How to Use I²S in Traveo II Family

About this document

Scope and purpose

AN224413 explains how to configure and use I²S in Traveo™ II Family MCU. The application note also explains the necessary settings for I²S such as port, clock, and interrupt. The document uses TLV320AIC26 (Audio Codec) as an example to explain the functions of I²S.

Associated Part Family

Traveo II Family CYT4B Series

Table of contents

1 Introduction

The Inter-IC Sound Bus (I²S) is a serial bus interface standard used to connect digital audio devices together. The specification is from Philips® Semiconductor (I²S bus specification: February 1986, revised June 5, 1996). In addition to the standard I²S format, the I²S block also supports the Left Justified (LJ) format and the Time Division Multiplexed (TDM) format.

This application note describes how to use I²S for Cypress Traveo II family CYT4B series MCUs. The application note uses TLV320AIC26 as an example to connect with I²S interface to record and play with headphones.

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

1.1 I²S structure

Figure 1, **Figure 2**, and **Figure 3** show the structure of I²S and three use cases of external and internal clock.

The I²S block configuration, control, and status registers, along with the FIFO data buffers are accessible through the AHB bus. AHB bus masters such as CPU and DMA can access the I²S registers through the AHB interface. See the **device datasheet** for information on port pin assignments of the I²S block signals and AC/DC electrical specifications.

