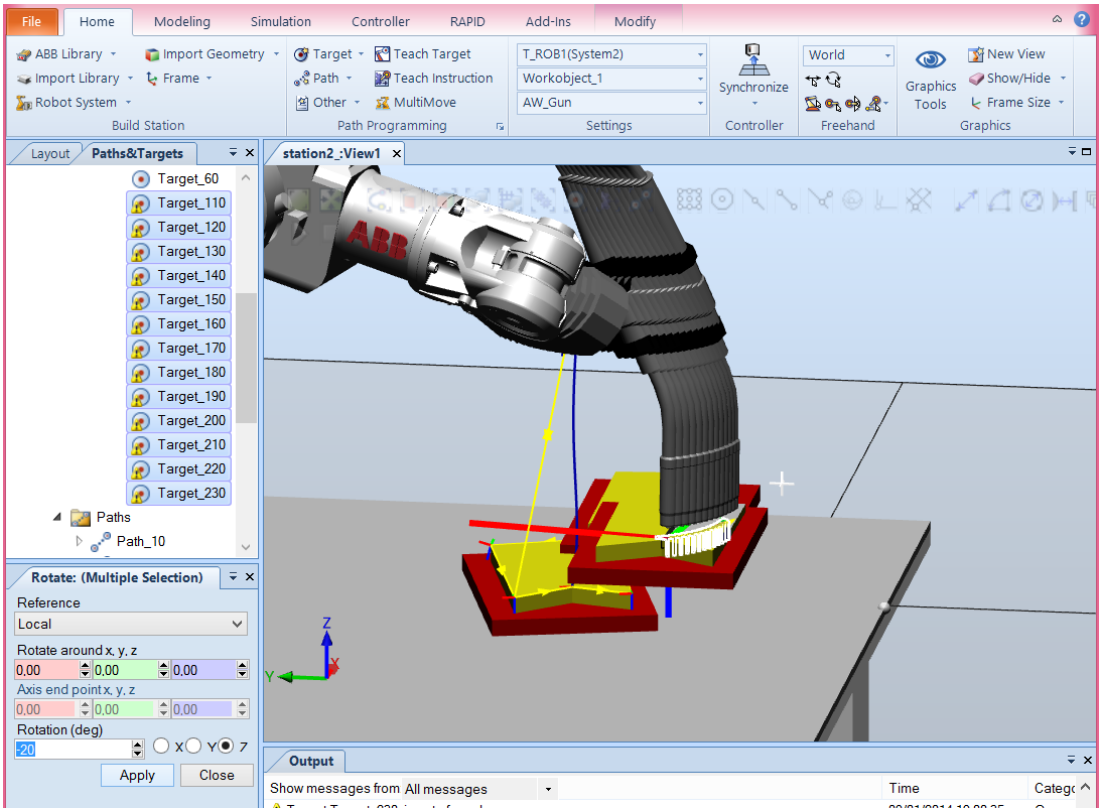
And select the edge:


station2_ - ABB RobotStudio 5.60 (64-bit)
File
Home
Modeling
Simulation
Controller
RAPID
Add-Ins
ABB Library
Import Library
Robot System
Import Geometry
Frame
Build Station
Target
Teach Target
Path
Teach Instruction
Other
MultiMove
Path Programming
T_ROB1(System2)
Workobject_1
AW_Gun
Settings
Synchronize
Controller
World
Freehand
Graphics Tools
New View
Show/Hide
Frame Size
Graphics
Layout
Paths&Targets
station2_*
Station Elements
System2
T_ROB1
Tooldata
AutoPath
Edge_1
Reverse
Reference
Remove
Start Offset (mm)
End Offset (mm)
0,00
0,00
Approximation Parameters
Linear
Circular
Constant
Min Distance
Max Radius
4,10
99996,00
Tolerance (mm)
1,00
station2_:View1
Output
Show messages from All messages
Time


Layout
Paths&Targets
station2_*
Station Elements
System2
T_ROB1
Tooldata
AutoPath
Edge_1
Reverse
Reference
Remove
Start Offset (mm)
End Offset (mm)
0,00
0,00
Approximation Parameters
Linear
Circular
Constant
Distance (mm)
Max Radius
3
99996,00
Tolerance (mm)
1,00
More >>
Clear
Create
Close
station2_:View1
Output
Show messages from All messages
Time
Imported C:\Users\Pedro Neto\Documents\RobotStudio\Geometry\WP1_C.ipt
09/01/2014 15:45:15
Border created (Part_1)
09/01/2014 15:53:02
Path created (Path_20) [2]
09/01/2014 16:06:40
MoveL
v1000
z100
AW_Gun
\WObj:=Workobject_1
Selection Level
Snap Mode
UCS: Station
-127,89 -935,11 106,8
Controller status: 1/1

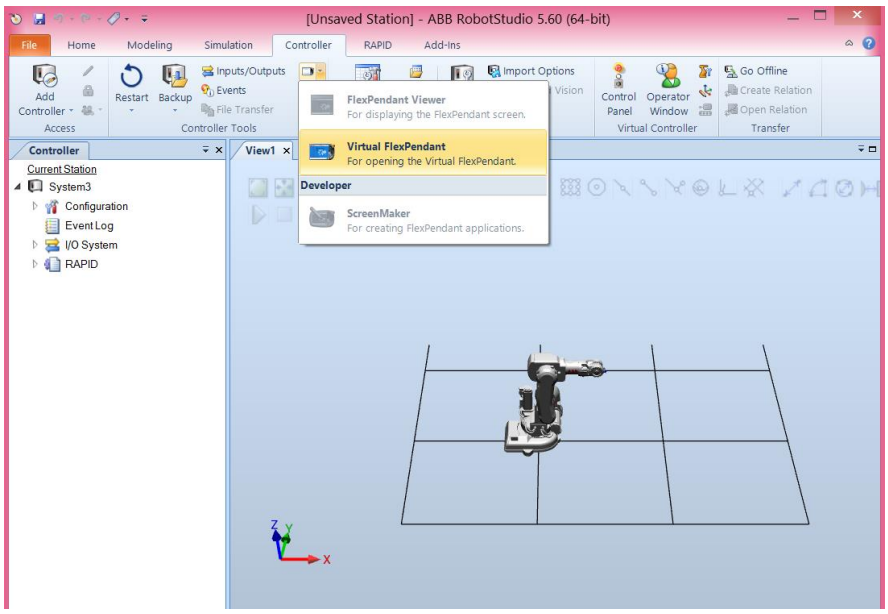
And we have the targets:



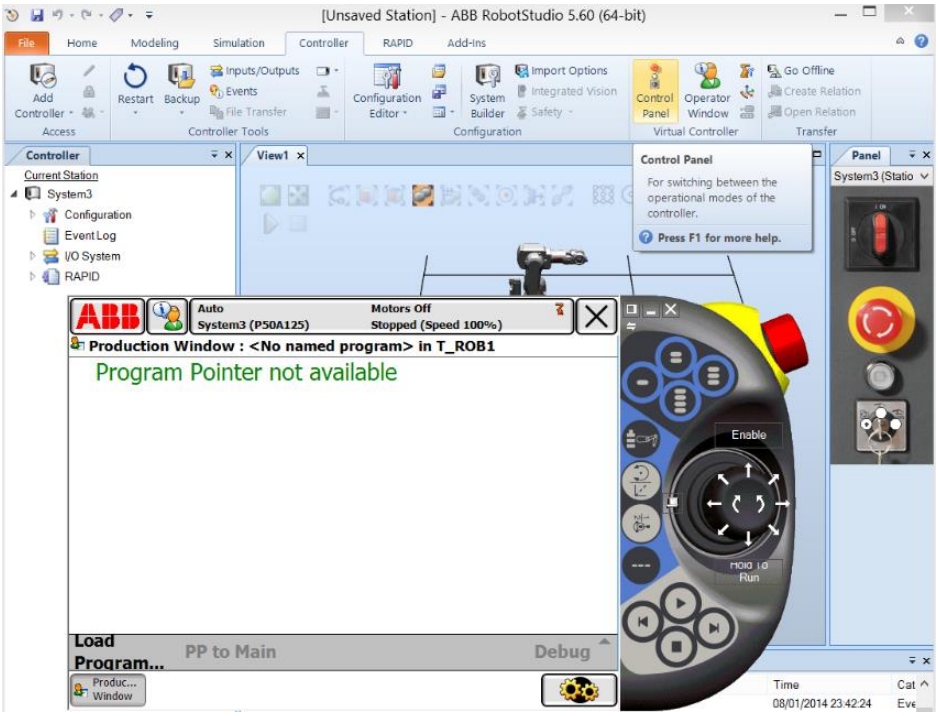


4. TEACH PENDANT

ABB calls the teach pendant FlexPendant. RobotStudio provides us a virtual teach pendant. Include a robotic arm and controller and then call the Virtual FlexPendant.

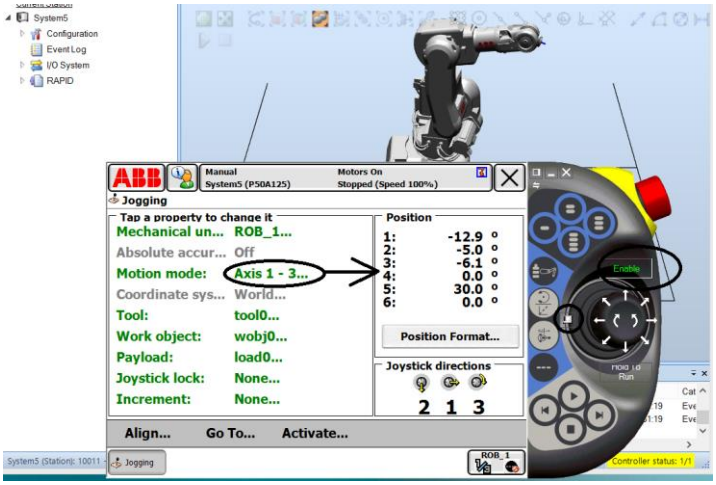


And we have this:

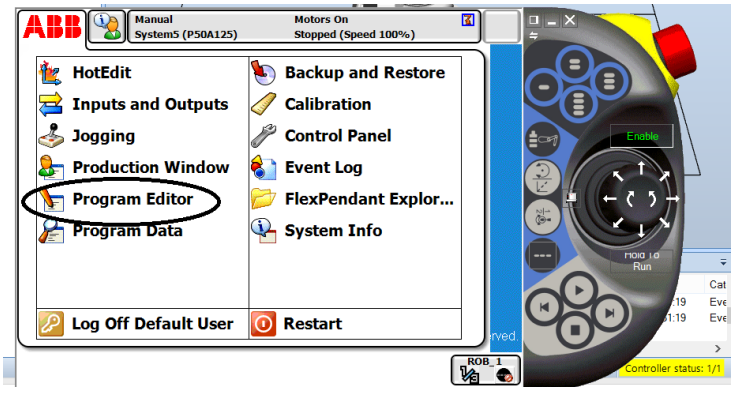


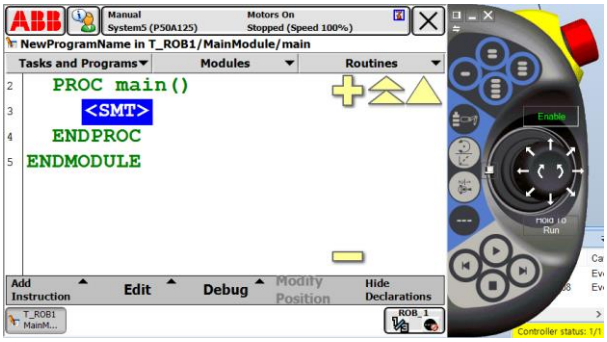
On the right you can see the virtual key to select the mode of operation. By default it is in automatic mode. To do something (move the virtual robot and create a robot program) change it to manual mode and select jogging:

You can jog the robot by enabling the joystick and pressing the arrows for some seconds.

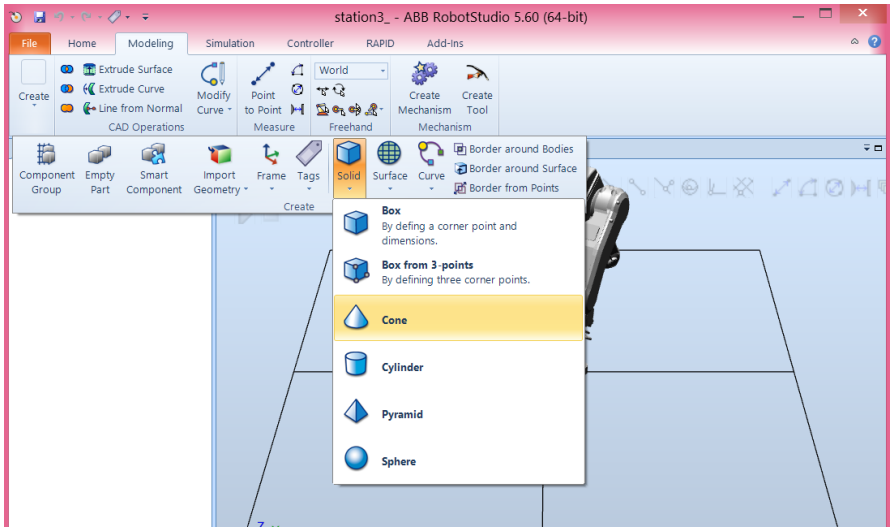


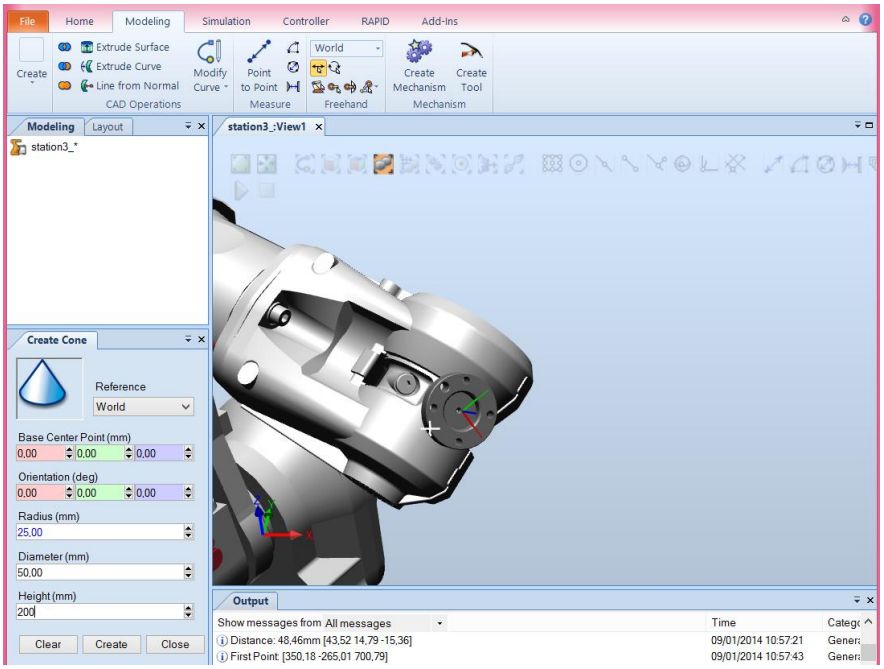
Create a program is also simple:



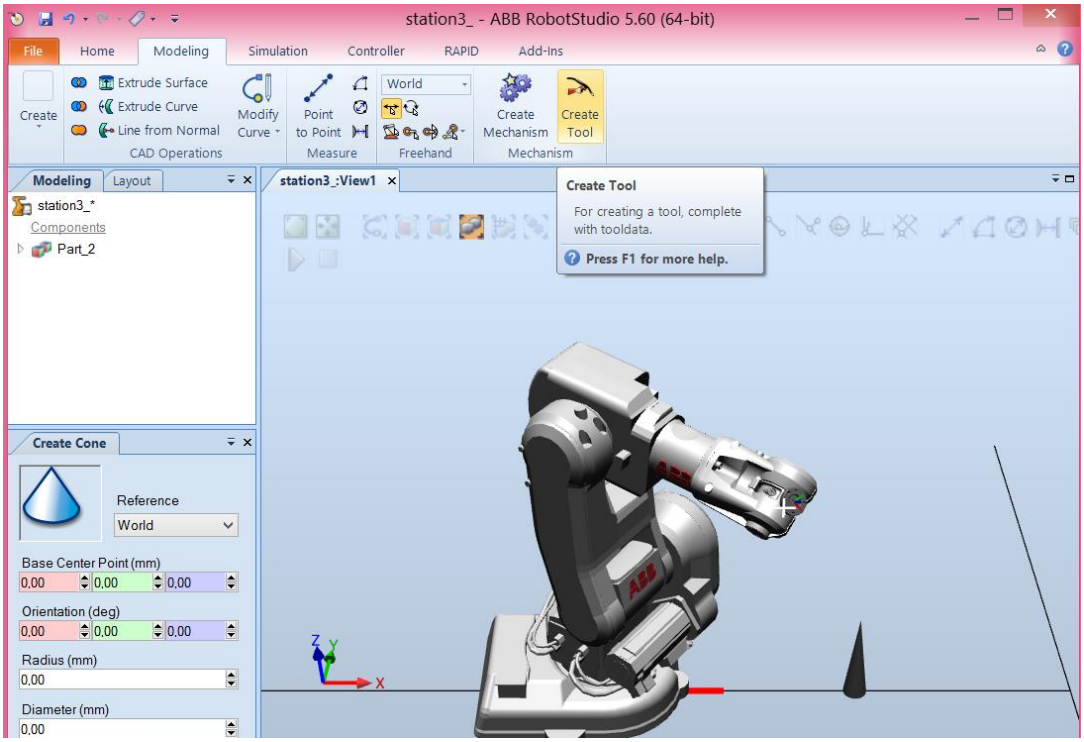


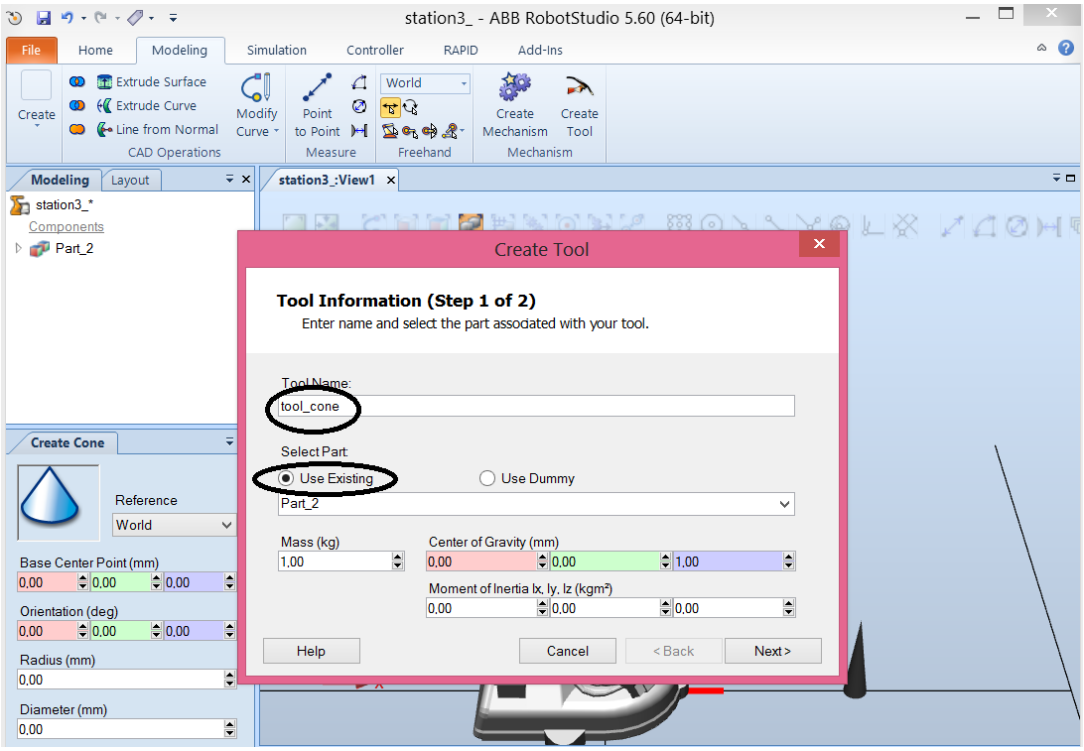
We can create a simple tool for this purpose:



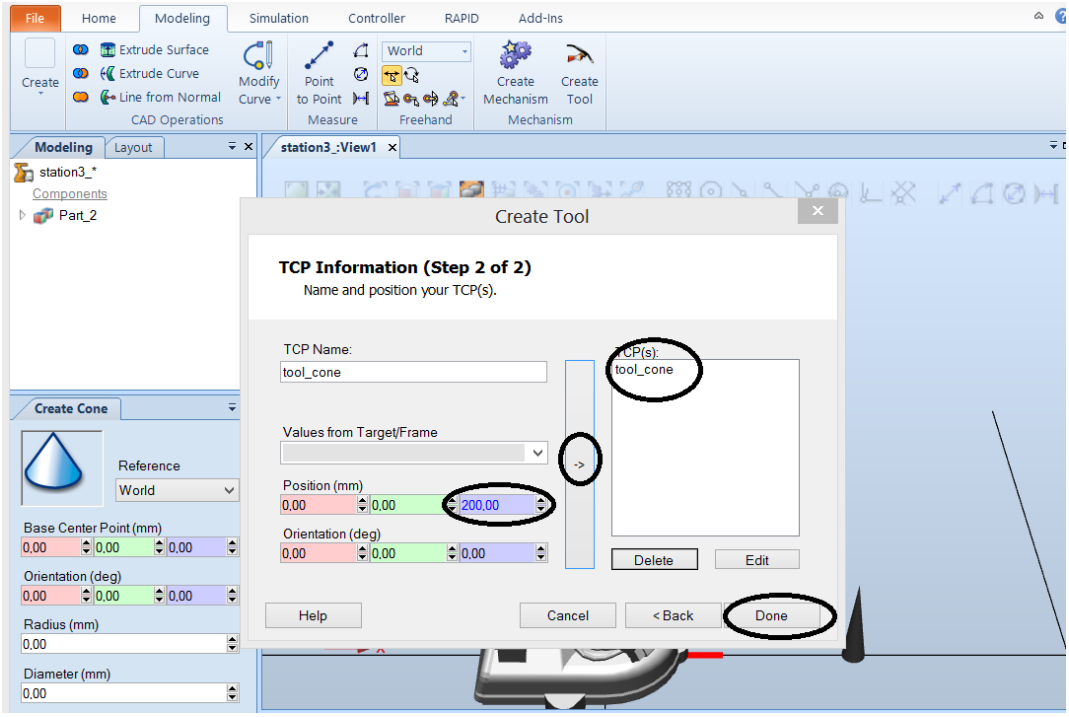


The geometry for the virtual tool with the name Part_1 was created. But we need to create a tool:

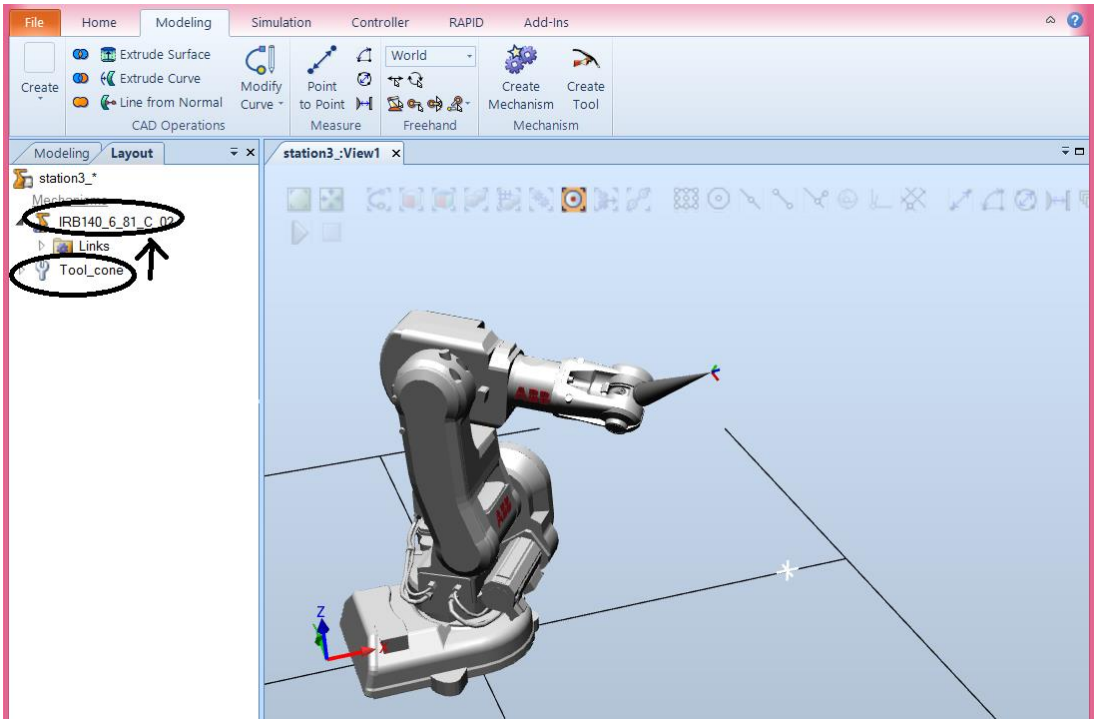




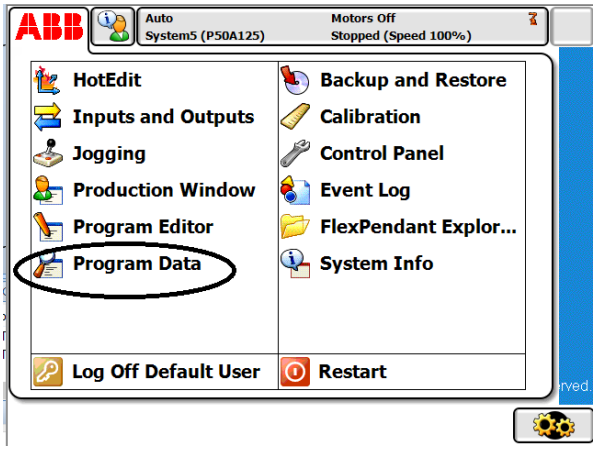
Define where the TCP is:

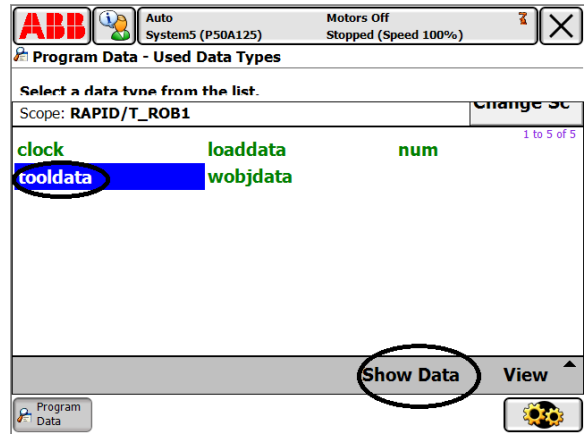


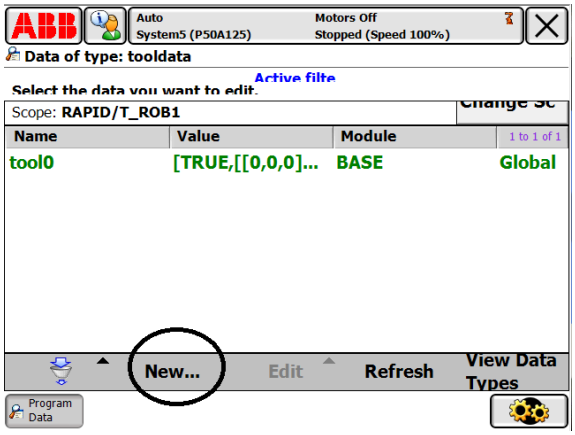
Now as in MODULE I include the tool into the robot:

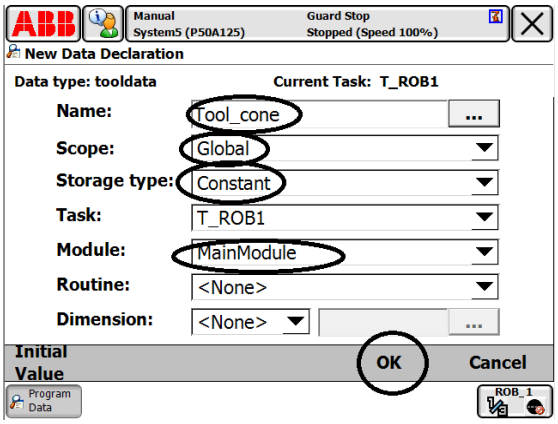


OK, we have a new tool in RobotStudio but we have to call it from the teach pendant:

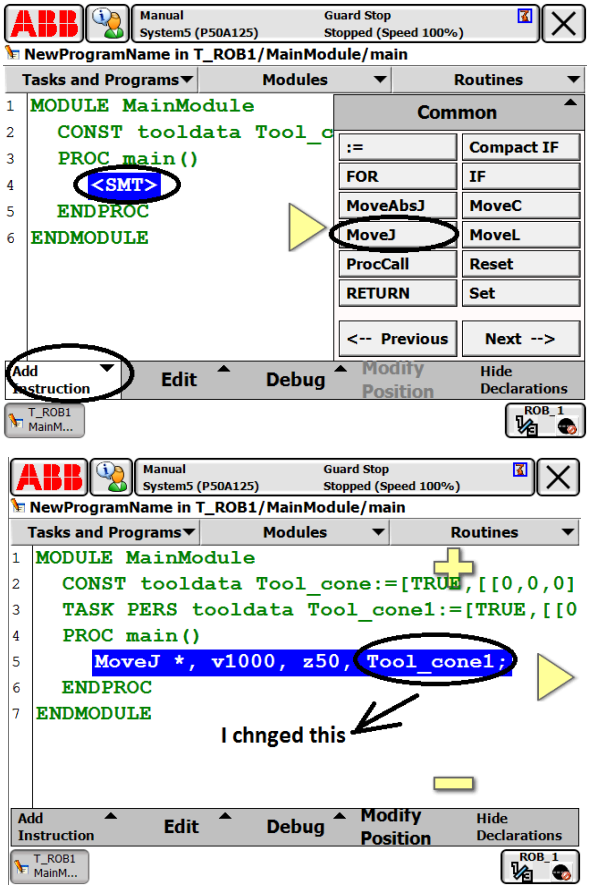

Auto
System5 (P50A125)
Motors Off
Stopped (Speed 100%)
HotEdit
Inputs and Outputs
Jogging
Production Window
Program Editor
Program Data
Backup and Restore
Calibration
Control Panel
Event Log
FlexPendant Explor...
System Info
Log Off Default User
Restart


Auto
System5 (P50A125)
Motors Off
Stopped (Speed 100%)
Program Data - Used Data Types
Scope: RAPID/T_ROB1
1 to 5 of 5
clock
loaddata
num
tooldata
wobjdata
Show Data
View
Program Data

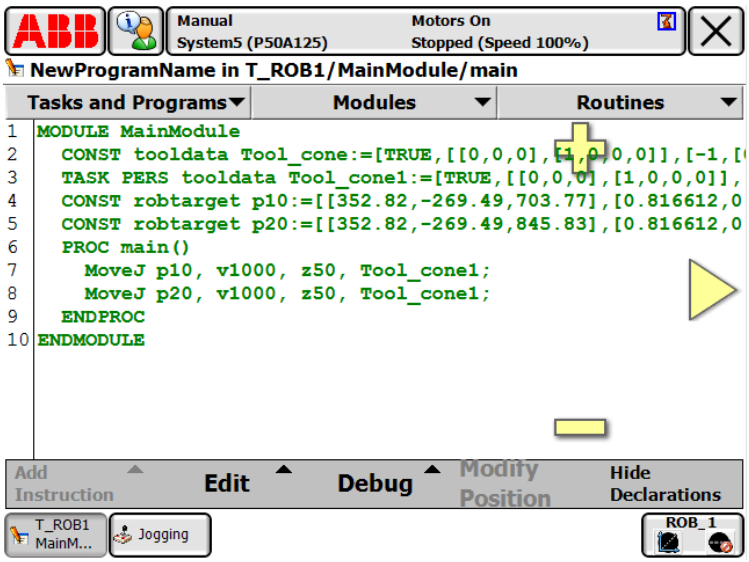

Auto
System5 (P50A125)
Motors Off
Stopped (Speed 100%)
Data of type: tooldata
Scope: RAPID/T_ROB1
Name
Value
Module
1 to 1 of 1
tool0
[TRUE,[[0,0,0]...
BASE
Global
New...
Edit
Refresh
View Data Types
Program Data

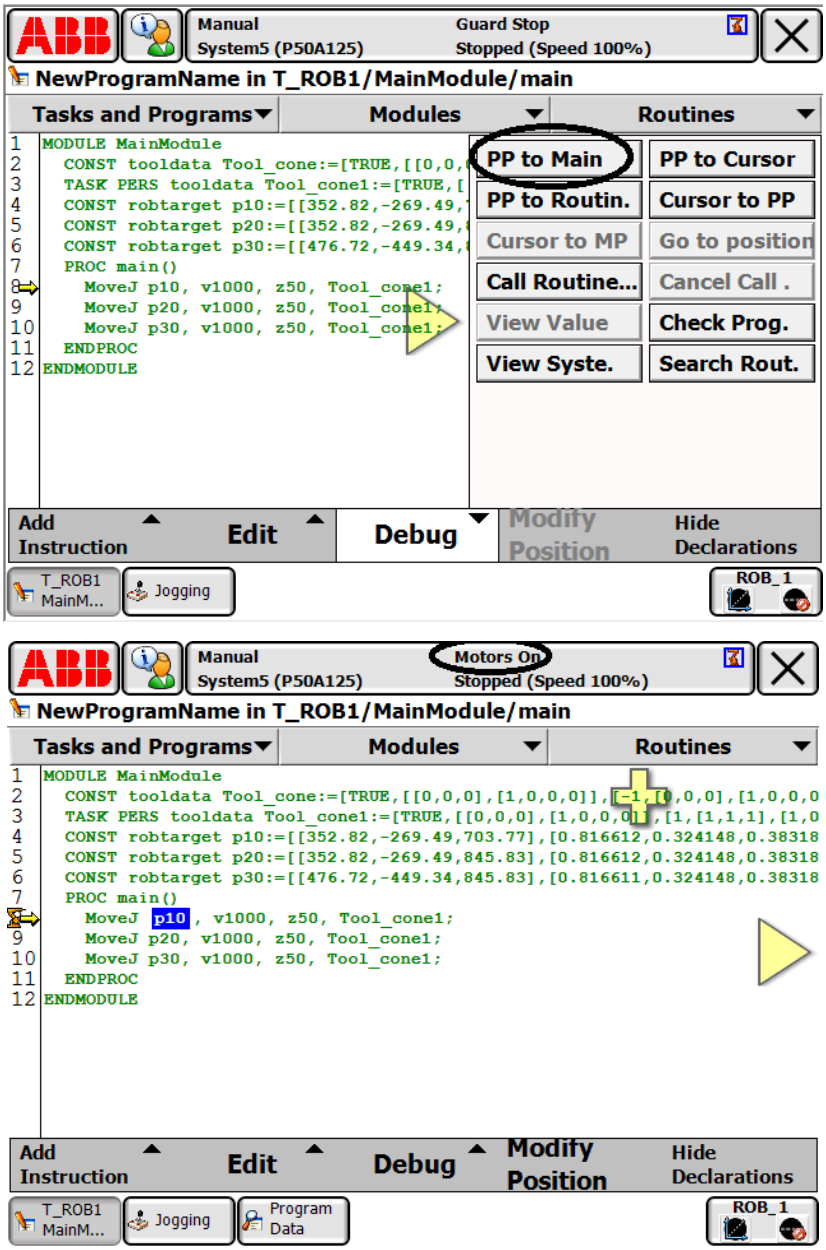

Manual
System5 (P50A125)
Guard Stop
Stopped (Speed 100%)
New Data Declaration
Data type: tooldata
Current Task: T_ROB1
Name:
Tool_cone
Scope:
Global
Storage type:
Constant
Task:
T_ROB1
Module:
MainModule
Routine:
<None>
Dimension:
<None>
Initial Value
OK
Cancel
Program Data
ROB_1

Backing to program edition, it is possible to create a motion command with the current pose of the robot:



You can create other points and just simulate the program created:

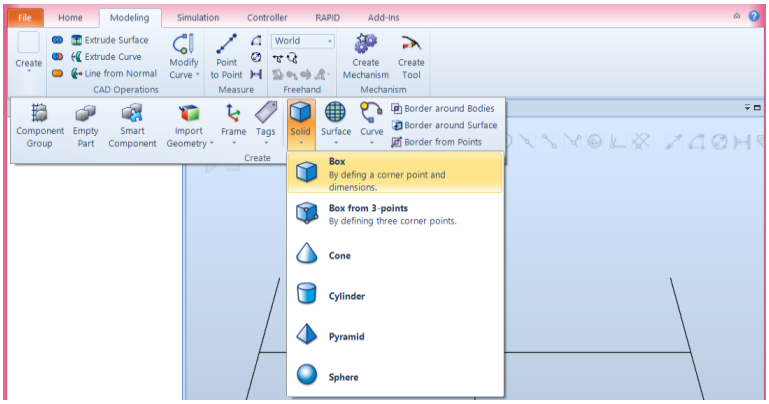


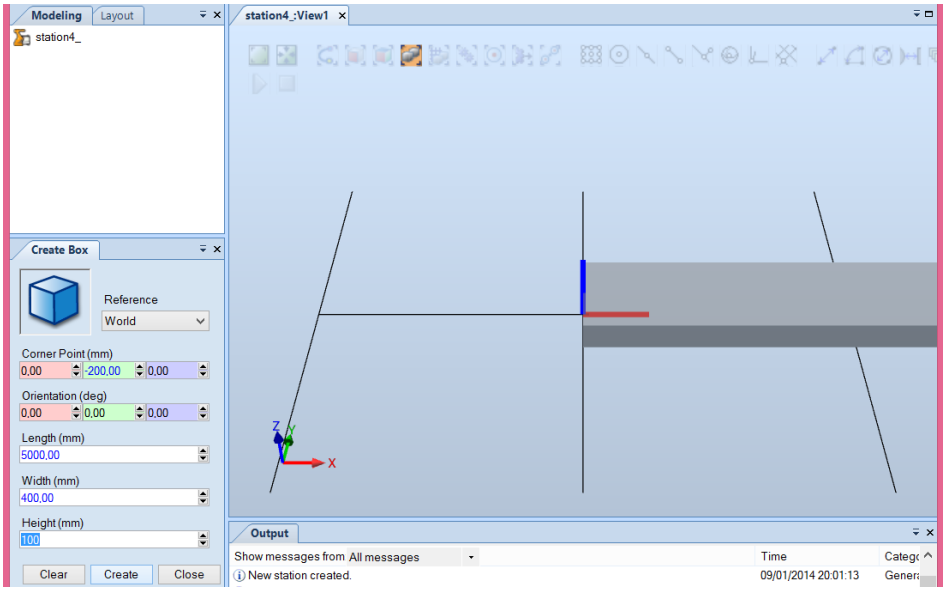

Manual
System5 (P50A125)
Guard Stop
Stopped (Speed 100%)
NewProgramName in T_ROB1/MainModule/main
Tasks and Programs
Modules
Routines
MODULE MainModule
CONST tooldata Tool_cone:=[TRUE,[[0,0,
TASK PERS tooldata Tool_cone1:=[TRUE,[
CONST robtarget p10:=[[352.82,-269.49,
CONST robtarget p20:=[[352.82,-269.49,
CONST robtarget p30:=[[476.72,-449.34,
PROC main()
MoveJ p10, v1000, z50, Tool_cone1;
MoveJ p20, v1000, z50, Tool_cone1;
MoveJ p30, v1000, z50, Tool_cone1;
ENDPROC
ENDMODULE
PP to Main
PP to Cursor
PP to Routin.
Cursor to PP
Cursor to MP
Go to position
Call Routine...
Cancel Call .
View Value
Check Prog.
View Syste.
Search Rout.
Add Instruction
Edit
Debug
Modify Position
Hide Declarations
T_ROB1 MainM...
Jogging
ROB_1
Manual
System5 (P50A125)
Motors On
Stopped (Speed 100%)
NewProgramName in T_ROB1/MainModule/main
Tasks and Programs
Modules
Routines
MODULE MainModule
CONST tooldata Tool_cone:=[TRUE,[[0,0,0],[1,0,0,0]],[-1,[0,0,0],[1,0,0,0
TASK PERS tooldata Tool_cone1:=[TRUE,[[0,0,0],[1,0,0,0]],[1,[1,1,1],[1,0
CONST robtarget p10:=[[352.82,-269.49,703.77],[0.816612,0.324148,0.38318
CONST robtarget p20:=[[352.82,-269.49,845.83],[0.816612,0.324148,0.38318
CONST robtarget p30:=[[476.72,-449.34,845.83],[0.816611,0.324148,0.38318
PROC main()
MoveJ p10, v1000, z50, Tool_cone1;
MoveJ p20, v1000, z50, Tool_cone1;
MoveJ p30, v1000, z50, Tool_cone1;
ENDPROC
ENDMODULE
Add Instruction
Edit
Debug
Modify Position
Hide Declarations
T_ROB1 MainM...
Jogging
Program Data
ROB_1

