

E-Ink pokaznik s NRF24 komunikacijom

Mađerić, Leon

Undergraduate thesis / Završni rad

2024

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

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:268134>

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

Download date / Datum preuzimanja: **2025-02-05**

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 prijediplomski studij Računarstvo

E-INK POKAZNIK S NRF24 KOMUNIKACIJOM

Završni rad

Leon Maderić

Osijek, 2024.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**Obrazac Z1P: Obrazac za ocjenu završnog rada na sveučilišnom prijediplomskom studiju****Ocjena završnog rada na sveučilišnom prijediplomskom studiju**

Ime i prezime pristupnika:	Leon Mađerić
Studij, smjer:	Sveučilišni prijediplomski studij Računarstvo
Mat. br. pristupnika, god.	R4674, 28.07.2021.
JMBAG:	0165091478
Mentor:	izv. prof. dr. sc. Tomislav Matić
Sumentor:	Josip Zidar, univ. mag. ing. comp.
Sumentor iz tvrtke:	
Naslov završnog rada:	E-Ink pokaznik s NRF24 komunikacijom
Znanstvena grana završnog rada:	Arhitektura računalnih sustava (zn. polje računarstvo)
Zadatak završnog rada:	U radu je potrebno razviti mikroupravljački sustav koji će bežičnim putem s NRF24 integriranim sklopom upravljati radom pokaznika temeljenog na E-Ink tehnologiji. Potrebno je izraditi odgovarajuće tiskane pločice te testirati rad s Inkplate pokaznikom. Tema rezervirana za: Leon Mađerić Sumentor: Josip Zidar
Datum prijedloga ocjene završnog rada od strane mentora:	09.09.2024.
Prijedlog ocjene završnog rada od strane mentora:	Izvrstan (5)
Datum potvrde ocjene završnog rada od strane Odbora:	25.09.2024.
Ocjena završnog rada nakon obrane:	Izvrstan (5)
Datum potvrde mentora o predaji konačne verzije završnog rada čime je pristupnik završio sveučilišni prijediplomski studij:	30.09.2024.



FERIT

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

IZJAVA O IZVORNOSTI RADA

Osijek, 30.09.2024.

Ime i prezime Pristupnika:

Leon Mađerić

Studij:

Sveučilišni prijediplomski studij Računarstvo

Mat. br. Pristupnika, godina upisa:

R4674, 28.07.2021.

Turnitin podudaranje [%]:

3

Ovom izjavom izjavljujem da je rad pod nazivom: **E-Ink pokaznik s NRF24 komunikacijom**

izrađen pod vodstvom mentora izv. prof. dr. sc. Tomislav Matić

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

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

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

Potpis pristupnika:

SADRŽAJ

1. UVOD	1
1.1. Zadatak završnog rada	1
2. PREGLED TRENUTNOG STANJA	2
3. INKPLATE 6 S NRF24 KOMUNIKACIJOM	4
3.1. Sklopovlje.....	4
3.1.1. Arduino platforma	4
3.1.2. NRF24L01	4
3.1.3. Inkplate 6	5
3.1.4. Izrada električne sheme	6
3.1.5. Tiskana pločica	8
3.2. Programska podrška.....	10
3.2.1. Obrada slike	10
3.2.2. Komunikacija	13
3.2.3. Smanjenje potrošnje	18
3.3. Testiranje	21
4. ZAKLJUČAK.....	25
LITERATURA	26
SAŽETAK.....	27
ABSTRACT	28
PRILOZI.....	29

1. UVOD

E-Ink pokaznici su vrsta zaslona sa malom potrošnjom u usporedbi sa tradicionalno korištenim LCD ili LED zaslonima. Često prikazuju samo crno-bijele slike ili crno-bijele slike sa nijansama sive. Glavni razlog upotrebe je smanjenje potrošnje, često u uređajima sa baterijskim napajanjem. Prijenos slike na E-Ink pokaznike može biti žičanim putem, u kojem slučaju se ne mogu postaviti na udaljena mjesta, te ponekad sporijim, bežičnim putem [1].

Bežična komunikacija omogućuje prijenos podataka između uređaja bez potrebe za fizičkim povezivanjem, koristeći elektromagnetske valove poput radio valova. NRF24 (točnije NRF24L01) je primopredajnik na frekvenciji 2.4 GHz. Prednosti naspram drugih, popularnijih tehnologija kao WiFi i Bluetooth je manja potrošnja energije, te lakše korištenje i prijenos podataka [2], [3]. Cilj ovog rada je upotrijebiti dva NRF24 uređaja kako bi se prenijeli podaci sa strane odašiljača, do prijemnika. Odašiljač je kontroliran računalom, a prijemnik je spojen na pokaznik.

Ostatak rada strukturiran je na sljedeći način. Drugo poglavlje će prikazati trenutne opcije koje postoje na tržištu. Zatim je u trećem poglavlju prikazana izrada sklopovlja, te programska podrška za svaki dio sustava, te na kraju postupak testiranja.

1.1. Zadatak završnog rada

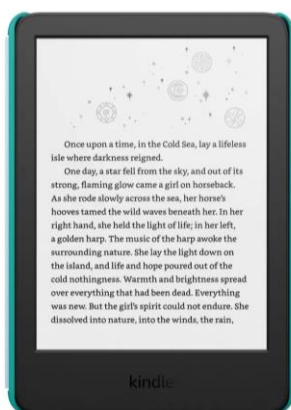
U radu je potrebno razviti mikroupravljački sustav koji će bežičnim putem s NRF24 integriranim sklopom upravljati radom pokaznika temeljenom na E-Ink tehnologiji. Potrebno je izraditi odgovarajuće tiskane pločice te testirati rad s Inkplate pokaznikom.

2. PREGLED TRENUTNOG STANJA

E-Ink pokaznici su često korišteni u urama, te za prikazivanje oglasa, jelovnika, medicinskih kartona i u drugim sličnim primjenama gdje bi se tradicionalno koristio papir, ali je potrebno napraviti povremene izmjene. Najveću popularnost su dobili pri korištenju u digitalnim knjigama (eng. *E-Book*), gdje se pri čitanju dobiva osjećaj pravog papira, sa prednostima bolje prenosivosti i učitavanja većeg broja knjiga. Još jedno mjesto korištenja je prikazivanje cijena u trgovinama umjesto uobičajenih papirnatih cijena. Takvi pokaznici često koriste vlastite bežične protokole za ažuriranje, koji su usporedivi sa protokolom u ovom radu, ali puno kraćeg dometa [1].

Primjeri proizvoda koji koriste E-Ink tehnologiju dani su na slici 2.1 [4], [5]. Amazon Kindle ima mogućnost postavljanja knjiga na uređaj pomoću WiFi komunikacije. Kod pokaznika za cijene ovisi o točnom modelu, ali često koriste NFC ili slične protokole za bežičnu komunikaciju kratke udaljenosti [6], [7].

E-Ink zasloni se sastoje od velikog broja malih kapsula, koje predstavljaju piksele na zaslonu. Unutar svake kapsule nalaze se bijele i crne čestice različitih naboja. Uglavnom su bijele čestice negativnog naboja, a crne čestice pozitivnog. Osim čestica, u kapsulama se nalazi i tekućina u kojoj čestice plutaju. Uzimanjem u obzir da su bijele i crne čestice različitih naboja, na njih možemo primjenjivati električna polja kako bi ih razdvojili, odnosno ovisno o vrstu električnog polja, jedne čestice bi privukli, a druge udaljili. Primjer toga može se vidjeti na slici 2.2 [8]. Također je prikazan jedan od načina kako se može ostvariti siva boja na takvim zaslonima. Nadalje, postoje i e-ink zasloni koji mogu prikazati više vrsta boja osim samo crne, bijele i nijansi



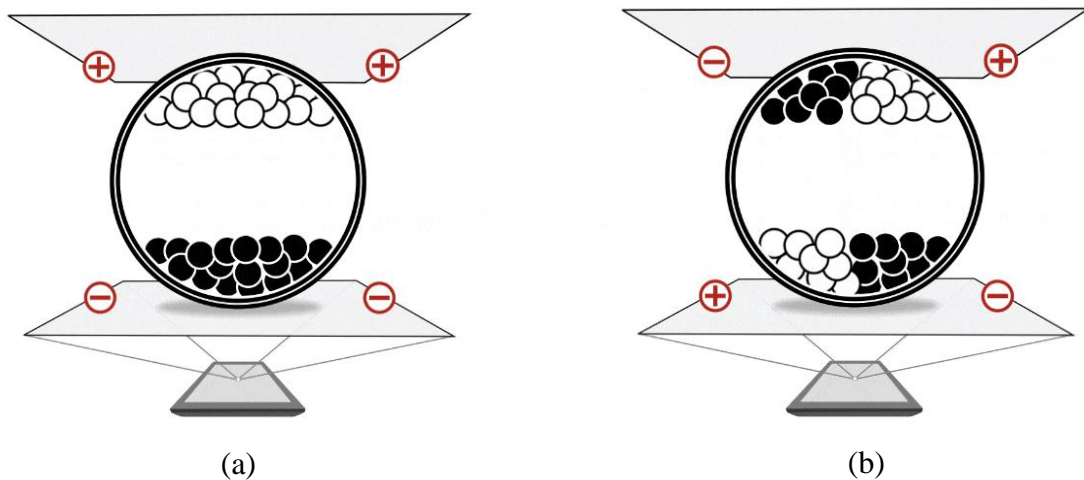
(a)



