

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

**Mogućnosti i ograničenja
razvojnog sustava
ESP32-C3-DevKitM-1 u razvoju
Bluetooth aplikacija**

Jelena Gavran

Voditelj: prof. dr. sc. Hrvoje Džapo

Zagreb, siječanj 2023.

SADRŽAJ

Popis slika	iii
1. Uvod	1
2. Razvojni sustav ESP32-C3-DevKitM-1	2
2.1. BLE protokol	3
3. Aplikacijska programska sučelja	8
3.1. Bluedroid	8
3.2. NimBLE	11
3.2.1. BLE Mesh	13
3.3. BluFi	14
4. Analiza razvojnog sustava i ograničenja	17
4.1. Analiza mobilnom aplikacijom <i>nRF Connect</i>	17
4.2. Ograničenja razvojnog sustava	19
5. Zaključak	21
6. Literatura	22

POPIS SLIKA

2.1.	Konfiguracija razvojnog sustava ESP32-C3-DevKitM-1 [9]	2
2.2.	Blok dijagram modula ESP32-C3 [8]	3
2.3.	Arhitektura BLE stoga [9]	4
2.4.	Struktura paketa koji se šalju BLE protokolom [14]	5
2.5.	Komunikacijske mogućnosti u BLE-u [4]	5
3.1.	Dijagram <i>Bluedroid</i> arhitekture	9
3.2.	Sekvencijski dijagram komunikacije GATT poslužitelja i klijenta [7] .	9
3.3.	Ispis klijenta u demo aplikaciji za GATT profil	10
3.4.	Ispis klijenta u demo aplikaciji za SPP profil	11
3.5.	Ispis klijenta u demo aplikaciji za propusnost	11
3.6.	Arhitektura <i>NimBLE</i> stoga [7]	12
3.7.	Ispis nakon uspješnog spajanja na Wi-Fi vezu	12
3.8.	Ispis nakon neuspješnog spajanja na Wi-Fi vezu	12
3.9.	Obavijest u mobilnoj aplikaciji nakon spajanja čvora	14
3.10.	Čvorovi spojeni u mrežu s mobilnim uređajem	14
3.11.	Sekvencijski dijagram povezivanja putem <i>Blu-Fi</i> konfiguracije [7] . .	15
3.12.	Korisničko sučelje mobilne aplikacije <i>ESP BLE Prov</i>	16
3.13.	Ispis demo aplikacije nakon povezivanja na Wi-Fi pristupnu točku . .	16
4.1.	Graf RSSI vrijednosti za tri razvojna sustava	18
4.2.	Graf RSSI vrijednosti u ovisnosti o udaljenosti	18
4.3.	Graf RSSI vrijednosti u ovisnosti o udaljenosti	19
4.4.	Razlike između klasične Bluetooth i BLE tehnologije [10]	20

1. Uvod

Pojam „internet stvari“ neizostavan je u današnjem razvoju bežičnih i pametnih uređaja. IoT (engl. *Internet of things*) krovni je naziv koji obuhvaća milijune uređaja, odnosno „stvari“ povezanih na internet, koji pohranjuju i razmjenjuju podatke s drugim uređajima i sustavima također povezanih na internet. [3]

Za razvoj IoT uređaja potrebni su mikrokontroleri s mogućnošću bežičnog povezivanja. Serija ESP32 mikrokontrolera tvrtke *Espressif* ozbiljan je konkurent među bežičnim uređajima zbog niske potrošnje, visoke otpornosti na temperature, te najvažnije, jednostavnom bežičnom povezivosti. [1] Jedan takav čip je ESP32-C3, koji pruža Wi-Fi i Bluetooth povezivanje. Čip je integriran u nekoliko različitih modula, koji su pak dio razvojnih sustava koje proizvodi *Espressif*. Za izradu ovog rada odabran je razvojni sustav ESP32-C3-DevKitM-1.

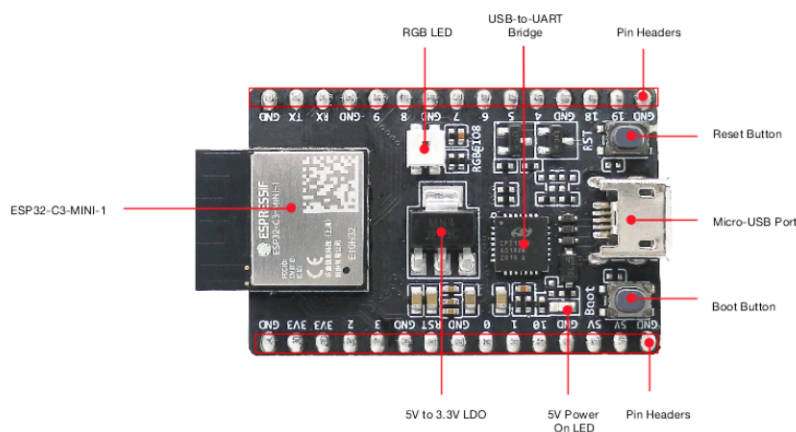
Ovaj seminar analizira mogućnosti koje pruža ESP32-C3-DevKitM-1 u razvoju Bluetooth programskih rješenja. Opisana su programska aplikacijska sučelja (engl. *Application Programming Interface* - *API*) koje modul podržava i demo aplikacije uz pripadna sučelja. Mobilnom aplikacijom za ispitivanje Bluetooth povezivosti ispitana su svojstva razvojnog sustava. Također, razložena su i ograničenja sustava pri korištenju Bluetooth protokola.

Rad je podijeljen u cjeline kako slijedi. Drugo poglavlje „*Razvojni sustav ESP32-C3-DevKitM-1*“ opisuje osnovne karakteristike korištenog razvojnog sustava kao ciljane hardverske platforme te su opisane najvažnije značajke BLE protokola. U trećem poglavlju „*Aplikacijska programska sučelja*“ opisani su API-ji koji se mogu koristiti uz razvojni sustav te su analizirane demo aplikacije uz pripadne API-je. U četvrtom poglavlju „*Analiza razvojnog sustava i ograničenja*“ korištenjem mobilne aplikacije *nRF Connect* ispitana su Bluetooth svojstva razvojnog sustava te su navedena ograničenja u izradi Bluetooth aplikacija.

2. Razvojni sustav

ESP32-C3-DevKitM-1

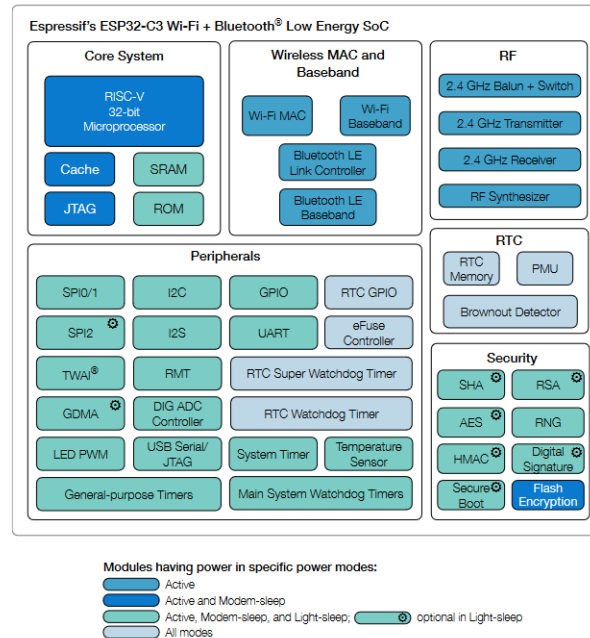
Razvojni sustav temelji se na modulu ESP32-C3-MINI-1. Modul je jedan u nizu ESP32-C3 serije SoC (engl. *System on Chip*) platformi tvrtke *Espressif*, te sadrži jednojezgri 32-bitni procesor s RISC-V arhitekturom koji radi na frekvenciji do 160 MHz. Modul sadrži 400 KB memorije tipa SRAM (engl. *Static random-access memory*), od kojih je 16 KB rezervirano za priručnu memoriju (engl. *cache*), 384 MB memorije tipa ROM (engl. *Read-only memory*) te 4 MB memorije tipa *Flash*. Od periferije sadrži 22 programabilna GPIO pina (engl. *General Purpose Input Output*), te digitalna sučelja SPI, UART, I2C i I2S. Također sadrži upravljače za sučelja USB i JTAG koji se mogu koristiti za efikasnije otklanjanje pogrešaka u kodu (engl. *debugging*). [8] Konfiguracija sustava prikazana je na slici 2.1.



