

# ARDUINO TARAQQIYOTI

Ma'ruza #5

## **Mavzu: Arduino yordamida motorlarni boshqarish texnologiyalari**

Reja:

1. O'zgarmas tok motorlari
2. Yuqori oqimli induktiv yuklamalarni nazorat qilish
3. Transistorlarning kalit sifatida ishlatilishi
4. Ximoya diodlaridan foydalanish
5. Ikkilamchi quvvat manбайдan foydalanish
6. Dvigatelni ulash
7. Puls kengligi modulyatsiyasi yordamida motor tezligini boshqarish
8. H-ko'prigidan foydalanib, O'zgarmas tok motorining yo'nalishini boshqarish
9. H-ko'priqli sxemani qurish
10. H-ko'priqli sxemaning ishlashi

### **O'zgarmas tok motorlari**

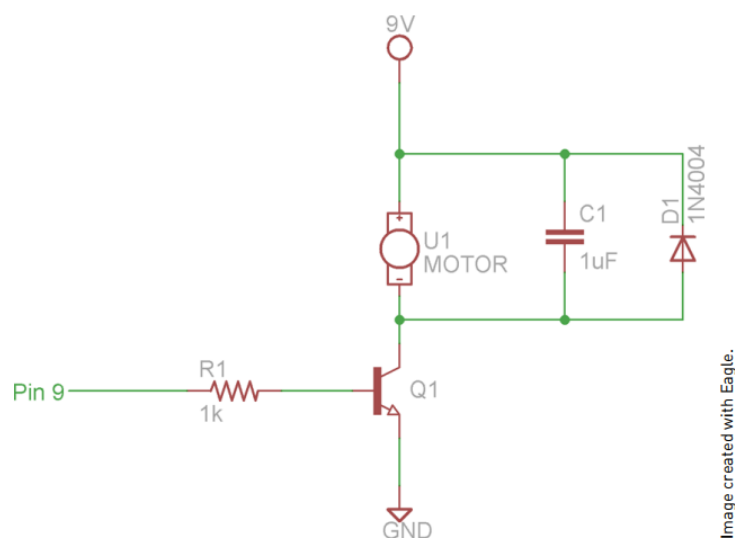
Uyingiz atrofidagi ko'plab qurilmalarda topishingiz mumkin bo'lgan o'zgarmas tok motorlari, ular bo'ylab doimiy kuchlanish qo'llanilganda, doimiy ravishda aylanadi. Bunday dvigatellar odatda radio boshqaruv (RC) avtomashinalarida harakatlantiruvchi dvigatellar va DVD pleyerda disklarni aylantiruvchi dvigatellar sifatida uchraydi. O'zgarmas tok motorlari juda yaxshi, chunki ular juda katta o'lchamlarga ega va odatda juda arzon. Ularga qo'llaniladigan kuchlanishni sozlash orqali siz ularning aylanish tezligini o'zgartirishingiz mumkin. Ularga qo'llaniladigan kuchlanish yo'nalishini teskari aylantirish orqali siz ularning aylanish yo'nalishini ham o'zgartirishingiz mumkin. Bu odatda H-ko'priqi yordamida amalga oshiriladi, siz bu bobda keyinroq bilib olasiz.

Cho'tkali o'zgarmas tok motorlari, masalan, siz ushbu bobda ishlatadigan statsionar magnitlar va aylanadigan lasanlardan foydalanasiz. Elektr "cho'tkalar" yordamida lasanga o'tkaziladi, shuning uchun ular cho'tkali o'zgarmas tok motorlari deb ataladi. Cho'tkasi bo'lmagan doimiy dvigatellardan (masalan, qadamli motorlar) farqli o'laroq, cho'tkali o'zgarmas tok motorlari arzon va tezlikni boshqarish osonroq. Biroq, cho'tkali o'zgarmas tok motorlari uzoq davom etmaydi, chunki cho'tkalar vaqt o'tishi bilan eskirishi mumkin. Bu motorlar induktiv kuch orqali ishlaydi. Agar tok aylanadigan lasan orqali o'tib ketsa, u magnit maydonini hosil qiladi, uni statsionar magnitlar tortadi yoki qaytaradi. Qutblariga bog'liq. Har yarim aylanishda qutbni almashtirish uchun cho'tkalardan foydalanib, siz burchak momentumini yaratishingiz mumkin. Agar siz ramkani qo'lda aylantirsangiz, xuddi shu konfiguratsiyani generator yaratish uchun ishlatish mumkin. Bu o'zgaruvchan magnit maydonni hosil qiladi, bu esa o'z navbatida oqim hosil qiladi. Bu generatorlar ish-tushib suv mil o'g'irsa, va hozirgi qanday gidroenergetika ishlab chiqarilmoqda. Qarama -qarshi yo'nalishda oqim hosil qilish qobiliyati, shuning uchun siz

dvigatelni majburan aylantirganda, kontaktlarning zanglashiga qaytarib yubormasligini ta'minlash uchun dioddan foydalanasiz.

