

Programska podrška za interaktivnu RGB LED kocku

Kosović, Željko

Undergraduate thesis / Završni rad

2024

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

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:037837>

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

Download date / Datum preuzimanja: **2025-01-14**

Repository / Repozitorij:

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



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

Preddiplomski stručni studij elektrotehnike

**Programska podrška za interaktivnu RGB LED
kocku**

Završni rad

Željko Kosović

Osijek, 2024.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**Obrazac Z1S: Obrazac za ocjenu završnog rada na stručnom prijediplomskom studiju****Ocjena završnog rada na stručnom prijediplomskom studiju**

Ime i prezime pristupnika:	Željko Kosović
Studij, smjer:	Stručni prijediplomski studij Elektrotehnika, smjer Automatika
Mat. br. pristupnika, god.	A4706, 27.07.2021.
JMBAG:	0165089415
Mentor:	prof. dr. sc. Drago Žagar
Sumentor:	Josip Spišić, univ. mag. ing. comp.
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	prof. dr. sc. Krešimir Grgić
Član Povjerenstva 1:	prof. dr. sc. Drago Žagar
Član Povjerenstva 2:	izv. prof. dr. sc. Višnja Križanović
Naslov završnog rada:	%naziv_rada%
Znanstvena grana završnog rada:	Automatizacija i robotika (zn. polje elektrotehnika)
Zadatak završnog rada:	Sumentor: Josip Spišić, mag.ing.comp. Tema ovog završnog rada je razvoj programske podrške u Arduino IDE sučelju za upravljanje i animiranje 16x16x16 RGB LED kocke s WiFi komunikacijom, a samo upravljanje je potrebno implementirati putem mobilne aplikacije. Tema uključuje i kreiranje korisničkog sučelja, programiranje animacija i efekata, te integraciju s hardverom. Cilj rješenja je postići dinamično i atraktivno vizualno iskustvo za korisnika.
Datum ocjene pismenog dijela završnog rada od strane mentora:	11.09.2024.
Ocjena pismenog dijela završnog rada od strane mentora:	Izvrstan (5)
Datum obrane završnog rada:	13.9.2024.
Ocjena usmenog dijela završnog rada (obrane):	Izvrstan (5)
Ukupna ocjena završnog rada:	Izvrstan (5)
Datum potvrde mentora o predaji konačne verzije završnog rada čime je pristupnik završio stručni prijediplomski studij:	13.09.2024.



FERIT

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

IZJAVA O IZVORNOSTI RADA

Osijek, 13.09.2024.

Ime i prezime Pristupnika:	Željko Kosović
Studij:	Stručni prijediplomski studij Elektrotehnika, smjer Automatika
Mat. br. Pristupnika, godina upisa:	A4706, 27.07.2021.
Turnitin podudaranje [%]:	9

Ovom izjavom izjavljujem da je rad pod nazivom: **Programska podrška za interaktivnu RGB LED kocku**

izrađen pod vodstvom mentora prof. dr. sc. Drago Žagar

i sumentora Josip Spišić, univ. mag. ing. comp.

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

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

Potpis pristupnika:

SADRŽAJ:

1. UVOD	1
2. ZADATAK ZAVRŠNOG RADA	2
3. TEORIJSKI OKVIR	2
3.1. 3D KOCKE: OSNOVNI KONCEPTI I VRSTE	2
3.1.1. OPĆENITI OPIS 3D KOCKE	2
3.1.2 VRSTE I VELIČINE 3D KOCKI	3
3.1.3. 16X16X16 RGB LED KOCKA	4
3.2. LED TEHNOLOGIJA	5
3.2.1.PRINCIP RADA LED DIODA	5
3.2.2. RGB LED DIODE I NJIHOVE KARAKTERISTIKE	6
3.2.3. WS2812 LED DIODE.....	7
3.3. ESP8266 MODUL	7
3.4. TEENSY 4.0 MIKROKONTROLER	8
3.5. VISUAL STUDIO CODE (VSC)	9
3.6. PLATFORMIO	9
3.7. IoT (INTERNET OF THINGS)	10
3.8. BLYNK	11
3.8.1. BLYNKEDGENT BIBLIOTEKA	12
4. PROJEKTIRANJE I IMPLEMENTACIJA PROGRAMSKE PODRŠKE ZA INTERAKTIVNU RGB LED KOCKU	14
4.1. OPIS KORIŠTENE HARDVERSKJE PLATFORME	14
4.2 POVEZIVANJE ESP8226 I TEENSY 4.0 PUTEM BLYNK APLIKACIJE	16
4.2.1. INSTALACIJA I POSTAVLJANJE BLYNK APLIKACIJE	16
4.2.2. DODJELJIVANJE VIRTUALNIH PINOVA.....	18
4.2.3. DIZAJN APLIKACIJE	19
4.3. POVEZIVANJE ESP8226 S BLYNK APLIKACIJOM	23
4.3.1. USPOSTAVLJANJE VEZE I KONFIGURACIJA.....	23
4.3.2. IMPLEMENTACIJA I KONTROLA ANIMACIJA POMOĆU BLYNK VIRTUALNIH PINOVA.....	25

4.4. PISANJE KODA ZA UPRAVLJANJE RGB LED KOCKOM.....	30
4.4.1. UPRAVLJANJE RGB LED KOCKOM	30
5. PREDNOSTI I NEDOSTACI RAZVIJENOG RJEŠENJA.....	32
5.1. PREDNOSTI	32
5.2. NEDOSTACI	33
6. ZAKLJUČAK.....	34
LITERATURA:.....	35
POPIS OZNAKA I KRATICA	36
SAŽETAK	37
ABSTRACT	37
ŽIVOTOPIS.....	38

1. UVOD

U današnjem svijetu tehnologije, LED kocke predstavljaju fascinantno područje istraživanja i inovacija u području vizualnih efekata i interaktivne rasvjete. Ove strukture, sastavljene od velikog broja LED dioda, omogućuju stvaranje kompleksnih trodimenzionalnih svjetlosnih prikaza koji mogu biti statični ili dinamični. LED kocke koriste se u raznim područjima, od umjetničkih instalacija do edukacijskih alata i interaktivnih igara.

Razvoj programske podrške za upravljanje i animiranje 16x16x16 RGB LED kocke putem Arduino IDE sučelja, uz korištenje WiFi komunikacije, integrira više aspekata razvoja softvera i hardvera kako bi se postiglo dinamično i atraktivno vizualno iskustvo za korisnika. Izazov ovog projekta je komunikacija između mikro kontrolera i povezivanje istih na internet kako bi se ostvarila precizna kontrola za željene efekte.

U ovom završnom radu, detaljno će biti opisano kako konfigurirati aplikaciju za upravljanje RGB LED kockom, uključujući korake pri kreiranju aplikacije i njenog korisničkog sučelja. Kroz razvojne faze, istražiti ćemo različite tehnike programiranja animacija i efekata, koristeći Arduino IDE i VSC kao glavna razvojna okruženja. Također, bit će objašnjeno kako uspostaviti WiFi komunikaciju između mobilne aplikacije i LED kocke, osiguravajući stabilnu i pouzdanu vezu koja omogućava upravljanje svjetlosnim efektima u stvarnom vremenu (engl. Real-Time).

Na kraju, bit će dan pregled konfiguriranja i dizajn aplikacije te kodovi namijenjeni za konekciju RGB kocke na lokalnoj mreži. Uspješna implementacija ovog projekta rezultirat će sustavom koji omogućava korisnicima kontrolu svjetlosnih prikaza putem intuitivnog mobilnog sučelja, pružajući jedinstveno i interaktivno vizualno iskustvo.

2. ZADATAK ZAVRŠNOG RADA

Tema ovog završnog rada je razvoj programske podrške za upravljanje i animiranje putem Arduino IDE razvojnog sučelja 16x16x16 RGB LED kocke s WiFi komunikacijom, pri čemu je upravljanje potrebno implementirati putem mobilne aplikacije. Rad obuhvaća kreiranje korisničkog sučelja, programiranje animacija i efekata te integraciju s hardverom. Cilj je postići dinamično i atraktivno vizualno iskustvo za korisnika, koristeći suvremena tehnološka dostignuća.

Osim tehničkih detalja, ovaj završni rad će se osvrnuti na teorijske osnove potrebne za razumijevanje korištenih tehnologija, prednosti i nedostatke tih tehnologija, te potencijalne izazove i rješenja u procesu razvoja. Integracijom softverskih i hardverskih komponenti, rad nastoji stvoriti sustav koji ne samo da pruža vizualni užitak, već i demonstrira napredne mogućnosti modernih IoT uređaja.

3. TEORIJSKI OKVIR

3.1. 3D KOCKE: OSNOVNI KONCEPTI I VRSTE

3.1.1. OPĆENITI OPIS 3D KOCKE

3D LED kocke su strukture koje koriste svjetlosne diode (LED) za stvaranje trodimenzionalnih svjetlosnih prikaza. Ove kocke predstavljaju uzbudljivo područje istraživanja i inovacija u vizualnim efektima i interaktivnoj rasvjeti. Osnovni koncept ovih kocki uključuje organizaciju LED dioda u matricu koja se proteže u tri dimenzije: dužinu, širinu i visinu. Svaka LED dioda unutar kocke može biti pojedinačno upravljana, omogućujući stvaranje složenih svjetlosnih uzoraka i animacija.

3D LED kocke nalaze primjenu u različitim područjima, uključujući umjetničke instalacije, edukacijske alate, vizualizacije podataka i interaktivne igre. Njihova sposobnost prikazivanja dinamičnih i atraktivnih svjetlosnih efekata čini ih posebno zanimljivima za kreativne projekte i istraživačke svrhe.

3.1.2 VRSTE I VELIČINE 3D KOCKI

Postoji nekoliko vrsta 3D LED kocki, koje se razlikuju po veličini, konfiguraciji i tehnologiji korištenoj za njihovo upravljanje:

1. Standardne 3D LED kocke:

- **Mali formati (npr. 4x4x4):** Ove kocke sadrže relativno mali broj LED dioda (64 ukupno) i često se koriste u edukacijske svrhe ili za manje projekte.
- **Srednji formati (npr. 8x8x8):** S većim brojem LED dioda (512 ukupno), ove kocke omogućuju složenije animacije i koriste se u raznim aplikacijama.
- **Veliki formati (npr. 16x16x16):** Ove kocke sadrže veliki broj LED dioda (4096 ukupno) i omogućuju stvaranje vrlo detaljnih i kompleksnih svjetlosnih prikaza.

