

Bluetooth Arduino Interface

Hisham Hassan / A42496486

4/5/13

Executive Summary:

An Arduino board communicates with a computer via serial USB cable. This application note is intended to demonstrate how to execute data communication between an Arduino Uno board and a Windows PC using wireless bluetooth communication. This will be done by turning an LED on/off wirelessly.

Keywords:

Bluetooth, Arduino, Sketch, Integrated Development Environment, Functions.

Introduction:

Bluetooth is a wireless technology standard that is used to exchange data over short distances using short-wavelength radio transmission. It was created by the telecom company Ericsson in 1994, as a standardized as IEEE 802.15.1. However, this standard is no longer maintained. Bluetooth operates at a range of 2400 - 2483.5 MHz as a packet based protocol using master-slave structure, and provides a secure way to connect and exchange information between devices.

The Arduino is a single-board microcontroller designed to make the process of using electronics in multidisciplinary projects more accessible. It is an open source hardware, which means that its information about the hardware design in addition to the software design are all released to the public.

Objective:

To turn on an LED on/off wirelessly by communicating to an arduino board from a windows machine via a bluetooth connection.

Hardware Needed:

- Arduino Uno Board <http://arduino.cc/en/Main/arduinoBoardUno>
- Bluetooth Maste Silver <https://www.sparkfun.com/products/10393>
- USB Bluetooth Adapter WRL-00150 <https://www.sparkfun.com/products/150>
- Wires
- Led
- 1 K Ω Resistor

Getting Started:

Configuring the USB bluetooth adapter on the computer is the first step. This is done by installing the necessary drivers, which are usually found from the CD that comes with bluetooth bundle, or alternatively can be downloaded and installed from the manufacturer's website. Furthermore, some bluetooth modules have their drivers auto install from the USB itself. Refer to the manufacturer documentation and install accordingly. The bluetooth used in this guide was installed via the CD that shipped with the product.

Once the drivers have been installed, we are ready to use the bluetooth module, where it is running through the USB serial port.

Installing Arduino Drivers:

First, download the Arduino IDE software from <http://arduino.cc/en/main/software>. The Arduino drivers must be installed to facilitate proper communication between the board and the windows computer, which will allow us to program the Arduino later. Plug in the board to a USB port and wait for windows to begin the driver installation process. This process will fail, and the Arduino will need to be manually configured.

On your computer, navigate to Control Panel, System and Security. Click on System, and open the Device Manager. Under Ports (COM), right click on the port named Arduino UNO (COMxx), and choose "Update Driver Software". Next, choose "Browse my computer for Driver software", navigate to the Uno driver file named "ArduinoUno.inf" which is located in the Drivers folder of the Arduino IDE software that was previously downloaded. Windows will finish up the driver installation from there.

Setting up the Arduino IDE:

Double click the arduino application. This will open a Sketch editor. A sketch is the name that Arduino uses for a program. It can be thought of as the unit of code that is uploaded to and run on an Arduino board. Select the proper board by navigating to tools, board, Arduino Uno, as shown in figure 1.

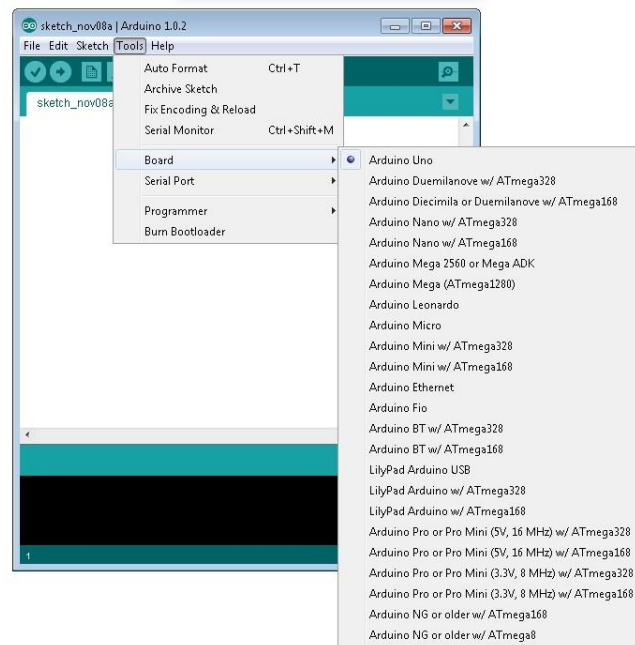


Figure 1 - Arduino IDE

Now we must figure out which serial port the Arduino is using via its USB connection. On the Arduino IDE, navigate to Tools, Serial Port. To find out which serial port the Arduino is connected to, disconnect your Arduino board and re-open the menu; the entry that disappears should be the correct serial port. Reconnect the board and select that serial port. We will assume it is COM4.

Writing the Code:

Now its time to write the code that will allow the Arduino to perform our intended function. First, let us initialize the appropriate variables:

```
char val; // variable to receive
data from the serial port

int ledpin = 8; // LED connected to pin
8 (on-board LED)
```

As the code shows, we will use pin 8 on the board for the LED. If you decide to connect the LED to another pin, the code must be changed accordingly to reflect that.

