

Kalibriranje boja ekrana

Stipić, Zvonimir

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://um.nsk.hr/um:nbn:hr:200:873303>

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

Download date / Datum preuzimanja: **2024-05-11**

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

Kalibriranje boja ekrana

Završni rad

Zvonimir Stipić

Osijek, 2023.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju**

Osijek, 20.09.2023.

Odboru za završne i diplomske ispite

**Prijedlog ocjene završnog rada na
preddiplomskom sveučilišnom studiju****Ime i prezime Pristupnika:** Zvonimir Stipić**Studij, smjer:** Programsko inženjerstvo**Mat. br. Pristupnika, godina upisa:** R4568, 28.07.2020.**OIB Pristupnika:** 11008798456**Mentor:** izv. prof. dr. sc. Josip Job**Sumentor:** ,**Sumentor iz tvrtke:****Naslov završnog rada:** Kalibriranje boja ekrana**Znanstvena grana rada:** Informacijski sustavi (zn. polje računarstvo)**Zadatak završnog rad:**

U okviru završnog rada student treba proučiti načine kalibracije boja na ekranima. Kao praktičan dio rada potrebno je, upotrebom kamere u kontroliranom okruženju, omogućiti pronalaženje optimalnih vrijednosti varijabli kako bi se omogućio vjeran prikaz crne, bijele i šest nijansi sive boje. Tema rezervirana za: Zvonimir Stipić Sumentor iz tvrtke: David Zovko (Soldered Electronics d.o.o.)"

Prijedlog ocjene završnog rada: Dobar (3)**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: 1 bod/boda
Jasnoća pismenog izražavanja: 2 bod/boda
Razina samostalnosti: 2 razina

**Datum prijedloga ocjene od strane
mentora:** 20.09.2023.**Datum potvrde ocjene od strane
Odbora:** 27.09.2023.

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, 27.09.2023.

Ime i prezime studenta:	Zvonimir Stipić
Studij:	Programsko inženjerstvo
Mat. br. studenta, godina upisa:	R4568, 28.07.2020.
Turnitin podudaranje [%]:	3

Ovom izjavom izjavljujem da je rad pod nazivom: **Kalibriranje boja ekrana**

izrađen pod vodstvom mentora izv. prof. dr. sc. Josip Job

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. PREGLED PODRUČJA.....	2
3. UREĐAJI I OKOLINA	3
3.1. Inkplate 10	3
3.2. Raspberry Pi 4 i Raspberry Pi camera	4
3.3. Rad s Inkplateom	5
3.4. Rad s računalom Raspberry Pi.....	6
3.5. Kontrolirano okruženje	6
4. PROGRAMSKA REALIZACIJA	9
4.1. Analiza postojećih primjera	9
4.2. Kodiranje prikaza boja	11
4.3. Kodiranje algoritma za mijenjanje boja.....	14
5. ISPITIVANJE SUSTAVA ZA AUTOMATSKO KALIBRIRANJE	18
6. ZAKLJUČAK.....	20
LITERATURA	Error! Bookmark not defined.
SAŽETAK.....	22
ABSTRACT	23
ŽIVOTOPIS.....	24

1. UVOD

Elektronički papir je vrsta digitalnog ekrana koji je dizajniran da imitira izgled i osjećaj tradicionalnog papira. Ovi se ekrani koriste za prikazivanje teksta i slika, a nude nekoliko ključnih prednosti u odnosu na tradicionalne ekrane. Prednosti elektroničkog papira su: niska potrošnja energije, dobra čitljivost te tanki dizajn. Zbog tih prednosti elektronički papir se koristi u e-čitačima, pametnim satovima te digitalnim natpisima. Takav ekran se koristi u uređaju Inkplate 10 tvrtke Soldered. Povećanjem potražnje i proizvodnje uređaja Inkplate 10 u tvrtki Soldered pojavila se potreba za ubrzanjem kalibriranja boja ekrana rečenog uređaja. Boje ekrana se kalibriraju tako da se u kodu uređaja izmjeni takozvani valni oblik (engl. *waveform*). Valni oblik za Inkplate 10 je matrica cjelobrojnih vrijednosti 8x9 gdje svaki od osam redaka predstavlja jednu nijansu (crna, bijela te 6 nijansi sive). Potreba za kalibriranjem postoji jer je rad upravljača valnih oblika na komercijalno dostupnim uređajima nepoznanica svima osim njihovim autorima. Ukoliko bi se takav upravljač koristio Inkplate uređaj prestaje biti otvorenog izvora (engl. *open source*) te gubi svoju svrhu. Ručno kalibriranje boja ekrana je mukotrpan posao jer se ekrani nabavljaju iz više izvora te nisu identični što znači da se za gotovo svaki ekran mora tražiti valni oblik metodom pokušaja i pogreške. Zbog metode i veličine valnog oblika proces može trajati i do 4 sata što je zbog opsega proizvodnje nepraktično rješenje te je ovaj rad pokušaj automatizacije rečenog procesa. U raduje u drugom poglavlju predstavljen ciljani uređaj i zamišljeno kontrolirano okruženje, u trećem poglavlju programska realizacija algoritma za kalibriranje dok je u četvrtom poglavlju ispitana uspješnost sustava za kalibriranje.

1.1. Zadatak završnog rada

U okviru završnog rada bilo je potrebno proučiti načine kalibracije boja na ekranima. Tijekom praktičnog dijela rada bilo je potrebno, upotrebom kamere u kontroliranom okruženju, omogućiti pronalaženje optimalnih vrijednosti varijabli kako bi se omogućio vjeran prikaz crne, bijele i šest nijansi sive boje. Ciljano vrijeme kalibracije je manje od 30 minuta, sa stavom – kraće je bolje jer se samim smanjivanjem vremena kalibracije smanjuju troškovi kao što su energija, a posebice ljudsko vrijeme. Ukoliko se znatno smanji vrijeme i trošak kalibracije moguće je povećati opseg proizvodnje kako bi se dostigla potražnja. Zadovoljavajuća kalibracija je oku vrlo slična već kalibriranom ekranu uz dozvoljeno manje odstupanje bez šuma u boji, u zadanom vremenu.

