

Sveučilište u Zagrebu
Fakultet strojarstva i brodogradnje

Seminarski rad iz kolegija
Mikroprocesorsko upravljanje

Daljinski upravljani ventilator

17.6.2024.

Zadatak zadao:

Prof. dr. sc. Mladen Crneković

Student:

Lovro Šibenik

0035239806

Sadržaj

1. Uvod.....	3.
2. Opis problema	4.
3. Prijedlog rješenja.....	5.
4. Električne komponente.....	6.
5. Arduino kod.....	11.
6. Rezultati i zaključak.....	16.
7. Električna shema.....	17.
8. Literatura.....	18.

1. Uvod

U sklopu kolegija Mikroprocesorsko upravljanje, kao temu seminarskog zadatka odlučio sam korištenjem pojedinih električnih komponenti i Arduino sučelja izraditi mali ventilator koji će biti upravljan pomoću daljinskog upravljača. Projekt uključuje mnoštvo komponenti (Arduino Uno mikrokontroler (slika 1.), breadbord (slika 2.), dc (slika 4.) i servo motor (slika 3.), IR reciever i controler (slika 7.), H-bridge (slika 5.), power suply module (slika 6.), 9V baterija i mnoštvo žica) čijim je povezivanjem, zajedno sa kodom, bilo potrebno projektirati daljinski upravljan ventilator koji bi se mogao rotirati i postaviti između 0° i 180° kao i pojačati odnosno smanjiti (snagu puhanja).

Glavni zadatak samog projekta je poboljšavanje već postojećeg proizvoda, odnosno ventilatora, tako da osoba ne mora ustajati kako bi pokrenula odnosno namjestila puhanje ventilatora, već to može napraviti pomoću daljinskog upravljača.

Kombinacija gore navedenih komponenti i Arduino koda koji će biti kasnije prikazan u ovom seminaru, daje izvrsno tehnološko rješenje za izradu prototipova daljinski upravljanog ventilatora bez potrebe za skupom i kompliciranom opremom.

2. Opis problema

Svi smo dobro upoznati s pojavom nesnosnih vrućina koje se u našim krajevima pojavljuju zajedno s dolaskom ljeta i dobro znamo kako to može biti nepodnošljivo. Jedno od rješenja za takve situacije su klime, ali one znaju biti izuzetno skupe i ne može si ih svatko priuštiti. Nadalje klime se postavljaju na neko mjesto u kući/stanu i tamo moraju biti. Puno bolje i pristupačnije rješenje su ventilatori. Ventilatori su uređaji koji služe za cirkulaciju zraka, hlađenje prostora i smanjenje temperature u unutrašnjim i vanjskim prostorima. Koriste se u domovima, uredima, industrijskim postrojenjima i mnogim drugim mjestima gdje je potrebna ventilacija ili dodatno hlađenje. Inspiriran ovom potrebom, osmislio sam ideju daljinski upravljano ventilatora za rashlađivanje. Ovaj ventilator omogućava korisniku podešavanje brzine puhanja i smjer u kojem zrak struji, kao i uključivanje i isključivanje uređaja, sve putem daljinskog upravljača.

Najveći problem ove ideje je povezanost i istovremeno upravljanje pojedinih komponenti koje zajedno čine isti proizvod, a opet svaka od njih obavlja svoj zasebni zadatak, neovisan o drugim komponentama.

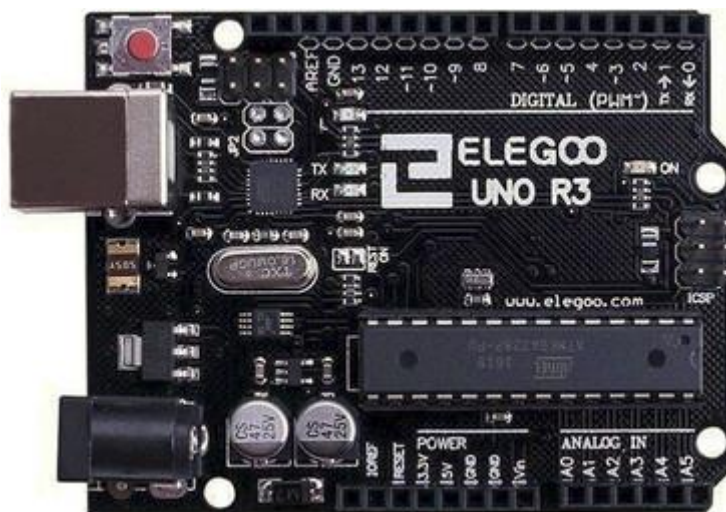
3. Prijedlog rješenja

Problem sam odlučio riješiti korištenjem Arduino Uno programskog jezika i električnih komponenti čijim sam sklapanjem i međusobnim povezivanjem mogao ostvariti zadani cilj.

