به نام خدا

اسپیدکنترلر موتور براشلس با سنسور

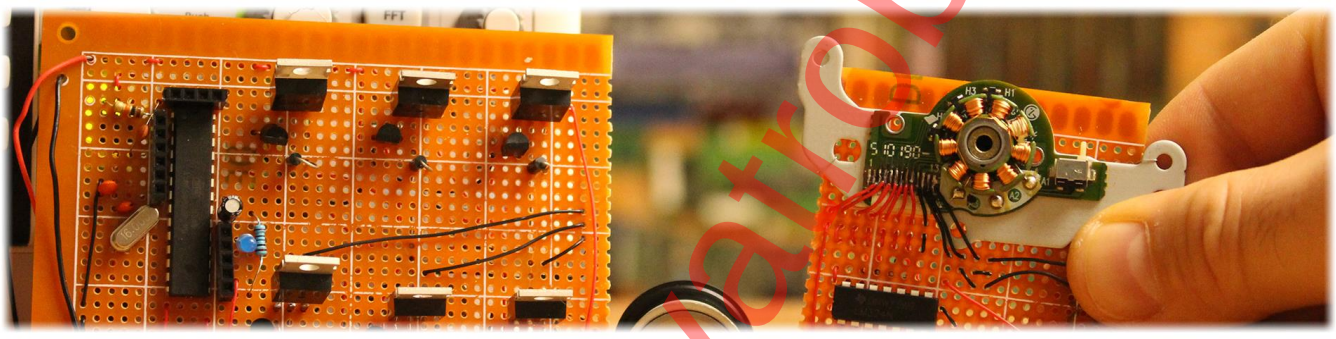
قطعات مورد نیاز

- آردوینو نانو
- میکروکنترلر ATMEGA328
- کریستال ۱۶ مگاهرتز
- ۲ عدد خازن ۲۲ پیکوفاراد
- رگولاتور ۵ ولت AMS1117
- ۳ عدد ترانزیستور ماسفت IRFZ44N
- ۳ عدد ترانزیستور ماسفت IRF4905
- ۶ عدد دیود IN5819
- ۷ عدد مقاومت ۱۰ کیلو اهم
- موتور براشلس سنسور دار (می توانید از روی CD رایتر باز کنید)
- ۶ عدد ترانزیستور S8050 (مشخصا می توانید از ترانزیستور مشابه هم استفاده کنید)
- آی سی آپ امپ LM324
- LED
- تعدادی سیم
- برد سوراخ دار

مقدمه

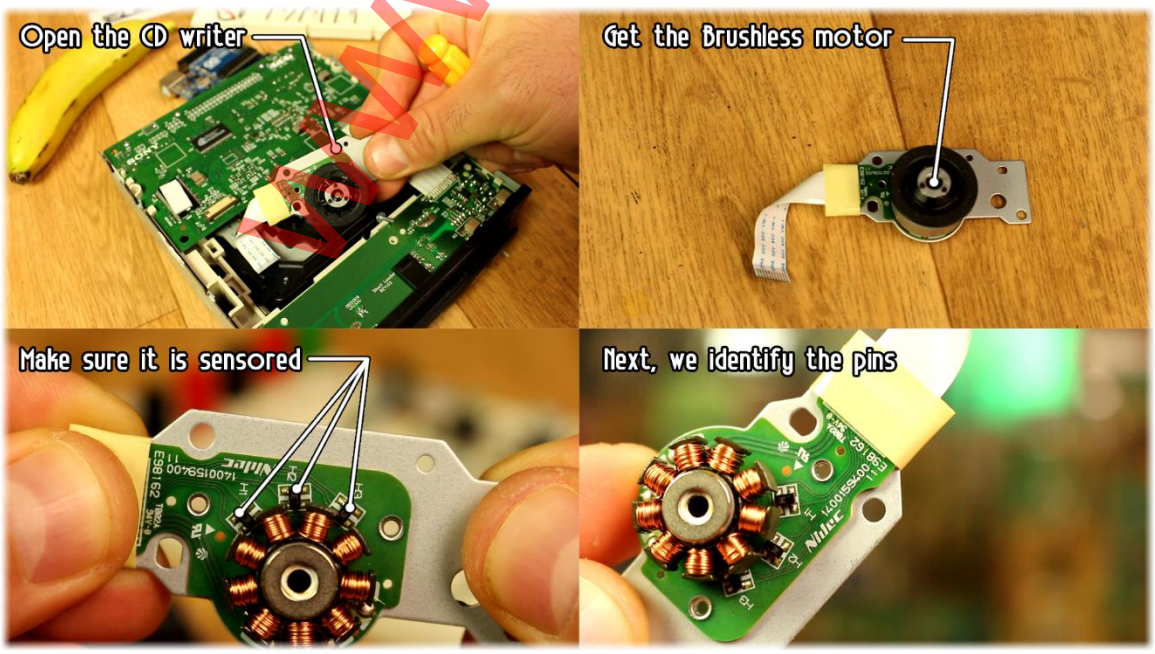
می خواهیم یک اسپیدکنترلر موتور براشلس با روش سنسور اثر هال (موتور های براشلس سنسور دار) درست کنیم . در اسپیدکنترلر هایی که از داده های سنسور استفاده نمی کنند باید نیروهای الکترومغناطیسی اندازه گیری شود و با توجه به آن به ترتیب سوئیچ زده شود . در مدل بدون سنسور مدار پیچیده تر است .

ما در این پروژه کد های برنامه را نسبت به اسپیدکنترلر بدون سنسور که در سایت قرار داده شده بهبود داده ایم . موتور های براشلسی که ما در این پروژه استفاده می کنیم دارای سنسور اثر هال است ، این سنسور ها برای به دست آوردن موقعیت چرخش است بنابراین مانند موتور های براشلس معمولی نیازی به اندازه گیری میدان مغناطیسی برگشتی از سیم پیچ ها که در سیم موتور جریان میابد نیست .



