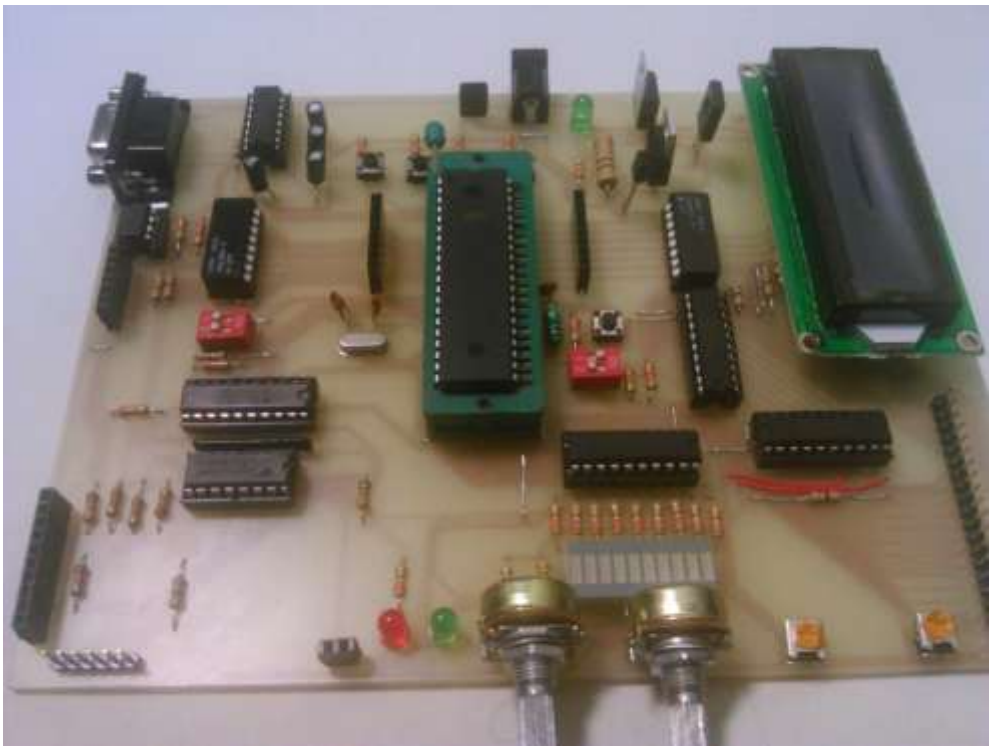
پروژه برد آموزشی میکروکنترلرهای AVR



مقدمه:

امروزه در اکثر پروژه های الکترونیکی میکروکنترلر ها نقش اصلی پروژه را ایفا می کنند که در این میان خانواده ی AVR به دلیل امکانات بالا، تنوع میکروکنترلرها، قیمت مناسب و... از محبوبیت بالایی برخوردار است پس چه بهتر که در دوره ی دانشجویی بتوان به خوبی با تمام امکانات این میکروکنترلر اشنا شد و در بازار حرفه ای کار از ان بهره گرفت، همچنین درس پروژه ساخت باعث می شود دانشجو تمام انچه که در دوره ی کاردانی به صورت تئوری فرا گرفته است را به عمل تبدیل کند پس بهتر است پروژه ی انتخابی برای درس پروژه ساخت در بازار حرفه ای کار نیز کاربرد داشته باشد که با این وجود ***برد اموزشی AVR*** باعث شناخت هرچه بیشتر امکانات این خانواده از میکروکنترلر شده و باعث می شود در بازارکار بتوان از ان بهره گرفت و راحت تر پروژه های الکترونیکی را پیاده سازی نمود.

اهمیت استفاده از برد آموزشی :

یکی از سریعترین راه های یادگیری میکروکنترلرها استفاده از بردهای آموزشی بوده که شما می توانید بدون اتلاف وقت جهت راه اندازی بخش سخت افزار بر روی برنامه نویسی تمرکز کرده و سریعاً با قسمت های مختلف میکروکنترلر آشنا شده و برنامه های خود را طراحی نمایید. همچنین می توانید بعد از تکمیل برنامه با الگو برداری از سخت افزار این برد آموزشی سخت افزارهای خود را پیاده سازی نمایید

امکانات روی برد های آموزشی:

1-دارای میکروکنترلر قدرتمند ATMEGA32

2-نمایشگر 2*16 کاراکتری

3-نمایشگر 128*64 گرافیکی

4-راه انداز استپ موتور

5- پورت سریال rs232

6-سنسور دما

7-مقایسه کننده آنالوگ با خروجی های مجزا

8-دارای 8 عدد LED

9- 4 عدد شستی

10-راه انداز موتور DCبا PWM

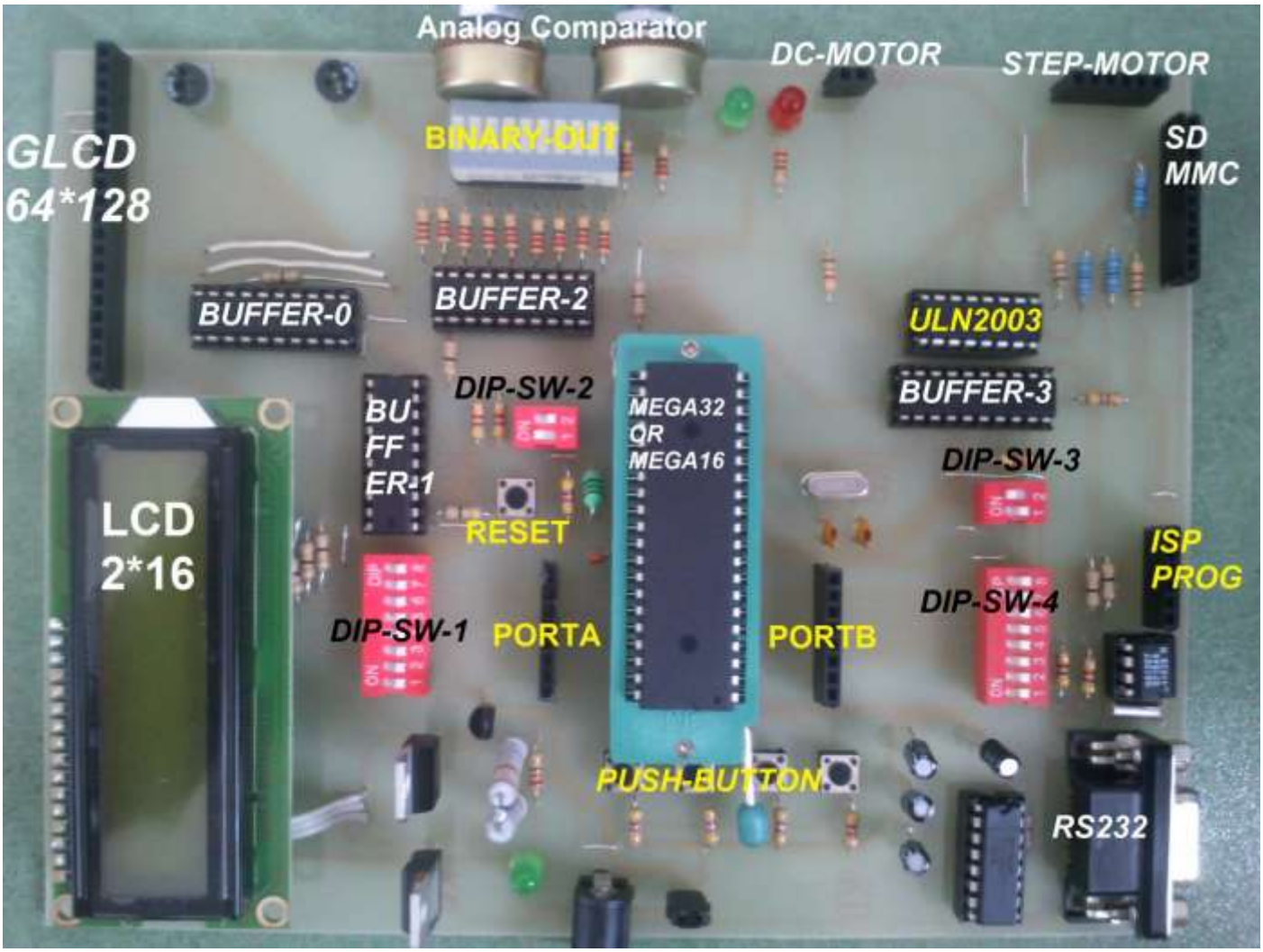
11-دارای حافظه SD

12-دارای آی سی ساعات DS1307

13- دارای خروجی های مجزا برای هر پورت

14-قابلیت پروگرام شدن روی سیستم (ISP)

15- وغیره



راه اندازی قسمت های مختلف برد آموزشی :

برای راه اندازی قسمت های مختلف برد باید با دقت شماتیک برد را برسی کرد .

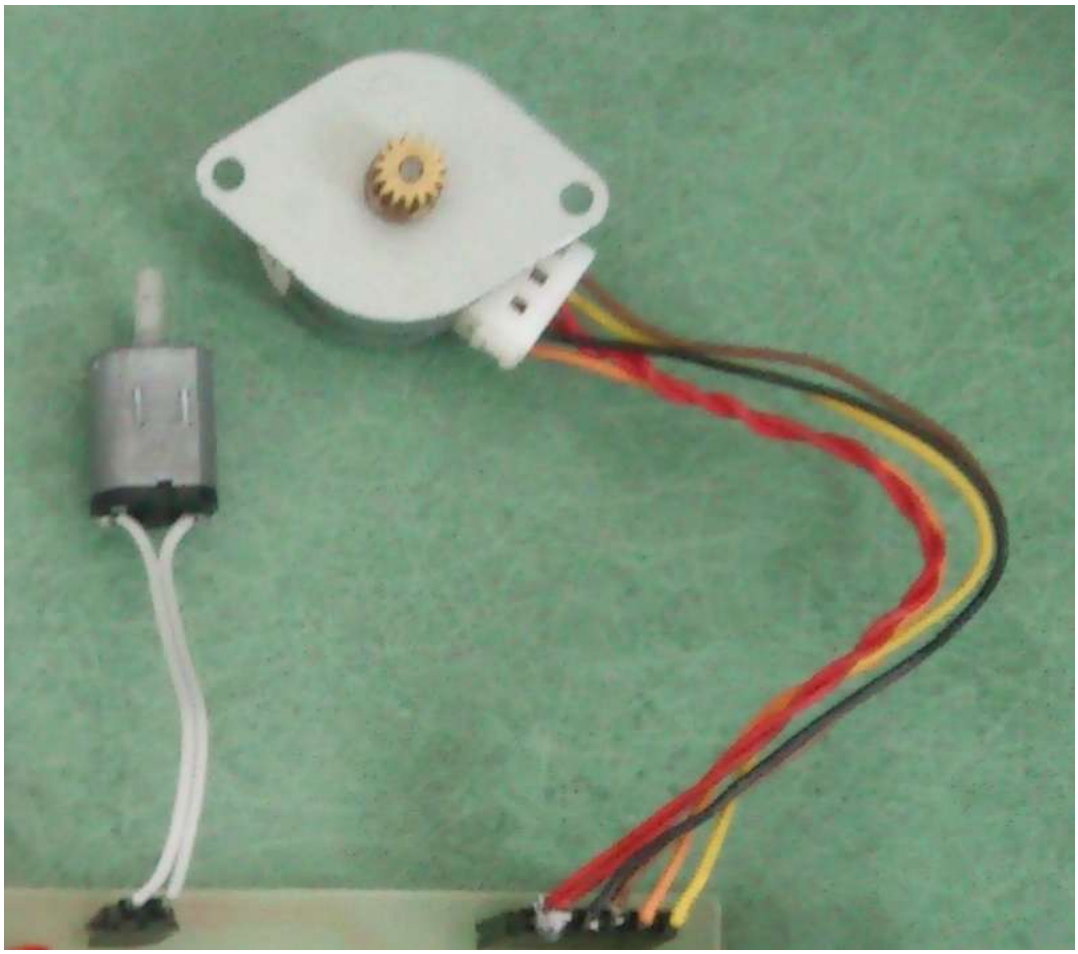
راه اندازی موتور پله ای (STEP-MOTOR):

اجرای پر مصرف برد در حالتی که استفاده نمی شوند بهتر است روی برد نباشند . موتور پله ای هم یکی از این قطعات است پس برای راه اندازی اول موتور را بشکل صحیح روی برد نصب کنید بعد با توجه به شماتیک برنامه ای را می

```
#include <mega32a.h>
#include <delay.h>
void main()
{
unsigned char shift;
DDRD=0X17;
while(1)
  {
    for(shift=1;shift<8;shift<<=1)
      {
        PORTD=shift;
        delay_ms(10);
      }
    PORTD.4=1;      delay_ms(10);
    PORTD.4=0;      delay_ms(10);
  }
}
```

برای استفاده از موتورها باید بافر متناظر با آن یعنی بافر 3 باید فعال باشد و بافر 2 خاموش .

DIP-SW-3 برای اینکار است.



راه اندازی LCD کاراکتری :

برای راه اندازی LCD در نرم افزار کد ویژن بصورت زیر پایه های LCD را تعریف می کنیم :

بعد با توجه به نیاز از توابع کتابخانه LCDاستفاده می کنیم.

هدر زیر را برای استفاده از توابع این کتابخانه باید فراخوانی کنیم .

#include <alcd.h>

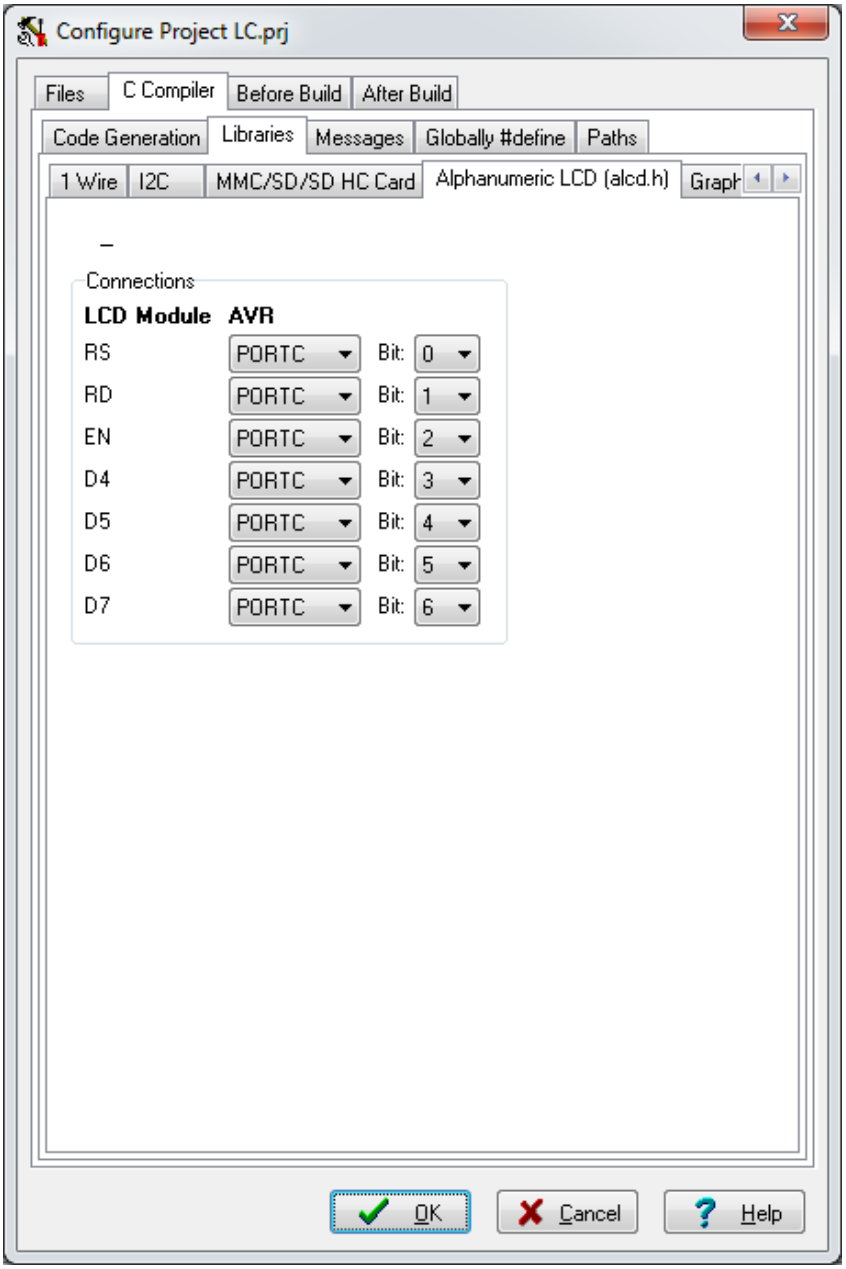
```
#include <mega32.h>
#include <alcd.h>
#include <delay.h>
#include <stdio.h>
```

IR

برای استفاده از LCD باید بافر 1 فعال و بافر 0 خاموش شود .DIP-SW-2

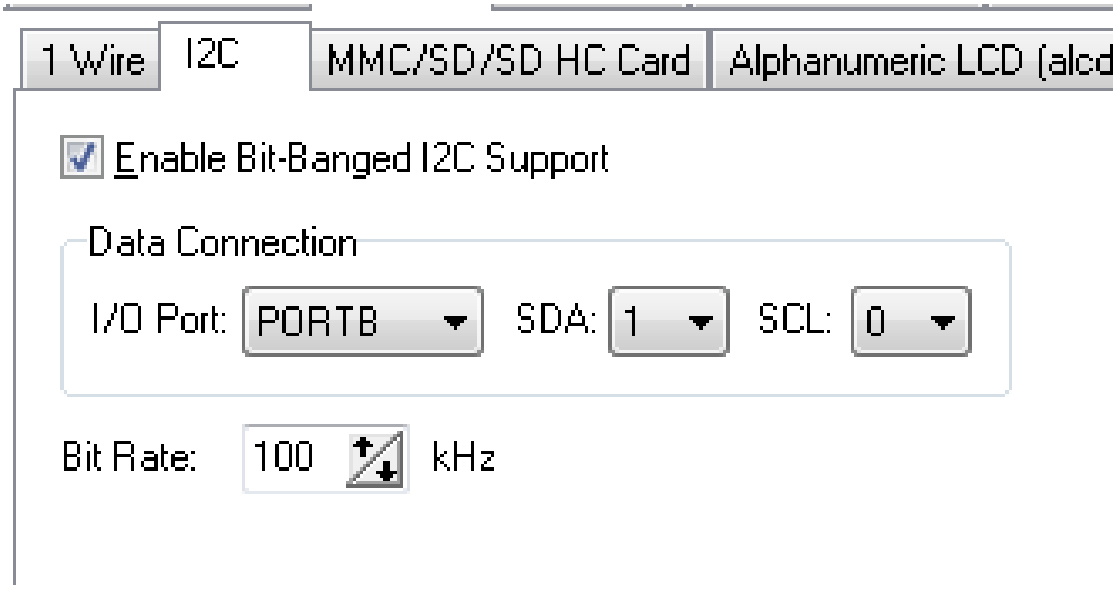
برای اینکار است .

```
void main(void){
lcd_init(16);
lcd_clear();
   lcd_putsf("* Board avr v1 * ");
   lcd_gotoxy(0,1);
   delay_ms(1);
   lcd_putsf(" *ATMEGA 32&16*");
while(1);
}
```



