

Ambijentalna LED rasvjeta automobila s OBD protokolom

Tandara, Marko

Master's thesis / Diplomski rad

2022

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:200:706370>

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

Download date / Datum preuzimanja: **2024-08-16**

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 S
OBD PROTOKOLOM**

Diplomski rad

Marko Tandara

Osijek, 2022.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit**

Osijek, 18.09.2022.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za diplomski ispit

Ime i prezime Pristupnika:	Marko Tandara
Studij, smjer:	Diplomski sveučilišni studij Računarstvo
Mat. br. Pristupnika, godina upisa:	D-1171R, 13.10.2020.
OIB studenta:	25018402247
Mentor:	Izv. prof. dr. sc. Tomislav Matić
Sumentor:	,
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Izv. prof. dr. sc. Ivan Aleksi
Član Povjerenstva 1:	Izv. prof. dr. sc. Tomislav Matić
Član Povjerenstva 2:	Izv. prof. dr. sc. Damir Blažević
Naslov diplomskog rada:	Ambijentalna LED rasvjeta automobila s OBD protokolom
Znanstvena grana diplomskog rada:	Arhitektura računalnih sustava (zn. polje računarstvo)
Zadatak diplomskog rada:	Potrebno je razviti mikroupravljački sustav koji će upravljati unutarnjom rasvjetom automobila na temelju korisničkih postavki i broja okretaja motora. Podatak o broju okretaja potrebno je dobiti putem OBD bluetooth uređaja.
Prijedlog ocjene pismenog dijela ispita (diplomskog 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 od strane mentora:	18.09.2022.
Potvrda mentora o predaji konačne verzije rada:	<i>Mentor elektronički potpisao predaju konačne verzije.</i>
	Datum:

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

Osijek, 28.09.2022.

Ime i prezime studenta:

Marko Tandara

Studij:

Diplomski sveučilišni studij Računarstvo

Mat. br. studenta, godina upisa:

D-1171R, 13.10.2020.

Turnitin podudaranje [%]:

15

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

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:

SADRŽAJ

1. UVOD.....	1
1.1. Zadatak završnog rada.....	1
2. TRENUTNO STANJE TEHNIKE	2
2.1. YZG ambijentalna rasvjeta.....	2
2.2. OPT7 Aura	2
2.3. SANLI LED	3
3. UREĐAJ ZA UPRAVLJANJE LED RASVJETOM	4
3.1. Sklopovlje.....	4
3.1.1. Arduino.....	4
3.1.2. Arduino Nano	6
3.1.3. HC05 bluetooth modul	7
3.1.4. OLED zaslon	8
3.1.5. WS2812B LED.....	9
3.1.6. EasyEDA.....	10
3.1.7. Tiskana pločica	11
3.1.8. Elektroničke komponente	15
3.2. Dijagnostika automobila.....	16
3.2.1. OBD.....	16
3.2.2. ELM327.....	17
3.3. Programska podrška.....	19
3.3.1. Kreiranje objekata i uspostava komunikacije	20
3.3.2. Korištenje EEPROM memorije	21
3.3.3. Provjera statusa tipkala	22
3.3.4. Određivanje određenog načina rada	23
3.3.5. Izbornik za postavke uređaja	24
3.4. Testiranje	26
4. ZAKLJUČAK.....	32
LITERATURA.....	33
SAŽETAK.....	34
ABSTRACT	35
ŽIVOTOPIS.....	36
PRILOZI.....	37

1. UVOD

U ovo moderno doba, sve više i više dolazi do zastupljenosti ambijentalne rasvjete u prostorijama, raznim tehnološkim uređajima pa tako i autima. Ambijentalna rasvjeta utječe na čovjekovo raspoloženje, pa tako uz pravu boju i temperaturu u prostoriji ili kao u ovom slučaju autu, može se postići ugodna i opuštajuća atmosfera. Tema ovoga rada je napraviti sklop koji će s obzirom na trenutne parametre auta (broj okretaja motora, brzina, temperatura rashladne tekućine...) upravljati ambijentalnom rasvjetom u autu. Uređaj pomoću Arduino Nano mikroupravljača na WS2812B LED (engl. *Light Emitting Diode*) traci, generira animacije s obzirom na podatke koje dobije od HC-05 modula. HC-05 modul prosljeđene podatke prima od ELM327 adaptera putem *bluetooth* veze, a sam ELM327 poslane podatke čita putem OBD-II (engl. *On-Board Diagnostics*) sučelja kao trenutne vrijednosti pojedinih parametara automobila. Uređaj sadrži neke dodatne funkcionalnosti kao što su: promjena boje osvjetljenja, postavljanje minimalnog i maksimalnog intenziteta osvjetljenja i određivanje broja ledica koje se žele osvijetliti na LED traci. Osim što trenutne podatke prikazuje u obliku svjetlosnih animacija, sadržava OLED (engl. *Organic Light-Emitting Diodes*) zaslon na kojem su ti podatci prikazani u numeričkom obliku. Korisnik ima mogućnost odabira između četiri načina rada, pritiskom na tipku *mode* na PCB-u (engl. Printed Circuit Board).

U drugom poglavlju rada navedena su postojeća rješenja ambijentalne rasvjete. U trećem poglavlju opisane su komponente i sklopovlje koji su korišteni za izradu uređaja. U četvrtom objašnjena je programska podrška, a u petom poglavlju prikazan je rad uređaja sa svim njegovim funkcijama.

1.1. Zadatak završnog rada

U radu je potrebno ostvariti upravljanje ambijentalnom rasvjetom u automobilu koristeći neki mikroupravljač, koji će preko *bluetootha* komunicirati s ELM327 adapterom koji putem OBD-a čita trenutne vrijednosti parametara automobila. Pomoću tih vrijednosti mikroupravljač generira određene animacije na LED traci.

2. TRENUTNO STANJE TEHNIKE

Ambijentalna rasvjeta unutar automobila se postiže na razne načine. Najčešće dolazi već implementirana u novijim autima od strane njihovih proizvođača. Dok postoje i opcije za kupovinom raznih paketa od prodavača trećih strana. U nastavku će biti navedeni neki najpopularniji i najdostupniji uređaji za ambijentalnu rasvjetu.

2.1. YZG ambijentalna rasvjeta

Ovaj paket za rasvjetu dolazi u obliku tanke trake, koja se sastoji od optičkih vlakana. Zbog svoje male širine, traka se može instalirati oko kontrolne ploče. Paket sadržava mikroupravljač kojim se može upravljati putem mobitela koristeći YZG aplikaciju. Traka može prikazivati 16.8 milijuna boja, ali mana je što nema sposobnost upravljanja pojedinom ledicom kao traka korištena u ovome radu. Okvirna cijena ovog paketa je 800kn, ali može biti i veća ako se uzme veća dužina trake [1].



Sl. 2.1. YZG ambijentalna rasvjeta.

2.2. OPT7 Aura

OPT7 aura paketi za LED rasvjetu su jedni od najprodavanijih i priuštivih paketa RGB LED rasvjete. Paketi dolaze u više varijanta, ovisno šta korisniku treba. U svim paketima koriste se LED trake koje sadržavaju ledice u 5050 formatu. Neke funkcionalnosti koje se mogu naći u paketima su soundsync (kontroliranje LED trake ovisno o zvuku), upravljanje trakom putem daljinskog upravljača ili putem aplikacije na mobilnom uređaju. Cijene paketa se kreću od 300 do 600kn [2].



Sl. 2.2. OPT7 Aura.

2.3. SANLI LED

Sanli paket ambijentalne rasvjete dolazi u obliku jedne RGBW 16W lampe. Na lampu se priključuju optičke niti koje provode svjetlo do vrhova niti. Niti se provode kroz krov auta, gdje se vrhovi niti usmjeravaju dolje, okomito na krov. Provlačenjem niti na taj način dobiva se efekt zvjezdanog neba. Lampa dolazi sa upravljačem i ugrađenim *bluetooth-om*, što omogućava spajanje s mobilnim uređajem. Aplikacija za upravljanje osvjetljenjem pruža velik izbor opcija osvjetljenja. Podržava 16 milijuna boja, te ima opciju gdje osvjetljenje reagira na ritam glazbe. Cijena urađaja iznosi 400kn [3].



Sl. 2.3. SANLI LED.

3. UREĐAJ ZA UPRAVLJANJE LED RASVJETOM

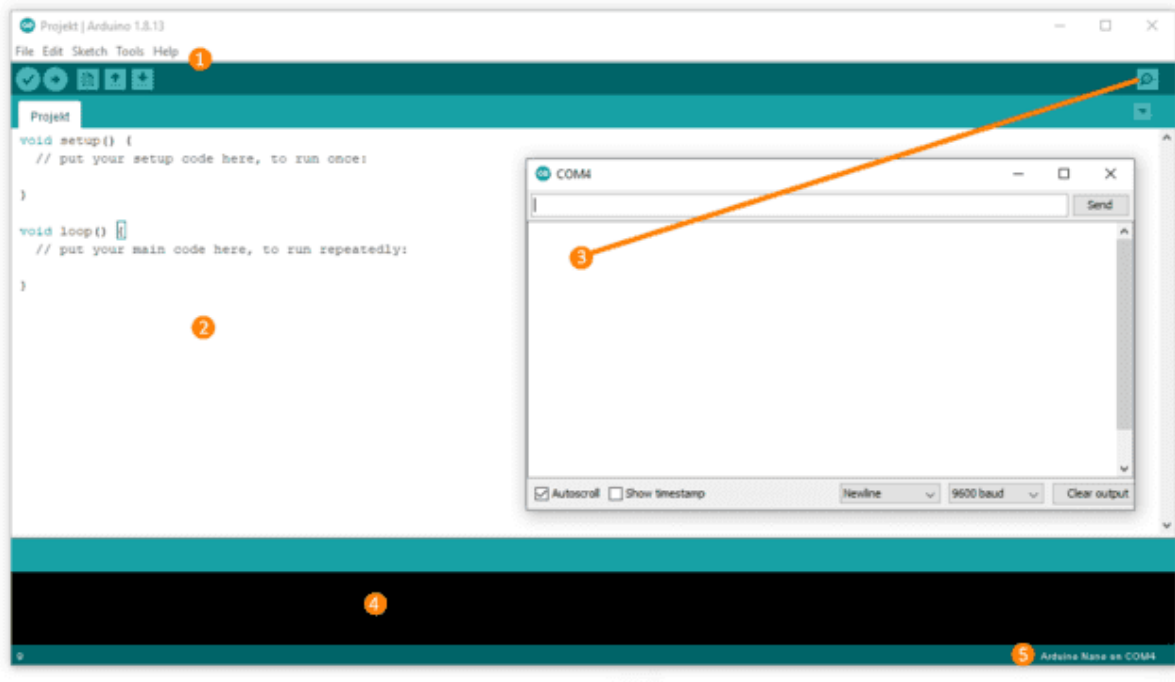
3.1. Sklopovlje

Glavne komponente korištene za izradu uređaja su Arduino Nano mikroupravljač, HC-05 modul za *bluetooth* komunikaciju, OLED zaslon za prikaz vrijednosti parametara automobila i WS2812B LED traka za prikaz animacija ambijentalne led rasvjete. Na uređaju se nalaze još neke dodatne komponente čije su funkcionalnosti detaljno opisane na kraju poglavlja (potencijometar, tipkalo i pretvarači napona).

3.1.1. Arduino

Arduino platforma se koristi za izgradnju elektroničkih projekata koji međusobno komuniciraju. Otvorenog je tipa (engl. *open-source*), što znači da je dozvoljeno dijeljenje podataka o programskoj podršci i sklopovljima koje je dozvoljeno izmjenjivati i nadograđivati. Isto tako dozvoljeno je stvaranje novih razvojnih pločica i razvojnih okruženja baziranih na Arduino platformi [4].

Za pisanje, kompajliranje i učitavanje programskog kôda na Arduino mikroupravljače koristi se Arduino IDE (engl. *Integrated Development Environment*) razvojno okruženje (slika 3.1.). Programski kôd u Arduino IDE piše se programskim jezikom koji je nastao iz C/C++ programskog jezika. Mikroupravljač za kojeg se piše programski kôd na sebi mora imati već prethodno isprogramiran Bootloader, odnosno dio kôda koji će primati podatke putem UART (engl. *Universal Asynchronous Receiver-Transmitter*) komunikacije i tako programirati samog sebe. On se također prvi pokreće prilikom uključivanja razvojne pločice, te dohvaća kôd iz memorije i pokreće njegovo izvršavanje [5].



Sl. 3.1. Izgled Arduino IDE sučelja.

