California State University

EGCP 450

Lab 4

Professor

Zaid Frayeh

Code :

Main.c :

```
#include <stdint.h>
#include "msp432p401r.h"
#include "SOUND.h"
#include "PIANO.h"
#include "SysTickInts.h"



void main(void){

     uint8_t PIANO = 0;


      PIANO_INIT();
      Sound_Init();


      EnableInterrupts();
      WaitForInterrupt();


      while(1) {
       PIANO = PIANO_IN();
                                      if(PIANO == 0x00)
                                      {
                                          SysTick->LOAD=0;

                                      }
                                      else if(PIANO == 0x01)
                                      {
                                          SysTick->LOAD=269;

                                      }

                                      else if(PIANO == 0x02)
                                      {
                                          SysTick->LOAD = 190;

                                      }

                                      else if(PIANO == 0x04)
                                      {
                                          SysTick->LOAD = 213;

                                      }


      }
};
```

Dac.h :

```c
#ifndef DAC_H_
#define DAC_H_


void DAC_INIT(void);
void DAC_OUT(uint32_t data);


#endif /* DAC_H_ */
```

Dac.c :

```c
#include <stdint.h>
#include "SOUND.h"
#include "msp432p401r.h"
#include "DAC.h"
#define DAC_out (*((volatile uint8_t *)0x40004C23))


void DAC_INIT(void)
{
            //output pins for dac port 4 pins 0-3
            P4SEL0 &= ~0x0F;
            P4SEL1 &= ~0x0F;
            P4DIR  |= 0x0F;

}
void DAC_OUT( uint32_t data )
{

    DAC_out=data;
}
```

Sound.h :

```c
#ifndef SOUND_H_
#define SOUND_H_
void Sound_Init(void);
void Sound_play(uint8_t note);

#endif /* SOUND_H_ */
```

Sound.c :

```c
#include <stdint.h>
#include "msp432p401r.h"
#include "SOUND.h"
#include "DAC.h"
#include "SysTickInts.h"


    int sound = 0;
    int indx = 0;

const uint8_t wave[32]= {
   8,9,11,12,13,14,14,15,15,15,14,
   14,13,12,11,9,8,7,5,4,3,2,
   2,1,1,1,2,2,3,4,5,7};

void Sound_Init(void){

    DAC_INIT();
    SysTick_Init(0);
}


void Sound_play(uint8_t note){


    if(note == 1)      {

            indx = (indx + 1) &0x1f ;
            DAC_OUT(wave[indx]);
                       }

    else if(note == 2) {

            indx = (indx + 1) &0x1f ;
            DAC_OUT(wave[indx]);

                        }

    else if (note == 4) {

           indx = (indx + 1) &0x1f ;
            DAC_OUT(wave[indx]);

                        }

    else if(note == 0)  {
            DAC_OUT(0);

                        }

}
```

Piano.h :

```c
#ifndef PIANO_H_
#define PIANO_H_

void PIANO_INIT(void);

int PIANO_IN( void);


#endif /* PIANO_H_ */
```

Piano.c :

```c
#include <stdint.h>
#include "msp432p401r.h"
#include "piano.h"
#define piano_Input (*((volatile uint8_t *)0x40004C40))
//input port 5 pins 5.4 5.5 5.6 because other pins are giving wrong values
void PIANO_INIT(void){
            P5SEL0 &= ~0x70;
            P5SEL1 &= ~0x70;
            P5DIR  &= ~0x70;
}
int PIANO_IN(void){
        uint32_t input;
        input = (piano_Input&0x70);
        input= input/16;
        return input;

}
```

SysTickints.h :

```c
#ifndef __SYSTICKINTS_H__
#define __SYSTICKINTS_H__

void SysTick_Init(uint32_t period);
void SysTick_Handler(void);

#endif
```

