```
/*************************************************
 * File: snd.c          Project by: David Hodgdon *
 * Class: ECE476        Instructor: Bruce Land    *
 *************************************************/
// Much of this code is taken from example code
// http://instruct1.cit.cornell.edu/courses/ee476/video/Video32v3.c
#include "snd.h"
unsigned char sndClock;
//B.3 is sound  and should have a 10k resistor to gnd
//Musical note values
//C below middle C to  C above middle C
//zeros are rests
/*************************************************
 * CLK=16e6, I want to base on 16th notes        *
 * 1 16th note = 1 time unit                     *
 * So use the 1024 prescalar and count to 977    *
 * 1024*977=1000448                              *
 * ***********************************************/
#ifdef ATMEL
#include <Mega32.h>
#if 0
flash char notes[] = {239,213,189,179,159,142,126,
               120,106,94,90,80,71,63,60,0,0,0,0};
#endif
/**********************
 *notes=[261.64 277.20 293.68 311.12 329.64 349.24 370 392 415.32 440 466.16 493
.92]
 *round(16e6/256./notes')
 */
flash char notes[12]={239,
   225,
   213,
   201,
   190,
   179,
   169,
   159,
   150,
   142,
   134,
   127};
static char noteDuration;

void SndInit() {
        //use OC0 (pin B.3) for music
        DDRB.3 = 1 ;

//      OCR1A = 1018;   //One NTSC line
        OCR1A = 977;
//      TCCR1B = 9;              //full speed; clear-on-match
        TCCR1B = 0b00001101; //1024 prescalar; clear-on-match

        TCCR1A = 0x00;  //turn off pwm and oc lines
        TIMSK = 0x10;           //enable interrupt T1 cmp
//   SndScale();

}

interrupt [TIM1_COMPA] void musicTic() {
        noteDuration = (noteDuration==0) ? 0 : noteDuration-1;
        sndClock++;
}

interrupt [TIM0_COMP] void  keyPadStateMachine() {

}


void SndPlayNote(int note, char time) {
```



```
        while(noteDuration!=0); // wait for previous note to finish
        TCCR0 = 0;
        if(note>=0)
        {
                switch(note/12) {
                        case 0:
                                TCCR0 = 0b00011101; // 1024 prescalar
                                OCR0=notes[note%12];
                                break;
                        case 1:
                                TCCR0 = 0b00011101; // 1024 prescalar
                                OCR0=notes[note%12]>>1;
                                break;
                        case 2:
                                TCCR0 = 0b00011100; // 256  prescalar
                                OCR0=notes[note%12];
                                break;
                        case 3:
                                TCCR0 = 0b00011100; // 256  prescalar
                                OCR0=notes[note%12]>>1;
                                break;
                        case 4:
                                TCCR0 = 0b00011011; // 64  prescalar
                                OCR0=notes[note%12];
                                break;
                        case 5:
                                TCCR0 = 0b00011011; // 64  prescalar
                                OCR0=notes[note%12]>>1;
                                break;
                        case 6:
                                TCCR0 = 0b00011011; // 64  prescalar
                                OCR0=notes[note%12]>>2;
                                break;
                        case 7:
                                TCCR0 = 0b00011010; // 8   prescalar
                                OCR0=notes[note%12];
                                break;
                        case 8:
                                TCCR0 = 0b00011010; // 8  prescalar
                                OCR0=notes[note%12]>>1;
                                break;
                        case 9:
                                TCCR0 = 0b00011010; // 8  prescalar
                                OCR0=notes[note%12]>>2;
                                break;
                        case 10:
                                TCCR0 = 0b00011001; // no   prescalar
                                OCR0=notes[note%12];
                                break;
                        case 11:
                                TCCR0 = 0b00011001; // no  prescalar
                                OCR0=notes[note%12]>>1;
                                break;
                        case 12:
                                TCCR0 = 0b00011001; // no  prescalar
                                OCR0=notes[note%12]>>2;
                                break;
                        default:
                                sprintf(lcd_buffer, "default");
                                LCDPrintString(lcd_buffer);
                                break;
                }
        }
        noteDuration=time;
}


// play scales
```

```
void SndScale() {
        char i,j;
        for(j=1;j<2;j*=2) {
                for(i=0;i<16;i++) {
                        SndPlayNote(i, j);
                }
        }
}

#if 0
        //Play a note
        //TCCR0 bits:
        //bit 7 no force:
        //bit 3 CTC on:
        //bit 5,4 toggle OC0 pin
        //bit 6 no PWM
        //bit 2-0 prescaler 256 so 1 tick is 16 microsec
        if((musicT++) > 15)        //each note 1/4 second
        {
                musicT = 0;
                TCCR0 = 0;
                if (notes[note]>0) TCCR0 = 0b00011100 ;  //not a rest
                OCR0 = notes[note++] ;
                if (note>18) note = 0;        //test for end of scale
        }
#endif
#else // not ATMEL
#include <stdio.h>
void SndPlayNote(int note, char time) {
        char str[6];
//      printf("SndPlayNote(%d, %d)\n", note, time);
        switch(note%12) {
                case 0:strcpy(str,"C");break;
                case 1:strcpy(str,"C#");break;
                case 2:strcpy(str,"D");break;
                case 3:strcpy(str,"D#");break;
                case 4:strcpy(str,"E");break;
                case 5:strcpy(str,"F");break;
                case 6:strcpy(str,"F#");break;
                case 7:strcpy(str,"G");break;
                case 8:strcpy(str,"G#");break;
                case 9:strcpy(str,"A");break;
                case 10:strcpy(str,"A#");break;
                case 11:strcpy(str,"B");break;
        }
        printf(str);
        printf("%d(%d), ", note/12, time);

}
// play scales
void SndScale() {
        char i,j;
        for(j=1;j<16;j++) {
                for(i=0;i<15;i++) {
                        SndPlayNote(i, j);
                }
        }
}


#endif
```