راه اندازی قسمت i2c میکرو (آی سی ساعت DS1307):

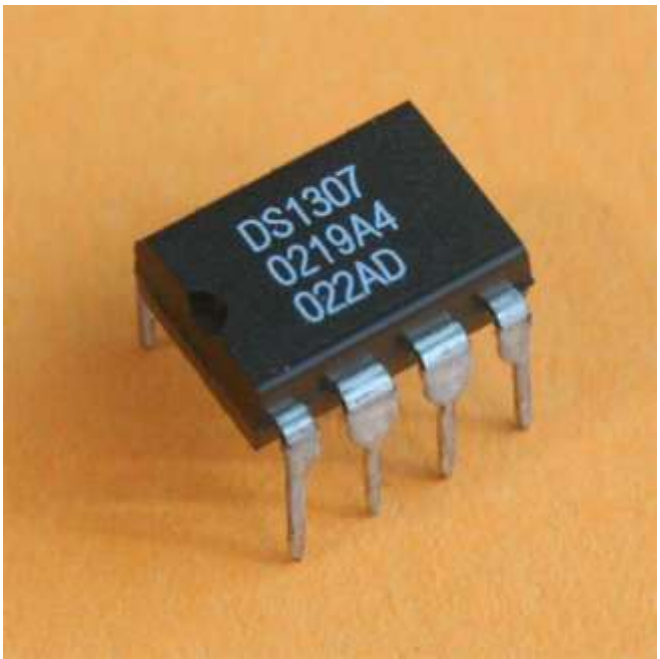
برای فعال سازی و انتخاب پایه های میکرو برای I2Cم مثل شکل زیر عمل می کنیم :



دوتا پایه 0 و 1 پورت B از DIP-SW-4 عبور می کند که باید فعال شوند.

بعد طبق روال های قبلی هدر های مورد نیاز را فراخوانی می کنیم

تنظیمات اولیه ساعت را انجام می دهیم و...



```
#include <mega32a.h>
#include <i2c.h>
#include <ds1307.h>
#include <alcd.h>
#include <stdio.h>
#include <delay.h>
char display_buffer[17]; /* LCD display buffer for 1 line */
void main(void)
{
unsigned char hour,min,sec,week_day,day,month,year;
DDRA&=0XE1;
PORTA|=0X1E;
lcd_init(16);
i2c_init();
rtc_init(0,0,0);
rtc_set_time(12,0,0);
rtc_set_date(2,1,2,11);
while (1)
    {
    rtc_get_time(&hour,&min,&sec);
    rtc_get_date(&week_day,&day,&month,&year);
    sprintf(display_buffer,"Time: %2d:%02d:%02d\n",hour,min,sec);
    lcd_puts(display_buffer);
    sprintf(display_buffer,"Date: %2d/%02d/%d",day,month,2000+year);
    lcd_puts(display_buffer);
    delay_ms(50);
    lcd_clear();
    }
}
```

فعال سازی قسمت مقایسه کننده :

```
#include <mega32a.h>

#include <delay.h>

#define low_level 5

#define high_level 6

void main()

{

ACSR=0X00;

DDRD|=((1<<low_level)|(1<<high_level));

while(1)

  {

    if(ACSR & 0x20)

      {

        PORTD.high_level=1;

        PORTD.low_level=0;

      }

    else

      {

        PORTD.high_level=0;

        PORTD.low_level=1;

      }     }   }
```

فعال سازی GLCD برد :

برای فعال سازی ال سی دی گرافیکی از یک هدر فایلی استفاده می کنیم که برای استفاده باید در هدر آن پایه ها بصورت زیر تغییر کنند

```
#define     RS          PORTA.1
#define     RW          PORTA.2
#define     EN          PORTA.3
 #define    CS1   PORTA.4
#define     CS2   PORTA.5
#define     RST         PORTA.6
#define     DATA        PORTC
```

برای استفاده از GLCD باید بافر 0 فعال و بافر 1 خاموش شود .DIP-SW-2

برای اینکار است .

همچنین سوئیچ سر راه آن یعنی DIP-SW-1باید روشن باشد.

کد برنامه GLCD

```
#include <mega32.h>
#include <delay.h>
#include <ks0108.h>
void init();
void main(void)
{
      unsigned char str[] = "E I T T C";
    init();
    PORTA.6=1;
    glcd_Init();
            glcd_DrawF(pat1);
            delay_ms(500);
            glcd_DrawF(pat2);
            delay_ms(500);
      glcd_Clear();
      glcd_Printf(1,40,str);
    while (1)
      {glcd_DrawF(pat2);
            delay_ms(500);};
}
```

راه اندازی ارتباط سریال برد آموزشی

همانطور که می دانید برای تطبیق ولتاژ بین پورت سریال کامپیوتر و میکرو باید از تطبیق دهنده ولتاژ استفاده کرد که max232 یکی از این آی سی ها است.

برای راه اندازی ابتدا باید رجیستر های USART را تنظیم کنیم . برای این کار طبق روال زیر دنبال می کنیم:

تنظیم رجیسترهای میکرو

برای مد فرستنده

```
UCSRB=0x08; //Enable TxD PIN
  UCSRC=0x86; //Active USART
  UBRRH=0x00; //baud=9600
  UBRRL=0x19; //baud=9600
```

برای اینکه بتوانیم راحتر اطلاعات را ارسال کنیم از کتابخانه STDIO استفاده می کنیم

```
#include <stdio.h>
```

بعد از انجام این مراحل با تابع زیر می توانیم داده ارسال و دریافت کنیم :

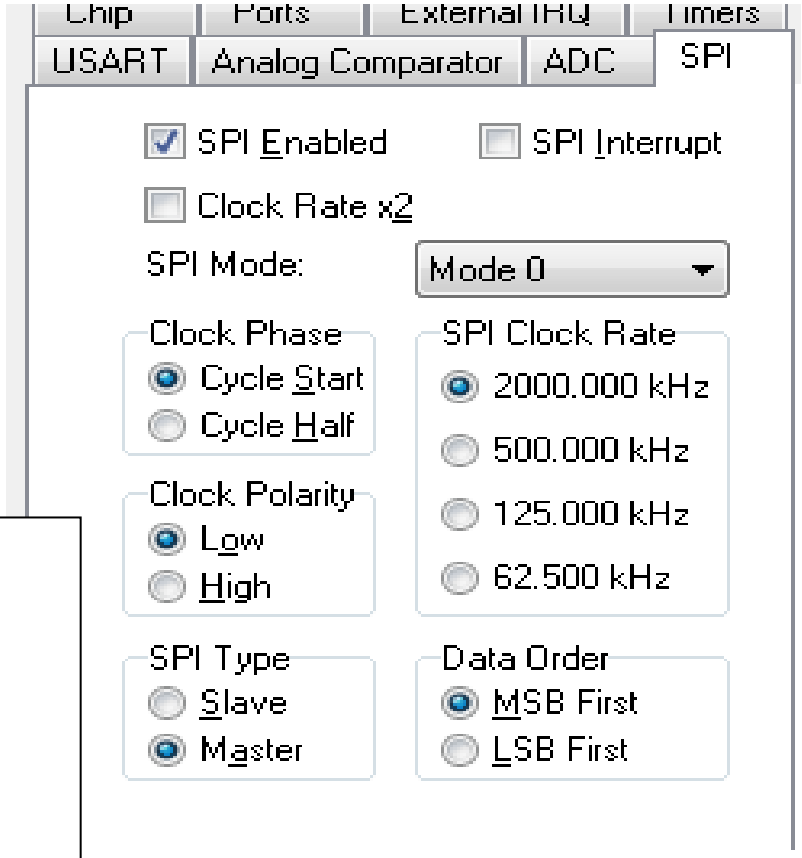
```
A=getchar()
Putchar(A);
```

راه اندازی پروتکل SPI برد آموزشی

از پروتکل spi برای ارتباط با وسایل جانبی میکرو مانند:حافظه های خارجی MMC،سنسورها و...

استفاده می شود .

طبق روال های قبلی برای استفاده از هر قسمت برد باید رجیستر ان در میکرو تنظیم شود.




```
// SPI initialization
// SPI Type: Master
// SPI Clock Rate: 2000.000 kHz
// SPI Clock Phase: Cycle Start
// SPI Clock Polarity: Low
// SPI Data Order: MSB First
SPCR=0x50;
SPSR=0x00;
```

همچنین باید پایه های متناظر با spi متناسب با عملکردشان وردی یا خروجی تنظیم شوند:

```
SPI_DDR=(1<<SPI_PORT_CS)|(1<<SPI_PORT_MOSI)|(1<<SPI_PORT_Sck)|(1<<SPI_PORT_SS);
```

برای خواندن و نوشتن اطلاعات در mmc می توانیم از کتابخانه های موجود در اینترنت استفاده کرد مثلا برنامه زیر برای خواندن و نوشتن از حافظه mmc نوشته شده است:

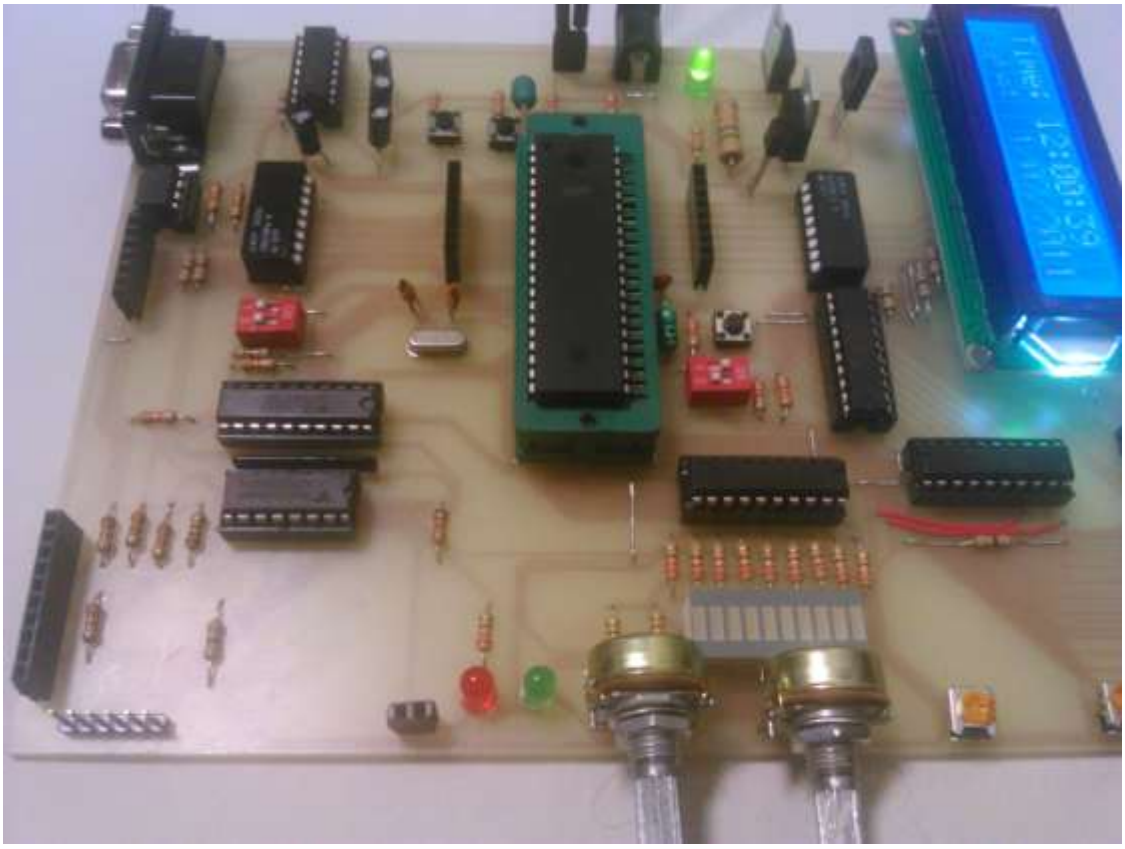
```
#include <mega32.h>
#include <delay.h>
#include <alcd.h>
#include <stdio.h>
#include <spi.h>
#include  "sd_lib.h"
char read_buff[512];
unsigned char buff2[16],sd_resp;
int nom;
void main(void){
lcd_init(16);
lcd_clear();
lcd_putsf("lcd is working");
delay_ms(200);
spi_init();   //first call spi_init then set registers
```

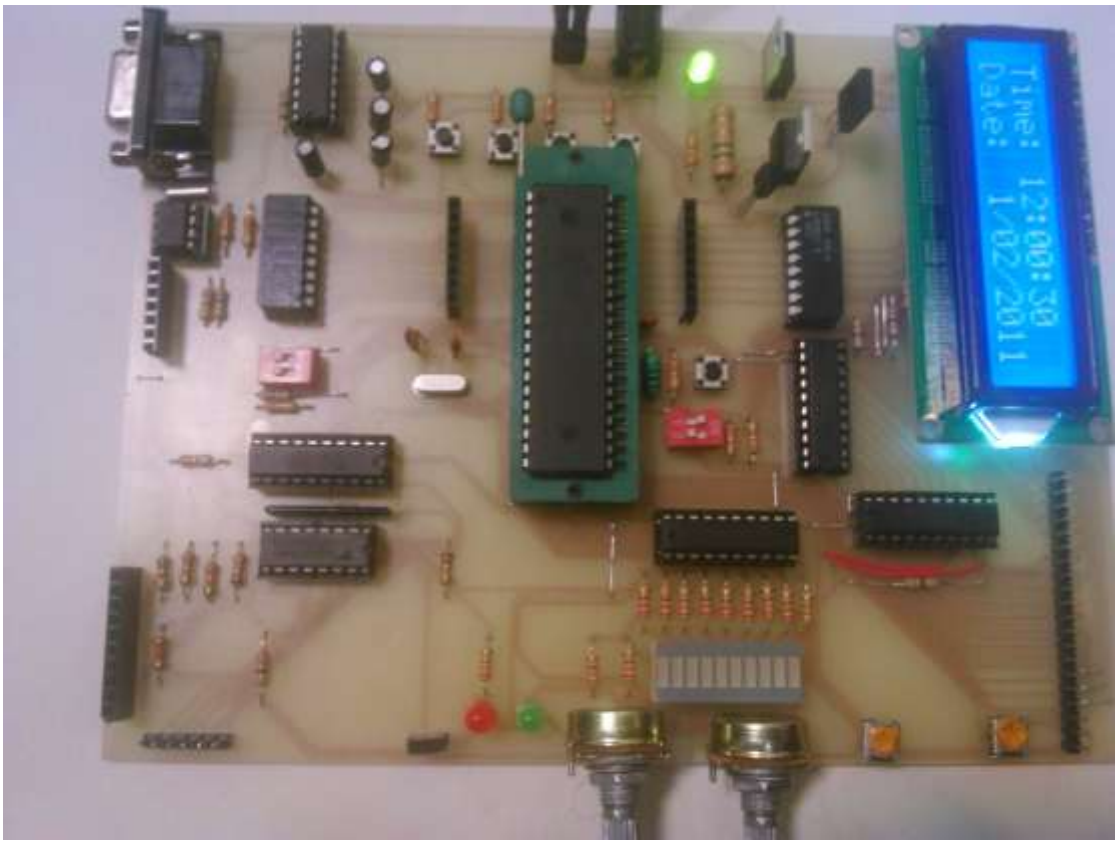
```
SPCR=0x50;   //spi registers
SPSR=0x01;


        do{
        sd_resp=sd_init();
        sprintf(buff2,"%d",sd_resp);
        }while(sd_resp !=0);
        lcd_clear();
        lcd_putsf("card init ok");

         sd_resp=read_sd(8192,read_buff);
          if(sd_resp==0){
            lcd_clear();
            lcd_putsf("read ok");
```

