TI Designs
66AK2G02 DSP + ARM Processor Audio Processing Reference Design

Design Overview

This TI Reference Design is a reference platform based on the 66AK2G02 DSP + ARM Processor and companion AIC3106 Audio codec. This audio solution design also includes real time application software using TI RTOS software that demonstrates audio processing block on the DSP to add audio effects. The reference design is tested and includes 66AK2G02 Evaluation Module (EVM) hardware reference, and Processor SDK software.

Design Resources

TIDEP0069	Design Folder
66AK2G02	Product Folder
AIC3106	Product Folder
K2G General Purpose EVM	EVM Tool Folder
Processor SDK for K2G	Software Folder

Block Diagram

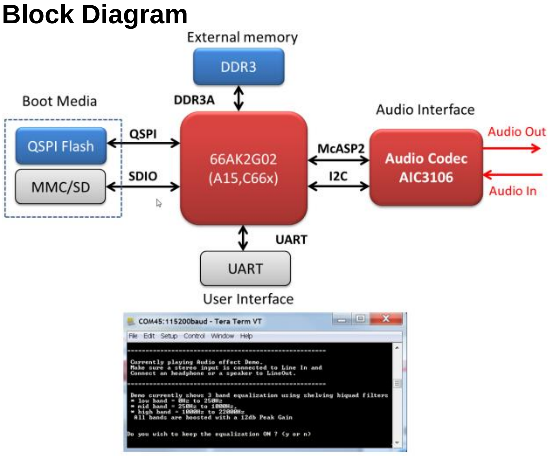


Figure 1: Audio Reference Design

Design Features

- Programmable three-band equalization
- Bass, mid and treble gain controls
- Simple Serial user interface
- Out-of-the box testing with K2G GP EVM
- 8Khz to 96Khz sample rates
- This design is tested with 66AK2G02 GP EVM and includes design files, software, Getting Started guide and User guide to rebuild and modify the software.

Featured Applications

- Home audio
- Automotive Amplifier system
- Soundbars
- Professional audio
- Automated Speech recognition (ASR) preprocessing

Board Image

Figure 2: TI 66AK2G02 General Purpose EVM

1 Introduction

This TI Reference Design is a reference platform based on the 66AK2G02 DSP + ARM Processor SoC and companion AIC3106 Audio codec. This audio solution design also includes real time application software using TI RTOS software that demonstrates audio processing block on the DSP to add audio effects. The reference design is tested and includes the 66AK2G02 EVM hardware reference and Processor SDK software. This Audio Reference Design demonstrates TI 66AK2G02 in an audio processing role targeted at the home audio and automotive audio application. The reference design runs on 66AK2G02 General Purpose EVM board with a TLV320AIC3106IRGZT audio codec. This reference design demonstrates the use of the 66AK2G02 in a stereo audio application, and implements a 3-band audio equalizer on the C66x DSP using shelving biquad filters with adjustable gain settings. The design demonstrates a simple implementation of a digital audio equalizer which allows audio developers to alter the tonal quality of audio passing through it. The developer can tune the gain applied to each of the 3 frequency bands over a UART based user interface, and test the performance in a realistic environment. Furthermore, the developer can leverage the Audio Equalization Reference Design software, to prototype their own audio algorithms on the 66AK2G02 processor or quickly adapting it as necessary to meet the product's specific requirements.

Some key features of the audio reference design are:

- Analog input and Analog output audio
- Programmable sampling rate – 8 – 96 kHz
- Cascade biquad filter implementation of 3-band equalizer.
- Bass, Mid and Treble control using low and high shelf filters.
- Output gain control setting from –18 dB to 18 dB
- TI RTOS based DSP application with user interface over UART.

2 System Description

A block diagram of a stereo audio application on TI 66AK2G02 General Purpose EVM is shown below.

