



Level:public information

Android BLE API User Guide

September 27, 2017

version 1.0

Copyright © ShenZhen ShengRun Technology Co.,Ltd.

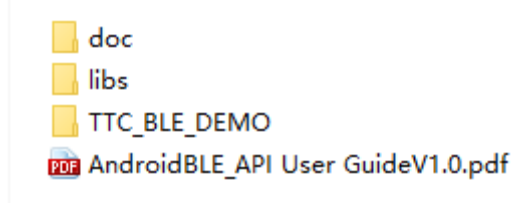
All Rights Reserved

Version	Revision Date	Revisionist	Reviewers	Revised Content
1.0	2016-09-27	Jia Jiefei	Zhang Yan	First release

Contents

1.Android BLE SDK Introduction.....	2
2.Project Configuration	2
3.Sample Code	3
3.1 Instantiate BleService	3
3.2 Connect device	4
3.3 Send data to device	4
3.3.1 Default send UUID.....	4
3.3.2 Other UUIDs	5
3.4 Receive data from device	5
3.4.1 Notify	5
3.4.2 Read	6
3.5 Advertising data	8
3.5.1 Manufacturer Specific Data	8
4.Contact us.....	10

1. Android BLE SDK Introduction



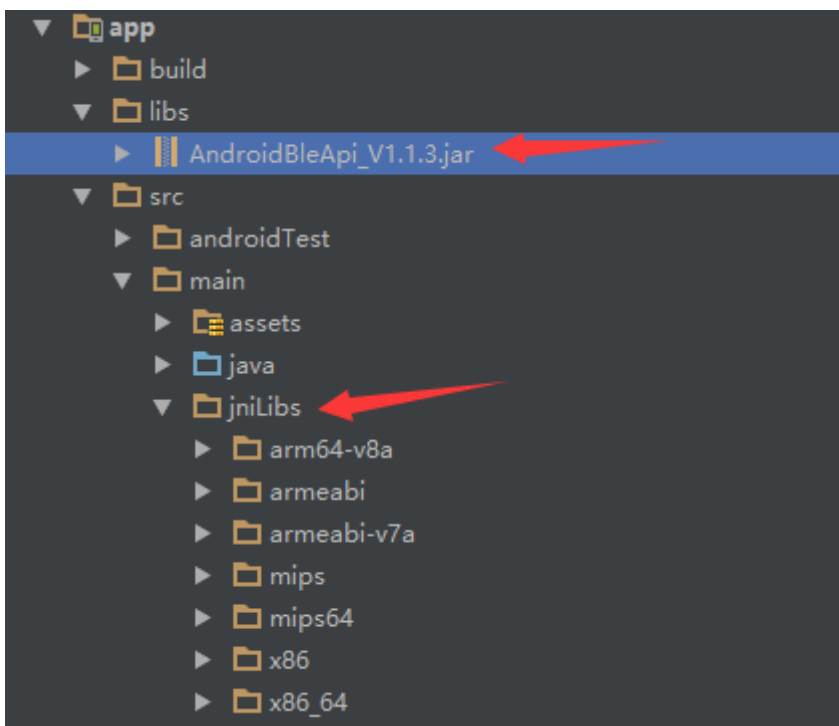
1)doc is
 java document;

2)TTC_
 BLE_DEMO is the sample code;

3)libs
 folder include jar file and so file.

2. Project Configuration

1) Copy lib files to your project :



2) BLE supported on android system since Android4.3(API Level 18),so need add

configs in AndroidManifest.xml file :

```
<uses-sdkandroid:minSdkVersion="18"/>
```

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

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

3) Enable the BleService in AndroidManifest.xml file:

```
<serviceandroid:name="com.ble.ble.BleService"
```

```
    android:enabled="true"android:exported="false"/>
```

3. Sample Code

3.1 Instantiate BleService

```
private BleService mLeService;
```

```
private final ServiceConnection conn = new ServiceConnection() {
```

```
    @Override
```

```
    public void onServiceDisconnected(ComponentName name) {
```

```
        mLeService = null;
```

```
    }
```

```
@Override
```

```
public void onServiceConnected(ComponentName name, IBinder service) {
```

```
    // BleCallBack is a important parameter that will be mentioned later
```

```
    mLeService =
```

```
((BleService.LocalBinder) service).getService(mBleCallback);
```

```
// Must call initialize() method
```

```
mLeService.initialize();
```

```
}
```

```
};
```

```
// Activity bind service
```

```
bindService(newIntent(this, BleService.class),
```

```
conn, BIND_AUTO_CREATE);
```

3.2 Connect device

```
mLeService.connect(mac,false);
```

parameter1: device address;

parameter2: whether reconnect device after disconnected.

