دانشگاه فنی و حرفه ای

آموزشکده شهید شمسی‌پور

پروژه بردآموزشی میکروکنترلرهایAVR

نگارش:

سید مصطفی احمدی سیاوشانی

استاد راهنما:

استاد شهرام رفیعی رفعت

پروژه کاردانی

در رشته برق الکترونیک

شهریور۱۳۹۳

فهرست

فصل اول : مقدمه

۱-۱- مقدمه ۱

۱-۲- اهمیت استفاده از برد آموزشی 1

۱-۳- امکانات روی برد های آموزشی 2

فصل دوم : راه اندازی قسمتهای مختلف برد

۲-۱- راه اندازی موتور پله ای(STEP-MOTOR) 4

۲-۲- راه اندازی LCDکاراکتری 6

۲-۳- راه اندازی قسمتi2cمیکرو (آی سی ساعتDS1307) 8

۲-۴- فعال سازی قسمت مقایسه کننده 10

۲-۵- فعال سازی GLCDبرد 11

۲-۶- راه اندازی ارتباط سریال برد آموزشی 13

۲-۷- راه اندازی پروتکلSPIبرد آموزشی 14

۲-۸- عکس هایی از پروژه و مراحل ساخت آن ۱۸

فصل سوم : ضمیمه یکمیکرو کنترلرATMEGA32

۳-۱- اطلاعاتی در مورد میکروکنترلرهای AVRمیکرو کنترلر 21

۳-۲- بخشهای مختلف میکروکنترلر 25

۳-۳- خانواده های میکروکنترلر 25

۳-۴- یک میکروکنترلر چگونه برنامه ریزی میشود؟ 25

۳-۵- امکانات میکرو کنترلرها 26

۳-۶- شروع کار با میکرو کنترلر 26

۳-۷- مقایسه خانواده های مختلف میکرو وکنترلرها 27

۳-۸- واژگان کلیدیAVR..... 28
۳-۹- خانواده های محصولاتAVR..... 29
۳-۱۰- ابعاد مختلف میکروهایAVR..... 32
۳-۱۱- نکات کلیدی و سودمند حافظه ی فلش خود برنامه ریز..... 33

فصل چهارم : ضمیمه دومLCDکارکتری

۴-۱- معرفیLCDکارکتری..... 34
۴-۲- معرفی خطهایlcd..... 40

فصل پنجم : ضمیمه سوم تایمر ها و کانتر ها در میکرو کنترلر

۵-۱- تایمر/کانترصفر..... 45
۵-۲- واحدCaptureتایمریک..... ۷۰

فصل ششم: ضمیمه چهارمADC&DAC

۶-۱- ADC &DAC..... ۷۲
۶-۲- راه اندازی یکADC..... ۷۲
۶-۳- کار کردن با DAC ۸بیتی..... ۷۸

منابع

فهرست شکل‌ها

شکل۲-۱ برد پروژه..... ۳

شکل۲-۲ استپ موتور و دی سی موتور..... ۵

شکل ۲-۳LCD کاراکتری..... ۷

شکل۲-۴ آی سی ساعت..... ۸

شکل۲-۵فعال سازی قسمت مقایسه کننده..... ۱۰

شکل ۲-۶ارتباط سریال برد آموزشی..... ۱۳

شکل ۲-۷ راه اندازی پروتکل SPI..... ۱۴

شکل ۲-۸عکس هایی از پروژه و مراحل ساخت آن..... ۱۸

شکل ۲-۹ عکس هایی از پروژه و مراحل ساخت آن..... ۱۹

شکل ۲-۱۰عکس هایی از پروژه و مراحل ساخت آن..... ۱۹

شکل ۲-۱۱عکس هایی از پروژه و مراحل ساخت آن..... ۲۰

شکل ۳-۱ اشکال مختلف میکروهایAVR..... ۳۲

شکل۴-۱LCDکاراکتری..... ۳۴

شکل۴-۲نحوه‌ی اتصال پایه‌های میکروکنترلر بهLCD..... ۳۶

شکل۴-۳نحوه‌ی اتصال پایه‌های میکروکنترلر بهLCD..... ۳۹

شکل۴-۴اتصال LCD به پورت چاپگر..... ۴۲

شکل۴-۵ اتصال LCD به پورت چاپگر..... ۴۳

شکل۴-۶اتصال LCD به پورت چاپگر..... ۴۴

شکل ۵-۱ مد Fast PWM..... ۵۲

شکل۵-۲ شمارش افزایشی وکاهشی عمل مقایسه بین رجیسترها..... ٥٤

شکل ۶-۱ (AD1674)..... ۷۲

شکل۶-۲ پالس R/C در روش low pulse..... ۷۴

شکل۶-۳ شماتیک مدار راه اندازی یک ADC..... ۷۴

شکل۶-۴ تراشه مبدل دیجیتال به آنالوگ ۱۲ بیتی..... ۷۵

شکل۶-۵مدار داخلی تراشه مبدل دیجیتال به آنالوگ..... ۷۶

شکل۶-۶مدار دو قطبی مبدل دیجیتال به آنالوگ..... ۷۷

شکل۶-۷ تراشه DAC0800 یک مبدل دیجیتال به آنالوگ..... ۷۸

شکل۶-۸ مبدل دیجیتال به آنالوگ..... ۷۸

شکل۶-۹ امتحان صحت کار DACO800..... ۸۰

شکل۶-۱۰ دیاگرام یک مالتی پلکر ۸ به ۱..... ۸۱

شکل۶-۱۱ آزمایش مالتی پلکسر ADG508..... ۸۲

فهرست جدول‌ها

جدول۳-۱ پیکربندی پین oc0 46

جدول۳-۲ مد عملکرد تایمر ۴۶

جدول۳-۳ مد Normal ۴۶

جدول۳-۴ وضعیت منبع کلاک باتوجه به بیت های SelectClock ٤٧

جدول۳-۵ خواندن ونوشتن ازطریق رجیستر TCNT ٤٧

جدول۳-۶ زمان سرریزتایمر ٤٨

جدول۳-۷ سرریزشدن تایمر به عنوان وقفه ٤٨

جدول۳-۸ تایمر /کانترباعملکردمقایسه ٤٨

جدول۳-۹ فراخوانی سابروتین وقفه ٤٩

جدول۳-۱۰ تغییروضعیت پین OC0 بوسیلهبیتهای COM00 و COM01 ..... ٤٩

جدول۳-۱۱ حالتISR ۵۰

جدول۳-۱۲ مقداربیت هایWGM00 و WGM01 ..... ۵۱

جدول۳-۱۳ عملکرد بیت های COM00 و COM01متفاوت ازدووضعیت قبلی ۵۳

جدول۳-۱۴ تغییر پین OC0 ..... ۵۵

جدول۳-۱۵ تنظیماتPrescaleبرای تایمر ۵۷

جدول۳-۱۶ پیکربندی RTCبارجیستر ۵۷

جدول۳-۱۷ رجیستر ICR1 ..... ۶۰

جدول ۳-۱۸ مد کاری تایمر بوسیله ی بیت ۶۱

جدول ۳-۱۹ تعیین منبع کلاک ۶۲

جدول۳-۲۰ بیتICES1 ..... ۶۳

جدول۳-۲۱ اجرا شدن ISRبه صورت خودکار ۶۳

جدول۳-۲۲مدعادی هردومقایسه گرA وB ۶۴

جدول ۳-۲۳PWM سریع دارای پنج مد سریع ۶۶

جدول۳-۲۴تغییروضعیت پین هادرحالت تطبیق مقایسه وسرریز ۶۷

جدول ۳-۲۵PWM تصحیح فاز ۶۸

جدول۳-۲۶تغییر وضعیت پین ها در حالت تطبیق مقایسه و سرریز ۶۹

جدول۳-۲۷بااجراشدنISRبه طورخودکار ۷۰

جدول۳-۲۸نوع سیگنال ورودی ازپین ICP1بوسیله بیت ICES1 از رجیستر TCCR1B..... ۷۱

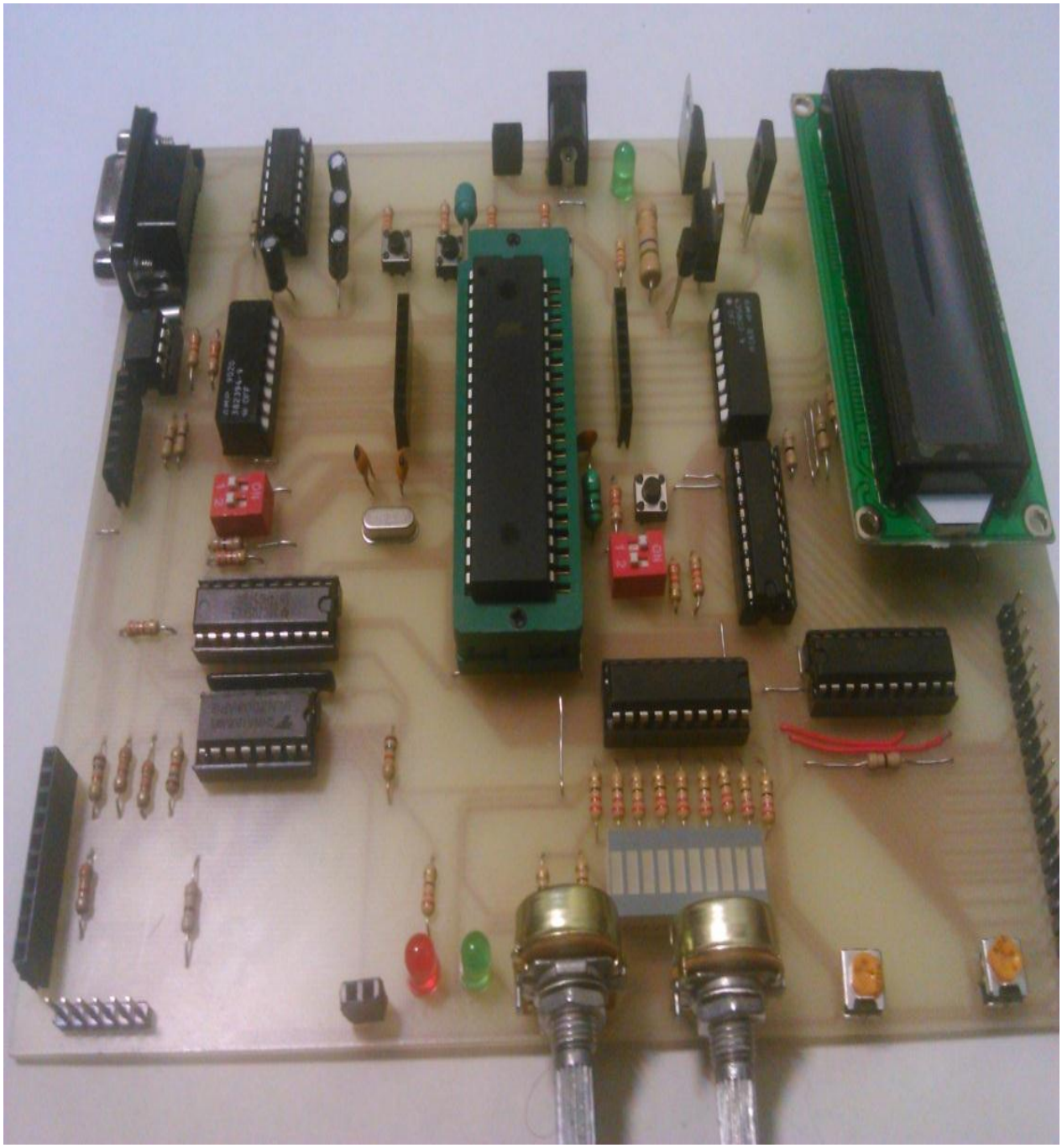
جدول۴-۱نتایج اندازه گیری کار کردن با ۸DAC بیتی..... ۷۹

جدول۴-۲ نتایج اندازه گیری کار کردن با ۸DAC بیتی در دیتاشیت..... ۸۰

جدول۴-۳ کار کردن با ۸DAC بیتی(خروجی آنالوگورودی دیجیتال)..... ۸۰

جدول۴-۴ جدول درستی یک مالتی پلکسر..... ۸۲

پروژه برد آموزشی میکروکنترلرهای AVR



فصل اول: مقدمه

1-1- مقدمه:

امروزه در اکثر پروژه های الکترونیکی میکروکنترلر ها نقش اصلی پروژه را ایفا می کنند که در این میان خانواده های AVR به دلیل امکانات بالا تنوع میکروکنترلر ها قیمت مناسب و... از محبوبیت بالایی برخوردار است پس چه بهتر که در دوره ی دانشجویی بتوان به خوبی با تمام امکانات این میکروکنترلر آشنا شد و در بازار حرفه ای کار از آن بهره گرفت همچنین درس پروژه ساخت باعث می شود دانشجو تمام آنچه 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 آموزشی AVR باعث شناخت هرچه بیشتر امکانات این خانواده از میکروکنترلر شده و باعث می شود در بازار کار بتوان از آن بهره گرفت و راحت تر پروژه های الکترونیکی را پیاده سازی نمود.

2-1- 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.

3-1- امکانات روی برد های آموزشی:

۱-دارای میکروکنترلر قدرتمند ATMEGA32

۲-نمایشگر ۲*۱۶ کاراکتری

۳-نمایشگر ۶۴*۱۲۸ گرافیکی

۴-راه انداز استپ موتور

۵- پورت سریال rs232

۶-سنسور دما

۷-مقایسه کننده آنالوگ با خروجی های مجزا

۸-دارای ۸ عدد LED

۹- ۴ عدد شستی

۱۰-راه انداز موتور DCبا PWM

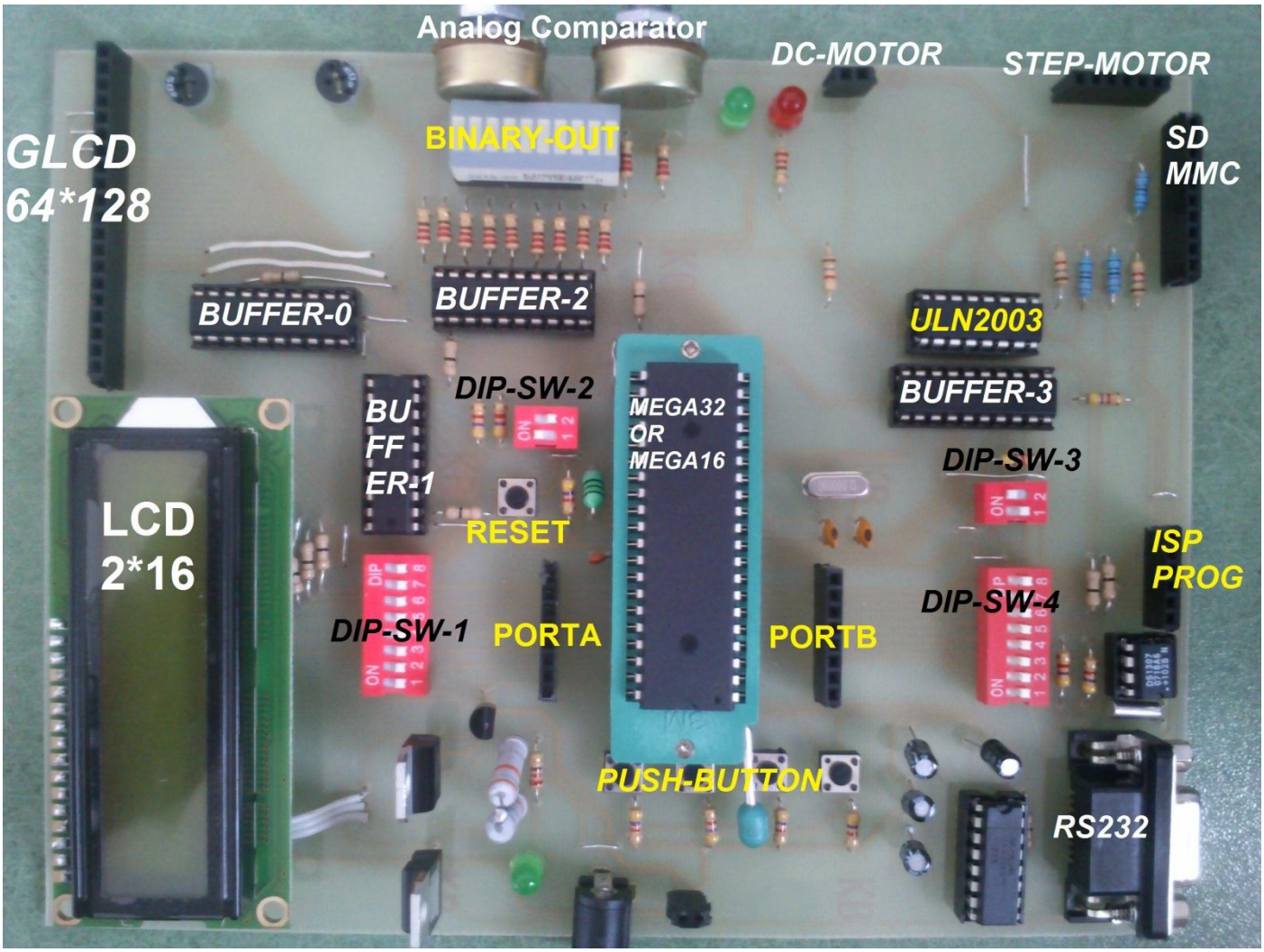
11-دارای حافظه SD

۱۲-دارای آی سی ساعات DS1307

۱۳- دارای خروجی های مجزا برای هر پورت

۱۴-قابلیت پروگرام شدن روی سیستم (ISP)

