

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

TEHNIČKA DOKUMENTACIJA

# **Određivanje topologije mreže i rasporeda čvorova**

*Filip Škoro*

Voditelj: *prof. dr. sc. Stjepan Bogdan*

Zagreb, siječanj 2023.

# SADRŽAJ

<b>1. Uvod</b>	<b>1</b>
<b>2. Oprema</b>	<b>2</b>
<b>3. Zadatak i cilj projekta</b>	<b>3</b>
3.1. Bluetooth i WiFi mreža . . . . .	3
3.2. Računanje udaljenosti . . . . .	4
3.3. Računanje pozicija . . . . .	5
3.4. Pseudokod i slikovni primjer problema . . . . .	5
<b>4. Programska izvedba</b>	<b>7</b>
4.1. ROS . . . . .	7
4.2. Pokretanje programa . . . . .	8
4.3. Proračunavanje udaljenosti i pozicija . . . . .	8
4.4. Rezultati računanja udaljenosti . . . . .	9
<b>5. Zaključak</b>	<b>11</b>
<b>6. Literatura</b>	<b>12</b>

# 1. Uvod

U današnjem digitalnom svijetu u mnogim poslovima čovjeka zamjenjuje tehnologija. Sve više poslova moguće je obaviti korištenjem određene tehnologije dok neki poslovi čak i zahtijevaju prisutnost specifičnih uređaja kako bi mogli biti adekvatno obavljeni. Napredak tehnologije omogućio je mjerenje određenih parametara koji prije nisu mogli biti eksplicitno određeni.

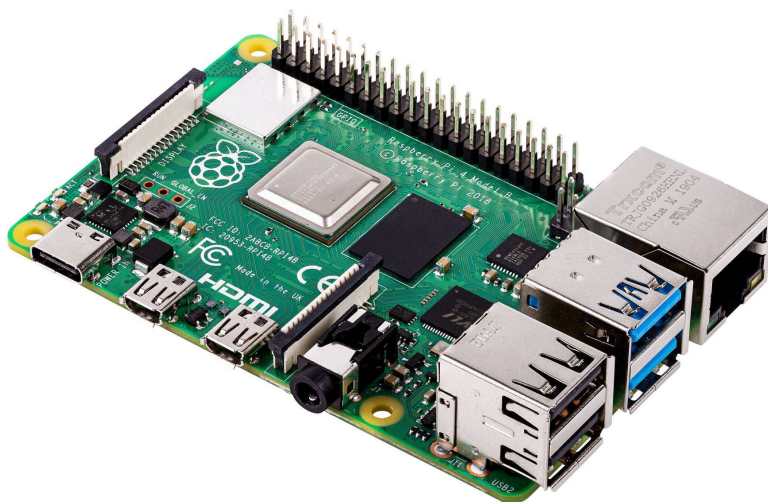
Jedan primjer uređaja koji se danas redovno koristi za mjerenje različitih parametara jest Raspberry Pi. Takvi uređaju su relativno maleni i pogodni za višestruka mjerenja i razmjenu podataka te se stoga često koristi više Raspberryja koji nerijetko moraju razmjenjivati mjerene podatke tijekom samog procesa mjerenja. Da bi se podatci mogli razmjenjivati potrebno je ostvariti dobru komunikaciju među svim uređajima. Također svi uređaji bi trebali imati informacije o ostalim uređajima poput svoje udaljenosti do svakoga od ostalih uređaja te pozicije ostalih uređaja.

Ova tehnička dokumentacija opisuje projekt koji je za cilj imao razviti bluetooth komunikaciju među Raspberryjima te omogućiti svakom od Raspberryja da izračuna svoju udaljenost do ostalih Raspberryja i odredi pozicije ostalih uređaja u kartezijskom koordinatnom sustavu. U poglavlju Oprema ukratko je opisan uređaj Raspberry Pi korišten u projektu. Zatim je u poglavlju Zadatak i cilj projekta objašnjen osnovni cilj projekta te je u poglavlju Programska izvedba opisan način na koji je program izveden i kako se došlo do rješenja.