Neki od glavnih dijelova Arduino IDE su:

1. Izbornička traka – na njoj se mogu otvarati pojedini izbornici u kojima se nalaze brojne funkcionalnosti: primjeri kôdova, upravljanje editorom, upravljanje bibliotekama, odabiranje porta, itd.
2. *Editor* – prostor namijenjen za pisanje programskog kôda
3. Serijski monitor – prozor koji se otvara kada se pritisne gore desno označeni gumb. Koristi za prikaz podataka koje mikroupravljač šalje putem UART komunikacije (za ispravnu komunikaciju potrebno je imati istu brzinu prijenosa (engl. *baud rate*) s obje strane)
4. Informacijska traka – na njoj se očitavaju trenutačna stanja i greške prilikom kompajliranja ili učitavanja programskog kôda na mikroupravljač (korisno kod lociranja grešaka u napisanom kôdu)
5. traka za prikazivanje trenutno odabranih razvojnih pločica i portova za učitavanje kôda.

Na slici 3.1. vidljivo je da kada se pokreće projekt u *Editoru* se pojavljuju 2 funkcijska bloka (*void setup* i *void loop*). Kod pokretanja mikrokontrolera prvo se izvodi funkcija *void setup* i ona se izvršava samo jedanput, zatim se pokreće funkcija *void loop*, koja sadrži glavni dio programa koji obavlja neku funkciju. *Void loop* funkcija se neprestano ponavlja, što znači da ako je izvršen sav programski kôd u njoj, funkcija će se ponovno pokrenuti. Svaki se program sastoji od te dvije funkcije.

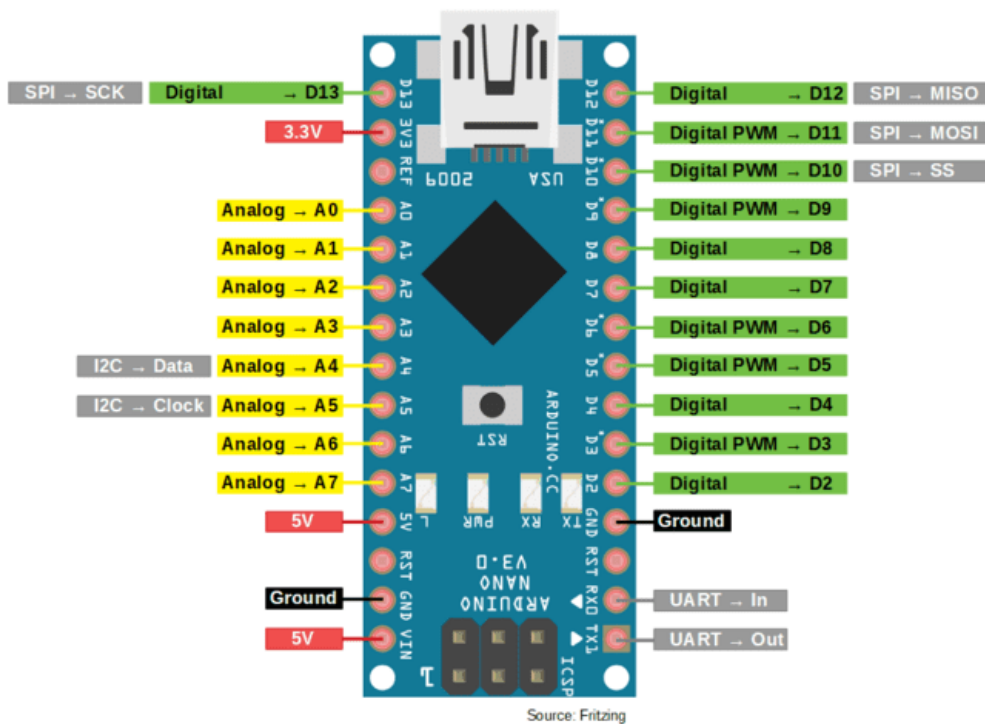
3.1.2. Arduino Nano

U ovom radu korištene su Arduino Nano v3.0 razvojne pločice (slika 3.2.). One su bazirane na ATmega328P mikroupravljaču od Atmela. Arduino Nano jedna je od najučinkovitiji i najjeftinijih razvojnih pločica Arduino okruženja. Za napajanje i učitavanje kôda na ATmega328P se koristi Mini-B USB priključak. Pločica sadrži 14 digitalnih ulazno/izlaznih nožica. Vrijednosti koje se šalju putem tih nožica mogu biti *HIGH* ili *LOW*, ovisno šalje li se na nožicu 1 ili 0. Struja za ulazno/izlazne nožice iznosi 40 mA. U tablici 3.1. su prikazana neka obilježja ATmega328P mikroupravljača.

Od 14 digitalnih nožica 6 se može koristiti kao PWM (*engl. Pulse Width Modulation*) izlaz. PWM ili pulsno-širinska modulacija je metoda dobivanja analognih vrijednosti pomoću digitalnih impulsa konstantne amplitude. Arduino Nano također ima 8 analognih nožica gdje se koristi 10 bitni ADC (*engl. Analog to Digital Converter*). AD pretvornik pretvara ulazni napon od 0 do 5 V u vrijednosti između 0 i 1023. Arduino Nano ima jednu od 3.3 V i dvije od 5 V naponske nožice od kojih je jedna VIN nožica. S VIN nožicom ulazni napon može biti od 7 V do 12 V koja se koristi za vanjski izvor napajanja. 5V nožica pruža regulirano napajanje mikroupravljača i drugih komponenti. 3.3 volta na nožici je generirano od ugrađenog naponskog regulatora na pločici. Sve tri nožice imaju protok struje od 50 mA. Također ima GND nožicu koja služi za uzemljenje. Sadrži tipku reset za ponovno pokretanje mikroupravljača. Rx i Tx nožice služe za slanje i primanje TTL (*engl. Transistor-Transistor Logic*) serijskih podataka [6].

Tab. 3.1. Obilježja Atmega328P.

Mikroupravljač	Atmega328P
Flash Memorija	32 KB (2 KB za <i>Bootloader</i>)
SRAM	2 KB
EEPROM	1024 B
Frekvencija	16 MHz



Sl. 3.2. Arduino Nano v3.0.

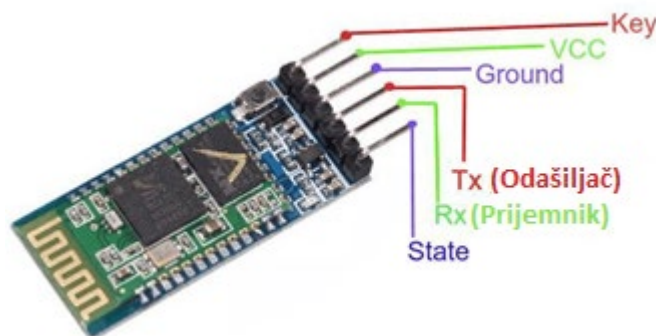
3.1.3. HC05 bluetooth modul

HC-05 je Bluetooth modul koji je dizajniran za bežičnu komunikaciju. Ovaj modul se može koristiti kao glavna ili prijemna jedinica (slika 3.3.). Koristi se u mnogim uređajima kao što su bežične slušalice, kontroleri za konzule, bežične tipkovnice i miševi, itd. Koristi FHSS (engl. *Frequency-Hopping Spread Spectrum*) radio tehnologiju za slanje podataka putem zraka, koja dijeli frekvencijski pojas na uže kanale i skače s kanala na kanal prilikom emitiranja. Za komunikaciju s mikrokontrolerom koristi serijsku komunikaciju putem USART (engl. *Universal Synchronous/Asynchronous Receiver/Transmitter*) sučelja. Ima doomet do 100 m što ovisi o odašiljaču/prijamniku, atmosferi i geografskim uvjetima. Ima dva glavna načina rada *data mode* (način rada za komunikaciju između uređaja) i *command mode* (način rada za konfiguraciju postavki modula putem AT komandi) [7]. Sam modul sadrži 6 nožica, koje imaju sljedeće uloge:

- VCC – služi za povezivanje napajanja od 5V ili 3.3V (ovisno o modulu)
- GND – nožica za uzemljenje
- TXD – nožica za slanje podataka kontroleru (podatci koji se prime bežično, šalju se mikrokontroleru putem serijske komunikacije)
- RXD – nožica za primanje podataka od strane kontrolera (podatci primljeni putem ove nožice poslat će se bežično uparenom uređaju)

- State – pokazuje da li je modul povezan ili ne
- Key – ako ova nožica ima vrijednost logičke jedinice, onda pri paljenju uređaja ulazimo u *command mode* (pritiskom tipkala na modulu ostvarujemo logičku jedinicu)

Prije korištenja ovog modula, potrebno ga je konfigurirati s ispravnim postavkama koje ovise o uređaju s kojim se želi povezati. Način same konfiguracije objašnjen je kasnije u radu. Ako je modul u *data* načinu rada i ako ledica treperi na modulu, znači da se modul pokušava povezati s *bluetooth* uređajem, a ako svijetli, pa se ugasi nakon dvije sekunde, znači da je već povezan.



Sl. 3.3. HC-05 bluetooth modul.

3.1.4. OLED zaslon

Za prikazivanje parametara automobila, u ovome radu se koristi jednobojni OLED zaslon SSD1306, dijagonale 2.44 cm i rezolucije 128x64 piksela, prikazan na slici 3.4. Zaslon radi na naponu od 3.3 V ali zbog već ugrađenog regulatora napona, moguće ga je koristiti na napajanju do 5 V, što ga čini kompatibilnim s većinom Arduino mikrokontrolera. Ovi zaslone koriste SPI (engl. *Serial Peripheral Interface*) ili I2C (engl. *Inter-Integrated Circuit*) komunikacijske protokole, neki mogu koristiti oba. Zaslon korišten u ovom radu koristi I2C serijsku komunikaciju. Modul sadrži 4 nožice: VCC, GND, SDA i SCL. VCC služi za napajanje zaslona. GND služi za njegovo uzemljenje. SDA i SCL nožice služe za serijsku komunikaciju putem I2C protokola i spajaju se na SDA i SCL nožice neke druge razvojne pločice [8]. Prosječna potrošnja ovih zaslona je 20 mA, a ovisi o tome koliko je piksela na zaslonu osvjetljeno, što ga čini učinkovitijim od ostalih LED zaslona na tržištu.

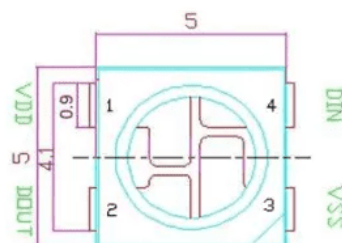


Sl. 3.4. OLED zaslon.

3.1.5. WS2812B LED

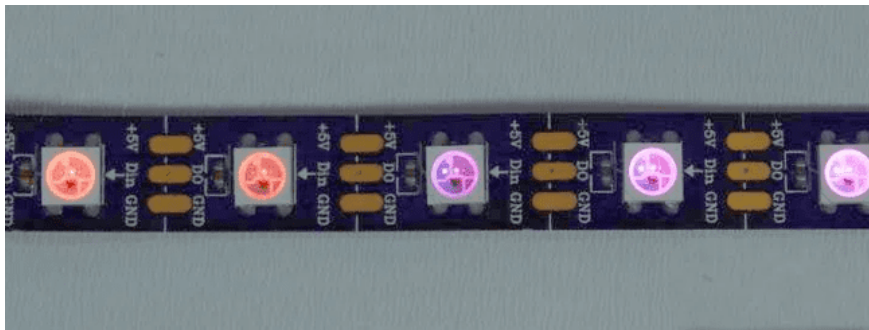
Za potrebe izrade ovog rada korištene su RGB (engl. *Red Green Blue*) LE diode za ambijentalnu rasvjetu unutar automobila. LE diode su poluvodičke komponente koje emitiraju svjetlost kada kroz njih prođe električna struja. Sastoje se od dva poluvodička sloja (P i N). Svjetlost nastaje kada se slobodni elektroni iz N sloja rekombiniraju sa šupljinama iz P sloja na P-N barijeri, što rezultira otpuštanjem energije u obliku svjetlosti [9].

Za razliku od LE dioda, postoje RGB LE diode, koje se sastoje od tri diode spojene u jednu, što im omogućava dobivanje gotovo svih boja, kombinacijom crvene, zelene i plave boje. WS2812B je upravljački integrirani sklop koji upravlja takvim diodama u obliku 5050 paketa (slika 3.5.). WS2812B integrirani sklop upravlja pojedinom ledicom putem jednog ulaza koristeći NRZ (engl. *Non-Return-to-Zero*) komunikacijski protokol.