2. RGB 3D LED kocke:

- RGB LED kocke koriste diode koje mogu emitirati crvenu, zelenu i plavu svjetlost. Kombiniranjem ovih boja moguće je stvoriti širok spektar boja, omogućujući stvaranje živopisnih i raznobojnih svjetlosnih efekata. Ove kocke zahtijevaju sofisticiraniju kontrolu, ali nude mnogo veće mogućnosti u kreiranju vizualnih prikaza.

3. Adresabilne LED kocke:

- Ove kocke koriste adresabilne LED diode, gdje svaka dioda ima jedinstvenu adresu i može biti kontrolirana nezavisno od drugih. To omogućuje preciznu kontrolu svjetlosnih efekata i animacija. Adresabilne LED diode često koriste protokole poput WS2812, APA102 ili PL9823. U ovom slučaju koriste se led diode s WS2812 protokolom.

3.1.3. 16X16X16 RGB LED KOCKA

Ovaj završni rad se zasniva na 16x16x16 RGB LED kocki koja spaja umjetnost i znanost kako bi se stvorili spektakularni vizualni efekti. Sastavljena od 4096 individualnih WS2812 LED dioda, ova kocka omogućava velik broj mogućnosti animacija i svjetlosnih prikaza. Svaka od ovih LED dioda može emitirati širok spektar boja zahvaljujući RGB (crvena, zelena, plava) tehnologiji, a WS2812 diode su posebno poznate po svojoj adresabilnosti i jednostavnosti upravljanja.

WS2812 LED diode integriraju kontrolni čip unutar svake diode, omogućujući preciznu kontrolu nad bojom i svjetlinom svake pojedine diode putem jednostavnog serijskog protokola. Ovo čini WS2812 diode idealnim za projekte poput ove 3D LED kocke, jer omogućavaju lako programiranje i stvaranje kompleksnih svjetlosnih efekata s minimalnim hardverskim zahtjevima.

U ovom završnom radu, korištena je kombinacija mikro kontrolera ESP8266 i Teensy 4.0, gdje ESP8266 služi za komunikaciju s internetom dok Teensy upravlja RGB LED kockom. Pored ESP8266 i Teensy-a, važan dio su i razvojna sučelja, tj. Arduino IDE i VSC, koji pružaju fleksibilno i korisnički prijateljsko razvojno okruženje. Kombinacija Arduino IDE i VSC-a omogućava jednostavno programiranje i brzu implementaciju željenih funkcionalnosti, dok brojne dostupne biblioteke olakšavaju rad s WS2812 LED diodama.

U sljedećim poglavljima, detaljno će biti istražene ključne tehnologije koje omogućavaju funkcioniranje ove RGB LED kocke. Raspravit će se o principima rada LED dioda, karakteristikama WS2812 dioda, arhitekturi ESP8266 i Teensy 4.0 mikro kontrolera, te mogućnostima koje pružaju Arduino i VSC platforme. Biti će objašnjeno kako moderni tehnološki alati i metode omogućavaju stvaranje raznih svjetlosnih efekata, te će biti razmotreni izazovi i rješenja vezana uz integraciju hardvera i softvera.

3.2. LED TEHNOLOGIJA

LED tehnologija prati nas u svakodnevnom životu - kao indikatorska svjetla u raznim vrstama elektroničkih uređaja, kao i pozadinsko osvjetljenje LCD zaslona i monitora. Također je standard u većini modernih svjetlosnih proizvoda. Vrlo je vjerojatno da će u bliskoj budućnosti, zbog brzo napredujuće tehnologije, LED-ovi potpuno zamijeniti tradicionalna svjetlosna rješenja. Stoga vrijedi naučiti o svojstvima ovog univerzalnog izvora svjetlosti [1].

3.2.1. PRINCIP RADA LED DIODA

Svjetleće diode (LED) su poluvodički uređaji koji emitiraju svjetlo kada kroz njih prolazi električna struja. Boja emitiranog svjetla ovisi o poluvodičkom materijalu i energetske širini pojasa materijala. LED diode se koriste u širokom rasponu primjena zbog svoje učinkovitosti, dugovječnosti i svestranosti [2].

Materijal od kojeg je LED dioda izrađena određuje boju svjetlosti koju emitira. Najčešće korišteni materijali uključuju galij-arsenid (GaAs), galij-fosfid (GaP) i indij-galij-nitrid (InGaN). Svaka kombinacija materijala ima specifičnu energetske razinu koja određuje valnu duljinu svjetlosti, a time i boju.

Napredni LED sustavi uključuju adresabilne LED diode, omogućujući pojedinačnu kontrolu svake diode u nizu. Ova značajka omogućuje složene svjetlosne uzorke i efekte, što je ključno za primjene koje zahtijevaju visoku preciznost i dinamičku kontrolu [2].

3.2.2. RGB LED DIODE I NJIHOVE KARAKTERISTIKE

RGB LED je posebna vrsta LED dioda koja se sastoji od nekoliko jednostavnih LED nizova poput onih koji se nalaze u drugim jednobojnim LED diodama. Na taj način mogu emitirati u ove tri osnovne boje, generirajući tako sve vrste različitih efekata i boja (čak i bijela kombinirajući istovremeno crvenu, zelenu i plavu boju) samo upravljajući jednom od pinova ovih komponenata. Te 3 ukomponirane LED diode u istoj je kapsulaciji sposoban proizvesti čitav taj raspon boja. Ima malo drugačiji pinout od uobičajenih LED dioda, jer uključuju 3 pina, po jedan za svaku boju (katode ili +) i još jedan zajednički svima, anodu (-) [3].

LED diode imaju niz prednosti u odnosu na tradicionalne izvore svjetlosti, kao što su žarulje sa žarnom niti i fluorescentne lampe. One su energetski učinkovitije, imaju duži vijek trajanja, obično između 50.000 i 100.000 sati, što ih čini dugoročnim rješenjem za razne primjene, manju potrošnju energije i veći faktor pouzdanosti. Osim toga, LED diode ne sadrže štetne kemikalije, poput žive, što ih čini ekološki prihvatljivijima.

Zbog svoje sposobnosti prikazivanja širokog spektra boja, RGB LED diode koriste se u raznim aplikacijama, uključujući dekorativnu rasvjetu, zaslone, signalizaciju i indikatore, te u umjetničkim instalacijama i interaktivnim sustavima poput 3D LED kocki.

RGB LED diode često se kontroliraju pomoću mikro kontrolera koji omogućuju precizno upravljanje intenzitetom svake boje. U modernim aplikacijama, ove diode mogu biti dio adresabilnih LED traka ili matrica, gdje se svaka dioda može nezavisno kontrolirati kako bi se stvorili kompleksni svjetlosni uzorci i animacije.

3.2.3. WS2812 LED DIODE

Jedna od ključnih karakteristika WS2812 LED dioda je njihova sposobnost adresabilnosti. Svaka dioda može se upravljati pojedinačno putem serijskog protokola, omogućavajući neovisno upravljanje bojom i svjetlinom svake diode. Osim toga, svaka dioda ima tri boje - crvenu, zelenu i plavu, što omogućuje stvaranje širokog spektra boja i raznovrsne svjetlosne efekte. WS2812 LED diode imaju jednostavnu strukturu s samo četiri pina za povezivanje: napajanje, uzemljenje i dva za podatke (ulaz i izlaz). Ove LED diode komuniciraju putem serijskog protokola, koji se obično temelji na jednostavnom protokolu poput nekih varijanti SPI (engl. Serial Peripheral Interface) protokola.

3.3. ESP8266 MODUL

ESP8266 je mikro kontroler s integriranim WiFi mogućnostima, što ga čini idealnim za ovaj završni rad. U kontekstu ovog završnog rada, ESP8266 modul omogućava bežičnu komunikaciju s 16x16x16 RGB LED kockom putem mobilne aplikacije.

Osnovne značajke ESP8266 modula uključuju WiFi povezivanje, jednoprocesorski dizajn, nisku potrošnju energije, te relativno ograničen, ali dovoljan broj I/O pinova. Integrirani WiFi omogućava stabilnu bežičnu komunikaciju s mrežom, čime se osigurava pouzdan prijenos podataka između kocke i mobilne aplikacije. ESP8266 koristi Tensilica L106 32-bitni mikro procesor, koji omogućava učinkovitu obradu podataka uz nisku potrošnju energije. Ostale značajke, navedene u [4], ukazuju na to koliko je ESP8266 napredan mikro kontroler s obzirom na svoju cijenu i veličinu.

Korištenje ESP8266 s Arduino IDE-om uključuje nekoliko koraka. Prvi korak je instalacija ESP8266 pločice u Arduino IDE-u, što se postiže dodavanjem URL-a za ESP8266 pločice u postavkama i instalacijom pločice putem "Board Managera". Nakon toga slijedi pisanje koda koji može obuhvatiti sve, od jednostavnih LED Blynk projekata do složenih mrežnih aplikacija. Kod za upravljanje 16x16x16 RGB LED kockom uključuje postavljanje WiFi veze, kontrolu LED dioda i implementaciju različitih svjetlosnih efekata. Nakon pisanja koda, isti se prevodi i učitava na ESP8266 modul putem USB-a, a serijski monitor se koristi za praćenje izlaza i otklanjanja pogrešaka.

3.4. TEENSY 4.0 MIKROKONTROLER

Teensy 4.0 je moćan mikro kontroler temeljen na ARM Cortex-M7 procesoru koji radi na 600 MHz, čineći ga jednim od najbržih dostupnih mikro kontrolera. Unatoč svojoj kompaktnoj veličini, nudi obilje značajki, uključujući 1024 KB RAM-a, 2 MB flash memorije, USB host podršku, te brojne digitalne i analogne ulaze i izlaze. Više o specifikacijama ovog mikro kontrolera priloženo je pod [5].

Teensy 4.0 je vrlo prilagodljiv i podržava razne komunikacijske protokole poput I2C, SPI, UART i CAN bus, što ga čini idealnim za različite primjene pa tako i za ovaj projekt. Kompatibilan je s VSC-om, omogućavajući jednostavno programiranje i integraciju s Arduino bibliotekama.

Jedna od ključnih funkcionalnosti Teensy 4.0 je sposobnost precizne kontrole signala. Ovaj mikro kontroler može brzo i učinkovito obrađivati signale, omogućavajući implementaciju složenih animacija i svjetlosnih efekata u stvarnom vremenu. U kombinaciji s ESP8266 mikro kontrolerom čini sjajan spoj za kontrolu i upravljanje LED diodama.

