RENSSELAER MECHATRONICS

Bluetooth Communication

Objectives

This lab will cover the following topics:

- Bluetooth module connection and installation of PC drivers
- Testing the module using RealTerm loopback circuit
- Reading the data to a file using RealTerm and importing this into Matlab

Part 1: Basic Bluetooth Installation and Setup

Background Information

Bluetooth is a wireless transmission technology designed to connect devices over short distances (less than 100m). While it transfers data at a lower bandwidth than technologies such as WiFi, it can be used to connect several devices concurrently. It has found popular adoption as a replacement for physical data lines such as USB cables. It is also widely used in applications such as telephone headsets, game controllers, and wireless music connections. Bluetooth profile standards continue to evolve, and advances such as Bluetooth Low Energy (BLE, or BTv4.0) enable the use of personal health trackers or medical devices to be worn by users throughout the day.

The JY-MCU Bluetooth module is an inexpensive solution to add short-range radio communication to an Arduino unit. This can enable data collection where a physical connection is impossible, or be used to send commands to the Arduino for robotics or automation applications.

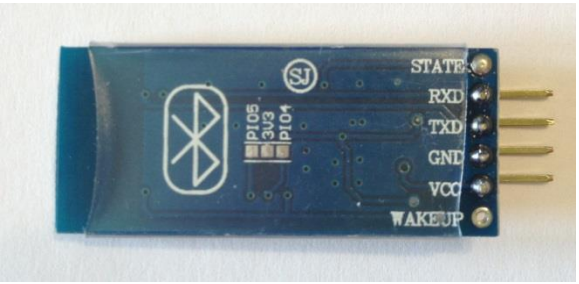


Fig. 1: JY-MCU Bluetooth module and pins

Installing the Bluetooth Module

- Connect the JY-MCU Bluetooth module to the Arduino. The MinSeg shield features a header pin for the Bluetooth module, as seen below.

Figure 1: Bluetooth MinSegShield Header

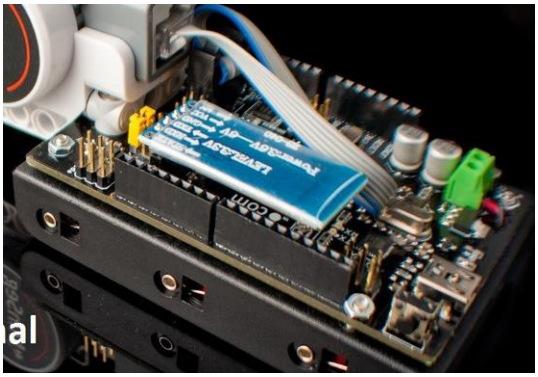
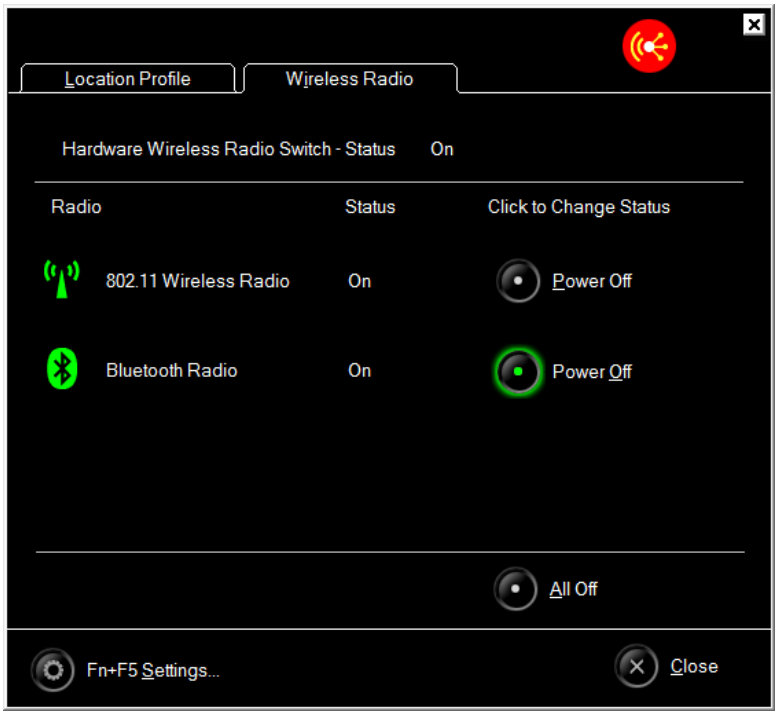
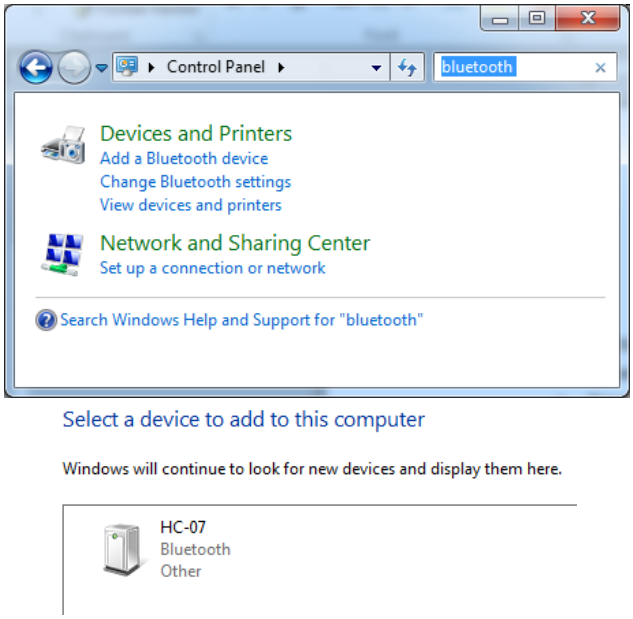


