

Digitalni kontroler za silazno/ulazni most na brodu

Kamenar, Dominik

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Maritime Studies, Rijeka / Sveučilište u Rijeci, Pomorski fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:187:891525>

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

Download date / Datum preuzimanja: **2024-06-29**



Sveučilište u Rijeci, Pomorski fakultet
University of Rijeka, Faculty of Maritime Studies

Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Maritime Studies - FMSRI Repository](#)



**SVEUČILIŠTE U RIJECI
POMORSKI FAKULTET**

DOMINIK KAMENAR

**DIGITALNI KONTROLER ZA SILAZNO/ULAZNI MOST NA
BRODU**

ZAVRŠNI RAD

Rijeka, 2023. godina.

**SVEUČILIŠTE U RIJECI
POMORSKI FAKULTET**

**DIGITALNI KONTROLER ZA SILAZNO/ULAZNI MOST NA
BRODU
DIGITAL CONTROLLER FOR BOAT'S GANGWAYS**

ZAVRŠNI RAD

Kolegij: Pomorski komunikacijski uređaji

Mentor: Doc. dr. sc. Zoran Mrak

Student: Dominik Kamenar

Studijski smjer: Elektroničke i informatičke tehnologije u pomorstvu

JMBAG: 0112084725

Rijeka, Rujan, 2023. godina.

Student: Dominik Kamenar

Studijski program: Elektroničke i informatičke tehnologije u pomorstvu

JMBAG: 0112084725

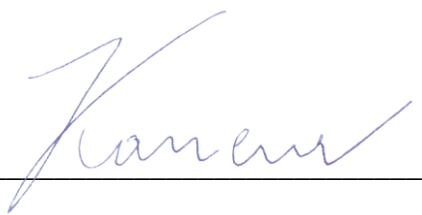
IZJAVA O SAMOSTALNOJ IZRADI ZAVRŠNOG RADA

Kojom izjavljujem da sam završni rad s naslovom
Digitalni kontroler za silazno / ulazni most na brodu
(naslov završnog rada)

izradio samostalno pod mentorstvom
doc. dr. sc. Zoran Mrak
(prof. dr. sc. / izv. prof. dr. sc. / doc dr. sc Ime i Prezime)

U radu sam primijenio metodologiju izrade stručnog/znanstvenog rada i koristio literaturu koja je navedena na kraju završnog rada. Tuđe spoznaje, stavove, zaključke, teorije i zakonitosti koje sam izravno ili parafrazirajući naveo u završnom radu na uobičajen, standardan način citirao sam i povezoao s fusnotama i korištenim bibliografskim jedinicama, te nijedan dio rada ne krši bilo čija autorska prava. Rad je pisan u duhu hrvatskoga jezika.

Student


(potpis)

Dominik Kamenar

Student: Dominik Kamenar

Studijski program: Elektroničke i informatičke tehnologije u pomorstvu

JMBAG: 0112084725

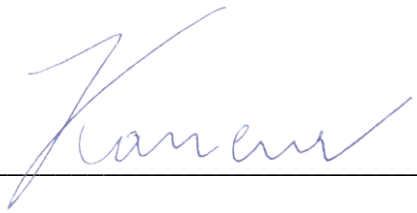
IZJAVA STUDENTA – AUTORA
O JAVNOJ OBJAVI OBRANJENOG ZAVRŠNOG RADA

Izjavljujem da kao student – autor završnog rada dozvoljavam Pomorskom fakultetu Sveučilišta u Rijeci da ga trajno javno objavi i besplatno učini dostupnim javnosti u cjelovitom tekstu u mrežnom digitalnom repozitoriju Pomorskog fakulteta.

U svrhu podržavanja otvorenog pristupa završnim radovima trajno objavljenim u javno dostupnom digitalnom repozitoriju Pomorskog fakulteta, ovom izjavom dajem neisključivo imovinsko pravo iskorištavanja bez sadržajnog, vremenskog i prostornog ograničenja mog završnog rada kao autorskog djela pod uvjetima *Creative Commons* licencije CC BY Imenovanje, prema opisu dostupnom na <http://creativecommons.org/licenses/>

Student - autor

(potpis)



SAŽETAK

U ovom završnom radu biti će opisan način rada digitalnog bežičnog kontrolera za silazno ulazni most na brodu. Biti će opisan princip rada bluetooth tehnologije te proces izrade mobilne aplikacije koja će poslužiti kao kontroler. Također opisati će se princip ostvarivanja bluetooth veze između mobilnog uređaja i bluetooth modula te način obrade odaslanih signala u Arduino Nano mikrokontroleru.

Ključne riječi :

Bluetooth, Bluetooth Low Energy, Android Studio, xCode, Arduino, Arduino Nano, Bluetooth modul, EasyEDA, Mobilna aplikacija, Silazno ulazni most

SUMMARY

In this paper will be describe the working principle of the digital wireless controller for the ship's gangway. It will be described how does bluetooth technology work and the process of creating a mobile application that will serve as a controller will be described. It also includes the description of the principle of establishing a bluetooth connection between a mobile device and a bluetooth module and the method of processing the transmitted signals in the Arduino Nano microcontroller.

Keywords:

Bluetooth, Bluetooth Low Energy, Android Studio, xCode, Arduino, Arduino Nano, Bluetooth modul, EasyEDA, Mobile app, Gangway

Sadržaj

SAŽETAK	II
SUMMARY	II
SADRŽAJ	III
1. UVOD	1
2. BLUETOOTH protokol	2
2.1. POVIJEST BLUETOOTH-A	2
2.2. PRINCIP RADA BLUETOOTH TEHNOLOGIJE	3
2.2.1. Način ostvarivanja veze Bluetooth tehnologijom	6
2.2.2. Potencijalni problemi i opasnosti Bluetooth tehnologije.....	7
2.2.3. Način komunikacije Bluetooth uređaja.....	8
2.2.4. BLE (Bluetooth Low Energy).....	9
3. DIZAJN ELEKTRIČNE ŠEME	10
3.1. ARDUINO MIKROKONTROLER	10
3.2. BLE (Bluetooth Low Energy) modul	11
3.2.1. Princip rada HM – 10 BLE modula.....	11
3.2.2. AT naredbe.....	11
3.2.3. Prepoznavanje modula u okolini	12
3.2.4. Povezivanje i prijenos podataka	12
3.2.5. Karakteristike i usluge HM – 10 modula	12
3.2. DIZAJN ELEKTRIČNE ŠEME SABIRNICE.....	12
4. RAZVOJ APLIKACIJE	16
4.1. RAZVOJ ANDROID APLIKACIJE	17
4.2. RAZVOJ iOS APLIKACIJE	22
4.3. KONAČNI REZULTAT	24
6. ZAKLJUČAK	25
LITERATURA	26
POPIS SLIKA	27

1. UVOD

Digitalizacija kontrolnih jedinica (uređaja) na luksuznim brodovima je sve češća, kao i potreba da većinu informacija o našem plovilu imamo na mobilnom uređaju. Napretkom tehnologije bluetooth-a, napretkom samih mobilnih uređaja, napretkom podrške za učenjem određenih područje u tehnologiji te razvoj tvrki i programa za dizajniranje i printanje električnih pločica, vrlo je jednostavno dizajnirati uređaj koji će kontrolirati ulazno/silazni most na brodu preko našeg mobilnog uređaja. Upravo taj uređaj i aplikacija koja će upravljati tim uređajem biti će opisani u ovom završnom radu.

Silazno ulazni mostovi na brodu (pasarele) su mostovi koji omogućuju ulazak i silazak s broda kada je on privezan na rivi. Sastoje se od mosta koji je pomičan u 2 smjera – gore i dolje, također sadrži teleskopski most koji omogućuje kretanje u još 2 smjera – unutra i van. Most je hidraulički kontroliran te je hidraulička pumpa smještena u motornom prostoru broda. Kontrolna jedinica ili računalo također je smješteno u motornom prostoru broda te je povezano s napajanjem i komandom. Pritiskom određenog gumba na komandi, kontrolna jedinica šalje signal koji će pokrenuti hidrauličku pumpu te drugi signal koji će otvoriti određeni električni ventil (ovisno o gumbu kojeg smo pritisnuli). Također postoje silazno/ulazni mostovi koji su potpuno uvlačivi u unutrašnjost broda (u trup broda). Tada takav silazno/ulazni most ima dodatne dvije opcije uvlačenja i izvlačenja (automatic in i automatic out).

Digitalni kontroler koji je opisan kroz ovaj završni rad se temelji na bluetooth tehnologiji. U prvom poglavlju opisan je način funkcioniranja bluetooth-a te princip rada bluetooth povezivanja i vršenje digitalne kontrole bluetooth modula. Zatim će u nastavku biti opisan princip primanja i dekodiranja podataka odaslanih iz mobilne aplikacije na sam uređaj te elementi i princip rada samog kontrolnog uređaja. U drugom djelu završnog rada biti će opisan postupak izgradnje same android i IOS aplikacije te kako se vrši povezivanje mobilne aplikacije s bluetooth modulom iz samog uređaja te način i izmjenjivanja podataka između aplikacije i samog uređaja.

