کنترل دورموتورDC باکنترلرPID بصورت WIRELESS

محمد جعفری

تابستان ۹۱

میکرو دیزاینر الکترونیک

Melec.ir

فهرست مطالب

پیشگفتار

این پروژه بگونه ای انتخاب شده است که توانسته از تمامی گرایشهای رشته برق بهره گیرد و از انها جهت پیاده سازی نتیجه ای مطلوب استفاده نماید. اما می توان گفت که قسمت اصلی پروژه بهره گیری از سیستمهای کنترلی است ، اینگونه که جهت کنترل موتور(dc) از اصول کنترل استفاده شده است و جهت رسیدن به مقدار مطلوب دور موتور از کنترلر (pid) استفاده شده است .

علاوه بر اصول کنترلی در این پروژه سعی شده که از مفاهیم الکترونیکی و مخابراتی نیز استفاده شود به همین منظور جهت راه اندازی موتور از درایورهای ترانزیستوری استفاده شده که در طراحی انها از اصول الکترونیک بهره گرفته شده،وجهت ارسال اطلاعات ودریافت ان از ماژولهای فرستنده وگیرنده ی (hm-tr) استفاده شده که قسمت مخابراتی پروژه را تشکیل می دهد.

در ادامه به بررسی کامل قسمتهای ذکر شده خواهیم پرداخت ، قسمت بسیار مهم این پروژه قسمت میکروکنترولر آن می باشد و میکرو استفاده شده در این پروژه ای سی (AT-mega16) از خانواده ی AVRمی باشد که به بررسی مشخصات این ای سی خواهیم پرداخت .

مشخصه ی بارز این پروژه پیاده سازی کنترولر(pid) بصورت نرم افزاری می باشد، بگونه ای که تمامی مراحل کنترل اعم از (تناسبی ، انتگرالی ، مشتقگیری) بصورت عددی صورت می گیرند.این موضوع در قسمت بررسی برنامه پروژه کاملا مشهود است.

موتورهای dc کاربردهای زیادی دارند،از کاربردهای آن ها می توان به ساخت روبات ها، شیشه بالابر خودروها، سشوارها، ضبط صوت ها، انواع ماشین های کنترلی، لوازم ورزشی مانند تردمیل ها و ... اشاره نمود. اغلب موتورهای الکتریکی دوارند، اما موتور خطی هم وجود دارد. در یک موتور دوار بخش متحرک (که معمولاً درون موتور است) روتور و بخش ثابت استاتور خوانده می‌شود. روتور شامل آهنرباهای الکتریکی است که روی یک قاب سیم پیچی شده است. گر چه این قاب اغلب آرمیچر خوانده می‌شود، اما این واژه عموماً به غلط بکار برده می‌شود. در واقع آرمیچر آن بخش از موتور است که به آن ولتاژ ورودی اعمال می‌شود یا آن بخش از ژنراتور است که در آن ولتاژ خروجی ایجاد می‌شود. با توجه به طراحی ماشین ، هر کدام از بخشهای روتور یا استاتور می‌توانند به عنوان آرمیچر باشند.

بطور کلی موتور الکتریکی وسیله ای است که جریان الکتریسیته را به حرکت مکانیکی تبدیل می نماید. اغلب موتورهای الکتریکی حرکت مکانیکی را به صورت دورانی ایجاد می نمایند. همانگونه که می دانیم، به سیم حامل جریان در میدان مغناطیسی نیرو وارد می شود. اساس عملکرد موتورهای الکتریکی نیز استفاده از این نیرو برای دوران دادن شافت موتور به دور خود می باشد.

سرعت چرخش موتورهای DC وابسته به ولتاژ برق ورودی به آن ها و قدرت این موتورها نیز وابسته به جریان عبوری از آن ها می باشد.که ما جریان موتور را از طریق ماسفت قدرت (irf250) تامین کردیم و اساس کنترل سرعت موتور نیز به میزان مقدار میانگین موج (pwm) حاصل از میکرو کنترلر می باشد.

تارخچه کنترل مدرن

تلاش اولیه بشربرای درک زمان وتعیین موقعیت خود درشبانه روزاز اولین گامها در طراحی سیستمهای کنترل است که به ساخت ساعتهای آبی منجرگردید اولین ساعتهای آبی توسط یونانیها ومصریان در حدود ۲۷۰ سال قبل ازمیلاد مسیح ساخته شد وتا قرن هفدهم میلادی نیزکاربرد داشت.درهمان دوران سیستمهای کنترل سطح روغن چراغها نیزطراحی شد.با وقوع انقلاب صنعتی دراروپا کوره‌ها، بویلرها، موتورهای بخارپیشرفته ورگولاتورهای شناور طراحی شد که امکان کنترل آنها توسط سیستمهای ساده امکان پذیر نبود لذا سیستمهای کنترل پیشرفته تری پس از انقلاب صنعتی طراحی شدند . کنترل آسیاب‌های بادی که برای اولین بار توسط ایرانیان در قرن هفتم میلادی ساخته شدند گام مهمی در پیاده سازی کنترل خودکار به حساب می‌آید.این آسیاب‌ها درسال ۱۲۰۰میلادی وارد اروپا شدند وتا سال ۱۶۰۰میلادی مورد استفاده قرار گرفتند. در این آسیاب‌ها دو نوع سیستم کنترل وجود داشت ؛ یکی کنترل جهت قرارگیری آسیاب به طوری که بتواند از حداکثر نیروی باد استفاده کند و دیگری کنترل میزان گندم وارده به درون آسیاب. هر دوی این سیستم‌ها به صورت کاملا خودکار عمل کرده و نیاز به حضور هیچ کارگری نبود.

تقدیر و تشکر

ابتدا لازم می دانم نهایت تشکروسپاس را از پدر و مادرعزیزم که در تمامی لحظات و مراحل زندگی یارو یاور من بوده اند ،داشته باشم وبر دستان پر مهر این عزیزان بوسه زنم.هرچند که هرگز نمی توان ذره ای از زحمات بی دریغ انها راجبران کرد.

با امید موفقیت ،پیروزی و سربلندی تمامی این عزیزان

محمد جعفری

تابستان ۹۱

فصل اول

عنوان پروژه و تحقیق

هدف این پروژه کنترل موتور dc است که برای این منظور احتیاج به یک ساختار کنترلی داریم ، در این پروژه از ساختار کنترل حلقه بسته استفاده شده است .

در حالت کلی این پروژه به چندین بخش تقسیم می شود و در هر بخش به نحوه عملکرد ان می پردازیم :

۱ – شمارش دور موتور
۲ – انتقال اطلاعات از موتور به میکرو
۳ – پردازش اطلاعات توسط میکرو
۴ – انتقال اطلاعات از میکرو به موتور

شمارش دور موتور:

شمارش تعداد دور موتور که ازقسمتهای مهم این پروژه است ، بدلیل اینکه فرمان کنترولی صادر شده توسط نرم افزار بر مبنای شمارش موتور می باشد . به همین دلیل برای داشتن یک فرمان کنترولی دقیق برای رسیدن به هدف نیازبه اطلاعات دقیق از دور موتور داریم که در ادامه به بررسی نحوه ی شمارش موتور با این دقت می پردازیم.

شمارش موتور توسط میکروکنترولرانجام میگیرد به این صورت که توسط انکدر نوری که روی شفت موتور تعبیه شده و دارای ۲۸ شکاف نوری است دقت خوبی را تضمین میکند . الگوریتم شمارش به این گونه است که از وقفه های خارجی میکروکنترولر AT mega 16 استفاده شده که با هر لبه ی بالا رونده ی موج تشکیل شده توسط انکدر نوری یک وقفه از میکرو کنترولر گرفته می شود وهمزمان تایمر میکروکنترولر عمل شمارش را در حالت مد نرمال انجام می دهد و عملیات شمارش را با امدن وقفه متوقف می کند . نکته ای که باید به ان توجه نمود این است که نباید تایمر قبل از امدن وقفه سرریز کند .برای اطمینان از این موضوع یک ازمایش ساده لازم است ، فرض می کنیم موتور در حداکثر دور خود است در این حالت شفت موتور قادر است ۶۰ دور در ثانیه بزند یعنی تعداد (۲۸*۶۰=۱۶۸۰) پالس را انکدرمی خواند درثانیه ، (X=1/1680) X مدت زمان یک دوره ی تناوب انکدراست. یعنی فاصله ی بین دو وقفه در ماکزیمم دور برابر ۰.6 ms است .

در حالی که تایمر ۱ شانزده بیتی (AT- mega16) بافرکانس (8 MHz) در مد نرمال (FFFF) کار میکند ،مدت زمان سرریز تایمر در این حالت برابر(۱/۸)
us*(FFFF)=3.19ms است که نشان می دهد تایمر ما در این مدت زمانی ، سرریز نخواهد داشت .

انتقال اطلاعات از موتور به میکرو:

جهت انتقال اطلاعات بصورت وایرلس از ماژول فرستنده و گیرنده (hm tr) استفاده شده است که برخی از مشخصات ان را بررسی می کنیم.

مشخصات این ماژول

پروتکل ونحوه عملکردمداردرارسال اطلاعات به fsk میباشدوبا توجه به نوع فرکانس کاری وبرد مدار نیازبه مجوز نداریم.
از دیگرویژگی های بارزمدار تعیین فرکانس کاری ارسال و دریافت اطلاعات ، نرخ ارسال اطلاعات ، تغییر رنج ارسال اطلاعات در یک رنج وسیع و دیگر مشخصات فرستنده توسط نرم افزار مخصوص این ماژول میباشد
فرکانس کاری این ماژول در انواع ۳۱۵ و ۴۳۴ و۸۶۸ و۹۱۵ مگاهرتز میباشد ، البته درهنگام خرید انتن باید با توجه به فرکانس ماژول انتن را خریدرای شود .
پروتکل ارسال و دریافت اطلاعات به صورت اتوماتیک توسط خود ماژول کنترل میشود ،همچنین این ماژول از حساسیت بالایی در میزان اطلاعات دریافتی و ارسالی میباشد
مهمترین ویژگی این ماژول تبادل اطلاعات با کامپیوترو میکرو از طریق پروتکلUART سریال میباشد که میتوان به صورت TTL یا rs232 تبادل اطلاعات را انجام داد

پردازش اطلاعات توسط میکرو

در حالت کلی پردازش اطلاعات بصورت نرم افزاری بوده که اطلاعات کامل ان در فصل سوم بررسی خواهد شد اما در این قسمت به نحوه عملکرد کنترل pid بصورت نرم افزاری و عددی خواهیم پرداخت .

نرم افزار این پروژه ساختار کنترل pid است که در سه مرحله ی جدا مقدار داده

ارسال شده از موتور را گرفته و ابتدا مقدار error را از فرمول زیر حساب می کند .

error= setpoint – sensor

مرحله ی اول محاسبه ی قسمت تناسبی است که مقدار error تنها در یک بهره استاتیکی ضرب می شود ، مهمترین قسمت بدست اوردن مقدار بهره تناسبی است که در این پروژه بدلیل نبود تابع تبدیل سیستم این مقدار از روش سعی و خطا بدست امده .

مرحله ی دوم محاسبه ی مقدار انتگرالی است که نرم افزار مقدار خطای هر مرحله را جمع می کند و در ثابت انتگرال که مشابه ضریب تناسبی بدست می اید ضرب می کند .

integral=integral+(error*dt)

مرحله ی سوم مقدار خطای فعلی از خطای قبلی کم شده و تقسیم بر مدت زمان بین دو خطا می شود سپس در ثابت مشتق ضرب می شود و مقدار قسمت مشتقگیری نیز بدست می اید

derivative=(error-pre_error)/dt

در اخر مقادیر بدست امده از سه مرحله ی قبل با هم جمع شده و فرمان کنترلی صادر می شود این مقدار در ضریبی ضرب شده تا در رنج مقادیر ocr میکرو قرار گیرد سپس اماده ارسال می شود.

```
output=kp*error+ki*integral+kd*derivative;

temp=output*4.2;
```

کنترل کننده PID، که یک کنترل کننده سه بخشه، شامل بخش های تناسبی، انتگرال گیری، و مشتق گیری است، پر کاربرد ترین کنترل کننده در صنعت است، به طوری که حدود نود درصد کل کنترل کننده های مورد استفاده در صنعت، یا PID هستند، و یا از آن در ساختار های کنترلی دیگر استفاده می کنند. این امر به تنهایی گویای اهمیت این کنترل کننده است.

بر خلاف ظاهر ساده PID، طراحی این کنترل کننده، در عمل، فراتر از تنظیم سه پارامتر اصلی آن است. عوامل مختلفی در عملکرد این کنترل کننده تاثیر گذار اند، که از جمله ساختار کنترل کننده، درجه پروسه، نسبت ثابت زمانی غالب سیستم به زمان مرده پروسه، دینامیک عنصر محرک، نوع فیلتر بخش مشتق گیر و تنظیم پارامتر آن، رفتار غیر خطی در سیستم و غیره را می توان برشمرد. هر یک از این عوامل می توانند نقشی در روند طراحی و تنظیم کنترل کننده PID داشته باشند.

از این میان، دو مسئله مهم، عبارت اند از: طراحی فیلتر بخش مشتق گیر، و جبران آثار اشباع شدن عنصر محرک. حال در این قسمت، مبانی و اصول کنترل PID، اصلاحات اساسی در قانون پایه این کنترل کننده، ساختار های مختلف و منطق حاکم بر سه عمل آن، و نیز راهکار هایی برای انتخاب نوع کنترل کننده، معرفی و بحث می شوند.

نحوه محاسبه ی (Kp,Ki,Kd)

در سیستم هایی مانند این پروژه که مقدار ضرایب تناسبی،انتگرالی و مشتقی معلوم نیست بدلیل نبود تابع تبدیل سیستم باید از روش سعی و خطا استفاده کرد.

این ضرایب در پایداری و پاسخ گذرا و پاسخ حالت دایمی تاثیر گذار هستند این روش به این گونه است که ابتدا ضرایب انتگرال گیرو مشتقگیر را صفر می کنیم سپس مقدارضریب تناسبی انقدرافزایش می دهیم تا مقدارخروجی به مقدار ورودی برسد و ناپایدار نشود سپس با ضریب انتگرالی میزان زمان نشست و زمان خیز را تنظیم می کنیم و در نهایت اگر لازم بود برای کاهش فرا جهش سیستم مقدار ضریب مشتقگیر را تنظیم می کنیم با این روش سعی و خطا می توان به نتیجه مطلوب در چنین سیستمهایی رسید.

انتقال اطلاعات از میکرو به موتور

فرمان کنترلی صادر شده توسط کنترلر pid توسط ماژول فرستنده و گیرنده ی(hm tr) به ورودی موتور ارسال می شود . ای فرایند انقدر ادامه پیدا می کند تا مقدار error به صفر برسد و مقدار دور خروجی برابر مقدار setpoint شود.

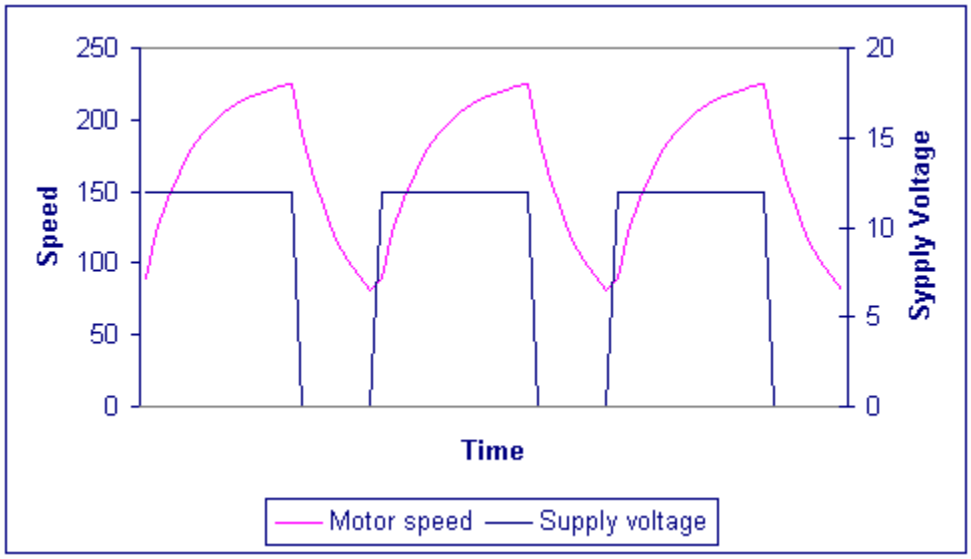
تئوری کنترل سرعت موتور دی سی

سرعت موتور DC به طور مستقیم متناسب با ولتاژ تغذیه، بنابراین اگر ولتاژ تغذیه از ۱۲ ولت تا ٦ ولت کاهش، موتور، با نصف سرعت اجرا می شود. چگونه می توان این زمان که باتری در ۱۲ ولت ثابت است به دست آورد؟

کنترل کننده سرعت با تغییر ولتاژ متوسط و فرستاده شده به موتور تنظیم می شود. این موضوع می تواند کار را به سادگی با تنظیم ولتاژ فرستاده شده به موتور انجام دهد، اما این کاملا ناکارآمد است . راه بهتر این است که تغییر عرضه موتور روشن و خاموش بسیار سریع باشد .

این وضعیت *بر روی* MOSFET های قدرت انجام می شود. MOSFET (ترانزیستور اثر میدانی بر، نیمه هادی اکسید فلز) یک وسیله ای است که می تواند جریان های بسیار بزرگ و خارج از تحت کنترل ولتاژ سیگنال را تحمل کند .

زمانی که طول می کشد یک موتور برای سرعت بخشیدن و کاهش سرعت تحت شرایط تغییرموج ورودی وابسته به اینرسی روتور است، و چقدر اصطکاک و گشتاور بار وجود دارد. نمودار زیر نشان می دهد سرعت یک موتور که در حال تبدیل شده است و نسبتا به آرامی خاموش و روشن می شود .



شما می توانید ببینید که سرعت به طور متوسط حدود ۱۵۰ است، با وجود آن که کمی متفاوت است. اگر ولتاژ عرضه شده به اندازه کافی سریع باشد ، زمان زیادی برای تغییر سرعت ندارد، و سرعت کاملا ثابت خواهد ماند .