3.3 Send data to device

3.3.1 Default send UUID

The default send UUID is 16-bit UUID 0x1001:

```
mLeService.send(mac,value,true);
```

parameter1: device address;

parameter2: data to send, can be hexadecimal string or byte array;

parameter3: whether encrypt data. If true, app could send up to 17 bytes at once . If false,

app could send up to 20bytes at once.

3.3.2 Other UUIDs

For example, service 0x2000, characteristic 0x2001:

```
BluetoothGatt gatt = mLeService.getBluetoothGatt(mac);

BluetoothGattService service = gatt.getService(

UUID.fromString("00002000-0000-1000-8000-00805f9b34fb"));

BluetoothGattCharacteristic characteristic = service.getCharacteristic(

UUID.fromString("00002001-0000-1000-8000-00805f9b34fb"));

byte[] value = {(byte) 0xaa, (byte) 0xbb, (byte) 0xcc};

mLeService.send(gatt, characteristic, value, true);
```

3.4 Receive data from device

App have two mode to receive data: notify and read.

3.4.1 Notify

In this mode the app will receive data when device initiate send operations. A `onCharacteristicChanged()` callback will be triggered when device initiate a send operation. For receiving data, app need to enable notification:

```
// Enable notification

BluetoothGatt gatt = mLeService.getBluetoothGatt(mac);

BluetoothGattService service = gatt.getService(
```

```
UUID.fromString("00001000-0000-1000-8000-00805f9b34fb");

BluetoothGattCharacteristic character = service.getCharacteristic(

UUID.fromString("00001002-0000-1000-8000-00805f9b34fb"));

mLeService.setCharacteristicNotification(gatt, character,true);

// Receive data

private final mBleCallback = new BleCallback() {

    @Override

    public void onCharacteristicChanged(

        String mac, BluetoothGattCharacteristic characteristic) {

//uuid

        UUID charUuid= characteristic.getUuid();

// Received data

        byte[] data = characteristic.getValue();

        }

};
```

Note: Must enable notification after onServicesDiscovered() callback triggered.

3.4.2 Read

In this mode app need to read from device:

```
BluetoothGatt gatt = mLeService.getBluetoothGatt(mac);
```



```

BluetoothGattService service = gatt.getService(

UUID.fromString("00001000-0000-1000-8000-00805f9b34fb"));

BluetoothGattCharacteristic characteristic = service.getCharacteristic(

UUID.fromString("00001004-0000-1000-8000-00805f9b34fb"));

gatt.readCharacteristic(characteristic);

// Receive data

private final mBleCallback = new BleCallback() {

    @Override

    public void onCharacteristicRead(

        String mac, BluetoothGattCharacteristic character, int status) {

if (status == BluetoothGatt.GATT_SUCCESS) {

//uuid

UUID charUuid = characteristic.getUuid();

// Received data

byte[] data = characteristic.getValue();

}

}

};
    
```

3.5 Advertising data

Since v1.0.7, sdk supported to parse advertising data. App can get a byte array named 'scanRecord' when scanning devices:

```
LeScanRecord record = LeScanRecord.parseFromBytes(scanRecord);
```

Supported advertising information: Advertise Flags ,Local Name ,Service UUID ,Service Data , Tx Power Level , Manufacturer Specific Data , Slave Connection Interval Range.
record.getBytes()return the raw advertising data.

3.5.1 Manufacturer Specific Data

Some of our products, contains some information in the advertising data appeared as manufacturer specific data. The standard format of manufacturer specific data is: manufacturer id (2 bytes, low byte first, high byte after) + data, and can contains multiple manufacturer data, our products are generally contains only one.You can get the original manufacturer data by the following steps:

```
List<byte[]>rawDataList = new ArrayList<byte[]>();
```

```
SparseArray<byte[]> mfrData = record.getManufacturerSpecificData();

for (inti = 0; i<mfrData.size(); i++) {

intid = mfrData.keyAt(i);

byte[] data = mfrData.get(id);

byte[] rawData = new byte[2 + data.length]; //single manufacturer data

rawData[0] = (byte) (id & 0xff);

rawData[1] = (byte) ((id >> 8) & 0xff);

System.arraycopy(data, 0, rawData, 2, data.length);

rawDataList.add(rawData);

}
```

4.Contact us

ShenZhenShengRun Technology Co.,Ltd.

Tel: 0755-86233846 Fax: 0755-82970906

Official website: www.tuner168.com

Alibaba website: <http://shop1439435278127.1688.com>

E-mail: marketing@tuner168.com

Address: Room 602, B Block of Jingu Pioneer Park, Longzhu 4th Road,Xili

Town,Nanshan District, Shenzhen