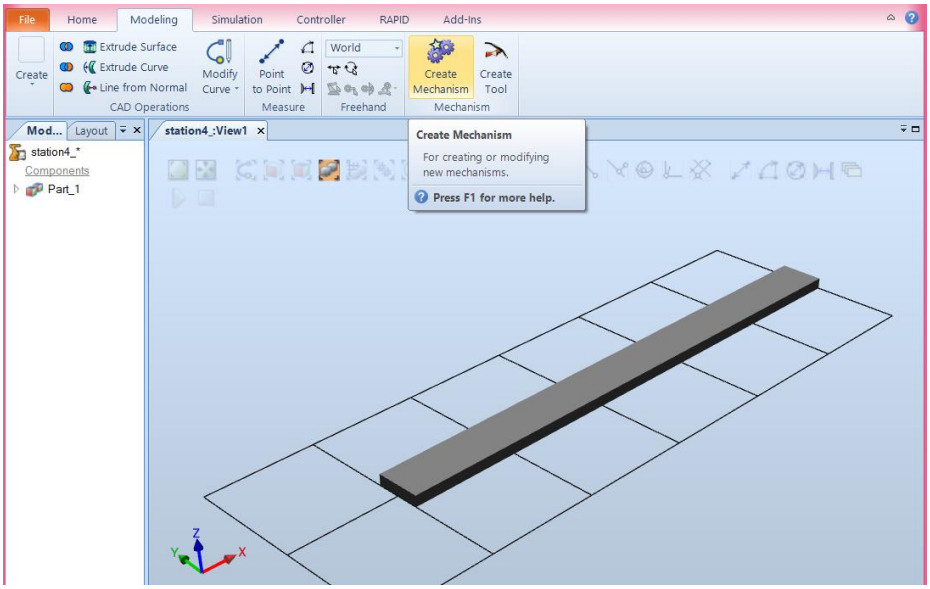
5. MECHANISMS

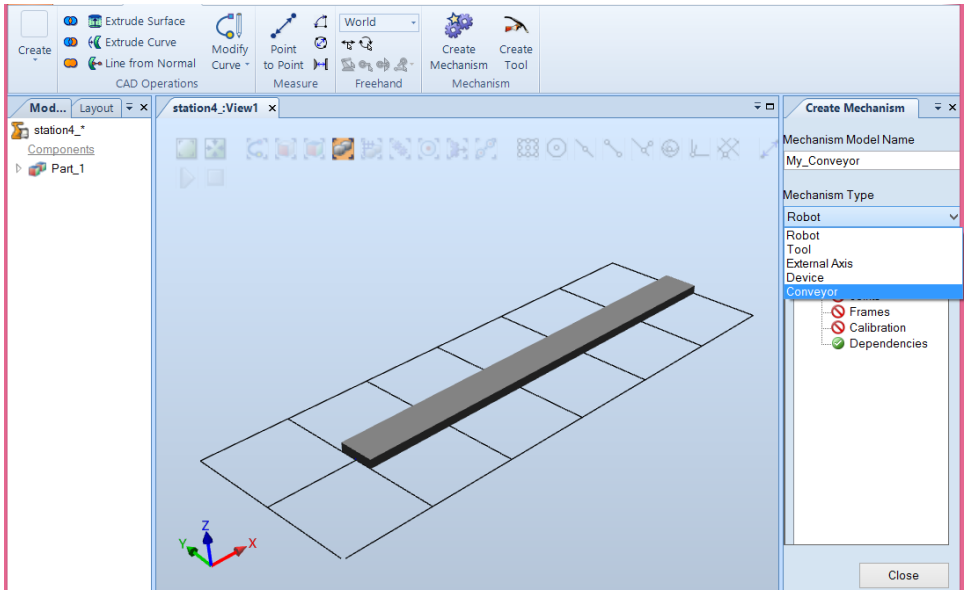
5.1. Conveyor

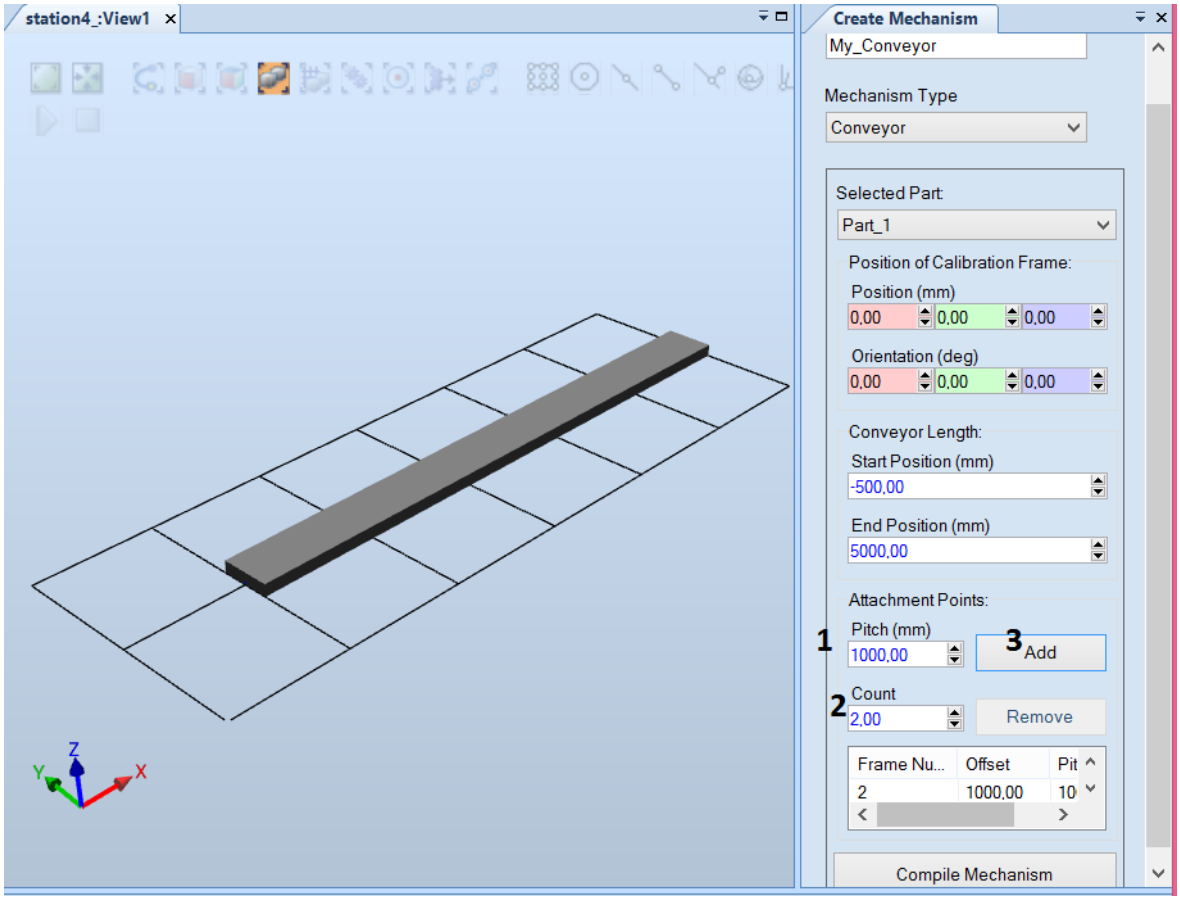
We will start by creating a mechanism, a conveyor.



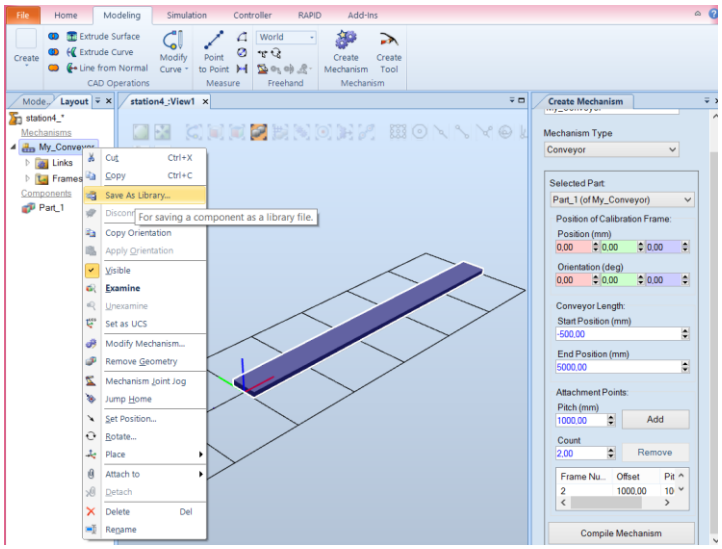



File
Home
Modeling
Simulation
Controller
RAPID
Add-Ins
Create
Extrude Surface
Extrude Curve
Line from Normal
CAD Operations
Modify Curve
Point to Point
Measure
World
Freehand
Create Mechanism
Create Tool
Mechanism
Mod...
Layout
station4_*
Components
Part_1
station4_:View1
Create Mechanism
For creating or modifying new mechanisms.
Press F1 for more help.


Create
Extrude Surface
Extrude Curve
Line from Normal
CAD Operations
Modify Curve
Point to Point
Measure
World
Freehand
Create Mechanism
Create Tool
Mechanism
Mod...
Layout
station4_*
Components
Part_1
station4_:View1
Create Mechanism
Mechanism Model Name
My_Conveyor
Mechanism Type
Robot
Robot
Tool
External Axis
Device
Conveyor
Frames
Calibration
Dependencies
Close

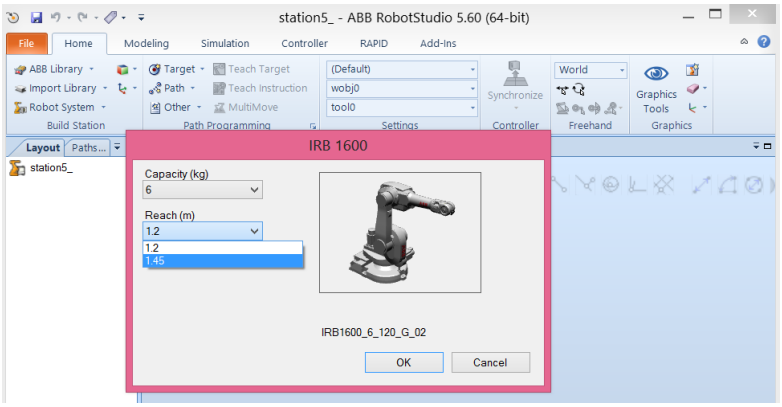


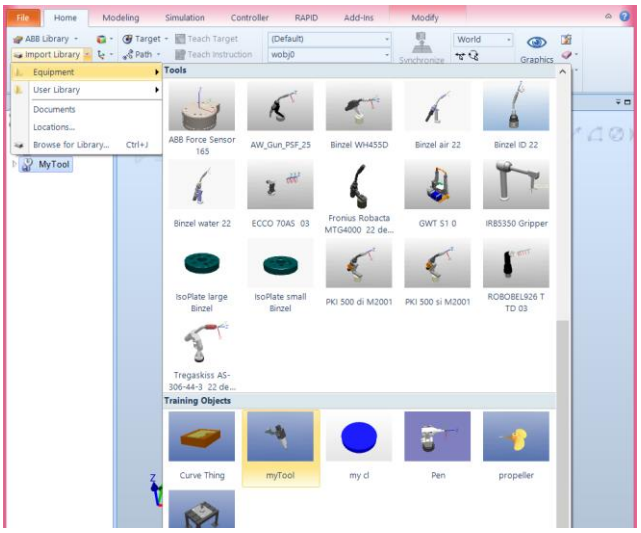
And compile the mechanism.



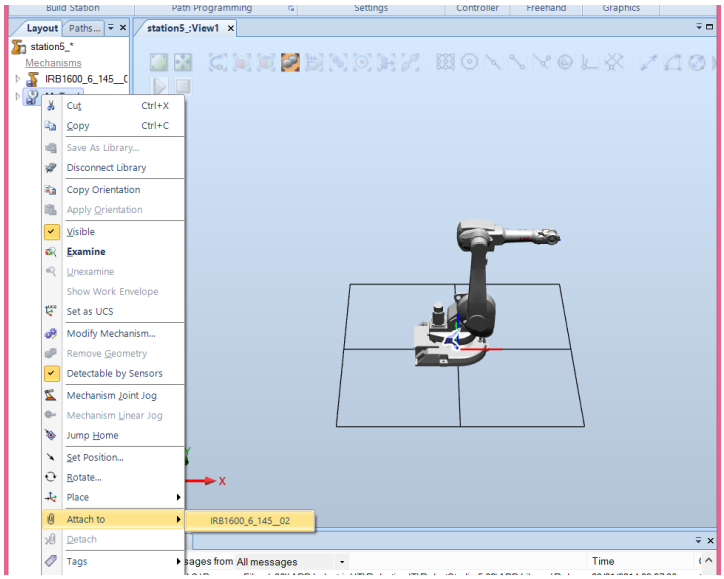
5.2. Creating a station

Start by selecting a robot, controller and tool:

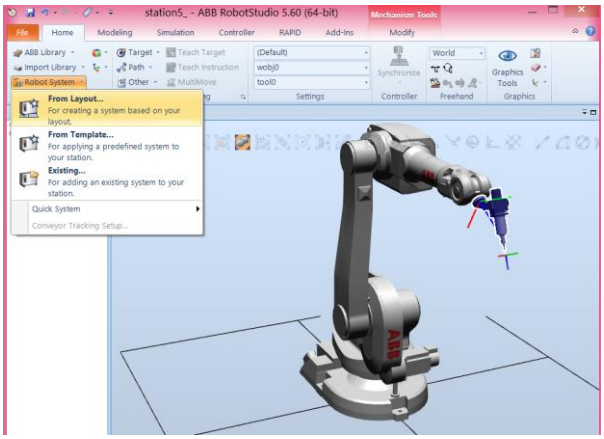


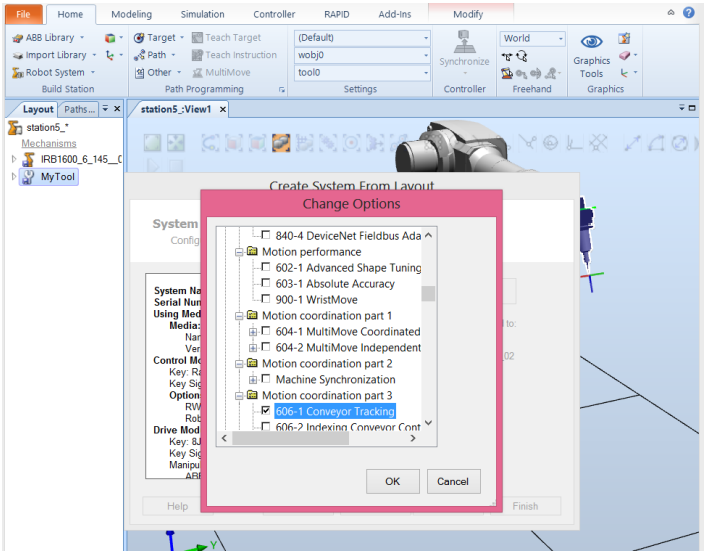


Attach the tool to the robot:

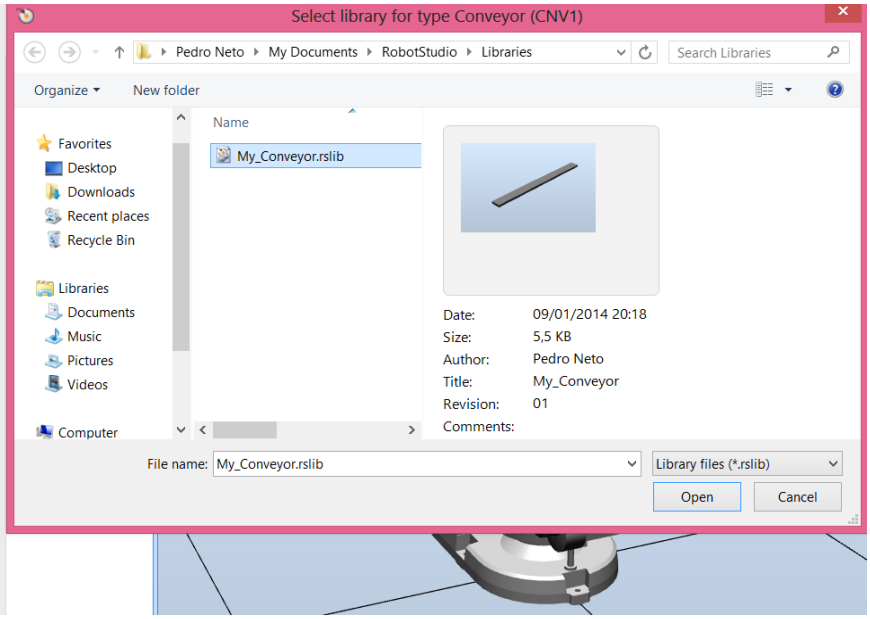


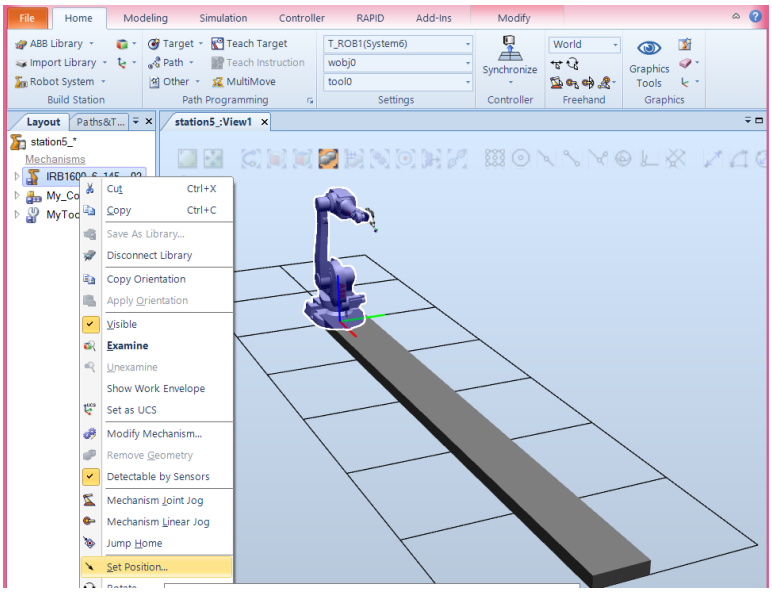
Add a controller:

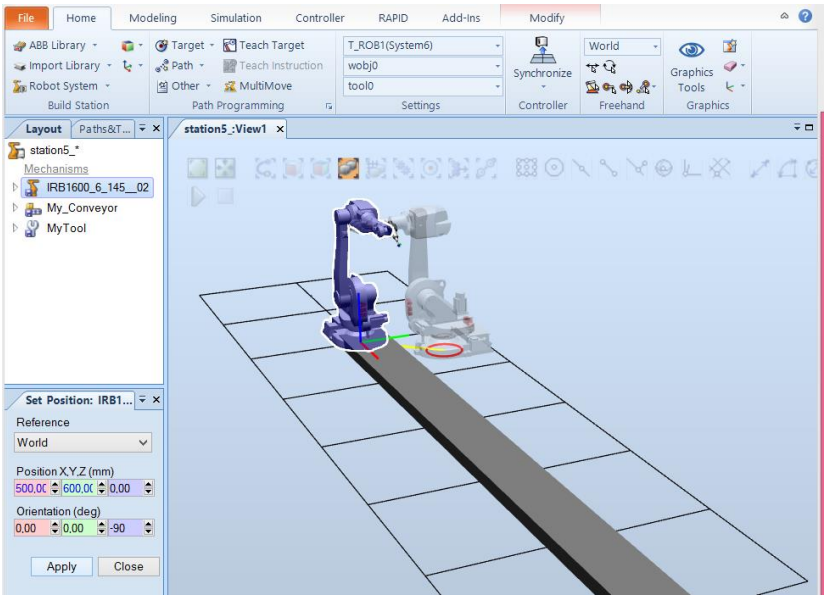


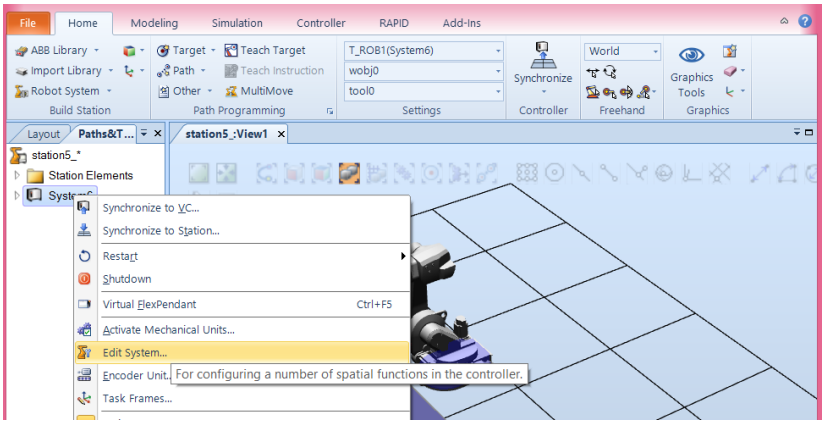


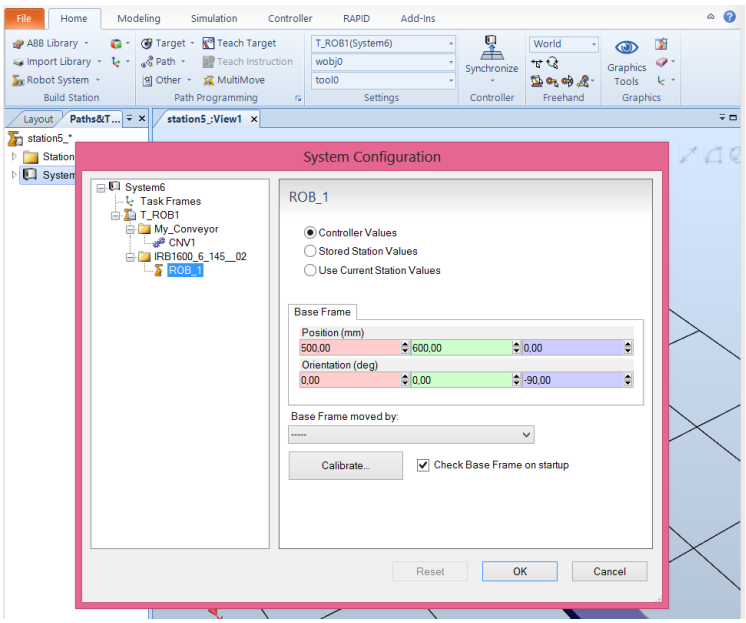
And import the conveyor by clicking import library:

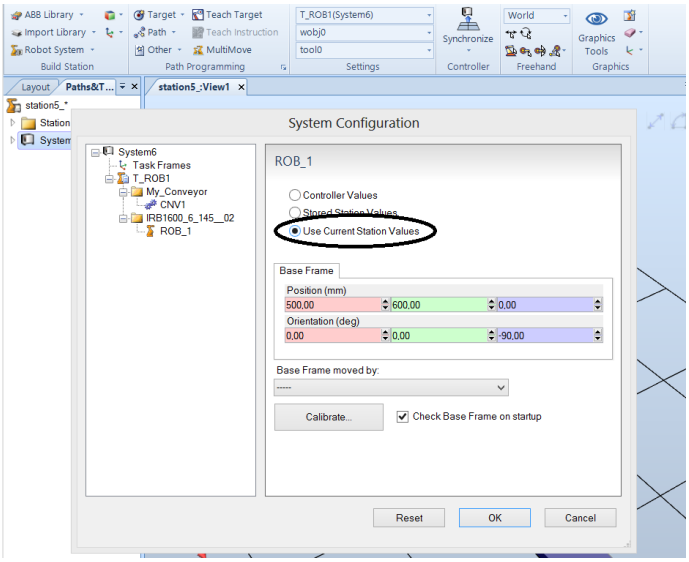

Select library for type Conveyor (CNV1)
Pedro Neto ▸ My Documents ▸ RobotStudio ▸ Libraries
Search Libraries
Organize
New folder
Favorites
Desktop
Downloads
Recent places
Recycle Bin
Libraries
Documents
Music
Pictures
Videos
Computer
Name
My_Conveyor.rslib
Date: 09/01/2014 20:18
Size: 5,5 KB
Author: Pedro Neto
Title: My_Conveyor
Revision: 01
Comments:
File name: My_Conveyor.rslib
Library files (*.rslib)
Open
Cancel

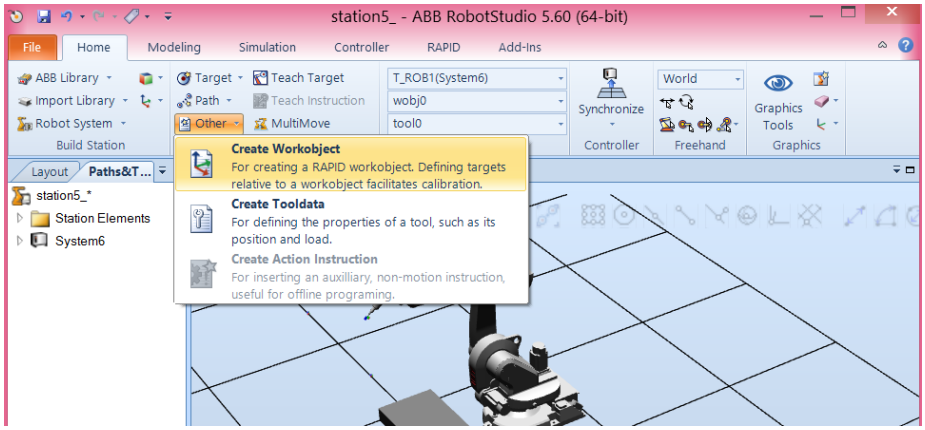

File
Home
Modeling
Simulation
Controller
RAPID
Add-Ins
Modify
ABB Library
Import Library
Robot System
Build Station
Target
Path
Other
Teach Target
Teach Instruction
MultiMove
Path Programming
T_ROB1(System6)
wobj0
tool0
Settings
Synchronize
Controller
World
Freehand
Graphics Tools
Graphics
Layout
Paths&T...
station5_*
Mechanisms
station5_:View1
Cut Ctrl+X
Copy Ctrl+C
Save As Library...
Disconnect Library
Copy Orientation
Apply Orientation
Visible
Examine
Unexamine
Show Work Envelope
Set as UCS
Modify Mechanism...
Remove Geometry
Detectable by Sensors
Mechanism Joint Jog
Mechanism Linear Jog
Jump Home
Set Position...

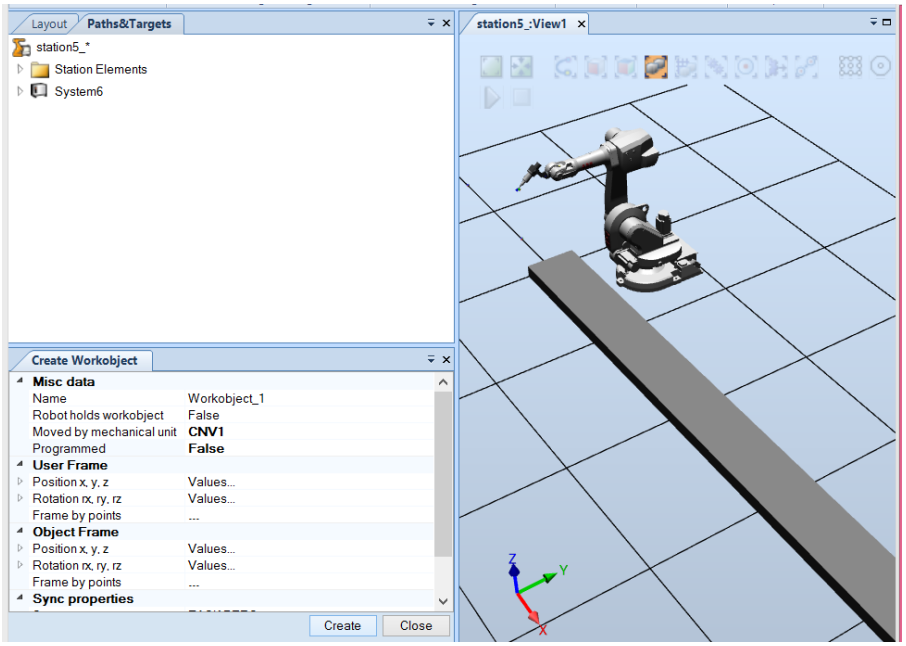

File
Home
Modeling
Simulation
Controller
RAPID
Add-Ins
Modify
ABB Library
Import Library
Robot System
Build Station
Target
Teach Target
Path
Teach Instruction
Other
MultiMove
Path Programming
T_ROB1(System6)
wobj0
tool0
Settings
Synchronize
Controller
World
Freehand
Graphics Tools
Graphics
Layout
Paths&T...
station5_*
Mechanisms
IRB1600_6_145__02
My_Conveyor
MyTool
station5_:View1
Set Position: IRB1...
Reference
World
Position X,Y,Z (mm)
500,00
600,00
0,00
Orientation (deg)
0,00
0,00
-90
Apply
Close

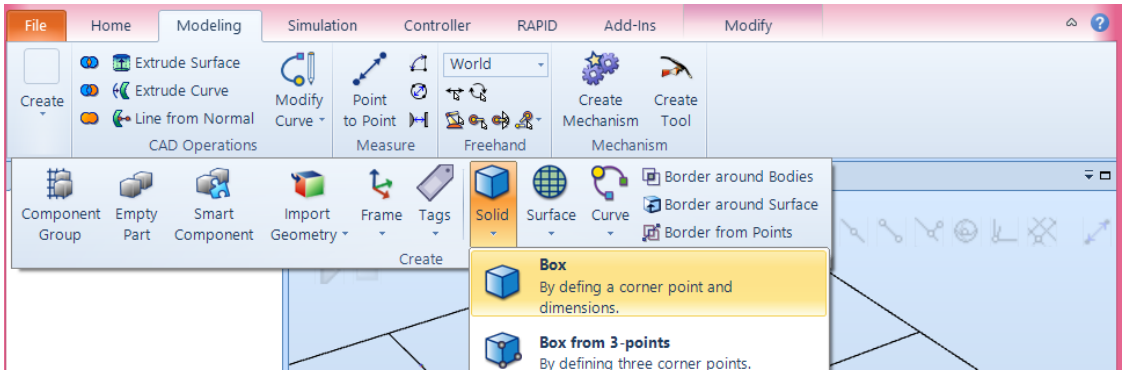

File
Home
Modeling
Simulation
Controller
RAPID
Add-Ins
Build Station
Path Programming
Settings
Controller
Freehand
Graphics
Layout
Paths&T...
station5_*
Station Elements
Synchronize to VC...
Synchronize to Station...
Restart
Shutdown
Virtual FlexPendant
Ctrl+F5
Activate Mechanical Units...
Edit System...
Encoder Unit...
Task Frames...
For configuring a number of spatial functions in the controller.
station5_:View1

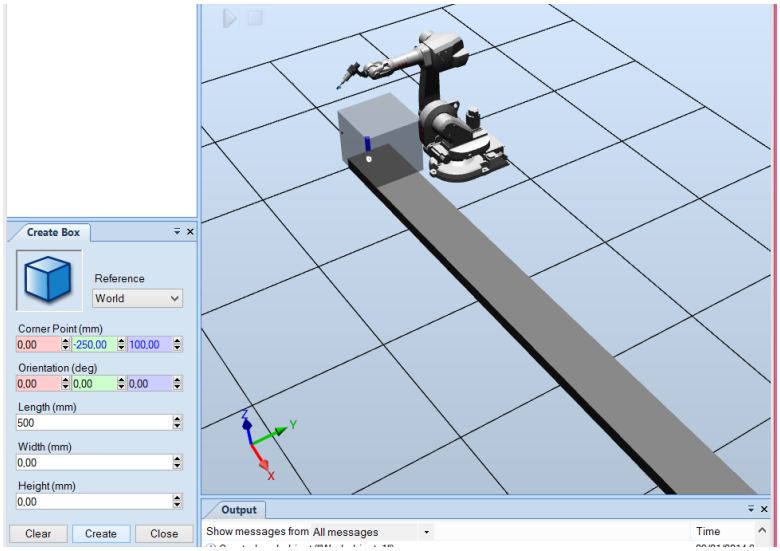

System Configuration
System6
Task Frames
T_ROB1
My_Conveyor
CNV1
IRB1600_6_145__02
ROB_1
ROB_1
Controller Values
Stored Station Values
Use Current Station Values
Base Frame
Position (mm)
500,00
600,00
0,00
Orientation (deg)
0,00
0,00
-90,00
Base Frame moved by:
Calibrate...
Check Base Frame on startup
Reset
OK
Cancel

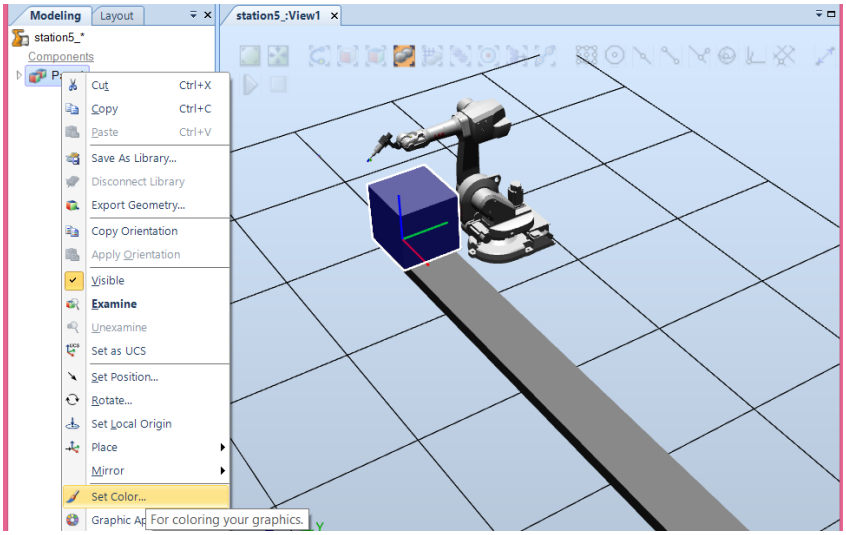

System Configuration
System6
Task Frames
T_ROB1
My_Conveyor
CNV1
IRB1600_6_145__02
ROB_1
ROB_1
Controller Values
Use Current Station Values
Base Frame
Position (mm)
500,00
600,00
0,00
Orientation (deg)
0,00
0,00
-90,00
Base Frame moved by:
Calibrate...
Check Base Frame on startup
Reset
OK
Cancel

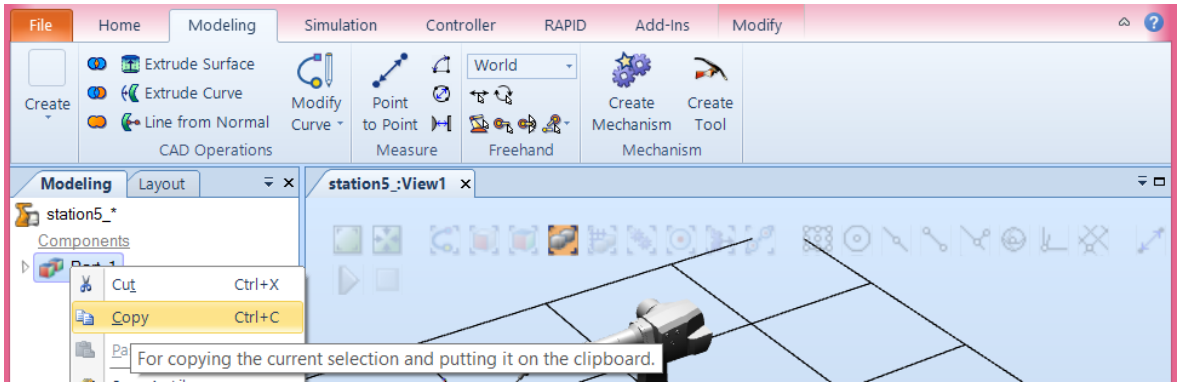

station5_ - ABB RobotStudio 5.60 (64-bit)
File
Home
Modeling
Simulation
Controller
RAPID
Add-Ins
Create Workobject
For creating a RAPID workobject. Defining targets relative to a workobject facilitates calibration.
Create Tooldata
For defining the properties of a tool, such as its position and load.
Create Action Instruction
For inserting an auxilliary, non-motion instruction, useful for offline programing.
station5_*
Station Elements
System6

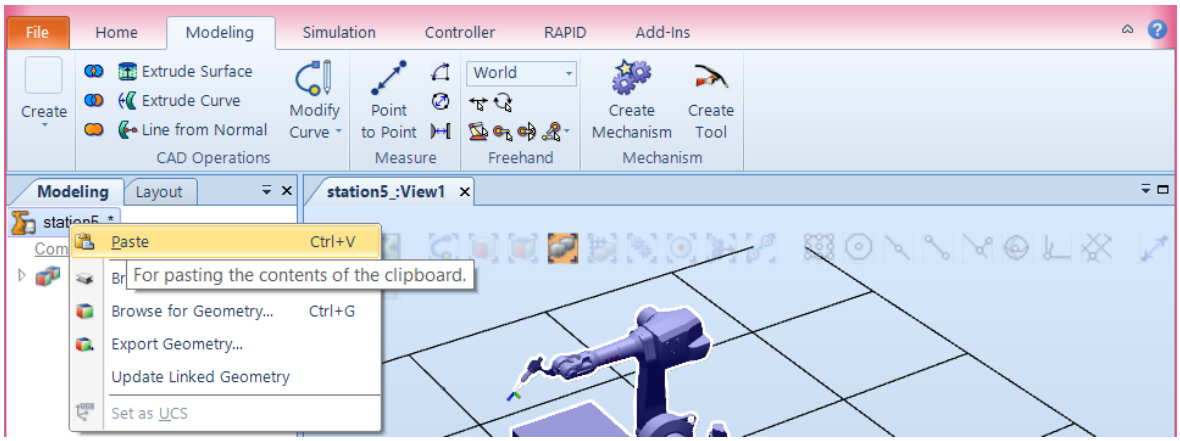

Paths&Targets
station5_*
Station Elements
System6
station5_:View1
Create Workobject
Misc data
Name
Workobject_1
Robot holds workobject
False
Moved by mechanical unit
CNV1
Programmed
False
User Frame
Position x, y, z
Values...
Rotation rx, ry, rz
Values...
Frame by points
Object Frame
Position x, y, z
Values...
Rotation rx, ry, rz
Values...
Frame by points
Sync properties
Create
Close

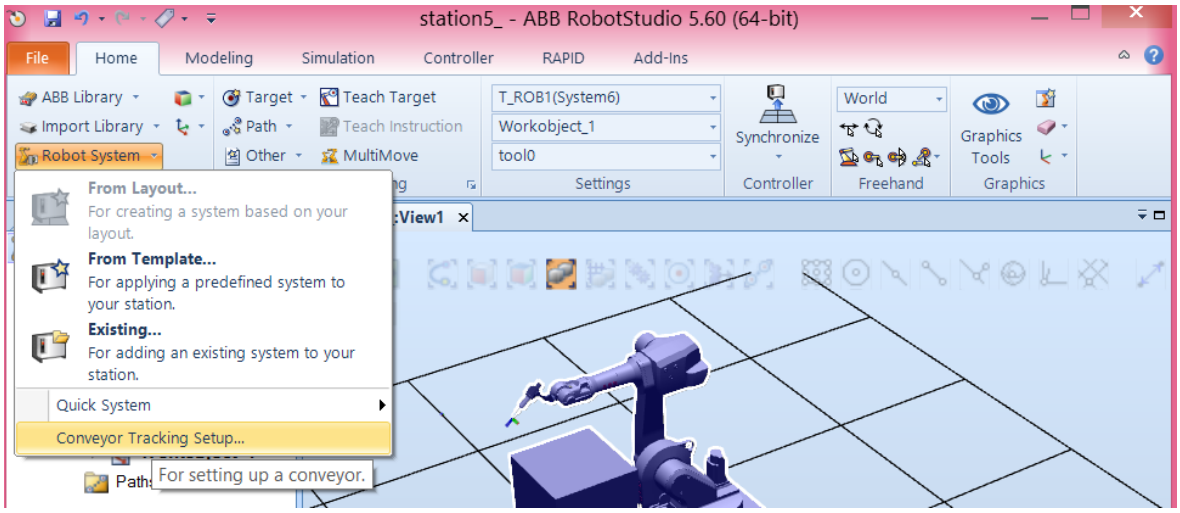

File
Home
Modeling
Simulation
Controller
RAPID
Add-Ins
Modify
Create
Extrude Surface
Extrude Curve
Line from Normal
CAD Operations
Modify Curve
Point to Point
Measure
World
Freehand
Create Mechanism
Create Tool
Mechanism
Component Group
Empty Part
Smart Component
Import Geometry
Frame
Tags
Solid
Surface
Curve
Border around Bodies
Border around Surface
Border from Points
Create
Box
By defing a corner point and dimensions.
Box from 3-points
By defining three corner points.


