

Тұрақты ток қозғалқыштары

Бұл тарауда біз қозғалтқыштардың әртүрлі түрлерін Arduino (UNO) тақтасымен байланыстырамыз және қозғалтқышты қалай қосуға және оны тақтадан басқаруға болатындығын көрсетеміз.

Негізі робототехникада көбіне тек тұрақты ток қозғалтқыштарын қолданады, өйткені, біріншіден роботтар далада немесе сымдық элетр қуаты жоқ жерде де жұмыс жасай білу керек, сондықтан осындай кезде аккумулятордан қорек көзін алады, ал барлық аккумулятор тек қана тұрақты ток қана бере алады. Екіншіден, тұрақты ток қозғалтқыштары салыстырмалы түрде көбірек күш береді немесе момент, осындай артықшылығының арқысында тұрақты ток қозғалтқыштарын қолдану өте тиімді.

Қозғалтқыштардың робототехникада қолданылатын негізгі 3 түрі –

- Тұрақты ток қозғалтқышы (щеткалы және щеткасыз)
- Серво қозғалтқышы
- Қадамдық қозғалтқыш

Щеткалы тұрақты ток қозғалтқышы екі полюсті электромагнит ретінде әрекет ететін орам сымдарының орамдарының конфигурациясын, якорьді пайдаланады. Токтың бағытын коммутатор, механикалық айналмалы қосқыш циклге екі рет өзгертеді. Бұл токтың якорь арқылы өтуін жеңілдетеді; осылайша электромагниттің полюстері қозғалтқыштың сыртындағы тұрақты магниттерді тартып, итереді. Содан кейін коммутатор якорь электромагнитінің полярлығын өзгертеді, өйткені оның полюстері тұрақты магниттердің полюстерін кесіп өтеді.

Щеткалы моторының артықшылықтары:

- Құрылыстың жалпы құнының төмендігі;
- Қызмет мерзімін ұзарту үшін оны жиі қайта құруға болады;
- Қымбат емес және тікелей алға контроллер;
- Бекітілген жылдамдық үшін контроллер қажет емес;
- Төтенше жұмыс орталары үшін өте қолайлы.

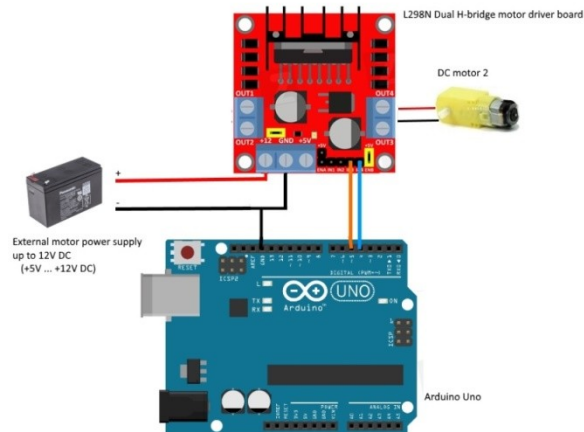
Щеткасыз тұрақты ток қозғалтқыш, керісінше, сыртқы роторы ретінде тұрақты магнитті пайдаланады. Сонымен қатар, ол қозғалтқыш катушкаларының үш фазасын және ротордың орнын бақылайтын арнайы сенсорды пайдаланады. Сенсор ротордың орнын бақылай отырып, контроллерге анықтамалық сигналдар жібереді. Өз кезегінде контроллер катушкаларды құрылымдық түрде іске қосады - бір фазадан кейін екіншісі.

Щеткасыз тұрақты ток қозғалтқыш:

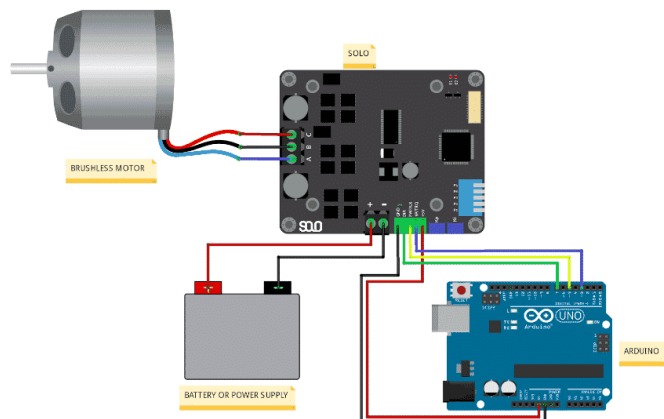
- Шөткелердің болмауына байланысты жалпы техникалық қызмет көрсетудің аз болуы;
- Номиналды жүктеме кезінде барлық жылдамдықтарда тиімді жұмыс істейді;
- Өлшемге қатынасы жоғары тиімділік және жоғары шығыс қуаты;
- Кіші өлшемде әлдеқайда жоғары жылу сипаттамалары бар