2. BLUETOOTH protokol

Već vrlo rano nastala je potreba za bežičnim mrežama i sustavima u tehnologiji. Između brojnih ostalih bežičnih načina povezivanja (tj. frekvencija korištenih za ostvarivanje veze/komunikacije) bluetooth iskače iz razloga što je u potpunosti standardiziran te ga sadrži gotovo svaki IoT uređaj. Iako znamo da je do sada digitalna elektronika jako puno napredovala, i dalje je fascinantno da prilikom slušanja glazbe pomoću bežičnih slušalica, pomoću bluetooth veze prenosimo sa mobilnog uređaja na slušalice preko milijun jedinica i nula u jednoj sekundi.

Način rada bluetooth protokola te njegova povijest razvoja biti će objašnjeni u sljedećim poglavljima.

2.1. POVIJEST BLUETOOTH-A

Naziv "Bluetooth" predložio je 1997. Jim Kardach iz tvrtke Intel, jedan od osnivača Bluetooth SIG-a. Dobio je inspiraciju za naziv u razgovoru sa Svenom Mattissonom koji mu je ispričao skandinavsku povijest kroz priče iz knjige 'The Long Ships', povijesnog romana o Vikinzima i danskom kralju Haroldu Bluetooth-u iz 10. stoljeća, autora Fransa G. Bengtssona. Nakon što je otkrio sliku runskog kamena Haralda Bluetootha u knjizi Povijest Vikinga autora Gwyna Jonesa, Kardach je predložio Bluetooth kao kodno ime za bežični program kratkog dometa.

Razvoj radio tehnologije bluetooth-a započeo je 1989. godine Nils Rydbeck, tehnički direktor u tvrtki Ericsson Mobile u Lundu u Švedskoj. Svrha je bila razviti bežične slušalice, na temelju dva izuma Johana Ullmana. Nils Rydbeck zadužio je Torda Wingrena za specifikaciju, a Nizozemce Jaapa Haartsena i Svena Mattissona za razvoj tehnologije. Obojica su radili za tvrtku Ericsson u Lundu u Švedskoj. Glavni dizajn i razvoj započeli su 1994. te do 1997. tim je imao djelotvorno rješenje. Od 1997. Örjan Johansson postao je voditelj projekta i pokrenuo tehnologiju i standardizaciju.

Godine 1997. Adalio Sanchez, tadašnji voditelj IBM-ovog odjela za istraživanje i razvoj proizvoda ThinkPad, obratio se prethodno navedenom Nilsu Rydbecku oko suradnje na integraciji mobilnog telefona u prijenosno računalo ThinkPad. Dva dodijeljena inženjera iz Ericssona i IBM-a proučila su ideju. Zaključak je bio da je potrošnja energije na tehnologiji mobilnih telefona u to vrijeme bila previsoka da bi se omogućila održiva integracija u prijenosno računalo i još uvijek postiglo odgovarajuće trajanje baterije. Umjesto toga, dvije su

se tvrtke složile integrirati Ericssonovu tehnologiju bluetooth-a na ThinkPad prijenosno računalo i Ericssonov mobilni uređaj. Budući da niti IBM ThinkPad prijenosna računala niti Ericssonovi mobilni uređaji nisu bili vodeći na tržištu u to vrijeme, Adalio Sanchez i Nils Rydbeck složili su se da će tehnologija bluetooth postati otvoreni industrijski standard kako bi se svakom proizvođaču digitalne elektronike omogućio pristup patentu. Adalio Sanchez iz IBM-a zatim je angažirao Stephena Nachtsheima iz Intela da se pridruži, a potom je Intel također angažirao Toshiba i Nokiju. U svibnju 1998. pokrenut je Bluetooth SIG s IBM-om i Ericssonom kao osnivačkim potpisnicima i ukupno pet članova: Ericsson, Intel, Nokia, Toshiba i IBM

Prvi Bluetooth uređaj predstavljen je 1999. Bile su to bežične mobilne slušalice koje su osvojile nagradu "Best of show Technology Award" na COMDEX sajmu. Prvi Bluetooth mobilni uređaj bio je Ericsson T36, ali bio je to ustvari model T39 koji zapravo dospio na police trgovina 2001. godine. Usporedno s tim, IBM je u listopadu 2001. predstavio IBM ThinkPad A30 koje je bilo prvo prijenosno računalo s integriranim Bluetooth-om.

2.2. PRINCIP RADA BLUETOOTH TEHNOLOGIJE

Bluetooth radi na frekvencijama između 2,402 i 2,480 GHz, ili 2,400 i 2,483.5 GHz, uključujući zaštitne pojaseve širine 2 MHz na donjem kraju spektra i 3,5 MHz na gornjem kraju spektra. Isto kao što i mi raspoznavamo boju svijetla sa semafora zato što je svaka boja predstavljena određenom frekvencijom elektromagnetskog vala tj. valnom duljinom, tako i mobilni uređaji ili bilo koji IoT uređaji koji koriste bluetooth način komuniciranja izmjenjuju informacije i ostvaruju komunikaciju koristeći određen spektar frekvencija tj. valnih duljina elektromagnetskih valova. Iz prethodno navedenog spektra frekvencija možemo zaključiti da se valna duljina elektromagnetskih valova korištenih u bluetooth tehnologiji kreće oko 123 milimetara, što je milijun puta veća valna duljina nego valna duljina spektra elektromagnetskih valova vidljivog ljudskom oku. Elektromagnetski valovi takvih valnih duljina mogu bez ometanja proći kroz prepreka kao što su zidovi, isto kao što i svjetlost može proći kroz staklene površine.

Kada naš mobilni uređaj šalje velik skup jedinica i nula koristeći bluetooth tehnologiju, tada elektromagnetski valovi valne duljine 121 milimetar predstavljaju logičke jedinice, a elektromagnetski valovi valne duljine 124 milimetara predstavljaju logičke nule. Naša antena na mobilnom uređaju ima mogućnost emitiranja prethodno navedenih valova čak do milijun

puta u sekundi što bi značilo da naša antena na mobilnom uređaju ima mogućnost promjene emitiranja valova valne duljine 121 milimetar i valova valne duljine 124 milimetar svakih 0.000001 sekundi. Takve karakteristike antene mobilnog uređaja znače da je kapacitet prijenosa podataka korištenjem bluetooth tehnologije milijun bita po sekundi (1000000 bit/sec).

Bitno je napomenuti da valne duljine spektra elektromagnetskih valova korištenih u bluetooth tehnologiji odgovaraju :

- 120.7 milimetara za najvišu frekvenciju spektra (2,4835 GHz)
- 124.9 milimetara za najnižu frekvenciju spektra (2,4 GHz)

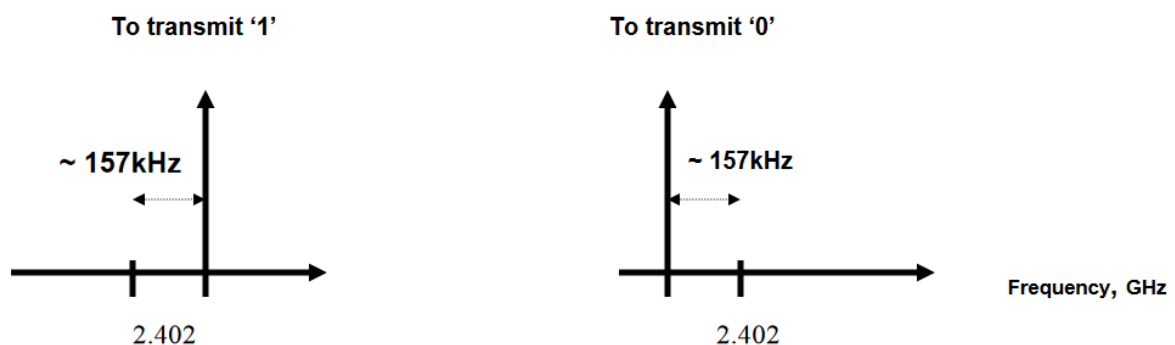
Kao što i naše oko može vidjeti samo određen spektar elektromagnetskih valova (od 400nm do 700 nm) tako i uređaji mogu komunicirati koristeći bluetooth tehnologiju samo u prethodno navedenom spektru elektromagnetskih valova.

Bluetooth tehnologija se temelji na Frequency-Hopping Spread Spectrum (FHSS) tehnologiji, koja se koristi da bi se smanjio utjecaj smetnji i preklapanja informacija. Snaga signala koji se odašilje ne iznosi više od 100 mW. FHSS prenosi informaciju prenošenjem kratkog niza paketa na jednoj frekvenciji tj. na jednom kanalu te zatim skoči na drugu frekvenciju, te ondje prenosi sljedeći niz podataka. U FHSS sustavu frekvencija nositelja odašiljača skače s frekvencije na frekvenciju u skladu s slučajnim redoslijedom. Brzina skakanja frekvencije je 1600 skokova po sekundi. Brzina skakanja s frekvencije na frekvenciju povećat će se na 3200 skokova/s ako već postoji uspostavljena veza. Skakanje s frekvencije na frekvenciju također pruža poboljšani prijem podataka u prisutnosti ometajućih signala i šumova. Ako dođe do smetnji na određenoj frekvenciji, samo će dio frekvencijskih skokova biti blokiran umjesto cijelog signala. Dva načina za ispravljanja pogrešaka u podacima su: kod za ispravljanje pogrešaka (Forward Error Correction Code - FEC) i automatski zahtjev za ponavljanjem (Automatic Repeat Request - ARQ). FEC kod je implementiran u samim paketima podataka kako bi se smanjio broj ponovnih prijenosa paketa. U normalnom okruženju, bez prisutnosti smetnji i grešaka na podacima stvara nepotrebno opterećenje, što smanjuje propusnost. U ARQ načinu ipravljanja pogrešaka, paketi se odašilju i ponovno odašilju sve dok uređaj za odašiljanje ne primi potvrdu uspješnog prijema.

