



PDM: Projeto para Dispositivos Móveis

Aula 08: Integrando o Android e o Arduino utilizando o Bluetooth

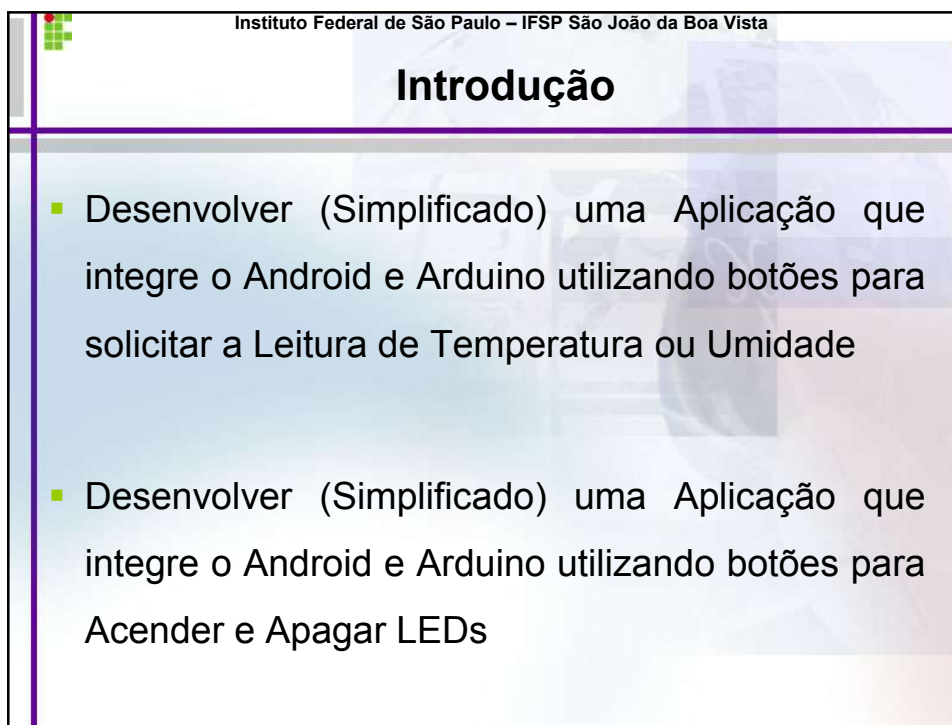
Breno Lisi Romano

<http://sites.google.com/site/blromano>

Instituto Federal de São Paulo – IFSP São João da Boa Vista
Especialização em Desenvolvimento para Dispositivos Móveis



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus São João da Boa Vista



Instituto Federal de São Paulo – IFSP São João da Boa Vista

Introdução

- Desenvolver (Simplificado) uma Aplicação que integre o Android e Arduino utilizando botões para solicitar a Leitura de Temperatura ou Umidade
- Desenvolver (Simplificado) uma Aplicação que integre o Android e Arduino utilizando botões para Acender e Apagar LEDs

Problem Based Learnd - PBL 01

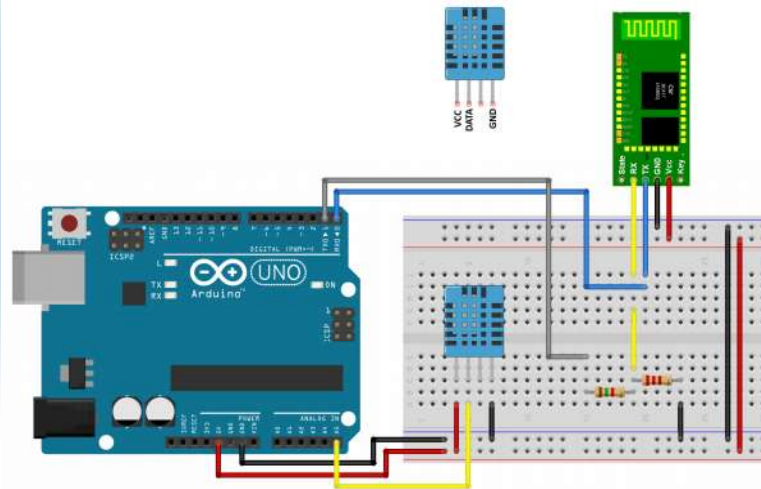
- **Problema:** Desenvolver (Simplificado) uma Aplicação que integre o Android e Arduino por Bluetooth utilizando botões a serem acionados pelo Usuário no Celular com Android para solicitar a Leitura de Temperatura ou Umidade, utilizando-se do Sensor DHT11



