



ZS110A Android 语音遥控器 patch 指南
发布 **1.0.0**

2019 年 04 月 12 日

1	文档介绍	1
1.1	文档目的	1
1.2	术语说明	1
1.3	参考文档	1
1.4	版本历史	2
2	适配语音遥控器的 android 端补丁介绍	3
2.1	语音通路流程补丁	3
2.2	语音按键映射补丁	3
2.3	android bluedroid 补丁	4
2.4	其它补丁	4
3	ble audio 补丁	5
3.1	ble audio 补丁原理介绍	5
3.2	sdk 固件编译的修改步骤	5
3.3	小机上使用命令方式快速验证补丁步骤	6
3.4	补丁常见问题分析	7
4	android voice key 补丁	10
4.1	android 语音键适配	10
4.2	如何确认打上补丁	10
5	android bluedroid 补丁	12
5.1	android 定向广播回连支持补丁	12
5.2	android 盒子掉电不回连问题修复补丁	12
5.3	bludroid ble srv instance 错误修复补丁	12
5.4	android hid 快速配对回连补丁	13
5.5	android bluedroid 配置推荐选补丁	13
6	bt module 补丁	14
6.1	模组适配补丁	14

1.1 文档目的

介绍如何在 android 盒子端适配 ZS110A 语音遥控器。

1.2 术语说明

表 1.1: 术语说明

术语	说明
ZEPHYR	为所有资源受限设备，构建了针对低功耗、小型内存微处理器设备而进行优化的物联网嵌入式小型、可扩展的实时操作系统（RTOS）
ZS110A	基于 ZEPHYR 建立的，用于在 ACTIONS ATB110X SOC 上开发 BLE 应用的 SDK

1.3 参考文档

- <http://docs.zephyrproject.org/>

1.4 版本历史

表 1.2: 版本历史

日期	版本	注释	作者
2019-03-27	1.0	初始版本	ZS110A 项目组

适配语音遥控器的 android 端补丁介绍

- 语音通路流程补丁
- 语音按键映射补丁
- android bluedroid 补丁
- 其它补丁

2.1 语音通路流程补丁

- 语音流程目录下的补丁必须打，否则遥控器的语音功能无法在 android 盒子端使用
- 打此补丁的流程，请参考 ble audio 补丁章节
- 打上此补丁后，使用 voice_pro.apk 等录音 APK 能够录制遥控器端的语音数据

2.2 语音按键映射补丁

- 在 android 盒子端有 2 个 android 键可以用来触发语音助手，搜索键 (KEYCODE_SEARCH), 语音助手键 (KEYCODE_VOICE_ASSIST)
- 如果使用语音助手键触发语音助手，HID 没有标准键值对应，当前遥控器端配置为 020c 扩展键，需要打此补丁做一个映射
- 如果使用搜索键触发语音助手，HID 有标准键值 0x0221 对应，遥控器端适配，请参考 rmc_atb1103_atv.h，其它版型需要配置此键，请配置上 CONFIG_ANDROID_TV_BOX

2.3 android bluedroid 补丁

- 使用 public addr 的回连的补丁: 用于支持定向广播包的回连, 原生 bluedroid 默认不支持定向广播的回连, 如系统已支持, 请忽略
- ble srv instance 错误的补丁: 用于修复配对连接成功, 但遥控器无法操作的 bug, 这个 android5.1 以上版本提取的官方补丁, 如系统已支持, 请忽略
- 掉电配对信息丢失补丁: 用于修复盒子端掉电回连 bug, 在 android 5.1 上, 盒子异常掉电, 盒子端的加密信息有丢失现象, 如系统已支持, 请忽略

2.4 其它补丁

- 模组适配方面, 在适配 ap6212 模组过程有发现开关蓝牙异常崩溃, 如有遇到此类似问题, 请更新 ap6212 模组到最新驱动

3.1 ble audio 补丁原理介绍

ble audio 补丁原理是利用 hidraw 节点捕捉协议栈发送的语音数据，目前 Android Blueroid 将 ble 语音数据和按键信息通过 hid 发送出去，通过建立 hidraw 节点，可以从中捕捉到语音数据。目前通过 ble hal 实现从 hidraw 中读取遥控器语音数据，在 Android 框架层上就通过配置文件将 ble hal 导入到音频框架中，并通过绑定 Android 原生已有的耳麦设备来完成 audio 音频策略选择，通过 apk 检测 ble 连接状态，通知 audio 服务耳麦设备的状态就可以使得录音通路切换至 ble hal，实现从 ble 获取录音数据功能

