کتاب PDF

میکرو کنترلرهای AVR با پروژه های ۱۰۰٪ عملی

تألیف جابر الوندی

منبع درس ریزپردازنده۱ دانشگاه پیام نور

فایل PDF موجود نسخه RF، در بین سایر فایل های PDF ریزپردازنده موجود در سایت های دانلود کتاب، کامل ترین فایل PDF محسوب میشود. چرا که در سایر کتاب های PDF مشابه، علاوه بر ناقص بودن صفحات کتاب، نظم و ترتیب خاصی بین صفحات وجود ندارد که این نقص، باعث ایجاد مشکل و سردرگمی برای دانشجویان در هنگام مطالعه کتاب PDF می شود.

از همین رو ویرایش RF کتاب پی دی اف ریزپردازنده۱ را که نواقص فایل های قبلی را برطرف کرده، برای استفاده دانشجویان عزیزی که به کتاب اصلی این درس دسترسی ندارند، آماده کرده ایم.

Edited by Aref.b *4123ph@gmail.com*

فصل ۴

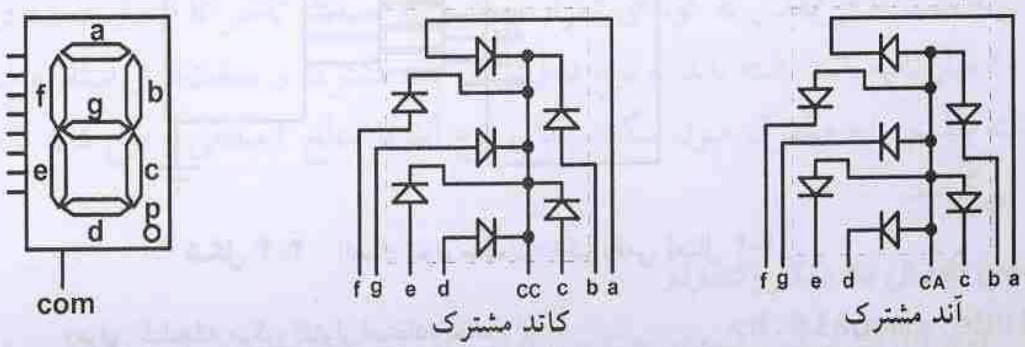
نمایشگرها و صفحه کلید ۴×۴

اهداف

۱. آشنایی با نمایشگر سون سگمنت تک رقمی
۲. بکارگیری روش مالتی پلکسری برای سون سگمنت‌های چند رقمی
۳. آشنایی و نحوه‌ی کار با نمایشگر LCD
۴. ایجاد کردن کاراکتر دلخواه در LCD
۵. اسکن صفحه کلید ۴×۴

۴-۱ نمایشگر سون سگمنت (Seven Segment Display)

یکی از نمایشگرهای پر استفاده در میکروکنترلر سون سگمنت‌ها در ابعاد و ارقام مختلف می‌باشند. این نوع نمایشگر از هفت قسمت (LED) تشکیل شده است به همین دلیل به آن سون سگمنت گفته می‌شود. این نمایشگرها به خاطر نوردهی و اندازه فیزیکی آنها نسبت به LCD ارجعیت دارند. برای اتصال سون سگمنت به میکروکنترلر نیاز به المانی مانند 7447 یا 7448 است که کد باینری یا BCD را به کد سون سگمنت تبدیل کند اما برای کاهش سخت‌افزار جانبی می‌توان کد سون سگمنت را در داخل برنامه ایجاد کرد.



شکل ۴-۱ نمای سون سگمنت تک رقمی و ترکیب داخلی آن

آند مشترک										کاتد مشترک									
عدد	کد	p	g	F	E	d	c	b	A	عدد	کد	p	G	f	e	d	c	b	a
0	C0	1	1	0	0	0	0	0	0	0	3F	0	1	1	1	1	1	1	1
1	F9	1	1	1	1	1	0	0	1	1	06	0	0	0	0	0	1	1	0
2	A4	1	0	1	0	0	1	0	0	2	5B	0	1	0	1	1	0	1	1
3	B0	1	0	0	1	0	0	0	0	3	4F	0	1	0	0	1	1	1	1
4	99	1	0	0	1	1	0	0	1	4	66	0	1	1	0	0	1	1	0
5	92	1	0	0	1	0	0	1	0	5	6D	0	1	1	0	1	1	0	1
6	82	1	0	0	0	0	0	1	0	6	7D	0	1	1	1	1	1	0	1
7	F8	1	1	1	1	1	0	0	0	7	07	0	0	0	0	0	1	1	1
8	80	1	0	0	0	0	0	0	0	8	7F	0	1	1	1	1	1	1	1
9	90	1	0	0	1	0	0	0	0	9	6F	0	1	1	0	1	1	1	1

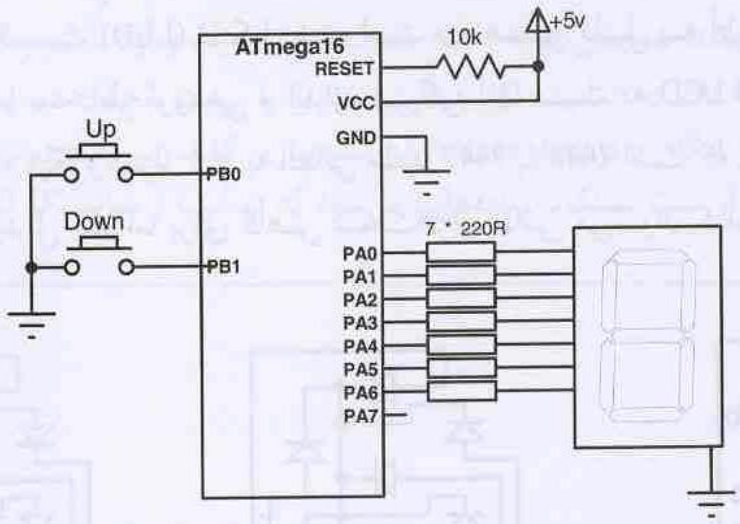
جدول ۴-۱ کدهای سون سگمنت نوع آند مشترک و کاتد مشترک

نکته: فرم پایه‌های سون سگمنت تک رقمی یا چند رقمی ترتیبی نیست و باید با یک آزمایش اولیه با تغذیه 2 ولت یا تغذیه 5 ولت همراه با یک مقاومت ۳۳۰ اهم، ترتیب پایه‌ها را متناسب با a تا g و p پیدا کنیم.

در فصل دوم، دو مثال از شمارنده 0 تا 9 مطرح کردیم که می‌توانند مثال‌های خوبی برای این فصل باشند. یک نمونهٔ دیگر از کار با سون سگمنت تک رقمی را مطرح می‌کنیم.

