

**Voice Control For Android 6.0**

**Porting Guide**

Version 1.0.0

2017-07-19

**Content**

[1 System Architecture 3](#_Toc749022684)

[2 Porting Requirement 4](#_Toc1570058402)

[3 Audio HAL Porting 5](#_Toc266921514)

[4 Audio Policy Porting 6](#_Toc1934875909)

[5 Bluetooth Parameter 12](#_Toc369711461)

[6 Version Information 14](#_Toc1896110417)

[7 Contact Information 15](#_Toc1984960052)

# System Architecture

The audio interface called by the voice recognition assistant is general audio interface of Android, which is used audio recorder here:

HID Driver

sklrm Audio HAL

Userspace

Kernel

Audio Flinger

Framework

/dev/hidraw\*

Audio Policy

When using voice assistant, audiorecorder call the remote audio HAL library audiorecorder, voice data is read directly from the HIDRAW node by audio HAL interface.

# Porting Requirement

Before porting, the android source code and compiler environmental of set-top box platforms should be supported , and access to root permissions. The audio system architecture is included the following parts:

a. Audio HAL

b. Audio Policy

According to the audio system, audio Hal are in the user space layer, the compiled files are directly pushed into the box; audio policy in the firmware layer and need to to modify the source code and recompile, then the corresponding library is pushed into the set-top box.

Before start the porting, android debug-tool box must be prepared. This porting guide is used for android 6.0. First open a terminal and connect the device, use the command ‘adb root’ to enter root mode, and use the command ‘adb remount’ to remount the box and access to push and pull files with device. As shown as following, it means success.

# Audio HAL Porting

When completing installation of virtual sound card driver, it needs to add a corresponding audio Hal to achieve the specific functions of the virtual sound card. When the new audio Hal is active, the audio input channel source will become the remote. This voice data is flowing through android general audio channel right now.

Enter the command line in the terminal:

adb push ‘PATH’/audio.sklrm.default.so /system/lib/hw

Push the corresponding audio HAL library ‘audio.sklrm.default.so’ into the path: /system/lib/hw and then reboot the box to act this library:

E3C8.tmp

When calling the voice library, it need to modify the permissions of the hidraw node, otherwise it will be shown: permission denied. To achieve the permission, the device system file ueventd.rc should be modified.

Add the line in file ueventd.rc

/dev/hidraw\* 0666 root root

If it wants to active this audio hal library, first should set property for the device. Add the following line in the boot file ‘init.rc’:

setprop audio.in.device.sklrm 1

Or add in the property file ‘default.prop’:

audio.in.device.sklrm = 1

And then recompile the system code to generate a new boot image file: boot.img. Push this image file into the set-top box and restart to take active.

# Audio Policy Porting

After completing porting of audio driver and audio Hal, it is necessary to modify audio policy to enforce the new sklrm Hal library active.

Before editing the code of audio policy, the header file should be modified at first, which is in the path:

/system/core/include/system/audio.h

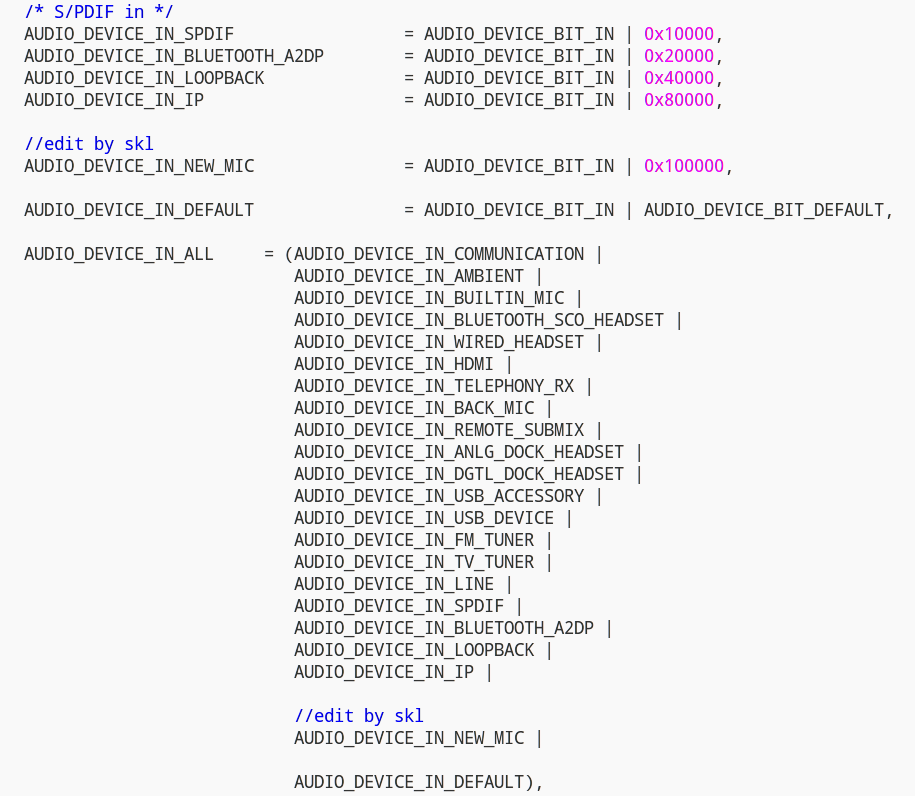
Add a new audio input device:AUDI\_DEVICE\_IN\_NEW\_MIC and its information, and assigning new id for this input device.

Add the code in the struct ‘audio\_devices\_t’:

AUDIO\_DEVICE\_IN\_NEW\_MIC = 0xXXXX，

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

Just like the following figure shown：



In Android 6.0+ version, the whole audio policy service is located at:

/frameworks/av/services/audiopolicy

When using the audio channel, inputgetInput () function is called to select the input audio channel. It will also call the getDeviceForInputSource () access to the source of the input device, which is located at:

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

Add the code to get property when the remote source using:

char value[PROPERTY\_VALUE\_MAX];

int prop\_rm;

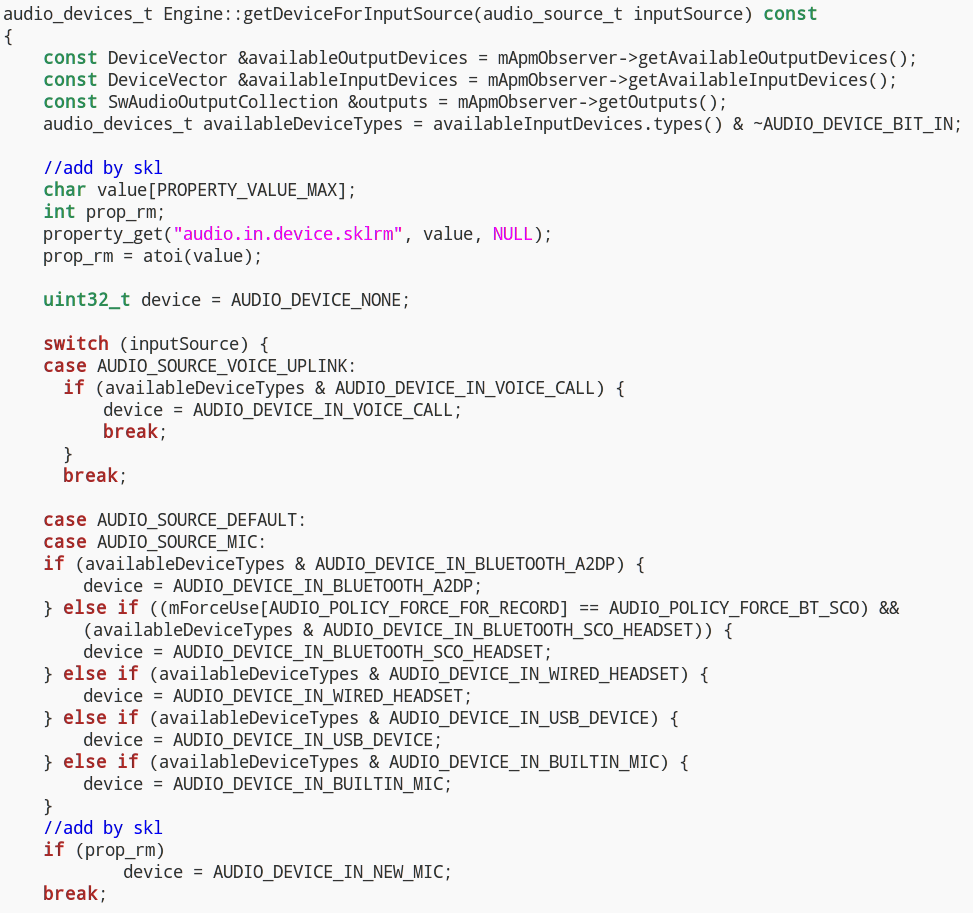
property\_get("audio.in.device.sklrm", value, NULL);

