CE218636 – PSoC 6 MCU Inter-IC Sound (I2S) Example

Objective

This example demonstrates how to use the I^{2}S hardware block in PSoC 6 to interface with an audio codec.

Overview

This code example shows how to play a short audio on a speaker or headphone. The application stores the audio data in the flash memory. This data is written to the output TX FIFO of the I^{2}S hardware block, which interfaces with an audio codec chip.

The audio codec chip converts a digital audio stream into analog. The chip is configured by the PSoC 6 over an I^{2}C interface. The code example includes the files codec.c/h, which wrap the configuration of the audio codec.

Requirements

Tool: PSoC Creator 4.2

Programming Language: C (ARM® GCC 5.4-2016-q2-update)

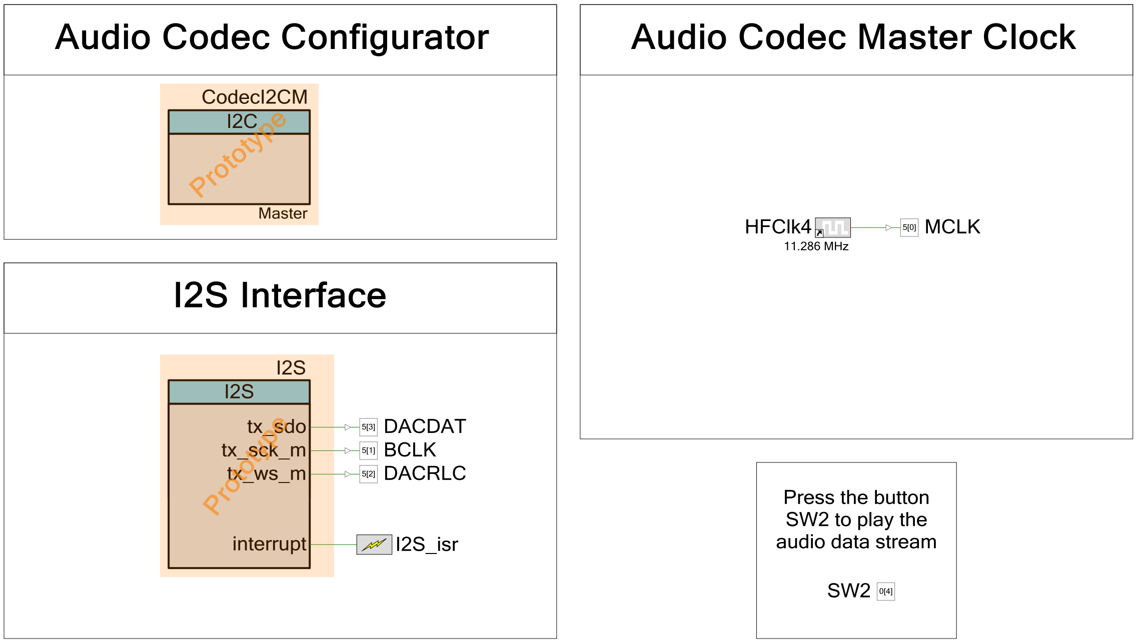
Associated Parts: All PSoC 6 parts with I^{2}S

Related Hardware: CY8CKIT-062 or CY8CKIT-062-BLE, CY8CKIT-028-TFT

Design

Figure 1 shows the PSoC Creator schematic for interfacing an audio codec with the I^{2}S Component.

Figure 1. Project Schematics

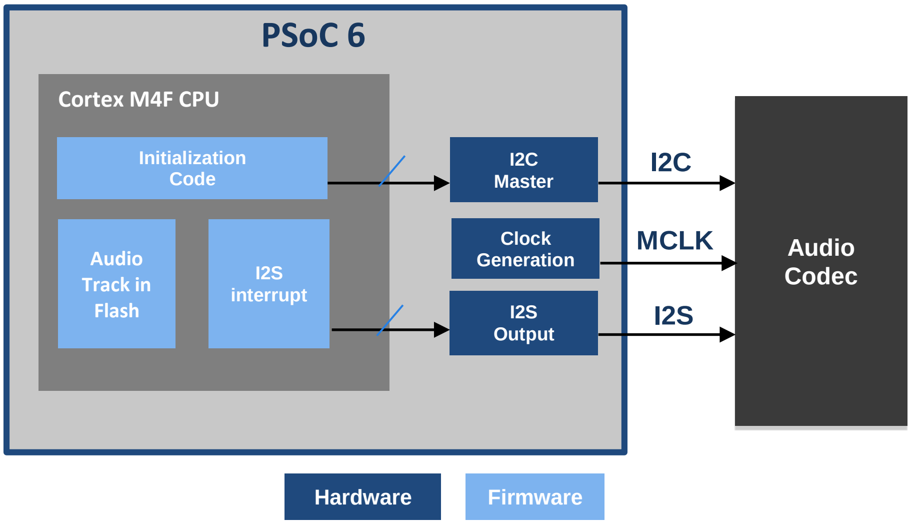


The CY8CKIT-028-TFT shield contains the audio codec AK4954A and an audio jack. This allows you to listen to any audio data stream transmitted to the audio codec over the I^{2}S interface. You can connect a speaker or headphone to the audio jack. The PSoC 6 in the CY8CKIT-062 kit generates the data over the I^{2}S. This is done by storing a short audio track in the flash memory, and writing it to the TX FIFO of the I^{2}S hardware block. The I^{2}S interface requires a continuous stream of data, which can be satisfied by feeding the TX FIFO with DMA transfers or interrupt service routine (ISR). In this example, an ISR is configured to be triggered when the TX FIFO is not full.

PSoC 6 also generates the clock fed to the audio codec. Based on the AK4954A datasheet, this codec requires an 11.2896 MHz MCLK and a 2.822 MHz BCLK to sample at 44.1 kHz. The code example contains an I^{2}C Master, through which the PSoC 6 configures the audio codec. The code example provides an API to easily configure the AK4954A.

Figure 2 shows the high-level block diagram of the interface between the PSoC 6 and the audio codec chip.

Figure 2. Block Diagram



Design Considerations

This code example runs on CY8CKIT-062-BLE or CY8CKIT-062 kits, which have a PSoC 6 device. To port the design to other PSoC 6 devices and kits, change the target device in the **Project > Device Selector**, and pin assignment in the Design Wide Resources.

Hardware Setup

This example requires the CY8CKIT-028-TFT shield to be connected to the CY8CKIT-062 (-BLE) kit.

Operation

1. Connect the CY8CKIT-028-TFT shield to the CY8CKIT-062 kit.
2. Build the “CE218636_I2S” project and program the CY8CKIT-062 kit. For more information on building projects and device programming, see PSoC Creator Help.
3. Connect a headphone or speaker to the audio jack of the CY8CKIT-028-TFT shield.
4. Press the CY8CKIT-062 SW2 button to play the short audio “PSoC Rocks”.

Components

Table 1 lists the PSoC Creator Components used in this example, as well as the hardware resources used by each.

Table 1. List of PSoC Creator Components

Component	Instance Name	Hardware Resources
I^{2}S	I2S	I^{2}S Block
I^{2}C (Master)	CodecI2CM	SCB
Clock	HFClk4	High Frequency Clock
Interrupt	I2S_isr	Interrupt [CM4]

Component	Instance Name	Hardware Resources
Digital Output Pin	MCLK	GPIO
Digital Output Pin	TX_SDO	GPIO
Digital Output Pin	TX_SCK	GPIO
Digital Output Pin	TX_WS	GPIO
Digital Input Pin	SW2	GPIO

Parameter Settings

This section shows the changed settings for various Components as well as the system clocks.

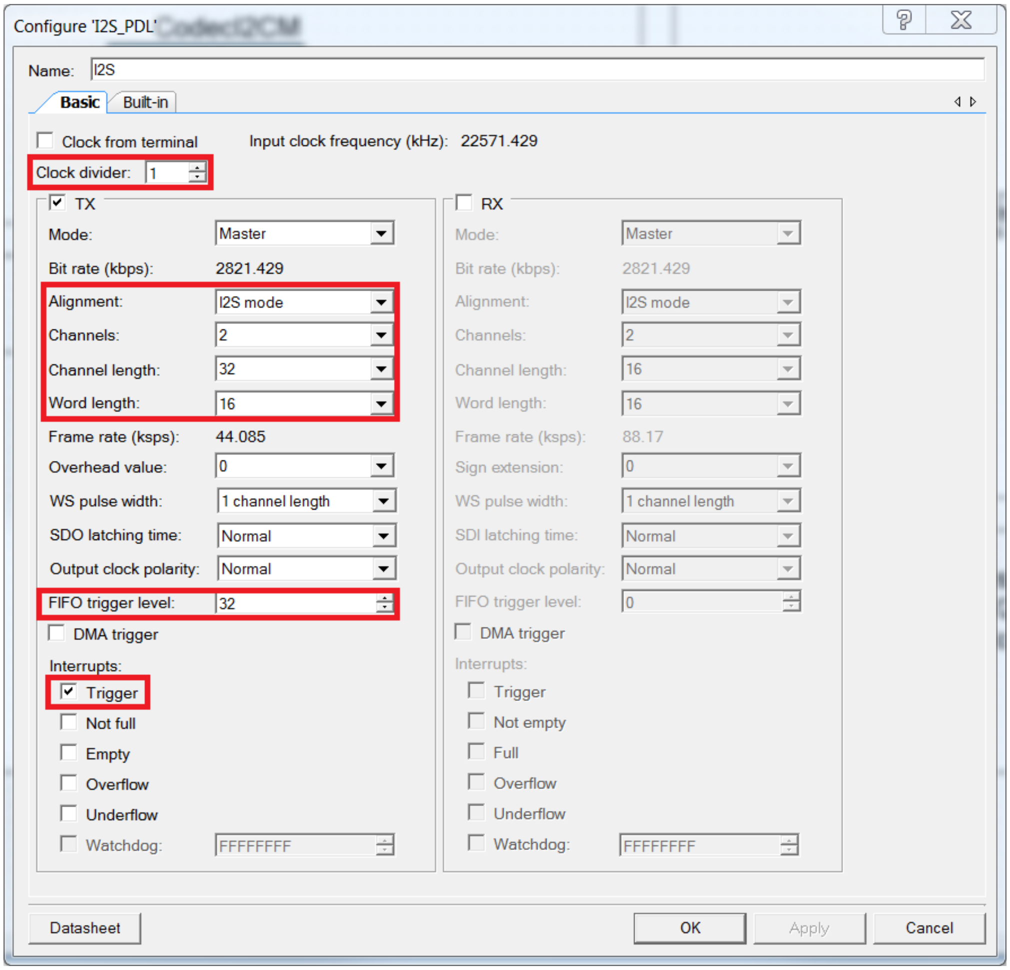
I^{2}S

The I^{2}S Component is configured for a frame rate of 44.1 ksps. Configure the HFClk1 clock to a specific frequency (see below). The Input clock frequency displayed in the Component is derived from HFClk1. The frame rate is calculated as:

$$Frame\ Rate\ (ksps) = \frac{HFClk1}{Clock\ divider\ \times 16 \times Channel\ Length}$$

Figure 3 shows the I^{2}S configuration window. Note that the frame rate is not exactly 44.1 ksps due to the PLL output frequency cannot achieve the desired frequency of 22.5792 MHz.

Figure 3. I^{2}S Component Configuration Window



Configure 'I2S_PDL'

Name: I2S

Basic | Built-in

□ Clock from terminal Input clock frequency (kHz): 22571.429

Clock divider: 1

☑ TX

Mode: Master
Bit rate (kbps): 2821.429
Alignment: I2S mode
Channels: 2
Channel length: 32
Word length: 16
Frame rate (ksps): 44.085
Overhead value: 0
WS pulse width: 1 channel length
SDO latching time: Normal
Output clock polarity: Normal
FIFO trigger level: 32
□ DMA trigger
Interrupts:
☑ Trigger
□ Not full
□ Empty
□ Overflow
□ Underflow
□ Watchdog: FFFFFFFF

□ RX

Mode: Master
Bit rate (kbps): 2821.429
Alignment: I2S mode
Channels: 2
Channel length: 16
Word length: 16
Frame rate (ksps): 88.17
Sign extension: 0
WS pulse width: 1 channel length
SDI latching time: Normal
Output clock polarity: Normal
FIFO trigger level: 0
□ DMA trigger
Interrupts:
□ Trigger
□ Not empty
□ Full
□ Overflow
□ Underflow
□ Watchdog: FFFFFFFF

I2S_isr

The I^{2}S Interrupt Type is configured to be **Auto-Select Trigger**.

HFClk4

The Component clock is configured to be 11.286 MHz, which is sourced from the PLL output and divided by 2.

SW2

The pin drive mode is configured to be Resistive Pull Up.

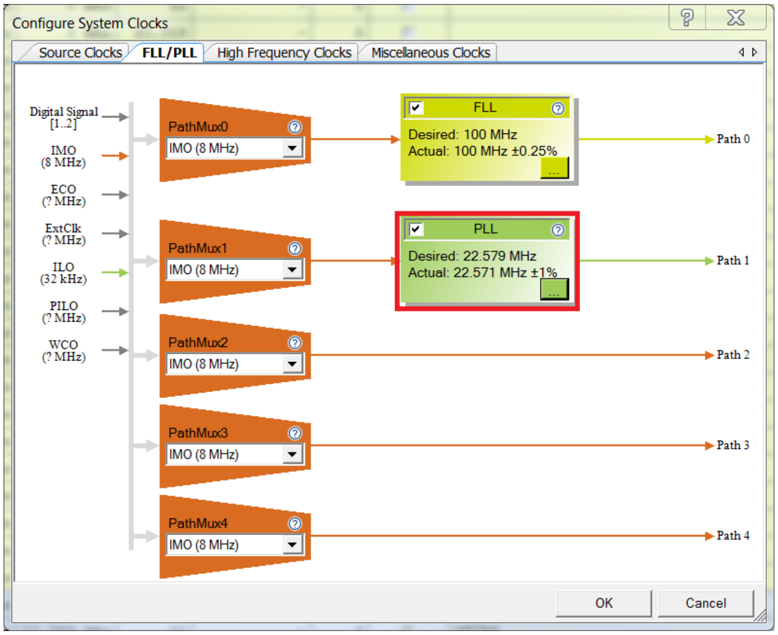
Design-Wide Resources

In the FLL/PLL tab, enable the PLL to be 22.5792 MHz. This clock frequency comes from the following equation:

$$HFClk1 = Frame\ Rate \times Clock\ divider\ \times 16 \times Channel\ Length = 44.1k\ \times 1 \times 16 \times 32 = 22579.2\ kHz$$

Figure 4 shows the configuration of the PLL. Note that the actual frequency is 22.571 MHz, instead of the desired frequency of 22.5792 MHz.

Figure 4. Clock Configuration Window



Configure System Clocks

FLL: ☑ Desired: 100 MHz Actual: 100 MHz ±0.25%

PLL: ☑ Desired: 22.579 MHz Actual: 22.571 MHz ±1%

PathMux0: IMO (8 MHz)

PathMux1: IMO (8 MHz)