مثال ۴-۱: برنامه‌ای بنویسید که وضعیت دو کلید فشاری Up و Down را بررسی نماید و مقدار عددی نمایشگر تک رقمی نوع کاتد مشترک را افزایش و یا کاهش دهد؟

حل:



شکل ۴-۲ اتصال سون سگمنت تک رقمی [مثال ۴-۱]

```
#include <mega16.h>          // معرفی کتابخانه میکروکنترلر استفاده شده
#include <delay.h>           // معرفی کتابخانه تأخیر زمانی
```

```
flash unsigned char display[]={    //تعریف آرایه ثابت حاوی کدهای سون سگمنت
0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};
void main(){                       //تابع اصلی برنامه
unsigned char i;                   //تعریف متغیر شمارشگر
PORTA=0x3F;                        //ابتدا بر روی سون سگمنت عدد صفر را نشان می‌دهیم
DDRA=0xFF;                         //پورت A متصل به سون سگمنت را خروجی تعیین می‌کنیم
PORTB=0x03;                        //فعال کردن مقاومت Pull-up پایه‌های متصل به کلیدها
DDRB=0x00;                         //تعیین پورت B به عنوان ورودی
while(1){                          //حلقه بی نهایت برای خواندن مکرر وضعیت کلیدها
 if(PINB.0==0 && i<9){             //اگر کلید Up فشرده شود و متغیر شمارشگر کوچکتر از ۹ باشد
  i++;                             //یک واحد به متغیر شمارشگر اضافه می‌کنیم
  while(PINB.0==0);                //تست برای رها شدن کلید Up
  }
 if(PINB.1==0 && i!=0){            //اگر کلید Down فشرده شود و متغیر شمارشگر مخالف صفر باشد
  i--;                             //یک واحد متغیر شمارشگر را کاهش می‌دهیم
  while(PINB.1==0);                //تست برای رها شدن کلید Down
 }
PORTA=display[i];                  //کد سون سگمنت معادل متغیر شمارشگر، به خروجی ارسال می‌شود
 };
}
```

روش مالتی پلکسری (Multiplexing Method)

اگر نیاز باشد از نمایشگرهای سون سگمنت چند رقمی استفاده کنیم این مسئله بوجود می‌آید که با محدودیت پورت میکروکنترلر مواجعه می‌شویم. برای این منظور از روش Refresh یا تازه‌سازی استفاده می‌شود این عمل بر اساس خطای چشم انسان صورت می‌گیرد. اگر یک LED را 50 بار در ثانیه روشن و خاموش کنیم چشم انسان LED را مدام روشن می‌بیند. بنابراین اگر پایه‌های مشترک سون سگمنت را در اختیار میکروکنترلر قرار دهیم و کدها را به صورت خروجی یک دیکدر، به پایه‌های مشترک ارسال کنیم و مانند یک مالتی‌پلکسر داده‌های نمایشی را با زمانی مشخص تسهیم کنیم، می‌توانیم ادعا کنیم که تمام داده‌های نمایشی را توانسته‌ایم نمایش دهیم. روش بدین صورت است که هر بار یکی از سگمنت‌ها را انتخاب کرده و دیتای آن سگمنت را ارسال می‌کنیم و این عمل را برای سگمنت‌های بعدی به گونه‌ای تکرار می‌کنیم که حداکثر تأخیر تا ارسال مجدد دیتای سگمنت اول، ۲۰ میلی‌ثانیه باشد. البته باید به نوع فعال شدن پایه مشترک و سخت‌افزار استفاده شده، توجه لازم را داشته باشیم. پایه مشترک سون سگمنت‌های نوع آند با سطح 1 منطقی و نوع کاتد با سطح 0 منطقی فعال می‌گردند.

