هو العلیم

درس آزمایشگاه ریزپردازنده

نیم‌سال دوم ۹۹-۹۸

دانشکده برق

شبیه‌سازی موتور DC

تهیه و تنظیم: حسن رضائی‌نسب - ۹۶۲۲۷۴۳

در این آزمایش به کمک میکرو و آی سی اینکدور، چرخش و سرعت موتور dc را کنترل می کنیم و آن را اندازه می گیریم و روی صفحه نمایش گر LCD نمایش می دهیم. در این آزمایش کنترل سرعت موتور را به کمک pwm انجام می دهیم.

آی سی اینکدور(قطعه واسط برای ایجاد جریان مورد نیاز) استفاده شده L298 است.

L298 قطعه ای جهت راه اندازی موتور است که با توجه به جریان دهی مناسب می تواند نیاز ما را برطرف کند. پایه های موتور را به OUT1-OUT4 متصل می کنیم. پایه های SENSA و SENSB برای تنظیم جریان موتور است که به زمین متصل می کنیم. پایه های VCC و VSS برای تغذیه L298 و موتور ها هستند که به ترتیب به ولتاژ ۵ ولت و ۱۲ ولت متصل می کنیم. پایه های ENB و ENA برای تنظیم سرعت موتور ها هستند که به پایه های OCR1A و OCR1B میکرو متصل می کنیم و پایه های IN1 تا IN4 برای تعیین جهت چرخش موتور ها هستند که در این آزمایش نیازی به آن ها نداریم.

تنظیمات لازم را به کمک ویزارد انجام می دهیم.

فرکانس کاری میکرو را ۱ مگا هرتز تعیین می کنیم. پورت B را برای LCD انتخاب می کنیم.

پین های ۰ و ۱ از پورت C و پین ۷ از پورت D که برای تولید موج pwm می باشد را خروجی می کنیم.

برای اندازه گیری ولتاژ آنالوگی که به ADC3 توسط پتانسیوتر اعمال می شود، موج PWM متناسب با آن را تولید می کنیم.پس ADC میکرو را هم فعال کنیم.

و در نهایت از تایمر ۲ یا OC2 برای ایجاد موج pwm استفاده می کنیم که تنظیمات ویزارد برای فعال شدن آن را انجام می دهیم. مقدار Clock value را ۱۲۵ کیلوهرتز قرار می دهیم.

متن کد برنامه به صورت زیر است:

```
#include <mega32.h>
#include <delay.h>
#include <stdlib.h>
#include <alcd.h>

//#define ADC_VREF_TYPE 0xc0
#define ADC_VREF_TYPE ((0<<REFS1) | (0<<REFS0) | (1<<ADLAR))
```

```
unsigned char read_adc(unsigned char adc_input)
{
ADMUX=adc_input | ADC_VREF_TYPE;
// Delay needed for the stabilization of the ADC input voltage
delay_us(20);
// Start the AD conversion
ADCSRA|=(1<<ADSC);
// Wait for the AD conversion to complete
while ((ADCSRA & (1<<ADIF))==0);
ADCSRA|=(1<<ADIF);
return ADCH;
}

void main(void)
{
  char text[40];
  int start;

  DDRA=0xf6;
  PORTA=0;
  DDRC=0x03;
  PORTC=0;
  DDRD=0x40;
  PORTD=0;

  lcd_init(16);

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: 125.000 kHz
```

```
// Mode: Phase correct PWM top=0xFF
// OC2 output: Non-Inverted PWM
// Timer Period: 4.08 ms
// Output Pulse(s):
// OC2 Period: 4.08 ms Width: 0 us
 ASSR=0<<AS2;
 TCCR2=(1<<PWM2) | (1<<COM21) | (0<<COM20) | (0<<CTC2) | (1<<CS22) |
(0<<CS21) | (0<<CS20);
 TCNT2=0;
 OCR2=0;


// Analog Comparator initialization
// Analog Comparator: Off
// The Analog Comparator's positive input is
// connected to the AIN0 pin
// The Analog Comparator's negative input is
// connected to the AIN1 pin
 ACSR=(1<<ACD) | (0<<ACBG) | (0<<ACO) | (0<<ACI) | (0<<ACIE) |
(0<<ACIC) | (0<<ACIS1) | (0<<ACIS0);


// ADC initialization
// ADC Clock frequency: 500.000 kHz
// ADC Voltage Reference: AREF pin
// ADC Auto Trigger Source: ADC Stopped
// Only the 8 most significant bits of
// the AD conversion result are used
 ADMUX=ADC_VREF_TYPE;
 ADCSRA=(1<<ADEN) | (0<<ADSC) | (0<<ADATE) | (0<<ADIF) | (0<<ADIE) |
(1<<ADPS2) | (0<<ADPS1) | (0<<ADPS0);
 SFIOR=(0<<ADTS2) | (0<<ADTS1) | (0<<ADTS0);
```

