



عنوان:

راه اندازی موتورهای DC و پله ای توسط AVR

به همراه درایورهای هر یک

نویسنده:

حمید بادامی نجات

کلمات کلیدی:

موتور- موتور DC- موتور پله ای- درایور- L293- L298- ULN2002- ULN2003- AVR

چکیده:

در این مقاله شما نحوه راه اندازی موتورهای DC و پله ای توسط میکروکنترلر AVR و همچنین نحوه اتصال هریک از موتورها به درایورهای مربوطه را مطالعه خواهید کرد.



فهرست مطالب

3	موتورهای DC
3	کنترل جهت موتور DC
4	کنترل موتورهای DC با استفاده از ترانزیستور
5	بررسی IC های درایو موتور DC
5	درایور L298
7	درایور L293
9	کنترل سرعت در موتورهای DC
11	تولید پالس PWM از طریق میکروکنترلر
11	تولید PWM از طریق تایمر صفر
12	مثال 1
14	تولید PWM بدون استفاده از تایمر
14	مثال 2
15	مثال 3
17	موتورهای پله ای
18	زاویه پله
19	پله در ثانیه و RPM
20	راه اندازه موتور به صورت نیم پله
20	مدارهای درایور موتور پله ای
20	راه اندازی موتور پله ای با استفاده از ترانزیستور و MOSFET
21	آی سی های درایور موتور پله ای
21	درایور ULN2002 و ULN2003
24	مثال

موتورهای DC:

موتورهای DC دارای مدارات درایو گوناگون و الگوریتم های مختلفی برای کنترل می باشند و هدف از این قسمت نشان دادن چگونگی استفاده از این درایورها برای کنترل موتورهای DC از طریق میکروکنترلر می باشد. از مزایای موتورهای DC نسبت به سایر موتورها می توان به ارزان بودن و ساده بودن مدارات راه انداز آنها اشاره کرد.

یک موتور DC از مجموعه ای از سیم پیچ ها و آهن رباها تشکیل شده است که با وصل کردن ولتاژ به سیم پیچ ها موتور به حرکت در می آید.

موتورهای DC دو ویژگی بسیار مهم دارند:

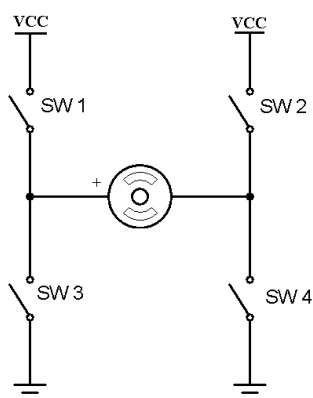
✓ سرعت موتور به وسیله ولتاژ اعمالی به دو سر آن تعیین می شود.

✓ کشتاور موتور به وسیله جریانی که از باتری می کشد تعیین می شود.

به عبارت دیگر اگر ولتاژ بیشتری به موتور اعمال شود سرعت چرخش موتور افزایش می یابد. موتورهای DC کوچک را می توان با ولتاژهای نامی در محدوده ۱/۵ ولت تا ۵۰ ولت تهیه کرد. بر روی موتورها ولتاژ نامی موتور نوشته می شود که موتور در این ولتاژ با حداکثر سرعت خود حرکت می کند. مقدار جریانی عبوری از موتور به مقدار بار بستگی دارد.

کنترل جهت موتور DC:

مهمترین ویژگی موتورهای DC این است که جهت چرخش آنها با تغییر جهت جریان عبوری از موتور تغییر می کند. به عبارتی اگر پلاریته ولتاژ اعمالی به موتور تغییر کند جهت چرخش موتور نیز تغییر پیدا می کند.



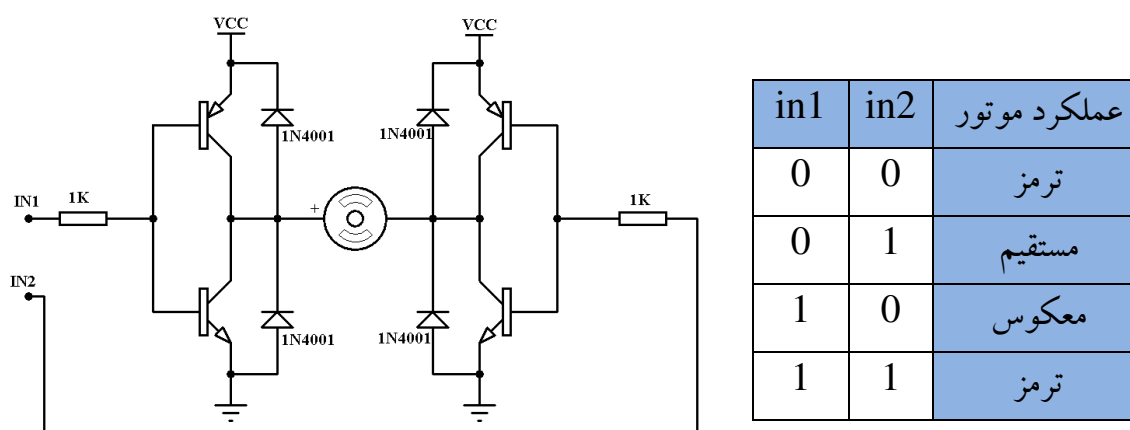
SW1	SW2	SW3	SW4	عملکرد موتور
1	0	0	1	راستگرد
0	1	1	0	چپگرد
0	0	1	1	ترمز
1	1	0	0	ترمز

شکل ۶-۱ کنترل جهت موتور DC

همانطوری که در شکل ۶-۱ مشاهده می کنید اگر پلاریته ولتاژ دو سر موتور تعویض شود جهت موتور تغییر می کند. در ضمن برای ترمز کردن موتور در هنگام حرکت کافی است که دو سر موتور اتصال کوتاه شود.

کنترل موتورهای DC با استفاده از ترانزیستور:

مدار شکل زیر یک مدار پل H است که از چهار ترانزیستور برای راه اندازی، کنترل جهت، و ترمز کردن موتور استفاده شده است.



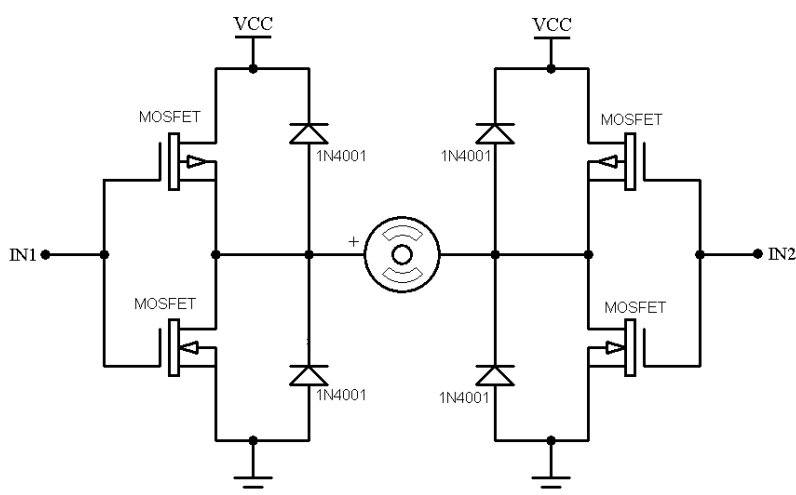
شکل ۶-۲ راه انداز موتور DC با استفاده از ترانزیستور

دیودهای استفاده شده بر روی ترانزیستورها دیودهای هرزگرد می باشند که برای محافظت ترانزیستور در برابر خاصیت خود القایی موتور بکار می روند.

