

PRAĆENJE LOKACIJA OPREME POMOĆU TEHNOLOGIJE RADIOFAROVA

Papić, Tomislav

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Algebra
University College / Visoko učilište Algebra**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:225:842693>

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

Download date / Datum preuzimanja: **2024-11-29**



Repository / Repozitorij:

[Algebra University - Repository of Algebra University](#)



VISOKO UČILIŠTE ALGEBRA

DIPLOMSKI RAD

**Praćenje lokacija opreme pomoću
tehnologije radiofarova**

Tomislav Papić

Zagreb, veljača 2020.

Predgovor

Srdačno se zahvaljujem mentoru dr.sc Goranu Đambiću koji mi je stručnim savjetima vodstvom pomogao u rješavanju mnogih dilema s kojima sam se susreo tijekom izrade završnog rada.

Sažetak

Cilj ovog rada je pokazati kako pomoću tehnologije radiofarova (engl. *beacons*) možemo pratiti lokaciju opreme, prije svega u zatvorenim prostorima, iako nema prepreka da se isti princip može primijeniti i na otvorenom. Radiofarovi su temeljeni na BLE¹ tehnologiji i često se koriste u kombinaciji s mobilnim uređajima. Hipoteza koju želimo dokazati je da se pomoću ovakvog pristupa može uspješno locirati dio opreme do preciznosti od 1 m i na taj način znatno efikasnije pratiti njihovu trenutnu i povijesnu lokaciju.

Za određivanje pozicije koristit će se trilateracija, jednako kao što se koristi i u GPS²-u, samo što će se u ovom radu koristiti snaga signala (RSSI³). Ovakav rad može se i komercijalizirati, primjerice, za praćenje lokacije skupljeg alata u nekom tvorničkom postrojenju.

Kao praktični dio rada dizajnirat će se i izraditi softverski i hardverski sustav. Nakon završetka, testiranje hipoteze će se napraviti u konkretnim uvjetima (funkcionalnosti sustava će se testirati u prostorima Algebra LAB-a i drugim adekvatnim prostorima za testiranje sustava).

Naposljetku će se analizirati preciznosti sustava, a hipoteza će ili biti potvrđena ili će se dati smjernice za poboljšanje preciznosti sustava.

Ključne riječi: radiofarovi; lokacija; navigacija; trilateracija; zatvoreni prostor

¹ BLE (engl. *bluetooth low energy*) je varijacija standarda Bluetooth koja ima manju potrošnju energije. <https://www.rfwireless-world.com/Terminology/Bluetooth-vs-BLE.html>

² GPS (engl. *global positioning system*) je globalni navigacijski sistem koji omogućuje korisnicima pomoću uvid u trenutnu poziciju <https://www.gps.gov/systems/gps/>

³ RSSI (engl. *received signal strength indicator*) je mjera koja označuje kvalitetu radio signala <https://www.metageek.com/training/resources/understanding-rssi.html>

Summary

The goal of this master thesis is to show how to track the location of the equipment using the technology of beacons both in closed spaces and outdoors if there are no obstacles. Beacons are based on BLE technology and are often used with mobile devices. The hypothesis that is to be proved is that using this technology it is possible to successfully locate the equipment with the precision of 1 meter and to track the current and previous location more efficiently. The location is determined via the trilateration, the method used in GPS, with the difference that in this case the received signal strength indication (RSSI) is used. This method can be commercialized for example for tracking the location of the expensive equipment in a factory plant.

In the practical part of the thesis, the software and hardware will be designed. After finishing, the hypothesis will be tested in real conditions (Algebra LAB and other adequate spaces for testing). In the end, the precision of the system will be tested and according to the results, the hypothesis will be confirmed or the guidelines will be given to improve the system.

Key words: beacons; location; navigation; trilateration; indoor;

Sadržaj

1. Uvod	1
2. Projektni zadatak	3
2.1. Opis projektnog zadatka	3
2.2. Hipoteza i opis testova za njeno validiranje	4
3. Tehnologije potrebne za izgradnju sustava	5
3.1. Osnovni pojmovi	5
3.2. Računalo Raspberry Pi	6
3.3. Računalo ESP32	9
3.4. Protokol MQTT	10
3.5. BLE i radiofarovi.....	16
3.6. Određivanje pozicije postupkom trilateracije.....	19
4. Metodika rada.....	23
4.1. Sklopovlje.....	23
4.1.1. Uloga i radno okruženje na Raspberry Pi.....	24
4.1.2. Postavljanje testnih uvjeta u zatvorenom prostoru	27
4.2. Softverske komponente	28
4.2.1. Programiranje ESP32 mikrokontrolera	29
4.2.2. Računanje trilateracije na klijentskoj strani.....	31
5. Testiranje i analiza.....	33
5.1. Opis provedenih testova	33
5.2. Analiza pogreške i validiranje hipoteze	35
Zaključak	41
Popis kratica	42
Popis slika.....	44

Popis tablica.....	45
Popis kôdova	46
Reference	47
Prilog	48

1. Uvod

Čovjek od samog početka postojanja traži načine kako da odredi svoju lokaciju i rješenja kako bi brže, jednostavnije i efikasnije došao na odredište. Najjednostavnija navigacija koja se i danas koristi je usmena predaja.

Kroz stoljeća ljudi su se orijentirali pomoću zvijezda, sunca i ostalih prirodnih pojava. S vremenom su ljudi izrađivali karte i mape kako bi prenosili znanje i otkrića drugim naraštajima. Jedan od najbitnijih uređaja koji je čovjek izumio kako bi odredio svoj smjer je kompas. Danas postoje puno sofisticiranije tehnologije koje nam omogućuju da preciznije i brže odredimo svoju lokaciju.

Jednu od tih tehnologija razvila je američka vojska početkom 80-ih godina prošlog stoljeća, a to je GPS. Osim GPS-a postoje i drugi GNSS⁴-ovi poput ruskog Glonnasa, europskog Galilea i kineskog BeiDoua koji je u upotrebi od 2016. godine. Svi ti sustavi koriste satelite koji se nalaze u zemljinoj orbiti, a uređaji koji koriste njihovu tehnologiju komuniciraju pomoću radio valova. Dva glavna nedostatka ovih sustava leže u tome, da ako se želi odrediti velika preciznost lokacije, sustav troši puno električne energije te se ne može koristiti u zatvorenim prostorima [1].

S vremenom se razvila potreba da se nađe način kojim bi se moglo locirati predmete u zatvorenim prostorima, s većom preciznošću i s manjom potrošnjom električne energije. Tu dolazi tehnologija radiofarova koja pomaže u rješavanju tih problema. Stoga će u ovom radu biti objašnjena jedna od primjena ove tehnologije radiofarova u zatvorenim prostorima.

U prvom poglavlju će se opisati projektni zadatak, postavljanje hipoteze te opis testova koji će validirati tu hipotezu. U drugom poglavlju bit će opisane tehnologije koje su potrebne za izgradnju sustava. Treće poglavlje bavit će se obrađivanjem metodike rada

⁴ GNSS(engl. *global navigation satellite system*) je globalni navigacijski sistem je termin koji opisuje sve one satelitske sisteme koji pružaju usluge pozicioniranja, navigacije itd.
<https://www.gps.gov/systems/gnss/>

tj. koje je sve radnje potrebne napraviti kako bi sustav radio. Naposljetku će se sustav testirati i analizirati kako bi se hipoteza mogla validirati.

2. Projektni zadatak

U ovoj dijelu rada će se objasniti i opisati što se želi postići ovim radom, koji su glavni i sporedni ciljevi rada te će se opisati hipoteze koje se želi dokazati. Također će biti opisani i testovi kojima će se validirati hipoteza.

2.1. Opis projektnog zadatka

Ovim radom se želi pokazati kako pomoću tehnologije radiofarova (engl. *beacons*) možemo pratiti lokaciju opreme, prije svega u zatvorenim prostorima, iako nema prepreka da se isto može raditi i na otvorenom. Cilj ovog rada, odnosno, hipoteza koja se želi dokazati je da se uz pomoć ovog pristupa može uspješno locirati dio opreme do preciznosti od 1 m i na taj način pratiti trenutnu i povijesnu lokaciju opreme.

Rješenjem ovog projektnog zadatka bit će realiziran i priložen dizajn te izrađen hardverski dio potreban za funkcioniranje projekta. Osim hardverskog, izradit će se i softverski dio arhitekture koja će biti u obliku web aplikacije sa svojim *frontend* i *backend* komponentama potrebnim za realizaciju rada. Nakon završetka izrade sustava, testiranje postavljene hipoteze će se odraditi u konkretnim uvjetima. Funkcionalnost sustava testirat će se u prostorima Algebra LAB-a i u drugim prostorima koji odgovaraju potrebama za testiranje.

Način na koji će se testirati je taj da će se uređaji koji primaju signal postaviti na dobro poznata mjesta u prostoriji tako da pokrivaju cijeli prostor. Radiofarovi će biti pričvršćeni na dijelove same opreme te će se zajedno s opremom pomicati kroz prostor. Primatelji signala očitavat će jačinu signala putem radio valova te će se po njoj izračunavati pozicija tog radiofara, a time i opreme. Na sučelju web aplikacije moći će se pratiti gdje se taj radiofar nalazi u prostoriji. Za svaki test će se izračunati odstupanje u metrima od stvarne pozicije što će na kraju koristiti za potvrdu ili opovrgavanje hipoteze.

2.2. Hipoteza i opis testova za njeno validiranje

Hipoteza koju želimo dokazati je da se pomoću ovakvog pristupa može uspješno locirati dio opreme do preciznosti od 1 m, te na taj način efikasnije pratiti njihovu trenutnu i povijesnu lokaciju. Za određivanje pozicije koristit će se trilateracija, jednako kao što se koristi i u GPS-u, samo što će se u ovom radu koristiti snaga signala (RSSI).

Na kraju rada će se napraviti analiza preciznosti sustava te će se ili zaključiti da je hipoteza potvrđena ili će se analizirati uzroci neuspjeha te dati smjernice za poboljšanje preciznosti sustava. Kako bi se hipoteza potvrdila ili bila opovrgnuta potrebnu su testovi koji će je validirati.

Prvi test koji će se napraviti pokazat će kolika su odstupanja od pravih vrijednosti tj. koliko su odstupanja prikazani na sučelju aplikacije i stvarnih vrijednosti iz prostorije.

Drugi test prikazat će koliko promjena konstante za okoliš utječe na izračun pozicije tj. hoće li izmjerena greška biti manja te koji je indeks najprikladniji za tu prostoriju. Konstante će se odabrati u rangu koji je predviđen u tom okolišu odnosno prostoru. Prostor u kojem će se obavljati testiranja ima i prepreke što će se uzeti u obzir kod testiranja, i prema tome odrediti onu konstantu koja najbolje odgovara tom prostoru.

3. Tehnologije potrebne za izgradnju sustava

U nastavku su opisane tehnologije koje će se koristiti u ovom radu. Tehnologije se sastoje od hardvera i softvera, u nastavku ovog poglavlja će se opisati hardver, a naposljetku je opisan također i softver.

3.1. Osnovni pojmovi

Ovakav koncept gdje se elektronički uređaji povezuju putem mreže kako bi razmjenjivali podatke naziva se Internet stvari (engl. IoT – *Internet of things*). Svrha IoT-a je da se olakšaju svakodnevne rutinske radnje kod kuće i na poslu. Primjene mogu biti različite u kontekstima medicine, turizma, logistike, industrije itd.

U ovom radu koristit će se raznoliku skup tehnologija. Osnova svakog IoT sustava je mikrokontroler. Mikrokontroler je obično minijaturno računalo smješteno na čipu (integriranom krugu) koje se koristi za specijalizirane aplikacije te je dizajnirano za izvršavanje ili provedbu strogo određenih zadataka. Sadrži svoju RAM memoriju, programibilnu memoriju, digitalne ulaze i izlaze, itd. Dizajniran je tako da se može ugrađivati u različite elektroničke uređaje i sisteme [2].

Mikrokontroler koji se koristi u ovom radu uz navedena svojstva također sadrži i ugrađen prijamnik koji može primati i odašiljati radio valove. Primljene radio valove mikrokontroler dekodira i potom ih obrađuje. Tehnologija koja će se koristiti i koja koristi te radio valove je ugrađena u sam mikrokontroler je Bluetooth⁵.

Uređaj koji emitira radio valove zove se radiofar (engl. *beacon*). Radiofar je uređaj koji koristi BLE tehnologiju kako bi oglasio svoje postojanje BLE prijammnicima, a u ovome radu će se koristiti za određivanje pozicije u zatvorenim ili otvorenim prostorima. Takvi uređaji često se upotrebljavaju iz razloga što imaju dobru preciznost u određivanju lokacija i financijski su isplativi tj. imaju relativno nisku cijenu te njihovo održavanje nije

⁵ Bluetooth – je standard bežične komunikacije koja omogućava elektroničkim uređajima da se spajaju i međusobno komuniciraju <https://www.androidpit.com/what-is-bluetooth>

kompleksno [3]. U ovom radu radiofarovi će komunicirati s mikrokontrolerom pomoću BLE protokola te će mikrokontroler obrađivati primljene podatke.

