In this article I will show how to control an electrical appliance remotely from your **Android** device using **bluetooth**.

**The hardwares used in this project are:**

1. Android Mobile
2. Bluetooth Module
3. Arduino UNO R3
4. 5V Relay Module

**Softwares used:**

1. Eclipse IDE configured for android development
2. Arduino compiler

**Bluetooth Module Details:**

JY-MCU Arduino Bluetooth Wireless Serial Port Module
Default password: 1234
Baud rate: 38400
Dimensions: 1.73 in x 0.63 in x 0.28 in (4.4 cm x 1.6 cm x 0.7 cm)
PINOUT
PIN DESCRIPTION
1 KEY
2 VCC
3 GND
4 TXD
5 RXD

**Adruino Code:**

```
/**********************
 Bluetooth switch arduino
**********************/
int state = 0;
int incomingByte;

void setup() {
  //set pin 12 to output mode
  pinMode(12, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  // see if there's incoming serial data:
  if (Serial.available() &gt; 0) {
    // read the oldest byte in the serial buffer:
    incomingByte = Serial.read();
    // depending on the incoming byte do the needful action
    if (incomingByte == '1') {
      digitalWrite(12, HIGH);
      delay(500);
      Serial.println("ON");
      state=1;
    }
    else if (incomingByte =='2') {
      digitalWrite(12, LOW);
      delay(500);
      Serial.println("OFF");
      state = 0;
    }
  }

  if(state)
    Serial.println("ON");
  else
    Serial.println("OFF");
  delay(250);
}
```

Here is the android user interface, it just works anyway..

For Android part I changed the code from here a little bit:

**This is how the code works basically:**
1. On click of the Open button it calls findBT() and openBT() methods
2. findBT() method searches for all Paired bluetooth devices (in this case it tries to find a bluetooth device named ‘linvor’ and saves it, in your case this name can be different and hence need to replace in the code)
3. openBT() using UUID (universally unique identifier) creates a bluetooth socket connection on a bluetooth device by calling connect(). Then it opens input and output streams on this socket connection. And finally calls beginListenForData().
4. beginListenForData() opens a thread where it listens for data and if there is data it accumulates and shows them in a label.
5. At the same time there are ON and OFF button in the interface which sends data via output stream, thread is still running and listening for data.
6. Arduino received the data by his bluetooth companion and acts on that data using his logic also replies back to the android device.
7. In the Arduino site similar to Android there is Serial.read() which read the serial data and returns as int which is then matched for setting output pin to LOW or HIGH state. And there is Serial.println() for sending data back to Android for confirming the Status.

> When pairing with the Bluetooth device (linvor) from Android it might ask a password which is generally ‘1234’.