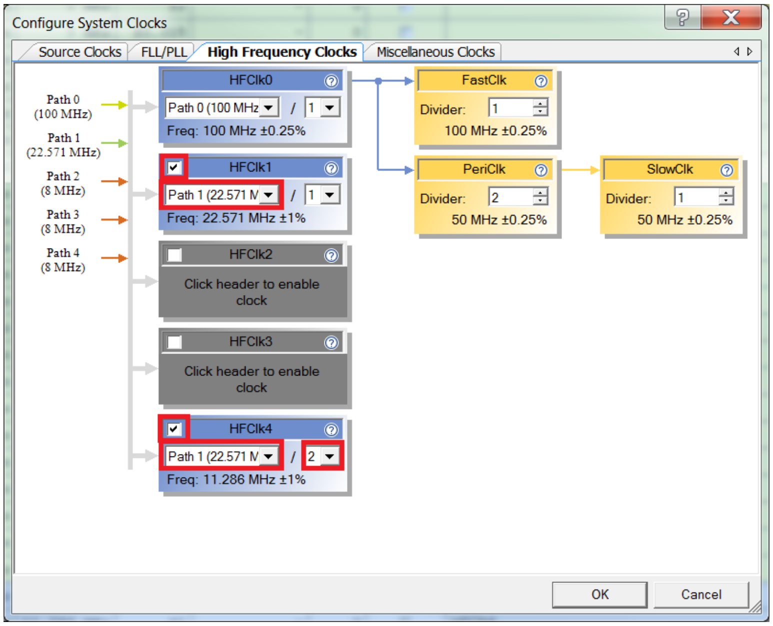
PathMux2: IMO (8 MHz)

PathMux3: IMO (8 MHz)

PathMux4: IMO (8 MHz)

Configure the High Frequency Clocks HFClk1 and HFClk4 to be linked to the Path 1. Note that ideally both the clocks created for the I^{2}S block (MCLK and BCLK) should come from the same source. Figure 4 shows how the high frequency clocks are configured.

Figure 5. High Frequency Clock Configuration



Configure System Clocks

Source Clocks | FLL/PLL | High Frequency Clocks | Miscellaneous Clocks

Path 0 (100 MHz)
Path 1 (22.571 MHz)
Path 2 (8 MHz)
Path 3 (8 MHz)
Path 4 (8 MHz)

☐ HFClk0: Path 0 (100 MHz / 1 — Freq: 100 MHz ±0.25%

☑ HFClk1: Path 1 (22.571 M / 1 — Freq: 22.571 MHz ±1%

☐ HFClk2: Click header to enable clock

☐ HFClk3: Click header to enable clock

☑ HFClk4: Path 1 (22.571 M / 2 — Freq: 11.286 MHz ±1%

FastClk — Divider: 1 — 100 MHz ±0.25%

PeriClk — Divider: 2 — 50 MHz ±0.25%

SlowClk — Divider: 1 — 50 MHz ±0.25%

Related Documents

Table 2 lists all relevant application notes, code examples, knowledge base articles, device datasheets, and Component / user module datasheets.

Table 2. Related Documents

Application Notes	
AN210781 Getting Started with PSoC 6 MCU with BLE Connectivity	Describes the PSoC 6 BLE, and how to build this code example
AN217666 PSoC 6 MCU Interrupts	Describes how to use interrupts in PSoC 6
PSoC Creator Component Datasheets	
Inter-IC Sound Bus (I^{2}S) Component	Sends digital audio streaming data to external I^{2}S devices
Intra-Integrated Circuit (I^{2}C) Component	Supports I^{2}C slave, master, and master-slave operation configurations.
Device Documentation	
PSoC 6 MCU: PSoC 63 with BLE Datasheet (PRELIMINARY)	
PSoC 6 MCU: PSoC 62 Datasheet	
PSoC 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual	
PSoC 6 MCU: PSoC 63 with BLE Register Technical Reference Manual	
PSoC 6 MCU Programming Specifications	
Development Kit (DVK) Documentation	
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit	

Document History

Document Title: CE218636 - PSoC 6 MCU Inter-IC Sound (I2S) Example

Document Number: 002-18636

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5697907	RLOS	06/21/2017	New Code Example
*A	5849304	RLOS	08/10/2017	Updated Project to PSoC Creator 4.2 Updated Clock Frequencies to use slower values
*B	5891795	RLOS	09/21/2017	Updated Audio Codec part Updated TopDesign screenshot

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.