- Жоғары жылдамдық диапазонында төмен элетрлік шуу болады

Тұрақты қозғалтқыш (тұрақты қозғалтқыш) - қозғалтқыштың ең көп таралған түрі. Тұрақты ток қозғалтқыштарында әдетте тек екі Шығыс бар, біреуі оң және біреуі теріс болады. Егер сіз осы екі сымды тікелей батареяға қоссаңыз, қозғалтқыш айналады. Егер сіз сымдарды ауыстырсаңыз, қозғалтқыш қарама-қарсы бағытта айналады.



Сурет-6.1.1 . Щеткалы тұрақты ток қозғалтқышы



Сурет-6.1.2. Щеткасыз тұрақты ток қозғалтқышы



Қозғалтқышты Arduino тақтасының контактілерінен тікелей шығармаңыз. Бұл тақтаға зақым келтіруі мүмкін. Драйвер схемасын немесе чипті қолданыңыз.

Сізге келесі компоненттер қажет –

- 1x Arduino Uno тақтасы
- 1x Транзистор Pn2222
- 1x шағын DC 6V қозғалтқышы
- 1n4001 1x диод
- 1x Резистор 270 Ом



Бағдарламасы/Код

Келесі кодты көшіріп, оны Arduino тақтасына жүктеңіз. Код жақсы түсіндірілген, сондықтан сіз оның қалай жұмыс істейтінін оңай түсініп, оны өз жобаларыңызға қосу үшін өзгерте аласыз.

Щеткалы тұрақты тоқ қозғалтқышы	Щеткасыз тұрақты тоқ қозғалтқышы
<pre> int motorPin = 9; void setup() { pinMode(motorPin, OUTPUT); Serial.begin(9600); while (! Serial); Serial.println("Speed 0 to 255"); } void loop() { if (Serial.available()) { int speed = Serial.parseInt(); if (speed >= 0 && speed <= 255) { analogWrite(motorPin, speed); } } } </pre>	<pre> const int pwm = 2 ; //initializing pin 2 as pwm const int in_1 = 8 ; const int in_2 = 9 ; //For providing logic to L298 IC to choose the direction of the DC motor void setup() { pinMode(pwm,OUTPUT) ; //we have to set PWM pin as output pinMode(in_1,OUTPUT) ; //Logic pins are also set as output pinMode(in_2,OUTPUT) ; } void loop() { //For Clock wise motion , in_1 = High , in_2 = Low digitalWrite(in_1,HIGH) ; digitalWrite(in_2,LOW) ; analogWrite(pwm,255) ; /* setting pwm of the motor to 255 we can change the speed of rotation by changing pwm input but we are only using arduino so we are using highest value to driver the motor */ //Clockwise for 3 secs delay(3000) ; //For brake digitalWrite(in_1,HIGH) ; digitalWrite(in_2,HIGH) ; delay(1000) ; //For Anti Clock-wise motion - IN_1 = LOW , IN_2 = HIGH digitalWrite(in_1,LOW) ; digitalWrite(in_2,HIGH) ; delay(3000) ; //For brake digitalWrite(in_1,HIGH) ; digitalWrite(in_2,HIGH) ; delay(1000) ; } </pre>

БАҒДАРЛАМАҒА ШОЛУ

motorPin

Транзистор қозғалтқыштың қуатын

	басқаратын қосқыш ретінде әрекет етеді. Arduino pin 3 транзисторды қосу және өшіру үшін қолданылады және эскизде "motorPin" деп аталады.
<code>Serial.println</code>	Бағдарлама іске қосылған кезде, ол сізге қозғалтқыштың жылдамдығын басқару үшін мәндер беруді ұсынады. Сериялық мониторға 0-ден 255-ке дейінгі мәнді енгізу керек.
<code>int speed = Serial.parseInt();</code>	"loop" функциясында "Serial" командасы.parseInt " сериялық мониторда мәтін түрінде енгізілген санды оқып, оны "int" етіп түрлендіру үшін қолданылады. Мұнда кез-келген санды енгізуге болады. Келесі жолдағы 'if' операторы, егер сан 0 мен 255 арасында болса, сол санмен аналогтық жазбаны орындайды.

Сіз не көруіңіз керек:

Тұрақты ток қозғалтқышы сериялық порт арқылы алынған мәнге (0-ден 250-ге дейін) сәйкес әртүрлі жылдамдықта айналады.

