Minecraft lampa

Cvrtila, Jana

Undergraduate thesis / Završni rad

2023

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://urn.nsk.hr/urn:nbn:hr:200:006788

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-11-30**

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

Sveučilišni studij

MINECRAFT LAMPA

Završni rad

Jana Cvrtila

Osijek, 2023.

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

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

Osijek, 04.09.2023.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime Pristupnika:	Jana Cvrtila
Studij, smjer:	Sveučilišni prijediplomski studij Elektrotehnika i informacijska tehnologija
Mat. br. Pristupnika, godina upisa:	4652, 22.07.2019.
OIB Pristupnika:	92089550535
Mentor:	izv. prof. dr. sc. Ivan Aleksi
Sumentor:	,
Sumentor iz tvrtke:	
Naslov završnog rada:	Minecraft lampa
Znanstvena grana rada:	**Arhitektura računalnih sustava (zn. polje računarstvo)**
Zadatak završnog rad:	U ovom završnom radu potrebno je izraditi maketu minecraft lampe koristeći Arduino Nano, WS2812B niz dioda, Li-Ion baterije te 3D printer za ispis modela. Na stisak tipke na maketi potrebno je mijenjati boju lampe prema bojama iz Minecraft igrice: redstone, gold, copper, emerald, lapiz, diamond, itd.
Prijedlog ocjene završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	04.09.2023.
Datum potvrde ocjene od strane Odbora:	24.09.2023.

Potvrda mentora o predaji konačne verzije rada:	*Mentor elektronički potpisao predaju konačne verzije.*
	Datum:

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

IZJAVA O ORIGINALNOSTI RADA

Osijek, 21.09.2023.

Ime i prezime studenta:	Jana Cvrtila
Studij:	Sveučilišni prijediplomski studij Elektrotehnika i informacijska tehnologija
Mat. br. studenta, godina upisa:	4652, 22.07.2019.
Turnitin podudaranje [%]:	3

Ovom izjavom izjavljujem da je rad pod nazivom**: Minecraft lampa**

izrađen pod vodstvom mentora izv. prof. dr. sc. Ivan Aleksi

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. PRIMJENJENE TEHNOLOGIJE 2
2.1. Croduino Nova 2 2
2.2. Arduino 2
2.3. Komponente 2
2.3.1. LED traka 3
2.3.2. Potenciometar 3
2.3.3. Baterija 3
3. PROGRAMSKI KOD 5
3.1. Definiranje pinova i varijabli 5
3.2. Funkcija *setup()* 6
3.3. Funkcija *loop()* 7
3.4. Ostale funkcije 8
4. 3D ISPIS 10
4.1. Dizajn 10
4.2. Ispis 13
4.3. Sastavljanje stranica 14
5. ZAKLJUČAK 19
LITERATURA 20
SAŽETAK 21
ABSTRACT 22

1. UVOD

Cilj je ovog završnog rada izrada Minecraft lampe pomoću 3D printera. Minecraft [1] je popularna videoigra koja igračima pruža visoki stupanj kreativnosti obično bez ikakvog unaprijed određenog cilja. Poznata je po svom jednostavnom dizajnu te velikim mogućnostima prilagodbe. U igri igrači istražuju svjetove sastavljene od blokova koje mogu oblikovati prema vlastitoj želji. Upravo je ova lampa dobila takvo ime zbog svoje prepoznatljive blokovske strukture. Lampa je upravljana mikroupravljačkom pločicom Croduino Nova 2 koja se temelji na softverskoj platformi Arduino. Ovaj se rad bavi dizajnom, razvojem i implementacijom Minecraft lampe. Svrha je lampe omogućiti promjenu između pet različitih boja pritiskom na gumb ili putem detekcije zvuka na potenciometru. Uz mikroupravljačku pločicu se koriste LED traka, gumb, potenciometar te baterija.

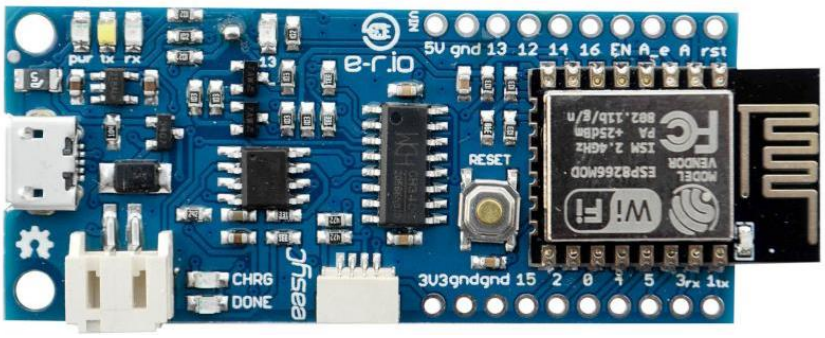
1.1. Zadatak završnog rada

U ovom je radu potrebno izraditi stranice lampe pomoću 3D printera. Potrebno je spojiti potrebne dijelove za rad lampe sa mikroupravljačkom pločicom te implementirati kod za mijenjanje boja na pločicu. Pločica ima JST konektor za litijsku bateriju kojom je omogućeno da ne mora biti stalno spojena na napajanje.

2. PRIMJENJENE TEHNOLOGIJE

2.1. Croduino Nova 2

Glavna komponenta Croduino Nova 2 [2] (Slika 2.1.) pločice je ESP8266 integrirani sklop nadopunjen integriranim TCP/IP stogom. ESP8266 [3] se može programirati putem Arduina te nudi 1MB prostora za kod. Ova pločica ima 9 GPIO (*General-Purpose Input/Output*) pinova, uz jedan analogni ulaz. Dodatno, uključuje CH340 čip za USB komunikaciju, regulator napona od 3,3 V te 4 LE diode. Ploča uključuje i *easyC* konektor koji olakšava povezivanje s različitim uređajima. Nadalje, ploča je opremljena standardnim JST konektorom koji omogućuje integraciju s litijskim baterijama za bežičnu funkcionalnost projekta. Punjenje baterije je jednostavno, samo se spoji USB kabel na pločicu, a ugrađeni punjač će napraviti sve ostalo.



Slika 2.1. *Croduino Nova 2 razvojna pločica*

2.2. Arduino

Arduino [4] platforma se sastoji od elektroničkih i softverskih elemenata koji se povezuju jedni s drugima omogućujući korisnicima da s lakoćom konstruiraju željene elektroničke sklopove. Arduino se sastoji od mnoštva mikroprocesora i mikroupravljača te je opremljen s digitalnim i analognim ulazno/izlaznim pinovima. Koriste se C i C++ programski jezici za programiranje, a hardver je otvorenog koda. To znači da se informacije, odnosno dizajn o hardveru lako mogu uočiti kako bi ga drugi mogli modificirati prema vlastitoj potrebi. Arduino je postao vrlo popularan jer je jednostavan za upotrebu, lako dostupan te nudi široki spektar mogućnosti.

