Razvoj sustava za udaljenu kontrolu time-lapse fotografije

Pitlović, Filip

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

Sveučilišni studij

RAZVOJ SUSTAVA ZA UDALJENU KONTROLU TIME-LAPSE FOTOGRAFIJE

Diplomski rad

Filip Pitlović

Osijek, 2024.

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

Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit

Osijek, 09.04.2024.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za diplomski ispit

Ime i prezime Pristupnika:	Filip Pitlović
Studij, smjer:	Diplomski sveučilišni studij Računarstvo
Mat. br. Pristupnika, godina upisa:	D-1240R, 07.10.2021.
OIB studenta:	86077136335
Mentor:	prof. dr. sc. Irena Galić
Sumentor:	Marin Benčević, mag. ing. comp.
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	doc. dr. sc. Krešimir Romić
Član Povjerenstva 1:	Marin Benčević, mag. ing. comp.
Član Povjerenstva 2:	dr. sc. Marija Habijan
Naslov diplomskog rada:	Razvoj sustava za udaljenu kontrolu time-lapse fotografije
Znanstvena grana diplomskog rada:	**Informacijski sustavi (zn. polje računarstvo)**
Zadatak diplomskog rada:	Istražiti i opisati komunikacijsko sučelje fotoaparata sa stajališta komponenti i programske podrške. Opisati mogućnosti kontrole funkcija kamere koje komunikacijsko sučelje nudi. Uz pomoć programabilnog mikrokontrolera i dodatnih komponenti razviti sustav udaljenog intervalometra za fotoaparat koji se kontrolira web aplikacijom i WebSocket protokolom. Sustav mora imati mogućnost okidanja slika zadanom ekspozicijom u zadanom intervalu. Omogućiti dinamičan interval koji se može s vremenom povećavati ili smanjivati. Detaljno opisati razvijeni sustav, web sučelje i programski kod. Opisati prednosti i nedostatke te usporediti s postojećim rješenjima. Sumentor: Marin Benčević
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	09.04.2024.

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, 20.05.2024.

Ime i prezime studenta:	Filip Pitlović
Studij:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D-1240R, 07.10.2021.
Turnitin podudaranje [%]:	6

Ovom izjavom izjavljujem da je rad pod nazivom**: Razvoj sustava za udaljenu kontrolu time-lapse fotografije**

izrađen pod vodstvom mentora prof. dr. sc. Irena Galić

i sumentora Marin Benčević, 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 studenta:

SADRŽAJ

1. UVOD

U suvremenom digitalnom dobu, fotografija je postala neizostavan dio našeg svakodnevnog života i široko korištena forma izražavanja i dokumentiranja različitih aspekata života. U eri digitalne tehnologije, fotografija nije samo zabilježavanje trenutka, već i sredstvo izražavanja, komunikacije i umjetnosti. U tom kontekstu, *time-lapse* fotografija, tehnika snimanja koja bilježi promjene kroz vrijeme i pretvara ih u dinamične sekvence, postaje sve popularnija među fotografskim entuzijastima i profesionalcima.

Time-lapse fotografiranje predstavlja iznimno učinkovit način stvaranja videozapisa pomoću slika. Poput videa koji se sastoji od uzastopnih sličica, *time-lapse* je tehnika koja omogućuje stvaranje impresivnih videozapisa od niza fotografija. Ova tehnika omogućuje snimanje dugotrajnih procesa ili prirodnih fenomena poput izlaska i zalaska sunca, kretanja oblaka ili promjena u prirodi, prikazujući ih na ubrzan način koji otkriva skrivene detalje i ljepotu vremenskih promjena.

U području *time-lapse* fotografije, važno je imati mogućnost daljinskog upravljanja fotoaparatom kako bi se osigurala precizna kontrola nad procesom snimanja. Razvoj sustava za daljinsko okidanje fotografija omogućuje fotografima i istraživačima fleksibilnost u postavljanju i izvođenju *time-lapse* projekata, bez potrebe za neprekidnim nadzorom nad fotoaparatom.

Ovaj rad istražuje razvoj sustava za daljinsko okidanje fotografija za *time-lapse* projekte. Cilj je razviti pouzdan, prilagodljiv i jednostavan za korištenje sustav koji omogućuje fotografima da precizno kontroliraju parametre snimanja, kao što su interval između snimaka, trajanje snimanja, promjena intervala tokom snimanja te postavke ekspozicije. Kroz analizu postojećih tehnologija te dizajniranje i implementaciju potrebnih komponenti, ovaj rad će se fokusirati na razvoj novog uređaja za daljinsko okidanje fotografija za *time-lapse* projekte. Cilj je stvoriti napredno rješenje koje će biti poboljšano u odnosu na već postojeća rješenja na tržištu. Nakon implementacije, sustav je testiran kako bi se osigurala funkcionalnost i pouzdanost u različitim scenarijima korištenja.

1.1. Zadatak diplomskog rada

Glavni cilj ovog diplomskog rada razvoj je sustava kamere koristeći mikrokontroler ESP32, te implementacija odgovarajućeg softverskog sučelja koje će omogućiti integraciju s DSLR kamerom. Kroz primjenu algoritama prilagođenih unaprijed definiranim parametrima, cilj je generirati fotografije potrebne za *time-lapse* snimanje. Ključni fokus bit će omogućiti dinamično prilagođavanje intervala između pojedinih snimaka tijekom procesa snimanja. Dodatno, implementirat će se funkcionalnost daljinskog upravljanja mikrokontrolerom putem web sučelja i *WebSocket* komunikacije, što će olakšati upotrebu sustava i omogućiti korisniku fleksibilnu kontrolu intervalometra, odnosno samog uređaja tijekom snimanja *time-lapsea*. Ovaj rad prikazuje primjenu znanja stečenog na fakultetu vezano uz predmete elektrotehnike za razvoj same pcb pločice i cijelog hardverskog dijela, pa sve do razvoja samog softvera i primjenu znanja vezanog za to područje.

Ovaj projekt prikazuje razvoj intervalne kamere koja se koristi za snimanje *time-lapse* fotografije od samog početka, hardvera, pa do razvijanja softverskog sučelja, te na samom kraju do razvoja web korisničkog sučelja koji omogućuje što jednostavniju interakciju između korisnika i uređaja. Fokus u izradi ovog diplomskog rada odnosi se na rješavanje problema promjene intervala tijekom samog snimanja *time-lapse* fotografije, bez potrebe za njegovim zaustavljanjem. Upravo iz tog problema proizišla je zamisao o pisanju ovog diplomskog rada.

2. ANALIZA PODRUČJA TEMATIKE I ODABIR ALATA

Diplomski rad temelji se na razvoju tri dijela: izradi hardvera, razvoju softvera i implementaciji web korisničkog sučelja.

Projekt se temelji na stvaranju intervalometra, uređaja ključnog za snimanje *time-lapse* fotografija. Nakon određenog broja snimljenih slika, stvara se konačan videozapis kao niz malih sličica. Glavna obilježja intervalometra obuhvaćaju unos vremenskog intervala i odabir broja fotografija za snimanje. Nakon postavljanja parametara, snimanje se pokreće, pri čemu se fotografije okidaju u zadanim vremenskim razmacima sve dok se ne dosegne unaprijed određeni broj željenih okinutih fotografija. Jedna od inovativnih značajki ovog projekta je *time ramping,* sposobnost mijenjanja vremenskog intervala tijekom snimanja fotografija. Ova funkcionalnost omogućuje korisniku neometano snimanje, uz mogućnost prilagodbe intervala prema promjenama svjetla ili drugim uvjetima snimanja, kao što je prijelaz iz dana u noć.

U izradi hardverskog dijela rješenja koristi se LilyGo razvojna pločica koja sadrži ESP32 mikrokontroler s ugrađenim TTGO zaslonom [1].
LilyGo razvojna pločica pouzdan je i kompaktan izbor za izradu projekata temeljenih na ESP32 mikrokontroleru, koji nudi širok spektar mogućnosti za razvoj različitih IoT projekata. Karakterizira ga visoka izvedba, niska potrošnja energije te podrška za bežičnu komunikaciju poput Wi-Fi-ja i Bluetootha [2]. Uz ugrađeni TTGO zaslon, olakšava razvoj korisničkih sučelja i prikazivanje informacija na ekranu.

Za izradu softverskog dijela, korišten je C jezik unutar Arduino Integrated Development Environment (IDE), koji omogućuje jednostavno programiranje i upravljanje ESP32 mikrokontrolerom. Ova kombinacija pruža fleksibilnost i efikasnost u razvoju različitih funkcionalnosti i algoritama potrebnih za kontrolu kamere i generiranje *time-lapse* fotografija.

Za komunikaciju između web sučelja i mikrokontrolera koristi se *WebSocket* protokol, koji omogućuje stvaranje stalne dvosmjerne veze između klijenta i poslužitelja putem *HTTP*-a. Ovaj protokol omogućuje brzu i učinkovitu razmjenu podataka u stvarnom vremenu između web aplikacije i uređaja temeljenih na ESP32 mikrokontroleru [3].

Proces uspostavljanja WebSocket veze započinje klijentskim slanjem posebnog HTTP zahtjeva prema poslužitelju, nazvanog "WebSocket handshake". Nakon što poslužitelj odobri zahtjev, uspostavlja se stalna veza između klijenta i poslužitelja.

Nakon uspostave veze, klijent i poslužitelj mogu međusobno slati podatke u realnom vremenu, bez potrebe za ponovnim uspostavljanjem veze za svaku razmjenu podataka. Ova stalna veza omogućuje bržu i učinkovitiju komunikaciju, što je idealno za aplikacije poput *chatova*, igara u stvarnom vremenu i nadzornih sustava.

3. ANALIZA POSTOJEĆIH RJEŠENJA

Na tržištu postoje uređaji namijenjeni za okidanje *time-lapse* fotografije, no većina ih ima određene nedostatke koji ograničavaju njihovu učinkovitost i funkcionalnost. U nastavku su navedeni samo neki od njih.