2. PREGLED PODRUČJA

Iako je elektronička tinta osmišljena 1970-ih te realizirana krajem 1990-ih odnosno ranih 2000-ih [1], područje kalibracije ekrana elektroničkog papira je izrazito neistraženo i neobrađeno zbog čega je i nastala potreba za ovim radom. Valni oblik je ključan dio rada elektroničkog papira jer je valni oblik skup električnih impulsa koji pokreću kretanje čestica u mikro kapsulama zaduženih za bojanje piksela [2]. Rad upravljača valnih oblika u komercijalno dostupnim uređajima kao što je na primjer Amazon Kindle su strogo čuvana tajna [3] pa je zbog toga nemoguće napraviti sličan uređaj odnosno čip koji bi automatski prilagodio boje ekrana na temelju modela ekrana. Postavljanje postojećih upravljača valnih oblika u Inkplate 10 je moguće, no s obzirom na ciljanu skupinu potrošača tvrtke Soldered koja se sastoji isključivo od hobista kojima je potrebno da uređaj bude otvorenog izvora [4], postavljanje takvog uređaja u Inkplate 10 je besmisleno. S obzirom na veličinu zajednice hobista koji koriste ekrane elektroničkog papira na tržištu se nije pojavilo rješenje poput ovog rada jer je moguće imati postojane boje bez upravljača, ali se onda žrtvuje djelomično osvježavanje.

3. UREĐAJI I OKOLINA

U ovom poglavlju su predstavljeni uređaji potrebni za rad te njihova uporaba. Glavni i ciljani uređaj je Inkplate 10 proizvodnje tvrtke Soldered, dok su ostali uređaji zamjenjivi uređajima sličnih funkcionalnosti i specifikacija. Ti uređaji su Raspberry Pi s modulom Raspberry Pi Camera Rev 1.3, te LED traka svjetla bijele boje. Za potrebe rada je također konstruirana okolina s kontroliranim uvjetima od crnih stranica napravljenih na 3D pisaču povezanih aluminijskim ekstruzijama na rubovima. Navedeni uređaji su odabrani zbog zadovoljavajućih specifikacija i nabave.

Oprema

Inkplate 10

Raspberry Pi 4

Raspberry Pi Camera Module

2 LED trake

Okruženje sastavljeno od crnih stranica povezanih aluminijskim ekstruzijama

Tablica 1. Popis opreme

3.1. Inkplate 10

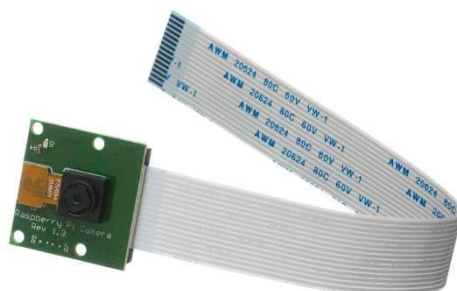
Inkplate 10 uređaj je tvrtke Soldered. Ekran uređaja je 9,7 inčni 1200*825 elektronički papir s podrškom za nijanse sive boje. Na ekranu su moguće djelomične izmjene što omogućava brze promjene sadržaja. Pogoni ga ESP32 mikro kontroler s ugrađenim WiFi-jem i Bluetoothom 4.0. Moguće ga je napajati USB kabelom ili litijском baterijom, a troši iznimno malo baterije te ga je moguće koristiti i godinama s jednom baterijom [5]. Za uređaj se može besplatno preuzeti Arduino

biblioteka, a kompatibilan je i s micropythonom. Inkplate uređaji su sasvim otvorenog izvora kako bi se mogao koristiti u sve zamislive svrhe.

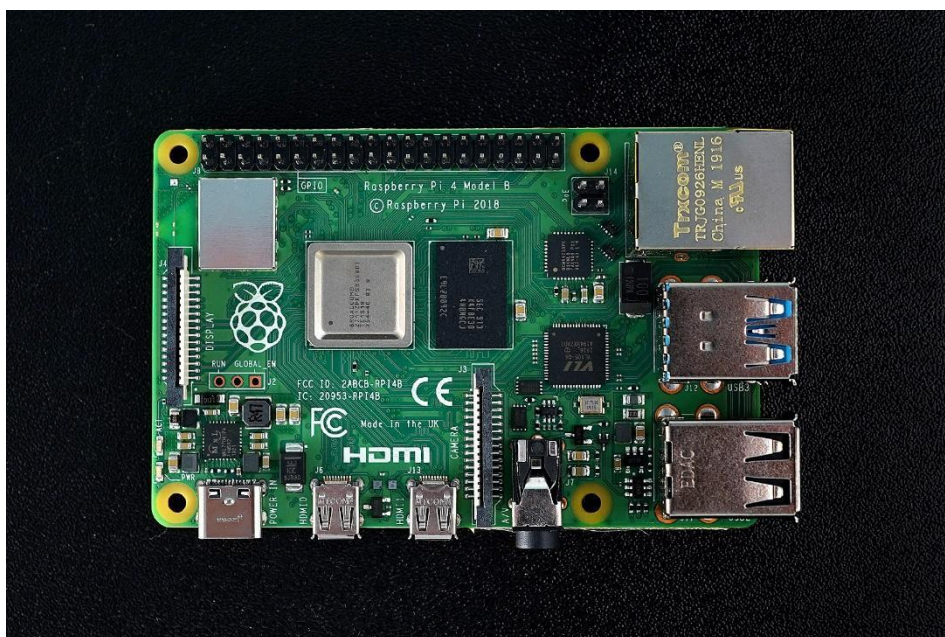
Slika [STYLEREF 1 \s 3](#). SEQ Slika * ARABIC \s 1 1. Inkplate 10

