



# Setting up a Wireless serial connection with your Wiring hardware using Bluetooth

By Mitchell Page Mpag5967 [at] mail.usyd.edu.au  
Key Centre of Design Computing and Cognition, University of Sydney , Australia

Normally you would use a serial USB cable connected from your computer to your Wiring board for receiving data sent from a computer running processing to your board.

This tutorial demonstrates how to execute this same process, but wirelessly using Bluetooth. We will test this by being able to turn an LED on/off wirelessly. You will need to use an adaptor to power your board for this tutorial. This tutorial covers communication from the computer to the board; not from the board to the computer.

## Equipment:

You will need Bluetooth Dongle for you computer and Bluetooth Modem for connecting to Wiring board. We specifically used the following

<http://www.sparkfun.com/products/9434> and [http://www.sparkfun.com/commerce/product\\_info.php?products\\_id=158](http://www.sparkfun.com/commerce/product_info.php?products_id=158)

This tutorial was developed for a computer running Windows XP Service Pack 2 (which has included Bluetooth support). It has not been tested on Mac's or Linux.

## Process:

Step 1. We firstly need to install the Bluetooth module onto the computer and install the drivers. If the module came with a CD, put this into your CD-ROM drive and then plug the module into one of your spare USB ports. Windows should automatically detect the device and if it asks for drivers, point to the CD-ROM drive. Once the drivers have been installed, Windows may need to reboot.

Step 2 . Once your computer has rebooted, plug your Bluetooth module in (if it isn't already). A Bluetooth icon should appear in your taskbar (Windows XP SP2 has this Bluetooth support feature). Your Bluetooth module is not set up and running as a serial port. Now we need to set up the modem on the wiring board.

Step 3. We are going to create the setup in Figure 1. First of all connect your LED to your wiring board on whichever pin you want EXCEPT pins 2 or 3 (we have used Pin 8 in this tutorial). See this example for how to connect an LED ( [http://wiring.org.co/learning/examples/led\\_blinks.html](http://wiring.org.co/learning/examples/led_blinks.html) ). Once you have connected your LED to the board it is time to connect your Bluetooth modem. The Bluetooth modem we are dealing with has 4 labeled pins, PWR, GND, Rx and Tx. It is pretty straight forward: connect the PWR pin to the 5V pin on Wiring, connect the GND pin to the GROUND pin on wiring. Then connect the Rx pin to the Tx pin of Wiring (digital pin 3) and connect the Tx pin to the Rx pin of Wiring (digital pin 2). Once it is connected then plug your AC adaptor into the board, if everything is connected right a small green LED will start blinking on the Bluetooth modem.