3.2. Računalo Raspberry Pi

Raspberry Pi je serija malih računala koja su razvijena i proizvedena u Ujedinjenom Kraljevstvu od strane Raspberry Pi fondacije koja promovira učenje osnovne računalne znanosti u školama i među entuzijastima. Prvi Raspberry Pi uređaj je proizveden 2012. godine i imao je ARM⁶ arhitekturu procesora od 700 MHz te RAM memoriju od 512 MB, dva USB konektora, HDMI konektor, audio konektor, Ethernet konektor i 26 GPIO pinova. Taj model nije imao ugrađen Wi-Fi niti Bluetooth pa su te funkcionalnosti bile postizane dodatnim hardverom. Dodatno, ako se uz Wi-Fi i Bluetooth module željelo priključiti miš i tipkovnicu to nije bilo moguće jer je originalni Raspberry Pi imao samo dva USB priključka. Rješenje tog problema je bio dodatni USB *hub* koji je od jednog USB konektora napravio više USB konektora. U sljedećim verzijama broj USB konektora se povećao na četiri, te su Wi-Fi i Bluetooth moduli ugrađeni u samu pločicu [4]. Izgled računala Raspberry Pi je prikazan na slici 3.1.



Slika 3.1 - Raspberry Pi

U najnovijoj inačici podržano je i sljedeće:

⁶ ARM(engl. *advanced RISC machine*) je arhitektura koja se koristi računalnim procesorima
<https://www.watelectronics.com/arm-processor-architecture-working/>

Slika 3.1 - https://cdn-reichelt.de/bilder/web/xxl_ws/A300/RASPBERRY_PI_3B_PLUS_001.png

- 4K video podrška
- Dva HDMI izlaza (za dupli ekran)
- Procesor Cortex-A72 sa četiri jezgre i brzinom od 1.5 GHz
- USB C konektor za napajanje

Raspberry Pi se koristiti uz pomoć operativnog sustava. Postoje razni operativni sustavi koji se mogu instalirati na Raspberry Pi, a najčešće se koristi Raspbian. Raspbian je temeljen na Debian Linux distribuciji i sadrži niz već predinstaliranih programa kao što su Scratch (za programiranje pomoću blokova) te alati za razvoj računalnih programa u programskim jezicima Python i C/C++. Također, postoji i jednostavna mogućnost instalacije alata za razvoj drugih programskih jezika poput PHP-a, Ruby-a, Jave i slično [4].

Osim Raspbiana, postoje i drugi operativni sustavi koji se mogu koristiti: Ubuntu MATE, Windows 10 IoT Core, RISC OS, OSMC, Kodi i ostali [20]. Wi-Fi i Bluetooth u verziji od 2015. godine nisu ugrađeni sve do nove verzije iz 2017. koja sadrži navedene tehnologije [5].

Jedna od najvažnijih stvari dostupnih na Raspberry Pi su GPIO (engl. *General-purpose input/output*) pinovi tj. pinovi koji omogućuju primanje i slanje podataka. GPIO pin dizajniran da bude izlazni pin može biti postavljen na napon od 3.3 V ili od 0 V. GPIO pin dizajniran da bude ulazni može očitati 3.3 V ili 0 V, dakle, to su zapravo digitalni ulazi i izlazi koji mogu biti 1 ili 0. Osim digitalnih pinova, također postoje i sljedeći pinovi:

- I2C⁷ – pinovi koji komuniciraju s drugim hardverskim modulima koji podržavaju I2C protokol.
- SPI⁸ – pinovi koji se mogu koristiti za komunikaciju sa hardverom koji podržava SPI protokol.

⁷ I2C (engl. *inter-integrated circuits*) je komunikacijski protokol za slanje podataka između elektroničkih komponenta <https://www.electronicshub.org/basics-i2c-communication/>

⁸ SPI (engl. *serial peripheral interface*) je komunikacijski protokol koji se koristi za slanje podataka između elektroničkih komponenta <http://www.circuitbasics.com/basics-of-the-spi-communication-protocol/>

- UART⁹ – pinovi koji se mogu spajati sa hardverom koji podržava UART protokol.
- PWM¹⁰ – pinovi koji mogu simulirati varijabilnu razinu napona na izlazu [6].

Raspberry Pi 3 GPIO Header

Pin#	NAME		NAME	Pin#
01	3.3v DC Power	⬇	DC Power 5v	02
03	GPIO02 (SDA1 , I ² C)	⬇	DC Power 5v	04
05	GPIO03 (SCL1 , I ² C)	⬇	Ground	06
07	GPIO04 (GPIO_GCLK)	⬇	(TXD0) GPIO14	08
09	Ground	⬇	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	⬇	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	⬇	Ground	14
15	GPIO22 (GPIO_GEN3)	⬇	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	⬇	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	⬇	Ground	20
21	GPIO09 (SPI_MISO)	⬇	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	⬇	(SPI_CE0_N) GPIO08	24
25	Ground	⬇	(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)	⬇	(I ² C ID EEPROM) ID_SC	28
29	GPIO05	⬇	Ground	30
31	GPIO06	⬇	GPIO12	32
33	GPIO13	⬇	Ground	34
35	GPIO19	⬇	GPIO16	36
37	GPIO26	⬇	GPIO20	38
39	Ground	⬇	GPIO21	40

Rev. 2
.29/02/2016

www.element14.com/RaspberryPi

Slika 3.2 - Raspberry Pi GPIO pinovi

Raspored GPIO pinova je prikazan na slici 3.2. Upravljanje GPIO pinovima se najčešće realizira kroz programske jezike Python ili C++ [6], a primjer je dan u kôdu 1:

```
import RPi.GPIO as GPIO
from time import sleep

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)
```

⁹ UART (engl. *universal asynchronous receiver-transmitter*) je komunikacijski protokol koji se koristi za slanje podataka između elektroničkih komponenata <http://www.circuitbasics.com/basics-uart-communication/>

¹⁰ PWM (engl. *pulse width modulation*) je tehnika moduliranja digitalnog signala za upravljanje različitim elektroničkih komponenata poput el. motora itd. https://www.tutorialspoint.com/arduino/arduino_pulse_width_modulation.htm

Slika 3.2 - <https://i.stack.imgur.com/KL4PZ.png>

```
GPIO.setup(17, GPIO.OUT)

while True:
    GPIO.output(17, True)
    sleep(1)
    GPIO.output(17, False)
    sleep(1)
```

Kôd 1 - Program za uključivanje i isključivanje svjetlosne diode u programskom jeziku Python

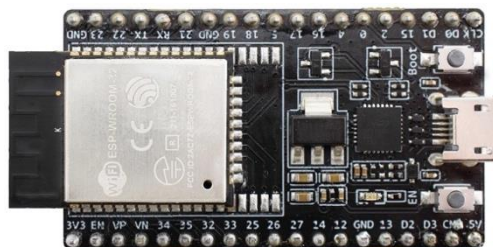
Glavna prednost Raspberry Pi-a u odnosu na slične platforme je činjenica da ima dobru zajednicu programera, dakle, veliki broj ljudi koji međusobno razmjenjuju iskustva, što znatno olakšava i učenje i profesionalni rad. Nadalje, prednosti Raspberry Pi-a su malene dimenzije same pločice koja ne zauzima puno prostora, podrška raznih operativnih sustava, mogućnost jednostavnog priključivanja kamere, ugrađeni Wi-Fi i Bluetooth moduli te podrška za 4K monitore. Također, podržava dobru i laganu programibilnu konekciju putem GPIO pinova koja omogućava i podržava jednostavnu konekciju sa raznim senzorima poput senzora temperature, vlažnosti zraka, senzora za otkrivanje pokreta i upravljanje raznim elektroničkim komponentama poput elektromotora ili OLED ekrana.

Mane Raspberry Pi-a su da ukoliko se krivo spoji neki senzor na GPIO pinove može doći do oštećenja i čak do toga da Raspberry Pi pločica više nije upotrebljiva tj. trajno se ošteti. Nema podržan analogni u digitalni konverter već se mora dodati druga komponenta kako bi se mogla koristiti ta funkcionalnost. Zbog toga što najnovija verzija ima jak procesor od 8 jezgri i 4 GB RAM-a dolazi do toga da se komponente na pločici dosta zagrijavaju, a ponajviše procesor zbog svoje snage.

3.3. Računalo ESP32

ESP32 je hardverski uređaj koji se koristi s malim naponima za razne projekte u IoT-u. Sadrži mikrokontroler koji u sebi posjeduje integrirani Wi-Fi i Bluetooth. ESP32 je

dizajniran s ciljem da troši malo energije [7, 8]. Prikaz jednog ESP32 prikazan je na slici 3.3.



Slika 3.3 - ESP32

Specifikacije ESP32 su sljedeće: 32-bitni mikrokontroler s jednom ili dvije jezgre s radnim taktom od 160 ili 240 MHz takt, ROM-om od 448 kB (koristi se za pokretanje i osnovne funkcionalnosti samog integriranog sklopa), SRAM od 520 kB (Koristi se za podatke i instrukcijski kod). Podržava do 16 MB vanjskog SRAM-a, A/D pretvarač, D/A pretvarač¹¹, I2C, UART i SPI protokole za komunikaciju s drugim hardverskim modulima te PWM. Mikrokontroler se najčešće programira uz pomoć alata Arduino IDE koristeći programski jezik C/C++ [7,8], iako postoje i druge opcije kao što su [8]:

- MicroPython – verzija Pythona 3 za mikrokontrolere
- Espressif IoT programski okvir (engl. *framework*)
- Mongoose OS – alat s jednostavnim sučeljem s kojim se može lako programirati mikrokontroler i čak spajati na servise u oblaku poput AWS IoT, Microsoft Azure IoT Core, IBM Watson IoT
- Espruino – JavaScript SDK

3.4. Protokol MQTT

MQTT (engl. *message queuing telemetry transport*) je komunikacijski protokol koji radi na principu objavi-pretplati (engl. *publish/subscribe*) i popularan je u izradi IoT projekata.

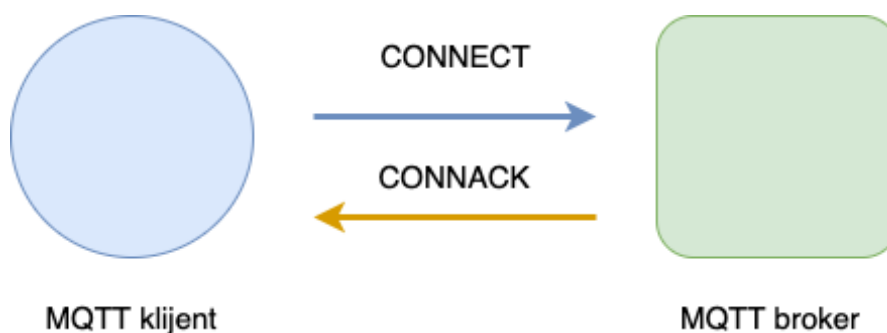
¹¹ A/D pretvarač pretvara analogni ulazni signal u digitalni dok D/A pretvarač pretvara digitalni ulazni signal u analogni signal <https://www.fpz.unizg.hr/hgold/ES/DE/a-d%20pretvara.htm>

Dizajniran je na način da bude što podatkovno lakši za slanje i primanje poruka. Koristan je za povezivanje s udaljenim lokacijama gdje su potrebne male količine kôda i gdje je mrežna propusnost važna. Popularan je iz razloga što zahtijeva malu potrošnju energije, podatkovni paketi koji se razmjenjuju su minimalizirani te što ima vrlo efektivnu distribuciju informacija prema jednom ili više primatelja [9].

MQTT protokol je osmišljen 1999. godine od strane Andy Stanford-Clarka iz IBM-a i Arlen Nipperera iz Arcom (današnji Eurotech) [9].

Što se tiče sigurnosti, može se koristiti autentifikacija pomoću korisničkog imena i lozinke. Enkripcija koja se može iskoristiti kod komunikacije kroz mrežu je SSL. Koristi TCP/IP port 1883 ili 8883 port ako koristi SSL [9].

MQTT protokol je realiziran nad TCP/IP protokolom. Broker (poslužitelj) i klijenti (uređaji) moraju imati implementiran TCP/IP stog kako bi mogli međusobno komunicirati. MQTT konekcija se uvijek ostvaruje između klijenta i brokera, dakle, klijenti nikad nisu spojeni međusobno. Konekciju započinje klijent prema brokeru slanjem poruke **CONNECT**. Broker vraća odgovor pomoću poruke **CONNACK** i status kôda. Kada je konekcija uspostavljena, broker drži tj. održava konekciju otvorenom sve dok klijent ne pošalje komandu za isključivanje ili se konekcija iz nekog razloga prekine [10].



Slika 3.4 - Slikovni prikaz uspostavljanja konekcije između MQTT klijenta i brokera

Ako **CONNECT** poruka nije dobro strukturirana ili je prošlo više vremena od dozvoljenog vremena za konekciju, broker će zatvoriti konekciju. Prikaz dobre sastavljene **CONNECT** poruke je prikazana u tablici 3.1.

