

SoundBox BTF Development	Date of revision
	20. 08. 22

SoundBox Audio Contol

2020.08.26



SoundBox BTF Development	Date of revision
	20. 08. 22

1 Audio Routing

1.1 Audio Routing

1.1.1 OverView

In the SoundBox, it is a function that changes the output sound source and output device settings according to each function and device status.

1.1.2 Output sound source type of the SoundBox

The types of output sound source of SOUNDBOX are as follows.

1.1.2.1 BTV Media audio (with VOD)

Do not go through the Android Framework (AUDIO_STREAM_BT (15).)

1.1.2.2 Android Audio

Through the Android Framework

In the Routing function, separate as follows:

- * NUGU Stream : STREAM_NUGU_TTS(11), STREAM_NUGU(12),
STREAM_NUGU_CALL(13), STREAM_NUGU_RING(14)
- * etc : Stream : 0 ~ 10

1.1.3 etc

The STREAM_NUGU (12) is set when play music(AVRCP) Phone to the SOUNDBOX through BT sink mode.

1.1.4 Output Device Type for SoundBOX

The default output device is shown below..

Speaker, Aux(Lineout), HDMI, Bluetooth(A2DP)

SoundBox BTF Development	Date of revision
	20. 08. 22

1.1.5 Audio Output Scenario

1.1.5.1 Routing by output sound source type

Device Routing by output type is described in the table below.

If the NUGU sound source is output, it is output to SPK (AUX), otherwise it is output to HDMI (BT).

BTV Media Audio	Android Audio - NUGU	Android Audio - ETC	Routing Device
X	X	X	HDMI(BT)
X	X	O	HDMI(BT)
X	O	X	SPK(AUX)
X	O	O	SPK(AUX)
O	X	X	HDMI(BT)
O	X	O	HDMI(BT)
O	O	X	SPK(AUX)
O	O	O	SPK(AUX)

[table 1] Routing by Stream Type

1.1.5.2 Routing by Output Device Connection Status

Audio Source	Connect state					Sound out	etc
	STB Speaker	HDMI(TV Speaker)	BT Source(RCU EarPhone)	BT Sink(MobilePhon	AUX In		
BTV	O	O	X	X	X	Output BTV only with STB speakers	Output Video only with HDMI
	O	O	O	X	X	Output BTV only with BT Source	
	O	O	X	O	X	BTV Mute, Outputs Connected BT Device Sound to STB Speaker	Same as AI2
	O	O	X	X	O	BTV Mute and output AUX In sound to STB speaker	Automatic path change to AUX In using H/W detection
	External device not set	O	X	X	X	TV Speaker	Same as regular boxed STB
NUGU TTS	Only STB Speaker						
Basic TV Apps (Netflix or YouTube, etc.)	There are only two ways to output audio from a TV to a sound bar set-top box instead of a TV speaker and Bluetooth or AUX In.						
	Synaptics BG5CT(VS550) not support of S/PDIF Input and HDMI RX(ARC)						

[table 2] Routing by Device Output

SoundBox BTF Development	Date of revision
	20. 08. 22

1.1.6 Work Flow

The basic structure of Routing is described in the figure below. When playing audio on the BTV APP, it operates as a flow through the Android Framework.

When the music playback starts and stops, the audioFlinger sends a message from the audioFlinger to the AudioManager.

