

Demonstracijska maketa V8 ICE motora upravljana mikroračunalnim sustavom

Kurtović, David

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:202774>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-03-27**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH
TEHNOLOGIJA

Stručni studij Automatika

DEMONSTRACIJSKA MAKETA V8 ICE MOTORA
UPRAVLJANA MIKRORAČUNALNIM SUSTAVOM

Završni rad

David Kurtović

Osijek, 2024.

Obrazac Z1S: Obrazac za ocjenu završnog rada na stručnom prijediplomskom studiju

Ocjena završnog rada na stručnom prijediplomskom studiju

Ime i prezime pristupnika:	David Kurtović
Studij, smjer:	Stručni prijediplomski studij Elektrotehnika, smjer Automatika
Mat. br. pristupnika, god.	A 4557, 19.07.2019.
JMBAG:	0165080771
Mentor:	izv. prof. dr. sc. Tomislav Keser
Sumentor:	prof. dr. sc. Damir Blažević
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	izv. prof. dr. sc. Alfonzo Baumgartner
Član Povjerenstva 1:	izv. prof. dr. sc. Tomislav Keser
Član Povjerenstva 2:	doc. dr. sc. Tomislav Galba
Naslov završnog rada:	%naziv_rada%
Znanstvena grana završnog rada:	Procesno računarstvo (zn. polje računarstvo)
Zadatak završnog rada:	Dizajnirati, izraditi i testirati maketu V8 motora s unutarnjim izgaranjem za edukativne potrebe. Maketu izraditi na način da je ista modelski sukladna pravom V8 motoru, na kojoj se mogu vidjeti svi pokretni dijelovi ali i upravljati brzinom vrtnje. Sustav pokretanja pomičnih dijelova makete riješiti uporabom dodatnog vanjskog elektromotora koji će biti upravljan putem odabrane mikroračunalne platforme. Brzinu vrtnje motora omogućiti lokalno putem upravljačkog panela, tipaka i potencijometra te daljinski, putem mobilne aplikacije za Android sustav.
Datum ocjene pismenog dijela završnog rada od strane mentora:	05.09.2024.
Ocjena pismenog dijela završnog rada od strane mentora:	Izvrstan (5)
Datum obrane završnog rada:	12.09.2024.
Ocjena usmenog dijela završnog rada (obrane):	Izvrstan (5)
Ukupna ocjena završnog rada:	Izvrstan (5)
Datum potvrde mentora o predaji konačne verzije završnog rada čime je pristupnik završio stručni prijediplomski studij:	12.09.2024.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****IZJAVA O IZVORNOSTI RADA**

Osijek, 12.09.2024.

Ime i prezime Pristupnika:

David Kurtović

Studij:

Stručni prijediplomski studij Elektrotehnika, smjer Automatika

Mat. br. Pristupnika, godina upisa:

A 4557, 19.07.2019.

Turnitin podudaranje [%]:

11

Ovom izjavom izjavljujem da je rad pod nazivom: **Demonstracijska maketa V8 ICE motora upravljana mikroračunalnim sustavom**

izrađen pod vodstvom mentora izv. prof. dr. sc. Tomislav Keser

i sumentora prof. dr. sc. Damir Blažević

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.

Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis pristupnika:

SADRŽAJ

1. UVOD.....	1
1.1. Zadatak i struktura rada.....	4
2. SUSTAV V8 MOTORA SA UNUTARNJIM IZGARANJEM	5
2.1. Teorijski osvrt na rad V8 motora	6
2.2. Prijedlog sklopovskog rješenja	8
2.3. Prijedlog programskog rješenja	10
3. REALIZACIJA MAKETE V8 MOTORA	12
3.1. Korišteni alati i programska okruženja	12
3.2. Realizacija sklopovskog rješenja	12
3.3. Realizacija programskog rješenja	19
4. TESTIRANJE I REZULTATI	21
4.1. Metodologija testiranja	21
4.2. Rezultati testiranja.....	22
4.3. Prikaz makete u funkciji.....	24
5. ZAKLJUČAK	28
LITERATURA.....	29
PRILOZI I DODACI	30
Primijenjeni hardver u radu	30
ESP32 mikroupravljač.....	30
L298N motor driver.....	31
Programski kod projekta	32
SAŽETAK	41
ABSTRACT	41
ŽIVOTOPIS	42

1. UVOD

Kako bi pojasnili rad demonstracijske makete V8 motora, trebamo se prvo upoznati sa radom V8 motora u stvarnom životu. V8 je vrsta motora sa unutrašnjim izgaranjem gdje su cilindri u bloku motora u obliku slova V pa tako svaki krak motora sadrži četiri cilindra. Oblik V8 motora se najčešće koristi kod motora sa velikim obujmom. Uglavnom se koriste u automobilima visokih performansa te kod teških terenaca, autobusa, kamiona itd.

Početkom 20. stoljeća, nije postojala izravna veza volumena motora i broja cilindara. No s vremenom, ideja za povećanjem brzine i snage i želja za smanjenjem troškova doveli su do uvođenja prosječnog volumena cilindara. Snaga motora je bila povezana sa brojem cilindara, tj. svaki cilindar ima određeni volumen, a snaga se uklanja iz vrijednosti volumena.

Prvi motor sa V8 konfiguracijom bio je izrađen 1914. godine, no postao je popularan tek nakon što ga je Ford uveo u svoju ponudu automobila 1932. godine. Do 1950-ih godina, V8 motori su postali sve popularniji u SAD-u, a vrhunac njihove popularnosti se dogodio 1970-ih godina prije naftne krize (u tom trenutku, većinu automobila koji su bili u prodaji u SAD-u su se pokretali sa V8 motorom).

Također, u Velikoj Britaniji se odjednom pojavio pojam V8 motora 1965. godine kada je Rover prisvojio prava na neuspješan Buickov motor (smatralo se da je mali za američko tržište), iako nije doživio dugoročni uspjeh kao u SAD-u. V8 motori su popularni u trkaćim sportovima, najviše kod američkih trkaćih sportova kao što su Indy Racing League (IRL) i NASCAR, koji za svoje automobile visokih performansi koriste isključivo V8 motore. [1]

V8 motor nije samo poznat po svojoj snazi, već i po karakterističnom zvuku koji je postao zaštitni znak mnogih kulturnih automobila. Ovaj zvuk, poznat kao "burbles" ili "rumble," proizlazi iz nepravilnog rasporeda paljenja cilindara, što stvara jedinstvenu harmoniju dubokih tonova koja se često povezuje s američkim muscle carovima. Osim toga, V8 motori su se pokazali iznimno prilagodljivima za različite primjene, od sportskih automobila do teških teretnih vozila. Njihova konfiguracija omogućuje značajnu fleksibilnost u dizajnu, omogućujući inženjerima da balansiraju između kompaktnosti i snage, što je bilo ključno za njihovu primjenu u različitim tipovima vozila.

Cilj ovog rada je pojasniti u teoriji kako se konstruira V8 motor s unutarnjim izgaranjem te dobiti uvid u izradu makete V8 motora. Kroz rad su opisane smjernice po kojima je izrađena demonstracijska maketa, koji je prijedlog prije izrade makete i kako je na kraju realizirana ideja demonstracijske makete u cijelosti.



Slika 1.1. 10-litreni atmosferski V8 ICE proizvođača Chevrolet [2]



Slika 1.2. 6.3-litreni V8 ICE proizvođača AMG [3]

1.1. Zadatak i struktura rada

Zadatak ovog projekta je izraditi maketu V8 motora s unutarnjim izgaranjem koja ima funkcionalne dijelove kao i kod motora u stvarnom životu, te izraditi sustav preko mikroupravljača kojim će se upravljati brzina vrtnje motora.

Demonstracijska maketa će sadržavati mehanički i elektronički dio. Pod mehaničke dijelove spadaju blok V8 motora, radilica, klipovi i istosmjerni motor koji služi za pokretanje radilice, a pod elektronički dio spadaju mikroupravljač i potencijometar za reguliranje brzine vrtnje istosmjernog motora.

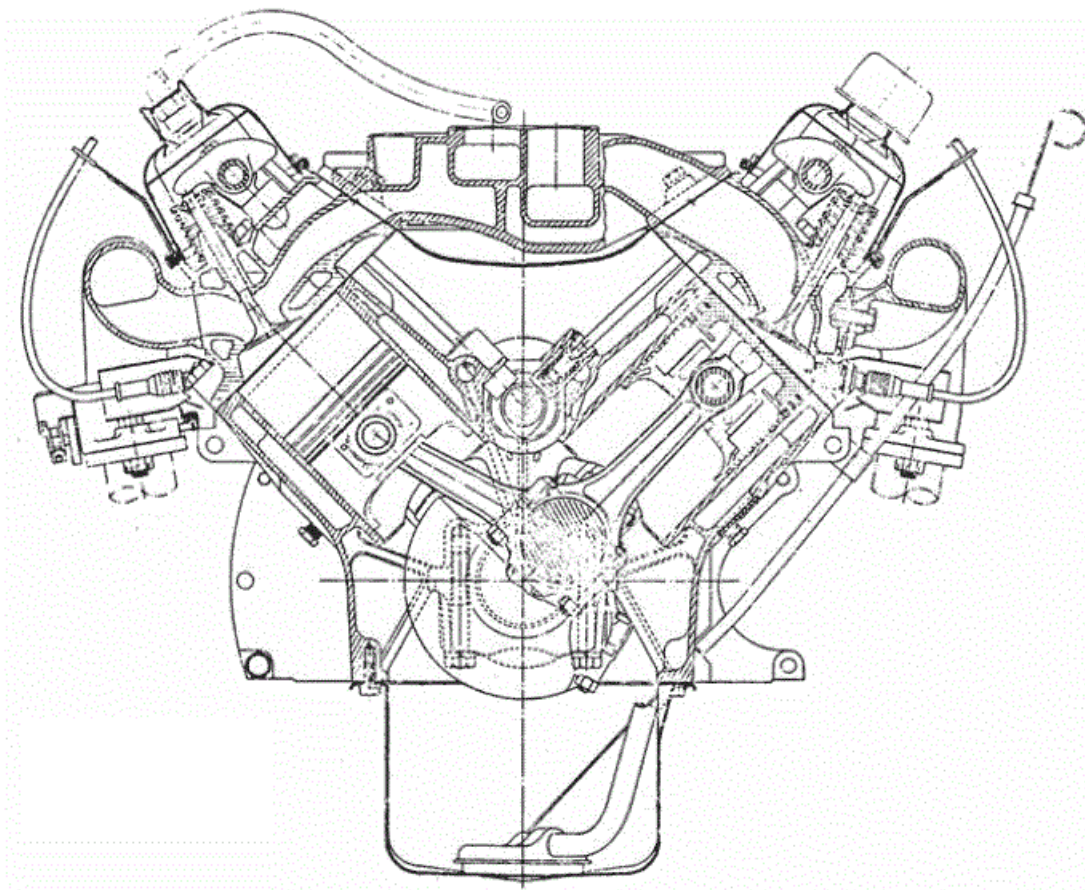
U poglavlju 2 se pojašnjava kako bi rješenje za izradu makete izgledalo u teorijskom smislu, te koji je prijedlog za sklopovsko i programsko rješenje, zajedno sa blok dijagramima i detaljno pojašnjenje za svaki blok dijagram.

U poglavlju 3 se pojašnjava koje programskog okruženje je korišteno za izradu makete, opisuje se realizacija sklopovskog i programskog rješenja za maketu. Kako bi bolje shvatili kako je realizirana maketa, rješenja su prikazana blokovskim dijagramom koji su detaljno opisani, te je slikovno prikazano kako je maketa izrađena u cijelosti.

U poglavlju 4 je prikazano na koji način je ispitana funkcija makete. Ispitivanje je prikazano tablicom sa vrijednostima određenih parametara, zajedno sa njihovom srednjom vrijednošću i mjernim odstupanjem.

