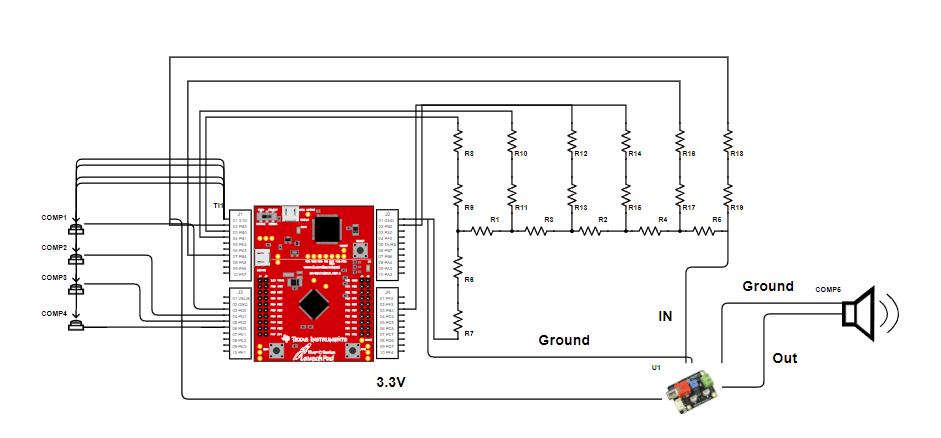
CECS447 Project 1 report

Introduction：

In this project I will build a digital piano with a music box function using a 6 bits R2R DAC.

Circuit diagram：



In this project I used three different ports which are Port E,PortB and Port F.

PortE: is for the piano buttons input.

PortB: the output of the 6 bits DAC.

PortF: two buttons on the board to change the mode and the music/tone and LED to show if there is a sound coming out.

Code:

There are three different C files that I used in this project. Sound.c is where the Port B (the DAC ) got defined.

unsigned char Index;

// 6-bit 64-element sine wave

const unsigned char SineWave[64] =

{32,35,38,41,44,47,49,52,54,56,58,59,61,62,62,63,

63,63,62,62,61,59,58,56,54,52,49,47,44,41,

38,35,32,29,26,23,20,17,15,12,10,8,6,5,3,2,2,

1,1,1,2,2,3,5,6,8,10,12,15,17,20,23,26,29};

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*DAC\_Init\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Initialize 6-bit DAC

// Input: none

// Output: none

void DAC\_Init(void){

volatile unsigned long delay;

SYSCTL\_RCGC2\_R |= SYSCTL\_RCGC2\_GPIOB; // 1) B clock

delay = SYSCTL\_RCGC2\_R; // delay

GPIO\_PORTB\_CR\_R = 0x3F; // allow changes to PB5-0

GPIO\_PORTB\_AMSEL\_R &= ~0x3F; // 3) disable analog function

GPIO\_PORTB\_PCTL\_R &= ~0x00FFFFFF; // 4) GPIO clear bit PCTL

GPIO\_PORTB\_DIR\_R |= 0x3F; // 5) PB5-0 output

GPIO\_PORTB\_AFSEL\_R &= ~0x3F; // 6) no alternate function

GPIO\_PORTB\_DEN\_R |= 0x3F; // 7) enable digital pins PB5-0

GPIO\_PORTB\_DR8R\_R |= 0x3F; // enable 8 mA drive on PB5-0

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*Sound\_Init\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Initialize Systick periodic interrupts

// Input: interrupt period

// Units of period are 12.5ns

// Maximum is 2^24-1

// Minimum is determined by lenght of ISR

// Output: none

void Sound\_Init(unsigned long period){

Index = 0;

