

Upravljanje DC motorom pomoću platforme Arduino

Za upravljanje raznim istosmjernim motorima koriste se mikroupravljači. U slučaju da mikroupravljači ne mogu podnijeti napone i struje potrebne istosmjernom motoru, koriste se upravljački sklopovi. U tom slučaju mikroupravljači upravljaju upravljačkim sklopom koji je spojen na viši napon i istosmjerni motor. **DC motor** je najčešći elektromotor koji se koristi u prijenosnim uređajima, igračkama, radio kontroliranim modelima i drugim uređajima. Stalni magneti učvršćeni su na malom elektromotoru na statoru, a namatanje na rotoru.

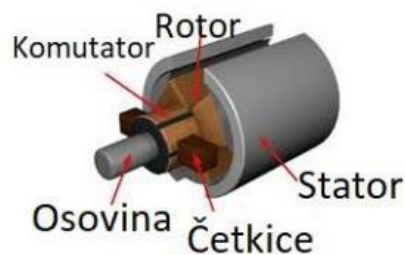
Istosmjerni motor je elektromehanički uređaj koji električnu energiju pretvara u mehaničku odnosno pretvara istosmjernu struju u kružno kretanje, ali mogu raditi i u suprotnom smjeru i pretvarati kružno kretanje u električnu energiju. Mali istosmjerni motori su uređaji koji rade na istosmjernom naponu što se može zaključiti i po nazivu, lakše im je kontrolirati brzinu i zbog toga su najpogodniji za razne hobi projekte. Istosmjerni motori nalaze primjenu u velikom broju uređaja, tako da ako ste nekada rastavili neku igračku ili uređaj koji je imao pokretnih dijelova unutra se nalazio istosmjerni motor koji je sve pokretao (slika 1.).



Slika 1.

OSNOVNI DIJELOVI

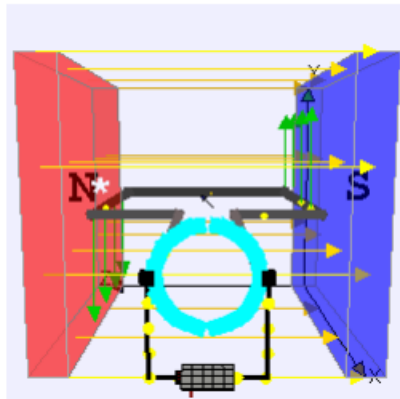
Glavni dijelovi motora su rotor i stator kao i kod svakog drugog motora, ali ova vrsta motora ima još i komutator s grafitnim četkicama. Istosmjerni motori se izvode s nezavisnom, serijskom ili složenom uzbuđom ili se na statoru kao uzbuđa nalazi permanentni magnet. Ovdje ćemo opisati istosmjerni motor koji ima permanentni magnet na statoru, a na rotoru se nalaze namotaji koji se pomoću komutatora i grafitnih četkica povezuju na napajanje. Komutator se sastoji od lamela koje su povezane na namotaje rotora i s kojim se smjer struje kroz namotaje prilikom svakog okretaja rotora dva puta mijenja.



Slika 2.

KAKO RADI?

Osnovno načelo koje koriste svi motori je da na vodič kroz koji teče struja u magnetskom polju djeluje sila. Kod istosmjernog motora magneti na statoru nam stvaraju magnetsko polje, a kroz namotaje rotora propuštamo struju i na namotaje djeluje sila koja zakreće rotor. Za primjer uzmimo da imamo jedan namotaj i komutator s dvije lamele na koje su spojeni krajevi namotaja. Lamela 1 se spaja na plus, a lamela 2 na minus. Kako se namotaj nalazi u magnetskom polju na njega djeluje sila i zakreće namotaj, prilikom zakretanja namotaja lamele 1 i 2 se prespajaju te je sada lamela 1 spojena na minus, a lamela 2 na plus te se zbog toga rotor i dalje zakreće i cijeli ciklus se ponavlja. Rotor motora se sastoji od više namotaja kako bi dobili veću silu, a zbog više namotaja imamo i više lamela na koje spajamo namotaje. Animacija nam prikazuje kako to izgleda te je iz nje lakše shvatiti kako motor radi i kako se namotaj stalno zakreće.



Slika 3.

Ako rotor istosmjernog motora zakrenemo na njegovim izvodima ćemo dobiti napon proporcionalan brzini vrtnje motora, tako da nam ovi motori mogu služiti i kao generatori električne struje.

KAKO KORISTITI?

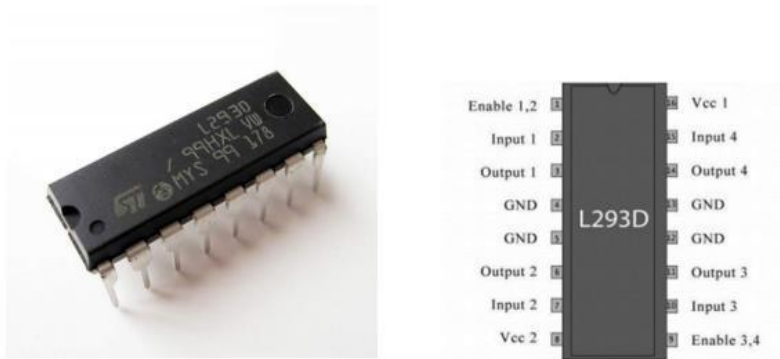
Istosmjerni motor ima 2 izvoda, jedan se spaja na plus pol napajanja, a drugi na minus. Ako želimo promijeniti smjer vrtnje motora samo trebamo zamijeniti plus i minus pol i motor će se vrtjeti u drugome smjeru. Brzinu motora mijenjamo s promjenom napona napajanja motora, ali moramo paziti da motoru ne dovedemo prevelik napon da ga ne oštetimo.