### Yuqori oqimli induktiv yuklamalarni nazorat qilish

O'zgarmas tok motorlari, odatda, Arduining o'rnatilgan quvvat manbalaridan ko'ra ko'proq oqim talab qiladi va ular induktiv xarakterga ega bo'lgani uchun zararli kuchlanish uchlarini yaratishi mumkin. Ushbu muammoni hal qilish uchun siz birinchi navbatda Arduino - dan o'zgarmas tok motorini samarali ajratib olishni, so'ngra uni ikkilamchi ta'minot yordamida quvvatlantirishni o'rganasiz. Transistor Arduino-ga dvigatelni xavfsiz yoqish va o'chirishga, shuningdek "Analog sensorlarni o'qish" avvalgi darslarda o'rganilgan puls kengligi modulyatsiyasi (PWM) texnikasi yordamida tezlikni boshqarishga imkon beradi. Doimiy dvigatelni ikkinchi darajali quvvat manbai bo'lgan Arduinoga ulashning turli komponentlari haqida bilib olish uchun, quyidagirasmda ko'rsatilgan sxemaga murojaat qiling. Elektr simlarini ulashni boshlashdan oldin, bu tushunchalarning barchasini tushunganingizga ishonch hosil qiling.



O'zgarmas tok motorini boshqarish sxemasi

O'zgarmas tok motorini ulashdan oldin, ushbu komponentlarning nima qilayotganini tushunish kerak :

■ Q1-bu alohida 9V quvvat manbaini dvigatelga ulash uchun ishlatiladigan NPN bipolyar o'tish tranzistoridir. Bipolyar tranzistorlarning ikki turi mavjud, ular NPN va PNP bo'lib, ular tranzistorni yaratish uchun ishlatiladigan yarimo'tkazgichli "doping" usullariga ishora qiladi. Siz oddiygina NPN tranzistorini oqim oqimini to'xtatishga yoki ruxsat berishga imkon beradigan kuchlanish boshqariladigan kalit sifatida tasavvur qilishingiz mumkin .

■ Transistorning asosiy pinini Arduino boshqaruv pinidan ajratish uchun  $1k\Omega$  qarshilik ishlatiladi.

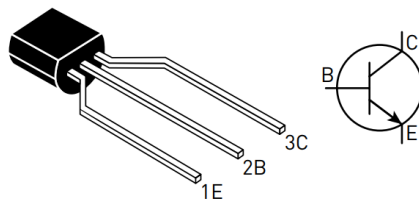
■ U1 - bu o'zgarmas tok motoridir.

■ C1 dvigateldan kelib chiqadigan shovqinlarni filtrlash uchun mo'ljallangan kondensator.

■ D1 - bu elektr ta'minotini induktör kabi harakat qiladigan vosita tufayli teskari kuchlanishdan himoya qilish uchun ishlatiladigan diod.

### Transistorlari kalit sifatida ishlatilishi

Transistorlar juda ko'p vazifalarni bajarishi mumkin: kuchaytirgich ishlab chiqarishdan tortib, kompyuter va smartfon ichidagi protsessorni yaratishgacha. Oddiy elektr boshqariladigan kalitni yaratish uchun bitta tranzistordan foydalanishingiz mumkin. Har bir bipolyar tranzistorda uchta pin bor (quyidagi rasimga qarang): emitter (E), kollektor (C) va baza (B).



Oqim kollektor orqali va emitterdan chiqadi. Asosiy pinni modulyatsiya qilish orqali siz oqimning oqishiga ruxsat berilganligini boshqarishingiz mumkin. Baza uchun etarlicha yuqori kuchlanish qo'llanilganda, tranzistor orqali oqim o'tishiga ruxsat beriladi va natijada vosita aylanadi. Arduino kirish/chiqish pinlari tomonidan ishlab chiqarilgan 5V tranzistorni yoqish uchun yetarli emas. Dvigatel tezlikni ushlab turishi mumkinligi sababli, PWM signalining aylanish vazifasi vosita tezligini aniqlaydi. Transistor, asosan, dvigatelning bitta terminalini yerdan uzib qo'yadi va batareyadan qachon to'liq zanjir yasash mumkinligini aniqlaydi.