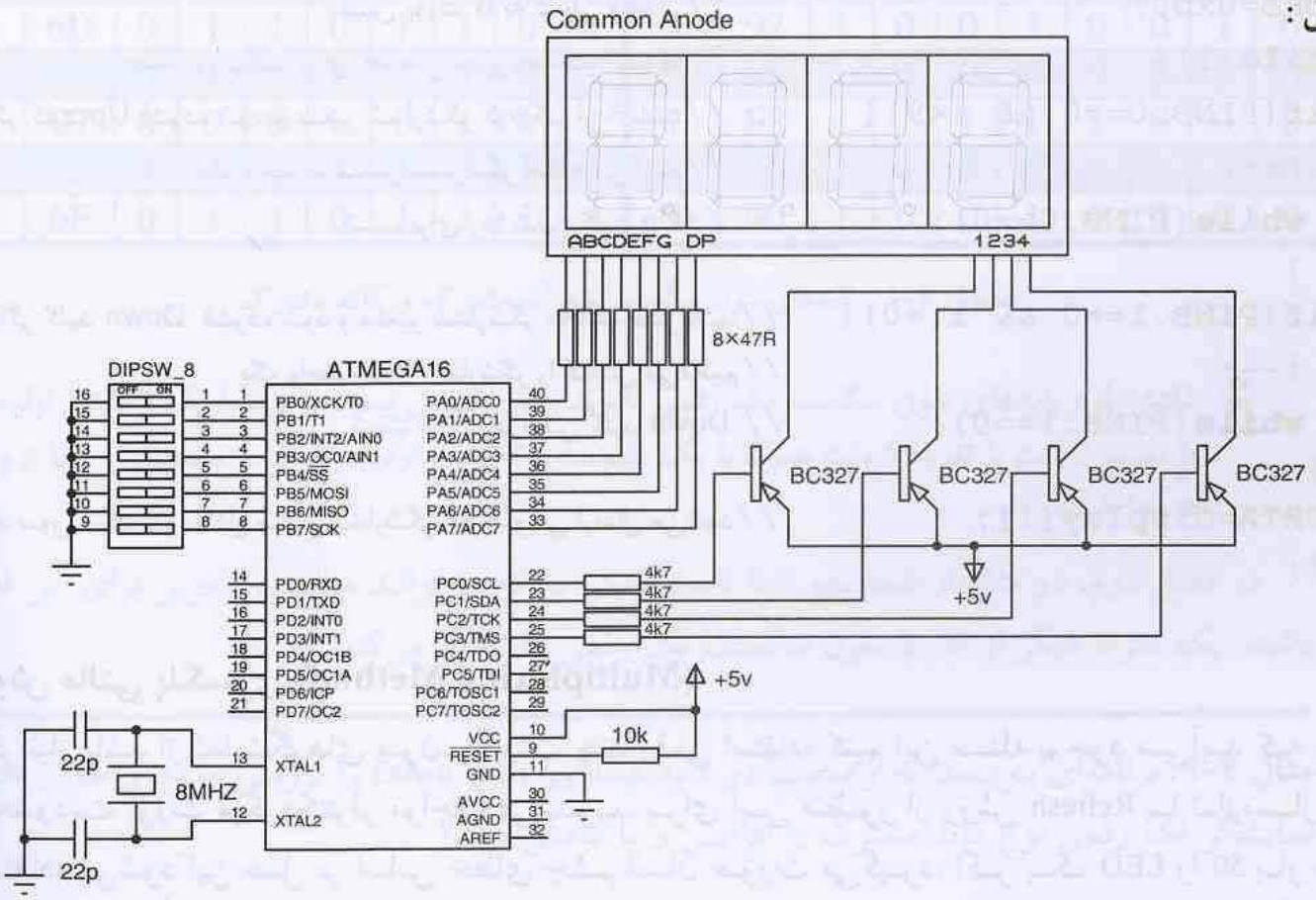
نحوه‌ی اتصال به میکروکنترلر

۱. **نوع آند مشترک**: پایه‌های a تا g و p به یک پورت دلخواه متصل می‌گردد و پایه‌های مشترک توسط ترانزیستور مطابق شکل ۴-۳ به چهار پایه از یک پورت دلخواه دیگر متصل می‌گردد.

۲. **نوع کاتد مشترک** : پایه‌های a تا g و p با مقاومت‌های 150Ω به یک پورت دلخواه متصل شده و پایه‌های مشترک، مستقیماً به چهار پایه از یک پورت دلخواه دیگر متصل می‌گردد. زیرا با مقاومت ۱۵۰ اهم، حداکثر از هر پایه پورت، 34mA کشیده خواهد شد.

مثال ۴-۲ : یک عدد ۸ بیتی از یک دسته کلید متصل به پورت B بخوانید و آن عدد را بر روی نمایشگر سون سگمنت از نوع آند مشترک چهار رقمی به روش مالتی پلکسری نمایش دهید؟

حل :



شکل ۴-۳ نحوه‌ی اتصال سون سگمنت مالتی پلکسری [مثال ۴-۲]

```
#include <mega16.h>                          //معرفی میکروکنترلر استفاده شده
#include <delay.h>                           //معرفی کتابخانه تأخیر زمانی
unsigned char  part1=0,part2=0,part3=0,part4=0;  //متغیرهای سگمنت‌ها
flash unsigned char C7seg[]={                //کدهای سون سگمنت ذخیره شده در حافظه ثابت
0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90};
void HEX_to_seg(unsigned char k){            //تابع تبدیل عدد از هگزاد به کد سون سگمنت
    part4 = C7seg[k%10];                     //تبدیل یکان به کد معادل سون سگمنت
    part3 = C7seg[k/10%10];                  //تبدیل دهگان به کد معادل سون سگمنت
    part2 = C7seg[k/10/10];                  //تبدیل صدگان به کد معادل سون سگمنت
    part1 = C7seg[0];                        //عدد ۸ بیتی است و سگمنت سمت چپ همیشه صفر است
}
void main(){                                 //تابع اصلی برنامه
```

```
unsigned char number=0;          // تعریف متغیر ۸ بیتی برای خواندن ورودی
PORTA=0xFF;                      // مقدار اولیه جهت خاموش بودن تمام سگمنت‌ها
DDRA=0xFF;                       // تعیین پورت A به عنوان خروجی
PORTB=0xFF;                      // فعال کردن مقاومت‌های Pull-up پورت B متصل به کلیدها
DDRB=0x00;                       // تعیین پورت B به عنوان ورودی
PORTC=0x0F;                      // مقدار اولیه جهت خاموش بودن تمام سگمنت‌ها
DDRC=0x0F;                       // تعیین نیبل کم ارزش پورت C به عنوان خروجی
while (1){                       // حلقه بی نهایت جهت خواندن ورودی و نمایش مکرر عدد خوانده شده
number=PINB;                     // خواندن پورت ورودی متصل به کلیدها
HEX_to_seg(number);              // فراخوانی تابع جهت تبدیل به کد سون سگمنت
PORTC=0b00001110;                // انتخاب سگمنت یکان
PORTA=part1;                     // ارسال عدد یکان
delay_ms(5);                     // تأخیر زمانی به مدت ۵ میلی ثانیه جهت تازه‌سازی
PORTC=0b00001101;                // انتخاب سگمنت دهگان
PORTA=part2;                     // ارسال عدد دهگان
delay_ms(5);                     // تأخیر زمانی به مدت ۵ میلی ثانیه جهت تازه‌سازی
PORTC=0b00001011;                // انتخاب سگمنت صدگان
PORTA=part3;                     // ارسال عدد صدگان
delay_ms(5);                     // تأخیر زمانی به مدت ۵ میلی ثانیه جهت تازه‌سازی
PORTC=0b00000111;                // انتخاب سگمنت هزارگان
PORTA=part4;                     // ارسال عدد هزارگان
delay_ms(5);                     // تأخیر زمانی به مدت ۵ میلی ثانیه جهت تازه‌سازی
 };
}
```

۴-۲ نمایشگر LCD متنی

امروزه در اکثر پروژه های میکروکنترلری از نمایشگر LCD استفاده می‌شود. این نمایشگر توانایی نمایش تمام حروف های موجود بر روی صفحه کلید کامپیوتر را دارد و از مزیت های آن نسبت به سون سگمنت این است که ارقام و حروف‌های بیشتری را می‌توان نمایش داد و نیازی به تازه‌سازی اطلاعات جهت نمایش نمی‌باشد یعنی اگر یک کاراکتر را به LCD ارسال کنیم، آن کاراکتر نمایش داده می‌شود و نیازی نیست که دوباره همان کاراکتر را ارسال کنیم تا از بین نرود.

LCDها در ابعاد مختلفی می‌باشند از جمله: 1×16 ، 2×16 ، 2×20 ، 4×20 ، 2×40 که عدد سمت چپ مشخص کننده تعداد ستون و عدد سمت راست مشخص کننده تعداد سطر می‌باشد.

نمایشگرهای LCD دارای دستوراتی می‌باشند که کاربر می‌تواند بنا به نیاز خود از آنها استفاده کند. در جدول ۴-۴ فهرستی از دستورات LCD آورده شده است توجه کنید که دستورات برای تمام LCDها در ابعاد مختلف یکسان است.