SysTickints.c :

```
#include <stdint.h>
#include "SOUND.h"
#include "PIANO.h"
#include "DAC.h"
#include "msp432p401r.h"

void DisableInterrupts(void); // Disable interrupts
void EnableInterrupts(void);  // Enable interrupts
long StartCritical (void);    // previous I bit, disable interrupts
void EndCritical(long sr);    // restore I bit to previous value
void WaitForInterrupt(void);  // low power mode
volatile uint32_t Counts;
volatile uint32_t note;

void SysTick_Init(uint32_t period) {
      long sr = StartCritical();

      Counts = 0;

      SysTick->CTRL = 0;                    // disable SysTick during setup
      SysTick->LOAD = period - 1;           // maximum reload value
      SysTick->VAL = 0;                     // any write to current clears it
      SCB->SHP[3] = (SCB->SHP[3]&0x00FFFFFF)|0x40000000;    // priority 2
      SysTick->CTRL = 0x00000007;           // enable SysTick with no
interrupts

      EndCritical(sr);
}

void SysTick_Handler(void){
                                   if(SysTick->LOAD == 213)
                                       {
                                           Sound_play(4);
                                       }
                                       else if(SysTick->LOAD == 190)
                                       {
                                            Sound_play(2);
                                       }

                                       else if(SysTick->LOAD == 269)
                                       {
                                           Sound_play(1);
                                       }

                                       else if(SysTick->LOAD == 0)
                                       {
                                           Sound_play(0);
                                       }
}
```

Table for voltages :

DECIMAL VALUE	VOLTAGE (V) lab	Voltage Theoreticle
0	35.6m	0
1	247.6m	0.22
2	460.7m	0.44
3	0.672	0.66
4	0.864	0.88
5	1.075	1.1
6	1.288	1.32
7	1.499	1.54
8	1.667	1.76
9	1.878	1.98
10	2.089	2.2
11	2.301	2.42
12	2.491	2.64
13	2.701	2.86
14	2.913	3.08
15	3.123	3.3

My values appear to be right for our voltage range is supposed to be between 0 and 3.3v for the DAC that I am designing and they are close to the theoretical values our values are supposed to be increments of 0.22 because that is our resolution.

	Theoretical	LAB
Range	0-3.3 v	35.6m – 3.123 v
Resolution	3.3/(2^4 -1) =0.22	3.123-35.6m/(2^4 -1) =0.206
Precision	2^4 = 16	16

I believe my values are correct they are close to the theoretical values our range is supposed to be between 0 and 3.3v range is close and as well as resolution. There is a slight difference in resolution due to the range since it depends on the range Precision will be the same since we have the same number of bits.

Systic load values:

A note :

3000000/440=6818.18
6818/32=> 213 counts

B note :

3000000/494=6072.87
6073/32=> 190 counts

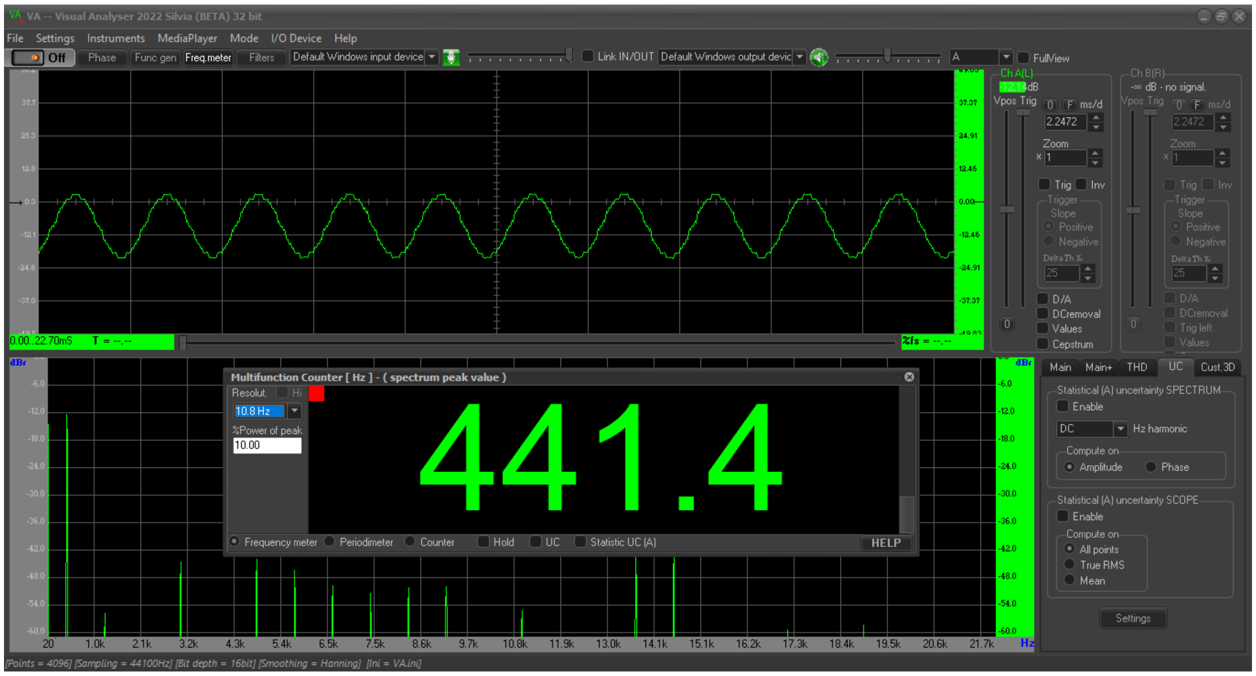
F note :

