

**肖克利语音遥控器Android 6.0**

**移植文档**

版本号：1.0.2

2018-08-02

**目录**

[1 系统架构 3](#_Toc469798776)

[2 移植需求 4](#_Toc2001358346)

[3 音频HAL移植 5](#_Toc2138182625)

[4 Audio Policy移植 6](#_Toc1287216258)

[4.1 安卓6.0移植 6](#_Toc1695944783)

[4.2 Configuration文件修改 9](#_Toc265216632)

[5 开机启动项修改 10](#_Toc1727254345)

[6 蓝牙连接 11](#_Toc1179118118)

[7 版本信息 12](#_Toc1945857949)

[8 联系方式 13](#_Toc735876688)

# 系统架构

语音识别助手所调用的音频接口为是安卓通用的音频通道。我司的遥控器语音系统架构为:

HID驱动

sklrm Audio HAL

Userspace

Kernel

Audio Flinger

Framework

/dev/hidraw\*

Audio Policy

当使用语音助手时的时候，audioflinger中的audiorecorder调取相对应遥控器的音频HAL库，语音数据通过音频HAL接口直接从HIDRAW节点读取语音数据。

另外语音遥控器也可以支持录音机、微信等使用标准安卓语音通道输入的app。

# 移植需求

在移植之前需要机顶盒端相关平台的系统源码和编译环境支持，并获取root权限。现在的移植的安卓平台为安卓6.0。语音遥控器支持mstar，amlogic，MTK等大部分平台。

根据语音通道的系统架构需要对以下部分进行移植：

Audio HAL

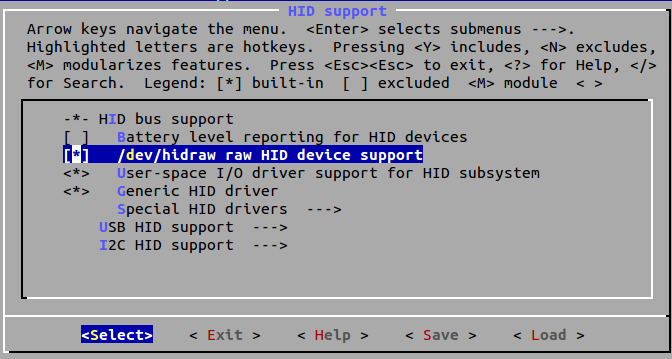
Audio Policy

Audio HAL位于安卓的 Userspace用户空间层，直接放入已经编译完成的文件即可；Audio Policy位于Framework固件层，需要修改源码然后重新编译，然后把相应的库刷入机顶盒。

首先打开终端，连接上设备, 可以使用adb调试指令，adb root进入root模式，使用adb remount进行重新挂载；也可以使用串口直接在设备下进行调试，使用su进入root模式，使用mount -o remount,rw /system进行重新挂载。

该语音方案基于HID profile，需要在移植前确认我司遥控器在安卓设备读取到的Product ID和 Vendor ID，并告知我司，以便与我司对软件做相应的修改。

\*此外，首先要检查当前机顶盒HIDRAW功能是否可以正常操作，尤其是在使用amlogic平台时候，该功能经常会被关闭。在HIDRAW生效时，将遥控器连上以后，安卓设备的/dev目录下会生成新的hidraw节点。如果没有相对应的节点生成，请先检查当前使用内核的menuconfig是否配置hidraw功能生效。Hidraw配置在Device Driver->HID support下，确认[]raw HID device support前打上星号。如下图所示：



