Daniel Kuzmenko

EEC 193 RADAR Project

How to connect a Bluetooth with an Arduino and Transfer Values in to MATLAB.

Purpose of this Paper

This paper will explain two concepts. The first is how to setup a wireless serial link via Bluetooth module connecting Arduino Uno and a computer. The second is how to transfer data from the Arduino UNO in to MATLAB and vice versa. This application note assumes that the person reading this already has some knowledge of programing Arduino UNO microcontrollers and MATLAB script programing.

Abstract

Sometimes it is desirable to have the Arduino microcontroller connected to a computer. One could potentially control the microcontroller or transfer data for processing. The simplest way of doing this is via a serial connection. A serial link is a bidirectional communication process by which information is sent over one bit at a time. The most common way of setting up a serial link is by using two wires. One would be a TX (transmit wire) and the other an RX (receive wire).

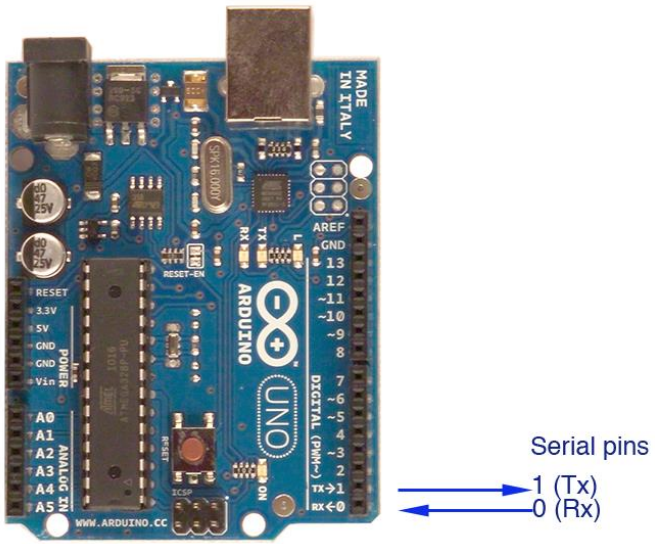

Figure 1

To connect two Arduino UNOs devices in a serial connection the first UNO's TX must be connected to the RX of the second UNO and the RX of the first UNO to the TX of the second UNO (see figure 2) .

Figure 2

Once the wires are connected as described above one can use the Serial.print() function in the Arduino UNO #1 and it will show up in the serial monitor of the UNO #2 and the opposite it also true. You can connect this also to a computer buy using a serial USB cable and connecting to the RX and TX ports of the Arduion UNO.

This is all good if one does not mind having a cable connection, but if a wireless link is needed for communication between the two UNOs or UNO-compute then one must consider a Bluetooth serial link.

Material and Software Needed for Bluetooth setup

1. HC-05 Bluetooth module.

These modules are really cheap on eBay and should not cost more than $15 and they should be mounted on a breakout boards. Make sure it is the **HC-05** module. Only the HC-05 can be reprogrammed to be a slave or a master device (will talk about this in more detail later on).

2. Bread board and some wires
3. A computer that is 32 bit and has Bluetooth (I only got the port forwarding to work on the 32 bit machine)
4. MATLAB
5. XPort by Curious Technology (used to forward com ports)
6. Arduino UNO (how can we forget)

Programing the Bluetooth module

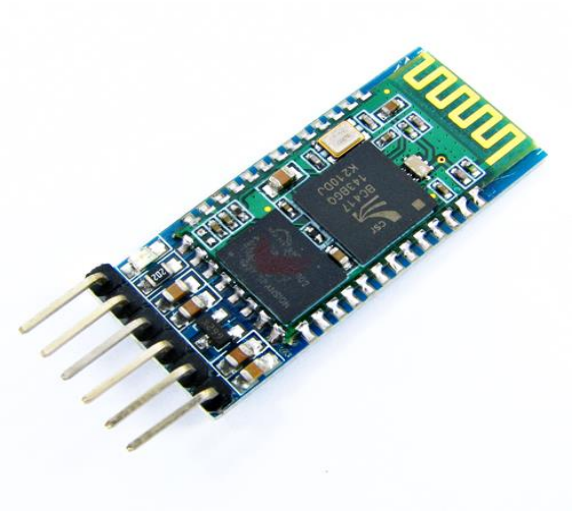

Figure 3 (HC-05 module on a breakout board)

The Bluetooth module first needs to be programmed before it can be used. You can reprogram the role of the Bluetooth to either a master or a slave, reprogram the pair password or the baud rate at which data is sent over the serial link. A slave module can pair with a master and a computer but the master can only pair with other modules. So for a UNO-computer link the module should be a slave. For an UNO-UNO link one module should be a master and the other a slave.