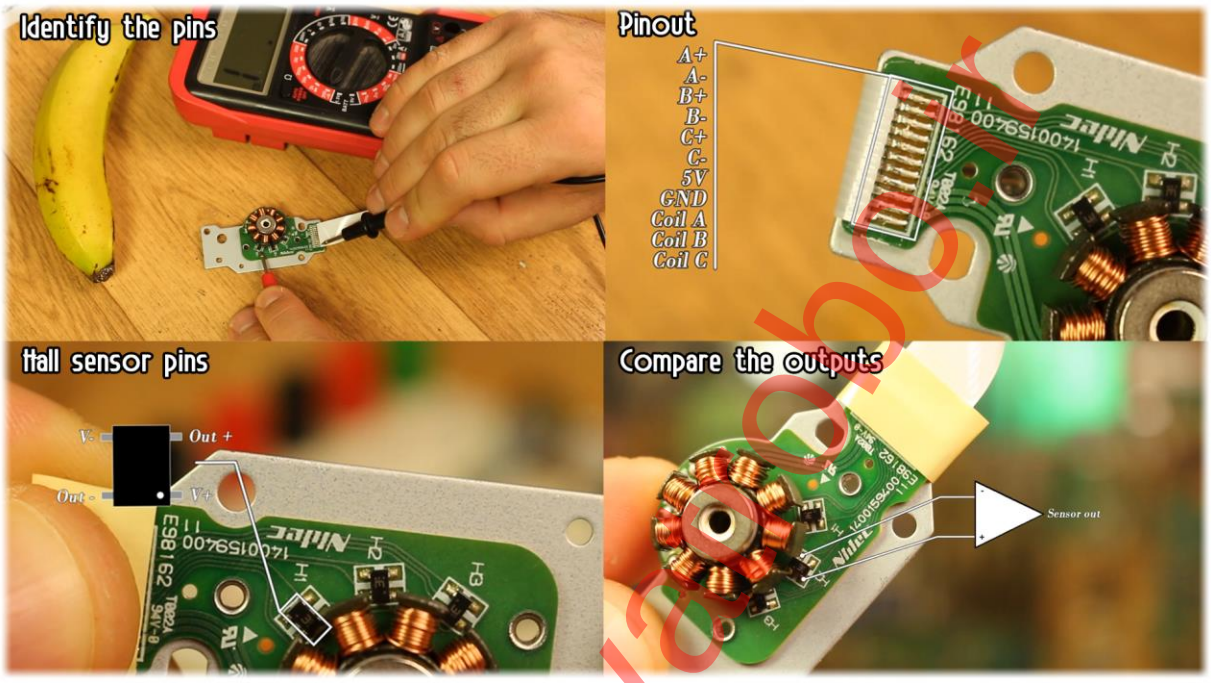
آماده سازی موتور

من یک موتور براشلس سنسور دار از دستگاه CD رایتر جدا جدا کردم . شما برای این کار مطمئن شوید که موتور موجود در CD رایتر شما دارای سه سنسور کوچک بر روی PCB موتور است مانند تصویر زیر . ۱۱ پین از برد موتور خارج می شود .

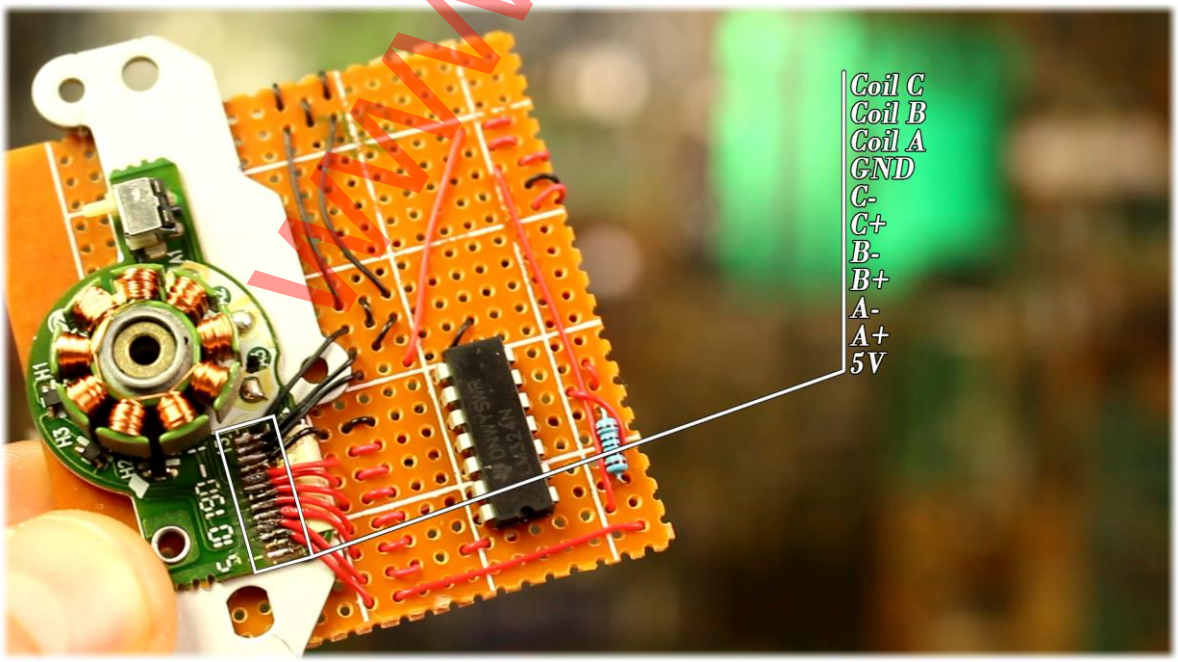


توسط تست دیود مولتی متر پین های خروجی قطعات را شناسایی می کنیم . موتور های CD رایتر های مختلف دارای ترتیب متفادتی هستند پس شما حتما برد موتور خود را توسط تست دیود بررسی کنید و پین های آن را مشخص کنید . پین های موتور ما طبق تصویر زیر می باشد . در تصویر زیر ، Coil A نشان گر سیمی است که به سیم پیچ اول موتور وصل است

و باید توسط ترانزیستور درایو شود ؛ کویل B و C هم به همین ترتیب . +A و -A خروجی های سنسور کویل اول هستند که باید به آپ امپ بروند ؛ B و C نیز به همین ترتیب هستند . پین GND به زمین مدار و 5V هم به ۵ ولت مدار متصل می شوند .