3.1. Žični daljinski okidač

Daljinski okidač je uređaj koji se koristi za aktiviranje fotoaparata na daljinu, omogućavajući snimanje s udaljenosti. Koristi se kada je pritiskanje gumba za okidanje fotoaparata ili korištenje ugrađenog brojača vremena nemoguće, nepraktično ili jednostavno nezgodno. Pod daljinskim okidačem mislimo na elektroničke uređaje koji rade s trenutnim digitalnim i filmskim fotoaparatima [4].

Canon TC-80N3, prikazan na slici 3.1., primjer je klasičnog žičanog daljinskog okidača. Ovaj uređaj omogućuje fotografima udaljeno okidanje fotoaparata bez potrebe za direktnim dodirivanjem uređaja [5]. Uređaj nudi funkcionalnosti brojača vremena, satova ili intervalometra, uparen s LCD zaslonom za postavljanje. Ovaj daljinski upravljač se gotovo isključivo koristi za stvaranje sekvenci *time-lapse* fotografije. Napredne funkcije ovog tipa daljinskih upravljača zahtijevaju baterije, jer kamere obično ne opskrbljuju energijom putem svojih priključaka za okidanje.

Slika 3.1. Canon TC-80N3 Timer Remote Controller

Nedostatak ovog uređaja leži u ograničenom broju funkcija. To znači da ima manje mogućnosti prilagodbe ili nedostatak dodatnih funkcija za snimanje *time-lapse* fotografije.

3.2. Infracrveni daljinski okidač

Najjednostavniji od svih su infracrveni daljinski upravljači za fotoaparate. Namijenjeni su u radu s ugrađenim IR prijemnikom fotoaparata [6]. Primjer takvog uređaja je jednostavni jednogumbni Nikon ML-L3 IR daljinski upravljač, prikazan na slici 3.2.

Slika 3.2. Nikon ML-L3 Infrared Remote Control

Infracrveni daljinski upravljači imaju određene nedostatke. Nemaju svi aparati ugrađenu podršku za infracrvene daljinske upravljače. Ako razmatrate infracrveni daljinski upravljač, prvo provjerite specifikacije svog fotoaparata. Još jedna stvar na koju treba obratiti pažnju jest da daljinski upravljači poznatih marki obično rade samo s fotoaparatima iste marke. Kao i daljinski upravljači za televizor, postoje i proizvodi koji podržavaju nekoliko marki.

Budući da se infracrveni daljinski upravljači oslanjaju na mogućnost IR senzora fotoaparata da vidi signal daljinskog upravljača, imaju ograničeni domet i obično zahtijevaju izravni vizualni kontakt između daljinskog upravljača i prednjeg dijela fotoaparata.

3.3. MIOPS Smart +

MIOPS Smart+ nudi različite mogućnosti kontrole okidanja, uključujući intervalometar za *time-lapse* fotografiju, kao i mogućnost detekcije pokreta, zvuka ili svjetlosti za automatsko okidanje. Osim toga, ima i dodatne funkcije poput *long exposure*, HDR i brzinske fotografije [7], slika 3.3.

Slika 3.3. MIOPS Smart +

MIOPS Smart+ također omogućuje bežično upravljanje putem pametnog telefona putem aplikacije. Uz to, MIOPS Smart+ ima kompaktan i izdržljiv dizajn, što ga čini praktičnim alatom za fotografe koji često putuju ili snimaju na terenu.
Nedostatak MIOPS Smart+ može biti njegova relativno visoka cijena u usporedbi s drugim rješenjima na tržištu. Također, neki korisnici mogu smatrati da je sučelje aplikacije komplicirano za korištenje ili da aplikacija nije uvijek potpuno stabilna.

3.4. LRTimelapse Pro

Možda i najsličnije rješenje onomu o čemu se govori i u ovom projektu je LRTimelapse Pro, prikazan na slici 3.4. LRTimelapse Pro je intervalometar za snimanje *time-lapse* fotografije s ugrađenim funkcijama koje predstavljaju nedostatke u prijašnje navedenim uređajima.

Slika 3.4. LRTimelapse Pro

LRTimelapse Pro nudi nekoliko značajki koje ga čine velikom inovacijom u svijetu intervalometara [8].
Uređaj pruža opciju automatskog izoštravanja prije svake snimke, osiguravajući jasne i oštre fotografije u *time-lapseu* te omogućuje kontrolu izloženosti kroz različite parametre kako bi se postigla željena svjetlina i kontrast u fotografijama. Podržava različite formate slika i videozapisa, omogućujući jednostavno uvoz i izvoz sadržaja.

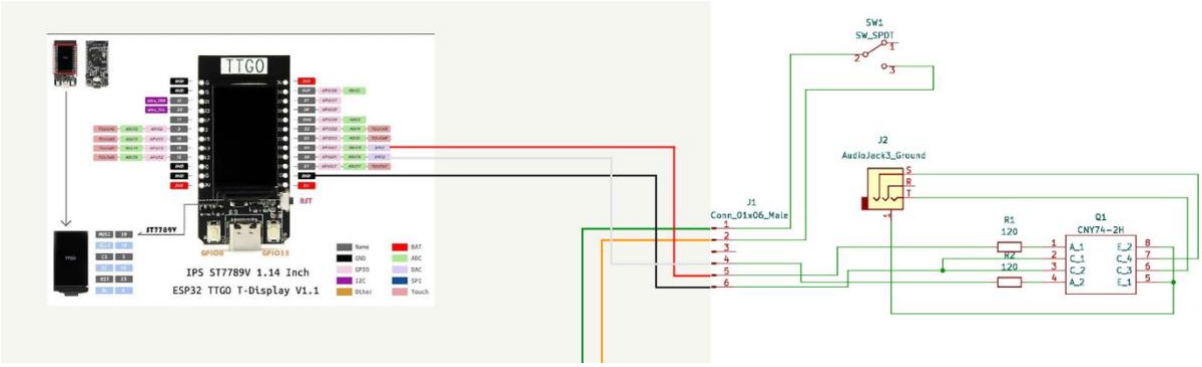
Dakako, najsigurniji način za korištenje je putem kabla. Veza i komunikacija s fotoaparatom putem kabla nudi točnost i sigurnost, dok uređaji povezani putem *Bluetooth* odnosno *Wifi* veze nisu uvijek pouzdani. Nedostatak ovog uređaja je masivnost te ne baš tako lako korištenje.

Nakon analize već postojećih rješenja zaključeno je kako postoji nekolicina problema koje otežavaju rad za snimanje *time-lapse* fotografije samom korisniku. Primjerice, nemogućnost promjene samog intervala prilikom snimanja, nesigurnost korištenja infracrvene veze, masivnost, nedovoljan broj funkcija i mnogi drugi.
U ovom projektu se rješavaju upravo ti nedostatci a fokus se odnosi na mogućnost promjene intervala *time-lapse* fotografije prilikom samog snimanja. Ta opcija predstavlja veliku prekretnicu u samom svijetu fotografije.

4. HARDVERSKO RJEŠENJE

4.1. Shematski prikaz

Za izradu hardvera, prvo je potrebno izraditi shematski prikaz prije prikupljanja potrebnih komponenti, slika 4.1. Shematski prikaz izrađen je pomoću programa KiCad. KiCad je besplatni, open-source softver za dizajniranje elektroničkih sklopova i štampanih pločica (PCB). Koristi se za stvaranje shematskih dijagrama, razvoj PCB dizajna, praćenje staze i postavljanje komponenti na pločici, generiranje proizvodnih datoteka za izradu štampanih pločica te simuliranje i analizu električnih sklopova [9].



Slika 4.1. Shematski prikaz

Na shemi je prikazano kako je za izradu potreban TTGO zaslon koji se pomoću konektora spaja na PCB pločicu. PCB pločica sadrži *audio jack* ulaz i optokupler, koji su detaljnije objašnjeni u nastavku.

4.2. Komponente

Potrebno je izraditi sklop elektroničkih komponenti kako bi se uređaj mogao spojiti s DSLR fotoaparatom.

4.2.1. LilyGo T-display

LilyGO TTGO T-Display je razvojna ploča koja sadrži mikrokontroler ESP32 i ugrađeni TFT zaslon osjetljiv na dodir, prikazana na slici 4.2. Sastoji se od ESP32 mikrokontrolera, TFT zaslona visoke razlučivosti osjetljivog na dodir, kao i različitih perifernih komponenti poput senzora, uređaja za povezivanje i drugih mogućih dodataka [1].

Slika 4.2. LilyGo TTGO Display razvojna ploča

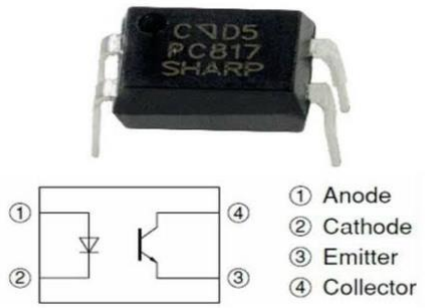
Spomenuti mikrokontroler dolazi s integriranim WiFi i bluetooth modulima, što omogućuje jednostavnu bežičnu komunikaciju s drugim uređajima i mrežama.

Programiranjem ESP32 mikrokontrolera omogućuje se kontrola raznih elektroničkih komponenti i uređaja putem programa napisanog u različitim programskim jezicima poput C/C++ ili *MicroPython*. ESP32 ima ugrađen set funkcija i perifernih uređaja koji se mogu kontrolirati putem programskog koda, uključujući digitalne i analogni ulazno/izlazne pinove, WiFi i bluetooth module, senzore, LCD ekrane, motorne vozače i još mnogo toga.

4.2.2. Optokupler

Za izolaciju digitalnih ulaza i izlaza mikrokontrolera i drugih dijelova hardvera koristi se optokupler, prikazan na slici 4.3.

Optokupler je elektronička komponenta koja se koristi za izolaciju električnih krugova. Ima dva dijela: LED (svjetlosni emiter) i fotootpornik (fotodioda ili kolektor). Kada se na LED primijeni električna struja, emitira se svjetlost koja aktivira fotootpornik. Fotootpornik tada proizvodi struju ili napon koji se koristi za kontrolu struje u drugom krugu. Optokupler se koristi za izolaciju visokih naponskih krugova od niskih naponskih krugova radi zaštite korisnika ili uređaja od visokih napona te prijenos signala između električkih krugova koji imaju različite potencijale zemlje, što omogućuje galvansku izolaciju [10].

