

Aripekka Nikupeteri

**BLUETOOTH LE -LAITTEIDEN ETÄSEURANTA AD HOC -MESH-
VERKON KAUTTA**

BLUETOOTH LE -LAITTEIDEN ETÄSEURANTA AD HOC -MESH- VERKON KAUTTA

Aripekka Nikupeteri
Opinnäytetyö
Syksy 2020
Tietotekniikan tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan tutkinto-ohjelma, ohjelmistokehitys

Tekijä: Aripekka Nikupeteri

Opinnäytetyön nimi suomeksi: Bluetooth LE -laitteiden etäseuranta ad hoc - mesh-verkon kautta

Opinnäytetyön nimi englanniksi: Remote monitoring of Bluetooth LE devices over ad hoc mesh network

Työn ohjaaja: Eino Niemi

Työn valmistumislukukausi ja -vuosi: Syksy 2020

Sivumäärä: 28

Työn aiheena oli tehdä Monidorille Android-pohjainen mobiilisovellus, joka vastaanottaa, lähettää sekä esittää käyttäjälle Monidrop-tippalaskurin mittaustietoja käyttämällä Bluetooth Low Energy -tiedonsiirtoa. Työssä suunniteltiin ja toteutettiin mobiilisovellus, joka osaa itsenäisesti hakea ja uudelleen lähettää Bluetooth-tiedonsiirtopaketteja ja esittää saapuneen tiedon mobiililaitteen käyttöliittymällä. Sovelluksen täytyy pystyä vastaanottamaan tietoa tippalaskurin lisäksi myös muilta mobiililaitteilta, jotta useammalla laitteella voi tarvittaessa muodostaa paikallinen ad hoc -verkko. Verkko mahdollistaa suuremman kantaman tippalaskurin tietojen vastaanottamisessa.

Työn toteutuksen aikana perehdyttiin tarkemmin Bluetooth Low Energy -kommunikointiin ja Android-ohjelmointiin. Työ toteutettiin käyttämällä Android Studio -kehitysympäristöä ja Java-ohjelmointikieltä.

Työn lopputuloksena syntyi ohjelma, joka vastaanottaa, lähettää ja esittää käyttäjälle Monidrop-laitteen lähettämät infuusiohoidon tiedot. Useasta mobiililaitteesta pystyy myös muodostamaan ad hoc -verkon, jossa laitteet kommunikoivat keskenään luoden näin laajemman toiminta-alueen.

Asiasanat: Bluetooth Low Energy, mobiilisovellus, langaton verkko, etäseuranta, Android, Java

ABSTRACT

Oulu University of Applied Sciences
Information Technology, Option of Software Development

Author: Aripekka Nikupeteri

Title of thesis: Remote monitoring of Bluetooth LE devices over ad hoc mesh network

Supervisor: Eino Niemi

Term and year when the thesis was submitted: Fall 2020

Pages: 28

The purpose of the work was to create an Android based remote monitoring software for Monidrop drip counter. The application utilizes the mobile devices onboard Bluetooth LE sensor to transfer and receive Monidrop infusion data. Objective was to create a dynamic ad hoc mesh network that is formed when the application is running on multiple devices at the hospital ward. Forming a network of devices will significantly increase the communication range of Bluetooth Low Energy devices. Finally, the application also has to present the received information on the UI for the user.

During this work, a deeper dive is done to examine Bluetooth Low Energy protocol and Android programming on a general level. Thesis was done by using Android Studio environment and Java programming.

The outcome of this work was an Android application, that can receive and display Bluetooth infusion data from a Monidrop drip counter. Application can also be used to create an ad hoc network using multiple mobile devices. Using an ad hoc network significantly increases the operating area of the infusion monitoring.

Keywords: Bluetooth Low Energy, Mobile application, Wireless network, Remote monitoring, Android, Java

SISÄLLYS

TIIVISTELMÄ	3
ABSTRACT	4
SISÄLLYS	5
SANASTO	6
1 JOHDANTO	7
2 KÄYTETYT TEKNOLOGIAT	8
2.1 Bluetooth-tiedonsiirtostandardi	8
2.2 Bluetooth Low Energy -teknologia	8
2.2.1 Generic Access Profile	9
2.2.2 Bluetooth-mainostus	10
2.3 Android-käyttöjärjestelmä	12
2.3.1 Android-sovellukset	12
2.3.2 Sovelluskehitys Androidissa	12
3 MONIDROP-LAITE INFUUSIOTIEDON SEURANTAAN	15
4 MOBIILISOVELLUS INFUUSIOTIEDON VASTAANOTTAMISEEN	17
4.1 Bluetooth mainostustiedon vastaanotto ja purku	17
4.1.1 Lupien asettaminen	17
4.1.2 Skannerin rakentaminen	18
4.1.3 Mittaustiedon purku	19
4.1.4 Muunnos bittioperaattoreilla	20
4.2 Ad hoc -verkon ja toistimen rakentaminen	21
4.3 Käyttöliittymä	22
4.3.1 Listanäkymän rakentaminen	22
4.3.2 Tiedonsiirto käyttöliittymälle	22
5 POHDINTA	24
LÄHTEET	26

SANASTO

Ad hoc Langattomista laitteista muodostettu yhteystapa, jossa laitteet liikennöivät keskenään ilman tukiasemaa.

BLE (Bluetooth Low Energy) Bluetooth 4.0 -versiossa lisätty spesifikaatio matalavirtaisille laitteille.

Bluetooth SIG (Special interest group) Bluetooth-standardia ylläpitävä järjestö.

GAP (Generic Access Profile) Bluetooth-mainostuksessa käytettävä protokolla.

IoT (Internet of Things) Esineistä koostuva verkko.

ISM (Industry, Medical, Science) Vapaa taajuusalue 2,4 GHz:n alueella.

LE (Low Energy) Bluetooth Low Energyn lyhenne.

Monidrop Monidorin kehittämä infuusiohoidossa käytettävä tippalaskuri.

PDU (Protocol Data Unit) on osa Bluetooth mainostuspakettia. Se sisältää mainostuspaketin otsikon sekä mahdollisen tietolastin.

