

# Ambijentalna LED rasvjeta automobila

---

**Đuranić, Dominik**

**Undergraduate thesis / Završni rad**

**2021**

*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:200:435935>

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

*Download date / Datum preuzimanja:* **2025-03-05**

*Repository / Repozitorij:*

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



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

**Sveučilišni studij**

**AMBIJENTALNA LED RASVJETA AUTOMOBILA**

**Završni rad**

**Dominik Đuranić**

**Osijek, 2021. godina.**

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA  
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 14.09.2021.

Odboru za završne i diplomske ispite

**Prijedlog ocjene završnog rada na  
preddiplomskom sveučilišnom studiju**

Ime i prezime studenta:	Dominik Đuranić
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R4055, 28.07.2017.
OIB studenta:	62257893270
Mentor:	Izv. prof. dr. sc. Tomislav Matić
Sumentor:	
Sumentor iz tvrtke:	
Naslov završnog rada:	Ambijentalna LED rasvjeta automobila
Znanstvena grana rada:	<b>Arhitektura računalnih sustava (zn. polje računarstvo)</b>
Predložena ocjena završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	14.09.2021.
Datum potvrde ocjene Odbora:	22.09.2021.
Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis: Datum:

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

Osijek, 23.09.2021.

**Ime i prezime studenta:**

Dominik Đuranić

**Studij:**

Preddiplomski sveučilišni studij Računarstvo

**Mat. br. studenta, godina upisa:**

R4055, 28.07.2017.

**Turnitin podudaranje [%]:**

3

Ovom izjavom izjavljujem da je rad pod nazivom: **Ambijentalna LED rasvjeta automobila**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Tomislav Matić

i sumentora

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 studenta:

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

**IZJAVA**

**o odobrenju za pohranu i objavu ocjenskog rada**

kojom ja Dominik Đuranić, OIB: 62257893270, student/ica Fakulteta elektrotehnike,

računarstva i informacijskih tehnologija Osijek na studiju Preddiplomski sveučilišni studij Računarstvo, kao autor/ica ocjenskog rada pod naslovom: Ambijentalna LED rasvjeta automobila,

dajem odobrenje da se, bez naknade, trajno pohrani moj ocjenski rad u javno dostupnom digitalnom repozitoriju ustanove Fakulteta elektrotehnike, računarstva i informacijskih tehnologija Osijek i Sveučilišta te u javnoj internetskoj bazi radova Nacionalne i sveučilišne knjižnice u Zagrebu, sukladno obvezi iz odredbe članka 83. stavka 11. *Zakona o znanstvenoj djelatnosti i visokom obrazovanju* (NN 123/03, 198/03, 105/04, 174/04, 02/07, 46/07, 45/09, 63/11, 94/13, 139/13, 101/14, 60/15).

Potvrđujem da je za pohranu dostavljena završna verzija obranjenog i dovršenog ocjenskog rada. Ovom izjavom, kao autor/ica ocjenskog rada dajem odobrenje i da se moj ocjenski rad, bez naknade, trajno javno objavi i besplatno učini dostupnim:

- a) široj javnosti
- b) studentima/icama i djelatnicima/ama ustanove
- c) široj javnosti, ali nakon proteka 6 / 12 / 24 mjeseci (zaokružite odgovarajući broj mjeseci).

*\*U slučaju potrebe dodatnog ograničavanja pristupa Vašem ocjenskom radu, podnosi se obrazloženi zahtjev nadležnom tijelu Ustanove.*

Osijek, 23.09.2021.

(mjesto i datum)

\_\_\_\_\_  
(vlastoručni potpis studenta/ice)

# SADRŽAJ

<b>1. UVOD.....</b>	<b>1</b>
<b>1.1. Zadatak završnog rada .....</b>	<b>1</b>
<b>2. POSTOJEĆA RJEŠENJA ZA MJERENJE BROJA OKRETAJA .....</b>	<b>2</b>
<b>2.1. OBD-II.....</b>	<b>2</b>
2.1.1. Projekt Stanleya Huanga .....	3
2.1.2. ELM327.....	4
<b>2.2. Proizvodi koji koriste signal alternatora .....</b>	<b>5</b>
2.2.1. Eko test.....	5
2.2.2. G-tech pro.....	6
<b>3. MJERENJE BROJA OKRETAJA AUTOMOBILSKIH MOTORA.....</b>	<b>8</b>
<b>4. REALIZACIJA SUSTAVA .....</b>	<b>12</b>
<b>4.1. Sklopovlje .....</b>	<b>12</b>
4.1.1. Pasivni niskopropusni i visokopropusni filtri .....	12
4.1.2. Operacijsko pojačalo LM358 .....	14
4.1.3. Arduino Nano .....	16
4.1.4. WS2812B .....	17
<b>4.2. Filtriranje i pojačanje signala .....</b>	<b>18</b>
<b>4.3. Detekcija i brojanje vrhova .....</b>	<b>19</b>
<b>4.4. Upravljanje RGB trakom .....</b>	<b>22</b>
<b>4.5. Tipkalo i promjena načina osvjetljenja .....</b>	<b>23</b>
<b>4.6. Način osvjetljenja u ovisnosti o frekvenciji okretaja motora .....</b>	<b>26</b>
<b>4.7. Izrada sustava na eksperimentalnoj pločici i testiranje .....</b>	<b>28</b>
<b>5. ZAKLJUČAK.....</b>	<b>33</b>
<b>SAŽETAK.....</b>	<b>35</b>
<b>ABSTRACT .....</b>	<b>36</b>
<b>ŽIVOTOPIS.....</b>	<b>37</b>
<b>PRILOZI.....</b>	<b>38</b>

# 1. UVOD

Tema ovog završnog rada je razvoj mikroupravljačkog sustava za upravljanje unutarnjom rasvjetom automobila koristeći Arduino platformu. U radu se opisuju i uspoređuju različite metode detekcije okretaja motora. Ideja za razvoj sustava je očitavanje broja okretaja automobila bez pristupa senzorima unutar motora, već samo na lako dostupnim kontaktima akumulatora. Očitani se broj okretaja koristi za upravljanje ambijentalnom rasvjetom u obliku RGB trake. Korisniku je pružena mogućnost mijenjanja nekoliko načina osvjetljenja tipkalom. Za očitavanje signala dizajniran je analogni sklop za obradu signala te napisana programska podrška za mikroupravljač.

Ostatak rada organiziran je kako slijedi. U poglavlju 2 predstavljena su postojeća rješenja i metode brojanja okretaja i upravljanja unutarnjom rasvjetom automobila. U poglavlju 3 opisan je izvor signala i metoda određivanja omjera frekvencija promatranog signala i okretaja motora. U poglavlju 4 opisano je korišteno sklopovlje i proces dizajniranja sklopa za filtraciju i pojačanje signala te pojašnjeni pojedini dijelovi programske podrške. Na kraju poglavlja prikazan je sustav na eksperimentalnoj pločici te rezultati testiranja. U poglavlju 5 dan je kratki pregled cijeloga rada te predložena unaprjeđenja sustava.

## 1.1. Zadatak završnog rada

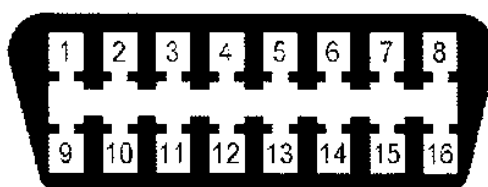
Potrebno je razviti mikroupravljački sustav temeljen na Arduino platformi koji će upravljati unutarnjom rasvjetom automobila na temelju korisničkih postavki i broja okretaja motora.

## 2. POSTOJEĆA RJEŠENJA ZA MJERENJE BROJA OKRETAJA

Postoji nekoliko čestih metoda i proizvoda za mjerenje broja okretaja motora te su neki od njih u ovome poglavlju predstavljani i uspoređeni s Arduino sustavom kojega ovaj rad opisuje. Proizvodi su podijeljeni u dvije skupine u ovisnosti o tome odakle uzimaju podatke o okretajima motora.

### 2.1. OBD-II

OBD (engl. *On-board diagnostics*) je sustav za dijagnostiku automobila napravljen kako bi se korisniku dao pristup podacima kao što su kodovi grešaka DTC (engl. *Diagnostic Trouble Codes*), podaci potrebni za testiranje ispušnih plinova i slično. OBD-II je standardiziran 1996. godine u SAD-u te je standardom obuhvaćen jedinstveni tip konektora s uvijek jednakim rasporedom terminala (Sl. 2.1. ) i njegov položaj u automobilu (udaljen najviše 90cm od vozača) [1].



<b>1 - Prazna</b>	<b>9 - Prazna</b>
<b>2 - J1850 bus</b>	<b>10 - J1850 bus</b>
<b>3 - Prazna</b>	<b>11 - Prazna</b>
<b>4 - Masa/Šasija</b>	<b>12 - Prazna</b>
<b>5 - Masa/Signal</b>	<b>13 - Masa/Signal</b>
<b>6 - CAN High</b>	<b>14 - CAN Low</b>
<b>7 - ISO 9141-2 K-Line</b>	<b>15 - ISO 9141-2 L-Line</b>
<b>8 - Prazna</b>	<b>16 - Bat. 12V</b>

Sl. 2.1. OBD-II konektor i raspored terminala [1].

Također su propisani i protokoli za primanje i slanje podataka, popisi kodova grešaka (DTC) te popisi identifikatora parametara (engl. *Parameter IDs*) koji se koriste kada se šalju upiti za određene informacije o motoru [2]. OBD-II se od 2003. i u Europi zakonski ugrađuje u sva vozila B kategorije, što znači da je danas prisutan u većini automobila na cesti. Tako su za OBD dostupni razni gotovi proizvodi za dijagnostiku, mjerenje i prikaz okretaja motora, kao i hobi *open source* projekti koji koriste Arduino.



### 2.1.1. Projekt Stanleya Huanga

Stanley Huang je na svom forumu „[arduino-dev.com](http://arduino-dev.com)“ početkom 2013. godine objavljivao projekte vezane za Arduino i čitanje podataka preko OBD-II sučelja. Za svoje potrebe je razvio, a kasnije počeo i prodavati OBD-II UART (engl. *Universal Asynchronous Receiver-Transmitter*) adapter koji se na Arduino spaja preko TX i RX pinova [3]. Adapter pruža jednostavno sučelje između CAN (engl. *Controller Area Network*) sabirnice i Arduina ili drugog mikroupravljača, no za Arduino je objavio *open source* biblioteke i primjere za korištenje adaptera [4]. Njegov je adapter jako često korišten u raznim projektima koji koriste Arduino za čitanje podataka preko OBD-a [5]. Jedan od njegovih projekata za prikaz okretaja motora na matričnom pokazniku je vidljiv na slici 2.2., a projekt s ambijentalnom rasvjetom koji koristi njegov adapter na slici 2.3.



Sl. 2.2. Prikaz okretaja motora na matričnom pokazniku (Stanley Huang) [6].



Sl. 2.3. Prikaz okretaja koristeći ambijentalnu rasvjetu (Caleb Rowe) [7].

Adapter je potreban kod spajanja Arduina na OBD jer CAN okviri putuju do svih čvorova mreže te se po identifikatorima filtriraju na odredištu. To znači da bi se bez adaptera i filtracije na Arduinu čitali svi okviri u mreži koji se šalju između ECU (engl. *Engine Control Unit*) i raznih senzora i sustava automobila. Adapter apstrakcijom postiže jako jednostavno slanje i dekodiranje primljenih informacija. Jedan od problema OBD-a je što iako su propisani PID identifikatori parametara, proizvođači nemaju zakonsku obvezu implementirati sve parametre [2]. To znači da informacije koje je moguće dobiti slanjem upita preko OBD-a i dalje ovise o pojedinom modelu automobila.

### 2.1.2. ELM327

ELM327 je najpoznatiji mikroupravljač programiran za prevođenje OBD sučelja u razumljive podatke. To postiže apstrakcijom protokola niske razine (npr. CAN) u jednostavno sučelje kojemu se može pristupiti koristeći UART. Na tržištu je dostupan velik broj proizvoda koji koriste ELM327 te su različite kvalitete i obuhvaćaju veliki cjenovni rang (od \$5 do \$100+).



Sl. 2.4. Primjer ELM327 proizvoda.

Primjer proizvoda je prikazan na slici 2.4. Ti proizvodi pružaju različite načine spajanja:

- Bluetooth
- Wi-fi
- USB
- RS-232.

Također, za svaku platformu je razvijen velik broj aplikacija o kojima ovisi koliko je napredna funkcionalnost uređaja, pruža li se ispis opisa kodova grešaka i slično [8]. Bolje razvijene aplikacije s više funkcionalnosti se često plaćaju. Za usporedbu s ostalim metodama očitavanja okretaja motora potrebno se dotaći frekvencije uzorkovanja podataka. Ona kod ELM327 uređaja ovisi o velikom broju čimbenika, kao što su kvaliteta proizvoda (pogotovo kod bežičnog spajanja), kvaliteta aplikacije koja komunicira s proizvodom, brzina CAN sabirnice (što ovisi o modelu vozila) i drugim. Tako je u testovima proizvoda često brzina uzorkovanja manja od optimalne i dolazi do vidljivog kašnjenja očitavanja broja okretaja motora u stvarnom vremenu [9].

## 2.2. Proizvodi koji koriste signal alternatora

Proizvodi ove skupine koriste isti signal za mjerenje okretaja motora kao i sustav opisan u ovome radu te su bili inspiracija za isti.

### 2.2.1. Eko test

Prvi je uređaj za mjerenje okretaja motora i temperature ulja pri testiranju ispušnih plinova motornih vozila (tzv. eko test), čiji je primjer prikazan na slici 2.5.



Sl. 2.5. Bosch BEA 030 uređaj za mjerenje broja okretaja motora i temperature ulja [10].

