TI Designs: TIDEP-0077

Audio Pre-processing System Reference Design for Voice-Based Applications Using C5517

Description

The TIDEP-0077 uses multiple microphones (two to four), beamforming, and other algorithms to clean up and extract clear speech from an environment containing noise sources. The rapid increase in voice-activated applications has created a demand for systems that can extract clear voice from noisy environments. These systems are especially important in applications that have voice triggering and speech recognition. This design guide walks through running a demonstration on the TMDSEVM5517 device using a linear microphone board (LMB) and also discusses the various concepts used to clean up audio.

Resources

TIDEP-0077	Design Folder
TIDA-01470 (LMB)	Design Folder
PCM1864	Product Folder
TMDSEVM5517	Tools Folder
SPRC133	Tools Folder
TELECOMLIB	Tools Folder

TI E2E™ Community

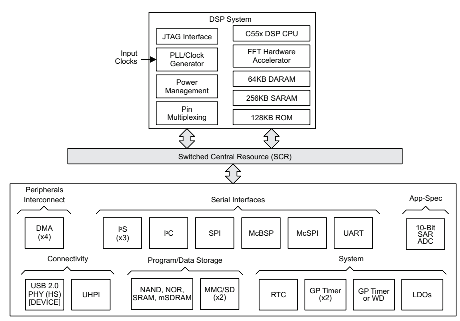
ASK Our E2E Experts

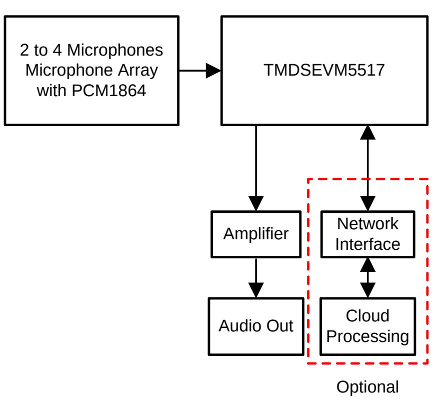
Features

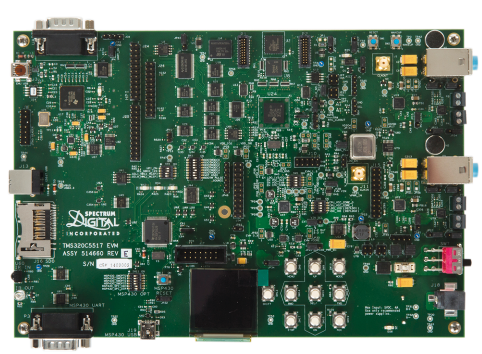
- Uses Single Digital Signal Processor (DSP) and Microphone Array to Extract Clear Speech from a Noisy Environment
- Eliminates Background Noise and Clutter From Audio Source
- Enables Better Voice Recognition by Presenting Clean Speech Audio to the Recognition Engine.
- Offers a Complete System Reference Design Using TI-Provided Software, Evaluation Module, and Microphone Array

Applications

- Interface-to-Cloud-Based Voice Recognition for Voice-Activated Digital Assistant Applications
- Interface-to-Cloud-Based Voice Recognition for Smart Home Applications
- Local (Limited Dictionary) Voice Recognition for Voice-Based Appliances Control
- Voice and Speech Applications (Such as Video Conferencing)







An IMPORTANT NOTICE at the end of this TI reference design addresses authorized use, intellectual property matters and other important disclaimers and information.

1 System Description

The TIDEP-0077 uses TI hardware and sophisticated, field-proven software algorithms to obtain clear speech and audio from noisy environments. The ability to extract clear speech or audio from a noisy environment is important to many applications that use voice-activation, such as digital assistants, telephone and video conferencing, and other high-quality speech systems. Typical sources of sound clutter are undesired background noise sources and so forth.

This TI Design uses a beamforming algorithm to form a virtual-directional microphone that points at the direction of the speaker or the desired audio source and then amplifies the speech signal from the desired direction, which attenuates all signals from all other directions. In addition to beamforming, TI offers a set of audio algorithms that may further improve the quality of sound.

Section 5 summarizes the theory of the beamforming, which uses multiple microphones, an associated adaptive spectral noise reduction (ASNR) filter, and the multiple source selection (MSS) algorithm to obtain the virtual-directional microphone signal.

The interface between the microphone array and the processor must support streaming of multiple data inputs. The data rate depends on the application requirements. The TIDEP-0077 streams two to four microphones mounted on a linear microphone array, samples in 16-bits at 16000 samples per second, and uses the analog-to-digital converter (ADC) PCM1864 for inter-IC sound (I2S) interface to the evaluation module (EVM) board.

The EVM supports multiple audio output venues. The audio data can be processed locally or sent out through one of the TMDSEVM5517 external ports. An application that uses local processing, such as voice-recognition remote-control appliances, can process the data locally. The TIDEP-0077 loops the clean audio back into the left channel of the stereo audio output interface from the onboard AIC3204 audio codec. The reference microphone (one of the microphones in the circular microphone array) plays out of the right channel. This setup enables the user to compare the quality of processed and unprocessed audio.

This TI Design includes full source code that can be modified to support various applications.

For an optional cloud-based, voice-activated digital assistant design, the output signal can be sent to a network interface device using external interface, such as UART, SPI, or USB. The return audio signal from the network can be sent to the device codec to be played by a speaker. The C5517 DSP has a total of three I2S lines of which two are used for four microphone inputs. The third I2S line can be used to pipe the audio out of the system to a secondary system if desired.

Local (limited dictionary) voice recognition for voice-activated digital assistant applications could use the DSP to do voice recognition. The DSP in the TMDSEVM5517 is a high-performance, fixed-point DSP clocked in 200 MHZ that supports up to 8MB of external memory in addition to 320KB of internal memory. The DSP has enough power and memory to support voice recognition of a limited dictionary.

Conference call and other speech-processing applications require additional features (mixing of signals, better acoustic echo cancellation, and so on). As stated above, the DSP in the EVM has enough power and memory to process limited speech algorithms. Note that TI audio libraries include optimized audio algorithms that can be used by speech applications.

1.1 Key System Specifications

Table 1 shows the key system specifications.

Table 1. Key System Specifications

COMPONENT	DESCRIPTION	DETAILS
Linear microphone array	Two to four microphones out of the LMB can be used for the C5517.	Section 2.2.1
PCM1864	PCM1864 audio ADC provides interfaces to the EVM. Each PCM1864 supports up to four audio microphones. Systems with more than four microphones require multiple PCM1864s.	Section 2.2.2
TMDSEVM5517 DSP EVM	Evaluation board based on the C55x DSP	Section 2.2.3
Chip support library (CSL)	Standard TI software release for the C55x family	Section 2.2.4
Executable BF_rt_bios	DSP executable code that processes multiple microphones streaming audio and generates a virtual-directional microphone audio stream	Section 2.2.5
Application source code and Code Composer Studio™ (CCS) projects	Source code for the data path unit test and for the applications that enables the user to modify or rebuild the code	—
TI audio libraries (or TELECOMLIB)	TI-optimized audio processing AEC-AER and VOLIB libraries	Section 2.2.7
CCS version 6.1.3. CCS v6.2 and v7 are not supported at this time.	TI-integrated development environment (IDE) that is used to run the executables and can be used to build the executables. The project was built and tested with CCS version 6.1.3 and code generation tools CGT for 5500 version 4.4.1 or higher (It is assumed that the user is familiar with CCS.)	—
TI tools and utilities	A set of tools and utilities that can be downloaded from ti.com.	Section 2.2.8

2 System Overview

2.1 Block Diagram

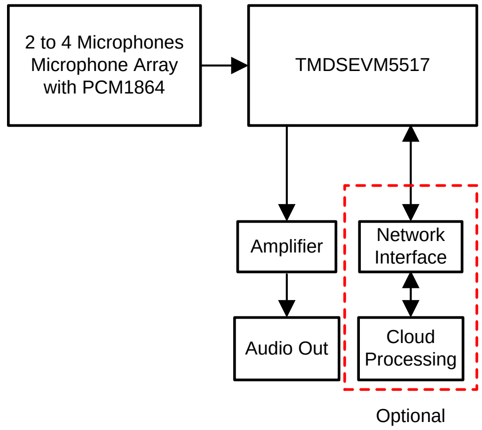


Figure 1. TIDEP-0077 Block Diagram

2.2 Highlighted Products

2.2.1 Linear Microphone Array

Four microphones are mounted equidistant at a linear geometry on the microphone board. The PCM1864 samples the microphones and streams the digital values using I2S interfaces to the TMDSEVM5517. Building a generic microphone array and calculating the filter coefficients associated with the linear array is described later in this document.

