

# SABT Programming Instructions

## Table of Contents

Overview	2
Main control module connections	2
User interface module connections	3
Programming connections	3
Programming the main control module	4
Programming the user interface modules	5
Using AVR Studio 4	6
Programmer settings	9
Programming the microcontroller	11
Setting up PuTTY	12
Using PuTTY	15

## Overview

The setup being used to program the SABL is the JTAG ICE MkII debugger from ATMEL (<http://www.atmel.com/tools/AVRJTAGICEMKII.aspx>). At the time of creation of this document, the most recent version of AVR Studio did not work with this setup so AVR Studio 4 is being used (<http://www.atmel.com/tools/AVRSTUDIO4.aspx>).

The SABL includes four different boards:

1. Main control module
2. Primary user interface module
3. Intermediate user interface module
4. Advanced user interface module

The main control module uses an ATmega1284P microcontroller and all the user interface modules use an ATmega168PA microcontroller.

**IMPORTANT** - Before making or unmaking any connections to the boards, please make sure there everything is turned off or otherwise there is no power being supplied. The boards are easy to short circuit.

## Main control module connections

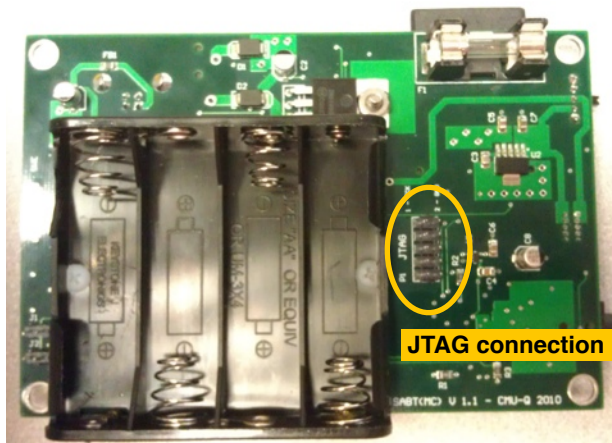


Figure 1 - Top of main control module.

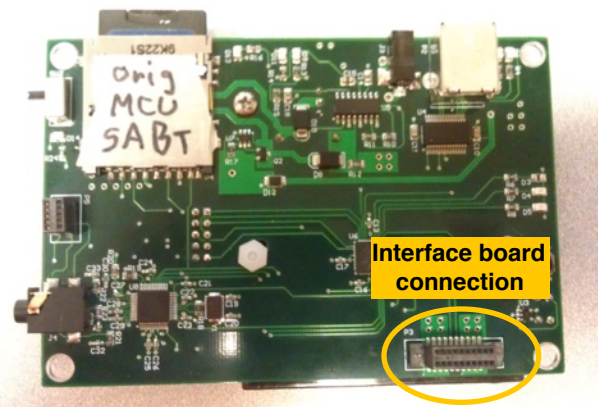


Figure 2 - Bottom of main control module.

The main control module powers itself and the connected user interface module through either 4 AA batteries or USB connection. If the batteries are used, the fuse does not need to be installed.

## User interface module connections

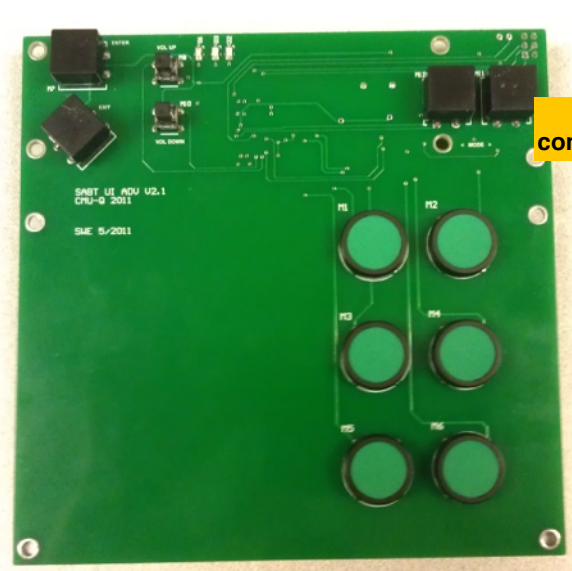


Figure 3 - Top of primary user interface module.

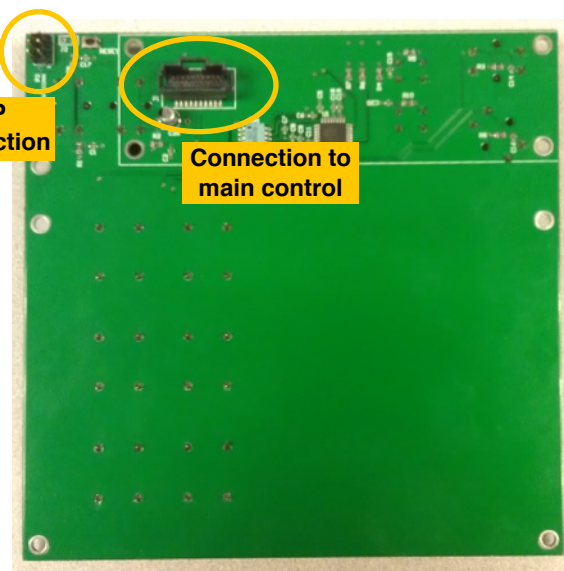


Figure 4 - Bottom of primary user interface module.

## Programming connections



Figure 5 - JTAG ICE MkII debugger with JTAG connector(top) and ISP adapter(bottom)

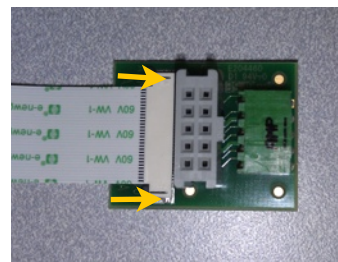
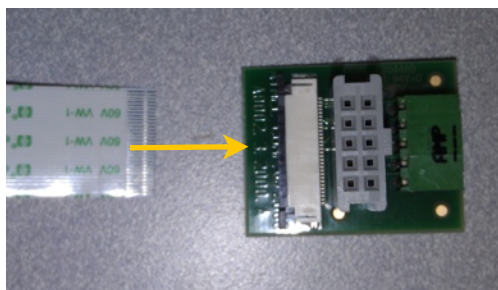


Figure 6 - Connecting the JTAG connector. Slide the ribbon cable in then push in the black clamps on both sides of the connection.