Figure 2: Bluetooth module on MinSegMega Header

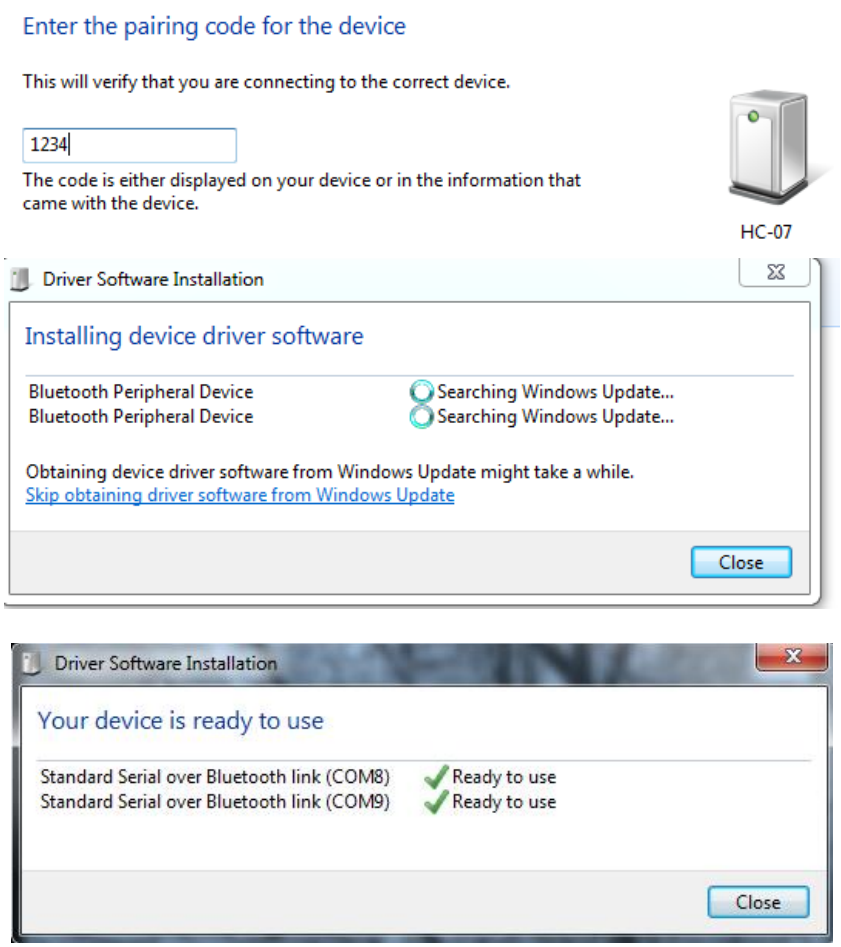
- Turn on your PC's Bluetooth radio. The panel to do this can usually be found by entering "radio" in the Windows or Control Panel search bar.



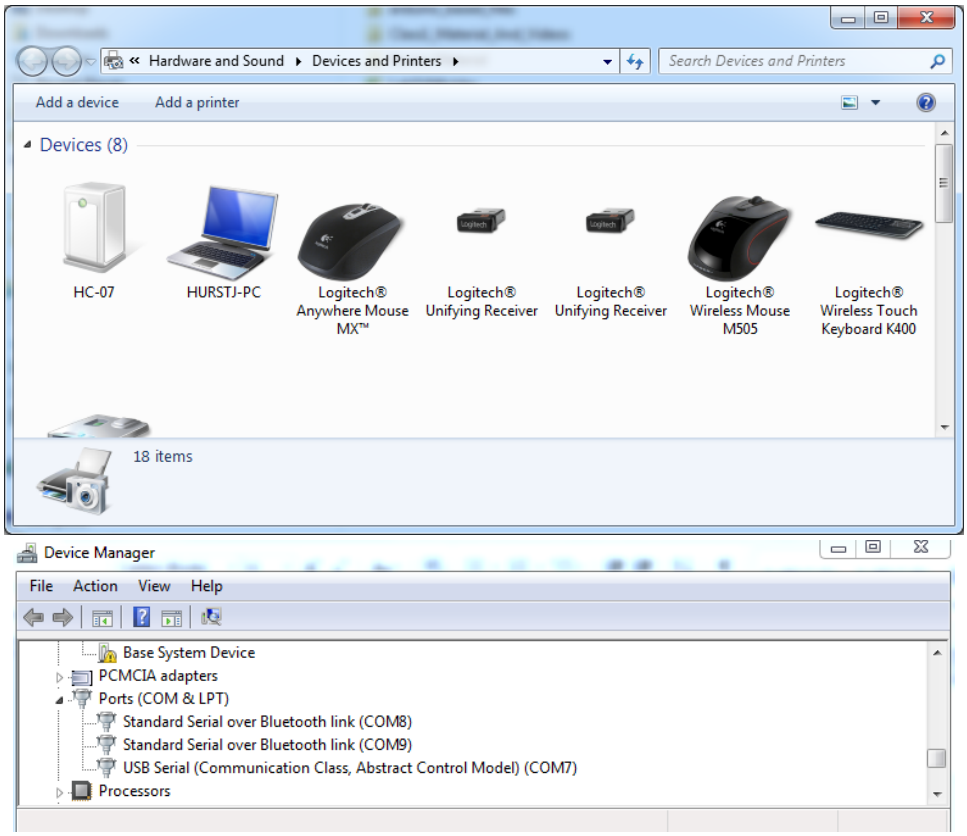
- Add the Bluetooth Device to your computer:
    1. Make sure the device is powered. The red 'State' LED will be flashing when turned on.
    2. Search for Bluetooth in the control panel. Select "Add a Bluetooth device"



