



Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

Read the guide

 [akexorcist](#) / [BluetoothSPPLibrary](#)

[Android] Bluetooth Serial Port Profile which comfortable to developer application to communication with microcontroller via bluetooth

 Apache-2.0 License

 1.7k stars  569 forks

 Star

 Watch ▼

Code Issues 79 Pull requests 10 Actions Projects Wiki Security Insights

 master ▼



akexorcist ...

on May 24, 2015 

[View code](#)

README.md

build unknown

Android-BluetoothSPPLibrary

BluetoothSPP

What is it?



Bluetooth Serial Port Profile which comfortable to developer application to communication with microcontroller or android device via bluetooth.

This libraly include all important methods for serial port profile on bluetooth communication. It has built-in bluetooth device list.

Feature

- It's very easy to use
- Solve the lack of data like as "abcdefg" which divided to "abc" and "defg" when receive these data
- Auto add LF (0x0A) and CR (0x0D) when send data to connection device
- No need to create layout for bluetooth device list to select device for connection. You can use built-in layout in this library and you can customize layout if you want
- Auto connection supported
- Listener for receive data from connection device

Download

Maven

```
<dependency>  
  <groupId>com.akexorcist</groupId>  
  <artifactId>bluetoothspp</artifactId>  
  <version>1.0.0</version>  
</dependency>
```

Gradle

```
compile 'com.akexorcist:bluetoothspp:1.0.0'
```

Simple Usage

- Import this library to your workspace and include in to your android project For Eclipse ADT : Download this library and import into your workspace and include this library to your project For Android Studio : Use Gradle to download this library from Maven

- Declare permission for library

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

- Declare BluetoothSPP like this

```
BluetoothSPP bt = new BluetoothSPP(Context);
```

- Check if bluetooth is now available

```
if(!bt.isBluetoothAvailable()) {
    // any command for bluetooth is not available
}
```

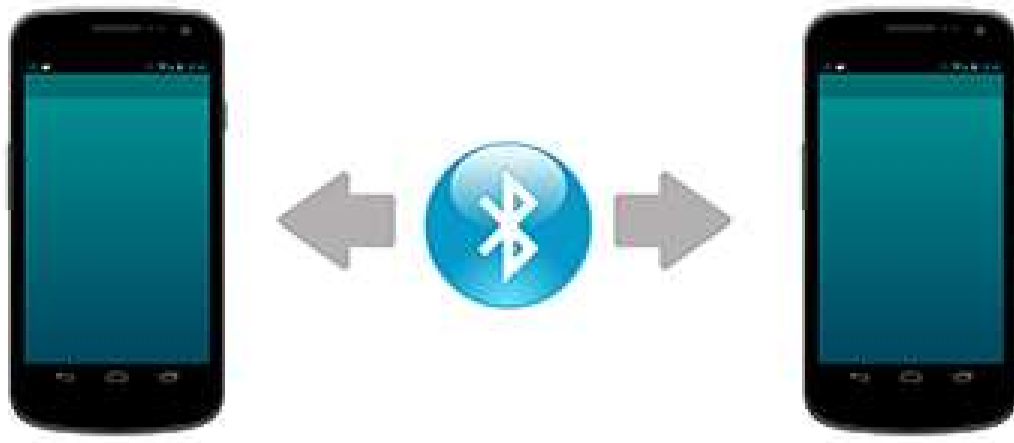
- Check if bluetooth is not enable when activity is onStart

```
public void onStart() {
    super.onStart();
    if(!bt.isBluetoothEnable()) {
        // Do something if bluetooth is disable
    } else {
        // Do something if bluetooth is already enable
    }
}
```