Create Box
Reference
World
Corner Point (mm)
0.00
-250.00
100.00
Orientation (deg)
0.00
0.00
0.00
Length (mm)
500
Width (mm)
0.00
Height (mm)
0.00
Clear
Create
Close
Output
Show messages from All messages
Time

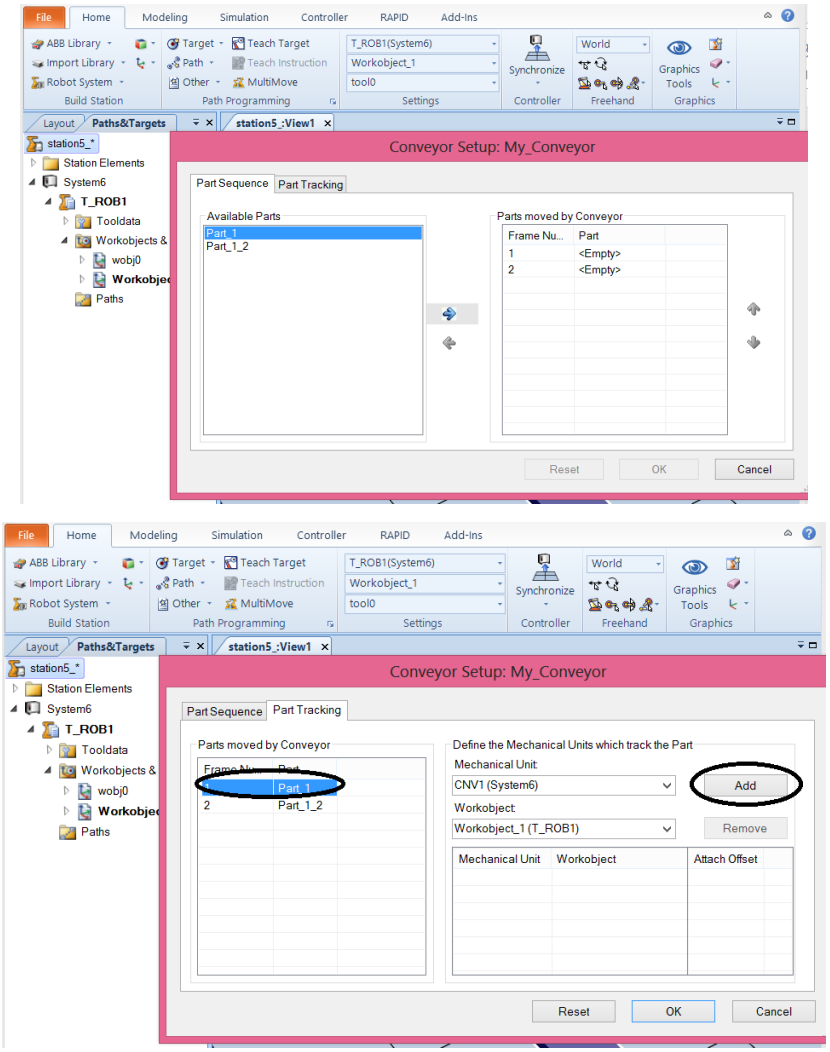

Modeling
Layout
station5_*
Components
station5_:View1
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Save As Library...
Disconnect Library
Export Geometry...
Copy Orientation
Apply Orientation
Visible
Examine
Unexamine
Set as UCS
Set Position...
Rotate...
Set Local Origin
Place
Mirror
Set Color...
For coloring your graphics.



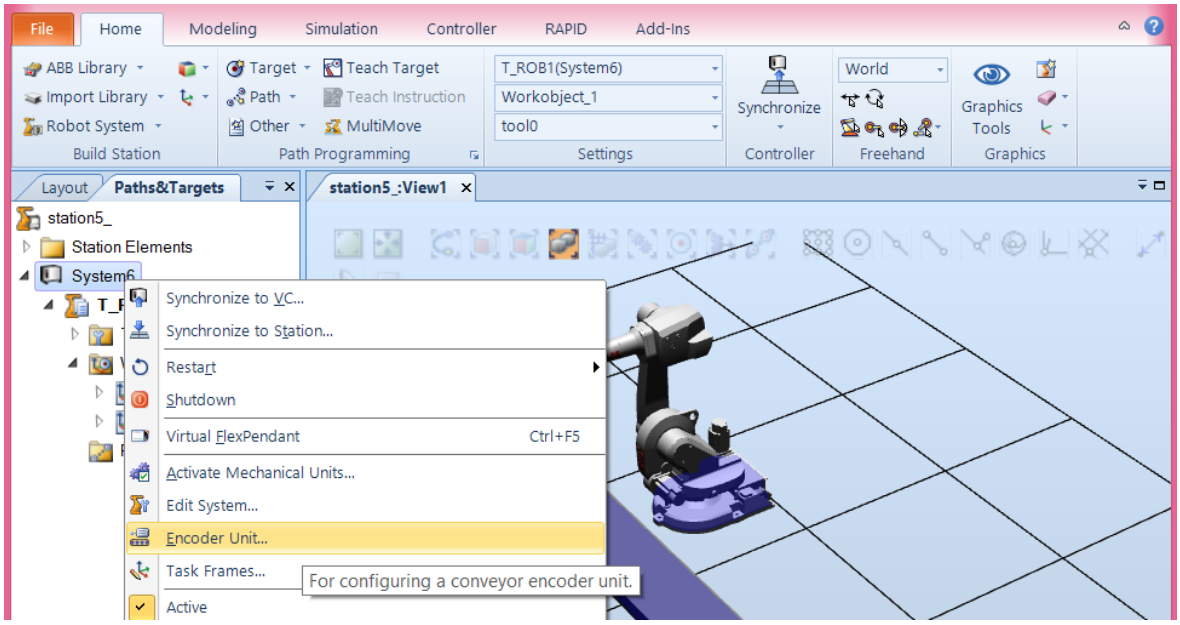


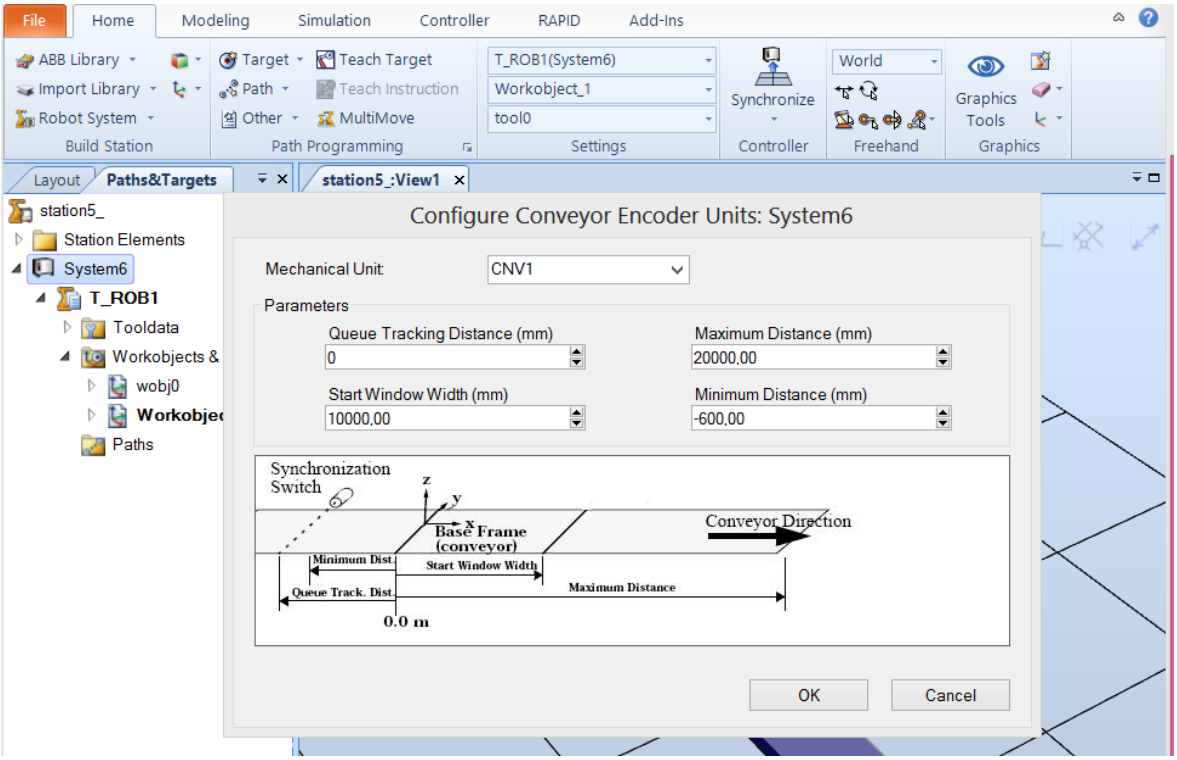


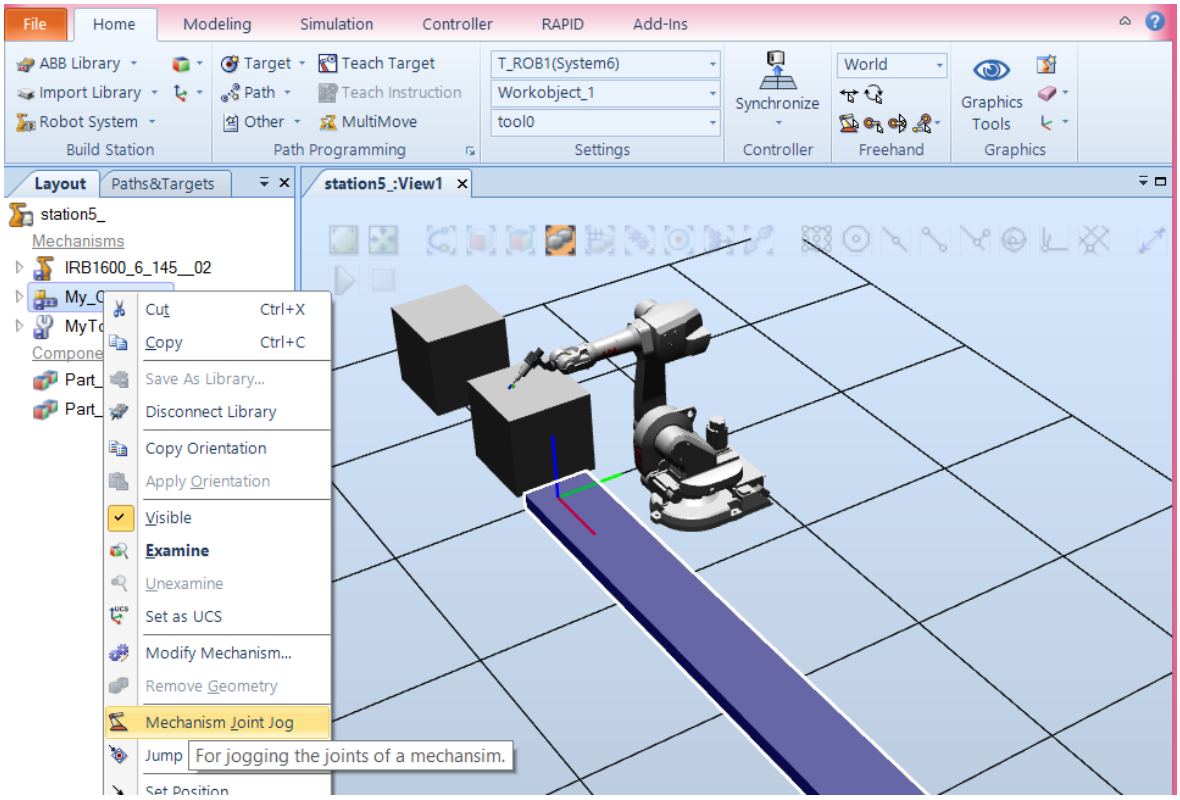
Below move both parts to the right:

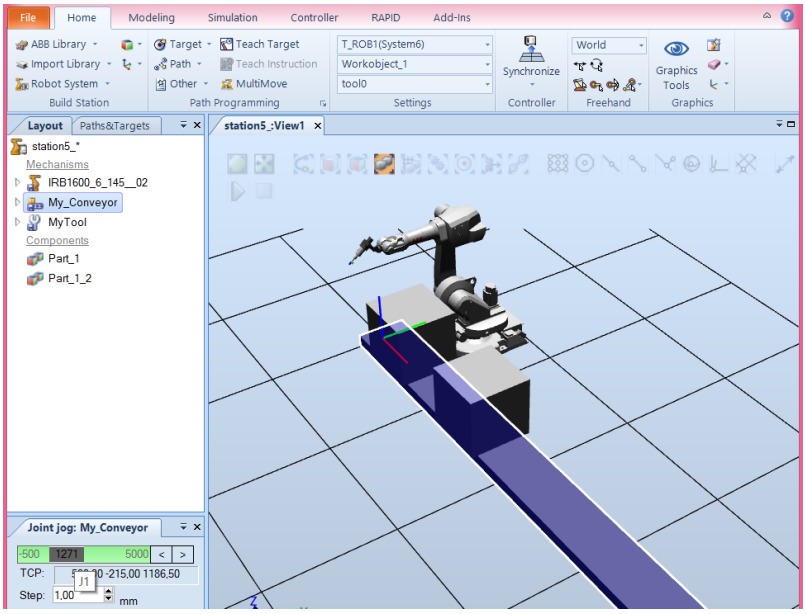


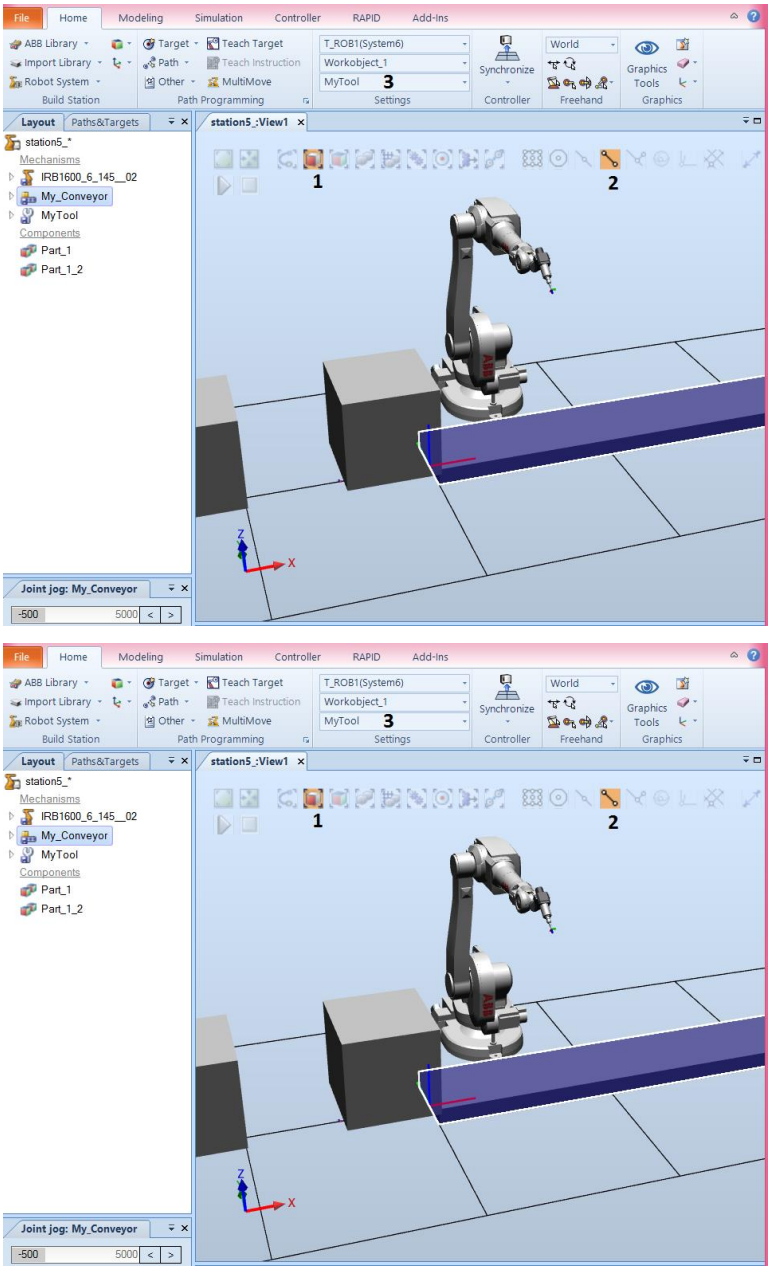
After this select also Part_1_2, click Add and OK. After this you can call the Encoder Unit:


File
Home
Modeling
Simulation
Controller
RAPID
Add-Ins
ABB Library
Import Library
Robot System
Build Station
Target
Teach Target
Path
Teach Instruction
Other
MultiMove
Path Programming
T_ROB1(System6)
Workobject_1
tool0
Settings
Synchronize
Controller
World
Freehand
Graphics Tools
Graphics
Layout
Paths&Targets
station5_
Station Elements
System6
T_R
station5_:View1
Synchronize to VC...
Synchronize to Station...
Restart
Shutdown
Virtual FlexPendant Ctrl+F5
Activate Mechanical Units...
Edit System...
Encoder Unit...
Task Frames...
Active
For configuring a conveyor encoder unit.

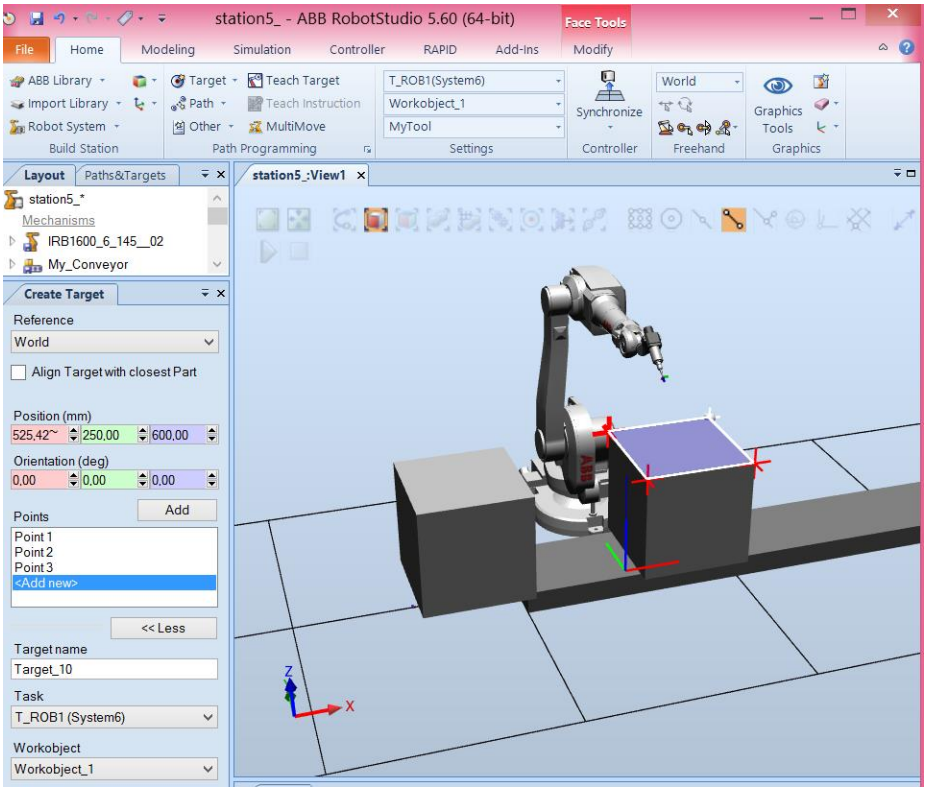

File
Home
Modeling
Simulation
Controller
RAPID
Add-Ins
ABB Library
Import Library
Robot System
Build Station
Target
Teach Target
Path
Teach Instruction
Other
MultiMove
Path Programming
T_ROB1(System6)
Workobject_1
tool0
Settings
Synchronize
Controller
World
Freehand
Graphics Tools
Graphics
Layout
Paths&Targets
station5_:View1
station5_
Station Elements
System6
T_ROB1
Tooldata
Workobjects &
wobj0
Workobjec
Paths
Configure Conveyor Encoder Units: System6
Mechanical Unit: CNV1
Parameters
Queue Tracking Distance (mm) 0
Maximum Distance (mm) 20000,00
Start Window Width (mm) 10000,00
Minimum Distance (mm) -600,00
Synchronization Switch
z
y
x
Base Frame (conveyor)
Conveyor Direction
Minimum Dist
Start Window Width
Queue Track. Dist
Maximum Distance
0.0 m
OK
Cancel