Bluetooth Modem



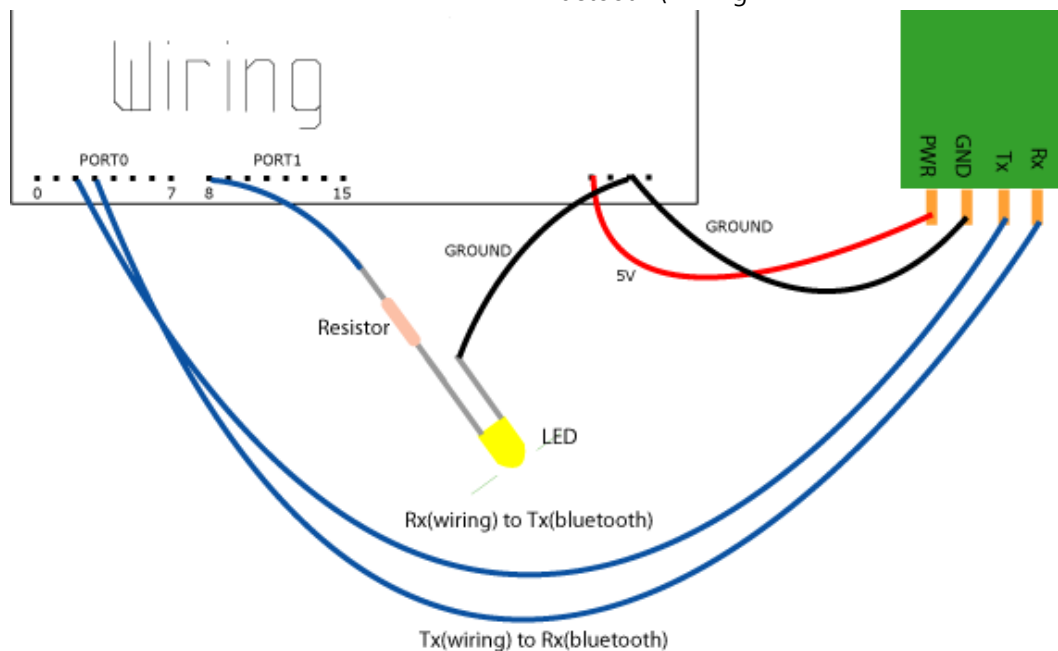


FIGURE 1.

Step 4. Now we need to set up a connection from the computer to the Bluetooth modem. With your Bluetooth module plugged in, double click the Bluetooth icon in the task bar and select 'ADD' a device. It should find a device called BlueRadios – which is your modem. Select to connect to it. It will ask you for a passkey. The passkey for the Bluetooth modem we used is 'default' (see [http://www.sparkfun.com/datasheets/RF/BlueSMiRF\\_v1.pdf](http://www.sparkfun.com/datasheets/RF/BlueSMiRF_v1.pdf)). If you used a different Bluetooth modem to the one we used, you will need to consult the product documentation to find the passkey. Once you have entered that passkey your computer will now remember that Bluetooth modem and be able to communicate with it now and in the future.

Step 5. We need to now find out which COM port our Bluetooth communication is on. Double click on the Bluetooth icon in the taskbar again and select the BlueRadios device from the devices tab, then click the COM Ports tab and take note of the port number the Bluetooth connection is using. In our case, the COM port is COM7

Step 6. Now that you have connected all your components, you need to upload your program to the Wiring board. YOU NEED TO USE A USB CABLE TO UPLOAD YOUR PROGRAM – YOU CANNOT USE BLUETOOTH WIRELESS TO UPLOAD PROGRAMS TO THE BOARD. YOU ALSO NEED TO DISCONNECT THE RX AND TX WIRES FROM THE BLUETOOTH MODEM WHILE YOU UPLOAD. The code for the program we use is attached below . Once you have connected the serial USB cable, upload your program. Once it has uploaded successfully, reset your board and remove the cable.

Step 7. Now we need to configure the processing code. The code we used is attached below . You need to find out which number in the array the Bluetooth COM port is. Run the code once and take note of the COM ports which appear in the output pane.

Eg. COM1 COM3 COM7 In our case, the Bluetooth was connected to COM7, which is the number 2 position in our serial array (COM1 = 0, COM3 = 1, COM7 = 2). So we update the processing code to the following so it will use the Bluetooth connection to send serial data. Once you have updated this code, run the program again (see Step 8).

Step 8. After you first run the program, the computer will establish a connection with the Bluetooth modems – you will see a red light begin to blink on the modem – it will finally stay red once it has established a connection but may take about a further 10 seconds before it can receive data. The connection process will take about 15 seconds all up. After about 15 to 20 seconds, your processing program will now be connected to the Wiring board *wirelessly* . Now press the buttons on the processing interface and watch the LED turn on and off.