Slika 2.1: Konfiguracija razvojnog sustava ESP32-C3-DevKitM-1 [9]

Budući da modul ima funkciju RF (engl. *radio frequency*) primopredajnika, podržava protokol Bluetooth s podrškom za velike udaljenosti. Druga važna značajka je podsustav za Wi-Fi, koji omogućava propusnost do 20 Mbps protokolom TCP te maksimalnu propusnost od 30 Mbps koristeći protokol UDP.

Modul ESP32-C3-MINI-1 bežični je uređaj niske potrošnje energije (engl. *ultra-low-power*) primarno namijenjen razvoju aplikacija koje koriste *Bluetooth Low Energy* (BLE) protokol ili Wi-Fi. Na slici 2.2 nalazi se blok shema modula sa svim dostupnim značajkama.



Slika 2.2: Blok dijagram modula ESP32-C3 [8]

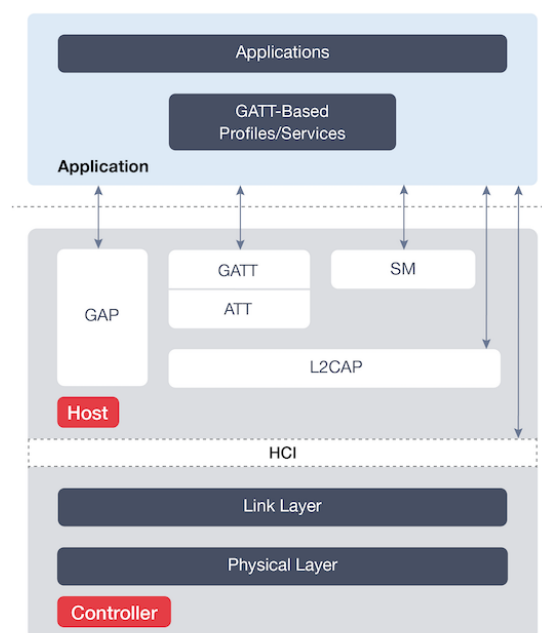
2.1. BLE protokol

BLE je vrsta bežične komunikacije namijenjena komunikaciji kratkog dometa s niskom potrošnjom energije. Razvijen je kako bi se postigao standard vrlo male snage koji radi s baterijom veličine kovanice (engl. *coin-cell batteries*) nekoliko godina. U odnosu na proizvode koji koriste klasičnu Bluetooth tehnologiju, BLE uređaji troše samo dio energije te omogućavaju malenim uređajima s malim baterijama bežično povezivanje s uređajima koji koriste Bluetooth. BLE protokol ostvaruje nisku potrošnju tako što boravi u stanju mirovanja dok nije povezan s drugim uređajima. Zbog toga može prenositi male količine podataka u kratkom vremenskom periodu. [13]

BLE radi u istom opsegu od 2,4 GHz kao i standardni Bluetooth, no koristi različite kanale od standardnog Bluetootha. Koristi 40 kanala od 2 MHz za prijenos podataka korištenjem modulacije Gaussova pomaka frekvencije (metoda koja se koristi za glatke prijelaze između podatkovnih impulsa), zbog čega skokovi frekvencije proizvode manje smetnji u usporedbi sa standardnom Bluetooth komunikacijom.

BLE je tehnologija adaptivnog skakanja frekvencije (engl. *Adaptive frequency hopping* - AFH) koja može koristiti samo podskup svih dostupnih frekvencija kako bi se izbjegle sve frekvencije koje koriste druge neprilagodljive tehnologije. To omogućuje prelazak s lošeg kanala na poznati dobar kanal korištenjem specifičnog algoritma za skakanje frekvencije, koji određuje sljedeći dobar kanal za korištenje.

Arhitektura BLE tehnologije naziva se još i BLE stog zbog slojevite strukture. Stog se sastoji od dvije glavne komponente: BLE upravljač (engl. *controller*), koji prenosi podatke, te BLE domaćin (engl. *host*), koji definira odnos povezanih uređaja. Prikaz arhitekture stoga nalazi se na slici 2.3.



Slika 2.3: Arhitektura BLE stoga [9]

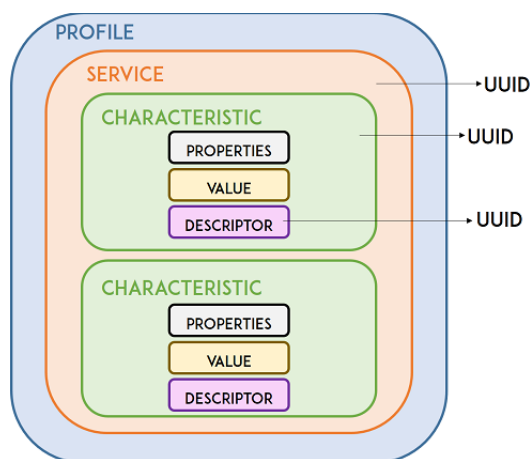
Osnovni profil koji implementiraju svi Bluetooth uređaji naziva se generički profil pristupa (GAP), koji pruža puni standardni okvir za kontrolu BLE uređaja u komunikacijskim metodama od točke do točke (engl. *point-to-point*) i emitiranju podataka. Profil definira kako BLE uređaji mogu otkriti druge uređaje i povezati se s njima te kako uspostaviti sigurnost i privatnost preko veze. Također detaljno opisuje na koji način uređaji mogu biti odašiljači i promatrači te prenositi podatke bez da su u stanju veze, odnosno bez direktne povezanosti s drugim uređajem. Postoje četiri uloge GAP profila:

- emiter (engl. *broadcaster*): šalje oglase,
- promatrač (engl. *observer*): prima oglase,

- periferija (engl. *peripheral*): uvijek u načinu oglašavanja i u ulozi *slave*,
- centar (engl. *central*): nikada ne šalje oglase, uvijek u ulozi master.

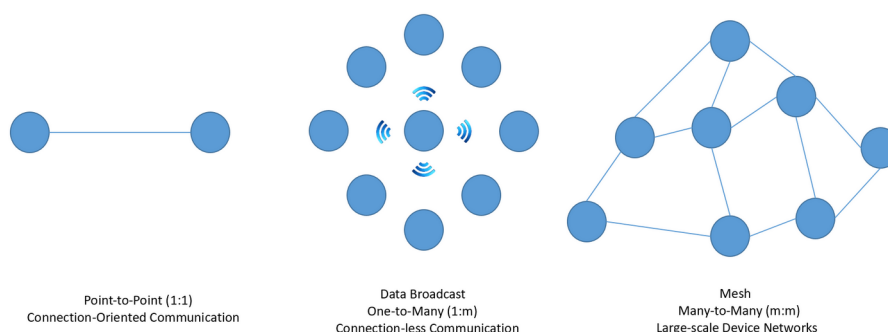
Generički atributni profil (GATT) odgovoran je za razmjenu podataka i određuje njihovu strukturu. Definira dvije uloge uređaja: GATT poslužitelj i GATT klijent. BLE uređaj može implementirati samo jednu, ili pak istovremeno obje uloge.

GATT poslužitelj najčešće je implementiran u ugradbenom računalu. Poslužitelj implementira tablicu atributa (adresirani dijelovi informacija) strukturiranu u obliku usluga i karakteristika; točnije, sadrži korisne podatke kojima može pristupiti udaljeni klijent. Usluge su skupine karakteristika, odnosno korisničkih podataka koje uređaj želi poslati. Na slici 2.4 prikazana je opisana hijerarhijska struktura podatkovnog paketa.



Slika 2.4: Struktura paketa koji se šalju BLE protokolom [14]

BLE protokol nudi tri komunikacijske mogućnosti različitih topologija veze. Prikazane su na slici 2.5.



Slika 2.5: Komunikacijske mogućnosti u BLE-u [4]

Prva i najraširenija metoda jest od točke do točke (engl. *point-to-point*). Koristi se u uređajima gdje je povezivanje 1:1, odnosno gdje je moguće međusobno spajanje

samo dva uređaja. Većina uređaja u svakodnevnoj uporabi koriste ovu topologiju, primjerice zvučnici, pametne igračke, satovi i uređaji za praćenje zdravlja. Također se naziva i komunikacijom usmjerenom na povezivanje (engl. *connection-oriented communication*), budući da se temelji na povezivanju dvaju uređaja. Iako početna verzija ove metode dopušta spajanje samo dva uređaja, nove verzije protokola omogućuju vezu $n:1$, što omogućava spajanje više uređaja na jedan centralni uređaj. Razlikuje se od metode emitiranja podataka zbog uloge čvorova u spoju. BLE stog podržava sigurnosnu zaštitu za ovu vrstu topologije 128-bitnim AES algoritmom, no ovom je metodom spajanja zaštita opcionalna.