کنترل سرعت موتور DC با استفاده از PWM

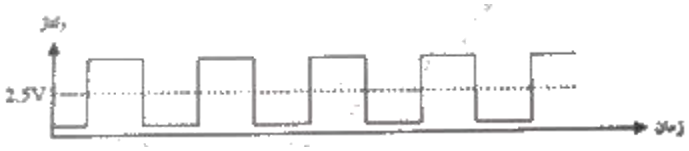
اصولاً برای کنترل سرعت موتور، باید مقدار ولتاژ اعمالی به آن را تغییر داد؛ سرعت موتور با کاهش سطح ولتاژ، کاهش پیدا می کند و با افزایش آن، افزایش می یابد. برای کاهش سطح ولتاژ نیز می توان از یک مقاومت متغیر در مسیر جریان عبوری به سمت موتور استفاده کرد. یکی از معایب اصلی این روش اتلاف انرژی است.

حال اگر به طریقی بتوانیم ولتاژ اعمالی به موتور را با سرعتی یکنواخت قطع و وصل کنیم، با بالاتر رفتن این سرعت از حدی خاص، دیگر قطع و وصل شدن موتور محسوس نخواهد بود؛ ولی موتور به علت قطع و وصل شدن ولتاژ، سطح ولتاژهای متفاوتی را با توجه به سرعت خاموش و روشن شدن احساس خواهد کرد. به این ترتیب سرعت چرخش موتور نیز متناسب با سطح ولتاژ تغییر خواهد کرد که همان هدف اولیه ما است .

قطع و وصل شدن ولتاژ با سرعتی یکنواخت، توسط موج PWMانجام می شود .

حال موج نشان داده شده در شکل ۱ با چرخه کاری ۵۰ درصد را در نظر بگیرید و فرض کنید مدت هر یک از دو زمان وصل و قطع بودن ۵ msاست .به این ترتیب زمان تناوب این موج PWMبرابر با 10msو فرکانش آن برابر است با:

$$\text{فرکانس} = \frac{1}{\text{زمان تناوب}} \rightarrow \frac{1}{10ms} = \frac{1}{10\times 0/001s} = 100\text{Hz}$$



موج PWM با چرخه کاری ۵۰ درصد

چرخه کاری این موج نیز چنین محاسبه می شود:

$$\text{چرخه کاری} = \frac{\text{مدت زمان وصل بودن}}{\text{زمان تناوب}} \quad \rightarrow \frac{5ms}{10ms} \times 100 = \%50$$

توان متوسط موتور برابر با P=VIاست، بنابراین مقدار انرژی مصرفی موتور در مدت زمان tبرابر با W=VITخواهد بود.

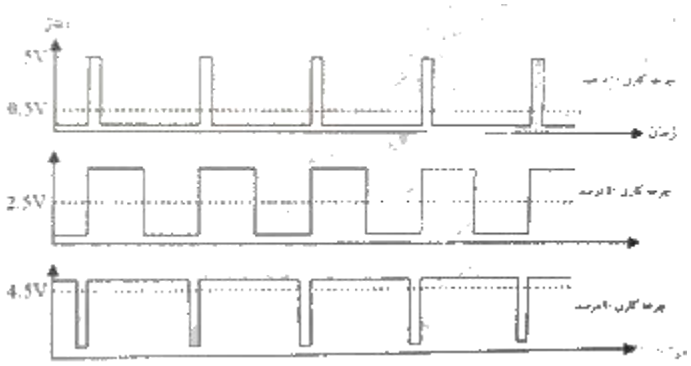
حال فرض کنیم یک موج PWM به موتور اعمال شود و جریان عبوری از آن I=1A باشد، در این صورت اگر مساحت زیر نمودار ولتاژ زمان را در I ضرب کنیم، کل انرژی مصرفی توسط موتور در مدت یک دوره تناوب به دست می آید که برابر خواهد بود با :

W=VIT=5×1×0/005+0×1×0/005=0/025J

این بار اگه فرض کنیم که این ولتاژ به صورت پیوسته به موتور وارد می شود، سطح ولتاژ برابر خواهد بود با:

W= VIT 0/025J=V×1×0/010 V=2/5v

به این ترتیب می توان به راحتی و بدون انجام محاسبات، چرخه کاری موج PWMبرای مثال ۵۰ در صد را در سطح ولتاژ کلی اعمال شدهV ۵ ضرب کرد تا سطح ولتاژی را که موتور احساس می کند 50%×5V=2/5Vبه دست آورد. در شکل زیر، موجهای PWM با چرخه های کاری مختلف و سطح ولتاژ متناسب با هر موج نشان داده شده اند.



موج های PWM با چرخه های کاری مختلف

فصل دوم

تشریح نقشه فنی پروژه وسخت افزار طراحی شده

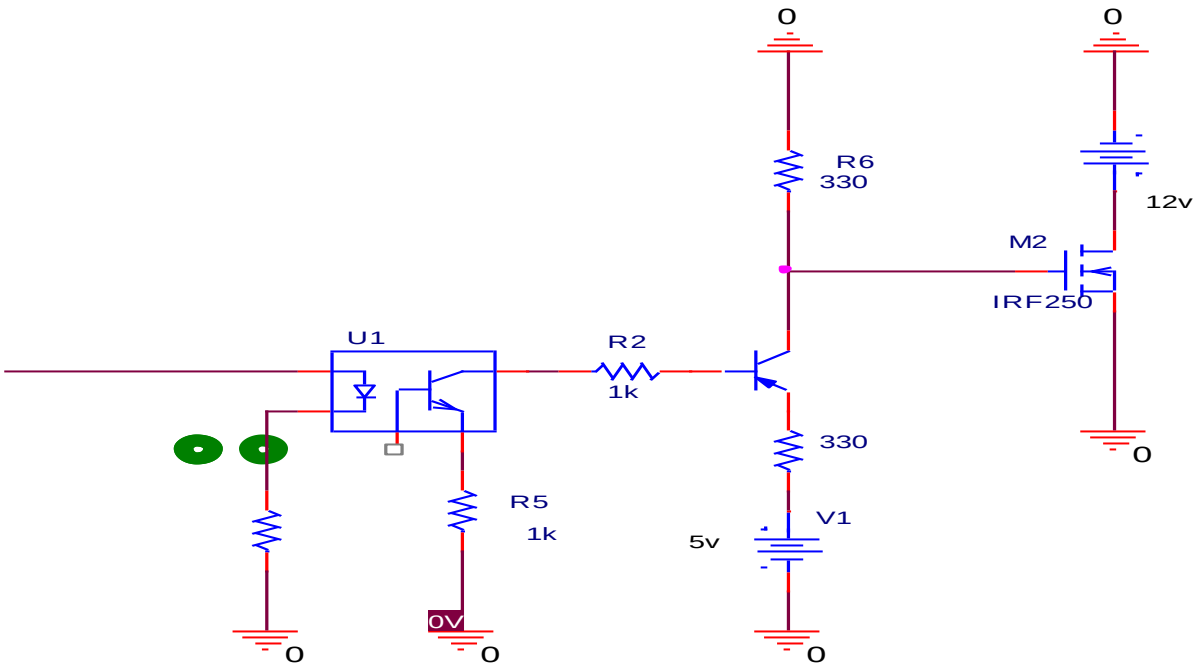
قسمت های مهم سخت افزاری این پروژه به چند بخش می توان تقسیم کرد ، هر مرحله را بطور مجزا بررسی خواهیم کرد و در اخر نقشه فنی طراحی شده توسط نرم افزار پرتل ونیز مدار شبیه سازی شده ی ان را بررسی می کنیم.

مدار درایور موتور:

از جمله قسمتهای سخت افزاری نام برده شده، مهمترین طراحی مربوط به درایور موتور می شود چرا که در این پروژه سعی شده به جای استفاده از ای سی های اماده ی درایور موتور از طراحی مدارات تقویت کننده الکترونیکی استفاده شود.

مدار درایور موتور وظیفه دارد ولتاژ PWM تولید شده توسط میکرو را به ولتاژ 12v وجریان 3A تبدیل کند با توجه به این نکته که مدار تقویت کننده در ناحیه ی غیر خطی بایاس شده است زیرا ولتاژ PWM ترانزیستورها را به ناحیه ی قطع و اشباع می برند و بطور کلی می توان به ترانزیستور ها بعنوان سوییچ های اکتیو نگاه کرد.

تقویت ولتاژ کار بسیار ساده ای است بدلیل اینکه دو منبع ولتاژ dc در مدار وجود دارد اما قسمت مهم کارایجاد جریان لازم برای موتوراست که برای این منظورازترانزیستور ماسفت با قدرت جریان دهی بالا (irf250) استفاده شده است.

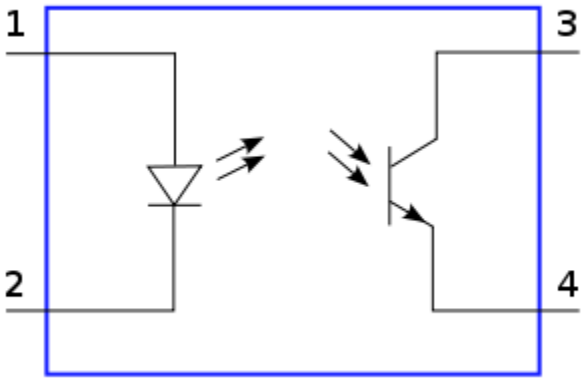


قطعات استفاده شده:

4n27 : جهت ایزوله کردن ودار موتور از مدار میکرو از یک اپتو کوپلر استفاده شده است .

OPTOCOUPLER یک قطعه الکترونیکی است که به صورت IC تولید میشود. کار اصلی اپتوکاپلر ایزوله کردن دو نقطه از مدار با استفاده از نور میباشد. همانطور که از نام آن مشخص است، وظیفه آن کوپل کردن یا اتصال دو نقطه از طریق نور میباشد، به عنوان مثال اگر بخواهیم از طریق میکرو مستقیما به موتوری فرمان بدهیم یا رله ای را فعال کنیم ممکن است عمکرد موتور باعث ایجاد نویز شده و بر روی عملکرد میکرو یا مدار فرمان تاثیر بگذارد و باعث اختلال در سیستم شود.

یکی از کاربرد های مهم این قطعه در کنترل دیجیتالی قدرت میباشد. از انواع اپتوکوپلر میتوان به نوع ترانزیستوری، دارلینگتون و دیاک اشاره کرد.



همانطور که در شکل های بالا مشخص است optocoupler از یک LED برای ارسال نور و یک فوتوترانزیستور برای دریافت نور استفاده میکند.

انواع اپتوکوپلر

۱-اپتوکوپلر ترانزیستوری: این قطعه از یک LED و یک ترانزیستور نوری تشکیل شده است که با کم و زیاد کردن ولتاژ پایه های میتوان بیس ترانزیستور را تحریک کرد.علاوه بر تغییر ولتاژ از موج برای تغییر نور نیز استفاده کرد .

۲- اپتوکوپلر دارلینگتون: دراین قطعه به جای ترانزیستور یک جفت دارلینگتون در مدار قرار دارد .

۳- اپتوکوپلر دیاک:از این قطعه برای تریگر کردن ترایاک استفاده می شود .

4n5401 : یک ترانزیستور bjt وبصورت pnp است ، و بعنوان طبقه ورودی تقویت کننده است.

پایه های این ترانزیستور همانند شکل زیر است.

4n5401

Irf 250 : این ماسفت جهت تامین جریان لازم برای موتور انتخاب شده است .ودر طبقه خروجی قرار دارد . این ماسفت وظیفه ی تبدیل ولتاژ 5v به 12v را نیز دارد.

موتور بر روی درین قرار گرفته و جریان لازم را از ماسفت قدرت می کشد.

قسمت مهم دیگر سخت افزاری ماژول های فرستنده و گیرنده (HM-TR) است که مشخصات و پایه های ان را بررسی می کنیم.

HM-TR

یکی از ماژول های فرستنده وگیرنده دیتا با قابلیت اتصال به کامپیوتر که با کمک آن میتوان براحتی اطلاعات را به صورت بی سیم بین دو میکرو یا بین دو کامپیوتر یا میکرو و کامپیوتر انتقال داد ماژول HM-TR میباشد. طبق مشخصات این ماژول جز خانواده برد 500 متر میباشد و طبق دیتا شیت ان در صورتی که ماژول حداقل ۱ متر از زمین فاصله داشته باشد اطلاعات را تا فاصله 230 متر ارسال میکند.

نحوه عملکرد ماژول بدین صورت میباشد که در صورتی که پایه های ان به صورتی به تغذیه وصل شده باشد که در حالت ارسال ودریافت اطلاعات باشد ابتدا ماژول اطلاعاتی که به پایه drx آن وصل شده باشد را ارسال میکند وبعد از اتمام ارسال به صورت اتوماتیک به حالت گیرنده سوئیچ میشود و در صورتی که اطلاعاتی باشد آن را دریافت میکند و از طریق پایه dtx قابل دریافت خواهد بود و مدت زمان سوئیچ بین حالت فرستنده و گیرنده 50 میلی ثانیه میباشد.

پروتکل و نحوه عملکرد مدار در ارسال اطلاعات به صورت fsk می باشد و با توجه به نوع فرکانس کاری و برد مدار نیاز به مجوز ندارد .
از دیگر ویژگی های بارز مدار تعیین فرکانس کاری ارسال و دریافت اطلاعات ، نرخ ارسال اطلاعات ، تغییر رنج ارسال اطلاعات در یک رنج وسیع و دیگر مشخصات فرستنده توسط نرم افزار مخصوص این ماژول می باشد .
فرکانس کاری این ماژول در انواع 315 و 434 و868 و915 مگاهرتز می باشد ، البته درهنگام خرید انتن باید با توجه به فرکانس ماژول انتن را خریدرای کرد.
پروتکل ارسال ودریافت اطلاعات به صورت اتوماتیک توسط خود ماژول کنترل می شود، همچنین این ماژول از حساسیت بالایی در میزان اطلاعات دریافتی و ارسالی دارد.
مهمترین ویژگی این ماژول تبادل اطلاعات با کامپیوترو میکرو از طریق پروتکل UART سریال می باشد که میتوان به صورت TTL یا rs232 تبادل اطلاعات را انجام داد .
همانطور که در بالا گفته شد نوع تبادل اطلاعات در سریال به ۲ صورت rs232 و TTL می باشد درضمن برروی ماژول میکروکنترلر atmega16 نصب شده است اگر بخواهیم ماژول را به کامپیوتر وصل کنیم باید از نوع rs232 آن استفاده کنیم یا این که خودمان یک آی سی max232 به مدار اضافه کنیم .
اما کارخانه سازنده ۲ نوع ماژول ارائه داده است که در یکی آی سی max232 نصب شده است (مانند شکل زیر)

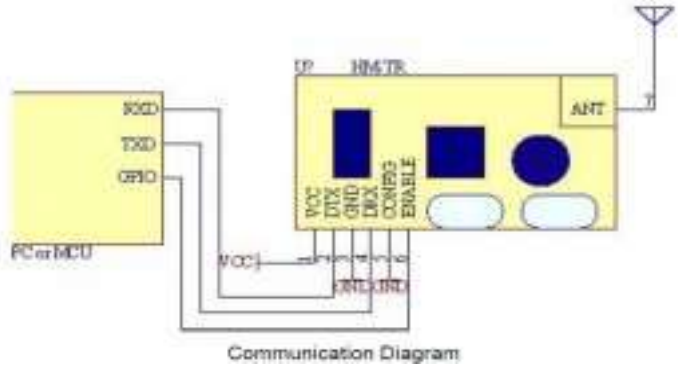
اما مدل دیگر ان TTL می باشد که هیچ تفاوتی با نمونه بالا ندارد مگر این که آی سی max23حذف شده و به دیتا به صورت TTL تبادل میشود که در این صورت به راحتی می توان به صورت مستقیم به میکرو وصل کرد.

اما اگر ماژول از نوع TTL را خریداری شود. اگر خواستید به صورت rs232 نیز مستقیما عمل کند میتوان ای سی max232 و خازن و قعطات smd انرا تهیه کرد و برروی ماژول لحیم کرد یا جداگانه در یک برد دیگری ان را ساخته و به ماژول اصلی وصل کرد اما اگرماژول با خروجی rs232 را تهیه کنید هم خروجی rs232 را دارید هم TTL .

به این صورت که اگر ماژول با خروجی rs232 را تهیه کنید براحتی با اتصال ۲ نقطه مدار به یکدیگر می توانید ان را به خروجی TTL تبدیل کنید.

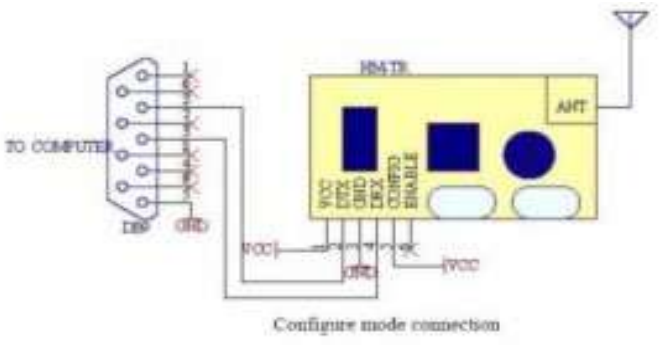
برای تبدیل خروجی از rs232به TTL باید مانند شکل زیر ۲ پایه تعیین شده را به هم لحیم کرد .

اما در مورد پایه های این ماژول ونحوه اتصال آن به کامپیوترو میکرو بدین صورت میباشد که در شکل زیر نمایش داده شده است .
در صورتی که ماژول را روبرو خود بگیرید ، وظیفه پایه ها از سمت چپ به راست به ترتیب به صورت زیر میباشد
VCC=پایه تغذیه مثبت مدار میباشد که باید 5 ولت به آن وصل کرد.
DTX=درصورتی که بخواهیم اطلاعات ارسال شده از فرستنده دیگر را دریافت کنیم ، اطلاعات دریافتی توسط ماژول از طریق این پایه دردسترس خواهد بود .
GND=این پایه برای تغذیه منفی مدار میباشد که باید GND و یا به عبارت دیگر منفی تغذیه را به آن وصل کرد .
DRX=درصورتی که بخواهیم اطلاعاتی را ارسال کنید باید اطلاعات به این پایه داده شود.
CONFIG=این پایه برای تنظیمات داخلی ماژول می باشد،درصورتی که خواسته باشید به تنظیمات داخلی ماژول توسط نرم افزار دسترسی داشته باشید باید این پایه را به مثبت VCCوصل کنید وبعد ماژول را روشن کنید اما اگر خواسته باشید ماژول به صورت عادی عمل کند و اطلاعات را ارسال و دریافت کند این پایه باید به زمین GND وصل شده باشد.
ENABLE=پایه فعال ساز ENABLE ماژول میباشد در صورتی که این پایه به زمین VCCوصل شده باشد ماژول روال عادی خود را انجام میدهد اما اگر خواستید ماژول هیچ دیتای را ارسال و یا دریافت نکند این پایه باید به GND وصل شود.
درشکل زیر یک نمونه ساده اتصال ماژول به میکرو را مشاهده میکنید البته حتما نیاز نیست پایه فعال ساز هم به میکرو وصل شده باشد و میتوانید ان را مستقیما به مثبت 5ولت وصل کنید.



