



دانشگاه صنعتی اصفهان  
دانشکده برق و کامپیوتر

## دستورکار آزمایشگاه طراحی مدارهای واسط

### آزمایش دوم

تهیه کننده: مهران صفایانی

پاییز ۹۵

# آشنایی با پورت های ورودی / خروجی و موتورهای پله ای

## ❖اهداف :

- آشنایی با پورت های ورودی و خروجی *GPIO* و رجیستر های آن
- کار با کلید های فشاری و *LED*
- کار با موتور پله ای (*Stepermotor*)

## ❖مقدمه:

هدف از این آزمایش آشنایی با ورودی خروجی های دیجیتال ، بررسی رجیسترهای مربوطه و همچنین کار با این ورودی و خروجی به صورت عملی است. بر این اساس شما داده هایی را از طریق کلیدهای فشاری به درون میکروکنترلر منتقل کرده و بروی *LED* های موجود در برد اطلاعاتی را نمایش می دهید همچنین با نحوه کار با موتور پله ای آشنا خواهید شد.

## ❖بررسی رجیستر های *GPIO*:

جهت کار با پورت های ورودی خروجی از پنج رجیستر استفاده می شود که در ادامه به بررسی و کار با این رجیسترها می پردازیم. در میکرو *LPC1768* می توان از پورت *p0* پین های [30:0] ، پورت *p1* پین های [31:0] ، پورت *p2* پین های [13:0] ، پورت *p3* پین های [26:25] و پورت *p4* پین های [29:28] به عنوان ورودی و خروجی استفاده کرد.

**رجیستر اول *LPC\_GPIOx -> FIODIR* :** این رجیستر برای تعیین جهت پورت استفاده می شود به نحویکه با نوشتن مقدار یک در این رجیستر پایه مربوط به عنوان خروجی و با نوشتن مقدار صفر پایه به عنوان ورودی تعیین می شود. پورتهای میکرو *LPC1768* به طور پیش فرض در جهت ورودی می باشند و تنها برای خروجی کردن پورت نیاز به مقدار دهی داریم . به جای *x* در این رجیستر شماره پورت قرار می گیرد .  
مثال : بیت اول رجیستر پورت را در جهت خروجی تعیین کنید.

```
FIODIR = 0X01 -> LPC_GPIO2;
```

مثال ۸: بیت اول رجیستر پورت صفر را در جهت خروجی تعیین کنید.

```
LPC_GPIO0 -> FIODIR = 0XFF;
```

**رجیستر دوم *LPC\_GPIOx -> FIOSET* :** این رجیستر جهت یک کردن پایه خروجی به کار می رود..

مثال ۸ : LED متصل به بیت کم ارزش پورت ۲ را روشن کنید.

```
LPC_GPIO2 -> FIODIR = 0xFF;
LPC_GPIO2 -> FIOSET = 0xFF;
```

رجیستر سوم **LPC\_GPIOx -> FIOCLR** : این رجیستر جهت صفر کردن پایه خروجی به کار می رود..

مثال ۸ : LED متصل به ۸ بیت کم ارزش پورت ۲ را خاموش کنید.

```
LPC_GPIO2 -> FIODIR = 0xFF;
LPC_GPIO2 -> FIOSET = 0x00;
```

---

برنامه ای بنویسید که ۸ LED متصل به ۸ بیت کم ارزش پورت ۲ به شکل چشمک زن خاموش و روشن شوند.

```
#include <lpc17xx.h>

void delay (uint32_t Time)
{
    uint32_t i;
    i = 0;
    while (Time-->0) {
        for (i = 0; i<5000 ; i++);
    }
}

int main(void)
{
    LPC_GPIO2->FIODIR = 0xff;
    LPC_GPIO2->FIOCLR = 0xff;    //turn off leds
    while(1)
    {
        LPC_GPIO2->FIOSET = 0xFF;
        delay(500);
        LPC_GPIO2->FIOCLR = 0xFF;
        delay(500);
    }
}
```

**رجیستر چهارم LPC\_GPIOx -> FIOPIN :** از این رجیستر برای خواندن از ورودی استفاده می شود . البته با این رجیستر می توان پورت ها را مقدار دهی کرد.

---

برنامه ای بنویسید که در صورتیکه کلید متصل به بیت اول پورت ۲ زده شد LED متصل به بیت صفرم پورت ۲ را روشن کند. نکته مهم در برنامه این است که پایه های ورودی میکرو *lpc1768* به طور پیش فرض *pull up* هستند و برای اینکه تنها بیت اول پورت ۲ مد نظر است باید آن را با یک AND کرد.

```
#include <lpc17xx.h>

void delay (uint32_t Time)
{
    uint32_t i;
    i = 0;
    while (Time--) {
        for (i = 0 ; i<5000 ; i++);
    }
}

int main (void)
{
    LPC_GPIO2->FIODIR = 0x1;
    LPC_GPIO2->FIOCLR = 0x1;
    while(1)
    {
        LPC_GPIO2->FIOCLR = 0x1;
        while((LPC_GPIO2->FIOPIN & 1 << 1) == 0)
        {
            LPC_GPIO2->FIOPIN = 0x1;
        }
    }
}
```