Sl. 3.5. WS2812B LED.

WS2812B ledice najčešće dolaze u obliku trake (slika 3.6.), gdje su sekvencijalno spojene jedna na drugu, tako da svaka ima izlaz (*Dout*) spojen na ulaz sljedeće (*Din*). Upravljanje svakom ledicom se ostvaruje konstantnim slanjem podataka na podatkovnoj liniji u binarnom obliku. Svaka WS2812B ledica sprema prvih 24 bita (jedan bajt za svaku boju), te ostale bitove prosljeđuje ostalim ledicama u nizu, sve dok se ne dođe do kraja. Dolaskom na kraj, izvršava se reset, te se proces ponovno izvršava za sljedeću iteraciju. Svaka ledica može maksimalno koristiti 60 mA, te je bitno omogućiti dovoljno jak izvor napajanja s obzirom na količinu ledica koju se želi koristiti [10].



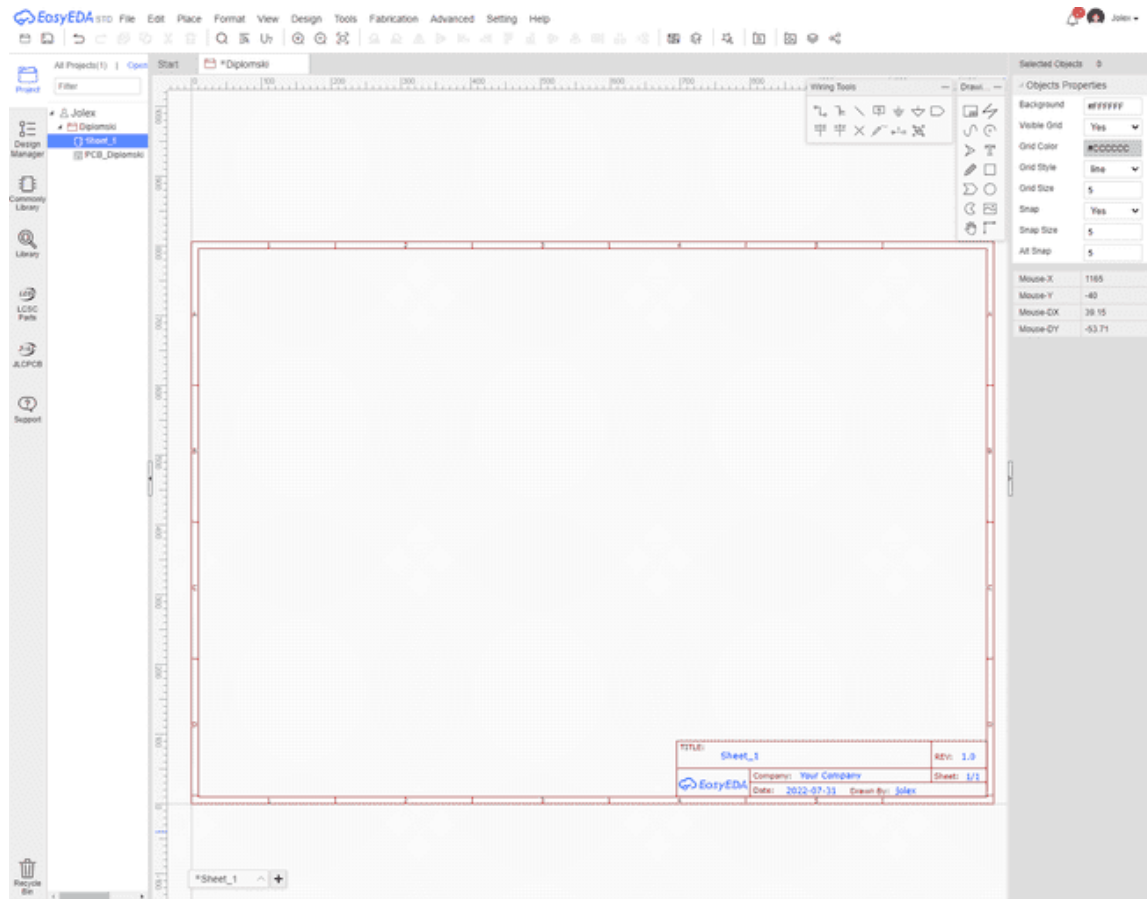
Sl. 3.6. WS2812B LED traka.

3.1.6. EasyEDA

EasyEDA je paket alata temeljen na web-u namijenjen za elektroničku automatizaciju dizajna (engl. *Electronic Design Automation*) PCB-a (engl. *Printed Circuit Board*). Osim što se u njemu mogu dizajnirati tiskane pločice, ima mogućnost pravljenja shema, njihove simulacije i dijeljenja među korisnicima. Besplatan je za korištenje i jako je jednostavan i intuitivan. Podržava LCSC komponente za odabir prilikom dizajniranja, uz velik broj komponenata dodanih od strane korisnika. Također ima integriran JLCPCB *service* za izradu tiskanih pločica [11].

Prilikom prijave na njihovoj stranici, u izborniku je dostupna opcija *EasyEda Designer*. Odabirom te opcije otvara se korisnikov *workspace* (slika 3.7.), gdje se pod *File* izbornikom može napraviti novi projekt. Prilikom kreiranja projekta, potrebno je napraviti novu mapu u kojoj je potrebno dodati novu *Schematic* datoteku. U *Schematic* datoteci se mogu jednostavno dodati već gotove komponente iz *Commonly library-a* na lijevoj izborničkoj traci. Ako nema određenih električnih komponenti u *Commonly library*, odabirom *Library* -> *EasyEDA* -> *PCB Module* -> *User Contributed* i upisivanjem imena tražene komponente, prikazat će se izlistanja komponenti pod tim nazivom dodanih od strane raznih korisnika. Kada su sve komponente dodane, potrebno ih je samo međusobno povezati. Kada su komponente ispravno povezane, potrebno je na alatnoj traci odabrati *Design* -> *Convert Schematic to PCB*, čime će se generirati PCB datoteka. U toj datoteci komponente su već spojene, te ih je potrebno samo posložiti i odrediti dimenzije tiskane pločice.

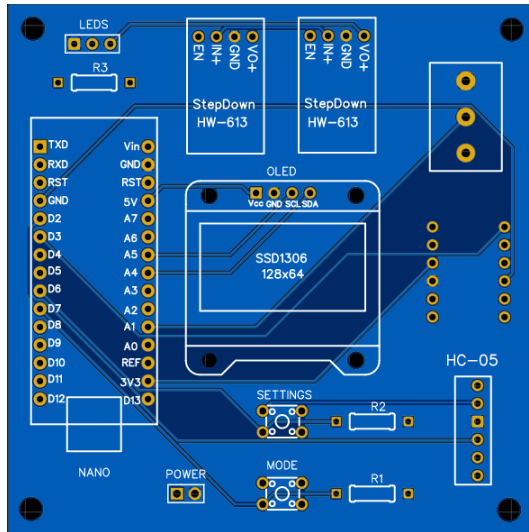
EasyEDA ima jako korisnu opciju za automatsko usmjeravanje veza među komponentama, što korisniku olakšava posao. Nakon što je PCB datoteka gotova, potrebno je na alatnoj traci stisnuti na *Fabrication* -> *PCB Fabrication File (Gerber)*, što će generirati Gerber datoteku. Gerber datoteku potrebno je učitati na stranici na kojoj se naručuje izradba tiskanih pločica.



Sl. 3.7. EasyEDA sučelje.

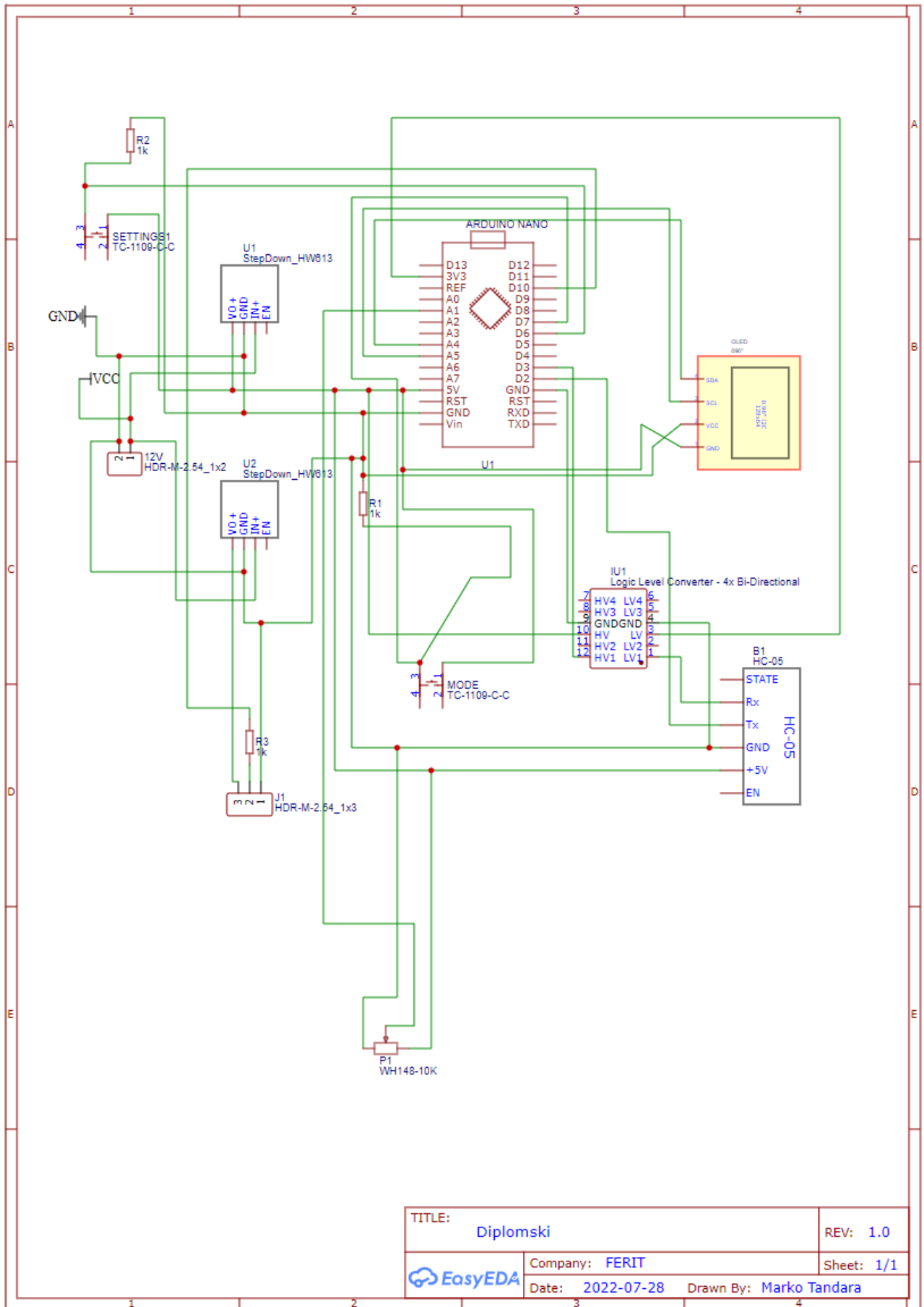
3.1.7. Tiskana pločica


Tiskana pločica je dizajnirana pomoću EasyEDA alata prethodno objašnjenog. Kako je JLCPCB servis integriran u EasyEDA alatu, za potrebe ovoga rada, tiskana pločice je naručena putem njihove stranice. Na slici 3.8. može se vidjeti konačan dizajn tiskane pločice.



