Voice Processing Tools and Software K2G & C5517

Lalindra Jayatilleke
EP Catalog Processor Applications

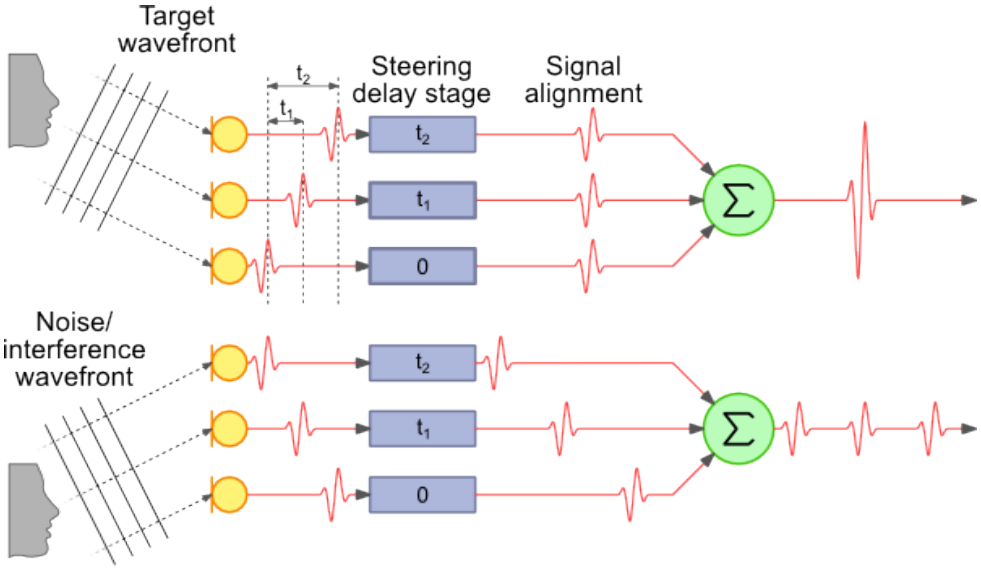
What is acoustic beamforming?

- Signal processing algorithm used to “listen” to an audio source in a noisy environment
- Primary purpose is to eliminate undesired audio and noise
- Based on the concepts first developed for radar, sonar, and electronic warfare
- Applications: Noise Reduction SNR improvement & Acoustic echo cancelling
  - Voice communication
  - Speech recognition

What is acoustic beamforming?

Audio physics of the beamformer (Delay-Sum)

- Sound source wave fronts reach the mics at varying phases. The outputs of the array are phase shifted to form directional virtual microphones.
- These outputs are fed into the delay stage of the beamformer to re-align the phases.
- Audio source from the undesired directions are attenuated.
- Set of coefficients in the delay stage “steers” the BF to the desired direction and form a virtual microphone covering that angle.
- Depending on the direction of the sound source, the Beamforming algorithm (MSS) chooses the virtual mic with the highest energy.



http://www.labbookpages.co.uk/audio/beamforming/delaySum.html

Design tools

Mic array design GUI

Located in the AER package at:
C:\ti\aer_c55l_cpuv3.3_obj_17_0_0_0\tools\bf_tool\bfgui.exe

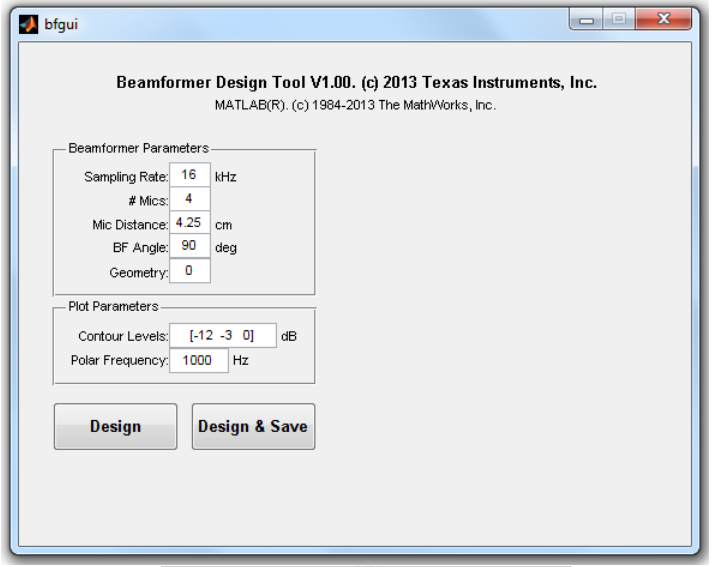
- **Sampling Rate**: 8kHz and 16kHz.
- **# Mics**: The number of microphones in the array
- **Mic Distance**: The distance between the microphone elements in cm
- **BF Angle**: The steering angle of the beamformer in degrees.
- **Geometry**: The supported microphone array geometries are:

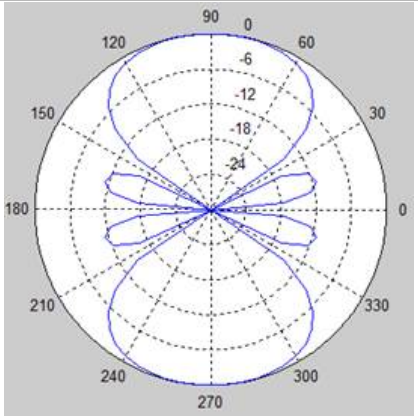
  0: 1-D linear array

  1: 2-D cross array

  2: 2-D rectangular array

  3: 2-D circular array
- **Contour Levels**: The levels in dB used for the contour plot
- **Polar Frequency**: The frequency in Hz for the polar plot. The default is 1000Hz





Design tools

Beamforming coefficients

What do you do with the BF coefficients generated by the BF tool?

It is embedded in the code as seen below.

bf_asnr_mss_vau.h | bf_asnr_mss_vau.c | sysbfflt.c

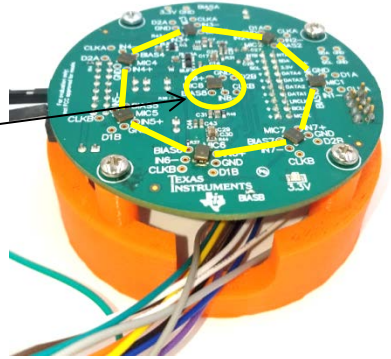
```
/*=================================================================
 *  sysbfflt.c: BF filters for all angles of interest
 *=================================================================*/

#include <xdc/std.h>
#include <ti/mas/types/types.h>

#include "sysbfflt.h"
#include "codec_pcm186x.h"

#if NUM_OF_MICS==6
/* SYS_BF_ANGLE_P0 */
Fract sysBfFltP0[][SYS_BF_FILTER_LENGTH] = {
          {  0,     0,     0,     0,     0,     0,     0,     0,     0,     0, 32767,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0,     0},
          {  4,   -17,    44,   -86,   147,  -236,   366,  -570,   945, -1954, 32534,  2240, -1020,   606,  -388,   251,  -157,    92,   -48,    20,    -5,     0,     0},
          {-14,    54,  -126,   238,  -401,   636,  -984,  1543, -2632,  6114, 31297, -4320,  2169, -1323,   853,  -549,   341,  -196,    98,   -38,     7,     0,     0},
          {  5,   -51,   152,  -323,   581,  -955,  1495, -2311,  3691, -6719, 21527, 19967, -6539,  3622, -2273,  1471,  -938,   570,  -315,   147,   -48,     0,     0},
          {  0,     0,   -48,   147,  -315,   570,  -938,  1471, -2273,  3622, -6539, 19967, 21527, -6719,  3691, -2311,  1495,  -955,   581,  -323,   152,   -51,     5},
          {  0,     0,     7,   -38,    98,  -196,   341,  -549,   853, -1323,  2169, -4320, 31297,  6114, -2632,  1543,  -984,   636,  -401,   238,  -126,    54,   -14},
};

/* SYS_BF_ANGLE_P60 */
Fract sysBfFltP60[][SYS_BF_FILTER_LENGTH] = {
          {  -3,    11,   -25,    49,   -84,   133,  -206,   322,  -540,  1169, 32701, -1086,   518,  -312,   200,  -129,    81,   -47,    24,   -10,     2,     0,     0},
          {  -9,    36,   -85,   161,  -273,   434,  -672,  1051, -1780,  4009, 32080, -3168,  1560,  -947,   609,  -393,   244,  -141,    72,   -28,     6,     0,     0},
          {   0,   -36,   117,  -259,   476,  -791,  1245, -1924,  3043, -5353, 14278, 26408, -6643,  3534, -2192,  1417,  -908,   557,  -314,   152,   -54,     8,     0},
          {   0,     0,   -33,   109,  -243,   449,  -747,  1178, -1820,  2874, -5033, 13128, 27251, -6486,  3427, -2121,  1370,  -879,   540,  -305,   149,   -54,     8},
          {   0,     0,     8,   -42,   110,  -220,   384,  -620,   963, -1494,  2443, -4820, 30795,  7223, -3058,  1786, -1137,   736,  -464,   276,  -147,    64,   -17},
          {   0,     0,   -17,    62,  -143,   269,  -453,   718, -1109,  1741, -2980,  7016, 30893, -4732,  2394, -1464,   943,  -607,   376,  -216,   108,   -41,     8},
};
```