#### Code:

Wiring code – this code will wait for signals from processing. When it receives a H signal, the LED on pin 8 will turn on. When it receives an L signal the LED on pin 8 will turn off.

```
char val;           // variable to receive data from the serial port
int ledpin = 8;    // LED connected to pin 48 (on-board LED)

void setup() {
  pinMode(ledpin, OUTPUT); // pin 48 (on-board LED) as OUTPUT
  Serial1.begin(9600);      // start serial communication at 9600bps
}

void loop() {
```

```

void loop() {
  if( Serial1.available() )      // if data is available to read
  {
    val = Serial1.read();        // read it and store it in 'val'
  }
  if( val == 'H' )                // if 'H' was received
  {
    digitalWrite(ledpin, HIGH); // turn ON the LED
  } else {
    digitalWrite(ledpin, LOW);   // otherwise turn it OFF
  }
  delay(100);                    // wait 100ms for next reading
}

```

Processing code – this code creates two buttons. When the left button is clicked it will send an 'H' signal to the board and turn the LED on. Or if the right button is clicked it will send an 'L' signal to Wiring board to turn the LED off.

```

//import class to set up serial connection with wiring board
import processing.serial.*;
Serial port;

//button setup
color currentcolor;
RectButton rect1, rect2;
boolean locked = false;

void setup() {
  //set up window
  size(200, 200);
  color baseColor = color(102, 102, 102);
  currentcolor = baseColor;

  // List all the available serial ports in the output pane.
  // You will need to choose the port that the Wiring board is
  // connected to from this list. The first port in the list is
  // port #0 and the third port in the list is port #2.
  println(Serial.list());

  // Open the port that the Wiring board is connected to (in this case 1
  // which is the second open port in the array)
  // Make sure to open the port at the same speed Wiring is using (9600bps)

  port = new Serial(this, Serial.list()[2], 9600);

  // Define and create rectangle button #1
  int x = 30;
  int y = 100;
  int size = 50;
  color buttoncolor = color(153, 102, 102);
  color highlight = color(102, 51, 51);
  rect1 = new RectButton(x, y, size, buttoncolor, highlight);

  // Define and create rectangle button #2
  x = 90;
  y = 100;
  size = 50;
  buttoncolor = color(153, 153, 153);
  highlight = color(102, 102, 102);
  rect2 = new RectButton(x, y, size, buttoncolor, highlight);
}

void draw() {
  background(currentcolor);
  stroke(255);
  update(mouseX, mouseY);
  rect1.display();
  rect2.display();
}

void update(int x, int y) {

```

```

    if(locked == false) {

        rect1.update();
        rect2.update();
    } else {
        locked = false;
    }

    //Turn LED on and off if buttons pressed where
    //H = on (high) and L = off (low)
    if(mousePressed) {
        if(rect1.pressed()) {          //ON button
            currentcolor = rect1.basecolor;
            port.write('H');
        } else if(rect2.pressed()) {    //OFF button
            currentcolor = rect2.basecolor;
            port.write('L');
        }
    }
}

class Button {
    int x, y;
    int size;
    color basecolor, highlightcolor;
    color currentcolor;
    boolean over = false;
    boolean pressed = false;

    void update() {
        if(over()) {
            currentcolor = highlightcolor;
        } else {
            currentcolor = basecolor;
        }
    }

    boolean pressed() {
        if(over) {
            locked = true;
            return true;
        } else {
            locked = false;
            return false;
        }
    }

    boolean over() {
        return true;
    }

    void display() {

    }
}

class RectButton extends Button {
    RectButton(int ix, int iy, int isize, color icolor, color ihighlight) {
        x = ix;
        y = iy;
        size = isize;
        basecolor = icolor;
        highlightcolor = ihighlight;
        currentcolor = basecolor;
    }

    boolean over() {
        if( overRect(x, y, size, size) ) {
            over = true;
            return true;
        } else {
            over = false;
            return false;
        }
    }
}

```

```
    }  
  
    void display() {  
        stroke(255);  
        fill(currentcolor);  
        rect(x, y, size, size);  
    }  
}  
  
boolean overRect(int x, int y, int width, int height) {  
    if (mouseX >= x && mouseX <= x+width &&  
        mouseY >= y && mouseY <= y+height) {  
        return true;  
    } else {  
        return false;  
    }  
}
```

Wiring is an open project initiated by [Hernando Barragán](#). It is developed by a [small team of volunteers](#).

[© Info](#), Processing [© Info](#)