### **Ximoya diodlaridan foydalanish**

O'zgarmas tok motorlarining induktor kabi harakat qilishidan kelib chiqadigan muammolarni ko'rib chiqish muhimdir. (Induktor - bu magnit maydonida energiyani saqlaydigan va tokning o'zgarishiga qarshilik ko'rsatadigan elektr qurilmalar.) O'zgarmas tok dvigateli aylanar ekan, energiya dvigatel bobinlarining induktansida to'planadi va saqlanadi. Agar elektr quvvati bir zumda dvigateldan chiqarilsa energiya teskari kuchlanish pog'onasi shaklida chiqariladi, bu esa elektr ta'minotiga zarar etkazishi mumkin. Bu yerda himoya diodlari paydo bo'ladi. Diodni dvigatel bo'ylab joylashtirib, siz dvigatel orqali hosil bo'ladigan tokning dioddan o'tishini va teskari kuchlanish diodning old kuchlanishidan oshmasligini ta'minlaysiz (chunki diodlar oqimning bitta oqimga o'tishiga imkon beradi). Bu, shuningdek, dvigatelni majburan aylantirish orqali hosil bo'lgan har qanday oqimni o'zlashtiradi.

### **Ikkilamchi quvvat manbaidan foydalanish**

E'tibor bering, eng yuqoridagi rasmda ko'rsatilgan elektron diagrammada, dvigatelning quvvat manbai siz foydalanayotgan USB ulanishidan odatdagi 5V o'rniga 9V. Ushbu tajriba uchun 9V batareya etarli, lekin siz AC-DC devor adapteridan ham foydalanishingiz mumkin. Arduino-ning o'rnatilgan 5V kuchlanishidan alohida quvvat manbasini ishlatishning sababi ikki xil:

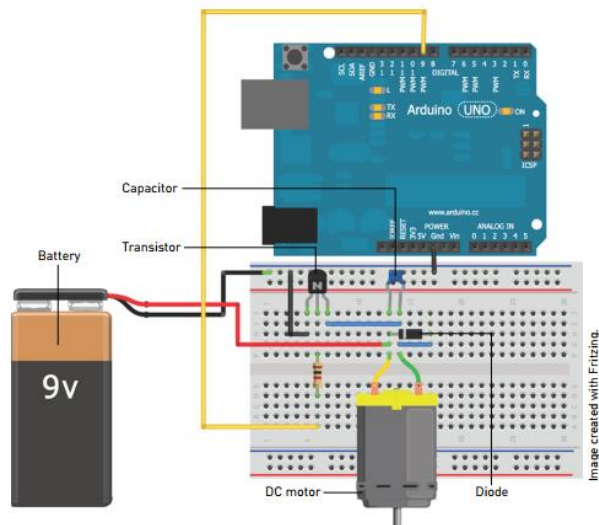
1. Alohida manbaidan foydalanib, siz yuqori quvvatli kontaktlarning zanglashiga olib kelishi Arduino'ga zarar etkazish ehtimolini kamaytirasiz.
2. Siz yuqori oqim chegaralari va yuqori kuchlanishlardan foydalanishingiz mumkin.

Ba'zi o'zgarmas tok motorlari Arduino 5V manbasidan ko'ra ko'proq oqim sarflashi mumkin. Bundan tashqari, ko'plab dvigatellar 5V dan yuqori kuchlanish bilan baholanadi. Garchi ular 5 V da aylanishi mumkin bo'lsa -da, siz ularning maksimal tezligiga atigi 9 V yoki 12 V ga erishishingiz mumkin (dvigatel xususiyatlariga qarab).

E'tibor bering, siz ikkilamchi quvvat manbaini ham , Arduino topraklamasini ham ulashingiz kerak. Bu ulanish kontaktlarning zanglashiga olib keladigan ikki qismidagi kuchlanish darajalari o'rtasida umumiy mos yozuvlar nuqtasini ta'minlaydi .

### Dvigatelni ulash

Endi siz doimiy dvigatelni boshqarishning murakkabligini tushunganingizdan so'ng, uni non paneliga ulash vaqti keldi. Faqat oldingi sxemaga (eng yuqoridagi rasmda ko'rsatilgan) havola qilib, sim o'tkazishga harakat qiling . Siz sxemani faqat sxemasi yordamida yig'ishga harakat qilganingizdan so'ng , uni to'g'ri simli ulaganingizni tasdiqlash uchun quyidagi rasmda ko'rsatilgan grafik versiyaga murojaat qiling.



Grafik sxemaga qaramasdan, elektr sxemalarini yaxshi o'qish muhim .Siz uni to'g'ri uladingizmi? Elektr simini ulashda quyidagilarni tekshirishni unutmang.