2. SUSTAV V8 MOTORA SA UNUTARNJIM IZGARANJEM

V8 motor s unutarnjim izgaranjem se sastoji od osam cilindara koji su podijeljeni na dvije polovice. Sadrži dvije grupe cilindara, od kojih se svaka sastoji od četiri cilindra, stvarajući kut u obliku slova V od 90 stupnjeva. Kada je prvi cilindar s lijeve strane u gornjoj mrtvoj točki, prvi cilindar na desnoj strani je u hodu prema dolje. Stoga se stvaraju primarne sile. Međutim, sekundarne sile nisu pravilno uravnotežene i zahtijevaju „neutralizator“ trenja i nosač motora kako bi se smanjio utjecaj vibracija. Rad V8 motora je sličan svakom 4-taktnom motoru s unutarnjim izgaranjem.



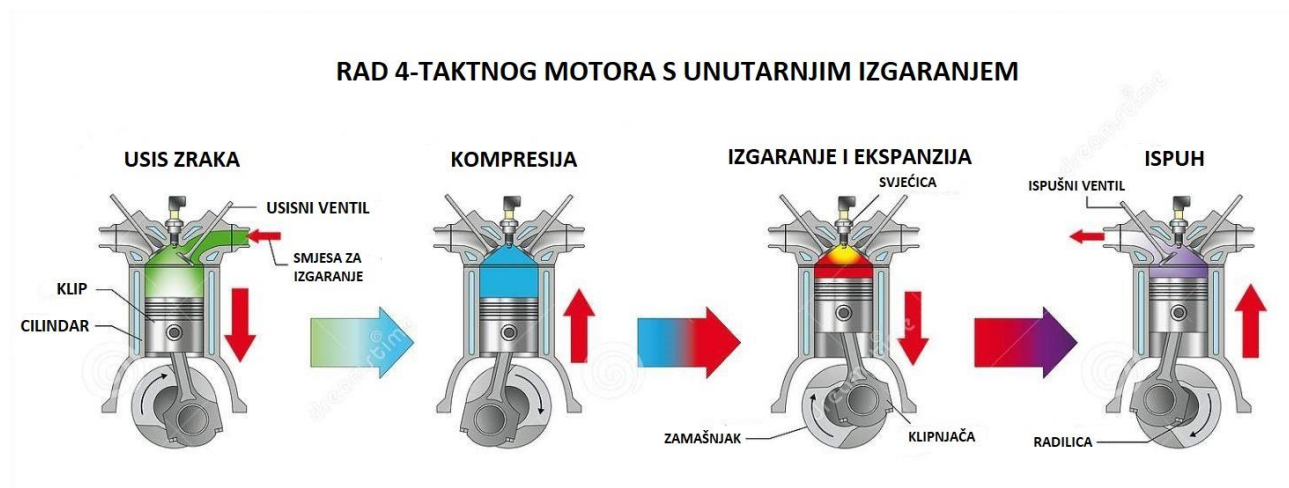
Slika 2.1. Tehnička skica V8 motora [4]

2.1. Teorijski osvrt na rad V8 motora

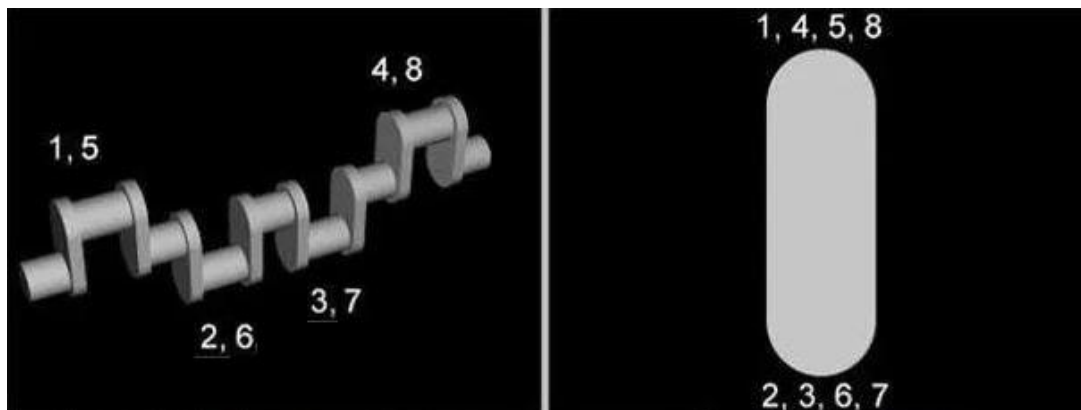
Omjer zraka i goriva u cilindru se vrši preko ventilskog mehanizma. Smjesa se komprimira i svjećice stvaraju električnu iskrnu za paljenje smjese. Snaga koja nastaje izgaranjem pomiče klipni prsten u cilindru gore-dolje. Postoje dvije različite vrste komponenti koje se koriste za otvaranje i zatvaranje ventila. Neki motori koriste bregasto vratilo s lančanim pogonom ili klackalice koje se aktiviraju potisnim polugama. Upotreba se razlikuje od proizvođača do proizvođača. Redoslijed paljenja V8 motora varira, no najčešći slijed je 1-8-7-2-6-5-4-3.

V8 ima vrlo značajnu poziciju u industriji vozila. Postoji niz prednosti zbog kojih se čak ni najnoviji motori ne mogu natjecati s klasičnim V8 motorom. Neki od prednosti V8 motora su:

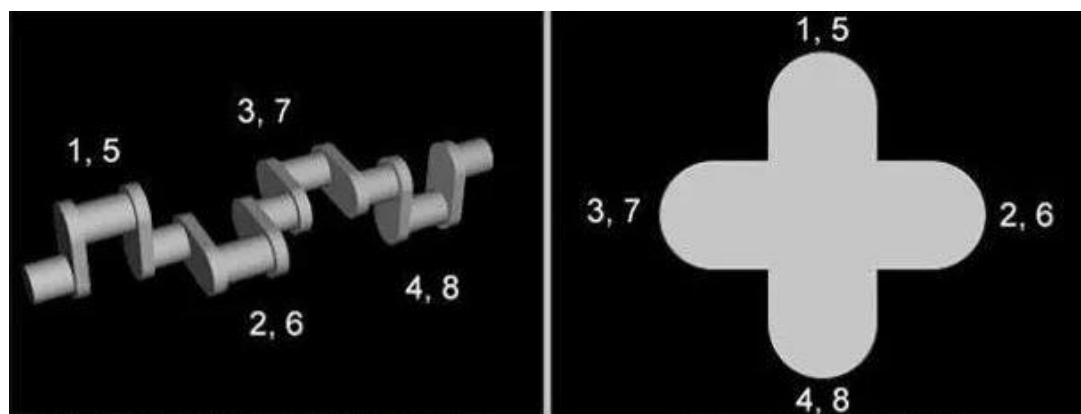
- Kompaktna veličina – konfiguracija motora s dva cilindra na dvije strane čini ga manjim te je kompaktne veličine. Uštedi se na prostoru i može se izraditi manji blok motora
- Primarne sile – dva suprotna klipa se kreću naizmjenično što proizvodi uravnotežene primarne sile
- Troškovi proizvodnje – glave sa dva cilindra znače manje pokretnih dijelova, što smanjuje ukupne troškove proizvodnje motora
- Održavanje – V8 motor je jednostavan za popravak i zahtijeva niske troškove održavanja. Glava cilindra se nalazi na vrhu, što smanjuje napor pri dostizanju glavnih komponenti motora
- Usisni razvodnik – omogućuje usisnu granu koja osigurava jednaku raspodjelu omjera zraka i goriva među svim cilindrima [5]



Slika 2.2. Prikaz 4-taktnog ciklusa u cilindru [6]

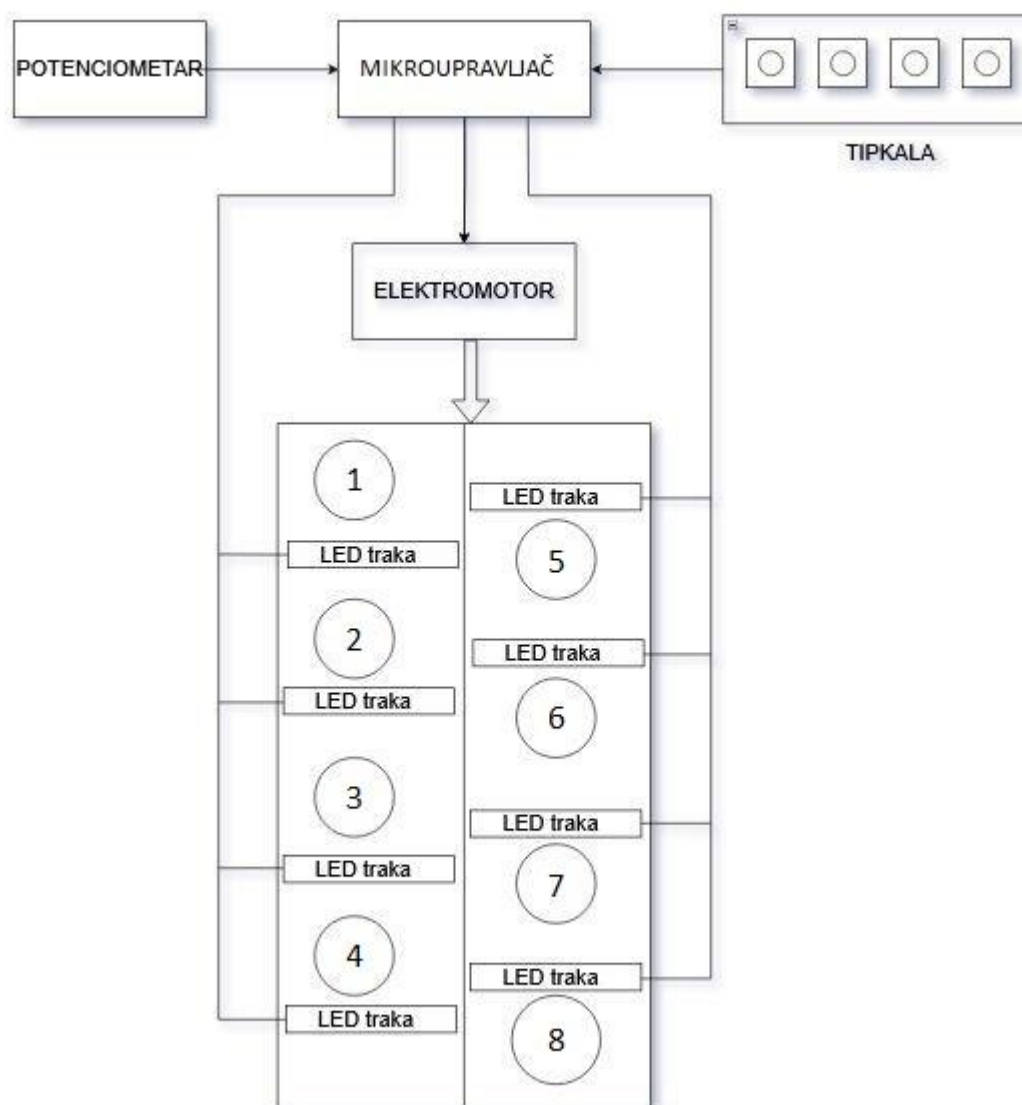


Slika 2.3. Prikaz radilice u okomitom položaju te smjer i redoslijed kretanja klipova [7]



Slika 2.4. Prikaz radilice u poprečnoj ravni i smjer te redoslijed kretanja klipova [7]

