

RAPID2SERIAL MANUAL

Complete setup guide for Rapid2Serial V0.1.7

Abstract

This manual will cover the setup of the S4C controller and Rapid2Serial application on the PC. A troubleshooting guide can be found at the end of this manual, this covers communication problems, the most common made mistakes and solutions.

brian brocken
Brian.brocken@hotmail.be

Summary

- 1 INTRODUCTION 2**
- 2 SETTING UP THE S4C CONTROLLER 3**
 - 2.1 CREATING A NEW WORKOBJECT: WOBJ10 3
 - 2.2 CHANGE THE BAUD-RATE OF SIO1 9
 - 2.3 LOAD THE FULTST09 PROGRAM ON THE CONTROLLER..... 13
 - 2.3.1 *Check rotation settings*..... 15
- 3 SETTING UP THE PC AND RAPID2SERIAL PROGRAM..... 16**
 - 3.1 CONNECTION BETWEEN PC AND S4C CONTROLLER 16
 - 3.2 SETTING UP THE RAPID2SERIAL PROGRAM 17
 - 3.2.1 *Importing a rapid file*..... 17
 - 3.2.2 *Rapid file criteria* 17
 - 3.2.3 *Converting to a serial format* 17
 - 3.2.4 *Selecting COM-port and Baud-rate* 19
- 4 START SENDING DATA 20**
- 5 TROUBLESHOOTING GUIDE 21**
- 6 LIST OF FIGURES 23**

1 Introduction

This Rapid-2-Serial program can be used to send coordinates directly from a PC to the ABB S4C controller. The Rapid-2-Serial program currently only supports milling in 3-axis.

Why would you need this?

The S4C controller is technology from 1999 and comes with floppy drives. Even if you change out the Floppy driver for a USB one, the internal memory used to execute the program is still only 1.4mb. With this memory size you can only execute programs with max 15.000 lines of code. Anyone who ever worked with CNC-machines know that even the simplest programs exceed 15.000 lines. So that's why this application was created.

How to does it work?

This program comes with a RAPID program (FULTST09) that runs on the S4C controller. This RAPID program (FULTST09) is needed to communicate with the Rapid2Serial program, running on the PC.

In the Rapid2Serial program you're able to import the rapid code you would like the robot to execute. When the Rapid code is imported and you press the "convert to serial" button, the Rapid2Serial program will look for coordinates found in the rapid program, puts these in a serial format "1,Xval,Yval,Zval,ChecksumXYZ" and sends these to the S4C controller. It will not send anything else, like for example setting an output high, to the s4c controller, only coordinates.

Full functionality of the program will be discussed further in this manual.

It's recommended to use Fusion 360 to generate the milling files for the best results.

2 Setting up the S4C controller

2.1 Creating a new WorkObject: wobj10

Before we load the FULTST09 Rapid program onto the S4C controller we need to create a new WorkObject named wobj10. The FULTST09 program uses this wobj as the origin of the workpiece. You can also create this wobj after you loaded FULTST09 on the S4C controller but you'll initially get error "40165: Reference error", this will go away after you created wobj10.

Why is wobj10 needed?

The origin point of the workpiece in fusion 360 is X=0, Y=0, Z=0. If you would send the generated code directly to the robot without using an offset the robot would try to start from it's own center point (Base coordinate system). That's why we use a WorkObject: wobj10 (Object coordinates). The values setup in here are an offset from the Base coordinate system (see picture below). The WorkObject allows for flexibility in choosing the starting point of the toolpath.

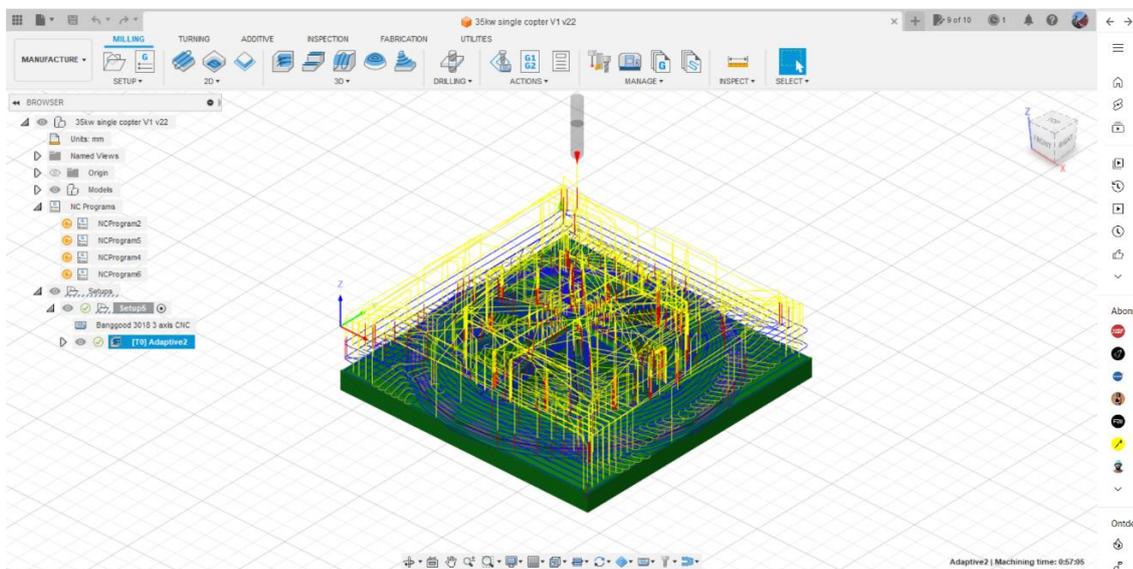


Figure 1: Fusion 360 origin point

Coordinate systems

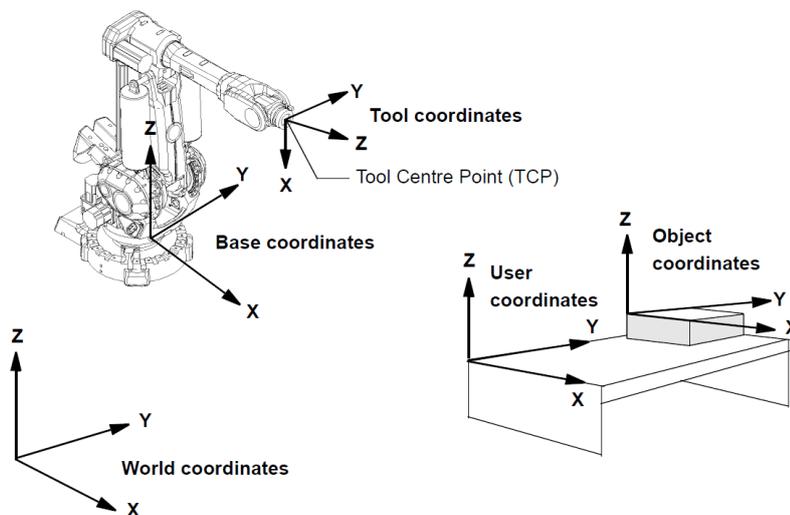
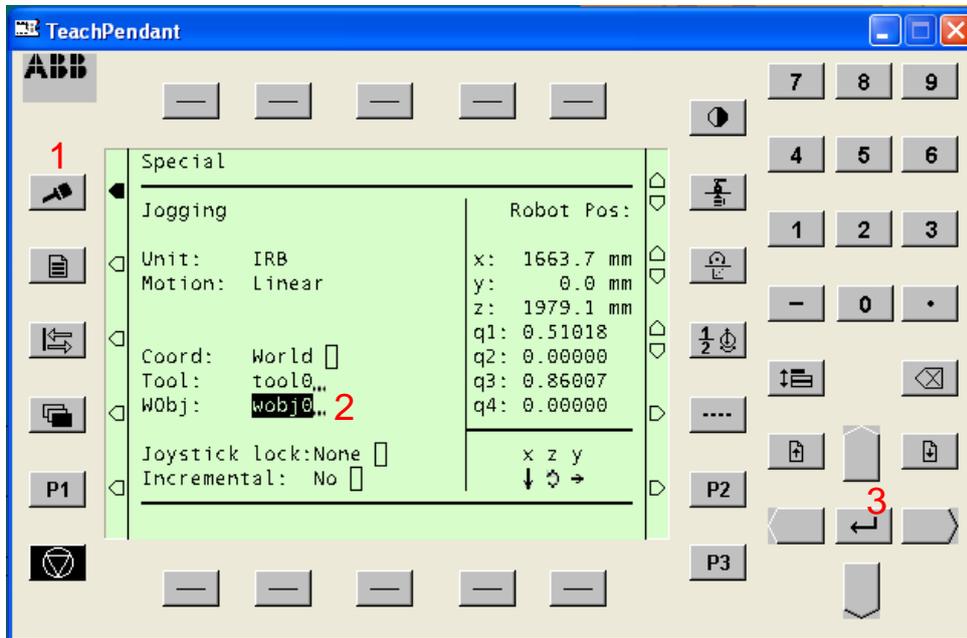


Figure 9 The coordinate systems, used to make jogging and off-line programming easier.