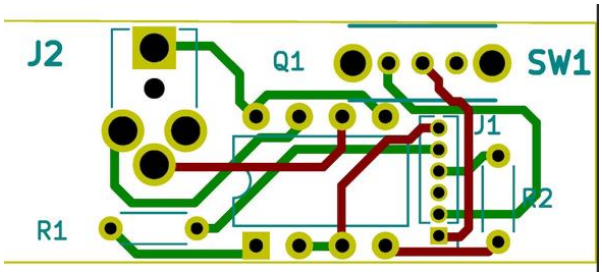


Slika 4.3. Optokupler

Anoda se odnosi na pozitivan (+) pol LED diode, dok se katoda odnosi na negativan (-) pol LED diode. Kod većine optokuplera, anoda je povezana s LED diodom, dok je katoda povezana s zemljom. Na strani prijemnika, fotodetektor ima svoje karakteristike, uključujući emiter i kolektor, koji su dio internog sklopa fotodetektora.

4.2.3. PCB pločica

PCB pločica *(Printed Circuit Board)* je temeljni element elektroničkog sklopovlja na kojem se fizički povezuju elektroničke komponente. Za ovaj projekt izrađena je PCB pločica prema modelu na slici 4.4.



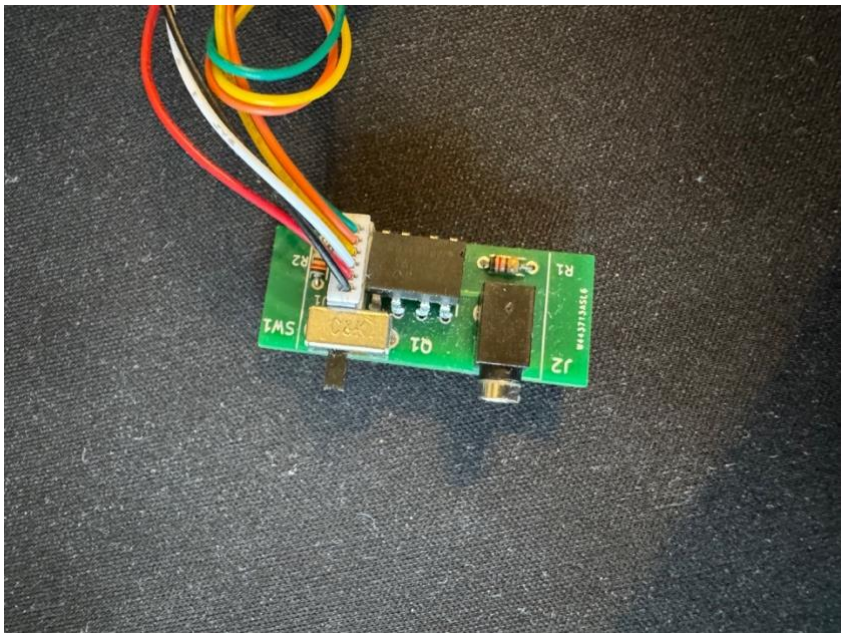
Slika 4.4. PCB pločica

4.2.4. Ostale komponente

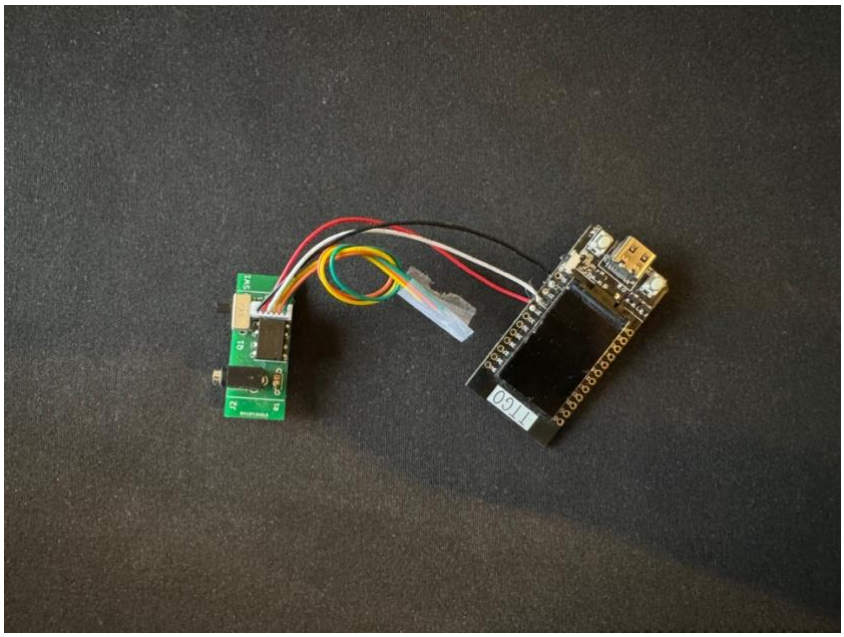
Ostale komponente potrebne za sklopovlje su:

- 2 otpornika od 120 Ohma
- Konektor koji se koristi za povezivanje komponenti i mikrokontrolera
- *Stereo* priključak za kabelsko povezivanje s DSLR fotoaparatom

Nakon procesa lemljenja i sastavljanja komponenti, konačni izgled intervalne kamere prikazan je na slikama 4.5. i 4.6.

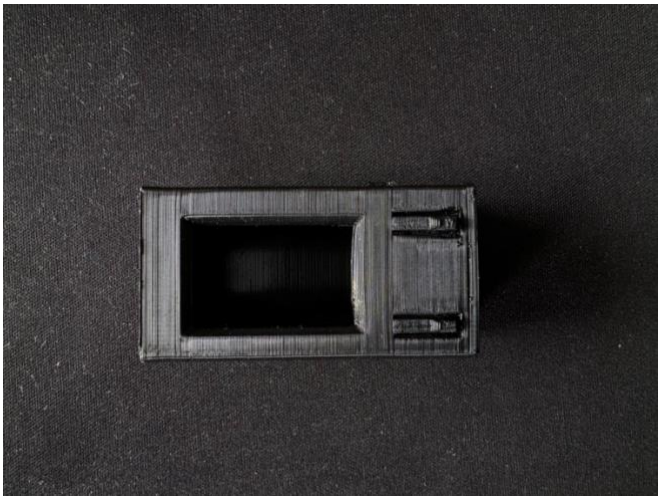


Slika 4.5. Pcb pločica s komponentama



Slika 4.6. Pcb pločica s komponentama i mikrokontroler

Kako bi korištenje intervalometra bilo jednostavnije, izrađeno je kućište pomoću 3D printera. Kučište je prikazano na slici 4.7. 3D model za printanje preuzet je s web stranice [11].



Slika 4.7. Kučište intervalne kamere

Samo kučište je malo te nije masivno što omogućuje jednostavno korištenje samog intervalometra u budućnosti.

5. PROGRAMSKO RJEŠENJE

5.1. Dodjela PIN-ova

Kako bi komunikacija bila uspješna između mikrokontrolera i sklopovlja potrebno je ispravno pridružiti pinove. Programski kod 5.1. Postavljanje pinova prikazuje kako pin 26 predstavlja okidač, dok pin 25 predstavlja fokus DSLR fotoaparata. Oba pina su DAC, *Digital to Analog Converter*, što znači da je sasvim svejedeno koji pin će predstavljati fokus, a koji okidač.
Pin koji kontrolira napajanje, ADC_EN, postavljen je na pin 14, dok je pin koji se koristi kao analogni ulaz postavljen na pin 34. Dva gumba koja su dio mikrokontrolera se postavljaju na pinove 0 i 35.

```
1. #define CAM_S 26
2. #define CAM_F 25
3. #define ADC_EN 14
4. #define ADC_PIN 34
5. #define BUTTON_1 0
6. #define BUTTON_2 35
```

Programski kod 5.1. Postavljanje pinova

5.2. Korištenje TTGO zaslona

TTGO ekrani popularni su zbog svoje visoke razlučivosti, dobrog prikaza boja i brzog odziva, što ih čini pogodnim za različite primjene poput IoT projekata, pametnih satova, mjerača i slično. TFT (*Thin-Film Transistor*) je vrsta tehnologije zaslona koja se koristi u TTGO zaslonima kako bi omogućila prikazivanje slika, teksta i grafike na ekranu.
Za korištenje TFT opcija potrebno je uključiti TFT_eSPI biblioteku [12]. Na slici 5.2. prikazan je početni zaslon prilikom pokretanja intervalometra.

Slika 5.2. Početni zaslon

5.3. Funkcionalnosti intervalometra

Kako je vidljivo na prethodnoj slici 5.2., početni zaslon nudi odabir korisniku želi li nastaviti rad i korištenje putem samog uređaja korištenjem gumbova ili pak spojiti se na web server i upravljati uređajem putem web-a. U nastavku je objašnjen rad svih funkcija i upravljanje ručno putem gumbova dok je rad pomoću web-a objašnjen u šestom poglavlju.

Pritiskom gornjeg gumba, otvara se *time-lapse* izbornik prikazan na slici 5.3. Prvi korak u pokretanju snimanja *time-lapse* fotografije je postavljanje intervala.