شماره پایه	نام پایه	توصیف
1	VSS	پایه زمین تغذیه LCD است
2	VDD	پایه مثبت تغذیه LCD است که به 5v+ متصل می‌گردد
3	VO	پایه درخشندگی که با یک پتانسیومتر 5k یا مقاومت ثابت 1k تنظیم می‌شود
4	RS	این پایه اگر صفر باشد دیتای ورودی فرمان و اگر یک باشد دیتای ورودی کاراکتر
5	R/W	صفر شدن این پایه یعنی عملیات نوشتن و یک شدن آن یعنی عملیات خواندن
6	E	پایه فعال ساز LCD است و نوع فعال‌سازی یک لبه پایین رونده می‌باشد
7	DB0	بیت اول دیتای ورودی LCD که در نوع راه‌اندازی ۴ بیتی آزاد است
8	DB1	بیت یکم دیتای ورودی LCD که در نوع راه‌اندازی ۴ بیتی آزاد است
9	DB2	بیت دوم دیتای ورودی LCD که در نوع راه‌اندازی ۴ بیتی آزاد است
10	DB3	بیت سوم دیتای ورودی LCD که در نوع راه‌اندازی ۴ بیتی آزاد است
11	DB4	بیت چهارم دیتای ورودی LCD می‌باشد
12	DB5	بیت پنجم دیتای ورودی LCD می‌باشد
13	DB6	بیت ششم دیتای ورودی LCD می‌باشد
14	DB7	بیت هفتم دیتای ورودی LCD و پرچم وضعیت مشغول LCD می‌باشد
15	A	پایه آند LED Backlight که با یک مقاومت 47Ω به 5v+ متصل می‌گردد
16	K	پایه کاتد LED Backlight که به زمین تغذیه متصل می‌شود

جدول ۴-۲ فرم پایه‌های LCD متنی 2×16

دستور	عملکرد	دستور	عملکرد
0x01	صفحه نمایش پاک شود	0x0E	نمایش روشن و مکان نما روشن
0x02	بازگشت به مکان اول	0x0F	نمایش روشن و مکان نما چشمک بزند.
0x04	جابجایی مکان نما به چپ	0x10	جابجایی محل مکان نما به چپ
0x06	جابجایی مکان نما به راست	0x14	جابجایی محل مکان نما به راست
0x05	جابجایی کاراکترها به راست	0x18	همهٔ کاراکترهای نمایش داده شده، یک خانه به چپ جابجا می‌شوند.
0x07	جابجایی کاراکترها به چپ	0x1C	همهٔ کاراکترهای نمایش داده شده، یک خانه به راست جابجا می‌شوند.
0x08	نمایش خاموش و مکان نما خاموش	0xC0	مکان نما به آغاز خط دوم می‌رود.
0x0A	نمایش خاموش و مکان نما روشن	0x38	سازماندهی ۸ بیتی و ماتریس ۵×۸
0x0C	نمایش روشن و مکان نما خاموش	0x28	سازماندهی ۴ بیتی و ماتریس ۵×۸

جدول ۴-۳ دستورات نمایشگر LCD

کتابخانه lcd.h

فایل کتابخانه‌ای LCD دارای توابعی به صورت زیر است و باید ابتدای برنامه معرفی گردد.

```
void _lcd_ready(void);
```
این تابع پرچم مشغول LCD را بررسی می‌کند.

```
Void _lcd_write_data(unsigned char data);
```
برای ارسال فرمان به LCD استفاده می‌شود.

```
void lcd_write_byte(unsigned char addr, unsigned char data);
```
برای نوشتن یک بایت کاراکتر دلخواه یا نمایشی در RAM داخلی LCD استفاده می‌شود.

```
unsigned char lcd_read_byte(unsigned char addr);
```
برای خواندن یک بایت کاراکتر دلخواه یا نمایشی در RAM داخلی LCD استفاده می‌شود.

```
void lcd_gotoxy(unsigned char x, unsigned char y);
```
برای تعیین موقعیت (ستون x و برای تعیین سطر y) نمایش کاراکتر LCD است.

```
void lcd_clear(void);
```
این تابع کل صفحه نمایش LCD را پاک می‌کند.

```
void lcd_putchar(char c);
```
برای ارسال یک کاراکتر به LCD استفاده می‌شود.

```
void lcd_puts(char *str);
```
برای نمایش یک رشته کاراکتری ذخیره شده در RAM میکروکنترلر به LCD استفاده می‌شود.

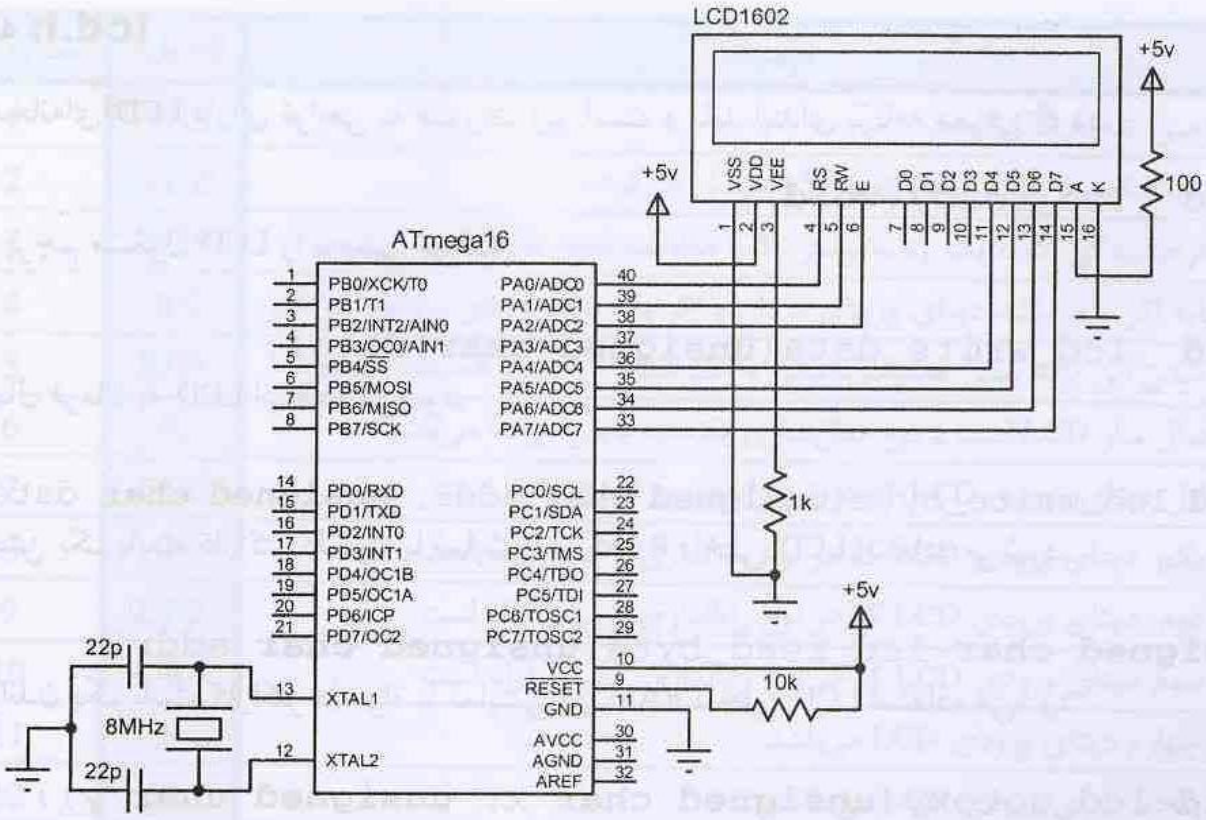
```
void lcd_putsf(char flash *str);
```
برای نمایش یک رشته کاراکتری ذخیره شده در Flash میکروکنترلر به LCD استفاده می‌شود.

```
unsigned char lcd_init(unsigned char lcd_columns);
```
برای مقدار دهی اولیه و تعیین تعداد ستون LCD این تابع در ابتدای برنامه فراخوانی می‌شود.