2.2.2 PCM1864

The PCM1864 is a 103-dB, two stereo channel (four channels total), SW-controlled audio ADC with universal front end.

See ti.com's PCM1864 product folder for a full description of this device.

2.2.3 TMDSEVM5517

The TMDSEVM5517 EVM is based on the C5517 processor.

For a full description of the TMDSEVM5517, see ti.com's TMDSEVM5517 tools folder.

Figure 2 shows a block diagram of the TMDSEVM5517.

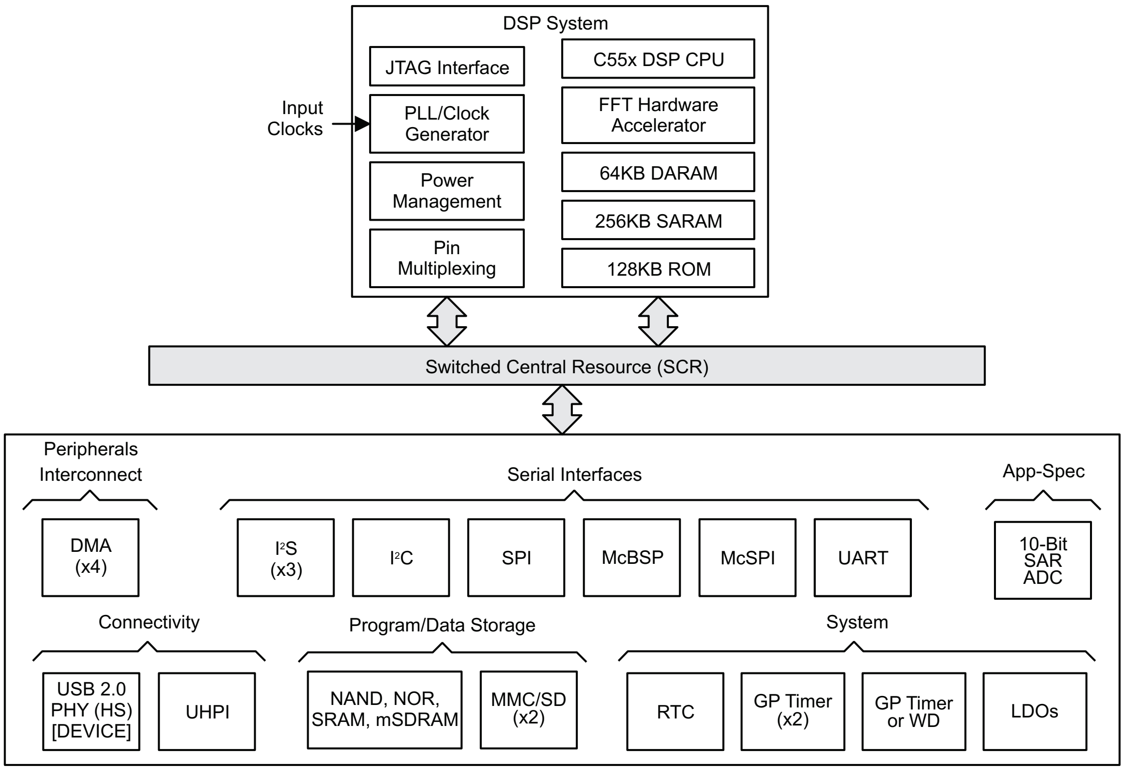


Figure 2. TMDSEVM5517 Block Diagram

2.2.4 CSL C55xx_csl

The CSL library contains utilities and drivers that are used to configure control and use all the peripherals and IP that are part of the C5517 chip as well as the TMDSEVM5517 peripherals.

Free download of CSL is available at ti.com's TMS320C55x CSL tools folder. The version of C55xx_csl should be 3.07.00 or higher.

2.2.5 BF_rt_bios Project

The BF_rt_bios project is part of the C55xx_csl release. The project gets two or four audio streams from the microphones array and applies beamforming, ASNR, and MSS to obtain a single, virtual-directional microphone directed at speech and to clean out clutter from a noisy environment. Using the C5517's onboard headphone jack, the demonstration is set up so the processed audio is output through the left channel of the stereo output and the unprocessed audio on the right channel. This setup enables the user to analyze the processed and unprocessed audio independently.

The Wiki page *C55x CSL Audio Pre-Processing*[5] provides the most updated details on the project including how to build, run, and test the results.

2.2.6 Application Source Code and CCS Projects

The C55xx_csl release contains a CCS project and all the source code that is required to build the BF_rt_bios project. Instructions how to build the project are given in Section 4.1.

2.2.7 TI Audio Libraries

TI audio libraries (TELECOMLIB) consists of two optimized libraries that are used in this reference design: the Acoustic Echo Cancellation-Removal (AEC-AER) library and the Voice Library (VOLIB). In addition, there is a DSPLIB package available for C55x devices, which contains many signal processing optimized algorithms.

AEC-AER and VOLIB can be downloaded from ti.com's TELECOM tools folder. The user must install the audio libraries in the same directory as the c55_csl_3.07 was installed. The libraries version that are used are the following:

- AER LIB for C55X CPU version 3.3, version 17.00.00.00, or higher
- The VOLIB for C55X CPU version 3.3, version 2.01.00.01, or higher

2.2.8 Set of Tools and Utilities

Table 2 lists the set of TI tools and utilities that are required for building the BF_rt_bios project.

Table 2. TI Tools and Utilities

TOOLS AND UTILITY NAME	LOAD LOCATION
DSP BIOS version 5.42.2.10	DSPBIOS 5.42.2.10
XDAIS version 7.24.0.4	XDAIS 7.24.0.4
XDC TOOLS version 3.24.05.48	XDC TOOLS

NOTE: The user must install AEC-AER and VOLIB libraries as well as the tools from Table 2 in the same directory as the C55xx_csl was installed.

NOTE: For a complete set of version requirements, see the list of dependencies in the release notes for C55X_CSL.

3 Getting Started Hardware and Software

3.1 Hardware

3.1.1 TMDSEVM5517 Hardware Setup

Detailed steps how to set up the TMDSEVM5517 are given in the *C5517 EVM Quick Start Guide*.

Additional information about the TMDSEVM5517 is available at C5517 Evaluation Module and in the *C5517 General Purpose EVM User Guide*[3].

Figure 3 shows the C5517 layout and key components.