Neki od sličnih znanstvenih radova i članaka koji se bave bliskom tematikom i opisuju slične probleme su Chen et al. (2022), Cui et al. (2015), Cao et al. (2021) te Liao et al. (2015).

## 2. Oprema

Za potrebe ovog projekta korišten je već spomenuti uređaj Raspberry Pi 4 Model B prikazan na slici 2.1. U projektu su korištena četiri takva uređaja koja su međusobno bila "vidljiva" ostalim uređajima pomoću bluetootha. Svaki od uređaja ima svoju MAC adresu za bluetooth i za WiFi mrežu koje služe za međusobno raspoznavanje. Uređaji su napajani pomoću klasičnog USB - C kabela. Više informacija o hardwareu i softwareu uređaja može se pronaći na web linku Raspberry Pi 4 Documentation.



**Slika 2.1:** Raspberry Pi 4 Model B

## 3. Zadatak i cilj projekta

Ovaj projekt sastojao se od više manjih zadataka koje je bilo potrebno povezati u jednu cjelinu. Kao što je spomenuto u uvodu, Raspberryji se mogu koristiti za različite svrhe te nerijetko moraju razmjenjivati mjerene podatke tijekom samog procesa mjerenja. Da bi mogli razmjenjivati podatke potrebno je razviti dobru i kvalitetnu komunikaciju među svim uređajima isto kao što je bitno da uređaji poznaju svoju udaljenost od ostalih Raspberryja, ali i međusobne udaljenosti i pozicije drugih Raspberryja. Ovaj projekt nastojao je razriješiti spomenute zadatke na sustavu koji se sastoji od četiri Raspberry Pi 4 uređaja.

### 3.1. Bluetooth i WiFi mreža

Da bi uređaji mogli međusobno komunicirati prvo moraju biti u mogućnosti "vidjeti" jedni druge. U ovom projektu to je ostvareno pomoću bluetootha s obzirom na to da korišteni uređaji imaju mogućnost skeniranja okolnih uređaja korištenjem bluetooth signala. Također parametar RSSI koji je dio BLE (*Bluetooth low energy*) signala bit će od ključne važnosti za određivanje međusobnih udaljenosti Raspberryja što je još jedan od razloga zašto je potrebno koristiti bluetooth signal. S obzirom na to da su uređaji upravljani s osobnog računala, svaki od Raspberryja mora biti spojen na istu WiFi mrežu kao i korišteno računalo kako bi ih računalo moglo vidjeti. To se ostvaruje tako da se svakom od Raspberryja *hardkodira* naziv mreže i lozinka mreže na koju bi se trebali spojiti te čim takva mreža postane dostupna Raspberryji se automatski spajaju na nju.

Nakon što je sigurno da su uređaji spojeni na zadanu mrežu i da ih je moguće upravljati s osobnog računala, moguće je krenuti sa proračunavanjem udaljenosti.

## 3.2. Računanje udaljenosti

Nakon što svi uređaji vide jedni druge pomoću bluetooth signala odnosno nakon što su skenirali jedni druge, potrebno je odrediti udaljenost svakog uređaja do svakog uređaja. Svaki od Raspberryja sam za sebe određuje udaljenosti do ostalih Raspberryja. Kao što je ranije spomenuto, za izračun udaljenosti koristi se jedan parametar BLE signala; RSSI vrijednost iz čega proizlazi važnost bluetooth komunikacije. Formula prema kojoj se računa udaljenost među Raspberryjima jest:

$$Distance = 10^{\frac{MeasuredPower - RSSI}{10 * N}}$$

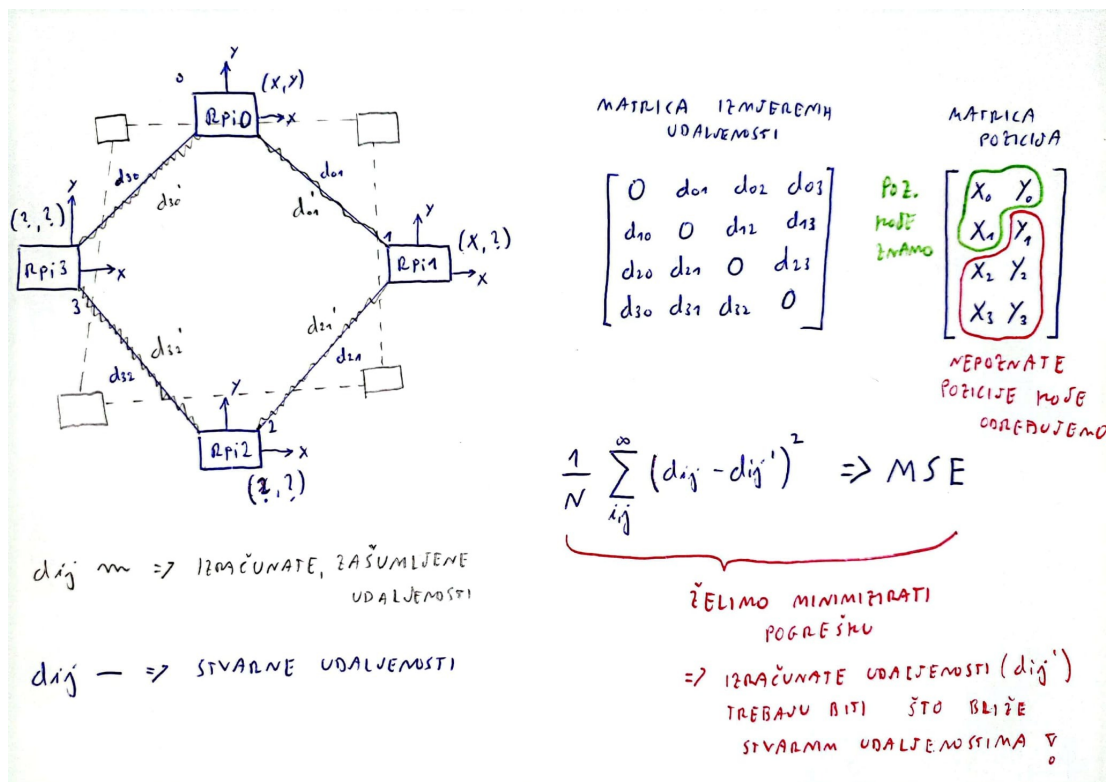
gdje *Measured Power* predstavlja izmjerenu RSSI vrijednost, *RSSI* izmjerenu RSSI vrijednost kada su Raspberryji udaljeni jedan metar - mjerenjem se dobilo da je vrijednost otprilike jednaka -70 dB te *N* predstavlja konstantu - za potrebe ovog projekta uzeta je vrijednost 2. Nakon što je svaki Raspberry izračunao sve udaljenosti potrebno ih je poslati drugim Raspberryjima kako bi svi znali svačije udaljenosti i mogli lakše izračunati koordinate drugih uređaja.

### 3.3. Računanje pozicija

Nakon što su poznate udaljenosti svih Raspberryja potrebno je još odrediti pozicije ostalih uređaja. Za početak potrebno je odabrati jedan Raspberry koji će biti referentni uređaj odnosno čije će koordinate biti (0,0) te drugi Raspberry čija će X-koordinata biti jednaka nuli. Potom je moguće odrediti ostale koordinate pazeći pritom na to da pogreška bude minimalna odnosno da pozicije budu što bliže stvarnim vrijednostima. Zbog nesavršenosti uređaja i drugih vanjskih faktora, nikada neće biti moguće dobiti savršeno točne pozicije, no moguće je dobiti dovoljno dobre koordinate koristeći različite ugrađene funkcije za optimizaciju. Ideja tih funkcija je sljedeća; s obzirom na to da su koordinate jednog uređaja fiksne isto kao i X-koordinata drugog uređaja, određuju se preostale koordinate tako da optimizacijski postupak konstantno proračunava nove pozicije na temelju izmjerenih udaljenosti koje dobiva od Raspberryja tako da se tim funkcijama predaje razlika trenutnih udaljenosti i udaljenosti iz prethodnog koraka s time da se u prvom koraku računanja određuju pozicije na temelju prvih izmjerenih udaljenosti. Očekivano je da će se tako dobivene pozicije s vremenom približavati točnim tj. stvarnim vrijednostima. Stoga je bitno izabrati određenu razinu pogreške koja je prihvatljiva te kada pogreška određenih pozicija u odnosu na stvarne bude ispod te razine onda se proces određivanja koordinata završava.

