

# MONITAAJUUSKAIKULUOTAIMEN KÄYTTÖÖNOTTO

Korpimaa Heikki

Opinnäytetyö

Tieto- ja viestintätekniikka  
Insinööri (AMK)

2022

Tieto- ja viestintätekniikka  
Insinööri (AMK)

---

|                       |                                        |              |      |
|-----------------------|----------------------------------------|--------------|------|
| <b>Tekijä</b>         | Heikki Korpimaa                        | <b>Vuosi</b> | 2022 |
| <b>Ohjaaja</b>        | Tommi Kokko                            |              |      |
| <b>Toimeksiantaja</b> | Mitta Oy                               |              |      |
| <b>Työn nimi</b>      | Monitaajuuskaikuluotaimen käyttöönotto |              |      |
| <b>Sivumäärä</b>      | 21                                     |              |      |

---

Opinnäytetyön tarkoituksena oli tehdä Mitta Oy:llä käyttämättömänä olevaan monitaajuuskaikuluotaimeen ohjauslaitteisto. Opinnäytetyön tavoitteena oli saada vesistöjen syvyyden mittaamiseen toimiva järjestelmä, josta saatu data voidaan kätevämmiin saada jatkokäsittelyyn.

Työssä toteutettiin Java-ohjelmointikielellä Android-laitteelle ohjelmisto, jolla voidaan etänä hallinnoida monitaajuuskaikuluotainta. Sovellus yhdistyy Raspberry Pi -tietokoneeseen käyttäen Bluetooth-yhteyttä. Raspberry Pi -tietokoneelle kehitettiin Python-ohjelmointikielellä ohjausohjelmat, jotka ajetaan Android-ohjelmasta. Monitaajuuskaikuluotain yhdistetään Raspberry Pi -tietokoneeseen USB-kaapelilla. Kaikuluotaimesta lähetetään mittausdata Android-laitteeseen.

Opinnäytetyö tehtiin toiminnallisena projektina. Työssä tutustuttiin, miten saadaan eri laitteet kommunikoimaan keskenään käyttäen langattomia ja langallisia menetelmiä sekä miten käsitellään dataa tiedostoon. Ohjausohjelman teko jäi puuttumaan järjestelmästä, mutta alustava laitteisto on valmiina jatkokehitystä varten. Työtä varten kehitetty ohjelmisto on todettu toimivaksi, mutta Mitta Oy koki projektissa käytetyn Raspberry Pi -tietokoneen liian kömpelöksi käytettäväksi käytännössä.

Study Programme in Information  
and Communication Technology  
Bachelor of Engineering

|                        |                                 |             |      |
|------------------------|---------------------------------|-------------|------|
| <b>Author</b>          | Heikki Korpimaa                 | <b>Year</b> | 2022 |
| <b>Supervisor(s)</b>   | Tommi Kokko                     |             |      |
| <b>Commissioned by</b> | Mitta Oy                        |             |      |
| <b>Title</b>           | Deployment of a Multiband Sonar |             |      |
| <b>Number of pages</b> | 21                              |             |      |

The purpose of this thesis study was to create a control system for an unused multiband sonar at Mitta Oy. The aim of the thesis was to create a functioning system for measuring the depth of water bodies. The data was obtained from the sonar in a more convenient manner to be used for further processing than with previously used methods.

The software was developed for an Android device using Java programming language, so it can be used to remotely control the multiband sonar. The application connects to Raspberry Pi computer using a Bluetooth connection. Control programs were developed for the Raspberry Pi computer using Python programming language, which can be run from the Android application. The multiband sonar was connected to the Raspberry Pi computer with an USB cable. The measurement data is sent from the multiband sonar to the Android application. The thesis was done as a functional project. During the thesis making different devices communicate using wireless and wired methods, and how to process data into a file were studied.

The creation of the control program was left undone from the system during the project, but the initial hardware is ready for further development. The software developed for the thesis has been deemed to be functional, but Mitta Oy felt that the Raspberry Pi computer used in the project was too cumbersome to be used in practice.

Keywords

Android, Bluetooth, data transfer, Raspberry Pi

## SISÄLLYS

|                                                         |    |
|---------------------------------------------------------|----|
| 1 JOHDANTO .....                                        | 5  |
| 2 PROJEKTIN SUUNNITTELU.....                            | 6  |
| 3 KEHITYSALUSTAT JA LAITTEET .....                      | 7  |
| 3.1 Android-käyttöliittymä.....                         | 7  |
| 3.2 Android Studio .....                                | 7  |
| 3.3 Raspberry Pi .....                                  | 8  |
| 3.4 Bluetooth.....                                      | 9  |
| 3.5 Monitaajuuskaikuluotain.....                        | 9  |
| 4 OHJELMISTOJEN JA LAITTEIDEN ASENNUS JA ALUSTUS .....  | 11 |
| 4.1 Android Studio .....                                | 11 |
| 4.2 Raspberry Pi .....                                  | 12 |
| 5 JÄRJESTELMÄN TOTEUTUS.....                            | 13 |
| 5.1 Kytkenät.....                                       | 13 |
| 5.2 Android-ohjelma.....                                | 13 |
| 5.3 Bluetooth-yhteys .....                              | 16 |
| 5.3.1 Bluetooth-yhteys Android-laitteessa .....         | 16 |
| 5.3.2 Bluetooth-yhteys Raspberry Pi -tietokoneessa..... | 18 |
| 6 POHDINTA .....                                        | 20 |
| LÄHTEET.....                                            | 21 |

## 1 JOHDANTO

Opinnäytetyö sai alkunsa vuonna 2021 syksyllä Lapin ammattikorkeakoulussa olleesta pilottihankkeesta, jossa paikalliset yritykset tarjosivat IT-opiskelijoille pienimuotoisia hankkeita ammattitaidon harjoittamiseksi. Olin harjoittelussa Lapin ammattikorkeakoulun IoT-laboratoriossa, kun minuun otetiin yhteyttä, että Mitta Oy:n paikallinen osasto haluaisi monitaajuuskaikuluotaimelle käyttöliittymän prototyyppin Android-laitteelle. Mitta Oy on suomalainen kansainvälisesti toimiva mittaus- ja kenttätutkimusalan konsulttiyhtiö.