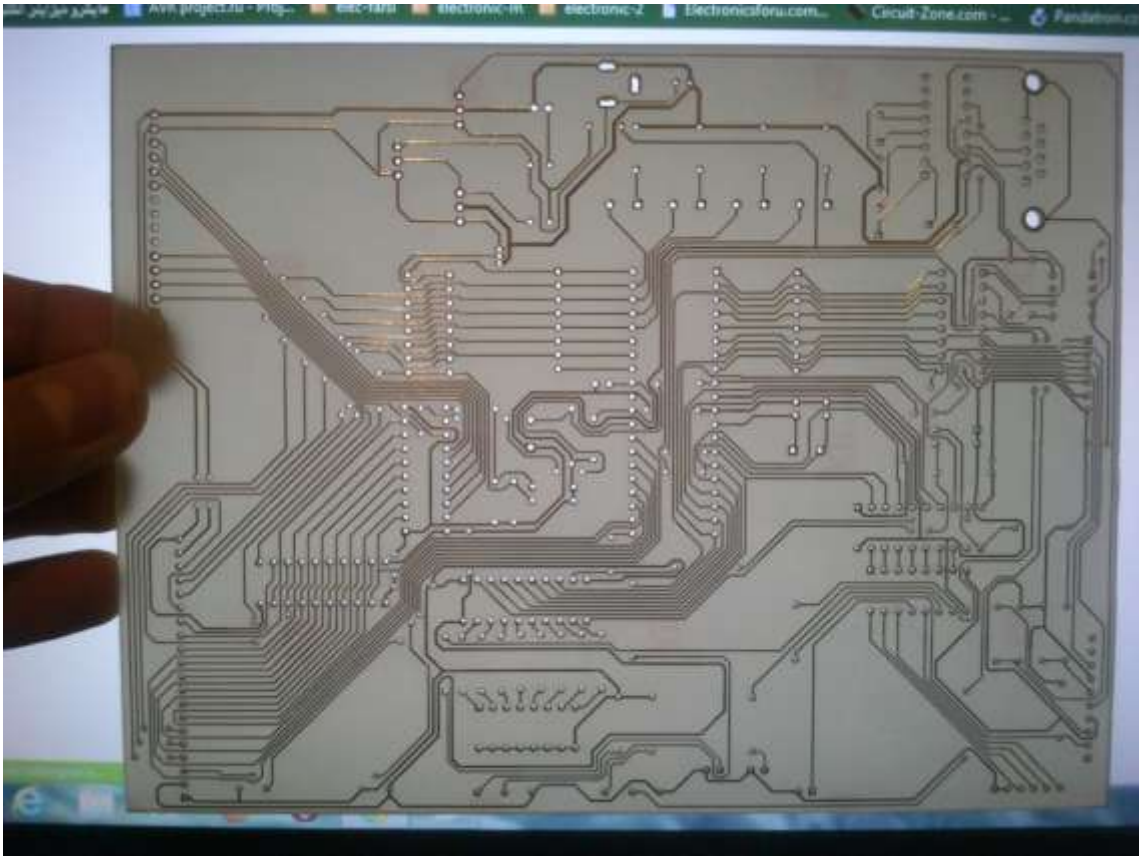
```
  delay_ms(1000);
 }

 sd_resp=write_sd(10000,read_buff);
  if(sd_resp==0){
 lcd_clear();
 lcd_putsf("wrire ok");
 delay_ms(300);
 }

       while(1){



sd_resp=read_sd(10000,read_buff);
```

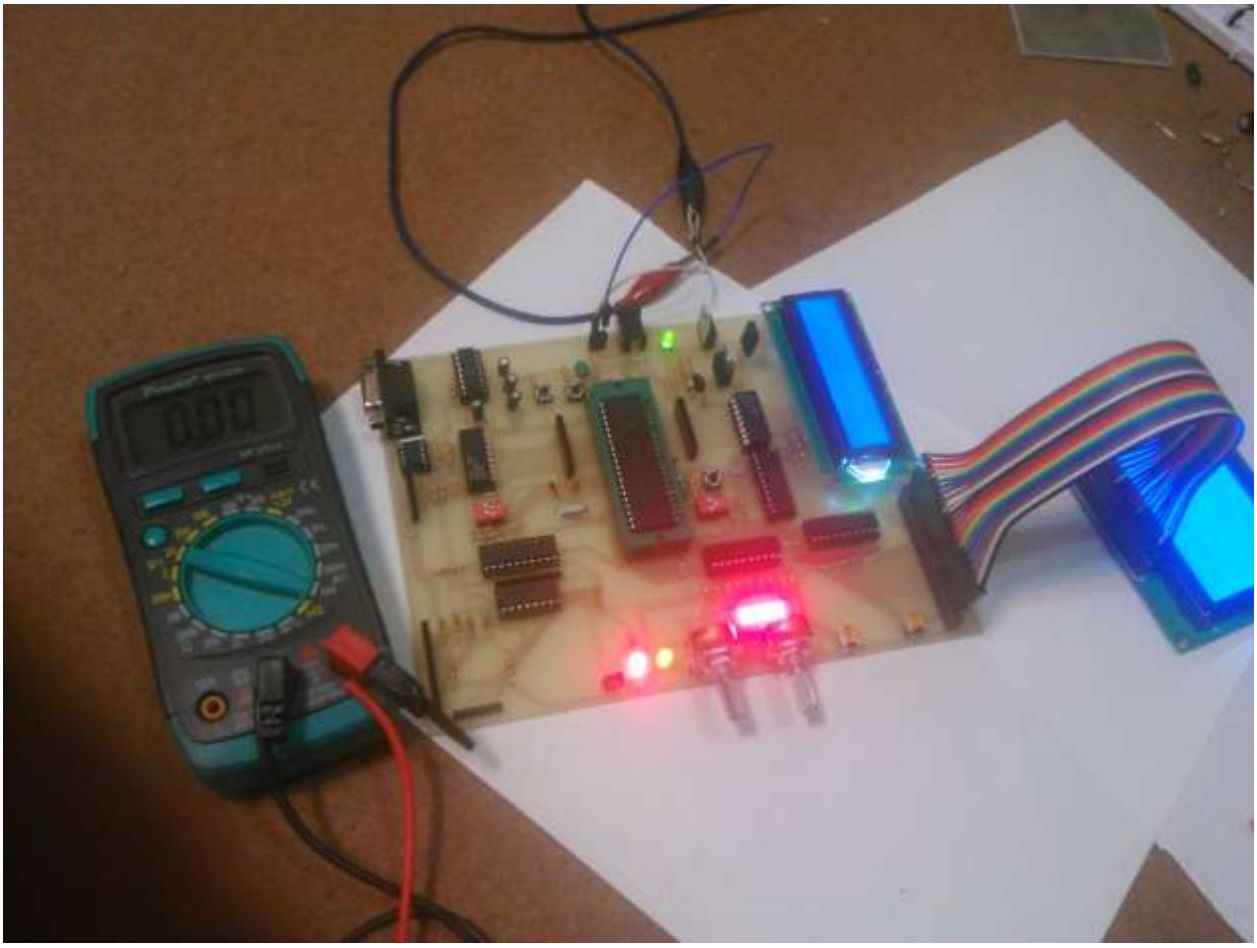
```
for(nom=0;nom<512;nom++){

                    sprintf(buff2,"%X
%X",nom,read_buff[nom]);
                    lcd_clear();
                    lcd_puts(buff2);
                    delay_ms(1000);
                     }

                }


   }
```

عکس هایی از پروژه و مراحل ساخت آن :









WWW.MicroDesigner.IR

ضمیمه ها :

ضمیمه یک

اطلاعاتی در مورد میکروکنترلرهای AVR

میکرو کنترلر چیست؟

مقدمه کلیه مدارات الکترونیکی نیاز به منبع تغذیه دارند. برای مدارات با کاربرد کم قدرت از باطری یا سلولهای خورشیدی استفاده میمیکروکنترلر چیست ؟ قطعه ای که این روزها دارد جای خود را در خیلی از وسایل الکتریکی باز از تلفن گرفته تا موبایل از ماوس لیزری که الان دستتان روی آن است و دارین باهاش کامپیوتر رو کنترل میکنید تا هر وسیله ای که بتوان پیچیدگی رو در اون دید میتونید یک میکروکنترلر رو ببینید . شود. منبع تغذیه به عنوان منبع انرژی دهنده به مدار مورد استفاده قرار می گیرد.

کلمه میکروکنترلر:

این کلمه از دو کلمه 1- میکرو 2- کنترلر تشکیل شده

*میکرو : میدونین که این یک واحد یونانی است و برابر با 10 به توان منفی 6 متر است. یعنی یک ملیونیوم متر واحده خیلی کوچیکیه نه....ولی واحدهای خیلی کوچیکتر از این هم داریم که در الکترونیک مورد استفاده قرار میگیرند در قسمتهای بعدی توضیحیهاتی راجع به این واحد ها و موارد استفاده آنها داده میشه.

*کنترلر : که همه معنی و مفهومشو میدونین . یعنی کنترل کننده به تعبیری یعنی "مغز "

البته بدون تفکر فقط دستوراتی که به اون داده میشه به نحو احسن انجام میده.

حالا چرا این کلمات ؟
به نظر من کلمه میکرو به دو منظور استفاده شده منظور اول و مهم سرعت عمل میکروکنترلر است که میتواند تا یک ملیونیوم ثانیه باشد و دستوارتی که به اون میدیم با این سرعت انجام بده به همین خاطر واژه میکرو رو به اون اختصاص دادن البته معنی دوم آن شاید کوچیکی این قطعه باشد که تا یک ملیونیوم متر کوچیک شده شاید باور کردنی نباشه ولی در یک تراشه ممکنه بیش از یک ملیون تراتزیستور به کار رفته باشه. این کلمه وقتی اهمیتش کامل میشه که با واژه کنترلر عجین بشه تا معنیش کامل بشود .

حالا نحوه انجام دادن کار میکروکنترلر را به صورت کلی بررسی میکنیم :
تا حالا همه شما با ماشین حساب کار کردین تا حالا به نحوه کار کردنش فکر کردین شما اطلاعاتتون را که همون عملیات ریاضی هست به وسیله صفحه کلید به اون میدید بعد ماشین حساب این اطلاعات رو بر مبنای دستوراتی که قبلا به اون داده شده پردازش میکند و جواب را رویlcd نمایش میدهد. در واقع یک میکروکنترلر برنامه ریزی شده به عنوان مغز ماشین حساب این اطلاعات یا داده رو از صفحه کلید میگیره روشون پردازش انجام میده و بعد بر روی lcd نمایش میده.

کار میکروکنترلر دقیقا مشابه این است میکرو کنترلر بر مبنای یک سری ورودی که به اون داده میشه مثلا این ورودی از یک سنسور دما باشه که درجه حرارت رو میگه یا از هر چیز دیگه مثل صفحه کلید بر مبنای این ورودی ها و برنامه ای که قبلا ما به اون دادیم خروجیشو تنظیم میکنه که ممکنه خروجیش یک موتور باشه یا یک lcd یا هر چیز دیگری که با الکتریسیته کار بکند. حالت دیگری هم میتونه باشه که فقط میکروکنترلر بر مبنای برنامه ای که به اون دادیم عمل کند و خروجیش رو فقط بر اساس برنامه بگیرد.

ساختمان داخلی میکروکنترلر:

کامپیوتری که الان بر روی اون دارین کار انجام میدین دارای یک پردازنده مرکزیه به نام cpu که از کنار هم قرار گرفتن چندین ملیون ترانزیستور تشکیل شده و بر روی اطلاعات پرداژش انجام میده . میکرو کنترلر هم عینا دارای یک پردازنده مرکزی به نام cpu است که دقیقا کار

cpu کامپیوتر رو انجام میده با این تفاوت که قدرت و سرعت پردازشش از cpu کمتره که به اون میکروپرسسور میگن در بخش بعدی فرق میکرو پرسسور و میکروکنترلر را بررسی میکنیم. میکروکنترلر علاوه بر cpu دارای حافظه است که ما برنامه ای که بهش میدیم در اون قرار بگیره در کنار حافظه در میکروکنترلرهای امروزی تایمرها برای تنظیم زمان کانتر ها برای شمردن کانال های آنالوگ به دیجیتال پورت های برای گرفتن و دادن اطلاعات و امکاناتی دیگر که بعدا مفصل راجع به هر کدام توضیح داده میشه تشکیل شده و همه اینها در یک چیپ قرار گرفته که تنکنولوژی جدید اونو تو یک تراشه به اندازه یک سکه قرار داده.

تفاوت میکروپروسسور و میکروکنترلر:

میکروپرسسور همانطور که گفته شد یک پردازنده است و برای کار باید به آن چیپ های حافظه و چیز های دیگری را به اون اضافه کرد این امکان به درد این میخورد که بر حسب کارمان حافظه مناسب و دیگر قطعات را مانند تایمرها و غیره به صورت بیشتری استفاده کنیم ولی مدار خیلی پیچیده میشود و از لحاظ هزینه هم هزینه بیشتر میشود به همین دلیل امروزه از میکروپرسسورها کمتر استفاده میشود اما این روزها میکرو کنترلر های جدید با حافظه های زیاد تعداد تایمر زیاد پورت های زیاد و تنوع بسیار زیاد انها بر حسب این امکانات دست ما را باز گذاشته است تا دیگر میکروپرسسورها را فراموش کنیم.

آیا میکروکنترلر چیز جدیدی را با خود آورده است ؟
جواب منفی است تمام کارهایی که ما با میکروکنترلر میتوانیم انجام بدهیم با قطعات دیگر هم میتوانیم انجام بدهیم چون ما قبلا هم تایمر داشتیم هم کانتر هم حافظه هم پردازنده و... . در واقع میکروکنترلر قطعه ای است با تمام این امکانات که به صورت یک آی سی آماده شده است و هزینه پیچیدگی و حجم را به نحوه قابل ملاحضه ای کاهش میدهد.

عیب میکروکنترلر:

میکروکنترلر با این همه مزایا که گفتیم دارای یک عیب کوچیک است .و آن سرعت پایین ! است آیا سرعتی معادل یک ملیونیوم ثانیه سرعت کمی است ؟ سرعت کمی نیست ولی یک مثال شاید بحثو بهتر باز کند
یک گیت منطقی رو در نظر بگیرین که با توجه به ورودی خروجیشو تنظیم مکنه سرعت عمل این گیت منطقی 10 به توان منفی 9 ثانیه است یعنی نانو ثانیه ولی اگر ما بخواهیم این گیت رو با میکروکنترلر کار کنیم سرعتی معادل میکرو ثانیه داریم پس از لحاظ سرعت برای

کاربردهای خیلی محدودی میکروکنترلر مناسب نیست.

خب حالا این میکروکنترلر را با این همه کاربرد کی ساخته؟
حدود 4 دهه پیش در سال 1971 میلادی شرکت اینتل اولین میکروکنترلر را ساخت و اولین میکروکنترلر را با نام 8080 در اوایل سال 1980 روانه بازار کرد .همین شرکت اینتلی که الان در ساخت cpu یکه تاز دنیاست .اما بعدا این امتیاز رو به شرکت های دیگری واگذار کرد و شرکت های زیادی در حال حاضر میکروکنترلر

های مختلف تولید میکنند

* بخشهای مختلف میکروکنترلر:

میکروکنترلر ها از بخشهای زیر تشکیل شده اند:

واحد پردازش Cpu

واحد محاسبات Alu

ورودی ها و خروجی ها I /O

حافظه اصلی میکرو Ram

حافظه ای که برنامه روی آن ذخیره می گردد Rom

برای کنترل زمان ها Timer

و . . .

*خانواده های میکروکنترلر:

خانواده : Pic - AVR - 8051

***یک میکروکنترلر چگونه برنامه ریزی میشود؟**

Assembly میکرو کنترلر ها دارای کامپایلرهای خاصی می باشد که با زبان های می توان برای آنها برنامه نوشت سپس برنامه نوشته شده را توسط basic, c که در این دستگاه ای سی قرار می گیرد و توسط programmerدستگاهی به نام یک کابل به یکی از در گاه های کامپیوتر وصل می شود برنامه نوشته شده روی Rom ذخیره می شود. آی سی انتقال پیدا میکند و در

با میکرو کنترلر چه کارهایی می توان انجام داد؟

این آی سی ها حکم یک کامپیوتر در ابعاد کوچک و قدرت کمتر را دارند بیشتر این آی سی ها برای کنترل و تصمیم گیری استفاده می شود چون طبق الگوریتم برنامه ی آن عمل می کند این آی سی ها برای کنترل ربات ها تا استفاده در کارخانه صنعتی کار برد دارد.

امکانات میکرو کنترلرها :

امکانات میکرو کنترلرها یکسان نیست و هر کدام امکانات خاصی را دارا می باشند و در قیمت های مختلف عرضه می شود.

شروع کار با میکرو کنترلر:

برای شروع کار با میکرو کنترلر بهتر است که یک زبان برنامه نویسی مثل cیا basicرا بیاموزید سپس یک برد programmer تهیه کرده و برنامه خود را روی میکرو ارسال کنید سپس مدار خود را روی برد برد بسته و نتیجه را مشاهده کنید .

چنان چه در مدارهای الکترو نیکی تجربه ندارید بهتر است از برنامه های آ موزش استفاده کنید.

مقایسه خانواده های مختلف میکرو وکنترلرها:

خانواده : AVR

این خانواده از میکرو کنترلرها تمامی امکانات 8051 را دارا می باشد و امکاناتی چون(ADC مبدل آنالوگ به دیجیتال) – نوسان ساز داخلی و قدرت و سرعت بیشتر (EEPROM –حافظه) از جمله مزایای این خانواده می باشد مهم ترین آی سی این خانواده Tiny و Mega است.

میکروهای AVR دارای انعطاف پذیری غیر قابل مقایسه و بی همتایی هستند.آنها قادر به ترکیب هر نوع کدی با یک معماری کارامد از طریق زبانهای C و Assembly هستند و قادرند از طریق این برنامه ها تمام پارامترهای ممکن در یک سیکل یا چرخه ماشین را با دقت بسیار بالا هماهنگ کنند.

میکرو AVR دارای معماریی است که میتواند در تمام جهات مورد استفاده شما،عمل کند میکرو AVR معماریی دارد که برای شما کارایی 16 بیتی ارائه می دهد که البته قیمتش به اندازه یک 8 بیتی تمام می شود.

بهره های کلیدی AVR :

دارای بهترین MCU برای حافظه فلش در جهان ! (MCU: **M**aster **C**ontrol **U**nit)

دارای سیستمی با بهترین هماهنگی

دارای بالاترین کارایی و اجرا در CPU (یک دستورالعمل در هر سیکل کلاک)

دارای کدهایی با کوچکترین سایز

دارای حافظه خود برنامه ریز

دارای واسطه JTAG که با IEEE 1149.1 سازگار است

(IEEE: **I**nstitute of **E**lectrical and **E**lectronics **E**ngineers.)

دارای سخت افزار ضرب کننده روی خود

دارای بهترین ابزارها برای پیشرفت و ترقی

دارای حالات زیادی برای ترفیع دادن یا Upgrade .

واژگان کلیدی AVR :

میکرو کنترلر AVR به منظور اجرای دستورالعملهای قدرتمند در یک سیکل کلاک(ساعت) به اندازه کافی سریع است و می تواند برای شما آزادی عملی را که احتیاج دارید به منظور بهینه سازی توان مصرفی فراهم کند.

میکروکنترلر AVR بر مبنای معماری RISC (کاهش مجموعه ی دستورالعملهای کامپیوتر) پایه گذاری شده و مجموعه ای از دستورالعملها را که با 32 ثبات کار میکنند ترکیب می کند.