- if bluetooth is ready call this method to start service

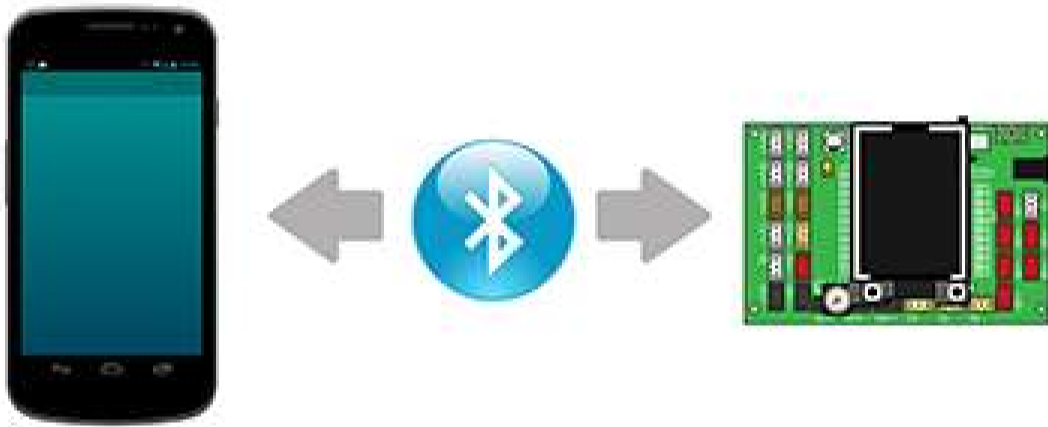
For connection with android device

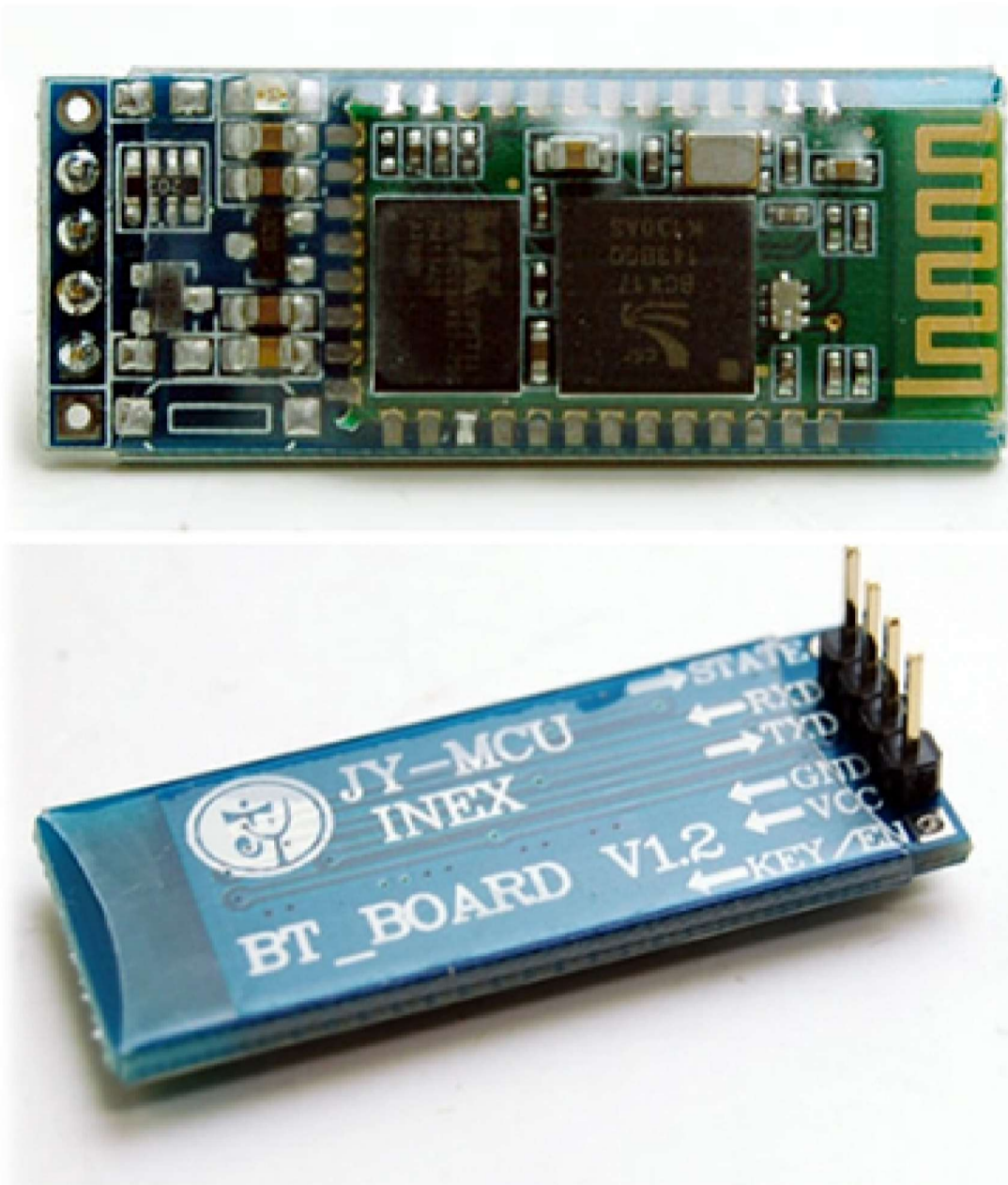
```
bt.startService(BluetoothState.DEVICE_ANDROID);
```