MQTT poruka CONNECT sadržava	Primjer
clientId	client-1
cleanSession	true
username (opcionalan)	john
password (opcionalan)	Pass123
lastWillTopic (opcionalan)	/home/room1
lastWillQos (opcionalan)	2
lastWillMessage (opcionalan)	unexpected exit
lastWillRetain (opcionalan)	false
keepAlive (opcionalan)	60

Tablica 3.1 - Prikaz dobro sastavljene CONNECT poruke

clientId predstavlja jedinstveni identifikator kojeg klijent definira prilikom spajanja na broker. Koristeći taj identifikator, broker zna o kojem se klijentu kasnije radi.

cleanSession je podatak koji se koristi kada klijent želi uspostaviti sesiju. Ako podatak ima vrijednost false, tada će broker spremati sve pretplate klijenta i sve njegove propuštene poruke. Ako podatak ima vrijednost true, broker neće ništa spremati za klijenta već će čistiti sve poruke iz bilo koje prethodne sesije.

username i *password* omogućuju da MQTT od klijenta traži autentifikaciju i autorizaciju. Te informacije nisu kriptirane tj. šalju se kao običan tekst što predstavlja sigurnosni problem. Iz tog razloga se preporučuje prilikom slanja korisničkog imena ili lozinke koristiti SSL certifikat. Slaba sigurnost na aplikacijskoj razini je jedan od nedostataka MQTT-a.

*lastWill** definiraju postavke za obavještanje drugih klijenata kada se klijent isključi nenamjerno, a to može biti zbog loše konekcije, slabe baterije, pucanje konekcije itd.

keepAlive definira vrijeme u sekundama u kojem klijent određuje i komunicira sa brokerom kada je konekcija uspostavljena. Taj interval predstavlja najduži period vremena u kojem broker i klijent mogu podnijeti bez da su razmijenili poruku. Klijent kako bi provjerio da li je u tom intervalu šalje zahtjev prema brokeru, a broker mu vraća odgovor na taj zahtjev, koji potvrđuje da je veza otvorena ili zatvorena[10].

Nakon što je klijent poslao **CONNECT** poruku, broker je prima i odgovara sa **CONNACK** porukom. **CONNACK** poruka sastoji se od dva podatkovna elementa:

- Zastavica (engl. *flag*) već prisutne sesije – govori klijentu ima li broker već njegovu sesiju dostupnu od prijašnjih interakcija.
- Statusni kôd – kôd koji govori klijentu o uspješnosti uspostavljanja konekcije [11]. Mogući kôdovi su prikazani u tablici 3.2.

Statusni kôd	Značenje statusnog kôda
0	Konekcija je prihvaćena
1	Konekcija je odbijena, neprihvatljiva verzija protokola
2	Konekcija je odbijena, neprihvatljiv identifikator
3	Konekcija je odbijena, broker je nedostupan
4	Konekcija je odbijena, krivo korisničko ime ili lozinka
5	Konekcija je odbijena, nije autoriziran

Tablica 3.2 - Statusni kodovi uspješnosti uspostavljanja konekcije

Slanje i primanje poruka je koncept gdje uređaj može poslati poruku na neku temu ili može biti pretplaćen na neku temu tj. onaj koji prima sve poruke koje stignu na tu temu. MQTT koncept sadrži nekoliko osnovnih koncepata, a to su slanje i primanje, poruke, teme i broker [11].

Primjer slanja i primanja poruka može biti opisan na sljedeći način:

- Uređaj A pošalje brokeru poruku na neku temu
- Uređaj B je pretplaćen na tu temu
- Uređaj B prima poruku primljenu od uređaja A

MQTT poruke su informacije koje se žele izmijeniti između uređaja. Poruke mogu biti u različitim podatkovnim formatima ili običnim tekstualnim porukama, ovisno o

kompleksnosti sadržaja same poruke. Primjerice, podatkovni format JSON¹² zadovoljava glavnu značajku MQTT protokola, a to je da poruke budu što manje veličine kako se informacije prilikom transfera ne bi izgubile zbog mogućih problema u komunikaciji (kod nestabilne konekcije često dolazi do pucanje konekcije). JSON ima strukturirane podatke i ti su podaci proslijeđeni u jednoj poruci. Prednost toga je što se podaci ne moraju slati jedan po jedan, već više podataka stane u jednu poruku [10, 11].

MQTT teme su način na koji se iskazuje interes za primanjem poruka ili se želi odrediti tko treba primiti poruke. Teme definiraju klijenti i sastoje se od jedne ili više razina odvojenim kosom crtom [11]. Primjerice, slika 3.6 prikazuje jednu moguću temu koja se koristi za lampu koja se nalazi u sobi u nekom stanu:



Slika 3.5 - Primjer MQTT teme

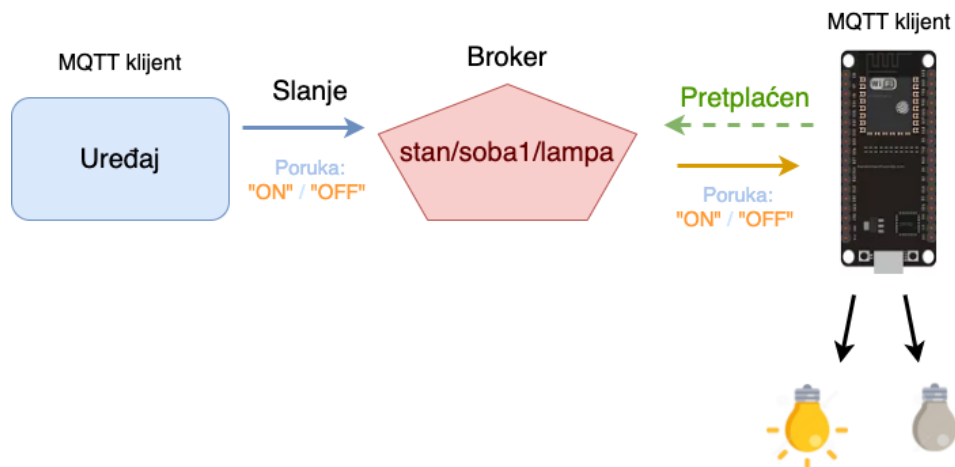
Kod pretplate je moguće pretplatiti se istovremeno na više tema pomoću zamjenskog znaka (engl. *wildcard*). Zamjenski znakovi se isključivo mogu koristiti kod pretplata, ne i kod objava poruka. Postoje dva zamjenska znaka:

- + za jednu razinu hijerarhije
- # za više razina hijerarhije

Primjerice, ako se klijent želi pretplatiti na sve lampe u nekom stanu, može koristiti temu: **stan/+lampa** gdje zamjenski znak zamjenjuje jednu razinu hijerarhije. Ako se, pak, želi pretplatiti na sve moguće senzore unutar stana, može koristiti temu: **stan/#**

Primjerice, ako se želi upaliti lampa u nekom stanu u sobi broj jedan, može se koristiti MQTT protokol, a mogući scenarij bi izgledao kao što je prikazano na slici 3.7 [11]:

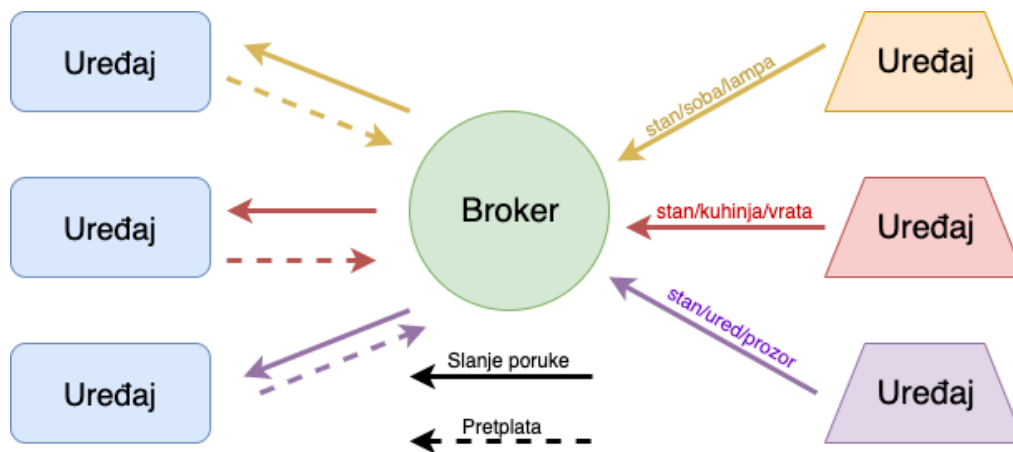
¹² JSON (engl. *JavaScript Object Notation*) je minimizirani čitljiv format kako bi se podaci prikazali u strukturiranom obliku <https://json.org/json-en.html>



Slika 3.6 - Shematski prikaz rada MQTT protokola

1. Uređaj ili MQTT klijent može biti web aplikacija, mobilna aplikacija ili bilo koji uređaj koji može slati MQTT poruke na temu **stan/soba1/lampa**.
2. U ovom primjeru, paljenjem lampe upravlja ESP32 i on je pretplaćen na temu **stan/soba1/lampa**.
3. Kada je poslana nova poruka na temu **stan/soba1/lampa**, ESP32 će primiti tu poruku koja sadrži predefimirane vrijednosti „ON“ ili „OFF“ i ovisno o sadržaju poruke će upaliti odnosno ugaziti lampu.

MQTT broker je zadužen za primanje svih poruka, filtriranje poruka i on odlučuje kome sve treba poslati poruku tj. onima koji su pretplaćeni na određenu temu [11], kako je prikazano na slici 3.8.



Slika 3.7 - Primjer spajanja uređaja na broker

3.5. BLE i radiofarovi

BLE je tehnologija za bežični prijenos podataka koja se koristi za međusobno povezivanje uređaja. S radom na BLE započela je tvrtka Nokia u sklopu projekta Wibree koji je započet 2006. godine, dok je 2010. godine Bluetooth Special Interest Group spojila Wibree u Bluetooth standard kao dio njegove četvrte verzije [12].

Glavna razlika između klasičnog Bluetootha i BLE-a je ta što BLE značajno smanjuje potrošnju energije kako bi očuvao i produžio izvor napajanja. Od popularnih platformi, podržan je na mobilnim operativnim sistemima iOS, Android i Blackberry kao i macOS-u, Linux, Windows 8 i Windows 10. Navedeni operativni sustavi sadrže već ugrađenu podršku za BLE [12].

BLE, kao i ostale verzije Bluetootha, koristi 2.4 GHz radio frekvencijsku mrežu preko koje ostvaruje dvosmjernu komunikaciju s uređajima koristeći jednu antenu. BLE je jako popularan u aplikacijama za IoT gdje su potrebne male količine prijenosa podataka te dulji vijek trajanja baterije. Primjena je raznolika od aplikacija za zdravlje, fitnesa, sigurnosti itd. [12]

Radiofarovi su mali BLE uređaji koji odašilju radio valove i napajaju se najčešće iz baterija. Kako bi se uštedila baterija, radiofarovi koriste BLE tehnologiju. Mobiteli i drugi uređajima koji podržavaju BLE primaju signal emitiran od radiofarova te na temelju

njegove snage mogu odrediti njegovu udaljenost. Radiofarovi su koncipirani tako da odašilju vrlo malu količinu podataka u određenim intervalima.

Radiofar radi na način da emitira podatke kako bi se znalo o kojem radiofaru je riječ. U tim podacima se nalaze barem dva podatka, a to su univerzalni identifikacijski broj i MAC adresa. Razlika je u tome što je MAC adresa definirana prilikom proizvodnje radiofara i nije je lako promijeniti, dok se univerzalni identifikacijski broj može puno lakše promijeniti [13].

Kako bi se što lakše približilo na koji način rade radiofarovi, može se iskoristiti primjer futurističke tržnice. Tržnica ima svoja prodajna mjesta gdje određeni proizvođači žele predstaviti i prodavati svoje proizvode. Svako od tih prodajnih mjesta ima svoj vlastiti radiofar, koji je svojom MAC adresom vezan uz prodavača.

Svaki od radiofarova odašilje signal putem radiovalova i svaka poruka sadrži MAC adresu, a domet svakog radiofara varira od 20 m do 300 m (ovisno o tipu i kvaliteti pojedinog radiofara).

Pametni mobitel kupca prima signal od najbližeg radiofara i prepoznaje pomoću MAC adrese o kojem se prodajnom mjestu, prodajnoj osobi ili kompaniji radi. Nakon odabira proizvoda korisnik plaća odabrane proizvode putem pametnog mobitela. Prilikom kupnje i nakon kupnje proizvoda kupac i prodavač mogu vidjeti je li transakcija uspješno obavljena putem obavijesti koju će dobiti putem aplikacije na pametnom mobitelu, e-mailom, SMS-om ili već nekim drugim putem.

Ovo je samo jedan od načina primjene radiofarova, a postoje različite situacije u kojima se mogu iskoristiti radiofarovi kao što su:

1. Rad u restoranima, supermarketima i slično kako bi se lakše slale dnevne ponude, promocije, itd.
2. Navigacija u malim i skućenim prostorima ili navigacija unutar supermarketa kako bi se mogao naći određeni proizvod
3. Praćenje određene opreme unutar nekog postrojenja
4. Praćenje ponašanja kupca u supermarketu, na kojim se proizvodima zadržava, koliko se vremenski zadržava na određenim policama proizvoda, itd.

