به نام خدا

سیستم‌های نهفته بیدرنگ

پروژه کامپیوتری اول

فاطمه ایزدی نژاد

سپیده رحیمیان

یاسمین رحیمی صادق

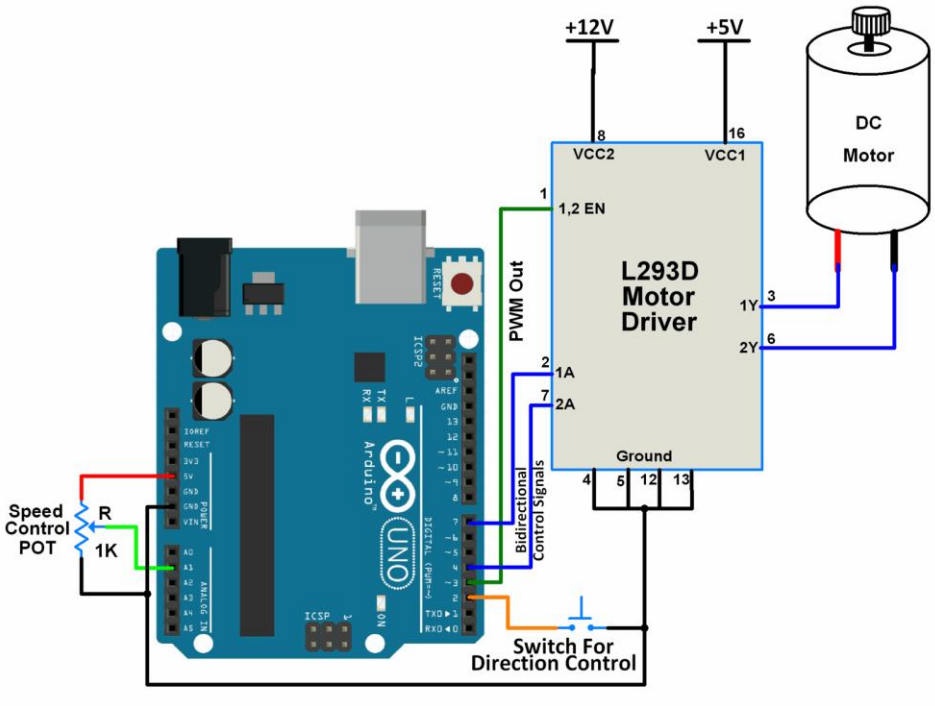
ریحانه گلی

دانشگاه تهران - پاییز ۹۹

مرحله اول- موتور DC

با استفاده از بلوک ARDUINO UNO در در نرم افزار proteus و یک driver از نوع L293D و موتور DC مداری طراحی کردیم که به واسطه آن بتوانیم چرخش یا عدم چرخش موتور و جهت چرخش آن (ساعتگرد یا پادساعتگرد) را کنترل کنیم.