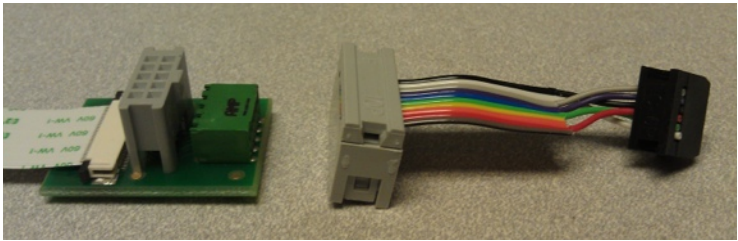
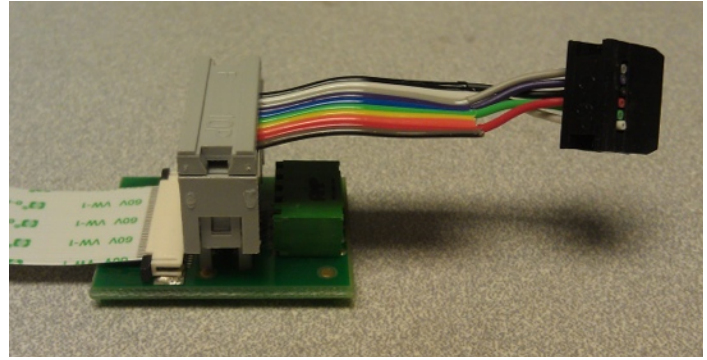


Figure 6 - Connecting the ISP adaptor.



## **Programming the main control module**

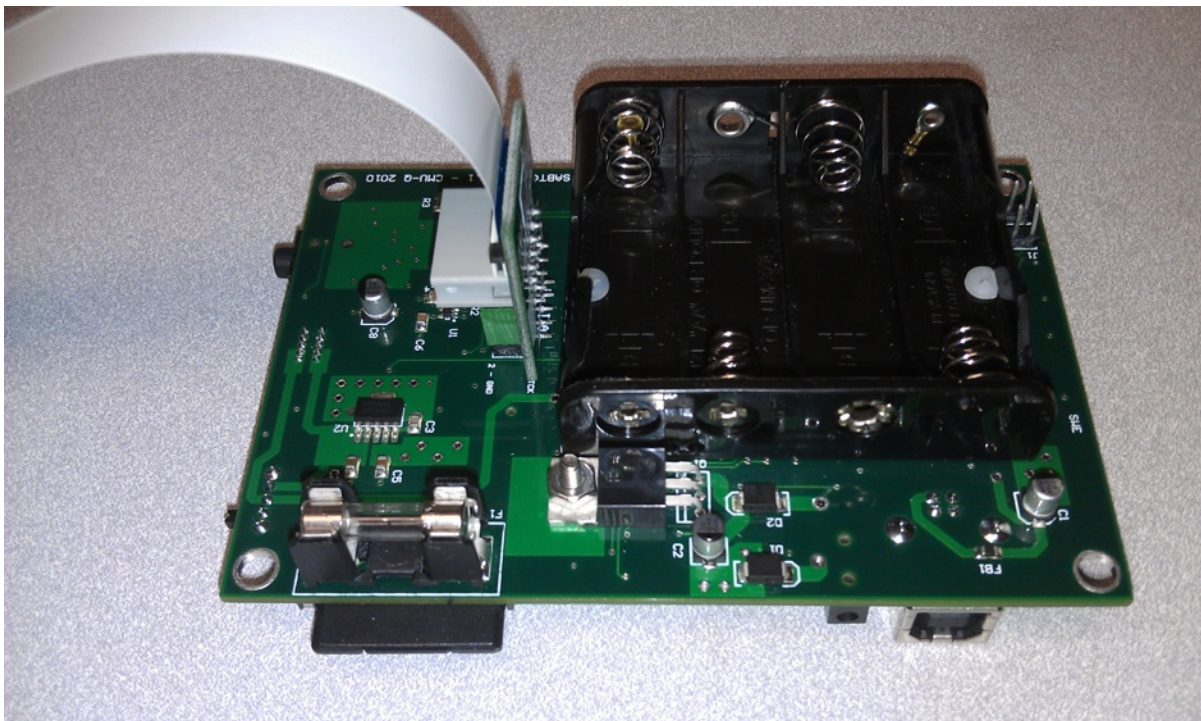


Figure 7 - Connecting the JTAG interface to program the main control module.

The orientation of the JTAG connection matters so be sure to double check it. If the wrong orientation is used, it may be necessary to power cycle the programmer in order for the computer to recognize its presence. Plug in the JTAG connector, plug in the USB cable to the computer, turn on the main control module, then follow the procedure below to use AVR Studio 4 to upload code.

## **Programming the user interface modules**

Only the primary user interface module is pictured below, but all of the user interface modules use the same connection to program. First, make sure power is off to the main control module, then connect the boards as shown below (Figure 8). Then connect the ISP adaptor as shown above (Figure 6). Turn on power to the main control module and follow the AVR Studio procedure below to upload code.

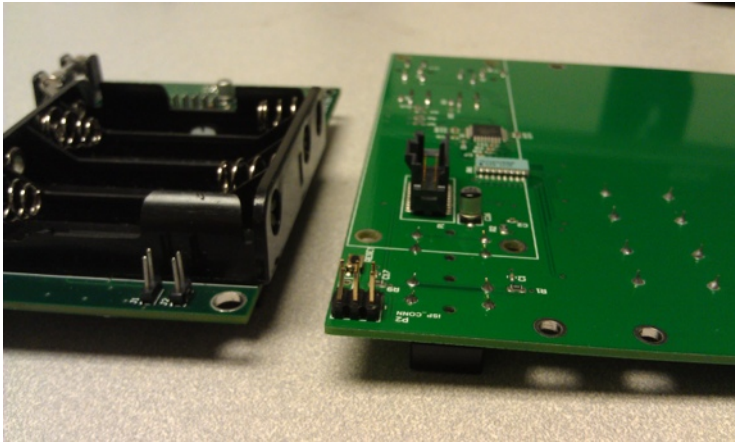


Figure 8 - Connecting the main control module to the primary user interface module.