For connection with any microcontroller which communication with bluetooth serial port profile module

```
bt.startService(BluetoothState.DEVICE_OTHER);
```





- Stop service with

```
bt.stopService();
```

- Intent to choose device activity

```
Intent intent = new Intent(getApplicationContext(), DeviceList.class);
startActivityForResult(intent, BluetoothState.REQUEST_CONNECT_DEVICE);
```

don't forget declare library activity like this

```
<activity android:name="app.akexorcist.bluetoothspp.DeviceList" />
```

- After intent to choose device activity and finish that activity. You need to check result data on onActivityResult

```
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    if(requestCode == BluetoothState.REQUEST_CONNECT_DEVICE) {
        if(resultCode == Activity.RESULT_OK)
            bt.connect(data);
    } else if(requestCode == BluetoothState.REQUEST_ENABLE_BT) {
```

```

        if(resultCode == Activity.RESULT_OK) {
            bt.setupService();
            bt.startService(BluetoothState.DEVICE_ANDROID);
            setup();
        } else {
            // Do something if user doesn't choose any device (Pressed back)
        }
    }
}

```

- If you want to send any data. boolean parameter is mean that data will send with ending by LF and CR or not. If yes your data will added by LF & CR

```
bt.send("Message", true);
```

or

```
bt.send(new byte[] { 0x30, 0x38, ....}, false);
```

- Listener for data receiving

```

bt.setOnDataReceivedListener(new OnDataReceivedListener() {
    public void onDataReceived(byte[] data, String message) {
        // Do something when data incoming
    }
});

```

- Listener for bluetooth connection atatus

```

bt.setBluetoothConnectionListener(new BluetoothConnectionListener() {
    public void onDeviceConnected(String name, String address) {
        // Do something when successfully connected
    }

    public void onDeviceDisconnected() {
        // Do something when connection was disconnected
    }

    public void onDeviceConnectionFailed() {
        // Do something when connection failed
    }
});

```

- Listener when bluetooth connection has changed

```

bt.setBluetoothStateListener(new BluetoothStateListener() {
    public void onServiceStateChanged(int state) {
        if(state == BluetoothState.STATE_CONNECTED)
            // Do something when successfully connected
        else if(state == BluetoothState.STATE_CONNECTING)
            // Do something while connecting
        else if(state == BluetoothState.STATE_LISTEN)
            // Do something when device is waiting for connection
        else if(state == BluetoothState.STATE_NONE)
            // Do something when device don't have any connection
    }
});

```

- Using auto connection

```

bt.autoConnect("Keyword for filter paired device");

```

- Listener for auto connection

```

bt.setAutoConnectionListener(new AutoConnectionListener() {
    public void onNewConnection(String name, String address) {
        // Do something when earching for new connection device
    }

    public void onAutoConnectionStarted() {
        // Do something when auto connection has started
    }
});