همانطور که در شکل ۶-۲ ملاحظه کردید برای کنترل موتور از چهار ترانزیستور BJT استفاده شده است که برای جریان دهی بیشتر می توان از ترانزیستورهای BJT زوج دالینگتون استفاده کرد. اما روش دیگری نیز برای کنترل موتورهای DC وجود دارد و آن استفاده از ترانزیستورهای MOSFET به جای ترانزیستورهای BJT می باشد.

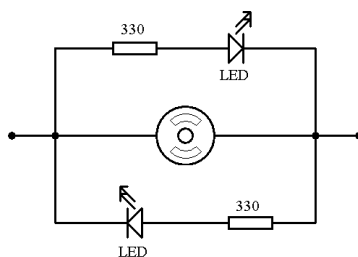
مهمترین تفاوت بین MOSFET و ترانزیستور در ولتاژ کاری و جریان عبوری از آنهاست. به صورت کلی MOSFET ها توانایی عبور جریان بیشتری را در مقایسه با BJT ها دارند.

شکل ۶-۳ یک مدار پل H با استفاده از چهار MOSFET را نمایش می دهد.



شکل ۳-۶ راه اندازی موتور DC با استفاده از MOSFET

برای نشان دادن جهت چرخش موتور می توان دو عدد LED را به صورت عکس هم در دو سر موتور به صورت موازی قرار داد. شکل ۴-۶ نحوه نمایش جهت موتور را با استفاده از دو عدد LED نشان می دهد.



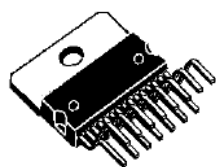
شکل ۴-۶ نمایش جهت موتور با استفاده از LED

بررسی IC های درایو موتور DC :

برای کنترل موتورهای آی سی های مختلفی به بازار عرضه شده است که می توان به درایورهای L298N و L293D و SN154410 و LMD18201 و ... اشاره کرد.

درایور L298 :

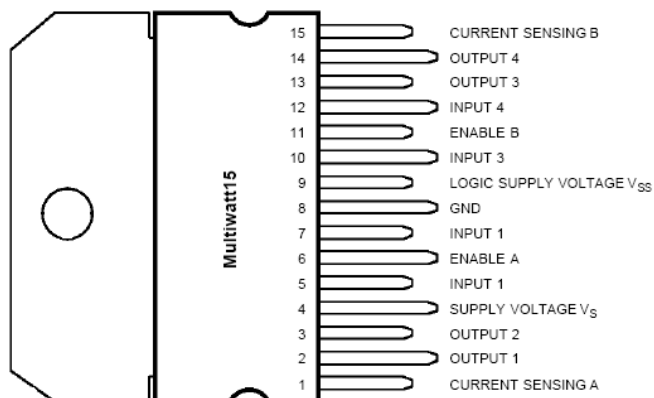
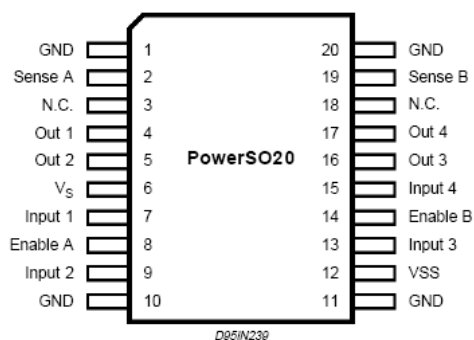
این درایور برای راه اندازی دو موتور بکار می رود. این درایور در دو نوع بسته بندی Multiwatt15 و PowerSo20 به بازار عرضه شده است. شکل ۵-۶ شمای ظاهری و ترتیب پایه های این درایور را نشان می دهد.



Multiwatt15



PowerSO20



شکل ۵-۶ پایه های L298

پایه	عملکرد
1 و 15	این پایه ها باید به وسیله یک مقاومت به زمین متصل شوند.
2 و 3	خروجی های موتور A
4	ولتاژ تغذیه موتور می باشد که باید به وسیله یک خازن 100 nf به زمین متصل شود.
5 و 7	ورودی های موتور A
6 و 11	پایه های فعال سازی برای خروجی های A و B
8	زمین
9	ولتاژ +۷ ولت برای تغذیه است که باید به وسیله یک خازن 100nf به زمین متصل شود.
10 و 12	ورودی های موتور B
16 و 17	خروجی های موتور B

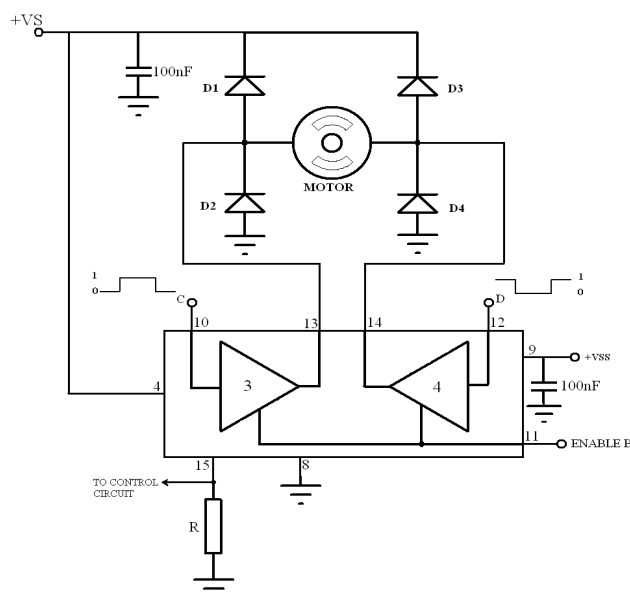
جدول ۱-۶ عملکرد پایه های درایور L298

جدول ۲-۶ مشخصات این درایور را مشخص می کند.

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_I, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_O	Peak Output Current (each Channel) - Non Repetitive ($t = 100\mu s$) - Repetitive (80% on -20% off; $t_{on} = 10ms$) - DC Operation	3 2.5 2	A A A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation ($T_{case} = 75^\circ C$)	25	W
T_{stg}, T_j	Storage and Junction Temperature	-40 to 150	$^\circ C$

جدول ۶-۲ مشخصات درایور L298

شکل ۶-۶ نحوه اتصال یک موتور را از طریق درایور L298 را نشان می دهد.

