AN3135

USB-to-I2S™ Bridging with Microchip Hubs

Author: *Jeffrey Hunt*
Microchip Technology, Inc.

1.0 INTRODUCTION

The Inter-IC-Sound (I^2S™) bus is the standard interface for connecting audio devices, such as audio codecs.

The USB Audio/Video (AV) Device Class Definition describes, among other things, the methods used to communicate with devices containing sound-related functionality. This includes digital and analog audio data with its associated meta-data and the functionality used to control the audio environment, such as volume controls.

Microchip USB hubs provide a bridge from USB to an audio device codec via I^2S. These devices support only USB Audio Device Class Specification v1.0. The I^2S interface is a five-wire connection to the ADAU1961 codec. There is sufficient configuration flexibility to change the settings of the Analog Devices ADAU1961 or to implement a different codec.

While the audio data to and from the audio codec is through the I^2S interface, USB Audio Device control data is transferred through the I^2C (controller) interface of the hub.

For more details regarding the implementation, refer to the device data sheets listed in the References section.

2.0 SECTION

This document includes the following topics:

3.0 REFERENCES

Consult the following documents for details on the specific parts referred to in this document:

- *Microchip USB4715 Data Sheet*
- *Microchip USB4914 Data Sheet*
- *Microchip USB4916 Data Sheet*
- *Microchip USB4925 Data Sheet*
- *Microchip USB4927 Data Sheet*
- *Microchip USB4712 Data Sheet*
- *Microchip USB4912 Data Sheet*
- *Microchip USB7002 Data Sheet*
- *Microchip USB7050 Data Sheet*
- *Microchip USB7051 Data Sheet*
- *Microchip USB7152 Data Sheet*
- *Microchip USB7056 Data Sheet*
- *Microchip USB7202 Data Sheet*

- *Microchip USB7250 Data Sheet*
- *Microchip USB7251 Data Sheet*
- *Microchip USB7252 Data Sheet*
- *Microchip USB7256 Data Sheet*
- *Microchip USB7206 Data Sheet*
- *Microchip USB7216 Data Sheet*
- *UM10204, I2C-Bus Specification and User Manual*
- *Audio Device Class Spec v1.0*
- *ADAU1961 Data Sheet*

4.0 GENERAL INFORMATION

Microchip hub USB-to-I^{2}S Bridging features work via Host commands sent to an embedded Hub Feature Controller within the device located on an additional internal USB port. In order for the bridging features to work correctly, this internal Hub Feature Controller must be enabled by default. Table 1 provides details on default Hub Feature Controllers settings per device that contains the I^{2}S Bridge implementation described in this document.

TABLE 1: DEFAULT SETTINGS FOR HUB FEATURE CONTROLLER ENABLE

Part Number	Part Summary	Hub Feature Controller Default Setting
USB4715	5-Port USB2.0 FlexConnect Hub	Enabled by default on Port 5
USB4914	3-Port USB2.0 Multi-Host Reflector Hub	Enabled by default on Port 5
USB4916	5-Port USB2.0 Multi-Host Reflector Hub	Enabled by default on Port 7
USB4925	3-Port USB2.0 Dual Upstream Hub	Enabled by default on Port 4
USB4927	5-Port USB2.0 Dual Upstream Hub	Enabled by default on Port 6
USB7002	4-Port USB3.1 Gen1 Hub	Enabled by default on Port 6
USB7050	4-Port USB3.1 Gen1 Hub with USB power delivery on 3 ports	Enabled by default on Port 6
USB7051	4-Port USB3.1 Gen1 Hub with USB power delivery on 2 ports	Enabled by default on Port 6
USB7052	4-Port USB3.1 Gen1 Hub with USB power delivery on 1 port	Enabled by default on Port 6
USB7056	6-Port USB3.1 Gen1 Hub with USB power delivery on 1 port	Enabled by default on Port 8
USB7202	4-Port USB3.1 Gen2 Hub	Enabled by default on Port 6
USB7250	4-Port USB3.1 Gen2 Hub with USB power delivery on 3 ports	Enabled by default on Port 6
USB7251	4-Port USB3.1 Gen2 Hub with USB power delivery on 2 ports	Enabled by default on Port 6
USB7252	4-Port USB3.1 Gen2 Hub with USB power delivery on 1 port	Enabled by default on Port 6
USB7256	6-Port USB3.1 Gen2 Hub with USB power delivery on 1 port	Enabled by default on Port 8
USB7206	6-Port USB3.1 Gen2 Hub	Enabled by default on Port 8
USB7216	6-Port USB3.1 Gen2 Hub	Enabled by default on Port 8

4.1 Supported I^{2}S Bridge Features

In basic operation, I^{2}S Bridging interface enumerates as a USB device, which complies with *Audio Device Class Spec v1.0*.

The default audio features supported by the device are:

- 48 kHz
- 6 bits/sample
- Stereo audio
- ADAU1961 audio codec

The I^{2}S Bridge can also be configured by OTP for different audio features and settings, or even a different codec.

4.2 Configurable Features of I²S Bridge

TABLE 2: CONFIGURABLE FEATURES OF I²S™ BRIDGE

Parameter	Supported Values
Sampling Frequency (fs)	8 kHz, 11.025 kHz, 12 kHz, 16 kHz, 22.05 kHz, 24 kHz, 32 kHz, 44.1 kHz, 48 kHz
MCLK Frequency (Multiple of Sampling frequency)	From 1*fs to 1024*fs MCLK can be any arbitrary multiple of fs up to 1024 times fs. However, because the I²S LRCLK signal is derived from the MCLK source, it is recommended that only even-integer-multiples of fs be used.
Audio Sample Size	16-bits/sample, 24-bits/sample, 32-bits/sample
I²S™ Audio Interface Format	I²S mode, Left Justified mode, Right Justified mode
I²C Controller Control Interface Frequency	100 kHz and 400 kHz
Audio channels	Mono mode and Stereo mode
Enabling and disabling the I²S Bridge interfaces	Audio Out only (Speaker interface) Audio IN only (Mic interface) Audio IN and Audio Out (Line interface) Jack Detection interface
Jack Detection	Audio jack insertion-detection can be enabled or disabled

