

# 「音频路由」Android端蓝牙音频连接感知方法

## 一、背景

蓝牙耳机作为当下音频中极其常见的设备，感知到其与手机的连接并正确使用，在音频路由模块中是非常重要的。通常情况下（由于android设备品牌和型号众多，少数设备需要主动与系统交互），我们不需要关心媒体模式的设备连接，因为在该音频模式下具体使用的设备完全由系统决策，作为开发者无法干预。**因此，本文主要讨论的范围为通话模式下的蓝牙设备连接。**

Android 12对蓝牙权限实施重大变更，引入了 `BLUETOOTH_SCAN`、`BLUETOOTH_ADVERTISE` 和 `BLUETOOTH_CONNECT` 三种权限。对应的，蓝牙设备相关接口中与隐私关系密切的部分也与权限挂钩，产生了以API 31为分界线的差异。

在蓝牙音频设备连接中，`BLUETOOTH_CONNECT`权限非常重要。如果APP的target version升级到了31，但是没有对蓝牙权限变更进行适配，那便会形成无法使用蓝牙音频设备通话音频的结果。为此，本文将着重描述蓝牙连接的两个方面：

- 具备蓝牙权限时感知蓝牙连接的方法；
- 不具备蓝牙权限时感知蓝牙连接的方法。

## 二、方法一：广播感知

### 权限申请方法

权限申请的方法需要以Android 12为分界线。当版本小于12时，只需要在AndroidManifest.xml中声明`BLUETOOTH`权限即可；而当版本大于等于12时，除了要在AndroidManifest.xml中声明`BLUETOOTH_CONNECT`权限外，还需要动态弹窗申请。

- 版本 < 12

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

- 版本 ≥ 12

```
1 <uses-permission android:name="android.permission.BLUETOOTH_CONNECT" />

1 // Check and request permissions
2 if (ActivityCompat.checkSelfPermission(this,
    Manifest.permission.BLUETOOTH_CONNECT) != PackageManager.PERMISSION_GRANTED)
3 {
4     ActivityCompat.requestPermissions(this, new String[] {
5         Manifest.permission.BLUETOOTH_CONNECT }, userRequestCode);
6 }
7 // Handling permission request results
8 @Override
9 public void onRequestPermissionsResult(int requestCode, String[]
    permissions, int[] grantResults) {
10     if (requestCode == userRequestCode) {
11         if (grantResults.length > 0 && grantResults[0] ==
12             PackageManager.PERMISSION_GRANTED) {
13             // Permission granted
14         } else {
15             // Permission denied
16         }
17     }
18     return;
```

### 权限使用方法


要拥有较好的音频控制能力，我们需要感知音频设备的插拔，并响应相关的音频路由动作。前述权限的成功申请只能保证系统相关的广播具有抵达用户的能力，而为了接收到目标广播，还需要继承`BroadcastReceiver`类，在其中注册广播接收器并重载广播接收函数。

在常规音频场景中，常用的蓝牙相关广播有以下几种：

- `BluetoothAdapter.ACTION_STATE_CHANGED`：关于安卓机的蓝牙开关的广播

 Android 11及以下需要`BLUETOOTH`权限

- `BluetoothHeadset.ACTION_CONNECTION_STATE_CHANGED`：关于安卓机与具有通话音频蓝牙设备连接动作的广播

 Android 11及以下需要`BLUETOOTH`权限  
Android 12及以上需要`BLUETOOTH_CONNECT`权限

- [BluetoothA2dp.ACTION\\_CONNECTION\\_STATE\\_CHANGED](#): 关于安卓机与具有媒体音频蓝牙设备连接动作的广播

 Android 11及以下需要BLUETOOTH权限  
Android 12及以上需要BLUETOOTH\_CONNECT权限

- [AudioManager.ACTION\\_SCO\\_AUDIO\\_STATE\\_UPDATED](#): 关于蓝牙耳机sco状态变更的广播

 不需要蓝牙相关权限

在上述权限描述中，可以看出被android 12分隔开的是headset与a2dp两种蓝牙连接的广播。而这两个广播的意义在于，开发者可以感知到蓝牙音频设备的连接与断开，这对于通话模式在适当时机启动sco，以使用蓝牙音频设备进行采集和播放至关重要。具体的使用方法分为两步：