به کارگرفتن حافظه از نوع Flash که AVR ها به طور یکسان از آن بهره می برند از جمله مزایای آنها است.

یک میکرو AVR می تواند با استفاده از یک منبع تغذیه 2.7 تا 5.5 ولتی از طریق شش پین ساده در عرض چند ثانیه برنامه ریزی شود یا Program شود.

میکروهای AVR در هرجا که باشند با 1.8 ولت تا 5.5 ولت تغذیه می شوند البته با انواع توان پایین (Low Power)که موجودند.

راه حلهایی که AVR پیش پای شما می گذارد، برای یافتن نیازهای شما مناسب است:

با داشتن تنوعی باور نکردنی و اختیارات فراوان در کارایی محصولات AVR، آنها به عنوان محصولاتی که همیشه در رقابت ها پیروز هستند شناخته شدند.در همه محصولات AVR مجموعه ی دستورالعملها و معماری یکسان هستند بنابراین زمانی که حجم کدهای دستورالعمل شما که قرار است در میکرو دانلود شود به دلایلی افزایش یابد یعنی بیشتر از گنجایش میکرویی که شما در نظر گرفته اید شود می توانید از همان کدها استفاده کنید و در عوض آن را در یک میکروی با گنجایش بالاتر دانلود کنید.

خانواده های محصولات AVR :

Tiny AVR:

میکروکنترلری با اهداف کلی و با بیش از 4 کیلو بایت حافظه فلش و 128 بایت حافظه استاتیک و قابل برنامه ریزی است.(منظور از حافظه استاتیک SRAM و حافظه قابل برنامه ریزی EEPROM است.)

Mega AVR:

این نوع میکروها قابلیت خود برنامه ریزی دارند و می توان آنها را بدون استفاده از مدارات اضافی برنامه ریزی کرد همچنین بیش از 256K بایت حافظه فلش و 4K بایت حافظه استاتیک و قابل برنامه ریزی دارند.

AVR های مدل Mega:

اگر شما به میکرویی احتیاج دارید که دارای سرعت و کارایی بالا باشد و توانایی اجرای حجم زیادی از کد برنامه را داشته و بتواند داده های زیادی را سروسامان دهد باید از AVR های مدل Mega استفاده کنید آنها به ازای هر یک مگا هرتز سرعت ، توانایی اجرای یک میلیون دستورالعمل در هر یک ثانیه را دارند همچنین قابل برنامه ریزی و بروزرسانی کدها با سرعت و امنیت بسیار بالایی هستند.

نکات کلیدی و سودمند مدل Mega :

- حافظه سریع از نوع فلش با عملکرد خود برنامه ریز و بلوکه ی بوت (Boot Block)
- دقت بسیار بالای 8-کانال در تبدیل آنالوگ به دیجیتال 10 بیتی
- USART و SPI و TWI بر طبق واسطه های سریال
- واسطه ی JTAG بر طبق IEEE 1149.1

TWI: **T**wo **W**ire **I**nterface is a byte oriented interface

USART: **U**niversal **S**erial **A**synchronous **R**eceiver/**T**ransmitter

SPI: **S**erial **P**eripheral **I**nterface

JTAG available only on devices with 16KB Flash and up

JATG واسط فقط در میکروهای با بیش از 16 کیلوبایت حافظه فلش موجود است.

مشخصات سخت افزاری ATMEGA32 :

شکل ظاهری و پایه ها:

ATMEGA32 در سه نوع بسته بندی PDIP با 40 پایه و TQFP با 44پایه و MLF با 44 پایه ساخته میشود که در بازار ایران بیشتر نوع PDIP موجود میباشد .

ATMRGA32 دارای چهار پورت 8بیتی (1 بایتی) دارد که علاوه بر اینکه بعنوان یک پورت معمولی میتوانند باشند کارهای دیگری نیز انجام میدهند . بطور مثال PORTA میتواند بعنوان ورودی ADC (تبدیل ولتاژ آنالوگ به کد دیجیتال) استفاده شود که این خاصیت های مختلف پورت در برنامه ای که نوشته میشود تعیین خواهد شد .
ولتاژ مصرفی این آی سی از 4.5 V تا V5.5 میتواند باشد .
فرکانس کار هم تا MHz16 میتواند انتخاب شود که تا MHz8 نیازی به کریستال خارجی نیست و در داخل خود آی سی میتواند تامین شود . فرکانس کار از جمله مواردی است که باید در برنامه تعیین شود . لازم به ذکر است که این فرکانس بدون هیچ تقسیمی به CPU داده میشود . بنابراین این خانواده از میکروکنترلرها سرعت بیشتری نسبت خانواده های دیگر دارند .
پایه ی شماره 9 نیز ریست سخت افزاری میباشد و برای عملکرد عادی آی سی نباید به جایی وصل شود و برای ریست کردن نیز باید به زمین وصل میشود .
پایه های 12 , 13 نیز برای استفاده از کریستال خارجی تعبیه شده است .

ساختار داخلی ATMGA32 :

برنامه ای که برای میکروکنترلر در کامپیوتر نوشته میشود وقتی که برای استفاده در آی سی ریخته میشود (توسط پروگرامر مخصوص آن خانواده) در مکانی از آن آی سی ذخیره خواهد شد بنام ROM . حال در ATMEGA32 مقدار این حافظه به 3232) KB کیلوبایت) میرسد .
در این آی سی مکانی برای ذخیره موقت اطلاعات یا همان RAM هم وجود دارد که مقدارش KB2 است .
در RAM اطلاعات فقط تا زمانی که انرژی الکتریکی موجود باشد خواهد ماند و با قطع باتری اطلاعات از دست خواهند رفت . به همین منظور در ATMEGA32 مکانی برای ذخیره اطلاعات وجود دارد که با قطع انرژی از دست نخواهند رفت . به این نوع حافظه ها EEPROM گفته

میشود که در این آی سی مقدارش 1KB است و تا 100,000 بار میتواند پر و خالی شود .

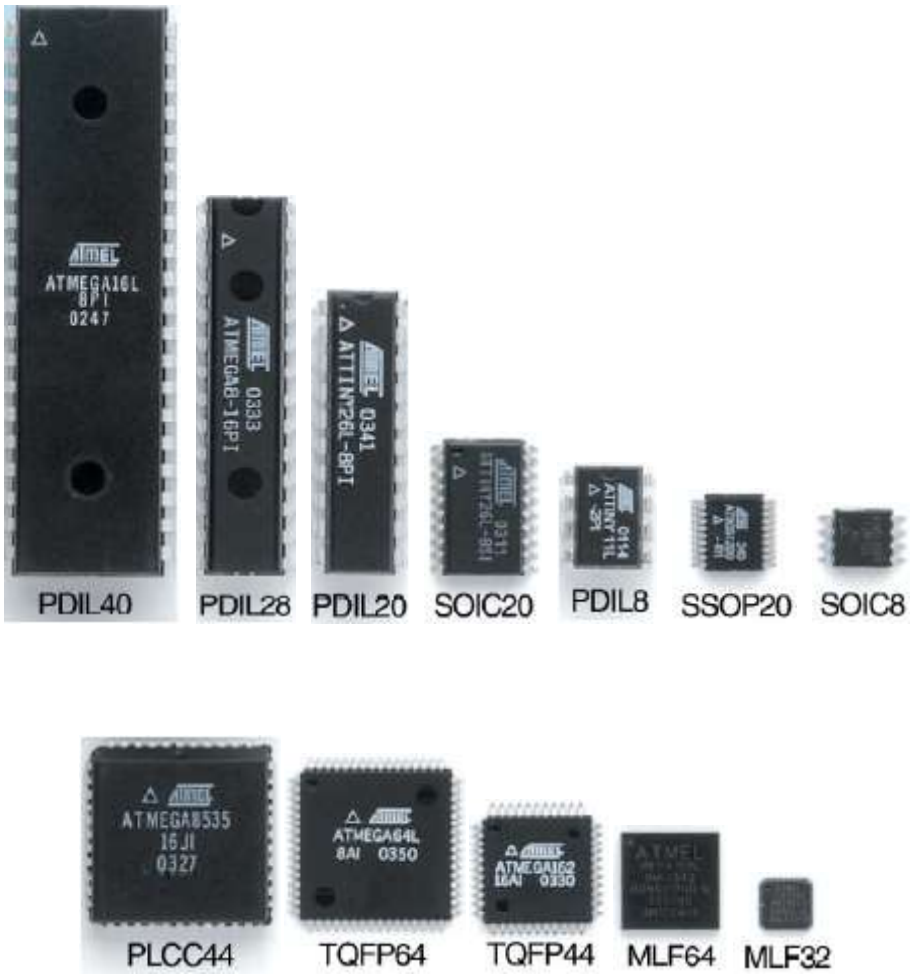
AVR های مدل tiny:

به خود اجازه ندهید که نام آن شما را گول بزند... میکروهای مدل tiny توانایی های عظیمی دارند.به خاطر کوچک بودن و داشتن MCU بسیار پر قدرت به اینگونه میکروها نیاز فراوانی هست آنها به هیچ منطق خارجی نیاز نداشته و به همراه یک مجتمع مبدل آنالوگ به دیجیتال و یک حافظه قابل برنامه ریزی EEPROM قابلیتهای خود را ثابت می کنند.

نکات کلیدی و سودمند مدل Tiny :

- آنها به منظور انجام یک عملیات ساده بهینه سازی شده و در ساخت وسایلی که به میکروهای کوچک احتیاج است کاربرد فراوان دارند.
- کارایی عظیم آنها برای ارزش و بهای وسایل موثر است.

ابعاد مختلف میکروهای AVR را در اشکال زیر مشاهده می کنید:



نکات کلیدی و سودمند حافظه ی فلش خود برنامه ریز:

- قابلیت دوباره برنامه ریزی کردن بدون احتیاج به اجزای خارجی
- 128 بایت کوچک که به صورت فلش سکتور بندی شده اند
- داشتن مقدار متغیر در سایز بلوکه ی بوت (Boot Block)
- خواندن به هنگام نوشتن
- بسیار آسان برای استفاده
- کاهش یافتن زمان برنامه ریزی
- کنترل کردن برنامه ریزی به صورت سخت افزاری

راههای مختلف برای عمل برنامه ریزی:

Parallelموازی:

- یکی از سریعترین روشهای برنامه ریزی
- سازگار با برنامه نویس های(programmers) اصلی

خود برنامه ریزی توسط هر اتصال فیزیکی:

- برنامه ریزی توسط هر نوع واسطه ای از قبیل TWI و SPI و غیره
- دارا بودن امنیت صد درصد در بروزرسانی و کدکردن

ISP:

- واسطه سه سیمی محلی برای بروزرسانی سریع
- آسان و موثر در استفاده

JTAGواسطه:

- واسطه ای که تسلیم قانون IEEE 1149.1 است و می تواند به صورت NVM برنامه ریزی کند یعنی هنگام قطع جریان برق داده ها از بین نروند.استفاده از فیوزها و بیتهای قفل.

ضمیمه دوم :

معرفی LCD کارکتری :
LCD های کارکتری خود به چند نوع دیگر از لحاظ اندازه تقسیم بندی میشوند . که از LCD هایی با 1 سطر و 1 ستون آغاز میشوند تا اندازهایی مثل 4 سطر و 40 ستون که البته تمام آنها از 16پایه تشکیل شده اند.

برای راه اندازی LCD توسط AVR نیازی به دانستن جزئیات طرز کار LCD نیست . برای کار با LCD علاوه بر پایه های تغذیه و CONTRAST (تنظیم روشنایی) که باید مانند شکل مداری پایین بایاس شوند نیاز به 6 پایه ی دیگر است که عبارتند از پایه های :
. RS , E , DB4 , DB5 , DB6 , DB7

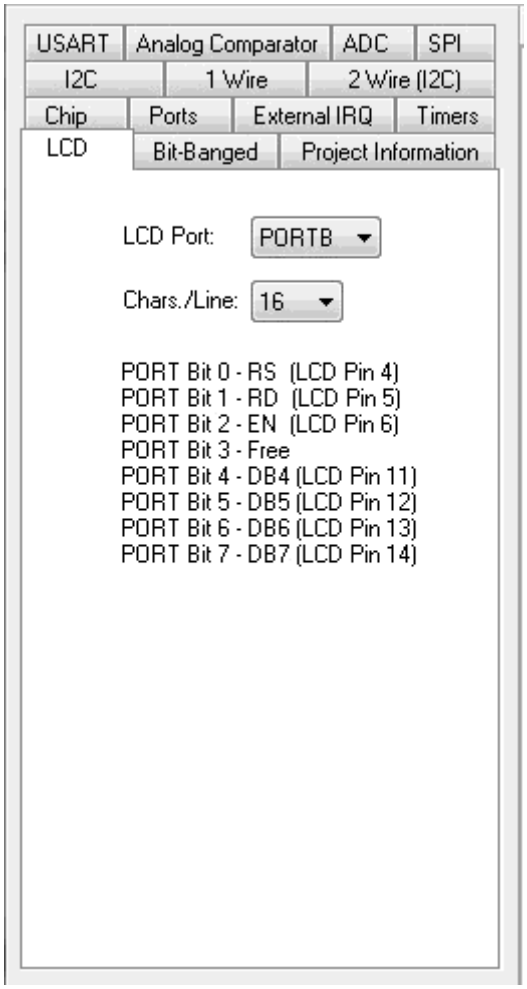
نمایشگرها در ساخت ربات‌ها و دستگاه‌های هوش‌مند الکترونیکی کاربرد بسیار زیادی دارند. با ذکر چند مثال شما را با کاربرد این نمایشگرها بیشتر آشنا می‌کنیم.
در ربات مین‌یاب برای اعلام مختصات مین‌ها به داور، باید روبات مجهز به نمایشگری باشد که بتوان این اطلاعات را بر روی آن به نمایش درآورد.
در ربات فوتبالیست، نمایشگر در زمان مسابقه کاربرد مستقیمی ندارد، اما در مراحل عیب‌یابی و تنظیمات اولیه سنسورها کاربرد زیادی دارد. مثلاً برای تنظیم حساسیت هر سنسور، اطلاعات آن بر روی صفحه نمایش به کاربر نشان داده می‌شود و کاربر می‌تواند آن را سریع تر تنظیم کند. به‌عنوان مثال برای تنظیم سنسورهای نوری می‌توان ولتاژ خروجی آن را توسط ADC اندازه‌گیری کرد و بر روی LCD نمایش داد.
از دیگر موارد کاربرد این نوعLCD ها می‌توان به دستگاه‌های تلفن خانگی اشاره کرد که به‌کمک آن، داده‌هایی مثل شماره‌ی تلفن فرد تماس‌گیرنده، دفترچه تلفن و ... را نمایش

می‌دهد.
LCDهای کارکتری در سایزهای مختلفی وجود دارند. سایز این نوع LCD را بر اساس تعداد کاراکترهایی که در هر سطر و ستون نمایش داده می‌شوند، تعیین می‌کنند. پرکاربردترین سایزLCD های کاراکتری 16*2 است، یعنی این LCD می‌تواند 2 ردیف 16 تایی کاراکتر را همزمان روی صفحه نمایش دهد.

چگونه از lcd استفاده کنیم؟
در ساختمان داخلی این LCDها مدارات متعددی وجود دارد که اطلاعاتی که برای نمایش دادن به LCD فرستاده می‌شود را پردازش کرده و اطلاعات مورد نظر ما را روی صفحه به‌نمایش در می‌آورند .این اطلاعات باید از طریق پایه‌های LCD به آن منتقل شوند. برقراری ارتباط و نمایش اطلاعات بر روی LCD کار چندان ساده ای نیست، اما CodeVision در اینجا هم به کمک ما آمده است و کار را بسیار ساده کرده است.
توضیح در مورد نحوه‌ی استفاده از LCD را از تنظیمات نرم‌افزاری آن در محیط codevisionشروع می‌کنیم.

تنظیمات اولیه در code vision برای راه اندازی lcd :

Codevisionرا باز کرده پروژه‌ی جدیدی بسازید. سپس در Code Wizard تنظیمات مربوط به لبه‌ی Chip را انجام دهید.
حالا سراغ لبه‌ی LCD می‌رویم.
برای راه‌اندازیLCD های کارکتری، باید تمام پایه‌های یکی از پورت‌های میکروکنترلر را به پایه‌های مربوطه در LCD متصل کنیم.
ابتدا باید تعیین کنیم می‌خواهیم کدام پورت را به LCD اختصاص دهیم.
سپس باید با تعیین تعداد کاراکترهای قابل نمایش در هر سطر از LCD نوع آن را مشخص کنیم .مثلاً اگر LCD ما 16*2 است، باید عدد 16 را انتخاب کنیم.
سپس نحوه‌ی اتصال پایه‌های میکروکنترلر به LCD را با توجه به نوع LCD به شما نشان می‌دهد.



برای مثال ترتیب اتصال پایه‌ها برای LCD 16*2 بر روی پورت "B" در زیر نشان داده شده است.
پایه PB.0به پایه‌ی چهارم LCD متصل شود.
پایه PB.1 به پایه‌ی پنجم LCD متصل شود.
پایه PB.2 به پایه‌ی ششم LCD متصل شود.
پایه PB.3 به جایی متصل نمی‌شود.
پایه PB.4 به پایه‌ی یازدهم LCD متصل شود.
پایه PB.5 به پایه‌ی دوازدهم LCD متصل شود.
پایه PB.6 به پایه‌ی سیزدهم LCD متصل شود.
پایه PB.7 به پایه‌ی چهاردهم LCD متصل شود.

بعد از اینکه طبق ترتیب ذکر شده پایه‌ها را متصل کردیم، و تنظیمات اولیه را در CodeWizardانجام دادیم، سراغ برنامه‌نویسی آن می رویم CodeVision .توابعی را آماده کرده است که به کمک آن‌ها می‌توانیم به‌سادگی اطلاعات موردنظر خودمان روی LCD نمایش دهیم.