UI (User interface) Käyttöliittymä.

1 JOHDANTO

Monidrop on oululaisen Monidor Oy:n kehittämä laite, joka mittaa tiputuksessa olevan potilaan infuusiotietoja. Monidrop kiinnitetään tippakammioon, jolloin sen sensorit mittaavat virtausnopeutta, määrää ja kulunutta aikaa. Monidropissa on sisäänrakennettu Bluetooth Low Energy (BLE) -lähetin, jota käyttäen se voi lähettää mittaustietoja lähistöllä sijaitseviin BLE-spesifikaatiota tukeviin laitteisiin. Tämän takia Monidorilla oli tarve kehittää ratkaisu, jossa hoitajat voisivat seurata mittaustietoja etänä. Lähtökohtana Monidorin tilaamalle työlle oli testata etämonitorointi-konseptia olemassa olevalla laitteella, jotta he saisivat palautetta sen tarpeellisuudesta ja mahdollisista rajoituksista ennen sitoutumista uuden etäseurantaan hyödyntävän tuotteen kehitykseen. (1.)

Työn tarkoituksena on luoda Android-pohjainen mobiilisovellus, joka vastaanottaa ja esittää käyttäjälle Monidropin lähettämää mittaustietoa. Tavoitteena on, ettei hoitajien tarvitsisi käydä potilaan huoneessa lukemassa laitteen mittaustietoja, vaan tarvittavat tiedot pystyttäisiin myös näkemään etänä. Projekti aloitettiin näistä lähtökohdista. Myöhemmin mobiilisovellusta testatessa todettiin kuitenkin Bluetooth Low Energyn liian lyhyt kantama. Sairaalaosastoissa on usein paksut seinät ja pitkät käytävät, jonka takia Bluetooth-signaalin kantama on tavallista lyhyempi. Ratkaisuksi ongelmaan päätettiin rakentaa mobiililaitteista koostuva ad hoc -verkko, joka lähettää Monidrop-laitteesta saapuvaa mittaustietoa mobiililaitteelta toiselle, kattaen näin koko osaston. Tästä syystä sovelluksen vaadittuihin ominaisuuksiin lisättiin myöhemmin vastaanotetun mittaustiedon uudelleenlähetyksen ja mahdollisuus luoda laitteista paikallinen verkko.

Tämä työ on rajattu vain Android-mobiilisovelluksen luontiin, Bluetooth LE -määntustiedon vastaanottamiseen, ad hoc -verkon luontiin sekä käyttöliittymän luomiseen. Toteutus Monidrop-laitteen lähettämälle Bluetooth tiedolle on jo tehty, eikä se ole osa tätä työtä.

2 KÄYTETYT TEKNOLOGIAT

2.1 Bluetooth-tiedonsiirtostandardi

Bluetooth on lyhyen kantaman tiedonsiirtostandardi, joka kehitettiin matkapuhelimen ja oheislaitteiden välistä kommunikointia varten. Tarkoituksena oli korvata aiemmin kaapelilla toimiva tiedonsiirto langattomaksi. Bluetooth toimii 2.4 GHz:n taajuusalueella, joka on lisensöimätön ISM (Industry, Medical, Science)-taajuusalue. Kanavia Bluetoothille on varattu yhteensä 79. Teknologiaa lähti alun perin kehittämään ruotsalainen telekommunikaatiojärjestelmien valmistaja Ericsson. Myöhemmin Ericsson perusti Bluetooth Special Interest Groupin (SIG), jonka tehtävänä on Bluetooth-protokollan kehittäminen ja standardisointi. (2; 3.)

Ensimmäinen versio Bluetoothista julkaistiin vuonna 1999, mutta sen kehitys jatkuu edelleen. Viimeisin versio on Bluetooth 5.2, joka julkaistiin tammikuussa 2020. Tärkeimpinä uusina ominaisuuksina on myöhemmissä versioissa tullut pidentetty kantama, tiedonsiirron nopeutuminen sekä Internet of Things (IoT) -laitteille suunniteltu Bluetooth Low Energy. (4; 5.)

2.2 Bluetooth Low Energy -teknologia

Bluetooth Low Energy (LE) on nimensä mukaan suunniteltu matalavirtainen versio Bluetoothista, joka julkaistiin osana Bluetooth 4.0 spesifikaatiota. Se suunniteltiin toimimaan pienellä paristolla toimivien IoT-laitteiden kanssa. Matala virrankulutus on saavutettu käyttämällä lyhyitä lähetysjaksoja nopeassa rytmissä, jolloin sensori on matalavirtaisessa lepotilassa lähetysten välissä. Kun muissa Bluetooth-versioissa lähetys saattaa kestää noin 100 ms, kestää lähetysten tekemisessä Bluetooth LE:llä vain muutaman millisekunnin. (6.)

Bluetooth Low Energy käyttää Bluetooth Classicista poiketen 40 eri kanavaa, joista vain kolme on varattu mainostusta varten. Muut kanavat on varattu nopeaan tiedonsiirtoon yhdistetyille laitteille. Kanavat on eritelty 2 MHz:n välein toisistaan, ja jokainen kanava vie 1 MHz:n leveyden taajuusalueesta. Koska WiFi-tiedonsiirto toimii samalla taajuusalueella, on mainostuskanavat aseteltu WiFi-kanavien väliin. (7.)