### 3.4. Pseudokod i slikovni primjer problema

Ovo poglavlje služi kao sažetak svega do sad opisanog. Kao što je rečeno, na temelju proračunatih udaljenosti određuju se pozicije; računaju se prosječne udaljenosti na temelju razlike stvarnih i izračunatih udaljenosti iz čega se potom određuje pogreška koju treba minimizirati. Da bi se pogreška minimizirala potrebno je optimirati postupak računanja distanci odnosno određivanja pozicija tj. formulu spomenutu u poglavlju 3.2. Postoje određeni algoritmi kojima se optimizacija može provesti, kao na primjer metoda najmanjih kvadrata, skalarna optimizacija itd. Neki od tih algoritama postoje kao ugrađene funkcije u programskom jeziku Python što ih čini lako dostupnima za korištenje prilikom rješavanja navedenog problema programskim putem. Na slici 3.1 nalazi se slikovni prikaz problema te kasnije pseudokod cijelog postupka.



Slika 3.1: Crtani prikaz problema

#### Algorithm 1 Određivanje topologije mreže i rasporeda čvorova

1. Skenirati bluetoothom ostale uređaje
2. Parsiranjem rezultata dohvatiti RSSI vrijednost
3. Pomoću formule odrediti udaljenosti do ostalih Raspberryja
4. Poslati svoje udaljenosti ostalim Raspberryjima
5. Pričekati da ostali Raspberryji pošalju svoje udaljenosti
6. Popuniti matricu sa odgovarajućim vrijednostima
7. Provesti optimizacijski postupak kako bi se dobile pozicije
8. Ponoviti postupak od 1. koraka



## 4. Programska izvedba

Za potrebe ovog projekta korištena su već spomenuta četiri Raspberry Pi uređaja. Svaki Raspberry može se doživljavati kao zasebno računalo. Na svakom od Raspberryja instaliran je Ubuntu 20.04. Unutar mape *catkin\_ws/src* na svakom od uređaja kreiran je projekt naziva *rpi\_project* koji sadržava sve potrebne skripte i datoteke za uspješno obavljanje zadatka. Svaki od Raspberryja unutar mape *catkin\_ws/src/rpi\_project/scripts* sadrži Python skriptu imena *skenerN.py* gdje se *N* odnosi na redni broj Raspberryja tako da postoje skripte imena *skener0.py*, *skener1.py*, *skener2.py* i *skener3.py* ovisno o tome o kojem se uređaju radi. Navedene skripte sadrže programski kod napisan u programskom jeziku Python3. Sve skripte su relativno slične, gotovo pa iste; svaka skripta je prilagođena svom Raspberry Pi uređaju kako bi mogao uspješno izvršavati zadatak no razlike su minimalne. Skripte sadrže sve potrebne metode za obavljanje zadatka koje se slijedno pozivaju u glavnom programu i taj se proces ponavlja. Za potrebe komunikacije razvijen je i vlastiti *ROS Message* za prijenos izračunatih udaljenosti koji se zove *DistanceMsg* i njegova definicija je smještena u mapi *catkin\_ws/src/rpi\_project/msg*.

### 4.1. ROS

ROS (Robot Operating System) operacijski je sustav korišten za komunikaciju između više "čvorova" gdje "čvor" predstavlja jedan zaseban objekt koji je dio velikog sustava, npr. u nekom više-robotskom sustavu svaki se fizički robot u programu doživljava kao jedan "čvor". U ovome projektu svaki Raspberry uređaj također se doživljava kao jedan "čvor" te svaki od njih u svojoj glavnoj funkciji odmah na početku inicijalizira svoj "čvor", npr. za Rpi0 to se odrađuje pomoću funkcije *rospy.init\_node("Rpi0", anonymous=True)*. To je potrebno napraviti analogno za svaki Raspberry kako bi mogli međusobno komunicirati i razmjenjivati izmjerene udaljenosti.