Slika 5.3 Postavljanje intervala

Korisnik ima mogućnost unosa intervala u minutama i sekundama. Interval predstavlja vrijeme koje će proći između okidanja fotografija. Programski kod 5.3. Postavljanje intervala sadrži implementaciju koda za prikaz podataka o postavljanju intervala na zaslonu. Postavljena je varijabla 'currentScreen' koja bilježi trenutni zaslon na kojem se nalazi korisnik i omogućuje komunikaciju unutar koda.

```
void setRampInterval()
{
  currentScreen=rampScreenInterval;
  intervalsScreen(rampIntervalMinutes,rampIntervalSeconds);
}

void intervalsScreen(int &x, int &y){
    tft.fillScreen(TFT_BLACK);
    tft.setCursor(20,5);
    if(currentScreen == rampScreenInterval){
        tft.print("Set ramp interval");
    }else if(currentScreen == intervalScreen){
        tft.print("Set interval");
    }
    tft.setTextSize(2);
    tft.setCursor(50,50);
    tft.print(String(x) + "m    ");
    tft.setCursor(90,50);
    tft.print(String(y) + "s    ");
    tft.setCursor(intervalCursorX,intervalCursorY);
    tft.print("-");
    tft.setCursor(160,110);
    tft.print("next>");
}
```

Programski kod 5.3. Postavljanje intervala

Programski kod 5.4. Povećanje i smanjenje intervala prikazan ispod, predstavlja implementaciju koda za povećanje i smanjenje vrijednosti intervala. Funkcionira na način da se konstantno unutar *'loop'* petlje provjerava je li gumb pritisnut. Dakle, ukoliko je pritisnut gornji gumb odnosno 'button2', povećava se vrijednost intervala u minutama ili sekundama za 1. Ukoliko je pritisnut donji gumb, smanjuje se interval po istom principu.

```
if(button2Flag==1){
    button2Flag=0;
    if(intervalCursorX==50){
      x++;
      if(flagFirstNumber==0){
        x=0;
        flagFirstNumber=1;
      }
      if(currentScreen == rampScreenInterval){
        setRampInterval();
      }else if(currentScreen == intervalScreen){
        setInterval();
      }
    }
    if(intervalCursorX==90){
      y++;
      if(currentScreen == rampScreenInterval){
        setRampInterval();
      }else if(currentScreen == intervalScreen){
        setInterval();
      }
    }
  }
```

Programski kod 5.4. Povećanje i smanjenje intervala

S obzirom na to da mikrokontroler ima samo dva gumba, razvijen je jednostavan način korištenja kako bi se olakšala upotreba uređaja.

Korisnik ima mogućnost 'kretanja' kroz izbornik, tako u primjeru postavljanja intervala, korisnik prvo postavlja broj minuta, zatim broj sekundi te nakon toga slijedi izbornik. Osmišljena je funkcija *setCursorPosition*(), prikazana u programskom kodu 5.5. Pomicanje kursora. Pomoću nje, postavlja se kursor u obliku strelice koji se nalazi ispod vrijednosti na kojoj se trenutno nalazi. Za pomicanje kursora koristi se dugi pritisak donjeg gumba. Ukoliko je gumb pritisnut duže od 0.5 sekundi očitava se dugi pritisak.

```
void setCursorPosition(){
   if(btn1LongClickFlag==1){
     btn1LongClickFlag=0;
     if(currentScreen==intervalScreen || currentScreen==rampScreenInterval){
       if(intervalCursorX == 140){
         intervalCursorX = 50;
         intervalCursorY = 70;
       }
       //set to menu
       else if(intervalCursorX==90){
         intervalCursorX = 140;
         intervalCursorY = 110;
       }else{
         intervalCursorX = intervalCursorX + 40;
       }
       if(currentScreen==intervalScreen){
         setInterval();
       }else if(currentScreen==rampScreenInterval){
         setRampInterval();
       }
     }
if (pressDuration > 500) {
     if(currentScreen == intervalScreen){
       btn1LongClickFlag=1;
       setInterval();
     }
}
```

Programski kod 5.5. Pomicanje kursora

Dakle, položaj kursora predstavlja ono što će se pokrenuti prilikom klika gumba 1 ili gumba 2 u tom trenutku.

Prilikom klika na opciju 'next' na zaslonu postavljanja intervala otvara se izbornik na slici 5.4.

Slika 5.4. Izbornik

Kako je vidljivo na slici, izbornik nudi mogućnosti unosa broja okidanja, ponovno postavljanje intervala, odlazak na glavni izbornik koji se pokreće prilikom pokretanja intervalometra te povratak nazad na postavljanje intervala, ali već unešene vrijednosti ostat će sačuvane. Za korištenje izbornika koriste se gumbovi gdje gornji gumb pokreće odabranu opciju dok donji gumb se koristi za kretanje kroz izbornik.

Prilikom klika na postavljanje broja okidanja otvara se zaslon za unos podataka prikazan na slici 5.5.

Slika 5.5. Unos broja okidanja

U programskom kodu 5.6. Unos broja okidanja prikazana je implementacija funkcije za unos broja okidanja. Ukoliko je broj okidanja postavljen na 0, broj okidanja bit će neograničen. Kako je već spomenuto, glavna funkcija za kretanje i postavljanje odabranog broja je *setCursorPosition*. Postavlja se vrijednost onog broja ispod kojeg se nalazi kursor. Dugi klik donjeg gumba omogućuje pomicanje kursora.

```
void setNumberOfShots()
{
   tft.fillScreen(TFT_BLACK);
   currentScreen=numberOfShotsScreen;
   tft.setCursor(20,5);
   tft.print("Set noShots");
   tft.setTextSize(2);
   tft.setCursor(50,50);
   tft.print(String(noShotsHundreds));
   tft.setCursor(70,50);
   tft.print(String(noShotsTens));
   tft.setCursor(90,50);
   tft.print(String(noShotsOnes));
   tft.setCursor(noShotX,noShotY);
   tft.print("-");
   tft.setCursor(160,110);
   tft.print("next>");

   //increase
   if(button2Flag==1){
     button2Flag=0;
     if(noShotX==50){
       noShotsHundreds++;
       if(flagFirstNumber==0){
         noShotsHundreds=0;
         flagFirstNumber=1;
       }
       setNumberOfShots();
     }
     if(noShotX==70){
       noShotsTens++;
       setNumberOfShots();
     }
     if(noShotX==90){
```

```
      noShotsOnes++;
      setNumberOfShots();
    }
   }
  }
 setCursorPosition();
}
```

Programski kod 5.6. Unos broja okidanja

Funkcija *setNumberOfShots* rekurzivna je funkcija, što znači kako prilikom povećavanja i smanjenja intervala poziva se ista funkcija pomoću koje se osvježava ekran.

Nakon što korisnik postavi broj okidanja, otvara se zaslon za konačni pregled postavljenih vrijednosti prije samog pokretanja *time-lapse*-a, kao što je prikazano na slici 5.6.

Slika 5.6. Pregled zadanih podataka

Kako se na početku unose minute i sekunde intervala i zatim broj okidanja potrebno je izračunati vrijeme odvijanja *time-lapse* fotografije. Kako je prikazano u Programskom kodu 5.7. Izračun vremena odvijanja *time-lapse* fotografije, potrebno je pomnožiti ukupan broj sekundi trajanja intervala, varijabla *finalSeconds*, s brojem unešenih okidanja. Prvi okidaj fotografije događa se odma na startu pa ja ukupno vrijeme okidanja smanjeno za vrijeme jednog intervala.

```
 intervalTimeSeconds = ((intervalMinutes*60)+(intervalSeconds));
   hoursInterval = ((intervalTimeSeconds * numberOfShots) - intervalTimeSeconds) / 3600;
   minutesForInterval = ((((intervalTimeSeconds * numberOfShots)-intervalTimeSeconds) -
(3600*hoursInterval))) / 60;
   finalSeconds = ((((intervalTimeSeconds * numberOfShots)-intervalTimeSeconds) -
(3600*hoursInterval)))  % 60;

   while(finalSeconds >= 60){
     finalSeconds = finalSeconds - 60;
     minutesForInterval++;
   }

   while(minutesForInterval >= 60){
     minutesForInterval = minutesForInterval - 60;
     hoursInterval++;
   }
   if(finalSeconds==0){
     minutesForInterval=minutesForInterval-1;
     finalSeconds=59;
   }
   hoursLeft=intervalHours;
   minutesLeft=intervalMinutes;
   secondsLeft=intervalSeconds;
```

Programski kod 5.7. Izračun vremena odvijanja time-lapse fotografije

Kada su svi parametri postavljeni prema željama korisnika, otvara se zaslon s kontinuiranim prikazom podataka, prikazan na slici 5.7. Podaci se osvježavaju svaku sekundu.

Slika 5.7. Zaslon koji prikazuje pokrenuti time-lapse

Na zaslonu za praćenje *time-lapse*-a, prikazuje se ukupno preostalo vrijeme do završetka procesa *time-lapse*-a, broj snimljenih fotografija, broj preostalih fotografija koje treba snimiti te vrijeme do snimanja slijedeće fotografije.

Funkcija *releaseCamera()* pokreće se kada vrijeme do snimanja slijedeće fotografije dobije vrijednost nula. Funkcija je prikazana u programskom kodu 5.8 Funkcija *releaseCamera*.

```
void releaseCamera(){
   digitalWrite(CAM_S, HIGH);
   digitalWrite(CAM_F, HIGH);
   delay(1000);
   digitalWrite(CAM_S, LOW);
   digitalWrite(CAM_F, LOW);
}
```

Programski kod 5.8. Funkcija releaseCamera

Priložena funkcija ima ulogu upravljanja kamerom radi postavljanja fokusa i okidanja fotografije. Nakon postavljanja fokusa i okidanja fotografije, dolazi do kratkog kašnjenja od jednu sekundu, nakon čega se postavke vraćaju na normalnu vrijedost. Ovo osigurava da se fotoaparat stabilizira i pripremi za sljedeći korak u procesu snimanja.