Za primjenu promjenjivog napona možemo koristiti PWM tehniku. Za preokretanje struje možemo koristiti H-Bridge sklop ili IC-ove pokretača motora koji koriste tehniku H-Bridge.

1. Zadatak: Pokretanje DC motora

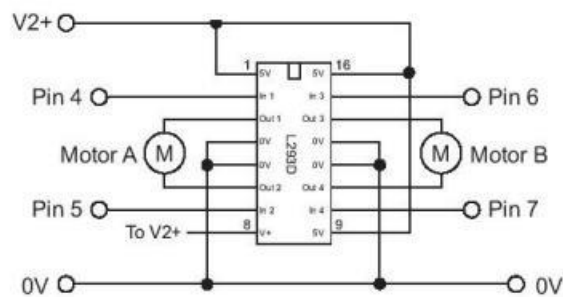
Integrirani sklop za upravljanje motorom - L293D

L293D je integrirani sklop za pogon motora s dvostrukim H-mostom (IC). Pogoni motora djeluju kao strujna pojačala budući da primaju niskostrujni kontrolni signal i daju signal veće struje. Ovaj signal veće struje koristi se za pogon motora.



Slika 4.

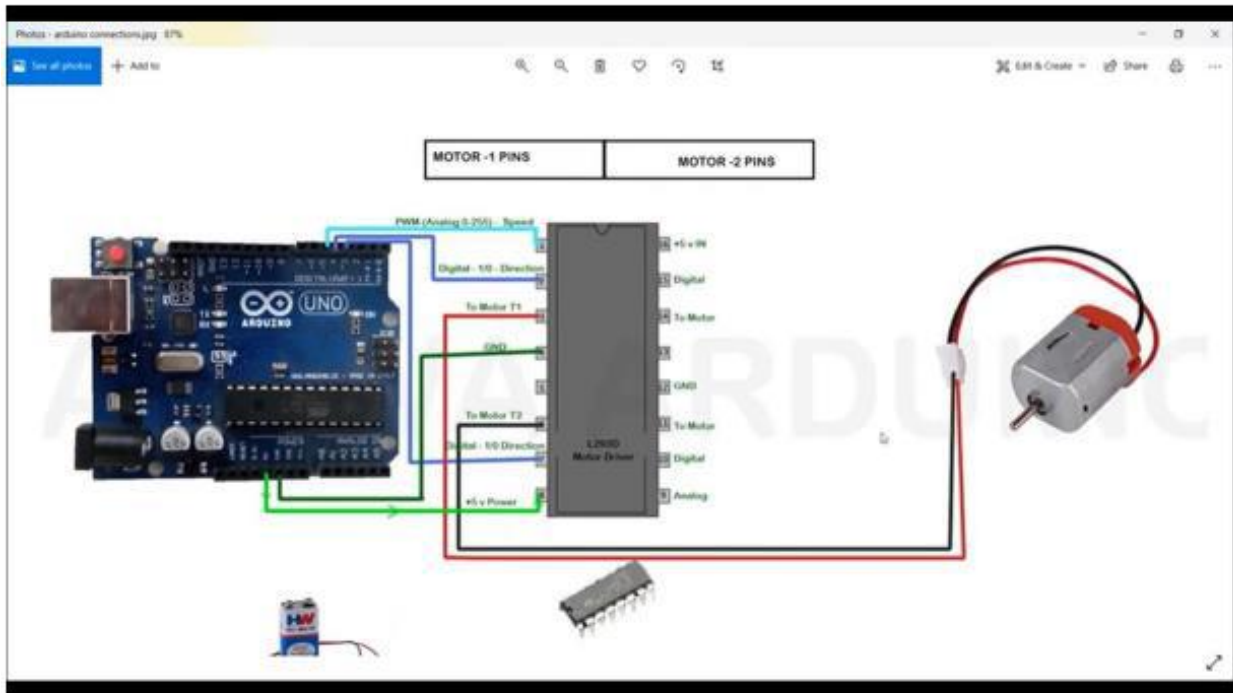
L293D sadrži dva ugrađena upravljačka kruga H-mosta. U svom uobičajenom načinu rada, dva istosmjerna motora mogu se pokretati istovremeno, i u smjeru naprijed i unatrag. Radovi motora dvaju motora mogu se kontrolirati ulaznom logikom na pinovima 2 i 7 i 10 i 15. Ulazna logika 00 ili 11 zaustavit će odgovarajući motor. Logički 01 i 10 će ga rotirati u smjeru kazaljke na satu i suprotnom smjeru kazaljke na satu. Omogućavanje pinova 1 i 9 (odgovarajući na dva motora) moraju biti visoki da bi motori počeli raditi.



Slika 5.

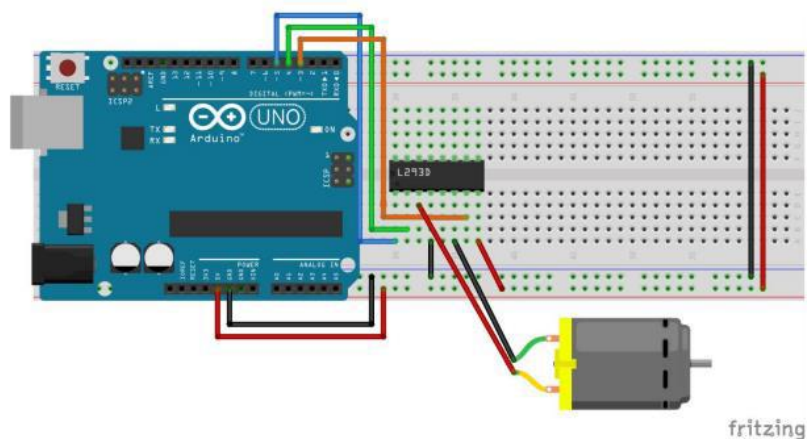
Kada je ulaz za omogućavanje visok, pridruženi upravljački program postaje omogućen. Kao rezultat toga, izlazi postaju aktivni i rade u fazi sa svojim ulazima. Slično, kada je ulaz za omogućavanje nizak, taj pokretač je onemogućen, a njihovi izlazi su isključeni i u stanju visoke impedancije. Dizajnirani su za pokretanje induktivnih opterećenja kao što su releji, solenoidi, istosmjerni i bipolarni koračni motori, kao i njihova visokostrojna/visokonaponska opterećenja u aplikacijama s pozitivnim napajanjem. Jedna od najčešćih primjena je upravljanje motorima u robotima. Zove se H-most jer koristi četiri tranzistora povezana na takav način da shematski dijagram izgleda kao "H".