۱۵- وغیره



شکل2-۱ برد پروژه

فصل دوم :راه اندازی قسمت‌های مختلف برد

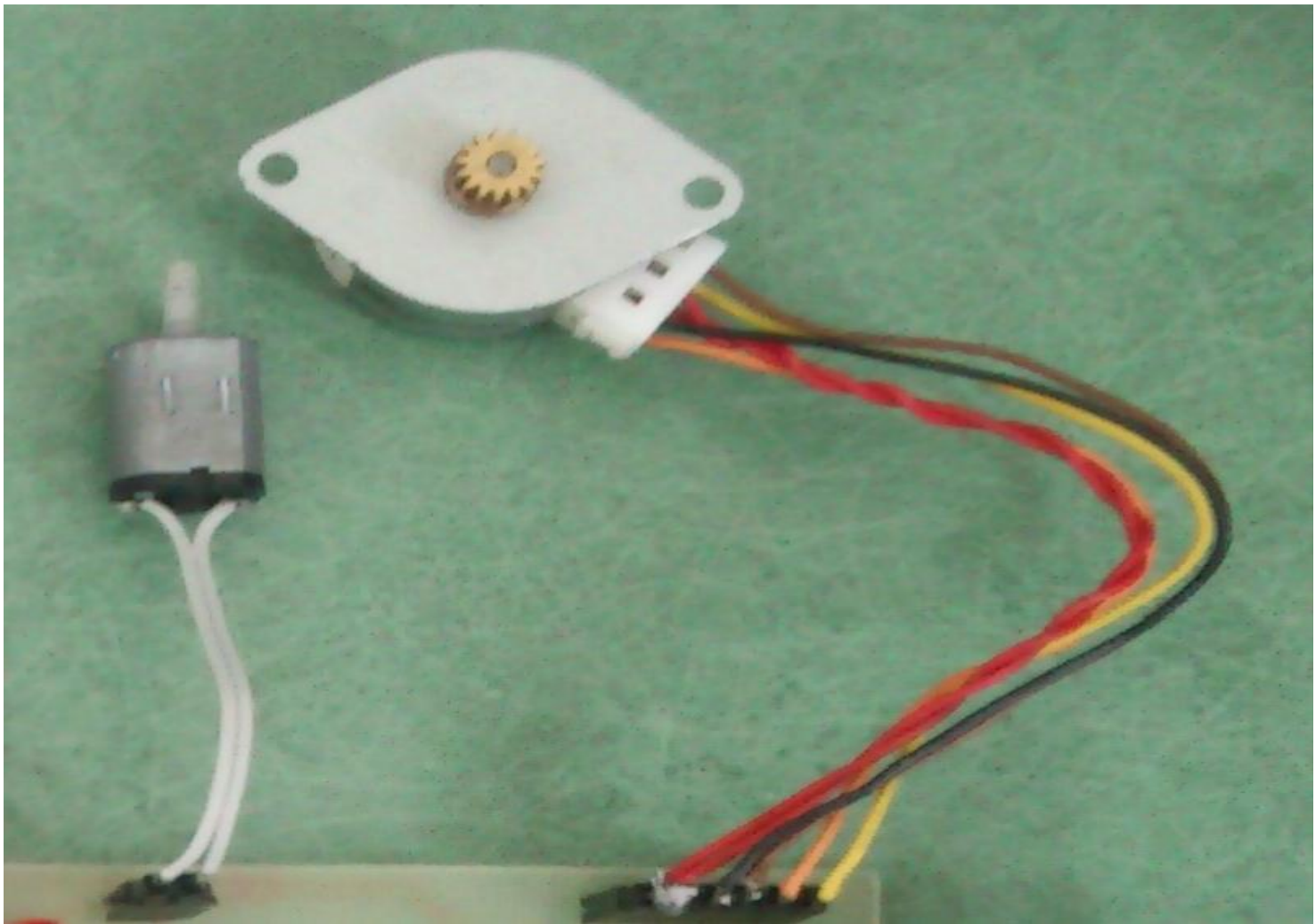
۲-۱- راه اندازی موتور پله ای(STEP-MOTOR)

اجزای 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 مینویسیم

برای استفاده از موتورها باید بافر متناظر با آن یعنی بافر ۳ باید فعال باشد و بافر ۲ خاموش .DIP-SW-3برای اینکار است.

```
#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);
}
```



}

شکل2-۲ استپ موتور و دی سی موتور

۲-۲- 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 .

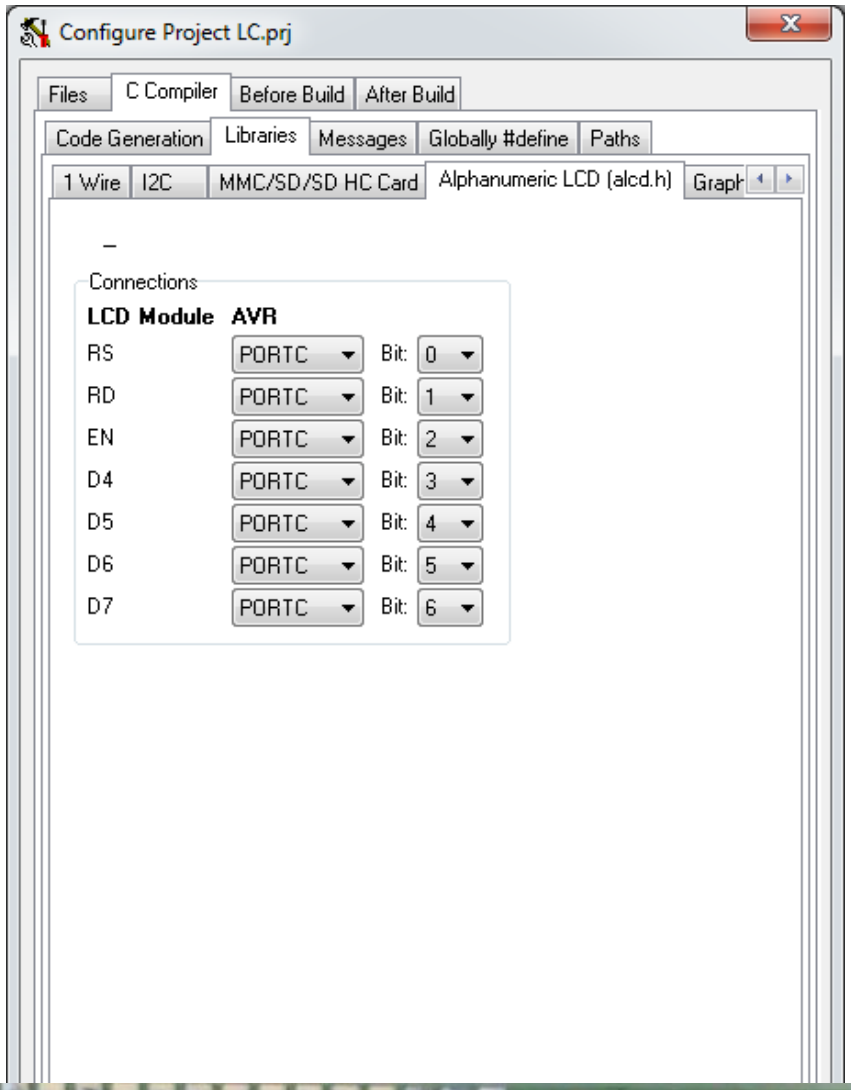
```
#include <mega32.h>
#include <alcd.h>
#include <delay.h>
#include <stdio.h>
```

برای استفاده از LCDباید بافر ۱ فعال و بافر ۰ خاموش شود .DIP-SW-2

برای اینکار است .

```
void main(void){
lcd_init(16);
lcd_clear();
```

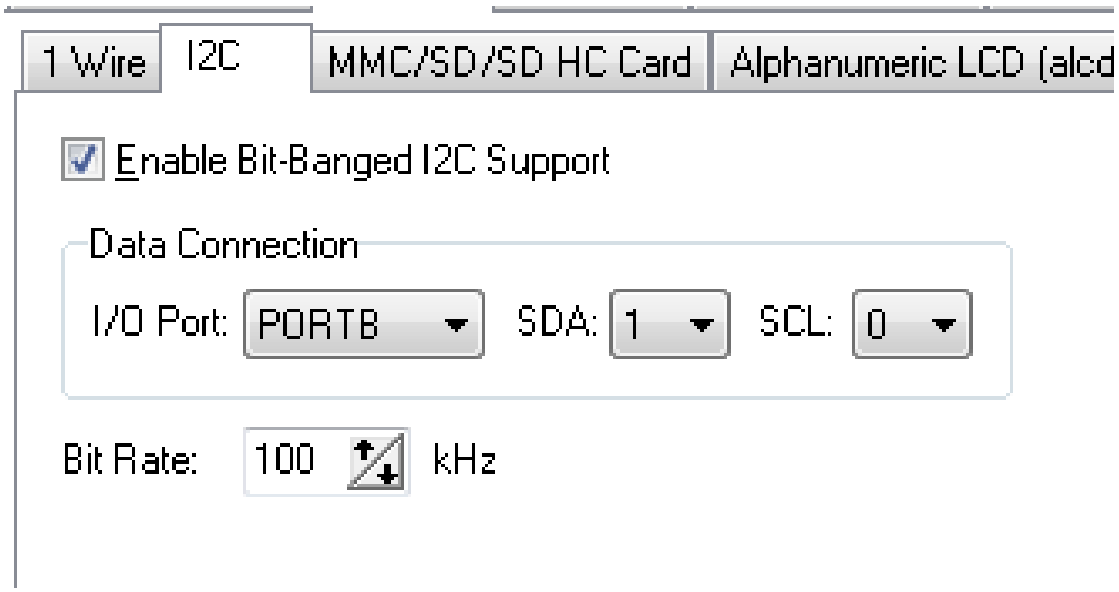
```
lcd_putsf("* Board avr v1 * ");
lcd_gotoxy(0,1);
delay_ms(1);
lcd_putsf(" *ATMEGA 32&16*");
while(1);
}
```



شکل 2-۳LCD کاراکتری

۲-۳- راه اندازی قسمت i2c میکرو (آی سی ساعت DS1307):

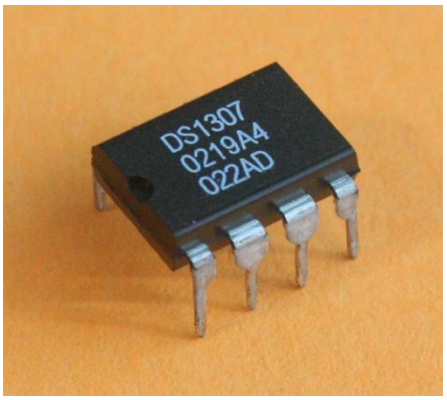
برای فعال سازی و انتخاب پایه های میکرو برای I2Cمثل شکل زیر عمل می کنیم :



دوتا پایه ۰و۱ 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...



شکل2-۴ آی سی ساعت

```
#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();
```

۲-۴- فعال سازی قسمت مقایسه کننده :

شکل ۲-۵ 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;
}        }}
```

۲-۵- 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.

```
#defineRS              PORTA.1
#defineRW              PORTA.2
#defineEN              PORTA.3
#defineCS1    PORTA.4
#defineCS2    PORTA.5
```

```
#defineRST              PORTA.6
#defineDATA PORTC
```

برای استفاده از GLCD باید بافر0 فعال و بافر 1 خاموش شود .DIP-SW-2برای اینکار است .

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

کد برنامهGLCD:

```
#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);};
}
```

۲-۶- راه اندازی ارتباط سریال برد آموزشی

شکل۲-۶ارتباط 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
```

برای اینکه بتوانیم راحتتر اطلاعات راارسالکنیماز کتابخانه STDIO استفاده می کنیم.

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

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

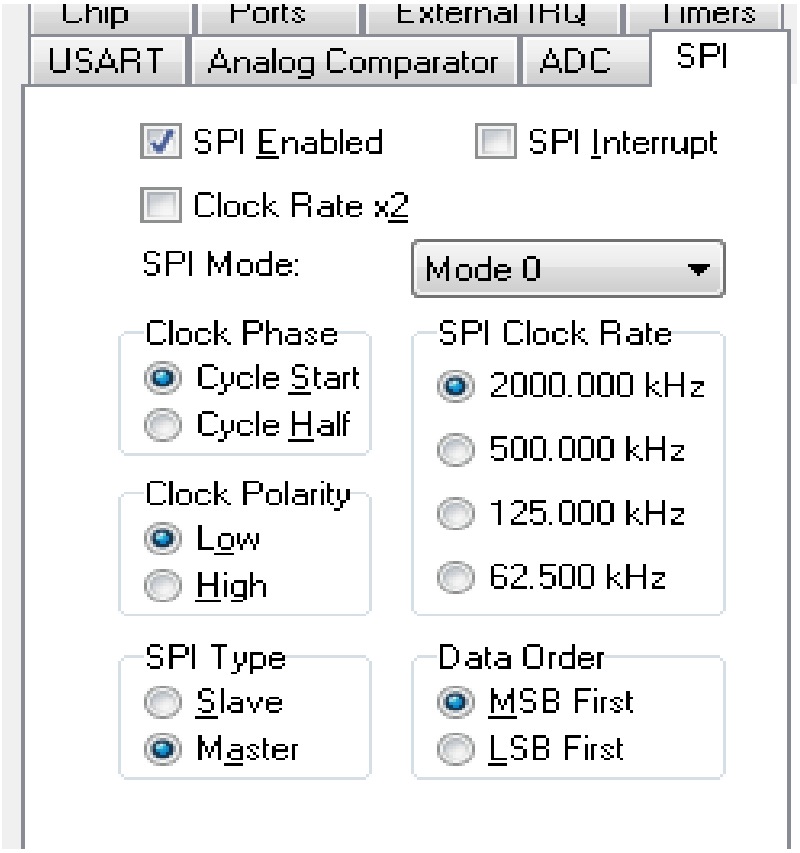
```
A=getchar()
Putchar(A);
```

۲-۷- 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 آن در میکرو تنظیم شود.