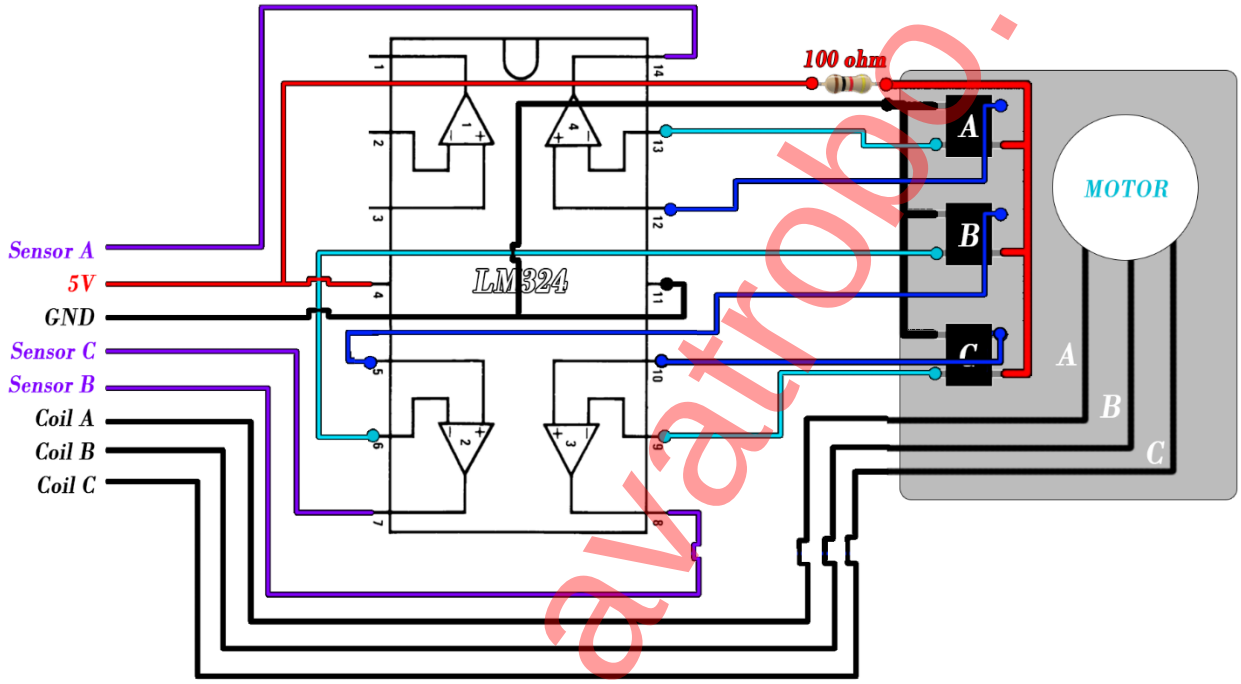


در پین های خروجی سنسور های اثر هال باید +A به ورودی مثبت آپ امپ وصل شود و -A به ورودی منفی آپ امپ وصل شود . اگر ولتاژ + از – بیشتر بود خروجی یک منطقی (۵ ولت) می دهد و اگر ولتاژ + از – کمتر شود خروجی صفر منطقی (۰ ولت) می دهد .

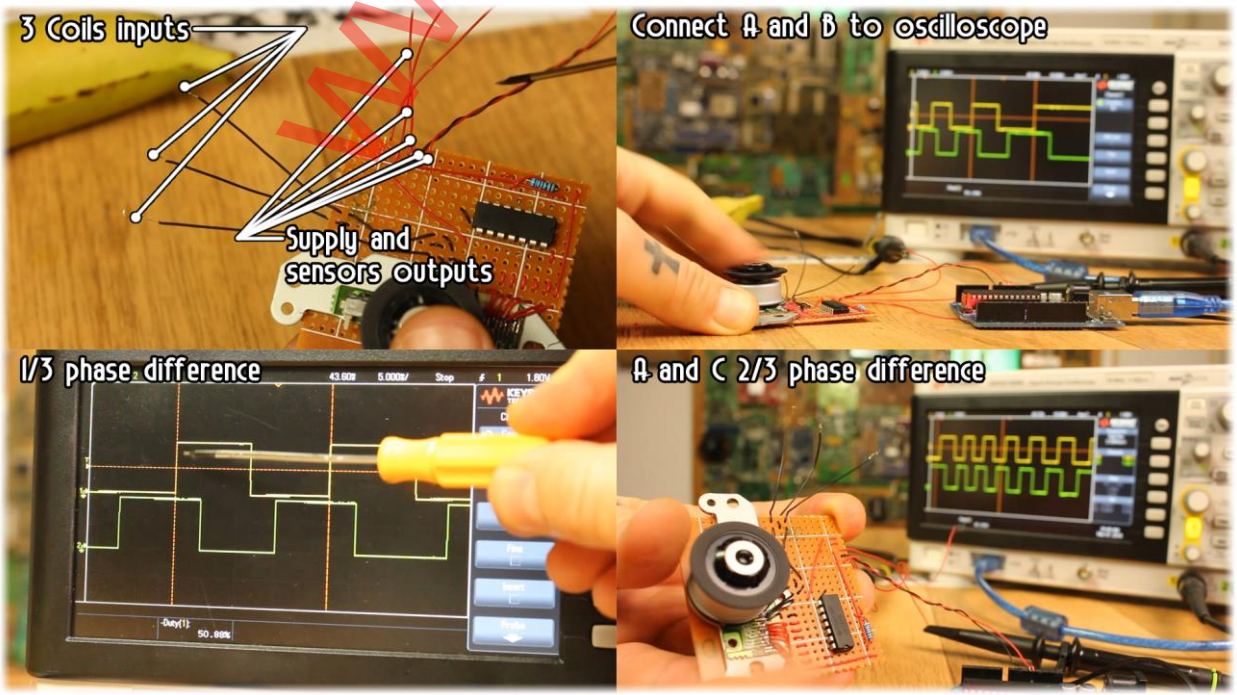


شماتیک سنسور های اثر هال

آی سی آپ امپی که ما در این پروژه استفاده کردیم آی سی LM324 می باشد . LM324 دارای ۴ عدد آپ امپ می باشد که ما به ۳ تای آن نیاز داریم . ولتاژ ۵ ولت سنسور های اثر هال توسط یک مقاومت ۱۰۰ اهم (محدود کننده جریان) تامین می شود . پین GND و 5V هم مستقیما به زمین و ۵ ولت LM324 متصل می شود . در تصویر زیر پین های Sensor A و B و C خروجی های آپ امپ هستند که صفر و یک می شوند و باید به آردوینو متصل شوند .