Prikaz spajanja:



Slika 6.
Pokretanje DC motora

U ovom zadatku potrebno je napisati programski kod koji će pokrenuti motor te mu povećavati i smanjivati broj okretaja, a istovremeno i upravljati smjerom vrtnje motora. U ovom zadatku potrebno je na eksperimentalnu pločicu spojiti driver za motor L293D, te spojiti njegove izvodima na Arduino UNO pločici. Također na driver L293D treba spojiti i istosmjerni motor. Nakon spajanja i povezivanja komponenti zadatak je napisati programski kod koji će pokrenuti motor te mu povećavati i smanjivati broj okretaja, a istovremeno i upravljati smjerom vrtnje motora.

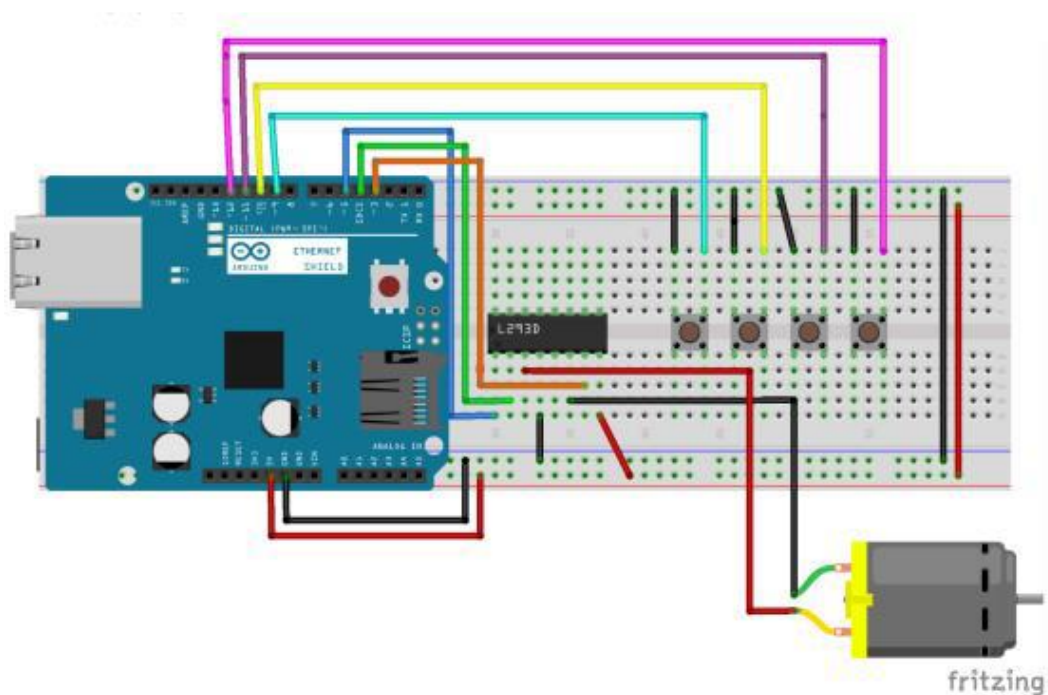


Programski kod rješenja:

```
void setup() {  
  pinMode (5, OUTPUT); // digitalni izvod PMW (analogno od 0 – 255) brzina vrtnje  
  pinMode (4, OUTPUT); // digitalni izvod o kojem ovisi smjer vrtnje  
  pinMode (3, OUTPUT); // digitalni izvod o kojem ovisi smjer vrtnje  
}  
  
void loop() {  
  analogWrite (5, 255); // izvod 5 spojen na najveću brzinu  
  digitalWrite (4, LOW); // smjer vrtnje – kazaljke na satu  
  digitalWrite(3, HIGH); // smjer vrtnje – kazaljke na satu  
  delay (5000); // vrijeme okretanja propelera  
  
  analogWrite(5, 125); // izvod 5 spojen na srednju brzinu  
  digitalWrite(4, LOW);  
  digitalWrite(3, HIGH);  
  delay (5000);  
  
  analogWrite(5, 85); // izvod 5 spojen na najmanju brzinu  
  digitalWrite(4, LOW);  
  digitalWrite(3, HIGH);  
  delay (5000);  
  
  analogWrite(5, 125);  
  digitalWrite(4, HIGH); // smjer vrtnje – obrnuto od kazaljke na satu  
  digitalWrite(3, LOW); // smjer vrtnje – obrnuto od kazaljke na satu  
  delay (5000);  
  
  analogWrite(5, 125); // izvod 5 spojen na određenu brzinu  
  digitalWrite(4, LOW); // zaustavlja vrtnju motora  
  digitalWrite(3, LOW); // zaustavlja vrtnju motora  
  delay (5000);  
}
```

2. Zadatak: Upravljanje DC motora pomoću tipkala

U ovom zadatku potrebno je spojiti 4 tipkala kojima ćemo upravljati radom motora. Prvo tipkalo pokreće motor na određenu brzinu okretanja. Drugo tipkalo treba zaustaviti okretanje motora. Treće tipkalo pokreće motor na maksimalnu brzinu. Četvrto tipkalo pokreće motor u suprotnom smjeru (obrnuto od kazaljke na satu).