另外也可以在当前内核的config文件添加，路径为“arch/arm/内核版本”：

CONFIG\_HIDRAW=y

修改后重新编译内核刷入机顶盒重启后即可生效。

# 音频HAL移植

语音通道需要音频hal来实现具体语音调用方法。在新的音频HAL生效时，系统的语音输入通道源会变成遥控器，此时语音数据即可传入安卓的通用语音通道。

此语音库默认的Vendor ID：0004和Product ID：0000，如果厂商提供的PID和VID不相同的话，需要我司重新编译语音库。

在终端输入指令：

adb push 路径/audio.sklrm.default.so /system/lib/hw

把遥控器对应的语音hal层库audio.sklrm.default.so：

E3C8.tmp

放在/system/lib/hw下。

例如：

385B.tmp

然后在该目录下修改文件的权限属性，输入指令：

chmod 666 路径/audio.sklrm.default.so

完成后重启盒子。

如果需要查看该库的log信息，可以使用指令：

logcat -s audio\_hw\_sklrm

# Audio Policy移植

在完成了虚拟声卡驱动和音频hal的移植，如果需要让新建的sklrm hal库生效，还需对audio policy进行修改。

## 安卓6.0移植

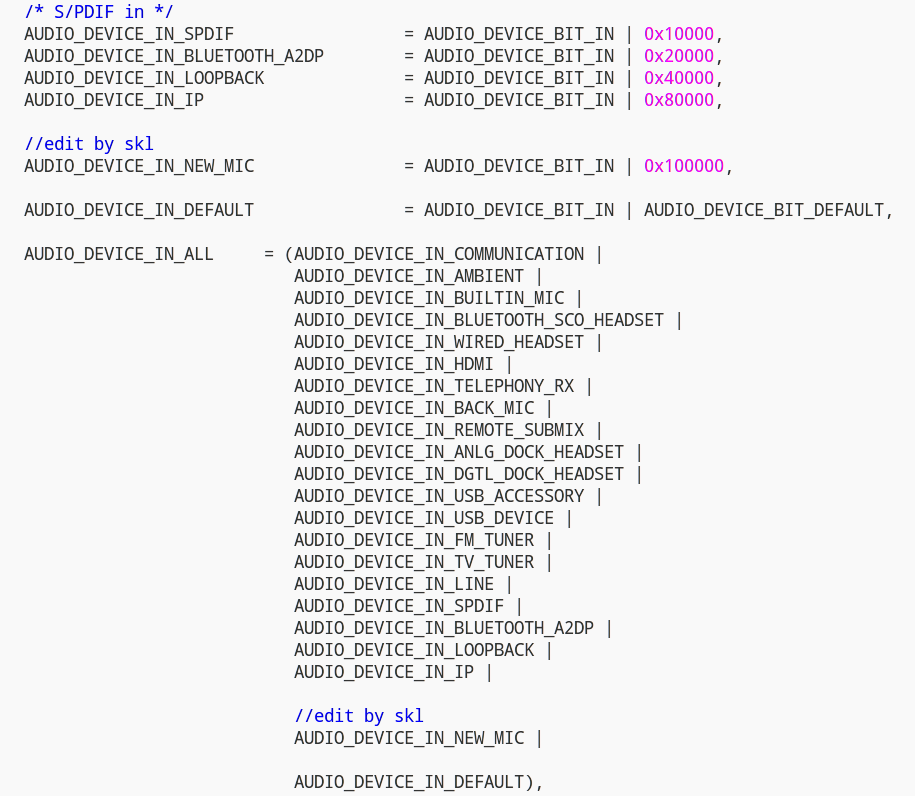
首先在源码下/system/media/audio/include/system/audio.h中添加新的输入设备：

AUDIO \_DEVICE\_IN\_NEW\_MIC 信息，并给该输入设备分配新的设备号。

在audio\_devices\_t 结构体中添加

AUDIO\_DEVICE\_IN\_NEW\_MIC = 0xXXXX，

AUDIO\_DEVICE\_IN\_ALL     = （...| AUDIO\_DEVICE\_IN\_NEW\_MIC ),



安卓6.0将audio policy的目录进行了重构，整个service的默认路径为：

/frameworks/av/services/audiopolicy。

在打开音频通道时，选择音频输入主要调用的是getDeviceForInputSource()该函数来获取音频设备的输入源，该函数现在位于以下位置：

/frameworks/av/services/audiopolicy/enginedefault/src/Engine.cpp

在此根据不同的输入需求添加获取遥控器连上property的代码：

char value[PROPERTY\_VALUE\_MAX];

int prop\_rm;

property\_get("audio.in.device.sklrm", value, NULL);