4.3 Implementation of an Alternative Codec

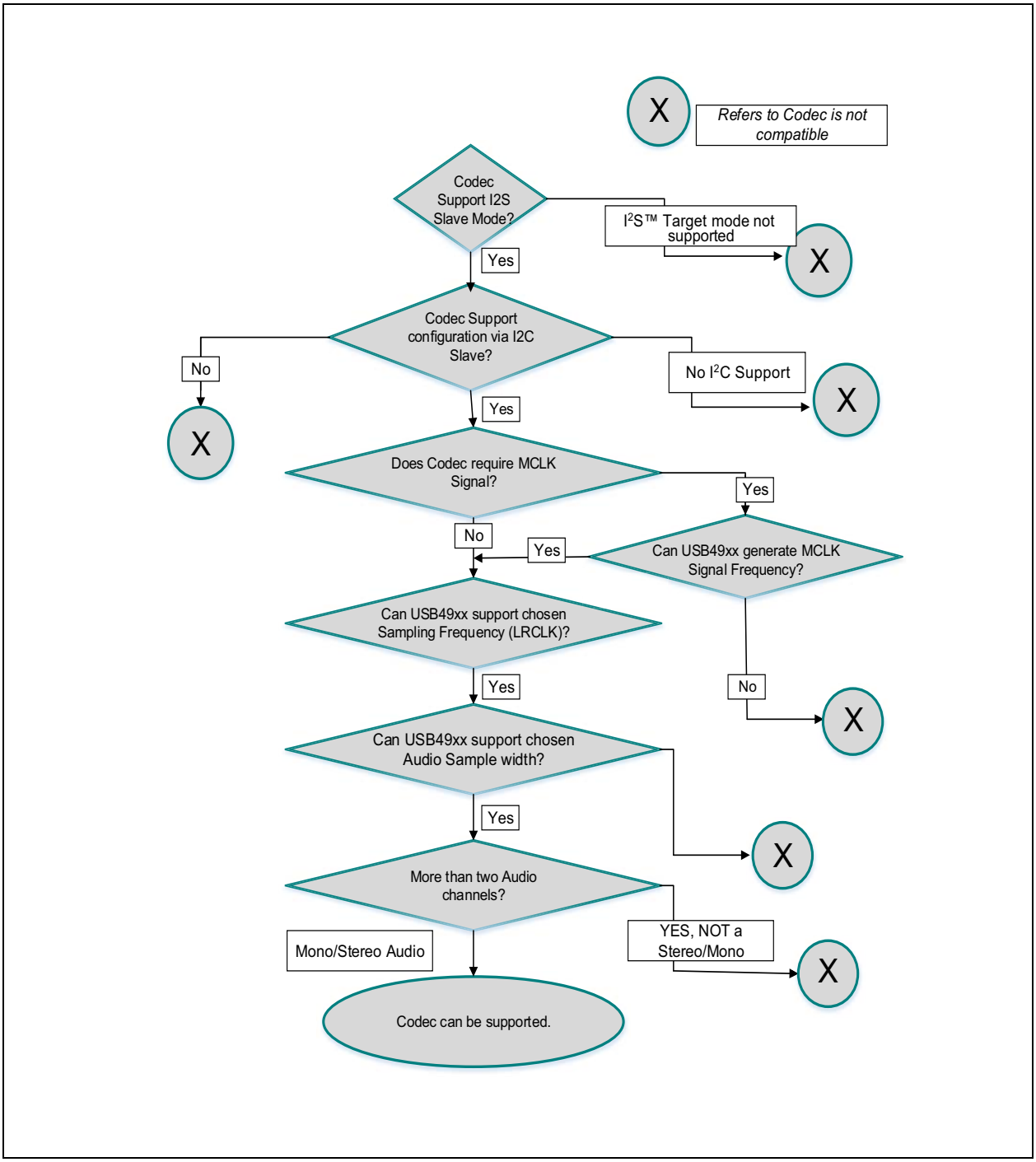
ADAU1960/1961 is the default example used with the I²S Bridge, but other codecs can also be used. The following checklist can be used to determine if a codec is compatible with the I²S Bridge. In order to be compatible with the I²S Bridge, a codec must meet all of the following criteria on the I²S Bridge Codec Compatibility Checklist.

4.3.1 I²S BRIDGE CODEC COMPATIBILITY CHECKLIST

1. The codec supports I²S Target mode.
2. The codec supports configuration via I²S (Target mode).
3. The codec does not require an external MCLK signal, or the I²S Bridge is capable of supplying the CLK frequency that the codec requires.
4. The codec can accept an LCRLK sampling frequency that the I²S Bridge can provide.
5. The codec can accept a sample width that the I²S Bridge can support.
6. The codec has one or two audio channels, but not more than two.

Note:	Traditionally, the I²S communication protocol uses the terminologies, “master” and “slave.” The equivalent Microchip terminologies used in this document are “controller” and “target.”

FIGURE 1: IDENTIFYING CODEC COMPATABILITY



4.3.2 CODEC IMPLEMENTATION STEPS

Once a codec has been identified as compatible, actions must be taken (by means of register access through OTP) to configure both the codec and the I^2S Bridge. Table 3 identifies configuration steps typical for several examples that might be required if changing to a different codec.

TABLE 3: AUDIO PARAMETERS CONFIGURATION

Steps	To Change Audio Sampling Frequency	To Change Audio Codec Device	To Change Audio Sample Resolution (Bits/Sample)	To Change Number of Audio Channel Stereo/Mono	To Disable Speaker/ Mic Interface
Step 1	Modifying the USB Audio Descriptors to advertise the new sampling frequency	Modifying the Codec Initialization	Modifying the USB Audio Descriptors to advertise the new sample resolution	Modifying the USB Audio Descriptors to advertise the number of Audio channels supported	Modify register configuration to choose Speaker/Mic interface
Step 2	Modifying the MCLK Frequency (The MCLK signal given to the codec has a frequency which is basically N times the Sampling frequency.)	Modifying the MCLK Frequency based on the operating requirements of the new codec (The MCLK signal given to the codec has a frequency which is basically N times the Sampling frequency. Value N differs between codecs.)	Modifying the I2S Baud Rate according to the new bits/sample value	Modify the I^2S registers to work in Configuring Mono Audio Mode.	Modifying the Codec Initialization sequence to be identified by the user
Step 3	(IN/OUT direction FIFOs accumulate the Audio samples and have high and low threshold levels used for maintaining synchronization with the Host. The threshold levels depend on the Sampling frequency.)	Configure the CODEC Access Information.	(IN/OUT direction FIFOs accumulate the Audio samples and have high and low threshold levels used for maintaining synchronization with the Host. The threshold level depends on the bits/sample. Also, the FIFO element width changes with the sample size.)	(IN/OUT direction FIFOs accumulate the Audio samples and have high and low threshold levels used for maintaining synchronization with the Host. The threshold level depends on the number of channel in the audio stream.)	—
Step 4	Modifying the Codec Initialization sequence to support the new Sampling frequency	Match other requirements of codec. For example, I^2S mode/ Left Justified/Right Justified modes.	Modifying the Codec Initialization sequence to support the new bit resolution	Modifying the Codec Initialization sequence to support Mono mode	—

