

دانشگاهصنعتیاصفهان

دانشکدهبرقوکامپیوتر

**دستورکار آزمایشگاه طراحی مدارهای واسط**

**آزمایش دوم**

تهیه کننده: مهران صفایانی

پاییز95

آشنایی با پورت های ورودی / خروجی و موتورهای پله ای

* اهداف :
  + آشنایی با پورت های ورودی وخروجی *GPIO*ورجیستر 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 بورد اطلاعاتی را نمایش می دهید همچنین با نحوه کار با موتور پله ای آشنا خواهید شد.

* **بررسی رجستر 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 پورتp0 پین های [30:0] ، پورت p1پین های [31:0] ، پورت p2 پین های [13:0] ، پورت p3پین 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 تعین 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.

LPC\_GPIO2; -> FIODIR = 0X01

**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 ۲را روشن کنید.

LPC\_GPIO2 -> FIODIR = 0XFF;

LPC\_GPIO2 -> FIOSET = 0XFF;

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

**مثال :** ۸ LED متصل به ۸ بیت کم ارزش پورت ۲را 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--) {

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 شدLED متصل به بیت صفرم پورت ۲را 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** اين رجيستر ها براي انتخاب پينهايي كه

مي خواهيد و يا نمي خواهيـد از رجيسـترهاي *FIOxCLR* و *FIOxSET* و *FIOxPIN* تأثير بپذيرند مورد استفاده قرار مي گيرند. عملكرد اين رجيستر را با ذكر مثالي بيان مي كنيم .فرض كنيد مي خواهيم از پـين هاي شماره *PORT1.16 , PORT1.17 , PORT1.18 , PORT1.19* به عنوان خروجي استفاده كنيم. در صورتي كه بقيـه بيت ها براي كار ديگري در نظر گرفته شده است، براي اينكه خيالتان از بابت تغيير بقيه بيت هـا راحـت باشـد مـي توانيـد بـا استفاده از رجيستر *FIOxMASK* دسترسي به بيت هاي مورد نظرتان را فعال و بقيه را غيرفعال كنيد. با قرار دادن صفر در هر يك از بيت هاي رجيستر *FIOxMASK* بيتهاي مورد نظرتان فعال (يعني در دسترس) و با قرار دادن يك غير فعال مي شوند . در مثال زير پين هاي *PORT1.16 , PORT1.17 , PORT1.18 PORT1.19* در ابتدا به صـورت خروجـي پيكـره بنـدي شده است و فقط دسترسي به همين 4 پين فعال شده است. حال اگر پين هاي ديگر را بخواهيم صفر و يك نماييم بدليل عدم دسترسي به آنها اين كار انجام نخواهد شد.

LPC\_GPIO1->FIODIR = 0x000F0000;

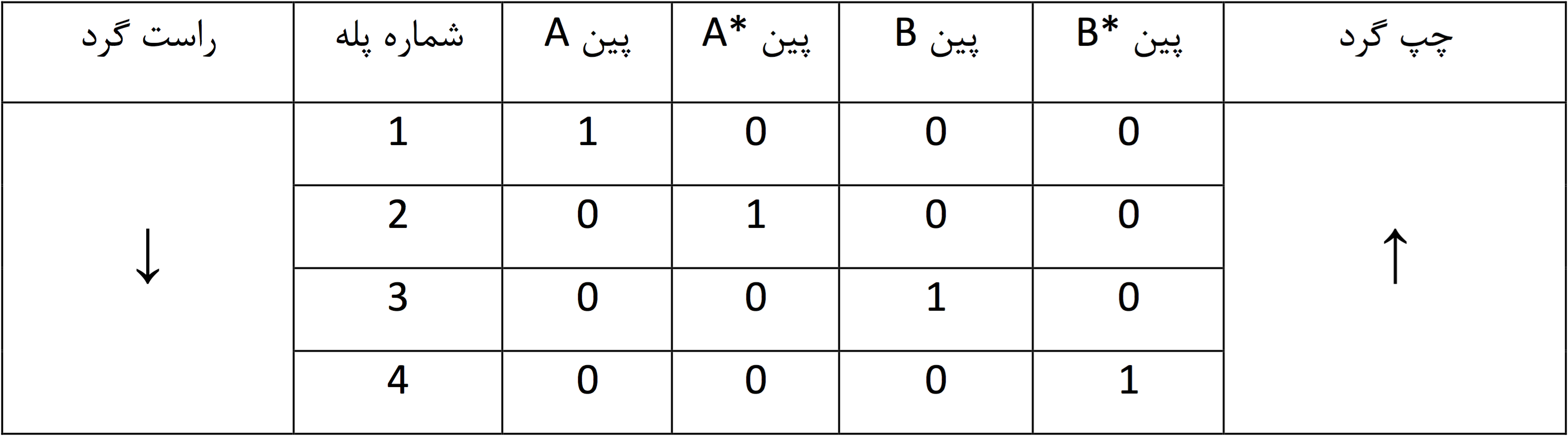
LPC\_GPIO1->FIOMASK = 0xFFF0FFFF;

LPC\_GPIO1->FIOSET = 0x548319FC;

LPC\_GPIO1->FIOCLR = 0xF2D398FF;