1. Erni 9V batareyadan Arduinodan yerga ulaganingizga ishonch hosil qiling . Yuqoridagi rasmda ko'rsatilgandek, buni amalga oshirish uchun siz non panelidagi gorizontali avtobusdan foydalanishingiz mumkin .
2. 9V kuchlanish 5V tarmog'iga ulanmaganligiga ishonch hosil qiling. Aslida, siz hatto 5V kuchlanishli panelni simga ulashingiz shart emas.
3. Transistoringizning yo'nalishi to'g'ri ekanligiga ishonch hosil qiling. Agar siz ushbu bo'lim uchun qismlar ro'yxatida ko'rsatilgan bir xil NPN bipolyar tranzistordan foydalanmasangiz, emitter, baza va kollektor bir xil pinlarga ulanganligiga ishonch hosil qilish uchun ma'lumotlar jadvaliga murojaat qiling . Agar ular bo'lmasa, simni sozlang.
4. Diodning yo'nalishi to'g'ri ekanligiga ishonch hosil qiling. Oqim chiziqsiz tomondan chiziq bilan yon tomonga oqadi . Jismoniy qurilmadagi chiziq sxematik belgidagi chiziq bilan mos keladi. Ushbu mashq uchun siz keramika kondansatörünü ishlatasiz, shuning uchun qutbning ahamiyati yo'q.

Endi, bu dvigatelni aylantirish vaqti keldi. Dvigatelning uchiga lentani yoki g'ildirakni yopishtirishni xohlashingiz mumkin, shunda siz uning aylanish tezligini osonroq ko'rasiz. Dasturni yozishdan oldin , Arduino -ni USB ulanishi orqali quvvat bilan ta'minlash , 9V batareyani ulash va tranzistorning asosiy pinini (rezistordan keyin) to'g'ridan -to'g'ri Arduinodan 5V ga ulash orqali sxemaning to'g'ri ishlashini tasdiqlashingiz mumkin . Bu mantiqiy yuqori

buyruqni simulyatsiya qiladi va dvigatelning aylanishiga olib kelishi kerak. Xuddi shu simni erga ulash uning aylanmasligini ta'minlaydi. Agar bu ishlamas , keyingi bosqichga o'tishdan oldin simni tekshiring.

### **Puls kengligi modulyatsiyasi yordamida motor tezligini boshqarish**

Birinchiidan , siz dvigatel tezligini sozlash uchun avvalgi darslarda tungi yorug'lik LED yoritilishini sozlashda ishlatilgan dasturga juda o'xshash dasturdan foydalanishingiz mumkin. Tomonidan tranzistorlar uchun turli vazifa-asr signallari yuborib, orqali joriy oqimi motor tez boshlanadi va tezligi o'zgarishi natijasida to'xtaydi. Dvigatel tezligini tez-tez ko'tarish va tushirish uchun quyidagi dasturni sinab ko'ring .

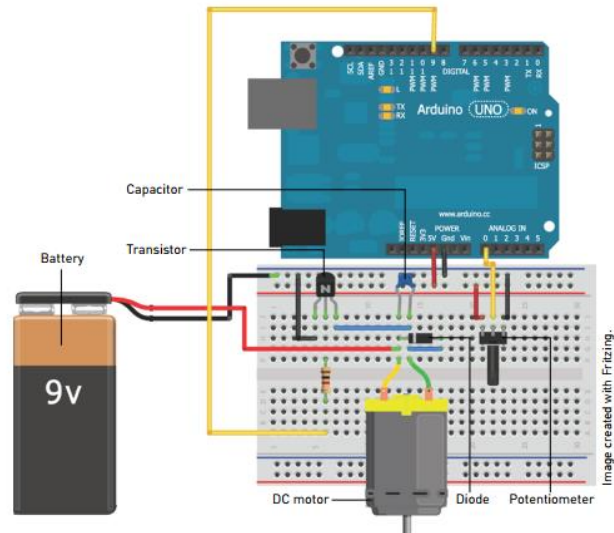
```
//Motor tezligini boshqarish dasturi

const int MOTOR=9; //9-pinga motor ulangan

void setup()
{
  pinMode (MOTOR, OUTPUT);
}
void loop()
{
  for (int i=0; i<256; i++)
  {
    analogWrite(MOTOR, i);
    delay(10);
  }
  delay(2000);
  for (int i=255; i>=0; i--)
  {
    analogWrite(MOTOR, i);
    delay(10);
  }
  delay(2000);
}
```

Agar hamma narsa to'g'ri ulangan bo'lsa, bu kod dvigatel tezligini asta -sekin oshirib , keyin asta orqaga qaytarishi kerak. Ushbu texnikadan foydalanib, siz oddiy aravachali robot yasashingiz mumkin.

