

## LED Control using Arduino Bluetooth and Android. (Part 1)

This instructable is a quick tutorial explaining how to control one LED using Android's Bluetooth.

You will need:

- Arduino UNO;
- LM35 (Or other sensor);
- Resistors 300Ω (3x)
- Jumpers;
- Bluetooth Module (HC-06/other);
- Android Phone;
- SPP Bluetooth Apk (unWired Lite);
- C programming skills

This is the first part of tutorial, You'll learn how to send a command to turn on and turn off a LED.

The second part will be a tutorial of *how to write an android apk to control the LED*.

### HC-06 and Arduino

Bluetooth is a type of wireless communication used to transmit voice and data at high speeds using waves of radio. It's widely used in mobile phones for making calls, headset and share data. This type of communication is a cheap and easy way to control something remotely using arduino.

HC-06 module has 4 pins to be connected to arduino, they are:

- RXD
- TXD
- VCC
- GND

RXD will receive data from arduino; TXD will send data to arduino; VCC is the power supply (3.3V 6.6V) and GND is the ground.

*You gotta pay attention about the RXD level, some module works with 5V, but **this one** works with 3.3V, and arduino TX will send a **5V signal**, then it needs a **voltage divider**.*

Voltage divider with R1 = 300Ω:

$$V_{out} = \frac{R2}{R2 + R1} * V_{in}$$

R2:

$$3.3 = \frac{R2}{R2 + 300} * 5$$

$$3.3 * R2 + 990 = 5 * R2$$

$$990 = 1.7 * R2$$

$$R2 = 582.35 \approx 600\Omega$$

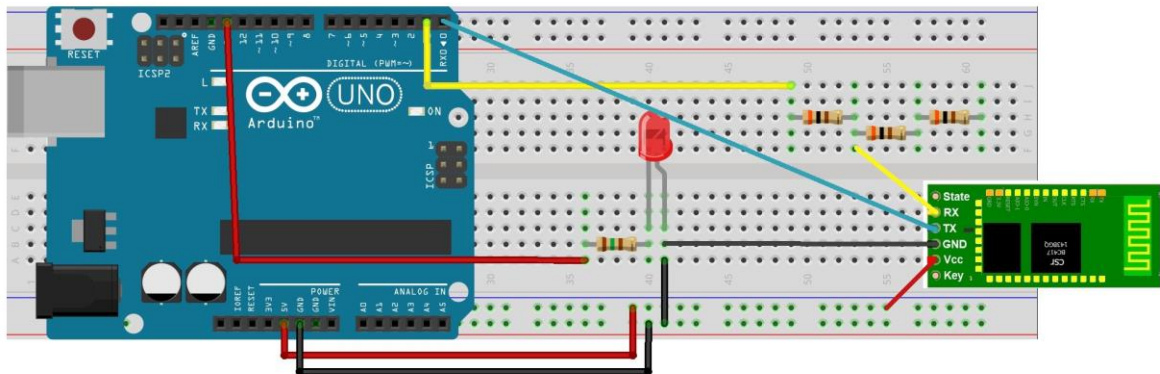
If you have a different resistor:

$$R2 = \frac{3.3 * R1}{1.7} \Omega$$

Setting up:

1<sup>a</sup> Connect the HC-06 module:

Arduino	HC-06
RX	TXD
TX	(Voltage Divider) RXD
+5V	VCC
GND	GND



2<sup>a</sup> C code:

The sketch for this Project is very simple, all you have to do is check the serial port if there's data available.

Using an android phone with a spp bluetooth apk, the command is sent to bluetooth (RX/TX). What happens is the bluetooth module communicates with android's bluetooth using a profile called SPP (Serial Port Profile). It emulates a USB Port connected to arduino and android.

Define all the pins and variables.

```
char command;  
String string;  
#define led 8
```

The default baud rate of HC-06 module is 9600. The void setup code:

```
void setup()  
{  
    Serial.begin(9600);  
    pinMode(led, OUTPUT);  
}
```

Void loop:

```
void loop()  
{  
    if (Serial.available() > 0)  
    {string = "";}  
  
    while(Serial.available() > 0)  
    {  
        command = ((byte)Serial.read());  
        if(command == ':')  
        {  
            break;  
        }  
        else  
        {  
            string += command;  
        }  
        delay(1);  
    }  
    if(string == "LO")  
    {  
        LEDOn();  
    }  
    if(string == "LF")  
    {  
        LEDOff();  
    }  
}
```

There are two functions in the code. Actually their names says everything.

