

Paineanturitiedon käsittely ja sen langaton tiedonsiirto Android-sovellukselle

Niko Viljanen

Opinnäytetyö

Joulukuu 2015

Tekniikan ja liikenteen ala

Insinööri (AMK), hyvinvointiteknologian tutkinto-ohjelma

Tekijä(t) Viljanen, Niko	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä 11.12.2015
	Sivumäärä 56	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty: x
Työn nimi Paineanturitiedon käsittely ja sen langaton tiedonsiirto Android-sovellukselle		
Tutkinto-ohjelma Hyvinvointiteknologian koulutusohjelma		
Työn ohjaaja(t) Matti Kurki, Matti Siistonen		
Toimeksiantaja(t) Saku Sourulahti		
Tiivistelmä <p>Opinnäytetyön aiheena oli luoda langaton tietoyhteys mittausjärjestelmän ja Android-sovelluksen välille sekä luoda ohjelma, joka prosessoi mittausjärjestelmän keräämää dataa reaaliajassa. Opinnäytetyö oli osa tuotekehitysprojektia, joka sai alkunsa projektiryhmäläisten omasta tuotteesta. Projektin tavoitteena oli kehittää prototyyppi juoniorijalkapalloilijoiden käyttöön tarkoitetusta älysukasta. Työryhmä sai JAMK Generatorilta apua liikeidean kehittämisessä sekä taloudellista tukea prototyyppilaitteen kehittämiseksi.</p> <p>Jalkapalloon tarkoitetun älysukan tarkoituksena on rekisteröidä käyttäjän jalkapallopelin aikana tapahtuvat pallokosketukset eri jalanosilla sekä käyttäjän askelten määrä ja askeltiheys pelin aikana. Käyttäjän on tarkoitus älysukan avulla pystyä kehittämään pallonhallintataitojaan. Opinnäytetyössä Android-laitteelle toteutettiin tätä tarkoitusta varten pelisovellus, jonka käyttäjä pystyy yhdistämään langattomasti älysukkaan.</p> <p>Pallokosketusten ja askelten tunnistamista varten luotiin algoritmeja sukassa sijaitsevalle mikrokontrollerille. Mikrokontrolleriin ohjelmoidut algoritmit prosessoivat jalkapallosukan sisään integroiduilta paineantureilta tulevan datan ja lähettävät prosessoidun datan Android-sovellukselle langatonta Bluetooth low energy -yhteystekniikkaa käyttäen.</p> <p>Tuloksena saatiin toimiva prototyyppilaittekokonaisuus. Prototyyppiälysukan avulla voidaan tunnistaa käyttäjän jalkapallokosketukset ja askeleet käyttötilanteessa. Langaton tietoyhteys mittausjärjestelmän ja Android-sovelluksen välillä toimii luotettavasti. Prototyyppiä kehitettäessä todettiin, että tuoteidean vieminen todelliseksi tuotteeksi vaatisi paljon resursseja ja se olisi teknisesti haastavaa toteuttaa siten, että se vastaisi käyttäjien odotuksia tuotteen hinnan ja laadun puolesta.</p>		
Avainsanat (asiasanat) Bluetooth low energy, puettava teknologia, Android, älysukka		
Muut tiedot		

Author(s) Viljanen, Niko	Type of publication Bachelor's thesis	Date 11.12.2015
	Number of pages 56	Language of publication: Finnish
		Permission for web publication: x
Title of publication The processing of pressure sensor data and transferring it wirelessly to an Android application		
Degree programme Degree Programme in Wellness Technology		
Supervisor(s) Kurki, Matti & Siistonen, Matti		
Assigned by Sourulahti, Saku		
Description <p>The purpose of the bachelor's thesis was to create a wireless connection between a sensor system and an Android application and to create a program for a microcontroller to process the data from the sensor system in real time. The bachelor's thesis was a part of a product development project where the product idea came from the project team itself. The project team received financial assistance and help developing the business concept for the product from JAMK Generator. The goal of the project was to create a prototype of a smart sock designed for junior football players.</p> <p>The purpose of the football smart sock is to gather data about the amount of kicks performed with the different parts of the foot during a football game and also about the amount of steps and the average amount of steps per minute during a game. The purpose is also to allow the user to train their football handling skills with a game application by connecting the sock wirelessly to the application. The application was programmed for Android.</p> <p>Algorithms were developed, for the microcontroller attached to the sock, so that the system is able to recognise when the user kicks the ball and when a step occurs. The algorithms programmed for the microcontroller process the data from the pressure sensors, which are integrated into the sock's textile. The processed data is sent to the Android application using Bluetooth low energy wireless technology.</p> <p>The end result was a functioning prototype of the smart sock and the Android application. The sock was able to recognise the user's steps and football kicks. The wireless connection between the smart sock and the Android application worked flawlessly. While working on the project, it became clear that developing the product for the actual market would require a lot of resources and it would be challenging to meet the user's expectations in terms of pricing and quality.</p>		
Keywords (subjects) Bluetooth low energy, wearable technology, Android, smart sock		
Miscellaneous		

Sisältö

Termit ja lyhenteet.....	3
1 Johdanto.....	5
2 Puettava teknologia.....	6
2.1 Puettavan teknologian alkutaival	6
2.2 Puettavan teknologian mahdollisuudet ja haasteet	7
3 Jalkapalloöllysukka	9
3.1 Harjoittelun merkitys oppimisen kannalta	9
3.2 Tuoteidea	10
3.3 Tekniikka	12
4 Algoritmit	15
4.1 Algoritmien tehtävä	15
4.2 Algoritmit käytännön sovelluksissa	16
5 WLAN-, Bluetooth- ja ZigBee -tekniikoiden vertailu akkukäyttöisissä laitteissa	16
5.1 WLAN-tiedonsiirtotekniikka.....	16
5.2 Bluetooth 4.0 -tiedonsiirtotekniikka	17
5.3 ZigBee-tiedonsiirtotekniikka	18
5.4 Vertailu	18
6 Opinnäytetyön toteutus.....	19
6.1 Vaatimusten määrittely	20
6.2 Algoritmien luominen mikro-ohjaimelle.....	21
6.3 Datan linearisointi ja mittausalueen kalibrointi	22
6.4 Pallokosketusten tunnistaminen	27
6.5 Askeleen tunnistaminen.....	28
6.6 Bluetooth low energy -yhteyden luominen.....	28
6.6.1 Bluetooth low energy -yhteyden valinta.....	28
6.6.2 Tiedon lähettäminen ja vastaanottaminen BLE-moduulilla	29
6.7 Android-sovelluksen ohjelmointi	30
6.7.1 Android-käyttöjärjestelmä	30
6.7.2 Komponentit ja aktiviteetit	31
6.7.3 Bluetooth low energy -yhteys Android-käyttöjärjestelmässä.....	36
7 Opinnäytetyön tulokset.....	44

7.1 Onnistumiset	44
7.2 Puutteet ja ongelmat	44
8 Pohdinta.....	46
Lähteet.....	48
Liitteet	52
Liite 1. Pallokosketuksen tunnistusalgoritmin vuokaavio.....	52
Liite 2. Askeleen tunnistusalgoritmin vuokaavio.....	55

Kuviot

Kuvio 1. Sensoria-älysukska.....	8
Kuvio 2. Zoi-juoksuanturi	9
Kuvio 3. Älysukskatuotekonseptin toimintasykli.....	11
Kuvio 4. Sukka, jonka sisässä anturit ovat integroituina	13
Kuvio 5. Havainnekuva sukan sisään integroitavien anturien sijainneista ja muodoista.....	13
Kuvio 6. Älysukan elektroniikkayksikkö	14
Kuvio 7. Näkymä CoolTerm-ohjelmasta	22
Kuvio 8. Anturin ja mikrokontrollerin kytkentäkaavio.....	23
Kuvio 9. Havainnekuva paineanturin rakenteesta.....	23
Kuvio 10. Jalkapöytäanturin raakadataa ajan funktiona 285 Hz:n näytteenottotaajuudella.....	24
Kuvio 11. Muuttujien arvot voiman funktiona	27
Kuvio 12. Raakadataa oikean jalan juoksuaskeleista 285 Hz:n näytteenottotaajuudella päkiäanturilta.....	28
Kuvio 13. Jalkapalloälysukan mittausjärjestelmä.....	29
Kuvio 14. Aktiviteetin elinkaari ja siirtymävaiheiden metodit	34
Kuvio 15. Kuvakaappaus sovelluksen laiteskannausnäkyvästä.....	37
Kuvio 16. Kuvakaappaus pelin ensimmäisestä tasosta	42
Kuvio 17. Tason läpäistyään käyttäjä näkee pisteensä	43
Kuvio 18. Näkymä, jos käyttäjä ei läpäise tasoa	43

Taulukot

Taulukko 1. Langattomien standardien ominaisuuksia.....	19
Taulukko 2. BLE-moduulin lähettämä merkkijono kosketustyyppistä riippuen	30
Taulukko 3. Aktiviteetin eri olotilat.....	35

Termit ja lyhenteet

Aktiviteetti	Aktiviteetti näyttää Android-sovelluksessa käyttäjälle yhden käyttöliittymänäkymän kerrallaan.
A/D-muunnin	(Analogia-digitaalimuunnin) Käytetään elektroniikassa muuntamaan analoginen signaali digitaaliseksi.
BLE	(Bluetooth low energy) Bluetooth 4.0 -versioon pohjautuva langaton tiedonsiirtoteknologia.
Intent	Intent toimii sitovana elementtinä eri aktiviteettien välillä. Intentin avulla lähetetään kutsuvan aktiviteetin tiedot kutsuttavalle aktiviteetille, ja sen välityksellä on mahdollista siirtää arvoja mm. kahden eri aktiviteetin välillä.
MAC-osoite	(Media Access Control) Yksilöllinen osoite, jolla IEEE 802 -standardeja käyttävät langattomat laitteet yksilöidään.
Mikrokontrolleri	Pieni tietokone, jossa on mikroprosessori, muisti ja ohjelmoitavia liityntälohkoja. Mikrokontrollereita käytetään elektroniikkalaitteissa.
Näytteenottotaajuus	Taajuus, jolla ilmoitetaan analogisignaalista otettavien näytteen määrän sekunnissa.

RX	(Receiver) Vastaanotin tiedonvälityksessä.
Sarjaportti	Kahden tietokonelaitteen väliseen tietoliikenteeseen käytettävä tietoliikenneportti.
TX	(Transmitter) Lähetin tiedonvälityksessä.
USB	(Universal Serial Bus) Sarjaväyläarkkitehtuuri oheislaitteiden liittämiseksi tietokoneeseen.

1 Johdanto

Ennusteen mukaan vuonna 2019 on maailmanlaajuisesti käytössä yli puoli miljardia puettavan teknologian laitetta ja 5,2 miljardia ihmistä omistaa älypuhelimien (Global Mobile Data Traffic Forecast Update 2015). Tämä puettavan teknologian markkinoiden räjähdysmäinen kasvu ja kysynnän lisääntyminen on ohjannut yrityksiä kehittämään uudentyyppisiä ratkaisuja ihmisten tarpeisiin puettavan teknologian avulla. Uusien teknologisten laitteiden kehittäminen on myös muuttunut viime vuosien aikana yhä helpommaksi ja halvemmaksi. Laitteiden prototyyppien suunnitteluun ja valmistamiseen ei enää tarvita kalliita välineitä ja vuosikausien tietotaitoa alalta. Suurelta osin tähän on vaikuttanut internetin tarjoama lähes rajaton tietomäärä, mutta myös oleellisesti kehitystyökalujen kehittyminen jatkuvasti helppokäyttöisemmiksi ja edullisemmiksi. Nykypäivänä käytännössä kuka vain, kenellä on internetyhteys, voi itseopiskellen viedä eteenpäin monimutkaisiakin tuoteideoita ja valmistaa niistä toimivia prototyyppijä lyhyessä ajassa.

Tässä opinnäytetyössä tavoitteena oli luoda langaton tietoyhteys paineanturimittausjärjestelmän ja Android-sovelluksen välille sekä luoda mikrokontrolleriohjelma, joka prosessoi mittausjärjestelmän keräämää dataa reaaliajassa. Opinnäytetyö oli osana jalkapalloon tarkoitettun älysukan tuotekehitystyötä. Älysukka on tarkoitettu nuorten jalkapalloilijoiden henkilökohtaisten pallonhallintataitojen kehittymisen tueksi. Opinnäytetyössä keskityttiin pääasiassa anturidatan prosessointiin, datan langattomaan siirtämiseen sekä Android-sovelluksen kehitystyöhön. Opinnäytetyön lopuksi arvioitiin työssä tuotettujen ohjelmien ja menetelmien soveltamisen hyödyllisyyttä jatkotuotekehityksen kannalta sekä ohjelmien toimintavarmuutta.

Tuotekehitystyöstä vastaava projektiryhmä sai JAMK Generatorilta apua liikeidean kehittämässä sekä taloudellista tukea prototyypilaitteen kehittämiseen. JAMK Generator on JAMK:n yrittäjyyden ja innovaatiotoiminnan yksikkö, jonka tehtävänä on tukea opiskelijoiden tuote- ja liikeideoiden kehittämistä.

2 Puettava teknologia

2.1 Puettavan teknologian alkutaival

Ensimmäinen puettavaksi teknologiaksi luokiteltava väline valmistettiin 1960-luvun alussa. Kyseessä oli tupakka-askin kokoinen kengän sisään asetettava analoginen tietokone, jolla oli tarkoitus saavuttaa paremmat mahdollisuudet voittaa ruletissa. Käyttäjä painoi varpaallaan kytkimen päälle kellottaakseen yhden rulettipallon kiertoajan ruletin ympäri. Kun rulettipallo ylitti ruletin referenssipisteen, käyttäjä lopetti ajanottamisen varpaallaan ja tietokone laski pallon nopeuden. Samanlainen kellotus tehtiin myös numerokiekon pyörimisnopeuden määrittämiseksi. Pallon ja numerokiekon nopeudesta tietokone päätteli, mihin kohtaan pallo tulee numerokiekolla tippumaan. Käyttäjälle oli kengästä johdettu korvaan lähes huomaamaton kuuloke, joka soitti yhtä kahdeksasta sävelestä. Sävelet kertoivat käyttäjälle, mihin numerokiekon kahdeksannekseen pallo tulee annettujen tietojen perusteella pysähtymään. (Thorp n.d.)

Puettavaa teknologiaa on pyritty hyödyntämään vuosien saatossa useisiin erilaisiin käyttökohteisiin, kuten kehon fysiologisiin mittauksiin ja terveydenhuollon sovelluksiin. Ensimmäinen patentti langattomalle sykkeenmittaukselle myönnettiin vuonna 1985 (Innovaatiot n.d.).

Ensimmäinen patentti lihasaktiivisuuden mittaukseen tekstiiliin integroitujen

EMG-elektrodien avulla myönnettiin vuonna 2009 (Method and outfit for measuring of action of muscles of body 2009). 2010-luvun alussa kuluttajien suosioon nousivat aktiivisuusrannekkeet. Puettava teknologia on kehittynyt paljon siitä, mistä se lähti liikkeelle ja nopeasti kehittyvän teknologian myötä myös puettavaa teknologiaa pyritään soveltamaan jatkuvasti uusiin käyttökohteisiin.

