



دانشگاه صنعتی اصفهان  
دانشکده برق و کامپیوتر

# دستورکار آزمایشگاه طراحی مدارهای واسط

## آزمایش سوم

تهیه کننده: مهران صفایانی

پاییز ۹۵

# آشنایی با نمایشگرهای ماتریسی نقطه ای، ۱۶ قسمتی و ۷ قسمتی

❖ اهداف :

• آشنایی و راه اندازی نمایشگر ماتریسی نقطه ای  
(*DOTMATRIX*)

• آشنایی و راه اندازی نمایشگر ۱۶ قسمتی

• آشنایی و راه اندازی نمایشگر ۷ قسمتی

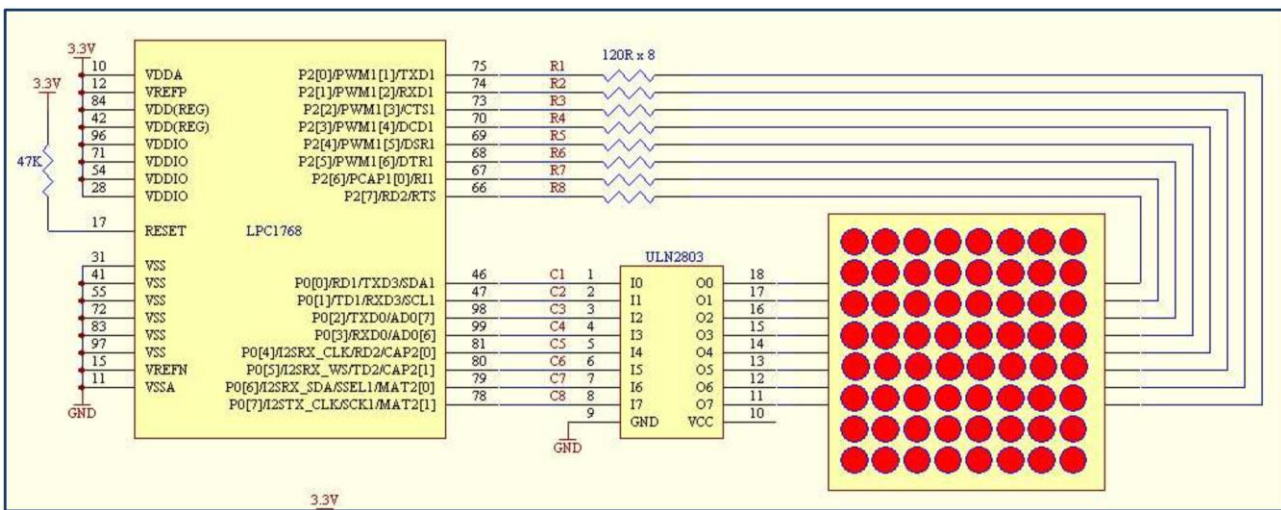
❖ مقدمه :

هدف از انجام این آزمایش آشنایی با نحوه کار کردن نمایشگرهای ماتریسی نقطه ای و نمایشگر ۱۶ و ۷ قسمتی به همراه نمایش اعداد و حروف بر روی آنهاست.

## ❖ نمایشگر ماتریسی نقطه ای (*DOTMATRIX*)

یک عدد نمایشگر *DOTMATRIX* از نوع ۸\*۸ (۸ ستون و ۸ ردیف) در بلوکی به نام *DOTMATRIX Display* قرار داده شده است. با اتصال ماتریسی LED ها به میکرو، تعداد پایه های خروجی را به ۱۶ پین کاهش می دهیم. برای کنترل نمایشگر باید از روش جاروب کردن سطر و ستون استفاده کرد، به این صورت که ابتدا سطر اول را یک (مابقی صفر) میکنیم و سپس مقادیر (خاموش یا روشن بودن هر LED در این سطر) مربوط به ستونها را روی ۸ پین ستون قرار می دهیم. حال در مرحله بعدی سطر دوم را یک (مابقی صفر) میکنیم و سپس مقادیر مربوط به ردیف دوم را بر روی ۸ پین ستون قرار می دهیم و همینطور تا سطر هشتم ادامه می دهیم. به یاد داشته باشید که تمامی این مراحل باید با سرعت بالا انجام پذیرد تا چشم انسان قادر به دیدن تاخیری که بین روشن نمودن سطرها مختلف است نباشد. در این مورد جریان ستون ها توسط آی سی *ULN2803* فراهم شده است و می توانیم سطرها را مستقیماً به GPIO متصل کنیم. در این صورت با جاروب کردن ستون ها می توان یک شکل را در نمایشگر نشان داد.