3000000/349=8595.9
8596/32=> 269 counts

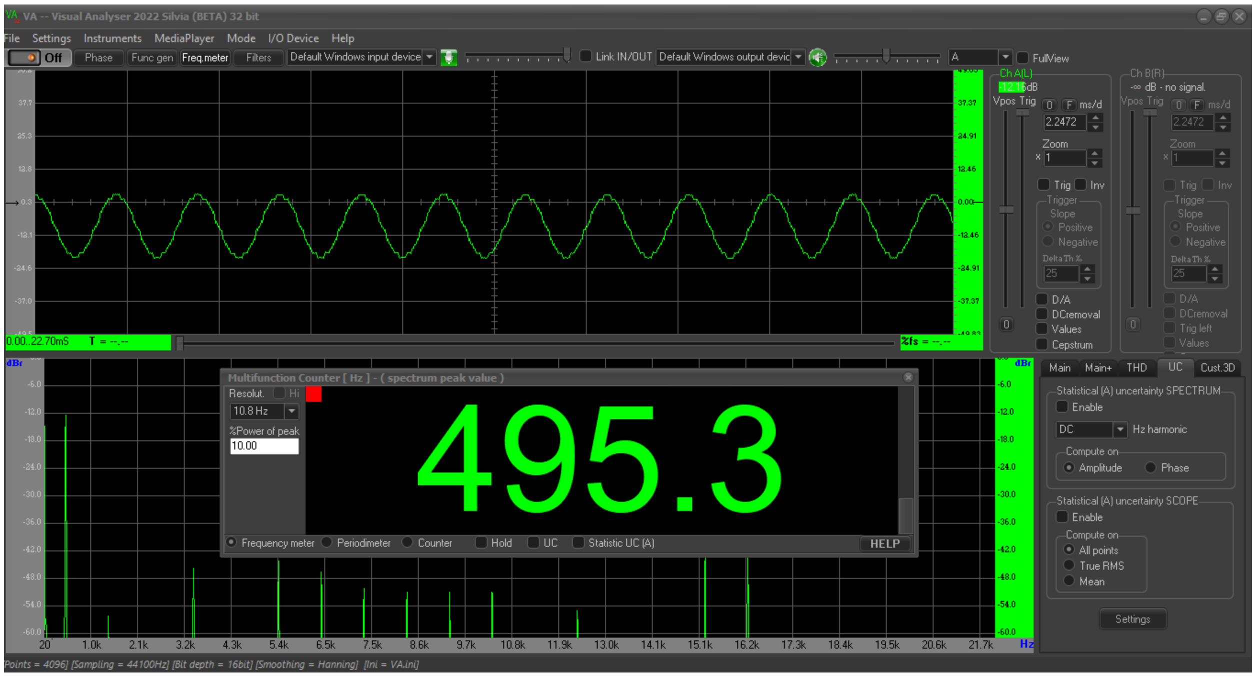
My values are correct because when I use the visual analyzer to check my frequency I am getting the expected frequency that I want my DAC to output for each note

Screenshots :

A note (actual value is 440Hz)