Kako piše u priručniku uređaja, broj okretaja se mjeri „pomoću vršne valovitosti akumulatora“, odnosno vrhova napona do kojih dolazi prilikom punjenja akumulatora. Osim tog načina podržano je čitanje preko OBD-a te čitanje s raznih mjesta kao što su senzori bregaste osovine, svitci za

paljenje i tahometri [10]. Međutim, ti su dijelovi u prostoru motora teško dostupni pa se najčešće koriste kontakti akumulatora.



Sl. 2.6. Mjerenje broja okretaja motora preko kontakata akumulatora.

Pri mjerenju preko kontakata akumulatora mjerno je područje uređaja od 450 do 6000 okretaja u minuti, slično mjernome području Arduino sustava koji je tema ovoga rada. Mjerenje na kontaktima akumulatora vidljivo je na slici 2.6. Pošto se uređaj BEA 030 koristi za modele automobila svih proizvođača, potrebna je kalibracija prilikom uključivanja uređaja kada određuje omjer prijenosa generatora na radilicu u praznom hodu. Ako omjer prijenosa nije dobro određen, korisnik preko računala može dodatno podesiti broj impulsa kako bi se dobilo točno očitavanje [10]. Omjer prijenosa nije standardiziran što znači da za svaki model vozila može biti drugačiji. Detaljnije objašnjenje mjerenja broja okretaja i računanja omjera frekvencija slijedi u poglavlju 3.

### **2.2.2. G-tech pro**

G-tech pro je uređaj američke tvrtke Tesla Electronics Inc. koji služi za testiranje performansi automobila kao što su krivulja konjskih snaga ili zakretnog momenta u odnosu na okretaje motora, ubrzanje od 0 do 100km/h i slično [11]. Na web stranici proizvođača stoji da je za aproksimaciju nekih od tih performansi potrebno točno i trenutno mjerenje broja okretaja motora za što je proizvođaču mjerenje koristeći OBD bilo previše sporo [11]. Zato je postignuto očitavanje signala okretaja motora na utičnici za auto upaljač, koja je najčešće spojena izravno na akumulator automobila (zanemarijući releje i osigurače koji ne mijenjaju napon).



Sl. 2.7. G-tech pro [11].

Uređaj je prikazan na slici 2.7. Kao i kod uređaja za eko test potrebna je kalibracija pri prvom mjerenju okretaja te su u priručniku opisani sljedeći koraci:

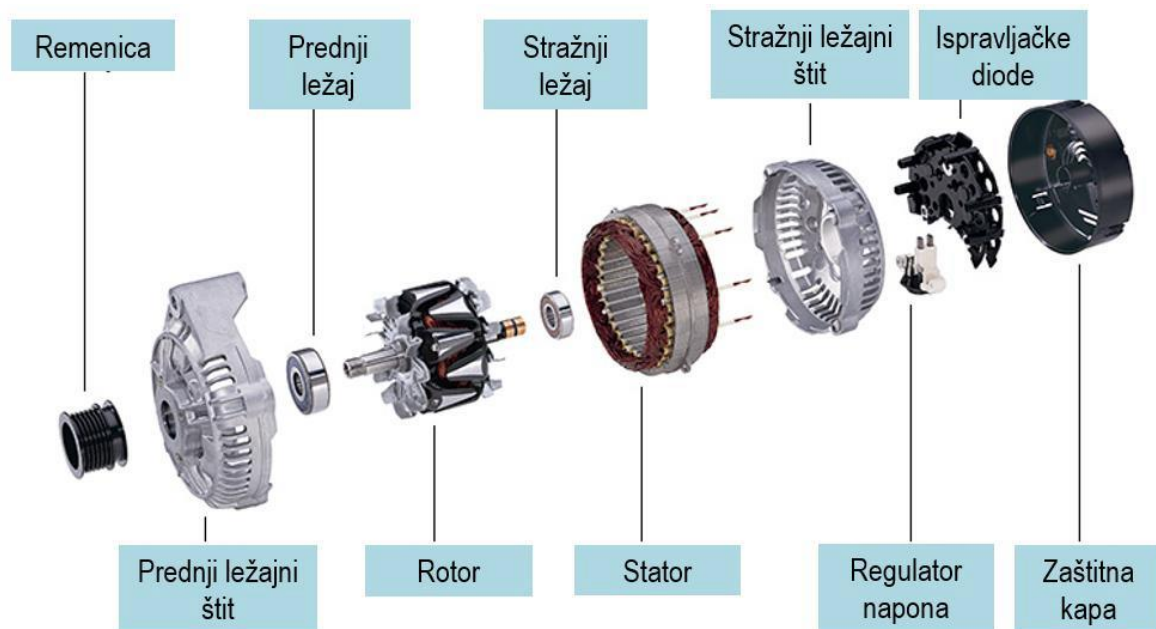
1. u uređaj upisati maksimalan broj okretaja automobila (engl. *redline*)
2. snimiti signal pri najmanjem broju okretaja (engl. *idle*)
3. snimiti signal na 2000 okretaja
4. snimiti signal na 4000 okretaja [12].

Sličan je proces mjerenja korišten za računanje omjera frekvencija signala i okretaja motora pri izradi sustava opisanog u ovome radu te je objašnjen u poglavlju 3.



### 3. MJERENJE BROJA OKRETAJA AUTOMOBILSKIH MOTORA

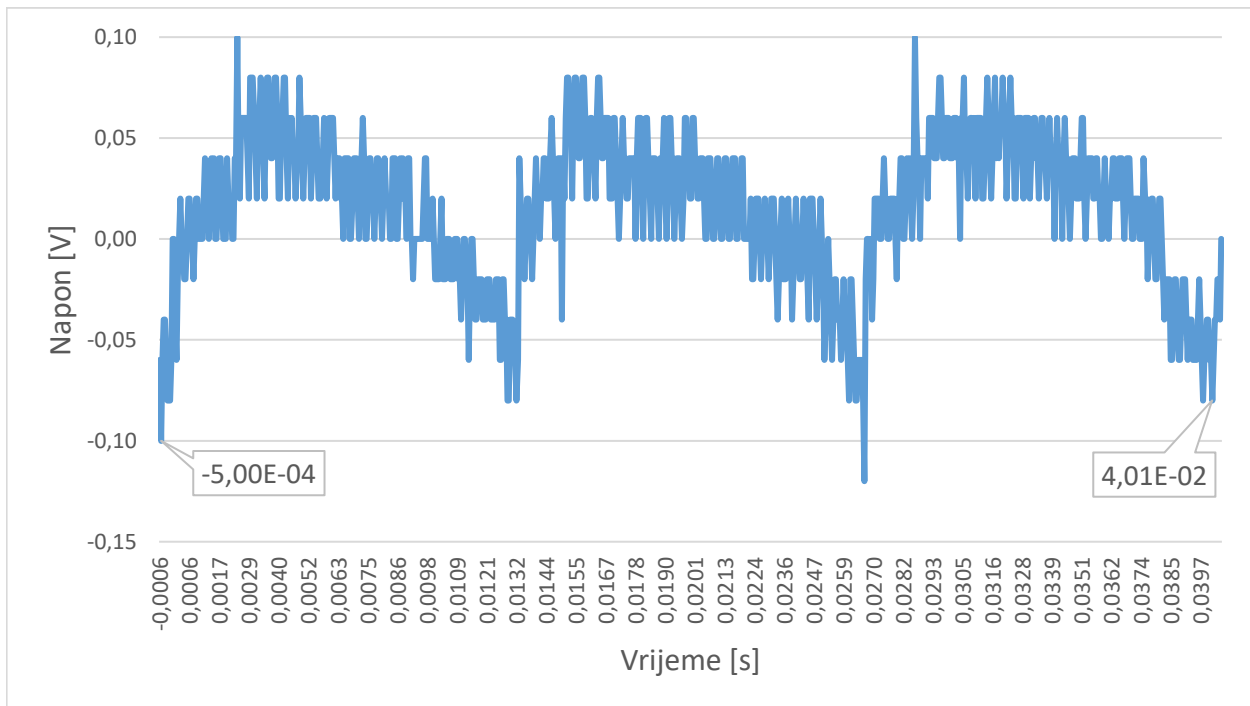
Kao što je u prethodnom poglavlju već navedeno, sustav razvijen za ovaj završni rad za mjerenje broja okretaja motora automobila koristi nadvišenje napona akumulatora. Do nadvišenja napona dolazi zbog punjenja akumulatora prilikom okretanja pogonskog motora automobila. Uređaj koji puni akumulator pri radu je alternator (generator izmjeničnog napona) ili dinamo (generator istosmjernog napona). Prednost alternatora je što su dugotrajniji, jači te već pri najmanjem broju okretaja daju dovoljno struje za punjenje akumulatora [13]. Mana alternatora je što se dobiveni napon mora regulirati i ispraviti u istosmjerni, no pojavom masovno proizvedenih poluvodičkih dioda alternator je skoro u potpunosti zamijenio korištenje dinama u automobilima. Prikaz dijelova alternatora s ugrađenim regulatorom i ispravljačkim diodama je vidljiv na slici 3.1.



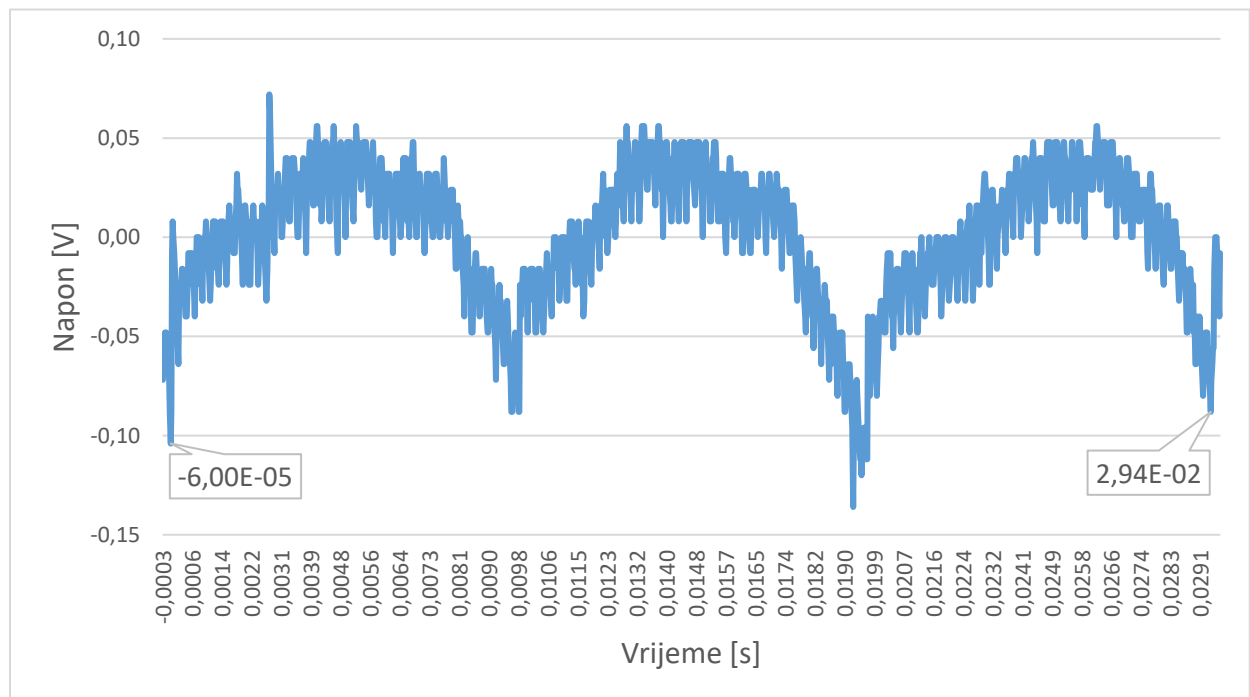
Sl. 3.1. Dijelovi alternatora automobila [14].

Alternator i ostali pomoćni agregati poput kompresora klime i pumpe vode čine sekundarni ili agregatni pogon. Pomoćne agregate pogoni koljenasto vratilo najčešće preko kanalnog remena. Omjer promjera remenice koljenastog vratila i remenice alternatora određuje omjer brzine njihove vrtnje. Pošto se za brojanje okretaja koriste vrhovi napona koji su nastali ispravljanjem izmjeničnog napona alternatora, za određivanje omjera frekvencije vrtnje motora i frekvencije dobivenog signala bilo bi potrebno znati točne promjere dvaju remenica te konstrukciju alternatora (broj faza i način ispravljanja). Te je informacije teško dobiti za svaki model automobila pa je za određivanja omjera frekvencija korišten proces snimanja signala osciloskopom pri određenom broju okretaja. Tako je moguće računski odrediti omjer frekvencije okretaja motora i frekvencije

signala sa kontakata akumulatora. Na slikama 3.2. i 3.3. su prikazani signali snimljeni osciloskopom na kontaktima akumulatora na modelu automobila Fiat Grande Punto. Snimljena je samo izmjenična komponenta signala, a u stvarnosti signal sadrži i istosmjerni napon od oko 13V.



Sl. 3.2. Izmjenična komponenta signala pri 2300 okretaja u minuti.



Sl. 3.3. Izmjenična komponenta signala pri 3200 okretaja u minuti.