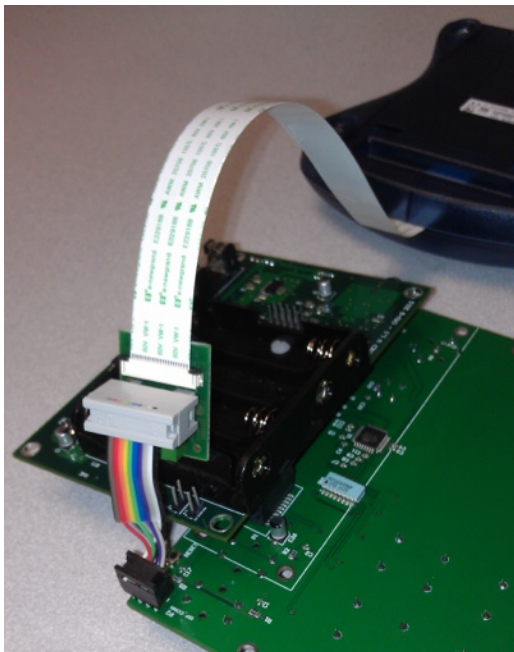
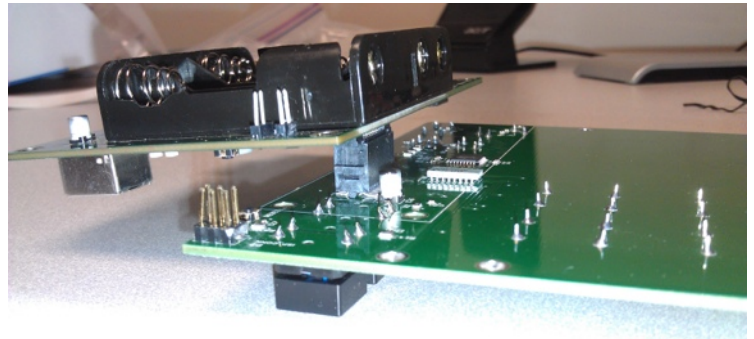


Figure 9 - Connecting the ISP adaptor to the primary board.

The orientation of the ISP adaptor does matter so be sure to double check it. If the wrong orientation is used, it may be required to power cycle the programmer in order for the computer to recognize its presence. Note the JTAG ICE MkII is upside down. This is done to reduce the strain on the ribbon cable connector as problems were had with them being fragile.

## Using AVR Studio 4

AVR Studio only works on windows platforms. Open AVR Studio and a prompt should open with recent projects (Figure 10). Select the project here or browse to the containing folder and open the .aps in order to open the project. If using the github repository, the .aps files are located in:

SABTSoftware/SABT\_MainUnit/SABT\_MainUnit.aps

SABTSoftware/UserInterface(Primary)/SABT\_Primary.aps

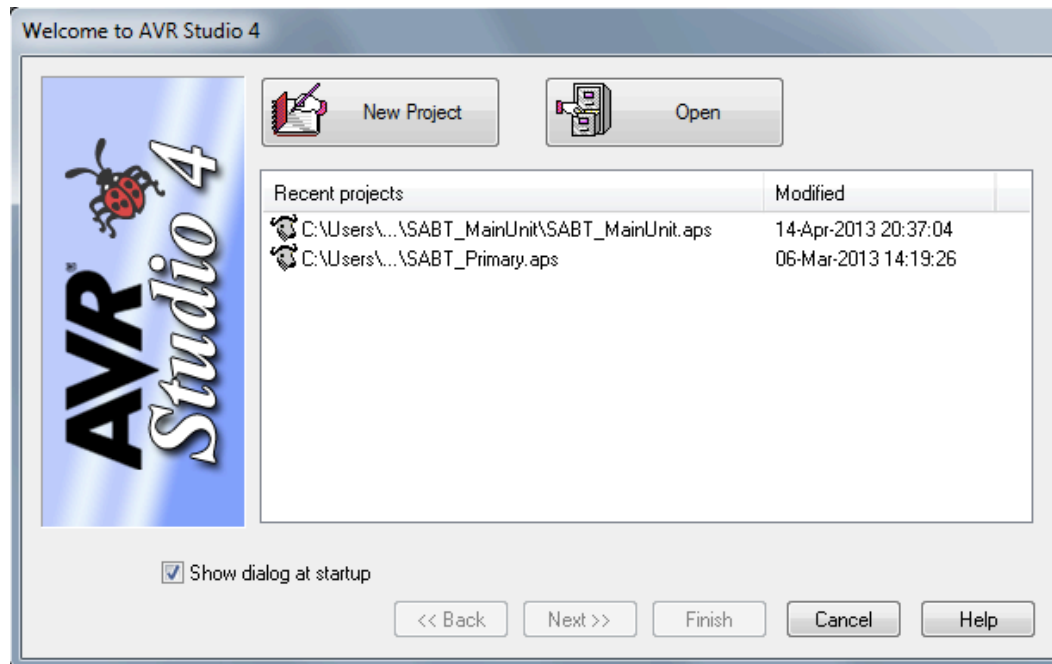


Figure 10 - Welcome screen, choose project.

First double check all of the configuration options (Figure 11). The settings shown below are known to be important, but if a particularly nasty, unexplainable bug occurs, it may be prudent to double check ALL of the other configuration options as well if the cause cannot be determined in the software. Do (Figure 12) for both the main control module and the user interface modules. Do (Figure 13) only for the user interface modules.

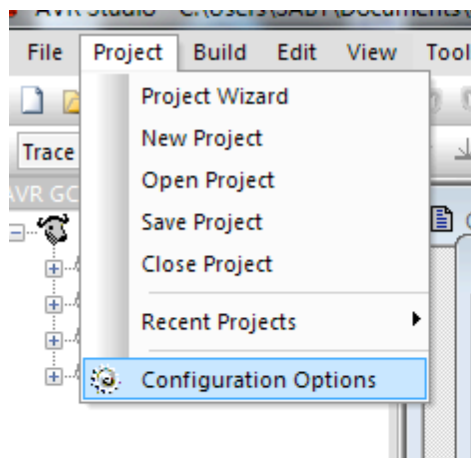


Figure 11 - Configuration options





After making changes to the code - rebuild the project and connect to the microcontroller as shown below.

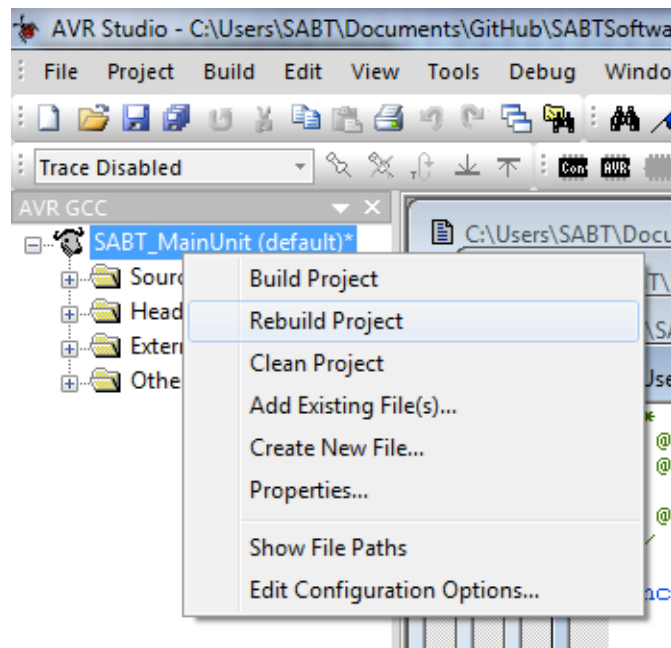


Figure 14 - Right click on the top level directory.