Next, we will write the setup function. The setup function is the start of a sketch, used to initialize variables, pin modes, start using libraries, and other functionalities. It will be run once after each Arduino power up or reset.

Let us use this to set up the LED pin (number 8 in this case) as an output, and set up the serial communication to the bluetooth by giving it the serial baud rate for the bluetooth USB module. THE WRL-00150 module uses a baud-rate of 115200. If you have chosen a different module, refer to the spec sheet and edit the code accordingly. Since this function is returning no information, the void keyword precedes it.

```
void setup()
{
  // pin 8 (on-board LED) as OUTPUT
  pinMode(ledpin, OUTPUT);
  // start serial communication at 115200bps
  Serial.begin(115200);
}
```

Now we must add a loop() function. This function loops consecutively, allowing your program to respond to input changes on the Arduino. This is where any “control” code is written. Let us set up the loop in to allow Arduino board to receive input signals from the bluetooth and react accordingly.

```
void loop() {
  // if data is available to read
  if( Serial.available() )
  {
    // read it and store it in 'val'
    val = Serial.read();
  }
  if( val == '1' ) // if '1' was received
  {
    digitalWrite(ledpin, HIGH); // turn LED on
  }
  if (val == '0') // if '0' was received
```

```

{
    digitalWrite(ledpin, LOW);    // Turn LED off
}

delay(100);
}

```

Note that the arduino still thinks of the bluetooth as a serial communication, which explains the use of the keyword “serial” in the code.

Uploading any program to the Arduino must be done using a serial connection, and cannot be done via bluetooth. Upload the code onto the board by clicking the Upload button as shown in figure 2. Reset the board and remove the serial cable.



Figure 2 - The upload button

Wiring:

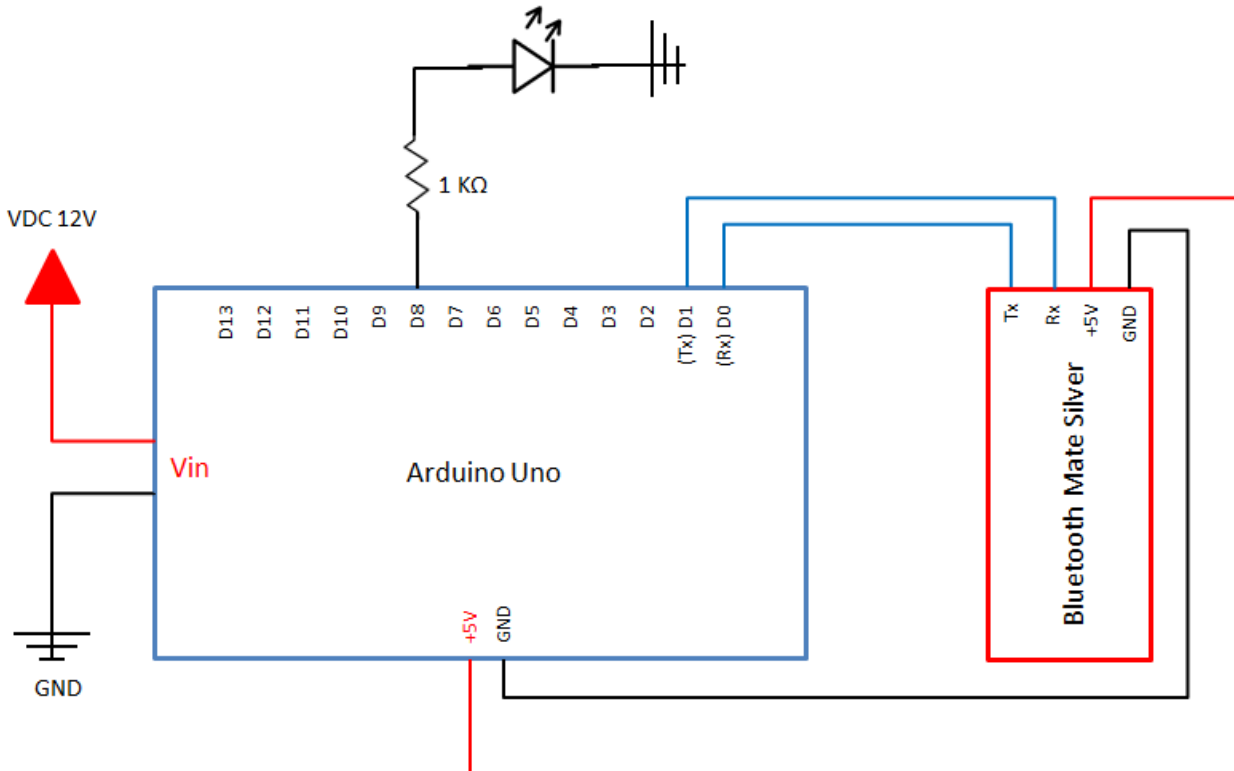


Figure 3 - Wiring

Connect the hardware as shown in figure 3. The LED can be connected to any PIN except to pins 0 and 1, which are the Rx and Tx pins on the Arduino respectively. As shown in the figure x, pin 8 has been used (with the other end of the LED connected to GND). Connect the Tx pin on the bluetooth to the Rx of the arduino (pin 0), and the Rx pin on the bluetooth to the Tx on the Arduino (pin 1). Connect the GND pin to gnd on arduino, and vcc to the 5v. Plug the AC adaptor into the arduino board. Make sure the small green LED on the Bluetooth Mate Silver is blinking. If not, check your wiring and correct accordingly