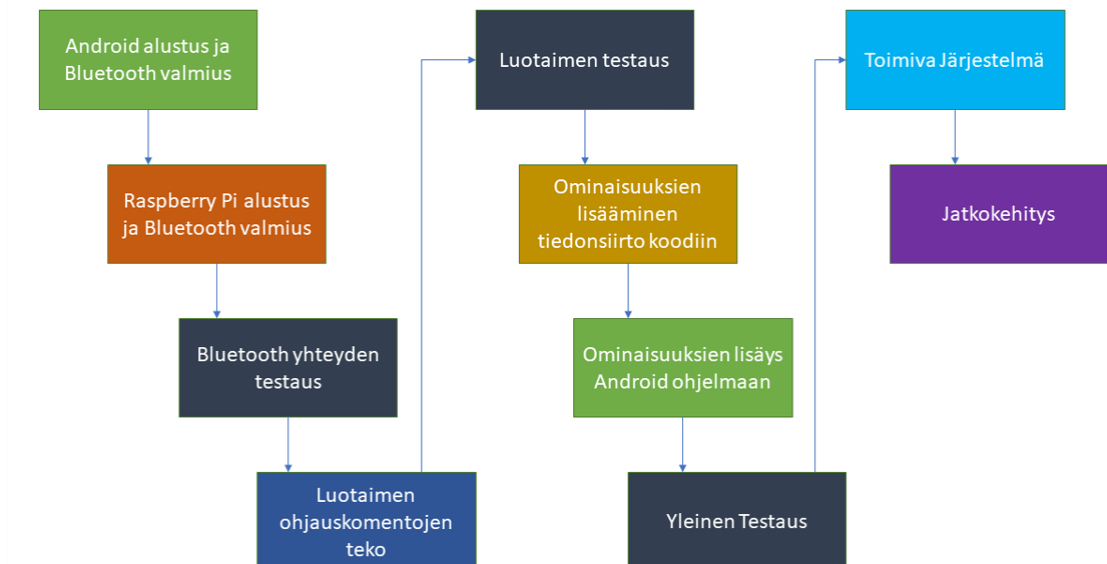
Olen aikaisemmin tehnyt Android-sovelluksen mallin ja totesin, että tämä voisi olla hyvä tapa harjoittaa omia Android-ohjelmointitaitoja. Vielä tässä kohdassa en arvellut, että voin jatkaa työn opinnäytetyöksi asti. Kun olin saanut käyttöliittymän tehtyä, todettiin Mitta Oy:n puolelta, että työ on alkanut hyvin ja heidän puoleltansa voisin tehdä siitä toimivan sovelluksen. Tämä tarkoittaa, että tehdään ohjauslaite, johon kaikuluotain kytketään. Yhdistetään ohjauslaite langattomasti Bluetooth-yhteydellä Android-laitteeseen, joka toimii kaukosäätimenä.

Opinnäytetyössä keskeisimpinä teknologioina ovat Linux-pohjaiset järjestelmät ja eri tiedonsiirron tekniikat. Android-laitteen ja Raspberry Pi -tietokoneen käyttöjärjestelmät pohjautuvat Linux-käyttöjärjestelmään. Työssä esitellään, miten luodaan Bluetooth-yhteys Android-laitteen ja Raspberry Pi -tietokoneen välille itse luotujen toiminnallisuuksien käyttämiseen ja miten voidaan Android-ohjelmasta kirjoittaa dataa muistiin.

Tarkoituksena on tuottaa järjestelmä, jolla voidaan mitata vesistöjen syvyyttä käyttäen korkea- ja matalataajuuskeilausta, joilla saadaan selville missä syvyydessä alkaa pehmeäpohja ja missä kovapohja. Päädyin käyttämään Raspberry Pi -tietokonetta ohjauslaitteena sen valmiiksi sisäänrakennettujen ominaisuuksien ja suoritustehon takia. Android-laitteesta halutaan pystyä sekä aloittamaan ja lopettamaan keilauksen, että antaa komennon siirtää mittausdata Raspberry Pi -tietokoneelta Android-laitteeseen. Valmiista järjestelmästä saadaan kätevästi mittausdata jatkokäsittelyyn, koska se on valmiiksi digitaalisessa muodossa.

## 2 PROJEKTIN SUUNNITTELU

Suunnittelin projektin käyttäen vesiputousmallia projektin kertaluonteisuuden takia ja että voin keskittyä yhteen osa-alueeseen kerrallaan. Seuraava kuvio 1 on apuna visualisoimaan suunnitelman selitystä. Aloitin asentamalla Android Studion ja tein TCP-protokollan tapaisen Handshake-ohjelman Android-sovellukseen. Sovellus lähettää Bluetooth-yhteyden kautta Raspberry Pi -tietokoneelle tervehdyksen, jonka vastaanotettuaan Raspberry Pi -tietokone tekee ilmoituksen, että on vastaanottanut tervehdyksen. Raspberry Pi -tietokone lähettää Android-sovellukseen vastatervehdyksen, jonka vastaanotosta sovellus vastaavasti tekee ilmoituksen.