2.3. Komponente

U ovom završnom radu korištene su sljedeće komponente:

1. LED traka
2. Tipka

3. Potenciometar

4. Baterija

2.3.1. LED traka

LED ili svjetleće diode [5] su poluvodički uređaji koji pretvaraju električni signal u optički. Elektroni se u poluvodiču spajaju s elektronskim prazninama, otpuštajući energiju u obliku svjetlosnih čestica, poznatih kao fotona. Boja je svjetlosti određena energijom potrebnom da elektroni prijeđu zabranjeni pojas poluvodiča. U ovom radu korištena je WS2812b LED traka (Slika 2.2.).

Slika 2.2. *LED traka*

2.3.2. Potenciometar

Potenciometar (Slika 2.3.) [6] je ručno podesivi promjenjivi otpornik s tri priključka. Dva su priključka spojena na suprotne krajeve otpornog elementa, dok je treći povezan s kliznim kontaktom koji se kreće preko otpornog elementa.

Slika 2.3. *Potenciometar*

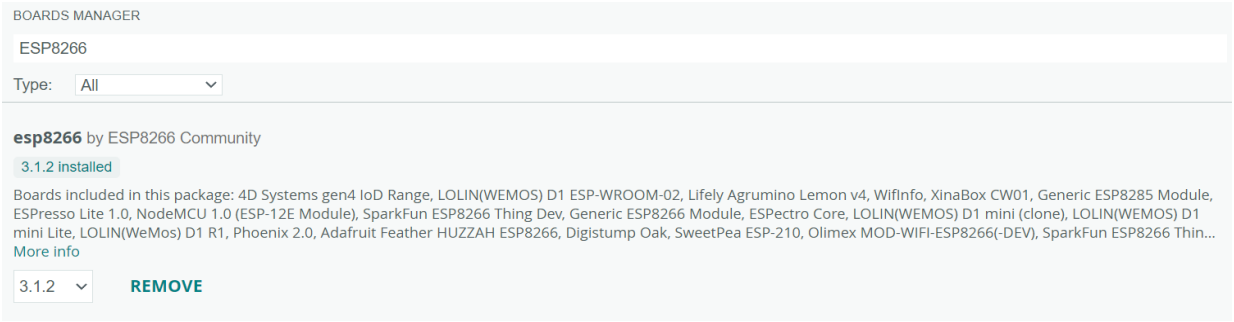
2.3.3. Baterija

Litij-ionske baterije [7] se sastoje od jedne ili više litij-ionskih ćelija, zajedno sa zaštitnom pločicom. U litij-ionskoj bateriji, litijevi se ioni u unutrašnjosti kreću između katode i anode, dok se elektroni u vanjskom krugu kreću u suprotnom smjeru te se zbog toga stvara električna struja koja omogućuje bateriji da napaja uređaj. Baterija korištena za ovaj rad nalazi se na slici 2.4.

Slika 2.4. *Baterija*

3. PROGRAMSKI KOD

U ovom je poglavlju opisan programski kod koji je napisan u Arduino IDE okruženju. Spajanjem ESP8266 mikroupravljača s računalom, potrebno je prilagoditi postavke kako bi se mogla koristiti željena biblioteka. Prvi je korak pronalazak ESP8266 u *Boards Manager-u* (Slika 3.1.). Kada je preuzeta željena biblioteka za rad sa ESP8266 slijedi odabir odgovarajuće pločice i *port-a* u kartici *Tools*.



Slika 3.1. *Prikaz Boards Manager-a u Arduino IDE*

3.1. Definiranje pinova i varijabli

U ovom je radu korištena biblioteka *Adafruit NeoPixel*. U dijelu koda na slici 3.2. definirani su pinovi te varijabla *LONGPRESS_OFF_TIME* koja predstavlja vremenski prag koji se koristi za odlučivanje hoće li dugi pritisak na tipku isključiti uređaj. Ako korisnik pritisne tipku dulje od tog vremena, uređaj će se isključiti. *WIPE_TIME* je broj koji predstavlja vremenski interval koji kontrolira brzinu izmjene boja. Što je taj broj veći to će prelazak boje biti sporiji.

```
#include <Adafruit_NeoPixel.h>

#define BUTTON_PIN          12
#define PIXEL_PIN           13
#define PIXEL_COUNT         15
#define MIC_PIN             14
#define LONGPRESS_OFF_TIME  1000
#define WIPE_TIME           315
```

Slika 3.2. *Definirani priključci*

```
boolean oldState = HIGH;
int mode     = 0;
unsigned int longPressTime = 0;
```

Slika 3.3. *Definirane varijable*

Na slici 3.3. su prikazane još neke varijable potrebne za rad sklopa. Varijabla *oldState* služi za praćenje prethodnog stanja tipke, a njeno postavljanje na *HIGH* znači da tipka nije pritisnuta. Ona se koristi kasnije u kodu za usporedbu sa trenutnim stanjima tipke. Varijabla *mode* prati trenutni način rada u kojem se uređaj nalazi, a on se mijenja ili pritiskom na tipku ili detekcijom zvuka te njeno postavljanje na '0' znači da je uređaj trenutno isključen. Također, dodana je i varijabla *longPressTime* koja prati vrijeme tokom dugog pritiska tipke. Dugim pritiskom tipke mjeri se koliko je vremena prošlo dok je tipka pritisnuta. Vrijednost ove varijable se povećava kako vrijeme prolazi dok je tipka pritisnuta. Kada ta varijabla dosegne određenu vrijednost (koja je definirana vrijednošću LONGPRESS_OFF_TIME) dolazi do reakcije na tu situaciju. Na temelju vrijednosti *longPressTime* donosi se odluka hoće li se uređaj isključiti ili promijeniti boju.

3.2. Funkcija *setup()*

Postoje dvije osnovne funkcije, *setup()* i *loop()*. U *setup()* se funkciji piše program koji se pokreće pri uključivanju ili nakon resetiranja mikroupravljača. Taj se program izvodi samo jednom te se u njemu postavljaju željena stanja pinova.

```
void setup() {
  pinMode(BUTTON_PIN, INPUT_PULLUP);
  pinMode(MIC_PIN,INPUT);
  strip.begin();
  strip.show();
}
```

Slika 3.4. *Prikaz funkcije setup()*

Na slici 3.4. je prikazano postavljanje pinova za tipku i potenciometar na INPUT_PULLUP i INPUT jer su to ulazni pinovi. Korištene su i dvije metode, *strip.begin()* i *strip.show()*. *Strip.begin()* omogućava komunikaciju između mikroupravljača i LED trake, dok *strip.show()* prenosi postavljene podatke u kodu na LED traku.

