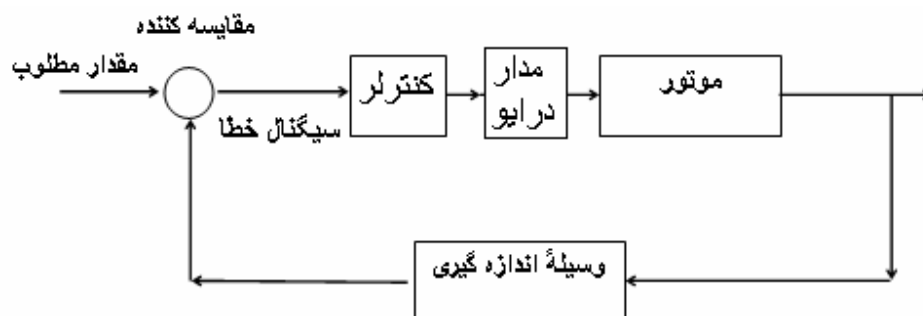


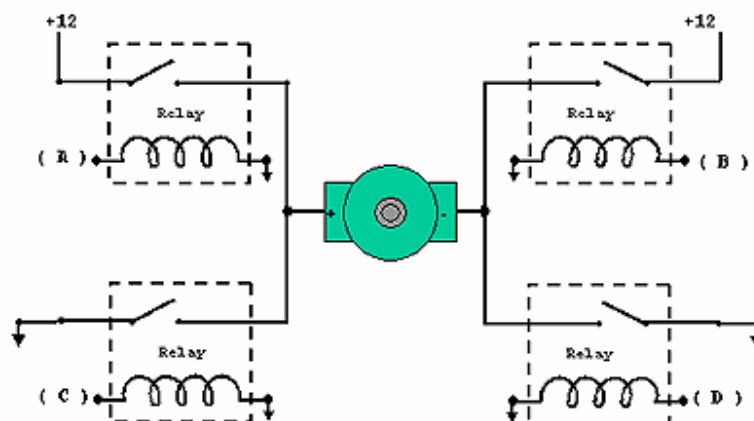
در بیشتر موارد کنترل دقیق موتورهای DC برای ما اهمیت زیادی دارد، به این دلیل که می‌خواهیم سرعت موتور مستقل از بار باشد و همیشه مقدار مطلوب ما را داشته باشد. مثلاً در یک ربات مسیریاب با دو موتور اگر سرعت موتورهای به دقت کنترل نشود و سرعت یکی از موتورها از دیگری کمی بیشتر باشد ربات از مسیر خارج خواهد شد، پس کنترل دقیق موتورهای امری اجتناب‌ناپذیر است.

برای کنترل دقیق یک موتور DC مانند کنترل سایر فرآیندها ما نیاز به یک حلقه کنترلی کامل داریم. در شکل زیر یک حلقه کنترلی کلاسیک شامل فیدبک نمایش داده شده است.



در قسمت قبل راجع به سنسور نوری (وسیله اندازه گیری) توضیحاتی دادم، البته لازم به یادآوری است که روش‌های دیگری هم برای اندازه گیری سرعت موتور وجود دارد، اما در این قسمت می‌خواهم راجع به مدارهای درایو و روش‌های درایو موتور صحبت کنم.

مدار پل H یکی از پرکاربردترین مدارات کنترل موتور DC است، در شکل زیر شمای بسیار ساده یک پل H نمایش داده شده است.



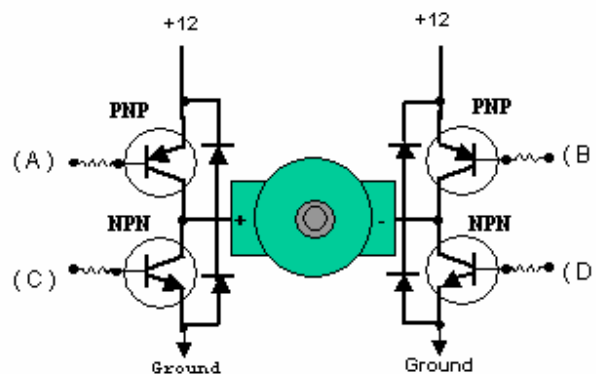
پل H رله ای

با توجه به شکل فوق با فعال شدن رله ها حالات مختلفی به وجود می آید که در جدول زیر نمایش داده شده اند.