Kuvio 1. Suunnitelma visualisoituna

Onnistuneen Bluetooth-yhteyden testauksen jälkeen tein Raspberry Pi -tietokoneelle luotaimen ohjausohjelman. Komennot luotaimelle ja sen lähettämä data kulkevat USB-yhteyden kautta, joten tein ohjelman, joka lähettää dataa USB-portin kautta ja kirjoittaa USB-porttiin tulevaa dataa tiedostoon.

Kun luotaimen ohjausohjelmien testaus onnistui, yhdistin sen Bluetooth-koodiin, jotta Android-ohjelman kautta voidaan ajaa luotaimen ohjausohjelmat sekä siirtää luotaimen mittausdataa Android-laitteeseen. Viimeistelynä testasin tavat, jotka voivat mahdollisesti rikkoa ohjelman toiminnallisuudet.

### 3 KEHITYSALUSTAT JA LAITTEET

#### 3.1 Android-käyttöliittymä

Android on älypuhelimille, tableteille ja muille vastaaville älylaitteille kehitetty käyttöliittymä, joka pohjautuu Linux-käyttöjärjestelmään. Sen kehitys alkoi vuonna 2003 käyttöjärjestelmäksi digitaalisille kameroille. Vuonna 2005 Google osti alkuperäisen yrityksen ja alkoi jatkokehittämään sitä mobiililaitteille. (Android Authority 2022.)

Vuonna 2008 julkaistiin ensimmäinen Android-älypuhelin, HTC Dream, joka tunnetaan paremmin nimellä T-Mobile G1. Puhelin ei saanut kovin hyviä arvosteluja, mutta se sisälsi sisäänrakennettuna Googlen palveluita, jotka ovat nykypäivänä oletuksena kaikissa Android-puhelimeissa. Merkittävimpinä ohjelmina olivat Google Maps, YouTube ja Android Market, joka tunnetaan nykypäivänä paremmin nimellä Google Play. (Android Authority 2022.)

#### 3.2 Android Studio

Android Studio on virallinen ohjelmointiympäristö Android-laitteille. Se sisältää työkalut ohjelmoinnin helpottamiseen, kuten debuggerin, jolla voidaan ajaa ohjelma askel kerrallaan ongelmien nappaamiseksi, sekä Android-laitteen virtualisoinnin, jolla voidaan nopeammin testata muutoksia ja toimivuutta. (Android Developers 2022a.)

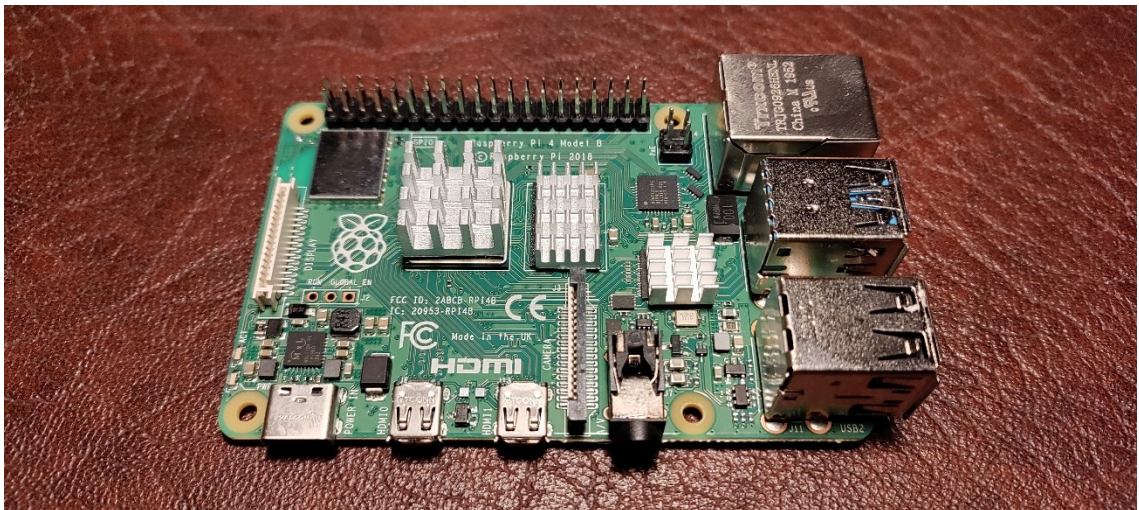
Android Studio pohjautuu JetBrains IntelliJ IDEA-ohjelmistoon, joka on suunniteltu pääasiallisesti sovellusten tekoon Android-laitteille ja sen voi asentaa Windows-, MacOS- ja Linux-käyttöjärjestelmille. Ohjelmointikielistä Android Studio tukee Javaa, C/C++ ja Kotlinia, joista Google suosittelee Kotlinia pääasiallisena ohjelmointikielenä. (Android Developers 2022b.)

Opinnäytetyössä käytetään Javaa ohjelmointikielenä, koska sitä on käytetty aikaisemmissa ohjelmointiprojekteissa ja -kursseissa. Java on Sun Microsystemsin kehittämä korkeantason oliopohjainen ohjelmointikieli, joka on suunniteltu toimimaan alustasta riippumatta. (Javatpoint 2022.)

### 3.3 Raspberry Pi

Raspberry Pi on isobritannialaisen Raspberry Pi Foundationin kehittämä yhden piirilevyn tietokone. Alun perin se on kehitetty tietojenkäsittelytieteiden perusteiden opiskelun edistämiseen, mutta se sai laajaa kannatusta pienielektroniikan ja tietokoneiden harrastelijoiden parissa sen matalan hinnan ja modulaarisuuden ansiosta. (Fromaget, P. 2022.)

