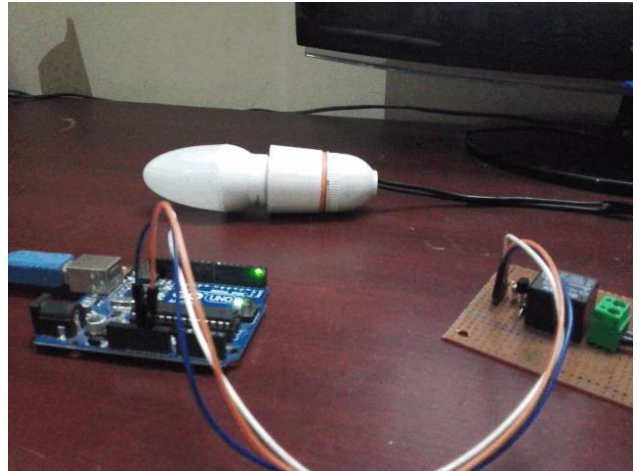

Project 16 : Controlling AC light or appliance with arduino

Overview

Today I will show you how to control a AC light/appliance with the same Blink sketch
Controlling AC light or appliance with arduino is simple as blinking a LED using arduino.
All you need to is take proper care while doing this project because here we are dealing with **AC** current (230V is more than enough to kill you).So **beware**

Things you need

- 5V Relay
- Arduino
- BC547 transistor
- 1n4007 Diode
- AC Bulb
- Bulb Socket
- screw terminal connector



Before we Build

What is a relay?

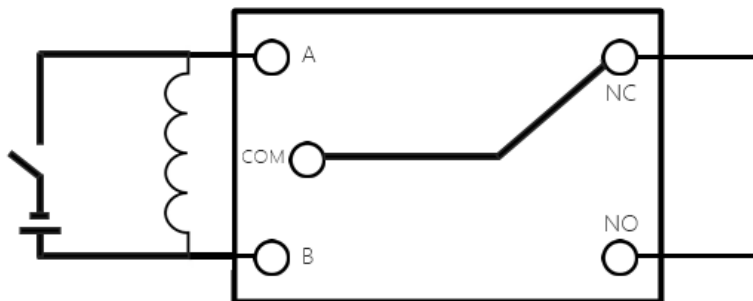
Relays are switches that open and close circuits electromechanically. They control one electrical circuit by opening and closing contacts in another circuit

Why BC547?

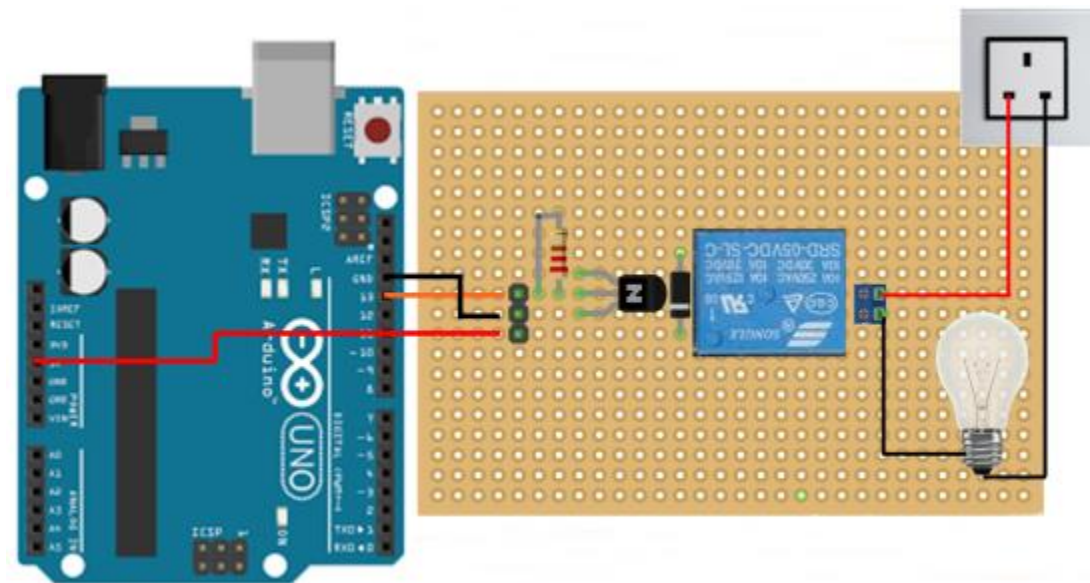
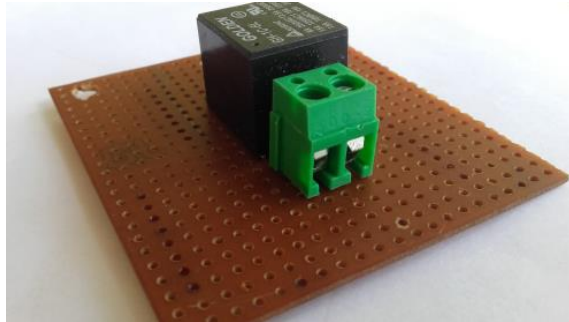
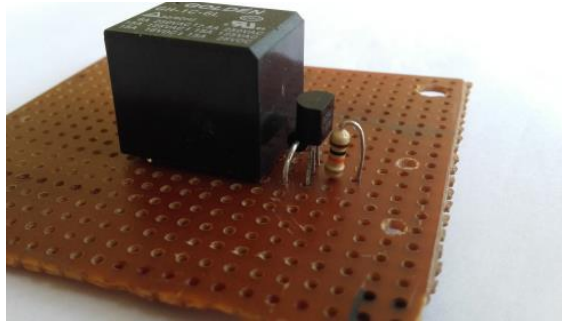
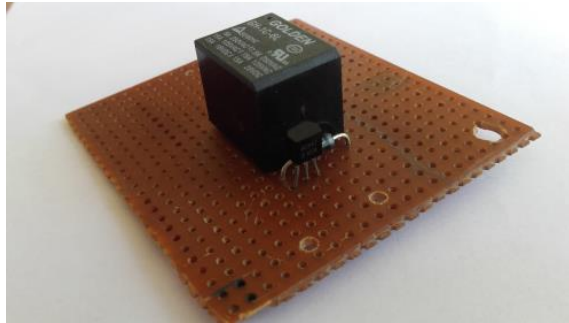
5V Relay needs about 200mA current at 5V. But the digital pins of arduino provides only 20 mA current. So arduino can not drive the relay directly. So as amplifier we need to use a transistor to drive the relay