U ovom načinu povezivanja uređaji implementiraju jednu od dvije uloge GAP profila, a to su centar i periferija. Centralni je uređaj najčešće onaj koji troši više resursa, primjerice računalo ili mobitel, dok je periferni uređaj ugradbeno računalo s niskom potrošnjom. Nove verzije protokola podržavaju spajanje više perifernih uređaja na centralni.

Iduća metoda je emitiranje podataka (engl. *data broadcast*), čija je topologija povezivanja $1:n$. Emitiranje podataka naziva se još i komunikacijom bez povezivanja. Ovo je jednosmjerna metoda komunikacije gdje uređaj emitira svoje podatke svim susjednim uređajima u RF rasponu. Također se koristi za usluge lokacije niske točnosti (margina pogreške od 1,5 metra), primjerice osnovna navigacija u zatvorenom prostoru i pronalaženje puta. Centralni čvor koji emitira podatke još se naziva i BLE odašiljačem. Arhitektura BLE protokola ne podržava sigurnost ovakve vrste povezivanja, no sigurnosni algoritmi se po potrebi mogu implementirati u aplikacijskom sloju.

Ovaj način povezivanja podržava uloge emitera i promatrača. Pri emitiranju podataka, središnji čvor odnosno odašiljač isporučuje podatke jednosmjernom vezom. Šalje reklamne pakete s podacima u ulozi oglašivača, dok promatrači skeniraju reklamne pakete i tako primaju podatke. Konfiguracija emitiranja podataka najprikladnija je za senzore koji otvoreno emitiraju svoje javne podatke svim zainteresiranim susjednim uređajima. Reklamni se paketi mogu konfigurirati tako da pri njihovu skeniranju promatrač, po potrebi, može zatražiti dodatne informacije od odašiljača posebnim zahtjevom. Paket zahtjeva za skeniranje ne može sadržavati nikakve korisničke podatke.

Postoji nekoliko nedostataka ovakve vrste povezivanja. Ova topologija podržava isključivo jednosmjernu vezu, što znači da odašiljač ne može primiti nikakvu korisnu informaciju od promatrača. Isto tako, svi uređaji u blizini primaju odašiljane pakete, te nije moguće slati podatke samo jednom uređaju.

Posljednji način jest mrežna topologija (engl. *mesh*). BLE *Mesh* koristi se za uspostavljanje komunikacije više-prema-više uređaja ($n:n$). Omogućuje stvaranje slo-

ženih velikih mreža i idealan je za nadzor, kontrolu i sustave automatizacije gdje je potrebna pouzdana međusobna komunikacija mnoštva uređaja. Ova metoda povećava domet i pokrivenost izvan BLE RF dometa, te pomaže u izbjegavanju fizičkih prepreka. Sigurnost je obavezna u ovom načinu rada, te je podržana BLE stogom. [4]

3. Aplikacijska programska sučelja

Aplikacijska programska sučelja služe za olakšano korištenje usluga, protokola i periferija koje nudi hardver. Dostupno je mnogo API-ja za ESP32-C3 [9], no ovdje je napravljen osnovni pregled programskih sučelja koja koriste Bluetooth.

ESP-IDF, službeni radni okvir za razvoj softvera na ESP mikrokontrolerima, podržava dva radna okvira kao BLE domaćin:

- *Bluedroid*, koji podržava klasični Bluetooth i BLE,
- *NimBLE*, koji podržava samo BLE.

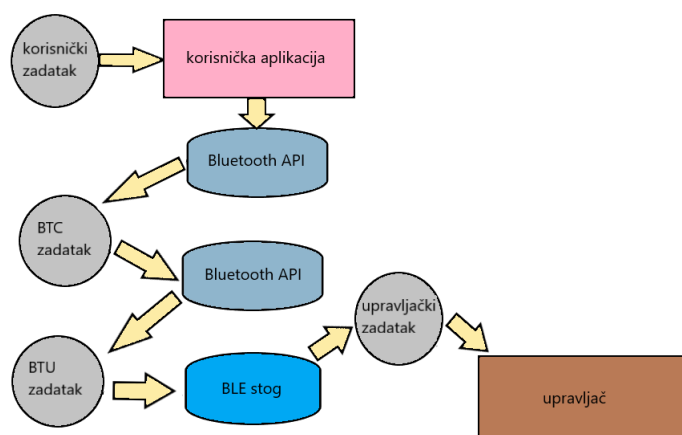
Preporuka je koristiti *Bluedroid* za aplikacije u kojima se može pojaviti potreba korištenja i klasičnog Bluetootha, dok je *NimBLE* prikladniji pri korištenju isključivo BLE protokola radi manje potrošnje memorije. [7]

3.1. Bluedroid

Bluedroid je zadani stog koji se koristi kao Bluetooth domaćin. Ovaj radni okvir, osim BLE protokola, podržava i klasični Bluetooth, te pruža jednostavnije istovremeno korištenje oba protokola. Arhitektura *Bluedroida* i njegov odnos sa BLE upravljačem prikazani su na slici 3.1.

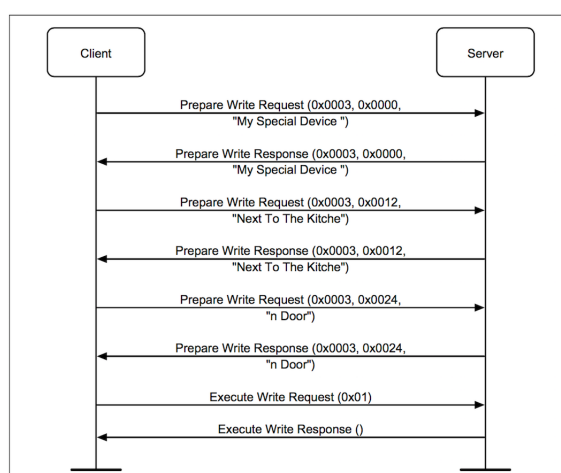
Bluedroid se može podijeliti na dva glavna sloja: BTC i BTU. BTC sloj služi kao sučelje aplikacijskom sloju te obrađuje zadatke vezane uz GATT profil. BTU sloj odgovoran je za interakciju sa BLE upravljačem. Pri obradi korisničkog zadatka, podzadaci vezani za Bluetooth proslijeđeni su nižim slojevima koji ih postupno obrađuju, te se proslijeđuju dalje niz hijerarhijsku strukturu. Svrha ovakvog dizajna je rasterećenje korisničkih zadataka i proslijeđivanje zadataka vezanih za Bluetooth nižim slojevima. [6]

U sklopu *Bluedroid* API-ja nudi se nekoliko demo aplikacija koje demonstriraju različite značajke. U nastavku su tri demo aplikacije koje nude softver za klijentsku i poslužiteljsku stranu.



Slika 3.1: Dijagram *Bluedroid* arhitekture

Prvi primjer implementira GATT poslužitelja i klijenta. Na početku svakog programa izvršavaju se konfiguracijski koraci, poput definiranja parametara oglašavanja te stvaranje usluga i karakteristika. Nadalje, obrađuju se događaji čitanja i pisanja, uključujući i zahtjev za pisanje dugačke karakteristike, što rezultira fragmentacijom nadolazećih podataka u manje pakete. Klijentska strana skenira uređaje u blizini te traži usluge i karakteristike željenog poslužitelja. Kada pronade traženi poslužitelj, uspostavlja se veza i vrši pretraga usluga. Na kraju, kada klijent pronade određenu karakteristiku u pretraženim uslugama, dobiva njezinu vrijednost i pretplaćuje se za obavijesti o toj karakteristici. Na slici 3.2 prikazan je sekvencijski dijagram komunikacije GATT poslužitelja i klijenta.