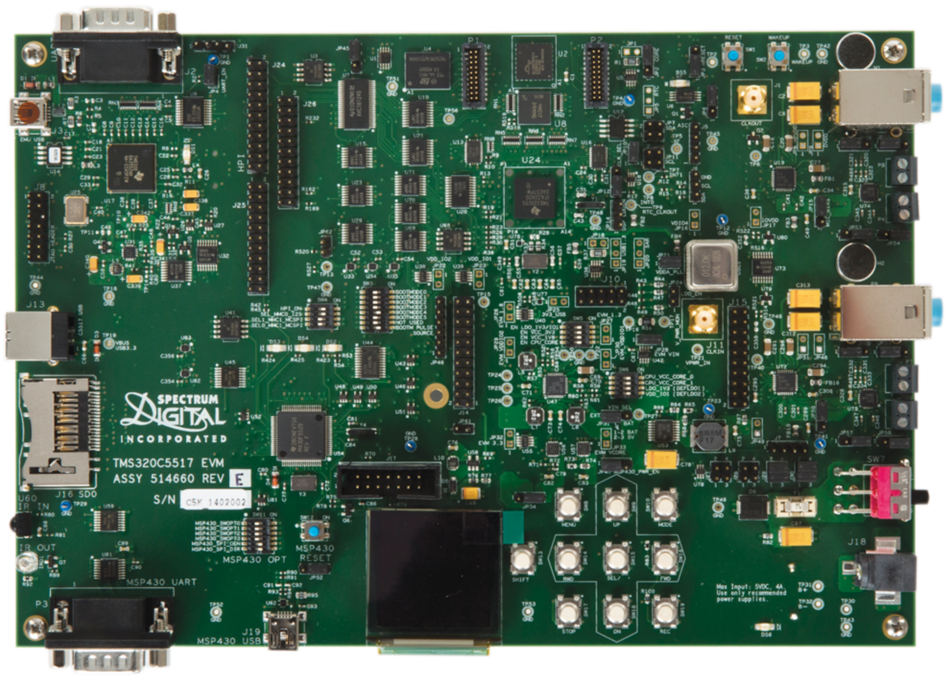


Figure 3. TMDSEVM5517

3.1.2 Connecting the Linear Microphone Array to TMDSEVM5517

The linear microphone array gets power and ground from the TMDSEVM5517. The PCM1864 can support two channels (equivalent to a single stereo channel) or four channels (equivalent to two stereo channels). Table 3 describes the signals that linear microphone array connects to the TMDSEVM5517. The third column in Table 3 shows the microphone array connection to the LMB.

Table 3. LMB Microphone Signals

SIGNAL NAME	TMDSEVM5517	LMB PIN
3.3 V	J10_Pin9	LMB_3.3v
Ground	J10_Pin5	LMB_GND
I2C SCL	J14_Pin16	LMB_SCL
I2C SDA	J14_Pin20	LMB_SDA
Bit clock (microphone one and microphone two)	J27_Pin3 (no jumper)	LMB_BCLK
Frame clock (microphone one and microphone two)	J27_Pin4 (no jumper)	LMB_LRCLK
Data one	J30_Pin2 (no jumper)	LMB_DATA1
	J29_Pin1_Pin3 (jumper on)	
	J29_Pin2_Pin4 (jumper on)	
	J30_Pin1_Pin3 (jumper on)	
Bit clock (microphone three and microphone four)	J31_Pin3	I2S_BCLK
Frame clock (microphone three and microphone four)	J31_Pin2	I2S_LRCLK
Data three	J31_Pin1	LMB_DATA3
	UART_EN (no jumper)	