Communication Diagram

همانطور که گفته شد از مشخصات بارز این ماژول قابلیت اتصال مستقیم به کامپیوتر از طریق پورت سریال com کامپیوتر میباشد، مثلا می توانید در ساده ترین روش اطلاعات دریافتی را توسط hyper terminal ویندوز مشاهده کنید.
اما در ابتدای متن گفتیم که میتوان مشخصات داخلی این ماژول را تغییر داد برای این کار باید درابتدا ماژول را مانند عکس زیر به پورت com سیستم خود وصل کنید.



Configure mode connection

قبلا گفته شد باید با توجه به فرکانس ماژول، آنتن ان را نیز خریدرای کنید درضمن حتما باید آنتن به ماژول وصل باشد تا ماژول کار کند و اطلاعات را ارسال و دریافت کند.

بررسی پایه های lcd :

LCDهای کاراکتری که عموما برای مشاهده خروجی ها، زمان کار با میکرو کنترلرها بکار می روند در اندازه های مختلف موجود هستند که این اندازه نشان دهنده تعداد کارکترهای LCD در سطر و ستون می باشد .
مثلا 16در ۲ ، ١٦ کاراکتر در هر سطر و دو سطردارد یا۲۰ در۴ ،۲۰ کاراکتر درهر سطر و ٤ سطر دارد .

در حالی که LCD های کاراکتری دراندازه های مختلف موجود هستند اما بطور عموم ١٦ پایه برای اتصال دارند.

اتصال پایه ها به ترتیب به شکل زیر است:

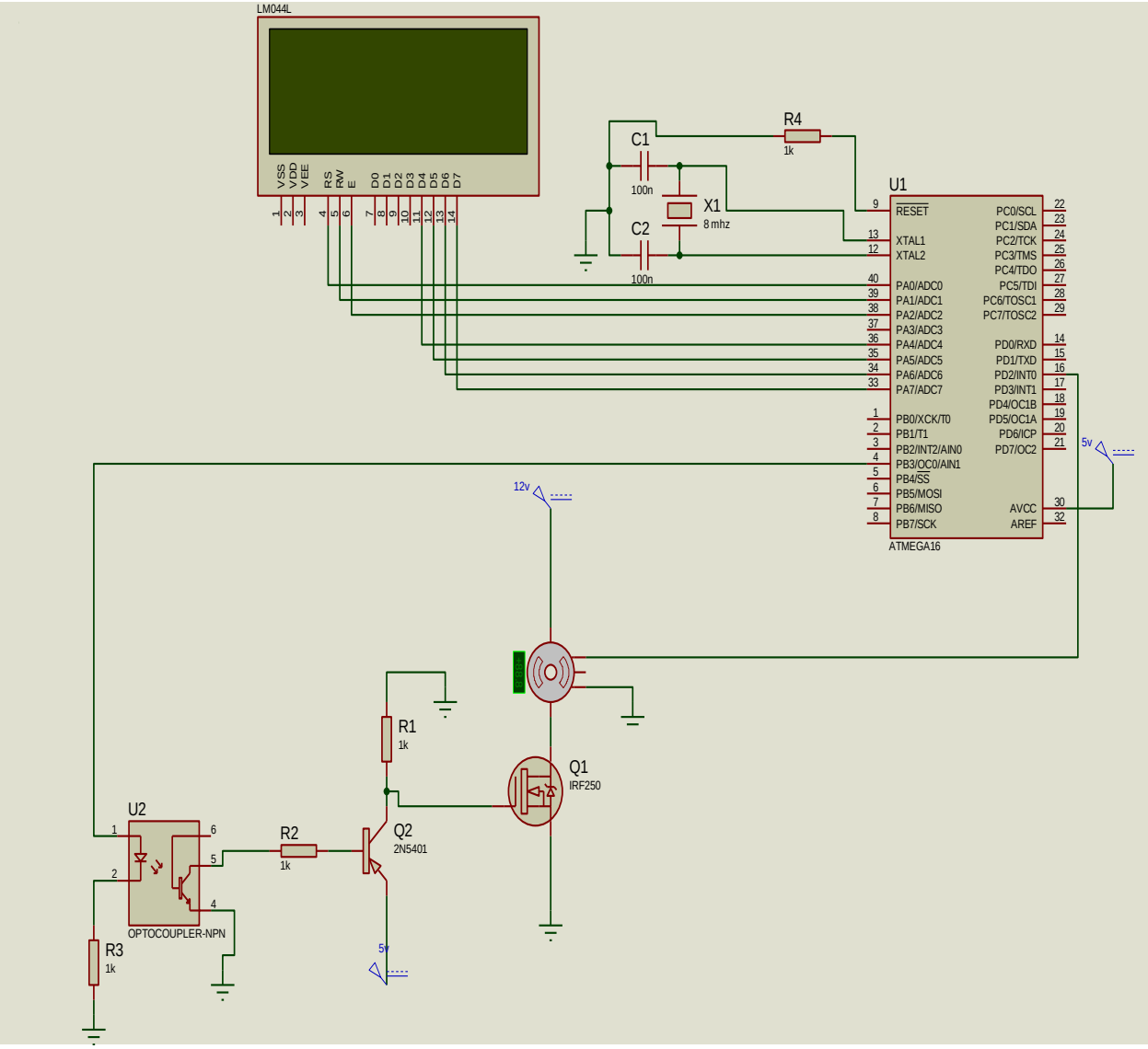
١- پایه اتصال به زمین
٢- پایه ٥ ولت
٣-پایه مقدار کنتراست کاراکترهای LCD میباشد با یک مقاومت ٠ الی ٥ کیلو به زمین متصل شود مقاومت ٤.٧ کیلوبایت کنتراست مناسبی در LCD ایجاد میکند
٤- پایه RS اتصال به میکرو
٥-پایه در صورت ١ بودن برای خواندن از LCD و صفر بودن برای نوشتن در LCD که بطور عموم این پایه اتصال به زمین(صفر (است
٦-پایه Enable اتصال به میکرو جهت فعال یا غیر فعال کردنLCD
٧- پایه DB0گرفتن Data ازLCD
٨- پایه DB1 گرفتن Data ازLCD
٩- پایه DB2 گرفتن Data ازLCD
١٠- پایه DB3 گرفتن Data ازLCD
١١- پایه DB4 رسال Data به LCD اتصال به میکرو
١٢- پایه DB5 ارسال Data به LCDاتصال به میکرو
١٣- پایه DB6 ارسال Data به LCD اتصال به میکرو
١٤- پایه DB7 ارسال Data به LCD اتصال به میکرو

١٥- اتصال مثبت LED پشت زمینه LCD میتوان به میکرو متصل کرد تا روشن

وخاموش کردن ان قابل کنترل باشد یا به ٥ ولت تا همیشه روشن باشد
١٦- اتصال زمین LED پشت زمینهLCD

پایه های ٧ الی ١٠ برای خواندن اطلاعات از LCD استفاده میشوند و چون معمولا اینکار انجام نمیشود این پایه ها به میکرو متصل نمیشوند

مدار شبیه سازی شده پروژه بصورت زیر توسط نرم افزار پروتوس انجام شده است



این پروژه از دو برد مدار چاپی استفاده شده که مدار pcb ان در این قسمت اورده شده است .

فصل سوم

تشریح نرم افزار و برنامه های مربوط به پروژه

می توان گفت مهمترین قسمت پروژه مر بوط به نرم افزار ان می باشد ، کنترل pid که در قسمت مقدمه با طرز کار ان تا حدودی اشنا شدیم در اینجا نرم افزاری پیاده می کنیم .تفاوت مهم پیاده سازی از طریق نرم افزاری با سخت افزاری در این است که محاسبات تناسبی ، انتگرالی، مشتقی در این حالت بصورت عددی هستند.

ابتدا نگاهی به برنامه کلی می اندازیم پس ازاینکه با مفاهیم کلی برنامه اشنا شدیم به بررسی تک تک زیر برنامه ها و خط به خط برنامه می پردازیم.

برنامه کامل پروژه:

```
**************************************************/

This program was produced by the

CodeWizardAVR V1.24.8b Professional

Automatic Program Generator

©Copyright 1998-2006 Pavel Haiduc, HP InfoTech s.r.l.

http://www.hpinfotech.com

Chip type           : ATmega16

Program type        : Application

Clock frequency     : 8.000000 MHz

Memory model        : Small

External SRAM size  : 0

Data Stack size     : 256

/**************************************************

#include <mega16.h>

#include <delay.h>
```

```
 //Alphanumeric LCD Module functions
#asm
.   equ __lcd_port=0x1B ;PORTA
#endasm
#include <lcd.h>
#include <stdlib.h>
#include <stdio.h>
//calculation revving parameter
unsigned int m@0x60;
unsigned char ring@0x62;
unsigned long int sum@0x63 ;
unsigned long int tesum@0x67;
bit ok;
bit send_ok;
float sensor@0x6c;
//pid parameter
signed int error@0x71;
unsigned char setpoint@0x73;
float temp@0x79;
int temp2@0x7d;
unsigned char sender@0x85;
```

```
float output;
float integral=0;
float derivative;
int pre_error;
//pid constant
#define kp  1
#define ki  1
#define kd  0.01
#define dt 0.005
char lcdbuf [20];
char lcd [20];
interrupt [USART_RXC] void usart_rx_isr(void)
{
unsigned char data;
data=UDR;
setpoint=data;
send_ok=1 ;  }
interrupt [EXT_INT0] void ext_int0_isr(void(
{
m=TCNT1;
TCNT1=0;
```

```
ring;++
if(ring<=5 }(
        sum=sum+m;
        ok=0;
}
Else
{
   tesum=sum;
   tesum=tesum/5;
//   time one cycle
   tesum=tesum*28;
//   rps
   sensor=8000000/tesum;
   ok=1;
   ring=0;
   sum=0;
} }
void main(void)
{
//Input/Output Ports initialization
//Port A initialization
```

```
//Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In

//State7=T State6=T State5=T State4=T State3=T State2=T
State1=T State0=T

PORTA=0x00;

DDRA=0x00;

//Port B initialization

// Func7=In Func6=In Func5=In Func4=In Func3=Out Func2=In
Func1=In Func0=In

//State7=T State6=T State5=T State4=T State3=0 State2=T
State1=T State0=T

PORTB=0x00;

DDRB=0x08;

//Port C initialization

//Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In

//State7=T State6=T State5=T State4=T State3=T State2=T
State1=T State0=T

PORTC=0x00;

DDRC=0x00;

//Port D initialization

//Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In
Func1=In Func0=In
```

```
//State7=T State6=T State5=T State4=T State3=T State2=T
State1=T State0=T

PORTD=0x00;

DDRD=0x00;

//Timer/Counter 0 initialization

//Clock source: System Clock

//Clock value: 8000.000 kHz

//Mode: Fast PWM top=FFh

//OC0 output: Non-Inverted PWM

TCCR0=0x6A;

TCNT0=0x00;

OCR0=0x99;

 //Timer/Counter 1 initialization

 //Clock source: System Clock

 //Clock value: 8000.000 kHz

 //Mode: Normal top=FFFFh

 //OC1A output: Discon.

//OC1B output: Discon.

//Noise Canceler: Off

//Input Capture on Falling Edge

//Timer 1 Overflow Interrupt: Off
```

```
//Input Capture Interrupt: Off
//Compare A Match Interrupt: Off
//Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x01;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
//External Interrupt(s) initialization
//INT0: On
//INT0 Mode: Rising Edge
//INT1: Off
//INT2: Off
GICR|=0x40;
MCUCR=0x03;
MCUCSR=0x00;
```

```
GIFR=0x40;
//USART initialization
//Communication Parameters: 8 Data, 1 Stop, No Parity
//USART Receiver: On
//USART Transmitter: On
//USART Mode: Asynchronous
//USART Baud Rate: 9600
UCSRA=0x00                    ;
UCSRB=0x98;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x33 ;
 //Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;
//Analog Comparator initialization
//Analog Comparator: Off
//Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;
//LCD module initialization
lcd_init(20;(
```

```
// Global enable interrupts
#asm("sei")
setpoint=35;
while (1)
{
                sender=sensor;
                putchar(sender);
     if(ok==1){
            ok=0;
//.................calculate error
            error=setpoint-sensor;
            if(error>0){
                    integral=integral+(error*dt);
                    derivative=(error-pre_error)/dt;
                    output=kp*error+ki*integral+kd*derivative ;
                    temp=output*4.2;
                    OCR0=temp;
                    pre_error=error;
}
            Else{
                integral=integral+(error*dt);
```

```
                derivative=(error-pre_error)/dt;
                 output=kp*error+ki*integral+kd*derivative ;
                 temp=output*(4.2);
                temp2=temp;
                OCR0=temp2;
                pre_error=error;
}
            lcd_clear();
            lcd_gotoxy(0,0);
            lcd_putsf("Setpoint”);
            lcd_gotoxy(9,0);
            itoa(setpoint,lcdbuf
        lcd_puts(lcdbuf);
--------------------//
            lcd_gotoxy(0,1);
            lcd_putsf("Sensor”);
            lcd_gotoxy(7,1);
            ftoa(sensor,0x02,lcdbuf);
            lcd_puts(lcdbuf)
----------------------//
            lcd_gotoxy(0,2);
```

```
            lcd_putsf("Error:”);
            lcd_gotoxy(6,2);
            ftoa(error,0x02,lcdbuf);
            lcd_puts(lcdbuf)
----------------------//
            lcd_gotoxy(13,0);
            lcd_putsf("Kp=”);
            lcd_gotoxy(16,0);
            ftoa(kp,0x02,lcdbuf);
            lcd_puts(lcdbuf)

            lcd_gotoxy(13,1);
            lcd_putsf("Ki=”);
            lcd_gotoxy(16,1);
            ftoa(ki,0x02,lcdbuf)
            lcd_puts(lcdbuf);

            lcd_gotoxy(13,2);
            lcd_putsf("Kd=”);
            lcd_gotoxy(16,2);
            ftoa(kd,0x02,lcdbuf);
```

```
            lcd_puts(lcdbuf);
----------------------//
}
};
}
```

چگونه اطلاعات انکدر را توسط میکرو بخوانیم؟

شمارش تعداد دور موتور که از مهمترین قسمت های این پروژه است بدلیل اینکه فرمان کنترولی صادر شده توسط نرم افزار بر مبنای شمارش موتور می باشد. به همین دلیل برای داشتن یک فرمان کنترولی دقیق برای رسیدن به هدف نیاز به اطلاعات دقیق از دور موتور داریم که در ادامه به بررسی نحوه ی شمارش موتور با این دقت می پردازیم.

شمارش موتور توسط میکروکنترولر انجام میگیرد به این صورت که توسط یک انکدر نوری که روی شفت موتور تعبیه شده و دارای ۲۸ شکاف نوری است دقت خوبی را تضمین میکند . الگوریتم شمارش به این گونه است که از وقفه های خارجی میکروکنترولر AT mega 16 استفاده شده که با هر لبه ی بالا رونده ی موج تشکیل شده توسط انکدر نوری یک وقفه از میکرو کنترولر گرفته می شود و همزمان تایمر میکروکنترولر عمل شمارش را در حالت مد نرمال انجام می دهد و عملیات شمارش را با امدن وقفه متوقف می کند و در زیر برنامه ی وقفه برنامه ی زیر اجرا می شود.که در ادامه به بررسی خط به خط ان می پردازیم.

بررسی زیر برنامه وقفه:

```
interrupt [EXT_INT0] void ext_int0_isr(void)
{
m=TCNT1;
TCNT1=0;
Ring++;
if(ring<=5)
{
        sum=sum+m;
```

```
        ok=0;
}
else
{
   tesum=sum;
   tesum=tesum/5;
//   time one cycle
   tesum=tesum*28;
//   rps
   sensor=8000000/tesum;
ok=1;
ring=0;
sum=0;
  }
}
```

ابتدا مقدار رجیستر TCNT1 را در متغییر m قرار می دهیم ، سپس بلا فاصله مقدار این رجیستر را صفر می کنیم تا برای مرحله بعدی شمارش را از صفر شروع کند.

برای اینکه تعداد پالسها قابل شمارش باشد توسط تایمر میکرو کنترولر مدت زمان ۵ پالس را باهم جمع کردیم سپس در ادامه جهت محاسبه ی دور در ثانیه زمان بدست امده را بر ۵ تقسیم خواهیم کرد.

متغییر sum زمان مجموع ۵ پالس است که درون متغییر tesum قرار می دهیم سپس بر عدد ۵ تقسیم کرده تا مدت زمان یک پالس بدست اید. چرخش یک دور موتور تعداد ۲۸

پالس ایجاد می کند (تعداد شکافهای انکدر) در ادامه مدت زمان یک پالس را در عدد ۲۸ ضرب می کنیم تا مدت زمان یک دور موتور بدست اید .

متغییر sensor مدت زمان یک دور در ثانیه را نشان می دهد به اینگونه که تایمر میکرو با فرکانس ۸ مگا هرتزکارمی کند یعنی دریک ثانیه ۸۰۰۰۰۰۰ پالس کلاک جهت شمارش دریافت می کند حال با تقسیم این عدد بر tesum که مدت زمان یک دور موتوراست تعداد دور های موتور در یک ثانیه بدست می اید.

پس از بررسی زیر برنامه وقفه میکروبه بررسی زیر برنامه usart می پردازیم

ابتدا لازم است توضیحاتی در رابطه با قسمت usart میکرو کنترلر AT-mega16 بدهیم سپس چند خط برنامه را بررسی کنیم.