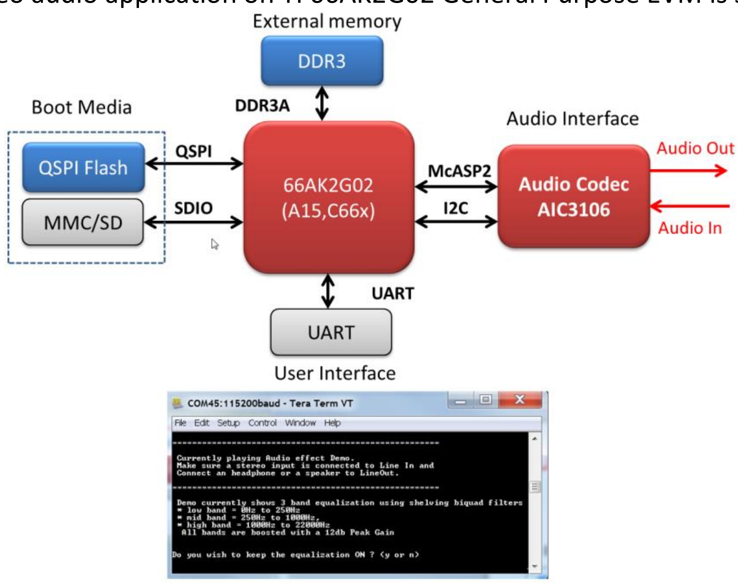


Figure 3: Audio Equalization application on 66AK2G02 General Purpose EVM

The audio equalizer is designed for use in various audio applications such as soundbars and automotive audio amplifiers. A typical design will include a speaker/headphone and audio input source. The speaker and stereo input are connected to a TI Analog Interface Chip (AIC3106). The AIC includes a microphone interface as well as a speaker amplifier, eliminating the need for additional circuitry. Furthermore, the AIC has integrated anti-aliasing filters. The AIC is connected to the 66AK2G02 System on chip (SoC) via the Multi-Channel Audio Serial Port (McASP) as shown in Figure 3. The McASP supports an I2S interface.

The design includes a C66x DSP on the 66AK2G02 plus a terminal, an audio source and speakers/headphones. In the 66AK2G02 General Purpose EVM-based demonstration, we use a PC to provide the terminal and audio source. The PC is connected to the DSP via a USB-Serial interface on 66AK2G02 General Purpose EVM. In case of the EVM, a 2GByte DDR3 is connected to the 66AK2G02 SoC DDR interface, and provides external RAM for the application. The audio equalizer application is booted on the SoC using either the S25FL512 flash device connected to the Quad SPI interface or the SD/MMC card using the SDIO interface on the chip.

The A15 on the 66AK2G02 can also be used as the control user interface where the designer can use IPC mechanism to pass control messages to the equalizer. This integration afforded by the 66AK2G02 will save cost, board space, and also allows for sharing of external memory and MSMC between the C66x and A15.

3 Block Diagram

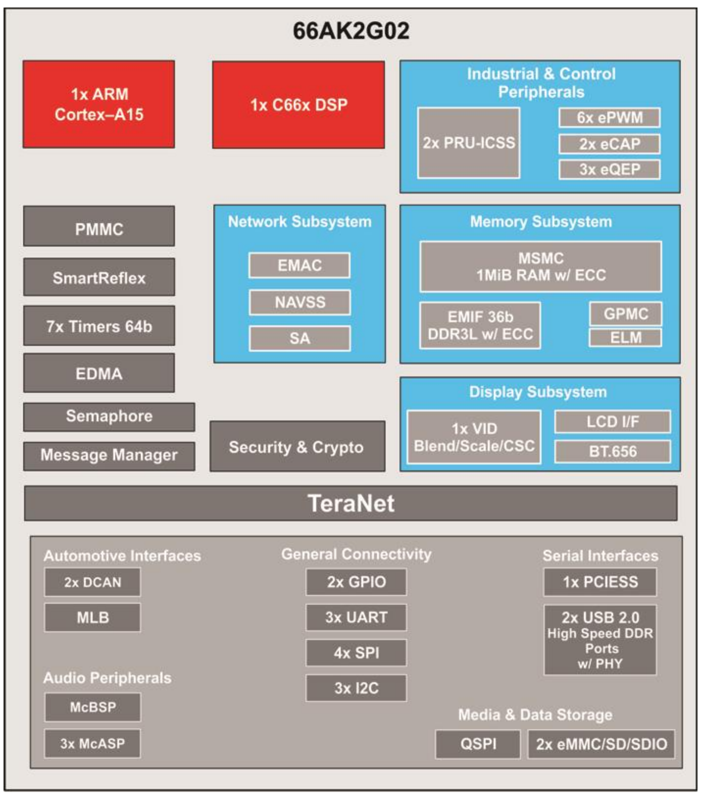


Figure 4: 66AK2G02 DSP + ARM Processor

3.1 Highlighted Products

The TI Audio Equalizer design is based on the TI 66AK2G02 DSP+ARM processor and its associated peripherals and is the engine of the design. The 66AK2G02 SoC is a high performance, highly integrated device based on TI KeyStone II Multicore SoC architecture. It incorporates the performance optimized Cortex™-A15 and a C66x™ DSP core, built to meet the processing and system level integration needs of automotive audio and performance audio applications and industrial communications and control.