Bluetooth Core Specification klasificira odašiljače u tri klase s obzirom na snagu radio prijenosa : 100 mW (20 dBm), 2,5 mW (4 dBm) i 1 mW (0 dBm). Sa snagom od 0 dBm, komunikacijski domet može biti do 10 metara, dok snaga prijenosa od 20 dBm povećava domet na 100 metara. Iznad 4dBm, kontroliramo snagu prijenosa odgovarajućih signala, koja je

proporcionalna udaljenosti na kojoj se komunikacije odvija. Bluetooth prijemnik zahtijeva razinu osjetljivosti od -70 dBm ili bolju.

Bluetooth modulacije koja se koristi jest GFSK (Gaussian Frequency Shift Keying) s indeksom modulacije između $0,28\sim 0,35$. Binarna FSK modulacija Gaussovog oblika smanjuje složenost primopredajnika. Binarna jedinica jest prikazana korištenjem pozitivne devijacije frekvencije dok je binarna nula predstavljena negativnom devijacijom frekvencije. Tijekom jednog vremenskog perioda, podaci mogu promijeniti vrijednost svake jedne mikro sekunde, tako da frekvencija prijenosa oscilira naprijed-natrag oko središnje frekvencije odgovarajućeg kanala. Ako se dopusti da se promjena frekvencije dogodi trenutno, to može dovesti do inter-simbolske interferencije (ISI) na prijemniku. ISI otežava tumačenje stanja koje bit pokušava prikazati što dovodi do pogrešaka u bitovima u prenesenim podacima. Kako bi smanjio spektralno širenje koje uzrokuje ISI, Bluetooth koristi 5BT Gaussov filter za usporavanje prijelaza između dviju frekvencija.



Slika 1: Prikaz frekvencija za odašiljanje

Izvor: Bluetooth Transmission Technology

Paketi su raščlanjeni na svoje sastavne dijelove kao što su pristupni kod, zaglavlje paketa, zaglavlje korisnog sadržaja i korisni sadržaj. Svaki tip paketa ima različitu razinu ispravljanja pogrešaka i zaštite te različite veličine nosivosti. Pristupni kod se koristi za otkrivanje prisutnosti paketa i za adresiranje paketa na određeni uređaj. Paket zaglavlja sadrži kontrolne informacije povezane s paketom kao što je adresa podređenog uređaja kojem je paket namijenjen. Na kraju, korisni sadržaj sadrži informacije o poruci.



Slika 2: Prikaz paketa

Izvor: Bluetooth Transmission Technology

2.2.1. Način ostvarivanja veze Bluetooth tehnologijom

Potrebno je uzeti u obzir da se u stvarnom svijetu nalazi velik broj uređaja koji mogu interferirati te omesti našu komunikaciju između mobilnog uređaja i bluetooth modula. Zbog toga se postavlja pitanje zašto naše bežične slušalice prepoznaju signale samo s našeg mobilnog uređaja.

Zbog toga što postoji mogućnost da veći broj korisnika koristi komunikaciju bluetooth tehnologijom, frekvencijski spektar korišten za komunikaciju podjeljen je na 79 kanala. Svaki od prethodno navedenih kanala sadrži dvije frekvencije. Jedna frekvencija predstavlja logičku jedinicu te druga logičku nulu. Tada naš mobilni uređaj i bežične slušalice komuniciraju koristeći jedan od prethodno navedenih 79 kanala. Primjerice za komunikaciju pomoću kanala 38, koristi se elektromagnetski val frekvencije 2.44387 GHz za logičku jedinicu te elektromagnetski val frekvencije 2.44313 GHz za logičku nulu a za kanal 54 koristi se elektromagnetski val frekvencije 2.45587 GHz za logičku jedinicu te 2.42713 GHz za logičku nulu.

Iako imamo 79 kanala gdje svaki kanal ima širinu pojasa od 1 MHz, ne dolazi do interferencije podataka ako se u određenoj prostoriji nalazi više uređaja koji istovremeno komuniciraju bluetooth tehnologijom. Prije svega bitno je razumjeti da se podaci prenose u paketima kod kojih prvih 72 bita predstavlja pristupni kod između dva uređaja koja komuniciraju tako da bežične bluetooth slušalice s kojima smo povezani znaju da paketi s tim predodređenim pristupnim kodom (eng. *access code*) su paketi koje slušalice moraju primiti i dekodirati. Takvi pristupni kodovi mogu se objasniti kao adresa na paketu kojeg šaljemo. Sljedećih 54 bita su bitovi koji označavaju naslov paketa (eng. *Header*) koji donosi detalje o podacima koji se nalaze unutar paketa. Za bolje objašnjenje, naslov možemo usporediti s veličinom kutije paketa kojg šaljemo. Podatak o veličini nam поближе opisuje sadržaj paketa te nam daje neke osnovne podatke. Sljedećih primjerice 500 bitova predstavlja sadržaj paketa tj. skup jedinica i nula koji predstavljaju u slučaju bežičnih slušalica audio signal koji je dekodiran iz digitalnog u analogni oblik u samim slušalicama da bismo ga mi mogli čuti. Bitno je naglasiti

da veličina sadržaja paketa ovisi osmoj prirodi informacije koju prenosi te može iznositi od 136 bitova do 8168 bitova. Primjerice, jednostavne komande s našeg mobilnog uređaja kao na primjer zaustavi ili pokreni glazbu će sadržavati manji broj bitova u paketu nego slanje ili primanje digitaliziranog audio signala velike kvalitete.

Kada naše bežične slušalice i mobilni uređaj komuniciraju, nisu vezani uz samo jedan kanal nego skaču od jednog kanala na drugi. Ovaj proces naziva se frekvencijsko skakanje (eng. *Frequency-Hopping Spread Spectrum*) te se događa u prosjeku 1600 puta u sekundi. U slučaju komunikacije bežičnih slušalica i mobilnog uređaja, mobilni uređaj ima zadatak odrediti na kojim kanalima će se u buduću odvijati komunikacija te to određuje ovisno o prisutnosti smetnji na određenom kanalu. Ukoliko su u nekim kanalima od prethodno navedenih 79 kanala prisutne smetnje (elektromagnetske interferencije, prisutnost ostalih uređaja koji komuniciraju na tim kanalima), mobilni uređaj će se prilagoditi te će odabrati kanale na kojima su manje smetnje te tako neće kvariti kvalitetu prijenosa svojih podataka niti interferirati s drugim uređajima ili smetnjama. Na taj način se također spriječava mogućnost čitanja odaslanih podataka s uređaja na bežične slušalice pomoću nekog drugog prijemnika zato što samo mobilni uređaj i bežične slušalice koje su međusobno ostvarili bluetooth vezu znaju redoslijed kanala na kojima će u buduću komunicirati. Ukoliko se dogodi da naše slušalice nisu primele određen paket, određenim kanalom pošalju povratnu informaciju o nedostatku određenog paketa te mobilni uređaj ponovno šalje isti paket.

2.2.2. Potencijalni problemi i opasnosti Bluetooth tehnologije

Bitno je za napomenuti da spektar frekvencija od 2.4 GHz do 2.4835 GHz koriste i drugi uređaji osim bluetooth tehnologije. Na primjer, mikrovalna pećnica koristi elektromagnetski val frekvencije 2.45 GHz koji kada smo u blizini može izazvati smetnje u komunikaciji između primjerice bežičnih slušalica i našeg mobilnog uređaja. Također brojni medicinski uređaji koriste ovaj spektar frekvencija. Bitno je također napomenuti da bluetooth uređaji ne predstavljaju gotovo nikakvu opasnost svojim zračenjem zato što je emitirana snaga bluetooth signala približno 0.5 W, dok je snaga emitiranog elektromagnetskog vala u mikrovalnoj pećnici približno 1000 W.

U prethodno navedenom spektru elektromagnetskih valova također rade i 2.4 GHz WiFi modemi tj. mreže. Slično kao i bluetooth tehnologija, WiFi mreže radni spektar podjele na 14 kanala kojim se vrši komunikacija i izmjena podataka međutim bitno je naglasiti da WiFi

modemi koriste znatno drugačije protokole od bluetootha kako bi se smanjile mogućnosti interferiranja dvaju signala.

Određen paket, odaslan bluetooth tehnologijom i protokolima, također sadrži bitove za određivanje greške te filtere smetnji (šuma) koji pridonose ispravnoj komunikaciji i razmjeni podataka između uređaja koji koriste bluetooth komunikaciju. Unutar svakog bluetooth modula nalazi se bluetooth mikročip koji je zaslužan za filtriranje smetnji (šuma), također za prepoznavanje grešaka u odaslanim paketima informacija prateći određene protokole te odabir kanala za sljedeći ciklus slanja podataka.