Primjer radiofara je prikazan na slici 3.9. Svaki radiofar sastoji se od CPU modula koji pomoću antene odašilje BLE radio valove i baterije koje napajaju radiofar. Pojedini radiofarovi se mogu napajati i preko USB konektora. Mogu biti u raznim oblicima i bojama neki mogu sadržavati i dodatne komponente poput akcelerometra, senzora temperature itd.



Slika 3.8 - Primjer jednog radiofara

Radiofarovi se najčešće dijele po dometu [14]:

- Radiofar za unutrašnje korištenje (stanovi, uredi): domet do 80 metara
- Vanjski radiofarovi koji mogu izdržati vremenske uvjete (kiša, snijeg, visoke temperature): domet do 80 metara
- Džepni radiofar: domet do 50 metara
- Radiofar kao privjesak: domet do 25 metara
- Dalekometni radiofar: domet do 300 metara

Način na koji se određuje pozicija radiofara temelji se na jačini primljenog signala ili RSSI. Jačina signala radiofara ovisi o udaljenosti radiofara te njegovoj jačini emitiranja signala. Utjecaj na jačinu signala također imaju prepreke u prostoru koje apsorbiraju i odbijaju signale, miješanju drugih frekvencija i slično. Sve te smetnje utječu na preciznost određivanje udaljenosti radiofara.

Svaki radiofar ima određenu snagu (engl. *measured power*) kojom može emitirati signal, a snaga je određena u tvornici prilikom izrade radiofara. To je konstantna vrijednost za radiofar koja označava koliko je očekivani RSSI na udaljenosti 1 metar od radiofara [13, 14].

Radiofarovi ne emitiraju signal konstantno nego u određenim intervalima šalje signal. Što je interval manji to je signal stabilniji, no to znači veću potrošnju električne energije iz baterije. Bilo bi poželjno analizirati koji je optimalni interval za određeno mjesto gdje bi se koristili radiofarovi. Na mjestu gdje nema puno prepreka može biti veći, dok na mjestu s više prepreka bi bilo poželjno da je interval kraći kako se signal ne bi izgubio [13, 14, 15].

Formula za izračunavanje udaljenosti radiofara pomoću RSSI glasi:

$$Udaljenost = 10^{\left(\frac{Određena\ snaga - RSSI}{10 * N}\right)}$$

- Određena snaga – tvornička konstanta za jačinu signala (RSSI) od 1 metar
- RSSI – izmjerena jačina signala na ESP32 ili pametnom mobitelu
- N – konstanta koja je određena ovisno o okolišu. Vrijednosti obično idu od 2 do 4 (2 za prazni prostor, 2.7 do 4 za urbana područja, 1.6 do 1.8 ako je blizu izvora u praznom prostoru) [15, 16]

Primjerice, formula se može iskoristiti za izračunavanje udaljenosti radiofara. Neka je određena snaga radiofara -69 dBm, izmjerena jačina signala odnosno RSSI neka bude -77 dBm dok je konstanta za okoliš u kojem se radiofar nalazi 2 (prazni prostor). Iz toga slijedi da je:

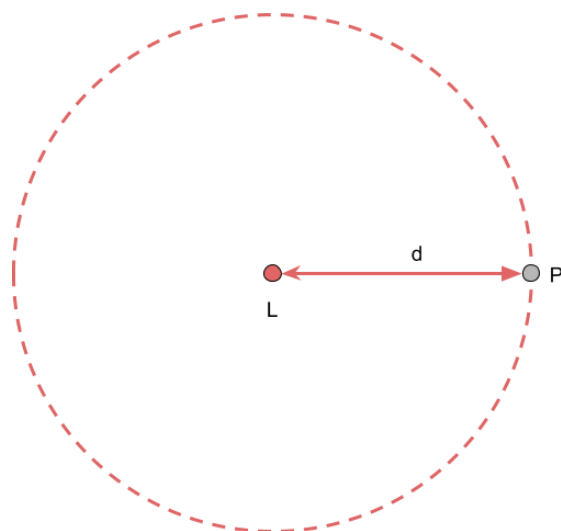
$$Udaljenost = 10^{\left(\frac{-69 - (-77)}{10 * 2}\right)} = 2.512 \text{ metara} \quad (1)$$

3.6. Određivanje pozicije postupkom trilateracije

Trilateracija je metoda računanja pozicije nekog objekta u prostoru. Da bi računanje pozicije bilo uspješno, potrebno je prikupiti barem tri različita podatka od satelita ili nekog drugog uređaja za mjerenje udaljenosti. Međutim, ako je broj mjernih uređaja veći,

tada će i sama preciznost izračuna biti točnija. Način na koji se određuje pozicija je da se nađe presjek udaljenosti svih mjernih uređaja u odnosu na objekt kojem se želi odrediti pozicija [17, 18].

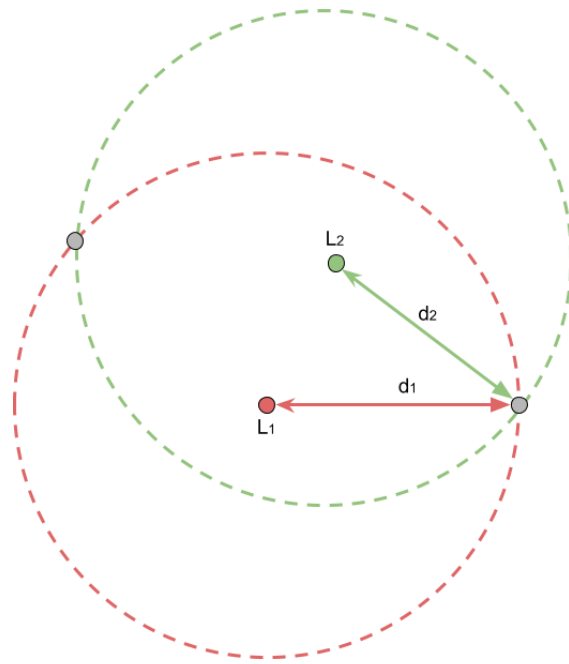
U sljedećem primjeru će se pokazati kako to funkcionira.



Slika 3.9 - Udaljenost jednog prijammika od objekta

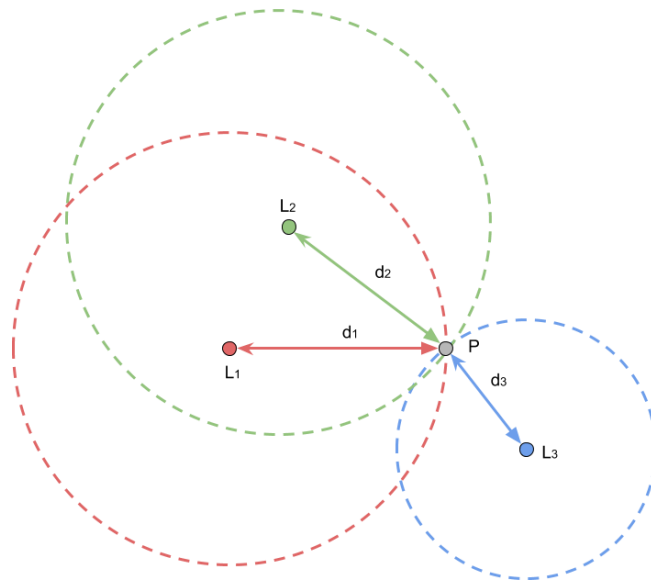
Na slici 3.10 je prikazana pozicija prijammika signala L , dok je P pozicija koju je potrebno detektirati tj. radiofar koji emitira signal. Udaljenost d određujemo putem prethodne formule. Točna lokacija nije poznata jer postoji samo jedan prijammik signala te lokacija može biti na bilo kojem mjestu na radijusu d [17, 18].

Ako tom prijammiku dodamo još jedan prijammik, situacija će se promijeniti iz razloga što je beskonačno mogućih lokacija na radijusu d smanjeno na samo dvije moguće lokacije. Na slici 3.11 se vidi o kojim presjecima tj. mogućim lokacijama se radi.



Slika 3.10 - Dva prijammika i moguće lokacije

Kako bi se precizno detektiralo o kojoj se lokaciji radi potrebno je dodati još barem jedan prijammnik koji će odrediti koji od ove dvije lokacije je tražena lokacija. Slika 3.12 prikazuje presjek tri prijammnika signala u jednoj točki P koja predstavlja lokaciju objekta. Svaki prijammnik dobiva različitu jačinu signala te je udaljenost različita i to je razlog zašto su radijusi veći ili manji od drugih [17, 18].



Slika 3.11 - Presjek svih tri prijavnika u točki P

Može se postaviti pitanje što ako dodamo još jedan prijavnik ili n dodatnih prijavnika, tj. bi li to poboljšalo preciznost u određivanju moguće lokacije. Moguće je dodati još prijavnika, ali samo izračunavanje lokacija kod trilateracije bi se promijenilo tj. matematičke formule koje računaju lokaciju bi se morale prilagoditi, no to nije predmet proučavanja i dokazivanja u ovom radu. Također, bez da se mijenja matematička formula moglo bi se ovisno o kvaliteti signala odrediti tri najbolja ili najkvalitetnija signala koja bi se iskoristila za određivanje lokacije. Takav sustav bi trebao imati poseban algoritam koji bi imao uvid u sve jačine signala od svih prijavnika i/ili na temelju povijesnih podataka tj. pomoću strojnog učenja (engl. *machine learning*) određivati tri najkvalitetnija signala koji bi se koristili za određivanje lokacije.

4. Metodika rada

U ovom poglavlju bit će opisano koji su sve sklopovi i softver potrebni za izradu ovog rada te koji su razlozi njihovih izbora. Uz to bit će opisani koraci ili postupci konfiguracije sklopovlja kako bi se praktični dio rada mogao izraditi.

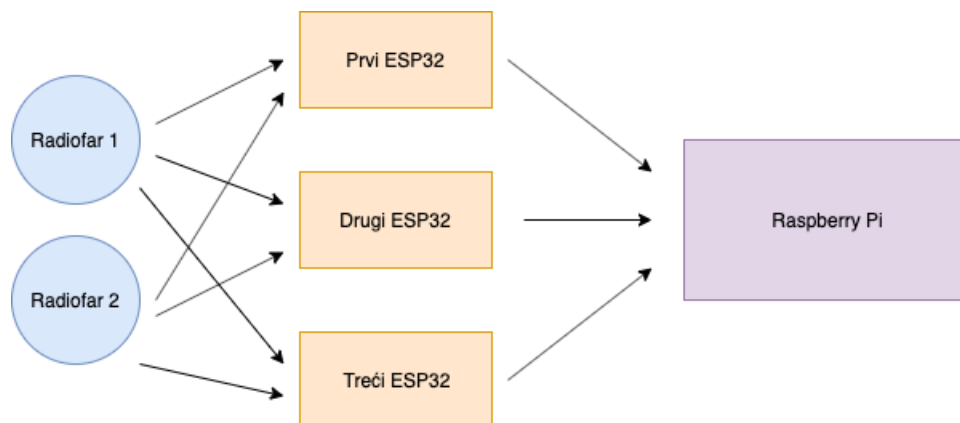
4.1. Sklopovlje

Prilikom izrade ovog rada vodilo se računa da odabrani sklop ima dobru podršku za softver koji je potreban kod izrade rada. Također, uzelo se u obzir na koji je način dokumentirana upotreba sklopa kako bi se rad mogao izraditi.

U ovom radu koriste se dva identična radiofara kupljena od istog proizvođača koji će se iskoristiti za testiranje rada projekta. Moguće je postaviti više istih radiofarova, ali za potrebe ovog rada dovoljna su samo dva. Također, moguće je postaviti više radiofarova različitih proizvođača i različitih tipova radiofarova, ali to neće biti obrađeno u ovom radu.

Kod izbora prijamnika radio valova izabran je ESP32 mikrokontroler. Glavni razlog za izbor je taj što mikrokontroler sadrži integrirani BLE i Wi-Fi modul. U radu se BLE koristi za očitavanja jačine signala radiofara, dok se Wi-Fi koristi kod prijenosa podataka u bazu. Također, ESP32 ima dobrih primjera kôdova, shema spajanja modula i dokumentacija koji su potrebni kod izrade ovakvog projekta. U ovom radu koriste se tri identična ESP32 mikrokontrolera, a razlog zašto se koristi samo tri mikrokontrolera je taj jer se pozicija određuje pomoću trilateracije koja koristi tri različita očitavanja svakog mikrokontrolera. Svaki mikrokontroler je smješten na dobro poznate lokacije u prostoriji.

Za platformu koja će u sebi čuvati, prikazivati i obrađivati podatke odabran je Raspberry Pi. Izabran je iz razloga što ima dobru dokumentaciju, podržava sve potrebne softvere za izradu rada i zadovoljava sve procesorske i memorijske kapacitete da može bez većih problema obrađivati i prikazivati podatke.