Sami zadatak je imao tri osnovna problema. Prvi problem je bio uspostava beskontaktna komunikacije s mikrokontrolerom (slika 1.) što sam htio riješiti pomoću IR (infra red) receiver modula i Remote controller-a (slika 7.). Nakon tog problema, idući na redu bio je problem povezanosti motora i kontrolera. Naime glavna ideja je bila slanjem određenog signala sa Remote controller-a na mikrokontroler ostvariti nekakvu promjenu stanja pojedinih motora. Motori korišteni u ovom projektu su Servo motor (slika 4.) SG90 (za rotaciju stupa samog ventilatora) i DC 3-6V (slika 3.) motor (za samu rotaciju ventilatora i ostvarivanje puhanja. Iz razloga što su motori različiti, različito je bilo i njihovo povezivanje sa mikrokontrolerom (slika 1.), a na kraju i sami kod za odrađivanje njihovih funkcija. Prilikom implementacije i korištenja DC motora javio se problem s njegovim napajanje. Naime, proučavanjem korištenja istosmjernog motora uočio sam da vrlo lako može oštetiti pin-ove mikrokontrolera te se iz tog razloga moraju koristiti dodatne komponente za zaštitu poput H-mosta L293D (slika 5.) i power-supply module-a (slika 6.) kako bi se samom motoru poslao dovoljan napon.

4. Električne komponente

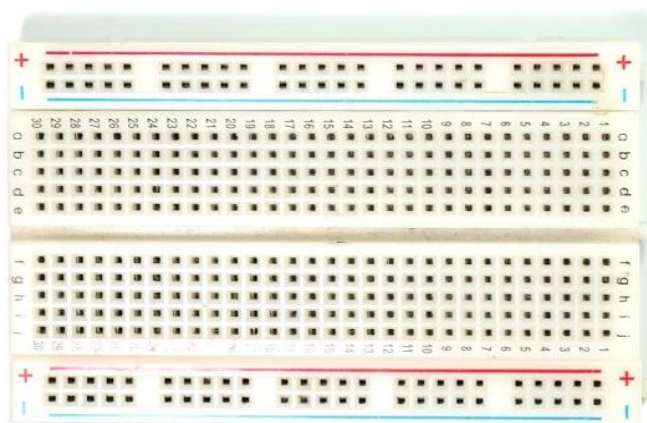
- **Arduino Uno Mikrokontroler**



Slika 1. Arduino Uno

Arduino Uno mikrokontroler koristi se za razvoj elektroničkih projekata, automatizaciju i učenje programiranja i elektronike. Ima 14 digitalnih I/O pinova (od kojih 6 mogu služiti kao PWM izlazi), 6 analognih ulaza, 32 KB flash memorije, 2 KB SRAM-a i radi na radnom naponu od 5V, dok se napaja putem USB priključka ili vanjskog izvora napona između 7-12V. Struja po I/O pinu je maksimalno 40 mA, a ukupno kroz Vcc i GND pinove može teći do 500 mA.

- **Breadboard**



Slika 2. breadboard

Breadboard je alat za izradu i testiranje prototipa elektroničkih krugova bez lemljenja, s mrežom rupa povezanih vodljivim tračnicama za jednostavno spajanje komponenti. Tipični breadboard ima dva naponska vodoravna reda za napajanje (pozitivni i negativni), te nekoliko vertikalnih kolona povezanih po grupama od po pet rupa, s razmakom od 2.54 mm između rupa.