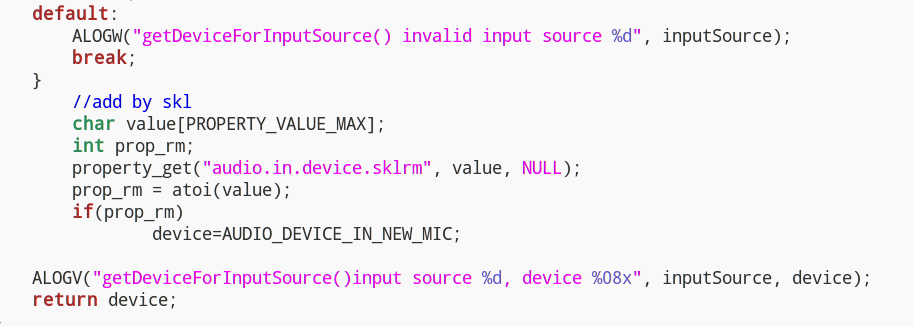
prop\_rm = atoi(value);

if(prop\_rm)

device=AUDIO\_DEVICE\_IN\_NEW\_MIC;

该设备判断可以添加在不同的音频输入源下，如果需要所有的音频输入源生效可以在如

图位置完成添加：



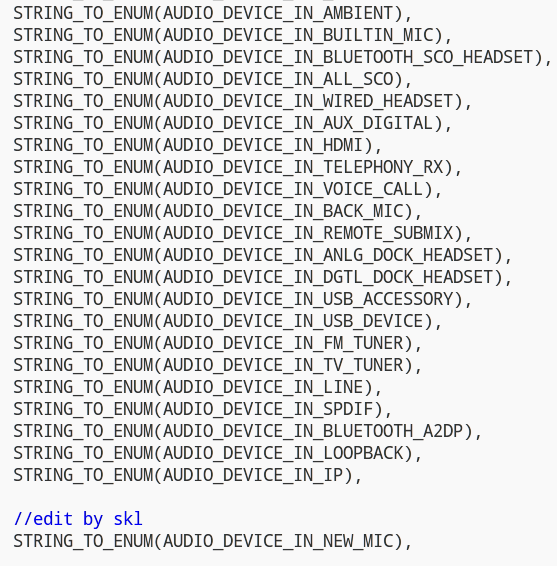
在安卓6.0中，sDeviceNameToEnumTable的结构体被移到以下位置：

/frameworks/av/services/audiopolicy/common/managerdefinitions/include/ConfigParsingUtils.h

同样如下添加行：

STRING\_TO\_ENUM(AUDIO\_DEVICE\_IN\_NEW\_MIC),

如图所示：



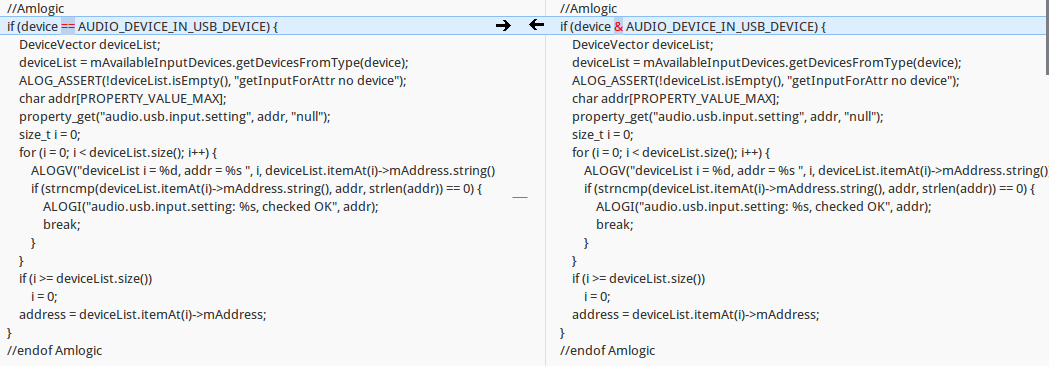
\*另外，在进行amlogic平台移植时，会发生和audio usb冲突而无法正常初始化音频输入的问题，需要对amlogic的部分代码进行修改，才可以正常使用。修改的文件名为AudioPolicyManager.cpp，路径为：