Why Diode?

The relay coil cannot change its current instantly, diode provides a path for the current when the coil is switched off. Otherwise, a voltage spike will occur causing arcing on switch contacts or possibly destroying switching transistors



Lets start building



Upload Sketch (existing code – Blinks LED) to arduino

There is not need to download any code you have the code within the software just upload the Blink Sketch and watch the AC light blinking as a LED blinks

Project 16 : Connecting Arduino from Android using Bluetooth

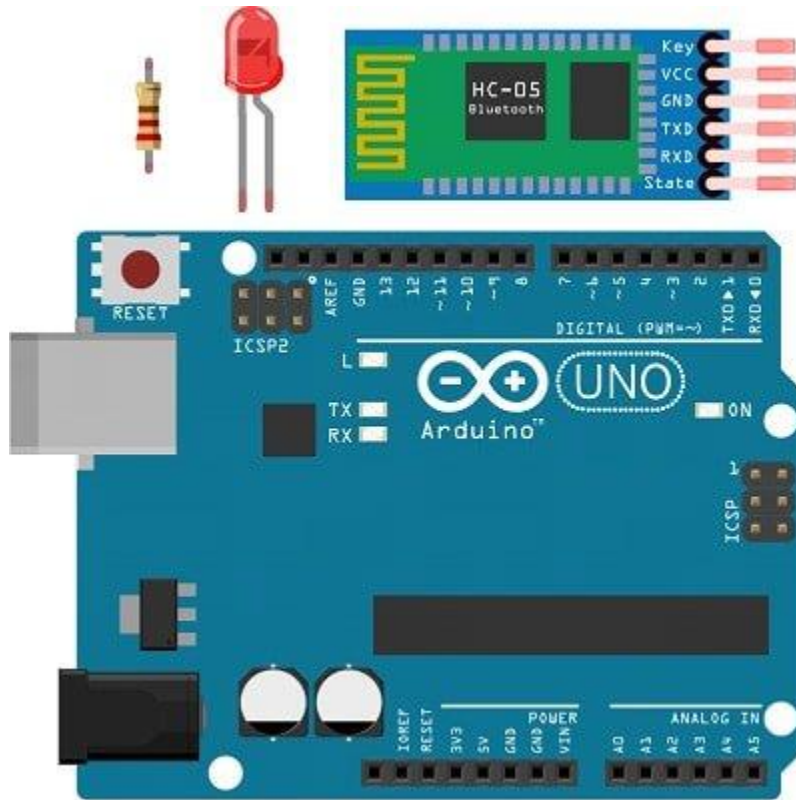
Overview

Here is a simple tutorial for interfacing an Android smartphone with Arduino via Bluetooth.

Tools and Parts Needed

In order to complete the projects in this tutorial, you'll need to make sure you have the following items.

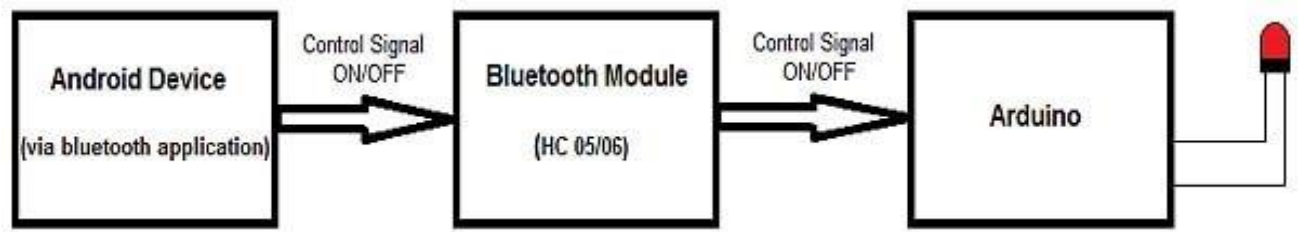
- Bluetooth Module HC 05/06
- Arduino
- LED
- 220Ω Resistor
- Android device
- Arduino IDE
- Android Studio (Not required - there are free apps provide the android application)



NOTE : Android Studio isn't really required here, since I will provide you with the Android application that I made. You can install the given **.apk file** to use the application.