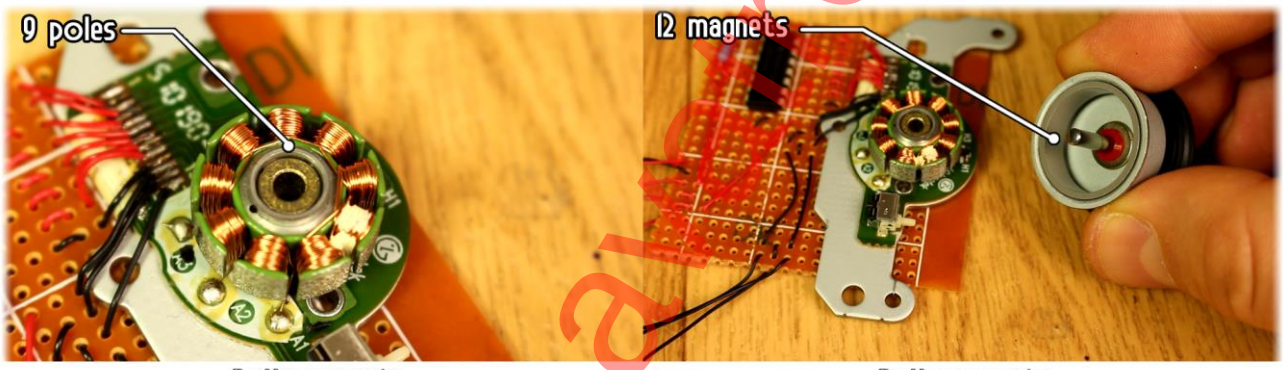
حال پین های خروجی آپ امپ A و B را به اسیلوسکوپ وصل کنید و موتور را بچرخانید ، دو پالس مربعی A و B دارای اختلاف فاز یک سوم خواهند بود . حال پین های خروجی آپ امپ A و C را به اسیلوسکوپ وصل کنید و موتور را بچرخانید ، دو پالس مربعی A و C دارای اختلاف فاز دو سوم خواهند بود .

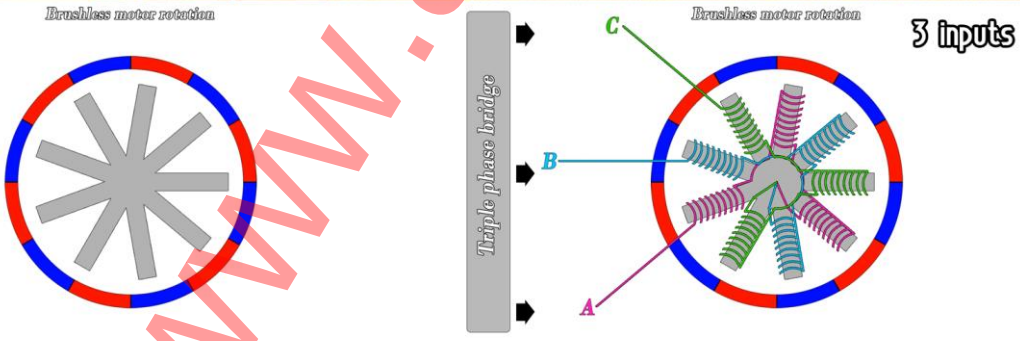


کنترل ۳ فاز موتور براشلس

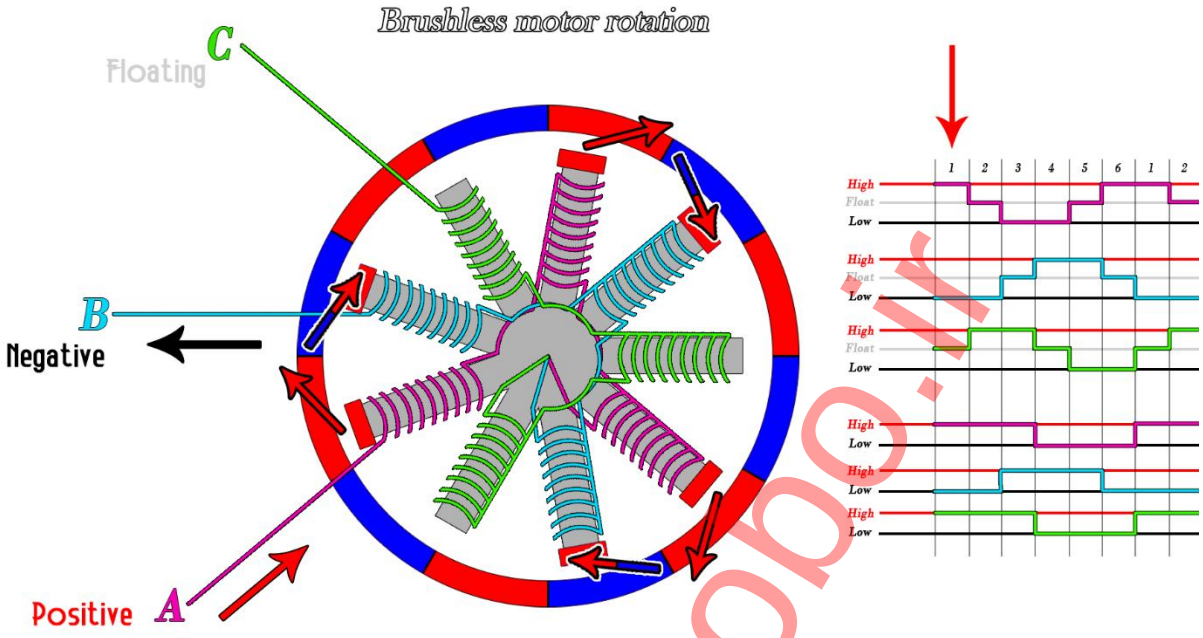
برای فهم بهتر ساختمان درونی موتور های براشلس سه فاز به شکل بالا نگاه می کنیم . ما می دانیم که این موتور دارای ۳ عدد کویل چند تایی است . این کویل ها به صورت مساوی در اطراف موتور توزیع شده اند . در واقع موتور برای هر فاز (ورودی) دارای ۴ کویل است یعنی در کل موتور ما دارای ۱۲ کویل می باشد . در شکل بالا برای هر ورودی فقط یک کویل را نشان داده است (برای فهم راحت تر) . بخش حرکتی موتور می تواند شفت داخلی با خارجی باشد که در موتور ما شفت خارجی است . قسمت متحرک (بخش بیرونی موتور که به شفت وصل شده) دارای ۱۲ آهنربای قوی است که با فاصله های مساوی از هم در دور این قسمت قرار داده شده است . در مجاورت این آهنرباهای ثابت ، آهنربا های متغیری (سیم پیچ) وجود دارد که همان بخش داخلی موتور است و پالس های اعمالی از مدار به این سیم پیچ ها (کویل ها) اعمال می شود .