The following 66AK2G02 features are highlighted in the TI Design:

DSP Subsystem

The C66x DSP subsystem provides the following main features, among others:

- Fixed/Floating-point C66x CPU based on a superset of the C64x+ and C67x+ ISA
- 32-KiB L1D and 32-KiB L1P cache or addressable SRAM
- 1024-KiB local L2 cache or addressable SRAM
- Extended Memory Controller (XMC)
- Address extension unit to 36-bit address
- Memory protections for multiple segments, and for internal L1/L2 RAM
- Error Detection for L1P
- Error Detection and Correction for L1D, and for L2
- Integrated interrupt controller
- Support for one integrated general-purpose timer, in addition to one device-level watchdog timer
- Debug and trace features

Audio Peripherals:

Three Multichannel Audio Serial Port (McASP) function as a general-purpose audio serial port optimized for the needs of multichannel audio applications.

The McASP has the following features:

- Transmit/Receive Clocks up to 50MHz
- Two Independent Clock Zones and Independent Transmit/Receive Clocks per McASP
- Up to 16-, 10-, 6-Serial Data Pins McASP0/1/2, Respectively
- Supports TDM, I2S, and Similar Formats
- Supports DIT Mode
- Built-In FIFO Buffers for Optimized System

General Connectivity:

- Three Inter-Integrated Circuit (I2C) Interfaces, Each Supports:
  - Standard (up to 100KHz) and Fast (up to 400KHz) Modes
  - 7-Bit Addressing Mode
  - Supports EEPROM Size Up to 4Mbit
- Three UART Interfaces
  - All UARTs are 16C750-Compatible and Operate at up to 3M Baud
  - UART0 Supports 8 Pins With Full Modem Control, With DSR, DTR, DCD, and RI Signals
  - UART1 and UART2 Are 4-Pin Interfaces

For more information on the 66AK2G02; including Datasheets, Silicon Errata, and Technical Reference Manuals; click here.

4 System Overview

The TI Design demonstrates an audio loopback scenario with an audio processing block implemented on 66AK2G02 DSP. This TI Design demonstrates an implementation of a digital audio processing block that can be used to enhance or add effect to the original audio. As a sample audio processing block, the TI design implements a 3 band audio equalizer using biquad shelving filters with adjustable gains. An audio equalizer is an audio processing block that allows sound engineers to boost or attenuate the volume of specific frequencies within the original audio. During the process of sound mixing, equalization is used to correct problems that may have been built into the audio during the recording or to tune compatibility among musical instruments. Equalization is also used to produce original effects or to produce desire response on a speaker or a headphone.

4.1 Audio Input and Output:

The TI Design is setup to receive the audio samples received on 66AK2G02 McASP port through the AIC3106 codec and plays the processed audio samples back through the audio output port of the 66AK2G02 General Purpose EVM.

4.1.1 AIC3106 Audio Codec

The AIC3106 audio codec on the TI 66AK2G02 General Purpose EVM acts as the front end that converts analog audio signal to digital audio samples in the required format. The AIC3106 contains stereo audio DAC and ADC that support sampling frequencies from 8 kHz to 96Khz with high signal to noise ratio. It has an Audio Serial Data Bus that Supports I2S, Left/Right Justified, DSP, and TDM Modes with 16, 20, 24 and 32 bit output data formats. The 66AK2G02 on the EVM is connected to the AIC3106 over an I2C interface which is used to configure the audio codec to provide input audio in either 16 or 32 bit format required by the biquad block on the DSP. The sampling rate is set to 44.1 kHz.

4.1.2 66AK2G02 McASP configuration

The 66AK2G02 digital audio input and output is provided by the McASP interface which is a general-purpose serial port. The McASP supports time-division multiplexed (TDM) stream, Inter-IC Sound (I2S) protocols, direct connection to analog to digital converters (ADC) and digital to analog converters (DAC). The McASP includes independent transmit and receive sections with separate master clocks, bit clocks, and frame syncs. Up to 16 serializers that can be individually enabled for either transmit or receive operation. The number of serializers supported on each McASP varies for device to device (see the device data sheet). Each McASP also includes a 256-byte Read FIFO and Write FIFO.

The McASP on the 66AK2G02 is configured in clock master mode with the input and output serializers configured for TDM format to read in the left and the right channel data in either 16 or 32 bit format. The transmit and receive clocks as well as the frame sync are configured to be generated using internal clocks. The EDMA3 controller is programmed to move input data from McASP FIFO to internal buffers on the receive side and from the internal L2 memory to McASP FIFO on the transmit side. The McASP generates transmit/receive DMA event (AXEVT/AREVT) requests when data is to be transferred to/from its serializer registers. The EDMA must service McASP FIFOs when AXEVT/AREVT is generated.