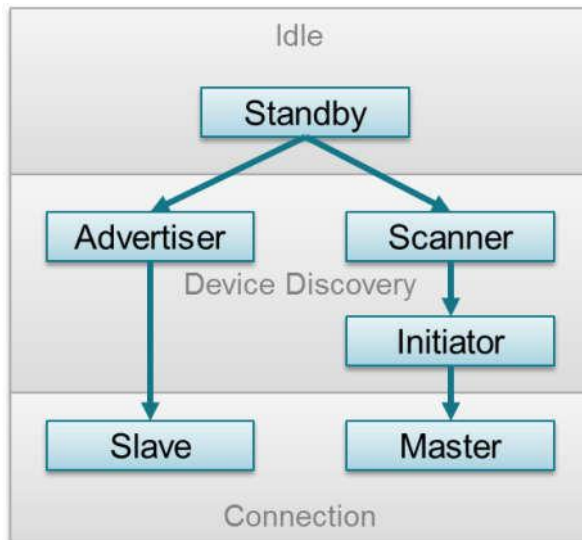
2.2.1 Generic Access Profile

Generic Access Profile (GAP) ohjaa laitteen toimia ennen mahdollisen yhteyden luomista. GAP-spesifikaatio määrittää laitteiden roolin. Bluetooth LE -laitteilla voi olla neljä eri roolia: lähettäjä, seuraaja, oheislaite ja keskuslaite. Lähettäjä on rooli, jossa laite kykenee ainoastaan lähettämään mainostusta. Seuraaja on lähettäjän vastakohta eli laite, joka vain vastaanottaa mainostusta. Oheis- ja keskuslaite roolissa olevat laitteet kykenevät sekä lähettämään että vastaanottamaan tietoa. Oheislaitteet ovat yleensä pienikokoisia ja vähävirtaisia laitteita, jotka yhdistävät keskuslaitteisiin. Keskuslaitteet puolestaan kuuntelevat oheislaitteita ja käynnistävät yhteyden laiteparin välille. Esimerkkinä oheislaitteesta voisi olla sykemittari, kun taas keskuslaitteena toimii yleensä matkapuhelin. (8.)

Roolin lisäksi GAP määrittää päällä oleville laitteille niiden tilan. Bluetooth LE -laite on aina jossain GAP-pinon määrittämässä tilassa. Tila määräytyy seuraavasti (kuva 1):

- **Standby.** Laite on lepotilassa, kun Bluetooth vastaanotinta ja lähetintä käynnistetään tai resetoidaan.
- **Advertiser.** Mainostustilassa laite mainostaa itseään alueella oleville keskuslaitteille. Mainostusta voivat tehdä vain oheislaitteet. Mainostusviestissä oheislaite viestittää tietoja, kuten laitteen osoitteen, ja Protocol Data Unit (PDU) -tyypin. PDU määrittää, millainen laitteen haluama yhteys on. Mainostusviesti saattaa myös sisältää rajoitetun määrän laitteen määrittämää tietoa, kuten sensorin arvoja.
- **Scanner.** Skannaus-tilassa laite kuuntelee oheislaitteiden lähettämiä mainostuspaketteja. Vain keskuslaitteet voivat olla skannaus tilassa. Jos skannaava keskuslaite vastaanottaa mainostuspaketin, se lähettää pyynnön mainostavalle oheislaitteelle, joka vastaa skannaus vahvistuksella. Kun keskuslaite vastaanottaa oheislaitteen skannaus vahvistuksen, voi tiedonsiirto laitteiden välillä alkaa.

- **Initiator.** Aloittaja on hyvin lyhykestoinen tila, jossa laite haluaa yhdistää tiettyyn oheislaitteeseen. Aloittajatilassa oleva laite odottaa saapuvaa mainospakettia, joka sisältää halutun oheislaitteen osoitteen. Vain keskuslaite voi olla aloittajatilassa.
- **Slave/Master.** Kun laitteet muodostavat yhteyden onnistuneesti, asetetaan mainostava laite orjaksi ja skannaava laite isännäksi. (9.)



KUVA 1. GAP-tilakaavio (9)

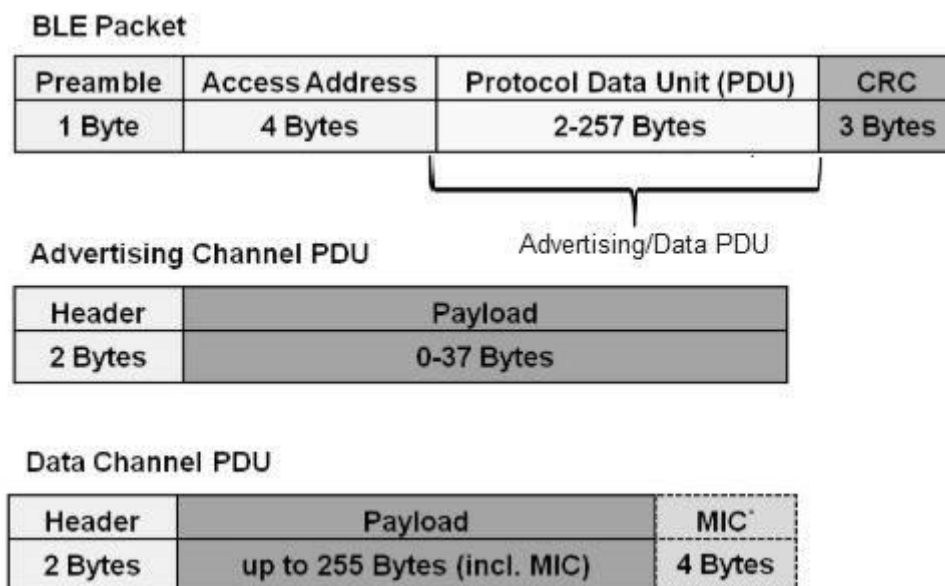
2.2.2 Bluetooth-mainostus

Bluetooth LE:ssä tiedonsiirto alkaa aina oheislaitteen suorittamalla mainostuksella. Oheislaite saa valita haluamansa mainostustavan, riippuen millaisiin laitteisiin se haluaa yhdistää. Mainostustavat ovat seuraavat (10):

- Yleinen mainostus (General Advertising) on mainostustapa, jossa oheislaite tekee liittymispyynnön kaikille lähistöllä oleville keskuslaitteille.
- Suunnattu mainostus (Directed Advertising) on tarkoitettu laitteille, jotka tietävät, minkä keskuslaitteen kanssa liittyminen halutaan suorittaa. Suunnattussa mainonnassa keskuslaite toimii aloittaja GAP roolissa, jotta laitteiden liittyminen olisi mahdollisimman nopeaa.