4 دستور اصلی برای نمایش اطلاعات روی lcd :

1- lcd_putchar(` `);

این دستور برای نمایش یک کاراکتر بر روی LCD استفاده می‌شود. مثلاً دستور زیر حرف F را بر روی LCD نمایش می‌دهد :

```
lcd_putchar(`F’);
```

2- lcd_putsf(“ “);

این دستور برای نمایش یک رشته از حروف بر روی LCD استفاده می‌شود. مثلاً دستور زیر جمله‌ی it is a test را بر روی LCD نمایش می‌دهد:

```
lcd_putsf(“it is a test”);
```

3- lcd_clear();

این دستور برای پاک کردن LCD مورد استفاده قرار می‌گیرد. این دستور هر کاراکتری را که روی LCD در حال نمایش باشد پاک می‌کند.

4- lcd_gotoxy(,);

به‌کمک این دستور می‌توان تعیین کرد کاراکتر یا جمله‌ی مورد نظر ما در کدام سطر و ستون در LCD نوشته شود .مثلاً دستورهاي زیر lcd را پاک کرده و واژه‌ی Helloرا از وسط سطر دوم می‌نویسد .شماره‌گذاری سطرها و ستون‌ها از 0 شروع می‌شود. پس سطر شماره‌ی 1 ، سطر دوم است.

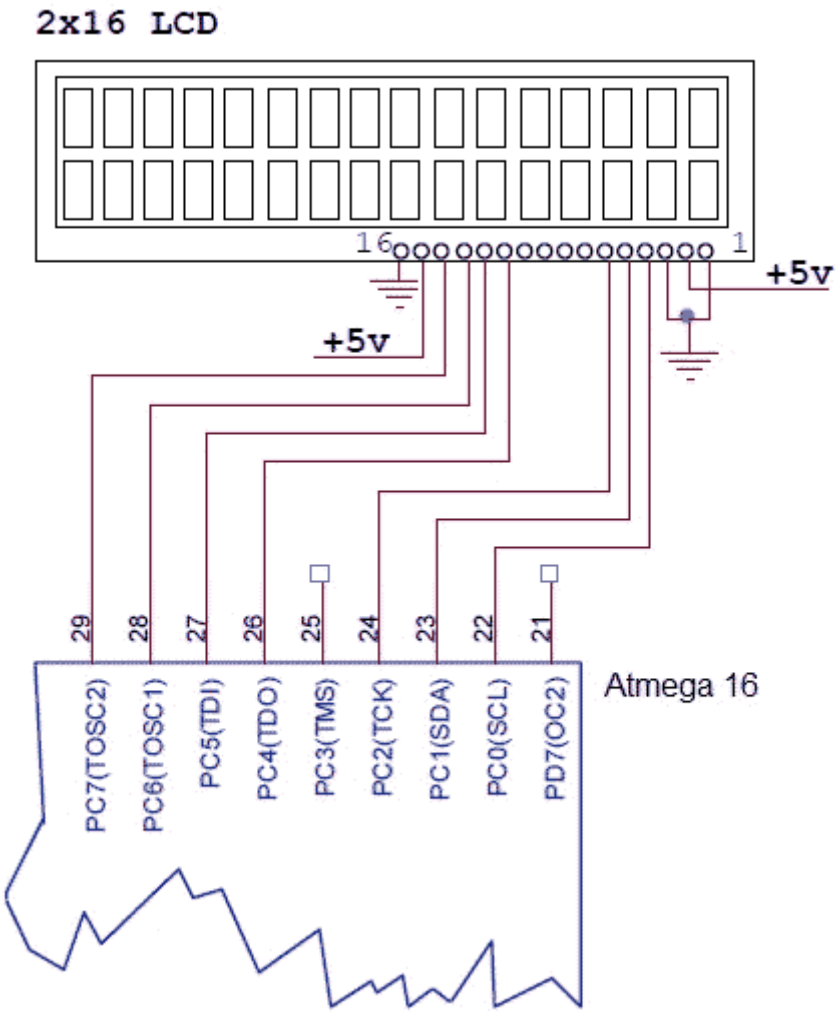
```
lcd_clear();
lcd_gotoxy(1,7);
lcd_putsf(“Hello”);
```

سایر پایه های lcd :

LCDنیز مانند هر قطعه‌ی الکترونیکی دیگر نیاز به 2 پایه برای تغذیه + و – دارد. در LCDهای 2*16 اختلاف پتانسیل مورد نیاز برای تغذیه باید 5 ولت باشد. پایه شماره‌ی 1 باید به GND و پایه‌ی شماره‌ی 2 باید به 5 ولت متصل شود. پایه‌ی شماره‌ی 3 نیز برای تنظیم نور زمینه در LCD تعبیه شده است. در حالت معمولی باید این پایه مستقیماً به GND متصل شود.

پایه‌های 15 و 16 نیز برای تغذیه‌ی نور پشت زمینه هستند. پایه‌ی 15 به Vcc (5 ولت) و پایه‌ی 16 به GND متصل می‌شود

.



معرفی خطهای lcd :

شماره و نام خط	عملکرد
Vss -1	زمین
Vcc -2	ولتاژ 5 ولت برای کنترلر
Vee -3	ولتاژ تنظیم درخشندگی(contrast)
RS -4	انتخابگر ثبات دستور / داده
RW -5	انتخابگر خواندن / نوشتن
Enable -6	فعال کننده
Bus 14-7	8 خط گذرگاه داد یا دستور
-15	ولتاژ 5 ولت برای لامپ پشت صفحه
-16	زمین برای لامپ پشت صفحه

Vee : برای تنظیم درخشندگی کاراکترها بکار می رود که باید ولتاژی بین صفر و 5 ولت به این پایه اعمال نمود. برای بیشترین درخشندگی این پایه را به زمین متصل کنید.

انتخابگر ثبات داده / دستور مشخص می کند که چه چیزی به LCD فرستاده می شود. اگر این خط صفر باشد کنترلر LCD
کرده و اگر این پایه یک باشد اطلاعات را بعنوان یک کد اسکی که باید کاراکتر معادل آنرا نمایش دهد در نظر می گیرد.

انتخابگر خواندن / نوشتن جهت اطلاعات را نشان می دهد. اگر این پایه صفر باشد اطلاعات به LCD ارسال می شود و اگر یک باشد عمل خواندن از LCD صورت می گیرد.

فعال کننده: برای هر دستور یا داده ای که به LCD میفرستیم یا میخواهیم از آن بخوانیم باید یک پالس پائین رونده (یعنی تغییر از سطح یک به صفر) را به این پایه اعمال کنیم تا دستور یا داده بوسیله کنترلر LCD پردازش شود.

در خطوط 7 تا 14 خط 7 کم ارزشترین بیت(LSB) و خط 14 پر ارزش ترین بیت (MSB) می باشد.

در صورت تمایل به روشن کردن لامپ پشت صفحه ولتاژ 5 ولت را به پایه 15 اعمال و پایه 16 را به زمین متصل می کنیم.

برای آزمایش می توان LCD را به پورت چاپگر متصل و اطلاعاتی را به آن ارسال نمود. در این حالت بطور معمول خطوط داده پورت به خطوط 7 تا 14 و سه خط کنترلی به پایه های 4 تا 6 اتصال داده می شود توجه داشته باشید که ولتاژ تغذیه و لامپ پشت صفحه LCD توسط منبع خارجی تامین می شود.

روش فرستادن یک کاراکتر:

خط خواندن نوشتن را صفر کنید تا نوشتن انتخاب شود.
خط داده / دستور را یک کنید تا داده انتخاب شود.
کد اسکی کاراکتر مورد نظر را روی خطوط 0D تا 7D قرار دهید.
خط انتخاب را ابتدا یک و سپس صفر کنید. حداقل 450 نانو ثانیه باید این خط را صفر نگه دارید تا داده پردازش شود. بعد از آن حالت خط تاثیری نخواهد داشت

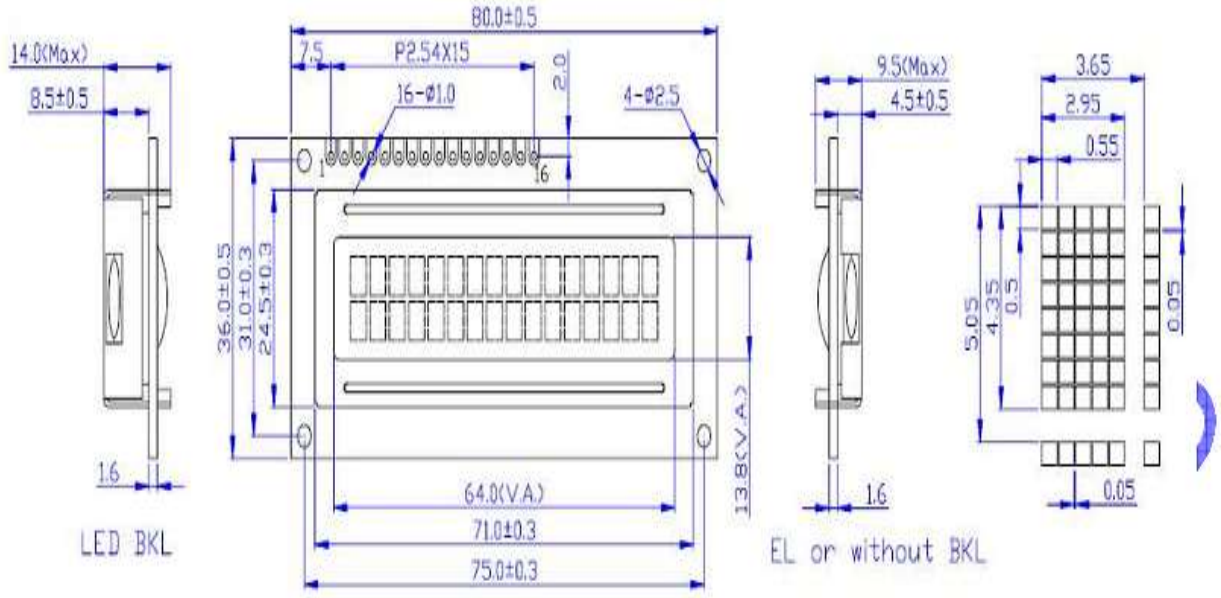
برای آزمایش می توان LCD را به پورت چاپگر متصل و اطلاعاتی را به آن ارسال نمود. در این حالت بطور معمول خطوط داده پورت به خطوط 7 تا 14 و سه خط کنترلی به پایه های 4 تا 6 اتصال داده می شود توجه داشته باشید که ولتاژ تغذیه و لامپ پشت صفحه LCD توسط منبع خارجی تامین می شود.
خط انتخاب را ابتدا یک و سپس صفر کنید. حداقل 450 نانو ثانیه باید این خط را صفر نگه دارید تا داده پردازش شود. بعد از آن حالت خط تاثیری نخواهد داشت.

Specifications:

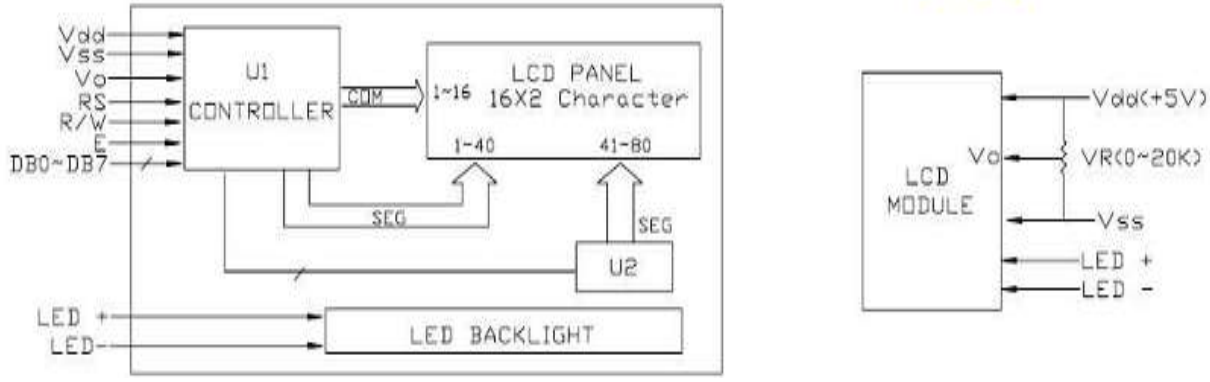
Display Format	: 16 characters (W) x 2 lines (H)
General Dimensions	: 80.0 mm (W) x 36.0 mm (H) x 9.5 mm (T)
Character Size	: 2.95 mm (W) x 4.35 mm (H)
Character Pitch	: 3.65 mm (W) x 5.05 mm (H)
Viewing Area	: 64.0 mm (W) x 13.8 mm (H)
Dot Size	: 0.55 mm (W) x 0.50 mm (H)
Dot Pitch	: 0.60 mm (W) x 0.55 mm (H)
Display Type	: Positive or Negative
LC Fluid	: STN Yellow-Green
Backlight LED	: Optional
Polarizer Mode	: Reflective
View Angle	: 6 o'clock or 12 o'clock
Controller	: S6A0069 or Equivalent
Temperature Range	: 0ºC to 50ºC (Operating), -20ºC to 70ºC (Storage)

WV

Diagrams:



Outlined Dimension



Block Diagram

Pin Connections:

Pin Number	Symbol	Function
1	Vss	Ground for Logic
2	Vdd	Power Supply for Logic
3	Vo	Power Supply for LCD
4	RS	Register Selection (H: Data, L: Instruction)
5	R/W	Read/Write Selection (H: Read, L: Write)
6	E	Enable Signal
7-14	DB0 – DB7	Data Bus Lines
15	A	BKL +
16	K	BKL -

Electrical Characteristics:

Item	Symbol	Test Condition	Minimum	Typical	Maximum
Operating Voltage (V)	Vdd	Ta=25°C	-	5.0	-
Operating Voltage for LCD (V)	Vlcd	Ta=25°C	-	4.5	-
Current Supply (mA)	Idd	Ta=25°C, Vdd=5.0V	-	2.0	3.0
Voltage Supply for LED (V)	Vf	Ta=25°C, R=6.8Ω	-	4.2	-
Current Supply for LED (mA)	If	Ta=25°C, Vf=4.2V	-	110	-

ضميمه سوم :

تايمر/كانتر صفر

تايمر /كانتر صفر يك تايمر 8 بيتي مي باشد كه داراي چهار مد كاری Normal ، CTC ، Fast PWM و Correct PWM Phase مي باشد . پين T0 به عنوان ورودي كانتر و پين OC0 خروجي بخش مقايسه ي تايمر مي باشد . اين تايمر داراي سه رجيستر به نام هاي TCCR0 ، TCNT0 و OCR0 مي باشد كه به ترتيب جهت پيكربندي تايمر، مقدار شمارنده و مقدار مقايسه استفاده مي شوند . همچنين اين تايمر در رجيسترهاي TIFR و TIMSK كه به ترتيب رجيسترهاي پرچم و وقفه تايمر مي باشند با ديگر تايمرها مشترك مي باشد.

مهم ترين رجيستر تايمر TTCR0 مي باشد كه بيت هاي Clock Select جهت انتخاب منبع

كلاك تايمر و بيت های Wave Generation Mode براي تنظيم مد كاري تايمر و بيت هاي Compare Match Output Mode پيكربندي پين OC0 را تعيين مي كنند.

TCCR0	7	6	5	4	3	2	1	0
نام بيت	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00

مد عملکرد تایمر به صورت زیر است :

مد	WGM01	WGM00	مد عملکرد
0	0	0	Normal
1	0	1	PWM, Phase Correct
2	1	0	) Clear Timer on Compare Match (CTC
3	1	1	Fast PWM

1. مد Normal

رجیستر های TIMER0 :

TCCR0	7	6	5	4	3	2	1	0
نام بیت	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00
سطح منطقی	0	0	0	0	0	X	X	X

تايمر/كانتر صفر و يك داراي يك Prescale مشترك بوده وضعيت منبع كلاك با توجه به بيت هاي Clock Select تعيين مي شود :

وضعيت منبع كلاك تايمر	CS00	CS01	CS02
بدون كلاك (متوقف)	0	0	0
كلاك سيستم (بدون تقسيم)	1	0	0
8/كلاك سيستم	0	1	0
64/كلاك سيستم	1	1	0
256/كلاك سيستم	0	0	1
1024/كلاك سيستم	1	0	1
لبه ي پايين رونده ي پالس خارجي (T0)	0	1	1
لبه ي بالا رونده ي پالس خارجي (T0)	1	1	1

مقدار تايمر در هر لحظه از طريق رجيستر TCNT قابل خواندن و نوشتن است:

Bit	7	6	5	4	3	2	1	0
TCNT0	TCNT0[7:0]							

در زمان سرريز تايمر، بيت TOV0 از رجيستر TIFR يك مي شود.

TIFR	7	6	5	4	3	2	1	0
نام بيت	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
سطح منطقی	0	0	0	0	0	0	0	X

در صورتي كه بيت فعال ساز عمومي وقفه فعال بوده و بيت هاي TOIE0 و TOIE2 در رجيستر TIMSK يك باشند مي توان با استفاده از وقفه ، از سرريز شدن تايمر به عنوان وقفه استفاده كرد:

TIMSK	7	6	5	4	3	2	1	0
نام بيت	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0
سطح منطقی	0	0	0	0	0	0	0	X

تايمر /كانتر با عملكرد مقايسه

Bit	7	6	5	4	3	2	1	0
OCR0/2	OCR0[7:0]							

محتواي رجيستر OCR0 به طور پيوسته با مقدار TCNT0مقايسه مي شود و در صورت برابري باعث تغيير وضعيت پين OC0 و يا وقفه ي تطابق مي شود . در حالت برابري بيت OCF0 يا OCF1 يک شده و با فراخو اني سابروتين وقفه به صورت سخت افزاري صفر
مي شود . در صورت عدم استفاده از وقفه كاربر مي تواند با نوشتن يك روي اين بيت آن را
پاك كند.

TIFR	7	6	5	4	3	2	1	0
نام بيت	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
سطح منطقی	0	0	0	0	0	0	X	X

تغيير وضعيت پين OC0 بوسيله بيت هاي COM00 و COM01 مي باشد:

TCCR0	7	6	5	4	3	2	1	0
نام بيت	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00
سطح منطقی	0	0	X	X	0	X	X	X

COM01	COM00	وضعيت پين OC0
0	0	غير فعال (I/O معمولی)
0	1	Toggle
1	0	Clear
1	1	Set

در صورت يك كردن بيت FOC0 يا FOC1 به صورت آني مقدار رجيستر TCNT0 با مقدار OCR0 مقايسه شده و در صورت تطبيق مقايسه يك تغيير وضعيت روي پين OC0 ايجاد
مي شود . در اين وضعيت بيت OCF0 يا OCF1 يك نشده و باعث ايجاد وقفه نيز نخواهد
شد.در تمام حالت هايي كه روي پين هاي OC شكل موج ايجاد مي شود بايد اين پين به
صورت خروجي تعريف شده باشد .