2.2.3. Način komunikacije Bluetooth uređaja

Shema slanja digitalnog signala tj. binarnog seta jedinica i nula u obliku elektromagnetskih valova različitih valnih duljina naziva se eng. *frequency-shift keying*. Ovaj naziv u osnovi označava promjenu frekvencije prilikom komuniciranja dvaju uređaja korištenjem bluetooth tehnologije te *keying* označava upravo to da određena frekvencija je dodjeljena logičkoj jedinici, a neka druga frekvencija logičkoj nuli.

Naš mobilni uređaj emitira elektromagnetski val određene frekvencije koji se naziva val nosioc. Da bi se taj val nosioc prilagodio frekvenciji koji mi želimo odaslati, koja će predstavljati logičku jedinicu ili nulu, tada taj val moramo modulirati. Dakle, bitno je razumjeti da naš uređaj ne proizvodi različiti elektromagnetski val za svaku logičku jedinicu ili nulu, za svaki od 79 kanala, nego se jedan val nosioc modulira da bi se dobila željena izlazna frekvencija. Ovaj proces modulacije vala nosioca naziva se frekvencijska modulacija vala.

Također postoji fazna modulacija vala nosioca. U ovoj modulaciji, valu nosiocu mjenjamo fazu prilikom emitiranja. Fazni pomak od 0 stupnjeva predstavljat će logičku jedinicu te fazni pomak od 90 stupnjeva predstavljat će logičku nulu.

Bitno je napomenuti da bluetooth omogućava komunikaciju u oba smjera. Kada razgovaramo na mobilni uređaj preko prethodno povezanih bluetooth slušalica, naše slušalice sada šalju pakete, koji su nastali kao modulirani signal našeg govora u mikrofon slušalica, natrag u mobilni uređaj. Bluetooth tehnologija komuniciranja ima mogućnost izmjeničnog prijenosa podataka između uređaja koji komuniciraju dok još uvijek zadržava redoslijed skakanja s kanala na kanal.

2.2.4. BLE (Bluetooth Low Energy)

Bluetooth Low Energy jest novija verzija bluetooth tehnologije koja je razvijena za IoT (Internet of Things) uređaje. Bluetooth Low Energy jest 4.0 (te novije) verzija bluetooth tehnologije, razvijena 2012. godine. Bitno je napomenuti da razmak između kanala kod Bluetooth Low Energy tehnologije jest 2 MHz što znači da sada više nemamo 79 kanala kao što smo imali u Bluetooth Classic tehnologiji (npr. Verzija 2.0) nego imamo 40 kanala. Samim time Bluetooth Low Energy i Bluetooth Classic ne mogu raditi zajedno tj. moramo imati kompatibilne uređaje da bi se mogla ostvarivati komunikacija. Mobilni uređaji koji nemaju Bluetooth Classic modul u sebi ne mogu vršiti komunikaciju s Bluetooth Classic uređajima i obrnuto.

3. DIZAJN ELEKTRIČNE ŠEME

Kao što je navedeno u uvodu, u ovom završnom radu biti će opisan uređaj, čiji je zadatak imati mogućnost upravljanja silazno/ulaznim mostom na brodu pomoću našeg mobilnog uređaja. Konstruirani uređaj namjenjen je za upravljanje silazno/ulaznih mostova brodova za razonodu (jahti). U ovom poglavlju biti će opisan svaki element koji sam za sebe čini zaseban modul u samom uređaju a to su :

- Arduino mikrokontroler
- BLE (Bluetooth Low Energy) modul
- Dizajn sabirnice s pripadajućim relejima koja povezuje Bluetooth modul i Arduino pločicu

3.1. ARDUINO MIKROKONTROLER

Arduino je open-source platforma koja se koristi za konstruiranje i programiranje elektroničkih pločica. Može primati i slati informacije na većinu uređaja, pa čak i putem interneta za upravljanje određenim elektroničkim uređajem. Koristi hardver koji se zove arduino uno tiskana pločica i softverski program (pojednostavljeni C++) za programiranje ploče. U današnjem svijetu arduino uređaji se koriste uglavnom kao mikrokontroleri. Odlikuje ih jednostavnost upravljanja i pouzdanost.

Arduino platforma je dobro poznata ljudima koji se bave elektronikom. Za razliku od većine prethodnih programibilnih sklopnih pločica, Arduino nema zaseban dio hardvera za učitavanje novog koda na pločicu, nego se koristiti USB kabel za učitavanje, a softver Arduina koristi pojednostavljenu verziju C++, što pruža lakše okruženje koje zaobilazi funkcije mikrokontrolera te samim time predstavlja pristupačniji paket tj. vrlo je jednostavno shvatiti i naučiti upravljati Arduino mikrokontrolerom.

Arduino pločica se sastoji od Atmega 328p mikrokontrolera te se koristi Arduino IDE programabilno sučelje za presađivanje koda u memoriju pločice.

U ovom projektu korišten je Arduino Nano mikrokontroler. Ova verzija mikrokontrolera se naziva AVR, što znači da neovisno o opterećenju na digitalnim pinovima uvijek će se nalaziti stalan napon – operativni napon pločice koji je u ovom slučaju 5 V. Arduino Nano mikrokontroler sadržava 32 KB memorije te operativna frekvencija mikrokontrolera je 16 MHz. Maksimalna struja koji ovaj mikrokontroler može povući jest 85 mA te potrošnja struje u stanju mirovanja jest 19 mA. Arduino Nano pločica sadrži 13 digitalnih I/O pinova te 8 analognih I/O pinova. Operativni napon pločice jest 5V te također sadrži buck pretvarač koji omogućuje

ulazni napon od 7 V do 12 V. Arduino Nano pločica također sadrži Mini - USB utor pomoću kojeg se presaduje kod iz Arduino IDE sučelja. Vrlo važni pinovi za ovaj projekt su također Rx te Tx pinovi, na koje se priključuje Bluetooth modul.

3.2. BLE (Bluetooth Low Energy) modul

BLE uređaji se koriste u mrežama/sustavima vrlo niskih snaga tj. sam modul po sebi troši vrlo malo snage/struje. Uređaj kojeg ćemo koristiti u ovom projektu jest Bluetooth modul HM – 10. Potrošnja ovog uređaja jest otprilike 10 mA. Ovakvi bluetooth moduli su vrlo rasprostranjeni te se koriste u brojnim IoT (Internet of Things) uređajima te uređajima koji su napajani iz baterijskog izvora zbog upravo toga što ih odlikuje vrlo niska potrošnja energije.

HM-10 Bluetooth modul popularan je Bluetooth Low Energy (BLE) modul koji je vrlo popularan među mladim inženjerima te koristi verziju 5.0 bluetootha. Modul se temelji na Texas Instruments CC2540 ili CC2541 BLE čipu. CC2540 ili CC 2541 BLE čipove odlikuje jednostavnost, pouzdanost te niska cijena. Upravo zbog tih karakteristika su jedni od najrasprostranjenijih čipova korištenih u ostvarivanju Bluetooth komunikacije.

3.2.1. Princip rada HM – 10 BLE modula

Modul HM-10 koristi Bluetooth Low Energy protokol koji je dizajniran za bežičnu komunikaciju malog dometa male snage. Ovaj modul je idealan za aplikacije u kojima uređaji trebaju razmjenjivati male količine podataka uz uštedu energije.

Modul HM-10 može raditi u dva načina: periferni i središnji. U perifernom načinu rada može djelovati kao podređeni uređaj i povezati se sa središnjim uređajem, poput pametnog telefona ili računala. U središnjem načinu rada može djelovati kao glavni uređaj i povezati se s više perifernih uređaja.

3.2.2. AT naredbe

Modul HM-10 se konfigurira i kontrolira pomoću AT naredbi. Ove se naredbe šalju s mikrokontrolera ili glavnog uređaja na modul putem serijskog sučelja (obično UART). AT naredbe koriste se za postavljanje radnih parametara modula, kao što su naziv uređaja, brzina prijenosa podataka, interval oglašavanja i parametri veze.

3.2.3. Prepoznavanje modula u okolini

U perifernom načinu rada, modul HM-10 dijeli svoju prisutnost drugim uređajima povremenim emitiranjem paketa koji sadrže njegov jedinstveni identifikator (UUID). Središnji uređaji, poput pametnih telefona, traže ove signale kako bi otkrili i povezali se s modulom.

3.2.4. Povezivanje i prijenos podataka

Kada se uspostavi veza između modula HM-10 i središnjeg uređaja, oni mogu razmjenjivati podatke. Modul pruža transparentno sučelje za serijske podatke, omogućujući mikrokontroleru ili glavnom uređaju slanje i primanje podataka.

3.2.5. Karakteristike i usluge HM – 10 modula

Karakteristike i usluge: Modul HM-10 podržava *Bluetooth Generic Attribute Profile* (GATT), koji definira strukturu i format podataka koji se razmjenjuju između uređaja. GATT koristi hijerarhijsku strukturu atributa, uključujući usluge i karakteristike. Usluge predstavljaju skup povezanih karakteristika, dok karakteristike predstavljaju specifične elemente podataka, kao što su temperatura, vlažnost ili očitavanja senzora.