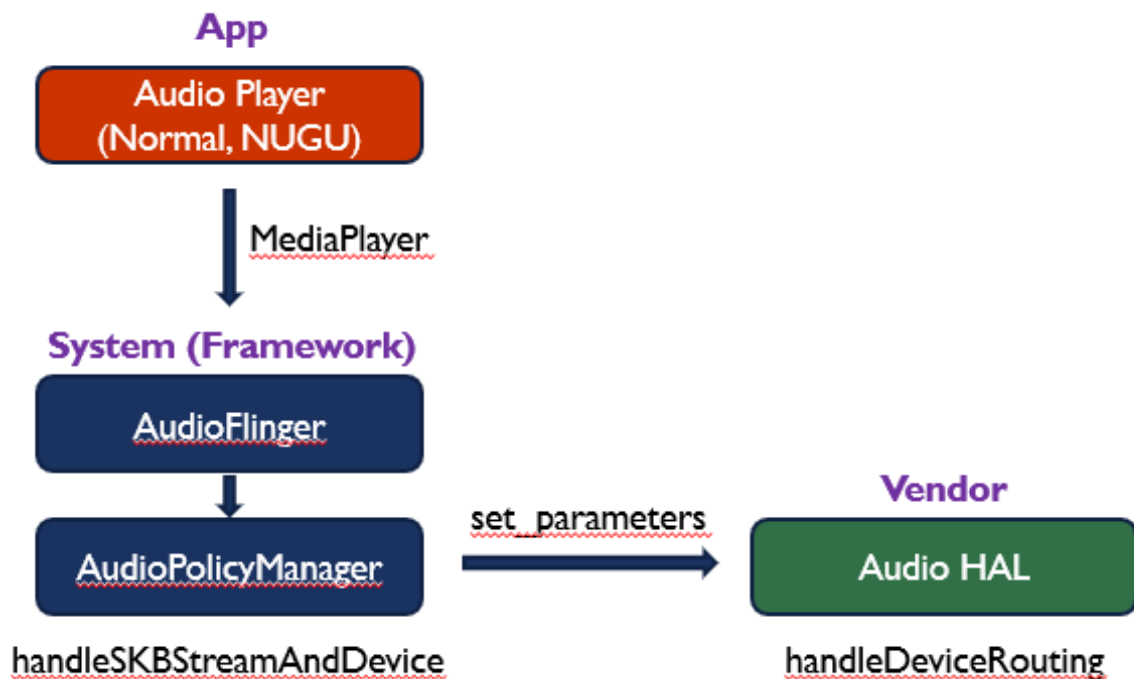
The AudioManager checks the stream type of the sound source in the `handleSKBStreamAndDevDevice()` method added to perform the Routing function, and selects stream mute and routing device.

The `handleSKBStreamAndDevice()` method is used for every start/end of a music source in the method below.

-startOutput(startSource), stopOutput(startSource)

The routing device information is sent to Audio HAL via the set parameter from the `handleSKBStreamAndDevice()`.

Audio HAL performs device routing with a parameter delivered by `handleDeviceRouting()`.



[Picture 1] Work flow of routing

SoundBox BTF Development	Date of revision
	20. 08. 22

1.2 AudioPolicyManager

1.2.1 Overview

A description of the methods below mentioned in 1.1.6 above to implement the Routing function within the AudioPolicyManager.

Source location :

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

Feature : #define VENDOR_AUDIO_ROUTING

handleSKBStreamAndDevice()

1.2.2 Structure

1.2.2.1 Method

```
status_t AudioPolicyManager::handleSKBStreamAndDevice(int event, audio_stream_type_t
stream) {
```

-argument

@event : Current Event Status

STARTING_OUTPUT // start

STOPPING_OUTPUT // stop

CHANGING_DEV_CONN //device connection status change (AUX)

@stream : Current playback starts and ends stream type

1.2.2.2 Description

- Check for Play Active Stream

You can see from the function below.

SoundBox BTF Development	Date of revision
	20. 08. 22

*isStreamActive(stream_type)

example)

isStreamActive(AUDIO_STREAM_NUGU);

isStreamActive(AUDIO_STREAM_NUGU_TTS);

- Set state of stream mute & device Routing

Check the status of the NUGU stream to determine the stream and the routing device..

When the mute request is received, check the mute control status for each stream and perform the mute through the method below.

```
setStreamMute((audio_stream_type_t)st, mute_value,
              desc,
              0 /*delay*/, AUDIO_DEVICE_NONE);
```

Set the mute and unmute status for each device and perform Routing to Audio HAL through the method below.

```
mpClientInterface->setParameters(AUDIO_IO_HANDLE_NONE, param_dev_ctrl.toString());
```

The above method is an interface method that delivers messages to the framework (AudioPolicyService) => Audio HAL..

