How to Use Sound Subsystem in Traveo II Family

Associated Part Family

- CYT2 Series
- CYT3 Series
- CYT4 Series

About this document

Scope and purpose

AN226043 explains how to use the Sound Subsystem for Infineon Traveo II Automotive Cluster Family MCUs. The application note uses four use cases to demonstrate each function. It also explains the necessary settings for I2S, TDM, PWM, Mixer, Sound Generator, and Audio Digital Analog Converter.

Intended audience

This document is intentded for anyone using Traveo II family CYT2/CYT3/CYT4 series.

Table of contents

Table of contents

1 Introduction

The Traveo II Cluster device is a Traveo II MCU targeted at automotive systems such as instrument clusters and head-up display (HUD). The CYT2 series has an Arm Cortex-M4F-based CPU (CM4) and a Cortex-M0+-based CPU (CM0+). The CYT4 series has two Arm Cortex-M7-based CPUs (CM7) and CM0+, and the CYT3 series has one CM7 and CM0+. These products enable a secure computing platform and incorporate Infineon's low-power flash memory along with multiple high-performance analog and digital functions.

This application note describes how to use the Sound Subsystem for Infineon Traveo II Automotive Cluster Family MCUs.

To understand the functionality and terminology used in this application note, see the "Sound Subsystem" chapter in the **Architecture Technical Reference Manual (TRM)**.

2 Sound Subsystem

The Sound Subsystem includes the following blocks:

- **Time Division Multiplexed (TDM) / Inter-IC Sound (I2S) Interface**
- **Pulse Width Modulated (PWM) Interface**
- **Mixer**
- **Sound Generator (SG)**
- **Audio Digital-to-Analog Converter (DAC)**

Each block of the sound subsystem will be described in the following sections.

2.1 Time Division Multiplexed (TDM) / Inter-IC Sound (I2S) Interface

The TDM/I2S interface consists of a TDM transmitter and a TDM receiver, which can function simultaneously. The I2S interface is obtained as a special case of TDM. Both transmitter and receiver support master and slave functionality. The TDM/I2S interface consists of a bit clock (TDM_SCK), frame synchronization (TDM_FSYNC), and serial data (TDM_SD). In addition, a master interface clock (TDM_MCK) is provided.

2.1.1 Features

The following are the features of TDM/I2S:

- Combined I2S and TDM functionality
- Master and slave functionality
- Full-duplex transmitter and receiver operation
- Support for up to 32 channels. Each channel can be individually enabled/disabled.
- Programmable interface clock
- Programmable channel size (up to 32 bits)
- Programmable late capture – extra delay of 1, 2, or 3 cycles for multi-cycle round-trip latencies in receiver master mode
- Delayed sampling support
- Programmable PCM sample formatting
- Programmable synchronization pulse type
- Left-aligned and right-aligned sample formatting
- 128-entry Tx FIFO with interrupt and trigger support
- 128-entry Rx FIFO with interrupt and trigger support
- Test mode (transmitter to receiver loopback)
- Debug/freeze support

2.2 Pulse Width Modulated (PWM) Interface

A PWM interface drives the PWM output lines and their complementary output lines. This interface processes PCM input signals into PWM output signals.

2.2.1 Features

The following are the features of PWM:

- Programmable interface clock

- Programmable doubling mode
- Programmable gain
- Programmable pulse width modulation
- Programmable PCM sample formatting
- 64-entry TX FIFO with interrupt and trigger support
- Debug/freeze support

2.3 Mixer

The mixer combines multiple PCM source streams in the memory to a single PCM destination stream. The PCM source streams are on-the-fly, which can be up/down-scaled along with the destination stream's sample frequency. The PCM destination stream is either written to memory or transmitted over an I^2S interface.

2.3.1 Features

The following are the features of Mixer:

- The mixer combines multiple PCM source streams into a single PCM destination stream.
- Typically, a PCM source stream consists of a repetition of a PCM sample pattern in memory. The number of repetitions is dynamic; that is, at the stream start, the stream ends may not be known. Typically, a system event activates (starts) a PCM source stream; another system event deactivates (ends) a PCM source stream.
- A PCM source stream can be gain or volume controlled.
- A PCM source stream can be faded in and faded out.
- A PCM source stream sample frequency has a specific ratio with respect to the PCM destination stream sample frequency: 0.5x, 1x, 2x, 3x, 4x, 6x, 8x,12x. The mixer upscales (2x, 3x, …, 12x) or down-scales (0.5x) the PCM source stream to the PCM destination stream.
- The PCM destination stream can be gain or volume controlled.
- The PCM destination stream can be faded in and faded out
- Fixed PCM sample formatting: 16-bit pairs.
- I^2S transmitter with master and slave functionality.

2.4 Sound Generator (SG)

The Sound Generator produces PWM tone (frequency) and amplitude (volume) signals (SG does not produce PCM signals). SG works on the principle that any audio waveform can be decomposed into a series of discrete samples termed as segments. SG works by capturing the behavior of individual segments in a series of configuration registers.

2.4.1 Features

The following are the features of SG:

- PWM-modulated (amplitude, tone) sound generation
- Double-buffered segment structure control
- Separate volume and frequency control (two signals) and combined volume-frequency control (one signal) formats
- Programmable interface clock

2.5 Audio Digital-to-Analog Converter (DAC)

The audio DAC takes the PCM data, converts it to analog, and drives to both left and right pins respectively. (except CYT2 series)

2.5.1 Features

The following are the features of audio DAC:

- Supports stereo (left and right)
- Programmable sampling rate and frequency control
- Cascaded Integrated-Comb (CIC) filter, Finite Impulse Response (FIR) filter, Interpolation filter, and Delta-Sigma modulator
- Multi-level DAC
- 64-entry Tx FIFO with interrupt and trigger support
- Debug/freeze support
- Test mode for analog block

3 Application

In this section, TDM, I²S, PWM, Mixer, DAC, SG, and PDM are implemented to show the following use cases:

- **Use Case 1**: Sound Generator Output
- **Use Case 2**: Mixer combines I2S data and PCM data where stocked in the Mixer's source FIFO, and combined data propagate by PWM.
- **Use Case 3**: Each Mixer's source FIFO will receive PCM data from one of the eight channels in TDM, respectively; the Mixer then combines the data to be propagated by DAC. This use case is only for CYT3/4 series. Note that CYT2 series does not have the audio DAC function.
- **Use Case 4**: Generate PCM data by PDM interface.

3.1 Use Case 1

This section explains how to use the sound generator to produce a PWM tone (use combined volume-frequency mode).

3.1.1 Description of Use Case 1

In use case 1, data is pre-prepared in flash to be used for the SG. Then, the data is input into the SG registers. The SG also uses the step control to change different amplitude (volume) in the same segment. Finally, the SG will produce the PWM tone (frequency) of 500 Hz to 2000 Hz, incremented by 500 Hz for each segment, and amplitude (volume) goes from high to low in each segment, and duty is set as 50 %. The amplitude frequency is set as 48 kHz. To average the amplitude frequency, use the low-pass filter cutoff frequency set as 5.8 kHz. **Figure 1** shows the block diagram for the sound generator. **Figure 3** shows the change of each segment.

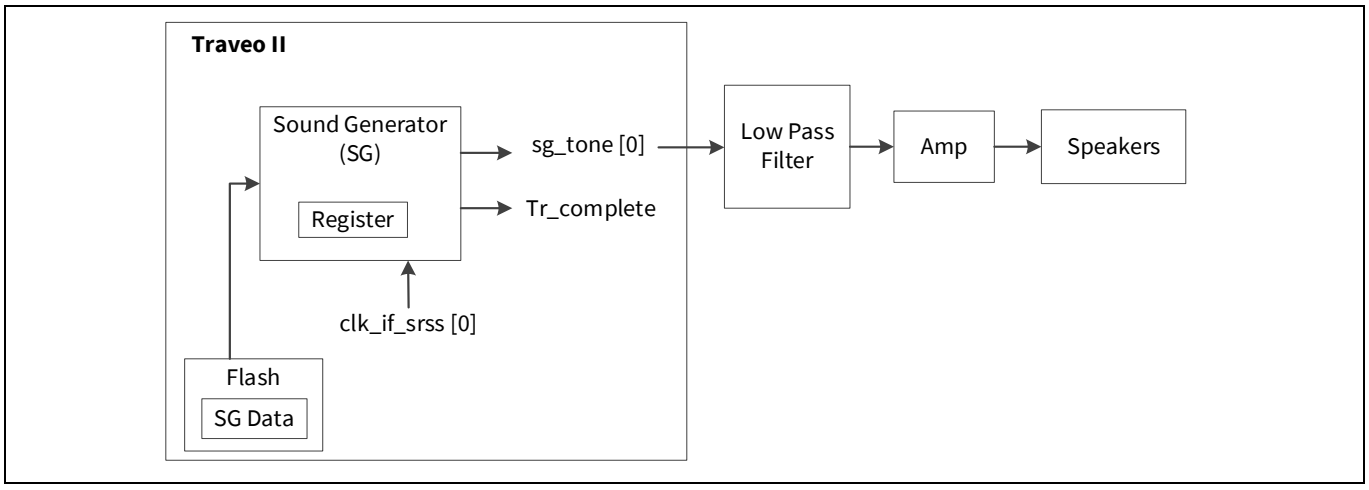

Figure 1 Sound Generator Block Diagram

Figure 2 shows the low-pass filter to be used with this use case. This filter passes signals with a frequency lower than a selected cutoff frequency and attenuates signals with frequencies higher than the cutoff frequency.

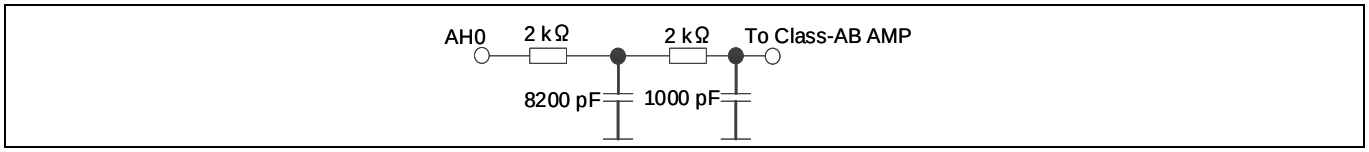

Figure 2 Low-Pass Filter

Application

Table 1 Characteristics of Filter

Characteristics	Value
First RC filter	2 kΩ, 8200 pF
Second RC filter	2 kΩ, 1000 pF
Cutoff frequency	5.8 kHz
PWM carrier attenuation	-55 dB (f_{PWM} = 48 kHz)

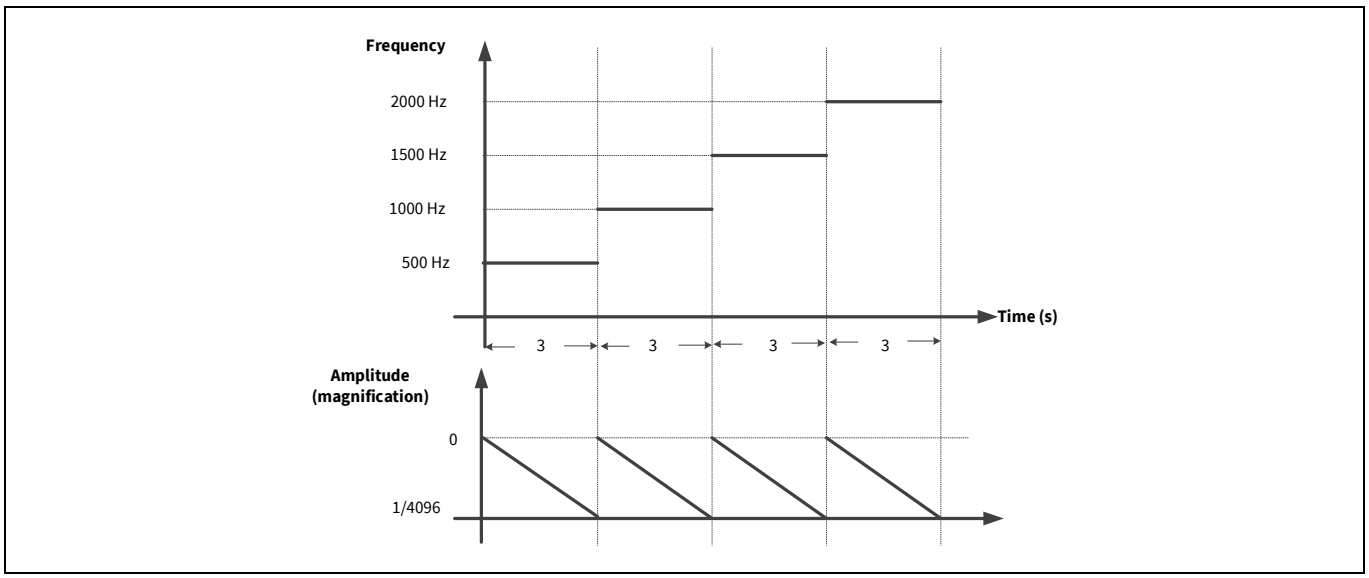

Figure 3 Sound Generator

Figure 4 illustrates SG processing. The SG uses the control register to combine volume-frequency, and to select the SRSS clock “clk_if_srss [0]”, which is connected to CLK_HF [5] .

CLK_HF [5] is used in CYT3/4 series. For other series and more details on clocks, see the “Clock Subsystem” chapter of the **Architecture Technical Reference Manual (TRM)** and **Datasheet**.

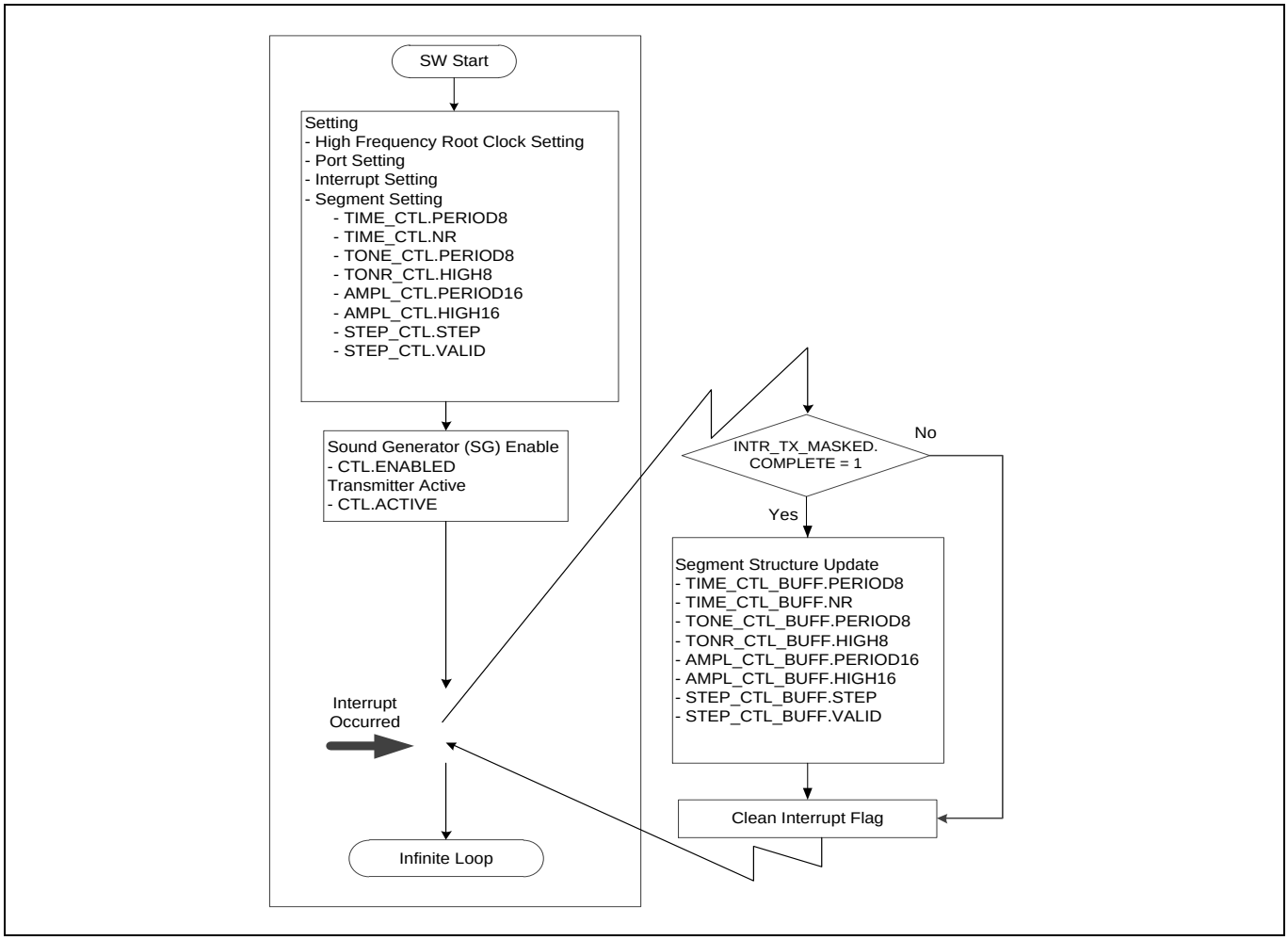

Figure 4 Sound Generator Processing

3.1.1.1 High-Frequency Root Clock Setting

This section explains the clock that can be used. Traveo II family has several high-frequency root clocks (CLK_HF). Each CLK_HF has a destination on the device.

This use case uses a high-frequency root clock (CLK_HF):

- CLK_HF for SG - Enables CLK_HF [5] when the register IF_CTL.CLOCK_SEL is “0” (clk_if_srss [0]). CLK_HF [5] is used in CYT3/4 series.
- The clock is supplied from CLK_HF, and clock selection is determined by SRSS. For other series and more details on clocks, see the “Clock Subsystem” chapter of the **Architecture TRM** and **Datasheet**.

3.1.1.2 Port Setting

In this use case, port SG0_SG_TONE is used.

To set the drive mode, the interrupt mask, and the edge detect at the port setting, see the “IO Subsystem” chapter of the **Architecture TRM**.

3.1.1.3 Interrupt Setting

The SG Complete interrupt is used in this use case.

See the Interrupts chapter of the **Architecture TRM** for details on the vector number of the interrupt and the procedure to configure the interrupt priority, vector address, and enabling/disabling interrupts.

3.1.1.4 SG Setting

Figure 5 shows the configuration that needs to be set.

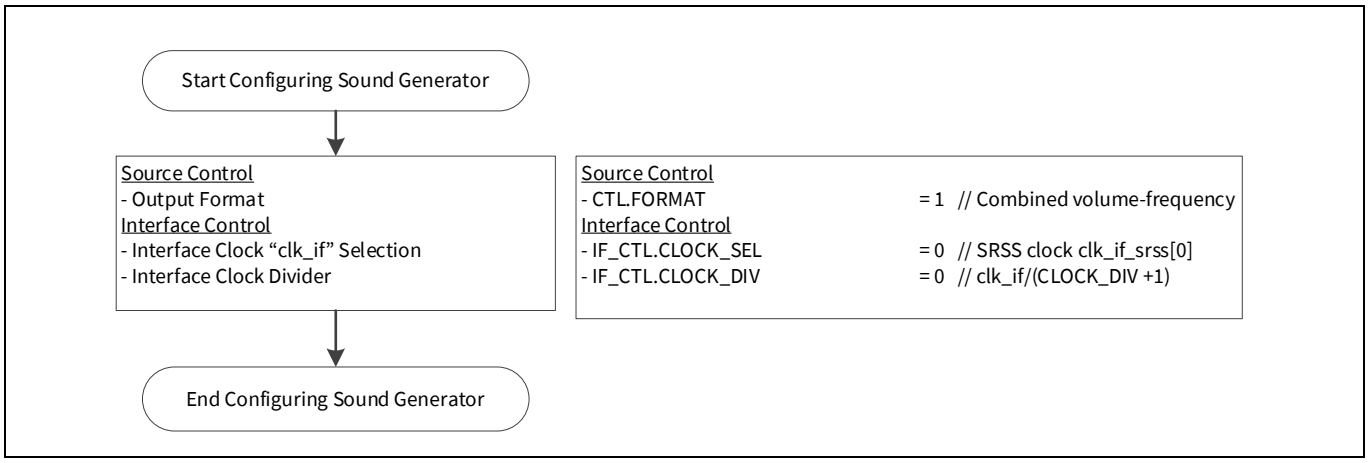

Figure 5 Sound Generator Setting

3.1.1.5 Interrupt Routine

Figure 4 shows the interrupt routine for the SG.

INTR_TX_MASKED.COMPLETE is the interrupt masked register that is used to check whether the complete interrupt occurs. The register value equals 1, which means that the SG Complete interrupt has occurred. Then, you need to clear the interrupt flag and wait for next interrupt to occur.

3.2 Use Case 2

This use case uses I^2S to transmit and receive the data. This use case uses the mixer to combine the I2S data and PCM data, which is stored in the mixer's source FIFO in advance, and then places the combined data in the PWM via DMA.

3.2.1 Description of Use Case 2

Figure 6 shows the block diagram of use case 2. This use case needs two devices, one for the I^2S transmitter and one for the I^2S receiver. A violin "Do – re - mi" data with 48-kHz sampling rate and 16-bit size is pre-prepared in the flash on Device 1, and a piano "Do – re - mi" data with the same sampling rate and size is pre-prepared in the flash on Device 2. The I^2S transmitter is set as slave mode and I^2S receiver is set as master mode. This means that the I^2S receiver outputs WS and SCK signals. The transmitter uses the WS and SCK signals from the receiver side. The clock is an internal clock derived from CLK_HF [5] (Pre-condition CLK_HF [5] = 196,608 kHz.). The I^2S clock is 1,536 kHz, because it uses the data with a 48-kHz sampling rate and 16-bit size ($48000\ \text{Hz} \times 16\ \text{bit} \times 2\ = 1536\ \text{kHz}$). CLK_HF [5] is used in CYT3/4 series.

Application

The operation of the I²S transmitter is to write the PCM data to the Tx FIFO on Device 1 via DMA [0] kicked by the Tx Trigger FIFO Level. Device 1 uses the standard I²S format to transmit the PCM data to the Rx FIFO on Device 2.

The I²S receiver receives the PCM data from the Tx FIFO on Device 1 using the standard I²S format, then DMA [1] is kicked by the Rx Trigger FIFO Level to start transmitting the PCM data to the Mixer Source FIFO [0].

The Mixer Source FIFO [0] receives the data from I²S. The Mixer Source FIFO [1] receives the PCM data via DMA [2] kicked by the Mixer Source Trigger FIFO Level from the flash. Thus, there are violin “Do – re - mi” data and piano “Do – re - mi” data in Source FIFO [0] and [1]. The mixer uses gain control to reduce the volume and avoid the hissing sound, and then combines the PCM data that inputs into the PWM FIFO via the DMA [3] kicked by Mixer Destination FIFO Trigger Level. Finally, through the H-bridge circuit (in **Figure 5**), the sound is played over the speakers; the violin and piano overlap can be heard.

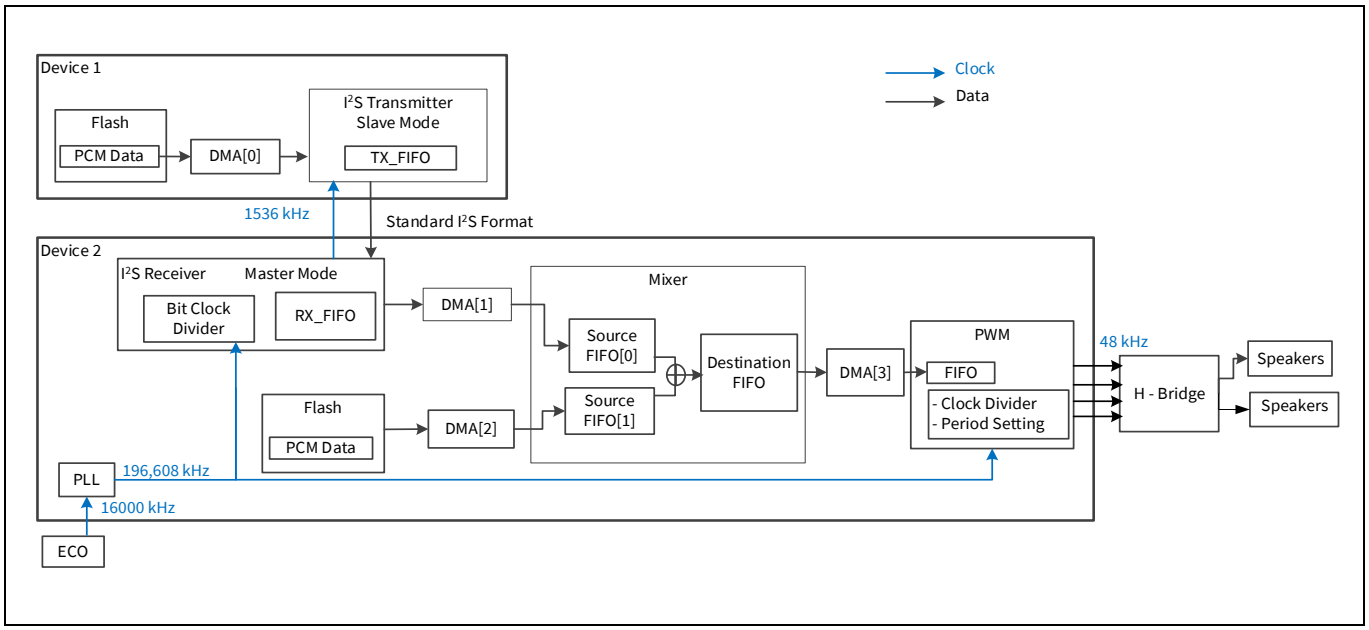

Figure 6 Block Diagram

Figure 7 provides the settings necessary for high-frequency root clock setting, port setting, interrupt setting, DMA setting, I²S setting, mixer setting, PWM interface setting, and Trigger MUX setting, and illustrates the procedure for the use case.

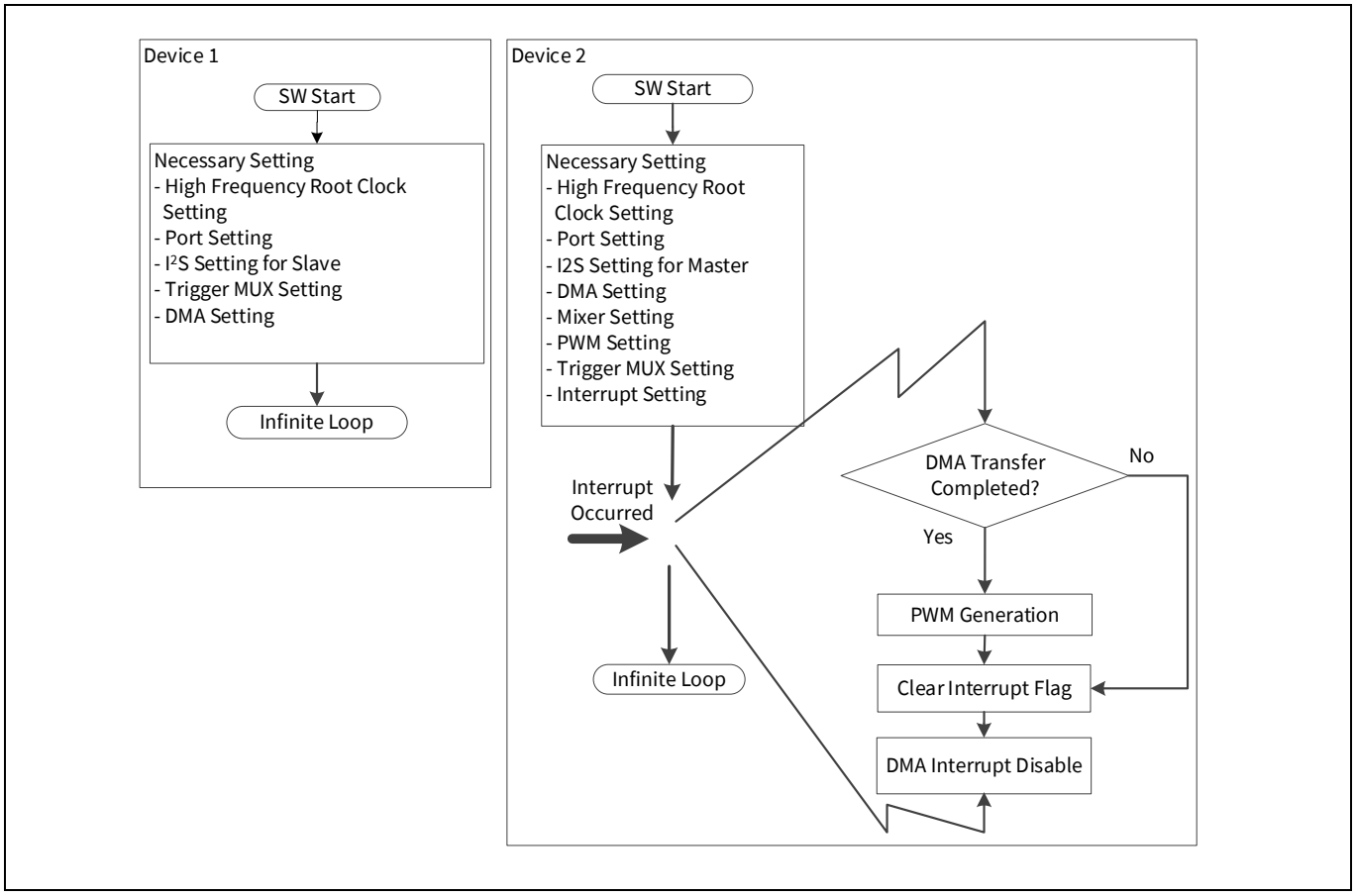

Figure 7 Transmitter and Receiver

3.2.1.1 High-Frequency Root Clock Setting

This section explains the clock that can be used. Traveo II family has several high-frequency root clocks (CLK_HF). Each CLK_HF has a destination on the device.

This use case uses three high-frequency root clocks (CLK_HF):

- CLK_HF for I^{2}S - Enables CLK_HF [5] when the TX_IF_CTL.CLOCK_SEL register is “0” (clk_if_srss [0]).
- CLK_HF for Mixer - Enables CLK_HF [5] when Mixer channel 0 is enabled (clk_if_srss [0]).
- CLK_HF for PWM - Enables CLK_HF [5] when the IF_CTL.CLOCK_SEL register is “0” (clk_if_srss [0]).

These clocks are supplied from CLK_HF; clock selection is determined by SRSS. CLK_HF [5] is used in CYT3/4 series.

For other series and more details on clocks, see the “Clock Subsystem” chapter of the **Architecture TRM** and **Datasheet**.

3.2.1.2 Port Setting

This section explains the port signal used in this use case.

Table 2 Port Setting of Use Case 2

Component	Port Signals
I^2S receiver (Rx)	I2S0_I2S_RX_SCK, I2S0_I2S _RX_MCK, I2S0_I2S _RX_FSYN, I2S0_I2S _RX_SD
I^2S transmitter (Tx)	I2S0_I2S_TX_SCK, I2S0_I2S_TX_MCK, I2S0_I2S _TX_FSYNC, I2S0_I2S _TX_SD
Mixer	MIXER_MCK, MIXER_SCK, MIXER_WS, MIXER_SD
PWM	PWM0_PWM_LINE1_P, PWM0_PWM_LINE1_N, PWM0_PWM_LINE2_P, PWM0_PWM_LINE2_N

To set the drive mode, the interrupt mask, and the edge detect at the port setting, see the “IO Subsystem” chapter of the **Architecture TRM**.

3.2.1.3 Interrupt Setting

This use case uses DMA interrupt to start the PWM.

See the “Interrupts” chapter of the **Architecture TRM** for details on the vector number of the DMA interrupt and the procedure to configure the interrupt priority, vector address, and enabling/disabling of interrupts.

3.2.1.4 DMA Setting

This use case uses DMA to transfer the data.

See the “Direct Memory Access” chapter of the **Architecture TRM** to set DMA.

3.2.1.5 Trigger MUX Setting

This use case uses the trigger mux to kick DMA using the FIFO level interrupt.

See the “Trigger Multiplexer” chapter of the **Architecture TRM** to set trigger group and trigger number.

3.2.1.6 I2S Setting

Figure 8 shows the configuration of Tx and Rx that need to be set on Device 1 and Device 2.

Application

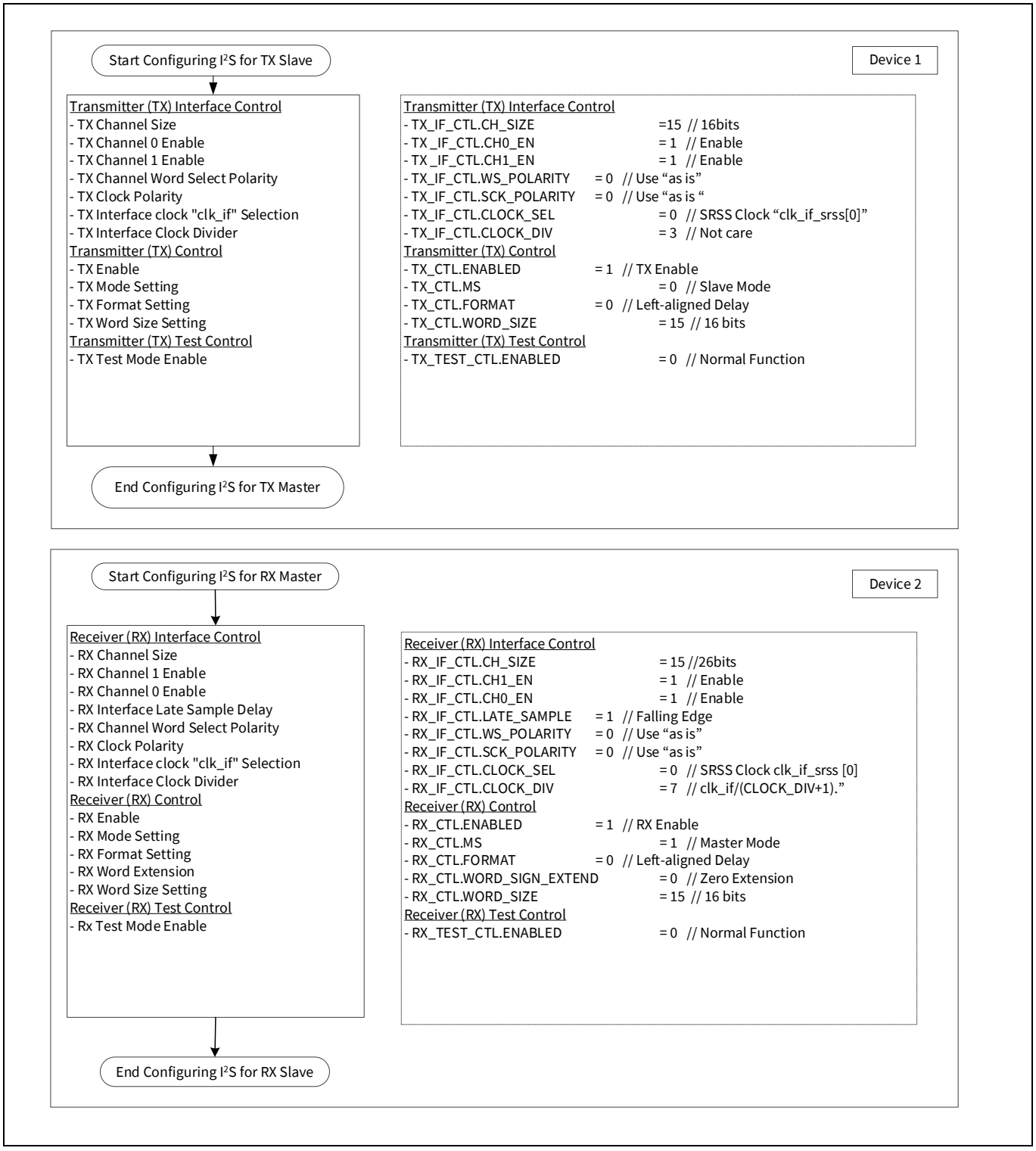

Figure 8 I²S Setting

3.2.1.7 Mixer Setting

Figure 9 shows the configuration that needs to be set.

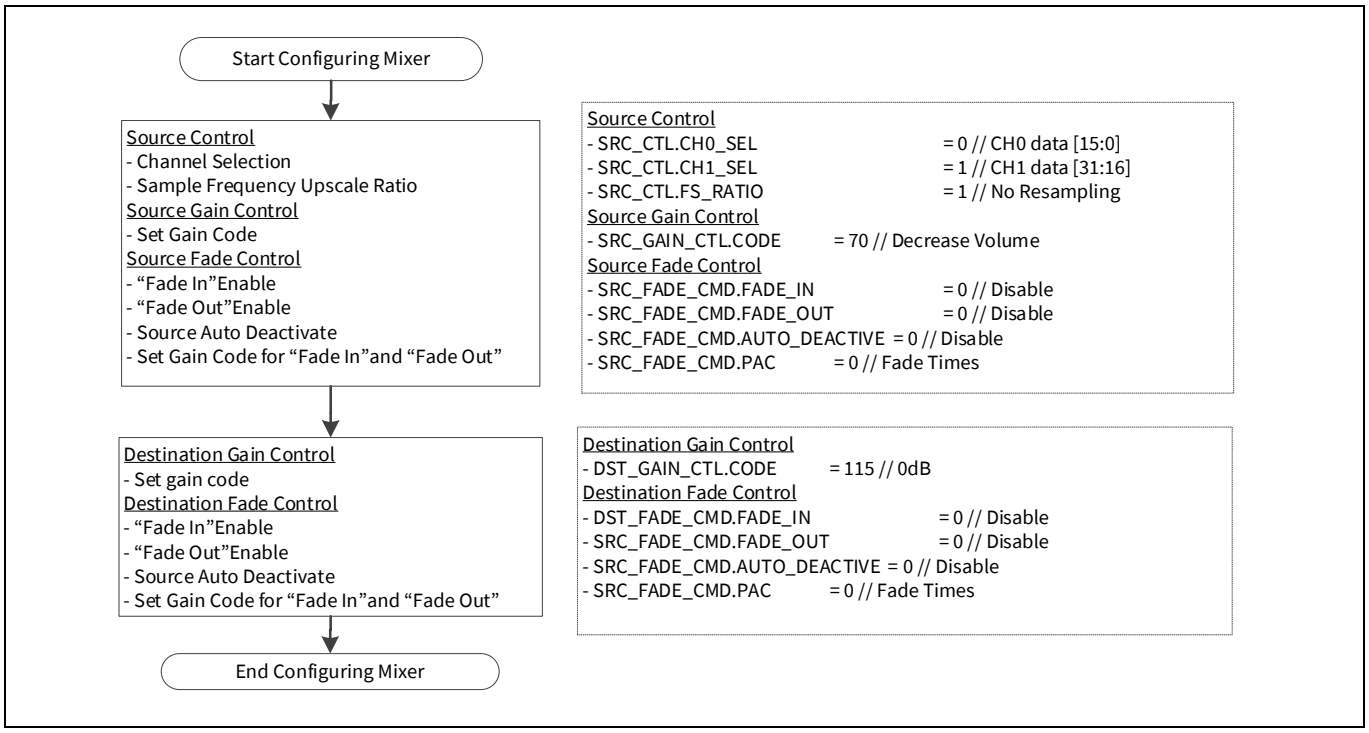

Figure 9 Mixer Setting

3.2.1.8 PWM Setting

Figure 10 shows the configuration that needs to be set:

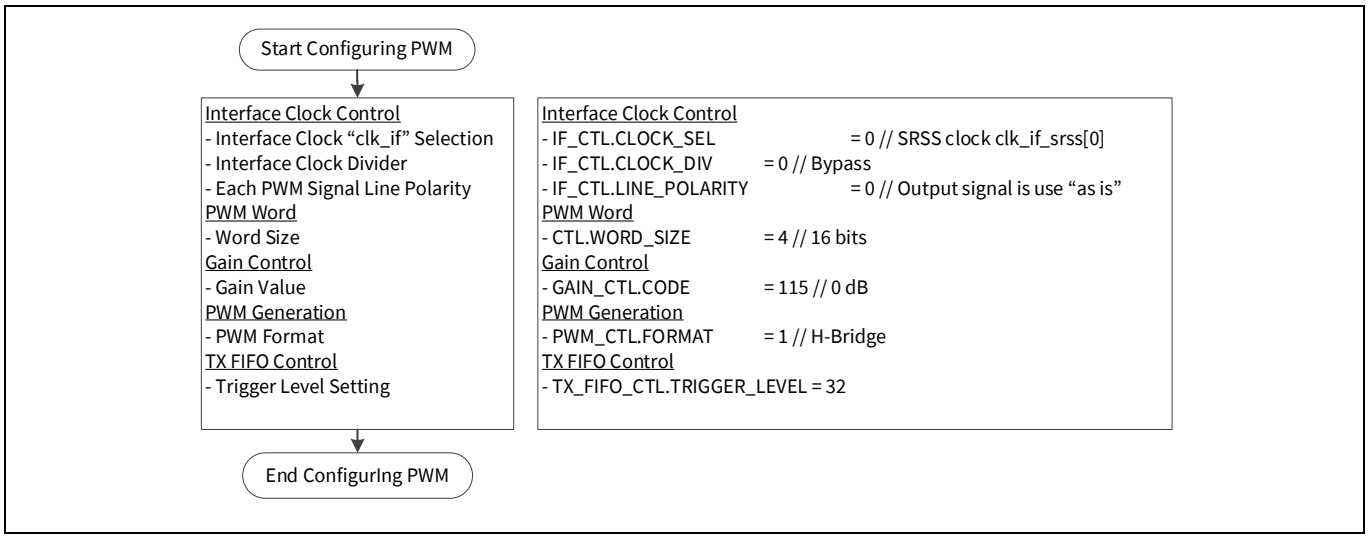

Figure 10 PWM Setting

In use case 2, the TX_FIFO_CTL.TRIGGER_LEVEL register equals 32, which means that when the Tx FIFO has fewer than 32 entries, a transmitter trigger event is generated.

For more information on interface formats, see the "Sound Subsystem" chapter of the **Architecture TRM**.

3.2.1.9 Interrupt Routine

Figure 7 shows the interrupt routine for the PWM.

INTR_TX_MASKED.COMPLETE is an interrupt-masked register which is used to check whether the interrupt occurs. If the register value equals 1, it means that the DMA Transfer Complete interrupt has occurred. Then, you can clear the interrupt flag and wait for next interrupt to occur.

3.3 Use Case 3

In this use case, each mixer's Source FIFO will receive the PCM data from the TDM via DMA. Then, the mixer will combine the data; the combined data will be propagated by the DAC. This use case is only for CYT3/4 series. Note that CYT2 series does not have the audio DAC function.

3.3.1 Description of Use Case 3

This use case needs two devices, one for the TDM transmitter and one for the TDM receiver. In this use case, a 16-bit size data (“Do- Re-Mi-Fa- So- Ra- Si-Do”) and with 48-kHz sampling rate is pre-prepared in the flash on the TDM transmitter. Then, the PCM data is input into the TDM Tx FIFO on Device 1 via DMA [0] kicked by the Tx Trigger FIFO Level. The receiver (Device 2) receives the PCM data in the Rx FIFO.

The mixer has eight source FIFOs, FIFO [0] to FIFO [7]. These FIFOs are used to store the PCM data (“Do- Re-Mi-Fa- So- Ra- Si-Do”) from the TDM via DMA [1] kicked by the TDM Rx Trigger FIFO Level. The mixer combines eight channels of the PCM data that is input into the DAC via DMA [2] kicked by the Mixer Destination FIFO Trigger Level. The DAC sampling frequency is set as 48 kHz.

Finally, the sound is played over the left and right speakers, which output the same sound.

In this use case, the TDM transmitter is set in slave mode and the TDM receiver is set in master mode. This means that the TDM receiver outputs the WS and SCK signals. The transmitter uses the WS and SCK signals from the receiver. The clock is an internal clock derived from CLK_HF [5] (Pre-condition CLK_HF [5] = 196,608 kHz.). The TDM clock is 6144 kHz, because it uses eight channels,16-bit size, and sampling set as 48 kHz ($48000\ \text{Hz} \times 16\ \text{bit} \times 8\ = 6144\ \text{kHz}$). CLK_HF [5] is used in CYT3/4 series.

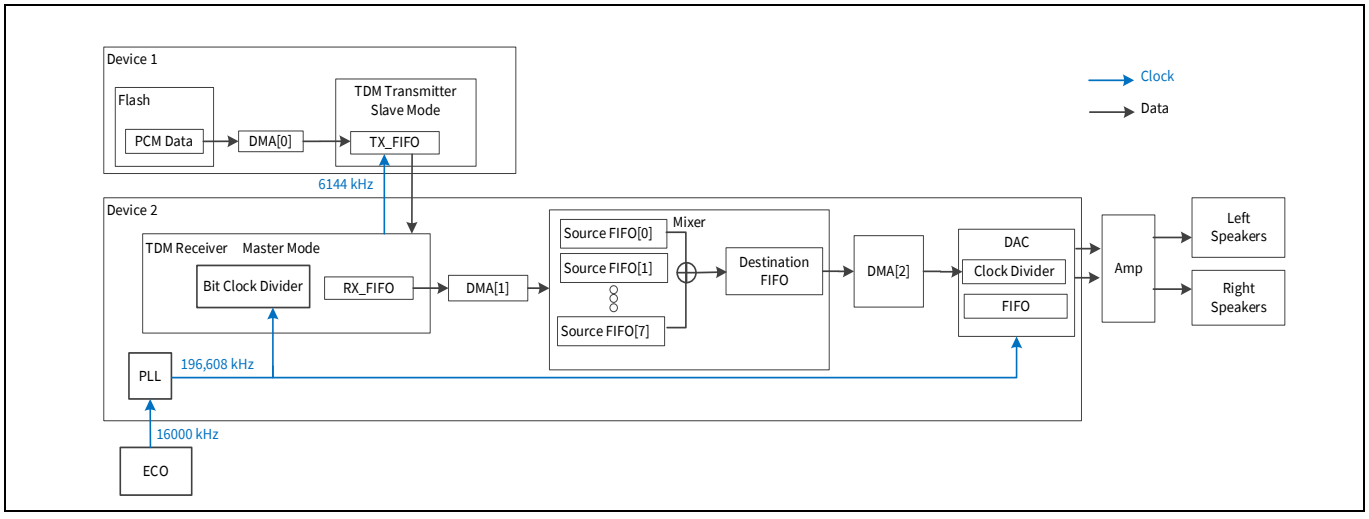

Figure 11 Block Diagram

Application

Figure 12 provides the settings necessary for high-frequency root clock setting, port setting, interrupt setting, DMA setting, I²S setting, mixer setting, PWM interface setting, and Trigger MUX setting, and illustrates the procedure for the use case.

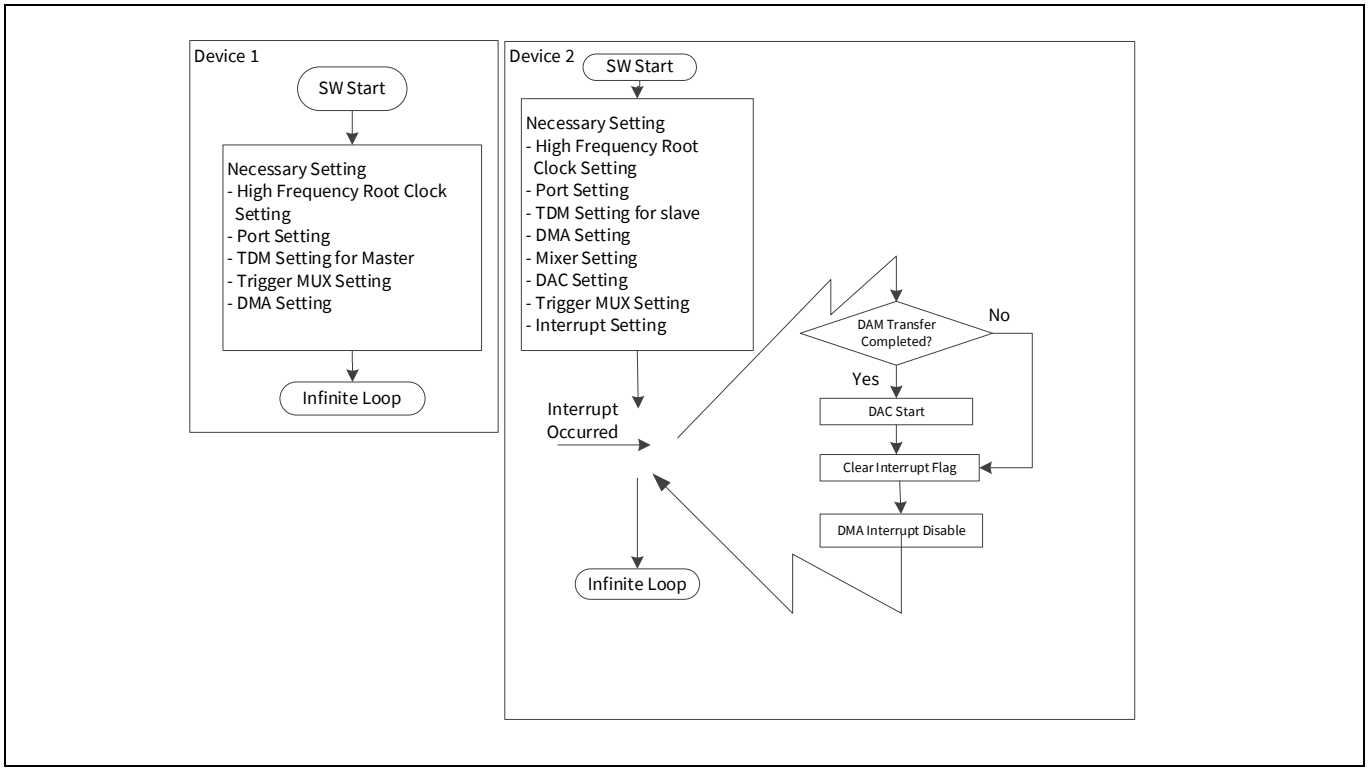

Figure 12 DAC Output

3.3.1.1 High-Frequency Root Clock Setting

This section explains the clock that can be used. Traveo II family has several high-frequency root clocks (CLK_HF). Each CLK_HF has a destination on the device.

This use case uses three high-frequency root clocks (CLK_HF):

- CLK_HF for TDM - Enables CLK_HF [5] when the TX_IF_CTL.CLOCK_SEL register is “0” (clk_if_srss [0]).
- CLK_HF for Mixer - Enables CLK_HF [5] when Mixer channel 0 is enabled (clk_if_srss [0]).
- CLK_HF [5] for audio DAC - Enables CLK_HF [5]

CLK_HF [5] is used in CYT3/4 series. For other series and more details on clocks, see the “Clock Subsystem” chapter of the **Architecture TRM** and **Datasheet**.

3.3.1.2 Port Setting

This section explains the port signal used in this use case.

Table 3 Table 3. Port Setting of Use Case 3

Component	Port Signals
TDM receiver (RX)	TDM0_TDM_RX_SCK, TDM0_TDM_RX_MCK, TDM0_TDM_RX_FSYNC, TDM0_TDM_RX_SD
TDM transmitter (TX)	TDM0_TDM_TX_SCK, TDM0_TDM_TX_MCK, TDM0_TDM_TX_FSYNC, TDM0_TDM_TX_SD
Mixer	MIXER_MCK, MIXER_SCK, MIXER_WS, MIXER_SD
DAC	DAC_LOUT, DAC_ROUT, DAC_LCOM, DAC_RCOM.

To set the drive mode, the interrupt mask, and the edge detect at the port setting, see the “IO Subsystem” chapter of the **Architecture TRM**.

3.3.1.3 Interrupt Setting

Use case 3 uses DMA interrupt for start DAC.

See the “Interrupts” chapter of the **Architecture TRM** for details on the vector number of the DMA interrupt and the procedure to configure the interrupt priority, vector address, and enabling/disabling of the interrupts.

3.3.1.4 DMA Setting

Use case 3 uses DMA to transfer the data.

See the “Direct Memory Access” chapter of the **Architecture TRM** to set DMA.

3.3.1.5 Trigger MUX Setting

Use case 3 uses trigger mux to kick DMA by the FIFO level interrupt.

See the “Trigger Multiplexer” chapter of the **Architecture TRM** to set the trigger group and trigger number.

3.3.1.6 TDM Setting

Figure 13 shows the configuration of Tx and Rx that need to be set on Device 1 and Device 2.

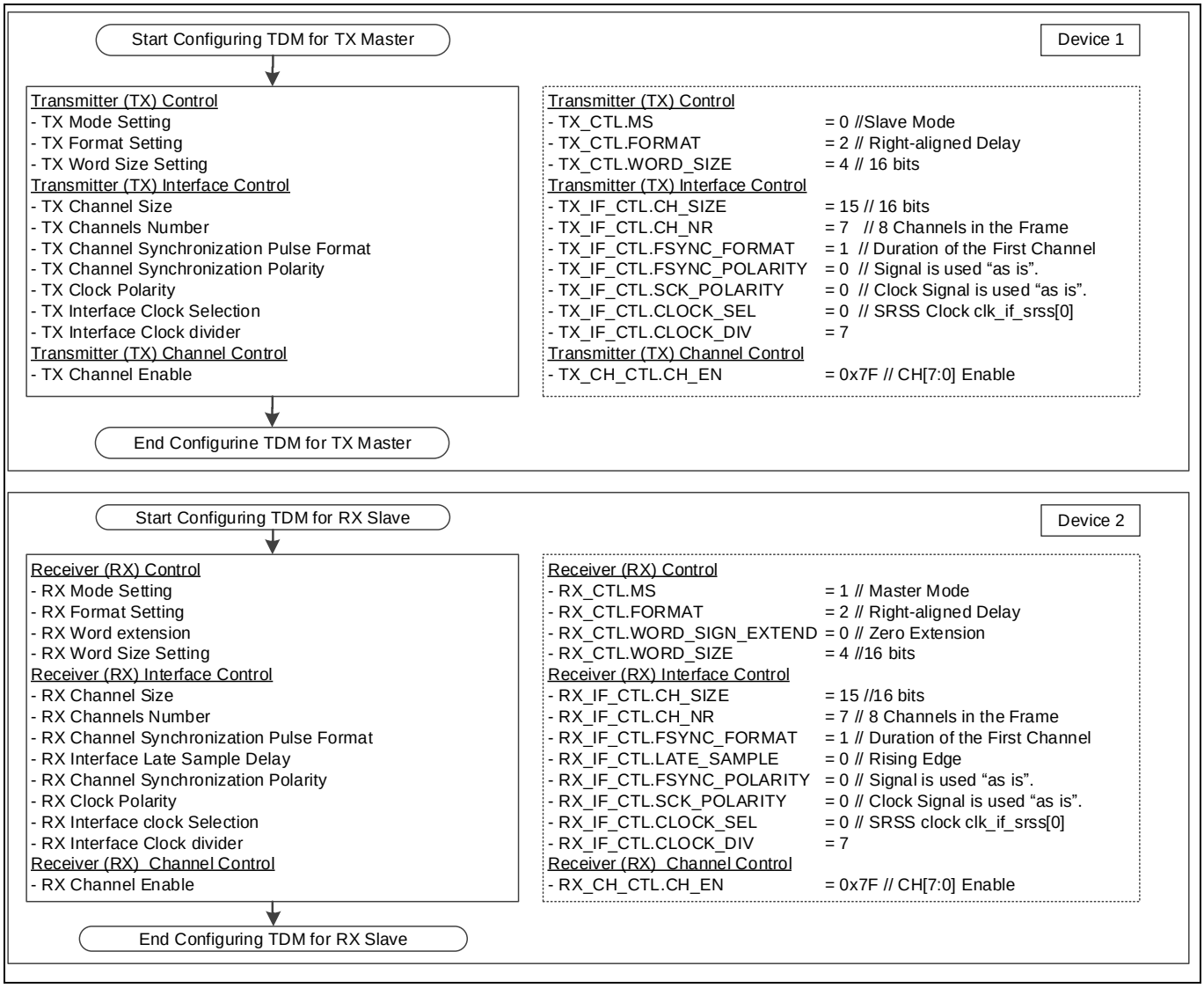

Figure 13 TDM Setting

3.3.1.7 Mixer Setting

The configuration of the mixer in use case 3 is similar to **3.2.1.7 Mixer Setting**. See **Figure 9** to set the mixer.

3.3.1.8 DAC Setting

Figure 14 shows the configuration that needs to be set.

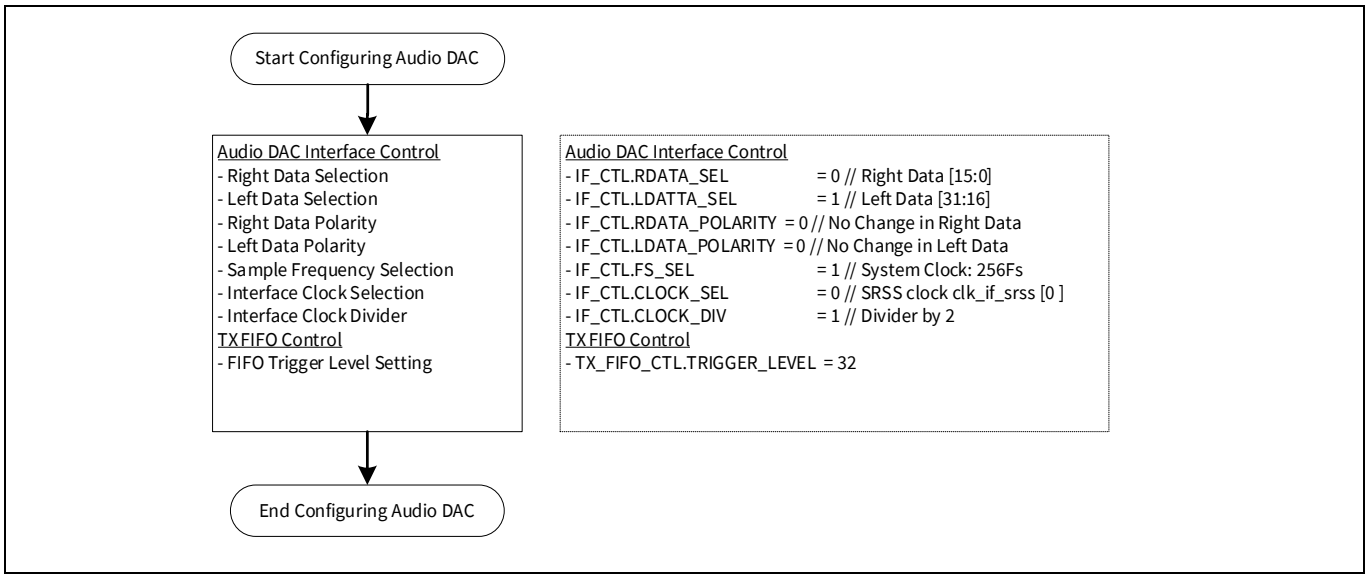

Figure 14 Audio DAC Setting

In this use case, the TX_FIFO_CTL.TRIGGER_LEVEL register equals 32, which means that when the Tx FIFO has fewer than 32 entries, a transmitter trigger event is generated.

For more information on interface formats, see the “Sound Subsystem” chapter of the **Architecture TRM**.

3.3.1.9 Interrupt Routine

Figure 12 shows the interrupt routine for the audio DAC.

INTR_TX_MASKED.COMPLETE is an interrupt-masked register used to check whether the interrupt occurs. If the register value equals 1, it means that the DMA Transfer Complete interrupt has occurred. Then, you clear the interrupt flag and wait for the next interrupt to occur.

3.4 Use Case 4

This use case uses a single PDM pattern generator to output the PDM data to the PDM receiver FIFO. Then, it converts the PDM data to PCM data. The PCM data will be played over speakers.

3.4.1 Description of Use case 4

This use case uses a single PDM pattern generator to generate a 1.910-kHz sine wave PDM sequence. The generated PDM data is further processed using filters such as CIC filter, FIR filter, and DC block filter to produce PCM in Rx FIFO. Then, the PDM Rx FIFO data is written to the I^2S TX FIFO via DMA kicked by the PDM Rx Trigger FIFO level.

Finally, the sound is played over the left and right speakers, which output 1.910 kHz.

In this use case, the clock is an internal clock derived from CLK_HF [5] (Pre-condition CLK_HF [5] = 196,608 kHz). The PDM clock is 1536 kHz, because it uses data with a 48-kHz sampling rate and data is 16-bit size ($48000\ \text{Hz} \times 16\ \text{bit} \times 2\ = 1536\ \text{kHz}$). CLK_HF [5] is used in CYT3/4 series.

Application

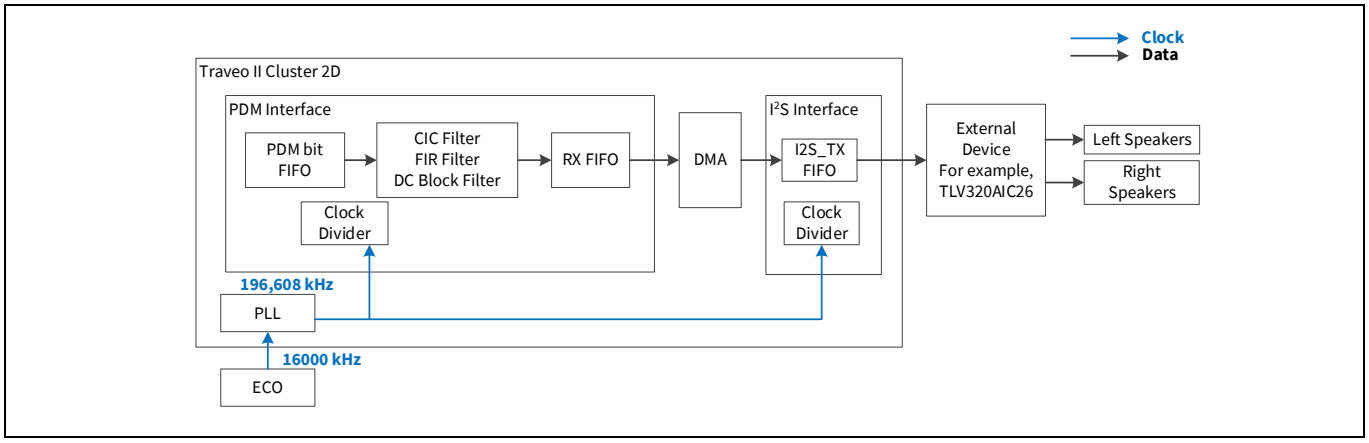

Figure 15 Block Diagram

Figure 16 provides the settings necessary for high-frequency root clock setting, port setting, interrupt setting, DMA setting, I²S setting, PDM interface setting, and Trigger MUX setting, and illustrates the procedure for the use case.

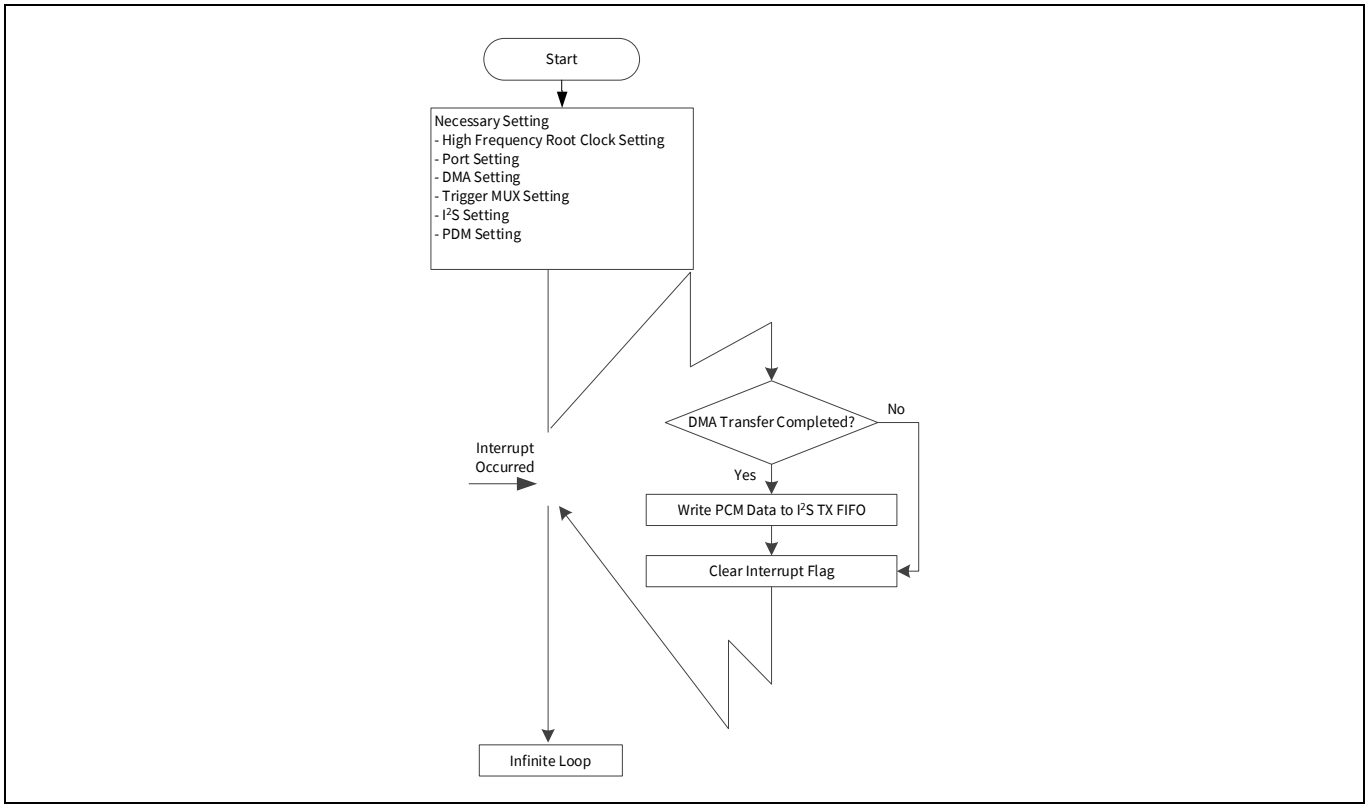

Figure 16 PDM Output Processing

3.4.1.1 High-Frequency Root Clock Setting

This section explains the clock that can be used. Traveo II family has several high-frequency root clocks (CLK_HF). Each CLK_HF has a destination on the device.

This use case uses two high-frequency root clocks (CLK_HF):

- CLK_HF for PDM - Enables CLK_HF [5] when the TX_IF_CTL.CLOCK_SEL register is “0” (clk_if_srss [0]).
- CLK_HF for I^2S - Enables CLK_HF [5] when Mixer channel 0 is enabled (clk_if_srss [0]).

CLK_HF [5] is used in CYT3/4 series. For other series and more details on clocks, see the “Clock Subsystem” chapter of the **Architecture TRM** and **Datasheet**.

3.4.1.2 Port Setting

Do the following to configure the PDM port setting:

- Configure the PDM port
  - PDM_CLK
  - PDM_DATA

To set the driver mode, interrupt mask, and edge detect as the port setting, see the “IO Subsystem” chapter of the **Architecture TRM**.

3.4.1.3 DMA Setting

Use case 4 uses DMA to transfer data from the PDM FIFO to I^2S FIFO.

See the “Direct Memory Access” chapter of the **Architecture TRM** to set DMA.

3.4.1.4 Trigger MUX Setting

Use case 4 uses trigger mux to kick DMA using the PDM FIFO level interrupt.

See the “Trigger Multiplexer” chapter of the **Architecture TRM** to set trigger group and trigger number.

3.4.1.5 I2S Setting

The configuration of I^2S in Use case 4 is similar to **Use Case 2**. See **Figure 8** to set I^2S.

3.4.1.6 PDM Setting

Figure 17 shows the configuration that needs to be set:

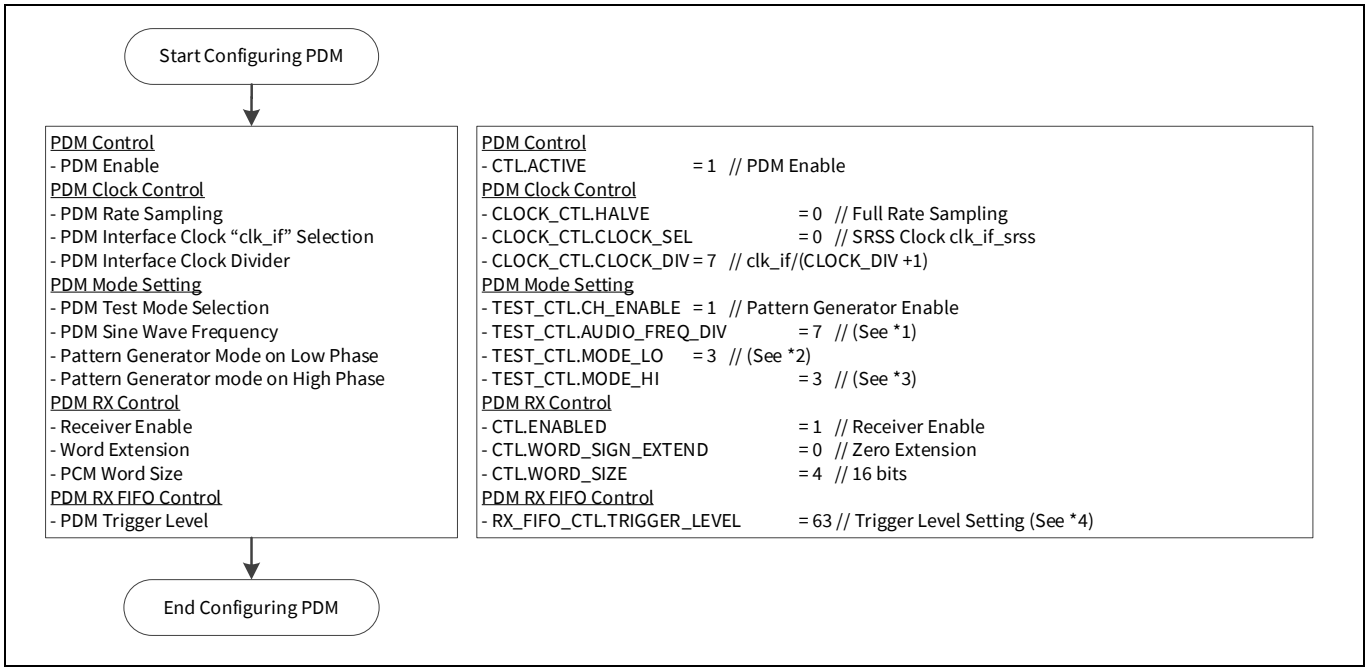

Figure 17 Configure for PDM

*1: TEST_CTL.AUDIO_FREQ_DIV

This clock divider is used to determine the frequency of the sine wave generated by the pattern generator.

The formula is:

$$\frac{\text{PDM Clock}}{2\pi \times 2^{AUDIO\_FREQ\_DIV}} = Sine\ wave\ Frequency$$

Where,

PDM Clock = 1.536 kHz and Sine Wave Frequency = 1.91 kHz.

By substituting values:

$$\frac{1536\text{kHz}}{2\pi \times 2^{7}} = 1.91kHz$$

Thus, the value of AUDIO_FREQ_DIV is derived as 7.

For more details, see the “Sound Subsystem” chapter of the **Architecture TRM**.

*2: TEST_CTL.MODE_LO = 3

TEST_CTL.MODE_LO specifies the type of pattern driven by the generator on the low phase.

- '0': Constant 0
- '1': Constant 1
- '2': Alternating 0 and 1
- '3': Sine wave

*3: TEST_CTL.MODE_HI = 3

TEST_CTL.MODE_HI specifies the type of pattern driven by the generator on the high phase.

Application

- '0': Constant 0
- '1': Constant 1
- '2': Alternating 0 and 1
- '3': Sinusoid

*4: TRIGGER_LEVEL = 63

TRIGGER_LEVEL is the set value of the trigger event condition.

- RX_FIFO_CTL.TRIGGER_LEVEL＝63

The trigger level is set as 63, which means that when the Rx FIFO is equal or greater than 63, an event trigger is generated.

For more information on interface formats, see the “Sound Subsystem” chapter of the **Architecture TRM**.

3.5 Hardware Design Guide

This section describes a hardware design example of the audio input / output using I2S mode of the TDM interface.

CYT4D has three TDM/I2S units, each of which supports eight multiple TDM modes. In other words, a total of 24 channels of simultaneous audio output is possible. In addition, the I2S input / output can operate independently.

In this example, the audio DAC uses CIRRUS Logic CS4385A. The audio ADC uses CS5368 from CIRRUS Logic.

The entire block diagram is shown below.

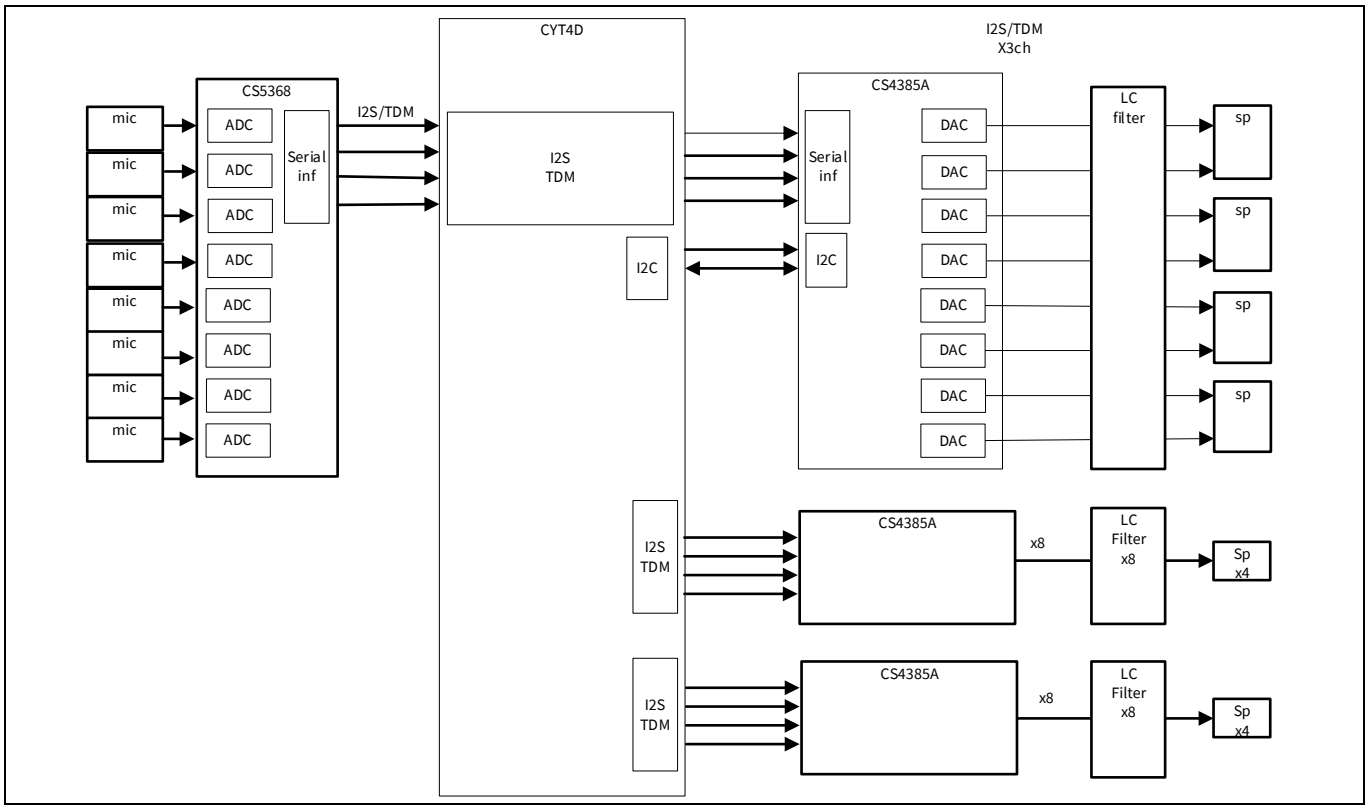

Figure 18 Entire Block Diagram

Application

The audio output interface is shown **Figure 19**. The LC filter is shown **Figure 21**.

The audio input interface is shown **Figure 20**. The MIC input is shown **Figure 22**.

"mic" in the figure is an abbreviation for microphone. "sp" in the figure is an abbreviation for Speaker.

3.5.1 Audio Output Interface

This is the audio output interface.

CS4385A is a complete 8-channel digital-to-analog system. This D/A system includes digital de-emphasis, half-dB step size volume control, ATAPI channel mixing, and selectable fast and slow digital interpolation filters followed by an oversampled, multi-bit delta sigma modulator which includes mismatch-shaping technology that eliminates distortion due to capacitor mismatch. Following this stage is a multi-element switched-capacitor stage and low-pass filter with differential analog outputs. It outputs an 8-channel analog signal using the I2S / TDM signal from the MCU.

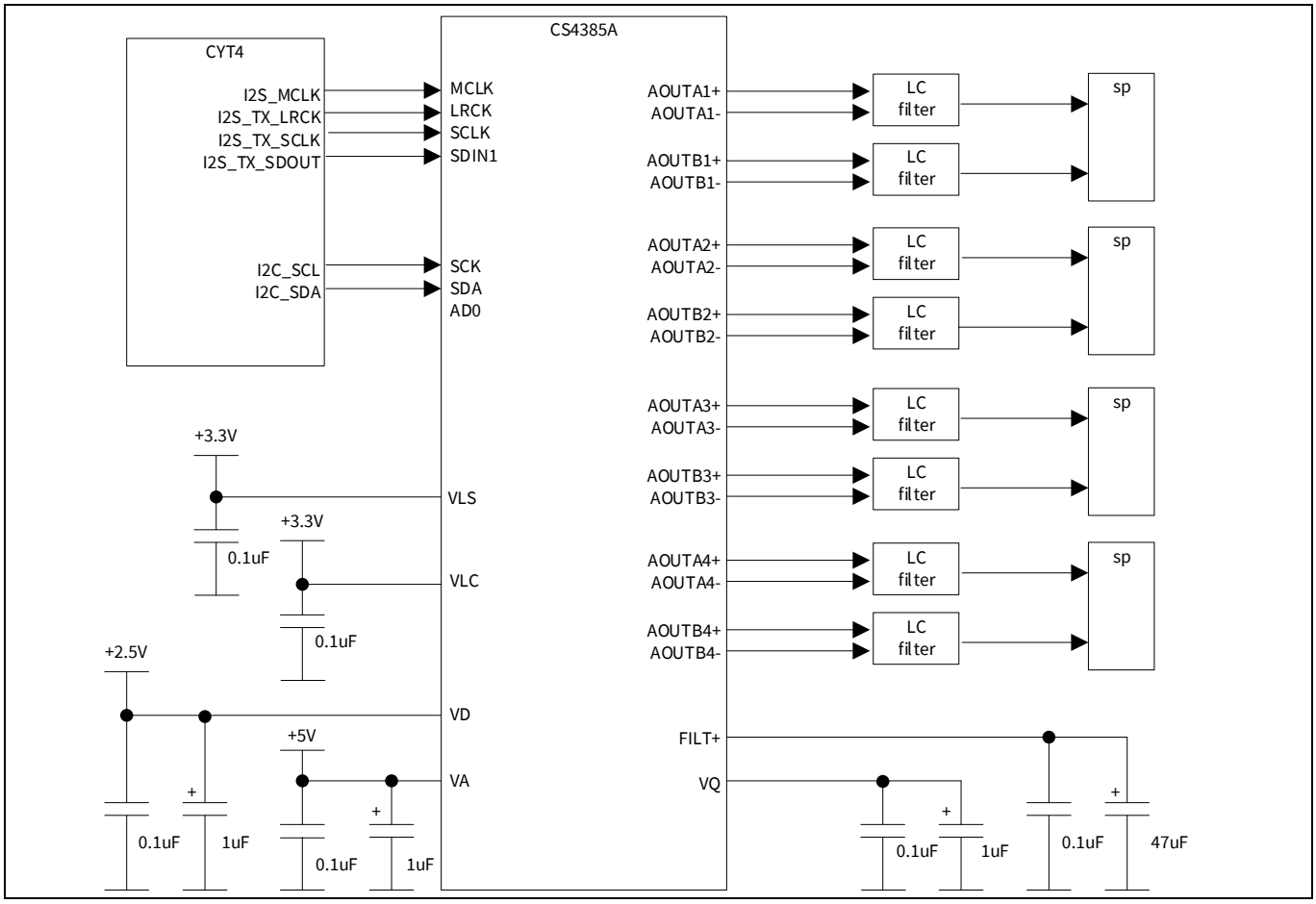

Figure 19 Audio Output Interface Block Diagram

See the CS4385A device datasheet for more details.

3.5.2 Audio Input Interface

This is the audio input interface.

CS5368 is a complete 8-channel analog-to-digital converter for digital audio systems. It performs sampling, analog-to-digital conversion, and anti-alias filtering, generating 24-bit values for all 8-channel inputs in serial form at sample rates up to 216 kHz per channel.