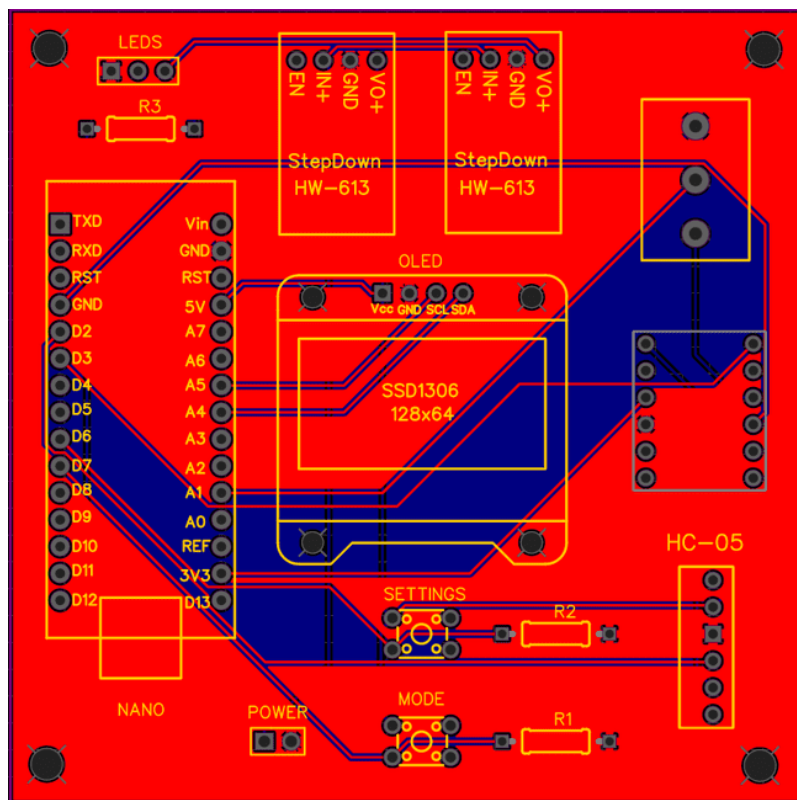
Sl. 3.8. Prikaz tiskane pločice.

Na slici 3.9. prikazan je shematski nacrt uređaja, gdje su crvenim okvirima predstavljene sve komponente, a zelenim linijama veze između njih. Koristeći tu shemu generirana je PCB datoteka čiji je prikaz vidljiv na slici 3.10. za gornji sloj i slika 3.11. za donji sloj. S obje strane je dodan bakreni sloj za uzemljenje kako bi se smanjile smetnje na komponentama.

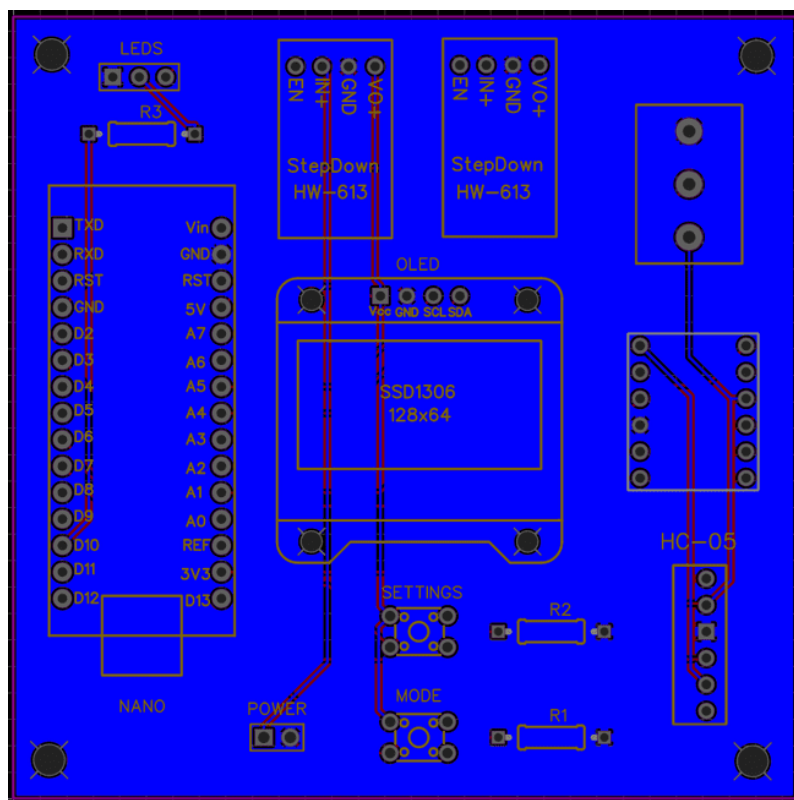


TITLE: Diplomski		REV: 1.0
	Company: FERIT	Sheet: 1/1
	Date: 2022-07-28	Drawn By: Marko Tandara

Sl. 3.9. Shematski prikaz tiskane pločice.

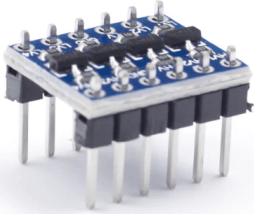
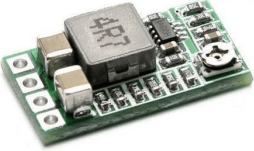




Sl. 3.10. Prikaz gornjeg sloja tiskane pločice.



Sl. 3.11. Prikaz donjnjeg sloja tiskane pločice.

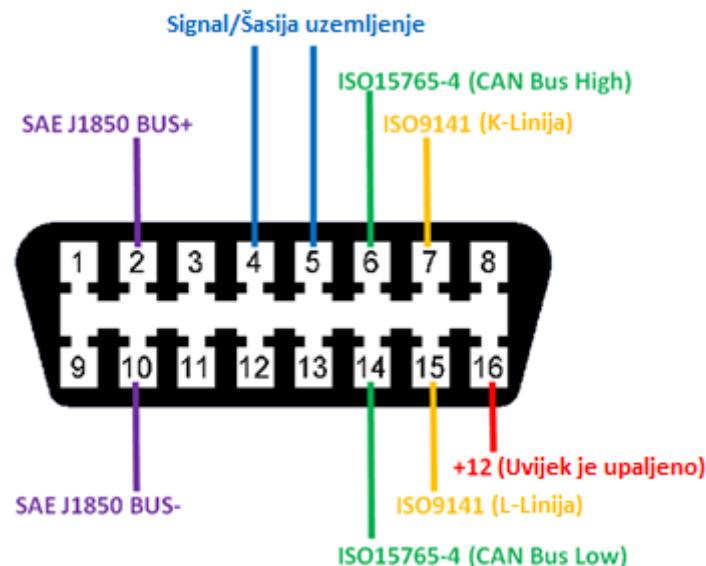
3.1.8. Elektroničke komponente

 <p>Sl. 3.12. I2C Logic Level pretvarač.</p>	<p>U radu je korišten četvero-kanali dvosmjerni I2C Logic Level pretvarač napona, za prilagođavanje naponskih razina s 5V na 3.3V. Ovu komponentu je potrebno spojiti na referentne vrijednosti napona s obje strane [12].</p>
 <p>Sl. 3.13. HW-613.</p>	<p>HW-613 je podesivi istosmjerni pretvarač napona, koji snižava napon s ulaza, a povećava struju na izlazu. Snižava 4.5V-24V na 0.8V-17V ili na fiksne vrijednosti (1.8V, 2.5V, 3.3V, 5V, 9V, 12V) što se postiže kratkim spajanjem željene vrijednosti na poleđini komponente [13].</p>
 <p>Sl. 3.14. 10 kΩ Potenciometar.</p>	<p>Potenciometar je električna komponenta koja služi kao podesivi djeliteľ napona. Najčešće se koriste dvije vrste potenciometara: linearni ili logaritamski. U ovom radu se koristi linearni potenciometar vrijednosti 10 kΩ [14].</p>
 <p>Sl. 3.15. Tipkalo.</p>	<p>Tipkalo je najčešće korištena sklopka u elektronici. Potrebno ga je spojiti preko <i>pull-down</i> otpornika na uzemljenje. Najčešće ga se koristi kao aktivator, prekidač ili brojač [15].</p>

3.2. Dijagnostika automobila

3.2.1. OBD

OBD (engl. *On-Board Diagnostics*) je standardni protokol koji je korišten u većini komercijalnih vozila za dohvaćanje dijagnostičkih informacija vozila. Informacije generira upravljačka jedinica vozila (engl. *ECU - Electronic Control Unit*) ili upravljački modul motora unutar vozila (engl. *ECM - Engine Control Module*). Nekoliko organizacija postavilo je temelje za ovaj standard 1960-ih godina, uključujući CARB (engl. *California Air Resources Board*), SAE (engl. *Society of Automotive Engineers*), ISO (engl. *International Organization for Standardization*) i EPA (engl. *Environmental Protection Agency*). Prije standardizacije ovog protokola, proizvođači su stvarali vlastite sustave. Alati svakog proizvođača (a ponekad i modeli istog proizvođača) imali su vlastitu vrstu konektora, zahtjeve elektroničkog sučelja, itd. U 1996. g. dolazi do novije verzije OBD standarda (OBD-II) koji postaje obavezan za sve automobile proizvedene u SAD-u. Kasnije 2001. g. postaje obavezan standard za sva vozila na benzin u EU, a 2003. g. postaje obavezan za sva vozila na dizel u EU (EOBD). OBD-II omogućuje pristup informacijama o statusu i dijagnostičkim kodovima kvarova (engl. *DTC - Diagnostic Trouble Codes*) za pogonski sklop i sustave kontrole emisija. OBD-II sučelje koristi standardiziranu utičnicu SAE J1962 prikazanu na slici 3.16., gdje je svaki pin označen imenom protokola za koji se koristi [16].



Sl. 3.16. OBD-II utičnica.

U suštini postoje pet vrsta komunikacijskih protokola koji se koriste za OBD-II sučelje.

- ISO15765-4 (CAN-BUS) – koristi nožice 6 i 14. Postoje četiri vrste ovog protokola, a razlikuju se po brzini i duljini identifikatora.
- ISO14230-4 (KWP2000)
- ISO9141-2
- SAE J1850 VPW
- SAE J1850 PWM

U ovome radu pošto je korišten *VW Golf VII* automobil za testiranje uređaja, korišten je ISO15765-4 (11bit, 500kBaud) protokol, jer je taj protokol implementiran na tom autu. ISO15765-4 je u sadašnjem vremenu najkorišteniji dijagnostički protokol u komercijalnim motornim vozilima. Naziva se i CAN (engl. *Control Area Network*) protokol, zato što koristi CAN sabirnicu za fizički i podatkovni sloj [17].

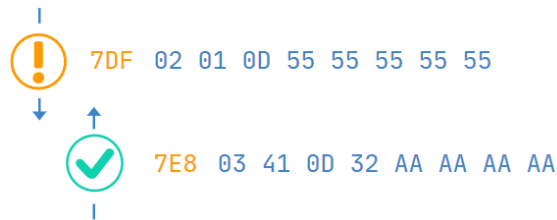
3.2.2. ELM327

ELM327 je dijagnostički uređaj, proizveden od strane ELM Electronics koji putem OBD-II protokola omogućava očitavanje dijagnostičkih grešaka i raznih parametara. Podržava sve ranije navedene protokole. Uređaj korišten u ovom radu (slika 3.17.), povezuje se bluetooth-om s nekim drugim uređajem koji te podatke može prikazati. U radu je povezan s Arduino Nanom koji sadrži zaslon na kojem se ispisuju ti podatci [18].



Sl. 3.17. ELM 327 dijagnostički uređaj.

Putem ELM327 dijagnostičkog uređaja šalju se zahtjevi s odgovarajućim modom i PID (engl. *Parameter ID*) kodom definiranih J1979 standardom. Primjer OBD-II poruke zahtjeva/odgovara prikazan je na slici 3.18. u heksadekadskom obliku za dohvaćanje trenutnog stanja parametra brzine vozila .



Sl. 3.18. Zahtjev/Odgovor struktura poruke za brzinu vozila.

Poruka se sastoji od:

- Identifikatora – označava radi li se o zahtjevu ili odgovoru (7DF ili 7E8)
- Dužine poruke – označava koliko narednih bajtova u poruci sadrži podatke, za zahtjev naredna 2 bajta (ostali 55 bajtovi su prazni), a za odgovor 3 bajta (ostali AA bajtovi su prazni)
- Mode – označava *mode/service*, postoji 10 modova (01-0A), u ovom radu korišten je 01 mode, jer on podržava PID kodove za trenutne parametre vozila. Kod odgovora modovi počinju s 4 (41-4A)
- PID – za svaki parametar čiji status se dohvaća, postoji jedinstveni PID ovisno o modu, npr. 0D u modu 01 označava trenutno stanje brzine vozila. Neki proizvođači automobila mogu implementirati vlastite PID kodove za neke druge parametre putem nekih drugih protokola
- Ostali bajtovi se odnose na same vrijednosti parametra, gdje je za neke vrijednosti parametara potrebno primijeniti dodatne formule za njihovu pretvorbu u smislene podatke (u slučaju gornje poruke potrebno je hex vrijednost 0x32 pretvoriti u dec vrijednost koja iznosi 50, što je ujedno vrijednost u kph za brzinu vozila u trenutku kada je zahtjev primljen).

U tablici 3.2. prikazana su tri PID koda korištena u radu u 01 modu. Svaki zahtjev s određenim PID kodom povratno vraća odgovor sa dužinom podatka u bajtovima, gdje je svaki bajt kada se pretvori iz heksadekadskog u decimalni oblik, redosljedom predstavljen jednim slovom (A, B, C, D). Decimalne vrijednosti bajtova predstavljene slovom ubacuju se u formule, prema kojima dobivamo smislene podatke. Svi ostali standardni PID kodovi u 01 modu podržani od strane ELM327 adaptera, nalaze se u prilogu.