4.3.3 MODIFYING THE USB AUDIO DESCRIPTORS

TABLE 4: USB DESCRIPTOR REGISTER DEFINITIONS FOR AUDIO OUT INTERFACE

Register Address	Parameter Name	Size	Description	Supported Values
0xBFD240BA	Spkr_NrChannels	1	The number of channels in the audio data stream	Mono – 0x01 Stereo – 0x02
0xBFD240BB	Spkr_SubFrameSize	1	The number of bytes occupied by one audio subframe (sample)	0x02, 0x03, 0x04
0xBFD240BC	Spkr_BitResolution	1	The number of effective bits from the available bits in the audio sub-frame	16, 24, 32
0xBFD240BE	Spkr_SamFreq	3	Sampling frequency in Hertz for this asynchronous data endpoint. (48 kHz)	8 kHz, 16 kHz, 32 kHz, 44.1 kHz, and 48 kHz (These are standard values.)
0xBFD240C5	Spkr_MaxPacketSize	2	Maximum packet size for this end-point	0 Bytes to 1023 Bytes

TABLE 5: USB DESCRIPTOR REGISTER DEFINITIONS FOR AUDIO IN INTERFACE

Register Address	Parameter Name	Size	Description	Supported Values
0xBFD240EE	Mic_NrChannels	1	The number of channels in the audio data stream	Mono – 0x01 Stereo – 0x02
0xBFD240EF	Mic_SubFrameSize	1	The number of bytes occupied by one audio subframe (sample)	0x02, 0x03, 0x04
0xBFD240F0	Mic_BitResolution	1	The number of effective bits from the available bits in the audio sub-frame	16, 24, 32
0xBFD240F2	Mic_SamFreq	3	Sampling frequency in Hertz for this asynchronous data endpoint (48 kHz)	8 kHz,16 kHz, 32 kHz, 44.1 kHz, and 48 kHz (These are standard values.)
0xBFD240F9	Mic_MaxPacketSize	2	Maximum packet size for this end-point	0 Bytes to 1023 Bytes

4.3.3.1 Sampling Frequency Customization in USB Descriptor

When customizing the sampling frequency, the following fields may require modification:

1) Spkr_SamFreq

This field is 24 bits in length, containing the hexadecimal value of the sampling frequency.

2) Mic_SamFreq

This field is 24 bits in length, containing the hexadecimal value of the sampling frequency.

3) Spkr_MaxPacketSize

This field requires modification because the amount of data received per second varies with the Sampling frequency variation. Refer to MaxPacketSize Calculation.

4) Mic_MaxPacketSize

This field requires modification because the amount of data received per second varies with the Sampling frequency variation. Refer to MaxPacketSize Calculation.

4.3.3.2 Audio Sample Resolution Customization in USB Descriptor

For customizing the Audio Sample Resolution, the following fields may require modification:

1) Spkr_SubFrameSize: The sample length in Number of Bytes

2) Spkr_BitResolution: The effective sample length in Number of Bits

3) Mic_SubFrameSize: The sample length in Number of Bytes

4) Mic_BitResolution: The effective sample length in Number of Bits

Additionally,

1) Spkr_MaxPacketSize

This field requires modification if the Sample length changes the Packet size. Refer to MaxPacketSize Calculation.

2) Mic_MaxPacketSize

This field requires modification if the Sample length changes the Packet size. Refer to MaxPacketSize Calculation.

4.3.3.3 MaxPacketSize Calculation

The amount of data received per millisecond = (Sampling Frequency * Number of Bytes per Sample * Number of Channels in the Audio stream) /1000

For the USB Audio, packet interval is 1 packet/ms based on the Audio endpoint descriptor Interval field. For a 48 kHz, 16-bit Stereo Audio, the amount of Data received per ms = (48000 * 2 * 2)/1000 = 192 bytes/ms. For a 44.1 kHz, 16-bit Stereo Audio, the amount of Data received per ms = 177 bytes/ms. For the ISOC OUT direction, the audio stream (to the speaker), packet size does not vary. This is to maintain audio synchronization with the Host.

For the ISOC IN direction, the audio stream (from the microphone), packet size varies by few samples in order to maintain the synchronisation with the Host. The maximum packet size of the ISOC IN endpoint can be derived from ISOC IN Endpoint Packet Size Calculations.

4.3.4 MODIFYING THE MCLK FREQUENCY

MCLK is used by the I^2S module to derive SCLK and LRCLK by frequency division. Some codecs also use the MCLK from the I^2S controller as an internal clock for performing operations.
In general, MCLK signal expected by the codec will have a frequency that is some multiple of 256 times the sampling frequency:

1) 256 times sampling frequency
2) 512 times sampling frequency
3) 768 times sampling frequency
4) 1024 times sampling frequency

and so on.

4.3.4.1 Deriving MCLK

To derive the MCLK frequency from the source clock, the Reference Oscillator Control module is used. (Refer to Source_Clock in Table 6.) By Default, the Source_Clock is 60 MHz (0000b).

MCLK frequency = Source_Clock/ (2 * Division Factor(D))

The clock division factor (D) is divided into:

1) Integer divider (N)
2) Fractional divider (F)

Example: Division factor = 5.2
Integer divider (N) = 5
Fractional divider (F) = 2

The Reference Oscillator Control module has two registers:
1) REFOCON Register – for configuring the Integer Division value (See Table 6.)
2) REFOTRIM Register – for configuring the Fractional Division value (See Table 7.)

Note 1: The REFOCON Register contains the ON and Active Status and control bits, which protect the REFOCON Register and REFOTRIM Register from being modified unintentionally.

2: Clearing the ON and Active bits is required before the Reference Oscillator module registers can be modified.