prop\_rm = atoi(value);

if(prop\_rm)

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

Add the code like the following figure：



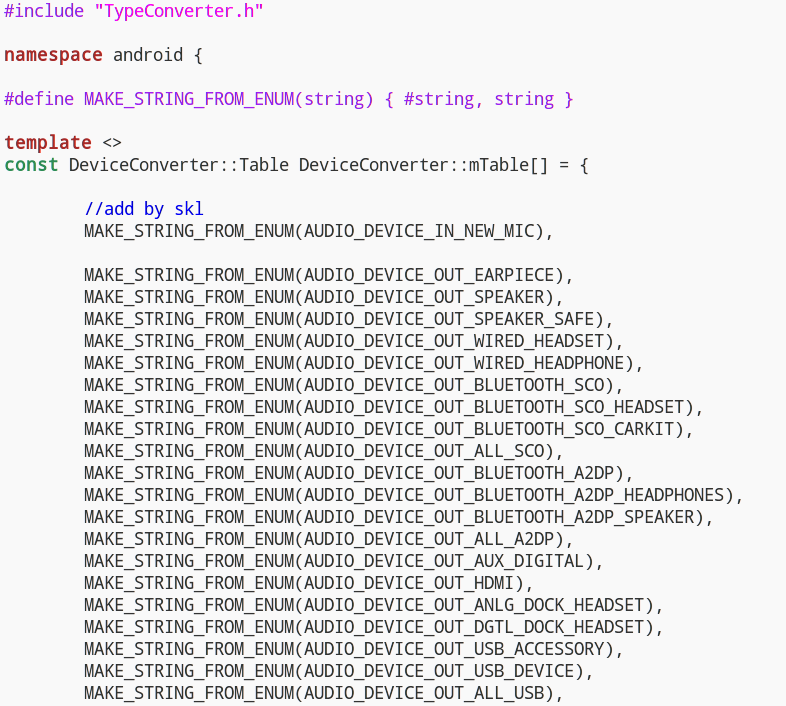
To recognize the new device name ‘AUDIO\_DEVICE\_IN\_NEW\_MIC’, the convert information should by edited in the structure ‘DeviceConverter::Table DeviceConverter::mTable[]’ , which is located at:

/frameworks/av/services/audiopolicy/common/managerdefinitions/include/TypeConverter.cpp

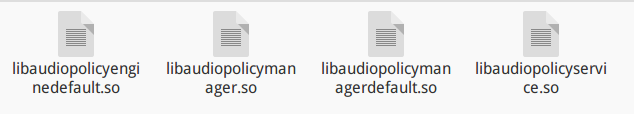
Add the following code：

MAKE\_STRING\_FROM\_ENUM(AUDIO\_DEVICE\_IN\_NEW\_MIC),

As shown in figure：



And then recompile the system code and generate following new audio policy libraries：



The new compiled library should be pushed into the path ‘/system/lib’ (if system is 64 bit, path is ‘/system/lib64’) by entrying the command in the terminal:

adb push PATH/libaudiopolicyenginedefault.so /system/lib

adb push PATH/libaudiopolicymanager.so /system/lib

adb push PATH/libaudiopolicymangerdefault.so /system/lib

adb push PATH/libaudiopolicyservice.so /system/lib

In addition, the audio policy configuration file should be modified. These relative files are generally located in the /device/ ‘VENDOR’ .

