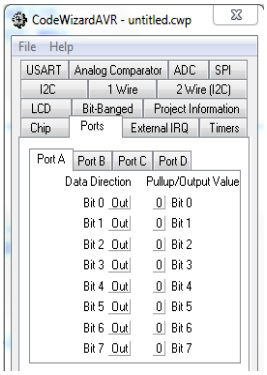
نمایشگرها

(LED و7seg و LCD)

در این بخش به ارائه نحوه کار با نمایشگرها خواهیم پرداخت که اولین گام برای آشنایی شما با میکروکنترلر خواهد بود و توسط تمرین با آنها شما به نحوه استفاده از پورت ها و پین های میکرو آشنا خواهید شد .

LED

LED ها نمایشگرهایی هستند که علیرغم سادگی شان بسیار کاربردی هستند. برای شروع می خواهیم توسط مثال زیر پین های پورت A میکرو را یکی یکی روشن کنیم برای این کار در کدویزارد پورت A را مانند زیر خروجی می کنیم :

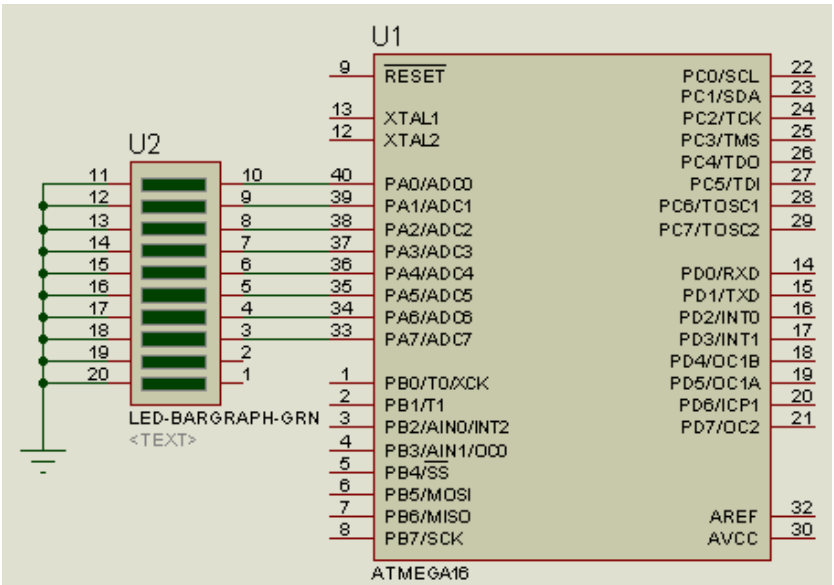


شکل4-1

البته در خلال برنامه نیز می توان با دستور DDRA=0XFF; نیز این کار را کرد.

```
#include <mega16.h>          // معرفی میکرو مورد استفاده
#include <delay.h>           // فراخوانی کتابخانه تاخیر
```

```
unsigned char a=0b00000001;                    // معرفی متغیر و نوع آن
void main(void) {                              // برنامه اصلی
while (1) {                                    // قرار دادن برنامه در حلقه بی نهایت
PORTA=a;                                       //a ریختن مقدار متغیر مورد نظر در پورت
delay_ms(1000);                                // تاخیر به مدت یک ثانیه
a<<=1; } }                                     // شیفت دادن اندازه یک بیت و پایان برنامه
```

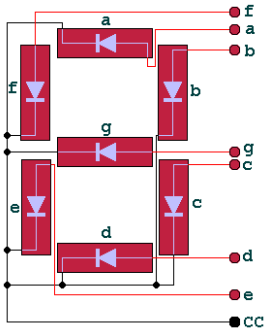


شکل2-4

7Segment

حال که با نحوه کارکرد پورت و پین میکرو آشنا شدید می خواهیم که به معرفی سون سگمنت (7seg) که یکی از پرکاربردترین نمایشگرها می باشد بپردازیم . همانطور که از نام این نمایشگر بر می آید از هفت LED یا سگمنت تشکیل شده است و به دو صورت آند

مشترک و کاتد مشترک موجود می باشند .ساختار داخلی این قطعه به صورت زیر می باشد :



شکل4-3

در این شکل بالا، قطعه ما کاتد مشترک است که اگر آند مشترک بود، آند همه سگمنت ها به هم وصل می شد و به یک پایه وصل می شد .

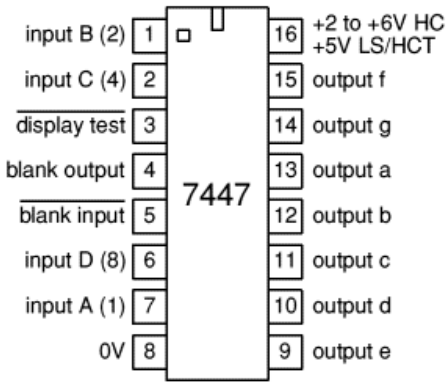
حال اگر بخواهیدکارکتری را بر روی این نمایشگر نشان دهید باید کد لازم برای نوشتن کارکتر را بدست آورید مثلا اگر بخواهید عدد چهار را روی آن نمایش دهید باید سگمنت های b ، g ، c ، f را روشن کنید که کد معادل ان به صورت 0X66= (01100110) بدست می آید .در زیر لیست کدهای اعداد از 0 تا 9 به صورت کاتد مشترک نوشته شده است .

(برای آند مشترک صفرها را به یک و یک ها را به صفر تبدیل می کنیم)

عدد	سگمنت						
	a	b	c	d	e	f	g
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	1	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1

جدول1-4

اما برای نمایش اعداد بر روی سون سگمنت راه دیگر نیز وجود دارد و آن استفاده از آی سی های 7447 (برای آند مشترک) و7448 (برای کاتد مشترک) می باشد که مدار داخلی 7447 در پایین آمده است :

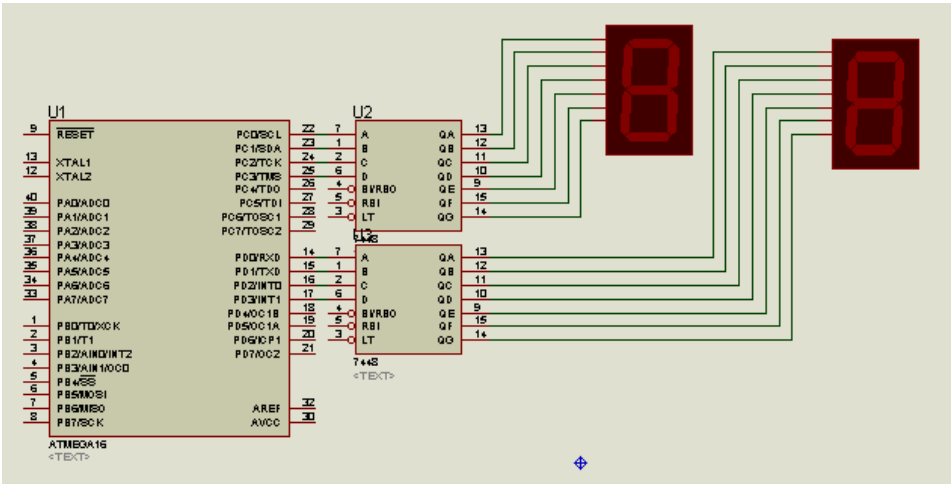


شکل4-4

حال می خواهیم یک شمارنده 0 تا 99 بر روی پرت های c و D طراحی کنیم که در آن از آی سی 7448 استفاده شده است . مانند مثال قبل در کد ویزارد پرت های C وD را خروجی می کنیم :

```
#include <mega16.h>                          // معرفی میکرو مورد استفاده
#include <delay.h>                           //فراخوانی کتابخانه تاخیر زمان
unsigned char i,b;                           // تعریف متغیر ها
void main(void)  {                           // برنامه اصلی
while (1)  {                                 //حلقه بی نهایت
    for(i=0;i<10;i++)   {                    //شمارش دهگان
    PORTC=i;                                 // نمایش دهگان
```

```
for(b=0;b<10;b++) {                                    // شمارش یکان
PORTD=b;                                               //نمایش یکان
delay_ms(1000); } } } }                                // تاخیر یک ثانیه ای و پایان برنامه
```