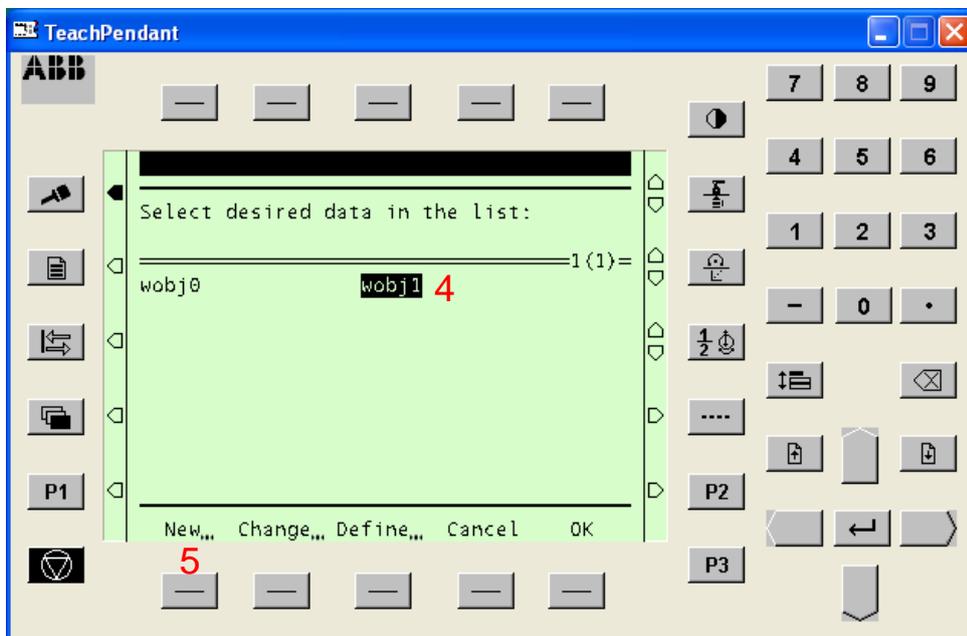
Figure 2: ABB IRB6400 coordinate systems

You can follow the steps below to create WorkObject named wobj10.

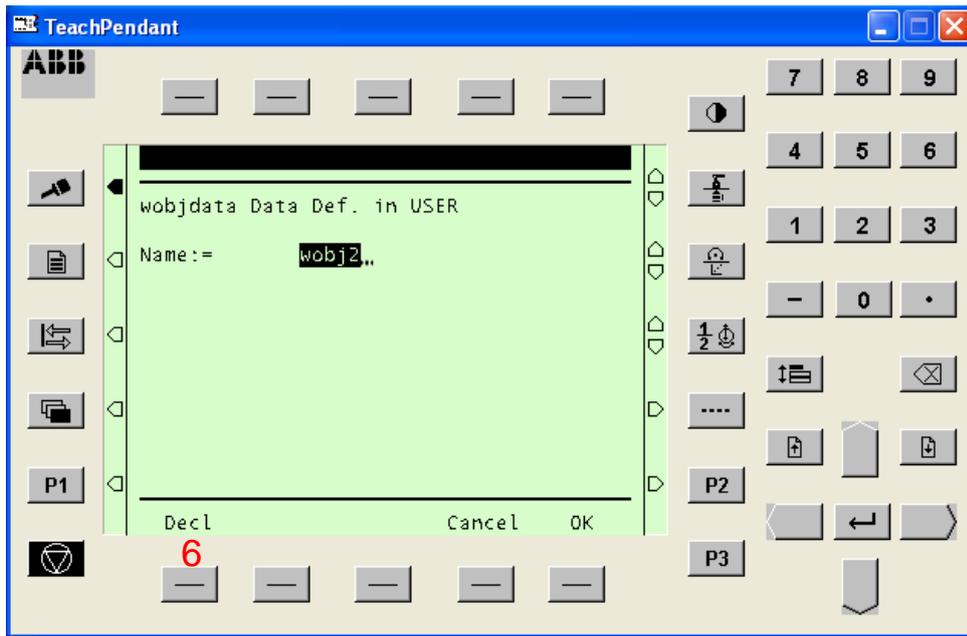
1. Go to Jogging
2. Stand on wobj0
3. Press enter