- Ei-yhdistettävä mainonta (Non-connectable Advertising) on mainostustapa, jossa oheislaite ei odota vastausta. Yleensä tätä mainostustapaa käyttävät lähettäjän roolissa olevat laitteet. Tämän takia lähettäjälaite ei välttämättä tarvitse edes Bluetooth-vastaanotinta.
- Löydettävä mainostus (Discoverable Advertising) on mainostustapa, jossa oheislaitteeseen ei voida yhdistää, mutta laite silti vastaanottaa keskuslaitteiden skannauspyyntöjä. Tätä mainontatyyppiä tekevät laitteet tarvitsevat sekä lähettimen että vastaanottimen.

Kaikki mainostuspaketit sisältävät lähettävän laitteen osoitteen, alustuksen (preamble), PDU:n (Protocol Data Unit), sekä CRC:n (Cyclic Redundancy Check). Esipuheessa olevaa arvoa käytetään signaalien ajoitukseen ja synkronointiin. PDU määrittää mainostuksen tyyppin ja sisältää paketissa olevan lastin. CRC on paketin tarkisteavain, jota käytetään virheellisen paketin tunnistamiseen. (11; 12; 13.) (Kuva 2.)



KUVA 2. Mainostuspaketin sisältö

2.3 Android-käyttöjärjestelmä

Android on tällä hetkellä maailman yleisin mobiilikäyttöjärjestelmä. Android on suunniteltu puhelimille, tablettitietokoneille ja älytelevisioille toimivaksi ohjelmistopinoksi, joka sisältää käyttöjärjestelmän, väliohjelmistoja, sekä käyttäjän perusohjelmia. Android pohjautuu Linux ytimeen, ja sen kehityksestä vastaavat Google ja Open Handset Alliance. (14.)

2.3.1 Android-sovellukset

Android-käyttöjärjestelmässä ajettavat ohjelmistot ovat Android-sovelluksia. Sovelluksia levitetään erillisessä APK-tiedostossa. APK-tiedosto on sovelluksen asennustiedosto, joka pakataan ZIP-muotoon ja ladataan yleensä suoraan laitteelle. (15.)

Android sovellusten jakelu on keskitetty erillisille Android kaappasovelluksille, kuten esimerkiksi useimpien Android-laitteiden mukana tulevalle Google Play-kaupalle.

2.3.2 Sovelluskehitys Androidissa

Toisin kuin yleensä, Android-sovelluksissa ohjelman luku aloitetaan aktiviteetiltä (Activity), eikä main()-metodilta. Aktiviteetit ovat yksi Android-ohjelman tärkeimmistä komponenteista, sillä aktiviteetit vastaavat käyttöliittymän ikkunoinnista, joka vastaa käyttäjälle näkyvästä toiminnallisesta kokonaisuudesta. Jotta aktiviteettia voidaan käyttää, täytyy se ensin lisätä ohjelman manifest-tiedostoon. Aktiviteetti luodaan tekemällä Activity-luokan aliluokka, joka sisällyttää onCreate()-metodin. Tämä metodi vastaanottaa käyttöjärjestelmän callbackin, kun aktiviteetti avataan. Käytössä oleva aktiviteetti vastaanottaa siihen rakennetuilla callback-funktioilla eri viestejä, riippuen sen elinkaaren tilasta. (16.)

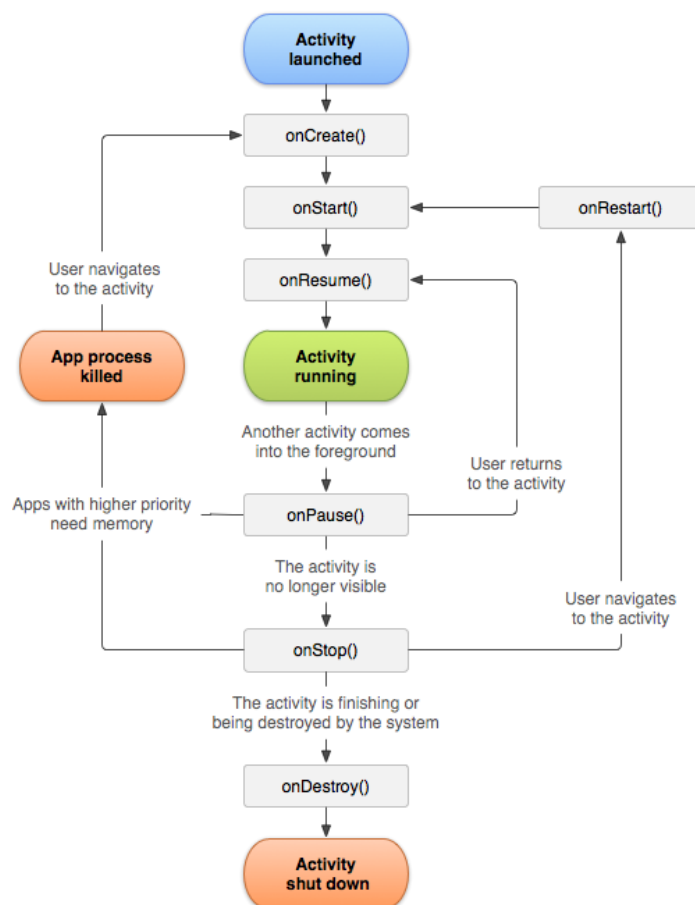
Manifesti

Kaikissa Android ohjelmissa on AndroidManifest.xml-tiedosto. Manifesti on ressurssitiedosto, johon kirjataan kaikki ohjelman vaatimat luvat, aktiviteetit, palve-

lut, vastaanottimet, sekä sisällöntarjoajat. Manifestissa määritetään myös sovelluksen käynnistysaktiiviteetti, joka avataan heti sovelluksen käynnistyessä. Manifesti sijaitsee projektin juuressa. (17.)

Aktiiviteettien elinkaari

Sovelluksen aktiiviteeteilla on kaikilla oma elinkaari. Elinkaari määrittää ohjelman ja sen aktiiviteettien tilan. Tila vaihtuu, kun Androidissa liikutaan toisten aktiiviteettien, tai sovellusten välillä. Sovelluksen aktiiviteetteihin tulee ilmoitus callback-funktiolla elinkaaren tilan muuttuessa. (18.) (Kuva 3.)