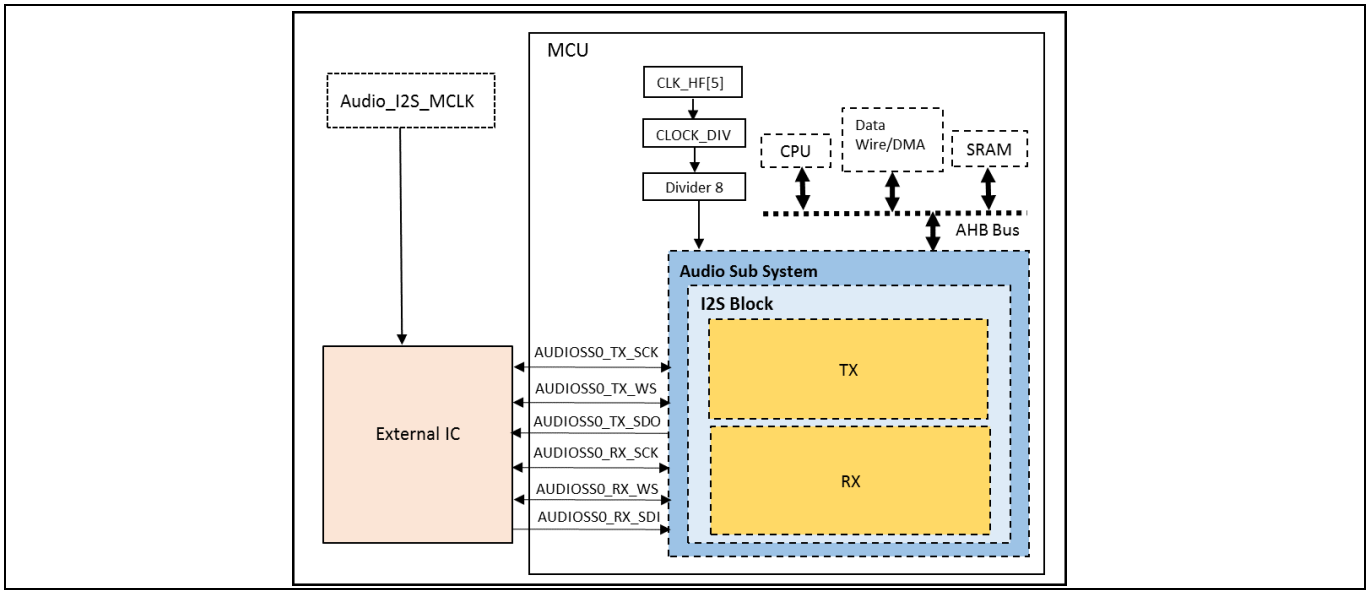

Figure 1 I²S block diagram for internal clock

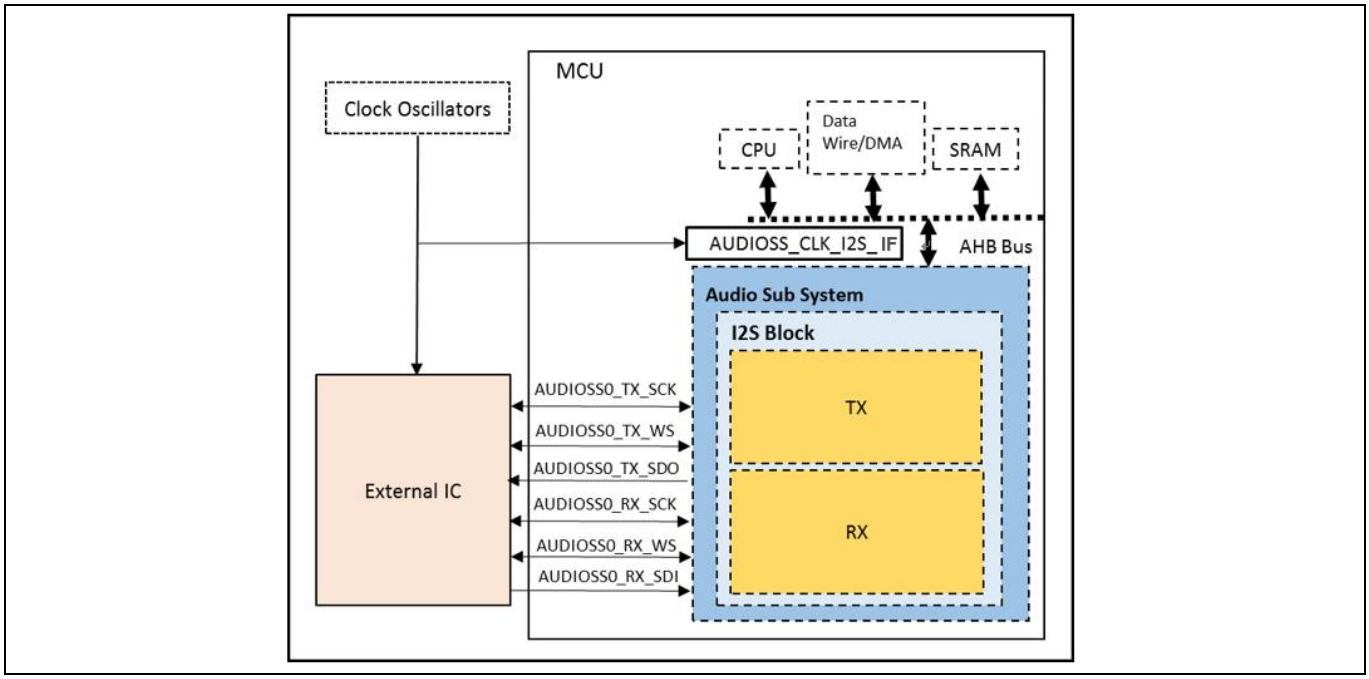

Figure 2 I²S block diagram for external clock

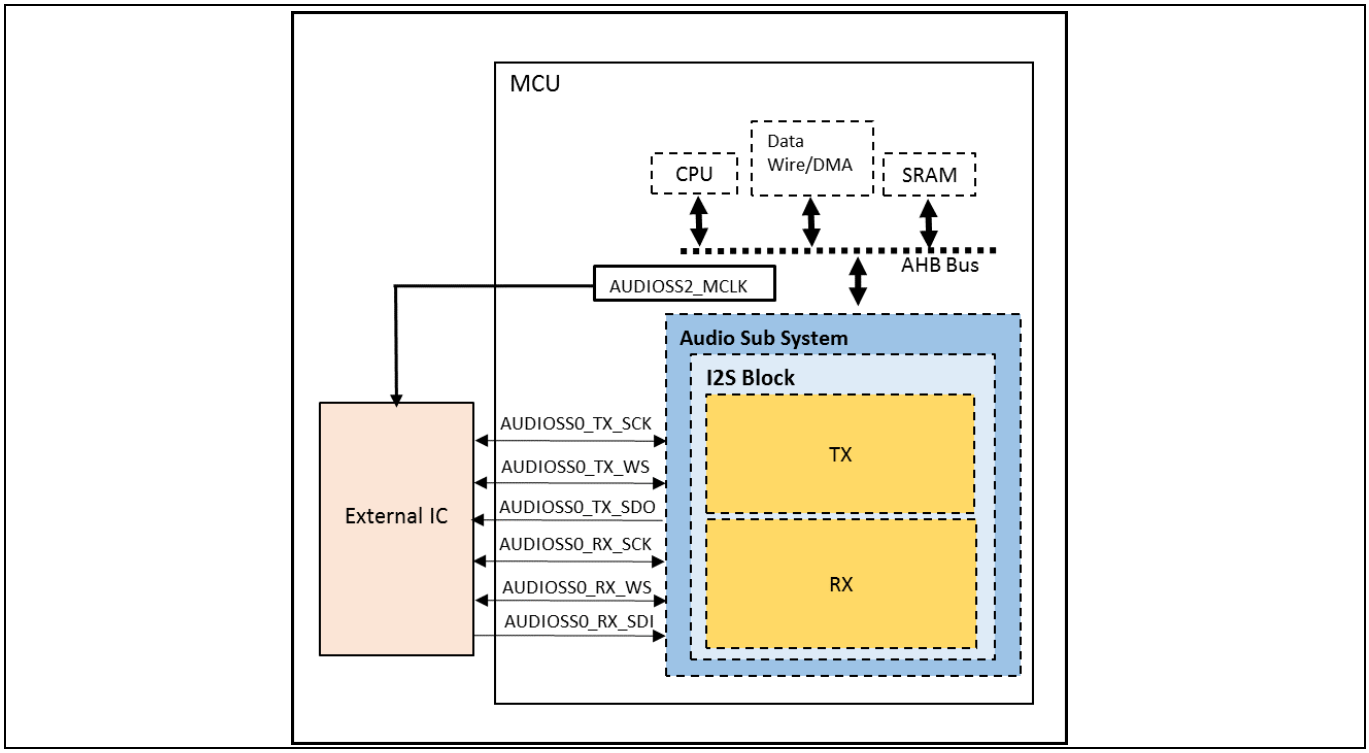

Figure 3 I²S block diagram for MCLK output

Figure 1, **Figure 2**, and **Figure 3** show the block diagrams of the I²S block of the Traveo II MCU (hereafter referred to as TVII), which consist of two sub-blocks-I²S Transmitter (TX) and I²S Receiver (RX). The digital audio interface format and master/slave mode configuration can be done independently for the TX and RX blocks. In the master mode, the word select (WS) and serial data clock (SCK) are generated by the I²S block in TVII. In the slave mode, the WS and SCK signals are input signals to the TVII I²S block and generated by the external master device.