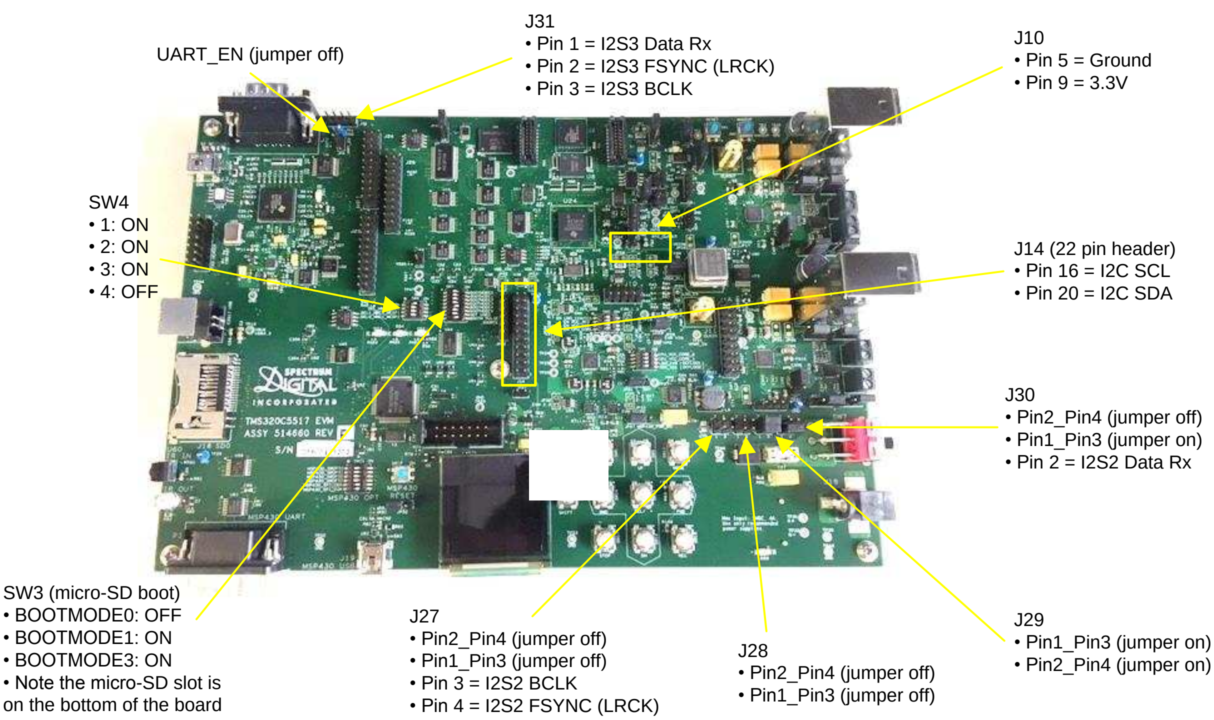


Figure 4. Location of Jumpers and Headers

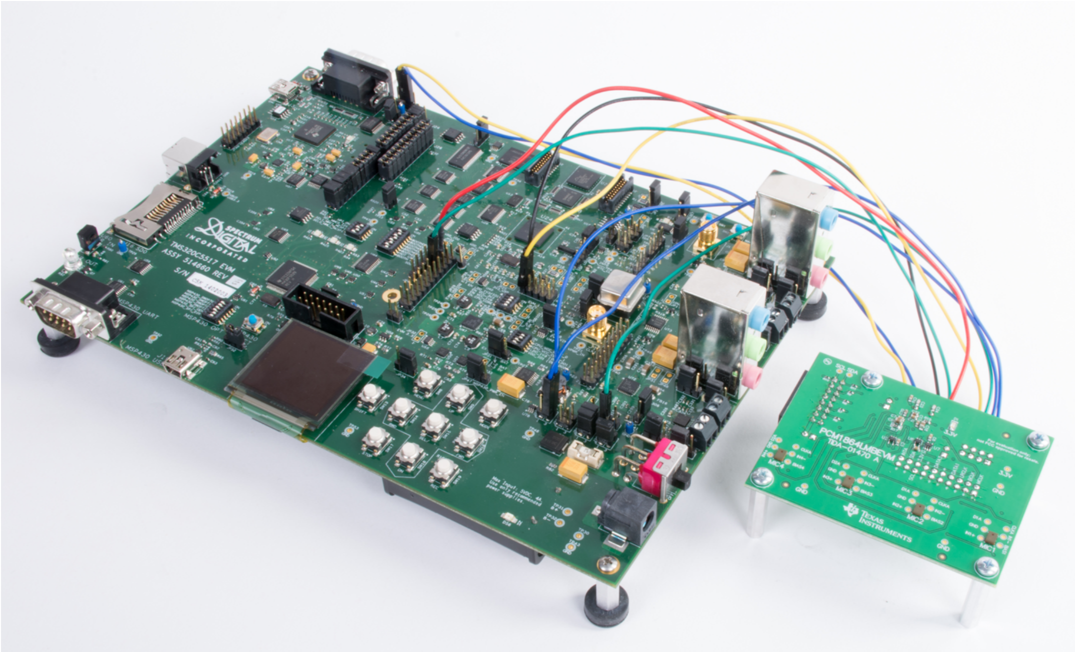


Figure 5. Connecting the LMB to TMDSEVM5517

3.1.3 Connect Headphone

A stereo headset should be connected to the output audio. The audio connector is the green connector of P9. P9 is located next to the power switch on the right side of the EVM. See the P9 green-out audio connection to TMDSEVM5517 in Figure 6.

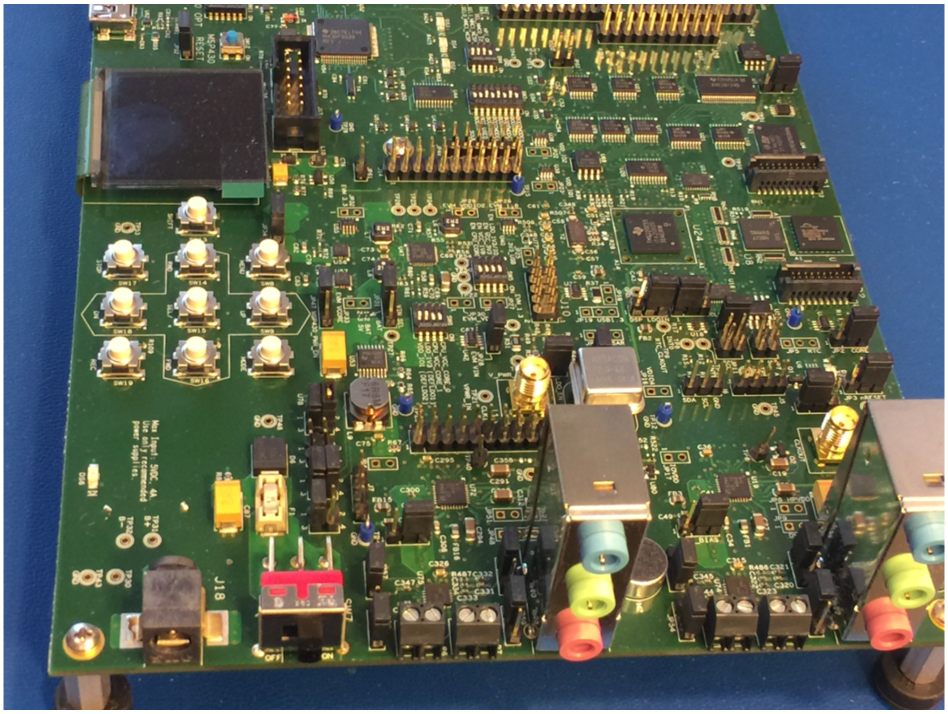


Figure 6. P9 Green-Out Audio Connection to TMDSEVM5517

3.2 Install Software

3.2.1 Install C55xx_SDK and the Audio Libraries

The latest release of C55xx_CSL can be downloaded from from the C55xx_csl tools folder on TI.com. The C55xx_csl version is 3.08.00 or newer. Install it anywhere on the system. The location where the directory c55_lp was installed is the main directory of the project. All other software must be installed in the directory parallel to c55_csl_3.08 directory.

The source code for the default CSL package audio preprocessing demo has a bug that requires a patch. Audio frequencies above 3 KHz are not output on the EVM5517 P9 headphone out. Therefore, the audio output quality is degraded.

The problem is an incorrect configuration of the DAC on the EVM5517 device. Patch the files with: C5517_TIDEP-0077_VoiceProcessing_freqcutoffpatch3.zip, which is available for download from: C5517 Audio Preprocessing TI Design Patch Files: 2) Applicable only to CSL v3.08.

The two files that require patching are:

- *C:\ti\c55_lp\c55_csl_3.08\demos\audio-preprocessing\c5517\codec_pcm186x.h*
- *C:\ti\c55_lp\c55_csl_3.08\demos\audio-preprocessing\c5517\codec_aic3254.c*

3.2.2 Install DSP/BIOS, XDAIS, and XDC TOOLS

Release 5.42.2.10 of DSP/BIOS can be downloaded from ti.com's BIOS 5_42_02_10 folder. Install it in the same directory where c55_csl_3.07 is installed.

Release 7.24.0.4 of XDAIS can be downloaded from ti.com's XDAIS 7_24_00_04 Product Download Page folder. Install it in the same directory where c55_csl_3.07 is installed.

Release 3.24.5.48 of XDC Tools can be downloaded from ti.com's XDCtools3_24_05_48 Product Download Page folder. Install it in the same directory where c55_csl_3.07 is installed.

Figure 7 shows how the directory structure should look like once all the packages are correctly installed .

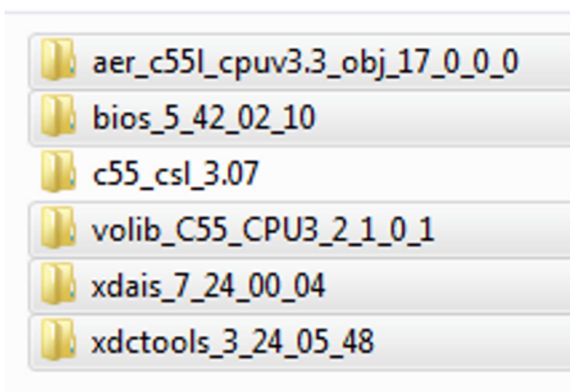
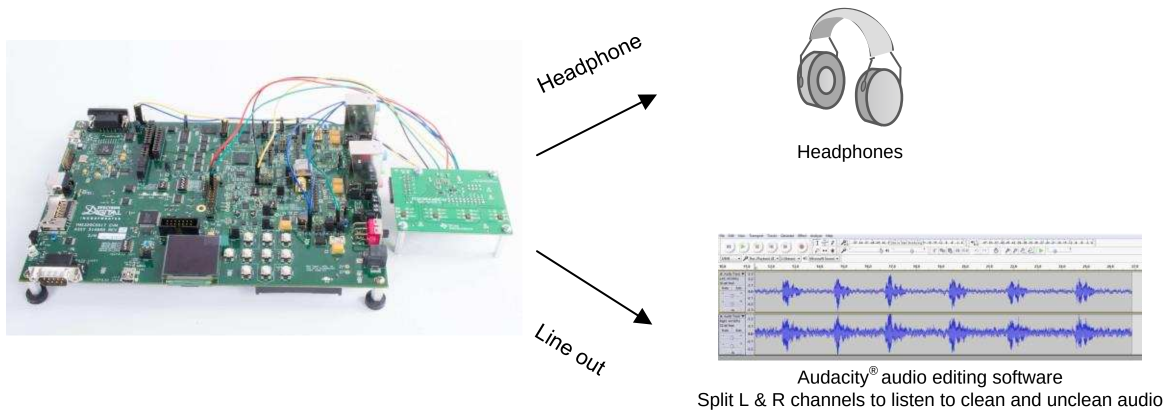


Figure 7. Installed Directory Structure Overview

4 Testing

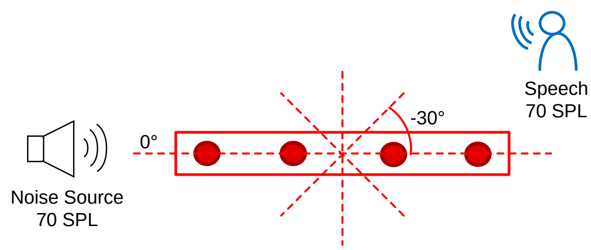
There are two ways to test the demonstration. The first is to listen to the audio through the headphones to access the delta between the processed and unprocessed audio. Another method is to connect the headphone out from the C5517 EVM from line-in to a PC. Using a program such as Audacity®, the audio can be recorded, split into left and right channels, and evaluated independently. Figure 8 illustrates the two mechanisms to evaluate the audio pre-processing demo on the C5517 EVM.