Slika 4.1 - Vizualni prikaz komunikacije sklopovlja

Na slici 4.1 prikazano je kako će u ovom radu sklopovlje komunicirati. Svaki radiofar emitira radio valove putem BLE-a. Te radio valove tj. jačinu signala (RSSI) će mjeriti svaki pojedini ESP32 za svaki radiofar. To znači da će sustav u jednom ciklusu prikupiti šest RSSI podatka (2 radiofara * 3 ESP32).

Svaki ESP32 će uz RSSI od ta dva radiofara putem MQTT protokola poslati svoju MAC adresu kako bi se znalo od kojeg ESP32 su podaci. Podaci se šalju na MQTT broker koji se nalazi u Raspberry Pi-u te se obrađuju i spremaju. Podaci će biti spremljeni u MariaDB bazu te će se spremati putem programskog jezika Python.

4.1.1. Uloga i radno okruženje na Raspberry Pi

Uloga Raspberry Pi u ovo radu je da prima podatke koje sva tri ESP32 mikrokontrolera šalju te ih sprema, obrađuje i prikazuje pomoću posebno napisanog softvera. Također, zbog svoje dobre dokumentacije i dobre podrške softvera omogućuje lakšu i kvalitetniju izradu ovog rada. Isto tako ima dobre sklopovske predispozicije poput jačine procesora i dosta veliku količinu memorije koje su potrebne za izradu ovog rada.

Prije instalacije programa koji su potrebni za razvoj rada treba se postaviti operativni sustav. U ovom radu se koristi Raspbian zbog toga što podržava sve potrebne programe za ovaj rad.

Nakon instalacije operativnog sustava potrebno je instalirati web server te bazu MariaDB. Programski jezik Python dolazi predinstaliran s operativnim sustavom te će on biti zadužen za upisivanje podataka u bazu te za Flask web okvir za razvijanje web aplikacija. Instalaciju paketa ili programa radi se kao u svim Debian distribucijama naredbom:

```
sudo apt-get install <IME_PAKETA>
```

MariaDB se instalira na sljedeći način:

```
sudo apt-get install mariadb-server
```

Nakon instalacije potrebno je konfigurirati MariaDB korisničku lozinku kako bi se aplikacija mogla spojiti na tu bazu i upisivati i dohvaćati podatke. Naredba za konfiguraciju baze je [19]:

```
sudo mysql_secure_installation
```

Prva naredba koja je potreba za instalaciju MQTT brokera je [20]:

```
sudo apt-get install mosquitto
```

Nakon instalacije brokera, poželjno je instalirati i MQTT mosquitto klijenta s kojim bi se provjerilo i testiralo radi li broker ispravno na način da se pošalje primjer poruke na neku temu. Naredba za instalaciju MQTT mosquitto klijenta je [20]:

```
sudo apt-get install mosquitto-clients
```

Nakon instalacije, potrebno je pokrenuti pretplatu klijenta na temu testTema naredbom [21], a rezultat je prikazan na slici 4.2:

```
mosquitto_sub -d -t testTema
```



```
pi@raspberrypi:~ $ mosquitto_sub -d -t testTema
Client mosqsub/1454-raspberryp sending CONNECT
Client mosqsub/1454-raspberryp received CONNACK
Client mosqsub/1454-raspberryp sending SUBSCRIBE (Mid: 1, Topic: testTema, QoS: 0)
Client mosqsub/1454-raspberryp received SUBACK
Subscribed (mid: 1): 0
```

Slika 4.2 - Primjer MQTT brokera

Opcija -d služi za prikaz primljenih poruka u terminalu, dok opcija -t definira na koju MQTT temu se klijent pretplatiti. Nakon obavljene pretplate, potrebno je napraviti novi prozor terminala u kojem će se poslati poruka „Pozdrav svijete“ na istu temu. Naredba za stvaranje MQTT klijenta i slanje poruke je [20]:

```
mosquitto_pub -d -t testTema -m "Pozdrav svijete"
```

Primjer kako izgleda slanje poruke prikazano je na slici 4.3.

```
pi@raspberrypi:~ $ mosquitto_pub -d -t testTema -m "Pozdrav svijete"
Client mosqpub/1361-raspberryp sending CONNECT
Client mosqpub/1361-raspberryp received CONNACK
Client mosqpub/1361-raspberryp sending PUBLISH (d0, q0, r0, m1, 'testTema', ...
(15 bytes))
Client mosqpub/1361-raspberryp sending DISCONNECT
```

Slika 4.3 - Slanje poruke na temu testTema

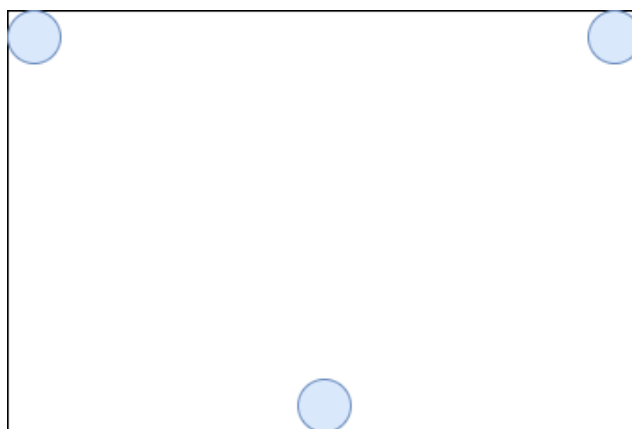
Primjer dobivanja poruke onog klijenta koje je pretplaćen da dobiva poruke koje su poslone na tu temu je prikazan na slici 4.4.

```
pi@raspberrypi:~ $ mosquitto_sub -d -t testTema
Client mosqsub/2945-raspberryp sending CONNECT
Client mosqsub/2945-raspberryp received CONNACK
Client mosqsub/2945-raspberryp sending SUBSCRIBE (Mid: 1, Topic: testTema, QoS: 0)
Client mosqsub/2945-raspberryp received SUBACK
Subscribed (mid: 1): 0
Client mosqsub/2945-raspberryp received PUBLISH (d0, q0, r0, m0, 'testTema', ... (15 bytes))
Pozdrav svijete
```

Slika 4.4 - Primjer primanja poruke na pretplaćenu temu testTema

4.1.2. Postavljanje testnih uvjeta u zatvorenom prostoru

Prijamnici implementirani pomoću ESP32 primaju signal koji odašilju radiofarovi. Kako bi se što bolje pokrila prostorija, potrebno je pravilno postaviti prijamnike. To ovisi o veličini prostorije ili površine u kojoj se želi postići mjerenje. Ta veličina prostorije također utječe prilikom izbora prijamnika jer svaki prijamnik ima svoj radijus u kojem može primiti signal. Poželjno je da se prilikom postavljanja prijamnika pazi da se ti radijusi preklapaju jer će to rezultirati boljom pokrivenošću i prilikom mjerenja signala rezultati će biti točniji. Također, uz vrstu prijamnika bitna je i vrsta radiofara. Kao što je opisano u poglavlju sa radiofarovima, jedan od važnih atributa radiofara je njihova jačina signala. U ovom radu se koriste radiofarovi koji odašilju signal maksimalno do 30 metara. Potrebno je uzeti u obzir koliko prostor ima prepreka, jer prepreke utječu na jačinu signala radiofara i prijamnika. Uz količinu potrebno je odrediti tip prepreka i veličinu prepreka. Pod tip se misli od kojeg materijala je sačinjena prepreka, primjerice, metal i beton različito odbijaju ili apsorbiraju radio valove. Prikaz prostora i na koji način su postavljeni ESP32 prijamnici prikazan je na slici 4.5.



Slika 4.5 - Prikaz postavljanja prijamnika u prostoru

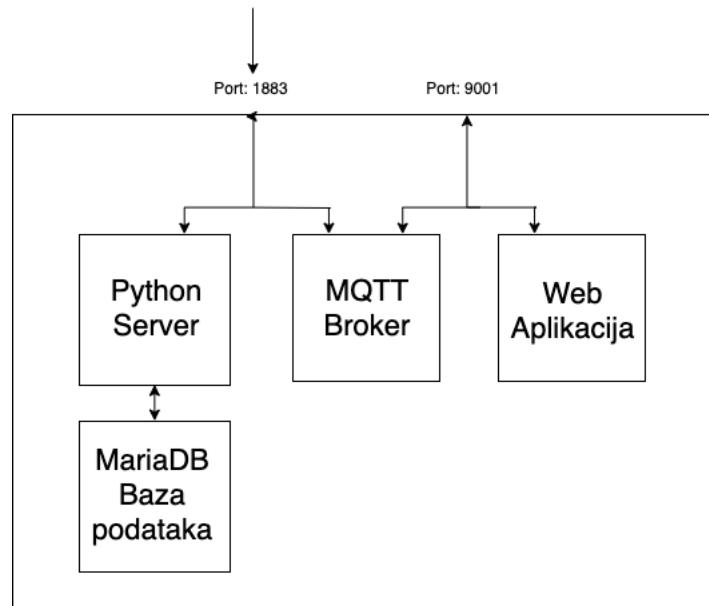
Prostor ima dimenzije duljine 420 cm i širine 300 cm. ESP32 prijamnici i njihove lokacije su prikazani plavim kružićima. Te pozicije su odabrane iz više razloga, a neki od njih su položaj izvora električne energije za prijamnike te što svojim pozicioniranjem podjednako obuhvaćaju prostor tj. radijus primanja signala prijamnika je ravnomjerno

raspoređen kroz prostor. Ti položaji prijammnika su bitni jer se njihova lokacija koristi u trilateraciji prilikom određivanja položaja radiofara u prostoru.

4.2. Softverske komponente

Za potrebe ovog rada su na Raspberry Pi-u instalirane ili izrađene te pokrenute četiri softverske cjeline: MQTT broker, Python server, MariaDB baza podataka i web aplikacija. MQTT protokol je izabran je iz više razloga, a jedan od glavnih je što ga podržavaju svi izabrani programski jezici (Python, C/C++ i JavaScript) te sklopovlje (ESP32, Raspberry Pi). Drugi razlog je taj što je relativno jednostavan protokol tj. ne sadrži puno nepotrebnih podataka kod slanja što omogućuje štednju baterije. Treći razlog je taj jer se temelji na principu objavi-pretplati što omogućuje neovisni rad cijelog sustava ako jedan od radiofarova iz nekog razloga prestane raditi. Programski jezik Python je izbran zato što ima dobru podršku za MQTT protokol i također ga se koristi kod spremanja podataka u MariaDB bazu. Baza MariaDB je izabrana zbog toga što je dobro podržana na Raspberry Pi-u. Programski jezik JavaScript je korišten kod prikazivanja podataka u web pregledniku.

Dijagram softverskih komponenti je prikazan na slici 4.6.



Slika 4.6 - Dijagram softverskih komponenti na Raspberry Pi

Podaci od prijavnika tj. od ESP32 dolaze preko porta 1883 gdje MQTT broker prima podatke na odabranu temu „home/room1/beacons“ i šalje tj. obavještava sve pretplatnike te teme da je došao novi podatak. U ovom slučaju se obavještava Python server koji je pretplaćen na temu „home/room1/beacons“. Python server prilikom dolaska podataka te podatke transformira i sprema na MariaDB bazu podataka gdje se ti podaci čuvaju. Podaci se spremaju u bazi kako bi se sačuvali sva mjerenja koji će pomoći kod analize. Isto tako postoji drugi otvoreni port 9001 gdje MQTT protokol obavještava sve pretplatnike teme preko *websockets*¹³ odnosno u ovom slučaju web aplikaciju koja u realnom vremenu prikazuje kretanje radiofarova u prostoru.

4.2.1. Programiranje ESP32 mikrokontrolera

Za razvoj softvera za ESP32 u ovom radu se koristi Arduino IDE razvojna platforma. U ovom radu su iskorištene ugrađene funkcionalnosti mikrokontrolera poput Bluetooth i Wi-Fi modula.

¹³ *Websockets* je tehnologija koja omogućuje dvosmjernu komunikaciju između korisnikovog preglednika i poslužitelja https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API

Svaki od tri ESP32 mikrokontrolera koristi kôd koji će slati izmjerene signale svakog radiofara u prostoriji. Na početku svakog kôda treba uključiti sve potrebne pomoćne biblioteke poput Wi-Fi, BLE itd. Nakon toga potrebno je inicijalizirati sve potrebne varijable koje se koriste u cijelom programu tj. one će biti globalnog doseg. Jedne od tih varijabli je ime ili univerzalni broj, a to ime ili broj je potrebno kako bi se prilikom slanja podataka mogao razlikovati od kojeg je mikrokontrolera odnosno ESP32 došao podatak.

