

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="aula08_acionar_led_botao_bluetooth.android.pdm.aula08_acionarledbotaoblueetooth" >

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".AcionarLedBotaoBluetoothActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

    <uses-permission android:name="android.permission.BLUETOOTH" />
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />

</manifest>
```

res/layout/activity_acionar_led_botao_bluetooth.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="fill_parent" android:layout_height="wrap_content" />

    <Button
        android:id="@+id/btnLedRed" android:layout_width="100dp"
        android:layout_height="wrap_content" android:text="LED VERMELHO" />

    <Button
        android:id="@+id/btnLedYellow" android:layout_width="100dp"
        android:layout_height="wrap_content" android:text="LED AMARELO" />

    <Button
        android:id="@+id/btnLedGreen" android:layout_width="100dp"
        android:layout_height="wrap_content" android:text="LED VERDE" />

    <Button
        android:id="@+id/btnLedWhite" android:layout_width="100dp"
        android:layout_height="wrap_content" android:text="LED BRANCO" />

    <Button
        android:id="@+id/btnDesligarTodos" android:layout_width="100dp"
        android:layout_height="wrap_content" android:text="DESLIGAR TODOS" />

    <Button
        android:id="@+id/btnLigarTodos" android:layout_width="100dp"
        android:layout_height="wrap_content" android:text="LIGAR TODOS" />

</LinearLayout>
```

AcionarLedBotaoBluetoothActivity.java

```
package aula08_acionar_led_botao_bluetooth.android.pdm.aula08_acionarledbotaobluetooth;

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.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import java.io.IOException;
import java.io.OutputStream;
import java.util.UUID;

public class AcionarLedBotaoBluetoothActivity extends Activity {
    private static final String TAG = "LEDOnOff";

    Button btnLedRed, btnLedYellow, btnLedGreen, btnLedWhite, btnDesligarTodos, btnLigarTodos;

    private static final int REQUEST_ENABLE_BT = 1;
    private BluetoothAdapter btAdapter = null;
    private BluetoothSocket btSocket = null;
    private OutputStream outputStream = null;

    // Well known SPP UUID
    private static final UUID MY_UUID =
        UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");

    // Insert your bluetooth devices MAC address
    private static String address = "98:D3:31:40:31:E8";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_acionar_led_botao_bluetooth);
        verificaStatusBluetooth();
        fazerConexoesDoLayoutListeners();
    }

    @Override
    public void onPause() { super.onPause(); }

    private void verificaStatusBluetooth() {
        btAdapter = BluetoothAdapter.getDefaultAdapter();

        // Check for Bluetooth support and then check to make sure it is turned on
        // Emulator doesn't support Bluetooth and will return null
        if(btAdapter == null) {
            errorExit("Fatal Error", "Bluetooth Not supported. Aborting.");
        } else {
            if (!btAdapter.isEnabled()) {
                Intent enableBtIntent = new Intent(btAdapter.ACTION_REQUEST_ENABLE);
                startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);
            }
        }
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        switch(requestCode){
            case REQUEST_ENABLE_BT:
                if(resultCode == Activity.RESULT_OK){
                    Toast.makeText(getApplicationContext(), "Bluetooth foi ativado", Toast.LENGTH_LONG).show();
                } else {
                    Toast.makeText(getApplicationContext(), "Bluetooth nao foi ativado", Toast.LENGTH_LONG).show();
                }
                break;
        }
    }

    public void fazerConexoesDoLayoutListeners() {
        btnLedRed = (Button) findViewById(R.id.btnLedRed);
        btnLedYellow = (Button) findViewById(R.id.btnLedYellow);
        btnLedGreen = (Button) findViewById(R.id.btnLedGreen);
        btnLedWhite = (Button) findViewById(R.id.btnLedWhite);
    }
}
```

```

btnDesligarTodos = (Button) findViewById(R.id.btnDesligarTodos);
btnLigarTodos = (Button) findViewById(R.id.btnLigarTodos);

btnLedRed.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        sendData("r");
        Toast msg = Toast.makeText(getApplicationContext(),
            "LED VERMELHO FOI ACIONADO", Toast.LENGTH_SHORT);
        msg.show();
    }
});

btnLedYellow.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        sendData("y");
        Toast msg = Toast.makeText(getApplicationContext(),
            "LED AMARELO FOI ACIONADO", Toast.LENGTH_SHORT);
        msg.show();
    }
});

btnLedGreen.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        sendData("g");
        Toast msg = Toast.makeText(getApplicationContext(),
            "LED VERDE FOI ACIONADO", Toast.LENGTH_SHORT);
        msg.show();
    }
});

btnLedWhite.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        sendData("w");
        Toast msg = Toast.makeText(getApplicationContext(),
            "LED BRANCO FOI ACIONADO", Toast.LENGTH_SHORT);
        msg.show();
    }
});

btnDesligarTodos.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        sendData("n");
        Toast msg = Toast.makeText(getApplicationContext(),
            "DESLIGAR TODOS OS LEDS", Toast.LENGTH_SHORT);
        msg.show();
    }
});

btnLigarTodos.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        sendData("rygw");
        Toast msg = Toast.makeText(getApplicationContext(),
            "LIGAR TODOS OS LEDS", Toast.LENGTH_SHORT);
        msg.show();
    }
});
}

@Override
public void onResume() {
    super.onResume();

    Log.d(TAG, "...In onResume - Attempting client connect...");

    // Set up a pointer to the remote node using it's address.
    BluetoothDevice device = btAdapter.getRemoteDevice(address);

    try {
        btSocket = device.createRfcommSocketToServiceRecord(MY_UUID);
    } catch (IOException e) {
        errorExit("Fatal Error", "In onResume() and socket create failed: " + e.getMessage() + ".");
    }

    // Discovery is resource intensive. Make sure it isn't going on when you attempt to connect and pass your message.
    btAdapter.cancelDiscovery();

    // Establish the connection. This will block until it connects.
    Log.d(TAG, "...Connecting to Remote...");
    try {
        btSocket.connect();
    } catch (IOException e) {
        try {
            btSocket.close();

```

```

    } catch (IOException e2) {
        errorExit("Fatal Error", "In onResume() and unable to close socket during connection failure" + e2.getMessage() + ".");
    }
}

// Create a data stream so we can talk to server.
try {
    outputStream = btSocket.getOutputStream();
} catch (IOException e) {
    errorExit("Fatal Error", "In onResume() and output stream creation failed:" + e.getMessage() + ".");
}

private void errorExit(String title, String message){
    Toast msg = Toast.makeText(getApplicationContext(),
        title + " - " + message, Toast.LENGTH_SHORT);
    msg.show();
    finish();
}

private void sendData(String message) {
    byte[] msgBuffer = message.getBytes();

    try {
        outputStream.write(msgBuffer);
    } catch (IOException e) {
        String msg = "In onResume() and an exception occurred during write: " + e.getMessage();
        if (address.equals("00:00:00:00:00:00"))
            msg = msg + ".\n\nUpdate your server address to the correct address in the java code";
        msg = msg + ".\n\nCheck that the SPP UUID: " + MY_UUID.toString() + " exists on server.\n\n";

        errorExit("Fatal Error", msg);
    }
}
}

```