مثال ۴-۳: ابتدا بر روی سطر اول رشته ثابت "LCD test" را نمایش داده و سپس بر روی سطر دوم رشته ثابت "ATmega16" را نمایش دهید و به انتهای سطر اول رفته و کاراکتر * را نمایش دهید و فرمان چشمک زدن مکان نما را ارسال نمائید؟

حل :

```
#include <mega16.h>                    // معرفی کتابخانه میکروکنترلر استفاده شده
#asm                                   // شروع برنامه اسمبلی
```



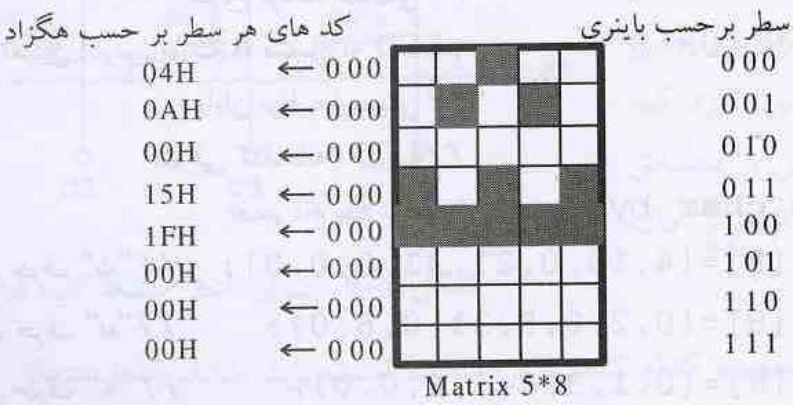
شکل ۴-۴ نحوه‌ی اتصال LCD به پورت دلخواه A [مثال ۴-۳]

```
   .equ __lcd_port=0x1B          ;تعیین آدرس پورت A متصل به LCD
#endasm                          //پایان برنامه اسمبلی
#include <lcd.h>                 //معرفی کتابخانه نمایشگر LCD
void main(){                     //تابع اصلی برنامه
lcd_init(16);                    //تعیین lcd با ۱۶ ستون و مقدار دهی اولیه
lcd_clear();                     //پاک کردن کل صفحه نمایش lcd
lcd_gotoxy(2,0);                 //رفتن به ستون سوم و سطر اول lcd
lcd_putsf("LCD test");           //نمایش رشته کاراکتری ذخیره شده در حافظه Flash
lcd_gotoxy(5,1);                 //رفتن به ستون ششم و سطر دوم lcd
lcd_putsf("ATmega16");           //نمایش رشته کاراکتری ذخیره شده در حافظه Flash
lcd_gotoxy(14,0);                //رفتن به سطر اول و ستون پانزدهم lcd
lcd_putchar('*');                //نمایش کاراکتر *
lcd_gotoxy(14,0);                //رفتن به سطر اول و ستون پانزدهم lcd
_lcd_ready();                    //خواندن پرچم مشغول lcd
_lcd_write_data(0x0f);           //ارسال فرمان نمایش روشن و مکان نما چشمک بزند
while (1);                       //حلقه بی نهایت و بیکار برنامه
}
```

ایجاد کردن کاراکتر دلخواه در LCD متنی

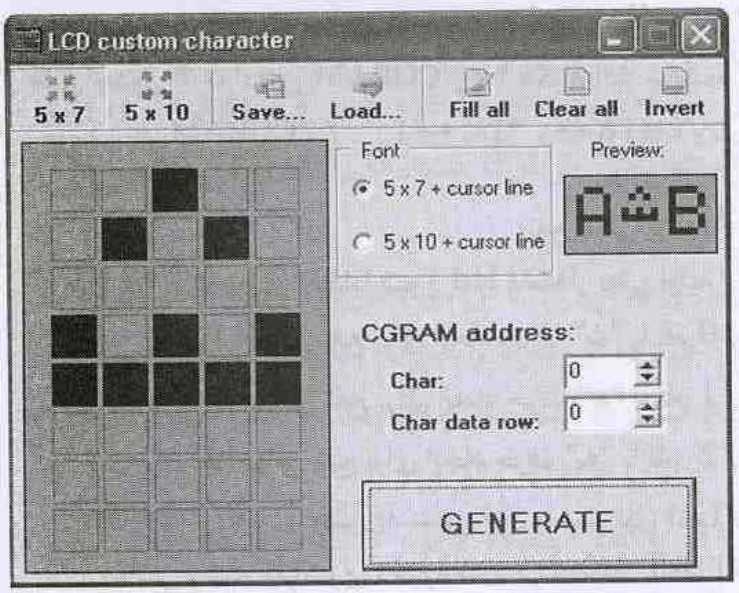
در LCDهای متنی حافظه‌ای بنام CGRAM وجود دارد که می‌توان ۸ کاراکتر دلخواه در آن ایجاد کرد.

نمایشگرهای LCD از ماتریس‌های کوچک ۸×۵ یا ۷×۵ ایجاد شده‌اند بنابراین برای نمایش کاراکتر دلخواه باید کد مربوط به سطر ماتریس را بدست بیاوریم و درحافظه CGRAM، آن کاراکتر را ایجاد کنیم. به طور نمونه حرف "ش" را به صورت زیر ایجاد می‌کنیم و کد هر سطر را بدست می‌آوریم و در آرایه برنامه قرار می‌دهیم. شکل ۴-۵ روش در آوردن کد به صورت دستی را نشان می‌دهد.



شکل ۴-۵ ماتریس ۸×۵ نمایشگر LCD

شکل ۴-۶ نمونه‌ای از یک برنامه کامپیوتری جهت در آوردن کدهای کاراکتر دلخواه را نشان می‌دهد. با کلیک چپ بر روی هر خانه از ماتریس می‌توان آن را روشن و یا خاموش کرد و در پایان با کلیک بر روی گزینه GENERATE کدها به سه زبان مختلف همراه با برنامه ایجاد می‌گردد که شما فقط کدهای ایجاد شده در آرایه را کپی می‌کنید.