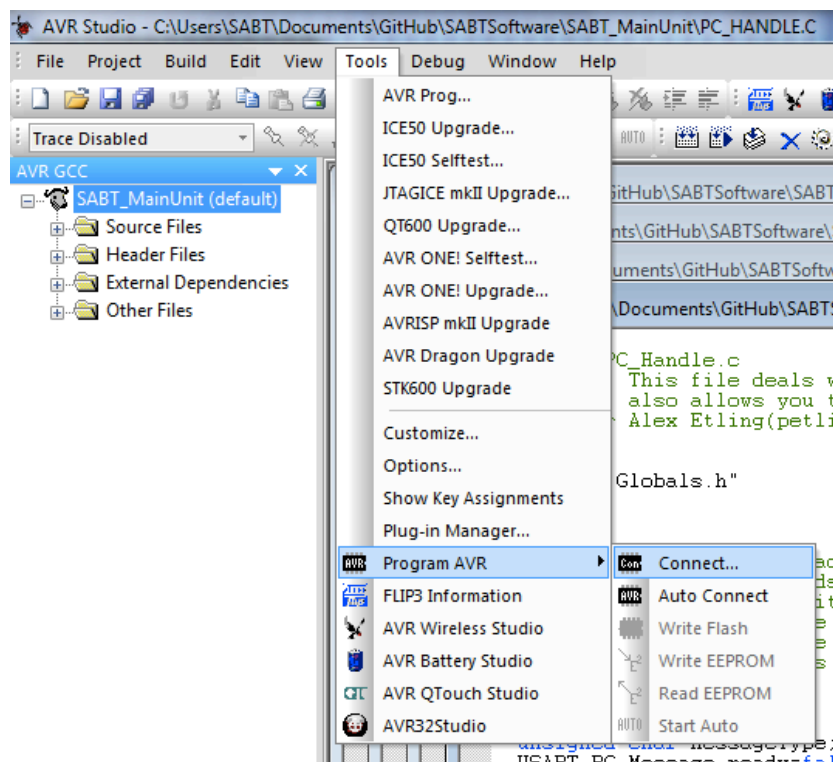


Figure 16 - Connect to the microcontroller



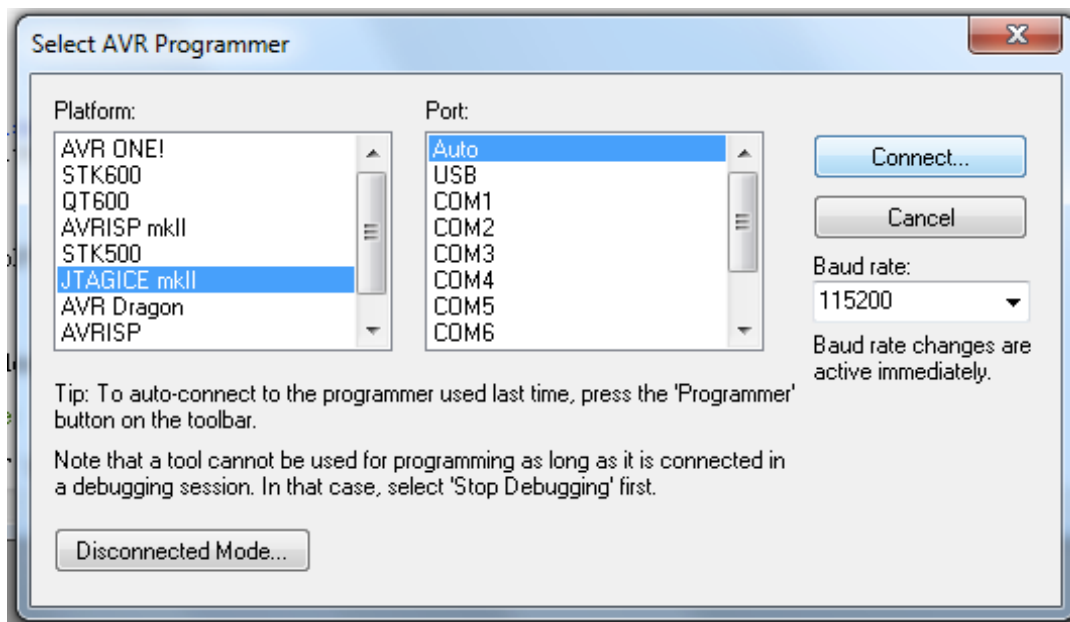


Figure 17 - Select programmer

**NOTE:** If/when connection problems arise, the most effective way to fix it is to turn off the main control module and programmer, unplug their USB cables and replug them. If this does not work, double check the orientation of the programming cable.