Keyinchalik, siz O'zgarimas tok motorlari haqidagi yangi bilimlaringizni analog sensorlar haqidagi bilimlaringiz bilan birlashtira olasiz . Potansiyometr yordamida siz motor tezligini qo'lda sozlashingiz mumkin . Boshlash uchun 4-4-rasmda ko'rsatilgandek, analog pin 0 ga potentsiometr qo'shing . E'tibor bering, agar siz potentsiometrni bordagi o'sha qatorga ulashni xohlasangiz , Arduino -dan 5V -pinni taxtadagi elektr rayiga ulashingiz kerak .



Potensiometrni qo'shish

Siz potensiometrning hozirgi sozlamalari asosida motor tezligini boshqarish uchun dasturni o'zgartirishingiz mumkin. Potensiometr nol bo'lsa, vosita to'xtaydi; potensiometr to'liq aylantirilganda, vosita to'liq tezlikda ishlaydi. Eslatib o'tamiz, Arduino juda tez ishlaydi; aslida har soniyada bir necha ming marta aylana bo'ylab harakat qiladi! Shunday qilib, siz potensiometr tezligini har safar pastadir orqali tekshirishingiz va har bir tekshiruvdan keyin vosita tezligini sozlashingiz mumkin. Dvigatel tezligi potensiometr yordamida real vaqtda sozlanishini tez-tez tekshiradi. Quyidagi kod bu hiylani bajaradi. Yangi eskiz yarating (yoki oldingi eskizni ushbu kodga mos keladigan tarzda yangilang) va uni Arduinoga o'rnatilgan ishlab chiqish muhitidan (IDE) yuklang.

// Potensiometr yordamida motor tezligini boshqarish dasturi

```
const int MOTOR=9; //Motor on Digital Pin 9

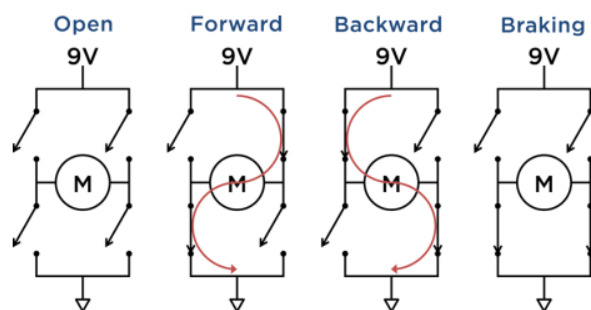
const int POT=0; //POT on Analog Pin 0
int val = 0;
void setup()
{
  pinMode (MOTOR, OUTPUT);
}
void loop()
{
  val = analogRead(POT);
  val = map(val, 0, 1023, 0, 255);
  analogWrite(MOTOR, val);
}
```

Ushbu kodning ko'p qismi analog sensorlar bilan ishlash tajribangizdan tanish bo'lishi kerak. E'tibor bering, potensiometrdan foydalanishda cheklash funktsiyasi talab qilinmaydi, chunki siz butun kirish diapazonidan foydalanishingiz mumkin va bu qiymat hech qachon 023 dan past yoki

1023 dan past bo'lmaydi. Arduino -ga kodni yuklaganingizdan so'ng , potni rostlang va tezlikni kuzating. Dvigatel shunga mos ravishda o'zgaradi.

### **H-ko'prigidan foydalanib, O'zgaras tok motorining yo'nalishini boshqarish**

Shunday qilib, siz o'zgaras tok motorining tezligini o'zgartirishingiz mumkin. Bu Arduino tomonidan boshqariladigan robotda g'ildirak yasash uchun juda yaxshi ... faqat oldinga siljishini xohlasangiz. Har qanday foydali o'zgaras tok dvigateli ikki yo'nalishda aylanishi kerak. Buni amalga oshirish uchun siz H-ko'prigi deb nomlangan qulay qurilmadan foydalanishingiz mumkin. H-ko'prigining ishini diagramma bilan yaxshiroq tushuntirish mumkin (Quyidagi rasmga qarang).



Nima uchun u H-ko'prik deb nomlanganini tushuna olasizmi? E'tibor bering, dvigatel to'rtta kalit bilan birgalikda katta harfni hosil qiladi. Diagrammada ularni kalit sifatida ko'rsatsada, kommutatsiya komponentlari aslida oldingi mashqda ishlatilganlarga o'xshash tranzistorlardir. H-ko'prikli integral mikrosxemaga himoya diodlarini o'z ichiga olgan ba'zi qo'shimcha sxemalar ham o'rnatilgan.