Explicação da Lógica de Integração entre Android e Arduino

- Caso os Botões Conectar e Desconectar sejam acionados, deve-se parear ou “desparear” a conexão por Bluetooth entre o Arduino e o Android
- Caso o Botão “Medir Temperatura” seja acionado:
 - Enviar um caractere “t” para o Arduino
 - Arduino solicita a Leitura da Temperatura
 - Arduino retorna a Temperatura Lida para o Android
- Caso o Botão “Medir Umidade” seja acionado:
 - Enviar um caractere “u” para o Arduino
 - Arduino solicita a Leitura da Umidade
 - Arduino retorna a Umidade Lida para o Android

Construindo o Circuito no Protoboard



Código Fonte para Solução Embarcada - Arduino (1)

```

1  #include "DHT.h"
2
3  #define DHTPIN A5 // pino que estamos conectado
4  #define DHTTYPE DHT11 // DHT 11
5  DHT dht(DHTPIN, DHTTYPE);
6
7  void setup()
8  {
9      Serial.begin(9600);
10     dht.begin();
11 }
12
13 void loop()
14 {
15     float umidade;
16     float temperatura;
17
18     if (Serial.available() > 0)
19     {
20         // read a single character over serial
21         int inByte = Serial.read();

```

Código Fonte para Solução Embarcada - Arduino (2)

```
24 switch (InByte)
25 {
26     case 'u':
27         umidade = dht.readHumidity();
28         if (isnan(umidade)) {
29             Serial.println("DHT11 com erro - Sem leitura de umidade");
30         }
31         else {
32             Serial.print("Umidade: ");
33             Serial.print(umidade);
34             Serial.println(" %");
35         }
36         break;
37
38     case 't':
39         temperatura = dht.readTemperature();
40         if (isnan(temperatura)) {
41             Serial.println("DHT11 com erro - Sem leitura de temperatura");
42         }
43         else {
44             Serial.print("Temperatura: ");
45             Serial.print(temperatura);
46             Serial.println(" C");
47         }
48         break;
49
50     default:
51         Serial.println("Sem leitura ");
52         break;
53 }
54 }
55 }
```

Embarcando o Código Desenvolvido no Arduino

- Embarcar o código-fonte na aplicação
 - Para **evitar conflito**, antes de **embarcar qualquer código-fonte** no Arduino que esteja conectado ao Módulo Bluetooth HC-05, **deixe o pino VCC do Módulo Bluetooth desconectado**

Instituto Federal de São Paulo – IFSP São João da Boa Vista

Desenvolvimento da Aplicação Android (1)

Application Name	Aula08-LerTemperaturaUmidadeDTH11Bluetooth
Package Name	pdm.android.aula08_ler_temperatura_umidade_dht_bluetooth
Activity Name	MainActivity



Instituto Federal de São Paulo – IFSP São João da Boa Vista

Desenvolvimento: AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="aula08_ler_temperatura_umidade_dht_bluetooth.android.aula08_pdm.android.pdm.aula08_ler_temperaturaumidadedth11bluetooth" >

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Aula08-LerTemperaturaUmidadeDTH11Bluetooth"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="Aula08-LerTemperaturaUmidadeDTH11Bluetooth" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