## Programmer settings

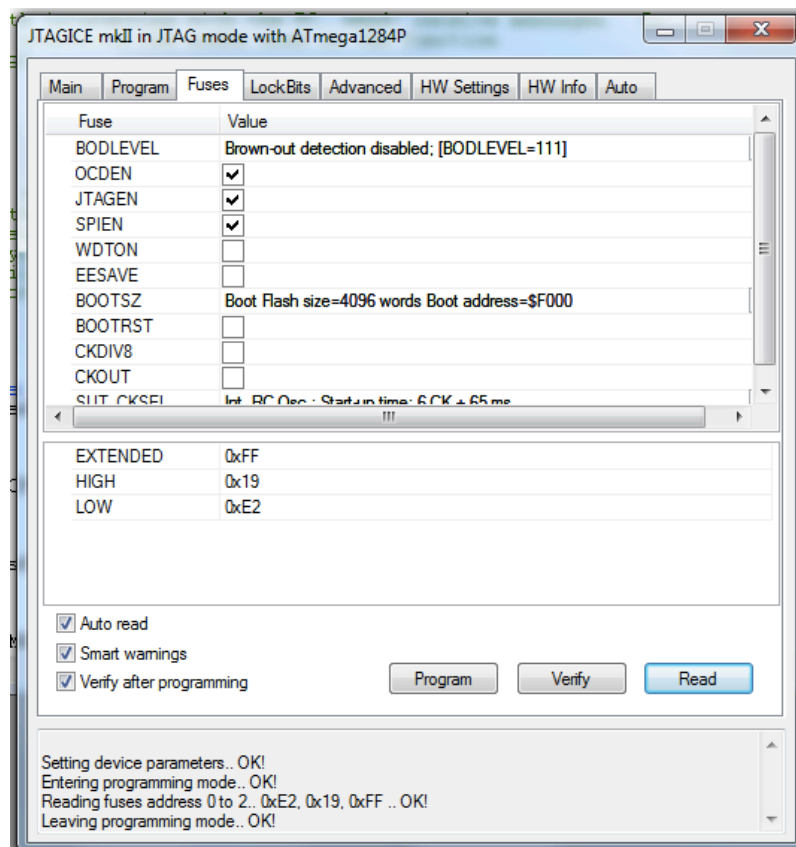


Figure 18 - Fuse settings

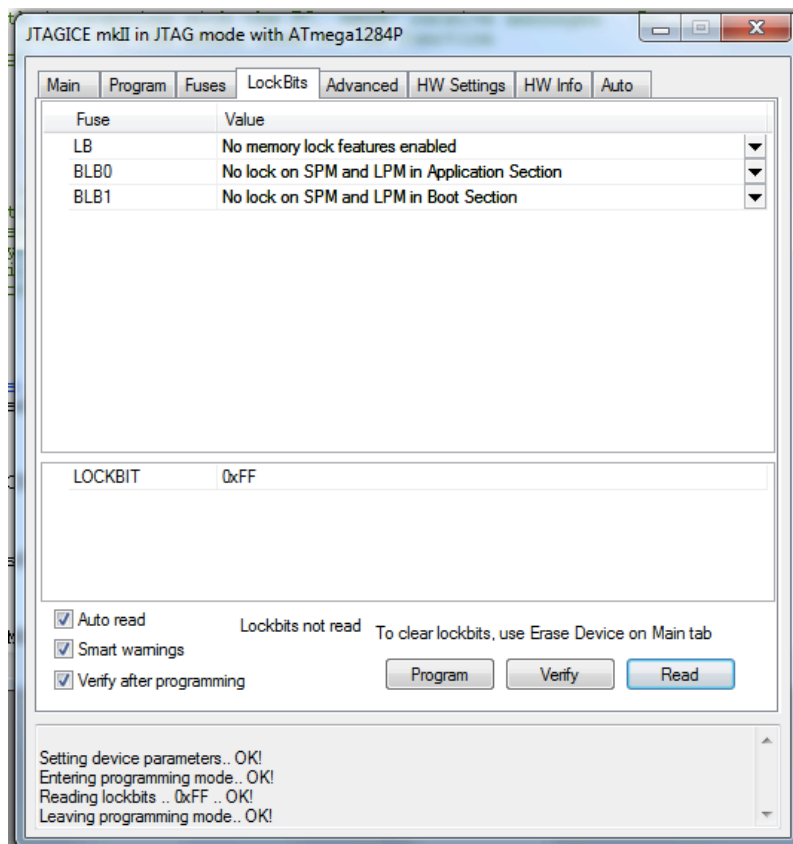


Figure 19 - Advanced settings

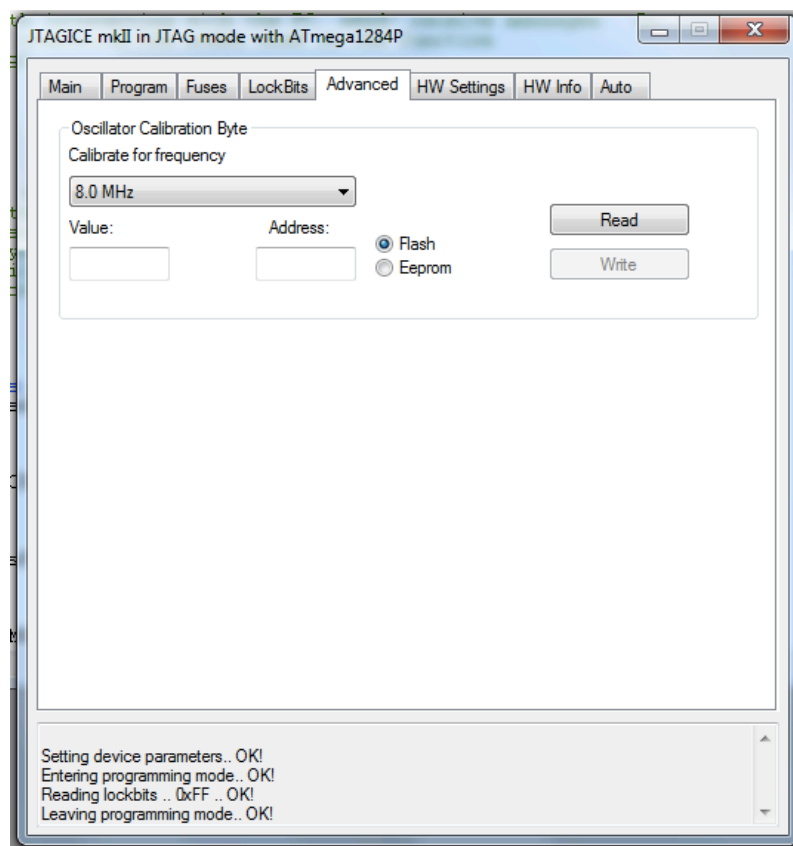


Figure 20 - LockBit settings

## Programming the microcontroller

In the 'Main' tab, select the proper device (ATmega 1284P for main control module, ATmega 168PA for user interface modules). If programming the main control module, use JTAG mode. If programming a user interface module, use ISP mode (Figure 21). Next, go to the 'Program' tab, select the proper .hex file and click 'Program' (Figure 22).