The analog input signal from the 8-ch microphone is multiplexed and input to the MCU using the I2S / TDM signal.

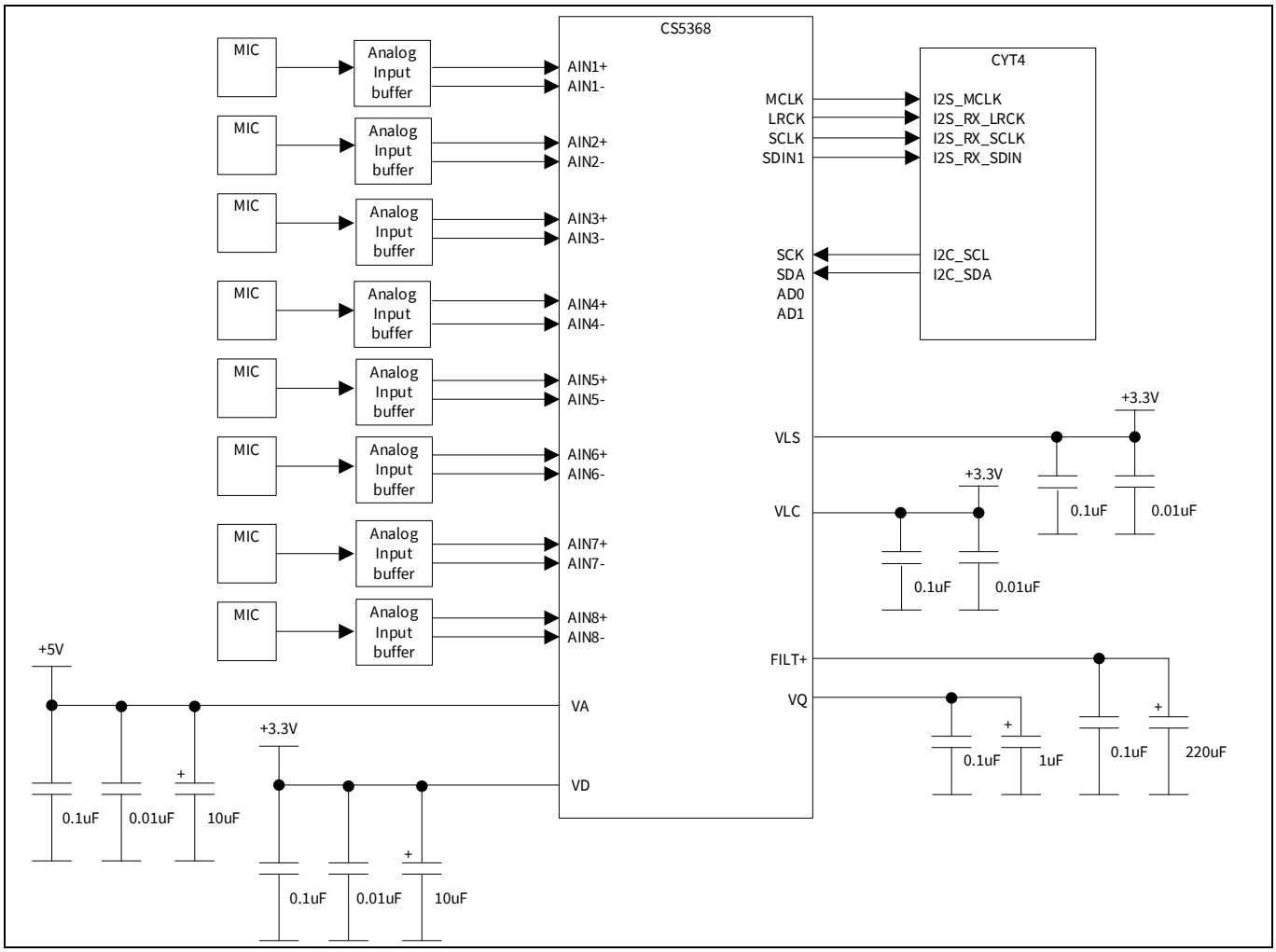

Figure 20 Audio Input Interface Block Diagram

See the CS5368 device datasheet for more details.

3.5.3 Audio Output Circuit (LC Filter)

This is example of an audio output circuit.

The analog output is designed according to the CS4385A datasheet. The output circuit includes an active 2-pole, 50-kHz filter which utilizes the multiple-feedback topology.

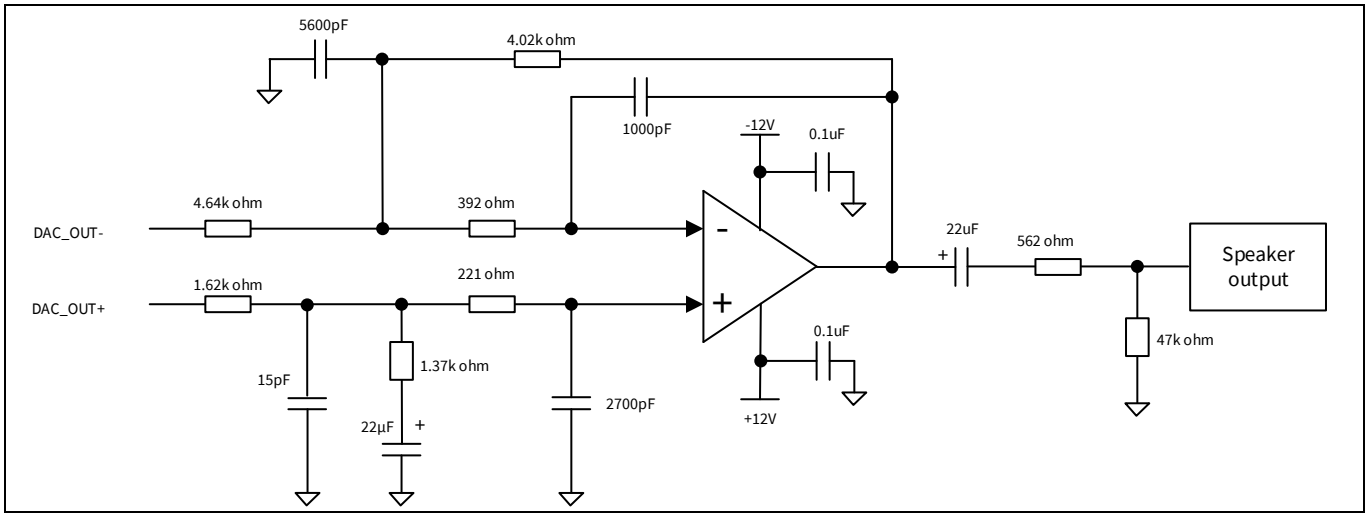

Figure 21 Example of Audio Output Circuit

See the CS4385A device datasheet for more details.

3.5.4 Audio Input Circuit (Analog Input Buffer)

This is example of an audio input circuit.

The analog input is an example of an active low-noise, single-ended-to-differential analog input buffer shown in **Figure 22**. Alternative active or passive, single-ended or differential topologies may be used as cost dictates, but it may compromise the high performance of CS5368. Optimum device performance is met by buffering CS5368 with a low-noise structure that is stable with a 2700-pF output load.

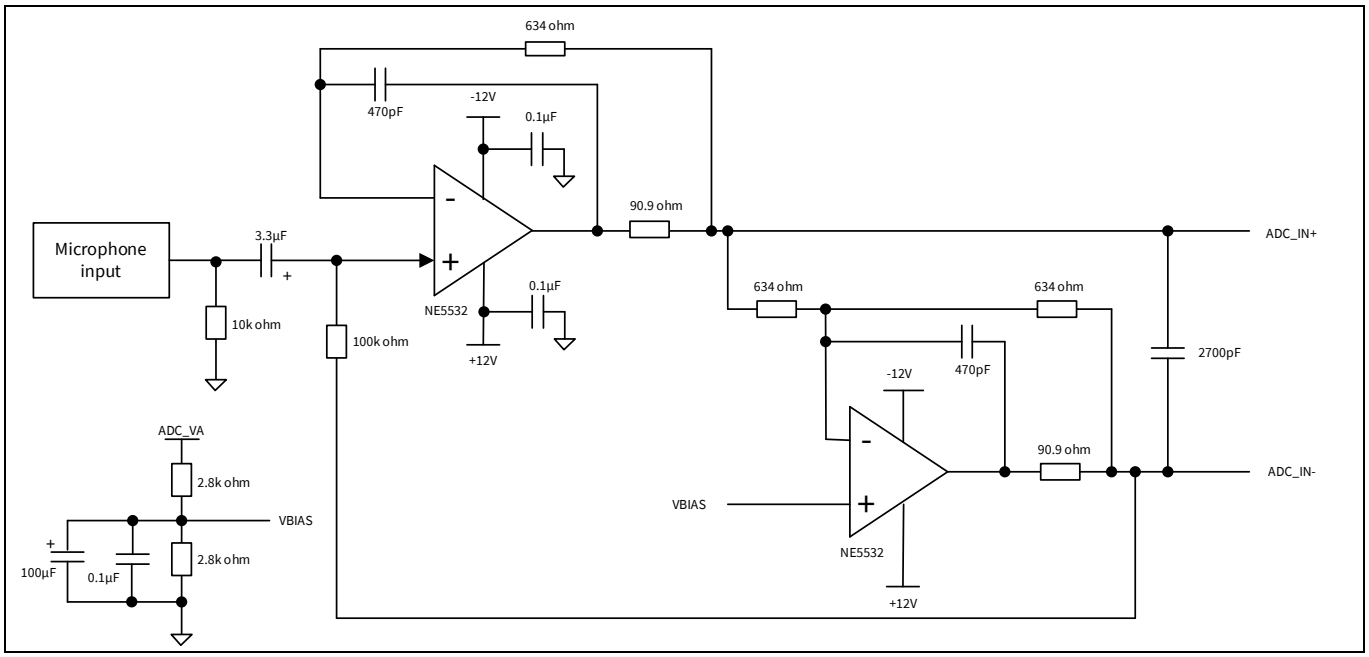

Figure 22 Example of Audio Output Circuit

See the CS5368 device datasheet for more details.

4 Glossary

Terms	Description
Amp	Amplifier.
CLK_HF	High-frequency root clock
DAC	Digital analog converter
DMA	Direct memory access
FPU	Single/double-precision floating point unit
Source FIFO	FIFO used to store data before combining them.
SRSS	System Resources Sub-System Core Registers
RX FIFO	FIFO for receiver
TX FIFO	FIFO for transmitter

5 Related Documents

The following are the Traveo II family series datasheets and Technical Reference Manuals. Contact **Technical Support** to obtain these documents.

- Device datasheet
  - CYT3DL Datasheet 32-Bit Arm® Cortex®-M7 Microcontroller Traveo™ II Family
  - CYT4DN Datasheet 32-Bit Arm® Cortex®-M7 Microcontroller Traveo™ II Family
- Architecture Technical Reference Manual (TRM)
  - Traveo™ II Automotive Cluster 2D Family Architecture Technical Reference Manual (TRM)
- Registers Technical Reference Manual (TRM)

  CYT3 Series
  - Traveo™ II Automotive Cluster 2D Registers Technical Reference Manual (TRM) for CYT3DL

  CYT4 Series
  - Traveo™ II Automotive Cluster 2D Registers Technical Reference Manual (TRM) for CYT4DN

Revision history

Document version	Date of release	Description of changes
**	06/05/2020	New Application Note.
*A	2021-02-16	Added Target device Modified Section 2.1 according to TRM Added Section 3.5 Hardware Design Guide MOVED TO INFINEON TEMPLATE.

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2021-02-16
Published by
Infineon Technologies AG
81726 Munich, Germany

Do you have a question about this document?
Go to www.cypress.com/support

Document reference
002-26043 Rev. *A

IMPORTANT NOTICE
The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.