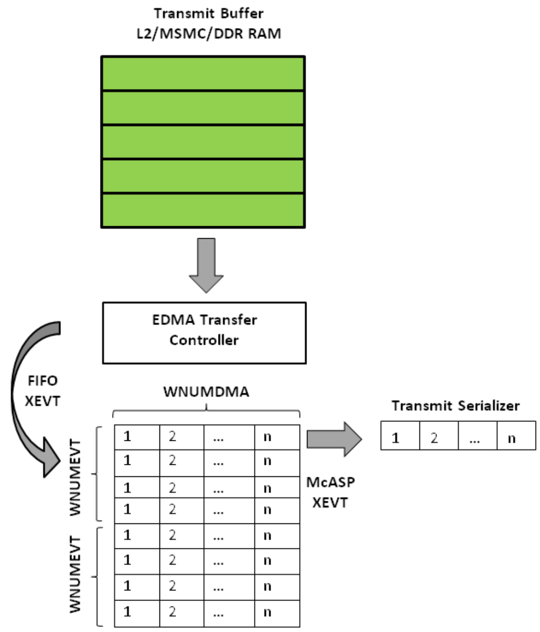


Figure 5: McASP FIFO Usage with EDMA

The software configures the McASP so that data can either be input in 16bit DSP Mode or in 32 bit I2S mode from the audio codec. For the 16 bit slot configuration in DSP mode on the audio codec, the McASP is configured for a single TDM slot of 32 bit width so that the data is received in packed format (LR) on the McASP. For the 32bit slot configuration in I2S mode on the audio codec, the McASP is configured for two TDM slots of width 32bit and the data is obtained as two channels data on different TDM slots. For both the mode, McASP generates Frame sync is internally. ACLKR and ACLKX generated internally using high-frequency clock (AHCLKX and AHCLKR) and clock dividers (CLKXDIV and CLKRDIV). The high-frequency clock generated internally from AUXCLK and clock dividers (HCLKXDIV and HCLKRDIV).

McASP Transmit Clock Generator **McASP Receive Clock Generator**

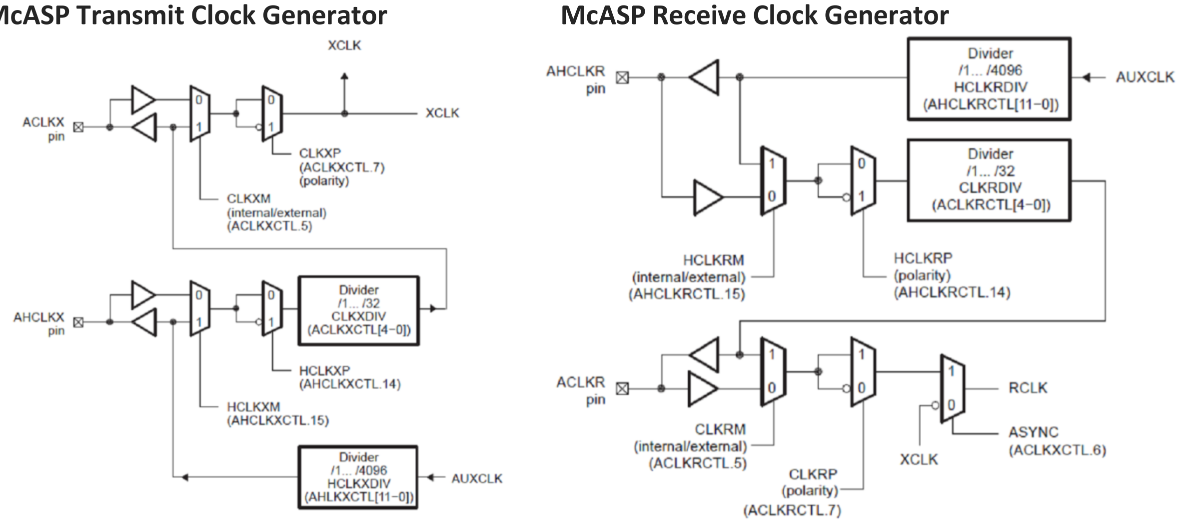


Figure 6: McASP transmit and Receive clock generator

4.2 Audio Processing Blocks:

4.2.1 Biquad Block

The biquad block consists of four digital biquad filters per channel organized in a cascade structure as shown in Figure 4−1. Each of these biquad filters has five 64-bit floating point coefficients.