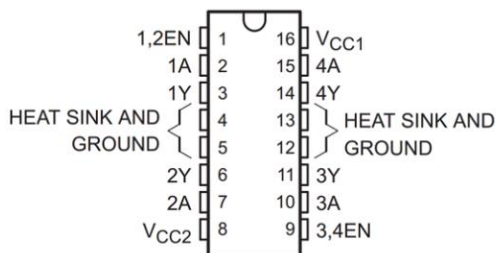
H-ko'prigi to'rtta asosiy holatga ega: ochiq, tormoz, oldinga va orqaga. Ochiq holatda, barcha kalitlar ochiq va dvigatel aylanmaydi. Oldinga holatda, diagonal qarama -qarshi bo'lgan ikkita kalit ishga tushadi , bu esa oqimning 9V dan, dvigateldan va erga tushishiga olib keladi. Qachon qarama-qarshi kalitlari çevrileceği, joriy keyin motor orqali ishlaydi qarama-qarshi yo'nalishda Spin uni sabab, qarama-qarshi yo'nalishda. Agar H-ko'prigi tormoz holatiga o'tkazilsa, momentumdan kelib chiqqan barcha qoldiq harakatlar to'xtatiladi va vosita to'xtaydi.

### **H-ko'prikli sxemani qurish**

Yuqoridagi fikrlarni hisobga olgan holda, sxemani qurish vaqti keldi. H-ko'prik chip SN754410 Quadruple yarim H-drayveridan foydalanish mumkin. Ikki yarim-H drayveri bitta Full-H drayveriga birlashtirilgan, masalan yuqoridagi rasmda ko'rsatilgan. Ushbu mashqni bajarish uchun siz bitta O'zgaras tok motorini boshqarish uchun to'rtta Half-H drayveridan ikkitasini ishlatasiz Agar siz RC avtomashinasini ishlab chiqarishni xohlasangiz, masalan, siz ushbu chip yordamida ikkita o'zgaras tok motorini boshqarishingiz mumkin (bittasi chap g'ildiraklar uchun, ikkinchisi o'ng g'ildiraklar uchun). Haqiqatan ham, uni ulashdan oldin, qismlar jadvalidan ajratish va mantiqiy jadvalga qarash lozim.

Integral mikrosxemalarda pinni raqamlash har doim yuqori chap pindan boshlanadi va soat sohasi farqli ravishda uning atrofida aylanadi. Chiplarda har doim qanday pin borligini ko'rsatish uchun biron bir ko'rsatkich bo'ladi, shunda siz ICni teskari tomonga ulamaysiz. Teshikli qismlarda (bu faqat shu bobda ishlatiladi), chipning bir uchidagi yarim doira chipning yuqori

qismini ko'rsatadi (1-pin joylashgan). Ba'zi mikrosxemalarda , yarim doira o'rniga, yoki uning o'rniga plastik korpusga mahkamlagichning yonida kichik doira o'rnatilgan bo'lishi mumkin .



Kirishlar		Chiqish
A	EN	Y
H	H	H
L	H	L
X	L	Z

H – yuqori daraja  
L – pastki daraja  
Z – yuqori qarshilik

Keling, pinlarni ko'rib chiqaylik va ularni qanday ishlatish mumkin:

- GND (4, 5, 12, 13 -pinlar): O'rtadagi to'rtta pin 9V va 5V quvvat manbalari o'rtasida umumiy yerga ulanadi .
- VCC2 (8 -pin): VCC2 dvigatel oqimini beradi, shuning uchun siz uni 9V ga ulaysiz.
- VCC1 (16 -pin): VCC1 chip mantig'ini quvvatlaydi, shuning uchun siz uni 5V ga ulaysiz.
- 1Y va 2Y (3 va 6 -pinlar): Bu chap drayverning chiqishlari. Motor simlari, bu pinlarga ulanadi
- 1A va 2A (2 va 7 -pinlar): Chapdagi kalitlarning holati bu pinlar tomonidan boshqariladi, shuning uchun ular Arduino dagi o'tish -chiqish pinlariga ulanadi .
- 1,2EN (1 -pin): Bu pin chap drayverni yoqish yoki o'chirish uchun ishlatiladi. U Arduinodagi PWM piniga ulangan, shuning uchun tezlikni dinamik boshqarish mumkin .
- 3Y va 4Y (11 va 14 -pinlar): Bu to'g'ri drayverning chiqishlari. Siz faqat chap drayverdan foydalanayotganingiz uchun ularni uzilgan holda qoldirishingiz mumkin.
- 3A va 4A (10 va 15 -pinlar): O'ngdagi kalitlarning holati bu pinlar tomonidan boshqariladi, lekin siz bu misolda faqat chap drayverni ishlatasiz, shuning uchun ularni uzib qo'yishingiz mumkin.
- 3,4EN (9 -pin): Bu pin to'g'ri drayverni yoqish yoki o'chirish uchun ishlatiladi. Agar siz to'g'ri drayverni ishlatmasangiz, bu pinni to'g'ridan -to'g'ri GND ga ulab o'chirib qo'yishingiz mumkin .