Kako je minimalni interval koji je moguće postaviti jednak jednoj sekundi, tako se i osvježavanje ekrana odvaja svaku sekundu. Programski kod za provjeru proteklog vremena od jedne sekunde nalazi se pod brojem 5.9. Ukoliko je prošlo više od jedne sekunde, brojač se ponovno pokreće, a zaslon se osvježava s novim vrijednostima podataka.

```
void startOneSecondTimer() {
  timerStartFlag = millis();
  timerFlag=0;
}
float isOneSecondElapsed() {
  return millis() - timerStartFlag;
}

void checkTimer(){
  if(timerStartFlag==0){
    startOneSecondTimer();
  }
  float tims = isOneSecondElapsed();
  if(tims >= 1000){
    timerFlag=1;
    startOneSecondTimer();
    runRunningMenu();
  }
}
```

Programski kod 5.9. Brojač sekunde

Funkcija koja se pokreće za vrijeme odvijanja *time-lapse* fotografije ima nekoliko uloga. Potrebno je osvježavati ekran svaku sekundu, postaviti brojač koji će brojati preostalo vrijeme odvijanja okidanja. Također, potrebno je računati broj okinutih fotografija što je napravljeno na način da se samo smanjuje varijabla koja sadrži broj unešenih okidanja. Potrebno je dakako dodati brojač koji će brojati broj okinutih fotografija. Sve to prikazano je u programskom kodu 5.10. Funkcija *runRunningScreen*.

```
void runRunninScreen(){
checkTimeUntilNextShot();

if(shotsDone == 0){
    shotsDone++;
```

```
    releaseCamera();
    numberOfShots--;
}
else{
 if(hoursInterval == 0 && minutesForInterval==0 && finalSeconds == 0 ){
   if(shotsCounter == intervalTimeSeconds ){
      secondsLeft=0;
      shotsCounter = 0;
      numberOfShots--;
      shotsDone++;
      releaseCamera();
   }
    if(currentScreen == runningScreen){
      updateRunningScreen();
    }
    runningFlag = 0;
}

else if(finalSeconds==0 && minutesForInterval != 0){
   if(minutesForInterval > 0){
      minutesForInterval--;
      finalSeconds = 59;
      if(shotsCounter == intervalTimeSeconds){
        shotsCounter = 0;
        numberOfShots--;
        shotsDone++;
        releaseCamera();
        if(changeTimeIndex==1){
          changeTimeIndex = 0;
          changeRunningTime();
          checkTimeUntilNextShot();
        }
      }
   }
```

```
    shotsCounter++;
    if(currentScreen == runningScreen){
      updateRunningScreen();
    }
}
  else{
     finalSeconds--;
      if(shotsCounter == intervalTimeSeconds){
        secondsLeft=0;
        shotsCounter = 0;
        numberOfShots--;
        shotsDone++;
        releaseCamera();
        if(changeTimeIndex==1){
          changeTimeIndex = 0;
          changeRunningTime();
          checkTimeUntilNextShot();
        }
      }
    shotsCounter++;
    if(currentScreen == runningScreen){
       updateRunningScreen();
    }
  }
 }
}
```

Programski kod 5.10. Funkcija runRunningScreen

Naposljetku, neizostavan dio je brojač koji govori korisniku koliko vremena je ostalo do okidanja slijedeće fotografije prikazan u programskom kodu 5.11 Funkcija *checkTimeUntilNextShot*.

```
void checkTimeUntilNextShot(){
 if(hoursLeft==0 && minutesLeft==0 && secondsLeft==0){
  hoursLeft=intervalHours;
  minutesLeft=intervalMinutes;
  secondsLeft=intervalSeconds;
 }
 if(secondsLeft==0){
  if(minutesLeft==0 && hoursLeft != 0){
   hoursLeft--;
   secondsLeft = 59;
   minutesLeft= 59;
  }
  else if(minutesLeft != 0){
   minutesLeft--;
   secondsLeft=59;
  }
 }
 else if(secondsLeft>0){
  secondsLeft--;
 }
}
```

Programski kod 5.11. Funkcija checkTimeUntilNextShot

Time ramping u kontekstu *time-lapse* fotografije označava postupno povećanje ili smanjenje vremenskih intervala između pojedinačnih fotografija tijekom snimanja *time-lapse-a*. Spomenuto omogućuje glatke prijelaze između scena i dodaje dinamiku i interes fotografijama.

Na primjer, ako je zadan interval između fotografija 2 sekunde, a želimo ga promijeniti na 5 sekundi, to možemo postići korištenjem *time rampinga*. U kontekstu *time-lapse* fotografije, kada koristimo *time ramping*, postavljamo novo vrijeme intervala između fotografija, na primjer, na pet sekundi, te određujemo vremenski period tijekom kojeg će se ta promjena postepeno dogoditi, na primjer, unutar jedne minute. To znači da će se interval postupno povećavati za jednu sekundu svakih dvadeset sekundi. Nakon jedne minute, interval za okidanje *time-lapse-a* bit će pet sekundi umjesto prvotnih dvije sekunde. Programski kod dan je pod brojem 5.12. Matematički izračun za postupno mijenjanje intervala.

```
void runRamping(){
   rampOn=1;
   rampTime=rampTime*60;
   rampIntervalTimeSeconds = (rampIntervalMinutes*60)+rampIntervalSeconds;
 if(intervalTimeSeconds<rampIntervalTimeSeconds){
   rampIndexUp = rampTime/(rampIntervalTimeSeconds - intervalTimeSeconds);
   rampIndexUpMiliseconds = rampTime % (rampIntervalTimeSeconds - intervalTimeSeconds);
 }
 else{
   rampIndexDown = rampTime/(intervalTimeSeconds - rampIntervalTimeSeconds);
   rampIndexDownMiliseconds = rampTime % (intervalTimeSeconds-
rampIntervalTimeSeconds);
 }
}
```

Programski kod 5.12. Matematički izračun za postupno mijenjanje intervala

Kako bi se postavio novi interval, potrebno je u izborniku, prikazanom na slici 5.8., odabrati *time ramping*. Otvara se zaslon za postavljanje vremena u kojemu će se interval mijenjati kao što je prikazano na slici 5.9. Ukoliko je *ramping* već pokrenut, Opcija *stop ramping* omogućuje prekidanje *ramping* funkcije te će interval ostati pri onoj vrijednosti koju je dosegao prilikom odvijanja tog istog *ramping*-a.

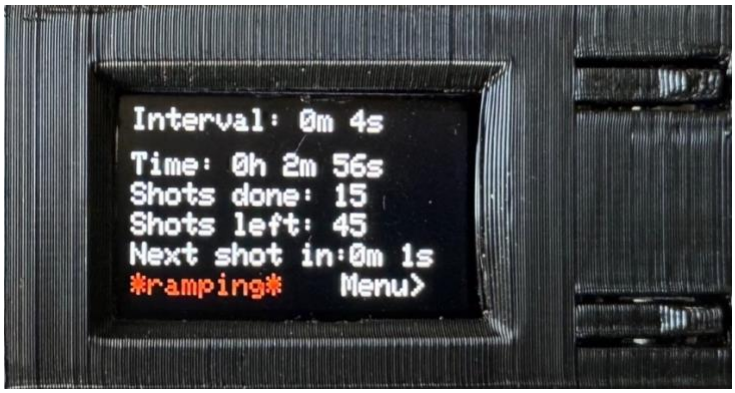
Slika 5.8. Izbornik za kontrolu izvođenja time-lapse-a

Slika 5.9. Postavljanje vremena za promjenu intervala

Nakon postavljanja vremena u kojemu će se interval promijeniti, potrebno je postaviti interval kao na slici 5.10.

Slika 5.10. Postavljanje novog intervala

Zbog smanjenja dupliciranja koda koristi se DRY (*Don't repeat yourself),* princip za postavljanje intervala za *ramping*. Kod je jednak onom prikazanom u programskom kodu 5.3. pri kojemu se pokreće dio koda ovisno o tomu koja je varijabla '*currentScreen'* trenutno pokrenuta. Kako je varijabla '*currentScreen'* trenutno jednak *rampingu,* sve vrijednosti odnose se na varijable povezane s *rampingom* intervala. Sav kod pisan je po istom principu.
Pokretanjem *ramping*-a prikazuje se zaslon na kojem se nesmetano izvodi *time-lapse* ali sada uz dodatnu radnju promjene intervala. Na zaslonu se u donjem lijevom kutu ispisuje riječ *'ramping'* crvenom bojom kako bi se dalo doznanja da je *ramping* u tijeku. Zaslon je prikazan na slici 5.11.



Slika 5.11. Izvođenje time-lapse-a uz ramping intervala

Ramping intervala radi na način da se nakon izračuna svakih koliko sekundi je potrebno mijenjanje intervala, interval promijeni ali se čeka do prvog najbližeg okidanja fotografije, kako bi *time-lapse* nesmetano se izvodio. Programski kod 5.13. Smanjivanje intervala pomoću *ramping*-a, prikazuje *ramping* intervala ukoliko se interval smanjuje.

```
void rampingLoop(){

if(startingMilisecondsFlag == 0 && startingMilisecondsDownFlag == 0){
   rampIndexCount++;
  }

  if(rampTime >0){
   rampTime--;
  }
  else{
   rampOn=0;
  }
  //decrease interval
  if(rampIndexDown == rampIndexCount){
   if(rampIndexDownMiliseconds==0){
     if(intervalSeconds != 0){
       intervalSeconds--;
     }
     else{
```

```
      intervalMinutes--;
      intervalSeconds = 59;
    }
    startingMilisecondsDownFlag = 0;
    rampIndexCount = 0;
    changeTimeIndex = 1;
  }else{
    startingMilisecondsDownFlag = 1;
  }
}
```

Programski kod 5.13. Smanjivanje intervala pomoću ramping-a

Ova funkcionalnost jedna je od najvećih prednosti i inovacija jer omogućuje promjenu intervala bez zaustavljanja *time-lapse*-a te omogućuje njegovo nesmetano izvođenje. Veliki problem kod već postojećih rješenja je potreba za zaustavljanjem izvođenja *time-lapse*-a što niti jednom fotografu nije u cilju. Ovo inovativno rješenje daje veliki doprinos u svijetu fotografije. Dakako, svaki fotograf mora poznavati način rada fotoaparata pa tako mora znati da postavljeni interval ne smije imati manju vrijednost od postavljene dužine okidanja fotografije na fotoaparatu. Postavljena dužina okidanja, odnosi se na otvor blende, odnosno na to koliko dugo će fotoaparat okidati fotografiju. Na primjeru, za okidanje fotografija predmeta u kretanju, postavlja se kratko vrijeme okidanja fotografije kako fotografija ne bi bila mutna. Ukoliko se fotografija zviježđe, vrijeme fotografiranja se postavlja i do trideset sekundi što stvara fotografiju s jasno vidljivim zvijezdama.

5.4. Funkcionalnosti tipkala

Za što jednostavnije korištenje intervalometra, osmišljeno je rukovanje istim koristeći samo dva gumba. Gornji gumb, odnosno gumb broj jedan, koristi se za povećavanje vrijednosti varijabli te potvrdu akcija. Donji gumb, odnosno gumb broj dva, koristi se za potvrdu akcija, smanjenje vrijednosti varijabli te dugi klik na donji gumb donosi promjenu pozicije kursora o kojem se govori na početku poglavlja.