Opinnäytetyössä käytetään kaikuluotaimen ohjaamiseen Raspberry Pi 4 Model B -yhden piirilevyn tietokonetta, josta on kuva kuviossa 2. Tietokoneessa on ARMv8-arkkitehtuurilla ja 1,5 gigahertsin kellotaajuudella oleva 64-bittinen nelilydinprosessori, jonka ansiosta laitteessa on tarpeeksi suoritustehoa. Laitteessa on myös Bluetooth 5.0 -tuki, jota käytetään kommunikaatioväylänä Android-laitteen kanssa. (Raspberry Pi Foundation 2022.)



Kuvio 2. Raspberry Pi 4 Model B

Raspberry Pi 4 Model B -tietokoneen voi yhdistää internetiin Ethernet-portin kautta, joka tukee gigabitin siirtonopeutta tai langattomasti WLAN-verkkoon, joka toimii 2,4:n tai 5,0 gigahertsin taajuudella. Laitteessa on kaksi Micro HDMI -porttia näyttöjä varten ja 3,5 millimetrin jakki kuulokkeita, kaiutinta tai komposiittivideota varten. Sisääntulovirraksi laitteelle suositellaan viiden voltin kolmen ampeerin tasavirtaa (5V3A DC). Tarvittaessa laitteessa voidaan käyttää myös käyttää 2,5 ampeeria, jos yhdistetyt USB-laitteet käyttävät yhteensä vähemmän kuin 500 milliampeeria. (Raspberry Pi Foundation 2022.)

Opinnäytetyössä olevassa Raspberry Pi -tietokoneessa on 4 gigabittiä käyttömuistia, mutta laitteen saa myös 1:n, 2:n ja 8 gigabitin käyttömuistilla. Kaksi USB 2.0 ja USB 3.0 -porttia sekä 40 kappaletta GPIO-pinnejä mahdollistavat useamman eri tavan liittää laitteita Raspberry Pi -tietokoneeseen. Siinä on paikka yhdelle MicroSD-muistikortille, jolle asennetaan käyttöjärjestelmä, ja joka toimii massamuistina. (Raspberry Pi Foundation 2022.)

### 3.4 Bluetooth

Bluetooth on langattoman tiedonsiirron standardi, joka lähettää dataa keskiarvoisesti 2,45 gigahertsin taajuudella. Koska samalla taajuusalueella toimivat esimerkiksi mikroaaltouunit ja langattomat paikallisverkot, käyttää Bluetooth taajuushyppelyhajaspektritekniikkaa. (Bluetooth 2022a.)

Bluetooth-tiedonsiirron kantama riippuu antennista, lähetystehosta, vastaanottimen herkkyydestä sekä signaaliin vaikuttavista vaimentavista asioista, kuten sade, seinät ja pinnat, jotka heijastavat radioaaltoja. Käyttäen pienempää taajuutta lähetykseen saadaan suurempi toimintasäde, mutta tiedonsiirtonopeus on hitaampi verrattuna korkeampiin taajuuksiin. (Bluetooth 2022b.)

### 3.5 Monitaajuuskaikuluotain

Opinnäytetyössä käytettävä kaikuluotain on Airmar Dual Frequency EchoRange Smart Sensor M195, joka lähettää ja vastaanottaa NMEA 0183-sanomaa. Luotain keilaa 30- ja 200-kHz taajuuksilla ja siinä on sisään rakennettuna lämpötila- ja syvyysanturit. Kuviossa 3 on kuva monitaajuuskaikuluotaimesta, jota opinnäytetyössä käytetään. (Airmar EchoRange 2022.)

Kaikuluotain painaa 6.5 kilogrammaa ja toimii -5–60 celsiusasteen välillä. Luotain tarvitsee 12 volttia 300 milliampeeria tasavirtaa, että se toimii ja siinä kiinni olevat virta- ja tiedonsiirtojohdot ovat 15 metriä pitkät. (Airmar EchoRange 2022.)

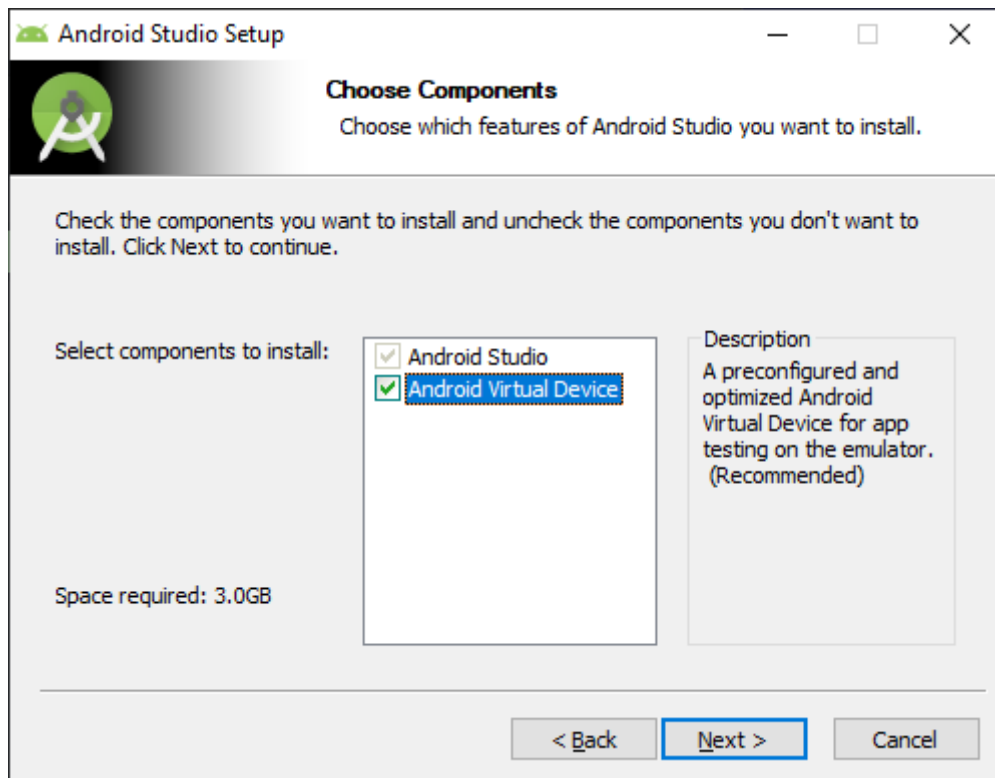


Kuvio 3. Airmar Dual Frequency EchoRange Smart Sensor M195 (Airmar EchoRange 2022)