2.2 Puettavan teknologian mahdollisuudet ja haasteet

Nykyisin teknologian integroiminen vaatteisiin mahdollistaa ruumiin eri fysiologisten ja biomekaanisten toimintojen monitoroinnin vaivattomasti. Näillä tekniikoilla voidaan seurata esimerkiksi elimistön happisaturaatiota ja sykettä suorituksen aikana. Näitä tietoja hyödynnetään mm. sotilaskäytössä ja pelastushenkilökunnan työssä. Monitoroinnilla voidaan parantaa vaikeissa olosuhteissa työskentelevien työturvallisuutta seuraamalla kehon fysiologisia toimintoja tai seuraamalla henkilön sijaintia GPS:n avulla. (Thitz 2015, 7.)

Suurimmat teknologiset haasteet puettavan teknologian kehityksessä liittyvät laitteiden akun kestoon, käyttöliittymään, sovelluksiin sekä näyttöteknologiaan. Älylaitteiden käyttäjistä 90 % pitää oleellisena sitä, että laitteen käyttöliittymä perustuu kosketukseen. Useat puettavaan teknologiaan keskittyvät yritykset käyttävät valmiita alustoja, kuten Googlen Android Wearia ja Intelin Edisonia, sillä tällöin kaikkea ei tarvitse kehittää alusta asti itse. (Puettava älyteknologia mahdollisuus suomalaisille yrityksille n.d.)

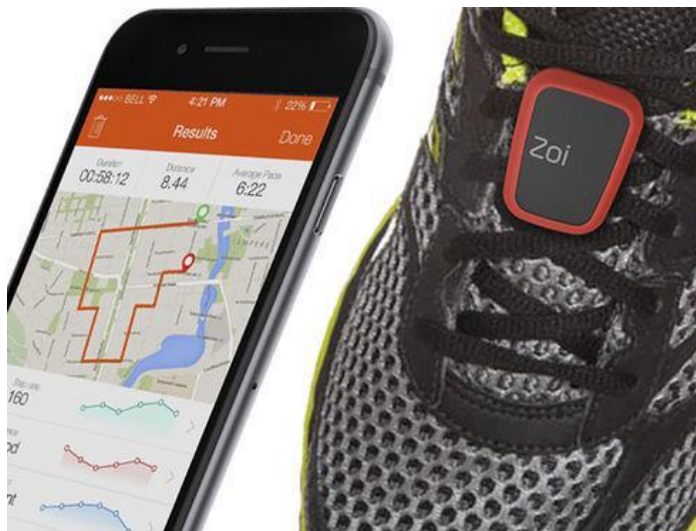
Puettavien älylaitteiden suurimmat markkinat löytyvät USA:sta ja Euroopasta. Vuonna 2014 kuntoilussa käytetyt älylaitteet kattoivat 97 % koko puettavan teknologian markkina-arvosta, mutta kuntoilun lisäksi älylaitteille on kasvamassa yhä suurempi kysyntä terveydenhuollon saralla.

Terveydenhuollossa on tällä hetkellä, maailman väestön ikääntyessä, kasvava tarve potilastyytyväisyydelle, reaaliaikaiselle potilaan seurannalle sekä kulujen minimoimiselle. (Puettava älyteknologia mahdollisuus suomalaisille yrityksille n.d.)

Kuluttajamarkkinoilla suurimmat puettavan teknologian mahdollisuudet ovat todennäköisesti kuntoilupuoella, sillä biomekaanisten toimintojen seuraamisen on todettu nopeuttavan kehittymistä urheilusuorituksessa (Maman, Kanupriya & Jaspal 2012). Kuntoilupuolen markkinoille on juuri saapunut ja saapumassa ensimmäisiä uuden sukupolven puettavan teknologian laitteita, kuten yhdysvaltalaisyhtiön kehittämä Sensoria-älysukka (ks. kuvio 1) ja suomalaisyrityksen kehittämä Zoi-juoksuanturi (ks. kuvio 2).



Kuvio 1. Sensoria-älysukka (Sensoria Is A New Smart Sock... 2015)



Kuvio 2. Zoi-juoksuanturi (Runteq loi huippuvalmentajan kuntojuoksijoille 2014)

Sensoria-älysukan tekstiiliin on integroitu paineantureita, joiden avulla älypuhelinsovellus kerää ja analysoi tietoa juoksijan juokсутekniikasta. Näiden tietojen perusteella sovellus antaa palautetta käyttäjälle juokсутekniikan parantamiseksi. Myös Zoi-juoksuanturin tarkoitus on auttaa käyttäjää parantamaan juokсутekniikkaansa, mutta tekniikka perustuu kahteen kiihtyvyyssanturiin, jotka analysoivat juoksijan biomekaanista liikettä suorituksen aikana. Toinen kiihtyvyyssanturi sijoitetaan juoksukengän nauhoihin ja toinen asetetaan käyttäjän rintaan kiinnitysremmillä.

3 Jalkapalloälysusukka

3.1 Harjoittelun merkitys oppimisen kannalta

Lihasmotoriikan oppiminen vaatii liikkeen toistamista lukuisia kertoja. Kaikki arkipäiväiset kehon motoriset liikkeet, kuten kävely ja hampaiden harjaaminen, ovat vaatineet hermoston harjoittelua. Jalkapallossa hyvän pallonhallinnan oppiminen vaatii hyvin suuria määriä motorisen hermoston harjoittelua. Hyvän pallonhallinnan oppimiseksi ei riitä pelkästään jalkojen

hermoston harjoittaminen, vaan tarvitaan myös silmän ja muun kehon yhteistyötä. Pallonhallinta on siis havaintomotorinen taito. (Sourulahti 2015, 6.)

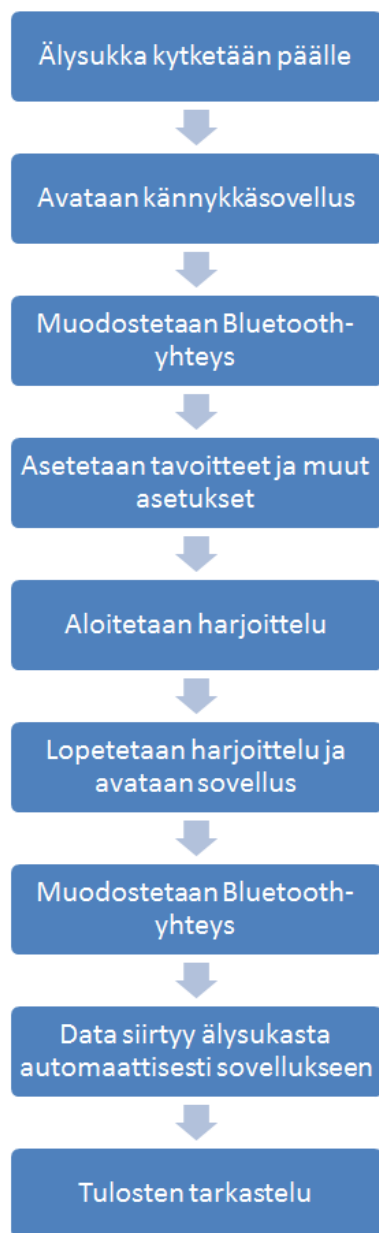
Ihmisen kädet ovat kehittyneet suorittamaan hienomotorisia toimintoja, kuten työkalujen käyttämistä ja esineiden liikuttamista. Jalat sen sijaan ovat kehittyneet kehon liikkumista varten, joten niiden päätehtävä on liikuttaa ihmistä. Jalkapallon pallonhallintataitoa pidetään motorisesti haastavana asiana oppia, sillä hienomotoriset taidot eivät ole jalkojen vahvuusalueella. Tästä syystä juniorijalkapalloilijan pallonhallintataitojen kehittymiselle on tärkeää suuret toistomäärät. Monipuolisia pallonhallintataitoja pidetään jalkapallossa tärkeänä, sillä nämä taidot ovat perusta mm. nopeille käännöksille, harhautuksille ja pallon kuljettamiselle pelitilanteissa. Ympäri maailmaa jalkapalloseurojen junioriakatemit asettavat yksilöllisen pallonhallinnan harjoittelulle ankarat tavoitetasot, joiden tarkoitus on saavuttaa maksimaalinen kehittyminen. Tavoitetasot eivät kuitenkaan takaa jalkapalloilijan harjoittelun laadullisuutta, joten tavoitetasojen täytyminen ei takaa maksimaalista kehittymistä. (Sourulahti 2015, 6-7.)

3.2 Tuoteidea

Opinnäytetyö on osana jalkapalloälyksukan tuotekehitystyötä. Tuoteideana on älysukka, jonka tarkoituksena on motivoida nuoria jalkapalloilijoita henkilökohtaisessa harjoittelussa. Älyksukan motivointikeino on käyttäjän henkilökohtaisen harjoittelun määrän seuraaminen ja pelillisen harjoittelun koukuttavuus.

Jalkapalloälyksukalla on mahdollista seurata pallon kosketusten kokonaismäärää, vasemman ja oikean jalan pallokosketusmäärää, jalkojen eri

osien välisten kosketusten määrää, pallon laukaisuvoimakkuutta ja pallonhallintataitoa. Harjoittelun määrää voidaan näillä tiedoilla seurata pitkällä aikavälillä ja käyttäjä voi asettaa harjoitusmäärälleen henkilökohtaisia tavoitteita. Näin käyttäjä saa tietoa päivän aikana saavutetusta kokonaisharjoittelumäärästä. Tuotekonseptin toimintasykli on esitetty kuviossa 3.



Kuvio 3. Älysuikkatuotekonseptin toimintasykli

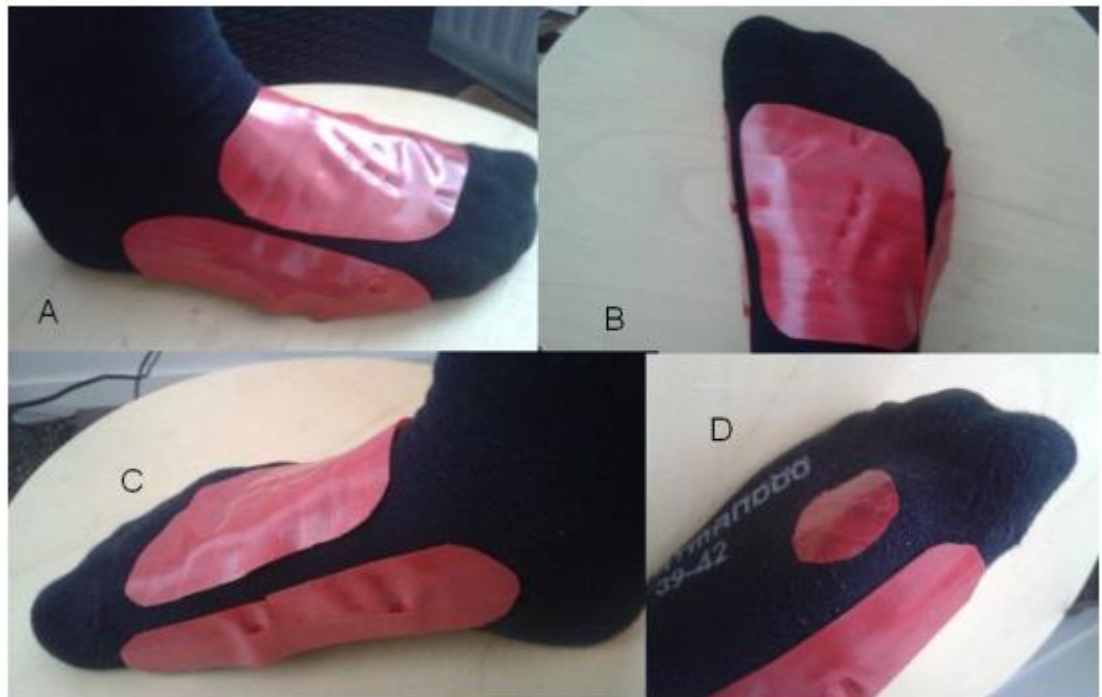
Älysukan motivointikeino on lähtöisin aktiivisuusmittarin motivointikeinosta. Suurimmaksi osaksi aktiivisuusmittalaitteen hyödyllisyys perustuu laitteen antamaan tietoon käyttäjän kokonaisliikunnan määrästä tietynä ajanjaksona. Laitteen käyttäjä tavoittelee kuntoilulla useimmiten parempaa kuntoa ja painon pudotusta. Yhtälailla jalkapalloon tarkoitettulla älysukalla pyritään soveltamaan harjoittelun määrän tietoa jalkapallon harjoitteluun, jossa harjoittelun määrällä on suuri vaikutus taitojen kehittymiseen.

3.3 Tekniikka

Älysukan (ks. kuvio 4) sisään integroitiin neljä resistiivistä paineanturia (ks. kuvio 5), jotka pyrittiin asettamaan sukkaan siten, että ne olisivat mahdollisimman huomaamattomat. Tavoitteena oli saada sukan ergonomia normaalin sukan tasolle. Anturit johdotettiin sukkaa pitkin elektroniikkaan, joka sijoitettiin sukkaan jalan pohjepuolelle. Sukassa käytettävä johdinmateriaali olisi lopullisessa tuotteessa sähkönjohtavuus- ja joustavuusominaisuuksiltaan erikoisjohdin. Elektroniikka (ks. kuvio 6) sisältää mikrokontrollerin anturidatan prosessointia ja tallentamista varten, akun virtalähteeksi ja Bluetooth-piirin tiedon lähettämistä ja vastaanottamista varten Android-puhelimen kanssa.



Kuvio 4. Sukka, jonka sisässä anturit ovat integroituina



Kuvio 5. Havainnekuva sukan sisään integroitavien anturien sijainneista ja muodoista. Anturien sijainnit: A = ulkosyrjä, B = jalkapöytä, C = sisäsyrjä, D = päkiä.

Käyttötilanteessa sukan anturit keräävät niihin kohdistuvista voimista tietoa. Anturit reagoivat paine-eroon, joka syntyy pallon kosketuksesta sukan antureihin sekä käyttäjän askeleista. Anturit ovat kytketty sukkaan integroitujen johtimien kautta mikrokontrollerille, joka prosessoi antureilta tulevaa dataa ja pyrkii erottamaan datasta pallokosketukset sekä askeleet. Pallokosketusten ja askelten määrä tallennetaan mikrokontrollerin sisäiseen muistiin ja suorituksen jälkeen tämä data lähetetään Bluetooth-yhteyden kautta Android-laitteelle datan jatkoanalysointia ja visualisointia varten. Toinen käyttötilanne on, että käyttäjä pelaa Android-laitteella pallonhallintapeliä, jolloin sukka lähettää pallokosketusdataa reaaliajassa Android-laitteelle. Tässä opinnäytetyössä paneuduttiin laitekokonaisuuden ohjelmalliseen puoleen sekä laitekokonaisuuden tiedonkäsittelyyn ja langattomaan tietoyhteyteen. Opinnäytetyön Android-prototyypisovellus ohjelmoitiin pääasiassa pallonhallintapelin käyttötilannetta ajatellen.



Kuvio 6. Älysukan elektroniikkayksikkö

4 Algoritmit

4.1 Algoritmien tehtävä

Matematiikassa ja tietotekniikassa algoritmi on kuvaus sarjasta lauseita, jotka toteutetaan jonkin ongelman ratkaisemiseksi. Lauseet voivat olla esimerkiksi matemaattisia tai loogisia operaatioita. Tietotekniikassa ohjelmat muodostuvat algoritmeista ja tietorakenteista, joten algoritmia voidaan pitää koko ohjelman ytimenä. Algoritmin nopeus, tarkkuus, luotettavuus sekä ongelmanratkaisukyky vaikuttavat kaikki ohjelman suorituskyykyyn. (Kolehmainen 2006, 7–8.)