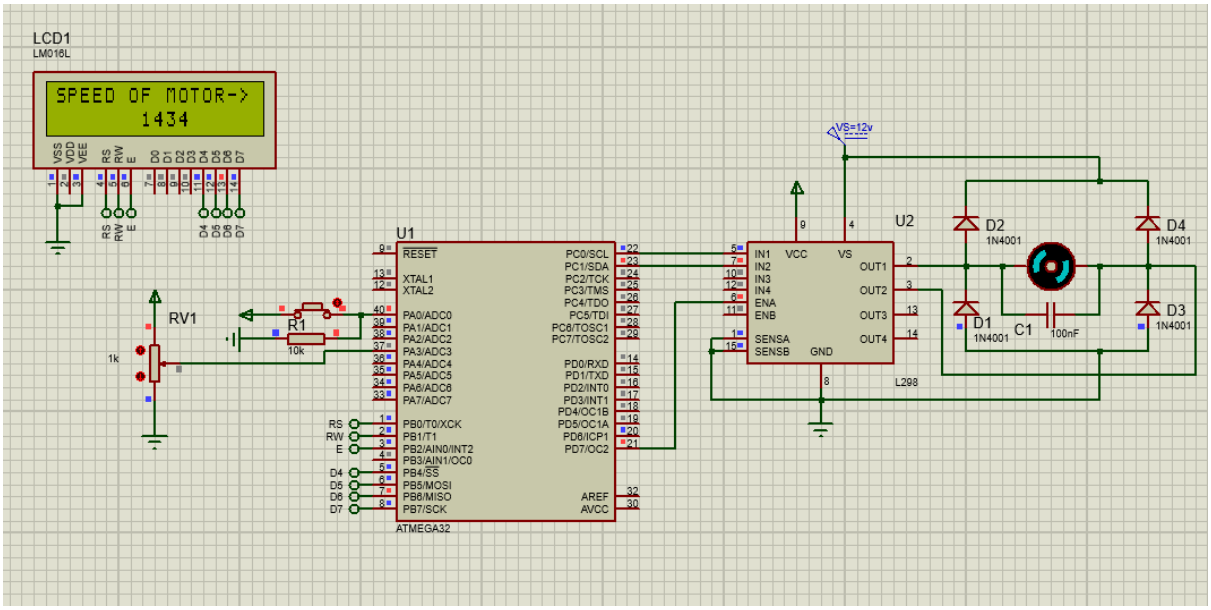
```
 OCR2=0;
while (1)
    {
      OCR2=read_adc(0);.
      if(PINA.0==1)
        {
         PORTC.0=0;
         PORTC.1=1;

         lcd_clear();
         lcd_gotoxy(0,0);

         start=read_adc(0)*5.625;
         lcd_putsf("speed of motor->");
         itoa(start,text);
         lcd_gotoxy(0,1);
         lcd_putsf("        ");
         lcd_puts(text);
         delay_ms(700);
        }
};
}
```

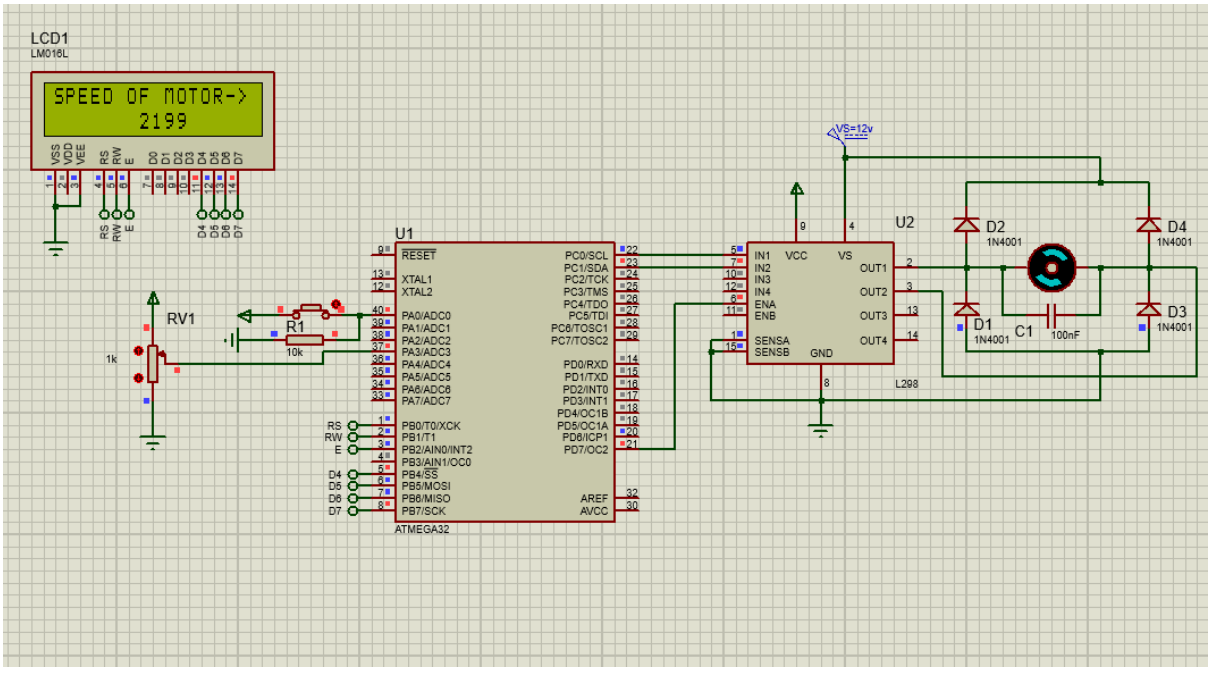
تصویری از شبیه سازی به صورت زیر است.

وقتی کلید بسته است موتور شروع به چرخش می کند و سرعت آن روی صفحه نمایش گر نشان داده می شود. هنگامی که کلید باز است موتور حرکتی ندارد و اطلاعاتی رو صفحه نمایش گر نشان داده نمی شود.



به کمک پتانسیومتر قرار داده شده سرعت موتور را کم و زیاد می کنیم.

برای مثال با فشردن+ پتانسیومتر سرعت چرخش زیاد می شود.



و با فشردن پتانسیومتر سرعت چرخش کم می شود.