Malumot uchun, sxamangizni quyidagi rasm bilan taqqoslang. Potensiometrni simli holatda saqlang, xuddi quyidagi rasmda bo'lgani kabi : H-ko'priqli simlarni ulash sxemasi. Siz uni dasturlashdan oldin, kontaktlarning zanglashini 5V ga ulash, A pinlaridan birini yerga, ikkinchisini 5V ga ulash orqali tasdiqlashingiz mumkin. A pinlari ulangan narsalarni almashtirish orqali siz yo'nalishni teskari o'zgartirishingiz mumkin.

### H-ko'priqli sxemaning ishlashi

Keyinchalik, siz potensiometr va H-ko'priqi yordamida dvigatelning yo'nalishini va tezligini boshqaradigan dastur yozasiz. Potansiyometrni o'rta diapazonda o'rnatish dvigatelni to'xtatadi, potensiometrni o'rtasidan yuqori diapazonga qo'yish oldinga tezlikni oshiradi va potensiometrni o'rtasidan pastroqqa qo'yish tezlikni orqaga oshiradi. Bu Arduino dasturida funktsiyalarni ishlatish

uchun yana bir ajoyib imkoniyat . Siz dvigatelni to'xtatish funksiyasini yozishingiz mumkin , bunda u belgilangan tezlikda oldinga siljiydi, ikkinchisi esa belgilangan tezlikda orqaga aylanadi . Dvigatel rejimini o'zgartirishdan oldin, funktsiya boshida H-ko'prigini to'g'ri o'chirib qo'yganingizga ishonch hosil qiling ; bu sizning xato qilish ehtimolini kamaytiradi va tasodifan H-ko'prigini qisqartiradi.

Yuqoridagi rasmdagi mantiqiy diagrammadan so'ng, kerakli natijalarga erishish uchun pinlarni qanday boshqarish kerakligini tezda aniqlashingiz mumkin :

- ■ Qurilma orqali oqim oqimini to'xtatish uchun yoqish pinini past qilib qo'ying.
- ■ Kalitlarni bir tomonga burilishga o'rnatish uchun birini baland, ikkinchisini past qilib qo'ying.
- ■ Kalitlarni teskari yo'nalishda aylantirish uchun, baland va pastni almashtiring .
- ■ Dvigatel zudlik bilan to'xtab qolishi uchun ikkala kalitni ham pastga qo'ying.

ESLATMA. Kalitlarning holatini o'zgartirmasdan oldin, oqim oqimini har doim o'chirib qo'ying, shunda kalitlar aylanayotganda bir lahzalik qisqa tutashuvni yaratib bo'lmaydi.

Birinchiidan, siz ilgari tasvirlangan harakatlarni xavfsiz bajaradigan funktsiyalarni o'ylab topishingiz kerak. Yangi Arduino eskizini yarating va yangi funktsiyalarni yozishdan boshlang :

```
//Motor oldinga harakatlanadi
```

```
void forward (int rate)
{
digitalWrite(EN, LOW);
digitalWrite(MC1, HIGH);
digitalWrite(MC2, LOW);
analogWrite(EN, rate);
}
//Motor orqaga harakatlanadi
void reverse (int rate)
{
digitalWrite(EN, LOW);
digitalWrite(MC1, LOW);
digitalWrite(MC2, HIGH);
analogWrite(EN, rate);
//Motor tormoz
void brake ()
{
digitalWrite(EN, LOW);
digitalWrite(MC1, LOW);
digitalWrite(MC2, LOW);
digitalWrite(EN, HIGH);
}
}
```

E'tibor bering, har bir funktsiyaning boshida EN pin har doim past bo'ladi, keyin esa MC1 va MC2 pinlari (Dvigatelni boshqarish pini) o'rnatiladi. Bu bajarilganda, oqim oqimi qayta yoqilishi mumkin. Tezlikni o'zgartirish uchun avval qilgan texnikadan foydalaning . PWM dan foydalanib,

siz EN pinining almashtiriladigan burchini o'zgartirishingiz va tezlikni boshqarishingiz mumkin. Tezlik o'zgaruvchisi 0 dan 255 gacha bo'lishi kerak. Asosiy tsikl kirish potentsiometrining ma'lumotlaridan to'g'ri tezlikni yaratishga e'tibor beradi.