## 4.2. Pokretanje programa

Za pokretanje programa potrebno je otvoriti pet terminala na Linux operacijskom sustavu. Na svakom terminalu potrebno se spojiti na jedan od Raspberry Pi uređaja pomoću naredbe `ssh pi@rpiN.local` gdje N predstavlja redni broj Raspberryja. Na petom terminalu potrebno se spojiti na Raspberry 2 odnosno potrebno je ponovno unijeti naredbu `ssh pi@rpi2.local` te u tom terminalu pokrenuti ROS naredbom `roscore`. Raspberry 2 je referentni uređaj koji pokreće ROS dok se ostali "spajaju" na njega. Pokretanje ROS-a je iznimno važno jer u protivnom uređaji neće biti u mogućnosti komunicirati. U ostalim terminalima pokreću se skripte i zadatak se obavlja. Rezultati izvedbe programa ispisuju se u svaki od terminala.

## 4.3. Proračunavanje udaljenosti i pozicija

Već opisani proces izračunavanja udaljenosti programski se odvija u nekoliko koraka. Prvi korak svakom Raspberryju je skenirati ostale uređaje pomoću bluetootha. Potom se iz skeniranih rezultata uzima RSSI vrijednost za sve ostale Raspberry uređaje te se prema već spomenutoj formuli određuje udaljenost do svakoga od njih. S obzirom na to da se postupak neprestano ponavlja i da se svakim novim mjerenjem dobivaju nove udaljenosti, prije proračunavanja pozicija udaljenosti je potrebno filtrirati, a to se radi tako da se uzima median vrijednost od zadnjih pet izračunatih udaljenosti te se takve udaljenosti šalju ostalim uređajima i koriste za proračun pozicija.

Pozicije se proračunavaju kako je opisano u ranijim poglavljima; koriste se izračunate udaljenosti svih uređaja formirane u matricu čije se vrijednosti potom predaju optimizacijskoj funkciji koja potom vraća koordinate za te vrijednosti te se taj proces odvija dok pogreška razlike stvarnih i izračunatih koordinata ne bude manja od zadane vrijednosti.