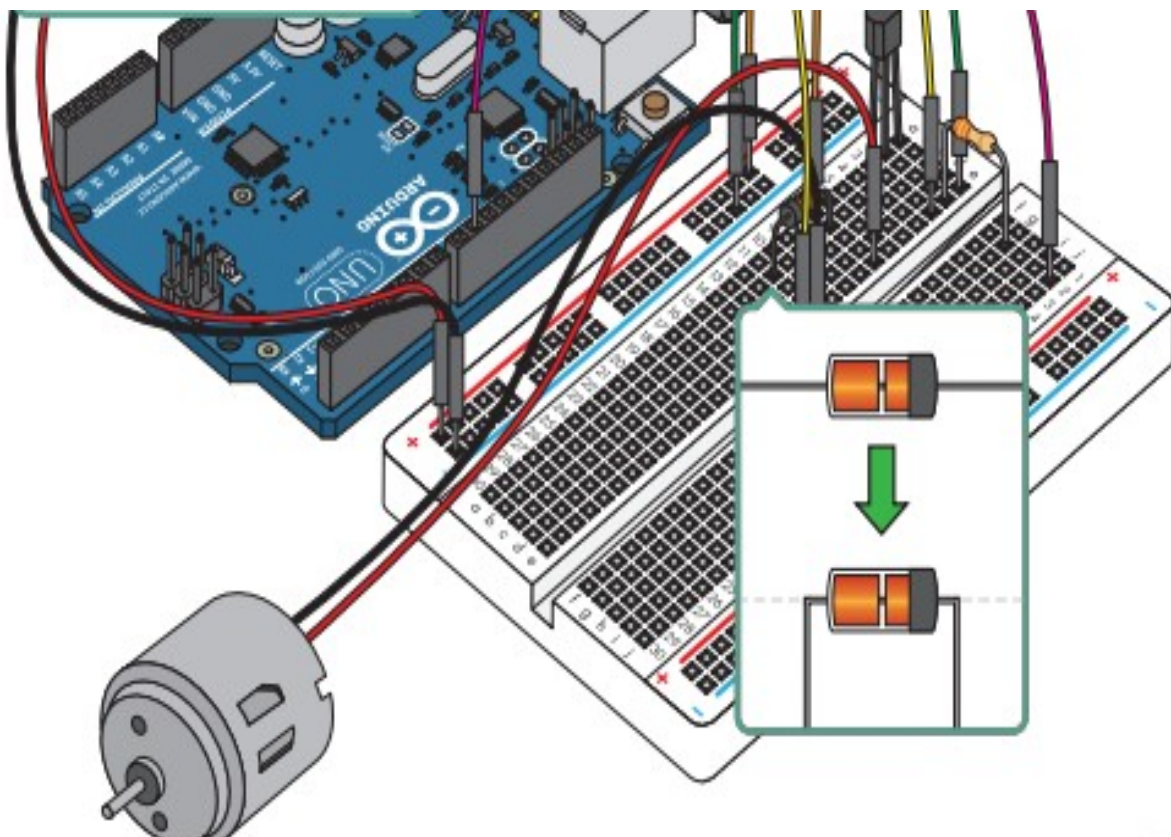


Алдымен транзистордың дұрыс қосылғанына көз жеткізіңіз. Транзистордың жалпақ жағы суретте көрсетілгендей Arduino тақтасына қарауы керек. Екіншіден, диодтың жолақты ұшы суретте көрсетілген орынға сәйкес +5 В қуат сызығына бағытталуы керек.

"Серво", сервоқозғалтқыштың қысқартылуы, бұл электронды кері байланыс тізбегін қамтитын қозғалтқыш. Сервоқозғалтқыштар электроника мен автоматикада қолдануға өте ыңғайлы, өйткені олар

Тұрақты ток қозғалтқышын басқару

Сервоқозғалтқыштардан кейін біз электр қозғалтқышымен танысамыз. Бұл Arduino-ға қарағанда көбірек токты ауыстыра алатын транзисторды қажет етеді. Транзисторды пайдаланғанда, оның параметрлері таңдалған жабдыққа сәйкес келетініне көз жеткізу керек. Біз қолданатын транзистор максималды кернеу 40 Вольт пен 200 миллиамперге арналған-бұл біздің қозғалтқышымыз үшін өте қолайлы болып табылады.





