

# اسکن صفحه کلید و کار

## کردن با دات ماتریس

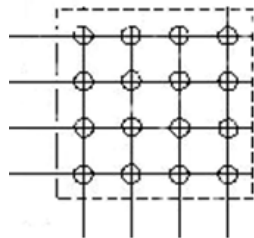
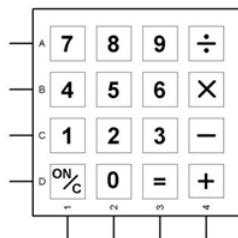
در این قسمت سعی ما بر این است که نحوه اسکن کردن صفحه کلید ها را آموزش داده و چند برنامه کاربردی را نیز به شما یاد بدهیم . همچنین نحوه کار با دات ماتریس یا همان تابلو روان را نیز به شما نشان می دهیم . دلیل آوردن این دو مطلب در یک بخش این است که صفحه کلید و تابلو روان بر اساس ماتریس و کار کردن با سطرها و ستون ها می باشد . این بخش یکی از بخش های کاربردی و مهم برای استفاده و کار کردن با همه میکروکنترلر ها از جمله AVR می باشد .

### صفحه کلید و اسکن کردن آن

صفحه کلید ها اصولا به صورت ماترسی هستند بدین معنی که همه کلید های یک سطر و یک ستون به هم متصل هستند و ساختاری مانند شکل شکل 1\_6 را دارند .

حال اگر سطر A را تغذیه کنیم و کلید 7 را فشار دهیم ، ستون یک به سطر A وصل

شده و دارای تغذیه می شود و اگر کلید 8 را فشار دهیم ستون دو به سطر اول وصل



شده و دارای تغذیه می شود .

شکل 1-5

## ریزپردازنده AVR

پس برای اینکه بدانیم کدام کلید فشرده شده است باید سطرها (یا ستون ها) را یکی یکی تغذیه کرده و تک تک ستون ها (یا سطرها) را چک کنیم تا ببینیم که کدام کلید فشرده شده است .

مثال : با وصل یک صفحه کلید به میکرو یک ماشین حساب ساده درست کنید :