3.2.Raspberry Pi 4 i Raspberry Pi camera

Za prepoznavanje nijansi boja i slanje povratnih informacija Inkplateu predviđeno je korištenje računala Raspberry Pi 4 s modulom Raspberry Pi Camera. Raspberry Pi 4 je malo računalo s jednom pločom (engl. *single-board computer*) s procesorom frekvencije 1,5 GHz i s 4 jezgre [6]. Rečeni uređaj je prigodan za uporabu u ovom projektu jer je dovoljno brz u obradi da ne usporava proces kalibracije, a dovoljno je kompaktan da ga nije teško primijeniti. Raspberry Pi Camera je kamera malih dimenzije te male mase, kompatibilna s računalom, a rezolucija slike 2592*1944p je dovoljna za raspoznavanje različitih nijansi, ali potencijalno i šuma (pikseli druge nijanse) u boji. Značajke rečene kamere je čine idealnom za uporabu u radu.



Slika 3.2. Raspberry Pi Camera



Slika 3.3. Raspberry Pi 4

3.3.Rad s Inkplateom

Za rad s Inkplateom potrebno je poznavati Arduino razvojno okruženje te je rad s rečenim uređajem prilično jednostavan uvelike zbog već spremnih primjera prikaza raznih oblika. Jedan takav primjer je uz male preinake savršen za testiranje uspješnosti algoritma jer prikazuje svih 6 nijansi sive boje te također bijelu i crnu boju. Rečeni prikaz je jako bitan za provjeru uspješnosti kalibracije jer ne smije biti prevelika razlika između prve tamnije i prve svjetlije nijanse svake

boje iako spadaju u prihvatljive spektre tražene nijanse. Nadalje, za svaku nijansu treba provjeriti ima li šuma u njoj, a zbog veličine zaslona i pojedinog polja na prikazu moguće je vidjeti sve te uvjete.

Za manipuliranje valnim oblikom potrebno je uz poznavanje rada s prikazom na Inkplateu poznavati i Inkplate biblioteku. U rečenoj biblioteci se nalazi datoteka „Inkplate_10_Wavefrom_EEPROM_Programming.ino“ [7] u kojoj se nalazi 5 predodređenih valnih oblika te funkcija za njihov prikaz. Rečena datoteka postoji kako bi krajnji korisnik mogao sam odabrati jedan od tih predodređenih valnih oblika koji najviše odgovara njegovom ekranu. Po uzoru na taj primjer, odnosno njegove funkcije te predodređene valne oblike je u okviru ovog završnog rada stvorena Arduino aplikacija koja i nove primjere valnih oblika u nadi da će jedan od tih najbolje odgovarati jedinici Inkplatea 10 s kojim se radi.

Linija Kod

```
uint8_t waveform1[8][9] = {{0, 0, 0, 0, 0, 0, 0, 1, 0}, {0, 0, 0, 2,  
33: 2, 2, 1, 1, 0}, {0, 0, 2, 1, 1, 2, 2, 1, 0},  
    {0, 1, 2, 2, 1, 2, 2, 1, 0}, {0, 0, 2, 1, 2, 2, 2, 1, 0}, {0, 2, 2,  
34: 2, 2, 2, 2, 1, 0},  
35: {0, 0, 0, 0, 0, 2, 1, 2, 0}, {0, 0, 0, 2, 2, 2, 2, 2, 0}};
```

Kôd 1. Primjer jednog predodređenog valnog

Linija Kod

```
1: image = cv2.imread(image.jpg)  
2: white = image[100,100]  
3: print(white)
```

Kôd 2. Primjer spremanja i ispisivanja bijele nijanse sa fotografije

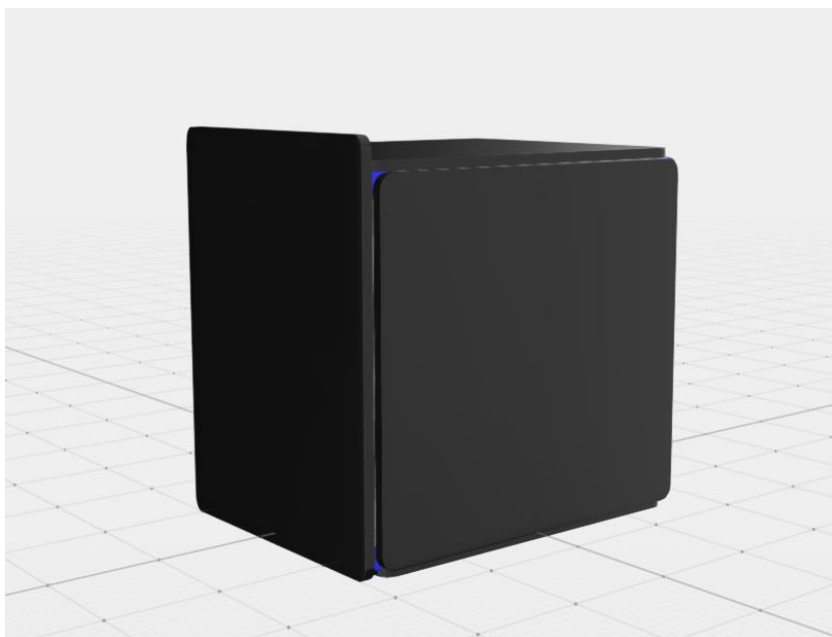
3.4. Rad s računalom Raspberry Pi

Za rad s računalom Raspberry Pi u ovom završnom radu nužno je bilo poznavanje programskog jezika Python s bibliotekama OpenCV i NumPy. Raspberry Pi, odnosno Raspberry Pi camera module je bio korišten tako da se nakon svake promjene valnog oblika fotografirao Inkplate te se spremila ta fotografija na SD karticu. Nakon spremanja fotografije u .py datoteci ta ista fotografija