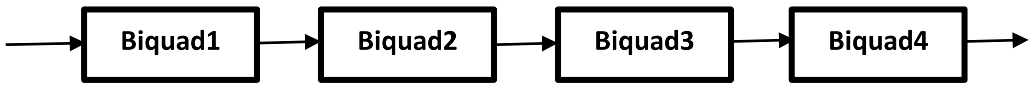


Figure 7: Cascaded Biquad Filters

The biquad structure that is used for the equalization filters are:

$$H(z) = \frac{b_0 + b_1 Z^{-1} + b_2 Z^{-2}}{1 + a_1 Z^{-1} + a_2 Z^{-2}}$$

The default filter coefficients for the each of the biquads are pre-computed but the demo, allows for updating the gain for each of the filter based on user inputs. Digital audio data coming into the device is processed by the biquad filters and then output from the device through the audio codec.

The TI Design supports coefficient calculation of following filter types: peak, high shelf, low shelf, high pass and low pass and notch filters. The filters used in this implementation are the high and low shelf filters to implement the 3 band equalization block. A shelving filter applies an equal gain change to all frequencies beyond a user-selected shelving frequency. The shelving filter is specified by specifying the shelf type (bass or treble), the corner frequency and the amount of cut or boost to be applied. The smooth slopes and the moderate phase shift of the shelving filters create sonically pleasing audio, allowing you to adjust the tonal qualities of audio in a musical way. The gain controls for these bands can be adjusted in range of 18 dB to –18 dB. In addition to this, there is an output gain control that can be used to ensure that the maximum gain through the system is less than 0dB. The filter updates are performed immediately following the filter specification. This means that there will be a gain discontinuity in the output produced by the different filter specifications.

4.2.2 High shelf filter implementation for Treble control

High Shelf filters cuts or boosts the frequency at the cutoff and all the frequencies higher than the cutoff frequency. These filters typically have only two parameters: the cutoff frequency and the gain. The TI Design uses a high shelf filter to manage the gain for all frequencies above 1000 Hz. The frequency response of the high shelf filter used to boost the high frequencies by 6db is shown below:

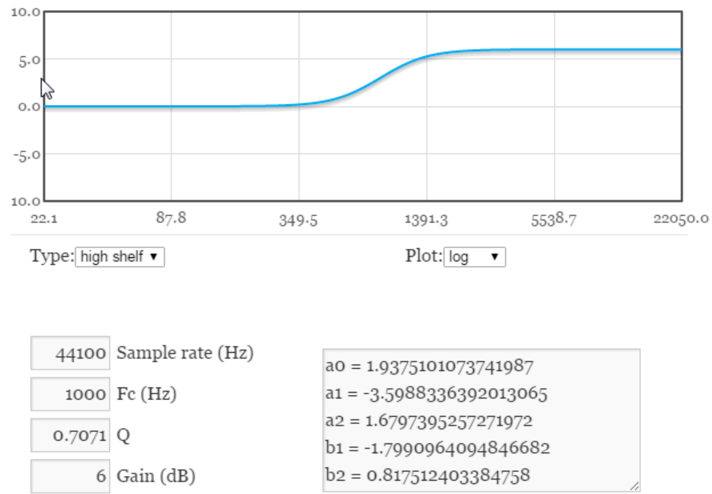


Figure 7: High Shelf Filter frequency response and coefficients

High shelf filters are usually used to attenuate or boost mid-high and high end of the spectrum that constitutes the treble. It is used to boost up a track by cutoff frequency of 1 kHz and higher. It can also be used to reduce the noise content in an audio track by cutting high frequencies. These controls can be adjusted throughout their entire range of 18 dB to –18 dB to boost or cut the treble.

4.2.3 Low shelf filter implementation for bass control

Lowshelf filters cuts or boosts the frequency at the cutoff and all the frequencies lower than the set cutoff point. It has only two parameters: the cutoff frequency and the gain. The TI Design uses a Low shelf filter to manage the gain/attenuation for all frequencies below 250 Hz. The frequency response of the low shelf filter used to boost the low frequencies by 6db is shown below:

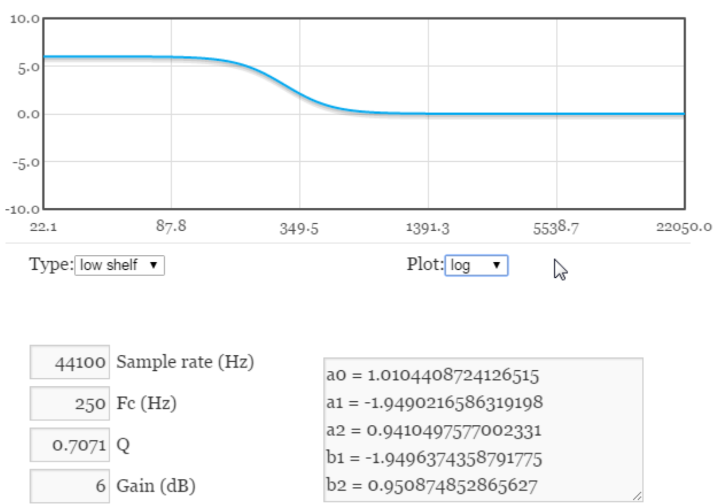


Figure 7: Low Shelf Filter frequency response and coefficients

Low shelf filters are usually used in the low-mid and low range of the audible spectrum that constitutes the bass content of the audio. It can be used to reduce some of the rumble noise caused by microphone stands and other low end sources or to boost the vocals of the artist which are typically at the lower end of the spectrum. These controls can be adjusted throughout their entire range of 18 dB to –18 dB to boost or cut the bass in the audio.