Modul HM-10 može se prilagoditi ažuriranjem. To nam omogućuje izmjenu ponašanja samog modula ili dodavanje novih funkcija. Međutim, prethodno navedeno ažuriranje zahtijeva vještine programiranja i znanje pristupa odgovarajućim alatima. Važno je napomenuti da se specifičnosti rada s modulom HM-10 mogu razlikovati ovisno o verziji software-a tj. ažuriranja te specifičnosti implementacije modula unutar pripadajućeg projekta.

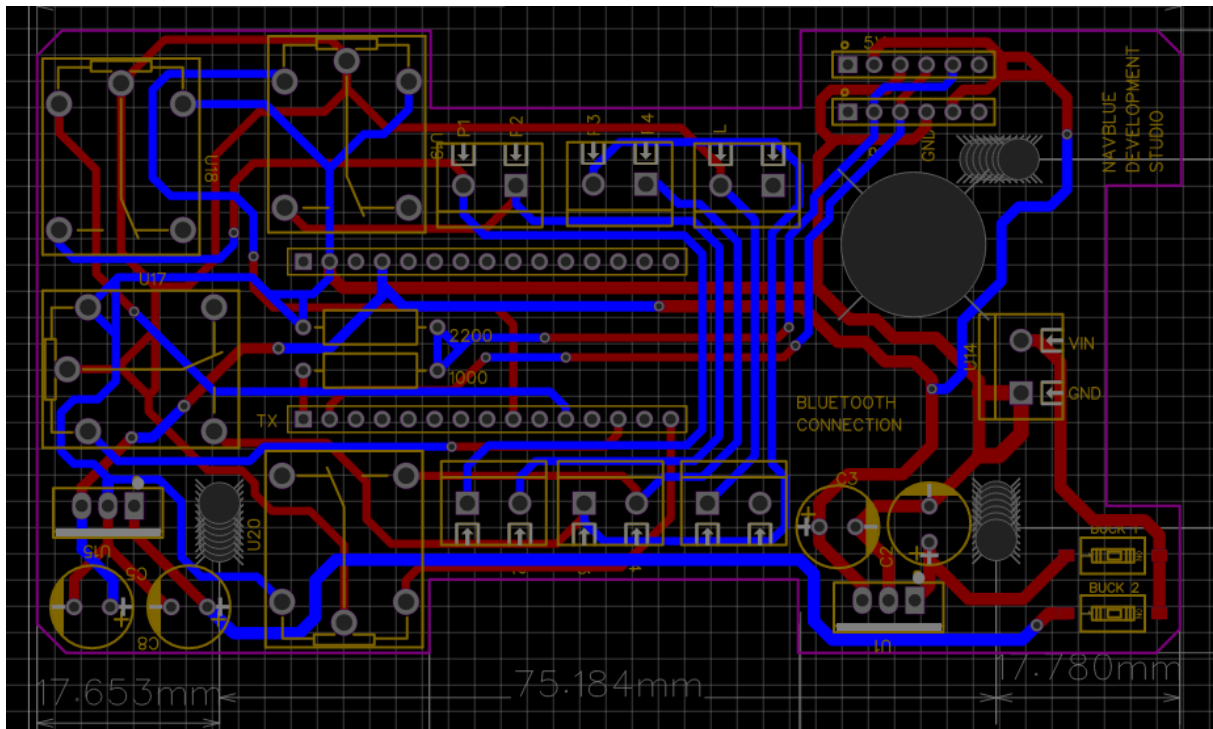
3.3. DIZAJN ELEKTRIČNE ŠEME SABIRNICE

U lipnju 2010. Dillon He i Eric Cui započeli su istraživanje EDA alata za vlastiti hardverski projekt. Potraga za EDA alatom sposobnim za shematsko projektiranje, simulaciju strujnog kruga i raspored elemenata na tiskanoj pločici imala je ograničeni uspjeh, te su zaključili da bi dizajniranje vlastitog EDA alata bilo korisno. Njihovi ključni ciljevi bili su da alati budu neovisni o platformi, besplatni, laki za učenje i korištenje. Prva verzija paketa alata pojavila se online u kolovozu 2013., a od službenog pokretanja u ožujku 2014. software se brzo razvijao s nekoliko velikih revizija i poboljšanja. Prva vanjska ulaganja EasyEDA software privukao je 2015. godine, te danas broji više od tri milijuna korisnika.

Dizajniranje i rad na *PCB-u (printed circuit board)* za ovaj projekt izvršen je u prethodnom navedenom programu EasyEDA. Ovaj software je vrlo jednostavan za savladati te

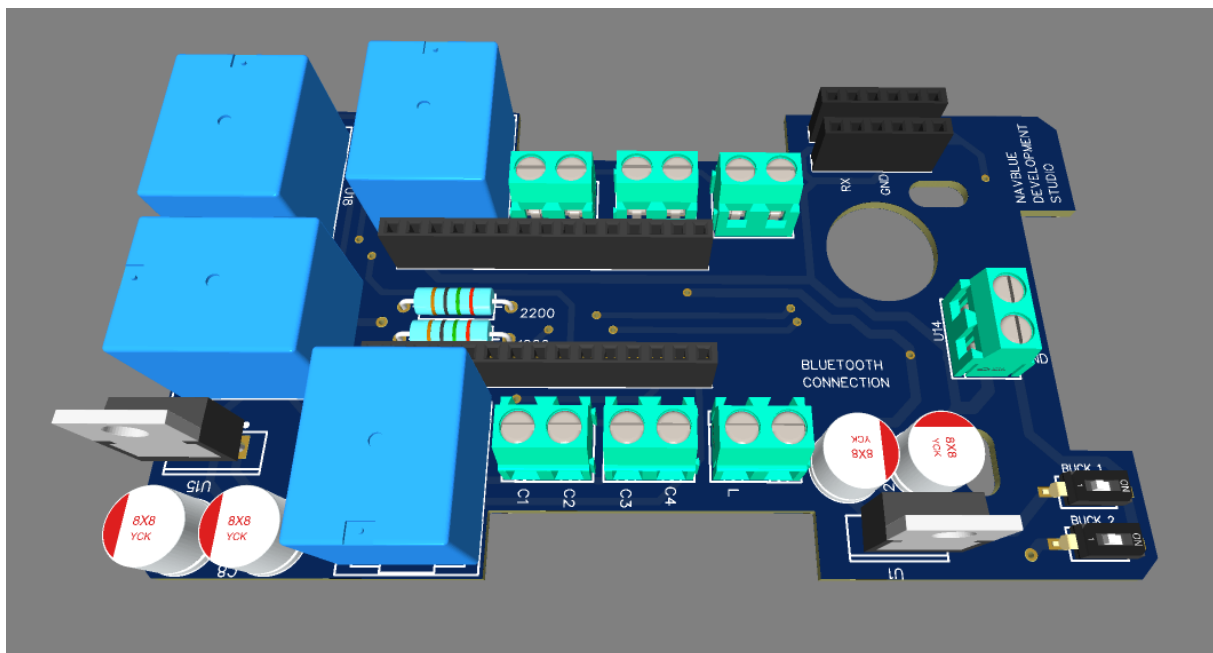
Sa šeme možemo vidjeti da nam je potreban ulazni napon (za ovaj slučaj je to 5V). Tim naponom napajati ćemo Arduino Nano mikrokontroler. Također možemo primjetiti da su četiri digitalna I/O pina povezana s kontaktima releja. Za ovaj projekt koristio sam C396953 releje čiji je okidni napon 5 V s *normally open* kontaktima. 5V je upravo napon kojeg ćemo dovesti s Arduino Nano mikrokontrolera koji ima mogućnost dostaviti maksimalno 40mA struje, što je i više nego dovoljno da se relej uključi. Ako promatramo pinove pod brojem 1 na relejima, možemo vidjeti da su povezani svi zajedno na vodič sabirnice koji povezuje priključak P8 i P3. Na ovom vodiču će biti prisutan pozitivan potencijal napona ako je elektronika tj. upravljačka jedinica silazno/ulaznog mosta priključena na napajanje. Prisutnost pozitivnog potencijala napona na ovom vodiču sabirnice upravo ovisi da li je uređaj priključen na elektroniku (glavno računalo) silazno / ulaznog mosta. U ovom uređaju releji služe upravo da kratko spoje prethodno naveden vodič s vodičima koji povezuju pinove P4, P5, P6 te P7. Ovisno na koji od tih vodiča priključimo prethodno navedeni pozitivan potencijal napona, taj napon će aktivirati određen hidraulički ventil, te će se silazno/ulazni most kretati u određenom smjeru. U4 na šemi nam predstavlja pojednostavljenu šemu HM-10 BLE modula na kojeg dovodimo napon od 5 V te masu. Tx kontakt s HM-10 BLE modula povezujemo s Rx kontaktom na Arduino Nano Mikrokontroleru te Rx kontakt s HM-10 BLE modula, preko djelila napona, povezujemo s Tx kontaktom na Arduino Nano mikrokontroleru. Djelilo napona izvedemo tako da operativni napon Arduino Nano mikrokontrolera koji iznosi 5 V, moramo spustiti na približno 3.3 V zato što je to napon komunikacije HM-10 BLE modula. U ovom slučaju korišteni su otpornici od 2200 Ω i 1000 Ω .