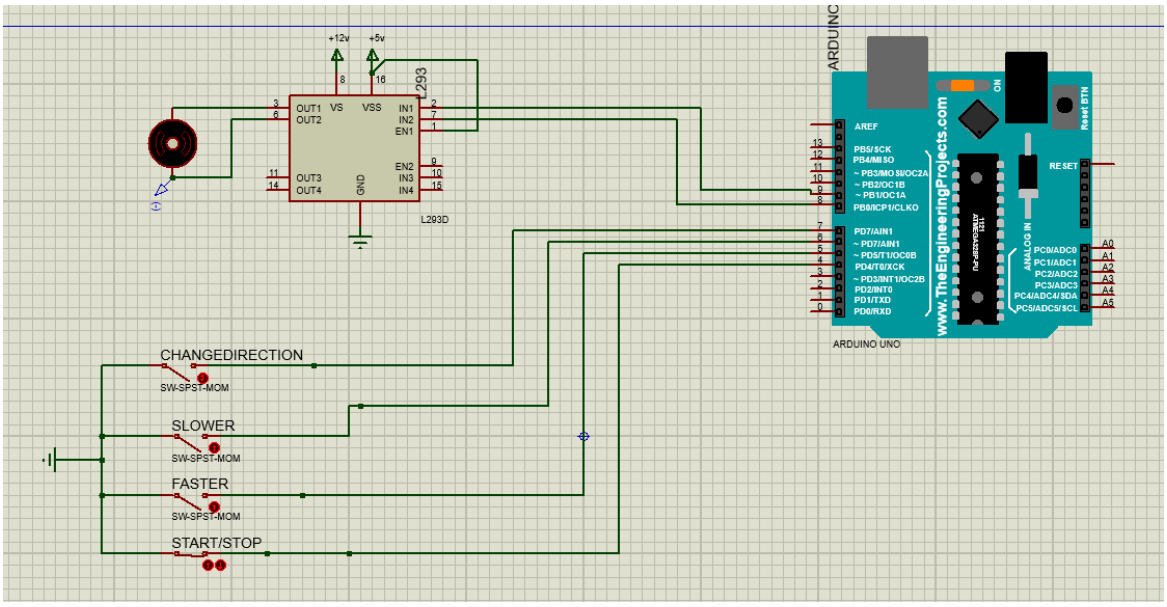
مدل مفهومی مدار:



به این منظور در ابتدا با اتصال پورت های دیجیتال آردوینو به ورودی های driver گفته شده (در این جا از پایه های 8 و 9 استفاده شده است.) و اتصال خروجی های آن به دو سر موتور ، مشکل پایین بودن جریان درگاه های بورد Arduino را حل کردیم. به این صورت که همان طور که گفته شد این driver هم به میکروکنترلر جهت دریافت سیگنال های کنترلی متصل است و هم برای دریافت جریان کافی به منبع تغذیه ای اتصال دارد.

برای کنترل چرخش موتور، چهار سوئیچ قرار داده شده است. با استفاده از سوئیچ START/STOP چرخش موتور متوقف و یا مجددا آغاز می‌شود و همچنین با استفاده از سوئیچ CHANGEDIRECTION جهت چرخش موتور تغییر می‌کند. همچنین سوییچ‌های FASTER و SLOWER به ترتیب سرعت چرخش موتور افزایش و کاهش می‌یابد.

طراحی با کمک نرم‌افزار Proteus:



عملکرد این موتور متشکل از وضعیت چهار دکمه است: دکمه روشن یا خاموش بودن، دکمه افزایش سرعت،دکمه کاهش سرعت و تغییر جهت.فشرده بودن این دکمه(پایین بودن آن) نشان دهنده‌ی روشن بودن آن است.در صورت روشن بودن دکمه روشن چراغ مربوط به موتور را روشن کرده و متناسب با مقدار clockwise، موتور یک یا دو روشن می‌شوند.

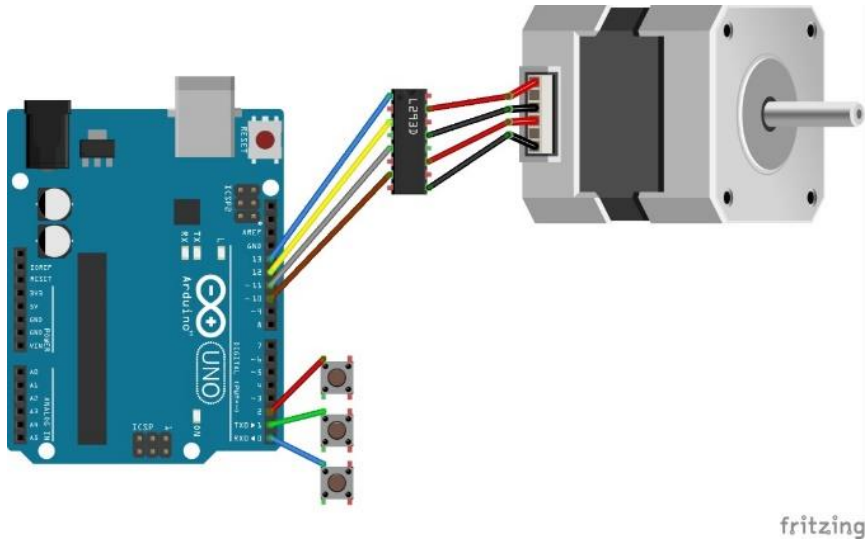
در صورت روشن بودن دکمه افزایش سرعت متغیر pwm را ده تا ده تا بالا می‌بریم و در برای دکمه کاهش سرعت آن را ده تا ده تا کم می‌کنیم.در هر دو صورت این متغیر باید مقداری بین ۰ و ۲۵۵ داشته باشد.