Kao i svaki Arduino program, i ovaj se sastoji od dvije glavne funkcije koje se moraju definirati, a to su *setup* i *loop*. Funkcija *setup* se izvršava samo jednom tj. pri pokretanju ESP32 mikrokontrolera se koristi za definiranje i inicijalizaciju parametara. U ovom radu će se iskoristiti za inicijalizaciju serijske veze te inicijalizaciju BLE biblioteke koja se koristi kod mjerenja jačine signala. Funkcija *loop* se poziva neprestano, u beskonačnoj petlji tj. kôd koji se nalazi u toj funkciji će se izvršavati sve dok je ESP32 uključen, i prvi put se poziva nakon što se izvrši funkcija *setup*. Na početku *loop* funkcije će se mjeriti koji sve radiofarovi postoje i njihove jačine signala. Nakon što je BLE modul izmjerio sve dostupne radiofarove u prostoriji i njihove jačine signala, kreirat će JSON objekt koji će sadržavati sljedeće podatke:

- Ime ili univerzalni broj ESP32 mikrokontrolera koji je inicijaliziran na početku kôda
- Ime radiofara ili njegova MAC adresa i
- Jačina signala odnosno RSSI

Nakon što je JSON objekt napravljen, potrebno je poslati podatke, a da bi se to moglo ESP32 se mora spojiti na Wi-Fi mrežu. Nakon što se uspješno spojio na mrežu, ESP32 se spaja na MQTT broker. Poslije uspješnog spajanja na MQTT broker, pomoću metode *publish* šalje podatke na MQTT broker. Ta metoda prima kao parametre ime teme na koju se šalju podaci, podaci ili sadržaj poruke koji se šalje i u slučaju da je poruka sačinjena od niza bajtova potrebno je unijeti i duljinu niza poruke. Metoda *publish* nakon slanja će vratiti je li slanje bilo uspješno ili nije.

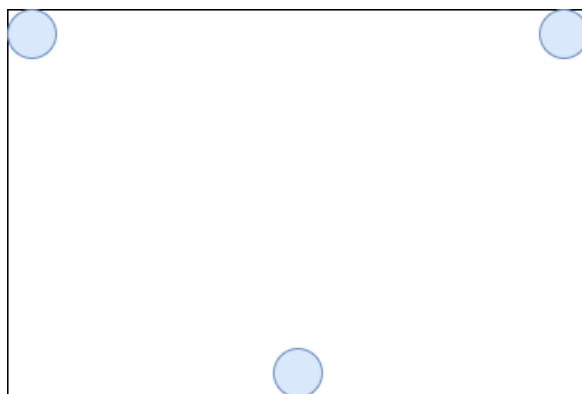
4.2.2. Računanje trilateracije na klijentskoj strani

Klijentska strana je dio tradicionalnog sustava klijent i poslužitelj, gdje klijentska strana zatraži podatke od poslužitelja, a poslužitelj ih vrati natrag klijentu, najčešće web pregledniku.

U ovom radu će se koristiti klijentska web aplikacija koja je napisana u skriptnom programskom jeziku JavaScript koji je zadužen za primanje i manipuliranje podacima koji će se prikazivati u web pregledniku.

U web pregledniku će biti prikazan tlocrt prostorije, odnosno prostorija će biti prezentirana u obliku pravokutnika. Taj pravokutnik se može zamisliti kao koordinatni sustav sa x i y osi. Duljina prostorije će biti os x , dok će širina biti os y . Fizičke dimenzije prostorije su 420 cm sa 300 cm. Mapiranje u piksele je napravljeno tako da jedan centimetar u prostoriji odgovara jednom pikselu u web pregledniku. To znači da će prostorija u web pregledniku biti predstavljena pravokutnikom dimenzija 420 piksela sa 300 piksela.

Ishodište koordinatnog sustava je postavljeno u gornji lijevi kut, iz čega slijedi da donja desna točka ima koordinate (420, 300). Po tom principu se može ubaciti tri ESP32 mikrokontrolera koji će imati svoje koordinate. Prikaz tlocrt prostora u obliku pravokutnika sa pozicijama ESP32 mikrokontrolera prikazana je na slici 4.7.



Slika 4.7 - Prikaz tlocrta prostorije sa ESP32 mikrokontrolerima

Gornji lijevi ESP32 mikrokontroler će imati koordinate (5, 5) jer neće baš moći biti stavljen u sami kut prostorije jer ESP32 ima svoj prostor koji zauzima. Gornji desni mikrokontroler će imati koordinate (415, 5), a donji srednji mikrokontroler će imati

koordinate (210, 295). Kao što je navedeno, te koordinate u prostoru će biti prikazane u centimetrima, dok prikaz tog prostora na web aplikaciji će biti u pikselima.

Podatke prikupljaju ESP32 pomoću BLE modula, transformiraju ih u JSON format i takve ih šalju pomoću MQTT protokola na temu „home/room1/beacons“. JavaScript kôd je pretplaćen na tu temu te prima sve podatke koje šalju ESP32 mikrokontroleri. JavaScript sadrži informacije koji svi radiofarovi postoje u prostoriji i njihove karakteristike poput imena, MAC adrese te njihove jačine signala prema svakom od ESP32 mikrokontrolera. Pošto postoje tri ESP32 mikrokontrolera, svaki radiofar imaće tri jačine signala odnosno RSSI koji se koristi za računanje trilateracije.

U JavaScriptu je korištena biblioteka *trilat* za izračunavanje pozicije radiofara. Ta biblioteka prima sljedeće podatke za svaki ESP32 mikrokontroler: lokaciju x osi i y osi pozicije ESP32, te udaljenost radiofara od ESP32.

Pošto je udaljenost radiofara izražena u RSSI, potrebno je izračunati kolika je jačina signala u metrima. Za to postoji formula (1) koja je objašnjena u potpoglavlju 3.5. Nakon što formula izračuna koliko je to u metrima, potrebno je metre pretvoriti u cm jer jedan cm u prostoru je jedan piksel u web aplikaciji.

Nakon toga biblioteka *trilat* će na temelju tih podataka izračunati i vratiti poziciju sa x i y osima tog radiofara. Nakon što je izračunata pozicija, JavaScript će ažurirati ili prilagoditi poziciju radiofara u web aplikaciji.

5. Testiranje i analiza

U ovom poglavlju će biti opisani testovi koji su napravljeni. Nakon mjerenja i prikupljanja podataka će biti napravljena analiza preciznosti sustava te će se ili zaključiti da je hipoteza potvrđena ili će se analizirati uzroci neuspjeha te dati smjernice za poboljšanje preciznosti sustava.

5.1. Opis provedenih testova

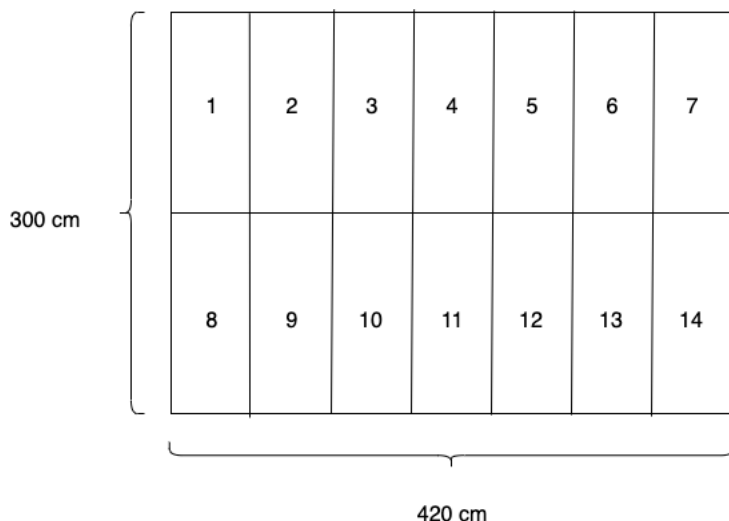
Prije samog testiranja potrebno je sustav dovesti u operativni rad, što znači da svaki od sustava mora pravilno raditi kako bi se testovi bili valjani. Potrebno je redom pokrenuti sljedeće komponente:

- Na Raspberry Pi je potrebno pokrenuti MQTT broker, web aplikaciju i server aplikaciju
- Postaviti ESP32 mikrokontrolere na njihove pozicije u prostoru. Bit će postavljeni na visinu od 1 m i priključeni na izvor napajanja

Razlog zašto su ESP32 mikrokontroleri postavljeni na visinu od 1 metra je taj što većina uređaja ili opreme u zatvorenim poslovnim prostorima odlaže ili njihove pozicije su negdje u visini 1 metra i zbog toga na toj visini točnost očitavanje podataka trebala biti točnija.

Nakon što se sve postavi u radnom okruženju, potrebno je prije početka testiranja provjeriti da se podaci koji se šalju ispravno spremaju u bazu te da web aplikacija reagira na promjenu pozicije radiofara.

Nakon što je sustav postavljen i validiran da dobro radi, može se krenuti s pripremljenim testovima. Prilikom svakog testa prostorija u kojoj se vrši testiranje podijeljena je na sektore. Razlog zašto je podijeljeno na sektore je taj kako bi se lakše moglo usporediti stvarne i dobivene vrijednosti. Ovaj princip sa sektorima će se koristiti u svim testovima kako bi se takvim pristupom jednako pokriti prostor i moći lakše analizirati u kojim sektorima ima najmanje odnosno najviše pogrešaka. Prikaz sektora u prostoriji je prikazana na slici 5.1.



Slika 5.1 - Prikaz prostorije podijeljen na sektore

Kao što je prikazano na slici 5.1, dimenzije prostorije su 4.2 metra sa 3 m. Prostor je podijeljen na 14 jednakih sektora, odnosno svaki sektor ima dimenzije 60 cm sa 150 cm.

Testovi koji se koriste u ovo radu su sljedeći:

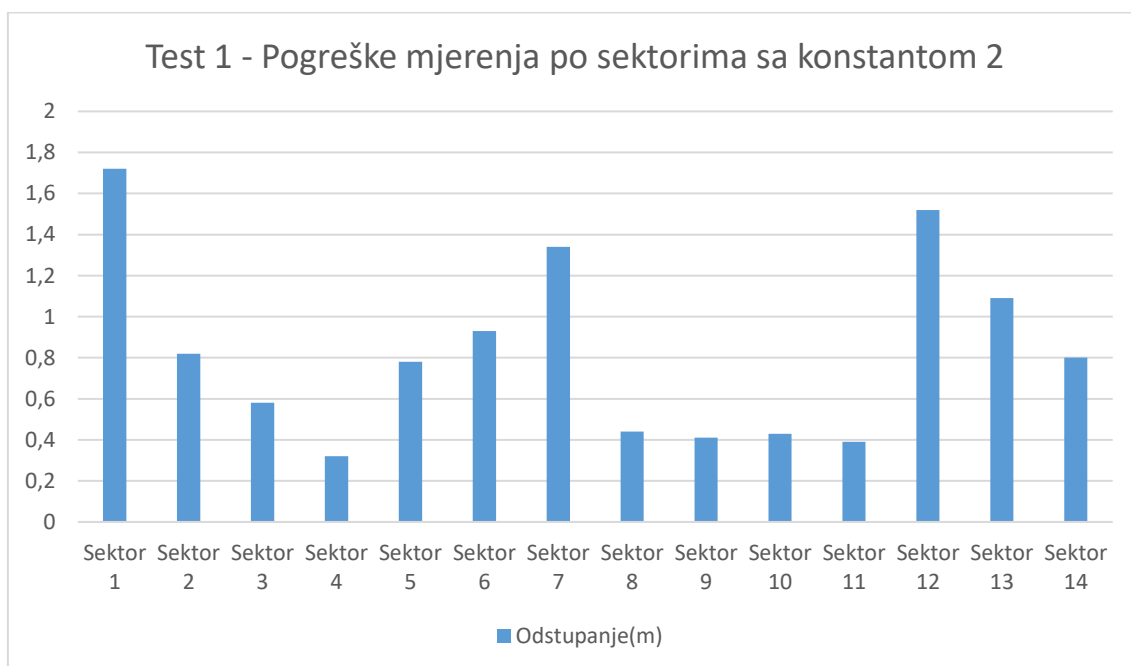
- u prvom testu će se testirati i pokazati kolika su odstupanja izmjerenih lokacija radiofarova od stvarnih lokacija, odnosno, kolika je razlika tj. udaljenost stvarne lokacije u prostoru od one iz web sučelja koje web preglednik pokazuje.
- u drugom testu će se testirati i prikazati koliko promjena konstante za okoliš utječe na izmjerene lokacije tj. hoće li kvaliteta podataka biti bolja odnosno hoće li se razlika udaljenosti stvarne i mjerene lokacije smanjiti ili povećati. Konstante koje su se uzete za mjerenje su 2, 2.3 i 2.7 koje su namijenjene za prazan prostor. Baš te konstante su uzete zbog toga što konstanta sa brojem 2 označava minimalnu konstantu u zatvorenom prostoru, dok 2.7 već predstavlja prostor sa jako puno prepreka.

Prilikom svih testiranja radiofarovi i ESP32 su se postavljali u svaki sektor na visini od 1 m jer su radni stolovi obično blizu te visine.

5.2. Analiza pogreške i validiranje hipoteze

U prethodnom poglavlju su objašnjeni testovi koji će se koristiti za analizu pogreške i analizu preciznosti sustava. Na kraju ovog poglavlja će se validirati hipoteza odnosno je li hipoteza potvrđena ili će se analizirati uzroci neuspjeha te dati smjernice za poboljšanje preciznosti sustava.