The programing will be done using an Arduino UNO and the serial monitor of the Arduino development environment.

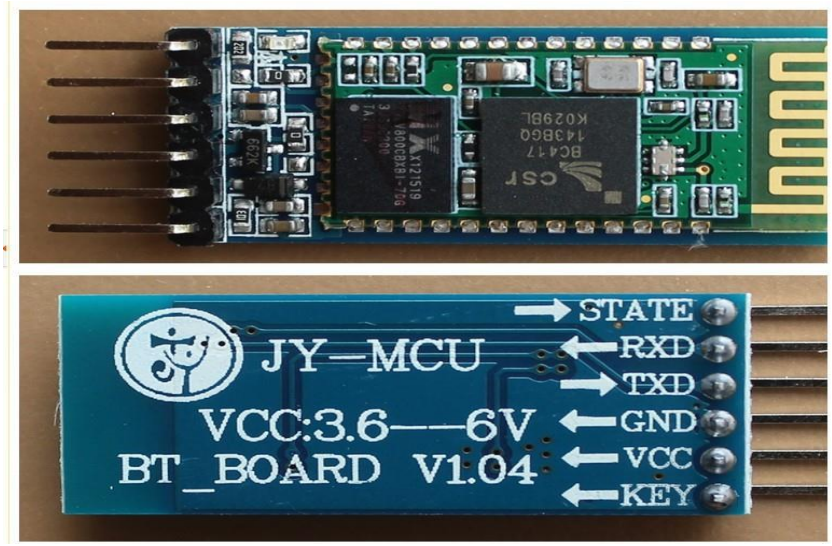

Figure 4 (HC-05 pin configuration)

In figure 4 the pins of the HC-05 module are shown. To start connect the module on to a breadboard. Connect the VCC of the HC-05 to the 5V pin on the UNO and GND pin to ground (keep the UNO powered off while connecting pins). Next connect the HC-05 TXD to the UNO's pin 10 and the HC-05 RXD to pin 11 on the UNO. Now under normal operation that is all the pins that need to be connected for the Bluetooth module to operate, but in order to program the module the HC-05 KEY pin will need to be set high so connect it to pin 9 rail (see figure 5). The *Bluetooth Programing Code* will take care of the rest

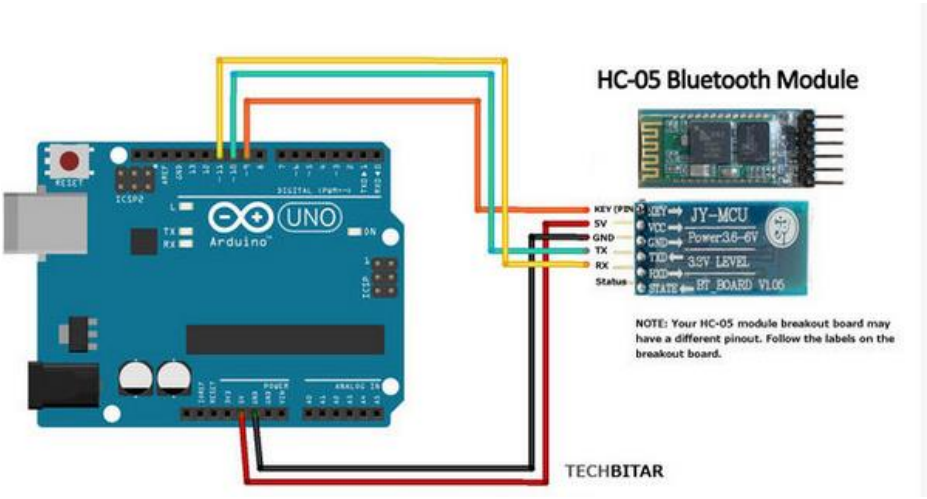

Figure 5 (programing connection)