se učita „imread“ metodom iz OpenCV biblioteke [8], [9]. Rečena metoda kao parametar prima tekst koji označava putanju do ciljane fotografije. Zatim se sprema vrijednost piksela u području fotografije gdje se nalazi ciljane nijansa. Spremanje vrijednosti piksela u varijable treba ponoviti za svaku nijansu te se te nijanse uspoređuju s predodređenim vrijednostima za valjanost nijanse te se uspoređuju susjedne nijanse kako se ne bi dogodila prevelika razlika susjednih nijansi.

3.5. Kontrolirano okruženje

Za vjerodostojnost vrijednosti pojedine nijanse Inkplate mora biti fotografiran u istim uvjetima svaki puta. Kako svjetlost ne bi utjecala na fotografije, a samim time i na vrijednosti pojedine nijanse potrebno je osigurati zatvoreno okruženje što se može lako realizirati pločama napravljenim na 3D pisaču ukoliko su one dovoljno duboke te crne boje koja najmanje propušta svjetlost. Te ploče se mogu spojiti aluminijskim ekstruzijama kako bi se osigurao oblik te čvrstoća kutije. Za efikasnost i brzinu pri promjeni pojedinog Inkplatea potrebno je imati jednu pomičnu stranicu te ostale fiksne. Još jedan razlog za korištenje Raspberry Pi Camera modula jest debljina kabela s kojim je on povezan s računalom Raspberry Pi. S obzirom da je bitno da kutija ne propušta svjetlost, te s obzirom na to da je potrebno montirati Raspberry izvan okruženja, rupa za kabel kamere treba biti minimalna i u okomitoj dimenziji što nam omogućuje rečena debljina kabela. Zbog kontroliranja vanjskih uvjeta Inkplate ostaje neosvijetljen te je potrebno uvesti izvor svjetlosti koji je nepromjenjiv, a dostupan, minimalan u dimenzijama te nije centraliziran. Jedan takav oblik osvjetljenja su LED trake. LED trake su idealne za problem osvjetljenja Inkplatea jer se mogu postaviti dužinom cijelog Inkplatea što znači da je za svaku nijansu na ekranu svjetlina otprilike jednaka što je nužno za očitavanje vrijednosti boje. Zbog pozicioniranja kamere na sredini gornje stranice kutije gledajući okomito na Inkplate, postoji mogućnost odsjaja svjetlosti na pojedinim dijelovima ekrana Inkplatea što remeti vrijednosti boja koje spremamo. Praktičnim istraživanjem utvrđeno je da je idealno mjesto za postavljanje LED traka bočno pod kutom od 90 stupnjeva u odnosu na Inkplate za koji je idealno nekoliko centimetara iznad. Za vjerodostojnost boja također je bitna i boja svjetla LED traka te je za potrebe razlikovanja nijansi boja idealna hladna bijela boja (engl. *cool white*) jer je najbližnja sunčevoj svjetlosti koja sadrži sve nijanse spektra ljudima vidljivih boja u istom iznosu.



Slika 3.4. 3D skica kutije za zaštitu od vanjskog svjetla



Slika 3.5. LED traka bijele boje za osvetljenje

4. PROGRAMSKA REALIZACIJA

U ovom poglavlju su analizirani postojeći primjeri valnih oblika u Arduino Inkplate 10 biblioteci te je na temelju tih primjera razrađen algoritam za kalibraciju kako bi se taj algoritam mogao kodirati.

4.1. Analiza postojećih primjera

Za dobro razumijevanje rada Inkplatea te načina prilagođavanja boja pojedinom Inkplateu potrebno je razumjeti prvotno postavljene valne oblike s kojima dolazi sam proizvod. Svaki uređaj dolazi s pet definiranih valnih oblika od kojih je jedan odabran i spremljen u memoriju te se taj valni oblik koristi za prikaz boja jer najbliže dočarava nijanse za tu jedinicu uređaja. Zbog raznolikosti ekrana s kojima dolaze pojedini Inkplate tih pet primjera nije dovoljno za vjerno predstavljanje nijansi boja, no analizom tih primjera se može doći do zaključaka ključnih za ubrzanje pronalaska idealnih valnih oblika pri kalibraciji. Ukoliko rastavimo valne oblike po bojama možemo vidjeti jasne uzorke. Tako na primjer možemo uzeti najtamniju nijansu i vidjeti da je u tri od pet valnih oblika crna predstavljena identičnim nizom od osam nula i jedne jedinice na pretposljednem mjestu. Rečene nule i jedinice predstavljaju faze u kojima se pikseli posvjetljuju, potamnjuju ili ostaju isti. U fazi gdje je nula, piksel ostaje isti, gdje je jedinica piksel se potamnjuje dok se u fazi gdje je dvojka piksel posvjetljuje. Ukoliko se pojavi više faza tamnjenja bez faza svijetljenja dolazi do rizika zadržavanja slike (engl. *image retention*), vidi Slika 4.1. U slučaju zadržavanja slike proces podešavanja valnih oblika potrebno je zaustaviti i pokrenuti primjer iz „Diagnostics“ datoteke Inkplate Arduino biblioteke pod naslovom „Inkplate10_Burn_In_Clean“. Zbog takvih potencijalnih problema je potrebno po uzorku valnih oblika ograničiti algoritam da bi se smanjili izgledi prekida procesa. Metodom pokušaja i pogreške te primjerima valnih oblika dolazimo do zaključka da se idealan niz faza za crnu nijansu sastoji od osam nula i jedne jedinice za koju u većini slučajeva nije bitno na kojem se mjestu nalazi.