3.5. VISUAL STUDIO CODE (VSC)

Visual Studio Code (VSC) je iznimno popularno razvojno okruženje koje pruža širok spektar moćnih alata za programiranje i razvoj softverskih rješenja. Poznat po svojoj fleksibilnosti i opsežnoj podršci za različite programske jezike i tehnologije, VSC se ističe kao izvrstan alat za sve vrste razvojnih zadataka. Njegove ključne karakteristike uključuju intuitivan uređivač koda koji podržava isticanje sintakse, automatsko dovršavanje koda i integriranu dokumentaciju. Ove značajke značajno olakšavaju pisanje i pregledavanje koda, što je posebno korisno pri razvoju složenih aplikacija kao što je upravljanje 16x16x16 RGB LED kockom. VSC također nudi bogat sistem ekstenzija koje omogućuju integraciju s različitim alatima i platformama. Ekstenzije za rad s Arduino IDE-om, Gitom, ili specifičnim bibliotekama za upravljanje LED diodama znatno poboljšavaju produktivnost i učinkovitost u razvoju. Osim toga, VSC pruža napredne mogućnosti za otklanjanje pogrešaka, uključujući podršku za postavljanje prijelomnih točaka, praćenje varijabli i analizu izlaznih podataka. Ove funkcionalnosti omogućavaju brže otkrivanje i ispravljanje grešaka u kodu, što je ključno za osiguranje stabilnog rada aplikacija za upravljanje LED kockom.

3.6. PLATFORMIO

PlatformIO je sofisticirani alat namijenjen razvoju aplikacija za ugrađene sustave, koji pruža podršku za različite platforme, arhitekture i razvojne okvire. Ovaj višenamjenski alat omogućuje rad s širokim spektrom hardverskih platformi unutar jedinstvene razvojne okoline, što povećava učinkovitost i produktivnost. PlatformIO se posebno ističe u kombinaciji s Visual Studio Code-om (VSC), gdje nudi cjelovito rješenje za razvoj softverskih rješenja za ugrađene sustave [6].

Jedna od ključnih funkcionalnosti PlatformIO-a je podrška za različite platforme i mikro kontrolere, kao što su ESP8266 i Teensy 4.0, što je posebno važno za složene projekte poput izrade 16x16x16 RGB LED kocke. Korištenje PlatformIO-a omogućava programerima da se fokusiraju na razvoj i optimizaciju softvera umjesto na prilagodbu razvojnog okruženja. PlatformIO također nudi napredne alate za upravljanje zavisnostima i bibliotekama, što pojednostavljuje dodavanje i ažuriranje potrebnih komponenti, poput WS2812 LED dioda ili

WiFi modula. Automatizirano upravljanje procesom gradnje i kompilacije projekta smanjuje potrebu za ručnim postavkama i smanjuje mogućnost pogrešaka.

Kao ekstenzija za VSC, PlatformIO omogućava korištenje prednosti oba alata u jednom integriranom okruženju, uključujući napredne alate za pisanje i debugiranje koda, integraciju s verzijskim sustavima te alate za testiranje i implementaciju. Ova integracija omogućava efikasno upravljanje cijelim razvojnim procesom, od pisanja koda do konačne implementacije i testiranja. PlatformIO ne samo da ubrzava razvojne procese, već omogućava programerima da se usmjere na inovacije i optimizaciju, rezultirajući kvalitetnijim i robusnijim softverskim rješenjima za ugrađene sustave.

3.7. IoT (INTERNET OF THINGS)

IoT je iniciran od RFID (engl. Radio-frequency identification) zajednice čiji članovi su tvrdili da se može pronaći podatak o označenom objektu pregledavajući Internet adresu ili bazu podataka koji odgovara traženom RFID-u ili sličnoj stvari koja koristi tehnologiju bliskog polja [7].

RFID je bežična upotreba elektromagnetskih polja za prijenos podataka u svrhu automatskog prepoznavanja i praćenja označenih predmeta. Primjenjiv je u širokom spektru aplikacija poput nadzora pacijenta, procesu distribucije te vojnih aplikacija. RFID zajednica je došla do zaključka da je ključ IoT-a RFID, zajedno sa senzorskom tehnologijom, nano tehnologijom i tehnologiji ugrađenoj u inteligenciju [8].

Kasnije se pojavila komunikacija niskog polja (engl. Near Field Communication - NFC) koja se koristila na mobilnim uređajima za očitavanje NFC oznaka. Uz pomoć NFC-a pojednostavilo se obavljanje transakcija te omogućilo jednostavnu razmjenu digitalnog sadržaja. Nakon uspostave komunikacijskih mreža i globalnog roaminga to je predstavilo novi val tehnologija [7].

Jedna od ključnih komponenti IoT sustava je komunikacijska infrastruktura, koja omogućava povezivanje uređaja putem različitih mrežnih protokola kao što su WiFi, Bluetooth, Zigbee i 5G. Ova infrastruktura omogućava prikupljanje i prijenos podataka između uređaja i centraliziranih sustava za obradu podataka. Napredni algoritmi za analizu podataka i strojno

učenje koriste se za donošenje odluka i automatizaciju procesa na temelju prikupljenih podataka.

U kontekstu projekta 16x16x16 RGB LED kocke, IoT tehnologija omogućava bežičnu komunikaciju između LED kocke i mobilne aplikacije, pružajući korisnicima mogućnost upravljanja svjetlosnim efektima u stvarnom vremenu. Korištenjem mikro kontrolera ESP8266 s integriranim WiFi modulom, LED kocka postaje dio IoT ekosustava, omogućavajući daljinsko upravljanje i prilagodbu svjetlosnih prikaza putem mobilne aplikacije.

3.8. BLYNK

Blynk je aplikacija dizajnirana za jednostavno stvaranje projekata povezanih s IoT-om omogućujući korisnicima kontrolu nad različitim uređajima putem mobilnih telefona. Korištenjem Blynka kao osnove za razvoj vlastite aplikacije za komunikaciju između mobitela i 16x16x16 RGB LED kocke, otvara se niz mogućnosti za interaktivno i intuitivno upravljanje svjetlosnim efektima.

Jedna od glavnih prednosti Blynk aplikacije je njeno jednostavno i korisnički prilagođeno sučelje koje omogućava brzo postavljanje i konfiguraciju projekata. Korisnici mogu kreirati virtualne ploče koje sadrže različite alate (engl. widgete) poput tipki, klizača, grafova i kontrola boja. Ovi alati se povezuju s mikro kontrolerima putem internetske veze, omogućavajući stvaranje složenih kontrolnih sustava bez potrebe za detaljnim programiranjem korisničkog sučelja.

Blynk podržava širok spektar mikro kontrolera i razvojnih ploča, uključujući ESP8266, koji je ključan za ovaj projekt. Integracija s ESP8266 omogućava bežičnu komunikaciju između mobitela i LED kocke, pružajući korisnicima mogućnost prilagodbe svjetlosnih efekata u stvarnom vremenu. Također, Blynk nudi funkcionalnosti poput praćenja stanja uređaja i automatskih obavijesti, što dodatno poboljšava interaktivnost i korisničko iskustvo.

Upotreba Blynk aplikacije za izradu vlastite aplikacije nudi nekoliko prednosti. Prvo, smanjuje vrijeme razvoja omogućavajući fokus na ključne funkcionalnosti projekta. Drugo, pruža visoku razinu fleksibilnosti i proširivosti, omogućujući dodavanje novih funkcija i prilagodbu postojećih prema specifičnim potrebama korisnika. Na kraju, Blynk zajednica i dostupna

dokumentacija pružaju podršku i resurse koji mogu pomoći u rješavanju problema i unapređenju projekta.

3.8.1. BLYNKEDGENT BIBLIOTEKA

BlynkEdgent.h je biblioteka koja je dio Blynk platforme i specijalizirana je za napredne IoT projekte. Namijenjena je olakšavanju integracije i upravljanja uređajem koji koristi Blynk, uključujući funkcionalnosti kao što su jednostavno povezivanje s Blynk serverom, OTA (engl. Over-The-Air) ažuriranja i ugrađeno upravljanje autentifikacijom. Ova biblioteka omogućava programerima brži razvoj i implementaciju IoT uređaja koji koriste Blynk kao platformu za daljinsko upravljanje i nadzor.

Biblioteka automatski upravlja povezivanjem uređaja s Blynk serverom, uključujući sve korake potrebne za autentifikaciju i ponovno uspostavljanje veze u slučaju prekida, što značajno olakšava razvoj jer programer ne mora brinuti o detaljima povezivanja. Jedna od najvažnijih funkcija ove biblioteke je podrška za OTA ažuriranja, koja omogućava ažuriranje firmvera vašeg uređaja putem interneta bez potrebe za fizičkim pristupom uređaju. BlynkEdgent upravlja ovim procesom, osiguravajući da se nova verzija firmvera sigurno i bez grešaka preuzme i instalira na uređaj.

Biblioteka također olakšava upravljanje autentifikacijom uređaja na Blynk platformi, uključujući rukovanje autentifikacijskim tokenima i osiguranje sigurnog povezivanja s Blynk serverom. Omogućuje jednostavno povezivanje virtualnih pinova, koji se koriste za interakciju s Blynk aplikacijom, s funkcijama u kodu, što pojednostavljuje rad s Blynk pinovima i omogućuje jednostavnu interakciju između uređaja i korisničkog sučelja aplikacije. Uz to, uključuje opcije za praćenje i debugging, poput ispisivanja podataka u serijskom monitoru, što pomaže programerima da prate status uređaja, otkriju potencijalne probleme i isprave greške tijekom razvoja [9].

Uređaji koji koriste BlynkEdgent.h mogu se jednostavno povezati s Blynk serverom uz minimalnu konfiguraciju. Uobičajeni postupak uključuje definiranje Blynk šablona i tokena, uključivanje biblioteke i pozivanje njenih funkcija u `setup()` i `loop()` funkcijama. U `setup()` funkciji se poziva `BlynkEdgent.begin()`, što inicijalizira sve potrebne komponente za

povezivanje s Blynk serverom i omogućava OTA ažuriranja, dok se u loop() funkciji kontinuirano poziva BlynkEdgent.run() kako bi se održavala veza s Blynk serverom i upravljalo svim procesima, uključujući ažuriranja i upravljanje pinovima.

Prednosti korištenja BlynkEdgent.h uključuju pojednostavljenje razvoja jer programeri ne moraju brinuti o detaljima vezanim uz povezivanje, autentifikaciju i OTA ažuriranja, što značajno ubrzava razvoj IoT aplikacija. Biblioteka je dizajnirana da bude robusna s automatskim rukovanjem prekidima veze i sigurnosnim značajkama, te podržava razne vrste uređaja i scenarije, omogućujući jednostavnu prilagodbu za specifične potrebe. Više informacija priloženo je na [10].