Tab. 3.2. PID kodovi korišteni u radu

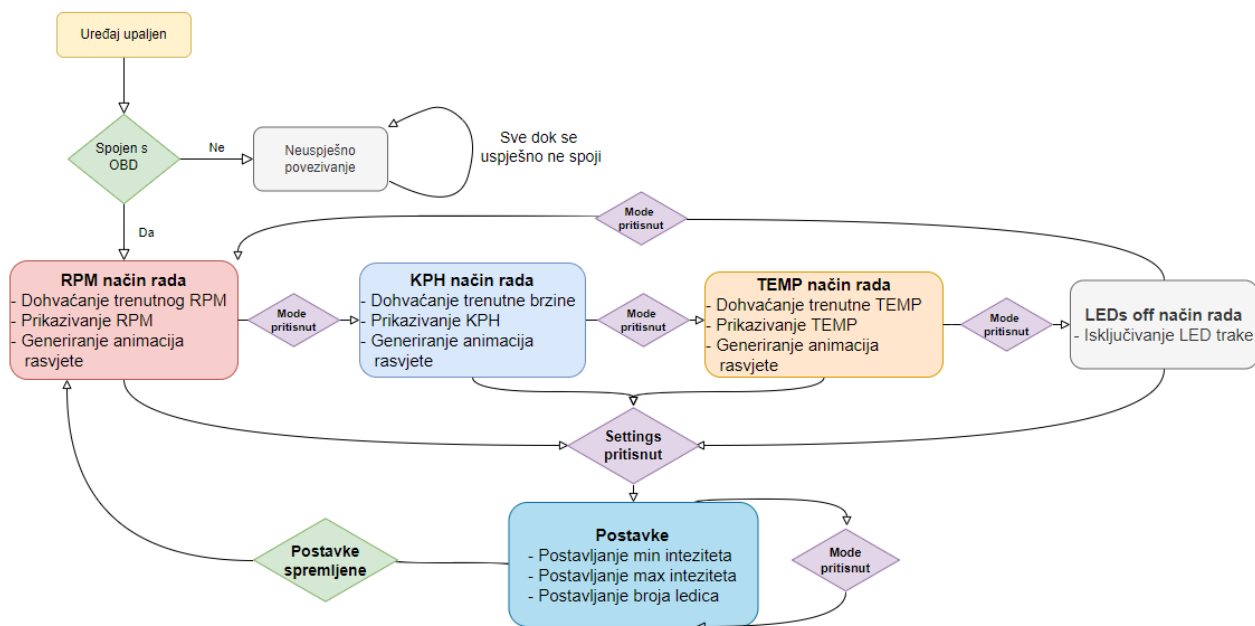
PID (hex)	Dužina (bajt)	Opis	Formula	Minimalna vrijednost	Maximalna vrijednost	Mjerna jedinica
0C	2 (A, B)	Broj okretaja motora	$\frac{256A + B}{4}$	0	16,383.75	rpm
0D	1 (A)	Brzina vozila	A	0	255	km/h
05	1 (A)	Temperatura rashladne tekućine	A - 40	-40	215	°C

3.3. Programaska podrška

Programski kôd za uređaj u ovome radu, pisan je koristeći Arduino IDE razvojno okruženje. Na početku programskog kôda definirane su sve biblioteke koje su potrebne za rad sa svim komponentama ranije navedene. Biblioteke se prvo moraju dodati putem *Manage libraries* opcije unutar Arduino IDE-a. Biblioteke potrebne za rad s komponentama u ovome radu su:

- <SoftwareSerial.h> - biblioteka koja omogućava serijsku komunikaciju na digitalnim nožicama
- "ELMdino.h" - biblioteka koja omogućava rad s ELM327 modulom
- <Fastled.h> - biblioteka za rad s WS2812B LED trakom
- <U8x8lib.h> - biblioteka koja omogućava ispisivanje teksta na OLED zaslon
- <Bounce2.h> - biblioteka koja omogućava debouncing tipkala
- <EEPROM.h> - biblioteka koja omogućava rad s EEPROM (engl. *Electrically Erasable Programmable Read-Only Memory*) na Arduino Nano mikrokontroleru.

U narednom dijelu, bit će opisani svi važniji dijelovi programskog rješenja, a cijeli kôd će biti dostupan na CD-u. Na slici 3.19. je prikazan dijagram toka programa.



Sl. 3.19. Dijagram toka programa.

3.3.1. Kreiranje objekata i upostava komunikacije

Nakon definiranja svih potrebnih biblioteka, napravljeni su njihovi objekti, koji mogu pozivati funkcije koje su definirane unutar biblioteka (slika 3.20.). Nakon objekata deklarirane su početne varijable za rad s funkcijama.

```
#define NUM_LEDS 20
CRGBArray<NUM_LEDS> leds;
U8X8_SSD1306_128X64_NONAME_SW_I2C u8x8(SCL, SDA, U8X8_PIN_NONE);
SoftwareSerial mySerial(2, 3);
#define ELM_PORT mySerial
ELM327 myELM327;
```

Sl. 3.20. Definiranje objekata.

Da bi uređaj uopće radio, najbitnije je da se funkcija za inicijalizaciju veze s OBD protokolom uspješno izvrši (slika 3.21.). U toj funkciji na myELM327 objektu, poziva se *begin* funkcija koja kao argumente prima četiri parametra: objekt za serijsku komunikaciju, *bool* parametar koji ako ima vrijednost *true* omogućava očitavanje grešaka, pauza između poziva u ms i broj protokola koji auto podržava. Na zaslonu uređaja ispisuje se greška pri povezivanju sve dok ta funkcija ne vrati istinitu vrijednost.

```

if (!myELM327.begin(ELM_PORT, true, 300, '6'))
{
  u8x8.clearDisplay();
  u8x8.setCursor(4, 1);
  u8x8.print("Failed to");
  u8x8.setCursor(5, 3);
  u8x8.print("connect");
  u8x8.setCursor(4, 5);
  u8x8.print("with OBDII");
  Serial.println("Couldn't connect to OBD scanner");
  while(!myELM327.begin(ELM_PORT, true, 300, '6'));
}

```

Sl. 3.21. Uspostava veze s CAN busom putem OBD.

3.3.2. Korištenje EEPROM memorije

Prilikom pokretanja uređaja u *setup()* funkciji, dohvaćaju se vrijednosti minimalnog i maksimalnog intenziteta svjetlosti, te broj ledica koji se želi osvijetliti iz EEPROM memorije uređaja (slika 3.22.). Vrijednosti se dohvaćaju s određene adrese te se spremaju u globalne varijable, koje se koriste kasnije u programu za osvjetljenje u pojedinom načinu rada. Na slici 3.23. je priazan primjer korištenja EEPROM memorije za postavljanje vrijednosti tih varijabli kada je uređaj u settings načinu rada. Za svaku varijablu odabire se njezina vrijednost okretanjem potenciometra, te pritiskom mode tipkala sprema se trenutna vrijednost u EEPROM memoriju, te se brojač povećava, što u sljedećoj iteraciji omogućava mijenjanje druge varijable, sve dok sve tri nisu postavljene.

```

EEPROM.get(0, minBrightness);
EEPROM.get(2, maxBrightness);
EEPROM.get(4, ledsNumb);

```

Sl. 3.22. Dohvaćanje vrijednosti iz EEPROM memorije.

```

if (btnModeForSettingsPushCounter == 1 && settingsMode == true) {
  potentionValue = analogRead(A1);
  potentionMapped = map(potentionValue, 0, 1023, 0, 255);
  EEPROM.put(0, potentionMapped);
  EEPROM.get(0, minBrightness);
  drawSettings();
}
else if (btnModeForSettingsPushCounter == 2 && settingsMode == true) {
  EEPROM.get(0, minBrightness);
  potentionValue = analogRead(A1);
  potentionMapped = map(potentionValue, 0, 1023, minBrightness, 255);
  EEPROM.put(2, potentionMapped);
  EEPROM.get(2, maxBrightness);
  drawSettings();
}
else if (btnModeForSettingsPushCounter == 3 && settingsMode == true) {
  potentionValue = analogRead(A1);
  potentionMapped = map(potentionValue, 0, 1023, 0, NUM_LEDS);
  EEPROM.put(4, potentionMapped);
  EEPROM.get(4, ledsNumb);
  drawSettings();
}

```

Sl. 3.23. Mijenjanje i spremanje postavki u EEPROM memoriju.

3.3.3. Provjera statusa tipkala

Ako je veza uspostavljena, na zaslonu se ispisuje poruka o uspješnom povezivanju, te uređaj kreće s izvođenjem *loop* petlje. Odmah na početku petlje izvršavaju se funkcijski blokovi za vremensku validaciju stisnutog settings tipkala (slika 3.24.). Ako je tipkalo pritisnuto dulje od 4 sekunde, postavlja se varijabla *settingsMode* u istinitu vrijednost, koja u nastavku petlje služi kao argument za izvršavanje funkcije za postavljanje *settings* načina rada.

```
currentMillis = millis();
int btnModeState = digitalRead(btnSettings);
if (btnModeState == HIGH && btnSettingsPreviousState == LOW && !btnModeStateLongPress) {
    minBtnSettingsLongPressMillis = currentMillis;
    btnSettingsPreviousState = HIGH;
    Serial.println("Button pressed");
}
btnSettingsPressDuration = currentMillis - minBtnSettingsLongPressMillis;
if (btnModeState == HIGH && !btnModeStateLongPress && btnSettingsPressDuration >= minBtnSettingsLongPressDuration) {
    btnModeStateLongPress = true;
    btnSettingsPushCounter = 0;
    u8x8.clearDisplay();
    settingsMode = true;
    Serial.println("Button long pressed");
}
if (btnModeState == LOW && btnSettingsPreviousState == HIGH) {
    btnSettingsPreviousState = LOW;
    btnModeStateLongPress = false;
    Serial.println("Button released");
}
previousBtnSettingsMillis = currentMillis;
```

Sl. 3.24. Provjera pritiska Settings tipkala.

Nakon funkcije za provjeru stanja settings tipkala, poziva se funkcija za provjeru stanja mode tipkala (slika 3.25.). Ova funkcija služi kao brojač, putem kojeg se kasnije u programu poziva određen način rada. Podržava provjeru debouncinga, što znači da unutar specificiranog intervala, brojač neće promijeniti vrijednost iako je tipka pritisnuta. Time se sprječava da mikroupravljač očita više pritisaka tipkala, iako je pritisnuto samo jednom.

```
void modeBtnCheck() {
    time_now = millis();
    while (millis() < time_now + period) {
        debouncer.update();
    }
    int value = debouncer.read();
    btnModeState = value;
    if (btnModeState != previousBtnModeState) {
        if (btnModeState == HIGH) {
            btnModePushCounter++;
            btnSettingsPushCounter++;
            u8x8.clearDisplay();
            for (int i = 0; i < 20; i++) {
                leds[i] = CHSV(0, 0, 0);
            }
            FastLED.show();
        }
    }
    if (btnModePushCounter == 4) {
        btnModePushCounter = 0;
    }
    if (btnSettingsPushCounter == 5) {
        btnSettingsPushCounter = 0;
    }
}
```

Sl. 3.25. Provjera pritiska Mode tipkala.

3.3.4. Određivanje određenog načina rada

Nakon očitavanja stanja tipkala, program ulazi u *if* funkcijski blok, koji ovisno o vrijednosti brojača mode tipkala i statusu *settingsMode* varijable određuje način rada uređaja (slika 3.26.). Uređaj sadržava četiri načina rada koji su ranije navedeni.

```
if (btnModePushCounter == 0 && settingsMode == false) {
  modeRPM();
}
else if (btnModePushCounter == 1 && settingsMode == false) {
  modeKPH();
}
else if (btnModePushCounter == 2 && settingsMode == false) {
  modeTEMP();
}
else if (btnModePushCounter == 3 && settingsMode == false) {
  modeLedsOFF();
}
```

Sl. 3.26. Određivanje načina rada uređaja.