Provjera stanja gumba neprestano se provjerava u neprekidnoj petlji '*loop*'. Programski kod koji prikazuje rad gornjeg gumba dan je pod brojem 5.14. Prikaz funkcije gornjeg gumba.

```
else if (pressDuration <= 500) {
      if(currentScreen == mainMenuScreen){
        setInterval();
      }
      if(currentScreen == intervalScreen){
        setInterval();
      }
      else if(currentScreen == rampScreenTime){
        setRampTime();
      }
      else if(currentScreen == rampScreenInterval){
        setRampInterval();
      }
…
```

Programski kod 5.14. Prikaz funkcije gornjeg gumba

Priloženi dio koda prikazuje odvijanje funkcije ukoliko je pritisnut gornji gumb u vremenskoj duljini manje od pola sekunde.

Kako se mikrokontroler sastoji od samo 2 gumba, bilo je potrebno osmisliti logiku za upravljanje cijelim uređajem samo putem ta 2 gumba. Baš zbog toga, implementiran je kod za prepoznavanje dugog klika. Ukoliko je određeni gumb stisnut duže od pola sekunde, očitava se dugi klik kako je prikazano u programskom kod 5.15. Implementacija koda za očitavanje dugog klika.

```
 if (lastState == 1 && currentState == 0) { // button is pressed
   pressedTime = millis();
 } else if (lastState == 0 && currentState == 1 && btnFlag == 1) { // button is released
     unsigned long releasedTime  = 0;
     releasedTime = millis();
     btnFlag = 0;
     long pressDuration = releasedTime - pressedTime;
   if (pressDuration > 500) {
```

```
    if(currentScreen == intervalScreen){
      btn1LongClickFlag=1;
      setInterval();
    }
…
```

Programski kod 5.15. Implementacija koda za očitavanje dugog klika

U situacijama kada je potrebno inkrementalno povećavati vrijednost varijable kao kod povećavanja intervala kako bi se olakšalo i ubrzalo povećavanje intervala, osmišljena je implementacija za povećavanje varijable sve dok gumb nije otpušten. Kod je prikazan u programskom kodu 5.16. Implementacija koda za inkrementalno povećavanje vrijednosti.

```
if(currentScreen==intervalScreen){
      while(currentState2==0){
         int elapsedTime = millis()-pressedTime2;
         if(elapsedTime >= 500){
           delay(200);
               if(intervalCursorX==50){
                 intervalMinutes++;
                 setInterval();
               }
               else if(intervalCursorX==90){
                 intervalSeconds++;
                 setInterval();
               }
           }
         currentState2=digitalRead(button_2_pin);
       }
…
```

Programski kod 5.16. Implementacija koda za inkrementalno povećavanje vrijednosti

6. WEB SUČELJE

Kako mikrokontroler koji se koristi za izradu intervalometra ima samo 2 gumba, korisnost njegove upotrebe može biti ograničena. Kako bi se olakšalo korištenje intervalometra, razvijeno je web sučelje koje omogućuje interakciju s mikrokontrolerom putem tipkovnice. Na taj način, unos podataka postaje brz, jednostavan i efikasan.
Da bi se uspostavila veza, potrebno je putem *wifi* veze spojiti se na mikrokontroler i putem ip adrese otvoriti web sučelje. Važno je znati kako *wifi* veza nema veliki doseg pa je primjerice mobitel na kojem se otvara web sučelje potreban biti unutar nekoliko metara od samog uređaja.

6.1. Spajanje putem *WebSocket* veze

Za početak, potrebno je uspostaviti vezu između mikrokontrolera i web sučelja. To je moguće ostvariti putem *WebSocket* veze.
WebSocket je tehnologija koja omogućuje dvosmjernu komunikaciju između klijenta (npr. web preglednika) i poslužitelja putem jedne TCP veze. To znači da omogućuje stvaranje trajne veze između klijenta i poslužitelja te omogućuje slanje podataka u oba smjera u realnom vremenu [13].

Za implementaciju WebSocket veze potrebno je učiniti slijedeće.
U Arduinu na samom početku potrebno je uključiti biblioteke te unijeti podatke (ssid i lozinka) za spajanje na mrežu kao što je prikazano u programskom kodu 6.1. Biblioteke i Wifi podaci. SSID predstavlja naziv vašeg interneta na koji se povezujete.

```
#include <WiFi.h>
#include <WebServer.h>
#include <WebSocketsServer.h>
const char* ssid = "Network name";
const char* password = "password";
```

Programski kod 6.1. Biblioteke i Wifi podaci

Nakon toga stvoriti web poslužitelj (*WebServer*) koji će osluživati HTTP zahtjeve na *port-u* 80, i *WebSocket* poslužitelj (*WebSocketsServer*) koji će osluživati *WebSocket* veze na *port-u* 81. Ovo omogućuje da se web aplikacija komunicira s mikrokontrolerom putem HTTP zahtjeva (preko

WebServer-a) i *WebSocket*-a za stvarnu dvosmjernu komunikaciju u stvarnom vremenu. Programski kod je prikazan pod brojem 6.2. Inicijalizacija poslužitelja.

```
WebServer server(80);
WebSocketsServer webSocket = WebSocketsServer(81);
```

Programski kod 6.2. Inicijalizacija poslužitelja

Odabirom opcije *'Connect to web'* na glavnom izborniku za spajanje na *WebSocket* kako je prikazano na slikama 6.1. i 6.2. pokreće se funkcija *connectToWifi*, kako je prikazano u kodu 6.3. Funkcija *connectToWifi.*

Slika 6.1. Glavni izbornik

Slika 6.2. Spajanje s Web serverom

```
void connectToWiFi() {
  Serial.println("Establishing connection to WiFi with SSID: " + String(ssid));
  WiFi.begin(ssid, password);
```

```
  while(WiFi.status() != WL_CONNECTED) {
     delay(1000);
     Serial.print(".");
  }
  if(WiFi.status() == WL_CONNECTED){
   Serial.println(WiFi.localIP());
   Serial.println("WiFi connected!");
   webFlag = 1;
   server.on("/main", [] (){
    server.send(200, "text\html", timelapseScreenWeb);
   });
   server.on("/start", [] (){
    server.send(200, "text\html", startScreenWeb);
   });

   server.on("/runningScreen", [] (){
    server.send(200, "text\html", runningScreenWeb);
   });

   server.on("/countdown", [] (){
    server.send(200, "text\html", countdownScreenWeb);
   });

   server.on("/setRampScreen", [] (){
    server.send(200, "text\html", setRampScreenWeb);
   });
   server.begin();
   webSocket.begin();
   webSocket.onEvent(webSocketEvent);
   Serial.println(currentScreen);
   if(currentScreen == connectToWebScreen){
     setInterval();
  }
 }
```

Programski kod 6.3. Funkcija connectToWifi

Za početak se provjerava status *wifi* veze, odnosno čeka se sve dok se ne stvori veza. Pokreće se server i *WebSocket*.

U glavnoj petlji programa provjerava se pristup web poslužitelju i obrađuju se dolazni klijentski zahtjevi pomoću 'server.handleClient()'. Također, provjerava se i stanje *webSocket* veze i obrađuju se dolazne poruke pomoću 'webSocket.loop()'. Prikazana dva dijela koda pod brojem 6.4. Klijentski zahtjevi i stanje *webSocket* veze, omogućuju komunikaciju s klijentima putem HTTP zahtjeva i *webSocket* poruka.

```
server.handleClient();
 webSocket.loop();
```

Programski kod 6.4. Klijentski zahtjevi i stanje webSocket veze

Za uspješnu komunikaciju između mikrokontrolera i web sučelja potrebno je implementirati *socket* na strani weba. To omogućuje slanje i primanje podataka između uređaja i korisničkog sučelja putem *web socket* veze. Na taj način mikrokontroler može primati naredbe ili slati podatke koji se prikazuju na web sučelju. Kako je prikazano u programskom kodu 6.5. Implementacija *Socketa* na Webu te slanje podataka, inicijalizira se *WebSocket*, kreira se varijabla *Socket* te se pomoću funkcije '*send* ' šalju podaci na mikrokontroler.

```
var Socket;
Socket = new WebSocket('ws://' + window.location.hostname + ':81/');
 function submitFormData() {
        var minutes = parseInt(document.getElementById("minutesInput").value);
        var seconds = parseInt(document.getElementById("secondsInput").value);


        Socket.send(minutes);
        Socket.send(seconds);
    }
```

Programski kod 6.5. Implementacija Socketa na Webu te slanje podataka

U programskom kodu 6.6. prikazano je primanje podataka s mikrokontrolera na poslužitelj.