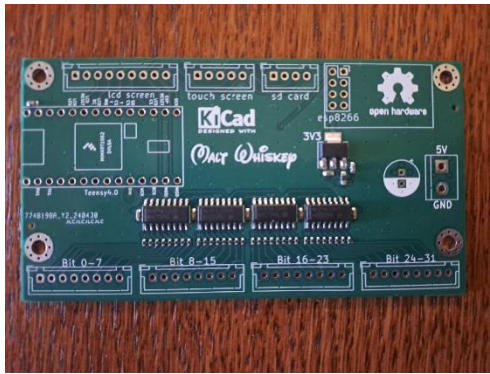
4. PROJEKTIRANJE I IMPLEMENTACIJA PROGRAMSKE PODRŠKE ZA INTERAKTIVNU RGB LED KOCKU

4.1. OPIS KORIŠTENE HARDVERSKE PLATFORME

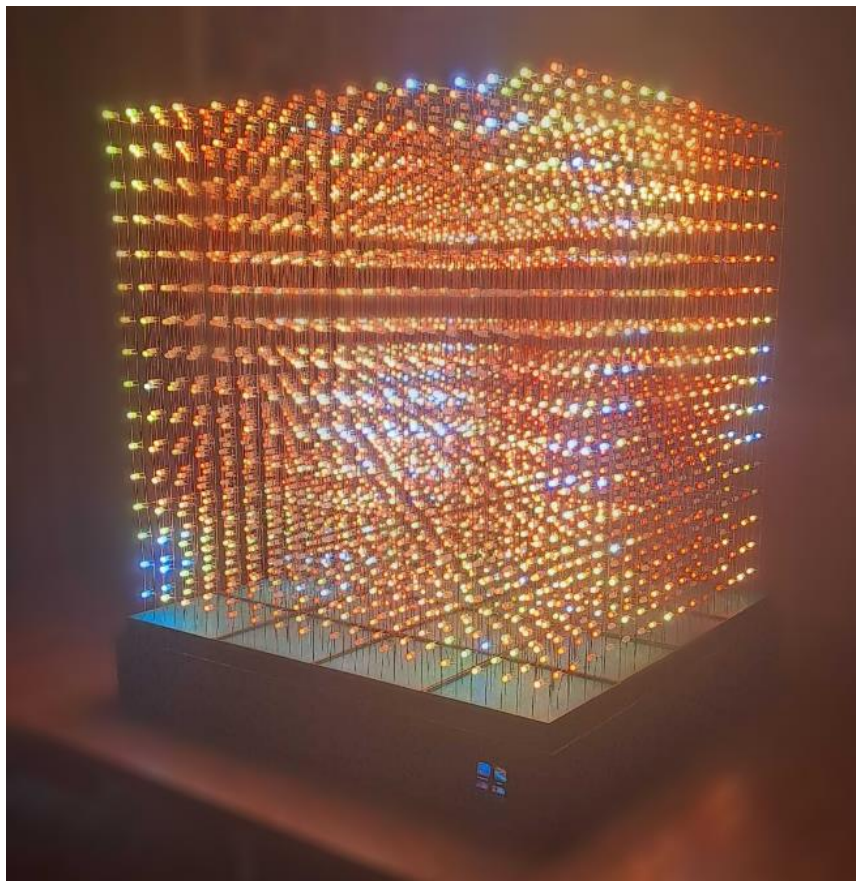
U svrhu razvoja rješenja završnog rada napravljena je tiskana pločica na koju su smještene sve komponente. Prednja strana tiskane pločice za upravljanje 16x16x16 RGB LED kockom vrlo je dobro organizirana s jasno označenim utorima i priključcima za glavne komponente. Na lijevoj strani ploče nalazi se utor za Teensy 4.0 mikro kontroler, koji je zadužen za upravljanje cjelokupnim radom ploče. Iznad ovog utora smješten je utor za LCD ekran s podrškom za dodir, čime se omogućuje vizualizacija ili interaktivno upravljanje funkcijama kocke. Pored njega, također iznad Teensy utora, nalazi se utor za SD karticu koja služi za spremanje ili učitavanje podataka koji se koriste za prikaz animacija ili drugih funkcija na kocki.

Desno od utora za Teensy, jasno je označen utor za ESP8266 modul, koji omogućava bežičnu komunikaciju za povezivanje s Blynk aplikacijom. Ovaj modul ključan je za daljinsko upravljanje kockom ili za primanje i slanje podataka putem Wi-Fi mreže. Na desnoj strani ploče, pored ESP8266 modula, nalazi se oznaka za 5V priključak za napajanje, kao i GND (uzemljenje), što omogućava stabilno napajanje svih komponenti.

Na donjoj polovici ploče nalaze se četiri odvojena niza pinova označena kao "Bit 0-7," "Bit 8-15," "Bit 16-23," i "Bit 24-31". Ovi pinovi su povezani s redovima i stupcima RGB LED dioda, omogućujući kontrolu nad svim LED diodama. Kroz ove pinove, moguće je upravljati svjetlima kocke na način da se kreiraju različite svjetlosne animacije. Ukupna struktura tiskane pločice omogućuje jednostavno povezivanje i kontrolu različitih modula i perifernih uređaja, pružajući fleksibilnost i mogućnost proširenja sustava. Tiskanu pločicu možemo vidjeti na slici ispod.



Slika 4.1. Tiskana pločica

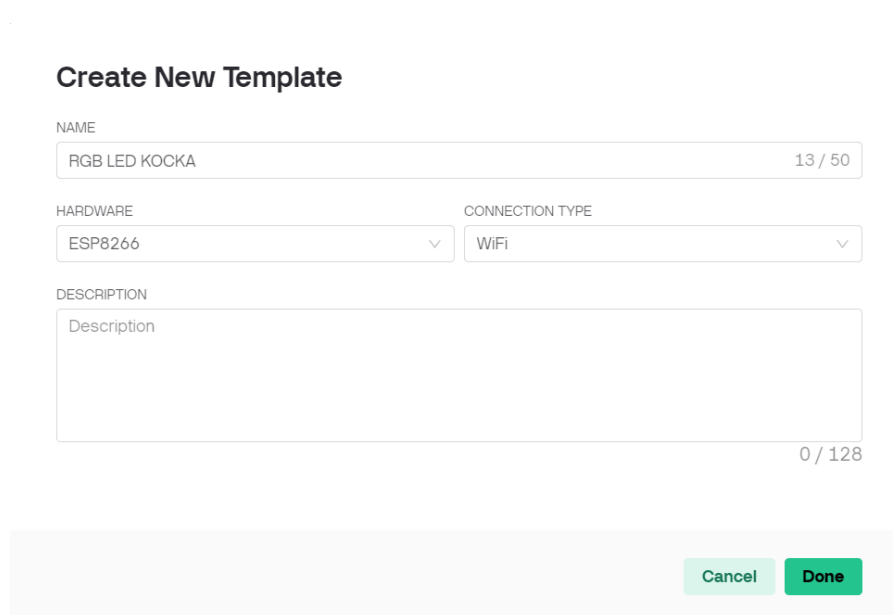


Slika 4.2. 16x16x16 RGB LED kocka

4.2 POVEZIVANJE ESP8226 I TEENSY 4.0 PUTEM BLYNK APLIKACIJE

4.2.1. INSTALACIJA I POSTAVLJANJE BLYNK APLIKACIJE

Prvi korak je preuzimanje Blynk aplikacije s trgovine aplikacija. Nakon instalacije, potrebno je otvoriti aplikaciju i registrirati se s e-mail adresom i lozinkom kako bi se kreirao korisnički račun. Nakon uspješne registracije, slijedi pristupanje Blynk platformi i kreiranje novog projekta. Prilikom kreiranja projekta, potrebno je odabrati vrstu mikro kontrolera što je u ovom slučaju ESP8226, što će aplikacija koristiti za slanje i primanje podataka s hardverom.



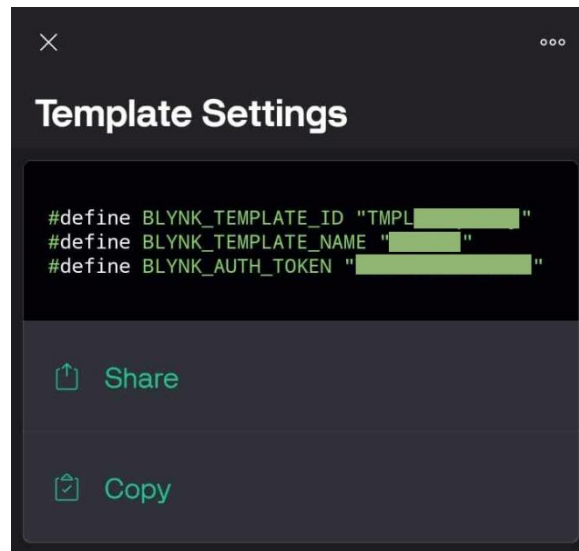
The screenshot shows the 'Create New Template' form in the Blynk application. The form has the following fields and options:

- NAME:** A text input field containing 'RGB LED KOCKA' with a character count of '13 / 50'.
- HARDWARE:** A dropdown menu with 'ESP8266' selected.
- CONNECTION TYPE:** A dropdown menu with 'WiFi' selected.
- DESCRIPTION:** A large text area containing the placeholder text 'Description' and a character count of '0 / 128'.

At the bottom right of the form, there are two buttons: 'Cancel' and 'Done'.

Slika 4.3. Kreiranje novog projekta u Blynk aplikaciji

Nakon definiranja imena projekta i odabira mikro kontrolera, aplikacija generira identifikaciju (engl. Template ID) i ime (engl. Template Name) predložka te token autentičnosti (engl. Auth Token) koji je poslan na e-mail. Ovaj token je ključ za povezivanje Blynk aplikacije s projektom. Token je potrebno unijeti u kod koji se nalazi na mikro kontroleru kako bi se omogućila komunikacija između aplikacije i hardvera. Također Blynk generira identifikaciju i ime projekta koje je potrebno navesti na samom vrhu koda.

