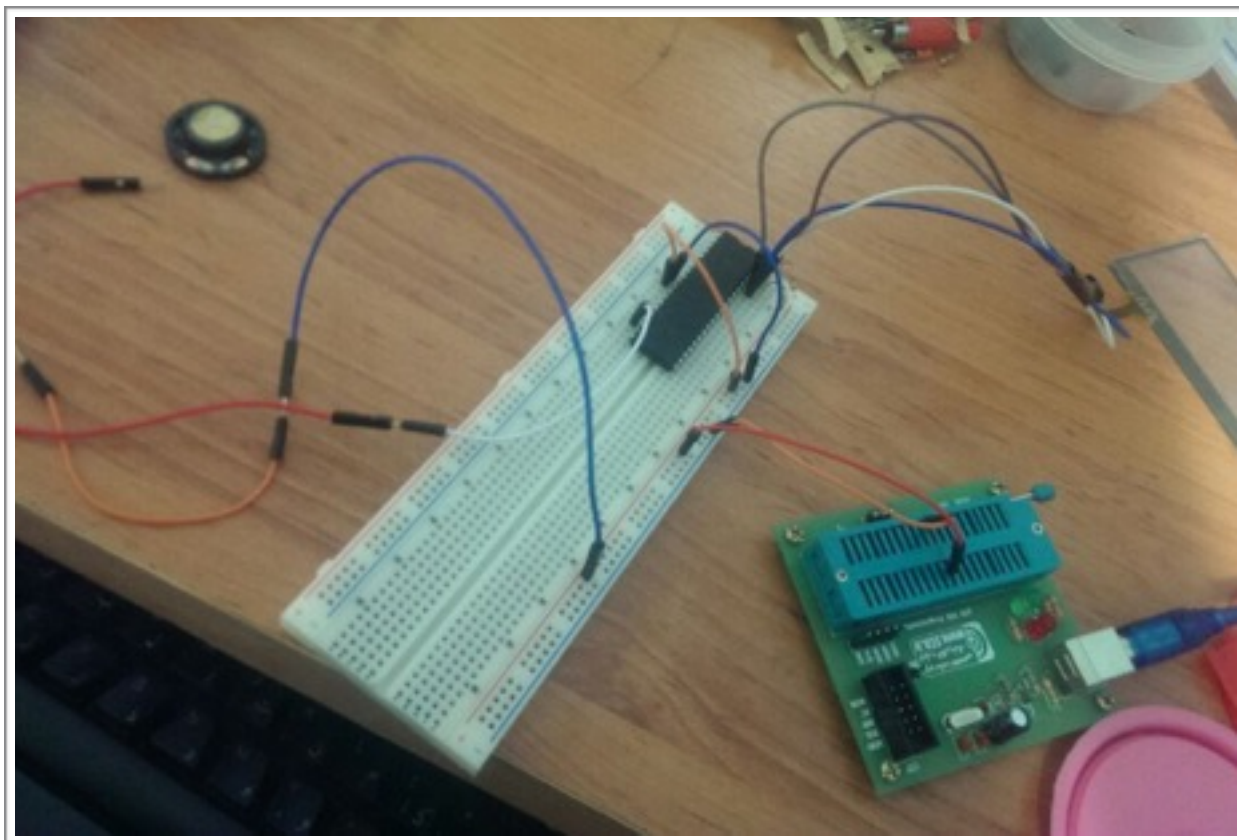


مستندات پروژه میکرو

تولید صدا با استفاده از صفحه ی لمسی



علی رحیمی

۹۲۲۱۱۷۳

ترم ۳۹۴۲

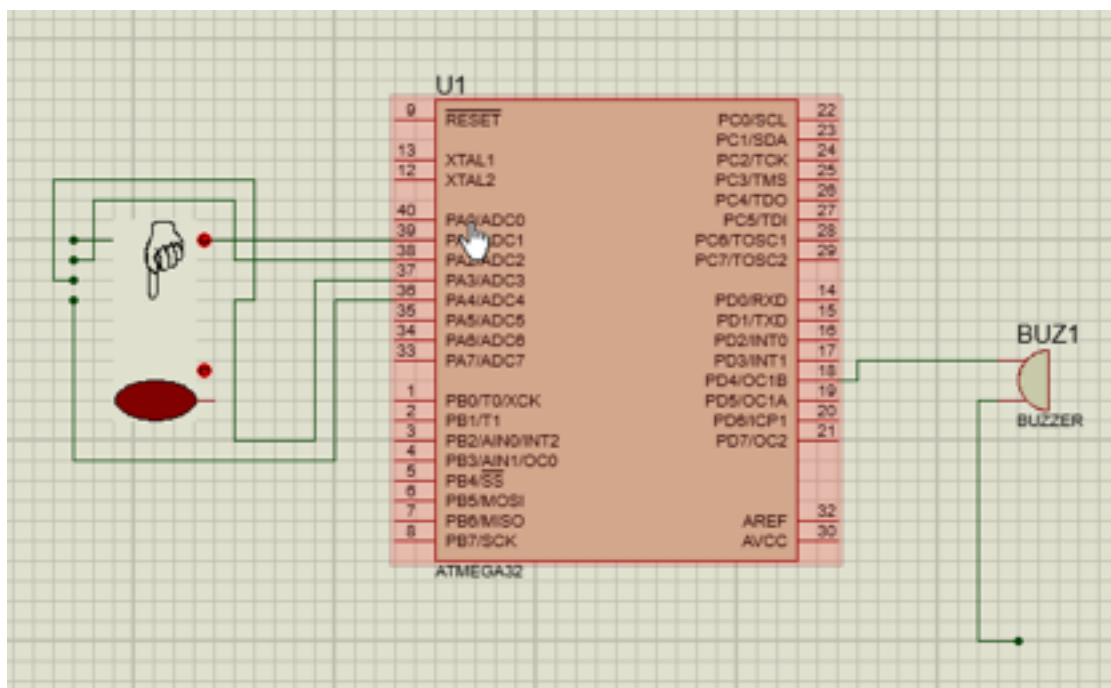
مقدمه

هدف پروژه تولید نت های موسیقی با رابط لمسی می باشد . به این صورت که نوت های موسیقی توسط تایمر ۱ موجود در میکرو در حالت تولید موج های مربعی تولید شده و توسط پورت d4 به خروجی صدا انتقال می یابد . استفاده لمسی در این پروژه به این صورت است که در صورت لمس صدایی توسط خروجی تولید شود .