TABLE 6: REFOCON REGISTER

Address: **0xBF80A030**			
Bit Position	**Field**	**Permission**	**Description**
31	Reserved_31	RO	When writing to this register, this bit must be 0.
30:16	Integer Division value	RW	0 to 0x7FFF is allowed. The value entered here is Multiplied by 2 and taken for division.
15	ON	RW	1 = Reference Oscillator Module is enabled. 0 = Reference Oscillator Module is disabled.
14	Reserved	RW	When writing to this register, this bit must be 0.
13	Reserved	RW	When writing to this register, this bit must be 0.
12	Reserved	RW	When writing to this register, this bit must be 0.
11	Reserved	RW	When writing to this register, this bit must be 0.
10	Reserved	RO	When writing to this register, this bit must be 0.
9	Reserved	RW	When writing to this register, this bit must be 0.
8	Active	RW	Reference Clock Request Status bit 1 = Reference clock request is active. (Users should not update this REFOCON register.) 0 = Reference clock request is not active. (Users can update this REFOCON register.)
7:4	Reserved	RO	When writing to this register, this bit must be 0.
3:0	Source_Clock	RW	Reference Clock Source Select bits. Select one of various clock sources to be used as the reference clock. 0100 – 1111 = Reserved 0011 = osc_clk[4] = 240 MHz USB Clock/2 (Usually POSC) 0010 = osc_clk[3] = 120 MHz USB Clock/4 (Usually POSC) 0001 = osc_clk[2] = 96 MHz USB Clock/5 0000 = osc_clk[1] = 60 MHz System Clock

TABLE 7: REFOTRIM REGISTER

Address: 0xBF80A034			
Bit Position	**Field**	**Permission**	**Description**
31:20	Value equivalent to fractional divisor. (=4096 * F)	RW	Trim bits – Provides fractional additive to Integer division value for 1/2 period of REFO clock: 0000_0000_0000 = 0/4096 (0.0) divisor added to Integer division value 0000_0000_0001 = 1/4096 (0.0002441) divisor added to Integer division value 1000_0000_0000 = 2048/4096 (0.500000) divisor added to Integer division value 1111_1111_1110 = 4094/4096 (0.9995117) divisor added to Integer division value 1111_1111_1111 = 4095/4096 (0.9997559) divisor added to Integer division value
19:0	Reserved	RO	When writing to this register, these bits must be 0.

4.3.5 MODIFYING THE FIFO THRESHOLD VALUES

USB Audio interface has two endpoints: Isochronous IN direction and Isochronous OUT direction.

- Data received on the ISOC Out endpoint is stored in the I2S_OUT_FIFO and transferred through the I^2S bus to the codec.
- Data received from the I^2S bus is stored in the I2S_IN_FIFO and transferred through the ISOC IN endpoint to the Host.

The size of basic element in the FIFO is either 16 or 32 bits and is configurable through the Table 13 field FIFO Element mode. Verifying this field is necessary when the Audio Sample Resolution is modified.

4.3.5.1 Threshold Setting for ISOC_OUT_FIFO

The Host will send the same amount of data on every frame. For example, 48 kHz of data based on the Host clock. The codec sampling clock is asynchronous to the Host clock. This will cause the amount of data in the OUT FIFO to vary. If the amount of data in the FIFO exceeds the high threshold, then the sampling clock is decreased. If the data is between the high and low thresholds, the sampling clock does not change. If the data falls below the low threshold, the sampling clock is increased.

ISOC Packet interval = 1 ms

Number of Samples in 1 packet = ((Sampling Frequency) /1000)

ISOC_OUT_FIFO contains 512 elements, refer to Table 13 for settings.
Number of Packets FIFO can hold = 512 / (Number of Samples in 1 packet)

It is desirable to operate with data filled up to 50% of the FIFO size. Until then, transmission through I^2S does not start. The Out-start level of the FIFO can be configured in OUT_START_THRESHOLD parameter of Table 15 register.
OUT_START_THRESHOLD = (Number of Packets FIFO can hold / 2) * (Number of Samples in 1 packet)

Maximum drift considered for the USB Packet arrival interval is 1 ms. To monitor this drift, the FIFO is marked with high and low thresholds.
OUT_FIFO_LOW_THRESHOLD = (OUT_START_THRESHOLD - Number of Samples in 1 packet)
OUT_FIFO_HIGH_THRESHOLD = (OUT_START_THRESHOLD + Number of Samples in 1 packet)

Note: Refer to Table 14 and Table 15 for configuring the threshold values.

4.3.5.2 Threshold Setting for ISOC_IN_FIFO

The data from the codec is fed into ISOC_IN_FIFO. Because the sampling clock is asynchronous to the Host clock, the amount of data captured in every USB frame varies. This is a problem left to the Host to deal with. The input FIFO has two markers, a low threshold (THRESHOLD_LOW_VAL), and a high threshold (THRESHOLD_HIGH_VAL). There are three registers to determine how much data to be sent back to each frame. If the amount of data in the FIFO exceeds the high threshold, then HI_PKT_SIZE worth of data is sent. If the data is between the high and low thresholds, the normal MID_PKT_SIZE amount of data is sent. If the data is below the low threshold, LO_PKT_SIZE worth of data is sent.

ISOC Packet interval = 1 ms

Number of samples recorded in 1 ms = ((Sampling Frequency) /1000)

ISOC_IN_FIFO contains 512 elements, refer to Table 13 for settings.

Number of packets FIFO can hold = 512 / (Number of samples in 1 ms)

It is desirable to operate with data filled up to 50% of the FIFO size. Until then, transmission through USB ISOC endpoint does not start. The IN-start level of the FIFO can be configured in IN_START_THRESHOLD parameter of Table 16 register.

IN_START_THRESHOLD = (Number of Packets FIFO can hold / 2) * (Number of samples per millisecond)

The maximum drift considered for the design is 1 ms. To monitor this drift, FIFO is marked with high and low thresholds.
IN_FIFO_HIGH_THRESHOLD = (IN_START_THRESHOLD – Number of samples per millisecond)
IN_FIFO_HIGH_THRESHOLD = (IN_START_THRESHOLD + Number of samples per millisecond)

Note: Refer to Table 16 and Table 17 for configuring the threshold values.

4.3.5.3 ISOC IN Endpoint Packet Size Calculations

The normal packet size of the ISOC In endpoint, when the data is between the high and low threshold levels, is MID-SIZE_PACKET.

MIDSIZE_PACKET = ((Number of samples per millisecond) * (Number of Audio channels) * (Bytes per sample)).

Adjust for drift by adding or subtracting the number of samples recorded in interval of a micro frame,

HIGH_SIZE_PACKET = ((MID_SIZE_PACKET) + (((Number of samples per 125 µS) * (Number of Audio channels) * (bytes per sample)))

4.3.6 MODIFYING THE I²S BAUD RATE

MCLK is used as the source clock for deriving the I2S_SCLK and I2S_LRCLK in the I²S Controller mode.

MCLK frequency = ((Sampling frequency) * (Bits per Sample) * (Number of channels in the Audio))

Refer to Table 21 for configuring the division factor.

4.3.7 MODIFYING THE CODEC INITIALIZATION

The communication between the Audio Codec and the UDC is established over two protocols:

1) I²S interface through which the Audio data is transferred.