1.1.1 Internal clock and external clock

Figure 1 shows the basic use case for I²S internal clock. In this case, the external device uses Audio_I2S_MCLK as clock and the I²S block of TVII uses HF_CLK [5] as clock. The external device and the TVII I²S block are not synchronized because they use different clocks; therefore, it may cause the FIFO to underflow or overflow. **Figure 2** shows external clock connections respectively. In this case, clock is the synchronization of the clock. Thus, there are no underflow or overflow due to the clock being out of sync.

The following are the settings required for internal clock:

- CTL.TX_ENABLE and RX_ENABLE = 1 (I²S is enabled)
- CLOCK_CTL.CLOCK_SEL = 0 (I²S clock is from internal clock: CLK_HF [5])
- CLOCK_CTL.CLOCK_DIV = 0, 1, 2… 63 (Division ratio: 1, 2, 3…64)

The following are the settings required for external clock:

- CTL.TX_ENABLE and RX_ENABLE = 1 (I²S is enabled)
- CLOCK_CTL.CLOCK_SEL = 1 (I²S clock is from external clock: AUDIO_CLK_I2S_IF)
- CLOCK_CTL.CLOCK_DIV = 0, 1, 2… 63 (Division ratio: 1, 2, 3…64)

To understand the functionality described and terminology used in this application note, see the “Audio Subsystem” chapter in the **Architecture TRM**.

1.1.2 MCLK output function

This section explains the third use case (see **Figure 3**), where the I²S unit generates MCLK output signal for the external audio DAC.

In **Figure 3**, AUDIOSS2_MCLK is used as the output for the external Audio DAC. In this use case, the MCLK output pin is active before I²S audio started (PLL start-up time of external IC).

MCLK output signal is generated only if the following conditions are met:

CTL.TX_ENABLE and CTL.RX_ENABLE = 1 (I²S is enabled)

- CLOCK_CTL.CLOCK_SEL = 0 (I²S clock is from internal clock: CLK_HF [5])
- CLOCK_CTL.MCLK_DIV = 0, 1, 2 or 3 (Division ratio: 1, 2, 4, or 8)
- CLOCK_CTL.MCLK_EN = 1 (MCLK output enabled)

Figure 4 shows the MCLK output function.

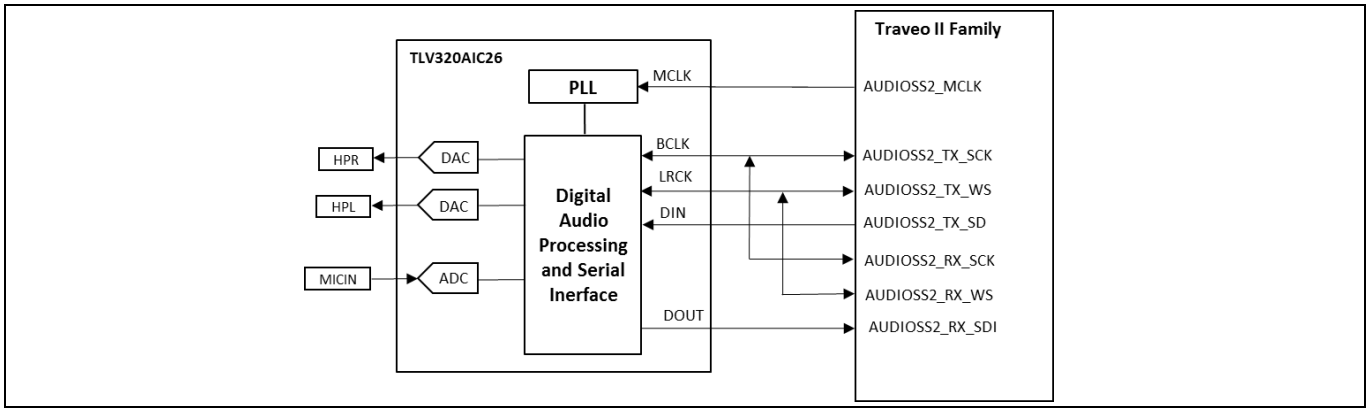

Figure 4 MCLK output function

For more information on Interface formations, see the “Audio Subsystem” chapter of the **Architecture TRM**.

1.2 Digital audio interface format

The TVII I^2S block supports the following digital audio interface formats:

- Standard I^2S
- LJ
- TDM

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

2 Operation overview

The external device TLV320AIC26 is used as an example to show the functions of the I²S block of TVII. The features of TLV320AIC26 of interest include Stereo Audio DAC and Mono Audio ADC. For more information, see **TLV320AIC26 datasheet**.

The TLV320AIC26 device converts the analog signal (for example, signal from a microphone) to the digital form and transmits it to the TVII MCU. TVII transmits the digital signals again to TLV320AIC26 which will do digital to analog conversion, and the analog signal can be played by a speaker. Thus, you can hear your voice from the headset while speaking over a microphone.

Figure 5 shows this example where the external device Audio Codec TLV320AIC26 converts the analog data from MICIN to RX_FIFO through DOUT. Then, TVII I²S writes the data to TX_FIFO and transmits it to TLV320AIC26 through DIN. TLV320AIC26 uses a DAC to convert this digital data. Then, TLV320AIC26 outputs the analog data via HPR/HPL.

TVII I²S block uses a standard I²S format. When TX is set in master mode, TX outputs WS, and SCK signals. When RX is set in slave mode, RX uses WS and SCK signals from TX. Sampling rate of I²S data transfer is set as 48 kHz. The clock is an internal clock derived from CLK_HF[5]. Word length and channel length of data transfer are 16 bits.