4.2.4 Mid band filter using Shelving filters:

The Design software is configured to use shelving filters instead of bandpass filters to cut and boost mid band frequencies between 250Hz and 1000Hz. The mid band gain and attenuation is applied by connecting a high shelf filter with cutoff frequency 250Hz and a low shelf filter with cutoff frequency at 1000Hz in cascade. The frequency response of the 2 filters is shown below:

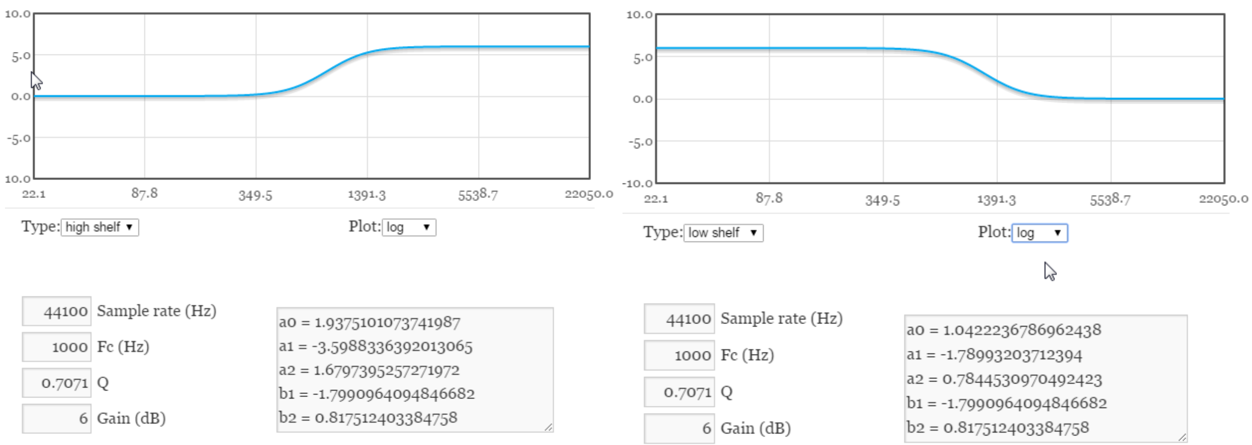


Figure 8: High and Low shelf Filter frequency response and coefficients for mid-band gain

These controls can be adjusted throughout their entire range of 18 dB to –18 dB to boost or cut the mid band frequencies in the input audio.

5 Performance

The following characterization data was measured using the 66AK2G02 with L2 configured as all RAM, 600 MHz CPU clock, and 800 MB/sec DDR3. The AIC was configured at a sampling rate of 44.1 kHz to input stereo audio data.

Input Sample size	Program (KB)	Data (KB)	Per Channel Data/Mcycles	MIPS (MHz)
32 bit	170	400	894	24.66
16 bit	170	400	932	23.64

Table 1: CPU and Memory Utilization

6 Getting Started

6.1 Equipment

You will need the following items in order to run the demonstration:

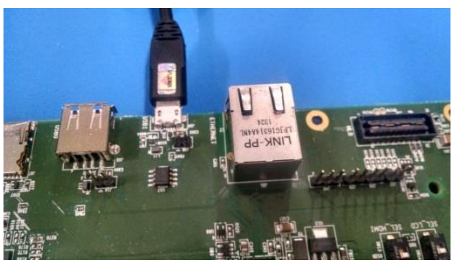
- TI 66AK2G02 General Purpose EVM kit
- Stereo input cable connected to Audio Source
- Audio Listening Device (speaker or headphones)
- PC with USB serial port and terminal program (such as Tera Term).