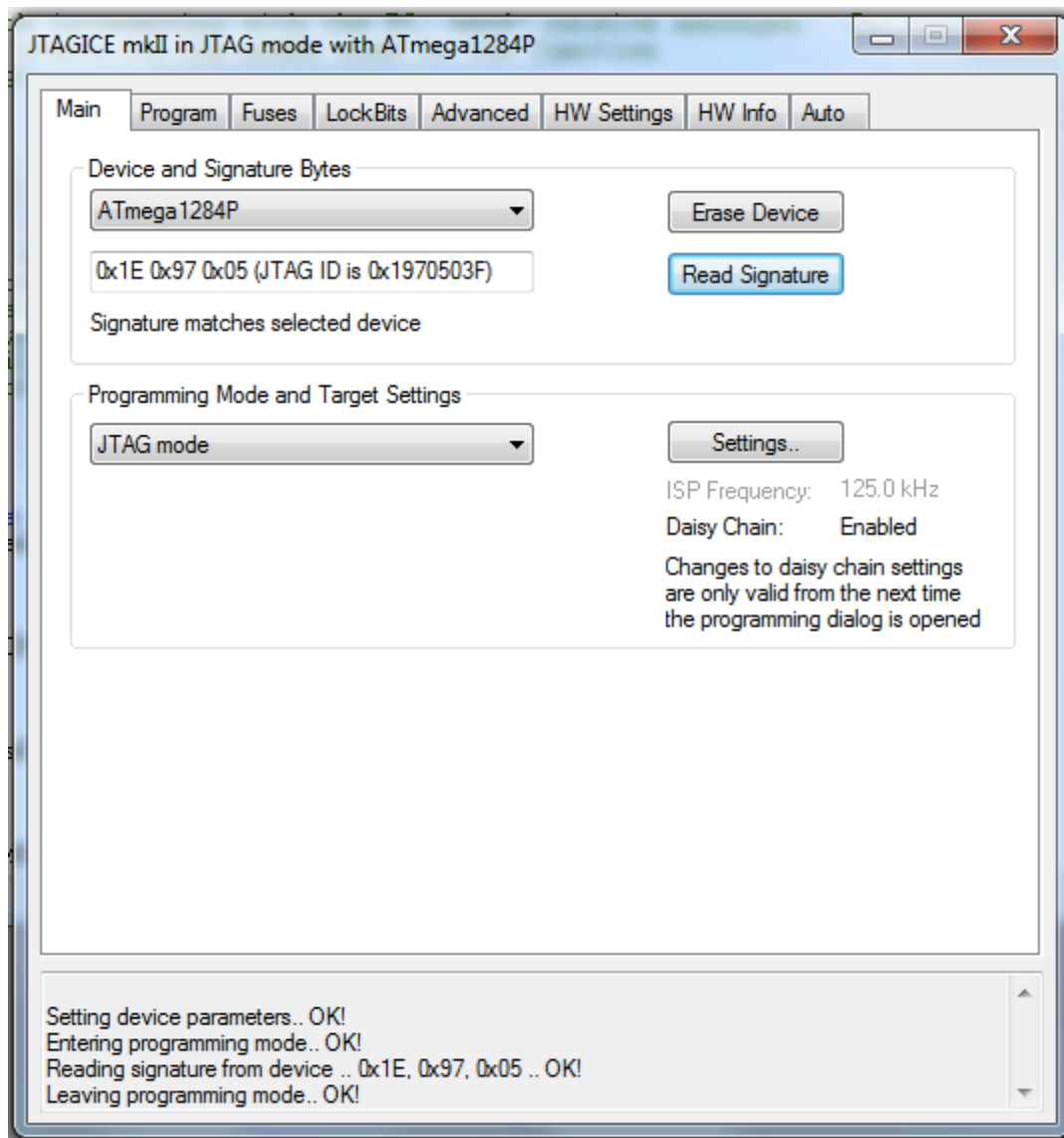


Figure 21 - Main tab settings

Check the printouts at the bottom of the display to make sure the device was programmed successfully. Enjoy!

## Setting up PuTTY

PuTTY can be used to make adjustments to the mode file if an SD card reader is not available. Additionally, PuTTY can be an invaluable tool for debugging since normal printf's cannot be used and the only alternative is serial messages passed to the computer over the USB connection. After plugging the main control unit into the computer, check which COM port it is using (Figures 22 & 23) then setup PuTTY to monitor that port (Figure 24).

Follow these pictures to determine which serial port the SABL is currently using.