شکل 4-5

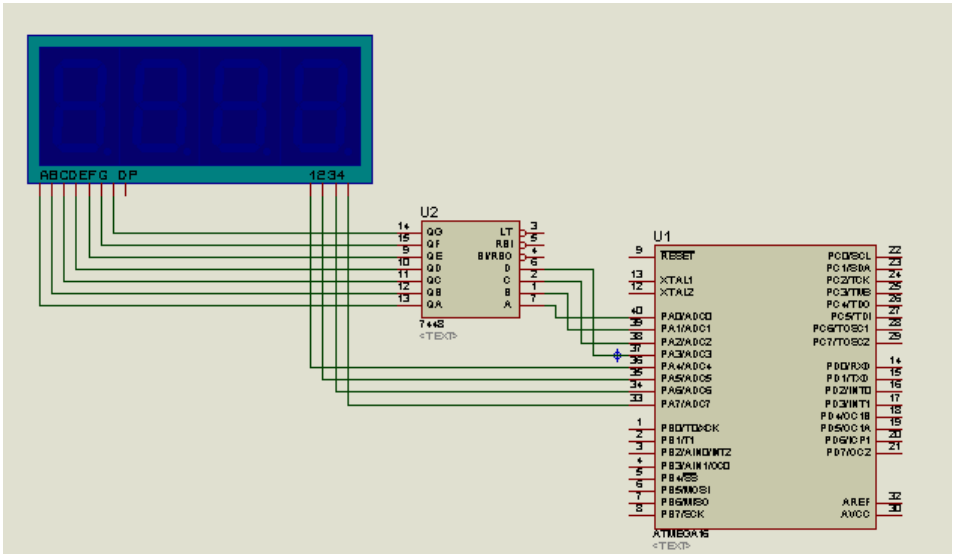
برنامه ی فوق ممکن است در مدارات پیچیده تر سبب ایجاد مشکل شود و ما با کمبود پورت و پین برای نوشتن برنامه مورد نظر مواجه شویم . برای رفع این مشکل می توان از خطای دید انسان و ماندگاری اثر نور در چشم انسان و همچنین از سرعت بالای CPU کمک گرفته و توسط یک پورت چندین سون سگمنت را روشن کرد بدین صورت که اگر ما سون سگمنت ها را با سرعت بالا یکی یکی روشن و خاموش کنیم اینطور به نظر می رسد که همه ی آنها به طور همزمان روشن هستند . فقط به این نکته توجه داشته باید که بعد از روشن کردن هر سون سگمنت ، یک بار همه آنها را خاموش کرده و سپس سون سگمنت بعدی را روشن می کنیم .

مثال :یک شمارنده 0 تا 9999 را بر روی پورت A ایجاد کنید :

```
#include <mega16.h>                                    //معرفی میکرو مورد استفاده
```

```
#include <delay.h>                                  //فراخوانی کتابخانه تاخیر زمانی
unsigned char d[4]={0,0,0,0},i;                     //تعریف متغیر ها
void main(void)  {                                  // برنامه اصلی
while (1)   {                                       //حلقه بی نهایت
   d[0]++;                                          //شمارش یکان
    if(d[0]==10)  {
    d[0]=0;
    d[1]++;                                         //شمارش دهگان
        if(d[1]==10)  {
          d[1]=0;
          d[2]++;                                   //شمارش هزارگان
          if(d[2]==10)  {
           d[2]=0;
           d[3]++;                                  //شمارش ده هزارگان
               if(d[3]==10)  {
                  d[3]=0;  } } } }
for(i=0;i<=20;i++)  {     //چندین بار نمایش برای اینکه همه سگمنت ها روشن به نظر برسند
PORTA=0x80|d[0];  //قرار دادن یکان روی چهار پین اول و یک کردن پین هفتم برای روشن کردن سگمنت اول
delay_ms(5);                                        //تاخیر جهت ماندگار شدن نور در چشم
PORTA=0xf0;                                         //خاموش کردن همه سگمنت ها
```

```
PORTA=0x40|d[1];   //قرار دادن دهگان روی چهار پین اول و یک کردن پین ششم برای روشن کردن سگمنت دوم
delay_ms(5);                                  //تاخیر جهت ماندگار شدن نور در چشم
PORTA=0xf0;                                   //خاموش کردن همه سگمنت ها
PORTA=0x20|d[2];   //قرار دادن صدگان روی چهار پین اول و یک کردن پین پنجم برای روشن کردن سگمنت سه
delay_ms(5);                                  //تاخیر جهت ماندگار شدن نور در چشم
PORTA=0xf0;                                   //خاموش کردن همه سگمنت ها
PORTA=0x10|d[3]; //  قرار دادن ده هزارگان روی چهار پین اول و یک کردن پین پنج  روشن کردن سگمنت چهار
delay_ms(5);                                  //تاخیر جهت ماندگار شدن نور در چشم
PORTA=0xe0; } } }                             //خاموش کردن همه سگمنت ها و پایان برنامه
```