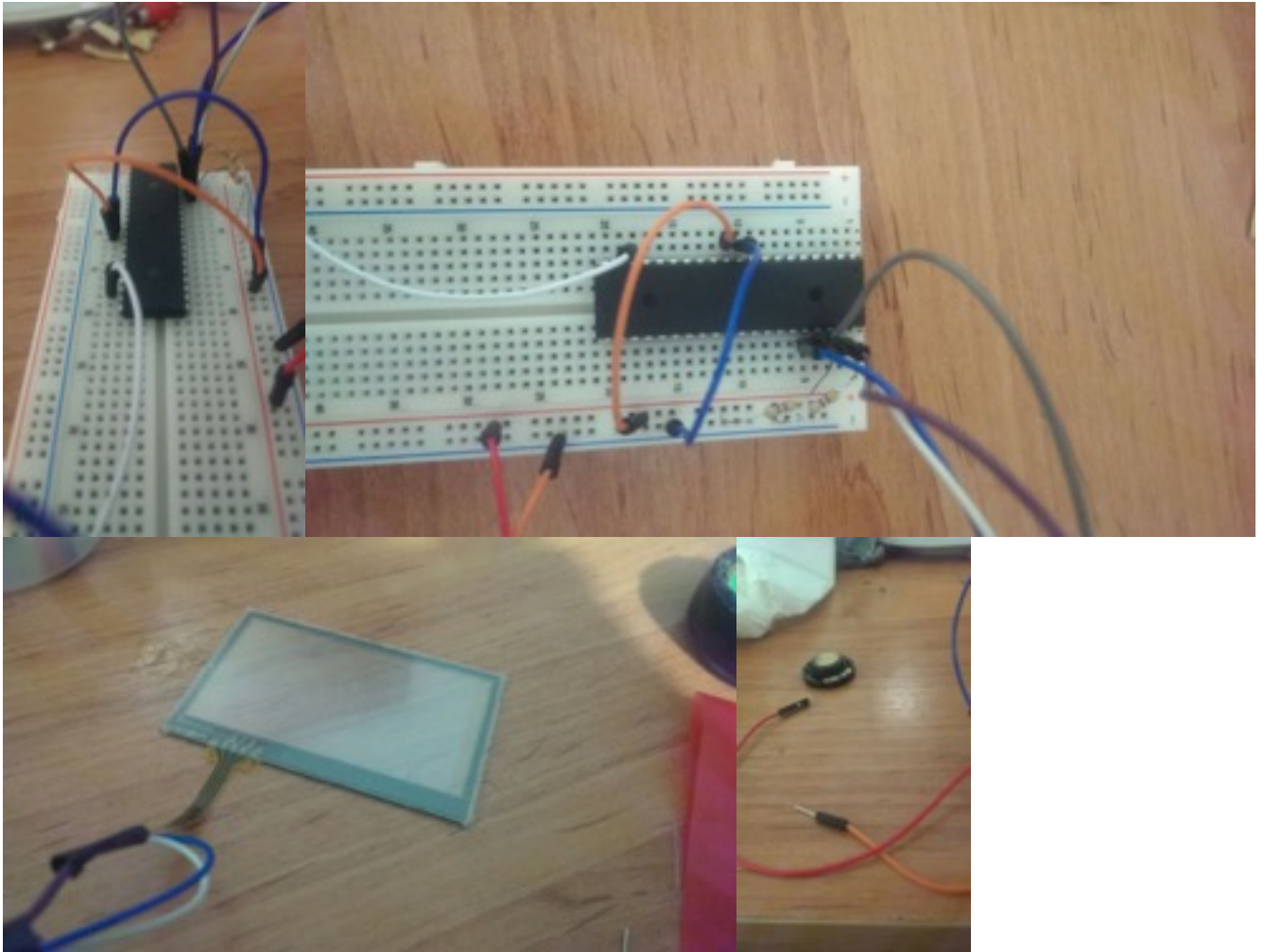
جدول قطعات استفاده شده

نام قطعه	شماره	قیمت
AVR Micro Controller	ATMEGA32-A	7T
Touch Screen 64 * 128	LGM12864B	13T
Speaker	Normal 8Ohm Speaker	2T

شماتیک :



عکس از مدار :



توضیح مدار :

پورت D4 که خروجی تولید موج تایمر ۱ می باشد به یک سر اسپیکر وصل می باشد . یک سر دیگر آن به زمین .

تاچ اسکرین داره ۴ پورت YL , YH , XL , XH می باشد . به دلیل انالوگ بودن این قطعه . به پورت A وصل می باشد . برای خواندن X باید باید XL و XH به صورت خروجی قرار گیرند (PINA در این پورت ها ۱ شود) سپس Pull Up را در XH فعال کرد . سپس با استفاده از ADC مقدار X را خواند . Y نیز به همین شکل می باشد.

۲ پورت اول نیز به وسیله مقاومت به زمین وصل می باشند (XH , YH).

توضیحات کد:

Notes.h

این هدر فایل شامل اعدادی می باشد که در صورت قرار گرفتن در تایمر ۱ . موج خاص برای نوت ها را تولید میکند

```
uint16_t getX(){
    DDRA=((1<<x1)|(1<<x2)); // setting
x1 and x2 output
    PORTA=((0<<x2)|(1<<x1)); // setting
default 1 to XH
    _delay_ms(10); // delay for touch

    return readADC(3); // reading x
from adc
}
```

گرفتن X : همانطور که در بالا گفته شد .
ابتدا XL و XH را خروجی گذاشته .