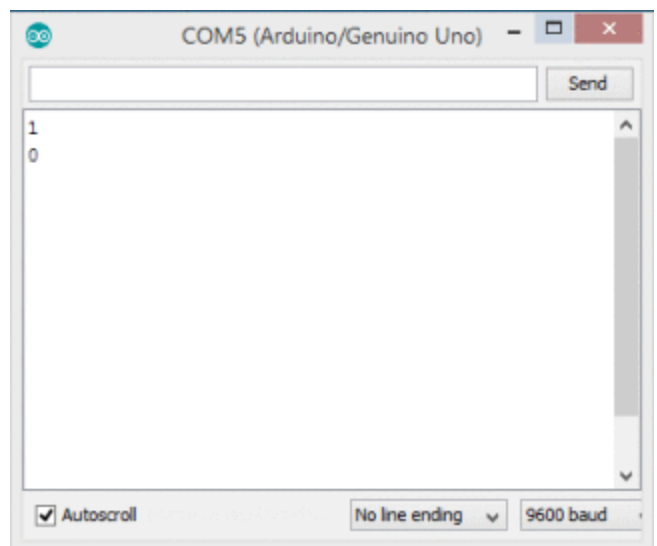
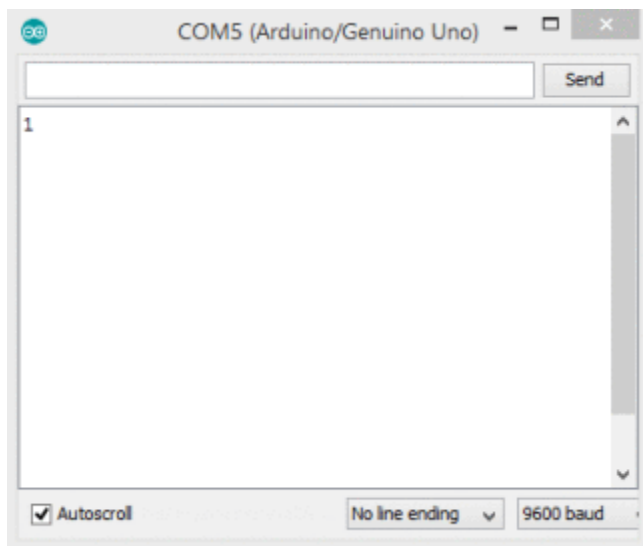
How Does It Work?

There are three main parts to this project. An Android smartphone, a Bluetooth transceiver, and an Arduino.



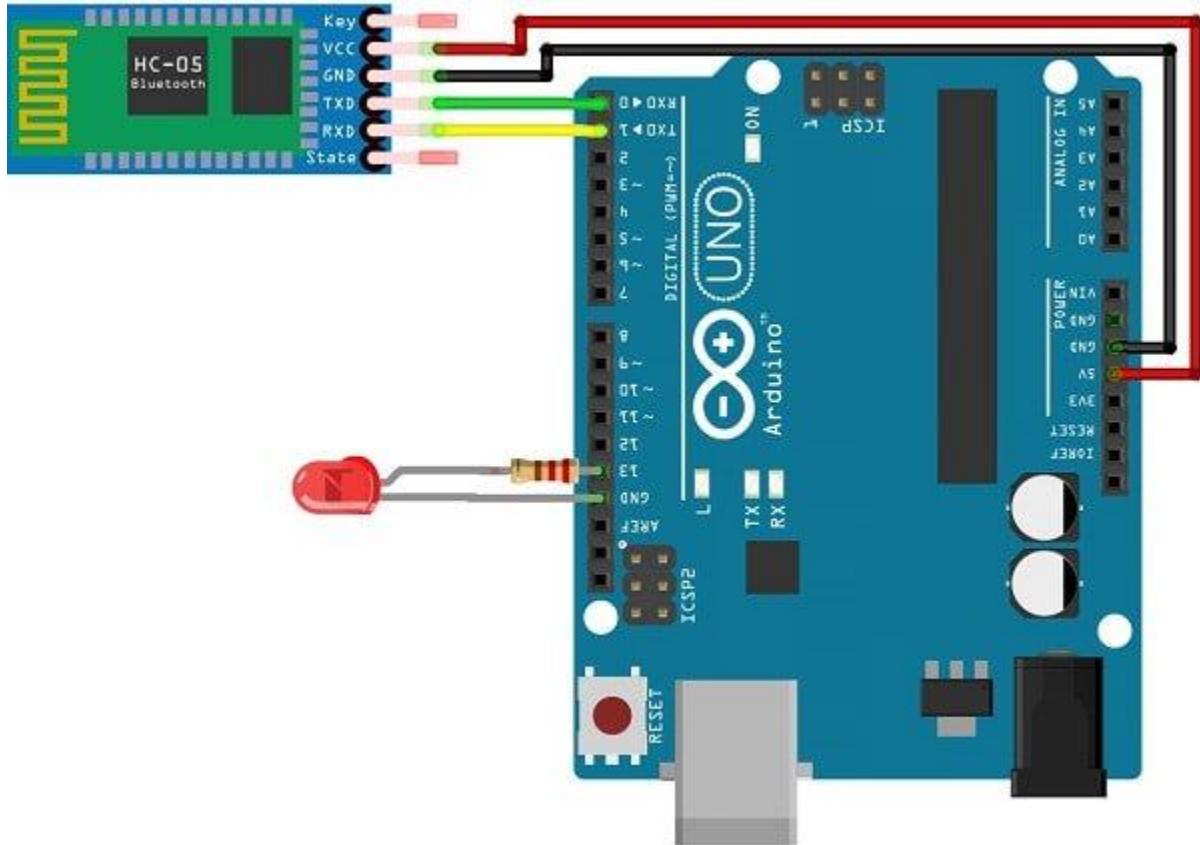
HC 05/06 works on serial communication. The Android app is designed to send serial data to the Arduino Bluetooth module when a button is pressed on the app. The Arduino Bluetooth module at the other end receives the data and sends it to the Arduino through the TX pin of the Bluetooth module (connected to RX pin of Arduino). The code uploaded to the Arduino checks the received data and compares it. If the received data is 1, the LED turns ON. The LED turns OFF when the received data is 0. You can open the serial monitor and watch the received data while connecting.