Svaki od prva tri načina rada uređaja ima specifičnu funkciju za animaciju osvjetljenja, ispisivanje parametara na zaslonu i slanje PID kodova (slika 3.27.). Četvrti način rada služi za gašenje osvjetljenja, te ne šalje nikakve zahtjeve. Svaki od tri načina šalju zahtjeve putem *myELM327* objekta, koristeći PID kodove za svoje parametre. *Elmduino.h* biblioteka sakriva kompleksnost slanja PID kodova unutar funkcija koje se pozivaju putem objekta. Nakon slanja zahtjeva svaki način rada, provjerava status tih zahtjeva. Ako je OBD uspješno primio zahtjev i poslao vrijednosti trenutačnih parametara uređaju, te vrijednosti se ispisuju na zaslonu i u obliku animacija osvjetljenja. U slučaju da zahtjev nije uspješno zaprimljen, uređaj će na zaslonu ispisati kako vrijednosti trenutačnih parametara nisu zaprimljene, te će se led traka ugasiti, sve dok zahtjev uspješno ne prođe. Animacije osvjetljenja, koje se ostvaruju promjenom vrijednosti parametara za svaki način rada su pokazani u poglavlju testiranja uređaja.

```

void modeKPH() {
    float tempKPH = (uint32_t)myELM327.kph();
    if ((myELM327.nb_rx_state == ELM_SUCCESS))
    {
        kph = tempKPH;
        mapKPH = map(kph, 0, 70, 0, ledsNumb);
        sensorValue = analogRead(A1);
        mapSensor = map(sensorValue, 0, 1023, 0, 255);
        for (int i = 0; i < mapKPH; i++) {
            leds[i] = CHSV(mapSensor, 255, 255);
            FastLED.show();
        }
        for (int i = mapKPH; i < ledsNumb; i++) {
            leds[i].fadeToBlackBy(255);
        }
        drawKPH(kph);
        rpm = 0;
    }
}

```

Sl. 3.27. RPM mode.

```

void modeRPM() {
    float tempRPM = myELM327.rpm();
    if (myELM327.nb_rx_state == ELM_SUCCESS)
    {
        rpm = (uint32_t)tempRPM;
        mapRPM = map(rpm, 0, 2000, minBrightness, maxBrightness);
        sensorValue = analogRead(A1);
        mapSensor = map(sensorValue, 0, 1023, 0, 255);
        for (int i = 0; i < ledsNumb; i++) {
            leds[i] = CHSV(mapSensor, 255, mapRPM);
            FastLED.show();
        }
        draw(rpm);
        temp = 0;
    }
}

```

Sl. 3.28. KPH mode.

```

void modeTEMP() {
    float tempTemp = myELM327.engineCoolantTemp();
    if ((myELM327.nb_rx_state == ELM_SUCCESS))
    {
        temp = tempTemp;
        mapTEMP = map(temp, 10, 90, 40, 0);
        for (int i = 0; i < ledsNumb; i++) {
            leds[i] = CHSV(mapTEMP, 255, 255);
            FastLED.show();
        }
        drawTEMP(temp);
        kph = 0;
    }
}

```

Sl. 3.29. TEMP mode.

```

void modeLedsOFF() {
    for (int i = 0; i < NUM_LEDS; i++) {
        leds[i].fadeToBlackBy(255);
    }
    FastLED.show();
    u8x8.setCursor(4, 4);
    u8x8.print("Leds off!");
}

```

Sl. 3.30. Leds OFF mode.

3.3.5. Izbornik za postavke uređaja

Kada se pritisne tipkalo Settings i drži se pritisnuto više od 4 sekunde, promijenit će se vrijednost *settingsMode* varijable, čime će program zanemariti sva četiri načina rada i otvorit će se Settings izbornik. Settings način rada prolazi kroz pet iteracija *If/ElseIf* petlje (slika 3.31.). Prva služi za prikaz trenutnih vrijednosti parametara spremljenih u EEPROM memoriji mikroupravljača (minBrightness, maxBrightness, ledsNumb). Druga služi za promjenu minBrightness parametra okretanjem potenciometra (u vrijednosti od 0 do 255) i spremanje te vrijednosti u EEPROM memoriji klikom mode tipkala. Treća iteracija radi isto kao i druga, samo za parametar maxBrightness (vrijednost maxBrightnessa ne može biti manja od minBrightnessa). Četvrta radi isto što i druga i treća za parametar ledsNumb (u vrijednosti od 0 do maksimalnog broja ledica). Nakon pritiska tipkala mode u četvrtoj iteraciji, na zaslonu se prikazuju promijenjene vrijednosti i nakon četiri sekunde, uređaj se vraća u RPM način rada.

```

if (btnModeForSettingsPushCounter == 0 && settingsMode == true) {
    EEPROM.get(0, minBrightness);
    EEPROM.get(2, maxBrightness);
    EEPROM.get(4, ledsNumb);
    drawSettings();
}

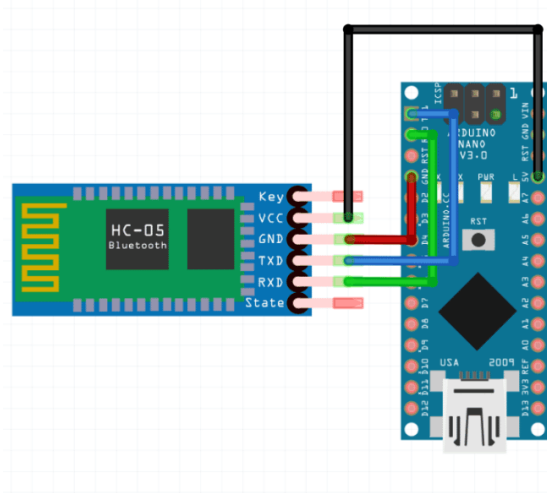
if (btnModeForSettingsPushCounter == 1 && settingsMode == true) {
    potentiometerValue = analogRead(A1);
    potentiometerMapped = map(potentiometerValue, 0, 1023, 0, 255);
    EEPROM.put(0, potentiometerMapped);
    EEPROM.get(0, minBrightness);
    drawSettings();
}
else if (btnModeForSettingsPushCounter == 2 && settingsMode == true) {
    EEPROM.get(0, minBrightness);
    potentiometerValue = analogRead(A1);
    potentiometerMapped = map(potentiometerValue, 0, 1023, minBrightness, 255);
    EEPROM.put(2, potentiometerMapped);
    EEPROM.get(2, maxBrightness);
    drawSettings();
}
else if (btnModeForSettingsPushCounter == 3 && settingsMode == true) {
    potentiometerValue = analogRead(A1);
    potentiometerMapped = map(potentiometerValue, 0, 1023, 0, NUM_LEDS);
    EEPROM.put(4, potentiometerMapped);
    EEPROM.get(4, ledsNumb);
    drawSettings();
}
else if (btnModeForSettingsPushCounter == 4 && settingsMode == true) {
    drawSettings();
}
}

```

Sl. 3.31. Izbornik za postavke uređaja.

3.4. Testiranje

Prije samog testiranja, potrebno je konfigurirati HC-05 modul da ispravno komunicira s ELM327 adapterom. Potrebno je učitati početnu skicu koja se sama generira pri pokretanju Arduino IDE-a na Arduino Nano i zatim spojiti HC-05 modul s Arduino Nanom prema shemi na slici 3.32. (na nožicama Tx i Rx ne smije ništa biti spojeno pri učitavanju kôda).



Sl. 3.32. HC-05 konfiguracija.

Nakon što je programski kôd učitani i komponente su spojene, potrebno je pritisnuti i držati tipku na HC-05 modulu pri uključivanju Arduino Nano-a na računalo. Ako je sve uspješno odrađeno, lampica na HC-05 modulu bi trebala svijetliti svake dvije sekunde, što znači da je modul u konfiguracijskom načinu rada. Zatim je potrebno otvoriti *serial monitor* u Arduino IDE-u i redoslijedom ukucavati sljedeće naredbe (bez navodnika). Svaka naredba bi trebala povratno vraćati "OK", što znači da je uspješno izvršena.

- "AT" – Provjerava da li je modul u konfiguracijskom načinu rada, ako povratno vraća neke čudne znakove, potrebno je provjeriti da li je postavljen odgovarajući *baud rate*
- "AT+ROLE=1" – Postavljanje HC-05 modula kao glavne jedinice
- "AT+CMODE=0" – Postavljanje HC-05 modula da se povezuje sa specifičnom adresom
- "AT+BIND=AABB,CC,112233" – Definiranje adrese uređaja s kojom se želi povezati, svaki uređaj ima specifičnu adresu, navedena je specifična za ELM327 adapter koji se koristi u ovome radu
- "AT+PAIR= AABB,CC,112233,20" – Postavljanje uparivanja s pauzom od 20 ms
- "AT+UART= 38400,0,0" – Postavljanje *baud rate-a* za serijsku komunikaciju

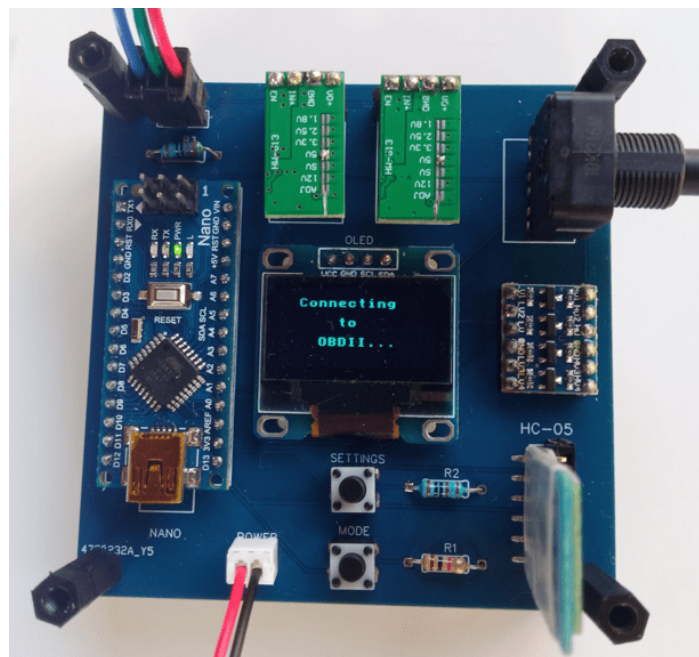
Nakon što su sve komponente ispravno zalemljene na svoja odgovarajuća mjesta na PCB-u, programski kôd je učitani i HC-05 modul je ispravno podešen, uređaj je spreman za korištenje. Kroz sljedeći dio rada, pokazane su sve funkcionalnosti koje ovaj uređaj pruža.

Prvi korak je priključivanje ELM327 adaptera na OBD-II priključak unutar automobila. U svakom automobilu, OBD-II priključak se može nalaziti na različitom mjestu. U ovome radu, testiranje se obavlja na autu marke Volkswagen, modela Golf, generacije VII, pa se priključak nalazi ispod volana s lijeve strane vidljivo na slici 3.33.



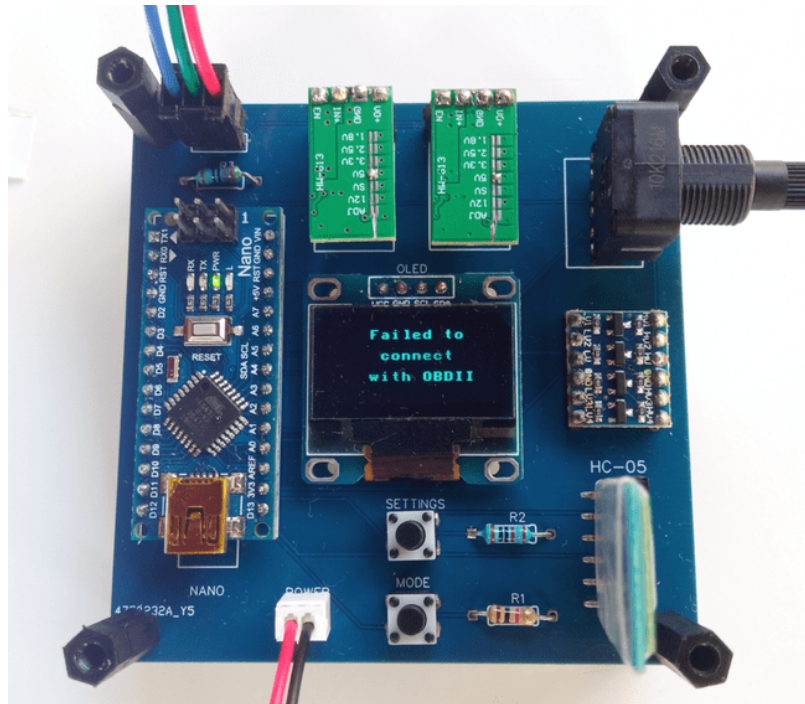
Sl. 3.33. ELM327.