```

- Customize device list's layout by create layout which include

list view with id name = "list_devices"

button with id name = "button_scan"

Example

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#FDE182" >

    <ListView
        android:id="@+id/list_devices"
        android:layout_width="match_parent"

```

```

android:layout_height="wrap_content"
android:layout_alignParentTop="true"
android:layout_marginLeft="20dp"
android:layout_marginRight="20dp"
android:layout_marginTop="20dp"
android:smoothScrollbar="true" />

```

```

<Button

```

```

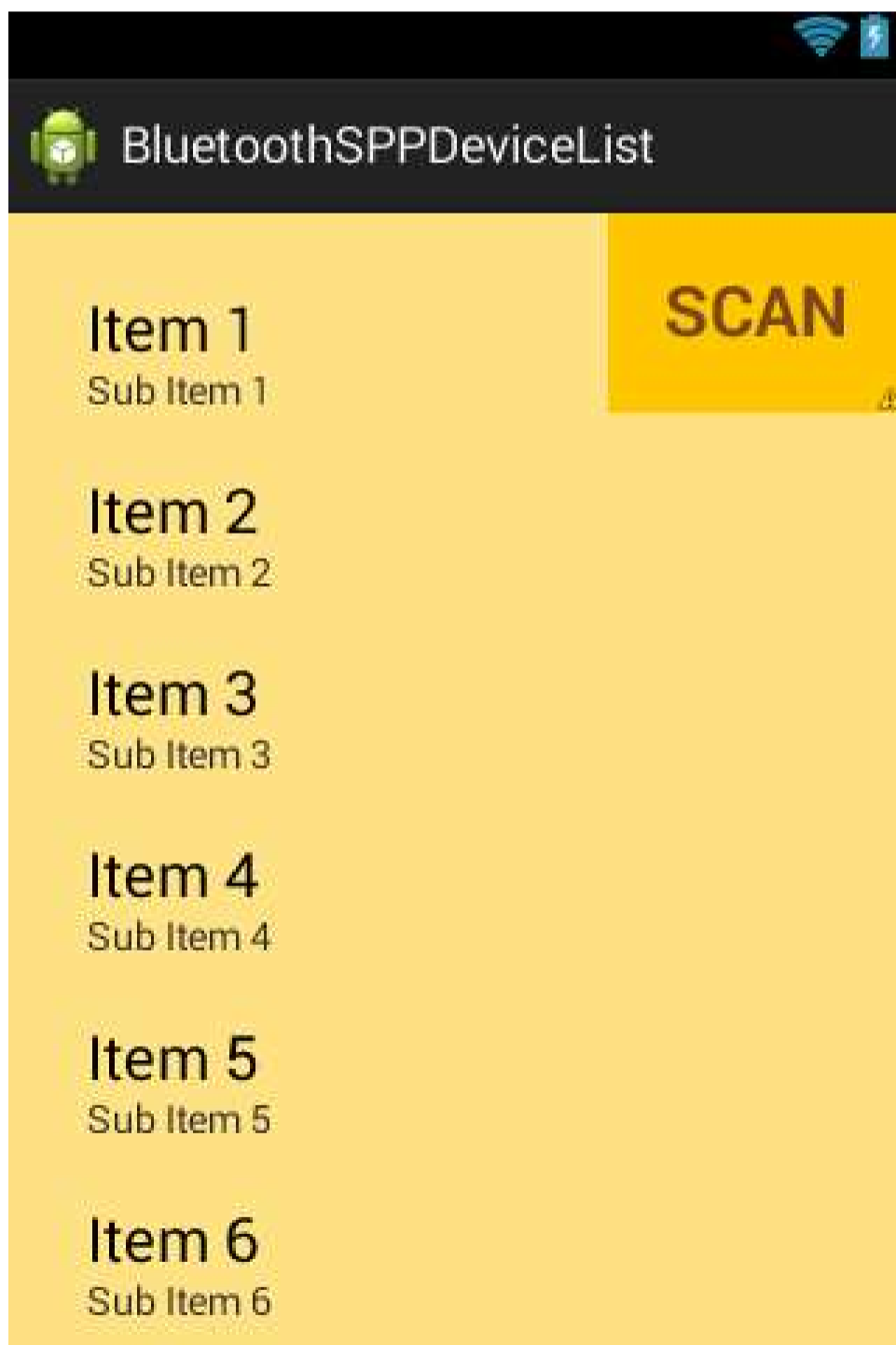
    android:id="@+id/button_scan"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:padding="20dp"
    android:background="#FFC600"
    android:text="SCAN"
    android:textSize="25sp"
    android:textColor="#7A481B"
    android:textStyle="bold" />

```

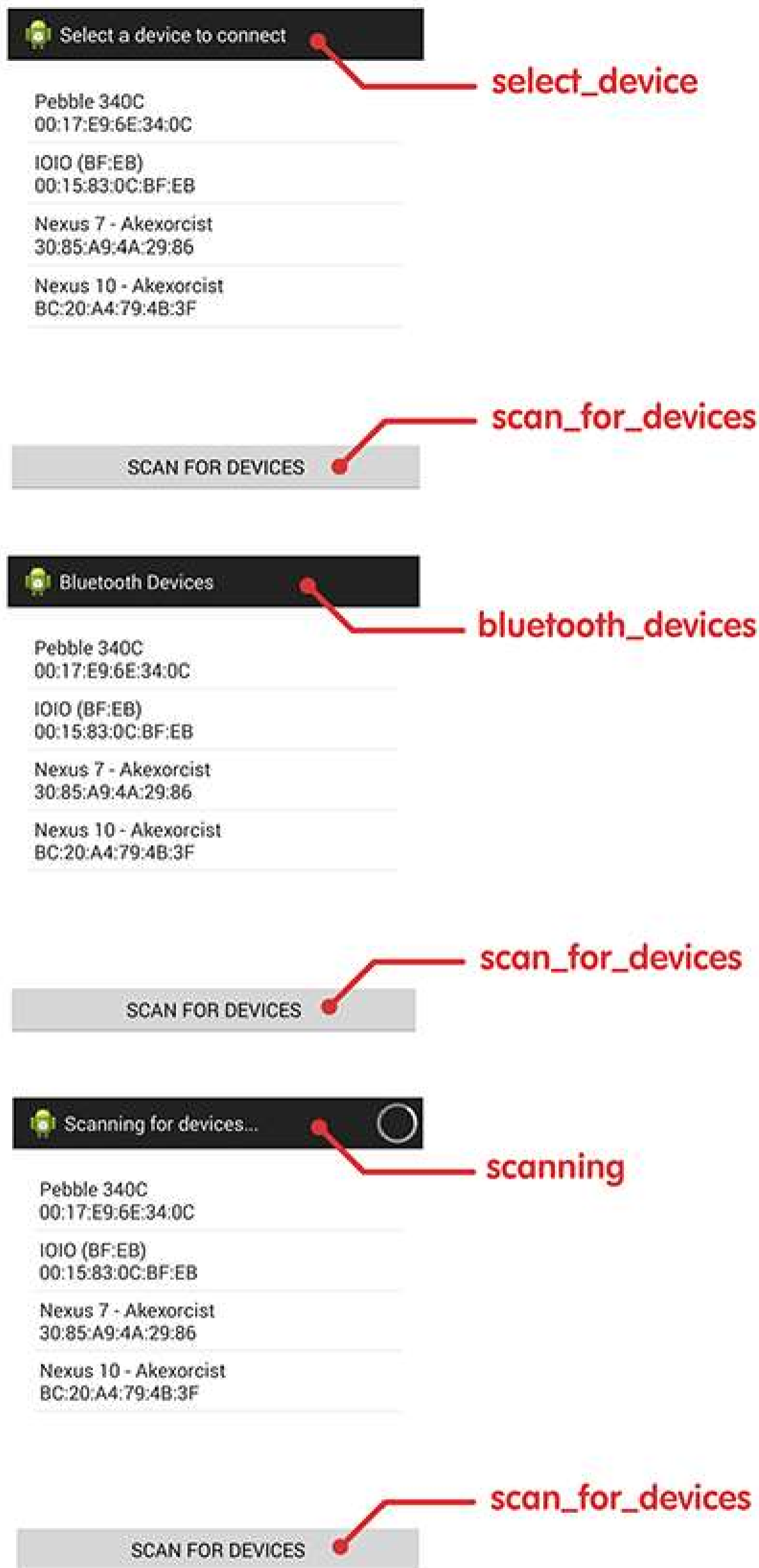
```