Iz grafova sa slika 3.2. i 3.3. očitani su periodi izmjerenih signala. Na grafovima su oblačićima označene vrijednosti vremenske osi u točkama, nazovimo lijevu  $X_1$  a desnu  $X_2$ , između kojih je trajanje 3 perioda signala. Period signala pri 2300 okretaja/min  $T_1$  se dobije izrazom (3-1)

$$T_1 = \frac{X_2 - X_1}{3} = \frac{4,01 \times 10^{-2} - (-5,00 \times 10^{-4})}{3} = 1,35 \times 10^{-2} s = 13,5 ms \quad (3-1)$$

Period signala pri 3200 okretaja/min  $T_2$  se dobije izrazom (3-2)

$$T_2 = \frac{X_2 - X_1}{3} = \frac{2,94 \times 10^{-2} - (-6,00 \times 10^{-5})}{3} = 9,82 \times 10^{-3} s = 9,82 ms \quad (3-2)$$

Frekvencija se računa po formuli  $f=1/T$  pa se tako dobije za frekvenciju prvog signala  $f_1=74,07\text{Hz}$ , a frekvenciju drugog signala  $f_2=101,83\text{Hz}$ . Za prvi signal se omjer frekvencije okretaja motora i frekvencije signala  $K_1$  tada računa prema izrazu (3-3)

$$K_1 = \frac{f_{m1}}{f_1} = \frac{2300 \text{ min}^{-1}}{74,07 \text{ s}^{-1}} = \frac{2300 \frac{1}{60s}}{74,07 \times \frac{1}{s}} = \frac{38,33 \times \frac{1}{s}}{74,07 \times \frac{1}{s}} = 0,5175 \quad (3-3)$$

gdje je  $f_{m1}$  frekvencija okretaja motora od 2300 okretaja u minuti.

Za drugi signal se omjer frekvencija  $K_2$  računa prema izrazu (3-4)

$$K_2 = \frac{f_{m2}}{f_2} = \frac{3200 \text{ min}^{-1}}{101,83 \text{ s}^{-1}} = \frac{3200 \frac{1}{60s}}{101,83 \frac{1}{s}} = \frac{53,33 \frac{1}{s}}{101,83 \frac{1}{s}} = 0,5237 \quad (3-4)$$

gdje je  $f_{m2}$  frekvencija okretaja motora od 3200 okretaja u minuti.

Iz dobivenih omjera frekvencija  $K_1$  i  $K_2$  moguće je dobiti srednju vrijednost omjera frekvencija  $K$  koja iznosi  $K=0,5206$ . Srednji omjer frekvencija  $K$  je zapravo koeficijent pomoću kojega je iz frekvencije izmjerenog signala moguće dobiti nepoznatu frekvenciju okretaja motora. Primjer izračuna frekvencije okretaja za izmjerenu frekvenciju signala  $f=195\text{Hz}$  prikazan je izrazom (3-5)

$$f_m = f \times K = 195\text{Hz} \times 0,5206 = 101,517 \frac{1}{s} = 101,517 \frac{1}{\frac{1}{60} \text{ min}} = 6091,02 \text{ min}^{-1} \quad (3-5)$$

gdje je  $f_m$  nepoznata frekvencija okretaja motora čija izračunata vrijednost iznosi 6091,02 okretaja u minuti. Koeficijent  $K$  je bezdimenzijska konstanta jer predstavlja omjer frekvencija. Kako bi se



u konstantu dodala i pretvorba frekvencije iz Hz ( $s^{-1}$ ) u  $min^{-1}$  potrebno je  $K$  pomnožiti sa 60 i dodati mjernu jedinicu s/min. Konstanta za pretvorbu je označena  $K_p$  te je izvod dan izrazom (3-6)

$$K_p = K \times \frac{60s}{min} = 0,5206 \times 60 \frac{s}{min} = 31,236 \frac{s}{min} \quad (3-6)$$

Izračun nepoznate frekvencije okretaja motora  $f_m$  uz frekvenciju signala  $f=195Hz$  koristeći  $K_p$  prikazan je izrazom (3-7)

$$f_m = f \times K_p = 195 \frac{1}{s} \times 31,236 \frac{s}{min} = 6091,02 \text{ min}^{-1} \quad (3-7)$$

## 4. REALIZACIJA SUSTAVA

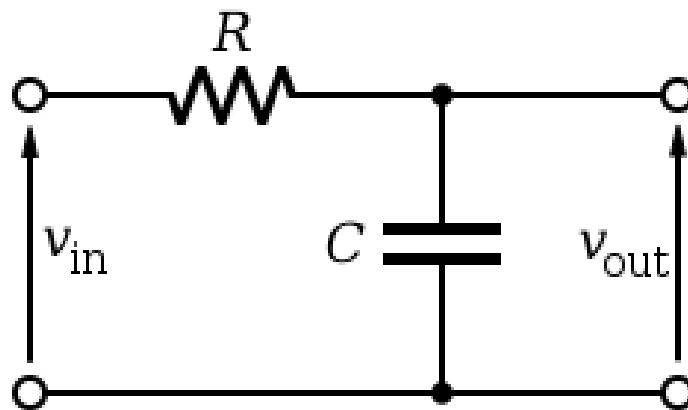
Sustav se sastoji od 2 glavna dijela: analognog i digitalnog. U analognome dijelu korištene su aktivne i pasivne komponente u svrhu filtracije i pojačanja signala, a u digitalnome dijelu se koristi Arduino Nano te RGB (engl. *Red, Green, Blue*) traka kojom je moguće upravljati mikrokontrolerom. Signal se filtrira koristeći pasivne niskopropusne i visokopropusne filtre, a pojačava operacijskim pojačalom LM358. U idućem potpoglavlju su pobliže objašnjene korištene komponente.

### 4.1. Sklopovlje

#### 4.1.1. Pasivni niskopropusni i visokopropusni filtri

Pasivne su elektroničke komponente otpornici (R), kondenzatori (C) i zavojnice (L). Njihovim spajanjem na određeni način moguće je dobiti niskopropusni ili visokopropusni filter. U ovome su radu korišteni RC niskopropusni filtri za filtraciju šuma visoke frekvencije te niskopropusni CR filter za filtraciju šuma niske frekvencije.

RC niskopropusni filter propušta signale frekvencije manje od granične frekvencije  $f_g$ , a guši signale frekvencije veće od  $f_g$ . On se realizira spajanjem otpornika R i kondenzatora C u seriju. Izlazni signal filtra jednak je padu napona na kondenzatoru [15]. Spoj je prikazan na slici 4.1.



Sl. 4.1. Shema RC niskopropusnog filtra.

Pošto kapacitivni otpor  $R_c$  ovisi o frekvenciji signala, ovaj spoj se ponaša kao naponsko djelilo koje ovisi o frekvenciji signala. Ovisnost kapacitivnog otpora o frekvenciji je dana izrazom (4-1)

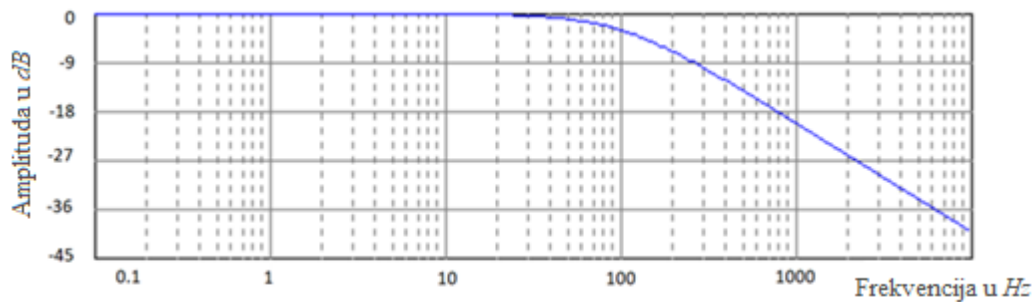
$$R_c = \frac{1}{2\pi f C} \quad (4-1)$$

gdje je  $f$  frekvencija signala, a  $C$  kapacitet kondenzatora.

Tada je izlazni napon spoja dan izrazom (4-2)

$$v_{out} = v_{in} \times \frac{R_c}{\sqrt{R^2 + R_c^2}} \quad (4-2)$$

Prema izrazima (4-1) i (4-2) vidljivo je da izlazni napon pada kada frekvencija raste jer se smanjuje kapacitivni otpor  $R_c$ . To znači da filter guši signale frekvencije veće od granične. Amplitudno frekvencijska karakteristika RC filtra vidljiva je na slici 4.2.

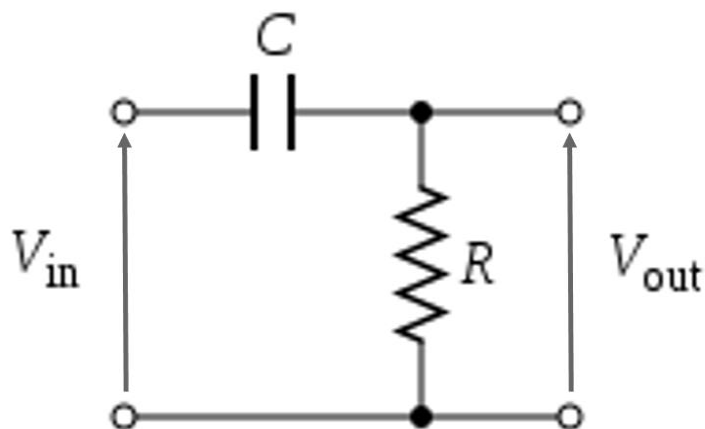


Sl. 4.2. Amplitudno frekvencijska karakteristika RC filtra s graničnom frekvencijom  $f_g = 100\text{Hz}$  [15].

Na slici je vidljivo da amplituda počinje naglo padati nakon određene granične frekvencije. Granična se frekvenciju RC filtra računa prema izrazu (4-3)

$$f_g = \frac{1}{2\pi RC} \quad (4-3)$$

CR visokopropusni filter propušta signale frekvencije veće od granične frekvencije  $f_g$ , a guši signale frekvencije manje od granične [15]. On se realizira obrnutim redoslijedom spajanja otpornika i kondenzatora u odnosu na RC filter. Spoj je prikazan slikom 4.3.

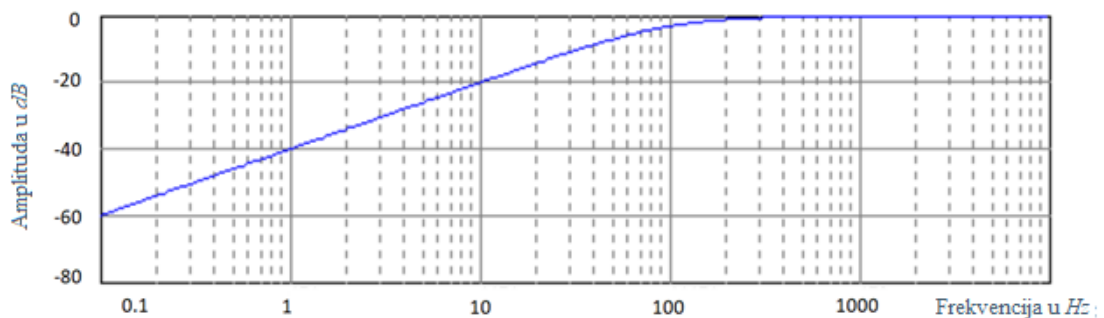


Sl. 4.3. Shema CR visokopropusnog filtra.

On je također djelilo napona, no izlazni napon je u ovom slučaju pad napona na otporniku. Izlazni napon se računa prema izrazu (4-4)

$$v_{out} = v_{in} \times \frac{R}{\sqrt{R^2 + R_c^2}} \quad (4-4)$$

Iz izraza (4-1) i (4-4) je vidljivo da izlazni napon pada s padom frekvencije signala jer se povećava kapacitivni otpor  $R_c$ . Amplitudno frekvencijska karakteristika CR filtra je vidljiva na slici 4.4.

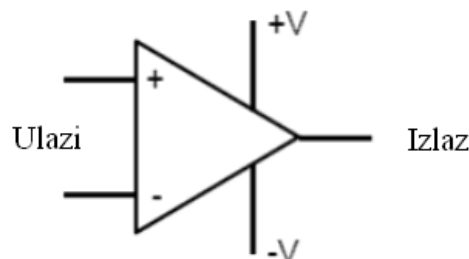


Sl. 4.4. Amplitudno frekvencijska karakteristika CR filtra s graničnom frekvencijom  $f_g = 100\text{Hz}$  [15].

Granična frekvencija se računa jednako kao kod RC filtra prema izrazu (4-3). Još jedno svojstvo CR filtra je da ne propušta istosmjerni napon jer kondenzator u spoju prethodi izlaznom naponu.

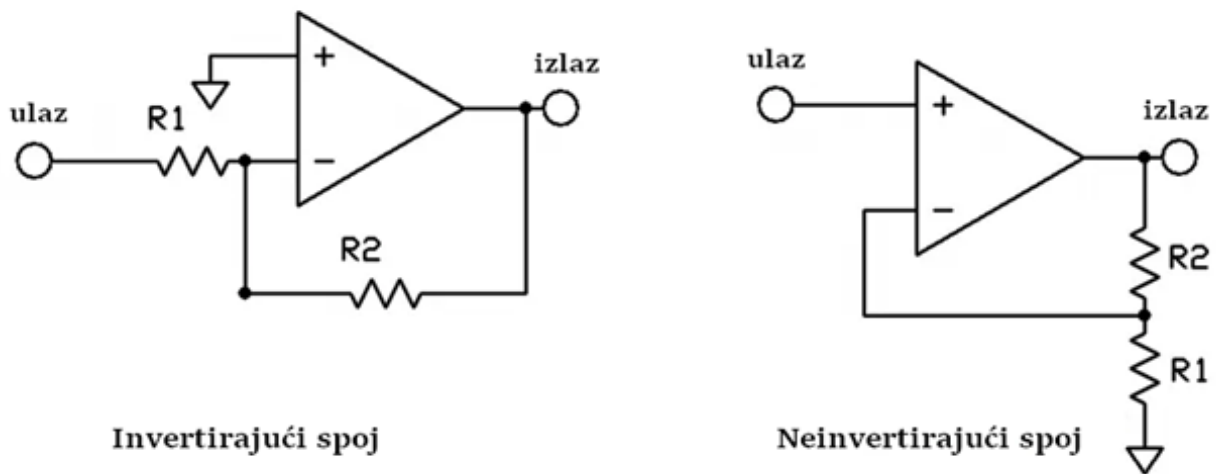
#### 4.1.2. Operacijsko pojačalo LM358

Operacijsko pojačalo je aktivna elektronička komponenta s dva ulaza, invertirajući i neinvertirajući, i jednim izlazom. Neinvertirajući ulaz se označava s „+“, a invertirajući s „-“. Također mu je potrebno napajanje na terminalima označenim „+V“ i „-V“ jer pojačalo može dati izlazni signal samo u rasponu napona između pozitivnog i negativnog napajanja. Ono se u elektronici koristi kao pojačalo, komparator, oscilator i u velikom broju drugih spojeva. Simbol je prikazan na slici 4.5.