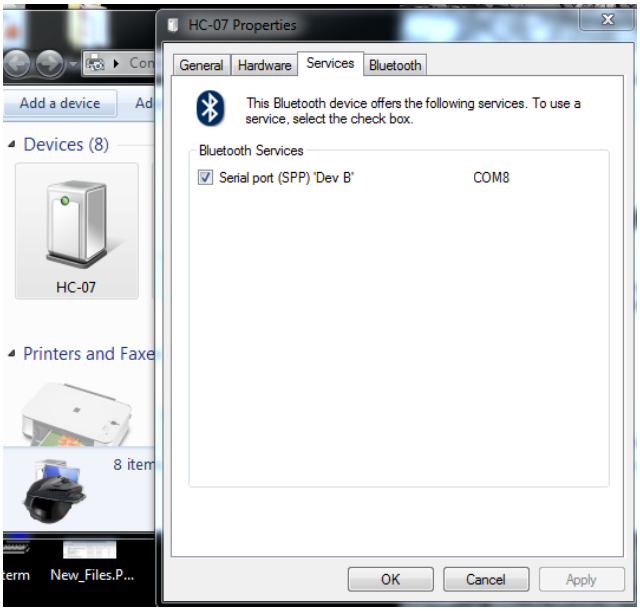
- Enter the device Paring code – 1234. Windows should search and install drivers for this device.



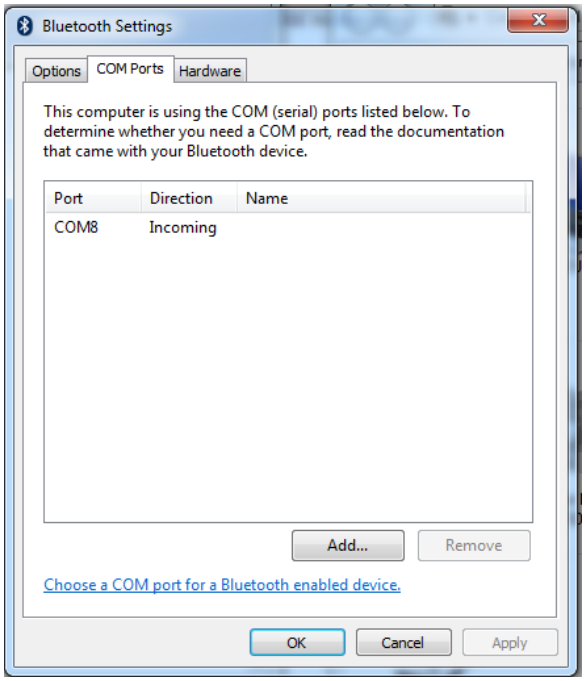
- You should now see the device in devices and printers and the device manager. It is seen below as ‘HC-07’ below, but your device name may appear as ‘linvor’.



Looking at the Device Manager, notice that two COM ports were installed. One is for receiving data, the other for sending. Sometimes this shows up in the properties windows of the Bluetooth device:



In this case, it did not show up and you have to check both to see which data is being sent to. Typically the lower value COM port is the one you will want to use.



Click “Add” and add an incoming our outgoing connection.

Occasionally Windows has trouble pairing the device after entering the paring code. In this case, restarting Windows and repeating the process usually solves the problem.