## 4 OHJELMISTOJEN JA LAITTEIDEN ASENNUS JA ALUSTUS

### 4.1 Android Studio

Android-studion uusimman version voi ladata Androidin omilta verkkosivuilta ja asennusprosessissa kestää muutama minuutti. Asensin samalla tuen virtuaaliselle Android-laitteelle ohjelman testausten helpottamiseksi. Tuki virtuaaliselle Android-laitteelle voidaan valita asennusohjelmassa, kuten kuviossa 4 havainnollistetaan.



Kuvio 4. Tuen valinta virtuaalilaitteelle asennuksessa

Asennuksen jälkeen käynnistin Android Studion ja tein uuden projektin tyhjällä aktiviteetti-ikkunalla. Kun uusi projekti oli luotu, tein virtuaalisen Android-laitteen helpottamaan ohjelman testausta. En valinnut pienimmäksi API-versioksi kovin uutta versiota, koska halusin varmistaa, että ohjelma toimii vanhemmillakin laitteilla, mutta ei liian vanhaakaan, ettei tarvitse olla kovin nopeasti päivittämässä sovellusta uudemmalle versiolle.

Virtuaali-laitteen luonnin jälkeen käynnistin ja ajoin sovelluksen siinä varmistaakseni, että kaikki toimii. Onnistuneen projektin ja virtuaalilaitteen luonnin jälkeen aloitin sovelluksen teon.

## 4.2 Raspberry Pi

Raspberry Pi -tietokoneen käyttöönottoon tarvitaan MicroSD-muistikortti, johon asennetaan käyttöjärjestelmä. Käytin opinnäytetyössä 16 gigabitin muistikorttia, mutta pienemmän muistikoon korttia voidaan käyttää. Asensin käyttöjärjestelmän käyttämällä Raspberry Pi Imager -ohjelmaa, jolla sain tehtyä MicroSD-kortille Raspberry Pi -tietokoneen käynnistyksessä ajettavan kuvakkeen. Käyttöjärjestelmäksi valitsin Raspberry Pi OS, koska siitä löytyy valmiiksi tuki Raspberry Pi -tietokoneen piirilevyn ominaisuuksille, joita mahdollisesti halusin käyttää, esimerkiksi GPIO-pinnit.

Käyttöjärjestelmän asennuksen jälkeen asetin MicroSD-kortin paikalleen Raspberry Pi -tietokoneeseen ja käynnistin sen kytkemällä virran päälle. Raspberry Pi -tietokoneessa ei ollut erillistä virtakytkintä, joten käynnistäminen tapahtui laittamalla virtajohto pistokkeeseen kiinni. Kytkin Raspberry Pi -tietokoneeseen näytön, näppäimistön ja hiiren, että sain aloitusasetukset tehtyä.

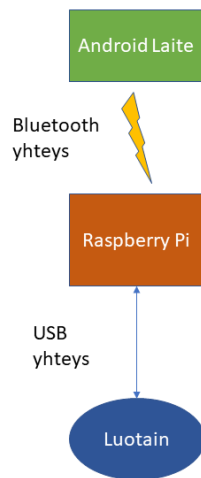
Aloitusasetuksissa asetin käyttäjänimen ja salasanan, määritin millä aikavyöhykkeellä, missä maassa ollaan sekä näppäimistön asettelun. Raspberry Pi -tietokone voidaan yhdistää internetiin joko langattomasti käyttäen Wifi-yhteyttä tai yhdistää langallisesti verkkokaapelilla. Yhdistin sen langattomasti, koska se oli kätevämpi vaihtoehto.

Kun Raspberry Pi -tietokone käynnistyi työpöydälle, avasin terminaali-ikkunan, jossa ajoin seuraavat komennot: `sudo apt update` && `sudo apt upgrade -y`. Näillä komennoilla päivitin ensin asennettujen pakettien tiedot, jotta saatiin uusimmat versiot paketeista ja `apt upgrade` -komennolla asensin päivitykset Raspberry Pi -tietokoneeseen. Päivitysten jälkeen kävin sallimassa SSH-yhteydet, että voin ottaa Raspberry Pi -tietokoneeseen etäyhteyden internetin kautta. Raspberry Pi oli tämän jälkeen valmis käyttöönottoon.

## 5 JÄRJESTELMÄN TOTEUTUS

### 5.1 Kytkenät

Yhdistin Android-laitteen Raspberry Pi -tietokoneeseen käyttämällä Bluetooth-yhteyttä ja yhdistin kaikuluotaimen Raspberry Pi -tietokoneeseen USB-kaapelilla. Seuraavassa kuviossa 5 ovat kytkenät visualisoituna.