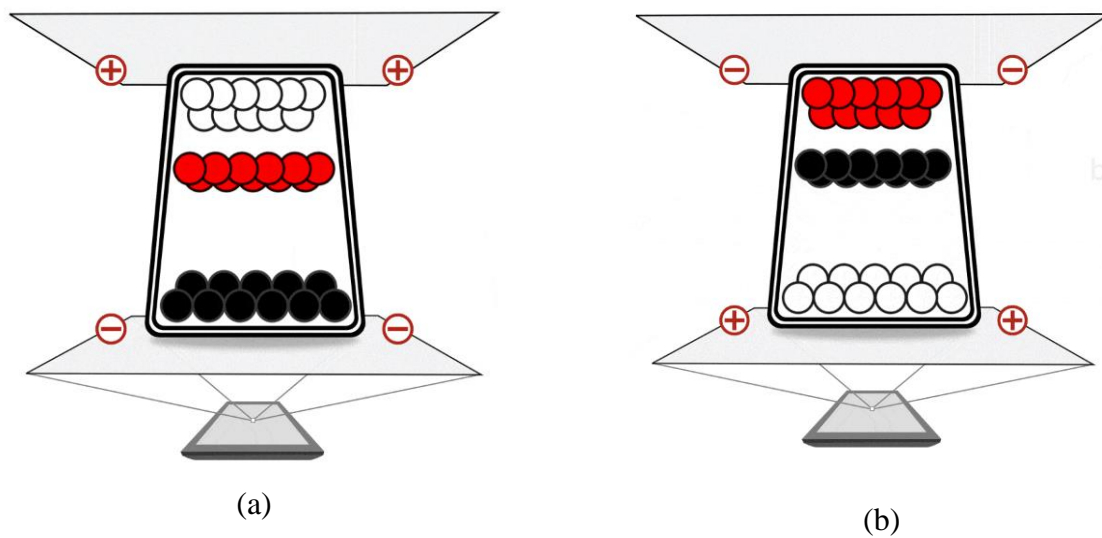
(b)

Sl. 2.1 Primjeri proizvoda sa E-Ink zaslonima (a) Amazon kindle. E-Ink čitač (b) E-Ink pokaznik za cijene

sive. Za takve zaslone je potrebna i kompleksnija kontrola. Prikazivanje više boja se može ostvariti dodavanjem dodatnih čestica koje po količini naboja između bijelih i crnih čestica. Pri promjeni boje potrebno je elektrizirati kapsulu dovoljno dugo da čestice u boji stignu do vrha, ali dovoljno kratko da ih jače elektrizirane čestice ne “izguraju”. Primjer višebojne E-Ink tehnologije može se vidjeti na slici 2.3 [8]. Na slici 2.3.a prikazana je bijela boja, a na slici 2.3.b crvena boja. To je rezultat kratkog (parcijalnog) elektriziranja, gdje su crvene čestice izbile na vrh, ali crne čestice nisu imale dovoljno vremena. Da se elektriziranje nastavilo, crne čestice bi izbile na vrh, a crvene čestice bi ponovno bile u sredini [8].



Sl. 2.2 Odvajanje bijelih i crnih čestica u e-ink zaslonu (a) crno/bijelo, (b) sivo



Sl. 2.3 Odvajanje višebojnih čestica u e-ink zaslonu (a) prije elektriziranja, (b) poslije parcijalnog elektriziranja

3. INKPLATE 6 S NRF24 KOMUNIKACIJOM

U poglavlju 3.1 objašnjene su sve komponente koje su korištene pri izradi, te električna shema i konačno pločica na koju su komponente postavljene. Zatim u poglavlju 3.2 objašnjen je programski kôd, na svakom sudioniku u komunikaciji, te su objašnjeni postupci pretvorbe slike, slanja podataka, prikaz slike i testiranje.

3.1. Sklopovlje

Za izradu sustava komunikacije, minimalno su potrebna dva mikroupravljača, kao i dva NRF24 primopredajnika i E-Ink uređaj. U ovom radu za izradu su korišteni Arduino Nano razvojna ploča na strani odašiljača odnosno Arduino Pro Micro na strani prijemnika za kontroliranje NRF24, te je za E-Ink pokaznik korišten Inkplate 6.

3.1.1. Arduino platforma

Za izradu odašiljača i prijemnika korištene su razvojne ploče Arduino platforme, kao i Arduino programski jezik koji je osnovan na C++ programskom jeziku.

Na strani odašiljača se koristi Arduino Nano, na kojemu se nalazi ATmega328P mikroupravljač i radi na naponu od 5 V, te taktu od 16 MHz. Za odašiljač najvažniji zahtjevi su mogućnost komunikacije sa računalom preko USB (eng. *Universal Serial Bus* – Univerzalna serijska sabirnica) priključka, te posjedovanje SPI (eng. *Serial Peripheral Interface* – Serijsko periferno sučelje) sučelja, radi komunikacije sa NRF24 [9].

Na strani prijemnika se koristi Arduino Pro Micro, na kojemu se nalazi ATmega32U4 mikroupravljač, koji radi na naponu od 3,3 V i taktu od 8 MHz. Navedeni mikroupravljač je izabran budući da koristi napajanje jednakog napona kao Inkplate, te nije potrebna dodatna regulacija napona. Nadalje, važno je i postojeće SPI sučelje, zbog komunikacije sa NRF24. Konačno je poželjna mala potrošnja kako bi, u slučaju da se koristi baterijsko napajanje, baterija trajala što duže i smanjilo potrebu za čestim zamjenjivanjem ili punjenjem. ATmega32U4 ima malu potrošnju u normalnom radu, ali važan dodatak je i postojanje načina rada spavanja (eng. *sleep mode*), što može uvelike smanjiti potrošnju energije kada nema prijenosa podataka, tj. kada se mikroupravljač ne koristi [10].

3.1.2. NRF24L01

NRF24L01 je primopredajnik niske potrošnje. Glavne značajke su rad pri frekvenciji od 2.4 GHz, maksimalna duljina prijenosa od 100 m, odnosno 1 km s vanjskom antenom, rad pri naponu od

3,3 V i 125 različitih kanala za komuniciranje. Za komuniciranje sa drugim uređajima koristi SPI protokol. Podržava brzine prijenosa od 250 kbps (eng. *kilobits per second*) do 2 Mbps (eng. *megabits per second*) sa GFSK (eng. *Gaussian Frequency Shift Keying* – Gaussova frekvencijska modulacija) modulacijom. Postoji i mogućnost korištenja prekidnog signala (eng. *interrupt*) kako bi mikroupravljač koji se koristi mogao biti u modu spavanja, te se probuditi na pristigli paket.

3.1.3. Inkplate 6

U radu je korišten E-Ink uređaj od tvrtke Soldered Electronics, model Inkplate 6. Pokaznik dolazi sa ugrađenim mikroupravljačem ESP32, te je kompatibilan sa Arduino razvojnim okruženjem. Mikroupravljač pruža mogućnost spajanja pomoću WiFi-a ili Bluetooth-a. Već postoji programska podrška za postavljanje slike na pokaznik pri spajanju na internet i preuzimanju slike sa zadane mrežne poveznice. Dok je WiFi vrlo raširena i dobro dokumentirana tehnologija, pri radu koristi velike količine struje [11], [12]. Specifično na ESP32 mikroupravljaču, pri korištenju WiFi-a u trenucima može doći i do struje reda 1 A. To može stvoriti probleme pri korištenju baterija malih izlaznih struja, koje se često koriste u sustavima niske potrošnje [13].

Cijelo sklopovlje i programska podrška su otvorenog kôda (eng. *open source*). Mikroupravljač koristi određene izlaze samo za kontroliranje pokaznika, a ostale nožice su izvedene, kako bi se mogle koristiti za druge stvari. Prikaz izvedenih nožica vidi se na slici 3.1. Neke od tih nožica su korištene da se uspostavi komunikacija sa Arduino prijemnikom. Neke značajke pokaznika su: rezolucija od 800x600 piksela sa podrškom za nijanse sive boje (eng. *grayscale*), mogućnost stavljanja baterije, sa ugrađenim punjačem za baterije, microSD čitač kartica, RTC (eng. *Real Time Clock* – sat stvarnog vremena) [13].



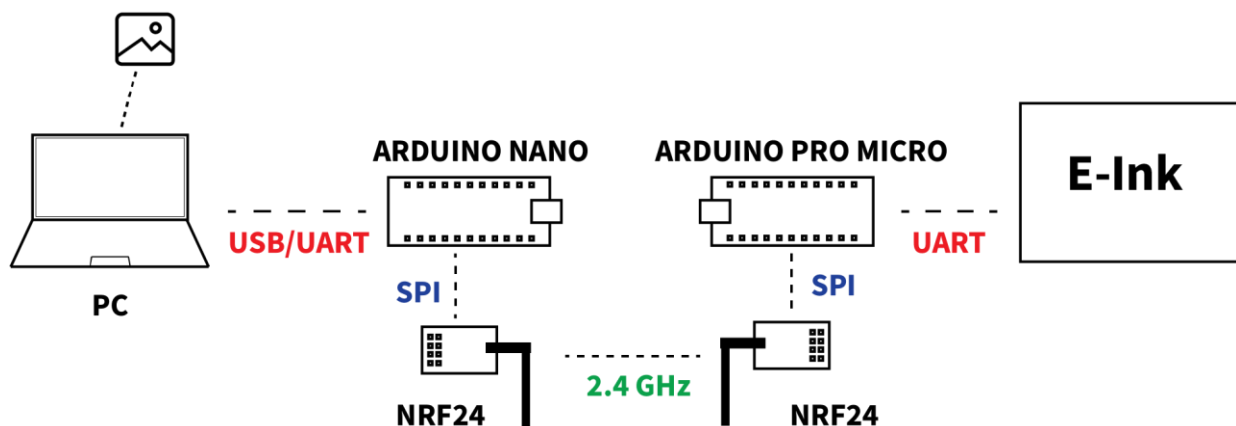
Sl. 3.1 Inkplate 6 izvedene nožice mikrokontrolera