Analizom sljedeće svjetlije nijanse zaključujemo da je izrazito velika vjerojatnost da se ona sastoji od niza koji većinom sadrži tri dvojke, dvije ili tri jedinice te je ostatak popunjen brojkom nula. Stoga je potrebno ograničiti algoritam da pri generiranju niza za tu nijansu on sadrži baš taj omjer faza svijetljenja i tamnjenja.

Linija Kod

```
1:      waveform[0] = {0, 0, 0, 0, 0, 0, 0, 1, 0}
```

Kôd 3. Primjer niza u većini slučajeva idealne nijanse crne

Linija ***Kod***

```
1:      waveform[1] = {0, 0, 0, 2, 2, 2, 1, 1, 0}
```

Kôd 4. Primjer mogućeg niza idealne druge nijanse

Linija ***Kod***

```
1:      waveform[3] = {0, 0, 2, 2, 1, 2, 2, 1, 0}
2:      waveform[4] = {0, 0, 2, 1, 2, 2, 2, 1, 0}
3:      waveform[5] = {2, 2, 2, 1, 1, 2, 1, 2, 0}
4:      waveform[6] = {0, 0, 0, 0, 0, 2, 1, 2, 0}
5:      waveform[7] = {0, 0, 0, 2, 2, 2, 2, 2, 0}
```

Kôd 5. Primjer nizova za nijanse od rednog broja četiri do 8

Linija ***Kod***

```
1:      waveform[2] = {0, 2, 2, 2, 1, 2, 2, 1, 0}
```

Kôd 6. Primjer mogućeg niza treće nijanse

Linija ***Kod***

```
1:      waveform[1] = {0, 0, 0, 2, 1, 2, 1, 1, 0}
```

Kôd 7. Primjer još jednog mogućeg niza druge nijanse

Daljnjom analizom valnih oblika zaključujemo da je treća nijansa većinom niz koji sadrži tri do pet faza svijetljenja te bar dvije faze tamnjenja. Četvrta nijansa može biti niz s tri do pet dvojki te između jedne i pet jedinica, peta dvije do četiri dvojke i dvije do tri jedinice, šesta četiri do šest

dvojki i jedne do tri jedinice, sedma dvije ili tri dvojke i jedne jedinice te je osma većinom niz četiri do pet dvojki bez jedinica.

Pri analiziranju ovih nizova može se zaključiti da nema jasnog uzorka kako se odnose faze svijetljenja i tamnjenja jedna na drugu te vrlo vjerojatno boljeg pristupa problemu traženja idealnih nizova za nijanse od nasumičnog odabira s uvjetima odnosa faza koje smo već izvukli iz primjera predodređenih valnih oblika.

4.2.Kodiranje prikaza boja

Za dobar prikaz nijansi koje se koriste potrebno ih je vidjeti u isto vrijeme kako bi se mogao vidjeti njihov međusobni odnos. Po uzoru na već spomenuti primjer „Inkplate_10_Wavefrom_EEPROM_Programming.ino“ (Slika 4.3) taj učinak se može postići prikazom pojedinih nijansi u jednoliko raspoređenim pravokutnicima iste širine, kao na Slika 4.3. Za takav prikaz potrebno je imati početni valni oblik kao na Kôd 1. i potrebno je imati funkciju koja se poziva nakon promjena boja. Ta funkcija se zove „showGradient“ (Kôd 8.) te kao parametar prima broj tipa int koji se zove „_selected“, a označava redni broj boje koju je korisnik posljednju promijenio kako bi mu se bilo lakše snaći pri kalibraciji. Prvo potrebno nakon svake promjene prikaza na Inkplateu je čišćenje zaslona odnosno poziv funkcije `display.clearDisplay` [10]. Nakon poziva rečene funkcije sljedeće je nužno pozvati funkciju `display.changeWaveform` jer ona primjenjuje vrijednosti nizova predanog valnog oblika na boje zaslona. Manje bitan dio funkcije

showGradient je ispisivanje bijelog pravokutnika na dnu zaslona te ispis teksta na njega koji prikazuje upute za izbor boje koja se mijenja (Kôd 9.). Najvažniji dio funkcije je for petlja od osam iteracija kojoj se nalazi poziv funkcije display.fillRect koja označava ispunu pravokutnika na koordinatama označenim prvim i drugim brojem, visine i širine označene trećim i četvrtim brojem te bojom označenom zadnjim parametrom funkcije odnosno brojem iteracije.