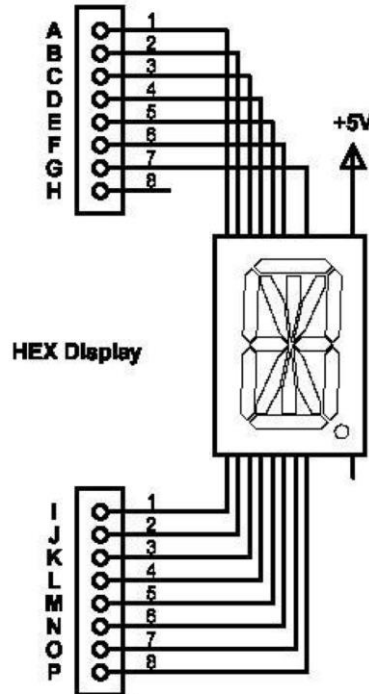
**برنامه نمونه :** در این مثال قصد داریم تا عدد ۲ را بر روی این نمایشگر نمایش دهیم برای این منظور همانطور که در شکل صفحه بعد نمایش داده شده است پین های سطر به هشت پین پایین پورت ۲ و پین های ستون از طریق آی سی به هشت پین پایین پورت صفر متصل شده است.



```
#include <LPC17xx.h>
void delay(int i){
    while(i--);
}
int main(void){
    LPC_GPIO2->FIODIR0 = 0xFF;
    LPC_GPIO0->FIODIR0 = 0xFF;
    while(1){
        LPC_GPIO0->FIOPIN0 = 0x01;
        LPC_GPIO2->FIOPIN0 = 0x00;
        delay(5000);
        LPC_GPIO0->FIOPIN0 = 0x02;
        LPC_GPIO2->FIOPIN0 = 0xc6;
        delay(5000);
        LPC_GPIO0->FIOPIN0 = 0x04;
        LPC_GPIO2->FIOPIN0 = 0xe3;
        delay(5000);
        LPC_GPIO0->FIOPIN0 = 0x08;
        LPC_GPIO2->FIOPIN0 = 0xb1;
        delay(5000);
        LPC_GPIO0->FIOPIN0 = 0x10;
        LPC_GPIO2->FIOPIN0 = 0xb1;
        delay(5000);
        LPC_GPIO0->FIOPIN0 = 0x20;
        LPC_GPIO2->FIOPIN0 = 0x9b;
        delay(5000);
        LPC_GPIO0->FIOPIN0 = 0x40;
        LPC_GPIO2->FIOPIN0 = 0x8e;
        delay(5000);
        LPC_GPIO0->FIOPIN0 = 0x80;
        LPC_GPIO2->FIOPIN0 = 0x00;
        delay(5000);
    }
}
```

## ❖ نمایشگر ۱۶ قسمتی :

این نمایشگر که از ۱۶ قسمت LED کوچک تشکیل شده است قادر به نمایش حروف لاتین و اعداد به صورت مجزا می باشد . نمایشگر مورد استفاده در برد از نوع کاتد مشترک است .