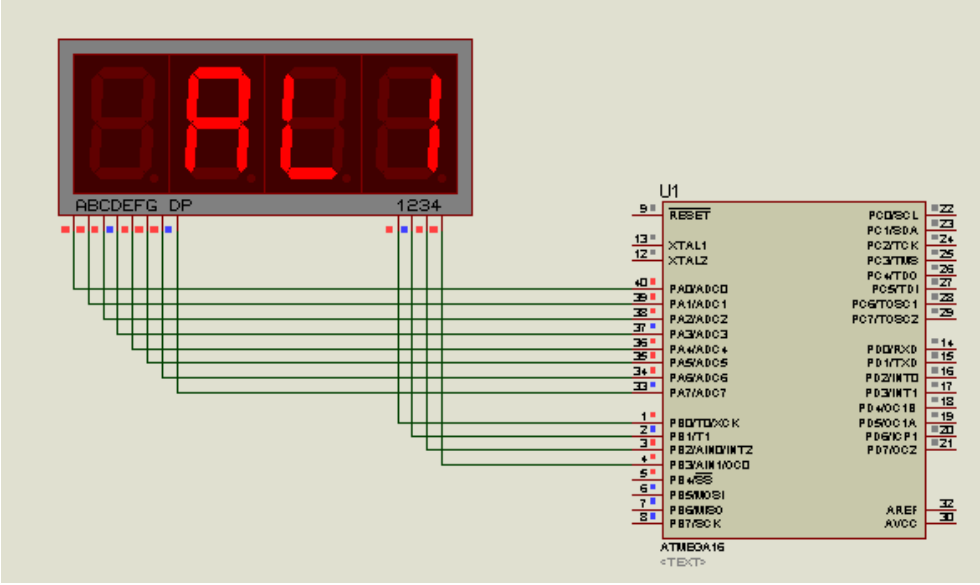


شکل6-4

مثال :می خواهیم کلمه ALI را روی سگمنت ها نمایش داده و آن را به سمت راست شیف دهیم . برای این کار ابتدا باید کد معادل حرف A ، L و I را بدست بیاوریم .

```
#include <mega16.h>                                   معرفی میکرو مورد استفاده//
#include <delay.h>                                    فراخوانی کتابخانه تاخیر زمانی//
char d[]={0x00,0x06,0x38,0x77},i,z=2,r=1,w=0,n=3;     کد ALI وتعریف متغییرها//
while (1)    {                                        حلقه بی نهایت //
    q=0b1111011                                       تعریف یک متغیر برای روشن کردن یکی یکی سگمنت ها //
    for(i=0;i<30;i++)    { // ایجاد یک حلقه برای چند بار نمایش که باعث تاخیر در حرکت است }
     PORTB=q;                                         روشن کردن سگمنت چهارم //
     PORTA=d[w];                                      نمایش حرف مورد نظر //
     delay_ms(15);                                    تاخیر جهت ماندگاری نور در چشم//
     PORTB=0xff;                                      خاموش کردن همه سگمنت ها//
     PORTB=q>>1;                                      روشن کردن سگمنت سوم //
     PORTA=d[r];                                      نمایش حرف مورد نظر//
     delay_ms(15);                                    تاخیر جهت ماندگاری نور در چشم//
     PORTB=0xff;                                      خاموش کردن همه سگمنت ها//
     PORTB=q>>2;                                      روشن کردن سگمنت چهارم //
     PORTA=d[z];                                      نمایش حرف مورد نظر//
     delay_ms(15);                                    تاخیر جهت ماندگاری نور در چشم//
     PORTB=0xff;                                      خاموش کردن همه سگمنت ها//
     PORTB=q>>3;                                      روشن کردن سگمنت چهارم //
```

```
    PORTA=d[n];                                  //نمایش حرف مورد نظر
    delay_ms(15);                                //تاخیر جهت ماندگاری نور در چشم
    PORTB=0xff;  }                               //خاموش کردن همه سگمنت ها
  z++;
  r++;
  w++;
  n++;
}}
```



شکل 7-4

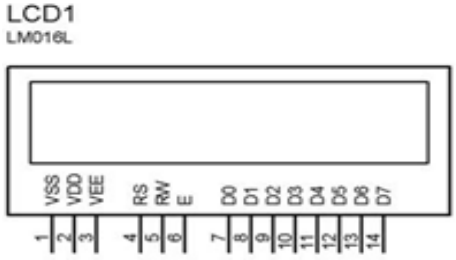
LCD

حال که با پورت ها و پین ها و نحوه کار با آنها تا حدودی آشنا شدید می خواهیم شما را با یکی از پرکاربردترین نمایشگرها که در صنعت استفاده می شود یعنی LCD آشنا کنیم . در این بخش ابتدا به معرفی و نحوه کار با LCD کارکتری می پردازیم و پس از آشنایی کامل با آن به معرفی LCD گرافیکی خواهیم پرداخت که کارکرد آن هم همانند LCD کاراکتری می باشد اما داری قابلیت ها بیشتری هستند .

LCD کاراکتری :

LCD ها عموما بر حسب تعداد سطرها و ستون هایشان نام گذاری می شوند مثلا 2×16 یا 2×20 ، که در آن عدد سمت راست مشخص کننده تعداد سطر و عدد سمت چپ تعداد ستون را نشان می دهد . در اینجا ما بیشتر با LCD های 2×16 کار خواهیم کرد که دارای کاربرد بیشتری هستند .

برای شروع و برای اینکه بدانید که اساس کار LCD ها چگونه است ، ابتدا کار را به صورت 8 سیمه شروع می کنیم و به شما نحوه نوشتن کتابخانه LCD را خواهیم گفت و سپس کار به صورت چهار سیمه و توسط کتابخانه lcd.h را فرا خواهید گرفت . شمای LCD به صورت زیر می باشد :



شکل8-4

VSS : این پایه به زمین وصل می شود .

VDD : به تغذیه 5 ولتی وصل می شود .

VEE : توسط یک پتانسیومتر که به این پایه وصل شده است می توان میزان درخشندگی LCD را تنظیم کرد .

RS : با صفر و یک کردن آن به LCD می فهمانیم که قصد ارسال دستور یا داده جهت نمایش را داریم (RS=0 ارسال دستور ، RS=1 ارسال داده جهت نمایش)

R/W : صفر شدن این پایه برای LCD به معنای قرار گرفتن در حالت دریافت ، و یک شدن آن به معنای قرار گرفتن در حالت دادن اطلاعات به است .

E : پایه فعال ساز LCD می باشد . بعد از هر بار ارسال(یا حتی دریافت) اطلاعات باید این پایه را یک بار یک و بار دیگر صفر کنیم تا طلاعات ذخیره شوند.

D0 تا D7 : توسط این پایه ها اطلاعات را به LCD داده و یا از آن می گیریم .البته اگر به صورت چهار سیمه از LCD استفاده کنیم از پایه های D4 تا D7 استفاده می کنیم .

تذکر :

توجه کنید که در برخی از LCD ها پایه 15 و 16 نیز تحت عنوان A و K وجود دارد . پایه A ، آند LED Backlight است که باید با یک مقاومت 47 اهم به مثبت تغذیه وصل شود . پایه K نیز به زمین وصل می شود .

دستورات LCD

برای اینکه بتوانید با LCD کار کنید و اعمالی را که می خواهید انجام دهید باید دستوراتی را که در جدول صفحه بعد آمده است را فرا بگیرید . این دستورات برای LCD ها با ابعاد مختلف یکسان بوده و قابل استفاده است .

عملکرد	کد دستور(هگز)
پاک کردن صفحه نمایش	1
برگشت مکان نما به مکان اولیه	2
نوشتن از چپ به راست (نوشتن فارسی)	4
نوشتن از راست به چپ(نوشتن ریاضی و لاتین)	6
شیفت کاراکتر نوشته شده به راست	5
شیفت کاراکتر نوشته شده به چپ	7
نمایشگر و مکان نما خاموش	8
نمایشگر خاموش و مکان نما روشن	0A
نمایشگر روشن و مکان نما خاموش	0C
مکان نمای چشمک زن	0D
نمایش و مکان نما روشن	0F
نمایش روشن و مکان نما چشمک بزند	0E
شیفت مکان نما به چپ	10
شیفت مکان نما به راست	14
شیفت همه کاراکترها به سمت راست	18
شیفت همه کاراکترها به سمت چپ	1C
آدرس اولین خانه سطر اول	80
آدرس اولین خانه سطر دوم	C0
آماده سازی **LCD** (قبل از شروع هر کاری باید این کد را ارسال کنیم)	38