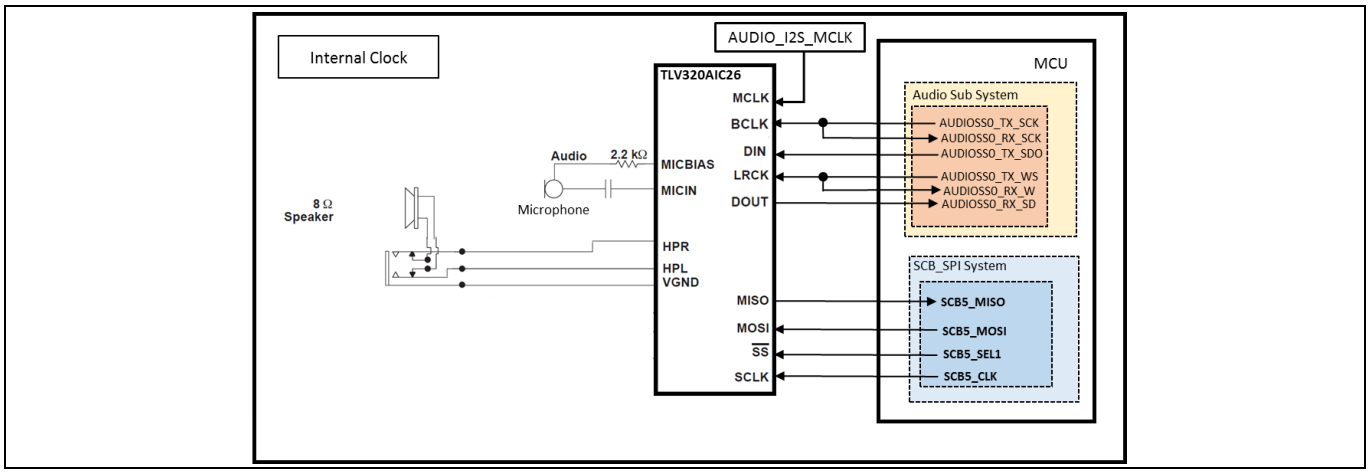

Figure 5 Example of use case

Control of the TLV320AIC26 and its functions is accomplished by writing to its registers. A simple command protocol is used to address the 16-bit registers. Registers control the operation of the A/D converter and Audio Codec. The control functions are accessed via a SPI Interface (in this example, it uses SCB5_SPI).

SCB5_SPI is used only to configure (write to its registers) the external device Audio Codec. The external Audio Codec device and I2S device use their own internal clocks AUDIO_I2S_MCLK and CLK_HF[5], respectively. AUDIO_I2S_MCLK is generated from an external oscillator.

Figure 6 shows an example of an external crystal oscillator.

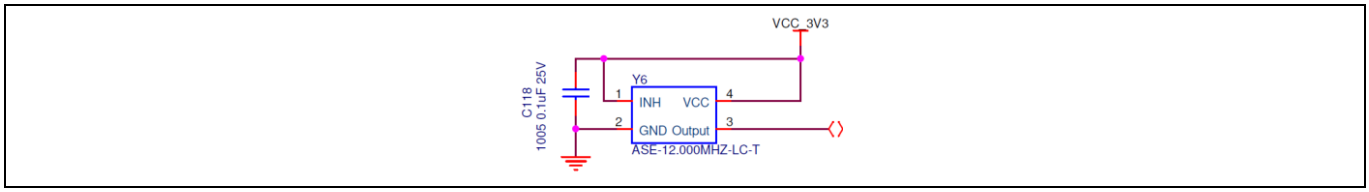

Figure 6 Example for external crystal oscillator

Operation overview

For more information about TLV320AIC26 and ASE series crystal oscillator, see **TLV320AIC26 datasheet** and **ASE Series datasheet**.

2.1 The signals of I²S block

This section explains the direction of signals.

In **Figure 7**, the TVII I²S TX block acts as the master and the codec, and the TVII I²S RX block act as slaves. Since the I²S TX block is the master, AUDIOSS_TX_SCK, AUDIOSS_TX_WS, AUDIOSS_RX_SCK, and AUDIOSS_RX_WS use the same clock.