Kuvio 5. Kytkenät visualisoituna

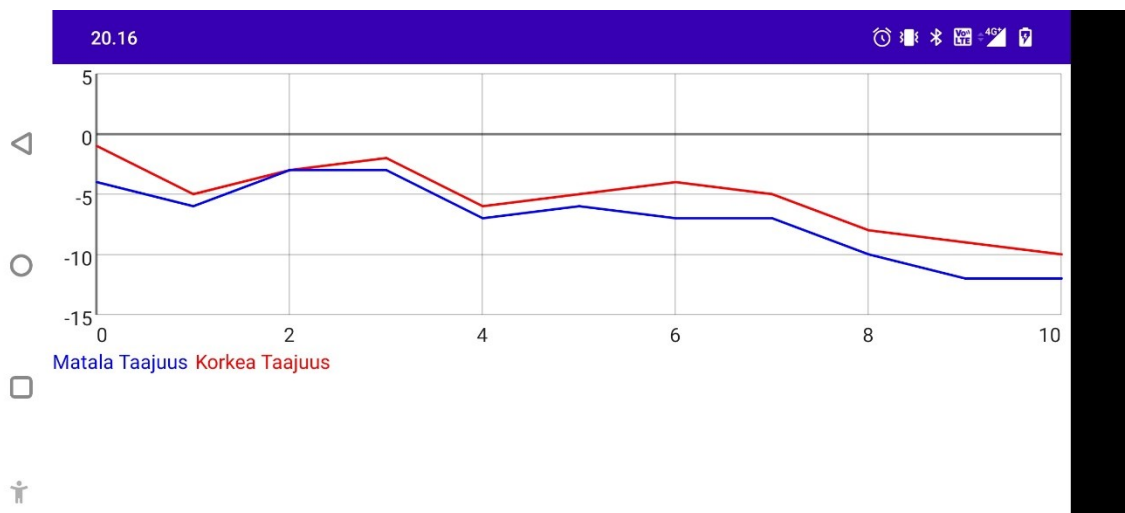
### 5.2 Android-ohjelma

Aloitin ohjelman teon tekemällä käyttöliittymän. Mitta Oy:n mielestä sen pitää olla tutun tuntuinen ja helppo käyttää, joten päädyin ottamaan mallia vanhemmista Windows-ohjelmista ja kasettisoittimesta. Seuraavassa kuviossa 6 on kuvakaappaus käyttöliittymän prototyypistä.



Kuvio 6. Käyttöliittymän prototyyppi

Haluttu ominaisuus oli nähdä verhokäyrää mittauksesta, joten tein kuvaajalle erillisen näkymän. Kaikuluotaimesta saadaan suoraan matalan- ja korkeantaajuuden luotauksen data heksadesimaaliarvoina, joka pitää muuttaa kokonaisluvuiksi, että sitä voidaan käyttää kuvaajan piirtämisessä. Seuraavassa kuviossa 7 on kuvaajanpiirron prototyyppi.



Kuvio 7. Kuvaajanpiirron prototyyppi

Mitta Oy halusi, että kaikki mittausdata tulee Android-laitteeseen, josta se voidaan siirtää jatkoprosessointiin. Koska data tulee suorana jonona Bluetooth-yhteyden kautta, tein ohjelmaan prosessin, joka kirjoittaa datan tiedostoon. Alla olevassa kuviossa 8 on esimerkkikoodista, jossa tehtiin kansio ja sen sisälle tiedosto, johon kirjoitettiin viesti.

```
public void Kirjoita(){
    Log.e( tag: "TEST", msg: "kirjoitetaan");

    KysyKirjoitusLupa();

    String FILENAME = "hello_file.txt";
    String message = "hello world";
    String kansioPolku = Environment.getExternalStorageDirectory()
        .getAbsolutePath()+"/Taajuusluotain";
    FileOutputStream fos;

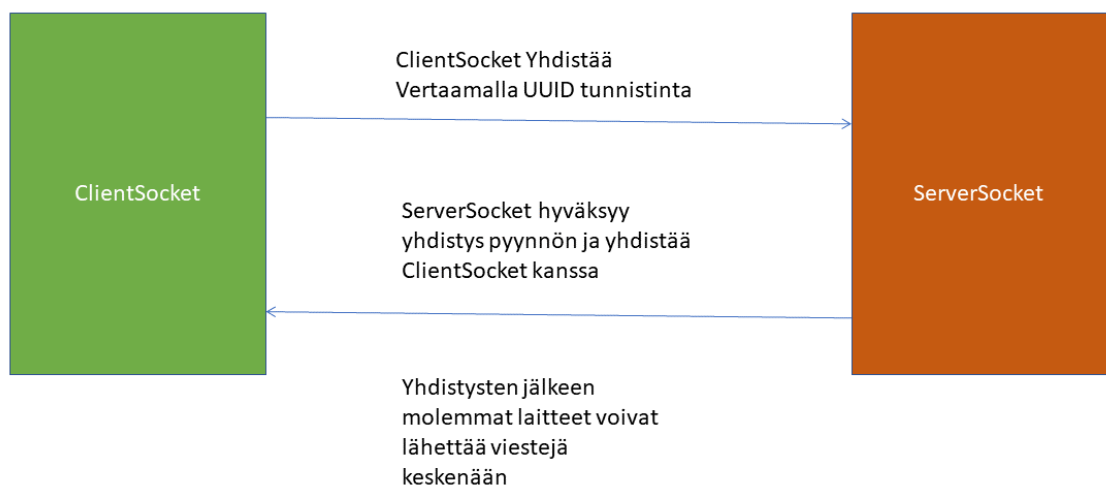
    //tehdään kansio johon tehdään tiedostot
    File tjKansio = new File(kansioPolku);
    if( !tjKansio.exists()){
        tjKansio.mkdirs();
        tjKansio.setExecutable(true);
        tjKansio.setReadable(true);
        tjKansio.setWritable(true);
    }

    //tehdään tiedosto kansioon
    File testfile = new File(kansioPolku,FILENAME);
    if(!testfile.exists()) {
        try {
            fos = new FileOutputStream(testfile);
            ObjectOutputStream oos = new ObjectOutputStream(fos);
            oos.writeObject(message);
            oos.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Kuvio 8. Esimerkki kansion ja tiedoston luonnista

### 5.3 Bluetooth-yhteys

Aloitin Bluetooth-yhteyden luonnin tekemällä ilmoituksen käyttäjälle, että yhdistää Android-laitteen Raspberry Pi -tietokoneeseen Bluetooth-yhteydellä. Bluetooth-ohjelmointi käyttää verkkotekniikan puolelta tuttua Socket -mallia, jossa ClientSocket yhdistää ServerSocketiin vertaamalla ennalta määritettyä tunnistinta. Tein Android-ohjelmaan ClientSocketin ja Raspberry Pi -tietokoneeseen ServerSocketin. Seuraavassa kuviossa 9 on Sockettien yhdistys visualisoituna.



Kuvio 9. Socket -yhdistys visualisoituna

#### 5.3.1 Bluetooth-yhteys Android-laitteessa

Tein Android-ohjelmassa uuden BluetoothManager-olion, josta tein uuden BluetoothAdapter-olion. BluetoothAdapter-oliota käyttämällä sain yhdistetyn Bluetooth-laitteen tiedot käyttöön ohjelmassa ja pystyin tekemään ohjelmalle ClientSocketin. Kuviosta 10 näemme esimerkin, miten yhdistys tapahtui ohjelmassa.

Onnistuneen yhdistyksen jälkeen tein BluetoothSocket-olion, joka on valmiina vastaanottamaan ja lähettämään dataa. Jotta voin lähettää ja vastaanottaa dataa, tein InputStream- ja OutputStream-oliot, jotka toimivat välikappaleina tiedon siirron ja tiedonkäsittelyn välissä.

Vastaanotettu data luettiin puskuriin, josta se voitiin siirtää jatkokäsittelyyn. Lähettävä data muutettiin tavuiksi, jonka jälkeen se oli valmis kirjoitusmetodiin

lähetettäväksi. Alla olevassa kuviossa 10 esimerkkikoodi tiedonlähetys- ja vastaanottometodista.