Once the module is connected to the UNO unload the *Bluetooth Programing Code* which is given in the appendix of this paper. This code allows you to use AT commands to interface with the module and change its parameters. AT is a set of commands that one can use to reprogram the parameter of the HC-05 Bluetooth module. These codes will be entered in to the Serial Monitor provided by Arduino IDE. A table of most important parameter is listed in table 1 below. On can essentially reprogram the pairing password, baud rate and master/slave role of this module. For more information regarding the AT codes and the programing mode checkouts the Bluetooth reference in the appendix section. Once the module is set in to the AT programming mode the status on it will start to blink once every two second this is the indication that the module is in AT mode. Under normal operation the led will blink 3 times a second when it is not pared and 1 time per second when it is paired.

Type of Function	Command	Response	Parameter
Test	AT	OK	None
Restore Default Status	AT+ORGL	OK	None
Role (master of slave ….)	AT+ROLE?	+ROLE<Parameter> OK	0 =Slave 1= Master
Change Role	AT+ROLE=<Param>	OK	
What baud rate	AT+ UART?	+ UART=<Param>,<Param2>,<Param3> OK	Param1: baud rate(bits/s) The value (Decimal) should be one of the following: 4800 9600 19200 38400 57600 115200 23400 460800 921600 1382400 Param2:stop bit: 0----1 bit 1----2 bits Param3: parity bit
Change baud rate	AT+UART=<Param>,<Param2>,<Param3>	Ok	

Table 1

The HC-05 module already has a preprogrammed pairing passcode of “1234” so in this document we will not change this but one can essential look at the link provided to change this and other parameters. In order for a simple serial com link with a computer the role of the module needs to be changed to slave and the baud reset to a desired rate.

Change the Role

Once the *Bluetooth Programing Code* is uploaded to the UNO and the module is connected as described above open Serial monitor in Arduino IDE and type in “AT” (no quotation marks). The module will respond with “OK”. Now type in “AT+ROLE?” the module will respond with +ROLE:# (the # will either be 0 for slave or 1 for master). If the role is 0 then the module is already in the correct mode and if it is not to change it type in “AT+ROLE=0” at this will change it to slave role.

Change Baud Rate

To change the baud rate to 115200 type in “AT+UART=115200,1,2,\r\n” and this will change the baud rate to 115200, stop bit to be 2 bits, parity bit to be even parity.

And now your module is programmed and ready to pair with a computer.

Pairing module with computer

Once the module is programmed pairing should be easy. Connect the module according for figure 6.