براي استفاده از وقفه بايد علاوه بر يك بودن فعال ساز عمومي وقفه ها، بيت فعال ساز مقايسه ي وقفه نيز Set شود.

TIMSK	7	6	5	4	3	2	1	0
نام بيت	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0
سطح منطقی	X	X	0	0	0	0	X	X

در اين حالت ISR به صورت زير تعريف مي شود:

```
interrupt [TIM0_COMP] void timer0_comp_isr(void)
```

{

زير برنامه ي سرويس وقفه

}

2. مد CTC

در اين مد تايمر همانند وضعيت نرمال با عملكرد مقايسه عمل مي كند با اين تفاوت كه در زمان تطابق رجيسترهاي OCR0 و TCNT0 مقدار رجيستر TCNT0 صفر شده و در واقع بر خلاف حالت قبل مقدار ماكزيمم TCNT0 عدد موجود در رجيستر OCR0 مي باشد . مقدار بيت هاي WGM00 و WGM01 در اين مد به ترتيب برابر 0 و 1 می باشد .

TCCR0	7	6	5	4	3	2	1	0
نام بيت	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00
سطح منطقی	X	0	X	X	0	X	X	X

در اين حالت فركانس موج ايجاد شده روي پين OC0 از رابطه ي زير بدست مي آيد:

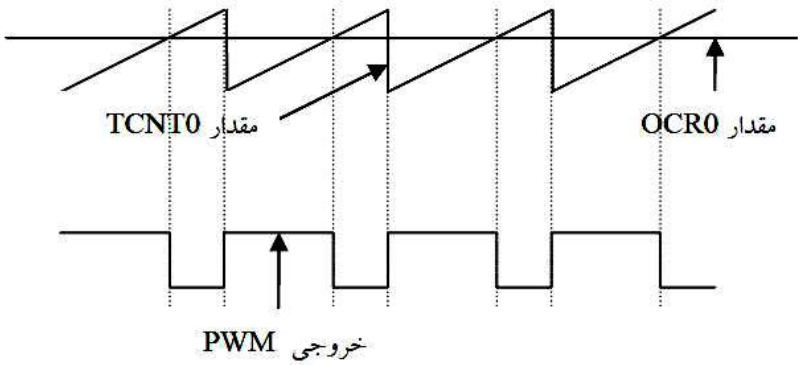
$$F_{OC0} = \frac{F_{CLK-I/O}}{2N(1+OCR0)}$$

وضعيت بيت هاي فعال ساز وقفه ي سرريز و وقفه ي مقايسه در مد CTC همانند وضعيت نرمال مي باشد. با يك بودن بيت OCIE0 وقفه ي مقايسه فعال مي باشد و مي توان در ISR اين وقفه مقدار OCR0 را تغيير داد .

مقدار دهي به رجيستر OCR0 بايد با دقت انجام شود زيرا در مدهای غير PWM اين رجيستر دارای بافر دوبل نمي باشد . وجود بافر دوبل باعث مي شود كه اگر OCR0 به مقداري كمتر از TCNT0 تغيير كند، براي از دست نرفتن مقايسه ي فعلي مقدار جديد در بافر دوبل ذخيره شده و پس از سرريز اين مقدار جديد درOCR0 بارگذاري شود.

3. مد Fast PWM

اين حالت مشابه مد نرمال مي باشد با اين تفاوت كه پين OC0 فقط در حالت برابري رجيسترهاي TCNT0 و OCR0 تغيير حالت نمي دهد ، بلكه در زمان سرريز رجيستر TCNT0 نيز مقدار اين پين تغيير می كند .



مقدار بيت هاي WGM00 و WGM01 در اين مد برابر 1 می باشد .

در مد های PWM عملكرد بيت های COM00 و COM01 متفاوت از دو وضعيت قبلي و به صورت زير می باشد :

وضعيت پين OC0	COM00	COM01
غير فعال (I/O معمولی)	0	0
رزرو شده	1	0
Clear دروضعيت تطابق ، Set در زمان سرريز	0	1

(PWM غير معكوس)		
Set دروضعيت تطابق ، Clear در زمان سرريز (PWM معكوس)	1	1

براي محاسبه ي فركانس موج PWM توليد شده مي توان از فرمول زير استفاد نمود:

N = Prescale = 1, 8, 64, 256, 1024 مقدار بارگذاری شده در تايمر X=256-

$$F_{PWM} = \frac{F_{CLK}}{N.X}$$

از وقفه ي سرريز تايمر مي توان براي مقدار اوليه دادن به TCNT0 و يا تغيير مقدار OCR0 استفاده نمود ، اگرچه بهتر است مقدار OCR0 در روتين وقفه ي مقايسه تغيير داده شود.

با مقدار اوليه دادن به TCNT0 مي توان فركانس موج PWM را تغيير داد.

با كمتر شدن OCR0 زمان وظيفه كمتر شده و تا حدي كه مقدار صفر يك پالس سوزني به عرض يك سيكل ايجاد خواهد كرد.

با مقدار دهی 256 به OCR0 مقدار مقايسه و سرريز برابر شده و پالس خروجي بسته به مقدار COM00 و COM01 همواره صفر يا يك خواهد بود.

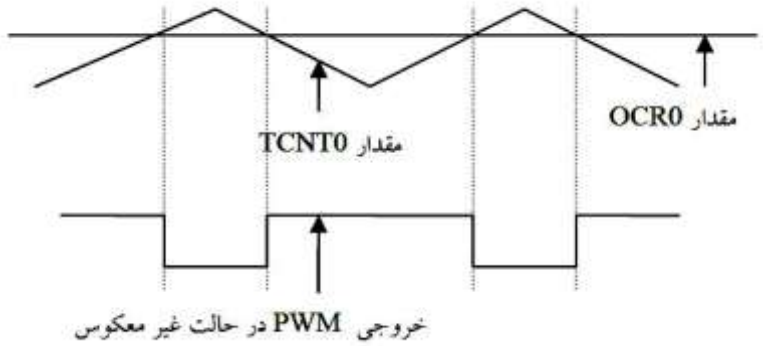
4. مد Phase Correct PWM

اين مد شبيه حالت Fast PWM مي باشد با اين تفاوت كه تايمر به صورت دو شيبه (Dual Slope) عمل شمارش را انجام مي دهد . به اينصورت كه تايمر از عدد صفر شروع به شمارش

كرده و به صورت افزايشي تا عدد 0xFF افزايش مي يابد و بعد از رسيدن به اين مقدار عدد موجود در بافر OCR0 در رجيستر OCR0 بارگذاری مي شود . بعد از اين

لحظه جهت شمارش تايمر عوض شده و به صورت كاهشي تا عدد صفر مي شمارد با رسيدن به اين عدد پرچم سرريز

تايمر يك شده و در صورت يك بودن بيت فعال ساز وقفه ، برنامه به ISR سرريز تايمر منشعب شده و بيت پرچم وقفه به وسيله سخت افزار صفر مي شود .
مسئله ي مهم اين است كه در هر دو حالت شمارش افزايشي و كاهشي عمل مقايسه بين رجيسترهاي TCNT0 وOCR0 انجام مي گيرد و در صورت برابري پرچم OCF0 يك شده و باعث تغيير در پين OC0 می شود .



تغيير پين OC0 مطابق جدول زير مي باشد:

COM01	COM00	وضعيت پين OC0
0	0	غير فعال (I/O معمولی)
0	1	رزرو شده
1	0	Clear در وضعيت تطابق، در زمان شمارش افزايشي (PWM غير معكوس) Set در وضعيت تطابق، در زمان شمارش كاهشي
1	1	Set در وضعيت تطابق، در زمان شمارش افزايشي (PWM معكوس) Clear در وضعيت تطابق، در زمان شمارش كاهشي

در صورت تغيير مقدار رجيستر OCR0 مقدار جديد در بافر اين رجيستر نوشته مي شود و در زمان رسيدن به 0xFF رجيستر OCR0 به روز مي شود.

پرچم سرريز تايمر صفر زماني فعال مي شود كه كه رجيستر TCNT0 برابر صفر شود و نه 0xFF بنابراين بايد دقت داشت كه اگر در زمان شروع به كار مقدار اين رجيستر صفر باشد پرچم سرريز فعال خواهد شد .

فركانس PWM در حالت Phase Correct از رابطه ي زير قابل محاسبه است:

$$F_{PWM} = \frac{F_{CLK}}{N.510} \qquad \text{N = 1, 8, 64, 256, 1024}$$

رابطه ي بالا نشان مي دهد فركانس موج PWM ثابت است و ارتباطي به رجيسترهاي OCR0 وTCNT0 ندارد .

در حالت PWM غير معكوس با افزايش مقدار OCR0 مقدار متوسط موج PWM افزايش يافته و با كاهش آن مقدار متوسط كاهش مي يابد و در حالت PWM معكوس ، عكس اين قضيه صحيح است.

عملكرد تايمر دو

به طور كلي عملكرد تايمر 2 مشابه تايمر صفر مي باشد و رجيسترهاي مربوطه با همان نام و داراي پسوند 2 مي باشند ، با اين تفاوت كه تايمر 2 برخلاف تايمرهاي صفر و يك نمي تواند از پايه خارجي T0 يا T1 كلاك دريافت كند و در عوض مي توان با وصل كردن يك كريستال 32.768 كيلوهرتز به پين هاي TOSC1 وTOSC2 از آن در وضعيت آسنكرون جهت RTC استفاده نمود . از آنجايي كه تايمر 2 داراي Prescaler مجزا از دو تايمر 0 و1 مي باشد با تقسيم كريستال 32768 هرتز بر 128 مي توان به زمان سرريز 1 ثانيه كه مناسب براي عملكرد ساعت است دست پيدا كرد .

تنظيمات Prescale براي اين تايمر به صورت زير مي باشد

وضعيت منبع كلاك تايمر	CS00	CS01	CS02
بدون كلاك (متوقف)	0	0	0
كلاك سيستم (بدون تقسيم)	1	0	0

8/کلاك سيستم	0	1	0
32/کلاك سيستم	1	1	0
64/کلاك سيستم	0	0	1
128/کلاك سيستم	1	0	1
256/کلاك سيستم	0	1	1
1024/کلاك سيستم	1	1	1

پيكربندی RTC با رجيستر وضعيت آسنكرون يا ASSR انجام مي شود:

Bit	7	6	5	4	3	2	1	0
ASSR					AS0	TCN2UB	OCR2UB	TCR2UB

Asynchronous Timer/Counter2 : با Set كردن اين بيت منبع كلاك تايمر 2 از كلاك سيستم به كريستال خارجي در پايه هاي TOSC1 وTOSC2 تغيير مي كند. با تغيير دادن اين بيت ممكن است مقدار رجيسترهاي TCNT2 ، OCR2 و TCCR2 خراب شود .

Timer/Counter2 Update Busy : براي تضمين عملكرد صحيح در وضعيت آسنكرون رجيسترهاي تايمر 2 برخلاف تايمر 0 و 1 به صورت بافر شده بروز مي شوند . بدين ترتيب كه

وقتي روي رجيستر TCNT2 مقداري نوشته شود ، بيت TCN2UB يك مي شود و مقداري كه در رجيستر موقتي ذخيره شده است به TCNT2 منتقل مي شود . با اتمام بروز رساني TCNT2 اين بيت توسط سخت افزار صفر مي شود . صفر بودن OCR2UB نشان دهنده ي آمادگي TCNT2 براي پذيرفتن مقدار جديد است.

Output Compare Register2 Update Busy : اين بيت همانند TCN2UB بوده با اين تفاوت كه بر روي رجيسترOCR2 عمل مي كند.

Timer/Counter Control Register2 Update Busy : این بیت همانند TCN2UB بوده با این تفاوت که بر روي رجیسترTCCR2 عمل مي كند.

در حالتي كه پرچم مشغول بودن يك رجيستر يك مي باشد، نوشتن بر روي آن رجيستر باعث مي شود كه مقدار بروز شده صحيح نباشد و ممكن است باعث وقفه ي ناخواسته شود.

مكانيسم خواندن اين سه رجيستر متفاوت مي باشد ، بدين صور ت كه زمان خواندن TCNT2 مقدار خود رجيستر خوانده شده و با خواندن OCR2 و TCCR2 مقدار موجود در رجيستر موقت خوانده مي شود.

تايمر 2 در وضعيت آسنكرون در حالت Power-Save نيز فعال بوده و پس از سرريز شدن تايمر از وضعيت Power-Save خارج شده و در فعال بودن وقفه ، ISR را اجرا نموده و مجددا وارد حالت Power-Save مي شود.

تايمر/كانتر يك

تايمر يك تايمري 16 بيتي است و در آن علاوه بر امكانات تايمر صفر، يك بخش ديگر به نام بخش Capture به آن افزوده شده است . اين بخش در زمان هاي خاص ، عدد شمارش شده توسط تايمر يك و زمان سپري شده را ثبت كرده و از طريق آن امكان اندازه گيري هاي زماني را

فراهم مي آورد . تايمر يك داراي پنج مد كاری به نام هاي Normal ، CTC ، Fast PWM ، Phase Correct PWM وPhase and Mode Frequency Correct مي باشد . مدهای

PWM در تايمر 1 بسيار متنوع و داراي 12 حالت PWM می باشد . در اين تايمر پين T1 به عنوان ورودي كانتر و پين هاي OC1A و OC1B به عنوان خروجي مقايسه گر عمل مي كنند .همچنين پين ICP1 براي ورودي بخش Capture تايمر يک می باشد .

به علت 16 بيتي بودن تايمر، رجيسترهاي TCNT1 ، OCR1A و OCR1B شانزده بيتي مي باشند كه هر كدام داراى دو بايت L وH هستند . همچنين تايمر يك داراي دو واحد مقايسه ي مجزا

مي باشد كه مقدار موجود در رجيسترهاي OCR1A و OCR1B را با TCNT1 مقايسه كرده و در صورت برابري وضعيت پين هاي OC1A وOC1B را تغيير مي دهند . همچنين رجيستر ICR1 نيز كه رجيستر واحد Capture است رجيستري 16 بيتى مى باشد .

رجيسترهاي 8 بيتي TCCR1A و TCCR1B كنترل تايمر را بر عهده دارند:

TCCR1A	7	6	5	4	3	2	1	0
نام بيت	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10

TCCR1B	7	6	5	4	3	2	1	0
نام بيت	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10

مد كاري تايمر بوسيله ي بيت هاي جدول زير تعيين مى شود :

	WGM13	WGM12	WGM11	WGM10	مد کاری	TOP	TOV=1
0	0	0	0	0	Normal	0xFFFF	0
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	0
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	0
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	0
4	0	1	0	0	CTC	OCR1A	0xFFFF
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICR1	0
9	1	0	0	1	PWM, Phase and Frequency Correct	OCR1A	0
10	1	0	1	0	PWM, Phase Correct	ICR1	0
11	1	0	1	1	PWM, Phase Correct	OCR1A	0
12	1	1	0	0	CTC	ICR1	0xFFFF

13	1	1	0	1	رزرو شده	-	-
14	1	1	1	0	Fast PWM	ICR1	TOP
15	1	1	1	1	Fast PWM	OCR1A	TOP

تعريف TOP : تايمر وقتی به مقدار TOP مي رس د که برابر با بالاترين مقدار در رشته ي شمارش خود است . اين مقدار مي تواند مقادير ثابتي مثل 0x03FF بوده و يا مقدار نگهداري شده در يكي از رجيسترهاي OCR1A يا ICR1 باشد .

FOC1A و FOC1B: بيت های Force بخش مقايسه گر که عملكرد آن ها همانند FOC1 در تايمر صفر و دو مي باشد . به اين ترتيب که در مد های غير PWM يك كردن اين بيت بدون اينكه وقفه اي ايجاد كند در صورت تطبيق مقايسه ، باعث تغيير وضعيت پين هاي OC1A وOC1B مطابق با وضعيت بيت هاي COM در TCCR1 می شود .

بيت های COM1A0 ، COM1A1 و COM1B0 ،COM1B1 : تغيير وضعيت پين هاي OC1A وOC1B را در حالت تطبيق معين مي كنند که مقدار آن ها بسته به مد كاري عملكرد متفاوتي را ايجاد مي كند .

بيت های زير برای تعيين منبع كلاک می باشند :

وضعيت منبع كلاك تايمر	CS00	CS01	CS02
بدون كلاك (متوقف)	0	0	0

كلاك سيستم (بدون تقسيم)	1	0	0
8/كلاك سيستم	0	1	0
64/كلاك سيستم	1	1	0
256/كلاك سيستم	0	0	1
1024/كلاك سيستم	1	0	1
لبه ي پايين رونده ي پالس خارجي (T1)	0	1	1
لبه ي بالا رونده ي پالس خارجي (T1)	1	1	1

بيت ICES1: بيت تعيين لبه ي ورودي بخش Capture از پين ICP1 . با صفر بودن اين بيت لبه ي پايين رونده و با يك بودن آن لبه ي بالا رونده باعث تريگر مي شود.

بيت ICNC1: بيت فعال ساز حذف نويز در ورودي پين ICP1

نتايج حاصل از كاركرد تايمر 1 در 4 بيت از رجيستر TIFR به نام هاى TOV1 (پرچم سرريز)

OCF1A (پرچم تطابق مقايسه گر A) OCF1B (پرچم تطابق مقايسه گر B) و ICF1 (پرچم بخش Capture تايمر يک) منعكس مى شوند :

TIFR	7	6	5	4	3	2	1	0
نام بيت	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
سطح منطقى	0	0	X	X	X	X	0	0

يك شدن هر يك از اين پرچم ها در صورت فعال بودن بيت فعال ساز عمومي و فعال بودن وقفه ي مربوطه در رجيستر TIMSK مي تواند باعث انشعاب برنامه به ISR مربوط به آن وقفه شود:

TIMSK	7	6	5	4	3	2	1	0
نام بيت	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0
سطح منطقی	0	0	X	X	X	X	0	0

با اجرا شدن ISR به صورت خودكار بيت پرچم وقفه صفر شده و يا مي تواند با نوشتن يك روي آن به وسيله ي نرم افزار آن را پاك كرد.