A	B	C	D	عملکرد
1	0	0	1	مستقیم
0	1	1	0	معکوس
1	1	0	0	ترمز
0	0	1	1	ترمز

توجه: بقیه حالات ممکن است باعث اتصال کوتاه شدن دو سر منبع تغذیه شود.

می بینید که به کمک این پل می توان به راحتی جهت چرخش موتور را تغییر داد. در شکل زیر یک مدار پل H به وسیله ترانزیستور نمایش داده شده است.



پل H ترانزیستوری

پس ما به راحتی توانستیم که جهت چرخش را کنترل کنیم، اما هدف ما کنترل سرعت و یا موقعیت موتور DC است. با توجه به معادلات موتور DC داریم (میدان ثابت فرض شده است):

$$V = K \cdot \omega + I \cdot R + L \cdot \frac{dI}{dt}$$

$$\tau = K' \cdot I$$

که در آن به ترتیب داریم:

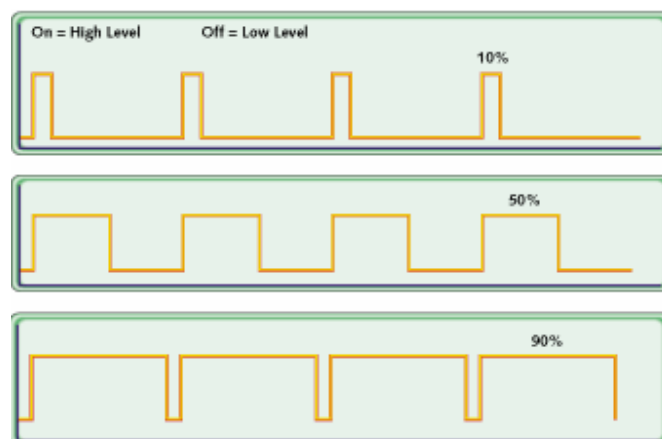
V	ولتاژ دو سر آرمیچر:
K	ثابت ضد EMF:
ω	سرعت زاویه ای روتور:
I	جریان آرمیچر:
R	مقاومت آرمیچر:
L	اندوکتانس آرمیچر:
τ	گشتاور نیروی موتور:
K'	ثابت گشتاور نیرو:

پس سرعت موتور DC را می توان با تغییر ولتاژ دو سر آن و گشتاور آن را می توان توسط جریان موتور کنترل کرد.

یکی از روش های کنترل موتور استفاده از PWM(Pulse Width Modulation) است. البته روشهای دیگری هم وجود دارد که مزیت این روش در کاهش تلفات حرارتی است.

PWM صورت های مختلفی دارد که من در اینجا به نوع فرکانس ثابت آن می پردازم.

در این روش یک موج مربعی با فرکانس ثابت داریم که در آن نسبت بین یک و صفر بودن قابل تنظیم است.