3.3. Funkcija *loop()*

Nakon što funkcija *setup()* prestane s izvođenjem, izvršava se funkcija *loop()*. Funkcija *loop()* predstavlja petlju koja se stalno iznova izvršava beskonačan broj puta. U ovoj se funkciji pozivaju još neke dodatno napisane funkcije koje će biti opisane kasnije u daljnjem tekstu.

```
void loop() {

  unsigned long startTime = millis();
  boolean newState = digitalRead(BUTTON_PIN);
  int soundValue = digitalRead(MIC_PIN);

  if ((newState == LOW) && (oldState == HIGH)) {
    delay(20);
    newState = digitalRead(BUTTON_PIN);
    if (newState == LOW) {
      if (++mode > 5)
        mode = 0;
      startShow(mode, WIPE_TIME / PIXEL_COUNT);
    }
  } else if (newState == LOW && oldState == LOW && longPressTime >= LONGPRESS_OFF_TIME) {
    mode = 0;
    startShow(mode, 0);
    longPressTime = 0;
  }
  if (soundValue) {
    if (++mode > 5)
      mode = 0;
    startShow(mode, WIPE_TIME / PIXEL_COUNT);
  }
  unsigned long stopTime = millis();
  updateLongPress(startTime, stopTime, newState);
  oldState = newState;
}
```

Slika 3.5. *Prikaz funkcije loop()*

Na slici 3.5. je prikazana *loop()* funkcija koja u ovom slučaju čita stanje tipke i zvuka te ovisno o tom stanju postavlja odgovarajuće efekte na LED traci koristeći *startShow()* funkciju. Također prati vrijeme dugog pritiska na tipki. Na početku petlje bilježimo trenutno vrijeme u milisekundama, zatim čitamo trenutno stanje tipke i trenutnu vrijednost potenciometra. Nakon toga *if* naredba provjerava je li tipka pritisnuta te se postavlja kratka odgoda (*delay*) zbog šuma da nebi došlo do krivo detektiranih pritisaka. Slijedi ponovno čitanje stanja tipke te ako je ona i dalje pritisnuta, varijabla *mode* se povećava za 1 te se resetira ako prođe broj 5 (broj mogućih boja). Pokreće se funkcija koja postavlja boje ovisno o varijabli *mode*. Nadalje, provjerava se dugi pritisak na tipki te ako je do njega došlo *mode* se postavlja na 0 i isključuju se svi efekti. U sljedećoj se *if* naredbi provjerava je li detektiran zvuk na potenciometru te ako je događa se isti proces kao i kod pritiska tipke. Naposljetku, bilježi se trenutno vrijeme na kraju petlje, ažurira se vrijeme dugog pritiska tipke te se postavlja prethodno stanje tipke za sljedeći ciklus petlje.

3.4. Ostale funkcije

Kako je u prošlom odlomku napisano u funkciji *loop()* pozvane su još neke dodatne funkcije koje su ključne za rad ovog projekta. Prva je takva funkcija *startShow()* (Slika 3.6.) koja je odgovorna za postavljanje određenih boja na LED traku.

```
void startShow(int i, uint8_t wipeTime) {
  switch (i) {
    case 0:
      colorWipe(strip.Color(0, 0, 0), wipeTime);  // Black/off
      break;
    case 1:
      colorWipe(strip.Color(255, 0, 0), wipeTime);  // Redstone
      break;
    case 2:
      colorWipe(strip.Color(0, 255, 0), wipeTime);  // Emerald
      break;
    case 3:
      colorWipe(strip.Color(0, 0, 255), wipeTime);  // Lapis Lazuli
      break;
    case 4:
      colorWipe(strip.Color(255, 100, 0), wipeTime);  // Gold
      break;
    case 5:
      colorWipe(strip.Color(100, 245, 228), wipeTime);  // Diamond
      break;
  }
}
```

Slika 3.6. *Prikaz funkcije startShow()*

Funkciji je predan argument *wipeTime* koji označava vremenski interval za kontrolu brzine prelaska boja. Ta vrijednost se dijeli s brojem LE dioda na traci kako bi se moglo odrediti koliko će vremena svaka LE dioda zadržati određenu boju prije prelaska na drugu. On je ovdje postavljen na vrijednost 315, a broj je LE dioda 15 (definirano s *PIXEL_COUNT*) što znači da će svaka LE dioda zadržati istu boju 21 milisekundu prije nego što pređe na sljedeću. Koristi se *switch* izraz koji određuje koju boju treba postaviti na LED traku na temelju vrijednosti iteratora. Ima pet slučaja (engl. *case*) gdje svaki slučaj odgovara jednom načinu rada, odnosno boji.

```
void colorWipe(uint32_t color, uint8_t wait) {
  for (int i = 0; i < strip.numPixels(); i++) {
    strip.setPixelColor(i, color);
    strip.show();
    delay(wait);
  }
}
```

Slika 3.7. *Prikaz funkcije colorWipe()*

Na slici 3.7. je prikazana još jedna ključna funkcija *colorWipe()*. Ova je funkcija odgovorna za efekt postupnog prelaska boje, odnosno „brisanja“ boje preko LED trake. Funkcija *for* petlje je da prolazi kroz svaki piksel na LED traci te postavlja boju trenutnog piksela na zadanu boju (argument „*color*“). Nakon postavljanja boje na trenutni piksel, poziva se funkcija *strip.show()* kako bi se ta ista boja prikazala i na fizičkoj LED traci. Nakon prikaza boja, funkcija pravi pauzu (engl. *delay*) koja daje dovoljno vremena da se ta boja primijeti prije nego što prijeđe na sljedeći piksel u petlji.

```
void updateLongPress(unsigned long startTime, unsigned long stopTime, bool newState) {
  if (newState == HIGH) {
    longPressTime = 0;
  } else {
    unsigned long diff;
    if (stopTime < startTime) {
      diff = 4294967295L - startTime + stopTime;
    } else {
      diff = stopTime - startTime;
    }
    longPressTime += diff;
  }
}
```