## 4.4. Rezultati računanja udaljenosti

Kao što je već rečeno rezultati izvođenja programa ispisuju se u terminal pa će stoga sve bitne informacije biti ispisane tamo. U ovom poglavlju bit će prikazani ispisi i matrice svakog Raspberryja u prvom odnosno drugom koraku i u kasnijim koracima.

```
pi@rpi0 - 100x26
Rp1: []
Rp2: [4.466835921509632]
Rp3: []

[[0. 0. 4.46683592 0. ]
 [0. 0. 0. 0. ]
 [0. 0. 0. 0. ]
 [0. 0. 0. 0. ]]

Korak 2.

Rp1: []
Rp2: [4.466835921509632, 4.466835921509632]
Rp3: [0.7943282347242815]

[[0. 0. 4.46683592 0.79432823]
 [0. 0. 0. 0. ]
 [0. 0. 0. 0. ]
 [0. 0. 0. 0. ]]

Korak 3.

pi@rpi1 - 101x26
Rp1: []
Rp2: [1.7782794100389228]
Rp3: []

[[0. 0. 0. 0. ]
 [0. 0. 1.77827941 0. ]
 [0. 0. 0. 0. ]
 [0. 0. 0. 0. ]]

Korak 2.

Rp1: []
Rp2: [1.7782794100389228, 1.4125375446227544]
Rp3: [0.31622776681683794]

[[0. 0. 0. 0. ]
 [0. 0. 1.59548848 0.31622777]
 [0. 0. 0. 0. ]
 [0. 0. 0. 0. ]]

Korak 3.

pi@rpi2 - 100x27
Korak 1.

Rp1: []
Rp2: [2.51188643150958]
Rp3: []

[[0. 0. 0. 0. ]
 [0. 0. 0. 0. ]
 [0. 2.51188643 0. ]
 [0. 0. 0. 0. ]]

Korak 2.

Rp1: []
Rp2: [2.51188643150958, 1.2589254117941673]
Rp3: []

[[0. 0. 0. 0. ]
 [0. 0. 0. 0. ]
 [0. 1.88548592 0. ]
 [0. 0. 0. 0. ]]

Korak 3.

pi@rpi3 - 101x27
Korak 1.

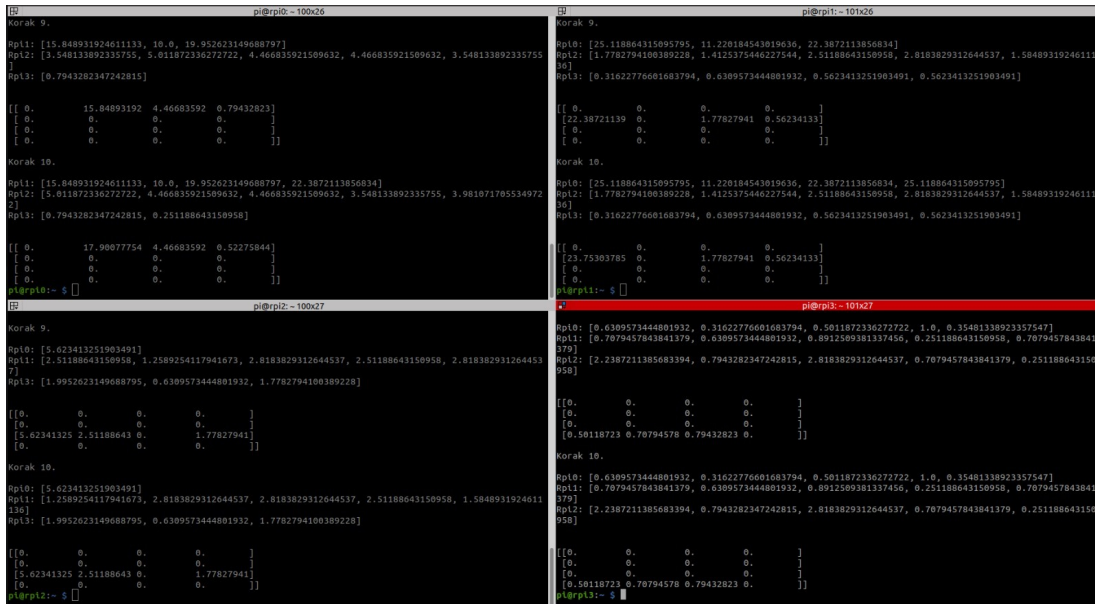
Rp1: []
Rp2: [0.35481338923357547]
Rp3: [1.9952623149688795]

[[0. 0. 0. 0. ]
 [0. 0. 0. 0. ]
 [0. 0. 0. 0. ]
 [0. 0.35481339 1.99526231 0. ]]

Korak 2.
```

Slika 4.1: Rezultati u početnim koracima

Na slici 4.1 prikazani su rezultati skeniranja i dobivenih matrica svakog od Raspberry za prvi odnosno drugi korak. Može se uočiti problem s bluetooth skeniranjem; određeni uređaji ne vide sve uređaje u svakom koraku. Idealno svaki Raspberry bi trebao vidjeti sve druge uređaje svakim novim skeniranjem, ali to nije slučaj. Taj problem riješen je tako da se samo primjene vrijednosti iz prethodnih koraka ako Raspberry nije vidio neki drugi Raspberry. Svaki Raspberry popunjava svoj redak matrice. U ostale retke matrice idu vrijednosti od ostalih Raspberryja kada ih oni pošalju.