جدول2-4

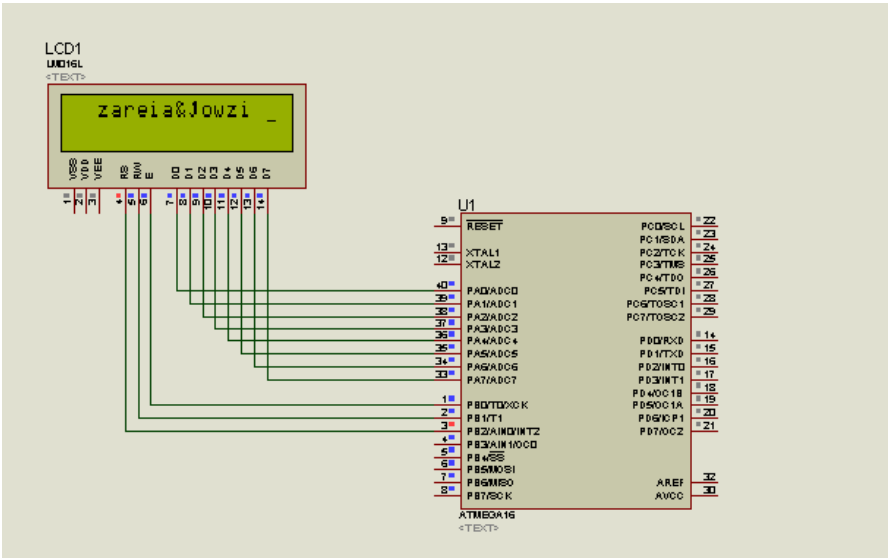
مثال : نام خود را بر روی یک LCD به صورت هشت سیمه نمایش دهید .

```
#include <mega16.h>                    //معرفی میکرو مورد استفاده

#include <delay.h>                     //فراخوانی کتابخانه تاخیر زمان
```

```
#define  en  PORTB.0                    // معرفی پین صفر پورت B با عنوان جدید en
#define  rw  PORTB.1                    // معرفی پین یک پورت B با عنوان جدید rw
#define  rs  PORTB.2                    // معرفی پین دوم پورت B با عنوان جدید rs
unsigned char i,x[13]= “zareia&jowzi”;  // معرفی متغیرها
void send_data(unsigned char data)  {   //تابع ارسال دیتا برای نمایشگر
PORTA=data;                             //قرار دادن متغیرdata بر روی پورت A
rw=0;                                   //قرار دادنLCD در حالت دریافت اطلاعات
rs=1;                                   // قرار دادنLCD در حالت دریافت اطلاعات جهت نمایش
en=1;                                   //ایجاد یک لبه پایین رونده برای ذخیره شدن اطلاعات
en=0;
delay_ms(5);    }                       // تاخیر  برای ذخیره شدن اطلاعات
void send_comm(unsigned char comm.)  {  // ایجاد تابع جهت ارسال دستور بهLCD
PORTA=comm.;                            //قرار دادن متغیرcomm بر روی پورت A
rw=0;                                   //قرار دادن LCD در حالت دریافت اطلاعات
rs=1;                                   //قرار دادنLCD در حالت دریافت دستور العمل و اجرای آن
en=1;                                   //ایجاد یک لبه پایین رونده برای ذخیره شدن اطلاعات
en=0;
delay_ms(5);  }                         // تاخیر  برای ذخیره شدن اطلاعات
void main(void)  {                      // برنامه اصلی
```

```
send_comm(0x38);                                    // آماده سازی LCD
send_comm(0x0e);                                    //نمایشگر روشن شود و مکان نما چشمک زن باشد
send_comm(6);                                       // نوشتن از چپ به راست
send_comm(1);                                       // پاک کردن نمایشگر
send_comm(0x82);                                    //شروع نوشتن از خانه سوم سطر اول
for(i=0;i<13;i++) {                                 //فرستادن یکی یکی کاراکتر های مورد نظر جهت نمایش
send_data(x[i]);     }    }
```

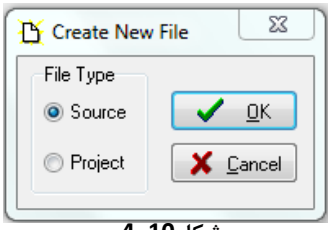


شکل4-9

نحوه ایجاد یک کتابخانه

در بسیاری از مواقع شما می خواهید از یک روند یا یک برنامه در برنامه های دیگر استفاده کنید . برای جلوگیری از تکرار در نوشتن می توانید خود یک کتابخانه بسازید و هر زمان که به آن نیاز داشتید آن را فراخوانی کنید . برای این کار ابتدا در کد ویژن

گزینه create new file را بزنید و گزینه source را انتخاب کنید . سپس برنامه خود را نوشته ودر پوشه inc که محل نصب کد ویژن قرار دارد با پسوند inc ذخیره کنید .



شکل10-4

مثال : می خواهیم برای کار با یک LCD هشت سیمه یک کتابخانه بنویسیم به طوری که

- با دستور lcd_ready() تنظیمات اولیه و آماده سازی نمایشگر انجام شود .
- با دستور lcd_pak نمایشگر پاک شود .
- با دستور lcd_write(" عبارت مورد نظر ") یک عبارت رشته ای نوشته شود .
- با دستور lcd_char(کارکتر مورد نظر) یک کارکتر را نمایش دهیم .
- با دستور lcd_xy(سطر , ستون) به سطر و ستون مورد نظر برود و نوشتن را از آنجا شروع کند .

```
#include <delay.h>                          // فراخوانی کتابخانه تاخیر زمانی
#define  en  PORTB.0                        // معرفی پین صفر پورت B با عنوان جدید en
#define  rw  PORTB.1                        // معرفی پین یک پورت B با عنوان جدید rw
#define  rs  PORTB.2                        // معرفی پین دوم پورت B با عنوان جدید rs
void send_data(unsigned char data)  {       //تابع ارسال دیتا برای نمایشگر
PORTA=data;                                 //قرار دادن متغیر data بر روی پورت A
rw=0;                                       //قرار دادن LCD در حالت دریافت اطلاعات
```

```
rs=1;                                   // قرار دادنLCD در حالت دریافت اطلاعات جهت نمایش
en=1;                                   //ایجاد یک لبه پایین رونده برای ذخیره شدن اطلاعات
en=0;
delay_ms(5);    }                       // تاخیر  برای ذخیره شدن اطلاعات
void send_comm(unsigned char comm.) { // ایجاد تابع جهت ارسال دستور بهLCD
PORTA=comm.;                            // قرار دادن متغیرcomm بر روی پورت A
rw=0;                                   //قرار دادن LCD در حالت دریافت اطلاعات
rs=1;                                   //قرار دادنLCD در حالت دریافت دستور العمل و اجرای آن
en=1;                                   //ایجاد یک لبه پایین رونده برای ذخیره شدن اطلاعات
en=0;
delay_ms (5);   }                       // تاخیر  برای ذخیره شدن اطلاعات
void lcd_pak (void)    {                // تابع پاک کردن نمایشگر
send_comm (0x1);       }
lcd_xy (unsigned char y,unsigned char x) {    //تابع آدرس دهی به نمایشگر برای شروع نوشتن
if(y==1) {  k=0x80|x;  }                // آدرس دهی برای رفتن به ستون مورد نظر در سطر اول
if(y==2) {  k=0xc0|x;  }                //آدرس دهی برای رفتن به ستون مورد نظر در سطر دوم
send_comm (k);              }           // فرستادن دستورات برای نمایشگر
void lcd_ready (void)       {           //تابع تنظیمات اولیه و آماده سازی نمایشگر
send_comm (0x38);                       //ارسال دستور آماده شدن برای نمایشگر
send_comm (0x6);                        //نوشتن از راسته به چپ
```

```
send_comm (0x1);                                    پاک کردن نمایشگر//
send_comm (0xf);                                    نمایشگر و مکان نما روشن //
lcd_xy  (1,0);          }                           رفتن نمایشگر به سطر و ستون اول جهت شروع //
نوشتن
void  lcd_char  (unsigned char p)       {           تابع نمایش یک کاراکتر//
send_data  (p);          }                          ارسال کاراکتر برای نمایش//
void lcd_write  (flash unsigned char *f)      {     تابع نمایش یک رشته //
for(i=0;i<32;i++)   {
if(*f!=0) {
send_data  (*(f++));          } }                   نمایش رشته ذخیره شده درROM //
```