Sl. 4.5. Simbol operacijskog pojačala.

Najčešće se spajaju u invertirajući i neinvertirajući spoj [16]. Sheme spojeva su prikazane na slici 4.6.



Sl. 4.6. Shema invertirajućeg i neinvertirajućeg spoja operacijskog pojačala.

Svojstvo invertirajućeg spoja je da je izlazni istosmjerni napon suprotnog predznaka ulaznom naponu, odnosno kod izmjeničnih signala je faza signala pomaknuta za 180°. Kod invertirajućeg spoja je ulazna impedancija mala, dok je kod neinvertirajućeg spoja impedancija velika. Još jedna od razlika je što kod invertirajućeg spoja pojačanje  $A_v$  može biti manje od 1, odnosno može se koristiti za smanjenje amplitude signala. To je vidljivo prema izrazu (4-5) koji definira računanje pojačanja invertirajućeg spoja.

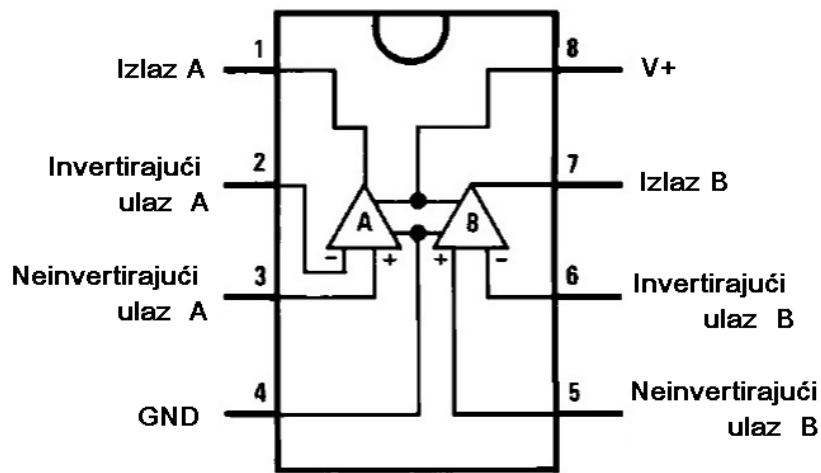
$$A_v = -\frac{R_2}{R_1} \quad (4-5)$$

Kod neinvertirajućeg spoja je pojačanje uvijek veće od 1 što je prikazano izrazom (4-6)

$$A_v = 1 + \frac{R_2}{R_1} \quad (4-6)$$

Izlazni napon operacijskog pojačala računa se prema izrazu (4-7)

$$v_{out} = A_v * v_{in} \quad (4-7)$$

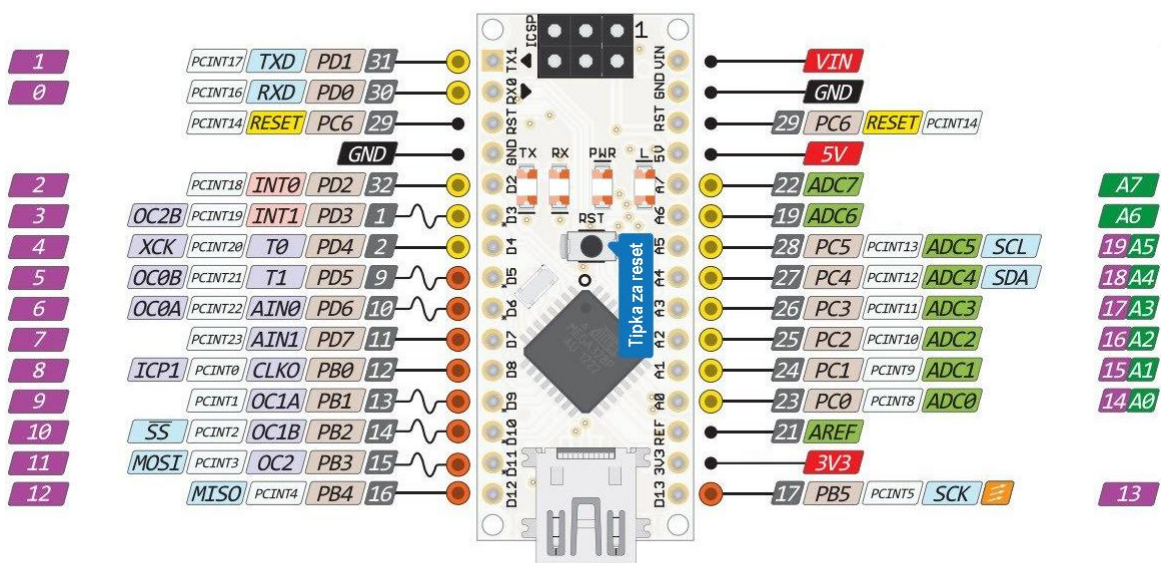


Sl. 4.7. LM358 raspored pinova.

LM358 je integrirani krug s dva operacijska pojačala niske struje i naponskog raspona od 3V do 32V [17]. U sustavu razvijenom za ovaj rad se koristi pojačalo A sklopa LM358 u neinvertirajućem spoju. Raspored pinova operacijskog pojačala LM358 je vidljiv na slici 4.7.

### 4.1.3. Arduino Nano

Arduino Nano je razvojni sustav baziran na ATmega328P mikroupravljaču. U ovome radu je izabran zbog svojih malih dimenzija i mogućnosti ubadanja izravno u eksperimentalnu pločicu (engl. *breadboard*). Frekvencija takta mu je 16MHz. On ima 22 digitalna pina, od kojih 6 mogu biti analogni ulazi. 2 pina, A6 i A7 koriste se samo kao analogni ulazi [18]. Raspored pinova je vidljiv na slici 4.8.



Sl. 4.8. Arduino Nano raspored pinova.

Kôd za Arduino platformu se piše u programskom jeziku C++ s dodatnim ugrađenim metodama i funkcijama. Kôd je organiziran u posebne funkcije *setup()* i *loop()*. Kôd unutar *setup()* funkcije se izvodi jednom prilikom pokretanja Arduina, dok se kôd unutar *loop()* funkcije stalno izvodi tijekom rada. U ovome radu se Arduino koristi za brojanje vrhova analognog signala, računanje broja okretaja motora te upravljanje LED (engl. *Light-Emitting Diode*) trakom.

#### 4.1.4. WS2812B

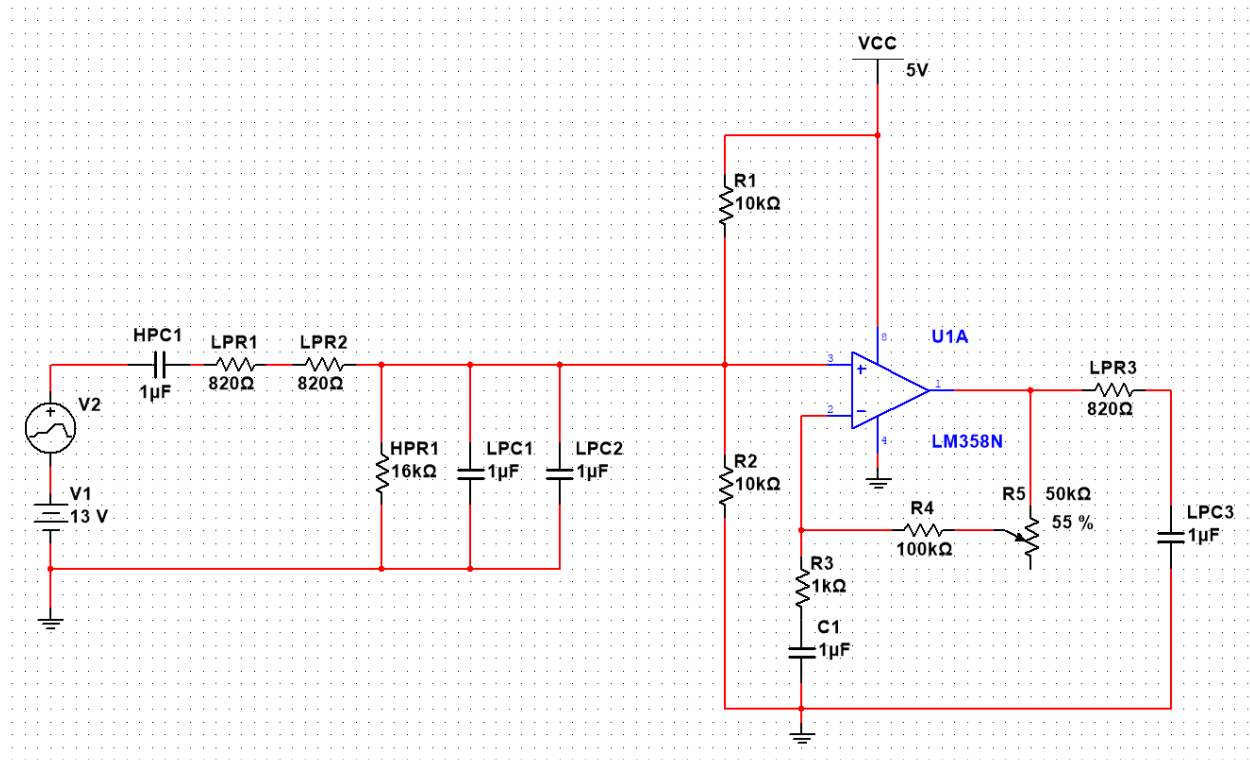
WS2812B je sklop sastavljen od RGB dioda s ugrađenim kontrolerom koji omogućava pojedinačno adresiranje svake diode kada su kaskadno spojene jedna za drugom. RGB diode i kontroler su upakirani u standardni SMD 5050 paket. Svaka ima 4 terminala: +5V, GND,  $D_{in}$ ,  $D_{out}$ . Kada su spojene u kaskadu  $D_{out}$  prve mora biti spojen na  $D_{in}$  druge. Time je omogućeno iznimno lagano skaliranje uz adresiranje preko samo jednog podatkovnog terminala. Zbog toga je na tržištu dostupan velik broj raznih proizvoda koji koriste WS2812B poput traka, prstena, matrica i drugih. WS2812B se još popularno zove NeoPixel kako je tvrtka Adafruit nazvala svoje proizvode bazirane na WS2812B. Na slici 4.9. je prikazana LED traka s WS2812B.



Sl. 4.9. WS2812B LED traka.

## 4.2. Filtriranje i pojačanje signala

Signal snimljen i prikazan na slikama 3.2. i 3.3. je u programu Multisim uvezen u generator izmjeničnog napona (V2) te je s njim u seriju spojen izvor istosmjernog napona (V1) od 13V kao što je vidljivo na slici 4.10.

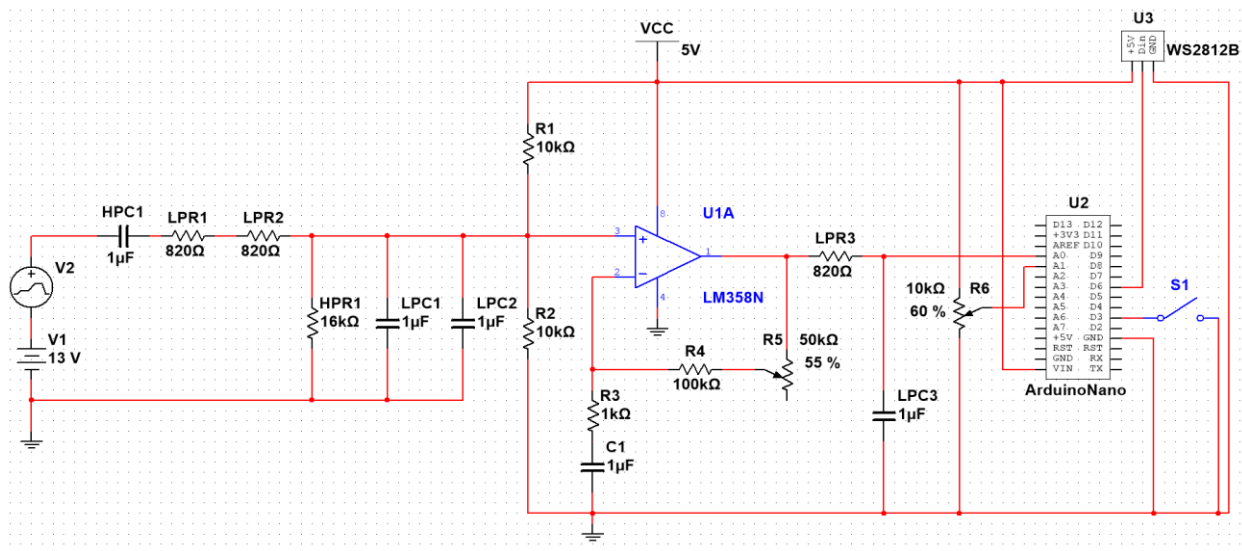


Sl. 4.10. Shema sklopa za pojačanje i filtraciju signala.