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

#asm // شروع برنامه اسمبلی

.equ __lcd_port=0x1B ;PORTA // معرفی پرت A به عنوان پورت نمایشگر

#endasm // پایان برنامه اسمبلی

#include <lcd.h> // فراخوانی کتابخانه lcd

#include <stdlib.h> // فراخوانی کتابخانه stdlib

#include <delay.h> // فراخوانی کتابخانه تاخیر زمانی

unsigned char s[10],p[],a,w,c,b,x,y,i,z,r; // معرفی متغیرها

float eq ; // معرفی متغیرها

void main(void) { // برنامه اصلی

while (1) { // حلقه بینهایت

PORTC=0xf0; // یک کردن چهار بیت پر ارزش پرت C و صفر کردن چهار بیت کم ارزش

//*****row1*****

PORTC.4=0; // خواندن سطر اول

delay_ms(3); // تاخیر زمانی
```

## اسکن صفحه کلید و کار کردن با دات ماتریس

```
if(PINC.0==0){ a=7;while(!PINC.0);c=1;} // اگر پین صفر برابر صفر شد یعنی عدد هفت وارد شده است
if(PINC.1==0){ a=4;while(!PINC.1);c=1;} // اگر پین یک برابر صفر شد یعنی عدد چهار وارد شده است
if(PINC.2==0){ a=1;while(!PINC.2);c=1;} // اگر پین دو برابر صفر شد یعنی عدد یک وارد شده است
if(PINC.3==0){while(!PINC.3);c=2;} // اگر پین سه برابر صفر شد یعنی کلید reset زده شده است
PORTC.4=1; // یک کردن پین چهار
//*****row2***** // خواندن سطر دوم
PORTC.5=0; // صفر کردن پین پنجم
delay_ms(3); // تاخیر زمانی
if(PINC.0==0) {a=8;while(!PINC.0);c=1;} // اگر پین صفر برابر صفر شد یعنی عدد هشت وارد شده است
if(PINC.1==0) {a=5;while(!PINC.1);c=1;} // اگر پین یک برابر صفر شد یعنی عدد پنج وارد شده است
if(PINC.2==0) {a=2;while(!PINC.2);c=1;} // اگر پین دو برابر صفر شد یعنی عدد دو وارد شده است
if(PINC.2==3) {a=0;while(!PINC.3);c=1;} // اگر پین سه برابر صفر شد یعنی عدد یک وارد شده است
PORTC.5=1;
//*****row3***** // خواندن سطر سوم
PORTC.6=0; // صفر کردن پین شش
delay_ms(3); // تاخیر زمانی
if(PINC.0==0) {a=9;while(!PINC.0);c=1;} // اگر پین صفر برابر صفر شد یعنی عدد نه وارد شده است
if(PINC.1==0) {a=6;while(!PINC.1);c=1;} // اگر پین یک برابر صفر شد یعنی عدد شش وارد شده است
if(PINC.2==0) {a=3;while(!PINC.2);c=1;} // اگر پین دو برابر صفر شد یعنی عدد سه وارد شده است
if(PINC.3==0) { while(!PINC.3);c=3;} // اگر پین سه برابر صفر شد یعنی کلید مساوی زده شده است
```

## ریزپردازنده AVR

```
PORTC.6=1; // یک کردن پین شش
//*****row4***** // خواندن سطر چهارم
PORTC.7=0; // صفر کردن پین هفت
delay_ms(3); // تاخیر زمانی
if(PINC.0==0) {w=1;while(!PINC.0);c=4;} // اگر پین صفر برابر صفر شد یعنی کلید تقسیم زده شده است
if(PINC.1==0) {w=2;while(!PINC.1);c=4;} // اگر پین یک برابر صفر شد یعنی کلید ضرب زده شده است
if(PINC.2==0) {w=3;while(!PINC.2);c=4;} // اگر پین دو برابر صفر شد یعنی کلید تفریق زده شده است
if(PINC.3==0){ w=4;while(!PINC.3);c=4;} // اگر پین سه برابر صفر شد یعنی کلید جمع زده شده است
PORTC.7=1; // یک کردن پین هفت
if(c==1&& i<1) { // اگر کلید یکی از اعداد زده شد
ftoa(a,0,p); // تبدیل عدد وارد شده به رشته
lcd_puts(p); // نمایش عدد وارد شده
b=(b*10)+a; // اگر عدد جدید وارد شد قبلی را در ده ضرب کن و با عدد جدید جمع کن
c=0; } // آماده کردن میکرو برای دریافت عدد بعدی
if(c==2) { // اگر کلید Reset زده شد
lcd_clear(); // پاک کردن نمایشگر
b=0; // صفر کردن متغیر دریافت عدد
i=0; // آماده شدن برای دریافت عدد جدید
w=0; // آماده شدن برای دریافت عدد جدید
```

```
z=0; // آماده شدن برای دریافت عدد جدید
c=0; } // آماده شدن برای دریافت عدد جدید
```

### // برنامه محاسبه و نمایش عدد حاصل

```
if(c==3) { // اگر کلید مساوی زده شد
y=b; // ریختن عدد دوم دریافتی در متغیر جدید
b=0; // صفر کردن متغیر دریافت عدد
if(r==1) { // اگر کلید تقسیم قبلا زده شده است
eq=(float)x/y; // تقسیم عدد اول بر عدد دوم
ftoa(eq,2,s); // حاصل تقسیم را تا دو رقم اعشار تبدیل به رشته کرده و در متغیر S بریز
lcd_putsf(""); // نمایش کاراکتر "="
lcd_puts(s); } // نمایش حاصل تقسیم
if(r==2) { // اگر کلید ضرب قبلا زده شده است
eq=x*y; // ضرب عدد اول در عدد دوم
ftoa(eq,0,s); // حاصل ضرب را رشته کرده و در متغیر S بریز
lcd_putsf(""); // نمایش کاراکتر "="
lcd_puts(s);} // نمایش حاصل ضرب
if(r==3) { // اگر کلید منها قبلا زده شده است
eq=x-y; // کم کردن عدد دوم از عدد اول
ftoa(eq,0,s); // حاصل تفریق را رشته کرده و در متغیر S بریز
```

## ریزپردازنده AVR