- **Servo motor**



Slika 3. servo motor

Micro servo motor je mali električni motor s integriranim kontrolerom koji omogućuje precizno upravljanje kutom zakreta u rasponu od obično 0 do 180 stupnjeva. Tipično radi na naponu od 4.8V do 6V, pružajući okretni moment od oko 1.5 do 2.5 kg/cm, i koristi PWM signal za upravljanje pozicijom. Ima tri žice za povezivanje: crvenu za napajanje, crnu ili smeđu za uzemljenje, i žutu, narančastu ili bijelu za kontrolni signal.

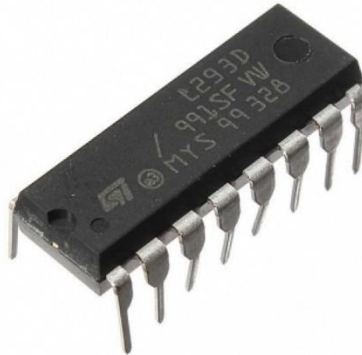
- **DC motor**



Slika 4. DC motor

DC motor je električni motor koji pretvara istosmjernu električnu energiju u mehaničku rotaciju. Radi na različitim naponskim razinama, najčešće između 3V i 12V, i nudi brzinu vrtnje koja varira ovisno o primijenjenom naponu i opterećenju, obično od nekoliko stotina do nekoliko tisuća okretaja u minuti (RPM). Kada se koristi s Arduinoom, brzina DC motora može se precizno kontrolirati putem PWM signala; što je veća vrijednost PWM signala, to će se motor brže vrtjeti. DC motor ima dvije žice za povezivanje, jednu za pozitivni napon i jednu za uzemljenje, a smjer rotacije može se promijeniti promjenom polariteta napajanja.

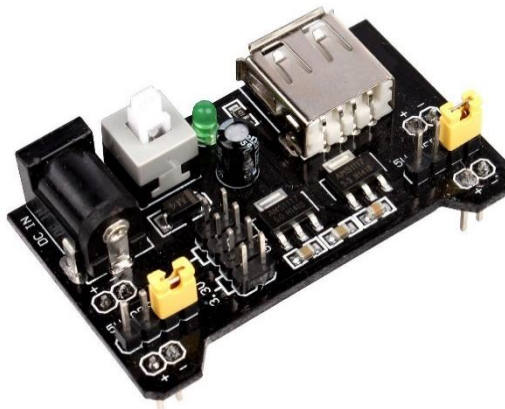
- **L293D**



Slika 5. H-most

L293D je H-mostni integrirani sklop koji omogućuje upravljanje smjerom i brzinom rotacije DC motora pomoću dva logička signala, pružajući struju do 600 mA po kanalu uz maksimalni napon od 36V. Sadrži četiri kanala koji omogućuju upravljanje s dva motora istovremeno, podržavajući funkcionalnosti kao što su naprijed, natrag, kočenje i slobodni hod.

- **Power supply module**



Slika 6. Power supply module

Power supply module je komponenta koja se koristi za jednostavno napajanje breadboarda i ostalih elektroničkih komponenti. Pri spajanju na breadboard, omogućuje napon od 3.3V i 5V na naponskim linijama breadboarda, obično pomoću standardnog ulaza za DC adapter ili USB priključka. Moduli obično podržavaju ulazni napon u rasponu od 6.5V do 12V, te pružaju struju do nekoliko stotina miliampera za napajanje različitih projekata.

- **IR remote controler and reciever**



Slika 7. IR components

IR receiver je uređaj koji prima infracrvene signale emitirane od strane IR kontrolera i pretvara ih u električne signale koje mikrokontroler može obraditi. Tipično radi na naponskom rasponu od 2.7V do 5.5V i osjetljiv je na frekvenciju od oko 38 kHz. IR kontroler emitira kodirane infracrvene signale pritiskom na tipke koje IR receiver dekodira i šalje dalje mikrokontroleru za izvršavanje odgovarajućih naredbi, omogućujući bežičnu komunikaciju i kontrolu uređaja.

5. Arduino kod

1. Dobivanje adrese gumba

```
if (IrReceiver.decode()) // kad je gumb pritisnut i kod primljen
{
    String ir_code = String(IrReceiver.decodedIRData.command, HEX);
    if (ir_code != "0") {
        Serial.println(ir_code);
        delay(250);
    }
}
```

Ovaj kod bio je preteča za nastavak rada jer smo morali doznati adrese pojedinih gumba na IR remote controller-u kako bismo ih kasnije mogli koristiti za pokretanje određenih funkcija. Ovaj kod ne nalazi se u konačnom kodu koji se koristi za pokretanje i upravljanje ventilatorom.