3.2 sdk 固件编译的修改步骤

- 打上 kernel.patch

开启 hidraw 配置，这样才可以通过该节点获取音频数据

- 打上 device.patch

由于 hal 需要访问 hidraw 节点，需要设置节点的权限，这样才能正常访问。

1. 添加 hidraw 访问权限
2. 添加 selinux 策略

注解： 如果有开启 selinux，也需要设置相应的 sepolicy 配置，才能使对应的进程有读写的权限。通过小机上使用命令 `adb shell getenforce` 查

看当前 selinux 模式。

```
Jetstream:/ # getenforce
getenforce
Enforcing//强制模式, 需要添加 sepolity 策略补丁
```

- 打上 frameworks_av.patch 补丁修改 audio 配置
 1. 老配置方式修改 audio_policy.conf
 2. 新方式修改 xml 文件
- 版型目录下 device.mk 新增选项拷贝 libasc_dec.so 到 /vendor/lib/ 并拷贝 audio.stub.default.so 到 /vendor/lib/hw/. 确保库权限。64 位系统拷贝对应的 64bit so
- 版型目录下 device.mk 新增选项拷贝 HidAudio.apk 到 /vendor/app/hidaudio

3.3 小机上使用命令方式快速验证补丁步骤

- 打上 kernel.patch 开启 hidraw 配置, 这样才可以通过该节点获取音频数据。配置了 hidraw 后, 更新小机的 kernel.img 内核镜像文件。如果该配置已经打开, 这一步可以跳过
- 按 frameworks_av.patch 修改小机上的配置文件, 如果 /vendor/etc/ 目录下存在 patch 中提到的任意配置文件, 则修改该目录下文件, 否则修改 /etc/ 或 /system/etc (/etc 为 /system/etc 的链接) 目录下文件
- 拷贝 libasc_dec.so 到 /vendor/lib/ 并拷贝 audio.stub.default.so 到 /vendor/lib/hw/. 确保库权限。64 位系统拷贝对应的 64bit so
- 在 /vendor/app/ 目录下新建 hidaudio, 拷贝 HidAudio.apk 到 /vendor/app/hidaudio
- 重启小机
 - 连接上 ble 后, 设置 /dev/hidraw* 节点权限

```
chown system:audio /dev/hidraw*
chmod 660 /dev/hidraw*
root@AD700A:/ # ls -l /dev/hid*
ls -l /dev/hid*
crw-rw---- system audio    248,   0 2015-01-01 08:00 hidraw0
```

- 如果 selinux 模式为 Enforcing, 设置 selinux 模式为 Permissive, 关闭强制模式

```
Jetstream:/ # setenforce 0
setenforce 0
Jetstream:/ # getenforce
```

(continues on next page)

(续上页)

```
getenforce
Permissive
```

- kill 掉 audioserver, 重启 audio 进程。之后可以使用录音 apk 录音

3.4 常见问题分析

- 补丁如若不能使用首先检查节点和权限, 正常节点权限如下:

```
ls -l /dev/hidraw*
crw-rw---- 1 system audio 241,  0 2018-12-18 13:42 /dev/
↳hidraw0
audio 用户组有读写权限。
```

- 如果 selinux 模式为 Enforcing, 可以通过 logcat 搜索 avc 关键字。有如下类似提示则为异常, 提示进程没有权限, 检查 sepolicy 是否设置正常

```
avc: denied { read } for name="/" dev="tmpfs" ino=6145
↳scontext=u:r:mediaserver:s0 tcontext=u:object_r:device:s0
↳tclass=dir permissive=0
/* Android 5.0 和 6.0 版本, audio hal 被 mediaserver 进程加载
↳*/
avc: denied { read } for name="/" dev="tmpfs" ino=8125
↳scontext=u:r:audioserver:s0 tcontext=u:object_r:device:s0
↳tclass=dir permissive=0
/* Android 7.0 版本, audio hal 被 audioserver 进程加载 */
avc: denied { read } for name="hidraw" dev="sysfs" ino=16332
↳scontext=u:r:hal_audio_default:s0 tcontext=u:object_
↳r:sysfs:s0 tclass=dir permissive=0
/* Android 8.0 和 9.0 版本, audio hal 被 android.hardware.
↳audio@2.0-service 进程加载 */
```

- 检查 audio 的配置, 打上 patch 后, 首先确认小机上文件是否有修改到, 目前文件可能位于/vendor/etc 或/system/etc 目录下, 其中/vendor/etc 下的配置文件是优先解析的。确保文件无误后, 通过 dumpsys media.audio_policy 查看 ble hal 是否正常加载。以下是相关说明:

```
AudioPolicyManager: 0xf20c5200
Command Thread: 0xf20af140
Tones Thread: 0xf20af020
...
- Available input devices:
  Device 1:
  - id: 3
  - type: AUDIO_DEVICE_IN_BUILTIN_MIC
  - Profiles:
    Profile 0:
```