Slika br. 4 prikazuje raspored elektroničkih elemenata te sabirnica koji povezuju ovaj uređaj u cjelinu. Ovaj uređaj je dizajniran za ulazni napon od 7 V do 35 V zato što se na ulazu koristi pretvarač za sniženje napona, koji će taj ulazni napon pretvoriti u željenih 5 V (napon koji nam je potreban za napajanje Arduino Nano mikrokontrolera). Na ulaznom i izlaznom pinu pretvarača za sniženje napona spojeni su elektrolitski kondenzatori koji služe za glađenje napona. Kapacitet kondenzatora jest 2200 nF. Prekidačima BUCK 1 i BUCK 2 određujemo koji pretvarač za sniženje napona će nam u određenom trenutku raditi. Naravno moguć je i rad oba pretvarača za sniženje napona.



Slika 4. Raspored elemenata i vodiča kontrolera

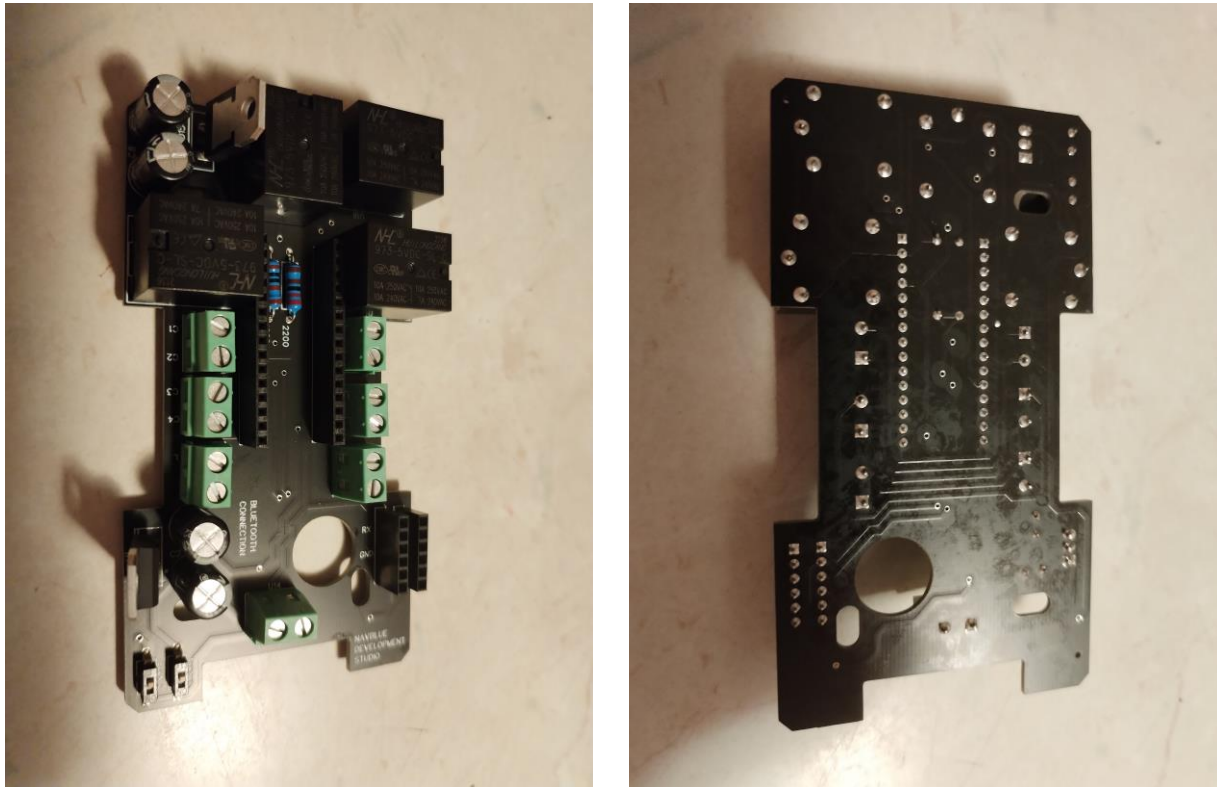
Izvor: izradio autor



Slika 5: 3D prikaz digitalnog kontrolera

Izvor: izradio autor

Slika 5. prikazuje 3D model uređaja dizajniranog u EasyEDA programu. Slika 6. prikazuje uređaj kojeg je proizvela tvrtka JLCPCB.



Slika 6. Prikaz digitalnog kontrolera

Izvor: izradio autor

4. RAZVOJ APLIKACIJE

Aplikacijsko programiranje postaje sve znatnija grana programiranja. Vrlo je bitno napomenuti da za pretvorbu ideje u stvarnost u aplikacijskom svijetu potrebno je poznavanje programiranja. U ovom projektu korištena su dva programa za izradu aplikacija. Jedan od njih se zove Android Studio te služi za izradu aplikacija za sustave koji koriste Android operativni sistem. Drugi program koji je korišten jest xCode. Ovaj program služi za razvoj aplikacija za uređaje koji podržavaju iOS (iPhone operativni sustav). Važno je napomenuti da xCode program podržava samo MAC operativne sustave, što znači da ga je moguće koristiti samo na MacBook prijenosnom računalu. Za izgradnju Android aplikacije potrebno je poznavanje Java programskog jezika te za razvoj iOS aplikacija potrebno je poznavanje Swift programskog jezika. U nastavku će biti detaljnije objašnjen postupak izgradnje android i iOS aplikacija te način povezivanja mobilnog uređaja s BLE HM – 10 modulom te slanje podataka tom istom modulu.

4.1. RAZVOJ ANDROID APLIKACIJE

Za razvoj Android aplikacije potreban nam je prethodno naveden program Android Studio. Ovaj program je besplatan te podržava Windows (8, 10, oili 11), Linux, macOS te ChromeOS.

Android Studio službeno je integrirano razvojno okruženje (IDE - *Integrated Development Environment*) za Android aplikaciju, te je tvrtka Google zaslužna za njegov razvoj. IDE sadrži alate koji programerima softvera omogućuju dizajn, izradu, pokretanje i testiranje softvera, u ovom slučaju aplikacija za platformu Android. Android Studio koristi IntelliJ IDEA kao svoju osnovu i uključuje unaprijed instalirani dodatak za Android zajedno s nekim izmjenama posebno za platformu Android. IntelliJ IDEA je integrirano razvojno okruženje za JVM programske jezike, osmišljeno za povećanje produktivnosti programera koji se njime koristi. Obavlja rutinske zadatke koji se ponavljaju umjesto nas, pružajući pametno dovršavanje koda, statičku analizu koda i refaktoriranje te omogućuje da se korisnici usredotoče na svijetlu stranu razvoja software-a, čineći ga ne samo produktivnim nego i ugodnim iskustvom.

JVM (Java Virtual Machine) programski jezici :

- Java
- Kotlin
- Scala
- Groovy
- Clojure
- Fantom
- Ceylon
- Jython
- JRuby
- Frege
- Xtend
- Golo

Ovaj popis JVM programskih jezika sadrži značajne računalne programske jezike koji se koriste za proizvodnju računalnog softvera koji radi na Java virtualnom stroju (JVM). Neke od ovih jezika interpretira Java program, a neki se kompajliraju u Java bajt kod i JIT (Just-In-Time) kompiliraju se tijekom izvođenja kao obični Java programi za poboljšanje performansi.

JVM je u početku dizajniran da podržava samo jezik Java. Međutim, s vremenom je sve više jezika prilagođeno ili dizajnirano za rad na Java platformi.

Prvu stvar koju moramo napraviti da bi uopće bili u mogućnosti pristupiti bluetooth prijemniku i predajniku na našem mobilnom uređaju, moramo pitati korisnika za pristup bluetooth-u. To činimo tako da prvo zapišemo sljedeća dva retka u *manifest file* našeg programa.


```
manifest
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="net.bane.bluetoothapplication">
4
5     <uses-permission android:name="android.permission.BLUETOOTH" />
6     <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
7
```

Slika 7. Manifest file – Android studio

Izvor: izradio autor

Sljedeći korak je provjeriti u samom Java class-u, da li naš uređaj podržava bluetooth te da li ima potreban *hardware*.

```
104
105     if (!getPackageManager().hasSystemFeature(PackageManager.FEATURE_BLUETOOTH_LE)) {
106         Toast.makeText(context, this, R.string.ble_not_supported, Toast.LENGTH_SHORT).show();
107         finish();
108     }
```

Slika 8. Android studio activity

Izvor: izradio autor

Sljedeći korak je inicirati *bluetoothManager* varijablu koja za provođenje cjelokupnog Bluetooth upravljanja te za dobivanje instance BluetoothAdapteta.

```
109
110     final BluetoothManager bluetoothManager =
111         (BluetoothManager) getSystemService(Context.BLUETOOTH_SERVICE);
112     bluetoothAdapter = bluetoothManager.getAdapter();
113
114     bluetoothLeScanner = bluetoothAdapter.getBluetoothLeScanner();
115
```