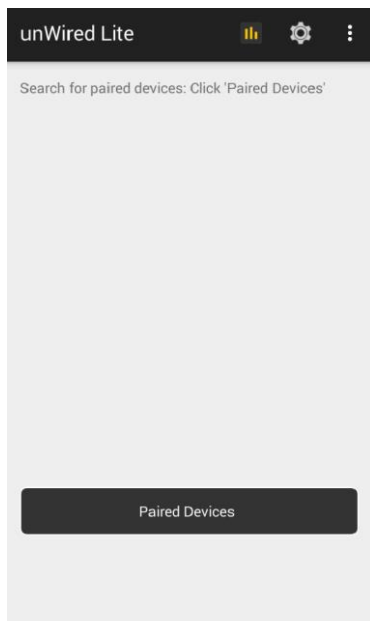
```
void LEDOn()
{
    digitalWrite(led, HIGH);
}

void LEDOff()
{
    digitalWrite(led, LOW);
    delay(500);
}
```

### ***unWired Lite and unWired Pro***

unWired Lite is an spp application that connects to arduino's bluetooth modules. You try this one or other you've already known.

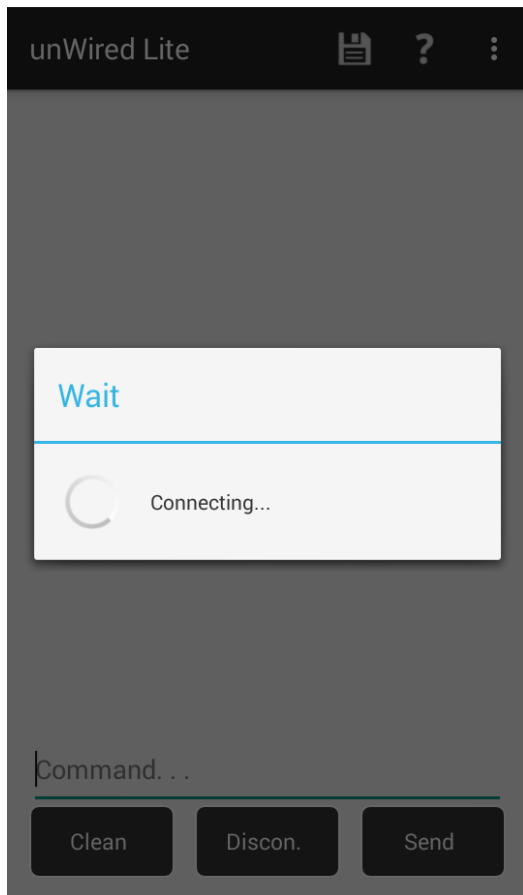
unWired Lite (Link)



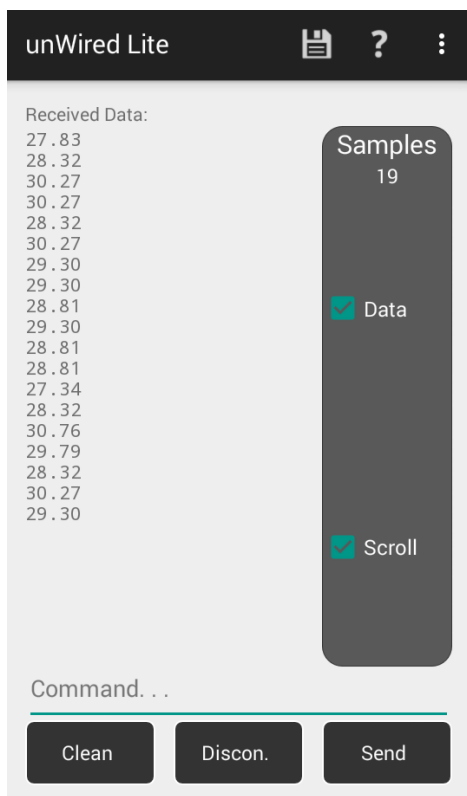
*Before running the apk, pair the bluetooth module.*

The first screen of the apk is shown above, click at 'Paired Devices'. It'll list all the paired devices, then you selected the bluetooth module you've paired before.

The next screen is where you write the commands to send to bluetooth. The commands are LO for Led On, and LF for Led Off.



*The screen connecting to bluetooth module*



*Done! Now you know how to set up bluetooth modules to arduino and send commands to it.*

*You can also acquire data with the spp bluetooth apk, all you have to do is change the functions. unWired Pro is also available and it allows you to plot realtime data.*

[unWired Lite](#)

[unWired Pro](#)

*The next tutorial is:*

*How to write an android apk to turn on/turn off the LED.*