Instituto Federal de São Paulo – IFSP São João da Boa Vista

Desenvolvimento do layout: activity_main.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/conectar" android:layout_width="200dp"
        android:layout_height="wrap_content" android:text="Conectar" />

    <Button
        android:id="@+id/desconectar" android:layout_width="200dp"
        android:layout_height="wrap_content" android:text="Desconectar" />

    <Button
        android:id="@+id/btnMedirTemperatura" android:layout_width="200dp"
        android:layout_height="wrap_content" android:text="Medir Temperatura" />

    <Button
        android:id="@+id/btnMedirUmidade" android:layout_width="200dp"
        android:layout_height="wrap_content" android:text="Medir Umidade" />

    <EditText
        android:layout_width="match_parent" android:layout_height="wrap_content"
        android:inputType="text" android:height="100dp"
        android:id="@+id/edtTxtResultadoMedicao"/>
</LinearLayout>
```

Instituto Federal de São Paulo – IFSP São João da Boa Vista

Desenvolvimento da Activity: MainActivity.java (1)

- Inicialização Atributos da Classe MainActivity e Método onCreate():

```
public class MainActivity extends Activity {
    Button conectar;
    Button desconectar;
    Button receberDadosTemperatura;
    Button receberDadosUmidade;
    EditText medicoesRecebidas;

    // Represents a remote Bluetooth device.
    private BluetoothDevice dispositivoBluetoothRemoto;

    // Represents the local device Bluetooth adapter.
    private BluetoothAdapter meuBluetoothAdapter = null;

    // A connected or connecting Bluetooth socket.
    private BluetoothSocket bluetoothSocket = null;
    private static final String endereco_MAC_do_Bluetooth_Remoto = "98:D3:31:40:31:E8";
    public static final int CÓDIGO_PARA_ATIVAÇÃO_DO_BLUETOOTH = 1;
    private static final UUID MEU_UUID = UUID.fromString("00001101-0000-1000-8000-F9834FE8");
    private InputStream inputStream = null;
    private OutputStream outputStream = null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        fazerConexoesDoLayout_e_Listeners();
        verificarCondiçãoDoBluetooth();
    }
}
```


Desenvolvimento da Activity: MainActivity.java (2)

- Métodos `fazerConexoesDoLayout_e_Listeners()` e `verificarCondiçãoDoBluetooth()`:

```
public void fazerConexoesDoLayout_e_Listeners() {

    conectar          = (Button) findViewById(R.id.conectar);
    desconectar       = (Button) findViewById(R.id.desconectar);
    receberDadosTemperatura = (Button) findViewById(R.id.btnMedirTemperatura);
    receberDadosUmidade = (Button) findViewById(R.id.btnMedirUmidade);
    medicoesRecebidas   = (EditText) findViewById(R.id.edtTxtResultadoMedicao);

    conectar.setOnClickListener(new Conectar());
    desconectar.setOnClickListener(new Desconectar());
    receberDadosTemperatura.setOnClickListener(new ReceberDados());
    receberDadosUmidade.setOnClickListener(new ReceberDados());

}

public void verificarCondiçãoDoBluetooth() {

    // Get a handle to the default local Bluetooth adapter.
    meuBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

    // Verifica se o celular tem Bluetooth
    if (meuBluetoothAdapter == null) {
        Toast.makeText(getApplicationContext(), "Dispositivo não possui adaptador Bluetooth",
            Toast.LENGTH_LONG).show();

        // Finaliza a aplicação.
        finish();
    } else {
        // Verifica se o bluetooth está desligado. Se sim, pede permissão para ligar.
        if (!meuBluetoothAdapter.isEnabled()) {
            Intent novoIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(novoIntent, CÓDIGO_PARA_ATIVAÇÃO_DO_BLUETOOTH);
        }
    }
}
```

Desenvolvimento da Activity: MainActivity.java (3)

- Métodos `onActivityResult()`, `onPause()` e `onResume()`:

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

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

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

@Override
protected void onResume() { super.onResume(); }
```