Pomoću ta dva izvora je modeliran signal koji se dobije na kontaktima akumulatora. Potreban je istosmjerni izvor V1 jer je snimljeni i uvezeni signal samo izmjenična komponenta signala. Prije ulaza u pojačalo korištena su tri filtra. Prvi je filtar visokopropusni CR filtar s vrijednosti otpora  $16\text{k}\Omega$  i kondenzatorom kapaciteta  $1\mu\text{F}$ . Te vrijednosti prema izrazu (4-3) daju graničnu frekvenciju od oko 10Hz. Time je uklonjen šum frekvencije niže od 10Hz, no bitnije uklonjena je istosmjerna komponenta signala jer istosmjerni napon od 13V ne može proći kroz kondenzator HPC1. Zatim slijede dva niskopropusna RC filtra, što se još naziva filtrom drugog reda, s vrijednostima otpora  $820\Omega$  i kapaciteta  $1\mu\text{F}$ . Time je oslabljen šum frekvencije više od 195Hz. Ta je brojka odabrana jer 195Hz pomnožen koeficijentom  $K_p$ , koji je definiran izrazom (3-6), odgovara broju okretaja od oko 6100 okretaja u minuti. Tako je dobiven pojasno propusni filtar od 10 do 195Hz, što nam daje područje signala koje odgovara frekvenciji okretaja od 310 do 6100 okretaja u minuti. Signal je zatim *pull-up* i *pull-down* otpornicima R1 i R2 u cjelosti doveden u pozitivan napon jer je pojačalo spojeno samo sa jednim napajanjem, odnosno druga nožica napajanja je uzemljena. To znači da



pojačalo ne bi moglo pojačati negativni dio signala, odnosno došlo bi do odsjecanja (engl. *clipping*) signala. Pojačalo je spojeno u neinvertirajućem spoju. U povratnoj vezi je pored otpornika od  $100\text{k}\Omega$  dodan u seriju potenciometar od  $50\text{k}\Omega$  radi lakšeg podešavanja pojačanja. Tako podešeno pojačalo prema izrazu (4-6) daje pojačanje od 101 do 151 ovisno o otporu potenciometra, što je praktično za testiranje na različitim automobilima ako je signal jači ili slabiji od izmjerenoga. Simulacijom je određen otpor potenciometra od oko  $30\text{k}\Omega$  pri kojem je signal dovoljno pojačan za precizno očitavanje na Arduinou, a ne dolazi do odsjecanja vrhova signala na  $3,5\text{V}$ . Do odsjecanja bi došlo pri većem pojačanju jer operacijsko pojačalo LM358 ima odstupanje maksimalnog izlaznog napona od oko  $1,5\text{V}$  u odnosu na napon napajanja koji je  $5\text{V}$ . Na izlazu iz pojačala je dodan još jedan niskopropusni filterar granične frekvencije od  $195\text{Hz}$  kako bi se dodatno filtrirao visoki šum signala te moguća interferencija na ulazu u pojačalo jer se koristi veliko pojačanje. Pojačan i filtriran signal se dovodi na ulaz A0 Arduinoa što je vidljivo na slici 4.11.



Sl. 4.11. Shema cijelog sustava.

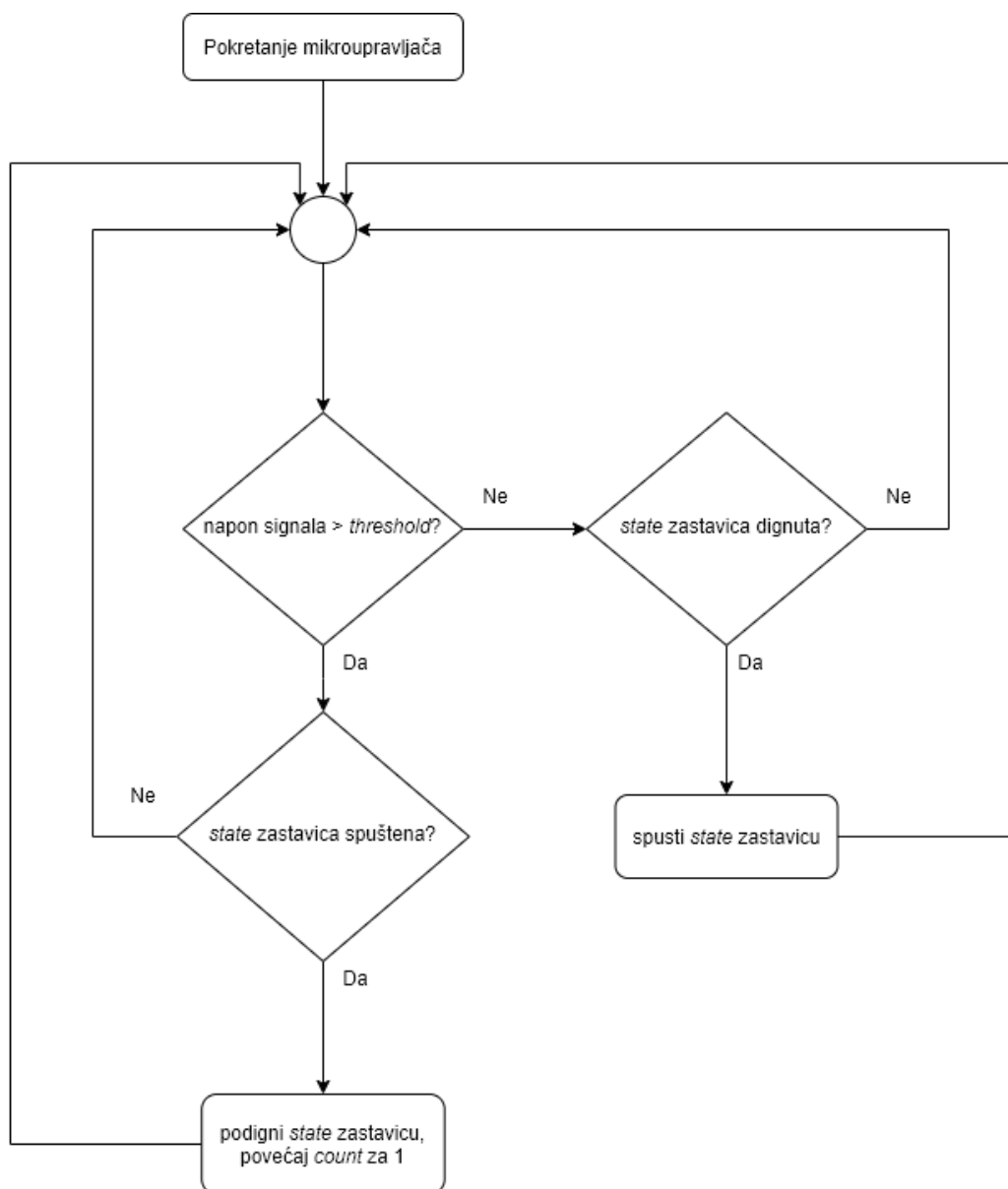
### 4.3. Detekcija i brojanje vrhova

Detekcija vrhova vrši se u *loop()* funkciji koja je maksimalno olakšana radi što veće frekvencije uzorkovanja. To je postignuto tako što je sav ostali kôd napisan unutar *interrupt* funkcija, a u *loop()* funkciji je pored kôda za detekciju i brojanje vrhova ostao samo kôd za promjenu načina osvjetljenja prikazan i objašnjen u potpoglavlju 4.5. U *threshold* varijablu će se pohraniti vrijednost napona očitano na analognom ulazu A1 prilikom pokretanja Arduinoa u *setup()* funkciji. Na ulaz A1 je spojen potenciometar na način prikazan slikom 4.11. Takvim spojem se dobije djelio napona te će napon na ulazu A1 biti jednak vrijednosti napona pohranjenoj u *threshold* varijabli.

**Linija**    **Kôd**

```
111: void loop() {  
112:     if (analogRead(A0) > threshold) {  
113:         if (state == false) {  
114:             state = true;  
115:             count++;  
116:         }  
117:     }  
118:     else {  
119:         if (state == true) {  
120:             state = false;  
121:         }  
122:     }  
}
```

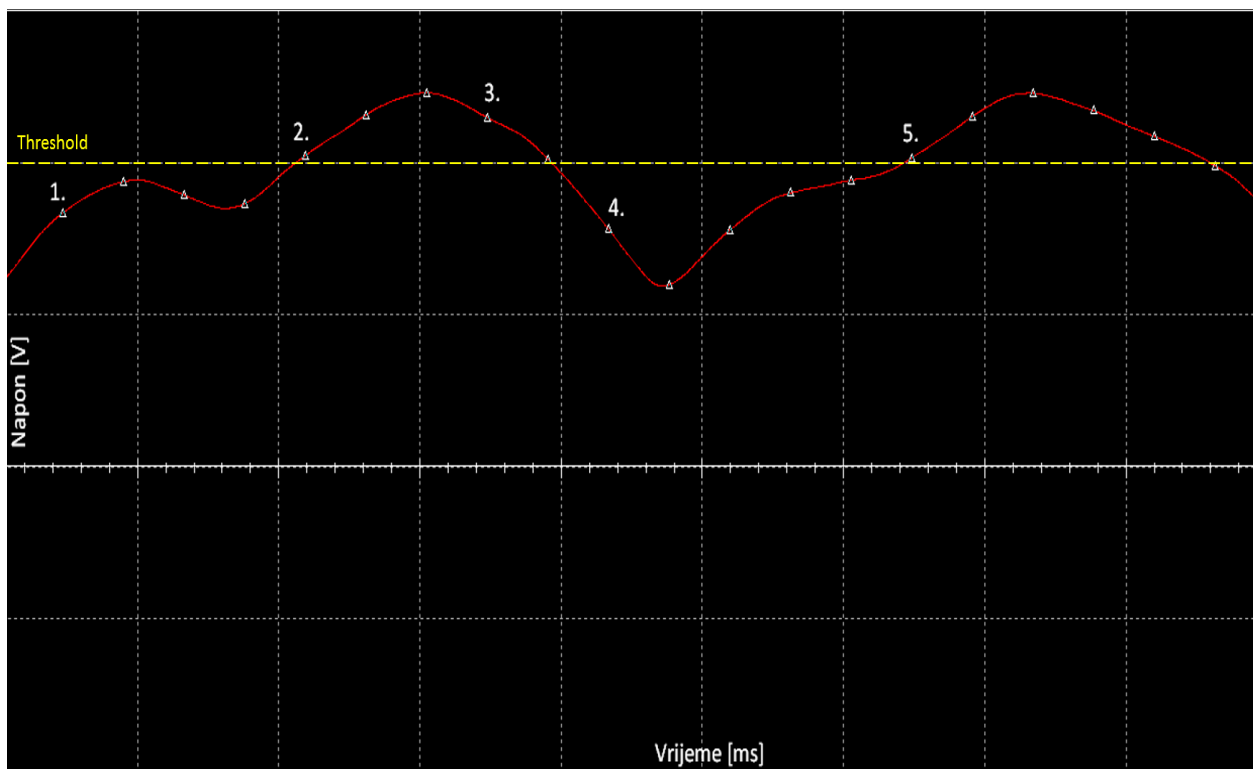
Sl. 4.12. Kôd za brojanje vrhova.



Sl. 4.13. Dijagram toka za proces brojanja vrhova.

Kôd na slici 4.12. pojašnjen je dijagramom toka sa slike 4.13. On služi tome da se napon signala može uzorkovati velikom frekvencijom, no da vrh bude izbrojan samo jednom. To je postignuto korištenjem zastavice *state*. Podizanje zastavice znači da se njena vrijednost postavi na vrijednost *true*, dok spuštanje zastavice znači postavljanje njene vrijednosti na vrijednost *false*. Iz dijagrama toka sa slike 4.13. vidljivo je da zastavica *state* onemogućava ponovno brojanje istog vrha. To je postignuto njenim podizanjem nakon što je uzorkovani napon veći od vrijednosti varijable *threshold*. Samo taj puta kada se podiže zastavica povećava se broj vrhova koji je spremljen u varijabli *count*. Zastavica *state* se ne spušta sve dok je uzorkovani napon signala veći od vrijednosti varijable *threshold*. Dok je zastavica *state* dignuta neće se ponovno povećavati vrijednost varijable *count* iako je uzorkovani napon veći od vrijednosti varijable *threshold*.

Radi lakšeg razumijevanja kôda vrijednosti varijabli prikazane su na primjeru.



Sl. 4.14. Primjer signala na ulazu A0 Arduina (rezultat simulacije).

Na slici 4.14. je primjer izlaza iz sklopa za pojačanje odnosno signala koji će Arduino čitati na ulazu A0. Trokutićima su označene točke kada je Arduino uzorkovao signal, a brojevima koraci u kojima se promatraju vrijednosti varijabli:

1. `count=0`  
`state=false`  
`analogRead(A0)>threshold` uvjet je neistinit pa program skače u naredbu *else*  
`state==true` uvjet je neistinit, zastavica *state* ostaje spuštена
2. `count=0`  
`state=false`  
`analogRead(A0)>threshold` uvjet je istinit pa program ulazi u naredbu *if*  
`state==false` uvjet je istinit, *state* zastavica se podiže i *count* se povećava za 1
3. `count=1`  
`state=true`  
`analogRead(A0)>threshold` uvjet je istinit pa program ulazi u naredbu *if*  
`state==false` uvjet je neistinit pa se u njega ne ulazi, *count* ostaje 1 i zastavica *state* dignuta
4. `count=1`  
`state=true`  
`analogRead(A0)>threshold` uvjet je neistinit pa program skače u naredbu *else*  
`state==true` uvjet je istinit, zastavica *state* se spušta te je brojanje ponovno omogućeno
5. `count=1`  
`state=false`  
`analogRead(A0)>threshold` uvjet je istinit pa program ulazi u naredbu *if*  
`state==false` uvjet je istinit, *state* zastavica se podiže i *count* se povećava za 1

Korak 5 je jednak koraku 2 osim povećane vrijednosti varijable *count* te se ciklus ponavlja za svaki idući vrh.