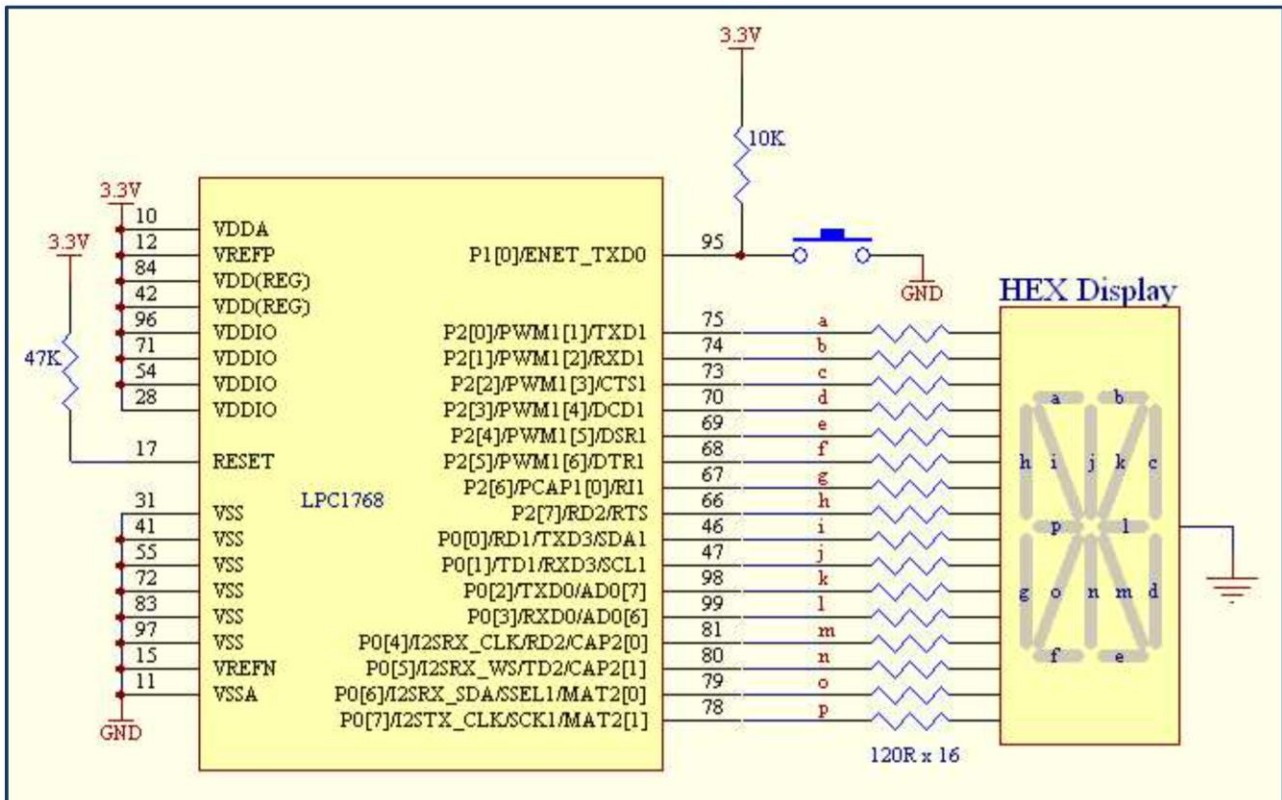


**برنامه نمونه :** در این برنامه قصد داریم در صورت ۱ بودن PORT1.0 حرف A و در صورت صفر بودن PORT1.0 عدد ۹ نمایش داده شود  
برای نمایش مقدار A باید مقادیر زیر را بر روی PORT1 و PORT2 جایگذاری کنیم .

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
1	1	1	1	0	0	1	1	0	0	0	1	0	0	0	1

همچنین برای نمایش مقدار ۹ باید مقادیر زیر را بر روی PORT1 و PORT2 جایگذاری کنیم .

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
1	1	1	1	1	1	0	1	0	0	0	1	0	0	0	1



```
#include "LPC17xx.h"

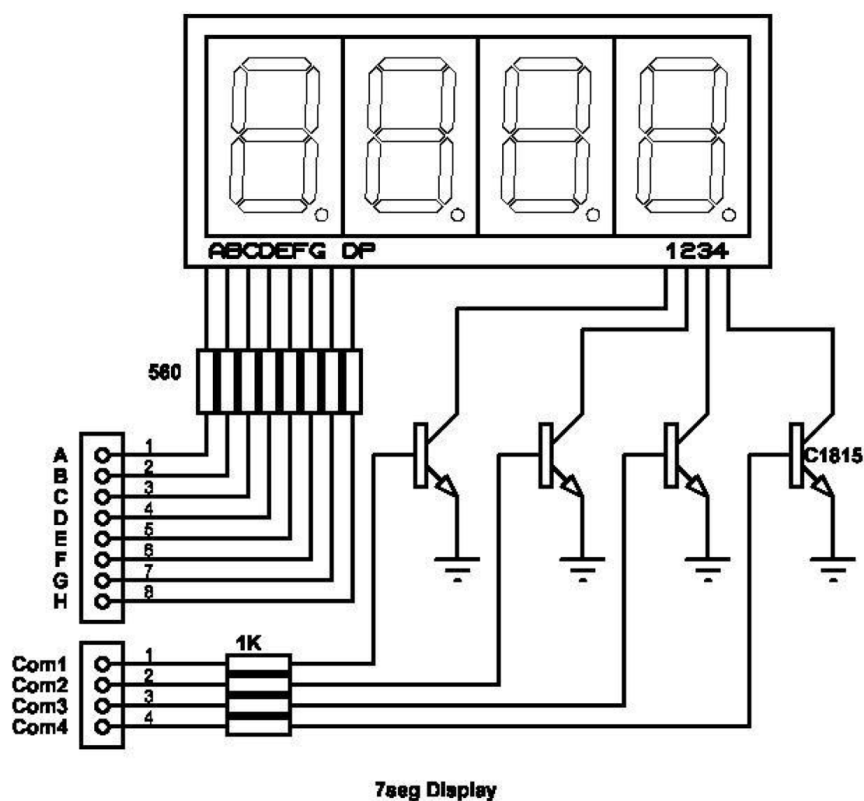
void delay (int i){
    while(i--);
}

int main(void) {
    LPC_GPIO2->FIODIR = 0xFFFFFFFF;
    LPC_GPIO0->FIODIR = 0xFFFFFFFF;
    LPC_GPIO1->FIODIR = 0x00000000;
    while(1){
        if((LPC_GPIO1->FIOPIN & (1<<0))==0){
            LPC_GPIO2->FIOPIN = 0xBF;
            LPC_GPIO0->FIOPIN = 0x88;
            delay(400000);
        }
        else {
            LPC_GPIO2->FIOPIN = 0xCF;
            LPC_GPIO0->FIOPIN = 0x88;
            delay(400000);
        }
    }
}
```