Setting Up the Bluetooth/Arduino Interface:

Now we need to set up the interface between the Bluetooth Mate Silver connected to the Arduino and the USB Bluetooth module on the Windows computer.

Double click the bluetooth icon on the taskbar and select add a device. It should find the Bluetooth Mate Silver (Although the naming may be different depending in the Bluetooth USB module you are using). Select it to connect, and enter the passkey which should be '1234' by default. If you have used another USB Bluetooth module, consult the product documentation.

Since the USB Bluetooth module is connected to a COM port, we need to figure out the designated name of the COM port it is connected to. Double click on the Bluetooth icon in the taskbar again and select the device from the devices tab, then click the COM Ports tab and take note of the port number the Bluetooth connection is using.

Sending Input to the Arduino:

The next step is to connect to your arduino board using a terminal. This can be done using the program PuTTY, which can be downloaded from <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>. This is a .exe file that does not require installation.

Open PuTTY, and navigate to session. Click the Serial button under connection line. Fill the serial line with the COM name you noted, and set the speed to 11520 to match the baud rate used in the code. This is shown in figure 4.

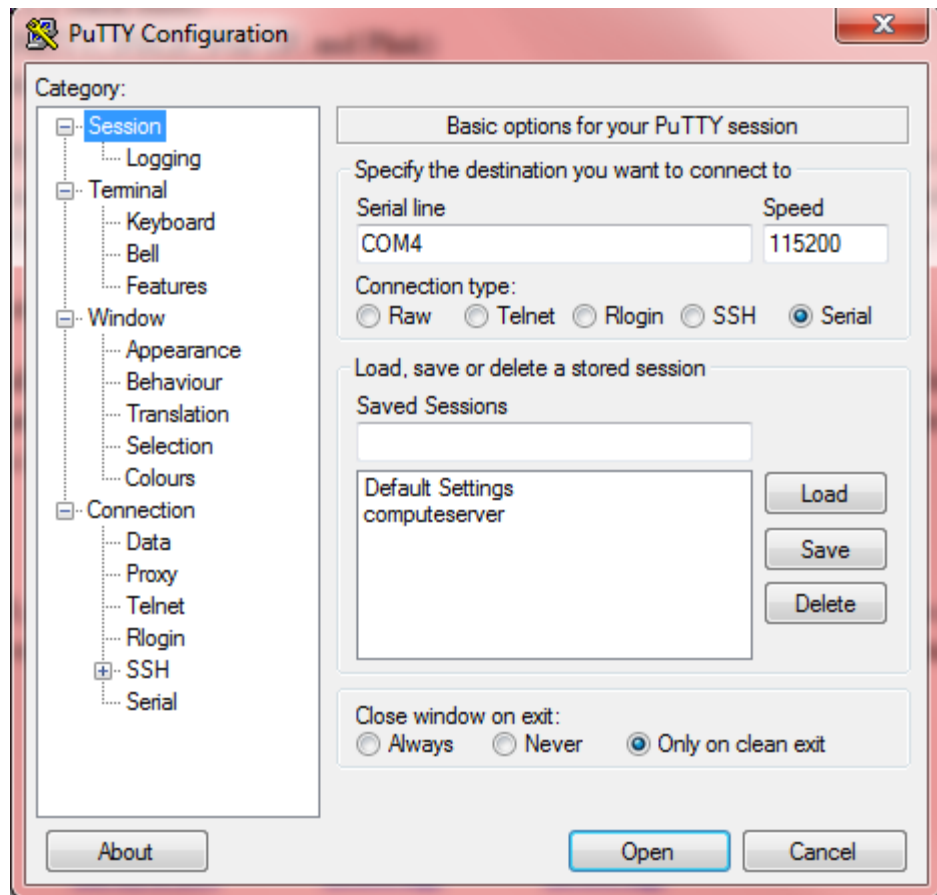


Figure 4 - Setting up PuTTY

Now click open. This will open a terminal window which can be used to send information to the Arduino via the bluetooth connection established through the COM port. Pressing 1 and hitting enter should light up the LED, will pressing 0 and hitting enter should turn the LED off.

References:

<http://arduino.cc/en/Reference/>

Kurose, James F., and Keith W. Ross. *Computer Networking: A Top-Down Approach*. Boston: Addison-Wesley, 2010. Print.