Open the **serial monitor** and watch the received data:



Connecting the Arduino Bluetooth Hardware

Connect an LED positive to pin 13 of the Arduino through a resistance (valued between 220Ω – $1K\Omega$). Connect its negative to GND, and you're done with the circuit!



This circuit is simple and small. There are only four connections to be made between the Arduino and Bluetooth module!

Arduino Pins		Bluetooth Pins
RX (Pin 0)	————>	TX
TX (Pin 1)	————>	RX
5V	————>	VCC
GND	————>	GND

Uploading the Sketch to Arduino

```
char data = 0;           //Variable for storing received data

void setup()
{
  Serial.begin(9600);     //Sets the data rate in bits per second (baud)
                          //for serial data transmission
  pinMode(13, OUTPUT);   //Sets digital pin 13 as output pin
}

void loop()
{
  if(Serial.available() > 0) // Send data only when you receive data:
  {
    data = Serial.read();    //Read the incoming data and store it into
                             //variable data
    Serial.print(data);      //Print Value inside data in Serial monitor
    Serial.print("\n");      //New line

    if(data == '1')          //Checks whether value of data is equal to 1
      digitalWrite(13, HIGH); //If value is 1 then LED turns ON
    else if(data == '0')     //Checks whether value of data is equal to 0
      digitalWrite(13, LOW);  //If value is 0 then LED turns OFF
  }
}
```

Installing the Android Application

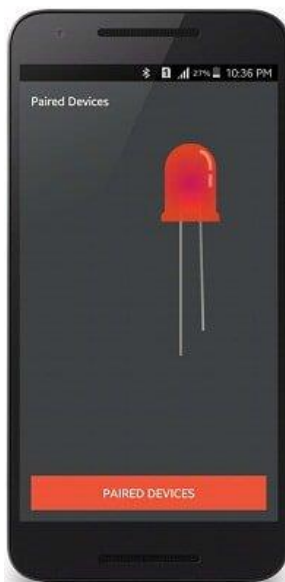
In this tutorial, we will not cover **Android app development**. But soon we will be learning in the next tutorial. You can download the Android application from [here](#) (Amazon - FREE) or [here](#) (Github)

STEPS AS FOLLOW :

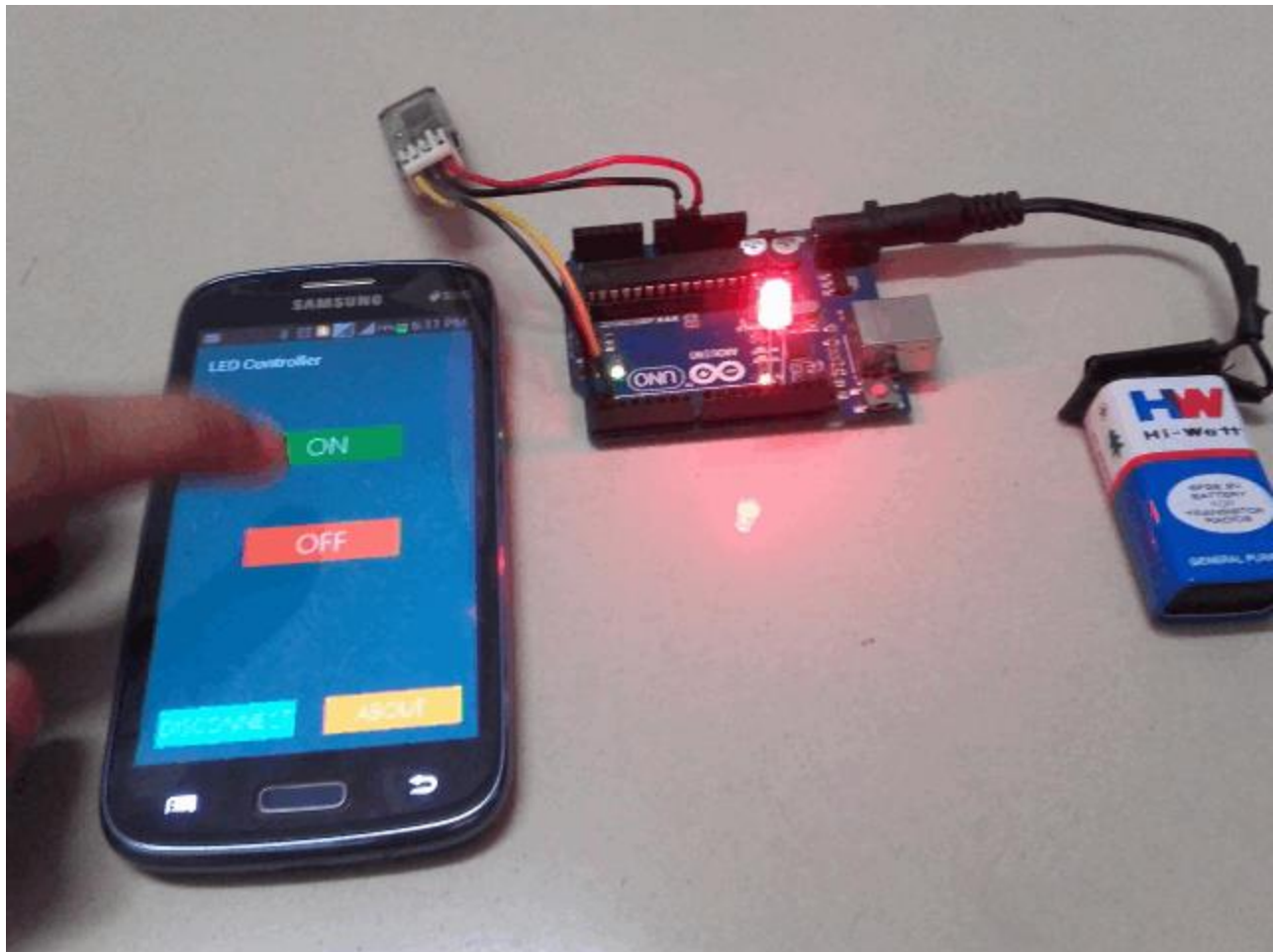
- Download the application from Amazon App Store or Github.
- Pair your device with the HC 05/06 Bluetooth module:
 - a. Turn ON the HC 05/06 Bluetooth module by powering the Arduino.
 - b. Scan your smartphone for available devices.
 - c. Pair your smartphone to the HC 05/06 by entering default password 1234 OR 0000.
- Install the LED application on your Android device.
- Open the application.