Note: C5517 can only use a maximum of six microphones.

Figure 8. Overview on Evaluating Audio Output of Voice Pre-processing Demonstration on C5517 EVM

Figure 9 describes the test environment.



Test Environment

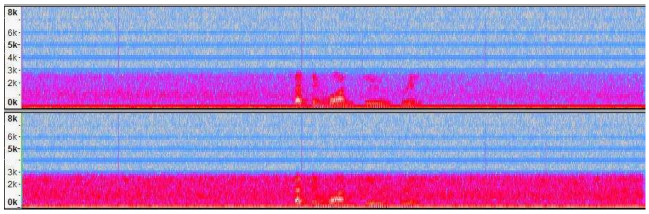
Office Room

White Noise 0° at 70 SPL

Speech –30° at 70 SPL

Figure 9. Test Environment

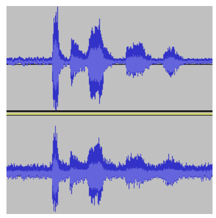
Figure 10 and Figure 11 show the test results of running the demonstration and capturing clean and unclean audio from line in to a PC with Audacity. The left channel is clean and the right unclean.



Clean

Unclean

Figure 10. Test Results: Spectrogram



Clean

Unclean

Figure 11. Test Results: Waveform

Refer to the training video at *Demonstrating Voice Preprocessing on the C5517*[6] for more details. Details about the test procedure are given in the wiki page *C55x CSL Audio Pre-Processing*[5].

4.1 Build and Run the Executable

The instructions below assume that the C55x CSL v3.07 was installed at C:\ti in a Windows® environment, and the relevant project files replaced as described in section 3.2 to support the LMB. CCS v6.1.3 was used. CCS usage is not in the scope of this document. The following are links to CCS training material:

- How to setup CCS target configurations at *Target Configuration - Custom Configurations* wiki page
- CCS training at *Category:CCSv6 Training*
- How to run the BF_rt_bios project at *Demonstrating Voice Preprocessing on the C5517*[6]

In order to run the demonstration, the BF_rt_bios, atafs_bios_drv_lib, and C55XXCSL_LP CCS projects must be imported into the CCS workspace. The following are steps to get started:

1. Launch CCS and create a workspace for the audio pre-processing demonstration.
2. Go to *Project → Import CCS Projects*.
3. Click the *Browse* button, and navigate to the CSL package at C:\ti\c55_lp\c55_csl_3.07.