عملکرد USART میکروکنترولر

قطعه ي ATmega16 داراي يك ماژول USART بوده كه از استاندارد RS-232 در دو حالت آسنكرون و سنكرون پشتيباني مي كند . دسترسي به پورت سريال AVR از طريق سه پين XCK RXD TXD كه به ترتيب پين ارسال، دريافت وكلاك مي باشند امكان پذير است.(پين xck فقط درمد سنكرون كاربرد دارد)

قالب اطلاعات در USART ميكروكنترلر AVR همانند UART كامپيوترهاي شخصي شامل يك بيت شروع بيت هاي داده، بيت اختياري توازن و يك يا دو بيت پايان است، با اين تفاوت كه در AVR مي تواند 5 تا 9 بيت Data تعريف شود . بيت توازن مي تواند فرد يا زوج باشد و از طريق بيت هاي[1:0]UPM از رجيسترUCSRC تنظيم مي شود.

بررسی رجيسترهاي USART (AT-mega16)

1 (UDR) USART Data Register

بافر دريافت و ارسال پورت سريال داراي يك آدرس مشترك به نام UDR در فضاي I/O Registers مي باشند. بافر ارسال، مقصد داده هاي نوشته شده در رجيسترUDR بوده و خواندن اين رجيستر محتويات بافر دريافت را به دست مي دهد . تنها زماني مي توان روي رجيستر UDR مقداري را نوشت كه بيت UDRE از رجيستر UCSRA يك شده باشد و در غير اينصورت ديتاي ارسالي توسط USART نا ديده گرفته مي شود . با ارسال اطلاعات به بافر ارسال USART در صورتي كه بيت TXEN از رجيسترUCSRB يك باشد اطلاعات در شيفت رجيستر بارگذاري شده و بيت به بيت از پين TXD ارسال مي شود.

٢ USART Control and Status Register A

UCSRA	٧	٦	٥	٤	٣	٢	١	٠
نام بيت	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM

Multi-processor Communication Mode: يك شدن اين بيت ميكروكنترلر را به حالت ارتباطات چند پردازنده اي مي برد.

Double the USART Transmission Speed: با يك كردن اين بيت در Mode آسنكرون BAUD RATE دو برابر خواهد شد .در Mode سنكرون اين بيت بايد صفر باشد.

Parity Error: در صورت فعال بودن توليد بيت توازن از طريق بيت هاي UPM[1:0]، با روي دادن خطاي توازن در بافر دريافت، اين بيت يك مي شود.

Data Overrun: با بروز Overrun اين بيت يك مي شود .شرايط Overrun يا لبريز وقتي روي مي دهد كه بافر دريا فت پر باشد و شيفت رجيستر نيز محتوي داده ي جديدي باشد و داده ي جديدي نيز از راه برسد، يعني يك بايت در بافر شيفت رجيستر منتظر باشد و بيت شروع جديدي دريافت شود . در اين حالت اطلاعات جديد از بين مي رود و با يك شدن بيت DOR بروز خطا اعلام مي شود.

Frame Error: اگر در Frame دريافت شده بيت پايان صفر باشد اين بيت يك شده و در غير اينصورت صفر خواهد بود.

USART Data Register Empty: يك بودن اين پرچم نشان دهنده ي اين است كه اطلاعات موجود در بافر ارسال براي شيفت رجيستر ارسال شده و بافر ارسال آماده ي دريافت كاراكتر جديد است . همچنين در صورتي كه بيت UDRIE از رجيستر UCSRB يك باشد باعث ايجاد وقفه شده و تا زماني كه بيت UDRE يك است با خارج شدن از ISR دوباره آن را اجرا مي كند **با نوشتن داده ي جديد در UDR پرچم UDRE صفر مي شود** . بعد از ريست شدن ميكرو اين بيت يك مي شود كه به معناي آماده بودن دريافت كاراكتر جديد است.

USART Transmit Complete: اين پرچم زماني يك مي شود كه تمام اطلاعات موجود در شيفت

رجيستر به بيرون شيفت داده شده و داده ي جديدي در بافر ارسال وجود نداشته باشد .با فعال بودن بيت TXCIE اين پرچم مي تواند باعث ايجاد وقفه ي كامل شدن ارسال شود

USART Receive Complete: اين بيت با كامل شدن دريافت يك Frame در UDR يك شده و **پس از خواندن UDR صفر می شود.**

۳- USART Control and Status Register B

UCSRB	7	6	5	4	3	2	1	0
نام بيت	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8

Transmit Data Bit 8: در حالتي كه از پورت سريال در Mode 9 بيتي استفاده مي شود اين بيت نهمين بيت كاراكتر ارسالي خواهد بود . بايد توجه داشت كه قبل از نوشتن در UDR بايد وضعيت اين بيت را مشخص كرد.

Receive Data Bit 8:در حالتي كه از پورت سريال درMode ۹ بيتي استفاده مي شود اين بيت نهمين بيت كاراكتر دريافتي خواهد بود .بايد توجه داشت كه قبل از خواندن UDR بايد اين بيت را خواند.

Character Size:با تركيب اين بيت و بيت هاي [1:0]UCSZ در رجيستر UCSRC تعداد بيت هاي داده را در يك Frame مشخص مي كند.

Transmitter Enable: با يك كردن اين بيت عملكرد عادي پينTxD به ارسال پورت سريال تغيير حالت داده و بعد از آن قابل استفاده به صورت I/O معمولي نيست . غير فعال كردن اين بيت د ر حاليكه UART سريال مشغول است تا اتمام ارسال اطلاعات تاثير نخواهد گرفت.

Receiver Enable: با يك كردن اين بيت عملكرد عادي پينRxD به دريافت پورت سريال تغيير حالت داده و بعد از آن قابل استفاده به صورت I/O معمولي نيست . با صفر كردن اين بيت بافر پورت سريال خالي شده و مقدار بيت هاي PEو FE، DOR نامعتبر خواهد بود

USART Data Register Empty Interrupt Enable:با يك شدن اين بيت، در صورتي كه بيت فعال ساز كلي وقفه ها (I) يك باشد، فعال شدن پرچم خالي بودن بافر ارسال (UDRE) مي تواند باعث ايجاد وقفه شود.

TX Complete Interrupt Enable:با يك شدن اين بيت، در صورتي كه بيت فعال ساز كلي وقفه ها (I) يك باشد، فعال شدن پرچم اتمام ارسال (TXC) مي تواند باعث ايجاد وقفه شود.

RX Complete Interrupt Enable: با يك شدن اين بيت، در صورتي كه بيت فعال ساز كلي وقفه ها (I) يك باشد، فعال شدن پرچم اتمام ارسال (RXC) مي تواند باعث ايجاد وقفه شود.

٤- USART Control and Status Register C

UCSRC	7	6	5	4	3	2	1	0
نام بیت	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL

ابتدا زیر برنامه مربوط به USART را بررسی خواهیم کرد سپس به رجیسترهای ان می پردازیم .

در این برنامه از وقفه ی دریافت اطلاعات در USART استفاده شده است بدلیل اینکه مقدار setpoint ازطریق کامپیوتر وارد برنامه می شود در هر وقفه این مقدار بررسی می شود واگر تغییر کرده باشد مقدار جدید جایگزین می شود.

بررسی زیر برنامه مربوط به وقفه ی دریافت USART:

```
interrupt [USART_RXC] void usart_rx_isr(void)
{
unsigned char data;
data=UDR;
setpoint=data;
send_ok=1;
}
```

همانطور که در ابتدا بررسی شد UDR بافر دریلفت یا ارسال اطلاعات است ، در این قسمت ما مقدار UDR را درون یک متغییر بنام DATA قراردادیم .سپس این اطلاعات را

درون setpoint برنامه قرار دادیم ، به این طریق اطلاعات دریافت شده که همان مقدار setpoint است، را درون برنامه قرار می دهیم.

رجیستر های USART بگونه ای پیکر بندی شدن که مشخصات زیر بدست امده است ، که تک تک بیت های ان را در بالا بررسی کرده ایم

```
//USART initialization
//Communication Parameters: 8 Data, 1 Stop, No Parity
//USART Receiver: On
//USART Transmitter: On
//USART Mode: Asynchronous
//USART Baud Rate: 9600
UCSRB=0x98;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x33 ;
```

پس از بررسی زیر برنامه ها به MAIN برنامه می رسیم که در ان رجیستر های مربوط به تایمرها و USART و LCD و ورودی یا خروجی بودن پورت ها تعیین شده اند .

پس از بررسی USART به قسمت TIMER برنامه می رسیم که وظیفه ی اصلی ان شمارش تعداد دور موتور در یک ثانیه است .

رجیستر های تایمر ها بگونه ای پیکر بندی شده اند که تایمر در حالت مد نرمال FFFF شمارش می کند و تایمر صفر در مد fast pwm موج مربعی کنترلی موتور را تولید می کند .

در قسمت ضمیمه تمامی رجیستر های تایمر ها بررسی شده اند.

پیکر بندی تایمر صفر جهت تولید موج pwm

```
//Timer/Counter 0 initialization
//Clock source: System Clock
//Clock value: 8000.000 kHz
//Mode: Fast PWM top=FFh
//OC0 output: Non-Inverted PWM
TCCR0=0x6A;
TCNT0=0x00;
OCR0=0x99;
```

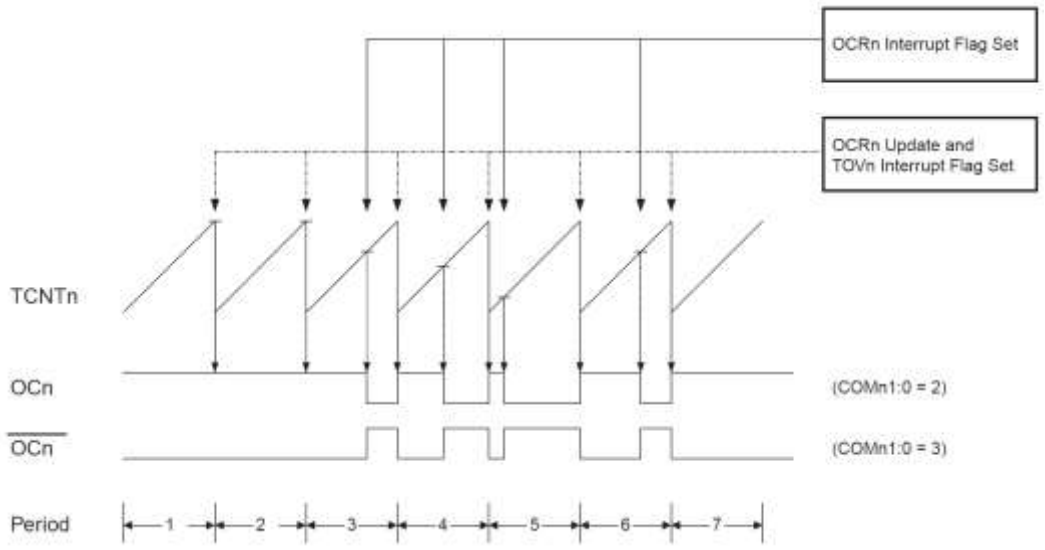
۱) تولید PWM از طریق تایمر صفر:

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 ۴ در ATmaga 16) را not کرده و به شمارش خود ادامه می دهد تا به مقدار 0×ff 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 non-inverted , inverted به ترتیب

خروجی های معکوس و غیرمعکوس 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:

$$FOC\circ = \frac{fclk - I/O}{N.(256 - TCNT\circ)}$$

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.

پیکر بندی تایمر یک جهت کار در مد normal ffff

```
//Timer/Counter 1 initialization
//Clock source: System Clock
//Clock value: 8000.000 kHz
//Mode: Normal top=FFFFh
//OC1A output: Discon.
//OC1B output: Discon.
//Noise Canceler: Off
//Input Capture on Falling Edge
//Timer 1 Overflow Interrupt: Off
//Input Capture Interrupt: Off
//Compare A Match Interrupt: Off
//Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x01;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
```

```
OCR1BH=0x00;
OCR1BL=0x00;
```

قسمت while برنامه وظیفه محاسبات کنترل pid را برعهده دارد.

محاسبه ی pid:

```
while (1){
                sender=sensor;
                putchar(sender);
    if(ok==1){
            ok=0;
//            calculate error
            error=setpoint-sensor;
            if(error>0){
                    integral=integral+(error*dt);
                    derivative=(error-pre_error)/dt;
                    output=kp*error+ki*integral+kd*derivative ;
                    temp=output*4.2;
                    OCR0=temp;
                    pre_error=error;
}
            Else{
```

```
integral=integral+(error*dt);
derivative=(error-pre_error)/dt;
 output=kp*error+ki*integral+kd*derivative ;
 temp=output*(-4.2);
temp2=temp;
OCR0=temp2;
pre_error=error;
}
```

بدلیل اهمیت بالای این قسمت از برنامه به بررسی خط به خط ان می پردازیم.

```
 sender=sensor;
 putchar(sender);
```

در این قسمت در هر مرحله اجرا برنامه مقدار sensor که همان مقدار دور موتور است درون متغییر sender قرار می گیرد تا از طریق تابع () putchar ارسال شود.

```
if(ok==1){
    ok=0;
```

این قسمت تعیین می کند که پس از اجرای هر مرحله ی وقفه ی میکرو این برنامه ها اعمال شوند سپس مقدار ok بلافاصله صفر شده تا دوباره مقدار ok درزیر برنامه وقفه یک شود.

```
error=setpoint-sensor;
```

تمام محاسبات مهم کنترلی برمبنای مقدار error هستند بنابراین مقدار error که از تفاضل میان مقدار مطلوب و مقدار شمارش شده است محاسبه می شود .

```
if(error>0){
    integral=integral+(error*dt);
```

```
derivative=(error-pre_error)/dt;
output=kp*error+ki*integral+kd*derivative ;
temp=output*4.2;
OCR0=temp;
pre_error=error;
}
```

زمانی که مقدار دور موتور به مقدار setpoint نرسیده است مقدار error مثبت است و زمانی که بدلیل فراجهش کوچکی مقدار دور موتور از مقدار setpoint بیشتر شود مقدار error منفی شده و جهت محاسبات صحیح در این حالت در قسمت else بررسی شده است .

مرحله ی اول محاسبه ی قسمت تناسبی است که مقدار error تنها در یک بهره استاتیکی ضرب می شود ، مهمترین قسمت بدست اوردن مقدار بهره تناسبی است که در این پروژه بدلیل نبود تابع تبدیل سیستم این مقدار از روش سعی و خطا بدست امده .

مرحله ی دوم محاسبه ی مقدار انتگرالی است که نرم افزار مقدار خطای هر مرحله را جمع می کند و در ثابت انتگرال که مشابه ضریب تناسبی بدست می اید ضرب می کند .

integral=integral+(error*dt)

مرحله ی سوم مقدار خطای فعلی از خطای قبلی کم شده و تقسیم بر مدت زمان بین دو خطا می شود سپس در ثابت مشتق ضرب می شود و مقدار قسمت مشتقگیری نیز بدست می اید

derivative=(error-pre_error)/dt

در اخر مقادیر بدست امده از سه مرحله ی قبل با هم جمع شده و فرمان کنترلی صادر می شود این مقدار در ضریبی ضرب شده تا در رنج مقادیر ocr میکرو قرار گیرد سپس اماده ارسال می شود.

```
output=kp*error+ki*integral+kd*derivative;

temp=output*4.2;

                OCR0=temp;
```

و در اخر مقدار خطا، درون مقدار خطای قبلی قرار می گیرد تا در مرحله ی بعد استفاده شود.

```
            pre_error=error;
```

در قسمت else برنامه تنها تفاوت این است که مقدار output در مقدار منفی 4.2 ضرب می شود تا منفی error مثبت شده ومحاسبات مانند حالت قبل انجام شود.

در انتهای برنامه دستوراتی جهت نمایش اطلاعات بر روی lcd قرار داده شده است که توابع lcd در قسمت ضمیمه بطور کامل بررسی شده است.

بر روی lcd مقدار setpoint ومقدارerror و مقدار sensor ومقادیر (Kp,Ki,Kd) نشان داده شده است.این مقادیر براساس توابع lcd قابل نمایش است.

```
        lcd_clear();

        lcd_gotoxy(0,0);

        lcd_putsf("Setpoint”);

        lcd_gotoxy(9,0);

        itoa(setpoint,lcdbuf

    lcd_puts(lcdbuf);

--------------------//

        lcd_gotoxy(0,1);

        lcd_putsf("Sensor”);

        lcd_gotoxy(7,1);
```

```
            ftoa(sensor,0x02,lcdbuf);
            lcd_puts(lcdbuf)
--------------------//
            lcd_gotoxy(0,2);
            lcd_putsf("Error:”);
            lcd_gotoxy(6,2);
            ftoa(error,0x02,lcdbuf);
            lcd_puts(lcdbuf)
----------------------//
            lcd_gotoxy(13,0);
            lcd_putsf("Kp=”);
            lcd_gotoxy(16,0);
            ftoa(kp,0x02,lcdbuf);
            lcd_puts(lcdbuf)

            lcd_gotoxy(13,1);
            lcd_putsf("Ki=”);
            lcd_gotoxy(16,1);
            ftoa(ki,0x02,lcdbuf)
            lcd_puts(lcdbuf);
```

```
            lcd_gotoxy(13,2);
            lcd_putsf("Kd=”);
            lcd_gotoxy(16,2);
            ftoa(kd,0x02,lcdbuf);
            lcd_puts(lcdbuf);
----------------------//
```

فصل چهارم

خلاصه پروژه و پیشنهادات

این پروژه می تواند دید عملی نسبت به ساختار های کنترلی وسیستم هایی اتوماتیکی به خواننده بدهد،بطور کلی مفهوم کنترل را درک کند.

هدف این پروژه کنترل موتور بصورت وایرلس است که با رسیدن به این هدف انسان قادر خواهد بود در بسیاری از تجهیزات بخصوص در کنترل رباتها توانایی بسیار بالایی داشته باشد .

بهتر است بعد از فراگیری مطالب کنترل موتور DC به بررسی کنترل موتورهای AC که در صنعت کاربرد بیشتری دارد پرداخت ، بگونه ای که این پروژه می تواند شروع خوبی برای پیشرفت در سیستم های کنترلی باشد.

