Intel® Atom™ x5-E8000 Processor Series: Android* - Selective Audio Routing Programming

White Paper

July 2016
You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting: http://www.intel.com/design/literature.htm.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at http://www.intel.com/ or from the OEM or retailer.

No computer system can be absolutely secure.

Intel, Atom, and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2016, Intel Corporation. All rights reserved.
Contents

1.0 Introduction .........................................................................................................................5
  1.1 Terminology ......................................................................................................................5

2.0 Background ..........................................................................................................................6

3.0 Dynamic Audio Output Settings GUI ....................................................................................7

4.0 Architecture Overview .........................................................................................................10
  4.1 Audio Routing Path Decision ..........................................................................................11
  4.2 Multiple Stream Output .....................................................................................................11
  4.3 Duplicate Stream Output ..................................................................................................12

5.0 Conclusion ............................................................................................................................13

Figures

Figure 1. Sounds and Notifications ..........................................................................................7
Figure 2. Dynamic Audio Output Settings .................................................................................8
Figure 3. Android Media Stack For Playback ............................................................................10

Tables

Table 1. Terminology ..................................................................................................................5
## Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>July 2016</td>
<td>1.0</td>
<td>Initial release.</td>
</tr>
</tbody>
</table>
1.0 Introduction

The Selective Audio Routing function provides three functionalities: (1) Allows users to select single playing audio stream to different output devices manually. For example, users can choose to output the audio stream to a speaker or HDMI monitor or even device detect the headset. (2) Allows multiple audio streams to be output on different devices concurrently. Users can play video clips on the HDMI monitor and play music through the headset or play different video clips on a different HDMI monitor. (3) Allows users to duplicate the audio stream output to different devices. For example, the users can hear the audio stream on both headset and speaker devices at the same time.

The Selective Audio Routing function also provides a GUI for users to set the output behavior. This document introduces the GUI and how to use it, as well as the key idea to implement the Selective Audio Routing feature.

1.1 Terminology

Table 1. Terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>POS</td>
<td>Point of Sale</td>
</tr>
</tbody>
</table>

§
On a standard Android* platform, the audio output device is chosen by a system predefined rule. There is no user interface to force the audio routing to the proper output target device. With the original mechanism, the media services detect the number of possible output devices and automatically choose either the last inserted device or the one with highest output priority. In an embedded environment, however, it is not practical because all the possible audio output devices are always connected by design. This selective audio routing feature provides a way for end users to choose which audio output devices they want. This feature also provides multiple stream output capacity to different audio codec devices. For example, on POS system, the cashier can hear the instruction in privacy from the system and the client can hear the amount of the bill at the same time. Another example is for a video conference device. If the system is connected to two HDMI monitors, the audio output from a different party can be played on different HDMI monitors at the same time, which can provide a better conference experience.
3.0 Dynamic Audio Output Settings GUI

The Dynamic Audio Output Settings GUI enables users to select audio routing either through on-board speakers (specifically for Intel® Atom™ x5/x7 Z8000 processor series, formerly Cherry Trail, only), integrated audio jack, or HDMI audio.

Follow the instruction below to set the audio routing:
1. Make sure that the monitor connected to the HDMI ports has audio capabilities.
2. Go to Settings > Sound & Notification > Dynamic Audio Output Settings

The options are available as below:

Figure 1. Sounds and Notifications
There are three options for audio routing available in the Dynamic Audio Output Settings:

1. **Primary audio output option.**
   - If users want to output the media stream to the proper device, then they can choose from the following options:
     a. System Default: Keep the audio routing policy as Android default design.
     b. Primary audio output for speaker: Select “Speaker” radio button.
     c. Primary audio for headset/headphone: Select “Headphone/Headset” radio button.
     d. Primary audio for HDMI audio: Select “HDMI Audio” radio button.
     e. Primary audio for HDMI2 audio: Select “HDMI2 Audio” radio button. (Applies to systems that connect two monitors with audio output.)

2. **Secondary audio output.**
   - The GUI can be used to assign the application's audio stream to output on proper audio sink devices. The following settings are required for secondary audio output:
     a. The "Secondary output app" textbox refers to the application to which the STREAM MUSIC audio would be rerouted. The name would derive from the package name defined in AndroidManifest.xml. For example, AOSP Music app is “com.android.music”.
     b. The radio group refers to one of the available outputs for secondary applications:
       - Secondary audio output for speaker: Select “Secondary Speaker” radio button (applies on Intel® Atom™ x5/x7 Z8000 processor series, formerly Cherry Trail, platform).
       - Secondary audio output for HDMI audio: Select “Secondary HDMI Audio” radio button.
• Secondary audio output for HDMI 2 audio: Select “Secondary HDMI 2 Audio” radio button.
• Secondary audio output for headphone/headset: Select “Secondary Headphone/Headset” radio button
• Secondary System Default: Secondary System Default audio output function is disabled
c. Restart the assigned application or reboot the device for the changes to take effect.