سپس XH را ۱ کرده . مدتی صبر کرده
تا صفحه لمسی به روز شود . بعد با
استفاده از Analog to Digital
Converter گفته شده در کتاب . خوانده
خواهد شد .

<pre> (uint16_t readADC(uint8_t ch { // setting channel (3 for x , 4 for y) ch &= 0b00000111; ADMUX &= 0b11100000; ADMUX = ch; ADCSRA = (1<<ADSC); while(!(ADCSRA & (1<<ADIF))); // wait for interrupt flag ADCSRA = (1<<ADIF); // invert interrupt flag return(ADC); } void InitMusic() { // Configure OC1B pin as output DDRD = 1 << (DDD4); //OC1B as output // timer1 configuration (for PWM) TCCR1A = 1 << (COM1B1); // Clear OC1A/OC1B on compare match TCCR1B = 1<< (WGM13) //mode 8, PWM, Phase and Frequency Correct (TOP value is ICR1) 1 << (CS11); // prescaler(8) } </pre>	<p>خواندن از ADC</p> <p>ابتدا channel برای مشخص کردن X یا Y بودن تعیین شده . سپس بعد از فعال کردن فلگ ADSC در ADSCRA منتظر رسیدن داده میشویم .</p> <p>بار گذاری اولیه تایمر ۱ :</p> <p>ابتدا PORTD4 را خروجی قرار داده سپس فلگ برای PWM را در TCCR1A ست کرده . سپس مود WGM13 را انتخاب کرده . را نیز ۸ انتخاب میکنیم . (نوت ها بر اساس این prescale نوشته شده اند)</p>
---	--