با ذخیره کردن برنامه فوق در پوشه inc ، هر گاه در خلال برنامه آن را فراخوانی کرده و از آن استفاده کنید. فرض کنید کتابخانه فوق را با نام bb ذخیره کرده ایم حال اگر بخواهیم آن را فراخوانی کرده و یک نام یا هر چیزی دیگر را بر روی آن نمایش دهیم به شیوه زیر عمل می کنیم .

```
#include <mega16.h>                                 معرفی میکرو مورد استفاده//
#include <bb.h>                                     فراخوانی کتابخانه مورد نظر//
void main(void)    {                                برنامه اصلی//
lcd_ready();                                        تنظیمات اولیه نمایشگر//
lcd_xy(2,0);                                        رفتن به اولین خانه سطر دوم//
```

```
lcd_putsf(" ***zareia*** “);  }                نوشتن عبارت مورد نظر //
```

کتابخانه lcd.h

این کتابخانه یکی از امکانات خوب کدویژن می باشد که کار با LCD را برای کاربر بسیار راحت کرده و نیاز به نوشتن برنامه بلند و ایجاد یک کتابخانه نیست . مزیت دیگر این کتابخانه، چهار سیمه بودن خط انتقال اطلاعات (به جای هشت سیمه) است که سبب می شود توسط یک پورت یک LCD را راه اندازی کرد .

توجه کنید که مطالبی که قبل از این برای کار کردن هشت سیمه با LCD بیان شد جهت درک و فهمیدن شما از نحوه کار LCD می باشد . مطالب مذکور می تواند به شما در هنگام برخورد با سایر میکروها همچون خانواده 8051 که به صورت هشت سیمه کار می کنند دچار مشکل نشوید .

توابع lcd به دو دسته توابع سطح بالا و توابع سطح پایین تقسیم می شود .

الف: دستورات سطح پایین

1- تابع lcd_write_data();

به کمک این تابع کد های مربوط به تنظیمات بر روی lcd ارسال می شود .

2- تابع lcd_ready();

این تابع اماده بودن lcd را جهت گرفتن اطلاعات جدید مشخص می کند و قبل از تابع قبلی نوشته می شود .

3- تابع lcd_write_byte(LCD RAM ادرس خانه , داده مورد نظر)

جهت نوشتن کد کاراکترهای دلخواه در خانه های ازاد LCD RAM

4- تابع (آدرس) lcd_read_byt

جهت خواندن کد کاراکترهای دلخواه از خانه های ازاد LCD RAM

ب : توابع سطح بالا

این توابع به کمک توابع سطح پایین ساخته می شود .

1- تابع (تعداد ستون های lcd) lcd-init

این تابع جهت اماده سازی و تنظیمات اولیه lcd می باشد .

2- تابع lcd_clear();

از این تابع جهت پاک کردن lcd استفاده می باشد .

3- تابع (سطر , ستون); lcd_gotoxy

این تابع ادرس مکان نما جهت نوشتن در lcd را مشخص می کند .

4- تابع (کاراکتردلخواه); lcd_putchar

برای نمایش یک کاراکتر دلخواه بر روی lcd از این تابع استفاده می کنیم .

5 و 6 : توابع (رشته ذخیره شده در ROM); lcd_puts و

(رشته ذخیره شده در RAM); lcd_putsf

از این توابع برای برای نمایش رشته های ذخیره در RAM یا ROM استفاده می شود .

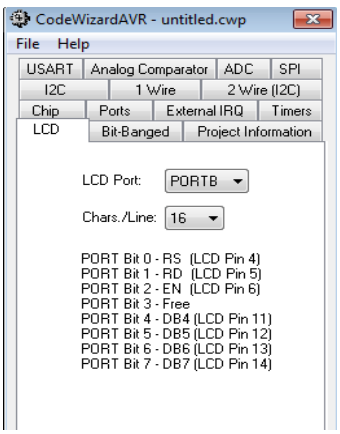
نکته : برای نوشتن کارکترهای استاندارد در تابع sprint از x/ مطابق مثال زیر استفاده می کنیم

```
Unsigned char a=10;  s[15];
Sprint(s,"a=%u/xf4",a)
Lcd_puts(s);
```

در مثال بالا کد f4 کد مربوط به علامت اهم می باشد که در این صورت ما بر روی lcd عبارت a=10Ω را شاهد خواهیم بود .

تنظیمات CodeWizard

برای انجام تنظیمات مربوط به LCD کافی است که در code wizard بر روی گزینه LCD کلیک کرده و پورتی را که می خواهیم LCD به ان متصل شود را انتخاب کنیم که به صورت زیر انجام می شود :

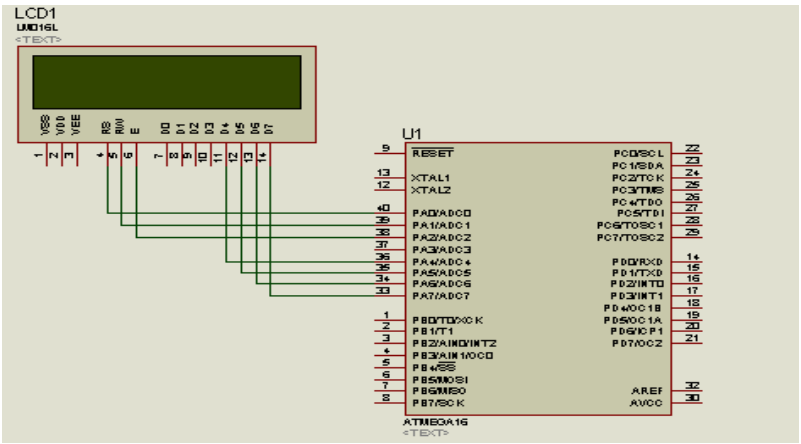


شکل 11-4

مثال : برنامه ای بنویسید که نام شما در سطر اول LCD دائما به سمت چپ و راست حرکت کند

```
#include <mega16.h>                                  // معرفی میکرو مورد استفاده
#asm                                                 // شروع برنامه اسمبلی
.equ __lcd_port=0x1B ;PORTB                          // معرفی پورت A به برای وصل شدن به نمایشگر
#endasm                                              // پایان برنامه اسمبلی
#include <lcd.h>                                    // فراخوانی کتابخانه lcd
#include <delay.h>                                   // فراخوانی کتابخانه تاخیر زمانی
unsigned char a,b;                                   // معرفی متغیر ها
void main(void)     {                                // برنامه اصلی
lcd_init(16);                                        // آماده سازی نمایشگر
while (1)    {                                       // حلقه بی نهایت
     delay_ms(700);                                  // تاخیر زمانی
      lcd_clear();                                   // پاک کردن نمایشگر
      lcd_gotoxy(a,0);                               // حرکت کلمات نوشته شده
     lcd_putsf("fariborz");                          // نمایش کلمه مورد نظر
     if (b==0)      {
     a--;                                            // حرکت دادن کلمات به چپ
     if(a==0)  {  b=1;  }    }
     if(b==1)   {
     a++;                                            // حرکت کلمات به راست
```

```
if (a==8) { b=0 ; }   } } }
```



شکل12-4

نحوه ایجاد کاراکترهای دلخواه بروی LCD

اگر بخواهیم کاراکتری دلخواه را بر روی LCD نمایش دهیم (مثلا نوشتن حروف فارسی) از حافظه LCD موسوم به CGRAM که می تواند تا هشت کاراکتر (64 بایت) را در خود ذخیره کند، کمک می گیریم بدین ترتیب که توسط یک جدول 8×5 کد معادل کاراکتر مورد نظر که یک کد هشت بایتی است را بدست آورده و در CGRAM ذخیره می کنیم و آنها را نمایش می دهیم .