2.2. Prijedlog sklopovskog rješenja



Slika 2.5. Blokovski prikaz makete V8 motora

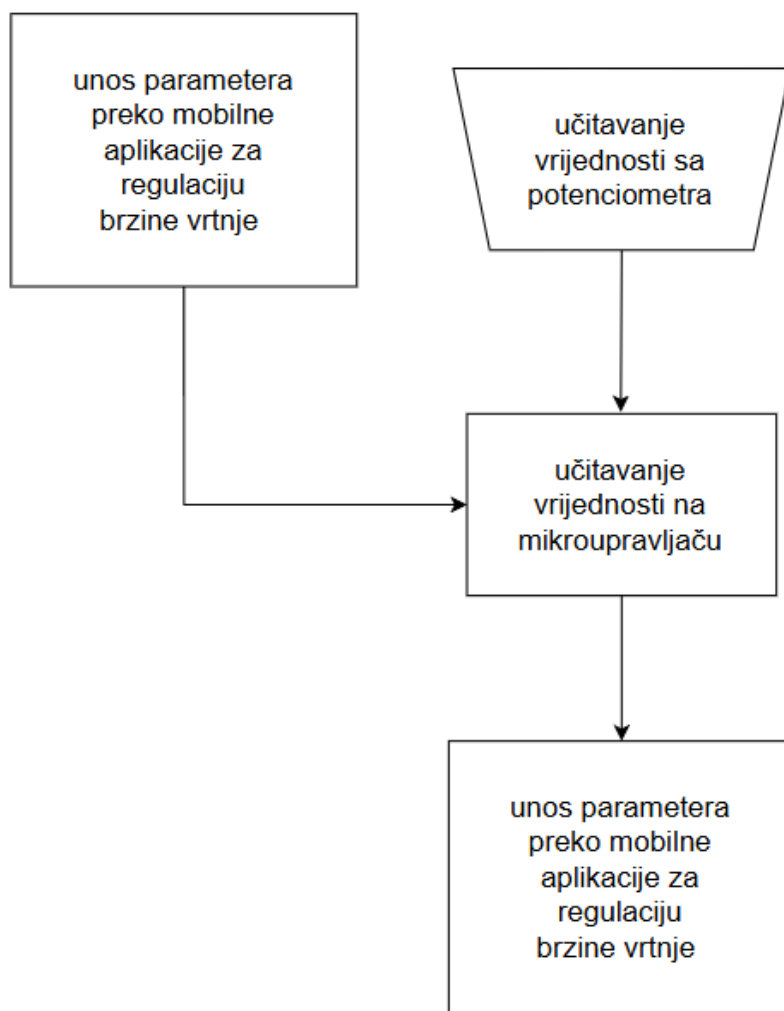
Na slici možemo vidjeti zamišljeni koncept po kojem će se maketa izraditi. Potenciometar ima ulogu unosa analognog signala preko mikroupravljača, te će isti potenciometar regulirati izlazni PWM signal iz mikroupravljača preko kojeg se upravlja brzinom vrtnje elektromotora. Četiri tipkala unose digitalni signal na mikroupravljač i određuju raspon PWM signala koji će potenciometar izregulirati.

Potenciometar, koji služi kao analogni ulaz, omogućava korisniku da precizno kontrolira vrijednost signala koji se šalje mikroupravljaču. Ovaj signal se potom obrađuje unutar mikroupravljača kako bi se generirao odgovarajući PWM (Pulse Width Modulation) signal. PWM signal je ključan za kontrolu brzine elektromotora, jer mijenjanjem *duty cycle-a* možemo kontrolirati količinu snage koja se dostavlja motoru, te time regulirati njegovu brzinu.

Elektromotor, povezan s mikroupravljačem, koristi ovaj PWM signal za prilagodbu svoje brzine vrtnje. Precizna kontrola brzine vrtnje elektromotora omogućava fino podešavanje performansi sustava prema potrebama specifične primjene.

Četiri tipkala pružaju dodatne mogućnosti konfiguracije sustava. Svako tipkalo šalje digitalni signal mikroupravljaču koji zatim podešava raspon unutar kojeg potenciometar može regulirati PWM signal. Ovo omogućava korisniku da brzo promijeni način rada ili osjetljivost sustava bez potrebe za fizičkim prilagođavanjem potenciometra.

2.3. Prijedlog programskog rješenja



Slika 2.6. Blokovski prikaz programskog rješenja za maketu V8 motora

U *setup* dijelu programa inicijaliziraju se potrebni pinovi i komponente za rad makete. To uključuje konfiguraciju pinova za motor, potenciometar i tipkala. Također se konfigurira Bluetooth veza za komunikaciju s mobilnom aplikacijom. PWM kanal za upravljanje brzinom motora također se postavlja s odgovarajućim frekvencijom i rezolucijom.

U *loop* dijelu programa odvija se neprekidni rad sustava. Prvo se provjerava postoji li novi unos podataka preko Bluetooth veze. Ako postoji, ažuriraju se vrijednosti potenciometra ili se prilagođava minimalna vrijednost PWM signala prema primljenom ID-u tipke iz aplikacije.

Nakon toga, program očitava trenutnu vrijednost potencijometra za ručno podešavanje brzine motora i stanje tipkala. Promjene stanja tipkala šalju se natrag aplikaciji putem Bluetooth veze radi vizualnog povratnog informiranja korisnika o promjenama.

Ako se koristi ručno podešavanje brzine, određuje se PWM signal temeljen na očitavanju potencijometra. U slučaju korištenja aplikacije za daljinsko upravljanje, brzina motora postavlja se prema vrijednostima koje aplikacija šalje. Motor se upravlja PWM signalom za koji se računa *dutyCycle*.

3. REALIZACIJA MAKETE V8 MOTORA

U nastavku rada detaljno će biti opisano kako je izrađena demonstracijska maketa, uključujući korištene alate i vrstu elektronike koja je primijenjena u procesu izrade. Također će biti objašnjeno kako su svi dijelovi međusobno povezani u jednu funkcionalnu cjelinu, uz naglasak na operativne principe i rad svakog pojedinog dijela sustava. Cilj ovog dijela rada je pružiti cjelovitu tehničku analizu koja će olakšati razumijevanje tehničkih aspekata projekta te omogućiti eventualnu reprodukciju i unaprjeđenje makete u budućim radovima.

3.1. Korišteni alati i programska okruženja

Za izradu mehaničkih dijelova demonstracijske makete korišten je program Fusion 360. Nakon izrade modela u Fusionu, svi dijelovi su izrađeni korištenjem 3D printera Prusa MK3S+ i Prusa MINI+, koristeći PLA plastiku kao materijal za printanje. Za programski dio makete, alat koji se primjenjivao u izradi je Arduino IDE. Arduino IDE najčešće služi za implementiranje programa u svim vrstama mikroupravljača Arduino, no specifično za realizaciju ovog projekta, nije dopušteno koristiti Arduino mikroupravljače. Kako bi Arduino IDE bio kompatibilan sa mikroupravljačem koji se koristio u izradi makete, trebalo je preko interneta skinuti paket namjenjen za korišteni mikroupravljač te biblioteke za korištene elektroničke uređaje.

3.2. Realizacija sklopovskog rješenja

Demonstracijska maketa izmodelirana je s ciljem prikaza funkcioniranja mehaničkog sustava, pri čemu su njezini dijelovi izrađeni pomoću 3D printera. Maketa je konstruirana u obliku sklopivog sustava koji se sastoji od više dijelova, a svaki dio se može samostalno printati i potom sastaviti u cjelovitu strukturu. Glavni dijelovi makete uključuju cilindre (ukupno 8 komada) koji su printani s prozirnim filamentom kako bi se omogućio uvid u unutarnju dinamiku, te klipove i klipnjače (također 8 komada) koji su izrađeni u različitim bojama filameta (siva i crvena). Ostali ključni elementi uključuju ležajevе radilice, ležajevе za DC motor, stalke za cilindre i ležajevе radilice te spojnice za cilindre. Dok su cilindre printani s prozirnim filamentom radi vizualizacije unutarnjih pokreta, ostali dijelovi su printani s sivim i crvenim filamentom za estetski i funkcionalni kontrast.

Za upravljanje ovim mehaničkim sustavom korišten je mikroupravljač *ESP32 DEVKIT V1*, poznat po svojoj fleksibilnosti i mogućnosti integracije s različitim perifernim uređajima. O ESP32

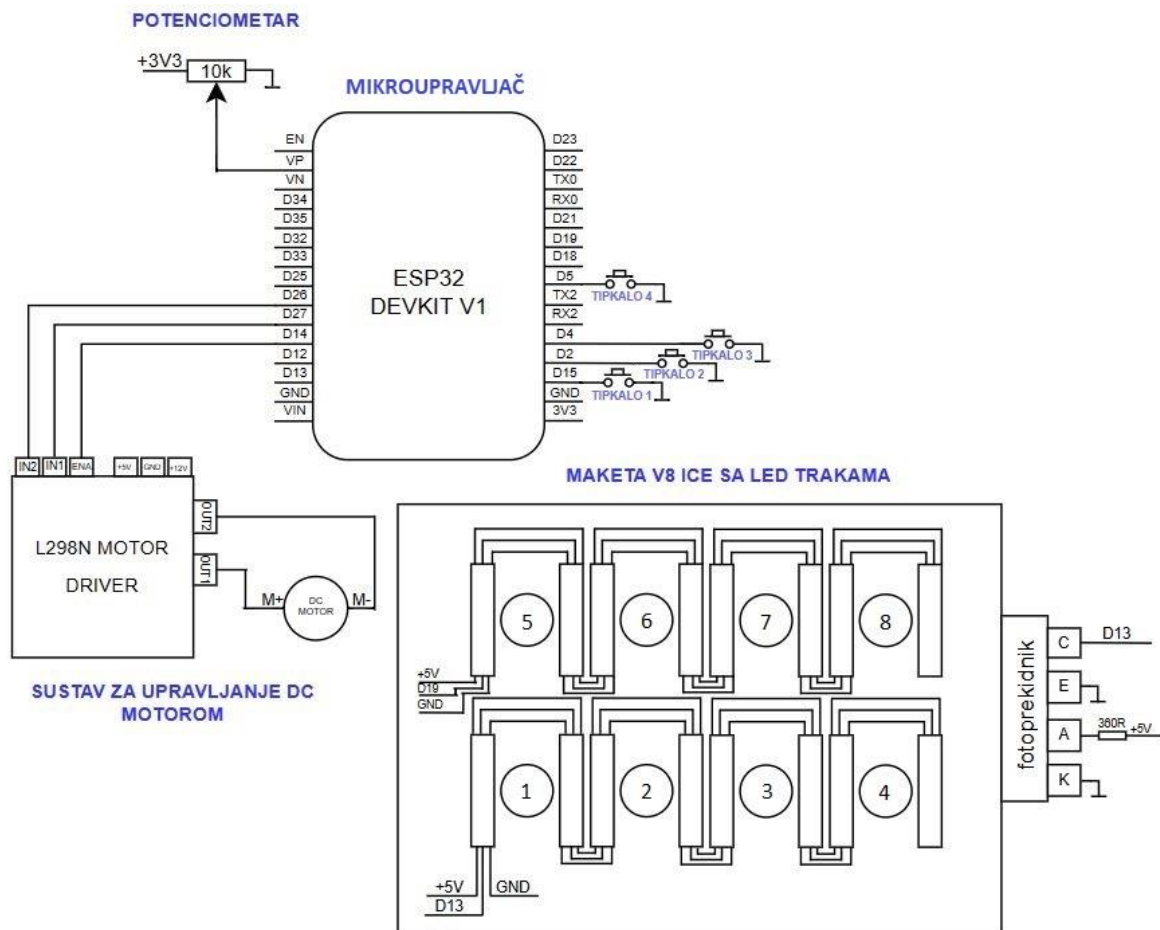
mikroupravljaču i njegovoj strukturi i primjeni, može se više pročitati u prilogu rada, pod naslovom “ESP32 mikroupravljač”. DC motor, koji se koristi u ovom projektu, napaja se istosmjernim naponom od 12V i ima nazivnu struju od 360mA te maksimalno 360 okretaja u minuti. Za upravljanje motorom korišten je L298N motor driver sa dvostrukim H-mostom, koji omogućava kontrolu smjera vrtnje motora i regulaciju brzine pomoću Pulse Width Modulation (PWM) signala.