شکل ۲-۷راه اندازی پروتکل SPI

```
// 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);
```

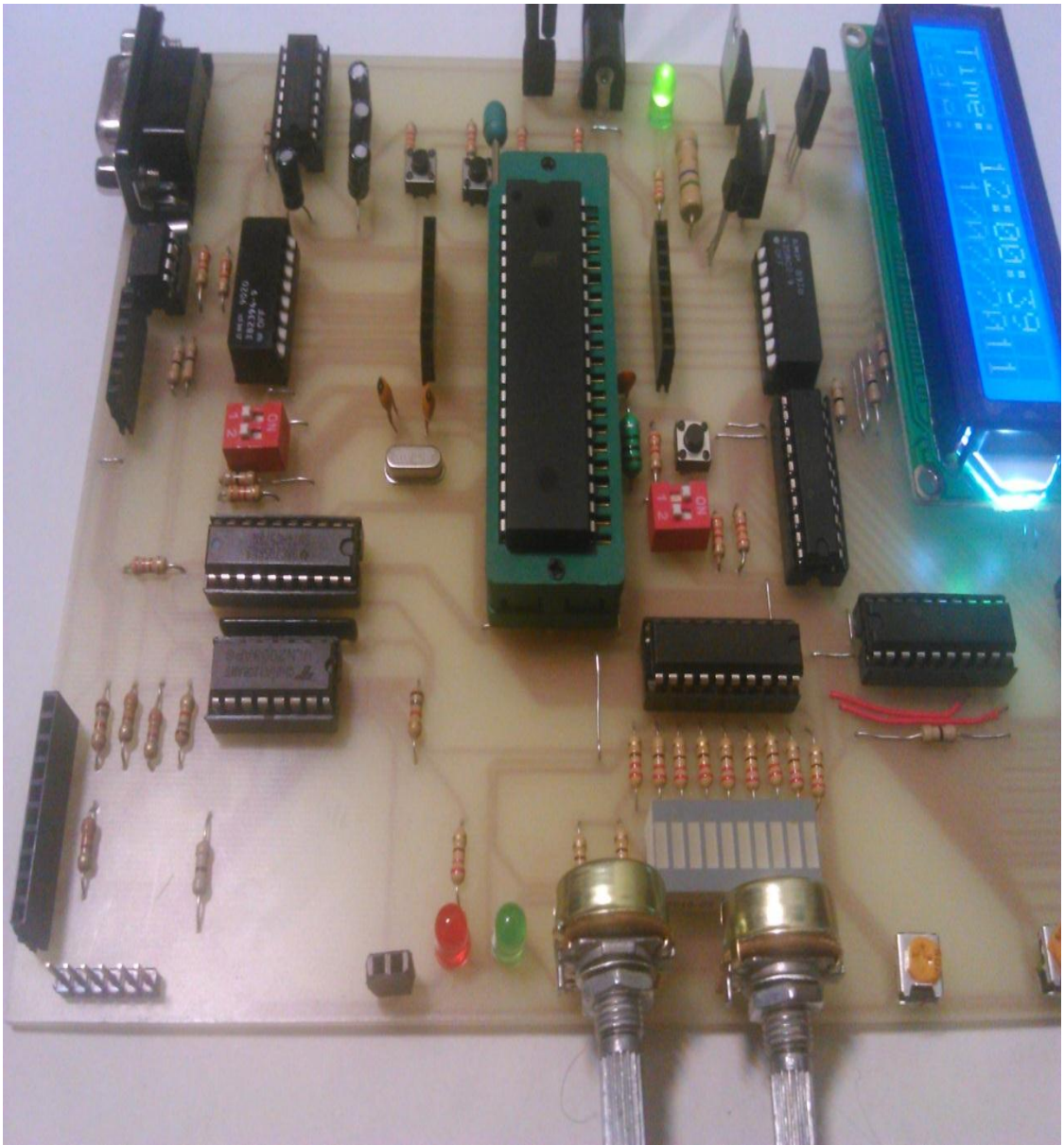
برای خواندن و نوشتن اطلاعات در mmc می توانیم از کتابخانه های موجود در اینترنت استفاده کرد مثلا برنامه زیر برای خواندن و نوشتن از حافظه mmc نوشته شده است:

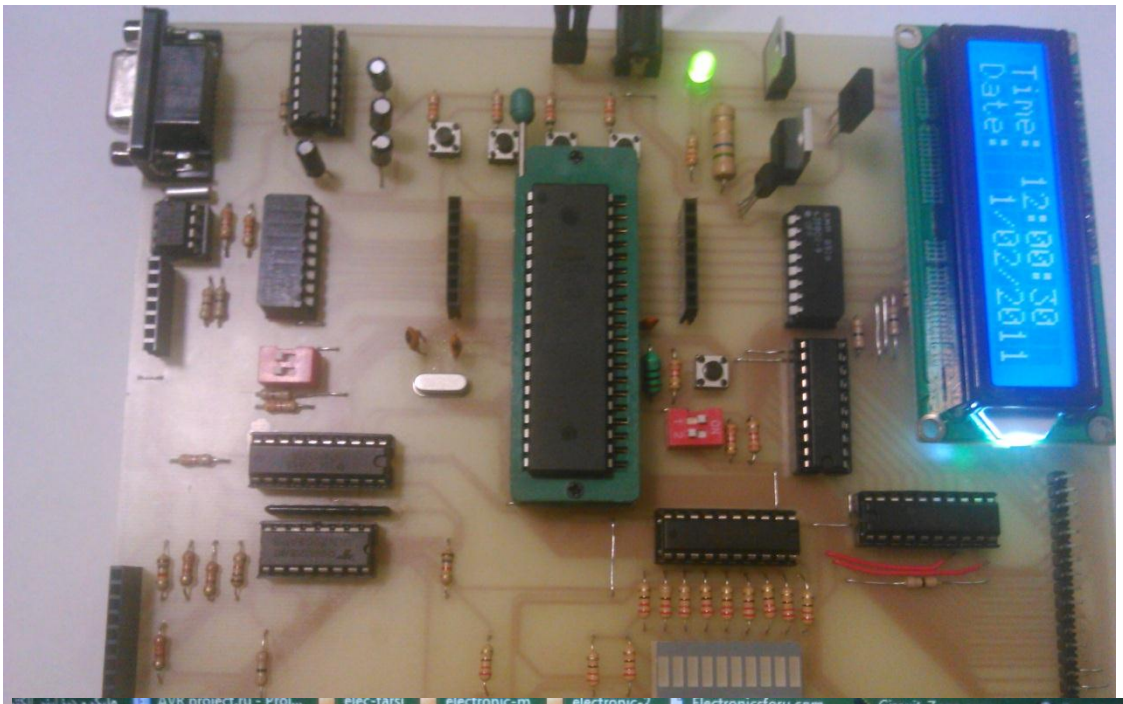
```
#include <mega32.h>
#include <delay.h>
#include <alcd.h>
#include <stdio.h>
#include <spi.h>
#include  "sd_lib.h"
char read_buff[512];
unsigned char buff2[16],sd_resp;
int nom;
void main(void){
lcd_init(16);
```

```
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();
```

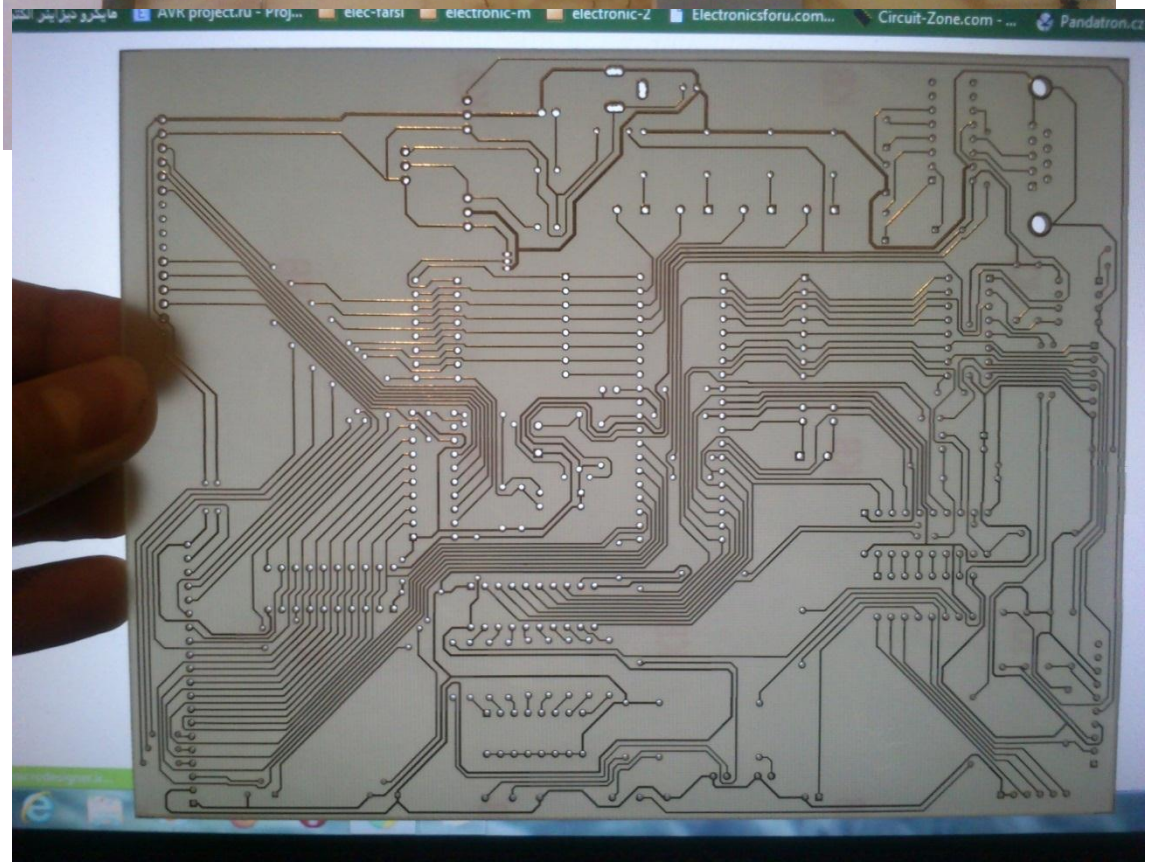
```
lcd_putsf("wrire ok");
delay_ms(300);
}




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);
}
}
}
```

۲-۸- عکس هایی از پروژه و مراحل ساخت آن :





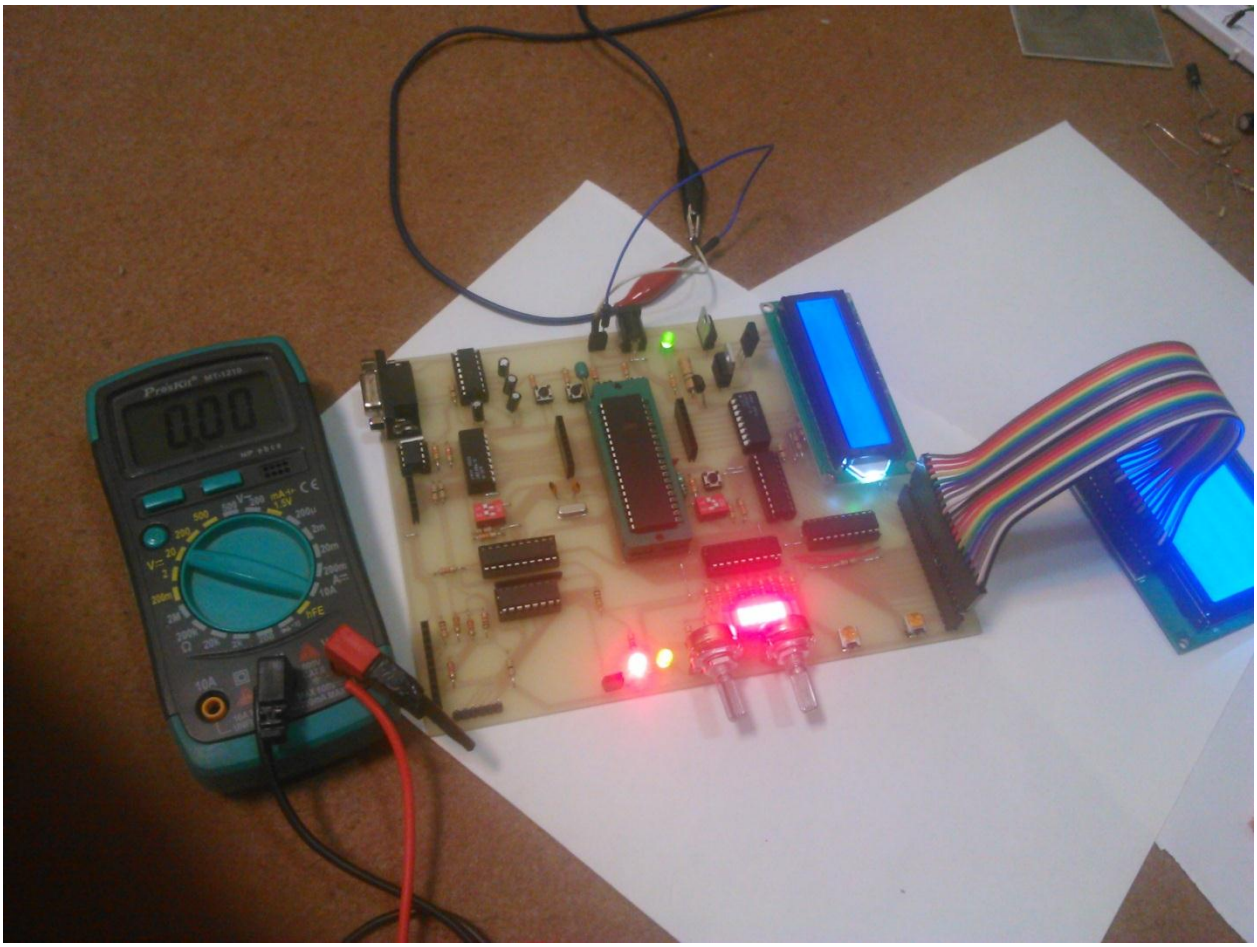
شکل ۲- ۸عکس هایی از پروژه و مراحل ساخت آن



شکل ۲- ۹عکس

هایی از پروژه و مراحل ساخت آن

شکل ۲-۱۰عکس هایی از پروژه و مراحل ساخت آن



شکل ۲-۱۱عکس هایی از پروژه و مراحل ساخت آن

فصل سوم : ضمیمه یک میکرو کنترلرATMEGA32

۳-۱- اطلاعاتی در مورد میکروکنترلرهای AVRمیکرو کنترلر چیست؟

مقدمه کلیه مدارات الکترونیکی نیاز به منبع تغذیه دارند. برای مدارات با کاربرد کم قدرت از باطری یا سلولهای خورشیدی استفاده میکروکنترلر 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 دست تان 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.

کلمهمیکروکنترلر:این کلمهازدوکلمه۱- میکرو -۲ کنترلر تشکیل شده*میکرو : میدونین که این یک واحد یونانی است و برابر با ۱۰ به توان منفی ٦ 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 مفهومش و میدونین. یعنی کنترل کننده به تعبیری یعنی "مغز " البته بدون تفکر فقطدستوراتیکه به اون داده میشه به نحواحسنانجاممیده.

حالاچرا 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 lcdیا 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.ساختماناداخلیامیکروکنترلر: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 قابلملاحضه 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 ۱۰ به توان منفی ۹ 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 ٤ دهه پیش در سال ۱۹۷۱ میلادی شرکت اینتل اولین میکروکنترلر را ساخت و اولین میکروکنترلر را با نام ۸۰۸۰ در اوایل سال ۱۹۸۰ 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 .

۳-۲- بخشهای مختلف میکروکنترلر:

میکروکنترلر ها از بخشهای زیر تشکیل شده اند:

واحد پردازشCpu

واحد محاسباتAlu

ورودی ها و خروجی هاI/O

حافظه اصلی میکروRam

حافظه ای که برنامه روی آن ذخیره می گرددRom

برای کنترل زمان ها Timer

و. . .

۳-۳- خانواده های میکروکنترلر:

خانواده: Pic - AVR - 8051

۳-٤- یک میکروکنترلر چگونه برنامه ریزی میشود؟

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

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 صنعتیکار برد دارد.

۳-۵- 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.

۳-٦- 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 programmerتهیه 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.

۳-۷- مقایسه خانواده های مختلف میکرو وکنترلرها:

خانواده : AVR

این خانواده از میکرو کنترلرها تمامی امکانات ۸۰۵۱ را دارا می باشد و امکاناتی چون(ADCمبدل 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 خانوادهTinyوMegaاست.

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 زبانهایC 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.

میکروAVR دارای معماریی است که میتواند در تمام جهات مورد استفاده شما،عمل کند میکرو AVR معماریی دارد که برای شما کارایی ۱۶ بیتی ارائه می دهد که البته قیمتش به اندازه یک ۸ بیتی تمام می شود.

بهره های کلیدیAVR :

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 .

۳-۸- 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 ۳۲ 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 (LowPower)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.

۳-۹- خانواده های محصولات AVR :

Tiny AVR:

میکروکنترلری با اهداف کلی و با بیش از ٤ کیلو بایت حافظه فلش و ۱۲۸ بایت حافظه استاتیک و قابل برنامه ریزی است.(منظور از حافظه استاتیک SRAM و حافظه قابل برنامه ریزی EEPROM است.)

MegaAVR:

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 (BootBlock)
- دقت بسیار بالای ۸-کانال در تبدیل آنالوگ به دیجیتال ۱۰ بیتی
- USART و SPI و TWI بر طبق واسطه های سریال
- واسطه ی JTAG بر طبق IEEE 1149.1

TWI: Two Wire **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

JATGواسط فقط در میکروهای با بیش از ۱٦ کیلوبایت حافظه فلش موجود است.

مشخصاتسختافزاریATMEGA32:

شکل.ظاهری.و.پایه.ها:

ATMEGA32 در سه نوع بسته بندی PDIP با ٤٠ پایه و TQFP با ٤٤پایه و MLF با ٤٤ پایه ساخته میشود که دربازارایران.بیشترنوعPDIPموجودمیباشد .