دکمه تغییر جهت clockwise را تغییر می‌دهد و اگر در جهت ساعتگرد باید به صورت پادساعت و برعکس تبدیل می‌شود.

مرحله دوم- موتور Stepper

در این مرحله با استفاده از بلوک Arduino UNO در نرم افزار proteus و یک driver از نوع L293D و موتور stepper در حالت Bipolar مداری طراحی کردیم که به واسطه آن بتوانیم چرخش یا عدم چرخش موتور و جهت چرخش آن (ساعتگرد یا پادساعتگرد) را کنترل کنیم.

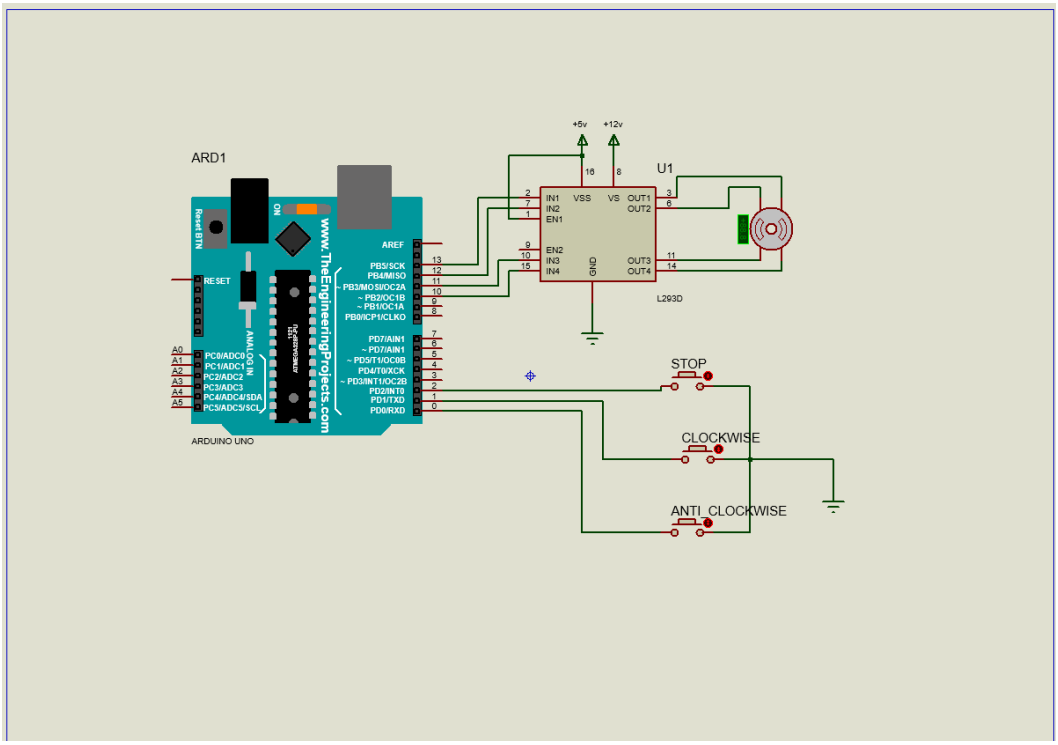
مفهومی مدار:



به این منظور در ابتدا با اتصال پورت های دیجیتال آردوینو به ورودی های driver گفته شده (در این جا از پایه های ۱۰ و ۱۱ و ۱۲ و ۱۳ استفاده شده است.) و اتصال خروجی های آن به دو سر موتور ، مشکل پایین بودن جریان درگاه های بورد Arduino را حل کردیم. به این صورت که همان طور که گفته شد این driver هم به میکروکنترلر جهت دریافت سیگنال های کنترلی متصل است و هم برای دریافت جریان کافی به منبع تغذیه ای اتصال دارد.