- Press "paired devices".
- Select your Bluetooth module from the list (HC-05/06)



- After connecting successfully, press the ON button to turn the LED on and the OFF button to turn the LED off.
- Disconnect the button to disconnect the Bluetooth module.



Here's how it works!

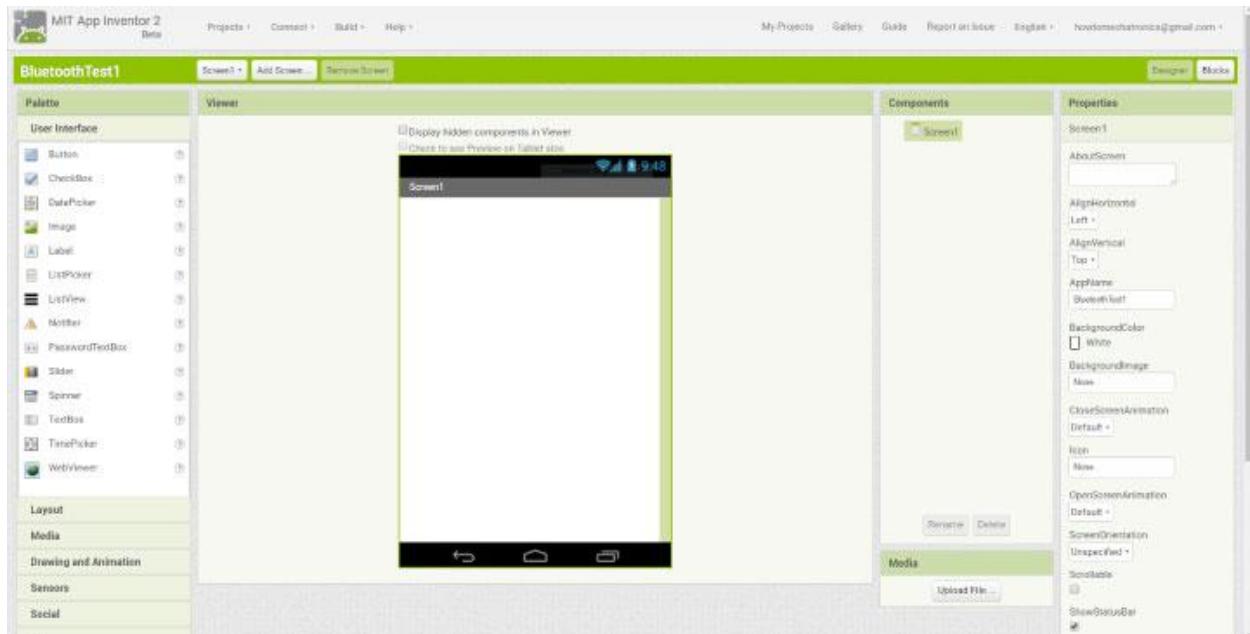
This is just a basic tutorial on interfacing a Bluetooth module with Arduino. This project can be taken to a higher level, like home automation using a smartphone, smartphone-controlled robots, and much more!

Android App for your Arduino Project using MIT App Inventor

In our above tutorial we have already learned how to make the Bluetooth communication between the Arduino Board and the Smartphone using the HC-05 Bluetooth module and explained the Arduino code needed for the first example.

In this tutorial we will learn how to build custom Android applications for controlling Arduino using the **MIT App Inventor online application**. For this tutorial we have two STEPS. 1) Wiring up a simple LED and 2) Controlling a **Stepper Motor** using the smartphone.