Linija Kod

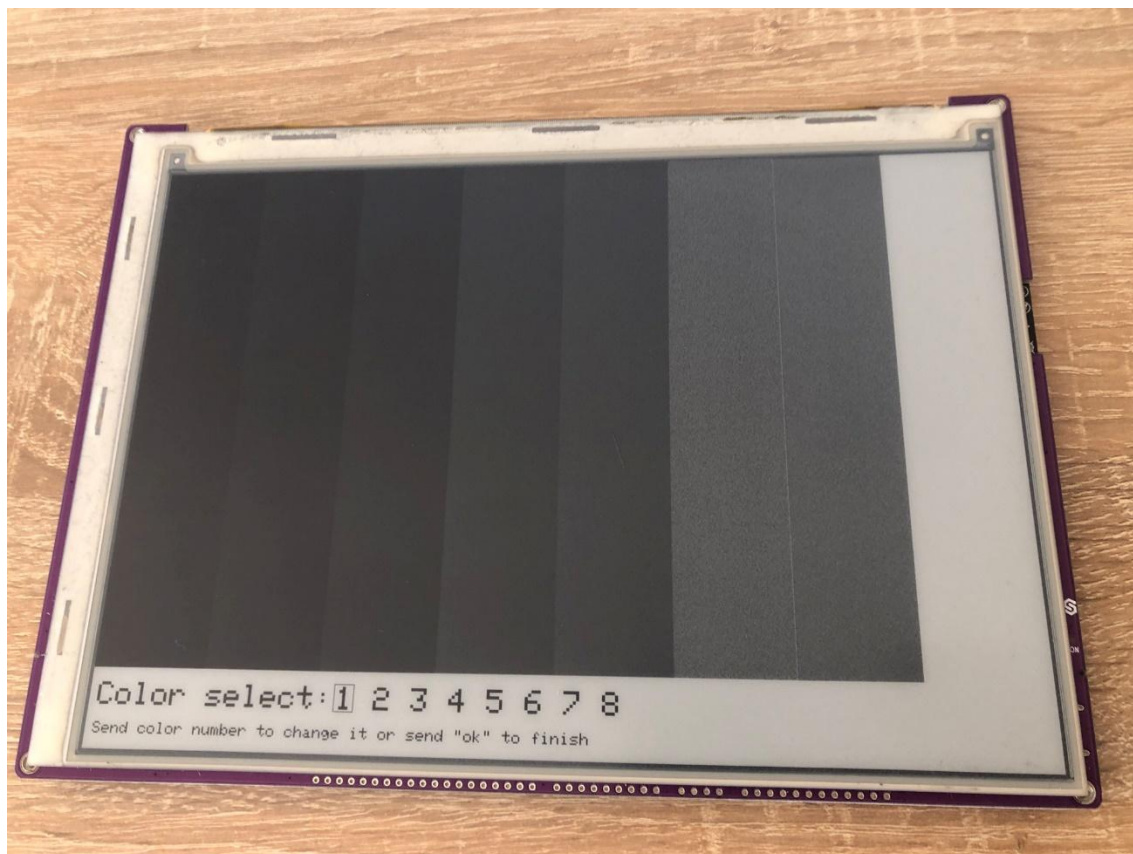
```
1:      void showGradient(int _selected)
```

Kôd 8. Funkcija showGradient

Linija Kod

```
130:    display.changeWaveform(&waveform[0][0]);
131:    display.fillRect(0, 725, 1200, 100, 7);
132:    display.setTextSize(4);
133:    display.setTextColor(0);
134:    display.setCursor(10, 743);
135:    display.print("Color select: ");
136:    display.setTextSize(2);
137:    display.setCursor(10, 792);
138:    display.print("Send color number to change it or send \"ok\" to finish");
139:    display.display();
```

Kôd 9. Niz poziva funkcija koje ispisuju i uređuju tekst uputa na zaslonu



Slika 4.2. Novi prikaz napravljen po uzoru na Slika 4.3.



Slika 4.3. Primjer prikaza korištenjem „Inkplate_10_Wavefrom_EEPROM_Programming.ino“



Slika 4.4. Dijagram procesa kalibracije

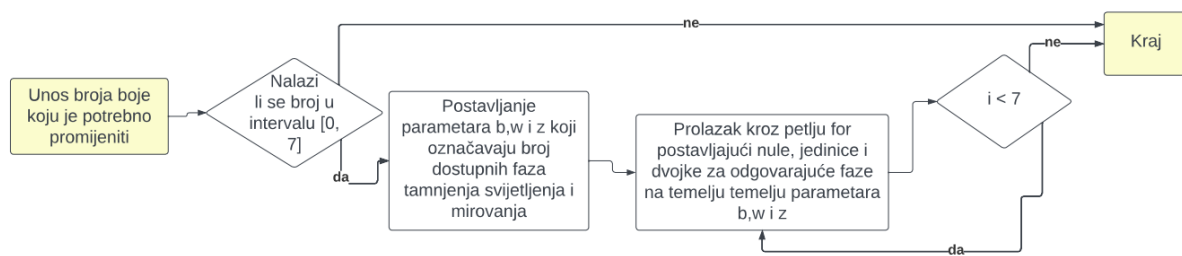
4.3. Kodiranje algoritma za mijenjanje boja

Za izmjenu boja odnosno valnog oblika potrebno je napisati novu funkciju koja se u ovom radu zove `changeColor`. Funkcija ima ti jedan parametar tipa `int`, a to je isti parametar `_selected` iz funkcije `showGradient`. Ovoj funkciji je taj parametar potreban jer se na temelju tog broja, indeksa boje, postavljaju uvjeti za stvaranje novog niza nula, jedinica i dvojki odnosno faza tamnjenja, svijetljenja i mirovanja. Za postavljanje tih parametara korištena je *switch-case* koji pod uvjetom indeksa odabrane boje na temelju analize predodređenih valnih oblika (poglavlje 4.1.) postavlja vrijednosti. Tako na primjer ako je `_selected` jednak nuli odnosno crnoj boji bit će nula faza svijetljenja, nula, jedna ili dvije faze tamnjenja te će ostatak biti faze mirovanja (Kôd 11.). Za postavljanje vrijednosti niza korištena je `for` petlja koja prolaskom kroz niz pri svakoj iteraciji nasumično odabire fazu te provjerava smije li se ta faza postaviti u niz na temelju uvjeta postavljenih u *switch-caseu*. Nakon postavljanja faze umanjuje se broj dozvoljenih pojava faze za jedan jer je jedna postavljena te ako je na primjer u crnoj boji postavljeno dvije faze tamnjenja

broj preostalih dozvoljenih će doseći nulu te će u ostatak niza biti postavljene samo nule odnosno faze mirovanja (Kôd 11.).