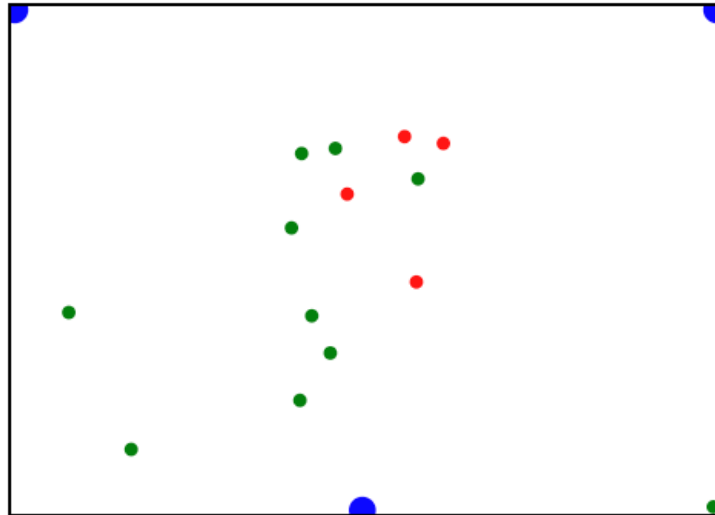
U prvom testu će se prikazati odstupanja lokacija radiofarova u prostoriji od mjerenih lokacija koji su prikazane na sučelju u web pregledniku. U ovom testu za svaki sektor imamo po jedno mjerenje odnosno 14 mjerenja sveukupno. Prilikom računanja trilateracije pomoću formule koja je objašnjena u prijašnjim poglavljima koristi se konstanta za okoliš u vrijednosti 2.



Slika 5.2 - Grafički prikaz odstupanja u Testu 1

Kao što se vidi na slici 5.2, pogreške postoje u svim sektorima. U hipotezi je pretpostavljeno da će pogreška biti manja od 1 metra, ali sa slike se može se vidjeti da imamo 4 sektora u kojima su očitavanja bila iznad 1 metra, dok su ostalih 10 unutar zadovoljavajuće pogreške. Važno je napomenuti ono što je vidljivo na grafu, a to je da su sektori na početnom dijelu ili na kraju prostorije imali najveća odstupanja (sektori 1, 7, 12, 13). Ostali sektori, odnosno oni sektori koji su bili u sredini prostorije, imali su manje

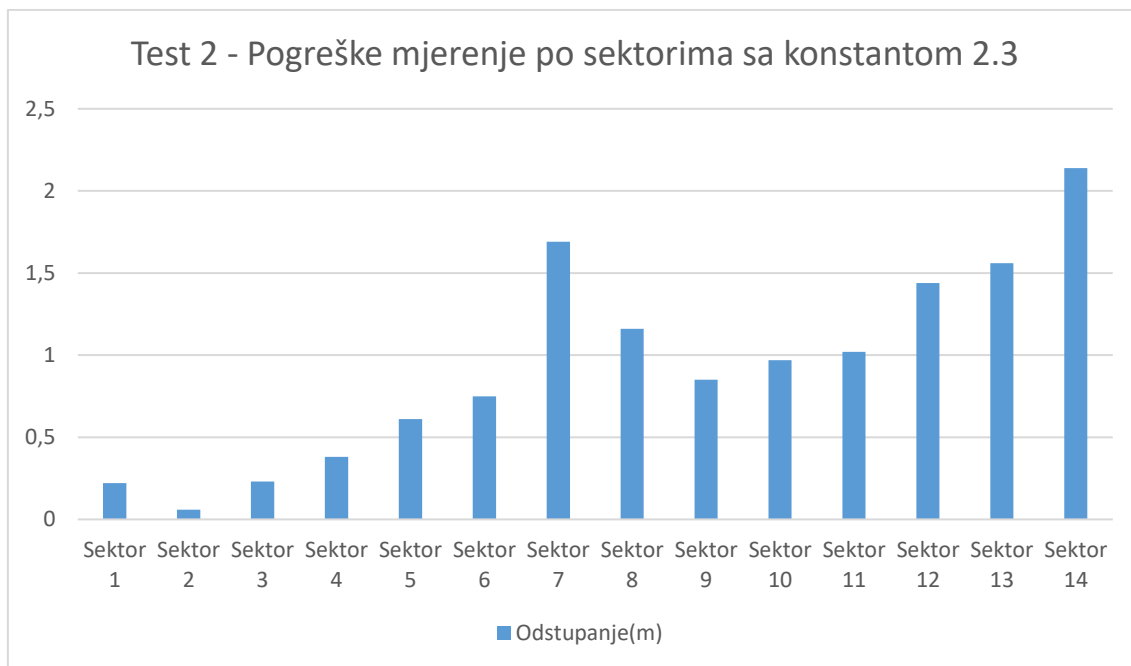
pogreške. Iz toga se može zaključiti da oni sektori koji imaju najveće odstupanje od barem jednog ESP32 mikrokontrolera (npr. Sektoru 1 je najudaljeniji ESP32 u gornjem desnom kutu) imaju najveće pogreške.



Slika 5.3 - Prikaz mjerenja pozicije radiofarova u Testu 1 sa konstantom za okoliš 2

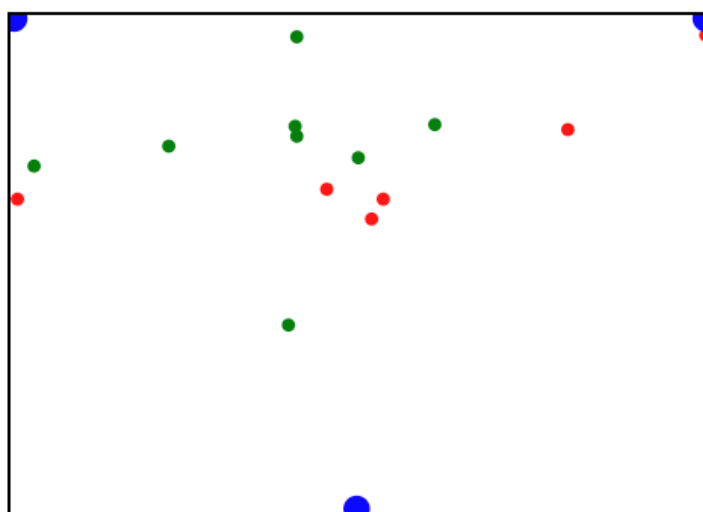
Na slici 5.3 prikazana su izmjerene pozicije (pozicije sa koje su prikazane u web pregledniku) i jesu li njihova mjerenja unutar jednog metra od pozicije u prostoriji. Ako su unutar jednog metra od prikazan je na slici pomoću zelenog kružića, dok oni koji odstupaju od jednog metra su označeni crvenim kružićima. Ono što se vidi u kombinaciji sa grafom i slikom 5.2 jest da ona očitavanja koji su unutar jednog metra dolaze iz sektora koji su u središtu prostorije, dok su oni koji odstupaju dolaze iz onih sektora koji su na početku ili na kraju prostorije.

U drugom testu će se s tri mjerenja i s različitim konstantama za okoliš pokušati odrediti najbolju konstantu za tu prostoriju. Za ovaj test koristile su se konstante od onih koje odgovaraju za prazan prostor do onih za urbane prostore koji imaju dosta prepreka. Konstante koje se koriste su 2, 2.3 i 2.7. Kod konstante 2 će se iskoristiti mjerenja iz prvog testa dok će se za preostale dvije konstante napraviti novi testovi.



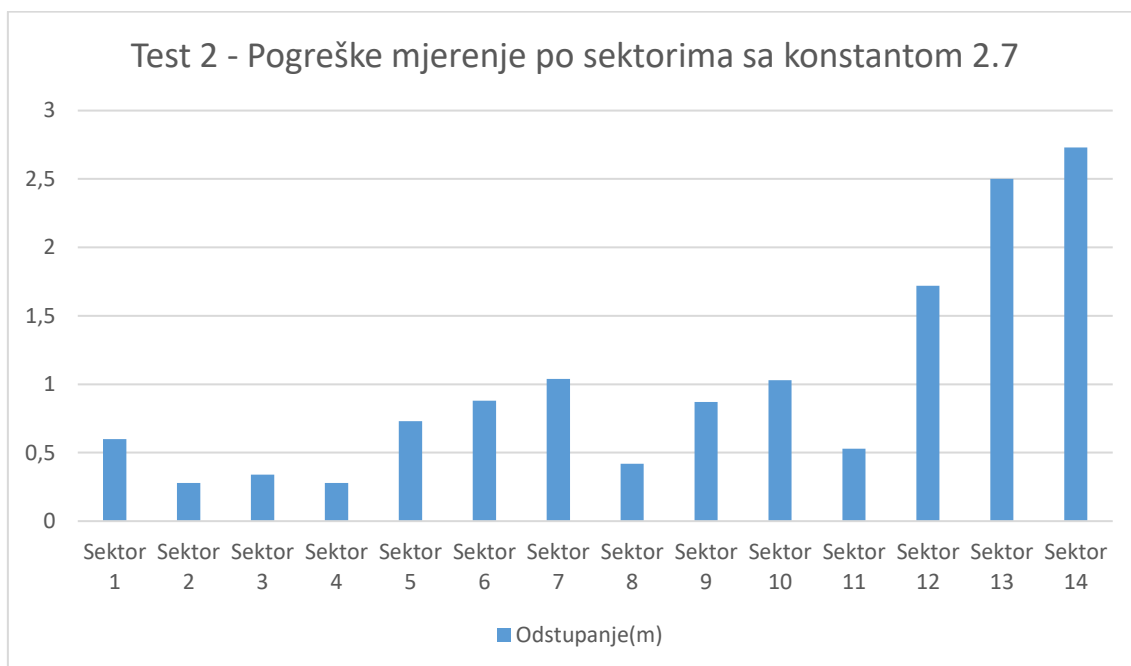
Slika 5.4 - Grafički prikaz odstupanja u Testu 2 sa konstantom 2.3

Kako je prikazano u sljedećem grafu (slika 5.4) mjerenja počinju sa malom pogreškom, ali što se ide dalje po sektorima pogreška je sve veća. Ono što je još vidljivo iz ovog grafa što su u drugom redu sektora od 8 do 14 puno veće pogreške nego u onom prvom redu od 1 do 7. Usporedno sa grafom iz mjerenja gdje je konstanta 2 je bolji u prvom redu sektora dok u sektorima drugog reda se vidi povećanje grešaka.



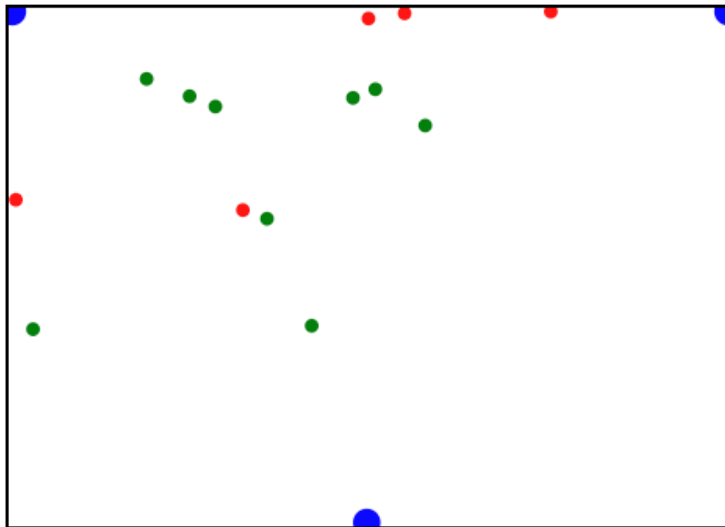
Slika 5.5 - Prikaz mjerenja pozicije radiofarova u Testu 2 sa konstantom za okoliš 2.3

Kao što je prikazano na slici 5.5, u prvom redu sektora su očitavanja unutar jednog metra, dok su ona koja prelaze jedan metar većinom iz donjeg reda odnosno sektora 11, 12, 13, 14. Isto kao i prvom mjerenju, oni sektori koji su na istim udaljenostima prema svim ESP32 mikrokontrolerima odnosno u sektorima koji se nalaze u sredini prostorije imaju manju pogrešku nego oni sektori na početku ili na kraju prostorije. Izuzetak u sektoru 1 u prvom testiranju je pogreška bila 1.72 metra dok u ovom testiranju je bila 0.22 što je velika razlika.