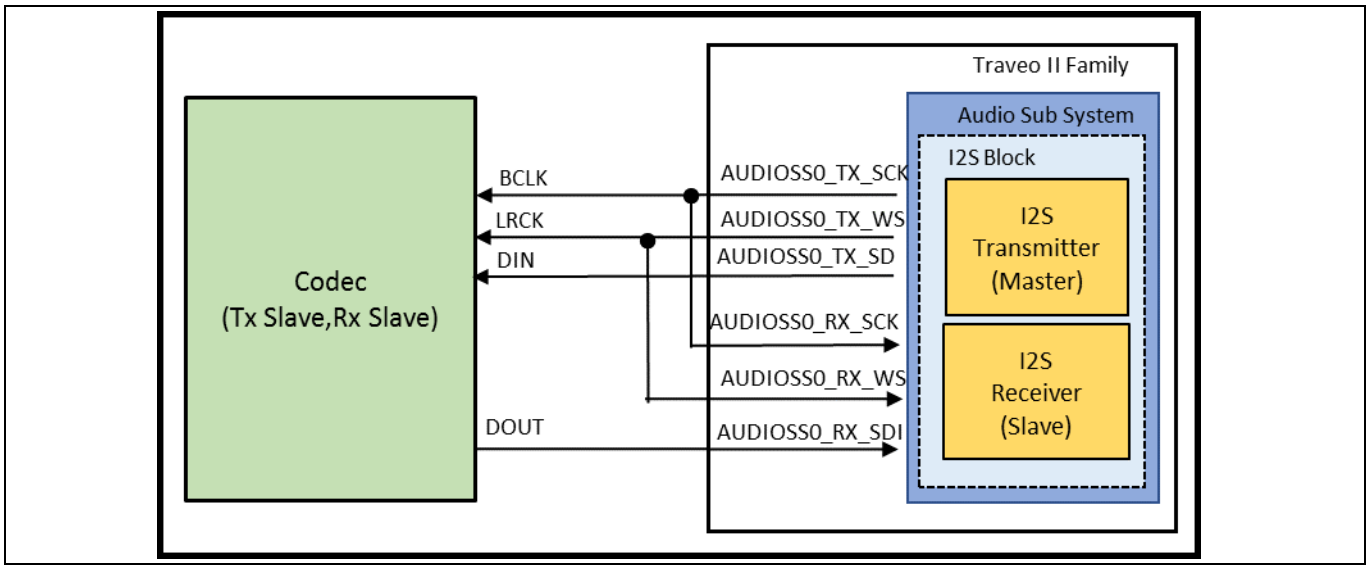

Figure 7 Connections for codecs with common WS and SCK signals for RX and TX

3 Setting procedure

Figure 8 provides the necessary settings for HF clock setting, port setting, interrupt setting, I2S setting, SPI setting, and audio codec setting and illustrates the procedure to write data for uses case. In this case, if an unintended interrupt occurs, the status is set to false. If a false status occurs, the software can perform appropriate error handling.

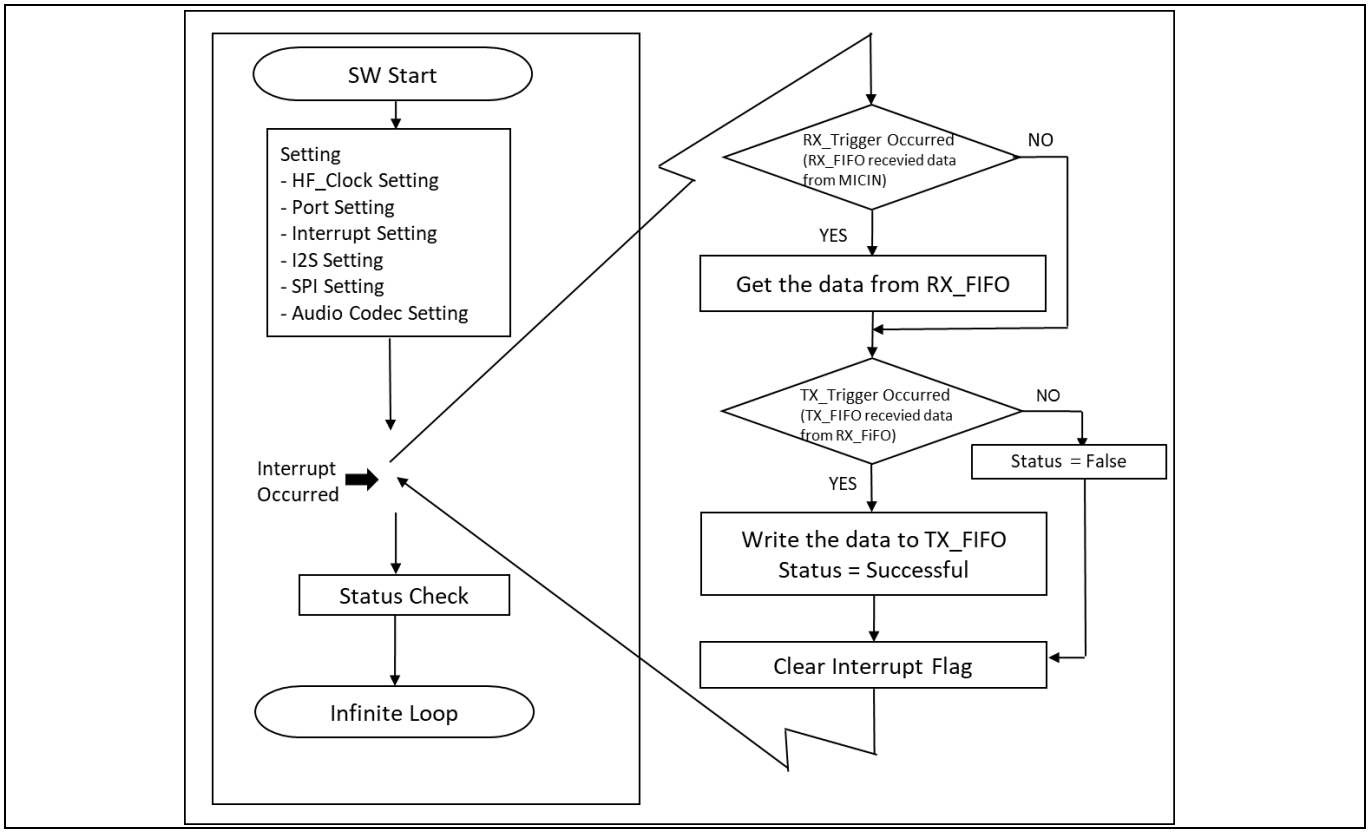

Figure 8 Write DATA to TX_FIFO

3.1 HF_Clock setting

This section explains the clock settings that can be used.

Traveo II family has several high-frequency root clocks (CLK_HF). Each CLK_HF has a destination on the device. In this use case, the I²S block uses CLK_HF [5] that is set to 24.576 MHz, for generating the 48 kHz sampling rate (details in section **3.5 Configure for I2S**).

The following are the settings and steps required for HF_Clock setting:

- HF_CLK5 for I²S
  (HF_CLK5 = 24.576 MHz)
- HF_CLK2 for SCB5_SPI
  1. Enable each HF_CLK
  2. Clock divider setting for SCB5_SPI
     - Assigns a programmable divider to a selected IP block
     - Sets the frequency and divider number (make sure that the frequency is defined)
     - Enables the selected divider

Setting procedure

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

3.2 Port setting

Follow this procedure to configure SPI and I²S port setting, using SCB5_SPI to address registers of TLV320AIC26:

- Configure SPI interface port
  - SCB5_SPI_MISO
  - SCB5_SPI_MOSI
  - SCB5_SPI_CLK
  - SCB5_SPI_SELECT1
- Configure I²S TX interface port
  - AUDIOSS0_TX_SCK
  - AUDIOSS0_TX_WS
  - AUDIOSS0_TX_SDO
- Configure I²S RX interface port
  - AUDIOSS0_RX_SCK
  - AUDIOSS0_RX_WS
  - AUDIOSS0_RX_SDI

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

3.3 Interrupt setting

The following are the interrupts that need to be set:

- SCB5_SPI interrupt
- I²S interrupt

See the Interrupts chapter of the **Architecture TRM** for details on the vector number of the I²S interrupt and the procedure to configure the interrupt priority, vector address, and enabling/disabling interrupts.

3.4 Configure for SPI

External device TLV320AIC26 communicates with MCU over SCB5_SPI interface and MCU acts as a SPI bus master.

All TLV320AIC26 control registers are programmed through a standard SCB5_SPI and the bitrate is set to 125000Hz.

To set SCB5_SPI, see the “Serial Communications Interface” chapter of the **Architecture TRM** to set SCB5_SPI.

3.5 Configure for I²S

The I²S block can support Standard I²S format, LJ format, and TDM format. In this use case, the I²S uses Standard I2S format. Also, the number of data bytes transmitted and received over I²S is 96 in each iteration.

Setting procedure

Figure 9 shows how to configure I²S.

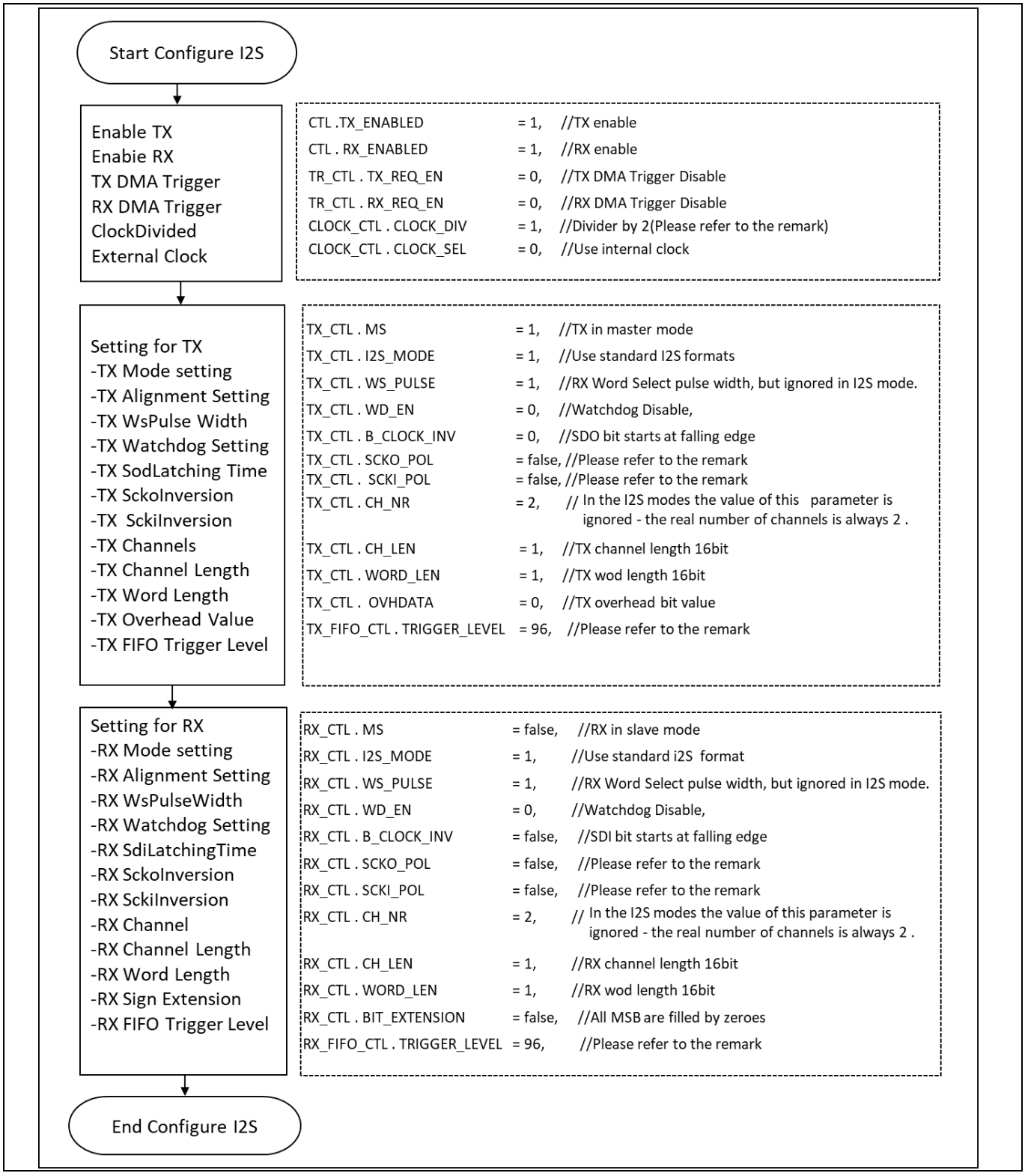

Figure 9 Example of configuration for I²S

Setting procedure

- CLOCK_CTL.CLOCK_DIV = 1

  This clock divider is used to generate the internal I²S master clock (MCLK_SOC).

  I²S device uses HF_CLK [5], which is 24.576 MHz and has a sampling rate set as 48 kHz.

  The formula is:

$$\frac{\text{Input Clock}}{8 \times CLOCK_{DIV} \times Bitsize} = Sampling\ Rate$$

  Where,

  8 is the second stage divider and is a constant.

  By substituting values:

$$\frac{24.576\text{MHz}}{8 \times CLOCK_{DIV} \times 32} = 48kHz$$

  Thus, the value of $CLOCK_{DIV}$ as derived as 2.

  For more details, see the “Clock System” chapter of the **Architecture TRM**.

- TX_CTL.SCKO_POL = false

  TX_CTL.SCKO_POL indicates the TX master bit clock polarity for transmitter control.

  – ‘false’: When transmitter is in master mode, serial data is transmitted from the falling bit clock edge
  – ‘true’: When transmitter is in master mode, serial data is transmitted from the rising bit clock edge

- TX_CTL.SCKI_POL = false

  TX_CTL.SCKI_POL indicates the TX slave bit clock polarity for transmitter control.

  – ‘false’: When transmitter is in slave mode, serial data is transmitted off the falling bit clock edge (according to the I²S Standard)
  – ‘true’: When transmitter is in slave mode, serial data is transmitted off the rising bit clock edge.

- RX_CTL.SCKO_POL = false

  RX_CTL.SCKO_POL indicates the RX SCKO polarity for Receiver Control.

  – ‘false’: When receiver is in master mode, serial data is captured by the rising bit clock edge (accordingly to the I²S Standard);
  – ‘true’: When receiver is in master mode, serial data is captured by the falling bit clock edge.

- RX_CTL.SCKI_POL = false

  RX_CTL.SCKI_POL indicates the RX slave bit clock polarity for Receiver Control.

  – ‘false’: When receiver is in slave mode, serial data is sampled on the rising bit clock edge.
  – ‘true’: When receiver is in slave mode, serial data is sampled on the falling bit clock edge.

- TRIGGER_LEVEL = 96

  TRIGGER_LEVEL is a set value of trigger event condition.

  – TX_FIFO_TRIGGER_LEVEL＝96

    When the trigger level is between 1 and 96, the number of data are stacked on TX FIFO, and an event trigger will be generated.

Setting procedure

- RX_FIFO_CTL.TRIGGER_LEVEL＝96

  When the trigger level is equal to or greater than 96, the number of data are stacked on RX FIFO, and an event trigger will be generated.

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

3.6 Audio codec setting

This section explains how to configure the audio codec.

Figure 10 shows an example for how to configure ADC and DAC in the audio codec.

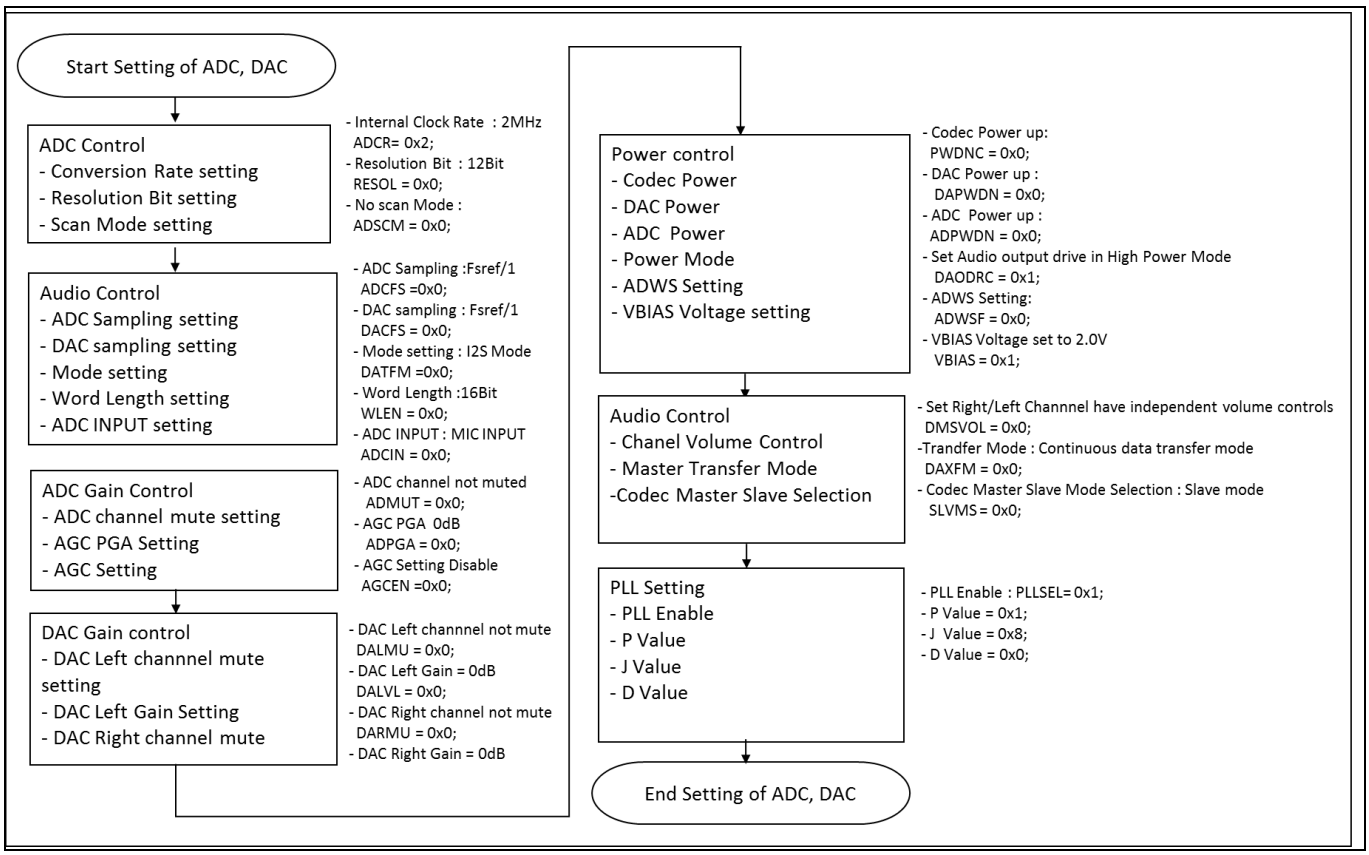

Figure 10 Example of setting for ADC and DAC in the audio codec

For more information on Audio Codec settings, see the **Related Documents**.

3.6.1 PLL setting

The reference sampling rate (Fsref) is necessary for the PLL and it must be set between 39 kHz and 53 kHz. Fsref can be set by the REG-06H/Page2 control register. For more information on sampling rate, see the **TLV320AIC26 datasheet**.Use the following formula to calculate the values of P, K, J, and D, when PLL is enabled.

$$\text{Fsref} = \frac{MCLK \times K}{2048 \times P}$$

Where,

P = 1, 2, 3 …,8

Setting procedure

K = J.D

J = 1, 2, 3, …,64

D = 0, 1, 2, …,9999

P, J, and D are set by registers. J is the integer part of K before the decimal point. D is a four-digit fractional part of K after the decimal point, including lagging zeros. P, J, and D are necessary values for the PLL.

In **Figure 10**, MCLK is 12 MHz. Fsref is set as 48 kHz and PLL is enabled. Therefore, the values are set as follows:

$$\text{Fsref} = \frac{MCLK \times K}{2048 \times P} = \frac{12000000 \times J.D}{2048 \times P} = 48000Hz$$

Where,

P = 1

J = 8

D = 1920

K = 8.192

You can confirm the validity of the values of P, J, and D from the following:

When PLL is enabled and D ≠ 0, the following condition of P must be satisfied:

$$10\ \text{MHz} \leq \frac{MCLK}{P} \leq 20\ MHz$$

After adding values to the formula, the condition should be:

$$10\ \text{MHz} \leq \frac{12MHz}{1} \leq 20\ MHz$$

Thus, P = 1 satisfies the condition.

The following is the condition to set K (= J.D):

$$80\ \text{MHz} \leq \frac{MCLK \times K}{P} \leq 110\ MHz$$

After adding values to the formula, the condition should be:

$$80\ \text{MHz} \leq \frac{12MHz \times 8.192}{1} \leq 110\ MHz$$

Thus, J = 8 and D = 192 satisfy the condition.

3.7 Interrupt routine

Figure 8 shows the interrupt routine for I²S.

Use the following register to check whether the interrupt has occurred:

- Register: INTR_MASKED.RX_TRIGGER

Setting procedure

This is Interrupt masked register. A register value of 1 means Rx trigger interrupt has occurred.

- Register: INTR_MASKED.TX_TRIGGER

  This is Interrupt masked register. A register value of 1 means Tx trigger interrupt has occurred.

In this use case, I²S gets the data from RX_FIFO when INTR_MASKED.RX_TRIGGER equals to 1.

I²S writes data to TX_FIFO when INTR_MASKED.TX_TRIGGER equals to 1 and status shows successful.

Status shows false when unexpected interrupt occurred.

Then, clear the interrupt flag and wait for the next interrupt to occur.

4 Glossary

Table 1 Glossary

Terms	Description
BCLK	Audio bit-clock
DIN	Audio data input
DOUT	Audio data output
HPL	Left channel audio output
HPR	Right channel audio output
LRCK	Audio DAC word-clock
MCLK	Master clock
MICBIAS	Microphone bias voltage
MICIN	Microphone input
MISO	SPI serial data output (Master in slave out)
MOSI	SPI serial data input (Master out slave in)
RX FIFO	FIFO for receiver
RX_SCK	I^2S serial clock for receiver
RX_SDI	I^2S serial data input for receiver
RX_WS	I^2S word select for receiver
SCLK	SPI serial clock input
SS	SPI slave select input
TX FIFO	FIFO for transmitter
TX_SCK	I^2S serial clock for transmitter
TX_SDO	I^2S serial data output for transmitter
TX_WS	I^2S word select for transmitter
VGND	Virtual ground for audio output

5 Related Documents

The following are the Traveo II Family Series datasheets and technical reference manuals:

- Traveo™ II Automotive Body Controller High Family Architecture Technical Reference Manual (Contact **Technical Support**)
- Traveo™ II Automotive Body Controller High Registers Technical Reference Manual (Contact **Technical Support**)
- Traveo™ Family Series Datasheet (Contact **Technical Support**)
- The Sample Software (Contact **Technical Support**)

The following is the TLV320AIC26 datasheet:

- TLV320AIC26: **Low Power Stereo Audio CODEC w/Headphone/Speaker Amp & 12-Bit Batt/Temp/Aux ADC**

The following is the ASE Series datasheet:

- ASE-12MHz-LC-T: **Crystal Oscillator Datasheet**

Revision history

Document version	Date of release	Description of changes
**	2019-06-04	New application note.
*A	2021-05-17	Updated to Infineon template.

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

Edition 2021-05-17
Published by
Infineon Technologies AG
81726 Munich, Germany

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

Document reference
002-24413 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.