## ❖ نمایشگر 7Segment:

در برورد آموزشی که در اختیار دارید ۴ عدد 7segment مالتی پلکس شده (منظور از مالتی پلکس شدن، موازی بودن خطوط DATA میباشد) به رنگ قرمز از نوع کاتدمشترک تحت بلوکی به عنوان 7segment Display قرارداده شده است.

همانطور که در شکل مشخص شده است با اعمال یک منطقی به خطوط com، ترانزیستور متناظر در ناحیه یاشباع قرار گرفته و پایه متناظر 7segment را به زمین سویچ می نماید. به این ترتیب با اعمال کدهای مربوط به هر عدد می توان اعداد دلخواه را بر روی هر کدام از 7segment ها نمایش داد.



**برنامه نمونه :** در برنامه زیر قصد داریم تا عدد ۱۵ را با نمایشگر 7Segment نمایش دهیم.

```
#include "lpc17xx.h"
unsigned char digit[10] =
{0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0xF};

void Delay (unsigned int Time){
    unsigned int i = 0;
    while (Time--){
        for (i = 0 ; i<5000 ; i++);
    }
}
```

```

int main(){
    LPC_GPIO0->FIODIR = 0xFFFFFFFF;
    LPC_GPIO2->FIODIR = 0xFFFFFFFF;
    while(1)
    {
        LPC_GPIO0->FIOPIN = 1;
        LPC_GPIO2->FIOPIN = digit[5];
        Delay(10);
        LPC_GPIO0->FIOPIN = 2;
        LPC_GPIO2->FIOPIN = digit[1];
        Delay(10);
    }
}

```

---

**برنامه نمونه دوم :** در برنامه زیر قصد داریم تا بر روی نمایشگر 7segment یک شمارنده ایجاد کنیم که قابلیت شمارش از عدد ۱ تا ۹۹ را داشته باشد .

```

#include "lpc17xx.h"
unsigned char digit[10] =
{0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0xF};

void Delay (unsigned int Time)
{
    unsigned int i = 0;
    while (Time--){
        for (i = 0 ; i<5000 ; i++);
    }
}

int i,j,y;
int main(){
    LPC_GPIO0->FIODIR = 0xFFFFFFFF;
    LPC_GPIO2->FIODIR = 0xFFFFFFFF;
    while(1){
        Delay(250);
        for(i=0;i<=9;i++){
            for(j=0;j<=9;j++){
                for(y=0;y<=100;y++){
                    LPC_GPIO0->FIOPIN = 1;
                    LPC_GPIO2->FIOPIN = digit[j];
                    Delay(10);
                }
            }
        }
    }
}

```

- I. برنامه ای بنویسید که از ۱ تا ۹۹۹ را به صورت صعودی و نزولی بشمارد.
- II. برنامه ای بنویسید که با فشار دادن یک کلید به مقدار فعلی 7Segment یک واحد اضافه شود و با فشردن کلید دیگر یک واحد از مقدار کم شود. بازه اعداد از صفر تا ۹۹ است.
- III. حرف اول اسم خود را بر روی DotMatirx نمایش دهید.
- IV. حروف اسم خود را با تاخیر ۱ ثانیه بر روی نمایشگر ۱۶ قسمتی نمایش دهید.