ATMRGA32 دارای چهار پورت ۸بیتی (۱ 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 خواهدشد .ولتاژ مصرفی این آی سی از V٤.٥ تا V٥.٥ میتواند باشد .فرکانس کار هم تا MHz١٦ میتواند انتخاب شود که تا MHz٨ 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 ۹ 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 ۱۲ ,۱۳ نیز برای استفاده از کریستال خارجی تعبیه شده است .

ساختارداخلیATMGA32:

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 ۳۲۳۲) KB کیلوبایت) میرسد .در این آی سی مکانی برای ذخیره موقت اطلاعات یا همان RAM هم وجود دارد که مقدارش KB۲ 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 ۱KB است و تا ۱۰۰,۰۰۰ بار میتواند پر و خالی شود .

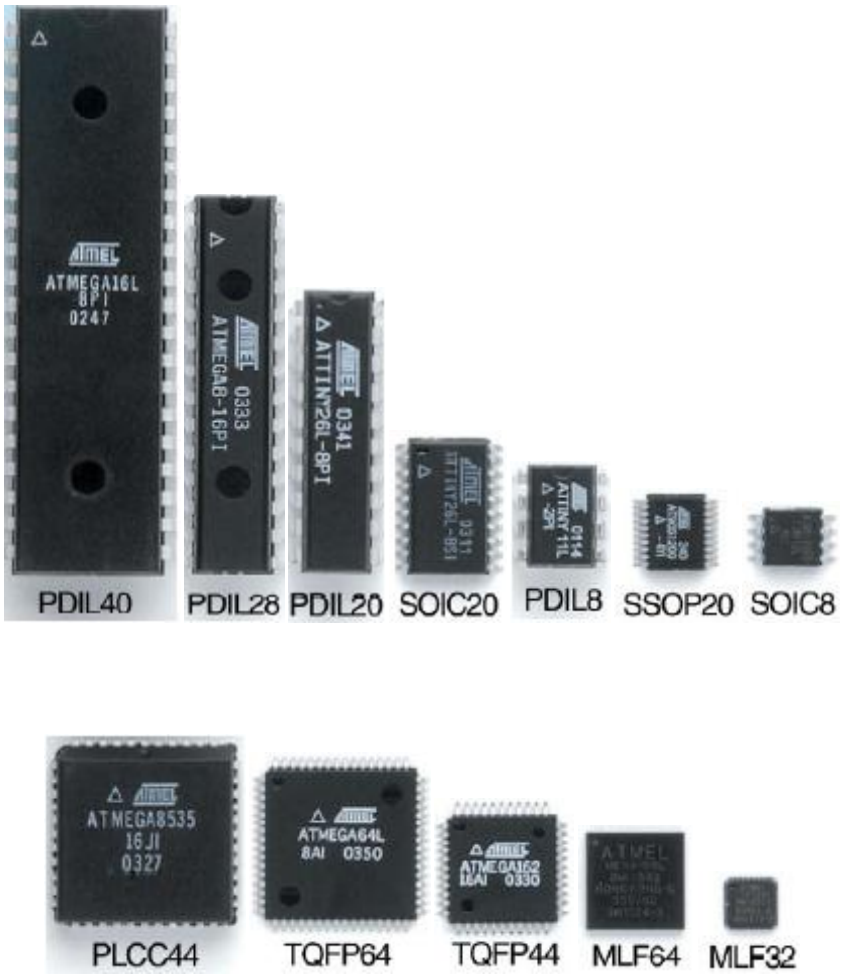
AVRهایمدلtiny:

به خود اجازه ندهید که نام آن شما را گول بزند... میکروهای مدل tinyتوانایی های عظیمی دارند.به خاطر کوچک بودن و داشتن MCUبسیار 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 EEPROMقابلیتهای خود را ثابت می کنند.

نکات کلیدی و سودمند مدلTiny :

- 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.

۳-۱۰- ابعاد مختلف میکروهای AVR را در اشکال زیر مشاهده می کنید:



شکل ۳-۱ اشکالمختلف میکروهایAVR

۳-۱۱- نکات کلیدی و سودمند حافظه ی فلش خود برنامه ریز:

- قابلیت دوباره برنامه ریزی کردن بدون احتیاج به اجزای خارجی

- ۱۲۸ بایت کوچک که به صورت فلش سکتور بندی شده اند
- داشتن مقدار متغیر در سایز بلوکه ی بوت (BootBlock)
- 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

واسطهJTAG**:**

واسطه ای که تسلیم قانون IEEE1149.1 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.

فصل چهارم :ضمیمه دومLCD کارکتری

شکل۴-۱LCDکاراکتری

۴-۱- معرفیLCDکارکتری:

LCDهای کارکتری خود به چند نوع دیگر از لحاظ اندازه تقسیم بندی میشوند . که از LCDهایی با ۱

سطر و ۱ ستون آغاز میشوند تا اندازهایی مثل ٤ سطر و ٤٠ ستون که البته تمام آنها از ١٦پایه تشکیل شدهاند.برای راه اندازی LCDتوسط AVRنیازی به دانستن جزئیات طرز کار LCDنیست . برای کاربا LCDعلاوه بر پایه های تغذیه و CONTRAST(تنظیم روشنایی) که باید مانند شکل مداری پایین بایاس شوند نیاز به ٦ پایه ی دیگر است که عبارتند از پایه های:DB7 , DB6 , DB5 , DB4 , E , RS.

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.

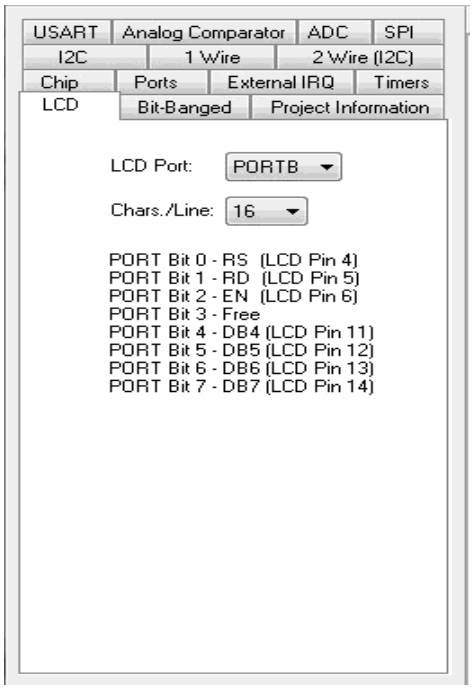
در ربات مین‌یاب برای اعلام مختصات مین‌ها به داور، باید روبات مجهز به نمایشگری باشد که بتوان این اطلاعات را بر روی آن به نمایش درآورد.در ربات فوتبالیست، نمایشگر در زمان مسابقه کاربرد مستقیمی ندارد، اما در مراحل عیب‌یابی و تنظیمات اولیه سنسورها کاربرد زیادی دارد. مثلاً برای تنظیم حساسیت هر سنسور، اطلاعات 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 توسطADCاندازه‌گیری کرد و بر رویLCDنمایش 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 نوعLCDرا بر اساس تعداد کاراکترهایی که در هر سطر و ستون نمایش داده می‌شوند، تعیین می‌کنند. پرکاربردترین سایزLCDهای کاراکتری ١٦*٢ است، یعنی این LCDمی‌تواند ٢ ردیف ١٦ تایی کاراکتر را هم‌زمان روی صفحه نمایش دهد.

چگونه.ازlcdاستفاده.کنیم؟

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

تنظیمات اولیه در codevision برای راه اندازی lcd :

Codevisionرا باز کرده پروژه‌ی جدیدی بسازید. سپس در CodeWizardتنظیمات مربوط به لبه‌یChipراانجامدهید.حالا سراغ لبه‌یLCDمی‌رویم.برای راه‌اندازیLCDهایکارکتری، باید تمام پایه‌های یکی از پورت‌های میکروکنترلر را به پایه‌های مربوطه درLCDمتصلکنیم.ابتدا باید تعیین کنیم می‌خواهیم کدام پورت را بهLCDاختصاص دهیم.سپس باید با تعیین تعداد کاراکترهای قابل نمایش در هر سطر ازLCDنوع آن را مشخص کنیم .مثلاً اگرLCDما ۲*۱۶ است، باید عدد ۱۶ را انتخاب کنیم. سپس نحوه‌ی اتصال پایه‌های میکروکنترلر بهLCDرا با توجه به نوعLCDبه شما نشان می‌دهد.



شکل۴-۲نحوه‌ی اتصال پایه‌های میکروکنترلر بهLCD

برای مثال ترتیب اتصال پایه‌ها برای 2*16 LCDبر روی پورت "B"در زیر نشان داده شده است.

پایه PB.0به پایه‌ی چهارمLCDمتصل شود.

پایه PB.1به پایه‌ی پنجمLCDمتصل شود.

پایهPB.2به پایه‌ی ششمLCDمتصل شود.

پایهPB.3به جایی متصل نمی‌شود.

پایهPB.4به پایه‌ی یازدهمLCDمتصل شود.

پایهPB.5به پایه‌ی دوازدهمLCDمتصل شود.

پایهPB.6به پایه‌ی سیزدهمLCDمتصل شود.

پایهPB.7به پایه‌ی چهاردهمLCDمتصل شود.

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

۴ 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’);

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

2- lcd_putsf(“ “);

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(,);

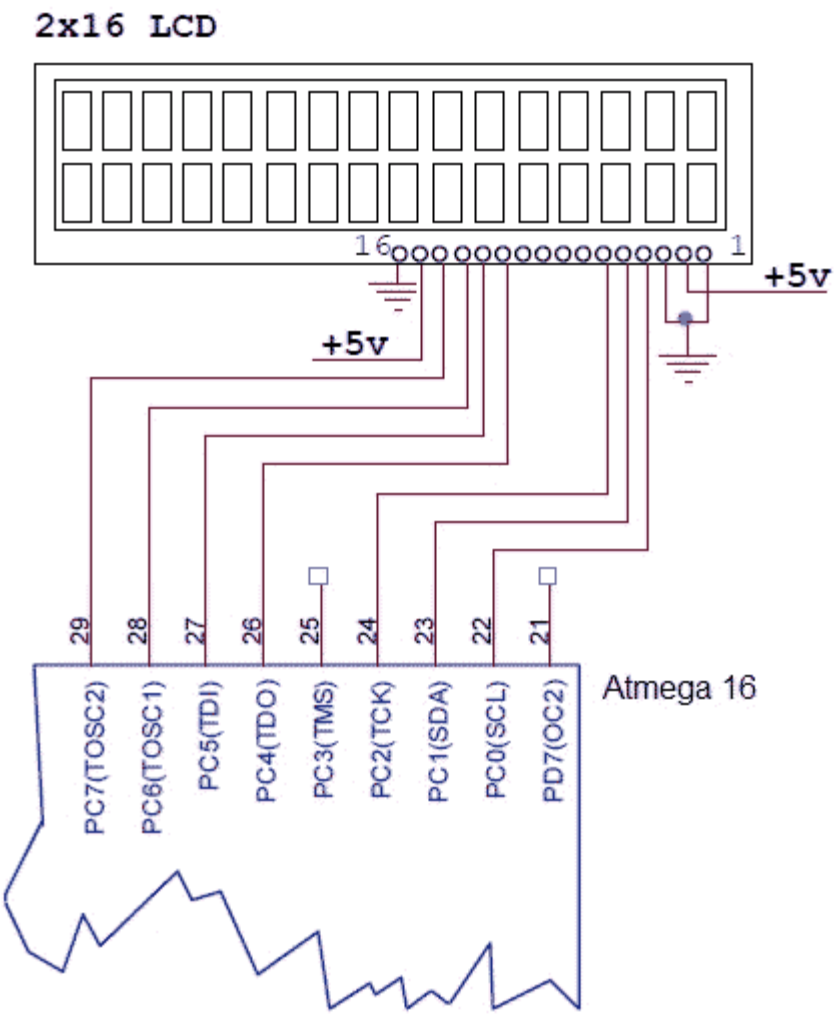
به‌کمک این دستور می‌توان تعیین کرد کاراکتر یا جمله‌ی مورد نظر ما در کدام سطر و ستون در LCD نوشته شود . مثلاً دستورهای زیر lcd را پاک کرده و واژه‌ی Hello را از وسط سطر دوم می‌نویسد .شماره‌گذاری سطرها و ستون‌ها از ۰ شروع می‌شود. پس سطر شماره‌ی ۱ ، سطر دوم است.

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

سایر پایه های lcd :

LCDنیز مانند هر قطعه‌ی الکترونیکی دیگر نیاز به ۲ پایه برای تغذیه + و – دارد. درLCDهای ۲*۱۶ اختلاف پتانسیل مورد نیاز برای تغذیه باید ۵ ولت باشد. پایه شماره‌ی ۱ باید بهGNDو پایه‌ی شماره‌ی ۲ باید به ۵ ولت متصل شود. پایه‌ی شماره‌ی ۳ نیز برای تنظیم نور زمینه درLCDتعبیه شده است.

در حالت معمولی باید این پایه مستقیماً بهGNDمتصل شود.پایه‌های ۱۵ و ۱۶ نیز برای تغذیه‌ی نور پشت زمینه هستند. پایه‌ی ۱۵ به5) Vccولت) و پایه‌ی ۱۶ بهGNDمتصل می‌شود.



شکل۴-۳نحوه‌ی اتصال پایه‌های میکروکنترلر به LCD

۴-۲- معرفی خطهای lcd :

شماره و نام خط	عملکرد
۱- Vss	زمین
۲- Vcc	ولتاژ ۵ ولت برای کنترلر
۳- Vee	ولتاژ تنظیم درخشندگی(contrast)
۴- RS	انتخابگر ثبات دستور / داده
۵- RW	انتخابگر خواندن / نوشتن
۶- Enable	فعال کننده
۷-۱۴ Bus	۸ خط گذرگاه داد یا دستور
۱۵-	ولتاژ ۵ ولت برای لامپ پشت صفحه
۱۶-	زمین برای لامپ پشت صفحه

Vee : برای تنظیم درخشندگی کاراکترها بکار می رود که باید ولتاژی بین صفر و ٥ 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 ۷ تا ١٤ خط ۷ کم ارزشترین بیت(LSB) و خط ١٤ پر ارزش ترین بیت (MSB) می باشد.

در صورت تمایل به روشن کردن لامپ پشت صفحه ولتاژ ٥ ولت را به پایه ١٥ اعمال و پایه ١٦ 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 ۷ تا ١٤ و سه خط کنترلی به پایه های ٤ تا ٦ 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 ۰D تا ۷Dقرار دهید.خط انتخاب را ابتدا یک و سپس صفر کنید. حداقل ٤٥٠ 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 ۷ تا ١٤ و سه خط کنترلی به پایه های ٤ تا ٦ 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 ٤٥٠ نانو ثانیه باید این خط را صفر نگه دارید تا داده پردازش شود. بعد از آن حالت خط تاثیری نخواهد داشت.