4. Select the projects as seen in Figure 12. The workspace after importing will look like Figure 13.

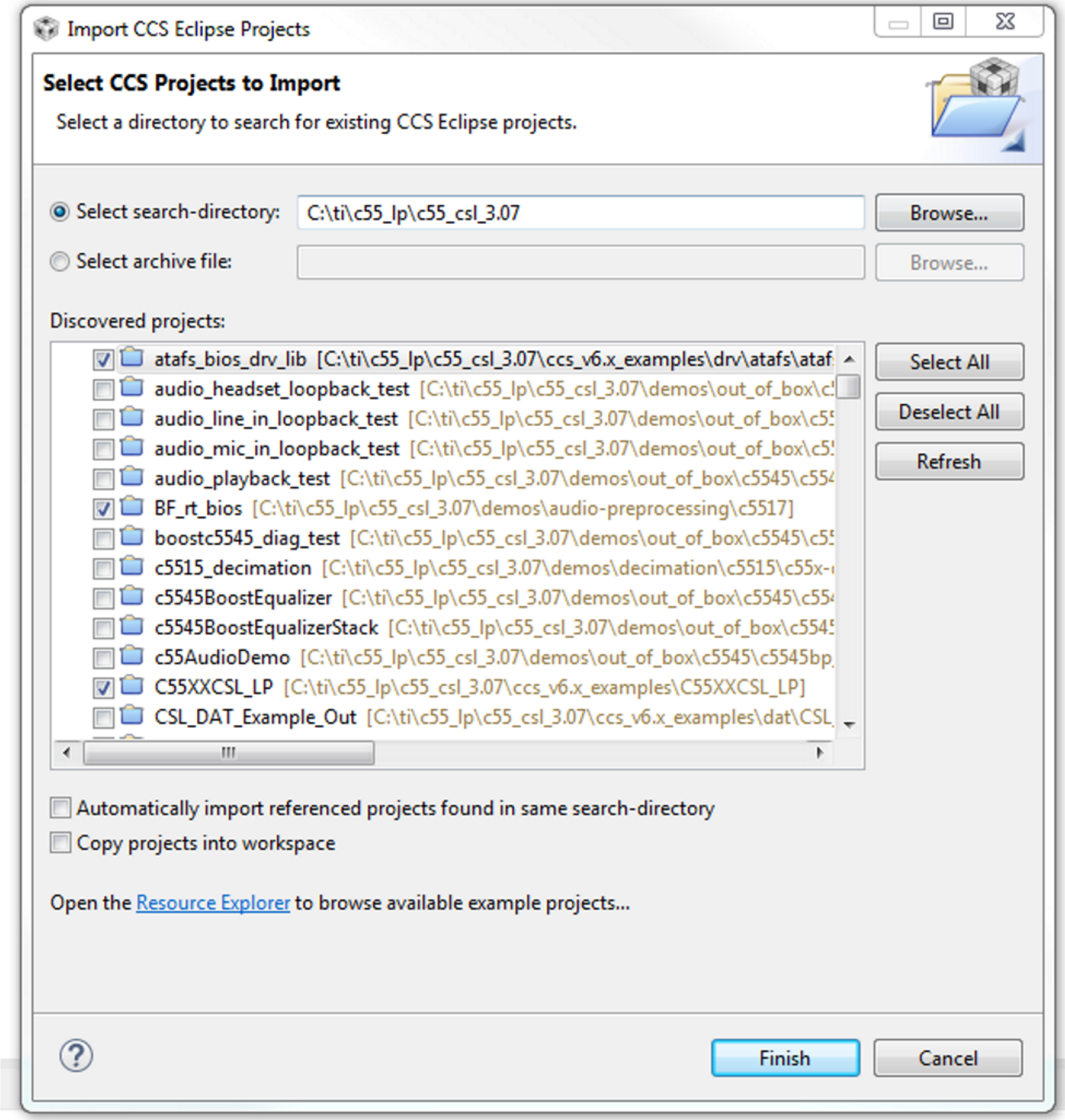


Figure 12. Projects to be Imported Into CCS Workspace

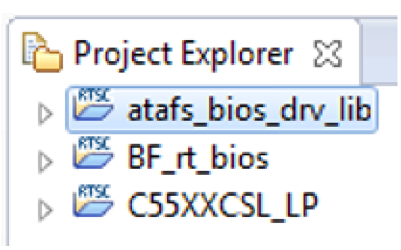


Figure 13. CCS Workspace After Importing Projects

5. BF_rt_bios is the project that contains the example for the C5517 voice pre-processing. Prior to building BF_rt_bios, ensure C55XX_LP is built with the correct *#define CHIP_C5517* macro defined in C:\ti\c55_lp\c55_csl_3.07\inc\ csl_general.h. This is to ensure the CSL library has the correct C55x platform definition.
6. Build BF_rt_bios by right-clicking on the project and *Build Project*.
7. Once the build completes, the binary to load on the C5517 would be located at C:\ti\c55_lp\c55_csl_3.07\demos\audio-preprocessing\c5517\Debug\BF_rt_bios.out.
8. Launch the C5517 target configuration.
9. Connect to the DSP core with a GEL file initialized on the core.
10. Load BF_rt_bios.out, and hit the resume button in CCS.
11. The demonstration will now continuously run. The audio captured on the LMB microphones is sent to the C5517 DSP. Assuming the headphones are connected to the C5517 EVM (as covered in Section 3.1.3), the processed audio will be output on the left earphone, and the unprocessed audio output on the right earphone.

4.1.1 Changing the Number of Microphones

Two flags in the file codec_pcm186x.h (location c55xx_csl\demos\audio-preprocessing\c5517) control the number of microphones. The user should un-comment one of the following two lines: *#define NUM_OF_MICS 2* or *#define NUM_OF_MICS 4*. In addition, un-commenting the line *#define LOOPBACK_ONLY* bypasses the beamforming. Loopback can be used for debugging the linear microphone array and verify that all microphones are working.

4.1.2 Changing the Filter Coefficients

The beamforming filter coefficients depend on the geometry of the microphone array. The filter coefficients in the this project were calculated based on a four microphone geometry of the LMB board. Should the user wish to use a different microphone array of a different geometry, new filter coefficients are required. Section 4.1.2.1 describes how to calculate a new set of filter coefficients for the geometry of the microphone array. The new filters' coefficients buffers are updated and the project should be rebuilt. The file sysbffilt.c has the values of the filters. The file sysbffilt.h is the include file associated with the filters. Both files are located in c55xx_csl\demos\audio-preprocessing\common subdirectory.

4.1.2.1 Calculating Filter Coefficients

The beamforming filter coefficients depend on the geometry of the microphone array and the angle of the direction of the source with respect to the microphone array. bfgui.exe is a tool to generate beamforming filter coefficients and is part of the AER library in directory *aer_c55l_cpuv3.3_obj_17_0_0_0\tools\bf_tool*. A user's guide for the beamforming design tool bfgui.pdf is in the same directory as well. The user is strongly encouraged to read bfgui.pdf because it gives insight into the general theory of beamforming.

Upon starting bfgui.exe, the user should configure the following values, as shown in Table 4.

Table 4. Beamforming Design Tool Values

VALUE	COMMENTS
Sampling rate (KHz)	This TI Design uses 16 (16000). The default value of the tool is 8000.
Number of microphones	Two or four microphones are used.
Microphone distance	Distance in centimeters between two adjacent microphones. Equal distance between any two microphones is assumed. For linear array, it is the linear distance, and for circular array, it is the linear distance of the chord between two microphones.
BF angle	The utility generates a set of filters for a single angle of arrival; that is, for a single virtual-directional microphone. For a system with multiple virtual-directional microphones like the one that is used in this TI Design, the utility must be called multiple times, each time for a different angle.
Geometry	Microphone array geometry: 0 for 1D linear, 1 for 2D linear, 2 for 2D rectangular, and 3 for circular. Using the LMB requires geometry be set to 0.
Contour levels	Required for graphical illustration. Leave as default.
Polar frequency	Required for graphical illustration. Leave as default.

The number of microphones on the array determines the uniqueness of the separation. The number of virtual-directional microphones determines the angle of separation. If the number of virtual microphones was set to twelve, then every 30° (360° divided by 12) is a virtual-directional microphone. There is a relationship between the number of microphones and the number of virtual-directional microphones. The processing load of the beamforming and the ASNR depends on the number of microphones in the array and linearly on the number of virtual-directional microphones. Benchmark results for typical C5517 DSP are given in .

Following the instructions in the user's guide (bfgui.pdf), configure the filter coefficients tool for linear geometry and four microphones with 2.125-cm equal distance between any two microphones. This filter is for 45° of arrival. Figure 14 shows the configuration of the filter generation tool. The filter coefficients are stored in filterCoeff.log.

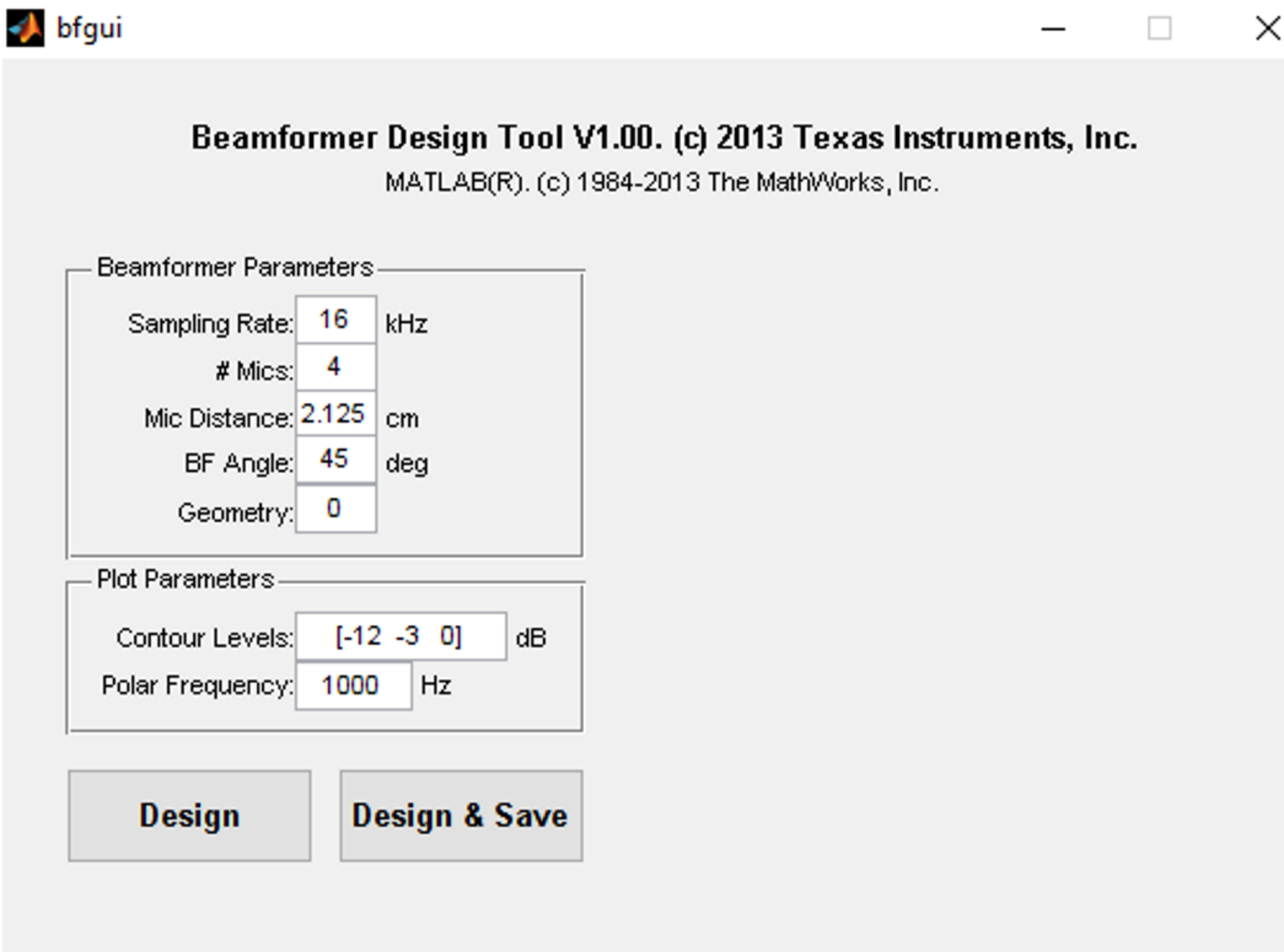


Figure 14. Configuration of Filter Generation Tool

Figure 15 illustrates a virtual microphone.

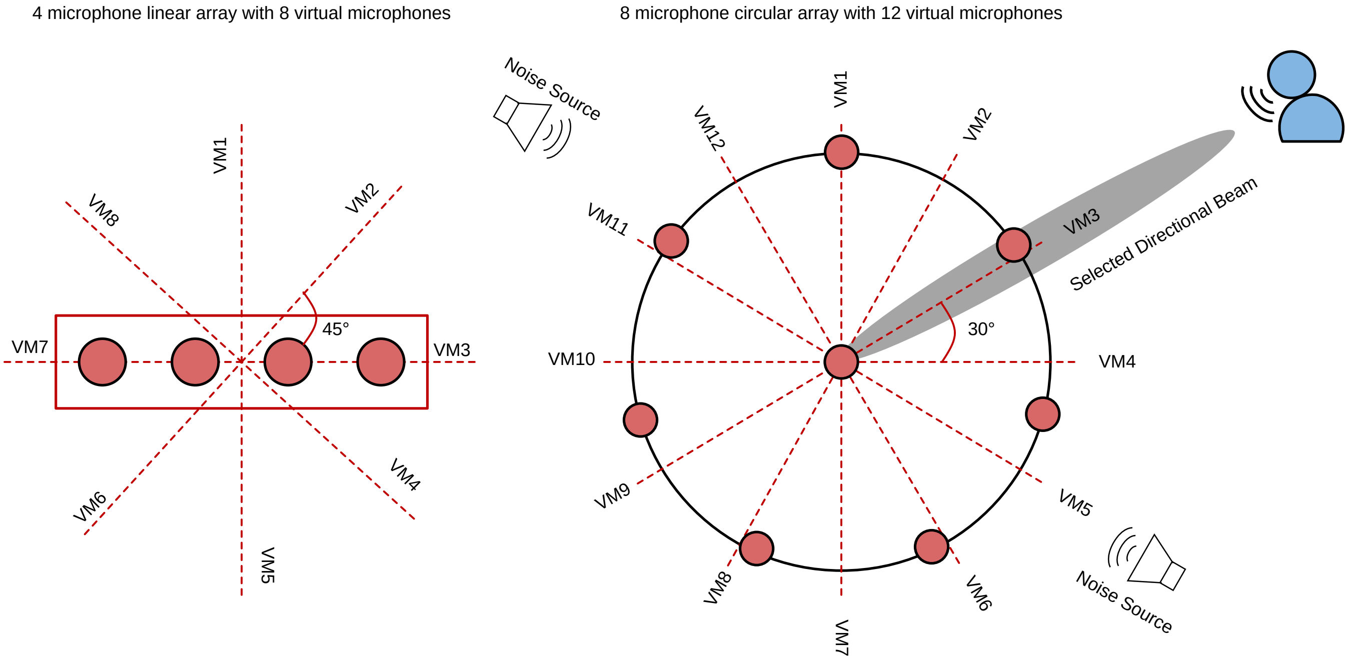


Figure 15. Description of a Virtual Microphone

4.1.3 Benchmarks

Table 5 shows the benchmarks attained by running the BF_rt_bios demo on the C5517 EVM.

Table 5. Benchmarks

	C5517 EVM	C5517 EVM
MIPS	200 Mhz	200 Mhz
Number of physical microphones	2 (1 I2S)	4 (2 I2Ss)
Microphone array type	LMB	LMB
Number of virtual microphones	2	4
Measured MIPS	10 + 13 + 1 = 24 MIPS	33 + 27 + 1 = 61 MIPS
Memory usage (SARAM + DARAM)	144 KB used 176 KB left	168 KB used 152 KB left

5 More About Beamforming

This TI Design uses a beamforming algorithm to form a virtual-directional microphone that points to the direction of the speaker or the desired audio source. The beamforming algorithm amplifies the speech signal from the desired direction, and attenuates all signals from all other directions. In addition to beamforming, TI offers a set of audio algorithms that may further improve the quality of sound-like dynamic range compression. An overview of the audio beamforming mathematics and algorithm can be found in *Acoustic Source Localization and Beamforming: Theory and Practice*[1] and *Beamforming*[2] on Wikipedia.

A group of microphones are mounted at predefined locations, which are either along a straight line or on a circle. A point sound source reaches different microphones with different phase delays. The phase delay depends on the frequency, the speed of sound, the distance between each microphone, and the sound source. The distances between the source and the microphones are function of the direction. Figure 16 shows the distance and the phase difference between two microphones as a function of the direction of the signal arrival.

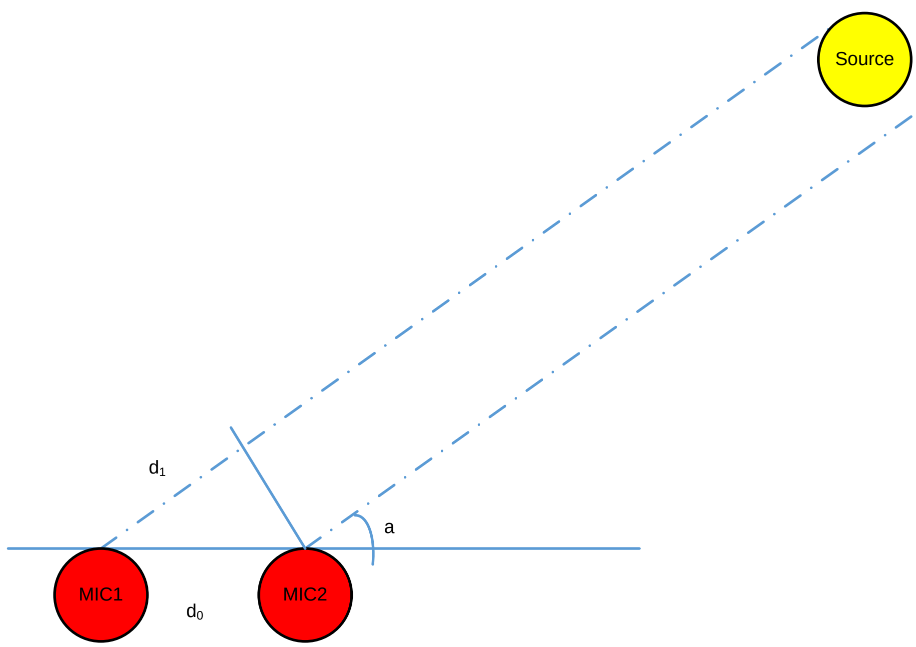


Figure 16. Differences in Distance, Time, and Phase Between Two Microphones

From Figure 16, d1 is calculated in Equation 1.