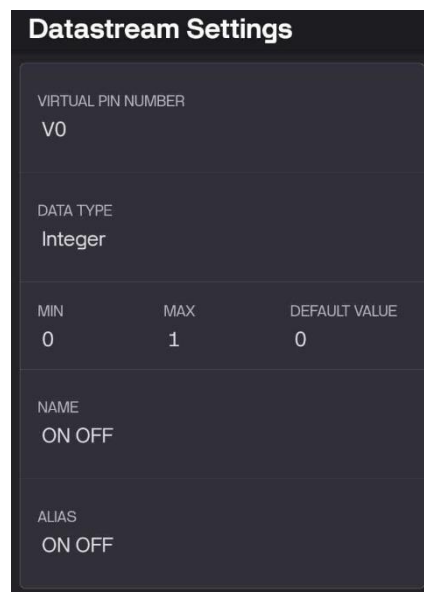


Slika 4.4. Postavke predloška

Unutar aplikacije, moguće je dodavati različite virtualne kontrole, kao što su tipke, klizači, LED indikatori, grafovi, itd., koje služe za upravljanje i nadzor rada LED kocke. Svaka kontrola može biti prilagođena prema potrebama projekta, uključujući ime, veličinu, boje i funkcionalnost. U ovom završnom radu su korišteni samo virtualni pinovi jer nije bilo potrebe za drugima.

4.2.2. DODJELJIVANJE VIRTUALNIH PINOVA

Proces započinje dodjeljivanjem virtualnog pina unutar Blynk aplikacije, gdje korisnik specificira ime pina, tip podataka, kao i broj virtualnog pina (u ovom slučaju V0) koji služi za uključivanje i isključivanje animacija. Ovaj proces omogućuje preciznu konfiguraciju svake kontrolne komponente, kao što su tipke, klizači ili senzori, koji će komunicirati s mikro kontrolerom. Osim osnovnih postavki, korisnik također može odrediti vrstu podataka koje će pin primiti ili slati (npr. cijeli brojevi, decimalni brojevi, tekst), kao i specificirati minimalne i maksimalne vrijednosti koje pin može primiti. Ove postavke omogućuju da aplikacija interpretira podatke na način koji najbolje odgovara specifičnim potrebama, čime se osigurava precizna kontrola i pouzdana komunikacija između Blynk aplikacije i mikro kontrolera.



The screenshot shows the 'Datastream Settings' interface for a virtual pin. The settings are as follows:

MIN	MAX	DEFAULT VALUE
0	1	0

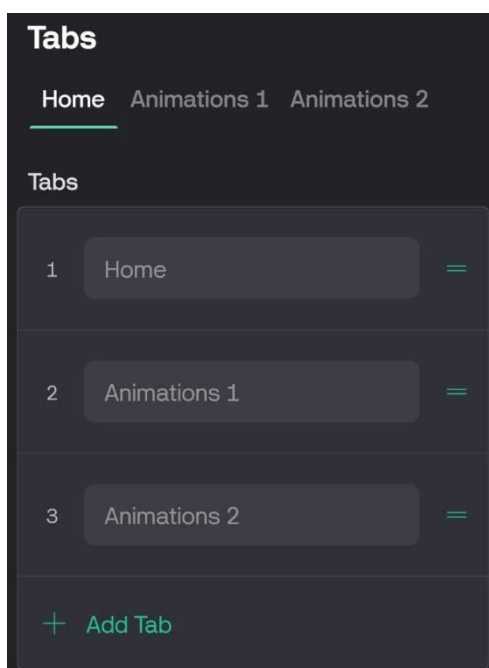
Other settings shown include:

- VIRTUAL PIN NUMBER: V0
- DATA TYPE: Integer
- NAME: ON OFF
- ALIAS: ON OFF

Slika 4.5. Kreiranje virtualnog pina V0

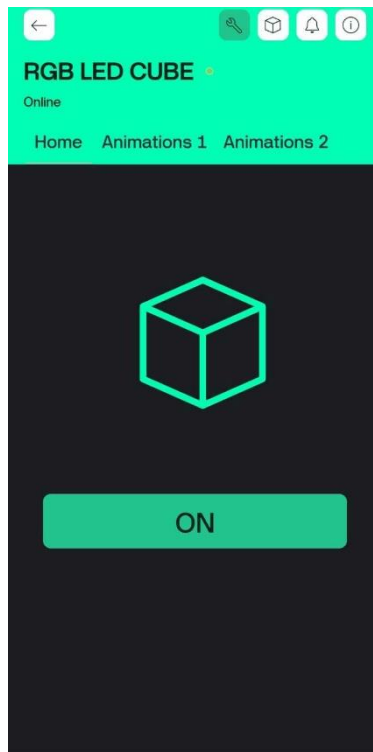
4.2.3. DIZAJN APLIKACIJE

Nakon što su postavljeni osnovni parametri za virtualne pinove, slijedi konfiguracija kontrola unutar Blynk aplikacije. Najprije se kreiraju “kartice” (engl. Tabs) “HOME”, “ANIMATIONS 1” i “ANIMATIONS 2” koje omogućuju organizaciju i upravljanje različitim sučeljima unutar ovog završnog rada.



Slika 4.6. Stvaranje kartica

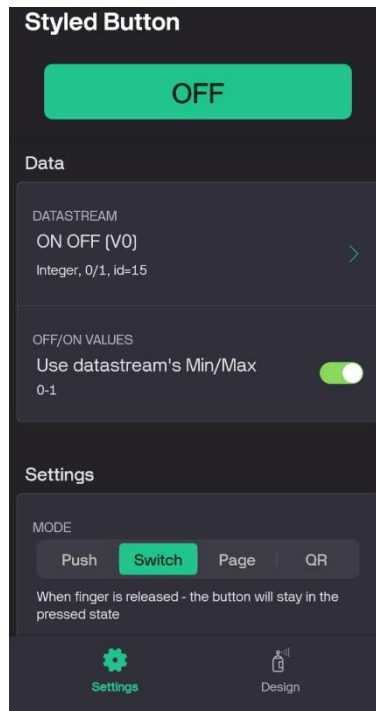
U kartici “HOME” odabire se odgovarajući alat iz "kutije s alatima" (engl. Widget Box), gdje se nalaze svi dostupni alati za upravljanje projektom poput tipki, prekidača, klizača, grafičkih prikaza, itd. Za potrebe uključivanja i isključivanja animacija na LED kocki, odabire se "prekidač" (engl. Switch), koji se koristi za jednostavnu binarnu kontrolu—paljenje i gašenje. Na početnu karticu je također dodana ikona kocke radi ljepšeg izgleda.



Slika 4.7. Izgled početne kartice

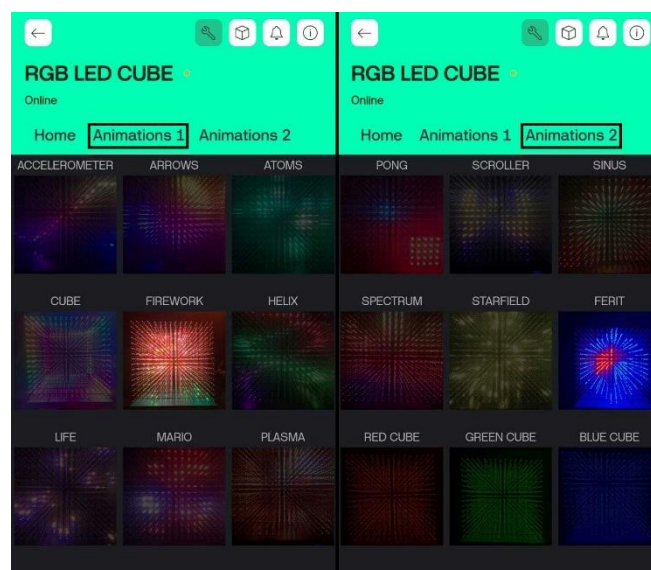
Nakon odabira "prekidača", dodjeljuje mu se naziv kako bi bio lako prepoznatljiv, te se povezuje s odgovarajućim virtualnim pinom koji je prethodno konfiguriran (V0). Također, definira se vrijednosti za "uključeno" (engl. on) i "isključeno" (engl. off) stanja, koje određuju kako će se mikro kontroler ponašati kada korisnik uključi ili isključi prekidač (slika 4.7.). Ove vrijednosti se mogu prilagoditi specifičnim potrebama projekta, osiguravajući da svaka akcija u aplikaciji odgovara točno određenom odgovoru u sustavu, kao što je pokretanje ili zaustavljanje animacije na LED kocki.

Tijekom izrade projekta, postupak dodjele virtualnih pinova ponavlja se više puta, jer svaka nova animacija na LED kocki zahtijeva zaseban virtualni pin u Blynk aplikaciji.



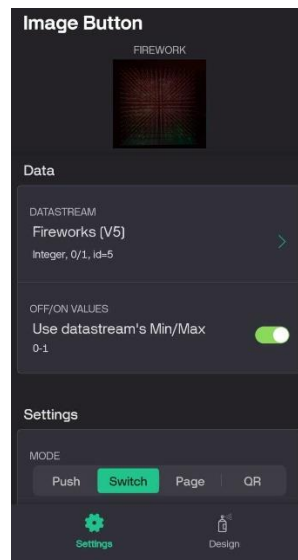
Slika 4.7. Konfiguracija prekidača

Zatim se u kartice “ANIMATIONS 1” i “ANIMATIONS 2” dodaju “slikovni gumbi” (engl. image buttons) koji služe za odabir animacija. Animacije je moguće odabrati samo kada je uključen virtualni pin V0 (“ON OFF” prekidač). U niti jednom trenutku ne mogu biti uključene dvije različite animacije.

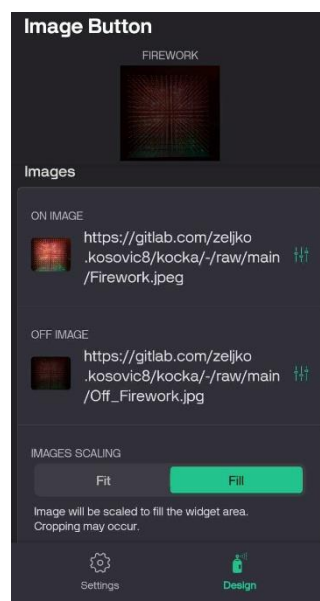


Slika 4.8. Izgled kartica ANIMATIONS 1 i ANIMATIONS 2

Nakon dodavanja “slikovnog gumba” treba odraditi njegove postavke. Prvo se povezuje gumb s virtualnim pinom (u ovom slučaju V5) koji će odabrati animaciju “FIREWORK”. Zatim se dodaje slika za isključeno i uključeno stanje pina. Dodatno, moguće je prilagoditi veličinu gumba prema potrebama dizajna projekta te postaviti parametre kao što su tekst i boja pozadine. Ovaj postupak se ponavlja za sve slikovne gumbe unutar aplikacije.