Tietokoneohjelmassa yhdellä algoritmilla ratkaistaan yksi ongelma ja ohjelmointivaiheessa on tärkeää kuvata ongelma siinä muodossa, jossa tietokone ymmärtää sen ja täten kykenee sen ratkaisemaan. Ennen algoritmin kehittämistä, on oleellista kuvata ratkaistava ongelma tarkasti. Ongelman kuvaaminen helpottaa ongelman hahmottamista ja sen osittelua pienempiin osaongelmiin. Kun monimutkaiselta tuntuva ongelma jäsenellään, esimerkiksi kaavion avulla, saadaan algoritmin kehittämislle luonteva pohja. Algoritmin kehittämistä tukee algoritmin simulointi manuaalisesti pienellä testiaineistolla, koska ensimmäinen algoritmiversio harvoin on aivan täydellinen. (Kolehmainen 2006, 7–8.)

Tietyn ongelman ratkaisemiseksi ei ole vain yhtä oikeaa tapaa, joten saman ongelman ratkaisuun kehitetyt algoritmit ovat harvoin täysin samanlaisia. Toinen algoritmi saattaa olla esimerkiksi nopeampi kuin toinen. Algoritmin tehokkuus saattaa vaikuttaa huomattavasti enemmän ohjelman suorituskyykyyn, kuin tietokonelaitteiston fyysiset ominaisuudet. (Kolehmainen 2006, 7–8; Cormen, Leiserson, Rivest & Stein 2001, 15–16.)

4.2 Algoritmit käytännön sovelluksissa

Algoritmit ovat monessa asiassa läsnä nykyaikana ja niiden avulla voidaan ratkaista hyvin monimutkaisia ongelmia. Käytännön sovelluksista on seuraavassa pari esimerkkiä:

- Vuonna 1990 aloitettiin The Human Genome Project -hanke, jonka tavoitteena oli tunnistaa ihmisen kaikki noin 20 500 geeniä, sekvensoida koko ihmisen DNA ja tallentaa kaikki tämä tieto tietokantoihin (Human Genome Project N.d.). Jokainen näistä vaiheista vaati hienostuneita algoritmeja. Tehokkaat algoritmit mahdollistivat tehokkaan resurssien käytön ja työ saatiin valmiiksi vuonna 2003.
- Maailman mullistanut internet mahdollistaa suurten tietomäärien välityksen ympäri maailmaa nopeasti. Tämä on mahdollista älykkäiden algoritmien takia, jotka käsittelevät internetin tietoverkossa liikkuvia massiivisia tietomääriä. Esimerkiksi internetin hakukoneiden taustalla on tehokas algoritmi. (Cormen ym. 2001, 10–11.)

5 WLAN-, Bluetooth- ja ZigBee -tekniikoiden vertailu akkukäyttöisissä laitteissa

WLAN, ZigBee ja Bluetooth ovat kaikki langattomaan tiedonsiirtoon käytettäviä tekniikoita, jotka on kuitenkin kehitetty eri käyttötarkoituksiin.

5.1 WLAN-tiedonsiirtotekniikka

IEEE 802.11 -standardin määritelmän mukaan WLAN on langaton lähiverkkotekniikka, joka on kehitetty pääasiassa korvaamaan tietokoneiden tiedonsiirto langallisesta Ethernet-lähiverkkoyhteydestä langattomaksi (IEEE

802.11). WLAN-tekniikan käyttö akkukäyttöisissä laitteissa on ollut rajoittunutta WLAN-tekniikan suhteellisen suuren virrankulutuksen vuoksi. Nykyisin uudet WLAN-piirit ovat kuitenkin virrankulutukseltaan huomattavasti vähävirtaisempia ja niiden virransäästötilat ovat kehittyneempiä kuin edeltäjänsä, mikä mahdollistaa akkukäyttöiset sovellukset ja jopa anturiverkostot. (Pärnänen 2011, 23.)

Bluetooth ja ZigBee ovat huomattavasti vähävirtaisempia vaihtoehtoja, kuin WLAN-tekniikka. Tästä syystä Bluetoothin tai ZigBeen käyttäminen akkukäyttöisissä laitteissa on perustellumpaa, vaikka näiden tekniikoiden tiedonsiirtonopeudet ovat huomattavasti WLANia pienempiä.

5.2 Bluetooth 4.0 -tiedonsiirtotekniikka

Bluetooth on avoin tiedonsiirtostandardi lyhyen kantaman langattomille yhteyksille, jossa laitteet viestivät keskenään radioaaltojen avulla.

Ensimmäinen Bluetooth-teknologiaa hyödyntävä laite ilmestyi markkinoille vuonna 2000. Vuonna 2015 Bluetooth-teknologiaa hyödyntäviä laitevalmistajia on ympäri maailmaa yli 27 000. Bluetooth-teknologiaa hyödynnetään suuressa määrin eri laitteissa, kuten puhelimissa, tietokoneissa ja urheiluteknologisissa tuotteissa. Tekniikan kantama soveltuu parhaiten lyhyille noin kymmenen metrin etäisyyksille, mutta myös pidemmät matkat ovat mahdollisia. (Bluetooth Technology Basics n.d.; Our History n.d.)

Bluetooth low energy on yksi Bluetooth 4.0 -version protokollista, jonka ominaisuuksiin kuuluu mm. matala virrankulutus, kyky operoida tavallisella nappiparistolla jopa vuosia ja pidempi kantomatka verrattuna vanhempiin Bluetooth-versioihin. Ensimmäinen laite, joka hyödynsi Bluetooth low energy -tekniikkaa, tuli markkinoille vuonna 2011. (Bluetooth Technology Basics n.d.)

5.3 ZigBee-tiedonsiirtotekniikka

ZigBee on suhteellisen tuore langattomien verkkojen standardi. Se standardisoitiin ensimmäisen kerran vuonna 2004. ZigBee sai alkunsa vuonna 1998, kun useat radiotekniikan insinöörit uskoivat WLAN- ja Bluetooth-yhteyksien olevan riittämättömiä kaikkiin langattomiin käyttösovelluksiin. ZigBee-teknologia on samankaltainen Bluetooth low energy -teknologian kanssa, mutta suurin ero tekniikoiden välillä on, että ZigBee-tekniikalla lähetetään dataa alhaisemmalla siirtonopeudella. (Sintonen 2012, 1-3.)

ZigBee on kehitetty lähettämään vähäisiä bittimääriä ja yksinkertaisia komentoja, kuten komento television käynnistämiseen tai oven lukitsemiseen. Alhainen siirtonopeus takaa huomattavasti alhaisemman virrankulutuksen verrattuna muihin langattomiin tiedonsiirtotapoihin, ja näin ollen ZigBeen virtalähteen kestoikä voidaan mitata vuosissa. Puhelimet eivät kuitenkaan tällä hetkellä tue ZigBee-yhteyttä, vaan tarvitsevat erillisen lähetin-vastaanottimen tiedonvälitykseen ZigBee-protokollan kautta. (Samosuyev 2010, 22.)

5.4 Vertailu

Taulukossa 1 on esitetty kolmen protokollan ominaisuuksien eroja. Taulukosta huomataan, että WLAN sopii hyvin suurten datamäärien siirtämiseen, koska sillä on paras hyötysuhde datan siirtämisessä (teho per bitti). WLAN kuitenkin kuluttaa suhteessa paljon virtaa kahteen muuhun standardiin verrattuna. Piikkivirta on oleellista huomioida, kun valitaan yhteystekniikkaa langattomalle laitteelle, jonka oleellisina vaatimuksina on alhainen virrankulutus ja pitkä akunkesto. Korkeat piikkivirrat saattavat vahingoittaa akkua ja lyhentää akunkestoja. (Smith 2011; Ligertwood & Povalac 2012.)

Taulukko 1. Langattomien standardien ominaisuuksia (Tiedot Smith 2011)

Standardi	WLAN	BLE	ZigBee
IEEE standardi	802.11a/b/g	802.15.1	802.15.4
Taajuus	2,4 GHz; 5 GHz	2,4 GHz	868/915 MHz; 2,4 GHz
Maks. tiedonsiirto- nopeus	54 Mb/s	1 Mb/s	250 kb/s
Kantama (teoreettinen maks.)	100 m	10 m	100 m
Teho per bitti	0,00525 μ W/bitti	0,153 μ W/bitti	185,9 μ W/bitti
Piikkivirta	116 mA	12,5 mA	30 mA

Bluetooth low energy ja ZigBee ovat ominaisuuksiltaan keskenään samankaltaisia, mutta olennaisena erona on, että lähestulkoon jokainen markkinoilla oleva mobiililaitte on varustettu Bluetooth-valmiudella. Näin ollen laitevalmistajat usein valitsevat Bluetoothin omien tuotteidensa langattomaksi tietoyhteydeksi, kun ollaan tekemisissä kuluttajamobiililaitteiden kanssa. Samasta syystä myös tämän työn langattomaksi tietoyhteystekniikaksi valikoitui Bluetooth low energy, sen alhaisen virrankulutuksen lisäksi. Tekniikan kantomatkalla ei ollut niin suurta painoarvoa, sillä käyttötilanteessa Bluetooth-vastaanotin ja -lähetin tulevat olemaan lähellä toisiaan.

6 Opinnäytetyön toteutus

Opinnäytetyön tarkoituksena oli suunnitella järjestelmä, joka prosessoi älytukin neljän paineanturin mittausdataa ja lähettää prosessoidun datan Bluetooth low energy -yhteydellä Android-puhelimeen. Koko mittausjärjestelmän tuli olla akkukäyttöinen, joten oli oleellista, että laitteiston virrankulutus oli pieni.

Paineanturit integroitiin jalkapallosukkaan, jolloin paineantureilla voitiin mitata jalkapallokosketusten ja -potkujen määrää ja tyyliä sekä askelten määrää ja askeltiheyttä. Jotta järjestelmä tunnistaisi pallokosketuksia ja askelluksia, täytyi mikrokontrolleriin ohjelmoida niitä tunnistavia algoritmeja.

6.1 Vaatimusten määrittely

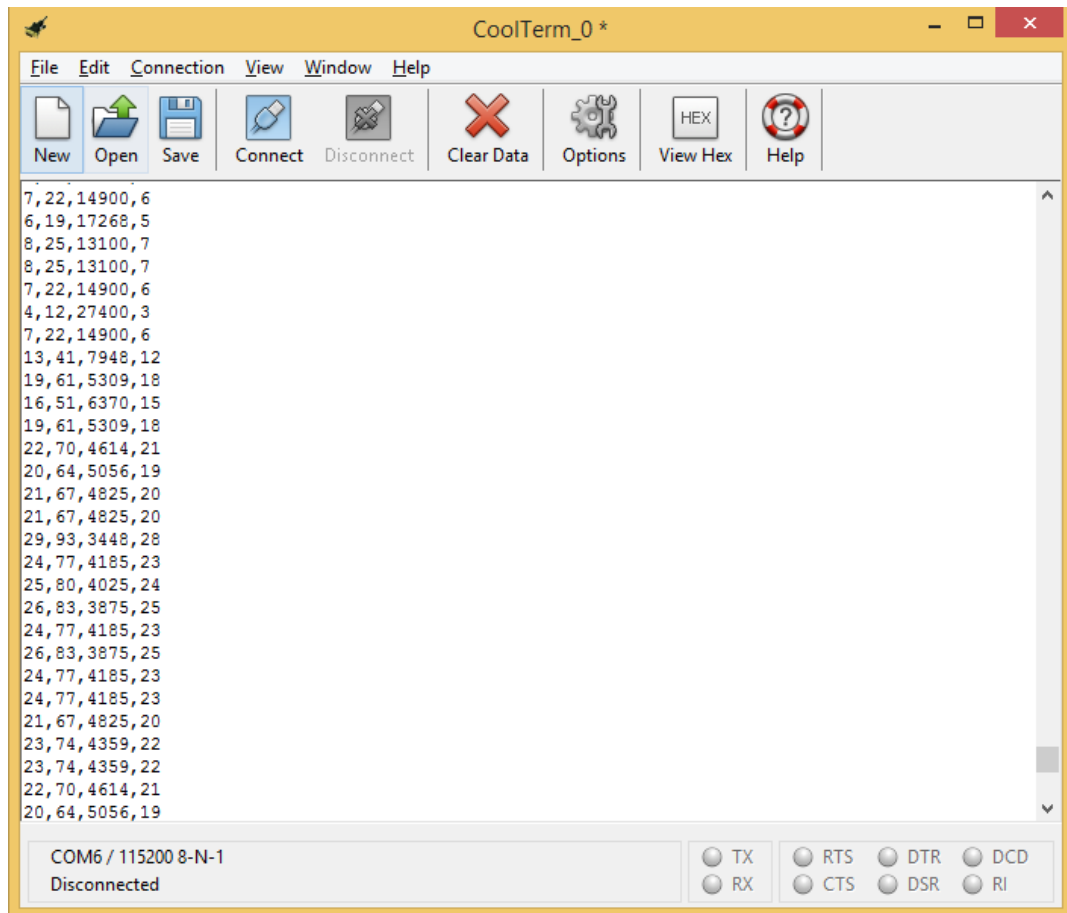
Jalkapallokosketuksia mitattaessa on oleellista huomioida pallokosketuksen aika, joka on keskimäärin 9 ms–16 ms, riippuen potkun voimakkuudesta ja potkaistavan pallon sisäilmanpaineesta (Ball 2008, 65). Koska yhden pallokosketuksen kontaktiaika saattaa olla alle kymmenen millisekuntia, täytyy tämä huomioida mikrokontrollerin näytteenottotaajuudessa. Aluksi dataa pyrittiin lähettämään raakadatana suoraan mikrokontrollerilta Android-puhelimeen noin 70 Hz:n näytteenottotaajuudella. Jos raakadataa yritettiin lähettää Android-puhelimeen tätä nopeammin, ei puhelin enää pysynyt mukana, vaan dataa hukkuu ja vääristyy vastaanottovaiheessa. 70 Hz:n näytteenottotaajuus ei kuitenkaan riitä jalkapallopotkujen tulkitsemiseen, sillä 70 Hz:n näytteenottotaajuudella mikrokontrolleri ottaa yhden näytteen noin 14 millisekunnin välein. Jos pallokosketus on esimerkiksi 10 millisekuntia, on mahdollista, että mikrokontrollerilta jää tämä kosketus kokonaan huomaamatta, koska se ei ota näytteitä tarpeeksi usein. Näiden ongelmien takia, data päädyttiin prosessoimaan suoraan mikrokontrollerissa, jolloin näytteenottotaajuus saatiin kasvatettua useaan sataan Hertziin.

Akun säästämisen kannalta on oleellista, että dataa lähetetään puhelimeen vain silloin, jos pallokosketus tai askel toteutuu. Suurten raakadatamäärien reaaliaikainen lähettäminen langattomasti puhelimeen prosessoitavaksi vaatii paljon virtaa akkukäyttöiseltä lähetinyksiköltä. Jos ajatellaan sukan käyttöä jalkapallossa pelitilanteessa, on akunkäytön kannalta järkevintä tallentaa data

mikrokontrollerin sisäiseen muistiin ja lähettää prosessoitu data vasta suorituksen jälkeen Android-sovellukselle.

