



FEASYCOM

Android FeasyBeacon SDK

Reference Manual

Version 1.0

深圳易盛科技有限公司
FEASYCOM



Copyright © 2013-2017 Feasycom Technology Co., Ltd. All Rights Reserved.

Revision History

Version	Date	Notes	Author
1.0	2018/8/22	First Release	Younger

深圳市飞易通科技有限公司
FEASYCOM

Table of Contents

1. Introduction.....	4
1.1 Android System Version Requirements	4
1.2 Supported Android devices	4
1.3 Supported Bluetooth Profile	4
2. Get started with FeasyBeacon.....	5
2.1 General Tools.....	5
2.2 FeasyBeacon Demo App Project Setup	5
2.3 Download and Run the FeasyBeacon Demo App.....	5
2. FeasyBeacon Architecture.....	6
2.1 FeasyBeacon System Architecture	6
2.2 Activity.....	6
3. Operating Examples.....	7
3.1 Typical Initialization and Connection Setup.....	7
4. SDK API	8
4.1 CALLBACKS.....	8
4.2 METHODS	9

1. Introduction

This reference manual presents design guidelines for software engineers that use Android FeasyBeacon SDK to create Android App for Bluetooth connectivity requirements.

1.1 Android System Version Requirements

- Android 4.3 and above

1.2 Supported Android devices

- Phone or tablet with OS of Android 4.3 and above.

1.3 Supported Bluetooth Profile

- GATT (Generic Attribute Profile, relevant to BLE)

2. Get started with FeasyBeacon

2.1 General Tools

The development of FeasyBeacon is base on Android Studio 3.0.1 and the version of Gradle is 4.1

2.2 FeasyBeacon Demo App Project Setup

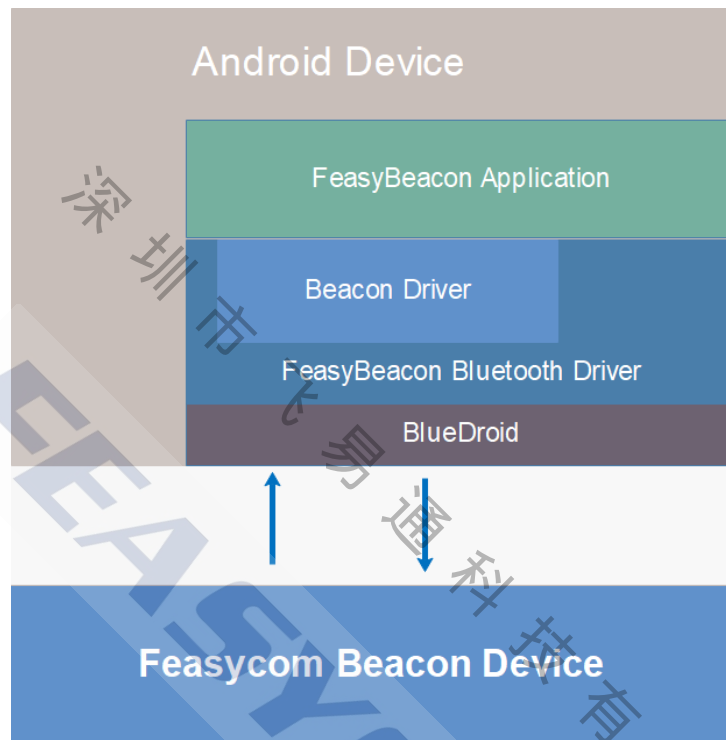
1. Start Android Studio 3.0.1
2. Choose “File->Open”
3. Browse the project folder
4. Open the project

2.3 Download and Run the FeasyBeacon Demo App

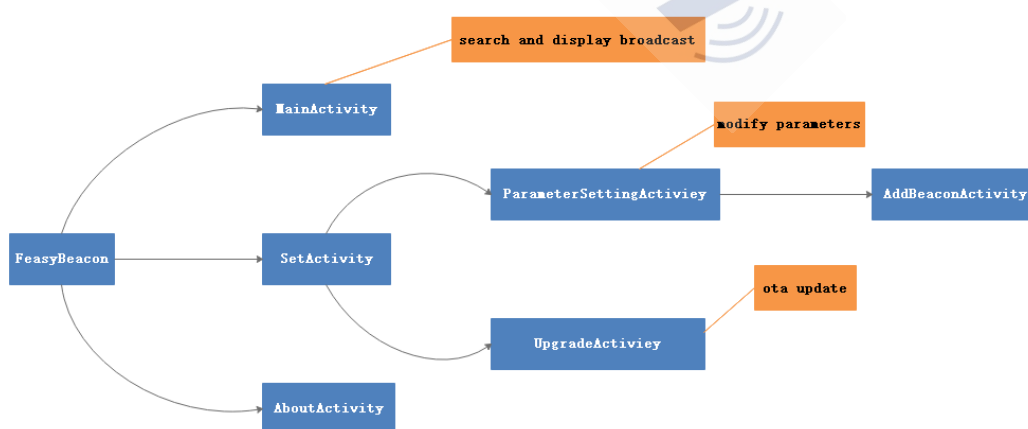
When FeasyBeacon the application starts, it will scan a nearby radio and show it out. You can try to connect your beacon equipment in the setup interface. When you connect successfully, you can modify your radio beacon information. FeasyBeacon another function for the OTA upgrade, when you need to use this feature, you need to provide a firmware upgrade.