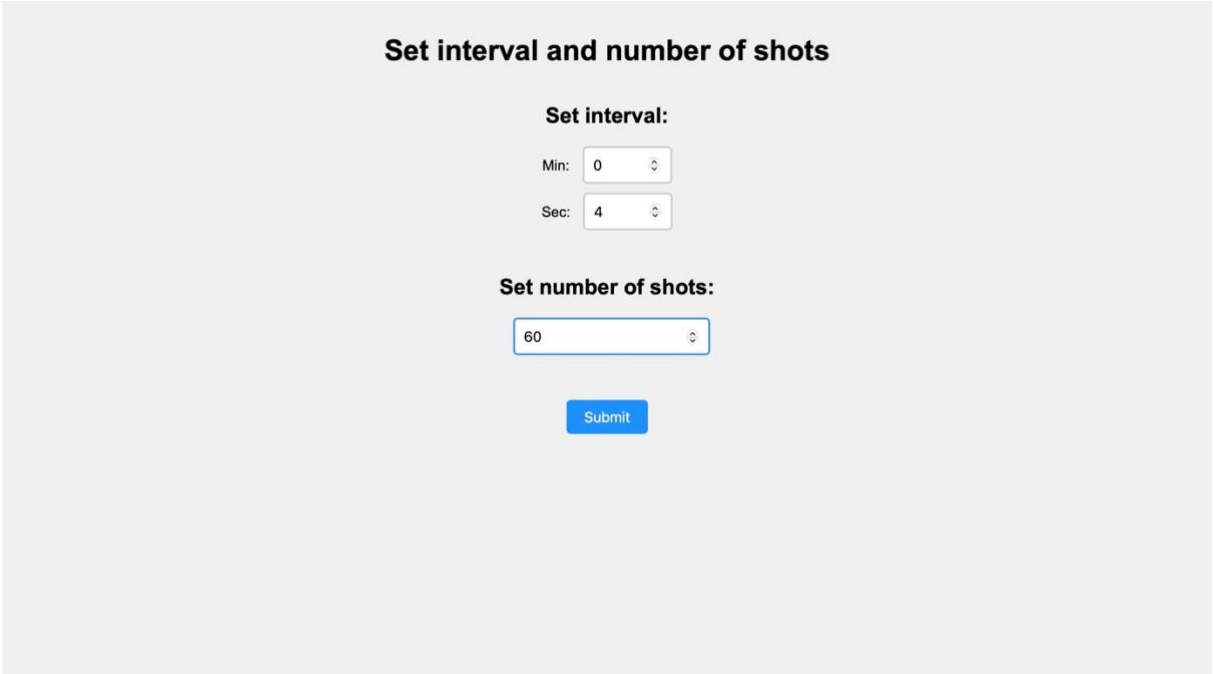
```
var Socket;
      Socket = new WebSocket('ws://' + window.location.hostname + ':81/');
      Socket.onmessage = function(event) {
         var data = JSON.parse(event.data);
         document.getElementById('intervalMinutes').textContent = data.intervalMinutes;
….
```

Programski kod 6.6. Primanje podataka s mikrokontrolera na web poslužitelj

Inicijalizira se WebSocket koji omogućuje komunikaciju između web sučelja i mikrokontrolera. Kada se primi poruka preko web *socketa*, podaci se parsiraju iz JSON formata, a zatim se ažurira HTML element s ID-om 'intervalMinutes' na web stranici s novim podacima koji su primljeni.

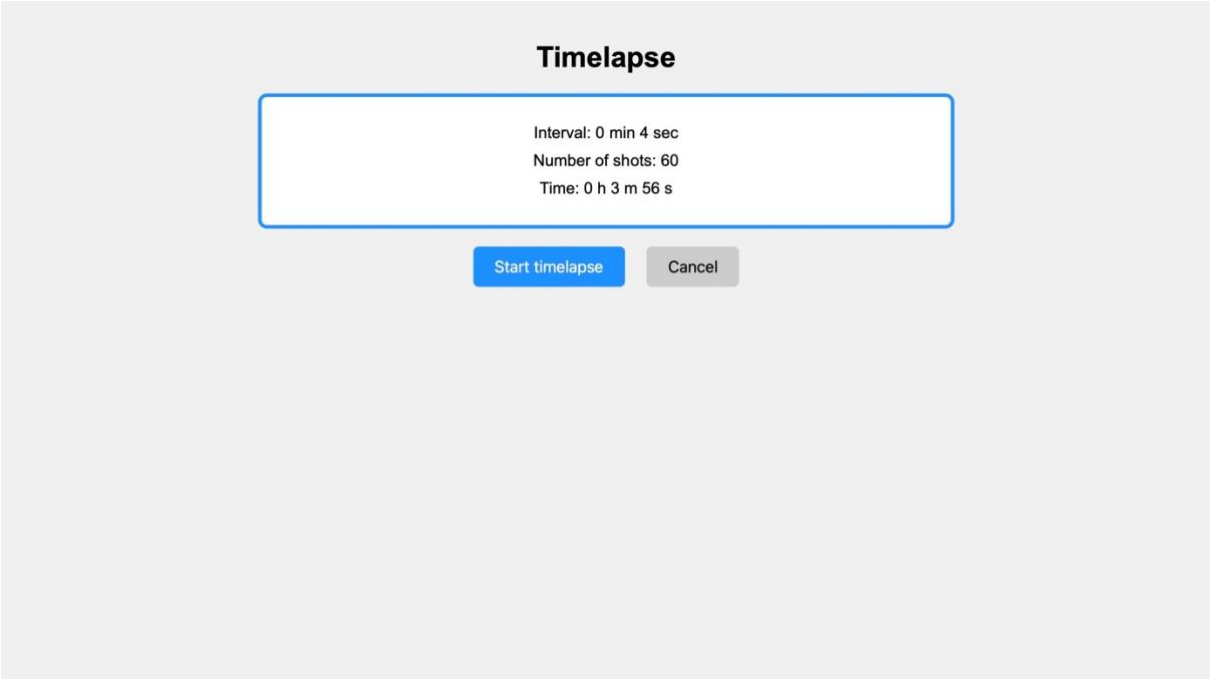
6.2. Prikaz web sučelja

Nakon uspješno uspostavljenje veze, koristeći IP adresu i '/main' otvara se početni zaslon te web adrese prikazan na slici 6.3. Početni zaslon web stranice za upravljanje intervalometrom.
Za izradu web sučelja koristi se Visual Studio Code.



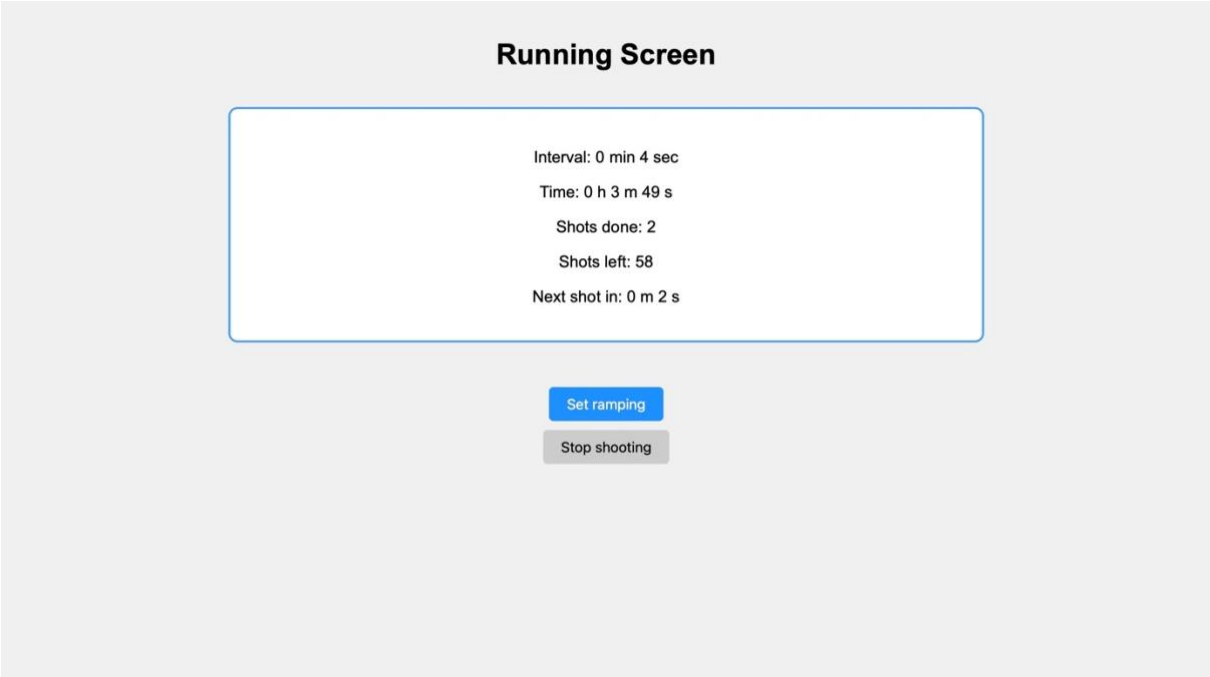
Slika 6.3. Početni zaslon web stranice za upravljanje intervalometrom

Web stranica funkcionira na način identičan prethodno opisanom kodu za upravljanje intervalometrom. Kako je pokazano na slici 6.3. za početak se postavlja interval te broj okidanja. Nakon postavljanja spomenutog otvara se zaslon za provjeru unesenog i mogućnost pokretanja procesa okidanja *time-lapse* fotografije kako je prikazano na slici 6.4.



Slika 6.4. Provjera unešenih podataka i pokretanje okidanja time-lapse fotografije

Slijedeći zaslon prikazan na slici 6.5. predstavlja odvijanje okidanja fotografije u kojem se prikazuje preostalo vrijeme okidanja, broj preostalih fotografije za okidanje, vrijeme do slijedećeg okidanja fotografije, trenutno postavljeni interval te mogućnost *'ramping-a'*.

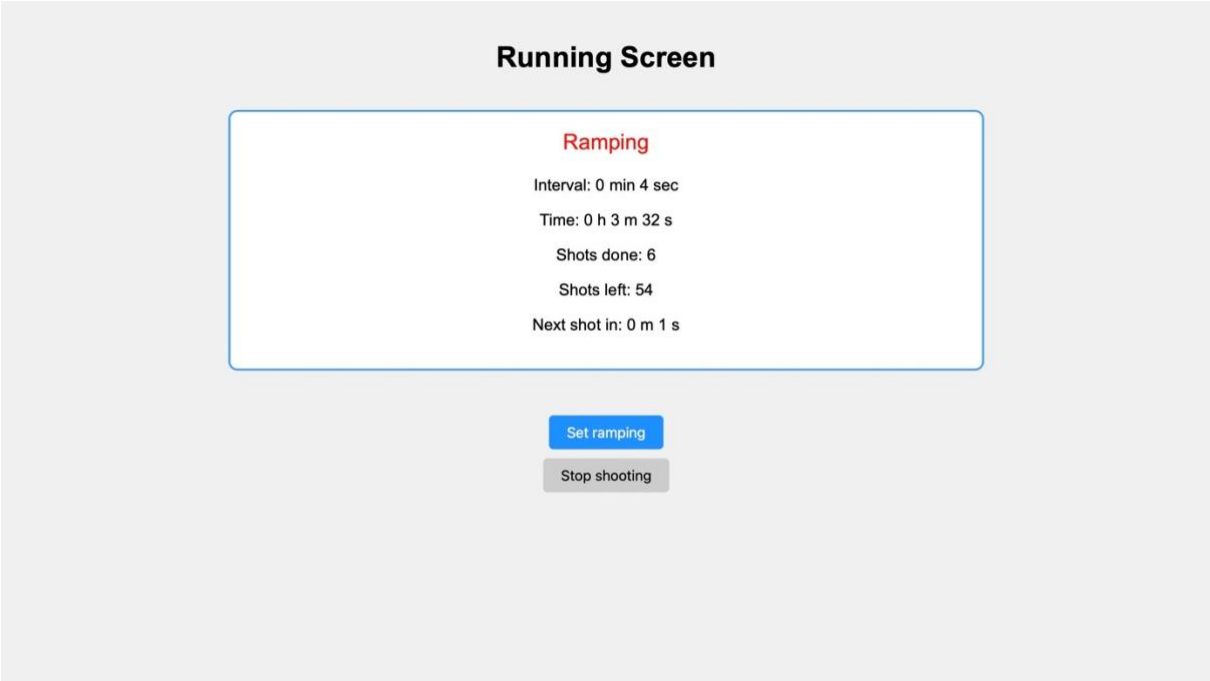


Slika 6.5. Running screen

Klikom na gumb '*set ramping'* otvaraju se zasloni za postavljanje *rampinga* prikazani na slikama 6.5 i 6.6.