NVIC\_ST\_CTRL\_R = 0; // disable SysTick during setup

NVIC\_ST\_RELOAD\_R = period-1;// reload value

NVIC\_ST\_CURRENT\_R = 0; // any write to current clears it

NVIC\_SYS\_PRI3\_R = (NVIC\_SYS\_PRI3\_R&0x00FFFFFF)|0x40000000; // priority 2

NVIC\_ST\_CTRL\_R = 0x0007; // enable SysTick with core clock and interrupts

}

void Sound\_stop(void)

{

NVIC\_ST\_CTRL\_R = 0;

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*DAC\_Out\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// output to DAC

// Input: 6-bit data, 0 to 63

// Output: none

void DAC\_Out(unsigned long data){

GPIO\_PORTB\_DATA\_R = data;

}

// Interrupt service routine

// Executed every 12.5ns\*(period)

void SysTick\_Handler(void){

Index = (Index+1)&0x3F; // 64 samples for each period

// Index = (Index+1)%16;

GPIO\_PORTB\_DATA\_R = SineWave[Index]; // output to DAC: 3-bit data

}

The second file is the Switchled.c which defines the Port F function and initializes it.

//---------------------Switch\_Init---------------------

// initialize switch interface

// Input: none

// Output: none

void SwitchLed\_Init(void){ volatile unsigned long delay;

SYSCTL\_RCGC2\_R |= 0x00000020; // 1) activate clock for Port F

delay = SYSCTL\_RCGC2\_R; // allow time for clock to start

GPIO\_PORTF\_LOCK\_R = 0x4C4F434B; // 2) unlock GPIO Port F

GPIO\_PORTF\_CR\_R = 0x1F; // allow changes to PF4-0

// only PF0 needs to be unlocked, other bits can't be locked

GPIO\_PORTF\_AMSEL\_R = 0x00; // 3) disable analog on PF

GPIO\_PORTF\_PCTL\_R = 0x00000000; // 4) PCTL GPIO on PF4-0

GPIO\_PORTF\_DIR\_R = 0x0E; // 5) PF4,PF0 in, PF3-1 out

GPIO\_PORTF\_AFSEL\_R = 0x00; // 6) disable alt funct on PF7-0

GPIO\_PORTF\_PUR\_R = 0x11; // enable pull-up on PF0 and PF4

GPIO\_PORTF\_DEN\_R = 0x1F; // 7) enable digital I/O on PF4-0

GPIO\_PORTF\_IS\_R &= ~0x11; // (d) PF4,PF0 is edge-sensitive

GPIO\_PORTF\_IBE\_R &= ~0x11; // PF4,PF0 is not both edges

GPIO\_PORTF\_IEV\_R &= ~0x11; // PF4,PF0 falling edge event

GPIO\_PORTF\_ICR\_R = 0x11; // (e) clear flags 4,0

GPIO\_PORTF\_IM\_R |= 0x11; // (f) arm interrupt on PF4,PF0

NVIC\_PRI7\_R = (NVIC\_PRI7\_R&0xFF1FFFFF)|0x00400000; // (g) bits:23-21 for PORTF, set priority to 2

NVIC\_EN0\_R = 0x40000000; // (h) enable interrupt 30 in NVIC

}

//---------------------Switch\_In---------------------

// read the values of the two switches

// Input: none

// Output: 0x00,0x01,0x10,0x11 from the two switches

// 0 if no switch is pressed

// bit4 PF4 SW1 switch

// bit0 PF0 SW2 switch

unsigned long Switch\_In(void){

return (GPIO\_PORTF\_DATA\_R&0x11)^0x11;

}

//---------------------Delay10ms---------------------

// wait 10ms for switches to stop bouncing

// Input: none

// Output: none

void Delay10ms(void){unsigned long volatile time;

time = 14545; // 10msec

while(time){

time--;

}

}

The last file is for the Port E initialization and the main function.