بطور کلی کنترلPID می تواند در هر سیستمی پیاده سازی شود ، اصولاً کنترلر وظیفه مقايسهٔ مقدار واقعي پروسه با مقدار مطلوب يا نقطه تنظيم و همچنين صدور فرمان براي اكچوئيتور جهت انجام اين تنظيم را دارد . براي مثال در كنترلرهاي ساده نظير ترموستات كه سنسور كه وظيفه اندازه گيري دما را دارد معمولاً از يک بيمتال تشكيل شده است و در حقيقت اين بيمتال با توجه به كم يا زياد شدن دما به يک طرف متمايل شده با تنظيم مناسب جهت دماي مطلوب خروجي كنترلر كه همان فرمان خاموش يا روشن كردن منبع حرارتي است صادر ميشود .اما در يک كنترلر PID با در نظر گرفتن پارامترهائي نظير نسبت تغييرات نقطه تنظيم با مقدار فعلي پروسه با فاكتورهائي نظير شدت تغييرات ناگهاني ، و ميزان تغييرات نسبت به زمان ميتواند كنترلي پيشرفته خصوصا در پروسه هاي دمائي كه استفاده از كنترلرهاي ساده باعث ايجاد نوسان دائمي دما ميشود را ايجاد كند . در كنترلر پي آي دي از الگوريتم نسبي ، مشتقي ، و انتگرالي بهره ميبرند همانگونه كه نام آن از P مخفف Proportionalيا نسبي و فاكتور I مخفف integralيا انتگرالي و فاكتور D مخفف Derivative يا مشتقي گرفته شده است .و در بسياري از فرايندهاي كنترلي نظير كنترل سرعت موتور DC ، كنترل فشار، كنترل دما و ... كاربرد دارد.

سوال مهمی بوجود می اید که فرمان کنترلی صادر شده توسط میکرو کنترولر چگونه باعث تولید موج PWM می شود در پاسخ می توان گفت که بطور کلی فرمان کنترلی در محدوده ی خاصی قرار دارد و مقدار OCR میکرو که تعیین کننده ی زمان وظیفه موج است نیز محدوده ای دارد که بیشتر از محدوده ی فرمان کنترلی است که با یک ضریب در نرم افزار می توان این دو محدوده را تحت پوشش قرار داد به همین ترتیب می توان فرمان کنترلی را تبدیل به موج PWMکرد که از قسمت های مهم پروژه است .

طراحی کنترلر از طریق محاسبات :

این پروژه بدلیل اینکه تابع تبدیل plan را نداریم مجبور بودیم که از روش سعی و خطا کنترلر را طراحی کنیم ولی در این قسمت محاسبات طراحی را بیان می کنیم:

مدلسازي موتور DC

يك موتور DC آهنرباي دائم نيروي الكتريكي را به نيروي مكانيكي تبديل مي كند. مدار الكتريكي يك موتور DC در شكل ۱.۱ رسم شده است كه در آن :

Va : ولتاژ تغذیه موتور DC

Ra: مقاومت آرمیچر

B: ضریب دمپینگ

La: اندوکتانس

Ea : ولتاژ القایی

Wa: سرعت موتور

Kt : ثابت گشتاور

Tw: گشتاور اصطکاک

Tw : گشتاور اینرسی

J: ممان اینرسی

Tl : گشتاور بار

Te : گشتاور الکترومغناطیسی

Ke: ثابت ولتاژ القایی

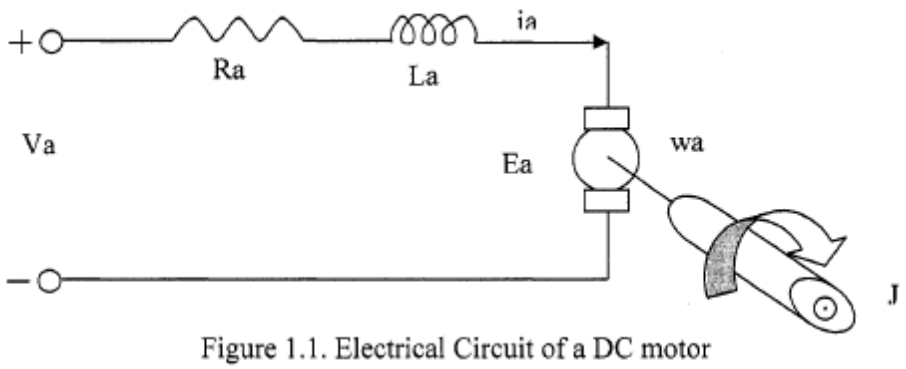


Figure 1.1. Electrical Circuit of a DC motor

مشخصات الكتريكي :

با زدن يك KVL در مدارشكل ١.١ يك معادلة ديفرانسيل از آن حاصل مي شود.

$$V_a - V_{ra} - V_{la} - V_c = 0 \qquad (1.1)$$

با توجه به قانون أهم داريم :

$$V_{ra} = i_a R_a \qquad (1.2)$$

ولتاژ دو سر سلف متناسب با مشتق جريان گذرنده از سلف نسبت به زمان مي باشد كه مي توان نوشت :

$$V_{la} = L_a \frac{di_a}{dt} \qquad (1.3)$$

ولتاژ القايي (back emf) را مي توان به صورت زير نوشت:

$$E_a = k_e \omega_a \qquad (1.4)$$

با جايگذاري روابط (١.٢) و (١.٣) و (١.۴) در (١.١) نتيجه مي شود :

$$V_a - i_a R_a - L_a \frac{di_a}{dt} - k_e \omega_a = 0 \qquad (1.5)$$

قانون بقاي انرژي ايجاب مي كند كه جمع گشتاور هاي موجود در موتور بايد برابر صفر باشد. بنابراين داريم :

$$T_e - T_\omega - T_{\omega'} - T_l = 0 \qquad (1.6)$$

گشتاور الكترومغناطيس با جريان سيم پيچ آرميچه متناسب است و مي توان نوشت :

$$T_e = k_t i_a \qquad (1.7)$$

$$T_{\omega'} = J\frac{d\omega_a}{dt} \qquad (1.8)$$

$$T_\omega = B\omega_a \qquad (1.9)$$

با جايگذاري روابط (۱.۷) و (۱.۸) و (۱.۹) در (۱.۶) معادلة ديفرانسيل زير نتيجه ميشود :

$$k_t i_a - J\frac{d\omega_a}{dt} - B\omega_a - T_l = 0 \qquad (1.10)$$

با مرتب كردن معادلات (۱.۵) و (۱.۱۰) به صورت معادلات حالت ، مي توان نوشت :

$$\frac{di_a}{dt} = -\frac{R_a}{L_a} i_a - \frac{k_e}{L_a}\omega_a + \frac{V_a}{L_a} \qquad (1.11)$$

$$\frac{d\omega_a}{dt} = \frac{k_t}{J} i_a - \frac{B}{J}\omega_a - \frac{T_l}{J} \qquad (1.12)$$

معادلات (۱.۱۱) و (۱.۱۲) در فرم ماتريسي به صورت زير در مي آيند:

$$\frac{d}{dt}\begin{bmatrix} i_a \\ \omega_a \end{bmatrix} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{k_e}{L_a} \\ \frac{k_t}{J} & -\frac{B}{J} \end{bmatrix}\begin{bmatrix} i_a \\ \omega_a \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} & 0 \\ 0 & -\frac{1}{J} \end{bmatrix}\begin{bmatrix} V_a \\ T_l \end{bmatrix}$$

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} i_a \\ \omega_a \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} V_a \\ T_l \end{bmatrix}$$

با اعمال تبديل لاپلاس بر روي معادلات (۱.۱۱) و (۱.۱۲) خواهيم داشت :

$$I_a(s) = \frac{V_a(s) - E_a(s)}{R_a + sL_a} \quad (1.13) \quad \text{where} \quad E_a(s) = k_e.\omega_a(s)$$

$$\omega_a(s) = \frac{T_e(s) - T_l(s)}{sJ + B} \quad (1.14) \quad \text{where} \quad \begin{matrix} T_e(s) = k_t I_a(s) \\ k_t = k_e \end{matrix}$$

طراحي موتور DC در سيمولينك :

از معادلات فوق بلوك دياگرام موتور DC بدست مي آيد كه در زير نشان داده شده است.

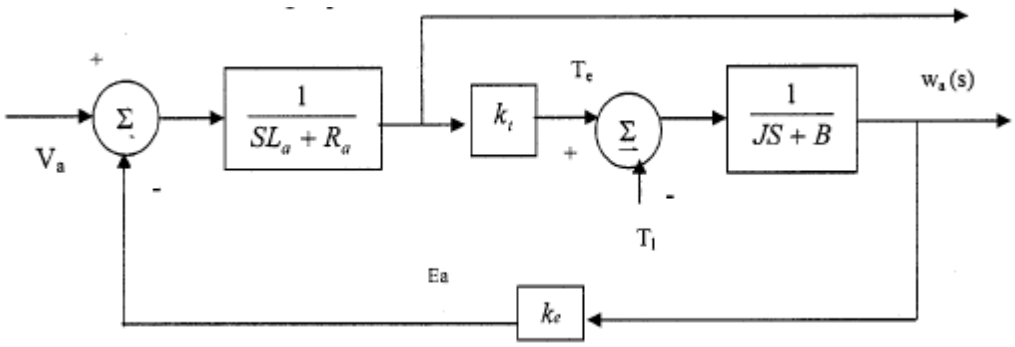


Figure 1.2. Block diagram representation of DC motor

بنابراين با ملاحظه بلوك دياگرام فوق ما مي توانيم بلوك دياگرام موتور DC را در سيمولينك بسازيم كه در شكل ۱.۳ نشان داده شده است:

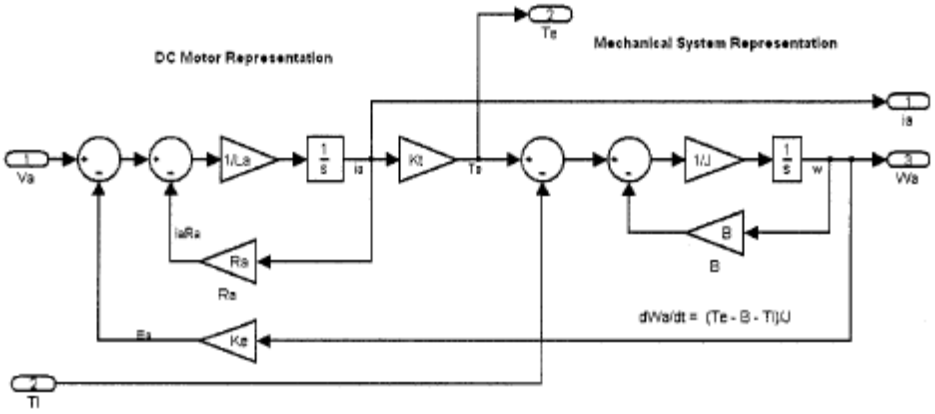


Figure 1.3. Simulink diagram of DC motor

كنترل آبشاري موتور DC

يك سيستم كنترلي آبشاري شامل دو حلقه فيدبك است كه براي كنترل سرعت موتور DC استفاده مي شود و عملكرد سيستم را نسبت به سيستم كنترلي تك حلقه ارتقا مي بخشد . حلقة كنترلي داخلي براي كاهش پارازيت هاي روي حلقة دوم استفاده مي شود در حالي كه حلقة خارجي براي كاهش حساسيت متغيرهاي مرحله اول نسبت به نوسانات بهره استفاده مي شود.

سيستم كنترل آبشاري مورد بحث در اين پايان نامه دو نوع كنترل كننده را با هم درگير مي كند .يكي كنترل كنندة سرعت كه براي كنترل جريان موتور DC مي باشد و ديگري كنترل كنندة جريان است كه براي كنترل جريان موتور DC مي باشد. شكل زير سيستم كنترل آبشاري موتور DC را نشان مي دهد.

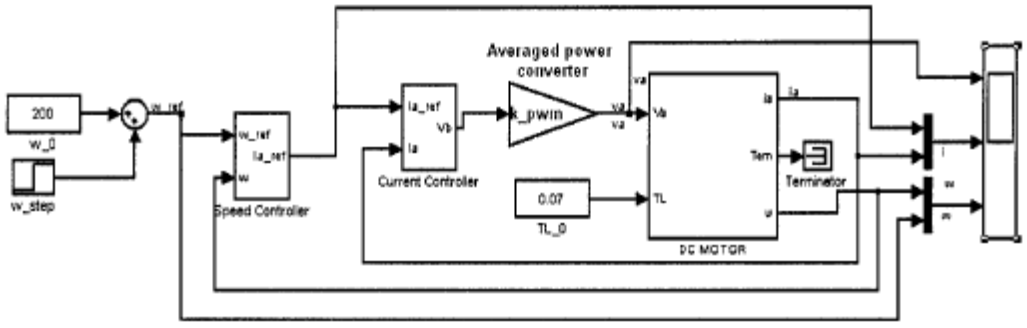


Figure 1.4. Cascade control of DC motor

كنترل كننده سرعت كه حلقه خارجي مي باشد دو ورودي دارد. يكي از ورودي ها سرعت مطلوب (مرجع)مي باشد و ورودي ديگر سرعت موتور DC مي باشد كه از خروجي موتور فيدبك گرفته مي شود.

طراحي كنترل كننده PID

بسياري از كنترل كننده هايي كه در حال حاضر در صنعت هستند ، از نوع PID مي باشند. ورودي PID يك سيگنال خطا (اختلاف بين سيگنال مرجع و سيگنال خروجي) مي باشد. خروجي حاصل جمع سه جمله مي باشد . جملة اول متناسب با سيگنال خطا، جملة دوم متناسب با انتگرال سيگنال خطا و جملة سوم متناسب با مشتق سيگنال خطا مي باشد. تابع تبديل PID به صورت زير بيان مي شود.

$$u(t) = ke(t) + k_i \int_0^t e(\tau)d\tau + k_d \frac{de}{dt} \qquad (3.1)$$

كه در آن :

K = بهرة متناسب (KP)

Ki = بهرة انتگرال

بهرة مشتق = Kd

سيگنال خطا = e(t)

معادلة (١-٣) را مي توان به صورت زير نوشت :

$$u(t) = k\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\frac{de(t)}{dt}\right) \qquad (3.2)$$

كه در آن Ti بازة زماني انتگرال گيري و Td بازة زماني مشتق گيري مي باشد.
پارامترهاي دو معادلة ١.٣ و ٣.٢ به صورت زير به هم مربوط مي شوند :

$$k = K \qquad k_i = \frac{K}{T_i} \qquad k_d = KT_d \qquad (3.3)$$

مشخصات كنترل كنندة PID

كنترل كننده متناسب (Proportional) زمان خيز سيستم را كاهش مي دهد اما اين كنترل كنندة خطاي حالت ماندگار را حذف نمي كند . كنترل كنندة انتگرال گير مي تواند خطاي حالت ماندگار را كاهش دهد اما اُورشوت را افزايش مي دهد. كنترل كنندة مشتق گير پايداري سيستم را افزايش مي دهد و همچنين اورشوت را كاهش مي دهد. عملكرد هر كدام از سه كنترل كنندة فوق در جدول زير خلاصه شده است :

Controller	Rise time	Overshoot	Settling time	Steady-state error
Kp	Decrease	Increase	Small Change	Decrease
Ki	Decrease	Increase	Increase	Eliminate
Kd	Small Change	Decrease	Decrease	Small Change

Table 3.1. Characteristics of each control parameters [14].

تنظيم كردن PID

كنترل كنندة PID به چند روش مي تواند تنظيم شود. به غير از روش هاي عمومي كه براي تنظيم PID به كار مي رود . چند روش خاص وجود دارد كه توسط « زيگلر » و « نيكول » معرفي شدند كه با تعداد پارامتر كم و معادلات آسان در ارتباط است .يكي از روش ها بر پاسخ شيب پايه گذاري شده است. روش ديگر روش پاسخ فركانسي است.كنترل كنندة PID كه در اين پايان نامه بحث شده با روش پاسخ فركانسي طراحي شده است .اين روش بر پيدا كردن نقطه تلاقي نمودار نايكوئيست با قسمت منفي محور حقيقي مبتني است.

پارامترهاي Ku (بهره نهايي) و Tu (پريود نهايي) از روي نمودار نايكوئيست تعيين مي شوند. جدول ۲.۳ جهت تعيين پارامترهاي PID به ما كمك مي كند. يكبار كه پارامترهاي D, I, P تعيين شدند اين پارامترها بوسيله روش آزمايش و خطا به گونه اي تنظيم مي شوند كه خروجي مطلوب بدست آيد.

Controller	K	Ti	Td
P	0.5Ku		
PI	0.4Ku	0.8Tu	
PID	0.6Ku	0.5Tu	0.125Tu

Table 3.2. Controller parameters using Ziegler-Nichols frequency method [15]

كنترل كنندة سرعت PID

خطاي بين سيگنال مرجع و خروجي واقعي به ورودي كنترل كنندة سرعت داده مي شود. يك ورودي شيب بين ۲۰۰ تا ۴۰۰ به عنوان مرجع به سيستم داده مي شود.

دياگرام يك كنترل كنندة سرعت را كه در سيمولينك كشيده شده نشان مي دهد.

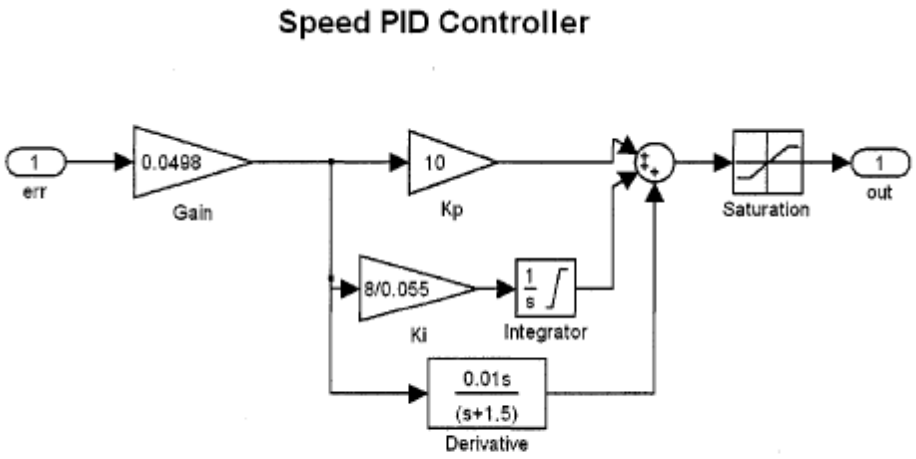


Figure 3.1. Speed PID controller

در زير پارامترهاي تنظيم شده كنترل كنندة سرعت نشان داده شده است .مقدار اشباع روي ۵± تنظيم شده است

Kp = 0.498 Ki = 7.24 Kd = 0.000488

اين كنترل كننده به گونه اي طراحي مي شود كه خروجي آن به عنوان سيگنال مرجع براي كنترل كنندة جريان PID باشد.