6.2 Algoritmien luominen mikro-ohjaimelle

Algoritmien tarkoituksena työssä oli suodattaa paineantureiden raakadatasta käyttäjän jalkapallokosketukset sekä askeleet. Jotta tiedettäisiin, miten algoritmien tulisi käsitellä dataa, tuli aluksi tallentaa anturien raakadataa laitteen käyttötilanteesta ja analysoida, kuinka anturidata käyttäytyy eri tilanteissa. Mikrokontrolleri ohjelmoitiin Arduino-kielellä, joka pohjautuu C-ohjelmointikieleen. Raakadatan tallentamiseen käytettiin CoolTerm-nimistä terminaaliohjelmaa (ks. kuvio 7), jolla voitiin lukea tietokoneen sarjaporttiin tulevaa dataa. Mikrokontrolleri yhdistettiin Bluetooth-yhteydellä tietokoneeseen, jolloin CoolTerm-ohjelmalla voitiin tallentaa raakadataa suoraan tekstitiedostoon, joka taas voitiin tuoda Excel-taulukkoon jatkotulkitsemista varten.



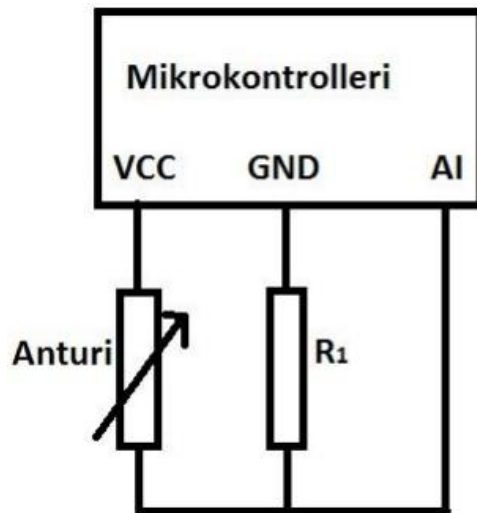
Kuvio 7. Näkymä CoolTerm-ohjelmasta

Yhtenä ongelmana algoritmien toimintavarmuuden kannalta on käyttäjän juoksuaskeleista aiheutuva kohina anturidataan. Juoksun aikana sukan anturit puristuvat kengän sisäreunoja vasten niin että antureihin kohdistuu jatkuvasti muuttuvaa painetta. Algoritmit ohjelmoitiin siten, että ne pystyvät erottamaan pallokosketukset ja askeleet muun ylimääräisen anturidatan seasta.

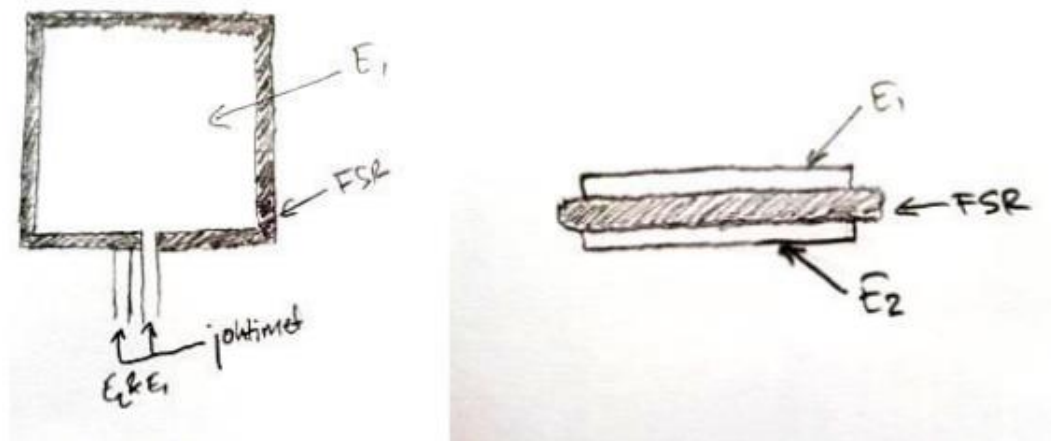
6.3 Datan linearisointi ja mittausalueen kalibrointi

Mittausjärjestelmän paineanturit perustuvat paineanturin resistanssin muutoksen havaitsemiseen (ks. kuvio 8). Mitä suurempi paine kohdistuu anturiin, sitä pienempi anturin sisäinen resistanssi on. Mitä pienempi paine anturiin kohdistuu, sitä suurempi anturin sisäinen resistanssi on. Paineanturit

valmistettiin asettamalla puolijohtava muovikalvo kahden sähköä johtavan johdinosan väliin (ks. kuvio 9).



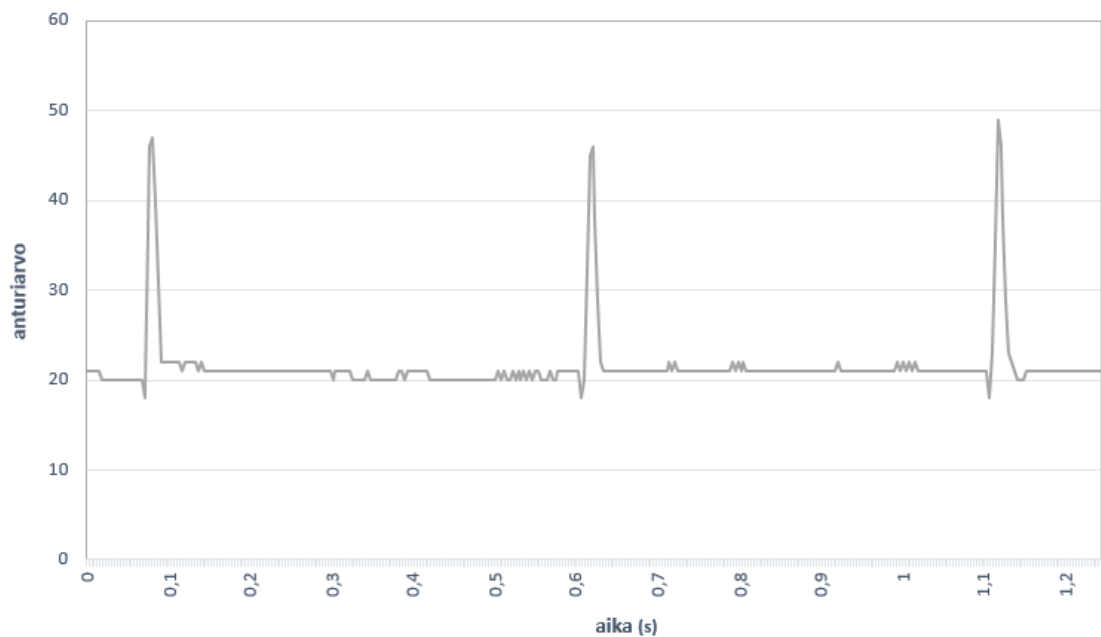
Kuvio 8. Anturin ja mikrokontrollerin kytkentäkaavio (Thitz 2015)



Kuvio 9. Havainnekuva paineanturin rakenteesta (Thitz 2015)

Kuviosta 10 nähdään, että anturin pohja-arvo on noin 20, kun älysuikka on asetettuna käyttäjän jalkaan ja on jalkapallokengän sisässä. Tämä tarkoittaa, että jalkapallokengän aiheuttama puristus puristaa paineanturia jo 20 arvon verran, jos anturi on skaalattu 0 ja 100 arvon välille. Tässä mittauksessa

jalkapallokengän aiheuttamasta puristuksesta ei ole haittaa, mutta käyttäjillä on erikokoisia kenkiä ja he kiristävät kengännauhat eri tiukkuudelle, joten tämä tulee huomioida mittausjärjestelmässä. Antureiden skaalan tulee olla dynaaminen eli riittävä, riippumatta siitä, kuinka kovaa kenkä puristaa antureita vasten. Tätä varten ohjelmaan luotiin kalibrointiominaisuus.



Kuvio 10. Jalkapöytäanturin raakadataa ajan funktiona 285 Hz:n näytteenottotaajuudella

Seuraavaksi on esitelty osa mikrokontrollerin ohjelmakoodia, joka vastaa anturidatan linearisoinnista sekä mittausalueen kalibroinnista.

1. sensorReading-muuttujaan luetaan A/D-muuntimelta anturin arvo analogRead-komennolla:

```
int sensorReading = analogRead(Apin);
```

sensorVoltage-muuttujaan syötetään millivolteiksi skaalattu anturiarvo. A/D-muuntimelta tuleva arvo on arvojen 0 ja 1023 välillä,

mutta skaalaamalla arvot map-komennolla 0:n ja 3300:n välille saadaan anturiarvo muutettua millivolteiksi. Koska mittauksessa käytetään 3,3 voltin käyttöjännitteellä toimivaa mikrokontrolleria, mikrokontrollerin analogisisäntuloon on sallittua syöttää jännitearvoja vain 0:n ja 3,3 voltin väliltä:

```
int sensorVoltage = map(sensorReading, 0, 1023, 0, 3300);
```

2. sensorResistance-muuttujaan ratkaistaan anturin sisäisen resistanssin arvo:

```
int sensorResistance = 3300 - sensorVoltage;
sensorResistance *= 1000;
sensorResistance /= sensorVoltage;
```

Tämä perustuu fysikaaliseen kaavaan

$$R_{anturi} = \frac{(U_{cc} - U_{anturi}) \cdot R_{vastus}}{U_{anturi}}$$

jossa R_{anturi} on anturin sisäinen vastus (Ω), U_{cc} on mikrokontrollerin käyttöjännite (V), U_{anturi} on anturin sisäinen jännite (V) ja R_{vastus} on jännitteenjakajan vastus (Ω).

3. sensorConductance-muuttujaan ratkaistaan anturin konduktanssi mikrosiemensinä:

```
int sensorConductance = 1000000;
sensorConductance /= sensorResistance;
```

4. Tämä perustuu fysikaaliseen kaavaan

$$G_{anturi} = \frac{1000000}{R_{anturi}}$$

jossa G_{anturi} on anturin sisäinen konduktanssi (S) ja R_{anturi} on anturin sisäinen vastus (Ω).

5. sensorConductance-muuttuja skaalataan arvojen 0 ja 100 välille map-komennolla. Constrain-komennolla muuttujan arvot rajataan 0:n ja 100:n välille. Kaikki 0:aa pienemmät arvot muutetaan 0:ksi ja kaikki 100:aa suuremmat arvot muutetaan 100:ksi. calibrationValue-muuttujan avulla kalibroidaan anturin arvoskaala, niin että käyttäjän komentassa laitteen kalibroinnin calibrationValue-muuttujaan syötetään sen hetkinen anturiarvo, joka on skaalattu 0:n ja 100:n välille, ja asetetaan tämä arvo uudeksi nolla-arvoksi. calibrationValue-muuttuja pitää huolen, että anturi on kalibroinnin jälkeen skaalattu siten että anturiin kohdistuva aloituspaine vastaa arvoa 0 ja anturin yläraja on 100 arvoa siitä ylöspäin:

```

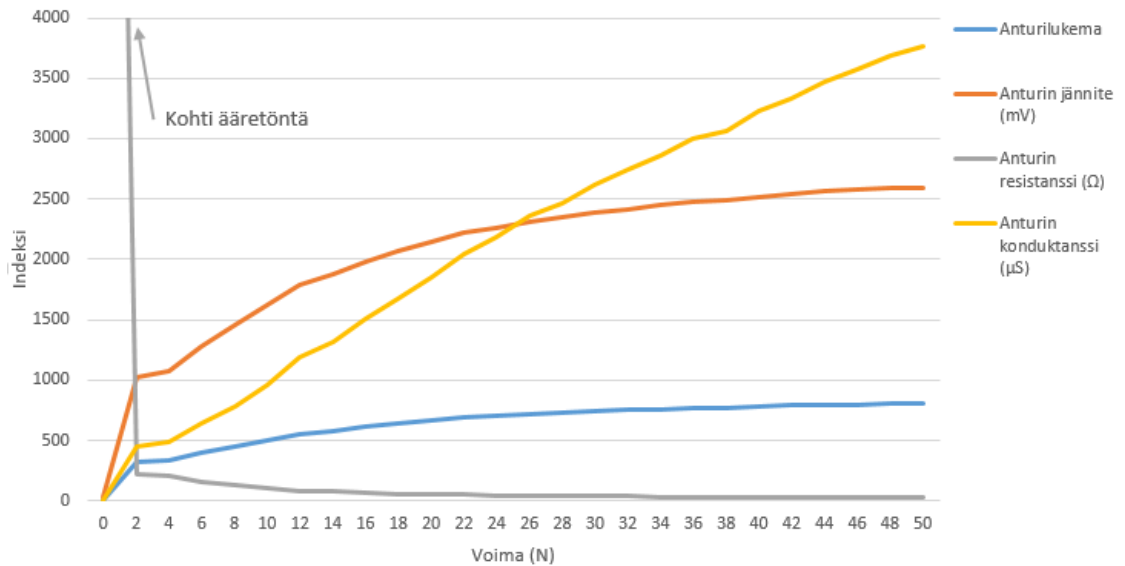
sensorConductance =
constrain(map(sensorConductance,0,20000,0,
(100 + calibrationValue)),0,(100 + calibrationValue));

int Aint = map(sensorConductance, calibrationValue,
(100 + calibrationValue), 0,100);

Aint = constrain(Aint,0,100);

```

6. Aint on lopulta muuttuja, joka pitää sisällään anturin skaalatun ja kalibroidun arvon.



Kuvio 11. Muuttujien arvot voiman funktiona

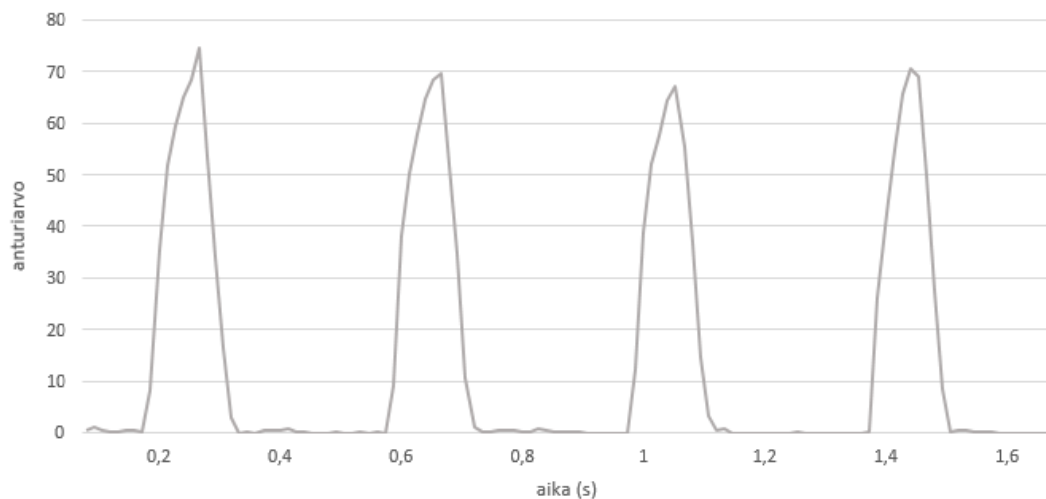
Kuviosta 11 nähdään, kuinka anturin konduktanssi on suhteellisen lineaarinen voiman funktiona. Anturin data linearisoidaan, jotta anturiin vaikuttava voima saadaan lineaarisesti vastaamaan anturin lukemaa. Voimavaikutuksen selventämiseksi anturit skaalataan arvojen 0 ja 100 välille.