2) I²C interface for executing the initialization and control operations.

4.3.7.1 Modifying Codec Initialization

The external codec chip is configured through the I²C interface during the initialization phase. The initialization procedure is a sequence of register configurations identified from the Codec data sheet.

The Codec Initialization Buffer provides memory for storing the data of maximum 75 I²C transactions. Each such I²C transaction can carry up to 11 bytes of data excluding the target address.

The organisation of the Codec Initialization Buffer is mentioned in Table 8. The Byte 0 field indicates the number of the data bytes going to be present in that I²C transaction.

By default, the device firmware will contain a list of transactions in the Codec Initialization Buffer. The values of registers and the order that they are written are not the same for all devices and may change in different firmware releases. In developing a list of transactions for the Codec Initialization Buffer, it is advisable to observe the default initialization sequence on the I²C bus with an I²C analyzer. This will reveal the default Codec Initialization Buffer contents. If any of the default transactions are not desired, then that transaction is removed by writing over that transaction in the buffer. The best practice is to always create a complete transaction list that is the same length or longer than the default list.

TABLE 8: CODEC INITIALIZATION BUFFER

Base-Address: 0xBFD23C00		
Address Offset	**BYTE 0 – (Maximum up to 11)**	**BYTE 1 to BYTE 11**
0x00	Length of Transaction 1	Register Address in the codec + Data to be configured
0x0C	Length of Transaction 2	Register Address in the codec + Data to be configured
0x18	Length of Transaction 3	Register Address in the codec + Data to be configured
:	Length of Transaction 4	Register Address in the codec + Data to be configured
:	Length of Transaction 5	Register Address in the codec + Data to be configured
:	Length of Transaction 6	Register Address in the codec + Data to be configured
:	Length of Transaction 7	Register Address in the codec + Data to be configured
....	....	Register Address in the codec + Data to be configured
0x378	Length of Transaction 75	Register Address in the codec + Data to be configured

4.3.7.2 CODEC Access Information

The codec information buffer contains:

- The pointers to the codec volume and MUTE registers, which are accessed when volume change and mute operations are done from the Host.
- The volume range and volume resolution information, which are sent to the Host upon Get_Max, Get_Min, and Get_Resolution requests.

TABLE 9: CODEC INFORMATION BUFFER

Base-Address: 0xBFD23780			
Field	**Address Offset**	**Size in Bytes**	**Value**
Speaker Left Channel Volume Control – Register Address	0	4	Codec Register Address of Left Channel Speaker Volume control
Speaker Right Channel Volume Control – Register Address	4	4	Codec Register Address of Left Channel Speaker Volume control
Speaker Left Channel Volume Mute – Register Address	8	4	Codec Register Address of Left Channel Speaker MUTE control
Speaker Right Channel Volume Mute – Register Address	0x0c	4	Codec Register Address of Left Channel Speaker MUTE control
Speaker Volume dB Resolution Value	0x10	4	Minimum change that can be produced in the volume, expressed in dB (IEEE754 representation for hex value)
Reserved	0x14	8	N/A
I^2C Control Interface – clock frequency to be used	0x1C	2	Value: 1) 0x0A0D for 400 kHz
Least volume that can be produced by the speaker	0x1E	2	Minimum volume supported by the codec in dB
Maximum volume that can be produced by the speaker	0x20	2	Maximum volume supported by the codec in dB
Register value equivalent of 0 dB volume	0x22	2	Codec Volume register value to be configured in order to get the 0 dB volume in speaker
Initial Left Channel volume to be set	0x24	2	Initial volume level for the Left Channel in dB
Initial Right Channel volume to be set	0x26	2	Initial volume level for the Right Channel in dB
RESERVED2	0x28	2	N/A
I^2C Target Address of the Codec	0x2C	1	7-bit hardware target address of codec
Initialization Sequence Count	0x2D	1	Specifies the number of transactions that are present in the Codec Initialization Buffer (a hexadecimal number)
Codec Register Address Width	0x2E	1	1 – Byte 2 – Word (2 bytes) 4 – Double word (4 bytes)
Codec Register Value Field Width	0x2F	1	1 – Byte 2 – Word (2 bytes) 4 – Double word (4 bytes)
Codec Volume Register Bit Size	0x30	1	Number of bits in the (Left/Right) Volume Control register used as volume field
Codec MUTE Register Bit Value	0x31	1	0 – LOGIC HIGH indicates MUTE (or) 1 – LOGIC LOW indicates MUTE

TABLE 9: CODEC INFORMATION BUFFER (CONTINUED)

Base-Address: 0xBFD23780			
Field	**Address Offset**	**Size in Bytes**	**Value**
Start bit of the Left Channel Volume field in the Left Channel Volume Control Register of Codec	0x32	1	Start bit value
Start bit of the Right Channel Volume Field in the Right Channel Volume Control Register of Codec	0x33	1	Start bit value
Start bit of the Left Channel MUTE field in the Left Channel MUTE Control Register of Codec	0x34	1	Start bit value
Start bit of the Right Channel MUTE field in the Right Channel MUTE Control Register of Codec	0x35	1	Start bit value
ADAU1961	0x36	1	1 – The codec is ADAU1961. 0 – A different codec is used.

4.3.7.3 Volume Control Parameters

Codecs generally have registers for volume control operation. Table 10 explains the example of AK4642EN codec in which the relationship between the register value and the volume in dB relation is mentioned.

TABLE 10: VOLUME CONTROL PARAMETERS FOR AK4642EN

DVL/R7-0	Gain in (dB)	Parameters Required in Codec Access Information Buffer
00H	+12.0 dB	Maximum volume that can be produced by the speaker
01H	+11.5 dB	—
02H	+11.0 dB	—
:	:	—
18H	0 dB	Register value equivalent of 0 dB volume
:	:	—
FDH	–114.5 dB	—
FEH	+115.0 dB	Minimum achievable volume that can be produced by the speaker

The following can be inferred from Table 10:
1. Volume increases with decrease in register value (indirectly proportional). By default, such codecs are not supported by Microchip I^2S Bridges. This requires an OTP PATCH for implementing this requirement. Codecs, in which register value and Volume are directly proportional, are supported by the default firmware.
2. The relationship between the register values and volume is linear. Only codecs with linear volume-to-Register value relationship are supported by the default firmware. A non-linear relationship requires an OTP patch.
3. The volume resolution is 0.5 dB.

4.4 Miscellaneous Settings

4.4.1 ENABLING AND DISABLING I^2S BRIDGE INTERFACES

I^2S Bridge allows control of additional settings. See Table 11 and Table 12.