2. Glavni kod

```
#include <IRremote.hpp>
#include <Servo.h>

#define ENABLE 5
#define DIRA 3
#define DIRB 4

int i=0;

const byte IR_RECEIVE_PIN = 2;

Servo myservo;

int motor_angle = 90;

bool continuousMode = false;

unsigned long lastUpdateTime = 0;

int increment = 1;

void setup()
{

```

```

pinMode(ENABLE,OUTPUT);

pinMode(DIRA,OUTPUT);

pinMode(DIRB,OUTPUT);

Serial.begin(115200);

IrReceiver.begin(IR_RECEIVE_PIN, ENABLE_LED_FEEDBACK);

myservo.attach(9);

myservo.write(motor_angle);

delay(2000);
}

void loop()
{
  if (IrReceiver.decode()) {
    String ir_code = String(IrReceiver.decodedIRData.command, HEX);
    if (ir_code == "16") {
      Serial.println("servo motor kut 0 stupnjeva");
      motor_angle = 0;
      myservo.write(motor_angle);
      continuousMode = false;
    }
    else if (ir_code == "d") {
      Serial.println("servo motor kut 180 stupnjeva");
      motor_angle = 180;
      myservo.write(motor_angle);
      continuousMode = false;
    }
    else if (ir_code == "19") {
      Serial.println("servo motor kut 90 stupnjeva");
      motor_angle = 90;
      myservo.write(motor_angle);
      continuousMode = false;
    }
  }
}

```

```

else if (ir_code == "43") {
    if (motor_angle < 180) motor_angle += 5;
    myservo.write(motor_angle);
    continuousMode = false;
}
else if (ir_code == "44") {
    if (motor_angle > 0) motor_angle -= 5;
    myservo.write(motor_angle);
    continuousMode = false;
}

else if (ir_code == "47") {
    Serial.println("continuous mode - on");
    continuousMode = true;
    increment = (motor_angle < 90) ? -1 : 1;
}
else if (ir_code == "c") {
    Serial.println("speed mode - min");
    analogWrite(ENABLE, 175);
    digitalWrite(DIRA, HIGH);
    digitalWrite(DIRB, LOW);
    delay(250);
}
else if (ir_code == "18") {
    Serial.println("speed mode - mid");
    analogWrite(ENABLE, 220);
    digitalWrite(DIRA, HIGH);
    digitalWrite(DIRB, LOW);
    delay(250);
}
else if (ir_code == "5e") {
    Serial.println("speed mode - max");