```
lcd_putsf(""); // نمایش کاراکتر "="  
lcd_puts(s); // نمایش حاصل تفریق  
if(r==4) { // اگر کلید جمع قبلا زده شده است  
    eq=x+y; // جمع کردن عدد اول با عدد دوم  
    ftoa(eq,0,s); // حاصل جمع را رشته کرده و در متغیر S بریز  
    lcd_putsf(""); // نمایش کاراکتر "="  
    lcd_puts(s); // نمایش حاصل جمع  
    r=0; // آماده شدن برای محاسبات بعدی  
    c=0; // آماده شدن برای محاسبات بعدی  
    i=3; } // آماده شدن برای محاسبات بعدی  
  
// ذخیره عدد اول و انجام عمل محاسباتی که بعدا باید انجام شود  
if(c==4&&z<1) { // اگر کلید یکی از عملگرهای محاسباتی زده شد  
    if(w==1) { // اگر کلید تقسیم زده شد  
        lcd_putsf("%"); // نمایش کاراکتر "%"   
        x=b; // ریختن عدد اول وارد شده در متغیر جدید  
        b=0; } // صفر کردن متغیر دریافت عدد  
        if(w==2) { // اگر کلید ضرب زده شد  
            lcd_putsf("*"); // نمایش کاراکتر "*"   
            x=b; // ریختن عدد اول وارد شده در متغیر جدید
```

## اسکن صفحه کلید و کار کردن با دات ماتریس

```

b=0; } // صفر کردن متغیر دریافت عدد

if(w==3) { // اگر کلید منها زده شد

lcd_putsf("-"); // نمایش کارکتر "-"

x=b; // ریختن عدد اول شده در متغیر جدید

b=0; } // صفر کردن متغیر دریافت عدد

if(w==4) { // اگر کلید جمع زده شد

lcd_putsf("+"); // نمایش کارکتر "+"

x=b; // ریختن عدد اول شده در متغیر جدید

b=0; } // صفر کردن متغیر دریافت عدد

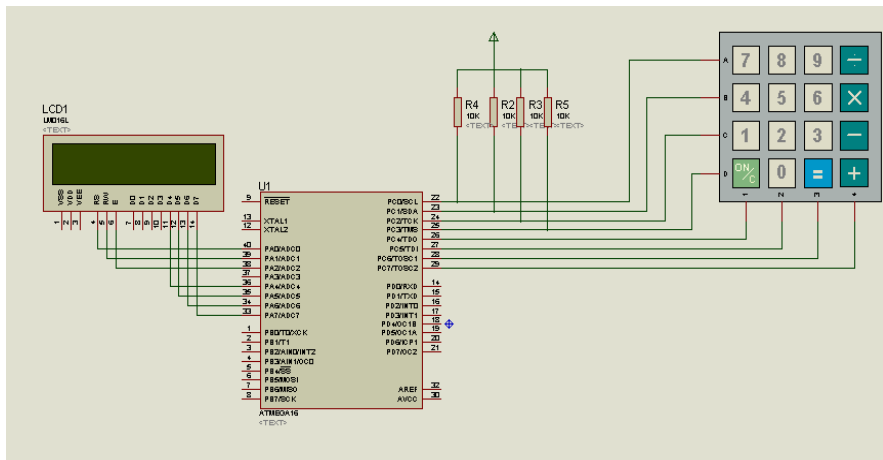
c=0; // آماده شدن برای دریافت عدد بعدی

r=w; // ریختن کد کلید عمل محاسباتی زده شده در یک متغیر کمکی برای انجام عمل محاسبات بعد از زدن کلید مساوی