#### 4.4. Upravljanje RGB trakom

Arduino upravljanje RGB WS2812B trakom je jako jednostavno zbog biblioteka poput korištene *Adafruit\_NeoPixel* biblioteke [19].

<i>Linija</i>	<i>Kôd</i>
101:	<code>void colorWipe(uint32_t color, int duration) {</code>
102:	<code>  for(int i=0; i&lt;NUMPIXELS; i++) {</code>
103:	<code>    strip.setPixelColor(i, color);</code>
104:	<code>    strip.show();</code>
105:	<code>    delay(duration);</code>
106:	<code>  }</code>
107:	<code>}</code>

Sl. 4.15. Funkcija *colorWipe*.

<i>Linija</i>	<i>Kôd</i>
5:	<code>#include &lt;Adafruit_NeoPixel.h&gt;</code>
6:	<code>#define PIN 6</code>
7:	<code>#define NUMPIXELS 5</code>
8:	<code>Adafruit_NeoPixel strip(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);</code>
9:	<code>uint32_t red = strip.Color(255, 0, 0);</code>
10:	<code>uint32_t green = strip.Color(0, 255, 0);</code>
11:	<code>uint32_t blue = strip.Color(0, 0, 255);</code>
12:	<code>uint32_t white = strip.Color(127, 127, 127);</code>
13:	<code>volatile uint32_t RPMcolor;</code>

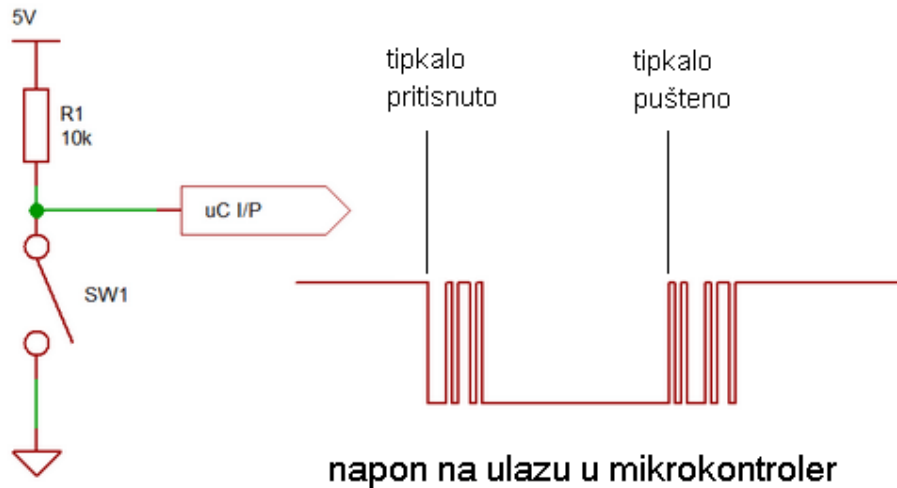
Sl. 4.16. Definicije varijabli i dodavanje biblioteke *Adafruit\_NeoPixel*.

Na slici 4.16. su prikazane definicije varijabli i konstanti korištenih za upravljanje RGB trakom te dodavanje biblioteke *Adafruit\_NeoPixel* u datoteku. U liniji 8 se kreira objekt klase *Adafruit\_NeoPixel* koji predstavlja RGB traku. U konstruktoru su navedeni parametri potrebni za kreiranje i pravilan rad trake kao broj RGB dioda, pin na koji je spojen  $D_{in}$  trake te način kodiranja boja (*GRB* u slučaju korištene trake) i brzinu prijenosa podataka. Također, definirane su boje koje će biti korištene u različitim načinima osvjetljenja. Na slici 4.15. prikazana je funkcija *colorWipe* koja se koristi prilikom promjene načina osvjetljenja kako bi se dobio zanimljiv prijelaz boja. Pri njenom pozivu moraju se predati varijabla s bojom koja će biti postavljena na sve diode trake te trajanje kašnjenja između promjene boja. Promjena se u funkciji postiže *for* petljom čiji iterator kreće od 0 i završava s vrijednosti za jedan manjom od broja dioda. U svakoj iteraciji se na objektu *strip* pozivaju metoda za postavljanje boja *setPixelColor*, kojoj je potrebno predati indeks diode kojoj se postavlja boja i boju u koju se postavlja, te metoda za prikazivanje trenutno postavljenog stanja u objektu na stvarnoj traci *show()*. Na kraju svakog prolaza petljom poziva se ugrađena funkcija *delay* s vrijednosti trajanja kašnjenja u milisekundama. Tako se postigne mijenjanje boja dioda jedne po jedne i dojam animacije.

## 4.5. Tipkalo i promjena načina osvjetljenja

Tipkalo (engl. *button*) se u ovome sustavu koristi za promjenu načina osvjetljenja. Kada se Arduino uključi bude postavljen u način osvjetljenja koje ovisi o izračunatom broju okretaja motora te se na pritisak tipkala ciklično izmjenjuju načini osvjetljenja. Tipkalo je spojeno na Arduino i na uzemljenje te se pritiskom tipkala dobije nula na digitalnom ulazu 3. Način spajanja tipkala S1 je vidljiv na slici 4.11. Pošto je tipkalo spojeno na uzemljenje, u Arduino kôdu treba odabrati *pinMode INPUT\_PULLUP* koji uključuje ugrađeni *pull-up* otpornik. To znači da je vrijednost ulaza D3 1 sve dok se ne pritisne tipkalo kada vrijednost pada u 0. Detekcija pritiska tipkala je uklonjena iz *loop()* funkcije jer bi pozivanje funkcije *digitalRead* u svakom prolazu

*loop()* funkcije usporilo izvođenje pa tako i frekvenciju uzorkovanja signala na analognom ulazu A0. Tako je detekcija pritiska tipkala realizirana *interruptom* na D3 koji se okida na padajući brid, vidljivo na slici 4.18. Međutim, kod svakog tipkala dolazi do pojave skakanja (engl. *bouncing*) napona u fazi promjene između logičke jedinice i logičke nule. Ta je pojava prikazana na slici 4.17.



Sl. 4.17. Pojava skakanja napona pri pritisku tipkala.

Taj se problem jednostavno rješava unutar kôda korištenjem kratke odgode (oko 20ms) nakon koje se provjerava logičko stanje tipkala. Međutim, u ovom je slučaju na ulaz tipkala postavljen *interrupt* unutar kojega se ne može koristiti *delay* funkcija Arduina. Radi toga se mora koristiti zastavica.

<i>Linija</i>	<i>Kôd</i>
27:	<code>void setup() {</code>
28:	<code>  Timer1.initialize(250000);</code>
29:	<code>  Timer1.attachInterrupt( calculateRPM );</code>
30:	<code>  pinMode( D3, INPUT_PULLUP );</code>
31:	<code>  attachInterrupt( digitalPinToInterrupt( D3 ), buttonInterrupt,</code>
	<code>  FALLING );</code>

Sl. 4.18. Postavljanje *interrupta* i načina rada digitalnog pina.

<i>Linija</i>	<i>Kôd</i>
87:	<code>void buttonInterrupt() {</code>
88:	<code>  if( button == false ) {</code>
89:	<code>    button = true;</code>
90:	<code>    mode++;</code>
91:	<code>    if( mode &gt; 4 ) {</code>
92:	<code>      mode = 0;</code>
93:	<code>    }</code>
94:	<code>  }</code>
95:	<code>}</code>

Sl. 4.19. *buttonInterrupt* funkcija.

<b>Linija</b>	<b>Kôd</b>
106:	void loop() {
...	...
119:	if(button) {
120:	delay(20);
121:	if(digitalRead(D3)==LOW) {
122:	button=false;
123:	switch(mode) {
124:	case 0:
125:	rpmMode=true;
126:	break;
127:	case 1:
128:	rpmMode=false;
129:	colorWipe(red, 20);
130:	break;
131:	case 2:
132:	rpmMode=false;
133:	colorWipe(green, 20);
134:	break;
135:	case 3:
136:	rpmMode=false;
137:	colorWipe(blue, 20);
138:	break;
139:	case 4:
140:	rpmMode=false;
141:	colorWipe(white, 20);
142:	break;
143:	}
144:	}
145:	}
146:	}

Sl. 4.20. *Debounce* tipkala u *loop()* funkciji.

Rješenje za debounce tipkala kôdom je vidljivo na slikama 4.19. i 4.20. S obzirom da se unutar *interrupta* ne može koristiti funkcija *delay* za *debounce*, on je realiziran korištenjem zastavice *button* koja se na pojavu padajućeg brida odnosno pritiska tipkala unutar funkcije *buttonInterrupt* podiže te se odrađuje promjena načina osvjetljenja povećanjem varijable *mode*. Kada je zastavica podignuta uvjet u liniji 119. je istinit te se unutar naredbe *if* poziva *delay* funkcija u trajanju od 20ms. Tada slijedi provjera je li skakanje napona završilo provjerom je li na ulazu D3 logička 0. Ako se napon stabilizirao u 0 zastavica se spušta i omogućuje se ponovno povećanje varijable *mode* na idući ulazak u *interrupt* funkciju odnosno pritisak tipkala. U *switch* naredbi se ovisno o vrijednosti varijable *mode* postavlja vrijednost varijable *rpmMode* i poziva funkcija *colorWipe* za promjenu boje LED trake u određenu boju s odgodom od 20ms. Funkcija *colorWipe* je prikazana na slici 4.15. Važnost postavljanja vrijednosti *rpmMode* varijable je objašnjena u idućem potpoglavlju.

## 4.6. Način osvjetljenja u ovisnosti o frekvenciji okretaja motora

<b>Linija</b>	<b>Kôd</b>
50:	void calculateRPM() {
51:	if (abs(previousCount-count)==1) {
52:	count=previousCount;
53:	}
54:	rpm=((float) count/0.250f) *31.236f;
55:	previousCount=count;
56:	count=0;
57:	if (rpm>6000) {
58:	R=255;
59:	G=0;
60:	B=0;
61:	}
62:	else if (rpm>=1800) {
63:	R=map(rpm, 1800, 6000, 0, 180);
64:	G=map(rpm, 1800, 6000, 180, 0);
65:	B=0;
66:	}
67:	else if (rpm>=1200) {
68:	R=0;
69:	G=map(rpm, 1200, 1800, 0, 180);
70:	B=map(rpm, 1200, 1800, 180, 0);
71:	}
72:	else {
73:	R=0;
74:	G=0;
75:	B=255;
76:	}
77:	
78:	RPMcolor=strip.Color(R, G, B);
79:	if (rpmMode==true) {
80:	for(int i=0; i<NUMPIXELS; i++) {
81:	strip.setPixelColor(i, RPMcolor);
82:	strip.show();
83:	}
84:	}
85:	}

Sl. 4.21. Funkcija *calculateRPM*.

Za računanje trenutnog broja okretaja u minuti je potreban neki vremenski period unutar kojega se izvodi brojanje vrhova signala. Arduinom je pozivanje neke funkcije u jako preciznom vremenskom intervalu najlakše realizirati korištenjem brojača. Radi toga je korištena biblioteka *TimerOne* kojom je dobiven pristup 16-bitnom brojaču, jednom od 3 ugrađena brojača mikroupravljača ATmega328 [20]. Pri inicijalizaciji brojača se upisuje vremenski interval u

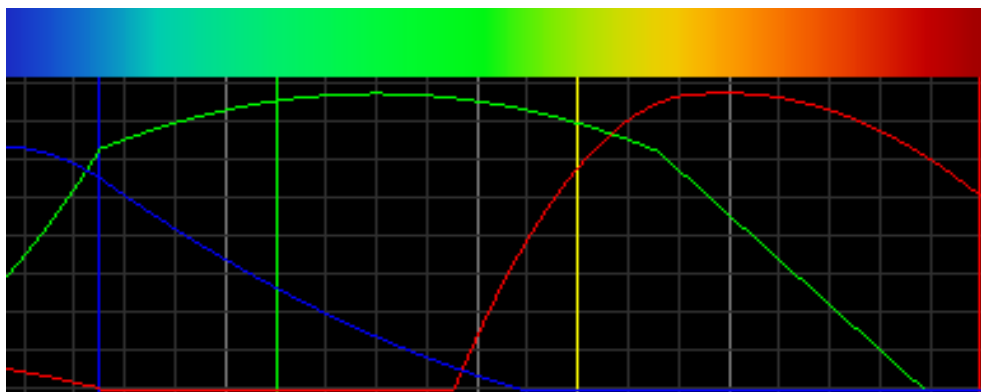
<b>Linija</b>	<b>Kôd</b>
4:	#include <TimerOne.h>
...	...
27:	void setup() {
28:	Timer1.initialize(250000);
29:	Timer1.attachInterrupt(calculateRPM);

Sl. 4.22. Postavljanje brojača.



mikrosekundama te postavlja *interrupt* rutina koja se poziva svaki puta kada prođe upisani vremenski interval. Kôd za postavljanje brojača i *interrupt* rutine je vidljiv na slici 4.22.