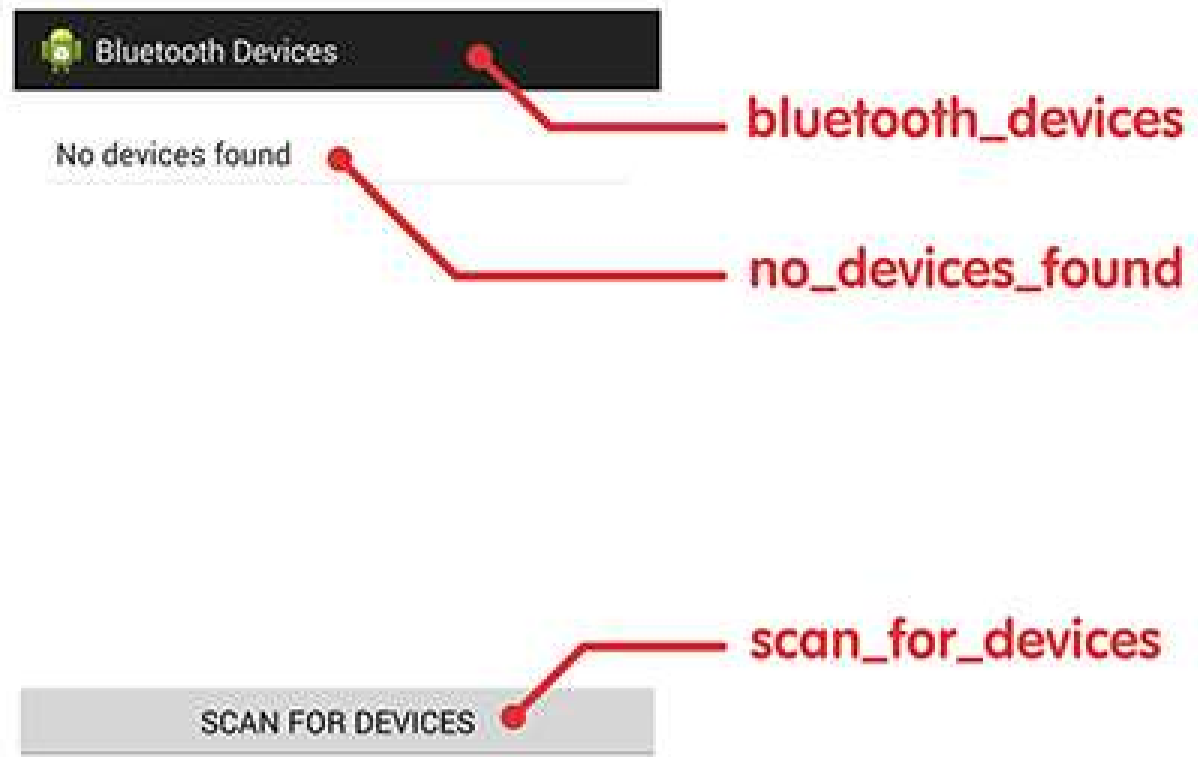
</RelativeLayout>

```

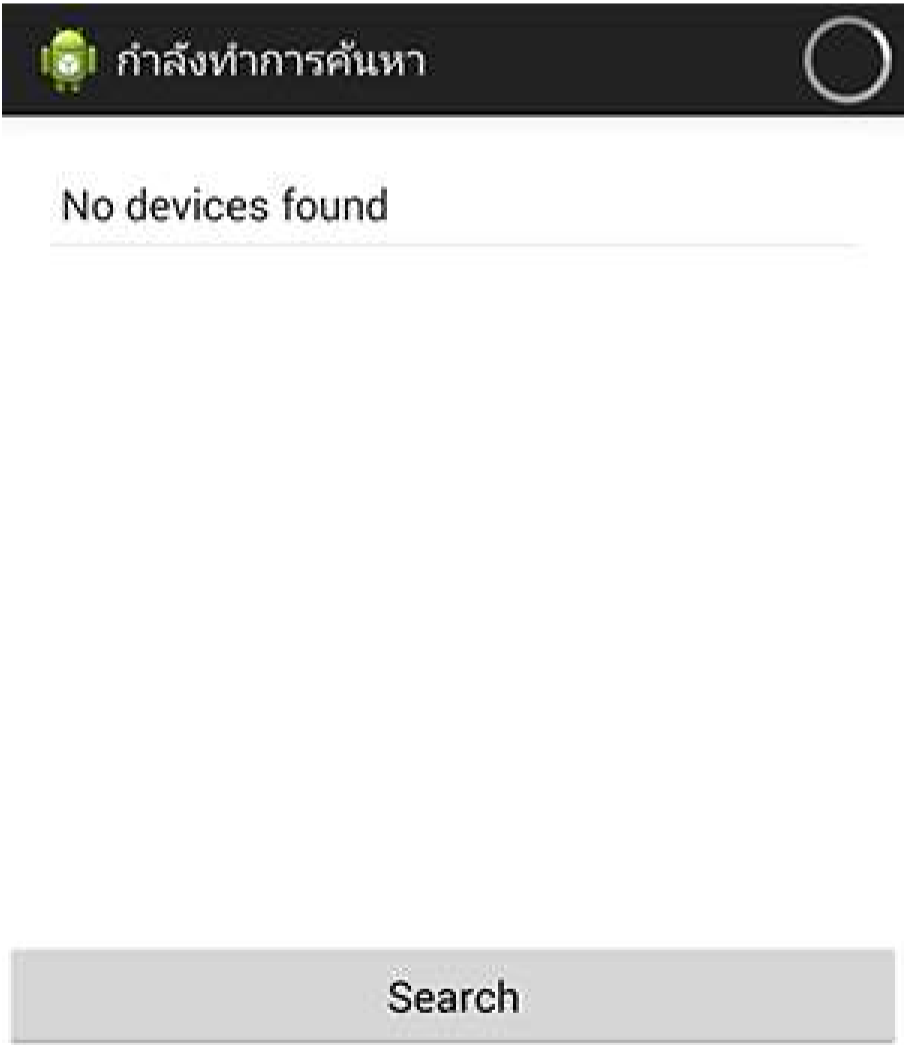


But if you don't need to create layout file. You just want to change only text on device list layout. You can use bundle to change text on device list





```
Intent intent = new Intent(getApplicationContext(), DeviceList.class);
intent.putExtra("bluetooth_devices", "Bluetooth devices");
intent.putExtra("no_devices_found", "No device");
intent.putExtra("scanning", "กำลังทำการค้นหา");
intent.putExtra("scan_for_devices", "Search");
intent.putExtra("select_device", "Select");
startActivityForResult(intent, BluetoothState.REQUEST_CONNECT_DEVICE);
```



What's next?

- Connection Dialog
- Add Insecure Connection

- Fix bug on this issue <https://github.com/akexorcist/Android-BluetoothSPPLibrary/issues/21>
- Merge the code from <https://github.com/akexorcist/Android-BluetoothSPPLibrary/pull/14> for a problem of auto connection
- Human Readable Log <https://github.com/akexorcist/Android-BluetoothSPPLibrary/issues/19>

License


Copyright (c) 2014 Akexorcist

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Releases 1

 **1.0: Update** Latest

on Jan 22, 2015

Packages

No packages published

Languages

● **Java** 100.0%