KUVA 3. Aktiiviteetin elinkaari (19)

Aktiiviteetilla on seuraavat elinkaarimetodit:

- **onCreate()**, kutsutaan aktiiviteetin käynnistyessä. Tässä metodissa aktiiviteetille yleensä asetetaan haluttu layout käyttämällä setContentView()-metodia.

- **onStart()**, kutsutaan, kun laite on ladannut näkymään asetettavan layoutin ja aktiviteetti siirretään etualalle.
- **onResume()**, kutsutaan, kun käyttöliittymä on valmis vuorovaikuttamaan käyttäjän kanssa.
- **onPause()**, kutsutaan, kun käyttäjä on lähdössä aktiviteetiltä eikä se ole enää etualalla.
- **onStop()**, jossa aktiviteetti ei ole enää näkyvässä. Aktiviteetti menee tähän tilaan, jos toinen sovellus otetaan etualalle tai jos sovellusta suljetaan.
- **onDestroy()**, joka kutsutaan ennen sovelluksen lopetusta tai jos aktiviteetti lopetetaan väliaikaisesti näkymän vaihtuessa vaak- tai pystyasentoon.
- **onRestart()**, kutsutaan, kun sovellus siirtyy onStop()-tilasta onStart()-tilaan. Tätä tilaa käytetään, kun sovellus on ollut taustalla ja käyttäjä valitsee sen takaisin etualalle. (19.)

3 MONIDROP-LAITE INFUUSIOTIEDON SEURANTAAN

Monidrop-tippalaskuri on oululaisen terveysteknologia-alan yritys Monidorin suunnittelema pienikokoinen laite, jota käytetään sairaaloissa potilaan suonensisäisen hoidon monitorointiin. Laite asetetaan nesteensiirtolaitteessa olevan tippakammion ympärille, jossa se seuraa tiputuksen etenemistä optisella sensorilla. Monidrop laskee sensorista saadun tiedon perusteella tiputusnopeutta, määrää ja kulunutta aikaa. Laite näyttää käyttäjälle tiedot suurikokoisesta kosketusnäytöstä, josta sen hälytysrajoja ja asetuksia voidaan muuttaa. Laite sisältää lisäksi kaiuttimen, kiihtyvyyssanturin sekä Bluetooth Low Energyä käyttävän lähettimen. (Kuva 4.)



KUVA 4. Monidrop

Käynnistyessään Monidrop muuttuu oheislaitteeksi, joka lähettää Bluetooth LE - mainostuspaketteja käyttäen mainostaessaan löydettävää mainostustyyppiä (Discoverable Advertising). Paketti sisältää 16-bittisen otsikon ja PDU:n, jossa on 24 tavun kokoinen tietolasti. Lasti sisältää kaikki Monidropin lähettämät mittaus-tiedot. Paketti sisältää lisäksi mainostavan laitteen osoitteen, tietoja lähetyksen ominaisuuksista sekä Monidorin rekisteröimän Bluetooth SIG -tunnistimen.

4 MOBIILISOVELLUS INFUUSIOTIEDON VASTAANOTTAMISEEN

Mobiilisovellus Monidrop-seurantalaitteen lähettämän tiedon vastaanottamiseen toteutettiin käyttämällä Android Studio -kehitysympäristöä sekä Java-ohjelmointikieltä. Testauksessa käytettiin Samsung J3- ja Samsung J4 -mobiililaitteita sekä Monidrop-tippalaskuria.

Työn tavoitteena on luoda Android-mobiilisovellus, joka vastaanottaa Monidrop-tippalaskurin lähettämää Bluetooth LE -mainostustietoa, ja esittää sen käyttäjälle yhdessä näkymässä. Näkymässä esitetään tärkeimmät infuusiohoidon seurantaan tarvittavat tiedot, kuten tiputusnopeus, määrä, aika sekä mahdolliset hälytysviestit. Lisäksi sovelluksen täytyy pystyä uudelleenlähettämään vastaanotettua mittaustietoa eteenpäin, jotta useasta mobiililaitteesta voidaan muodostaa koko sairaalaosaston kattava ad hoc -verkko.

4.1 Bluetooth mainostustiedon vastaanotto ja purku

Monidrop-laite lähettää Bluetooth LE -teknologiaa käyttäen mainostuspaketin, joka pitää sisällään koko laitteen mittaaman infuusiotiedon. Android-pohjaiseen mobiililaitteeseen toteutetaan BLE-skanneri, joka vastaanottaa oheislaitteiden lähettämää mainostusta. Mobiilisovelluksessa käytetään Androidin BLE APIa, jotta voidaan kuunnella alueella tapahtuvaa mainostusta.

4.1.1 Lupien asettaminen

Jotta Android-mobiililaitteella voi vastaanottaa Bluetooth-mainostustietoa, täytyy sovellukselle ensin määrittää lupa käyttää ja löytää Bluetooth-laitteita. Luvat lisätään projektin AndroidManifest.xml -tiedostoon, jolloin ne hyväksytään joko automaattisesti sovelluksen käynnistyessä, tai käyttäjän suostumuksesta. Projektin manifestiin lisätään luvat BLUETOOTH, BLUETOOTH_ADMIN, ACCESS_FINE_LOCATION ja ACCESS_COARSE_LOCATION. Näillä luvilla sovellus voi käyttää laitteen Bluetooth-vastaanotinta, säätää sen asetuksia sekä lähettää ja vastaanottaa Bluetooth-mainostuspaketteja. (21.)

