**Department of Electrical Engineering**

**Faculty Member: Dr. Wajahat Hussain** **Dated: 7-3-2107**

# Course/Section: BEE6-B Semester: 6th Semester

**EE-330 Digital Signal Processing**

**Lab4: Audio Processing using DSP Kit TMS 320C6713 DSK**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Reg. no.** | **Report**  **Marks**  **/ 10** | **Lab Quiz-**  **Viva Marks**  **/ 5** | **Total / 15** |
| **Saad Iqbal** | **111394** |  |  |  |
| **Usman Iqbal** | **111393** |  |  |  |
| **Abdullah Bin Asif** | **111596** |  |  |  |

# Lab4: Audio Processing using DSP Kit TMS 320C6713 DSK

**Objectives**

The objective of this lab is to explore some more features of Code Composer Studio (CCS) that is time domain and frequency domain plots of audio signal in addition to that we will also do real time processing of audio input.

* Time domain and frequency Domain Plots in CCS
* Real Time processing of Audio Signal
* Working with basic sinusoids on DSP Kit

**Lab Instructions**

* This lab activity comprises of three parts: Pre-lab, Lab Exercises, and Post-Lab Viva session.
* The students should perform and demonstrate each lab task separately for step-wise evaluation (please ensure that course instructor/lab engineer has signed each step after ascertaining its functional verification)
* Only those tasks that completed during the allocated lab time will be credited to the students. Students are however encouraged to practice on their own in spare time for enhancing their skills.

**Lab Report Instructions**

All questions should be answered precisely to get maximum credit. Lab report must ensure following items:

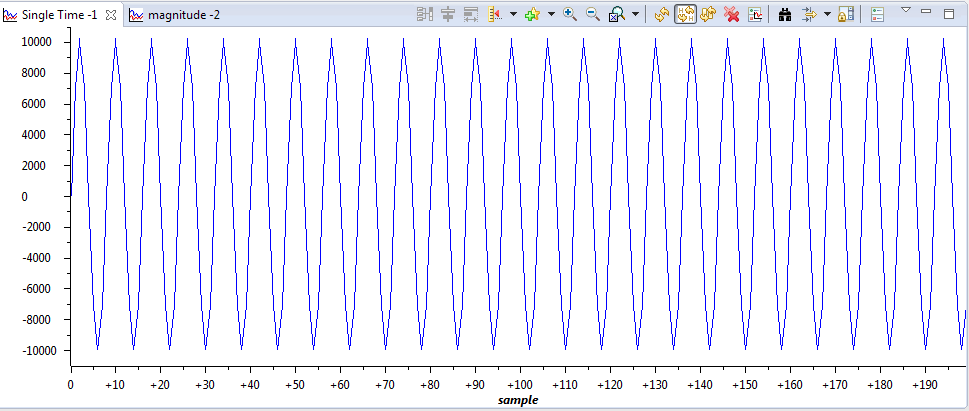
* Lab objectives
* MATLAB/C codes
* Results (graphs/tables) duly commented and discussed
* Conclusion

**Introduction**

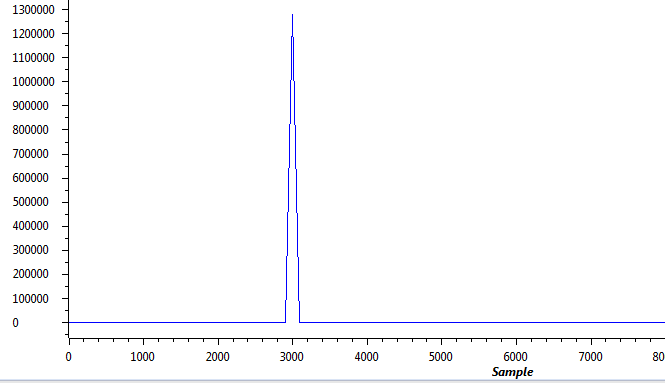
In this Lab we will explore some more features of Code Composer Studio that is time domain and frequency domain plots of audio signal in addition to that we will also learn real time processing of audio input. We will also see the details of support files that we used in previous Lab.

**LAB TASK 1**

**Waveform plot**

****

**Magnitude plot (FFT)**

****

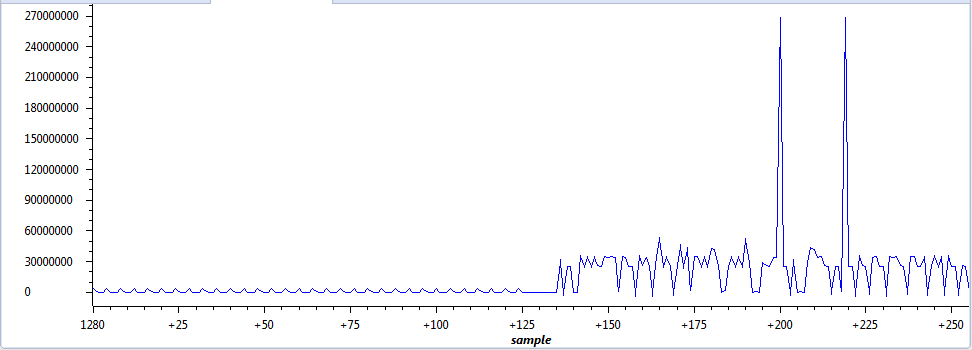
**LAB TASK 2**

|  |
| --- |
| **CODE:**  **//sine8\_LED.c sine generation with DIP switch control**  **#include "dsk6713\_aic23.h" //codec support**  **Uint32 fs = DSK6713\_AIC23\_FREQ\_8KHZ; //set sampling rate**  **#define DSK6713\_AIC23\_INPUT\_MIC 0x0015**  **#define DSK6713\_AIC23\_INPUT\_LINE 0x0011**  **Uint16 inputsource=DSK6713\_AIC23\_INPUT\_MIC;//select input**  **#define LOOPLENGTH 16**  **short loopindex = 0; //table index**  **short gain = 10; //gain factor**  **short sine\_table[LOOPLENGTH]={1000,914,669,309,-104,-500,-809,-978,-978,-809,-500,-104,309,669,914,1000}; //sine values**  **int i;**  **void main()**  **{**  **comm\_poll(); //init DSK,codec,McBSP**  **DSK6713\_LED\_init(); //init LED from BSL**  **DSK6713\_DIP\_init(); //init DIP from BSL**  **while(1) //infinite loop**  **{**  **if(DSK6713\_DIP\_get(3)==0) //=0 if DIP switch #0 pressed**  **{**  **DSK6713\_LED\_on(3); //turn LED #0 ON**  **for (i=1;i<40000;i++){**  **output\_left\_sample(sine\_table[loopindex++]\*gain); //output sample**  **if (loopindex >= LOOPLENGTH) loopindex = 0; //reset table index**  **}**  **}**  **else DSK6713\_LED\_off(3); //turn LED off if not pressed**  **} //end of while(1) infinite loop**  **} //end of main** |

**LAB TASK 3**

|  |
| --- |
| **CODE:**  **//sine8\_buf.c sine generation with output stored in buffer**  **#include "DSK6713\_AIC23.h" //codec support**  **Uint32 fs=DSK6713\_AIC23\_FREQ\_16KHZ; //set sampling rate**  **#define DSK6713\_AIC23\_INPUT\_MIC 0x0015**  **#define DSK6713\_AIC23\_INPUT\_LINE 0x0011**  **Uint16 inputsource=DSK6713\_AIC23\_INPUT\_MIC; // select input**  **#define LOOPLENGTH 8**  **#define BUFFERLENGTH 128**  **int loopindex = 0; //table index**  **int bufindex = 0; //buffer index**  **int out\_buffer[BUFFERLENGTH]; //output buffer**  **short gain = 10;**  **interrupt void c\_int11() //interrupt service routine**  **{**  **short out\_sample;**  **// possible sampling rates: 8, 16, 24, 32, 44, 48, 96 kHz**  **out\_sample =input\_sample();**  **output\_sample(out\_sample); //output sample value**  **out\_buffer[bufindex++] = out\_sample; //store in buffer**  **if (bufindex >= BUFFERLENGTH) bufindex = 0; //check for end of buffer**  **return; //return from interrupt**  **}**  **void main()**  **{**  **comm\_intr(); //initialise DSK**  **while(1); //infinite loop**  **}** |

**Waveform**

****

**Conclusion**

In this lab we studied the fundamental concepts of signal processing using the DSK kit. We learned to generate a tone signal with a specific frequency, also we observed the FFT of a signal. Also the effects on the waveform were seen when the sampling rate was changed. In the last part we took an input from the user and displayed the waveform on the interface.