Pinout L298N drivera je izveden na sljedeći način:

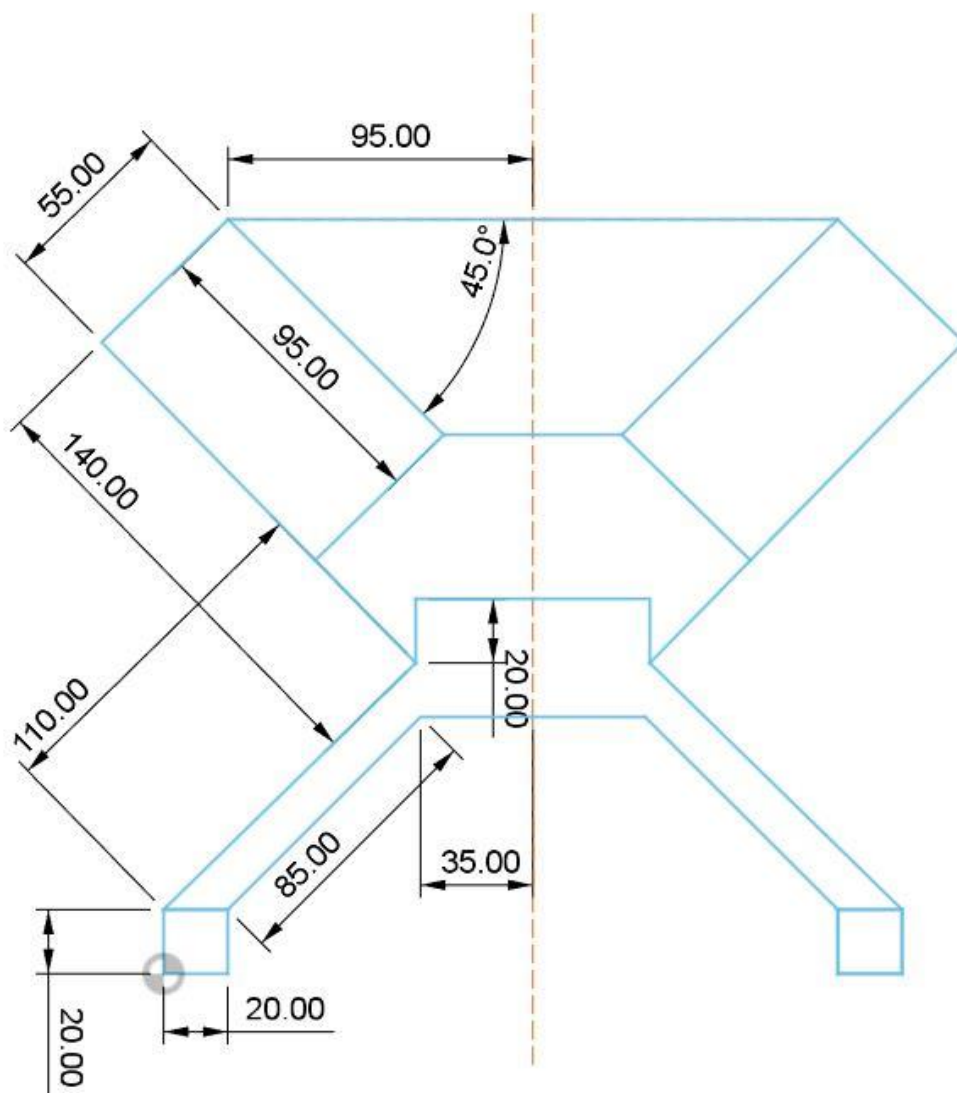
- IN1 i IN2 su spojeni na pinove D27 i D26 na ESP32 mikroupravljaču i koriste se za upravljanje smjerom rotacije motora A.
- ENA je spojen na pin D14, koji služi za aktivaciju PWM signala koji regulira brzinu motora A.
- +5V i GND pinovi su namijenjeni za napajanje elektroničkih komponenti L298N drivera.
- +12V i GND pinovi su za napajanje samog DC motora.

Kako bi se omogućila regulacija brzine vrtnje motora, upotrijebljen je potencijometar s otpornošću od 10k Ω . Potencijometar je povezan s analognim ulazom VP na ESP32 mikroupravljaču i napaja se s +3V3 i GND pinova. Povezivanje potencijometra omogućava korisniku da kontinuirano prilagođava brzinu motora ovisno o vrijednosti koju čita iz potencijometra.

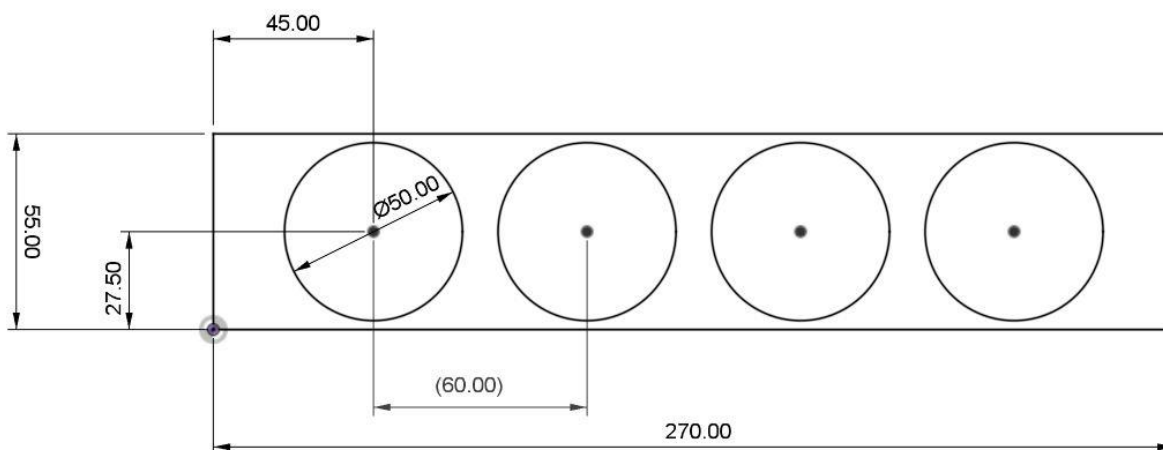
Osim toga, za kontrolu raspona brzine vrtnje motora korištena su četiri tipkala. Tipkala su povezana s pinovima D15, D2, D4 i D5 na mikroupravljaču i omogućuju korisniku odabir između različitih raspona brzine vrtnje motora. Svako tipkalo može aktivirati različite razine brzine, čime se omogućava precizno upravljanje performansama motora.



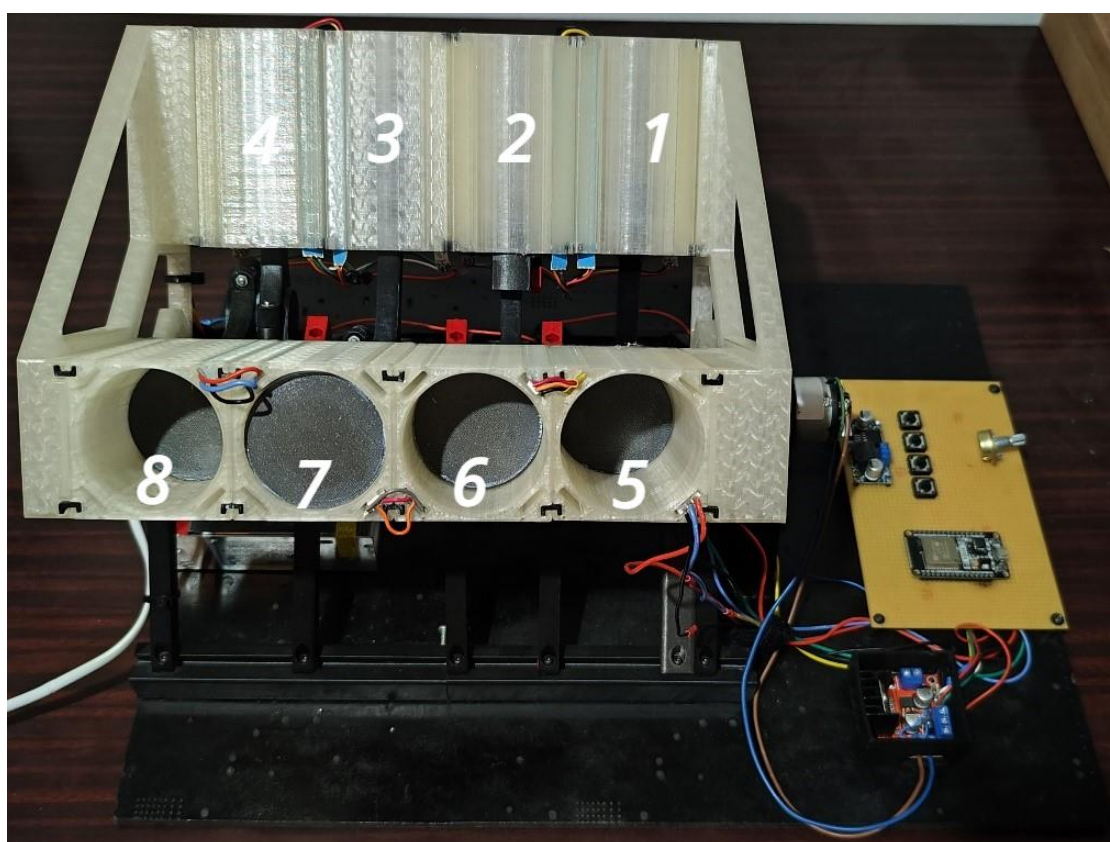
Slika 3.1. Prikaz realizirane sheme demonstracijske makete



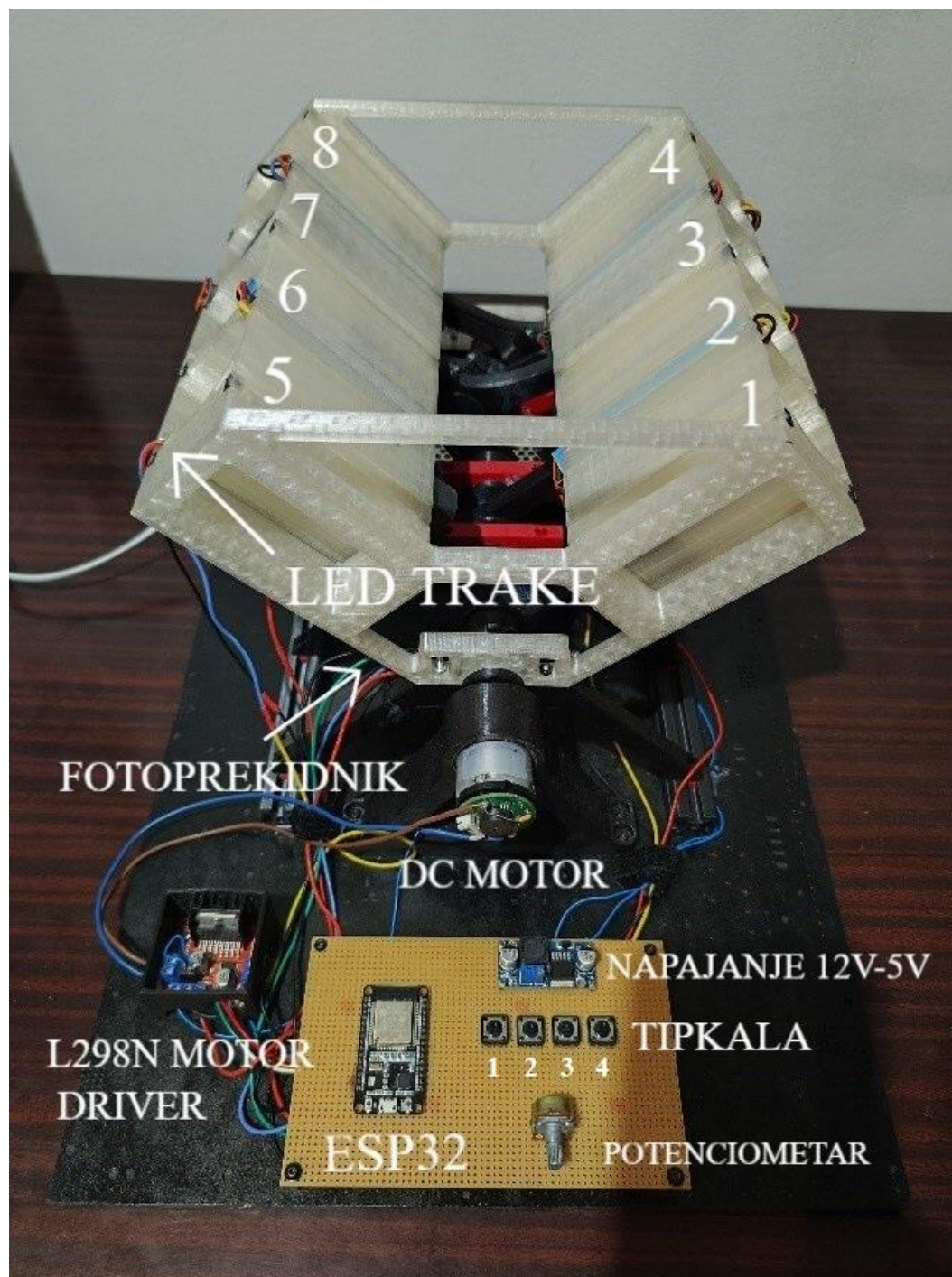
Slika 3.2. Skica realizirane makete V8 motora (prednji dio)