2. FeasyBeacon Architecture

2.1 FeasyBeacon System Architecture

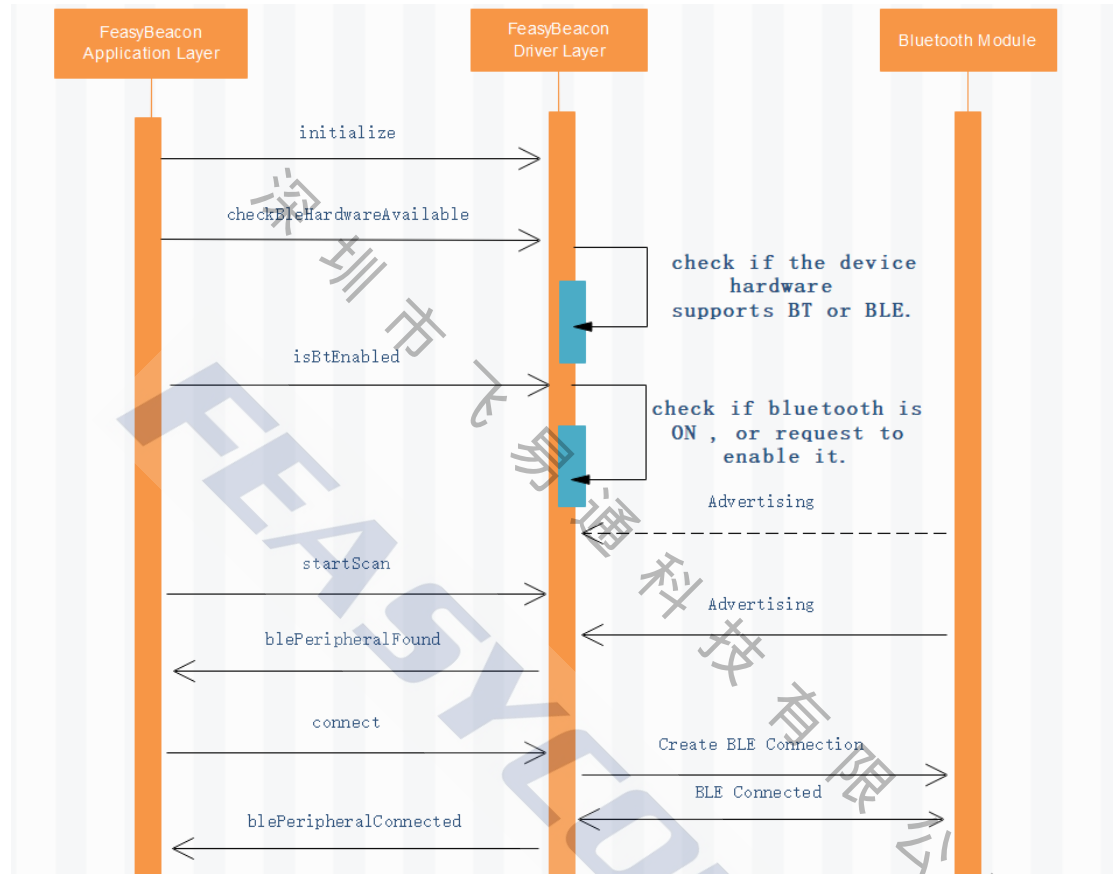


2.2 Activity



3. Operating Examples

3.1 Typical Initialization and Connection Setup



4. SDK API

4.1 CALLBACKS

<pre> /* * Peripheral found callback, * @param device The peripheral device. * @param rssi The current RSSI of device, in dBm. * @param record The scan record. * @discussion Call startScan(),the discovered devices will be returned. */ -(void)blePeripheralFound(BluetoothDeviceWrapper device, int rssi, byte[] record) </pre>
<pre> /* * Peripheral connected callback, * @param gatt The gatt used by the connection process * @param device Current connected device. * @discussion This method is invoked when a connection is set up * successfully */ -(void)blePeripheralConnected(BluetoothGatt gatt, BluetoothDevice device) </pre>
<pre> /* * Discover services callback, * @param gatt The gatt used by the connection process * @param device Current connected device. * @param services The array of services information. */ -(void)servicesFound(BluetoothGatt gatt, BluetoothDevice device, ArrayList<BluetoothGattService> service) </pre>
<pre> /* * Peripheral disconnected callback, * @param gatt The gatt used by the connection process * @param device Current connected device. * @discussion This method is invoked when a disconnect event occurs. */ -(void)blePeripheralDisconnected(BluetoothGatt gatt, BluetoothDevice device) </pre>
<pre> /* * Received packet callback, * @param gatt The gatt used by the connection process * @param device Current connected device. * @param service The service of current characteristic * @param ch The current characteristic </pre>