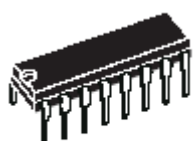


ورودی			عملکرد
Eable	C	D	
1	0	0	ترمز
	0	1	راستگرد
	1	0	چپگرد
	1	1	ترمز
0	C=X, D=X		چرخ ها آزاد

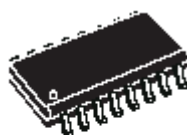
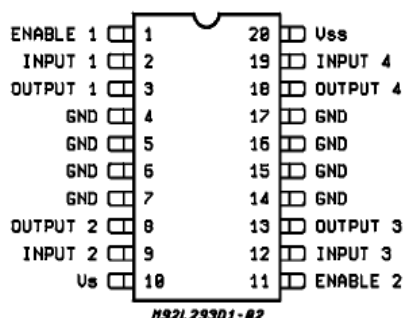
شکل ۶-۶ نحوه اتصال موتور به L298

درایور L293:

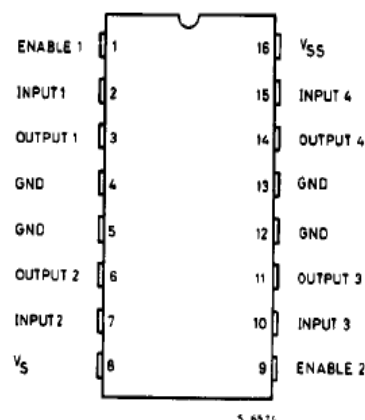
آی سی L293 نیز همانند آی سی L298 یک مدار راه انداز برای دو موتور می باشد. درایور L293D می تواند جریان ۰/۶ آمپر را به صورت پیوسته و ۱/۲ آمپر را به صورت لحظه ای از خود عبور دهد و درایور L298N نیز قادر به عبور جریان ۲A به صورت پیوسته و ۴ آمپر به صورت لحظه ای می باشد. شکل ۶-۷ شمای کلی و ترتیب پایه های درایور L293D را نشان می دهد.



DIP-16

SO-16
(Narrow)

SO(12+4+4)



Powerdip(12+2+2)

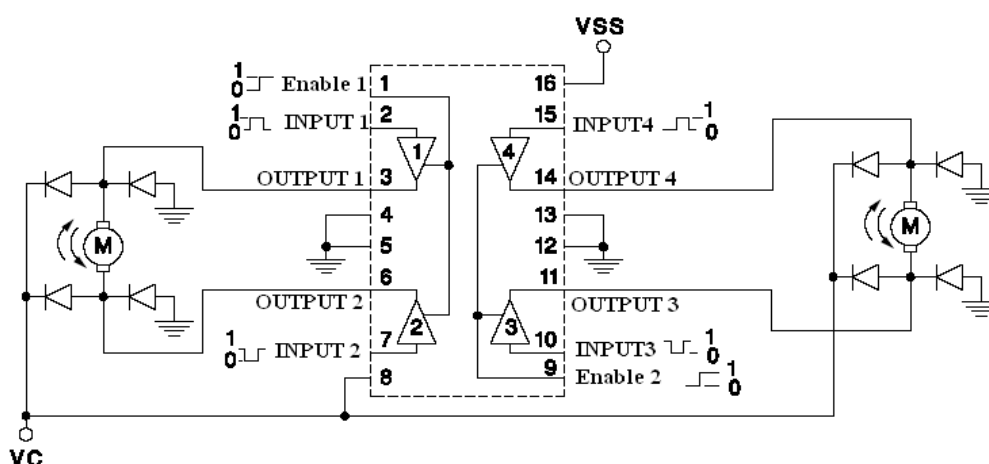
شکل 6-7 پایه های درایور L293D

جدول ۳-۶ مشخصات این درایور را مشخص می کند:

Symbol	Parameter	Value	Unit
V_S	Supply Voltage	36	V
V_{SS}	Logic Supply Voltage	36	V
V_i	Input Voltage	7	V
V_{en}	Enable Voltage	7	V
I_o	Peak Output Current (100 μ s non repetitive)	1.2	A
P_{tot}	Total Power Dissipation at $T_{pins} = 90^\circ C$	4	W
T_{stg}, T_j	Storage and Junction Temperature	- 40 to 150	$^\circ C$

جدول ۳-۶ مشخصات درایور L293D

شکل ۶-۸ نحوه اتصال دو موتور را به درایور LM293 را نشان می دهد.



شکل ۶-۸ اتصال دو موتور به L293D

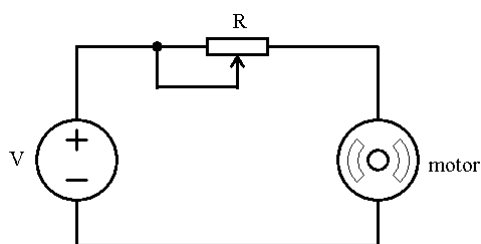
ورودی موتور A			عملکرد
Eable	In2	In2	
1	0	0	ترمز
	0	1	راستگرد
	1	0	چپگرد
	1	1	ترمز
0	in1=x , in2=x		چرخ ها آزاد

نکته: در شکل بالا نیازی به استفاده از دیودهای بازگشتی بر روی موتور نمی باشد. زیرا این دیودها در داخل آی سی تعبیه شده است.

کنترل سرعت در موتورهای DC:

تاکنون درباره چگونگی تغییر جهت در موتورهای DC مطالبی ارائه شد. ولی برای کنترل موتور DC تنها تغییر جهت کافی نمی باشد بلکه کنترل سرعت موتور نیز به اندازه کنترل جهت چرخش و یا حتی بیشتر اهمیت دارد.

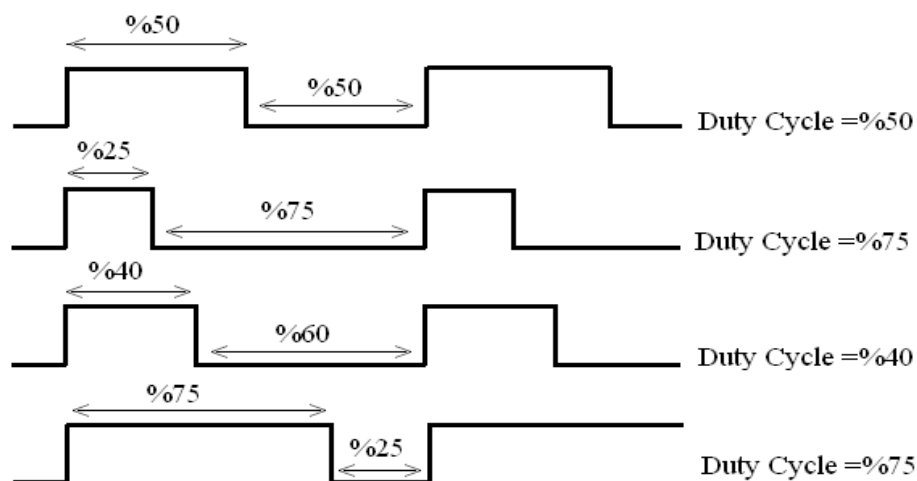
همانطور که گفته شد سرعت موتور تابعی از ولتاژ اعمال شده به دو سر آن است بنابراین با تغییر ولتاژ دو سر موتور به سادگی می توان سرعت موتور را کنترل کرد. ساده ترین روش برای کنترل موتورهای DC استفاده از یک رئوستا که به صورت سری با بار قرار گرفته شده است میباشد. این رئوستا ولتاژ اعمالی به بار را تغییر می دهد.