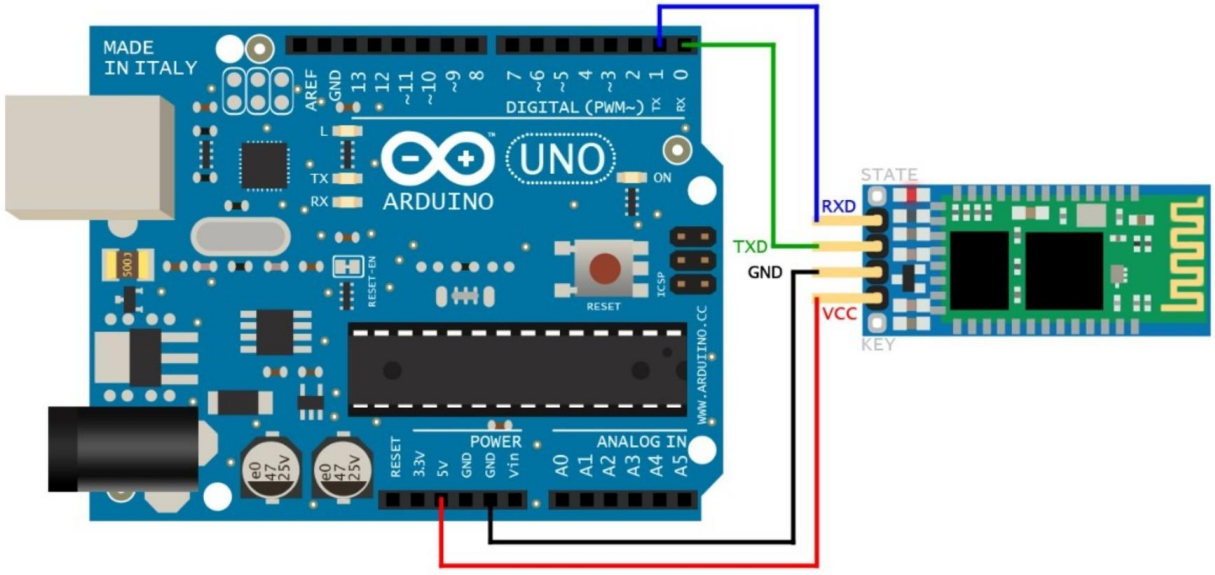

Figure 6

Once everything is connected the status led on the module will be blinking really fast to indicate that it is unpaired. Make sure that the Bluetooth module is turned ON on the computer. In Windows 7 go to ControlPanel>HardwareandSound>DevicesAndPrinters. Next click the *Add a Device* button on the banner in the upper left corner. In the window that opens you should see the HC-05 icon select it and on the next page click *Enter the device's pairing code* button.

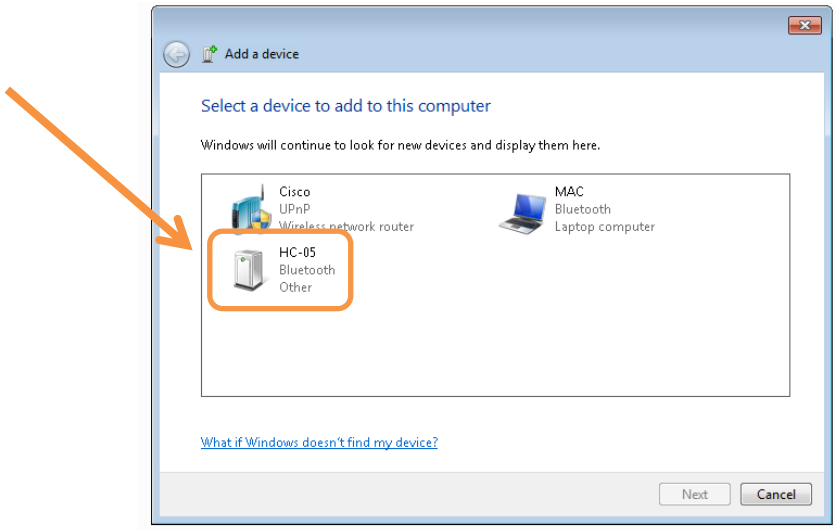

Figure 7

Enter the pairing code “1234” in to the slot provided and click next.