**Схеманы жинаған кезде мұқият және абай болыңыз,
транзисторды температура датчигімен шатастырып алмаңыз,
олар өте ұқсас келеді.**

Біз қозғалтқыштың жылдамдығын басқару үшін ендік импульстік модуляцияны (PWM) қолданамыз. Arduino порттары бірнеше жарық диодтарының жұмысына төтеп бере алады (40 мА дейін), бірақ олар қозғалтқыштан және басқа да қуатты тұтынушылардан жүктемені жеңе алмайды. Қозғалтқыш шамамен 50-ден 100 мА-ға дейін тұтынады, бұл Arduino-ға қарағанда әлдеқайда көп. Сондықтан біз негізгі жұмысты орындайтын транзисторды қолданамыз. Транзистор "қатты күйдегі ажыратқыш" рөлін атқарады, біз оған аз мөлшерде ток береміз және әлдеқайда үлкен ток ала аламыз. 2n2222 Транзисторы токты 200 мА дейін ауыстыра алады.



Бағдарламасы/Код

Келесі кодты көшіріп, оны Arduino тақтасына жүктеңіз. Код жақсы түсіндірілген, сондықтан сіз оның қалай жұмыс істейтінін оңай түсініп, оны өз жобаларыңызға қосу үшін өзгерте аласыз.

```
const int motorPin = 9;
void setup()
{
  pinMode(motorPin, OUTPUT);
  Serial.begin(9600);
}
void loop()
{
  serialSpeed();
}
void motorOnThenOff()
{
  int onTime = 3000;
  int offTime = 3000;
  digitalWrite(motorPin, HIGH);
  delay(onTime);
  digitalWrite(motorPin, LOW);
```

```

    delay(offTime);
}
void motorOnThenOffWithSpeed()
{
    int Speed1 = 200;
    int Time1 = 3000;
    int Speed2 = 50;
    int Time2 = 3000;
    analogWrite(motorPin, Speed1);
    delay(Time1);
    analogWrite(motorPin, Speed2);
    delay(Time2);
}
void motorAcceleration()
{
    int speed;
    int delayTime = 20;
    for(speed = 0; speed <= 255; speed++)
    {
        analogWrite(motorPin,speed);
        delay(delayTime);
    }
    for(speed = 255; speed >= 0; speed--)
    {
        analogWrite(motorPin,speed);
        delay(delayTime);
    }
}
void serialSpeed()
{
    int speed;
    Serial.println("Type a speed (0-255) into the box above,");
    Serial.println("then click [send] or press [return]");
    Serial.println();
    while(true)
    {
        while (Serial.available() > 0)
        {
            speed = Serial.parseInt();
            speed = constrain(speed, 0, 255);
            Serial.print("Setting speed to ");
            Serial.println(speed);
            analogWrite(motorPin, speed);
        }
    }
}

```

БАҒДАРЛАМАҒА ШОЛУ

while	Arduino тізбектей портын деректерді алу үшін де, беру үшін де пайдалануға болады. Деректер кез-келген уақытта келуі
-------	---

<code>(Serial.available() > 0)</code>	мүмкін болғандықтан, олар алдымен "сақтау" немесе Arduino алмасу буферіне келіп, оларды пайдалануға дайын болғанша сақталады. Serial Командасы.available () порт алған таңбалар санын қайтарады, бірақ сіз оларды әлі пайдаланған жоқсыз. Нөл дегеніміз, деректер жоқ.
<code>speed = Serial.parseInt();</code>	Егер деректер портқа келіп түссе және олар буферде болса, оны сол жерден шығарудың бірнеше жолы бар. Осындай әдістердің бірі - Serial командасын пайдалану.parseInt (), ол сериялық порт буферінде келесі бүтін санды іздейді және егер сіз портқа "1" "0" "0" енгізсеңіз, бұл функция 100 санын қайтарады.

Сіз не көруіңіз керек?

Егер тізбектің барлық компоненттері дұрыс жиналса, тұрақты ток қозғалтқышы айналуы керек. Егер бұлай болмаса, схеманы дұрыс жинағаныңызға көз жеткізіп, кодтың Arduino-ға жүктелгенін тексеріңіз немесе төмендегі ақауларды жою кеңестерін қараңыз.

Мүмкін болатын қиындықтар:	Мұндай схемаларды өмірде қалай қолдануға болады?
Мотор айнамайды Егер сіз 2n2222-ден басқа транзисторды қолданған болсаңыз, бірақ параметрлерге сәйкес келсе, оның аяқтарының сәйкестігін тексеріңіз.	Қашықтан басқарылатын автомобильдер дөңгелектерін бұру үшін және алға, артқа жылжыту үшін тұрақты ток қозғалтқыштарын пайдаланады.
Әлі де сәттілік жоқ Қозғалтқыш 5 Вольттан жұмыс істеп тұрғанын және оған қуат көзі жеткілікті екенін тексеріңіз.	
Әлі жұмыс істемейді Кейде Arduino компьютерден ажыратылуы мүмкін. Одан USB қосқышын ажыратып көріңіз, содан кейін оны қайта қосыңыз.	