Kôd SEQ Kôd_ * ARABIC 10. Postavljanja vrijednosti niza za pojedinu boju

Kôd SEQ Kôd_ * ARABIC 11. Postavljanje uvjeta za algoritam mijenjanja boja

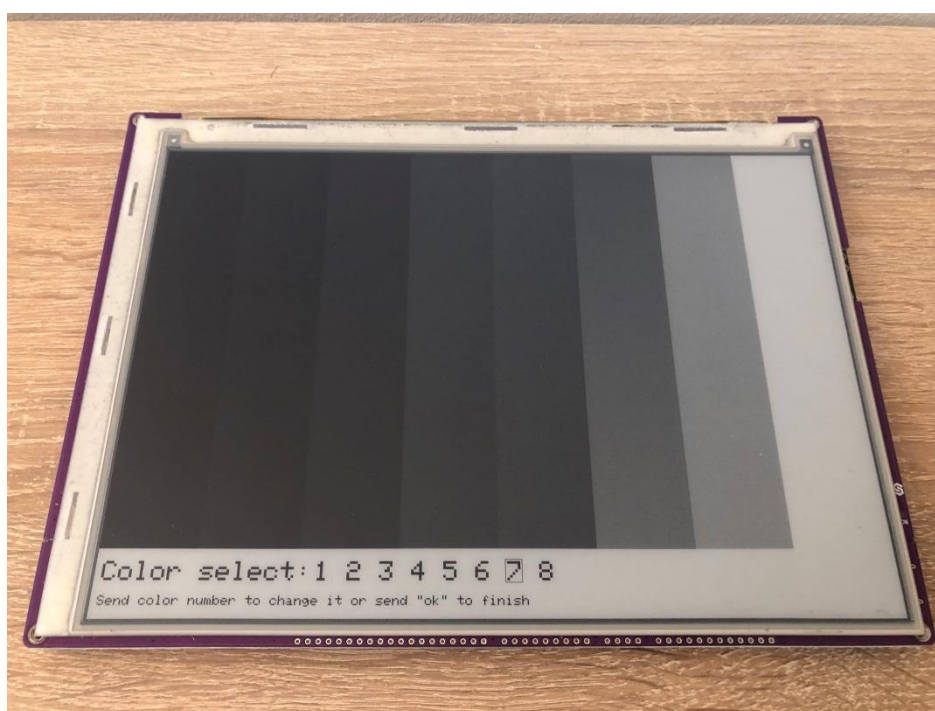


Slika 4.5. Dijagram algoritma kalibracije

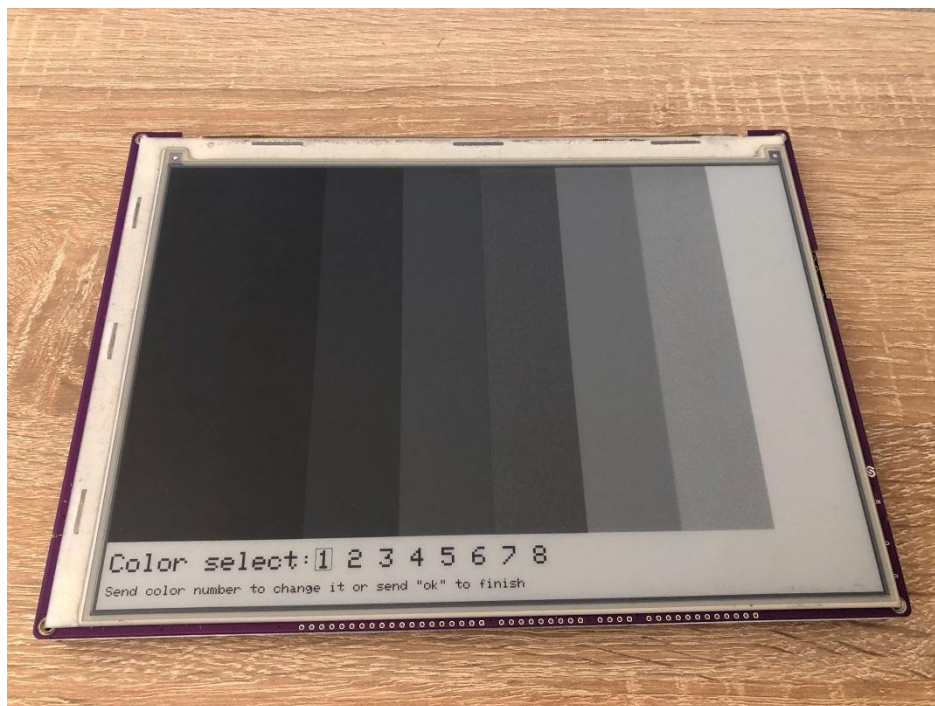
Slika STYLEREF 1 \s 4. SEQ Slika * ARABIC \s 1 6. Kôd postavljanja vrijednosti niza za pojedinu boju

5. ISPITIVANJE SUSTAVA ZA AUTOMATSKO KALIBRIRANJE

Testiranjem napisanog programa do rezultata se dolazi izrazito brzo. Za početni valni oblik koristi se onaj koji najbliže odgovara uređaju s kojim se radilo te se izmjenjivanjem boja koje previše odskaku ili imaju šuma do optimalnog dolazi do optimalnog u otprilike deset minuta. Iako se pri testiranju pojavilo više velikih odstupanja od ciljane boje, razlog nije algoritam već nepredvidivost valnog oblika koji ne potamnjuje jednako boje pri svakoj fazi tamnjenja te je zaključak da preinake koda nisu potrebne. S obzirom da je ciljano vrijeme za nedosegnuti cilj bilo trideset minuta, ovaj rezultat daje više nego dovoljno vremena za implementaciju sinkroniziranog rada Inkplatea i Raspberry računala, odnosno Raspberry kamere.