موتور ما ۹ سیم پیچ (استاتور) دارد یعنی هر فاز ۳ پیچ دارد . موتور ما داری ۱۲ آهنربا بر روی روتور است و ۹ سیم پیچ هم بر روی استاتور قرار دارد .



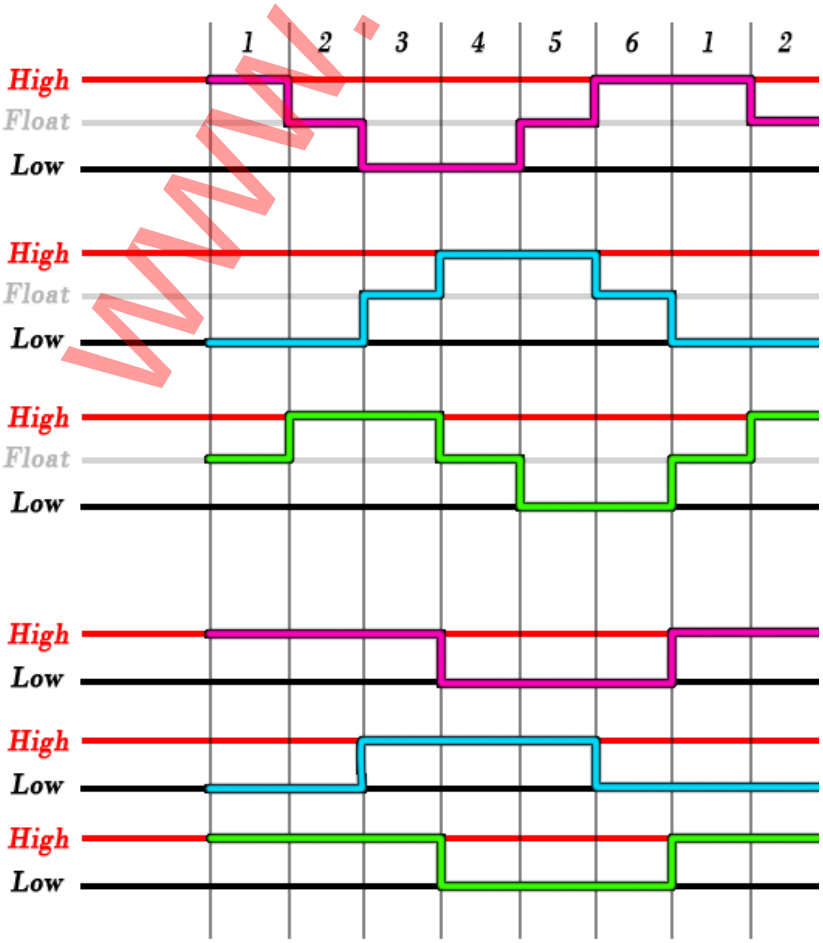


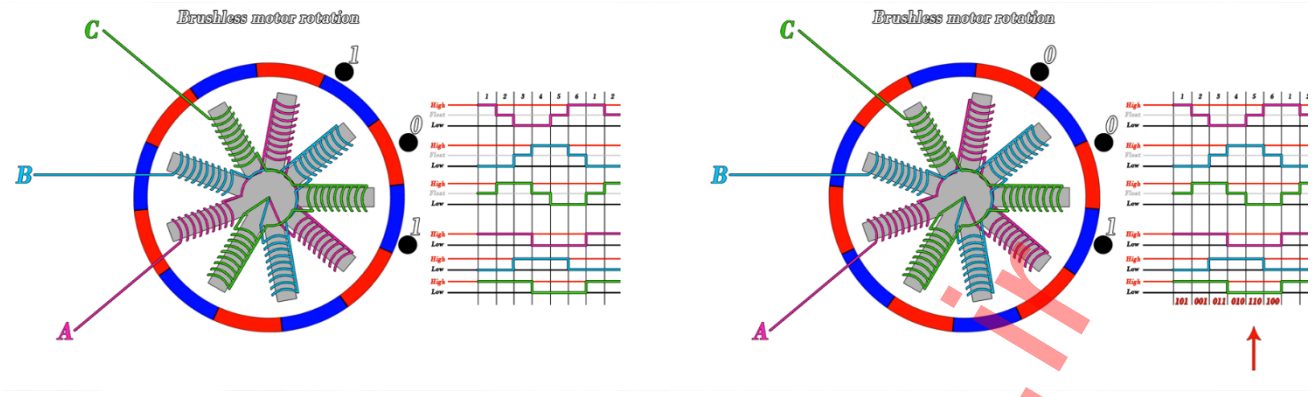
اگر ما به دو سیم A و B موتور براشلس ولتاژ مثبت و منفی وصل کنیم و سیم C را به جایی وصل نکنیم به خاطر جاذبه و دافعه بین سیم پیچ ها و آهنربا ها ، موتور حرکت کوچکی خواهد داشت و بعد می ایستد اگر بخواهیم چرخش ادامه پیدا کند باید طبق ترتیب خاصی ولتاژ مثبت و منفی بین پایه های موتور تعویض شود مثلا در حرکت بعدی ، ولتاژ مثبت به جای A به C وصل شود . به تصویر زیر دقت کنید .