1.3 Device Volume Control

1.3.1 Overview

Explanation of SOUNDBOX's interface with Audio HAL on each volume change

SoundBox BTF Development	Date of revision
	20. 08. 22

1.3.2 Method

<frameworks/base/services/core/java/com/android/server/audio/AudioService.java>

```
protected void adjustStreamVolume(int streamType, int direction, int flags,
```

```
    String callingPackage, String caller, int uid) {
```

```
private void setStreamVolume(int streamType, int index, int flags, String callingPackage,
```

```
    String caller, int uid) {
```

Each of the above methods is a method called when controlling the volume and setting the mute in RCU, button, and app.

The function above is to lower the interface set by Audio HAL to control the volume of the device.

1.3.3 Volume Control

```
AudioSystem.setParameters("vol_ctrl=dev=" + ( DEVICE ) + ( INDEX ) );
```

@ DEVICE (SPK, AUX, HDMI, BT Set which device you want to control for each device

(Select multiple using bit or Operator)

```
AudioManager.DEVICE_OUT_SPEAKER
```

```
AudioManager.DEVICE_OUT_LINE
```

```
AudioManager.DEVICE_OUT_HDMI
```

```
AudioManager.DEVICE_OUT_BLUETOOTH_A2DP
```

SoundBox BTF Development	Date of revision
	20. 08. 22

@ INDEX : Volme level set 0 ~ 32

1.3.4 Mute Control

```
AudioSystem.setParameters("force_mute="+
```

```
( MUTE/UNMUTE ) |
```

```
( ENBLE ) |
```

```
( AUDIO_EVENT ) |
```

```
( DEVICE ));
```

@ MUTE/UNMUTE :

0 : Mute

0x80000000 : Unmute

@ ENABLE

0 : Set

0x40000000 : Release

@ AUDIO_EVENT :

It is divided into two ways. The first is a mute by RCU, button, and the second is a mute by a specific event, such as a whole family earphone.

Button,RCE is 0 and Whole family earphone is passed on as "0x20000000". The priority is higher for the whole family earphone.

For example, even though it was unmuted by RCU, the final state becomes mute when muted by the whole family earphone..

This part is implemented and managed in Audio HAL (1.5)..

SoundBox BTF Development	Date of revision
	20. 08. 22

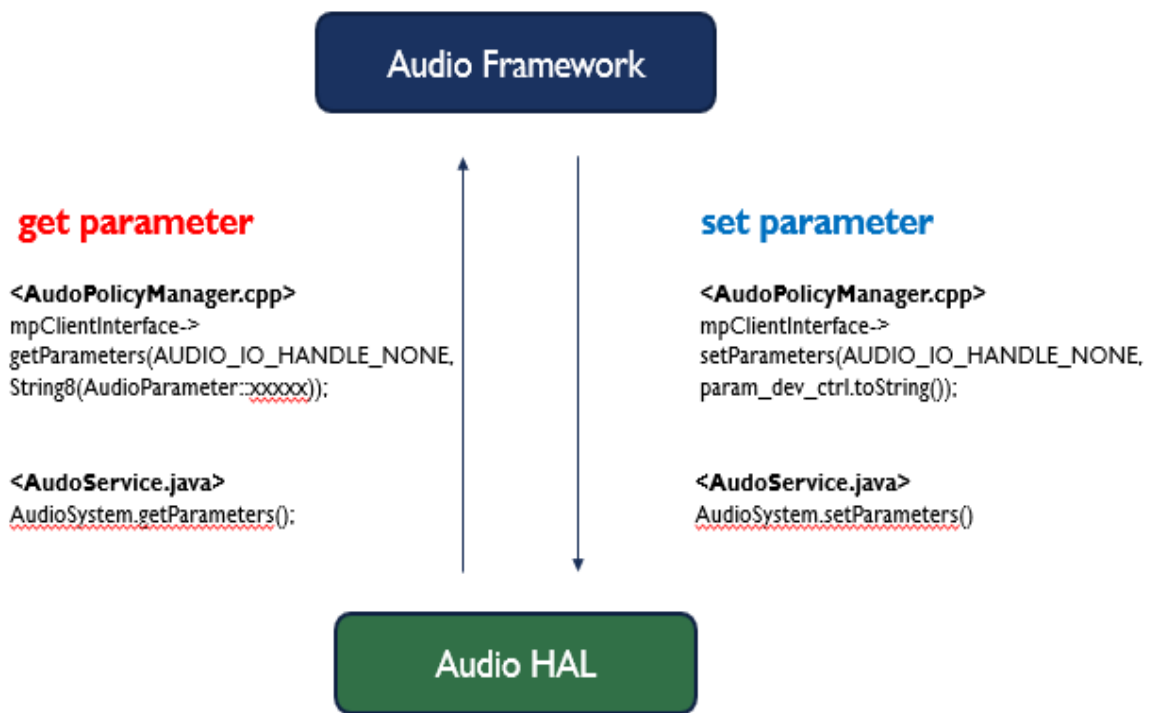
@ DEVICE : The Argument of use is the same as in 1.3.3