Slika 3.2: Sekvencijski dijagram komunikacije GATT poslužitelja i klijenta [7]

Ispis klijentske strane nalazi se na slici 3.3. Slika prikazuje klijentovo skenira-

nje dostupnih uređaja odnosno reklamnih paketa. Pretraživanje se ponavlja sve dok ne pronađe paket, te se tada spaja sa pronađenim uređajem, što je vidljivo u prvoj polovici ispisa. Druga polovica prikazuje ažuriranje parametara povezivanja te dohvat informacija o poslužitelju. Događaj `ESP_GATT_REG_FOR_NOTIFY_EVT` označava da se klijent pretplaćuje na poslužiteljevu promjenu parametara. `ESP_GATT_NOTIFY_EVT` označava da je klijent dobio obavijest na koju se pretplatilo.

```
I (3960) GATTTC_DEMO: 75 08 f3 e8 9a 43
I (3960) GATTTC_DEMO: searched Adv Data Len 31, Scan Response Len 0
I (3960) GATTTC_DEMO: searched Device Name Len 0
I (3970) GATTTC_DEMO:

I (3980) GATTTC_DEMO: 58 cf 79 e2 c1 7d
I (3980) GATTTC_DEMO: searched Adv Data Len 31, Scan Response Len 28
I (3980) GATTTC_DEMO: searched Device Name Len 14
I (3990) GATTTC_DEMO: ESP_GATTTC_DEMO
I (3990) GATTTC_DEMO:

I (4000) GATTTC_DEMO: searched device ESP_GATTTC_DEMO

I (4000) GATTTC_DEMO: connect to the remote device.
I (4010) GATTTC_DEMO: stop scan successfully
I (4060) GATTTC_DEMO: ESP_GATTTC_CONNECT_EVT conn_id 0, if 3
I (4060) GATTTC_DEMO: REMOTE BDA:
I (4060) GATTTC_DEMO: 58 cf 79 e2 c1 7d
I (4060) GATTTC_DEMO: open success
I (4370) GATTTC_DEMO: update connection params status = 0, min_int = 16, max_int = 32, conn_int = 32, latency = 0, timeout = 400
I (5490) GATTTC_DEMO: discover service complete conn_id 0
I (5490) GATTTC_DEMO: SEARCH RES: conn_id = 0 is primary service 1
I (5490) GATTTC_DEMO: start handle 40 end handle 43 current handle value 40
I (5500) GATTTC_DEMO: service found
I (5500) GATTTC_DEMO: UUID16: ff
I (5510) GATTTC_DEMO: Get service information from remote device
I (5510) GATTTC_DEMO: ESP_GATTTC_SEARCH_CMPL_EVT
I (5520) GATTTC_DEMO: ESP_GATTTC_REG_FOR_NOTIFY_EVT
I (5570) GATTTC_DEMO: ESP_GATTTC_CFG_MTU_EVT, Status 0, MTU 500, conn_id 0
I (5730) GATTTC_DEMO: ESP_GATTTC_NOTIFY_EVT, receive notify value:
I (5730) GATTTC_DEMO: 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e
I (5730) GATTTC_DEMO: write descr success
I (5810) GATTTC_DEMO: write char success
```

Slika 3.3: Ispis klijenta u demo aplikaciji za GATT profil

Iduća aplikacija također simulira vezu GATT poslužitelja i klijenta, no virtualnom serijskom vezom. U sustavima s klasičnim Bluetoothom, profil serijskog ulaza (engl. *Serial Port Profile* - *SPP*) koristi se za oponašanje serijske veze putem bežične Bluetooth veze. Budući da BLE nema standardnu SPP uslugu, mora se posebno implementirati. U *Bluedroid* okviru za SPP aplikacije koristi se UART transportni sloj, no po potrebi se može modificirati da koristi i druge serijske protokole, primjerice SPI. Svrha ovog načina komunikacije jest emulacija serijske veze, odnosno brzih i kratkih prijenosa podataka, zbog čega jednosmjerna propusnost može doseći do 1900 Kbps.

Kao i u prethodnoj aplikaciji, prikazano je spajanje klijenta na poslužitelj skeniranjem reklamnih paketa. Ispis klijentske aplikacije za virtualnu serijsku vezu prikazan je na slici 3.4. Ovdje su također ispisani dostupni atributi odnosno karakteristike koje klijent može pratiti. Kao što je ranije prikazano na slici 2.4, svaka karakteristika ima pridružena svojstva. Druga polovica ispisa prikazuje događaje koji se zbivaju nad pojedinim atributima. Svakom je atributu pridružena jedinstvena oznaka (UUID) na temelju koje se zaključuje o kojem se atributu radi. Popis jedinstvenih identifikatora te što koji označava nalazi se na službenoj stranici Bluetootha. [2]

```

I (4406) GATT_C_SPP_DEMO: Scan stop succeeded
I (4406) GATT_C_SPP_DEMO: Connect to the remote device.
I (4476) GATT_C_SPP_DEMO: EVT 40, gattc if 3
I (4476) GATT_C_SPP_DEMO: ESP_GATT_C_CONNECT_EVT: conn_id=0, gatt_if = 3
I (4476) GATT_C_SPP_DEMO: REMOTE BDA:
I (4486) GATT_C_SPP_DEMO: 00 00 00 00 00 00
I (4486) GATT_C_SPP_DEMO: EVT 2, gattc if 3
I (5326) GATT_C_SPP_DEMO: EVT 46, gattc if 3
I (5326) GATT_C_SPP_DEMO: EVT 7, gattc if 3
I (5326) GATT_C_SPP_DEMO: ESP_GATT_C_SEARCH_RES_EVT: start_handle = 40, end_handle = 65535, UUID:0xabf0
I (5336) GATT_C_SPP_DEMO: EVT 6, gattc if 3
I (5336) GATT_C_SPP_DEMO: SEARCH_CMPL: conn_id = 0, status 0
I (5426) GATT_C_SPP_DEMO: EVT 18, gattc if 3
I (5426) GATT_C_SPP_DEMO: +MTU:200

I (5426) GATT_C_SPP_DEMO: attr_type = PRIMARY_SERVICE,attribute_handle=40,start_handle=40,end_handle=65535,properties=0x0,uuid=0xabf0
I (5436) GATT_C_SPP_DEMO: attr_type = CHARACTERISTIC,attribute_handle=42,start_handle=0,end_handle=0,properties=0x6,uuid=0xabf1
I (5446) GATT_C_SPP_DEMO: attr_type = CHARACTERISTIC,attribute_handle=44,start_handle=0,end_handle=0,properties=0x12,uuid=0xabf2
I (5466) GATT_C_SPP_DEMO: attr_type = DESCRIPTOR,attribute_handle=45,start_handle=0,end_handle=0,properties=0x0,uuid=0x2902
I (5476) GATT_C_SPP_DEMO: attr_type = CHARACTERISTIC,attribute_handle=47,start_handle=0,end_handle=0,properties=0x6,uuid=0xabf3
I (5486) GATT_C_SPP_DEMO: attr_type = CHARACTERISTIC,attribute_handle=49,start_handle=0,end_handle=0,properties=0x12,uuid=0xabf4
I (5496) GATT_C_SPP_DEMO: attr_type = DESCRIPTOR,attribute_handle=50,start_handle=0,end_handle=0,properties=0x0,uuid=0x2902

I (5506) GATT_C_SPP_DEMO: Index = 2,UUID = 0xabf2, handle = 44
I (5516) GATT_C_SPP_DEMO: EVT 38, gattc if 3
I (5526) GATT_C_SPP_DEMO: Index = 2,status = 0,handle = 44

I (5576) GATT_C_SPP_DEMO: EVT 9, gattc if 3
I (5576) GATT_C_SPP_DEMO: ESP_GATT_C_WRITE_DESCR_EVT: status =0,handle = 45

I (5626) GATT_C_SPP_DEMO: Index = 5,UUID = 0xabf4, handle = 49

I (5626) GATT_C_SPP_DEMO: EVT 38, gattc if 3
I (5626) GATT_C_SPP_DEMO: Index = 5,status = 0,handle = 49