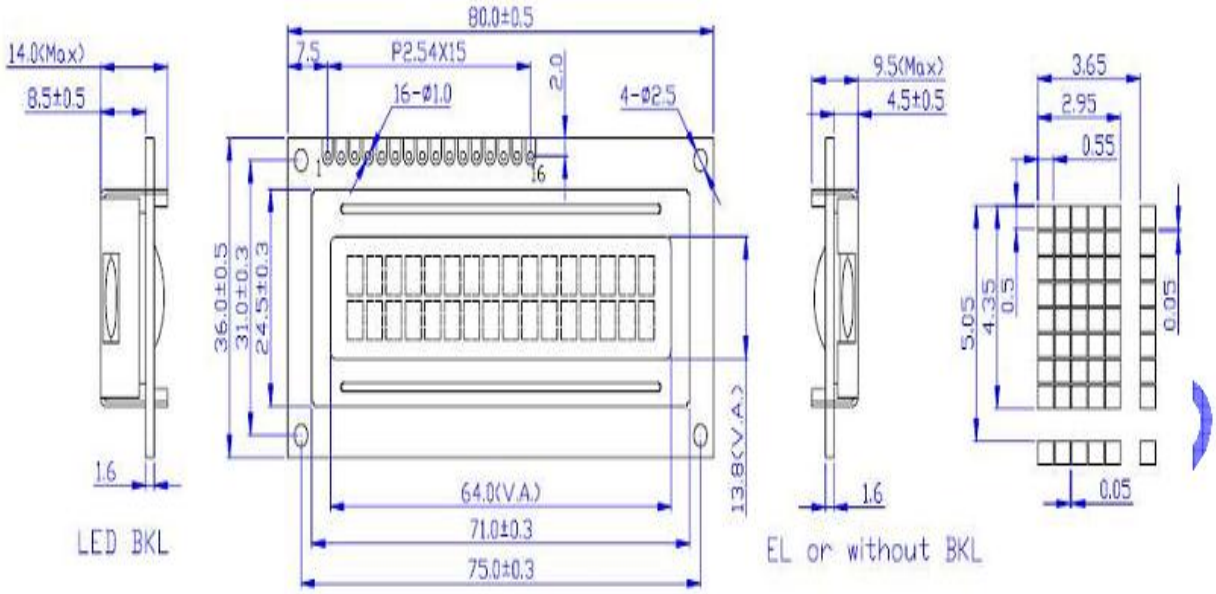
XIAMEN ELAINE E

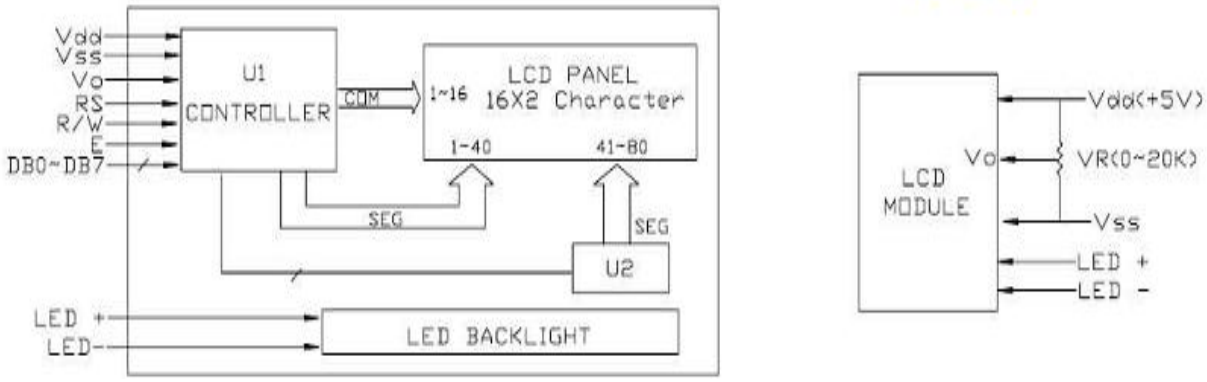
Specifications:

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)

Diagrams:

شکل۴-۴ اتصال LCD به پورت چاپگر





شکل۴-۵ اتصال LCD به پورت چاپگر

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	-

XIAMEN ELANE ELE

شکل۴-۶ اتصال LCD به پورت چاپگر

فصل پنجم : ضمیمه سوم تایمر ها و کانتر ها در میکرو کنترلر

۵-۱- تایمر/کانترصفر

تایمر /کانترصفریک تایمر8بیتی می باشدکه دارای چهار مد کاری Normal، CTC ، Fast PWM و Correct PWM Phaseمی باشد .پینT0به عنوان ورودی کانتروپینOC0خروجی بخش مقایسه ی تایمرمی باشد . این تایمردارای سه رجیستربه نامهای TCCR0 ، TCNT0 وOCR0می باشدکه بهترتیب جهت پیکربندی تایمر،مقدارشمارنده ومقدارمقایسه استفاده میشوند . همچنین این تایمر در رجیسترهای TIFRو TIMSKکه به ترتیب رجیسترهای پرچم ووقفه تایمرمی باشندبادیگرتایمرهامشترک می باشد.

مهمترین رجیستر تایمرTTCR0می باشدکه بیت هایClockSelectجهت انتخاب منبع کلاک تایمر و بیتهایWaveGenerationModeبرای تنظیم مد کاري تایمر وبیت هايCompareMatchOutputModeپیکربندیپین OC0راتعیین می کنند.

جدول۳-۱ پیکربندی پین oc0

TCCR0	۷	۶	۵	۴	۳	۲	۱	۰
نام بیت	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00

مد عملکرد تایمر به صورت زیر است :

جدول۳-۲مد عملکرد تایمر

مد	WGM01	WGM00	مد عملکرد
۰	۰	۰	Normal
۱	۰	۱	PWM, Phase Correct
۲	۱	۰	(CTC) Clear Timer on Compare Match
۳	۱	۱	Fast PWM

۱. مد Normal

رجیستر های TIMER0 :

جدول۳-۳ مد Normal

TCCR0	۷	۶	۵	۴	۳	۲	۱	۰
نامبیت	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00
سطحمنطقی	۰	۰	۰	۰	۰	X	X	X

تایمر/کانترصفرویک دارای یکPrescaleمشترک بوده وضعیت منبع کلاک با توجه به بیت های SelectClockتعیین می‌شود :

جدول۳-۴ وضعیت منبع کلاک باتوجه به بیت هایSelectClock

وضعیت منبع کلاک تایمر	CS00	CS01	CS02
بدون کلاک(متوقف)	۰	۰	۰
کلاک سیستم(بدون تقسیم)	۱	۰	۰
کلاک سیستم/8	۰	۱	۰
کلاک سیستم/۶۴	۱	۱	۰
کلاک سیستم/۲۵۶	۰	۰	۱
کلاک سیستم/۱۰۲۴	۱	۰	۱
لبه ی پایین رونده ی پالس خارجی (T0)	۰	۱	۱

لبه ی بالارونده ی پالس خارجی (T0)	۱	۱	۱

مقدارتایمردرهرلحظه ازطریق رجیستر TCNTقابل خواندن ونوشتن است:

جدول۳-۵ خواندن ونوشتن ازطریق رجیستر TCNT

Bit	۷	۶	۵	۴	۳	۲	۱	۰
TCNT0	TCNT0[7:0]							

درزمان سرریزتایمر،بیت TOV0ازرجیستر TIFRیک می شود.

جدول۳-۶ زمان سر ریز تایمر

TIFR	۷	۶	۵	۴	۳	۲	۱	۰
نام بیت	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
سطح منطقی	۰	۰	۰	۰	۰	۰	۰	X

درصورتی که بیت فعال سازعمومی وقفه فعال بوده و بیت های TOIE0 و TOIE2درجیستر TIMSKیک باشندمیتوان بااستفاده ازوقفه،ازسرریزشدن تایمر به عنوان وقفه استفاده کرد:

جدول۳-۷ سرریز شدن تایمر به عنوان وقفه

TIMSK	۷	۶	۵	۴	۳	۲	۱	۰

نام بیت	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0
سطح منطقی	۰	۰	۰	۰	۰	۰	۰	X

تایمر /کانترباعمل کردمقایسه

جدول۳-۸ تایمر /کانترباعمل کردمقایسه

Bit	۷	۶	۵	۴	۳	۲	۱	۰
OCR0/2	OCR0[7:0]							

محتوای رجیستر0OCRبه طور پیوسته با مقدار TCNT0مقایسه می شود ودرصورت برابری باعث تغییروضعیت پین0OCویا وقفه ی تطابق می شود . درحالت برابری بیتOCF0 یا OCF1 یک شده وبافراخوانی سابروتین وقفه به صورت سخت افزاری صفرمی شود.درصورت عدم استفاده ازوقفه کاربرمی تواند با نوشتن یک روی این بیت آنرا پاک کند.

جدول۳-۹ فراخوانی سابروتین وقفه

TIFR	۷	۶	۵	۴	۳	۲	۱	۰
نام بیت	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
سطح منطقی	۰	۰	۰	۰	۰	۰	X	X

تغییروضعیت پین OC0بوسیله بیت های COM00 و COM01می باشد:

جدول۳-۱۰ تغییروضعیت پین OC0بوسیله بیتهای COM00 و COM01

TCCR0	۷	۶	۵	۴	۳	۲	۱	۰
نام بیت	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00
سطح منطقی	۰	۰	X	X	۰	X	X	X

COM01	COM00	وضعیتپین OC0
۰	۰	غیر فعال (I/Oمعمولی)
۰	۱	Toggle
۱	۰	Clear
۱	۱	Set

درصورت یک کردن بیتFOC0 یا FOC1به صورت آنیمقدارریجیستر TCNT0بامقدار OCR0مقایسه شده ودرصورت تطبیق مقایسه یک تغییروضعیت رویپین OC0 ایجاد می شود . دراین وضعیت بیت OCF0 یا OCF1یک نشده وباعث ایجاد وقفه نیزنخواهدشد.درتمام حالت هایی که روی پین های OCشکل موج ایجاد می شود باید این پین به صورت خروجی تعریف شده باشد .

برای استفاده از وقفه بایدعلاوه بریک بودن فعال سازعمومی وقفه ها،بیت فعال سازمقایسه ی وقفه نیز Set شود.

جدول۳-۱۱ حالتISR

TIMSK	۷	۶	۵	۴	۳	۲	۱	۰
نام بیت	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0
سطح منطقی	X	X	۰	۰	۰	۰	X	X

دراین حالتISRبه صورت زیرتعریف می‌شود:

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

{

زیر برنامه ی سرویس وقفه

}
```

۲. مد CTC

در این مد تایمر همانندوضعیت نرمال باعمل کردمقایسه عمل می کند با این تفاوت که درزمان تطابق رجیسترهایOCR0 و TCNT0مقداررجیسترTCNT0صفرشده ودر واقع برخلاف حالت قبل مقدار ماکزیمم TCNT0عدد موجود در رجیسترOCR0می باشد . مقداربیت هایWGM00 و WGM01 در این مد به ترتیب برابر ۰ و ۱ می باشد .

جدول۳-۱۲ مقداربیت هایWGM00 و WGM01

TCCR0	۷	۶	۵	۴	۳	۲	۱	۰
نام بیت	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00
سطح منطقی	X	۰	X	X	۰	X	X	X

دراین حالت فرکانس موج ایجاد شده روی پینOC0از رابطه ی زیر بدست می آید:

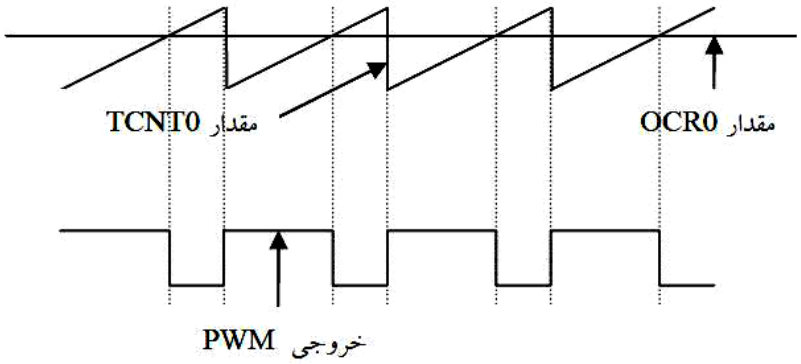
$$F_{OC0} = \frac{F_{CLK-I/O}}{2N(1+OCR0)} \qquad (۱-۳)$$

وضعیت بیت های فعال ساز وقفه ی سرریز و وقفه ی مقایسه در مد CTCهمانندوضعیت نرمال می باشد. بایک بودن بیتOCIE0وقفه ی مقایسه فعال می باشدومی توان درISR این وقفه مقدارOCR0 را تغییر داد .

مقداردهی به رجیسترOCR0بایدبادقت انجام شود زیرا در مدهای غیر PWM این رجیستر دارای بافر دوبل نمی باشد . وجود بافردوبل باعث می شودکه اگرOCR0به مقدار یک متر از TCNT0تغییر کند ، برای ازدست نرفتن مقایسه ی فعلی مقدار جدید در بافردوبل ذخیره شده و پس از سرریز این مقدار جدید درOCR0 بار گذاری شود.

۳. مد Fast PWM

این حالت مشابه مد نرمال می باشد با این تفاوت که پینOC0فقط درحالت برابری رجیسترهایTCNT0 و OCR0تغییرحالت نمی دهد،بلکه درزمان سرریز رجیستر TCNT0 نیزمقدار این پین تغییر می کند .



شکل ۵-۱ مد Fast PWM

مقداربیت هایWGM00 و WGM01 در این مد برابر ۱ می باشد .

درمدهای PWM عملکرد بیتهای COM00 و COM01متفاوت از دو وضعیت قبلی وبه صورت زیر میباشد :

جدول۳-۱۳ عملکرد بیت های COM00 و COM01متفاوتازدووضعیتقبلی

COM01	COM00	وضعیتپین OC0
۰	۰	غیرفعال (I/O معمولی)
۰	۱	رزروشده
۱	۰	Clearدروضعیت تطابق ، Setدرزمان سرریز (PWMغیرمعکوس)
۱	۱	Setدروضعیت تطابق ، Clearدرزمان سرریز (PWMمعکوس)

برای محاسبه ی فرکانس موجPWMتولیدشده می توان از فرمول زیراستفاده نمود:

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

$$F_{PWM} = \frac{F_{CLK}}{N.X} \qquad (۲-۳)$$

ازوقفه ی سرریزتایمرمی توان برای مقدار اولیه دادن بهTCNT0 و یاتغییرمقدارOCR0 استفاده نمود ، اگرچه بهتر است مقدارOCR0درروتین وقفه ی مقایسه تغییرداده شود.

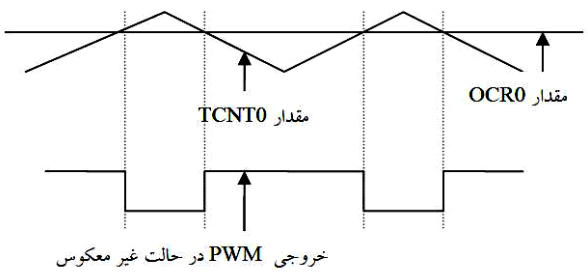
بامقداراولیه دادن بهTCNT0میتوان فرکانس موجPWM را تغییر داد.

با کمتر شدن OCR0زمان وظیفه کمترشده وتاحدیکه مقدارصفریک پالس سوزنی به عرض یک سیکل ایجادخواهدکرد.

با مقدار دهی ۲۵۶ به OCR0مقدارمقایسه وسرریزبرابرشده وپالس خروجی بسته به مقدارCOM00 و COM01همواره صفریایک خواهدبود.

۴. مد Phase Correct PWM

این مد شبیه حالت Fast PWMمیباشدبااین تفاوت که تایمربه صورت دوشیبه (Dual Slope) عمل شمارش را انجام می دهد . به این صورت که تایمرازعددصفرشروع به شمارش کرده و به صورت افزایشی تاعدد 0xFF افزایش می یابدوبعد از رسیدن به این مقدارعددموجوددر بافر OCR0 در رجیستر OCR0 بارگذاری میشود . بعدازاین لحظه جهت شمارش تایمرعوض شده وبه صورت کاهشی تاعددصفرمی شماردبارسیدن به این عددپرچم سرریز تایمر یک شده ودرصورتی ک بودن بیت فعال سازوقفه،برنامه بهISRسرریزتایمرمنشع بشده وبیت پرچم وقفه به وسیله سخت افزارصفرمیشود . مسئله ی مهم این است که درهردوحالت شمارش افزایشی وکاهشی عمل مقایسه بین رجیسترهایTCNT0 وOCR0 انجام میگیرد ودرصورت برابری پرچمOCF0یک شده وباعث تغییردرپینOC0 می شود .



شکل۵-۲ شمارش افزایشی وکاهشی عمل مقایسه بین رجیسترها

تغییر پین OC0 مطابق جدول زیرمی باشد:

جدول۳-۱۴ تغییر پین OC0

وضعیت پینOC0	COM00	COM01
غیرفعال (I/O معمولی)	۰	۰
رزروشده	۱	۰
Clearدروضعیت تطابق،درزمان شمارش افزایشی (PWMغیرمعکوس) Setدروضعیت تطابق،درزمان شمارش کاهشی	۰	۱
Setدروضعیت تطابق،درزمان شمارش افزایشی (PWMمعکوس) Clearدروضعیت تطابق،درزمان شمارش کاهشی	۱	۱

درصورت تغییرمقدار رجیستر0OCRمقدارجدید در بافر این رجیسترنوشته می شود ودرزمان رسیدن به ۰xFF رجیستر 0OCRبه روزمیشود.

پرچم سرریزتایمرصفرزمانی فعال می شودکه که رجیستر0TCNTبرابرصفرشودونه۰xFFبنابراین بایددقت داشت که اگردرزمان شروع به کارمقداراین رجیسترصفرباشدپرچم سرریزفعال خواهد شد .

فرکانس PWM در حالت Phase Correctاز رابطه ی زیر قابل محاسبه است:

$$F_{PWM} = \frac{F_{CLK}}{N.510} \quad \text{N = 1, 8, 64, 256, 1024} \qquad (3\text{-}3)$$

رابطه ی بالا نشان می دهد فرکانس موج PWMثابت است و ار تباطی به رجیسترهایOCR0 وTCNT0 ندارد.

در حالت PWMغیرمعکوس با افزایش مقدارOCR0مقدارمتوسط موجPWMافزایش یافته وباکاهش آن مقدارمتوسط کاهش می یابدودرحالتPWMمعکوس،عکس این قضیه صحیح است.

عملکردتایمردو

به طورکلی عملکردتایمر 2 مشابه تایمرصفرمی باشد و رجیسترهای مربوطه با همان نام و دارای پسوند 2 می باشند،بااین تفاوت که تایمر 2 برخلاف تایمرهای صفرویک نمیتواند ازپایه خارجیT0 یا T1کلاک دریافت کندودرعوض میتوان باوصل کردن یک کریستال32.768کیلوهرتزبه پین هایTOSC1 وTOSC2 از آن دروضعیت آسنکرون جهت RTC استفاده نمود . ازآنجایی که تایمر 2 دارایPrescalerمجزا ازدوتایمر 0 و۱ میباشدباتقسیم کریستال32768هرتز بر ۱۲۸ می توان به زمان سر ریز 1 ثانیه که مناسب برای عملکردساعت است دست پیدا کرد .

تنظیمات Prescale برای این تایمر به صورت زیر می باشد.

جدول۳-۱۵ تنظیمات Prescale برای تایمر

وضعیت منبع کلاک تایمر	CS00	CS01	CS02
بدون کلاک(متوقف)	۰	۰	۰
کلاک سیستم(بدون تقسیم)	۱	۰	۰
کلاک سیستم/8	۰	۱	۰
کلاک سیستم/۳۲	۱	۱	۰
کلاک سیستم/۶۴	۰	۰	۱
کلاک سیستم/۱۲۸	۱	۰	۱
کلاک سیستم/۲۵۶	۰	۱	۱
کلاک سیستم/۱۰۲۴	۱	۱	۱

پیکربندی RTC با رجیستر وضعیت آسنکرون یا ASSRانجام میشود:

جدول۳-۱۶ پیکربندی RTC با رجیستر

Bit	۷	۶	۵	۴	۳	۲	۱	۰
ASSR					AS0	TCN2UB	OCR2UB	TCR2UB

Asynchronous Timer/Counter2 : با Setکردن این بیت منبع کلاک تایمر 2 ازکلاک سیستم به کریستال خارجی در پایه های TOSC1 وTOSC2تغییرمیکند. باتغییردادن این بیت ممکن است مقدار رجیسترهایTCNT2 ، OCR2 و TCCR2 خراب شود .

Timer/Counter2 Update Busy :برای تضمین عملکردصحیح دروضعیت آسنکرون رجیسترهای تایمر 2 برخلاف تایمر 0 و 1 به صورت بافر شده بروزمی شوند . بدین ترتیب که

وقتی روی رجیستر2TCNTمقداری نوشته شود،بیتTCN2UBیک می شودومقدار یک هدر رجیسترموقتی ذخیره شده است به2TCNTمنتقل می شود . با اتمام بروزرسانیTCNT2این بیت توسط سخت افزارصفرمیشود . صفربودنOCR2UBنشان دهنده ی آمادگیTCNT2برای پذیرفتن مقدارجدیداست.

Output Compare Register2 Update Busy :این بیت همانندTCN2UBبوده با این تفاوت که برروی رجیستر OCR2عمل می کند.

Timer/Counter Control Register2 Update Busy :این بیت همانندTCN2UBبوده بااین تفاوت که برروی رجیسترTCCR2 عمل میکند.

درحالتی که پرچم مشغول بودن یک رجیستریک می باشد،نوشتن برروی آن رجیستریاعث می شودکه مقدار بروزشده صحیح نباشدوممکن است باعث وقفه ی ناخواسته شود.

مکانیسم خواندن این سه رجیسترمتفاوت می باشد،بدین صورت که زمان خواندن TCNT2مقدارخود رجیستر خوانده شده و باخواندنOCR2 و TCCR2مقدارموجوددررجیسترموقت خوانده می شود.

تایمر 2 دروضعیت آسنکرون درحا لتPower-Saveنیزفعال بوده وپس ازسرریزشدن تایمرازوضعیت-Power Saveخارج شده ودرفعال بودن وقفه ، ISRرا اجرا نموده ومجددا وارد حالتPower-Saveمی شود.

تایمر/کانتریک

تایمر یک تایمری 16 بیتی است ودرآن علاوه برامکانات تایمرصفر،یک بخش دیگر به نام بخشCapture به آن افزوده شده است . این بخش در زمان های خاص،عددشمارش شده توسط تایمریک وزمان سپری شده راثبت کرده وازطریق آن امکان اندازه گیری های زمانی را

فراهم می آورد . تایمری ک دارای پنج مد کاری به نام هایNormal ، CTC ، Fast PWM ، Phase Correct PWM وPhase and Mode Frequency Correct می باشد . مدهای

PWMدرتایمر 1 بسیارمتنوع ودارای 12 حالتPWM می باشد . دراین تایمرپینT1به عنوان ورودی کانتروپین هایOC1A و OC1Bبه عنوان خروجی مقایسه گرعمل می کنند .همچنین پینICP1برای ورودی بخشCapture تایمر یک می باشد .

به علت 16 بیتی بودن تایمر،رجیسترهایTCNT1 ، OCR1A و OCR1Bشانزده بیتی می باشندکه هرکدام دارای دو بایت L وHهستند . همچنین تایمریک دارای دو واحد مقایسه ی مجزا می باشدکه مقدارموجوددر رجیسترهایOCR1A و OCR1B را با TCNT1مقایسه کرده ودرصورت برابری وضعیت پین های OC1A و OC1Bراتغییرمی دهند . همچنین رجیستر ICR1نیزکه رجیستراحدCaptureاست رجیستری ۱۶ بیتی می باشد .

رجیسترهای 8 بیتیTCCR1A و TCCR1Bکنترل تایمررا برعهده دارند:

جدول۳-۱۷ رجیستر ICR1

TCCR1A	۷	۶	۵	۴	۳	۲	۱	۰
نام بیت	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10

جدول۳-۱۷ رجیستر ICR1

TCCR1B	۷	۶	۵	۴	۳	۲	۱	۰
نام بیت	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10

مد کاری تایمر بوسیله ی بیت های جدول زیر تعیین می شود :

جدول ۳-۱۸ مد کاری تایمر بوسیله ی بیت

	WGM13	WGM12	WGM11	WGM10	مد کاری	TOP	TOV=1
۰	۰	۰	۰	۰	Normal	0xFFFF	۰
۱	۰	۰	۰	۱	PWM, Phase Correct, 8-bit	0x00FF	۰