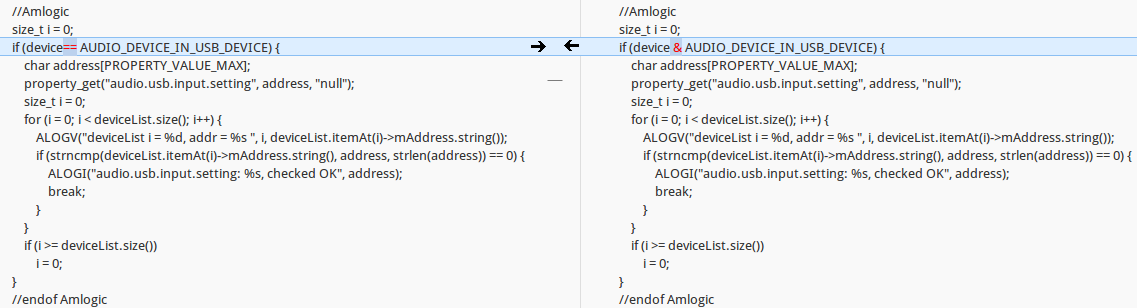
/frameworks/av/services/audiopolicy/managerdefault/AudioPolicyManager.cpp

找到getInputForAttr(...)和setInputDevice(...)这两个与输入设备相关的函数，在amlogic添加的部分中，将判断device是否为AUDIO\_DEVICE\_IN\_USB\_DEVICE的方式从and改为equal：

if (device & AUDIO\_DEVICE\_IN\_USB\_DEVICE) =>if (device == AUDIO\_DEVICE\_IN\_USB\_DEVICE)

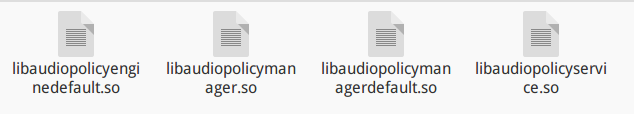
如下图进行修改即可：





按照此图进行修改，左边为修改后的代码，右边为修改前的代码，在修改完成后将会避免该问题的发生。

然后重新编译整个audiopolicy的目录，根据系统需要使用32位或者64位的编译方式，生成以下新的库：



将以下新编译生成的库放在/system/lib下，如果是64位的库则放在/system/lib64下，在终端输入指令：

adb push 路径/libaudiopolicyenginedefault.so /system/lib

adb push 路径/libaudiopolicymanager.so /system/lib

adb push 路径/libaudiopolicymangerdefault.so /system/lib

adb push 路径/libaudiopolicyservice.so /system/lib

即可。

## Configuration文件修改

另外audio policy还需要修改系统下的audio\_policy.conf:

DD6B.tmp

该文件一般位于/device/厂商/路径下，把新建的sklrm模块的信息添加进去然后保存。

sklrm {

inputs {

sklrm {

sampling\_rates 16000

channel\_masks AUDIO\_CHANNEL\_IN\_MONO

formats AUDIO\_FORMAT\_PCM\_16\_BIT

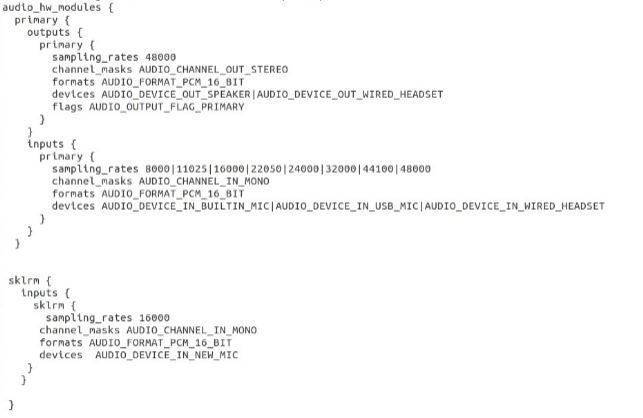
devices AUDIO\_DEVICE\_IN\_NEW\_MIC

}

}

}

如图添加完：



然后把更新后的audio\_policy.conf放在/system/etc下，在终端输入指令：

adb push 路径/audio\_policy.conf /system/etc

或者重新编译镜像代码并刷入。

# 开机启动项修改

在调用语音库的时候，会对hidraw节点进行操作，需要先修改该节点的权限，否则会发生权限被拒绝的问题，具体操作为对系统文件ueventd.rc进行修改。

在ueventd.rc下添加：

/dev/hidraw\* 0666 root root

另外要在开机时候自动让语音库生效需要设置属性，在开机启动项init.rc中添加property属性：

setprop audio.in.device.sklrm 1

或者修改启动的.prop文件，添加：

audio.in.device.sklrm = 1

在完成对该两文件的修改后，重新编译生成boot.img或者整个镜像文件刷入，完成以上所有步骤后，重启机顶盒，语音识别移植即可成功。

# 蓝牙连接

一般情况下，如果需要正常的语音使用，遥控器端与平台端蓝牙的连接参数需要互相匹配。平台端的连接参数文件为 btm\_ble\_api.h。

安卓6.0的路径为： /system/bt/stack/include/btm\_ble\_api.h

需要修改最小和最大连接interval，从机的latency以及timeout。

其中我司建议修改以下数值：

#define BTM\_BLE\_CONN\_INT\_MIN 0x0006

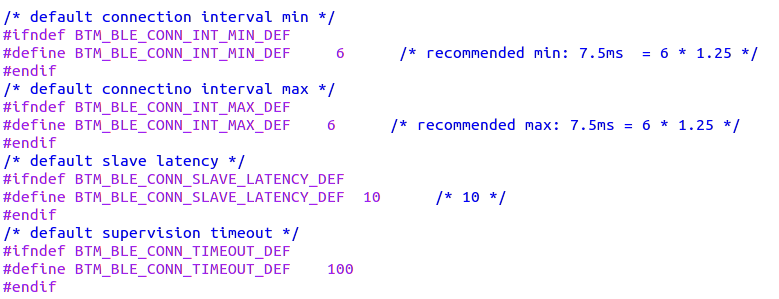
#define BTM\_BLE\_CONN\_INT\_MIN\_DEF 6

#define BTM\_BLE\_CONN\_INT\_MAX\_DEF 6

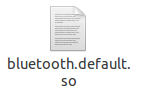
#define BTM\_BLE\_CONN\_SLAVE\_LATENCY\_DEF 10

#define BTM\_BLE\_CONN\_TIMEOUT\_DEF 100

修改后如图所示：



然后重新编译生成新的蓝牙库文件：



在终端输入指令：

adb push 路径/bluetooth.default.so /system/lib/hw

把新编译生成的蓝牙库bluetooth.default.so放在/system/lib/hw下重启后即可。

# 版本信息

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **日期** | **作者** | **描述** |
| 1.0.0 | 2018-08-02 | 李通越 | 初始版本；Android4.4语音移植; |
| 1.0.1 | 2018-08-02 | 李通越 | 新增加Android5.0语音移植; |
| 1.0.2 | 2018-08-02 | 李通越 | 新增加Android6.0语音移植; |

# 联系方式

上海肖克利信息科技股份有限公司

网址：http://www.shockley-elect.com

座机：021-64327686

联系人：Geroge

邮箱：geroge@shockley-elect.com