```

Slika 3.4: Ispis klijenta u demo aplikaciji za SPP profil

Treći primjer testira propusnost između klijentske i poslužiteljske aplikacije. U optimalnim uvjetima ona može dosegnuti do 700 Mbps između dvije ESP32 pločice. U ovom primjeru klijent kontinuirano šalje podatke poslužitelju, a ispis prikazuje količinu podataka poslanu svake sekunde. Vidljivo je da su vrijednosti puno manje od idealnih 700 Mbps, te dosežu maksimalno oko 1500 bps. Ispis je prikazan na slici 3.5.

```

I (6060) GATT_C_DEMO: connect to the remote device.
I (6070) GATT_C_DEMO: stop scan successfully
I (6140) GATT_C_DEMO: ESP_GATT_C_CONNECT_EVT conn_id 0, if 3
I (6140) GATT_C_DEMO: REMOTE BDA:
I (6140) GATT_C_DEMO: 58 cf 79 e2 c1 7d
I (6150) GATT_C_DEMO: open success
I (7420) GATT_C_DEMO: ESP_GATT_C_CFG_MTU_EVT, Status 0, MTU 517, conn_id 0
I (7420) GATT_C_DEMO: ESP_GATT_C_SEARCH_RES_EVT
I (7420) GATT_C_DEMO: service found
I (7430) GATT_C_DEMO: UUID16: ff
I (7430) GATT_C_DEMO: ESP_GATT_C_SEARCH_CMPL_EVT
I (7440) GATT_C_DEMO: ESP_GATT_C_REG_FOR_NOTIFY_EVT
I (7580) GATT_C_DEMO: write descr success
I (8590) GATT_C_DEMO: write success, write Bit rate = 197 Byte/s, = 1576 bit/s, time = 1s
I (9590) GATT_C_DEMO: write success, write Bit rate = 188 Byte/s, = 1504 bit/s, time = 1s
I (10600) GATT_C_DEMO: write success, write Bit rate = 188 Byte/s, = 1504 bit/s, time = 1s
I (11600) GATT_C_DEMO: write success, write Bit rate = 188 Byte/s, = 1504 bit/s, time = 1s
I (12600) GATT_C_DEMO: write success, write Bit rate = 187 Byte/s, = 1496 bit/s, time = 1s

```

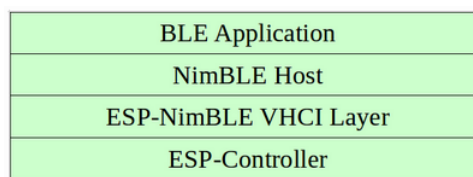
Slika 3.5: Ispis klijenta u demo aplikaciji za propusnost

3.2. NimBLE

NimBLE je konfigurabilan BLE stog koji pruža funkcije BLE domaćina i upravljača. *ESP-IDF* podržava samo funkcionalnost domaćina, dok je temeljni upravljač isti za

sve API-je. Podržava većinu značajki koje nudi *NimBLE* API, uključujući BLE *Mesh* koji omogućava mrežnu topologiju. *NimBLE* domaćin može se izvoditi unutar dretve aplikacije ili može imati vlastitu dretvu. Ova je fleksibilnost podržana konfiguracijom samog okvira te omogućuje višedretvenost Bluetooth procesa.

Arhitektura *NimBLE* stoga nalik je strukturi *Bluedroid* API-ja. Budući da je podržan samo *NimBLE* domaćin, ne i upravljač, postojeći upravljač nije kompatibilan sa domaćinom *NimBLE* okvira. Stoga je implementiran dodatni sloj koji služi kao sučelje između domaćina i upravljača (engl. *Virtual Host Controller Interface - VHCI*) specifičan za *NimBLE* stog. Arhitektura je prikazana na slici 3.6.



Slika 3.6: Arhitektura *NimBLE* stoga [7]

U nastavku je analizirana aplikacija koja koristi *NimBLE* stog, ali i Wi-Fi pove-
zivost. Primjer prikazuje paralelno izvođenje Wi-Fi i Bluetooth naredbi u modulu. Potrebno je unaprijed navesti parametre Wi-Fi mreže, odnosno ime mreže i lozinku, na koju će se sustav spojiti. Nakon inicijalizacije okvira, uređaj se spaja na Wi-Fi mrežu i šalje *ping* poruke za provjeru veze. Istovremeno razvojni sustav oglašava pa-
kete Bluetooth vezom koje klijenti mogu pročitati kako bi se povezali. Ispis aplikacije nakon uspješnog povezivanja na Wi-Fi prikazan je na slici 3.7, dok se na slici 3.8 nalazi ispis nakon neuspješnog spajanja na mrežu. Vidljivo je da je u oba slučaja omogućeno korištenje BLE stoga bez obzira na ostvarenu Wi-Fi vezu.

```

1 (572) wifi:state: assoc -> auth (0)
1 (572) wifi:connected with lokalnaveza, aid=1, channel 11, bssid= b2:0f:0e:01:f8:7e
1 (572) wifi:security: WPA2-PSK, phy: bgn, rssi: -37
1 (572) wifi:pm start, type: 1
1 (572) wifi:set rx beacon pti, rx_bcn_pti: 14, bcn_timeout: 14, mt_pti: 25000, mt_time: 10000
1 (582) wifi:0x-addba0 (1fa9, b2:0f:0e:01:f8:7e), tid=0, send, winSize=0
1 (622) wifi:binInt:00400, 0/102
1 (1382) esp_netif_handlers: sta ip: 192.168.43.11, mask: 255.255.255.0, gw: 192.168.43.1
1 (1382) wifi_prph_coex: get link:192.168.43.11
1 (1382) wifi_prph_coex: connected to ap SSID:lokalnaveza password:1234567890
1 (1382) BTDM_INIT: esp_bt_controller new release not implemented, return OK
1 (1402) BTDM_INIT: BT controller compile version [3407264]
1 (1402) BTDM_INIT: Bluetooth MAC: 58:c1:79:e3:62:76
1 (1412) wifi_prph_coex: BLE Host Task Started
1 (1412) BTDM: GAP procedure initiated: stop advertising.
1 (1412) BTDM: GAP procedure initiated: stop advertising.
1 (1412) BTDM: Device Address:58:c1:79:e3:62:76
1 (1412) BTDM: GAP procedure initiated: advertising.
1 (1412) BTDM: disc_mode=2
1 (1412) BTDM: adv_channel_map=0 own_addr_type=0 adv_filter_policy=0 adv_int_min=0 adv_int_max=0
1 (1412) BTDM:
64 bytes from 93.184.216.34 icmp_seq=1 ttl=64 time=155 ms
64 bytes from 93.184.216.34 icmp_seq=2 ttl=64 time=205 ms

```

Slika 3.7: Ispis nakon uspješnog spajanja na Wi-Fi vezu

```

1 (14962) wifi_prph_coex: connect to the AP fail
1 (14962) wifi_prph_coex: Failed to connect to SSID:myssid, password:mypassword
1 (14972) ping_sock: send error=0
1 (14972) BTDM_INIT: esp_bt_controller new release not implemented, return OK
1 (14982) BTDM_INIT: BT controller compile version [3407264]
1 (14992) BTDM_INIT: Bluetooth MAC: 58:c1:79:e3:62:76
1 (14992) wifi_prph_coex: BLE Host Task Started
1 (15002) BTDM: GAP procedure initiated: stop advertising.
1 (15002) BTDM: GAP procedure initiated: stop advertising.
1 (15002) wifi_prph_coex: Device Address:58:c1:79:e3:62:76
1 (15012) BTDM: GAP procedure initiated: advertising.
1 (15012) BTDM: disc_mode=2
1 (15022) BTDM: adv_channel_map=0 own_addr_type=0 adv_filter_policy=0 adv_int_min=0 adv_int_max=0
1 (15032) BTDM:
From 93.184.216.34 icmp_seq=1 timeout
1 (15972) ping_sock: send error=0
From 93.184.216.34 icmp_seq=2 timeout
1 (16972) ping_sock: send error=0

```

Slika 3.8: Ispis nakon neuspješnog spajanja na Wi-Fi vezu

3.2.1. BLE Mesh

Bluetooth *Mesh* umrežavanje omogućuje komunikaciju više-na-više uređaja (n:n) i optimizirano je za stvaranje mreža uređaja velikih razmjera. Uređaji mogu prenositi podatke drugim uređajima koji nisu u izravnom radijskom dometu izvornog uređaja. Na taj način isprepletene mreže mogu obuhvatiti vrlo velika fizička područja i sadržavati velik broj uređaja. Bluetooth *Mesh* nije bežična komunikacijska tehnologija, već mrežna tehnologija koja ovisi o BLE protokolu. Programsko sučelje *ESP-BLE-MESH* implementirano je u okviru *NimBLE* radnog okvira, te podržava samo BLE protokol.