1. 使用context中的registerReceiver函数注册相关广播

```
1 context.registerReceiver(this, new
  IntentFilter(BluetoothHeadset.ACTION_CONNECTION_STATE_CHANGED));
```

2. 继承BroadcastReceiver并override其中的广播接收函数onReceive

通过onReceive中的intent可获取到蓝牙的连接状态，并根据连接/断开便可以转到路由处理模块中完成相应逻辑

```
1 public class BluetoothHeadsetConnectionReceive extends BroadcastReceiver{
2     // --- other contents --- //
3     // ...
4
5     // override onReceive
6     @Override
7     public void onReceive(Context context, Intent intent) {
8         if (intent == null) return;
9         String action = intent.getAction();
10        if (action == null) return;
11
12        if (action.equals(BluetoothHeadset.ACTION_CONNECTION_STATE_CHANGED)){
13            // --- Process bluetooth headset action --- //
14            // ...
15        }
16    }
17 }
```

**需要注意的是**，上述这个权限仅适用于HFP的连接状态感知，即蓝牙音频设备的通话音量。如果想要感知仅有A2DP蓝牙音频设备连接状态，即蓝牙音频设备仅具备媒体音量(如[无mic的蓝牙音箱](#)，[手动关闭通话音量的蓝牙耳机](#))，需要监听 [BluetoothA2dp.ACTION\\_CONNECTION\\_STATE\\_CHANGED](#)，由于处理方式类似，便不再赘述。

该广播的监听使用方法略有不同，原因在于媒体模式下的路由在正常情况下由系统自身控制，开发者无法介入。因此如果本身处于的音频模式为媒体模式，使用仅具备媒体音量的设备播放音频功能不会受到影响。

**但万一处于通话模式呢，比如常见的会议场景**？这就存在很大的问题。**举个例子**，如果用户在 T 时刻没有任何外接设备，音频路由走的是**内置播放器**；在 T+1 时刻连接上了一个**天猫精灵**，那么对用户而言，他期望 T+2 时刻用**天猫精灵**进行播放。但实际情况却不符合预期，音频这时仍从**内置播放器**中播出。此处的问题就在于，通话模式下，由于蓝牙音频设备仅具有媒体音量，那么通话模式下该设备就不会被使用。

针对上述问题，一种解决方案就是监听A2DP连接状态，并且在适当的时机切换到媒体模式，以保证用户能够用上预期的蓝牙音频设备进行播放。但此时由于系统限制，采集只能用内置麦克风，可能会存在采集声音小甚至是无声的问题，解决方案只能是告知用户有此方面的影响。

## 三、方法二：回调感知

部分APP会将targetSdkVersion升级到31及以上，但是并没有做好蓝牙权限适配，就存在着无法正常使用蓝牙通话音频的问题。因此，本节介绍第二种感知蓝牙连接的方法，该方法能够在未申请蓝牙连接权限的条件下，仍能在第一时间知道蓝牙音频设备的连接状态。该方法主要分为两步：

1. 继承 [AudioDeviceCallback](#)，并实现其中的设备连接通知函数 [onAudioDevicesAdded](#) 和设备断开通知函数 [onAudioDevicesRemoved](#)。回调处理函数可以获取到发生事件的所有设备，后通过type即可判断是否为蓝牙headset设备。若要处理a2dp设备，同理可得；

```
1 private class BluetoothConnectCallback extends AudioDeviceCallback {
2     public void onAudioDevicesAdded(AudioDeviceInfo[] addedDevices) {
3         for (AudioDeviceInfo device : addedDevices) {
4             if (device.isSource()) continue;
5
6             int deviceType = device.getType();
7             if (deviceType == AudioDeviceInfo.TYPE_BLUETOOTH_SCO){
8                 // --- Handle Bluetooth headset connection --- //
9                 // ...
10            }
11        }
12    }
13
14    public void onAudioDevicesRemoved(AudioDeviceInfo[] removedDevices) {
15        for (AudioDeviceInfo device : removedDevices) {
16            for (AudioDeviceInfo device : addedDevices) {
17                if (device.isSource()) continue;
18            }
19        }
20    }
21 }
```

```
19         int deviceType = device.getType();
20         if (deviceType == AudioDeviceInfo.TYPE_BLUETOOTH_SCO){
21             // --- Handle Bluetooth headset disconnection --- //
22             // ...
23         }
24     }
25 }
26 }
```