**Slika 4.2:** Rezultati u kasnijim koracima

Slika 4.2 prikazuje vrijednosti koje poprimaju izračunate udaljenosti nakon više mjerenja; proces se odvija konstantno, ali za potrebe prikaza rezultata program je prekinut u desetom koraku. Veći broj mjerenja trebao bi značiti i preciznije rezultate. Na prikazu se ponovno vide matrice s izračunatim udaljenostima za svaki Raspberry uređaj kao i liste koje sadržavaju udaljenosti od zadnjih pet mjerenja. Konstantnim mjerenjem i proračunavanjem očekivano je da će se dobiti precizniji rezultati koji će težiti ka točnim udaljenostima. Ovaj prikaz to i potvrđuje; udaljenosti u desetom koraku svakako su preciznije nego u prvom tj. drugom koraku.

## 5. Zaključak

Nakon iznesenih uvodnih primjera može se zaključiti da je tehnološki napredak neizbježan i da je tehnologiju potrebno što više razvijati i prilagođavati joj se. Sukladno tome ovaj projekt pokušava razjasniti upotrebu i način korištenja uređaja Raspberry Pi Model B. Objašnjen je način korištenja spomenutog uređaja u ovom projektu, prikazan osnovni cilj zadatka, način na koji je zadatak programski riješen te su prikazani neki od primjera. Razrađena je komunikacija među svim korištenim uređajima te su uređaji izračunali potrebne udaljenosti.

Smatram da Raspberry Pi uređaji imaju veliki potencijal za širu primjenu i da je konstantno potrebno razvijati i poboljšavati komunikaciju kojom se oni međusobno služe. Ovaj projekt pruža jednostavan primjer komunikacije među Raspberryjima i zadatka kojeg su trebali obaviti te se nadam da će to potaknuti daljnji napredak i daljnja istraživanja.

## 6. Literatura

- Zhiqiang Cao, Ran Liu, Chau Yuen, Achala Athukorala, Benny Kai Kiat Ng, Muraleetharan Mathanraj, i U-Xuan Tan. Relative localization of mobile robots with multiple ultra-WideBand ranging measurements. U *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Rujan 2021. doi: 10.1109/iros51168.2021.9636017. URL <https://doi.org/10.1109/iros51168.2021.9636017>.
- Siyuan Chen, Dong Yin, i Yifeng Niu. A survey of robot swarms' relative localization method. *Sensors*, 22(12):4424, Lipanj 2022. doi: 10.3390/s22124424. URL <https://doi.org/10.3390/s22124424>.
- Yong Cui, Qiusheng Wang, Haiwen Yuan, Xiao Song, Xuemin Hu, i Luxing Zhao. Relative localization in wireless sensor networks for measurement of electric fields under HVDC transmission lines. *Sensors*, 15(2):3540–3564, Veljača 2015. doi: 10.3390/s150203540. URL <https://doi.org/10.3390/s150203540>.
- Ke Liao, Bin Da, Wei Wang, Hong Yi, i Haihua Yu. Relative localization without beacon nodes based on TDOA distance measurement. U *2015 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*. IEEE, Kolovoz 2015. doi: 10.1109/apwimob.2015.7374926. URL <https://doi.org/10.1109/apwimob.2015.7374926>.

## **Određivanje topologije mreže i rasporeda čvorova**

### **Sažetak**

U današnjem digitalnom svijetu u mnogim poslovima čovjeka zamjenjuje tehnologija. Sve više poslova moguće je obaviti korištenjem određene tehnologije. Jedan primjer uređaja koji se danas redovno koristi za mjerenje različitih parametara jest Raspberry Pi. Ovaj projekt predstavlja jednostavan zadatak koji je trebao biti obavljen pomoću četiri Raspberry Pi uređaja. Ova tehnička dokumentacija navodi više informacija o samom uređaju, zadatke koje su trebali biti izvršeni te način na koji je to postignuto.

**Ključne riječi:** Raspberry Pi, komunikacija, tehnologija

## **Determination of network topology and node arrangement**

### **Abstract**

In today's digital world, man replaces technology in many jobs. More and more jobs can be done using certain technology. One example of a device that is regularly used today to measure various parameters is the Raspberry Pi. This project represents a simple task that should be done using four Raspberry Pi devices. This technical documentation provides more information about the device itself, the tasks it was supposed to perform and the way in which it was achieved.

**Keywords:** Raspberry Pi, communication, technology