Part 2: Testing the Bluetooth device with loopback test

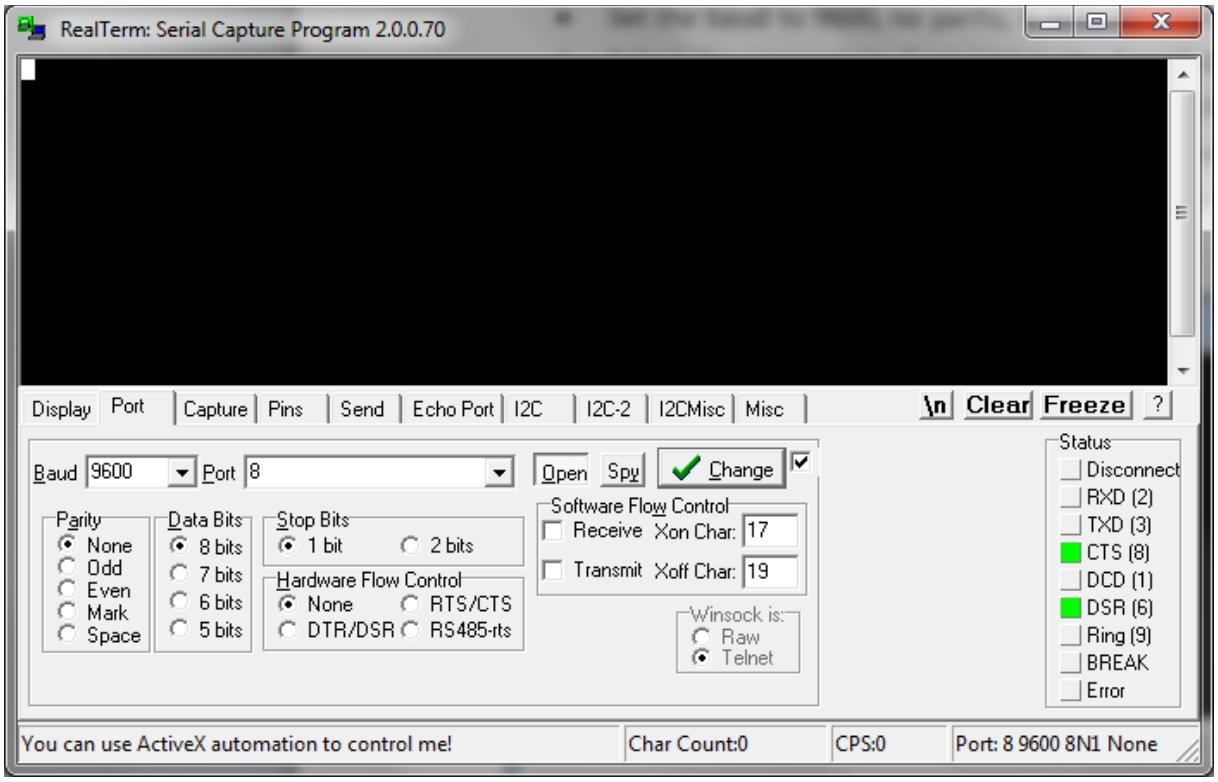
Background Information

The easiest method to test the Bluetooth module is a loopback test, where the Rx and Tx pins of the Bluetooth module are wired together. This way, the data sent over one pin will be physically routed to the receiving pin.

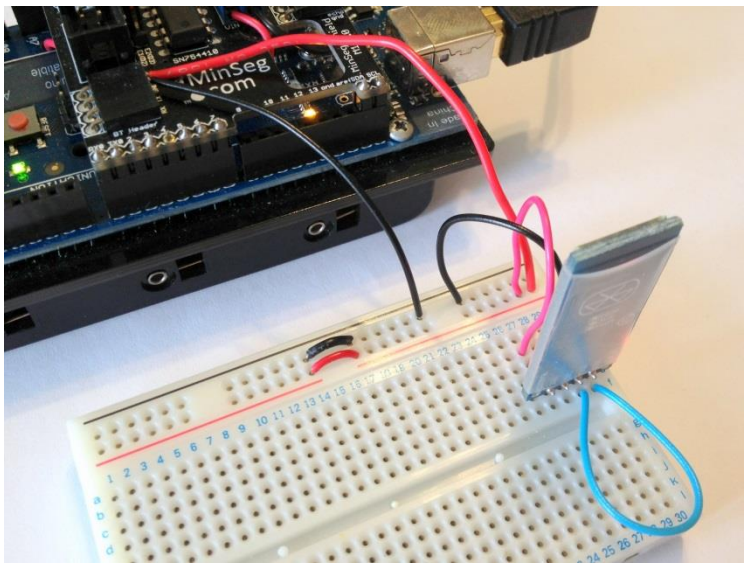
RealTerm is a terminal program that can be used to send, receive, and monitor the COM port. After installing RealTerm, it must be run in administrator mode.

Note: You do not have to have the Rx and Tx pins connected to anything in order to connect to the Bluetooth device from RealTerm.

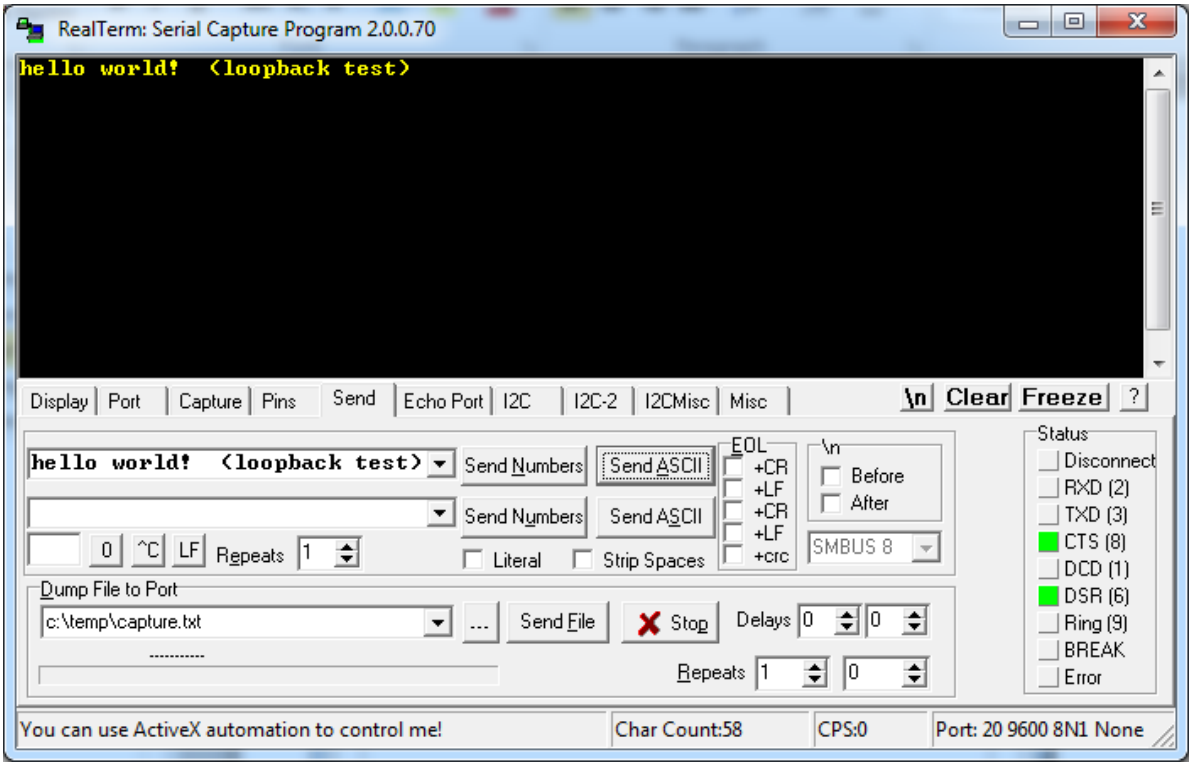
- Open Realterm in administrator mode
- Set the baud to 9600, no parity, 8 data bits, 1 stop bit
- Select the appropriate Com port in the "port" dropdown. You should select the lower port associated with the Bluetooth module, or whichever says "BthModem0"
- Click "Open" or "change" (the Open button needs to be depressed. If it connects, the blinking light on the Bluetooth module will be a solid color. The bottom right of the RealTerm window also shows the port information:



- Now you can wire the Rx and Tx pins together and connect the module to power. Using header wires or an external breadboard, connect the VCC and GND pins to the header slots on the MinSeg shield. Now, simply connect the RXD and TXD pins together, as seen below:



- Perform the loopback test:
    - Click on the “Send” tab. Enter a value and click “Send ASCII”. Whatever you sent should be received and displayed in the terminal window.



This is the basic loopback and tests the basic functionality of the Bluetooth module.

Part 3: Reading the Serial Data to a file and importing into Matlab

A Note on Ports

If you plan on using port 0 (which Arduino uses to download programs), you can NOT have the Bluetooth module hooked up to the RX0 on the Arduino (TXD on the Bluetooth module) when you try to program it. It will not be able to download the code. If you are only interested in obtaining data with Bluetooth, only use the TX0 pin on Arduino (or the RXD pin on the Bluetooth module). This way you can still receive data to your computer and program the Arduino.

NOTE: You can NOT use external mode with port0 and obtain data with Bluetooth. You must disable external mode to use port 0.

Sending 8 bit data from Arduino to Matlab using the Serial Port

In the following exercise, a signal is produced by a 'repeating sequence block', and then sent to the output ports. This is the easiest way to view and obtain data. Create the following Simulink diagram:

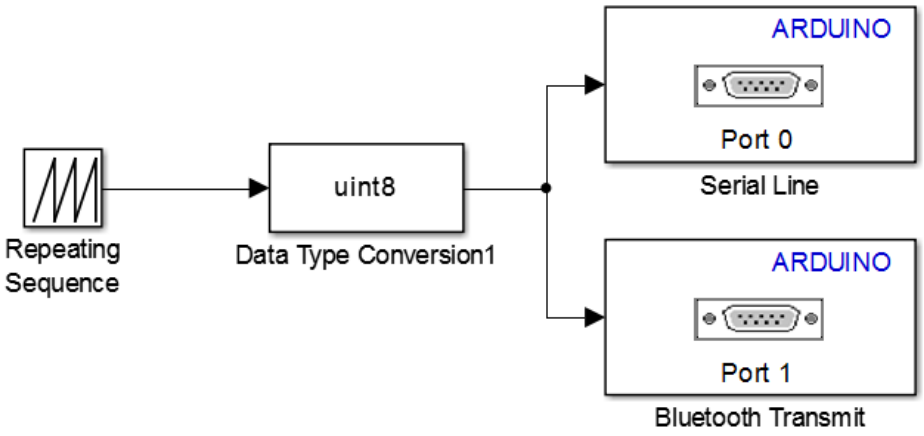
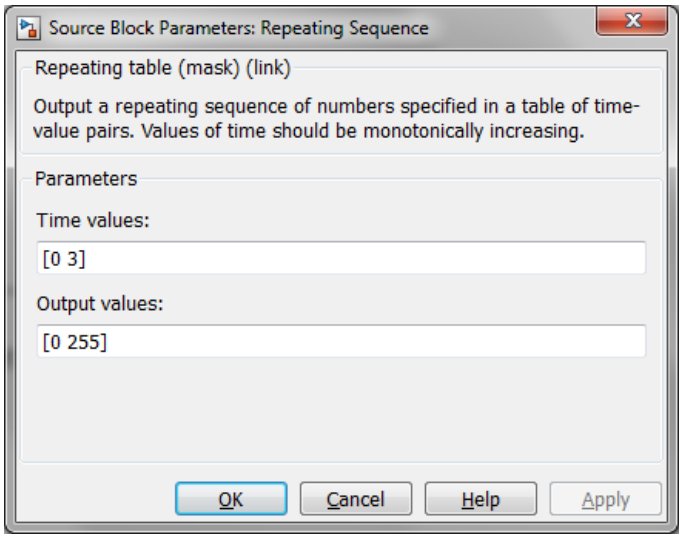


Figure 3: Bluetooth header is on Port 1 for MinSegShield, Port 3 for, MinSegMega

The transmission block sends data out over both the USB cable (port 0 – pins 0 and 1 on MEGA 2560) and to the Bluetooth module (port 1 – pins 18 and 90 on MEGA 2560). If Matlab generates an error when setting the second transmit block to Port 1, make sure you have first set up the model to be run on the Arduino hardware. Click 'Tools' -> 'Run on target hardware' -> 'Prepare to run'. Then select the ATMEGA 2560.