Time se postigne pozivanje *calculateRPM* funkcije svakih  $250000\mu\text{s}$  odnosno 250ms. Funkcija *calculateRPM* je prikazana na slici 4.21. U njoj se koristi varijabla *count* u koju se pohranjuje broj detektiranih vrhova iz dijela kôda na slici 4.12. Prvo se u provjerava razlikuje li se vrijednost *count* za 1 od prethodne vrijednosti *count* čime je eliminirano titranje boja pri stalnom broju okretaja. Do titranja je dolazilo jer se frekvencija ulaznog signala ne podudara uvijek s vremenskim intervalom u kojemu se broje vrhovi pa se u jednom intervalu izbroji jedan vrh više, a u drugom jedan vrh manje naizmjenice iako je stalna frekvencija ulaznog signala. Nakon toga se računa broj okretaja u minuti i pohranjuje u varijablu *rpm*. Broj okretaja u minuti se računa tako da se vrijednost varijable *count* podijeli s 0.25 jer je period u kojem se broje vrhovi 250ms. Rezultat dijeljenja je frekvencija signala u Hz, stoga se prema izrazu (3-7) množi s  $K_p$  kako bi se dobio broj okretaja motora u minuti. Nakon toga se sprema vrijednost varijable *count* u *previousCount*, a *count* se postavlja u 0 kako bi se moglo ponovo izvršiti brojanje. Zatim slijedi niz uvjeta kojima je dobiveno aditivno miješanje boja pomoću funkcije *map*. Funkcija *map* linearno preslikava vrijednost varijable *rpm* u zadanom rasponu (npr. za crvenu boju od 1800 do 6000 okretaja) u drugu varijablu u zadanom rasponu (npr. u varijablu *R* vrijednosti od 0 do 180 za crvenu boju). Krivulje RGB vrijednosti koje se tim nizom uvjeta aproksimiralo vidljive su na slici 4.23.



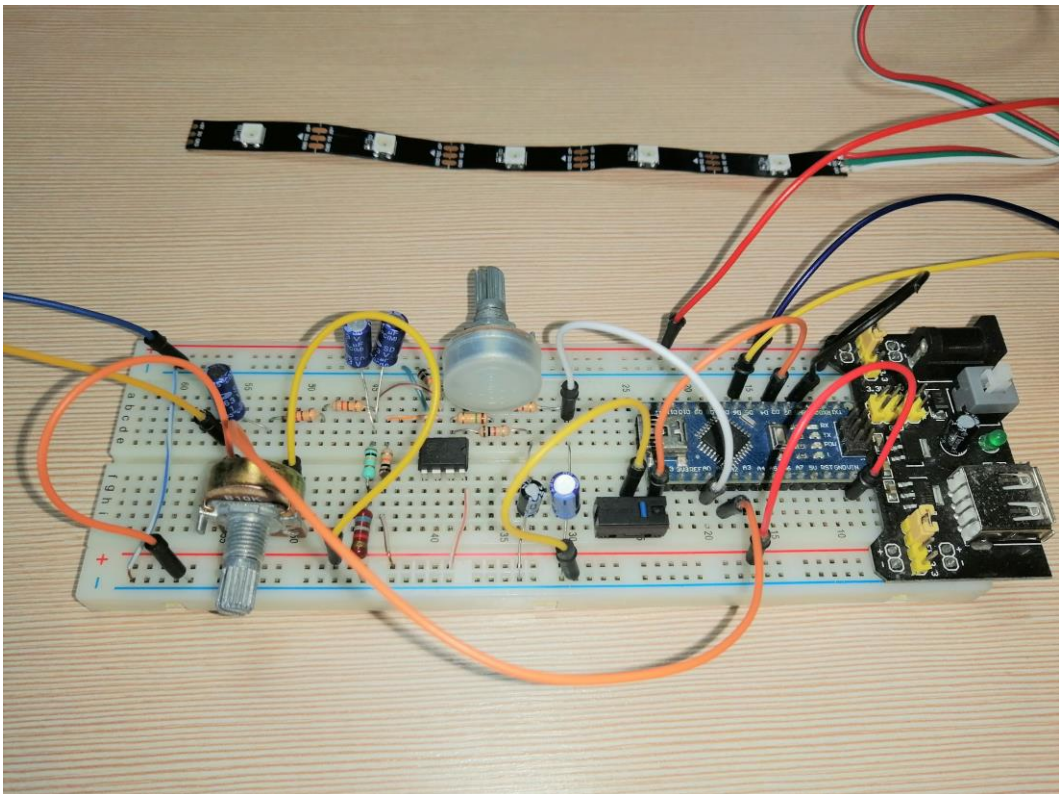
Sl. 4.23. Krivulje R, G i B vrijednosti na spektru vidljive svjetlosti.

Tim načinom dobivene R, G i B vrijednosti spremaju se u varijablu *RPMcolor*. *If* naredbom u liniji 79 se provjerava je li odabran način za osvjetljenje koje ovisi o broju okretaja varijablom *rpmMode* koja se postavlja u dijelu kôda sa slike 4.20.

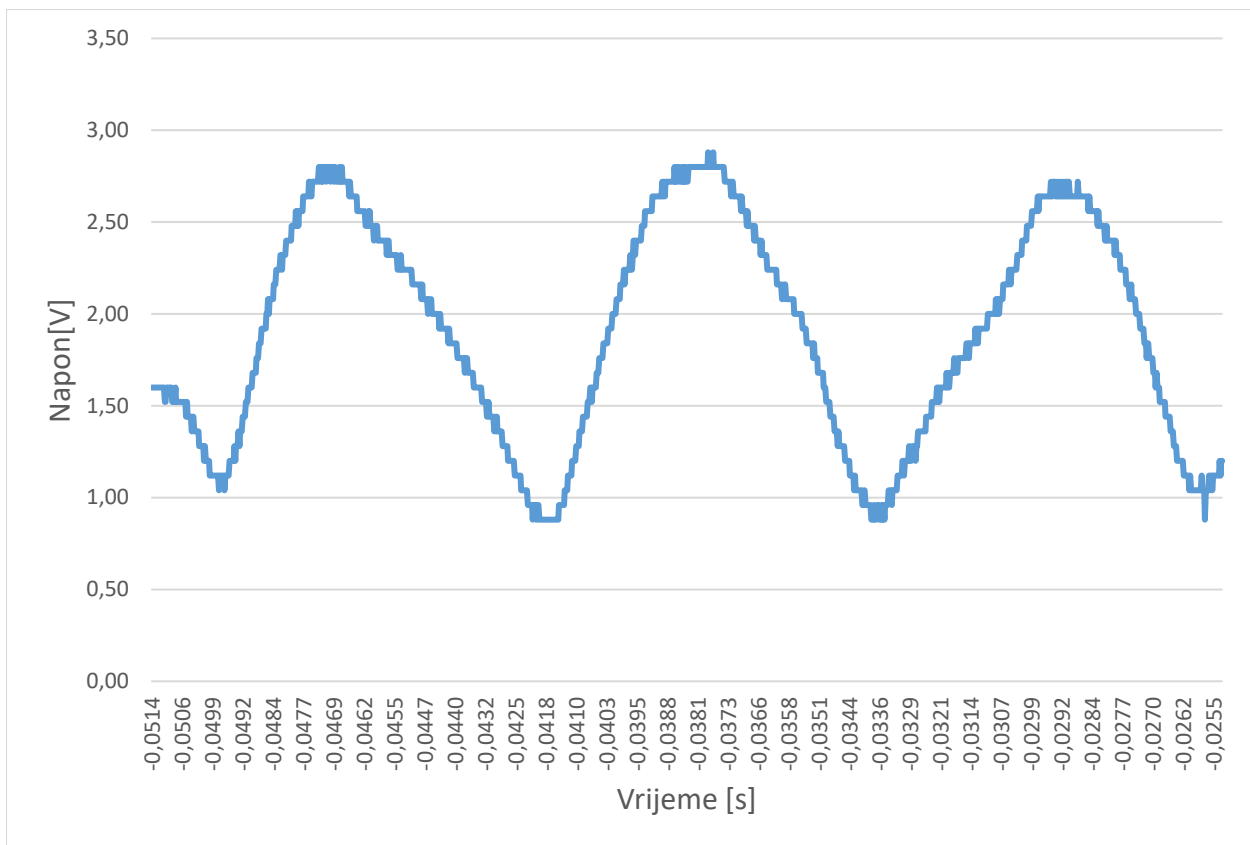
Ako je vrijednost varijable *rpmMode true*, izvodi se *for* petlja slična onoj u funkciji *colorWipe* samo bez odgode između postavljanja boje pojedinih dioda na boju spremljenu u varijablu *RPMcolor*.

## 4.7. Izrada sustava na eksperimentalnoj pločici i testiranje

Sustav je složen na eksperimentalnoj pločici radi lakog ubadanja *DIP* (engl. *Dual In-line Package*) komponenti: Arduino Nano i pojačalo LM358. Sustav je vidljiv na slici 4.24. Za istosmjerno napajanje korištena je pločica za napajanje koja ulazni istosmjerni napon u rasponu od 6V do 12V spušta na 5V. Za napajanje pločice korišten je adapter za pretvaranje izmjeničnog napona gradske mreže u istosmjerni napon 12V. Napon gradske mreže je korišten samo radi testiranja kako bi se osiguralo da se kroz napajanje ne uvodi šum, a radi prenosivosti je sustav moguće napajati unutar automobila preko utičnice upaljača.

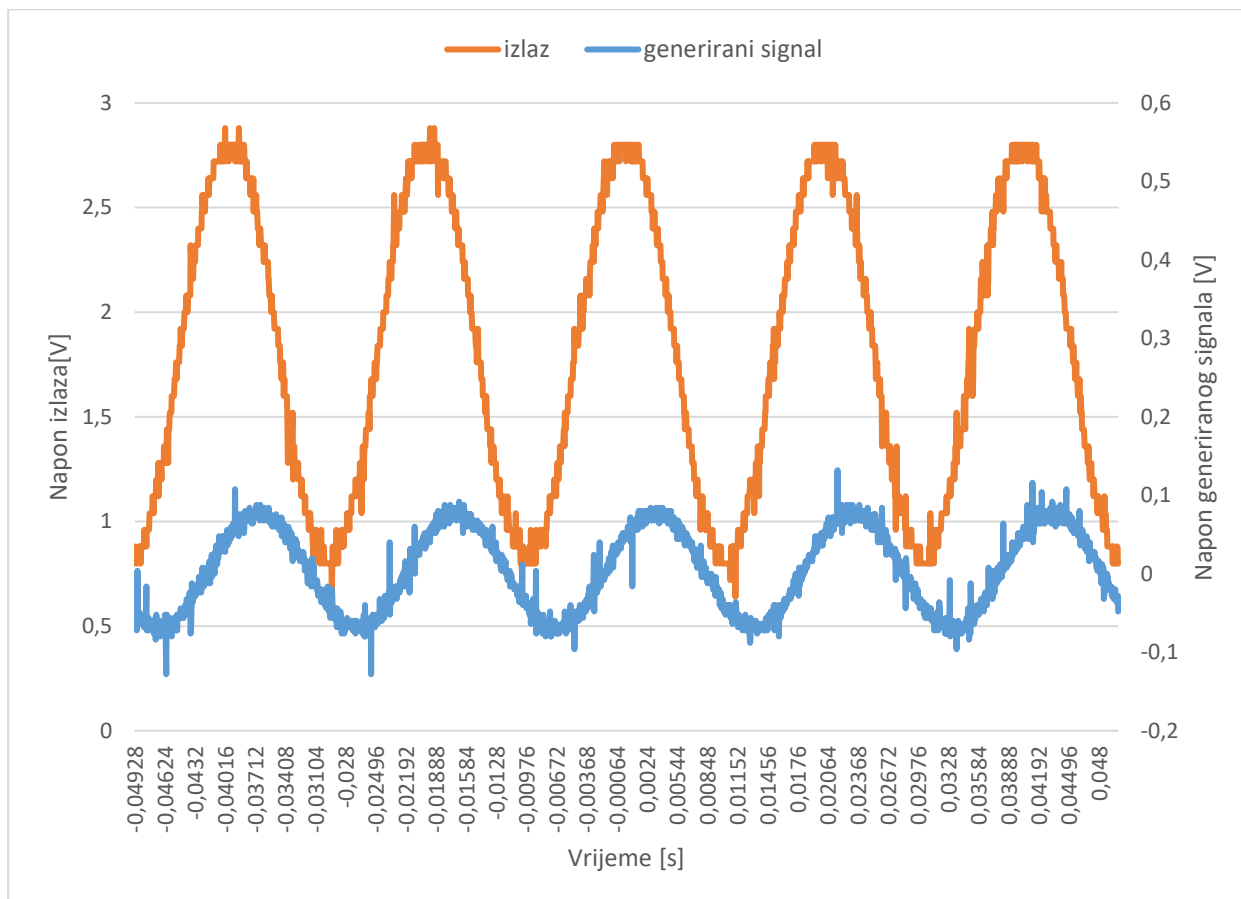


Sl. 4.24. Sustav na eksperimentalnoj pločici.



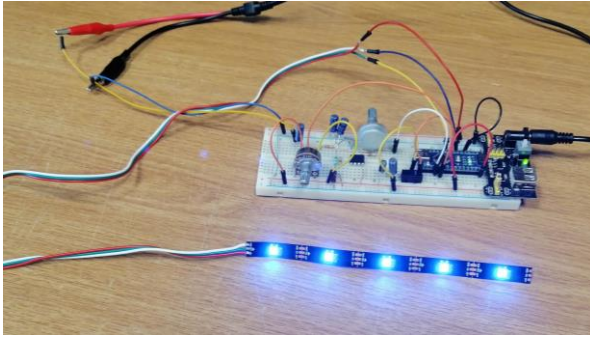
Sl. 4.25. Izlaz analognog dijela sustava pri 3900 okretaja u minuti.

Analogni dio sustava za pojačanje i filtraciju je testiran na istom automobilu na kojem je snimljen signal prikazan na slikama 3.2. i 3.3. Izlaz analognog dijela sustava osciloskopom je snimljen pri 3900 okretaja u minuti i prikazan na slici 4.25. Na slici je vidljivo da je signal prihvatljivo filtriran i pojačan na raspon od 1V do 3V što je dobro područje za mjerenje analognim ulazom Arduina.