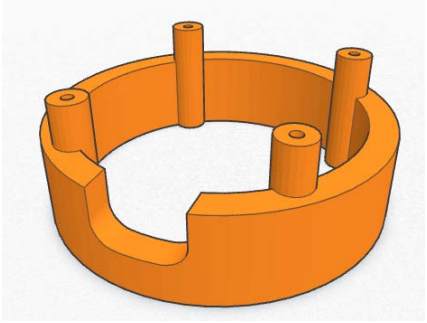
Hardware

Circular Microphone Board (CMB)

- Features 7 microphones around periphery and center (reference) mic.
  Knowles SPH1642HT5H-1 mic.
- 8 microphones feed 2 x PCM1864 Audio ADC
  4 channels, 103dB SNR, programmable gain control
- I2S communication to the DSP
- 4-mic Linear Microphone Board (LMB) is also in the works.
- 3D Printer .STL file available for stand-off.
- With K2G EVM:
  All 8 microphones provided to DSP for processing
- With C5517 EVM:
  2,4,6 microphone inputs from CMB to DSP
  - Limitation on I2S lines (3)
  - CMB geometry (non-equidistant)
- With C6747 EVM:
  All 8 microphones provided to DSP for processing

Center
(Reference) Mic





Hardware

Audio preprocessing demo flow

8-mic CMB

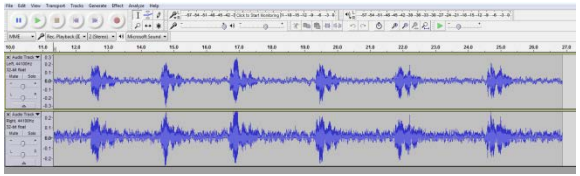
I2S

K2G or C5517 EVM

Headphone

Line out

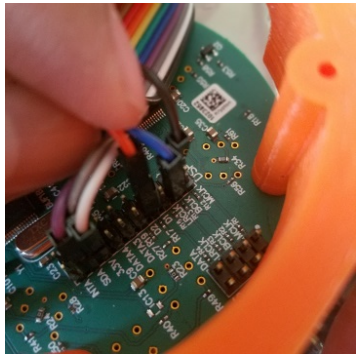
Headphones



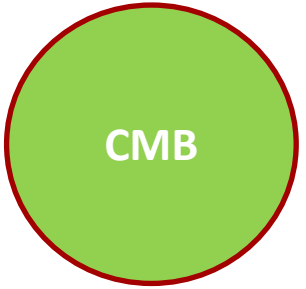
Audacity® audio editing software:
Split the left and right channels to listen to clean and unclean audio.