3. Duplicate audio stream output:
   This feature can let the application using “music stream” type to output on both analog codec (speaker or headset) and HDMI audio output at the same time. To enable this, choose “Duplicate audio stream on analog default and HDMI devices” checkbox. After enabling this, the radio group for (1) and (2) features are disabled.

§
4.0 **Architecture Overview**

The selective audio routing mechanism has major code change on the Android audio policy module because this module’s functionality is making the audio routing decisions according the user scenario and connects the device instantly. This section discloses the design changes as compared with the original Android OS source code.

**Figure 3** below shows the media stack for media playback. The application uses the media player (5) to play the music file or to play a tone by itself. The audioflinger (3) tries to get the output target devices through the audiopolicy module (4). Within the original audio policy implementation, the routing rule has been defined by rules that cannot be changed during runtime. To make the Android system fit the various embedded system requirements, the audiopolicy module (4) is changed to be runtime configurable.

Another requirement for IOT use cases is the multiple stream output, which means two applications can play audio through different devices. For example, the in-car infotainment system's main function is to play the music, but it also needs to play the navigation notification tones through different devices such as the speaker near the driver's ear or the headset used by the driver. In this scenario, the passenger is not disturbed by those tones, which creates a better riding experience. In the Android operating system, however, the audio sound from different applications is mixed into one audio stream and is played through one audio sink because it does not support multiple stream output.

**Figure 3. Android Media Stack For Playback**

![Android Media Stack For Playback Diagram](image-url)
4.1 Audio Routing Path Decision

This feature allows users to choose the output device manually. For example, users can choose the speaker to receive the output audio sound even if the headset is plugged in. In the audio policy module, when the audio stream starts to playback, the audioflinger (3) module invokes the getOutput function to get the HAL handler. In this function, it passes the stream type and audio format to let the module (4) find the proper output HAL for playing. For dynamic audio routing, changes are made in the getDeviceForStrategy function to dynamically change the return device bit mask or invoke the setForceUse API according to the system property. The system property can be changed by using the Setting application. Users can change the audio routing setting in the Setting application and restart the application to make it take effect.

4.2 Multiple Stream Output

This feature provides dual audio stream output to different outputs devices. For example, users can open two applications and output their sounds to two different devices. In an Android media framework, the audio streams from applications would be mixed into single audio stream in audioflinger (3) module. But for low latency audio type like DTMF or touch tone, there is a special track type called fastTrack. It runs individual audio mixer and routing policy control, so that the stream can go to different HAL modules. With this special behavior, the idea is to use the fastTrack to be the second audio stream and it can separate the audio stream from the main one and does not mix together. There are two major modules changes in Android system. One is to identify (a) which is the target application to be the second stream output. The other one is affecting (b) the chosen HAL by the DTMF stream type in audio policy module.

For identifying (a), in Dynamic Audio Setting menu, the user can type the application name in "Secondary output application name" textbox and choose the target output in secondary output radio group. This application name is saved in the System property. When the application starts, the application pId is retrieved by the Android framework and also saved into the system property. When the application tries to play sound, the audio track module is created and the pId is passed into the audioTrack constructor. In the AudioTrack:Set function, our change is checking if the creator’s pId is the same pId as predefined and changes the stream type from AUDIO_STREAM_MUSIC to AUDIO_STREAM_DTMF. In audiopolicy module, the getDeviceForStrategy function in audio policy (4) checks the stream type and returns the assigned device bit by system property, which would affect the selecting of output HAL (b). The result is the application using stream type equal music goes to main HAL output device and the other application using the DTMF stream type goes to a different one.
4.3 **Duplicate Stream Output**

This feature allows users to duplicate the single audio stream to different devices at the same time. For example, when a user is playing a video clip, this feature can allow the audio stream output to HDMI monitor plus headset or HDMI monitor plus speaker.

The design is referring the a2dp duplicate output capacity. In Android systems, the audio stream can output audio to the Bluetooth* headset and speaker in parallel. It can just let the audio policy manager choose both devices on judgment step. By default, the pair of devices (speaker + Bluetooth headset) could be chosen when the device wants to play the notification sound.

The major change for this feature is changing the chosen paired devices return by the getDeviceForStrategy function for music stream type in audio system policy manager. This duplication function in this section only applies for different physical audio codecs. If the device wants to play audio through same audio codec to the output device, it is better to create a different output device.

§
5.0 Conclusion

The Selective Audio Routing feature provides a flexible way to let users choose the audio output device needed without any user application source code modification. Furthermore, it also provides multiple audio streams output features for embedded systems. It could be used on the vehicle infotainment system, the video conference system, and so on. On the next Android version, this feature could be modified to use the native Android API like Media Router APIs to achieve similar function while still maintaining no modification to the application source code.