4.1.2 Skannerin rakentaminen

Jotta pystytään vastaanottamaan mainostuspaketteja, täytyy käyttää Android BLE API:ssa olevaa BluetoothLeScanner-luokkaa. Luokalla rakennetaan muokattu skanneri antamalla sen startScan()-metodille parametreina ScanFilter-suodatin, ScanSettings-vastaanottoasetukset, sekä ScanCallback-vastakutsu. (Kuva 6.)

```
final ScanSettings scanSettings = new ScanSettings.Builder()
    .setScanMode(ScanSettings.SCAN_MODE_LOW_LATENCY)
    .setReportDelay(0)
    .setMatchMode(ScanSettings.MATCH_MODE_AGGRESSIVE)
    .build();

ScanFilter.Builder builder = new ScanFilter.Builder();
final Vector<ScanFilter> filter = new Vector<>();
filter.add(builder.build());

if (mBluetoothLeScanner != null && bluetoothAdapter.isEnabled() &&
    bluetoothAdapter.getState() == BluetoothAdapter.STATE_ON) {
    mBluetoothLeScanner.startScan(filter, scanSettings, leScanCallback);
    Logger.d(TAG, format: "SUCCESS: Scan initialized");
}
else {
    mBluetoothLeScanner = bluetoothAdapter.getBluetoothLeScanner();
    Logger.e(TAG, format: "Scanner could not be initialized");
}
```

KUVA 6. Bluetooth-skanneri, jolle annetaan parametreinä suodatin, vastaanottoasetukset sekä callback

ScanSettings-oliolle asetetaan nopein mahdollinen vastausnopeus sekä herkin laukaisuraja, jotta saadaan maksimoitua signaalin kantama. ScanFilter-suodattimelle ei anneta asetuksia, jotta voidaan vastaanottaa kaiken mittaustiedon riippumatta siitä, onko se peräisin Monidrop-tippalaskurista vai toisesta mobiililaitteesta.

Lopuksi lisätään ScanCallback-vastaanotin, johon Android lähettää edellä määritetyllä tavalla havaitut Bluetooth-paketit. Callbackin sisällä tarkistetaan, että vastaanotetun paketin valmistajan id vastaa Monidorin käyttämää. Tässä työssä käytetään Monidorin virallista Bluetooth SIG -järjestöön rekisteröimää tunnistinta, joka on 1449. Lopuksi vastaanotettu mittaustietopaketti asetetaan muuttujaan, josta erillisellä säikeellä oleva tiedonpurkaja-luokka osaa sen käydä hakemassa threadNotify()-metodikutsun jälkeen. (Kuva 7.)

```

private ScanCallback leScanCallback = new ScanCallback() {
    @Override
    public void onScanResult(int callbackType, ScanResult result) {
        ScanRecord record = result.getScanRecord();
        assert record != null;
        byte[] array = record.getManufacturerSpecificData( manufacturerId: 1449);
        if (array != null) {
            byteArray = array;
            threadNotify();
        }
    }
};

```

KUVA 7. Mainostuspaketin vastaanotin

4.1.3 Mittaustiedon purku

Vastaanotettu Bluetooth-mainostuspaketti sisältää 24 yksikköä pitkän tavujonon. Jokainen tavu sisältää kokonaisen tai osittaisen osan Monidrop-laitteen mittamaa tietoa. Koska osa mittaustiedosta vaatii enemmän kuin yhden tavun verran tilaa, on niille varattu useampi tavu jonosta. Jono sisältää seuraavia tietoja:

- laitetyyppi
- protokolla versio
- paketin tunnistin (2 tavua)
- laitteen tila
- akun tila
- hälytystila (2 tavua)
- tippanopeus (2 tavua)
- tiputusmäärä (2 tavua)
- hoidon kesto (2 tavua)
- tippatavoite (2 tavua)
- tavoitemäärä (2 tavua)
- tavoitekesto (2 tavua)
- sarjanumero (3 tavua).

Kun Bluetooth-mainostuspaketti vastaanotetaan, siitä lähetetään tieto erilliselle säikeelle, joka on vastuussa tiedon purkamisesta ja palautuksesta käyttöliittymälle. Purkuprosessi on aikaa vievää, jonka takia sille oli järkevää omistaa oma säie käyttöliittymän sulavuuden varmistamiseksi.

Purkuprosessi alkaa tunnistamalla vastaanotetun tiedon alkuperä. Ensimmäiseksi puretaan kolmetavuinen sarjanumero, josta tiedetään mille Monidrop-laitteelle mittaustieto kuuluu. Tämän jälkeen puretaan paketin tunnistin. Paketin tunnistin kertoo, kuinka tuore vastaanotettu paketti on. Jos tunnistin on korkeampi kuin aikaisemmissa vastaanotetuissa lähetyksissä, on kyseessä tuore paketti, joka puretaan loppuun ja näytetään käyttöliittymässä. Mikäli tunnistin on matalampi kuin saman laitteen aikaisemmin lähettämä tunnistin, purkuprosessi loppuu ja paketti hylätään.

4.1.4 Muunnos bittioperaattoreilla

Vastaanotetun Bluetooth-datapaketin arvot voivat olla maksimissaan yhden tavun kokoisia. Tämän takia osa enemmän tilaa vievistä mittaustiedoista varaa useamman tavun jonosta. Koska vastaanotetun paketin tavut ovat etumerkillisiä, on niiden arvo väliltä $-128-127$. Tämän takia niitä ei voida suoraan muuntaa kokonaisluvuksi. Jotta muunnoksesta saadaan oikea arvo, on haluttuun mittaustietoon käytettävä bittioperaattoreita. Ensin tavut siirretään oikeille paikoilleen käyttäen siirto vasemmalle -operaattoria. Tämän jälkeen jokainen tavu suorittaa & (AND)-operaation itselleen, jossa kaikki tavun ympärillä olevat bitit maskataan 0-tilaan. Lopullinen muunnos saadaan, kun kaikki saadut arvot yhdistetään käyttäen | (OR)-operaattoria. (Kuva 8)

```

public static int bytesToInteger(byte value) {
    return (value & 0xff);
}

public static int bytesToInteger(byte value1, byte value2) {
    return ((value1 << 8) & 0xff00) | (value2 & 0x00ff);
}

public static int bytesToInteger(byte value1, byte value2, byte value3) {
    return ((value1 << 16) & 0xff0000) | ((value2 << 8) & 0x00ff00) | (value3 & 0x0000ff);
}