برای این که کاربر بتواند چرخش موتور را کنترل نماید از سه دکمه جانبی استفاده شده است. کاربر با استفاده از فشردن کلید STOP می تواند چرخش موتور را متوقف کند و همچنین با استفاده از کلید های CLOCKWISE و ANTI_CLOCKWISE به ترتیب می تواند حرکت موتور را در جهت حرکت عقربه های ساعت و خلاف جهت حرکت عقربه های ساعت کنترل کند.به این منظور لازم است که یک سر کلید ها به بورد Arduino (پورت های ۰ و ۱ و ۲) و سر دیگر آن ها به زمین متصل شده باشند.

طراحی با کمک نرم‌افزارProteus:



برای ارسال دستور های کنترلی، بعد از فشرده شدن کلید ها و از طریق میکروکنترلر به موتور می بایست برنامه ای نوشت و آن را روی بورد Arduino اجرا کرد. در ابتدا شبه کدی برای آن طراحی شده است که در قسمت زیر قابل مشاهده می باشد:

```
main function definition
{
        //Initialization for motor state and pins;
        while(true)
        {
                if (stop_pin)
                        stop the motor;
                if (clockwise_pin)
                        change direction of motor to clockwise;
                if (anticlockwise_pin)
                        change direction of motor to anticlockwise;
                update state;
        }
}
```

به این منظور در ابتدا متغیری با نام steps_per_revolution تعریف شده است که نشان دهنده ی تعداد step هایی است که موتور در هر یک بار چرخش (۳۶۰ درجه) طی می کند. از آن جایی که می خواهیم حرکت موتور در فاصله های ۱۰ درجه ای باشد این مقدار از رابطه ی ۳۶=۳۶۰/۱۰ به مقدار ۳۶، مقداردهی شده است. همچنین برای مشخص کردن پورت های متصل به کلید ها از تعریف متغیر هایی نظیر هر یک از آن ها و مقدار دهی آن ها به شماره پورت های مربوطه استفاده شده است. برای هر یک از سه دکمه استفاده شده در این طراحی متغیر هایی با نام های clockwise_pin , anticlockwise_pin , stop_pin در نظر گرفته شده است که در ابتدا همگی آن ها به ۰ مقدار دهی اولیه شده اند چرا که در ابتدا هیچ یک از آن ها توسط کاربر فشرده نشده اند. برای تعیین جهت حرکت موتور نیز متغیری با نام Direction داریم که در ابتدا مقدار آن ۱+ است چرا که می خواهیم در ابتدا و قبل از فشرده شدن کلید ها حرکت موتور در جهت حرکت عقربه های ساعت باشد.

در ابتدا در تابع setup متغیر های clockwise_pin , anticlockwise_pin , stop_pin به صورت INPUT PULLUP تعریف شده اند در نتیجه برای آن ها یک مقاومت درونی در نظر گرفته شده است. به همین دلیل قرار دادن مقاومت در مدار طراحی شده به طور جداگانه لازم نیست.در این بخش می توانستیم این متغیر ها را به صورت INPUT تعریف کنیم و در مدار از مقاومت های با مقدار یکسان بر سر هر یک از این کلید ها استفاده کنیم.

برای تعیین چرخش یا عدم چرخش موتور، flag ای با نام start در نظر گرفته شده است که در ابتدا به مقدار True ست شده است.تا زمانی که مقدار این flag برابر True باشد موتور به چرخش خود ادامه می دهد و در صورت False بودن از حرکت باز می ایستد.

در تابع loop سرعت موتور مقدار دهی شده است و موتور تا زمانی که مقدار flag برابر True باشد در جهت گفته شده به حرکت خود ادامه می دهد.