4. Stand on wobj1 (if you stand on wobj0 you won't be able to create a new wobj)
5. Press "New"

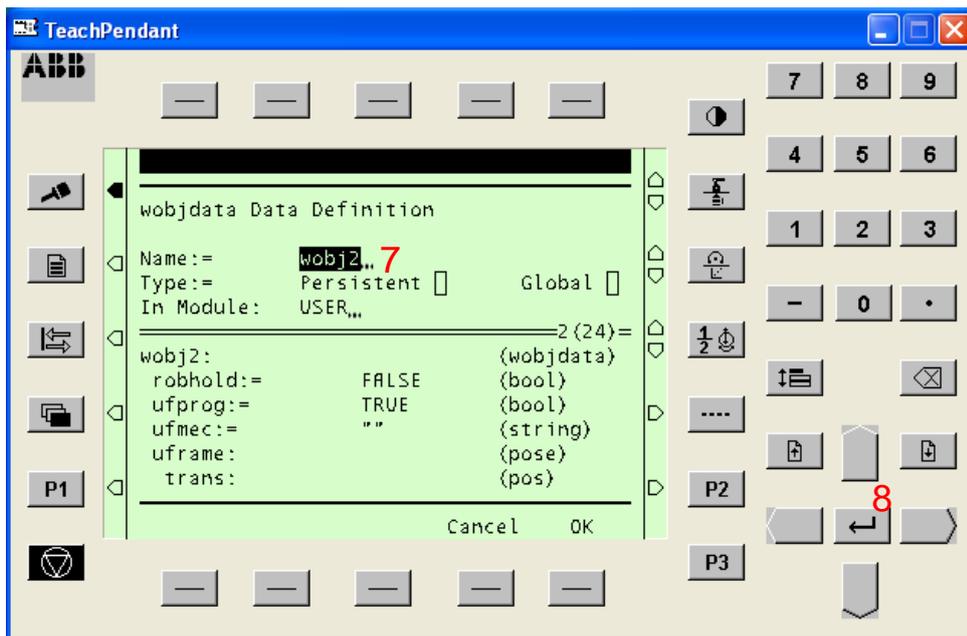


6. Press "Decl"

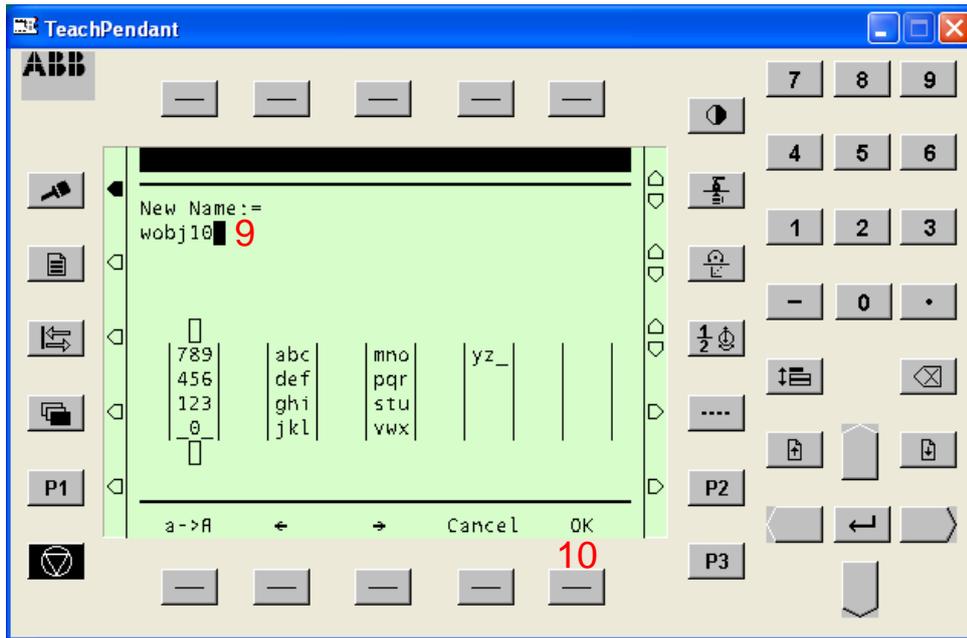


7. Stand on "wobj2" (can have another number if you already have other wobj, see picture below)

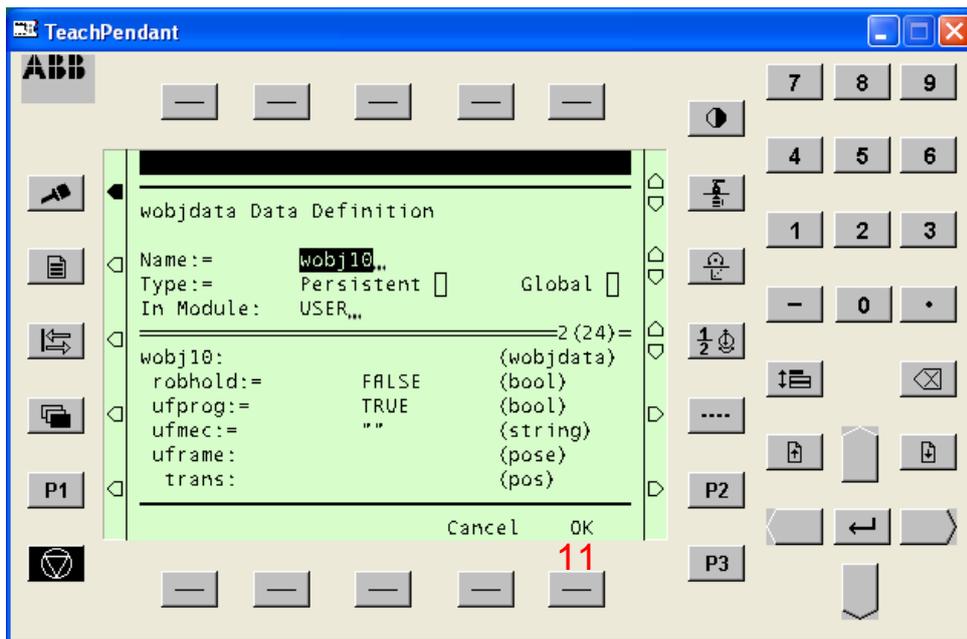
8. Press enter



9. Change the name to "wobj10"
10. Press "OK"



11. Press "OK"

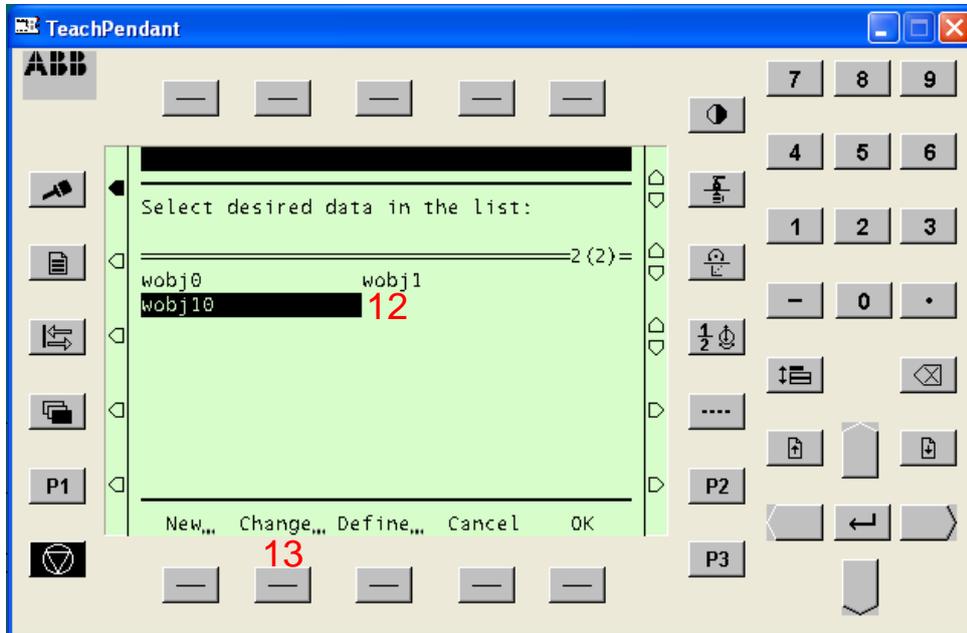


You have now created a new WorkObject: wobj10.

To change the coordinates of wobj10 to the coordinates of the origin of your workpiece, follow these steps:

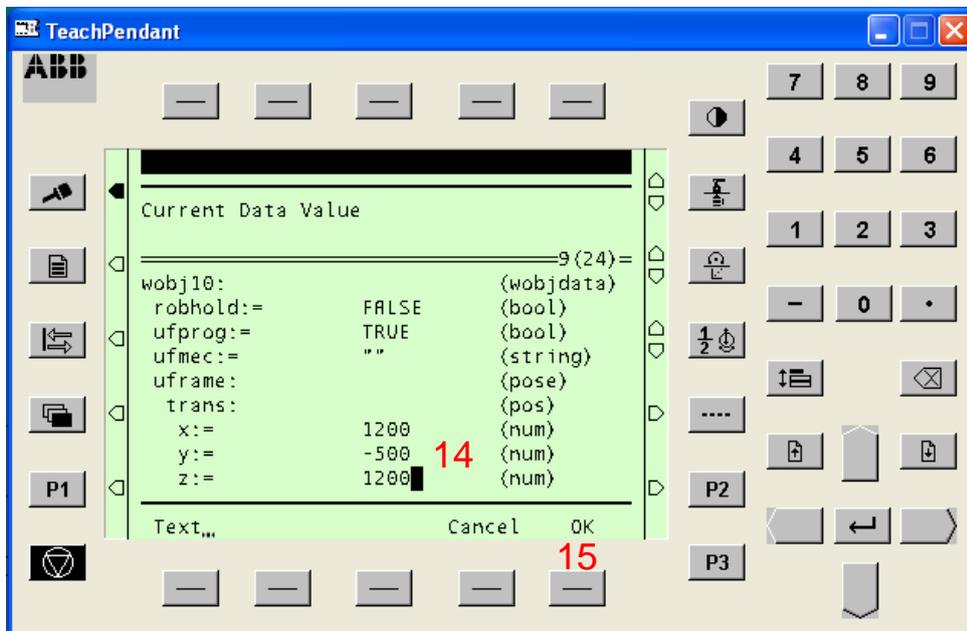
12. Stand on "wobj10"

13. Press "Change"

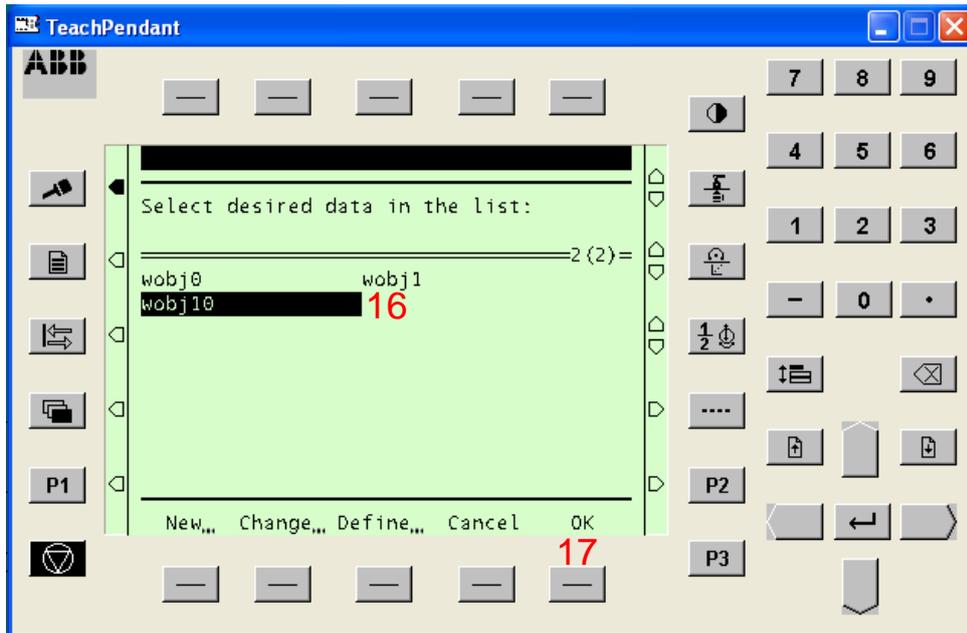


14. Go down to the X,Y,Z values and change them to your workpiece origin, including tool (use your own workpiece origin coordinates, coordinates shown in the picture below are for illustrative purposes only).

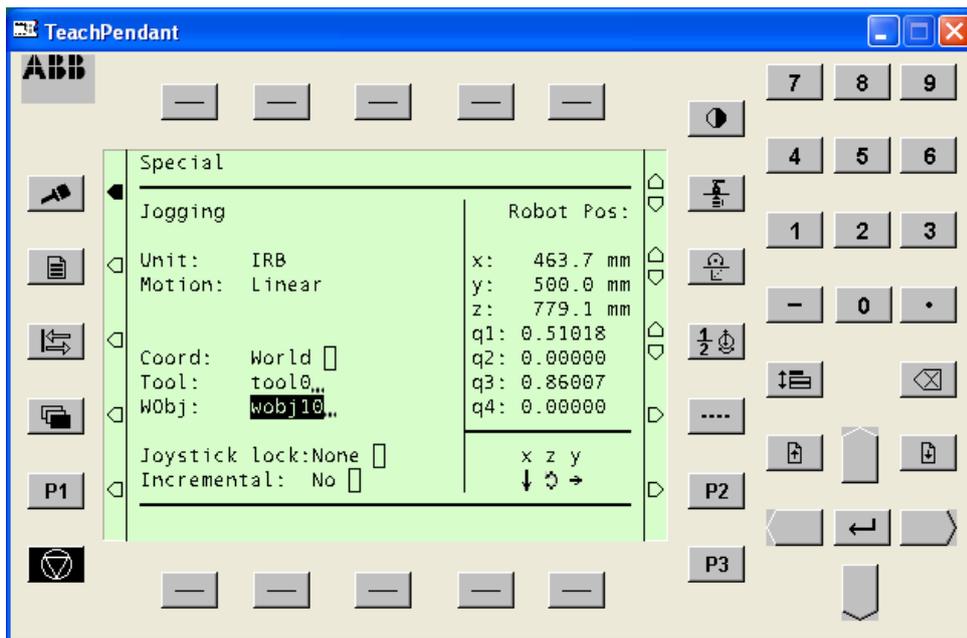
15. Press "OK"



16. Stand on wobj10
17. Press "OK"



Wobj10 has now been created and setup according to your workpiece origin



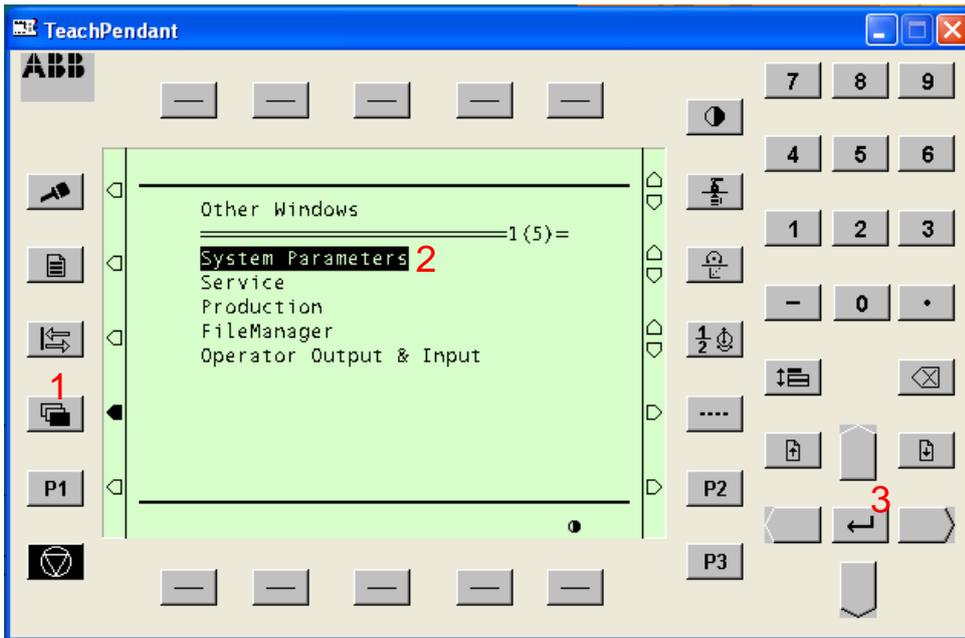
If you've setup wobj10 correctly, wobj10 is selected and your mill is at the origin point of your workpiece than X,Y,Z should indicate 0 or a value very close to 0.