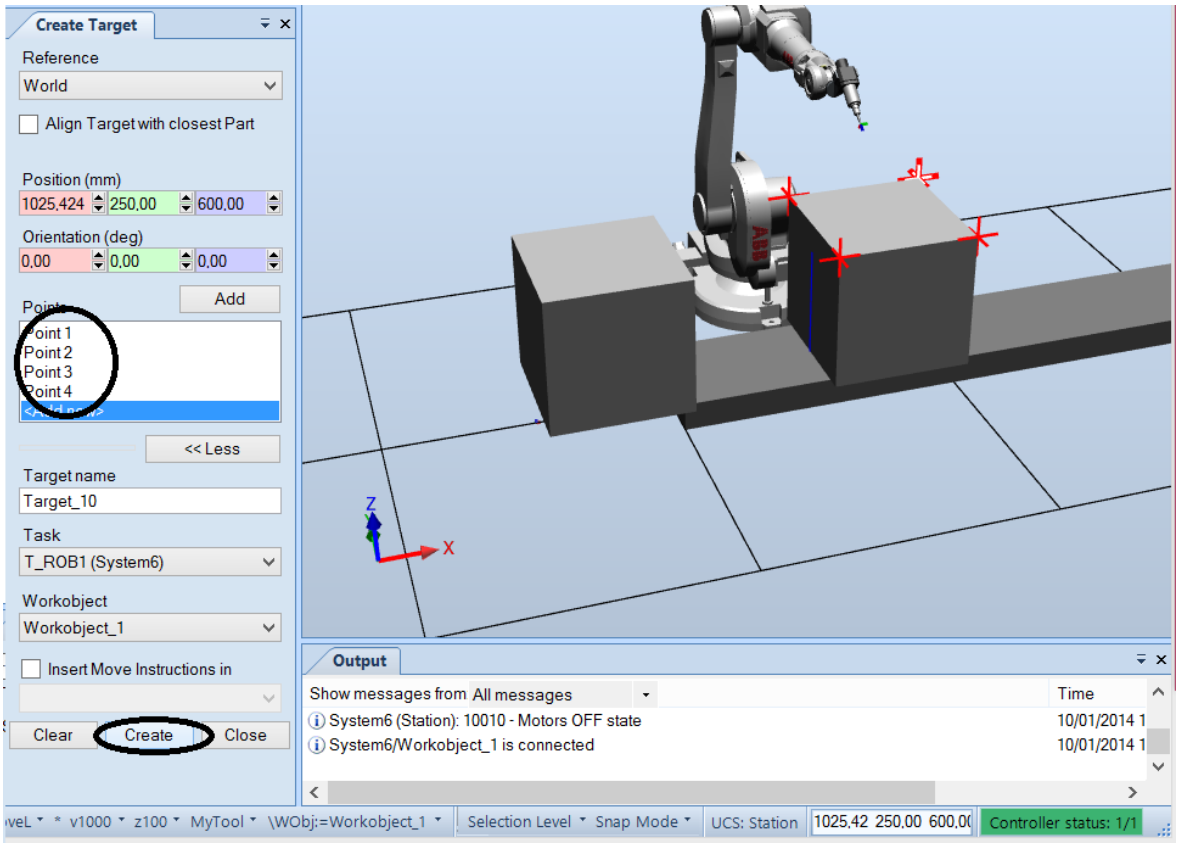

File
Home
Modeling
Simulation
Controller
RAPID
Add-Ins
ABB Library
Import Library
Robot System
Build Station
Target
Teach Target
Path
Teach Instruction
Other
MultiMove
Path Programming
T_ROB1(System6)
Workobject_1
tool0
Settings
Synchronize
Controller
World
Freehand
Graphics Tools
Graphics
Layout
Paths&Targets
station5_
Mechanisms
IRB1600_6_145__02
Cut Ctrl+X
Copy Ctrl+C
Save As Library...
Disconnect Library
Copy Orientation
Apply Orientation
Visible
Examine
Unexamine
Set as UCS
Modify Mechanism...
Remove Geometry
Mechanism Joint Jog
Jump
For jogging the joints of a mechansim.
station5_:View1

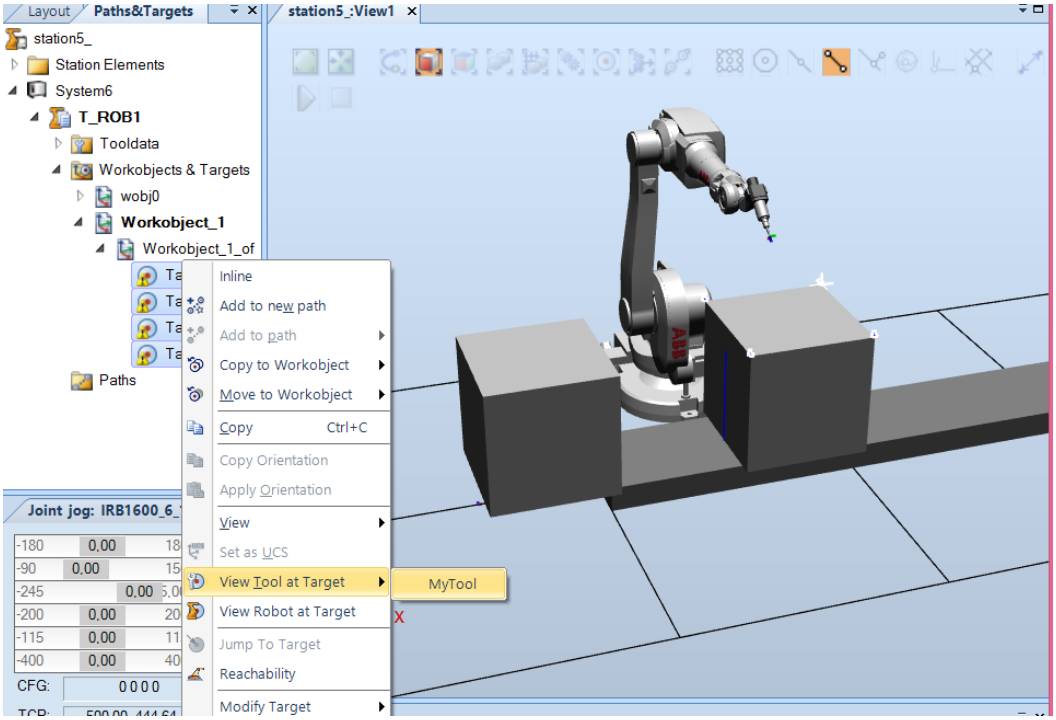

File
Home
Modeling
Simulation
Controller
RAPID
Add-Ins
Layout
Paths&Targets
station5_*
Mechanisms
IRB1600_6_145__02
My_Conveyor
MyTool
Components
Part_1
Part_1_2
station5_:View1
Joint jog: My_Conveyor
TCP:
Step:
mm

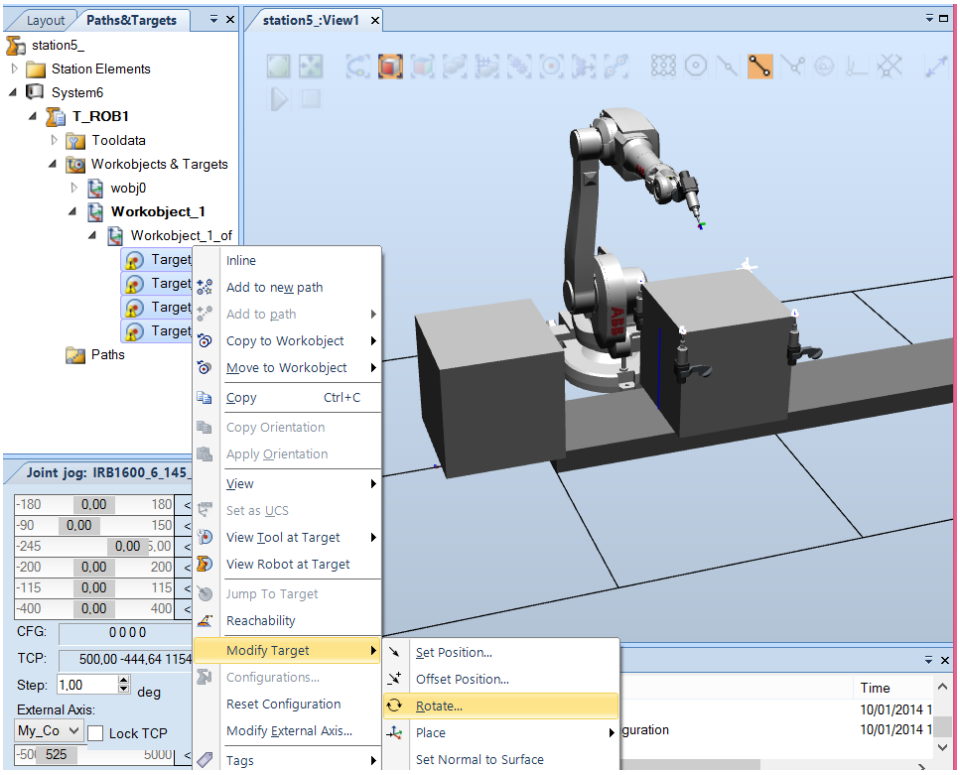

File
Home
Modeling
Simulation
Controller
RAPID
Add-Ins
ABB Library
Import Library
Robot System
Build Station
Target
Path
Other
Teach Target
Teach Instruction
MultiMove
Path Programming
T_ROB1(System6)
Workobject_1
MyTool
3
Settings
Synchronize
Controller
World
Freehand
Graphics Tools
Graphics
Layout
Paths&Targets
station5_*
Mechanisms
IRB1600_6_145__02
My_Conveyor
MyTool
Components
Part_1
Part_1_2
station5_:View1
1
2
Joint jog: My_Conveyor
-500
5000

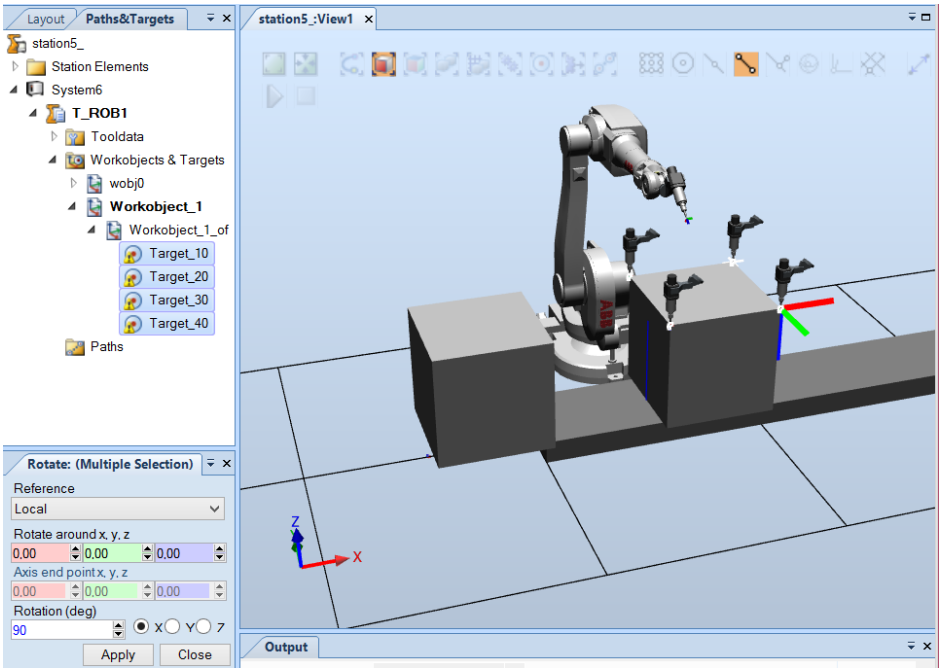
File
Home
Modeling
Simulation
Controller
RAPID
Add-Ins
ABB Library
Import Library
Robot System
Build Station
Target
Path
Other
Teach Target
Teach Instruction
MultiMove
Path Programming
T_ROB1(System6)
Workobject_1
MyTool
3
Settings
Synchronize
Controller
World
Freehand
Graphics Tools
Graphics
Layout
Paths&Targets
station5_*
Mechanisms
IRB1600_6_145__02
My_Conveyor
MyTool
Components
Part_1
Part_1_2
station5_:View1
1
2
Joint jog: My_Conveyor
-500
5000

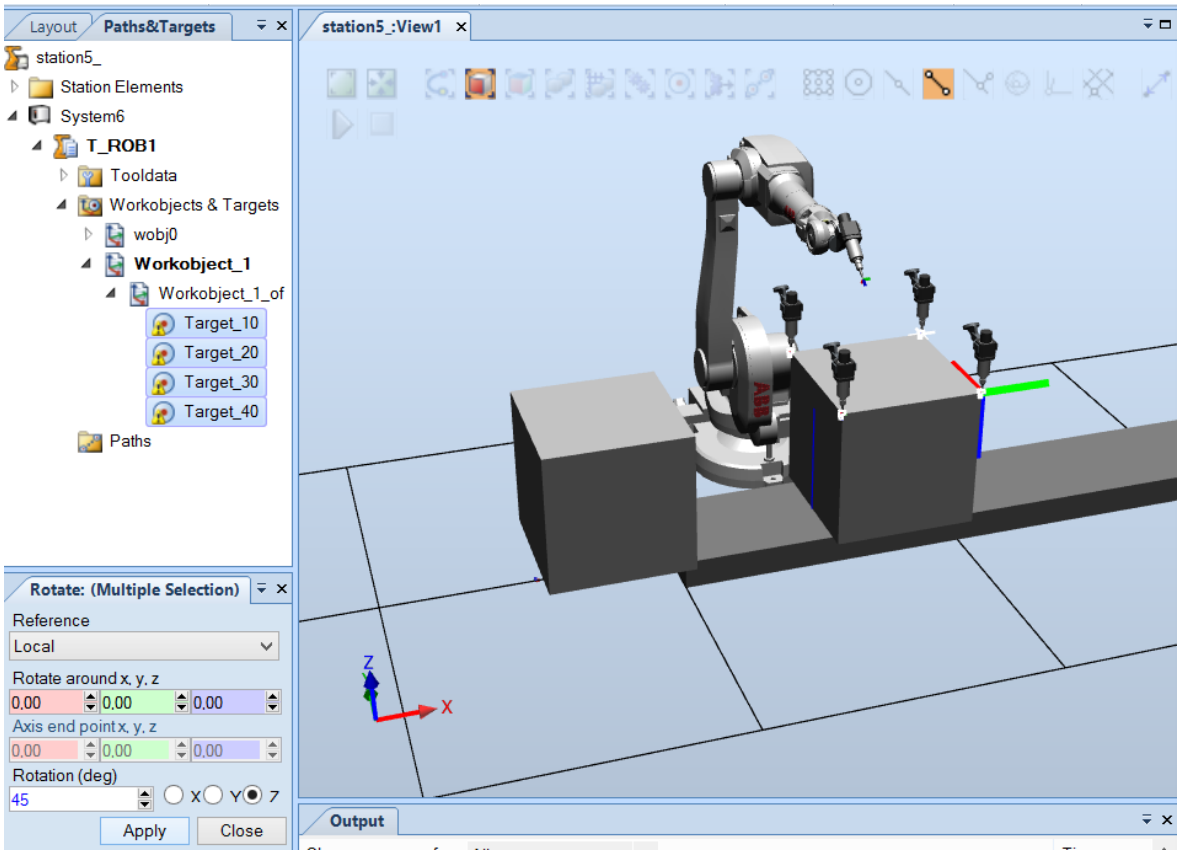

station5_ - ABB RobotStudio 5.60 (64-bit)
Face Tools
File
Home
Modeling
Simulation
Controller
RAPID
Add-Ins
Modify
ABB Library
Import Library
Robot System
Build Station
Target
Path
Other
Teach Target
Teach Instruction
MultiMove
Path Programming
T_ROB1(System6)
Workobject_1
MyTool
Settings
Synchronize
Controller
World
Freehand
Graphics Tools
Graphics
Layout
Paths&Targets
station5_*
Mechanisms
IRB1600_6_145__02
My_Conveyor
station5_:View1
Create Target
Reference
World
Align Target with closest Part
Position (mm)
525,42
250,00
600,00
Orientation (deg)
0,00
0,00
0,00
Points
Add
Point 1
Point 2
Point 3
<Add new>
<< Less
Target name
Target_10
Task
T_ROB1 (System6)
Workobject
Workobject_1
Z
X

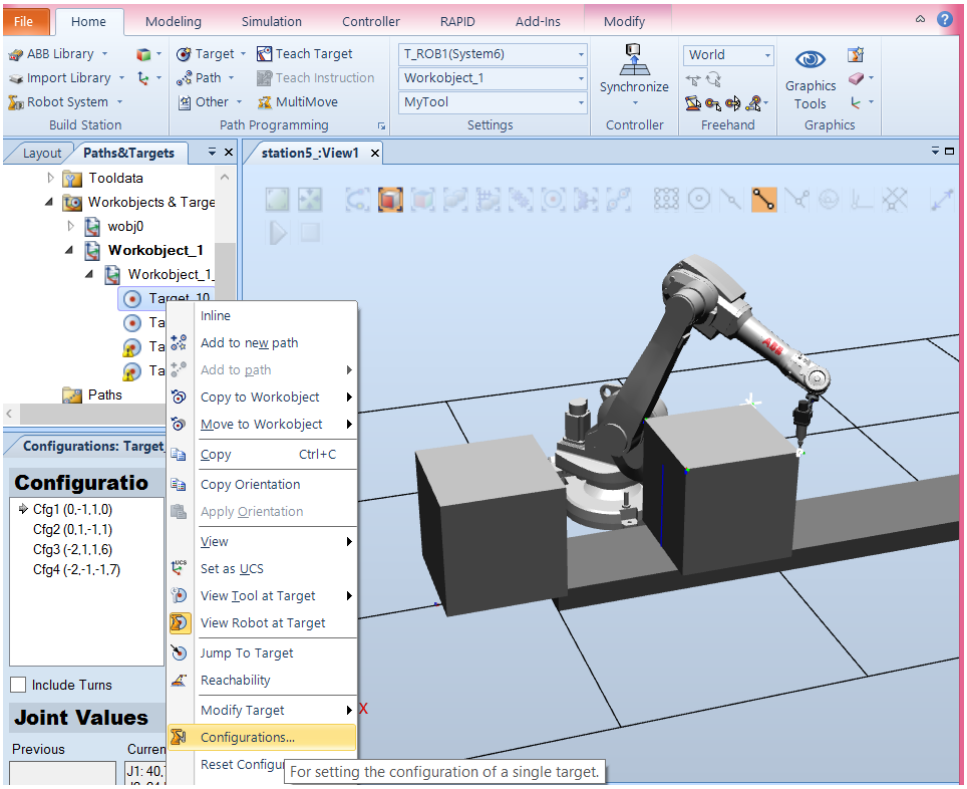

Create Target
Reference
World
Align Target with closest Part
Position (mm)
1025,424
250,00
600,00
Orientation (deg)
0,00
0,00
0,00
Add
Point 1
Point 2
Point 3
Point 4
<< Less
Target name
Target_10
Task
T_ROB1 (System6)
Workobject
Workobject_1
Insert Move Instructions in
Clear
Create
Close
Output
Show messages from All messages
Time
System6 (Station): 10010 - Motors OFF state
10/01/2014 1
System6/Workobject_1 is connected
10/01/2014 1
v1000
z100
MyTool
\WObj:=Workobject_1
Selection Level
Snap Mode
UCS: Station
1025,42 250,00 600,0(
Controller status: 1/1
Z
X

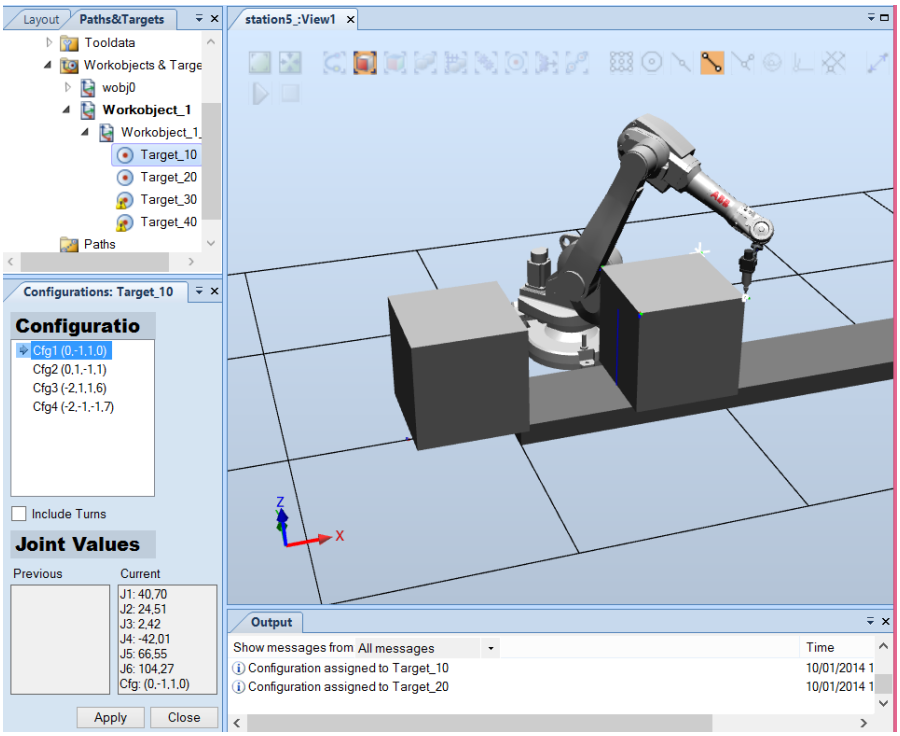

Layout
Paths&Targets
station5_
Station Elements
System6
T_ROB1
Tooldata
Workobjects & Targets
wobj0
Workobject_1
Workobject_1_of
Paths
station5_:View1
Inline
Add to new path
Add to path
Copy to Workobject
Move to Workobject
Copy Ctrl+C
Copy Orientation
Apply Orientation
View
Set as UCS
View Tool at Target
MyTool
View Robot at Target
Jump To Target
Reachability
Modify Target
Joint jog: IRB1600_6_
CFG:
0 0 0 0