From the MIT App Inventor website we need to log in into the online building application by clicking the "Create apps!" button. In order to log in we need to have a Gmail account. Once we are logged in now we can create our first project. Here's how the design window looks and now we can start building our application.

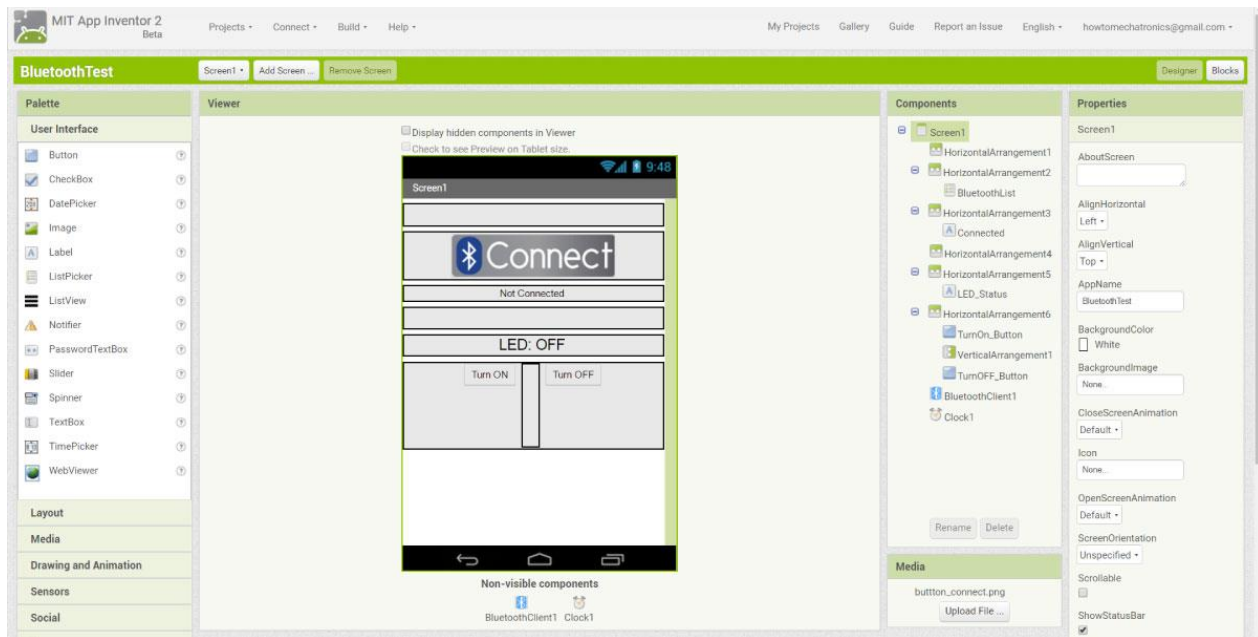


<http://appinventor.mit.edu/explore/>

But before do that, we can connect our smartphone to this project so that we can see how the app is taking shape directly on our smartphone in real time. In order to do that first we have to download the MIT AI2 Companion app from the Play Store and install it on our smartphone. Then from the Connect menu from the online editor we will select AI Companion and a barcode will appear which we just need to scan it or insert the code into the smartphone app and the connection between the online editor and the smartphone app will be established.



So now for example, if we insert a button in the screen of the online editor, the button will appear in real time on the smartphone as well. Similar to this, if you don't want to use your smartphone while building the app, you can install the Android Emulator on your computer and use in the same way. You can find more details how to set up the Emulator on their website.



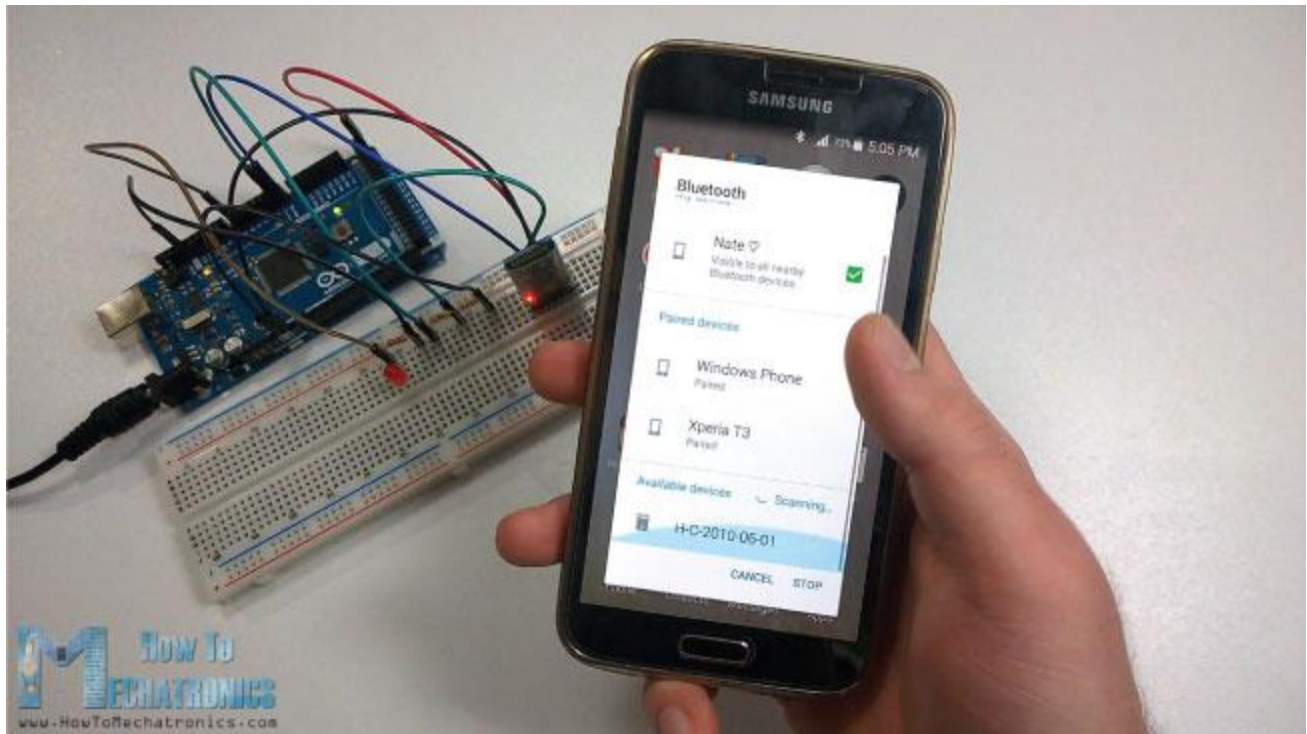
Here's a download file of the above MIT App Inventor project:



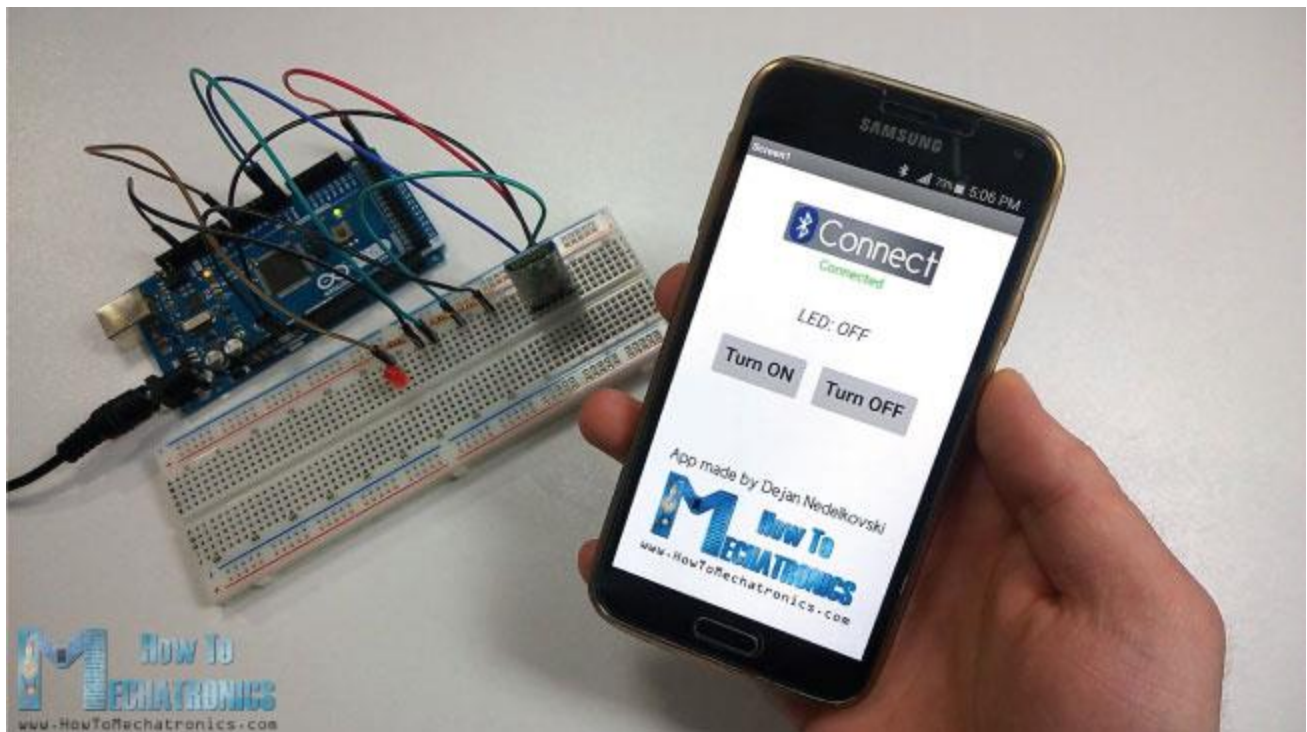
BluetoothTest.aia | [Download](#)

Connecting the Smartphone to the HC-05 Bluetooth Module and the Arduino

Now we are ready to connect the smartphone to the Bluetooth module and the Arduino. What we need to do here is to activate the Bluetooth and the smartphone will find the HC-05 Bluetooth module.



Then we need to pair the devices and the default password of the HC-05 module is 1234. After we have paired the devices we need an application for controlling the Arduino. There are many application in the Play Store for this purpose which will work with the Arduino code that we wrote. However, I made my own custom application for this tutorial using the MIT App Inventor online application. This is a great and easy to use application for building Android application and in my next tutorial you can find a detailed step by step guide how to build your own custom Android application for your Arduino Project.



