TFA9896 SW RELEASE NOTE
IMX595 Integration

6.0.0_REL
September 19 2019

Document information

Info	Content
Keywords	Driver, codec, calibration, startup, volume control
Abstract	This document gives a high-level overview of TFA9896 driver integration with MIMXRT595 EVK devkit
Document Owners	Mehdi CHEMSI (mehdi.chemsi@nxp.com)

Revision history

Rev	Date	Description
6.0.0	2019-09-19	Initial Version

. Introduction

This document describes how to use the TFA9896 device with MIMXRT595 EVK.

The TFA9896 is a high efficiency class-D audio amplifier with a sophisticated

speaker Boost protection algorithm, accordingly, HW and internal DSP need to be programmed

to ensure the optimal behavior of device, below picture shows the board used to validate the SW

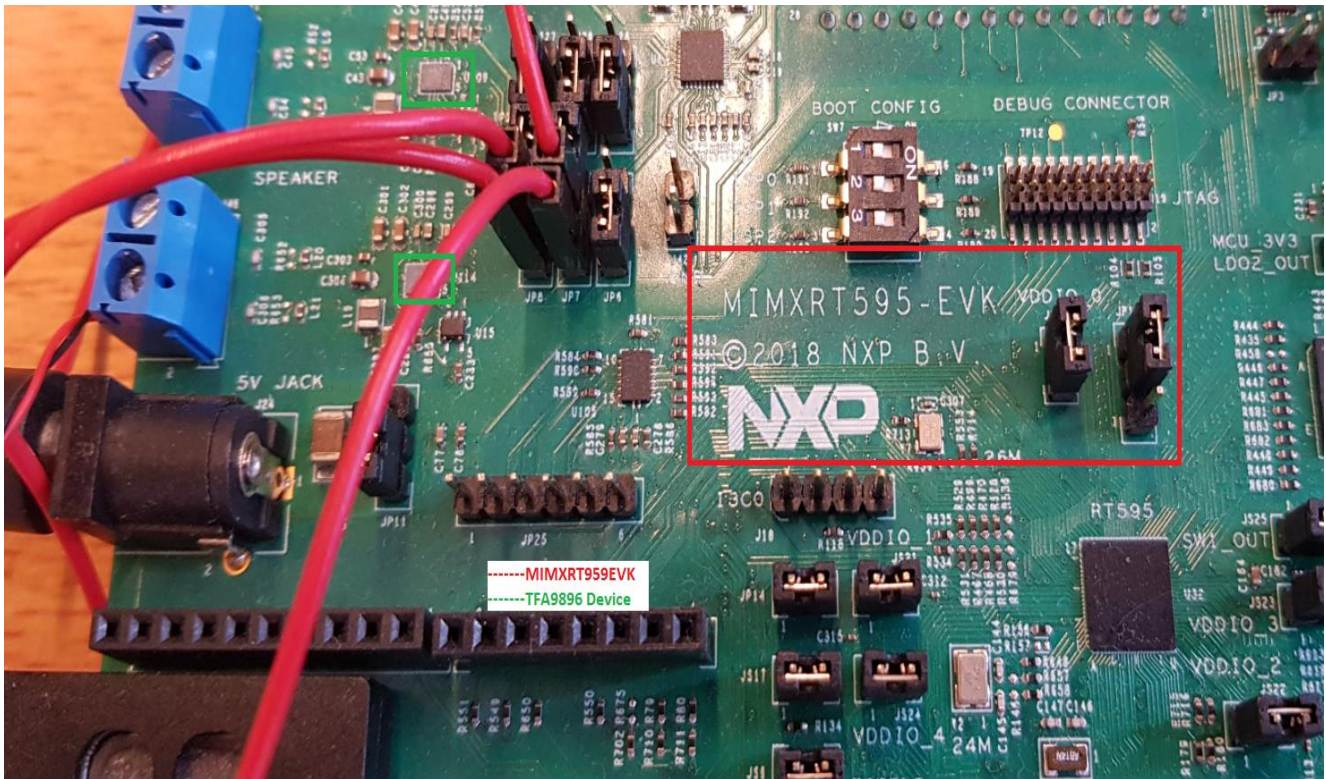driver of TFA9896 object of this document.

TFA9896 SW RELEASE NOTE IMX595 Integration

6.0.0_REL

September 19 2019

1.TFA9896 Device with MIMXRT595 EVK board

. TFA9896 Startup:

This TFA9896 driver complies with codec interface scheme of SDK software adopted by NXP MICR team, accordingly, to start device, to configure HW/DSP or to calibrate the codec API must be used as follows (left and right device as we are in stereo configuration here):

```
if (CODEC_Init(&codecHandleLeft, &boardCodecConfigL) != kStatus_Success)
{
    PRINTF("TFA9896_Init failed for left device!\r\n");
}
PRINTF("Initialize right TFA9896\r\n");
if (CODEC_Init(&codecHandleRight, &boardCodecConfigR) != kStatus_Success)
{
    PRINTF("TFA9896_Init failed for right device!\r\n");
}
```

Codec_init () API is the top-level call to use, underneath an abstraction sequence is being called till calling the main TFA9896 API which is TFA9896_Init () with following prototype:

status_t TFA9896_Init(tfa9896_handle_t *handle, tfa9896_config_t *tfa9896Config)

This API is the handling everything related to device startup, DSP configuration and device calibration.

. TFA9896 factory setting:

For optimal operation of DSP embedded in the TFA9896 some MTP registers values must be pre-programed accordingly, this is done automatically when calling **CODEC_Init ()** the first time with fresh sample via the API **TFA98XX_SetFactoryValues ().**

. TFA9896 device calibration:

If the use case requires temperature protection the user should trigger speaker calibration

that will also set some MTP registers and make communication between the TFA9896 and the

actual speaker.

To trigger calibration user should set .calibrate field of tfa9896_config_t to TRUE as

below (default value is FLASE)

```
tfa9896_config_t tfa9896ConfigL = {

    .i2cConfig        = {.codecI2CInstance = BOARD_CODEC_I2C_INSTANCE,
.codecI2CSourceClock = 99000000U},

    .slaveAddress      = TFA9896_I2C_ADDRESS_LEFT,

    .protocol         = kTFA9896_ProtocolI2S,

    .format          = {.sampleRate = ktfa9896_SampleRate48kHz, .bitWidth =
ktfa9896_BitWidth16},

    .master         = true,

    .calibrate       = true,

};
```

After that **CODEC_Init ()** must be called.

After finalizing that .calibrate field of tfa9896_config_t must be reset to FALSE once again

which is the normal behavior of driver.

Accordingly, user may each decide to calibrate or not.

TFA9896 SW RELEASE NOTE IMX595 Integration

6.0.0_REL

September 19 2019

. **TFA9896 device volume control:**

User might need to control volume from device level in case control from host is not possible.

This is possible via the codec interface that follows:

if (**CODEC_SetVolume**(&codecHandleLeft,ANY_CHANNEL, vol_iter) != kStatus_Success)

      PRINTF("TFA9896_SetVolume failed for left device!\r\n");

if (**CODEC_SetVolume**(&codecHandleRight,ANY_CHANNEL,vol_iter) != kStatus_Success)

      PRINTF("TFA9896_SetVolume failed for right device!\r\n");

CODEC_SetVolume() User is the codec interface API, underneath the TFA9896 API

TFA9896_SetVolume() is being called to set volume interface.