Programski kod rješenja:

```

int tpk1=9;
int tpk2=10;
int tpk3=11;
int tpk4=12;
int stanje = LOW;

void setup() {
pinMode (tpk1, INPUT_PULLUP);
pinMode (tpk2, INPUT_PULLUP);
pinMode (tpk3, INPUT_PULLUP);
pinMode (tpk4, INPUT_PULLUP);
pinMode (5,OUTPUT);
pinMode (4,OUTPUT);
pinMode (3,OUTPUT);
}
void loop() {

if(digitalRead(tpk1)==LOW) {
analogWrite(5, 255);
digitalWrite(4, LOW);
digitalWrite(3, HIGH);
}

if(digitalRead(tpk2)==LOW) {
analogWrite(5, 225);
digitalWrite(4, LOW);
digitalWrite(3, LOW);
}

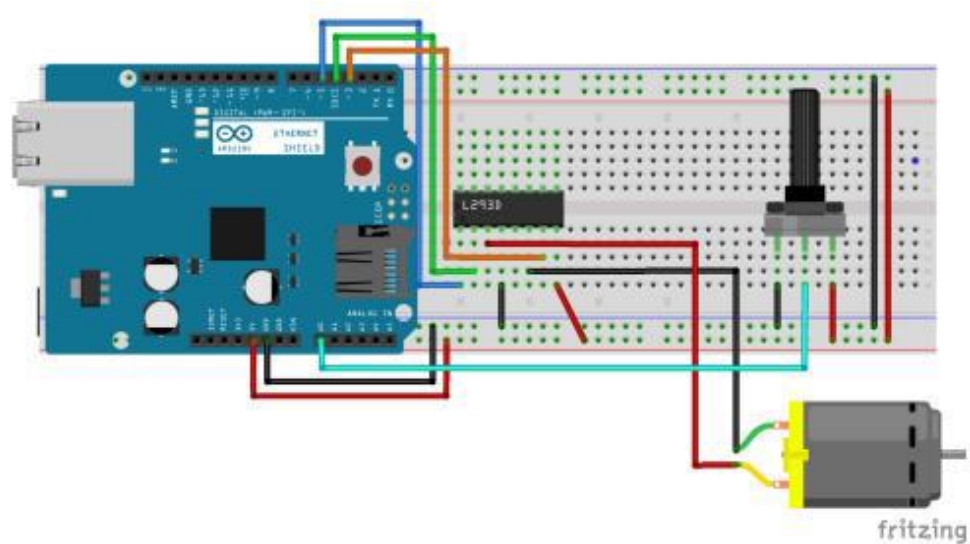
if(digitalRead(tpk3)==LOW) {
analogWrite(5, 155);
digitalWrite(4, LOW);
digitalWrite(3, HIGH);
}

if(digitalRead(tpk4)==LOW){ //ako je pritisnuto tipkalo
analogWrite(5, 150);
digitalWrite(4, HIGH); // mijenja se smjer okretanja motora
digitalWrite(3, LOW); // mijenja se smjer okretanja motora
}
}

```


3. Zadatak: Upravljanje DC motorom pomoću potenciometra

Zadatak je spojiti potenciometar na eksperimentalnu pločicu kao što je prikazano na slici, te napisati program koji će omogućiti potenciometru da upravlja brzinom okretanja motora.



Programski kod rješenja:

```
int pot = A0;           //definiraj potenciometar = A0
int ocitanjepot;       //definiraj varijablu ocitanjepot
int jacinaled;         //definiraj varijablu jacinaled

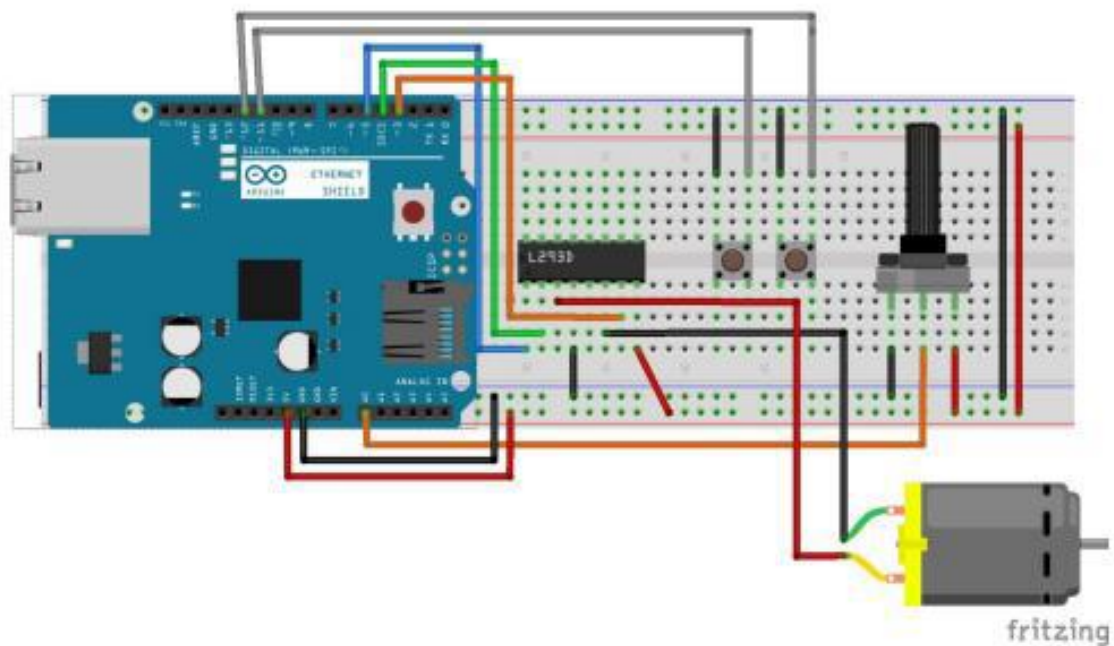
void setup() {
  pinMode(5,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(3,OUTPUT);
  Serial.begin(9600);   //inicijaliziraj serijske veze
}

void loop() {
  ocitanjepot=analogRead(pot); //očitaj stanje analognog ulaza
  jacinaled=map(ocitanjepot, 0, 1023, 0, 255);
                          //skaliraj očitane vrijednosti na 0-255
  analogWrite(5, jacinaled); //postavi jačinu LED diode
  digitalWrite(4, LOW);
  digitalWrite(3, HIGH);
  Serial.print(ocitanjepot); //pošalji očitavanje serijskim putem
  Serial.print("\t");       //pošalji znak tabulator serijskim putem
  Serial.println(jacinaled); //pošalji jačinu LED serijskim putem

  delay(2000);             //čekaj 50 ms
}
```