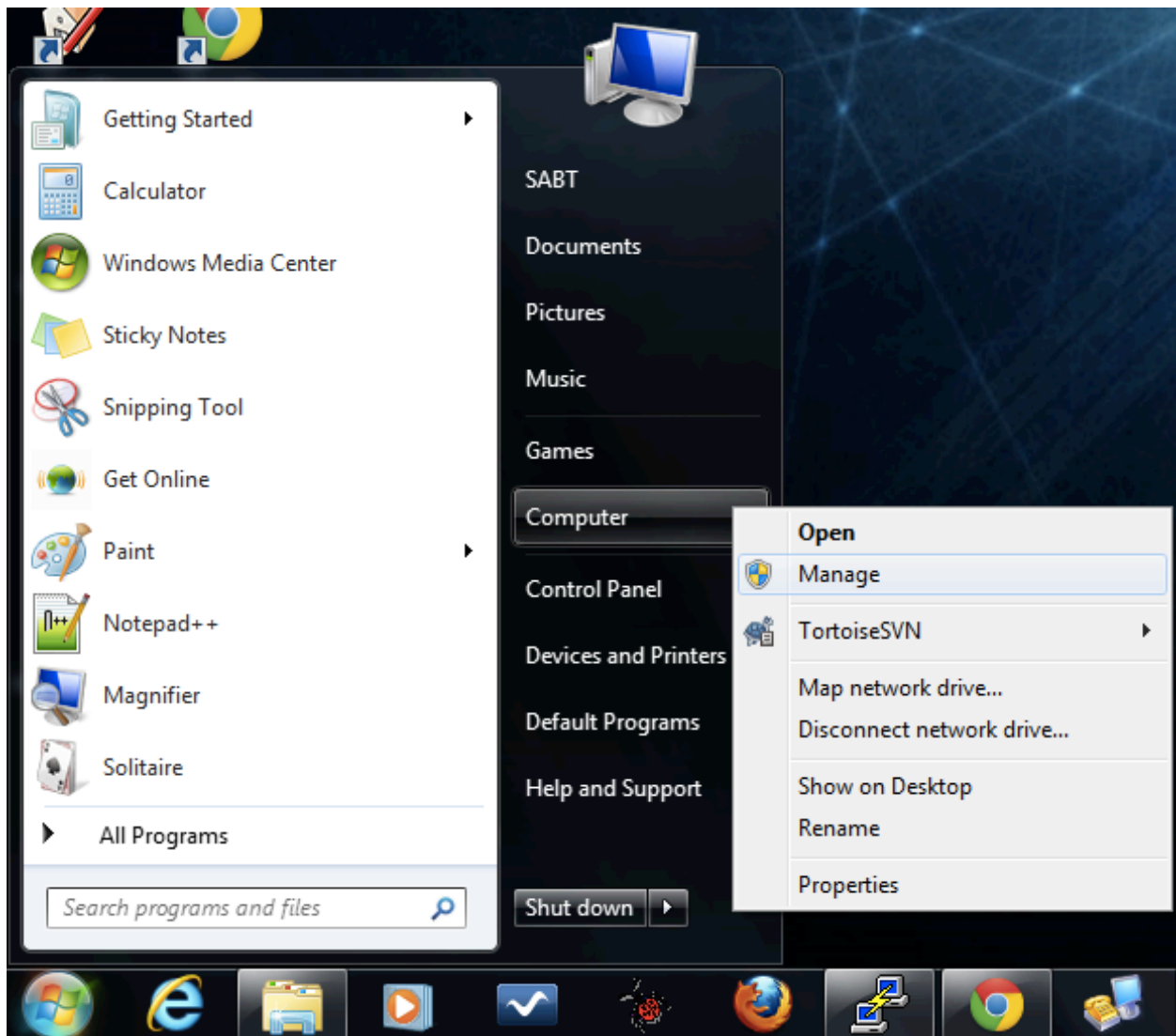


Figure 22 - Right click on Computer->Manage



Go to Device Manager then expand the Ports (COM & LPT) bullet. If you don't have anything else connected to your computer, the only one there will be the SABL connection. If there is more than one serial port listed, you might have to guess and check until you find the right one. In our case, the serial port the SABL is using is COM16, and this is what we will connect to with PuTTY.

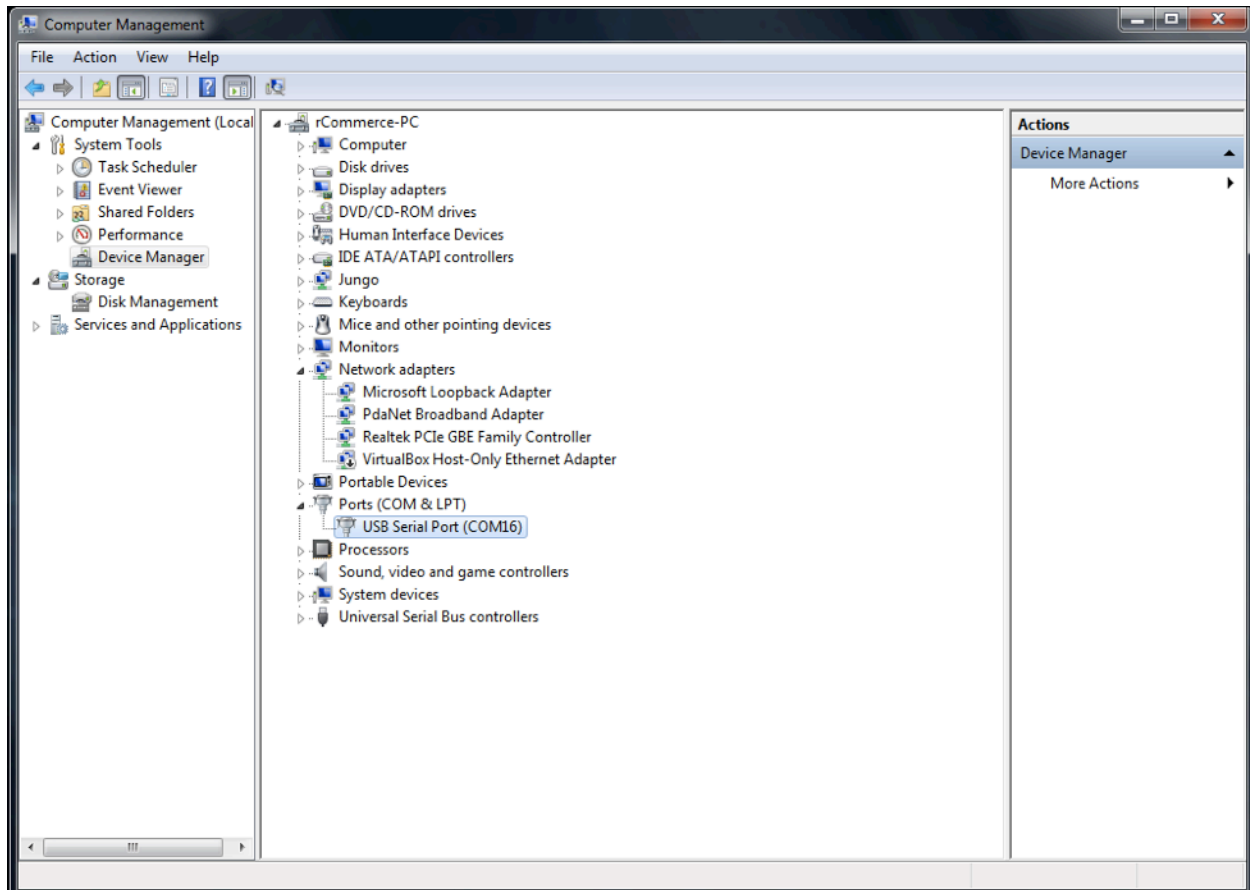


Figure 23 - Determining the serial port

Using the COM port you found above, setup PuTTY as below (Figure 24).

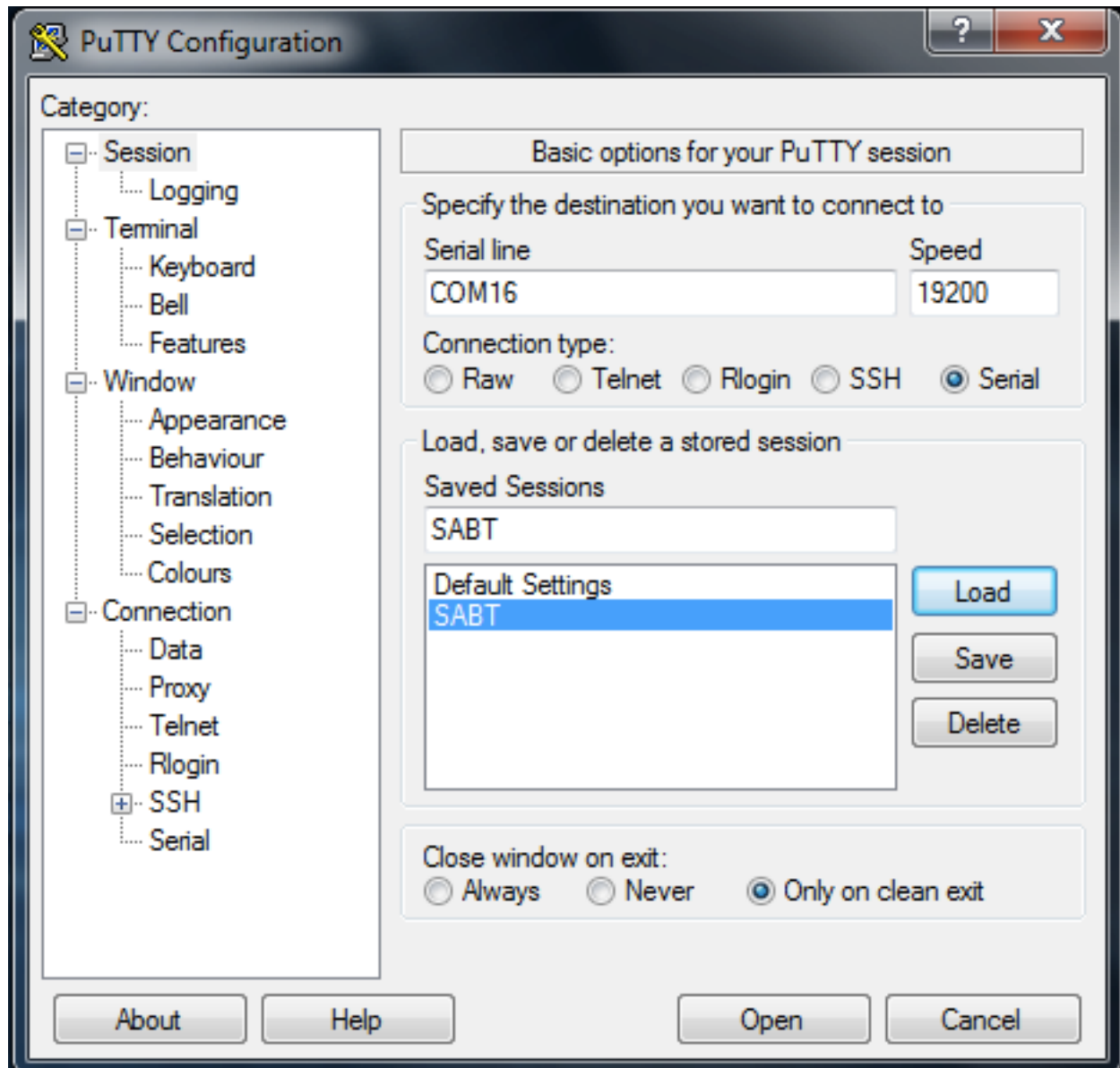
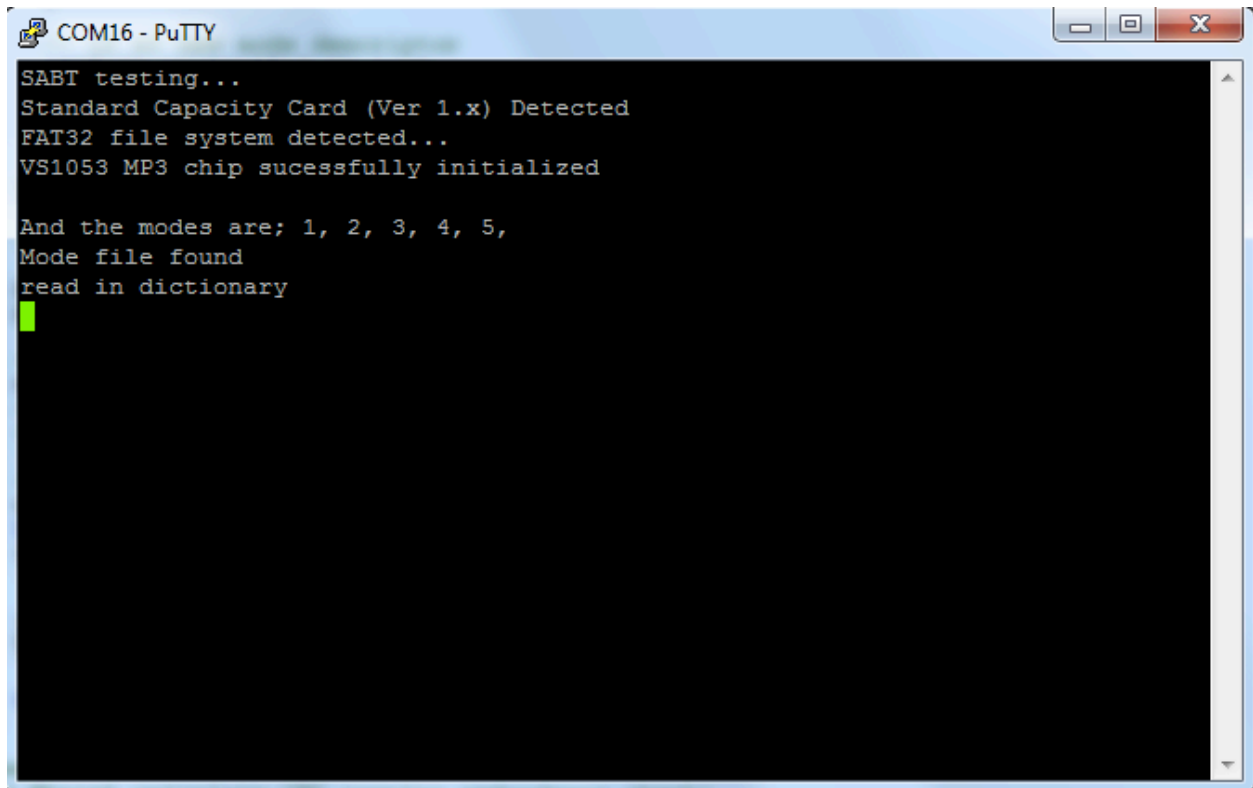


Figure 24 - Setting up PuTTY

## Using PuTTY

Start up the connection using the procedure defined above. After making sure all of the connections are made properly turn on the SABT. If everything was done properly, you should see something like below (Figure 25).



```
SABT testing...
Standard Capacity Card (Ver 1.x) Detected
FAT32 file system detected...
VS1053 MP3 chip sucessfully initialized

And the modes are; 1, 2, 3, 4, 5,
Mode file found
read in dictionary
█
```

Figure 25 - SABT initialization displayed in PuTTY

When you add print statements into the code using the PRINTF macro we made, you should see them here!

**IMPORTANT** - Be wary in using PRINTF debug statements. As a real-time system depending on timers and interrupts, the added time for transmitting messages over serial may cause things to break.

The following are the commands used to modify the mode file:

PCx

- Initializes the communication between the computer and the SABT.

PCM<1><2>...<n>\$

- Writes the line into the mode file. Use to specify which modes are available.