BLE Bluetooth Communication Practice under Android Platform

Xiaolong Zheng[1]✎, Jinkai Che[1], Ren Chen[1]

[1]Chengdu university of Technology,Chengdu, 610059, China

Abstract: Bluetooth technology is a solution for wireless transmission of consumer electronics products, especially low-power Bluetooth technology is more and more widely used in recent years due to its ultra-low operating power consumption and standby power consumption. This article details the communication process of Bluetooth Low Energy version 4.2 based on the Android platform application.

Keywords: Bluetooth low energy, Android platform, Bluetooth 4.2

Introduction

Bluetooth technology has an important application in wireless communication technology. It is suitable for communication transmission in a short distance and can reliably exchange information between related peripheral devices including mobile phones, wireless headphones, portable computers and other related devices. Bluetooth low energy is a new type of ultra-low power wireless transmission technology introduced after Bluetooth 4.0, which has extremely low operating and standby power consumption. The Android platform supports Bluetooth low energy technology after version 4.3 of its system and provides corresponding APIs. Through this article, we will sort out the Bluetooth low energy communication process under the Android platform.

1 Common Bluetooth low energy terms

Before detailing Bluetooth low energy communication, it is necessary to understand some of the terms and concepts of Bluetooth low energy! Common terms include general attribute configuration file (GATT), attribute protocol (ATT), characteristic and service.

1.1 General attribute configuration file (GATT)

The GATT configuration file is a general specification for the transmission and reception of "attribute" data fragments on Bluetooth low energy links.

1.2 Attribute Agreement (ATT)

ATT is the foundation of GATT construction. Each attribute must have a unique identifier (UUID) for unique identification. The attributes transmitted by ATT are the format of features and services. The attributes transmitted by ATT are formatted as features and services.

1.3 Characteristic

Contains one value and zero to more defined attributes that describe characteristic values. You can think of features as types, similar to classes.

1.4 Service

Service refers to a series of features, which is a collection of features. For example, you may have a service called "Temperature Detector" which includes features like "Maximum Temperature".

2 Android Bluetooth Low Energy Development

Android has provided low-power Bluetooth APIs since version 4.3, so the minimum system requirements must be version 4.3 or higher. The overall process is shown in Figure 1:
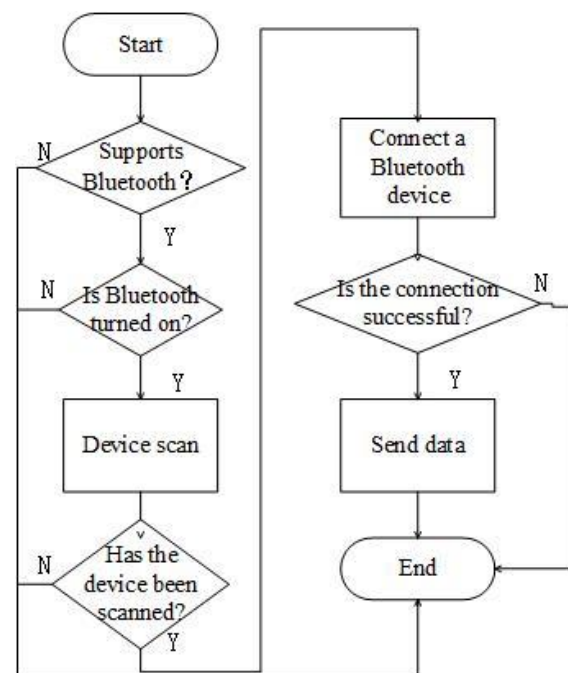

Figure 1 Bluetooth low energy communication flow chart

2.1 Obtain application permissions

If you want to use the Bluetooth function in the program, you need to open the Bluetooth permission,

✎ **Xiaolong Zheng (Correspondence)**
☎ +

and you can only perform all related operations related to Bluetooth communication after obtaining the permission, such as searching for devices, requesting connections, and sending and receiving data.

Before we start programming, we need to add application permissions in the AndroidManifes file. When the Android API is less than 23:

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

When the Android API is greater than or equal to 23, you need to add:

```
<uses-permission
android:name="android.permission.ACCESS_COARS
E_LOCATION>
```

2.2 Get Bluetooth adapter

All Bluetooth low energy requires a Bluetooth adapter. Android 4.3 introduced BluetoothManager, the application can get Bluetooth adapter through BluetoothManager class. The whole system has only one Bluetooth adapter, which is used for various general operations of Bluetooth low energy. The following code snippet shows how to obtain the adapter.

```
final BluetoothManager bluetoothManager =
(BluetoothManager)
getSystemService(Context.BLUETOOTH_SERVICE);
bluetoothAdapter = bluetoothManager.getAdapter();
```

2.3 Turn on Bluetooth

Check whether Bluetooth is turned on by calling the isEnabled() method. If it returns false, the user is prompted to enable Bluetooth. code show as below:

```
if          (bluetoothAdapter         ==         null
|| !bluetoothAdapter.isEnabled())
  {
            Intent enableBtIntent = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENAB
                  LE);
startActivityForResult(enableBtIntent,     REQUEST_
ENABLE_BT);
}
```

2.4 Search for Bluetooth devices

With Bluetooth enabled and enabled, call the startLeScan() method and pass in the BluetoothAdapter.LeScanCallback callback method as a parameter to search for Bluetooth.

```
private            BluetoothAdapter.LeScanCallback
leScanCallback =
new BluetoothAdapter.LeScanCallback() {
@Override
public void onLeScan(final BluetoothDevice device,
int rssi,
byte[] scanRecord)
}
}
```

2.5 Connect Bluetooth low energy devices

The first step in communicating with a Bluetooth low energy device is to connect to the device's GATT server. Connect to the GATT server of the device by calling the connecGatt() method, and it will return a BluetoothGatt instance, which can perform discovery, read, and write operations on Bluetooth low energy devices. After the connection is successful, use the onConnectionstateChange function in the Bluetooth Gatt Call Back interface in the callback function, and then call mBluetooth Gatt.discover Services() to obtain the service and determine whether those features can be notified, readable, or writable.

2.6 Receive GATT notification

When the Bluetooth low energy device sends out a notification, the application will set the feature notification through Notification, mBluetooth Gatt.setCharacteristic Notification(characteristic, true). When the feature on the device changes, the onCharacteristicChanged() method callback will be triggered.

2.7 Sending data

When the application sends data, the data can be sent by calling the mBluetoothGatt.write Charateristic (charateristic) method.

2.8 Turn off Bluetooth Low Energy

When we finish the Bluetooth low energy communication, we should call close() to release the resources and the entire Bluetooth low energy communication ends.

```
            Public void close()
                    {
            If (bluetoothGatt==null)
              {     Return ;   }
```

3. Conclusion

In this article, we introduced you to the Bluetooth low energy communication process under the Android platform. This communication process is simple to implement, and is convenient for developing applications related to Bluetooth low energy communication under the Android platform. With more and more smart wearable devices entering our lives at present, the smart wearable devices can be well controlled by using low-power Bluetooth through smart terminals equipped with Android systems. BLE communication under the Android platform will have broader application prospects.

References:
1. Wan Yan, Li Lili. Implementation of Bluetooth communication system based on Android and BLE [J]. Science and Technology Innovation and Application, 2018(16): 15-17.
2. Wang Yuepeng. Design and implementation of smart home control system based on BLE on Android platform [J]. Internet of Things Technology, 2016, 6(03): 81-82.
3. Cai Lu. Design and implementation of wrist pedometer based on Android and ARM platform BLE4.0 [D]. Guangxi Normal University, 2015.