```
public void run() {
    pus kuri = new byte[1024];
    int numBytes; // bittimäärä input striimistä

    // Kuunnellaan Input Striimiä kunnes keskeytetään
    while (true) {
        try {
            // Luetaan input striimiä
            numBytes = inputStream.read(pus kuri);
            // lähetetään luettu data eteenpäin
            Message readMsg = handler.obtainMessage(
                numBytes, pus kuri);
            // tähän laittaa minne haluaa lähettää datan
            readMsg.sendToTarget();

        } catch (IOException e) {
            Log.d( tag: "BT", msg: "Input Striimi keskeytettiin", e)
            break;
        }
    }
}

public void write(byte[] bytes) {
    try {
        outputStream.write(bytes);
        Message writtenMsg = handler.obtainMessage(
            what: 2, pus kuri);
        // Lähetetään viesti yhdistettyyn bluetooth laitteeseen
        writtenMsg.sendToTarget();

    } catch (IOException e) {
        Log.e( tag: "BT", msg: "Virhe tapahtui viestin lähetyksessä", e);
    }
}
```

Kuvio 10. Esimerkki datan vastaanotto- ja lähetysmetodista

Testasin yhteyttä tekemällä ohjelman Raspberry Pi -tietokoneelle, joka ilmoitti onnistuneesta tiedonsiirron vastaanotosta ja lähetti Android-ohjelmaan komennon.

Android-ohjelma vastaanotti komennon ja ilmoitti, että tiedonsiirto on onnistunut. Kun molemmat laitteet ilmoittivat tiedonsiirron onnistuneen, pystyin lisäämään ohjelmaan toiminnallisuuksia.

### 5.3.2 Bluetooth-yhteys Raspberry Pi -tietokoneessa

Aloitin Raspberry Pi -tietokoneelle Bluetooth-tiedonsiirto-ohjelman teon asentamalla `libbluetooth-dev` -ja `pybluez` -paketit. Näistä paketeista sain käyttöön ohjelmien työkalut, jotka hyödyntävät Bluetooth-yhteyttä. Sain paketit asennettua ajamalla komennot `sudo apt install bluetooth libbluetooth-dev` ja `pip3 install pybluez`. Pakettien asennusten jälkeen pystyin aloittamaan Python-koodin kirjoittamisen. Alustin ohjelman tekemällä `RFCOMM BluetoothSocket`in ja yhdistämällä sen mihi tahansa porttiin. Tein muuttujan, johon asetin UUID-tunnistimen, ettei sitä tarvitsisi kirjoittaa erikseen kaikkiin tarvittaviin kohtiin.

UUID on 128-bittinen numerosarja, joka tehdään yhdistämällä laitteen MAC-osoitteen ja luontihetken aikaleiman bitit keskenään. Opinnäytetyössä käytin UUID generator-internetsivulla olevaa UUID generaattoria UUID-tunnistimen tekoon. (TransparenTech LLC 2022).

Seuraavaksi tein palvelimeen mainostuskohdan, jota `ClientSocket` etsii yhdistyksen aikana. Kun mainostuskohta löytyi vertasi `ClientSocket` omaa UUID-tunnistintansa `ServerSocket`in UUID-tunnistimeen. Samalla UUID-tunnistimella olevan `ServerSocket`in löydyttyä, hyväksyi `ServerSocket`-ohjelma yhteyden `ClientSocket`-laitteeseen. Yhteyden muodostuksen jälkeen pystyin vastaanottamaan ja lähettämään dataa laitteiden välillä. Seuraavassa kuviossa 11 on esimerkkiohjelma `ServerSocket`in tekemiseen ja tulevan datan kuunteluun.