در هر زمان توسط تابع digitalRead مقدار کلید ها خوانده می شود. اگر این مقدار برابر ۰ باشد نشان دهنده آن است که کلید مربوطه فشار داده شده است.پس تا زمانی که این مقدار ۰ باشد، بسته به آن که چه کلیدی فشار داده شده است یکی از عملیات زیر انجام می شود:

۱اگر کلید STOP فشار داده شود مقدار Flag به False تغییر کرده و موتور از حرکت باز می ایستد.

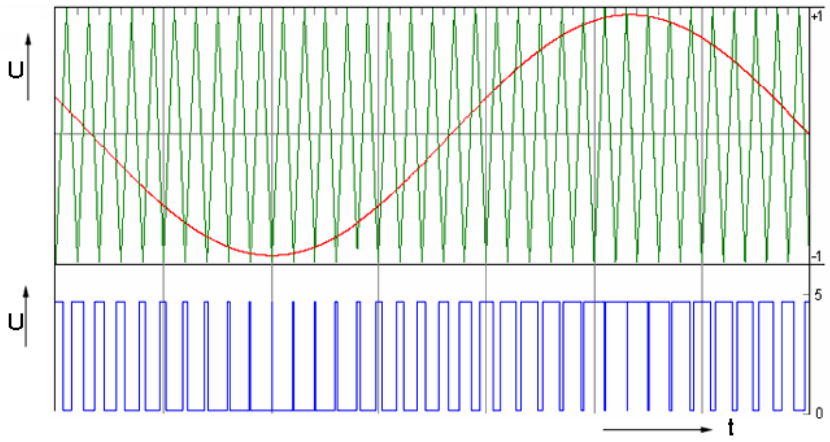
۲ اگر کلید CLOCKWISE فشار داده شود مقدار Flag به True تغییر کرده و مقدار متغیر Direction به دلیل چرخش در جهت حرکت عقربه های ساعت به ۱+ تغییر خواهد کرد.

۳اگر کلید ANTI_CLOCKWISE فشار داده شود مقدار Flag به True تغییر کرده و مقدار متغیر Direction به دلیل چرخش در خلاف جهت حرکت عقربه های ساعت به ۱- تغییر خواهد کرد.

پاسخ سوالات:

۱. نحوه‌ی ساخت PWM را شرح دهید.

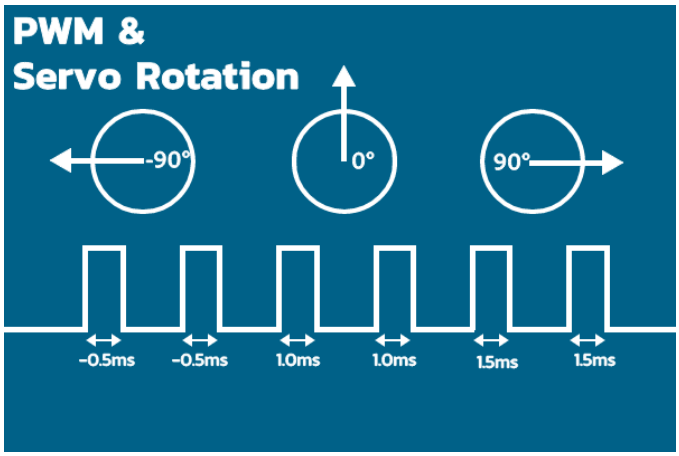
به طور کلی برای ساختن ماژول PWM از چند wave مانند sin , triangle,sawtooth و ramp استفاده می شود.همچنین برای ساخت wave نهایی به یک comprator نیاز داریم و wave های گفته شده را نیز می توانیم با generator های مربوط به هریک از آن ها و با فرکانس و دامنه های دلخواه تولید نماییم.به طور مثال در شکل زیر برای تولید سیگنال PWM از دو سیگنال triangle و sin استفاده شده است. و با استفاده از یک مقایسه کننده سیگنال نهایی را تولید کرده ایم. به طوری که در زمان هایی که سیگنال sin بالای سیگنال triangle قرار دارد(این کار با استفاده از comprator انجام می پذیرد) مقدار سیگنال نهایی برابر یک خواهد بود و زمان هایی که سیگنال sin زیر سیگنال triangle قرار دارد مقدار سیگنال نهایی برابر ۰ خواهد شد.