Hardware

K2G EVM connections to the CMB (8 mics)



Header/Pin	Pin
JP1 - 100	GND
JP1 - 10	3.3V
JP1 - 97	I2C SCL
JP1 - 98	I2C SDA
JP1 - 33	Bit Clock
JP1 - 41	Frame Clock
JP1 - 29	Data 1
JP1 - 36	Data 2
JP1 - 47	Data 3
JP1 - 40	Data 4
JP1 - 48	MCLK

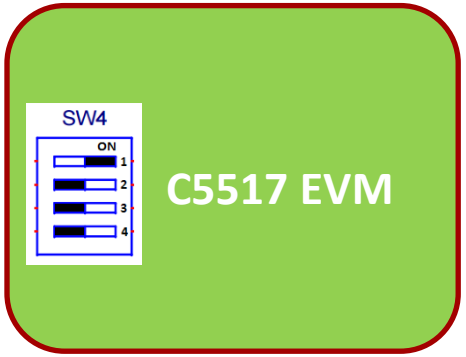


CMB Jumper Settings

J3 – ON
J10 – ON
J11 – ON
J8 (Pins 1&2) – ON
J8 (Pins 3&4) – OFF

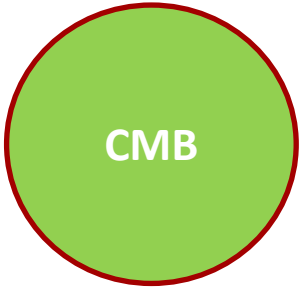
Hardware

C5517 EVM connections to the CMB (6 mics)



J28_Pin1_Pin3 (jumper off)
J28_Pin2_Pin4 (jumper off)
J29_Pin1_Pin3 (jumper on)
J29_Pin2_Pin4 (jumper on)
J30_Pin1_Pin3 (jumper on)
UART_EN (no jumper)

Header/Pin	Pin
J10 - 5	GND
J10 - 9	3.3V
J14 - 16	I2C SCL
J14 - 20	I2C SDA
J27 - 3	Bit Clock
J27 - 4	Frame Clock
J30 - 2	Data 1
J31 - 2	Frame Clock
J31 - 3	Bit Clock
J31 - 1	Data 3
J27 - 1	Bit Clock
J27 - 2	Frame Clock
J28 - 2	Data 2



CMB Jumper Settings

J3 – ON
J10 – ON
J11 – ON
J8 (Pins 1&2) – ON
J8 (Pins 3&4) – OFF

Software components & dependencies

Software components:

- **K2G Processor SDK RTOS v3.03**
  Located at C:\ti\processor_sdk_rtos_k2g_3_03_00_00\demos\audio-preprocessing
- **C55x Chip Support Library (CSL) v3.07**
  Located at C:\ti\c55_lp\c55_csl_3.07\c55xx_csl\demos\audio-preprocessing\

Dependencies:

- **DSP components: AER 17.0.0.0 (C64P/C55x)**
  Beamforming (BF)
  Adaptive Spectral Noise Reduction (ASNR)
  Dynamic Range Compression (DRC)
- **VOLIB 2.1.0.1 (C64P, C66x, C55x)**
  Multi-Source Selection (MSS)

Software demonstrations

- **K2G (C66x) Processor SDK RTOS package:**
  - The real-time demo takes the inputs from the mic array puts them through BF+ASNR+MSS+DRC and outputs to the headphones.
  - The File IO demo processes audio data fed from a file. Good for simulating.
  - The loopback demo puts the raw input from the mics to the headphones directly without processing.

- **Chip Support Library (C55x)**:
  - The real-time demo takes the inputs from the mic array puts them through BF+ASNR+MSS+DRC and outputs to the headphones.
  - The loopback demo puts the raw input from the mics to the headphones directly without processing.

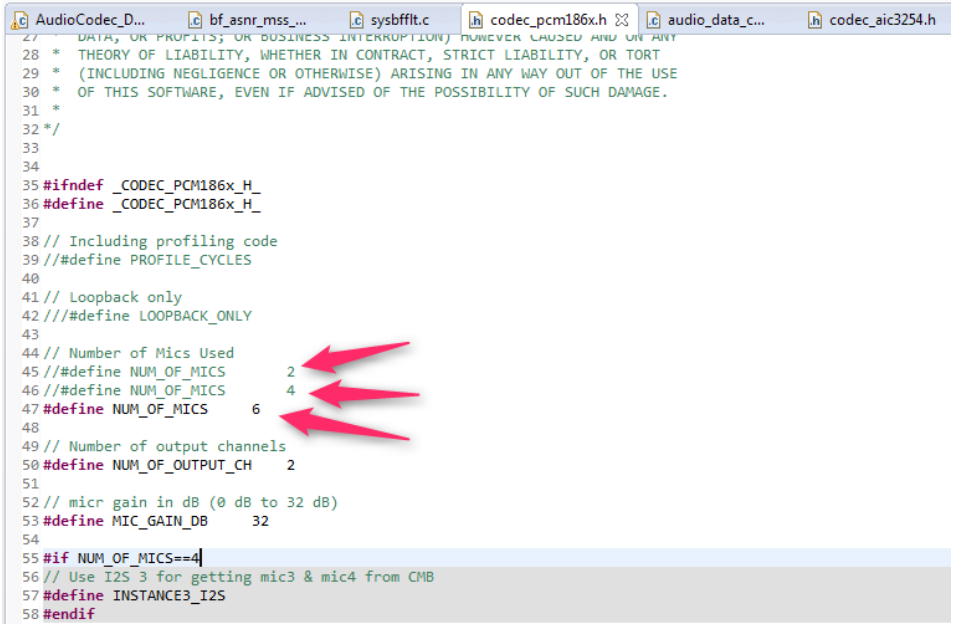
Software tuning

Tuning knobs in the software:

- ASNR parameters: Signal delay, frequency band boundary, max attenuation, max signal update rate, noise threshold, noise hangover.
- Filter coefficients are located in sysbfflt.c. These are values attained from the BF GUI tool.
- Number of microphones used 2, 4 ,6 and 8. This is restricted by the number of available audio input channels (McASP, I2S), MIPS, and memory.
- Individually analyze clean (from BF) or unclean audio (directly from the array). These are fed to either the left or right channels. The sides can also be flipped.
- Loopback mode to test individual microphone functionality.
- Codec gain settings can be manipulated as needed.

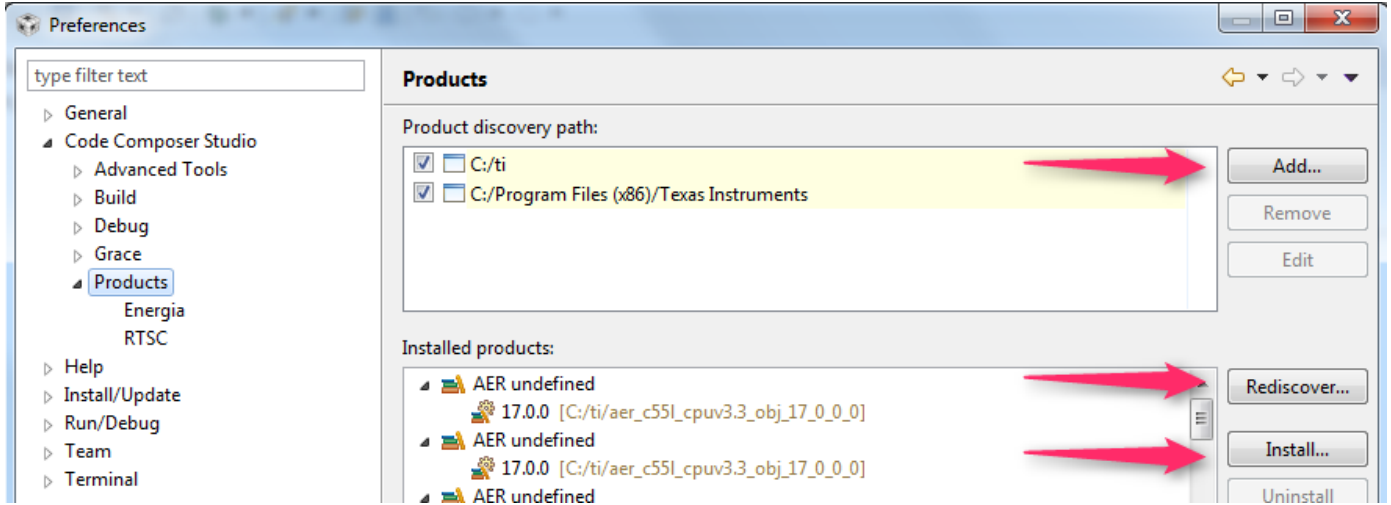
How to specify the number of mics in software?

- #define NUM_OF_MICS parameter in codec_pcm186x.h.
- The NUM_OF_MICS parameter is manipulated in the same file in K2G and C5517.
- C5517 supports a maximum of 6 mics due to the hardware limitation on I2S line availability.