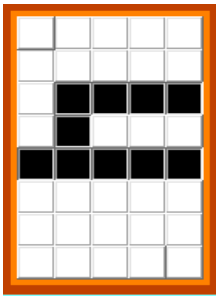
نکته :

در LCD RAM از خانه حافظه 0X40 به بعد 64 بایت برای ذخیره کدهای کاراکترهای دلخواه در نظر گرفته شده است از آنجا که هر کد دلخواه 8 بایت را اشغال می کند پس حداکثر به صورت همزمان 8 کد دلخواه خواه خواهیم داشت که برای استفاده از آنها در LCD به صورت زیر عمل می کنیم.

Lcd_putchar(شماره کد 0 تا 7)

نحوه ساخت کد کاراکتر دلخواه :

برای این کار مطابق شکل زیر از یک جدول 8×5 استفاده می کنیم . کاراکتر دلخواه را روی آن می نویسیم خانه های رنگی به معنای یک و خانه های خالی به منزله صفر است.



شکل 13-4

تذکر :

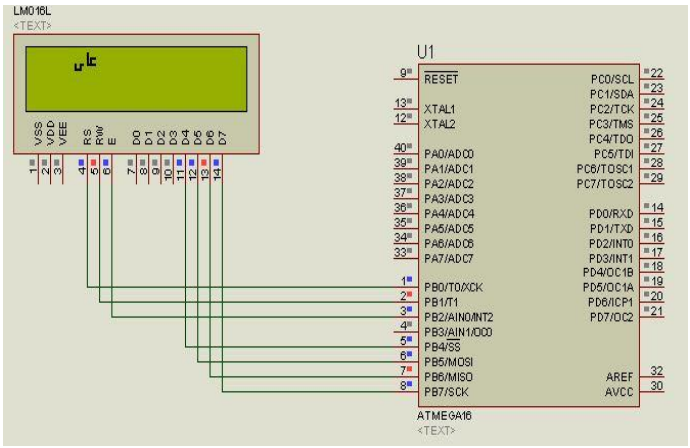
کدهای ساخته شده از نوع ثابت می باشد .

مثال:

کلمه "علی" را بر روی LCD نمایش دهید .

```
#include <mega16.h>                                  // معرفی میکرو مورد استفاده
#include <lcd.h>                                     //فراخوانی کتابخانه lcd.h
flash char x0[8]= {0x0,0x0,0x0,0x0,0x0,0x11,0x11,0x11,0x1F};  //کد معادل حرف "ی"
flash char x1[8]={0x10,0x10,0x10,0x10,0x1F,0x0,0x0,0x0};  //کد معادل حرف "ل"
flash char x2[8]={0x0,0x0,0x0,0x7,0x4,0x1F,0x0,0x0,0x0};  //کد معادل حرف "ع"
void ram_lcd(flash char*a,char b) {     //ایجاد یک تابع برای ذخیره کاراکتر ها در حافظه نمایشگر
char i,c;                                            // معرفی متغیر ها
c=(b<<3)|0x40;                                       //رفتن به آدرس اولین خانه حافظه نمایشگر
```

```
for(i=0;i<8;i++)     {                    //ارسال یک حلقه برای ارسال هشت بایت
lcd_write_byte(c++,*a++);  }  }  //فرستادن کد کاراکتر برای ذخیره شدن در حافظه نمایشگر
void main(void)     {                    // برنامه اصلی
lcd_init(16) ;                           // آماده سازی نمایشگر
ram_lcd(x0,0);                           // ذخیره کاراکتر"ی" با نام 0
ram_lcd(x1,1);                           // ذخیره کاراکتر"ل" با نام1
ram_lcd(x2,2);                           // ذخیره کاراکتر"ع" با نام2
while (1)  {
    lcd_gotoxy                           //رفتن نمایشگر به ستون چهارم سطر اول برای شروع نمایش
    lcd_putchar(0);                      //نمایش کاراکتر "ی"
   lcd_putchar(1);                       // نمایش کاراکتر "ل"
    lcd_putchar(1);       }     }        //نمایش کاراکتر "ع" و پایان برنامه
```



شکل14-4

LCD گرافیکی

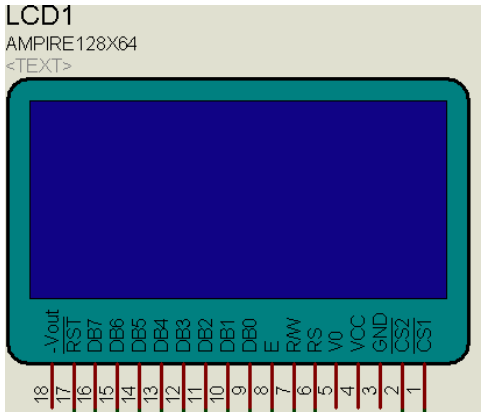
این LCD ها دارای پیکسل بالایی هستند به همین دلیل می توان توسط آنها عکس یا کاراکتر های دلخواه را نمایش داد . دو مشخصه مهم این LCD ها تعداد سطر و ستون آنها می باشد که از میان انها دو نوع 128×64 و 240×128 از همه رایج تر می باشد .

در اینجا ما در مورد نوع 128×64 توضیحاتی را ارائه خواهیم کرد و یک نقاشی را بر روی آن نمایش خواهیم داد . تصویر این LCD را می توانید در زیر مشاهده کنید .

همانگونه که از شکل پیداست

این نوع LCD دارای 18 پایه می باشد

که پایه های هم نام آن با پایه های LCD کارکتری همان کارها را انجام می دهند اما چند پایه اضافی که موجود هستند را با عملکردشان توضیح خواهیم داد .



شکل15-4

این LCD خود از دو LCD با ابعاد 64×64 که کنار هم قرار داده شده اند تشکیل شده است توسط پایه های CS1 و CS2 می توان مشخص کرد که از کدام یک می خواهید استفاده کنید .

Reset : هرگاه این پایه فعال شود رجیسترهایX,Y,Z بر روی مقادیر اول تنظیم میشود.

رجیستر X : در ای LCD 64 سطر داریم که هر هشت سطر یک Page نامیده می شود. رجیستر X مشخص می کند که کدام Page فعال باشد . برای Page0 ، x=0xB8 و اخرین Page ، x=0xBF می باشد .

رجیستر Y : این رجیستر مشخص می کند که کدام ستون فعال باشد . برای ستون اول y=0x40 و برای ستون آخر Y= ox7F می باشد . توجه کنید که هرگاه Data نوشته شود Y به طور خودکار یک واحد افزایش یابد .

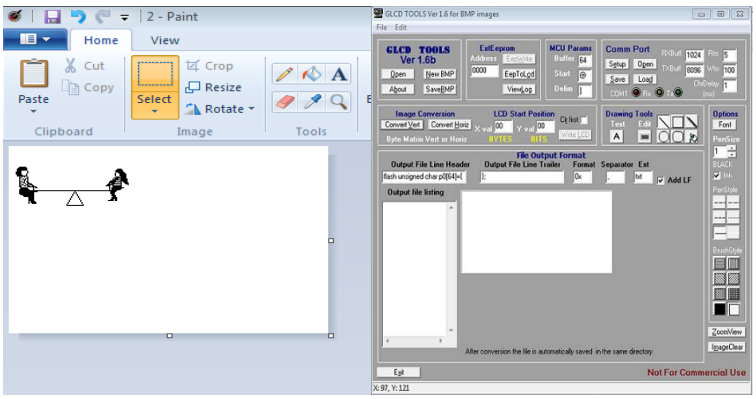
رجیستر Z : این رجیستر مشخص می کند که Page صفر از کدام سطر شروع شود و آدرس آن 0xc0 تا 0xFF می باشد .