1.4 Control Paramter

1.4.1 Overview

It performs control of Device Routing, Volume, and Mute Control through the Framework -> Audio HAL and uses the parameters provided by Android..

This is the default flow that Android provides to forward events to Audio Service (system) -> Audio HAL (vendor).



[Picture 2] Parameters FLOW

1.4.2 Additions in SOUNDBOXOUNDBOX

For controlling Routing, Volume, and Mute Control in SOUNDBOX, we added parameters as follows:

```
<system/media/audio/include/system/audio.h>
```

SoundBox BTF Development	Date of revision
	20. 08. 22

```
#define AUDIO_PARAMETER_DEVICE_MUTE "mute"
```

```
    // Mute control by RoutingRouting
```

```
#define AUDIO_PARAMETER_DEVICE_FORCE_MUTE "force_mute"
```

```
    // Mute control by event (RCU,Button, Whole family earphone)
```

```
#define AUDIO_PARAMETER_DEVICE_VOL_CTRL "vol_ctrl"
```

```
    //Volume Control
```

```
<frameworks/av/include/media/AudioParameter.h>
```

```
static const char * const keyDeviceMute;
```

```
static const char * const keyDeviceForceMute;
```

```
static const char * const keyDeviceVolCtrl;
```

```
<system/media/audio/include/system/audio.h> // using with Audio HAL
```

```
#define AUDIO_PARAMETER_DEVICE_MUTE "mute"
```

```
#define AUDIO_PARAMETER_DEVICE_FORCE_MUTE "force_mute"
```

```
#define AUDIO_PARAMETER_DEVICE_VOL_CTRL "vol_ctrl"
```

1.5 Audio HAL

SoundBox BTF Development	Date of revision
	20. 08. 22

1.5.1 Overview

Execute H/W device changes for Routing, Volume, and Mute Control as requested by the Framework.

1.5.2 Interface

The interface parameters defined during Routing, Volume, and Mute are received and operated in the Framework.

The function of transmitting and receiving parameters is as follows:

<vendor/synaptics/vsxxx/audio/audio_hw.cpp>

```
static int adev_set_parameters(struct audio_hw_device *dev,
                             const char *kvpairs)
```

```
static char *adev_get_parameters(const struct audio_hw_device *dev,
                                const char *keys)
```

Check and perform the operation by checking the character value of AUDIO_PARAMETER_DEVICE_XX mentioned in each of the above functions..

ex)

```
ret = str_parms_get_int(parms, AUDIO_PARAMETER_DEVICE_MUTE, &value);
if (ret >= 0) { // execute
}
}
```

SoundBox BTF Development	Date of revision
	20. 08. 22

1.5.3 audio device util

1.5.3.1 Overview

This part is intended to implement major functions of Routing, Volume and Mute Control and to connect them with the manufacturer's device control (VENDOR_HAL_AUDIO). 소스