4. Zadatak: Upravljanje DC motora s tipkalima i potenciometrom

U ovom zadatku spojite dva tipkala i potenciometar na eksperimentalnu pločicu, te ih povežite na Arduino UNO pločicu. Zatim upišite programski kod koji će omogućiti da prvo tipkalo rotira motor u smjeru kazaljke na satu uz potenciometar kojim ćemo mijenjati brzinu rotiranja, a drugo tipkalo da omogući sve isto, samo rotiranje motora treba biti u obrnutom smjeru od kazaljke na satu.



Programski kod rješenja:

```

int tpk1=11;
int tpk2=12;
int pot = A0;           //definiraj potenciometar = A0
int ocitanjepot;       //definiraj varijablu ocitanjepot
int jacinaled;         //definiraj varijablu jacinaled

void setup() {

  pinMode(tpk1, INPUT_PULLUP);
  pinMode(tpk2, INPUT_PULLUP);
  pinMode (5,OUTPUT);
  pinMode (4,OUTPUT);
  pinMode (3,OUTPUT);
  Serial.begin(9600);   //inicijaliziraj serijske veze
  // put your setup code here, to run once:
}

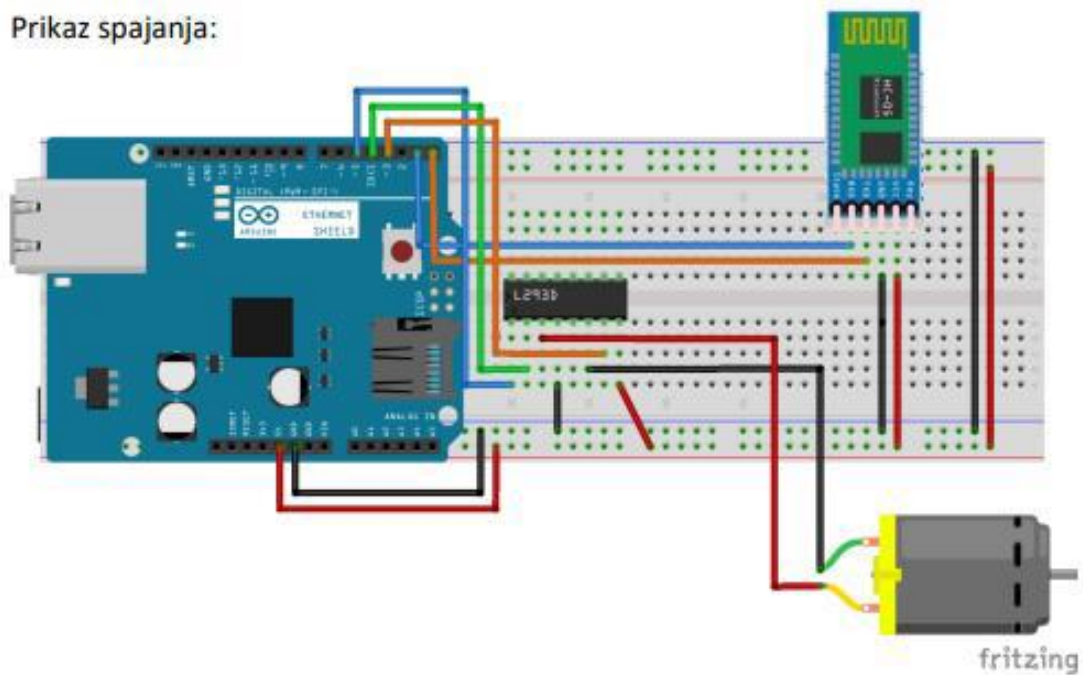
void loop() {
if (digitalRead(tpk1)==LOW){
  ocitanjepot=analogRead(pot);   //očitaj stanje analognog ulaza
  jacinaled=map(ocitanjepot, 0, 1023, 0, 255);
                                   //skaliraj očitane vrijednosti na 0-255
  analogWrite(5, jacinaled);     //postavi jačinu LED diode
  digitalWrite(4, LOW);
  digitalWrite(3, HIGH);
  Serial.print(ocitanjepot);     //pošalji očitane vrijednosti serijskim putem
  Serial.print("\t");           //pošalji znak tabulatora serijskim putem
  Serial.println(jacinaled);    //pošalji jačinu LED serijskim putem
  delay(1000);                  //čekaj 50 ms
}
if (digitalRead(tpk2)==LOW){
  ocitanjepot=analogRead(pot);   //očitaj stanje analognog ulaza
  jacinaled=map(ocitanjepot, 0, 1023, 0, 255);
                                   //skaliraj očitane vrijednosti na 0-255
  analogWrite(5, jacinaled);     //postavi jačinu LED diode
  digitalWrite(4, HIGH);
  digitalWrite(3, LOW);
  Serial.print(ocitanjepot);     //pošalji očitane vrijednosti serijskim putem
  Serial.print("\t");           //pošalji znak tabulatora serijskim putem
  Serial.println(jacinaled);    //pošalji jačinu LED serijskim putem
  delay(1000);
}
}
}

```

5. Zadatak: Daljinsko upravljanje DC motora

U ovom zadatku potrebno je instalirati IR senzor te pomoću daljinskog upravljača kontrolirati smjer i brzinu okretanja DC motora. Potrebno je napraviti program koji će s određenim gumbima mijenjati smjer vrtnje motora i brzinu okretanja. Od gumba 2 do gumba 4 podesiti različite brzine rotiranja od minimalne do maksimalne u smjeru gibanja kazaljke na satu. Od gumba 5 do gumba 6 postupno povećavati brzinu okretanja motora, ali u drugom smjeru.