Desenvolvimento da Activity: MainActivity.java (4)

■ Classe Conectar:

```
public class Conectar implements View.OnClickListener {

    @Override
    public void onClick(View v) {

        // Validate a Bluetooth address, such as "00:43:45:23:10:FD" (Alphabetic characters must be uppercase to be valid)
        if (BluetoothAdapter.checkBluetoothAddress(endereco_MAC_do_Bluetooth_Remoto)) {
            dispositivoBluetoothRemoto = newBluetoothAdapter.getRemoteDevice(endereco_MAC_do_Bluetooth_Remoto);
        } else {
            Toast.makeText(getApplicationContext(), "Endereço MAC do dispositivo Bluetooth remoto não é válido", Toast.LENGTH_SHORT).show();
        }

        try {
            bluetoothSocket = dispositivoBluetoothRemoto.createInsecureRfcommSocketToServiceRecord(MY_UUID);
            bluetoothSocket.connect();
            medicamentosRecebidas.setText("");
            Toast.makeText(getApplicationContext(), "Conectado", Toast.LENGTH_SHORT).show();
        } catch (IOException e) {
            Log.e("ERRO AO CONECTAR", "O erro foi" + e.getMessage());
            Toast.makeText(getApplicationContext(), "Conexão não foi estabelecida", Toast.LENGTH_SHORT).show();
        }
    }
}
```

Desenvolvimento da Activity: MainActivity.java (5)

■ Classe Desconectar:

```
public class Desconectar implements View.OnClickListener {

    @Override
    public void onClick(View v) {
        medicamentosRecebidas.setText("");
        if (bluetoothSocket != null) {
            try {
                // Immediately close this socket, and release all associated resources.
                bluetoothSocket.close();
                bluetoothSocket = null;
                Toast.makeText(getApplicationContext(), "Conexão encerrada", Toast.LENGTH_SHORT).show();
            } catch (IOException e) {
                Log.e("ERRO AO DESCONECTAR", "O erro foi" + e.getMessage());
                Toast.makeText(getApplicationContext(), "Erro - A conexão permanece estabelecida", Toast.LENGTH_SHORT).show();
            }
        } else {
            Toast.makeText(getApplicationContext(), "Não há nenhuma conexão estabelecida a ser desconectada", Toast.LENGTH_SHORT).show();
        }
    }
}
```


Desenvolvimento da Activity: MainActivity.java (6)

- Classe **ReceberDados** – Método **sendData()**:

```
public class ReceberDados implements View.OnClickListener {

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

        try {
            outputStream.write(msgBuffer);
        } catch (IOException e) {
            Toast.makeText(getApplicationContext(), "Erro - Ao enviar dados", Toast.LENGTH_SHORT).show();
        }
    }
}
```

Desenvolvimento da Activity: MainActivity.java (7)

- Classe **ReceberDados** – Método **onClick()** – Parte 1:

```
@Override
public void onClick(View v) {

    // Verifica se há conexão estabelecida com o Bluetooth.
    if(bluetoothSocket != null){

        medicoesRecebidas.setText("");
        switch (v.getId()) {
            case R.id.btnMedirTemperatura:
                try{
                    //Envia Temperatura e umidade
                    outputStream = bluetoothSocket.getOutputStream();
                    sendData("t");
                    SystemClock.sleep(1000);

                    // Get the input stream associated with this socket.
                    inputStream = bluetoothSocket.getInputStream();

                    // Reads bytes from this stream and stores them in the byte array
                    byte[] msgBuffer = new byte[1024];
                    inputStream.read(msgBuffer);

                    medicoesRecebidas.setText(new String(msgBuffer));

                } catch (IOException e){
                    Log.e("ERROR", "O erro foi" + e.getMessage());
                    Toast.makeText(getApplicationContext(), "Mensagem não recebida", Toast.LENGTH_LONG).show();
                }
                break;
            }
        }
    }
}
```