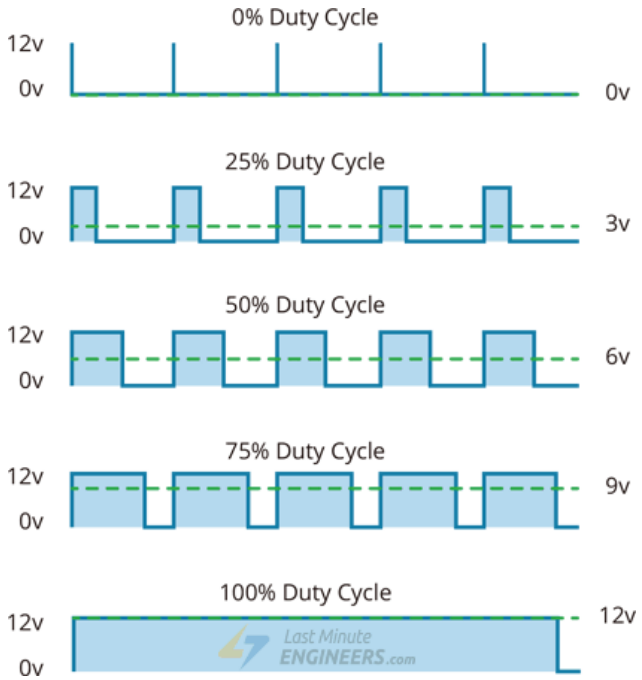


۲.استفاده از PWM در موتورهای DC و Servo چه تفاوتی دارد؟

معمولا سرعت موتورهای DC با استفاده از مدولاسیون عرض پالس (PWM) کنترل می‌شود. با استفاده از این روش ما به سرعت سیگنال را ON و OFF می‌کنیم. درصد زمانی که سیگنال ON است (نسبت به کل دوره تناوب سیگنال) سرعت موتور را تعیین می‌کند. به عنوان مثال، اگر سیگنال duty cycle برابر با ۵۰٪ داشته باشد (نیمه روشن و نیمه خاموش)، در این صورت موتور با نیمی از سرعت حداکثر سرعت خود می‌چرخد. هر پالس آنقدر سریع است که به نظر می‌رسد موتور بطور مداوم در حال چرخش است و وقفه ندارد.

بر خلاف موتورهای DC ، برای کنترل موتورهای servo از سیگنال PWM استفاده می‌شود. مدت زمان پالس مثبت (ON) موقعیت، و نه سرعت محور(shaft) موتور servo را تعیین می‌کند. مقدار پالس خنثی(OFF) که بستگی به موتور servo دارد(معمولاً حدود 1.5 میلی ثانیه) محور موتور servo را در موقعیت مرکزی نگه می دارد. افزایش مقدار پالس باعث می‌شود موتور servo در جهت عقربه های ساعت بچرخد و با یک پالس کوتاهتر محور در خلاف جهت عقربه ساعت قرار خواهد گرفت. پالس کنترل موتورهای servo معمولاً هر ۲۰ میلی ثانیه تکرار می‌شود (بستگی به نوع موتور دارد). اما ما همیشه باید به موتور servo بگوییم کجا باید برود، حتی اگر این به معنای باقی ماندن در همان موقعیت قبلی باشد.





۳.تفاوت موتورهای Servoو Stepper را شرح دهید و یک مورد کاربرد برای هر کدام بیان کنید.