Slika 3.8. *Prikaz funkcije updateLongPress()*

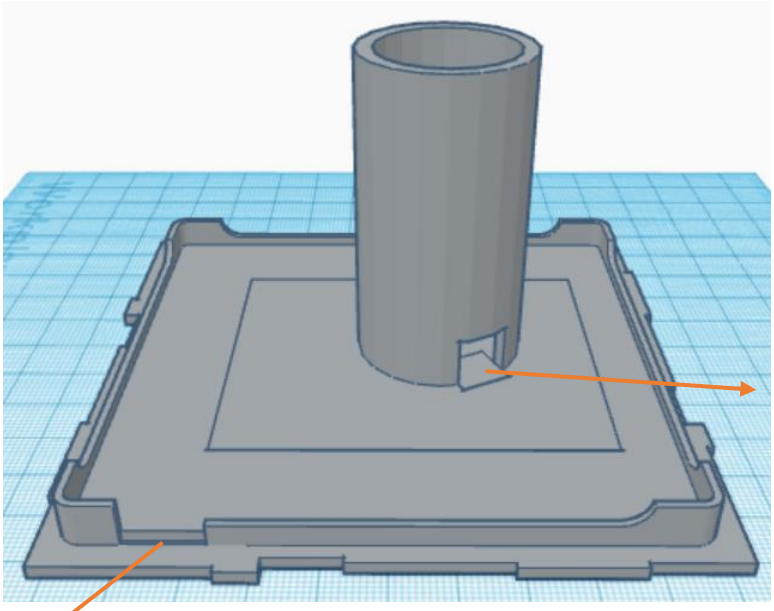
Zadnja funkcija koja se koristi za pravilan rad lampe je *updateLongPress()* koja prati trajanje dugog pritiska tipke. Ako tipka nije pritisnuta, postavlja varijablu za dugi pritisak na nulu, ali ako je pritisnuta tada računa vremensku razliku između trenutnog i proteklog vremena. U drugom slučaju, ako je se premaši vrijeme brojanja, koristi se posebna metoda za računanje razlike.

4. 3D ISPIS

3D ispis je proces stvaranja trodimenzionalnih predmeta pomoću dizajna koji se radi na računalu. Materijal se nanosi slojevito, a najčešće korišten materijal je plastika. Objekti napravljeni na ovakav način mogu biti različitih veličina i boja.

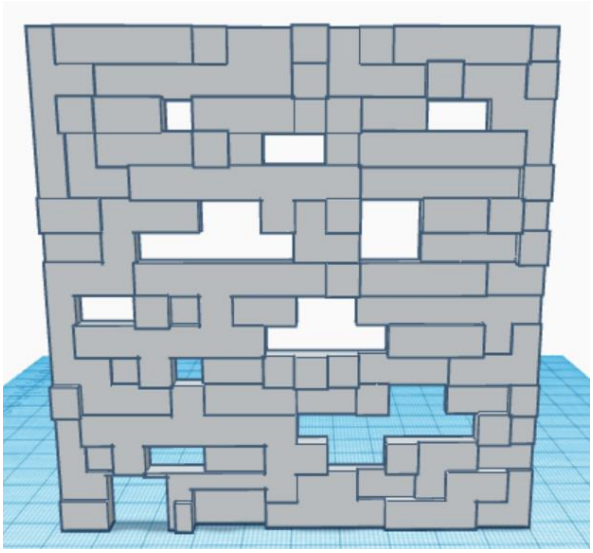
4.1. Dizajn

Neke su od stranica ove Minecraft lampe preuzete sa internetske stranice *thingiverse* [8] dok su neke, zbog preinaka koje je bilo potrebno napraviti, dizajnirane u aplikaciji *tinkercad* [9]. *Tinkercad* je besplatna web aplikacija za 3D dizajn, elektroniku i kodiranje. Koristi jednostavan način stvaranja modela korištenjem osnovnih geometrijskih oblika.



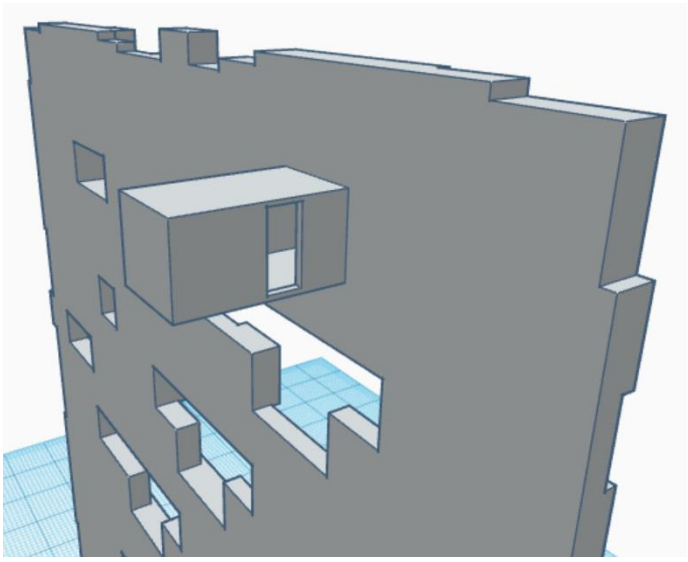
Slika 4.1. *Prikaz donje stranice lampe*

Prva ovakva dizajnirana stranica je stranica dna lampe (Slika 4.1.) čiji je ispis trajao 6 sati. Dimenzije su ove stranice 10,5 cm x 10,5 cm. Broj 1 označava otvor, odnosno prorez na rubu lampe kako bi se moglo napajati mikroupravljačku pločicu. Duguljasti valjak čija je unutrašnjost šuplja predstavlja stalak za bateriju, odnosno u njega se stavi baterija kako bi ona imala svoje mjesto i ne bih smetala ostalim komponentama. Na vanjsku stranu valjka dolazi LED traka koje se nalijepi oko njega. Broj 2 označava otvor na valjku kroz koji se provuče kabel od baterije kako bi se mogao spojiti na pločicu.



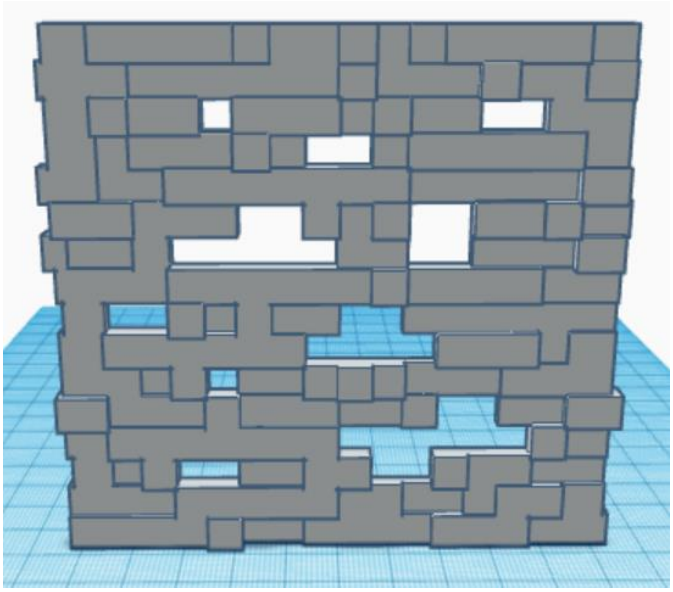
Slika 4.2. *Prikaz bočne stranice lampe sa otvorom za kabel za napajanje*

Na slici 4.2. prikazana je jedna bočna stranica lampe dimenzija 10,5 cm x 10,5 cm koja je modificirana tako da ima prorez u donjem lijevom kutu koji se poklapa sa prorezom na stranici dna kako bi se mogao spojiti kabel za napajanje na pločicu. Ispis ove stranice je trajao 4 sata i 19 minuta.