Desenvolvimento da Activity: MainActivity.java (8)

Classe ReceberDados – Método onClick() – Parte 2:

```

case R.id.btnMedirUmidade:
    try{
        //Envia Temperatura e umidade
        outputStream = bluetoothSocket.getOutputStream();
        sendData("u");
        SystemClock.sleep(1000);

        // Get the input stream associated with this socket.
        inputStream = bluetoothSocket.getInputStream();

        // Reads bytes from this stream and stores them in the byte array
        byte[] msgBuffer = new byte[1024];
        inputStream.read(msgBuffer);

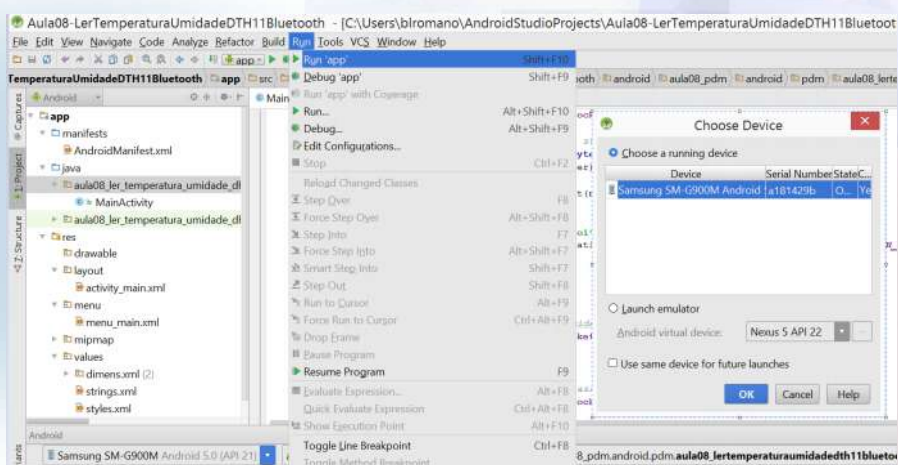
        medicoesRecebidas.setText(new String(msgBuffer));

    } catch (IOException e) {
        Log.e("ERROR", "O erro foi" + e.getMessage());
        Toast.makeText(getApplicationContext(), "Mensagem não recebida", Toast.LENGTH_LONG).show();
    }
    break;

default:
    Toast.makeText(getApplicationContext(), "Botão de medição indisponível", Toast.LENGTH_LONG).show();
    break;
}
} else {
    Toast.makeText(getApplicationContext(), "Bluetooth não está conectado", Toast.LENGTH_LONG).show();
}
}

```

Embarcar o Código do Android no Celular



Após Embarcar o Código, verificar a integração entre o Android e o Arduino

Problem Based Learnd - PBL 02

- **Problema:** Desenvolver (Simplificado) uma Aplicação que integre o Android e Arduino por Bluetooth utilizando botões para Acender e Apagar LEDs (Vermelho, Amarelo, Verde e Branco) de cores diferentes utilizando como entrada a escolha do Usuário ao Acionar os Botões

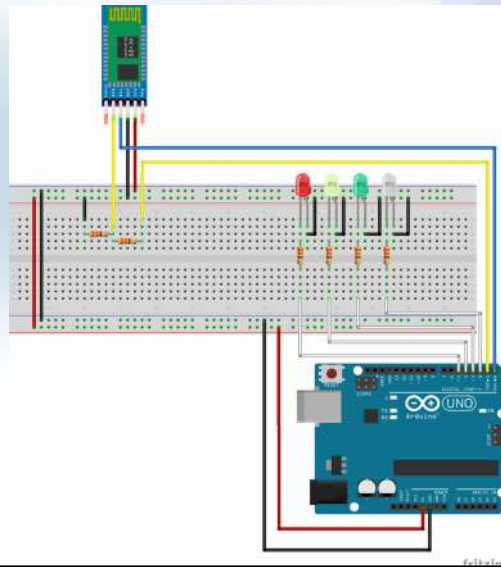
- Pode-se apagar todos os LEDs ou Acender todos os LEDs com botões específicos para eles