Slika 4.9. Pridruživanje virtualnog pina slikovnom gumbu



Slika 4.10. Umetanje slika za ON i OFF stanje slikovnog gumba

4.3. POVEZIVANJE ESP8226 S BLYNK APLIKACIJOM

Ovaj kod implementira sistem za daljinsko upravljanje animacijama putem Blynk platforme, koristeći napredne funkcionalnosti koje nudi Blynk.Edgent. Blynk.Edgent je modernizirana verzija Blynk biblioteke koja omogućava jednostavno povezivanje uređaja sa Blynk serverom, podržava OTA (Over-The-Air) ažuriranja i nudi veću fleksibilnost pri razvoju IoT aplikacija. Ovaj kod koristi Blynk kao sučelje između korisnika i uređaja, omogućujući kontrolu animacija putem virtualnih pinova. Za implementaciju ovog koda na ESP8266 korišten je Arduino IDE razvojno sučelje.

4.3.1. USPOSTAVLJANJE VEZE I KONFIGURACIJA

Na početku koda, nalaze se makro definicije koje specificiraju identifikatore predloška i verziju firmvera:

```
#define BLYNK_TEMPLATE_ID "TMPL[REDACTED]"
#define BLYNK_TEMPLATE_NAME "[REDACTED]"
#define BLYNK_FIRMWARE_VERSION "0.1.0"
#define BLYNK_PRINT Serial
//#define BLYNK_DEBUG
#define APP_DEBUG
```

Slika 4.11. Definiranje identifikatora i firmvera

- **BLYNK_TEMPLATE_ID** i **BLYNK_TEMPLATE_NAME** su identifikatori specifični za svaki Blynk projekt. Oni omogućuju aplikaciji da prepozna koji se predložak koristi i kako treba postupati.
- **BLYNK_FIRMWARE_VERSION** označava trenutnu verziju firmvera, što je korisno za OTA ažuriranja kako bi se osigurala kompatibilnost između servera i uređaja.
- **BLYNK_PRINT Serial** omogućuje ispisivanje informacija u serijskom monitoru, što pomaže u otklanjanju grešaka i praćenju toka programa.

Biblioteka **BlynkEdgent.h** se uključuje kako bi se koristile funkcije specifične za Blynk.Edgent platformu. Ova biblioteka upravlja povezivanjem sa Blynk serverom, rukuje autentifikacijom i omogućava napredne opcije kao što su OTA ažuriranja.

Funkcija **setup()** se izvršava samo jednom kada se uređaj pokrene (kada se mikro kontroler uključi ili resetira). Koristi za inicijalizaciju različitih parametara i postavljanje početnih postavki.

```
void setup() {  
  Serial.begin(115200);  
  delay(100);  
  
  BlynkEdgent.begin(); // Pokretanje Blynk.Edgent  
}
```

Slika 4.12. Setup funkcija

BlynkEdgent.begin() pokreće Blynk.Edgent, dio Blynk biblioteke koji je odgovoran za inicijalizaciju i povezivanje uređaja na Blynk server. Također omogućuje funkcionalnosti kao što su OTA (Over-The-Air) ažuriranja, gdje možete bežično ažurirati firmware na uređaju

Funkcija **loop()** se izvršava kontinuirano u petlji sve dok je uređaj uključen. Ovo je mjesto gdje se izvršava glavni kod koji kontrolira ponašanje uređaja.

```
void loop() {  
  BlynkEdgent.run(); // Održava vezu s Blynkom i omogućava OTA ažuriranja  
  
  // Dodaj bilo koju dodatnu funkcionalnost ovdje, kao što su provjere veze i drugo  
}
```

Slika

4.13. Loop funkcija

BlynkEdgent.run() funkcija održava aktivnu vezu između uređaja i Blynk servera. Ona osigurava da komunikacija s aplikacijom na mobilnom telefonu funkcionira ispravno, omogućuje daljinsko upravljanje i nadzor, te održava funkcionalnosti poput OTA ažuriranja.

4.3.2. IMPLEMENTACIJA I KONTROLA ANIMACIJA POMOĆU BLYNK VIRTUALNIH PINOVA

Nakon što je veza s Blynk aplikacijom uspostavljena, sljedeći korak je implementacija virtualnih pinova. Virtualni pinovi omogućuju dvosmjernu komunikaciju između aplikacije i mikro kontrolera, omogućujući upravljanje uređajem putem aplikacije i slanje podataka natrag na mobilni uređaj.

Kod koristi niz animations koji se sastoji od 18 bool vrijednosti, gdje svaka predstavlja status jedne animacije (uključeno/isključeno):

```
bool animations[18] = { false };  
int currentAnimation = -1;  
bool enableAnimations = false;
```

Slika 4.14. Niz animations

- **animations:** Svaki element ovog niza prati da li je određena animacija aktivna ili ne. Na početku, sve animacije su isključene (false).
- **currentAnimation** prati koja je animacija trenutno aktivna, omogućavajući sistemu da zna koja se animacija trenutno izvodi.
- **enableAnimations** je kontrolna varijabla koja omogućava ili onemogućava aktiviranje animacija. Ako je ova varijabla postavljena na false, nijedna animacija neće moći biti pokrenuta, bez obzira na komande iz Blynk aplikacije.

Dvije ključne funkcije u ovom kodu su **deactivateAllAnimations()** i **activateAnimation(int pinNumber)**:

```
void deactivateAllAnimations() {
  for (int i = 0; i < 18; i++) {
    if (animations[i]) {
      Blynk.virtualWrite(V1 + i, 0);
      animations[i] = false;
    }
  }
}
```

Slika 4.14. Funkcija deactivateAllAnimations()

deactivateAllAnimations(): Ova funkcija prolazi kroz cijeli niz animations i isključuje sve aktivne animacije. To postiže slanjem vrijednosti 0 na svaki od odgovarajućih virtualnih pinova (V1 do V18), čime se zaustavljaju sve animacije.

```
void activateAnimation(int pinNumber) {
  if (enableAnimations) {
    deactivateAllAnimations();
    Blynk.virtualWrite(V1 + pinNumber, 1);
    animations[pinNumber] = true;
    currentAnimation = pinNumber;
  } else {
    Blynk.virtualWrite(V1 + pinNumber, 0);
    Serial.println("Animacije su onemogućene. Nije moguće promijeniti stanje.");
  }
}
```

Slika 4.15. Funkcija activateAllAnimations()

activateAnimation(int pinNumber): Ova funkcija se koristi za aktiviranje određene animacije na specifičnom virtualnom pinu. Prije nego što aktivira novu animaciju, funkcija prvo deaktivira sve ostale animacije kako bi osigurala da samo jedna animacija bude aktivna u tom trenutku. Aktivacija se vrši slanjem vrijednosti 1 na odgovarajući virtualni pin. Ako su animacije globalno onemogućene (tj. enableAnimations je false), funkcija neće aktivirati animaciju i ispisat će poruku u serijskom monitoru.

U kodu se koriste funkcije **BLYNK_WRITE(Vx)** za svaki od 18 virtualnih pinova (V1 do V18), koje se pozivaju svaki put kada korisnik interagira s odgovarajućim elementom u Blynk aplikaciji:

```
BLYNK_WRITE(V1) { activateAnimation(0); }
BLYNK_WRITE(V2) { activateAnimation(1); }
BLYNK_WRITE(V3) { activateAnimation(2); }
BLYNK_WRITE(V4) { activateAnimation(3); }
BLYNK_WRITE(V5) { activateAnimation(4); }
BLYNK_WRITE(V6) { activateAnimation(5); }
BLYNK_WRITE(V7) { activateAnimation(6); }
BLYNK_WRITE(V8) { activateAnimation(7); }
BLYNK_WRITE(V9) { activateAnimation(8); }
BLYNK_WRITE(V10) { activateAnimation(9); }
BLYNK_WRITE(V11) { activateAnimation(10); }
BLYNK_WRITE(V12) { activateAnimation(11); }
BLYNK_WRITE(V13) { activateAnimation(12); }
BLYNK_WRITE(V14) { activateAnimation(13); }
BLYNK_WRITE(V15) { activateAnimation(14); }
BLYNK_WRITE(V16) { activateAnimation(15); }
BLYNK_WRITE(V17) { activateAnimation(16); }
BLYNK_WRITE(V18) { activateAnimation(17); }
```

Slika 4.16. Deklaracija funkcije BLYNK_WRITE(Vx)

Svaka od ovih funkcija poziva **activateAnimation()** sa odgovarajućim indeksom, omogućavajući korisniku da putem Blynk aplikacije aktivira željenu animaciju.

Funkcija **BLYNK_WRITE(V0)** kontrolira je su li animacije globalno omogućene ili ne:

```
BLYNK_WRITE(V0) {
  int pinValue = param.asInt();
  if (pinValue == 1) {
    enableAnimations = true;
    Serial.println("Animacije su omogućene.");
  } else {
    enableAnimations = false;
    deactivateAllAnimations();
    Serial.println("Animacije su onemogućene. Sva stanja su postavljena na nulu.");
  }
}
```

Slika 4.17. Funkcija *BLYNK_WRITE(V0)*

Kada korisnik promijeni vrijednost na virtualnom pinu V0 u Blynk aplikaciji, ova funkcija se poziva. Ako je vrijednost 1, animacije se omogućuju. Ako je vrijednost 0, sve animacije se deaktiviraju, a **enableAnimations** se postavlja na false, čime se sprječava aktivacija bilo koje animacije dok se ponovo ne omoguće.