TABLE 11: Enable/Disable I^2S

Address: 0xBFD23412; Size: 1 Byte	
Setting	**Value**
USB – I^2S™ Bridge disable	0x00
Only Audio IN mode	0x01
Only Audio Out mode	0x02
Both Audio IN and Out mode	0x03 (Default)

TABLE 12: ENABLING/DISABLE THE HID INTERFACE FOR MIKE_DETECT

Address: 0xBFD23413; Size: 1 Byte	
Setting	**Value**
MIKE-Detection HID Interface disable	0x00
Mute/Unmute Audio IN mode	0x01
Mute/Unmute Audio Out mode	0x02
Mute/Unmute Audio IN and Out mode	0x03 (Default)

4.5 Modifying the I²S Modes

4.5.1 I²S MODE

In I²S mode, the transmitter drives the MSB of the audio data on the first falling edge of SCK after an LRC transition. The receiver samples the MSB on the second rising edge of SCK. The left channel data transmits while LRC is low, and the right channel transmits while LRC is high. A frame transmits left channel first then the right channel.

To be I²S-compliant, the configuration bits in SPIxCON2 must be set as follows:
AUDMOD = 00, FRMPOL = 0, and CKP = 1.
These values cause SDO and LRC transitions to occur on the falling edge of SCK and sampling of SDI to occur on the rising edge of SCK. It also starts a frame with LRC falling edge transition. The register configuration field is present in Table 22 register.

4.5.2 LEFT JUSTIFIED MODE

In Left Justified mode, the transmitter drives the MSB of the audio data on the SCK edge that is coincident with an LRC transition. The receiver samples the MSB on the next SCK edge. Codecs using justified protocols usually default to transmitting data on the rising edge of SCK and receiving data on the falling edge of SCK. Another convention is that LRC is high for the left channel and low for the right channel which is opposite of I²S. However, they maintain the left channel followed by the right channel (in a frame).

To configure for the left justified standard convention, set the following bits in SPXIxCON as follows:
AUDMOD = 01, FRMPOL = 1, CKP = 0. These fields are present in Table 22 register.

4.5.3 RIGHT JUSTIFIED MODE

In Right Justified mode, the transmitted drives the MSB of the audio data on the n^{th} transmit edge of SCK, such that the LSB is available on the receive edge of SCK, preceding a transition of LRC. When set to transmit (DISSDO = 0), this device drives the unused bit slots (preceding the audio) with logic level 0. When set to receive (DISSDI = 0), this device ignores the unused bit slot. Right Justified mode is configured as follows:
AUDIOD = 10, FRMPOL = 1, CKP= 0. These fields are present in Table 22 register.

4.5.4 CONFIGURING MONO AUDIO MODE

The Audio Protocol function can transmit mono audio data on both the left and right channels. When AUDMONO = 1, the Shift register uses each FIFO location twice. This gives each channel the same mono stream of audio data. When AUDMONO = 0, the Shift register uses each FIFO location once. This gives each channel a unique stream of data for stereo audio. Receive data is not affected by AUDMONO. This bit is present in Table 22 register.

4.5.5 CONFIGURATION REGISTERS

TABLE 13: SPIXCON REGISTER

Address: 0xBF80A040			
Bit Position	**Field**	**Permission**	**Description**
31	frmen	Reserved	Strictly do not modify
30	frmsync	Reserved	Strictly do not modify
29	frmpol	Reserved	Strictly do not modify
28	mssen	Reserved	Strictly do not modify
27	frmsypw	Reserved	Strictly do not modify
26:24	frmcnt	Reserved	Strictly do not modify
23	mclken	Reserved	Strictly do not modify
22:18	reserved_22_18	Reserved	Strictly do not modify
17	spife	Reserved	Strictly do not modify
16	enhbuf	Reserved	Strictly do not modify
15	on	Reserved	Strictly do not modify
14	frz	Reserved	Strictly do not modify

Note 1: To avoid changing Reserved bits, use the SetBit and ClearBit features of MPLAB® Connect Configurator.

TABLE 13: SPIXCON REGISTER (CONTINUED)

Address: 0xBF80A040			
Bit Position	**Field**	**Permission**	**Description**
13	sidl	Reserved	Strictly do not modify
12	dissdo	Reserved	Strictly do not modify
11:10	FIFO Element mode	RW (Note 1)	1 1 => 32-bit FIFO element, with Audio Sample effective size = 24 bits, Number of bits per channel = 32 (Refer SubFrameSize in USB Audio descriptor) 1 0 => 32-bit FIFO element, with Audio Sample effective size = 32 bits, Number of bits per channel = 32 (Refer SubFrameSize in USB Audio descriptor) 0 1 =>16-bit FIFO element, with Audio Sample effective size = 16 bits, Number of bits per channel = 32 (Refer SubFrameSize in USB Audio descriptor) 0 0 =>16-bit FIFO element, with Audio Sample effective size = 16 bits, Number of bits per channel = 16 (Refer SubFrameSize in USB Audio descriptor)
9	smp	Reserved	Strictly do not modify
8	cke	Reserved	Strictly do not modify
7	ssen	Reserved	Strictly do not modify
6	ckp	Reserved	Strictly do not modify
5	msten	Reserved	Strictly do not modify
4	dissdi	Reserved	Strictly do not modify
3:2	stxisel	Reserved	Strictly do not modify
1:0	srxisel	Reserved	Strictly do not modify
31	fifo_full	RO	—
30	fifo_threshold_high	RO	—
29	fifo_threshold_low	RO	—
28	fifo_empty	RO	—
27:19	threshold_high_val	RW	OUT_FIFO_HIGH_THRESHOLD
18:10	threshold_low_val	RW	OUT_FIFO_LOW_THRESHOLD
9:0	occupied	RO	—

Note 1: To avoid changing Reserved bits, use the SetBit and ClearBit features of MPLAB® Connect Configurator.