```

```

    analogWrite(ENABLE,255);
    digitalWrite(DIRA,HIGH);
    digitalWrite(DIRB,LOW);
    delay(250);
}
else if (ir_code == "45") {
    Serial.println("fan - off");
    analogWrite(ENABLE,0);
    digitalWrite(DIRA,HIGH);
    digitalWrite(DIRB,LOW);
    delay(250);
}
IrReceiver.resume();    }
if (continuousMode && millis() - lastUpdateTime >= 15) {
    lastUpdateTime = millis();
    motor_angle += increment;
    if (motor_angle >= 180) {
        motor_angle = 180;
        increment = -1;
    }
    if (motor_angle <= 0) {
        motor_angle = 0;
        increment = 1;
    }
    myservo.write(motor_angle);
}
}

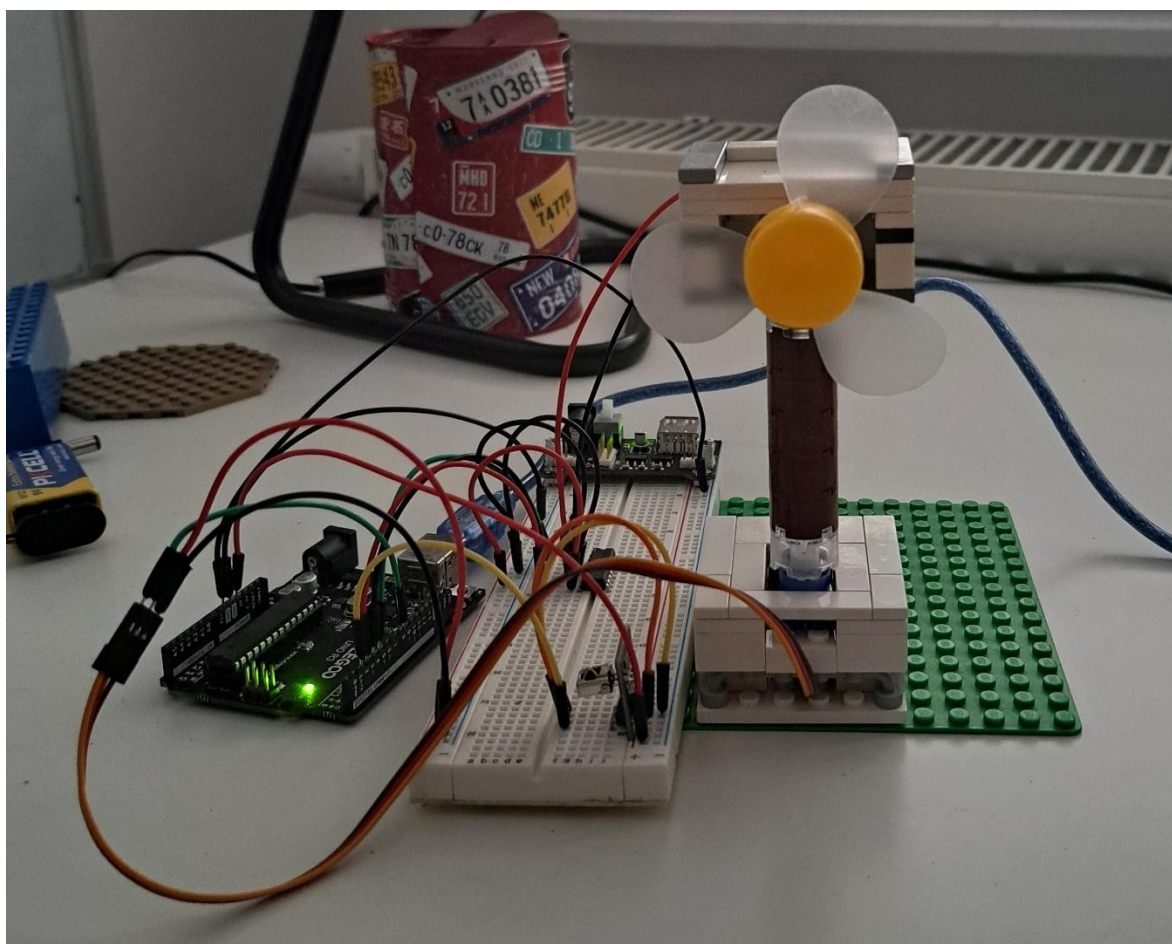
```

Objašnjenje koda

Na početku se pomoću naredbe `'#include'` uključuju sve potrebne biblioteke za korištenje određenih naredbi za IR komunikaciju i servo motor. Pomoću `'#define'` definiramo pinove za kontrolu DC motora te poslije toga u kodu dolazi dio gdje se deklariraju varijable koje će se kasnije koristiti u funkcijama. U `void setup()` su određeni pinovi deklarirani kao output-i i input-i, ovisno o njihovim zadaćama, zadaje se serijska komunikacija pomoću funkcije `Serial.begin(115200)` i uključuju se objekti klase `myservo`. U `void loop()` prvo se nalazi if funkcija koja očitava signale poslone sa IR receiver-a te se deklariraju kao HEX (heksadecimalne vrijednosti) koje će služiti kao adrese gumba koje će se pozivati kada će se pojedina funkcija obavljati. Svaki dio if petlje je generalno isti, kaže da ako se očita adresa, da se servo motor postavi na vrijednost koju smo mu definirali za taj pritisnuti gumb. Nešto drugačija je logika za pomicanje ručno, finije i preciznije okretanje ventilatora gdje se kaže da se u odnosu na staru vrijednost kuta doda ili oduzme (ovisno o pritisnutom gumbu) 5° , a ako dođe do granice (0° i 180°) da ostane na toj vrijednosti. Najkompliciraniji dio sa servo motorom je bila ideja kontinuirane rotacije između 0° i 180° , a to je riješeno tako da smo zadali motoru da ako je trenutni kut manji od 90° da se rotira prema 0° , a ako je veći, odnosno bliži 180° , da ide prema njemu. Nakon toga dolazi podešavanje DC motora i njegove brzine rotacije. Sam DC motor radi na principu PWM signala, odnosno što je jači signal to će se on brže rotirati. Kao što u kodu piše, najbrža rotacija je kada dobije 255 (najveću moguću vrijednost), a za manje ulazne podatke se vrti sporije. Zadnja funkcija u toj petlji je `IrReceiver.resume()` koja služi da se nakon primanja signala IR receiver pripremi za novi signal. Na kraju koda nalazi se nužna matematika kako bi sve gore navedene funkcije mogle biti obavljene.