6.4 Pallokosketusten tunnistaminen

Paineanturit sekä mikrokontrolleri- ja BLE-lähetinyksikkö kiinnitettiin jalkapallosukkaan pienessä kotelossa, minkä jälkeen voitiin tallentaa paineantureiden raakadataa jalkapallon potkusuorituksesta. Kuviossa 10 (s. 24) on jalkapöytäpaineanturin raakadatan kuvaaja siihen kohdistuneesta kolmesta pallonpomputtelukosketuksesta. Kuvaajasta nähdään, että pallokosketus näkyy selkeänä terävänä piikkinä, joten se on suhteellisen helppo tulkita algoritmin avulla (ks. liite 1). Yhden pallokosketuksen kontaktiaika oli tässä mittauksessa 16–18 millisekuntia. Anturi oli skaalattu lineaarisesti arvojen 0 ja 100 välille.

6.5 Askeleen tunnistaminen

Yksi paineanturi asetettiin sukkaan siten, että se oli käyttäjän päkiän alla. Käytännössä aina kävely- tai juoksuaskelta otettaessa käyttäjän päkiä osuu askeleen aikana maahan, joten pelkkä päkiäanturi oli riittävä askeleen tunnistamiseen. Askeleen anturidatasta (ks. kuvio 12) voisi selvittää myös mm. askeltiheyden ja askelkontaktiajan. Liitteessä 2 on esitetty askeltunnistusalgoritmin toiminta vuokaaviona.



Kuvio 12. Raakadataa oikean jalan juoksuaskeleista 285 Hz:n näytteenottotaajuudella päkiäanturilta. Askeltiheys mittauksessa oli noin 180 askelta per minuutti.

6.6 Bluetooth low energy -yhteyden luominen

6.6.1 Bluetooth low energy -yhteyden valinta

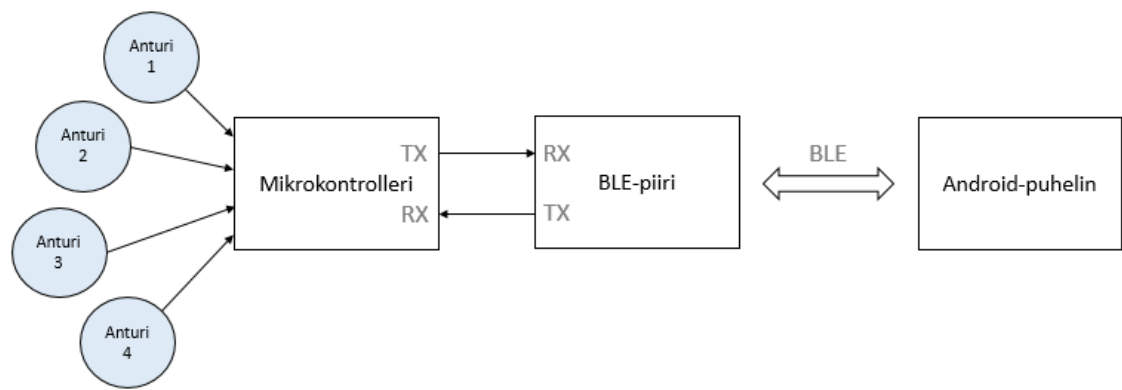
Työssä käytettiin Bluetooth low energy (BLE) -yhteyttä tiedon lähettämiseen ja vastaanottamiseen. Bluetooth low energy valittiin yhteystavaksi, koska lähestulkoon jokaisessa uudessa älypuhelinmallissa on sisäänrakennettu Bluetooth low energy -yhteensopivuus. Tuoteidean kehityksen jatkon

kannalta on oleellista, että tiedonsiirtotekniikka mittalaitteen ja eri puhelinmallien välillä on laajalti tuettu.

BLE-teknikka on myös vähäinen virrankulutukseltaan, mikä on ensiarvoisen tärkeää akkukäyttöisen mittalaitteen käyttäjäkokemuksen kannalta.

Mittausjärjestelmän virtalähteenä toimi pieni 3,7 V:n, 105 mAh:n akku.

Mittausjärjestelmä on havainnollistettu kuviossa 13.



Kuvio 13. Jalkapalloölysukan mittausjärjestelmä

6.6.2 Tiedon lähettäminen ja vastaanottaminen BLE-moduulilla

Alhaisen virrankulutuksen kannalta on oleellista pyrkiä lähettämään mahdollisimman vähän tietoa mikrokontrollerilta Android-puhelimeen. Kun algoritmit tunnistavat pallokosketuksen tai askeleen, BLE-moduuli lähettää kosketustyyppiä vastaavan merkkijonon Android-puhelimeen (ks. taulukko 2).

Taulukko 2. BLE-moduulin lähettämä merkkijono kosketustyyppistä riippuen

Kosketustyyppi	Lähetettävä merkkijono
Sisäsyryjäkosketus	#i+
Jalkapöytäkosketus	#t+
Ulkosyryjäkosketus	#o+
Askel	#s+

Merkkijonon #-merkin ja +-merkin tarkoituksena on toimia tunnistemerkkeinä Android-puhelimelle. Tunnistemerkkien tarkoitus on suodattaa virheellisiä lähetyksiä pois ja varmistaa, että sisään tuleva data on sitä mitä sovellus odottaa.

Mittalaitteen kalibrointia varten sovellus lähettää sitä vastaavan komennon BLE-moduulille. Vastaanottaessaan tämän komennon mikrokontrolleri kalibroi mittalaitteen.

6.7 Android-sovelluksen ohjelmointi

Opinnäytetyössä prototyypisovellus ohjelmoitiin Android-alustalle Android 4.4 KitKat -käyttöjärjestelmää käyttäen. Työssä oleellisena vaatimuksena oleva BLE-yhteys on tuettuna Android-alustalla versioista 4.3 ylöspäin. Sovellus ohjelmoitiin Android Studio -kehitysympäristöä käyttäen. Työssä Android-sovelluksen pääasiallinen tarkoitus oli vastaanottaa ja visualisoida mittalaitteen lähettämä data, toistaiseksi vain reaaliajassa.

6.7.1 Android-käyttöjärjestelmä

Android Inc. perustettiin Kalifornian Palo Altossa vuonna 2003 Andy Rubinin toimesta. Vuonna 2005 Google osti yhtiön ja ensimmäinen Android-käyttöjärjestelmää käyttävä älypuhelin tuli markkinoille vuoden 2008 loppupuolella. Android-käyttöjärjestelmän kehityksestä vastaa Open Handset

Alliance, joka on 84 yrityksen yhteenliittymä, joka kehittää kännykkäalan avoimia standardeja. Yleisellä standardilla pyritään siihen, että sama sovellus toimii kaikilla puhelimilla. Tällöin samaa sovellusta ei tarvitse kehittää yksitellen jokaiselle erimalliselle puhelimelle. (Halkola 2014, 3) Vuoden 2010 lopulla Android syrjäytti johtavan Nokian Symbian-käyttöjärjestelmän laitteiden myyntimäärissä (Kobie 2011).

Android-sovellukset kirjoitetaan useimmiten Java-ohjelmointikielellä. Android ei kuitenkaan tue suoraan Java-kieltä, vaan Androidin oma Dalvik-virtuaalikone esikäntää Java-luokat Dalvik executables -tiedostomuotoon (.dex). (Korkiakoski 2013, 13.) Androidin kehitysympäristö on avoin, joten sen kehittäminen ja käyttäminen on täysin ilmaista kaikille, ja kehittäjillä on tämän vuoksi vapaa pääsy nk. ydinohjelmien käyttämiin rajapintoihin. Ydinohjelmia ovat mm. tekstiviestit, puhelinluettelo ja muut laitteissa vakiona mukana tulevat sovellukset. Kun kaikille ohjelmille on yhteiset rajapinnat, avaa tämä mahdollisuudet kolmannen osapuolen kehittäjille luoda rikkaita ja monipuolisia sovelluksia. Ohjelmoija pääsee käsiksi mm. puhelimen datayhteyksiin, sijaintitietoihin ja sisäiseen laitteistoon, joka usein sisältää kiihtyvyyssanturin, gyroskoopin sekä läheisyysanturin. Joissain sovelluksissa hyödynnetään näitä antureita ja esimerkiksi käyttäjän puhuessa puhelimeen kännykän näyttö sammuu läheisyysanturin ansiosta säästäten akkua. (Juola 2011, 16; Halkola 2014, 3.)

6.7.2 Komponentit ja aktiviteetit

Androidin sovelluslogiikkaan kuuluu neljä komponenttia: aktiviteetti (Activity), palvelu (Service), sisällöntarjoaja (Content provider) ja vastaanottaja (Broadcast receiver).

Aktiviteetti periytetään Activity-luokasta. Aktiviteetin tehtävä on näyttää sovelluksessa yksi näkymä ja näyttää käyttöliittymän näkymä käyttäjälle. Tässä näkymässä käyttäjä voi tehdä erilaisia toimintoja, kuten ottaa kuvia, lähettää viestejä tai soittaa puheluita. Aktiviteetti on usein koko näyttöalueen peittävä ikkuna, mutta se voi olla myös esimerkiksi pienempi kelluva ikkuna toisen ikkunan päällä. Vain yksi aktiviteetti-ikkuna voi kerrallaan reagoida käyttäjän syötteisiin. (Activities N.d.; Korkiakoski 2013, 15.)

Palvelu-komponentti suoritetaan taustalla, joten se ei ole suoraan vaikutuksessa käyttäjän kanssa. Sovellus voi palvelu-komponentilla komentaa järjestelmää suorittamaan tehtävän taustalla, myös silloin kun käyttäjä ei ole suorassa vuorovaikutuksessa sovelluksen kanssa. Sovellus voi palvelu-komponentilla hallita taustalla esimerkiksi Bluetooth-yhteyttä ulkopuoliseen laitteeseen tai toistaa musiikkia. Palvelu-komponentti periytetään Service-luokasta. (Services N.d.)

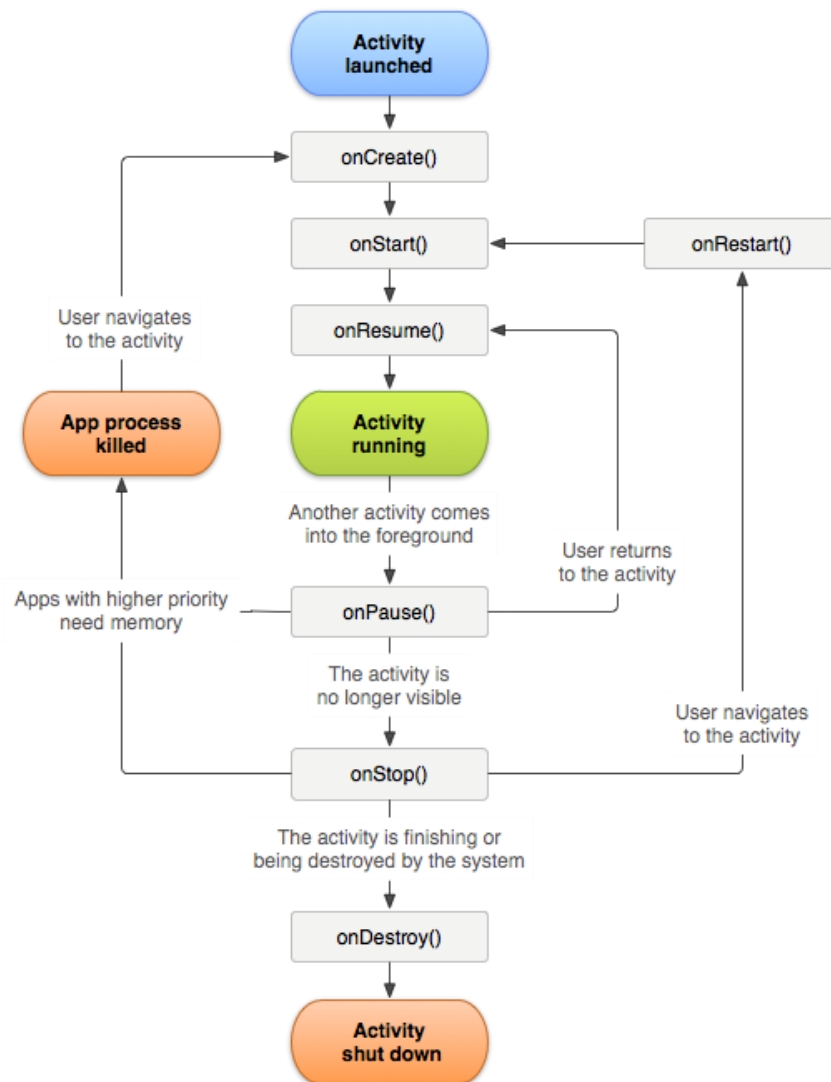
Sisällöntarjoaja-komponentin avulla voidaan jakaa tietoa sovellusten välillä ja sen avulla sovellus voi myös vastaanottaa ja lähettää tietoa erilaisten tietokantojen ja tiedostojen välillä. (Content Providers N.d.)

Vastaanottaja-komponentilla sovelluksen voi komentaa rekisteröimään halutut sovelluksissa tai käyttöjärjestelmässä tapahtuvat tapahtumat, jolloin sovellus voidaan ohjelmoida toimimaan näiden tapahtumien mukaan. Vastaanottaja-komponenttia luodessa tulee määrittää mitä tapahtumia sen halutaan kuuntelevan. (BroadcastReceiver N.d.)

Aktiviteetti voidaan jakaa neljään erilaiseen tilaan:

1. Jos aktiviteetti on koko näytöllä näkyvässä (pakan päällimmäisenä), on aktiviteetti aktiivinen.

2. Jos aktiviteetti on sumennettuna koko näytöllä, mutta on silti näkyvässä, on aktiviteetti keskeytetty, jolloin aktiviteetti säilyttää kaiken tietonsa muistissa.
3. Jos toinen aktiviteetti peittää aktiviteetin kokonaan, aktiviteetti on pysäytetty. Tällöinkin aktiviteetti säilyttää tietonsa muistissa, mutta järjestelmä sulkee aktiviteetin kokonaan, jos muistia tarvitaan muualla. Jos järjestelmä tarvitsee lisää muistia, voi järjestelmä sulkea kyseisen sovelluksen vapauttaakseen muistia muihin toimintoihin. Jos järjestelmä sulkee sovelluksen, käyttäjän palatessa kyseiseen sovellukseen, sovellus kutsuu metodia `onCreate(Bundle)` ja ajaa sovelluksen alusta asti uudelleen.
4. Tyhjä prosessi on prosessi, joka ei sisällä aktiviteetteja tai muita sovelluskomponentteja. Järjestelmä sulkee tyhjät prosessit nopeasti, kun järjestelmälle tulee tarve lisämuistille. Tämän vuoksi, jos sovelluksen tulee tehdä toimintoja, myös silloin kun se ei ole käyttäjälle näkyvässä, tulee nämä toiminnot suorittaa esimerkiksi palvelukomponentissa, jotta järjestelmä osaa olla tuhoamatta sovellusta. (Activities N.d.; Korkiakoski 2013, 15.)



Kuvio 14. Aktiviteetin elinkaari ja siirtymävaiheiden metodit (Android Developers N.d.)