<vendor/synaptics/vsxxxx/audio/audio_device_util.cpp>

1.5.3.2 Description

- Routing

```
int f(int *mute_status);
```

- Mute : All Mute

```
int setForceMute(int param_value, int device_mute);
```

- Mute : Each Device

```
int setHDMIAudioMute(bool mute);
```

```
int getHDMIAudioMute(bool *mute);
```

```
int setBtvAudioMute(bool mute);
```

```
int getBtvAudioMute(bool *mute);
```

```
int setSpeakerAudioMute(bool mute);
```

```
int getSpeakerAudioMute(bool *mute);
```

```
int setAUXAudioMute(bool mute);
```

```
int getAUXAudioMute(bool *mute);
```

SoundBox BTF Development	Date of revision
	20. 08. 22

```
int setBTAudioMute(bool mute);
```

```
int getBTAudioMute(bool *mute);
```

- Volume

```
int setHDMIVolume(int level);
```

```
int getHDMIVolume(int *level);
```

```
int setSpeakerVolume(int level);
```

```
int getSpeakerVolume(int *level);
```

```
int setAUXVolume(int level);
```

```
int getAUXVolume(int *level);
```

```
int setBTVolume(int level);
```

```
int getBTVolume(int *level);
```

```
int setRefVolume(int level);
```

1.5.4 setBTAudioMute()

1.5.4.1 Overview

- When playing a NUGU music source, the output is converted to speaker, and the output of BTV should be mute.

- In this case, mute in device of HDMI is possible. However, on SOUNDBOX's BSP, the synaptcis BG5CT platform, there is no way to control BTV output (H/W control) at the low level due to structural problems.

- A change in the state of the meter was requested to the location (Medle Ware - SPTEK) where the output of BTV Media is controlled and applied.

SoundBox BTF Development	Date of revision
	20. 08. 22

1.5.4.2 Description

- The method of forwarding events used the file input method. Mute when there is a below file, or Unmute if not.

* /btv_home/tmp/audio_route/btv_muted

1.5.5 Routing Control

The parameter (AUDIO_PARAMETER_DEVICE_MUTE) is used to invoke the function below, which is implemented in this function.

```
int handleDeviceRouting(int *mute_status);
```

Using this parameter, the Mute status of each device (SPK, AUX, HDMI, BT) is transmitted to handleDeviceRouting(). This function changes the mute state of each device using the value passed.

The setBTAudioMute() is used to change the status of the BTV mute according to the routing state.

1) When NUGU starts playing music, => setBTAudioMute(true);

2) End NUGU Sound Playback => setBTAudioMute(false);

1.5.6 Mute Control

1.5.6.1 Mute type

There are three types of Mute Control.

<vendor/synaptics/common/audio/audio_device_util.h>

```
enum DEVICE_FORCE_MUTE_E {
```

```
    DEVICE_FORCE_MUTE_M_NORMAL /*Control for RCU, Button, UI */,
```

```
    DEVICE_FORCE_MUTE_M_AUDIO_EVENT, /*Contorl for whole family earphone, child
safety setting */
```

SoundBox BTF Development	Date of revision
	20. 08. 22

DEVICE_FORCE_MUTE_M_REMOVE_NOISE/* Control for Mute Processing to eliminate speech recognition start/end noise */,

DEVICE_FORCE_MUTE_M_MAX

};

DEVICE_FORCE_MUTE_M_NORMAL, DEVICE_FORCE_MUTE_M_AUDIO_EVENT 는 위 6.4 에서 언급하였다.

1.5.6.2 setForceMute()

The function above runs Mute on all devices.

This function is called by parameter (AUDIO_PARAMETER_DEVICE_FORCE_MUTE)..

```
int handleDeviceRouting(int *mute_status);
```

To increase the success rate of voice recognition, when Mute enable is requested through DEVICE_FORCE_MUTE_M_REMOVE_NOISE interface, be setBTAudioMute(true).