Slika 9. Bluetooth manager

Izvor: izradio autor

Sljedeći korak jest da pokrenemo *onResume* naredbu te u njoj prvo provjerimo je li bluetooth uključen. Ako nije, koristimo naredbu Intent koja će pokrenuti dialog koji će nas upitati dali želimo uljučiti bluetooth te u nastavku skenirati BLE uređaje u blizini pozivom naredbe `scanLeDevice(true)`. Ukoliko je bluetooth na našem mobilnom uređaju bio uključen, tada se zaobilazi if naredba (`if (!bluetoothAdapter.isEnabled())`). Naredba `scanLeDevices()` jest naredba koju smo sami inicializirali te ćemo u nastavku proučiti što ona radi. *OnResume* naredba jest naredba koja čeka interakciju korisnika ili bilo kakvu promjenu stanja unutar Java activity class-a, nakon koje će se pripadajuće naredbe unutar nje izvršiti.

```

123
124     @SuppressWarnings("MissingPermission")
125     @Override
126     protected void onResume() {
127         super.onResume();
128         // Ensures Bluetooth is enabled on the device. If Bluetooth is not currently enabled,
129         // fire an intent to display a dialog asking the user to grant permission to enable it.
130         if (!bluetoothAdapter.isEnabled()) {
131             Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
132             startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);
133         }
134
135         // Initializes list view adapter.
136         scanLeDevice(enable: true);
137     }

```

Slika 10. Poziv "scan" funkcije

Izvor: izradio autor

Sljedeća naredba koja će biti analizirana u ovom završnom radu jest *scanLeDevice()* naredba. Možemo primjetiti da kao varijablu u *scanLeDevice()* šaljemo *final boolean* pod nazivom *enable*. Ovisno da li je vrijednost varijable *true* ili *false*, *bluetoothAdapter* pokreće ili zaustavlja skeniranje BLE uređaja u okolini. *startLeScan()* naredba pokreće skeniranje BLE uređaja dok naredba *stopLeScan()* zaustavlja traženje/skeniranje uređaja.

```

2 usages
157     @SuppressWarnings("MissingPermission")
158     private void scanLeDevice(final boolean enable) {
159         if (enable) {
160             // Stops scanning after a pre-defined scan period.
161             handler.postDelayed(new Runnable() {
162                 @SuppressWarnings("MissingPermission")
163                 @Override
164                 public void run() {
165                     scanning = false;
166                     bluetoothAdapter.stopLeScan(mLeScanCallback);
167                     invalidateOptionsMenu();
168                 }
169             }, SCAN_PERIOD);
170
171             scanning = true;
172             bluetoothAdapter.startLeScan(mLeScanCallback);
173         } else {
174             scanning = false;
175             bluetoothAdapter.stopLeScan(mLeScanCallback);
176         }
177         invalidateOptionsMenu();
178     }

```

Slika 11. Skeniranje BLE uređaja

Izvor: izradio autor

```

283     if (deviceAddress.equals("00:E4:4C:00:2B:38")) {
284         final Intent intent = new Intent(packageContext, ConnectDeviceActivity.this, DeviceControlActivity.class);
285         intent.putExtra(DeviceControlActivity.EXTRAS_DEVICE_NAME, device);
286         intent.putExtra(DeviceControlActivity.EXTRAS_DEVICE_ADDRESS, deviceAddress);
287         if (scanning) {
288             bluetoothAdapter.stopLeScan(mLeScanCallback);
289             scanning = false;
290         }
291         startActivity(intent);
292         finish();
293     }

```

Slika 12. Uspostava veze s BLE modulom

Izvor: izradio autor

Sa prethodne slike možemo vidjeti da ako je adresa uređaja s kojom se želimo povezati “00:E4:4C:00:2B:38“, aplikacija će nas odvesti na novi Java activity class pod nazivom DeviceControlActivity.class.

```

2 usages
92     private final ServiceConnection mServiceConnection = new ServiceConnection() {
93
94         @Override
95     public void onServiceConnected(ComponentName componentName, IBinder service) {
96         mBluetoothLeService = ((BluetoothLeServices.LocalBinder) service).getService();
97         if (!mBluetoothLeService.initialize()) {
98             Log.e(TAG, "Unable to initialize Bluetooth");
99             finish();
100        }
101        // Automatically connects to the device upon successful start-up initialization.
102        mBluetoothLeService.connect(mDeviceAddress);
103    }
104
105    @Override
106    public void onServiceDisconnected(ComponentName componentName) {
107        mBluetoothLeService = null;
108    }
109 };

```

Slika 13: Uspostava veze s BLE modulom

Izvor: izradio autor

ServiceConnection() jest sučelje za praćenje stanja aplikacijske usluge. U naredbi “*mBluetoothLeService.connect(mDeviceAddress)*“ vrši se povezivanje s uređajem čija je adresa jednaka onoj koju smo poslali iz prethodnog Java activity class-a.

Sa sljedeće slike možemo vidjeti kako šaljemo podatke s našeg mobilnog uređaja na naš prethodno povezani bluetooth modul.

```

587 @ private void makeChange(String str) {
588     Log.d(TAG, msg: "Sending result=" + str);
589     final byte[] tx = str.getBytes();
590
591     Toast.makeText(mBluetoothLeService, text: "A", Toast.LENGTH_SHORT).show();
592     if(mConnected) {
593
594         Toast.makeText(mBluetoothLeService, text: "B", Toast.LENGTH_SHORT).show();
595         characteristicTX.setValue(tx);
596         mBluetoothLeService.writeCharacteristic(characteristicTX);
597         //mBluetoothLeService.setCharacteristicNotification(characteristicRX, true);
598     }
599 }

```

Slika 14: Komunikacija BLE modula i mobilnog uređaja

Izvor: izradio autor

Bitno je napomenuti da tekst koji želimo odaslati sa našeg mobilnog uređaja šaljemo u obliku skupa bajtova.

U nastavku možemo vidjeti na koji način čitamo znakove na Arduino Nano mikrokontroleru te kako te znakove pretvaramo u informaciju tj. tekst.

```

21 void recvWithEndMarker() {
22     static byte ndx = 0;
23     char endMarker = '\n';
24     char rc;
25
26     while (Serial.available() > 0 && newData == false) {
27         rc = Serial.read();
28         if (rc != endMarker) {
29             Serial.println(rc);
30             receivedChars[ndx] = rc;
31             ndx++;
32             if (ndx >= numChars) {
33                 ndx = numChars - 1;
34             }
35         }
36         else {
37             receivedChars[ndx] = '\0'; // terminate the string
38             ndx = 0;
39             newData = true;
40         }
41     }
42 }

```

Slika 15. Obrada podataka u Arduino Nano mikrokontroleru

Izvor: izradio autor

Prvo postavimo varijable koje će nam biti potrebne:

- ndx – varijabla koja odgovara broju članova elemenata teksta (npr. za riječ world će nakon provođenja while petlje iznositi 4).
- endMarker – predstavlja char koji naznačuje kraj informacije koju šaljemo sa mobilnog uređaja (u ovom slučaju odabrao sam '\n' što u programerskom svijetu znači novi redak).
- rc – varijabla u koju pohranjujemo odaslani char.

Princip rada ove petlje jest da pohrani primljeni znak (*character*) kao varijablu *rc* (to radimo u sljedećem retku – *rc = Serial.read()*), zatim pod određenim indexom pohranjujemo taj znak na određenu/željenu poziciju. Za prvi prolazak kroz petlju index '*ndx*' će iznositi 0 te će se prvi char pohraniti na poziciju 0. Zatim povećavamo index *ndx* za jedan (*ndx++*;) za sljedeći prolazak kroz petlju.

Bitno je napomenuti da se *while* petlja izvršava pod dva uvjeta. Prvi uvjet jest taj da je '*Serial.available() > 0*'. *Serial.available()* naredba nam ustvari govori da li je na komunikacijskom vodiču koji povezuje Arduino Nano mikrokontroler i bluetooth modul prisutan podatak tj. informacija. Kada komuniciramo našim mobilnim uređajem s bluetooth modulom, šaljemo određene informacije tek nakon interakcije korisnika s mobilnom aplikacijom tj. pritiskom na gumb unutar mobilne aplikacije, što predstavlja vrlo kratki vremenski interval slanja podataka. Svo ostalo vrijeme, bluetooth modul sluša odaslane signale s našeg mobilnog uređaja međutim nema što čuti te kao rezultat koda *Serial.available()* dobijemo 0. Drugi uvjet jest taj da nam je varijabla *newData* jednaka *false*. To nam je vrlo bitno jer tako Arduino Nano mikrokontroler zna kad završava određena informacija tj. riječ te dalje nastavlja s njom određen proces. Dok nam pročitani znak ne bude bio jednak '\n', *newChar* varijabla će iznositi *false*. Kada pročitani char postane jednak '\n', varijabla *newChar* promjeniti će vrijednost u *true* te će završiti *while* petlja.