// initial values for piano major tones:

// Assume system clock is 16MHz

const unsigned long Piano[]={

30534\*2,27211\*2,24242\*2,22923\*2,

15289\*2,13621\*2,12135\*2,11454\*2,

7645\*2,6810\*2,6067\*2,5727\*2,0

};

const unsigned long Happy\_Birthday[] = {

10204,4,10204,2,9091,4,10204,4,7645,4,8099,4,

10204,2,10204,2,9091,4,10204,4,6810,4,7645,4,

10204,2,10204,2,5102,4,6067,4,7645,4,8099,4,9091,8,

5727,2,5727,6067,4,7645,4,6810,4,7645,8,0,0

};

const unsigned long Little\_Lamb[]={

6067,4,6810,4,7645,4,6810,4,6067,4,6067,4,6067,8,

6810,4,6810,4,6810,8,6067,4,5102,4,5102,8,

6067,4,6810,4,7645,4,6810,4,6067,4,6067,4,6067,8,

6810,4,6810,4,6067,4,6067,4,6810,4,7645,8,0,0

};

const unsigned long Little\_Starts[]={

7645,4,7645,4,5102,4,5102,4,4545,4,4545,4,5102,8,5727,4,5727,4,6067,4,6067,4,6810,4,6810,4,6810,8,

5102,4,5102,4,5727,4,5727,4,6067,4,6067,4,6810,8,5102,4,5102,4,5727,4,5727,4,6067,4,6067,4,6810,8,

7645,4,7645,4,5102,4,5102,4,4545,4,4545,4,5102,8,5727,4,5727,4,6067,4,6067,4,6810,4,6810,4,7645,8,0,0

};

//unsigned long Songselect;

unsigned long mode,counter,key,input,previous,Pcounter,Pmode ;

unsigned char pressed;

//static unsigned long c\_note;

#define NUM\_VALs 64 // Assume 6-bit DAC is used, that will give 64 values for one period.

#define NUM\_NOTEs 8 // number of notes to be played repeatedly

// basic functions defined at end of startup.s

extern void DisableInterrupts(void); // Disable interrupts

extern void EnableInterrupts(void); // Enable interrupts

extern void WaitForInterrupt(void); // low power mode

void PortE\_Init(void);

int main(void){

unsigned char i,j,a;

DisableInterrupts();

SwitchLed\_Init(); // use Port F for onboard switches and LEDs

DAC\_Init(); // Port B is DAC

PortE\_Init(); // Port E is piano input PD 0-3

mode=0; //mode 0 means piano mode

counter = 0; // counter has different use in different mode,also keep the data

// on how many time the PF0 is pressed

Pmode =0;

Pcounter =0;

EnableInterrupts();

while(1){

if (mode == 0){//piano mode

key = GPIO\_PORTE\_DATA\_R;

previous = 0x00;

if(key == 0x01){

input = GPIO\_PORTE\_DATA\_R&0x01; // bit 0 means SW0 pressed

if(input&&(previous==0)){ // just pressed

a = 0 + counter\*4;

Sound\_Init(Piano[a]/NUM\_VALs); // Play 8 major notes

}

if(previous&&(input==0)){ // just released

Sound\_stop();

}

previous = input;

Delay10ms(); // remove switch bounce

}

else if(key == 0x02){

input = GPIO\_PORTE\_DATA\_R&0x02; // bit 1 means SW1 pressed

if(input&&(previous==0)){ // just pressed

a = 1 + counter\*4;

Sound\_Init(Piano[a]/NUM\_VALs); // Play 8 major notes

}

if(previous&&(input==0)){ // just released

Sound\_stop(); // stop sound

}

previous = input;

Delay10ms(); // remove switch bounce

}

else if(key == 0x04){

input = GPIO\_PORTE\_DATA\_R&0x04; // bit 2 means SW2 pressed

if(input&&(previous==0)){ // just pressed

a = 2 + counter\*4;

Sound\_Init(Piano[a]/NUM\_VALs); // Play 8 major notes

}

if(previous&&(input==0)){ // just released

Sound\_stop(); // stop sound

}

previous = input;

Delay10ms(); // remove switch bounce

}

else if(key == 0x08){

input = GPIO\_PORTE\_DATA\_R&0x08; // bit 3 means SW3 pressed

if(input&&(previous==0)){ // just pressed

a = 3 + counter\*4;

Sound\_Init(Piano[a]/NUM\_VALs); // Play 8 major notes

}

if(previous&&(input==0)){ // just released

Sound\_stop();; // stop sound

}

previous = input;

Delay10ms(); // remove switch bounce

}

else { // when none of the key was pressed

a = 12; // it's quite.