2.2 Change the Baud-rate of Sio1

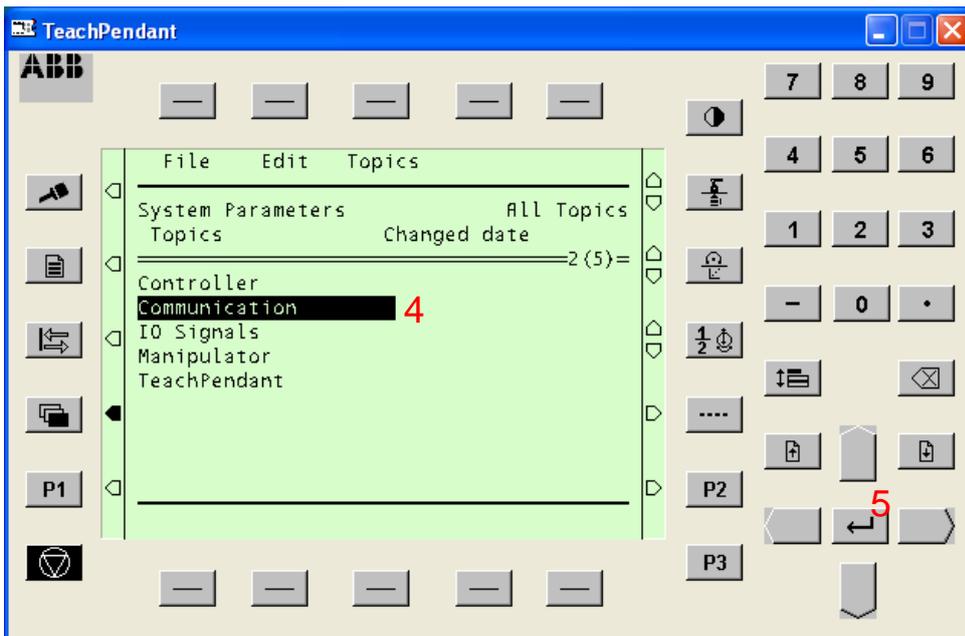
In order for the S4C controller and the Rapid2Serial program to communicate with each other they need to have the same baud-rate (speed of data transmission). To prevent the robot from pausing between data transmissions it's recommended to use the highest available baud-rate on the S4C controller, which is 19200. Changing the baud-rate will require a restart of the S4C controller.

Follow the steps below to change the baud-rate on the S4C controller to 19200.

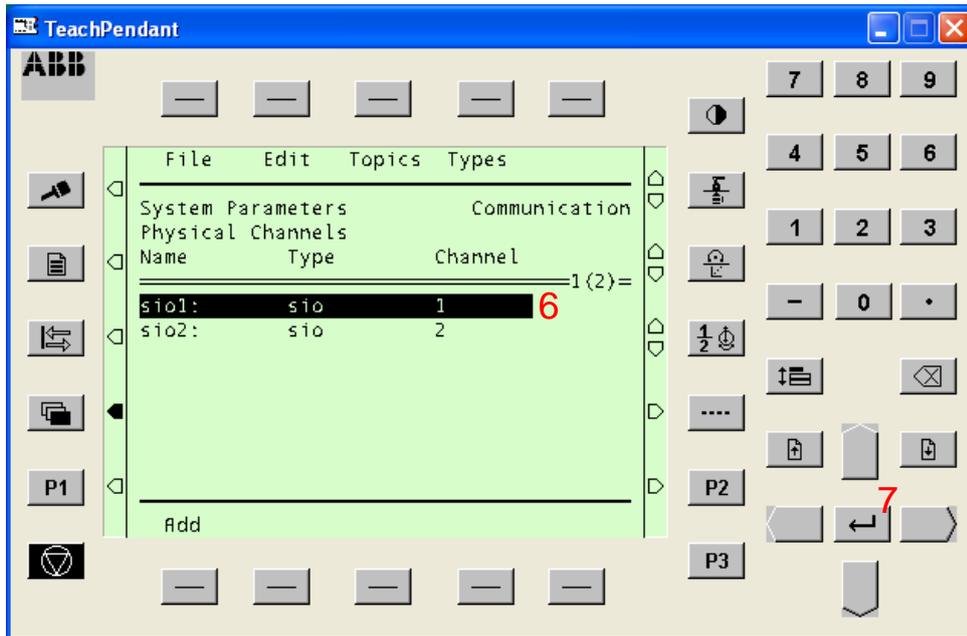
1. Go to "Other Windows"
2. Stand on "System Parameters"
3. Press enter



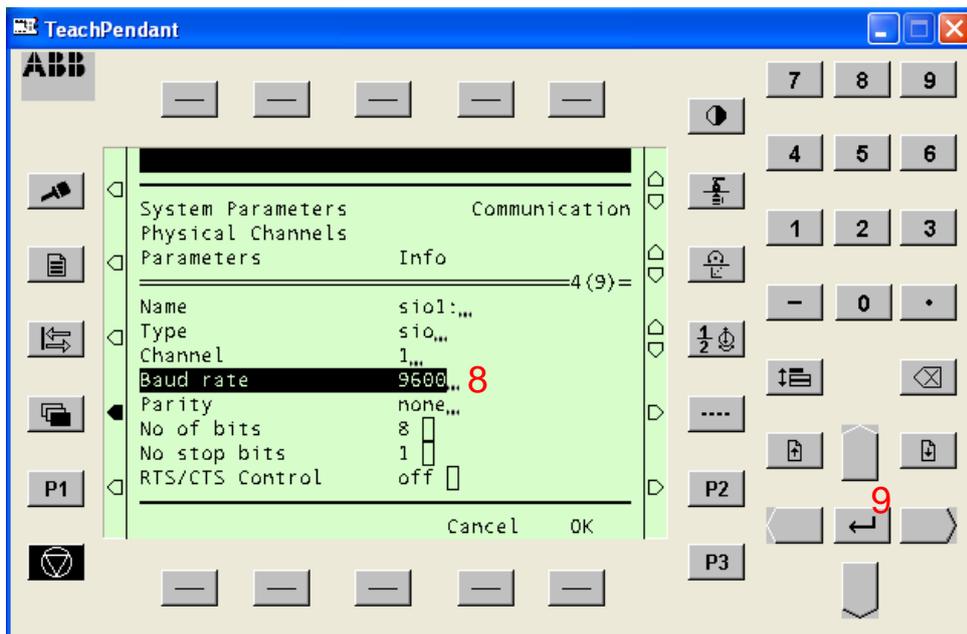
4. Stand on "Communication"
5. Press enter