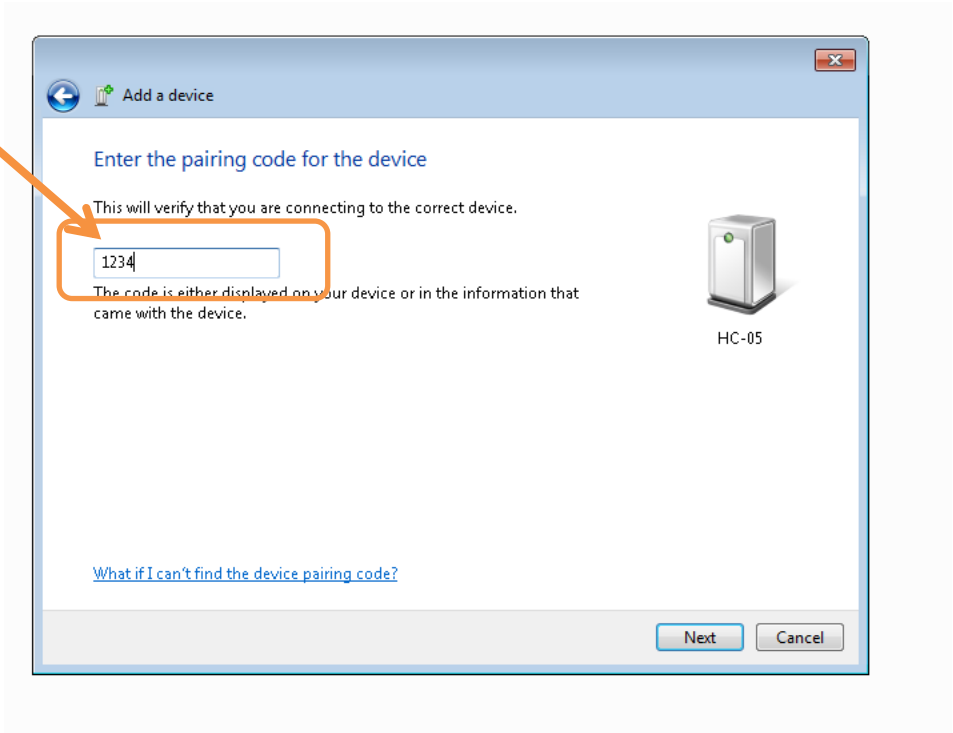

Figure 8

Once the module paired the indication led will start to blink once per second and the *Devices and Printer* will have a HC-05 icon indicating that it successfully paired with the computer.

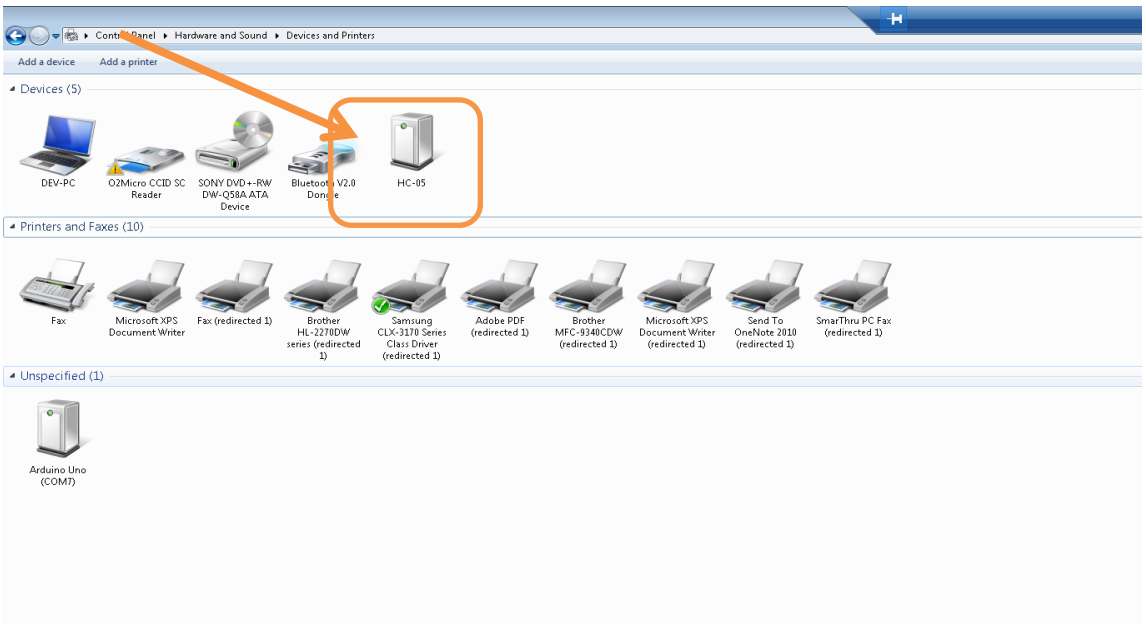

Figure 9

Double click the HC-05 icon and click the Services tab. This tab will show what serial COM port was assigned to the module. In this case the port is COM16.