1. مد Normal

اين مد همانند مشابه آن در تايمر صفر مي باشد با اين تفاوت كه تايمر تا عدد 0xFFFF شمارش كرده و با رسيدن به آن تايمر سرريز كرده و بيت TOV1 يك شده و در صورت فعال بودن وقفه مي تواند

باعث اجراي ISR مربوطه شود . در مدعادي هر دو مقايسه گر A و B فعال بوده و هر كدام به طور مستقل عمل مقايسه را روي رجيستر TCNT1 وOCR1A و OCR1B انجام مي دهند . در صورت برابري بيت OCF1A يا OCF1B يك شده و خروجي OC1A يا OC1B مطابق جدول زير تغيير وضعيت داده و در صورت فعال بودن وقفه مي توانند باعث ايجاد وقفه شوند.

وضعيت پين OC1A يا OC1B	COM1A0/COM1B0	COM1A1/COM1B1
غير فعال (I/O معمولی)	0	0

Toggle	1	0
Clear	0	1
Set	1	1

در صورت استفاده ازOC1A يا OC1B براي توليد شكل موج ، بايد اين پين ها به صورت خروجي پيكربندي شوند.

2. مد CTC

در اين حالت مقدار رجيستر TCNT1 به طور پيوسته با مقدار ر جيستر OCR1A يا ICR1 مقايسه مي شود و در صورت برابري مقدار رجيستر TCNT1 برابر صفر مي شود . بنابراين در اين حالت مقدار TOP تايمر را با توجه به مقدار موجود در بيت هاي WGM مقدار رجيسترهاي OCR1A يا ICR1 تعيين مي كنند.

با رسيدن تايمر به مقدار TOP خود بر حسب اينكه مقدار ماكزيمم OCR1A يا ICR1 انتخاب شده باشد به ترتيب پرچم هاي OCF1A يا ICF1 يك شده و در صورت فعال بودن وقفه از آن
مي توان براي تغيير دادن مقدار مقايسه استفاده كرد . اين عمل بايد با دقت صورت گيرد زيرا رجيستر مقايسه ي تايمرها فقط در مدهای PWM داراي بافر دو بل مي باشند . در اين حالت فركانس موج ايجاد شده روي پايه هاي OC1A يا OC1B مطابق رابطه ي زير مي باشد:

$$F_{OC1X} = \frac{F_{CLK-I/O}}{2N(1+OCR1A)}$$

3. مد Fast PWM

بر خلاف تايمرهاي صفر و دو كه در آن موج هاي PWM توليد شده داراي دقت ثابت 8بيتي هستند، تا يمر1 قادر است سيگنال هاي PWM اي با دقت متغير را ارائه كند، اين مسئله باعث

مي شود که کاربر بتواند علاوه برتغيير Duty Cycle فرکانس موج را به صورت سخت افزاري کنترل کند .

PWM سريع داراى پنج مد سريع مى باشد: (5 ، 6 ، 7 ، 14 و15 = [3:0]WGM1)

	PWM سريع
1	8 بيتى (TOP = 0Xff)
2	9 بيتى (TOP = 0x1FF)
3	10 بيتى (TOP = 0x03FF)
4	با TOP = ICR1
5	با TOP= OCR1A

در اين مد تايمر از صفر تا مقدار TOP خود شروع به شمارش کرده و پس از از سرريز مجددا از صفر شروع به کار مى کند . در صورتي که مقايسه ي خروجي در حالت PWM غير
معکوس باشد در حالت تطبيق مقايسه بين رجيسترهاي TCNT1 و OCR1x پين OC1x يك شده و با رسيدن به مقدار TOP پاك مي شود . در صورتي که خروجي PWM معکوس باشد وضعيتي عکس وجود خواهد داشت . دقت موج PWM خروجي مي تواند مقادير ثابت 8 ، 9 يا 10 بيتي داشته ويا به وسيله ي رجيسترهاي ICR1 يا OCR1A به مقدار دلخواه تنظيم شود . در اين حالت حداقل مقدار مجاز 2 بيت (با دادن مقدار 0x0003 به رجيسترهاى ICR1 يا OCR1x)
و حداکثر آن 16 بيت مي باشد .

دقت موج PWM بر حسب مقدار ماکزيمم از رابطه ي زير به دست مي آيد:

$$\mathrm{Re}solution = \frac{\log(TOP+1)}{\log(2)}$$

با رسيدن تايمر به مقدار TOP پرچم سرريز TOV1 فعال شده و با تطبيق مقايسه نيز بيت OCF1A يا OCF1B يك مي شود . در اين حالت ها اگر وقفه ي مربوطه فعال شده باشد مي تو ان درISR آن وقفه مقدار مقايسه را تغيير داد . بايد توجه داشت كه مقدار رجيسترهاي مقايسه بايد از مقدار TOP كمتر باشد در غير اين صورت هيچ گاه مقايسه اي صورت نمي گيرد.

تغيير وضعيت پين ها در حالت تطبيق مقايسه و سرريز مطابق جدول زير خواهد بود:

وضعيت پين OC1A يا OC1B	COM1A0/COM1B0	COM1A1/COM1B1
غير فعال (I/O معمولی)	0	0
اگر WGM1[3:0]=15 باشد ؛ Toggle پين OC1A در وضعيت تطابق و OC1B پين I/O معمولی براي ديگر حالت هاي WGM1[3:0]: غير فعال (I/O معمولی)	1	0
Clear در وضعيت تطابق وSet در وضعيت TOP (PWM غير معكوس)	0	1
Set در وضعيت تطابق وClear در وضعيت TOP (PWM معكوس)	1	1

فرکانس موج PWM حاصل از رابطه زیر بدست می آید :

$$F_{PWM} = \frac{F_{CLK\_I/O}}{N(1+TOP)}$$

4. مد Phase Correct PWM

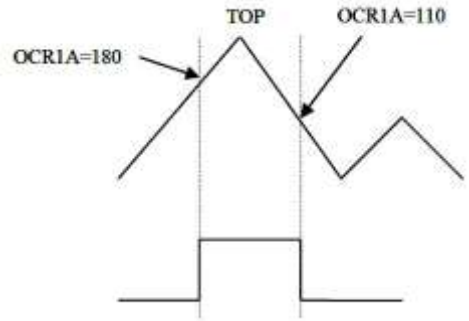
PWM تصحیح فاز داراي پنج مد کاری می باشد : (1 ، 2 ، 3 ، 10 و11 [WGM1[3:0 =)

	PWM تصحیح فاز
1	8 بیتی (TOP = 0Xff)
2	9 بیتی (TOP = 0x1FF)
3	10 بیتی (TOP = 0x03FF)
4	با TOP = ICR1
5	با TOP= OCR1A

در این مد تایمر به طور پیوسته از مقدار صفر تا TOP وازTOP تا صفر مي شمارد . در حالت PWM غيرمعكوس در حالي كه تايمر به صورت صعودي مي شمارد در لحظه ي برابري رجيسترهاي TCNT1 وOCR1 پين OC1x صفر شده و در حالت شمارش نزولي با تطابق دو رجيستر اين پين يك مي شود . در حالت PWM معكوس، عكس اين قضيه برقرار است.

دقت موج PWM خروجي مي تواند مقادير ثابت 8، 9 يا 10 بيتي داشته و يا به وسيله ي رجيسترهاي ICR1 يا OCR1A به مقدار دلخواه تنظيم شود . در اين حالت حداقل مقدار مجاز
2 بيت و حداكثر آن 16 بيت مي باشد .

پرچم سرريز تايمر TOV1 با رسيدن تايمر به مقدار صفر يك خواهد شد و با تطبيق مقايسه نيز بيت OCF1A يا OCF1B يك مي شود . در اين حالت ها اگر وقفه ي مربوطه فعال شده باشد برنامه مي تواند به ISR آن وقفه منشعب شود . مقدار مقايسه (OCRx) را در ISR يا هر زمان ديگر مي توان تغيير داد اما اين مقدار در بافر رجيسترهای OCR1A و OCR1B
ذخيره شده و با رسيدن تايمر به مقدار TOP در خود رجيستر Load می شود بنابراين تغيير
دادن مقدار رجيسترهاي OCR1x به دليل تغيير آن با رسيدن به TOP مي تواند باعث
خروجی PWM نامتقارن شود :



مشكل بالا در PWM تصحيح فاز و فركانس با بروز كردن رجيسترهاي OCR1x در زمان رسيدن به صفر، حل مي شود.

تغيير وضعيت پين ها در حالت تطبيق مقايسه و سرريز به صورت جدول زير خواهد بود :

وضعيت پين OC1A يا OC1B	COM1A0	COM1A

	/COM1B0	1/COM1B1
غير فعال (I/O معمولی)	0	0
اگر WGM1[3:0]=9,14 باشد ؛ Toggle پين OC1A در وضعيت تطابق و OC1B پين I/O معمولی براي ديگر حالت هاي WGM1[3:0]: غير فعال (I/O معمولی)	1	0
Clear در وضعيت تطابق وشمارش صعودی ، Set در وضعيت تطابق وشمارش نزولی	0	1
Set در وضعيت تطابق وشمارش صعودی ، Clear در وضعيت تطابق وشمارش نزولی	1	1

5. مد Phase and Frequency Correct

همانطور كه گفته شد به دليل بروز كردن رجيستر OCR1x با رسيدن به TOP ممكن است شكل موج خروجي نامتقارن شود

بنابراين براي حل اين مشكل مد پنجم تايمر يك اين رجيستر را با رسيدن به صفر بروز مي كند. تفاوت ديگر اين مد و عملكرد قبلي در اين است كه تايمر تنها در دو حالت زير كار مي كند:

1. PWM تصحيح فاز و فركانس با ICR1 - TOP

2. PWM تصحيح فاز و فركانس با OCR1A - TOP

واحد Capture تايمر يك

عملكرد اين واحد به اين صورت است كه در اثر تريگر شدن ورودي Capture از پين ICP1 يا خروجي مقايسه گر آنالوگ مقدار موجود در رجيسترTCNT1 در رجيستر ICR1 نوشته شده و همزمان پرچم Capture تايمر يک (ICF1) يك مي شود . در اين زمان در صورت فعال بودن بيت پرچم ورودي Capture (TICIE1) اين تريگر

شدن مي تواند باعث ايجاد وقفه شود

با اجر ا شدن ISR به طور خودكار بيت ICF1 صفر شده و يا در صورت فعال نبودن وقفه مي تواند با نوشتن يك بر روي آن پاك شود.

Bit	7	6	5	4	3	2	1	0
ICR1H	ICR1[15:8]							
ICR1L	ICR1[7:0]							

رجيستر ICR1 به جز در حالتي كه به عنوان TOP جهت مقايسه به كار مي رود (مدهای 8 ،10، 12و14) يك رجيستر فقط خواندني است.

همان طور كه گفته شد تريگر شدن واحد Capture مي تواند از دو منبع مختلف صورت گيرد
كه اين از طريق بيت ACIC در رجيستر ACSR صورت مي گيرد . صفر بودن اين بيت پين ICP1 و يك بودن آن خروجي مقايسه كننده ي آنالوگ را انتخاب مي كند .
همچنين نوع سيگنال ورودي از پين ICP1 بوسيله بيت ICES1 از رجيستر TCCR1B تعيين مي شود ، به اين ترتيب كه صفر بودن اين بيت لبه ي پايين رونده و يك بودن آن لبه ي بالا رونده ي سيگنال ورودي را انتخاب مي كند.

TCCR1B	7	6	5	4	3	2	1	0
نام بيت	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10

ورودی capture داراي يك واحد كاهش نويز نيز مي باشد كه با استفاده از يك فيلتر ديجيتال ايمني ورودي را بهبود مي بخشد . اين واحد با يك كردن بيت ICNC1

از رجيستر TCCR1B فعال مي شود . با فعال شدن اين فيلتر بايد سيگنال نمونه برداري شده روي پايه ي ICP1 براي چهار سيكل كلاك معتبر باشد.

ضميمه چهارم :

ADC &DAC

:ADC

راه اندازی یک ADC:

تراشه مبدل آنالوگ به دیجیتال دوازده بیتی ما AD1674 ساخت شرکت Analog devices است. شکل این تراشه به همراه نام پایه ها در زیر آمده است:

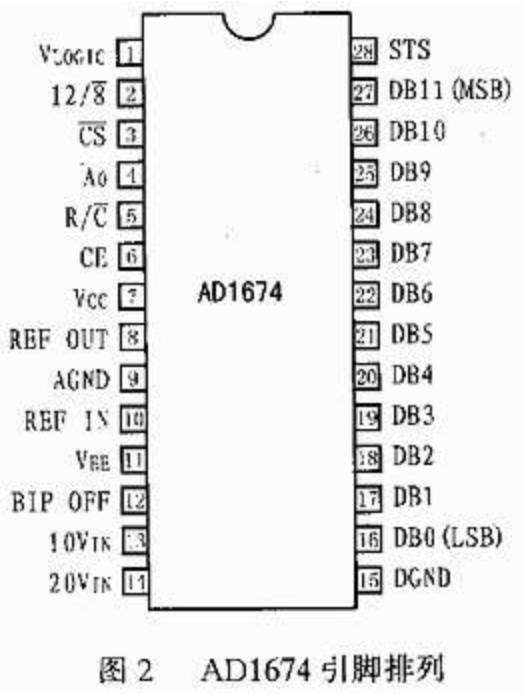


图 2　AD1674 引脚排列

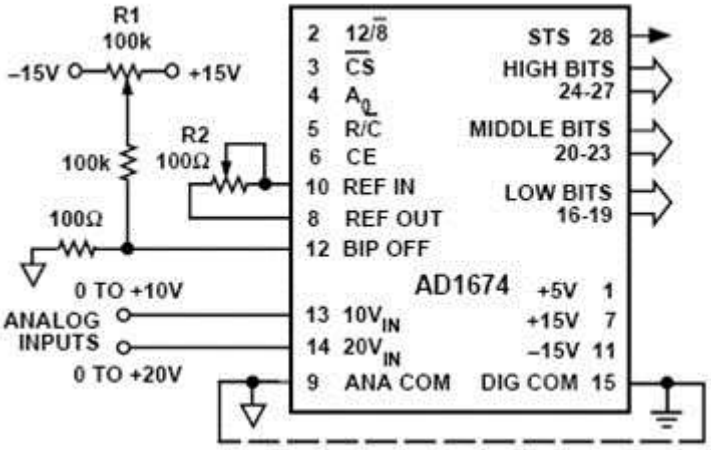
این ای سی دارای 3 نوع تغذیه می باشد. پایه شماره 1 آن سطح ولتاژ منطقی می باشد که بایستی به آن ولتاژی بین 4.5 تا 5.5 ولت اعمال کرد. پایه شماره 7 تغذیه مثبت مدار بوده که می تواند 12 یا 15 ولت باشد که ما به آن 12 ولت را اعمال می کنیم. و پایه شماره 11 تغذیه منفی بوده که می تواند 12- یا 15- ولت باشد که باز هم ما از سطح ولتاژ 12- استفاده می کنیم. این ای سی همچنین 5 پایه کنترلی دارد پایه 6 فعال کننده چیپ بوده و اگر سطح منطقی 1 را داشته باشد ای سی فعال است.همین طور پایه 3 انتخاب گر ای سی بوده که اگر low باشد چیپ انتخاب می شود. پایه 2 یا 12/8 مشخص می کند که ما از 8 بیت در خروجی استفاده کنیم یا 12 بیت که ما این پایه را high کرده تا هر 12 بیت را در خروجی داشته باشیم پایه 5(R/C) مشخص می کند که ای سی از تبدیل را انجام دهد یا اینکه بر روی پین های خروجی اطلاعات را قرار دهد. در حالتی که این پایه high است می توان داده ها را از خروجی خواند. پایه A0 نیز در طول کار بایستی low باشد.پایه STS خروجی می باشد و وقتی که یک تبدیل انجام شد این پایه low می شود در واقع این پایه یک پرچم می باشد.از پایه 16 تا 27 نیز خروجی های دیجیتال ما بوده که کم ارزشترین آن ها پایه 16 می باشد.

همانند D/A این ای سی نیز در دو حالت تک قطبی و دو قطبی می تواند کار کند. پایه REF IN و REF OUT برای همین موضوع به کار می روند که مدار های آن ها در ادامه آورده شده است. در ضمن این ای سی می تواند با دو حالت ورودی بگیرد اگر از پایه 13 به عنوان ورودی استفاده کنیم حداکثر تا بین 0 تا 10 ولت در حالت تک قطبی یا بین 5- تا 5 ولت در حالت دوقطبی می توان به آن اعمال کرد. و اگر از پایه 14 استفاده کنیم این محدوده 20 ولت می شود.

این ای سی را می توان به روش های مختلفی کنترل کرد که ما از روش لبه پایین رونده استفاده می کنیم. در این روش تنها پایه کنترلی مورد نیاز پایه R/C می باشد. بقیه پایه ها طبق آنچه گفته شد تنظیم می شود.در این روش پایه R/C همیشه high بوده و ما در یک لحظه آن را low می کنیم که در این هنگام یک تبدیل انجام می شود. شکل پالس آن در زیر آمده است:

شکل پالس R/C در روش low pulse

شکل شماتیک مدار در حالت تک قطبی به صورت زیر است:



مدار تک قطبی

پایه شماره 12 برای کم کردن offset به صورتی که در شکل دیده می شود بسته شده است. اگر این موضوع خیلی مهم نباشد می توان این پایه را مستقیما به زمین وصل کرد. همین طور پتانسیومتر بین پایه 8 و 10 به منظور تنظیم حد بالای ولتاژ بهکار میرود یعنی اینکه چه ولتاژی را به عنوان خروجی حداکثر بگیرد در حالت معمول که این حد 10ولت است می توان یک مقاومت 50 اهمی رای به جای پتانسیومتر قرار داد.