Kuviossa 14 on esitetty aktiviteetin koko elinkaari. Suorakulmiot esittävät eri metodeja ja ovaalin muotoiset kuviot havainnollistavat aktiviteetin erilaisia tiloja. Taulukossa 3 on esitetty aktiviteetin siirtymävaiheiden metodien toiminta.

Taulukko 3. Aktiviteetin siirtymävaiheiden metodien toiminta

Metodi	Metodin toiminta	Tuhottavissa?	Seuraava metodi
<code>onCreate()</code>	Metodia kutsutaan, kun aktiviteetti luodaan. Tässä alustetaan mm. käyttöliittymänäkymä. Seuraava metodi on aina <code>onStart()</code> .	Ei	<code>onStart()</code>
<code>onRestart()</code>	Metodia kutsutaan, kun aktiviteetti on pysäytetty, ennen kuin aktiviteetti käynnistetään uudelleen. Seuraava metodi on <code>onStart()</code> .	Ei	<code>onStart()</code>
<code>onStart()</code>	Kutsutaan, kun aktiviteetti on tulossa käyttäjälle näkyväksi. Seuraava metodi on <code>onResume()</code> , jos aktiviteetti tulee näkyväksi tai <code>onStop()</code> , jos aktiviteetti pysäytetään.	Ei	<code>onResume()</code> tai <code>onStop()</code>
<code>onResume()</code>	Kutsutaan, kun aktiviteetti alkaa vuorovaikuttamaan käyttäjän kanssa. Tässä metodissa aktiviteetti on aktiviteettipakan päällimmäisenä ja kuuntelee käyttäjän syötteitä.	Ei	<code>onPause()</code>
<code>onPause()</code>	Kutsutaan, kun järjestelmä on alkamassa käynnistää aiempaa aktiviteettia. Metodissa pysäytetään usein mm. animaatiot ja muuta, joka vie järjestelmältä paljon muistia. Metodien implementoinnit täytyy olla nopeita, sillä seuraava aktiviteetti ei käynnisty, ennen kuin tämä on suoritettu loppuun. Seuraava metodi on joko <code>onResume()</code> , jos aktiviteetti palautuu takaisin päällimmäiseksi, tai <code>onStop()</code> , jos aktiviteetti siirtyy käyttäjälle näkymättömäksi.	Ei	<code>onResume()</code> tai <code>onStop()</code>
<code>onStop()</code>	Kutsutaan silloin, kun aktiviteetti ei ole enää näkyvässä käyttäjälle, koska toista aktiviteettia on jatkettu, joka peittää tämän aktiviteetin alleen. Tämä tapahtuu silloin, jos uusi aktiviteetti on käynnistymässä, olemassa oleva aktiviteetti tuodaan tämän aktiviteetin eteen tai jos tämä aktiviteetti tuhoetaan.	Kyllä	<code>onRestart()</code> tai <code>onDestroy()</code>
<code>onDestroy()</code>	Viimeinen kutsu, jota kutsutaan ennen kuin aktiviteetti tuhoetaan. Tämä tapahtuu silloin, jos aktiviteetti on lopettamassa tai järjestelmä tuhoaa tämän aktiviteetin säästääkseen tilaa.	Kyllä	<i>ei mitään</i>

6.7.3 Bluetooth low energy -yhteys Android-käyttöjärjestelmässä

Jotta Android-sovelluksella voidaan käyttää Android-laitteen Bluetooth low energy -yhteysominaisuuksia, tulee sovelluksen lähdekoodissa määrittää mitä yhteysominaisuuksia halutaan käytettävän ja miten.

Sovelluksen lähdekoodin AndroidManifest.xml-tiedostoon määritellään mitä käyttöoikeuksia sovellus tarvitsee. Nämä oikeudet ovat pakollisia Bluetooth-yhteyden kannalta. Manifestitiedosto ohjelmoidaan XML-ohjelmointikielellä.

Uses-feature-elementin määrittelyllä varmistetaan, että laitteesta löytyy Bluetooth low energy -moduuli. Uses-permission-elementin määrittelyllä varmistetaan, että sovellus pyytää itselleen oikeutta käyttää laitteen Bluetooth-ominaisuuksia:

```
<uses-feature android:name="android.hardware.bluetooth_le"
android:required="true"/>
<uses-permission android:name=
"android.permission.BLUETOOTH"/>
<uses-permission android:name=
"android.permission.BLUETOOTH_ADMIN"/>
```

Näihin käyttöoikeuksiin sovellus pyytää käyttäjältä lupaa sovelluksen asennusvaiheessa.

Kun käyttäjä haluaa yhdistää sovelluksen Bluetooth-piiriin, sovellus aloittaa Bluetooth-laitteiden skannauksen startLeScan()-metodilla, jossa mBluetoothAdapter-objekti on Android-laitteen Bluetooth-adapteriobjekti. Tämä käynnistää LeScanCallback-takaisinkutsuluokan onLeScan()-metodin:

```

private BluetoothAdapter.LeScanCallback mLeScanCallback =
new BluetoothAdapter.LeScanCallback() {
    @Override
    public void onLeScan(final BluetoothDevice device, int
rssi, byte[] scanRecord) {
        runOnUiThread(new Runnable() {
            @Override
            public void run() {
                mLeDeviceListAdapter.addDevice(device);
                mLeDeviceListAdapter.notifyDataSetChanged();
            }
        });
    }
};

```

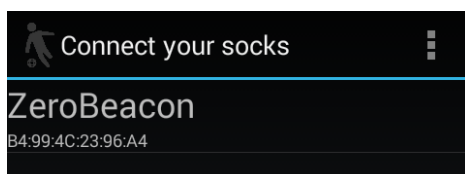
Metodi onLeScan käynnistää omalle säikeelleen metodin, joka listaa skannauksessa löydetyt Bluetooth-laitteet BluetoothDevice-muuttujaan. Device-muuttujasta voidaan kysyä tietoja Bluetooth-laitteesta String-tyyppiseen muuttujaan:

```

String mDeviceName = device.getName();
String mDeviceAddress = device.getAddress();

```

String-muuttujan sisältö voidaan asettaa laitteen näytölle näyttämään mm. löydettyjen Bluetooth-laitteiden nimet ja MAC-osoitteet (ks. kuvio 15).



Kuvio 15. Kuvakaappaus sovelluksen laiteskannausnäkyvästä

Kun käyttäjä valitsee näytöllä olevasta Bluetooth-laitelistasta laitteen, johon haluaa yhdistää, tallentaa sovellus muistiin kyseisen Bluetooth-laitteen nimen ja MAC-osoitteen ja lähettää nämä tiedot seuraavaksi avattavaan sovelluksen aktiviteettiin Intentin avulla:


```
final Intent intent = new Intent(this, MainActivity.class);
intent.putExtra(MainActivity.EXTRAS_DEVICE_NAME,
device.getName());
intent.putExtra(MainActivity.EXTRAS_DEVICE_ADDRESS,
device.getAddress());
startActivity(intent);
```

Sovellus yhdistää Bluetooth-laitteeseen tämän yksilöllisen MAC-osoitteen perusteella komennolla `mBluetoothLeService.connect(mDeviceAddress)`, jossa `mDeviceAddress`-muuttuja sisältää edellisessä aktiviteetissa saadun MAC-osoitteen. `mBluetoothLeService` on `BluetoothLeService`-luokan objekti. `BluetoothLeService`-luokka ylläpitää ja käsittelee ohjelman taustalla Bluetooth-yhteyttä.

Vaikka Bluetooth-laite ja Android-laite on nyt yhdistetty keskenään, ei niiden välillä voi vielä vapaasti lähettää tietoa. Sovellukseen on määritetty tietty Generic Attribute Profile -profiili eli GATT-profiili, joka on yleinen spesifikaatio datan lähettämiseen BLE-yhteyden kautta. Jotta laitteet voivat vuorovaikuttaa keskenään, tulee Android-laite yhdistää BLE-laitteen GATT-serveriin `connectGatt()`-metodilla:

```
mBluetoothGatt = device.connectGatt(this, false,
mGattCallback);
```

`connectGatt()`-metodi käyttää kolmea parametria, jotka ovat `Context`-objekti, `autoConnect`-parametri (boolean-tyyppinen muuttuja, joka määrittää yhdistääkö sovellus BLE-laitteeseen automaattisesti vai ei, kun BLE-laite tulee käytettäväksi Android-laitteen havaintoalueella) ja referenssi `BluetoothGattCallback`-luokkaan. `connectGatt()`-metodi yhdistää Android-sovelluksen BLE-laitteen GATT-palvelimeen (server), jolloin Android-sovellus toimii GATT-asiakkaana (client).

Kun sovellus on yhdistänyt BLE-laitteen GATT-palvelimeen, voivat laitteet lähettää ja vastaanottaa tietoa keskenään tuettujen attribuuttien kautta, jotka ovat GATT-palveluita ja -characteristicseja. Sovellus iteroi läpi tuetut GATT-palvelut ja -characteristicsit ja jos sovellus tukee tiettyä GATT-palvelua ja -characteristica, voidaan niiden kautta lähettää tietoa sovelluksen ja BLE-laitteen välillä. Jotta sovellus tietää milloin BLE-laite lähettää dataa sovellukselle, käytetään setCharacteristicNotification()-metodia, jolla sovellus asetetaan reagoimaan, kun sille lähetetään tietoa:

```
public void setCharacteristicNotification(
BluetoothGattCharacteristic characteristic, boolean enabled)
{
    if (mBluetoothAdapter == null || mBluetoothGatt == null)
        {return;}
    mBluetoothGatt.setCharacteristicNotification(
characteristic, enabled);
    if (UUID_HM_RX_TX.equals(characteristic.getUuid())) {
        BluetoothGattDescriptor descriptor =
characteristic.getDescriptor(
            UUID.fromString(GattAttributes.CLIENT_CHA-
RACTERISTIC_CONFIG));
        descriptor.setValue(
BluetoothGattDescriptor.ENABLE_NOTIFICATION_VALUE);
        mBluetoothGatt.writeDescriptor(descriptor);
    }
}
```

BluetoothLeService-luokka käsittelee sovellukseen sisään tulevaa ja ulos lähtevää dataa. broadcastUpdate()-metodi kuuntelee muutoksia BluetoothGattCharacteristic-muuttujassa ja lähettää uuden muuttujan arvon sendBroadcast()-metodin avulla MainActivity-luokkaan, joka käsittelee graafista käyttöliittymäsäiettä:

```

private void broadcastUpdate(final String action, final
BluetoothGattCharacteristic characteristic) {
    final Intent intent = new Intent(action);
    final byte[] data = characteristic.getValue();
    if (data != null && data.length > 0) {
        final StringBuilder stringBuilder = new
StringBuilder(data.length);
        for (byte byteChar : data)
            stringBuilder.append(String.format("/x%02X
*/132", byteChar));
        intent.putExtra(EXTRA_DATA, String.format("%s", new
String(data)));
        sendBroadcast(intent);
    }
}

```

Lähetetty tieto vastaanotetaan MainActivity-luokassa broadcastReceiver-vastaanottajalla. Vastaanottajakomponentti asetetaan kuuntelemaan onko sisään tulevaa dataa tarjolla:

```

private final BroadcastReceiver mGattUpdateReceiver = new
BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        final String action = intent.getAction();
        if (BluetoothLeService.ACTION_DATA_AVAILABLE.equals(
action)) {
            displayData(intent.getStringExtra(
mBluetoothLeService.EXTRA_DATA));
        }
    }
};

```

Jos sisään tulevaa dataa on tarjolla, lähetetään data displayData()-metodille, jossa selvitetään vastaanotetun tietopaketin sisältö. Tietopaketissa on kirjain s, i, t tai o, riippuen millä jalanosalla pallokosketus sukkaan on tapahtunut vai onko kyseessä askel (ks. taulukko 2, s. 30). Jokaiselle eri kosketustyyppille on oma laskuri, joka laskee suorituksen aikana tapahtuneet kosketukset ja askeleet:

```

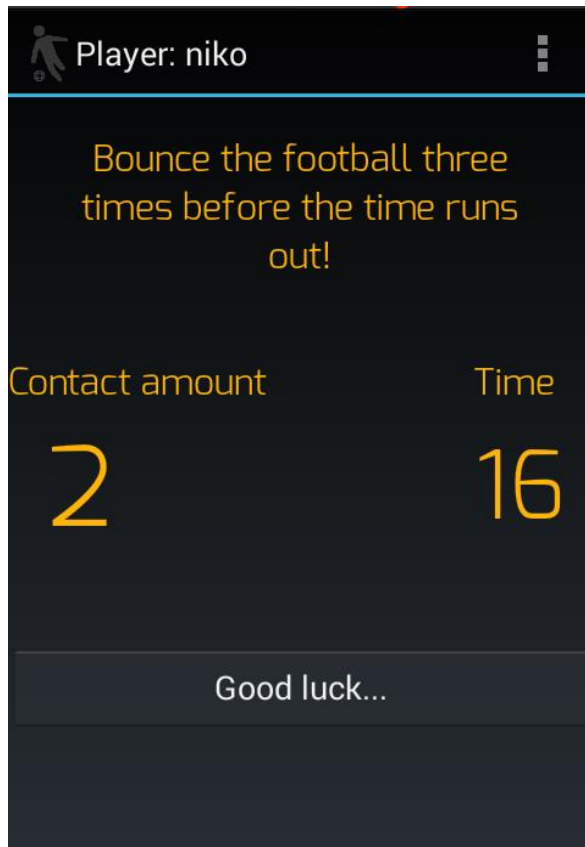
private void displayData(String data) {
    if (mIsStart) {
        String readMessage = data;
        recDataString.append(readMessage);
        int endOfLineIndex = recDataString.indexOf("~");
        if (endOfLineIndex > 0) {
            String dataInPrint = recDataString.substring(
0, endOfLineIndex);
            int dataLength = dataInPrint.length();
            if (dataInPrint.contains("#")){
                if (dataLength == 2){
                    inString =
recDataString.substring(1, 2);
                    if(inString.contains("s"))
                    {
                        stepCounterRC++;
                        stepCounterR.setText(
String.valueOf(stepCounterRC));
                    }

                    if(inString.contains("i"))
                    {
                        insideFootRC++;
                        insideFootR.setText(
String.valueOf(insideFootRC));
                    }
                    if(inString.contains("t"))
                    {
                        topFootRC++;
                        topFootR.setText(
String.valueOf(topFootRC));
                    }
                    if(inString.contains("o"))
                    {
                        outsideFootRC++;
                        outsideFootR.setText(
String.valueOf(outsideFootRC));
                    }
                }
            }
            recDataString.delete(0,
recDataString.length());
            inString = null;
        }
    }
}

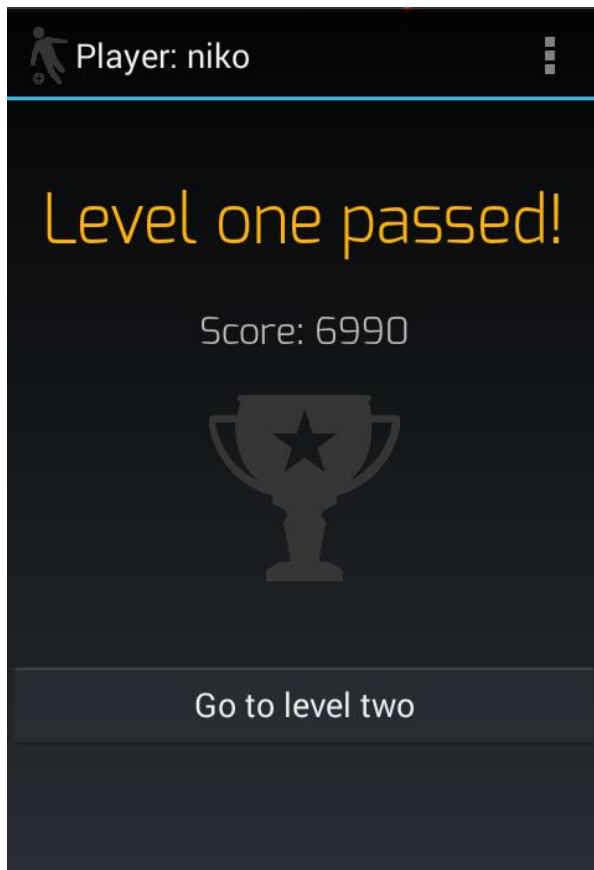
```