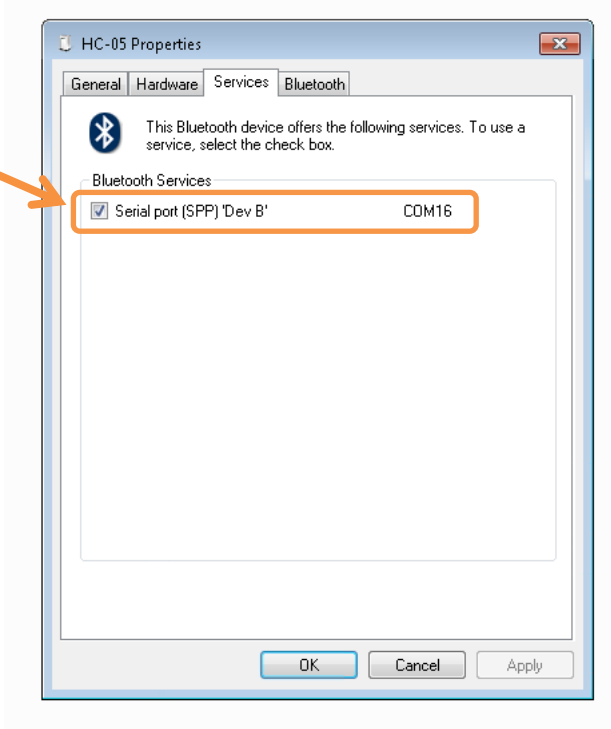

Figure 10

This is the COM port that will be used to send and receive serial data to and from the Arduino UNO.

Connecting to MATLAB to send and receive data

In order to use the Bluetooth COM port with Matlab first the port must be forwarded. Port forwarding is when a port say COM16 is forwarded to a different port say COM7. This is done via a software called XPort (See appendix for detail on XPort and download link). **XPort only works on 32bit processor.** This is why this guide is only for computer with 32 bit processors. If on can find a software for 64bit processor then all other instruction in this guide still apply. Port forwarding is needed when Matlab tries to connect directly to the Bluetooth COM port the Bluetooth will disconnect, but if you forward the port then the link will stay stable.

Port Forwarding

Download and open the XPort software (run it as administrator). Select the input port to be the one that was found to be allocated to the Bluetooth module (in this case it is COM16). Next select the correct baud rate (in this case 115200). Last select the virtual ports that Bluetooth will be forwarded to and click *Enable Ports* check box (COM9, COM30). This will create com ports linked to the Bluetooth port.

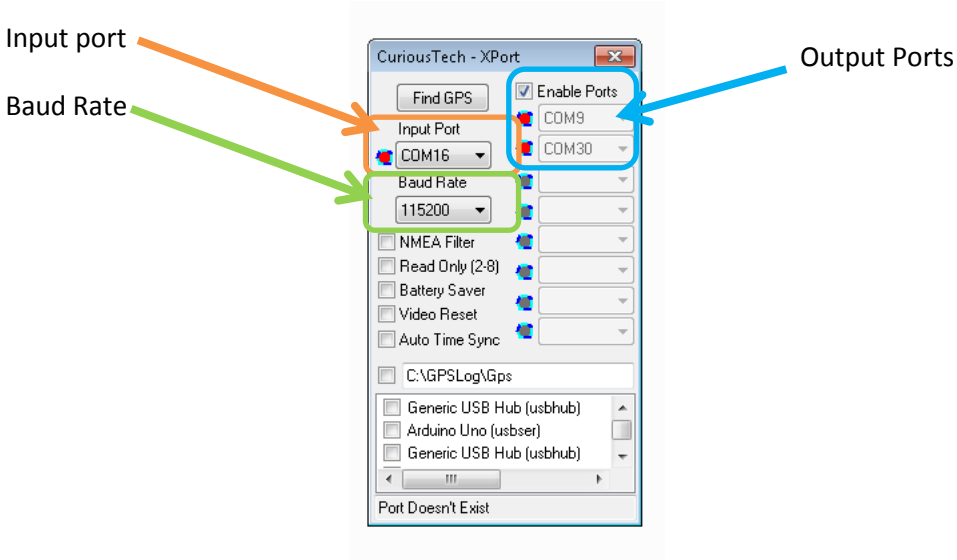

Figure 11

Serial link with MATLAB

The last step in this guide is to connect the UNO to Matlab via the Bluetooth serial link. Matlab has a very large rage of built in functions that control the serial communication (see appendix for more information on the functions). In Matlab create a serial com link object with the desired com port and baud rate.

```
com=serial('COM9','BaudRate',115200);
```

Next open the com object. Once this is done Matlab workspace should have a defined com object (see figure 12)

```
fopen(com);
```