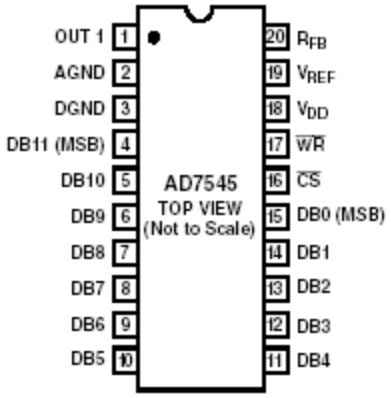
ما فعلا از مدار تک قطبی استفاده می کنیم.

برای آزمایش این IC ورودی آنالوگ را به پین 13 دادم(Vin10) . بنابراین به ازای 10 ولت ورودی آنالوگ باید xFFF0 و به ازای 0 ولت باید 0×000 بدهد.

به ورودی آنالوگ ولتاژ V5 دادم خروجی دیجیتال "011110101110" شد. به ازای V10، خروجی دیجیتال "111111111111" شد و به ازای V0، خروجی "000000000001" شد.

کار کردن با DAC 12 بیتی:

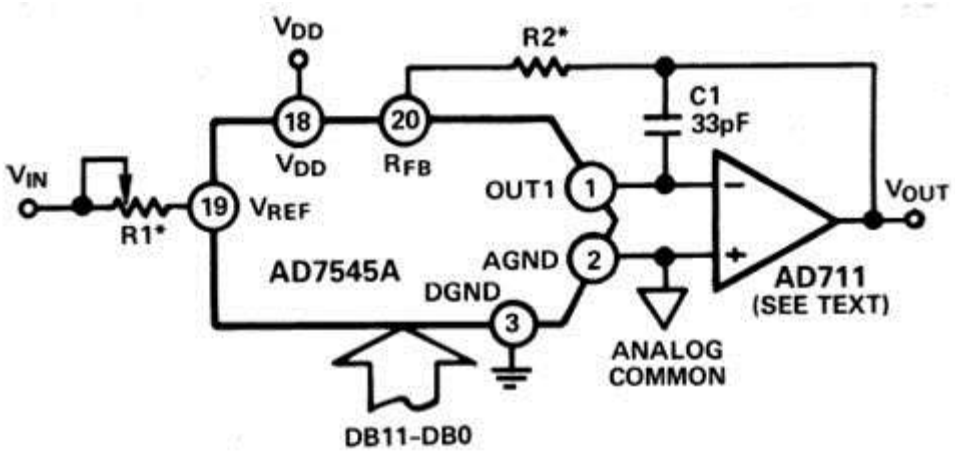
تراشه مبدل دیجیتال به آنالوگ 12 بیتی که در اینجا از آن استفاده میکنیم AD7545 ساخت شرکت Analog devices است:



برای مقداردهی به ورودی های دیجیتال این تراشه از میکرو کنترلر atmega 16 استفاده میکنیم.

این تراشه در دو مد تک قطبی و دو قطبی کار میکند. در مد تک قطبی خروجی آنالوگ فقط شامل ولتاژهای منفی میشود ولی در دوقطبی، هم ولتاژ های منفی و هم ولتاژ های مثبت را در بر میگیرد.

برای کار کردن با این تراشه در مد تک قطبی، پایه‌هاي CS, WR،DGND،AGND را به زمین ، VDD را به V5 و V_{IN} را به V10 وصل میکنیم. ورودی های دیجیتال را به پورت A و 4 بیت کم ارزش پورت B میدهیم و مدار تک قطبی (شکل زیر) را میبندیم.



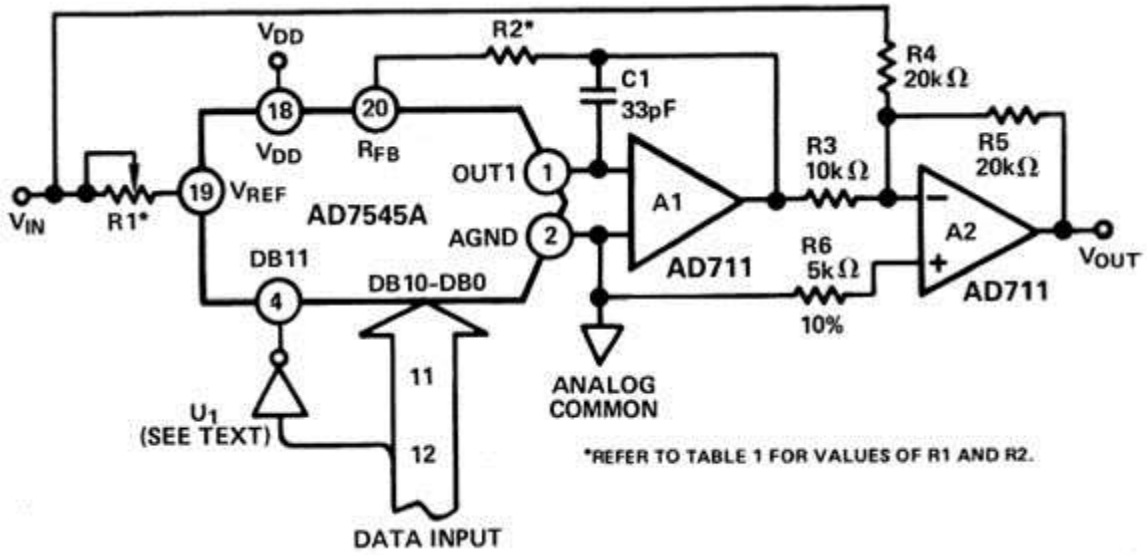
مقادیر $R_{,2}R_1$ دراین شکل در datasheet قطعه آمده است. با استفاده از میکروکنترلر به ورودیهاي دیجیتال AD7545صفر و 1 مي‌دهیم ،خروجي آنالوگ با تقریب خوبی مطابق دیتاشیت به دست مي‌آید:

0xFFF -> -8.6V

0×800 -> -5V

0×000 -> 0V

حال مدار دوقطبی(شکل زیر)را مي‌بنديم:

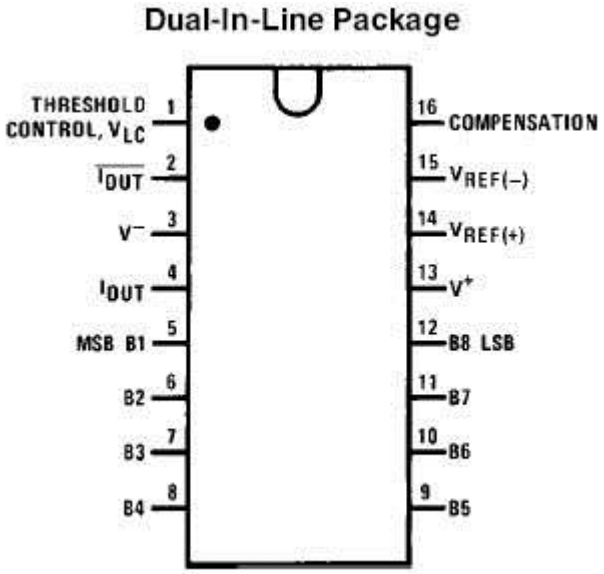


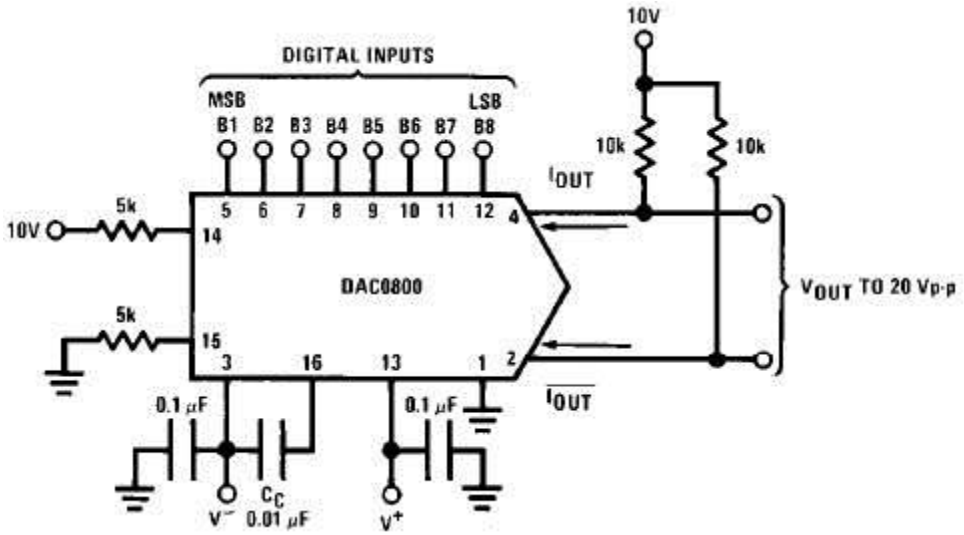
و به ورودیهاي ديجيتال مقدار ميدهيم. ميبينيم که خروجي آنالوگ تغييرمنطقي مي‌کند اما مطابق آنچه در دیتاشیت قطعه آمده نیست.برای اینکه نتایج مطابق دیتاشیت به دست آید، همانطور که در شکل بالا میبینید، پین متناظر با پرارزش ترین بیت را در میکرو کنترلر باید not کنیم وسپس به پایه 4 بدهیم.

توضیح: OP-AMP مورد استفاده در اینجا TL074 است که Quad OP-AMP می باشد

کار کردن با DAC 8 بیتی:

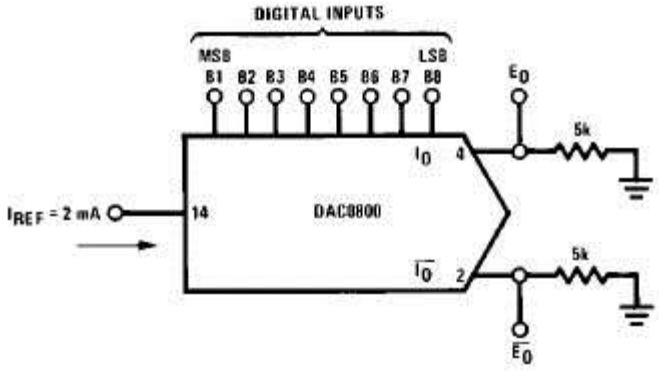
تراشه DAC0800 یک مبدل دیجیتال به آنالوگ 8 بیتی است. در شکل زیر این تراشه و همچنین مدار کاربردی سفارش شده در دیتاشیت این قطعه را می بینیم:





پایه 3 (V-) به v10- و پایه 13 (v+) به +v10 وصل می‌شود و به جای مقاومت K5 از مقاومت K4.7استفاده می‌کنیم خروجی این IC جریان است که ما برای تبدیل آن به ولتاژ از یک OP-AMP مثلاً AD648 استفاده می‌کنیم.

اما قبل از بستن مدار با OP-AMP برای اطمینان از درست کار کردن DACO800 پایه های 4 و 2 را با یک مقاومت K4.7 به زمین میبریم.



ورودیهای دیجیتال این IC را به پورت D میکرو می‌دهیم و به پورت D سه مقدار0×00 و 0×08 و xFF0 می‌دهیم و خروجی آنالوگ را اندازه می گیریم. نتایج اندازه گیری و داده های دیتاشیت را در جدول های زیر آورده ایم.

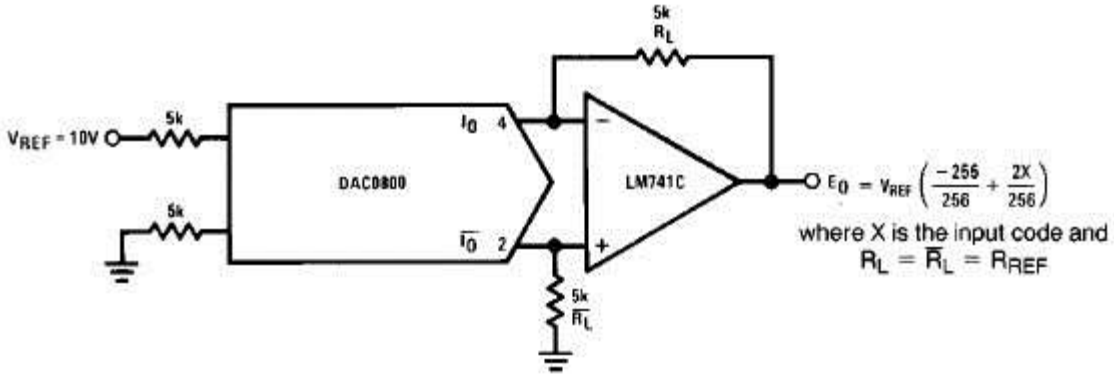
نتایج اندازه گیری:

ورودی دیجیتال	جریان پین 2	جریان پین 4	ولتاژ پین 2	ولتاژ پین 4
00×0	mA1.55-	mA0.5-	V7.3-	V2.38-
80×0	mA1.04-	mA1.05-	V4.9-	V4.98-
xFF0	mA0.5-	mA1.54-	V2.44-	V7.27-

داده های همین آزمایش در دیتاشیت:

	B1	B2	B3	B4	B5	B6	B7	B8	I_O mA	$\overline{I_O}$ mA	E_O	$\overline{E_O}$
Full Scale	1	1	1	1	1	1	1	1	1.992	0.000	−9.960	0.000
Full Scale − LSB	1	1	1	1	1	1	1	0	1.984	0.008	−9.920	−0.040
Half Scale + LSB	1	0	0	0	0	0	0	1	1.008	0.984	−5.040	−4.920
Half Scale	1	0	0	0	0	0	0	0	1.000	0.992	−5.000	−4.960
Half Scale − LSB	0	1	1	1	1	1	1	1	0.992	1.000	−4.960	−5.000
Zero Scale + LSB	0	0	0	0	0	0	0	1	0.008	1.984	−0.040	−9.920
Zero Scale	0	0	0	0	0	0	0	0	0.000	1.992	0.000	−9.960

هرچند نتایج اندازه گیری دقیقاً با مقادیری که در datasheet ذکر شده برابر نیستند اما رابطه منطقی بین ورودی دیجیتال و خروجی آنالوگ وجود دارد بنابراین از صحت کار DACO800 مطمئن می‌شویم. حال برای بدست آوردم خروجی ولتاژ سر های 2 و 4 را مطابق شکل زیر به OP-AMP متصل می‌کنیم.



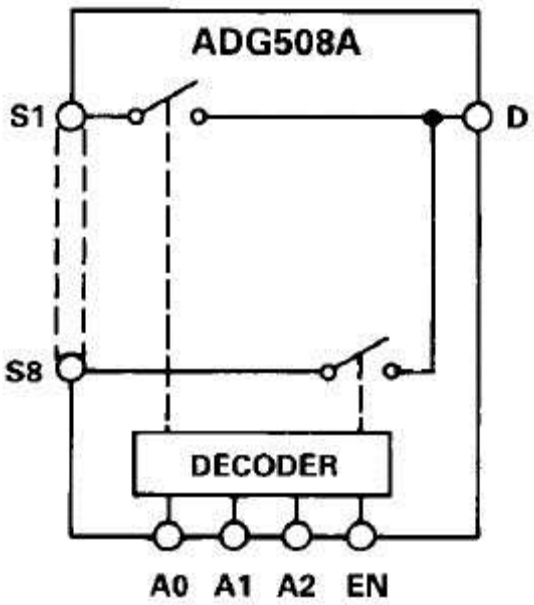
مانند دفعه قبل به ورودیهای دیجیتال مقادیر0×00 و0×80 وxFF0 می‌دهیم و خروجی آنالوگ را اندازه می‌گیریم:

خروجی آنالوگ	ورودی دیجیتال
V8.6-	00×0
V0.08	80×0
V9.28	xFF0

توضیح غیر ضروری: حتماً از درست بودن بایاس مدار مطمئن شوید.

کار کردن با یک مالتی پلکسر:

تراشه مالتی پلکسر ما ADG508 است. يك مالتي پلكر 8 به 1 كه دياگرام آن به صورت زير است.



اين IC در دو مد Dual supply , single spply كار مي‌كند.

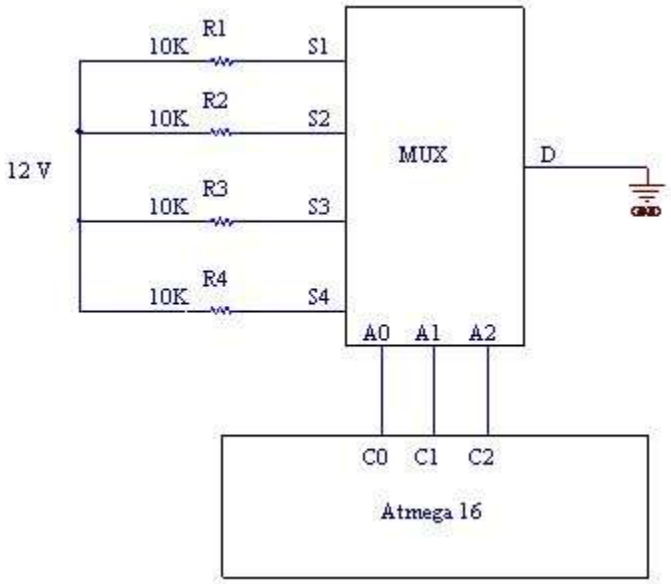
Dual supply:

v to 16.5 v10.8- =v to 16.5 v, V_{SS}10.8=V_{DD}

:Single supply

GND=OV=v to 16.5 v, V_{SS}10.8 =V_{DD}

ما دراينجا از مد Dual supply استفاده مي‌كنيم. براي آزمايش اين IC مدار زير را مي‌بنديم .



R_{on} اين IC در حدود 280 اهم است كه در مقايسه با مقاومت K10 كه ما در مدار قرار داديم قابل چشم پوشي است با استفاده از ميكرو كنترلر به پايه‌هاي $_0$A ,$_1$A ,$_2$A ورودي مي‌دهيم . جدول درستي در زير آورده شده:

A1	A0	EN	ON Switch pair
X	X	0	NONE
0	0	1	1
0	1	1	2
1	0	1	3
1	1	1	4

به اين ترتيب مثلاً هنگامي كه EN=1 و A1=0 =A0 تمام V12 دوسر مقاومت R1 افتاد و بقيه مقاومت‌ها قطع بودند.

ضمیمه پنجم : منابع

www.ATMEL.com

WWW.AVRFREAKS.NET

مجموعه کتاب های آموزش AVR

شماتیک بردهای آموزشی شرکت های تولید کننده برد های آموزشی