Sound\_Init(Piano[a]/NUM\_VALs);}

}

else if(mode ==1){//atuo-play mode

i = 0;

if(counter == 0 )

{

Pcounter = counter;

Pmode = mode;

while (Pcounter == counter && mode == 1){

Sound\_Init(Happy\_Birthday[i]/NUM\_VALs);

for (j=0;j<20;j++)

Delay10ms();

i = i+1;

if (i == 55){i = 0;}

}

}

else if(counter == 1 )

{

Pcounter = counter;

Pmode = mode;

while (Pcounter == counter && mode == 1){

Sound\_Init(Little\_Lamb[i]/NUM\_VALs);

for (j=0;j<20;j++)

Delay10ms();

i = i+1;

if (i == 55){i = 0;}

}

}

else if(counter == 2 )

{

Pcounter = counter;

Pmode = mode;

while (Pcounter == counter && mode == 1){

Sound\_Init(Little\_Starts[i]/NUM\_VALs);

for (j=0;j<20;j++)

Delay10ms();

i = i+1;

if (i == 55){i = 0;}

}

}

}

}

}

void PortE\_Init(){

volatile unsigned long delay;

SYSCTL\_RCGC2\_R |= 0x10; // 1) E clock

delay = SYSCTL\_RCGC2\_R; // delay

GPIO\_PORTE\_AMSEL\_R &= ~0x0F; // 3) disable analog function

GPIO\_PORTE\_PCTL\_R &= ~0x0000FFFF; // 4) GPIO clear bit PCTL

GPIO\_PORTE\_DIR\_R &= ~0x0F; // 5) PE3-0 input

GPIO\_PORTE\_AFSEL\_R &= ~0x0F; // 6) no alternate function

GPIO\_PORTE\_DEN\_R |= 0x0F; // 7) enable digital pins PE3-0

GPIO\_PORTE\_IS\_R &= ~0x0F;

GPIO\_PORTE\_IBE\_R &= ~0x0F;

GPIO\_PORTE\_IEV\_R &= ~0x0F;

GPIO\_PORTE\_ICR\_R = 0x0F;

GPIO\_PORTE\_IM\_R |= 0x0F;

NVIC\_PRI7\_R = (NVIC\_PRI7\_R&0xFFFF1FFF)|0x00040000; // (g)

NVIC\_EN0\_R = 0x40000000; // (h) enable interrupt 30 in NVIC

}

void GPIOPortF\_Handler(void) {

Delay10ms();

Delay10ms();

if(GPIO\_PORTF\_RIS\_R&0x01){// when the PF0 is pressed

GPIO\_PORTF\_ICR\_R = 0x01;// the counter goes up

if(counter ==0){ // in different mode couter has different function

Delay10ms(); // in mode 0 which is the piano mode,counter can change the tone

counter = 1; // in mode 1 which is the auto play mode, counter can change between songs.

Delay10ms();}

else if (counter ==1){

Delay10ms();

counter = 2;

Delay10ms();}

else if (counter ==2){

Delay10ms();

counter = 0;

Delay10ms();}

}

if( GPIO\_PORTF\_RIS\_R&0x10)// when the PF4 is pressed,the mode will change between 0 and 1

{ // and change between different mode.

GPIO\_PORTF\_ICR\_R |= 0x10;

if(mode == 0){mode = 1;}

else{mode =0;}

}

}