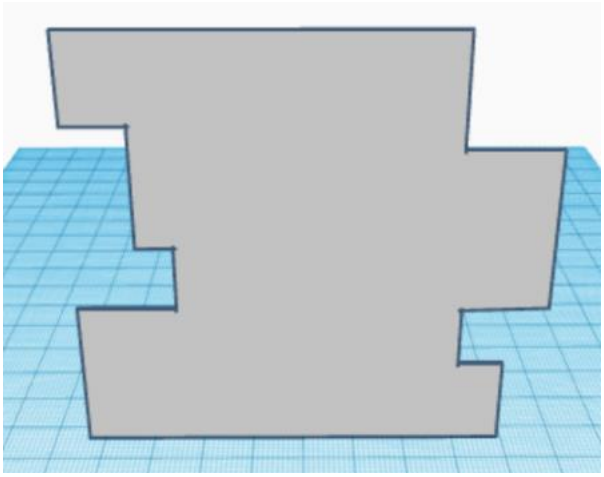
Slika 4.3. *Prikaz gornje stranice lampe sa kućištem za tipku*

Na slici 4.3. se nalazi dizajn gornje stranice lampe čiji je ispis trajao 5 sati. Na njoj se nalazi kućište za tipku. Napravljeno je tako da se tipka umetne te se njene nožice, tj. priključci stave u otvore sa bočnih strana kućišta. Na taj je način osigurano da je tipka učvršćena te da se može sigurno pritiskati bez da će doći do ikakvog pomjeranja sklopa unutar kutije.



Slika 4.4. *Bočna stranica*

Na slici 4.4. je prikazana bočna stranica koju treba ispisati tri puta. Ispis jedne ovakve stranice je trajao 4 sata i 39 minuta.

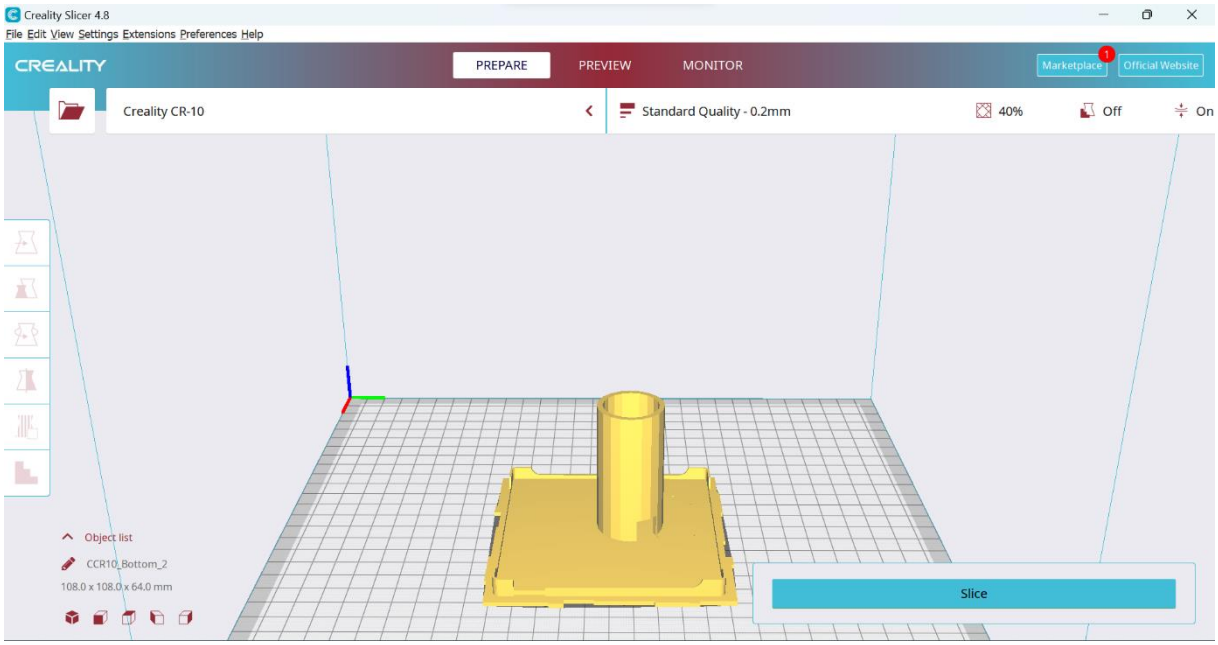


Slika 4.5. *Bijela površina za unutrašnjost lampe*

Na slici 4.5. se vidi još jedan dio potreban za ovu lampu, a to je bijela površina koja se mora staviti na unutarnje stranice lampe. Bilo je potrebno ovakvih pet primjeraka za svaku stranicu lampe. Njihov je ispis trajao par minuta.

4.2. Ispis

Kada su stranice dizajnirane potrebno ih je spremiti u *.stl* datoteku kako bi se mogle ispisati. Kako bi se one mogle ispisati potrebno je imati na računalu aplikaciju u kojoj se one mogu otvoriti. Ovdje je korištena aplikacija *Creality Slicer* koja je namijenjena za *Creality CR-10* 3D printer. Kada se otvori aplikacija potrebno je u njoj otvoriti *.stl* datoteku koja se želi ispisati.