$$d_1 = d_0 \times \cos(\alpha) \tag{1}$$

The signal time difference, Δ_t , between mic1 and mic2 is d1 divided by the speed of sound, as shown in Equation 2.

$$\Delta_t = \frac{d_1}{sos} \tag{2}$$

The phase difference between mic1 and mic2 is shown in Equation 3.

$$\Delta_0 = 2 \times \pi \times \Delta_t \times f = 2 \times \pi \times f \times \frac{d_1}{sos} = 2 \times \pi \times f \times d_0 \times \frac{\cos(\alpha)}{sos} \tag{3}$$

Where:

- Δ_θ is the phase difference
- f is the signal frequency
- d_0 is the distance between two microphones
- α is the angle of arrival and sos is the speed of sound

In a multi-microphone beamforming system, the algorithm applies a set of delay filters to the microphones' signals to shift the signal phase and get the same phase for all the signals (from all microphones) that arrive from one direction. The contribution of all filtered microphones signals are summed together. Thus, the process amplifies signals that arrive from that direction. Because the phase shift (see Equation 3) depends on the angle of arrival (AOA), the phases of filtered signals that arrive from other directions are not the same. Therefore, the sum of all the signals from another direction is decreased, and the energy of the noise (undesired signal that comes from another direction) is reduced.

From Equation 3, it is clear that the quality of the reduction of noise depends on the noise frequency. While the beamforming filters are designed to reduce noise from typical mid-range and higher frequencies, low-frequency noise will not be reduced. An adaptive ASNR filter is applied to reduce the effect of low-frequency noise. Figure 17 shows the reduction of noise as a function of frequency when the beamforming and the ASNR are applied.

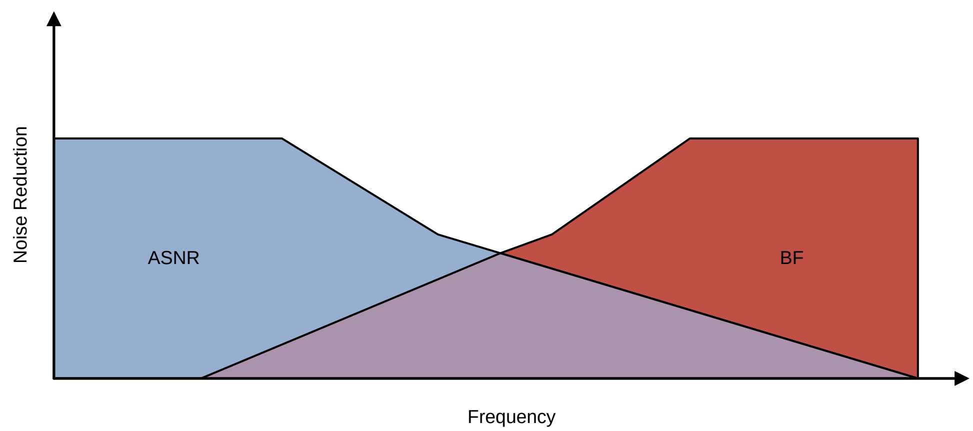


Figure 17. ASNR and BF Noise Reduction as a Function of Noise Frequency

5.1 Multi-Angle Beamforming

Figure 18 shows a typical multi-angle beamforming where multiple beamforming delay filters and ASNR filters are applied to the microphone sets' data. Each BF and ASNR output corresponds to a different angle of arrival (AOA) and behaves like a directional microphone; therefore, it is called virtual microphone. An MSS algorithm chooses the best fit virtual microphone.

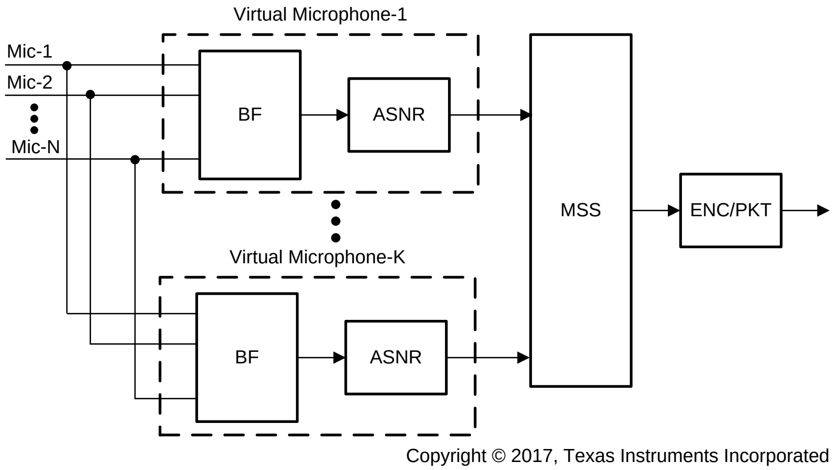


Figure 18. Multi-Angle Beamforming System

6 Design Files

6.1 Schematics

To download the Schematics for each board, see the design files at TIDEP-0077.

6.2 Bill of Materials

To download the Bill of Materials (BOM) for each board, see the design files at TIDEP-0077.

6.3 PCB Layout Recommendations

6.3.1 Layout Prints

To download the Layout Prints for each board, see the design files at TIDEP-0077.

6.4 Altium Project

To download the Altium project files for each board, see the design files at TIDEP-0077.

6.5 Gerber Files

To download the Gerber files for each board, see the design files at TIDEP-0077.

6.6 Assembly Drawings

To download the Assembly Drawings for each board, see the design files at TIDEP-0077.

7 Related Documentation

1. Chen, Joe C., Kung Yao, and Ralph E. Hudson. *Acoustic Source Localization and Beamforming: Theory and Practice*. EURASIP Journal on Advances in Signal Processing 2003, no. 4 (2003): 359-70.
2. Wikipedia, *Beamforming*, Article
3. Texas Instruments, *C5517 General Purpose EVM User Guide*, Wiki Article
4. Texas Instruments, *Audio Pre-Processing Reference Design for Voice-Based Applications*, TIDEP-0088 TI Design (TIDUCR7)
5. Texas Instruments, *C55x CSL Audio Pre-Processing*, Wiki Article
6. Texas Instruments, *Demonstrating Voice Preprocessing on the C5517*, Training Video

7.1 Trademarks

8 About the Authors

LALINDRA JAYATILLEKE is a Digital Applications Engineer at TI specializing in embedded processing applications including audio processing systems.

RAN KATZUR is a senior application engineer at TI where he supports the Sitara™ and the DSP families of system-on-a-chip (SOC) devices. Ran brings his extensive experiences and knowledge in parallel processing and optimization to this role. Ran earned B.Sc., M.Sc. and Ph.D. in applied mathematics from Tel- Aviv University

MING WEI is a senior software engineer at TI where he develops and supports the Processor SDK RTOS for Sitara and the DSP families of SOC devices. Ming brings his extensive experiences and knowledge in real-time systems, signal processing, and code optimization to this role. Ming earned B.Sc., M.Sc. and Ph.D. in computer sciences from Xi'an Jiao-tong University and University of North Texas.

BOBBY TUFINO is an audio systems engineer at TI, where he supports the Sitara and DSP families of SOC devices. He has spent over 15 years supporting customers in the home audio and automotive audio markets. Bobby earned a B.Sc. in audio engineering and a B.Mus. in music engineering technology from the University of Miami.

Revision C History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from B Revision (June 2017) to C Revision **Page**

Revision B History

Changes from A Revision (June 2017) to B Revision **Page**

Revision A History

Changes from Original (June 2017) to A Revision **Page**