Endi, dasturning asosiy tsiklini ko'rib chiqing:

```
void loop()
{
  val = analogRead(POT);
  //go forward
  if (val > 562)
  {
    velocity = map(val, 563, 1023, 0, 255);
    forward(velocity);
  }
  //go backward
  else if (val < 462)
  {
    velocity = map(val, 461, 0, 0, 255);
    reverse(velocity);
  }
  //brake
  else
  {
    brake();
  }
}
```

Asosiy tsiklda potentsiometr qiymati o'qiladi va potentsiometr qiymatiga qarab tegishli funktsiyani chaqirish mumkin. Eslatib o'tamiz, analog kirish 0 dan 1023 gacha raqamli qiymatlarga o'tkaziladi.

Potentsiometr o'rta nuqtani o'rab turgan 100 birlik ichida bo'lsa, tormoz funktsiyasi chaqiriladi. Potentsiometr qiymati 562 dan 1023 gacha oshishi bilan oldinga tezlik oshadi. Xuddi shunday, tezlik 462 va 0 potentsiometrlari orasidagi teskari yo'nalishda ortadi. Xarita funktsiyasi sizga oldingi bobdan tanish bo'lishi kerak. Bu erda, teskari tezlikni aniqlayotganda, o'zgaruvchilar tartibiga e'tibor bering: 461 - 0 ga, 0 - 255 ga teng; xarita funktsiyasi o'zgaruvchilar kamayish tartibida o'tkazilganda xaritani o'zgartirishi mumkin. Loopni funktsiyalar va sozlamalar bilan birlashtirib, siz Quyidagi kodda ko'rsatilgan dasturga o'xshab tugallangan dasturni olasiz. Sizning dasturingiz bu yerga mos kelishiga ishonch hosil qiling va uni Arduinoga yuklang.

```
//Hbridge Motor kontrol
const int EN=9; //Half Bridge 1 Enable
const int MC1=3; //Motor kontrol 1
const int MC2=2; //Motor kontrol 2
const int POT=0; //POT Analog Pin Oda
int val = 0; // POTdan keluvchi signal qiymatini o'qish uchun
int velocity = 0; //Mo'ljallangan tezlikni to'plash uchun ( 0-255)
```

```

void setup()
{
pinMode(EN, OUTPUT);
pinMode(MC1, OUTPUT);
pinMode(MC2, OUTPUT);
brake(); //Motor tormozlangan holatda boshlash
}
void loop()
{
val = analogRead(POT);
//oldinga
if (val > 562)
{
velocity = map(val, 563, 1023, 0, 255);
forward(velocity);
}
//oldinga
else if (val < 462)
{
velocity = map(val, 461, 0, 0, 255);
reverse(velocity);
}
//tormoz
else
{
brake();
}
}
//Motor goes forward at given rate (from 0-255)
void forward (int rate)
{
digitalWrite(EN, LOW);
digitalWrite(MC1, HIGH);
digitalWrite(MC2, LOW);
analogWrite(EN, rate);
}
//Motor goes backward at given rate (from 0-255)
void reverse (int rate)
{
digitalWrite(EN, LOW);
digitalWrite(MC1, LOW);
digitalWrite(MC2, HIGH);
analogWrite(EN, rate);
}
//Stops motor
void brake ()

```

```
{  
digitalWrite(EN, LOW);  
digitalWrite(MC1, LOW);  
digitalWrite(MC2, LOW);  
digitalWrite(EN, HIGH);  
}
```

Hammasi kutilganidek ishlayaptimi? Agar yo'q bo'lsa, kontaktlarning to'g'ri ulanganligiga ishonch hosil qiling. Qo'shimcha qiyinchilik sifatida, ikkinchi o'zgarmas tok motorini oling va uni H-ko'prik chipining boshqa yarmiga ulang. Siz minimal harakat bilan bir vaqtning o'zida ikkita dvigatelni boshqarishingiz mumkin.

### Foydalanilgan Adabiyotlar

- Jeremy Blum, “Exploring arduino: Tools and techniques for engineering wizardry”, 2013
- Electronics: A Systems Approach (6th edition), Neil Storey, Pearson Education UK, 2017
- Arduino.cc veb sayti ma'lumotlari