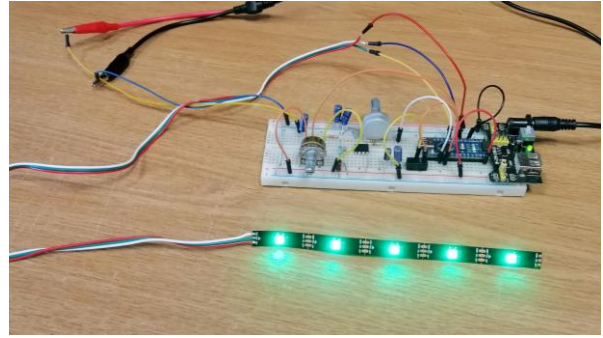


Sl. 4.26. Izlaz analognog dijela sustava i generirani signal frekvencije 60Hz.

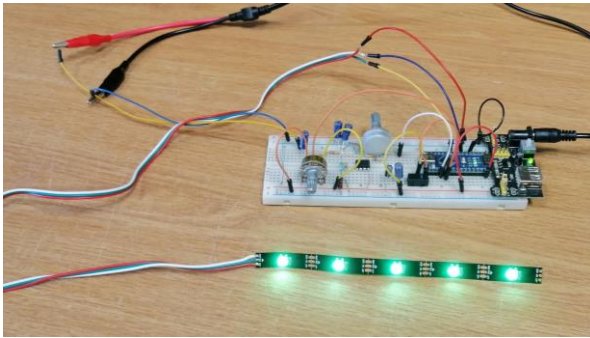
U laboratoriju je sustav testiran pomoću funkcijskog generatora radi mogućnosti proizvoljne promjene frekvencije i oblika signala. Generiran je sinusni signal amplitude 150mV na ulazu u sustav, a frekvencija je mijenjana od 30Hz do 180Hz u koracima od 30Hz. Generirani signal i izlaz analognog dijela sklopa snimljeni su osciloskopom, a na slici 4.26. je vidljiva snimka pri frekvenciji od 60Hz.



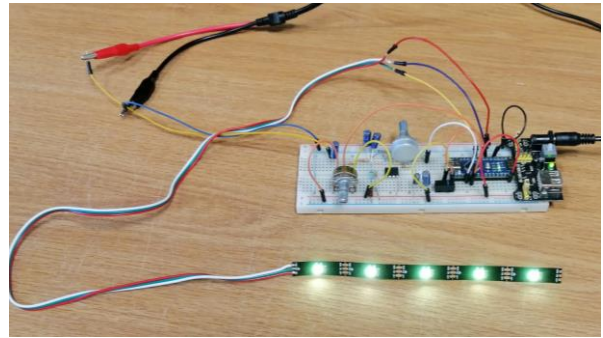
(a)



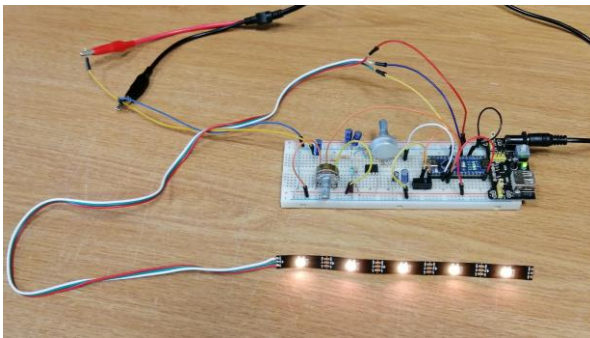
(b)



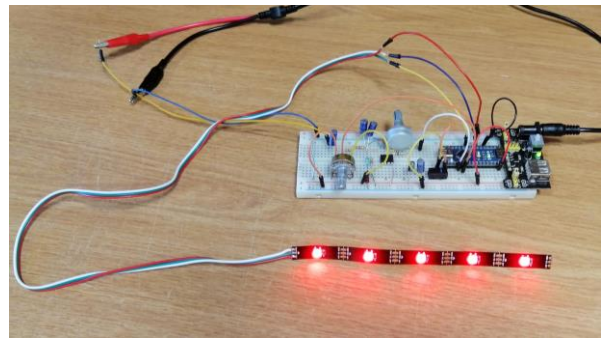
(c)



(d)



(e)



(f)

Sl. 4.27. (a-f) Boje osvjetljenja uz ulazni signal postupno rastuće frekvencije.

Na slici 4.27. vidljive su boje RGB trake pri odabranim frekvencijama ulaznog signala. Za pojedinu sliku (a-f) prema izrazu (3-7) odabrane frekvencije ulaznog signala odgovaraju sljedećim frekvencijama okretaja motora:

- (a) 937 okretaja u minuti
- (b) 1874 okretaja u minuti
- (c) 2811 okretaja u minuti
- (d) 3748 okretaja u minuti
- (e) 4685 okretaja u minuti
- (f) 5622 okretaja u minuti.



Sustav je testiran i na automobilu modela Opel Astra te je pokazao ispravan rad u većem dijelu mjernog područja. Testiranje na automobilu prikazano je slikom 4.28. Na frekvencijama okretaja do 4000 okretaja po minuti je točno određen broj okretaja, no na većim je okretajima dolazilo do pogreške u ponekim vremenskim intervalima. Pogreška je bila vidljiva titranjem boje kada u ponekom intervalu sustav izbroji manje vrhova od stvarnog broja. Do pogreške vjerojatno dolazi zbog smanjenja amplitude signala pri većem broju okretaja. Prijedlog za poboljšanje sustava je korištenje aktivnih filtara uz veće pojačanje.



Sl. 4.28. Testiranje na automobilu.

## 5. ZAKLJUČAK

U ovome radu opisan je proces razvoja vlastitog sustava, od mjerenja i dizajna analognog sklopa za pojačanje i filtraciju signala, do pisanja Arduino kôda za računanje broja okretaja i upravljanje RGB trakom. Proučene su i objašnjene metode za brojanje okretaja automobila te napravljena usporedba sustava razvijenog u ovome radu s postojećim proizvodima za brojanje okretaja. Opisan je izvor korištenog signala te prikazana mjerenja potrebna za podešavanje analognog sklopa. Opisane su komponente korištene za razvoj analognog sklopa čija je uloga filtracija i pojačanje signala na razinu na kojoj Arduino razvojna platforma točno mjeri skokove napona. Objašnjene su pojedinosti Arduino platforme i kako su utjecale na strukturu kôda. Objašnjen je način rada pojedinih dijelova kôda i njihova međusobna interakcija. Sustav je testiran na automobilu i ispunio sve postavljene zahtjeve, no određenim se promjenama može postići pouzdaniji i jednostavniji rad.

Moguće poboljšanje sustava je korištenje aktivnih filtera za smanjenje šuma. Iako je razvijeni sustav moguće prilagoditi za druge automobile čiji signal može biti drugačije amplitude i frekvencije, bilo bi veliko unaprjeđenje razviti algoritam za automatsku kalibraciju sustava i napredniju detekciju vrhova signala. Također za lakše korištenje i ljepši izgled uređaja bi bilo potrebno dizajnirati tiskanu pločicu i kućište.

## LITERATURA

- [1] V., Kondić, M., Horvat, F., Maroević, „PRIMJENA DIJAGNOSTIKE KAO OSNOVE ODRŽAVANJA PO STANJU NA PRIMJERU MOTORA OSOBNOG AUTOMOBILA“, str. 7, 2013.
- [2] A. X. A., Sim, B., Sitohang, „OBD-II standard car engine diagnostic software development“, u *2014 International Conference on Data and Software Engineering (ICODSE)*, str. 1–5, Bandung, Indonesia, 2014.
- [3] „Freematics OBD-II UART Adapter V2.1 (for Arduino)“ [online]. Dostupno na: [https://freematics.com/store/index.php?route=product/product&path=20&product\\_id=83](https://freematics.com/store/index.php?route=product/product&path=20&product_id=83). [Pristupljeno: 31.8.2021.].
- [4] S. Huang, „ArduinoOBD/libraries/OBD at master · stanleyhuangyc/ArduinoOBD“ [online]. Dostupno na: <https://github.com/stanleyhuangyc/ArduinoOBD>. [Pristupljeno: 31.8.2021.].
- [5] I. Hare, „How to Make a Gear Display Screen for Your Car with an Arduino Uno“ [online]. Dostupno na: <https://www.youtube.com/watch?v=PhU6Jy2ZvGY>, [Pristupljeno: 31.8.2021.].
- [6] S. Huang, „DIY a digital tachometer with Arduino & RGB LED matrix (video)“ [online]. Dostupno na: <http://arduino4dev.com/arduino-obd-data-indicator-rgb-matrix/>, [Pristupljeno: 31.8.2021.].
- [7] C. Rowe, „Reading ECU data with an obd-ii Arduino module“ [online]. Dostupno na: <https://www.youtube.com/watch?v=JdtRn1oTF4A>, [Pristupljeno: 31.8.2021.].
- [8] „OBD Software – Elm Electronics“ [online]. Dostupno na: <https://www.elmelectronics.com/help/obd/software/>, [Pristupljeno: 31.8.2021.].
- [9] „How to easy Read/Clear car Fault Codes [ELM327] OBD II“ [online]. Dostupno na: <https://www.youtube.com/watch?v=JJa0nsrQXI0>, [Pristupljeno: 31.8.2021.].
- [10] „Download Bosch BEA 030 Original Instructions Manual | ManualsLib“ [online]. Dostupno na: <https://www.manualslib.com/download/1691603/Bosch-Bea-030.html>. [Pristupljeno: 29.8.2021.].
- [11] „G-TECH/Pro SS“ [online]. Dostupno na: <http://www.gtechpro.com/G-TECH-Pro%20SS>. [Pristupljeno: 30.8.2021.].
- [12] „Support - User Guides (Manuals)“ [online]. Dostupno na: <http://www.gtechpro.com/Support User Guides Manuals>. [Pristupljeno: 30.8.2021.].
- [13] Auto-mane, „Što je alternator?“ [online]. Dostupno na: <https://auto-mane.com/abeceda-automobila/sto-je-alternator>. [Pristupljeno: 5.9.2021.].
- [14] S., Čerlek, „Električna i elektronička oprema u vozilima – senzori i izvršni sklopovi“, Veleučilište Velika Gorica, Velika Gorica, 2010.
- [15] A. D., Belegić, „OSNOVNI TIPOVI PASIVNIH FILTERA“, str. 129.
- [16] B., Carter, T. R., Brown, „HANDBOOK OF OPERATIONAL AMPLIFIER APPLICATIONS“, 2001.
- [17] „LM358 Dual Op-Amp“ [online]. Dostupno na: <https://protosupplies.com/product/dual-op-amp-lm358/>, [Pristupljeno: 5.9.2021.].
- [18] A., Kurniawan, "Arduino Nano A Hands-On Guide for Beginner.", PE Press, 2019.
- [19] Adafruit Industries, „Adafruit NeoPixel Library“ [online]. Dostupno na: [https://github.com/adafruit/Adafruit\\_NeoPixel](https://github.com/adafruit/Adafruit_NeoPixel), [Pristupljeno: 5.9.2021.].
- [20] „Arduino Playground - Timer1“ [online]. Dostupno na: <https://playground.arduino.cc/Code/Timer1/>. [Pristupljeno: 8.9.2021.].



## SAŽETAK

Cilj ovog završnog rada bio je razvoj mikroupravljačkog sustava za upravljanje unutarnjom rasvjetom automobila koristeći Arduino razvojnu pločicu. Korisniku je kreirano pet načina osvjetljenja, jedan od kojih je osvjetljenje koje ovisi o broju okretaja motora. Razvijen je analogni sklop za filtraciju i pojačanje signala na razinu koju mikroupravljač točnije mjeri. Pri razvoju analognog sklopa korištena je simulacija u programu Multisim, te su rezultati simulacije bili vjerni prikaz stvarnih mjerenja. Prilikom pisanja programa uzimala su se u obzir ograničenja sklopovlja, kao što je korištenje *interrupta* radi veće frekvencije uzorkovanja te pouzdanijeg rada. Nakon izrade sklopa i razvoja programa sustav je testiran te su rezultati bili zadovoljavajući.

**Ključne riječi:** Arduino, filtriranje, interrupt, okretaji, osvjetljenje

## **ABSTRACT**

Title: Ambient LED car lighting

The aim of this paper was development of a microcontroller-based system for controlling car interior lighting using an Arduino development board. There were five modes of lighting created for the user, one of which was engine RPM dependent. An analog signal amplification and filtration circuit was developed in order to modulate the signal to better suit a microcontroller input. Circuit simulation in Multisim was used to develop the analog circuit, and simulation results were a good representation of measured results with the finished circuit. Hardware constraints of the microcontroller were taken into account during software development and interrupts were used to achieve a higher polling rate and a more reliable system. The system was tested and the results were satisfactory.

**Key words:** Arduino, filtering, interrupt, revolutions, lighting

## ŽIVOTOPIS

Dominik Đuranić rođen je 12. svibnja 1998. godine u Osijeku. Završio je Osnovnu školu Ladimirevci, te opću gimnaziju u Srednjoj školi Valpovo. Tijekom srednje škole sudjelovao je na brojnim županijskim natjecanjima iz fizike, engleskog jezika te na državom natjecanju iz logike. Trenutno je student 3. godine preddiplomskog sveučilišnog studija Računarstva na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek. Na fakultetu je sudjelovao u izradi plakata za Festival znanosti 2019. godine te radio kao demonstrator iz predmeta Digitalna elektronika.

---

Potpis autora

## **PRILOZI**

Na CDu nalaze se sljedeći prilozi:

- .ino datoteka koja sadrži Arduino kôd