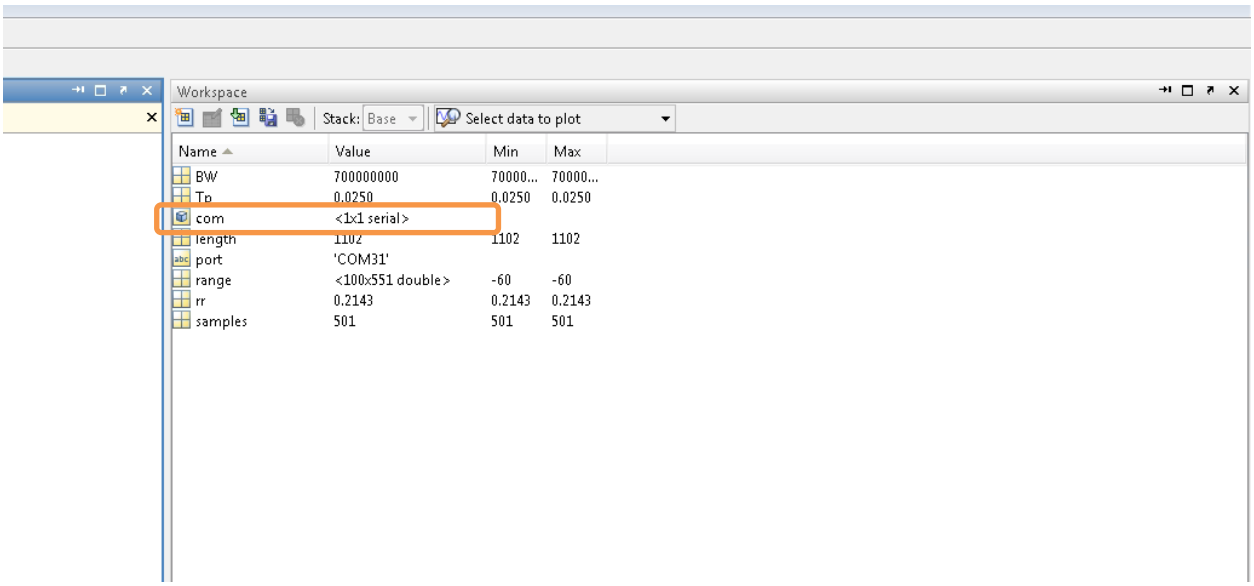

Figure 12

Remember that once the object is no longer needed one must disconnect from the com port in order to release it. Also the UNO should have the same baud rate that was programmed in to the module and Matlab

```
fclose(com);
```

To send data from Matlab in the scripted or terminal type

fprintf(com,'Hello');

This will send an ASCII sting to the UNO via Bluetooth. In the UNO to read the transmitted string use

Serial.read()

This will read that data coming in on the serial line. To send data from UNO use the function

Serial.print('Hello')

This will output data from UNO on to the serial link and to read this data in Matlab us the function.

```
fgets(com);
```

This function will read that data coming in to Matlab through the com object. The data is in ASCII format so if one desires number then built-in conversion functions in Matlab or UNO should be used

More information on this can be found in the links provided in the appendix and there is also much more information on the wed specifically the Arduino forums.

Appendix

Bluetooth Programing Code

```
#include <SoftwareSerial.h>
//Defining the TX-11 and RX-10
SoftwareSerial BTSerial(10, 11);

void setup()
{
  //This is the Pin KEY is connected to and this is how it is activated
  pinMode(9, OUTPUT);
  digitalWrite(9, HIGH);
  //set baudrate
  Serial.begin(9600);
  Serial.println("Enter AT commands:");
  BTSerial.begin(38400);
}

void loop()
{
  if (BTSerial.available())
    Serial.write(BTSerial.read());
  if (Serial.available())
    BTSerial.write(Serial.read());
}
```

References

HC-05 Bluetooth module AT codes

http://www.linotux.ch/arduino/HC-0305_serial_module_AT_commamd_set_201104_revised.pdf

Xport link

http://www.curioustech.net/xport.html

Matlab Serial Communication

http://www.mathworks.com/help/matlab/serial-port-devices.html