شکل ۶-۹ کنترل موتور DC با استفاده از رئوستا

روش بالا دارای معایبی می باشد که کاربرد آنرا کم کرده است ولی یکی از پرکاربردترین روش ها بر کنترل دور موتورهای DC استفاده از مدولاسیون عرض پالس می باشد. مدولاسیون عرض پالس (PWM) یک تکنولوژی بسیار موثر برای کنترل توان می باشد. ایده اصلی این روش بر مبنای استفاده از پالس های ولتاژ مربعی برای تغذیه موتور می باشد که در آن مقدار توان اعمالی به بار به درصد زمان وظیفه (Dutycycle) بستگی دارد.

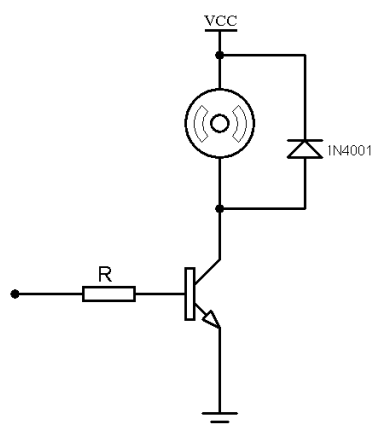
نحوه کنترل موتور به وسیله مدولاسیون عرض پالس بدین گونه است که ابتدا یک فرکانس ثابت را انتخاب کرده و سپس برای افزایش سرعت موتور Duty Cycle را افزایش و برای کاهش سرعت Duty Cycle را کاهش می دهیم.



با کنترل عرض پالسها، توان اعمال شده به بار را می توان کنترل کرد. همانطور که می دانید توان بار مجذوری از ولتاژ اعمال شده به بار است با استفاده از رابطه زیر ولتاژ متوسط اعمالی به بار در Duty Cycle های مختلف بدست می آید.

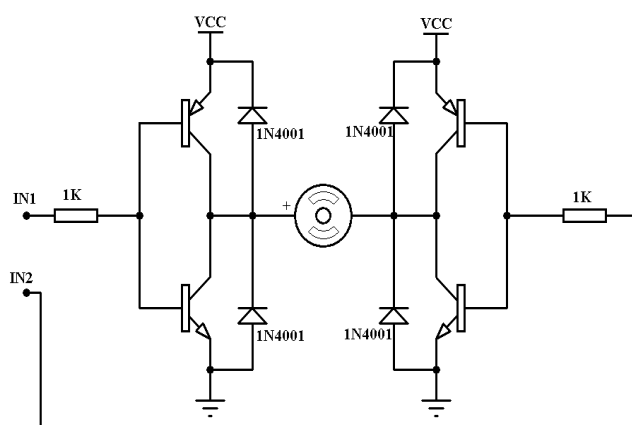
$$V_{rms} = \sqrt{\frac{1}{T} \int_0^T V(t)^2 dt}$$

همانطور که ملاحظه شد به سادگی با تغییر زمان وظیفه یک پالس سرعت موتور کنترل شد. حال مدار زیر را در نظر بگیرید.



در این مدار ما قادر هستیم به وسیله زمان وظیفه پالسهای که به بیس ترانزیستور اعمال می شود سرعت موتور را در یک جهت مشخص کنترل کنیم.

حال برای کنترل جهت و سرعت موتور از مدار شکل ۶-۱۰ استفاده می کنیم.



عملکرد	In2	In1
ترمز	0	0
راستگرد	پالس PWM	0
چپگرد	0	پالس PWM
ترمز	1	1

شکل ۶-۱۰ کنترل جهت و سرعت موتورهای DC

تولید پالس PWM از طریق میکروکنترلر:

ما به سادگی از طریق میکروکنترلر قادر هستیم پالسهای PWM را با زمان وظیفه و فرکانسهای دلخواه تولید کنیم. همانطور که میدانید تایمرها دارای چندین مد تولید PWM می باشند که با مقداردهی رجیسترها و تنظیماتی در ابزار Code Wizard ما قادر به تولید PWM هستیم.

1) تولید PWM از طریق تایمر صفر:

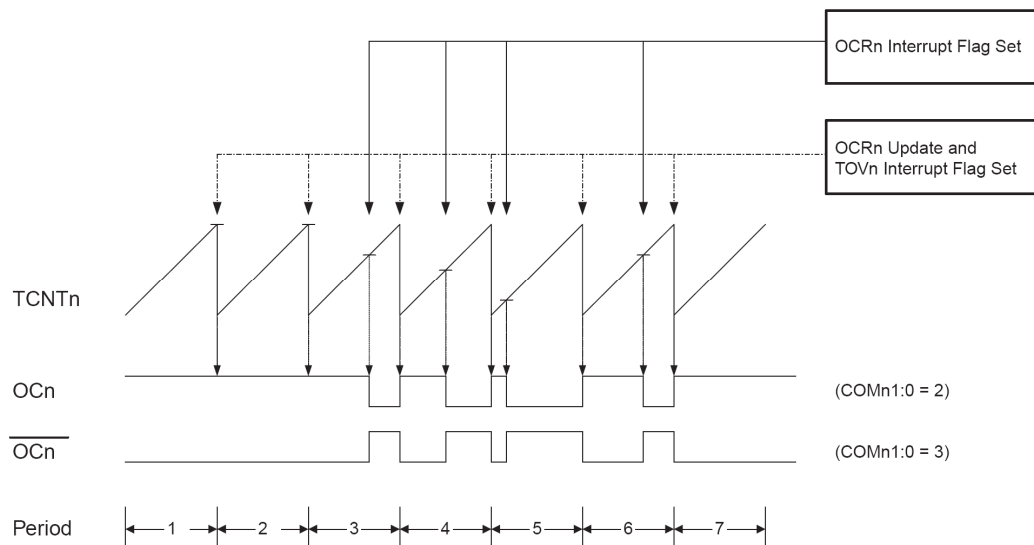
ما در اینجا قصد داریم نحوه تولید یک PWM متغیر را به شما آموزش دهیم. به همین خاطر وارد مبحث تایمرها نمی شویم و به طور کلی مد PWM سریع در تایمر صفر را مورد بررسی قرار می دهیم.

مد Fast PWM :

تایمر از مقدار صفر شروع به شمردن می کند و با رسیدن به مقدار قرارداده شده در رجیستر OCR0 (یا در قسمت Compare) پایه OC0 (یا همان پایه 4 در ATmega16) را not کرده و به شمارش خود ادامه می دهد تا به مقدار 0xff برسد و با رسیدن به این مقدار پایه مذکور را دوباره not کرده و تایمر را پاک می کند. بدین ترتیب ما قادر هستیم با تغییر محتوای رجیستر OCR0 پهنای PWM را تغییر دهیم.