Explicação da Lógica de Integração entre Android e Arduino

- Deve-se parear automaticamente o Aplicativo Android ao Arduino quando a Aplicação for acionada
 - Solicitar que o Usuário habilite o Bluetooth, caso não esteja habilitado
- Caso um dos Botões dos LEDs Vermelho, Amarelo, Verde e Branco seja acionado:
 - Enviar o caractere correspondente ao LED para o Arduino
 - r – Vermelho / y – Amarelo / g – Verde / w – Branco
 - Arduino acende o LED caso esteja apagado ou apaga o LED caso esteja aceso
 - A Aplicação Android exibe uma mensagem informando qual LED foi acionado
- Caso um dos Botão “Desligar Todos” ou “Acender Todos” seja acionado:
 - Realizar o mesmo processo descrito anteriormente, mas apagando todos os LEDs ou Acendendo todos os LEDs, respectivamente

Construindo o Circuito no Protoboard



Código Fonte para Solução Embarcada - Arduino (1)

```

1 // The LEDs are connected to these pins
2 int LEDRedPin = 5;
3 int LEDYellowPin = 4;
4 int LEDGreenPin = 3;
5 int LEDWhitePin = 2;
6
7 void setup()
8 {
9     // set up serial at 9600 baud
10    Serial.begin(9600);
11
12    // set all four LED pins to output mode
13    pinMode(LEDRedPin, OUTPUT);
14    pinMode(LEDYellowPin, OUTPUT);
15    pinMode(LEDGreenPin, OUTPUT);
16    pinMode(LEDWhitePin, OUTPUT);
17 }
18
19 void toggleLED(int LEDPin)
20 {
21     // toggle the LED on the pin passed as an argument
22     digitalWrite(LEDPin, !digitalRead(LEDPin));
23 }

```

Código Fonte para Solução Embarcada - Arduino (2)

```

25 void loop()
26 {
27     if (Serial.available() > 0)
28     {
29         // read a single character over serial
30         int inByte = Serial.read();
31
32         switch (inByte)
33         {
34             // if we receive r, y, g, or w toggle the respective LED using our function
35             case 'r':
36                 toggleLED(LEDRedPin);
37                 break;
38             case 'y':
39                 toggleLED(LEDYellowPin);
40                 break;
41             case 'g':
42                 toggleLED(LEDGreenPin);
43                 break;
44             case 'w':
45                 toggleLED(LEDWhitePin);
46                 break;
47             default:
48                 // if we receive any other character turn all the LEDs off
49                 digitalWrite(LEDRedPin, LOW);
50                 digitalWrite(LEDYellowPin, LOW);
51                 digitalWrite(LEDGreenPin, LOW);
52                 digitalWrite(LEDWhitePin, LOW);
53                 break;
54         }
55     }
56 }

```

Embarcando o Código Desenvolvido no Arduino

- Embarcar o código-fonte na aplicação
 - Para **evitar conflito**, antes de **embarcar qualquer código-fonte** no Arduino que esteja conectado ao Módulo Bluetooth HC-05, **deixe o pino VCC do Módulo Bluetooth desconectado**

Instituto Federal de São Paulo – IFSP São João da Boa Vista

Desenvolvimento da Aplicação Android (1)

Application Name	Aula08-AcionarLEDBotaoBluetooth
Package Name	pdm.android.aula08_acionar_led_botao_bluetooth
Activity Name	AcionarLedBotaoBluetoothActivity



Instituto Federal de São Paulo – IFSP São João da Boa Vista

Desenvolvimento: 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="Aula08-AcionarLEDBotaoBluetooth"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".AcionarLedBotaoBluetoothActivity"
            android:label="Aula08-AcionarLEDBotaoBluetooth" >
            <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>
```


Desenvolvimento do 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>
```