موتورهای stepper چرخش کامل موتور را به چند مرحله (step) تقسیم می‌کنند. آنها این کار را با استفاده از یک آهنربای دائمی یا روتور آهنی و یک سری آهنرباهای الکتریکی اطراف آن انجام می‌دهند. این اجزا قسمت ساکن موتور را تشکیل می‌دهند که به آن استاتور نیز می‌گویند. موتور این آهنرباهای الکتریکی را با توالی دقیق و کنترل شده‌ای فعال می‌کند تا هر بار روتور را به صورت پله ای پیش ببرد. این فعال سازی نتیجه مدارهای driver مختلف است و آرایش مغناطیسی دقت موتور را تعیین می‌کند که معمولاً سطح دقت ۱،۸ درجه در هر مرحله (۳۶۰ درجه تقسیم بر ۲۰۰ مرحله مغناطیسی) را مشاهده می‌شود.

موتورهای servo ، مانند موتورهای stepper، دارای طیف گسترده ای از اشکال، اندازه‌ها و قیمت‌ها هستند. موتورهای servo صنعتی گرانتر می توانند هزاران دلار هزینه داشته باشند و دارای بازخورد (feedback) موقعیت و سرعت باشند. کاربران اغلب این موتورها را با جعبه دنده (گیربکس) جفت می‌کنند تا گشتاور آن‌ها را نیز افزایش دهند. شما می توانید با استفاده از یک سخت افزار کنترلی خارجی به این موتورها دستور دهید تا به موقعیت خاصی حرکت کرده و به صورت کنترل شده بپیچند تا از فرمان اطاعت کنند.

حلقه بسته (servo) در مقابل حلقه باز (stepper)

بین این دو موتور تفاوت اساسی در بازخورد یا feedback است. حال به تفاوت حلقه باز با حلقه بسته می پردازیم

- موتورهای stepper: بدون بازخورد هستند. موتورهای stepper می توانند یک انتخاب عالی باشند، اما در مورد اینکه آیا آن‌ها مسافت مورد نظر خود را طی کرده‌اند یا خیر بازخوردی ارائه نمی‌دهند. اگر هنگام چرخش در سیستم اشکال ایجاد شود یا جسمی مانع شود، آن‌ها می توانند "از برخی stepها بپرند"، سپس مثل حالتی که مشکلی وجود ندارد، به حرکت ادامه می‌دهند.

- موتورهای servo: دارای ویژگی بازخورد هستند. اگر به یک موتور servo دستور داده شود که به موقعیت خاصی برود، تمام تلاش خود را برای رسیدن به آن نقطه انجام می‌دهد. در setup های ضعیف‌تر، این ممکن است به این معنی باشد که یک موتور دائماً برای دستیابی به موقعیت خاص تلاش می‌کند و یا در سیستم های توانمندتر این امر معادل بازخورد خطای تقریباً فوری در شرایطی است که نتواند خود را اصلاح کند.

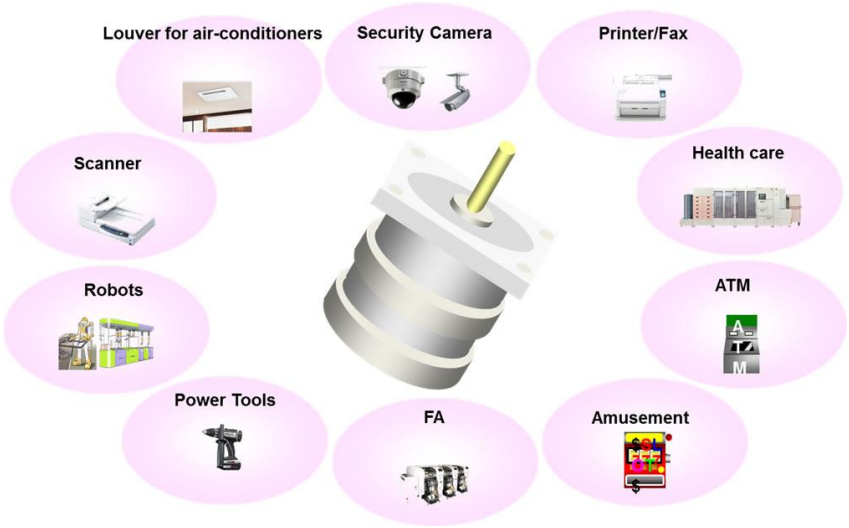
بسته به نوع کاربرد هر یک از این دو نوع موتور می‌توانند مناسب باشند:

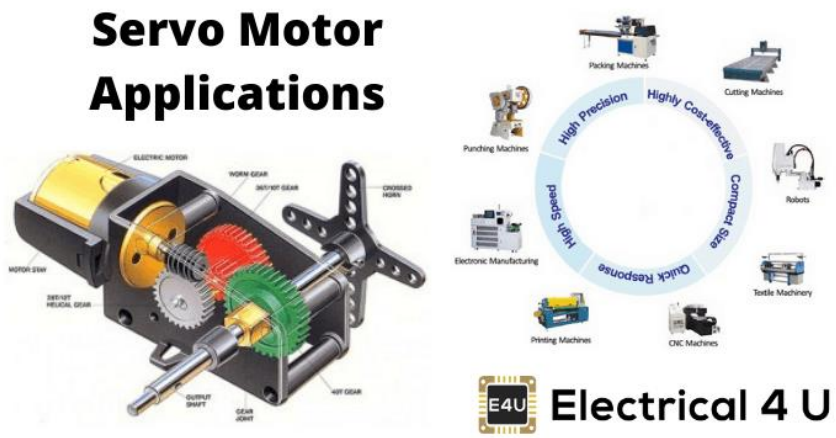
- حسگرها: اگر از stepper استفاده می کنید، احتمالاً باید روشی برای صفر کردن سیستم به یک نقطه مشخص داشته باشید تا سیستم بتواند وظیفه تعیین شده خود را انجام دهد. تنظیمات Stepper نیز ممکن است از این سنسور صفر یا سایر موارد برای تأیید مراحل خود استفاده کنند. افزودن حسگر می تواند به این معنی باشد که استفاده از موتورهای servo در طولانی مدت پیچیده تر یا گران خواهد بود.

- سرعت در برابر گشتاور: عامل دیگری که باید در نظر گرفت این است که موتورهای stepper با افزایش سرعت تمایل به از دست دادن گشتاور دارند، این بدان معنی است که یک موتور servo برای سرعت های بالاتر می‌تواند گزینه بهتری باشد. با این وجود موتورهای stepper در دورهای پایین گشتاور بسیار خوبی را اعمال می‌کنند. در واقع موتورهای stepper دقت بیشتر و موتورهای servo سرعت بیشتر دارند.

- دقت زاویه‌ای: اگر برنامه شما به سرعت بیش از دقت زاویه ای بستگی دارد ، انتخاب دیگر یک موتور DC به اصطلاح brushless است. آنها به عنوان نوعی موتور کنترل ترکیبی عمل می کنند که برای سرعت بهینه شده است - مثلاً دقیقاً تا 2000 دور در دقیقه می‌چرخند.

برای مثال در تجهیزات پرینت ۳ بعدی و یا ماشین‌آلات نساجی از موتور stepper استفاده می‌شود. از موتورهای servo در ماشین‌های کنترلی اسباب بازی و یا در صنعت در ربات‌ها و یا هواپیماها استفاده می‌شود.





۴.در بسیاری از تلفن های همراه، برای ساخت vibrator از ساختاری به نام ERM استفاده می شود. این ساختار از چه نوع موتوری استفاده میکند؟ ساختار کلی آن را شرح دهید.

موتور استفاده شده در ERM از نوع DC motor می باشد. با چرخش ERM ، نیروی گریز از مرکز ایجاد شده و این باعث جابجایی موتور می شود. با تعداد زیاد دور در دقیقه ، موتور به طور مداوم در حال جابجایی است و توسط این نیروها جابجا می شود. این جابجایی مکرر است که به عنوان ارتعاش درک می شود.

ارتعاش تولید شده توسط ERMs نمونه ای از "لرزش هارمونیک رانده شده" است. این بدان معنی است که یک نیروی محرکه خارجی وجود دارد که باعث لرزش سیستم می شود

در مورد مدل ERM ، محرک ، ولتاژ DC اعمال شده بر روی موتور نیست. در عوض ، چرخش جرم در اطراف شافت موتور مرکزی است.