روشن و خاموش کردن نمایش : برای این منظور از دو دستور 3E برای روشن کردن و 3F برای خاموش کردن استفاده می کنیم .

نکته : قبل از روشن کردن LCD با دستور 3F هر دو CS را فعال کنید .

انتقال تصویر به LCD :

ابتدا توسط یک برنامه نقاشی مانند Paint صفحه ای به ابعاد LCD باز کنید . برای مثال در Paint از منوی فایل گزینه Properties را انتخاب کنید و طول و عرض LCD را وارد کرده و رنگ را سیاه و سفید انتخاب کنید . سپس نقاشی مورد نظر را بکشید . آن را با نامی دلخواه و پسوند bmp ذخیره کنید . حال یکی از نرم افزار های تبدیل تصویر به کد (دراینجا ما از نرم افزار GLCD TOOLS استفاده کرده ایم) را اجرا کرده فایل نقاشی را open کرده و در قسمت Output File Line Header عبارت Output File Line Header را وارد کنید .



شکل16-4

در قسمت Trailer عبارت ;{و در قسمت Format عبارت 0x و در قسمت separator" , " را قرار می دهیم بسته به نوع LCD (که دو نمایشگر 64*64 آن عمودی به وصل شده اند یا افقی)یکی از دو گزینه عمودی (Conververt Vert) یا عمودی (Convert Horiz)را انتخاب می کنیم که در اینجا ما از Convert Vert استفاده می کنیم . کد های در قسمت output File listing ایجاد می شود آنها را کپی کرده و در برنامه خود معرفی کنید . به دلیل حجم بالای برنامه از نوشتن آن در این قسمت خود داری می کنیم .

```
#include <mega32.h>                    //معرفی میکرو مورد استفاده
#include <delay.h>                     // فراخوانی کتابخانه تاخیر زمانی
#define rs PORTB.0                     //معرفی پین صفر پورت B با نام جدید rs
#define rw PORTB.1                     //معرفی پین صفر پورتB با نام جدید rs
#define e  PORTB.2                     //معرفی پین صفر پورتB با نام جدید rs
#define cs1 PORTB.3                    //معرفی پین صفر پورتB با نام جدید rs
#define cs2 PORTB.4                    //معرفی پین صفر پورتB با نام جدید rs
#define reset PORTB.5                  //معرفی پین صفر پرتB با نام جدید rs
unsigned char i,x,z;                   //معرفی متغییر ها
flash unsigned char d0[128]={کد بدست آمده}
flash unsigned char d1[128]={کد بدست آمده}
flash unsigned char d2[128]={کد بدست آمده}
flash unsigned char d3[128]={کد بدست آمده}
flash unsigned char d4[128]={کد بدست آمده}
```

```
flash unsigned char d5[128]={کد بدست آمده}
flash unsigned char d6[128]={کد بدست آمده}
flash unsigned char d7[128]={کد بدست آمده}
void comm(unsigned char c)    {                    //تابع ارسال دستور به نمایشگر
PORTD=c;                                          //قرار دادن دستورات بر روی پورت c
rs=0;                                             //قرار دادنLCD در حالت دریافت دستور العمل و اجرای آن
rw=0;                                             //قرار دادنLCD در حالت دریافت اطلاعات
e=1;                                              //ایجاد یک لبه پایین رونده برای ذخیره شدن اطلاعات
e=0;
delay_us(200);     }                              // تاخیر  برای ذخیره شدن اطلاعات
void datw(unsigned char c)    {                   //تابع ارسال دیتا به نمایشگر جهت نمایش
PORTD=c;                                          //قرار دادن دیتا بر روی پورتc
rs=1;                                             // قرار دادنLCD در حالت دریافت اطلاعات جهت نمایش
rw=0;                                             //قرار دادنLCD در حالت دریافت اطلاعات
e=1;                                              //ایجاد یک لبه پایین رونده برای ذخیره شدن اطلاعات
e=0;
delay_us(200);       }                            // تاخیر  برای ذخیره شدن اطلاعات


void cls(void)  {                                 //تابع پاک کردن نمایشگر
```

```
for(x=0xb8;x<0xc0;x++)  {        //ایجاد یک حلقه برای ارسال آدرس سطر مورد نظر(سطر یکم تا هشتم)
comm(x);                                                    //رفتن به سطر مورد نظر
comm(0x40);                                                 // ارسال آدرس ستون اول و رفتن به آن
for(i=0;i<64;i++)                                           //ایجاد یک حلقه برای پاک کردن تمام ستون های سطر مورد نظر
datw(0x00);          } }                                    //ارسال دستور پاک کردن
void main(void)    {                                        // برنامه اصلی
DDRD=0xff;                                                  //خروجی کردن پورت D
DDRB=0xff;                                                  //خروجی کردن پورتB
reset=0;                                                    //صفر کردن پایه reset
reset=1;                  //یک کردن resetو قرار گرفتن رجیستر هایx,y,x بر روی مقادیر اولیه خود
cs1=1;                                                      //روشن کردن یکی ازLCDهای64*64
cs2=1;                                                      //روشن کردن یکی دیگر از LCD های64*64
comm(0x3f);                                                 //فعال کردن هر دو نمایشگر
cls();                                                      // پاک کردن هر دو نمایشگر

                                        //------PAGE 0 ------
cs1=1;                                                      // روشن کردن نمایشگر شماره یک
cs2=0;                                                      //خاموش کردن نمایشگر شماره دو
comm(0x40);                                                 //رفتن به ستون اول
comm(0xb8);                                                 //رفتن به سطر اول
```

```
for(i=0;i<64;i++)                    // ارسال تصویر بر روی نیمی از نمایشگر
datw(d0[i]);                         // ارسال اطلاعات برای نمایش
cs1=0;                               // خاموش کردن نمایشگر اول
cs2=1;                               //روشن کردن نمایشگر دوم
comm(0x40);                          //رفتن به ستون اول
comm(0xb8);                          //رفتن به سطر اول
for(i=64;i<128;i++)                  // ارسال تصویر بر روی نیمه دوم نمایشگر
datw(d0[i]);                         // ارسال اطلاعات برای نمایش
                                     //------PAGE1 ------
cs1=1;                               // روشن کردن نمایشگر شماره یک
cs2=0;                               //خاموش کردن نمایشگر شماره دو
comm(0x40);                          //رفتن به ستون اول
comm(0xb9);                          //رفتن به سطر دوم
for(i=0;i<64;i++)                    //ارسال نیمی از کارکترهای متغیر دوم بر روی نمایشگر شماره یک
datw(d0[i]);                         // ارسال اطلاعات برای نمایش
cs1=0;                               // خاموش کردن نمایشگر اول
cs2=1;                               //روشن کردن نمایشگر دوم
comm(0x40);                          //رفتن به ستون اول
comm(0xb9);                          //رفتن به سطردوم
```

```
for(i=64;i<128;i++)     ارسال نیمه دوم از کارکترهای متغیر دوم بر روی نمایشگر شماره دو//
datw(d0[i]);            ارسال اطلاعات برای نمایش //
                        //------PAGE 2 ------
cs1=1;                  روشن کردن نمایشگر شماره یک //
cs2=0;                  خاموش کردن نمایشگر شماره دو//
comm(0x40);             رفتن به ستون اول//
comm(0xba);             رفتن به سطرسوم//
for(i=0;i<64;i++)       ارسال نیمی از کارکترهای متغیر سوم بر روی نمایشگر شماره یک//
datw(d0[i]);            ارسال اطلاعات برای نمایش //
cs1=0;                  خاموش کردن نمایشگر اول //
cs2=1;                  روشن کردن نمایشگر دوم//
comm(0x40);             رفتن به ستون اول//
comm(0xba);             رفتن به سطرسوم//
for(i=64;i<128;i++)     ارسال نیمه دوم از کارکترهای متغیر سوم بر روی نمایشگر شماره دو//
datw(d0[i]);            ارسال اطلاعات برای نمایش //
                        //------PAGE 3 ------
cs1=1;                  روشن کردن نمایشگر شماره یک //
cs2=0;                  خاموش کردن نمایشگر شماره دو//
comm(0x40);             رفتن به ستون اول//
```

```
comm(0xbb);                                    رفتن به سطر چهارم//
for(i=0;i<64;i++)          ارسال نیمی از کارکترهای متغیرچهارم بر روی نمایشگر شماره یک//
datw(d0[i]);                                   ارسال اطلاعات برای نمایش //
cs1=0;                                         خاموش کردن نمایشگر اول //
cs2=1;                                         روشن کردن نمایشگر دوم//
comm(0x40);                                    رفتن به ستون اول//
comm(0xbb);                                    رفتن به سطر چهارم//
for(i=64;i<128;i++)    ارسال نیمه دوم از کارکترهای متغییر چهارم بر روی نمایشگر شماره دو//
datw(d0[i]);                                   ارسال اطلاعات برای نمایش //
                                    //------PAGE 4 ------
cs1=1;                                         روشن کردن نمایشگر شماره یک //
cs2=0;                                         خاموش کردن نمایشگر شماره دو//
comm(0x40);                                    رفتن به ستون اول//
comm(0xbc);                                    رفتن به سطر پنجم//
for(i=0;i<64;i++)          ارسال نیمی از کارکترهای متغیر پنجم بر روی نمایشگر شماره یک//
datw(d0[i]);                                   ارسال اطلاعات برای نمایش //
cs1=0;                                         خاموش کردن نمایشگر اول //
cs2=1;                                         روشن کردن نمایشگر دوم//
comm(0x40);                                    رفتن به ستون اول//
```

```
comm(0xbc);                                    رفتن به سطر پنجم//
for(i=64;i<128;i++)    ارسال نیمه دوم از کارکترهای متغییر پنجم بر روی نمایشگر شماره دو//
datw(d0[i]);                                   ارسال اطلاعات برای نمایش //
                                    //------PAGE 5------
cs1=1;                                         روشن کردن نمایشگر شماره یک //
cs2=0;                                         خاموش کردن نمایشگر شماره دو//
comm(0x40);                                    رفتن به ستون اول//
comm(0xbd);                                    رفتن به سطر ششم//
for(i=0;i<64;i++)          ارسال نیمی از کارکترهای متغیر ششم بر روی نمایشگر شماره یک//
datw(d0[i]);                                   ارسال اطلاعات برای نمایش //
cs1=0;                                         خاموش کردن نمایشگر اول //
cs2=1;                                         روشن کردن نمایشگر دوم//
comm(0x40);                                    رفتن به ستون اول//
comm(0xbd);                                    رفتن به سطر ششم//
for(i=64;i<128;i++)    ارسال نیمه دوم از کارکترهای متغیر ششم بر روی نمایشگر شماره دو//
datw(d0[i]);                                   ارسال اطلاعات برای نمایش //
                                    //------PAGE 6------
cs1=1;                                         روشن کردن نمایشگر شماره یک //
cs2=0;                                         خاموش کردن نمایشگر شماره دو//
```