ترتیب کلید زنی در موتور براشلس

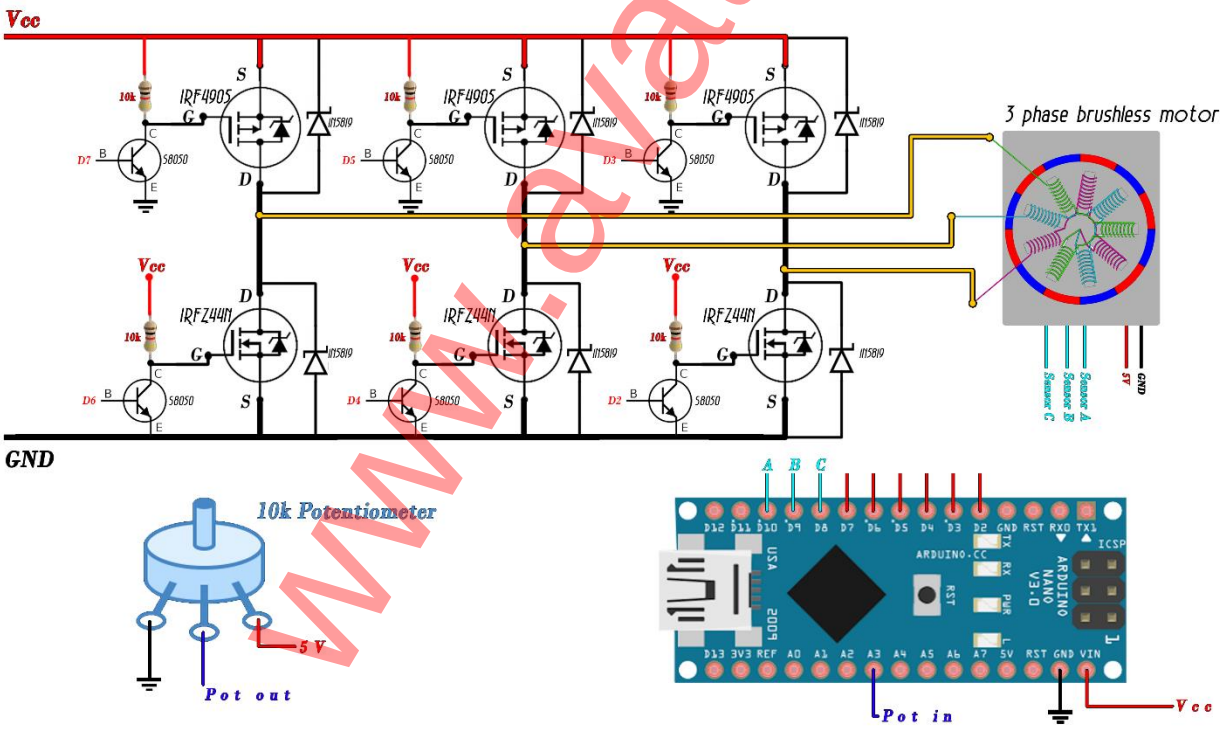
ترتیب کلید زنی به صورت شکل زیر خواهد بود . بنفش برای سیم A ، آبی برای سیم B و سبز برای سیم C می باشد . اگر به شکل زیر توجه کنید می بینید که در بازه زمانی ۱ سیم A مثبت ، سیم B منفی و C آزاد است . در بازه زمانی ۲ سیم A آزاد ، سیم B منفی و سیم C مثبت است . ۶ بازه زمانی داریم و دوباره از سر گرفته می شود . نمودار پایینی هم خروجی صفر و یک سنسور های سه فاز A و B و C را نشان می دهد . با توجه به همین دو نمودار کد های برنامه را می نویسیم .





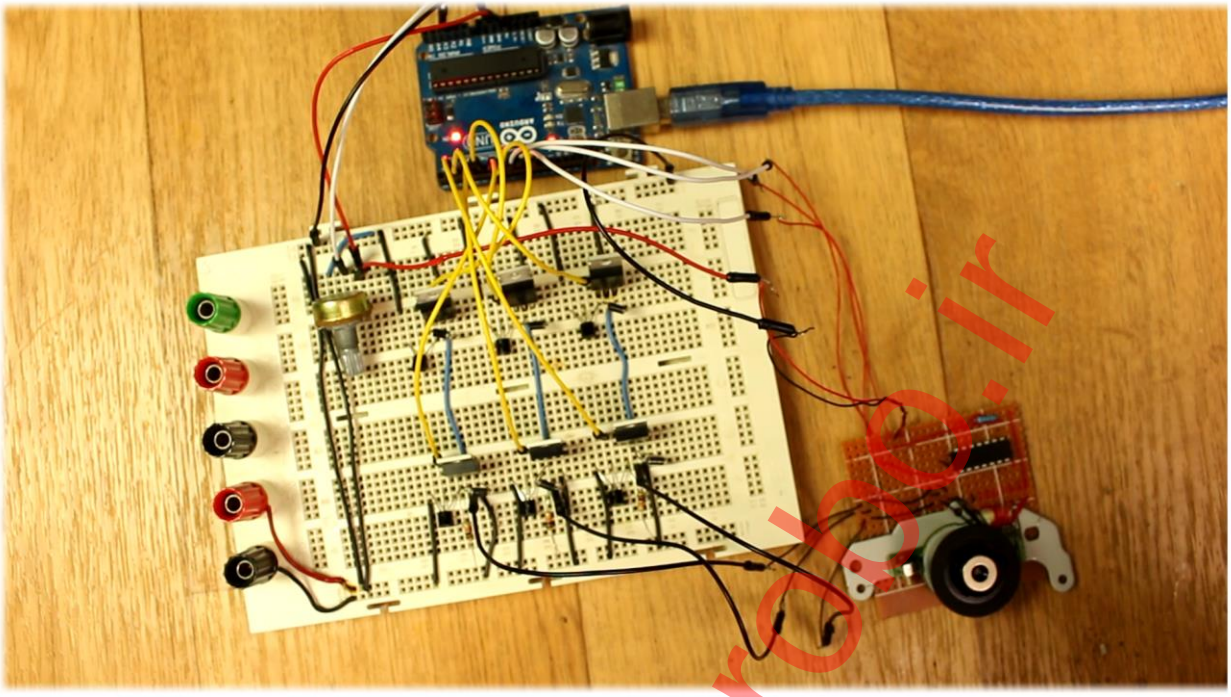
پل سه فاز

تصویر زیر شماتیک پل سه فاز می باشد . در بالای مدار ترانزیستور های ماسفت از نوع P داریم و در پایین مدار ترانزیستور های ماسفت نوع N داریم . ما در این اسپیدکنترلر از درایور ماسفت استفاده نمی کنیم به جای درایور ماسفت از ترانزیستور BJT استفاده می کنیم و گیت هر ماسفت را پول آپ می کنیم یعنی با مقاومت به ۵ ولت وصل می کنیم .