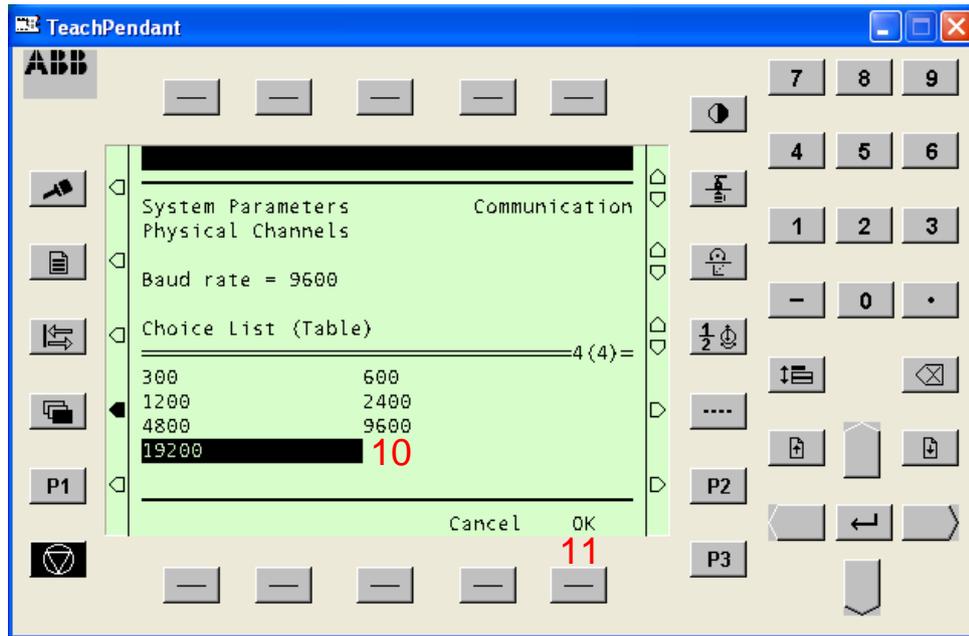
- Stand on "sio1"
- Press enter



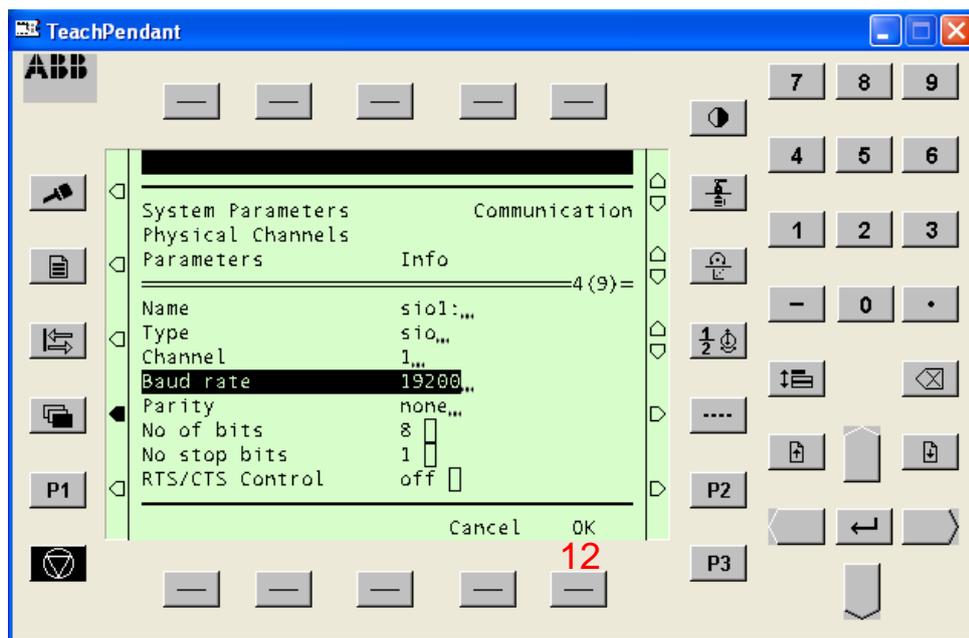
- Stand on "Baud rate"
- Press enter



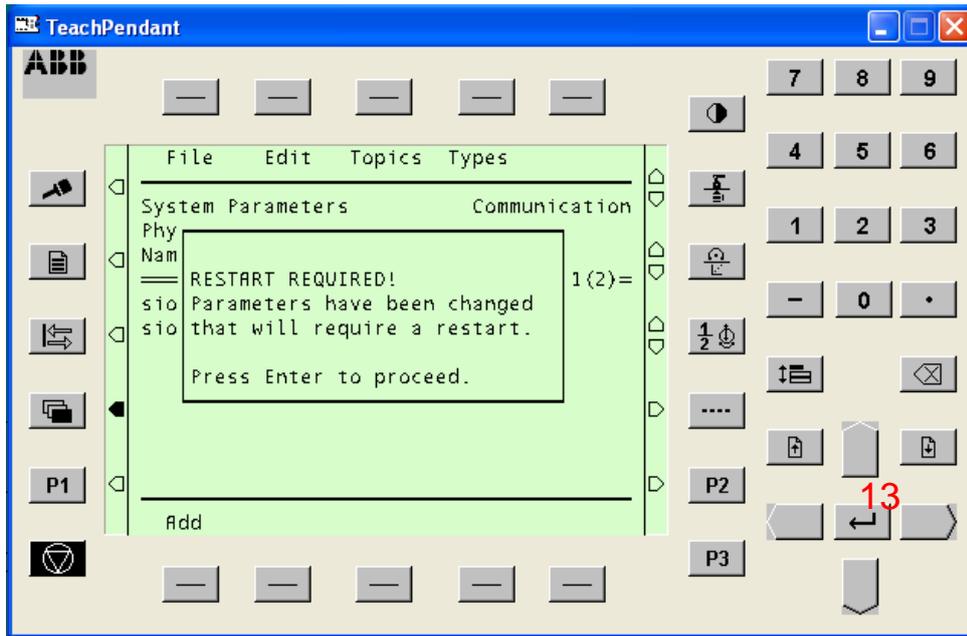
- 10. Stand on "19200"
- 11. Press "OK"



- 12. Press "OK"



13. Press enter
14. Restart the controller

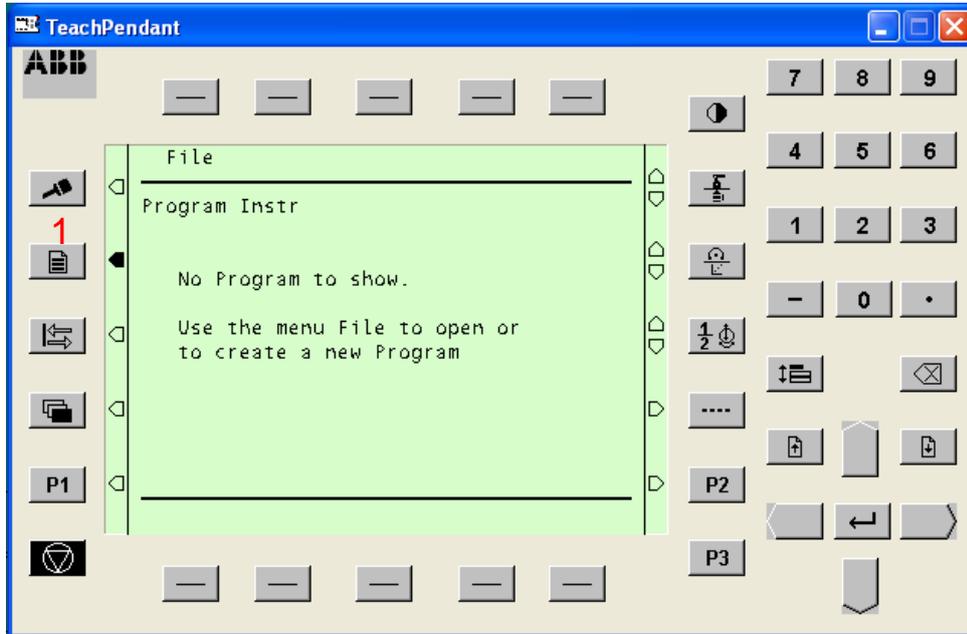


You have now successfully changed the baud-rate of Serial channel 1 on the S4C controller.

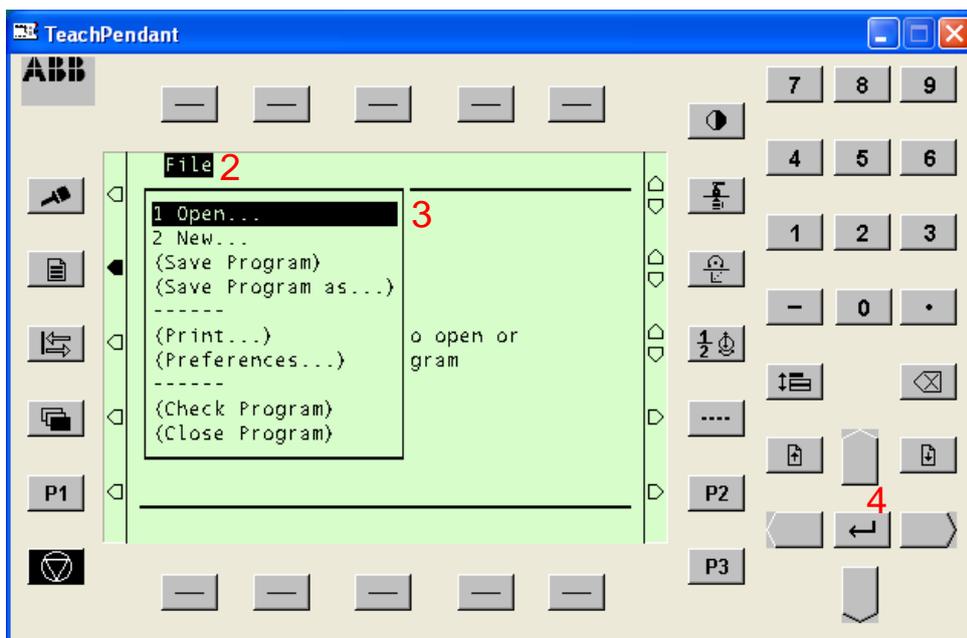
2.3 Load the FULTST09 program on the controller

You can follow the steps below to load the “FULTST09” rapid program on the S4C controller. It’s recommended to load “FULTST09” directly on the S4C controller from either a floppy drive or, if you’ve already done the Floppy USB emulator upgrade, from a USB drive. If you’ve followed the previous steps closely (creating and setting up wobj10 and setting the Baud rate) you shouldn’t encounter any errors when loading the FULTST09 program.