w=0; // آمادگی برای انجام محاسبات بعدی

z=2; } // reset چند باره عمل محاسباتی تا قبل از زدن کلید
    
```

}}



شکل 5-2

## تابلو روان (LED Dot Matrix)

یکی از نمایشگرهایی که بسیار پر کاربرد می باشد و شما آنها را در بیشتر مراکز خرید، خیابان ها و ... می بینید، تابلو روان یا دات ماتریس می باشد . این نمایشگرها دارای ابعاد مختلف بوده و از تعدادی LED تشکیل شده است . مثلا یک تابلو 8x8 دارای هشت سطر و هشت ستون بوده که از 64 عدد LED تشکیل شده است . برای اینکه بتوانید هر LED را روشن کنید باید به سطر و ستون متناظر آن تغذیه وصل کنید . برای اینکه هر کارکتر یا نوشته ای را بر روی تابلو نمایش دهید باید به صورت زیر عمل کنید :

فرض کنید که صفحه ای دارید که هشت سطر دارد اما دارای بینهایت ستون می باشد، حال کاراکتر یا نوشته خود را بر روی این صفحه بکشید ، اکنون باید کد معادل هر ستون را بدست آورید، بدین صورت که خانه هایی را که پر شده اند را عدد صفر و آنهایی که پر نشده اند را عدد یک فرض کنید (عکس این قضیه نیز صادق است) حالا هر ستون شما دارای یک کد هشت بیتی است ، مثلا اگر بخواهیم یک ستون را کاملا روشن کنیم کد معادل آن 00000000 یا 0x00 می شود . برای نمایش کارکتر هر ستون باید کد بدست آمده را به پایه سطرها داده و ستون مورد نظر را تغذیه کنیم .

مثال : می خواهیم کلمه "علی" را بر روی تابلو روان نمایش داده ، به طوری که از یک طرف وارد و از طرف دیگر خارج شود .

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

#include <delay.h> // فراخوانی کتابخانه تاخیر زمانی

unsigned char
j,c,i,x,a[]={0xff,0x8f,0xbf,0x8f,0xe0,0xe3,0xed,0xed,0xff,0xff,0xff,0x8f,0xbf,0
x8f,0xe0,0xe3,0xed,0xed,0xff,0xff}; // معرفی متغیرها و کد معادل بدست آمده

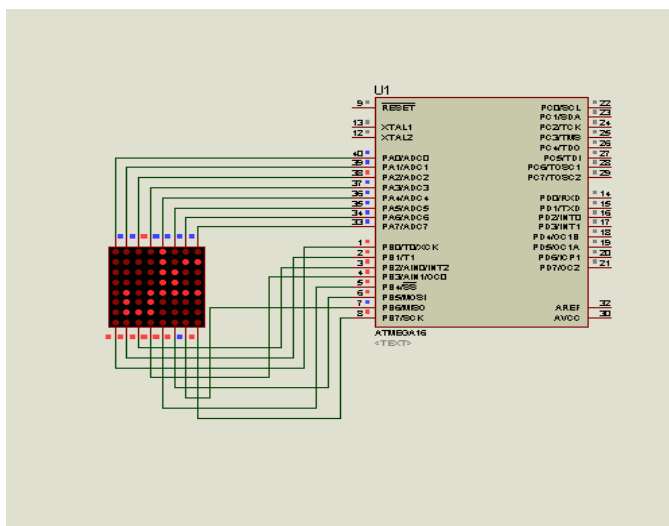
void main(void) { // برنامه اصلی
```



```

while (1) {   c=1; // حلقه بی نهایت
for(j=0;j<20;j++) { // نمایش همه کاراکتر ها
for(i=0;i<50;i++) { // نمایش تکراری یک کاراکتر جهت ماندگاری اثر نور در چشم
for(x=j;x<8+j;x++) { // نمایش همزمان هشت ستون
PORTA=c; // روشن کردن اولین ستون
if(x>=20) { // اگر همه کاراکتر ها نمایش داده شد
PORTA=0xff; } // تمام صفحه خاموش شود
else { // در غیر این صورت
PORTB=a[x]; } // نمایش کاراکتر ستون مورد نظر
delay_us(1500); // تاخیر زمانی
c=c<<1; // روشن کردن تک تک ستون های متناظر کدها
if(c==0) { // اگر آخرین ستون نمایش داده شد
c=1; } } } } // دوباره از ستون اول شروع کن

```



شکل 2-5