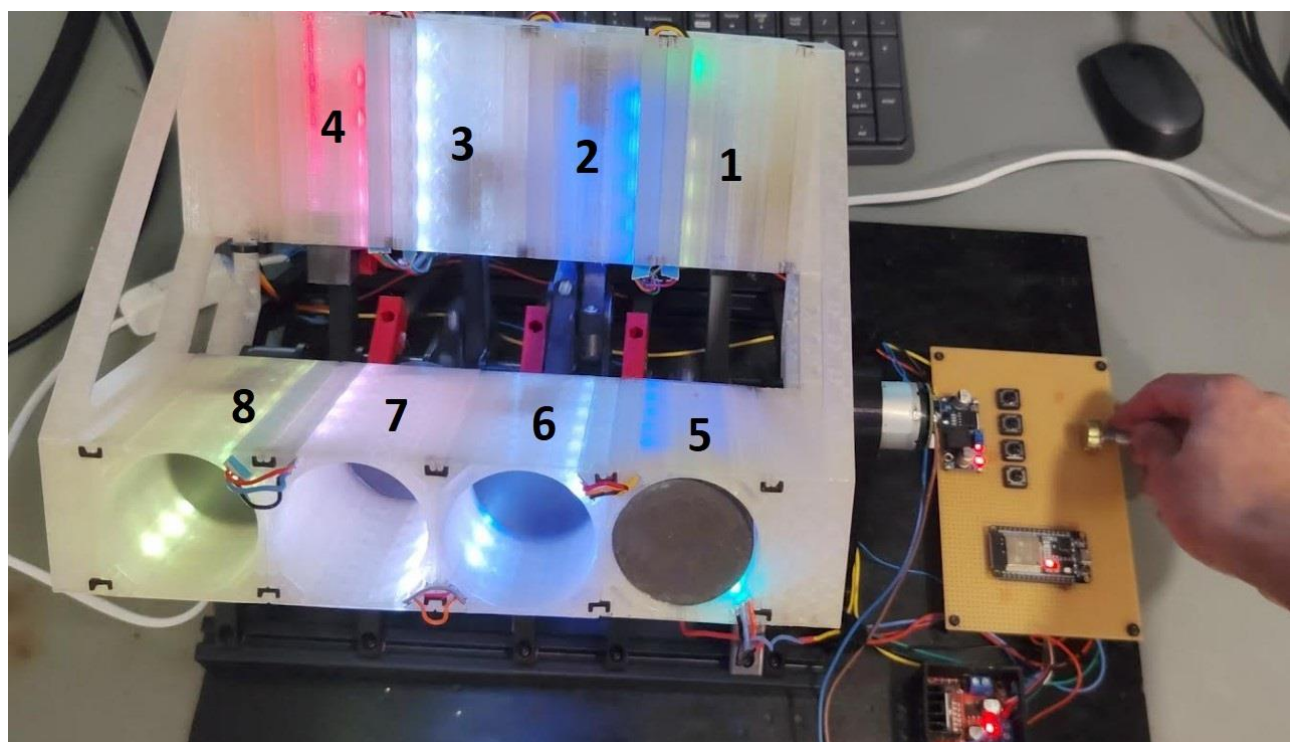
Slika 3.3. Skica realizirane makete V8 motora (gornji dio)



Slika 3.4. Realizirana demonstracijska maketa V8 motora (model bloka motora)

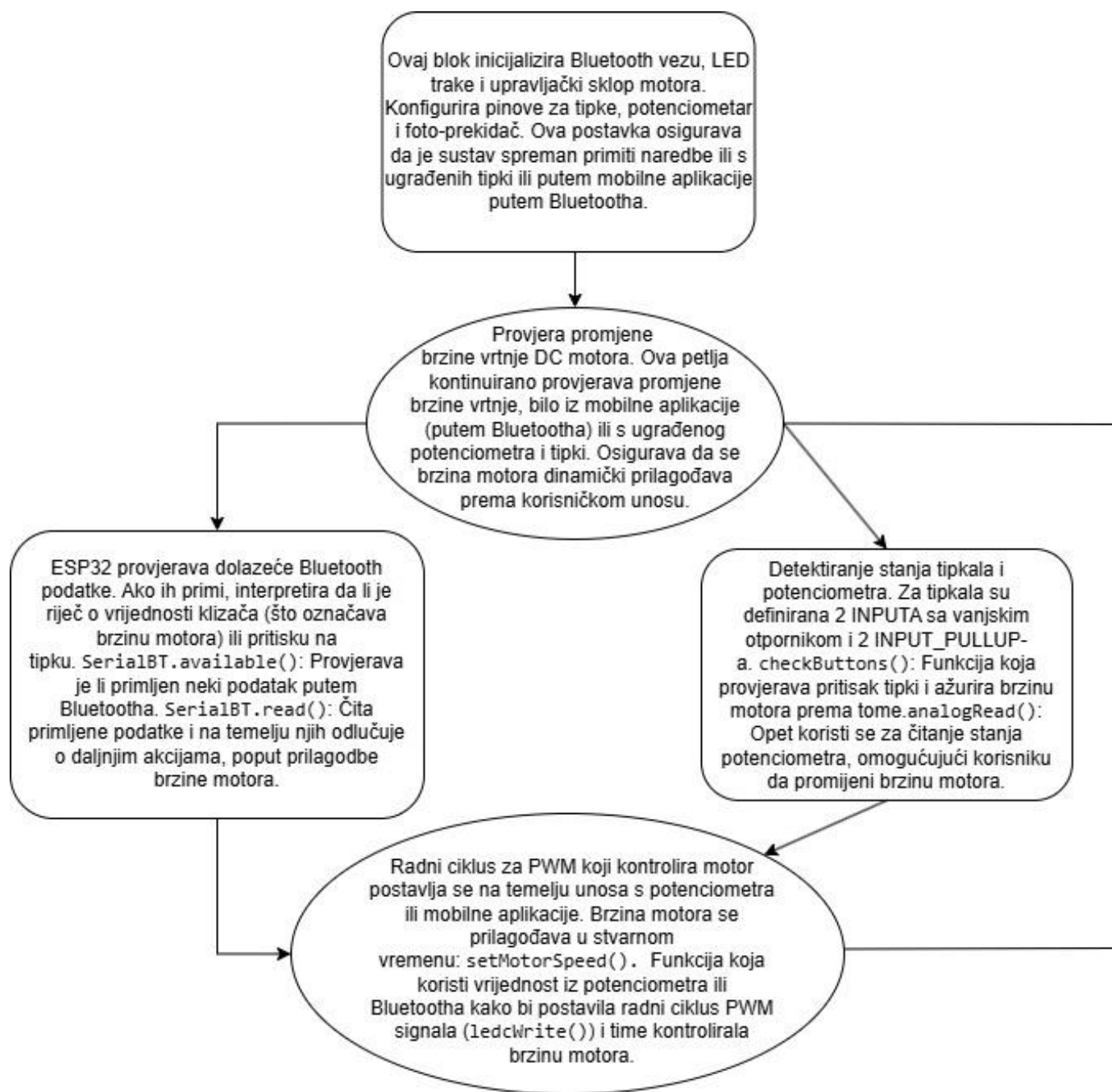


Slika 3.5. Realizirana demonstracijska maketa V8 motora (blok motora, DC motor i elektronički dio)



Slika 3.6. Prikaz makete u funkciji

3.3. Realizacija programskog rješenja



Slika 3.7. Prikaz blok dijagram toka realiziranog programskog koda

Na početku koda, uključena je ključna biblioteka: *BluetoothSerial.h*. Biblioteka omogućava Bluetooth komunikaciju između ESP32 mikroupravljača i mobilne aplikacije.

Sljedeće su definicije upravljačkih pinova i potrebnih varijabli:

- Bluetooth: varijable potrebne za komunikaciju preko Bluetootha.
- DC motor: pinovi i varijable za kontrolu DC motora, uključujući Pulse Width Modulation (PWM) za kontrolu brzine vrtnje.
- Potenciometar: pin za čitanje vrijednosti potenciometra.

Unutar funkcije *setup()*, inicijaliziraju se različiti pinovi određivanjem njihovog načina rada (ulaz ili izlaz). Definiraju se parametri za PWM, uključujući kanal, frekvenciju i rezoluciju. Glavna petlja *loop()*, koja se kontinuirano izvršava, ima nekoliko ključnih funkcija:

- Komunikacija s mobilnom aplikacijom: očitavanje podataka s aplikacije putem Bluetootha.
- Detekcija promjena stanja tipkala i potenciometra: Na osnovu tih podataka, određuje se vrijednost PWM-a, odnosno brzina vrtnje motora.
- Kontrola motora: Motor se postavlja na odgovarajuću brzinu vrtnje na osnovu izračunate PWM vrijednosti. Za potpuni ciklus četverotaktnog motora, potrebna su dva puna okreta motora

Programski kod, koji je napisan za realiziranu demonstracijsku maketu, naveden je u prilogu rada, pod naslovom „Programski kod projekta“.

4. TESTIRANJE I REZULTATI

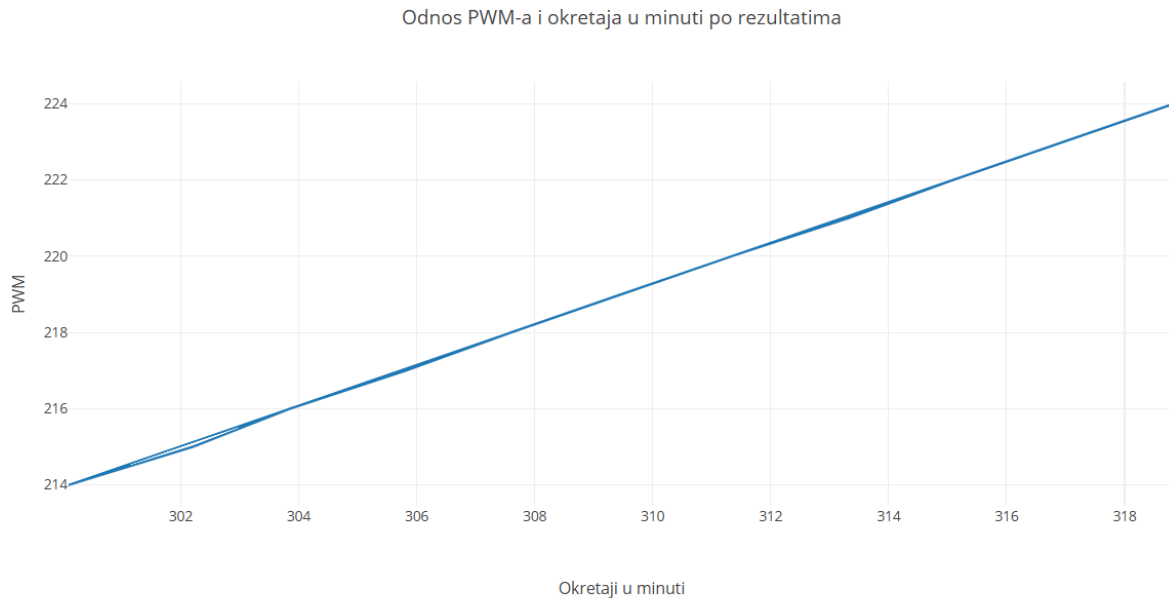
4.1. Metodologija testiranja

Maketa motora je montirana na čvrstu površinu kako bi se minimizirale vibracije i osigurao stabilan rad tijekom testiranja. Korišteno je napajanje koje pretvara 220V izmjenične struje u 12V istosmjerne struje, te napaja L298N motor driver koji spaja motor i ESP32 mikroupravljač koji je povezan s motorom i potencijometrom. Potencijometar je kalibriran sa 1. tipkalom tako da njegovo potpuno okretanje od 0% do 100% odgovara PWM vrijednostima od 214 do 224. Za svaki test, vrijednost potencijometra je postavljena na određeni postotak (0%, 10%, 20%, itd.), i očitana analogna vrijednost je pretvorena u odgovarajuću PWM vrijednost. Brzina vrtnje motora (RPM) je izračunata na temelju PWM vrijednosti. Očitane PWM vrijednosti i izmjerene brzine vrtnje motora su zabilježene. Za svaku vrijednost potencijometra, proces je ponavljen 22 puta kako bi se osigurala statistička pouzdanost podataka. Nakon prikupljanja podataka, izračunate su srednje vrijednosti brzine vrtnje za svaki postotak potencijometra. Također, izračunata je standardna devijacija kako bi se procijenilo mjerno odstupanje i preciznost sustava.