Slika 5.1. Primjer kalibriranog ekrana



Slika 5.2. Primjer nekalibriranog ekrana s vidljivim šumom i prevelikom odstupanjima

6. ZAKLJUČAK

U ovom je radu programiran algoritam koji uz ljudsku pomoć zamjenjuje upravljač valnih oblika koji se ne može koristiti na ciljanom uređaju ovog rada. Cilj ovog rada je bio napraviti potpuno automatiziran uređaj koji bi samostalno mijenjao valne oblike dok ne bi došao do zadovoljavajućeg rezultata, no zbog problema nabave komponenti, te manjka vremena za prilagodbu uvjetima cilj rada nije sasvim dosegnut. Umjesto predviđene analize rezultata upotrebom Raspberry Pi 4 i Raspberry Pi Camera modula u kontroliranom okruženju, rezultate je potrebno potvrditi ljudskim okom, kojem ne treba kontrolirano okruženje. Iako cilj rada nije dosegnut, vrijeme kalibracije je znatno smanjeno upotrebom algoritma koji prikazuje boje stvorene novim valnim oblicima bez potrebe za ručnom promjenom svake vrijednosti matrice koja sadrži sedamdeset dvije vrijednosti. Vrijeme kalibracije je također umanjeno za vrijeme kompilacije koda pri svakoj izmjeni valnih oblika koja je bila potrebna u prijašnjem pristupu. Testiranjem se utvrdilo da su rezultati sasvim prihvatljivi za opseg rada koji je dostiživ.

1. LITERATURA

- [1] B.-R. Yang, E-Paper Displays.
- [2] B. Comiskey, J. D. Albert, H. Yoshizawa i J. Jacobson, An electrophoretic ink for all-printed reflective electronic displays.
- [3] M. Kozłowski, »GOODEREADER,« [Mrežno]. Available: <https://goodereader.com/blog/e-paper/e-ink-waveforms-are-a-closely-guarded-secret>.
- [4] Soldered, »PRIČA O SOLDERED-U,« [Mrežno]. Available: <https://soldered.com/hr/about/>.
- [5] Soldered, »Specifikacije proizvoda i Opis proizvoda,« [Mrežno]. Available: <https://soldered.com/hr/proizvod/soldered-inkplate-10-plocica-s-9-7-elektronickim-papirom/>.
- [6] Raspberry Pi, »Raspberry Pi 4 Tech Specs,« [Mrežno]. Available: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/>.
- [7] Soldered, »Welcome to Inkplate's documentation!,« [Mrežno]. Available: <https://inkplate.readthedocs.io/en/latest/>.
- [8] Raspberry Pi, »Raspberry Pi Begginer's Guide 4th Edition,« [Mrežno]. Available: https://magpi.raspberrypi.com/books/beginners-guide-4th-ed?_gl=1*_tph2te*_ga*MTc3ODMxMzQ5My4xNjc3MTU4ODM1*_ga_22FD70LWD S*MTY4ODA1NzgxOS4yLjEuMTY4ODA1NzgZNS4wLjAuMA...
- [9] OpenCV, »Image file reading and writing,« [Mrežno]. Available: https://docs.opencv.org/4.8.0/d4/da8/group__imgcodecs.html.
- [10] Arduino, »ArduinoGraphics,« [Mrežno]. Available: <https://www.arduino.cc/reference/en/libraries/arduinographics/>.

2. SAŽETAK

U radu je opisan uređaj Inkplate 10 te uređaji Raspberry Pi 4 i Raspberry Pi Camera Module koji bi se zajedno u kontroliranom okruženju, koje je također skicirano u radu, mogli koristiti za kalibraciju ekrana uređaja Inkplate. S obzirom na nedostatak svih potrebnih dijelova za stvaranje okoline s instaliranim rečenim uređajima umjesto Raspberry Pi 4 i Raspberry Pi Camera Modula za analizu i testiranje korištena je ljudska pomoć. Rad je djelomično i uspješan jer je napisan program uz čiju se pomoć svejedno ubrzala kalibracija ekrana te je i dalje moguća nadogradnja s ciljem da se izbací potreba za ljudskom pomoći.

Ključne riječi: Arduino, e-papir, kalibracija ekrana, waveform

3. ABSTRACT

Screen color calibration

This thesis describes the Inkplate 10 device, as well as the Raspberry Pi 4 and Raspberry Pi Camera Module, which can be used together in a controlled environment, also outlined in the article, for the calibration of the Inkplate device's screen. Due to the lack of all the necessary components to create an environment with the mentioned devices installed, human assistance was used instead of the Raspberry Pi 4 and Raspberry Pi Camera Module for analysis and testing. The work is partially successful because a program was written to expedite the screen calibration, and further upgrades are possible with the aim of eliminating the need for human assistance.

Keywords: Arduino, e-paper, screen calibration, waveform

4. ŽIVOTOPIS

Autor ovog završnog rada, Zvonimir Stipić, rođen je 10.10.2001. u Đakovu. Osnovnu školu je pohađao u Gašincima i Satnici Đakovačkoj u Osnovnoj školi Josipa Antuna Čolnća, kasnije Osnovnoj školi Satnica Đakovačka. Srednju školu je pohađao u Gimnaziji Antuna Gustava Matoša u Đakovu. Za vrijeme pisanja ovog rada studira na trećoj godini preddiplomskog sveučilišnog studija računarstva, smjer Programsko inženjerstvo na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek.

Potpis autora