Prikaz spajanja:



Programski kod rješenja:

```

char t;

/*int tpk1=9;
int tpk2=10;
int tpk3=11;
int tpk4=12;
int stanje = LOW;*/

void setup() {
Serial.begin(9600);
/*pinMode (tpk1, INPUT_PULLUP);
pinMode (tpk2, INPUT_PULLUP);
pinMode (tpk3, INPUT_PULLUP);
pinMode (tpk4, INPUT_PULLUP);
pinMode (5,OUTPUT);
pinMode (4,OUTPUT);
pinMode (3,OUTPUT);*/
}
void loop() {

if(Serial.available()){
t = Serial.read();
Serial.println(t);
}
if (t == '1'){

analogWrite(5, 255);
digitalWrite(4, LOW);
digitalWrite(3, HIGH);
}

if (t == '2'){
analogWrite(5, 255);
digitalWrite(4, HIGH);
digitalWrite(3, LOW);
}

if (t == '3'){
analogWrite(5, 205);
digitalWrite(4, LOW);
digitalWrite(3, HIGH);
}
if (t == '4'){

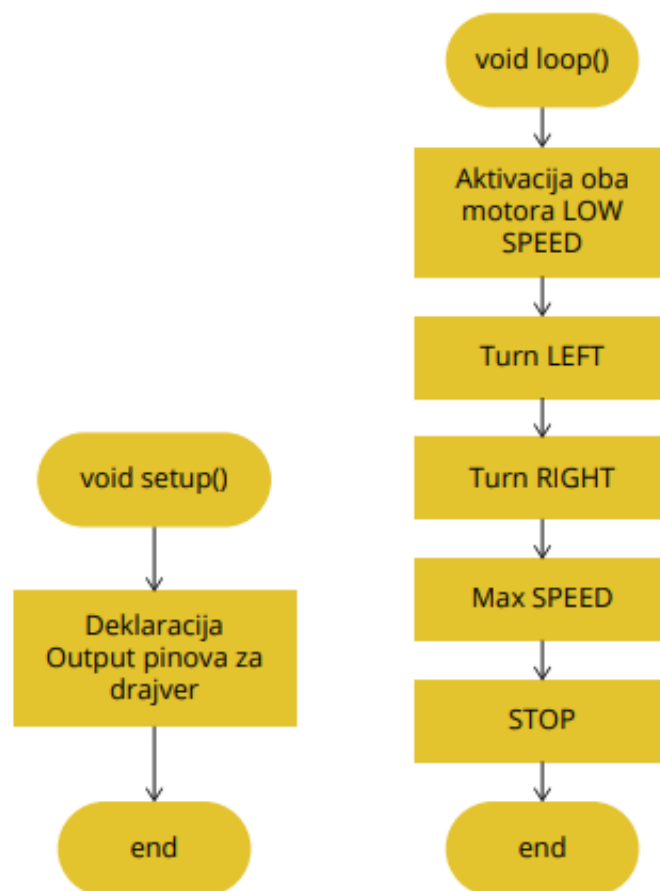
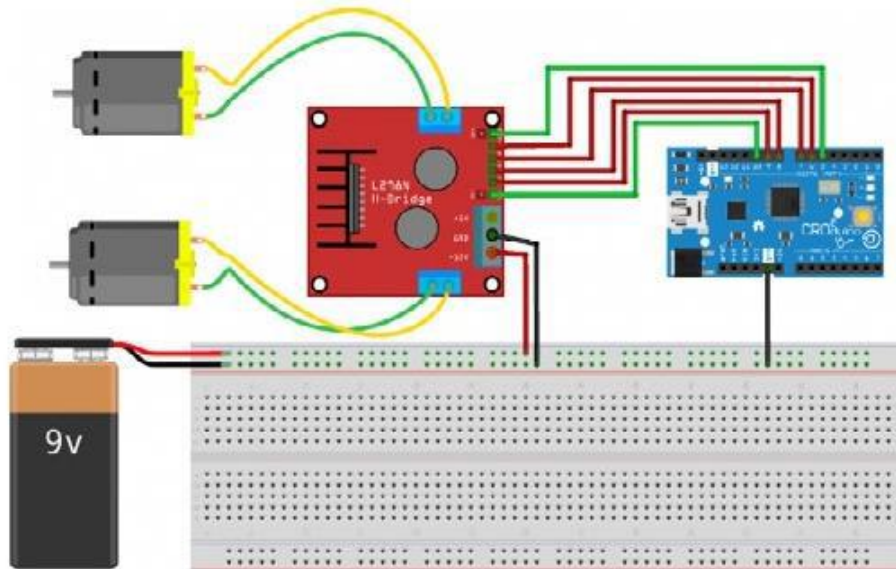
analogWrite(5, 205);
digitalWrite(4, HIGH);
digitalWrite(3, LOW);
}
if (t == '5'){

analogWrite(5, 235);
digitalWrite(4, LOW);
digitalWrite(3, HIGH);
}
if (t == '6'){ //ako je pritisnuto tipkalo
analogWrite(5, 255);
digitalWrite(4, LOW); // mijenja se smjer okretanja motora
digitalWrite(3, LOW); // mijenja se smjer okretanja motora
}
}
}