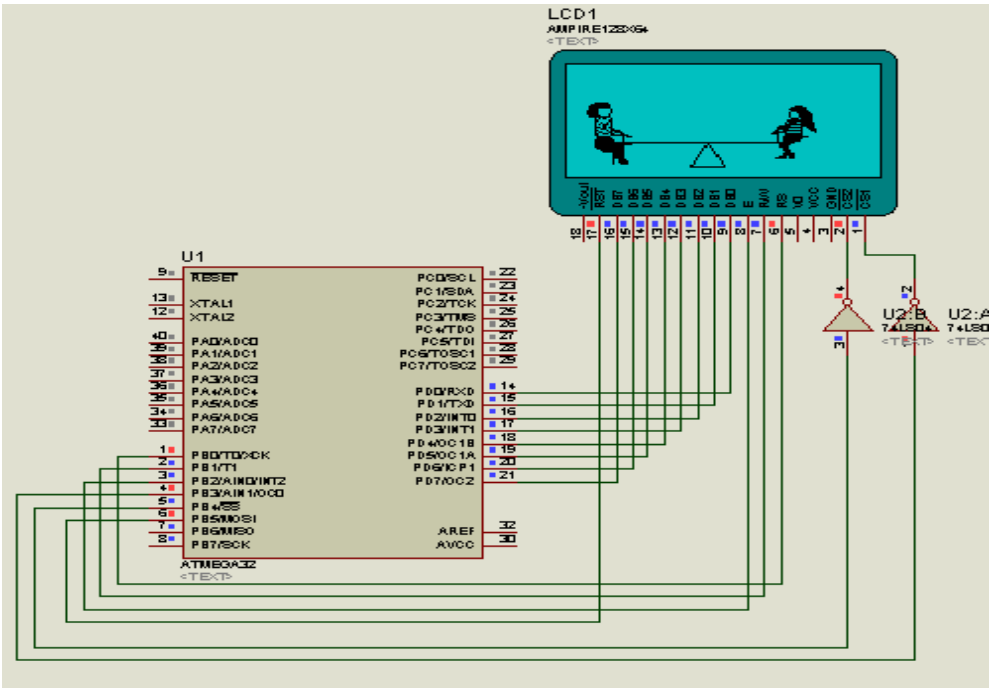
```
comm(0x40);                                   رفتن به ستون اول//
comm(0xbe);                                   رفتن به سطر هفتم//
for(i=0;i<64;i++)          ا رسال نیمی از کارکترهای متغیر هفتم بر روی نمایشگر شماره یک//
datw(d0[i]);                                  ارسال اطلاعات برای نمایش //
cs1=0;                                        خاموش کردن نمایشگر اول //
cs2=1;                                        روشن کردن نمایشگر دوم//
comm(0x40);                                   رفتن به ستون اول//
comm(0xbe);                                   رفتن به سطر هفتم//
for(i=64;i<128;i++)  ارسال نیمه دوم از کارکترهای متغیر هفتم بر روی نمایشگر شماره دو//
datw(d0[i]);                                  ارسال اطلاعات برای نمایش //
                          //------PAGE 7 ------
cs1=1;                                        روشن کردن نمایشگر شماره یک //
cs2=0;                                        خاموش کردن نمایشگر شماره دو//
comm(0x40);                                   رفتن به ستون اول//
comm(0xbf);                                   رفتن به سطر هشتم//
for(i=0;i<64;i++)    ارسال نیمی از کارکترهای متغیر هشتم بر روی نمایشگر شماره یک//
datw(d0[i]);                                  ارسال اطلاعات برای نمایش //
cs1=0;                                        خاموش کردن نمایشگر اول //
cs2=1;                                        روشن کردن نمایشگر دوم//
```

رفتن به ستون اول// comm(0x40);

رفتن به سطر هشتم// comm(0xbf);

ارسال نیمه دوم از کارکترهای متغیر هشتم بر روی نمایشگر شماره دو// for(i=64;i<128;i++)

ارسال اطلاعات برای نمایش // } datw(d0[i]);



شکل 17-4