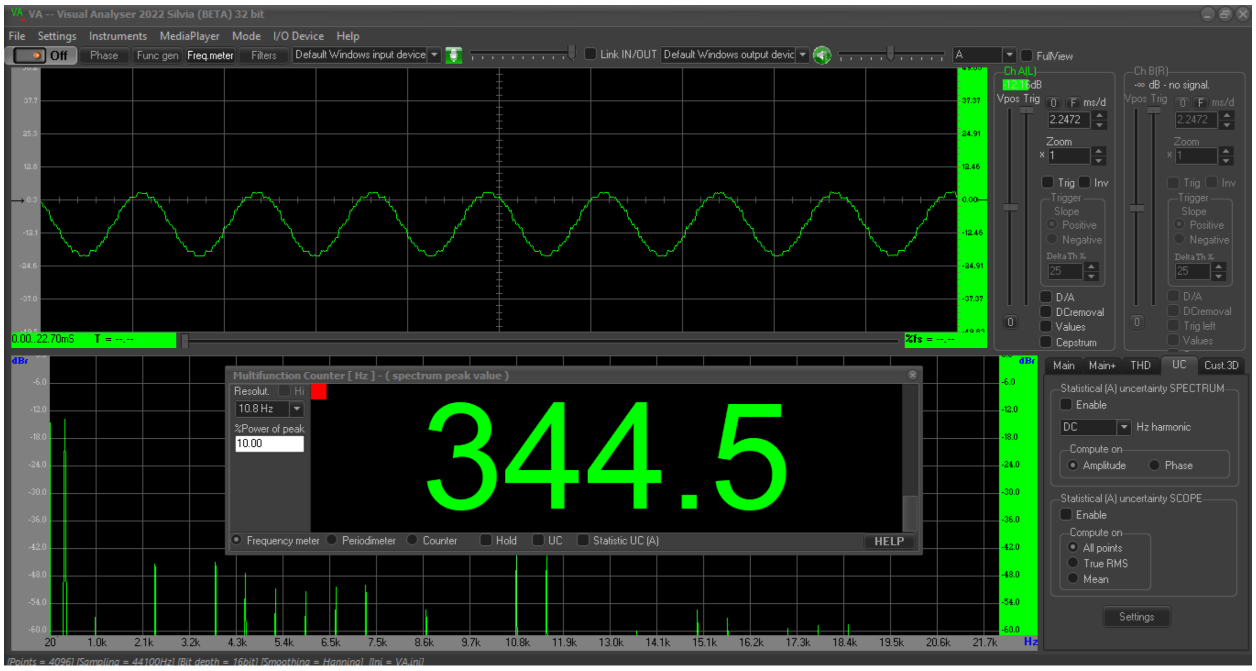

VA -- Visual Analyser 2022 Silvia (BETA) 32 bit
File Settings Instruments MediaPlayer Mode I/O Device Help
Off Phase Func gen Freq.meter Filters Default Windows input device Link IN/OUT Default Windows output devic A FullView
Ch A(L) -12.14dB
Ch B(R) -∞ dB - no signal
Vpos Trig
ms/d 2.2472
Zoom x 1
Trig Inv
Trigger Slope Positive Negative
Delta Th % 25
D/A DCremoval Values Cepstrum
Multifunction Counter [Hz] - (spectrum peak value)
Resolut. Hi
10.8 Hz
%Power of peak 10.00
441.4
Frequency meter Periodimeter Counter Hold UC Statistic UC (A) HELP
Main Main+ THD UC Cust.3D
Statistical (A) uncertainty SPECTRUM Enable
DC Hz harmonic
Compute on Amplitude Phase
Statistical (A) uncertainty SCOPE Enable
Compute on All points True RMS Mean
Settings
[Points = 4096] [Sampling = 44100Hz] [Bit depth = 16bit] [Smoothing = Hanning] [Ini = VA.ini]

B note (actual value is 494 Hz)



```
VA -- Visual Analyser 2022 Silvia (BETA) 32 bit
Multifunction Counter [ Hz ] - ( spectrum peak value )
Resolut. 10.8 Hz
%Power of peak 10.00
495.3
Frequency meter   Periodimeter   Counter   Hold   UC   Statistic UC (A)
Points = 4096] [Sampling = 44100Hz] [Bit depth = 16bit] [Smoothing = Hanning]  [Ini = VA.ini]
```

F note (actual value is 349 Hz)



```
VA -- Visual Analyser 2022 Silvia (BETA) 32 bit
Multifunction Counter [ Hz ] - ( spectrum peak value )
Resolut. 10.8 Hz
%Power of peak 10.00
344.5
Frequency meter   Periodimeter   Counter   Hold   UC   Statistic UC (A)
[Points = 4096] [Sampling = 44100Hz] [Bit depth = 16bit] [Smoothing = Hanning]  [Ini = VA.ini]
```

Sources :
I used the sources from the HTML document provided as well as brainstorming with David Mouser about the functions and where each function belongs in the project’s file .