Now audio policy use audio\_policy.conf file to configure the module information. At this time, add the new sklrm module information and save it:

sklrm {

inputs {

sklrm {

sampling\_rates 16000

channel\_masks AUDIO\_CHANNEL\_IN\_MONO

formats AUDIO\_FORMAT\_PCM\_16\_BIT

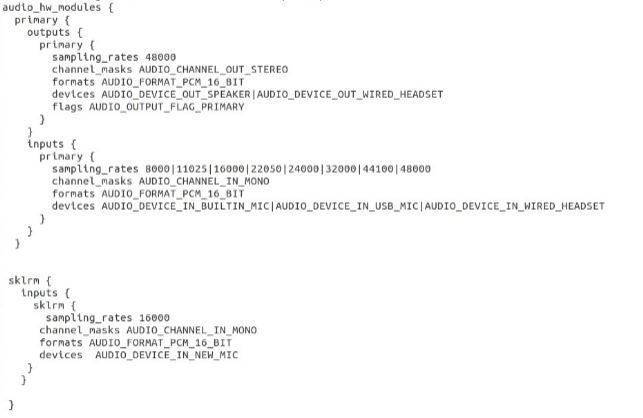
devices AUDIO\_DEVICE\_IN\_NEW\_MIC

}

}

}

As the following figure shown：



And then push the updated ‘audio\_policy.conf’ into the path ‘/system/etc’, type the command in the terminal:

adb push ‘PATH’/audio\_policy.conf /system/etc

# Bluetooth Parameter

To work the voice function, It is necessary to edit the bluetooth LE parameters .The path is：

/system/bt/stack/include/btm\_ble\_api.h

The main parameters are max and min interval, slave latency and timeout.

The recommend values are as following：

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

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

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

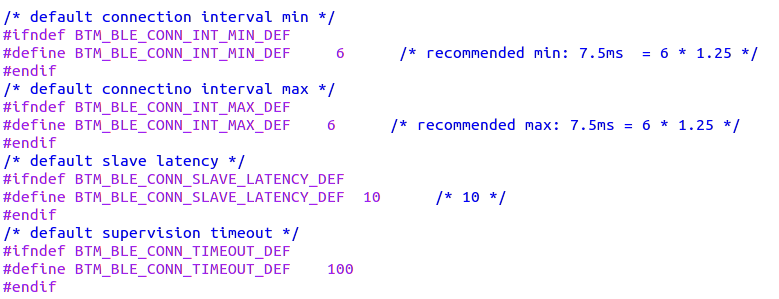
#define BTM\_BLE\_CONN\_SLAVE\_LATENCY\_DEF 10

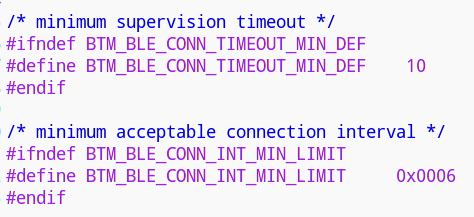
#define BTM\_BLE\_CONN\_TIMEOUT\_DEF 100

#define BTM\_BLE\_CONN\_TIMEOUT\_MIN\_DEF 10

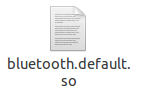
#define BTM\_BLE\_CONN\_INT\_MIN\_LIMIT 0x0006

The file after editing is like:





And then recompile the system code and generate following new bluetooth HAL library：



The new compiled library should be pushed into the path ‘/system/lib/hw’ by typing the command in the terminal:

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

# Version Information

|  |  |  |  |
| --- | --- | --- | --- |
| **Ver.** | **Data** | **Writer** | **Description** |
| 1.0.0 | 2017-07-19 | George | First Version； |

# Contact Information

Shanghai Shockley Electronics Tech. Corp. LTD

Web：http://www.shockley-elect.com

Tel：021-64327686

Contact：George

E-mail：[wythe@shockley-elect.com](mailto:wythe@shockley-elect.com)