4.2. Rezultati testiranja

Broj Mjerenja	Vrijednost Potenciometra (%)	PWM Vrijednost	Brzina Vrtnje Motora (RPM)
1	0%	214	300.1
2	10%	215	302.2
3	20%	216	303.85
4	30%	217	305.825
5	40%	218	307.6
6	50%	219	309.475
7	60%	220	311.35
8	70%	221	313.325
9	80%	222	315.1
10	90%	223	316.975
11	100%	224	318.85
12	0%	214	300.1
13	10%	215	302.2
14	20%	216	303.85
15	30%	217	305.825
16	40%	218	307.6
17	50%	219	309.475
18	60%	220	311.35
19	70%	221	313.325
20	80%	222	315.1
21	90%	223	316.975
22	100%	224	318.85
Srednja vrijednost		219	309.19
Standardna devijacija			5.83

Tablica 4.1. Rezultati mjerenja sa srednjom vrijednošću i standardnom devijacijom

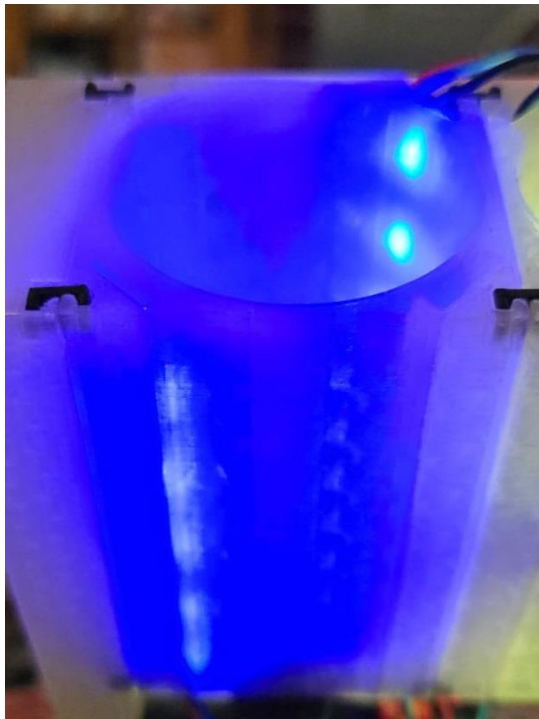


Slika 4.1. Rezultati u grafičkom obliku

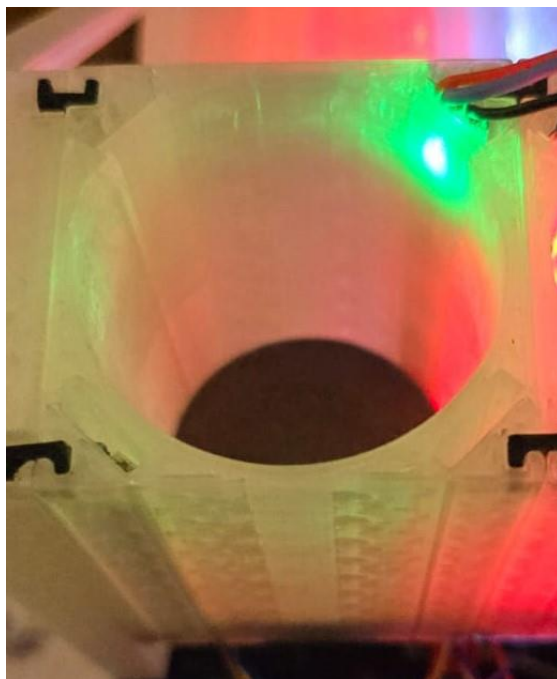
Testiranje makete motora pokazalo je dosljedan i precizan odnos između vrijednosti potencijometra i brzine vrtnje motora, što je ključno za pouzdan rad sustava. Rezultati su pokazali da se PWM vrijednost može koristiti kao efektivni alat za kontrolu brzine vrtnje motora. Vrijednosti potencijometra od 0% do 100% uspješno su prevedene u odgovarajuće PWM vrijednosti od 214 do 224, što je omogućilo precizno upravljanje brzinom motora.

Rezultati mjerenja brzine vrtnje motora (RPM) za različite PWM vrijednosti prikazani su u tablici, gdje se može vidjeti da se brzina vrtnje postupno povećavala s povećanjem PWM vrijednosti. Prosječna brzina vrtnje iznosila je 309.19 RPM, dok je standardna devijacija iznosila 5.83 RPM. Ovo odstupanje je vrlo malo, što ukazuje na visoku stabilnost i pouzdanost sustava.

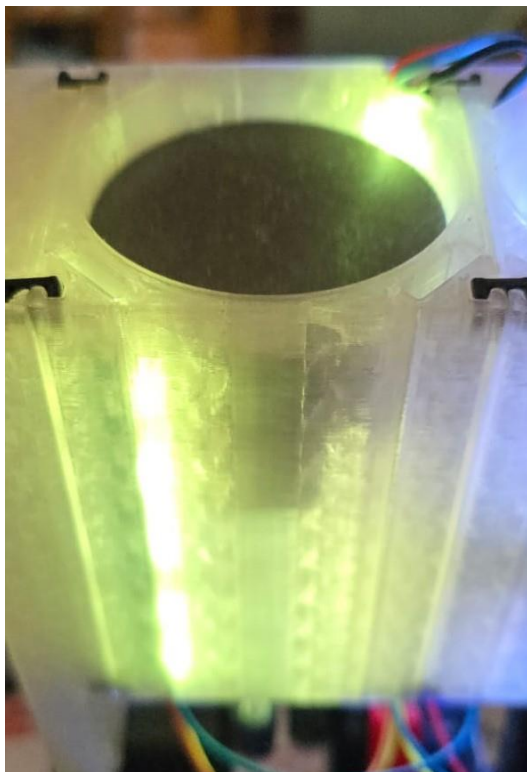
4.3. Prikaz makete u funkciji



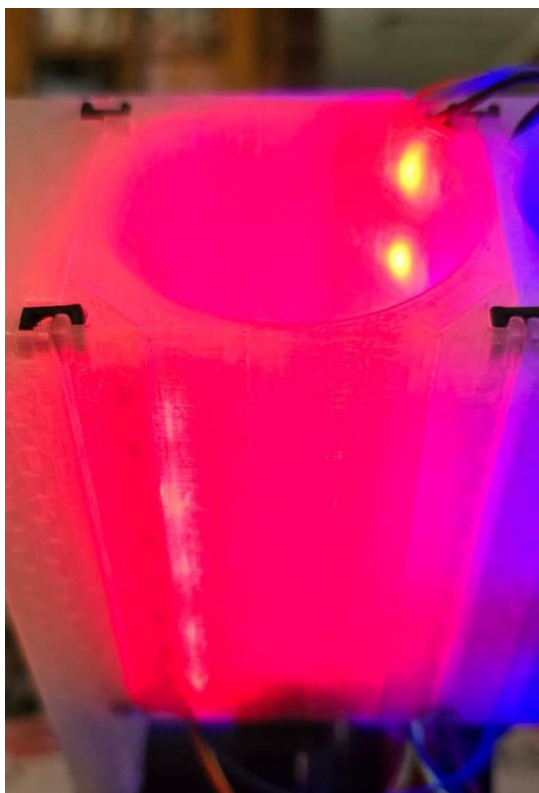
Slika 4.2. Prikaz cilindra sa usisom (plava)



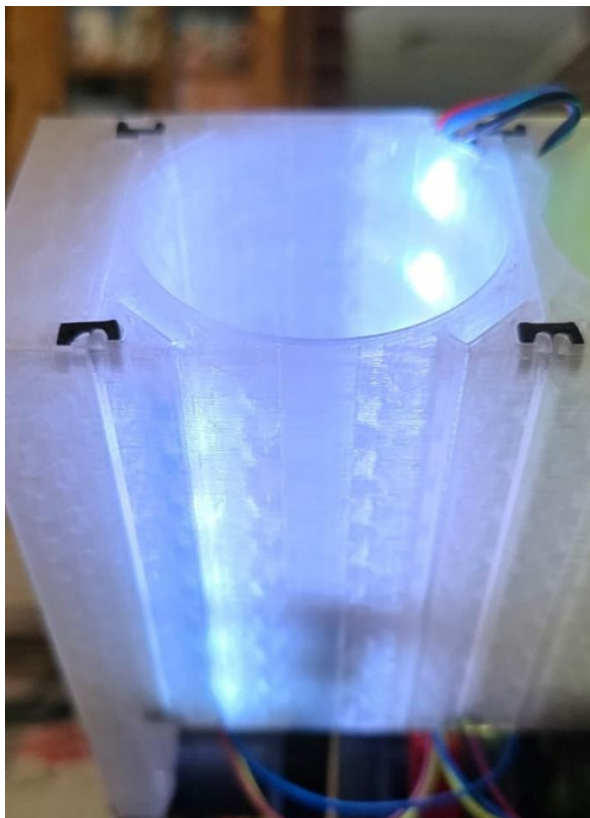
Slika 4.3. Prikaz cilindra sa ubrizgavanjem goriva (zelena)



Slika 4.4. Prikaz cilindra sa kompresijom (žuta)



Slika 4.5. Prikaz cilindra sa ekspanzijom (crvena)



Slika 4.6. Prikaz cilindra sa ispuhom (bijela)



Slika 4.7. Prikaz zajedničkog rada klipova u cilindrima



Slika 4.8. Prikaz zajedničkog rada klipova u cilindrima

5. ZAKLJUČAK

Ovaj rad detaljno opisuje konstrukciju i funkcioniranje V8 motora, počevši od osnovnih karakteristika V8 motora u stvarnom životu, poput konfiguracije cilindara i primjene u vozilima visokih performansi i teškim terencima. Glavni fokus rada je na izradi demonstracijske makete V8 motora koja simulira rad pravog motora. Maketa se sastoji od mehaničkih komponenti poput bloka motora, radilice i klipova te elektroničkih komponenti, potencijometar za kontrolu brzine vrtnje i tipkala za određivanje raspona PWM signala. Cilj projekta je bio integrirati ove dijelove u funkcionalni sustav koji omogućuje simulaciju radnog ciklusa motora i kontrolu brzine motora preko Bluetootha putem mobilne aplikacije. Kroz opis postupka izrade, programiranja i testiranja, dokazano je da je maketa uspješno replicirala princip rada četverotaktnog motora. Korištenje potencijometra za kontrolu brzine, precizno upravljanje PWM signalom za motore te detekcija svakog ciklusa klipova pokazali su se kao ključni elementi za stabilan i pouzdan rad sustava. U konačnici, testiranje je potvrdilo dosljednost i preciznost mjerenja brzine vrtnje motora u skladu s postavkama potencijometra, pružajući validaciju funkcionalnosti makete. Ova demonstracijska maketa ne samo da ilustrira teoretske koncepte V8 motora, već i praktično demonstrira implementaciju istih kroz integraciju mehaničkih i elektroničkih komponenti.