پس از انتخاب مد مورد نظر در قسمت تنظیمات تایمر صفر باید نوع خروجی را نیز از قسمت output در زیر منوی Mode انتخاب کنید. گزینه Disconnected باعث غیرفعال شدن تولید PWM شده و گزینه های inverted و non-inverted به ترتیب خروجی های معکوس و

غیر معکوس PWM را مشخص می کنند. شکل زیر دیاگرام زمانی مد Fast PWM را نمایش می دهد.



دیاگرام زمانی مد Fast PWM

فرکانس موج PWM از رابطه زیر بدست می آید:

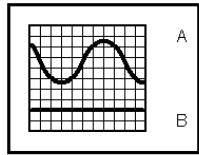
$$FOC0 = \frac{f_{clk} - I/O}{N \cdot (256 - TCNT0)}$$

در رابطه بالا N ضریب تقسیم فرکانس پالس ساعت سیستم بوده و یکی از مقادیر ۱، ۸، ۶۴، ۲۵۶، ۱۰۲۴ را به خود اختصاص می دهد (این ضریب در قسمت Clock Value مشخص می شود) و fclk-I/O کلاک تایمر می باشد.

مثال ۱: یک فرکانس یک کیلو هرتز با زمان وظیفه متغیر درست کنید. (مقدار اولیه ۲۰٪).

$$\text{DutyCycle} = \frac{OCR0}{250} \times 100\% \Rightarrow 20\% = \frac{OCR0}{250} \times 100\% \Rightarrow OCR0 = 50$$

$$FOC0 = \frac{f_{clk} - I/O}{N \cdot (256 - TCNT0)} = \frac{16000000}{64 \cdot (256 - TCNT0)} = 1000 \Rightarrow TCNT0 = 6$$

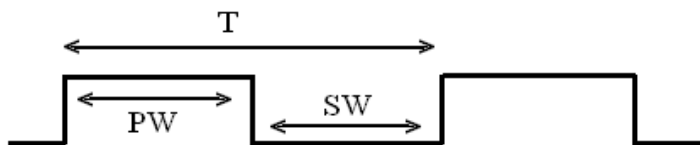


WWW.ECA.IR

```
};
```

(2) تولید PWM بدون استفاده از تایمر:

در اینجا ما قصد داریم به صورت دستی یک موج PWM با زمان وظیفه های متغیر تولید کنیم. تولید PWM از این طریق بسیار ساده می باشد بدین صورت که شما پس از انتخاب فرکانس کاری موتور، دوره تناوب (T) را بدست می آورید و برای تولید DutyCycle های مختلف از فرمول زیر استفاده کنید.



$$\% \text{Dutycycle} = \frac{PW}{T} \times 100$$

با در دست داشتن مقدار PWM ما مدت زمان یک و صفر بودن یک پالس را بدست می آوریم. برای درک بیشتر موضوع می توانید مثال زیر را مورد مطالعه قرار دهید.

مثال 2:

با استفاده از میکروکنترلر یک فرکانس 1KHZ با زمان وظیفه 70٪ تولید کنید.

با استفاده از فرمول زیر مدت زمان صفر بودن و یک بودن پالس را تعیین می کنیم.

$$T = \frac{1}{F} = \frac{1}{1K} = 1mS$$

$$\%70 = \frac{PW}{1m} = 0.7mS = 700\mu S$$

$$SW = T - PW = 0.3mS = 300\mu S$$

```
#include <mega16.h>
#include <delay.h>
void main(void)
{
```

```

DDRB=0x01;
while (1)
{
PORTB.0=1;
delay_us(700);
PORTB.0=0;
delay_us(300);
};
}

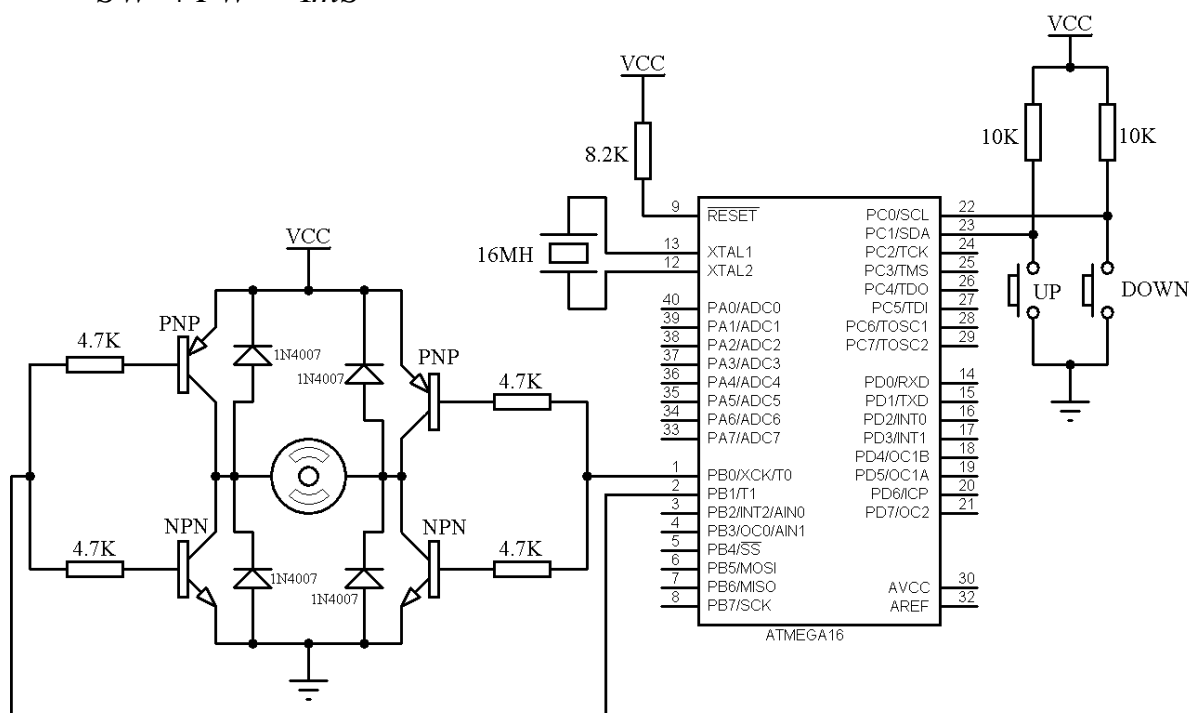
```

مثال 3:

برنامه ای بنویسید که یک موتور DC با فشار کلید up به مدت ۵ ثانیه به صورت راستگرد و راه اندازی نرم (با کمترین سرعت شروع و به بیشترین سرعت برسد) شروع به حرکت کند و با فشار کلید DOWN موتور به مدت ۵ ثانیه به صورت چپگرد و راه اندازی نرم شروع به حرکت نماید. (فرکانس موتور 1KHZ در نظر گرفته شود).