2. 使用 `AudioManager.registerAudioDeviceCallback` 接口将 `AudioDeviceCallback` 实现子类注册到系统。

```
1 mBluetoothDeviceCallback = new BluetoothConnectCallback();
2 mAudioManager.registerAudioDeviceCallback(mBluetoothDeviceCallback, null);
```

#### 🔥 【注意事项】

1. 正常情况下，不推荐使用方法二，应该使用方法一正常申请权限
2. 使用方法二时，不要使用需要蓝牙连接权限的相关接口，否则会产生无法预期的后果（比如鸿蒙设备获取设备名会出现crash）
3. 方法二系统会多出一个线程 `AudioPortEventHandler`
4. 方法二可能存在蓝牙采集的隐私问题，这点要非常重视
5. 方法二google可能会在未来某次更新中受到限制，因此存在功能突然失效的问题

## 四、Android 12权限申请说明

前提条件：测试用Android 12手机（OPPO find x5 pro + Redmi Airdot3）

### 结论

1. 若 `targetSdkVersion ≥ 31`，"附近的设备"权限是否在系统中出现由是否静态声明 `BLUETOOTH_CONNECT` 权限决定，且系统不会默认将权限设为允许【对比row 3、4、7、8】
2. 若 `targetSdkVersion < 31`，"附近的设备"权限是否在系统中出现由是否静态声明 `BLUETOOTH` 权限决定，且系统会默认允许该权限；  
但能够真正接收到蓝牙连接广播是否静态声明 `BLUETOOTH_CONNECT` 权限决定，若声明了该权限则无法收到蓝牙连接广播，需要手动关闭后开启权限后才有效【对比row 1、5】
3. `compileSdkVersion` 不影响权限行为【对比row 1、3、5、7、9、11、13、15】

### 推荐适配方法

1. 若适配了 android 12 及以上，即 `targetSdkVersion ≥ 31`，若需要正常获取蓝牙连接的广播，则必须主动弹窗申请"附近的设备"权限
2. 若未适配 android 12 及以上，即 `targetSdkVersion < 31`，若需要正常获取蓝牙连接的广播，则不能静态申请 `BLUETOOTH_CONNECT` 权限，而必须仅申请 `BLUETOOTH` 权限

	A	B	C	D	E	F	G	H	I
1		bluetooth权限	bluetooth_connect权限	compileSdkVersion	targetSdkVersion	是否显示"附近的设备权限"	是否系统默认允许"附近的设备权限"	若默认显示有权限，是否有蓝牙连接广播	备注
2	1	√	√	≤ 30	≤ 30	是	是	不能	需要手动关闭再开启权限才能有connect通知
3	2	√	√	≤ 30	≥ 31		√		
4	3	√	√	≥ 31	≤ 30	是	是	不能	需要手动关闭再开启权限才能有connect通知
5	4	√	√	≥ 31	≥ 31	是	否	不满足条件	
6	5	√	×	≤ 30	≤ 30	是	是	能	
7	6	√	×	≤ 30	≥ 31		√		
8	7	√	×	≥ 31	≤ 30	是	是	能	
9	8	√	×	≥ 31	≥ 31	否	否	不满足条件	
10	9	×	√	≤ 30	≤ 30	是	否	不满足条件	
11	10	×	√	≤ 30	≥ 31		√		
12	11	×	√	≥ 31	≤ 30	是	否	不满足条件	
13	12	×	√	≥ 31	≥ 31	是	否	不满足条件	
14	13	×	×	≤ 30	≤ 30	否	否	不满足条件	
15	14	×	×	≤ 30	≥ 31		√		
16	15	×	×	≥ 31	≤ 30	否	否	不满足条件	
17	16	×	×	≥ 31	≥ 31	否	否	不满足条件	