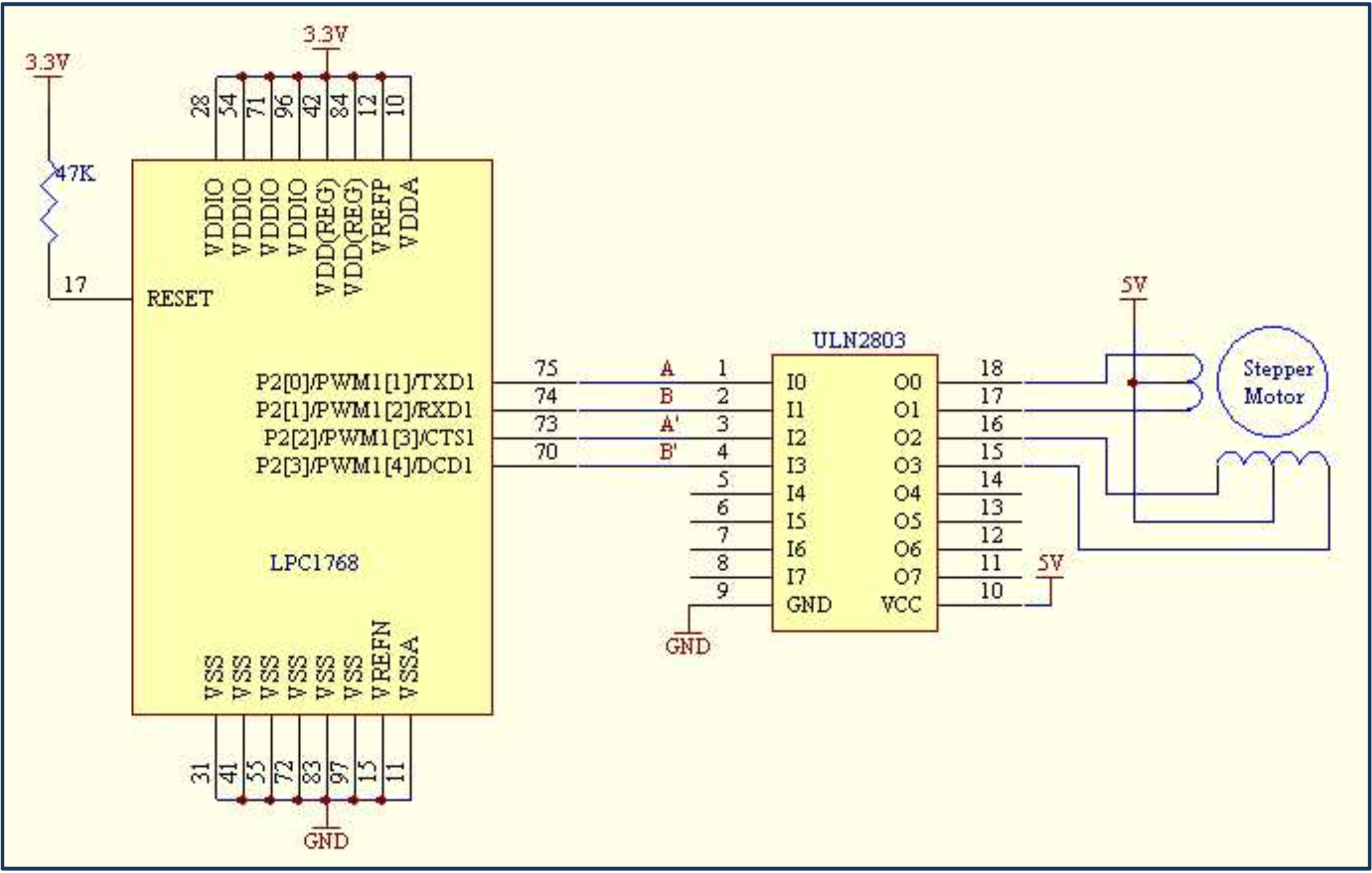
* **راه اندازی موتور پله ای با GPIO :**

موتور های پله ای گونه ای از مو تور ها هستند که حرکت شافت رتورآنها به صورت پله ای و تحت زاویه خاص انجام می شود . عموماً این نوع موتورها 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 میکروکنترلر و موتور از یک مدار واسط به عنوان تامین کننده جریان استفاده نماییم(دراینجا از درایو *ULN2803* استفاده شده است .)



موتور مورد استفاده در این بورد دارای زاویه چرخش ۱۵ درجه می باشد.

**برنامه اول** : برنامه ای بنوسید 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 بنوسید 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++);

}

}

* **دستورکار:**

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);

}

}

1. 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.
2. برنامه ای بنوسید که LED ها را از بیت صفر تا ۷ به ترتیب روشن شود.
3. هشت بیت اول پورت صفررا به کلید وصل کنید و سپس ۸ بیت دوم پورت صفر را به ۸ LED متصل کنید و برنامه ای بنویسید که متناظر با هرکلید ، LED مربوطه به آن روشن شود.
4. برنامه ای بنوسید که موتور پله ای یک دور راستگرد و دو دور جهت چپگرد بچرخد.
5. برنامه ای بنوسید که موتور پله ای به اندازه ۱۸۰ درجه در جهت ساعتگرد به چرخد.