Laskurimuuttujien arvot voidaan näyttää käyttäjälle graafisessa käyttöliittymässä ja pallonhallintapelin idea perustuu näiden arvojen vertailuun keskenään ja ajastinta vastaan. Kuviossa 16 on kuvakaappaus pallonhallintapelin ensimmäisestä tasosta, jossa palloa tulee pomputtaa jalalla kolme kertaa, ennen kuin aika loppuu. Kyseisessä tasossa ei ole väliä millä jalanosalla palloon

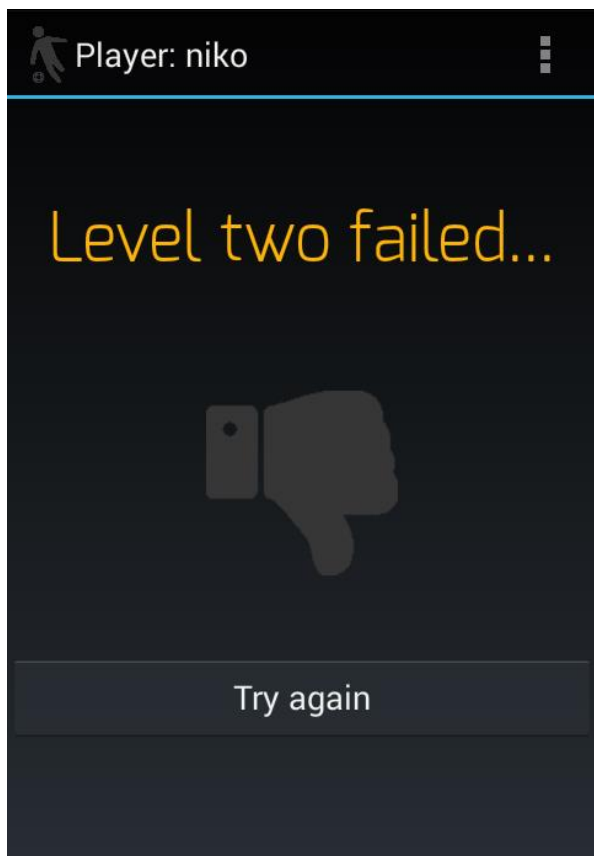
koskee. Kun käyttäjä läpäisee tason, tulee suoritetusta tasosta saatu piste-
määrä näkyviin ja käyttäjä voi jatkaa seuraavaan tasoon (ks. kuvio 17). Jos
käyttäjä ei läpäise tasoa, tulee hänen yrittää uudelleen ja läpäistä taso, jos ha-
luaa edetä pelissä (ks. kuvio 18).



Kuvio 16. Kuvakaappaus pelin ensimmäisestä tasosta



Kuvio 17. Tason läpäistyään käyttäjä näkee pisteensä



Kuvio 18. Näkymä, jos käyttäjä ei läpäise tasoa

7 Opinnäytetyön tulokset

7.1 Onnistumiset

Opinnäytetyön suurimpana onnistumisena voidaan pitää toimivaa prototyyppisukkaa sekä Android-sovellusta, jotka toimivat kuten ennalta oli suunniteltu. Laitteiston prototyypin kehittämisessä oli tarkoituksena selvittää tuoteidean teknisen toteutuksen haastavuutta ja mahdollisia ongelmakohtia. Opinnäytetyössä valmistettu prototyyppi palveli hyvin tätä tarkoitusta ja mahdollista jatkotuotekehitystä ajatellen, se antoi paljon tietoa siitä, kuinka jatkokehitystyötä olisi järkevintä lähteä viemään eteenpäin.

Toimiva BLE-yhteys mikrokontrollerin ja Android-sovelluksen välillä onnistui myös suunnitelmien mukaan ja se toimi luotettavasti. Yhteyden luominen Android-sovellukseen oli kuitenkin hyvin työlästä, sillä tällä hetkellä olemassa oleva dokumentaatio liittyen Bluetooth low energy -yhteyden ohjelmoimisesta Android-sovellukseen on kehno. Tämän takia yhteyden ohjelmointiin kului huomattava määrä aikaa ja ohjelmointi täytyi suurelta osin tehdä yrityksen ja erehdyksen kautta. Tämä toisaalta johti monen uuden asian oppimiseen ja syvällisempään ymmärrykseen BLE-yhteyden mahdollisuuksista ja puutteista.

7.2 Puutteet ja ongelmat

Opinnäytetyön lopputuloksena asetetut tavoitteet saavutettiin tyydyttävästi, mutta joitain tavoitteita jäi saavuttamatta. Algoritmien toimintavarmuutta todellisessa käyttötilanteessa on vaikea arvioida, sillä niiden luotettavuutta ei ehditty aikatauluongelmien vuoksi testaamaan kunnollisesti todellisessa

käyttötilanteessa. Algoritmien toimintavarmuus oli kuitenkin hyvä testausolosuhteissa, jolloin ne toimivat kuten oli tarkoitus. Testausolosuhteet ja oikea käyttötilanne eivät kuitenkaan ole suoraan verrattavissa keskenään, sillä sukan antureihin kohdistuu todellisessa käyttötilanteessa useita erilaisia voimia eri suunnista, joita on hyvin vaikea simuloida testaustilanteessa. Sukan toimintaan vaikuttaa myös se, kuinka sukka on asetettuna käyttäjän jalkaan. Jos sukka on huonosti asetettuna jalkaan, saattavat anturit olla väärissä kohdissa, jolloin ne eivät reagoi pallokosketuksiin ja askeleisiin algoritmien vaatimalla tavalla. Tätä ongelmaa lopullisessa tuotteessa voisi pyrkiä ehkäisemään esimerkiksi sukan muotoilulla ja kuvioilla kankaassa.

Yhtenä suurena ongelmana laitteen testauksessa oli saada elektroniikka valmistettua siten, että se kestäisi fyysistä rasitusta. Käyttäjän juostessa ja liikkeessa kohdistuu askelluksista suhteellisen kovia iskuvoimia elektroniikkakomponentteihin. Elektroniikka oli projektiryhmän itse rakentama, joten juotosliitosten mekaaninen kestävyys oli yksi ongelmakohta. Elektroniikan rakentamisessa tuli huomioida erityisesti se, että sen koko pysyy pienenä ja että komponentit kestävätkä toistuvia tärähdyksiä. Itse valmistettu elektroniikka kesti kuitenkin hyvin testaustilanteissa, vaikka välillä pienimuotoisia kosketushäiriöitä saattoi esiintyä.

Android-sovellusta ei ehditty kehittämään niin pitkälle, että se olisi tukenut yhteyttä kahteen älysuikkaan yhtä aikaa. Lopullisen tuotteen olisi tarkoitus olla sukkapari, joista molemmista välittyy dataa Android-sovellukselle. Sukan akunkäyttöä ei myöskään ehditty tutkimaan, joten toistaiseksi jäi epäselväksi riittäisikö akussa varaus esimerkiksi kokonaisen jalkapallopelin ajaksi. Jos varaus ei riitä, tulisi miettiä keinoja tehokkaammalle akun käytölle tai käyttäjä akkua, jossa on suurempi varauskapasiteetti. Akun koko ei saisi kuitenkaan kasvaa nykyisestään.

Jos tuoteideaa lähdetäisiin jatkokehittämään, olisi mahdollisesti seuraavina kehitettävänä ominaisuuksina askeltiheyden laskuri, askeleen kontaktiaikalaskuri ja tieto jalkapallon potkun voimakkuudesta.

8 Pohdinta

Näillä näkymin projektiryhmä laittaa tuotekehitysprojektin jäihin. Syy miksi tuotekehitystä ei todennäköisesti enää jatketa, on että projektiryhmän tekemien alustavien markkinatutkimusten ja laitteen teknisen toteutuksen haastavuuden takia tuotteesta olisi hyvin haastavaa saada liiketoiminnallisesti menestyvä tuote. Vaikka nuoria jalkapallon harrastajia on maailmassa miljoonia, on asiakkaiden odotusten ja tuotteen hinnan saaminen sopusuhteeseen keskenään toistaiseksi iso haaste. Tuotekehityksen vaatimat resurssit nostaisivat jalkapalloälyskan hinnan niin korkealle tasolle, että suurten asiakasmäärien hankinta olisi todennäköisesti sen vuoksi hankalaa. Puettavan teknologian suosio kasvaa kuitenkin jatkuvasti ja tekniikka halventuu, joten vastaaminen asiakkaiden tarpeisiin ja vaatimuksiin voi lähitulevaisuudessa hyvinkin olla mahdollista huomattavasti pienemmillä resursseilla.

Toiseksi ongelmaksi tuotteen jatkokehityksen kannalta nousee immateriaalioikeudet. Projektiryhmän tekemän omatoimisen patenttien selvityksen pohjalta voidaan sanoa, että jalkapalloälyskassa käytetty tekniikka ei suoranaisesti riko voimassaolevia patenteja. Ongelmaksi kuitenkin muodostuvat vireillä olevat patenttihakemukset, joiden immateriaalioikeuksia jalkapalloälyskassa käytettävä tekniikka rikkoisi, mikäli nämä patenttihakemukset tulisivat voimassaoleviksi patenteiksi. Patenttihakemukset usein muotoillaan mahdollisimman ympäröyöreiksi ja

kaikenkattaviksi, mutta varsinkin tekniset ratkaisut, joita patentti suojaa, tarkentuvat patenttihakemuksen mahdollisen hyväksymisen myötä, jolloin myös patentin suojan laajuus usein kaventuu.

Opinnäytetyön aikana valmistettu prototyyppi älylukasta osoittaa, että suhteellisen pienillä resursseilla voi valmistaa toimivan prototyypin tuoteideasta. Prototyyppi ei tietenkään ole ominaisuuksiltaan tai laadultaan lopullisen tuotteen tasoinen, mutta se antaa arvokasta tietoa siitä, minkälaisia ongelmia lopullisessa tuotteessa mahdollisesti olisi. Näiden ongelmien havaitseminen tuotekehityksen varhaisessa vaiheessa on hyvin olennainen osa mitä tahansa tuotekehitysprojektiä. Puettavan teknologian tuotteilla on paljon mahdollisia käyttösovelluksia, joita ei nykypäivänä vielä hyödynnetä. Useat yritykset ovat nähneet puettavan teknologian mahdollisuudet ja kilpailu puettavan teknologian markkinoilla on kiihtynyt huimasti viime vuosien aikana. Kova kilpailu karsii heikoimmat tuoteideat pois, mutta ne tuoteideat jotka pärjäävät kovassa kilpailutilanteessa, tulevat todennäköisesti toimimaan suunnannäyttäjinä seuraavan sukupolven älykkäälle teknologialle.

Kokonaisuudessaan opinnäytetyön tekeminen oli haastava ja pitkä prosessi, mutta suurimpana motivoivana tekijänä työssä oli onnistumisen tunne, kun jonkin haastavan asian sai ratkaistua pitkän yrityksen jälkeen. Ongelmien ratkaiseminen ja tilaisuus päästä soveltamaan omia taitojaan hyvin monipuolisesti, olivat kaikkein antoisimmat kokemukset opinnäytetyöprosessissa. Antoisaa oli myös se, että kyseessä oli projektiryhmän oma tuotekehityskohde, jolloin pystyi itse vaikuttamaan, mihin asioihin opinnäytetyössä halusi keskittyä.

Lähteet

Activities. N.d. Aktiviteettikomponentin dokumentaatio Androidin kehittäjä sivustolla. Viitattu 8.4.2015.

<http://developer.android.com/guide/components/activities.html>.

Activity. N.d. Android Developers -sivuston julkaisema kuva. Viitattu

23.2.2015. <http://developer.android.com/reference/android/app/Activity.html>.

Android Developers. 2015. Android-kehittäjien sivusto. Viitattu 23.2.2015.

<Http://developer.android.com/>.

Ball, K. 2008. Science and Football VI: The Proceedings of the Sixth World Congress on Science and Football. Viitattu 12.4.2015.

<http://www.wata.cc/up/2012/07/files/w-b6c9afb540.pdf>.

Bluetooth. 2015. Bluetooth-kehittäjien sivusto. Viitattu 16.3.2015.

<http://www.bluetooth.com/Pages/History-of-Bluetooth.aspx>.

Bluetooth Technology Basics. N.d. Bluetoothin virallinen sivusto. Viitattu

16.3.2015. <http://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics>.

BroadcastReceiver. N.d. Vastaanottajakomponentin dokumentaatio Androidin kehittäjä sivustolla. Viitattu 8.4.2015.

<http://developer.android.com/reference/android/content/BroadcastReceiver.html>.

Content Providers. N.d. Palveluntarjoajakomponentin dokumentaatio Androidin kehittäjä sivustolla. Viitattu 8.4.2015.

<http://developer.android.com/guide/topics/providers/content-providers.html>.

Cormen, T., Leiserson, C., Rivest, R. & Stein, C. 2001. Introduction to Algorithms, Second Edition. Vilnan yliopiston julkaisema tiedosto. Viitattu 30.11.2015.

<http://www.mif.vu.lt/~valdas/ALGORITMAI/LITERATURA/Cormen/Cormen.pdf>.

Global Mobile Data Traffic Forecast Update. 2015. Ciscon 3.2.2015 julkaisema artikkeli. Viitattu 27.7.2015.

http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html.

Halkola, J. 2014. Androidilla toteutettu liikuntapäiväkirjasovellus. Opinnäytetyö. Metropolia ammattikorkeakoulu, tietotekniikka. Viitattu 30.3.2015.

https://www.theseus.fi/bitstream/handle/10024/83838/insinoorityo_halkola_VALMIS_theseus.pdf?sequence=1.

Human Genome Project. N.d. Human Genome Project Information Archive - sivuston etusivu. Viitattu 30.11.2015.

http://web.ornl.gov/sci/techresources/Human_Genome/index.shtml.

IEEE 802.11. 2012. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Standards Association. Vahvistettu 6.2.2012. Viitattu 5.4.2015.

<http://standards.ieee.org/getieee802/download/802.11-2012.pdf>.

Innovaatiot. N.d. Polar.com-sivuston Innovaatiot-sivu. Viitattu 10.12.2015.

http://www.polar.com/fi/tietoa_polarista/keita_olemme/innovaatiot.

Juola, I. 2011. Mobiilisovellusten kehittäminen. Opinnäytetyö. Jyväskylän ammattikorkeakoulu, tietojenkäsittely. Viitattu 3.4.2015.

https://www.theseus.fi/bitstream/handle/10024/27039/juola_ilkka.pdf?sequence=1

Kobie, N. 2011. Android tops ailing Nokia's Symbian. Alphr-sivustolla 31.1.2011 julkaistu artikkeli. Viitattu 27.7.2015.