موقعی که ماسفتی که به سیم پیچ A وصل است از ردیف بالا و مافتی که به سیم پیچ B وصل است از ردیف پایین فعال باشند جریان از سیم پیچ A به سیم پیچ B می رود . در هر بازه زمانی همیشه به دو ماسفت دستور خواهیم داد .

در تصویر زیر مداری را که بسته ام را می بینید . خروجی های سنسور های A و B و C به پین های ۸ و ۹ و ۱۰ برد آردوینو متصل شده است . پین های ۷ و ۶ و ۵ و ۴ و ۳ و ۲ به ۶ عدد ماسفت متصل شده است . یک پتانسیومتر به پین آنالوگ A3 وصل کرده ام . تغذیه برد را با ولتاژ ۱۲ ولت تامین کرده ام .



کد های برنامه

در خطوط کد زیر در ابتدا پین هایی که سنسور ها به آن ها وصل می شوند را به عنوان ورودی تعریف کرده ام . سپس توسط رجیستر هایی که در برنامه می بینید پایه های ۸ و ۹ و ۱۰ را برای اینتراپت فعال کرده ام .

```
void setup() {
  //Define sensor pins and potentiometer as inputs
  pinMode(pot,INPUT);
  pinMode(SensorA,INPUT);
  pinMode(SensorB,INPUT);
  pinMode(SensorC,INPUT);
  pinMode(pwm_pin,INPUT);

  PCICR |= (1 << PCIE0);    //enable PCMSK0 scan
  PCMSK0 |= (1 << PCINT0);  //Set pin D8 trigger an interrupt on state change.   sensor
A
  PCMSK0 |= (1 << PCINT1);  //Set pin D9 trigger an interrupt on state change.   sensor
B
  PCMSK0 |= (1 << PCINT2);  //Set pin D10 trigger an interrupt on state change.  sensor
C
}
```

با توجه به حالت سنسور ها ما به موتور ها پالس می دهیم . در حلقه زیر همیشه در هر گام ۲ عدد ماسفت کلید زنی می شوند که این کلید زنی طبق ترتیب خاصی است . سرعت چرخش موتور از طریق ایجاد تاخیر بین دو گام ایجاد می شود . هر چه تاخیر بیشتر شود سرعت کمتر می شود . ما سرعت را از طریق یک پتانسیومتر که به پایه آنالوگ A3 متصل است تنظیم می کنیم . عدد خوانده شده از پایه A3 که عددی از ۰ تا ۱۰۲۳ است به ۱ تا ۴۰۰۰ تبدیل می شود (توسط دستوری مانند map) و این عدد بر حسب میکروثانیه تاخیر بین دو گام را مشخص می کند .

```
ISR(PCINT0_vect){
  //This part will run any time any of the sensor pins will change its value.
  if(  (PINB & B00000101) && fase == 6  ){
    fase = 1;
  }

  if(  (PINB & B00000100) && fase == 1 ){
    fase = 2;
  }

  if(  (PINB & B00000110) && fase == 2 ){
    fase = 3;
  }

  if(  (PINB & B00000010) && fase == 3 ){
    fase = 4;
  }

  if(  (PINB & B00000011) && fase == 4 ){
    fase = 5;
  }

  if(  (PINB & B00000001) && fase == 5 ){
    fase = 6;
  }
}
```

کد های کامل این مدار برای کنترل سرعت توسط پتانسیومتر در فایل های دانلودی قرار داده شده است . با نام ESC_sensored_potentiometer