---

**رجیستر پنجم LPC\_GPIOx -> FIOMASK :** این رجیستر ها برای انتخاب پینهایی که می خواهید و یا نمی خواهید از رجیسترهای *FIOxCLR* و *FIOxSET* و *FIOxPIN* تأثیر بپذیرند مورد استفاده قرار می گیرند. عملکرد این رجیستر را با ذکر مثالی بیان می کنیم. فرض کنید می خواهیم از پین های شماره *PORT1.16* , *PORT1.17* , *PORT1.18* , *PORT1.19* به عنوان خروجی استفاده کنیم. در صورتی که بقیه بیت ها برای کار دیگری در نظر گرفته شده است، برای اینکه خیالتان از بابت تغییر بقیه بیت ها راحت باشد می توانید با استفاده از رجیستر *FIOxMASK* دسترسی به بیت های مورد نظرتان را فعال و بقیه را غیرفعال کنید. با قرار دادن صفر در هر یک از بیت های رجیستر *FIOxMASK* بیت های مورد نظرتان فعال (یعنی در دسترس) و با قرار دادن یک غیر فعال می شوند . در مثال زیر پین های *PORT1.16* , *PORT1.17*

```
LPC_GPIO1->FIODIR = 0x000F0000;
LPC_GPIO1->FIOMASK = 0xFFFF0FFF;
LPC_GPIO1->FIOSET = 0x548319FC;
LPC_GPIO1->FIOCLR = 0xF2D398FF;
```

موتور های پله ای گونه ای از موتور ها هستند که حرکت شافت رتور آنها به صورت پله ای و تحت زاویه خاص انجام می شود . عموماً این نوع موتور ها دارای ۶ سیم می باشند که دو سیم آن به صورت مشترک و ۴ سیم دیگر به سیم پیچ های استاتور موتور متصل است . اگر به صورت متوالی با رعایت فاصله زمانی پالس اعمال نماییم ، موتور در جهت راست و یا چپ طبق جدول زیر می چرخد.

حال اگر جهت پالس ها را عوض کنیم جهت چرخش موتور عوض خواهد شد .

با اتصال سیم های مشترک به قطب منفی و اتصال ۴ سیم استاتور به میکرو کنترلر میتوانیم موتور را کنترل نماییم .  
**نکته مهم :** به دلیل عدم جریان دهی کافی پورت های میکرو کنترلر ، باید بین میکرو کنترلر و موتور از یک مدار واسطه به عنوان تامین کننده جریان استفاده نماییم (در اینجا از درایو *ULN2803* استفاده شده است .)



**برنامه اول:** برنامه ای بنویسید که موتور پله در جهت خلاف جهت عقربه ساعت شروع به چرخش نماید.

```
#include <lpc17xx.h>

void delay (uint32_t Time)
{
    uint32_t i;
    i = 0;
    while (Time--){
        for (i = 0 ; i<5000 ; i++);
    }
}

int main(void)
{
    LPC_GPIO2->FIODIR0 = 0x0F;          //turn off
    while(1)
    {
        LPC_GPIO2->FIOPIN = 0x01;
        delay(100);
        LPC_GPIO2->FIOPIN = 0x02;
        delay(100);
        LPC_GPIO2->FIOPIN = 0x04;
        delay(100);
        LPC_GPIO2->FIOPIN = 0x08;
        delay(100);
    }
}
```

---

**برنامه دوم:** برنامه ای بنویسید که موتور پله ای به اندازه ۱۵۰ درجه در جهت ساعتگرد بچرخد.

```
#include <lpc17xx.h>
unsigned k = 0;
void delay (uint32_t Time)
{
    uint32_t i;
    i = 0;
    while (Time--){
        for (i = 0 ; i<5000 ; i++);
    }
}
```

```

int main(void)
{
    LPC_GPIO2->FIODIR0 = 0x0F;
    for(k = 0 ; k<3 ; k++)
    {
        LPC_GPIO2->FIOPIN = 0x08;
        delay(10000);
        LPC_GPIO2->FIOPIN = 0x04;
        delay(10000);
        LPC_GPIO2->FIOPIN = 0x02;
        delay(10000);
        LPC_GPIO2->FIOPIN = 0x01;
        delay(10000);
    }
}

```

---

## ❖ دستورکار:

- I. هشت بیت کم ارزش پورت ۲ را به ۸ LED متصل کنید سپس برنامه ای بنویسید که LED های آن یک در میان روشن شود.
- II. برنامه ای بنویسید که LED ها را از بیت صفر تا ۷ به ترتیب روشن شود.
- III. هشت بیت اول پورت صفر را به کلید وصل کنید و سپس ۸ بیت دوم پورت صفر را به ۸ LED متصل کنید و برنامه ای بنویسید که متناظر با هر کلید ، LED مربوطه به آن روشن شود.
- IV. برنامه ای بنویسید که موتور پله ای یک دور راستگرد و دو دور جهت چپگرد بچرخد.
- V. برنامه ای بنویسید که موتور پله ای به اندازه ۱۸۰ درجه در جهت ساعتگرد به چرخد.