3.1.4. Izrada električne sheme

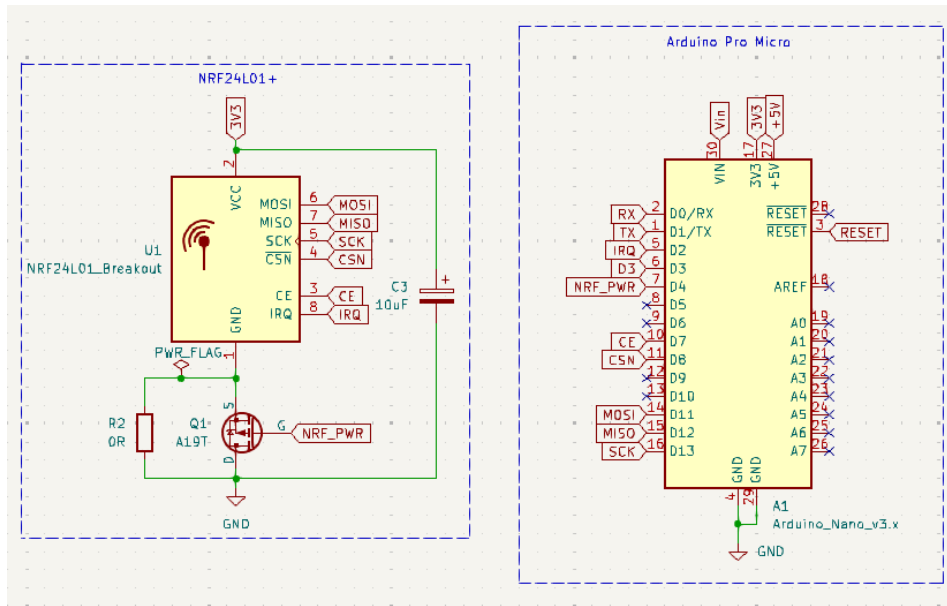
Shema i tiskana pločica izrađene su programskim alatom KiCAD 8 [14]. Razvoj sheme KiCAD alatom odvija se u *Schematic Editor-u*. Izrađene su sheme za odašiljač i prijemnik. Pregled povezanosti cijelog sustava može se vidjeti na slici 3.2.

Vidljivo je da je na strani odašiljača potrebno uspostaviti povezanost između računala i Arduina, te između Arduina i NRF24 modula. Povezanost sa računalom se ostvaruje pomoću USB konektora, pa nije potrebno izvoditi nožice na pločici, a povezanost sa NRF24 se ostvaruje pomoću SPI sučelja. SPI sučelje se sastoji od 4 signala: CS (eng. *Chip Select*), SCK (eng. *Serial Clock*), MISO (eng. *Master In Slave Out*), MOSI (eng. *Master Out Slave In*), te je za kontroliranje NRF24 potreban i CE (eng. *Chip Enable*) signal kojim se kontrolira slanje i primanje paketa. Prema tome, napravljena je shema na slici 3.3.

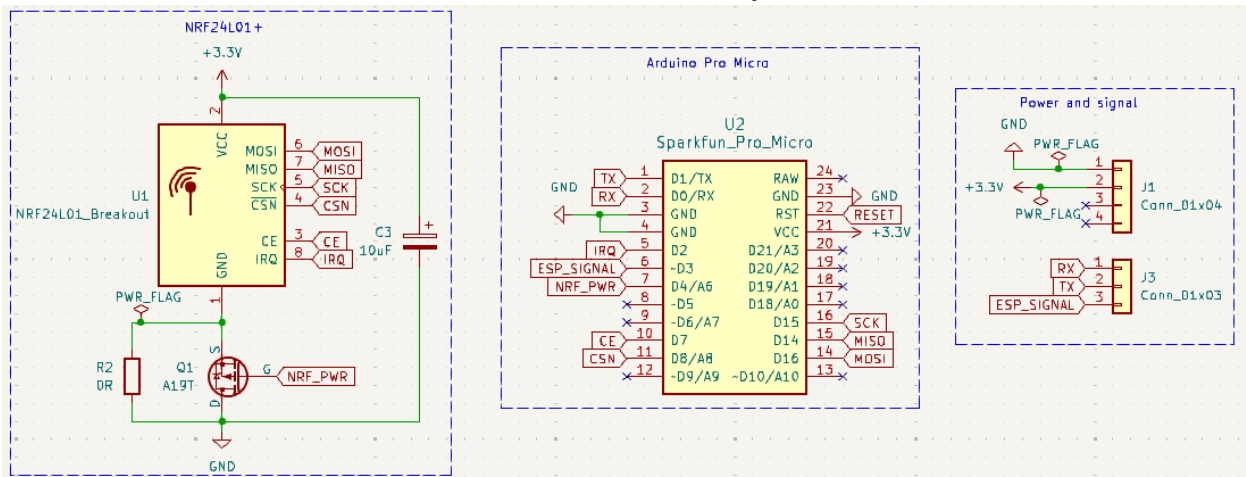
Na strani prijemnika također je spojen Arduino sa NRF24 modulom, jedina razlika je što se koristi Pro Micro umjesto Nano-a, pa se koriste druge nožice. U ovom slučaju se koristi i tranzistor pomoću kojeg Arduino može potpuno isključiti NRF24, kako bi se uštedilo što više energije. Paralelno sa tranzistorom je dodan otpornik od $0\ \Omega$, u slučaju da želi izbjeći korištenje tranzistora. Za povezivanje Arduina sa Inkplate pokaznikom koristi se UART (eng. *Universal Asynchronous Receiver-Transmitter* – Univerzalni asinkroni prijemnik-odašiljač) protokol, za koji su potrebne dvije nožice: TX (eng. *Transmitter pin*) i RX (eng. *Receiver pin*). Također, za vrijeme kada se ne prenose podaci, Inkplate će biti u modu spavanja, pa je potrebno i koristiti jednu nožicu pomoću koje će Arduino probuditi Inkplate, odnosno ESP32 mikroupravljač. Slika 3.3 prikazuje shemu povezivanja Arduina, NRF24 i konektora, pomoću kojeg je prijemnik spojen sa pokaznikom.



Sl. 3.2 Prikaz potpunog toka podataka



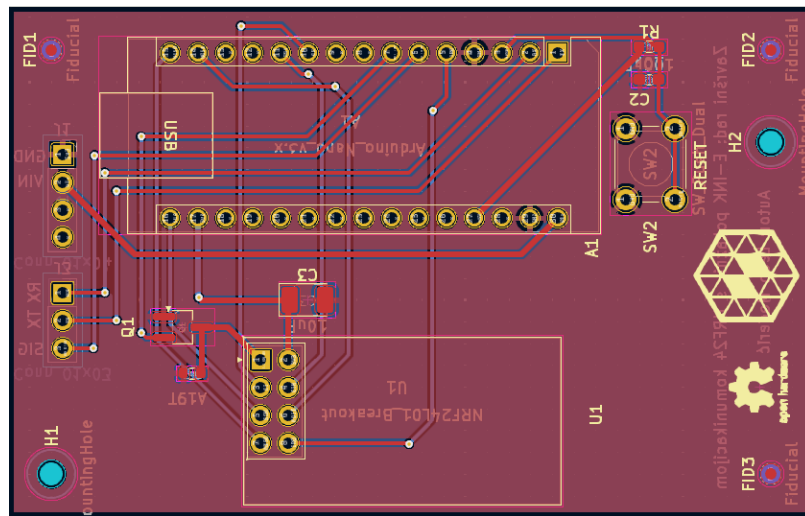
Sl. 3.4 KiCAD shema odašiljača



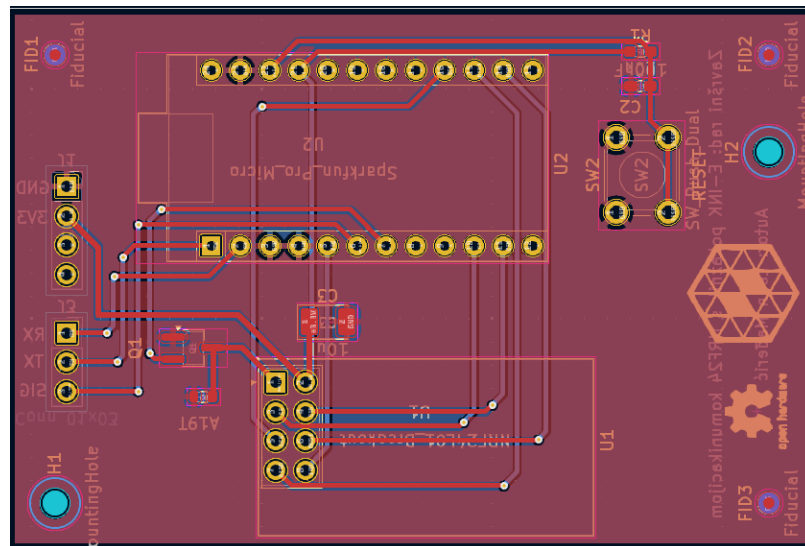
Sl. 3.3 KiCAD shema prijemnika

3.1.5. Tiskana pločica

Tiskana pločica ili PCB (eng. *Printed Circuit Board*) je također izrađena u KiCAD programskom paketu, otvaranjem *PCB Editor* prozora. U ovom koraku izrađujemo stvarni oblik pločice, a time se pridjeljuje svakoj komponenti određeni otisak (eng. *footprint*) veličine koju ta komponenta stvarno zauzima. Pri izradi pločice važno je provjeriti jesu li veličine izabranih otisaka jednake kao i stvarne komponente. Također je važno paziti na električne greške, kao neželjeni spojevi između vodiča, nedovoljna širina vodiča, i slično. Izrađene su pločice za odašiljač i prijemnik. Na slikama 3.5 i 3.6 mogu se vidjeti gotove pločice odašiljača i prijemnika.