كنترل كننده جريانPID

ورودي كنترل كننده جريان PID يك سيگنال خطا است. اين سيگنال خطا اختلاف بين جريان مرجع (كه از خروجي كنترل كنندة سرعت گرفته مي شود) و جريان واقعي (كه همان جريان مصرفي موتور است) مي باشد. مقدار اشباع روي ۱± تنظيم شده است. در زير پارامترهاي D, I, P از كنترل كنندةجريان PID نشان داده شده است .

Kp = 0.04 Ki = 0.366 Kd = 0.000412

شكل زير يك كنترل كننده جريان PID را نشان مي دهد.

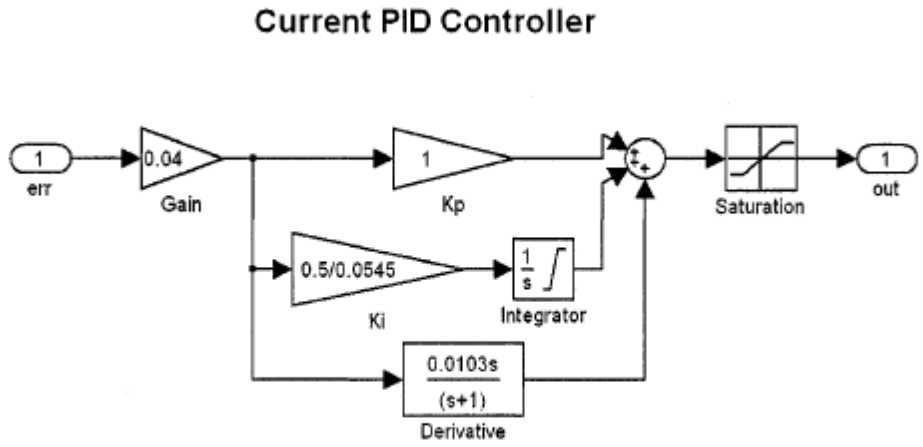


Figure 3.2. Current PID controller

خروجي كنترل كننده جريان PID به يك مبدل توان داده مي شود كه براي مدوله كردن پهناي پالس ولتاژ در كنترل كنندة توان استفاده مي شود . خروجي كنترل كننده توان متوسط به عنوان تغذيه به موتور DC داده مي شود.

منابع

الکترونیک صنعتی م . ه .رشید (مترجمان علیرضا صداقتی ،بهزاد قهرمان)

مبانی الکترونیک دکتر سید علی میرعشقی

اصول مفاهیم کنترل دکتر علی خاکی صدیق

پیوست ۱

بررسی امکانات AT-mega16

خصوصیات ATMEGA 16L, ATMEGA 16:

- از معماری AVR RISC استفاده می کند.

- کارایی بالا و توان مصرفی کم

- دارای ۱۳۱ دستورالعمل با کارایی بالا که اکثراً تنها در یک کلاک سیکل اجرا می شوند.

- ۸-۳۲ رجیستر کاربردی.

- سرعتی تا 16 MIPS در فرکانی 16 MHZ.

- حافظه برنامه و داده غیر فرار.

- 16K بایت حافظه FLASH داخلی قابل برنامه ریزی.

- پایداری حافظه FLASH؛ قابلیت ۱۰۰۰۰ بار نوشتن و پاک کردن.

- ۱۰۲۴ بایت حافظه داخلی SRAM.

- ۵۱۲ بایت حافظه EEPROM داخل قابل برنامه ریزی.

- پایداری حافظه EEPROM؛ قابیت ۱۰۰۰۰۰ بار نوشتن و پاک کردن.

- قفل برنامه EEPROM, FLASH.

خصوصیات جانبی

- دوتایمر/ کانتر ۸ بیتی با Prescaler مجزا و مود Capture.

- یک تایمر/ کانتر ۱۶ بیتی با Prescaler مجزا و دارای مودهای Compare, Capture.

- ۴ کانال PWM.

- ۸ کانال مبدل آنالوگ به دیجیتال ۱۰ بیتی.

- یک مقایسه کننده آنالوگ داخلی.

- WATCHDOG قابل برنامه ریزی با اسیلاتور داخلی.

- قابلیت ارتباط با پروتکل سریال دوسیمه (TWO WIRE) ,12C.

- قابلیت ارتباط سریال SPI (Serial Peripheral interface) به صورت Master یا Slave.

- USART سریال قابل برنامه ریزی.

خصوصیات ویژه میکروکنترلر

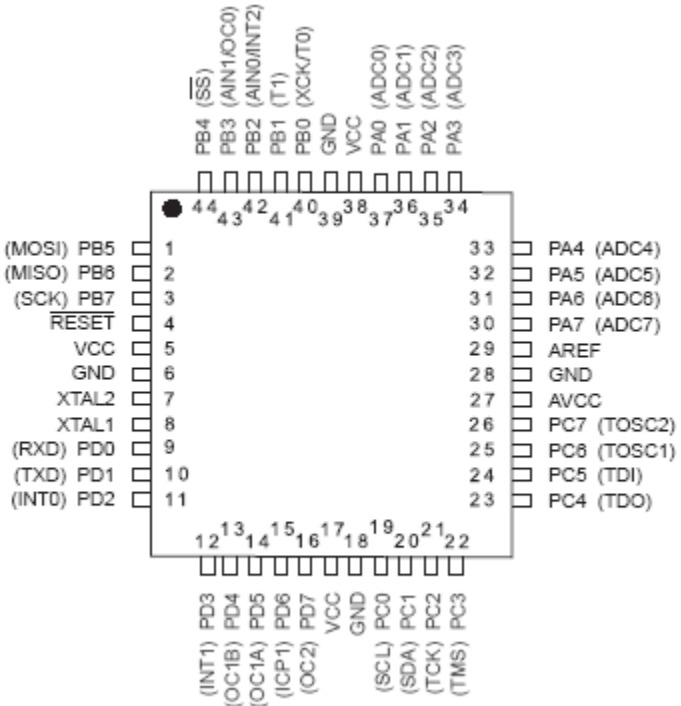
- دارای اسیلاتور RC داخلی کالیبره شده.

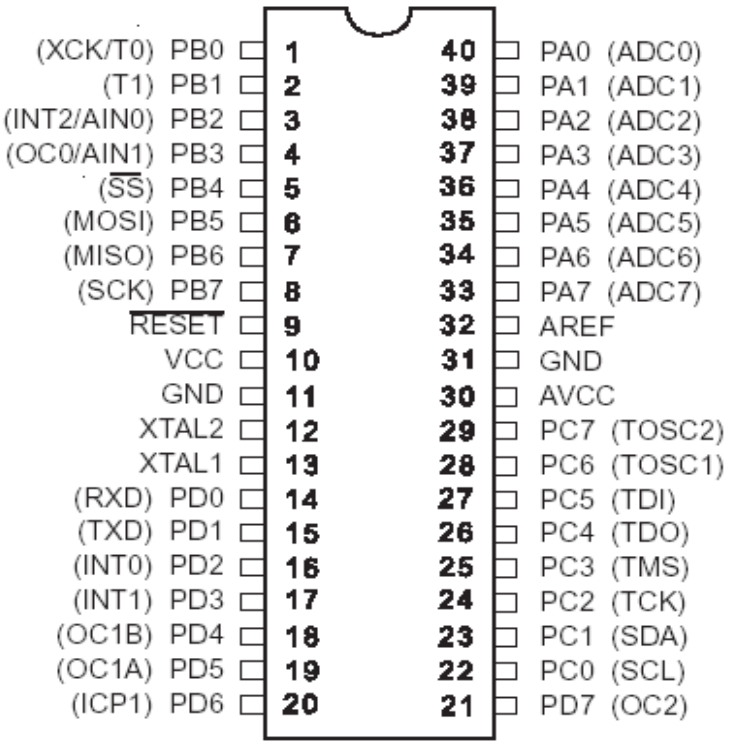
- منابع وقفه (interrupt) داخلی و خارجی.

- توان مصرفی پایین و سرعت بالا توسط تکنولوژی COMS.

- ولتاژ کاری ۲. ۷ تا ۵. ۵ ولت برای Atmega 16L و ۴. ۵ تا ۵. ۵ وبت برای Atmega 16 فرکانس های کاری 0 تا 8MHZ برای Atmega 16L ,0 تا 16MHZ برای Atmega 16.

- ۳۲ خط ورودی خروجی قابل برنامه ریزی.





همانطور که در شکل دیده می شود ATMEGA 16 دارای ۴ پورت D,C,B,A می باشد که علاوه بر اینکه به عنوان ورودی- خروجی مورد استفاده قرار می گیرند، کاربردهای جانبی دیگری نیز دارند. در این بخش به شرح این پورت ها می پردازیم:

پورت A

پورت A یک I/O دو طرفه ۸ بیتی است. سه آدرس از مکان حافظه I/O اختصاص به PORTA دارد؛ یک آدرس برای رجیستر داده PORTA، دومی برای رجیستر جهت داده DDRA و سومی برای رجیستر ورودی PINA.

پورت A به عنوان ADC هم استفاده می شود. اگر تعدادی از پایه های پورت A خروجی تعریف شوند، این نکته بسیار مهم است که در زمان نمونه برداری از سیگنال آنالوگ توسط ADC، سوئیچ نشوند.این کار ممکن است عملیات تبدیل ADC را نامعتبر کند.

پورت B

پورت B یک I/O دو طرفه ۸ بیتی است. سه آدرس از مکان حافظه I/O اختصاص به پورت B دارد؛ یک آدرس برای رجیستر داده PORTB، دومی برای رجیستر جهت داده DDRB و سومی برای رجیستر ورودی PINB.

دیگر کاربردهای پورت B

- PORTB.7 (SCK): کلاک خروجی MASTER و کلاک ورودی SLAVE برای ارتباط SPI است. زمانی که SPI به عنوان SLAVE شکل دهی می شود این پایه با توجه به تنظیم DDRB.7 ورودی و در حالت MASTER خروجی تعریف می شود.

- PORTB. 6 (MISO): ورودی داده MASTERو خروجی داده SLAVE که برای ارتباط SPI استفاده می شود.

- PORTB.5 (MOSI): ورودی داده SLAVE و خروجی داده MASTER که برای ارتباط SPIاستفاده می شود.

- PORTB.4 (SS): زمانی که SPI به عنوان SLAVE شکل دهی می شود PORTB.4 با توجه به DDRB.4 ورودی تعریف می شود و در SLAVE با LOW شدن این پایه SPI فعال می شود.

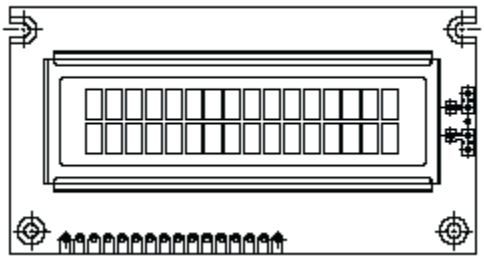
پورت C

پورت C یک I/O دو طرفه ۸ بیتی است. سه آدرس از مکان حافظه I/O اختصاص به PORTC دارد؛ یک آدرس برای رجیستر داده PORTC، دومی برای رجیستر جهت داده DDRC و سومی برای رجیستر ورودی PINC.

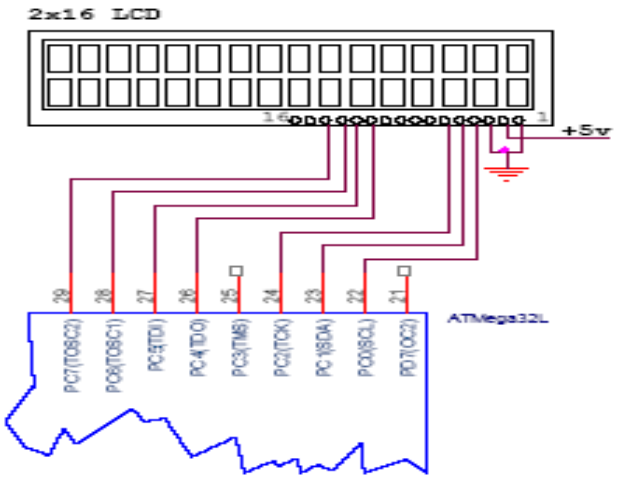
پورت D

پورت D یک I/O دوطرفه ۸ بیتی است. سه آدرس از مکان حافظه I/O اختصاص به PORTD دارد؛ یک آدرس برای رجیستر داده PORTD، دومی برای رجیستر جهت داده DDRD و سومی برای رجسیتر ورودی PIND.

نمایش بر روی LCD:



نحوه اتصال LCD به میکروکنترلر:



پایه های ۱و ۲ به ترتیب به زمین و ولتاژ ۵ ولت متصل می شوند. پایه ۳ میزان شدت نمایش کاراکترها بر روی LCD را مشخص می کند (contrast) پایه های ۶، ۵، ۴ نیز به کنترل LCD مربوط می شود که باید توسط میکروکنترلر تنظیم شود. از پایه های ۷ تا ۱۴ نیز به عنوان یک باس ۸ بیتی برای انتقال اطلاعات مابین LCD و میکروکنترلر استفاده می شود. برای تبادل اطلاعات بین LCD و میکروکنترلر از روش ارتباط باس چهار سیمه استفاده می شود.

پایه های ۱۰، ۹، ۸، ۷ در محیطهای پر نویز استفاده می شود که با مقاومتهای ۱۰ کیلو اهم به زمین متصل می کنند (Pull down).

پایه های ۱۶، ۱۵ به ترتیب پلاریته مثبت و منفی برای روشن کردن نور زمینه LCD استفاده می شود.

PIN NUMBER	SYMBOL	FUNCTION
1	Vss	GND
2	Vdd	+ 3V or +5V
3	Vo	Contrast Adjustment
4	RS	H/L Register Select Signal
5	R/$\overline{W}$	H/L Read/Write Signal
6	E	H →L Enable Signal
7	DB0	H/L Data Bus Line
8	DB1	H/L Data Bus Line
9	DB2	H/L Data Bus Line
10	DB3	H/L Data Bus Line
11	DB4	H/L Data Bus Line
12	DB5	H/L Data Bus Line
13	DB6	H/L Data Bus Line
14	DB7	H/L Data Bus Line
15	A/Vee	4.2V for LED/Negative Voltage Output
16	K	Power Supply for B/L (OV)

MECHANICAL DATA		
ITEM	STANDARD VALUE	UNIT
Module Dimension	84.0 x 44.0	mm
Viewing Area	66.0 x 16.0	mm
Dot Size	0.55 x 0.65	mm
Character Size	2.95 x 5.55	mm

ABSOLUTE MAXIMUM RATING					
ITEM	SYMBOL	STANDARD VALUE			UNIT
		MIN.	TYP.	MAX.	
Power Supply for Logic	VDD-VSS	- 0.3	–	7.0	V
Input Voltage	VI	- 0.3	–	VDD	V

NOTE: VSS = 0 Volt, VDD = 5.0 Volt

توابع کتابخانه LCD.h:

قبل از هر چیز لازم است مشخص شود که ماژول lcd به کدامیک از پورتهای میکروکنترلر وصل شده است البته با استفاده از Code Vizard نیز می توان پورت را مشخص کر.

Unsigned char lcd- init (unsigned char lcd- columns)

این تابع ماژول LCD را مقدار دهی اولیه می کند. با فراخوانی این تابع، صفحه نمایش LCD پاک شده، مکان نما نیز حذف می گردد و LCD برای نوشتن کاراکتر در محل سطر و ستون صفر آماده می شود.

Void icd-clear (void)

این تابع، صفحه نمایش LCD را پاک می کند و برای چاپ کاراکتر در محل سطر و ستون صفر آماده می شود.

Void lcd- gotoxy (unsigned char x,unsigned char y)

این تابع موفقیت فعلی برای چاپ کاراکتر را به محل ستون x و سطر y منتقل می کند. به طور کلی مبدأ مختصات (۰و ۰) در lcd های کاراکتری، بالا و سمت چپ صفحه نمایش است.

Void lcd- putchar (char c)

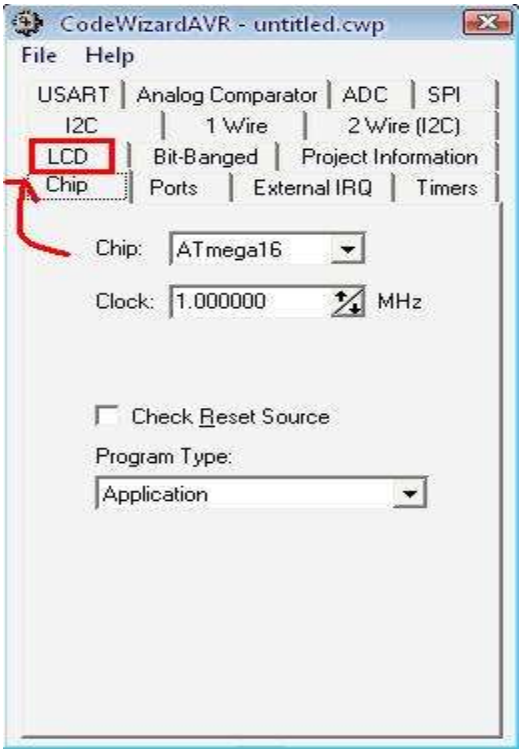
این تابع کاراکتر c را بر روی موقعیت فعلی انتخاب شده بر روی lcd می نویسد.

Void lcd- putsf (char flash * str)

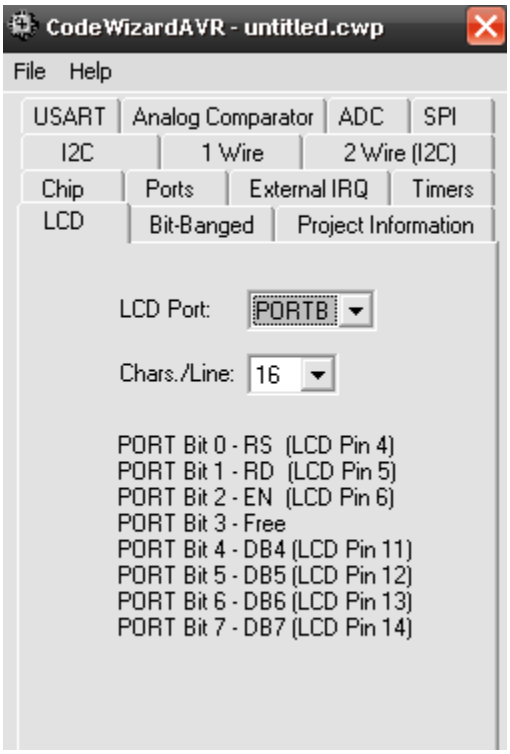
این تابع رشته ی واقع در حافظه flash را با شروع از موقعیت فعلی انتخاب شده، بر روی lcdمی نویسد.