شکل ۴-۶ پنجره نرم‌افزار ایجاد کننده کاراکتر دلخواه

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	0	0	0	0	0	0
				0	0	1	0	0	1
				0	1	0	0	1	0
				0	1	1	0	1	1
				1	0	0	1	0	0
				1	0	1	1	0	1
				1	1	0	1	1	0
				1	1	1	1	1	1
شرط دستور		شرط دستیابی به آدرس CGRAM		محل ذخیره سازی کاراکتر (نام کاراکتر برای فراخوانی)			آدرس هر سطر از کاراکتر دلخواه		
				آدرس CGRAM					

جدول ۴-۵ نحوه‌ی دستیابی CGRAM برای ایجاد کاراکتر دلخواه

مثال ۴-۴: کلمه "شنبه" را در حافظه گرافیکی LCD ایجاد کرده و آن را نمایش دهید. از سخت‌افزار شکل ۴-۴ استفاده کنید؟

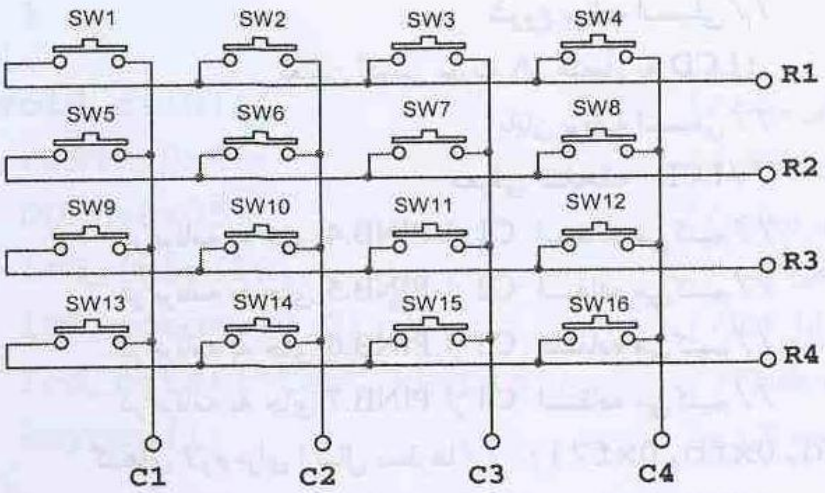
حل:

```
#include <mega16.h>                    //معرفی میکروکنترلر استفاده شده
#asm                                   //شروع برنامه اسمبلی
   .equ __lcd_port=0x1B                ;تعیین آدرس پورت A متصل به LCD
#endasm                                //پایان برنامه اسمبلی
#include <lcd.h>                       //معرفی کتابخانه LCD
typedef unsigned char byte;            //تغییر نام نوع داده
flash byte char0[8]={4,10,0,21,31,0,0,0};   //کدهای حرف "ش"
flash byte char1[8]={0,2,0,5,31,0,8,0};     //کدهای حرف "نب"
flash byte char2[8]={0,1,3,3,1,0,0,0};      //کدهای حرف "ه"
//____>>> این تابع کدهای هر سطر کاراکتر دلخواه را از آرایه آن به حافظه CGRAM کپی می‌کند
void define_char(byte flash *pc,byte char_code){
byte i,a;                              //معرفی دو متغیر ۸ بیتی بدون علامت
a=(char_code<<3)|0x40;                 //طبق جدول ۴-۵ آدرس CGRAM از ۴۰ هگزاد آغاز می‌شود
for (i=0; i<8; i++) lcd_write_byte(a++,*pc++);  //نوشتن در CGRAM
}
void main(){                           //تابع اصلی برنامه
lcd_init(16);                          //تعیین تعداد ستون lcd و مقدار دهی اولیه
define_char(char0,0);                  //فراخوانی تابع برای ایجاد حرف "ش" با نام 0
define_char(char1,1);                  //فراخوانی تابع برای ایجاد حرف "نب" با نام 1
define_char(char2,2);                  //فراخوانی تابع برای ایجاد حرف "ه" با نام 2
lcd_gotoxy(0,0);                       //رفتن به ستون اول و سطر اول Lcd
lcd_putsf("Display day:");             //نمایش رشته کاراکتری ذخیره شده در حافظه Flash
lcd_putchar(2);                        //نمایش حرف "ه"
lcd_putchar(1);                        //نمایش حرف "نب"
lcd_putchar(0);                        //نمایش حرف "ش"
while (1){                             //حلقه بی نهایت
 };
}
```

۴-۳ اسکن صفحه کلید ۴×۴

در بعضی از پروژه‌ها لازم است تا میکروکنترلر اطلاعاتی را توسط کاربر جهت تنظیم کردن، دریافت کند. به طور نمونه یک میکروکنترلر ۴۰ پایه، دارای ۳۲ خط ورودی و خروجی است. بنابراین برای اتصال ۱۶ کلید فشاری، به نصفی از ورودی و خروجی‌ها نیاز داریم. ولی می‌توان ۱۶ کلید فشاری را طوری به میکروکنترلر وصل کرد که با ۸ پایه، بتواند کلیدها را تشخیص دهد برای این منظور از روش

ماتریسی استفاده می‌شود. به صفحه کلیدی که از به هم پیوستن کلیدهای فشاری به صورت ماتریسی بوجود می‌آید، صفحه کلید هگزاد (keypad) گفته می‌شود. نمونه‌های آماده این صفحه کلیدها در بازارهای الکترونیکی با روکش‌های متفاوت وجود دارد که می‌توانید یا خود این ماتریس را مطابق شکل ۴-۷ بسازید یا اینکه از نوع آماده آن استفاده کنید.



شکل ۴-۷ نمای داخلی صفحه کلید 4×4 (روش ماتریسی)

برای اینکه میکروکنترلر بفهمد کدام کلید فشرده شده است باید بداند، کلید متعلق به کدام سطر و کدام ستون است. روش ماتریسی به این صورت است که ستون‌ها را با مقاومت‌های Pull-up داخلی یا بیرونی به 5v+ متصل می‌کنیم. پس در حالت عادی (کلید فشرده نشده) هر چهار ستون در وضعیت یک منطقی هستند و جهت اسکن صفحه کلید باید به سطرها به صورت دیکدر اکتیو Low دیتا بفرستیم و ستون‌ها را برای صفر شدن بررسی نمائیم. پس در این حالت سطرها خروجی و ستون‌ها ورودی هستند این روش به صورت معکوس نیز قابل اجرا است یعنی می‌توان ستون‌ها را خروجی و سطرها را ورودی در نظر گرفت.