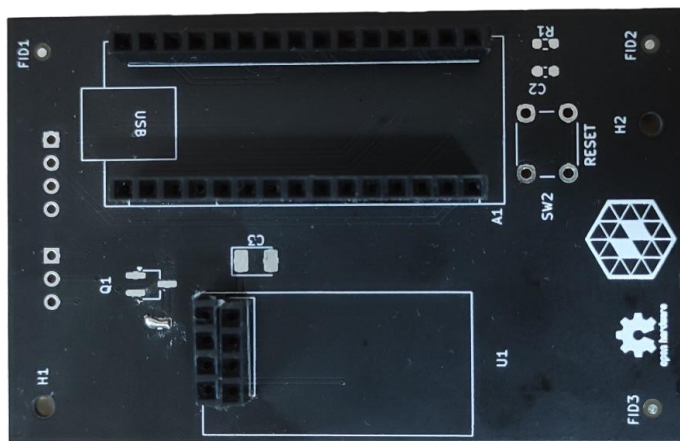


Sl. 3.5 Tiskana pločica odašiljača

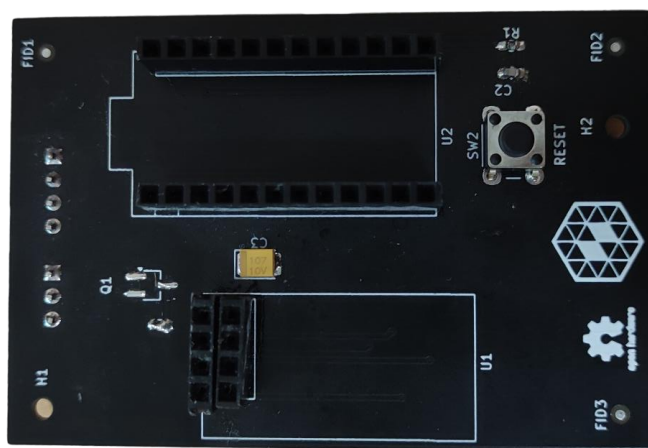


Sl. 3.6 Tiskana pločica prijemnika

Izrađene tiskane pločice, sa zalemljenim komponentama su prikazane na slikama 3.8 i 3.7.



Sl. 3.8 Izrađena tiskana pločica odašiljača



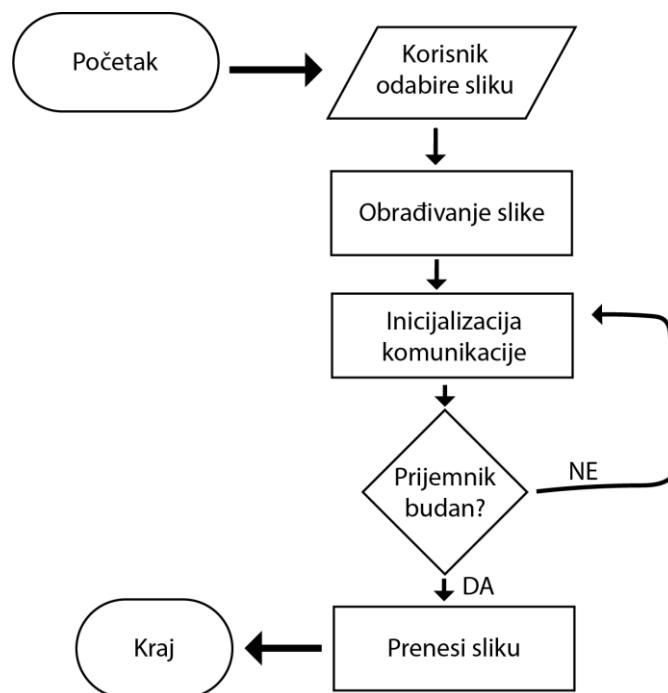
Sl. 3.7 Izrađena tiskana pločica prijemnika

3.2. Programska podrška

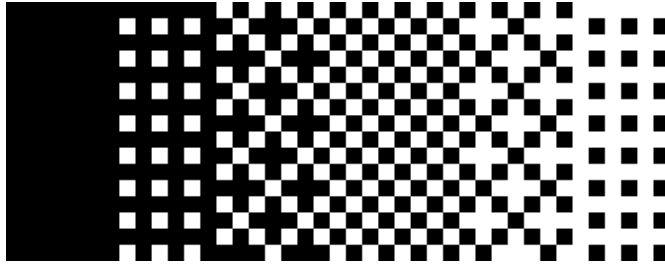
Za izradu sustava komunikacije potrebno je 4 različita programska kôda. Prvo je potreban kôd na računalu, koji učitava i priprema sliku ili neke druge podatke za prijenos, zatim kôd na Arduino odašiljaču i prijemniku, te konačno kôd na pokazniku koji prima podatke i prikazuje ih na zaslonu. Rad sustava je prikazan slikom 3.9. Korisnik jedino odabire sliku, a ostalo se sve odvija bez utjecaja korisnika. Sliku je potrebno obraditi, te se nakon toga pokušava pokrenuti komunikacija. Prijemnik nije uvijek aktivan, nego se izmjenjuje između stanja rada i spavanja, sa ciljem očuvanja energije. Radi toga, računalo je postavljeno kao server koji neprestano ispituje stanje prijemnika, te ako dobije potvrdu od prijemnika da je aktivan, komunikacija započinje. Nakon prenesene slike, ona se prikazuje na pokazniku, te se pokaznik i prijemnik vraćaju u stanje spavanja. Postupci su detaljnije objašnjeni u sljedećim poglavljima.

3.2.1. Obrada slike

Programski kôd na računalu je napisan programskim jezikom C#. Korištena biblioteka podržava učitavanje raznih formata slike, te ih može spremiti na razne načine. U ovom radu, slike su učitane u formatu RGBA32 (eng. *Red-Green-Blue-Alpha 32*), gdje broj 32 označava ukupni broj bitova kojih jedan piksel slike zauzima. Svakoj osnovnoj boji (crvena, zelena, plava) je zadano 8 bita, odnosno 1 bajt. Zadnja vrijednost *alpha* predstavlja prozirnost piksela i ona se neće koristiti pri



Sl. 3.9 Algoritam rada sustava

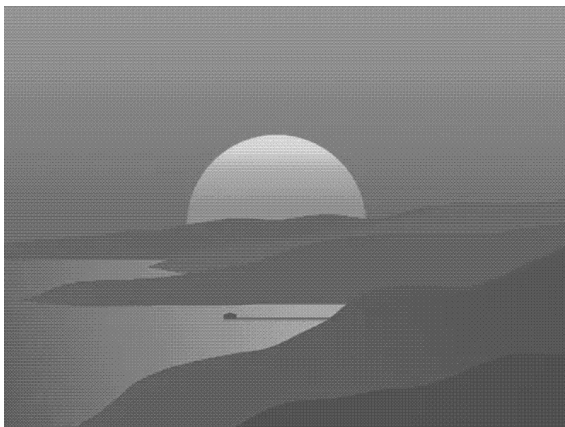


Sl. 3.10 Primjer dither uzoraka, od najtamnije do najsvjetlije boje

prikazivanju slike. Inkplate pokaznik podupire crnu, bijelu i nijanse sive, pa je prvo potrebno pretvoriti boje na izabranoj slici u samo nijanse sive. To se na razini piksela može postići tako da za vrijednost svake boje zapišemo prosječnu vrijednost svih boja tog piksela.

$$R_{nova} = G_{nova} = B_{nova} = \frac{R + G + B}{3} \quad (3-1)$$

Ako formulu 3-1 primijenimo na svakom pikselu na slici, slika će sadržati samo nijanse sive boje. Sljedeći problem je mala količina nijansi kojih pokaznik zapravo može prikazati. Inkplate može prikazati samo 8 različitih nijansi sive, uključujući bijelu i crnu. Zbog toga, slične boje se neće moći raspoznati, nego će se na kraju prikazati kao ista nijansa sive. Taj problem se može popraviti dodavanjem eng. *dither* [15]. To je tehnika za sjenčanje ili simuliranje boja, odnosno ljudima se može činiti da se na slici nalazi više boja nego što stvarno ima. Osnova rada te tehnike je dodavanje šuma u sliku, a može se koristiti i u video ili zvučnim datotekama, te tiskanim medijima kao slike u novinama i stripovima. Primjer uzoraka dither-a sa različitim percipiranim razinama svjetlosti može se vidjeti na slici 3.10. Korištenjem različitih razina dither-a možemo dobiti glatki prijelaz između različitih svjetlina boje, bez da zapravo prikazujemo nove boje. Na slici 3.11 dan je primjer korištenja dither uzorka. Za primjer su uzete slike sa nijansama sive i rezolucijom od 3 bita, odnosno slika koja može prikazati 8 različitih razine sive boje, uključujući bijelu i crnu, jednako



(a)

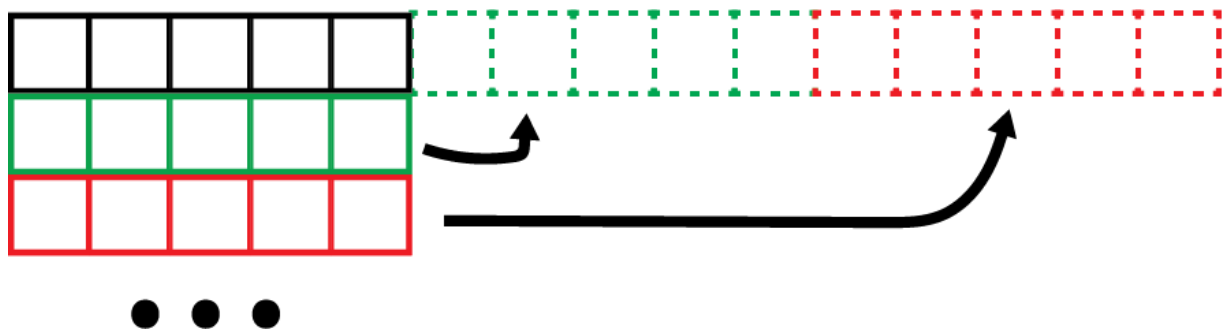


(b)

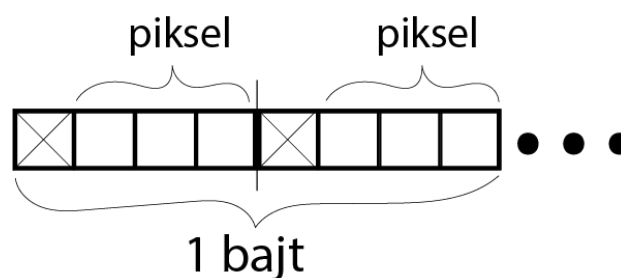
Sl. 3.11 Primjer korištenja dither uzorka (a) slika sa dither-om, (b) slika bez dither-a

kao i Inkplate pokaznik. Na slici bez dithera ograničeni broj boja postaje očit, te je teško prepoznati izgled izvorne slike. Dodavanjem dithera na slici se daje dojam kao da se koristi više boja, ali zapravo je napravljena sa istim bojama i lakše raspoznati izvornu sliku.