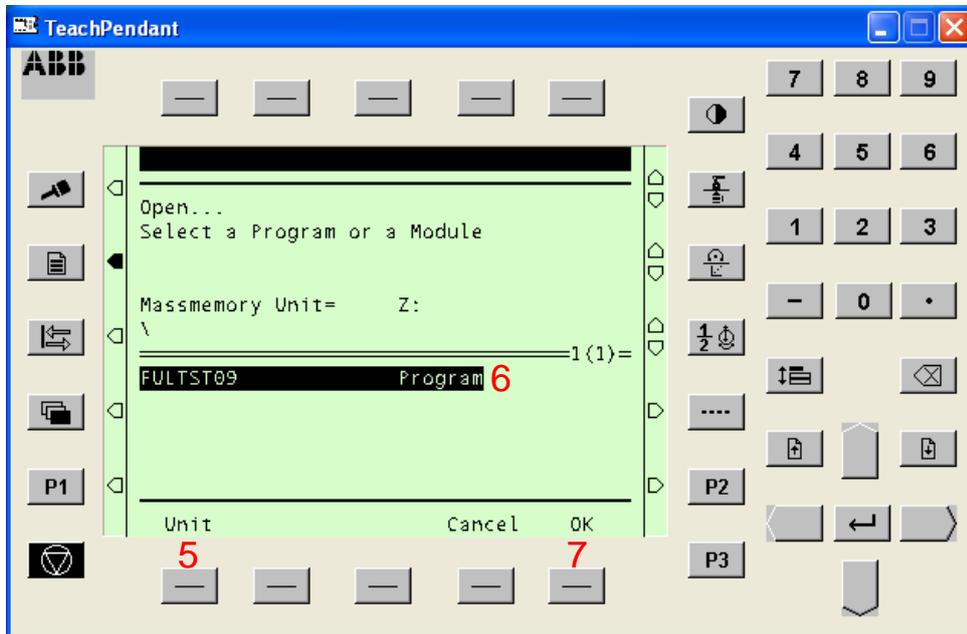
1. Go to the program window



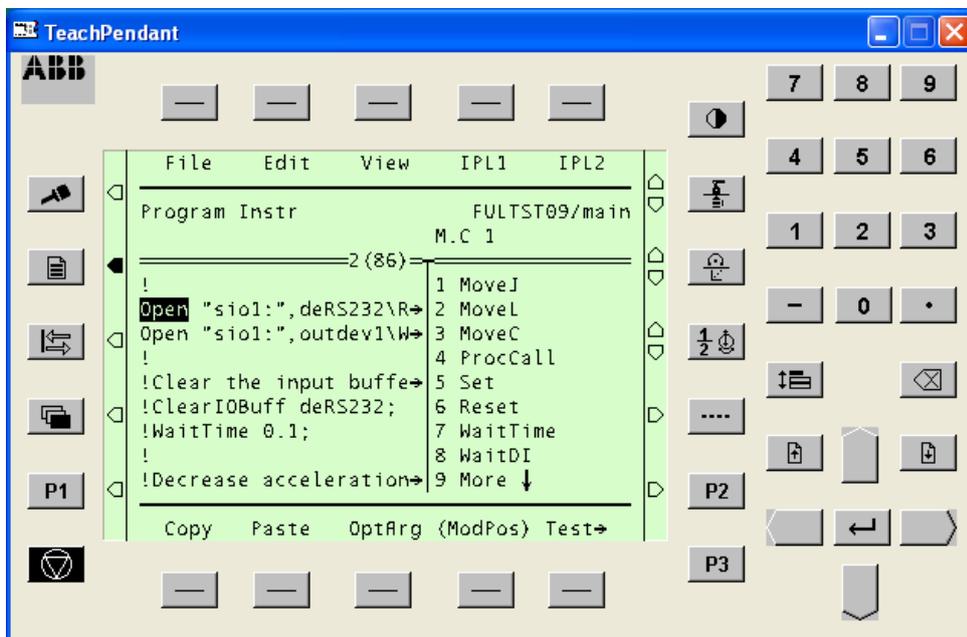
2. Press “File”
3. Stand on “Open”
4. Press enter



5. Press "Unit" to select the correct Massmemory Unit (this should be "flp1" if you're loading the program locally from a floppy or USB drive, picture below is from a simulation which is only able to open hard-drive locations on the PC).
6. Stand on "FULST09"
7. Press "OK"



The rapid program "FULST09" has now been loaded onto the S4C controller.



2.3.1 Check rotation settings

Depending on the milling configuration of your robot you might need to change the rotation settings in "FULST09" (see picture below). If your axis 5 is at 90 degrees use "0,0,1,0". If your axis 5 is at 0 degrees use "0.7,0,0.7,0". If you use another configuration you can look at a rapid program you know works with your robot configuration and copy and paste the rotation settings into "FULST09".

```
!TPWrite "Zval: "\Num:=Zval;
!TPWrite "Sumval: "\Num:=Sumval;
!TPWrite "XYZsumval: "\Num:=XYZsumval;
!TPWrite "Sumdifval: "\Num:=Sumdifval;
!
!if the checksum found in the line string is the same as the calculated sum of X,Y,Z val then the data is good.
IF Sumdifval<1 THEN
  Write outdev1,"OK";
  MoveL [[Xval,Yval,Zval] [0,0,1,0] [0,0,0,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009]],v40,z1,tool10\WObj:=wobj10;
ELSE
  TPWrite "Error: Checksum difference detected";
  TPWrite "Checksum difference: "\Num:=Sumdifval;
  Finished:=TRUE;
  Write outdev1,"NK";
ENDIF
ELSE
  TPWrite "Identifier not found, bad data";
```

Rotation settings

Figure 3: Change rotation settings in "FULST09"

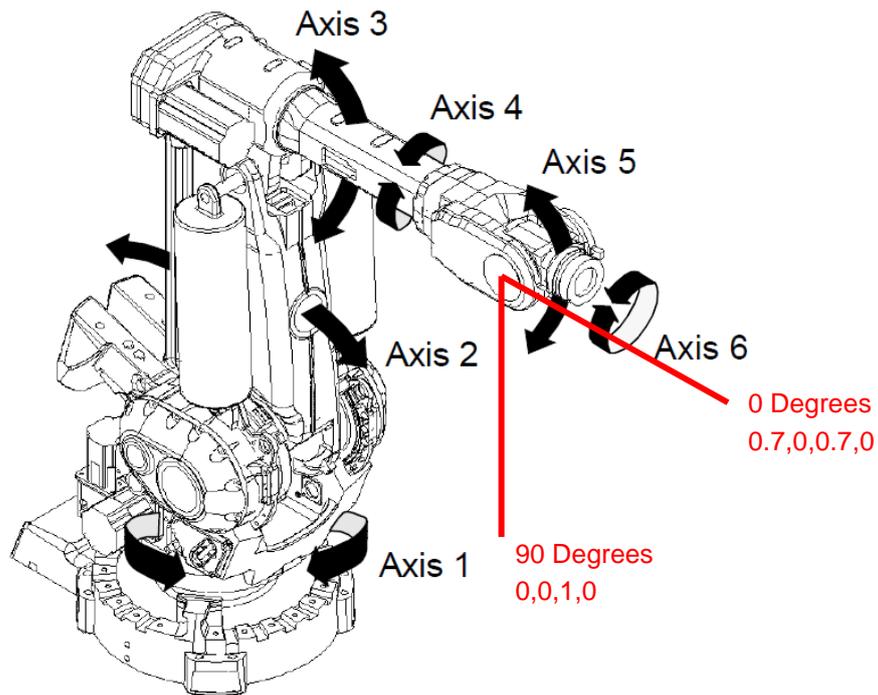


Figure 4: Rotation of axis 5

3 Setting up the PC and Rapid2Serial program

3.1 Connection between PC and S4C controller

The Rapid2Serial program uses serial communication to send the coordinates to the controller. Therefore a RS232 to USB cable is needed.



Figure 5: RS232 to USB cable

The USB connection goes to any port on the PC (used COM-port can be later selected in the Rapid2Serial program). The RS232 connection goes to the “SIO1” port on the S4C controller. The “SIO1” port can be found on top of the S4C controller.

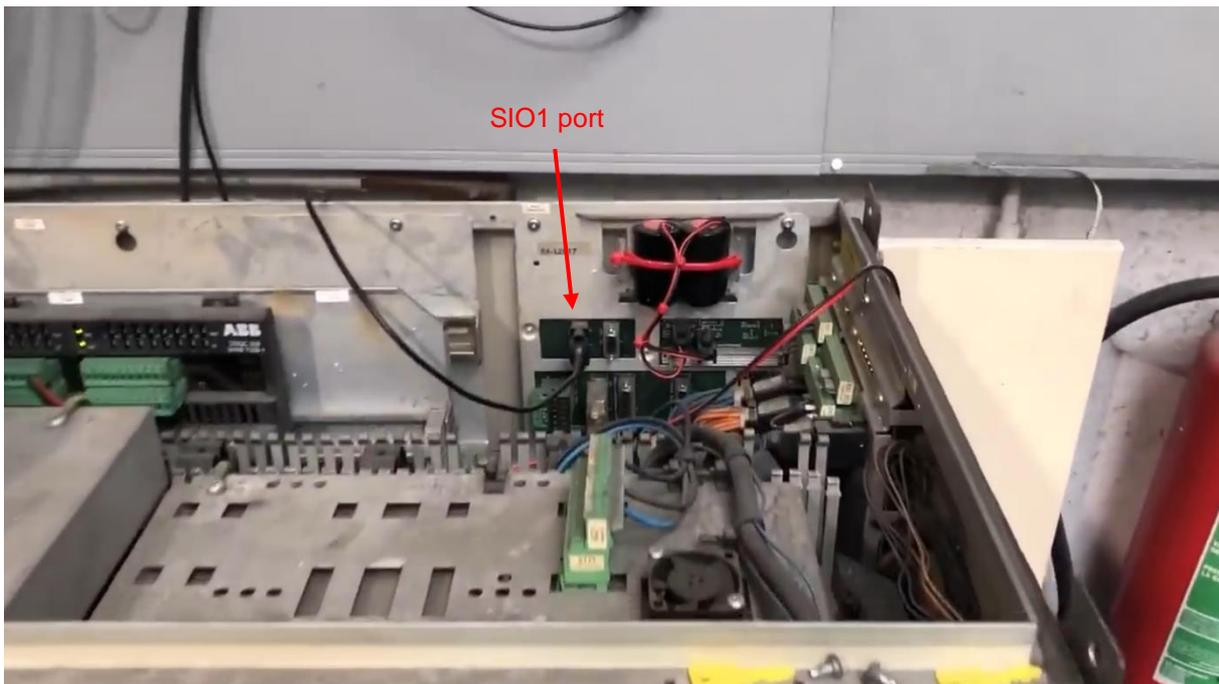


Figure 6: "SIO1" port location on the S4C controller

3.2 Setting up the Rapid2Serial program

When you start the Rapid2Serial exe (run as administrator) you should get the following screen:

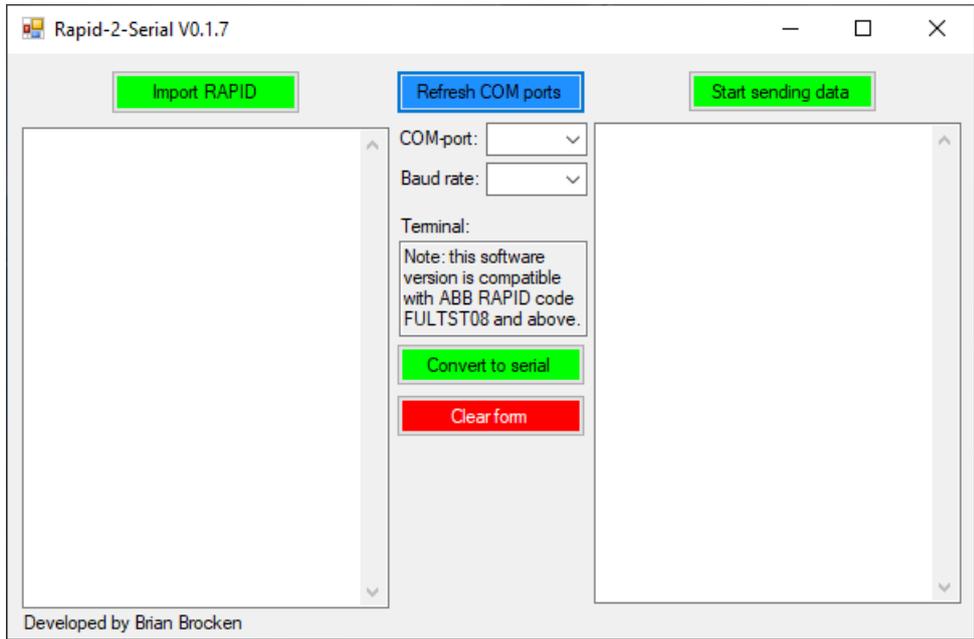


Figure 7: Rapid2Serial program

3.2.1 Importing a rapid file

If you click on “Import RAPID” you’re able to select the rapid file you would like to send to the robot. **.prg** and **.mod** are valid files. It’s recommended to use rapid files that are generated by Fusion 360. Note that depending on the file size it can take up to a few seconds for the file to be imported.

3.2.2 Rapid file criteria

If your file is not generated by Fusion 360 then make sure the file meets the following criteria:

- Make sure the file only contains “MoveL” or “MoveJ” instructions and avoid using instructions like “MoveAbsJ”. The reason for this is that Rapid2Serial searches for 3 coordinates in every line (X,Y,Z) and instructions like “MoveAbsJ” contain 6 absolute positioning values for all the 6 axis.
- If your file contains tool-coordinates you should delete these as the Rapid2Serial program will send these to the robot as positional coordinates (robot targets).

Included with the Rapid2Serial program is a file called “mAdaptive2.mod” which is generated by Fusion 360 and is known to work with the Rapid2Serial program. You can use this as a test file.

3.2.3 Converting to a serial format

If you’ve imported a valid Rapid file you can now convert this file to a serial format with the button “Convert to serial”. The purpose of this function is to find and extract the 3 coordinates (X,Y,Z) from every line, add an identifier in front of the serial line and add a checksum of the X,Y and Z coordinates in the back. The checksum protects the robot from going to random places when data gets corrupted during transmission. Sending only the X,Y,Z values (+ identifier and checksum) allows for a faster data transmission compared to sending the complete MoveL or MoveJ instructions (less data needs to be send).

Once you've converted the Rapid file to a serial format it's important to check if the Serial data is correctly formatted. The format should be as follows: "1,Xval,Yval,Zval,ChecksumXYZ". Pay close attention to the checksum, points and commas. A decimal number should have a point and the values should be separated by a comma.

The lines below are formatted correctly

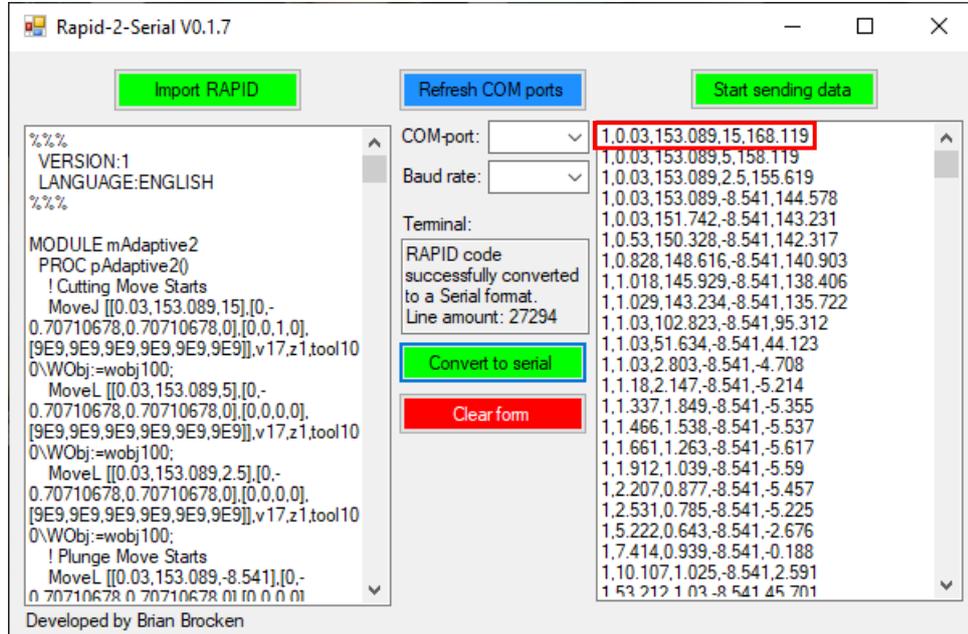


Figure 8: Serial data formatted correctly

The lines below are not formatted correctly.

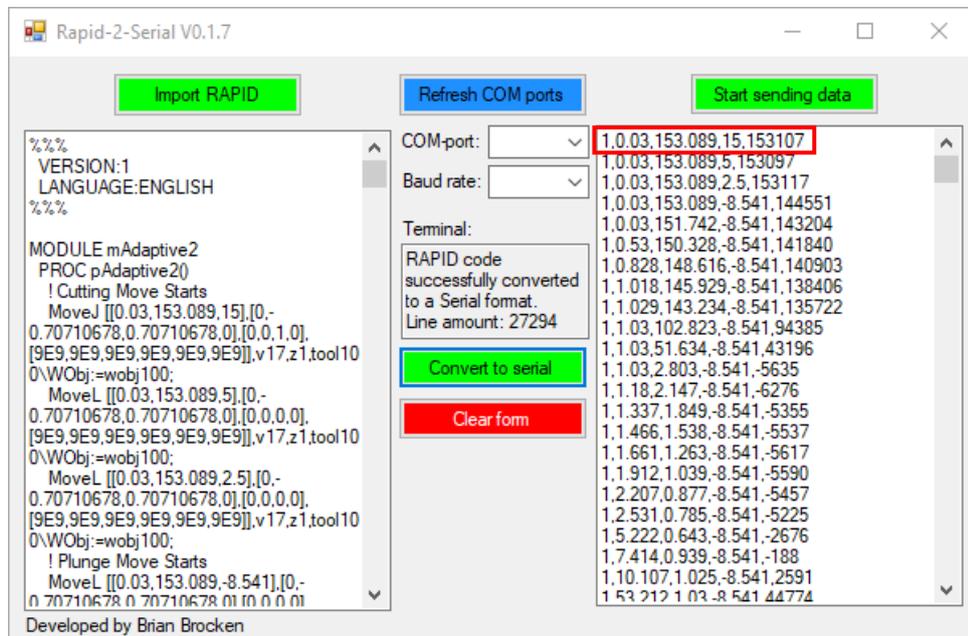


Figure 9: Serial data not formatted correctly

If you have comma's or points missing or your checksum doesn't equal to the sum of the X,Y,Z values then it's possible your regional settings in windows are not setup correctly. You can check these settings as follows:

go to **Start** >type **Control Panel** and press enter > **Region**

Click **Additional Settings**

Make sure that for a **Decimal Symbol** a **dot/point** is selected. And for a **List Separator** a **comma** is selected. See pictures below