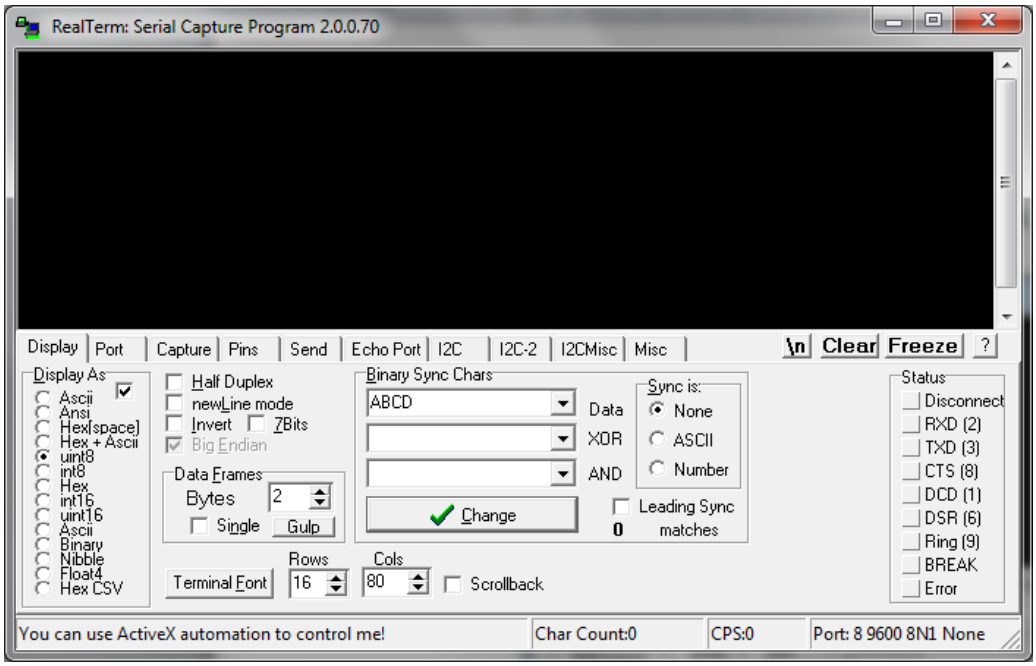
Notes: This configuration writes to both ports, so this will require twice as much processing. If you need performace, just use one.

Now, create the parameters for the signal block. By entering ‘time values’ of [0 3], you are creating a 3 second period for the sequence. ‘Output values’ of [0 255] will create a ramped signal between these limits.

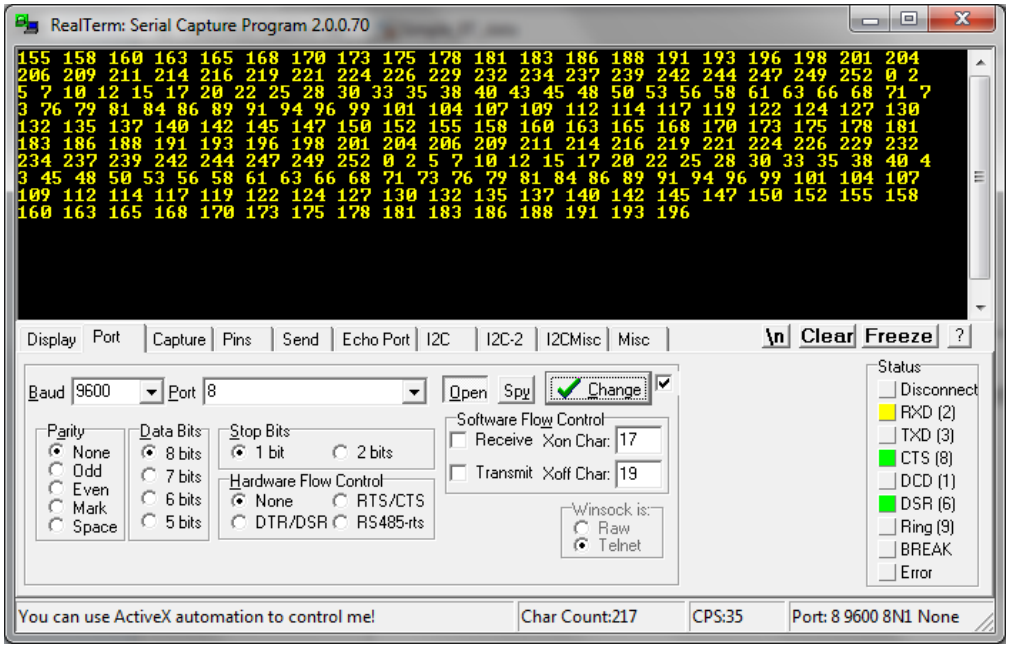


- Download and run this Simulink model on the Arduino.

- Open RealTerm to monitor the serial port and to save this data to a file. Make sure you are running the program in ‘Administrator’ mode.
    - In the Display Tab select “Display as” unit8. This way it knows that it is binary data and NOT ASCII characters.