$$T = \frac{1}{F} = \frac{1}{1K} = 1mS$$

$$SW + PW = 1mS$$



شکل ۱۱-۶ شماتیک مثال کنترل موتور DC

```

#include <mega16.h>
#include <delay.h>

```

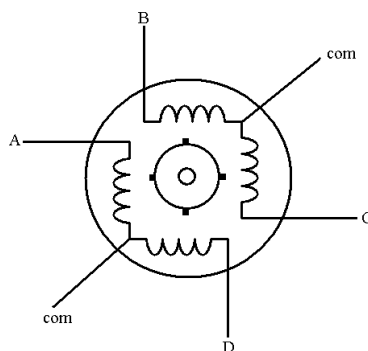
```
int sw,pw,i,i2,i3;

void delay_usec(long int ha){
for(i2=0;i2<=ha;i2++){
}
}

void main(void)
{
DDRB=0xFF;
DDRC=0x00;
PORTC=0xFF;
while (1)
{
PORTB.0=0;
PORTB.1=0;
//*****right
if(PINC.0==0){
PORTB.0=0;
sw=1000,pw=0;
for(i=0;i<=100;i++){
sw=(sw-10),pw=(pw+10);
for(i3=0;i3<=50;i3++){
PORTB.1=1;
delay_usec(pw);
PORTB.1=0;
delay_usec(sw);
}
}
}
//*****left
if(PINC.1==0){
PORTB.1=0;
sw=1000,pw=0;
for(i=0;i<=100;i++){
sw=(sw-10),pw=(pw+10);
for(i3=0;i3<=50;i3++){
PORTB.0=1;
delay_usec(pw);
PORTB.0=0;
delay_usec(sw);
}
}
}
};
}
```

موتورهای پله ای :

از موتورهای پله ای می توان برای جابجایی، حرکت، تعیین موقعیت و بسیاری از کارهای دیگر که در آنها کنترل دقیق موقعیت یک محور، اهرم و ... مورد نیاز باشد استفاده کرد. موتور پله ای وسیله پر مصرفی است که پالس های الکتریکی را به حرکت مکانیکی تبدیل می کند و در بسیاری از وسایل از جمله دیسک ها، چاپگرهای ماتریسی و رباتیک کاربرد دارد. هر موتور پله ای دارای یک هسته متحرک مغناطیسی دائمی است که رتور یا شفت هم خوانده می شود و بوسیله یک بخش ثابت به نام استاتور احاطه شده است. شکل ۶-۱۲ یکی از متداولترین انواع موتور پله ای را نشان می دهد که این موتور دارای چهار سیم پیچ استاتور و یک سر وسط مشترک می باشد.



شکل ۶-۱۲ ساختمان داخلی موتور پله ای

نحوه عملکرد یک موتور پله ای تفاوت زیادی با یک موتور dc ندارد تنها تفاوت این دو موتور در نحوه حرکت محور است. همانطور که در شکل ۶-۱۲ ملاحظه کردید یک موتور پله ای متداول دارای شش سیم می باشد که ۴ سیم برای چهار سیم پیچ استاتور و دو سر مشترک برای سیم های سر وسط (در اکثر موتورها این دو سر وسط از داخل به هم وصل می شوند و در نتیجه موتور دارای ۵ سیم می شود). با اعمال پالسهایی به هر یک از سیم پیچ ها موتور شروع به حرکت می نماید. برای حرکت یک موتور پله ای باید پالسهایی مانند جدول ۶-۴ به صورت متداول به هر یک از سیم پیچ ها اعمال شود

A	B	C	D	جهت موتور	A	B	C	D	جهت موتور
1	0	0	0	در جهت عقربه های ساعت	0	0	0	1	خلاف جهت عقربه های ساعت
0	1	0	0		0	0	1	0	
0	0	1	0		0	1	0	0	
0	0	0	1		1	0	0	0	

جدول ۴-۶ پالسهای اعمالی به هر یک از سیم پیچ ها

بنابراین برای دستیابی به عملکرد صحیح، سیم پیچهای موتور پله ای باید به صورت مناسب تغذیه شوند. این بدان معنی است که به هنگام استفاده از یک موتور پله ای نه تنها باید از مشخصه های الکتریکی دستگاه اطلاع داشت بلکه باید ویژگیهای مکانیکی آن را نیز به خوبی شناخت.

زاویه پله:

هنگامی که پالس به یکی از سیم پیچ ها اعمال شود، موتور به اندازه یک پله حرکت می کند. زاویه پله حداقل زاویه ای از چرخش مربوط به یک پله است که در موتورهای مختلف بین محدوده ۰/۷۲ درجه تا ۹۰ درجه می باشد و متداولترین زاویه پله ۱/۸ درجه است. جدول ۵-۶ تعداد پالسهای را که باید به هر یک از موتورها با زاویه پله مشخص داده شود را تعیین می کند. به عنوان مثال تعداد پالسهای که باید به یک موتور با زاویه پله ۱/۸ داده شود تا یک دور کامل بچرخد برابر ۲۰۰ می باشد.

زاویه پله	تعداد پله در یک دور
0.72	500
1.8	200
7.5	48
15	24
90	4

جدول ۵-۶ پالسهای اعمالی به موتورها با زاویه پله های مختلف

مثال) برنامه ای بنویسید که یک موتور پله ای با زاویه پله ۱/۸، ده دور بچرخد.

```
#include <mega16.h>
#include <delay.h>

#define stepper_motor PORTB
```

```

int i;
void main(void)
{
  DDRB=0xFF;
  for(i=0;i<=500;i++){
    stepper_motor=0b1000;
    delay_ms(5);
    stepper_motor=0b0100;
    delay_ms(5);
    stepper_motor=0b0010;
    delay_ms(5);
    stepper_motor=0b0001;
    delay_ms(5);
  };
}

```

پله در ثانیه و RPM:

رابطه بین RPM یا دور در دقیقه، پله در دور و پله در ثانیه از رابطه زیر بدست می آید:

$$\text{پله در ثانیه} = \frac{\text{تعداد پله در یک دور} \times \text{RPM}}{60}$$

پله در ثانیه:

اگر به مثال قبل توجه کنید بین ارسال هر پالس به موتور یک تأخیر در نظر گرفته شده است که این تأخیر مقدار پله در ثانیه را مشخص می کند. به عنوان مثال اگر تأخیر بین پالسهای ارسالی ۵ میلی ثانیه باشد مقدار پله در ثانیه برابر ۲۰۰ می شود.

RPM:

تعداد دور موتور در یک دقیقه را RPM می گویند که به سادگی با داشتن مقدار پله در ثانیه و تعداد پله موتور می توان مقدار RPM را محاسبه کرد.
نکته: موتورهای پله ای دارای گشت آور پایینی هستند بنابراین این موتور برای جاهایی که گشت آور زیاد مورد نیاز است توصیه نمی شود.