1.5.6.3 device_ctrl_thread

- Overview

The above mentioned setBTAudioMute() is a thread created to operate by delay at a specific moment.

- Flow

SetBTAudioMute() is not called directly and operates using a thread..

- Description

<vendor/synaptics/common/audio/audio_device_util.h>

SoundBox BTF Development	Date of revision
	20. 08. 22

```
enum MSG_DEV_CTRL_E{

    MSG_DEV_CTRL_BTV_MUTE,    //(0) mute request , request mute, execute immediately

    MSG_DEV_CTRL_BTV_UNMUTE, //(1) unmute request, request mute, execute immediately

    MSG_DEV_CTRL_BTV_UNMUTE_DEB, //(2) unmute Time Delayed Execution

    MSG_DEV_CTRL_MAX

};
```

The types of events are as above. MSG_DEV_CTRL_BTV_MUTE and MSG_DEV_CTRL_BTV_UNMUTE are used in handleDeviceRouting(). If Routing is changed, it is performed immediately. MSG_DEV_CTRL_BTV_UNMUTE_DEB is used when calling DEVICE_FORCE_MUTE_M_REMOVE_NOISE.

Mute and Unmute are frequently called in a short period of time during voice recognition. In this case, repeated Mute and Unmute unnecessarily would adversely affect speech recognition, so the delay of Unmute was given to prevent unnecessary change of state for a short period of time.

MSG_DEV_CTRL_BTV_UNMUTE_DEB request, MSG_DEV_CTRL_BTV_MUTE request, MSG_DEV_CTRL_BTV_UNMUTE request is deleted when MSG_DEV_CTRL_UNMUTE request is received..

For example, when a request is made quickly in the order below, the final request status is Mute.

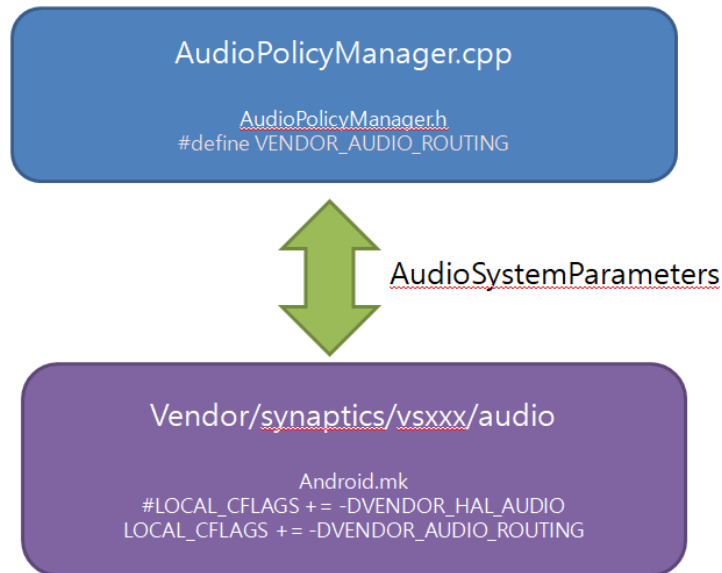
A) Mute(0) ->

B) Unmute(DEB, 2) ->

C) Mute(0)

SoundBox BTF Development	Date of revision
	20. 08. 22

1.6 Manufacturer's Review Function



1.6.1 Overview

The Soundbox source code already has the contents of the previous manufacturer's work. These functions are summarized in FEATURE below.

The manufacturer shall refer to this FEATURE for functional verification and modification.

- 1.2 AudioPolicyManager (Routing , mute , volume control)

AudioPolicyManager.h #define **VENDOR_AUDIO_ROUTING**

- 1.5.3 Audio device util

Vendor/synaptics/vsxxx/audio/Android.mk

LOCAL_CFLAGS += **-DVENDOR_AUDIO_ROUTING**

- 1.5 Audio hal

Vendor/synaptics/vsxxx/audio/Android.mk

LOCAL_CFLAGS += **-DVENDOR_HAL_AUDIO.**