TABLE 14: OUT_FIFO_THRESHOLD_CTL

Address: 0xBF80A000			
Bit Position	**Field**	**Permission**	**Description**
31	out_enable	RW	Strictly do not modify
30	out_started	RW	Strictly do not modify
29:13	reserved_29_13	RO	Strictly do not modify
12	out_swap	RW	Strictly do not modify
11	mcu_access_enable	RW	Strictly do not modify
10	dma_access_enable	RW	Strictly do not modify
9	enable	RW	Strictly do not modify
8:0	out_threshold_start	RW	OUT_THRESHOLD_START

TABLE 15: OUT_FIFO_START_CTL

Address: 0xBF80A004			
Bit Position	**Field**	**Permission**	**Description**
31	out_enable	RW	Strictly do not modify
30	out_started	RW	Strictly do not modify
29:13	reserved_29_13	RO	Strictly do not modify
12	out_swap	RW	Strictly do not modify
11	mcu_access_enable	RW	Strictly do not modify
10	dma_access_enable	RW	Strictly do not modify
9	enable	RW	Strictly do not modify
8:0	out_threshold_start	RW	OUT_THRESHOLD_START

TABLE 16: IN_FIFO_START_CTL

Address: 0xBF80A004C			
Bit Position	**Field**	**Permission**	**Description**
31	in_enable	RW	Strictly do not modify
30	in_started	RW	Strictly do not modify
29:13	reserved_29_13	RO	Strictly do not modify
12	in_swap	RW	Strictly do not modify
11	mcu_access_enable	RW	Strictly do not modify
10	dma_access_enable	RW	Strictly do not modify
9	enable	RW	Strictly do not modify
8:0	in_threshold_start	RW	IN_THRESHOLD_START

TABLE 17: IN_FIFO_THRESHOLD_CTL

Address: 0xBF80A008			
Bit Position	**Field**	**Permission**	**Description**
31	fifo_enable	RO	—
30	fifo_threshold_high	RO	—
29	fifo_threshold_high	RO	—
28	fifo_empty	RO	—
27:19	threshold_high_val	RW	IN_FIFO_HIGH_THRESHOLD
18:10	threshold_low_val	RW	IN_FIFO_LOW_THRESHOLD
9:0	occupied	RO	—

TABLE 18: IN_ENDPOINT_HIGHPACKET_SIZE

Address: 0xBF80A010			
Bit Position	**Field**	**Permission**	**Description**
31:10	reserved_31_10	RO	Write 0s
9:0	hi_pkt_size	RW	HIGH_PACKET_SIZE

TABLE 19: IN_ENDPOINT_MIDPACKET_SIZE

Address: 0xBF80A014			
Bit Position	**Field**	**Permission**	**Description**
31:10	reserved_31_10	RO	Write 0s
9:0	mid_pkt_size	RW	MID_PACKET_SIZE

TABLE 20: IN_ENDPOINT_LOWPACKET_SIZE

Address: 0xBF80A014			
Bit Position	**Field**	**Permission**	**Description**
31:10	reserved_31_10	RO	Write 0s
9:0	low_pkt_size	RW	LOW_PACKET_SIZE

TABLE 21: I2S_BAUDRATE_GENERATOR

Address: 0xBF80A070			
Bit Position	**Field**	**Permission**	**Description**
31:10	reserved_31_13	RO	Reserved Write 0s
9:0	BRG	RW	BRG value is such that, Baudrate = (MCLK Frequency) / (2* (BRG + 1))

TABLE 22: SPICON2

Address: 0xBF80A080			
Bit Position	**Field**	**Permission**	**Description**
31:16	Reserved_31_16	RO	Strictly do not modify
15	spisgnext	RW	Strictly do not modify
14:13	Reserved_14_13	RW	Strictly do not modify
12	frmerren	RW	Strictly do not modify

TABLE 22: SPICON2 (CONTINUED)

Address: 0xBF80A080			
Bit Position	**Field**	**Permission**	**Description**
31:16	Reserved_31_16	RO	Strictly do not modify
11	spiroven	RO	Strictly do not modify
10	spituren	RW	Strictly do not modify
9	ignrov	RW	Strictly do not modify
8	igntur	RW	Strictly do not modify
7	auden	RW	Strictly do not modify
6:4	Reserved_6_4	RO	Strictly do not modify
3	audmono	RW	1 = Audio Data is Mono. (Each data word is transmitted on both left and right channels.) 0 = Audio Data is Stereo.
2	Reserved_2	RO	Strictly do not modify
1:0	audmod	RW	Audio Protocol mode 11 = PCM/DSP mode 10 = Right Justified mode 01 = Left Justified mode 00 = I^2S™ mode

4.6 Sample Codec Configuration – Following the Guidelines

By default, the I²S Bridge is configured to operate with the ADAU1961 codec as specified in Table 23. Table 24 shows how to configure the I²S Bridge for an alternative codec, the AK4642EN.

TABLE 23: CODEC INITIALIZATION SEQUENCE FOR AK4642 CODEC

Address	Byte 0 (Length of the I²C Transaction)	Byte 1 (Register Address in the AKM Codec)	Byte 2 (Register Configuration Value)
0x00	2	0x5	0x23
0x0C	2	0x0F	0x1
0x18	2	0x0E	0x1
0x24	2	0x9	0x91
0x30	2	0x0C	0x91
0x3c	2	0x0A	0x18
0x48	2	0x0D	0x18
0x54	2	0x0	0x6D
0x60	2	0x1	0x39
0x6c	2	0x1	0x79
0x78	2	0x2	0x14
0x84	2	0x4	0x3
0x90	2	0x10	0x27

TABLE 24: CODEC INFORMATION BUFFER (FOR AK4642EN)

Field	Address Offset	Value	Description
Speaker Left Channel volume control – Register Address	0	0xA	AK4642_REG_LDAC_VOL
Speaker Right Channel volume control – Register Address	4	0x0D	AK4642_REG_RDAC_VOL
Speaker Left Channel Mute – Register Address	8	0x0E	AK4642_REG_MODE_CTRL3
Speaker Right Channel Mute – Register Address	0x0c	0x0E	AK4642_REG_MODE_CTRL3
Speaker Volume dB resolution value	0x10	0.5 dB	Resolution of the volume is 0.5 db (Use IEEE754 representation for hex value https://www.h-schmidt.net/FloatConverter/IEEE754.html)
Reserved	0x14	NA	Reserved
I²C Control Interface – Clock frequency to be used	0x1C	0x0A0D	400 kHz I²C frequency
Least volume that can be produced by the speaker	0x1E	–115 dB	Least achievable volume is –115 dB.
Maximum volume that can be produced by the speaker	0x20	12 dB	0 dB volume corresponds to value 12 in the volume control register.
Register value equivalent of 0 dB volume	0x22	12 dB	0 dB volume corresponds to value 12 in the volume control register.

Note 1: In AK4642, volume decreases with increase in register value, which is not supported by USB49xx. It requires a firmware OTP patch to fix the volume control operation, which is not included in this document.