```

from bluetooth import *

bt_serverSocket = BluetoothSocket(RFCOMM)
bt_serverSocket.bind(("",PORT_ANY))
bt_serverSocket.listen(1)

port = bt_serverSocket.getsockname()[1]

uuid = "5b68338a-6377-11ed-81ce-0242ac120002"

advertise_service([ bt_serverSocket, "TESTISERVERI",
                    service_id = uuid,
                    service_classes = [ uuid, SERIAL_PORT_CLASS ],
                    profiles = [ SERIAL_PORT_PROFILE ],
                    #
                    protocols = [ OBEX_UUID ]
                    ])

print("Odotetaan yhteyttä RFCOMM kanavalta: %d" % port)

bt_clientSocket, bt_clientInfo = bt_serverSocket.accept()
print("Hyväksyttiin yhteys: ", bt_clientInfo)

try:
    while True:
        data = bt_clientSocket.recv(1024)
        if len(data) == 0: break
        print("vastaa otettiin: %d" % data)
except IOError:
    pass

print("yhteys katkaistu")

bt_clientSocket.close()
bt_serverSocket.close()
print("Jobs Done")

```

Kuvio 11. ServerSocketin esimerkki

## 6 POHDINTA

Perehdyin minulle kokonaan uusiin teknologioihin ja vaihdoin langattoman tiedonsiirron tavan toiseen tekniikkaan kesken opinnäytetyön, minkä takia en ehtinyt tehdä ohjaus- ja tiedonsiirto-ohjelmaa Raspberry Pi -tietokoneeseen. Nämä ovat keskeisimmät järjestelmän toimivuuteen liittyvät asiat, mitkä olen tekemässä vuoden 2023 alkupuolesta.

Henkilökohtaisena tavoitteena opinnäytetyössä oli oppia syvemmin, miten voidaan siirtää tietoa eri laitteiden välillä, mikä on omasta mielestä onnistunut. Opinnäytetyön aikana olen kerrannut miten TCP- ja UDP-protokollat toimivat, koska Bluetooth-yhteys toimii samankaltaisesti kuin TCP-yhteydenotto.

Työskentelin kahden erilaisen Linux-pohjaisen käyttöjärjestelmän kanssa, Android ja Raspberry Pi OS, joista Android poikkeaa huomattavasti useimmista Linux-järjestelmistä. Opin, että Android-järjestelmän tiedostojärjestelmä on aivan erilainen verrattuna muihin Linux-käyttöjärjestelmiin, minkä takia Java-ohjelmointikielen SSH-yhteys kirjasto vaatii huomattavasti muokkausta, että sen saisi toimimaan.

Jatkokehitysideoina olisi lisätä toiminnallisuuksia, joilla voidaan hienosäätää kailuotaimen asetuksia tarkempien mittaustulosten saamiseksi. Lisättävien toiminnallisuuksien yhtenä esimerkkinä on tiedon säätäinen ohjelmaan, kuinka syvällä luotain on veden pinnasta, että se otetaan huomioon mittausdatassa. Toisena esimerkkinä on lisätä ohjelmaan luotaus tiheyden säätö. Lisäksi ohjelmaan olisi mahdollista tehdä täysin manuaalinen tila, jossa käyttäjä nappia painamalla saa yhden pulssin luotaimesta.

Kartan ja GPS-paikannuksen, lisääminen Android-sovellukseen helpottaisi ajan tasalla pysymistä, mistä on jo luodattu. GPS-paikannin löytyy lähes kaikista Android-laitteissa ja koodikirjastot sen käyttöön sisältyy Java- ja Kotlin-ohjelmointikieliin. Google Maps API:a olisi mahdollista käyttää kartan näyttämiseen soveluksessa. Voitaisiin myös tutkia, olisiko mahdollista korvata Raspberry Pi -tietokone jollakin muulla laitteella, kuten esimerkiksi jollakin Arduino-alustalla tai ESP32-mikrokontrollerilla, koska Raspberry Pi -tietokone saattaa olla liian kookas ja kuluttaa liikaa energiaa luotaukseen.

## LÄHTEET

Airmar EchoRange 2022. EchoRange™ Dual-frequency Smart™ Sensor product description Viitattu 1.12.2022  
<https://www.airmar.com/productdescription.html?id=113>.

Android Authority 2022. The history of Android. Viitattu 20.11.2022  
<https://www.androidauthority.com/history-android-os-name-789433/>.

Android Developers 2022a. Android Studio. Viitattu 29.11.2022  
<https://developer.android.com/studio>.

Android Developers 2022b. Features. Viitattu 29.11.2022  
<https://developer.android.com/studio/features>.

Bluetooth 2022a. Learn about Bluetooth. Viitattu 29.11.2022  
<https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>.

Bluetooth 2022b. Range. Viitattu 29.11.2022 <https://www.bluetooth.com/learn-about-bluetooth/key-attributes/range/>.

Fromaget, P. 2022. The Epic Story of the Raspberry Pi. Viitattu 29.11.2022  
<https://raspberrytips.com/raspberry-pi-history/>.

Javatpoint 2022. History of java. Viitattu 29.11.2022  
<https://www.javatpoint.com/history-of-java>.

Raspberry Pi Foundation 2022. Raspberry Pi 4 Model B. Viitattu 29.11.2022  
<https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/>.

TransparenTech LLC 2022. UUID Generator version 1. Viitattu 1.12.2022  
<https://www.uuidgenerator.net/version1>.