ارسال کد سطرها با سرعت بالایی انجام می‌گیرد. در شکل ۴-۷ فرض بگیرید R1 را صفر کرده و مابقی سطرها را 1 کرده‌ایم و در این حالت کلید SW1 فشرده می‌شود و ستون C1 را 0 می‌کند. برنامه ما طوری نوشته می‌شود که در این حالت کد کلید فشرده شده شماره یک را، از حافظه Flash میکروکنترلر برگرداند

از کد بدست آمده می‌توان برای تنظیمات خاصی در برنامه استفاده کرد و یا کد مربوطه را بر روی نمایشگرهای LCD یا Sevensegment و . . . نمایش داد.

همچنین برای لرزش گیری، از یک تأخیر زمانی به مدت 50ms استفاده می‌کنیم و کلید فشرده شده را تست می‌کنیم که این کلید رها شده باشد و به معنی دو بار فشردگی نیز محسوب نشود.

البته به روش‌های سخت‌افزاری نیز مانند قرار دادن خازن و یا اشمیت تریگر می‌توان لرزش کلیدها را مهار نمود اما بهترین روش آن است که کمترین سخت‌افزار را استفاده نمائیم.

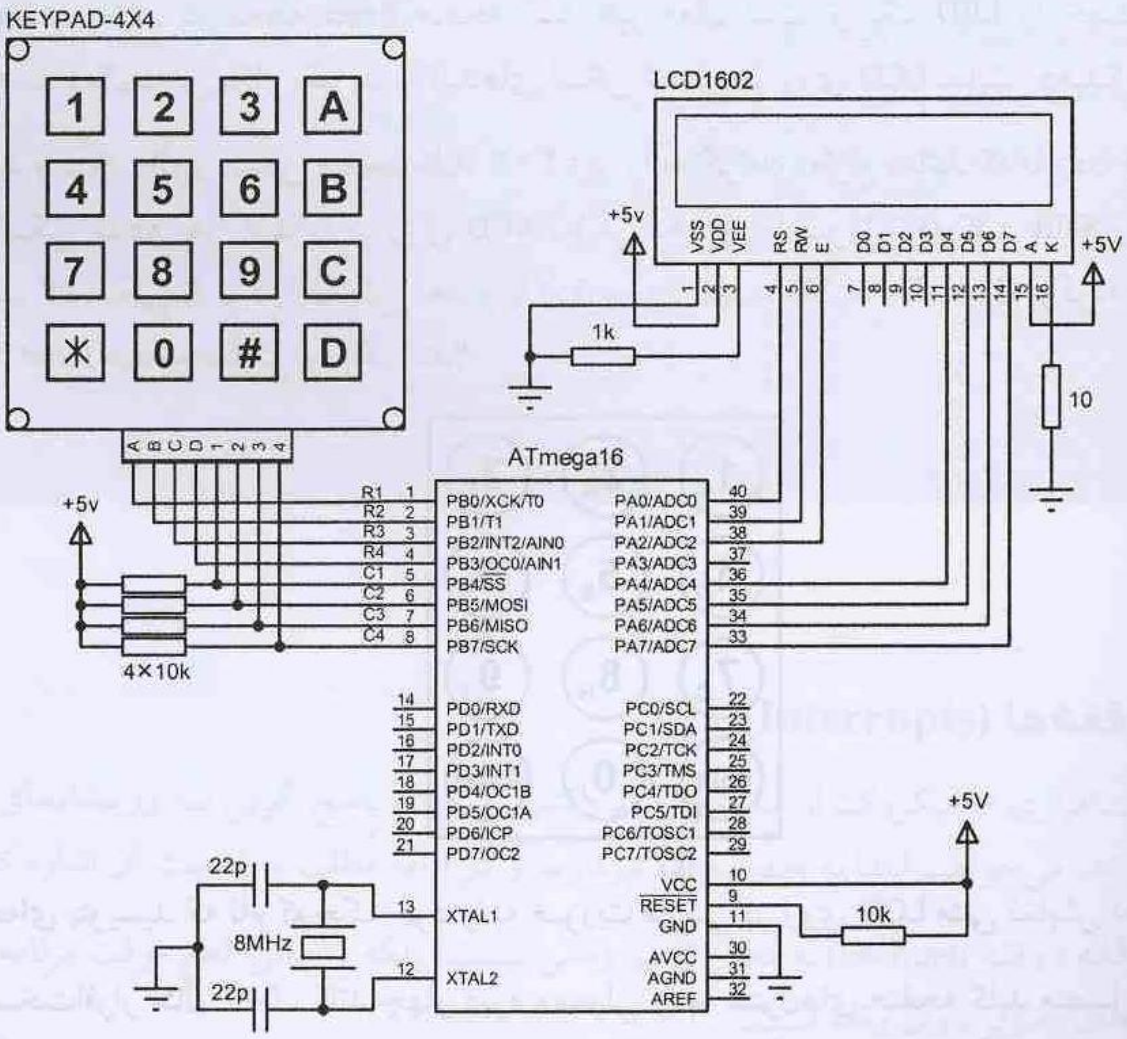
مثال ۴-۵: برنامه‌ای بنویسید که صفحه کلید ۴×۴ را اسکن کرده و کد مربوط به هر کلید را بر روی LCD نمایش دهد در این برنامه ستون‌ها را ورودی و سطرها را خروجی در نظر بگیرید؟