۲	۰	۰	۱	۰	PWM, Phase Correct, 9-bit	0x01FF	۰
۳	۰	۰	۱	۱	PWM, Phase Correct,	0x03FF	۰

					10-bit		
۴	۰	۱	۰	۰	CTC	OCR1A	0xFFFF
۵	۰	۱	۰	۱	Fast PWM, 8-bit	0x00FF	TOP
۶	۰	۱	۱	۰	Fast PWM, 9-bit	0x01FF	TOP
۷	۰	۱	۱	۱	Fast PWM, 10-bit	0x03FF	TOP
۸	۱	۰	۰	۰	PWM, Phase and Frequency Correct	ICR1	۰
۹	۱	۰	۰	۱	PWM, Phase and Frequency Correct	OCR1A	۰
۱۰	۱	۰	۱	۰	PWM, Phase Correct	ICR1	۰
۱۱	۱	۰	۱	۱	PWM, Phase Correct	OCR1A	۰
۱۲	۱	۱	۰	۰	CTC	ICR1	0xFFFF
۱۳	۱	۱	۰	۱	رزروشده	-	-
۱۴	۱	۱	۱	۰	Fast PWM	ICR1	TOP
۱۵	۱	۱	۱	۱	Fast PWM	OCR1A	TOP

تعریف TOP : تایمر وقتی به مقدار TOP می رسدکه برابر با بالا ترین مقدار در رشته ی شمارش خود است .این مقدارمی تواندمقادیرثابتی مثل 0x03FFبوده ویامقدارنگهداریشده دریکی از رجیسترهایOCR1A یا ICR1 باشد .FOC1A و FOC1B: بیت های Forceبخش مقایسه گرکه عملکردآنها همانندFOC1درتایمرصفر و دومی باشد .به این ترتیب که در مد های غیر PWM یک کردن این بیت بدون اینکه وقفه ای ایجادکند در صورت تطبیق

مقایسه،باعث تغییروضعیت پین هایOC1A وOC1B مطابق باوضعیت بیت هایCOM در TCCR1 می شود .بیت های COM1A0 ، COM1A1 و COM1B0 ،COM1B1 :تغییروضعیت پین هایOC1A وOC1B رادرحالت تطبیق معین می کنندکه مقدارآنهابسته به مد کاری عملکردمتفاوتی را ایجادمی کند .بیت های زیر برای تعیین منبع کلاک می باشند :

جدول ۳-۱۹ تعیین منبع کلاک

وضعیت منبع کلاک تایمر	CS00	CS01	CS02
بدون کلاک(متوقف)	۰	۰	۰
کلاک سیستم(بدون تقسیم)	۱	۰	۰
8/کلاک سیستم	۰	۱	۰
۶۴/کلاک سیستم	۱	۱	۰
۲۵۶/کلاک سیستم	۰	۰	۱
۱۰۲۴/کلاک سیستم	۱	۰	۱
لبه پیایین رونده ی پالس خارجی (T1)	۰	۱	۱
لبه ی بالارونده ی پالس خارجی (T1)	۱	۱	۱

بیت ICES1:بیت تعیین لبه ی ورودی بخشCapture از پین ICP1 . باصفربودن این بیت لبه ی پایین رونده وبایک بودن آن لبه ی بالارونده باعث تریگرمی شود.

بیت ICNC1: بیت فعال سازحذف نویز درورود ی پین ICP1

نتایج حاصل ازکارکردتایمر 1 در 4 بیت از رجیستر TIFR به نام های TOV1 (پرچم سر ریز)

OCF1A (پرچم تطابق مقایسه گر A) OCF1B (پرچم تطابق مقایسه گر B) و ICF1 (پرچم بخش Capture تایمر یک) منعکس می شوند :

جدول۳-۲۰ بیت ICES1

TIFR	۷	۶	۵	۴	۳	۲	۱	۰
نام بیت	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
سطح منطقی	۰	۰	X	X	X	X	۰	۰

یک شدن هریک ازاین پرچم هادرصورت فعال بودن بیت فعال سازعمومی وفعال بودن وقفه ی مربوطه در رجیستر TIMSK میتواندباعث انشعاب برنامه به ISR مربوط به آن وقفه شود:

جدول۳-۲۱ اجرا شدن ISR به صورت خودکار

TIMSK	۷	۶	۵	۴	۳	۲	۱	۰
نام بیت	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0
سطح منطقی	۰	۰	X	X	X	X	۰	۰

با اجرا شدن ISR به صورت خودکار بیت پرچم وقفه صفرشده ویامی تواند با نوشتن یک روی آن به وسیله ی نرم افزارآن را پاک کرد.

۱. مد Normal

این مد همانند مشابه آن درتایمرصفرمی باشد با این تفاوت که تایمرتاعدد0xFFFFشمارش کرده وبا رسیدن به آن تایمرسرریزکرده وبیتTOV1یک شده ودرصورت فعال بودن وقفه می تواند

باعث اجرایISR مربوطه شود . در مدعادی هردومقایسه گرA و Bفعال بوده وهرکدام به طورمستقل عمل مقایسه راروی رجیسترTCNT1 وOCR1A و OCR1Bانجام می دهند . درصورت برابری بیتOCF1A یا OCF1Bیک شده وخروجیOC1A یا OC1Bمطابق جدول زیرتغییروضعیت داده ودرصورت فعال بودن وقفه می توانندباعث ایجادوقفه شوند.

جدول۳-۲۲مدعادی هردومقایسه گرA و B

وضعیت پینOC1A یا OC1B	COM1A0/COM1B0	COM1A1/COM1B1
غیر فعال (I/O معمولی)	۰	۰
Toggle	۱	۰
Clear	۰	۱
Set	۱	۱

درصورت استفاده ازOC1A یا OC1Bبرای تولیدشکل موج،بایداین پین ها به صورت خروجی پیکربندی شوند.

۲. مد CTC

دراین حالت مقدار رجیسترTCNT1به طور پیوسته بامقداررجیستر OCR1A یا ICR1مقایسه می شود و در صورت برابری مقداررجیسترTCNT1برابرصفرمیشود . بنابراین دراین حالت مقدارTOPتایمر را با توجه به مقدارموجود دربیت هایWGMمقدار رجیسترهایOCR1A یا ICR1تعیین می کنند.

با رسیدن تایمر به مقدارTOPخود برحسب این که مقدارماکزیمم OCR1A یا ICR1انتخاب شده باشدبهتر تیب پرچم هایOCF1A یا ICF1یک شده ودرصورت فعال بودن وقفه ازآن می توان برای تغییردادن مقدارمقایسه استفاده کرد . این عمل باید بادقت صورت گیرد زیرا رجیسترمقایسه ی تایمرها فقط در مدهای PWMدارای بافردوبل می باشند . دراین حالت فرکانس موج ایجادشده روی پایه هایOC1A یا OC1Bمطابق رابطه ی زیرمی باشد:

$$F_{OC1X} = \frac{F_{CLK-I/O}}{2N(1+OCR1A)} \qquad (۳-۳)$$

۳. مد Fast PWM

برخلاف تایمرهای صفر و دو که درآن موج های PWM تولیدشده دارای دقت ثابت 8 بیتی هستند،تایمر ۱ قادر است سیگنال های PWM ای بادقت متغیر را ارائه کند،این مسئله باعث می شود که کاربر بتواندعلاوه برتغییر Duty Cycle فرکانس موج را به صورت سخت افزاری کنترل کند .

PWM سریع دارای پنج مد سریع می باشد: (۵ ، ۶ ، ۷ ، ۱۴ و۱۵ = [3:0]WGM1)

جدول ۳-۲۳PWM سریع دارای پنج مد سریع

	PWM سریع
۱	۸ بیتی (TOP = 0Xff)
۲	۹ بیتی (TOP = 0x1FF)
۳	۱۰ بیتی (TOP = 0x03FF)
۴	با TOP = ICR1
۵	با TOP =OCR1A

در این مد تایمر از صفر تا مقدار TOPخودشروع به شمارش کرده وپس از از سرریزمجددا ازصفرشروع به کارمیکند .

درصورتی که مقایسه ی خروجی درحالت PWMغیرمعکوس باشد درحالت تطبیق مقایسه بین رجیسترهایTCNT1 و OCR1x پین OC1xیک شده وبا رسیدن به مقدارTOPپاک می شود .درصورتی که خروجیPWMمعکوس باشد وضعیت معکوس وجود خواهدداشت . دقت موجPWMخروجی میتواند مقادیرثابت ۸ ، ۹ یا ۱۰ بیتی داشته و یا به وسیله ی رجیسترهای ICR1 یا OCR1Aبه مقداردلخواه تنظیم شود .دراین حالت حداقل مقدارمجاز 2 بیت(بادادن مقدار0x0003 به رجیسترهای ICR1 یا OCR1x) و حداکثرآن 16 بیت می باشد .

دقت موج PWMبرحسب مقدارماکزیمم ازرابطه ی زیر به دست می آید:

$$\mathrm{Re}\,solution = \frac{\log(TOP+1)}{\log(2)} \qquad (۴-۳)$$

بارسیدن تایمر به مقدار TOPپرچم سرریزTOV1فعال شده وبا تطبیق مقایسه نیز بیتOCF1A یا OCF1Bیک می شود . دراین حالت ها اگر وقفه ی مربوطه فعال شده باشدمی توان درISRآن وقفه مقدارمقایسه راتغییرداد . بایدتوجه داشت که مقدار رجیسترهای مقایسه بایدازمقدارTOPکمترباشددرغیراین صورت هیچ گاه مقایسه ای صورت نمی گیرد.

تغییروضعیت پین هادرحالت تطبیق مقایسه وسرریزمطابق جدول زیرخواهدبود:

جدول۳-۲۴تغییروضعیت پین هادرحالت تطبیق مقایسه وسرریز

وضعیت پینOC1A یا OC1B	COM1A0/COM1B0	COM1A1/COM1B1
غیر فعال (I/O معمولی)	۰	۰
اگر WGM1[3:0]=15 باشد ؛ Toggle پین OC1AدروضعیتتطابقOC1B پین I/O معمولی برایدیگرحالتهایWGM1[3:0]:غیرفعال (I/O معمولی)	۱	۰
ClearدروضعیتتطابقSet در وضعیت TOP (PWM غیر معکوس)	۰	۱
Setدروضعیتتطابق وClear در وضعیت TOP (PWM معکوس)	۱	۱

فرکانس موج PWM حاصل از رابطه زیر بدست می آید :

$$F_{PWM} = \frac{F_{CLK_I/O}}{N(1+TOP)} \qquad (۵-۳)$$

۴. مد PWM Phase Correct

PWM تصحیح فازدارای پنج مد کاری می باشد : (۱، ۲، ۳، ۱۰ و۱۱ = WGM1[3:0])

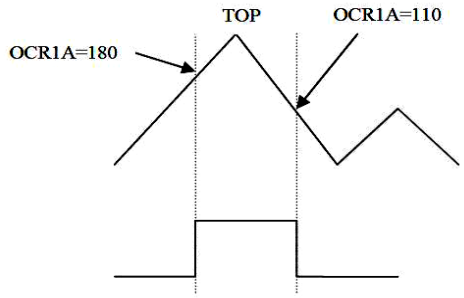
جدول ۳-۲۵PWM تصحیح فاز

	PWM تصحیح فاز
۱	۸ بیتی (TOP = 0Xff)
۲	۹ بیتی (TOP = 0x1FF)
۳	۱۰ بیتی (TOP = 0x03FF)
۴	با TOP = ICR1
۵	با TOP =OCR1A

در این مد تایمربه طورپیوسته ازمقدارصفرتاTOP وازTOPتاصفرمی شمارد . درحالتPWM غیرمعکوس درحالی که تایمربه صورت صعودی می شمارد درلحظه ی برابری رجیسترهایTCNT1 وOCR1 پین OC1xصفرشده ودرحالت شمارش نزولی باتطابق دو رجیستر این پین یک می شود . درحالتPWMمعکوس،عکس این قضیه برقراراست.دقت موج PWMخروجی می تواندمقادیرثابت ۸، ۹ یا ۱۰ بیتی داشته و یا به وسیله ی رجیسترهایICR1 یا OCR1Aبه مقداردلخواه تنظیم شود . دراین حالت حداقل

مقدارمجاز2 بیت وحداکثرآن 16 بیت می باشد .پرچم سرریز تایمر TOV1بارسیدن تایمربه مقدارصفر یک خواهدشد وباتطبیق مقایسه نیزبیتOCF1A یا OCF1Bیکمیشود .

دراینحالتهااگروقفهیمربوطهفعالشدهباشدبرنامهمیتواندبه ISR آن وقفه منشعبشود . مقدارمقایسه(OCRx) را در ISRیاهرزمان دیگرمیتوان تغییرداد اما این مقداردربافر رجیسترهای OCR1A وOCR1Bذخیره شده و با رسیدن تایمربه مقدارTOPدرخودرجیسترLoad می شودبنابراین تغییر دادن مقدار رجیسترهای OCR1xبه دلیل تغییرآن با رسیدن به TOPمی تواندباعث خروجی PWM نامتقارن شود :



مشکل بالا در PWMتصحیح فازوفرکانس بابروزکردن رجیسترهایOCR1xدرزمان رسیدن به صفر،حل می شود.تغییر وضعیت پین ها در حالت تطبیق مقایسه و سرریز به صورت جدول زیر خواهد بود : جدول۳-

۲۶ تغییر وضعیت پین ها در حالت تطبیق مقایسه و سرریز

وضعیت پینOC1A یا OC1B	COM1A0/ COM1B0	COM1A1 /COM1B 1
غیر فعال (I/O معمولی)	۰	۰
اگر WGM1[3:0]=9,14 باشد ؛ Toggle پین	۱	۰

OC1Aدروضعیت تطابق وOC1B پین I/Oمعمولی برای دیگرحالتهای[WGM1[3:0:غیرفعال (I/O معمولی)		
Clearدروضعیت تطابق وشمارش صعودی ، Set دروضعیت تطابق وشمارش نزولی	۰	۱
Set دروضعیت تطابق وشمارش صعودی ، Clear دروضعیت تطابق وشمارش نزولی	۱	۱

۵. مد Phase and Frequency Correct

همانطورکه گفته شد به دلیل بروزکردن رجیسترOCR1x با رسیدن بهTOPممکن است شکل موج خروجی نامتقارن شود.بنابراین برای حل این مشکل مد پنجم تایمر یک این رجیستررا با رسیدن به صفر بروزمی کند. تفاوت دیگر این مد وعملکرد قبلی در این است که تایمرتنها در دوحالت زیرکارمیکند:

۱. **PWM** تصحیح فازوفرکانس با **ICR1 - TOP**

۲. **PWM** تصحیح فازوفرکانس با **OCR1A - TOP**

۵-۲- واحد Capture تایمریک:

عملکرد این واحد به این صورت است که در اثر تریگرشدن ورودیCapture از پین ICP1یاخروجی مقایسه گرآنالوگ مقدارموجود در رجیسترTCNT1دررجیسترICR1نوشته شده وهمزمان پرچمCapture تایمر یک (ICF1) یک می شود . دراین زمان درصورت فعال بودن بیت پرچم ورودی(TICIE1)Capture این تریگرشدن

می تواند باعث ایجاد وقفه شودبااجراشدنISRبه طورخودکار بیتICF1صفرشده ویادرصورت فعال نبودن وقفه می تواندبانوشتن یک برروی آن پاک شود.

جدول۳-۲۷ بااجراشدنISRبه طورخودکار

Bit	۷	۶	۵	۴	۳	۲	۱	۰
ICR1H	ICR1[15:8]							
ICR1L	ICR1[7:0]							

رجیستر ICR1 به جزدرحالتی که به عنوانTOPجهت مقایسه به کارمیرود (مدهای ۸ ،۱۰، ۱۲و۱۴) یک رجیسترفقط خواندنی است.همان طورکه گفته شد تریگرشدن واحد Captureمی تواندازدومنبع مختلف صورت گیردکه این ازطریق بیتACIC در رجیستر ACSRصورت می گیرد . صفربودن این بیت پینICP1ویک بودن آن خروجی مقایسه کننده ی آنالوگ را انتخاب می کند . همچنین نوع سیگنال ورودی ازپینICP1بوسیله ب یتICES1 از رجیستر TCCR1Bتعیین می شود،به این ترتیب که صفربودن این بیت لبه ی پایین رونده و یک بودن آن لبه ی بالارونده ی سیگنال ورودی را انتخاب می کند.

جدول۳-۲۸ نوع سیگنال ورودی ازپینICP1بوسیله ب یتICES1 از رجیستر TCCR1B

TCCR1B	۷	۶	۵	۴	۳	۲	۱	۰
نامبیت	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10

ورودی captureدارای یک واحدکاهش نویز نیزمی باشدکه بااستفاده ازیک فیلتردیجیتال ایمنی ورودی رابهبود می بخشد . این واحدبایک کردن بیتICNC1 از رجیستر TCCR1Bفعال می شود . بافعال شدن این فیلتر بایدسیگنال نمونه برداری شده روی پایه یICP1برای چهارسیکل کلاک معتبرباشد.

فصل ششم:ضمیمه چهارمADC & DAC

۶-۱- ADC &DAC

:ADC

۶-۲- راه اندازی یکADC:

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:

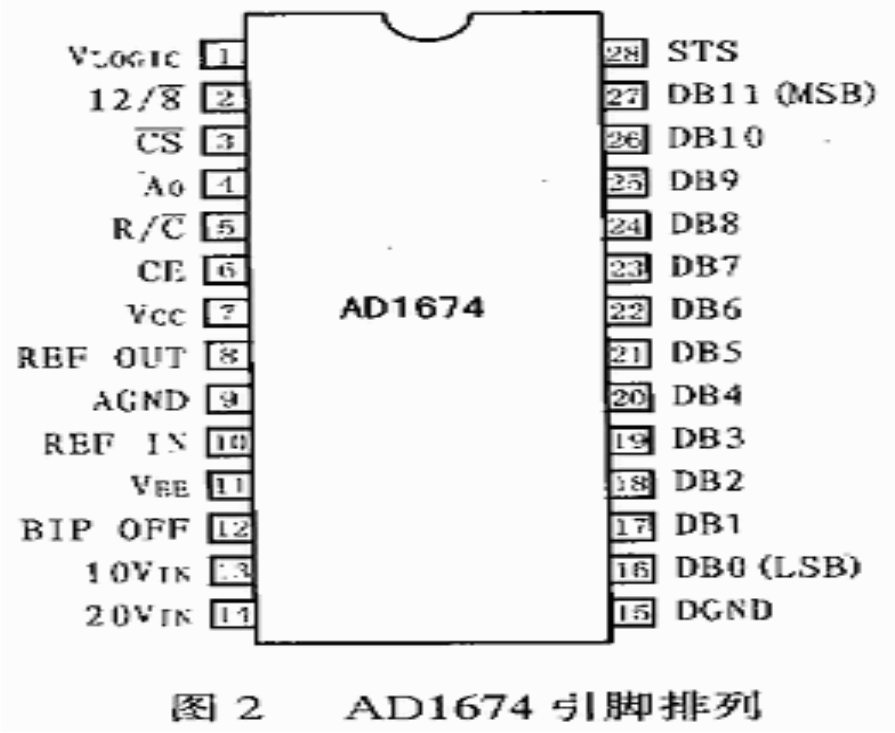


图 2　AD1674 引脚排列

شکل ٦-١ (AD1674)

این ای سی دارای ۳نوع تغذیه می باشد. پایه شماره ۱آن سطح ولتاژ منطقی می باشد که بایستی به آن ولتاژی بین ۴.۵ تا ۵.۵ ولت اعمال کرد. پایه شماره۷تغذیه مثبت مدار بوده که می تواند ۱۲یا ۱۵ولت باشد که ما به آن ۱۲ولت را اعمال می کنیم. و پایه شماره ۱۱تغذیه منفی بوده که می تواند ۱۲-یا ۱۵-ولت باشد که باز هم ما از سطح ولتاژ ۱۲-استفاده می کنیم. این ای سی همچنین ۵پایه کنترلی دارد پایه ۶فعال کننده چیپ بوده و اگر سطح منطقی ۱را داشته باشد ای سی فعال است.همین طور پایه ۳انتخاب گر ای سی بوده که اگر lowباشد چیپ انتخاب می شود. پایه ۲یا ۱۲/۸مشخص می کند که ما از۸ بیت در خروجی استفاده کنیم یا ۱۲بیت که ما این پایه را high کرده تا هر ۱۲بیت را در خروجی داشته باشیم پایه ۵(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 ۱۶ تا ۲۷ نیز خروجی های دیجیتال ما بوده که کم ارزشترین آن ها پایه ۱۶ می باشد.

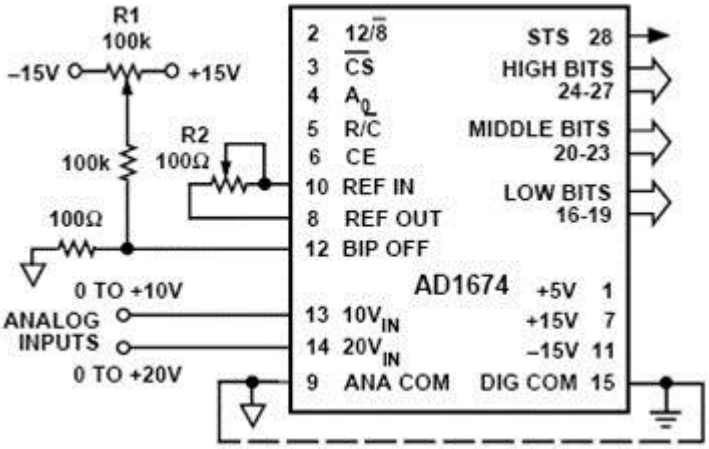
همانند D/Aاین ای سی نیز در دو حالت تک قطبی و دو قطبی می تواند کار کند. پایه REFINو REFOUT برای همین موضوع به کار می روند که مدار های آن ها در ادامه آورده شده است. در ضمن این ای سی می تواند با دو حالت ورودی بگیرد اگر از پایه ۱۳ به عنوان ورودی استفاده کنیم حداکثر تا بین ۰ تا ۱۰ ولت در حالت تک قطبی یا بین۵-تا ۵ ولت در حالت دوقطبی می توان به آن اعمال کرد. و اگر از پایه ۱۴ استفاده کنیم این محدوده ۲۰ 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:

شکل۶-۲ پالس R/C در روش low pulse

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



شکل۶-۳ شماتیک مدار راه اندازی یک ADC

مدار تک قطبی

پایه شماره ۱۲ برای کم کردن offsetبه 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 ۸ و ۱۰ به منظور تنظیم حد بالای ولتاژ بهکار میرود یعنی اینکه چه ولتاژی را به عنوان خروجی حداکثر بگیرد در حالت معمول که این حد ۱۰ولت است می توان یک مقاومت ۵۰ اهمی رای به جای پتانسیومتر قرار داد.

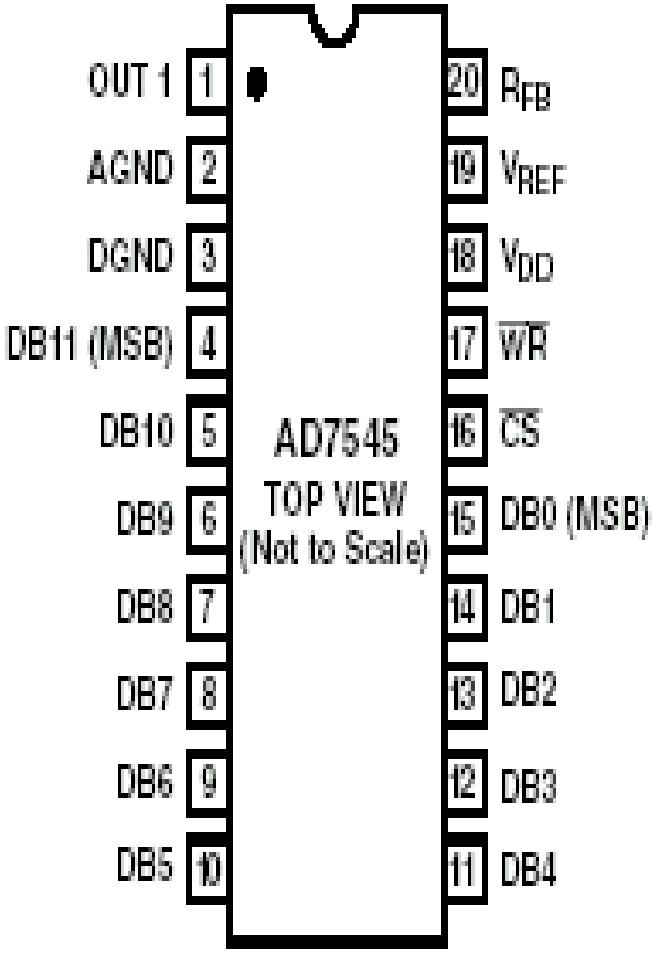
ما فعلا از مدار تک قطبی استفاده می کنیم.

برای آزمایش این ICورودی آنالوگ را به پین ۱۳ دادم(Vin۱۰) . بنابراین به ازای ۱۰ ولت ورودی آنالوگ باید xFFF۰ و به ازای ۰ ولت باید ۰۰۰×۰ بدهد.

به ورودی آنالوگ ولتاژ V۵ دادم خروجی دیجیتال "۰۱۱۱۱۰۱۰۱۱۱۰" شد. به ازای V۱۰، خروجی دیجیتال"۱۱۱۱۱۱۱۱۱۱۱۱" شد و به ازای V۰، خروجی "۰۰۰۰۰۰۰۰۰۰۰۱" شد.

کار کردن با DAC12 بیتی:

تراشه مبدل دیجیتال به آنالوگ ۱۲ بیتی که در اینجا از آن استفاده میکنیم AD7545 ساخت شرکت Analogdevices است:

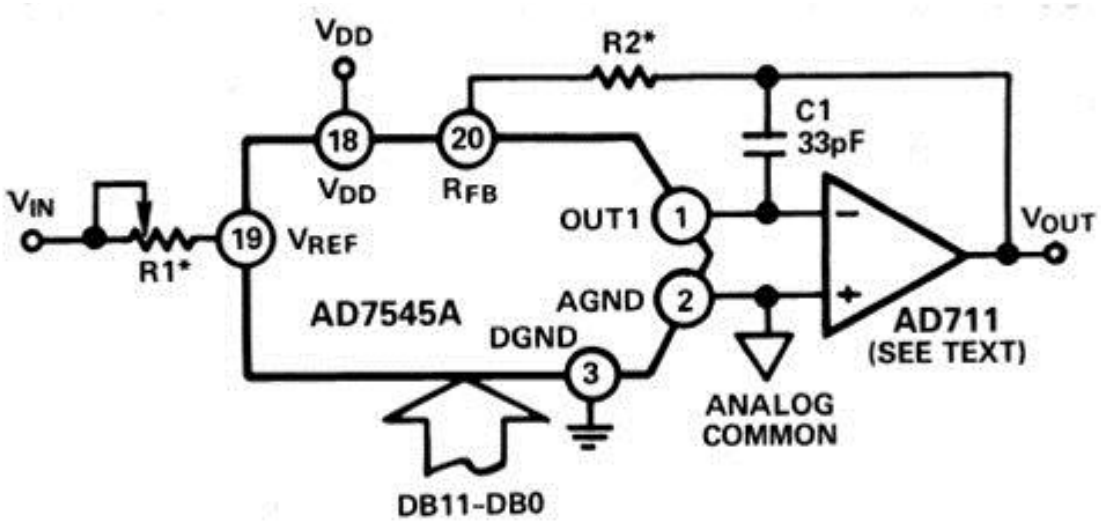


شکل۶-۴ تراشه مبدل دیجیتال به آنالوگ ۱۲ بیتی

برای مقداردهی به ورودی های دیجیتال این تراشه از میکرو کنترلر atmega۱۶استفاده 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، پایه‌های CS, WR،DGND،AGND را به زمین ، VDD را به V۵ و V_{IN} را به V۱۰ وصل میکنیم. ورودی های دیجیتال را به پورت A و ۴ بیت کم ارزش پورت B میدهیم و مدار تک قطبی (شکل زیر) را میبندیم.



شکل۶-۵مدار داخلی تراشه مبدل دیجیتال به آنالوگ

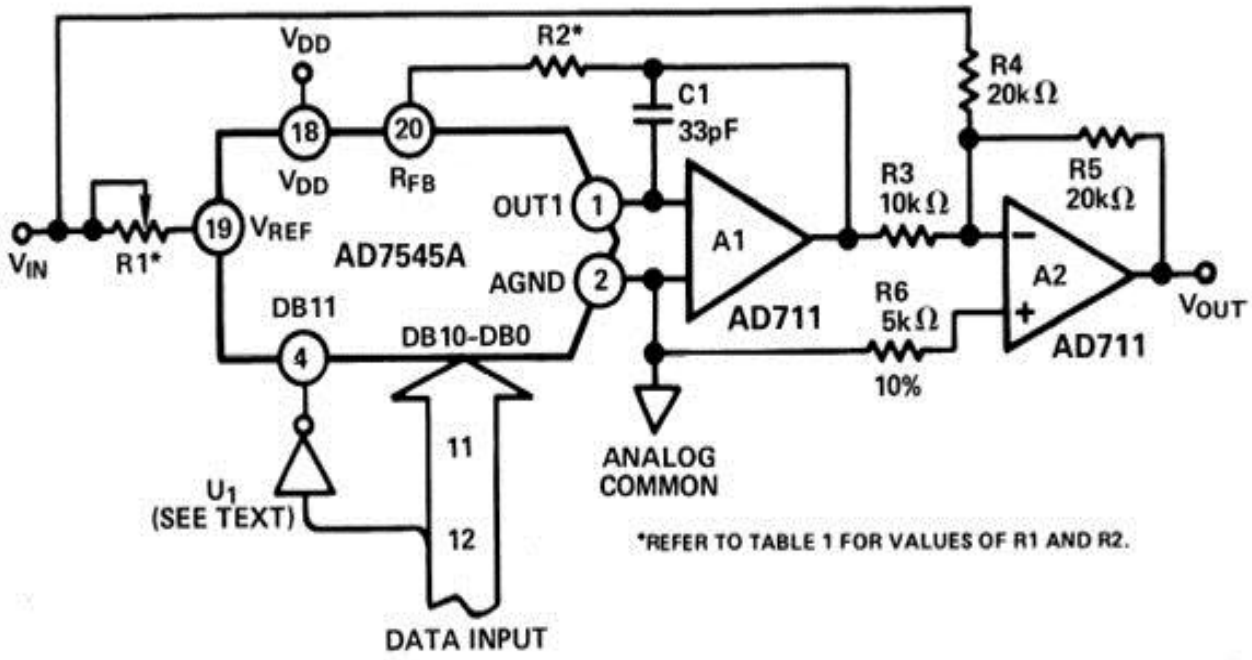
مقادیر R,۲R ،۱R دراین شکل در datasheet قطعه آمده است. با استفاده از میکروکنترلر به ورودیهای دیجیتال AD7545صفر و ۱ می‌دهیم ،خروجی آنالوگ با تقریب خوبی مطابق دیتاشیت به دست می‌آید:

0xFFF -> -8.6V

0×800 -> -5V

0×000 -> 0V

حال مدار دوقطبی(شکل زیر)را می‌بندیم:



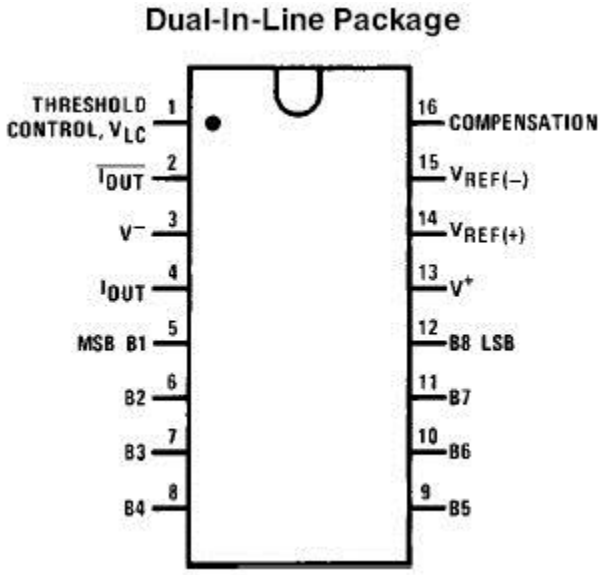
شکل۶-۶مدار دو قطبی مبدل دیجیتال به آنالوگ

و به ورودیهای دیجیتال مقدار می دهیم. میبینیم که خروجی آنالوگ تغییرمنطقی می‌کند 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 ۴ بدهیم.

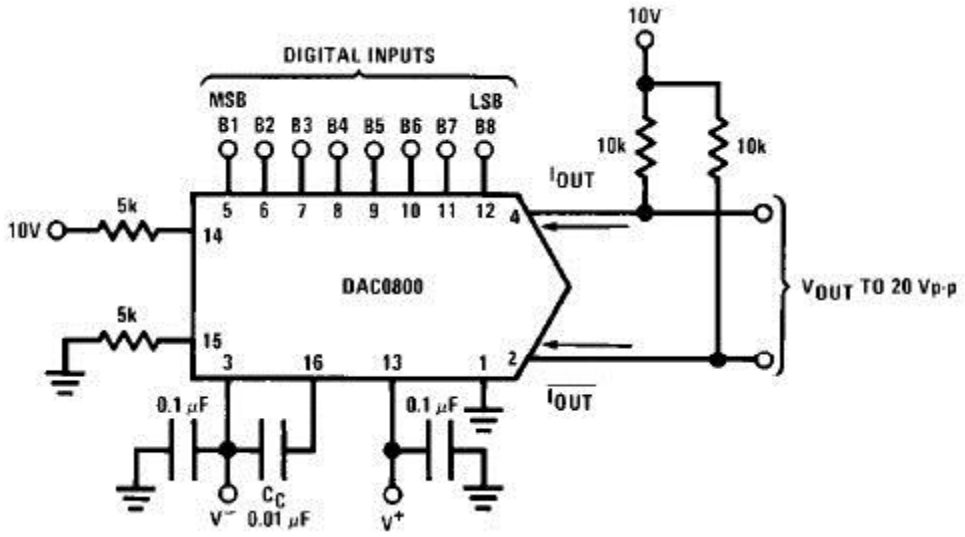
توضیح: OP-AMP مورد استفاده در اینجا TL074 است که Quad OP-AMPمی باشد

٦-٣- کار کردن باDAC۸بیتی:

تراشه DAC0800 یک مبدل دیجیتال به آنالوگ ۸ بیتی است. در شکل زیر این تراشه و همچنین مدار کاربردی سفارش شده در دیتاشیت این قطعه را می بینیم:



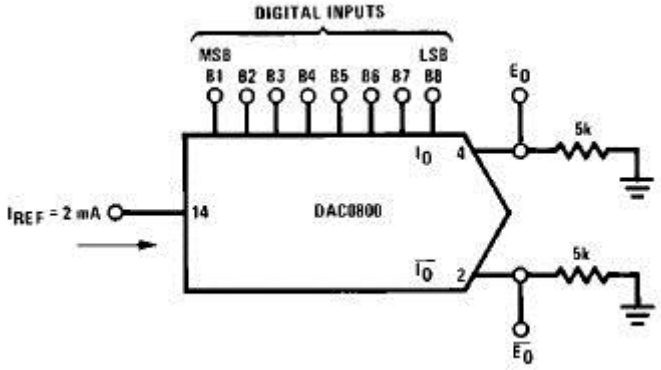
شکل ۶-۷ تراشه DAC0800 یک مبدل دیجیتال به آنالوگ



شکل۶-۸ مبدل دیجیتال به آنالوگ

پایه ۳ (V-) به ۱۰v- و پایه ۱۳ (v+)به+۱۰vوصل می‌شود و به جای مقاومت K۵ از مقاومت K۴.۷استفادهمی کنیم خروجی این IC جریان است که ما برای تبدیل آن به ولتاژ از یک OP-AMP مثلاً AD648 استفاده می کنیم.

اما قبل از بستن مدار با OP-AMPبرای اطمینان از درست کار کردن DACO800 پایه های ۴ و ۲ را با یک مقاومت K۴.۷ به زمین میبریم.



ورودیهای دیجیتال این IC را به پورت D میکرو می‌دهیم و به پورت D سه مقدار ۰۰× و ۰۸×۰ و xFF۰می‌دهیم و خروجی آنالوگ را اندازه می گیریم. نتایج اندازه گیری و داده های دیتاشیت را در جدول های زیر آورده ایم.

نتایج اندازه گیری:

جدول۴-۱نتایج اندازه گیری کار کردن با DAC۸ بیتی

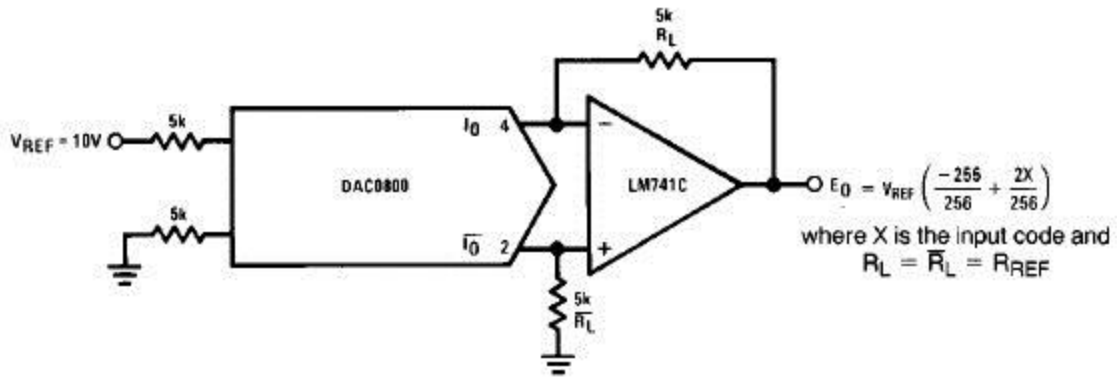
ورودی دیجیتال	جریان پین ۲	جریان پین ۴	ولتاژ پین ۲	ولتاژ پین ۴
۰۰×۰	mA۱.۵۵-	mA۰.۵-	V۷.۳-	V۲.۳۸-
۸۰×۰	mA۱.۰۴-	mA۱.۰۵-	V۴.۹-	V۴.۹۸-
xFF۰	mA۰.۵-	mA۱.۵۴-	V۲.۴۴-	V۷.۲۷-

داده های همین آزمایش در دیتاشیت:

جدول۴-۲ نتایج اندازه گیری کار کردن با DAC۸ بیتی 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 ۲ و ۴ را مطابق شکل زیر به OP-AMP متصل می کنیم.



شکل ۶-۹ امتحان صحت کار DACO800

مانند دفعه قبل به ورودیهای دیجیتال مقادیر ۰×۰۰ و ۰×۸۰ و ۰xFF می‌دهیم و خروجی آنالوگ را اندازه می‌گیریم:

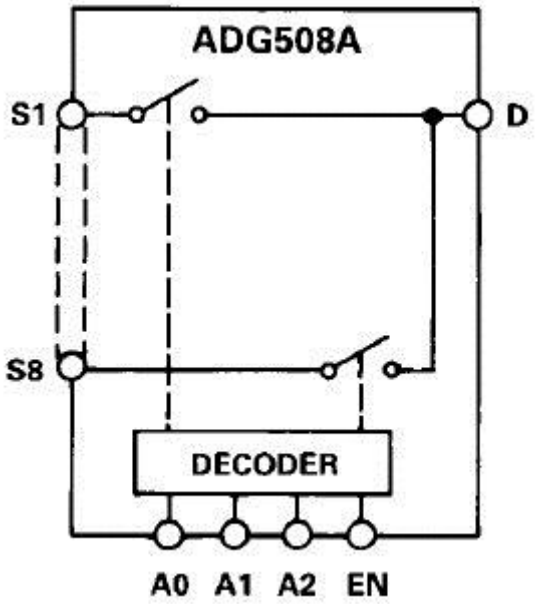
جدول۴-۳ کار کردن با DAC۸ بیتی(خروجی آنالوگورودی دیجیتال)

خروجی آنالوگ	ورودی دیجیتال
-۸.۶V	۰×۰۰
۰.۰۸V	۰×۸۰
۹.۲۸V	۰xFF

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

کار کردن با یک مالتی پلکسر:

تراشه مالتی پلکسر ما ADG508 است. یک مالتی پلکر ۸ به ۱ که دیاگرام آن به صورت زیر است.



شکل ۶-۱۰ دیاگرام یک مالتی پلکر ۸ به ۱

این ICدر دو مد Dual supply , single spply کار می کند.

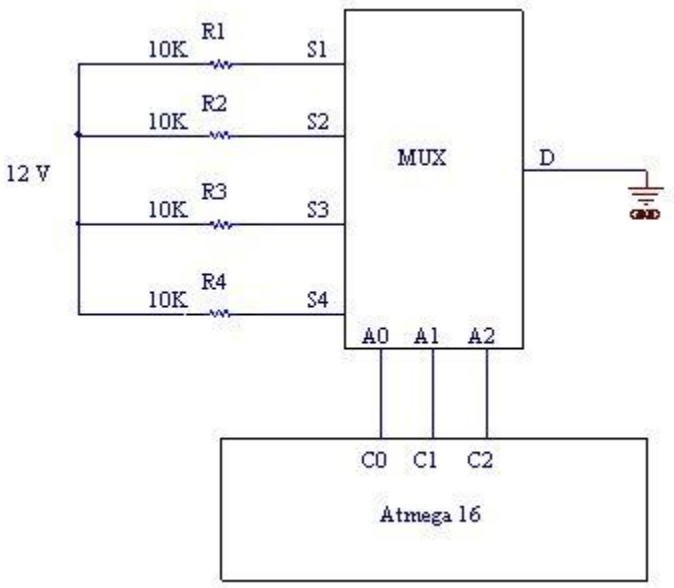
Dual supply:

v to 16.5 v۱۰.۸- =v to 16.5 v, V_{SS}۱۰.۸=V_{DD}

Single supply:

GND=OV=v to 16.5 v, V_{SS}۱۰.۸ =V_{DD}

ما دراینجا از مد Dual supply استفاده می کنیم. برای آزمایش این IC مدار زیر را می‌بندیم .



شکل۶-۱۱ آزمایش مالتی پلکسر ADG508

R_{on} این IC در حدود ۲۸۰ اهم است که در مقایسه با مقاومت K۱۰ که ما در مدار قرار دادیم قابل چشم پوشی است با استفاده از میکرو کنترلر به پایه‌های A_0, A_1,A_2 ورودی می‌دهیم .

جدول درستی در زیر آورده شده:

جدول۴-۴ جدول درستی یک مالتی پلکسر

ON Switch pair	EN	A0	A1
NONE	۰	X	X
۱	۱	۰	۰
۲	۱	۱	۰
۳	۱	۰	۱
٤	۱	۱	۱

به این ترتیب مثلاً هنگامی که EN=1 و A0=A1=0 تمام V۱۲ دوسر مقاومت R1 افتاد و بقیه مقاومت‌ها قطع بودند.

منابع

www.ATMEL.com

WWW.AVRFREAKS.NET

مجموعه کتاب های آموزش AVR

شماتیک بردهای آموزشی شرکت های تولید کننده برد های آموزشی