Zatim je potrebno spojiti uređaj na napajanje od 12V u autu. Prilikom dodavanja kontakta okretanjem ključa u autu, uređaj se pali i odmah se pokušava spojiti s ELM327 adapterom putem *bluetootha* (slika 3.34.).



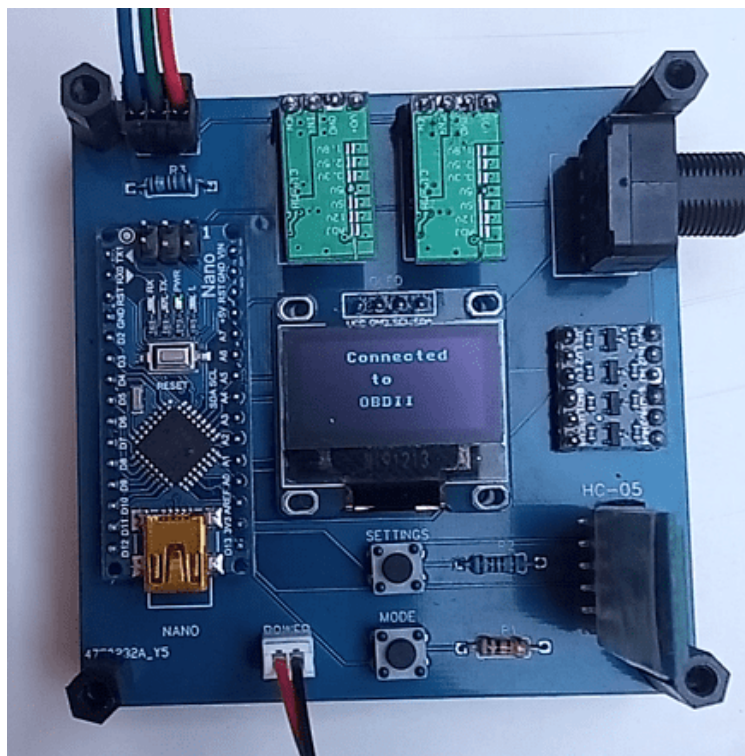
Sl. 3.34. Povezivanje uređaja s ELM327.

U slučaju da se uređaj ne uspije povezati s ELM327 adapterom unutar pet sekundi, na zaslonu se ispisuje poruka da je povezivanje neuspješno (slika 3.35.). U slučaju ispisivanja ove poruke, potrebno je uređaj restartirati, pritiskom *reset* tipke na Arduino Nano mikroupravljaču.



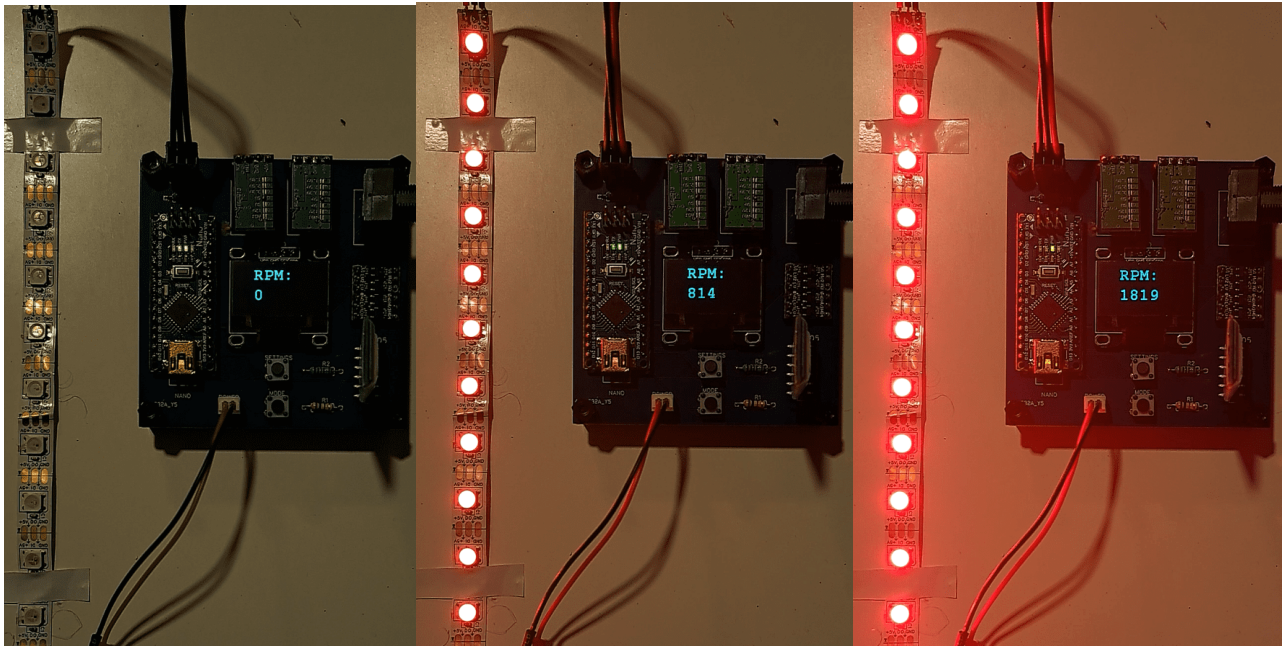
Sl. 3.35. Neuspješno povezivanje s ELM327 adapterom.

Ako se uređaj uspješno povezoao s ELM327 adapterom na zaslonu se prikazuje odgovarajuća poruka prikazana na slici 3.36.



Sl. 3.36. Uspješno povezivanje s ELM327 adapterom.

Nakon uspješnog povezivanja, kroz par sekundi na zaslonu se ispisuje RPM koji je početni način rada i pokazuje trenutni broj okretaja motora auta. Kako je u autu samo dodan kontakt i još nije upaljen motor, njegova vrijednost iznosi 0 (slika 3.37.). Prilikom paljenja motora, njegova vrijednost raste na nekih 800 rpm-a (slika 3.38.), koja je srednja vrijednost broja okretaja motora u stanju mirovanja. Uz rast rpm-a, može se primijetiti da raste intenzitet svjetla LED trake, koja je prethodno bila ugašena. Dodavanjem gasa u autu, raste broj okretaja motora preko 1800 rpm-a, pa tako i intenzitet svjetla led trake (slika 3.39.).

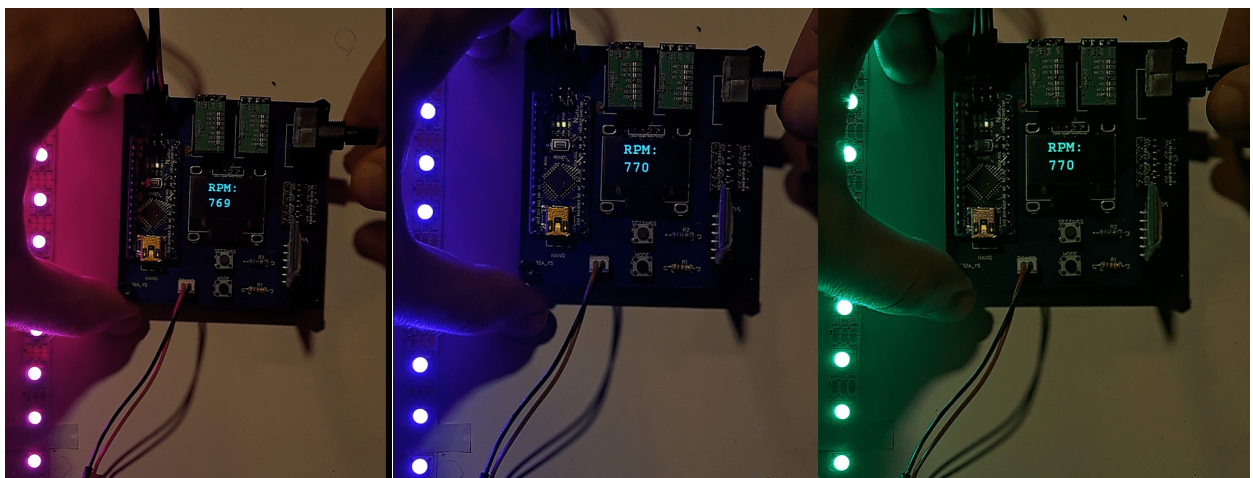


Sl. 3.37. Motor ugašen (0 rpm).

Sl. 3.38. Motor Idle (814 rpm).

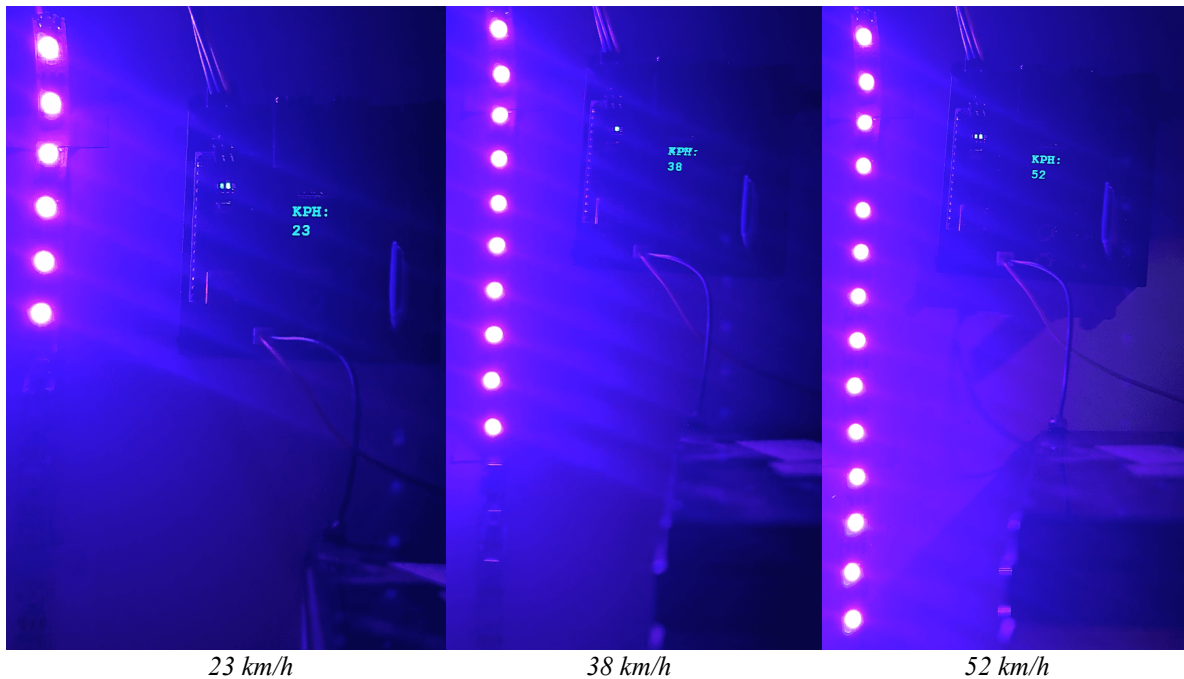
Sl. 3.39. Motor pod gasom (1819 rpm).

U ovome načinu rada, okretanjem potencijometra može se mijenjati boja osvjetljenja led trake (slika 3.40.).



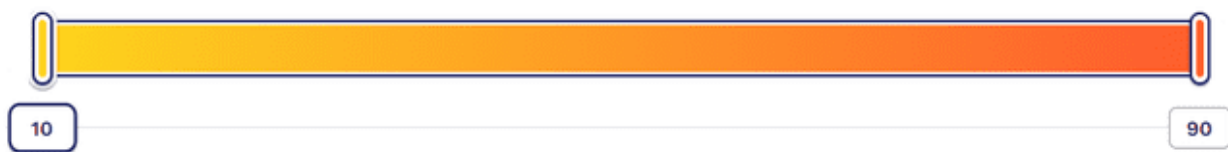
Sl. 3.40. Promjena boje osvjetljenja okretanjem potencijometara.

Na uređaju se nalazi tipka mode, koja služi za promjenu načina rada. Pritiskom te tipke, prelazimo iz RPM načina rada u KPH način rada, koji pokazuje trenutnu brzinu auta u km/h. Taj način rada ima drugačiju animaciju osvjetljenja nego RPM mode. Prilikom povećanja brzine povećava se broj ledica koji je osvijetljen umjesto njihovog intenziteta (slika 3.41.).



Sl. 3.41. Paljenje ledica povećanjem brzine.