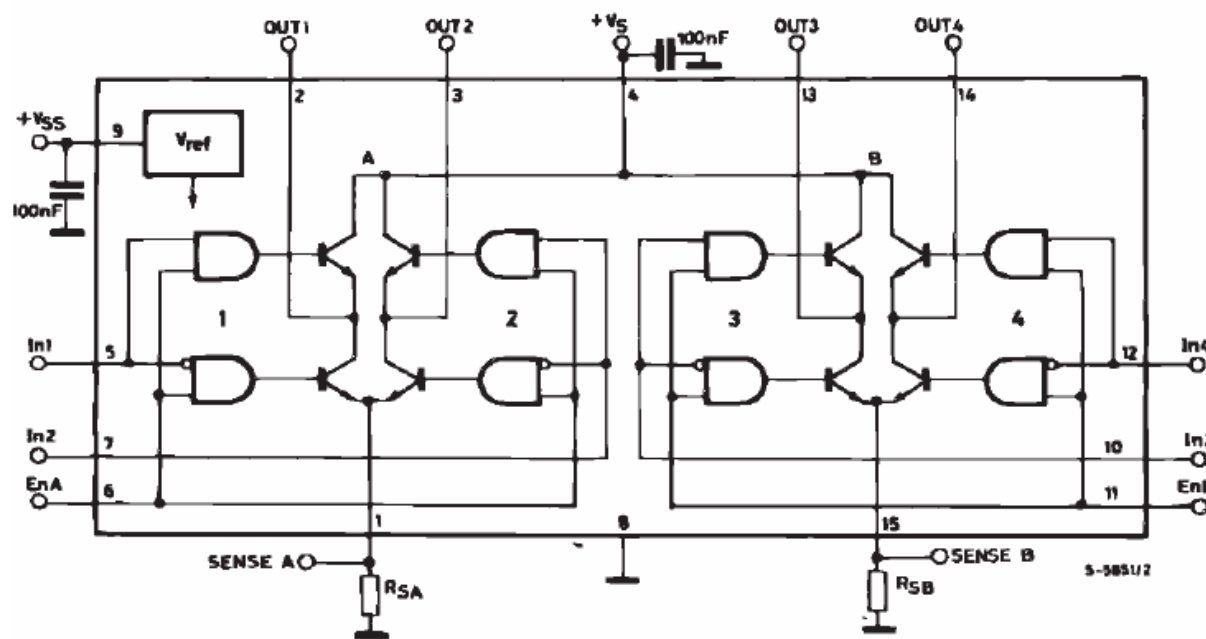


چند سیگنال PWM با نسبت صفر و یک متفاوت

با تغییر نسبت یک به صفر در سیگنال PWM توان متوسطی که به موتور می رسد تغییر می کند، در نتیجه سرعت چرخش موتور نیز متغیر خواهد بود.

حال اگر این سیگنال PWM را به پل H (ترمینال های A,B,C,D) اعمال کنیم علاوه بر کنترل جهت حرکت موتور می توانیم سرعت موتور را نیز به کمک این پل کنترل کنیم. آی سی های کوچکی مانند L293D، L298 همین پل H هستند، البته نمونه های جدیدتری هم موجود است که

شما می توانید بر حسب نوع موتور و یا جریان مورد نیاز از آنها استفاده کنید ولی اساس کار همه یک چیز است. بلوک دیاگرام آی سی L298 در شکل زیر نمایش داده شده است.

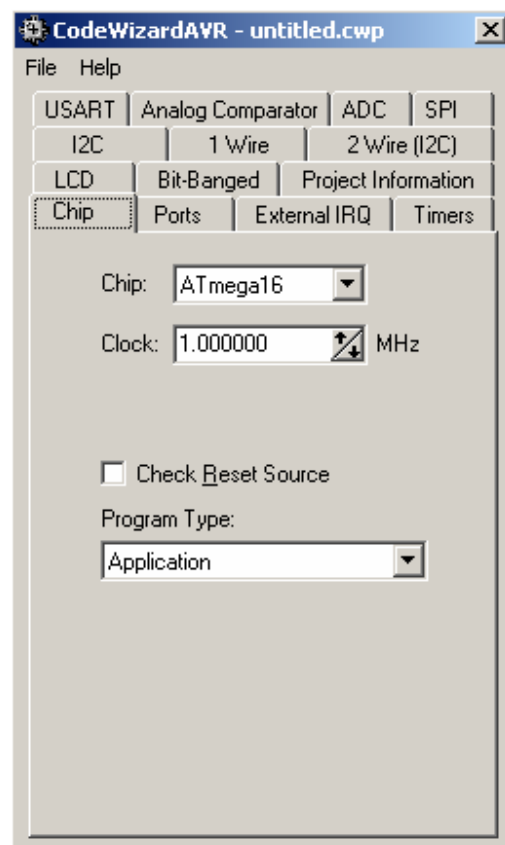


همانطور که می بینید این آی سی شامل 2 پل H است که دارای 4 پین خروجی، 4 پین ورودی، 2 پین Enable، 2 پین Sense و پین های تغذیه و زمین است. پین های خروجی مربوط به ترمینال های موتور می شود، به کمک پین های ورودی می توانید جهت چرخش را مشخص کنید و در عین حال سیگنال PWM را برای کنترل توان به آنها اعمال کنید، توسط پینهای Enable شما می توانید ورودی ها را فعال و یا غیر فعال کنید (توجه کنید که شما می توانید سیگنال PWM را به این پین اعمال کنید)، پین های Sense هم برای محدود کردن جریان بار استفاده می شود.

شما به راحتی می توانید به کمک یکی از این چپ ها یک یا دو موتور DC را کنترل کنید. برای اینکار کافیست که ابتدا توسط پینهای ورودی یکی از پلها جهت را مشخص کنید و سپس سیگنال PWM را به پین Enable همان پل اعمال کنید. (به این روش اعمال سیگنال PWM، sign-magnitude می گویند.)

