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 .................................................................................................................. 1
2. MODULE I .............................................................................................................................. 2  
   2.1. Robotic arms and positioning ......................................................................................... 3  
   2.2. Controller ...................................................................................................................... 7  
   2.3. Tools ............................................................................................................................ 10  
   2.4. Workobjects .................................................................................................................. 11  
   2.5. Targets ......................................................................................................................... 13  
   2.6. Paths ............................................................................................................................ 21  
   2.7. Simulation ..................................................................................................................... 23  
3. MODULE II ............................................................................................................................. 27  
   3.1. Importing geometries ...................................................................................................... 27  
   3.2. Workobjects and targets ................................................................................................. 30  
   3.3. RAPID .......................................................................................................................... 40  
   3.4. Time ............................................................................................................................. 44  
   3.5. Collisions ....................................................................................................................... 45  
   3.6. Curves .......................................................................................................................... 48  
4. Teach pendant ......................................................................................................................... 51  
5. Mechanisms ............................................................................................................................. 59  
   5.1. Conveyor ......................................................................................................................... 59  
   5.2. Creating a station ............................................................................................................ 62
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 how 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 thing 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

It is 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 reach 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.

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:
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 configure:
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.
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:
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:
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:
5. MECHANISMS

5.1. Conveyor

We will start by creating a mechanism, a conveyor.
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:
Box
By defining a corner point and dimensions.

Box from 3 points
By defining three corner points.
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:
Create a home position for the robot:
Create an empty path:
Pedro Neto
Pedro Neto
Drag the targets to Path_10:
Simulate:
If it is not working change the conveyor speed.
References in CAD based robotics:

