**LAB#04**

**Object:** Introduction to DSKC64167 processor & related software.

**Theory:**

**DSK 6416** is a [DSP Starter Kit](https://en.wikipedia.org/wiki/DSP_Starter_Kit) by [Texas Instruments](https://en.wikipedia.org/wiki/Texas_Instruments). The kit has TMS320C6416 [fixed point](https://en.wikipedia.org/wiki/Fixed-point_arithmetic) [digital signal processor](https://en.wikipedia.org/wiki/Digital_signal_processor), a member of C6000 series of processors that is based on VelociTI.2 architecture. **DSK 6416** is a [DSP Starter Kit](https://en.wikipedia.org/wiki/DSP_Starter_Kit) by [Texas Instruments](https://en.wikipedia.org/wiki/Texas_Instruments). The kit has TMS320C6416 [fixed point](https://en.wikipedia.org/wiki/Fixed-point_arithmetic) [digital signal processor](https://en.wikipedia.org/wiki/Digital_signal_processor), a member of C6000 series of processors that is based on VelociTI.2 architecture.

**TASK #01:** Simulate a C language code to generate 1KHz tone.

**CODING:**

#include "tonecfg.h"

#include "dsk6416.h"

#include "dsk6416\_aic23.h"

#define SINE\_TABLE\_SIZE 48

DSK6416\_AIC23\_Config config = {

0x0017, // 0 DSK6416\_AIC23\_LEFTINVOL Left line input channel volume

0x0017, // 1 DSK6416\_AIC23\_RIGHTINVOL Right line input channel volume

0x00d8, // 2 DSK6416\_AIC23\_LEFTHPVOL Left channel headphone volume

0x00d8, // 3 DSK6416\_AIC23\_RIGHTHPVOL Right channel headphone volume

0x0011, // 4 DSK6416\_AIC23\_ANAPATH Analog audio path control

0x0000, // 5 DSK6416\_AIC23\_DIGPATH Digital audio path control

0x0000, // 6 DSK6416\_AIC23\_POWERDOWN Power down control

0x0043, // 7 DSK6416\_AIC23\_DIGIF Digital audio interface format

0x0001, // 8 DSK6416\_AIC23\_SAMPLERATE Sample rate control

0x0001 // 9 DSK6416\_AIC23\_DIGACT Digital interface activation

};

/\* Pre-generated sine wave data, 16-bit signed samples \*/

Int16 sine table [SINE\_TABLE\_SIZE] = {

0x0000, 0x10b4, 0x2120, 0x30fb, 0x3fff, 0x4dea, 0x5a81, 0x658b,

0x6ed8, 0x763f, 0x7ba1, 0x7ee5, 0x7ffd, 0x7ee5, 0x7ba1, 0x76ef,

0x6ed8, 0x658b, 0x5a81, 0x4dea, 0x3fff, 0x30fb, 0x2120, 0x10b4,

0x0000, 0xef4c, 0xdee0, 0xcf06, 0xc002, 0xb216, 0xa57f, 0x9a75,

0x9128, 0x89c1, 0x845f, 0x811b, 0x8002, 0x811b, 0x845f, 0x89c1,

0x9128, 0x9a76, 0xa57f, 0xb216, 0xc002, 0xcf06, 0xdee0, 0xef4c

};

void main()

{

DSK6416\_AIC23\_CodecHandle hCodec;

Int16 msec, sample;

DSK6416\_init();

hCodec = DSK6416\_AIC23\_openCodec(0, &config);

for (msec = 0; msec < 5000; msec++)

{

for (sample = 0; sample < SINE\_TABLE\_SIZE; sample++)

{

while (!DSK6416\_AIC23\_write(hCodec, sinetable[sample]));

while (!DSK6416\_AIC23\_write(hCodec, sinetable[sample]));

}

}

DSK6416\_AIC23\_closeCodec(hCodec);

**TASK #02:** Simulate a C language code to blink led 0 & turn on led 8 with switch 3.

**CODING:**

#include "dsk6416.h"

#include "dsk6416\_led.h"

#include "dsk6416\_dip.h"

void main()

{

DSK6416\_init();

DSK6416\_LED\_init();

DSK6416\_DIP\_init();

while(1)

{

DSK6416\_LED\_toggle(0);

if (DSK6416\_DIP\_get(3) == 0)

DSK6416\_LED\_on(8);

else

DSK6416\_waitusec(200000);

}

}

**CONCLUSION:**

Through this lab we have understand the DSKC6416T processor & by implementing C code & perform two tasks by using its libraries, one is of about running LEDS and another is of about generating 1KHz frequency tone.