تنظیمات اولیه LCD در Code Wizard:

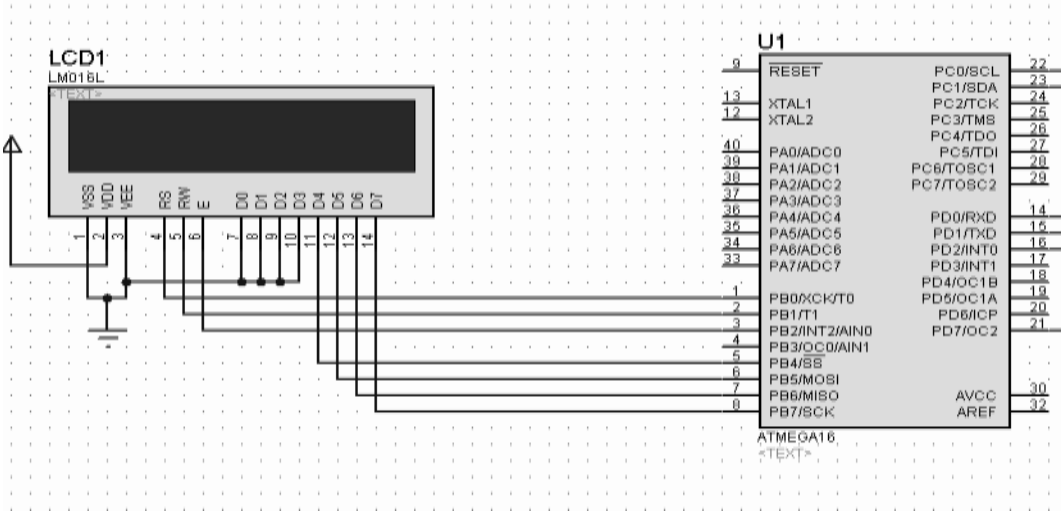
برای این کار کافی است تا پس از ایجاد یک پروژه جدید و استفاده از Wizard (که در ادامه نحوه انجام این کار در قسمت تولید PWM با جزئیات بیشتر بحث شده است) ایجاد کنید. حال در پنجره Wizard به قسمت LCD بروید.



حال در این قسمت تنها کافی است پورت مورد نظر را که قصد دارید LCD به آن وصل کنید، معین کنید.



با توجه به شکل بالا با انتخاب پورت B نحوه ی اتصال پایه های lcd به میکرو مشخص شده است. که در زیر مدار آن رسم شده است.



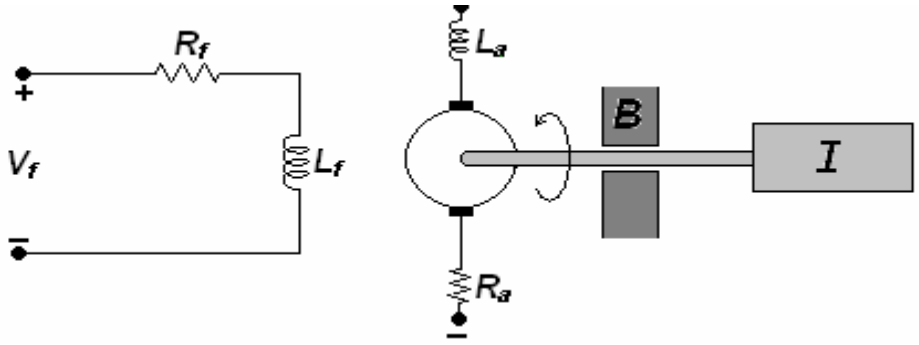
پیوست ۲

بررسی موتور های DC

موتور dc چگونه کار می کند؟

یکی از اولین موتورهای دوار،اگر نگوییم اولین، توسط میشل فارادی در سال 1821م ساخته شده بود و شامل یک سیم آویخته شده آزاد که در یک ظرف جیوه غوطه وربود ، می شد . یک اهنربای دائم در وسط ظرف قرار داده شده بود . وقتی که جریانی از سیم عبور می کرد ، سیم حول آهنربا به گردش در می آمد و نشان می داد که جریان منجر به افزایش یک میدان مغناطیسی دایره ای اطراف سیم می شود.
موتور کلاسیک DC دارای آرمیچری از آهنربای الکتریکی است. یک سوییچ گردشی به نام کموتاتور جهت جریان الکتریکی را در هر سیکل دو بار برعکس می کند تا در آرمیچر جریان یابد و آهنرباهای الکتریکی، آهنربای دائمی را در بیرون موتور جذب و دفع کنند .

مدل دینامیکی موتور



سرعت موتور DC به مجموعه ای از ولتاژ و جریان عبوری از سیم پیچ های موتور و بار موتور یا گشتاور ترمزی، بستگی دارد. سرعت موتور DC وابسته به ولتاژ و گشتاور آن وابسته به جریان است. معمولاً سرعت توسط ولتاژ متغیر یا عبور جریان و با استفاده از تپ ها (نوعی کلید تغییر دهنده وضعیت سیم پیچ) در سیم پیچی موتور یا با داشتن یک منبع ولتاژ متغیر، کنترل می شود .بدلیل اینکه این نوع از موتور می تواند در سرعتهای پایین گشتاوری زیاد ایجاد کند، معمولاً از آن در کاربردهای ترکشن (کششی) نظیر لکوموتیوها استفاده می کنند .

موتور الکتریکی، نوعی ماشین الکتریکی است که الکتریسیته را به حرکت مکانیکی تبدیل می‌کند. عمل عکس آن که تبدیل حرکت مکانیکی به الکتریسیته‌است، توسط ژنراتور انجام می‌شود. این دو وسیله بجز در عملکرد، مشابه یکدیگر هستند. اکثر موتورهای الکتریکی توسط الکترومغناطیس کار می‌کنند، اما موتورهایی که بر اساس پدیده‌های دیگری نظیر نیروی الکترواستاتیک و اثر پیزوالکتریک کار می‌کنند، هم وجود دارند.

ایده کلی این است که وقتی که یک ماده حامل جریان الکتریسیته تحت اثر یک میدان مغناطیسی قرار می‌گیرد، نیرویی بر روی آن ماده از سوی میدان اعمال می‌شود. در یک موتور استوانه‌ای، چرخانه روتور به علت گشتاوری که ناشی از نیرویی است که به فاصله‌ای معین از محور چرخانه به چرخانه اعمال می‌شود، می‌گردد.

اغلب موتورهای الکتریکی دوار هستنند، اما موتور خطی هم وجود دارند. در یک موتور دوار بخش متحرک (که معمولاً درون موتور است) چرخانه یا روتور و بخش ثابت ایستانه یا استاتور خوانده می‌شود. موتور شامل آهنرباهای الکتریکی است که روی یک قاب سیم پیچی شده‌است. گر چه این قاب اغلب آرمیچر خوانده می‌شود، اما این واژه عموماً به غلط بکار برده می‌شود. در واقع آرمیچر آن بخش از موتور است که به آن ولتاژ ورودی اعمال می‌شود یا آن بخش از ژنراتور است که در آن ولتاژ خروجی ایجاد می‌شود. با توجه به طراحی ماشین، هر کدام از بخش‌های چرخانه یا ایستانه می‌توانند به عنوان آرمیچر باشند.

موتورهاي الكتريكي

يك موتورالكتريكي الكتريسيته رابه حركت مكانيكي تبديل مي كند.

عمل عكس آن كه تبديل حركت مكانيكي به الكتريسيته است توسط ژنراتور صورت مي گيرد.

اغلب موتورهای الکتریکی دوارند، اما موتور خطی هم وجود دارد. در یک موتور دوار بخش متحرک كه معمولاً درون موتوراست روتوروبخش ثابت استاتور خوانده مي شود.

انواع موتورها

1- موتورهاي DC

2– موتورهاي AC

موتورهايDC

يك موتوردي سي ماشيني است كه وقتي به آن جريان مستقيم داده مي شودمي تواند براي كارهاي مكانيكي از قبيل به حركت درآوردن پمپ ها گرداندن ماشين هاي ابزار و غيره به كار رود.همچنين از موتورهاي الكتريكي جريان مستقيم به طور وسيعي براي كاربردهاي كه به كنترل سرعت نياز دارد استفاده مي شود.بعضي از اين موارد عبارتند از:پرسهاي دستگاه چاپ. قطارهاي برقي .وسايل نقليه و آسانسورها. موتورهاي جريان مستقيم در اندازه هاي مختلفي از ۱/۱۰۰ اسب بخار تا هزارها اسب ساخته مي شود.

سرعت موتور دي سي به مجموعه ای از ولتاژ و جریان عبوری از سیم پیچهای موتور بستگی دارد.سرعت موتور دي سي وابسته به ولتاژ و گشتاور آن وابسته به جريان است. معمولاً سرعت توسط ولتاژ متغیر یا عبور جریان و با استفاده از تپها(نوعی کلید تغییر دهنده وضعیت سیم پیچ) در سیم پیچی موتور یا با داشتن یک منبع ولتاژ متغیر ، کنترل می‌شود.به دلیل اینکه این نوع از موتورمی‌تواند در

سرعت هاي پايين گشتاوری زیاد ایجاد کند معمولااز آن در كاربردهاي كششي نظير لوكوموتيوها استفاده مي كنند.

ساختمان موتور DC

قسمت هاي اصلي موتوردي سي عبارتند از:

۱- آرميچر

۲- قطب هاي ميدان تحريك

۳- قاب بدنه

۴- كاسه موتور(قالپاق ها)

۵- جاروبك هاي نگه دارنده

انواع موتورهاي دي سي

۱- موتورهاي سري

۲- موتور شنت(موازي)

۳-موتورهاي كمپوند

این سه نوع از نظر شکل ظاهری مشابه اند. تفاوت آنها در ساختمان سیم پیچی کلافهای میدان تحریک و اتصال بین کلافهای میدان و آرمیچر است.

موتورهاي سري

موتورهای سری شامل سیم پیچ های میدان متشکل از چندین دورسیم که به طور سری با آرمیچر قرار گرفته اند ، می باشد که در شکل دیده می شود. . این موتور دارای گشتاور راه اندازی زیاد و مشخصهٔ سرعت متغیر می باشد . هر چه بار بیشتر باشد ، سرعت آن کاهش می یابد . موتور سری عموماً در جرثقیل های معمولی و کابل دار ، قطارهای برقی و غیره به کار می رود

موتور شنت (موازی)

شامل یک میدان متشکل از تعداد زیادی دورسیم پیچی است که به طور موازی با آرمیچرمتصل شده است و در شکل دیده می شود. این موتور دارای گشتاور متوسط و مشخصهٔ سرعت ثابت می باشد و برای کاربردهایی که احتیاج به سرعت ثابت دارند ، از قبیل پرس مته ای و دستگاههای تراش و غیره به کار می رود.

موتور کمپوند

در موتورهای کمپوندکه در شکل دیده مي شوند،هرکلاف میدان ترکیبی ازمیدانهای سری وشنت بوده ومتشکل ازدوقسمت می باشد. یک قسمت(میدان سری) به طورسری به آرمیچرمتصل شده وقسمت دیگر(میدان شنت) به است.مشخصات این موتور ترکیبي از موتور سري وموتور شنت مي باشد.

ساختمان کلافهای میدان تحریک

کلافهای میدان سری با دورنسبتاً کم ازسیم کلفت پیچیده می شوند که قطرسیم بستگی به قدرت ولتاژ موتور دارد

کلاف های میدان های شنت شامل تعداد دورهای زیاد از سیم نازک می باشند

کلاف میدان کمپوند ترکیبی از میدان سری و شنت می باشد

اتصال قطب های میدان :

در موتورهای دي سي کلاف های میدان طوری به هم متصل می شوند که پلاریته قطب ها یک در میان شمال و جنوب باشد . بنابراین در موتور دو قطبی یکی از قطبها شمال و دیگری در جنوب است و در یک موتور چهار قطبی قطبها باید یک در میان شمال و جنوب باشد.

موتورهای سري اتصال

موتورهاي سری به طوری که در شکل نشان داده شده متصل میگردد. میدانها به طور سری متصل شده اند و سپس به طور سری با آرمیچر قرار گرفته اند

معكوس كردن جهت گردش موتورهای دي سي

جهت چرخش موتورهاي جريان مستقيم با تغيير جهت جريان آرميچر يا ميدان معكوس می شود . در موتورهای سری روش معمول اين است كه جهت جريان در آرميچر را معكوس كنيم.

تغيير جهت چرخش يک موتور شنت همانند موتور سری انجام می شود

- روشهاي كنترل سرعت

رابطه سرعت – گشتاور موتورهاي dc معادله (۱-۵) نشان مي دهد كه سرعت را مي توان با هر كدام از روشهاي زير كنترل نمود :

۱- كنترل ولتاژ آرميچر

۲- كنترل شار ميدان

۳- كنترل مقاومت آرميچر

-۱ كنترل ولتاژ آرميچر

اگر ولتاژ آرميچر يك موتور dc تحريك جداگانه يا تحريك سري كه در يك سرعت پايدار كار مي كند به مقدار كمي كاهش يابد ، آنگاه جريان آرميچر و بنا براين گشتاور موتور كاهش خواهند يافت . چون گشتاور موتور از گشتاور بار كوچكتر خواهد بود ، شتاب

موتور منفي خواهد بود. در نهايت موتور در سرعتي كمتر كه در آن گشتاور موتور و بار برابر هستند مستقر مي شود. اگر ولتاژ آرميچر يك موتور تحريك جداگانه به مقدار بزرگي كاهش يابد ،ممكن است از ولتاژ ضد محركه كوچكترشود .جريان آرميچر معكوس شود و موتور همانند يك ژنراتور كار كرده و گشتاور منفي توليد كند . اين وضيعت ادامه خواهد يافت تا سرعت بحدي كاهش يابد كه نيروي ضد محركه موتور با ولتاژ اعمال شده برابر شود . پس از آن نحوه ادامه كار موتور مشابه حالت اول است كه توضيح داده شد . در مورد يك موتور سري ، حتي هنگاميكه ولتاژ آرميچر تغيير پله اي بزرگ داشته باشد ، موتوربه صورت ژنراتور كار نمي كند و كاهش سرعت آن فقط به آن علت است كه گشتاور موتور از گشتاور بار كمتر است.

از طرف ديگر ، اگر ولتاژ آرميچر يك موتور dc كه در حالت دائمي كار مي كند ، افزايش يابد .جريان آرميچر ، و بنابراين گشتاور موتور افزايش خواهند يافت . و موتور شتاب خواهد گرفت ، نتيجتا" سرعت موتور ونيروي ضد محركه افزايش خواهند يافت . در نهايت موتوردر سرعتي بالاتر كه در آن گشتاور موتور با باربرابر مي شود مستقر خواهد شد .

در روند افزايش يا كاهش سرعت ، تغييرات پله اي ولتاژ آرميچر بايستي كوچك باشد . يك تغيير بزرگ در ولتاژ آرميچر باعث ايجاد مقادير بزرگ جريان در آرميچر مي شود كه ممكن است به كموتاتور آن آسيب رسانده و عمر آن كاهش يابد .

با كاهش ولتاژ آرميچر ، موتور مي تواند بر روي هر كدام از منحني ها كه بين منحني سرعت- گشتاور طبيعي و محور گشتاور قرار گيرد ، كاركند. براي يك موتور تحريك جداگانه ، سرعت بي باري نيز تغيير مي كند و مشخصه هاي سرعت – گشتاور براي ولتاژهاي مختلف خطوط موازي هستند . چون ولتاژآرميچررا نمي توان از مقدار نامي بيشتر نمود اين روش كنترل سرعت فقط براي كار موتور در زير مشخصه سرعت – گشتاور طبيعي به كار مي رود .

ويژگي مهم اين روش كنترل سرعت آن است كه شكل و شيب مشخصه هاي سرعت – گشتاور با تغيير سرعت عوض نمي شوند . با اين روش يك محركه گشتاور ثابت بدست مي آيد چونكه حد اكثر جريان مجاز آرميچر ، و بنا براين حداكثر ظرفيت گشتاور موتور براي تمام سرعتها ثابت باقي مي ماند .

ولتاژ dc متغيير با استفاده از هركدام از مبدل‌هاي نيمه هادي زيرمي تواند به دست آيد:

۱- يكسو كننده كنترل شده (يا مبدل ac به dc).

۲- برشگر (يا مبدل dc به dc).

- كنترل ميدان

اگر در يك موتور تحريك جداگانه يا سرعت خاصي مي چرخد، ميدان تضعيف شود ، نيروي ضد محركه القايي آن كاهش مي يابد . به دليل كوچك بودن مقاومت آرميچر مقدار افزايش در جريان آرميچر نسبت به مقدار كاهش ميدان بسيار بزرگتر خواهد بود . در نتيجه

با وجود تضيعف ميدان گشتاور بطور قابل ملاحظه اي افزايش مي يابد به نحوي كه از گشتاور بار بيشتر مي شود . فزوني گشتاور موتور بر گشتاور بارموجب شتاب گيري موتور و افزايش ولتاژ القايي آرميچر مي شود . در حاليكه ميدان موتور تضعيف شده نهايتا موتور در سرعتي بالاتر از سرعت قبل مستقر مي شود ، كه در آن گشتاور موتور با گشتاور بار برابر است. هر تضعيف شديدي در ميدان منجر به ايجاد جريان هجومي خطرناكي مي‌شود لذا تضعيف ميدان بايستي به آرامي و به تدريج انجام شود .

از طرف ديگر ، هنگامي كه ميدان يك موتور تحريك جداگانه زياد مي شود ، ولتاژ القايي افزايش مي يابد و اغلب از ولتاژ منبع بيشتر مي شود . پس ، جريان آرميچر نه فقط كاهش مي يابد بلكه جهت آن نيز اغلب عوض مي شود . در اينحالت موتور همانند يك ژنراتور كار مي كند و به منبع انرژي مي دهد . اين انرژي از انرژي جنبشي موتور و بار گرفته مي شود . يك كاهش سريع در سرعت رخ مي دهد و نهايتا موتور در يك سرعت جديد كمتر از سرعت قبل ، مستقر مي شود كه در آن گشتاور موتور و بار برابر هستند . در موتور سري افزايش ميدان ، جريان آرميچر را به نسبت بيشتري افزايش مي دهد (اما جهت آن عوض نمي شود) . چون گشتاور موتور از گشتاور بار كمتر است ، سرعت موتور كم مي شود و نهايتا در سرعتي پايين تر مستقر مي شود كه در آن گشتاور موتور و بار هم برابرهستند.