You can download the app that I made for this example here:

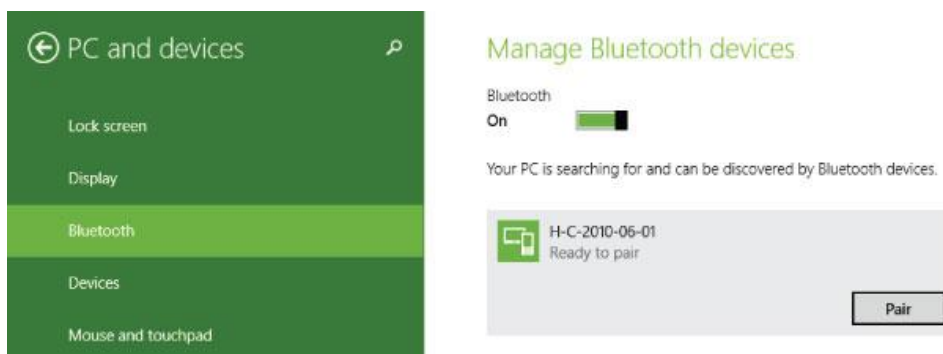


Arduino Bluetooth Tutorial Example Android App | [Download](#)

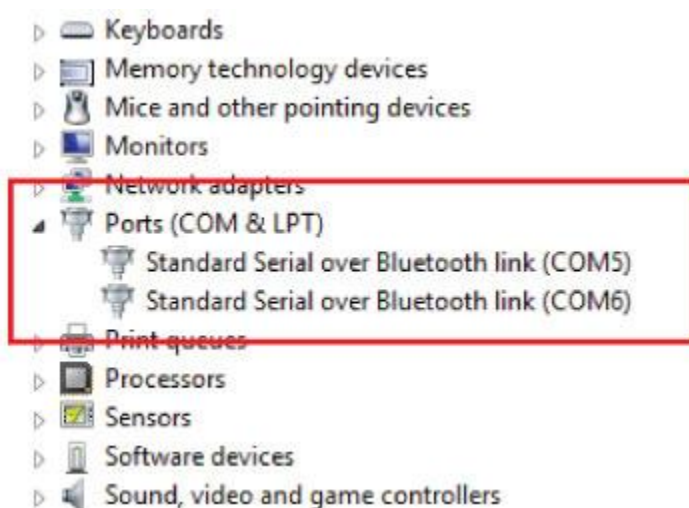
With the connect button we will connect the smartphone to the Bluetooth module and the status text below the button will tell us whether we have successfully connected. Using the “Turn ON” and “Turn OFF” buttons we can turn on and off the LED. The text above the buttons is the one that the Arduino is sending back to the smartphone when a particular button is pressed.

Controlling Arduino Using a Laptop or PC via Bluetooth Communication

Let's see how we can control the Arduino via Bluetooth using a Laptop or a PC. So, first we need to pair our laptop to the HC-05 Bluetooth module and we can do that from the Laptop Bluetooth Settings. The laptop will discover the HC-05 module and using the '1234' password we will pair the devices.



Once we will pair the devices in the Laptop Device Manager, under Ports (COM & LPT), two new entities will appear named “Standard Serial over Bluetooth link”. From here we can see the COM Port number of the serial port through which the devices will communicate.



We will stick with the same example as previously, turning on and off a LED and sending back a string to the laptop, so we will use the same Arduino code as previously described.

```
1. import processing.serial.*;
2. Serial myPort;
3. String ledStatus="LED: OFF";
4.
5. void setup(){
6.   size(450, 500);
7.   myPort = new Serial(this, "COM5", 38400); // Starts the serial communication
8.   myPort.bufferUntil('\n'); // Defines up to which character the data from the serial port will
   be read. The character '\n' or 'New Line'
9. }
10. void serialEvent (Serial myPort){ // Checks for available data in the Serial Port
11.   ledStatus = myPort.readStringUntil('\n'); //Reads the data sent from the Arduino (the String
   "LED: OFF/ON) and it puts into the "ledStatus" variable
12. }
13.
14. void draw(){
15.   background(237, 240, 241);
16.   fill(20, 160, 133); // Green Color
17.   stroke(33);
18.   strokeWeight(1);
19.   rect(50, 100, 150, 50, 10); // Turn ON Button
20.   rect(250, 100, 150, 50, 10); // Turn OFF Button
21.   fill(255);
22.
23.   textSize(32);
24.   text("Turn ON",60, 135);
25.   text("Turn OFF", 255, 135);
26.   textSize(24);
27.   fill(33);
28.   text("Status:", 180, 200);
29.   textSize(30);
30.   textSize(16);
31.   text("Program made by Dejan Nedelkovski,\n    www.HowToMechatronics.com", 80, 320);
32.
33.   text(ledStatus, 155, 240); // Prints the string coming from the Arduino
34.
35.   // If the button "Turn ON" is pressed
36.   if(mousePressed && mouseX>50 && mouseX<200 && mouseY>100 && mouseY<150){
37.     myPort.write('1'); // Sends the character '1' and that will turn on the LED
38.     // Highlights the buttons in red color when pressed
39.     stroke(255,0,0);
40.     strokeWeight(2);
41.     noFill();
42.     rect(50, 100, 150, 50, 10);
43.   }
44.   // If the button "Turn OFF" is pressed
45.   if(mousePressed && mouseX>250 && mouseX<400 && mouseY>100 && mouseY<150){
46.     myPort.write('0'); // Sends the character '0' and that will turn on the LED
47.     stroke(255,0,0);
48.     strokeWeight(2);
49.     noFill();
50.     rect(250, 100, 150, 50, 10);
51.   }
52. }
```