Slika 5.6 - Grafički prikaz odstupanja u Testu 2 sa konstantom 2.7

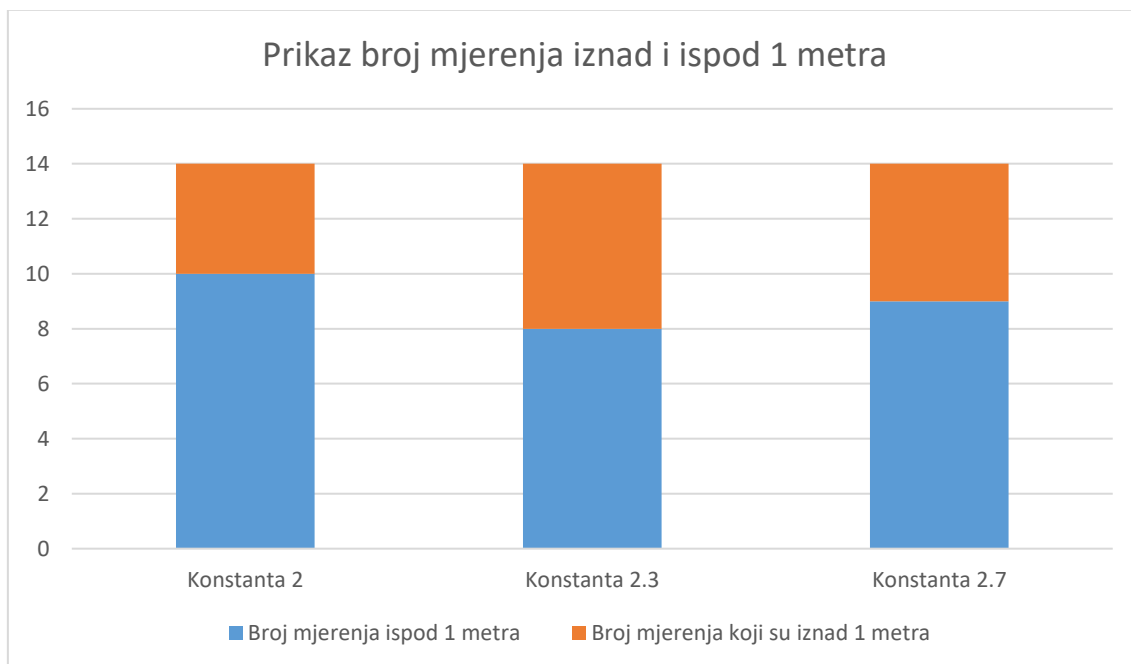
Kako je prikazano u grafu (slika 5.6) mjerenja počinju s manjim greškama tj. pogreške su ispod 1 metra. Tek u sektoru 7 koji je na kraju prostorije ima blago odstupanje od 1 metra. U drugom redu, odnosno u sektorima 8 do 14 se vidi veća razlika u odnosu na prvi red sektora. Sektori 12, 13, 14 kao u svim testovima imaju veću grešku od 1 metra, međutim u ovom testu imaju najveću pogrešku. Ono što je također zanimljivo da su samo dva odstupanja od 1 metra, koji čak nisu ni tako veliki sve do sektora 12, 13, 14 koji su stvarno velika odstupanja. Što se tiče cijelog testa, bio bi najbolji da nema onoliko velike pogreške u sektorima 12, 13, 14.



Slika 5.7 - Prikaz mjerenja pozicije radiofarova u Testu 2 sa konstantom za okoliš 2.7

Na slici 5.7 se također može vidjeti da većina točnih mjerenja dolazi iz sektora koji su bliže sredini prostorije odnosno mjesta gdje su podjednako udaljena od svih ESP32 mikrokontrolera.

Sveukupni broj mjerenja u svim testovima je 42, odnosno za svaki sektor po jedno mjerenje. Od toga su 27 mjerenja koji su ispod 1 metra dok preostalih 15 mjerenja su iznad 1 metra, odnosno to je 35.7% pogreške. Na sljedećem grafu (slika 5.8) su prikazane sva mjerenja i njihov broj mjerenja koji su ispod i iznad 1 metra.



Slika 5.8 - Grafički prikaz mjerenja iznad i ispod 1 metra

Ukupna pogreška po testu mjerenja gdje je konstanta 2 je 11.57 metara, mjerenja gdje je konstanta 2.3 je 13.08, dok je za konstantu mjerenja 2.7 ukupna greška je 13.95 metara. Iz toga se može zaključiti da je konstanta koja najbolje odgovara ovoj prostoriji 2, jer broj grešaka i ukupna pogreška je najmanja od svih drugih mjerenja. Međutim, treba uzeti i u obzir konstantu 2.7 koja je imala jako dobra mjerena na početku samo s dvije mala odstupanja od jednog metra sve do zadnja tri sektora koja su najveća odstupanja od svih tih mjerenja. Da su ta zadnja tri mjerenja ispod pogreške 1.5, metra ova konstanta bi bila najbolja.

Analiza je pokazala da većina mjerenja je unutar jednog metra, ali sa pogreškom od 35,7%, odnosno svako treće mjerenje je veće od jednog metra. Hipoteza je potvrđena, ali uz ovakvu veliki broj pogreške koja bitno utječe na kvalitetu ovakvog sustava. Međutim ovakva tehnologija s radiofarovima ima jako puno faktora koji utječu na njihovo mjerenje: od veličine prostorije, njihove prepreke, algoritma prikupljanja podataka, kvalitete radiofarova, algoritma koji računa poziciju pomoću trilateracije itd. Sve navedeno bitno utječe na kvalitetu mjerenja i samih podataka.

Poboljšanja koji se mogu napraviti su doraditi algoritam koji prikuplja podatke koji će preciznije i više podataka slati, kako bi kvaliteta podataka bila veća i pri tome sama pogreška manja. Ako je interval slanja podataka manji odnosno podaci će se slati više puta u jedinici vremena, tada će izračunavanje pozicije biti češće i samim time preciznije. Taj algoritam je smješten u svim ESP32 mikrokontrolerima i potrebno je poraditi na dijelu koji je zadužen za interval slanja podataka. Također se može poraditi na različitim konstantama ovisno u prostoriji o kojoj se radi, ovisno o njenoj veličini, broju prepreka i vrsti prepreka s kojim se susreće sustav.

Zaključak

U ovom radu se pokazalo kako se pomoću tehnologije radiofarova može pratiti lokacija radiofarova, prije svega u zatvorenim prostorima iako nema prepreka da se ovakav sustav koristi i na otvorenim prostorima. Hipoteza koja je se željela dokazati da se pomoću ovakvog pristupa može uspješno locirati radiofar do preciznosti od 1 metra.

Kao praktični dio rada se izradio dizajn i izrada softverskog i hardverskog sustava. Nakon završetka izrade, za validiranje hipoteze napravljena su dva testa. U prvom testu su se izmjerila pozicije radiofarova sa konstantom 2 dok u drugom testu su izmjerena još dva slučaja s konstantama 2.3 i 2.7. Nakon prikupljanje podataka iz testova i njihovih analiza odredila se ona konstanta koja najbolje odgovara toj prostoriji. Način na koji se to odredilo je taj da mora imati što manji broj pogrešaka, odnosno broj sektora koji su pogreške ispod jednog metra te sve ukupni broj pogrešaka mora biti najmanji.

Hipoteza koja se postavila je potvrđena tj. većina mjerenja je ispod 1 metra, ali sa pogreškom od 35.7%, što znači da je svako treće mjerenje bilo iznad jednog metra. Takva pogreška uvelike utječe na kvalitetu takvog sustava i pitanje je može li se ovakav sustav pustiti u rad sa ovakvom pogreškom.

Ovakav sustav ima jako puno faktora koji utječu na samu kvalitetu podataka poput veličine prostorije, broja prepreka, vrste prepreka, kvalitete radiofarova i algoritma trilateracije koji računa lokaciju. Kao što se može vidjeti, zbog tih faktora ne može se tvrditi da će ovakav sustav biti točan za sve prostorije, već to uvelike ovisi gdje se nalazi takav sustav, odnosno nema jedinstvenog rješenja za sve prostorije, već je potrebno prilagoditi sve faktore toj prostoriji u kojoj se sustav nalazi.

Popis kratica

A/D	Analog/digital	analogni/digitalni
AES	Advanced encryption standard	napredni enkripcijski standard
BLE	Bluetooth low energy	bluetooth niske potrošnje
CPU	Central processing unit	centralna procesorska jedinica
D/A	Digital/analog	digitalni/analogni
GNSS sistemi	Global navigation satellite systems	globalni navigacijski satelitski
GPIO	General purpose input/output	opće namjenski ulazi/izlazi
GPS	<i>Global positioning system</i>	globalni pozicijski sistem
I2C	Inter-Integrated Circuit	
IoT	Internet of things	internet stvari
MQTT	MQ Telemetry Transport	MQ telemetrijski transport
OS	Operating system	operacijski sustav
PWM	Pulse-width modulation	modulacija širine impulsa
RAM	Random access memory	memorija s nasumičnim pristupom
RSSI	Received signal strength indication	indikator jačine primljenog signala
SPI	Serial Peripheral Interface Bus	serijsko periferno sučelje
SRAM pristupom	Static random access memory	tatički memorija s nasumičnim pristupom
SSL	Secure Socket Layer	sloj sigurnosnog priključka
UART	Universal asynchronous receiver/transmitter. prijamnik/predajnik	univerzalni asinkroni
UUID	Universally unique identifier	univerzalni jedinstveni identifikator

USB	Universal serial bus	univerzalna serijska sabirnica
Wi-Fi	Wireless Fidelity, wireless Internet	bežična vjernost, bežični internet

Popis slika

Slika 3.1 - Raspberry Pi.....	6
Slika 3.2 - Raspberry Pi GPIO pinovi	8
Slika 3.3 - ESP32.....	10
Slika 3.4 - Slikovni prikaz uspostavljanja konekcije između MQTT klijenta i brokera	11
Slika 3.5 - Primjer MQTT teme.....	14
Slika 3.6 - Shematski prikaz rada MQTT protokola	15
Slika 3.7 - Primjer spajanja uređaja na broker.....	16
Slika 3.8 - Primjer jednog radiofara	18
Slika 3.9 - Udaljenost jednog prijammnika od objekta	20
Slika 3.10 - Dva prijammnika i moguće lokacije	21
Slika 3.11 - Presjek svih tri prijammnika u točki <i>P</i>	22
Slika 4.1 - Vizualni prikaz komunikacije sklopovlja	24
Slika 4.2 - Primjer MQTT brokera	26
Slika 4.3 - Slanje poruke na temu testTema	26
Slika 4.4 - Primjer primanja poruke na pretplaćenu temu testTema	26
Slika 4.5 - Prikaz postavljanja prijammnika u prostoru	27
Slika 4.6 - Dijagram softverskih komponenti na Raspberry Pi	29
Slika 4.7 - Prikaz tlocrta prostoriije sa ESP32 mikrokontrolerima	31
Slika 5.1 - Prikaz prostoriije podijeljen na sektore.....	34
Slika 5.2 - Grafički prikaz odstupanja u Testu 1	35
Slika 5.3 - Prikaz mjerenja pozicije radiofarova u Testu 1 sa konstantom za okoliš 2..	36
Slika 5.4 - Grafički prikaz odstupanja u Testu 2 sa konstantom 2.3	37
Slika 5.5 - Prikaz mjerenja pozicije radiofarova u Testu 2sa konstantom za okoliš 2.3	37
Slika 5.6 - Grafički prikaz odstupanja u Testu 2 sa konstantom 2.7	38
Slika 5.7 - Prikaz mjerenja pozicije radiofarova u Testu 2 sa konstantom za okoliš 2.7	39
Slika 5.8 - Grafički prikaz mjerenja iznad i ispod 1 metra.....	39

Popis tablica

Tablica 3.1 - Prikaz dobro sastavljene CONNECT poruke.....	12
Tablica 3.2 - Statusni kodovi uspješnosti konekcije.....	13

Popis kôdova

Kôd 1 - Program za paljenje i gašenje LED lampice u programsko jeziku Python 9

Reference

1. <https://www.gps.gov/systems/gnss/>
2. https://www.tutorialspoint.com/microprocessor/microcontrollers_overview.htm
3. <https://www.intellectsoft.net/blog/what-are-beacons-and-how-do-they-work/>
4. <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/>
5. <https://www.raspberrypistarterkits.com/resources/best-programming-language-raspberry-pi/>
6. <https://www.raspberrypi.org/documentation/usage/gpio/>
7. https://www.megunolink.com/articles/wireless/what-is-the-esp32/?utm_referrer=https%3A%2F%2Fduckduckgo.com%2F
8. <http://esp32.net/>
9. <https://mqtt.org/>
10. <https://www.hivemq.com/blog/mqtt-essentials-part-3-client-broker-connection-establishment/>
11. <https://randomnerdtutorials.com/what-is-mqtt-and-how-it-works/>
12. <https://www.novelbits.io/basics/bluetooth-low-energy/>
13. <https://kontakt.io/beacon-basics/what-is-a-beacon/>
14. <https://blog.beaconstac.com/2018/08/things-you-need-to-know-before-you-buy-beacons/>
15. <https://iotandelectronics.wordpress.com/2016/10/07/how-to-calculate-distance-from-the-rssi-value-of-the-ble-beacon/>
16. <https://community.estimote.com/hc/en-us/articles/201636913-What-are-Broadcasting-Power-RSSI-and-other-characteristics-of-a-beacon-s-signal->
17. <https://appelsiini.net/2017/trilateration-with-n-points/>
18. <https://www.alanzucconi.com/2017/03/13/positioning-and-trilateration/>
19. <https://r00t4bl3.com/post/how-to-install-mysql-mariadb-server-on-raspberry-pi>
20. <https://randomnerdtutorials.com/testing-mosquitto-broker-and-client-on-raspbberry-pi/>

Prilog