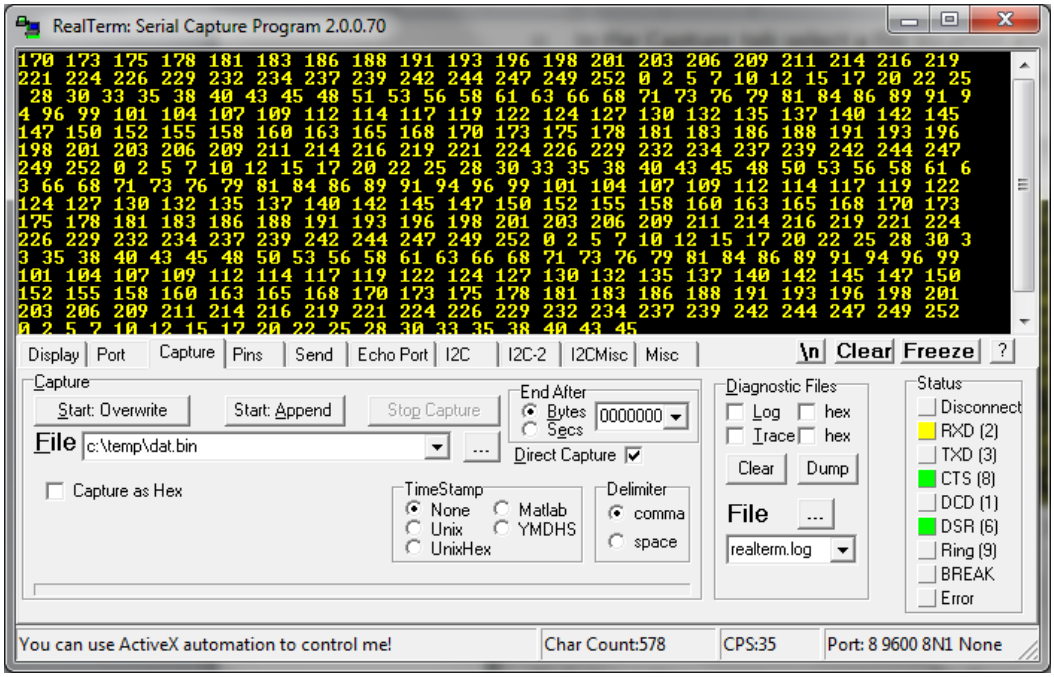


- In the Port Tab, select the proper baud rate and port number and click “Change”. You should start seeing your binary nicely displayed as a number:



Note: If you get an error, first try resetting Arduino, then closing reopening Matlab, then logging off/on your PC user. You may even need to restart the computer. Serial ports can hang and have to be reset.

- In the Capture tab select a file location and name the file. It shold end in “.BIN” to indicate it is binary data. Then click “Start Overwrite” to begin writing this data to a file.



- Click “Stop Capture” to stop taking data.

Saving Settings

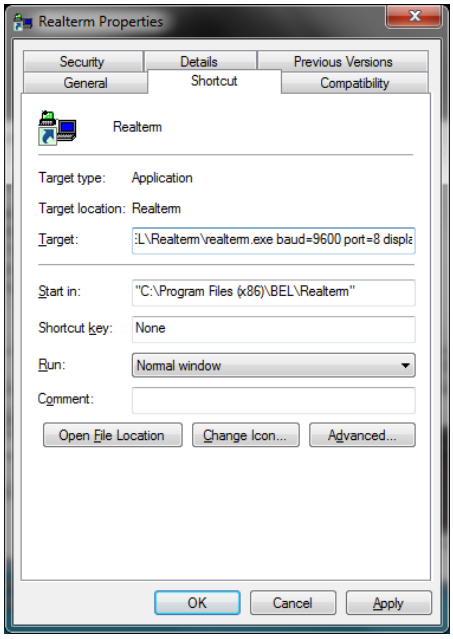
RealTerm doesn’t save settings. Instead, they are set from the comamnd line. We can edit the shortcut to the program to have it set the default values for us.

- In the “Target” line we can append add the following:

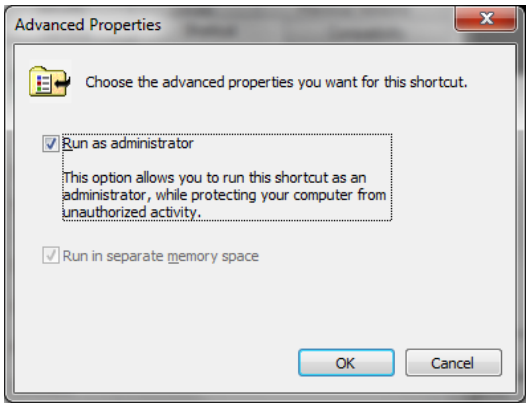
```
baud=9600 port=8 display=4 capture=C:\arduino \dat.bin
```

This sets the baud rate, the port to COM8, the display to the 4th checkbox item (which is uint8), and the file to capture the data to. With the ‘capture=’ line it will imediately begin taking data. If you want to just set the path, and begin taking data lager use this line:

```
baud=9600 port=8 display=4 capfile=C:\arduino \dat.bin
```



It is also convient to have it run as adminstrator automatically. Click on the 'Advanced' tab in the shortcut properties:

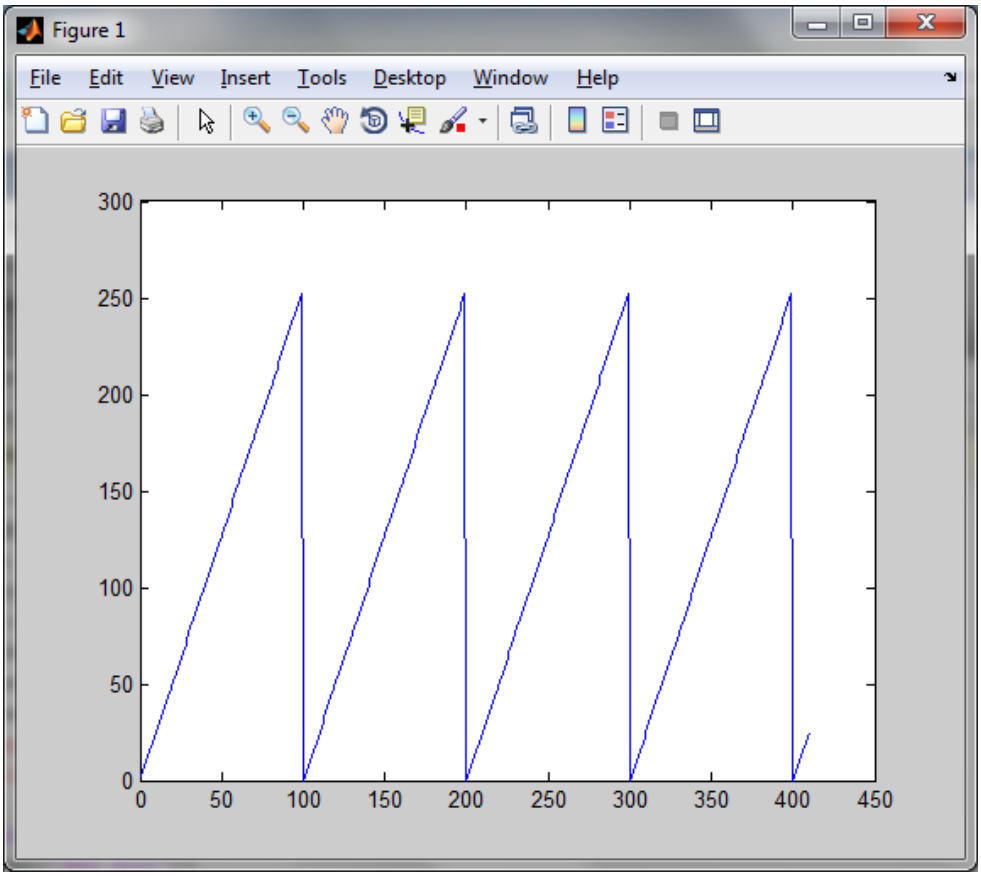


Reading the data in Matlab

Since the data collected is in binary format, it is easy to import into MATLAB. Create a script based on the following code snippet.

```
% Import binary serial data from a file:
fid=fopen('c:\temp\dat.bin');
dat=fread(fid);
plot dat
```

This will produce a plot such as the following:



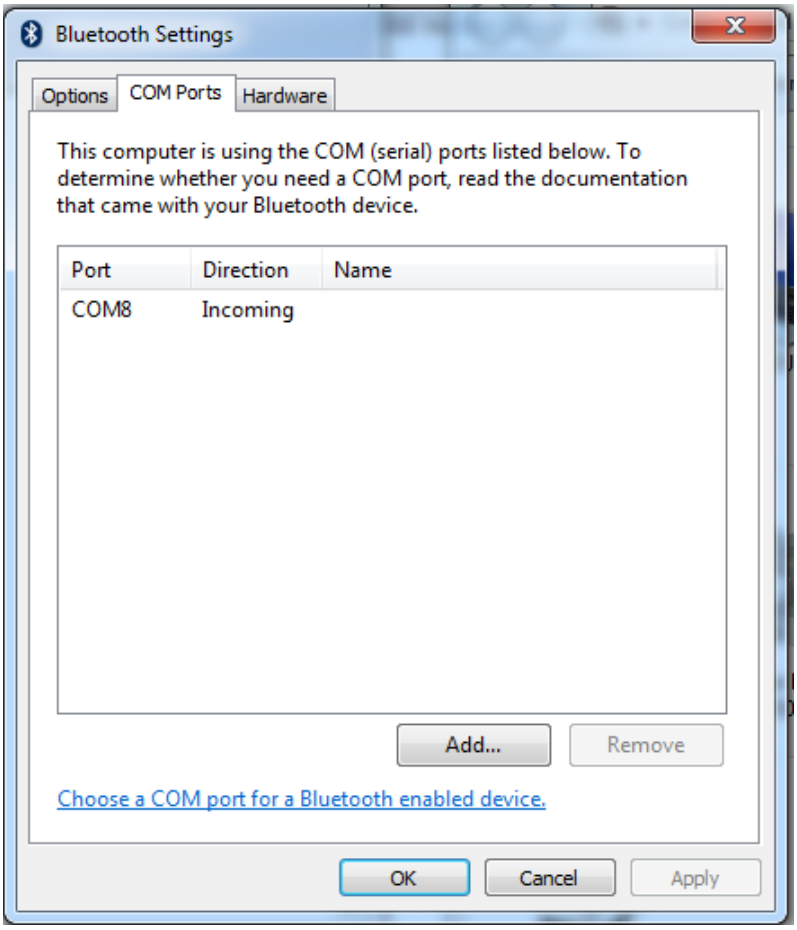
Checkpoint:

You have just successfully collected data from the Arduino hardware using Bluetooth. If you have batteries in the unit, you can turn on the "Batt On" switch and continue collecting data even after you unplug the USB cable. Switch the unit off, uncheck the "open" button under RealTerm ports, and then switch the unit on again. When you re-open the RealTerm port channel, you will notice the hardware has resumed signal generation and Bluetooth transfer.

Troubleshooting

Installation

Sometimes the Bluetooth device will install but will not show up as a COM port. In this case you have to assign a COM port to the device in the Bluetooth settings:



Click “Add” and add an incoming our outgoing connection.

Occasionally Windows has trouble pairing the device after entering the paring code. In this case, restarting Windows and repeating the process usually solves the problem.

RealTerm

If RealTerm is having trouble connecting to the serial port, try disabling and re-enabling the COM port in the device manager: