Introduction:

Pioneer Kit PSoC 5LP USB Audio demonstrates the capability of PSoC 5LP to communicate with any USB Hosts, provide a high-quality recording interface to it and extract high-quality digital audio from the USB host. The USB host includes those which comply with USB audio class v1.0 (Windows, Mac, Linux, Android, Raspberry Pi, BeagleBone Black etc.)

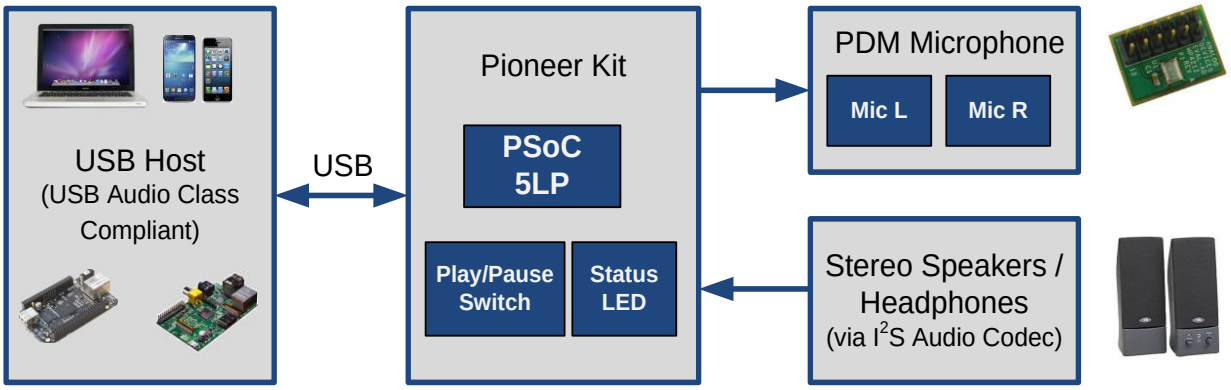


Figure 1: Pioneer Kit PSoC 5LP USB Audio

USB Audio capabilities of Pioneer Kit PSoC 5LP USB Audio project:

- USB-IF 2.0 Specification compliant
- USB Audio Class v1.0 compliant
- Two audio streams: 1 stereo microphone input (via PDM MEMS Microphone) and 1 stereo speaker output (via Codec)
- 24-bit and 16-bit resolution
- Both 44.1 kHz and 48 kHz clock tree support (upgradable to higher sample rate)
- Volume control feature unit support (optional depending on the codec)
- USB HID consumer control support (for Play/Pause action)

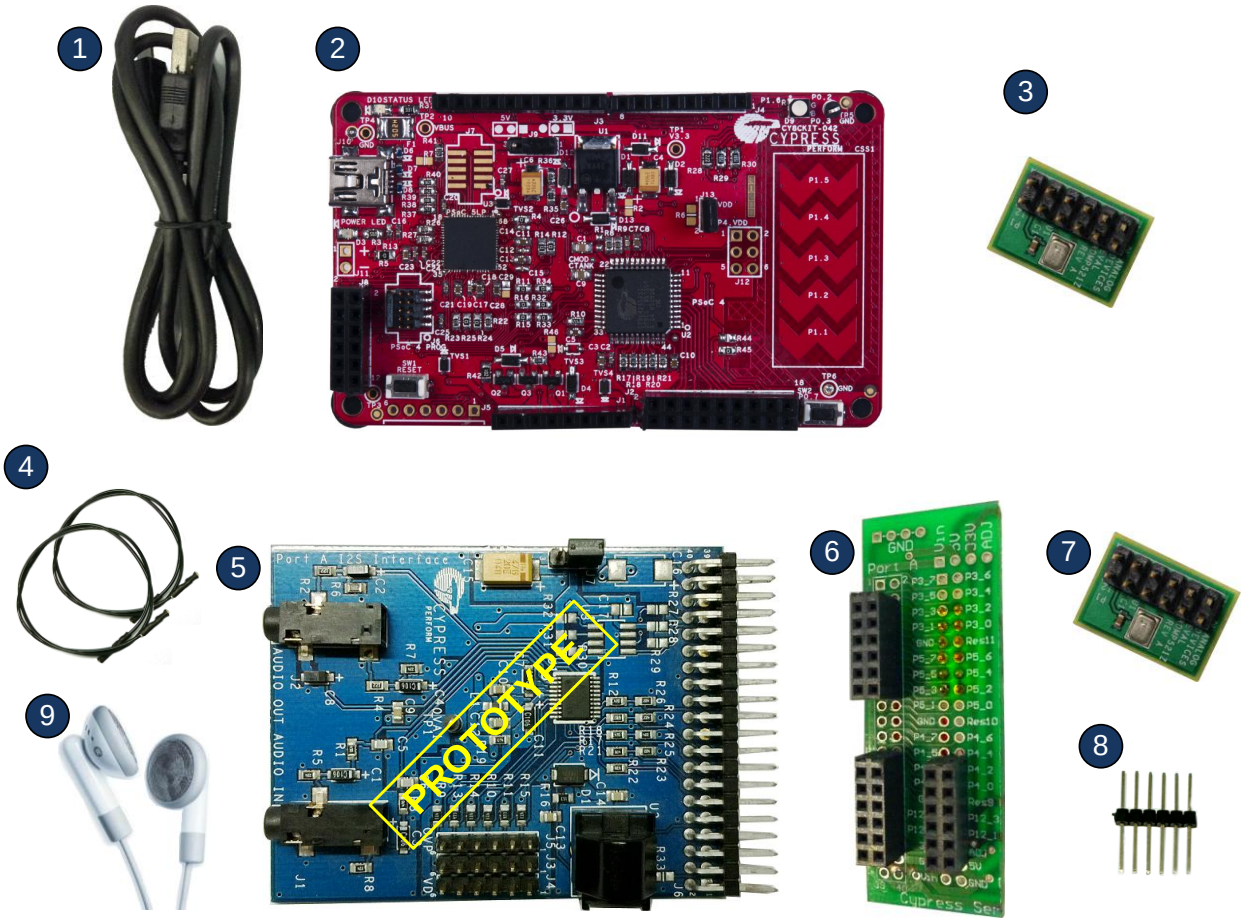
This user guide provides you detailed instructions on how to use the Pioneer kit as a USB audio device across various USB host platforms

Hardware Requirements

Table 1: Pioneer Kit PSoC 5LP USB - Hardware Requirements

S. No.	Content	Description
1	CY8CKIT-042 (Pioneer Kit)	Main controller board
2	ADMP521Z EVAL	Analog devices' PDM MEMS Microphone module
3	USB Cable	Type A to min B cable for connecting Pioneer kit to the USB Host / PC
4	Audio Codec board	Audio codec acts as I2S slave and converts digital audio from PSoC to analog audio for headphones or speakers

5	2x6 Header Duplicator board	For connecting two PDM Mics (stereo audio IN) and audio codec board simultaneously. This is optional and alternatively a breadboard can also be used
6	Jumper Wires	For connecting 12 (2x6) pin P5LP header of Pioneer kit to the audio codec board
7	Burg Strips	For connecting jumper wires to P5LP header
8	Headphones	For listening to the audio-out from codec board. (Speakers with in-built amp can also be used)



1. USB Cable Type A – mini B
2. Pioneer Kit
3. PDM Microphone Evaluation module (ADMP521Z EVAL)

Optional Requirements

4. Jumper wires
5. Audio Codec board
6. 2x6 pin Header Duplicator board
7. ADMP521Z EVAL PDM Module (additional)
8. Burg Strips
9. Headphone

Figure 2: Hardware Requirements

Software Requirements

- PSoC Creator 2.2 SP1
- Pioneer Kit PSoC 5LP USB Audio example project firmware and documentation
- Any audio recording application (Preferably **Audacity**)
- Any audio playback application (Preferably **Window Media Player** / **iTunes**)

System Requirements

For programming the Pioneer kit:

- PC With USB 2.0 Host
- Windows 8 / 7 / XP (32 or 64 bit)

For the USB Host:

- PC with USB 2.0 Host (at least one USB port)
- USB audio class v1.0 compliant

Getting Started:

Jumper Settings in the Pioneer Kit:

Configure the jumper settings in the kit hardware according to the following instructions:

- Jumper J9 should be placed in the 2-3 position as shown in **Figure 3.**
- Jumper J13 should be placed in the 1-2 position (Note: This jumper can be left as such / removed once programming the Pioneer kit is done)

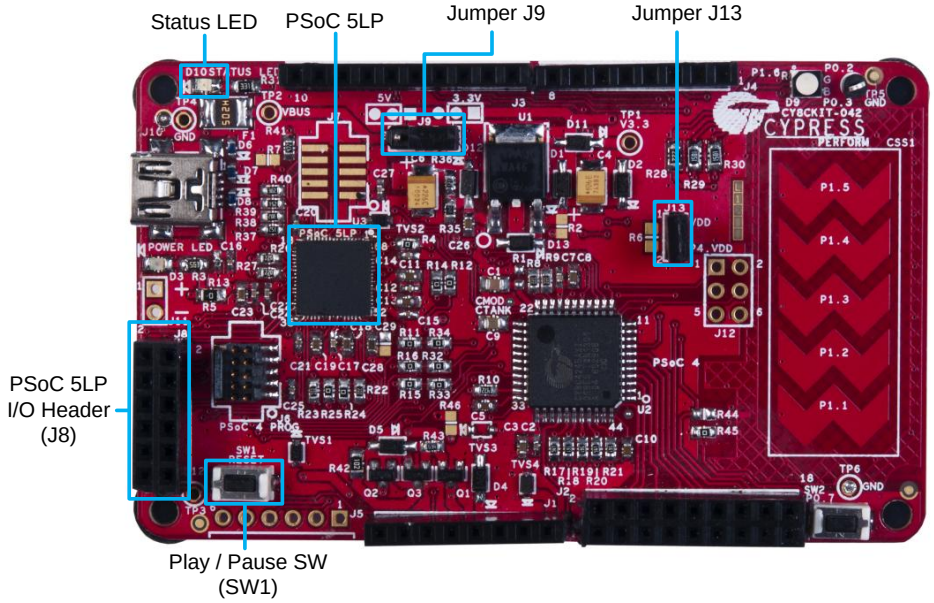


Figure 3: Pioneer Kit Hardware - Jumper settings

Building the Example Project:

- Unzip PioneerKit_P5LP_USB_Audio.zip to a location of user's choice.
- Open PSoC Creator 2.2 SP1. Select File -> Open -> Project / Workspace. See **Figure 4.**

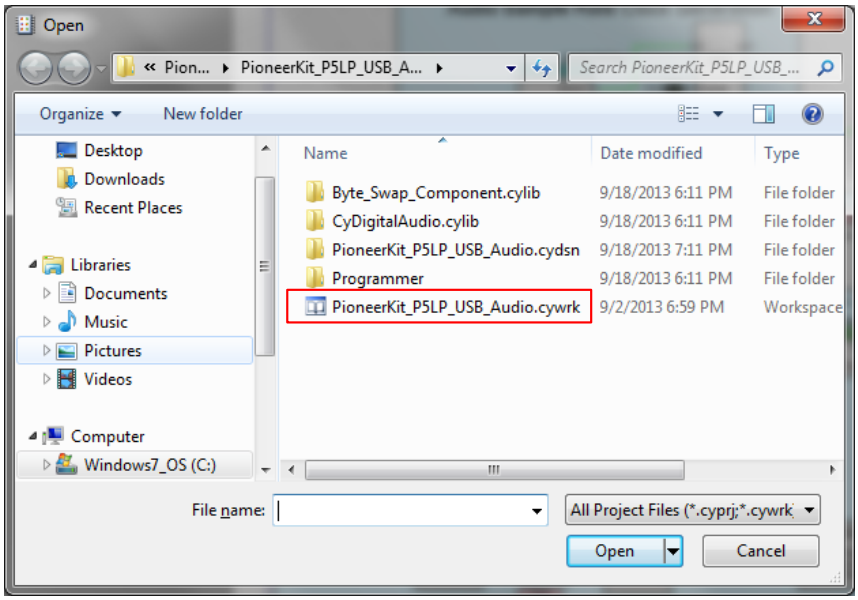


Figure 4: Open Project Dialog box - PSoC Creator

- Locate and open the PioneerKit_P5LP_USB_Audio.cywrk present in *<Install directory>\Firmware\PioneerKit_P5LP_USB_Audio*
- Build the project by selecting Build> Build PioneerKit_P5LP_USB_Audio

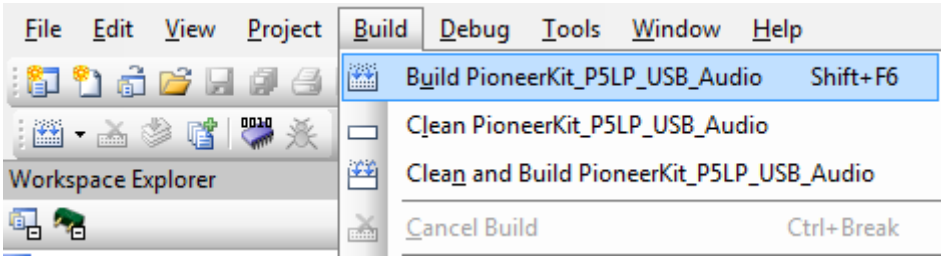


Figure 5: Building the Example Project

- The Build output will be shown in the output window as shown in the **Figure 6**.

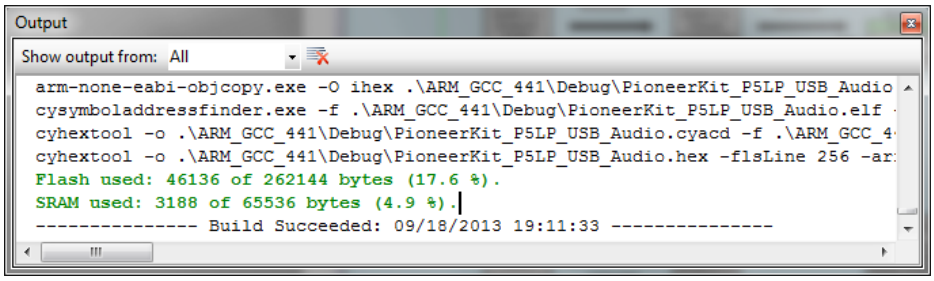


Figure 6: Output Window - PSoC Creator

Downloading the Example Project into the Pioneer Kit PSoC 5LP:

- Open the Bootloader Host Application from Tools > Bootloader Host in PSoC Creator as shown in the **Figure 7**.

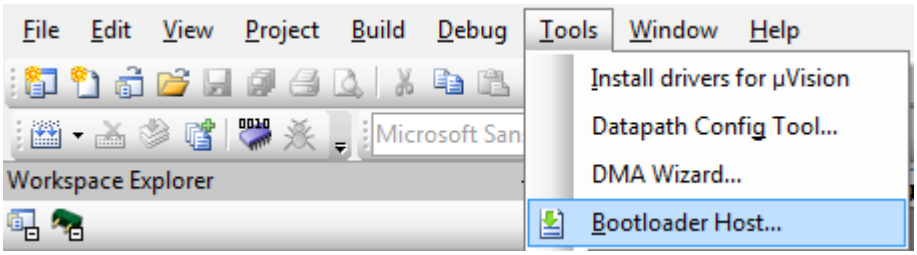


Figure 7: Opening Bootloader Host

- Point the File location to the *PioneerKit_P5LP_USB_Audio.cyacd* file as shown in **Figure 8**. The file is present in *<Install Directory>\ PioneerKit_P5LP_USB_Audio\ PioneerKit_P5LP_USB_Audio.cydsn\ ARM_GCC_441\ Debug\ PioneerKit_P5LP_USB_Audio.cyacd*

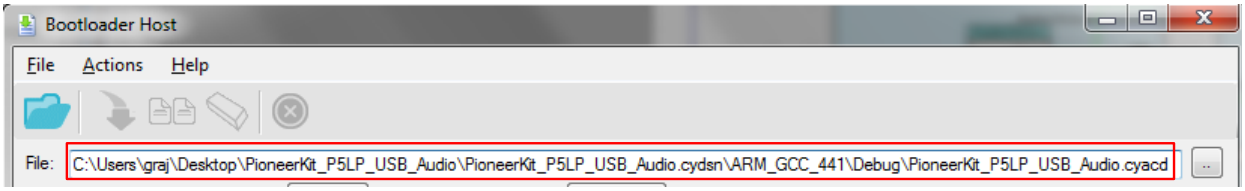


Figure 8: Target File location in Bootloader Host

- Enable Show USB devices in the Port Filters options windows as shown in the **Figure 9**.

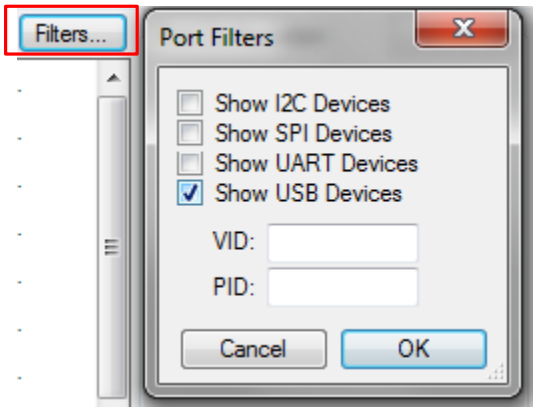


Figure 9: Port Filters Options Window - Bootloader Host

- Connect the USB Cable (Type A header) into any of the USB 2.0 ports in the PC. Now hold the switch SW1 in the Pioneer kit and plug in the USB cable as shown in **Figure 10**. The status LED D10 in the Pioneer kit should start blinking. It indicates that the PSoC 5LP has entered bootload mode.

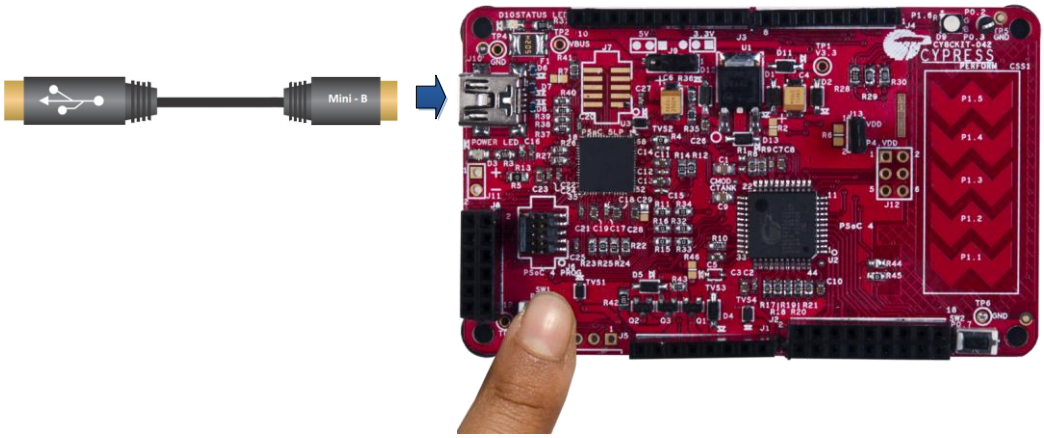


Figure 10: Holding SW1 to Enter Bootload Mode

- The kit will now be identified as a USB HID device with VID: 04B4 and PID: F13B in the Bootloader host application. Select the corresponding USB HID device and hit Program.

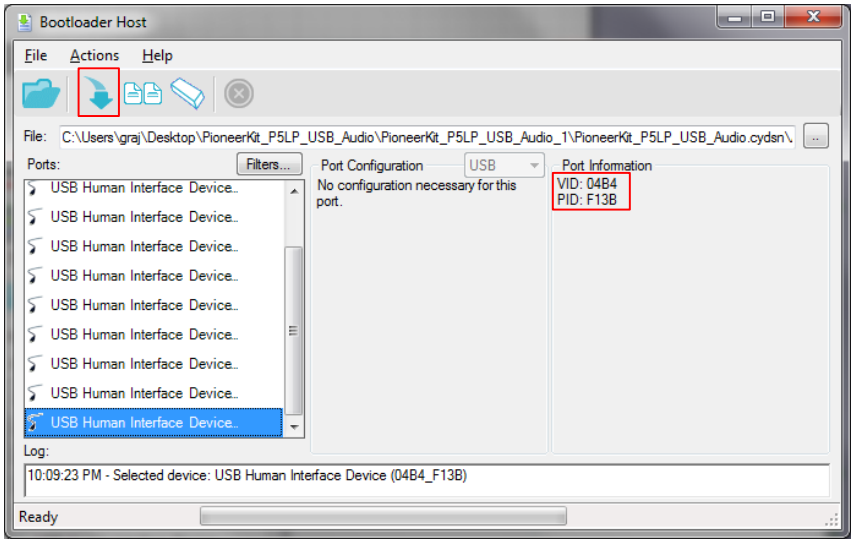


Figure 11: Bootloading the Pioneer Kit using Bootloader Host Application

USB Audio Demo:

Audio recording:

The instructions for audio recording with Windows 8 / 7 OS as USB Host are detailed as follow.

- Plug in the PDM Microphone module into the Pioneer Kit as shown in the **Figure 12**.

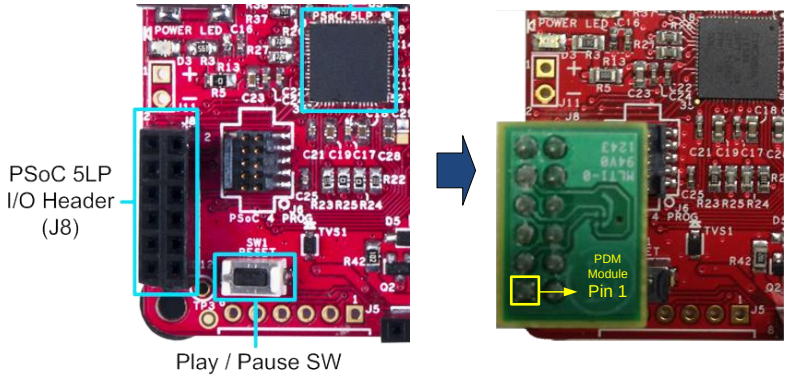


Figure 12: Plugging in PDM Module into P5LP Header

- Connect the Pioneer kit to the PC running Windows 8 or 7. The kit will be identified by the host as *Cypress Digital Audio DVK*
- Open *Recording* tab in *Sound* properties window. (Select *Sound* from *Control Panel* or right click on the *Sound* icon in the *Task bar* as shown in **Figure 13**)

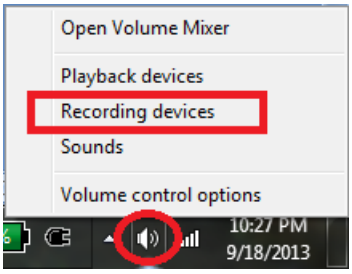


Figure 13: Opening Sound Properties window

- In the *Recording* tab, right click on "*Cypress Digital Audio DVK*" and set it as the default audio device as shown in the **Figure 14.**

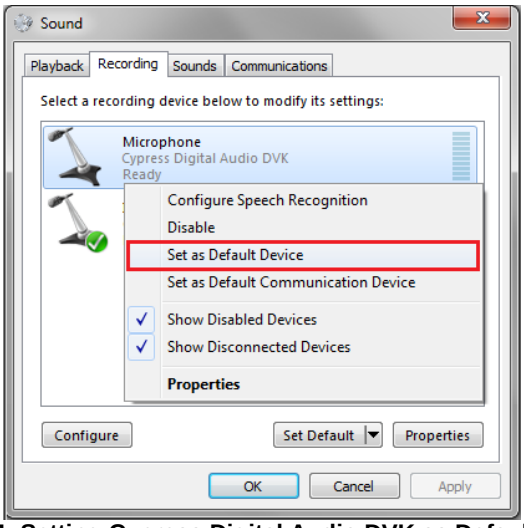


Figure 14: Setting Cypress Digital Audio DVK as Default Device

- Open Audacity. Ensure whether the recording device is chosen as Cypress Digital Audio DVK. See **Figure 15**
- Hit record. You can see your voice getting recorded in the PC via Pioneer Kit PSoC 5LP and PDM microphone.

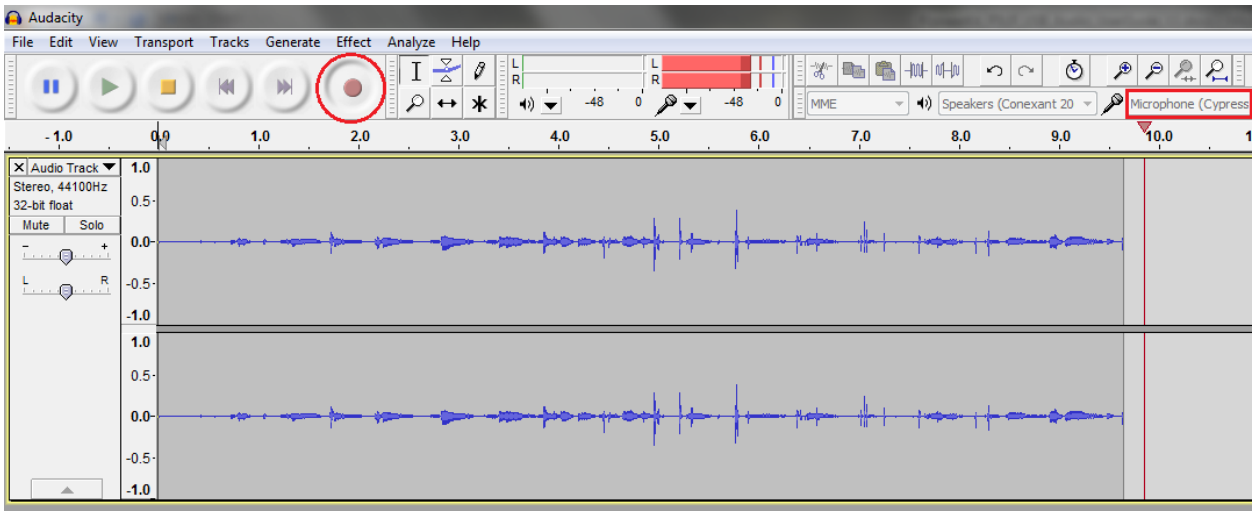


Figure 15: Audio Recording in Audacity

- The volume level of the recorded audio can be updated in *Levels* tab in *Properties* window of *Cypress Digital Audio DVK*. See **Figure 16**

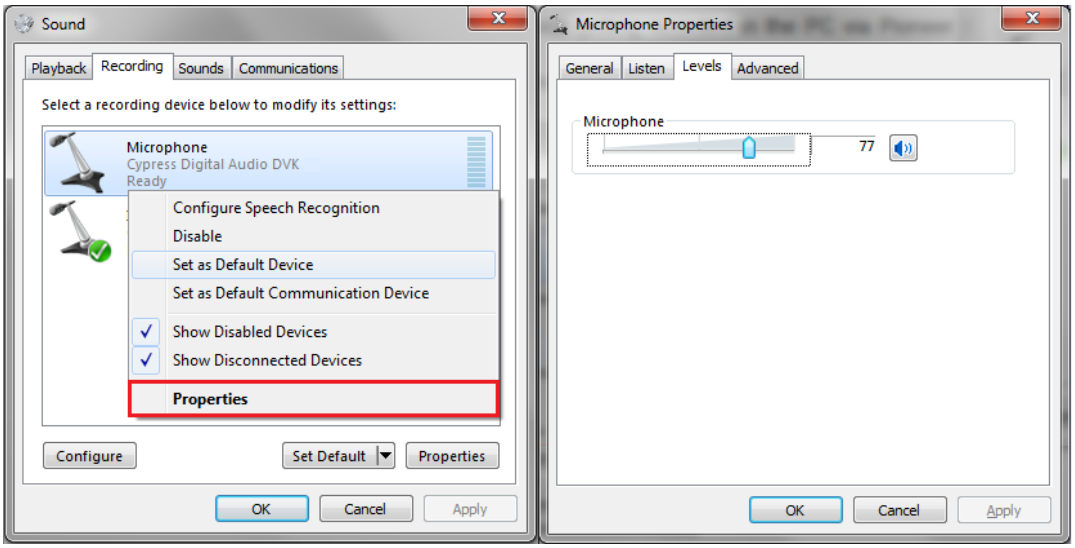


Figure 16: Recording Volume update in Windows 8 / 7

- Sampling rate[1] of the USB Audio device can be changed using the *Advanced* tab in *Microphone Properties* window shown in **Figure 17**.

[1] While changing sampling rate, care should be taken to ensure that the sampling rate of *Microphone* as well as *Speakers* of the *Cypress Digital Audio DVK* are same. Else, it might result in audio corruption.

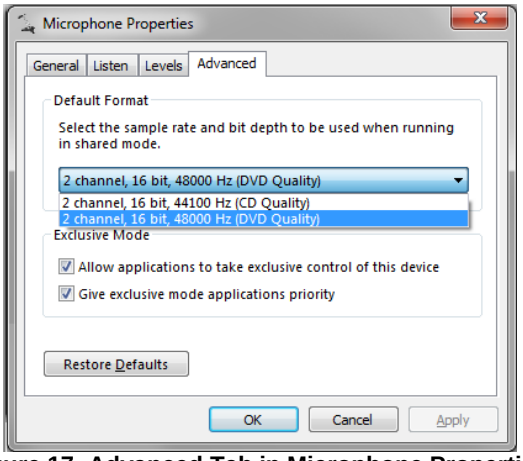


Figure 17: Advanced Tab in Microphone Properties

The similar procedure can be followed for any other USB host which supports audio recording via USB. (For e.g., in case of Mac OS X, open Audio MIDI Setup and set Cypress Digital Audio DVK as the default recording device.)

Audio Playback[2]:

For extracting audio-out from Pioneer kit, an audio codec board with following features is needed.

a) The codec should support I^2S Slave mode
b) The board should provide facility to route external MCLK to the codec. (i.e., for routing MCLK from PSoC 5LP to the codec)
c) The codec may / may not have control interface. If present, it should support I^2C communication
d) Easier pin-out options should be available to blue-wire the MCLK, SCLK, LRCLK, SDTO from the PSoC to the MCLKI, SCLK, LRCLK (or WS), SDTI of the codec respectively.

The I^2C configurations in this project are done for the Cirrus codec **CS42L51**. Refer **Codec Interface** section for more details on configuring the project for a different codec supporting I^2C.

For the purpose of demo, the audio codec prototype board made out of **AK4556** (no control interface needed for this codec) is used. Refer to **Audio Codec Board** in Hardware configurations section for details on the design schematic of the codec board

The following instructions provide step by step details about USB audio-playback from Window 8 / 7 via Pioneer Kit PSoC 5LP.

- Connect the I^2S pins of the audio codec board to the P5LP header in Pioneer kit using jumper wires as shown in the **Figure 18**.

[2] Audio Playback is provided as an optional feature in this project as there are no audio codec boards available along with this kit (or readily available for online purchase). However the functionality of the audio-path is fully tested with the codecs referred in this document.

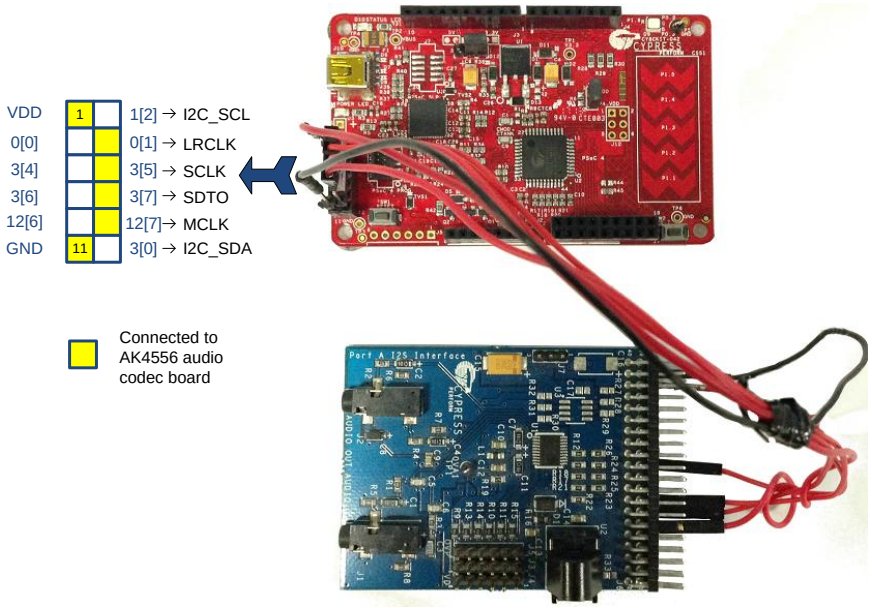


Figure 18: Pioneer Kit - Audio Codec Board Pin Connections

- Connect the Pioneer kit to the PC running Windows 8 or 7. The kit will be identified by the host as *Cypress Digital Audio DVK*
- Open *Playback* tab in *Sound* properties window. (Select *Sound* from *Control Panel* or right click on the *Sound* icon in the *Task bar* as shown in **Figure 19**)

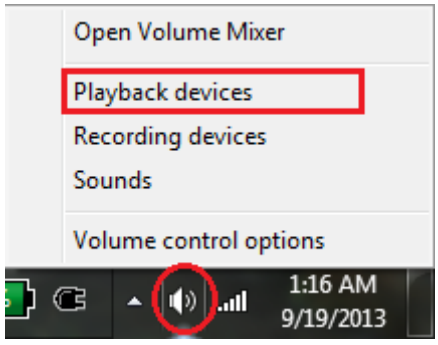


Figure 19: Opening Sound Properties window

- In the *Playback* tab, right click on "*Cypress Digital Audio DVK*" and set it as the default audio device as shown in the **Figure below.**

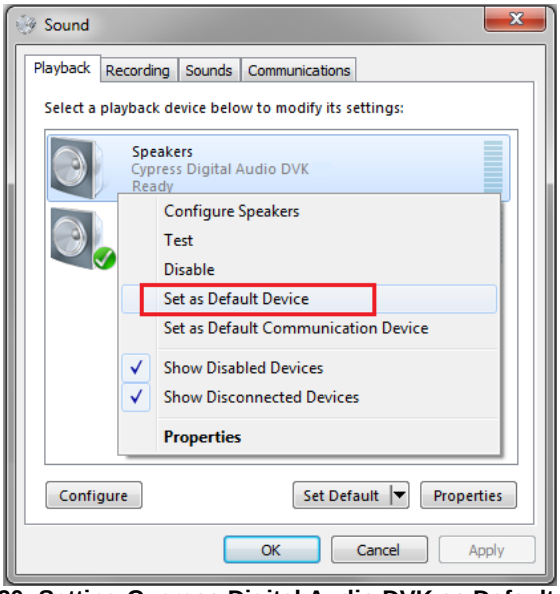


Figure 20: Setting Cypress Digital Audio DVK as Default Device

- Open *Windows Media player* / *iTunes* and play any song of your choice. (Alternatively, the *Test* option shown in **Figure 20** can also be used to quickly test the functionality of audio out via pioneer kit)
- Sampling rate[3] of the USB Audio device can be changed using the *Advanced* tab in *Speakers Properties* window.

Note: Volume update will not be functioning if the codec used doesn't have control interface to update the codec audio volume. If an audio codec with volume update feature is used, the same can be updated via Levels tab in Speakers Properties window.

The similar procedure can be followed for any other audio class v1.0 compliant USB host. (For e.g., in case of Mac OS X, open Audio MIDI Setup and set Cypress Digital Audio DVK as the default playback device.)

Simultaneous audio recording and Playback (optional):

The Pioneer Kit PSoC 5LP USB Audio project is capable of supporting stereo audio-in and stereo audio-out. For utilizing the full functionality of the project, all the modules need to be connected to the Single P5LP header in the Pioneer kit. Hence, a small custom board (or a bread board) needs to be used to duplicate the PSoC 5LP header x3 (i.e., to plug in 2 PDM mics for stereo-in and 1 codec board pin for stereo-out). A typical custom board setup will look like the one shown in the **Figure 21**.

[3] While changing sampling rate, care should be taken to ensure that the sampling rate of *Microphone* as well as *Speakers* of the *Cypress Digital Audio DVK* are same. Else, it might result in audio corruption.

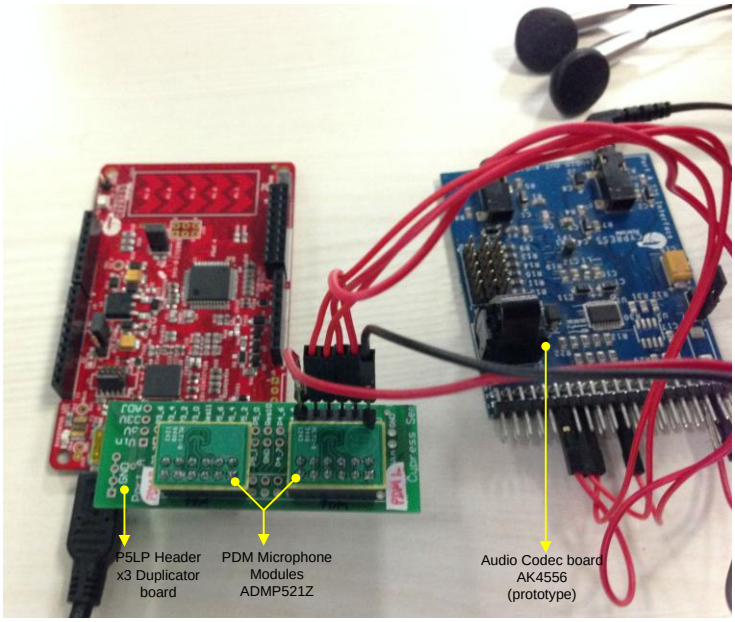


Figure 21: Pioneer Kit P5LP USB Audio - Complete HW Setup example

Note: In the P5LP header x3 duplicator board, the L/R select pin of the headers corresponding to PDM mics should be connected to either VDD or GND based on the choice of PDM mic (left /right). Refer to **P5LP Header x3 Duplicator Board** *section for more details.*

System Architecture:

The USB digital audio solution system supports Mac/PC/any USB Host acting as a USB Audio Class 1.0 master and PSoC 5LP acting as a USB Audio Class 1.0 slave device. The digital audio-out of the USB host is sent over USB to PSoC 5LP (or sent over USB from PSoC 5LP to the USB host for audio-in). PSoC 5LP generates an audio master clock for the desired audio sample rate that is synchronized to the USB host clock. With this synchronized audio master clock, PSoC 5LP sends audio samples to an external codec (or receives samples from the codec) to convert the digital audio to analog for playback. **Figure 23** illustrates the digital audio signal flow.

USB Audio Clock Recovery Process in PSoC

PSoC 5LP supports all the elements required for USB digital audio streaming. The heart of the solution is the patent-pending USB audio clock recovery process implemented using PSoC 5LP's digital blocks and PLL; **Figure 22** outlines the basic configuration.

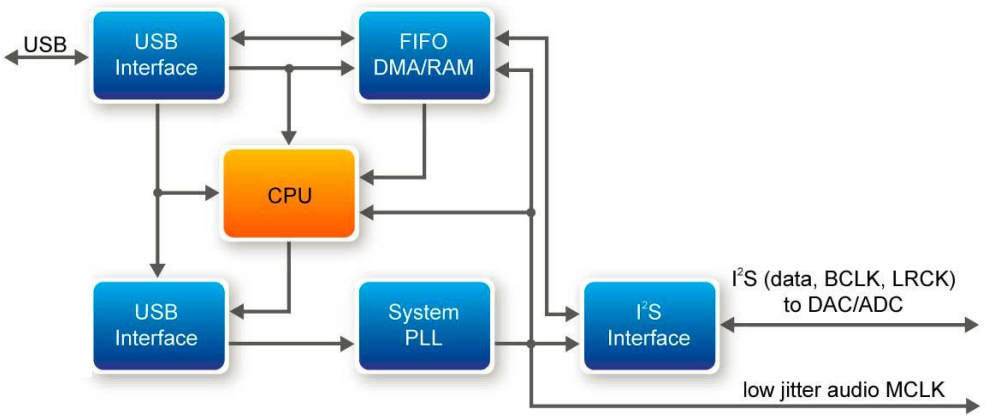


Figure 22: PSoC's USB Audio Clock Recovery Process

PSoC 5LP's programmable USB interface allows combining several audio and control protocol endpoint functions. The matrix of programmable digital logic blocks, known as Universal Digital Blocks (UDBs), implement a frequency synthesis system that derives any standard USB audio sample rate clock from a single stable crystal source. In normal synchronous mode, the clock is locked to the received USB time base using the USB start-of-frame token pulse timing. The system clock PLL is integrated into this synthesizer through the flexible clock routing framework. The whole system tracks the source sample rate and delivers a high-quality audio master clock for the system's audio converters, at a jitter level that matches the needs of modern quality audio systems Fixed-function microcontrollers cannot meet the tough performance requirements of this clock generation process. Their inflexible clock generation systems cannot be adjusted to be both correct and low in jitter; they usually fall back on an 'add/lose samples' approach. This can work in telephony, but is unacceptable for high-quality audio. Meanwhile, dedicated USB audio interface devices (another entry in the BOM in addition to the control microprocessor) cannot simultaneously manage the critical bidirectional control protocol traffic that delivers innovative new functionality in the latest media players. PSoC 5LP's clock recovery algorithm is encapsulated in a single component (AudioClkGen) within PSoC Creator in this project.

Audio – OUT[4] Interface:

The following PSoC 5LP system blocks are required to extract out high-quality USB digital audio:

- USBFS block to send/receive digital audio data to a USB host according to the USB Audio Class 1.0 Specification
- ByteSwap to handle the endian difference between the USB and I²S interface, and handle 16-bit and 24-bit digital audio data dynamically
- I²S master to interface digital audio-out data to an external codec
- DMA to move audio data from USB to the I²S domain
- Audio clock generator to generate high-quality audio clock for I²S out

[4] - The directions audio-OUT and audio-IN are always with respect to the USB Host. Hence, audio OUT always mean the data from the USB Host to the codec through I²S and audio IN mean the PDM data from PDM Microphone to the USB Host.

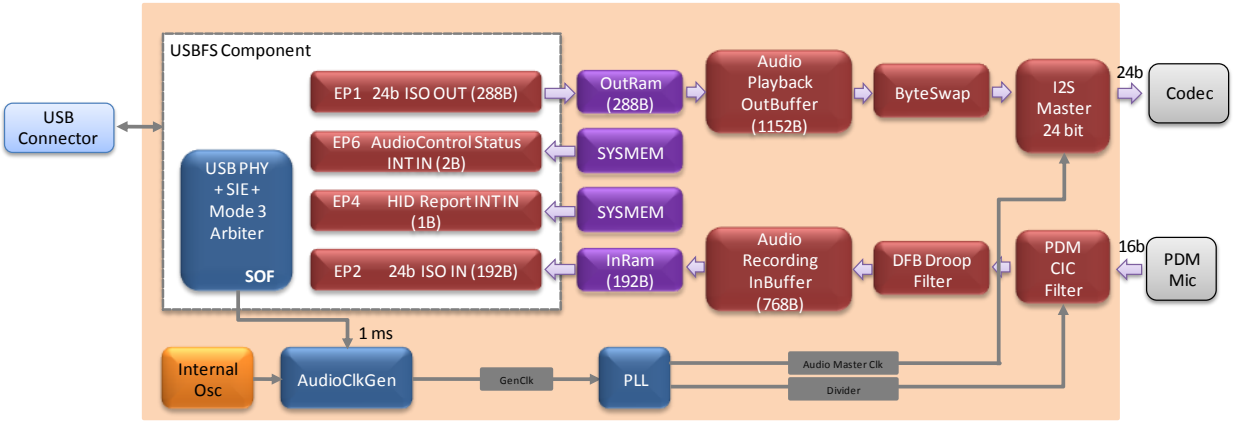


Figure 23: Pioneer Kit P5LP USB Audio - System Architecture

The digital audio-out data of the host is sent over USB (component: USBFS) to one of the USB audio isochronous endpoints of PSoC 5LP. For isochronous audio, one millisecond of audio data is received in each USB frame. The received audio samples are transferred to PSoC 5LP's system memory SRAM (OutRam) from the audio-out endpoint by the DMA channel embedded within the USB component. The audio data from OutRam is transferred to the PSoC 5LP system audio circular buffer (Audio Playback OutBuffer) in the hardware using a DMA channel (component: USBOutDMA). This moves the full block of audio samples from the USB frame to the SRAM buffer in one DMA transfer. When there are more than two USB audio frame samples in the audio circular buffer, another DMA channel (component: TxDMA) transfers the audio data to a dedicated real-time ByteSwap component to convert the 24-bit audio samples from the USB interface (LSB first) to an I^2S compatible format (MSB first). The byte swapped 24-bit audio samples from the ByteSwap component are moved to the PSoC 5LP I^2S component's internal hardware FIFO by another hardware DMA (component: I2S_TX_DMA), one byte at a time. When I^2S transmit is enabled (component: I2S), the I^2S component transfers audio samples to the external DAC of the codec. For 16-bit audio, the ByteSwap component inserts an extra byte to make it 24-bit, so the same I2S interface (24-bit) works for both 16-bit and 24-bit streams. **Figure 24** shows the schematic block diagram of the audio OUT path.

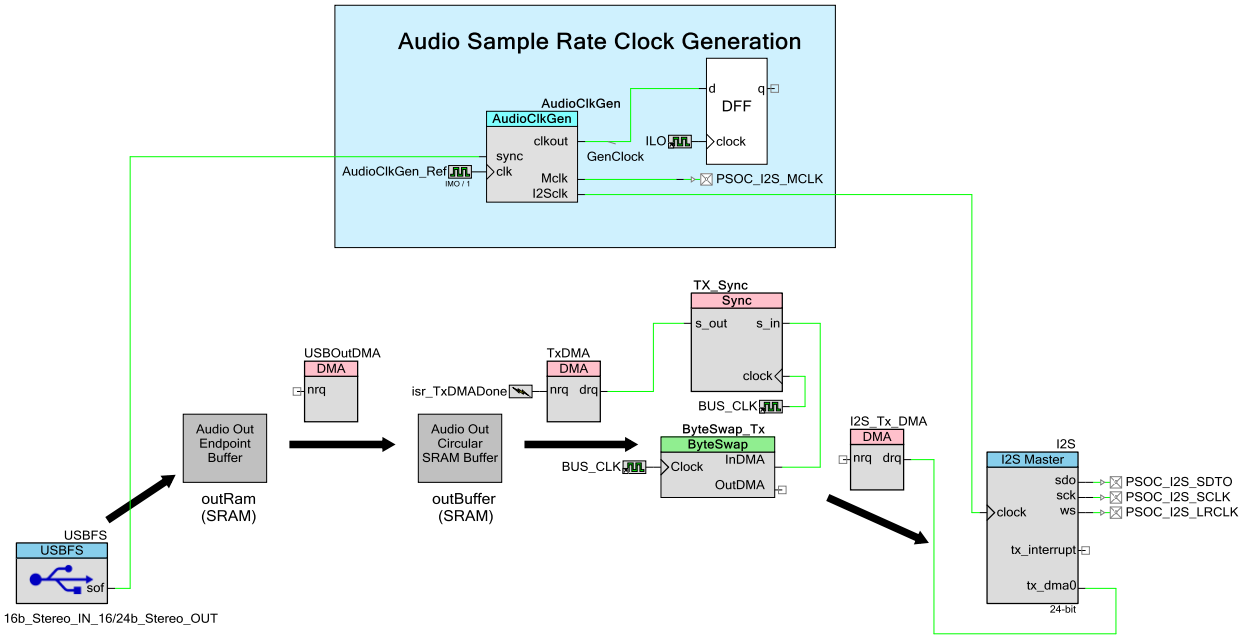


Figure 24: Audio OUT path Schematic

Audio-IN interface:

In case of audio recording, the PDM_CIC component replaces the I^2S as in audio playback. PDM (Pulse Density Modulation) is a serial 1-bit data obtained from the PDM MEMS microphone. This has to be converted to PCM (Pulse Code Modulation) to be sent over through the USB. The PDM decoding interface consists of a 5th order CIC filter implemented in PSoC 5LP's UDBs and an additional droop compensation filter implemented in PSoC 5LP's DFB which compensates the pass band droop introduced by the CIC stage. In case of stereo PDM support, the same GPIO pin (PSOC_PDM_DataIn) is used for obtaining the PDM data from both left and right PDM mics. The left channel PDM data is sampled during the rising edge of the PSOC_PDM_ClkOut and right PDM data during the falling edge. The PDM clock is provided by the PLL based on the sampling rate request obtained from the host. (For e.g., for a decimation rate of 64 and 48 kHz sampling rate, the PDM clock should be 48000*64 = 3.072 MHz). The Pioneer Kit PSoC 5LP USB Audio project, by default, is configured for a decimation rate of 64 (decimation rate of x32 is done in PDM_CIC and x2 in the DFB component) **Figure 25** shows the schematic block diagram of the PDM recording path carried out in PSoC 5LP in this project.

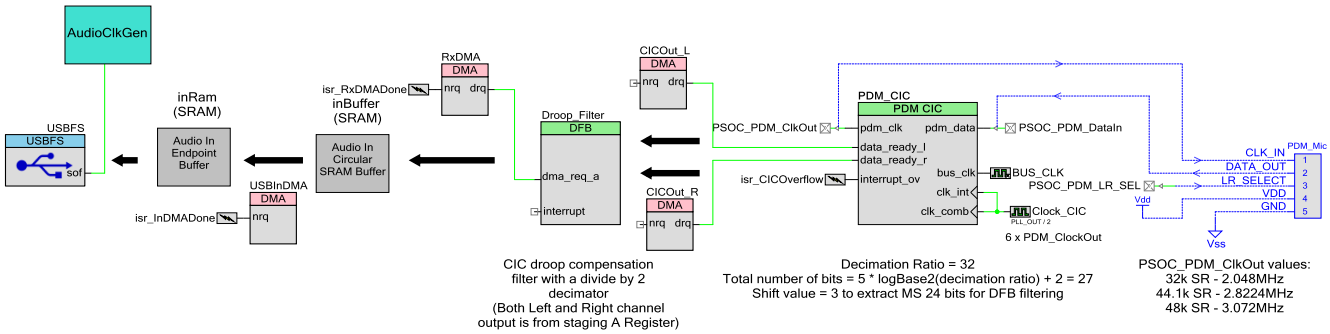


Figure 25: Audio IN Path Schematic

User Interface:

1. STATUS LED:
   The LED D10 in the Pioneer kit is configured as the audio status LED. It turns ON when either of the audio streams (IN or OUT) is active. Also, if the host enters sleep, the LED blinks twice indicating that the Pioneer kit is entering suspend mode.

2. PLAY/PAUSE SW:
   The push button SW1 is configured as the Play/Pause switch for any media application which is active.

Example Project configurations:

USBFS Component

USBFS component contains all the USB descriptor details needed during the enumeration process. The example project provides the following four different configurations as per the design requirement

1. USBFS_Default: This is the default component available out-of-box in the project. This configuration has descriptors for 16 bit stereo IN and 16/24 bit stereo OUT. The kit will be recognized by the USB host as a USB Audio Playback device and a Microphone.
2. USBFS_Default_with_IN_FeatureUnit: The Feature unit support is needed if we need volume control support. Adding a feature unit in the audio IN path will make the host not to use its internal microphone gain when the volume is updated. Rather, the volume change request will be passed via USB to the PSoC 5LP to handle the request[5]. In case, if feature unit is not present, most of the USB hosts take care of the volume control update with their internal amplifier techniques.
3. USBFS_IN_ONLY: This component has the descriptors for audio IN path only. The Pioneer Kit will be recognized as Digital Microphone device only.
4. USBFS_OUT_ONLY: As the name says, the component has descriptors for audio OUT path only.

Step by step procedure needed for implementing any of these configurations is explained in detail in the “Misc” page of the TopDesign schematic. **Figure 26** shows the TopDesign Schematic Misc page.

[5] This version of the project doesn’t have the volume update code for the IN path. The user can easily handle the IN volume change request by writing their custom code in VolumeControl.c by updating the shift count of the PDM_CIC component whenever there is a request.

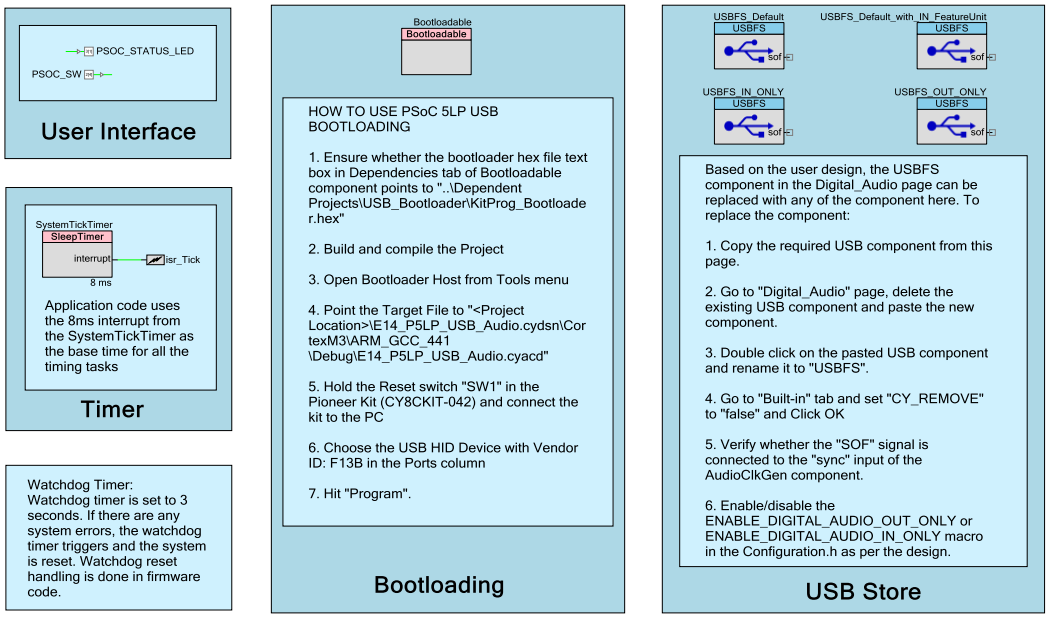


Figure 26: Misc Page - TopDesign Schematic

Configuration.h file:

The firmware is developed in a user-friendly way to enable/disable features by just commenting/un-commenting the pre-processor directives inside Configuration.h file. The macros are

CODEC_I2C_SUPPORTED: Enable/Disable the firmware code related to I2C configuration of the codec. This should be enabled / disabled based on the type of codec used in the design.

ENABLE_VOLUME_CONTROL: Volume control related code is enabled.

ENABLE_WATCHDOG: Enable watchdog timer for resetting the firmware if the code is stuck in some loop for more than the pre-set time.

MCLK_ALWAYS_ENABLED: Firmware is designed in way that all the I2S clocks are shutdown when audio stream is not active. But some of the codecs will require Master clock to be always enabled. This macro should be enabled in those cases.

HANDLE_USB_SUSPEND: The example project supports USB suspend mode. If the host enters sleep and there is no activity in the bus for certain time, PSoC and all its functional blocks are put in sleep. It is woken up once the USB host wakes up. This firmware code can be enabled / disabled by this macro.

ENABLE_DIGITAL_AUDIO_IN_ONLY: If this macro is enabled, only the components & code related to audio-IN path are enabled. (USBFS_IN_ONLY component should be used while enabling this macro)

ENABLE_DIGITAL_AUDIO_OUT_ONLY: If this macro is enabled, only the components & code related to audio-OUT path are enabled. (USBFS_OUT_ONLY component should be used while enabling this macro)

Codec Interface:

By default the example project is configured to work with **CS42L51** codec. However any codec that can support I²C for configuration and *I²S slave mode*[6] can be used to work with this. If a different codec is used, the register configurations values and addresses have to be replaced with that of the codec being used. These configurations can be found in Codec.h and Codec.c files of the codec.

If the codec used doesn't have I2C configuration support (i.e., No Volume control and other features support), the codec configuration is not required. Comment out the macro CODEC_I2C_SUPPORTED in this case. Else the firmware will hang waiting for the codec to respond via I²C.

Hardware Configuration:

The PSoC 5LP GPIO pins in the Pioneer kit are available in the 12 pin header J8. The pin configuration of the PSoC 5LP header is shown in the **Figure 27.**

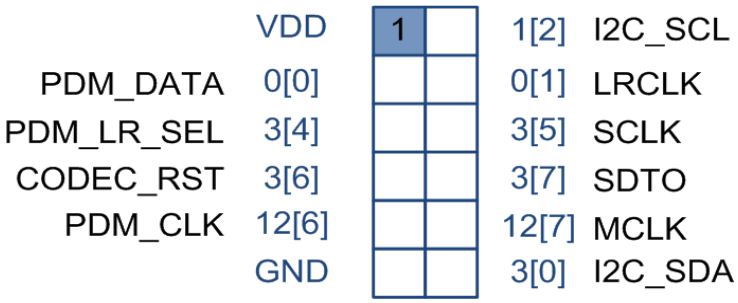


Figure 27: Pioneer Kit P5LP Header - Pin Configuration

PDM Microphone Module:

The example project is configured so as to work out-of-box (without any blue-wiring) as a microphone by directly plugging in the **ADMP521Z-EVAL** PDM microphone module to the P5LP header as shown in the **Figure 28**. However any PDM module can be used to operate with this project. See Introduction page in TopDesign schematic for more details.

Figure 28: Pioneer Kit P5LP Header with ADI PDM Module

Audio Codec Board:

[6] I²S Slave mode refers to all the I2S Clocks (Master clock, Bit clock, Word Select or LR clock) supplied to the codec externally (from PSoC 5LP).

The Cirrus Logic codec CS42L51 is used in this project. The schematic of the codec board used is shown in **Figure 29**. Alternately, the codec evaluation board[7] **CDB42L51** can be used.

Cirrus Logic CODEC

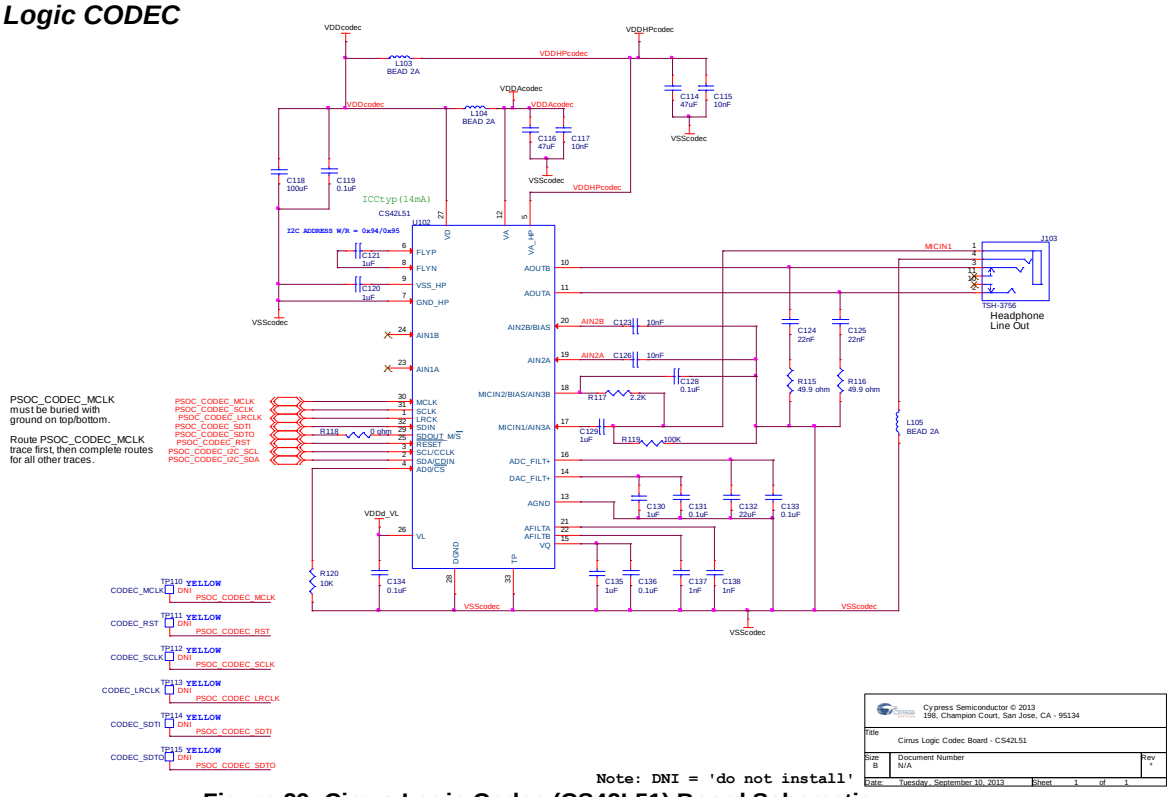


Figure 29: Cirrus Logic Codec (CS42L51) Board Schematic

If the user wishes to use a simple codec without any I^2C configuration option (i.e. no volume control support), any codec which can act as I^2S Slave can be used to extract audio out of the Pioneer kit. In this case, it is preferred to use **AK4556** codec. **Figure 30** shows the schematic of a custom board developed with AK4556 codec.