```

6. Zadatak: Pokretanje 2 DC motora

Integrirani sklop za upravljanje motorom - L298N

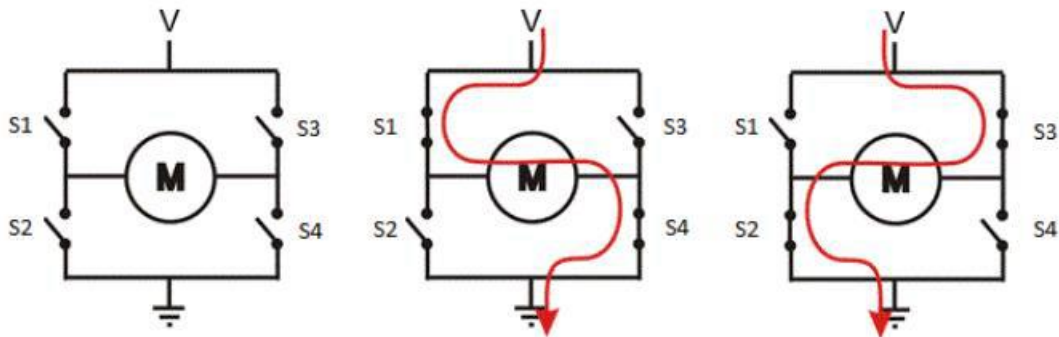


Slika 6.1 Pojednostavljena algoritamska struktura aplikacije

OUT1 – +/- Izlaz za motor A	In1 – Služi za kontrolu smjera okretaja motora A
OUT2 – +/- Izlaz za motor A	In2 – Služi za kontrolu smjera okretaja motora A
OUT3 – +/- Izlaz za motor B	In3 – Služi za kontrolu smjera okretaja motora B
OUT4 – +/- Izlaz za motor B	In4v – Služi za kontrolu smjera okretaja motora B
CON5 – Služi za uključivanje 5V regulatora napona	+5V – Ulaz u koji ide vanjski izvor napajanja
EnA – Služi za kontrolu PWM-a za motor A	GND – Ground
EnB – Služi za kontrolu PWM-a za motor B	+12V – Ulaz vanjskog izvora napajanja za motore



Da bismo mogli napisati kod za neki od svojih budućih projekata koji bi koristili ovaj drajver, trebalo bi opisati princip rada drajvera L298N i kao prva stvar koju treba spomenuti je da on koristi takozvani integrirani čip H-most (engl. H-bridge). H-most je strujni krug koji se sastoji od četiri prekidača spojenih s motorom na način prikazan na slici ispod. Ako S2 i S3 prekidače zatvorimo, a S1 i S4 prekidače otvorimo, struja će teći u određenome smjeru. Ako zamijenimo stanja prekidača tako da su S2 i S3 otvoreni, a S1 i S4 zatvoreni, struja će kroz motor proteći u suprotnome smjeru. Uz to ne smijemo zatvoriti sva četiri prekidača odjednom ili oba prekidača na jednoj strani H-mosta (npr. S1 i S2) jer time izazivamo kratki spoj.



Slika pr. 6.4. Princip rada drajvera L298

Tablica ispod prikazuje sve moguće kombinacije stanja prekidača i njihov ishod.

S1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
S2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
S3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
S4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Rezultat	Nema	Nema	Nema	Kratki	Nema	Režim	Motor se	Kratki	Nema	Motor se	Režim	Kratki	Kratki	Kratki	Kratki	Kratki
	promjene	promjene	promjene	spoj	promjene	kočenja	okreće u	spoj	promjene	okreće u	kočenja	spoj	spoj	spoj	spoj	spoj
							lijevu stranu			desnu stranu						

S pomoću H-mosta vrlo je jednostavno zamijeniti polaritet na trošilu. Najčešće se koristi za zamjenu smjera vrtnje DC motora. H-most se koristi i u raznim drugim aplikacijama poput DC/ AC, AC/AC ili DC/DC konvertera. Iako je namijenjen za motore, može se koristiti i za druge uređaje koji za određenu

potrebu zahtijevaju zamjenu polariteta na svojim ulaznim krajevima. Uz mogućnost zamjene polariteta drajver L298N ima još mogućnost kontrole brzine okretaja motora koristeći signale PWM (engl. Puls-Width Modulation). Ovaj put za jednostavnu aplikaciju nećemo koristiti knjižnicu, ali je ovo idealan hardver za koji bismo na predmetu Programiranje mogli napisati jednostavan „drajver“. Za sada je dovoljno kontrolom stanja izlaza s pomoću funkcije digitalWrite() pokrenuti motore. Za brzinu okretaja motora koristi se funkcija analogWrite().

```
// deklaracija pinova za prvi motor
int enA = 10;
int in1 = 9;
int in2 = 8;
// deklaracija pinova za drugi motor
int enB = 5;
int in3 = 7;
int in4 = 6;

void setup()
{
  // Postavi GPIO kao izlaze
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
}

void loop()
{
  // aktiviraj prvi motor
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  // postavi brzinu okretaja prvog motora 200 (maks. 255)
  analogWrite(enA, 200);
  // aktiviraj drugi motor
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
  // postavi brzinu okretaja drugog motora 200 (maks. 255)
  analogWrite(enB, 200);
  delay(2000);

  // Zamjeni smjerove okretaja motora
  digitalWrite(in1, LOW);
  digitalWrite(in2, HIGH);
  digitalWrite(in3, LOW);
  digitalWrite(in4, HIGH);
  delay(2000);

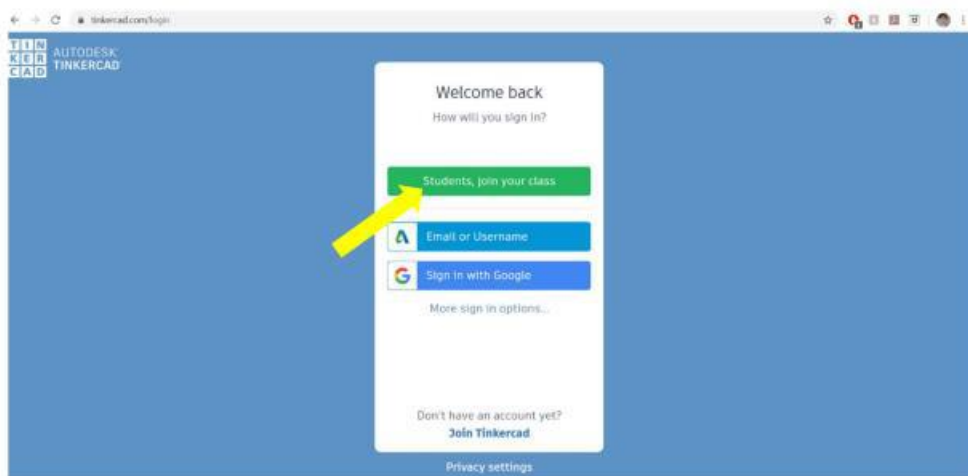
  // iskljuci oba motora
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
  delay(2000);
}
```