حل:

```
#include <mega16.h>          // معرفی میکروکنترلر استفاده شده
#include <delay.h>           // معرفی کتابخانه تأخیر زمانی
```

```
#asm                                    //شروع برنامه اسمبلی
 .equ __lcd_port=0x1B                   ;تعیین آدرس پورت A متصل به LCD
#endasm                                 //پایان برنامه اسمبلی
#include <lcd.h>                        //معرفی کتابخانه LCD
#define c1 PINB.4                       //در برنامه به جای PINB.4 از C1 استفاده می‌کنیم
#define c2 PINB.5                       //در برنامه به جای PINB.5 از C2 استفاده می‌کنیم
#define c3 PINB.6                       //در برنامه به جای PINB.6 از C3 استفاده می‌کنیم
#define c4 PINB.7                       //در برنامه به جای PINB.7 از C4 استفاده می‌کنیم
flash char row[]={0xfe,0xfd,0xfb,0xf7}; //کدهای لازم برای ارسال سطرها
flash char data_key[]={                 //قرار دادن کدهای هر کلید در حافظه ثابت
'1','2','3','A',                        //کدهای اسکی سطر اول صفحه کلید
'4','5','6','B',                        //کدهای اسکی سطر دوم صفحه کلید
'7','8','9','C',                        //کدهای اسکی سطر سوم صفحه کلید
'*','0','#','D'};                       //کدهای اسکی سطر چهارم صفحه کلید
unsigned char ac,table;                 //معرفی متغیرهای کلی و ۸ بیتی
unsigned int r;                         //معرفی متغیر کلی و ۱۶ بیتی
void keypad(){                          //تابع اسکن صفحه کلید
lcd_gotoxy(0,1);                        //رفتن به ستون اول و سطر دوم LCD
lcd_putsf("~");                         //نمایش کاراکتر →
while (1){                              //حلقه بی نهایت برای اسکن مداوم
 for (r=0;r<4;r++){                     //ایجاد یک حلقه، برای ارسال ۴ دیتا دیکدری به سطرها
 ac=4;                                  //متغیر ac را ۴ قرار می‌دهیم و بعداً آن را تست می‌کنیم
 PORTB=row[r];     //متغیر r توسط حلقه افزایش و کدهای سطرها را بر می‌گرداند و به پورت می‌فرستد
 DDRB=0x0f;                             //نیبل پایینی خروجی و نیبل بالایی ورودی
   if (c1==0)   ac=0;                   //اگر کلیدی از ستون اول فشرده شد متغیر ac=0
   if (c2==0)   ac=1;                   //اگر کلیدی از ستون دوم فشرده شد متغیر ac=1
   if (c3==0)   ac=2;                   //اگر کلیدی از ستون سوم فشرده شد متغیر ac=2
   if (c4==0)   ac=3;                   //اگر کلیدی از ستون چهارم فشرده شد متغیر ac=3
   if (!(ac==4)){                       //اگر متغیر ac≠4 باشد نشان دهندهٔ فشردگی کلید است
   table=data_key[(r*4)+ac];            //این فرمول کد اسکی کلید را بر می‌گرداند
   lcd_putchar(table);                  //نمایش کاراکتر برگشتی از آرایه
   while (c1==0){}                      //تست برای فشردگی کلیدهای ستون اول
   while (c2==0){}                      //تست برای فشردگی کلیدهای ستون دوم
   while (c3==0){}                      //تست برای فشردگی کلیدهای ستون سوم
   while (c4==0){}                      //تست برای فشردگی کلیدهای ستون چهارم
   delay_ms(50);                        //تأخیر زمانی به مدت ۵۰ میلی ثانیه
    }
 }
```

```
 }
}
void main(){                                    //تابع اصلی برنامه
 PORTB=0xff;                                    //فعال کردن مقاومت بالا کش داخلی
 DDRB=0x0f;                                     //نیبل پایینی خروجی و نیبل بالایی ورودی
 lcd_init(16);                                  //تعیین ستون LCD و مقدار دهی اولیه
 lcd_gotoxy(0,0);                               //رفتن به سطر اول و ستون اول lcd
 lcd_putsf("test keypad");                      //نمایش رشته کاراکتری ذخیره شده در حافظه Flash
 keypad();                                      //فراخوانی تابع اسکن صفحه کلید
}
```



شکل ۴-۸ نحوه‌ی اتصال صفحه کلید به میکروکنترلر [مثال ۴-۵]

تمرین‌های فصل ۴

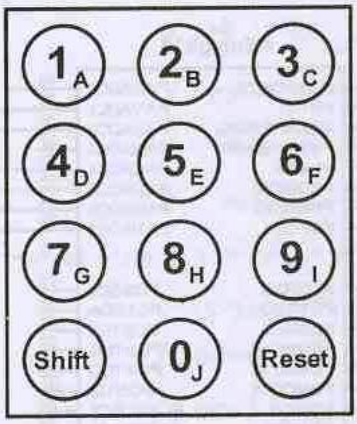
۴-۱ برنامه و سخت‌افزاری طراحی کنید که یک عدد ۸ بیتی علامتدار را از PORTB دریافت نماید و آن را بر روی نمایشگر سون سگمنت چهار رقمی از نوع آند مشترک به روش مالتی پلکسری نمایش دهد و بر روی رقم چهارم نمایشگر علامت را نشان دهد؟

راهنمایی: اگر بیت PB.7 یک باشد علامت منفی و اگر صفر باشد علامت مثبت است بنابراین محدوده عدد خوانده شده از 127+ تا 128- خواهد بود.

۲-۴ برنامه‌ای بنویسید که با زدن یک کلید فشاری، یک LED به مدت ۱۰ ثانیه روشن شود و اگر کلید را در هر لحظه‌ای از زمان فشار دهیم مدت زمان روشن ماندن را به مدت ۱۰ ثانیه تمدید کند به طور نمونه اگر کلید را سه بار فشار دهیم باید ۳۰ ثانیه به زمان سپری شده اضافه گردد و سپس خاموش شود؟

۳-۴ برنامه و سخت‌افزار اسکن صفحه کلید ۴×۴ را به گونه‌ای طراحی نمائید که علاوه بر ۱۶ کلید فشاری متعلق به صفحه کلید، کلید فشاری دیگری بنام Num Lock جهت فعال یا غیر فعال کردن آن داشته باشد. در لحظه Reset صفحه کلید غیر فعال است و یک LED را جهت نمایش وضعیت فعال‌سازی بکار بگیرید و کلیدهای اسکن شده را بر روی LCD نمایش دهید؟

۴-۴ برنامه و سخت‌افزار اسکن صفحه کلید ۴×۳ زیر را به گونه‌ای طرح نمائید که با زدن هر دکمه، کد اسکی عددی هر کلید را بر روی LCD نمایش دهد و در صورتی که کلید Shift را انتخاب کردیم کد اسکی A تا J را نمایش دهد و با زدن مجدد کلید Shift به حالت اول برگردد و با زدن کلید Reset صفحه نمایش را پاک کند؟



۵-۴ برنامه‌ای بنویسید که نام کوچک خود را به صورت فارسی بر روی LCD متنی نمایش دهد؟

۶-۴ در سخت‌افزار مثال ۴-۵، کاتد چهار دیود معمولی را به ستون‌های صفحه کلید متصل کنید و آند آنها را به وقفه خارجی صفر وصل نمایید و حال برنامه آن را به گونه‌ای بنویسید که با فشردن هر کلید، کد آن کلید، توسط اجرای وقفه برگردد؟

۷-۴ برنامه یک شمارنده دو رقمی را بنویسید که با فشردن یک کلید فشاری شمارش آغاز شده و با فشردن همان کلید شمارش متوقف شده و به صفر برگردد. نمایش بر روی سون سگمنت دو رقمی به روش مالتی پلکسری انجام گیرد؟

۸-۴ برنامه یک شمارنده سه رقمی را بنویسید که توسط یک کلید فشاری شمارش آغاز شده و با فشردن همان کلید شمارش متوقف شده و به صفر برگردد. نمایش بر روی LCD انجام گیرد؟