<http://www.alphr.com/news/364816/android-tops-ailing-nokias-symbian>.

Kolehmainen, K. 2006. C++-ohjelmointi – Algoritmit ja mallit. Jyväskylä: readme.fi.

Korkiakoski, P. 2013. Minigolf-tulossovelluksen ohjelmointi Androidille. Opinnäytetyö. Oulun seudun ammattikorkeakoulu, tietojenkäsittelyn koulutusohjelma. Viitattu 3.4.2015.

https://www.theseus.fi/bitstream/handle/10024/66456/Korkiakoski_Pipsa.pdf?sequence=1.

Ligertwood, P. & Povalac, K. 2012. Low Power Considerations for ZigBee Applications Operated by Coin Cell Batteries. Freescale Semiconductor, Inc. - yhtiön julkaisema tiedosto. Viitattu 4.12.2015.

http://cache.freescale.com/files/rf_if/doc/app_note/AN4573.pdf.

Maman, P., Kanupriya, G. & Jaspal, S. 2012. Role of Biofeedback in Optimizing Psychomotor Performance in Sports. Asian Journal of Sports Medicine. Viitattu 18.5.2015.

<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3307964/>.

Method and outfit for measuring of action of muscles of body. 2009. Google Patents -patenttitietokannan patenttijulkaisu. Viitattu 10.12.2015.

<http://www.google.com/patents/EP1531726B1?cl=en>.

Our History. N.d. Bluetoothin virallinen sivusto. Viitattu 16.3.2015.

<http://www.bluetooth.com/media/our-history>.

Puettava älyteknologia mahdollisuus suomalaisille yrityksille. N.d. Export Finlandin julkaisema artikkeli. Viitattu 24.9.2015.

<http://www.exportfinland.fi/markkinamahdollisuus/puettava-alyteknologia>.

Pärnänen, J. 2011. Langattoman tiedonsiirron toteuttaminen mittalaitteeseen. Opinnäytetyö. Savonia-ammattikorkeakoulu, elektroniikan koulutusohjelma. Viitattu 9.3.2015.

https://www.theseus.fi/bitstream/handle/10024/33204/Parnanen_Juha.pdf?sequence=1.

Runteq loi huippuvalmentajan kuntojuoksijoille. 2014. Kauppalehden julkaisema kuva 14.10.2014. Viitattu 30.11.2015.

<http://www.kauppalehti.fi/uutiset/runteq-loi-huippuvalmentajan-kuntojuoksijoille/22qFvTLB>.

Samosuyev, V. 2010. Bluetooth low energy compared to ZigBee and Bluetooth Classic. Opinnäytetyö. Mikkelin ammattikorkeakoulu, informaatiotekniikan koulutusohjelma. Viitattu 30.3.2015.

https://www.theseus.fi/bitstream/handle/10024/15812/FinalThesis_Samosuyev.pdf?sequence=1.

Sensoria Is A New Smart Sock That Coaches Runners In Real Time. 2014. Techcrunch-sivuston julkaisema kuva 7.1.2014. Viitattu 30.11.2015.

<http://techcrunch.com/2014/01/07/sensoria-is-a-new-smart-sock-that-coaches-runners-in-real-time/>.

Service. N.d. Palvelukomponentin dokumentaatio Androidin kehittäjä sivustolla. Viitattu 8.4.2015.

<http://developer.android.com/reference/android/app/Service.html#WhatIsAService>.

Services. N.d. Palvelukomponentin dokumentaatio Androidin kehittäjä sivustolla. Viitattu 8.4.2015.

<http://developer.android.com/guide/components/services.html>.

Sintonen, A. 2012. ZigBee – langaton verkkostandardi. Opinnäytetyö. Metropolia ammattikorkeakoulu, tietotekniikan koulutusohjelma. Viitattu 7.4.2015.

https://www.theseus.fi/bitstream/handle/10024/46085/antti_sintonen.pdf?sequence=1.

Smith, P. 2011. Comparing Low-Power Wireless Technologies. Digikey-sivuston julkaisema artikkeli 8.8.2011. Viitattu 4.12.2015.

<http://www.digikey.com/en/articles/techzone/2011/aug/comparing-low-power-wireless-technologies>.

Sourulahti, S. 2015. Käyttöliittymäsuunnittelu älysukan mobiilisovellukselle. Opinnäytetyö. Jyväskylän ammattikorkeakoulu, hyvinvointitekniikan koulutusohjelma. Viitattu 30.11.2015.

https://www.theseus.fi/bitstream/handle/10024/96285/Sourulahti_Saku.pdf?sequence=1.

Thitz, H. 2015. FSR-tyyppisten kosketusanturien integrointi tekstiiliin. Opinnäytetyö. Jyväskylän ammattikorkeakoulu, hyvinvointitekniikan koulutusohjelma. Viitattu 30.11.2015.

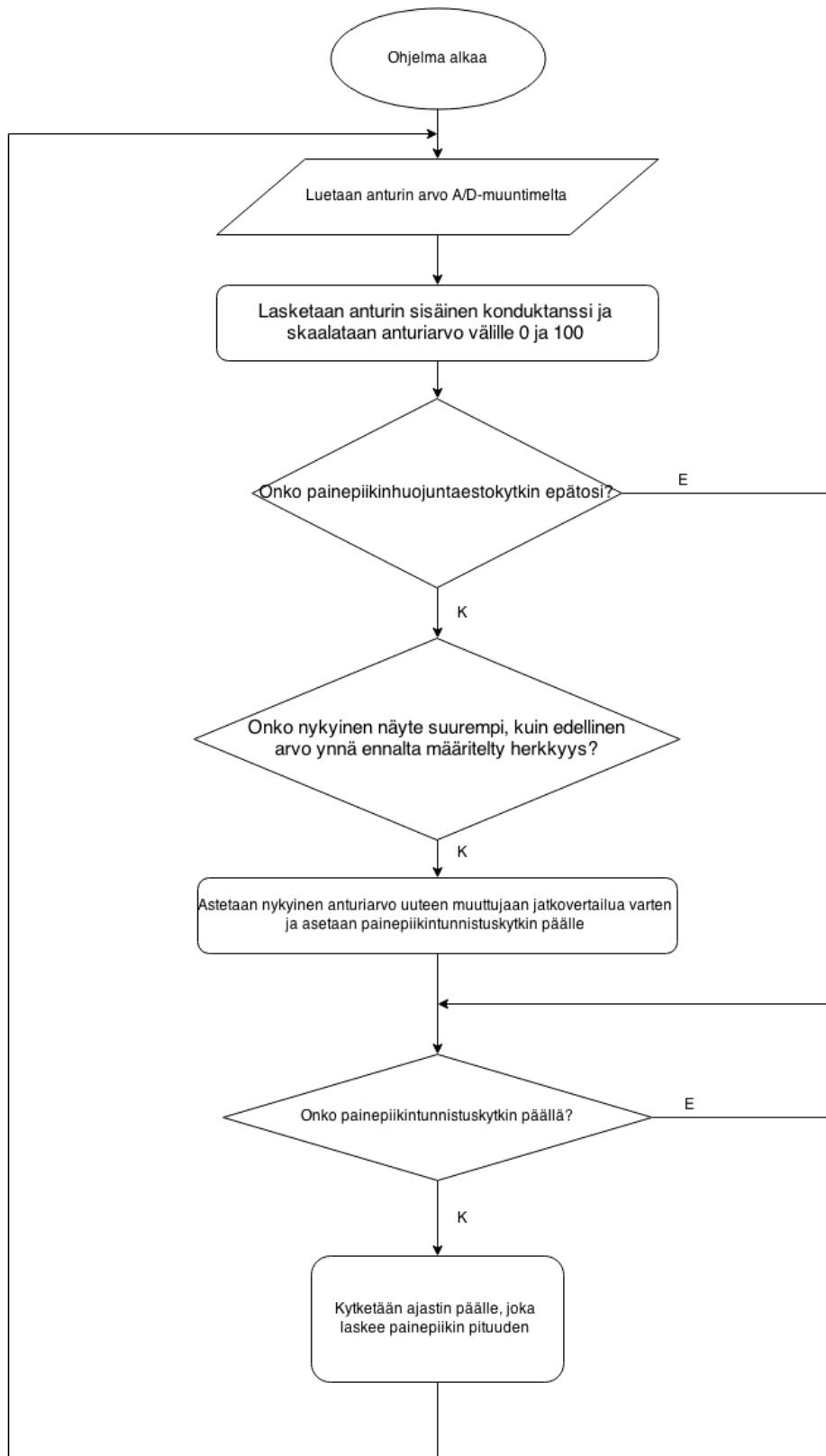
https://www.theseus.fi/bitstream/handle/10024/95881/Thitz_Henri.pdf?sequence=1.

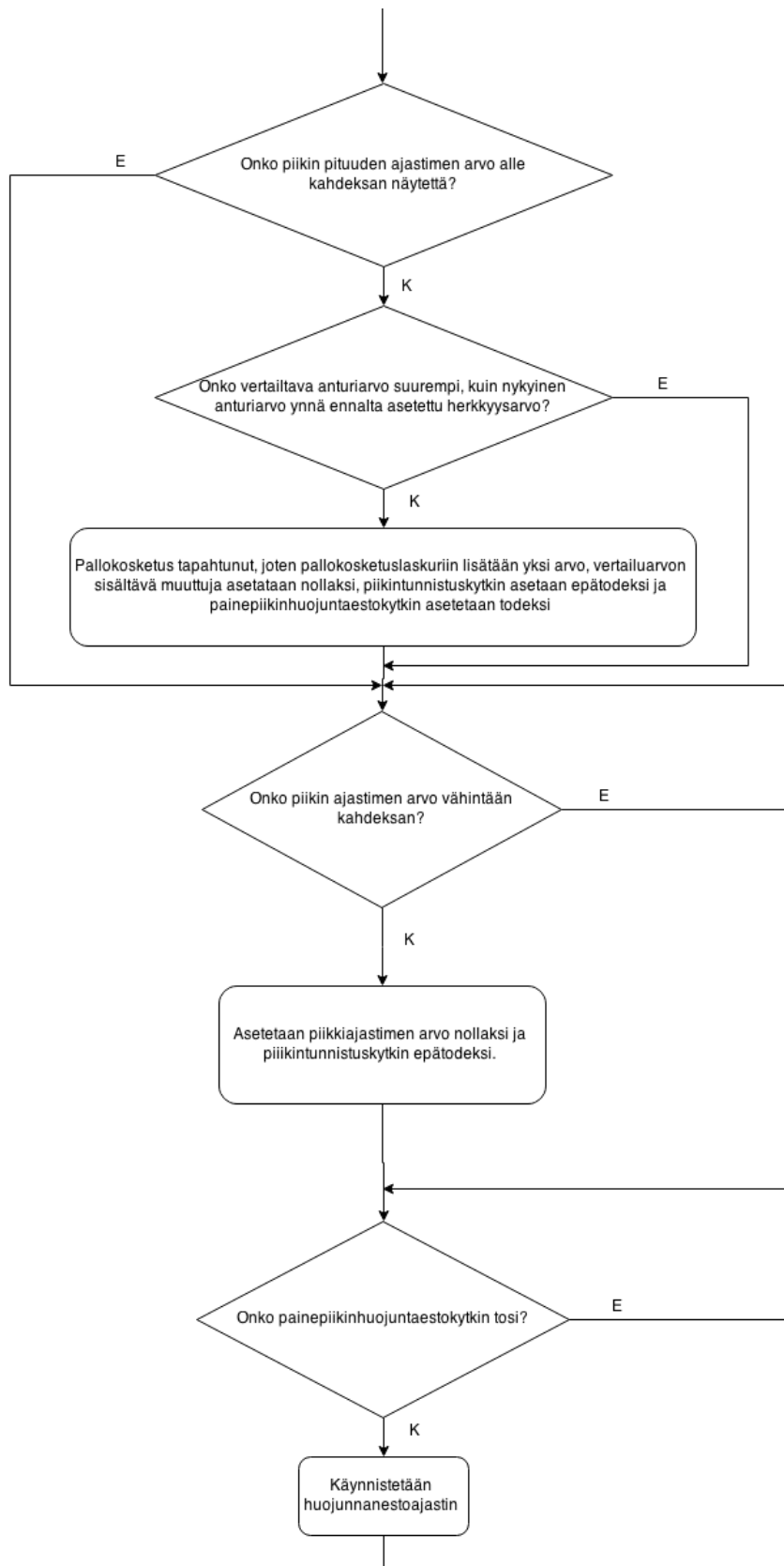
Thorp, E. N.d. The Invention of the First Wearable Computer. Columbian yliopiston julkaisema tiedosto. Viitattu 23.9.2015.

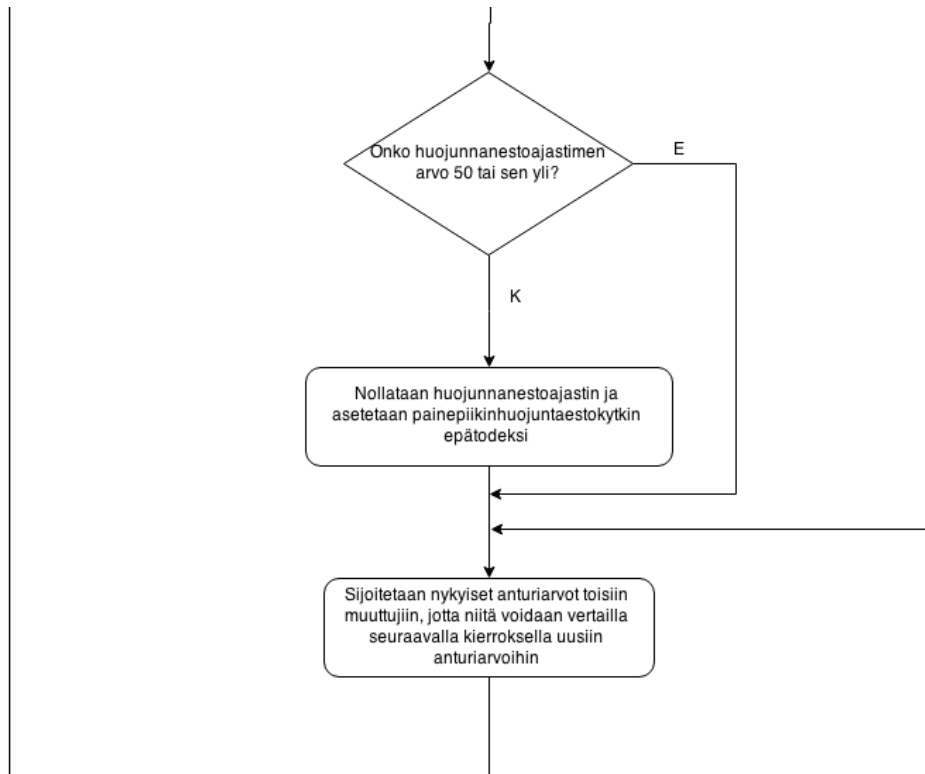
<http://monet.cs.columbia.edu/courses/mobwear/resources/thorp-iswc98.pdf>.

Liitteet

Liite 1. Pallokosketuksen tunnistusalgoritmin vuokaavio







Liite 2. Askeleen tunnistusalgoritmin vuokaavio