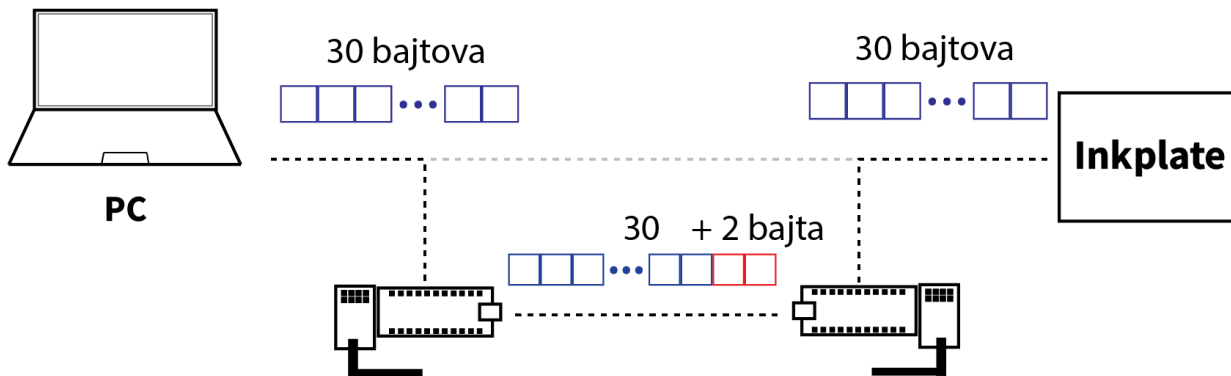
Slika je u ovom koraku spremna za prikaz na Inkplate pokazniku, te je sljedeći korak prijenos slike. Jedno ograničenje NRF24 je mogućnost slanja samo 32 bajta u jednom paketu. To znači da nije moguće poslati cijelu sliku odjednom, nego je potrebno podijeliti sliku u manje dijelove, te ponovno spojiti te dijelove na strani prijemnika. Sliku možemo gledati kao matricu piksela, a tako je i spremljena u memoriji u korištenoj biblioteci. Da bi prenesli sliku, ona se može zapisati kao niz piksela, gdje će se u svakom paketu prenijeti jedan dio tog niza. Matrica je pretvorena u niz tako da se svaki red matrice nadovezuje na kraj prošlog reda, kao što je vidljivo na slici 3.12. Radi prikazivanja slike na pokazniku i bržeg prijenosa slike, svaki piksel je potrebno pretvoriti u 3-bitnu vrijednost, za razliku od trenutnog prikazivanja slike sa 32 bita za RGBA vrijednosti. Vrijednost *alpha* nije potrebna, a nijanse sive se mogu zapisati samo jednom vrijednosti, jer zapisivanje crvene, zelene i plave, koje sadrže jednake vrijednosti je redundantno. Tako, dobivenu nijansu sive možemo zapisati uzimanjem jedne vrijednosti, R, G ili B vrijednosti. Time dobivamo nijansu sive zapisanu jednim bajtom, odnosno vrijednosti 0-255, gdje je vrijednost 0 crna boja, a 255 bijela boja. Skaliranjem te vrijednosti na raspon 0-7 (3 bita) se smanjuje vrijednost jednog piksela na



Sl. 3.12 Pretvaranje matrice slike u niz



Sl. 3.13 Spremanje piksela pri prijenosu



Sl. 3.14 Primjer slanja paketa

željenih 3 bita. Zatim se u svaki bajt može zapisati dvije vrijednosti piksela, gdje dva bita ostaju redundantna. Vrijednosti piksela će biti zapisane kao na slici 3.13.

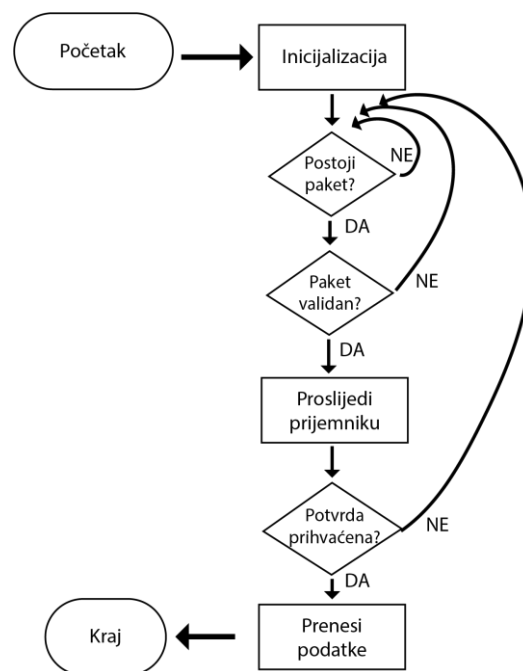
3.2.2. Komunikacija

Tijekom komunikacije sudjeluju članovi prikazani na slici 3.2. Komunikacija je napravljena tako da je sloj prijenosa slike sa računala neovisan o sloju prijenosa podataka preko Arduina uparenih NRF24 modulom. Razlog razdvajanja slojeva je ta da se stvori neovisnost odašiljača i prijemnika o vrsti podataka koji se šalju. Arduino odašiljač i prijemnik znaju samo veličinu datoteke koja se prijenosi, odnosno broj bajtova, a ne i njenu strukturu. Primjer slanja paketa može se vidjeti na slici 3.14. U nižem sloju se dodaje brojač paketa, radi kontrole toka podataka, dok računalo i pokaznik vide samo podatke koje oni komuniciraju.

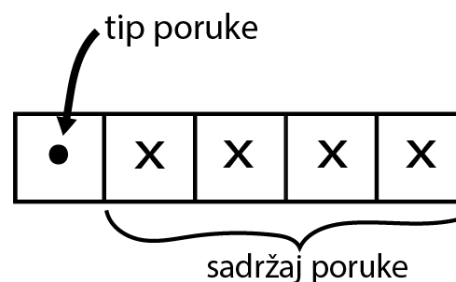
Na Arduino odašiljaču i prijemniku je korištena biblioteka RF24 [16]. Dok postoje i druge biblioteke, one nisu kvalitetno dokumentirane i ne pružaju dovoljnu razinu fleksibilnosti korisniku. Prvi korak za uspostavljanje komunikacije preko NRF24 je postavljanje integriranog sklopa. NRF24 pruža postavljanje različitih postavki, razina jačine prijenosa, brzine prijenosa, kanala (točne frekvencije preko koje se prijenosi) i druge. Još važno napomenuti postavljanje u stanje slanja ili primanja, jer nema mogućnost slanja i primanja podataka u isto vrijeme. Za potrebe ovog rada, korištena je maksimalna brzina i jačina prijenosa, radi boljeg dometa i kraćeg vremena slanja. Uz postavljanje NRF24, u *setup* dijelu kôda, još je potrebno i uspostaviti komunikaciju sa računalom na strani odašiljača, odnosno pokaznikom na strani prijemnika. Arduino i NRF24 u ovom stanju su spremni za prijenos, te je potrebno odrediti algoritam kojim će se prenositi i verificirati paketi. Algoritam za inicijalizaciju komunikacije na Arduino odašiljaču dan je slikom 3.15. U kontekstu tog algoritma, paket označava inicijalnu poruku obavijesti koja predstavlja tip komunikacije. Tim paketom odašiljač i prijemnik znaju na koji način i koliko podataka prijenose. Izgled općenite poruke obavijesti se može vidjeti na slici 3.16. Poruka se sastoji od ukupno 5 bajta,

gdje prvi označava tip poruke, a ostali su rezervirani za sadržaj te poruke, ovisno o kontekstu i tipu poruke koji se koristi. Tipovi poruke koji se mogu pojaviti i njihove odgovarajuće heksadecimalne vrijednosti su sljedeće:

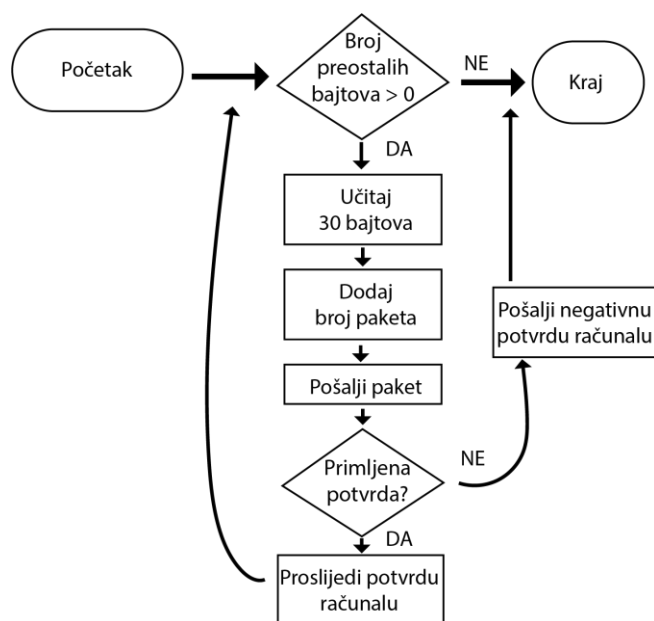
- **prijenos bajtova 0x01** – sadržaj poruke označava broj bajtova koje treba prenijeti
- **prijenos bajtova uz buđenje prijemnika 0x02** – prije početka komunikacije, prijemnik budi uređaj koji prima podatke, a sadržaj poruke označava broj bajtova koje treba prenijeti
- **prijenos *string* tipa podatka 0x03** – koristi se za ispisivanje poruka obavijesti na računalo
- **poruka potvrde** (eng. *ACK - acknowledge*) **0xFF** – koristi se za potvrdu uspješno izvedene operacije, a sadržaj poruke ovisi o kontekstu u kojem se koristi
- **poruka negativne potvrde** (eng. *NAK – negative acknowledge*) **0x00** – koristi se za označavanje greške u komunikaciji, sadržaj ovisi o kontekstu.