در يك شار تضعيف شده ، براي يك افزايش معين در گشتاور ، جريان آرميچر . بنابر اين افت ولتاژ آرميچر به نسبت بيشتري افزايش مي يابند . نتيجتا نيروي ضد محرك القايي ، و بنا بر اين سرعت نيز به نسبت بيشتري افت مي كنند . پس هر چه شار كمتر باشد ، شيب

منحني ها ي سرعت – گشتاور بيشتر است . در مقادير كم شار ، اگر مقدار گشتاور كم باشد، يك كاهش در شار ممكن است حتي منجر به كاهش در سرعت بشود .

در مورد يك موتور شنت ، كمترين سرعت قابل حصول متناظر با حد اكثر تحريك و بدون حضور هيچ مقاومت خارجي در مدار تحريك است. در مورد يك موتور تحريك جداگانه كمترين سرعت توسط گرماي توليد شده در سيم‌بندي تحريك و اشباع مدار مغناطيسي معين مي‌شود چون در تحريك كامل ، ماشينهاي جديد در مقدار قابل ملاحظه‌اي از اشباع مدار مغناطيسي كار مي‌كنند. سرعت فقط به مقدار كمي در زير مشخصه سرعت- گشتاور طبيعي مي تواند كاهش يابد . حداكثر سرعت توسط ناپايداري موتور ناشي از اثر ضد مغناطيسي عكس العمل آرميچر در يك ميدان ضعيف و همچنين تحمل مكانيكي موتور محدود مي شود .

در موتورهاي dc با طراحي عادي ، رساندن سرعت به ١/٥ تا ٢ برابر سرعت نامي و در موتورهاي با طراحي مخصوص ، رساندن سرعت به ٦ برابر سرعت نامي امكان پذير است. براي جلوگيري از ناپايداري ، در موتورهاي تحريك جداگانه از يك سيم بندي سري با ميدان نسبتا ضعيف كه به ميدان اصلي كمك مي‌كند، استفاده مي شود. در بارهاي سنگين لحظه‌اي، يك جريان بزرگ، ميدان را تقويت ميكند و سرعت موتور تمايل به كاهش پيدا مي كند .

كنترل ميدان موتورهاي تحريك جداگانه و شنت ، يك كنترل در قدرت ثابت فراهم مي آورد، چونكه حداكثر قدرت موتور در تمامي سرعتها تقريبا" باقي ثابت مي ماند . فرض مي شود كه حداكثر جريان مجاز آرميچر هنگاميكه ميدان تضعيف مي شود، عوض نمي

شود . در جريان آرميچر برابر با ، ولتاژ ضد محركه القايي E در تمام سرعتها به علت ثابت بودن ولتاژ تغذيه در مقدار v ، ثابت باقي مي ماند . در نتيجه حداكثر قدرت حاصله موتور EI در تمام محدوده سرعت ثابت باقي مي ماند و حداكثر گشتاور بطور معكوس با سرعت تغيير مي كند.

اين فرض كه حداكثر جريان مجاز آرميچر با كاهش شار عوض نمي شود ، بطور تقريبي قابل قبول است . زيرا عكس العمل آرميچر در شرايطي كه شار اصلي كاهش مي يابد بنحو موثرتري بر شار تاثير مي گذارد ، بنا براين حداكثر جرياني كه موتور مي تواند عبور دهد بدون آنكه در كموتاتور آن جرقه ايجاد شود كاهش مي يابد ، و موجب كاهش حداكثر قدرت مجاز توليدي در سرعتهاي بالا مي شود . در يك موتور تحريك جداگانه، كنترل شار با تغييرات ولتاژ سيم پيچي تحريك توسط يك يكسو كننده قابل كنترل يا يك برشگر ، بسته به اينكه منبع موجود acيا dc باشد انجام مي شود .

موتورهاي با اندازه كوچك به صورت شنت بسته مي شوند ، و تغييرات شار با وارد كردن يك مقاومت متغير درمدار تحريك به دست مي‌آيد.در يك موتور سري ، كنترل‌شار با اتصال يك مقاومت انشعابي به دو سر سيم پيچي تحريك ، حاصل مي‌شود.

-۳ تركيب روشهاي كنترل ولتاژ آرميچر و ميدان

در محركه‌هايي كه كنترل سرعت در محدوده اي وسيع ضروري است ، دو روش كنترل ولتاژ آرميچر و ميدان با هم تركيب مي شوند . در روش كنترلي ولتاژ آرميچر امتياز ثابت

ماندن حداكثر ظرفيت گشتاوري ماشين در تمامي سرعتها وجود دارد . لذا در هر جايي كه امكان داشته باشد اين روش بكار گرفته مي شود ، و از روش كنترل ميدان براي دستيابي به سرعتهايي كه با روش كنترل ولتاژ آرميچرقابل حصول نيستند ، استفاده مي شود . در چنين محركه هايي ، سرعت مربوط به حالتي كه ولتاژ آرميچر در مقدار نامي و تحريك هم كامل است ، سرعت مبنا ناميده مي شود. اين سرعتي است كه موتور در آن بر روي مشخصه سرعت – گشتاور طبيعي كار مي كند . از حالت سكون تا سرعت نامي تغييرات سرعت با كنترل ولتاژ آرميچر به دست مي آيد و در اين محدوده سرعت ، ميدان در مقدار نامي خود ثابت نگاه داشته مي شود . سرعتهاي بالاتر از سرعت مبنا با روش كنترل ولتاژ آرميچر نمي توانند به دست آيند چونكه ولتاژ آرميچر موتور نبايستي از مقدارنامي بيشتر شود . بنا براين ، سرعتها ي بالاتر از سرعت مبنا با روش كنترل ميدان به دست مي آيند اين نوع محركه ها عبارتند از غلطكها ي نورد ، كاربردهاي كششي (قطارها) و غيره .

- كنترل مقاومت آرميچر

اشكال اصلي اين روش كنترل سرعت بازده كم آن مي باشد . براي مثال ، براي باري با گشتاور ثابت كل قدرت ورودي به موتور (تحريك سري و جداگانه) و مقاومت سري، مقدار ثابتي است ، درهمان در حاليكه قدرت تحويلي به بار متناسب با سرعت كاهش مي يابد . بنا بر اين درصد بازدهي موتور همان درصد سرعت نسبت به سرعت نامي آن است، بدين معني كه در سرعتي معادل ۱۰ درصد سرعت نامي بازدهي موتور دقيقا ۱۰ درصد است.

در روش كنترل آرميچر شكل منحني هاي سرعت – گشتاور در يك موتور تحريك جداگانه (يا شنت) از حالت سرعت تقريبا ثابت براي تمام گشتاورها به يك حالت سرعت متغير عوض مي شود . به همان دليل و به سبب بازدهي كم ، از اين روش براي كنترل موتورهاي تحريك جداگانه بندرت استفاده مي شود مگر براي رسيدن به سرعتها يي كه براي لحظات بسيار كوتاه ضروري باشند .

براي محركه هايي كه در سرعتها ي پائين و به صورت تكراري و كوتاه مدت كار مي‌كننده، كاهش بازدهي كه محركه زياد نخواهد بود . به دليل سادگي و پادين بودن هزينه اوليه ، اين روش براي محركه‌هايي با كار تكراري كوتاه مدت كه از موتورهاي سري استفاده مي كنند كاملا مناسب و اقتصادي است .

ـ راه اندازي

حداكثر جرياني كه از يك موتور dc در حالت هاي گذاري كوتاه مدت مي تواند عبور كند ، جرياني است كه يك كمو تاسيون بدون جرقه داشته باشد و از نقطه نظر تئوري ، با بكار گيري سيم پيچي هاي جبران ساز در تمامي مقادير سرعت و جريان مي توان ولتاژ هايي را كه با كموتا سيون جريان مخالفت كرده و ايجاد جرقه مي كنند ، بطور كامل حذف نمود. اما در عمل مشاهده شده است كه با افزايش مقدار جريان ،كامل انجام نمي شود و با عبور جريان از يك حد معين ، جرقه پديدار مي شود.

اگر موتور با ولتاژ نامي راه اندازي شود ، براي يك موتور با اندازه متوسط ، جريان به حدود ۲۰ برابر جريان نامي خواهد رسيد . جرياني به اين بزرگي منجر به جرقه هاي شديد

در كموتا تور و افزايش بيش از حد درجه حرارت در سيم پيچ هاي موتور شده و به آن آسيب مي‌رساند . بنا بر اين محدود نمودن جريان به يك حد بدون خطر در زمان راه‌اندازي و ضروري مي شود . اينكار با كاهش ولتاژ دو سر موتور در لحظه راه اندازي و افزايش تدريجي آن با سرعت گرفتن موتور حاصل مي‌شود .

ولتاژ موتور با كاهش ولتاژ منبع يا با ايجاد افت قسمتي از ولتاژ منبع بر روي يك مقاومت سري شده با موتور، انجام مي‌شود. در كاربردهايي كه به سرعت قابل تنظيم نياز دارند، يك كنترل كننده براي كنترل سرعت موتور فراهم مي شود . همين كنترل كننده‌ها براي محدود نمودن جريان موتور در مدت راه اندازي مي تواند گرفته شود . در مواردیكه كنترل سرعت ضروري نيست ، براي محدود نمودن جريان از يك راه انداز استفاده مي‌شود. در موارد يكه راه اندازي مكرر لازم نيست، با قرار دادن يك مقاومت اضافي چندين قسمتي درمدار آرميچر و خروج تدريجي آن از مدار به نحوي كه جريان موتور از حد سالم و بي خطر بيشتر نشود و در ضمن گشتاور توليدي موتورهمواره از گشتاور بار بزرگتر بماند، راه اندازي انجام مي شود. اين روش بطور گسترده بكار گرفته مي شود.

– توصيه هايي درمورد موتورهاي برق :

١- انتخاب موتورهاي پربازده براي بهره برداري بلند مدت و متناسب با گشتاور مكانيكي مورد نياز

٢- محل استقرار موتورها مي بايد به گونه اي باشد , كه گرماي حاصل از موتورها به آساني تهويه شود.

٣- بررسي مقدار توان راكتيو و در صورت نياز طراحي و نصب خازن مناسب در كنار مصرف كنندگان مجهز به موتور

٤- بررسي استفاده از مبدل فركانس براي تغيير سرعت موتورهاي آسنكرون به تناسب نياز

٥- بهتر است موتورها با بارنامي كار كنند و با برنامه ريزي مناسب از قطع و وصل بيش از حد آنها جلوگيري شود (خاموش بودن موتورها در ساعات غيرضروري)

نگهداري موتورهاي جريان مستقيم :

موتورهاي جريان مستقيم به علت عمل يكسو سازي مكانيكي كه درآنها انجام ميشود , داراي اجزايي هستند كه پيوسته به تعمير و نگهداري نياز دارند . براي حفظ يا ثابت نگه داشتن بازده موتور , رعايت موارد زير بايسته است :

الف) بازديد كموتاتور (از لحاظ تشكيل لايه قهوه اي بر روي آن)

ب) زدودن آلودگي روي كموتاتور

ج) تراشيدن لايه هاي ميكا در بين هاديهاي كموتاتور و تراشيدن لبه هاديها

د) تعويض كل جارو بكهاي ماشين در صورت نياز به تعويض بعضي از آنها

تعمیر موتور وعیب یابی اشکالات ممکنه:

<u>۱- اگر موتور با روشن کردن کلید بکار نیفتد ، عیب ممکن است به علل زیر باشد :</u>

الف – فیوز یا مدار محافظتی باز است .

ب- جاروبکهای کثیف یا عایق شده است .

ج- میدان تحریک ، اتصال کوتاه شده یا ببدنه وصل می باشد .

د- آرمیچر یا کلکتور اتصال کوتاه شده است.

ه- یاطاقانها ، فرسوده شده اند.

و- جاروبک نگهدار ببدنه وصل شده است .

ز- اضافه بار موجود است.

ح- کنترل کننده خراب و ناقص است .

<u>۲ - اگر موتور کند کار کند ، عیب ممکن است بعلل زیر باشد :</u>

الف – آرمیچر یا کلکتور اتصال کوتاه شده .

ب- یاطاقانها ، فرسوده شده اند.

ج- کلافهای آرمیچر قطع شده است.

د- جاروبکها خارج از وضعیت خنثی قرار گرفته اند.

ه- اضافه بار موجود است.

ولتاژ غلط است

۳- اگر موتور سریعتر از سرعت اسمی بچرخد ، عیب ممکن است بعلل زیر باشد :

الف – مدار میدان شنت باز است .

ب- موتور سری بدون بار کار می کند.

ج- میدان تحریک اتصال کوتاه شده یا ببدنه وصل میباشد .

د- اتصال نقصانی در یک موتور کمپوند.

۴ -اگر موتور جرقه بزند، عیب از علل زیر باشد :

الف – تماس ضعیف جاروبکها روی کلکتور .

ب- کلکتور کثیف شده است.

ج- پلاریته غلط قطب کمکی .

د- میدان اتصال کوتاه شده یا ببدنه اتصال یافته .

و- سرهای سیم آرمیچر برعکس شده است.

ز- اتصال غلط سرهای سیم.

ح- جاروبکها در خارج از محل خنثی قرار گرفته اند.

ط- مدار میدان قطع (باز) است .

ی- تیغه های کلکتور بلند و کوتاه است.

ک- عایق میکا بین تیغه ها بلند تر از تیغه های کلکتور است.

ل- آرمیچر نامتعادل است.

۵- اگر موتور با سر و صدا کار کند ، عیب ممکن است از علل زیر باشد :

الف – یاطاقانها ، فرسوده شده اند. .

ب- تیغه های کلکتور بلند و کوتاه است.

ج- کلکتور ناصاف است.

د- آرمیچر نامتعادل است .

۶- اگر موتور در هنگام کار داغ کند، عیب ممکن است از علل زیر باشد :

الف – اضافه بار موجود است .

ب- موتور جرقه می زند

ج- یاطاقانها سفت (تنگ) است .

د- کلافها اتصال کوتاه شده است .

و- فشار بیش از حد روی جاروبکها وجود دارد

_ عوامل فرسودگي و كاهش بازده موتورهاي برق :

الف) آلودگي محيط كارموتور براثر گردوغبار , رطوبت و بخار روغن

ب) افزايش بيش از اندازه درجه حرارت محيط كار موتور

ج) عدم روغنكاري منظم

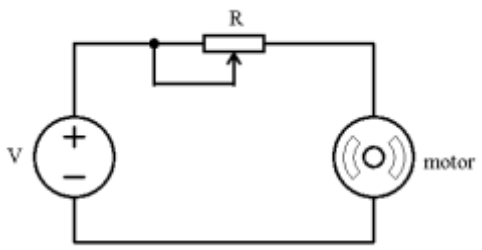
کاربرد موتورdc

از موتورهاي جريان مستقيم مغناطيس دايم به علت كارآيي بالا، سهولت استفاده، آسان نمودن كارها، گشتاور راه اندازی بالا، راه اندازی در زير بار و ... به وفور استفاده مي‌شود.

از جمله موتورهاي برف پاک کن، فن رادياتور، شيشه بالابر،پمپ شيشه شوي، محرک‌هاي فقل دربها، آنتن ، محرک صندلي، محرک سفق، تغيير وضعيت چراغها در صنعت خودروسازی، موتور های اسياب ، اب ميوه گيری، موتور سشوار، در لوازم الكترونيكی بعنوان فن، در لوازم صوتی ، در دندان پزشكی ، در كاربردهاي كششي نظير لوكوموتيوها ، منبع انرژي براي تحريك مولدهاي نيروگاهي ، ماشينهاي خودكار، هواپيماها، جوشكاري با قوس الكتريكي، قطارهاي راه آهن، اتوبوسهاي برقي، زير دريايیها و ... اشاره كرد

کنترل سرعت در موتورهای DC:

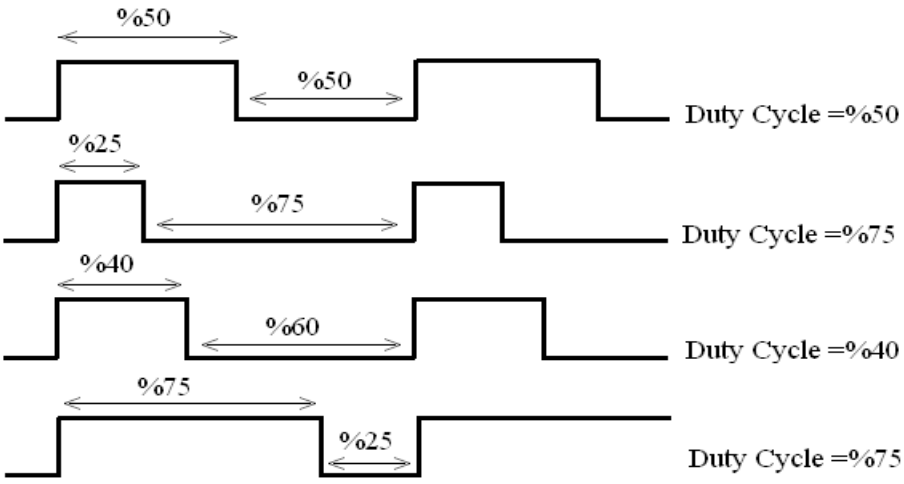
ساده ترین روش برای کنترل موتورهای DC استفاده از یک رئوستا که به صورت سری با بار قرار گرفته شده است می باشد. این رئوستا ولتاژ اعمالی به بار را تغییر می دهد.



کنترل موتور DC با استفاده از رئوستا

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

نحوه کنترل موتوربه وسیله مدولاسیون عرض پالس بدین گونه است که ابتدا یک فرکانس ثابت راانتخاب کرده و سپس برای افزایش سرعت موتور Dutycycle را افزایش و برای کاهش سرعت Dutycycle 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 Dutycycle های مختلف بدست می آید.

$$Vrms = \sqrt{\frac{1}{T}\int_{0}^{T} V(t)^2\, dt}$$

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.همانطورکه 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.

پیوست ۳

برگه اطلاعات قطعات استفاده شده