Layout
Paths&Targets
station5_
Station Elements
System6
T_ROB1
Tooldata
Workobjects & Targets
wobj0
Workobject_1
Workobject_1_of
Target
Target
Target
Target
Paths
station5_:View1
Inline
Add to new path
Add to path
Copy to Workobject
Move to Workobject
Copy Ctrl+C
Copy Orientation
Apply Orientation
View
Set as UCS
View Tool at Target
View Robot at Target
Jump To Target
Reachability
Modify Target
Configurations...
Reset Configuration
Modify External Axis...
Tags
Set Position...
Offset Position...
Rotate...
Place
Set Normal to Surface
Joint jog: IRB1600_6_145_
CFG:
0 0 0 0
TCP:
500,00 -444,64 1154
Step:
1,00
deg
External Axis:
Lock TCP
Time
10/01/2014 1
10/01/2014 1

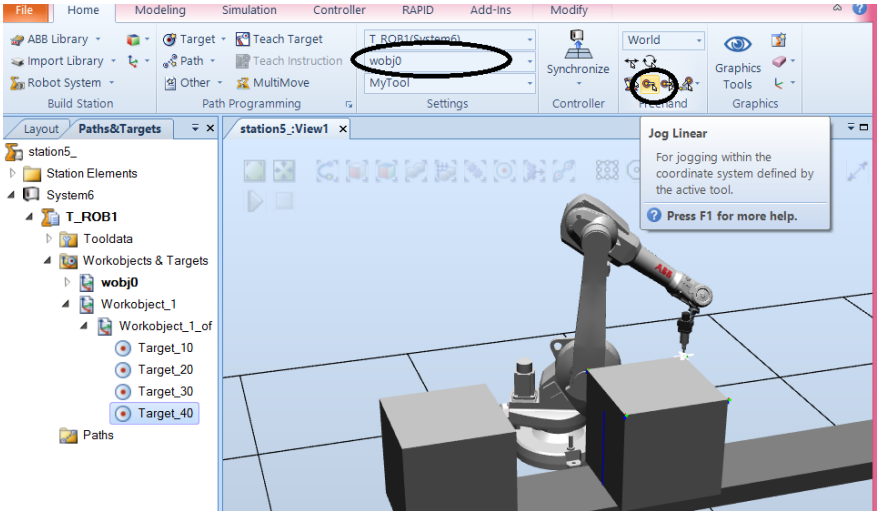

Layout
Paths&Targets
station5_
Station Elements
System6
T_ROB1
Tooldata
Workobjects & Targets
wobj0
Workobject_1
Workobject_1_of
Target_10
Target_20
Target_30
Target_40
Paths
station5_:View1
Rotate: (Multiple Selection)
Reference
Local
Rotate around x, y, z
0,00
0,00
0,00
Axis end point x, y, z
0,00
0,00
0,00
Rotation (deg)
90
X
Y
Z
Apply
Close
Output

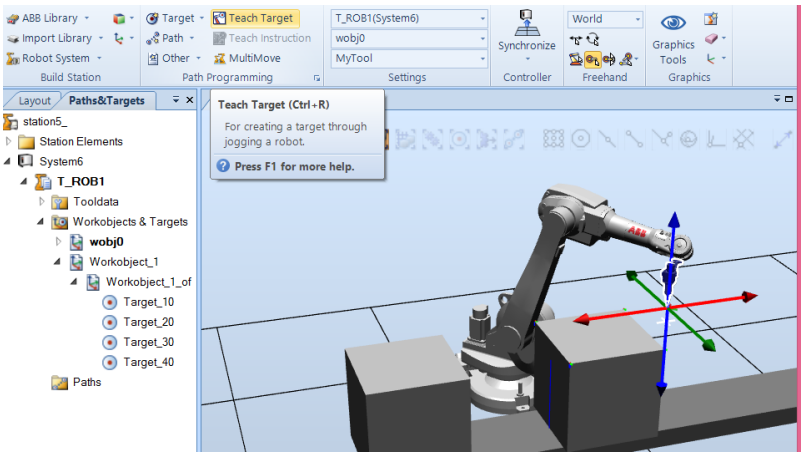

Layout
Paths&Targets
station5_
Station Elements
System6
T_ROB1
Tooldata
Workobjects & Targets
wobj0
Workobject_1
Workobject_1_of
Target_10
Target_20
Target_30
Target_40
Paths
station5_:View1
Rotate: (Multiple Selection)
Reference
Local
Rotate around x, y, z
0,00
0,00
0,00
Axis end point x, y, z
0,00
0,00
0,00
Rotation (deg)
45
X
Y
Z
Apply
Close
Output



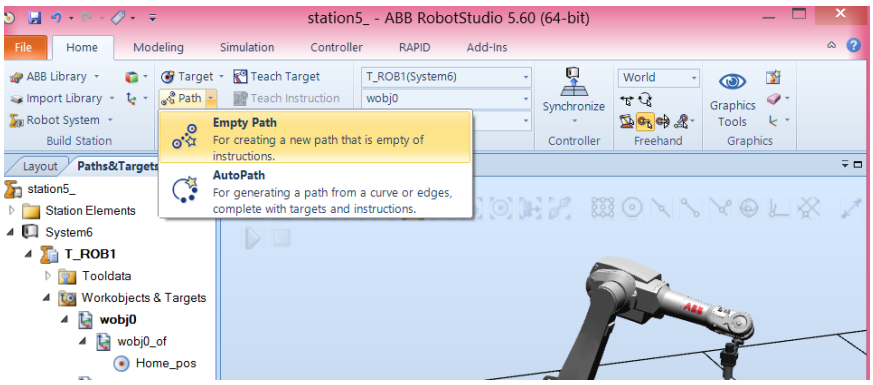


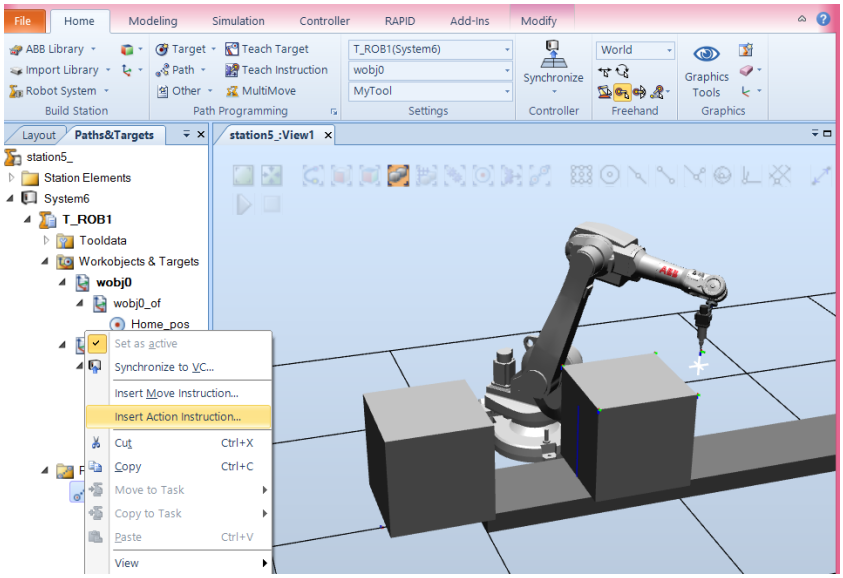
Create a home position for the robot:

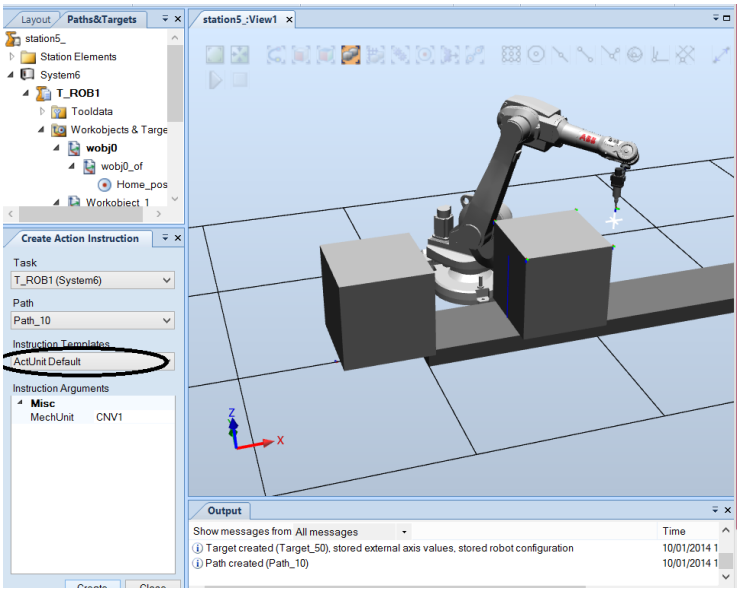


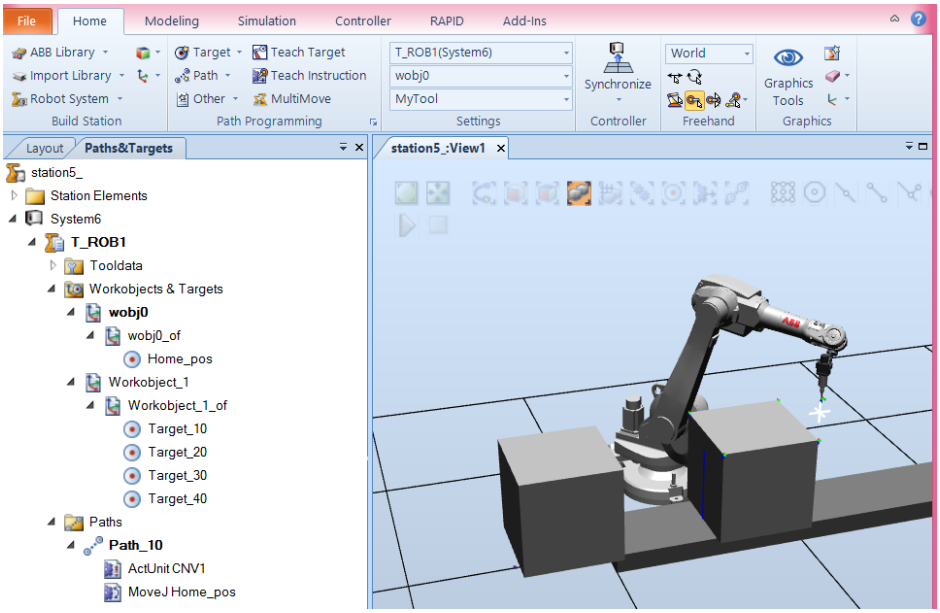


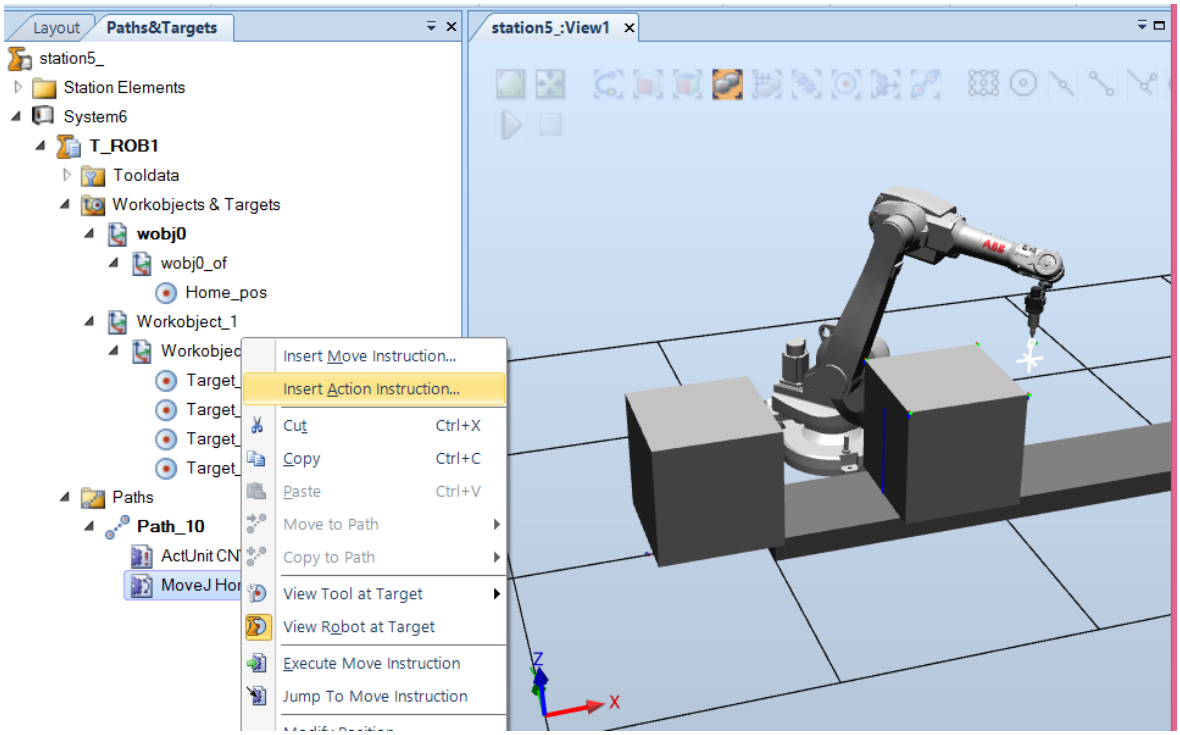
Create an empty path:

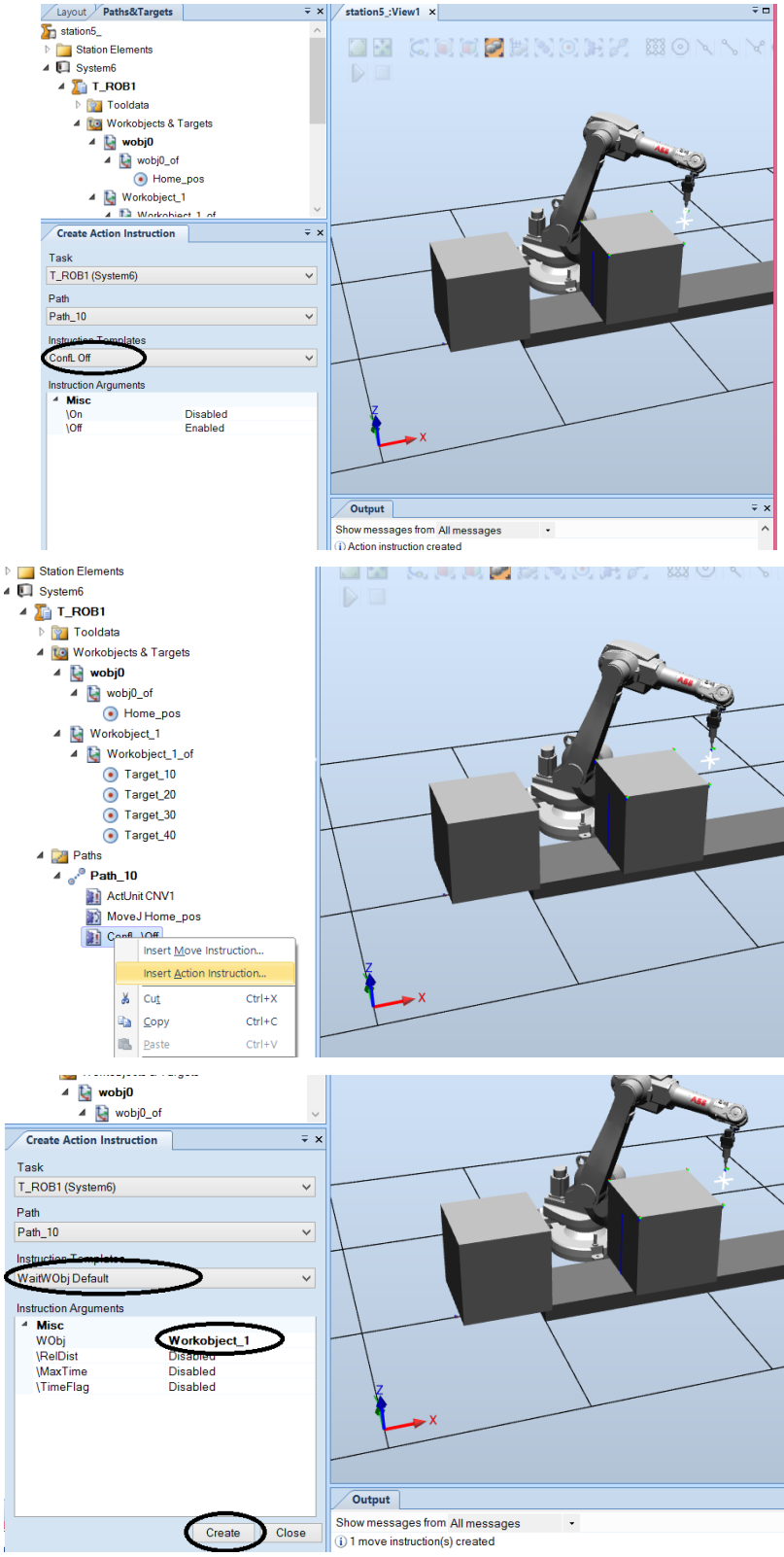


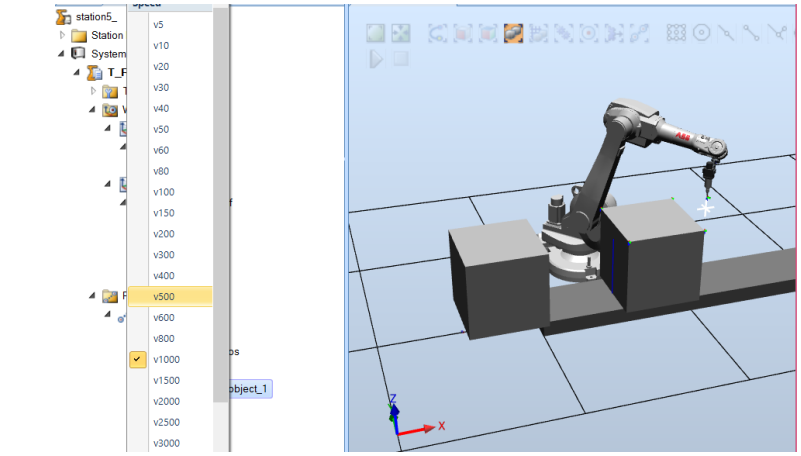

File
Home
Modeling
Simulation
Controller
RAPID
Add-Ins
Modify
ABB Library
Import Library
Robot System
Build Station
Target
Path
Other
Teach Target
Teach Instruction
MultiMove
Path Programming
T_ROB1(System6)
wobj0
MyTool
Settings
Synchronize
Controller
World
Freehand
Graphics Tools
Graphics
Layout
Paths&Targets
station5_
Station Elements
System6
T_ROB1
Tooldata
Workobjects & Targets
wobj0
wobj0_of
Home_pos
Set as active
Synchronize to VC...
Insert Move Instruction...
Insert Action Instruction...
Cut Ctrl+X
Copy Ctrl+C
Move to Task
Copy to Task
Paste Ctrl+V
View
station5_:View1

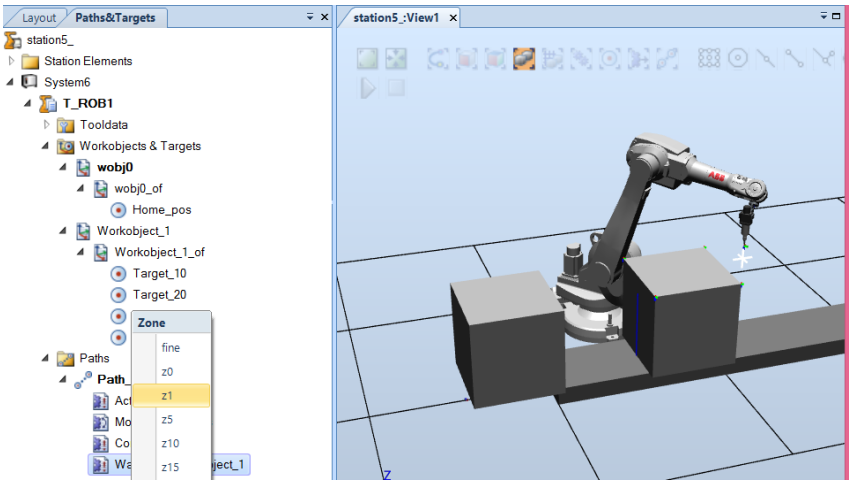

Layout
Paths&Targets
station5_
Station Elements
System6
T_ROB1
Tooldata
Workobjects & Targe
wobj0
wobj0_of
Home_pos
Workobject_1
Create Action Instruction
Task
T_ROB1 (System6)
Path
Path_10
Instruction Templates
ActUnit Default
Instruction Arguments
Misc
MechUnit CNV1
station5_:View1
Output
Show messages from All messages
Time
Target created (Target_50), stored external axis values, stored robot configuration
10/01/2014 1
Path created (Path_10)
10/01/2014 1