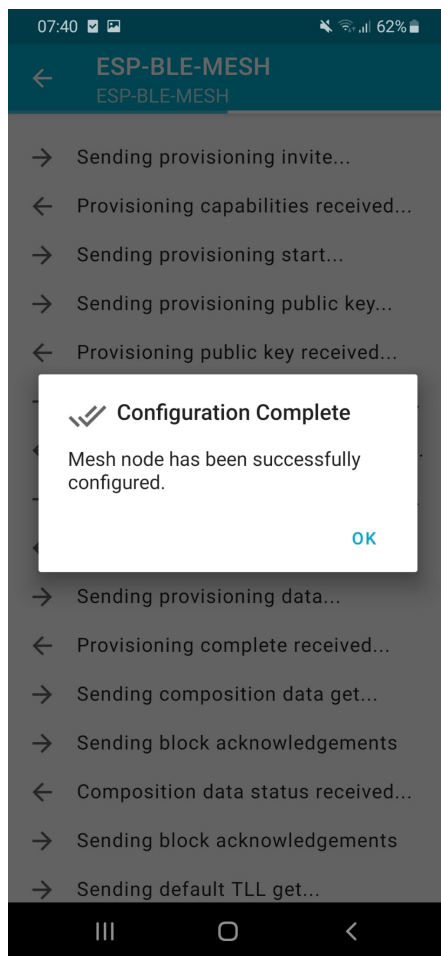
Arhitektura *ESP-BLE-MESH* sučelja sastoji se od pet ključnih dijelova:

- *Mesh* protokolni stog - osigurava protok mreže i razmjenu podataka,
- mrežno upravljanje - obavlja razne procedure za upravljanje mrežom,
- značajke - dodatne mogućnosti koje sučelje nudi,
- sloj nositelja mreže - sloj ključan za komunikaciju s BLE stogom,
- aplikacije - komuniciraju s *mesh* protokolnim stogom.

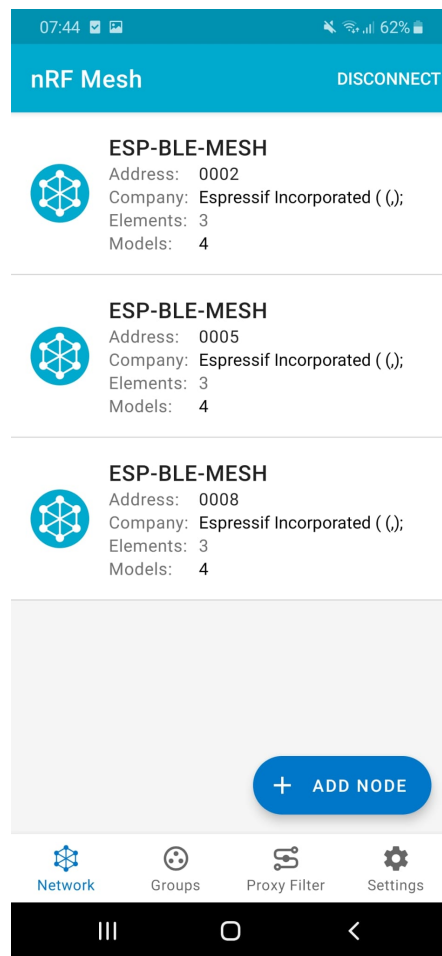
Sljedeći primjer koristi tri razvojna sustava ESP32-C3-DevKitM-1 i mobilni uređaj za demonstriranje mrežne topologije. Na sva tri razvojna sustava nalazi se ista poslužiteljska aplikacija koja oglašava reklamne pakete i tako omogućava spajanje mobilnog uređaja. Na mobilnom uređaju korištena je aplikacija *nRF Mesh* tvrtke *Nordic Semiconductors* koja omogućava mobilnom uređaju stvaranje mreže. Nakon skeniranja uređaja, u aplikaciji je moguće odabrati razvojni sustav s kojim će se uređaj spojiti. Odabirom željenog čvora aplikacija će pokušati pripojiti čvor mreži na sljedeći način:

- najprije se odspaja od čvora,
- zatim se pokušava ponovno spojiti,
- nakon uspješnog spajanja uspostavlja GATT uslugu,
- na kraju dohvaća podatke čvora i dodaje im aplikacijski ključ koji služi za kasnije ponovno spajanje.

Na slici 3.9 prikazana je obavijest koja se pojavi u mobilnoj aplikaciji nakon uspješnog pripajanja čvora u mreži. U pozadini iza obavijesti moguće je vidjeti gore opisani postupak uspostave veze. Na slici 3.10 nalazi se prikaz čvorova u mreži prikazanih u mobilnoj aplikaciji.



Slika 3.9: Obavijest u mobilnoj aplikaciji nakon spajanja čvora



Slika 3.10: Čvorovi spojeni u mrežu s mobilnim uređajem

3.3. BluFi

Iako nije namijenjen razvoju Bluetooth aplikacija, važno je istaknuti i *BluFi* radni okvir. *BluFi* služi za konfiguraciju Wi-Fi mreže putem Bluetooth kanala. Omogućuje siguran protokol za prijenos Wi-Fi konfiguracije na ESP32-C3. Koristeći te informacije, razvojni se sustav može spojiti na pristupnu točku ili postati pristupna točka na koju se drugi uređaji mogu spojiti.

Fragmentacija, enkripcija podataka i autentifikacija ključni su elementi okvira *BluFi*. Također je moguće prilagoditi način enkripcije proizvoljnim odabirom enkripcijskih algoritama.

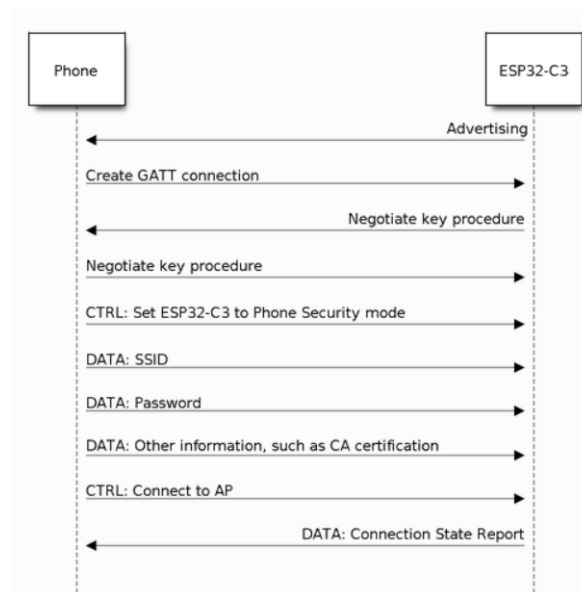
Povezivanje ESP32-C3 mikrokontrolera putem Bluetooth protokola na Wi-Fi mrežu odvija se na sljedeći način:

- ESP32-C3 najprije mora biti postavljen u način rada kao GATT poslužitelj

kako bi odašiljao reklamne podatke.

- Drugi uređaj s ulogom GATT klijenta skenira reklamne pakete i povezuje se s poslužiteljem po primitku željenih paketa.
- Nakon uspješne GATT veze, GATT klijent šalje podatkovni okvir radi uspostave ključa za šifriranje.
- Nakon dogovorene enkripcijske metode i uspostavljenog ključa, klijent šalje podatkovni okvir sa podacima o Wi-Fi konfiguraciji, uključujući ime mreže i lozinku.
- Klijent šalje kontrolni okvir nakon čijeg se primitka poslužitelj može spojiti na Wi-Fi.
- Nakon spajanja, ESP32-C3 šalje kontrolni okvir s podacima o statusu Wi-Fi veze.

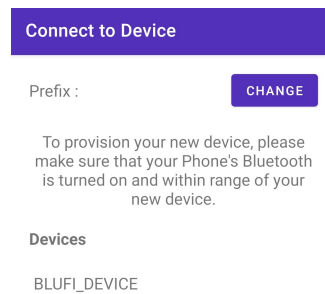
Dijagram opisanog povezivanja nalazi se na slici 3.11.