Slika 4.6. *Prikaz donje stranice u aplikaciji Creality Slicer*

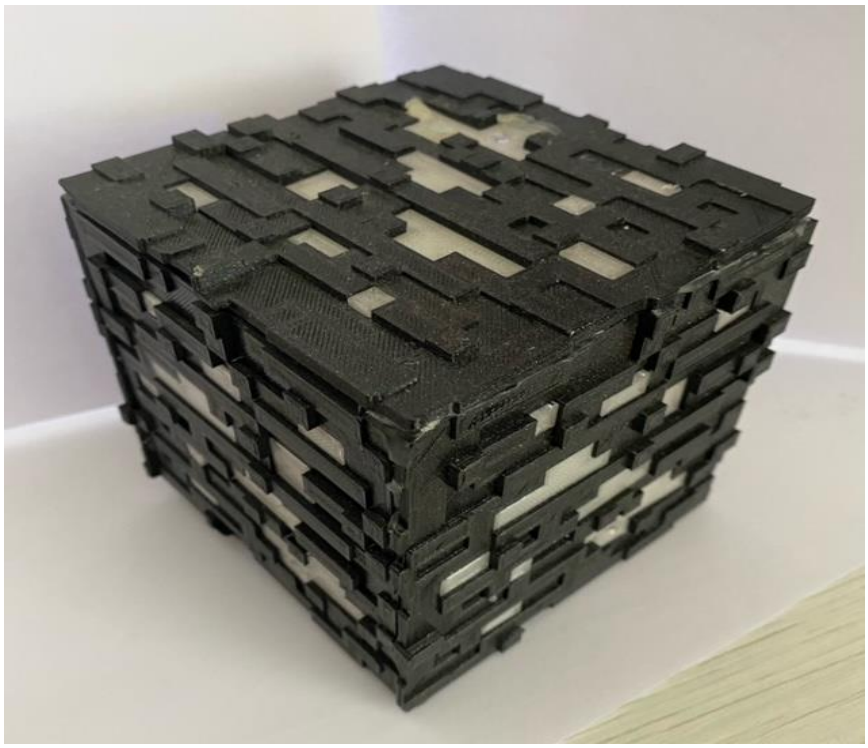
Kada je otvorena *.stl* datoteku kao što je prikazano na slici 4.6. potrebno je namjestiti postavke ispisa kao što su ispuna (engl. *infill*) te želimo li prijanjajuću podlogu (engl. *adhesion*). Prijanjajuća podloga je u obliku ravne površine koja se printa ispod ili okolo objekta kako bi se objekt koji se printa dodatno učvrstio. Vrlo je lagana i tanka tako da se nakon ispisa može odrezati.

Slika 4.7. *Prikaz postavki za ispis u aplikaciji Creality Slicer*

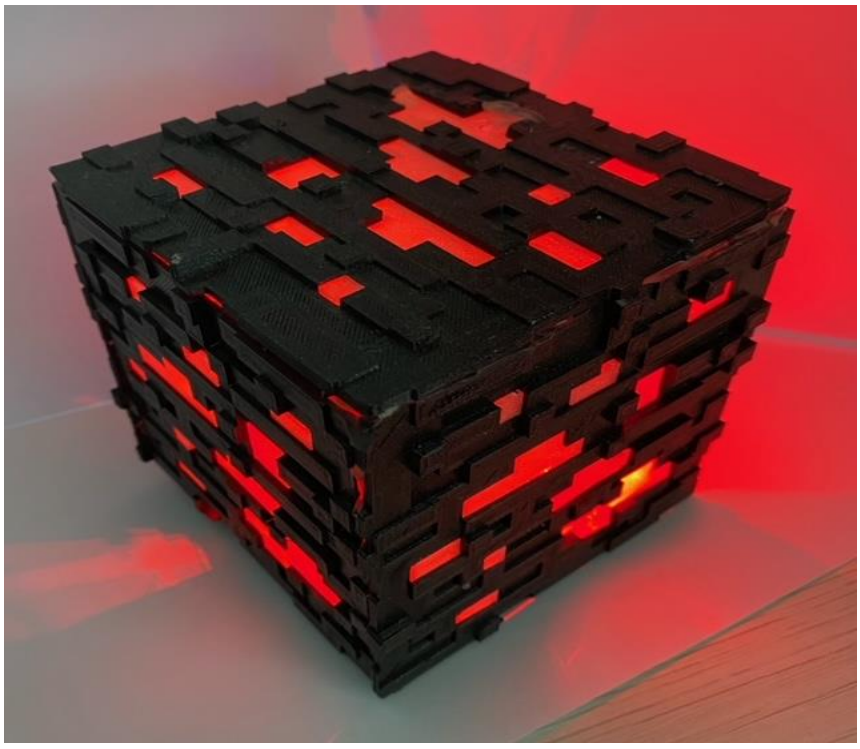
Na slici 4.7. su prikazane postavke za ispis ovog rada u kojima se vidi da je ispuna podešena na 40% te da je uključena prijanjajuća podloga. Nakon ovog koraka potrebno je kliknuti opciju *Slice* koja procijeni vrijeme ispisa te je rad spreman za ispis.

4.3. Sastavljanje stranica

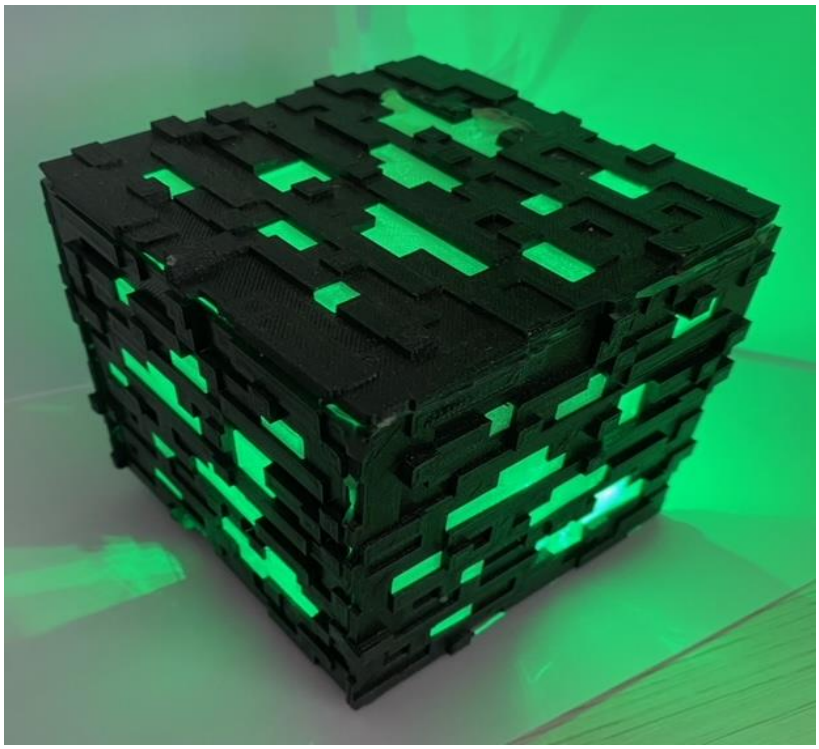
Nakon ispisa svih potrebnih stranica lampa je sastavljena pomoću ljepila i magneta. Sklop koji upravlja lampom je stavljen unutar lampe te su bočne stranice spojene ljepilom. Na gornjoj stranici su stavljeni magneti u kutove kako bi ju se kasnije moglo maknuti u slučaju da se baterija mora promijeniti. Na slikama 4.8., 4.9., 4.10., 4.11., 4.12. i 4.13. je vidljivo kako lampa izgleda kada se sve sastavi i kada je spremna za rad. Slike 4.14. i 4.15. prikazuju unutrašnjost lampe.