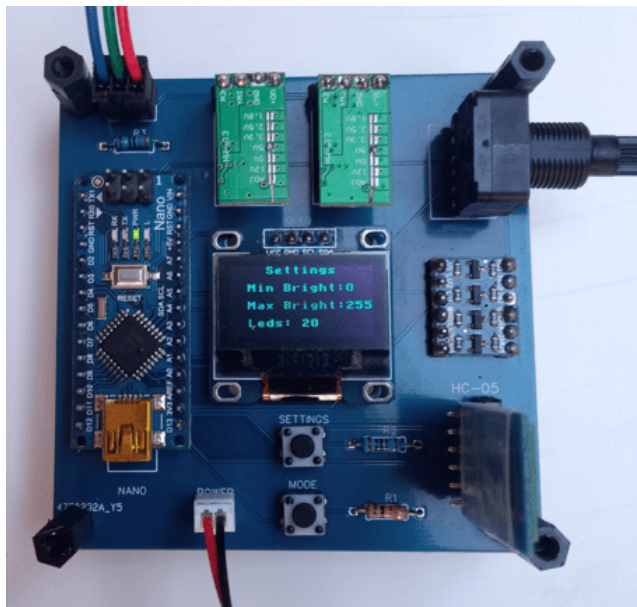
Ponovnim pritiskom tipke mode, prelazi se u TEMP način rada, koji na zaslonu prikazuje trenutnu temperaturu rashladne tekućine u stupnjevima Celzijusa. Njegova animacija za osvjetljenje radi tako da temperature od 10 do 90 prikazuje u spektru boja prikazanih na slici 3.42.



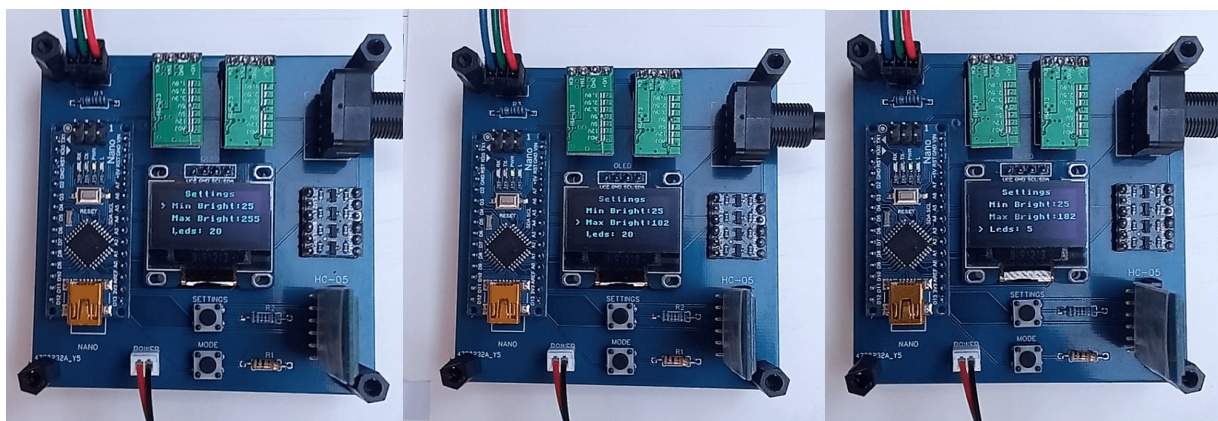
Sl. 3.42. Spektr boja temperature rashladne tekućine.

Na uređaju se može zapaziti još jedna tipka naziva Settings. Ova tipka služi za postavljanje broja ledica koji se želi koristiti, te određivanje njihovog maksimalnog i minimalnog intenziteta osvjetljenja u pojedinim načinima rada uređaja. To se postiže pritiskom i držanjem te tipke dulje od četiri sekunde, prilikom čega se otvara *Settings* izbornik (slika 3.43.). *Settings* izbornik prikazuje trenutne vrijednosti postavki koje su već pohranjene unutar EEPROM memorije mikroupravljača. Prednost korištenja ovakve memorije je to što vrijednosti ostaju pohranjene prilikom gašenja i paljenja uređaja. Pritiskom tipke *mode* u settings izborniku, mogu se promijeniti te vrijednosti (slika 3.44.). Vrijednosti mijenjamo okretanjem potencijometara, koji mijenja vrijednosti u sljedećem rasponu:

- Minimalan intenzitet – od 0 do 255
- Maksimalan intenzitet – od minimalnog intenziteta do 255
- Broj ledica – od 0 do 20 (optimalan broj ledica koje uređaj može napajati)



Sl. 3.43. Settings izbornik.



Sl. 3.44. Promjena postavki.

Kada se postavi broj ledica, pritiskom na tipku mode se ispisuje poruka da su definirane postavke promijenjene i sačuvane unutar EEPROM memorije. Nakon nekoliko sekundi uređaj se vraća u početni način rada RPM, te se može nastaviti s normalnim radom uređaja.

4. ZAKLJUČAK

Uređaj za upravljanje ambijentalnom led rasvjetom u autu putem OBD-II protokola je uspješno izrađen. Uređaj se sastoji od Arduino Nano mikroupravljačke jedinice koja komunicira s ELM327 *bluetooth* adapterom putem HC-05 bluetooth modula. Uređaj pruža mogućnost odabira između četiri načina rada ambijentalne led rasvjete (s obzirom na broj okretaja motora, brzine, temperature rashladne tekućine i mogućnost gašenja rasvjete). Pritiskom na tipku *mode*, rotiramo između navedenih načina rada, te ovisno o načinu rada koji je odabran, Arduino Nano šalje određene PID kodove OBD-II ELM327 adapteru. OBD-II adapter povratno šalje trenutne podatke ovisno o kodu koji je primio, te se ti podatci prikazuju na zaslonu uređaja i putem određenih animacija na led traci. Uređaj također ima tipku *settings*, pritiskom koje se omogućava unošenje broja ledica i podešavanje maksimalnog i minimalnog intenziteta osvjetljenja. S obzirom na način rada koji je odabran, korisnik putem potenciometra može odabrati željenu boju osvjetljenja između svih duginih boja.

Uređaj bi se mogao poboljšati tako da uvedemo veći broj animacija, što bi zahtijevalo veću memoriju mikroupravljača. Korištenjem kvalitetnijeg OBD-II adaptera, poboljšali bi brzinu kojom adapter vraća podatke, te bi tako dobili animacije bez kašnjenja, a isto tako i raznovrsne ostale podatke pomoću kojih se može implementirati nove animacije osvjetljenja. Korištenjem kvalitetnijih led traka moglo bi se proširiti osvjetljenje na vanjski dio automobila.

LITERATURA

- [1] „Yzg App“ [online]. Dostupno na: https://www.alibaba.com/product-detail/YZG-APP-and-Remote-Control-12V_1600437146814.html?spm=a2700.shop_plser.41413.18.2be173e4zZysNZ. [Pristupljeno: 4.8.2022.].
- [2] „Aura LED OPT7“ [online]. Dostupno na: <https://www.opt-7.com/aura-led-accent-light-kits/aura-interior-kits/>. [Pristupljeno: 5.8.2022.].
- [3] „SANLI LED“ [online]. Dostupno na: <https://azimom.shop/products/16w-rgbw-rolls-royce-roof-star-ceiling-lights-with-bluetooth-app-remote-control-sound-activated>. [Pristupljeno: 5.8.2022.].
- [4] „Arduino“ [online]. Dostupno na: <http://e-elektro.blogspot.com/2014/06/sto-je-arduino.html>. [Pristupljeno: 3.8.2022.].
- [5] „What is Arduino“ [online], 14-ruj-2015. Dostupno na: <https://stackabuse.com/what-is-arduino/>. [Pristupljeno: 3.8.2022.].
- [6] „Arduino Nano“ [online], 16-kol-2019. Dostupno na: <https://diy10t.com/arduino-nano-tutorial/>. [Pristupljeno: 3.8.2022.].
- [7] „HC-05“ [online]. Dostupno na: <https://components101.com/wireless/hc-05-bluetooth-module>. [Pristupljeno: 11.8.2022.].
- [8] „OLED I2c Display“ [online]. Dostupno na: <https://www.instructables.com/Monochrome-096-i2c-OLED-display-with-arduino-SSD13/>. [Pristupljeno: 3.8.2022.].
- [9] „LED“ [online]. Dostupno na: <https://www.rohm.com/electronics-basics/leds/what-are-leds>. [Pristupljeno: 12.8.2022.].
- [10] „WS2812B LED“ [online]. Dostupno na: <https://www.suntechleds.com/info/how-does-the-ws2812b-led-light-work-55934182.html>. [Pristupljeno: 12.8.2022.].
- [11] „EasyEDA“ [online]. Dostupno na: <https://www.watelectronics.com/easyeda-pcb-designing-circuit-simulation/>. [Pristupljeno: 25.8.2022.].
- [12] „I2C Logic Level Converter“ [online]. Dostupno na: <https://protosupplies.com/product/i2c-logic-level-converter-with-regulator-module/>. [Pristupljeno: 26.8.2022.].
- [13] „HW-613 Buck Converter“ [online]. Dostupno na: <https://hallroad.org/dc-adjustable-buck-converter-module-3a.html>. [Pristupljeno: 27.8.2022.].
- [14] „Potenciometar“ [online]. Dostupno na: <https://www.hwlibre.com/bs/potenciometar-sve-%C5%A1to-biste-trebali-znati/>. [Pristupljeno: 27.8.2022.].
- [15] „Tipkalo“ [online]. Dostupno na: <https://e-radionica.com/hr/blog/2016/07/26/tipkalo-push-button/>. [Pristupljeno: 28.8.2022.].
- [16] „What is OBDII“ [online]. Dostupno na: <https://www.geotab.com/blog/obd-ii/>. [Pristupljeno: 18.8.2022.].
- [17] „OBD2 protocols“ [online]. Dostupno na: http://www.obdtester.com/obd2_protocols. [Pristupljeno: 19.8.2022.].
- [18] „ELM327 Scanner“ [online]. Dostupno na: <https://carinfo24.com/blog/2020-01-08-what-is-elm327-scanner-and-how-to-use-it/>. [Pristupljeno: 21.8.2022.].

SAŽETAK

Zadatak ovoga rada je bio napraviti uređaj koji bi upravljao ambijentalnom led rasvjetom u automobilu putem OBD-II protokola. Uređaj komunicira s ELM327 OBD-II adapterom putem bluetootha i sastoji se od četiri načina osvjetljenja, koja se odabiru pritiskom tipke *mode*. Tri načina rada ovise o trenutačnim parametrima automobila (brzina, broj okretaja motora i temperatura rashladne tekućine), te četvrti način služi za gašenje osvjetljenja. Svaki način rada ima drukčije animacije, te je korisniku omogućeno postavljanje maksimalnog i minimalnog intenziteta osvjetljenja, broja ledica koje se želi osvijetliti putem pritiska na tipku *settings* kao i promjena boje osvjetljenja okretanjem potenciometra. Uz prikaz trenutnih vrijednosti raznih parametara automobila kao ambijentalnog osvjetljenja, uređaj sadrži zaslon za prikaz tih vrijednosti u numeričkom obliku.

Ključne riječi: OBD-II, Arduino, WS2812B, ambijentalno osvjetljenje za auto.

ABSTRACT

Title: Ambient car lighting with OBD protocol

The task of this work was to create a device that would control the ambient LED lighting in the car via the OBD-II protocol. The device communicates with the ELM327 OBD-II adapter via bluetooth and consists of four lighting modes, which can be selected by pressing mode button. Three operating modes depend on the car's current parameters (speed, engine revolutions per minute and coolant temperature), and the fourth mode is used to turn off the lighting. Each operating mode has different animations and the user can set the maximum and minimum intensity of the lighting, the number of LEDs to be lit by pressing the settings button, as well as changing the color of the lighting by turning the potentiometer. In addition to displaying the current values of various car parameters as some form of lighting animation, the device contains a screen for displaying these values in numerical form.

Key words: OBD-II, Arduino, WS2812B, car ambient lighting.

ŽIVOTOPIS

Autor ovog diplomskog rada, Marko Tandara rođen je 01.04.1996. godine u Slavonskom Brodu u Republici Hrvatskoj. Pohađao je osnovnu školu Josipa Antuna Čolnića u Đakovu od 2003. do 2011. godine. 2011. godine upisuje srednju školu, matematičku gimnaziju Antuna Gustava Matoša u Đakovu koju završava 2015. godine. Nakon uspješnog polaganja državne mature upisuje preddiplomski studij na fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, smjer Računarstvo. Nakon završenog preddiplomskog studija 2020. godine, upisuje diplomski studij, te se opredjeljuje na smjer Informacijske i podatkovne znanosti (DRD).

PRILOZI

Programski kôd za Arduino Nano mikroupravljač

EasyEDA shema i dizajn za PCB

Popis standardnih PID kodova za mode 01

Rad u .docx i .pdf formatu