File
Home
Modeling
Simulation
Controller
RAPID
Add-Ins
ABB Library
Import Library
Robot System
Build Station
Target
Path
Other
Teach Target
Teach Instruction
MultiMove
Path Programming
T_ROB1(System6)
wobj0
MyTool
Settings
Synchronize
Controller
World
Freehand
Graphics Tools
Graphics
Layout
Paths&Targets
station5_
Station Elements
System6
T_ROB1
Tooldata
Workobjects & Targets
wobj0
wobj0_of
Home_pos
Workobject_1
Workobject_1_of
Target_10
Target_20
Target_30
Target_40
Paths
Path_10
ActUnit CNV1
MoveJ Home_pos
station5_:View1

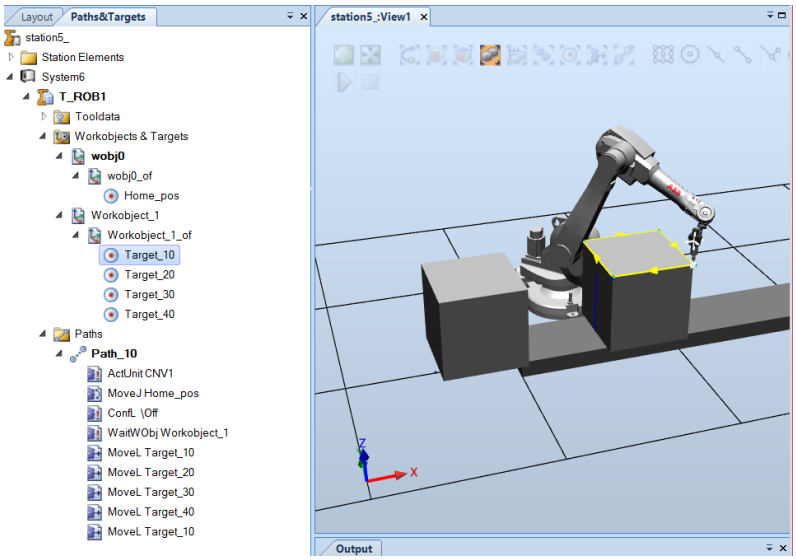

Layout
Paths&Targets
station5_
Station Elements
System6
T_ROB1
Tooldata
Workobjects & Targets
wobj0
wobj0_of
Home_pos
Workobject_1
Paths
Path_10
station5_:View1
Insert Move Instruction...
Insert Action Instruction...
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Move to Path
Copy to Path
View Tool at Target
View Robot at Target
Execute Move Instruction
Jump To Move Instruction
Z
X

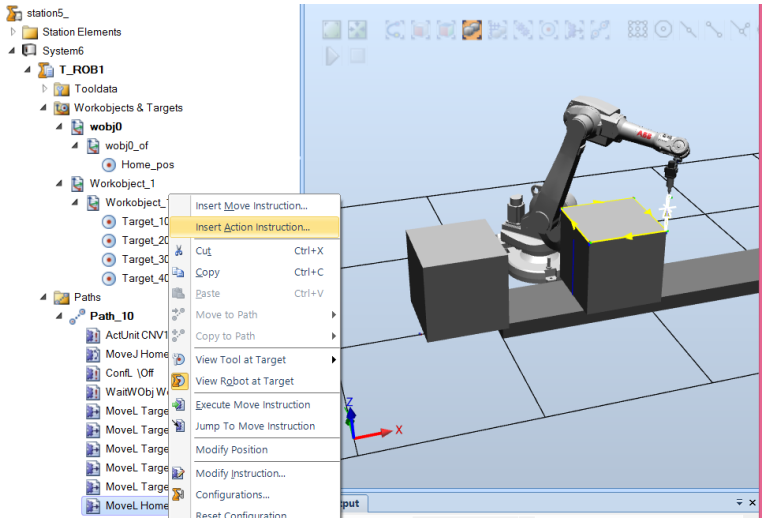

Layout
Paths&Targets
station5_
Station Elements
System6
T_ROB1
Tooldata
Workobjects & Targets
wobj0
wobj0_of
Home_pos
Workobject_1
Create Action Instruction
Task
T_ROB1 (System6)
Path
Path_10
Instruction Templates
ConfL Off
Instruction Arguments
Misc
\On Disabled
\Off Enabled
station5_:View1
Output
Show messages from All messages
Action instruction created
Station Elements
System6
T_ROB1
Tooldata
Workobjects & Targets
wobj0
wobj0_of
Home_pos
Workobject_1
Workobject_1_of
Target_10
Target_20
Target_30
Target_40
Paths
Path_10
ActUnit CNV1
MoveJ Home_pos
Insert Move Instruction...
Insert Action Instruction...
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
wobj0
wobj0_of
Create Action Instruction
Task
T_ROB1 (System6)
Path
Path_10
Instruction Templates
WaitWObj Default
Instruction Arguments
Misc
WObj Workobject_1
\RelDist Disabled
\MaxTime Disabled
\TimeFlag Disabled
Create
Close
Output
Show messages from All messages
1 move instruction(s) created

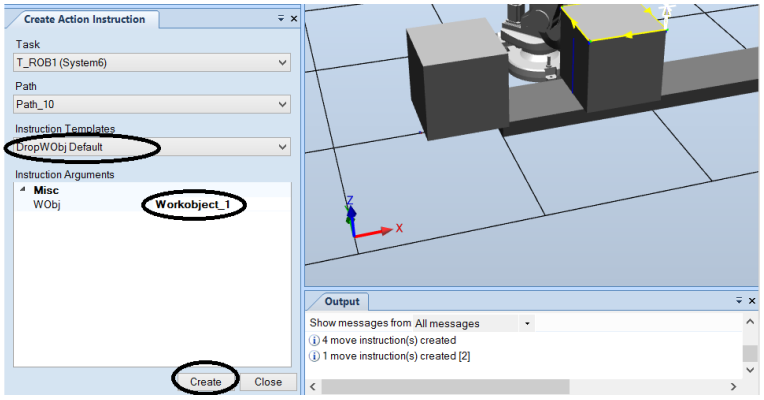


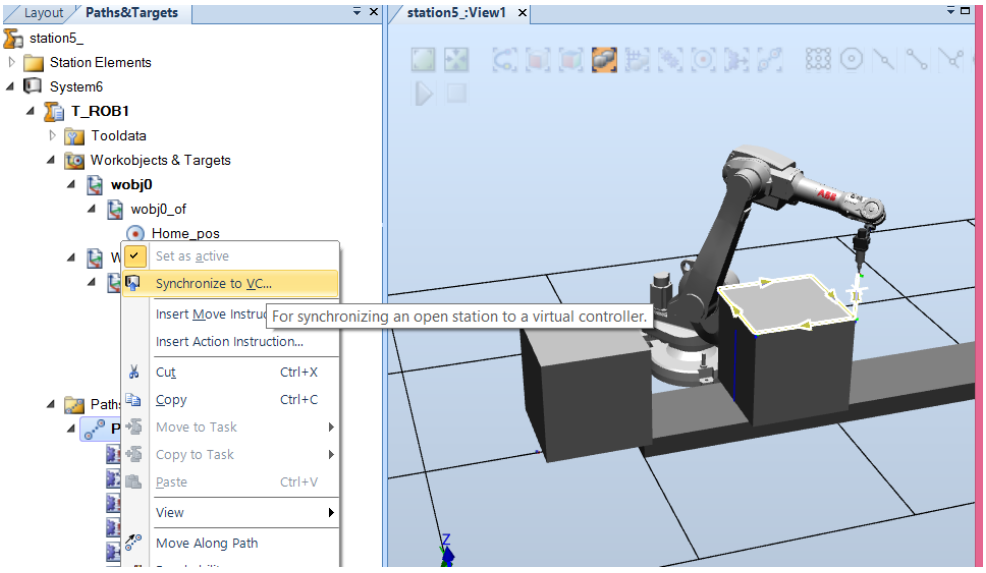


Drag the targets to Path_10:

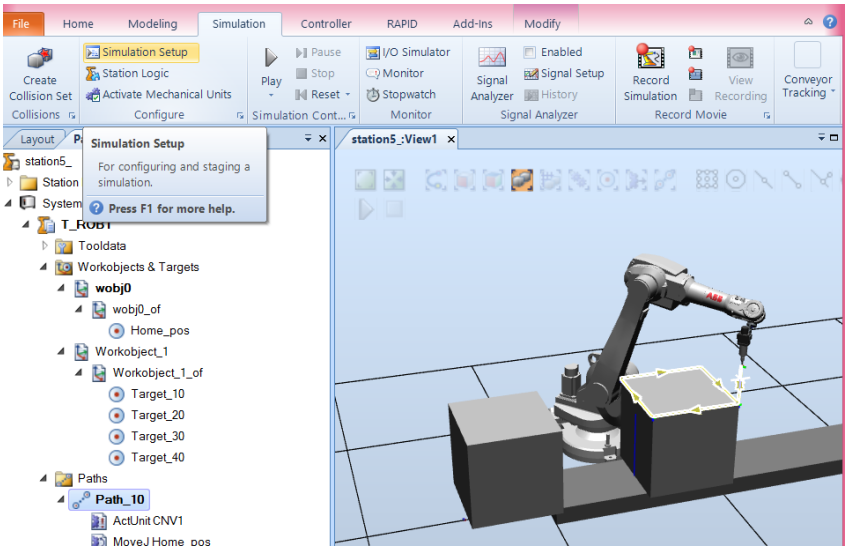


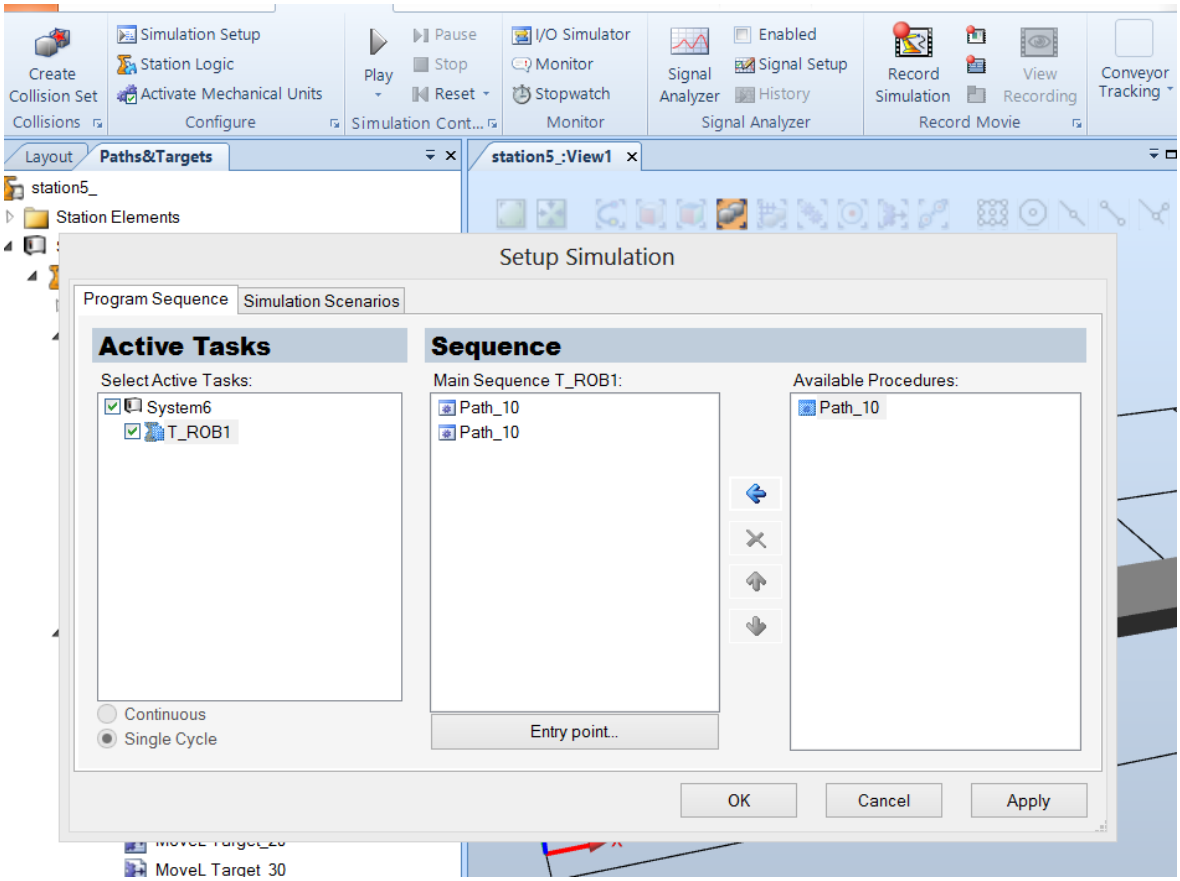

station5_
Station Elements
System6
T_ROB1
Tooldata
Workobjects & Targets
wobj0
wobj0_of
Home_pos
Workobject_1
Paths
Path_10
Insert Move Instruction...
Insert Action Instruction...
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Move to Path
Copy to Path
View Tool at Target
View Robot at Target
Execute Move Instruction
Jump To Move Instruction
Modify Position
Modify Instruction...
Configurations...
Reset Configuration

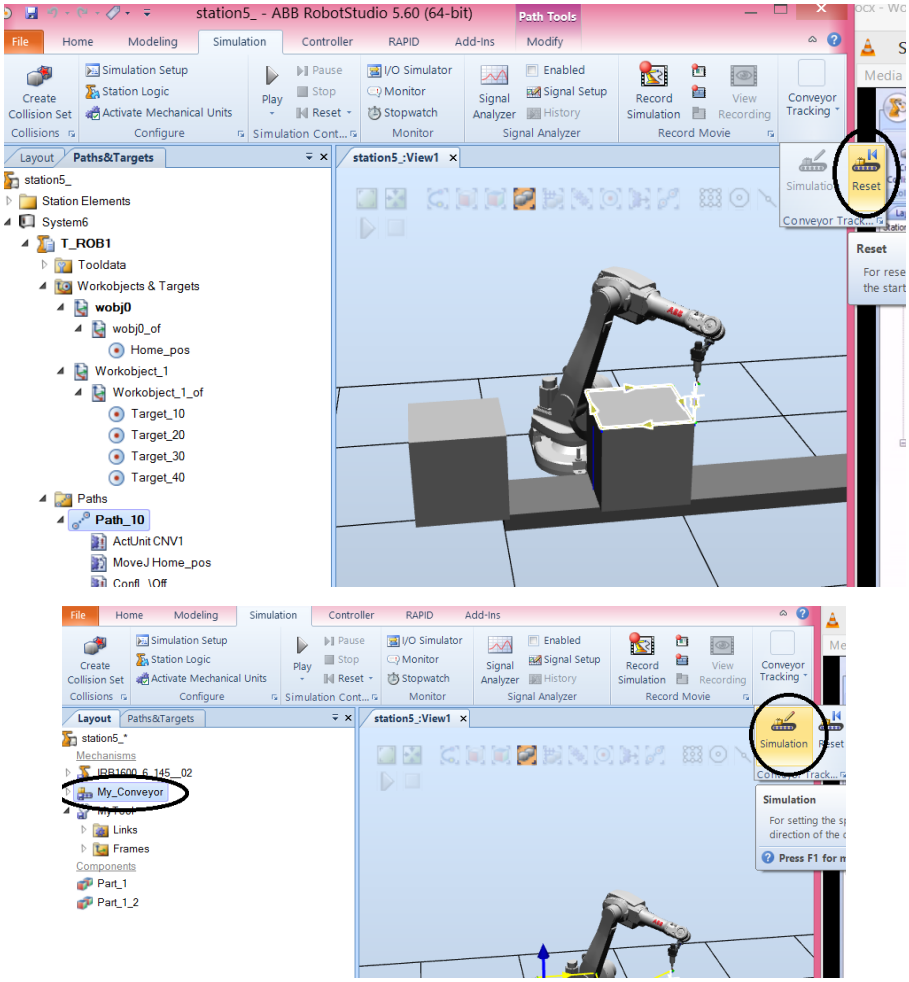

Create Action Instruction
Task
T_ROB1 (System6)
Path
Path_10
Instruction Templates
DropWObj Default
Instruction Arguments
Misc
WObj
Workobject_1
Create
Close
Output
Show messages from All messages
4 move instruction(s) created
1 move instruction(s) created [2]


Layout
Paths&Targets
station5_:View1
station5_
Station Elements
System6
T_ROB1
Tooldata
Workobjects & Targets
wobj0
wobj0_of
Home_pos
Set as active
Synchronize to VC...
For synchronizing an open station to a virtual controller.
Insert Action Instruction...
Cut Ctrl+X
Copy Ctrl+C
Move to Task
Copy to Task
Paste Ctrl+V
View
Move Along Path

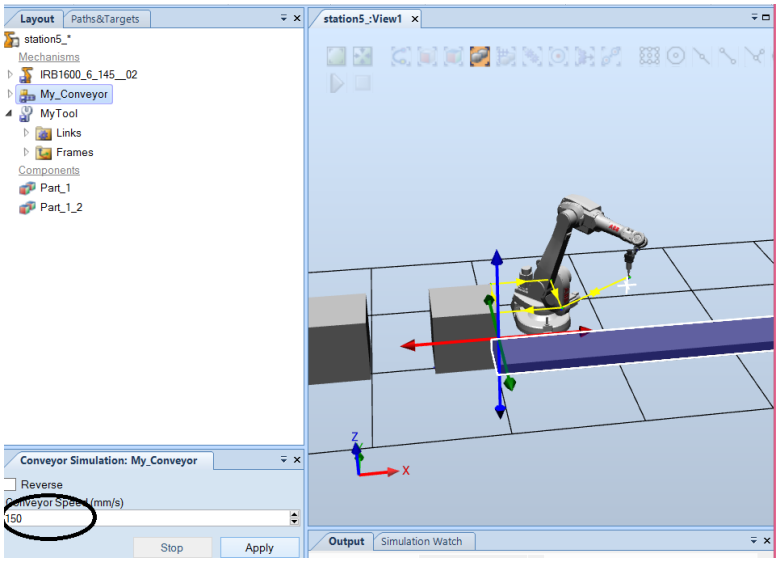
Simulate:


File
Home
Modeling
Simulation
Controller
RAPID
Add-Ins
Modify
Create Collision Set
Simulation Setup
Station Logic
Activate Mechanical Units
Play
Pause
Stop
Reset
I/O Simulator
Monitor
Stopwatch
Signal Analyzer
Enabled
Signal Setup
History
Record Simulation
View Recording
Conveyor Tracking
Collisions
Configure
Simulation Cont...
Monitor
Signal Analyzer
Record Movie
Layout
Simulation Setup
For configuring and staging a simulation.
Press F1 for more help.
station5_
Station
System
T_ROB1
Tooldata
Workobjects & Targets
wobj0
wobj0_of
Home_pos
Workobject_1
Workobject_1_of
Target_10
Target_20
Target_30
Target_40
Paths
Path_10
ActUnit CNV1
MoveJ Home pos
station5_:View1


Create Collision Set
Collisions
Simulation Setup
Station Logic
Activate Mechanical Units
Configure
Play
Pause
Stop
Reset
Simulation Cont...
I/O Simulator
Monitor
Stopwatch
Monitor
Signal Analyzer
Enabled
Signal Setup
History
Signal Analyzer
Record Simulation
View Recording
Record Movie
Conveyor Tracking
Layout
Paths&Targets
station5_
Station Elements
station5_:View1
Setup Simulation
Program Sequence
Simulation Scenarios
Active Tasks
Select Active Tasks:
System6
T_ROB1
Sequence
Main Sequence T_ROB1:
Path_10
Path_10
Available Procedures:
Path_10
Continuous
Single Cycle
Entry point...
OK
Cancel
Apply
MoveL Target_30



If it is not working change the conveyor speed.



References in CAD based robotics:

Neto P., Mendes N: “Direct off-line robot programming via a common CAD package,” Robotics and Autonomous Systems, Elsevier, Vol. 61, No. 8, pp. 896-910, **2013**

Neto P., Mendes N., Araújo R., Pires J.N. and Moreira A.P.: “High-level robot programming based on CAD: dealing with unpredictable environments,” Industrial Robot, Emerald, Vol. 39, No. 3, pp. 294-303, **2012**

Neto P.: “Off-line programming and simulation from CAD drawings: robot-assisted sheet metal bending,” Annual Conference of the IEEE Industrial Electronics Society, IECON 2013, pp. 4233-4238, Vienna, Austria, **2013**