[7] It is preferred to develop the codec board instead of purchasing evaluation boards as they are costlier.

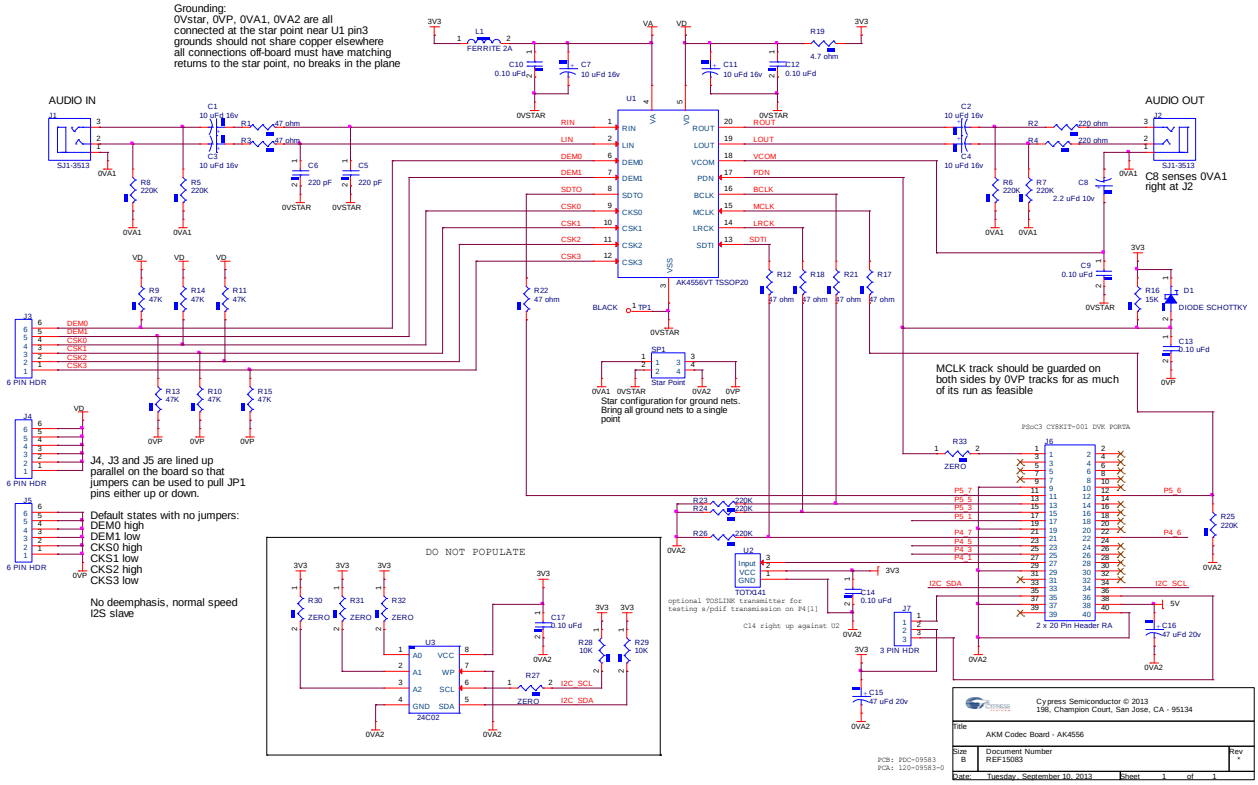


Figure 30: AKM Codec (AK4556) Board Schematic

P5LP Header x3 Duplicator Board:

This board helps in connecting the codec board and 2 PDM mic modules to the P5LP header in Pioneer kit. **Figure 31** shows the duplicator board used in this project.

In the P5LP header x3 duplicator board, the L/R select pin of the headers corresponding to PDM mics need not be connected to any GPIO of PSoC 5LP. It can be connected to either VDD or GND based on the choice of PDM mic (left /right). Hence, in the duplicator board, the pin corresponding to L/R select in one of the header is connected to VDD (for installing PDM Mic left) and the same pin in another header is connected to Ground (for installing PDM Mic right).

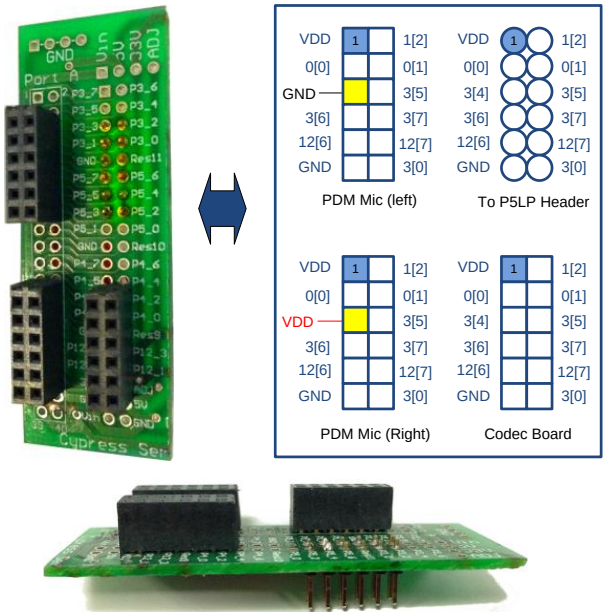


Figure 31: P5LP Header x3 Duplicator Board

Reverting the Pioneer Kit PSoC 5LP to Factory Default Firmware:

- Open the Bootloader Host application
- Point the target file location to *<Project Install Directory>\ PioneerKit_P5LP_USB_Audio\ Programmer\ KitProg\ KitProg.cyacd*
- Follow rest of the procedure same as detailed in **Downloading the Example Project** section.
- Now the PSoC 5LP will be restored back to factory default firmware and can act as a programmer for PSoC 4 in Pioneer kit.

Frequently Asked Questions (FAQs):

1. If the single PDM module is plugged in directly to the Pioneer kit header, will it work as Mono microphone?
   Yes. The Pioneer kit will be working as a mono microphone and the single PDM mic's data is fed to both left and right channels.
   Note: The Pioneer kit, by default, will be recognized as a stereo IN device. Change the number of channels in USB descriptors to 1 if it needs to be recognized as a Mono IN device.

2. Is the PSoC 4 in the Pioneer kit used in this project?
   No. The PSoC 4 is not used in the current version of the Example project. However, user can write their own code to map the CapSense Slider (connected to PSoC 4 in Pioneer kit) for volume control via I^2C interface between PSoC 5LP and PSoC 4.

3. The jumper J13 is supplying power to PSoC 4 VDD. Can it be removed to save power?
   Yes. J13 can be removed except while programming/bootloading PSoC 5LP. While powering the kit (via USB mini B), the PSoC 5LP enters bootload mode if J13 is not connected.
4. How to program PSoC 5LP in the Pioneer Kit?
   Programming PSoC 5LP in Pioneer kit can be done in two ways[8]. a) Hold the switch SW1 and plug in the USB (The kit enters Bootload mode). Open Bootloader Host and bootload the corresponding .cyacd file. See "Misc" page of TopDesign schematic for more details. b) Install the 10-pin programming header J7 in Pioneer kit and program it via **MiniProg3**.

5. What are the requirements of a codec to work with Pioneer Kit PSoC 5LP USB Audio example project?
   The codec should be capable of supporting I2S Slave mode (MCLK, BCLK, LRCLK provided externally). If volume control support is required, the codec should support control interface via I2C.

6. What changes are needed to make the example project work with a different codec?
   If the codec doesn't have any control interface (no volume control support), just commenting out the macro #define CODEC_I2C_SUPPORTED will be sufficient. Alternately, if the I2C control interface is available, the I2C register write addresses and configurations in Codec.h file should be modified as per the codec datasheet.

7. Does the project support audio- IN via Codec I^2S?
   No. The Pioneer Kit PSoC 5LP USB Audio project (with stereo PDM and stereo I^2S audio out) is a complex project and utilizes all the **datapaths** available in PSoC 5LP. Hence, there are no enough resource left to support audio-IN via Codec I^2S additionally. (However, it can be done by modifying the project to Mono PDM mic and stereo I^2S out)

[8] Debugging can be done via MiniProg3 (option b) only.