Sl. 3.15 Algoritam slanja inicijalnog paketa



Sl. 3.16 Izgled paketa obavijesti



Sl. 3.17 Algoritam prijena podataka na strani odašiljača

Odašiljač kontinuirano provjerava dolazeće poruke od strane računala, te pri primanju poruke, provjerava sadržaj i tip poruke koji je poslan. Ukoliko je primljena poruka točne strukture, komunikacija se može nastaviti. Poruku prosljeđuje prijemu i ukoliko je poruka točno primljena, prijemnik odgovara sa porukom potvrde (ACK) i kopiranim sadržajem iz primljene poruke. Sa time je potvrđena inicijalizacija komunikacije, te može započeti prijenos podataka.

Prijenos podataka ponavlja jednaki postupak učitavanja i slanja podataka sve dok prijenos podataka ne dođe do kraja, odnosno dok se ne pošalju svi podaci. Taj postupak je definiran algoritmom prikazanim na slici 3.17. Taj algoritam predstavlja funkciju *Prenesi podatke* prikazanu na slici 3.15. Jedini ulazni parametar funkcije je broj bajtova sadržaja kojeg se prenosi. Glavni dio funkcije sastoji se od *while* petlje koja u svakoj iteraciji prenosi maksimalno 30 bajtova, te se zaustavlja kada su svi poslani. Razlog slanja 30 bajtova je limit NRF24 modula da prenese 32 bajta u jednom paketu, a 2 bajta se ostavljaju za brojač paketa, koji se koristi za provjeru propuštenih paketa i sličnih problema toka podataka. Brojač se povećava za jedan pri slanju svakog paketa, pa je maksimalni broj paketa koji se može poslati u jednom prijenosu 2^{16} , odnosno 65 536, što znači da uz paket veličine 30 bajta, ukupna veličina datoteke koja se može poslati u jednom prijenosu je: $30 * 2^{16} \approx 2 \text{ MB}$. Nakon dodavanja brojača paket se šalje prijemu i odašiljač čeka potvrdu poruku od prijemnika da je poruka uspješno stigla. Ukoliko je poruka potvrđena, odašiljač to prosljeđuje računalu i računalo šalje novih 30 bajtova za slanje.

Postoji nekoliko grešaka koje se mogu dogoditi prilikom slanja paketa. Paket može "pasti", odnosno on je poslan, ali ne stiže do prijemnika. U tom slučaju odašiljač pokušava ponovno poslati paket sve dok ne istekne unaprijed definirano maksimalno vrijeme slanja. U slučaju da je paket

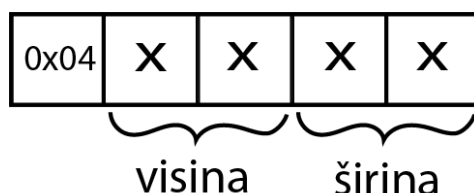
stigao do prijemnika, ali nije uspješno slanje potvrdnog paketa, odašiljač također pokušava ponovno poslati paket, sve dok ne dobije potvrdu o točno poslanom paketu. Dolazi do problema da prijemnik čeka sljedeći paket, a odašiljač još uvijek šalje stari. U tom slučaju pomaže brojač paketa. Ako prijemnik vidi da se šalje stari paket, on ponovno šalje potvrdu za taj paket, te nastavlja čekati novi paket. Nakon uspješnog slanja zadnjeg paketa, komunikacija se prekida i odašiljač i prijemnik su spremni za ponovno slanje.

Na višem sloju se događa komunikacija između računala i Inkplate pokaznika. Ovdje se komunikacija također odvija u dijelovima od 30 bajta, ali bez provjere integriteta i potvrđivanja paketa. Na ovom sloju je potrebna i mogućnost slanja različitih vrsta paketa, pa je ponovno potrebno definirati pakete obavijesti. Oni imaju jednak oblik kao paketi obavijesti za komunikaciju nižeg sloja sa slike 3.16, ali sa drugim oznakama za tip poruke i drugim značenjima poruke. Definirani su sljedeći tipovi poruke:

- **prijenos bajtova 0x01** – primarno radi testiranja, ne služi za prikazivanje slike
- **prijenos slike 0x02** – prijenos slike u RGB formatu, bez prethodne obrade na računalo. Obrada slike se odrađuje na mikroupravljaču pokaznika
- **prijenos *string* tipa podatka 0x03** – također primarno radi testiranja
- **prijenos slike u 3-bitnom formatu 0x04** – prijenos slike sa nijansama sive, značajno manja količina podataka za prijenos i preusmjeri računski teže operacije na računalo.

Sadržaj poruke za prijenos bajtova ima jednaku strukturu kao i kod Arduino prijenosnika. Sadržaj poruke za slanje slike u 3-bitnom formatu se sastoji od 2 dijela. Pomoću samo duljine podataka nije moguće saznati dimenzije slike, pa se pri prijenosu slike 2 bajta koriste za definiranje visine, a druga 2 za definiranje širine slike, kao na slici 3.18. Prilikom prijensa, slika se gleda kao niz bajtova. To znači da je algoritam prijensa podataka sličan kao i kod Arduino primopredajnika, uz razliku veličine paketa (30 bajta umjesto 32) i brojača za pakete. Pokaznik može jednostavno izračunati ukupni broj bajtova koji će primiti pomoću formule 3-2.

$$n = \frac{\text{visina} * \text{širina}}{2} \quad (3-2)$$



Sl. 3.18 Poruka obavijesti za 3-bitnu sliku

Računanjem $visina * širina$ dobivamo ukupni broj piksela od kojeg se slika sastoji, a uzimajući u obzir da se u svakom bajtu nalazi 2 piksela, radi korištenja 3-bitnih piksela, to znači da je potrebno prenijeti duplo manje bajtova nego piksela.

Prikazivanje slike na Inkplate pokazniku se odvija u dva koraka. Prvo je potrebno promijeniti vrijednost unutarnjeg stanja piksela na željenu vrijednost, a drugi korak je prikazivanje tog stanja na pokazniku. Ovo je napravljeno da bi se izbjeglo često osvježavanje zaslona pri svakoj promjeni, koje je relativno sporo u usporedbi sa tradicionalnim LCD ili LED zaslonima. Prema tome, pri svakom primljenom paketu, potrebno je postaviti vrijednosti piksela u unutarnjem spremniku. Lokacija i vrijednosti piksela se postavljaju prema algoritmu na slici 3.19.

Linija Kôd

```

1:        Za i od 0 do duljina_paketa
2:        indeks_piksela = ukupni_bajtova - broj_preostalih + i
3:        X = (indeks_piksela * 2) mod širina
4:        Y = (indeks_piksela * 2) / širina
5:        boja_piksela = paket[i] >> 4 // bit-wise pomak u desno
6:        nacrtaj_piksel(X, Y, boja_piksela) // prvi piksel
7:
8:        X = (indeks_piksela * 2 + 1) mod širina
9:        Y = (indeks_piksela * 2 + 1) / širina
10:       boja_piksela = paket[i] & 0x0F // uzimanje druge polovice bajta
11:       nacrtaj_piksel(X, Y, boja_piksela) // drugi piksel

```

Sl. 3.19. Algoritam postavljanja piksela

Na kraju prijenosa, prenesena slika se prikazuje na pokazniku.

Postupak slanja slike sa računala radi se sljedećim redoslijedom. Pri svakom slanju podataka pokazniku, potrebno je i inicijalizirati komunikaciju na nižem sloju. Da bi se uspostavila i započela komunikacija između računala i pokaznika potrebno je prvo poslati paket obavijesti, sa informacijom o vrsti podataka koji se šalju, zatim pakete koji sadrže podatke koji se stvarno šalju. Na kraju računalo mora poslati 4 različite poruke.

1. Inicijalizacijski paket obavijesti koji govori odašiljaču da se prenosi 5 bajtova.
2. Inicijalizacijski paket obavijesti (duljine 5 bajtova) koji govori pokazniku da se prenosi slika.
3. Inicijalizacijski paket obavijesti koji govori odašiljaču da se prenosi broj bajtova jednak veličini slike.
4. Prijenos slike od računala do pokaznika.

3.2.3. Smanjenje potrošnje

Inkplate pokaznik ima mogućnost napajanja pomoću baterije, radi korištenja na udaljenim lokacijama. Arduino prijemnik je također spojen na način da koristi napajanje pokaznika, pa mogu zajedno funkcionirati bez konstantnog izvora napajanja. U tu svrhu, radi smanjena potrošnje energije, kako bi baterija trajala što duže, koristi se način rada spavanja (eng. *sleep mode*). Arduino Pro Micro, odnosno ATmega32u4 mikroupravljač, te ESP32 mikroupravljač sadrže različite načine spavanja, pa je potrebno svakog postaviti zasebno.

Komunikacija radi tako da se prijemnik probudi u određenim intervalima, te provjeri ako je dostupan paket za slanje od strane odašiljača. Ukoliko postoji paket, prijemnik će probuditi pokaznik, te će se ostatak komunikacije odvijati kao i prije.

ATMega mikroupravljač sadrži 6 različitih stanja spavanja [17]. Buđenje iz bilo kojeg stanja ne resetira mikroupravljač, nego nastavlja od zadnje instrukcije, osim ako je probuđen pomoću *reset* pina ili gumba. Stanja spavanja su dana u tablici Tablica 3.1.

Tablica 3.1 Modovi spavanja ATMega32u4

<i>Idle</i>	mikroprocesor je zaustavljen, dok svi ostali periferni uređaji nastave raditi, buđenje je omogućeno bilo kojim internim ili eksternim prekidnim (eng. <i>interrupt</i>) signalima
<i>ADC Noise Reduction</i>	kao u <i>idle</i> načinu rada, mikroprocesor je zaustavljen, uz dodatno isključivanje clkI/O signala takta za ulazno/izlazne pinove. Ovaj način rada je namijenjen za točnije čitanje sa ADC (eng. <i>Analog to Digital Converter</i> – Analogni na digitalni pretvarač) pretvarača. Većina prekida ostaje omogućena, uz neke iznimke.
<i>Power-down</i>	zaustavlja sve generirane signale takta, te ostavlja omogućene samo asinkrone module. To je najveće moguće stanje spavanja, a buđenje je omogućeno samo asinkronim modulima kao <i>Two-wire</i> serijsko sučelje pomoću podudaranja adrese, te <i>Watchdog reset</i> prekidom.