4.2. RAZVOJ iOS APLIKACIJE

Za razvoj iOS aplikacije potreban nam je softver pod nazivom xCode. Ovaj program je razvijen od tvrtke Apple. Xcode pruža sve alate za izradu aplikacija (dizajn, razvoj i objavljivanje) za sve Appleove platforme: iOS, iPadOS, tvOS, watchOS i macOS. Osim toga, Xcode podržava izvorni kod za mnoge popularne programske jezike, uključujući Swift, Objective-C, Objective-C++, C, C++, Java, Python i druge. Xcode je jedini službeni alat za izradu i objavljivanje aplikacija na Apple App Storeu. Nova verzija Xcode-a (Xcode 14.0.1) dizajnirana je da bude lakša i brža od prethodnih verzija i omogućuje razvojnim programerima stvaranje višeploformskih aplikacija sa Swiftom i SwiftUI-om.

```

93     func centralManagerDidUpdateState(_ central: CBCentralManager) {
94         if central.state == CBManagerState.poweredOn {
95             print("BLE powered on")
96             // Turned on
97             //-----central.scanForPeripherals(withServices: nil, options: nil)
98         } else {
99             print("Something wrong with BLE")
100             // Not on, but can have different issues
101         }
102     }
103 }

```

Slika 16. Provjera bluetooth stanja mobilnog uređaja

Izvor: izradio autor

Funkcija `centralManagerDidUpdateState` govori središnjem upravitelju da li uređaj uopće sadrži hardverski dio potreban za uspostavu Bluetooth konekcije te da li je on upaljen za korištenje. Upravo to provjeravamo if naredbom koju možemo primjetiti na prethodnoj slici. U slučaju da je Bluetooth na našem mobilnom uređaju uključen, biti ćemo obavješteni porukom “BLE powered on“. U protivnom isprintati ćemo informaciju “Something wrong with BLE“.

```

104     func centralManager(_ central: CBCentralManager, didDiscover peripheral:
105         CBPeripheral, advertisementData: [String : Any], rssi RSSI: NSNumber) {
106         if let pname = peripheral.name {
107             if pname == "HMSoft" {
108                 self.centralManager.stopScan()
109
110                 self.myPeripheral = peripheral
111                 self.myPeripheral.delegate = self
112                 self.centralManager.connect(peripheral, options: nil)
113
114                 discoverServices(peripheral: myPeripheral)
115             }
116         }
117     }

```

Slika 17. Skeniranje BLE modula

Izvor: izradio autor

Sa prethodne slike možemo primjetiti da jest sljedeći korak uspostava veze između mobilnog uređaja i bluetooth modula. U ovom slučaju kao uvjet za nastavak s procesom povezivanja koristimo naziv bluetooth modula. Ukoliko je naziv bluetooth modula jednak “HMSoft“, ostvarivanje veze će se nastaviti te će se prekinuti traženje drugih uređaja koji se nazivaju “HMSoft“. Naziv samog bluetooth modula određuje tvrtka koja ih proizvodi te ovakvo rješenje ne bi bilo idealno ukoliko želimo ovaj uređaj postaviti na tržište (može postojati veći

broj BLE uređaja pod nazivom “HMSoft“ u našoj blizini te će se naš mobilni uređaj povezati upravo s prvim kojeg otkrije).

```
140     func peripheral(_ peripheral: CBPeripheral, didDiscoverServices error:
        Error?) {
141         guard let services = peripheral.services else { ▲ Value 'services' was de...
142             return
143         }
144         discoverCharacteristics(peripheral: peripheral)
145     }
146
147     func peripheral(_ peripheral: CBPeripheral, didDiscoverCharacteristicsFor
        service: CBService, error: Error?) {
148         guard let characteristics = service.characteristics else { ▲ Value 'c...
149             return
150         }
151
152         for characteristic in service.characteristics! {
153             txCharacteristic = characteristic
154         }
155     }
156
157     func write(value: Data, characteristic: CBCharacteristic) {
158         self.myPeripheral?.writeValue(value, for: characteristic, type:
            .withoutResponse)
159     }
```

Slika 18: Uspostava veze i komunikacija s BLE modulom

Izvor: izradio autor

Sa slike br. 16, vrlo je važna funkcija *write* koju ćemo pozvati kada želimo emitirati određenu informaciju sa našeg mobilnog uređaja.

4.3. KONAČNI REZULTAT

Kada su se sve prethodno navedene naredbe izvršile, ovisno da li koristimo Android ili iOS mobilni uređaj, te nakon što je Arduino kontroler primio zadovoljavajuće znakove, silazno / ulazni most na brodu kretati će se u željenom smjeru. Silazno / ulazni most na brodu može se kretati u četiri smjera (uvlačenje, izvlačenje, podizanje i spuštanje). U ovom završnom radu, znak 'U', kojeg će zaprimiti Arduino mikrokontroler, korišten je kao znak koji će pokrenuti postupak podizanje silazno / ulaznog mosta . Znak 'D' korišten je kao znak koji će izvršiti naredbu spuštanja silazno / ulaznog mosta, te znakovi 'L' i 'R' korišteni su za izvršavanje uvlačenja i izvlačenja silazno / ulaznog mosta. Također, prilikom slanja određenog od prethodo navedenih znakova, u istoj informaciji odašilje se četveroznamenkasti broj koji služi kao '*safety key*'. Ukoliko '*safety key*' kojeg smo odaslali odgovara postavljenom '*safety key-u*' zapisanom na Arduino mikrokontroleru, željena radnja će se izvršiti.

6. ZAKLJUČAK

Ovaj završni rad temelji se na urađenom projektu te je u osnovi sažetak svih postupaka i tehnologija koje su korištene za razvoj digitalnog kontrolera za silazno / ulazni most na brodu. Bluetooth tehnologija u novije vrijeme postaje sve više i više popularna u pogledu bežičnih kontrolera te brojne male tvrtke koriste prednosti ove tehnologije.

Isto tako razvoj mobilnih aplikacija obuhvaća veliko područje te iziskuje mnogo iskustva i rada da bi određen projekt mogli privesti kraju. Mobilna aplikacija napravljena za ovaj projekt (Android i iOS aplikacija) nije potpuno opisana u ovom završnom radu te je naveden i obrazložen samo onaj dio aplikacije koji vrši uspostavu veze s bluetooth modulima te komunikaciji samog mobilnog uređaja i bluetooth modula.

LITERATURA

- [1] Inigo Puy, 05.05.2008, 'Bluetooth', p. 5-12
- [2] Jakob Iversen i Michael Eierman, 2014, 'Learning Mobile App Development', izdavač : Pearson Education, Inc.
- [3] Mark Geddes, 2016, 'Arduino Project Handbook', izdavač : No Starch Press, Inc., p.229-240
- [4] David watson, 'Introduction to Arduino', online : <https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-nano.html>
- [5] Electrow, 'Arduino Nano', pharagraph: *Features i Specifications*, online : [https://www.elecrow.com/wiki/index.php?title=Nano_328\(Arduino_Compatible\)](https://www.elecrow.com/wiki/index.php?title=Nano_328(Arduino_Compatible))
- [6] Arduino Store, 'Arduino Nano', online : <https://store.arduino.cc/products/arduino-nano>
- [7] DSD TECH, 'HM-10 DataSheet', p 5-23, online : <https://people.ece.cornell.edu/land/courses/ece4760/PIC32/uart/HM10/DSD%20TECH%20HM-10%20datasheet.pdf>
- [8] Cypress, ' Bluetooth Low Energy (BLE)', p 1-18, online : [https://www.infineon.com/dgdl/Infineon-Component_Datasheet_Bluetooth_Low_Energy_\(BLE\)_3.53-Software+Module+Datasheets-v03_66-EN.pdf?fileId=8ac78c8c7d0d8da4017d0eaf6f052a2d&utm_source=cypress&utm_medium=referral&utm_campaign=202110_globe_en_all_integration-files](https://www.infineon.com/dgdl/Infineon-Component_Datasheet_Bluetooth_Low_Energy_(BLE)_3.53-Software+Module+Datasheets-v03_66-EN.pdf?fileId=8ac78c8c7d0d8da4017d0eaf6f052a2d&utm_source=cypress&utm_medium=referral&utm_campaign=202110_globe_en_all_integration-files)
- [9] Spec India, 'Top 14 JVM Languages To Choose From For Your Software Development', online : <https://www.spec-india.com/blog/jvm-languages>
- [10] Bluetooth Transmission Technology, Chapter 2, p 6-21, online : <https://vtechworks.lib.vt.edu/bitstream/handle/10919/35741/Chapter2.pdf?sequence=5>

POPIS SLIKA

Slika 1. Prikaz frekvencija za odašiljanje, Izvor: Bluetooth Transmission Technology.....	5
Slika 2. Prikaz paketa, Izvor: Bluetooth Transmission Technology	5
Slika 3. Električna šema kontrolera, Izvor: izradio autor.....	12
Slika 4. Raspored elemenata i vodiča kontrolera, Izvor: izradio autor	13
Slika 5. 3D prikaz digitalnog kontrolera, Izvor: izradio autor	14
Slika 6. Prikaz digitalnog kontrolera, Izvor: izradio autor.....	14
Slika 7. Manifest file – android studio, Izvor: izradio autor	16
Slika 8. Android studio activity, Izvor: izradio autor.....	16
Slika 9. Bluetooth manager, Izvor: izradio autor	16
Slika 10. Poziv “scan“ funkcije, Izvor: izradio autor.....	17
Slika 11. Skeniranje BLE uređaja, Izvor: izradio autor	18
Slika 12. Uspostava veze s BLE modulom, Izvor: izradio autor	18
Slika 13. Uspostava veze s BLE modulom, Izvor: izradio autor	19
Slika 14. Komunikacija BLE modula i mobilnog uređaja, Izvor: izradio autor	19
Slika 15. Obrada podataka u Arduino Nano mikrokontroleru, Izvor: izradio autor	20
Slika 16. Provjera bluetooth stanja mobilnog uređaja, Izvor: izradio autor.....	21
Slika 17. Skeniranje BLE modula, Izvor: izradio autor	22
Slika 18. Uspostava veze i komunikacija s BLE modulom, Izvor: izradio autor	22