Slika 3.11: Sekvencijski dijagram povezivanja putem *Blu-Fi* konfiguracije [7]

Za demonstraciju mogućnosti *BluFi* okvira korištena je demo aplikacija u kojoj je modul ESP32-C3 povezan na Wi-Fi pristupnu točku mobilnog uređaja. Za povezivanje je korištena mobilna aplikacija *ESP BLE Prov* tvrtke *Espressif* koja je namijenjena demonstraciji povezivanja razvojnog sustava s mobilnim uređajem. Korisničko sučelje aplikacije prikazano je na slici 3.12. Za uspostavu veze potrebno je na mobilnom uređaju omogućiti Bluetooth, lokaciju te mobilnu pristupnu točku. Nakon uspješnog

povezivanja, aplikacija na razvojnom sustavu ispisat će poruku o uspješnom spajanju. Ispis se nalazi na slici 3.13.



Slika 3.12: Korisničko sučelje mobilne aplikacije *ESP BLE Prov*

```
I (600) BLUFI_EXAMPLE: BD ADDR: 58:cf:79:e3:62:75
I (610) BLUFI_EXAMPLE: BLUFI init finish
I (610) BLUFI_EXAMPLE: BLUFI VERSION 0102
I (394900) BLUFI_EXAMPLE: BLUFI ble connect
```

Slika 3.13: Ispis demo aplikacije nakon povezivanja na Wi-Fi pristupnu točku

4. Analiza razvojnog sustava i ograničenja

4.1. Analiza mobilnom aplikacijom *nRF Connect*

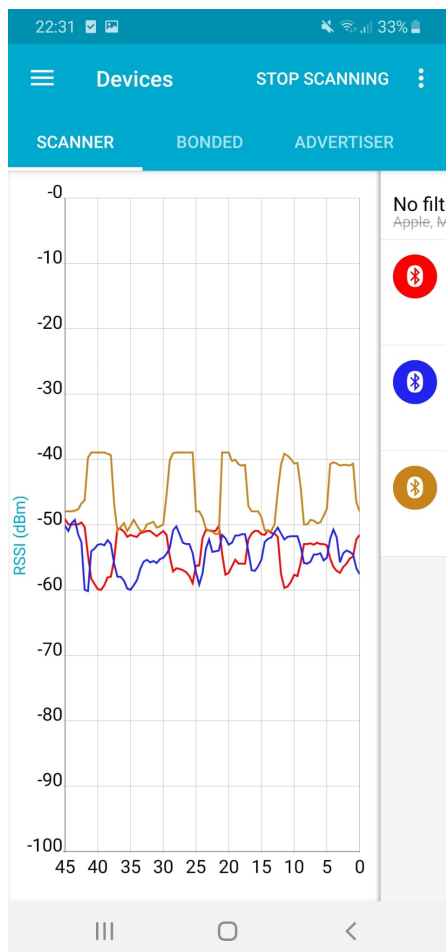
Za analizu razvojnog sustava korištena je mobilna aplikacija *nRF Connect* tvrtke *Nordic Semiconductor*. Pomoću nje moguće je pretražiti i povezati se sa BLE uređajima, kao i komunicirati s njima. Aplikacijom se također mogu analizirati podaci koje uređaj šalje pri oglašavanju te čitati informacije o samim uređajima i uslugama koje nude. [12]

Jedna od mogućnosti aplikacije je i prikaz RSSI (engl. *Received Signal Strength Indicator*) grafa. To je pokazatelj jačine primljenog signala te služi za mjerenje snage u primljenom radio signalu. RSSI je glavni indikator o jačini signala u danoj točki prostora. RSSI je relativna mjera, stoga je na grafičkom prikazu os RSSI vrijednosti označena dBm skalom, koja je negativna.

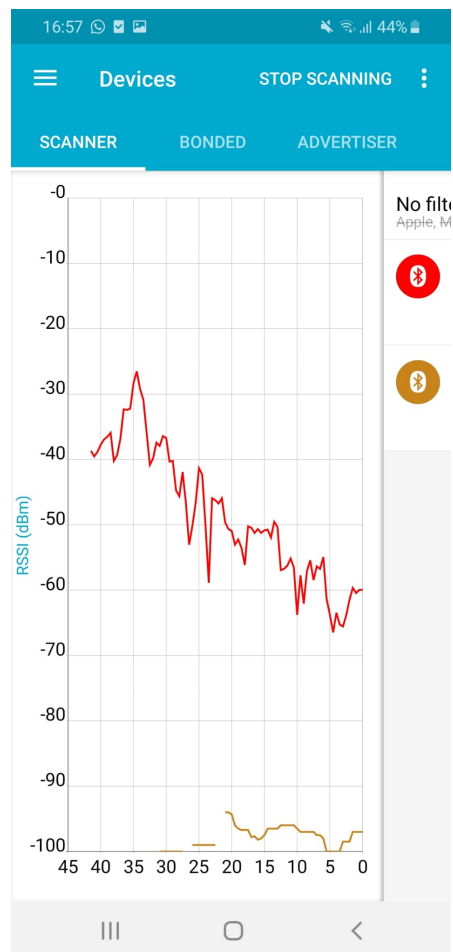
Na slici 4.1 prikazan je RSSI graf za *Bluedroid* demo aplikaciju *BLE_COMP_TEST* učitane na tri razvojna sustava. Mjerenje je obavljeno s udaljenosti od jednog metra, sa sustavima postavljenim u međusobnoj neposrednoj blizini. Očitana srednja RSSI vrijednost je otprilike -50 dBm te je vidljiva pravokutna struktura signala. Na temelju RSSI skale [11], dobivena vrijednost označava jako dobru jačinu signala. Budući da RSSI vrijednosti uvelike variraju između proizvođača i modela proizvoda, te ovise o mobilnom uređaju koji ih mjeri, dobivene su drukčije vrijednosti za razvojne sustave s istim softverom. Iako graf nije indikator valjanosti softvera, zanimljivo je prikazati kako izmjerene vrijednosti dobivene s čipova istog proizvođača iz iste linije rezultiraju različitim vrijednostima.

Iduće je mjerenje obavljeno na jednom razvojnom sustavu, također s demo aplikacijom *BLE_COMP_TEST*, te je svake sekunde jednakom količinom povećavana udaljenost mobilnog uređaja od razvojnog sustava radi prikazivanja ovisnosti RSSI vrijednosti o udaljenosti. Graf dobiven ovim mjerenjem prikazan je na slici 4.2. Mjerenje je

započeto na udaljenosti od nekoliko centimetara, što na grafu odgovara vrijednosti od -35 dBm, te završeno na udaljenosti od devet metara. Za konačnu udaljenost dobivena je vrijednost od otprilike -65 dBm, što je prema RSSI skali [11] minimalna jačina za aplikacije koje zahtijevaju pouzdanu i pravovremenu isporuku podatkovnih paketa. Za ovisnost RSSI vrijednosti o udaljenosti dobivena je obrnuta proporcionalnost, što je u skladu sa zakonom inverznog kvadrata, koji tvrdi da je određena fizikalna veličina obrnuto proporcionalna kvadratu udaljenosti od izvora te fizikalne veličine.



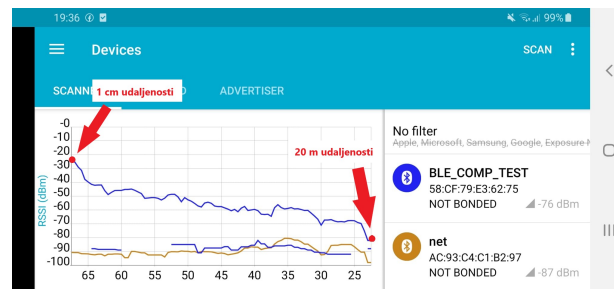
Slika 4.1: Graf RSSI vrijednosti za tri razvojna sustava



Slika 4.2: Graf RSSI vrijednosti u ovisnosti o udaljenosti

Provedeno je još jedno mjerenje kako bi se utvrdila maksimalna udaljenost na kojoj uređaj i dalje može slati podatkovne pakete. Prema skali [11] minimalna jačina signala mora iznositi oko -80 dBm kako bi se paketi uopće mogli razmjenjivati. Na slici 4.3 prikazan je još jedan graf RSSI vrijednosti, no mjerenje je završeno u trenutku kada je RSSI vrijednost pala na -80 dBm, odnosno 20 metara udaljenosti od uređaja. Ovo mjerenje pokazuje da je domet razvojnog sustava dostatan za kućanske uređaje

i aplikacije, no u slučaju da je potrebno pokriti veće područje, trebalo bi spojiti više uređaja u mrežnu topologiju.