**Android Code:**

```
package Android.Arduino.Bluetooth;

import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.util.Log;
import android.view.View;
import android.widget.TextView;
import android.widget.EditText;
import android.widget.Button;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.Set;
import java.util.UUID;

public class BluetoothTest extends Activity
{
    TextView myLabel;
    EditText myTextbox;
    BluetoothAdapter mBluetoothAdapter;
    BluetoothSocket mmSocket;
    BluetoothDevice mmDevice;
    OutputStream mmOutputStream;
    InputStream mmInputStream;
    Thread workerThread;
    byte[] readBuffer;
    int readBufferPosition;
    int counter;
    volatile boolean stopWorker;

    @Override
    public void onCreate(Bundle savedInstanceState)
    {
```

```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        Button openButton =
(Button)findViewById(R.id.open);
       // Button sendButton =
(Button)findViewById(R.id.send);
        Button closeButton =
(Button)findViewById(R.id.close);
        Button onButton =
(Button)findViewById(R.id.onButton);
        Button offButton =
(Button)findViewById(R.id.offButton);

        myLabel = (TextView)findViewById(R.id.label);
       // myTextbox = (EditText)findViewById(R.id.entry);

        //Open Button
        openButton.setOnClickListener(new
View.OnClickListener()
        {
            public void onClick(View v)
            {
                try
                {
                    findBT();
                    openBT();
                }
                catch (IOException ex) { }
            }
        });

        //Send Button
        /*sendButton.setOnClickListener(new
View.OnClickListener()
        {
            public void onClick(View v)
            {
```

```
            try
            {
                sendData();
            }
            catch (IOException ex) { }
        }
    });*/

    //ON SWITCH
    onButton.setOnClickListener(new
View.OnClickListener() {

        public void onClick(View v) {
            try {
                onButton();
            } catch (Exception e) {
                // TODO: handle exception
            }

        }
    });

   //OFF SWITCH
    offButton.setOnClickListener(new
View.OnClickListener() {

        public void onClick(View v) {
            try {
                offButton();
            } catch (Exception e) {
                // TODO: handle exception
            }

        }
    });

    //Close button
    closeButton.setOnClickListener(new
```

```
View.OnClickListener()
        {
            public void onClick(View v)
            {
                try
                {
                    closeBT();
                }
                catch (IOException ex) { }
            }
        });
    }

    void findBT()
    {
        mBluetoothAdapter =
BluetoothAdapter.getDefaultAdapter();
        if(mBluetoothAdapter == null)
        {
            myLabel.setText("No bluetooth adapter available");
        }

        if(!mBluetoothAdapter.isEnabled())
        {
            Intent enableBluetooth = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(enableBluetooth, 0);
        }

        Set pairedDevices =
mBluetoothAdapter.getBondedDevices();
        if(pairedDevices.size() &gt; 0)
        {
            for(BluetoothDevice device : pairedDevices)
            {
                if(device.getName().equals("linvor")) //this name
have to be replaced with your bluetooth device name
                {
```

```
                mmDevice = device;
                Log.v("ArduinoBT", "findBT found device
named " + mmDevice.getName());
                Log.v("ArduinoBT", "device address is " +
mmDevice.getAddress());
                break;
            }
        }
    }
    myLabel.setText("Bluetooth Device Found");
}

void openBT() throws IOException
{
    UUID uuid =
UUID.fromString("00001101-0000-1000-8000-00805f9b34f
b"); //Standard SerialPortService ID
    mmSocket =
mmDevice.createRfcommSocketToServiceRecord(uuid);

    mmSocket.connect();
    mmOutputStream = mmSocket.getOutputStream();
    mmInputStream = mmSocket.getInputStream();

    beginListenForData();

    myLabel.setText("Bluetooth Opened");
}

void beginListenForData()
{
    final Handler handler = new Handler();
    final byte delimiter = 10; //This is the ASCII code for
a newline character

    stopWorker = false;
    readBufferPosition = 0;
    readBuffer = new byte[1024];
```

```
        workerThread = new Thread(new Runnable()
        {
            public void run()
            {
                while(!Thread.currentThread().isInterrupted()
&amp;amp;&amp;amp; !stopWorker)
                {
                    try
                    {
                        int bytesAvailable =
mmInputStream.available();
                        if(bytesAvailable &gt; 0)
                        {
                            byte[] packetBytes = new
byte[bytesAvailable];
                            mmInputStream.read(packetBytes);
                            for(int i=0;i&lt;bytesAvailable;i++)
                            {
                                byte b = packetBytes[i];
                                if(b == delimiter)
                                {
                                    byte[] encodedBytes = new
byte[readBufferPosition];
                                    System.arraycopy(readBuffer, 0,
encodedBytes, 0, encodedBytes.length);
                                    final String data = new
String(encodedBytes, &quot;US-ASCII&quot;);
                                    readBufferPosition = 0;

                                    handler.post(new Runnable()
                                    {
                                        public void run()
                                        {
                                            myLabel.setText(data);
                                        }
                                    });
                                }
                                else
```

```
                        {
                            readBuffer[readBufferPosition++] =
b;
                        }
                    }
                }
            }
            catch (IOException ex)
            {
                stopWorker = true;
            }
        }
    }
});

    workerThread.start();
}

void sendData() throws IOException
{
    String msg = myTextbox.getText().toString();
    msg += &quot;&quot;;
    //mmOutputStream.write(msg.getBytes());
    //mmOutputStream.write(msg.getBytes());
    //mmOutputStream.flush();
    //mmOutputStream.close();
    //mmSocket.close();
    myLabel.setText(&quot;Data Sent&quot;+msg);
}

void onButton() throws IOException
{
    mmOutputStream.write(&quot;1&quot;.getBytes());
}

void offButton() throws IOException
{
    mmOutputStream.write(&quot;2&quot;.getBytes());
```

```
    }

    void closeBT() throws IOException
    {
        stopWorker = true;
        mmOutputStream.close();
        mmInputStream.close();
        mmSocket.close();
        myLabel.setText(&quot;Bluetooth Closed&quot;);
    }
}
```

As you can see this is just scratch, by adding more relays you can control more than one devices upto as much as your arduino board supports. Also fine tuning the code there are lot more possibilities. Let me know about your project in comments and/or any suggestions most welcome.