<i>Power-save</i>	radi kompatibilnosti sa drugim mikroprocesorima, jednako kao <i>power-down</i>
<i>Standby</i>	identično <i>power-down</i> modu, uz razliku omogućenog oscilatora
<i>Extended Standby</i>	radi kompatibilnosti sa drugim mikroprocesorima, jednako kao <i>standby</i>

U radu nije potrebna aktivnost drugih modula, te se može koristiti stanje spavanja *power-down*. Pri tome, za periodično buđenje se koristi *watchdog timer*. Na ovom mikroupravljaču, *watchdog timer* ima maksimalno vrijeme čekanja od 8 s. Period spavanja od 8 s, zatim provjeravanje dostupnih paketa nije povoljno, jer će mikroupravljač znatnu količinu vremena provesti u uključenom stanju. Zato je potrebno napraviti brojač, koji će se, svaki put kada se mikroupravljač probudi iz stanja spavanja, povećati za 1. Time je moguće postaviti period spavanja u inkrementima od 8 s. Odnosno, ako postavimo da brojač broji do 1, vrijeme spavanja će biti 8 s, ako postavimo na 2, 16 s, i tako dalje. U tom slučaju, mikroupravljač je uključen na vrlo kratko vrijeme, samo da bi povećao brojač, što ne predstavlja veliku potrošnju. Potrošnja se nadalje može smanjiti isključivanjem pojedinih modula, kao eng. *Analog Comparator*, eng. *Brown-out Detector* i slično.

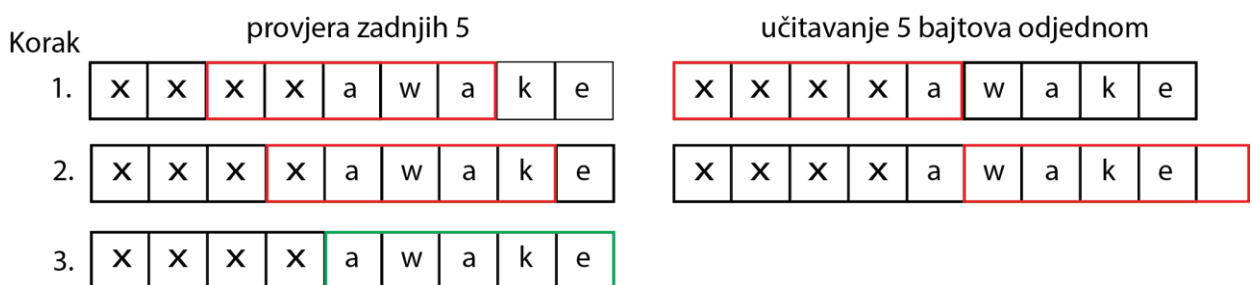
ESP32 ne nudi razne unaprijed definirane načine spavanja, nego razlikuje samo eng. *light-sleep* i eng. *deep-sleep*, te korisniku ostavlja slobodu uključivanja i isključivanja pojedinih komponenti. *Light sleep* uglavnom usporava takt procesora i ostalih modula, s time da većina modula ostaje funkcionirati. *Deep sleep* isključuje sve najveće potrošače energije, kao CPU, RAM i druge periferne module. Dostupni su razni načini buđenja iz spavanja, ovisno o željenoj funkcionalnosti [18].

Tablica 3.2 Načini spavanja ESP32

<i>Timer</i>	buđenje pomoću brojača, kojeg kontrolira RTC upravljač
<i>Touch pad</i>	buđenje na dodir, moraju se koristiti definirane nožice koje imaju mogućnost aktiviranja na dodir
<i>External wakeup (ext0)</i>	koristi RTC IO za detektiranje postavljanja određene RTC GPIO nožice na neku logičku razinu, 0 ili 1. Radi korištenja RTC nožica, ostali RTC periferni uređaji moraju također ostati uključeni
<i>External wakeup (ext1)</i>	koristi RTC kontroler za detektiranje logičkog stanja. Korištenje RTC kontrolera umjesto RTC IO omogućuje isključivanje ostalih

	RTC periferija. Ali to znači da se ne mogu koristiti niti <i>pullup</i> i <i>pulldown</i> otpornici
<i>ULT coprocessor wakeup</i>	može pratiti i probuditi glavni procesor uslijed određenih događaja, kao i vrijednosti na dodirnim sensorima
<i>GPIO wakeup</i>	do buđenja dolazi uslijed određenog događaja na postavljenoj ulazno/izlaznoj nožici. Može se koristiti samo u <i>light</i> načinu spavanja
<i>UART wakeup</i>	buđenje se događa uslijed aktivnosti na UART komunikaciji. Može se koristiti samo u <i>light</i> načinu spavanja

Za svrhe rada, korist će se buđenje pomoću ext0 prekida, jer je potrebno buđenje na signal Arduino prijemnika. ESP32 je postavljen na *deep sleep* način spavanja, te da se probudi pri visokom logičkom signalu. Pri korištenju tog načina spavanja, ESP32 ne pamti zadnje stanje, nego se na pojavu prekida resetira. To znači da je, prije primanja poruka potrebno neko vrijeme da se ESP32 potpuno postavi i probudi. Kako bi znali kada je on spreman za rad, pri završetku buđenja i postavljanja, poslati će poruku "awake". U postupku buđenja, izlazi još nisu potpuno postavljeni, pa je moguće da će se na UART komunikaciji pojaviti slučajne i neželjene poruke. Radi toga, prijemnik ne čeka da se u komunikaciji pojavi 5 bajtova (duljina poruke "awake"), nego kontinuirano uzima bajt po bajt i provjerava da su zadnjih 5 bajtova jednaki poruci za buđenje. Taj problem je prikazan na slici 3.20, gdje se može primijetiti da učitavanjem 5 bajtova odjednom može doći do preskakanja poruke buđenja.



Sl. 3.20 Provjera poruke buđenja. X predstavlja šum u komunikaciji pri buđenju

3.3. Testiranje

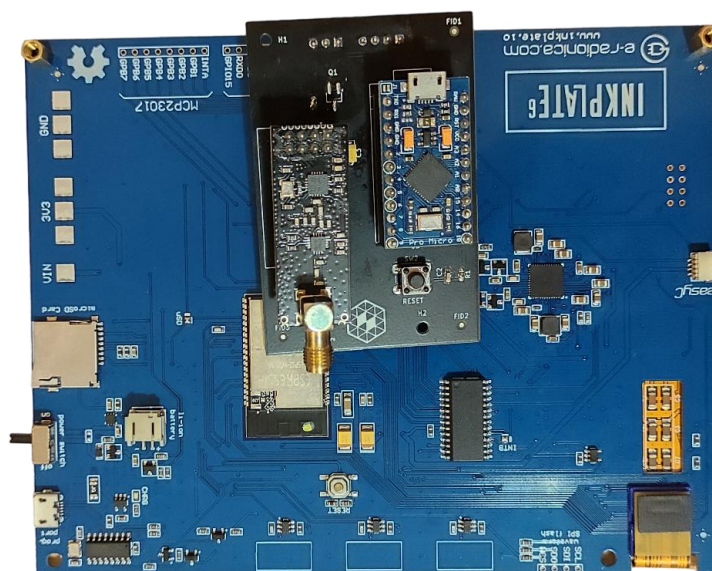
Testiranje komunikacije i prijenosa podataka izvedeno je ispisivanjem poruka na zaslon računala. Kako bi se mogle ispisati poruke o otklanjanju grešaka (engl. *debug message*) potrebno je koristiti drugi komunikacijski kanal ili isti kanal, ali dodati mogućnost razlikovanja poruka koje se šalju pri komunikaciji i poruke za otklanjanje grešaka. Arduino odašiljač je već spojen sa računalom preko USB konekcije, pa je jednostavnije implementirati razlikovanje poruka normalne komunikacije i poruka grešaka. Prijemnik je spojen na pokaznik, koji ne treba dobivati informacije o statusu komunikacije, nego samo podatke, pa je bolje koristiti drugi komunikacijski kanal (USB priključak, SPI ili slično) preko kojeg će se slati poruke računalu.

Računalo neprestano čeka nadolazeće poruke od odašiljača. Različite vrste poruka se raspoznaju prema već spomenutim oznakama za tip poruke. Na dolaznim porukama sa odašiljača je glavno raspoznati poruke potvrde i poruke za otklanjanje grešaka. U slučaju dolaska poruke potvrde, ona se sprema u listu, te se komunikacija nastavlja, a pri dolasku poruke za otklanjanje grešaka, ona se ispisuje na konzolu. Kao oznaka poruke za otklanjanje grešaka koristi se string tip poruke. Za razliku od drugih tipova poruka obavijesti, pri slanju string tipa podatka, u poruci obavijesti se ne predaju drugi parametri osim tipa poruke (odnosno string tip). Kraj dolazećeg paketa je označen znakom za novi red (engl. *newline character*). To također znači da odašiljač ne treba računati duljinu poruke koju šalje, ali nije moguće poslati poruku u više redova, bez slanja više paketa obavijesti. Izgled sklopovlja pri testiranju može se vidjeti na slici 3.21, te slika 3.22 prikazuje način spajanja prijemnika sa Inplate zaslonom.

Poruke sa prijemnika je moguće čitati preko USB konekcije i bilo kojeg programa za serijsku komunikaciju, pa tako i Arduino IDE. Moguće se i spojiti na programski definirane nožice za



Sl. 3.21 Sklopovlje spojeno sa računalom pri testiranju



Sl. 3.22 Arduino prijemnik spojen na Inkplate pokaznik

UART komunikaciju, te prevesti poruke u USB komunikaciju pomoću UART na USB pretvarača. Razlog korištenja programski definiranog UART-a su problemi sa prekidom USB komunikacije pri uključivanju moda spavanja, pa su implementirane obje metode.