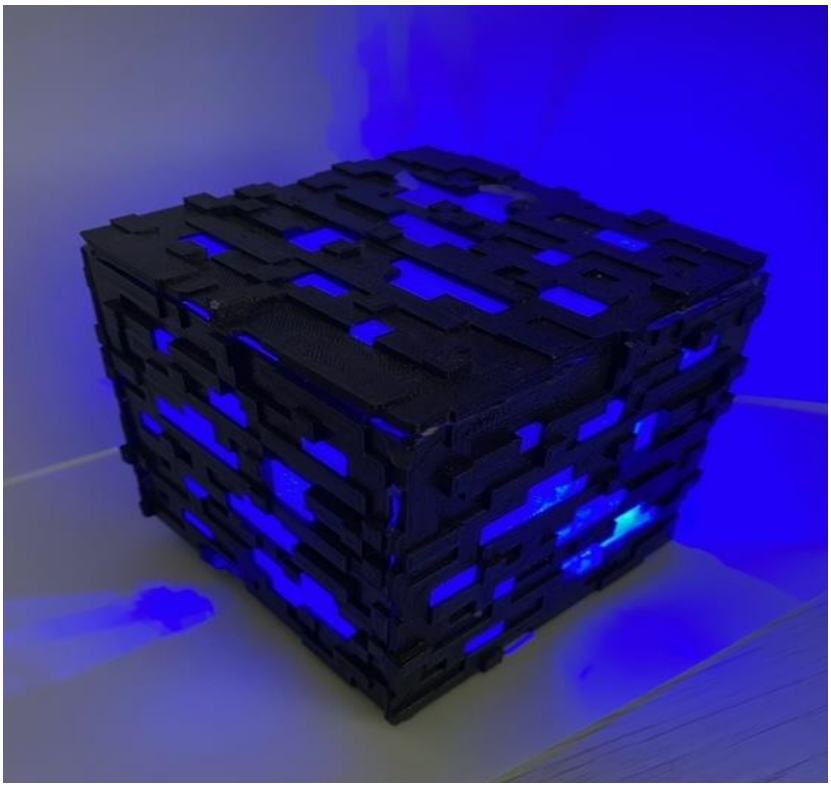
Slika 4.8. *Prikaz lampe kada je ugašena*