(continues on next page)

(续上页)

```

- format: AUDIO_FORMAT_PCM_16_BIT
- sampling rates:8000, 11025, 12000, 16000,
↪ 22050, 24000, 32000, 44100, 48000
- channel masks:0x000c, 0x0010

Device 2:
- id: 20
- type: AUDIO_DEVICE_IN_WIRED_HEADSET //对应的数值是
0x80000010
- name: RemoteDM1204
- Profiles:

```

注解: Available input devices 指示当前可用设备, 目前 ble hal 是和 AUDIO_DEVICE_IN_WIRED_HEADSET 设备绑定 (4.4 补丁中是和 AUDIO_DEVICE_IN_BUILTIN_MIC 设备绑定), 如果需要录音走 ble hal, AUDIO_DEVICE_IN_WIRED_HEADSET 设备必须出现在可用设备中, 如果没有, 就可能是补丁中 hidaudio.apk 的问题。4.4 补丁中绑定的是默认的 AUDIO_DEVICE_IN_BUILTIN_MIC 设备, 不需要 hidaudio.apk。

```

HW Modules dump:
...
- HW Module 5:
- name: ble
- handle: 42
- version: 2.0
- inputs:
  input 0:
  - name: stub
  - Profiles:
    Profile 0:
    - format: AUDIO_FORMAT_PCM_16_BIT
    - sampling rates:8000, 11025, 12000, ↪
    ↪16000, 22050, 32000, 44100, 48000
    - channel masks:0x000c, 0x0010
  - flags: 0x0000 (AUDIO_INPUT_FLAG_NONE)
  - Supported devices:
    Device 1:
    - type: AUDIO_DEVICE_IN_WIRED_HEADSET

```

注解: 上面信息是确保配置文件中添加的值有被正常解析, 目前我们添加了一个 ble 新 hal, 绑定设备 AUDIO_DEVICE_IN_WIRED_HEADSET, 并且支持如上采样率和格式。正常解析后 audio 进程起来后会寻找 audio.ble(stub).XXX.so 库来加载, 上述中 handle 为 0 表示加载库过程中失败。可能是库存在不兼容问题。

```
Inputs dump:
- Input 86 dump:
  ID: 24
  Sampling rate: 16000
  Format: 1
  Channels: 00000010
  Devices 80000010
  Audio Sessions:
```

注解: 录音时 `dumpsys media.audio_policy` 可以看到录音的 input 线程信息，主要是看当前录音设备是否是 ble 所绑定设备。这里就是 80000010，也就是 `AUDIO_DEVICE_IN_WIRED_HEADSET`，当前录音通路正常。

- 检测 apk 是否正常，hidaudio.apk 主要是检测 ble 设备连接状态的，检测到状态变化时会通知 audio 进程音频设备状态变化，这样录音时可以做到动态切换。该 apk 会在后台一直监测，启动后可以查看是否有如下进程在后台。

```
u0_a44    1591  128   1213188 26604 ffffffff 00000000 S com.
↳ actions.hidaudio
```

android voice key 补丁

4.1 android 语音键适配

在 android 盒子端有 2 个 android 键可以用来触发语音助手，搜索键 (KEYCODE_SEARCH), 语音助手键 (KEYCODE_VOICE_ASSIST)

- 如果使用语音助手键触发语音助手，HID 没有标准键值对应，当前遥控器端配置为 020c 扩展键，需要打此补丁做一个映射
- 如果使用搜索键触发语音助手，HID 有标准键值 0x0221 对应，遥控器端适配，请参考 rmc_atb1103_atv.h，其它版型需要配置此键，请配置上 CONFIG_ANDROID_TV_BOX

注解：在 android google tv 已经 (HID: page 0x0C, command: 0x221) 作为触发语音助手键，建议采用此方式，如果采用此方式，也就不需要打补丁。

4.2 如何确认打上补丁

- 通过 `getevent -p` 应该会看到 hid 已经支持的键值

```
shell@AD700A:/ $ getevent -p
getevent -p
add device 1: /dev/input/event4
  name:      "BLE_YT_RMC"
  events:
      KEY (0001): 0001  001c  004a  004e  0067  0069  006a
      ↪ 006c
```

(continues on next page)

(续上页)