پلی کردن نت :

```
void PlayMusic( int* pMusicNotes )
{
    int duration;
    int note;
    int i;
    uint16_t delay = 20 * 1000;

    while( *pMusicNotes )
    {
        note = *pMusicNotes;
        pMusicNotes++;

        duration = *pMusicNotes;
        pMusicNotes++;

        if( p == note )
        {
            //pause, do not generate any sound
            OCR1B = 0;
        }
        else
        {
            //not a pause, generate tone
            OCR1B = DEFAULT_VOLUME;

            //set frequency
            ICR1H = (note >> 8);
            ICR1L = note;
        }

        //wait duration
        for(i=0;i<32-duration;i++)
        {
            _delay_loop_2( delay );
        }

        //turn off any sound
        OCR1B = 0;
    }
}
```

این تابع . آرایه ای از نت میگیرد .
و آن ها را پخش میکند .

پخش کردن به این صورت می باشد
که عدد نت را در ICR1 قرار داده .

آخر هر آرایه MUSIC_END

قرار میدهیم به این علت که

ORC1B را صفر کند . در نتیجه
صدایی تولید نشود .

برای هر نت . صبری قرار میگیرد تا
این نت پخش شود .

```
const int map[6][7] = {
    {g2,f2,e2,d2,c2,b2,a2},
    {g3,f3,e3,d3,c3,b3,a3},
    {g4,f4,e4,d4,c4,b4,a4},
    {g5,f5,e5,d5,c5,b5,a5},
    {g6,f6,e6,d6,c6,b6,a6},
    {g7,f7,e7,d7,c7,b7,a7}
};
```

```
main()
{
    InitMusic();
    setADC();
    while(1)
    {

        uint16_t x = getY();
        uint16_t y = getX();

        if(x == 0 || y == 0 )
            continue;

        int indx = (x/256.0) * 8;
        int indy = (y/256.0) * 6;

        if(indy == 6 || indx == 8)
            continue;

        if(indx < 7){
            int temp[3] = { map[indy][indx] , 8 , MUSIC_END};
            PlayMusic(temp );
        }

    }

}
```

یک map ای از نت ها
تبعین شده هست . به
این صورت که در هر
کجا از صفحه ی لسی
خورد . یک نوت پلی
شود .

مقدار دهی برای
ADC و Timer1

تغییر مقدار به ایندکس
برای مپ

چک کردن ایندکس .
اگر خراب باشد .
اتفاقی غیر قابل پیش
بینی رخ خواهد داد .

مشکلات :

ابتدا پروژه صفحه ی لمسی بر روی LCD بود . به اشتباه این جانب این قطعه سوخت و مجبور شدم پروژه را تغییر دهم . اشتباه وصل کردن ۵ولت به پین هایی که ۳ بیشتر نباید بگیرند . مشکل بعدی ابتدا استفاده از تایمر ۰ بود . که به اندازه ی کافی نمیتوانست نت ها را پوشش دهد . با استفاده از تایمر ۱ این مشکل حل شد .

مشکل بعدی استفاده در وصل کردن صفحه ی لمسی بود . که با استفاده از مقاومت بر طرف شد . در صورتی که چک کردن ایندکس نبود . انتقافی غیر قابل پیش بینی (به علت خواندن از محدوده فرا تر از ارای ی map .) در صورت تاچ در نقاط اخر پیش می آمد . که با گذاشتن شروط این مشکل بر طرف شد .

منابع :

کتاب AVR MAZIDI

CHAPTER 13: ADC

CHAPTER 15: INPUT CAPTURE AND WAVE GENERATION IN AVR

AVR341: Four and five-wire Touch Screen Controller : <http://www.atmel.com/Images/doc8091.pdf>

<http://www.engineersgarage.com/contribution/expert/interfacing-4-wire-resistive-touchscreen-with-atmega-16-microcontroller>

فرکانس های استفاده شد :

<http://www.seventhstring.com/resources/notefrequencies.html>

کاتالوگ تاچ اسکرین :

[/http://roboeq.ir/files/id/2136/name/TouchSchreen_LGM12864B.pdf](http://roboeq.ir/files/id/2136/name/TouchSchreen_LGM12864B.pdf)