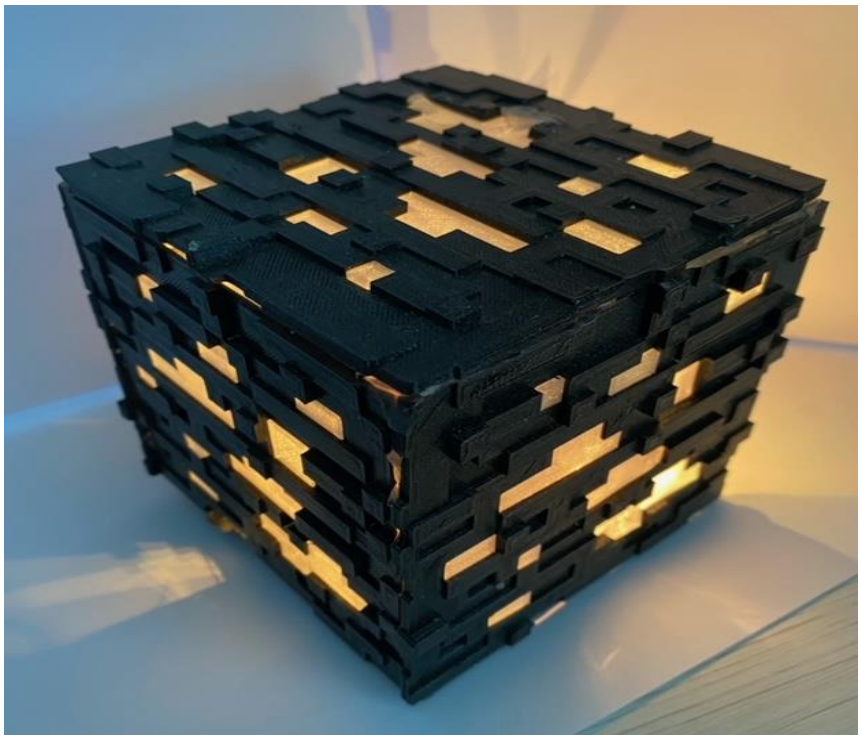
Slika 4.9. *Boja Redstone*



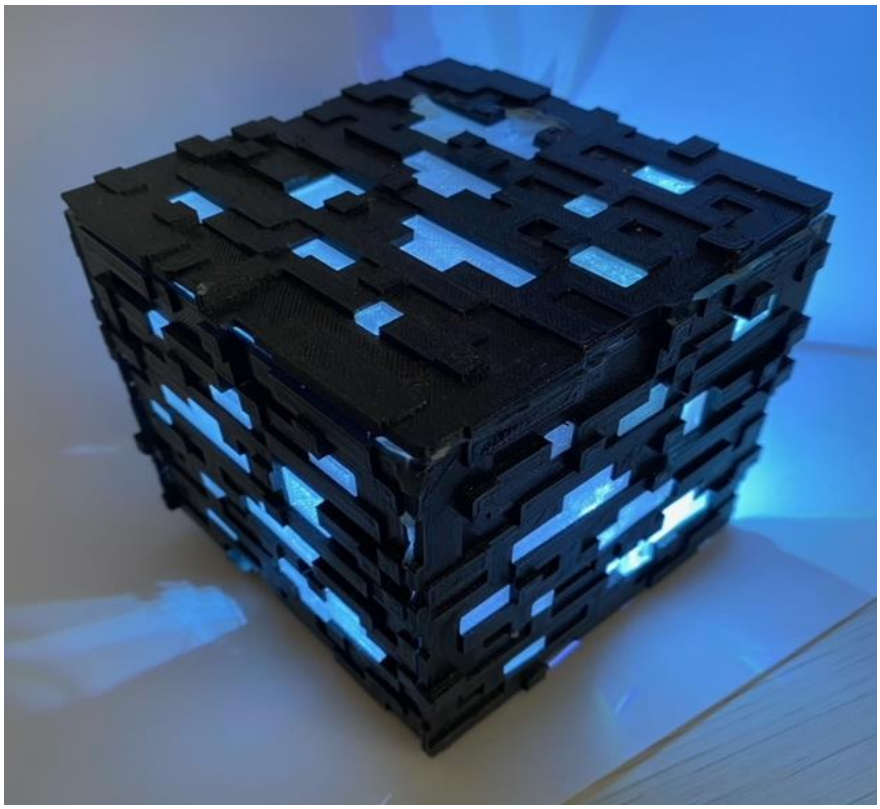
Slika 4.10. *Boja Emerald*



Slika 4.11. *Boja Lapis Lazuli*

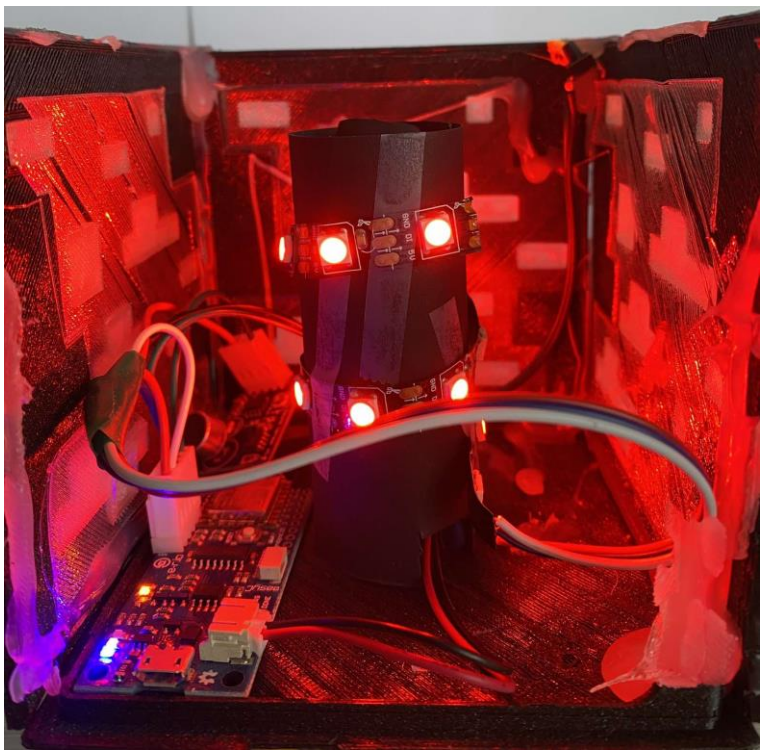


Slika 4.12. *Boja Gold*



Slika 4.13. *Boja Diamond*

Slika 4.14. *Unutrašnjost lampe*



Slika 4.15. *Unutrašnjost lampe kada svijetli u Redstone boji*

5. ZAKLJUČAK

U ovom je radu napravljena Minecraft lampa prema poznatoj videoigri. Prednost je ove lampe što u sebi sadrži bateriju pa tako ona postaje lako prenosiva i bežična. Uz njenu laku prenosivost, velika je prednost i što se njome može upravljati bez fizičkog dodira. Zbog potenciometra koji sadrži, lampu je moguće upaliti i pljeskom ruku ili bilo kakvim drugim zvukom. U radu je prvo objašnjen proces stvaranja sklopa zbog kojeg lampa svijetli, a zatim proces 3D ispisa koji će tu lampu fizički napraviti. Napravljena je od jednostavnih komponenti zbog čega je njena proizvoljna vrlo povoljna. Smatram da se ostvario postavljeni cilj ovog rada jer je programski kod vrlo razumljiv te ostavlja korisnicima prostora za nadogradnju prema osobnim željama.

LITERATURA

[1] „Minecraft“, dostupno na: https://en.wikipedia.org/wiki/Minecraft (pristupljeno 31. kolovoza 2023.)

[2] „ESP8266“, dostupno na: https://en.wikipedia.org/wiki/ESP8266 (pristupljeno 20. srpnja 2023.)

[3] „Croduino Nova 2“, dostupno na: https://advancedgwt.wixsite.com/klub/product-page/croduino-nova2 (pristupljeno 24. kolovoza 2023.)

[4] „What is Arduino?“, dostupno na: https://www.arduino.cc/en/Guide/Introduction (pristupljeno 20. srpnja 2023.)

[5] „Light-emitting diode“, dostupno na: https://en.wikipedia.org/wiki/Light-emitting_diode (pristupljeno 20. srpnja 2023.)

[6] „Potentiometer: Definition, Types, And Working Principle“, dostupno na: https://www.electrical4u.com/potentiometer/ (pristupljeno 16. kolovoza 2023.)

[7] „What Are Lithium-Ion Batteries? | UL Research Institutes“, dostupno na: https://ul.org/research/electrochemical-safety/getting-started-electrochemical-safety/what-are-lithium-ion (pristupljeno 16. kolovoza 2023.)

[8] „thingiverse“, dostupno na: https://www.thingiverse.com/ (pristupljeno 16. studenoga 2022.)

[9] „tinkercad“, dostupno na: https://www.tinkercad.com/ (pristupljeno 14. svibnja 2023.)

SAŽETAK

U ovom je završnom radu napravljena Minecraft lampa koja mijenja boju svijetla na pritisak tipke ili kada detektira zvuk. Mijenjanje boje detekcijom zvuka je omogućeno zbog potenciometra koji je dio sklopa. Lampa je upravljana Croduino Nova 2 razvojnom pločicom na kojoj se nalazi ESP8266 mikroupravljač. Sadrži i bateriju tako da ne mora skroz biti spojena na napajanje te je lako prenosiva. Korištena je WS2812b LED traka. Programski kod za ovu lampu napisan je u Arduino IDE okruženju. Stranice za lampu su prvo dizajnirane, a potom ispisane pomoću 3D printera. Nakon njihovog ispisa spojene su ljepilom te je sklop umetnut unutar kućišta.

Ključne riječi: 3D printer, Arduino, Croduino Nova 2, ESP8266, Minecraft

MINECRAFT LAMP

ABSTRACT

In this final work, a Minecraft lamp was made that changes the color of the light at the press of a button or when it detects a sound. Changing the color by sound detection is possible due to the potentiometer that is part of the circuit. The lamp is controlled by a Croduino Nova 2 development board with an ESP8266 microcontroller. It also contains a battery so that it does not have to be completely connected to the power supply and is easily portable. WS2812b LED strip was used. The programming code for this lamp is written in the Arduino IDE environment. The pages for the lamp are first designed and then printed using a 3D printer. After their printing, they are joined with glue and the circuit is inserted inside the case.

Keywords: 3D printer, Arduino, Croduino Nova 2, ESP8266, Minecraft