ISHOD UČENJA	RAZRADA ISHODA – INDIKATORI
Učenik/-ica: <ul style="list-style-type: none"> - opisuje ulogu elektronskih komponenti koje su korištene u projektu 	Učenik/-ica: <ul style="list-style-type: none"> - objašnjava ulogu drajvera motora - opisuje vezu između drajvera L298 i DC motora - demonstrira ulogu H-bridge kroz shematski prikaz - objašnjava princip rada drajvera L298
<ul style="list-style-type: none"> - koristi električnu shemu spoja za povezivanje Arduina i elektronskih komponenti 	<ul style="list-style-type: none"> - povezuje drajver L298 i DC motor - prema shemi spoja
<ul style="list-style-type: none"> - kreira algoritamsku strukturu spoja 	<ul style="list-style-type: none"> - procjenjuje prednosti i ograničenja algoritamskog pristupa u rješavanju problema - vizualizira projektni zadatak kroz algoritamsku shemu
<ul style="list-style-type: none"> - kreira programski kod za drajver L298 i DC motor 	<ul style="list-style-type: none"> - programira Arduino deklarirajući prvo pinove motora - rekonstruira programski kod prema vlastitim idejama
<ul style="list-style-type: none"> - verificira i izvršava program 	<ul style="list-style-type: none"> - analizira programski kod - otklanja pogreške ako su nastale u kodu prema smjernicama nastavnika/-ice
<ul style="list-style-type: none"> - testira projektni zadatak 	<ul style="list-style-type: none"> - učitava kod na platformu Arduino - analizira projektni zadatak - nadograđuje programski kod vlastitim idejama

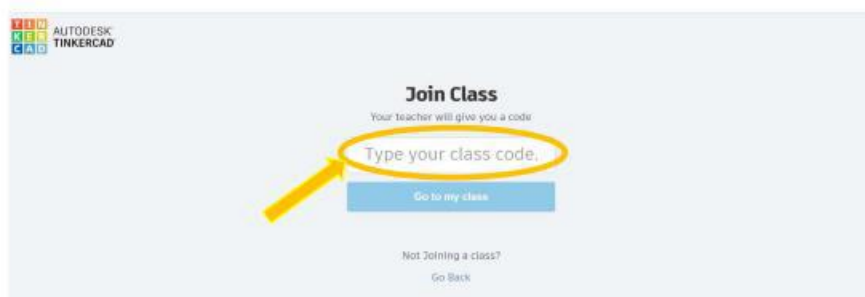
Dakle program se zove TinkerCAD i odete na stranicu <https://www.tinkercad.com/> a zatim na Sign in:



Nakon toga dobit ćete ovo i vi kliknete ovo zeleno **Students, join your class**



Sada će vas tražiti Class code:

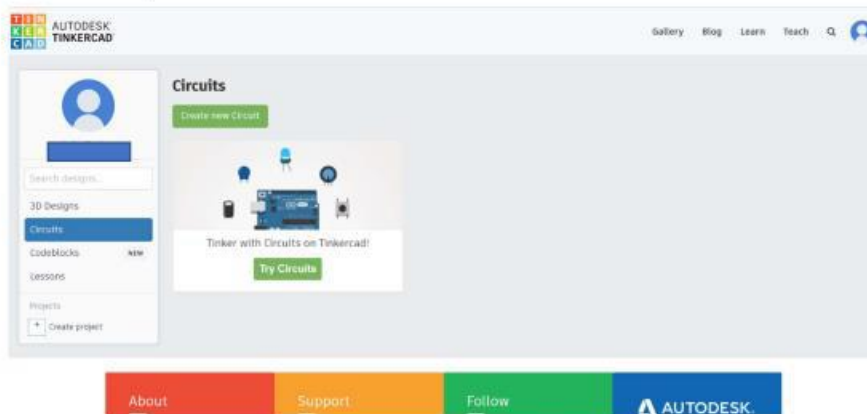


Evo ga, dakle vi upisujete ovo u zelenom (ovo sa slovima i brojevima od 12 znamenki):

I kad ste se uspešno logirali u naš razred dobit ćete ovo:



I sada smo tu, spremni na Arduino izazove!



Nakon što odete na **Create new Circuit**, pogledajte upute na poveznici!

Literatura:

- [1] <https://www.arduino.cc/>
- [2] <https://www.wikipedia.org/>
- [3] <https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/>
- [4] <https://components101.com/microcontrollers/arduino-uno>
- [5] <https://components101.com/wireless/hc-05-bluetooth-module>
- [6] <https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-1298n-pwm-h-bridge/>