6.2 Getting Started

Please refer to the following instructions in order to setup the audio equalizer software on the 66AK2G02 General Purpose EVM:

6.2.1 Setting up hardware:

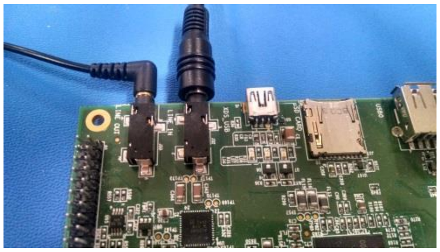
Step1: Connect the micro USB cable to the on board emulator USB1 on the 66AK2G02 General Purpose EVM and to a USB port of a host PC with CCS v6.1.2 connected.



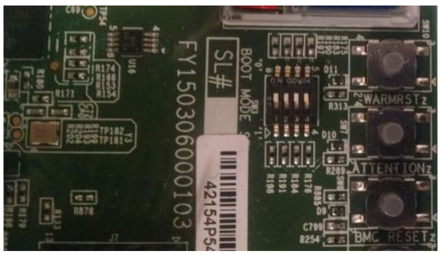
Step2: Connect mini USB cable to the USB to SOC UART0 connector and connect the cable to the host PC. On the host machine(PC), open a serial terminal using Teraterm or Minicom and set the baud rate to 115200

Note: Connection will show up on host as a Silicon Labs Dual USB to UART port. Connect to the SoC port connection specified in the EVM manual or Quick Start guide.

Step3: Connect a stereo input cable to the Line In connector and connect it to an audio source like a PC or a phone or any other media interface. Also, connect a 2.1 speaker or a headphone to the Line out jack



Step4: Set boot switches to “No boot mode” and connect a 12V supply to the power jack and power up the 66AK2G02 GP EVM.



6.2.2 Installing and setting up software

Step1: Download Processor SDK RTOS 2.x.x for K2G platform from ti.com

Step2: Install Codec composer Studio IDE (CCSv6.x) compatible with the version of the SDK.

http://processors.wiki.ti.com/index.php/Processor_SDK_RTOS_Getting_Started_Guide#Setup_software

Step3: Setup Processor SDK RTOS and CCS as described here:

http://processors.wiki.ti.com/index.php/Processor_SDK_RTOS_Getting_Started_Guide#Setting_up_CCS_for_EVM_and_Processor-SDK_RTOS

Step4: Download the ti-design-sw from the external GIT using

git clone git@git.ti.com:ti-design-sw/ti-design-sw.git audio_equalization

Note: This step only applies for users of Processor SDK RTOS 2.0.2. If you are using later version of the SDK, the software can be located under directory path processor_sdk_rtos_x_xx_xx_xx/ti-design-sw

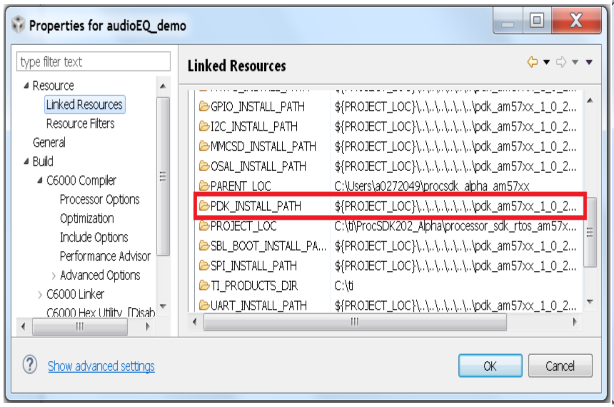
6.2.3 Building the software

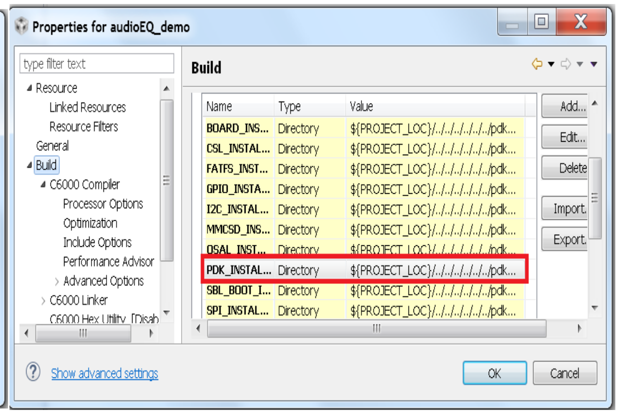
The Design software can be built either in TI Code composer Studio (CCS) and also using make files. Both the procedures are described below:

Using CCS :

Step1: Import the CCS project audioEQ_demo from the folder path audio_equalization\evmK2G\build\ccs

Step2: Set the path to PDK_INSTALL_PATH in the CCS environment in the Project Properties under Resources->Linked Resources and in Build –>Variables as shown below:





Note: This step only applies for users of Processor SDK RTOS 2.0.2. If you are using later version of the SDK, the software can be located under directory path processor_sdk_rtos_x_xx_xx_xx/ti-design-sw