راه اندازه موتور به صورت نیم پله:

برای افزایش گشت آور در موتورهای پله ای آنها را به صورت نیم پله راه اندازی می کنند. این بدان معنی است که ممکن است در یک لحظه دو سیم پیچ وارد مدار شوند. در راه اندازی به صورت نیم پله زاویه پله نصف می شود و با نصف شدن زاویه پله گشتاور موتور زیادتر خواهد شد. جدول ۶-۶ پالسهای ارسالی به هر یک از سیم پیچ ها برای راه اندازی موتور به صورت نیم پله را نشان می دهد.

A	B	C	D	جهت موتور	A	B	C	D	جهت موتور
0	0	0	1	در خلاف جهت عقربه های ساعت	1	0	0	1	موافق جهت عقربه های ساعت
0	0	1	1		1	0	0	0	
0	0	1	0		1	1	0	0	
0	1	1	0		0	1	0	0	
0	1	0	0		0	1	1	0	
1	1	0	0		0	0	1	0	
1	0	0	0		0	0	1	1	
1	0	0	1		0	0	0	1	

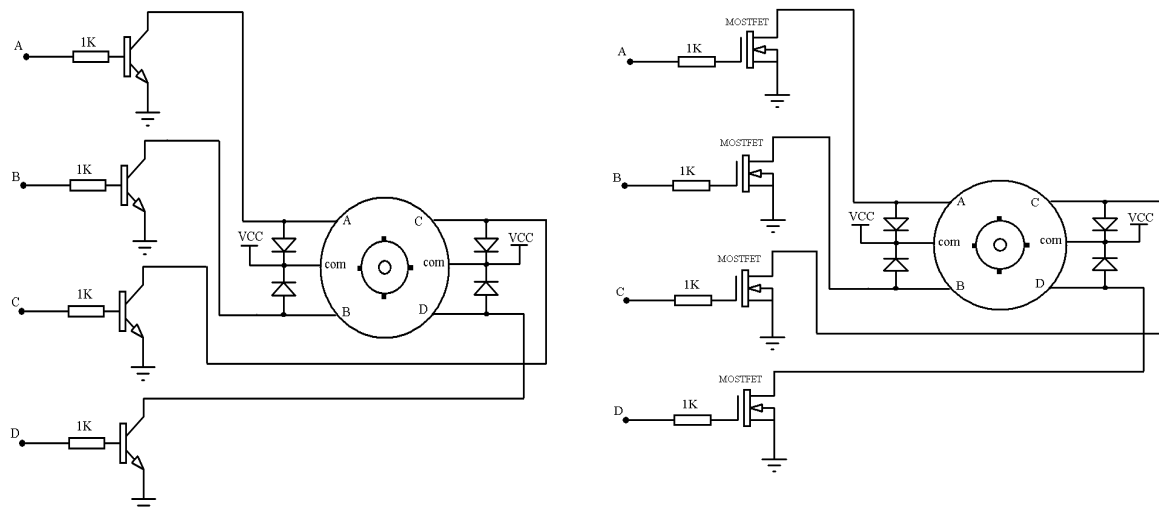
جدول ۶-۶ راه اندازی موتور به صورت نیم پله

مدارهای درایور موتور پله ای:

همانطور که می دانید نمی توان یک موتور پله ای را به صورت مستقیم به میکروکنترلر متصل کرد، برای اتصال میکروکنترلر به موتورهای پله ای از مدارهای درایور استفاده می کنیم. این مدارات می توانند شامل ترانزیستور، MOSFET، آی سی های درایور و ... باشند.

راه اندازی موتور پله ای با استفاده از ترانزیستور و MOSFET:

مدار شکل ۶-۱۳ یک راه انداز برای موتورهای پله ای می باشد. که نوع MOSFET دارای جریان دهی بیشتری نسبت به ترانزیستور می باشد.



شکل ۶-۱۳ مدار راه انداز موتورهای پله ای

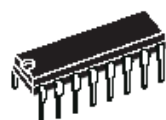
نکته: برای افزایش جریان دهی می توان ترانزیستورهای BJT را به صورت دارلینگتون بکار برد.

آی سی های درایور موتور پله ای:

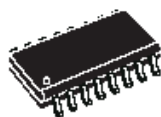
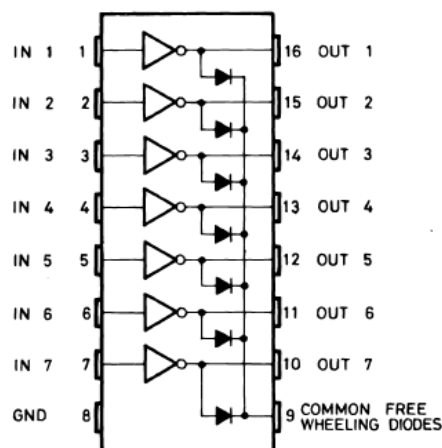
تاکنون آی سی های مختلفی برای کنترل موتورهای پله ای به بازار عرضه شده اند که ساختمان داخلی اکثر آنها از تعدادی ترانزیستور تشکیل شده است. معروفترین نوع درایور موتورهای پله ای آی سی ULN2002 و ULN2003 می باشد.

درایور ULN2002 و ULN2003 :

این درایورها برای راه اندازی موتورهای پله ای با جریان 50mA بکار می روند. ساختمان داخلی این درایورها از تعدادی ترانزیستور زوج دالینگتون تشکیل شده است.



DIP-16

SO-16
(Narrow)

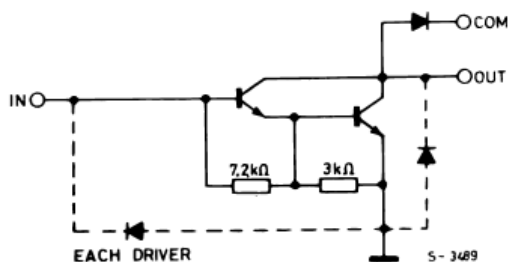
شکل ۶-۱۴ ترتیب پایه ها و ساختمان داخلی درایور ULN2002A

جدول ۶-۷ مشخصه های این درایورها را نشان می دهد.

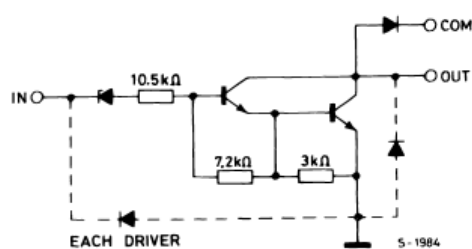
Symbol	Parameter	Value	Unit
V_O	Output voltage	50	V
V_I	Input voltage (for ULN2002A/D - 2003A/D - 2004A/D)	30	V
I_C	Continuous collector current	500	mA
I_B	Continuous base current	25	mA

جدول ۶-۷ مشخصه درایور ULN2002A

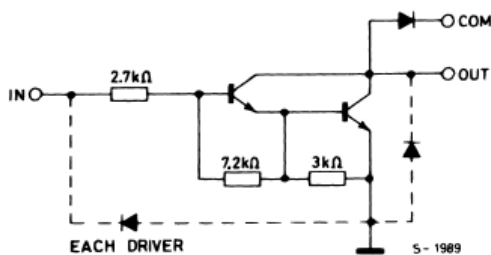
درایورهای خانواده ULN200X تقریباً شبیه یکدیگرند و تنها تفاوت آنها در مقدار مقاومت روی بیس ترانزیستورها می باشد که شکل ۶-۱۵ این مقدار را برای هر یک از درایورهای خانواده ULN200X مشخص کرده است.