Slika 6.6. Postavljanje podataka za ramping intervala

Ukoliko se pokrene *ramping*, otvara se stari zaslon s prikazom odvijanja *time-lapse* fotografije te natpisom *'ramping'* sve dok je proces izmjene intervala u izvođenju, kako je prikazano na slici 6.7.



Slika 6.7. Ramping intervala

7. TESTIRANJE

Nakon završetak izrade projekta slijedi testiranje istoga.

Za testiranje intervalne kamere u ovom slučaju koriste se:

- DSLR Nikon D5500 kamera
- kabel koji omogućuje daljinsko okidanje fotografije prikazan na slici 7.1.– razlikuje se od modela do modela [14]
- prijenosna baterija koja se spaja na mikrokontroler kao izvor napajanja
- tronožac za kameru po potrebi
- sama intervala kamera

Slika 7.1. kabel za daljinsko okidanje fotografije

Intervalna kamera koja je izrađena kroz ovaj projekt kompatibilna je sa svim DSLR i *mirrorless* fotoaparatima. Potrebno je obratiti pozornost da za svaki model fotoaparata se koristi drugačiji kabel za daljinsko okidanje. Projektu je potrebno dodati i prijenosnu bateriju kao izvor napajanja iako postoji mogućnost i spajanja litijumske baterije na uređaj. Za što čistiju i mirniju fotografiju koristi se tronožac kao postolje za fotoaparat. Na slici 7.2. prikazano je testiranje svega navedenog.

Slika 7.2. Prikaz testiranja projekta

Nakon okidanja fotografija u zadanom intervalu potrebno je sve fotografije spojiti u jedno koristeći jedan od programa za uređivanje kako bi se stvorio *time-lapse*.
Za potrebe ovog diplomskog rada, fotografije su prikazane kao slijed događaja na slici 7.3.

Slika 7.3. Rezultat testiranja

8. ZAKLJUČAK

Glavni cilj ovog projekta je razviti uređaj zvan intervalometar za snimanje *time-lapse* fotografije. U ovom diplomskom radu prikazan je proces dizajniranja i stvaranja hardverskog dijela projekta, implementacija i razvoj interaktivnog softverskog sučelja te razvoj web sučelja za korištenje istog. Intervalometri postaju neizostavan dio svijeta fotografije. Omogućuju snimanje *time-lapse* fotografije kroz određeno vrijeme bez mnogo rukovanja samim fotoaparatom. Kroz godine, uređaji se razvijaju i nude razne mogućnosti kako bi korištenje samih uređaja bilo još lakše korisniku. U samom kontekstu *time-lapse* fotografije bitno je okidati fotografije bez imalo šuma, držati fotoaparat statičnim što sami intervalometri i nude.

U ovom diplomskom radu, intervalometar je podignut na još višu razinu. Korisnik ima mogućnost postavljanja intervala i broja okidanja te nakon toga nije više dužan biti pokraj fotoaparata. Ono što čini ovaj rad inovativnim je opcija koja nudi promjenu intervala za vrijeme samog snimanja *time-lapse*-a. Ne ometajući snimanje postoji mogućnost promjene samog intervala. Primjerice, ukoliko je snimanje pokrenuto za vrijeme dana i polako noć, fotograf će htjeti produžiti vrijeme okidanja jedne fotografije za dobivanje željene fotografije.

Za stvaranje samog uređaja kako bi bio kompaktan i jednostavan za nošenje i korištenje, korišten je mali mikrokontroler LilyGo TTGO *Display*. Razvnovrstan kontroler uz dva gumba s ugrađenim wifi modulom. Uz dodatno razvijeni hardverski dio koji sadrži audio *jack* ulaz omogućuje se povezivanje fotoaparata i mikrokontrolera putem kabla. Kabel omogućuje sigurnu vezu između fotoaparata i mikrokontrolera.

Kako bi se korištenje istoga znatno olakšalo, razvijeno je i web korisničko sučelje. Korisnik ima opciju upravljanja uređajem putem mobitela ili laptopa koristeći *wifi* vezu. Ta inovativnost doprinosi korištenju uređaja na jednostavniji način bez korištenja gumbova. Također, ukoliko se fotografiranje obavlja vani na hladnom, upravljanje iz auta daje znatnu prednost.

Nakon suočavanja s brojnim izazovima i usvajanja novih znanja tijekom izrade projekta, konačno je stvoren funkcionalan uređaj spreman za uporabu. Ono što ovaj uređaj čini posebnim su poboljšanja u odnosu na postojeća rješenja. Ovaj intervalometar nudi jednostavno sučelje, učinkovitu upotrebu te dodatnu praktičnost koja olakšava korištenje.

LITERATURA

[1] LilyGo T-Display, https://www.lilygo.cc/products/lilygo®-ttgo-t-display-1-14-inch-lcd-esp32-control-board , Pristupljeno [25. ožujka 2024.]

[2] S. Spanulescu, „ESP32 Programming for the Internet of Things [2 ed.] “, 2020.

[3] Websocket – V. Maksimovic, „WebSocket Essentials - Building Apps with HTML5 WebSockets.", Packt Publishing, 2018.

[4] Lens Notes, Remote Shutter Release, wired, https://lensnotes.com/photography/remote-shutter-release/, Pristupljeno [25. ožujka 2024.]

[5] Canon Eos, Remote Shutter Release Switch, https://www.canon-europe.com/camera-accessories/rs-80n3-remote-switch/ , Pristupljeno [25. ožujka 2024.]

[6] Lens Notes, Remote Shutter Release, Infrared, https://lensnotes.com/photography/remote-shutter-release/, Pristupljeno [25. ožujka 2024.]

[7] MIOPS, MIOPS Smart +, https://www.miops.com , Pristupljeno [25. ožujka 2024.]

[8] G. Wegner, LRTimelapse Pro, https://lrtimelapse.com/lrtimelapse-pro-timer-free/, Pristupljeno [26. ožujka 2024.]

[9] KiCad, https://www.kicad.org , Pristupljeno [26. ožujka 2024.]

[10] J.Electronics, Optocoupler, https://www.jameco.com/Jameco/workshop/Howitworks/what-is-an-optocoupler-and-how-it-works.html#:~:text=An%20optocoupler%20(also%20called%20optoisolator,detects%20light%20from%20the%20LED , Pristupljeno [26. ožujka 2024.]

[11] Stevech, TTGO T-Display Case with header space, https://www.thingiverse.com/thing:4813428 , Pristupljeno [26. ožujka 2024.]

[12] T. Lee, Extending a TFT_eSPI Example With TTGO T-Display Using PlatformIO, With DumbDisplay, https://www.instructables.com/Extending-a-TFTeSPI-Example-With-TTGO-T-Display-Us/ , Pristupljeno [26. ožujka 2024.]

[13] Wallarm, WebSocket Protocol, https://www.wallarm.com/what/a-simple-explanation-of-what-a-websocket-is , Pristupljeno [27. ožujka 2024.]

[14] MJKZZ, Shutter release cable - https://www.mjkzz.de/products/camera-shutter-release-cable?variant=18436006183027 , Pristupljeno [27. ožujka 2024.]

SAŽETAK

U diplomskom radu objašnjen je proces razvijanja uređaja za snimanje *time-lapse* fotografije. Primijenjeno je znanje u razvijanju hardvera, softvera te razvijanje web sučelja. Uređaj zvan intervalometar ima mogućnost okidanja *time-lapse* fotografije. Nudi mogućnost promjene intervala okidanja fotografija za vrijeme snimanja. Za lakše korištenje uređaja razvijeno je web korisničko sučelje, pa se tako uređaj može kontrolirani i putem mobitela odnosno laptopa.

Ključne riječi: intervalometar, LilyGo, *time-lapse,* web sučelje, interval, okidanja, *ramping*

ABSTRACT

Development of a Remote Control System for Time-Lapse Photography

In the thesis, the process of developing a device for capturing time-lapse photography is explained. Knowledge was applied in developing hardware, software, and creating a web interface. The device, called an intervalometer, has the capability to trigger time-lapse photography. It offers the possibility to change the interval of taking photos during shooting. For easier usage, a web user interface has been developed, allowing the device to be controlled via mobile phones or laptops.

Keywords: intervalometer, LilyGo, time-lapse, web interface, interval, triggering, ramping

ŽIVOTOPIS

Filip Pitlović rođen je 14. svibnja 1999. godine u Slavonskom Brodu gdje je pohađao osnovnu školu Đuro Pilar u Brodskom Vinogorju. Godine 2014. upisuje Opću gimnaziju u Slavonskom Brodu. Nakon završenog srednjoškolskog školovanja upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, gdje 2021. godine završava preddiplomski sveučilišni studij i upisuje diplomski smjer Programsko inženjerstvo.

_____

Potpis autora