<pre> * @param strValue received data in String form * @param hexString received data in hex String form * @param rawValue received data * @param timestamp invalid value * @discussion This method is called when data is returned from the * peripheral. */ -(void)packetReceived(BluetoothGatt gatt, BluetoothDevice device, BluetoothGattService service, BluetoothGattCharacteristic ch, String strValue, String hexString, byte[] rawValue, String timestamp) </pre>
<pre> /* * Device information callbacks, * @param parameterName Name of the device information * @param parameter Value of the device information * @discussion This method returns the result of a * startGetDeviceInfo() call. */ -(void)deviceInfo(String parameterName, Object parameter) </pre>
<pre> /* * OTA update callbacks, * @param percentage This parameter is the upgrade progress. * @param status This parameter is the upgrade status. * @discussion This method returns the result of a startOTA() call. */ -(void)otaProgressUpdate(int percentage, int status) </pre>
<pre> /* * AT command callbacks * @param command Content of command * @param param value of command parameter * @param status 0: fail * 1: success * 2: finished * 3: no need to modify. */ -(void)atCommandCallback(String command, String param, String status) </pre>

4.2 METHODS

<pre> /* * @discussion Initialization */ -(void)initialize() </pre>
<pre> /* * @discussion Check if the device has available BT and BLE hardware */ </pre>

boolean checkBleHardwareAvailable()
<pre> /* * @discussion check bluetooth is ON or not. */ </pre>
boolean isBtEnabled()
<pre> /* * @param time The scan time. * @discussion Start scan peripherals and stop after "time" ms. */ </pre>
-(void)startScan(int time)
<pre> /* * @discussion Stop scan peripherals. */ </pre>
-(void)stopScan()
<pre> /* * Connect peripheral, */ boolean connect(BluetoothDeviceWrapper device, String pin2Connect); </pre>
<pre> /* * @discussion Disconnect peripheral. */ </pre>
-(void)disconnect()
<pre> /* * @discussion Discover services. */ </pre>
-(void)discoverServices
<pre> /* * @discussion Get the device information. */ </pre>
-(void)startGetDeviceInfo(String moduleString)
<pre> /* * @discussion Set the device name, this parameter "deviceName" * represents the name you want to modify. */ </pre>
-(void)setDeviceName(String value)
<pre> /* * @discussion Set the beacon pin, this parameter "pin" represents the * password you want to modify. */ </pre>
-(void)setFscPin(String value)
<pre> /* * @discussion Set the broadcast interval, this parameter "interval" * represents the interval you want to modify. */ </pre>

<pre> -(void)setBroadcastInterval(String value) </pre>	
<pre> /* * @discussion * * */ </pre>	<p>Set the tx power, this parameter “txPower” represents the transmitted power you want to modify.</p>
<pre> -(void)setTxPower(String value) </pre>	
<pre> /* * @discussion * * */ </pre>	<p>Set the connectable or not, this parameter “connectable” represents the connectable you want to open or closed.</p>
<pre> -(void)setConnectable(boolean on) </pre>	
<pre> /* * @discussion * * */ </pre>	<p>Query whether beacon broadcast to upper limit. If yes, to limit.</p>
<pre> -boolean isBeaconInfoFull() </pre>	
<pre> /* * @discussion * * */ </pre>	<p>Add beacon broadcast, four types, iBeacon, URL, UID and AltBeacon.</p>
<pre> -boolean addBeaconInfo(BeaconBean beaconBean) </pre>	
<pre> /* * @discussion * * * */ </pre>	<p>Delete beacon broadcast, this parameter “index” represents the number of beacon broadcasts you want to delete.</p>
<pre> -boolean deleteBeaconInfo(String index) </pre>	
<pre> /* * Update beacon broadcast, * @param beaconBean * @param index * */ </pre>	<p>Beacon broadcast type :iBeacon, URL, UID and AltBeacon. This number of beacon broadcasts you want to update.</p>
<pre> -(void)updateBeaconInfo(BeaconBean beacon, String index) </pre>	
<pre> /* * Get beacon information, * @param index * * * @discussion * */ </pre>	<p>This parameter represents which one beacon broadcast you want to get. Returns a beacon type.</p>
<pre> -(BeaconBean *)getBeaconInfo(String index) </pre>	
<pre> /* * @discussion * * */ </pre>	<p>Save beacon broadcast information, after when you modify the beacon broadcast, call this method to save the beacon broadcast information.</p>

```
*/  
-(void)saveBeaconInfo()  
/*  
 * This method is called to upgrade,  
 * @param dfuFile          This parameter is the name of the upgrade file.  
 * @param restoreDefaultSetting  Restore the factory settings.  
 */  
-boolean startOTA(byte[] dfuFile , Boolean restoreDefaultSetting)
```

深圳市飞易通科技有限公司
FEASYCOM