LITERATURA

- [1] B. Ponsford, »The History and Development if the V8 Engine, [http://bmp.thefloatingwidget.net/work/The History and Development of the V8 Engine.pdf](http://bmp.thefloatingwidget.net/work/The_History_and_Development_of_the_V8_Engine.pdf), pristup: 29. 6. 2022.
- [2] Motor Illustrated, <https://motorillustrated.com/chevrolet-reveals-10-litre-1004hp-naturally-aspirated-v8-engine/87420/>, pristup: 29. 6. 2022.
- [3] OldtimerArchiv.com, <https://oldtimerarchiv.com/the-amg-6-3-litre-v8/>, pristup: 29. 6. 2022.
- [4] GetDrawings, <https://getdrawings.com/get-drawing#v8-engine-drawing-9.gif>, pristup: 15.7.2024.
- [5] carsblog, <https://www.dubizzle.com/blog/cars/v8-engine/>, pristup: 29. 6. 2022.
- [6] dreamstime, <https://www.dreamstime.com/type-internal-combustion-engine-called-four-stroke-there-movements-strokes-piston-entire-firing-image158245426>, pristup: 29. 6. 2022.
- [7] Curbside Tech: V8 Engine Crankshafts and Firing Orders – Good Vibrations, Curbside Classic, <https://www.curbsideclassic.com/blog/tech/curbside-tech-v8-engine-crankshafts-and-firing-orders-good-vibrations/>, pristup: 29. 6. 2022.
- [8] Random Nerd Tutorials, <https://randomnerdtutorials.com/getting-started-with-esp32/>, pristup: 26.6. 2024.
- [9] Last Minute Engineers, <https://lastminuteengineers.com/esp32-pinout-reference/>, pristup: 30.5.2024.
- [10] Components 101, <https://components101.com/modules/l293n-motor-driver-module>, pristup: 26.6. 2024.
- [11] Random Nerd Tutorials WS2812B, <https://randomnerdtutorials.com/guide-for-ws2812b-addressable-rgb-led-strip-with-arduino/>, pristup: 26.6.2024.

PRILOZI I DODACI

Primijenjeni hardver u radu

ESP32 mikroupravljač

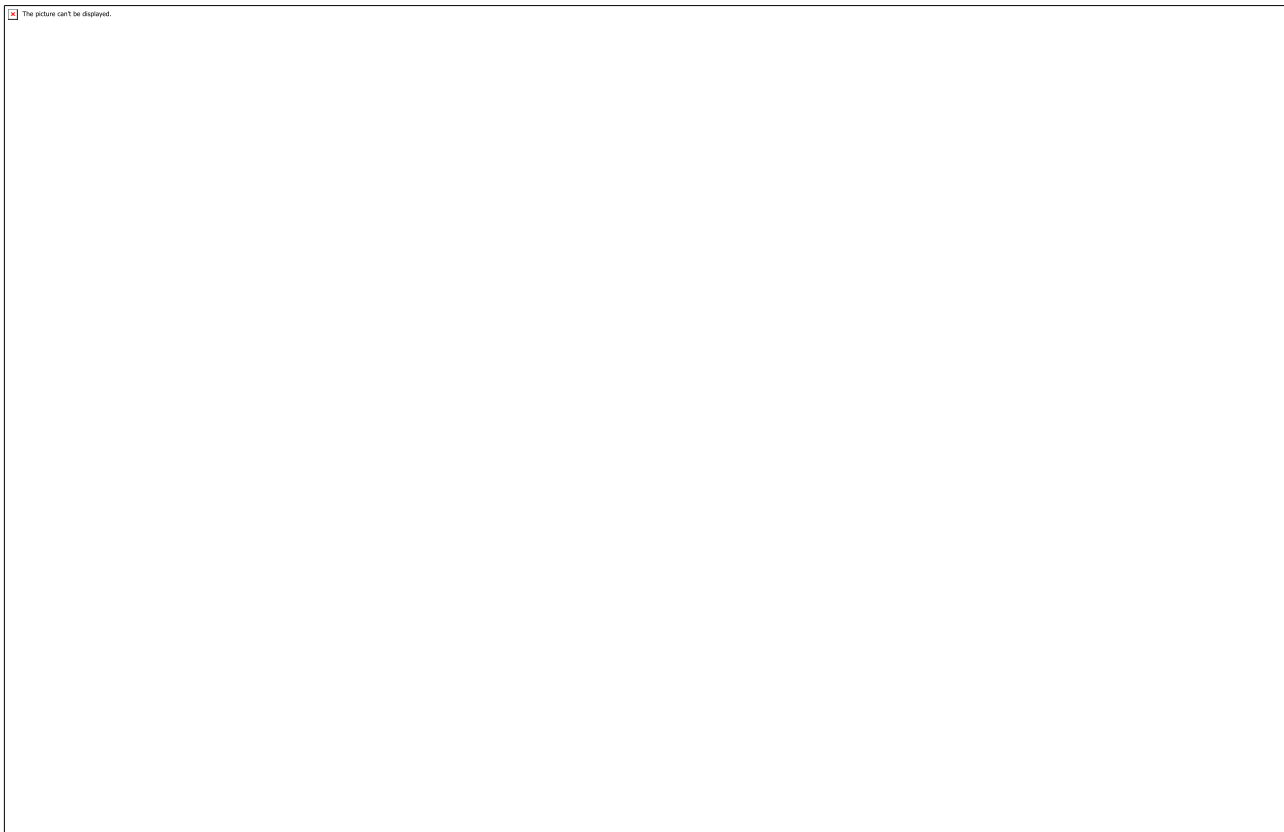
ESP32 je mikroupravljač napravljen od strane tvrtke Espressif Systems. Najčešće ga se koristi i IoT (Internet of Things) aplikacijama zbog bogatog skupa značajki i zbog velikih mogućnosti. ESP32 posjeduje dvojezgreni Tensilica LX6 mikroprocesor koji može raditi na frekvenciji do 240MHz. Ova dvojezrena arhitektura omogućava visoke performanse i učinkovit rad, čak i kod složenih zadataka.

ESP32 je dizajniran s vrlo niskom potrošnjom energije, podržavajući različite načine rada koji štede energiju, kao što su light sleep, deep sleep i hibernacija. Zbog takve značajke, vrlo je pogodan za aplikacije koje se napajaju baterijama, gdje je energetska učinkovitost ključna.

Što se tiče memorije, ESP32 nudi 520 KB SRAM-a i 448 KB ROM-a, što omogućava dovoljno prostora za pohranu koda i podataka. Uz to, standardno dolazi s 4 MB ugrađene flash memorije, iako se može koristiti i vanjska flash memorija za dodatni prostor za pohranu.

ESP32 mikroupravljač posjeduje bogat skup ulazno-izlaznih (I/O) mogućnosti. Ima do 36 programabilnih GPIO pinova, 18 kanala 12-bitnog ADC-a (Analog to Digital Converter), dva 8-bitna DAC kanala (Digital to Analog Converter) i 10 kapacitivnih touch senzora. Također, podržava do 16 kanala PWM-a (Pulse Width Modulation), što omogućava preciznu kontrolu izlaznih signala.

ESP32 mikroupravljač dolazi u raznim paketima, uključujući QFN i WLCSP, koji su pogodni za različite primjene i prostorne ograničenja. Zahvaljujući svojoj svestranosti, ESP32 se koristi u mnogim primjenama, uključujući IoT uređaje, pametne kućne uređaje, nosive uređaje i industrijsku automatizaciju. Njegova niska potrošnja energije i bogat skup značajki čine ga idealnim za uređaje na baterijski pogon, kao i za aplikacije koje zahtijevaju visoke performanse. [8]



Slika P1.1. Raspodjela pinova na ESP32 mikroupravljaču [9]

L298N motor driver

L298N je integrirani krug dizajniran za kontrolu kretanja motora kroz H-bridge konfiguraciju. H-bridge je električni sklop koji omogućuje promjenu smjera struje kroz motor, omogućujući mu da se kreće naprijed ili natrag. L298N može kontrolirati dva DC motora istovremeno, ili jedan bipolarni korakni motor, što ga čini vrlo fleksibilnim u raznim aplikacijama.

Ključna značajka L298N drivera je njegova sposobnost da radi s različitim naponima, od 5V do 35V, te da isporučuje struju do 2A po kanalu kontinuirano, uz vršnu struju do 3A. Ova snaga omogućuje kontrolu motora različitih veličina i snaga, od malih igračih motora do većih industrijskih motora. Također, integrirana zaštita od pregrijavanja osigurava dugotrajnost i pouzdanost uređaja, sprječavajući oštećenja uslijed prekomjernog zagrijavanja. [10]

Programski kod projekta

```
#include "BluetoothSerial.h"
#include<FastLED.h>

/*led traka*****/
#define NUM_LEDS 47 //broj LED-ica u LED traci
#define LED_PIN1 19 //upravljacki pin desnog dijela LED-ica
#define LED_PIN2 13 //upravljacki pin lijevog dijela LED-ica

int i; //koristi se u for petljama za funkcije LED-ica
int nula = 0; // za brojanje zubaca
int ciklus = 0; // detekcija novog okretaja motora
CRGB leds1[NUM_LEDS]; //polje LED-ica, koisti se za pristup pojedinoj
LED-ici desne strane
CRGB leds2[NUM_LEDS]; //polje LED-ica, koisti se za pristup pojedinoj
LED-ici lijeve strane

/*BLUETOOTH I APLIKACIJA*****/
BluetoothSerial ESP_BT;
int incoming;
int slider_value = 0; //vrijednost potencijometra iz aplikacije
int appFlag = 0; //detektira upotrebu aplikcije, kada je 0 onda se
upravlja sa pločice,
// a kada je 1 onda iz aplikacije

//prvo tipkalo sa lijeva na pločici je pin 15, drugo pin 2
// (oba tipkala moraju imati vanjski pullup otpornik),
// a treće tipkalo pin 4 i četvrto tipkalo pin 5 mogu preko internog
pullup otpornika
/*DEFINIRANJE PINOVA NA MOTOR DRIVERU ZA DC
MOTOR*****/
int motor1Pin1 = 27;
int motor1Pin2 = 26;
int enable1Pin = 14;

// postavke PWM-a za DC MOTOR
const int freq = 30000;
const int pwmChannel = 0;
const int resolution = 8;
int dutyCycle = 200;
int minDutyCycle = 214;
```

```

/*POTENCIOMETAR******/
const int potPin = 34;
int potValue = 0;

/*TIPKALA ZA BRZINE******/
int button_pin_1 = 15; //0-25% vrijednost dutycycla od 214-224(jer
dijelimo s 400)
int button_pin_2 = 2; //25-50% dutycycle od 224-234
int button_pin_3 = 4; //50-75% dutycycle od 234-244
int button_pin_4 = 5; //75-100% dutycycle od 244-254

// varijable za spremanje stanja tipki
int currentState1;
int currentState2;
int currentState3;
int currentState4;

int lastState1;
int lastState2;
int lastState3;
int lastState4;

/*FOTOPREKIDNIK******/
int senzor=33; //pin na koji spajamo senzor
int stanjeSenzora; //varijabla u koju spremamo stanje senzora
int zastavica=0; //koristi se za detekciju prolaska zubca
int timer = 0; //brojač pomoću kojeg se detektira razmak između zubaca
// kako bi se odredio položaj nedostajećeg zubca koji označava nulti
položaj

void setup() {
    ESP_BT.begin("ESP32_Control"); //ime bluetooth uredaja

    //inicijalizacija LED-ica
    FastLED.addLeds<WS2812B, LED_PIN1, GRB>(leds1, NUM_LEDS);
    FastLED.addLeds<WS2812B, LED_PIN2, GRB>(leds2, NUM_LEDS);
    FastLED.setBrightness(30);

    //konfiguracija DC motora