TABLE 24: CODEC INFORMATION BUFFER (FOR AK4642EN) (CONTINUED)

Field	Address Offset	Value	Description
Initial Left Channel volume to be set	0x24	0x60	–9 dB is the initial volume.
Initial Right Channel volume to be set	0x26	0x60	–9 dB is the initial volume.
RESERVED2	0x28	NA	Reserved
I^2C Target Address of the external codec (7-bit address)	0x2C	0x13	Target address of AKM codec
Initialization sequence count	0x2D	13	Number of I^2C transaction required to initialize the AKM codec
Codec Register Address Width	0x2E	1	Codec register width is 1 byte.
Codec Register Value Field Width	0x2F	1	Codec register address size is 1 byte.
Codec Volume Register Bit Size	0x30	8	Speaker volume is an 8-bit register value. Range 0-255
Codec MUTE Register Bit Value	0x31	1	0 – If the bit is set to logic high, mute is implemented. 1 – If the bit is set to logic low, then mute is implemented.
Start bit of the Left Channel Volume field in the Left Channel Volume Control register of codec	0x32	0	Starts in bit 0.
Start bit of the Right Channel Volume field in the Right Channel Volume Control register of codec	0x33	0	Starts in bit 0.
Start bit of the Left Channel MUTE field in the Left Channel MUTE Control register of codec	0x34	5	Mute position is bit 5.
Start bit of the Right Channel MUTE field in the Right Channel MUTE Control register of codec	0x35	5	Mute position is bit 5.
Is the codec - ADAU 1961	0x36	0	0 – Different codec chip.

Note 1: In AK4642, volume decreases with increase in register value, which is not supported by USB49xx. It requires a firmware OTP patch to fix the volume control operation, which is not included in this document.

4.7 Sample Configuration for Audio Sampling Frequency Change

Table 25 shows examples of changing the sampling frequency to 8 kHz.

TABLE 25: CHANGING THE SAMPLING FREQUENCY TO 8 KHZ

Parameter	Address	Size in Bytes	Value	What it means	Description
USB Audio Descriptor-related Changes					
tSamFreq (I^2S™ IN)	0xBFD240F2	3	40 1F 00	8 kHz	Sampling frequency IN direction
tSamFreq (I^2S OUT)	0xBFD240BE	3	40 1F 00	8 kHz	Sampling frequency IN direction
Maximum Packet Size IN Direction	0xBFD240F9	2	0x28 0x00	40 Bytes	Endpoint descriptor's, maximum packet size
Maximum Packet Size IN Direction	0xBFD240C5	2	0x28 0x00	40 Bytes	Endpoint descriptor's, maximum packet size
Register Settings for I^2S IN Direction to Operate in 8 kHz					
IN Start Threshold	0xBF80A002	4	0x00 0x07 0x00 0xc0	256 Samples	Bits 0-8 Start threshold value
IN High Threshold	0xBF80A008	4	0x00 0xE0 0x43 0x08	264 Samples	Bits 19-27 High threshold value
IN Low Threshold				248 Samples	Bits 10-18 Low threshold value
IN Mid Packet Size	0xBF80A014	4	0x20	32 Bytes	—
IN High Packet Size	0xBF80A010	4	0x24	36 Bytes	—
IN Low Packet Size	0xBF80A018	4	0x1C	28 Bytes	—
Register Settings for I^2S OUT Direction to Operate in 8 kHz					
Out Start Threshold	0xBF80A004	4	0x00 0x07 0x00 0x00	256 Samples	Bits 0-8 Start threshold value
Out High Threshold	0xBF80A000	4	0x00 0xE0 0x43 0x08	248 Samples	Bits 19-27 High threshold value
Out Low Threshold				264 Samples	Bits 10-18 Low threshold value
Frequency-related Register Settings for 8 kHz Operation					
REFOCONN	0xBF80A030	4	0x03 0x10 0x09 0x00	ON and Active bit clear	Step 1 => To make the register write enable
REFOCONN	0xBF80A030	4	0x03 0x10 0x3A 0x00	—	Configures the Integer Division value
REFOTRIM	0xBF80A034	4	0x00 0x00 0x00 0x98	—	Configures the Fractional division value
REFOCONN	0xBF80A030	4	0x03 0x91 0x3A 0x00	—	Makes the register settings active. (Register is write protected again.)
SPIXBRG	0xBF80A070	1	0x03	—	Baud Rate Configuration register

Note 1: In these examples, the coded changes related to 8 kHz sampling frequency are not included.

4.8 Summary of Unsupported Features

1) The I^2S Bridge supports only USB Audio Class 1.0. USB Audio class 2.0 is not supported.

2) The codec must have an I^2C target interface for control. Other control interfaces (such as SPI) are not supported.

3) Volume control of MIC Audio IN is not supported.

APPENDIX A: APPLICATION NOTE REVISION HISTORY

TABLE A-1: REVISION HISTORY

Revision Level & Date	Section/Figure/Entry	Correction
DS00003135B (11-12-21)	Section 4.3.4, "Modifying the MCLK Frequency" and Section 4.3.4.1, "Deriving MCLK"	Defined MCLK frequency variable and provided examples of how to configure and modify its frequency.
	All	Made minor text changes throughout the document.
DS00003135A (07-03-19)	All	Initial release.

THE MICROCHIP WEBSITE

Microchip provides online support via our WWW site at www.microchip.com. This website is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the website contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip website at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the website at: http://microchip.com/support

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

ISBN:978-1-5224-9294-8

For information regarding Microchip’s Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733

China - Beijing
Tel: 86-10-8569-7000

China - Chengdu
Tel: 86-28-8665-5511

China - Chongqing
Tel: 86-23-8980-9588

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115

China - Hong Kong SAR
Tel: 852-2943-5100

China - Nanjing
Tel: 86-25-8473-2460

China - Qingdao
Tel: 86-532-8502-7355

China - Shanghai
Tel: 86-21-3326-8000

China - Shenyang
Tel: 86-24-2334-2829

China - Shenzhen
Tel: 86-755-8864-2200

China - Suzhou
Tel: 86-186-6233-1526

China - Wuhan
Tel: 86-27-5980-5300

China - Xian
Tel: 86-29-8833-7252

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444

India - New Delhi
Tel: 91-11-4160-8631

India - Pune
Tel: 91-20-4121-0141

Japan - Osaka
Tel: 81-6-6152-7160

Japan - Tokyo
Tel: 81-3-6880- 3770

Korea - Daegu
Tel: 82-53-744-4301

Korea - Seoul
Tel: 82-2-554-7200

Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

Malaysia - Penang
Tel: 60-4-227-8870

Philippines - Manila
Tel: 63-2-634-9065

Singapore
Tel: 65-6334-8870

Taiwan - Hsin Chu
Tel: 886-3-577-8366

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600

Thailand - Bangkok
Tel: 66-2-694-1351

Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4485-5910
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-72400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-7288-4388

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820