Slika 4.3: Graf RSSI vrijednosti u ovisnosti o udaljenosti

4.2. Ograničenja razvojnog sustava

Klasična Bluetooth tehnologija razvijena je kao bežični standard, što je omogućilo razvoj bežičnih i prenosivih uređaja. *Classic BT* tehnologija koristi se za *streaming* aplikacije, poput prijenosa audiozapisa i datoteka. Radi na istim frekvencijama kao i BLE, no ima veći broj RF kanala. Klasični Bluetooth ima veću propusnost podataka, čak do 3 Mbps, dok BLE propušta maksimalno 0.27 Mbps. Također ima veću brzinu prijenosa podataka, do 3 Mbps, u usporedbi sa BLE protokolom čija brzina doseže najviše 1 Mbps. Za razliku od BLE protokola čija je glavna odlika niska potrošnja, zbog brze i nepredvidive komunikacije te složenih postupaka povezivanja *Classic BT* troši znatno više energije i time brže troši bateriju uređaja na kojem se nalazi. Latencija prijenosa je čak 16 puta veća nego u BLE uređajima. Isto tako, podržava samo *peer-to-peer* topologiju, odnosno 1:1, što znači da se istovremeno mogu povezati samo dva uređaja. Detaljnije razlike između verzija Bluetooth protokola nalaze se na slici 4.4. [10]

Jedan nedostatak modula ESP32-C3 jest što ne podržava klasični Bluetooth. Iako BLE nudi prednosti u odnosu na *Classic BT*, poput niske potrošnje i raznovrsnije podržane topologije, uređaji s BLE protokolom i s klasičnom Bluetooth tehnologijom ne mogu se međusobno povezati. Ta činjenica ograničava povezivanje i korištenje ESP32-C3 modula, stoga ne može komunicirati s uređajima koji rade na temelju klasičnog Bluetootha. Većina audio uređaja, poput Bluetooth zvučnika, zbog potrebe za prijenosom velike količine podataka koriste klasični Bluetooth radi boljih performansi u prijenosu podataka. ESP32-C3 ne može se povezati niti razmjenjivati podatke s takvim uređajima.

Feature	Bluetooth Classic	Bluetooth Low Energy (BLE)
Data rate	1Mbps for B 2-3Mbps for EDR	Up to 1Mbps
Power Consumption	High (up to 1W)	Low (0.01W-0.5W)
Audio Streaming	Original audio protocol used in most of devices	New LE audio protocol will replace the classic protocol because of low energy requirements and easy to handle
Range	Limited range 10m – 50m	Long range introduced in Bluetooth 5.0 up to 1km in line of sight
RF bandwidth	2.4 GHz ISM band (2400-2483.5 MHz)	2.4 GHz ISM band (2400-2483.5 MHz)
Number of channels	79 RF channel each of 1MHz	40 RF channel each of 2MHz
Modulation Technique	GFSK for Basic data rat 8-DPSK or $\pi/4$ -DQPSK for Enhanced data rate	GFSK
Topology	Peer to peer (1:1)	Peer to peer (1:1) Star topology (many:1) Broadcast (1:many) Mesh (many:many)

Slika 4.4: Razlike između klasične Bluetooth i BLE tehnologije [10]

Isto tako, BLE koristi pojas od 2,4 GHz kao i drugi bežični protokoli među kojima su Bluetooth i Wi-Fi. Zbog toga može doći do interferencije, što znači da dolazi do usporavanja prijenosa podataka među uređajima. Što se više uređaja nalazi na istoj frekvenciji, to su veća interferencija i latencija. Jedno od rješenja je fizički udaljiti uređaje koji koriste istu frekvenciju. [5]

Kao što je ranije opisano, BLE uređaji mogu prenositi samo male količine podataka u kratkom vremenskom periodu. Zbog tog ograničenja ne može primiti izravne glasovne naredbe, što se često koristi u IoT uređajima. Ipak, postoje pomoćni uređaji koji pretvaraju zvučne signale u kratke nizove bajtova koji se zatim prosljeđuju uređaju. Jedan takav uređaj je tzv. *voice assistant*.

5. Zaključak

Provedeno je istraživanje mogućnosti razvojnog sustava ESP32-C3-DevKitM-1 u razvoju Bluetooth aplikacija. Opisana su tehnička svojstva razvojnog sustava te je detaljnije opisan *Bluetooth Low Energy* (BLE) protokol. Navedena su aplikacijska programska sučelja koja sustav podržava te su demonstrirane njihove mogućnosti demo aplikacijama. Mobilnom aplikacijom *nRF Connect* ispitana su svojstva razvojnog sustava kao Bluetooth uređaja. Naposljetku su opisana ograničenja modula pri izradi Bluetooth aplikacija.

Ovaj seminar ponajprije služi kao uvid u mogućnosti razvojnog sustava ESP32-C3-DevKitM-1. Analiza programskih sučelja pokazuje da razvojni sustav nudi velik broj značajki koje se, koristeći analizirana sučelja, mogu jednostavno koristiti i implementirati. Isto tako, analiza jačine emitiranog Bluetooth signala pokazuje da uređaj dobro funkcionira na udaljenostima dovoljnima za pokrivanje kućanstva, no isto tako nudi mogućnost stvaranja mreže za pokrivanje veće površine. Jedino ozbiljnije ograničenje sustava jest nekompatibilnost s uređajima koji rade isključivo na temelju klasičnog Bluetootha. Ipak, provedenim istraživanjem dokazano je da modul ESP32-C3 opravdava svoju nadmoć u IoT primjeni.

6. Literatura

- [1] Top 25 iot development boards. 2021. URL <https://www.intuz.com/guide-on-top-iot-development-boards>.
- [2] 2023. Bluetooth. URL <https://www.bluetooth.com/>.
- [3] Mohammad Afaneh. What are iot devices. 2022. URL <https://www.simplilearn.com/iot-devices-article>.
- [4] Embedded Centric. Ble profiles, services, characteristics, device roles and network topology. 2022. URL <https://embeddedcentric.com/lesson-2-ble-profiles-services-characteristics-device-roles-and-network-topology/>.
- [5] Developex. Limitations of ble in smart home. 2017. URL <https://developex.com/blog/limitations-of-ble-in-smart-home/>.
- [6] Espressif. *ESP32 Bluetooth Architecture*, 2019. URL https://www.espressif.com/sites/default/files/documentation/esp32_bluetooth_architecture_en.pdf.
- [7] Espressif. *Bluetooth API*, 2022. URL <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/bluetooth/index.html>.
- [8] *ESP32-C3 Series Datasheet*. Espressif Systems, 2023. URL https://www.espressif.com/sites/default/files/documentation/esp32-c3_datasheet_en.pdf.
- [9] *ESP-IDF Programming Guide*. Espressif Systems, 2023. URL <https://docs.espressif.com/projects/esp-idf/en/v5.0/esp32c3/index.html>.

- [10] Linux Hint. Esp32 – bluetooth classic vs bluetooth low energy (ble). 2022. URL <https://linuxhint.com/bluetooth-classic-bluetooth-low-energy-esp32/>.
- [11] Metageek. Understanding rssi. URL <https://metageek-mix.netlify.app/training/resources/understanding-rssi.html>.
- [12] *nRF Connect for Mobile*. Nordic Semiconductor, 2023. URL <https://www.nordicsemi.com/Products/Development-tools/nRF-Connect-for-mobile>.
- [13] Bob Proctor. Bluetooth vs. bluetooth low energy. 2021. URL <https://www.link-labs.com/blog/bluetooth-vs-bluetooth-low-energy>.
- [14] Random Nerd Tutorials. Getting started with esp32 bluetooth low energy (ble) on arduino ide. URL <https://randomnerdtutorials.com/esp32-bluetooth-low-energy-ble-arduino-ide/>.

Mogućnosti i ograničenja razvojnog sustava ESP32-C3-DevKitM-1 u razvoju Bluetooth aplikacija

Sažetak

U ovom radu analizirane su mogućnosti razvojnog sustava ESP32-C3-DevKitM-1 kao Bluetooth uređaja. Opisane su biblioteke za kreiranje Bluetooth aplikacija koje uređaj podržava, te je provedena analiza demo aplikacija iz pripadnih biblioteka. Bluetooth povezivost sustava ispitana je mobilnom aplikacijom *nRF Connect*. Razložena su ograničenja razvojnog sustava u razvoju Bluetooth aplikacija.

Ključne riječi: ESP32-C3-DevKitM-1, BLE, Bluetooth API, nRF Connect