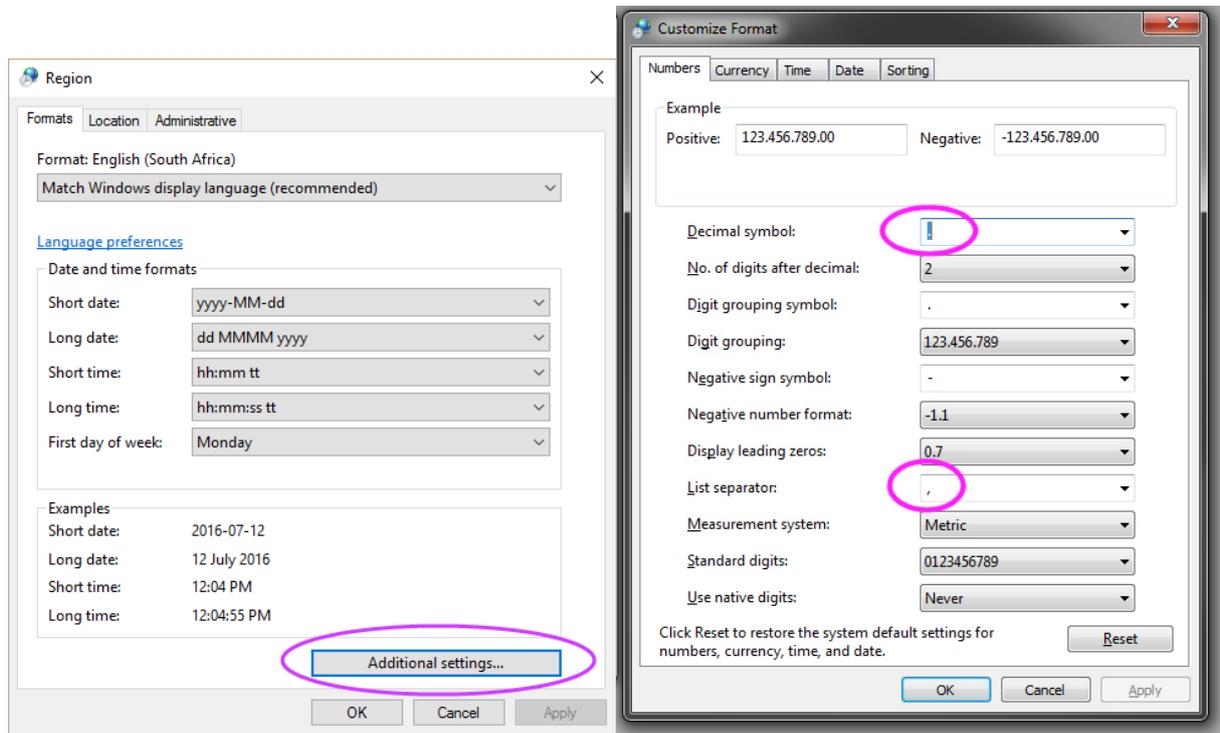


Figure 10: Region settings

3.2.4 Selecting COM-port and Baud-rate

If you've imported the Rapid file and converted it into a serial format you can almost start sending the data to the robot but first you have to select the right COM-port and Baud-Rate.

The Rapid2Serial programs checks which devices are connected to the PC upon startup. If you've connected the controller to the PC after you started the Rapid2Serial program you can refresh the COM-ports with the button "Refresh COM ports". Be sure to select the right COM-port if you have multiple devices connected to your PC.

After you've selected the right COM-port you can select the right Baud-rate. This should be the same Baud-rate you've setup in the S4C controller (see **2.2 Change the Baud-rate of Sio1**). It's recommended to use the highest Baud-rate available on the S4C controller (19200).

4 Start sending data

It's important to make sure the area around the robot is clear of people or objects before you start sending data. It's recommended to test out the Rapid2Serial program in manual mode at a reduced speed.

Follow these steps to start sending data to the robot:

1. Reduce speed to **75%** on the S4C controller to prevent getting the error: "50082: Deceleration limit Calculation of joint deceleration time exceeds internal limits for this motion."
2. Start the program "FULTST09" first on the S4C controller
3. Within **30 seconds** of starting the program on the S4C controller click on "Start sending data" on the Rapid2Serial program, the robot will now start to execute the selected program, line by line.

The "Lines send" value in the terminal box should now be starting to count up and the robot should be executing the coordinates send by the Rapid2Serial program.

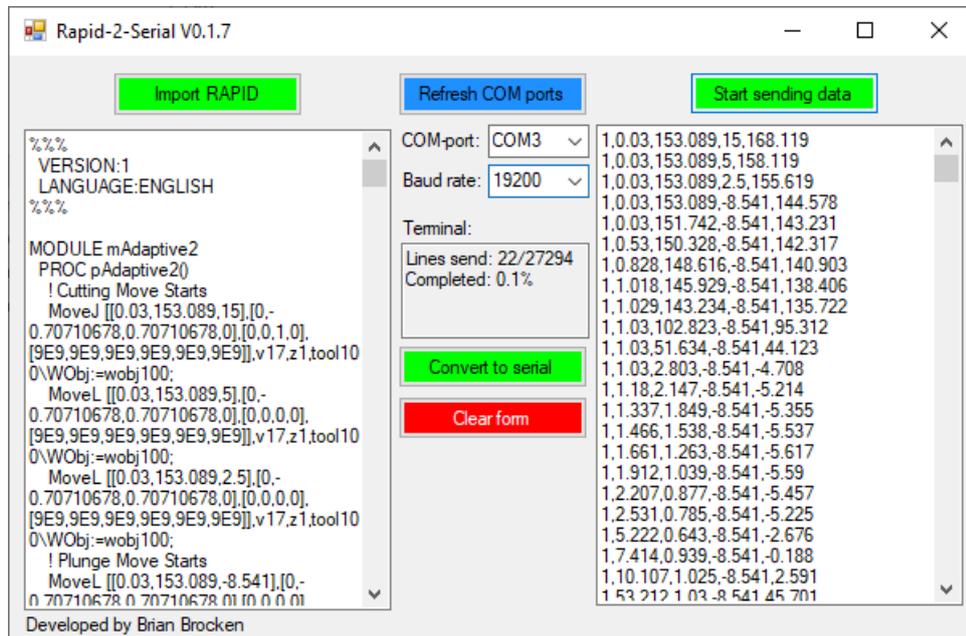


Figure 11: Rapid2Serial program sending data

5 Troubleshooting guide

This guide can be used to solve the most common problems like communication problems, timeout exceptions and other errors.

Problem	Possible causes	Possible solutions
<p>-Timeout exception on the Rapid2Serial program</p>	<p>-There's no communication between PC and S4C controller</p> <p>-You didn't start sending the data on the Rapid2Serial program within 30 seconds of starting the "FULST09" rapid program on the S4C controller.</p>	<p>-Make sure "FULST09" is loaded and running on the S4C controller (see 2.3 Load the FULST09 program on the controller).</p> <p>-Make sure the S4C controller and Rapid2Serial program have the same Baud-rate (see 2.2 Change the Baud-rate of Sio1 and 3.2.4 Selecting COM-port and Baud-rate).</p> <p>-Make sure you've selected the right COM-port (see 3.2.4 Selecting COM-port and Baud-rate).</p> <p>-Make sure you connected the RS232-to-USB cable to "SIO1" on the S4C controller (see 3.1 Connection between PC and S4C controller).</p> <p>-Make sure you start sending the data on the Rapid2Serial program within 30 seconds of starting the "FULST09" rapid program on the S4C controller.</p>
<p>-Not being able to select a COM-port in the Rapid2Serial program</p>	<p>-You don't have full administrator rights on the PC</p> <p>-Your Firewall is blocking the Rapid2Serial program from accessing your PC's COM-ports</p>	<p>-Run the Rapid2Serial program as administrator</p> <p>-Add the Rapid2Serial program to trusted programs in your firewall. Go to Settings > Security > Windows Security > Open Windows-Security > Firewall and Network protection > Allow an app through firewall > Change settings > Allow another app > Click OK</p>
<p>-Getting the message "Failed to send line xx, would you like to retry or cancel sending?" on the Rapid2Serial program</p>	<p>-There is communication between PC and S4C controller but the controller does not accept the data being send by the Rapid2Serial program.</p>	<p>-Make sure the serial lines in the right textbox are formatted correctly. Pay close attention to the checksum, points and comma's. If this is not correct you might have to change your region settings in windows (see 3.2.3 Converting to a serial format).</p>

<p>-Getting the message “Please import a valid RAPID file first” on the Rapid2Serial program when trying to convert to a serial format.</p>	<p>-Your Rapid file contains functions that are not supported by Rapid2Serial</p>	<p>-The Rapid2Serial program is only able to send X,Y,Z coordinates to the S4C controller. MoveL and MoveJ functions are accepted. Other functions like MoveAbsJ or IO control are not accepted (see 3.2.2 Rapid file criteria).</p>
<p>-Getting error “40165: Reference error” on the S4C controller.</p>	<p>-WorkObject: wobj10 cannot be found in the controller</p>	<p>-Make sure you’ve created a wobj10 in the controller (see 2.1 Creating a new WorkObject: wobj10).</p>
<p>-Getting error “50143: Robot axes config. error” on the S4C controller</p>	<p>- Rotation settings in “FULTST09” are different from your used robot configuration.</p>	<p>-Make sure the rotation settings in “FULTST09” are setup correctly (see 2.3.1 Check rotation settings).</p>
<p>-Getting error “50082: Deceleration limit” on the S4C controller.</p>	<p>-Calculation of joint deceleration time exceeds internal limits for this motion. Robot targets are too close to each other</p>	<p>-Increase the PathResol value in “FULTST09” from 175 to 200.</p> <p>-Avoid robot targets (points) lying very close to each other like for example small detailed circles. Try to reduce the resolution of the generated path (less robot targets). In fusion 360 change the “tolerance” value from 0.1 to 1mm, enable smoothing with a value of about 0.6mm.</p> <p>-Lower the speed of the robot (choose a speed reduction value lower than 75%)</p>

6 List of figures

Figure 1: Fusion 360 origin point.....	3
Figure 2: ABB IRB6400 coordinate systems	3
Figure 3: Change rotation settings in "FULTST09"	15
Figure 4: Rotation of axis 5.....	15
Figure 5: RS232 to USB cable	16
Figure 6: "SIO1" port location on the S4C controller	16
Figure 7: Rapid2Serial program	17
Figure 8: Serial data formatted correctly	18
Figure 9: Serial data not formatted correctly	18
Figure 10: Region settings.....	19
Figure 11: Rapid2Serial program sending data.....	20