↪ 0074 0077 0080	006e 006f 0071 0072 0073↵
↪ 0088 0089 008a	0082 0083 0085 0086 0087↵
↪ 0098 009b 009c	008b 008c 008e 0090 0096↵
↪ 00a5 00a6 00a7	009e 009f 00a1 00a3 00a4↵
↪ 00ae 00b0 00b1	00a8 00a9 00ab 00ac 00ad↵
↪ 00d0 00d1 00d2	00b2 00b5 00b6 00ce 00cf↵
↪ 00e0 00e1 00e4	00d4 00d8 00d9 00db 00df↵
↪ 00f1 00f4 0100	00e7 00e8 00e9 00ea 00eb↵
↪ 0106 0107 0108	0101 0102 0103 0104 0105↵
↪ 0113 0114 0161	0109 010a 0110 0111 0112↵
↪ 0174 0176 0178	0162 0166 016a 016e 0172↵
↪ 017f 0180 0182	0179 017a 017b 017c 017d↵
↪ 018d 018e 018f	0183 0185 0188 0189 018c↵
↪ 0197 0198 0199	0190 0191 0192 0193 0195↵
↪ 01a3 01a4 01a5	019a 019c 01a0 01a1 01a2↵
↪ 01ab 01ac 01ad	01a6 01a7 01a8 01a9 01aa↵
↪ 01ba 0240 0241	01ae 01af 01b0 01b1 01b7↵
↪ 0250 0251	0242 0243 0244 0245 0246↵
REL (0002): 0000 0001 0006	
ABS (0003): 0020 : value 0, min 0, max 668, fuzz 0,↵	
↪ flat 0, resolution 0	
MSC (0004): 0004	

- 通过 getevent 可以看到按键信息

/dev/input/event4: EV_MSC	MSC_SCAN	000c020c
/dev/input/event4: EV_KEY	0246	DOWN
/dev/input/event4: EV_SYN	SYN_REPORT	00000000

5.1 android 定向广播回连支持补丁

一般 android 默认情况下，不支持定向广播回连，要支持定向广播回连，需要增加此补丁。

- 非定向广播回连速度会慢一些
- 定向广播回连速度会快一些

注解： 当前遥控器配置的定向广播做回连，这个补丁需要打上。

5.2 android 盒子掉电不回连问题修复补丁

- 在 android 5.1 参考平台上，做遥控器回连测试时，发现盒子和遥控器进行连接完成后，然后盒子掉电，发现盒子的配对信息丢失，无法回连上，如遇到此问题，请打上此补丁。
- 补丁的原理是 bluedroid 原生设计会将配对信息延迟写入文件系统，打上补丁后，配对及 hid profile 储存信息会及时写入文件系统

5.3 bludroid ble srv instance 错误修复补丁

- 在 android 5.1 参考平台上，做遥控器反复配对连接测试时，发现盒子和遥控器进行配对连接后，遥控器无法操作，通过 adb logcat 发现 uhid 节点没有创建成功，

导致无法操作, 如遇到此问题, 请打上此补丁.

注解: 该补丁是从 android 高版本拿过来的, 如发现系统已经打上此补丁, 请忽略。

5.4 android hid 快速配对回连补丁

- 在 android 5.1 参考平台上, profile 遥控器和盒子的配对连接性能时, 发现配对完, 要延迟 5S 才能做 hid profile 的连接, 导致整个过程比较慢
- 将 stack/smp/smp_int.h 中的 #define SMP_WAIT_FOR_REL_DELAY_TOUT 5 从 5S 修改为 1S

注解: android8.x 已经修改为 500ms

5.5 android bluedroid 配置推荐选补丁

1. 修改 stack/include/btm_ble_api.h 中连接参数

```
#define BTM_BLE_CONN_INT_MIN_DEF      24      /* recommended ↵
↵min: 30ms = 24 * 1.25 */
#define BTM_BLE_CONN_INT_MAX_DEF      40      /* recommended ↵
↵max: 50 ms = 56 * 1.25 */
```

注解: 默认的连接间隔偏大, 在回连过程中, 如果 slave 的连接参数请求没有更新过来, 立马进行语音操作, 会导致回连初始阶段语音性能不足, 影响体验

建议修改为 10 ~ 12 即 12.5 ms ~ 15 ms

```
#define BTM_BLE_CONN_TIMEOUT_DEF      2000
```

注解: 默认的连接超时时间为 20S, 对 BLE 连接的稳定性有帮助, 但是在遥控器掉电回连过程中, 有可能出现很久才能回连上现象, 如上次连接参数 timeout 是 20s, 遥控器端也没有请求更新合适的连接参数, 遥控器立即掉电, 盒子端要等 20s 才能超时断开进而进行回连, 比较影响回连体验。

建议修改为 400 即 4S。

6.1 模组适配补丁

模组适配方面，在适配 ap6212 模组过程有发现开关蓝牙异常崩溃，如有遇到此类似问题，请更新 ap6212 模组到最新驱动

List of Figures

List of Tables

1.1	术语说明	1
1.2	版本历史	2