Na kraju, funkcija **BLYNK_CONNECTED()** se poziva automatski kada se uređaj poveže na Blynk server:

```
BLYNK_CONNECTED() {
  Serial.println("Povezano na Blynk server.");
}
```

Slika 4.18. Funkcija *BLYNK_CONNECTED()*

Ova funkcija jednostavno ispisuje poruku u serijski monitor kako bi potvrdila da je uređaj uspješno povezan.

```

1  #define BLYNK_TEMPLATE_ID "TMPL4HCqFA46y"
2  #define BLYNK_TEMPLATE_NAME "ON OFF"
3
4  #define BLYNK_FIRMWARE_VERSION    "0.1.0"
5
6  #define BLYNK_PRINT Serial
7  //#define BLYNK_DEBUG
8
9  #define APP_DEBUG
10
11 #include "BlynkEdgent.h"
12
13 // Varijable za status animacija
14 bool animations[18] = { false };
15
16 // Trenutno aktivna animacija
17 int currentAnimation = -1;
18
19 // Kontrolni pin za omogućavanje animacija
20 bool enableAnimations = false;
21
22 void setup()
23 {
24   Serial.begin(115200);
25   delay(100);
26
27   BlynkEdgent.begin(); // Pokretanje Blynk.Edgent
28 }
29
30 void loop() {
31   BlynkEdgent.run(); // Održava vezu s Blynkom i omogućava OTA ažuriranja
32
33   // Dodaj bilo koju dodatnu funkcionalnost ovdje, kao što su provjere veze i drugo
34 }
35
36 // Funkcija koja isključuje sve animacije
37 void deactivateAllAnimations() {
38   for (int i = 0; i < 18; i++) {
39     if (animations[i]) {
40       Blynk.virtualWrite(V1 + i, 0); // Isključuje animaciju
41       animations[i] = false;
42     }
43   }
44 }
45
46 // Funkcija koja aktivira animaciju na odabranom virtualnom pinu
47 void activateAnimation(int pinNumber) {
48   if (enableAnimations) { // Provjera da li su animacije dozvoljene
49     deactivateAllAnimations(); // Isključuje sve animacije
50     Blynk.virtualWrite(V1 + pinNumber, 1); // Aktivira animaciju na odabranom pinu
51     animations[pinNumber] = true;
52     currentAnimation = pinNumber;
53   } else {
54     // Ako su animacije onemogućene, postavi sve vrijednosti na nulu
55     Blynk.virtualWrite(V1 + pinNumber, 0);
56     Serial.println("Animacije su onemogućene. Nije moguće promijeniti stanje.");
57   }
58 }
59
60 // BLYNK_WRITE funkcije za svaki virtualni pin
61 BLYNK_WRITE(V1) { activateAnimation(0); }
62 BLYNK_WRITE(V2) { activateAnimation(1); }
63 BLYNK_WRITE(V3) { activateAnimation(2); }
64 BLYNK_WRITE(V4) { activateAnimation(3); }
65 BLYNK_WRITE(V5) { activateAnimation(4); }
66 BLYNK_WRITE(V6) { activateAnimation(5); }
67 BLYNK_WRITE(V7) { activateAnimation(6); }
68 BLYNK_WRITE(V8) { activateAnimation(7); }
69 BLYNK_WRITE(V9) { activateAnimation(8); }
70 BLYNK_WRITE(V10) { activateAnimation(9); }
71 BLYNK_WRITE(V11) { activateAnimation(10); }
72 BLYNK_WRITE(V12) { activateAnimation(11); }
73 BLYNK_WRITE(V13) { activateAnimation(12); }
74 BLYNK_WRITE(V14) { activateAnimation(13); }
75 BLYNK_WRITE(V15) { activateAnimation(14); }
76 BLYNK_WRITE(V16) { activateAnimation(15); }
77 BLYNK_WRITE(V17) { activateAnimation(16); }
78 BLYNK_WRITE(V18) { activateAnimation(17); }
79
80 BLYNK_WRITE(V0) {
81   int pinValue = param.asInt();
82   if (pinValue == 1) {
83     enableAnimations = true;
84     Serial.println("Animacije su omogućene.");
85   } else {
86     enableAnimations = false;
87     deactivateAllAnimations(); // Isključi sve animacije
88     Serial.println("Animacije su onemogućene. Sva stanja su postavljena na nulu.");
89   }
90 }
91
92 // Funkcija koja se poziva kada se uređaj poveže na Blynk server
93 BLYNK_CONNECTED() {
94   Serial.println("Povezano na Blynk server.");
95 }
96

```

Slika 4.19. Cjelokupan kod za ESP8266

4.4. PISANJE KODA ZA UPRAVLJANJE RGB LED KOCKOM

U ovom poglavlju, fokus će biti na kodiranju koje omogućuje upravljanje RGB LED kockom pomoću Teensy 4.0 mikro kontrolera. Bit će prikazano kako implementirati različite svjetlosne efekte i animacije, te kako integrirati te funkcije s Blynk aplikacijom preko ESP8266 mikro kontrolera.

4.4.1. UPRAVLJANJE RGB LED KOCKOM

Ovdje je prikazan glavni kod pisan u VSC-u koji povezuje sve komponente kodova na jednom mjestu. Dio koda za prikaz animacija preuzet je iz vanjskih izvora [11], dok je dio koji se odnosi na serijsku komunikaciju napisan vlastoručno. Također, ovaj kod koristi PlatformIO, točnije platformio.ini datoteku koja učitava sve potrebne biblioteke za rad s kockom.

```
src > @ main.cpp
1 #include <Arduino.h>
2 #include "core/Config.h"
3 #include "core/ESP8266.h"
4 #include "core/LCD.h"
5 #include "space/animation.h"
6
7 // Deklaracija vanjskih varijabli
8 extern const uint8_t ANIMATIONS; // Broj animacija
9 extern Animation *Animations[]; // Niz pokazivača na animacije
10
11 /*-----
12  * Globalne varijable
13  *-----*/
14 Config config;
15
16 /*-----
17  * Inicijalizacija parametara u setup funkciji
18  *-----*/
19 void setup() {
20     // Počni s brisanjem pinovih LED dioda što je prije moguće
21     Animation::begin();
22     // Serijski izlaz na USB za prikaz na konzoli
23     Serial.begin(115200);
24     // ESP8266 UART brzina na Hardware Serial
25     Serial1.begin(460800);
26     // Sprečava prelijevanje RX međuspremnika ako se ne čita dovoljno brzo
27     DMAEN static char read_buffer[4096];
28     Serial1.addMemoryForRead(read_buffer, sizeof(read_buffer));
29     // Sprečava prelijevanje TX međuspremnika i blokiranje programa
30     DMAEN static char write_buffer[1024];
31     Serial1.addMemoryForWrite(write_buffer, sizeof(write_buffer));
32
33     // Pokretanje LCD drivera
34     LCD::begin();
35 }
36
37 // Funkcija za obradu serijskih poruka
38 void handleSerialInput() {
39     String command = Serial1.readStringUntil('\n'); // Čita cijelu liniju do '\n'
40     command.trim(); // uklanja sve nepotrebne praznine
41
42     if (command.startsWith("ANIMATION ")) {
43         int animationIndex = command.substring(10).toInt() - 1; // Dohvaća broj animacije (od 1 do N) i pretvara ga u indeks (0 do N-1)
44         if (animationIndex >= 0 && animationIndex < ANIMATIONS) {
45             Serial.print("Pokreće animaciju: ");
46             Serial.println(animationIndex + 1);
47             Animation::next(true, animationIndex); // Pokreće specifičnu animaciju
48             Serial.println("Nepoznata naredba.");
49         }
50     }
51 }
52
53 /*-----
54  * Početak glavne petlje
55  *-----*/
56 void loop() {
57     // Ispis FPS-a jednom svakih x sekundi
58     static Timer print_interval = 10.0f;
59     LCD::loop();
60
61     if (Serial1.available()) {
62         handleSerialInput(); // Poziv funkcije za rukovanje serijskim unosom
63     }
64
65     if (print_interval.update()) {
66         static char fps[20];
67         sprintf(fps, "FPS-%1.2f", Animation::fps());
68         Serial.println(fps);
69     }
70 }
71
72
73
74
75
76
77
```

Slika 4.20. Glavni kod za Teensy 4.0

Ovaj kod omogućuje upravljanje animacijama na RGB kocki koristeći serijsku komunikaciju između Teensyja i ESP8266 modula, dok se upravljanje obavlja putem Blynk aplikacije. Cjelokupni sustav se pokreće unutar `setup()` funkcije, gdje se prvo inicijalizira sustav za animacije pomoću `Animation::begin()`, koji osigurava da zaslon bude spreman za prikazivanje bilo kakvih vizualnih efekata odmah po uključivanju uređaja. Nadalje, serijska komunikacija se postavlja na `Serial1` kanal, koji je povezan s ESP8266 modulom za primanje naredbi. Postavke za serijsku komunikaciju na `Serial1` uključuju korištenje dodatnih međuspremnikâ za čitanje i pisanje, što omogućuje rukovanje velikim količinama podataka bez gubitaka i bez rizika od blokiranja programa.

Unutar glavne `loop()` funkcije, kod kontinuirano izvršava tri osnovne operacije: upravljanje animacijama, osvježavanje LCD zaslona, i provjeru dolaznih podataka putem serijskog porta. Animacije se osvježavaju kroz `Animation::loop()`, koja pokreće renderiranje okvira ako je zaslon dostupan, ažurira unutarnje timere za animacije, i osigurava da se aktivna animacija prikaže na zaslonu. Ako neka od animacija u nizu nije aktivna, funkcija `Animation::next()` odabire sljedeću animaciju iz unaprijed definiranog niza i pokreće je.

Jedan od ključnih elemenata koda je obrada dolaznih podataka putem serijskog porta, što omogućuje Teensyju da reagira na naredbe poslâne iz Blynk aplikacije preko ESP8266 modula. Funkcija `handleSerialInput()` provjerava da li su pristigli novi podaci na `Serial1` portu. Kada se otkrije dolazna poruka, ona se očitava i dešifrira kako bi se identificirala odgovarajuća animacija koja se treba pokrenuti. Očekuje se da poruka bude u obliku "ANIMATION_x", gdje 'x' predstavlja indeks animacije. Nakon što se prepozna broj animacije, `Animation::next()` aktivira tu specifičnu animaciju iz niza `Animations`, koji sadrži sve definirane animacije poput "firework", "plasma", "sinus", i drugih.

Uz ove osnovne funkcionalnosti, kod također periodično ispisuje broj sliĉica u sekundi (engl. FPS) svakih 10 sekundi na serijski izlaz povezan s USB-om. Ovaj dio koda služi kao dijagnostički alat za praćenje performansi sustava u stvarnom vremenu, omogućujući programeru da vidi koliko brzo sustav može renderirati animacije.

Cijeli sustav je dizajniran tako da može brzo i efikasno odgovoriti na naredbe pristigle iz Blynk aplikacije, obraditi ih na serijskom portu, i pokrenuti odgovarajuće animacije na zaslonu. Korištenje dodatnih međuspremnikâ za serijsku komunikaciju i redovito osvježavanje zaslona osigurava stabilan rad bez rizika od preopterećenja ili gubitka podataka. Time se osigurava pouzdan rad sustava i učinkovito upravljanje složenim animacijama u stvarnom vremenu.