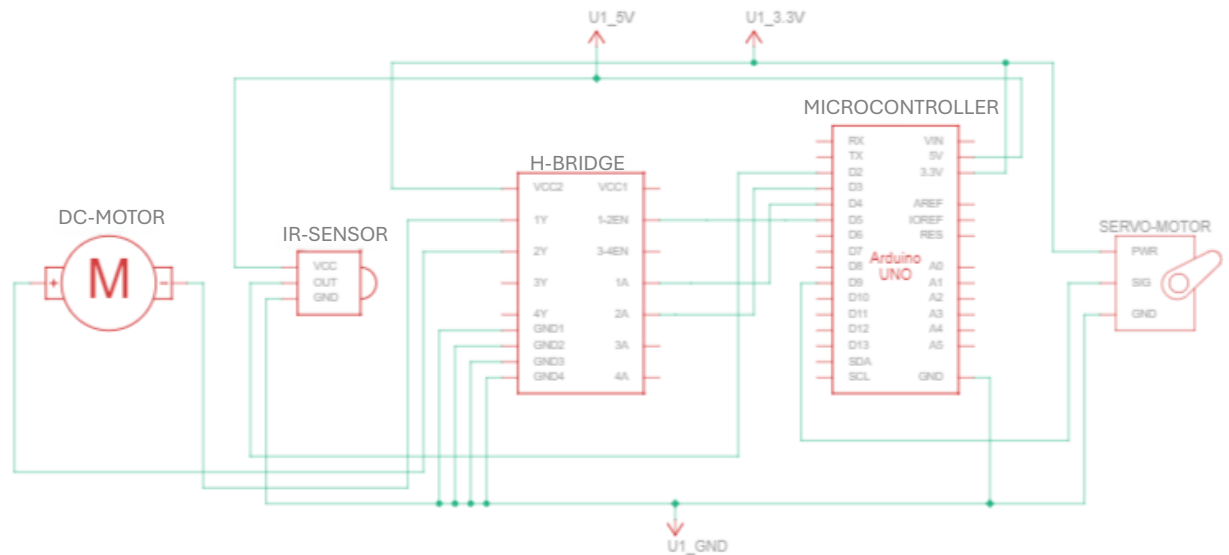
6. Rezultati

Ideja seminarskog zadatka provedena je u djelo te je svrha zadatka ispunjena. Naravno, rješenje nije idealno i savršeno, korištene su poprilično jednostavne i neprofesionalne komponente, odnosno, da su se koristile skuplje i bolje komponente, sam proizvod bio bi puno bolji. Ovaj rad/zadatak jako je dobra početna ideja koja se može razviti u prvi proizvod koji bi mogao biti konkurentan na tržištu.



Slika 8. sklop

7. Električna shema



Slika 9. električna shema

8. Literatura

DC motor

<https://docs.arduino.cc/tutorials/motor-shield-rev3/msr3-controlling-dc-motor/>

<https://www.youtube.com/watch?v=YU17L650k3s>

Servo motor

<https://docs.arduino.cc/learn/electronics/servo-motors/>

https://www.youtube.com/watch?v=qJC1nt_eJZs

IR communication

<https://www.electronicwings.com/arduino/ir-communication-using-arduino-uno>

<https://www.makerguides.com/control-servo-motor-with-ir-remote/>