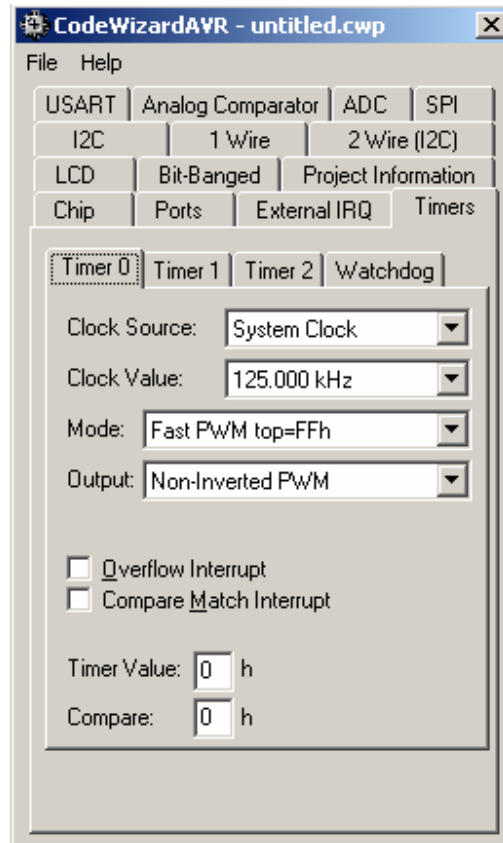
چگونه سیگنال PWM را بسازیم؟

برای ساختن PWM روش ها و مدارات مختلفی وجود دارد اما یکی از آسانترین روش ها استفاده از میکروکنترلر است. به کمک تایمرهای 8 بیتی و 16 بیتی میکروکنترلر های AVR شما به راحتی می توانید سیگنال PWM با فرکانس و نسبت یک به صفر متغیر بسازید. من در این مقاله از نرم افزار Codevision AVR استفاده می کنم. به کمک قسمت CodeWizardAVR این نرم افزار به راحتی می توانید تایمرها را برای ساخت سیگنال PWM تنظیم کنید. برای اینکار ابتدا از نوار Tools گزینه CodeWizardAVR را انتخاب کنید، پنجره ای به شکل زیر باز می شود، همانطور که در شکل می بینید نوع چیپ و فرکانس آنرا می توان از این محل تعیین کرد؛ که در این مورد چیپ Atmega 16 با فرکانس 1MHz انتخاب شده است.



سپس روی نوار Timers کلیک کنید پنجره ای باز می شود، که در آن 3 تایمر Atmega 16 نمایش داده می شود، ما برای تولید PWM از تایمر

0 که 8 بیتی است استفاده می کنیم، همانطور که در شکل زیر می بینید کلاک، مد و خروجی را برای تایمر 0 مشخص می کنیم.



فرکانس سیگنال PWM خروجی از رابطه زیر به دست می آید:

$$f_{PWM} = \frac{Clock_Value_Timer0}{256}$$

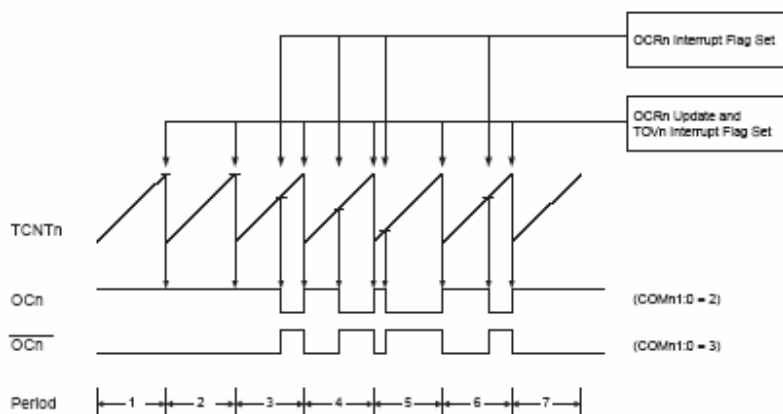
که در این مورد:

$$f_{PWM} = \frac{125000}{256} = 448.28$$

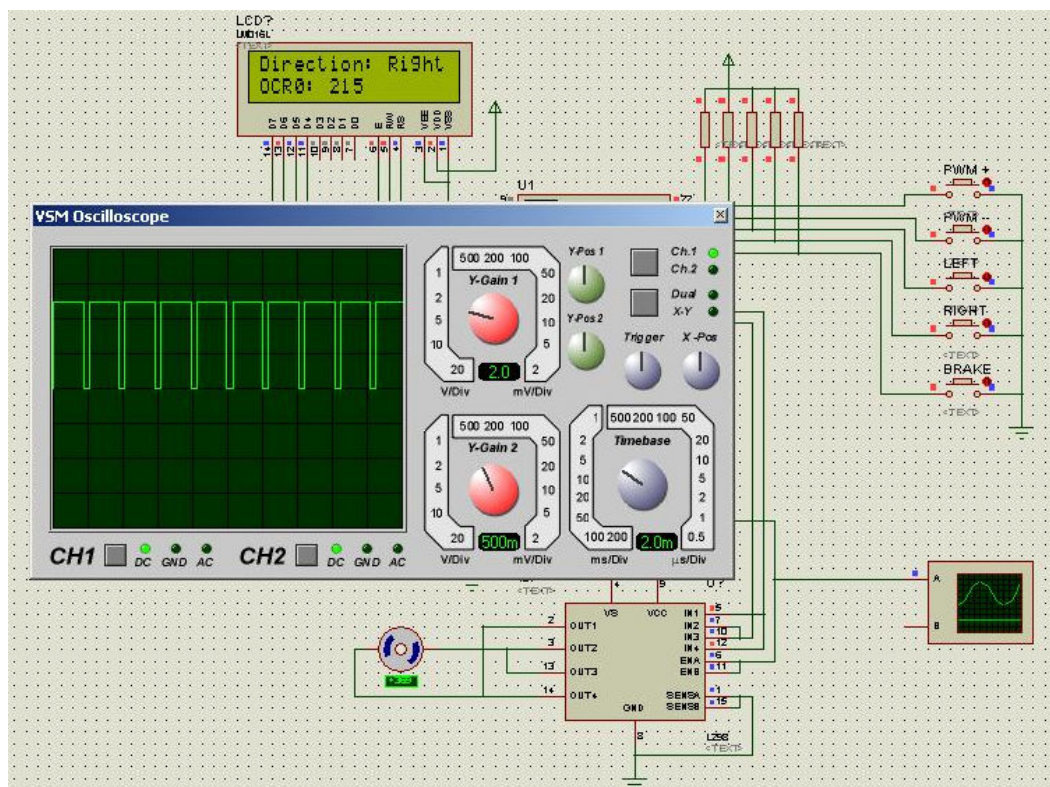
لازم به ذکر است که فرکانس PWM مناسب برای موتور های کوچک 200Hz و برای موتورهای بزرگ 50Hz است. شما می توانید بسته به موتور خود فرکانس مناسب را با تغییر فرکانس میکروکنترلر و یا مقدار کلاک تایمر بدست آورید.

بعد از انجام تغییرات از نوار File گزینه Generate, Save and Exit را انتخاب کنید. یک پیغام برای خروجی کردن پایه OC0 ظاهر می شود، گزینه Yes را انتخاب کنید.

اکنون با تغییر مقدار رجیستر OCR0 می توانید نسبت یک به صفر سیگنال خروجی را تغییر دهید. شکل زیر نحوه کار تایمر را توضیح می دهد، مجال شرح کامل آن در این مطلب نیست، اگر در این مورد مشکلی داشتید با ایمیل من تماس بگیرید.



همراه این مطلب یک فایل شبیه سازی به وسیله پروتئوس، سورس برنامه میکروکنترلر در اختیار شما قرار می دهم، که در آن سرعت یک موتور DC به صورت حلقه باز توسط میکروکنترلر و یک آی سی L298 کنترل می شود.



با استفاده از کلید PWM+ میتوان نسبت یک به صفر را افزایش و با کلید PWM- کاهش داد، سه کلید دیگر هم همانطور که از نامشان پیدا است مربوط به جهت حرکت موتور هستند. روی LCD شما می توانید مقدار OCR0 را مشاهده کنید، توجه کنید که این رجیستر 8 بیتی است و ماکزیمم مقدار آن 255 است که در این حالت پایه OC0 (خروجی PWM) همواره یک است. یعنی خروجی PWM ما 100% است.