```

KUVA 8. Tiedon muunnos

4.2 Ad hoc -verkon ja toistimen rakentaminen

Useasta mobiililaitteesta koostuvan ad hoc -verkon luonti lisättiin myöhemmin osaksi projektin vaatimuksia. Testatessa selvisi, että Monidrop-laitteen lähettämä mainostus ei kuulu tarpeeksi kauas, mikä johtuu sairaalaosastojen paksuista seinistä ja pitkistä etäisyyksistä. Tämän takia päädyttiin ratkaisuun, jossa Android-mobiililaitteita sijoitetaan tasaisin välimatkoin pitkin sairaalaosastoa. Laitteet vastaanottavat Monidropista peräisin olevaa mainostusdataa ja lähettävät sitä eteenpäin toisille mobiililaitteille. Näin saadaan muodostettua koko osaston kattava verkko, jonka avulla hoitajat tietävät potilaan hoidon etenemisen poistumatta työpisteeltään.

Ohjelmaan toteutettiin mainostosoperaatio, joka suoritetaan omassa säikeessään. Mainostuksen aloitukseen tarvitaan BluetoothLeAdvertiserin startAdvertising()-metodi, jolle annetaan parametreina AdvertiseSettings, AdvertiseData ja AdvertiseCallback. BluetoothLeAdvertiser saadaan muodostettua hakemalla se BluetoothAdapter.getBluetoothLeAdvertiser()-metodilla. AdvertiseSettings luokassa asetetaan mainostuksen teho ja taajuus maksimiin käyttämällä ADVERTISE_MODE_LOW_LATENCY- ja ADVERTISE_TX_POWER_HIGH-asetuksia. AdvertiseData-oliossa määritetään lähtevän mainostuspaketin sisältö. Tässä tapauksessa sisältö asetetaan identtiseksi vastaanotetun paketin kanssa. Lopuksi toteutetaan AdvertiseCallback, joka palauttaa mainostuksen tilan metodeille onStartSuccess() ja onStartFailure().

4.3 Käyttöliittymä

Käyttöliittymässä on tarkoitus näyttää käyttäjälle lista kaikista löydetyistä Mo-nidrop laitteista, sekä tärkeimmät mittaustiedot. Tarkoituksena on tarjota käyttäjälle kattava kokonaiskuva yhdeltä ruudulta. Käyttöliittymä sisältää listan laitteista, jossa jokaisella laitteella näytetään tietoja, kuten tippanopeus, kokonaisuus, tavoitearvot, mahdolliset hälytykset, sekä laitteen akun varaus.

4.3.1 Listanäkymän rakentaminen

Koska tavoitteena on luoda löydetyistä laitteista lista, oli järkevää käyttää ListView-komponenttia. ListView näyttää kaikki laitteet vertikaalisena vieritettävänä listana, jotka sen adapteriin lisätään. Adapterissa määritetään kaikkien listan laitteiden näkymä getView()-metodissa. Kun laitteita lisätään tai muokataan, voidaan käyttää adapterin add()- ja notifyDataSetChanged()-metodeja. (Kuva 9.)

F21A30010247	100%	1250ml/h	38ml	1h 16min
(300ml/h)		(250ml)		(1h 30min)
Infusion rate to high!				
F21A30010314	100%	145ml/h	16ml	32min
(150ml/h)		(100ml)		(1h 28min)

KUVA 9. Käyttöliittymässä näkyvät laitteet

4.3.2 Tiedonsiirto käyttöliittymälle

Bluetooth-mainostuspaketin saapuessa se lähetetään itsetekooselle Bluetooth-DataParserThread-oliolle. Olion tehtävänä on purkaa vastaanotettu tieto omassa säikeessään. Kun purkuprosessi on valmis, BluetoothDataParserThread rakentaa niistä laitteen tiedot sisältävän Device-olion, joka lähetetään takaisin käyttö-

liittymää hoitavalle pääsäikeelle. Säikeen vaihto tapahtuu kutsumalla pääaktiiviteetissa sijaitsevaa metodia, joka vaihtaa koodin ajamisen pääsäikeelle käyttäen Activity-luokan `runOnUiThread()`-metodia. (22.)

5 POHDINTA

Työn tuloksena saatiin valmiiksi Android-mobiilisovellus, joka vastaanottaa Bluetooth Low Energy -spesifikaatiota käyttävän Monidrop-laitteen mittaustietoja mainostuspaketeista. Lisäksi sovellus kykenee uudelleenlähettämään vastaanotettuja mittaustietoja toisille laitteille, minkä avulla useasta mobiililaitteesta voidaan rakentaa koko sairaalaosaston kattava verkko. Lähetyksen ja vastaanoton lisäksi sovellus pystyy esittämään käyttöliittymällä kaikkien löydettyjen Monidrop-laitteiden lähettämät tiedot.

Tarkastellessa lopputulosta voidaan sanoa, että työn alussa laaditut tavoitteet täytettiin. Onnistunut toteutus oli myös toimeksiantajan kannalta tärkeä, sillä Monidor pääsi aloittamaan heti ensimmäisen toimivan version jälkeen testauksen ja koekäytöt asiakkailla.

Kaikilta ongelmilta projektissa ei kuitenkaan vältytty. Ensimmäisissä versioissa käyttöliittymä alkoi takkuilemaan, jos käynnissä olevia Monidrop-laitteita oli lähitöllä useita. Tähän auttoi kuitenkin korjauksena tehty Bluetooth-mittaustiedon purku omalla säikeellään sekä purkuprosessin optimointi. Lisäongelmia aiheutti Androidin tapa sulkea Bluetooth-mainostuspakettien skannaus, jos vastaanotto- ja lähetyksen prosessi olivat yhtä aikaa käytössä yli puoli tuntia. Tähänkin keksittiin lopulta ratkaisu, jossa mobiililaitteen Bluetoothia käytettiin automaattisesti puolen tunnin välein pois päältä.