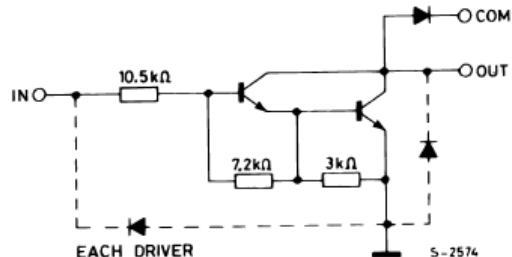
ULN2001 (each driver)



ULN2002 (each driver)



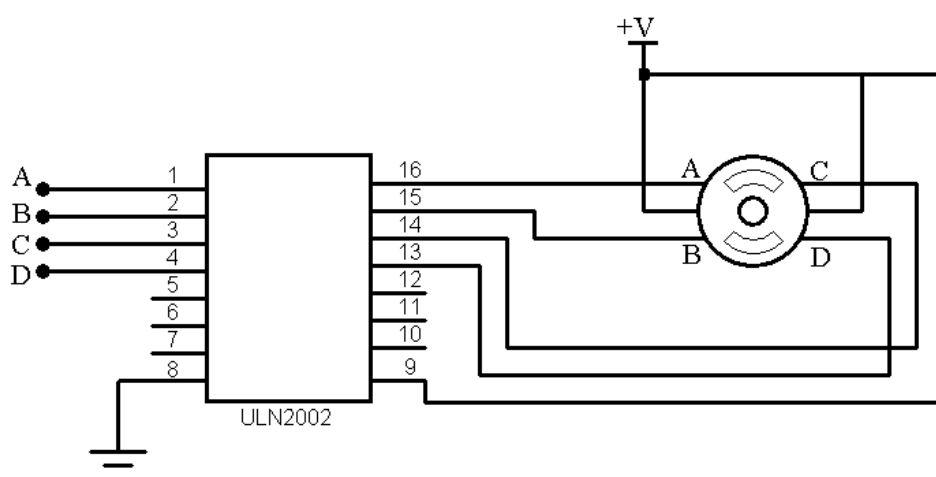
ULN2003 (each driver)



ULN2004 (each driver)

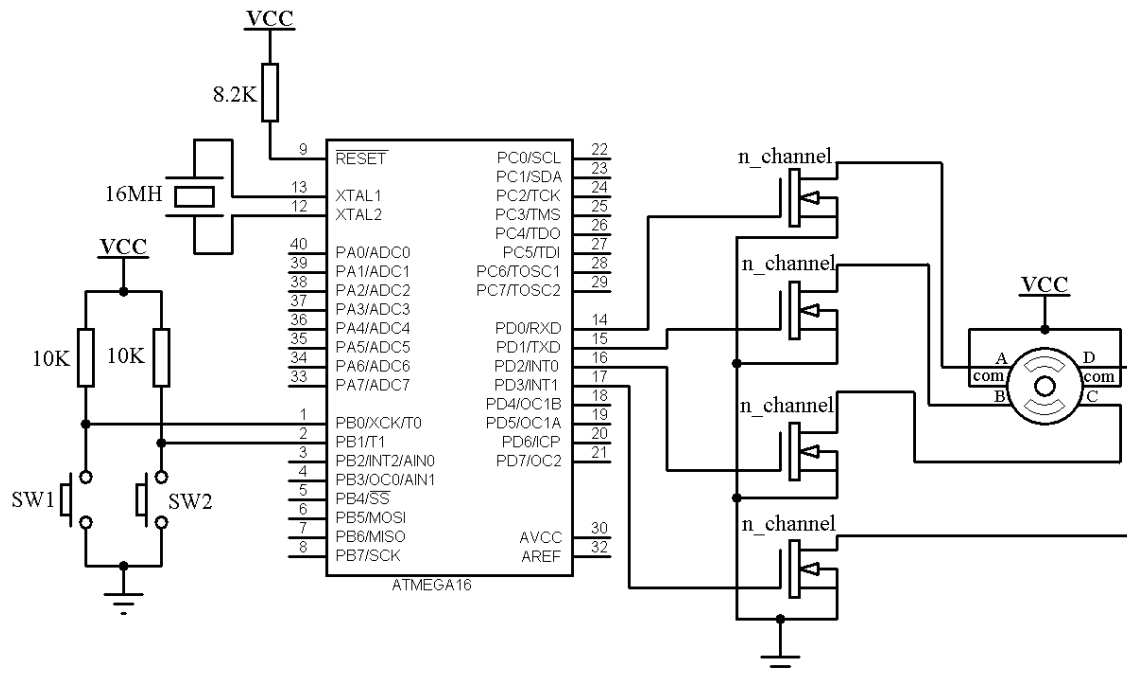
شکل ۶-۱۵

در آی سی شماره ULN2003 مقدار مقاومت بیس 2700Ω می باشد که مشخصه آی سی را با مدارات منطقی TTL سازگار می سازد. در آی سی شماره ULN2002 مقاومت موجود بر روی بیس $10.5K\Omega$ است که مشخصه آی سی را با مدارات منطقی Cmos سازگار می کند. شکل ۶-۱۶ نحوه اتصال یک موتور پله ای چهار فاز را به درایور ULN2002 را نشان می دهد.



شکل ۶-۱۶ نحوه اتصال موتور پله ای به درایور ULN2002

مثال) برنامه ای بنویسید که با هر بار فشار کلید SW1 موتور ۷۲۰ درجه به صورت نیم پله و در جهت حرکت عقربه های ساعت بچرخد و با فشار کلید SW2 موتور ۵ دور به صورت تمام پله و در خلاف جهت عقربه های ساعت بچرخد.



شکل ۶-۱۷ شماتیک مثال کنترل موتور پله ای

```
#include <mega16.h>
#include <delay.h>

int i;
void main(void)
{
  DDRB=0x00;
  PORTB=0xFF;
  DDRD=0xFF;
  while(1) {
    if (PINB.0==0) {
      for(i=0; i<=100; i++) {
        PORTD=0b1001;
        delay_ms(5);
        PORTD=0b1000;
        delay_ms(5);
        PORTD=0b1100;
        delay_ms(5);
        PORTD=0b0100;
        delay_ms(5);
      }
    }
  }
}
```

```
PORTD=0b0110;
delay_ms(5);
PORTD=0b0010;
delay_ms(5);
PORTD=0b0011;
delay_ms(5);
PORTD=0b0001;
delay_ms(5);
}
}
if (PINB.1==0) {
    for (i=0; i<=250; i++) {
        PORTD=0b0001;
        delay_ms(5);
        PORTD=0b0010;
        delay_ms(5);
        PORTD=0b0100;
        delay_ms(5);
        PORTD=0b1000;
        delay_ms(5);
    }
}

};
}
```