Ispisivanje poruka za otklanjanje grešaka moguće je lagano isključiti ili uključiti korištenjem pretprocesorskih naredbi. Ostavljanje tih poruka uključenim u konačnom radu nepotrebno može usporavati komunikaciju i zauzimati dodatne resurse na mikroupravljaču.

Na slici 3.23 vidi se primjer poruke greške kada odašiljač ne primi poruku potvrde buđenja.

U slučaju grešaka pri komunikaciji, svaki sudionik komunikacije pokušava ponovno emitirati ili primiti poruku unutar nekog određenog vremena, nakon kojeg se komunikacija prekida. Nakon prekida, računalu se javlja o pojavi greške, te je sustav ponovno spreman za slanje novih podataka. Vrijeme ponovnog pokušavanja je proizvoljno postavljeno na 1 sekundu, ali to može ovisiti do željene funkcionalnosti. U slučaju na slici 3.23, odašiljač je pokušao poslati početnu poruku obavijesti primatelju, kojemu je cilj probuditi mikroupravljač pokaznika, te vratiti potvrdu. Odašiljač niti nakon 1 sekunde nije dobio potvrdu o buđenju, te se komunikacija prekida i svi mikroupravljači se vraćaju na početno stanje.

```
send 5
Sent: 0x00 0x01 0x02 0x03 0x04
Transmitter: no wake flag ack
```

Sl. 3.23 Primjer poruke greške na odašiljaču

Vrijeme prijenosa slike, bez vremena koje je potrebno za buđenje pokaznika, je između 25 i 40 sekundi. Razlog tolikoj razlici u brzini slanja je do grešaka u komunikaciji, gdje paketi ne stižu do prijemnika, ili potvrda o primanju paketa ne dolazi do odašiljača. U većini slučajeva, vrijeme slanja ne prelazi preko 30 sekundi, pa se to može uzeti kao brzina prijenosa pri normalnoj komunikaciji. Ako izračunamo veličinu slike u bajtima, što je jednako polovici ukupnog broja piksela, pošto se u svakom bajtu nalaze podaci za dva piksela, dobijemo sljedeće:

$$veličina = \frac{visina * širina}{2} = \frac{800 * 600}{2} = 240000 [B] = 240 [kB] \quad (3-3)$$

Sa time možemo izračunati ukupnu prosječnu brzinu prijenosa podataka koja je jednaka:

$$brzina = \frac{240}{30} = 8 [kBps] \quad (3-4)$$

Dobivena brzina je znatno sporija od maksimalne brzine prijenosa NRF24 modula (2 Mbps), a razlog tome su potrebe za potvrdom paketa i komunikacije preko Arduino modula, umjesto direktno sa računalom i pokaznikom. Slanje slike sivih boja sa računalnog programa omogućeno

```
Type something and press Enter to send it to the Arduino:
sending3
0%
1%
2%
3%
4%
```

Sl. 3.24 Primjer slanja slike sa računala



Sl. 3.25 Prikaz poslana slike na pokazniku

je komandom *sendimg3*. Time se učitava slika, obrađuje, te na kraju šalje pokazniku. Slika 3.24 prikazuje izgled sučelja, a na slici 3.25 se vidi konačni rezultat poslane slike.

Problemi u bežičnoj komunikaciji s modulom NRF24 mogu nastati zbog raznih čimbenika koji utječu na prijenos ili potvrdu podataka. Jedan od glavnih razloga za neuspjeh prijenosa ili potvrde mogu biti smetnje na frekvenciji od 2.4 GHz, kao što su drugi uređaji koji koriste isti kanal ili frekvencijski pojas, što uzrokuje kolizije i gubitak podataka. Loša kvaliteta antena ili neadekvatno napajanje modula također mogu pridonijeti problemima u komunikaciji. Način slanja slike ne omogućuje obnovu u slučaju gubitka paketa, stoga je neophodno da svi paketi budu poslani te da postoji sustav za ponovno slanje izgubljenih paketa.

4. ZAKLJUČAK

U ovom završnom radu postavljeni su ciljevi razvoja sustava za prijenos podataka s računala na e-ink zaslon korištenjem Arduino platforme i NRF24 bežičnih komunikacijskih modula. Glavni cilj bio je omogućiti pouzdanu i energetske učinkovitu komunikaciju između dva Arduina, pri čemu jedan uređaj prima podatke s računala, a drugi upravlja e-ink zaslonom za prikaz slike.

U sklopovskom dijelu, projekt je uključivao uspješan dizajn i povezivanje komponenti. Poseban naglasak stavljen je na izradu električne sheme i tiskane pločice. S programske strane, glavni dio je uključivao izradu sustava prijenosa podataka i optimizacija istog, a također su objašnjeni načini obrade slike, prikaz slike, te načini spavanja mikroupravljača.

Budući rad može se usmjeriti na poboljšanje dometa i brzine prijenosa podataka bežične veze, dodavanje podrške za višebojne E-Ink zaslone te razvoj kompleksnijih primjena. Ovaj rad pruža osnovu za buduće upravljanje E-Ink zaslonima bežičnom komunikacijom.

LITERATURA

- [1] S. A., Rabbani, M., Ahmed, A. H., Zahid, „E-Ink; Revolution of Displays“, *MATEC Web Conf.*, sv. 381, str. 02003, 2023.
- [2] „nRF24 Series - Nordic Semiconductor“ [online]. Dostupno na: <https://www.nordicsemi.com/Products/nRF24-series>. [Pristupljeno: 3.9.2024.].
- [3] „nRF24L01 – How It Works, Arduino Interface, Circuits, Codes“ [online]. Dostupno na: <https://howtomechatronics.com/tutorials/arduino/arduino-wireless-communication-nrf24l01-tutorial/>. [Pristupljeno: 3.9.2024.].
- [4] „Amazon Kindle image“ [online]. Dostupno na: https://target.scene7.com/is/image/Target/GUEST_37e35ac8-fdb4-451e-966c-236d0a0c9ae5?wid=1200&hei=1200&qtl=80&fmt=webp. [Pristupljeno: 6.9.2024.].
- [5] „E Ink Products four colors ePaper“ [online]. Dostupno na: <https://www.eink.com/product/detail/EL024TS1>. [Pristupljeno: 6.9.2024.].
- [6] „How to send books to your Kindle — even if they’re not from Amazon - The Verge“ [online]. Dostupno na: <https://www.theverge.com/23842102/kindle-amazon-import-files-epub>. [Pristupljeno: 3.9.2024.].
- [7] „How Do Electronic Price Tags Work – ZhSunyco“ [online]. Dostupno na: <https://www.zhsunyco.com/how-do-electronic-price-tags-work/>. [Pristupljeno: 3.9.2024.].
- [8] „Electronic Ink | E Ink Technology“ [online]. Dostupno na: https://www.eink.com/tech/detail/How_it_works. [Pristupljeno: 3.9.2024.].
- [9] „Nano | Arduino Documentation“ [online]. Dostupno na: <https://docs.arduino.cc/hardware/nano/>. [Pristupljeno: 3.9.2024.].
- [10] „Pro Micro - 5V/16MHz - DEV-12640 - SparkFun Electronics“ [online]. Dostupno na: <https://www.sparkfun.com/products/12640>. [Pristupljeno: 3.9.2024.].
- [11] K., Pothuganti, A., Chitneni, „A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi“, sv. 4, str. 655–662, ruj. 2014.
- [12] F., Cavalcanti, „Power Usage Comparison of Communication Protocols on the ESP32: Wi-Fi, Zigbee, and ESP NOW“ [online], 30-svi-2024. .
- [13] „E-ink Arduino 6" - Soldered Electronics“ [online]. Dostupno na: <https://soldered.com/product/soldered-inkplate-6-6-e-paper-board/>. [Pristupljeno: 3.9.2024.].
- [14] „KiCad EDA - Schematic Capture & PCB Design Software“ [online]. Dostupno na: <https://www.kicad.org/>. [Pristupljeno: 5.9.2024.].
- [15] H., coding academic, „Image Dithering — How to Deceive your Eyes?“ [online], 31-srp-2022. .
- [16] „Optimized high speed nRF24L01+ driver class documentation: Optimized High Speed Driver for nRF24L01(+) 2.4GHz Wireless Transceiver“ [online]. Dostupno na: <https://nrf24.github.io/RF24/>. [Pristupljeno: 3.9.2024.].
- [17] „ATmega16U4/32U4 Datasheet“.
- [18] „Sleep Modes - ESP32 - — ESP-IDF Programming Guide v5.3.1 documentation“ [online]. Dostupno na: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/system/sleep_modes.html. [Pristupljeno: 3.9.2024.].

SAŽETAK

Ovaj rad predstavlja novi bežični komunikacijski sustav za E-Ink zaslone. Sustav koristi NRF24 module za omogućavanje učinkovitog prijenosa podataka između računala i E-Ink zaslona. Rad se fokusira na minimiziranje potrošnje energije i maksimiziranje dometa prijenosa uz osiguranje pouzdanog prikaza slika. Rješenje je ostvareno kroz dizajn električne sheme, izradu tiskane pločice, te razvoj softvera za obradu slike i optimizaciju komunikacijskih protokola.

Ključne riječi: Arduino, bežična komunikacija, E-Ink, komunikacijski protokol, NRF24

ABSTRACT

Title: E-Ink display with NRF24 communication

This work presents a new wireless communication system for E-Ink displays. The system uses NRF24 modules to enable efficient data transfer between a computer and an E-Ink display. The focus of the work is on minimizing power consumption and maximizing transmission range while ensuring reliable image display. The solution is achieved through the design of an electrical schematic, the production of a printed circuit board, and the development of software for image processing and optimization of communication protocols.

Keywords: Arduino, wireless communication, E-Ink, communication protocol, NRF24

PRILOZI

Uz rad su priloženi sljedeći dokumenti:

- Kôd odašiljača i prijemnika
- Kôd pokaznika
- Kôd računala
- Tekst završnog rada u PDF i Word formatu