```

```

pinMode(motor1Pin1, OUTPUT);
pinMode(motor1Pin2, OUTPUT);
pinMode(enable1Pin, OUTPUT);
//PWM za DC motor
ledcSetup(pwmChannel, freq, resolution);
ledcAttachPin(enable1Pin, pwmChannel);

//konfiguracija pinova za tipkala
pinMode(button_pin_1, INPUT); //koristi se vanjski pullup otpornik
pinMode(button_pin_2, INPUT); //koristi se vanjski pullup otpornik
pinMode(button_pin_3, INPUT_PULLUP); //koristi se interni pullup
otpornik
pinMode(button_pin_4, INPUT_PULLUP); //koristi se interni pullup
otpornik

//fotoprekidnik
pinMode(senzor, INPUT);
}

void loop() {

    // -----komunikacija s mobilnom aplikacijom -----
    -----
    if (ESP_BT.available()){
        incoming = ESP_BT.read(); //spremanje primljenih podataka iz
        aplikacije u varijablu incoming
        int button;

        //
        if(incoming < 10){
            slider_value = incoming; //vrijednost potencijometra iz aplikacije
            mogu biti od 1 do 9
        }
        else{
            button = incoming; //primanje oznake tipke koja je pritisnuta - ID
            tipke 10,20,30,40
            appFlag = 1;
        }
        switch (button){
            case 10:
                minDutyCycle = 214;
                break;
            case 20:
                minDutyCycle = 224;
                break;

```

```

        case 30:
            minDutyCycle = 234;
            break;
        case 40:
            minDutyCycle = 244;
            break;
    }
}

//očitavanje vrijednosti sa pločice
potValue = analogRead(potPin);

//očitavanje trenutno stanje tipki
currentState1 = digitalRead(button_pin_1);
currentState2 = digitalRead(button_pin_2);
currentState3 = digitalRead(button_pin_3);
currentState4 = digitalRead(button_pin_4);

//detektira pritisak tipkala i šalje tu informaciju aplikaciji
if(lastState1 == LOW && currentState1 == HIGH){
    minDutyCycle = 214;
    appFlag = 0;
    ESP_BT.write(10); // slanje stanja tipkala u aplikaciju
}
if(lastState2 == LOW && currentState2 == HIGH){
    minDutyCycle = 224;
    appFlag = 0;
    ESP_BT.write(20);
}
if(lastState3 == HIGH && currentState3 == LOW){
    minDutyCycle = 234;
    appFlag = 0;
    ESP_BT.write(30);
}
if(lastState4 == HIGH && currentState4 == LOW){
    appFlag = 0;
    minDutyCycle = 244;
    ESP_BT.write(40);
}

//spremanje stanja tipkala
lastState1 = currentState1;
lastState2 = currentState2;

```

```

lastState3 = currentState3;
lastState4 = currentState4;

//određivanje koriste li se vrijednosti sa pločice ili sa aplikacije
if(appFlag == 0){
    dutyCycle = minDutyCycle + (potValue/400);
}
else{
    dutyCycle = minDutyCycle + slider_value;
}

digitalWrite(motor1Pin1, LOW);
digitalWrite(motor1Pin2, HIGH);
ledcWrite(pwmChannel, dutyCycle);
//detekcija prolaska zubca
stanjeSenzora=digitalRead(senzor); //u varijablu stanjeSenzora
spremamo vrijednost senzora koju očitavamo s pina 2

if ((stanjeSenzora==LOW)&&(zastavica==0)){ //if petlja u kojoj
provjeravamo je li senzor u LOW stanju
// (predmet je prisutan, nema svijetla pa ne provodi i je li zastavica
jednaka nuli)
    zastavica=1; //postavljamo zastavicu na 1 i ona će biti jedan sve
dok je predmet prisutan
// unutar senzora te nam osigurava da se isti predmet ne broji više
puta
    nula++; //povećavamo dok ne prodju svi zubci

//detektira pocetnu točku rotacije
if(timer > 400){
    ciklus = !ciklus;
    nula=0;
}

timer = 0;

/*****upravljanje LED traka*****/
//1. usis - plava
//2. ubrizgavanje - zelena
//3. kompresija - zuta
//4. ekspanzija - crvena
//5. ispuh - bijela

```

```

//prvi ciklus
if(ciklus == 0){
    if((nula == 0) || (nula == 1) || (nula == 2) || (nula == 3)){
        usis_d(36,47); //1.cilindar desno
        usis_l(36,47); //4.cilindar lijevo

        ispuh_d(24,35); //2.cilindar desno
        ispuh_l(0,11); //1.cilindar lijevo

        ekspanzija_d(12,23); //3.cilindar desno
        ekspanzija_l(12,23); //2.cilindar lijevo

        kompresija_d(0,11); //4.cilindar desno
        kompresija_l(24,35); //3.cilindar lijevo
        FastLED.show();
    }
    else if((nula == 4) || (nula == 5) || (nula == 6)){
        ubrizgavanje_d(36,47); //1.cilindar desno
        ubrizgavanje_l(36,47); //4.cilindar lijevo

        usis_d(24,35); //2.cilindar desno
        usis_l(0,11); //1.cilindar lijevo

        ispuh_d(12,23); //3.cilindar desno
        ispuh_l(12,23); //2.cilindar lijevo

        ekspanzija_d(0,11); //4.cilindar desno
        ekspanzija_l(24,35); //3.cilindar lijevo
        FastLED.show();
    }
    else if((nula == 7) || (nula == 8) || (nula == 9) || (nula == 10)){
        kompresija_d(36,47); //1.cilindar desno
        kompresija_l(36,47); //4.cilindar lijevo

        ubrizgavanje_d(24,35); //2.cilindar desno
        ubrizgavanje_l(0,11); //1.cilindar lijevo

        usis_d(12,23); //3.cilindar desno
        usis_l(12,23); //2.cilindar lijevo

        ispuh_d(0,11); //4.cilindar desno
        ispuh_l(24,35); //3.cilindar lijevo
        FastLED.show();
    }
}
}

```

```

    //drugi ciklus
    else if(ciklus == 1){
        if((nula == 0) || (nula == 1) || (nula == 2) || (nula == 3) ||
(nula == 4)){
            ekspanzija_d(36,47); //1.cilindar desno
            ekspanzija_l(36,47); //4.cilindar lijevo

            kompresija_d(24,35); //2.cilindar desno
            kompresija_l(0,11); //1.cilindar lijevo

            ubrizgavanje_d(12,23); //3.cilindar desno
            ubrizgavanje_l(12,23); //2.cilindar lijevo

            usis_d(0,11); //4.cilindar desno
            usis_l(24,35); //3.cilindar lijevo
            FastLED.show();
        }
        else if((nula == 5) || (nula == 6) || (nula == 7) || (nula == 8) ||
(nula == 9) || (nula == 10)){
            ispuh_d(36,47); //1.cilindar desno
            ispuh_l(36,47); //4.cilindar lijevo

            ekspanzija_d(24,35); //2.cilindar desno
            ekspanzija_l(0,11); //1.cilindar lijevo

            kompresija_d(12,23); //3.cilindar desno
            kompresija_l(12,23); //2.cilindar lijevo

            ubrizgavanje_d(0,11); //4.cilindar desno
            ubrizgavanje_l(24,35); //3.cilindar lijevo
            FastLED.show();
        }
    }
}
//detekcija praznine
else if(stanjeSenzora==HIGH){
    //ispitujemo da li je predmet unutar senzora i ako nema vrijednost
    // će biti HIGH i izvršit će se ova petlja
    zastavica=0;    //postavljamo zastavicu na 0 jer je predmet prošao kroz
senzor
    // te time osiguravamo da se pri slijedećem prolazu predmet može
izvršiti
    timer++;
}

```

```

}

/*FUNKCIJE za mijenjanje boja LED
trake*****/

//funkcije za LED-ice desnih cilindara
int usis_d(int x, int y){ //prima vrijednosti x i y koje označavaju
poziciju LED dioda u traci
    for(i = x;i<y;i++){
        leds1[i] = CRGB::Blue; //postavlja boju u plavu koja predstavlja usis
    }
}

int ubrizgavanje_d(int x, int y){
    for(i = x;i<y;i++){
        leds1[i] = CRGB::Black;
    }
    for(i = x;i<y-10;i++){
        leds1[i] = CRGB::Green;
    }
}

int kompresija_d(int x, int y){
    for(i = x;i<y;i++){
        leds1[i] = CRGB::Yellow;
    }
}

int ekspanzija_d(int x, int y){
    for(i = x;i<y;i++){
        leds1[i] = CRGB::Red;
    }
}

int ispuh_d(int x, int y){
    for(i = x;i<y;i++){
        leds1[i] = CRGB::White;
    }
}

//funkcije za LED-ice lijevih cilindara
int usis_l(int x, int y){

```

```

    for(i = x;i<y;i++){
        leds2[i] = CRGB::Blue;
    }
}

int ubrizgavanje_1(int x, int y){
    for(i = x;i<y;i++){
        leds2[i] = CRGB::Black;
    }
    for(i = x;i<y-10;i++){
        leds2[i] = CRGB::Green;
    }
}

int kompresija_1(int x, int y){
    for(i = x;i<y;i++){
        leds2[i] = CRGB::Yellow;
    }
}

int ekspanzija_1(int x, int y){
    for(i = x;i<y;i++){
        leds2[i] = CRGB::Red;
    }
}

int ispuh_1(int x, int y){
    for(i = x;i<y;i++){
        leds2[i] = CRGB::White;
    }
}

```

SAŽETAK

Tema završnog rada „Demonstracijska maketa V8 ICE motora upravljana mikroračunalnim sustavom“ pruža detaljan uvid u izrađenu maketu koja ilustrira rad motora s unutarnjim izgaranjem s osam cilindara. U radu se objašnjava definicija V8 motora i teorijski princip njegovog rada. Budući da je rad uglavnom praktične prirode, teorijski dio opisuje kako će maketa biti izvedena, uz ilustracije u obliku blok dijagrama. Nakon teorijskog dijela, detaljno je opisano kako je maketa izrađena, koji je mikroupravljač korišten te na kojem se principu reguliraju komponente koje pokreću cijelu maketu. Osim sklopovskog rješenja, objašnjeno je kako je pisan programski kod te kojim redoslijedom mikroračunalo upravlja maketom. Rad također pokriva testiranja i rezultate, opisuje način na koji je motor testiran te daje objašnjenja rezultata mjerenja.

Ključne riječi: demonstracijska maketa, mikroupravljač, V8 ICE motor

ABSTRACT

Demonstration Model of a V8 ICE Engine Controlled by a Microcomputer System

The topic of the final thesis "Demonstration Model of a V8 ICE Engine Controlled by a Microcomputer System" provides a detailed insight into the created model that illustrates the operation of an eight-cylinder internal combustion engine. The thesis explains the definition of the V8 engine and the theoretical principle of its operation. Since the work is mostly practical in nature, the theoretical part describes how the model will be constructed, with illustrations in the form of block diagrams. After the theoretical part, the thesis details how the model was constructed, which microcontroller was used, and the principle by which the components driving the entire model are regulated. Besides the hardware solution, it explains how the program code was written and the sequence in which the microcomputer operates the model. The thesis also covers the testing and results, describing how the engine was tested and providing explanations of the measurement results.

Keywords: demonstration model, microcontroller, V8 ICE engine

ŽIVOTOPIS

David Kurtović rođen je u Osijeku, 3. studenoga 1999. Živi i odrasta u Ladimirevcima gdje pohađa Osnovnu školu Ladimirevci. Nakon završenog osnovnoškolskog obrazovanja, upisuje smjer elektrotehničar u ustanovi Srednja škola Valpovo. Nakon završene srednje škole, upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku i opredjeljuje se za preddiplomski stručni studij elektrotehnike, smjer automatika.