5. PREDNOSTI I NEDOSTACI RAZVIJENOG RJEŠENJA

U ovom dijelu razmatrat će se sve prednosti i nedostaci vezani uz izradu i implementaciju programske podrške za interaktivnu RGB LED kocku. Pokrit će se fleksibilnost, funkcionalnost, mogućnosti proširenja i tehničke karakteristike sustava, uključujući upotrebu Blynk aplikacije, Teensy 4.0 mikro kontrolera, i ESP8266 modula za bežičnu komunikaciju. Također će se obraditi nedostaci kao što su financijski troškovi, ovisnost o aplikaciji, složenost konfiguracije, ovisnost o Wi-Fi vezi te ograničenja hardverskih resursa. Ovaj pregled pružit će cjelovitu sliku o potencijalu i izazovima sustava.

5.1. PREDNOSTI

Izrada programske podrške za interaktivnu RGB LED kocku donosi brojne prednosti u smislu fleksibilnosti, funkcionalnosti i mogućnosti proširenja sustava. Upotrebom Blynk aplikacije omogućeno je intuitivno daljinsko upravljanje LED kockom putem mobilnog uređaja, pružajući korisnicima jednostavan način za prilagodbu svjetlosnih efekata i animacija. Blynk platforma nudi jednostavno dodjeljivanje i konfiguriranje virtualnih pinova, što olakšava integraciju novih funkcionalnosti bez potrebe za dubokim tehničkim znanjem, čime se povećava razina personalizacije i interaktivnosti.

Korištenje Teensy 4.0 mikro kontrolera omogućuje visoku brzinu obrade podataka, što je ključno za izvođenje kompleksnih animacija u stvarnom vremenu. Ovaj napredni mikro kontroler omogućuje precizno upravljanje LED svjetlom i stvaranje sofisticiranih vizualnih efekata, omogućujući fluidne i dinamične animacije.

Integracija ESP8266 modula za bežičnu komunikaciju dodatno poboljšava mobilnost i praktičnost uređaja, omogućujući jednostavno povezivanje s različitim mrežama i uređajima. Ovaj modul omogućuje komunikaciju s internetom i drugim uređajima bez potrebe za fizičkim kablovima, što povećava fleksibilnost u korištenju LED kocke.

Mogućnost proširenja sustava putem Blynk aplikacije i ESP8266 modula omogućuje jednostavno dodavanje novih funkcionalnosti, poput integracije sa pametnim kućnim sustavima ili automatizacijom događaja temeljenih na senzorskim podacima. Ovo dodatno

proširuje mogućnosti i potencijal LED kocke, omogućujući korisnicima stvaranje jedinstvenih vizualnih prikaza i interakcija prema svojim specifičnim potrebama.

5.2. NEDOSTACI

S druge strane, postoji nekoliko nedostataka prilikom implementacije ovog sustava. Jedan od značajnijih nedostataka je potreba za plaćanjem mjesečne pretplate za Blynk aplikaciju kako bi se dobile dodatne mogućnosti koje su bile potrebne za izradu projekta. Ova dodatna financijska obaveza može biti neugodna, pogotovo ako se uzme u obzir da su mnoge ključne funkcionalnosti bile vezane uz ovu pretplatu.

Također, ovisnost o Blynk aplikaciji i njenim uslugama može predstavljati problem ako dođe do prekida u radu aplikacije ili ako se ona prestane razvijati ili podržavati. Iako je mogućnost takvog scenarija mala, ona i dalje postoji, što može izazvati nesigurnost u dugoročnu održivost sustava.

Proces konfiguracije virtualnih pinova može postati složen i zamoran u projektima s većim brojem animacija i funkcionalnosti, što povećava mogućnost pogrešaka. Korištenje više virtualnih pinova može zahtijevati dodatno vrijeme za testiranje i ispravljanje bugova, što može usporiti razvojni proces i dovesti do frustracije.

Budući da se komunikacija između modula i aplikacije oslanja na Wi-Fi vezu, svaka smetnja ili nestabilnost u mreži može utjecati na pouzdanost i performanse cijelog sustava. U područjima s lošom ili nestabilnom Wi-Fi pokrivenošću, performanse sustava mogu biti značajno degradirane, što može umanjiti korisničko iskustvo.

Na kraju, iako je Teensy 4.0 izuzetno moćan mikro kontroler, zahtjevni projekti mogu iscrpiti njegove resurse, što može dovesti do ograničenja u složenosti animacija ili broju istovremeno aktivnih funkcija. Ovaj nedostatak može zahtijevati optimizaciju koda ili kompromis u pogledu funkcionalnosti, čime se može smanjiti ukupni potencijal sustava.

6. ZAKLJUČAK

U ovom završnom radu obrađena je izrada programske podrške za interaktivnu RGB LED kocku, koja korisnicima omogućuje kreiranje i upravljanje raznim svjetlosnim animacijama putem jednostavnog sučelja Blynk aplikacije. Kroz proces projektiranja i implementacije, pokazalo se kako je moguće koristiti moćne mikro kontrolere poput Teensy 4.0 i ESP8266 modula za izradu složenih sustava koji nude visoku funkcionalnost uz jednostavno upravljanje. Prednosti ovog pristupa ogledaju se u fleksibilnosti sustava i mogućnosti daljinskog upravljanja, dok su glavni nedostaci povezani s ovisnošću o vanjskim aplikacijama, potrebom za plaćanjem mjesečne pretplate te stabilnosti mreže. Unatoč tim nedostacima, završni rad je uspješno realiziran, demonstrirajući kako se moderni alati i tehnologije mogu koristiti za izradu naprednih interaktivnih uređaja koji pružaju bogato korisničko iskustvo. Ovaj završni rad predstavlja solidnu osnovu za daljnji razvoj i nadogradnju sustava, s potencijalom za proširenje funkcionalnosti i poboljšanje performansi u budućim iteracijama projekta.

LITERATURA:

- [1]. LED diodes – characteristics of key parameters, pristup 03.09.2024.,
<https://intercom.com/en/knowledge-base/led-diode-characteristics-of-the-most-important-parameters-64757>
- [2]. Safe O. Kasapa : "Optoelectronics and Photonics: Principles and Practices", 2001.g., pristup 03.09.2024.,
https://www.academia.edu/41062157/Optoelectronics_and_Photonics_Principles_and_Practices_S_O_Kasap_Prentice_Hall
- [3]. “RGB LED: sve što trebate znati o ovoj komponenti”, pristup 03.09.2024.,
<https://www.hwlibre.com/hr/rgb-vodio/>
- [4]. “ESP8266 data sheet”, pristup 03.09.2024.,
https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf
- [5]. “Teensy 4.0 Pinout, Specifications & Board Layout”, pristup 03.09.2024.,
<https://www.etechnophiles.com/teensy-4-0-pinout/>
- [6]. PlatformIO, pristup 03.09.2024., <https://docs.platformio.org/en/latest/what-is-platformio.html>
- [7]. Somayya Madakam, Ramaswamy, R., & Tripathi, S. (2015). Internet of Things (IoT): A Literature Review. Journal of Computer and Communications, 3, 164–173., pristup 03.09.2024., <https://doi.org/10.4236/jcc.2015.35021>
- [8]. Farooq, M. U., Waseem, M., Mazhar, S., Khairi, A., & Kamal, T. (2015). A Review on Internet of Things (IoT). International Journal of Computer Applications, 113, 1–7., pristup 03.09.2024., <https://doi.org/10.5120/19787-1571>
- [9]. Pregled Blynk.Edgent-a, pristup 03.09.2024.,
<https://docs.blynk.io/en/blynk.edgent/overview>
- [10]. BlynkEdgent ESP8266 Library, pristup 03.09.2024., https://github.com/blynkkk/blynk-library/tree/master/examples/Blynk.Edgent/Edgent_ESP8266

[11]. Malt Whiskey, Mega Cube, pristup 03.09.2024.,

<https://github.com/MaltWhiskey/Mega-Cube/tree/master/Software/LED%20Display>

POPIS OZNAKA I KRATICA

LED - Light-Emitting Diode

IDE - Integrated Development Environment

RGB – Red Green Blue

WiFi – Wireless Fidelity

IoT – Internet of Things

LCD - Liquid Crystal Display

SPI - Serial Peripheral Interface

URL - Uniform Resource Locator

USB - Universal Serial Bus

I2C - Inter-Integrated Circuit

UART - Universal Asynchronous Receiver/Transmitter

CAN - Controller Area Network

VSC – Visual Studio Code

RFID - Radio Frequency Identification

NFC - Near-field communication

OTA - Over-The-Air

GND – Ground

SAŽETAK

U okviru ovog završnog rada razvijena je programska podrška za interaktivnu RGB LED kocku, koristeći Teensy 4.0 mikro kontroler i ESP8266 Wi-Fi modul za upravljanje i povezivanje s mobilnom aplikacijom Blynk. Serijska komunikacija između ova dva mikro kontrolera omogućuje sinkronizaciju i izvršavanje komandi poslanih iz aplikacije. Glavni cilj projekta bio je omogućiti jednostavno upravljanje svjetlosnim efektima i animacijama putem intuitivnog korisničkog sučelja. Korišten je Arduino IDE za programiranje ESP8266 modula, dok je kod za upravljanje animacijama na Teensy 4.0 razvijen u Visual Studio Code-u. Projekt obuhvaća opis hardverske platforme, dizajn aplikacije, implementaciju koda te analizu prednosti i nedostataka odabranog pristupa.

Ključne riječi: Arduino IDE, Visual Studio Code, Blynk, RGB LED kocka

ABSTRACT

This bachelor thesis explores the development of software support for an interactive RGB LED cube, utilizing the Teensy 4.0 microcontroller and the ESP8266 Wi-Fi module for control and connection with the Blynk mobile application. Serial communication between these two microcontrollers enables synchronization and execution of commands sent from the application. The main goal of the project was to enable easy control of lighting effects and animations through an intuitive user interface. The Arduino IDE was used for programming the ESP8266 module, while the code for controlling the animations on the Teensy 4.0 was developed in Visual Studio Code. The project includes a description of the hardware platform, application design, code implementation, and an analysis of the advantages and disadvantages of the chosen approach.

Keywords: Arduino IDE, Visual Studio Code, Blynk, RGB LED cube

ŽIVOTOPIS

Željko Kosović rođen je 10.06.2001. godine u Pakracu. Od rođenja živi u Kutini, gdje stječe osnovnoškolsko obrazovanje u OŠ Mate Lovraka. Godine 2016. upisuje srednju Tehničku školu u Kutini, smjer Računalstvo, te nakon završetka srednje škole i polaganja državne mature upisuje preddiplomski stručni studij Automatike na fakultetu Elektrotehnike, Računarstva i Informatičkih tehnologija u Osijeku.