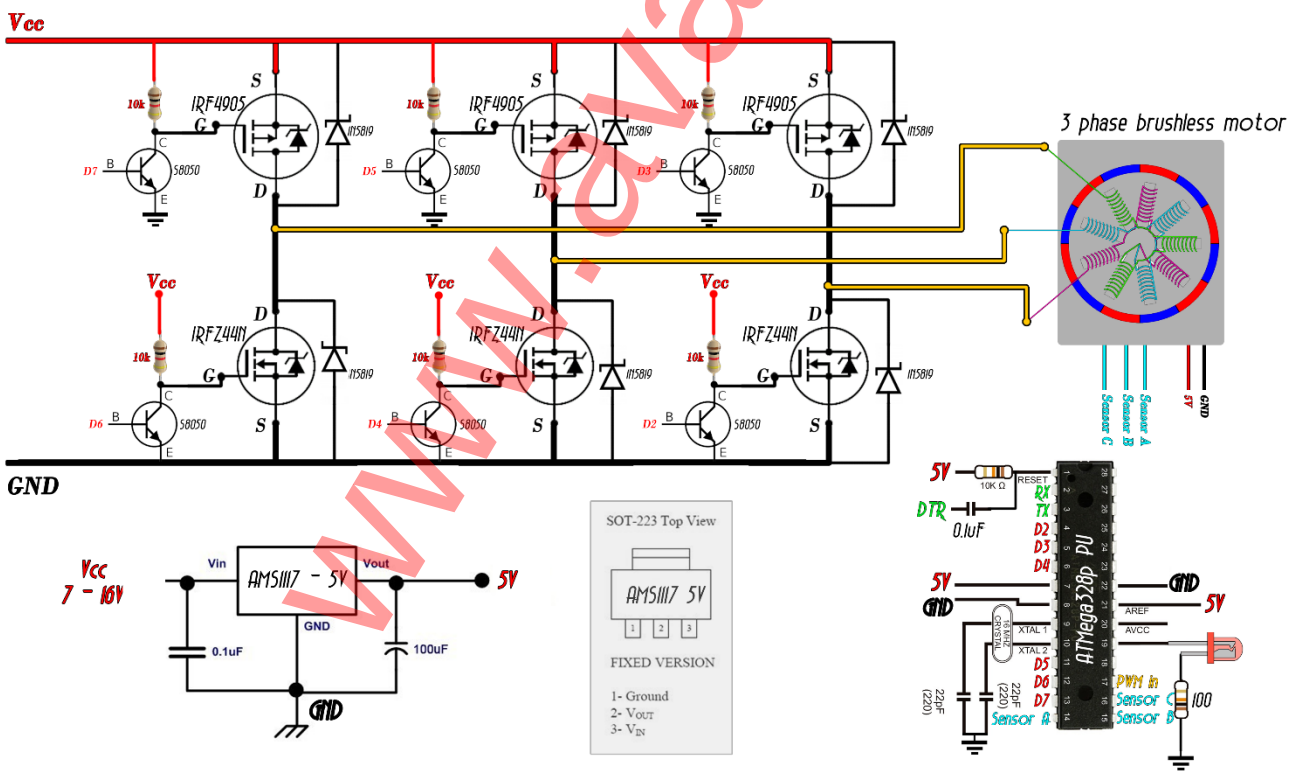
در اسپیدکنترلر های معمول داخل بازار سرعت موتور ها توسط پتانسیومتر کم و زیاد نمی شوند بلکه توسط پالس PWM که از فلایت کنترلر یا گیرنده دریافت می کند کم و زیاد می شود . پهنای پالس این پالس PWM بین ۱۰۰۰ تا ۲۰۰۰ میکروثانیه است . ما برای دریافت پالس PWM پین ۱۱ آردوینو را به عنوان ورودی تعریف کرده ایم . توسط وقفه ها زمان بین دو پالس را اندازه می گیریم و آن را به بازه ۱۰ تا ۳۰۰۰ میکروثانیه (برای تاخیر بین گام ها) تبدیل می کنیم . موتور با مقدار بیشتر از ۱۰۷۰ شروع به چرخش می کند .

```
ISR(PCINT0_vect){
  current_count = micros();
  if(PINB & B00001000 ){
    if(lastPWM_state == 0){
      lastPWM_state = 1;
      counter = current_count;
    }
  }
  else if(lastPWM_state == 1){
    lastPWM_state = 0;
    PWM_width = current_count - counter;
    }

  //This part will run any time any of the sensor pins will change its value.
  if(  (PINB & B00000101) && fase == 6  ){
    fase = 1;
  }

  if(  (PINB & B00000100) && fase == 1 ){
    fase = 2;
```

```
  }

  if(  (PINB & B00000110) && fase == 2 ){
    fase = 3;
  }

  if(  (PINB & B00000010) && fase == 3 ){
    fase = 4;
  }

  if(  (PINB & B00000011) && fase == 4 ){
    fase = 5;
  }

  if(  (PINB & B00000001) && fase == 5 ){
    fase = 6;
  }
}
```

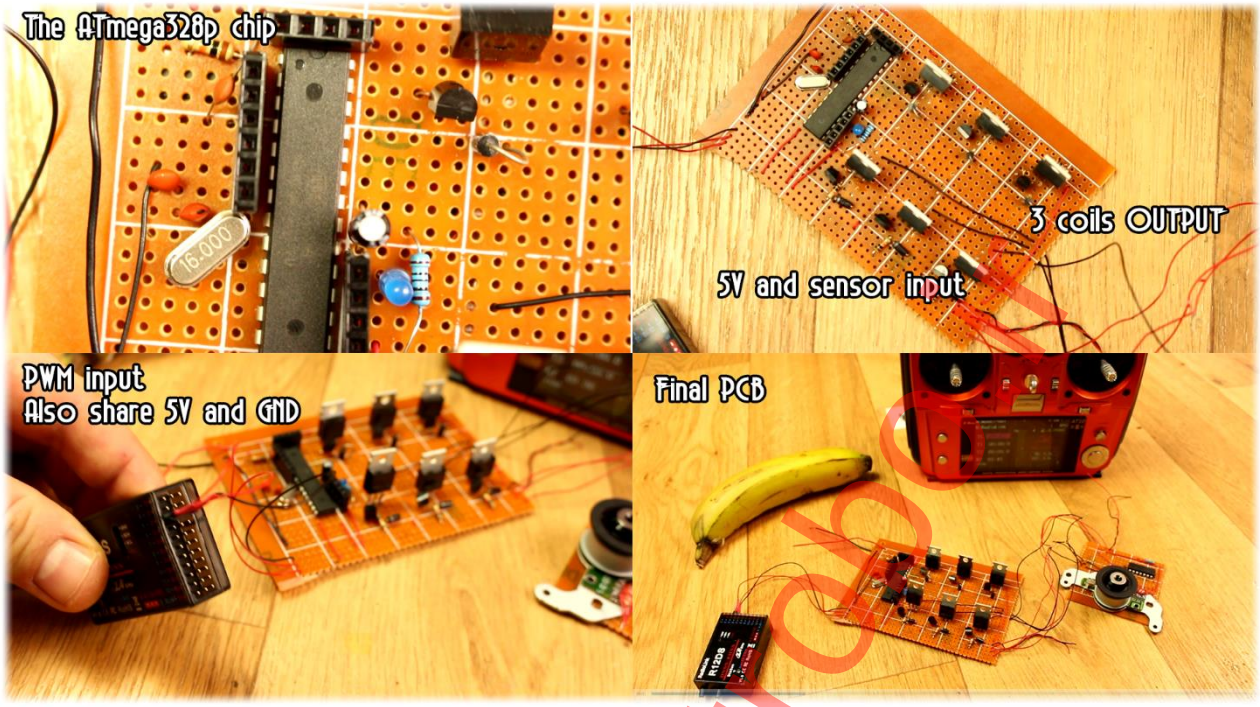
شماتیک مدار

کریستال متصل شده به ATMEGA328 کریستال ۱۶ مگاهرتز است .



برای پروگرام کردن میکروکنترلر ATMEGA328 می توان میکروکنترلر را برداشت و بر روی برد آردوینو گذاشت و پروگرام کرد و همچنین می توان توسط ماژول مبدل USB به سریال (مدلی که پین DTR داشته باشد) استفاده کرد و توسط پین های RX ، TX و DTR میکروکنترلر را پروگرام کرد .

من برای تست پروژه از یک رادیوکنترل استفاده کرده ام و رسیور را به اسپیدکنترلر وصل کرده ام .



در نهایت به این نکته هم توجه داشته باشید که می توانید به جای ترانزیستور های MOSFET از IGBT استفاده کرد .