Step3: Build the audioEQ_demo project in your work space to generate the application binary audioEQ_demo.out

Using Make:

Step 1: Setup the processor SDK RTOS build environment by following the steps specified here:
http://processors.wiki.ti.com/index.php/Processor_SDK_RTOS_Building_The_SDK#Setup_Environment

Step2: Change directory to root directory of the design software and invoke make

For Windows:
gmake clean
gmake all

For Linux Environment:
make clean
make all

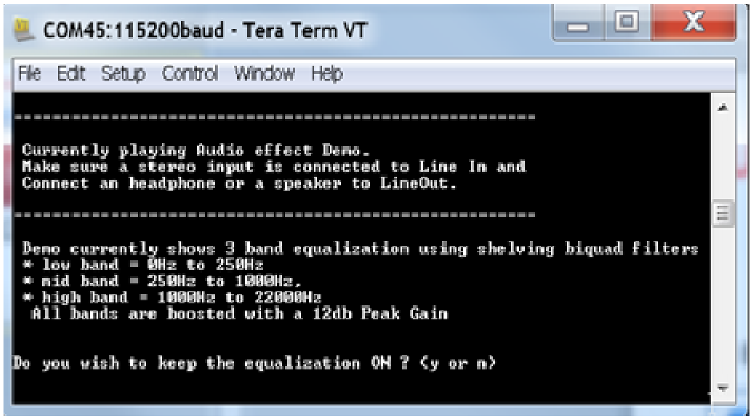
6.2.4 Running the software

The design software can be run using the CCS environment or can be booted on the EVM using SD card or a flash image. Make sure that you have setup the EVM as described in the Setting Up hardware section of the design guide.

Using CCS

Step1: Create a Target configuration file for the GP EVM as described in http://processors.wiki.ti.com/index.php/Processor_SDK_RTOS_Getting_Started_Guide#Setup_EVM_hardware

Step 2: Launch the target configuration file and connect to the DSP
Step 3: Go to Menu option Run->Load Program and browse to the audioEQ_demo.out file built in previous section of the design guide.
Step 4: When you run the demo, you will hear the audio playback from the source boosted by 12 db and the following User console will observed on the UART serial terminal



Step 5: Users can use the UART based user interface to disable the equalizer to hear the audio without the equalization or choose to adjust the gains on the 3 bands used in the equalizer.

6.2.5 Customizing the Design Software

Modifying the Equalizer bands: The frequency bands used in the equalizer are defined by using the cutoff frequency of the biquad filters that are defined in the file audioEQ_main.c. Users can choose to modify the cutoff frequencies and default gain that have been defined in the file. They can also choose to introduce more frequency bands by defining more biquads. However this will also require developers to modify the audio_Echo_Task function defined in the file audioEQ_filtering.c

Modifying the type of filters: The reference design uses shelving biquad filters to boost and attenuate the frequency bands. Users can modify the type of filter used to pass filters or notch filters to test the performance on the device. They are also free to use their own coefficients from biquad filters that they may have designed using Matlab or other filter design software.

Modifying the input data format and sample rate: Developers can choose to get the input inform of 16, 24 and 32 bit format. This can be modified by changing the slot length in the AIC_config (AIC31.c) and the McASP channel params. It will also require changing receive and transmit format in the mcasp2XmtSetup and mcasp2RcvSetup structures defined in the file audioEQ_filtering.c. The sampling rate for the input analog data is defined in the AIC31_config structure.

Modifying the processing block and user interface: The software for the UART based user interface is defined in the function ReadInputTask in audioEQ_main.c and the processing block is applied to the audio loopback in the function audio_Echo_Task defined in the file audioEQ_filtering.c. Developers can insert their own processing blocks in the code to test their audio processing algorithms and create custom user interface to test it.

7 Design Files

7.1 Schematics

The design files for 66AK2G02 General Purpose EVM may be found at http://www.ti.com/tool/TIDEP0069

8 Software Files

To download the software files for this reference design, please see the link at

- **ProcessorSDK_RTOS_K2G**
- **External GIT**: http://git.ti.com/ti-design-sw (Separate download only for Processor SDK RTOS 2.0.2)

9 References

1. Texas Instruments E2E Community, http://e2e.ti.com/

10 About the Author

Rahul Prabhu is a Senior Software Applications Engineer in TI's Embedded Processing organization supporting ARM-based and the DSP -based SoCs such as AM57xx, 66AK2EXX, OMAPL138 and C667x devices. Rahul brings to this role his extensive experiences and knowledge in signal processing, optimizing algorithms, system integration and application development. Rahul earned M.S in Electrical and Computer Engineering from University of Houston.