Desenvolvimento da Activity: AcionarLedBotaoBluetoothActivity.java (1)

- Inicialização Atributos da Classe AcionarLedBotaoBluetoothActivity e Método onCreate():

```
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();
    }
}
```

Instituto Federal de São Paulo – IFSP São João da Boa Vista

Desenvolvimento da Activity: AcionarLedBotaoBluetoothActivity.java (2)

- Métodos `onPause()`, `verificaStatusBluetooth()` e `onActivityResult()`:

```

@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;
    }
}

```

Instituto Federal de São Paulo – IFSP São João da Boa Vista

Desenvolvimento da Activity: AcionarLedBotaoBluetoothActivity.java (3)

- Métodos `fazerConexoesDoLayoutListeners()` – Parte 1:

```

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();
        }
    });
}

```

Instituto Federal de São Paulo – IFSP São João da Boa Vista

Desenvolvimento da Activity: AcionarLedBotaoBluetoothActivity.java (4)

- Métodos `fazerConexoesDoLayoutListeners()` – Parte 2:

```

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

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

```

Instituto Federal de São Paulo – IFSP São João da Boa Vista

Desenvolvimento da Activity: AcionarLedBotaoBluetoothActivity.java (5)

- Métodos `onResume()`:

```

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() + ".");
    }
}

```

Instituto Federal de São Paulo – IFSP São João da Boa Vista

Desenvolvimento da Activity: AcionarLedBotaoBluetoothActivity.java (6)

- Métodos `errorExit()` e `sendData()`:

```
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);
    }
}
```

Instituto Federal de São Paulo – IFSP São João da Boa Vista

Embarcar o Código do Android no Celular



Após Embarcar o Código, verificar a integração entre o Android e o Arduino

Miniprojeto 12

- Implementar Melhorias e Refatorar o PBL01 desta aula: Desenvolver (Simplificado) uma Aplicação que integre o Android e Arduino utilizando botões para solicitar a Leitura de Temperatura ou Umidade
 - Corrigindo todos os problemas encontrados
 - Reportar o Problema e Explicar como Resolveu
 - Melhorar a IHC da proposta apresentada
 - Adicionar a Leitura do Sensor de Luminosidade para Reportar o Nível de Luminosidade do Ambiente (Baixo, Médio e Alto)
 - Sugestão: Utilizar Threads

Miniprojeto 13

- Implementar Melhorias e Refatorar o PBL02 desta aula: Desenvolver (Simplificado) uma Aplicação que integre o Android e Arduino utilizando botões para Acender e Apagar LEDs
 - Corrigindo todos os problemas encontrados
 - Reportar o Problema e Explicar como Resolveu
 - Melhorar a IHC da proposta apresentada
 - Adicionar um Potenciômetro no Circuito que definirá o nível da Luminosidade dos LEDs, quando eles forem acionados pelos Botões da Aplicação Android
 - Sugestão: Utilizar Threads

Miniprojeto 14

- Desenvolver uma Solução Genérica (Completa) que integre o Android e Arduino para realizar tanto as Leituras de Temperatura/Umidade quanto Acender/Apagar os LEDs

Miniprojeto 15

- Desenvolver uma Aplicação que integre o Android e Arduino por Bluetooth utilizando uma forma iterativa para aumentar ou diminuir a luminosidade de um dos 04 LEDs (Vermelho, Amarelo, Verde e Branco) de cores diferentes utilizando como entrada a escolha do Usuário
 - Inovar na forma que a luminosidade dos LEDs será realizada



PDM: Projeto para Dispositivos Móveis
Aula 08: Integrando o Android e o Arduino utilizando o Bluetooth
Breno Lisi Romano

Obrigado!

Instituto Federal de São Paulo – IFSP São João da Boa Vista
Especialização em Desenvolvimento para Dispositivos Móveis



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus São João da Boa Vista