Toteutuksen aikana sain korvaamattoman määrän kokemusta Android-ohjelmien kehityksestä, Java-ohjelmoinnista sekä Bluetooth Low Energy -tiedonsiirrosta. Android Studio oli erinomainen valinta kehitysympäristöksi, sillä sen käyttöliittymän rakennustyökalu sekä Profiler- suorituskyvyn monitorointityökalu säästivät monia tunteja kehitysaikaa.

Kaiken kaikkiaan voidaan työtä pitää onnistuneena. Haluttuihin tavoitteisiin päästiin ja ilmenneet ongelmat ratkaistiin. Projektiin on työn suorituksen jälkeen tullut useita lisäominaisuuksia, kuten mahdollisuus näyttää käyttöliittymällä vain valitut laitteet, laitteiden nimeäminen, laitteeseen asetetun nimen lähetyksen muille alueella oleville laitteille osana mainostusta, vastaanotetun mittaustiedon lähetyksen Wi-Fi-

yhteyden kautta sekä Monidrop-laitteiden tietojen tallentaminen ja haku puhelimen muistista. Ohjelmaa on sittemmin käytetty useissa kymmenissä eri sairaal-osastoissa eri puolella Suomea, joista työn tilaaja on saanut paljon arvokasta palautetta infuusioidon etämonitoroinnin potentiaalista. Samalla työ pohjusti Monidorin tuotekehitystä tulevien tuotteiden osalta, joissa etämonitorointi on keskeisessä roolissa.

LÄHTEET

1. Monidor. Saatavissa: <http://monidor.com/>. Hakupäivä 29.09.2020.
2. Bluetooth. 2016. ScienceDirect. Saatavissa: <https://www.sciencedirect.com/topics/engineering/bluetooth/>. Hakupäivä 28.04.2020.
3. Miller, Brent 2000. Bluetooth Technology and SIG Overview. Pearson Education. Informit. Saatavissa: <https://www.informit.com/articles/article.aspx?p=19735&seqNum=4/>. Hakupäivä 01.05.2020.
4. Woolley, Martin 2020. Bluetooth Core Specification Version 5.2 Feature Overview. Bluetooth SIG. Saatavissa: https://www.bluetooth.com/wp-content/uploads/2020/01/Bluetooth_5.2_Feature_Overview.pdf. Hakupäivä 01.05.2020.
5. Burgess, Matt 2020. What is the Internet of Things? Wired. Saatavissa: <https://www.wired.co.uk/article/internet-of-things-what-is-explained-iot/>. Hakupäivä 01.05.2020.
6. Kamath, Sandeep - Lindh, Joakim 2019. Measuring Bluetooth Low Energy Power Consumption. Texas Instruments. Saatavissa: <https://www.ti.com/lit/an/swra347a/swra347a.pdf?ts=1589460592015/>. Hakupäivä 01.05.2020.
7. Argenox Technologies LLC 2020. BLE Advertising Primer. Saatavilla: <https://www.argenox.com/library/bluetooth-low-energy/ble-advertising-primer/>. Hakupäivä 07.04.2020.
8. Townsend, Kevin 2014. Introduction to Bluetooth Low Energy. Adafruit Industries. Saatavissa: <https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gap/>. Hakupäivä 07.08.2019.
9. Texas Instruments 2018. Generic Access Profile. Saatavilla: http://software-dl.ti.com/lprf/simplelink_cc26x2_sdk-1.60/docs/ble5stack/ble_user_guide/html/ble-stack/. Hakupäivä 29.04.2020.

10. Woolley, Martin 2016. Types of Advertising. Bluetooth SIG. Saatavilla: <https://www.bluetooth.com/blog/advertising-works-part-1/>. Hakupäivä 05.05.2020.
11. Katsandres, Jim 2017. Bluetooth Low Energy – It Starts with Advertising. Bluetooth SIG Saatavilla: <https://www.bluetooth.com/blog/bluetooth-low-energy-it-starts-with-advertising/>. Hakupäivä 20.04.2020.
12. Argenox Technologies LLC 2016. BLE Advertising Primer. Saatavilla: <https://www.argenox.com/library/bluetooth-low-energy/ble-advertising-primer/>. Hakupäivä 20.04.2020.
13. Molkenthin, Bastian 2019. Understanding CRC. sunshine2k. Saatavilla: http://www.sunshine2k.de/articles/coding/crc/understanding_crc.html. Hakupäivä 20.04.2020.
14. Karch, Mariah 2019. What is Google Android? Lifewire. Saatavissa: <https://www.lifewire.com/what-is-google-android-1616887/>. Hakupäivä 05.05.2020.
15. Fisher, Tim 2020. What Is an APK File? Lifewire. Saatavissa: <https://www.lifewire.com/apk-file-4152929/>. Hakupäivä 05.05.2020.
16. Open Handset Alliance 2020. Introduction to Activities. Saatavissa: <https://developer.android.com/guide/components/activities/intro-activities/>. Hakupäivä 05.05.2020.
17. Open Handset Alliance 2020. App Manifest Overview. Saatavissa: <https://developer.android.com/guide/topics/manifest/manifest-intro/>. Hakupäivä 05.05.2020.
18. Silva, Sergio 2018. What Is the Android Activity Lifecycle? Nearsoft. Saatavissa: <https://nearsoft.com/blog/what-is-the-android-activity-lifecycle/> Hakupäivä 15.09.2020.

19. Understand the Activity Lifecycle. 2020. Android Developers. Saatavissa: <https://developer.android.com/guide/components/activities/activity-lifecycle/>. Hakupäivä 05.05.2020.
20. Bluetooth SIG 2020. Monitor Bluetooth SIG. Saatavissa: <https://www.bluetooth.com/develop-with-bluetooth/join/member-directory/>. Hakupäivä 10.05.2020.
21. PlotProjects 2020. Dangerous Permissions on Android Marshmallow. Saatavissa: <https://www.plotprojects.com/blog/dangerous-permissions-on-android-marshmallow/>. Hakupäivä 10.05.2020.
22. Goshen, Eliran 2017. Android Threading: All You Need to Know. Saatavissa: <https://www.toptal.com/android/android-threading-all-you-need-to-know/>. Hakupäivä 07.09.2019.