Setting up build environment for K2G demos

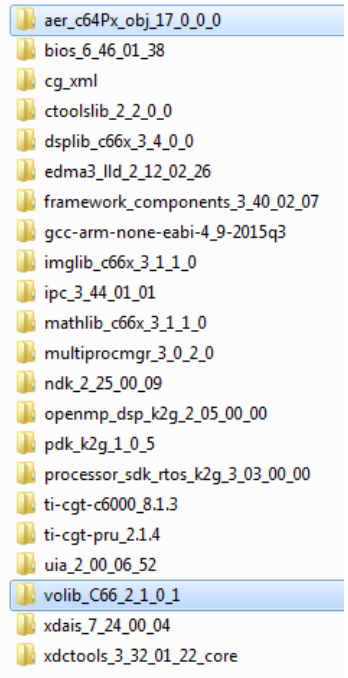
- Download and install the Processor SDK K2G RTOS package. Default location is C:\ti.
- Ensure that all the components installed as part of the SDK are registered with CCS.



- Manipulate C:\ti\pdk_k2g_1_0_5\packages\pdksetupenv.bat if not using the default installation location. The key changes are the CCS and PRSDK installation paths.

K2G: AER and VOLIB software installation

Install AER and VOLIB alongside the Processor SDK RTOS K2G installation location. By default this is at C:\ti



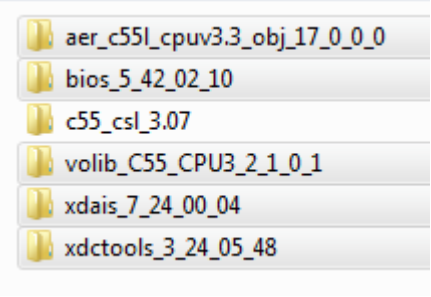
AER (CPU rev 3)
http://software-dl.ti.com/libs/aer/latest/index_FDS.html

VOLIB (CPU rev 3)
http://software-dl.ti.com/sdoemb/sdoemb_public_sw/volib/latest/index_FDS.html

C55x: AER and VOLIB software installation

Install AER and VOLIB alongside the C55x CSL installation location. By default this is at C:\ti\c55_lp

Ensure that all the other components highlighted below are also installed in this directory. Download links can be found at the end of the presentation.



AER (CPU rev 3)
http://software-dl.ti.com/libs/aer/latest/index_FDS.html

VOLIB (CPU rev 3)
http://software-dl.ti.com/sdoemb/sdoemb_public_sw/volib/latest/index_FDS.html

Building and running K2G real-time demo

- Setenv.bat in PRSDK.
- Building K2G RT demo.
- Running the demo in CCS.
- Capture of audio and analysis in Audacity

K2G_BF_demo_blds2.mp4

Building and running 5517 real-time demo

- Capture of importing & building C5517 RT demo.
- Capture of running the demo in CCS

C5517_BF_demo_blds.mp4

For more information

- Software downloads:
  - Processor SDK RTOS for K2G: http://software-dl.ti.com/processor-sdk-rtos/esd/K2G/latest/index_FDS.html
  - C55x Chip Support Library (CSL): http://software-dl.ti.com/dsps/dsps_public_sw/dsps_swops_houston/C55X/latest/index_FDS.html
  - BIOS 5.42.02.10: http://software-dl.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/dspbios/5_42_02_10/index_FDS.html
  - AER: http://software-dl.ti.com/libs/aer/latest/index_FDS.html
  - VOLIB: http://software-dl.ti.com/sdoemb/sdoemb_public_sw/volib/latest/index_FDS.html
- RTOS Software Developer Guide: Building The SDK: http://processors.wiki.ti.com/index.php/Processor_SDK_RTOS_Building_The_SDK
- Code Composer Studio (CCS) Training: http://processors.wiki.ti.com/index.php/Category:CCS_Training
- Audacity® Audio analysis tool: http://www.audacityteam.org/
- CMB 3D printer stand-off file: http://software-dl.ti.com/public/download/CMB_Enclosure.zip
- For questions about this training, refer to the E2E Community Forums: http://e2e.ti.com